Serial Line Communication Monitor

SECS-1 communication monitor

RS232C <--> TCP/IP Repeater

(tdlSerMon)

(Trust Design Simple Serial Line Communication Monitor)

Instruction Manual

Version 19.040 : 2019.04.15
Version 19.070 : 2019.07.05
Version 19.100 : 2019.10.25
Version 21.062 : 2021.08.17

Trust Design Limited Liability Company

Chino City Nagano Prefecture Japan

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

Table of Contents

1. Introduction

This program runs on Windows PC to monitor communication using RS232C Serial Line.

This program has the following features.

+ Monitoring is performed by a relay method using this AP as a repeater.
  (It is not a capture method of RS232C line using Y cable.)
  For details, please refer to “2. Connection method”.

+ This program can be made to function as a RS232C <--> TCP/IP conversion relay by specifying a COM port for serial communication on one port and a TCP/IP port for the other port.

+ Communication data can be displayed on the screen in various formats in real time, and can be stored in same format in log file if specified.

+ When the communication data is a SECS message, communication message can be analyzed in SECS-2 format, and message display in SML format is performed.
  Therefore, it can also be used as a SECS(SECS-1,SECS-2) monitor.

+ Multiple APs can be operated in one PC, and multiple communication lines can be monitored.

+ After setting and checking the operation, if automatic startup setting of this program is performed to Windows, it is possible to operate automatically by turning on the power of the Windows PC that operates this program thereafter.

[Note 1]

This program relays the monitored serial connection in relay mode using two COM ports implemented in Windows PC. Therefore, when running this program on a PC other than monitored device, it is necessary to reconnect the cable. In addition, when operating in the monitored device, it is necessary to connect this program and the monitored application using a virtual serial port driver, etc. In this case, it may be necessary to change the settings related to serial port of application running in the monitored device that operates this program.

This program processes input data one byte at a time and outputs monitor information to the screen and log file. Because processing time is required for that, communication speed between monitored applications may be delayed. Therefore, the use of this program in a severe environment during communication processing time may affect the overall system performance.

To reduce the execution overhead of this program, reduce the display of input bytes on the screen (especially communication Base display screen) and the output to log file. In other words, reduce the output items. It is also recommended to reduce height of display window as much as possible.

+ 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.

[Note 2]

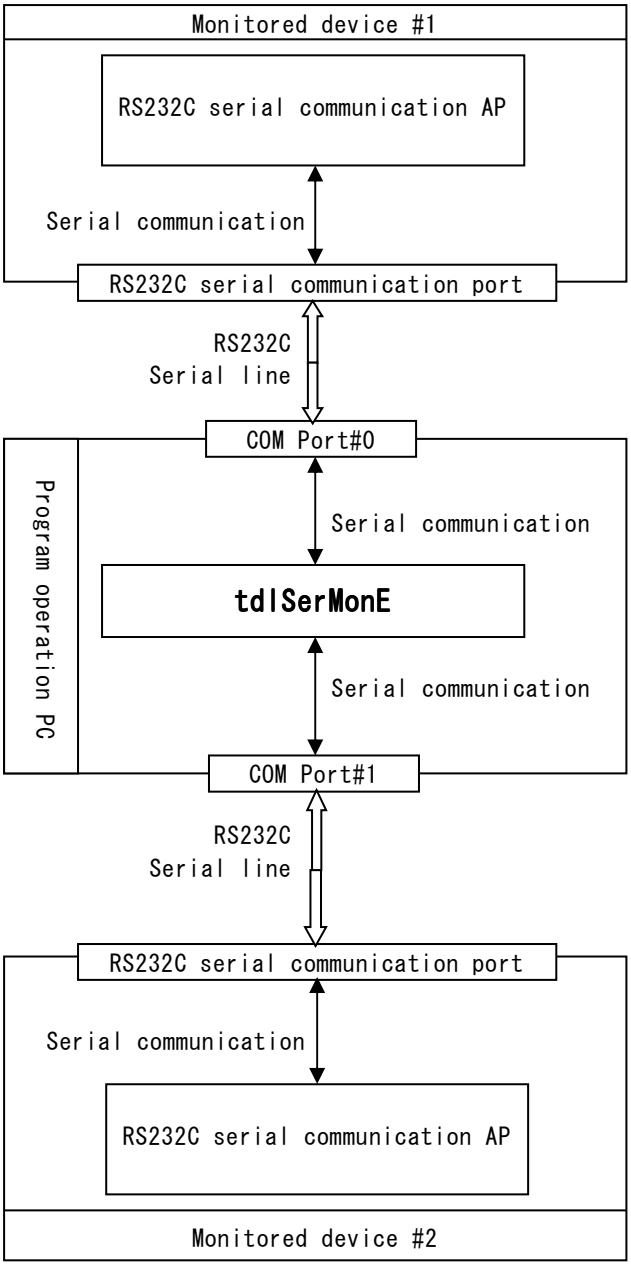
This package uses following ports of UDP/IP for license management. Also use following class D address as UDP/Multicast address. Please set not to block theese by firewall etc. of your computer.

- 36261/udp
- 239.254.200.60

However, including Internet connection environment, it can be used even when network connection can not be made and NIC does not exist, there are no functional restrictions on usage as compared with internet connection environment.

