

LaurTec

**Il Protocollo
RS232**

Autore : *Mauro Laurenti*

email: info.laurtec@gmail.com

ID: AN4003-IT

INFORMATIVA

Come prescritto dall'art. 1, comma 1, della legge 21 maggio 2004 n.128, l'autore avvisa di aver assolto, per la seguente opera dell'ingegno, a tutti gli obblighi della legge 22 Aprile del 1941 n. 633, sulla tutela del diritto d'autore.

Tutti i diritti di questa opera sono riservati. Ogni riproduzione ed ogni altra forma di diffusione al pubblico dell'opera, o parte di essa, senza un'autorizzazione scritta dell'autore, rappresenta una violazione della legge che tutela il diritto d'autore, in particolare non ne è consentito un utilizzo per trarne profitto.

La mancata osservanza della legge 22 Aprile del 1941 n. 633 è perseguibile con la reclusione o sanzione pecuniaria, come descritto al Titolo III, Capo III, Sezione II.

A norma dell'art. 70 è comunque consentito, per scopi di critica o discussione, il riassunto e la citazione, accompagnati dalla menzione del titolo dell'opera e dal nome dell'autore.

AVVERTENZE

I progetti presentati non hanno la certificazione CE, quindi non possono essere utilizzati per scopi commerciali nella Comunità Economica Europea.

Chiunque decida di far uso delle nozioni riportate nella seguente opera o decida di realizzare i circuiti proposti, è tenuto pertanto a prestare la massima attenzione in osservanza alle normative in vigore sulla sicurezza.

L'autore declina ogni responsabilità per eventuali danni causati a persone, animali o cose derivante dall'utilizzo diretto o indiretto del materiale, dei dispositivi o del software presentati nella seguente opera.

Si fa inoltre presente che quanto riportato viene fornito così com'è, a solo scopo didattico e formativo, senza garanzia alcuna della sua correttezza.

L'autore ringrazia anticipatamente per la segnalazione di ogni errore.

Tutti i marchi citati in quest'opera sono dei rispettivi proprietari.

Introduzione

Lo standard RS232 rappresenta un protocollo di comunicazione che permette senza un eccessivo hardware, la comunicazione tra due sistemi elettronici. Verranno spiegate le caratteristiche principali di questo protocollo illustrando anche l'hardware necessario per una applicazione pratica.

A fine lettura il lettore avrà le conoscenze e il materiale necessario per implementare un sistema di comunicazione tra dispositivi elettronici per mezzo del protocollo RS232.

Il problema della comunicazione

Un qualunque dispositivo hardware dopo aver elaborato un segnale in maniera analogica o digitale deve poter comunicare per mezzo di un'interfaccia di output i risultati ottenuti. In particolare la visualizzazione o un'ulteriore elaborazione possono avvenire per mezzo di un altro dispositivo che può trovarsi fisicamente distante dal primo. Sorge dunque il problema della trasmissione di dati.

Per breve tratto si tende a far prevalere una comunicazione parallela laddove effettivamente i dati da trasmettere siano digitali. Con trasmissione parallela si intende l'utilizzo di più linee dati¹ (fili) per trasmettere l'informazione. Si pensi ad esempio ai bus collegati al microprocessore, che permettono lo scambio tra quest'ultimo e le varie periferiche sulla scheda madre. Questo tipo di collegamento permette velocità particolarmente elevate. Nonostante le apparenti brevi distanze che separano il microprocessore dalle varie periferiche, è necessario attendere un tempo non nullo affinché tutti i bit che costituiscono l'informazione da trasmettere arrivino alla fine del bus. Per tale ed altre ragioni la trasmissione parallela non viene utilizzata per lunghe distanze. In realtà la massima distanza viene a dipendere anche dalla massima frequenza con la quale si vuole trasmettere l'informazione. A tal proposito si pensi alla porta parallela utilizzata per la stampante, che può raggiungere distanze dei 3m senza molti problemi. Questo non

¹ Si è specificato linee dati poiché generalmente nella trasmissione dell'informazione sono presenti anche linee di controllo necessarie per la corretta trasmissione delle informazioni.

sarebbe possibile se si volesse trasmettere le informazioni alla stampante alla stessa velocità con cui il microprocessore può leggere o scrivere nella memoria RAM (Random Access Memory).

