

TECHNICAL BULLETIN

[Issue No.] T08-0006

[Page] 1/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E Basic Communication Support Tool
MELSEC-A Series Ethernet Interface Module, Computer Link Module
MELSEC-QnA Series Ethernet Interface Module,
Serial Communications Module

Thank you for your continued patronage of the Mitsubishi general-purpose programmable logic controller MELSEC-A/QnA Series.

By using the Mitsubishi communication support software tool for supporting communication between the MELSEC-A Series and QnA Series Programmable Logic Controller (hereinafter PLC) and personal computer, the communication program on the personal computer side connected with Ethernet or a computer link, etc., can be simplified.

The following sections describe the outline functions and data link functions of the basic communication support tool (SWnD5F-CSKP-E, hereinafter CSKP-E). The programming methods on personal computer side for reading/writing data to the MELSEC PLC CPU using CSKP-E are also described.

* For details on the CSKP-E, refer to the New Product News (No. 95E) and CSKP-E manual (manual No. is given in New Product News).

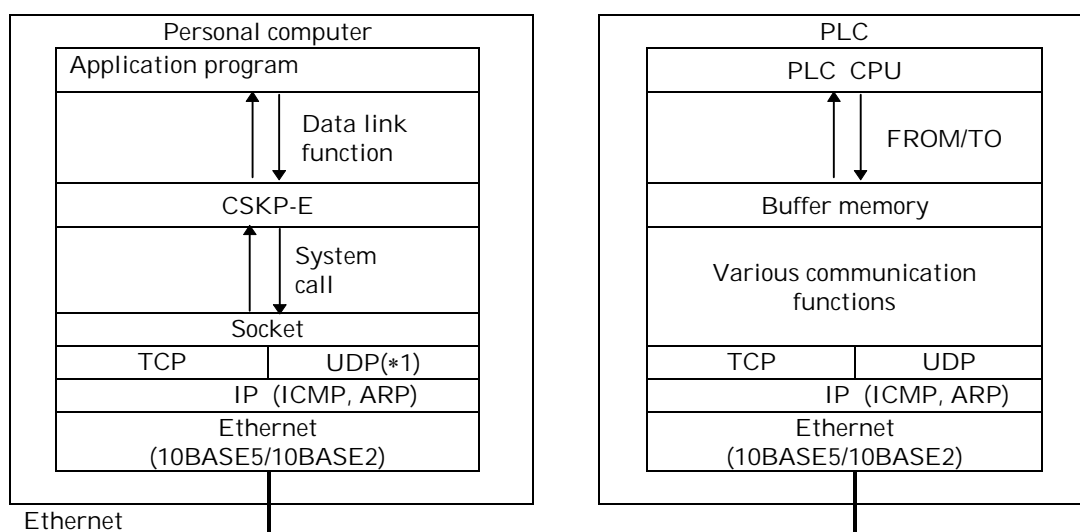
Whereas

[Details]

1. Outline of basic communication support tool (CSKP-E)

(1) The CSKP-E is installed into the personal computer and used.

(Example) Software structure diagram for Ethernet connection



*1 The UDP/IP communication for using CSKP-E will be planned in the future.

