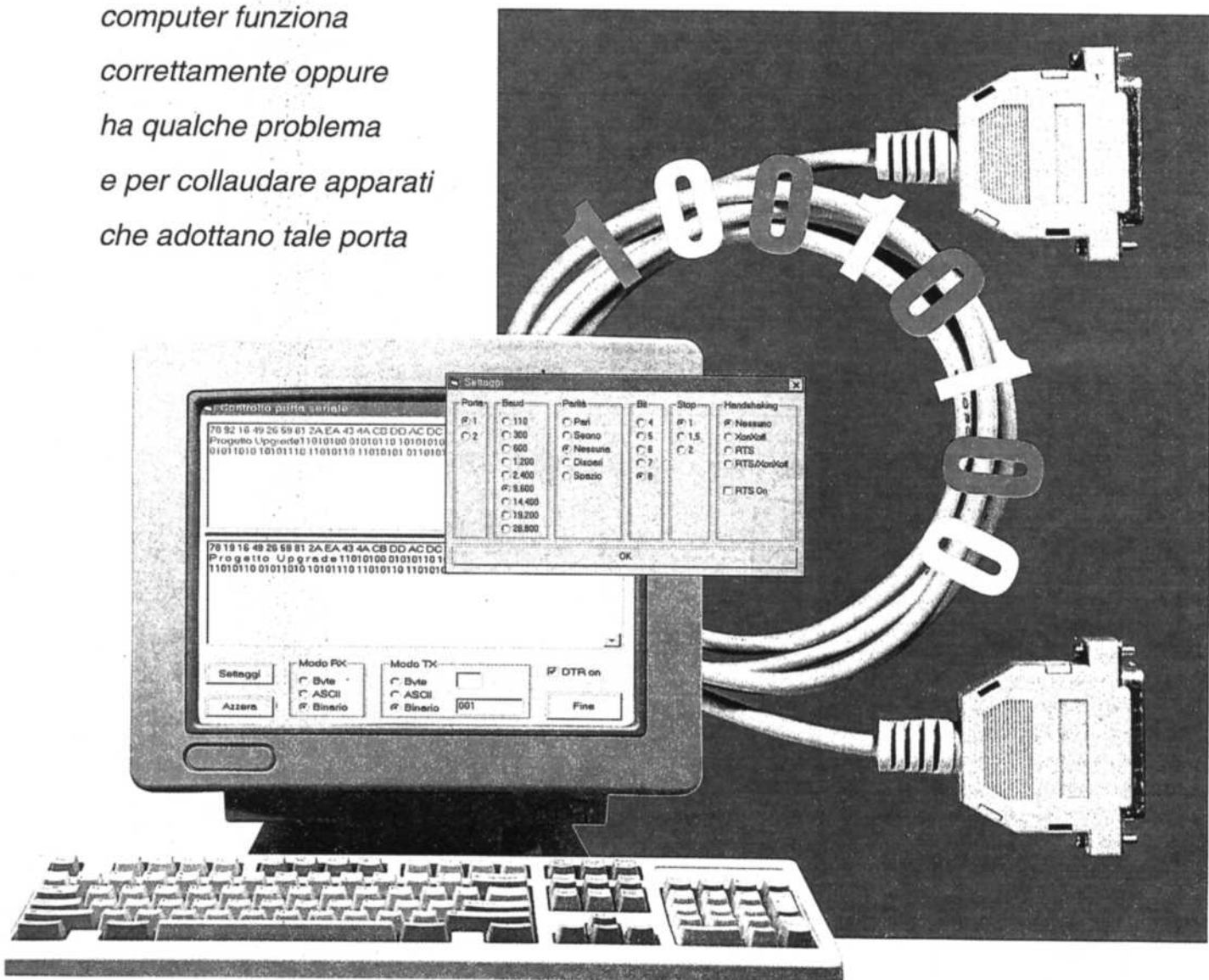


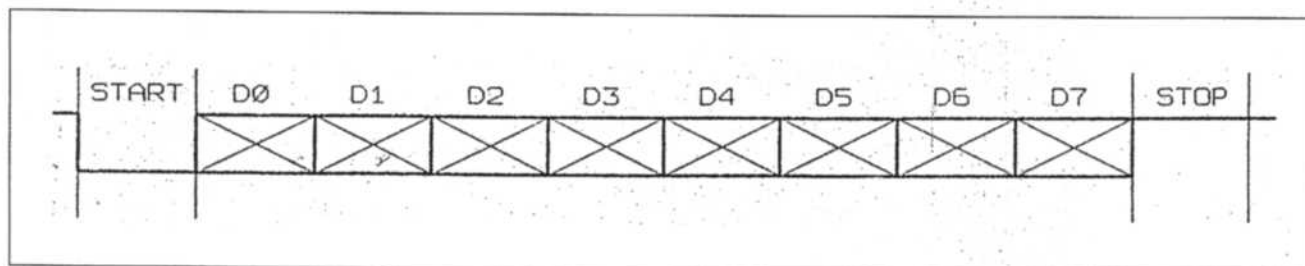


# Dialogare con la seriale RS 232

di Simone Argentini

*Viene proposto un valido strumento per stabilire se la porta RS232 del proprio computer funziona correttamente oppure ha qualche problema e per collaudare apparati che adottano tale porta*





