

# Serial Lines Analyzer

**v3.0.0**

Manuale d'uso.

SLA è un programma che permette di analizzare una linea seriale RS-232C, monitorare i criteri di interfaccia in real-time, rappresentare i dati in transito in fari formati, decodificare i dati rispetto ad un protocollo settato dall'utente ed emulare con il protocollo stesso, sia in modalità Master che in modalità Slave, una serie di comandi settati dall'utente.

## **Connessione alla linea seriale.**

SLA può gestire una oppure due linee seriali contemporaneamente.

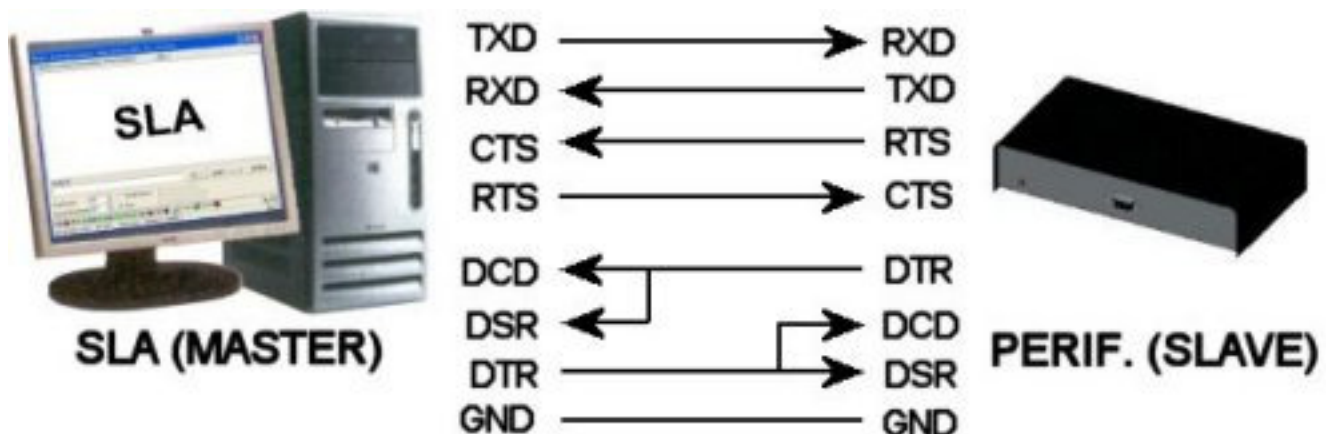
La connessione ad una singola seriale permette a SLA di 'vedere' una singola periferica ( o rete di periferiche) e di comportarsi da Master, oppure di essere 'visto' da un Host come singola periferica e quindi di comportarsi da Slave.

La connessione ad una doppia seriale contemporaneamente permette a SLA di 'spiare' il flusso di dati tra due unità, in modo bidirezionale, senza interferire con il dialogo.

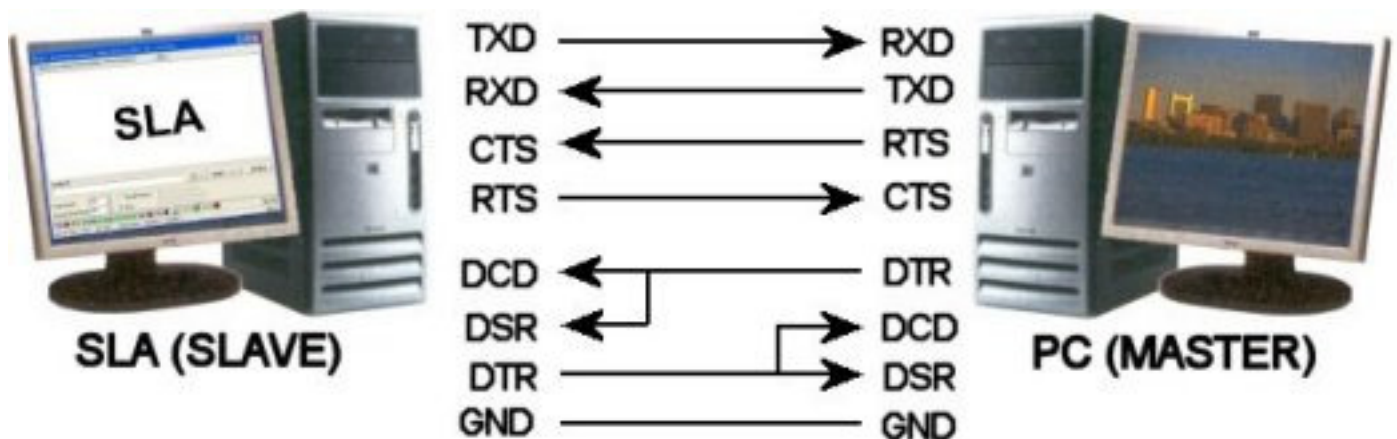
## **Connessione ad una singola porta RS-232C**

( Le piedinature dei connettori 9 e 25 poli sono nella Appendice A)