TECHNICAL BULLETIN

[Issue No.] T08-0006

[Page] 2/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E

Basic Communication Support Tool

MELSEC-A Series

Ethernet Interface Module, Computer Link Module

MELSEC-QnA Series

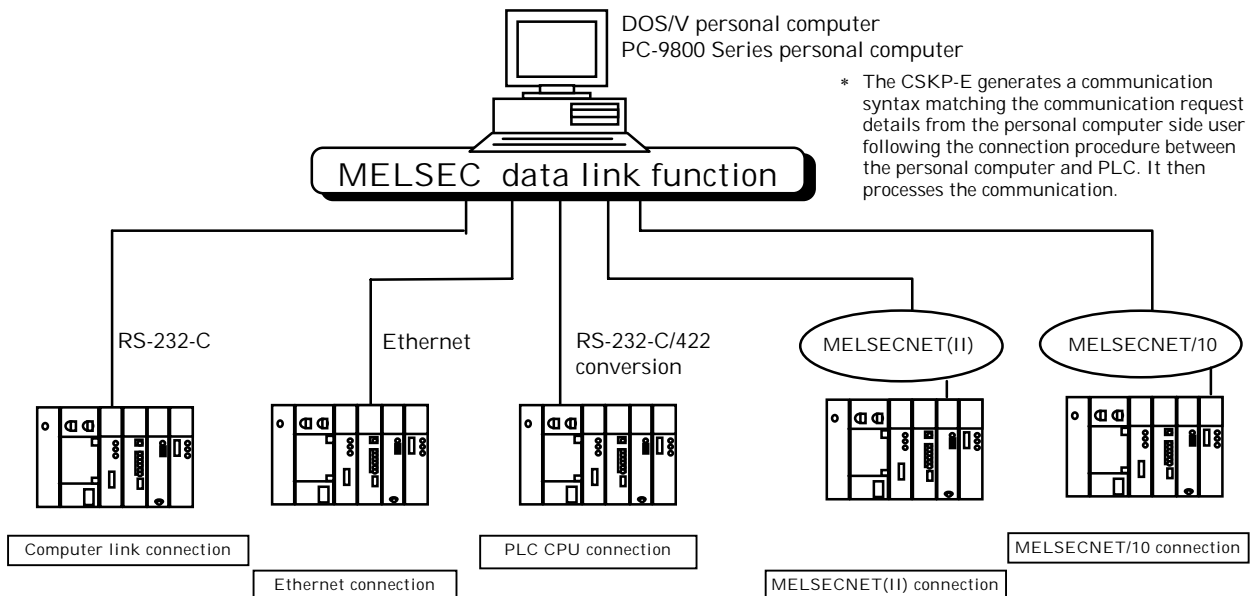
Ethernet Interface Module,
Serial Communications Module

(2) Ample support of communication protocols

The personal computer side user can easily access the PLC's CPU by designating the connection type and communication request details using the following CSKP-E data link function arguments.

* The user does not need to understand the communication syntax unique to the special function module on the PLC side. The connection procedure can be changed just by changing part of the program (mdOpen argument's communication channel No.) in the program created by the user.

Once created, the user program can be used effectively.



(3) Data link functions dedicated for communicating data with PLC are supported

- Functions required for communicating data with the PLC, such as communication line open/close and device read/write, are prepared as data link functions.
- Various communication programs can be created easily, so development of the communication system using Ethernet/serial communication, etc. with the PLC side is easily.

Function name	Data link function's function
mdOpen	Initializes and opens the designated communication line channel.
mdClose	Closes the designated communication line channel.
mdSend	Writes the data in a batch to the designated device.
mdReceive	Reads the data in a batch from the designated device.
mdRandR	Reads data from a randomly designated device.
mdRandW	Writes the data to a randomly designated device.
mdDevSet	Sets (ON) the designated bit device.
mdDevRst	Resets (OFF) the designated bit device.
mdControl	Carries out remote RUN/STOP/PAUSE in respect to the designated PLC CPU.
mdTyperead	Reads the type of designated PLC CPU.

[Issue No.] T08-0006

[Page] 3/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E

Basic Communication Support Tool

MELSEC-A Series

Ethernet Interface Module, Computer Link Module

MELSEC-QnA Series

Ethernet Interface Module,
Serial Communications Module

(4) Various utilities are enclosed

The communication between the PLC and personal computer can be tested and monitored with the communication diagnosis utility and device monitor utility enclosed with the product.