2. Connection method

(1) When running this program on a PC other than the machine on which the application for RS232C serial communication to be monitored operates



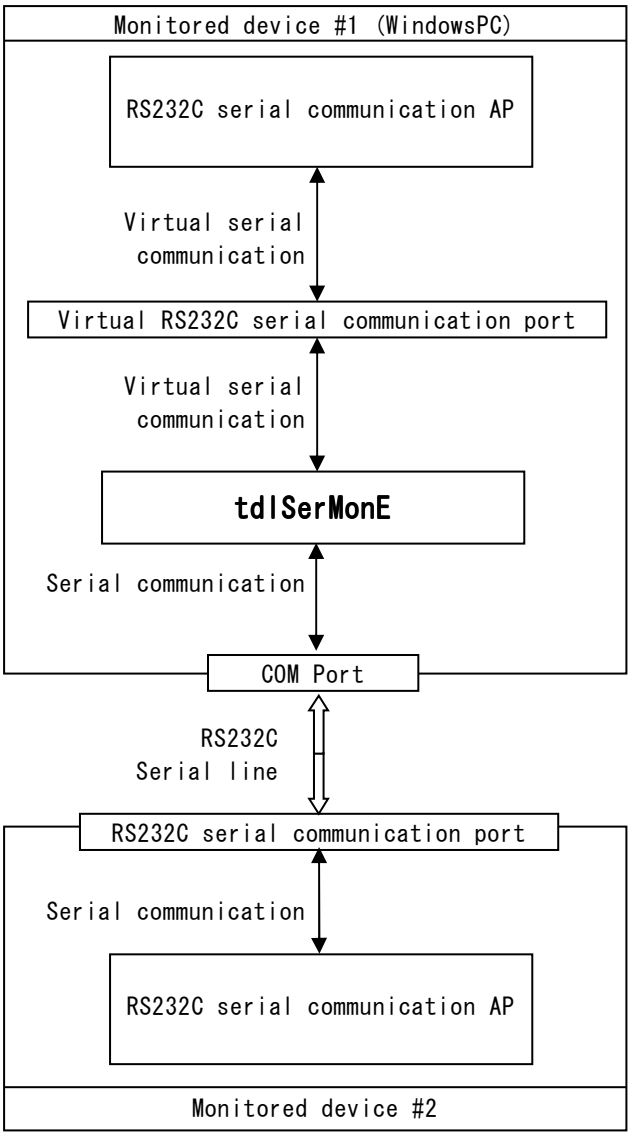
It is not necessary to change the settings of the monitored application that runs in monitored device #1.

Under normal circumstances, you can use the cable connected with monitored equipment #1 and #2 as it is.

It is necessary to prepare a cable equivalent to the cable (usually RS232C cross cable) connected to the monitored equipment #1 and #2.

It is not necessary to change the settings of the monitored application that runs in monitored device #2.

(2) When running this program on any PC of a machine that runs RS232C serial communication target for monitoring



It is necessary to change the settings of the monitored application that runs in monitored device #1.
Change the name of connection destination COM port to the name of COM port for virtual RS232C serial communication.

Use the virtual RS232C serial communication driver to cross connect COM ports such as "com0com"(*1), and connect this program with the monitored AP.

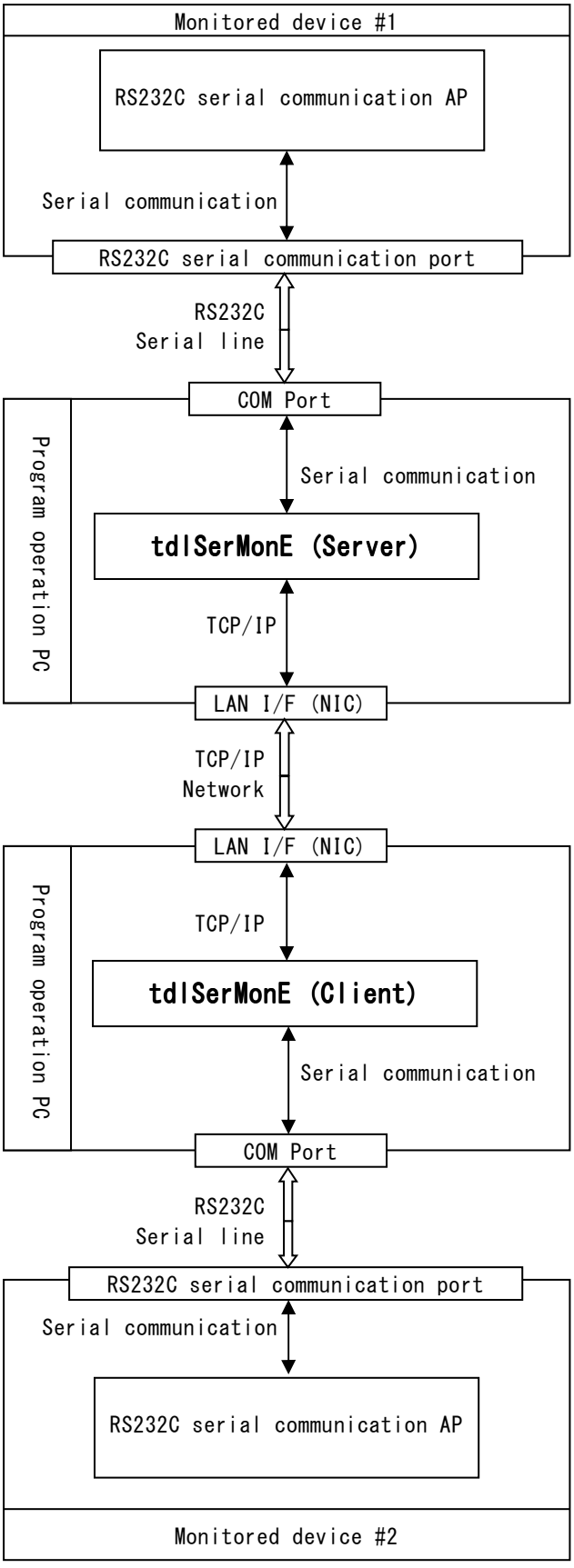
The COM port used by this program is virtual RS232C serial communication port on one side and COM port on the other side.

You can use the cable that connected monitor target device # 1 and # 2 as it is

It is not necessary to change the settings of the monitored application that runs in monitored device # 2.

(*1) com0com
An open source, virtual serial port driver that runs in Windows kernel mode under the GPL (GNU General Public License). Please refer to "http://com0com.sourceforge.net" for details.