Per lunghe distanze si tende ad utilizzare una trasmissione seriale. Per trasmissione seriale si intende che l'informazione può viaggiare su un'unica linea dati². In questa classe di trasmissioni rientra lo standard RS232.

Sia che la trasmissione sia parallela o seriale si parla di trasmissioni simplex, half-duplex e full-duplex. Con questi nomi si caratterizza ulteriormente il tipo di trasmissione. In particolare si ha una trasmissione simplex quando i dati viaggiano in un unico senso, ovvero si invia un'informazione senza preoccuparsi di nessuna risposta da parte del ricevitore. Con questo tipo di trasmissione non è dunque possibile sapere se l'informazione è effettivamente giunta a destinazione ed inoltre non c'è un vero scambio di informazioni³.

Per trasmissione half-duplex si intende una trasmissione che può avvenire in ambo i sensi ma non contemporaneamente.

Il fatto che la trasmissione non debba avvenire contemporaneamente può essere dovuta alla presenza di un solo cavo per la trasmissione dati o poiché le risorse hardware non permettono di ottenere una comunicazione bidirezionale nello stesso intervallo di tempo.

In ultimo vi è la trasmissione full-duplex che permette una comunicazione in ambo i sensi e in contemporanea. Questo non necessariamente significa che siano presenti due linee dati, una per la trasmissione e una per la ricezione, infatti con la cosiddetta moltiplicazione di frequenza, utilizzata per esempio in telefonia, è possibile trasmettere più informazioni in contemporanea su un unico cavo.

Due altri sottogruppi di trasmissioni che è possibile individuare sia tra quelle parallele che

² Utilizzando segnali LVDS (Low Voltage Differential Signals) o differenziali di altro tipo, è possibile effettuare trasmissioni parallele a distanze piuttosto lunghe ma le problematiche tecniche associate al ritardo dei bit (skew) non sono poche. In ogni modo si cerca di ridurre il numero di cavi necessari.

³ Un esempio di trasmissione simplex che sfrutta i canali radio è rappresentata dalla televisione e dalla radio stessa. Questo tende ad essere sempre meno vero con l'avvento dei decoder e delle TV via cavo.

quelle seriali è l'insieme delle trasmissioni sincrone e quelle asincrone. Per trasmissione sincrone si intende che assieme alle linee dati viene trasmesso anche un segnale di clock in modo da sincronizzare le varie fasi di trasmissione e ricezione dati⁴. Con trasmissioni sincrone, avendo a disposizione una base dei tempi, è possibile raggiungere velocità di trasmissione più alte che non nel caso asincrono.

Anche se letteralmente asincrono vuol dire senza sincronismo, non significa che questo non sia presente. Per asincrono si intende solo che il sincronismo non viene trasmesso per mezzo di un clock presente su una linea ausiliaria.

Il sincronismo è sempre necessario. Nelle trasmissioni asincrone, sia il trasmettitore che il ricevitore sono preimpostati a lavorare ad una certa velocità. Quando il ricevitore si accorge che il trasmettitore ha iniziato a trasmettere, conoscendo la sua velocità saprà interpretarlo.

Per l'invio di dati sono presenti molti tipi di trasmissioni sia seriali che parallele. Come anche in ogni lingua affinché i due interlocutori si possano capire è necessario che parlino la stessa lingua.

Ogni idioma ha una struttura grammaticale che permette una corretta trasmissione dell'informazione. In termini tecnici si parla di protocollo per intendere l'insieme di regole e specifiche che il trasmettitore e il ricevitore devono avere affinché si possano comprendere.

Caratteristiche del protocollo RS232

Lo standard che va sotto il nome di RS232 rappresenta un protocollo, ovvero insieme di regole, per la trasmissione seriale di dati in formato digitale, questo avviene in maniera asincrona e in modalità full-duplex. Il protocollo impone dei vincoli sia di natura meccanica che elettrica, oltre che al formato dei dati stessi.