2. Examples of using basic communication support tool (CSKP-E)

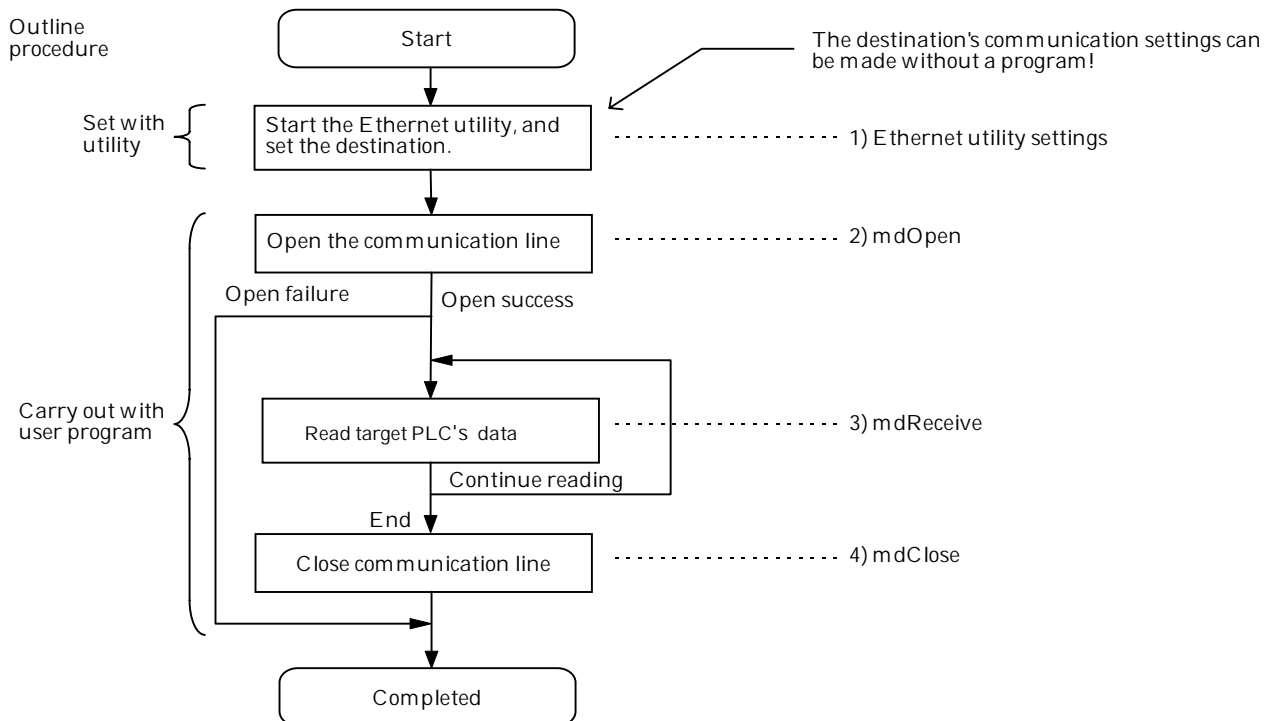
Programming procedures and program examples for connecting to the PLC's Ethernet interface module and computer link module and accessing the PLC's CPU from a personal computer are described in this section.

For the program example, the personal computer reads the data register (D) in the A/QnA Series PLC CPU, and displays the results on the screen.

2.1 Examples of using with Ethernet connection

(1) Programming procedures (Outline flow of procedures)

The flow of the data read process for creating a personal computer side program to access the PLC CPU using CSKP-E is shown below.



1) Ethernet utility settings

Set the module type (E71, QE71), host name (IP address) and port No., etc., of the PLC to be communicated with over an Ethernet connection. This information is saved as a logic station No.

TECHNICAL BULLETIN

[Issue No.] T08-0006

[Page] 4/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E Basic Communication Support Tool
MELSEC-A Series Ethernet Interface Module, Computer Link Module
MELSEC-QnA Series Ethernet Interface Module,
Serial Communications Module

2) mdOpen

This function initializes and opens the Ethernet communication line channel (No. = 61).
The path of the opened communication line is returned as the execution results.
The path is required to execute the function (mdReceive) for reading the PLC data, and the function (mdClose) for closing the line.

3) mdReceive

This function reads the PLC CPU's device data.
The target PLC's logic station No., device to be read and No. of points are assigned to the argument.

4) mdClose

This function closes the opened communication line channel.
The communication line must be closed when reading and writing are completed.

