

SECS/PLC Communication Connection
(tdiSPLC)

(Trust Design Simple SECS/PLC Communication Connection)

Instruction Manual

Version 15.120 : 2015.12.04
Version 16.011 : 2016.01.15
Version 16.012 : 2016.02.25
Version 16.040 : 2016.04.05
Version 18.032 : 2018.03.28
Version 18.041 : 2018.04.23
Version 19.070 : 2019.07.05
Version 19.100 : 2019.10.25

Trust Design Limited Liability Company

Chino City Nagano Prefecture Japan

E-mail: info@trust-design.co.jp
URL: <http://www.trust-design.co.jp>

T a b l e o f C o n t e n t s

1. Introduction	1
2. PLC communication specification and setting file description	3
3. Operation explanation	16

1. Introduction

This program connects the PLC control device to the host system using the SECS protocol.

This program has the following features.

- + It is an application that runs on a Windows PC to connect device controlled by PLC to host system using SECS protocol.
- + The data generated in the PLC can be converted to S6F11 etc. SECS message and notified to the upper layer, or can be indicated by setting the S2F49 instruction from the upper layer in the memory in the PLC.
- + By specifying a pseudo PLC device, you can easily connect a Windows AP to the host above with only access to the shared memory and without performing processing related to SECS.
- + Multiple program can be operated in one PC, and connection between host and multiple PLC device is possible.
- + After setting and checking operation, if automatic start setting of this program is performed to Windows, it is possible to carry out automatic operation by turning on power of Windows PC which runs this program thereafter.
- + The following PLC devices are currently available.
 - Mitsubishi MELSEC MC Protocol (3E, 4E Frame)
 - Omron OMRON FINS Command
 - Keyence KEYENCE TCP/IP
 - Pseudo PLC Data exchange as shared memory as pseudo PLC memory

[Note 1] -----+
| This program is an implementation example for connecting PLC device to a host system with |
| SECS protocol. |
| In addition to the implementation items by this program, I think that there are functions |
| that you need to implement in order to connect the PLC device used in the actual |
| manufacturing site etc. to the host system by SECS. (For example, variable-length list |
| support, open transaction, data type conversion, heartbeat, error processing, etc.) |
| Since they are considered to be implementation specific to each line and each device, it |
| seems that the function of this program alone is not enough implementation. |
| This program is an example of implementation. |
+-----+

[Note 2] -----+
| When using this program, it is necessary for the customer to implement the processing on |
| the PLC side (in the case of a pseudo PLC device, the Windows program). |
| Refer to this specification 2. (1), and set/acquire necessary information, and turn ON/OFF |
| the flag used for communication. |
+-----+

- + For development of communication system by SECS/HSMS, our SECS/HSMS communication package (Trust Design Simple SECS Communication Library) is available.
For more information, please visit our home page.
- + You can use SECS/HSMS simulator (Trust Design simple SECS/HSMS Simulator (Simplified Version)) as an inspection application to simulate communication processing (host side, device side) by SECS-1 and HSMS-SS, HSMS-GS is released.
For more information, please visit our home page.
- + You can use SECS/HSMS protocol converter program (Trust Design Simple SECS/HSMS Protocol converter) is released.
For more information, please visit our home page.
- + For monitoring communications by SECS (HSMS), our network communications monitor (Trust Design Simple) Network Communication Monitor) is available.
For more information, please visit our home page.
- + For monitoring communications by SECS-1 protocol using RS232C Serial Port, our serial communication monitor (Trust Design Simple Serial Port Communication Monitor) is available.
For more information, please visit our home page.

[Note 3] -----+
| This package uses following ports of UDP/IP for license management. |
| Also use following class D address as UDP/Multicast address. Please set not to block these |
| by firewall etc. of your computer. |
| - 36277/udp |
| - 239.254.200.75 |
| However, even if you can not connect to Internet connection environment, you can use it, and |
| there are no functional restrictions on usage as compared with same environment. |
+-----+

2. PLC communication specification and setting file description

(1) PLC communication specification

(a) PLC device memory area used

(Note) (bss0:PLC) or the like described in the second line of each item is a mnemonic indicating address of each device described in each item, and will be used in the following description
(b). PLC or PC after ':' indicates the device whose device value is to be updated.

1. Bit device for which PLC sets operating status of PLC (1bit) PLCSSDEVO
(ssb0:PLC)
Set to 1 when PLC is in a state where SECS communication with the host is possible, and 0 when communication is not possible.
2. Bit device for which PC sets the operating status of PC (1bit) PLCSSDEV1
(ssb1:PC)
Set to 1 when PC is in SECS connection state with host, and 0 when not connected.
3. Bit device used for handshake when sending SECS message from PLC (2bit) PLCHSDEVO
(hsb00:PLC, hsb01:PC)
When PLC performs SECS message sending, first, information on the SECS message is set in the areas shown in 5 and 7 below, and then PLC setting bit (hsb00) is set to 1. PC detects this, receives information in 5 and 7 areas, and sets PC setting bit (hsb01) to 1 when the SECS message sending is completed.
When PLC confirms that PC setting bit (hsb01) is 1, PLC setting bit (hsb00) is set to 0.
When PC confirms that PLC setting bit (hsb00) is 0, PC setting bit (hsb01) is set to 0.
4. Bit device used for handshake when sending SECS message to PLC (2bit) PLCHSDEV1
(hsb10:PLC, hsb11:PC)
When PC receives SECS Message from upper host, information related to the SECS Message is stored in the areas shown in 6 and 7 below, and PC setting bit (hsb11) is set to 1. When PLC detects this and the information 6 and 7 regions is received and processing is completed, PLC setting bit (hsb10) is set to 1.
After confirming that PLC setting bit (hsb10) is 1, PC sets the PC setting bit (hsb11) to 0.
When PLC confirms that PC setting bit is 0 (hsb11), it sets PLC setting bit (hsb10) to 0.
5. Word device that stores SECS Message information when sending SECS Message from PLC (20W) PLCMSDEVO
Specified address(msd0:PLC) + 0 : SF-Code and W-Bit

