

L'interfaccia seriale RS232 è uno standard costituito da una serie di protocolli meccanici, elettrici ed informatici che rendono possibile lo scambio di informazioni a bassa velocità tra dispositivi digitali. In questo tutorial si illustreranno principalmente le caratteristiche di natura elettrica con l'obiettivo preciso di fornire le poche nozioni necessarie per progettare l'hardware di dispositivi elettronici che comunicano con un PC attraverso questa porta.

Lo standard RS232 nacque nei primi anni '60 per opera della "Electronic Industries Association" ed era orientato alla comunicazione tra i mainframe e i terminali attraverso la linea telefonica, utilizzando un modem. Esso includeva le caratteristiche elettriche dei segnali, la struttura e temporizzazioni dei dati seriali, la definizione dei segnali e dei protocolli per il controllo del flusso di dati seriali su un canale telefonico, il connettore e la disposizione dei suoi pin ed infine il tipo e la lunghezza massima dei possibili cavi di collegamento.

Nel corso di questi oltre 40 anni lo standard si è evoluto pur mantenendosi in larga parte invariato. L'evoluzione è riconoscibile dalla sigla, leggendo l'ultima lettera; l'ultima revisione è del 1997 ed è indicata come RS232f. Probabilmente la versione più diffusa è la RS232c, del 1969, corrisponde alle specifiche europee CCITT raccomandazione V.24

Pur essendo un protocollo piuttosto vecchio, attualmente la rs232 è ancora largamente utilizzata per la comunicazione a bassa velocità tra microcontrollori, dispositivi industriali ed altri circuiti relativamente semplici che non necessitano di particolare velocità; è invece praticamente scomparsa in ambito "desktop", ambito nel quale lo standard è nato per la comunicazione tra un computer ed un modem.

Le informazioni raccolte in questo tutorial, relative principalmente al livello elettrico, sono state codificate inizialmente dallo standard EIA/TIA-232-E (sezione 2) e successivamente modificate in alcuni dettagli secondari dallo standard EIA/TIA-562.

Le caratteristiche generali delle interfacce seriali

L'interfaccia RS-232 utilizza un protocollo seriale asincrono; il segnale elettrico è non bilanciato; il collegamento è di tipo point-to-point. Vediamo di spiegare il significato di questi termini:

- **Seriale** specifica che i bit che costituiscono l'informazione sono trasmessi sequenzialmente, uno alla volta su di un solo "filo". Questo termine è in genere contrapposto a "parallelo", termine che indica una tipologia di

trasmissione in cui i dati viaggiano contemporaneamente su più fili, per esempio 8, 16 o 32. Parlando astrattamente si potrebbe pensare che la trasmissione seriale sia intrinsecamente più lenta di quella parallela (su di un filo possono passare meno informazioni che su 16...). In realtà questo non è vero in assoluto, soprattutto a causa della difficoltà di controllare lo skew, cioè il disallineamento temporale tra i vari segnali, sempre presente nel caso di molti trasmettitori in un bus parallelo; per esempio in una fibra ottica, in un cavo ethernet, USB o FireWire, in un bus PCI-Express (tutti standard seriali) le informazioni transitano ad una velocità spesso superiore a quella di un bus PCI a 32 fili. In questa nota applicativa si parlerà solo di interfacce seriali "lente" cioè gestibili da PC e microcontrollori "normali"

- **Asincrono** significa, in questo contesto, che i dati sono trasmessi senza l'aggiunta di un segnale di clock, cioè senza alcun segnale comune tra trasmettitore e ricevitore destinato a sincronizzare il flusso di informazioni; ovviamente sia il trasmettitore che il ricevitore devono comunque essere dotati di un proprio clock locale per poter interpretare correttamente i dati
- Un segnale **non bilanciato** (o single ended) è caratterizzato dal fatto che la tensione associata al bit trasmesso o ricevuto viene misurata rispetto ad un riferimento comune detto massa. Nel caso dei segnali RS232 questa tensione può essere sia positiva che negativa.
- Una trasmissione è di tipo **point-to-point** quando nella comunicazione è presente, per ciascun segnale utilizzato, un solo trasmettitore ed un solo ricevitore; tale termine può essere contrapposto a **point-to-multipoint**, che indica la situazione in cui, a fronte di un trasmettitore, esistono più ricevitori.

La velocità di trasmissione

Le unità di misura della velocità di trasmissione sono essenzialmente due: il **baud** ed il **bps** (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 **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.

Half-duplex e full-duplex

I due termini fanno riferimento alla situazione in cui due dispositivi si scambiano informazioni tra di loro, comportandosi entrambi sia da sorgente di informazioni (cioè da talker o, in sigla, Tx) sia da ricevitore (listener o Rx).