### Configurazione 1

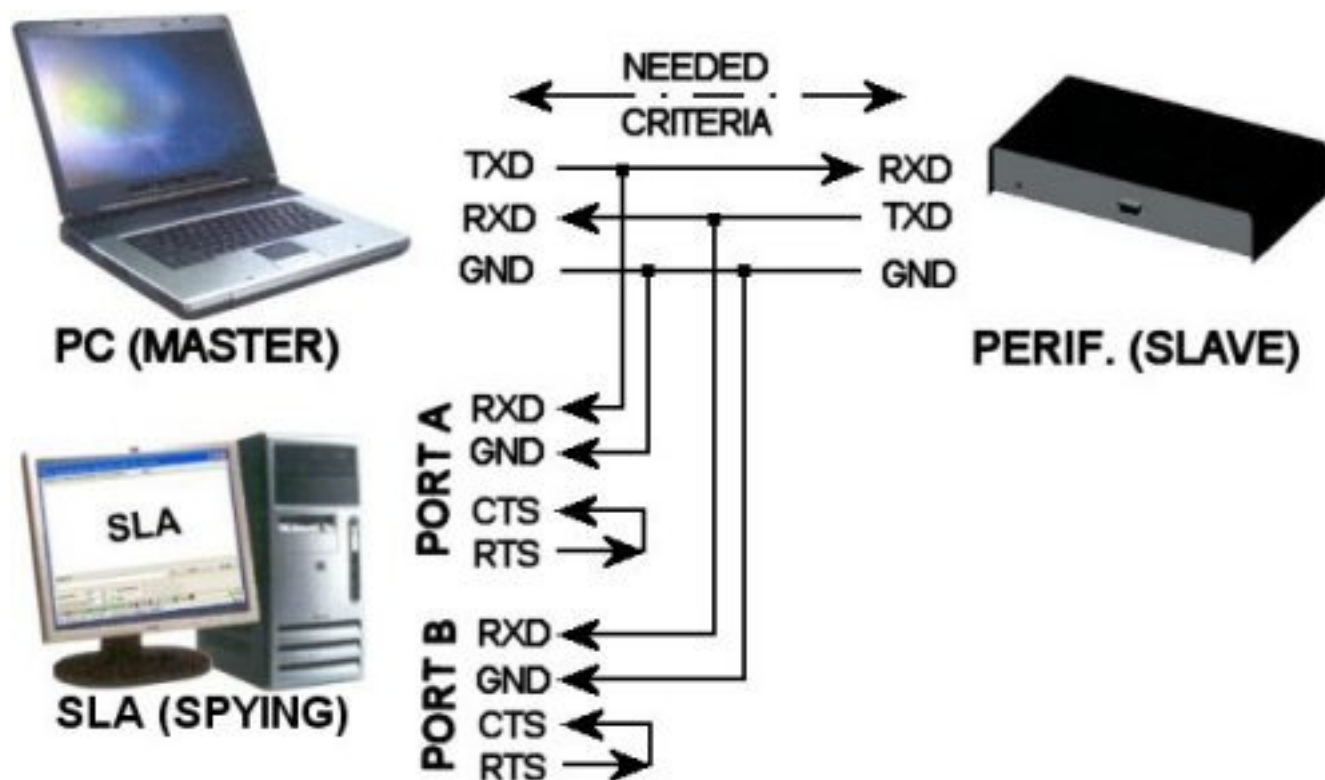


### Configurazione 2



## **Connessione a due porte RS-232C**

### Configurazione 3

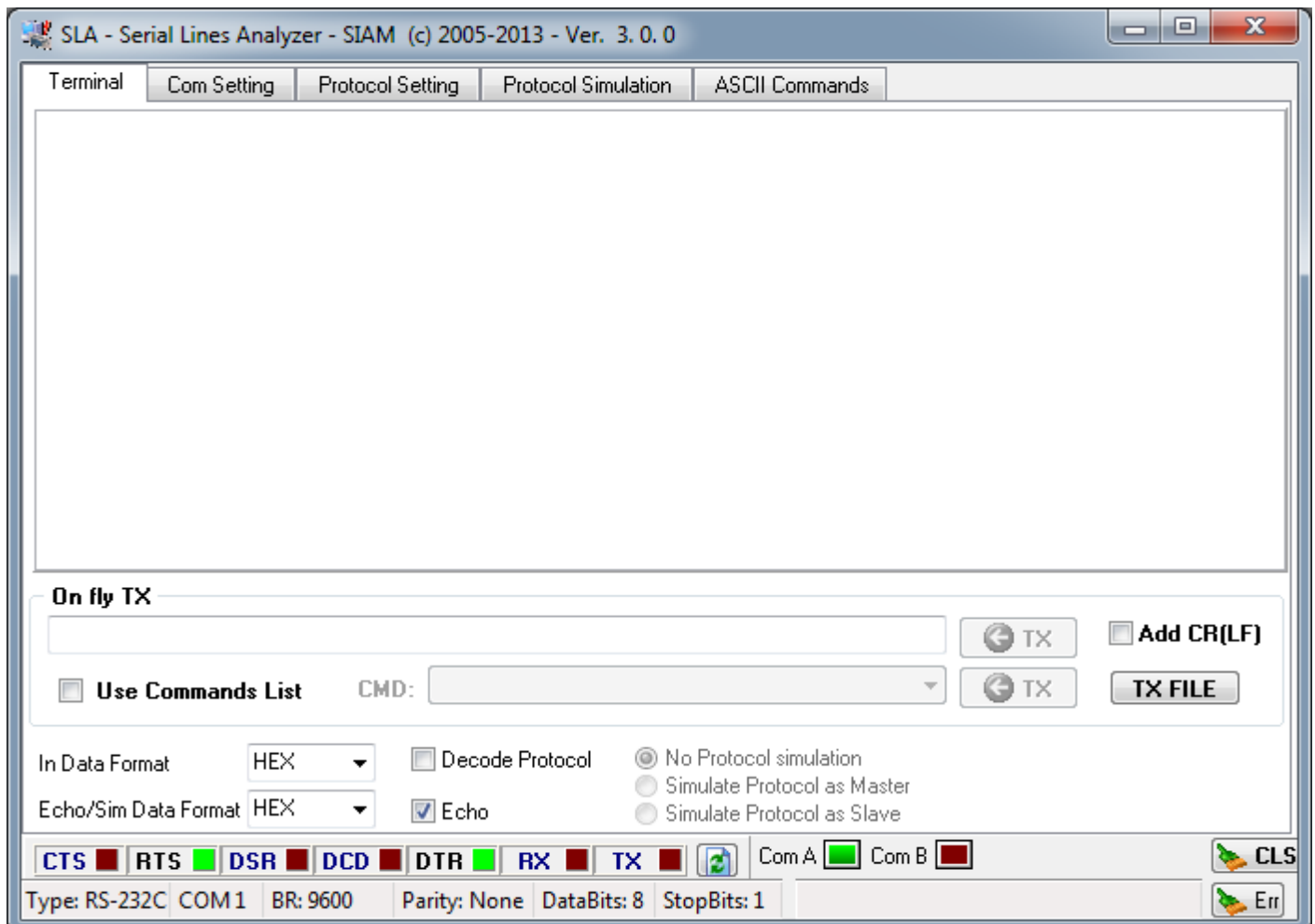


### **Il programma**

SLA configura dinamicamente l'interfaccia utente in funzione dei settaggi specifici sia delle porte, sia del protocollo, sia della modalità Master / Slave. Alcuni oggetti quindi sono visibili solo in determinate situazioni di configurazione.

La finestra principale dispone di 5 *Tab* :

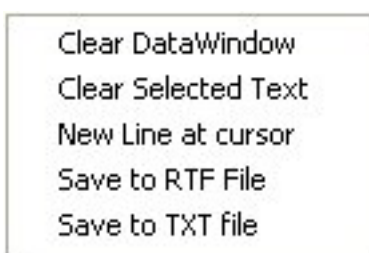
# Terminal



Questa *Tab* contiene la finestra di rappresentazione dei dati e quindi costituisce la *Tab* in cui si rimane normalmente per lavorare.

I dati rappresentati nella finestra sono differenziati in base al loro colore. Ogni tipologia di dato ha un colore diverso e configurabile dall'utente (vedi più avanti).

La finestra dati ha associato il seg. PopUp Menu:



che permette di interagire con i dati ricevuti o trasmessi.

L'area **OnFly TX** permette in modalità **No Protocol simulation** e **Simulate as Slave** di trasmettere una stringa, un comando tabulato oppure un file.

La stringa può essere scritta al momento nel campo apposito, salvata con il tasto **Save** e ricaricata a piacere con il tasto **Load**.

Un comando tabulato, inserito da file salvato precedentemente o a mano nella tabella della TAB 'ASCII Commands', viene selezionato dalla ComboBox 'CMD' e abilitato dalla spunta 'Use Command List'.

Il file viene trasmesso tutto in una volta, come fosse una stringa unica.

Sia la stringa, sia il file sono trasmessi così come introdotti, senza essere considerati facienti parte del protocollo.

Per le funzioni OnFly è possibile aggiungere automaticamente un CR ( o CR/LF) alla fine del comando, in base alla spunta 'Add CR(LF)' e alla spunta relativa della TAB 'Com Setting'.

#### Nota sul formato di introduzione dei dati.

Tutti i campi di SLA che prevedono introduzione di uno o più caratteri accettano gli stessi nei seg. formati ( anche mischiati fra loro a piacere):

#### **Formato**

Caratteri ASCII diretti

Caratteri control

Notazione decimale

Notazione esadecimale

#### **Esempi**

ALBERO prova Temp.

^C ( 03hex) ^@ (00hex)

#003 #200 #017

