COMUNICAZIONE IN SERIE

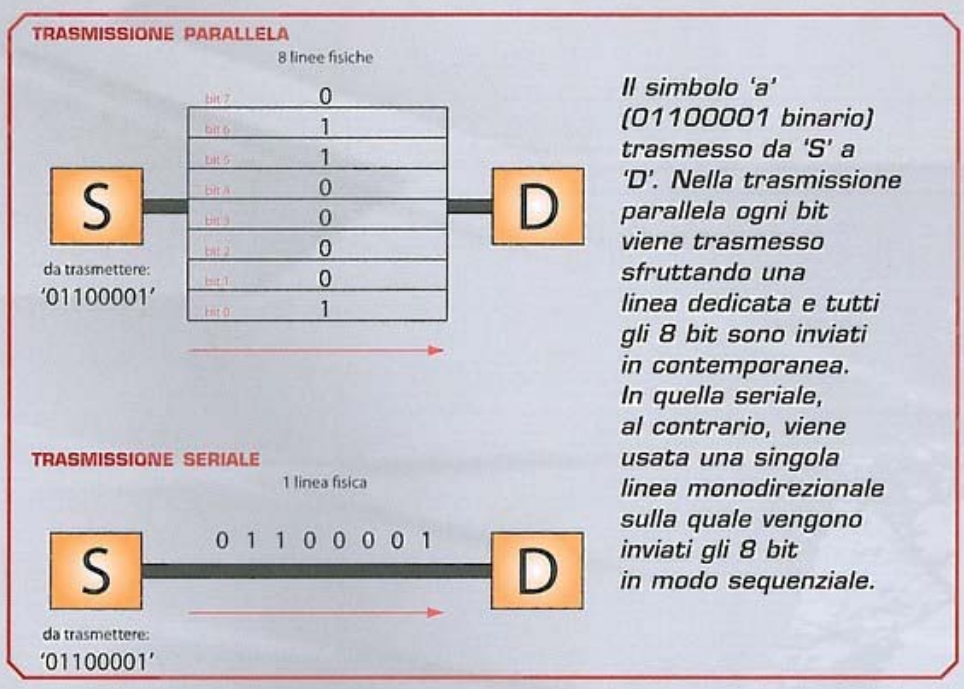
Con questo Workshop iniziamo a studiare la modalità di comunicazione seriale dei PIC, che ci permetterà di interfacciare i microcontrollori al nostro personal computer.

Un connettore DB-9, comunemente utilizzato per la comunicazione seriale all'interno dei personal computer.

La possibilità di **comunicare** con altri dispositivi è fondamentale per i sistemi digitali complessi, ed è così che nel corso degli anni dell'evoluzione tecnologica vi è stato un continuo susseguirsi di nuove tecnologie orientate ad assolvere questo compito. Poiché in questa sede è impossibile analizzare tutte le problematiche legate alle comunicazioni digitali, ci limiteremo a introdurre solo alcune delle loro caratteristiche.

COMUNICAZIONI IN SERIE E IN PARALLELO ›››

Comunicare significa, innanzitutto, **trasmettere informazioni tra due o più dispositivi.** Chiameremo **'sorgente'** il sistema che produce l'informazione, mentre al ricevente daremo il nome di **'destinatario'**, e poiché il 'colloquio' avviene tra dispositivi digitali, **l'informazione sarà scambiata in forma di 'gruppi' di bit,** fatti transitare attraverso un apposito canale. Proprio a partire dalla 'struttura' di questo canale possiamo classificare la tecnologia di comunicazione secondo due distinte modalità operative dette **'comunicazione parallela'** e **'comunicazione seriale'**. Per capire più semplicemente di cosa stiamo parlando, ipotizziamo un sistema composto da **due dispositivi** identificati attraverso i simboli **'S'** (dispositivo sorgente) e **'D'** (dispositivo destinatario). Ipotizziamo, inoltre, che **entrambi operino manipolando simboli di 8 bit** (esattamente come avviene con il PIC 16F628) e che **'S' voglia inviare a 'D' il carattere 'a'**, rappresentato nella codifica ASCII estesa a 8 bit dal valore binario **'01100001'** ('97' decimale). Data questa premessa, vediamo in cosa differiscono i due tipi di trasmissione. Il principio alla base della **trasmissione parallela** è mostrato nella parte superiore dello schema di pag. 2. In esso puoi osservare come gli 8 bit costituenti il carattere da spedire vengano trasmessi dalla sorgente **tutti in contemporanea su 8 distinte linee digitali.** Un esempio di comunicazione parallela si ha, appunto, nelle porte parallele dei personal computer, che fino a qualche tempo fa erano impiegate per connettere le stampanti. Nella trasmissione seriale, al contrario, **tutti i bit viaggiano in maniera sequenziale attraverso un'unica linea fisica di comunicazione monodirezionale** (disegno nella parte inferiore dello schema di pagina 2). Ma quale delle due tecnologie dà risultati migliori? Se da un lato **le comunicazioni parallele sono molto più**