**A**nche questo mese proponiamo un valido aiuto a coloro che non si intendono troppo di elettronica ma che lavorano con il computer. Il motivo per cui viene trattata spesso la porta di comunicazione seriale di un computer è assai banale: ad oggi non ci sono tanti altri sistemi altrettanto validi e sicuri per poter dialogare con un computer in modo estremamente semplice ed affidabile.

Se ci pensiamo bene infatti, un computer entry-level (ovvero con la minima configurazione standard proposta dal commerciante) di oggi, dispone generalmente di una porta parallela e di due porte seriali. Sono disponibili ovviamente altri sistemi di comunicazione, ma non così comuni e facili da maneggiare come i due accennati: i portatili per esempio, hanno a bordo sempre più frequentemente il modulino che consente loro di "parlare" con i raggi infrarossi (come anche alcune stampanti) mentre non di rado si inseriscono nei computer le ormai conosciutissime schede di rete.

Ma nella pratica quotidiana, se acquistiamo una periferica magari non propriamente attinente al mondo dei computer, sicuramente avrà una delle due porte sopra menzionate e cioè o la parallela o la seriale.

La seriale in particolar modo viene preferita perché è possibile gestirla per mezzo di soli tre fili (nel caso migliore)

contro gli 8-10 della parallela e quindi permette l'impiego di connettori molto piccoli che trovano posto in tutte quelle apparecchiature portatili oggi tanto di moda.

Se ci guardiamo intorno, troviamo una miriade di data-bank e di calcolatrici programmabili che sicuramente si interfacciano con i computer via porta seriale, oppure abbiamo i classici telefonini cellulari che da tempo non si smontano più per la taratura e la messa a punto, ma vengono collegati alla seriale del computer dove ovviamente è caricato un apposito programma dato dalla casa produttrice al centro di assistenza.

I televisori più evoluti hanno un collegamento seriale per l'aggiornamento del firmware interno, come pure i più moderni ricetrasmittitori, anche

**Figura 1.**  
**Esempio**  
**di frame**  
**asincrono**

se questo non viene individuato tanto facilmente per l'assenza di connettori esterni.

Nel mondo dell'elettronica poi, la porta seriale rappresenta, unitamente alla porta parallela, l'unico mezzo di interfacciamento rapido per qualsiasi periferica, sia autocostruita che acquistata.

Tutti i programmatori di chip, di memorie, di gal, di Eprom e così via sono generalmente connessi alla RS232, come pure la maggior parte dei più moderni strumenti.

Se sfogliamo cataloghi di strumentazione elettronica infatti, troviamo che adesso anche il comune multimetro dialoga con il computer attraverso la porta seriale. Ma anche chi realizza autonomamente un piccolo circuito che poi dovrà essere connesso al computer lo

