

シリアルライン通信モニタ
SECS-1 通信モニタ
RS232C <--> TCP/IP 中継器

(tdlSerMon)

(Trust Design Simple Serial Line Communication Monitor)

取扱説明書

Version 19.040 : 2019.04.15

Version 19.070 : 2019.07.05

合 同 会 社 ト ラ ス ト デ ザ イ ン

長野県 茅野市 中大塩 3-4-3

Tel:0266-75-2279

E-mail:info@trust-design.co.jp

URL:http://www.trust-design.co.jp

Please refer to 'tdlSerMonE.pdf' if you want to refer to the contents of this file in English.

Please refer to 'tdlSerMonE.pdf' if you want to refer to the contents of this file in English.

目

次

1. はじめに	1
2. 接続方法	3
3. 動作条件設定ファイル	7
4. 操作説明	13

1. はじめに

本プログラムは、RS232C シリアル・ラインを利用した通信を監視するための Windows PC で動作するアプリケーション・プログラムです。

本プログラムは以下の特徴を持ちます。

- ・ モニタリングは、本 A P を中継器として使用するリレー方式で行います。
(Y ケーブルを用いた、RS232C ラインのキャプチャ方式ではありません。)
詳しくは「2. 接続方法」をご参照ください。
- ・ 本プログラムが使用する一方のポートにシリアル通信を行う COM ポート、もう一方のポートに TCP/IP ポートを指定することにより、本プログラムを RS232C <--> TCP/IP の中継器として機能させることができます。
- ・ 通信データをリアルタイムに様々な形式で画面上に表示し、指定によりログ・ファイルに同様の形式で格納することができます。
- ・ 通信データが SECS メッセージの場合、通信メッセージの SECS-2 形式の解析が可能であり、SML 形式でのメッセージ表示が可能です。
従って、SECS (SECS-1, SECS-2) モニターとしても使用可能です。
- ・ 1 台の P C 内で複数の本 A P を稼働させることができ、複数の通信ラインのモニターが可能です。
- ・ 設定、動作確認後、Windows に対して本 A P の自動起動設定を行えば、以降は本 A P が稼働する Windows PC の電源 ON にて、自動運転する事が可能です。

[注意 1]

本プログラムは、WindowsPC に実装した 2 つの COM Port を使用して、モニター対象のシリアル接続をリレー方式で中継します。従って、本プログラムをモニター対象機器とは別の P C で動作させる場合は、ケーブルの再接続が必要です。また、モニター対象機器内で動作させる場合は、仮想シリアル・ポート・ドライバ等を使用して本プログラムとモニター対象アプリケーションと接続する必要があります。その場合、本プログラムを動作させるモニター対象機器内で動作するアプリケーションのシリアル・ポートに関する設定を変更する必要がある場合があります。

本プログラムは、入力データを 1 バイトずつ処理し、モニター情報を画面 及び ログ・ファイルに出力するための処理時間が必要となるため、モニター対象のアプリケーション間での通信速度には遅れが出る可能性があります。そのため、通信処理時間にシビアな環境での本プログラムの使用は、システム全体のパフォーマンスに影響を与える可能性があります。

本プログラムの実行オーバーヘッドをなるべく少なくするためには、入力バイトの画面（特に通信 Base 表示画面）への表示、ログ・ファイルへの出力を減らす（出力項目を少なくする）と共に、表示ウインドの高さをなるべく小さくすることを勧めます。

- ・ SECS/HSMS による通信システムの開発には、弊社 SECS/HSMS 通信パッケージ (Trust Design Simple SECS Communication Library) (使用ライセンス無料) を公開しております。詳しくは、弊社ホームページをご覧ください。
- ・ SECS-1 及び HSMS-SS、HSMS-GS による通信処理 (ホスト側、装置側) をシミュレートする検査アプリケーションとして、SECS/HSMS シミュレータ (暫定版) (Trust Design Simple SECS/HSMS Simulator (Preliminary Version)) (使用ライセンス無料) を公開しております。詳しくは、弊社ホームページをご覧ください。
- ・ SECS-1 と HSMS-SS、HSMS-GS 相互の通信プロトコルを変換するアプリケーションとして、SECS/HSMS プロトコル変換プログラム (Trust Design Simple SECS/HSMS Protocol Converter) (使用ライセンス無料) を公開しております。詳しくは、弊社ホームページをご覧ください。
- ・ PLC にて制御する装置を、SECS 通信 I/F で上位システムに接続するためのアプリケーションとして、SECS/PLC 通信接続プログラム (Trust Design Simple SECS/PLC Communication Connection) (使用ライセンス無料) を公開しております。詳しくは、弊社ホームページをご覧ください。
- ・ SECS (HSMS) による通信のモニターには、弊社 ネットワーク通信モニター (Trust Design Simple Network Communication Monitor) (使用ライセンス無料) を公開しております。詳しくは、弊社ホームページをご覧ください。

[注意 2]