FE	8	7	0	
+-----+-----+-----+-----+				
+-----+-----+				
		+-----		F-Code (0 - 127)
	+-----			S-Code (0 - 63)
+-----				W-Bit

+ 1 : DeviceID

FE	0	
+-----+-----+		
+-----+		
		+----- DeiceID (0 - 0x7FFF)
+----- Reserved (0)		

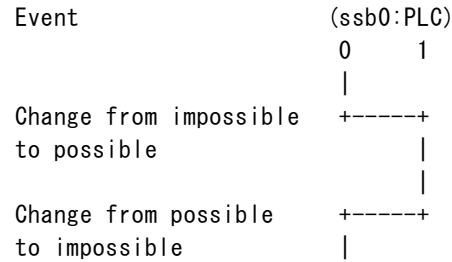
+ 2 : SourceID (0 - 0xFFFF)
+ 3 : TransacionID (0 - 0xFFFF)
+ 4 - 19 : SECS Message name (S1F2_H, S6F11_E00 etc.)
Name specified in the SECS message definition file

6. Word device that stores SECS Message information when sending SECS Message to PLC (20W)
..... PLCMSDEV1
(msd1:PC)
Configuration similar to 5.
7. Word device that stores each item value that composes SECS Message
..... Definition for each Message
Word device that stores value of each message item that configures SECS Message from PLC
to upper host or SECS Message from upper host to PLC. Define an area for storing values
for each message item in the message definition file specified in SML format.

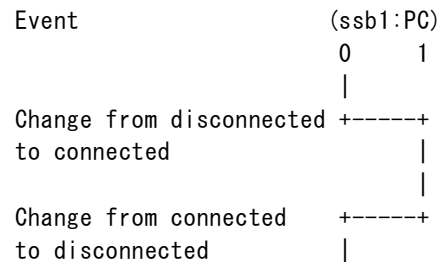
(b) Timing chart

The vertical direction is time series. (bss0:PLC) etc. are PLC device types shown in (a).

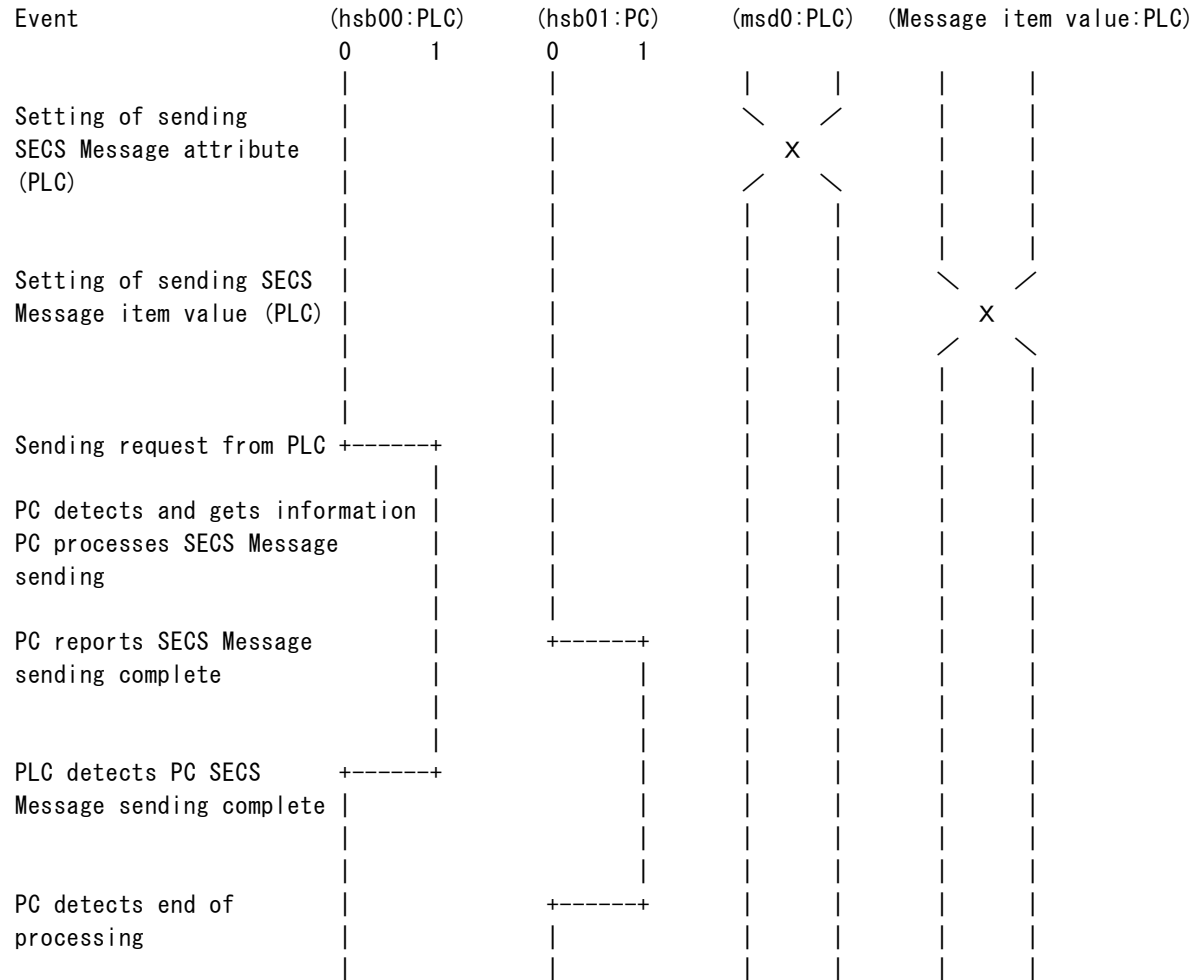
1. PLC SECS communication processing status