**Figura 2.**  
**Rapporto**  
**tempo di**  
**transizione /**  
**velocità**

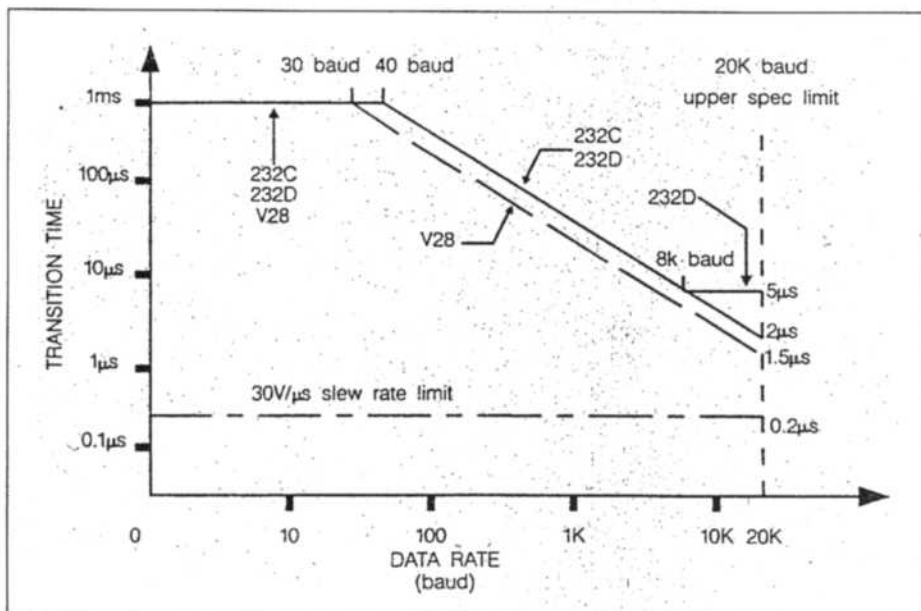


Tabella 1. Confronto tra standard seriali asincroni

Parameter	EIA-232	RS-423-A	RS-422-A	RS-485	
Mode of Operation	Single-Ended	Single-Ended	Differential	Differential	
Number of Drivers and Receivers	1 Driver 1 Receiver	1 Driver 10 Receivers	1 Driver 10 Receivers	32 Drivers 32 Receivers	
Maximum Cable Length (m)	15	1200	1200	1200	
Maximum Data Rate (baud)	20 k	100 k	10 M	10 M	
Maximum Common-Mode	±25	±6	6 to -0.25	12 to -7	
Voltage (V)					
Driver Output Levels (V)	Unloaded Loaded	±5 ±15	±3.6 ±6	±2 ±5	±1.5 ±5
Driver Load (Ω)	3 k to 7 k	450 (Min)	100 (Min)	60 (Min)	
Driver Slew Rate	30 V/μs (Max)	External Control	NA	NA	
Driver Output Short Circuit	500 to Vcc	150 to GND	150 to GND	150 to GND	
Current Limit (mA)				250 to -7 or 12 V	
Driver Output Resistance-High Z	Power On	NA	NA	NA	12 k 12 k
State (Ω)	Power Off	300	60 k	60 k	
Receiver Input Resistance (Ω)	3 to 7	4	4	12	
Receiver Sensitivity	±3 V	±200 mV	±200 mV	±200 mV	

predispone o per la porta seriale o per la porta parallela, per la semplicità di sviluppo anche del software: molti linguaggi di programmazione offrono la possibilità di dialogare con queste due porte per mezzo di semplici istruzioni.

Non molto tempo fa per esempio, abbiamo proposto una scheda di acquisizione che dialogava con il computer via porta seriale.

Per funzionare aveva bisogno di un comando che gli giungeva sotto forma di byte sulla linea seriale e poi rispondeva al computer con una serie di uno o più byte in funzione del comando impartito. Per comunicare con quella scheda era quindi necessario disporre di un software che inviava facilmente sulla linea seriale dei byte e che li visualizzasse in

modo molto chiaro sullo schermo del computer.

La soluzione fu trovata per il sistema operativo MS-DOS con l'adozione di un software di controllo della porta seriale che consentiva varie modalità di impiego.

Purtroppo o per fortuna (dipende dai punti di vista!) il sistema operativo MS-DOS non è oggi quasi più utilizzato, soprattutto con l'arrivo di Windows95 e Windows98 che propongono FAT (File Allocation Table) a 32 bit contro i 16 precedenti.

Per risolvere egregiamente questo ostacolo, abbiamo allora preparato un software simile come gestione al precedente, ma arricchito con altre funzioni molto interessanti e che necessita obbligatoriamente del sistema operativo Windows95

(oppure Windows98), in modo da interagire con gli altri programmi che si trovano comunemente sul mercato.

## La rappresentazione dei dati

Quali sono le esigenze principali di chi desidera sfruttare la porta seriale in modo semplice e senza doversi preoccupare di scrivere in complicati linguaggi?

La prima è senz'altro quella di poter facilmente impostare i parametri di dialogo e la seconda è quella di poter inviare e ricevere i dati nella rappresentazione desiderata: chi lavora con periferiche homemade per esempio, sa che cosa significa poter inviare lo stesso dato in varie rappresentazioni.

Ma che cosa si intende per "rappresentazione di un dato"? Supponiamo di voler testare una periferica che in base al comando 0xA5 risponde con un byte (ad esempio la lettura di un convertitore A/D ad 8 bit). Allora avremo sicuramente vita facile se potessimo inviare il byte direttamente in esadecimale, ovvero scrivendo solo i due caratteri A5 sulla tastiera ed allo stesso tempo se potessimo visualizzare sempre in esadecimale il carattere di risposta.

Supponiamo invece di possedere una scheda che ci converte il dato in ingresso nella corrispondente attivazione di 8 linee parallele ed allo stesso tempo che ci rinvia lo stato di altre otto linee.

In questo caso, per poter attivare per esempio le linee 0, 3 e 5, si dovrà inviare il byte 0x49 ma sarebbe più semplice poter inviare direttamente la combinazione binaria ovvero

00101001 dove gli "1" corrispondono alle linee da attivare. Allo stesso modo, per vedere quali delle otto linee sono attive, sarebbe più comodo avere la visualizzazione del tipo 00101011 invece del corrispondente numero esadecimale 0x4B.