Il connettore utilizzato nei personal computer è di due tipi, il DB25 maschio e il DB9 maschio. Ai suoi albori il connettore utilizzato era il DB25 rimpiazzato con il DB9. L'RS232 è legato al DB25 mentre RS232C è legato al DB9, per i segnali che verranno presentati non c'è però alcuna differenza.

Per tale ragione alcuni modem esterni hanno un cavo che termina sia con un DB25 femmina che un DB9 femmina.

Sulla sinistra di Figura 1 sono riportati i due connettori DB9 delle porte seriali generalmente presenti sul retro del computer. I nomi che il sistema operativo dà alle seguenti porte è COMM1 e COMM2. Eventuali altre porte vengono chiamate sempre COMM ma con numerazione crescente.



Figura 1: Porte seriali sul retro del PC

In Tabella 1 sono riportate in funzione della numerazione, che è visibile sulla parte frontale del connettore stesso, il significato dei vari pin, sia nel caso di connettore DB25 che connettore DB9.

DB25	DB9	in/out	Sigla
2	3	O	TxD Trasmissione Dati
3	2	I	RxD Ricezione Dati
4	7	O	RTS Request To Send
5	8	I	CTS Clear To Send
20	4	O	DTR Data Terminal Ready
6	6	I	DSR Data Set Ready
22	9	I	RI Ring Indicator
8	1	I	DCD Data Carrier Detect
7	5	-	GND Ground (massa)

Tabella 1: Corrispondenza pin-funzione

E' possibile subito osservare che sono presenti più di due linee. Questo è dovuto al fatto che nel protocollo RS232 sono presenti anche regole per handshaking (stretta di mano) ovvero linee che permettono al trasmettitore e al ricevitore di avere informazioni dirette l'uno dell'altro. I modem fanno uso di queste linee,

⁴ Si veda per esempio il Tutorial sul Bus I²C.

ma si può ottenere una trasmissione full-duplex con il solo utilizzo delle linee TxD, RxD e GND.

Alcune volte le altre linee di output vengono utilizzate per prelevare l'alimentazione del dispositivo collegato al computer. Questo viene fatto per esempio dai mouse seriali⁵. Bisogna comunque stare attenti e placare ogni entusiasmo poiché questo non significa che si possa alimentare ogni circuito. Le correnti che è possibile prelevare dalle linee di output non sono generalmente superiori ad un paio di mA. La corrente che è possibile prelevare dipende poi da computer a computer.

Vediamo ora a cosa servono le altre linee non di trasmissione.

La linea DTR viene attivata quando il PC è pronto per il collegamento con una periferica esterna.

La periferica esterna segnala al PC che è pronta al collegamento attivando DSR.

La linea RTS viene posta alta dal PC quando è pronto per la trasmissione dati verso la periferica esterna.

La linea CTS viene attivata in risposta al segnale RTS quando la periferica è pronta per ricevere dati. Prima di questo evento il PC non inizia la trasmissione.

Il segnale DCD è pensato appositamente per i modem. Questo viene attivato quando il modem collegato al nostro PC rileva la trasmissione dell'altro modem collegato...chissà dove!

Il segnale RI viene attivato dal modem quando rileva il trillo di una chiamata telefonica.

Quanto fin ora descritto rappresenta la parte più grammaticale del protocollo, ovvero la descrizione dei segnali e il loro significato in una trasmissione. In realtà le linee che si sono appena descritte sono controllabili anche via software il che permette di utilizzarle anche con un significato differente creando così un proprio

protocollo.

Quello che però non potremmo variare sono le specifiche elettriche a cui devono soddisfare i segnali utilizzati per la trasmissione dei dati.

Questo discende semplicemente dal fatto che l'hardware per il protocollo RS232 è progettato per soddisfare queste specifiche.

In particolare i livelli di tensione sono riportati in Tabella 2.