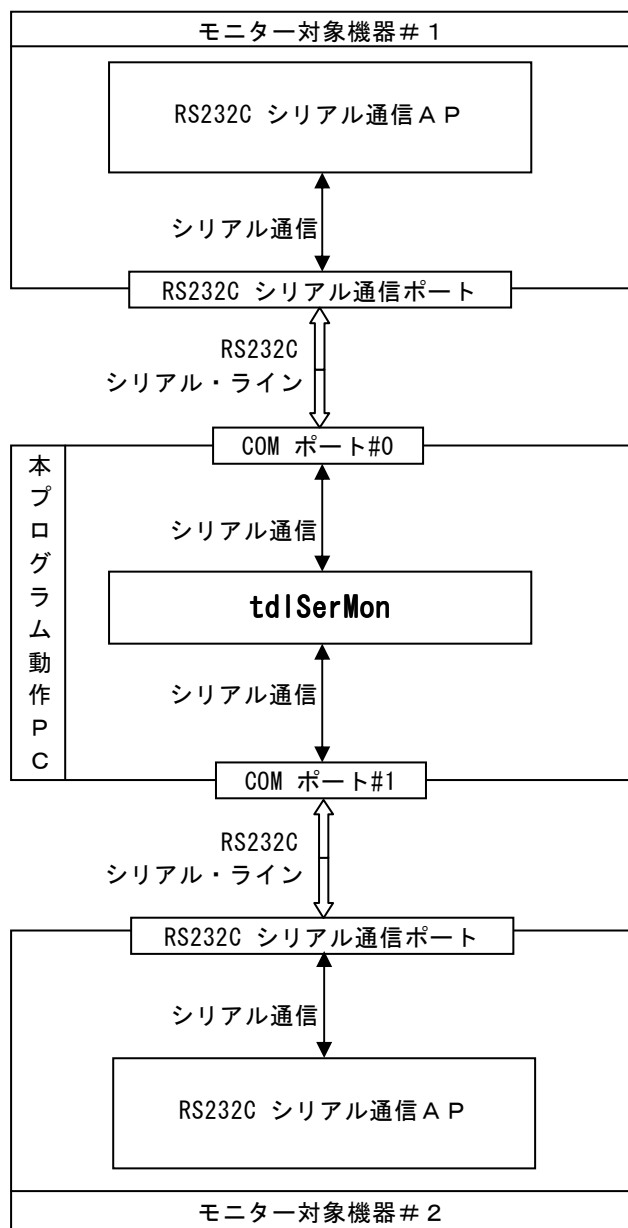
本プログラムは、ライセンス管理用として、UDP/IP の以下のポートを使用します。また、UDP/Multicast アドレスとして、以下のクラスDアドレスを使用します。ご使用になるコンピュータのファイアウォール等により、これらをブロックしないよう設定してください。

- ・ 36261/udp
- ・ 239.254.200.60

なお、インターネット接続環境を含め、ネットワーク接続ができない状態、NIC が存在しない状態でも、ご使用は可能であり、使用に関する機能上の制限等は、同環境がある場合に比して、一切ありません。

2. 接続方法

- (1) 本プログラムを、モニター対象の RS232C シリアル通信を行うアプリケーションが動作するマシンとは別の PC で動作させる場合



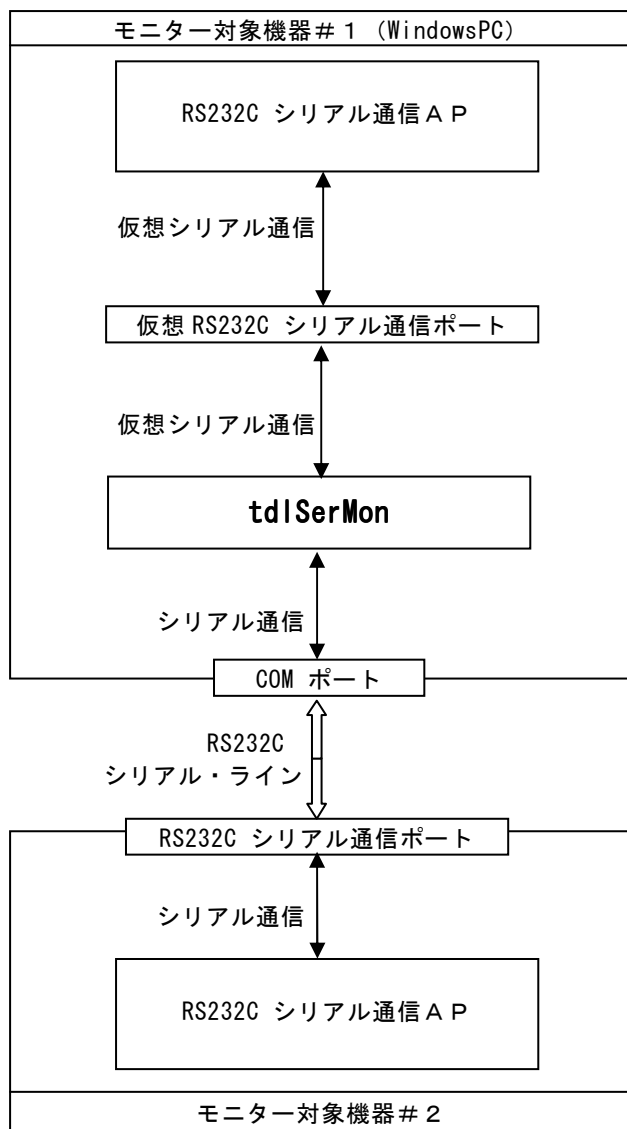
モニター対象機器 # 1 内で動作するモニター対象アプリケーションの設定を変更する必要はありません。

通常の場合、モニター対象機器 # 1、# 2 を接続していたケーブルを、そのまま使用できます。

モニター対象機器 # 1、# 2 を接続していたケーブル（通常 RS232C クロス・ケーブル）と同等のケーブルを別途用意する必要があります。

モニター対象機器 # 2 内で動作するモニター対象アプリケーションの設定を変更する必要はありません。

- (2) 本プログラムを、モニター対象の RS232C シリアル通信を行うアプリケーションが動作するマシンのどちらかの PC で動作させる場合



モニター対象機器 # 1 内で動作するモニター対象アプリケーションの設定を変更する必要があります。
接続先の COM ポートの名称を、仮想 RS232C シリアル通信用の COM ポートの名称に変更してください。

“com0com”(*1) 等の、COM ポート間をクロス接続する仮想 RS232C シリアル通信ドライバを使用して、モニター対象 A P と本プログラムを接続してください。