Infine, potrebbe anche essere richiesto un dialogo tramite caratteri ASCII (American Standard Code for Information Interchange) ovvero le lettere che si vedono direttamente sul video. Abbiamo visto indirettamente che cosa si intende per rappresentazione: il dato che viene trattato (o inviato) ha sempre lo stesso valore numerico, ma viene visualizzato o inserito in una forma (rappresentazione) diversa dal valore decimale.

Nel primo esempio analizzato abbiamo infatti il numero decimale 165 che equivale al numero 0xA5 che equivale al numero 10100101.

Inviare allora una delle tre rappresentazioni, equivale in ogni caso ad inviare sempre lo stesso dato sulla porta seriale, ma per l'utilizzatore può essere più comodo impiegare una rappresentazione piuttosto che un'altra in funzione del lavoro che deve eseguire.

### La nostra proposta

Il programma che proponiamo in queste pagine, è stato realizzato espressamente per assolvere alle esigenze sopra specificate. Nella schermata 1, è visibile la pagina principale

dell'applicativo. In alto è presente una casella di testo che visualizza i caratteri trasmessi dal computer e più in basso è presente la casella di testo che visualizza i caratteri ricevuti dal computer. Entrambe queste caselle hanno lo scorrimento verticale, ovvero permettono comunque la ricezione o trasmissione dei dati inserendo un cursore per il controllo della porzione da visualizzare.

Nella parte inferiore sinistra sono presenti due pulsanti: Settaggi e Azzerà.

Con il primo si entra nella pagina che consente di impostare i valori della porta seriale, con il secondo si puliscono le due finestre in alto.

Ci sono poi due blocchi decisionali, uno per la ricezione ed uno per la trasmissione.

Generatore di forme d'onda arbitrario Analogiche e/o Digitali DA 8 bit.  
Sample Rate fino a 20,480 Ms/s, 2 canali separati,  
4 forme d'onda in memoria. 2048 byte per canale  
(8192 byte totali). Editor Digitale e Analogico  
con funzioni matematiche. Importazione di  
file da Oscilloscopio ETC M 221. massima  
frequenza Analogica di 2,56 MHz.  
Tensione programmabile fino a 10 Vpp.  
Regolazione di fase da 0 a 360 gradi  
4 modalità di Trigger. Connessione  
su bus ISA. Software Windows 3.1  
e WIN 95. Prezzo estremamente  
contenuto.

**M321**

ETC

**SONDE** — 1:1 - 1:10  
1:100  
differenziali SI 9000

**SENSORI** — Di Temperatura  
Di Pressione  
Di Umidità

**PINZE** — Capacitive per alta tensione  
Induttive per alta tensione  
Amperometriche

**MICROSCOPE**

TIEPIE

A/D 8 bit, un canale bipolare.  
Collegamento su porta parallela.  
Non richiede alimentazione.  
Campionamento 100 KS/s.  
Accoppiamento DC, da 2,5 a 20 volts AC/DC.  
Oscilloscopio, Voltmetro, Analizzatore di Spettro,  
Registratore di Transitori, Software DOS.  
Personalizzazioni Software su richiesta per quantità.  
L. 219.000

**ARTEK**

**ELECTRONIC SOLUTIONS**

**BASIC BUG**  
PARALLAX

Il kit contiene:  
1 Micro Modulo BS1, cavo di programmazione  
Manuale descrizione assemblaggio + codice sorgente BS1  
Led, resistenze, condensatori, switch + antenne  
zampette, contenitore per batterie, 3 muscoli (servo motori)  
Una realizzazione didattica, simpatica e creativa

**AMPIA GAMMA DI SCHEDE**  
- PER ACQUISIZIONE DATI  
- PER CONTROLLO ASSI

**TDA3**  
ACQUADATA

Acquisizione dati, AD 12 bit, 3 I/O digitali, 2 ingressi analogici da 0 a 4,096 V, 3 ingressi differenziali da -5 a 45 mV, 1 sensore di temperatura ambiente incluso  
Connessione su LPT, non richiede alimentazione  
Plot grafico, 6 Voltmetri, Auto Log su disco, DDE, Software Windows, esempi e sorgenti in vari linguaggi

**TP508**  
TIEPIE

AD 8 Bit- 2 canali separati.  
Connessione su bus ISA.  
Accoppiamento AC-DC via software.  
Memoria di 32 Kbyte per ciascun canale.  
Ingressi da  $\pm 20$  mV a  $\pm 80$  V su intera scala.  
Oscilloscopio, Voltmetro, Analizzatore di spettro.  
campionamento a 50 Ms/s su un canale 25 Ms/s su 2 canali.  
Registratore di transitori, Software Windows e Dos. DLL e routine sorgenti incluse.