Livello logico	Trasmittitore	Ricevitore
0	da +5V a +15V	da +3V a +25V
1	da -5V a -15V	da -3V a -25V
non definito	da -3V a +3V	

Tabella 2: Corrispondenza livelli logici-tensioni

Come è possibile osservare l'intervallo di tensione che caratterizza ogni livello logico è molto ampio.

Confrontando i livelli di tensione di questo protocollo con lo standard TTL in cui i livelli di tensione sono compresi tra 0V (0 logico) e 5V (1 logico) è possibile notare che un livello di tensione alto corrisponde ad uno zero logico.

Nonostante l'ampiezza dell'intervallo di tensioni ammissibili nel protocollo RS232, generalmente si usano livelli di tensione compresi tra $\pm 12V$.

I livelli di tensione più bassi vengono utilizzati al fine di limitare l'irradiazione di campi elettromagnetici verso l'esterno. Questi potrebbero dare fastidio agli altri cavi che viaggiano in prossimità del nostro cavo RS232. Al fine di limitare interferenze elettromagnetiche con altri cavi è imposto un limite anche per lo slew-rate a $30V/\mu s$.

Questo parametro quantifica la rapidità con la quale avviene la variazione dei livelli logici ovvero della tensione⁶.

Per ottenere una migliore immunità al rumore proveniente dall'ambiente esterno si tende ad usare i livelli di tensione più alti. Tra le altre caratteristiche tecniche che bisogna ricordare c'è quella relativa alle impedenze di ingresso e di uscita delle varie porte di trasmissione e ricezione. In particolare l'impedenza d'ingresso deve essere compresa tra

⁵ Anche i mouse USB prendono l'alimentazione direttamente dal computer, ma in questo caso la presa USB prevede di suo le linee di alimentazione.

⁶ Lo slew rate rappresenta la pendenza del fronte di salita o di discesa relativo alle transizioni di tensione associate ad un cambio di livello logico.

i $3K\Omega$ e i $7K\Omega$, mentre l'impedenza di uscita deve essere maggiore di 300Ω . Queste ultime caratteristiche possono interessare meno poiché si suppone che l'hardware che si compra per le trasmissioni RS232 sia realmente idoneo per l'RS232!

In ultimo prima di fare un esempio di trasmissione utilizzando le linee TxD e RxD, è bene ricordare alcune modalità con cui è possibile accorgersi di un errore.

Supponiamo di dover trasmettere il byte 10010011, sia che si trasmetta serialmente un bit alla volta sia che si trasmetta tutti i bit in parallelo, ci sarà possibilità che qualche bit possa cambiare di livello prima di giungere a destinazione a causa del rumore esterno. L'errore, di suo, non è eliminabile, ma si può limitare la possibilità che questo si verifichi.

Per limitare il verificarsi di un errore si provvede alla schermatura del cavo e al suo intreccio in modo da limitare gli effetti nocivi di eventuali campi elettromagnetici che si possono accoppiare con il nostro cavo. Nonostante questo accorgimento l'errore può avvenire.

Subentrano a questo punto le tecniche per la correzione. Queste presumono in primo luogo che sia possibile accorgersi dell'errore. Per fare questo è necessario aggiungere, come si dice in gergo, della ridondanza, ovvero degli ulteriori bit che permettano di fare dei controlli sulle informazioni ricevute.

Maggiore è l'informazione ridondante che viene aggiunta, maggiore è la probabilità di accorgersi e correggere l'errore. Con questo si vuole dire che nonostante si faccia uso di ridondanza, l'errore può ancora una volta solo essere limitato.

Il protocollo RS232 prevede la possibilità di utilizzare le seguenti tecniche per la rilevazione degli errori :

- parità pari
- parità dispari
- mark
- space

In particolare è possibile anche non abilitare nessuno controllo. Questi controlli permettono solo di verificare se i dati in arrivo sono corretti o meno, non permettendo alcuna correzione da parte del ricevitore.

