



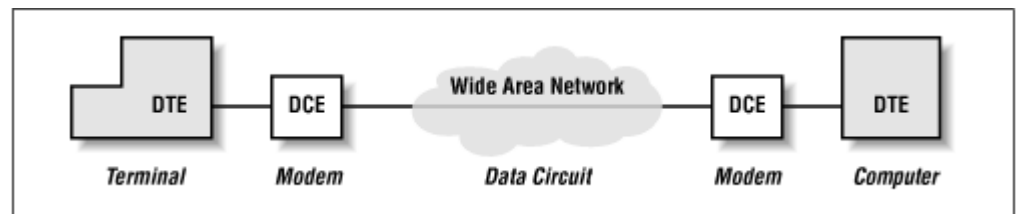
Date: 19/11/2014
 Source: [LINK](#)
 Permalink: [LINK](#)
 Document Version: 1.0
 Total Chars: 17348
 Total Words: 3206
 Created by: HeelpBook
 Page: 1



L'INTERFACCIA RS232

RS-232 è lo standard per la trasmissione seriale definito dell'EIA nel **1969** che caratterizza l'interfaccia per l'interconnessione tra un dispositivo digitale (computer o terminale), identificato dalla sigla **DTE (Data Terminal Equipment)**, e il dispositivo per l'accesso alla rete telefonica, associato alla sigla **DCE (Data Communication Equipment)**. Questo standard supporta la trasmissione asincrona su linee dedicate, affittate, commutate o private, in configurazione punto punto o multipunto, su due o quattro fili, con modalità simplex, half duplex e full duplex tra apparecchiature che hanno un collegamento comune a massa.

La velocità massima consentita è **115kbps**. Il cavo di collegamento **DTE-DCE** non può superare i 15 metri. L'aumento di lunghezza è consentito a patto che la capacità complessiva per circuito rimanga al di sotto dei 2,5nF, pena la riduzione della velocità di trasmissione.



TRASMISSIONE DATI DTE-DCE

Trasmissione dati DTE-DCE: L'interfaccia fisica è costituita da un connettore del tipo DB25 generalmente maschio. Per particolari applicazioni su PC, collegamenti di *mouse* o di *microterminali*, sono stati disposti connettori di dimensioni ridotte, a **9 poli (DB9)**. Secondo lo standard il connettore femmina deve essere disposto sulla porta del **DCE** mentre il connettore maschio deve essere posto sul **DTE**.

Di seguito vediamo la tabella di connessione dei pin del connettore:

Pin DB25	Pin DB9	Nome	Significato	Descrizione
1	5	GND	Ground	Massa del circuito
2	3	TXD	Trasmissione Data	Trasmissione dati dal DTE al DCE
3	2	RXD	Recieve Data	Ricezione dati dal DCE al DTE
4	7	RTS	Request To Send	Disponibilità a trasmettere dal DTE al DCE
5	8	CTS	Clear To Send	Consenso alla trasmissione dal DCE al DTE
6	6	DSR	Data Set Ready	Indica che il DCE ha terminato la fase di test ed è pronto a ricevere I dati dal DCE al DTE
7	5	Data GND	Data Ground	Schermatura cavo
8	1	CD	Carrier Detect	Il DCE riceve correttamente la portante dalla linea
20	4	DTR	Data Terminal Ready	Indica che il DTE è pronto sia per ricevere che trasmettere dati dal DTE al DCE
22	9	RI	Ring Indicator	Indica la presenza sulla linea del segnale di chiamata



Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 2

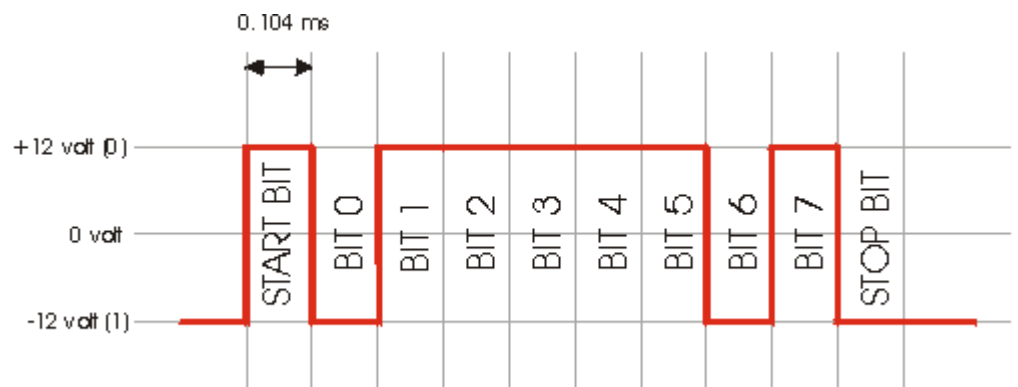


Lo standard **RS232** supporta due tipi di trasmissioni, **sincrona** e **asincrona**. Quest'ultima è la più usata perché più economica e supportata dalla maggior parte dei PC in circolazione. A differenza della trasmissione sincrona, l'asincrona non necessita di particolari accorgimenti per sincronizzare i due dispositivi collegati, per questo è anche meno costosa è però meno veloce infatti il flusso di dati non è continuo a causa della presenza dei bit di controllo.

Analizziamo cosa avviene quando si deve trasferire un byte da un **DTE** ad un **DCE**. In assenza di trasmissione la tensione sulla linea è di **-12V** che corrisponde ad un livello logico "1".

Per avvisare il **DCE** che sta per avvenire la trasmissione, il **DTE** porta la tensione della linea a **+12V** (livello logico "0") per un tempo **t** pari all'inverso della velocità di trasmissione (**1/bps**). A questo punto il **DCE** è pronto a ricevere gli 8 bit del byte che vogliamo trasmettere, partendo da quello meno significativo al più significativo. Alla fine della trasmissione, dal **DTE** viene inviato il bit di stop.

Quest'ultimo ha un livello di tensione di **-12V** ed una durata di circa **1,5 – 2 volte il bit di start**: questa differenza è necessaria per far in modo che il **DCE** o ricevitore non confonda questo bit con quelli dei dati precedentemente trasmessi.



Via software è possibile cambiare alcuni parametri della seriale, per adattarsi alle proprie necessità. In particolare è possibile agire su: **Baud rate**, **Numero di BIT**, **Parità**, **Bit di stop**.

L' impostazione **Baud rate** serve per indicare la velocità di trasmissione dei bits. La velocità minima consentita è di **300Bit/Sec** fino ad un massimo di **115.000bps**.

L'impostazione Numero di BIT consente di scegliere se inviare tutti gli 8 bit del carattere da trasmettere, o escludere l'ultimo inviando quindi solo 7 bit. Se si adopera quest'ultima modalità, non potranno essere inviati caratteri con valore superiore a **127 (7Fh)**. Se si tenta di inviare caratteri con valore superiore a 127 usando la modalità a 7 bit, il bit più significativo non verrà inviato.



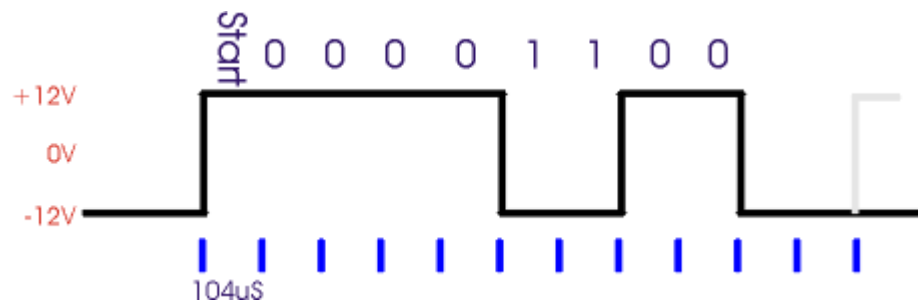
Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 3



Con l'opzione **Parità**, è possibile aggiungere ad ogni invio di carattere un bit che indica se il numero di bit a 1 del carattere è pari o dispari. Questa soluzione consente di individuare eventuali errori di trasmissione dovuti a disturbi di linea.