本プログラムが使用する COM ポートは、一方は仮想 RS232C シリアル通信ポート、もう一方は実装した COM ポートとなります。

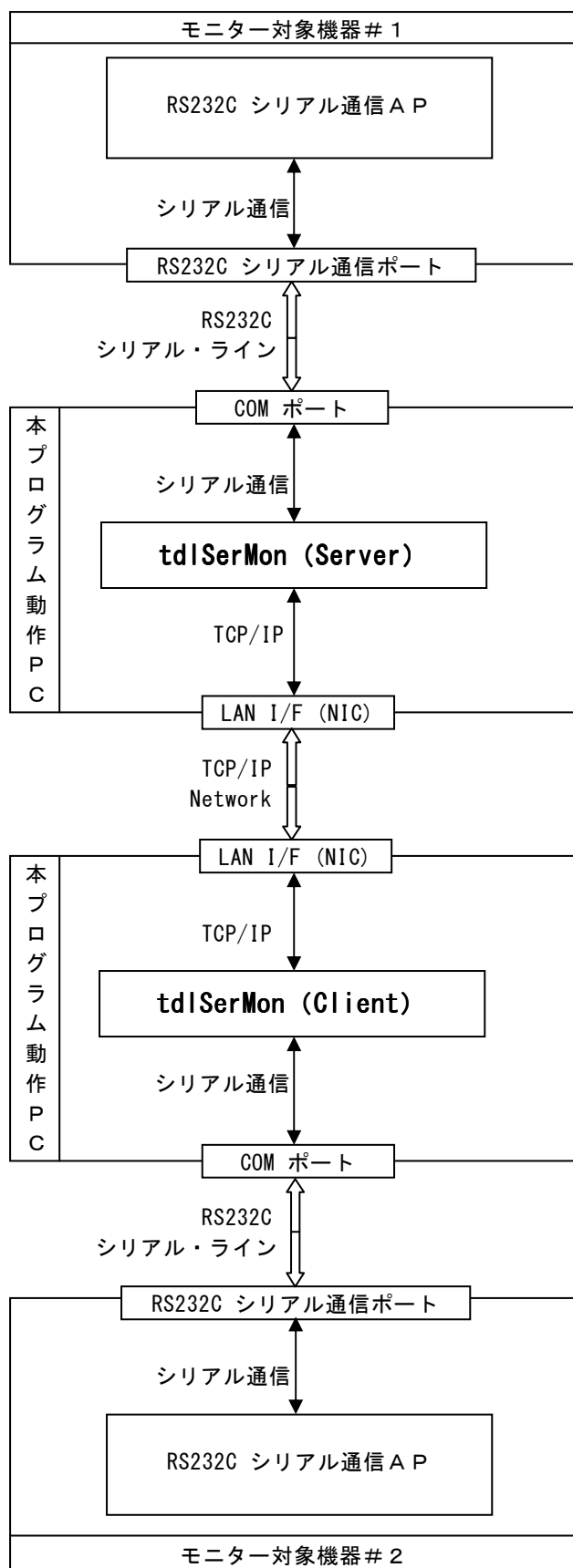
モニター対象機器 # 1、# 2 を接続していたケーブルをそのまま使用できます。

モニター対象機器 # 2 内で動作するモニター対象アプリケーションの設定を変更する必要はありません。

(*1) com0com

GPL (GNU General Public License) に基づくオープンソースの、Windows のカーネルモードで動作する仮想シリアルポート・ドライバです。詳細は “<http://com0com.sourceforge.net>” をご参照ください。

- (3) 本プログラムを、RS232C <--> TCP/IP 中継器として動作させ、モニター対象の RS232C シリアル通信を行うアプリケーションが動作するマシンとは別の P C で動作させる場合



モニター対象機器 # 1 内で動作するモニター対象アプリケーションの設定を変更する必要はありません。

通常の場合、モニター対象機器 # 1、# 2 を接続していたケーブルを、そのまま使用できます。

本プログラムが使用する COM ポートは、一方は PC に実装するシリアル通信ポート、もう一方は使用する TCP/IP ポート番号を指定します。

通常の LAN ケーブル、もしくは無線 LAN I/F を使用できます。

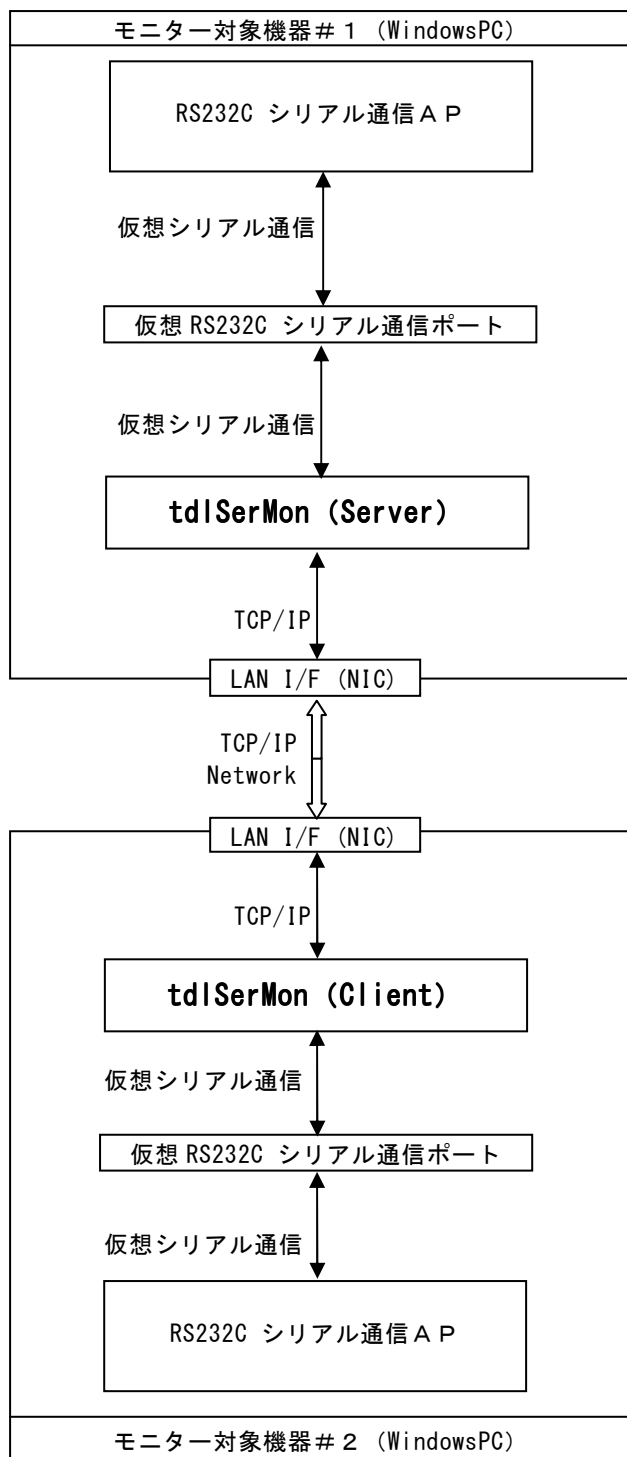
本プログラムが使用する COM ポートは、一方は PC に実装するシリアル通信ポート、もう一方は接続先ホストの IP-Address と TCP/IP ポート番号を指定します。