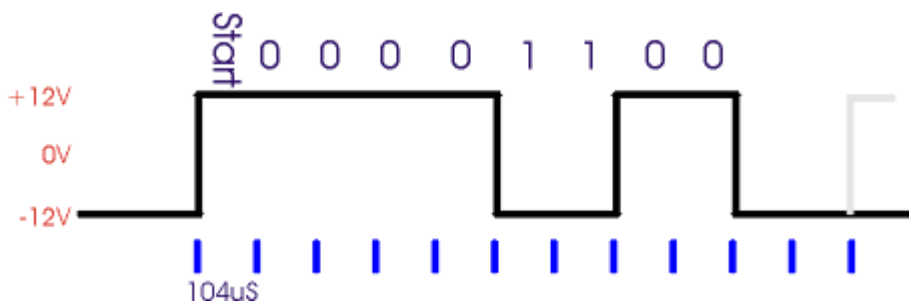
- **Half-duplex** indica che la trasmissione è bidirezionale ma non contemporanea nei due versi: in un determinato istante uno solo dei due dispositivi emette segnali, l'altro ascolta. Quando è necessario, si scambiano di ruolo.
- La trasmissione **full-duplex** indica che la trasmissione è bidirezionale e contemporanea. In questo caso sono necessari due fili, uno per ciascun verso di trasmissione. In alcuni sistemi di comunicazione, quali il comune telefono, possono essere adottati meccanismi che permettono la trasmissione full-duplex con un solo filo.
- Se la trasmissione è sempre in un solo verso, si parla di **simplex**.

Lo standard RS232 permette tutte e tre queste modalità di funzionamento in quanto è utilizzato un conduttore separato per ciascun verso di trasmissione. In genere nel caso di trasmissione duplex è necessario che i dati in trasmissione e ricezione abbiano lo stesso formato e la stessa velocità. Inoltre ciascuno dei due nodi deve avere sufficiente potenza di calcolo per la gestione del duplice flusso di informazioni, condizione non sempre possibile quando la codifica del segnale è fatta solo utilizzando il software e senza assistenza di hardware dedicato.

Come è fatto un segnale RS-232

La cosa più semplice per descrivere un segnale RS232 è partire con un esempio.

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 (più avanti verrà spiegata questa sigla) 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 $2 \times 104 = 208$ 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).

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 cifra 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). 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 seconda 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 bit di parità a volte viene mantenuto sempre ad un livello prestabilito, per esempio in alcuni protocolli usati da macchine industriali. Ciò dà origine ad ulteriori due tipologie di parità, peraltro non molto comuni:

- Mark: il bit di parità vale sempre mark
- Space: il bit di parità vale sempre space

Tali configurazioni sono a volte usate per identificare la tipologia del byte trasmesso, per esempio potrebbe indicare se si tratta di un dato piuttosto che di un indirizzo.

I parametri elettrici della RS-232


La tensione di uscita da un trasmettitore RS232 deve essere compresa in valore assoluto tra 5V e 25V (quest'ultimo valore ridotto a 13V in alcune revisioni dello standard). A volte le tensioni in uscita sono intenzionalmente diminuite a +/- 6V anziché 12V per permettere minori emissioni EMC, peraltro sempre critiche, e favorire maggiori velocità di trasmissione.

Il ricevitore deve funzionare correttamente con tensioni di ingresso comprese, sempre in modulo, tra i 3V ed i 25V. Molti ricevitori commerciali considerano semplicemente una tensione di soglia al valore di +2V (sopra viene riconosciuto un segnale alto, sotto uno basso) anche se ciò non è pienamente aderente alla norme. E' però una scelta utile per effettuare una trasmissione "rs232" con livelli TTL...

L'impedenza di uscita del trasmettitore deve in ogni situazione essere maggiore di 300 ohm; l'impedenza di ingresso deve essere compresa tra i 3 ed i 7 kohm, anche a dispositivo spento. La corrente prelevabile in uscita mantenendo i corretti valori logici deve essere di almeno di 1.6 mA (potrebbe però essere maggiore, anche di un ordine di grandezza) e nel caso di corto circuito deve comunque essere minore di 100mA.

Infine lo slew-rate (cioè la pendenza del grafico del segnale nel passare da 1 a 0 o viceversa) deve essere minore di 30V/us per evitare eccessive emissioni elettromagnetiche.

Come collegare una porte TTL o CMOS alla RS232

In genere i segnali utilizzati dai sistemi digitali sono "TTL compatibili", cioè variano tra 0 e 5V, oppure variano tra 0 V e 3,3 V: non sono quindi direttamente compatibili con la standard RS232. In commercio esistono appositi  che hanno il compito di fornire sia in trasmissione che in ricezione gli opportuni livelli pur non modificando la forma del segnale trasmesso.

