

Rice-Trasmissioni seriali

L'RS232 è un argomento che, con la progressiva scomparsa dai PC, in particolare dai notebook, potrebbe sembrare obsoleto, invece in certi ambienti è ancora molto attuale. Mi riferisco ad esempio all' "industrial" e più in generale ai **sistemi embedded**.

Il protocollo RS232, in genere 9600 8 N 1 è ancora la comunicazione più semplice, robusta e facile da implementare.

RS232: Convertitore TTL

<http://it.emcelettronica.com/rs232-convertitore-ttl>

RS232 e la logica TTL non sono compatibili. Per connettere un cavo RS232 ad un circuito TTL bisogna usare un convertitore in grado di presentare $\pm 12V$ (lato RS232) e 0-5V (lato microcontrollore).

Convertire i segnali **RS232** da $\pm 12V$ a 0-5V e viceversa è un'operazione abbastanza semplice, e fondamentale per poter connettere un cavo RS232 ad un microcontrollore.

Infatti molte schede elettroniche riportano in uscita i segnali TXD e RXD della UART. Questi però operano a 0V ed a 5V (on-off) quindi NON possono essere connessi direttamente all'RS232 del PC.

Devono essere convertiti (in alcuni casi anche isolati).

Della **RS232** abbiamo già parlato in vari articoli come:

"[Convertitore Seriale](#)", "[Cavo RS232](#)", "[DB9](#)", "[Convertitore USB RS232](#)" e anche del più famoso driver per RS232 abbiamo parlato: il MAX232.

Se avete poca familiarità con l'RS232 vi consiglio di andare a leggere prima questi articoli, per poter avere un minimo di nozioni sia sul pinout dell'RS232 che sul driver usato dall'RS232. Quest'ultimo infatti, il MAX232 appunto, è il cuore del convertitore RS232-TTL. All'interno contiene 2 trasmettitori e 2 ricevitori sia **RS232** che **TTL**.

Se proprio non sapete [cosa è l'RS232 allora vi rimando da Wikipedia](#)

Convertitore RS232 TTL

Abbiamo quindi appurato che bisogna convertire questi segnali, TX e RX almeno, ma se usiamo tutti i criteri dobbiamo convertire anche questi ultimi ovviamente.

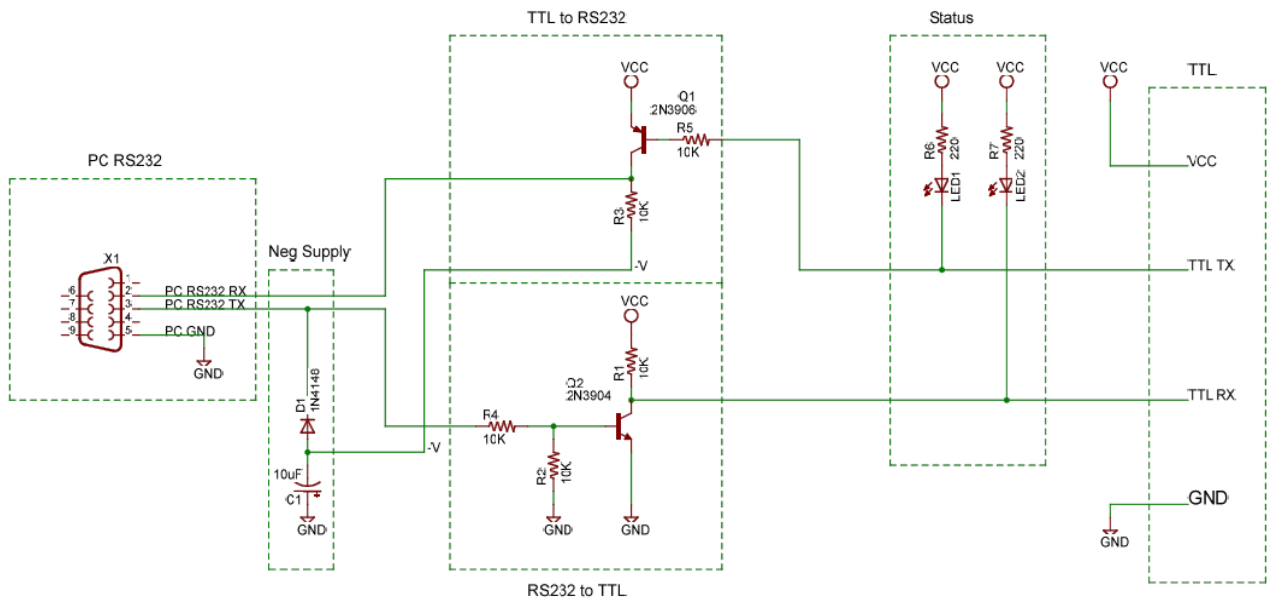
Per fare questo esistono 2 modi, il primo è di utilizzare un integrato che svolge tale compito come ad esempio il MAX232 (se utilizziamo solo TX e RX) oppure il MAX3238 se usiamo tutti i criteri (RTS CTS DSR DTR DTD RI).