-formato obbligato a 3 caratteri dopo il # - i.e. #xxx

\$03 \$FA \$a2

-formato obbligato a 2 caratteri dopo il \$ - i.e. \$xx

-la parte letterale può essere sia in minuscolo che in maiuscolo

Le stringhe possono essere formate da un misto di tutti i formati contemporaneamente, quindi stringhe valide sono : \$02#001TEMP\$FF #002Prova\$03 ecc...

Se la stringa contiene formati non validi o caratteri non validi per il formato, SLA presenta un messaggio **Malformed Input String : xxx** ove xxx è la parte errata della stringa.

Le due caselle di selezione **IN Data Format** e **Echo-Sim Data Format** permettono, in modo differenziato, di selezionare il formato di rappresentazione dei dati. Le possibili scelte sono:

|                |  |
|----------------|--|
| ASCII          | ASCII standard – tutto il set di caratteri   |
| ASCII filtered | Come sopra ma i caratteri di controllo ( 0-31) vengono rappresentati in esadecimale. |
| HEX            | Tutto in esadecimale   |
| DEC            | Tutto in decimale  |
| BIN            | Tutto in binario 8 bit   |

La casella di spunta **Echo** , se spuntata, permette di rappresentare nella finestra dati tutto ciò che viene trasmesso.

La casella di spunta **Decode Protocol** permette di attivare il motore di decodifica del protocollo configurato nella **Tab Protocol Setting**.

L'attivazione di questa spunta abilita le tre possibilità presenti al suo fianco:

#### **No Protocol simulation**

I dati in ingresso dalla seriale vengono interpretati come pacchetti del protocollo configurato, analizzati e rappresentati di conseguenza. SLA si limita a decodificare e rappresentare i dati in ingresso.

#### **Simulate Protocol as Master**



Attivando questa configurazione, appare un nuovo oggetto, **Command to Simulate**, che permette a SLA di trasmettere uno dei comandi configurati nella tabella di simulazione,

presente nella **Tab Protocol Simulation**. In pratica SLA diviene Master di un sistema Master / Slave e comincia a gestire tutte le operazioni legate alla trasmissione di un comando. Queste operazioni comprendono la gestione dell'Hand-Shake, della ricezione di eventuali risposte al comando di protocollo trasmesso e all'attesa di eventuale sincronismo finale ( SYNC).