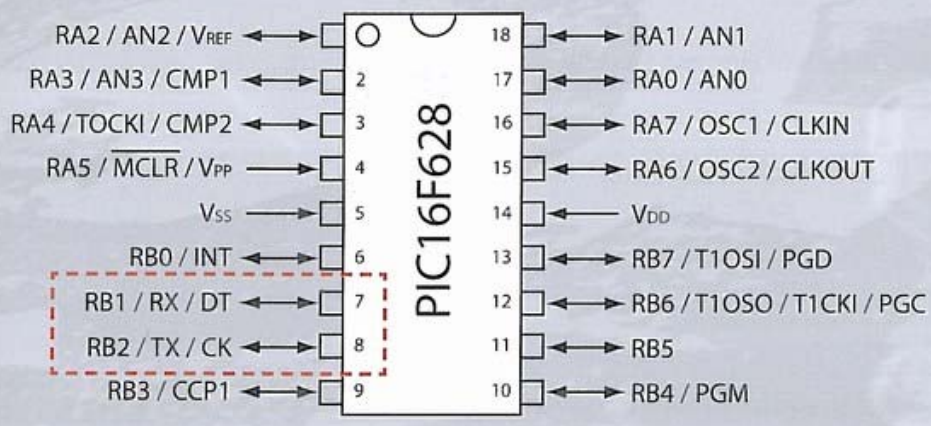


Il simbolo 'a' (01100001 binario) trasmesso da 'S' a 'D'. Nella trasmissione parallela ogni bit viene trasmesso sfruttando una linea dedicata e tutti gli 8 bit sono inviati in contemporanea. In quella seriale, al contrario, viene usata una singola linea monodirezionale sulla quale vengono inviati gli 8 bit in modo sequenziale.

semplici da implementare a livello 'architetturale' rispetto a quelle seriali, dall'altro **la comunicazione seriale offre una maggiore capacità di resistere ai disturbi e alle alte velocità di trasmissione.** Proprio per queste ragioni, la maggior parte degli standard di trasmissione ad alte prestazioni attualmente sul mercato impiega principi di tipo seriale (la tecnologia USB, ad esempio, deve il suo nome alla contrazione dei termini *Universal Serial Bus*, ossia 'Bus Seriale Universale').