Alcuni integrati (per esempio i classici 1488 e 1489, rispettivamente un trasmettitore ed un ricevitore, ambedue a quattro canali) sono molto usati in sistemi in cui è presente (oltre all'alimentazione logica di 5V o 3.3V) anche un'alimentazione duale a +/-12V.

Questi integrati, come praticamente tutti i circuiti di questo tipo, contengono un inverter per ciascun canale e quindi, per esempio, se in ingresso è presente una tensione di -12V (cioè "uno" logico) in uscita saranno ovviamente presenti 5V (cioè "uno" logico).

L'uso di questi integrati è semplice ma non è sempre attuabile a causa della necessità di disporre di tre alimentazioni: si pensi per esempio alle apparecchiature alimentate a batteria. Il MAX232 (ed integrati simili, fatti da un po' tutti i produttori di semiconduttori) è un circuito integrato che permette il collegamento tra logica TTL o CMOS a 5V e le tensioni RS-232, partendo solo da un'alimentazione a 5V.

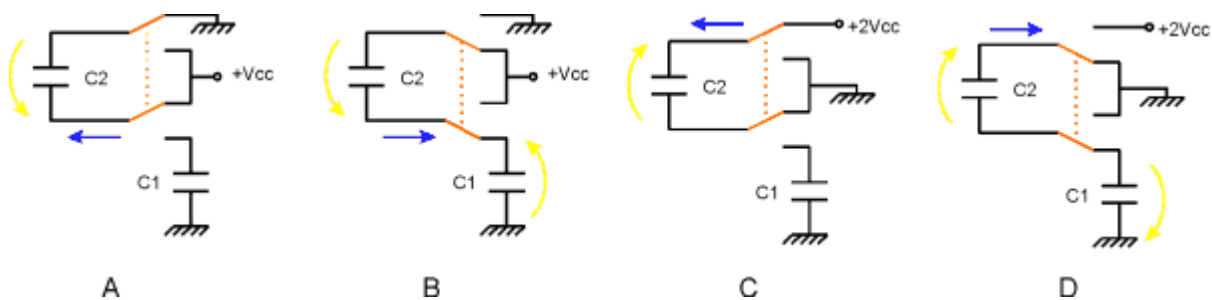
Per ottenere la tensione positiva e negative necessarie per il funzionamento dell'integrato è usata una configurazione a pompa di carica, costituito da circuiti interni all'integrato e quattro condensatori esterni. La capacità effettiva dipende dal tipo di integrato e dalla relativa frequenza di commutazione; a volte i condensatori sono presenti all'interno dell'integrato stesso. Sono disponibili anche integrati che richiedono un'alimentazione di soli 3.3V (p.e. il MAX3232).

La sezione ricevente del MAX232 è costituita da due porte invertenti che accettano in ingresso una tensione di +/- 12V (o altra tensione compatibile allo standard RS232) ed in uscita presentano un segnale TTL compatibile.

La sezione trasmittente ha due driver invertenti con in ingresso TTL compatibile e capaci di erogare a vuoto una tensione di poco meno di +/- 10V, compatibile con lo standard RS232.

Un circuito a pompa di carica

Per ricavare le tensioni positive e negative necessarie per garantire i livelli richiesti dalla RS232 è pratica comune utilizzare un duplicatore ed un invertitore di tensione a pompa di carica.



Le figure A e B mostrano come viene ottenuto il raddoppio della tensione. Una immagine che rende l'idea è quella di un contenitore (C2) che preleva acqua da una fonte e la riversa in un secondo contenitore (C1) posto a maggiore altezza. Più in dettaglio:

- Inizialmente il condensatore C2 viene connesso tra massa e Vcc; quindi la corrente (in blu) carica C2 alla tensione di alimentazione (in giallo). Quindi $V_{c2} = V_{cc}$
- C2 viene successivamente connesso tra Vcc ed un secondo condensatore C1; la tensione ai capi di C1 deve essere uguale alla somma di Vcc e V_{c2} e quindi C2 si scarica verso C1, che aumenta la propria tensione rispetto a massa
- Il processo è ripetuto fino a quando la tensione ai capi di C1 è uguale a $2V_{cc}$: in questo caso infatti C2 non si può più scaricare.

Da notare che, nel funzionamento normale, il processo non può mai interrompersi in quanto il carico collegato a C1, non disegnato, assorbe corrente e quindi tende a scaricare C2 stesso.