**Electron**  
**ARTEK**

Con il primo si seleziona la rappresentazione dei dati in ricezione: si hanno a disposizione la rappresentazione a byte, in ASCII e binaria. In pratica, a seconda della rappresentazione scelta, nella finestra relativa alla ricezione appariranno i dati ricevuti sulla porta del computer. Con il secondo gruppo decisionale, si seleziona la modalità di trasmissione: a byte, in ASCII o in modo binario. Per la modalità byte, è presente a fianco una casella di testo che consente l'immissione di massimo due caratteri esadecimali (da 0 a 9 e da A a F).

All'arrivo del secondo carattere, il byte viene automaticamente trasmesso e la casella azzerata.

Il byte trasmesso viene inoltre visualizzato nella relativa finestra superiore.

Per la modalità binaria, si ha un funzionamento simile: è presente una casella di testo in cui è possibile inserire soltanto 0 e 1. Al raggiungimento dell'ottava cifra, tutto il byte viene trasmesso e visualizzato nella finestra superiore.

Infine, per la modalità ASCII, il testo viene inserito e subito trasmesso direttamente nella finestra di trasmissione superiore.

In basso a destra è presente un pulsante per l'abilitazione o meno del segnale DTR.

Per coloro che hanno dimestichezza con la RS485, è facile intuire a che cosa serva: per dialogare con il bus RS485, lo standard impiegato dai software per computer per invertire la direzione dei dati sul bus è costituito dall'abilitazione o meno del segnale DTR. Con l'opzione inserita, sul pin DTR della porta arriverà un segnale positivo, viceversa si otterrà un segnale negativo.

Questo metodo può essere utilizzato anche per portare o togliere alimentazione ad apparecchiature che prelevano la loro alimentazione direttamente dalla porta seriale.

### I settaggi

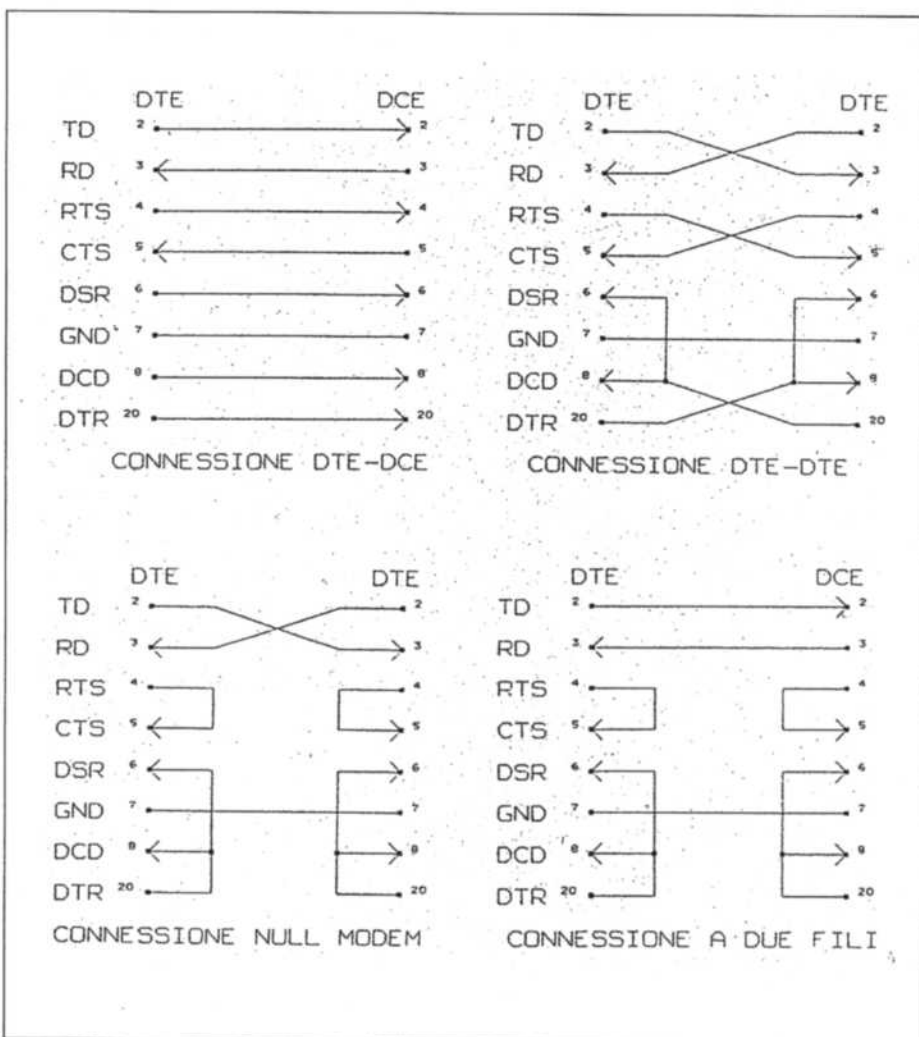
Veniamo quindi a vedere la schermata 2 cui si arriva premendo il pulsante "Settaggi". La prima impostazione è relativa al numero di porta che può essere la 1 o la 2. Se scegliete la porta "sbagliata", ad esempio perché già aperta dal mouse, il programma uscirà con un errore. La seconda impostazione riguarda la velocità di dialogo:

si va da un minimo di 110 baud ad un massimo di 28.800. La terza impostazione controlla la parità che può essere pari, dispari, nessuna, di segno o di spazio. La quarta casella consente la scelta del numero di bit per carattere trasmesso: da un minimo di 4 ad un massimo di 8.