(3) When this program is operated as a RS232C <--> TCP/IP relay and operated on a PC different from the machine on which the application to perform RS232C serial communication to be monitored oerates



It is not necessary to change the settings of the monitored application that runs in monitored device #1.

Under normal circumstances, you can use the cable connected with monitored equipment #1 and #2 as it is.

As COM ports used by this program, one is a serial communication port installed on the PC, and the other is a TCP/IP port number to be used.

Usual LAN cable or wireless LAN I/F can be used.

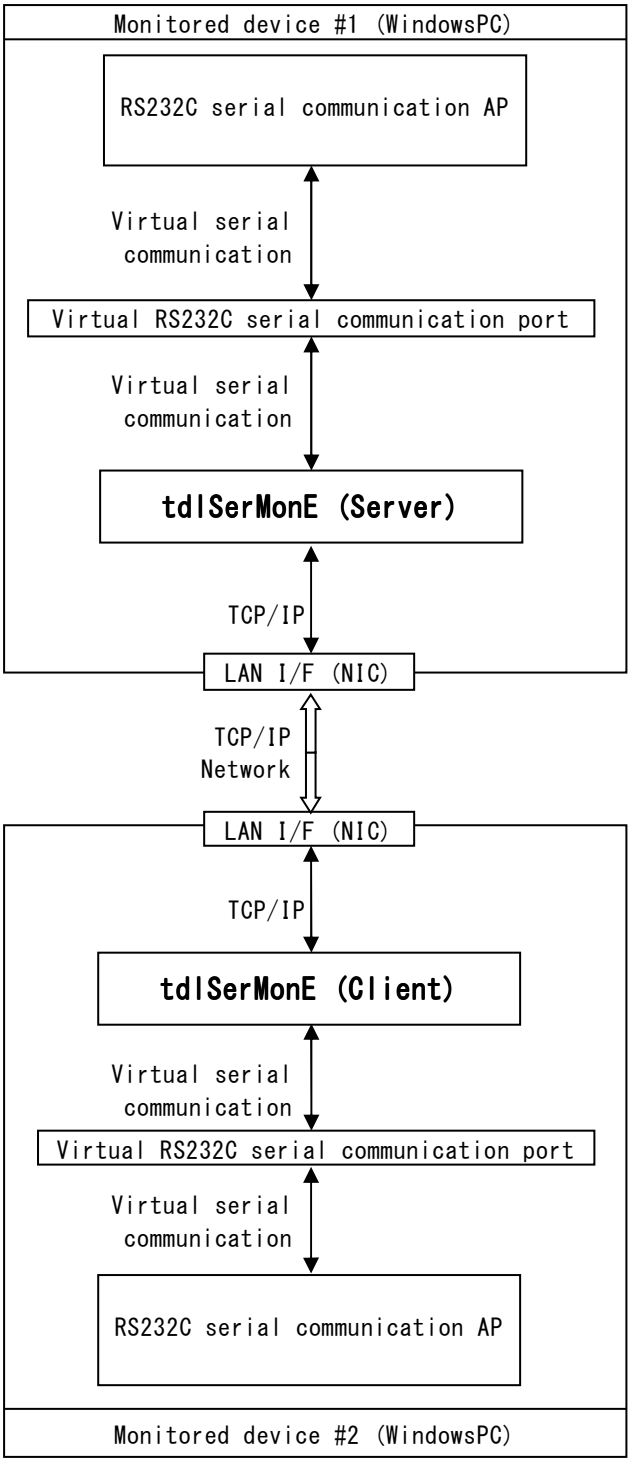
One of the COM ports used by this program specifies serial communication port implemented on the PC, and the other specifies the IP-Address and TCP/IP port number of the destination host.

A cable equivalent to the cable (usually RS232C cross cable) connected to monitored equipment #1 and #2 needs to be prepared.

It is not necessary to change the settings of monitored application that runs in monitored device #2.

(Note 1) The tdlSerMonE, which functions as TCP/IP Server, does not allow connections from different Clients to TCP/IP Port while processing with the opposite Client.

(4) When this program is operated as RS232C <--> TCP/IP relay, and it is operated on a PC of a machine on which an application that performs RS232C serial communication to be monitored operates



It is necessary to change settings of the monitored application that runs in monitored device #1.
Change the name of connection destination COM port to the name of COM port for virtual RS232C serial communication.

Use the virtual RS232C serial communication driver to cross connect COM ports such as "com0com"(*1), and connect this program with the monitored AP.

The COM port used by this program specifies one virtual RS232C serial communication port and the other TCP/IP port number to use.

Usual LAN cable or wireless LAN I/F can be used.

As with the tdlSerMon setting on monitored device #1, one specifies the virtual RS232C serial communication port, and the other specifies IP-Address and TCP/IP port number of connection destination host.

Use the virtual RS232C serial communication driver to cross connect COM ports such as "com0com"(*1), and connect this program with the monitored AP.

As with AP on Monitored Device #1, you need to change setting of AP operating conditions.
Change the name of connection destination COM port to the name of COM port for virtual RS232C serial communication.

(*1) com0com
An open source, virtual serial port driver that runs in Windows kernel mode under the GPL (GNU General Public License). Please refer to "http://com0com.sourceforge.net" for details.

(Note 1) The tdlSerMon, which functions as TCP/IP Server, does not allow connections from different Clients to TCP/IP Port while processing with the opposite Client.

3. Operating condition setting file

Before starting this program, be sure to set and prepare the "Operation condition setting file" shown below correctly.

(1) File attribute

+ Operating condition setting file : tdlSerMonE.ini

(Note 1) Normally tdlSerMonE.ini should be placed in the same folder as this program (tdlSerMonE.exe). You can use any file by specifying startup parameters.

