

# SCHEDA A/D 24 CANALI CON 2 PWM

**Se dovete monitorare lo stato di periferiche o di sensori ed al tempo stesso controllare 8 uscite, provate la scheda AD8CHRS: un vero prodigio tecnologico, dal costo contenuto e dalle grandissime potenzialità**

Paola Sbrana

**R**ecentemente ci siamo accorti che sul mercato circolano con sempre maggior frequenza schede di acquisizione dati, di controllo e di gestione apparati che si interfacciano con il personal computer attraverso la porta seriale RS232.

Anche Progetto Elektor è stato partecipe di questo sviluppo, presentando non molto tempo fa una scheda di acquisizione a 4 canali analogici ed in tempi ancor più remoti una scheda a 16 canali digitali di ingresso e uscita. Ultimamente poi, abbiamo presentato anche un software che permette, anche ai meno esperti, di gestire la porta seriale del computer con estrema semplicità.

Ciò che però ancora mancava, era una scheda che avesse a bordo una serie di canali adatti a ciascun tipo di applicazioni e che dialogasse ad una velocità di almeno 9.600 baud.

La periferica che stiamo per mostrare, incorpora tutte le migliori funzionalità richieste, unite a un software di gestione che gira sotto Windows 3.1 e di immediato accesso.

## La nostra proposta

La scheda che andiamo ora ad analizzare, presenta le seguenti caratteristiche:

- 8 canali analogici di ingresso ad 8 bit;
- 8 canali digitali di ingresso;
- 8 canali digitali di uscita;
- 2 canali con uscita PWM a 3.500 Hz con risoluzione di 4 bit ciascuno;
- protocollo seriale RS232 a 9.600 baud, 8 bit, nessuna parità.

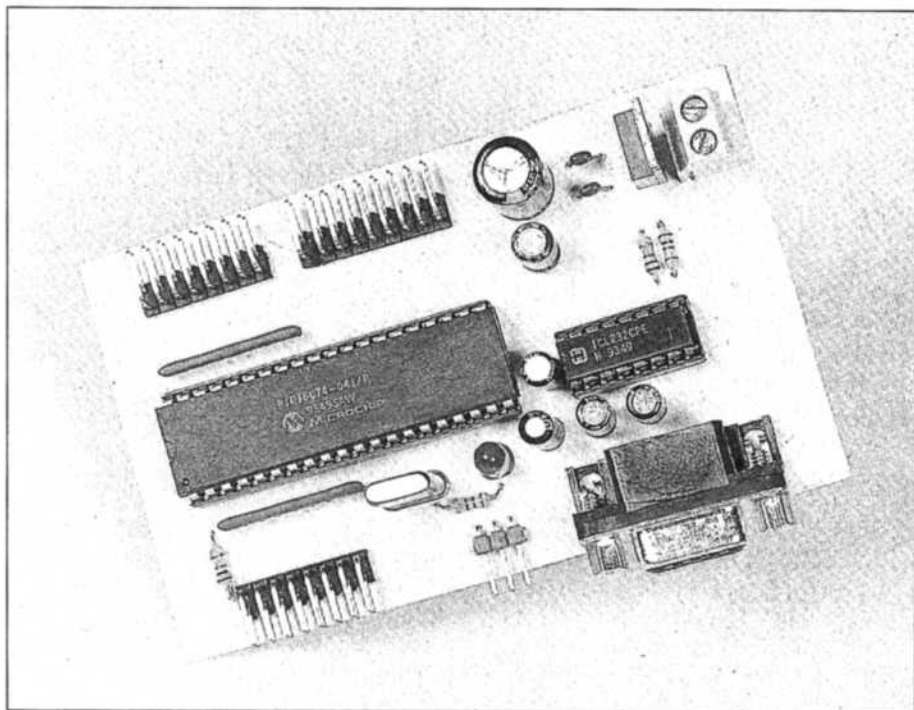
Dallo schema elettrico riportato in Figura 1, vediamo subito l'estrema semplicità del circuito: un integrato che converte i livelli della seriale, un integrato che stabilizza la tensione di lavoro a 5 volt ed un integrato che svolge tutte le funzioni richieste, ovvero un PIC16C74 appositamente programmato.

Non ci dilungheremo sulla struttura interna di questo integrato, dato che verrà successivamente presentato con un articolo completo dedicato esclusivamente ai nuovi prodotti della Microchip, ma a grandi linee possiamo anticipare che al suo interno troviamo una porta USART completa che può lavorare in full-duplex sia con protocolli sincroni che asincroni (compreso il bus IIC e l'SPI), tre timer di cui uno a 16 bit, 12 sorgenti di interrupt, 4K di memoria di programma, 192 byte di RAM, un convertitore a 8 bit multiplexato su 8 canali, due uscite PWM con risoluzione possibile fino a 10 bit, una parallel slave port, e diverse altre caratteristiche che lo rendono uno dei più potenti chip oggi sul mercato (ovviamente dello stesso livello).