Se il ricevitore si dovesse accorgere di un eventuale errore l'unico modo per correggerlo è farselo ritrasmettere. Vediamo in dettaglio come funziona la parità pari e la parità dispari che sono quelle più comunemente utilizzate.

Per parità pari si intende che il dato ricevuto deve possedere sempre un numero pari di 1.

Quindi se si vuole trasmettere il byte 11001000 affinché sia verificata la parità pari è necessario aggiungere un 1 che rappresenterà la ridondanza aggiunta. Dunque la nuova sequenza realmente trasmessa sarà 110010001.

Se si volesse trasmettere il byte 10001000 bisognerebbe aggiungere uno 0 e si otterrebbe dunque 100010000.

La parità dispari consiste nell'inserire un eventuale 1 al fine di ottenere un numero dispari di 1.

Da quanto esposto si capisce che se un disturbo esterno dovesse far variare contemporaneamente un numero pari di bit né la parità pari né la parità dispari si accorgerebbero di nulla.

Le modalità mark e space consistono rispettivamente nell'aggiungere in maniera fissa un 1 o uno zero.

Se nessuna delle sopra citate metodologie di rilevazione di errori viene abilitata non viene aggiunto nessun bit di ridondanza.

Vediamo ora un esempio di trasmissione che fa uso del protocollo RS232. Supponiamo di voler trasmettere il byte 10001100. In questo caso non si considera abilitata nessuna modalità di rilevazione dell'errore. Secondo il protocollo RS232 la sequenza verrà trasmessa a partire dal bit meno significativo⁷ (bit a destra) fino al bit più significativo⁸. Si capisce dunque che eventuali bit di parità essendo messi in coda al byte verranno posti al fianco del bit più significativo.

Dal momento che il protocollo RS232 è di tipo asincrono è necessaria qualche strategia per sincronizzare la fase di trasmissione con quella di ricezione. La tecnica adottata è quella di rimanere a livello logico 1 (tensione inferiore⁹ a -5V) per poi trasmettere un bit di start oltre alla normale sequenza. Se collegassimo un

⁷ Noto anche LSB (Least Significant Bit)

⁸ Noto anche come MSB (Main Significant Bit)

⁹ Si ricorda che -6 è inferiore a -5.

oscilloscopio alla linea di trasmissione potremmo vedere la sequenza d'invio del byte. In Figura 2 è riportato quello che verrebbe visualizzato.

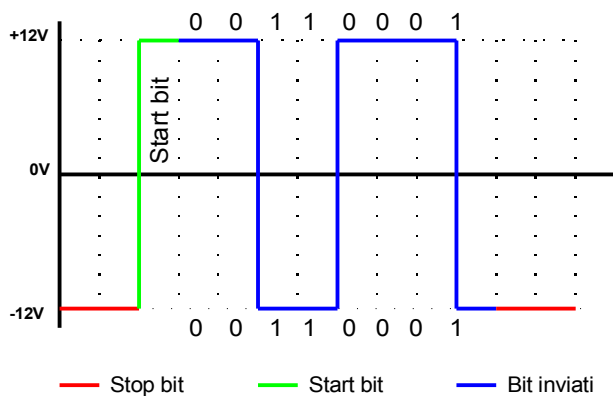


Figura 2: Andamento della tensione sulla linea TxD

E' possibile osservare che prima dell'inizio della trasmissione, la linea è posizionata a -12V. Quando la trasmissione inizia il livello passa a +12V per la durata di un bit. L'ampiezza temporale di un bit viene a dipendere dalla velocità di trasmissione. Al rilevamento del bit di start il ricevitore inizia a leggere il segnale. La lettura del valore di un bit viene effettuata nel mezzo poiché questo è il punto più probabile per trovare un'informazione corretta.

Dopo il bit di start la tensione rimane alta per altri due bit per poi passare per altri due bit a -12V. Successivamente ripassa a +12V per la durata di tre bit per poi passare nuovamente a -12V per la durata di un bit. Questo bit rappresenta l'ultimo della sequenza voluta.