(2) Overall structure

```
[Section]                       // Section name
                                // (Note 2) For the section that describes operating
                                //    conditions of this program, use [SERMON] when not
                                //    specified in the start parameter.
Token = Parameter               // Comment

# Comment line
```

(Note 3) Section names [DEFAULT] and [SECSCOMMON] are scanned regardless of section names. Therefore, it is possible to specify common specifications in the [DEFAULT] or [SECSCOMMON] section and specify only individual specifications in each section.
Each section is read in the following order regardless of the position where the section is defined, and the values read later are prioritized.

Reading order	Priority	Section name
1.	3.	[DEFAULT]
2.	2.	[SECSCOMMON]
3.	1.	[Specified section]

If you do not specify section name in the startup parameter, use [SERMON] as [Specified section] name.

(Note 4) Specify section names and token names with alphanumeric characters, and are not case sensitive.

(Note 5) When an integer value is given as a parameter, if the first character is x or 0x, it can be specified in hexadecimal. If the parameter string contains a space or comma (,), enclose the entire parameter with ".

(Note 6) Bits and items described as <reserved> in this section, and undescribed bits must be =0.

(Note 7) This library ignores descriptions of the sections in this file other than [DEFAULT], [SECSCOMMON], and specified section. Even in the specified section, tokens other than the tokens described below are ignored. Therefore, it is possible to describe user-specific parameters in this file.

(3) Example

```
[SERMON]

SDEVICE0   = "COM1"          // Serial Port #0
SDEVICE1   = "COM2"          // Serial Port #1
#SDEVICE0  = ":5000"             // In case of RS232C <--> TCP/IP Repeater server
#SDEVICE1  = "192.168.2.1:5000"  // In case of RS232C <--> TCP/IP Repeater client

SSPEED     =    9600         // BPS
SCSIZE     =       8         // Character Size                                  (5/6/7/8)
SPARITY    =       0         // Parity                                          (0/1/2)
SSTOP      =       1         // Stop bit                                        (1/2)
SDTR       =       0         // DTR/DSR Control                                 (0/1/2/3)
SRTS       =       0         // RTS/CTS Control                                 (0/1/2/3/4)

STO        =    1000         // Serial Port Timeout                             (ms)
CWAIT      =    8000         // Wait time after Port Close processing           (ms)
XINTER     =       1         // Port data arrival confirmation interval         (ms)

XCOL       =     100         // Number of columns in communication Base display
XROW       =   10000         // Maximum number of lines in data display area
DOT        = " .."           // Displayed string of receiver                    (3 characters)
UNPRINT    = " "             // Display character of non-displayable character  (1 character)
DOUT       =  0xb111         // Presence and format of screen display

XMSGSIZE   = 1000000         // Presence and format of screen display
XITMSIZE   = 1000000         // Maximum SECS item byte length

TRCDIR     = "."             // Communication log storage directory
TRCSIZE    = 5000000         // Communication log maximum size                  (Byte)
TRCOUT     =  0x0340         // Communication log output mode

LICENSENAME= "Reiwa Research Institute Ltd. Suwa Office"
LICENSECODE= "Reiwa Research Institute Ltd. Suwa Office"
LICENSESER = "1"
LICENSEDATE= "20190101"
LICENSEKEY = "ABCDE-FGHIJ-KLMNO-PQRST-UVWXY-Z1234"
```

(4) Token and Parameter

```
SDEVICE0   = "COM1"          // Name of Serial Communication Port #0 to use
SDEVICE1   = "COM2"          // Name of Serial Communication Port #1 to use
                             //
                             // (Note 1) When using this program as a RS232C <--> TCP/IP relay
                             //    device, please set DEVICE0 or DEVICE1 as below.
                             //    + When acting as a TCP/IP Server
                             //      ":99999"      (First character is ':')
                             //    + When acting as a TCP/IP Client
                             //      "XXXXXX:99999"
                             //
                             //      XXXXXX : Destination IP-Address or host name
                             //      99999  : TCP/IP port number to use


SSPEED     = 9600            // Serial connection bit rate
                             // = 300,600,1200,2400,4800,9600,19200,38400 etc.
SCSIZE     = 8               // Serial connection character bit size
                             // = 5 / 6 / 7 / 8
                             // (In case of SECS-1, setting is usually impossible except =8)
SPARITY    = 0               // Serial connection parity format
                             // = 0 : None
                             //   1 : ODD  Parity
                             //   2 : EVEN Parity
                             // (In case of SECS-1, setting is usually impossible except =0)
SSTOP      = 1               // Serial connection number of stop bits
                             // = 1 / 2
                             // (In case of SECS-1, setting is usually impossible except =1)
SDTR       = 0               // DTR/DSR Control
                             // = 0: Do not set anything (keep current state)
                             //   1: Control disable.
                             //   2: Control enable. Maintain as ON when OPEN.
                             //   3: Control enable. Enable Handshake.
SRTS       = 0               // RTS/CTS Control
                             // = 0: Do not set anything (keep current state)
                             //   1: Control disable.
                             //   2: Control enable. Maintain as ON when OPEN.
                             //   3: Control enable. Enable Handshake.
                             //      The RTS line is turned off when input data exceeds three
                             //      quarters of buffer, and RTS line is turned on when input
                             //      data falls less than half.

STO        =    1000         // Serial Port Timeout                          (ms)
CWAIT      =    8000         // Wait time after Port Close processing        (ms)
XINTER     =       1         // Port data arrival confirmation interval      (ms)
                             // The above three parameters control the operating state of this
                             // program, and usually the above values are not changed.
```