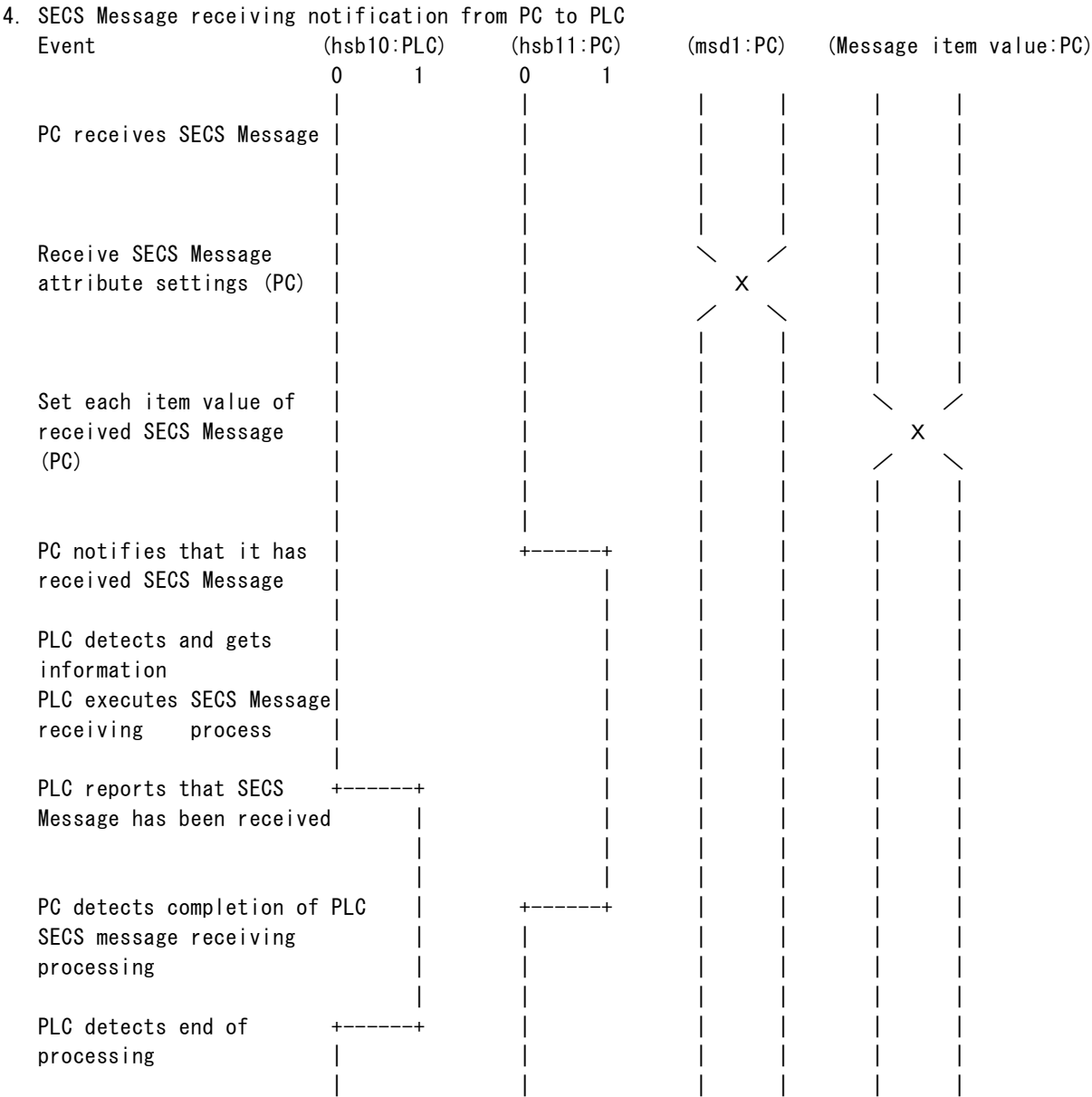


2. PC SECS connection status



3. PLC to PC SECS Message sending processing





(Note 1) This program does not support SECS Open Transaction. For sending and receiving of SECS message, it is necessary to generate next transaction after completion of transaction that combines secondary message response to one primary message.