La situazione massima configurabile è rappresentata dalla tabella:

| Master  |   | Slave    |
|---------|---|----------|
| Domanda | → |          |
|         | ← | ACK      |
|         | ← | Risposta |
| ACK     | → |          |
|         | ← | SYNC     |



### Simulate protocol as Slave


In questa configurazione SLA diviene invece lo Slave del sistema, cioè, alla ricezione di un pacchetto di protocollo che contiene un comando, oltre ad analizzarlo, decodificarlo e rappresentarlo, comincia anche la sequenza tipica dello Slave, cioè gestire il relativo Hand-Shake, trasmettere una eventuale risposta connessa e, se settato, trasmettere anche un SYNC finale.



Le operazioni compiute da SLA in modalità di simulazione Master o Slave dipendono dalla configurazione del protocollo eseguita nella **Tab Protocol Setting**.


### Area LED di Stato

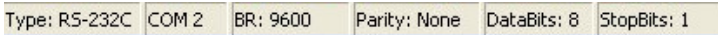
SLA monitorizza in tempo reale tutte le linee hardware della porta A in uso e rappresenta lo stato dei segnali sui Led opportuni.

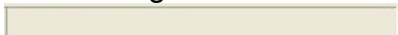

RTS  e DTR  sono anche due tasti che permettono di attivare o disattivare le relative linee hardware.

Il tasto Refresh Lights Status  permette di aggiornare manualmente lo stato dei LEDs.

I due LEDs   danno informazioni sullo stato di apertura delle porte A e B, in base al loro settaggio.

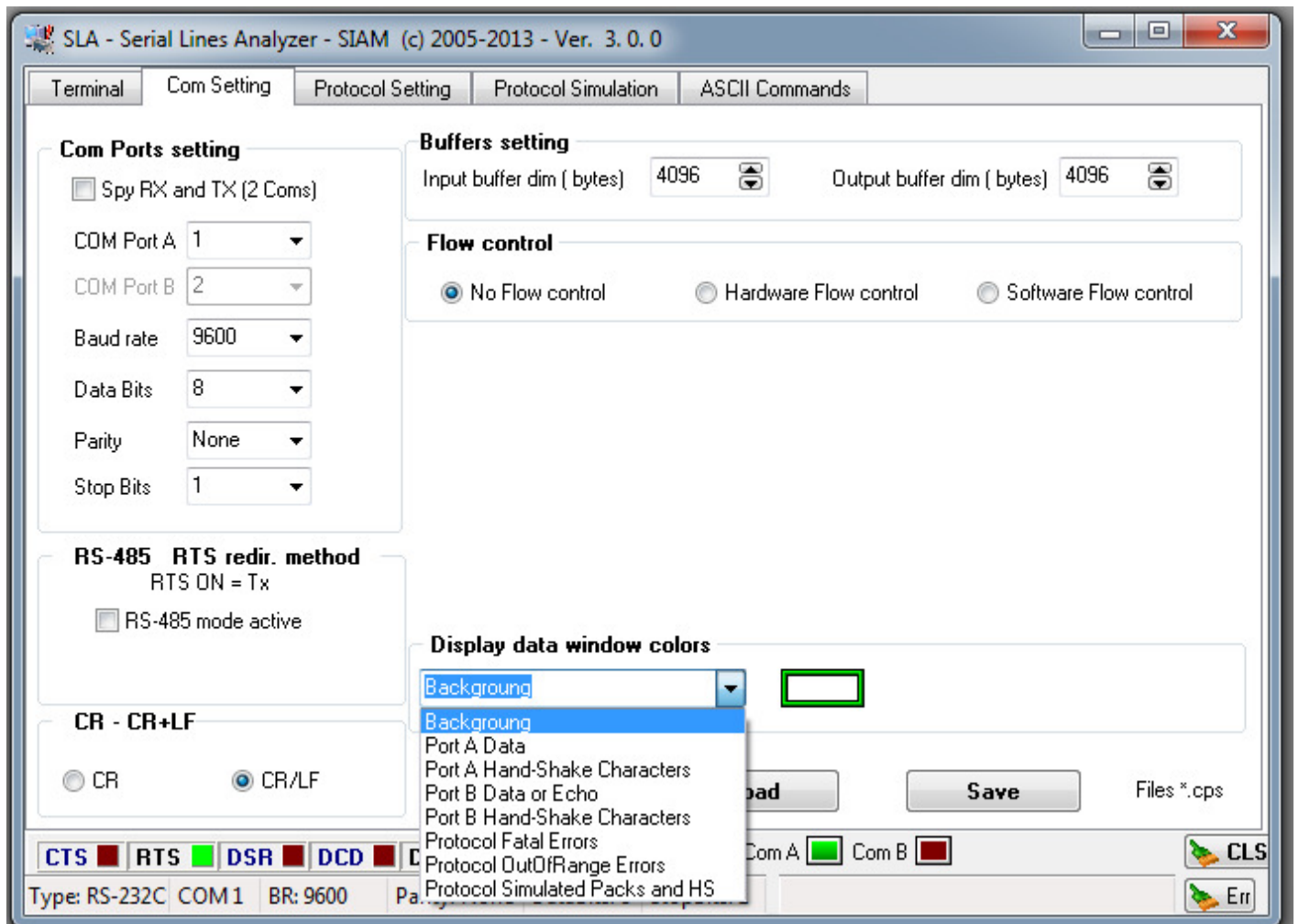
Il tasto CLS  permette di pulire la finestra dati ( Clear Screen), voce presente anche nel PopUp Menu della finestra dati.