LA TRASMISSIONE SERIALE E I PIC

Veniamo ora ai PIC. Quanto abbiamo visto nel paragrafo precedente ci ha fornito una descrizione introduttiva del significato di trasmissione in serie e in parallelo, ma non abbiamo ancora fatto fronte a come tali metodologie possano essere impiegate nel mondo dei PIC (faremo riferimento, come sempre, al modello 16F628 e al linguaggio di programmazione mikroC). Una caratteristica molto interessante, comune praticamente a tutti i microcontrollori in commercio, è la loro capacità di supportare in modo nativo uno o più sistemi di trasmissione dei dati. Iniziamo dalla **trasmissione parallela:** in sé, **il 16F628 non include un sottosistema dedicato alla gestione di questa modalità di comunicazione.** Tuttavia, è possibile supplire a questa mancanza sviluppando firmware specifici in grado di 'emulare' i principi alla base di questa metodologia di trasmissione. Il discorso, invece, diventa più interessante a livello sperimentale quando si parla di comunicazione seriale. **Il microcontrollore 16F628 integra, infatti, un modulo USART** (contrazione di *Universal Synchronous Asynchronous Receiver Transmitter*, ossia 'RiceTrasmettitore Sincrono Asincrono Universale'). Tale modulo non è altro che un dispositivo hardware in grado di **implementare elettronicamente** (senza controlli software aggiuntivi) **i meccanismi di comunicazione seriale standard partendo dai flussi di dati paralleli interni al PIC.** Proprio il modulo USART ci consentirà di far comunicare il PIC con il nostro PC. Per questioni di 'utilità sperimentale' non ci addentreremo nei dettagli dei protocolli di comunicazione e di come avviene lo scambio di informazioni sul piano 'elettrico'; al contrario, ci soffermeremo sulle modalità pratiche con cui far dialogare il microcontrollore con il nostro computer. Innanzitutto identifichiamo **due particolari pin del PIC, ossia i piedini numero 7 e 8.** Come puoi osservare dallo schema della piedinatura riproposto nella prossima pagina, questi due piedini includono nel loro nome le sigle **TX** (pin 8) e **RX** (pin 7), sigle utilizzate come abbreviazione dei termini *Transmit* e *Receive* (ossia 'trasmetti' e 'ricevi'). La funzione di questi due piedini, se il modulo USART è stato configurato, è molto semplice ed è quella di mettere a disposizione del progettista **due linee di comunicazione seriale, atte a trasmettere (TX) e ricevere (RX) dati.** I collegamenti seriali che sperimenteremo nei nostri Workshop saranno principalmente di tipo 'punto-punto' (ossia collegamenti



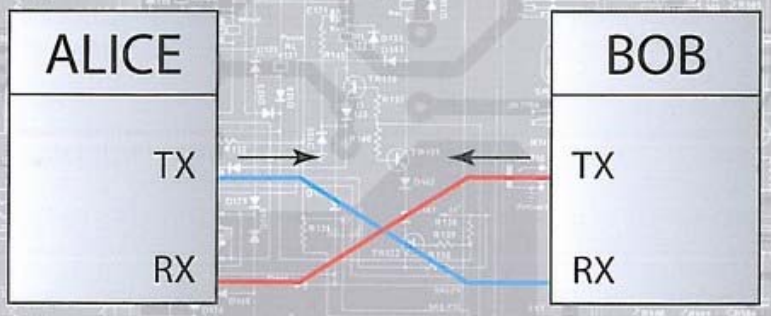
I pin 7 e 8 del PIC 16F628, come puoi vedere dallo schema, riportano all'interno del loro nome la dicitura RX e TX. Tali pin, infatti, possono essere configurati come linee di trasmissione (TX) e ricezione (RX) attraverso le quali è possibile, tra l'altro, far comunicare il PIC con i comuni personal computer dotati di porta seriale RS-232.

diretti tra due soli dispositivi, come un PIC e un PC o un PIC e un PIC) ottenuti con connessioni 'incrociate' (vedi box sotto). In realtà, però, esistono numerosissimi standard di trasmissione seriale in grado di fornire configurazioni topologiche differenti dalla struttura 'punto-punto', tra i quali troviamo, ad esempio, l'I²C che è molto utilizzato nel campo dell'elettronica digitale e della robotica amatoriale (il 16F628 non è dotato di un circuito interno dedicato alla comunicazione I²C, ma può 'emulare' questo protocollo 'via software'). Molti PIC, invece, includono moduli hardware che supportano in modo nativo anche questo tipo di standard.

