

# Esercizi di apprendimento

```

1 ;
2 ;Programma combinatoriale
3 ;
4 ;A seconda dello stato degli interruttori RC0 e RC1, attivare i leds RB0-RB7 collegati alla
5 ;porta B, seguendo la seguente tabella della verita':
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;

```

	RC1	RC0	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
9 :	0	0	1	0	1	0	1	0	1	0
10 :	0	1	0	1	0	1	0	1	0	1
11 :	1	0	0	0	0	0	1	1	1	1
12 :	1	1	1	1	1	1	0	0	0	0

```

16 List p=16F870 ;Tipo di processore
17 Include "P16F870.INC" ;Definizione dei registri interni
18 ORG 0x00
19 Initio clrf PORTB ;Cancella i latch di uscita
20 bsf STATUS,RP0 ;Seleziona il banco 1
21 clrf TRISB ;Configura la Porta B come uscita
22 movwf BCF
23 movwf TRISC ;Porta C configurata come ingresso
24 bcf STATUS,RP0 ;Seleziona il banco 0
25

```

Realizziamo ora un altro esercizio di apprendimento. Si tratta di un esercizio combinatoriale; abbiamo due interruttori (SW3 e SW4) che gestiscono lo stato dei piedini RC0 e RC1 del microcontroller. Possiamo avere quattro stadi differenti in funzione del valore 1 o 0 di questi interruttori. Per ogni combinazione di ingresso genereremo una combinazione di uscita differente che consisterà in un dato a 8 bit da inviare alla porta B del microcontroller, in modo che si possa visualizzare sugli otto diodi LED.

```

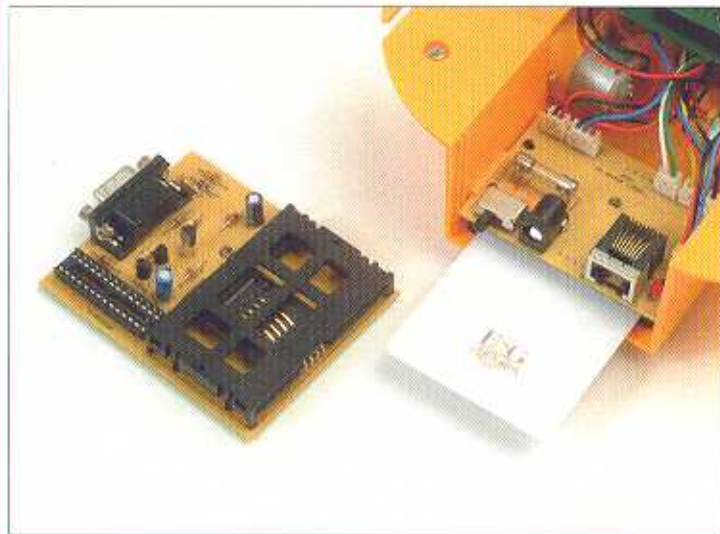
26 Loop: clrf PORTC ;Aggiorna il PORTC
27 btfsc PORTC,0 ;Testa lo stato di RC0
28 goto RC0_0_1 ;E' a "1"
29 btfsc PORTC,1 ;E' a "0". Testa lo stato di RC1
30 goto Sono_a_10
31 movlw b'10101010'
32 movwf PORTB ;Sono a 10. Esci dalla sequenza
33 goto Loop
34
35 Sono_a_10 movlw b'00001111'
36 movwf PORTB ;Sono a 10. Esci dalla sequenza
37 goto Loop
38
39 RC0_0_1 btfsc PORTC,1 ;Testa lo stato di RC1
40 goto Sono_a_11 ;E' a "1"
41 movlw b'01010101'
42 movwf PORTB ;Sono a 01. Esci dalla sequenza
43 goto Loop
44
45 Sono_a_11 movlw b'11110000'
46 movwf PORTB ;Sono a 11. Esci dalla sequenza
47 goto Loop
48
49 end ;Fine del programma sorgente

