

PC 128S : ALLA SCOPERTA DELLE COMUNICAZIONI CON LA RS 232

Per entrare nel mondo della trasmissione dati, l'utilizzo della porta RS232 è un passaggio obbligato: questo articolo è il primo di una mini monografia in due puntate, dedicata alla RS232 del PC128S, in cui cercheremo di illustrare in maniera chiara quei concetti fondamentali per la realizzazione di un efficiente collegamento seriale.

Trasmissione seriale asincrona

Un sistema molto vantaggioso per la trasmissione di dati, specie sulle lunghe distanze, è quello della trasmissione seriale asincrona: seriale, perchè i vari bit che compongono il dato da trasmettere, scorrono "in fila indiana" su di un'unica linea (e non contemporaneamente, ognuno sul proprio conduttore, come avviene nei collegamenti "paralleli"); asincrona, perchè trasmettente e ricevente non sono vincolati ad essere sincronizzati con una linea di "clock", ma si regolano automaticamente tramite appositi "bit di start" e "bit di stop".

Ovviamente trasmettente e ricevente devono essere d'accordo sulla velocità di trasmissione (che si misura in "Baud": bit al secondo), e sul protocollo, ovvero sulle regole secondo le quali inserire bit di start, di stop, di controllo. Bisogna fissare delle norme da seguire, per evitare il caos al momento di connettere tra loro apparecchiature differenti. A questo scopo esistono ben precisi standard a cui uniformarsi, uno di questi è la raccomandazione CCITT V.24, meglio nota come EIA RS 232 C.

Lo standard RS 232 C

Questo standard è divenuto di gran lunga quello più diffuso per l'interfacciamento di micro e personal computer con i più svariati tipi di unità periferiche, come

stampanti, plotter, modem, altri computer... Lo standard RS232 prevede che i bit vengano codificati come segnali elettrici (riferiti rispetto ad una linea di "massa" a potenziale zero), in modo che ad un "1" corrisponda un livello di tensione compreso tra -3V e -15V, mentre per uno "0" si va da +3V a +15V, il tutto su cavi lunghi al massimo una trentina di metri (ma è meglio non esagerare in lunghezza, per non catturare una valanga di disturbi).

Il protocollo prevede che il "pacchetto" di bit da trasmettere sia così composto: un bit di start, gli otto (oppure sette) bit del dato da inviare, un eventuale bit di parità, infine uno oppure due bit di stop. Il bit di parità, se si sceglie di farne uso, serve per effettuare un veloce controllo sulla esattezza dei dati ricevuti. Se si sceglie la "parità pari", questo bit viene settato da un circuito apposito del trasmettitore, in modo che il numero di cifre "1" nella parola trasmessa sia pari; il dispositivo ricevente a sua volta effettuerà un controllo sul numero di "1", e se non lo troverà pari interpreterà questo fatto come un errore di trasmissione. Analogamente funziona il controllo per la "parità dispari"; questi controlli non sono sicuri al cento per cento, ma nel caso di una linea mediamente disturbata, riescono ad individuare buona parte degli errori.

Il collegamento: hardware

Lo standard RS 232 prevede oltre alle linee di trasmissione e ricezione dati, tutta una serie di collegamenti ausiliari, non meno importanti, come ad esempio le linee di "handshake" (letteralmente "stretta di mano": con appositi segnali trasmettente e ricevente si informano a vicenda di essere pronti a comunicare); sempre stando allo standard, si dovrebbe adoperare per il collegamento un connettore a va-

schetta "MINI D" a 25 poli (ad es. Cannon 7592). In realtà pochi costruttori seguono a pieno queste regole, e nella maggior parte dei casi si limitano ad adottare solo un insieme ridotto di segnali, per non parlare poi dei connettori, con i quali se ne vedono veramente di tutti i colori. Realizzare un collegamento RS 232 può dunque spesso divenire una operazione complicata, in quanto molti sono i fattori che possono ingenerare confusione. Tanto per cominciare, le apparecchiature RS232 possono essere di due tipi: DTE e DCE, e fra queste i collegamenti sul connettore sono esattamente opposti: quella linea che per il DTE è una uscita, sarà una entrata per il DCE. Perchè tutto ciò? Inizialmente la RS232 era stata concepita per la trasmissione di dati su lunga distanza tramite modem (vedi figura 1); il DTE, Data Terminal Equipment, ovvero il terminale del computer, trasmette i dati alla unità di comunicazione: il DCE, Data Communication Equipment (tipicamente un modem); questo si occupa della trasmissione a distanza, magari con un ponte radio, od un canale telefonico. Dall'altra parte c'è un altro DCE che riceve i dati e li affida infine all'altro DTE: il terminale ricevente. Da notare che la trasmissione può avvenire in entrambi i sensi, o come si usa dire, in "duplex".