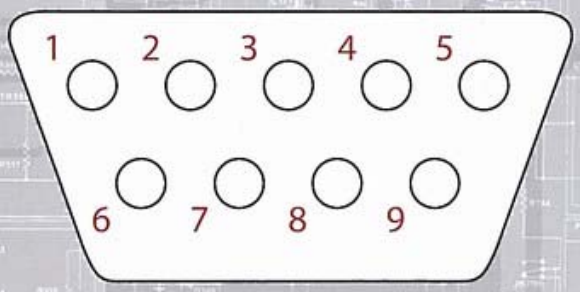
LA PORTA SERIALE DEI PERSONAL COMPUTER ›››

Se ora è chiaro cosa possiamo aspettarci dal lato dei PIC, non sappiamo ancora nulla in merito alla **struttura dei connettori seriali dei PC** (faremo riferimento alle porte seriali 'classiche', realizzate sulla base del cosiddetto **standard 'RS-232'**). Le porte seriali, spesso identificate anche con la sigla 'COM' e presenti nella maggior parte dei desktop (mentre sono state praticamente eliminate dalla fascia laptop e soppiantate dalle più moderne USB), sono interfacce con cui tutti coloro

CONNETTERE DUE DISPOSITIVI SERIALI IN MODO INCROCIATO ›››



I PIC dotati di modulo USART sono in grado di gestire la comunicazione seriale attraverso due distinte linee digitali monodirezionali. La prima, identificata con la sigla TX (Trasmit) gestisce il flusso di dati in uscita, mentre sulla seconda, detta RX (Receive), viaggiano i dati in ingresso al PIC. Il collegamento punto-punto tra due dispositivi USART avviene 'incrociando' (spesso in gergo di dice anche 'crossando', italianizzazione del verbo inglese *to cross*, ossia 'incrociare') le rispettive linee seriali. In questa maniera, se ipotizziamo di chiamare 'Alice' e 'Bob' due dispositivi che necessitano di scambiare dati tramite connessione seriale, la linea TX di 'Alice' dovrà essere collegata alla linea RX di 'Bob' (in modo che Bob riceva ciò che Alice trasmette) e, analogamente, la linea TX di 'Bob' dovrà essere connessa alla RX di 'Alice' (in modo che Alice riceva ciò che Bob trasmette). Lo schema soprastante mostra l'incrocio dei canali di trasmissione dei due dispositivi.



IL CONNETTORE DB-9 DELLA PORTA SERIALE ›››

In questo box ti presentiamo la struttura del connettore DB-9 impiegato per le porte seriali dei PC. Lo schema mostra la disposizione dei pin, a partire dalla vista frontale del connettore. Nella tabella, invece, trovi l'elenco delle linee con la loro funzione specifica. Nei nostri esperimenti, utilizzeremo solo i tre pin evidenziati in rosso.

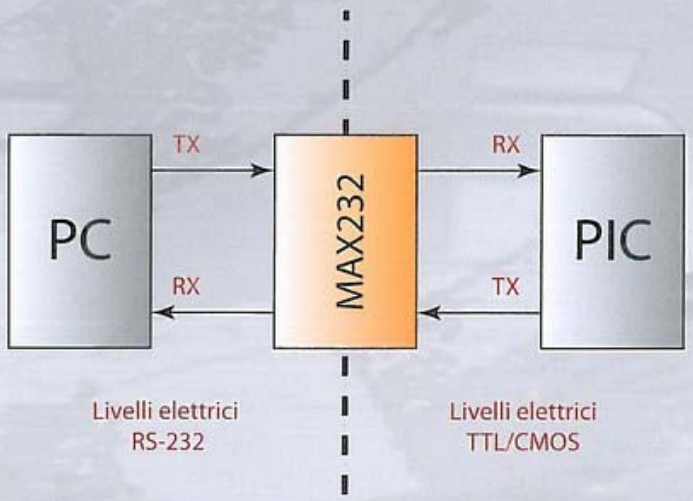
NUMERO PIN	NOME PIN	FUNZIONE
1	DCD	Data Carrier Detect: indica la presenza di una portante dati valida.
2	RX	Receive: linea seriale di ricezione dei dati
3	TX	Transmit: linea seriale di trasmissione dei dati
4	DTR	Data Terminal Ready: segnala al dispositivo remoto che il dispositivo è attivo e pronto a trasmettere.
5	GND	Ground: massa di riferimento
6	DSR	Data Set Ready: segnala al dispositivo remoto che l'unità è collegata e pronta a ricevere
7	RTS	Request To Send: permette al trasmittente di comunicare al ricevente che vi sono dati da inviare.
8	CLS	Clear To Send: comunica al trasmittente che l'unità è pronta a ricevere i dati in partenza.
9	RI	Ring Indicator: è utilizzata dai modem telefonici per segnalare una chiamata in arrivo (come se fosse lo squillo di un comune telefono).