(c) About pseudo PLC device

(i) Shared memory allocation as a pseudo PLC device

Use the shared memory in Windows PC as the memory of pseudo PLC device.

- + Object name of shared memory is name specified in tdISPLC.ini [PLC] PLCHOST.
- + The size of shared memory is byte size specified by tdISPLC.ini [PLC] PLCPORT.

+ Structure of shared memory

```
typedef struct{
    unsigned char  ssb [ 2];      // Status      bit  (Use only the 0th bit)
    unsigned char  dmy0[ 2];
    unsigned char  hsb [ 2][ 2]; // Handshake bit  (Use only the 0th bit)
    unsigned short msd [ 2][32]; // SECS Basic data
    unsigned short ssd [ 1];      // SECS Item  data Storage area
} DevPseudo;
```

(Note 1) For the contents, refer to (2) (a) (ii).

(Note 2) The size of the SECS Item data storage area is the byte size specified by tdISPLC.ini [PLC] PLCPORT minus 136 Bytes (=2+2+4+2*64). Therefore, for PLCPORT, specify a value obtained by adding 136 Bytes or more to the area size for storing the data value of SECS Item. Also note that ssd[0] can not be used (designated) as a storage area. (Refer to (2) (b) (note 3))

- + That is, in the case of C language, shared memory is allocated by the following code.

```
HANDLE      sd;
DevPseudo  *md;
// key : Shared memory Name      specified by tdISPLC.ini [PLC] PLCHOST
// sz   : Shared memory Byte size specified by tdISPLC.ini [PLC] PLCPORT
sd=CreateFileMapping(INVALID_HANDLE_VALUE, 0, PAGE_READWRITE, 0, sz, key);
md=(DevPseudo*) MapViewOfFile(sd, FILE_MAP_ALL_ACCESS, 0, 0, sz);
```

(ii) Shared memory configuration

See (i) above for the overall configuration of shared memory

The meaning of each item value is shown below.

ssb0: The notation such as PLC indicates PLC device type shown in (a).

```
ssb[ 0]      : ssb0:PLC
ssb[ 1]      : ssb1:PC
```

```
dmy0         : unused
```

```
hsb[ 0][ 0]  : hsb00:PLC
hsb[ 0][ 1]  : hsb01:PC
hsb[ 1][ 0]  : hsb10:PLC
hsb[ 1][ 1]  : hsb11:PC
```

```
msd[ 0]      : msd0:PLC  (msd[ 0][20]~[ 0][31] are unused)
msd[ 1]      : msd1:PC   (msd[ 1][20]~[ 1][31] are unused)
```

```
ssd[ 0]--    : Store SECS Message Item data value at position specified in SECS Message
                definition file.
                Refer to (2) (b). 0x0000 indicates ssd[0].
```

(2) Description of configuration file

Before starting this program, be sure to set up and prepare following two files correctly.

- + SECS/PLC communication connection parameter setting file (tdISPLCE.ini)
- + SECS Message configuration file (SML format) (Specified by tdISPLCE.ini)

(a) SECS/PLC communication connection parameter setting file

- + This file is usually called tdISPLCE.ini and should be placed in same folder as this program (tdISPLCE.exe). You can use any file by specifying startup parameters.
- + For the setting method of tdISPLCE.ini, please refer to applicable part 2.1(1) of "Programmer's Manual (TDSE.pdf)" attached to our "SECS/HSMS Communication Package (Trust Design Simple SECS Communication Library) (TDS)". You can download SECS/HSMS communication package (TDS) from our ewb site (<http://www.trust-design.co.jp/>).
- + The settings of SECS connection side are described in [DEFAULT] and [SECS] sections, and settings of PLC connection side are described in [PLC] section.
Please refer to bundled tdISPLCE.ini and use it after correcting the same file.

(i) [SECS] Section

Pay particular attention to following items when describing [DEFAULT] and [SECS] sections.
(For details, please refer to TDSE.pdf 2.1(1).)

- + SECSMODE SECS communication parameters
 - bit#0,1 Communication type (SECS-1 or HSMS-SS)
 - 4 Equipment or Host
 - 5 SECS : Master or Slave
 - 6 HSMS : Active or Passive
- + DEVMODE Device control mode
 - bit#0 Device ID check
 - 1 Processing for Secondary Messages not waiting for receiving
 - 4 Primary message transmission transaction management
 - 8-12 ... S9Fx, Reject with automatic transmission
 - 15 Processing when T6T0 occurs

If PLC itself performs transaction management when sending a primary message from PLC to host, it is necessary to set (DEVMODE&0x10)!=0 so that tdISPLC does not perform transaction management for data transmission. In this case, even if (DEVMODE&0x0100)=0, tdISPLC does not send S9F9 when T3T0 occurs. (First of all, T3T0 does not occur in tdISPLC.) Therefore, it is necessary to keep in mind that S9F9 transmission in this case also needs to be instructed to send from the PLC. Set (DEVMODE&0x10)=0 if you do not perform transaction management at the time of primary message sending from PLC to host by PLC itself not by tdISPLC. In this case, the information on T3T0 generated by tdISPLC is not transmitted to PLC, and PLC may have to perform some T0 processing independently.