Volendo collegare direttamente tra di loro i due computer (DTE) bisognerà invertire tutti i collegamenti, in quanto una porta DTE può collegarsi "pin to pin" solo con una DCE e viceversa. La situazione è però ancora più caotica, in quanto è possibile trovare dei computer la cui porta RS232 non è configurata come DTE, ma come DCE; in pratica il collegamento dovrà essere studiato caso per caso.

I segnali disponibili

Nella figura 2 sono riassunti i segnali

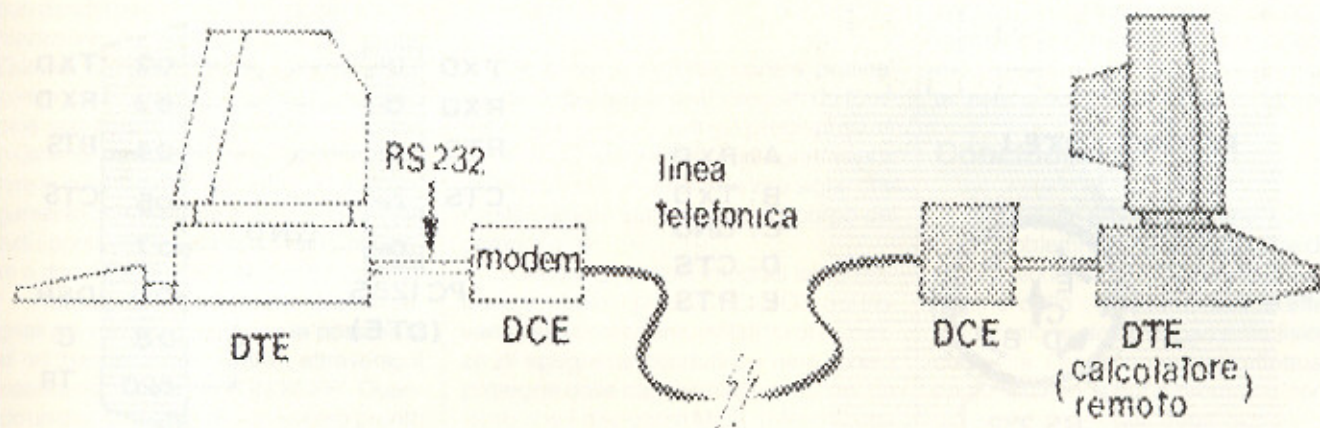


Fig. 1: Il colloquio RS 232 avviene in genere tra DTE e DCE

più comunemente usati per i collegamenti RS 232; premettendo che si tratta solo di un sottoinsieme dello standard vero e proprio, vediamo il significato.

GND (ground): è la "massa segnale", ovvero la linea di riferimento elettrico comune ad un potenziale di zero volt (da non confondersi con la massa telaio, presente solo in alcuni casi, ed usata per motivi di schermatura).

TXD (Transmitted Data): è la linea dalla quale DTE trasmette i dati (e dunque sulla quale li riceverà il DCE), sarà quindi una uscita per il DTE ma una entrata per il DCE.

RXD (Received Data): su questa linea il DTE riceve i dati provenienti dal DCE (sarà dunque un ingresso per DTE, ma una uscita per DCE; il meccanismo dovrebbe ora essere chiaro).

RTS (Request To Send): attivando questa linea, DTE dice a DCE "ti chiedo di trasmettere".

CTS (Clear To Send): con questa linea invece DCE risponde a DTE "Sono pronto a trasmettere", può dunque aver luogo lo scambio dei dati; si noti come con i due segnali RTS/CTS si possa realizzare il già accennato "handshake".

DSR (Data Set Ready): quando questa linea è attiva, DCE segnala a DTE di essere pronto alla trasmissione.

DTR (Data Terminal Ready): qui è invece il DTE che segnala a DCE d'esser pronto a ricevere. Anche la coppia DTR/DSR può essere usata per l'handshake (ma questa non è una regola: possono andar bene anche RTS/DTR).