L'impostazione **Bit di stop** serve per indicare il numero di **bit di Stop** da inserire in coda ad ogni invio di carattere. Questi servono per marcare la fine del treno di bit del carattere. Le possibili opzioni di scelta sono: **1-2-1,5**.

Nell'immagine che segue è visualizzato, in modo idealizzato, cosa appare collegando un oscilloscopio ad un filo su cui transita un segnale **RS-232 a 9600 bps** del tipo **8n2 ()** rappresentante il valore binario **00110000**.



L'ampiezza del segnale è caratterizzata da un valore “**alto**” pari a circa +12V ed un valore “**basso**” pari a circa -12V. Da notare che, nello standard **RS-232** un segnale alto rappresenta lo zero logico ed uno basso un uno, come indicato nel disegno e rovesciato rispetto al “comune pensare”.

A volte un segnale alto (+12V, cioè **uno zero logico**) è indicato come **space** ed uno basso (-12V, **uno logico**) come **mark**. Tutte le transizioni appaiono in corrispondenza di multipli di **104 us** (104 microsecondi; ho usato una notazione molto diffusa nel settore elettronico, senza scomodare l'alfabeto greco), pari ad **1/9600** cioè ciascun bit dura esattamente l'inverso del baud rate.

La linea si trova inizialmente nello stato di riposo, bassa (nessun dato in transito); la prima transizione da basso in alto indica l'inizio della trasmissione (inizia il “**bit di start**”, lungo esattamente **104us**). Segue il **bit meno significativo (LSB)**, dopo altri 104 us il secondo bit, e così via, per otto volte, fino al **bit più significativo (MSB)**. Da notare che il byte è trasmesso “**al contrario**”, cioè va letto da destra verso sinistra.

Segue infine un periodo di riposo della linea di almeno 208 us, cioè due bit di stop e quindi (eventualmente) inizia un nuovo pacchetto di bit con un nuovo bit di start (in grigio nel disegno).



Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 4



Le varianti possibili sono le seguenti:

- Se la trasmissione è più veloce o più lenta, la distanza tra i fronti varia di conseguenza (p.e. a 1200 bps le transizioni avvengono a multipli di 0,833 ms, pari a 1/1200)
- Invece di trasmettere 8 bit, ne posso trasmettere 6, 7 o anche 9 (ma quest'ultima possibilità non è prevista dalle porte seriali dei normali PC)
- Alla fine è possibile aggiungere un bit di parità, descritto più avanti
- Alla fine la linea rimane nello stato di riposo per almeno 1 o 1.5 o 2 bit; notare che, se non ho più nulla da trasmettere, il “**riposo**” è molto più lungo, ovviamente. Molti sistemi non possono utilizzare 1.5 bit di stop.

In genere il formato del pacchetto trasmesso è indicato da una sigla composta da numeri e cifre, per esempio 8n1 e 7e2:

- La prima cifra indica quanti bit di dati sono trasmessi (nei due esempi rispettivamente 8 e 7)
- La prima lettera il tipo di parità (rispettivamente nessuna ed even-parity, cioè parità pari)
- La seconda cifra il numero di bit di stop (rispettivamente 1 e 2)

Tenendo conto che esiste sempre un solo **bit di start**, un singolo blocco di bit è quindi, per i due esempi riportati, costituito rispettivamente da **10 (1+8+0+1)** e **11 (1+7+1+2)** bit. Da notare che di questi bit solo 8 e, rispettivamente, 7 sono effettivamente utili.