<< Continue to next page >>

<< Continue from previous page >>

- + DEVID Connected device ID
- + XDEV Maximum number of connected device
- + XMSGSIZE Maximum SECS message byte length
Specify a numerical value with some margin.

- + SDEVICE SECS-1 Serial connection device name ("COM1" etc)
- + HOST HSMS-SS TCP/IP Destination host name/IP-Address
- + PORT HSMS-SS TCP/IP Connection port number
- + LINKINT HSMS-SS Link test execution interval
When performing link test, specify number of seconds for execution interval.

- + TRCDIR Trace file storage directory
In the case of relative path name, it is relative to the position where the specified .ini file exists. Refer to TDSE.pdf 2.1(3) for file name of communication trace file.
- + TRCTTYPE Communication message output format to communication trace
(Refer to TDSE.pdf 2.1(2)(c) Note)
- + TRCTOUT Communication trace output mode
- + TRCTLEVEL Communication trace output level
When outputting communication control code when connecting SECS-1, specify value of 6 or more. When outputting HSMS-SS connection trace, also specify value of 9 or 10 or more.

The following is description of message definition file that describes information related to item values that make up each SECS Message at I/F with PLC and that is also used for communication trace output. This definition is required in this program.

- + MDMSSG Specify the file (.sml) that defines the SECS message structure that performs SECS communication and PLC communication. This file is also used for communication trace output.
- + MDMXITEM Maximum number of total data items
- + MDMXMSSG Maximum number of messages to define
- + MDMXMITEM Maximum number of total data items + maximum number of items when expanding messages
- + MDMXPOOL Message definition Setting data storage area size
For these items, specify a numerical value with some margin.

(ii) [PLC] Section

Define connection conditions with PLC, information storage location for PLC communication (except for storage location of each SECS message item), etc.
(The storage location of each SECS message item on the PLC device is defined in message definition file shown in MDMSSG.)

The following is the setting for PLC communication trace file.
The settings are same as for SECS communication trace except for TRCXDATA. If not
specified in this section, the description in [DEFAULT] is valid.
For details, please refer to TDSE.pdf 2.1(1)(c) above.
PLC communication log file is created with file name such as "tp0000t.trc" in directory
specified in TRCDIR.
TRCXDATA is maximum number of PLC memories to output as PLC communication log. When not
specified and when 0 is specified, all will be output. (However, in the middle line, if
all of them are 0, it does not output.)

```
TRCDIR    = "."           // PLC Communication trace file storage location
TRCTTYPE  = 0x0008        // PLC Communication trace output type
TRCTOUT   = 0x0542        // PLC Communication trace output mode
TRCTLEVEL = 3             // PLC Communication trace output level
                        // When 3 or more is specified, acquisition value and setting value
                        // are output.
                        // When 5 or more is specified, ON/OFF of Handshake bit is output.
TRCTATTR  = 0x810f        // PLC Communication trace output attribute
TRCTSIZE  = 5000000        // PLC Communication trace file byte size
TRCXDATA  = 0             // PLC Communication trace output maximum number of memories
```

The following settings are for PLC connection conditions, etc.
This setting is unique to this program and must be specified in the [PLC] section.

```
PLCINTER  = "5000, 100"   // PLC Communication interval (ms)
                        // "Connection request interval, memory monitoring interval"
PLCTO     = "5000, 1000"  // PLC communication timeout time (ms)
                        // "When requesting connection, when monitoring memory"

PLCMODE    = "0x0000"     // PLC Memory byte order conversion instruction etc.
                        //      8 7   3210
                        // +---+---+---+---+ (0:Don't convert 1:Convert)
                        //      | |   |||+-- BIN, LOG, I1, U1
                        //      | |   ||+--- I2, I4, I8, U2, U4, U8
                        //      | |   |+--- F4, F8
                        //      | |   +---- A, J, K
                        //      | |
                        //      | +----- Message Name
                        //      +----- Handling of BIN, LOG, I1, U1
                        //              0: 2Bytes stored in PLC 1W
                        //              1: 1Byte stored in PLC 1W
                        // bit#0~7: Instructs Byte Order conversion when storing PLC Word
                        //              Device for each SECS Message item and Message Name (stored
                        //              in wms0+4, wms1+4).
                        // bit#8 : Indicates how 1-byte items are stored in PLC Word
                        //              Device.
```

<< Continue to next page >>

<< Continue from previous page >>

The following items are settings for each PLC device to be used.

```
PLCTYPE   = "15"           // Connection PLC type
                               // = 0: MELSEC (3E or 4E)
                               //  2: OMRON  (FINS)
                               //  4: KEYENCE (TCP/IP)
                               // 15: Pseudo PLC (Uses shared memory in PC)

PLCOPT0    = "0"           // PLC Connection option
                               // In case of PLCTYPE= 0 (MELSEC)
                               // = 0: 3E  =1: 4E
                               // In case of PLCTYPE= 2 (OMRON)
                               // = My node address
                               // In case of PLCTYPE= 4 (KEYENCE)
                               // = 0: KV-LE21V
                               // In case of PLCTYPE=15 (Pseudo PLC)
                               // = 0: Fixed

#PLCHOST    = "192.168.1.1" // PLC Connection host IP address
PLCHOST     = "tdISPLC00"   // In case of PLCTYPE=15, Identification name of shared memory

#PLCPORT     = "9600"        // PLC Connection TCP port number
PLCPORT      = "0x4200"     // In case of PLCTYPE=15, Shared memory byte size
```