Il secondo è quello di usare componenti elettronici per livellare a 0 e +5V i segnali **RS232** ed aumentare invece a +12 e -12 i segnali TTL.

Analizziamo le due soluzioni.



Una soluzione interessante è quella del [convertitore proposto da SparkFun](#) e questo circuito [convertitore RS232-TTL](#)



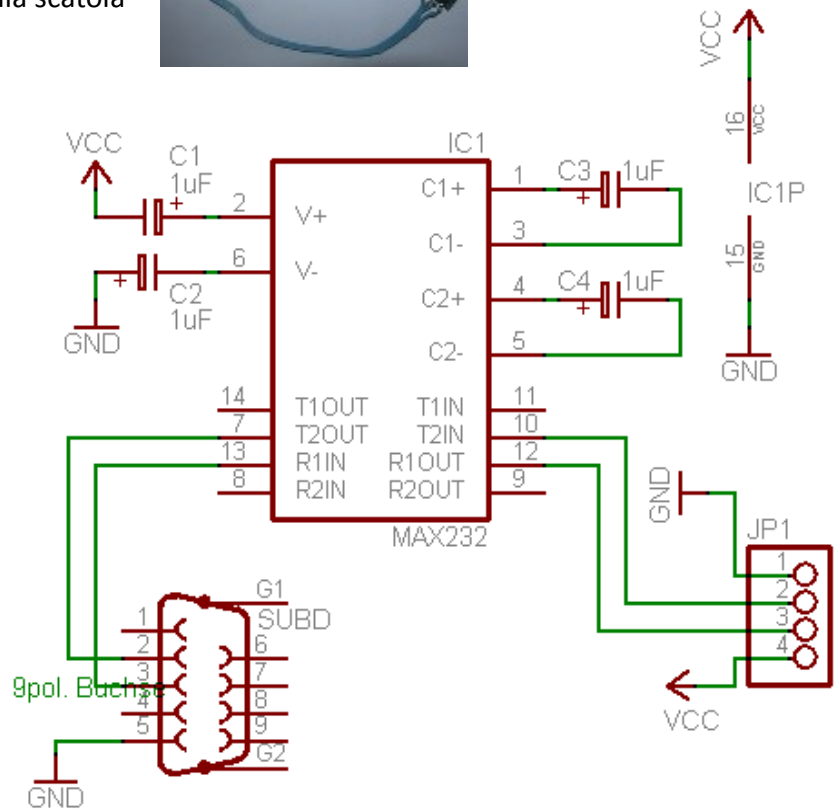
Ecco altri esempi di convertitori:

[RS232 / TTL Level converter](#)

[Semplice convertitore TTL RS232 nella scatola](#)