モニター対象機器 # 1、# 2 を接続していたケーブル（通常 RS232C クロス・ケーブル）と同等のケーブルを別途容易する必要があります。

モニター対象機器 # 2 内で動作するモニター対象アプリケーションの設定を変更する必要はありません。

(注 1) TCP/IP Server として機能する tdlSerMon は、相対する Client 側との処理中は、異なる Client からの TCP/IP Port への接続を許可しません。

- (4) 本プログラムを、RS232C <--> TCP/IP 中継器として動作させ、モニター対象の RS232C シリアル通信を行うアプリケーションが動作するマシンの PC で動作させる場合



モニター対象機器 # 1 内で動作するモニター対象アプリケーションの設定を変更する必要があります。
接続先の COM ポートの名称を、仮想 RS232C シリアル通信の COM ポートの名称に変更してください。

“com0com”(*1) 等の、COM ポート間をクロス接続する仮想 RS232C シリアル通信ドライバを使用して、モニター対象 A P と本プログラムを接続してください。

本プログラムが使用する COM ポートは、一方は仮想 RS232C シリアル通信ポート、もう一方は使用する TCP/IP ポート番号を指定します。

通常の LAN ケーブル、もしくは無線 LAN I/F を使用できます。

モニター対象機器 # 1 での tdlSerMon の設定と同様、一方は仮想 RS232C シリアル通信ポート、もう一方は接続先ホストの IP-Address と TCP/IP ポート番号を指定します。

“com0com”(*1) 等の、COM ポート間をクロス接続する仮想 RS232C シリアル通信ドライバを使用して、モニター対象 A P と本プログラムを接続してください。

モニター対象機器 # 1 での A P と同様、A P の動作条件の設定を変更する必要があります。
接続先の COM ポートの名称を、仮想 RS232C シリアル通信の COM ポートの名称に変更してください。

(*1) com0com

GPL (GNU General Public License) に基づくオープンソースの、Windows のカーネルモードで動作する仮想シリアルポート・ドライバです。詳細は “<http://com0com.sourceforge.net>” をご参照ください。

(注 1) TCP/IP Server として機能する tdlSerMon は、相対する Client 側との処理中は、異なる Client からの TCP/IP Port への接続を許可しません。

3. 動作条件設定ファイル

本プログラムを起動する前に、必ず以下に示します「動作条件設定ファイル」を正しく設定し、用意する必要があります。

(1) ファイル属性

- ・ 動作条件設定ファイル : tdlSerMon.ini

(注1) 通常 tdlSerMon.ini は、本プログラム (tdlSerMon.exe) が動作するディレクトリと同じディレクトリに配置する必要があります。起動パラメータを指定することにより、任意のファイルを使用することができます。

(2) 全体構成

```
[Section]                // セクション名称
                        // (注2) 本プログラムの動作条件を記述するセクションは、
                        // 起動パラメータで指定しない場合 [SERMON] を使用します。

Token = Parameter        // コメント

# コメント行
```

(注3) セクション名 [DEFAULT] および [SECSCOMMON] は、セクション名の指定に関わらず走査します。従って、全体に共通な指定を [DEFAULT] もしくは [SECSCOMMON] セクションに指定し、個別の指定のみを各セクションにて指定する事が可能です。

各セクションは、セクションを定義する位置には無関係に、以下の順番で読み込み、後で読み込んだ値を優先します。

読み込み順番	優先順位	セクション名
1.	3.	[DEFAULT]
2.	2.	[SECSCOMMON]
3.	1.	[指定のセクション名]

起動パラメータでセクション名を指定しない場合は、[指定のセクション]名として、[SERMON] を使用します。

(注4) セクション名、トークン名は英数字で指定し、大文字、小文字の区別はありません。

(注5) パラメータとして整数値を与える場合、先頭文字を 0x とすると 16 進数で指定することができます。また、パラメータ文字列中に、スペース、カンマ(,) を含む場合は、パラメータ全体を " で括弧することが必要です。

(注6) 本項での記述中 <reserved> と記述したビット 及び 項目、また 未記述のビットは =0 としなければなりません。

(注7) 本プログラムは、本ファイル内の記述の内、[DEFAULT]、[SECSCOMMON] および指定のセクション以外のセクションの記述、また指定セクションであっても、以下に記述するトークン以外のトークンに関しては、無視します。従って本ファイルにユーザ固有のパラメータを記述することも可能です。

(3) 設定例

[SERMON]

```

SDEVICE0 = "COM1"          // Serial Port #0
SDEVICE1 = "COM2"          // Serial Port #1
#SDEVICE0 = ":5000"         // RS232C<-->TCP/IP 中継器サーバの場合
#SDEVICE1 = "192.168.2.1:5000" // RS232C<-->TCP/IP 中継器クライアントの場合

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 制御        (0/1/2/3)
SRTS       =      0         // RTS/CTS 制御        (0/1/2/3/4)

STO        =    1000        // Serial Port Timeout      (ms)
CWAIT      =    8000        // Port Close 処理後の待ち時間 (ms)
XINTER     =      1         // Port データ到着確認する間隔 (ms)

XCOL       =     100        // 通信 Base 表示のカラム数
XROW       =   10000        // データ表示領域の最大行数
DOT        = "  ."         // 被受信側の表示文字列      (3 文字)
UNPRINT    = " "           // 表示不可文字の表示文字    (1 文字)
DOUT       = 0xb111        // 画面表示の有無及び形式

XMSGSIZE   = 1000000        // 最大 SECS メッセージ・バイト長
XITMSIZE   = 1000000        // 最大 SECS アイテム・バイト長

TRCDIR     = " ."          // 通信ログ格納ディレクトリ
TRCSIZE    = 5000000        // 通信ログ最大サイズ        (Byte)
TRCOUT     = 0x0340        // 通信ログ出力モード

LICENSENAME= "株式会社令和研究所 諏訪事業所" // ライセンス先団体名称
LICENSECODE= "Reiwa Research Institute Ltd. Suwa Office" // ライセンス・コード
LICENSESER  = "1"           // ライセンスー連番号
LICENSEDATE= "20190101"     // ライセンス付与日
LICENSEKEY  = "ABCDE-FGHIJ-KLMNO-PQRST-UVWXY-Z1234" // ライセンス・キーコード