(2) Programming example (Using Visual C++)

The D0 and D1 values of the target PLC CPU are read over the logic station No. 0 communication line, and are displayed on the screen.

```
/******  
* Include *  
*****/  
#include <stdio.h>  
#include <windows.h> /* Windows include file */  
#include "mdfunc.h" /* Data line function include file */  
/******  
* Constant declaration *  
*****/  
#define CHSN_ETHERNET 61 /* Ethernet communication channel (common for E71, QE71)*/  
#define MODE_DUMMY -1 /* Dummy (fixed to -1) */  
#define STNO_ROGIC 0 /* Logic station No. */  
#define DEVTYPE_D 13 /* Device type "D" */  
#define DEVNO_0 0 /* Read top device No. */  
#define SIZE_R_BYTES 4 /* No. of read data bytes */  
#define DATA_INITIAL 0 /* Read area initial value */  
/******  
* The logic station No. "0" PLC data is read. *  
* The logic station No. is preset with the Ethernet utility. *  
*****/  
void main()  
{  
 /* Declare mdOpen argument */  
 long path; /* Path save function */  
 short chan; /* Communication line channel No. */  
 short mode; /* Dummy (fixed to -1) */  
 short oret; /* mdOpen function return value */  
 /* Declare mdReceive argument */  
 short stno; /* Station No. */  
 short devtyp; /* Device type */
```

TECHNICAL BULLETIN

[Issue No.] T08-0006

[Page] 5/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E Basic Communication Support Tool
MELSEC-A Series Ethernet Interface Module, Computer Link Module
MELSEC-QnA Series Ethernet Interface Module,
Serial Communications Module

```
short devno; /* Read top device No. */
short size; /* No. of read data bytes */
short data[2]; /* Read data buffer */
short rret; /* mdReceive function return value */
/* Declare mdClose argument */
short cret; /* mdClose function return value */
/* Declare mdOpen argument setting */
chan = CHSN_ETHERNET; /* Ethernet communication channel */
mode = MODE_DUMMY; /* Dummy (fixed to -1) */
/* Execute mdOpen and open line */
oret = mdOpen( chan, mode, &path );
if( oret != 0 ){
    /* If function fails, return error */
    printf( "mdOpen error[%04x]\n", oret );
}
else{
    /* Continue process only when mdOpen succeeds. */
    /* Set mdReceive argument */
    stno = STNO_ROGIC; /* Logic station No. */
    devtyp = DEVTYPE_D; /* Device type "D" */
    devno = DEVNO_0; /* Read device No. 0, 1 */
    size = SIZE_R_BYTES; /* 2-point, 4-byte (word device is 1-point 2-byte) */
    data[0] = DATA_INITIAL; /* Initialize read data area */
    data[1] = DATA_INITIAL; /* Initialize read data area */
    /* Execute mdReceive and read */
    /* Use path acquired with mdOpen */
    rret = mdReceive( path, stno, devtyp, devno, &size, &data );
    if( rret != 0 ){
        /* If function fails, return error */
        printf( "mdReceive error[%04x]\n", rret );
    }
    else{
        /* Display read data */
        printf( "data0 [%04x]\n", data[0] ); /* Data display */
        printf( "data1 [%04x]\n", data[1] ); /* Data display */
    }
    /* Execute mdClose and close line */
    /* Use path acquired with mdOpen */
    cret = mdClose( path );
    if( cret != 0 ){
        /* If function fails, return error */
        printf( "mdClose error[%04x]\n", cret );
    }
}
}
```