Il firmware con cui è stato programmato il PIC è stato scomposto in tante piccole subroutine, sfruttando anche il fatto che tale PIC ha lo stack ad otto livelli. In Figura 2 vediamo il diagramma a blocchi del software assemblato: dopo la consueta fase di inizializzazione, che per questo chip è più lunga e particolareggiata degli altri, il chip si mette in attesa di un comando proveniente da una linea seriale RS232.

Analizzeremo poi la struttura di tale comando, in quanto composto di due parti: il comando vero e proprio e un possibile operando.

Al ricevimento di uno dei possibili 14 comandi, il chip esegue la funzione richiesta, poi si rimette in attesa di un nuovo comando.



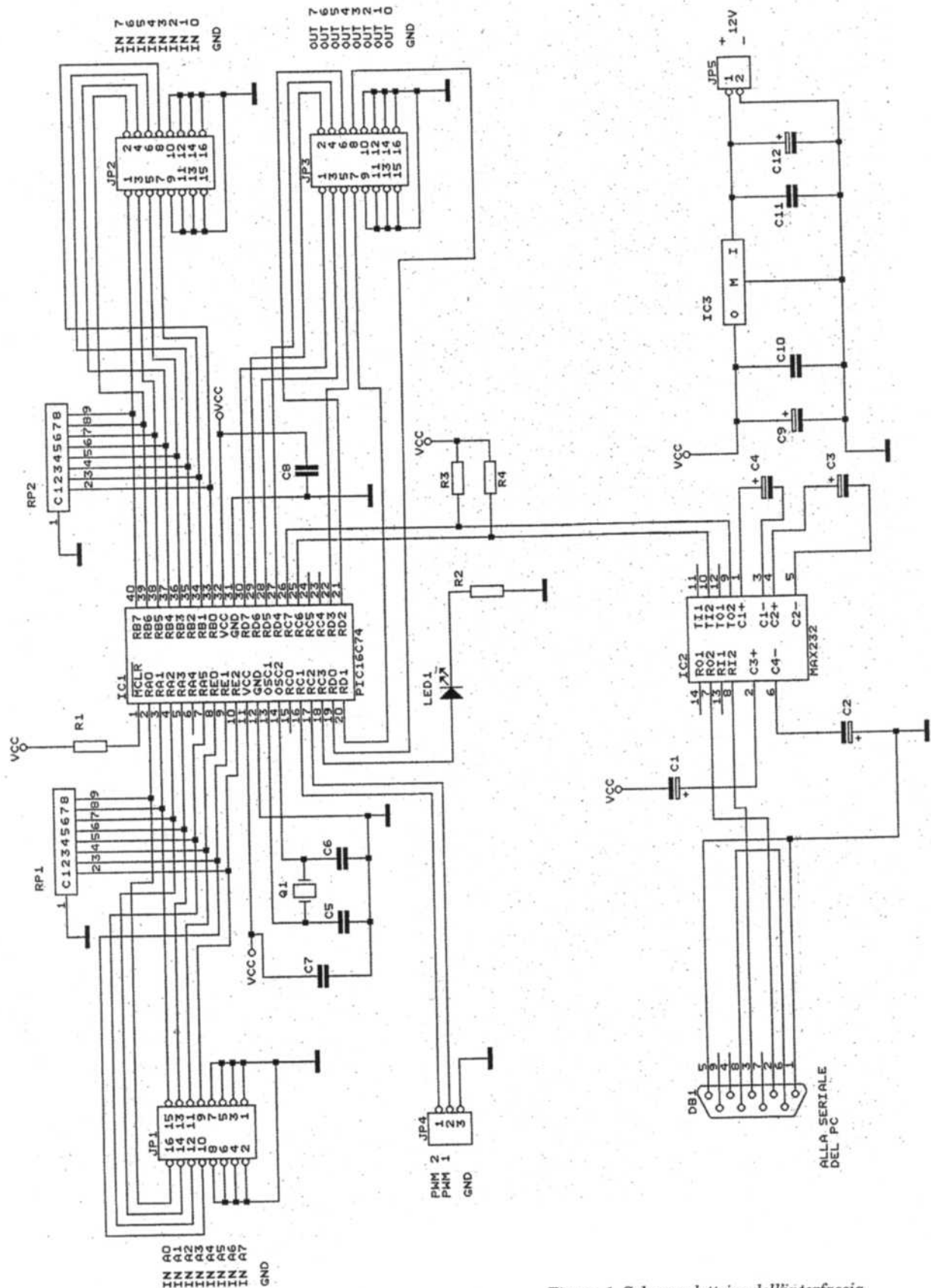


Figura 1. Schema elettrico dell'interfaccia

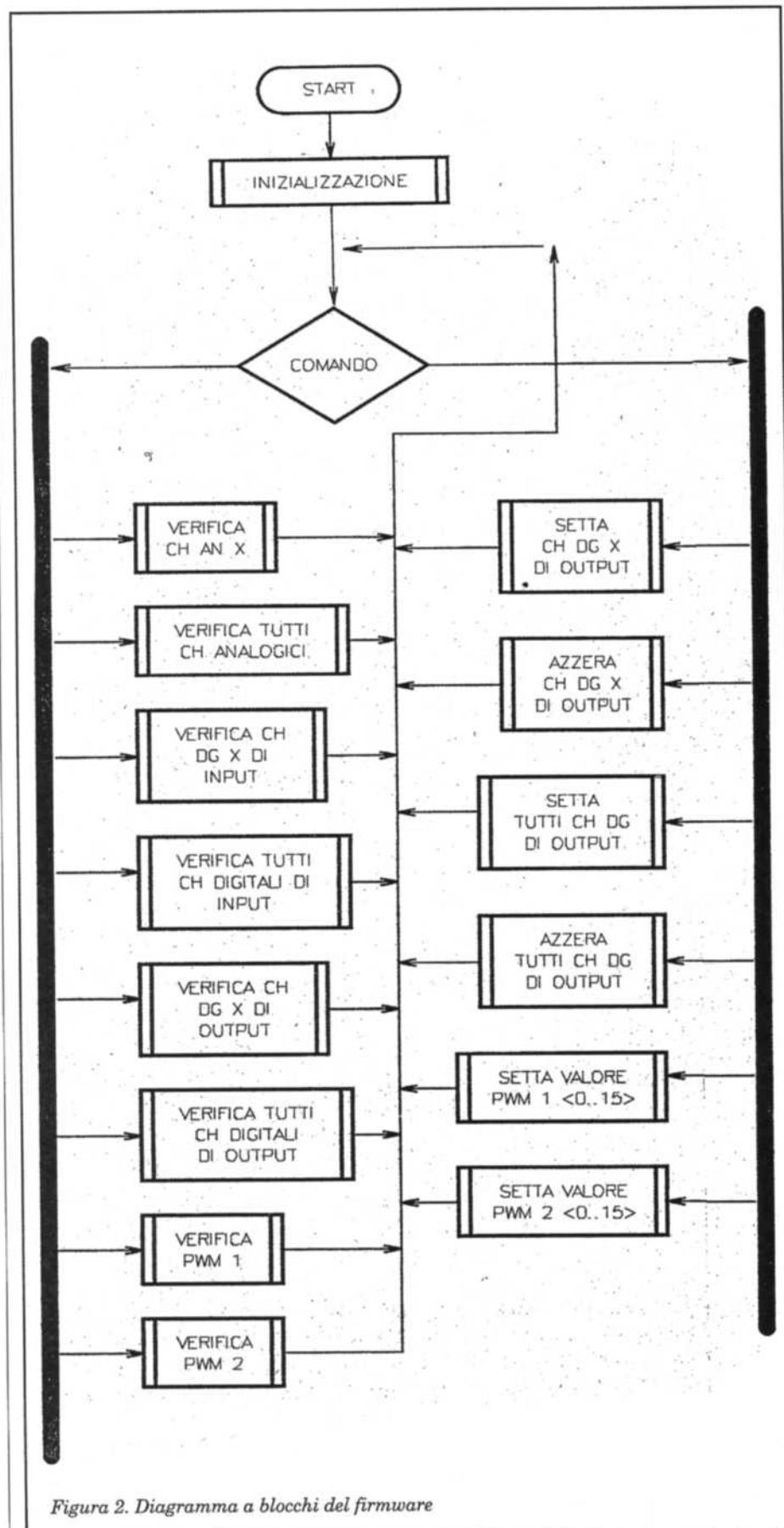


Figura 2. Diagramma a blocchi del firmware

## ELENCO COMPONENTI

### Semiconduttori

IC1: PIC16C74 programmato  
(0347/2643514)  
IC2: ICL232  
IC3: 7805

### Resistori

R1, R3, R4: 10 kΩ  
R2: 220 Ω  
RP1: Rete resistiva 1 kΩ  
RP2: Rete resistiva 10 kΩ

### Condensatori

C1-C4: 4,7 μF 12V  
C5, C6: 18 pF  
C7-C11: 100 nF  
C9: 47 μF 12V  
C12: 220 μF 16V

### Varie

Q1: Quarzo 3,579545 MHz  
LED1: Led 5 mm rosso

Le funzioni ammesse sono 12:

- Invio del valore relativo ad un ben preciso canale analogico.
- Invio di tutti gli 8 valori relativi ai canali analogici.
- Invio dello stato di tutti gli 8 canali di input.
- Invio dello stato di tutti gli 8 canali di output.
- Invio del valore dell'uscita PWM 1.
- Invio del valore dell'uscita PWM 2.
- Ricezione e modifica dello stato di un canale di output.
- Ricezione e modifica dello stato di tutti i canali di output.
- Settaggio di tutti i canali di uscita ad off.
- Settaggio di tutti i canali di uscita ad on.
- Settaggio del canale di uscita PWM 1.
- Settaggio del canale di uscita PWM 2.