```

(4) トークン 及び パラメータ

```

SDEVICE0  = "COM1"      // 使用する Serial Communication Port #0 の名称
SDEVICE1  = "COM2"      // 使用する Serial Communication Port #1 の名称
//
// (注1) 本プログラムを RS232C <--> TCP/IP 中継器として使用する
//       場合 DEVICE0 もしくは DEVICE1 の設定を以下としてください。
//       ・ TCP/IP Server として機能させる場合
//         ":99999"       (先頭文字は ':' であること)
//       ・ TCP/IP Client として機能させる場合
//         "XXXXXX:99999"
//
//       XXXXXX : 接続先の IP-Address もしくはホスト名称
//       99999   : 使用する TCP/IP ポート番号


SSPEED    = 9600        // シリアル接続 ビット・レート
// = 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 等

SCSIZE    = 8           // シリアル接続 文字ビット・サイズ
// = 5 / 6 / 7 / 8
// (SECS-1 接続の場合、通常 =8 以外設定不可)

SPARITY    = 0          // シリアル接続 パリティ形式
// = 0 : 無し
//   1 : ODD パリティ
//   2 : EVEN パリティ
// (SECS-1 接続の場合、通常 =0 以外設定不可)

SSTOP     = 1           // シリアル接続 ストップ・ビット数
// = 1 / 2
// (SECS-1 接続の場合、通常 =1 以外設定不可)

SDTR      = 0           // DTR/DSR 制御
// = 0: DTR/DSR 制御に関して何も設定しない (現在状態を維持)
//   1: DTR/DSR 制御無効 とする
//   2: DTR/DSR 制御有効 OPEN 時に ON として維持する
//   3: DTR/DSR 制御有効 Handshake を有効とする

SRTS      = 0           // RTS/CTS 制御
// = 0: RTS/CTS 制御に関して何も設定しない (現在状態を維持)
//   1: RTS/CTS 制御無効 とする
//   2: RTS/CTS 制御有効 OPEN 時に ON として維持する
//   3: RTS/CTS 制御有効 Handshake を有効とする
//   入力データがバッファの 4 分の 3 を超えると RTS ラインを OFF
//   とし、半分以下になると RTS ラインを ON とする


STO       = 1000        // Serial Port Timeout (ms)
CWAIT    = 8000         // Port Close 処理後の待ち時間 (ms)
XINTER   = 1            // Port データ到着確認する間隔 (ms)
// 上記3パラメータは、本プログラムの動作状態を制御するもので、
// 通常、上記値を変更することはありません。

```

<< 次ページに続く >>

<< 前ページから続く >>

```

XC0L      =      100      // 通信 Base 表示のカラム数
                        // 画面左側の基本ログ表示ウインド（及び基本ログ・ファイル）
                        // に受信データを表示するカラム数
XROW      =      10000    // 画面左側の基本ログ表示ウインド 及び 右側の SECS ログ表示ウイン
                        // ドの表示領域の最大行数

DOT        = " .."        // 通信 Base 表示における被受信側の表示文字列                (3文字)
                        // 後述する通信 Base 表示において、受信データの表示側ではない側に
                        // 表示する文字。
                        // 通信 Base 表示において、受信データ 1Byte に使用する文字数により
                        // 使用する文字は以下の通りです。
                        // 1文字 : DOT に指定する文字列の 3 文字目
                        // 2文字 :      :      :      2、3 文字目
                        // 3文字 :      :      :      1、2、3 文字目
UNPRINT    = " "          // 通信 Base 表示において ASCII 文字表示を行う表示位置に、表示デー
                        // タ値が、<0x20 もしくは >0x7e の場合に表示する文字        (1文字)

DOUT       = 0xb111       // 画面表示、ログ出力の有無及び形式
                        // F DC A98 54 10
                        // +---+---+---+---+
                        // bit# 0   : 通信 Base 表示実行                (0:No 1:Yes)
                        //      1   : 通信 SECS 表示実行                (0:No 1:Yes)
                        // bit# 4   : 通信 Base ログ出力実行            (0:No 1:Yes)
                        //      5   : 通信 SECS ログ出力実行            (0:No 1:Yes)
                        //
                        // bit# 8-10 : 通信 Base 表示、ログの表示形式
                        //      =0: Hexa 標記のみ（縦 2 文字）            (1 桁 2 行)
                        //      1: ASCII 文字と Hexa 標記                (1 桁 3 行)
                        //      2: 制御文字のニモニックと Hexa 標記        (1 桁 3 行)
                        //      3: 制御文字のニモニックと ASCII            (1 桁 3 行)
                        //      4: 制御文字のニモニックと ASCII or Hexa    (1 桁 3 行)
                        //      5: 制御文字のニモニックと ASCII と Hexa    (1 桁 3 行)
                        //      6: 表示可能 ASCII 文字 (0x20~0x7e) のみ    (1 桁 1 行)
                        //      7: Hexa 標記のみ（横 2 文字）            (2 桁 1 行)
                        //
                        // bit#12-15 : 通信 SECS 表示、ログの表示形式
                        //      12   : SML 形式リスト出力                (0:No 1:Yes)
                        //      13   : Hexa ダンプ 形式出力                (0:No 1:Yes)
                        //      SML 形式、Hexa ダンプ形式共に、SECS-2 メッセージ
                        //      単位（即ち END Block 受信後）の出力
                        //      15   : 通信制御コード等（以下のデータ）出力 (0:No 1:Yes)
                        //      ・ ENQ, EOT, ACK, NAK
                        //      ・ SECS-1 データ・パケットのバイト長を示す 1Byte
                        //      ・ SECS-1 データ・パケット（各送信ブロック）
                        //
                        // (注 2) 通信 Base 表示形式の詳細は「4. (2) ■ 通信 Base 表示形式」
                        //      の説明を参照してください。
                        //
                        // (注 3) 本設定は、本プログラムの実行中に [表示] メニューで変更する
                        //      ことができます。ただし、ログ出力をするには必ず本パラメータ
                        //      で bit#4, 5 を ON(1) にする必要があります。その場合 [表示]
                        //      で「ログ出力しない」設定とすると、ログ出力を一時停止します。