[Issue No.] T08-0006

[Page] 6/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E

Basic Communication Support Tool

MELSEC-A Series

Ethernet Interface Module, Computer Link Module

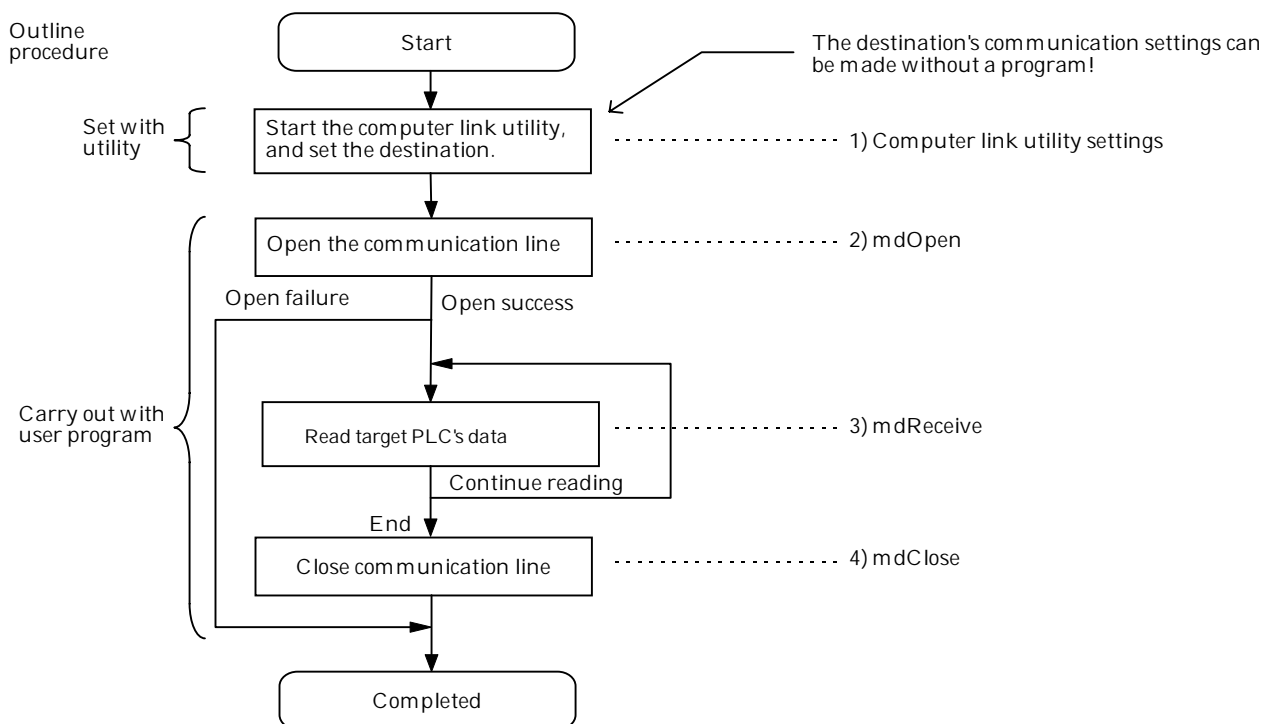
MELSEC-QnA Series

Ethernet Interface Module,
Serial Communications Module

2.2 Examples of using with computer link connection

(1) Programming procedures (Outline flow of procedures)

The flow of the data read process for creating a personal computer side program to access the PLC CPU using CSKP-E is shown below.



1) Computer link utility settings

Set the computer link module/serial communication module type (C24, UC24, QC24), module station No., and COM port communication conditions of the PLC to be communicated with over a computer link connection. This information is saved as a logic station No.

2) mdOpen

This function initializes and opens the serial communication line channel (serial port (COM port) channel No. = 31 to 40).

The path of the opened communication line is returned as the execution results.

The path is required to execute the function (mdReceive) for reading the PLC data, and the function (mdClose) for closing the line.

3) mdReceive

This function reads the PLC CPU's device data.

The target PLC's logic station No., device to be read and No. of points are assigned to the argument.

4) mdClose

This function closes the opened communication line channel.

The communication line must be closed when reading and writing are completed.

TECHNICAL BULLETIN

[Issue No.] T08-0006

[Page] 7/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E Basic Communication Support Tool
MELSEC-A Series Ethernet Interface Module, Computer Link Module
MELSEC-QnA Series Ethernet Interface Module,
Serial Communications Module

(2) Programming example (Using Visual C++)