Lo standard originale prevede una velocità fino a **20Kbps**. Uno standard successivo (**RS-562**) ha portato il limite a **64Kbps** lasciando gli altri parametri elettrici praticamente invariati e rendendo quindi i due standard compatibili a bassa velocità. Nei normali PC le cosiddette interfacce seriali **RS-232** arrivano in genere almeno a **115Kbps, 230Kbps** o anche più: pur essendo tali valori formalmente al di fuori di ogni standard ufficiale non si hanno particolari problemi di interconnessione.

Una precisazione: trasmettitore e ricevitore devono accordarsi sul modo di trasmettere prima di iniziare la trasmissione stessa, pena l'impossibilità di instaurare la trasmissione o ricevere bit che appaiono casuali. Questa operazione va fatta configurando opportunamente il software e/o modificando manualmente alcuni dip-switch o altri dispositivi hardware. E' importante garantire il rigoroso rispetto della durata dei singoli bit: infatti non è presente alcun segnale di clock comune a trasmettitore e ricevitore e l'unico elemento di sincronizzazione è dato dal fronte di salita del bit di start.

Come linea guida occorre considerare che il campionamento in ricezione è effettuato di norma al centro di ciascun bit: l'errore massimo ammesso è quindi, teoricamente, pari alla durata di mezzo bit (circa il 5% della frequenza di clock, considerando che anche il decimo bit deve essere correttamente sincronizzato).



Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 5



Naturalmente questo limite non tiene conto della difficoltà di riconoscere con precisione il fronte del bit di start (**soprattutto su grandi distanze ed in ambiente rumoroso**) e della presenza di interferenze intersimboliche tra bit adiacenti: per questo spesso si consiglia caldamente di usare un clock con una precisione migliore dell'1% imponendo, di fatto, l'uso di oscillatori a quarzo.

Si potrebbe anche ipotizzare un meccanismo che tenta di estrarre il clock dai fronti intermedi ma si tratta nel caso specifico di un lavoro poco utile, visto che la lunghezza del pacchetto è piuttosto breve.

IL BIT DI PARITÀ

Oltre ai bit dei dati (in numero variabile tra 5 ed 9) viene inserito un bit di parità (opzionale) per verificare la correttezza del dato ricevuto. Esistono diversi tipi di parità:

None: nessun tipo di parità, cioè nessun bit aggiunto

Pari (even): il numero di mark (incluso il bit di parità) è sempre pari

Dispari (odd): il numero di mark (incluso il bit di parità) è sempre dispari

L'idea è quella di predeterminare la quantità di 1 (e di conseguenza di 0) da trasmettere, facendo in modo che il loro numero sia sempre pari (o dispari, a secondo della scelta che si vuole fare): così facendo, se durante la trasmissione dovesse accadere un errore su un singolo bit, il ricevitore sarebbe in grado di rilevare l'errore, ma non di correggerlo. Si tratta ovviamente di un protocollo di controllo degli errori elementare e di conseguenza in disuso a favore di altri sistemi basati su codici a ridondanza ciclica (CRC) o altri algoritmi più complessi.

Il collegamento **RS-232** (il cosiddetto “**collegamento seriale**“) consente di collegare quasi qualunque cosa assieme e farla colloquiare.

Vi e' sempre piaciuto collegare cavi e vedere se funziona? Non avete mai capito perche' non funziona un collegamento? Se avete questi dubbi leggete di seguito.

Il collegamento **RS-232** permette lo scambio di informazioni tra due apparecchiature. Lo scambio di informazioni avviene su una linea seriale a bassa velocità e con modalità piuttosto semplici. Per tale motivo il protocollo **RS-232** consente lo scambio di dati non solo tra un **terminale** (DTE) ed un **modem** (DCE) (come in origine previsto dallo standard) ma tra componenti di diverso tipo (eg. PC e stampanti, host e terminali, tra due sistemi, ...praticamente qualsiasi dispositivo informatico ha un collegamento seriale). Per collegare due sistemi in **RS-232** debbono essere soddisfatte le seguenti condizioni:

- Utilizzare su entrambe i sistemi con prese adatte (non è scontato).
- Definire la corretta logica sulla linea.
- Utilizzare i corretti programmi di comunicazione su entrambe i sistemi.
- Utilizzare su entrambe i sistemi con prese adatte.



Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 6



La prima condizione per inserire un cavo in una apparecchiatura è quella che il cavo “**ci entri**”. Per soddisfare questo requisito si può utilizzare il cavo multiplo che presenta i principali tipi di presa. Sono anche disponibili alcuni scambiatori di sesso (*gender charger*) o adattatori da 25 a 9 pin.

Naturalmente debbono essere collegate tra loro due porte seriali (sui PC le parallele sono femmine con 25 pin) i cavi di alimentazione ed i collegamenti **SCSI** sono ben riconoscibili. Riassumendo:

- Utilizzare il cavo multipoint e tutti i gender charger necessari.
- Definire la corretta logica sulla linea.

Per far sì che la comunicazione avvenga è necessario le due apparecchiature si riconoscano. Poiché nel mondo **RS-232** esistono solo **DCE** e **DTE** basta che ciascuno ritenga di parlare con qualcun'altro. Il modo banale per far sì che ciò avvenga è:

- Se tutto funziona non fare nulla, lasciare il cavo multipoint così com'è.
- Se non funziona provare ad incrociare il cavo (utilizzare uno Switch box e collegare i segnali 2 con il 3, 3×2 e 7×7).

LA VELOCITÀ DI TRASMISSIONE

Le unità di misura della velocità di trasmissione sono essenzialmente due: il **baud** ed il **bit per secondo** (bps o b/s), spesso trattate erroneamente come sinonimi.

Il **baud** (o anche **baud rate**) indica il numero di transizioni al secondo che avvengono sulla linea. Esso è associato alla banda occupata dal segnale, inteso come segnale analogico. Il cavo utilizzato per la trasmissione viene dimensionato in funzione a tale valore: in teoria serve una banda passante pari ad almeno alla metà del baud rate anche se nella realtà è bene utilizzare cavi con banda molto più ampia

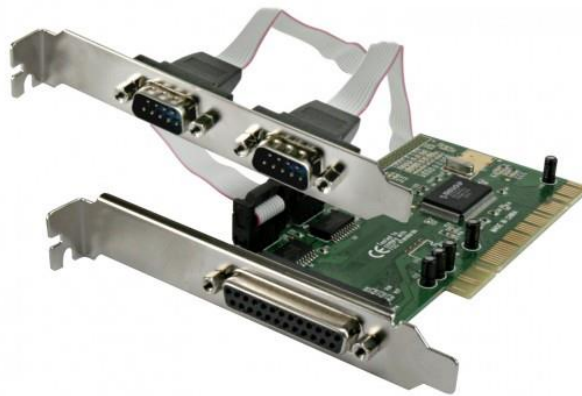
Il bps indica, come dice il nome, quanti bit al secondo sono trasmessi lungo la linea. Questa è la velocità effettiva della trasmissione vista dai dispositivi digitali.

Nel caso di **trasmissione binaria** (cioè è presente un livello di tensione alto ed uno basso) **baud rate** e **bps** coincidono numericamente, da cui la parziale equivalenza dei due termini. Nel caso di trasmissioni a più livelli, invece, è possibile trasmettere con una sola transizione più bit, ottenendo un baud rate minore a parità di informazioni trasmesse, guadagnando in termini di minore occupazione di banda a spese di una maggiore complicazione circuitale e peggioramento del rapporto **segnale/rumore**. Per esempio la codifica multilivello **PAM5** (dove il bit rate per ciascun segnale è doppio del baud rate) permette alle reti gigabit ethernet di raddoppiare la velocità di trasmissione rispetto alla fast ethernet a parità di banda occupata e quindi usando gli stessi cavi.

Lo standard **RS232** utilizza due livelli quindi il **baud rate** coincide numericamente con il **bps**.



Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 7



Lo standard **RS-232**, poiche' funzionava, e' stato utilizzato per tutto e non solo per la comunicazione tra **DCE** e **DTE**. E' quindi normale vedere collegati in **RS-232** PC, stampanti, plotter, server Unix ed altro ancora in varie combinazioni. Anche le recenti schede **Arduino** (che cito soprattutto perche' sono italiane) utilizzano il classico protocollo seriale. Ogni periferica ha una sua particolare idea di come funziona lo standard **RS-232**, ritiene infatti di essere un **DTE** o un **DCE** ed utilizza un certo numero di segnali.

Cosa ne pensa una periferica dello standard **RS-232** lo si legge sul suo manuale oppure lo si sperimenta. Per far si' che tutto funzioni e' necessario far credere ad una periferica **DTE** di stare parlando con un **DCE** e viceversa.

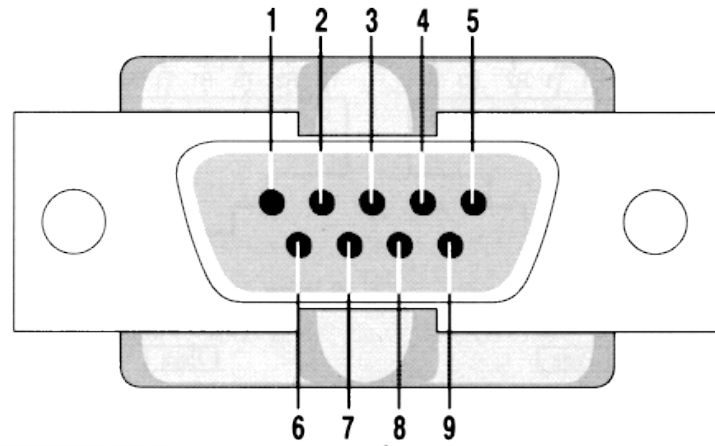
CARATTERISTICHE FISICHE

Teoricamente nello standard erano state previste impedenze, voltaggi, lunghezze di cavi, ... In generale fino ad una decina di metri funziona sempre tutto! Tra gli elementi importanti dello standard vi e' quello che e' possibile porre in corto circuito qualsiasi elemento senza che il dispositivo ne risenta. Il voltaggio, positivo o negativo, e' in genere limitato a **12 Volt**. Quindi di solito non si prende la scossa (attenzione alla massa).

COLLEGAMENTI

I principali collegamenti utilizzati sono le prese **DB 25 pin M/F**, **DB 9 pin M/F** ed **RJ12**. La presa **RJ12** e' quella del telefono (collegamento all'americana), per sapere com'e' fatta una presa DB guardate sul retro del vostro PC. Per riconoscere il sesso della presa (M/F) e' sufficiente un po' di immaginazione! I collegamenti sono i seguenti:

DB25 pin	DB9 pin	RJ12
1 HEART	1 CD	1 bianco Tx
2 Tx	2 Rx	2 nero Rx
3 Rx	3 Tx	3 rosso GND
4 RTS	4 DTR	4 verde GND
5 CTS	5 GND	5 giallo DTR
6 DSR	6 DSR	6 blu DSR CD
7 GND	7 RTS	loop RTS CTS
8 CD	8 CTS	
20 DTR	9 Ring	
22 Ring		



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

SEGNALI

I segnali definiti sull'**RS-232** sono:

Segnale	Descrizione
DTR	Data Terminal Ready - Il DTE e' pronto
DSR	Data Set Ready - Il DCE accetta la connessione
RTS	Ready To Send - Il DTE e' pronto
CTS	Clear To Send - Il DCE da il permesso a trasmettere
...	

NOMENCLATURA

La nomenclatura dei segnali **RS-232** e' a volte differente. Spesso sui modem (sempre parlando con tecnici Telecom) erano utilizzati i seguenti nomi:

Segnale	Alias
DTR	108
DSR	107
RTS	105
CTS	106
Tx	103
Rx	104
CD	109



DIMMI CHI SEI...