La quinta scelta setta il numero di bit di stop, che potrà essere 1, 1,5 o 2.

Infine, l'ultima casella imposta il tipo di handshaking, che potrà essere di tipo XON/XOFF oppure RTS oppure la combinazione di entrambi, oppure nessuno. In aggiunta è disponibile il pulsante per la selezione dell'RTS.

Figura 3. Connessioni DTE-DCE



## Uno sguardo alla comunicazione seriale asincrona

Prima di andare a collaudare il programma, cerchiamo di spiegare a grandi linee come funziona una porta seriale asincrona.

Il nome stesso ci dice che i dati scorrono in modo seriale, ovvero uno dopo l'altro sullo stesso filo, e senza collegamenti a segnali di clock esterni.

Lo standard seriale asincrono prevede quindi un certo formato del dato da inviare, con dei parametri che devono essere identici per il trasmettente ed il ricevente.

Per trasferire un byte, cioè otto bit, si deve per prima cosa inviare un bit detto di START.

La linea seriale per definizione sta sempre a livello logico 1.

Il bit di start è per forza di cose un bit che vale 0, in modo tale da far vedere al ricevente una variazione di stato logico. In figura 1 si può vedere un esempio di trasmissione asincrona.

Poi si devono inviare gli otto bit di DATO. Questo può avvenire con modalità MSB-first oppure con modalità LSB-first. La prima significa che i bit vengono inviati a partire dal più significativo (l'ottavo), la seconda che vengono inviati a partire dal meno significativo (il primo). Dopo i bit di dato, è possibile inserire un bit detto di PARITA'.

Diciamo che è possibile perché non è obbligatorio, ma a discrezione del softwareista. Ovviamente con il bit di parità si può capire se ci sono stati degli errori o meno e recuperare così dati ambigui.

Tabella 2. Connessioni DB9 e DB25 standard

DB 25	DB 9	Segnale	Funzione	DTE	DCE
1		CG	Chassis ground		
2	3	TD	Trasmissione dati	out	in
3	2	RD	Ricezione dati	in	out
4	7	RTS	Request To Send	out	in
5	8	CTS	Clear To Send	in	out
6	6	DSR	Data Set Ready	in	out
7	5	SG	Signal Ground		
8	1	DCD	Data Carrier Detect	in	out
9			Positive test voltage		
10			Negative test voltage		
11			Non utilizzato		
12		SDCD	Secondary DCD	in	out
13		SCTS	Secondary CTS	in	out
14		STD	Secondary TD	out	in
15		TC	Transmit Check	in	out
16		SRD	Secondary RD	in	out
17		RC	Receive Clock	in	out
18			Non utilizzato		
19		SRTS	Secondary RTS	out	in
20	4	DTR	Data Terminal Ready	out	in
21		SQ	Signal Quality Detect	in	out
22	9	RI	Ring Indicator	in	out
23		SEL	Speed selector DTE	in	out
24		TCK	Speed selector DCE	out	in
25		BSY	Data line busy	in	out

Il bit di parità viene calcolato in vari modi, ma l'importante è che i due dialoganti lo calcolino nello stesso modo!

Infine viene trasmesso un bit detto di STOP, che per definizione vale sempre un 1 logico. È molto importante che i due sistemi dialoghino con gli stessi settaggi, ed in particolar modo con la stessa velocità di comunicazione.

In mancanza di un clock comune infatti, il solo sistema che due periferiche hanno di parlare è quello di riferirsi ad un clock interno ma di identica frequenza.

Per esempio, supponiamo che venga trasmesso un byte alla velocità di 9.600 baud. Questo significa che la durata di un bit è per definizione di  $1/9.600 = 416$  microsecondi.

Allora, il ricevente, dopo l'arrivo dello start bit, deve attendere un tempo pari a 416 microsecondi per essere sicuro di bypassare lo start bit, poi deve attendere il tempo corrispondente a  $416/2 = 208$  microsecondi per essere sicuro di andare a testare il primo bit esattamente al centro, ed infine deve attendere 416 microsecondi per ogni bit successivo.