<< Continue to next page >>

<< Continue from previous page >>

```
# Following is a description of the memory allocation used to communicate with
# the PLC.
#
# Specify memory area to set the basic items below to exchange SECS messages
# with PLC.
# The storage location of each item value included in each SECS message on PLC
# memory is specified in message definition file (specified in SML format).
#
# PLCSSDEV0 : Indicates bit device area in which PLC sets PLC operation status.
# PLCSSDEV1 : Indicates bit device area in which PC sets SECS operation status.
# Specify with "Device code, address".
#
# PLCHSDEV0 : Bit device for handshake when sending SECS message from PLC
# PLCHSDEV1 : Bit device for handshake when sending SECS message from SECS
# Specify with "Device code, PLC request address, PC response address".
#
# PLCMSDEV0 : Word device that stores SECS Message information from PLC
# PLCMSDEV1 : Word device that stores SECS Message information from SECS
# Specify with "Device code, address".
# (Note) The SECS Message information uses 20W and has the following
# configuration.
# Specified address + 0 : SF-Code and W-Bit
#                      FE      8 7      0
#                      +-----+
#                      | +-----+ +-----+
#                      | |           +----- F-Code (0 - 127)
#                      | +----- S-Code (0 - 63)
#                      +----- W-Bit
#
# + 1 : DeviceID (0 - 0x7FFF)
#      FE      0
#      +-----+
#      | +-----+
#      | +----- DeviceID
#      +----- Reserved (0)
#
# (Note) If the specified value from the PLC is 0,
# assign the specified value to the DEVID of
# this file on the PC.
#
# + 2 : SourceID (0 - 0xFFFF)
# + 3 : TransactionID (0 - 0xFFFF)
# + 4 - 19: Message name (S1F2_H, S6F11_E00 etc..)
# Name specified in message definition file
```

<< Continue to next page >>

<< Continue from previous page >>

```
# In case of PLCTYPE=0 (MELSEC) -----
# Bit device code :      0x90 : Internal relay
#                        0x92 : Latch relay
#                        0x94 : Edge relay
#                        0x9c : Input relay
#                        0x9d : Output relay
#                        0xa0 : Link relay
# Word device code :      0xa8 : Data register
#                        0xb4 : Link register
PLCSSDEV0 = "0xa0, 0x1002"           // PLC Status
PLCSSDEV1 = "0xa0, 0x1012"           // SECS Status
PLCHSDEV0 = "0xa0, 0x1000, 0x1010"    // PLC --> PC Handshake
PLCHSDEV1 = "0xa0, 0x1001, 0x1011"    // PLC <-- PC Handshake
PLCMSDEV0 = "0xa8, 0x1008"           // PLC --> PC Data
PLCMSDEV1 = "0xa8, 0x2008"           // PLC <-- PC Data
```

```
# In case of PLCTYPE=2 (OMRON) -----
# Bit device code :      0x02 : Data memory
#                        0x20 : Extended data memory Bank #0
#                        0x21 : Extended data memory Bank #1
#                        :      :
#                        0x2c : Extended data memory Bank #C
#                        0x30 : Channel I/O
#                        0x31 : Internal auxiliary relay
#                        0x32 : Holding relay
#                        0x33 : Special auxiliary relay
# Word device code :      0x82 : Data memory
#                        0xa0 : Extended data memory Bank #0
#                        0xa1 : Extended data memory Bank #1
#                        :      :
#                        0xac : Extended data memory Bank #C
# (Note) The bit device specification is specified in the
# "word address: bit address" format.
PLCSSDEV0 = "0x02, 0x1000:2"         // PLC Status
PLCSSDEV1 = "0x02, 0x1001:2"         // SECS Status
PLCHSDEV0 = "0x02, 0x1000:0, 0x1001:0" // PLC --> PC Handshake
PLCHSDEV1 = "0x02, 0x1000:1, 0x1001:1" // PLC <-- PC Handshake
PLCMSDEV0 = "0x82, 0x1008"           // PLC --> PC Data
PLCMSDEV1 = "0x82, 0x2008"           // PLC <-- PC Data
```

<< Continue to next page >>

<< Continue from previous page >>

```
# In case of PLCTYPE=4 (KEYENCE) -----
# Bit device code :      R   : Relay
#                        B   : Link relay
#                        MR  : Internal auxiliary relay
#                        LR  : Latch relay
#                        CR  : Control relay
#                        VB  : Work relay
# Word device code :     DM  : Data memory
#                        EM  : Extended data memory
#                        CM  : Control memory
#                        WM  : Work memory
PLCSSDEV0 = "B,0x1002"           // PLC Status
PLCSSDEV1 = "B,0x1012"           // SECS Status
PLCHSDEV0 = "B,0x1000,0x1010"    // PLC --> PC Handshake
PLCHSDEV1 = "B,0x1001,0x1011"    // PLC <-- PC Handshake
PLCMSDEV0 = "DM,0x1008"          // PLC --> PC Data
PLCMSDEV1 = "DM,0x2008"          // PLC <-- PC Data
```

```
# In case of PLCTYPE=15 (Pseudo PLC) -----
# Fixed to below (Not changeable)
PLCSSDEV0 = "0x00,0x0000"        // PLC Status
PLCSSDEV1 = "0x00,0x0001"        // SECS Status
PLCHSDEV0 = "0x01,0x0000,0x0001" // PLC --> PC Handshake
PLCHSDEV1 = "0x01,0x0002,0x0003" // PLC <-- PC Handshake
PLCMSDEV0 = "0x02,0x0000"        // PLC --> PC Data
PLCMSDEV1 = "0x02,0x0020"        // PLC <-- PC Data
```


(b) SECS Message configuration file (SML format)

Specify the storage location on PLC memory of each item value included in each SECS message in the message definition file (specified in SML format). The message definition file is indicated by MDMSSG in SECS/PLC communication connection parameter setting file (tdISPLCE.ini).