Per quanto riguarda queste due ultime uscite, precisiamo che la frequenza del PWM è fissata a 3.500 Hz e non è possibile modificarla. È, invece, possibile variarne il duty-cycle, come si vede in Figura 3. La tempistica è visualizzata in microsecondi.

Quando il valore del PWM è 0, l'uscita viene tenuta a basso livello. Poi, dando un valore compreso tra 1 e 14, si ottengono duty-cycle con passi da circa 20 μS.

Quando il valore è 15 (0fh), l'uscita si troverà sempre a livello 1 (5 volt).

Potevamo sfruttare tutti e gli otto o i dieci bit a disposizione del PWM, ma all'utilizzatore sarebbe risultato difficile gestire il dialogo sulla porta seriale in modo semplice, quindi abbiamo optato per sedici livelli di duty-cycle, fra l'altro anche troppi per determinate applicazioni.

### Montaggio

In Figura 4 possiamo trovare la traccia per la realizzazione del circuito stampato necessario alla realizzazione della scheda. Ricordiamo che è possibile reperire la scheda anche già montata, collaudata e comprensiva di software per PC telefonando al numero 0347/2643514.

In Figura 5, invece, abbiamo la disposizione dei componenti: non ci sono particolari attenzioni da rilevare, eccetto il ricordarsi di inserire i tre ponticelli in filo di rame e i tre condensatori sotto il circuito integrato IC1.

Sia per IC1 che per IC2, dovete assolutamente montare uno zoccolo per circuiti integrati.

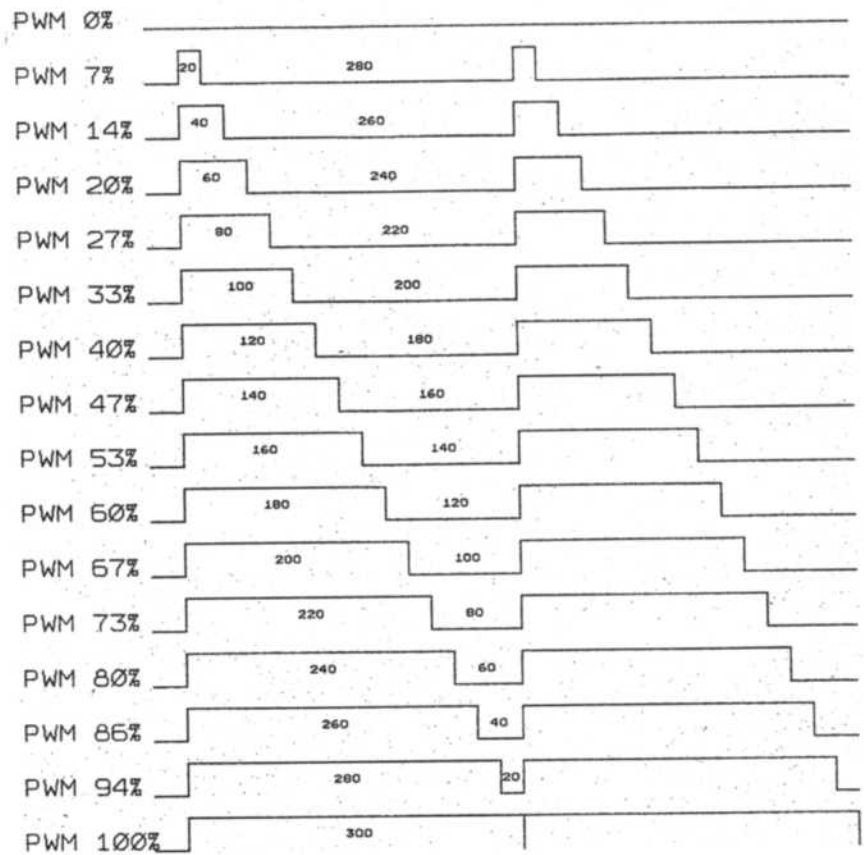
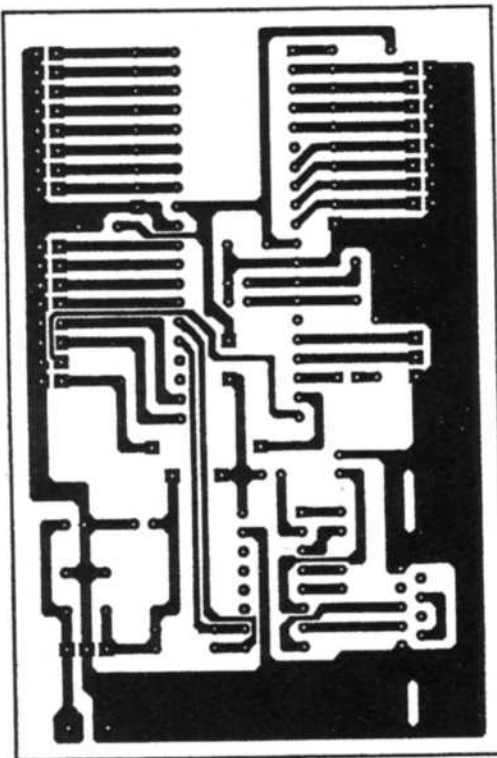