Alcune apparecchiature hanno un sesso ed un comportamento ben definito. Dato il sesso ed il comportamento si puo' predisporre un cavo ed un settaggio di parametri appropriato. Vediamo gli esempi piu' comuni:

Date: 19/11/2014
Source: [LINK](#)
Permalink: [LINK](#)
Document Version: 1.0
Total Chars: 17348
Total Words: 3206
Created by: HeelpBook
Page: 9



- **Modem.** Un modem e' un **DCE** sempre. Tratta tutti i segnali (e' possibile comunque configurarlo diversamente). Lavora alle velocita' per cui e' stato progettato.
- **Personal Computer.** I PC standard sono **DTE**, tipicamente avevano delle porte **DB25 M** (i portatili avevano tipicamente le **DB9 M**). La parallela tipicamente era una **DB25 F**. Attenzione andava riposta ai segnali di controllo, non sempre sono tutti implementati. Attenzione alle velocita', PC non di recente costruzione possono ospitare chip UART che hanno problemi a lavorare con velocita' oltre i 9600. Attenzione ai portatili, alcuni decidono di disabilitare le porte seriali poiche' non necessarie (per risparmiare energia elettrica).
- **Sistemi Unix.** I sistemi Unix hanno tipicamente connessioni **DTE**. Tra le eccezioni vi sono i sistemi HP che hanno uno strano comportamento sono infatti dei **DCE** con i pin del **DTE**. Questo consente di utilizzare un cavo dritto verso un terminale ma richiede cavi ad hoc per altre applicazioni.
- **Stampanti.** Le stampanti sono solitamente dei **DTE**. Sono tuttavia moltissime le eccezioni (una cosa banale: non confondere una porta parallela con una seriale, alcune stampanti le hanno entrambe). La gestione dei segnali e' aleatoria (generalmente la documentazione al riguardo e' incomprensibile). Nei casi piu' fortunati basta un cavo con due poli: **GND** ed **Rx** (una stampante non e' in grado di trasmettere!); quando e' cosi' il controllo di flusso viene effettuato con caratteri speciali (i famosi **XON** ed **XOFF**). La non corretta gestione dei controlli di flusso porta a problemi simpaticissimi. Infatti le stampanti sono lente ed hanno tipicamente un **buffer**: quindi alcune stampe avvengono correttamente ed altre no, quando non c'e' carta sembrano accettare lo stesso i dati, ogni tanto stampano caratteri strani,
- **Terminali.** Sono quasi sempre **DTE**. Tipicamente non gestiscono i segnali di controllo e sono quindi sufficienti tre poli **GND**, **Rx**, **Tx**.
- **Altro.** Hub, router, schede di management, robot, GPS, antifurti, cucine componibili, ... Quasi tutte le apparecchiature hanno un'uscita **RS-232** per la programmazione, il controllo o altro. Sono solitamente **DTE DB9M** (piu' raramente **DCE DB25F**), non gestiscono i segnali di controllo e ci si collega con i parametri magici 9600, 8 bit, 1 stop, no-parity.



ALCUNI COLLEGAMENTI DI ESEMPIO

Alcune situazioni vengono risolte con cavi sempre uguali. Riportiamo i piu' comuni.

Il cavo dritto

- **Caso tipico:** collegamento verso un modem.
- **Cavo:** 2×2, 3×3, 4×4, 5×5, 6×6, 7×7, 8×8, 20×20, 22×22 (insomma tutto dritto)

Il cavo incrociato

- **Caso tipico:** collegamento tra due PCs.
- **Cavo:** 2×3, 3×2, 7×7, (cosi' si invertono Tx ed Rx)

Il collegamento in loopback

- **Caso tipico:** prova della seriale DB25 su PC.
- **Cavo:** 2×3, 3×2, 4×5, 5×4, 6×20, 20×6 tutti in sullo stesso lato (cosi' ci si risponde da soli) su seriale di tipo **DB25**.
- **Cavo:** 3×2, 8×7, 4x1x6 (cosi' ci si risponde da soli) su seriale di tipo **DB9**.