DCD (Data Carrier Detected): questa linea, un po' meno usata, serve al DCE per segnalare al DTE che si stanno ricevendo dei segnali, risulta cioè attivo il collegamento sulla linea telefonica od attraverso il ponte radio (ciò ovviamente ha significato solo per un modem od un TNC).

Tutti i segnali descritti nella figura 2, si riferiscono ad un connettore a 25 poli (identificare il contatto non è difficile: accanto ad ogni terminale è stampigliato il suo numero!), in quanto è quello maggiormente usato per i collegamenti RS232; sul

Figura 2: i segnali principali della interfaccia RS 232, riferiti ad un connettore "MINI D" da 25 poli.

contatto n.	segnale	DTE	DCE
1	massa telaio		
7	GND		
2	TXD	out	in
3	RXD	in	out
4	RTS	out	in
5	CTS	in	out
6	DSR	in	out
20	DTR	out	in
8	DCD	in	out

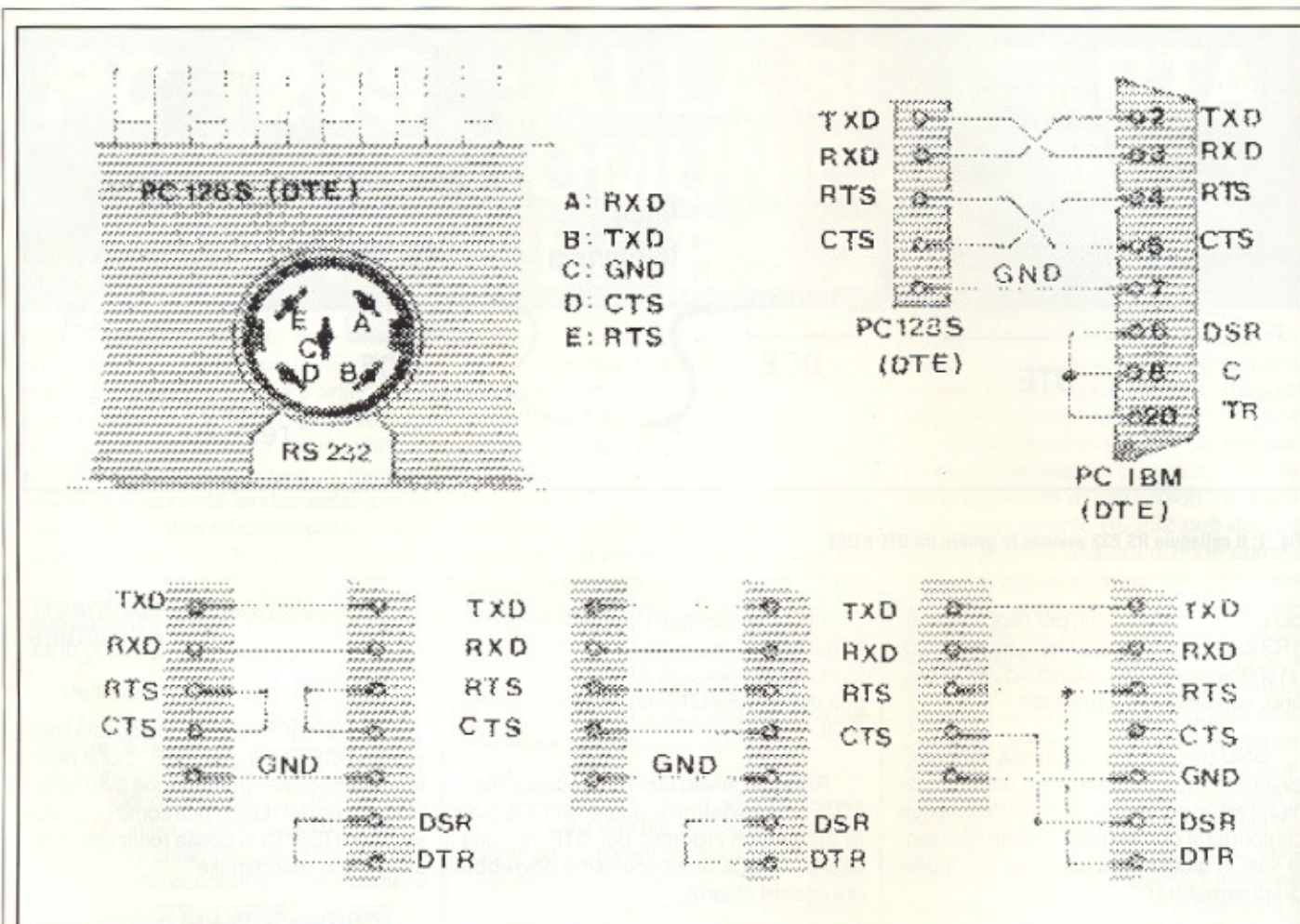


Fig. 3: La porta seriale del PC 128S con alcuni esempi di collegamento

PC128S invece il connettore è differente, e sono presenti solo alcuni dei segnali; bisognerà dunque armarsi di pazienza e saldatore, per realizzare un collegamento su misura. Per fare ciò bisogna tenere sempre sotto controllo la tabella di fig.2, per non confondere ingressi con uscite, ed identificare correttamente i terminali.

Riguardo al PC128S

La porta seriale del PC128S è configurata come DTE ed il connettore presente sul retro è un DIN a cinque poli (illustrato in figura 3, ma lo trovate anche sulla guida all'uso, a pag.179), e dispone, oltre alle indispensabili GND, TXD, RXD, solo di due linee di handshake: CTS e RTS. L'uso di questi segnali è necessario specialmente alle alte velocità di trasmissione,

per "non perdere i dati per strada": senza la loro presenza può accadere infatti che una unità trasmetta anche quando l'altra non è pronta a ricevere. Tuttavia, in casi abbastanza semplici, ci si può limitare alle sole linee TXD, RXD, cortocircuitando insieme CTS e RTS con un corto spezzone di filo. Così facendo il computer si autoinforma di essere pronto a trasmettere e/o ricevere.

Credo che sia giusto spendere ancora qualche parola al riguardo, in quanto questa è una situazione che capita di frequente. Il PC128S dispone di un circuito dedicato (il 6850) che gestisce il traffico seriale, composto di due unità completamente indipendenti, una per la trasmissione ed una per la ricezione (in teoria potrebbero anche funzionare con dei baud rate differenti!). Questo implica la possi-