<< Continue to next page >>

<< Continue from previous page >>

```
XCOL       =     100   // Number of columns in communication Base display
                       // Number of columns for displaying received data in the basic log
                       // display window (and basic log file) on the left side of screen.
XROW       =   10000   // Maximum number of lines in display area of basic log display window
                       // on the left side of screen and the SECS log display window on the
                       // right side.

DOT        = " .."     // Displayed character string of receiver in communication Base display
                       //                                                    (3 characters)
           // In Communication Base display described later, characters to be displayed on the
           // side other than the display side of received data.
           // The following characters are used according to the number of characters used for
           // 1 byte of received data in Communication Base display.
           // 1 character  :  Third character of string specified in DOT
           // 2 characters : Second and third character of string specified in DOT
           // 3 characters : First, Second and third character of string specified in DOT

UNPRINT    = " "       // Character to be displayed at the ASCII character display position in
                       // Communication Base display when display data value is <0x20 or >0x7e.

DOUT       =  0xb111   // Screen display, log output presence and format
           //  F DC  A98   54   10
           // +----+----+----+----+
           // bit# 0    : Communication Base display execution                    (0:No 1:Yes)
           //      1    : Communication SECS display execution                    (0:No 1:Yes)
           // bit# 4    : Communication Base log output execution                 (0:No 1:Yes)
           //      5    : Communication SECS log output execution                 (0:No 1:Yes)
           //
           // bit# 8-10 : Communication Base display, log display format
           //        =0 : Hexa notation only           (2 vertical letters) (1 column  2 lines)
           //         1 : ASCII letter and Hexa notation                    (1 column  3 lines)
           //         2 : Mnemonic of control characters and Hexa notation  (1 column  3 lines)
           //         3 : Mnemonic of control characters and ASCII letter   (1 column  3 lines)
           //         4 : Mnemonic of control characters and ASCII or Hexa  (1 column  3 lines)
           //         5 : Mnemonic of control characters and ASCII and Hexa (1 column  3 lines)
           //         6 : Only displayable ASCII characters (0x20-0x7e)     (1 column  1 line)
           //         7 : Hexa nonation only       (2 horizontally letters) (2 columns 1 line)
           //
           // bit#12-15 : Communication SECS display, log display format
           //     12    : SML format list output                                  (0:No 1:Yes)
           //     13    : Hexadecimal dump format output                          (0:No 1:Yes)
           //             Output of SECS-2 message unit (ie after receiving END block) in both
           //             SML format and Hexadecimal dump format
           //     15    : Communication control code etc. (following data) output  (0:No 1:Yes)
           //             - ENQ,EOT,ACK,NAK
           //             - 1 byte indicating the byte length of SECS-1 data packet
           //             - SECS-1 data packet (each transmission block)
           //
           // (Note 2) For details on the Communication Base display format, refer to the
           //    explanation in "4. (2) [] Communication Base Display Format".
           //
           // (Note 3) This setting can be changed in the [Display] menu while this program is
           //   running. However, to output a log, it is necessary to set bit #4 and 5 to ON(1)
           //  in this parameter.  In that case, if "Log file output" is uncheked in [Display],
           //  log output is suspended.
```

  << Continue to next page >>

<< Continue from previous page >>

```
XMSGSIZE   = 1000000         // Maximum byte length of SECS message to be analyzed
XITMSIZE   = 1000000         // Maximum byte length of items constituting the SECS message to be
                             // analyzed
                             //
                             // (Note 4) If you analyze communication data as a SECS message,
                             //   and display and log it in SECS-2 SML format, you need to
                             //   specify a value larger than the maximum byte length of the
                             //   SECS message that may be received.


TRCDIR     = "."             // Directory for storing communication Base log and communication
                             // SECS log
TRCSIZE    = 5000000         // Maximum byte size of Communication Base log and Communication
                             // SECS log
TRCOUT     =  0x0340         // Output mode of Communication Base log and Communication SECS log
                             //    D     8  654
                             // +----+----+----+----+
                             //    +-+---+  |++-- Log file daily switch specificationt
                             //      |      |   =0: No
                             //      |      |    1: Yes (Create date directory and store it in)
                             //      |      |    2: Yes (Date is attached to head of file name)
                             //      |      |    3: Yes (Date is attached to end of file name)
                             //      |      |
                             //      |      +-- File switching specification is over specified
                             //      |          size (TRCSIZE)
                             //      |          =0: No
                             //      |           1: Yes
                             //      |
                             //      +-- Number of trace files to keep
                             //          =0      : Leave all files.
                             //           1 - 63 : Delete files before the specified number from
                             //                    the time of switching files.
                             //
                             // (Note 5) Communication Base log and communication SECS log files
                             //   are created in the directory indicated by TRCDIR with the
                             //   following names.
                             //   + Communication Base log : tdlSerMonB.log
                             //   + Communication SECS log : tdlSerMonS.log
                             //
                             //   If (TRCOUT&0x0030)!=0, the date (YYMMDD) is given in the
                             //   specified format.
                             //   If (TRCOUT&0x0040)!=0, the file name is as follows.
                             //   + Communication Base log : tdlSerMonB.9999.log
                             //   + Communication SECS log : tdlSerMonS.9999.log
                             //                                         +-- Sequence number
```

  << Continue from next page >>

<< Continue from previous page >>

```
LICENSENAME= "XXXXXXXX"      // Name of individual or group who received icense for operation
LICENSECODE= "XXXXXXXX"      // Code name of the individual or organization who received license
LICENSESER = "999"           // Serial# in the individual or group who received the license
LICENSEDATE= "YYYYMMDD"      // Acquisition date of execution license       (YYYYMMDD format)
LICENSEKEY = "XXXX-XXXX-..." // Execution license key code
          //
          // [Warning]
          // If the license key code is not set, if the wrong license key code is set, if the
          // expired operation check license key code is set, the operation is stopped after
          // operation for a fixed time.  Also, even if a valid operation confirmation license
          // key code (Default License) is set, the operation is stopped after several hours
          // of operation.
          // In the case of a license that has been officially licensed for execution but is
          // not a license key for operation confirmation (Default License), different PCs
          // must be written with different license key code information.
          // When running multiple programs in the same PC, it is not necessary to set different
          // license key codes separately for those operating programs.
```

4. Operation explanation

(1) Start-up

Start tdlSCerMonE.exe (or a shortcut to tdlSerMonE.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.