Analogamente le figure C e D mostra l'inversione di tensione:

- Inizialmente C2 è caricato alla tensione di alimentazione (magari, come nel disegno da $2V_{cc}$, ricavata con il precedente circuito)
- Quindi C2 è connesso tra massa e C1 avendo cura di invertire le polarità. In questo modo C1 si carica a $-2V_{cc}$

Il limite dei circuiti a pompa di carica è la limitata quantità di corrente disponibile: infatti se prelevo corrente da C1 questo tende a scaricarsi, facendo scendere la tensione; la corrente generata da un circuito integrato tipo Max232 è generalmente tutta utilizzata per il solo funzionamento del driver e quindi non è disponibile per altri circuiti.

La piedinatura del connettore RS-232 del PC

Nei personal computer sono disponibili due tipi di connettori RS-232: DB9 (nove pin) e DB25 (25 pin, il connettore originale e presente solo sui PC più vecchi); ambedue

i connettori sono maschi e praticamente identici dal punto di vista funzionale anche se non coincidente con quello proposto dallo standard ufficiale.



Nell'immagine sono visualizzati in basso i due tipi di connettori utilizzati sui PC di vecchia generazione; in quelli più recenti sono utilizzati solo connettori DB9 (quando ci sono, oggi lo standard RS232 tende ad essere relegato solo in ambiti industriali e ad essere sostituito dallo USB).

Il connettore femmina in alto è relativo alla porta parallela.

Di seguito la tabella con indicati i nomi dei segnali, il numero dei pin e la direzione del segnale (O = uscita dal PC).

Sigla	25pin	9pin	In/Out	Nome
TxD o Tx	2	3	O	Dati trasmessi
RxD o Rx	3	2	I	Dati ricevuti
RTS	4	7	O	Request To Send
CTS	5	8	I	Clear To Send
DTR	20	4	O	Data Terminal Ready
DSR	6	6	I	Data Set Ready
RI	22	9	I	Ring Indicator
DCD	8	1	I	Data Carrier Detect
GND	7	5	-	Massa
-	1	-	-	Terra

Non ho riportato un disegno dello schema per un semplice motivo: in genere crea solo problemi interpretativi in particolare se non è specificato se si tratta del connettore maschio o di quello femmina e se la vista è dal lato connettore o dal lato dei pin. Molto meglio leggere i numeri sempre riportati sul connettore plastico.

Una nota: quello usato dal PC non è il connettore previsto ufficialmente dallo standard che assegna una funzione a tutti e 25 i pin del connettore DB25.

Perché tanti fili ?

In teoria per ricevere e trasmettere un segnale RS-232 bastano tre fili: ricezione, trasmissione e massa. Spesso lo è anche in pratica.

Gli altri fili (spesso opzionali, ma dipende dall'applicazione) servono per il cosiddetto handshake tra PC e periferica (o tra PC e PC) cioè per sincronizzare in hardware la comunicazione.

Sono presenti due coppie di fili:

- RTS/CTS: quando il PC inizia la trasmissione pone RTS alto, la periferica risponde quando pronta ponendo CTS alto. Per interrompere la trasmissione la periferica pone CTS basso.
- DTR/DSR: Quando il PC è collegato per la prima volta, pone alto DTR. La periferica risponde ponendo alto DSR

Purtroppo questo modo di procedere ha un'infinità di variazioni. La parola d'ordine è: arrangiarsi per tentativi!

Un uso alternativo dei pin RTS e DTR è l'utilizzo come fonte di alimentazione del dispositivo collegato alla porta seriale stessa. L'esempio classico è il mouse seriale ma nulla impedisce di collegare un microcontrollore generico o qualche altro circuito. Unico ed importante limite è la corrente erogata, visto che questi pin non sono pensati per questo uso: è opportuno limitarsi ad un paio di mA anche se molti PC permettono di arrivare tranquillamente a 10mA o anche più.

Attenzione che è una sorgente non regolata: aumentando la corrente assorbita la tensione scende in modo significativo; inoltre spesso vi è un forte rumore sovrapposto e la tensione effettivamente erogata cambia da macchina a macchina.

Uart

Gli UART sono circuiti integrati che permettono di trasformare il segnale parallelo proveniente dal processore in segnale seriale. In genere vengono gestite dall'hardware tutte le funzioni a basso livello necessarie (inserimento dei bit di start e di stop, generazione o riconoscimento del bit di parità, generazione di interrupt) e spesso è presente un buffer FIFO che permette di ricevere ed inviare dati anche quando la CPU è impegnata. La descrizione di una UART va oltre gli scopi di questo tutorial: vi rimando quindi ad [altre risorse](#) disponibili in rete.

A volte, usando microcontrollori, questa funzione viene svolta interamente dal software anche se spesso in questo modo non si supera la velocità di 9600 bps e non è possibile un funzionamento full-duplex.