It is shown as a line comment specified by “//” on definition line of the SECS message item that PLC needs to use or set. By defining following items at the beginning of line comment, specify PLC memory area to store and hold item values in “1.(b) 3., 4.” Setting each item value”.

“DeviceCode, Address1, Address2, Number”

- + DeviceCode : Word device code for each PLC type shown in the description of PLCMSDEV0 and PLCMSDEV1 in (a)
- + Address1 : Address to use when PLC-->SECS(PC)
- + Address2 : Address to use when SECS(PC)-->PLC
- + Number : Number of word devices used (ie, number of words)

(Note 1) Leave one or more white spaces after SECS message item definition and write “//” to indicate that it is a comment. After that, separate the above 4 items with ‘,’ and specify them without blanks. Also, be sure to enclose the above four items in double quotes (“”). Comments other than PLC address specification are not described in SECS message definition line.

(Note 2) It is not necessary to perform this specification for items not required by the PLC at the time of receiving of PLC and items not required to be set by PLC at the time of sending from PLC. If this specification is not made, that item will be ignored when receiving PLC. When sending from PLC, the item value becomes item default value specified in the message definition file.

(Note 3) The address value =0 can not be used.
If you specify =0, no processing is performed for that item.

(Note 4) The device code is fixed at 0x03 when PLCTYPE=15 (Pseudo PLC). Other values can not be used.

(Note 5) The number of list items can not be undefined. The number of list items (n in L[n]) always specifies a fixed number.

(Example 1) For PLCTYPE = 0 (MELSEC)

```

S1F2_E      E
<L[2]      // The number of LIST items must be fixed.
           // Write comments on a line other than the message item
           // specification line.
<A[6]MDLN "EQUIP1">      // "0xa8, 0x1020, 0x0000, 3"
<A[6]SOFTREV "01.000">    // "0xa8, 0x1023, 0x0000, 3"
>

S2F49_H      W
<L[3]
<A[6]RCMD "LOAD">      // "0xa8, 0x0000, 0x21a0, 3"
<B[1]STID 0x01>      // "0xa8, 0x0000, 0x21a3, 1"
<A[16]LOTID "">      // "0xa8, 0x0000, 0x21a4, 8"
>

```

3. Operation explanation

(1) Start-up

Start tdISPLCE.exe (or a shortcut to tdISConvE.exe) in the installed folder by double-clicking etc.

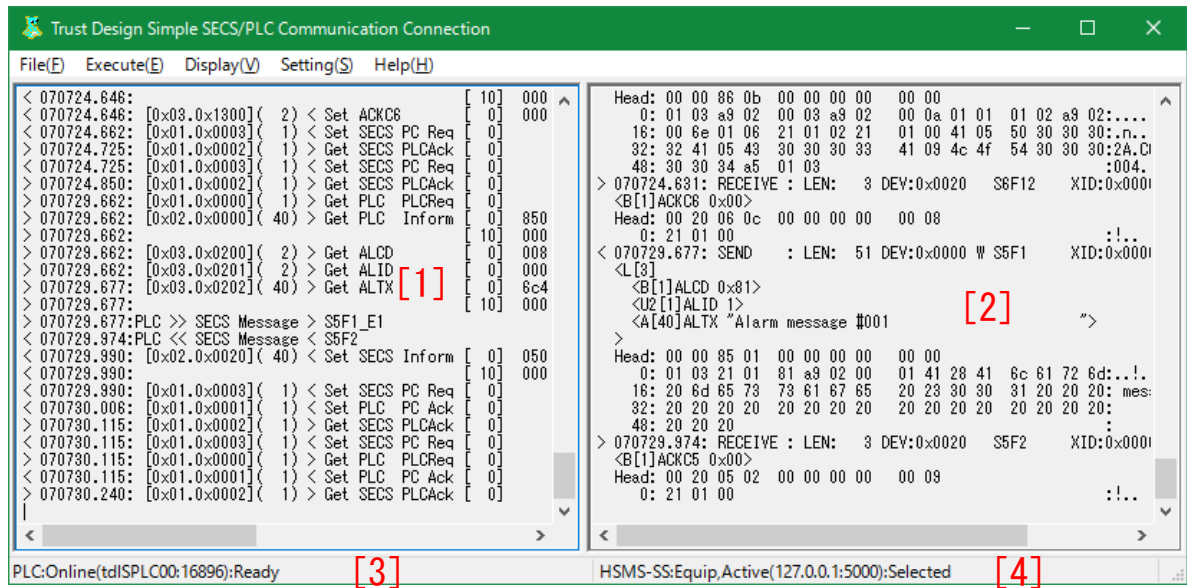
This program uses "MS Gothic" as the font used.