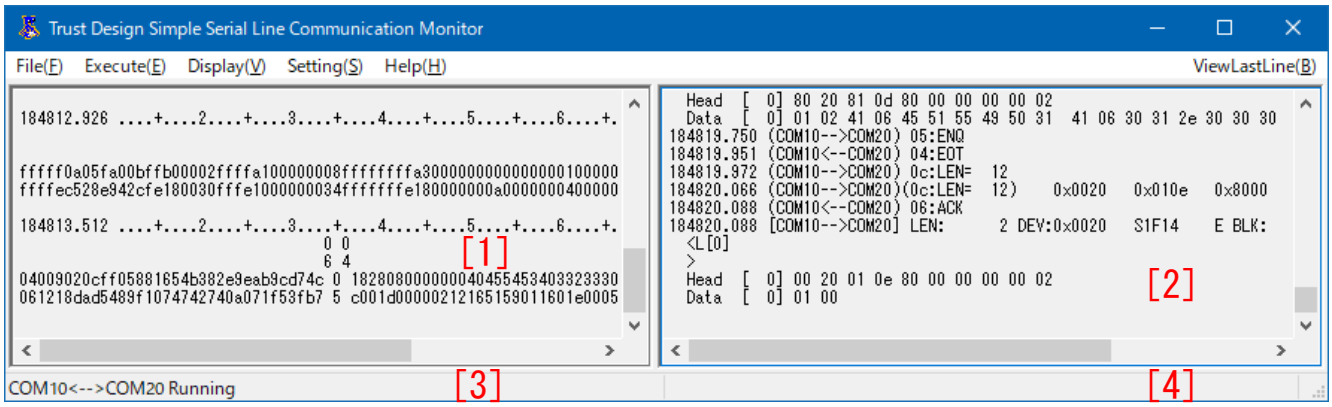
(Reference) The following can be specified as startup options.

+i ini_file : Setting file name
Default : tdlSerMonE.ini

+s SECS_Section : Section name of SECS-1 setting
Default : SERMON

+S style_file : Name of file to save operation style (position, size, specified parameter)
Default : tdlSerMonEWin.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.

(2) Screen operation explanation



[1] : Communication Base display
Displays each byte value of received serial communication.
The display format is described in detail in "[] Communication Base Display Format" below.

[2] : Communication SECS display
Displays the SECS-2 message as a result of analyzing received serial communication data as SECS-2 message transmission/reception.
The display format is described in detail in "[] Communication SECS Display Format" below.

[3] : Execution status
Displays the current program execution status.

[4] : TCP/IP Status
When this program is used as RS232C <--> TCP/IP converter, TCP/IP connection status is displayed.

[] Communication Base display format

0. Hexadecimal display only (2 characters vertically) (1 column, 2 lines)

```
      Received time of 0th byte (hhmmss.msec)
      |
132020.838 ....+....2....+....3....+....4 —  Header line .. 1st byte received time and ruler
0 082808000000a  0                        }   Data received from SDEVICE0 (2 rows/1 set)
5 a001101000114  4                        }   Display the Hexadecimal notation in vertical 2 letters
 0              00 c0200800000004044454240 }  Data received from SDEVICE1 (2 rows/1 set)
 4              65 600120100011f16cd12b016 }
                   |
                   The 20th Byte data (0x20 = ' ') received in this writing line
```

(Note 1) Each line on the screen does not show the output time, but for the output to log file, the display date, time (YYMMDD.hhmmss.m seconds) is displayed at beginning of each line.

(Note 2) The time given to beginning of each line recorded in log file (YYMMDD.hhmmss.msec) is the time when line was recorded in log file. Therefore, normally, it is the time immediately after the time when last byte data recorded on the line was received. The time when the first baye of the line is received is the time recorded in the header line.

1. ASCII character and Hexadecimal display (1 column, 3 lines)

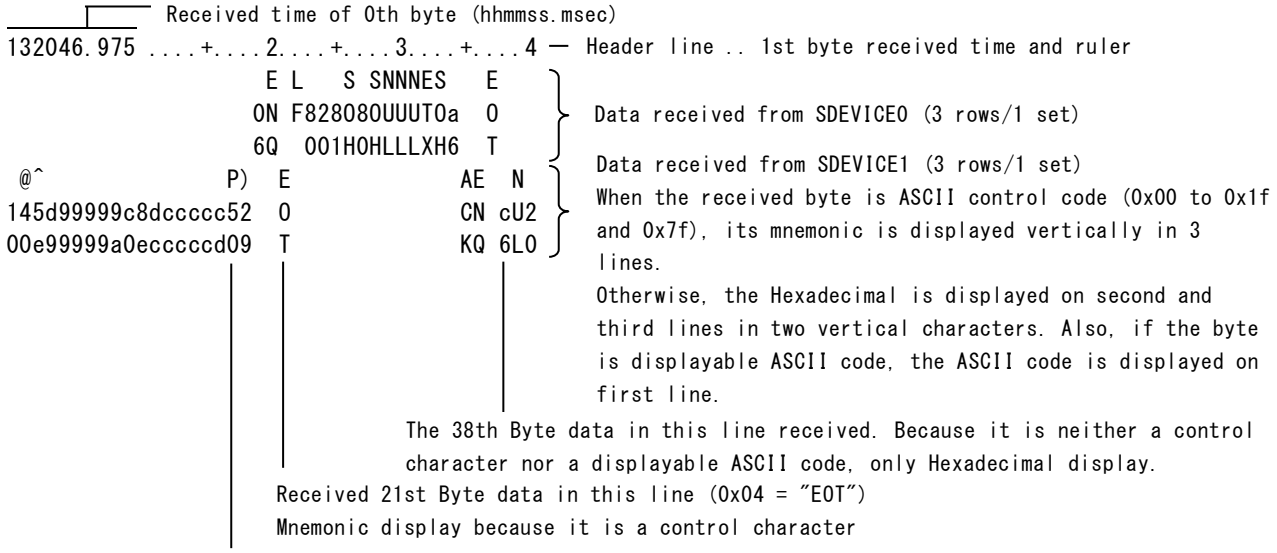
```
      Received time of 0th byte (hhmmss.msec)
      |
132045.338 ....+....2....+....3....+....4 —  Header line .. 1st byte received time and ruler
                                          }   Data received from SDEVICE0 (3 rows/1 set)
0 082808000000a  0                        }   Displays ASCII code that can be displayed on first line,
5 a001101000215  4                        }   and Hexadecimal on second and third line
                                A LMARK A }
 0              00 c0200800000004044454240 }  Data received from SDEVICE1 (3 rows/1
 4              65 600120100021f16cd12b016 }
                                  |
                                  The 31st Byte data (0x41 = 'A') in this description line received
```