bilità di lavorare normalmente in "full duplex", cioè di poter contemporaneamente ricevere e trasmettere dati, senza il rischio di interferenze. Se anche l'unità periferica con cui ci si collega è dotata delle stesse possibilità, si potrà anche fare a meno dell'handshake. Ecco perchè spesso sui connettori di collegamento di diverse apparecchiature RS232 si vedono dei ponticelli che collegano ad esempio il terminale 5(CTS) con il 20(DTR).

L'apparecchio per trasmettere ha bisogno di ricevere la conferma che il suo interlocutore sia in ascolto; un simile ponticello serve dunque a mantenere la linea di conferma (ad esempio DTR) perennemente attiva, facendo quindi credere alla RS232 che ci sia dall'altra parte del filo un interlocutore pronto. In questo modo il traffico dei dati avviene solo attraverso le

linee RXD e TXD senza praticamente nessun handshake e quindi nessuna sincronizzazione tra trasmettente e ricevente. Qualora la unità periferica sia in grado di comunicare solamente in "half duplex", cioè non possa ricevere e contemporaneamente trasmettere, ma sia costretta a fare le cose una alla volta, un simile collegamento potrà entrare in crisi, via dunque tutti i ponticelli: è necessario sincronizzare a dovere le due apparecchiature.

A questo scopo non esistono solo i segnali già visti, ma c'è anche la possibilità di un "handshake software", attraverso il cosiddetto "protocollo XON/XOFF". Quando un dispositivo sa di non essere pronto a ricevere dati dall'unità remota, può trasmettere uno speciale codice di controllo detto XOFF (corrispondente a CTRL/S ovvero al CHR\$(19)), per bloccare il flusso dei dati. Appena l'unità remota riceve l'XOFF, smette di trasmettere, e rimane in attesa di ricevere il carattere XON (corrispondente a CTRL/Q cioè CHR\$(17)), che segnala la possibilità di riprendere la comunicazione in quanto il dispositivo principale si è liberato ed è nuovamente in grado di ricevere. Con un opportuno scambio di segnali XON/XOFF sarà quindi possibile ottenere una perfetta sincronizzazione tra i due dispositivi.

Adesso le informazioni per quanto riguarda il collegamento hardware sono pressochè complete: basta usare un po' la testa, tenendo ben presente il significato di ciascun segnale e regolandosi volta per volta. In figura 3 sono illustrati alcuni esempi di collegamenti possibili, tra cui uno già collaudato per la comunicazione tra il PC128S ed un IBM PC. Un suggerimento: fidatevi poco dei cavi già pronti, ma verificate sempre tutto di persona, questo può far evitare grandi perdite di tempo.