```

<< 次ページに続く >>

<< 前ページから続く >>

```
XMSGSIZE    = 1000000    // 解析対象となる SECS メッセージの最大バイト長
XITMSIZE    = 1000000    // 解析対象となる SECS メッセージを構成する項目の最大バイト長
//
// (注 4) 通信データを SECS メッセージと見なしして解析し、SECS-2 SML
// 形式での表示及びログ出力を行う場合は、受信する可能性のある
// SECS メッセージの最大バイト長よりも大きな値を指定する必要
// があります。
```

```
TRCDIR      = "."        // 通信 Base ログ、通信 SECS ログを格納するディレクトリ
TRCSIZE     = 5000000    // 通信 Base ログ、通信 SECS ログの最大 Byte サイズ
TRCOUT      = 0x0340     // 通信 Base ログ、通信 SECS ログの出力モード
//      D      8  654
// +-----+-----+-----+
//      +-----+ |+++ 出力時ログ・ファイル日毎切替指定
//      |         |      =0: 行わない
//      |         |      1: 行う (日付のディレクトリを作成し、
//      |         |           その中に格納する)
//      |         |      2: 行う (日付はファイル名の先頭に付与)
//      |         |      3: 行う (日付はファイル名の後ろに付与)
//      |         |
//      |         +----- 指定サイズ (TRCSIZE) 超でファイル切替指定
//      |         =0: 行わない
//      |         1: 行う
//      |
//      +--- 残しておくトレース・ファイル数
//      =0      : 全てのファイルを残す
//      1 - 63 : ファイルを切り替えた時点より、指定番号以前
//              のファイルを削除する
//
// (注 5) 通信 Base ログ、通信 SECS ログ・ファイルは、TRCDIR が示す
// ディレクトリに、以下の名称で作成します。
// ・通信 Base ログ : tdISerMonB.log
// ・通信 SECS ログ : tdISerMonS.log
//
// (TRCOUT&0x0030) != 0 の場合、指定の形式で日付 (YYMMDD) を付与
// します。
// (TRCOUT&0x0040) != 0 の場合、ファイル名は以下となります。
// ・通信 Base ログ : tdISerMonB.9999.log
// ・通信 SECS ログ : tdISerMonS.9999.log
//      +--- 一連番号
```

<< 次ページに続く >>

<< 前ページから続く >>

```
LICENSENAME= "XXXXXXXX" // 実行ライセンスの供与を受けた個人、もしくは団体の名称
LICENSECODE= "XXXXXXXX" // 実行ライセンスの供与を受けた個人、もしくは団体のコード名
LICENSESER = "999" // 実行ライセンスの供与を受けた個人、もしくは団体でのシリアル#
LICENSEDATE= "YYYYMMDD" // 実行ライセンスの取得年月日 (YYYYMMDD 形式で指定)
LICENSEKEY = "XXXX-XXXX-..." // 実行ライセンスのキーコード
//
// [注意]
// ライセンス・キーコードを設定しない場合、過ったライセンス・キー
// コードを設定した場合、期限切れの動作確認用ライセンス・キーコー
// ドを設定した場合、一定時間の動作後に動作を停止します。
// また、有効な動作確認用ライセンス・キーコード (Default License)
// を設定した場合も、数時間の動作後に動作を停止します。
// 動作確認用ライセンス・キー (Default License) ではない、正式に
// 実行ライセンスの供与を受けたライセンスの場合は、異なるPCには、
// 必ず異なるライセンス・キーコード情報を記述しなければなりません。
// 同一PC内で本プログラムを複数動作させる場合は、それらの動作
// プログラムに個別に異なるライセンス・キーコードを設定する必要は
// ありません。
```

4. 操作説明

(1) 起動

インストールしたフォルダにある tdlSerMon.exe（もしくは tdlSerMon.exe へのショートカット）をダブル・クリック等により起動します。

（参考）起動オプションとして、以下を指定可能です。

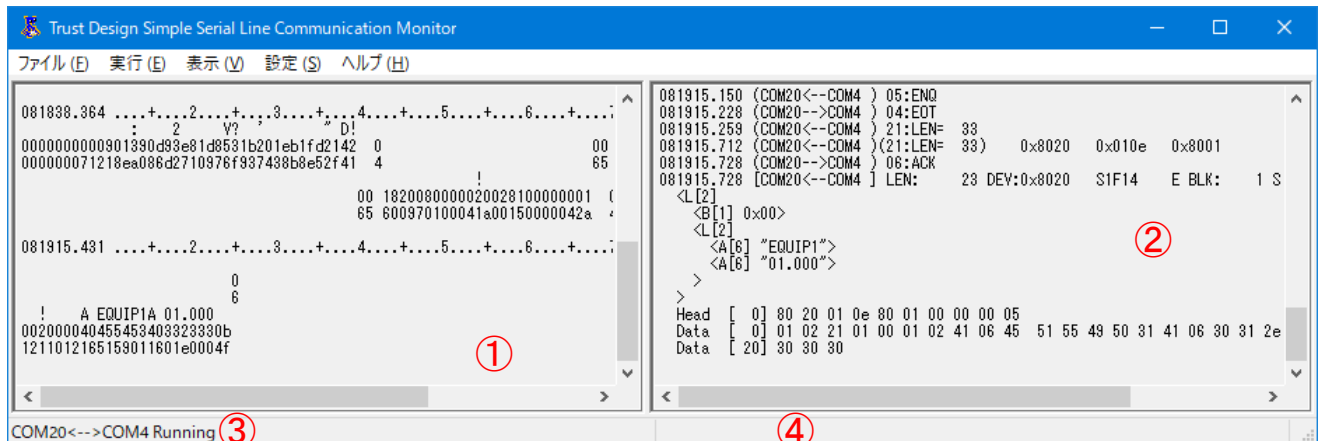
+i ini_file : 動作条件設定ファイルの名称
省略時 : tdlSerMon.ini