2. Control character mnemonic and Hexadecimal display (1 column, 3 lines)
   This is almost same as "5. Mnemonics of control characters and ASCII and Hexadecimal display", but ASCII code that can display byte data does not display on first line.

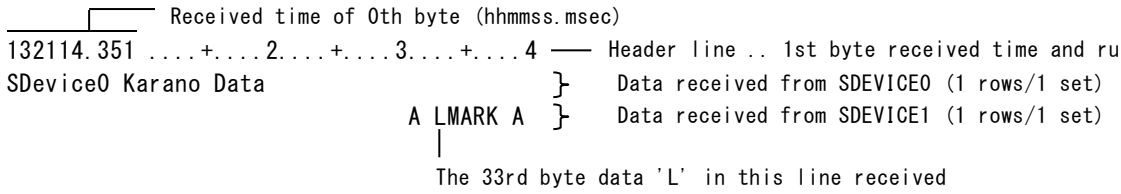
3. Mnemonic of control characters and ASCII display (1 column, 3 lines)
   This is almost same as "5. Mnemonics of control characters and ASCII and Hexadecimal display", does not display Hexadecimal on lines 2 and 3.

4. Mnemonic of control character and ASCII or Hexadecimal display (1 column, 3 lines)
   This is almost same as "5. Mnemonics of control characters and ASCII and Hexadecimal display", If the display target byte is a displayable ASCII character, it will be displayed on first line, and if it can not be displayed, the Hexadecimal display will be on lines 2 and 3.

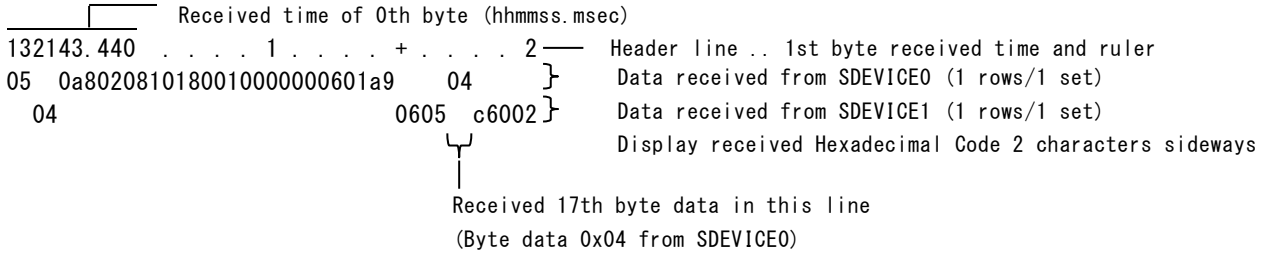
5. Mnemonics of control characters and ASCII and Hexadecimal display (1 column, 3 lines)



6. Only displayable ASCII characters (0x20-0x7e) (1 column, 3 line)



7.Hexa display only (2 letters horizontally) (2 columns, 3 lines)



[] Communication SECS display format

```
145057.257 (COM11-->COM21) 05:ENQ
145057.537 (COM11<--COM21) 04:EOT
```

Mnemonic display when Receive Byte is a control code
If the received Byte is out of SECS-1 protocol, and if the Baye is an ASCII Code that can be displayed, the ASCII Code is displayed.

Display Received Byte in Hexadecimal

Indicates the direction of the data.

Received time (hhmmss.mili seconds)
For log file output, give the received time in form "YYMMDD.hhmmss.msec" including the date at the beginning of the line.

```
145057.547 (COM11-->COM21) fe:LEN= 254
```

If Receive Byte is "Transmission block length (Byte count)" on SECS-1 protocol, the value is displayed in decimal. (In this example, 254 bytes.)

```
145057.777 (COM11-->COM21)(fe:LEN= 254)    0x8020   0x860b   0x0001          0x0000.0001:0x36e6
```

If previously received Byte data of "transmission block length" is received, its SECS block header information and checksum value are displayed

SECS Header: Device ID
SECS Header: Message ID
SECS header: block number
SECS Header: Source ID
SECS header: transaction ID
Checksum

```
145057.787 (COM11<--COM21) 06:ACK
145057.797 (COM11-->COM21) 05:ENQ
145057.807 (COM11<--COM21) 04:EOT
145057.817 (COM11-->COM21) fe:LEN= 254
145057.987 (COM11-->COM21)(fe:LEN= 254)    0x8020   0x860b   0x0002          0x0000.0001:0x3cf1
145057.997 (COM11<--COM21) 06:ACK
145058.007 (COM11-->COM21) 05:ENQ
145058.017 (COM11<--COM21) 04:EOT
145058.027 (COM11-->COM21) 9b:LEN= 155
145058.127 (COM11-->COM21)(9b:LEN= 155)    0x8020   0x860b   0x8003          0x0000.0001:0x23fd
145058.137 (COM11<--COM21) 06:ACK
```