Si può osservare che il livello di tensione rimane fisso a -12V. Qualora l'ultimo bit fosse stato uno zero logico il livello sarebbe comunque¹⁰ sceso a -12V. Questa situazione rappresenta il punto di partenza per una nuova sequenza ovvero per un nuovo bit di start. Il passaggio finale a -12V viene chiamato bit di stop. In particolare è possibile impostare il numero di bit di stop a 1, 1.5 o 2. L'eventuale bit di parità avrebbe preceduto il o i bit di stop.

L'invio e la ricezione di una sequenza di byte

è un operazione molto pesante se gestita interamente via software. I PC hanno hardware dedicato per la gestione delle trasmissioni che prende il nome di USART¹¹ (Universal Synchronous Asynchronous Receiver Transmitter).

Questo hardware si preoccupa anche dell'inserimento degli eventuali bit di parità. La versatilità di queste periferiche permette inoltre di impostare la velocità di trasmissione che sarà uguale a quella di ricezione e il numero di bit della parola da inviare.

La lunghezza della parola che è possibile inviare può essere di 6,7,8,9 bit¹². Le velocità che possono essere impostate variano a seconda dell'USART, in particolare il protocollo RS232 prevede una frequenza massima fino a 20Kb/s (20000 bit per secondo) mentre molti PC prevedono comunque frequenze fino a 115Kb/s.

Collegamento Null-Modem

Per mezzo della porta seriale RS232 è possibile collegare direttamente due PC. Questo tipo di collegamento prende il nome di null-modem e viene per esempio utilizzato dal programma Hyperlink di Windows® per permettere lo scambio di file tra due computer.

Questo tipo di collegamento non rappresenta la soluzione più veloce ma può risultare pratica nel caso in cui uno dei due computer risulti datato e non abbia una scheda di rete.

Tale collegamento viene inoltre sfruttato in alcuni giochi di strategia in modalità multiplayer; generalmente la modalità null-modem rappresenta solo una delle varie scelte di connessione con l'altro giocatore.

Per effettuare la connessione null-modem non bisogna semplicemente collegare pin to pin le due porte seriali dei PC, perché se così si facesse non verrebbe trasmesso nulla.

In particolare dovrebbe risultare intuitivo che la linea di trasmissione di un PC andrà alla linea di ricezione dell'altro PC e viceversa.

Già con questi due soli collegamenti si è ottenuto un cavo null-modem funzionante.

Un miglioramento può essere ottenuto

¹⁰ Il livello scende automaticamente a -12V se la trasmissione è gestita da hardware dedicato. Se la trasmissione è implementata per mezzo di un software bisogna abbassare il livello a -12V manualmente.

¹¹ Alcune volte è presente solo l'UART

¹² Non tutte le USART supportano tutte le lunghezze. La lunghezza di 8 bit è generalmente quella standard poiché coincide con la dimensione dei caratteri ASCII.

collegando per ogni connettore la linea RTS con la linea CTS mentre la linea DTR con la linea DSR e DCD. Questi collegamenti vanno fatti sul singolo connettore e non tra i due connettori¹³.

Per chi comunque non volesse realizzare tale cavo è possibile comprarne di già fatti. Il costo non è elevato e in mancanza di tempo o voglia, sono la soluzione migliore!

Costruiamo la nostra interfaccia RS232

Supponiamo di dover realizzare un sistema

(bibliografia).

Questi microcontrollori lavorano generalmente fino a 5V che rappresenta appunto la tensione tipica dei circuiti digitali, mentre la tensione richiesta dal protocollo RS232 ha livelli differenti.