+s Section : 動作条件設定ファイル内の使用するセクション名称
省略時 : SERMON

+S style_file : 動作スタイル（位置、サイズ、指定パラメータ）を保存するファイルの名称
省略時 : tdlSerMonWin.ini
 （使用する動作条件設定ファイルの拡張子を除くファイル名に“Win”
 を付与したファイル名）

（注 1）動作スタイル保存ファイルは、動作条件設定ファイルと同一のフォルダに格納します。

(2) 画面操作説明



- ① : 通信 Base 表示 ... 受信したシリアル通信の各バイト値を表示します。
表示形式を下方の「■ 通信 Base 表示形式」に詳述します。
- ② : 通信 SECS 表示 ... 受信したシリアル通信データを SECS-2 メッセージの送受信とみなして解析した結果の SECS-2 メッセージを表示します。
表示形式を下方の「■ 通信 SECS 表示形式」に詳述します。
- ③ : 実行状態 現在のプログラム実行状態を表示します。
- ④ : TCP/IP 状態 本プログラムを RS232C<-->TCP/IP 変換器として使用した場合、TCP/IP 接続状態を表示します。

■ 通信 Base 表示形式

0. Hexa 表示のみ (縦 2 文字) (1 桁 2 行)

```

00Byte 目の受信時刻 (hhmmss.m 秒)
132020.838 .....2.....3.....4 — ヘッダ行 .. 1Byte 目の受信時刻とルーラ
0 082808000000a 0 } SDEVICE0 より受信したデータ (2 行 1 組)
5 a001101000114 4 } Hexa 標記を縦 2 文字で表示する
0 00 c0200800000004044454240 } SDEVICE1 より受信したデータ (2 行 1 組)
4 65 600120100011f16cd12b016
|
受信したこの表記行内の 20 番目の Byte データ (0x20=' ')

```

(注 1) 画面上の各行には、出力時刻の表示はありませんが、ログ・ファイルへの出力には出力日、時刻 (YYMMDD. hhmmss.m 秒) の表示を各行の先頭に付与します。

(注 2) ログ・ファイルに記録される各行の先頭に付与する時刻 (YYMMDD. hhmmss.m 秒) は、その行をログ・ファイルに記録した時刻です。従って、通常はその行に記録した最後の Byte データを受信した時刻の直後の時刻となります。その行の先頭 Byte を受信した時刻は、ヘッダ行に記録された時刻となります。

1. ASCII 文字と Hexa 表示 (1 桁 3 行)

```

00Byte 目の受信時刻 (hhmmss.m 秒)
132045.338 .....2.....3.....4 — ヘッダ行 .. 1Byte 目の受信時刻とルーラ
0 082808000000a 0 } SDEVICE0 より受信したデータ (3 行 1 組)
5 a001101000215 4 } 1 行目に表示可能な ASCII Code を表示し、2、3 行目に Hexa
                        A LMARK A } 標記を縦 2 文字で表示
0 00 c0200800000004044454240 } SDEVICE1 より受信したデータ (3 行 1 組)
4 65 600120100021f16cd12b016
|
受信したこの表記行内の 31 番目の Byte データ (0x41=' A')

```


2. 制御文字のニモニクと Hexa 表示 (1 桁 3 行)

5. に示す「制御文字のニモニクと ASCII と Hexa 表示」とほぼ同じですが、Byte データが表示可能な ASCII コードでも、1 行目への表示を行いません。

3. 制御文字のニモニクと ASCII 表示 (1 桁 3 行)

5. に示す「制御文字のニモニクと ASCII と Hexa 表示」とほぼ同じですが、2、3 行目への Hexa 表示を行いません。

4. 制御文字のニモニクと ASCII もしくは Hexa 表示 (1 桁 3 行)

5. に示す「制御文字のニモニクと ASCII と Hexa 表示」とほぼ同じですが、表示対象 Byte が表示可能な ASCII 文字の場合は、それを 1 行目に表示し、表示可能でない場合は、2、3 行目へ Hexa 表示します。

5. 制御文字のニモニクと ASCII と Hexa 表示 (1 桁 3 行)

0Byte 目の受信時刻 (hhmmss.m 秒)
 132046.975 2 3 4 — ヘッダ行 ... 1Byte 目の受信時刻とルーラ

E L S SNNNES E	}	SDEVICE0 より受信したデータ (3 行 1 組)
ON F828080UUUTOa 0		
6Q 001H0HLLLXH6 T		
@^ P) E AE N	}	SDEVICE1 より受信したデータ (3 行 1 組)
145d99999c8dcccc52 0 CN cU2		
00e99999a0eccccc09 T KQ 6L0		

受信 Byte が ASCII 制御コード (0x00~0x1f 及び 0x7f) の場合、そのニモニクを 3 行で縦表示。
 その他の場合は Hexa 標記を縦 2 文字で 2、3 行目に表示。
 また、その Byte が表示可能な ASCII コードの場合は、1 行目にその ASCII コードを表示。

受信したこの表記行内の 38 番目の Byte データ
 制御文字でなく、表示可能な ASCII コードでもないで、Hexa 表示のみ

受信したこの表記行内の 21 番目の Byte データ (0x04="EOT")
 制御文字なのでニモニク表示

受信したこの表記行内の 17 番目の Byte データ (0x50='P')
 制御文字でなく、表示可能な ASCII コードなので、
 1 行目に ASCII コード (P) を表示し、2、3 行目に Hexa 表示。

6. 表示可能 ASCII 文字 (0x20~0x7e) のみ (1 桁 1 行)

0Byte 目の受信時刻 (hhmmss.m 秒)
 132114.351 2 3 4 — ヘッダ行 ... 1Byte 目の受信時刻とルーラ

SDevice0 Karano Data	}	SDEVICE0 より受信したデータ (1 行 1 組)
A L MARK A		

受信したこの表記行内の 33 番目の Byte データ ('L')