che usano il computer da alcuni anni hanno potuto confrontarsi. Storicamente, infatti, mouse e modem erano collegati proprio per mezzo di questa tecnologia ed era diffusa anche la realizzazione di connessioni 'punto-punto' a bassa velocità tra computer con cavi 'null modem' (non sono altro che cavi seriali 'incrociati'). Ma **come è fatta fisicamente una porta seriale?** Il connettore più diffuso per le porte COM è il **DB-9** (vedi box sopra), uno spinotto di **forma trapezoidale a 9 pin,** normalmente presente in forma 'maschio' sui computer e come 'femmina' sulle periferiche (nella foto in apertura di pagina 1 puoi vedere un connettore femmina da circuito stampato); tuttavia, in passato sono stati impiegate anche porte basate sul DB-25 (simile al DB-9 nella forma, ma dotato di 25 pin). Dei 9 piedini, **solo tre saranno indispensabili** per i nostri esperimenti: quello di **massa (pin n. 5),** il pin di **trasmissione TX (pin n. 3)** e quello di **ricezione RX (pin n. 2).** La funzione dei pin TX e RX della porta seriale è identica a quella dei pin del PIC. Il pin di massa, invece, deve essere messo in comune tra il circuito elettronico

del microcontrollore e il PC. In questo modo si stabilisce un potenziale di riferimento comune, che permette ai dispositivi coinvolti di rilevare correttamente le tensioni. Teoricamente, quindi, potrebbe sembrare che per far comunicare un computer e un PIC sia sufficiente applicare il collegamento incrociato visto nel box in basso di pagina 3 tra le rispettive linee di trasmissione seriale. La realtà, però, è ben differente. **Lo standard delle porte COM (l'EIA RS-232), infatti, prevede livelli elettrici non compatibili con i livelli di I/O digitale dei PIC.** Serve un circuito di interfacciamento che adatti le tensioni, circuito che identificheremo in ambito pratico nell'integrato **MAX232**, prodotto dalla Maxim (vedi Focus On nell'ultima pagina di questo Workshop). Una volta inserito correttamente questo componente all'interno del circuito elettrico, PC e PIC potranno dialogare scambiandosi dati grazie ai rispettivi moduli USART.