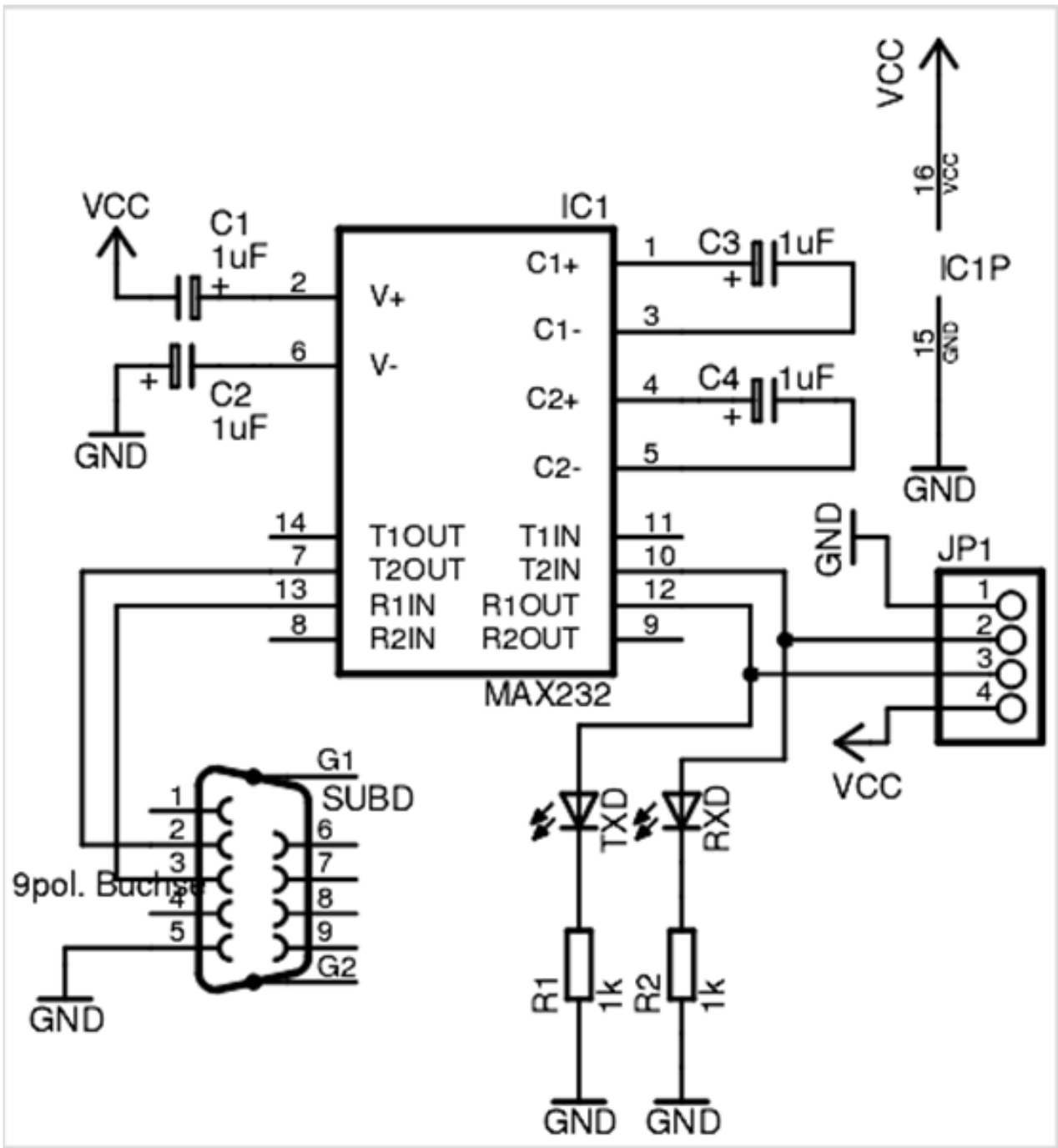


all'interno della scatola:



(in figura è illustrato un connettore seriale a 9 pin)

Interfaccia TTL – RS232 con visualizzatori led



Data sheets MAX 232 <http://www.ti.com/lit/ds/symlink/max232.pdf>

Alternativa

Per trasformare un livello logico TTL in RS-232 compatibile occorre un traslatore di livello in modo che il valore di tensione compreso tra +2V e +5V (1 logico TTL) si trasformi in -12V mentre un valore di tensione compreso tra 0 e 0.8V (0 logico TTL) si trasformi in +12V.

Vi sono integrati in grado di convertire un livello TTL in RS-232C e viceversa come, ad esempio, il 1448 e 1449 (fig.11).

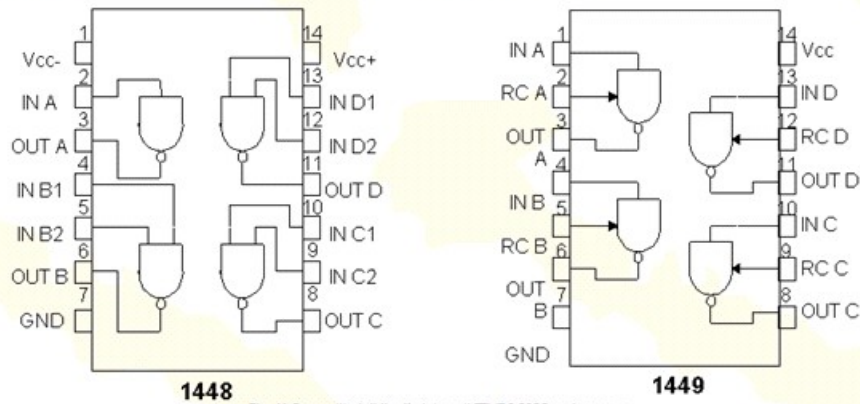


Fig. 11 Convertitori di livello integrati TTL/RS-232C e viceversa.