7. Hexa 表示のみ (横 2 文字) (2 桁 1 行)

0Byte 目の受信時刻 (hhmmss.m 秒)
 132143.440 1 2 — ヘッダ行 ... 1Byte 目の受信時刻とルーラ

05 0a8020810180010000000601a9 04	}	SDEVICE0 より受信したデータ (1 行 1 組)
04 0605 c6002		

受信 Byte の Hexa Code 2 文字を横に表示

受信したこの表記行内の 17 番目の Byte データ
 (SDEVICE0 からの Byte データ 0x04)

■ 通信 SECS 表示形式

145057.257 (COM11-->COM21) 05:ENQ

145057.537 (COM11<--COM21) 04:EOT

受信 Byte が制御コードの場合、ニモニック表示

受信 Byte が SECS-1 プロトコルから外れている場合で、その Byte が表示可能な ASCII Code の場合は、その ASCII Code を表示

受信 Byte を Hexa 表示

データの方向を示す。(この例の場合 COM21 の Port からの受信を示します)

受信時刻 (hhmmss.m 秒)

ログ・ファイル出力の場合は、行頭に日付を含めた受信時刻を
"YYMMDD.hhmmss.m 秒"の形式で付与します。

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

受信 Byte が SECS-1 プロトコル上の「送信ブロック長(Byte 数)」である場合、その値を 10 進表示します。(この例の場合 254Bytes であることを示します)

145057.777 (COM11-->COM21) (fe:LEN= 254)

0x8020

0x860b

0x0001

0x0000.0001:0x36e6

先に受信した「送信ブロック長」の Byte
データを受信した場合、その SECS ブロック
ヘッダ情報及びチェックサム値を表示します

SECS ヘッダ：デバイス I D

SECS ヘッダ：メッセージ I D

SECS ヘッダ：ブロック番号

SECS ヘッダ：ソース I D

SECS ヘッダ：トランザクション I D

チェックサム

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

ブロック番号の'E'ビットがON

受信 SECS ヘッダのブロック番号の'E'ビットがONであった場合、一連の SECS
メッセージ受信が完了したものと、受信 SECS メッセージのヘッダ及び
メッセージ本体を SML 形式で表示します。

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]

システムバイト

ブロック数

E-Bit ON 受信を示す

SF コード

W-Bit ON の場合に表示

デバイス I D (R-Bit を含む)

SECS メッセージのバイト数

<< 次ページへ続く >>

<< 前ページから続く >>

```

        <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>

```

以下に Hexa 形式でのダンプ出力例を示します。

```

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

```

(注 1)

一連の Hexa 表示の先頭行、最終行を除き（即ち中間行では）、その行の全データが =0 である場合は、その行を出力しません。

[999]が、その行の先頭バイトの全体での位置を示しますので、未表示データであるか否かを[999]の連続性でご判断ください。

(3) メニュー

(a) [ファイル]

- ・ アプリケーションの終了 ... tdlSerMon を終了します。

(b) [実行]

- ・ 開始 シリアル通信ラインのモニタ処理を開始します。
- ・ 停止 シリアル通信ラインのモニタ処理を停止します。

(注 1) シリアル通信ラインの I/O において何らかの障害が発生している等の場合、[停止] 処理に時間がかかる場合があります。そのような場合に、即座にプログラムを終了したい場合は、タイトルバー右端の [X] により、終了してください。

(c) [表示]

- ・ 通信ログ表示クリア 通信 Base 表示ウインド ①、通信 SECS 表示ウインド ② をクリアします。
- ・ 通信ログ最終行表示 通信 Base 表示ウインド ①、通信 SECS 表示ウインド ② のスクロールバーを、最終行を表示されている状態にします。
(高速で自動スクロールしている状態等で、通常のスクロール・バールの操作では、なかなか最終行が表示された状態にならない場合に使用します。)
- ・ 基本表示出力 通信 Base 表示ウインド ① に、指定シリアル・ポートから受信した Byte データを指定の形式で表示します。
- ・ 基本ログ出力 通信 Base 表示ウインド ① に表示する内容と同様の出力を所定の通信 Base ログ・ファイルに行います。
- ・ 基本出力形式 通信 Base 表示ウインド ① に表示する通信 Byte データの表示形式を以下の選択肢から選択します。
 - 0. 16 進数(縦 2 行)のみ
 - 1. ASCII+16 進数
 - 2. 制御文字ニモニック+16 進数
 - 3. 制御文字ニモニック+ASCII
 - 4. 制御文字ニモニック+ASCII もしくは 16 進数
 - 5. 制御文字ニモニック+ASCII+16 進数
 - 6. 表示可能 ASCII コードのみ
 - 7. 16 進数(横 2 桁)のみ

(注 2) 通信 Base 表示出力形式の詳細は「4. (2) ■ 通信 Base 表示形式」の説明を参照してください。

- ・ SECS 表示出力 通信 SECS 表示ウインド ② に、指定シリアル・ポートから受信した Byte データ SECS-2 メッセージと想定して解析した結果を表示します。
- ・ SECS ログ出力 通信 SECS 表示ウインド ② に表示する内容と同様の出力を所定の通信 SECS ログ・ファイルに行います。
- ・ SECS 出力形式 通信 SECS 表示ウインド ② に表示する内容を指定します。
 - ・ リスト形式表示 ... 最終ブロックを受信後、SML 形式で表示
 - ・ 16 進数表示 最終ブロックを受信後、16 進数ダンプ出力
 - ・ 制御コード表示 ... SECS メッセージ・ブロックの送受信プロトコルに関する制御コードを出力

(d) [設定]

- ・ 自動開始 プログラム起動後、自動的に処理を開始する設定とします。
本設定は、次のプログラム起動時に有効となります。
- ・ 位置サイズ保存 プログラム終了時に、本プログラムの終了時点での位置、サイズ等を保存し、次回起動時にウインドの状態を同じ状態に復元します。

(e) [ヘルプ]

- ・ バージョン情報 本プログラムのバージョン情報を表示します。