Come inserire l'espansione RS232

Il PC128S non dispone di serie della porta RS232, ed è dunque necessario acquistarla separatamente. Alla confezione è allegato un foglio di istruzioni in inglese ed italiano; questo è sufficientemente dettagliato e basterà seguire alla lettera le sue indicazioni. La operazione d'installazione risulta comunque sconsigliabile per una persona inesperta, in quanto si tratta

di smontare il computer per aggiungerci dei componenti.

L'espansione RS232 consta praticamente di quattro circuiti integrati da inserire in altrettanti zoccoli già predisposti all'interno del computer; gli integrati vanno identificati per mezzo della loro sigla, che risulta sempre stampigliata sul corpo del componente.

Tutti questi circuiti integrati sono realizzati in tecnologia MOS o C-MOS, e si trovano nella confezione, infilati su di un pezzo di spugnetta conduttrice nera, che li protegge dalle cariche elettrostatiche. Va detto che i dispositivi MOS (Metal Oxide Semiconductor) sono molto sensibili alle cariche statiche, al punto che i primi circuiti integrati potevano essere distrutti semplicemente toccandone i piedini con le dita. Attualmente questi dispositivi incorporano tutti dei circuiti di protezione, per cui possono essere maneggiati con relativa tranquillità. E' comunque buona norma evitare i maglioni di lana e acrilico, le suole di gomma sui pavimenti in linoleum, e tutte le possibili cause di elettricità. Alla occorrenza si può "toccare ferro": un tubo dell'acqua, un termosifone od un grosso oggetto metallico sono sufficienti per scaricare a terra le cariche acquistate.

I piedini dei circuiti integrati sono numerati, partendo dal numero uno; questo è contraddistinto dalla presenza di una tacca sul corpo del componente, e nella inserzione è FONDAMENTALE che questa corrisponda con quella analoga presente sullo zoccolo. Per evitare equivoci, i progettisti hanno predisposto tutti i componenti con la medesima orientazione sul circuito stampato.

Bisogna afferrare gli integrati fra pollice ed indice, senza toccarne i piedini, per inserirli nei corrispondenti zoccoli. Accadrà però sicuramente che i piedini risulteranno troppo divaricati per poter entrare: occorrerà dunque raddrizzarli leggermente, premendoli contro una superficie piana, ad esempio il piano del tavolo. Questa operazione va fatta su entrambi i lati del componente, finchè questo non sarà in grado di entrare a fondo nel suo zoccolo. Per estrarre, qualora ce ne fosse bisogno, un integrato dallo zoccolo, MAI usare le dita, ma aiutarsi facendo leva con un piccolo cacciavite!

Ultima raccomandazione: inserire a

fondo ma dolcemente gli integrati negli zoccoli, senza forzare, e soprattutto assicurarsi che tutti i piedini entrino correttamente, senza che qualcuno sia piegato all'interno od all'esterno del componente.

Concludendo

In questo articolo si sono esaminati tutti i problemi relativi all'hardware di una trasmissione RS232, partendo dalla definizione dello standard, fino all'effettiva realizzazione del collegamento fisico tra due unità. Ora bisogna però effettuare la comunicazione vera e propria, e per fare ciò sono necessari degli appositi programmi in grado di gestire il funzionamento della interfaccia RS232. Delle problematiche software parleremo nel prossimo articolo di questa mini monografia, ma per dare una specie di "contentino" a chi ha avuto la pazienza di seguirci fin qui, oltre ad una anticipazione sul prossimo articolo, ecco un minuscolo programma BASIC, che però è di per sè già un piccolo "emulatore di terminale":

```

10 REM *****
20 REM   Miniterm
30 REM *****
40 REM
50  MODE 128:CLS
60  *FX 2,2
70  *FX 3,7
80 REM Input da tastiera
90 REM Output su RS232
100 PROCInOut
110  *FX 2,1
120  *FX 3,0
130 REM Input da RS232
140 REM Output su schermo
150 PROCInOut
160 GOTO 60
170 END
180 DEFPROCInOut
190 C=INKEY(0)
200 IF C>0 THEN PRINT CHR$(C);
210 ENDPROC

```

Questo programmino sarà utile inoltre per provare l'efficienza di un collegamento RS232, tenendo presente però che il formato di trasmissione utilizzato è quello che il PC128S assume come default all'accensione.