Il primo converte un livello TTL in RS-232C. Ha bisogno di doppia alimentazione (da $\pm 12V$ a $\pm 15V$) e contiene 4 dispositivi di tipo invertente. Il secondo, invece, converte un livello RS-232 in TTL. E' alimentato con una tensione di alimentazione a +5V e contiene 4 elementi invertenti. Il collegamento bidirezionale seriale tra due dispositivi che lavorano con segnali TTL si può effettuare come in fig.12.

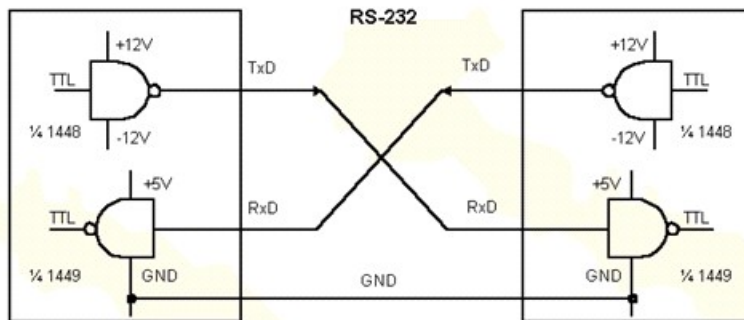


Fig. 12 Collegamento seriale con segnali nello standard RS-232C tra dispositivi TTL.

I valori di tensione assegnati allo standard RS-232C consentono di ottenere una elevata immunità al rumore che aumenta all'aumentare della "distanza" tra i due livelli logici.

<http://ospitiweb.indire.it/puglia/pug03/td/td2.htm>

Convertitore USB<->RS232

<http://it.emcelettronica.com/usb2usart-convertitore-usb-seriale-rs232>

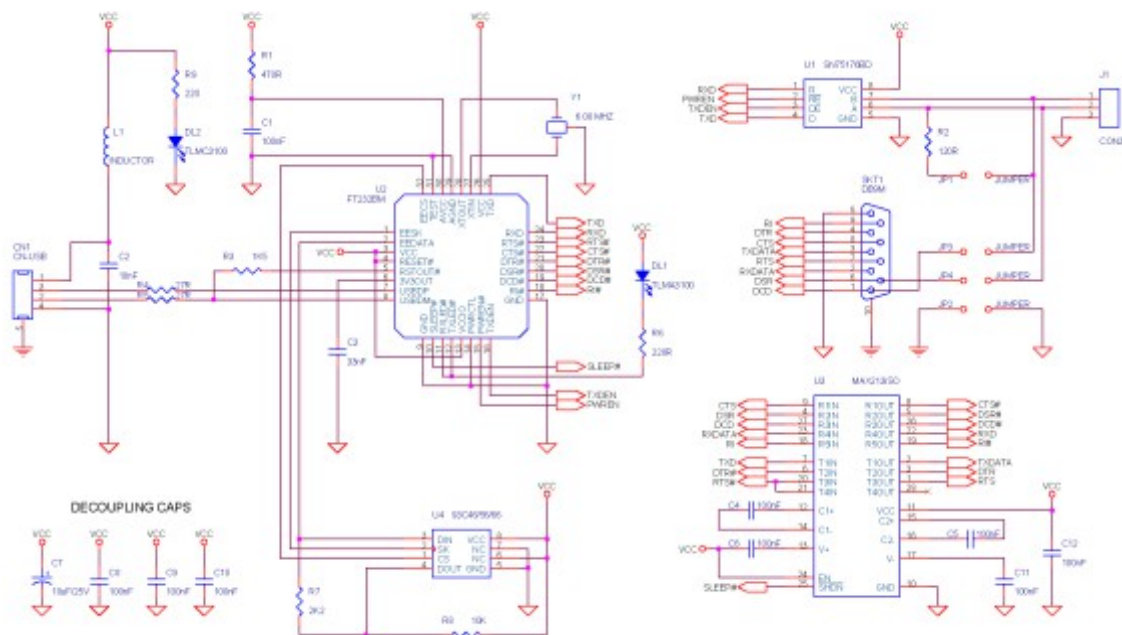
Un convertitore **USB<->RS232** ci offre la possibilità di interfacciare la nostra applicazione a dispositivi moderni (come ad esempio i nuovi portatili dai quali è ormai scomparsa la porta seriale COM) con la semplicità della seriale asincrona che ormai tutti ben conosciamo