```

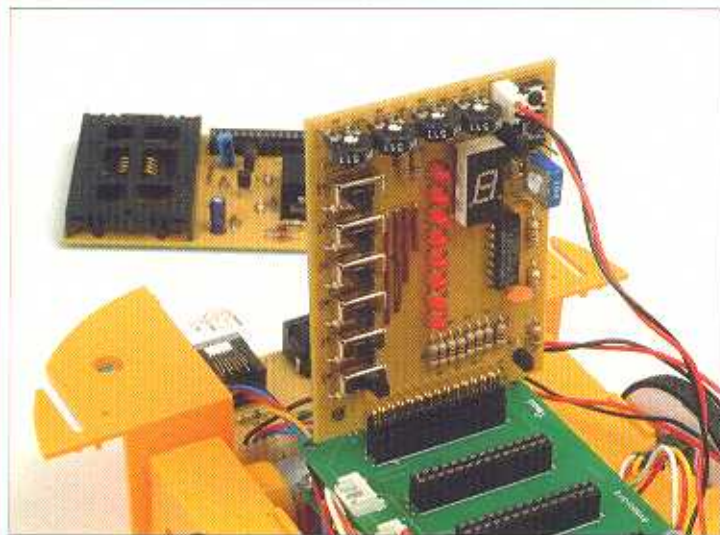
Questo esercizio si trova nel CDROM sotto il nome di es13.asm. Copieremo il file su una directory nel nostro hard disk per procedere alla sua compilazione. Il programma configura la porta C come ingresso per gli interruttori e la porta B come uscita per i diodi LED. Successivamente si entra nei cicli che testano mediante istruzioni di salto btfsc il valore dei bit RC0 e RC1, per le quattro combinazioni possibili. Per ogni stato viene inviato il valore corrispondente degli otto diodi della porta B.

Dopo aver compilato il programma otterremo il file es13.hex, che dovrà essere caricato sulla Smartcard. Metteremo la Smartcard sulla scheda di scrittura e quest'ultima la collegheremo al PC mediante la porta seriale. Tramite il programma ICPRQG selezioneremo il dispositivo 24C16, apriremo il file es13.hex e procederemo alla programmazione. Vi consigliamo di eseguire una verifica dopo aver programmato la Smartcard mediante l'opzione "Verifica" del menù "Comando".

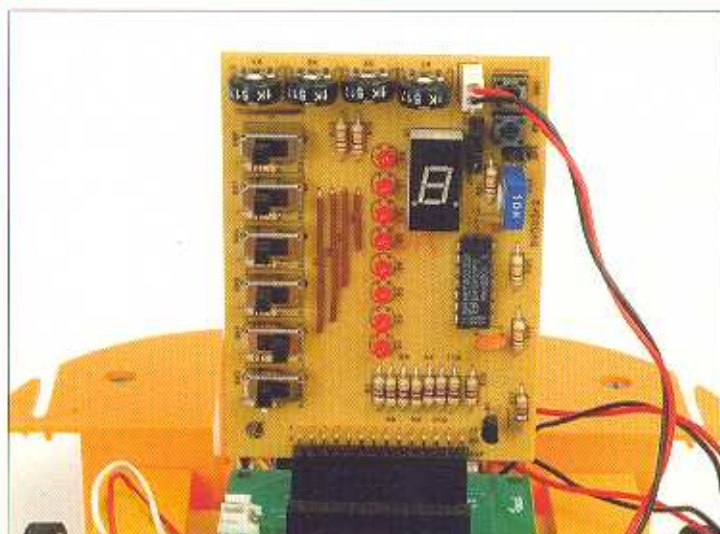




La procedura di programmazione di Pathfinder è la stessa che stiamo utilizzando per verificare gli esercizi di ingresso e uscita. Prima si inserisce la Smartcard sulla scheda di scrittura, per trasferire il programma sulla memoria, quindi dopo averla programmata la monteremo sul robot mediante la scheda di alimentazione. È importante che la Smartcard venga sempre inserita con l'orientamento adeguato su entrambe le schede.



Per verificare questo esercizio la scheda di ingresso e uscita deve essere inserita sul connettore della scheda di interfaccia JP13. Dopo aver alimentato il robot, il microcontroller impiegherà alcuni secondi a leggere il contenuto della Smartcard quindi inizierà l'esecuzione dell'esercizio. Per fare in modo che l'esercizio funzioni, il jumper JP1 deve essere chiuso, mentre JP4 dovrà essere aperto, per attivare i diodi LED e disattivare il display a sette segmenti.



Nell'immagine possiamo vedere l'esercizio durante il funzionamento. Modificando i valori dei commutatori SW3 e SW4 inseriremo una delle quattro combinazioni possibili. Per ogni combinazione inserita osserveremo come cambia lo stato dei diodi LED che corrisponderà alle combinazioni di 1 e 0 selezionate nel programma.