The D0 and D1 values of the target PLC CPU are read over the logic station No. 0 communication line, and are displayed on the screen.

```
/******  
* Include *  
*****/  
#include <stdio.h>  
#include <windows.h> /* Windows include file */  
#include "mdfunc.h" /* Data line function include file */  
/******  
* Constant declaration *  
*****/  
#define CHSN_COM1 31 /* Computer link communication channel (common  
for C24, UC24, QC24) */  
#define MODE_DUMMY -1 /* Dummy (fixed to -1) */  
#define STNO_ROGIC 0 /* Logic station No. */  
#define DEVTYPE_D 13 /* Device type "D" */  
#define DEVNO_0 0 /* Read top device No. */  
#define SIZE_R_BYTES 4 /* No. of read data bytes */  
#define DATA_INITIAL 0 /* Read area initial value */  
/******  
* The logic station No. "0" PLC data is read. *  
* The logic station No. is preset with the computer link utility. *  
*****/  
void main()  
{  
 /* Declare mdOpen argument */  
 long path; /* Path save function */  
 short chan; /* Communication line channel No. */  
 short mode; /* Dummy (fixed to -1) */  
 short oret; /* mdOpen function return value */  
 /* Declare mdReceive argument */  
 short stno; /* Station No. */  
 short devtyp; /* Device type */  
 short devno; /* Read top device No. */  
 short size; /* No. of read data bytes */  
 short data[2]; /* Read data buffer */  
 short rret; /* mdReceive function return value */  
 /* Declare mdClose argument */  
 short cret; /* mdClose function return value */  
 /* Declare mdOpen argument setting */  
 chan = CHSN_COM1; /* Computer link communication channel */  
 mode = MODE_DUMMY; /* Dummy (fixed to -1) */  
 /* Execute mdOpen and open line */  
 oret = mdOpen( chan, mode, &path );
```

TECHNICAL BULLETIN

[Issue No.] T08-0006

[Page] 8/8

[Title] MELSEC Communication Support Software Tool

[Date of Issue] Oct. '98

[Relevant Models] SWnD5F-CSKP-E Basic Communication Support Tool
 MELSEC-A Series Ethernet Interface Module, Computer Link Module
 MELSEC-QnA Series Ethernet Interface Module,
 Serial Communications Module

```

if( oret != 0){
    /* If function fails, return error          */
    printf( "mdOpen error[%04x]\n", oret );
}else{
    /* Continue process only when mdOpen succeeds. */
    /* Set mdReceive argument          */
    stno   = STNO_ROGIC;                /* Logic station No.          */
    devtyp = DEVTYPE_D;                /* Device type "D"           */
    devno  = DEVNO_0;                  /* Read device No. 0, 1      */
    size   = SIZE_R_BYTES;             /* 2-point, 4-byte (word device is 1-point 2-byte)*/
    data[0] = DATA_INITIAL;           /* Initialize read data area  */
    data[1] = DATA_INITIAL;           /* Initialize read data area  */
    /* Execute mdReceive and read          */
    /* Use path acquired with mdOpen      */
    rret = mdReceive( path, stno, devtyp, devno, &size, &data );
    if( rret != 0){
        /* If function fails, return error          */
        printf( "mdReceive error[%04x]\n", rret );
    }else{
        /* Display read data          */
        printf( "data0 [%4x]\n", data[0] ); /* Data display */
        printf( "data1 [%4x]\n", data[1] ); /* Data display */
    }
    /* Execute mdClose and close line      */
    /* Use path acquired with mdOpen      */
    cret = mdClose( path );
    if( cret != 0){
        /* If function fails, return error          */
        printf( "mdClose error[%04x]\n", cret );
    }
}
}
}

```

3. Basic communication support tool (CSKP-E) operation environment (outline)

Product name	CSKP-E operation environment
CPU (memory)*1	Pentium 100MHz or more (32MB or more)
Comaptible OS	English Windows 95/98, or English Windows NT. Workstation 4.0*2
Programming software	Visual Basic 4.0/5.0, Visual C++4.2/5.0

*1. A multiprocessor compatible personal computer cannot be used.

*2. Service Pack 3 and above is required.

Microsoft Windows, Microsoft Visual Basic, Microsoft Visual C++ are registered trademarks of Microsoft Corp.
 Ethernet is a registered trademark of XEROX, Corp.
 All other company names and product names in this manual are the trademarks or registered trademarks of the respective companies.