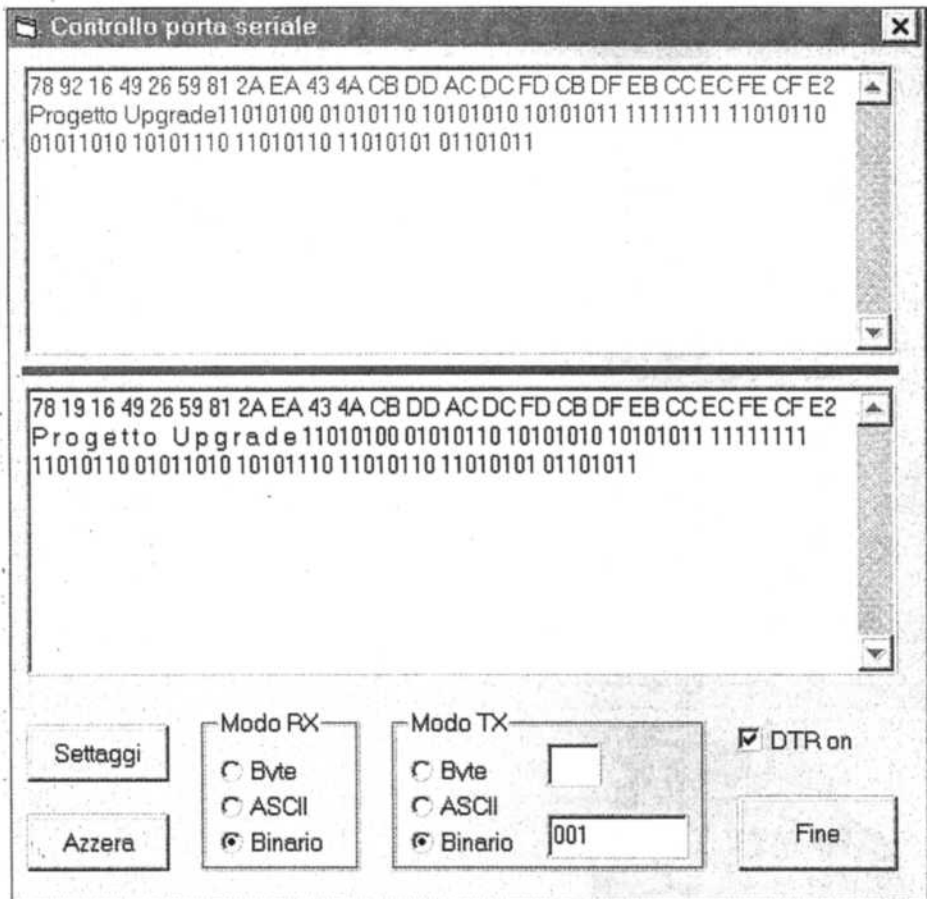
Si comprende che se la velocità di trasmissione non coincide perfettamente con quella di ricezione, si possono avere dei dialoghi con errori di ricezione.

Spesso, per limitare questi errori vengono anche impiegati dei segnali di controllo detti di HANDSHAKING che vedremo tra poco. In figura 2 è possibile vedere la curva del rapporto tra il tempo di transizione di un segnale (passaggio da 0 a 1 logico e viceversa) e la velocità in baud: si vede ovviamente come il primo debba necessariamente diminuire con l'aumentare della velocità.

In tabella 1 si hanno a disposizione i parametri degli standard seriali più utilizzati mentre in tabella 2 vediamo i segnali relativi alla RS232 ed ai due connettori standard per computer ovvero il DB9 ed il DB25.

Il segnale TX è in uscita dal DTE (Data Terminal Equipment) verso il DCE (Data Control Equipment) mentre l'RX è in ingresso. Su questi due fili passano i dati del dialogo.

Il segnale RTS (Request To Send) viene sfruttato dal DTE per dire al DCE che è pronto a trasmettere.



*Pagina principale del programma*

*Pagina dei settaggi*

Il segnale CTS (Clear To Send) è la risposta del DCE al DTE che è pronto a ricevere.

Il DCR (DCe Ready) detto anche DSR (Data Set Ready) è un altro segnale che va dal DCE al DTE per informarlo che esiste la comunicazione tra i