Please execute in the environment where the font can be used.

(Reference) The following can be specified as startup options.

+i ini_file	: Setting file name Default : tdISPLCE.ini
+S style_file	: Name of file to save operation style (position, size, specified parameter) Default : tdISPLCEWin.ini (File name excluding the extension of the setting file to be used, with "Win" appended to the file name) (Note 1) Operation style save file is stored in the same folder as the configuration file.
+s SECS_Section	: Section name of SECS setting Default : SECS
+p PLC_Section	: Section name of PLC setting Default : PLC

(2) Screen operation explanation



- [1] : PLC connection side communication trace display
Trace display of communication contents with PLC.
- [2] : SECS connection side communication trace display
Trace display of communication contents with SECS.
- [3] : PLC connection side status
Displays the current connection status of PLC I/F.
- [4] : SECS connection side status
Displays the current connection status of SECS I/F.

(a) Menu

(a-1) [File]

- + End of application Exit td\SPICE.

(a-2) [Execute]

- + Start Start SECS/PLC communication connection processing.
- + Stop Stops SECS/PLC communication connection processing.

(Note 1) If connection with other party is not established, it may take some time for [Stop] process. In such a case, if you want to end the program immediately, please end by [X] at right end of title bar.

(a-3) [Display]

- + Clear communication trace ... Clear the communication trace window ([1], [2]).
- + View last line of communication trace
 - ... Make the scroll bar of communication trace window ([1], [2]) display last line.
 - (Note) It is used when the last line does not easily appear in normal scroll bar operation, for example, in the state of scrolling at high speed.
 - When the last line is displayed, it scrolls automatically.
- SECS List format Send/receive SECS communication trace window [2] Displays SECS messages in specified List format.
- SECS Hexa format Send/receive SECS communication trace window [2] Displays SECS message in hexadecimal format.
- PLC Detail Communication SECS message information is displayed in hexadecimal format and ASCII characters in the PLC communication trace window [1].
- PLC Handshake Communication with PLC I/O contents of Handshake bit are displayed on PLC communication trace window [1].
- PLC Redundant display When PLC detail display and PLC Handshake display are ON, communication data display on PLC communication trace window [1] is displayed for all communication including communication in idle state regarding each display contents.
If this specification is not made, it will be displayed only when there is a change in state regarding I/O of bit device.

(Note 1) The indication of each display status does not affect the output to communication trace file for both SECS communication and PLC communication. The output to communication trace file is determined by setting file TRCTTYPE, TRCTLEVEL, etc.

(a-4) [Setting]

- Automatic start It is set to start processing automatically after program startup. This setting will be effective when the program starts next time.
- Save window position and size ... When program ends, position, size, etc. are saved, and the state of window is restored to the same state at next start.

(a-5) [Help]

- Version information of td\SPICE .. Display program version information.

(b) PLC communication trace display

It displays traces of events related to entire program and PLC communication processing contents.

(Note 1) If [PLC Detail] and [PLC Handshake] are not checked in [Display], only sending and receiving SECS message name is displayed. When checked, the PLC communication contents shown in (a-3) are displayed.

(Note 2) The output to trace file will display PLC communication data contents in hexadecimal if TRCLEVEL value is 3 or more, regardless of specification in [Display]. If it is 5 or more, it also displays I/O of bit device, including handshake with PLC. However, communication contents are not output in idle state. Please refer to 2. (2) (b) (ii) TRCLEVEL.

(c) SECS communication trace display

Displays a trace of SECS communication messages sent and received.

If [SECS List format] and [SECS Hexa format] are both unchecked in [Display], only information on send/receive header will be displayed.

(Note 1) The form of list format display of SECS communication messages is determined by following parameters in [SECS] section of tdISPLCE.ini file.

- + TRCTTYPE Communication message output format to communication trace
 - bit#2 Item data display format
 - =0: Display each item on one line only and omit the back if it does not fit on one line.
 - 1: Each item is displayed on multiple lines, 20 items for numerical items and 100 bytes for character string items.
 - bit#4,5,6 .. List output format (usually use as 2)
 - =0: TDS Format
 - 2: SML Format
 - bit#7 Data item name display (If you do not use message definition, set it to 0)
 - bit#8,9 Message definition file format
 - In this program, be sure to use SML format message definition file with =0)
 - =0: SML Format

(Note 2) The form of hexadecimal display of SECS communication messages is determined by following parameters in [SECS] section of tdISPLCE.ini file.

- + TRCTTYPE Communication message output format to communication trace
 - bit#3 Hexadecimal display format
 - =0: Display 16 Bytes per line
 - 1: Display 20 Bytes per line

(Note 3) Communication control code send/receive when SECS-1 is connected is not displayed in communication trace display window.

Output to communication trace file (if TRCTLEVEL>6 in tdISPLEC.ini).

(Note 4) The LinkTest message when connecting HSMS-SS is not displayed in communication trace display window.

Output to communication trace file (if TRCTLEVEL>9 in tdISPLCE.ini).

(Note 5) The output to communication trace file depends on TRCDIR, TRCTTYPE, TRCTOUT, TRCTLEVEL, and TRCTSIZE of tdISPLCE.ini. For details, please refer to TDSE.pdf 2.1(1).