La riga di stato in fondo alla finestra principale  riassume i parametri di configurazione della porta al momento in uso, così come settati nella **Tab Com Setting**.

L'ultima parte della riga di stato serve a rappresentare eventuali errori di comunicazione hardware.   Il tasto Err permette di pulire questo campo, dato che, se avviene un errore di comunicazione, questo rimane indicato fino alla pressione del tasto Err.

# Com Setting

Questa *Tab* permette di configurare le porte del sistema dove 'gira' SLA.



## Area Com Port Setting

### Spy RX and TX ( 2 Coms)

Se spuntata, SLA spia la comunicazione tra un Master ed uno Slave. Vedi immagine *Configurazione 3* all'inizio di questo documento.

### COM PORT A

Porta RS-232C principale

### COM PORT B

Porta RS-232C secondaria ( solo per Spy mode)

### BAUD RATE

Configurabile da 150 a 115.200 bps in tutti gli steps standard

### DATABITS

Configurabile come 5,6,7,8

### PARITY

Configurabile come: NONE, ODD, EVEN, MARK,SPACE

### STOP BITS

Configurabile come 1,1.5,2

### RS-485 mode (RTS redir)

Se spuntato, permette di gestire una linea monodirezionale RS-485, la cui direzione è controllata dal segnale hardware RTS.



## Area CR - CR+LF

Permette di selezionare se aggiungere CR opp. CR+LF ai comandi OnFly e tabulati

## Area Buffer Setting

Permette di dimensionare i buffer di ricezione e trasmissione usati dalle porte.  
I buffers di entrambe le porte vengono settati al medesimo valore.

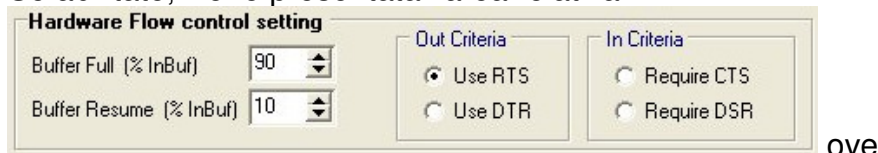
## Area Flow Control

### No Flow Control

Nessun controllo di flusso, ne hardware ne software

### Hardware Flow Control

Se abilitato, viene presentata l'area relativa:



### Buffer Full (% InBuf)

Stabilisce la percentuale del buffer di ingresso a cui bloccare il trasmittente.

### Buffer Resume (%InBuf)

Stabilisce la percentuale del buffer di ingresso a cui riabilitare il trasmittente.

### Out Criteria

#### Use RTS

Usa RTS come criterio hardware per bloccare il trasmittente

#### Use DTR

Usa DTR come criterio hardware per bloccare il trasmittente

### In Criteria

#### Require CTS

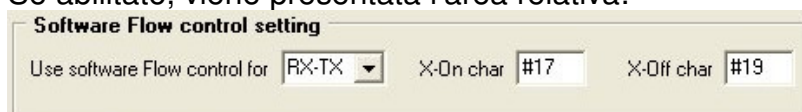
Vuole CTS come criterio hardware per trasmettere

#### Require DSR

Vuole DSR come criterio hardware per trasmettere

## Software Flow Control

Se abilitato, viene presentata l'area relativa:



### Use software Flow Control for

Stabilisce quando usare il controllo di flusso software. Le possibilità sono:  
RX, TX, RX-TX

### X-On char

Carattere usato come X-On

### X-Off char

Carattere usato come X-Off

## Area Display Data Window Colors

Qui è possibile configurare i colori relativi alla tipologia del dato rappresentato dalla finestra dati.

Le tipologie previste a cui è possibile associare un colore differente sono:

### Background

sfondo finestra

### Port A Data

Dati ricevuti dalla porta A

|                                     |   |
|-------------------------------------|---|
| <b>Port A Hand-Shake Characters</b> | Se decodifica protocollo attiva, colore dei soli dati validati.<br>Caratteri di Hand-Shake ricevuti dalla porta A<br>( ACK, NAK,SYNC) |
| <b>Port B Data or Echo</b>          | Colore dei dati ricevuti dalla porta B opp. colore di Echo<br>Se decodifica protocollo attiva, colore dei soli dati validati.         |
| <b>Port B Hand-Shake Characters</b> | Caratteri di Hand-Shake ricevuti dalla porta B<br>( ACK, NAK,SYNC)  |
| <b>Protocol Fatal Errors</b>        | Messaggi di errore fatale relativi alla decodifica del<br>Protocollo ( CKS error, Time Out vari)                                      |
| <b>Protocol OutOfRange Errors</b>   | Errori del protocollo non fatali, ma di dato fuori dai limiti<br>Settati ( ID, COM, LEN)  |
| <b>Protocol Simulated Packs</b>     | Colore di tutto ciò che viene trasmesso per simulazione<br>Di protocollo, sia in modo Master che in modo Slave.                       |

Al variare di anche un solo dato, il tasto **Apply** diventa attivo e permette di applicare le varianti fatte alla configurazione. E' consigliabile apportare prima tutte le modifiche poi metterle in uso con il tasto **Apply**.

La configurazione di questa *Tab* può essere salvata 'in toto' con un nome qualsiasi,per poi essere ricaricata a piacere. I tasti **Load** e **Save** permettono la carica ed il salvataggio.

Non è necessario il tasto **Apply** quando si carica una configurazione da disco. Questa viene messa in linea al volo.

L'estensione dei files di salvataggio di config. Delle porte deve essere obbligatoriamente .CPS (ComPortSetting).