Figura 3. Tempistiche del PWM in µs



▲ Figura 4. Circuito stampato scala 1:1

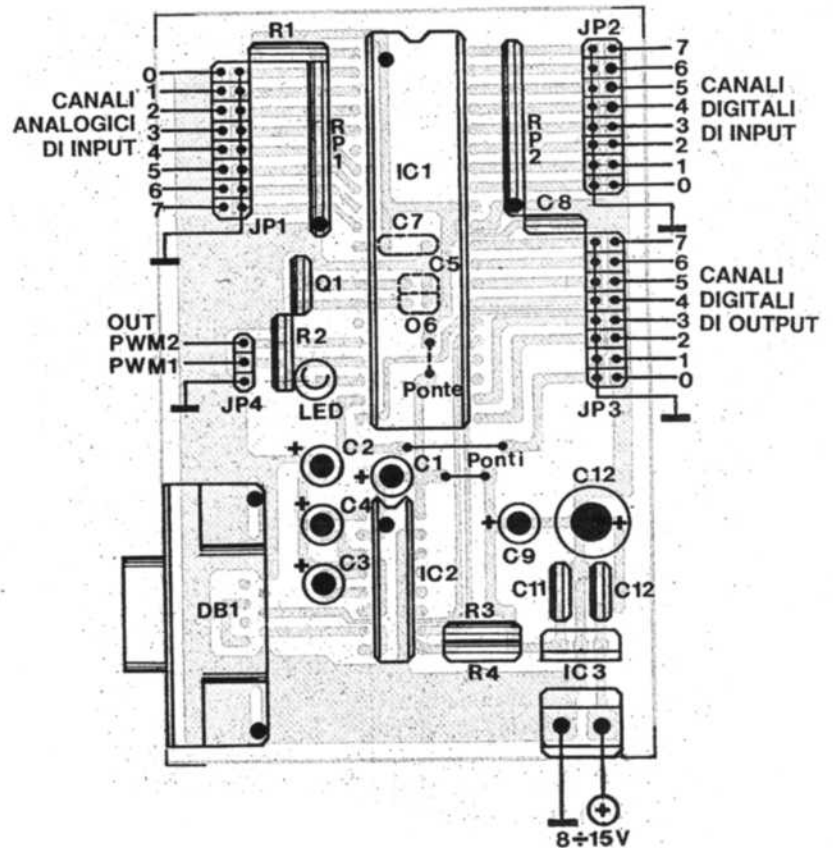


Figura 5. Disposizione dei componenti ▲



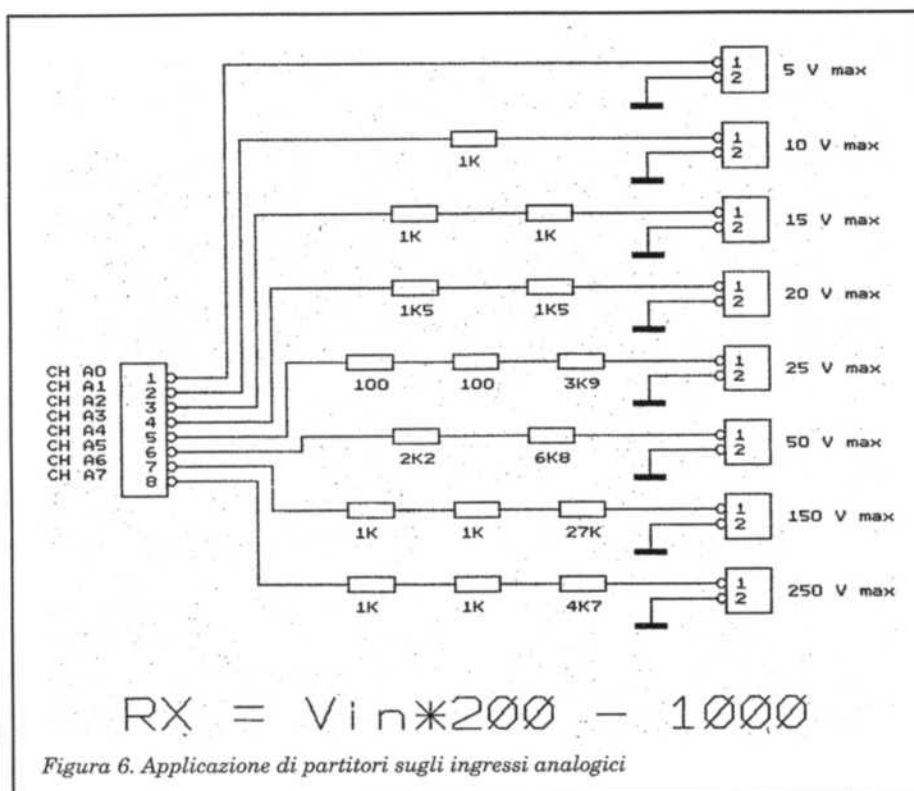


Figura 6. Applicazione di partitori sugli ingressi analogici

Una volta pronta la scheda, potrete divertirvi a vederla funzionare, o con il software che già vi abbiamo presentato, oppure con quello dedicato.

Nella disposizione dei componenti di Figura 5 si può notare sulla parte sinistra gli otto canali rappresentati (con otto barre colorate) che si alzeranno da 0 a 255 quando la tensione ai capi degli ingressi analogici corrispondenti salirà da 0 a 5 volt.

Sul lato destro, invece, abbiamo 16 canali che rappresentano i rispettivi 16 led.

Gli otto in alto sono relativi ai canali di input, gli otto in basso sono dedicati ai canali di output.

Il colore di questi led visualizza lo stato dei rispettivi canali: il colore verde significa che l'ingresso digitale è a basso livello così come l'uscita digitale.

Il colore rosso identifica un ingresso con valore logico a 1 ed un'uscita con valore di tensione di 5 volt.

Mentre per gli otto canali di input non è possibile fare variazioni di stato, per gli otto canali di output è molto semplice: per portarli tutti insieme allo stato attivo (uscita alta e led rosso) è sufficiente "cliccare" con il mouse sul pulsante con su scritto "ch ON", mentre per portarli tutti quanti a basso livello (uscita bassa e led verde) basta cliccare sul pulsante con su scritto "ch OFF".

Invece, per modificare lo stato di un solo canale, è possibile agire sui piccoli pulsanti posti sotto ogni led relativo ad un canale di uscita. La funzione sarà quella di "toggle", ovvero di inversione dello stato attuale.