Block number 'E' bit is ON

If the 'E' bit in the block number of received SECS header is ON, the header and message body of received SECS message will be displayed in SML format as if reception of a series of SECS messages has been completed.

```
145058.137+[COM11-->COM21] LEN:    633 DEV:0x8020 W S6F11    E BLK:    3 SBT:0x0000.0001
  <L[3]
    <U2[1] 0>
    <U2[1] 1103>
    <L[2]
      <L[2]
        <U2[1] 38>
        <L[3]
```

System byte
Number of blocks
Indicates E-Bit ON receaved
SF-Code
Displayed when W-Bit ON
Device ID (including R-Bit)
SECS message byte count

<< Continue to next page >>

<< Continue from previous page >>

```
          <B[1] 0x01>
          <A[20] "LOTID (001)         ">
          <U1[1] 5>
        >
      >
      <L[2]
        <U2[1] 46>
        <L[5]
          <A[20] "waferid (0001 01)   ">
          <A[20] "waferid (0001-02)   ">
          <A[20] "waferid (0001-03)   ">
          <A[20] "waferid (0001-04)   ">
          <A[20] "waferid (0001-05)   ">
        >
      >
    >
  >
```

```
145058.647 (COM11<--COM21) 05:ENQ
145058.657 (COM11-->COM21) 04:EOT
145058.667 (COM11<--COM21) 0d:LEN=  13
145058.697 (COM11<--COM21)(0d:LEN=  13)    0x0020   0x060c   0x8001          0x0000.0001:0x01d5
145058.707 (COM11-->COM21) 06:ACK
145058.707 [COM11<--COM21] LEN:      3 DEV:0x0020   S6F12    E BLK:    1 SBT:0x0000.0001
145058.707   <B[1] 0xff>
```

The following is an example of dump output in Hexadecimal format.

```
151603.822-  Head  [  0] 00 22 8a 05 80 01 00 00 05 4b                           :.".......K
151603.822-  Data  [  0] 01 03 21 01 50 01 02 01 03 41  19 4d 65 73 73 61 67 65 20 70:..!.P....A.Message p
151603.822-  Data  [ 20] 61 72 74 32 20 3a 20 30 30 30  36 37 2e 30 30 34 21 01 01 a9:art2 : 00067.004!...
151603.832-  Data  [ 80] 03 00 04 41 0a 41 44 44 53 54  52 44 45 53 55              :...A.ADDSTRDESU
```

(Note 1) Except for the first line and the last line of a series of Hexadecimal displays (ie, in the middle line), if all the data in that line is 0, that line is not output. [999] indicates the position of the first byte of the line in its entirety, so determine whether it is undisplayed data with the continuity of [999].

(3) Menu

(a) [File]

+ End of application .. Exit tdlSerMonE.

(b) [Execute]

+ Start ..... Start serial communication line monitor.
+ Stop ..... Stop serial communication line monitor.

(Note 1) If any failure occurs in I/O of the serial communication line, etc., it may take some time for the [Finish] process.
In such a case, if you quit the program immediately, please exit with [X] at the right end of the title bar.

(c) [Display]

+ Clear communication trace
Clear the Communication Base display window [1] and the communication SECS display window [2].
+ View last line of communication trace
Set the scroll bar of Communication Base display Window[1] and Communication SECS display Window[2] to the state where the last line is displayed.
(Used when final line does not easily appear in normal scroll bar operation, etc., with auto scrolling at high speed, etc.)

+ Basic Display output
The Byte data received from specified serial port is displayed in specified format on Communication Base display window[1].
+ Basic Log file output
The same output as the contents displayed in Communication Base display window[1] is applied to Communication Base log file.
+ Basic Display format
Select the display format of communication byte data to be displayed on Communication Base display window[1] from the options below.

0. Only Hexa (2lines)		
	Hexadecimal display only (2 characters vertically)	(1 column, 2 lines)
1. ASCII+Hexa		
	ASCII character and Hexadecimal display	(1 column, 3 lines)
2. ControlChar+Hexa		
	Control character mnemonic and Hexadecimal display	(1 column, 3 lines)
3. ControlChar+ASCII		
	Mnemonic of control characters and ASCII display	(1 column, 3 lines)
4. ControlChar+ASCII or Hexa		
	Mnemonic of control character and ASCII or Hexadecimal display	(1 column, 3 lines)
5. ControlChar+ASCII+Hexa		
	Mnemonics of control characters and ASCII and Hexadecimal display	(1 column, 3 lines)
6. Only Displayable ASCII Code		
	Only displayable ASCII characters (0x20-0x7e)	(1 column, 1 line)
7. Only Hexa (2Columns)		
	Hexa display only (2 letters horizontally)	(2 columns, 1 line)

(Note 2) For details on Communication Base display output format, refer to the explanation in “4. (2) [] Communication Base Display Format”.

<< Continue to next page >>

<< Continue from previous page >>

+ SECS Diaplay output
  Communication SECS display window[2] displays the analysis result assuming that the byte data SECS-2 message received from specified serial port.
+ SECS Log file output
  The same output as the contents displayed in Communication SECS display window[2] will be output to Communication SECS log file.
+ SECS Display format
  Communication specify the content to be displayed on SECS display window[2].
  - List style : Display in SML format after receiving last block
  - List Hexa Mix : A dump display of corresponding SECS data value is added to right side of display in SML format.
  - Hexa style : Hexadecimal dump output after receiving the last block
  - Control code : Outputs control code related to send/receive protocol of SECS message block

(d) [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 the program ends, the position, size, etc. at the end of this program are saved, and the state of the window is restored to the same state at the next start.

(e) [Help]

+ Version information of tdlSConvE
  Display version information of this program.

(f) [ViewLastLine]
+ Positioned on the last line of communication trace display.
  (Same as (c) [View last line of communication trace])