Quando si salva una configurazione, fornire il nome senza l'estensione e SLA penserà a mettere quella giusta.

SLA ha anche un file default di configurazione delle porte ( Port\_CFG.CPS) che viene caricato allo start del programma. Per modificare le impostazioni di default delle porte, salvare quindi con il nome Port\_CFG.CPS.



# Protocol Setting

Questa Tab permette di configurare il protocollo in uso.

SLA permette solo le combinazioni possibili.

L'area Packet Format rappresenta in tempo reale la forma del pacchetto dati in base ai settaggi che vengono effettuati. Questa caratteristica permette una comprensione immediata se i settaggi impostati sono quelli che servono.

## Area Pack Start-Stop

**STX char** Carattere di Start pacchetto

**ETX char** Carattere di Stop pacchetto

Nota: ETX e LEN sono a mutua esclusione.

## ID

**Valid range min** Valore minimo di IDentificativo periferica

**max** Valore massimo di IDentificativo periferica

Questi valori servono ad un controllo di sicurezza sugli ID che possono transitare. In caso di ID fuori dal range di questi due valori, SLA marca l'ID ricevuto con il colore di OutOfRange.

## ID in simulated packs

Valore dell'ID in simulazione di protocollo.

In modo Master, questo sarà l'ID della periferica interrogata.

In modo Slave, questo sarà l'ID della periferica simulata.

In tutti i casi è gestito l'ID 0, di solito utilizzato per i comandi in Broadcast.

**Area Pack Lenght**      Nota: ETX e LEN sono a mutua esclusione.  
**Valid range min**      Valore minimo di lunghezza pacchetto

**max**      Valore massimo di lunghezza pacchetto

Questi valori servono ad un controllo di sicurezza sulla lunghezza Dei pacchetti che possono transitare. In caso di LEN fuori dal range di questi due valori, SLA segnala la cosa marcando la LEN con il colore di OutOfRange.

**Area Command**  
**Valid range min**      Valore minimo del comando

**max**      Valore massimo del comando

Vale quello che è stato dello per la lunghezza.

#### **Swap CMD nibble in Anwser ( i.e. 07 > 70)**

Se spuntata, questa opzione permette di fare in modo che il comando delle risposte abbia invertiti fra loro i primi 4 bits con i secondi 4. Es. Se il comando della domanda è 07hex, la risposta avrà come codice comando 70hex. Questo permette di discriminare tra una domanda ed una risposta senza stare sincroni con la trasmissione, cioè senza sapere chi è il Master e chi lo Slave. Notare però che, se si desidera univocità assoluta, lo Swap Nibble limita a 16 il numero di comandi possibili.

#### **Area Hand-Shake**

**ACK**      Carattere di Aknowledge

**NAK**      Carattere di Not Aknowledge

**SYN**      Carattere di sincronismo. L'uso del SYNC può essere disabilitato 'in toto' anche se presente nelle tabelle di simulazione di protocollo ( vedi più avanti).  
Nota: il carattere di sincronismo viene usato a volte in alcune comandi che impiegano diverso tempo per essere completati. Un esempio tipico è la scrittura in EEPROM di un pacchetto di dati. Il Master invia il pacchetto di dati allo Slave che, dopo eventuale risposta associata al comando, comincia la scrittura reale dei dati in EEPROM. Alla fine della scrittura lo Slave informa il Master del completamento del comando con un carattere di SYNC. Per scritture successive il SYNC alleggerisce il lavoro del Master e velocizza la comunicazione.

#### **Area Various**

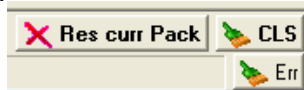
##### **Time Out to complete protocol packs**

Espresso in secondi, è il tempo entro cui il pacchetto dati deve arrivare tutto.

Per pacchetto dati si intende da STX a ETX opp da STX per LEN bytes.

Se settato = 0, non viene più gestito il Timeout di pacchetto. In questo caso appare nella finestra Terminal un tasto optional Res cur Pack che permette di resettare manualmente il motore di decodifica del protocollo in caso di pacchetti tronchi. Se è attivo questo Time Out, in caso di pacchetto tronco, il motore di decodifica viene resettato automaticamente allo scadere del tempo.

Questo Time Out è attivo in tutte le condizioni in cui è attiva la decodifica del protocollo.



### Time Out for ACK – NAK

Espresso in secondi, è il tempo max di attesa per ACK – NAK prima della segnalazione di errore. Se = 0, il Time Out viene disabilitato. Il tempo di attesa inizia al termine della trasmissione di un comando in modalità Master e al termine dell'invio di una risposta in modalità Slave.

Questo TimeOut viene applicato solo in modalità di simulazione di protocollo, sia Master che Slave.

### Time Out for SYNC

Espresso in secondi, è il tempo max di attesa per SYNC prima della segnalazione di errore. Se = 0, il Time Out viene disabilitato. Il tempo di attesa inizia dopo l'ultimo carattere di Hand-Shake del dialogo ricevuto dal Master.

Questo Time Out è attivo solo in modalità Master di simulazione di protocollo.

### CKS used ( checksum from ID to last data byte included)

Stabilisce la presenza o meno del CKS, del pacchetto di protocollo, cioè della somma di controllo in 8 bits eseguita dal primo byte dopo STX fino all'ultimo byte di dati, inclusi gli estremi.

Nota: Il CKS può essere usato solo se il protocollo include la LEN del pacchetto e non usa ETX. Questo perché il CKS, antecedente a ETX, potrebbe assumere il valore stesso di ETX e troncatura il pacchetto di protocollo in anticipo e quindi invalidarlo.