Qui ci viene in aiuto il MAX232 che permette di ottenere i livelli giusti di tensione necessari per il protocollo RS232 partendo da soli 5V. La tecnica utilizzata consiste nell'utilizzare una pompa di carica. Questo circuito per mezzo di capacità esterne, giocando sul collegare e scollegare le capacità in modo

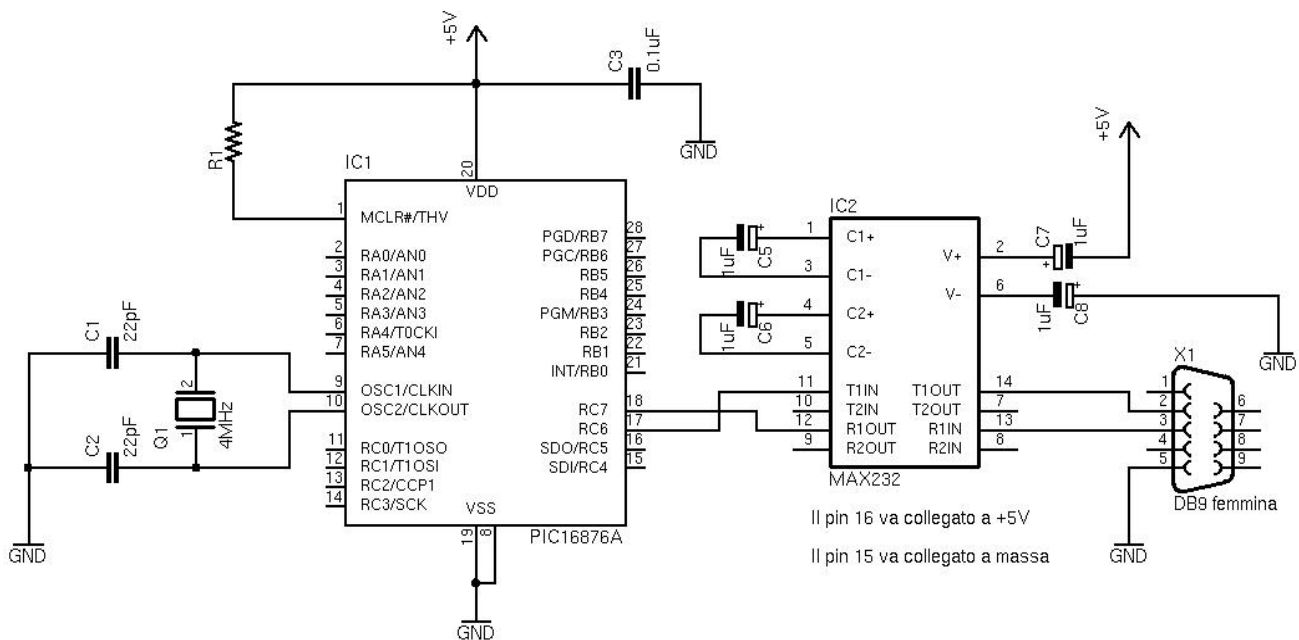


Figura 3: Schema elettrico per un collegamento tra il PIC16F876A e il PC

che debba scambiare informazioni con il PC per mezzo della porta seriale RS232.

Dal momento che molti microcontrollori contengono al loro interno delle USART e i costi rispetto a quelle che non la contengono sono inesistenti, è bene scegliere un microcontrollore che abbia un'USART. Per esempio si può utilizzare il microcontrollore della Microchip PIC16F876A (si veda

opportuno, permette di ottenere i livelli di tensioni richiesti.

In Figura 3 è riportato lo schema elettrico che permette una comunicazione full-duplex con il PC.

In particolare il resistore R1 di 10KΩ serve per resettare automaticamente il PIC. Il quarzo di 4MHz non può essere sostituito con un qualsiasi altro valore. Come riportato sul data sheet i valori consentiti oltre a 4MHz sono : 3.6864MHz, 10MHz, 16MHz, 20MHz.

Questo apparente limite discende dal fatto che il

¹³ Sono presenti anche altre versioni di null-modem in cui si intrecciano anche quest'ultime linee.

protocollo RS232 permette la trasmissione delle informazioni scegliendo tra un numero di frequenze limitato.

Bibliografia

www.LaurTec.com : sito di elettronica dove poter scaricare gli altri articoli menzionati, aggiornamenti e progetti.

www.microchip.com : sito della casa produttrice del PIC16F876A.

www.maxim-ic.com : sito della casa produttrice del MAX232.