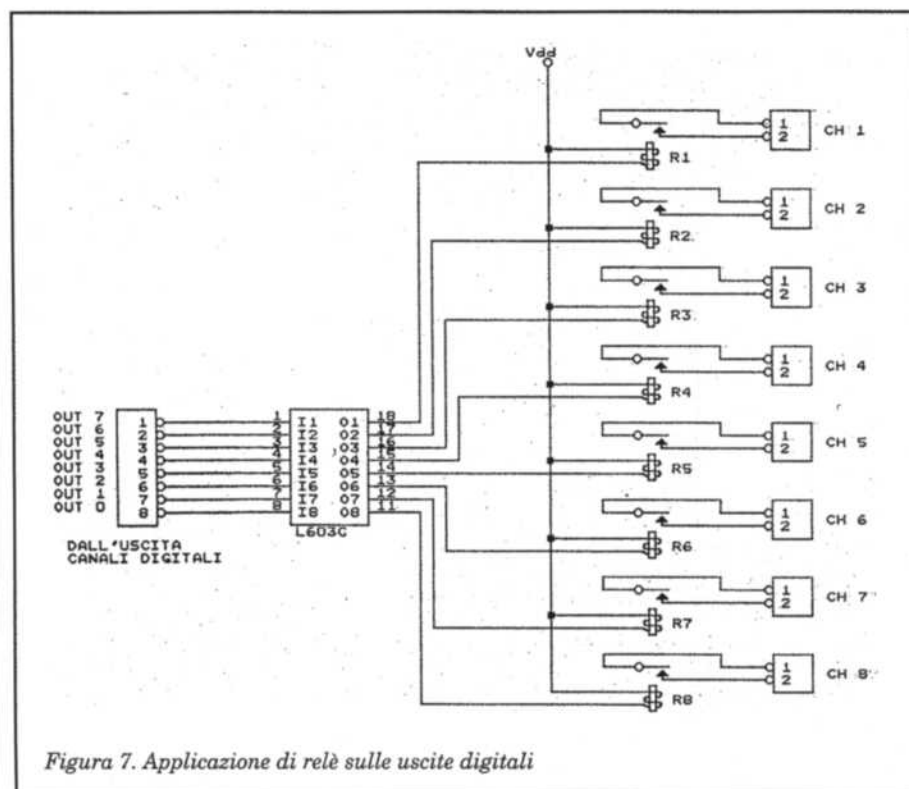


Figura 7. Applicazione di relè sulle uscite digitali

In pratica, se il canale è off (led verde), cliccando sul relativo pulsante tale canale assumerà lo stato di on (led rosso).

**FATE ATTENZIONE:** a fare un clic nel momento giusto, ovvero quando non ci sono comunicazioni in corso tra la scheda e la seriale, altrimenti il comando verrà perso. Nel software proposto, lo scambio di tutti i dati avviene automaticamente ogni 5 secondi, per aggiornare i valori di ogni canale.

Tornando alla schermata del programma, in basso troviamo due indicatori e due selettori per l'impostazione dei due canali PWM.

Cliccando sui selettori è possibile modificare lo stato delle uscite PWM e vederne il risultato sugli indicatori.

Uscendo dal programma con il tasto "Fine", non saranno modificate le impostazioni della scheda, quindi i canali digitali di uscita posti ad alto livello rimarranno in tale stato e così pure le due uscite PWM.

Apriamo una piccola parentesi per far notare che tale programma deve essere attivato esclusivamente dopo che la scheda è stata collegata alla porta seriale e alimentata, altrimenti a causa della determinazione automatica del numero di porta seriale (COM1 o COM2), la schermata non apparirà e sarà necessario uscire e rientrare in Windows per ripartire.

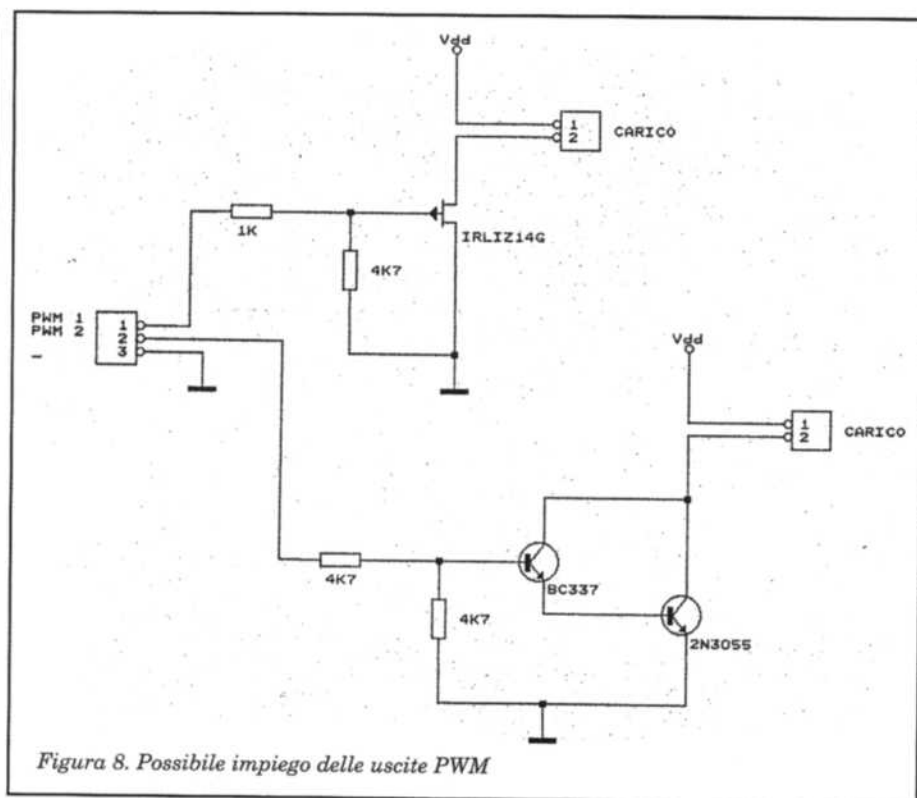


Figura 8. Possibile impiego delle uscite PWM

## Collaudo ed applicazioni

Per collaudare la scheda, o sfruttare il software in dotazione al chip, oppure vi create del software specifico, ricordando che la comunicazione avviene a 9.600 baud, 1 bit di Start, 8 bit di dati, nessuna parità, 1 bit di Stop.