Al variare di anche un solo dato dei campi di edit, il tasto **Apply** diventa attivo e permette di applicare le varianti fatte alla configurazione. E' consigliabile apportare prima tutte le modifiche poi metterle in uso con il tasto **Apply**.

Tutte le spunte non necessitano dell'uso del tasto Apply. Sono interpretate in modo dinamico.

La configurazione di protocollo può essere salvata 'in toto' con un nome qualsiasi, per poi essere ricaricata a piacere. I tasti **Load** e **Save** permettono la carica ed il salvataggio.

Non è necessario il tasto **Apply** quando si carica una configurazione da disco. Questa viene messa in linea al volo.

L'estensione dei files di salvataggio di config. del protocollo deve essere obbligatoriamente .PTD ( ProtocolDefinitionTable).

Quando si salva una configurazione, fornire il nome senza l'estensione e SLA penserà a mettere quella giusta.

## Protocol simulation

SLA - Serial Lines Analyzer - SIAM (c) 2005-2013 - Ver. 3.0.0

Terminal Com Setting Protocol Setting Protocol Simulation ASCII Commands

**Protocol Simulation setting**

| COMMAND     | Query Data Pack ( Data only ) | Answer Data Pack ( Data only )   | SYNC (Yes/*) |
|-------------|-------------------------------|----------------------------------|--------------|
| #000 - \$00 |                               |                                  |              |
| #001 - \$01 |                               | \$00\$00                         | Sync         |
| #002 - \$02 |                               | #100#000#250#000#020#000#030#000 | Sync         |
| #003 - \$03 | #150#000                      |                                  | Sync         |
| #004 - \$04 | #200#000                      |                                  | Sync         |
| #005 - \$05 | #100#000#150#000              |                                  | Sync         |
| #006 - \$06 |                               |                                  | Sync         |
| #007 - \$07 |                               |                                  | Sync         |
| #008 - \$08 |                               | OPTRONIK                         | Sync         |
| #009 - \$09 |                               | #006ST 1.2345E+02 3 0000#013     | Sync         |
| #010 - \$0A |                               |                                  |              |
| #011 - \$0B |                               |                                  |              |
| #012 - \$0C |                               |                                  |              |
| #013 - \$0D |                               |                                  |              |
| #014 - \$0E |                               |                                  |              |
| #015 - \$0F |                               |                                  |              |
| #016 - \$10 |                               |                                  |              |

\$xx = Hex #xxx = dec ^x = Ctrl x

Apply Load Save Clear Files \*.pst - SYN + SYN

CTS RTS DSR DCD DTR RX TX Com A Com B CLS Err

Type: RS-232C COM1 BR: 9600 Parity: None DataBits: 8 StopBits: 1

Questa tabella permette di configurare le domande e le risposte per la simulazione di protocollo. Ogni casella della tabella può essere editata con un doppio click lento ( non il doppio click veloce di Windows).

Le stringhe possono essere formate con le stesse modalita di formato di tutti gli altri campi carattere.

Le stringhe devono contenere solo i dati utili del pacchetto, senza canoni iniziali e senza comando. SLA penserà a creare il pacchetto completo di tutti i canoni in funzione del settaggio del protocollo.

La prima colonna specifica la domanda legata al comando ennesimo. (solo i dati utili, cioè DATA) I contenuti della prima colonna verranno usati da SLA quando è Master. Alla richiesta dell'operatore di trasmettere il comando xx, SLA utilizza la relativa stringa presente nella prima colonna.

**N.B.** Se il comando prevede una risposta, anche se SLA viene settato come Master, è necessario inserire la risposta nella seconda colonna della tabella per permettere al programma di capire che deve gestire l'Hand-Shake, cioè trasmettere ACK o NAK dopo aver ricevuto la risposta.

La seconda colonna specifica la risposta legata al comando ennesimo.

I contenuti della seconda colonna verranno usati da SLA quando è Slave. Al riconoscimento del comando xx ricevuto dal Master, SLA utilizza la relativa stringa presente nella seconda colonna per inviare la risposta.

La terza colonna stabilisce se il comando ha un SYNC finale oppure no. Per associare o togliere un SYNC ad un comando, selezionare una cella qualsiasi della riga del comando ed usare i tasti



E' anche possibile scrivere a mano la parola **Sync** (esattamente così) nella cella desiderata della terza colonna.