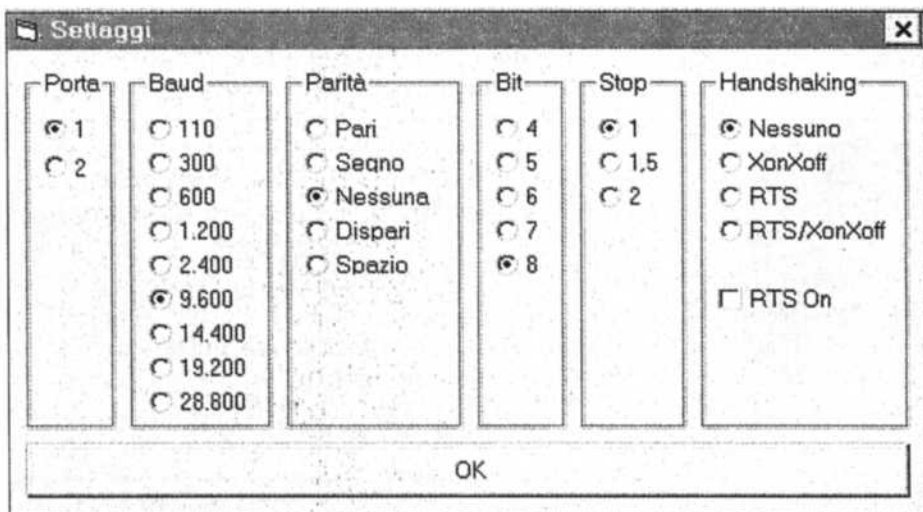
due. Con il segnale DTR (Data Terminal Ready) il DTE informa il DCE che è pronto per dialogare. Spesso viene impiegato anche per il controllo della direzione nei sistemi half-duplex (come nel caso dello standard RS485).

Con il DCD (Data Carrier Detect) il DCE informa il DTE che sta ricevendo un segnale esterno valido.

Il segnale RI (Ring Indicator) infine è un segnale che il DCE invia al DTE per avvisare di una chiamata esterna.

Viene usato in genere per il rilevamento della chiamata telefonica.

Per capire come connettere tra di loro le varie periferiche, in figura 3 abbiamo riportato le 4 connessioni principali: DTE-DCE, DTE-DTE, NULL-MODEM e A DUE FILI.



## Il collaudo

Per entrare in possesso del software di controllo della porta seriale, potete telefonare al numero 0338/4890048.

Per il collaudo la cosa può essere fatta in modo banale predisponendo un connettore DB25 (o DB9) in modo tale che abbia il pin 2 connesso con il pin 3.

A questo punto si potrà far partire il programma. La prima schermata non implica l'apertura della porta, ovvero la porta verrà aperta solo dopo l'apertura della seconda schermata per mezzo del pulsante "Settaggi".

Se la porta che si tenta di aprire è già impegnata, il software esce con un avviso di errore, altrimenti la seconda schermata sparirà per far posto alla prima.

Adesso si potrà scegliere la modalità di trasmissione e quella di ricezione, dato che sono indipendenti: per esempio è possibile inviare dati in modo byte e riceverli in modo binario o viceversa. Con il controllo DTRon si decide se attivare o meno l'uscita DTR, ad esempio per fare delle prove con un bus RS485.

Se in casa si hanno a disposizione due computer è possibile inserire il software su entrambi e connetterli via porta seriale con i pin 2 e 3 invertiti (TX con RX e viceversa).

In questo modo tutto ciò che verrà inviato dal primo sarà visibile sull'altro e sarà così possibile implementare un sistema di comunicazione a bassissimo costo. Si ricorda che per utilizzare questo programma, è indispensabile che sul computer sia installato il sistema operativo Windows 95 oppure Windows 98. ■

# DOVE?

## NEI NEGOZI SPECIALIZZATI



## DI BIASE LEONARDO

**GBC**

COMPONENTI ELETTRONICI - RADIO  
TV COLOR - AUTORADIO - HI/FI - TELEFONIA  
ANTENNISTICA - VIDEOREGISTRAZIONE  
SISTEMI RICEZIONE SATELLITE

P.zza Kennedy, 3 - **FASANO**  
Tel. 080/713337 - Fax 080/713615

## NUOVA HALET ELECTRONICS S.r.l.

**GBC**

COMPONENTI ELETTRONICI - RADIO  
TV COLOR - AUTORADIO - HI/FI - TELEFONIA  
ANTENNISTICA - VIDEOREGISTRAZIONE  
SISTEMI RICEZIONE SATELLITE

Via Ex. Capruzzi, 192 - **BARI**  
Tel. 080/5428822 - Fax 080/5428319

## VIDEOCOMPONENTI S.r.l.

**GBC**

COMPONENTI ELETTRONICI - RADIO  
TV COLOR - AUTORADIO - HI/FI - TELEFONIA  
ANTENNISTICA - VIDEOREGISTRAZIONE  
SISTEMI RICEZIONE SATELLITE

V.le P. Togliatti, 22/32 - **BRINDISI**  
Tel./Fax 0831/542994

**GBC**

## DBL ELETTRONICA S.r.l.

COMPONENTI ELETTRONICI - RADIO - TV COLOR  
AUTORADIO - HI/FI - TELEFONIA  
ANTENNISTICA - VIDEOREGISTRAZIONE  
SISTEMI RICEZIONE SATELLITE

Via Lupiae, 14/14a - **73100 LECCE** - Tel./Fax 0832/318641