Con questo progetto potete convertire i segnali **USB** in una porta **RS232** completa e viceversa, quindi potete sia dotare il Vostro Notebook di una porta seriale RS232, sia dotare la Vostra applicazione di una porta USB, semplicemente pilotandola con la **UART** del vostro microcontroller.

Ci sono varie possibilità per realizzare il progetto in oggetto, in commercio esistono vari convertitori (vedi link all'ultima pagina) sia singoli (i cosiddetti "bridge") che integrati nei microcontrollori, ma la scelta è caduta sull'integrato **FTDI**, semplice, di immediata realizzazione, non necessita programmazione ed i **driver USB** che realizzano una COM virtuale sul PC, sono disponibili gratuitamente.

Va ricordato, a scanso di equivoci, che il collegamento USB è lato **DEVICE**

Schema elettrico



Lo schema è molto semplice, praticamente fa tutto l'integrato **FT232RL**. Come particolarità segnale l'alimentazione VCC=5V presa direttamente dalla porta USB e la memoria 93C46 opzionale, per salvare un eventuale identificativo e quindi distinguere la scheda da altri dispositivi simili collegati alla porta USB.

Inoltre, in alto a destra ho inserito un **SN75176 (RS485 driver)** che tramite i ponticelli JP1-4 vi permette di modificare la scheda in convertitore USB-RS485

Risorse e link utili alla realizzazione del progetto USB2USART (convertitore USB <-> RS232)

[FTDI](#) --->> Produttore dell'integrato FT232 per la conversione dei segnali RS232 USB

[Driver FTDI](#) --->> Driver per il PC che permette il funzionamento dell'integrato FT232 realizzando una VCP (Virtual COM Port) sul Vostro PC sprovvisto di porta seriale COM

[ApplicationNote FTDI](#) --->> Molto utile anche per capire meglio la gestione delle alimentazioni USB a +5V

[Schematics FTDI](#) --->> Tutti gli schemi applicativi. Preziosa base di partenza

[VINCULUM](#) --->> Se dovete realizzare convertitori anche [USB HOST](#)

[Maxim](#) --->> Produttore dell'integrato [MAX213](#) e di svariati transceiver (convertitori di segnali RS232 - TTL)

[AnalogDevice](#) --->> Produttore dell'integrato [ADM213](#) e di svariati transceiver (convertitori di segnali RS232 - TTL)

USB-RS232 Serial Converter Driver Download <http://www.tri-plc.com/USB-RS232/drivers.htm>

USB to RS232 converter

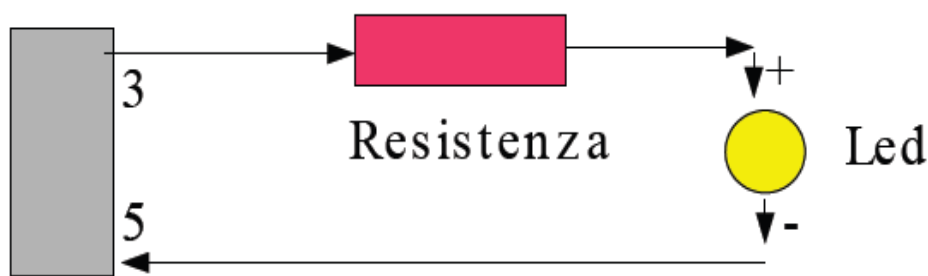


Interessante: per accendere un LED con porta RS-232

“Linux e l'interfaccia seriale RS-232 nell'interfacciamento e uso di dispositivi esterni con Linux”

<http://cslug.linux.it/cslug/linuxday/linuxday04/seminari/seminario-rs232.pdf>

dallo schema elettrico



RS-232

alla validazione

con semplice SW:

```
supercomputer :/ home/ predacon# cat nomefile > /dev/ ttyS0
```

La lunghezza del file implica il tempo di accensione del LED (più è lungo meglio si vede)