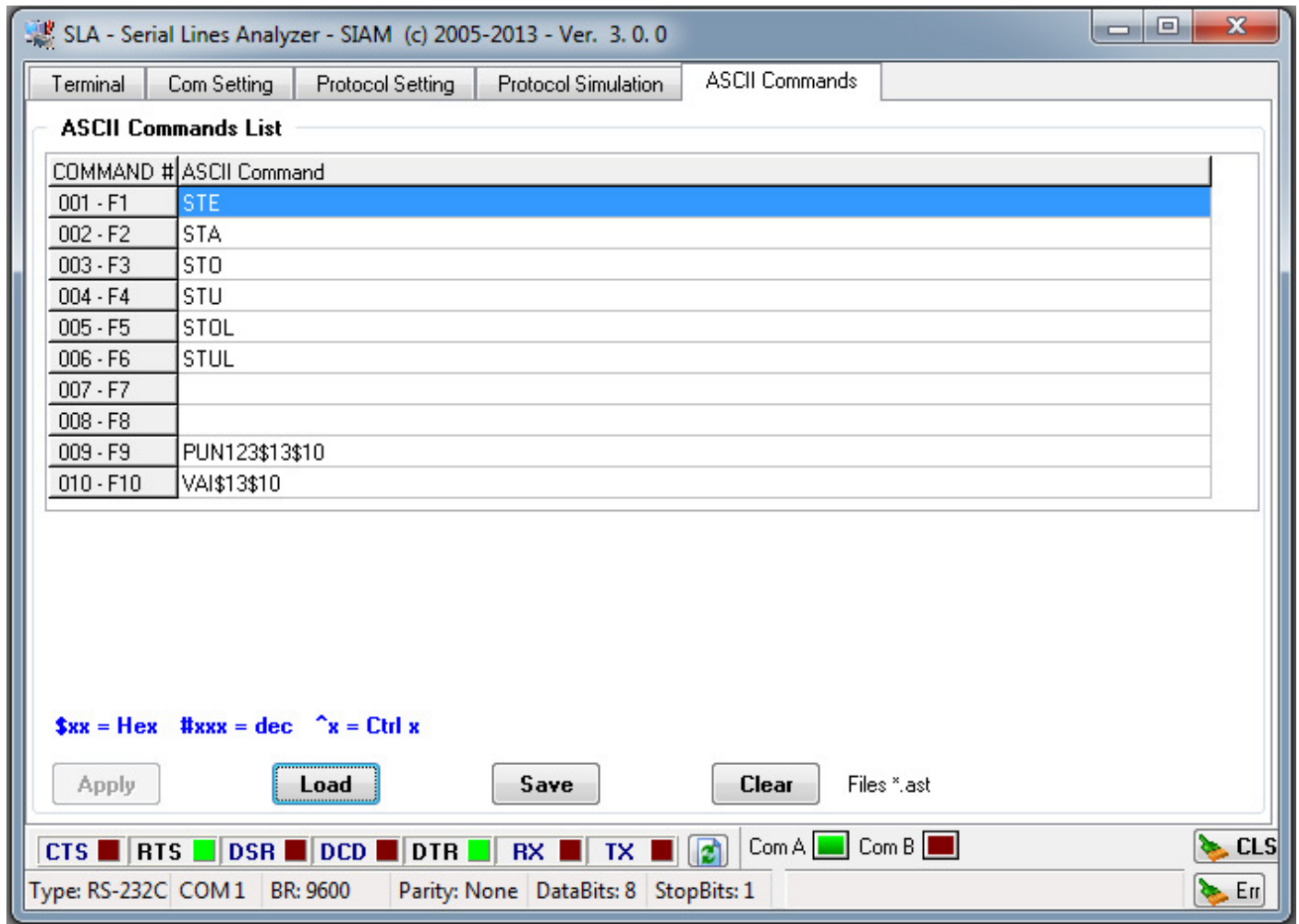
Anche qui, al variare di anche un solo dato in una cella qualsiasi, il tasto **Apply** diventa attivo e permette di applicare le varianti fatte alla tabella. E' consigliabile apportare prima tutte le modifiche poi metterle in uso con il tasto **Apply**.

La tabella di simulazione di protocollo può essere salvata 'in toto' con un nome qualsiasi, per poi essere ricaricata a piacere. I tasti **Load** e **Save** permettono la carica ed il salvataggio. Non è necessario il tasto **Apply** quando si carica una configurazione da disco. Questa viene messa in linea al volo.

L'estensione dei files di salvataggio della tabella di simulazione di protocollo deve essere obbligatoriamente .PST ( ProtocolSimulationTable).

Quando si salva una configurazione, fornire il nome senza l'estensione e SLA penserà a mettere quella giusta.

# ASCII Command

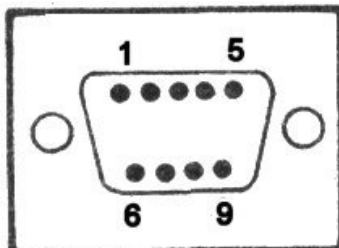


Come per la TAB Protocol Simulation, è possibile definire 10 comandi, legati poi ai tasti F1 - F10. Valgono le stesse modalità operative specificate per la tabella dei comandi di Protocol Simulation, con le relative funzioni legate ai tasti Apply, Load, Save e Clear.

## Appendice A

Piedinatura standard dei connettori seriali RS-232 presenti sui personal computers

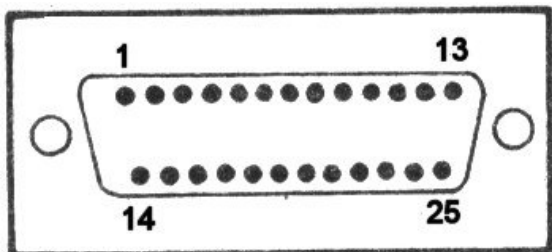
### 9 Poli maschio



Vista di fronte

|   |     |                       |
|---|-----|-----------------------|
| 1 | DCD | Data Carrier Detected |
| 2 | RXD | Received Data         |
| 3 | TXD | Transmit Data         |
| 4 | DTR | Data Terminal Ready   |
| 5 | GND | Ground                |
| 6 | DSR | Data Set Ready        |
| 7 | RTS | Request To Send       |
| 8 | CTS | Clear To Send         |
| 9 | RI  | Ring Indicator        |

### 25 Poli maschio ( solo pins usati)



Vista di fronte

|    |      |                       |
|----|------|-----------------------|
| 1  | SGND | Shield Ground         |
| 2  | TXD  | Transmit Data         |
| 3  | RXD  | Received Data         |
| 4  | RTS  | Request To Send       |
| 5  | CTS  | Clear To Send         |
| 6  | DSR  | Data Set Ready        |
| 7  | GND  | Ground                |
| 8  | DCD  | Data Carrier Detected |
| 20 | DTR  | Data Terminal Ready   |
| 22 | RI   | Ring Indicator        |