MIKROC E LE COMUNICAZIONI SERIALI

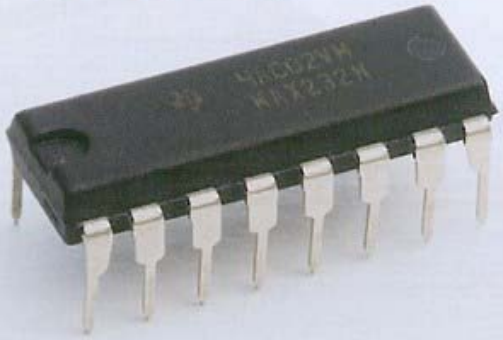
Anche per l'uso del modulo USART mikroC offre un validissimo aiuto, sollevando il programmatore dall'accesso diretto ai registri del PIC e automatizzando tutto il processo di comunicazione. In particolare, l'ambiente di sviluppo di MikroElektronika mette a disposizione **quattro funzioni specifiche per la gestione del modulo USART dei PIC** (funzioni utilizzabili, ovviamente, con i modelli di microcontrollori in cui tale sottosistema è disponibile) chiamate rispettivamente: **'Usart_Init', 'Usart_Read', 'Usart_Data_Ready' e 'Usart_Write'** (maggiori informazioni sul loro utilizzo verranno fornite nel prossimo fascicolo). La procedura **'Usart_Init' si occupa di inizializzare il modulo seriale,** impostando la **velocità di trasmissione,** che deve essere **identica a quella impiegata dal secondo dispositivo coinvolto nel collegamento punto-punto** (sia che si tratti di un altro PIC, sia che si tratti di un personal computer). La porta seriale dei microcontrollori che utilizzeremo opererà, infatti, in **maniera 'asincrona', ossia senza un segnale di temporizzazione** (clock) che sincronizzi la comunicazione dei dispositivi. Tuttavia, se da un lato l'assenza di questo elemento può rendere più semplice il protocollo di trasmissione, dall'altro impone che i sistemi comunichino con la medesima velocità di trasmissione. La velocità di trasmissione è solitamente indicata in **'baud'** o in **'bit al secondo'.** In particolare, **il 'baud rate' è legato al numero di cambi di stato generabili dal sistema di trasmissione,** mentre **il 'bit rate' corrisponde al numero di bit trasmessi** per unità di tempo (nel caso specifico della trasmissione seriale che sperimenteremo, poiché il sistema opera su due stati, ogni singola transizione in trasmissione può inviare solo 1 bit: avremo quindi 1 baud/sec = 1 bit/sec, ma ciò non è sempre vero!). La seconda funzione messa a disposizione da mikroC è la **'Usart_Data_Ready', che può essere chiamata per verificare la presenza di dati nel buffer di ricezione del PIC.** Se la chiamata restituisce '1', significa che all'interno del buffer vi sono dati che possono essere letti ('0' in caso opposto). Le ultime due funzioni **'Usart_Read' e 'Usart_Write' permettono, infine, di leggere e scrivere byte tramite il modulo USART.** Nel prossimo fascicolo realizzeremo il primo programma di esempio.



Le linee seriali dei PC e dei PIC operano secondo livelli elettrici differenti, che devono essere adattati con circuiti specifici, come l'integrato MAX232.

FOCUS ON

L'INTEGRATO MAX232



Abbiamo visto nelle pagine precedenti che alcuni modelli di PIC possono essere configurati per supportare la comunicazione seriale. Ciò non è, però, sufficiente a permettere il collegamento di questi microcontrollori alle comuni porte seriali dei Personal Computer, a causa della differenza dei livelli elettrici dei dispositivi coinvolti. Infatti, mentre i livelli logici dei PIC sono associati ai valori di tensione +5 V (valore logico '1') e 0 V (valore logico '0'), la porta seriale dei comuni PC ha una architettura elettronica rispondente allo standard EIA RS-232. Tale standard industriale, opera suddividendo la tensione di output in tre distinti range chiamati 'MARK', 'SPACE' e 'INCERTEZZA'. Si identifica il segnale elettrico MARK se sulla linea di trasmissione è presente una tensione compresa tra i -3 V e i -15 V, che è impiegata per identificare il livello logico '1'. In contrapposizione a questo segnale troviamo il segnale SPACE, definito tra +3 V e +15 V e associato al livello logico '0'. La tensione compresa tra -3 V e + 3 V, invece, è un range di incertezza, entro il quale non è possibile identificare con esattezza il bit trasmesso. Come puoi osservare da quanto abbiamo appena descritto, non solo lo standard RS-232 lavora con tensioni maggiori in modulo, ma persino negative e con livelli elettrici opposti rispetto ai PIC (nei PIC la tensione 'alta' corrisponde al valore logico '1', nello standard RS-232 il valore logico non solo non è la tensione maggiore, ma è addirittura negativa). Come rendere compatibili, allora, questi segnali? Una soluzione semplice ed economica è data dall'uso del circuito integrato MAX232, introdotto dalla Maxim semiconduttori (foto sopra) e distribuito anche da altre multinazionali dell'elettronica. Il MAX232, utilizzato assieme a una serie di condensatori aggiuntivi secondo uno schema circuitale praticamente 'fisso' (mostrato a sinistra), permette di interfacciare i livelli elettrici dei PIC con quelli dello standard RS-232, rendendo così possibile la comunicazione tra i comuni personal computer dotati di porta seriale e i nostri microcontrollori. Nel corso dei nostri esperimenti ricorreremo sempre alla configurazione mostrata nella figura a lato.