Il formato dei comandi è del tipo XY, dove XY è un byte composto dai due nibble X e Y. X rappresenta il comando, come da seguente elenco:

- 1h Leggi canale analogico Y (con Y compreso tra 0 e 7).
- 2h Leggi tutti i canali analogici.
- 3h Leggi lo stato di tutti i canali di input.
- 4h Leggi lo stato di tutti i canali di output.
- 5h Leggi il duty-cycle dell'uscita PWM1.
- 6h Leggi il duty-cycle dell'uscita PWM2.
- 7h Setta ad ON il canale di uscita Y (con Y compreso tra 0 e 7).
- 8h Setta ad OFF il canale di uscita Y (con Y compreso tra 0 e 7).
- 9h Setta ad ON tutti i canali di uscita.
- ah Setta ad OFF tutti i canali di uscita.
- bh Modifica il valore del PWM 1 (con Y compreso tra 0 e fh).
- ch Modifica il valore del PWM 2 (con Y compreso tra 0 e fh).

Come si vede, alcuni comandi sono seguiti da un operando. Quelli che non lo richiedono devono egualmente avere un secondo nibble, che noi consigliamo essere identico al primo per non sbagliare.

Inoltre, ci sono alcuni comandi che fanno trasmettere al chip dei valori, altri che non lo fanno: i comandi 1, 3, 4, 5, 6 fanno sì che il chip risponda con un byte. Il comando 2 fa rispondere al chip con 8 byte consecutivi, mentre i comandi dal 7 in poi non fanno trasmettere al chip alcun byte di risposta.

Vediamo allora un esempio di dialogo senza il software predisposto, ma con un semplice programma tipo terminale.

Per prima cosa inviamo il comando per leggere lo stato di tutti i canali di input, trasmettendo alla scheda 33h (il secondo 3 serve solo per riempire il secondo nibble, infatti avremmo ottenuto lo stesso risultato scrivendo 31h, 35h, 36h e così via).

Dopo tale trasmissione, il PC riceverà un byte che indicherà lo stato di TUTTI i canali di input: il bit 0 per il primo canale, il bit 1 per il secondo e così via fino al bit 7 per l'ottavo canale.

Se un bit è a zero, significa che quel canale è non attivo, se il bit è a 1 significa che quel canale è attivo.

Se invece mandiamo il comando 13h, vogliamo sapere il valore del canale analogico relativo al pin RA3.

Il chip allora ci rinvia un byte che corrisponde al valore della conversione per quel canale. Tale valore è rapportato ad una tensione da 0 a 5 volt, quindi se sul pin sono presenti 5 volt, avremmo di ritorno il valore ffh, se sul pin sono presenti 2,5 volt, avremmo di ritorno un 80h e via di seguito.

Se, invece, desideriamo conoscere il valore di tutti e 8 i canali analogici, dovremo inviare il comando 22h (anche qui il secondo 2 è random) ed il chip ci invierà di ritorno 8 byte, con la sequenza relativa ai canali analogici dal primo all'ultimo. Ogni byte conterrà il valore del relativo canale.

Pensiamo invece di voler settare a 1 il canale di uscita 5: allora il comando da inviare al chip sarà 75h, dove 7 è il comando richiesto e 5 è il numero del canale (sarà 06h se partirete con la numerazione dei canali da 1 anziché da 0).

Il chip eseguirà la modifica ma non ritornerà alcun messaggio di conferma.

Supponiamo adesso di voler azzerare tutti i canali di uscita: invieremo il comando aah, ed anche qui non avremo conferme.

Ora vogliamo prima vedere il valore dell'uscita PWM1 e poi modificarlo: invieremo il comando 55h, riceveremo un byte con un valore compreso tra 0 e 0fh, poi invieremo per esempio bah, che indica al chip di passare al livello ah (dieci) di duty-cycle per il PWM 1. Anche qui non avremo conferme ulteriori.

Come vedete, il dialogo tra il PC e la nostra scheda non è tra i più complessi, trattandosi sempre di un byte per volta.

Per quanto riguarda le applicazioni, queste sono infinite e limitate soltanto dalla vostra fantasia.

In Figura 6 trovate dei partitori già calcolati per adattare l'ingresso a 5 volt analogico ad ogni tipo di tensione.

Dovete comunque stare attenti a non far mai superare il livello di 5 volt su qualsiasi ingresso analogico, perché questo, oltre a poter danneggiare tale ingresso, influenzerebbe la lettura anche degli altri canali.

Ricordiamo che le resistenze impiegate, dovranno essere all'1%.

In Figura 7, invece, proponiamo una possibile soluzione per le uscite digitali, ma ne potrete adottare anche altre, ad esempio pilotando direttamente dal chip dei relè statici, oppure realizzando circuitazione mista.

Infine, nello schema di Figura 8, suggeriamo due possibili soluzioni per le uscite PWM.