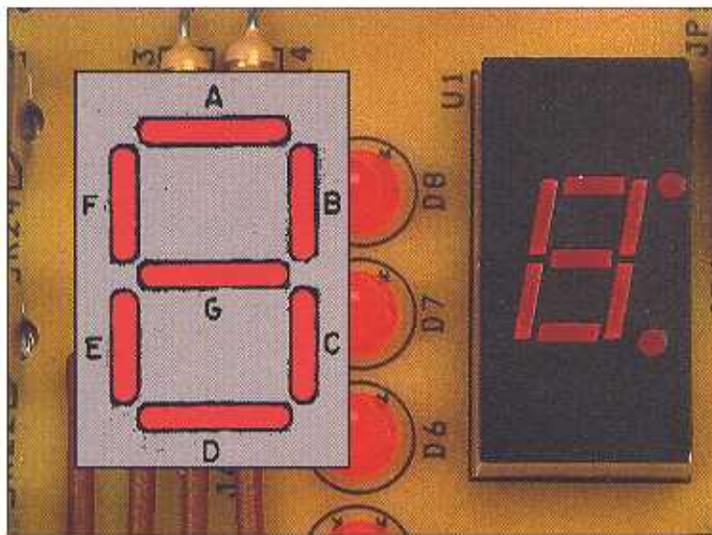


# Esercizi di apprendimento



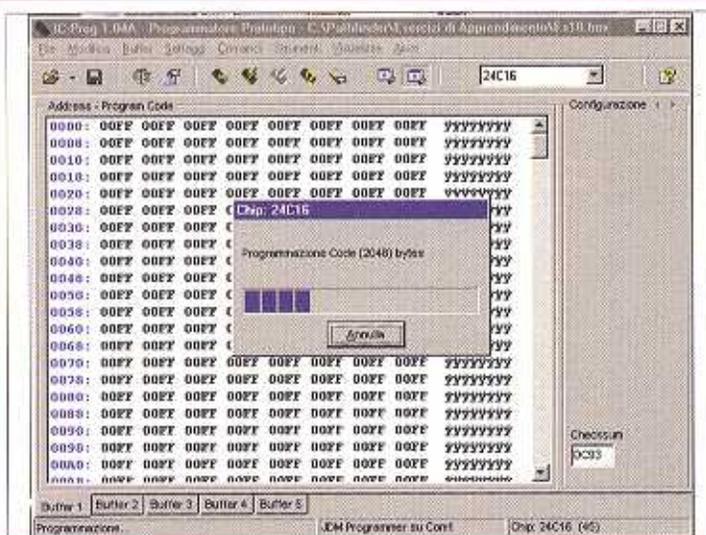
Realizzeremo un esercizio dimostrativo dell'uso di un display a 7 segmenti. Questo dispositivo è composto da 8 diodi LED, disposti in modo da rappresentare un numero decimale quando si illumina. Questi segmenti utilizzano 7 diodi LED, l'ottavo diodo è il punto decimale che si trova nella parte inferiore del display. I segmenti sono denominati con le lettere dalla "a" sino alla "f", secondo la disposizione mostrata nell'immagine. Tutti i segmenti sono collegati in modo consecutivo alla porta B, dal segmento "a" su RB0, sino al segmento "f" su RB6. Infine, RB7 è collegato al punto decimale del display (segmento "h").

```
c:\spdm\lavoro\25018.asm
1
2 ;Il Display a 7 segmenti
3
4 ;Sul display a catodo comune collegato alla porta B, si desidera visualizzare lo stato
5 ;logico "0" o "1" dell'interruttore RC0. Mediante l'interruttore RC1 si attiva o no il
6 ;punto decimale.
7
8 List p-16F870 ;tipo di processore
9 include "P16F870.INC" ;definizione dei registri interni
10
11 org 0x0000
12
13 inizio      clrwf  PORTB      ;Cancella i latch di uscita
14           bsf   STATUS,RP0   ;Seleziona banco 1
15           clrwf  TRISB      ;Configura la Porta B come uscita
16           movlw 0x0F        ;
17           movwf TRISC       ;Configura la Porta C come ingresso
18           bcf   STATUS,RP0   ;Seleziona banco 0
19
20
```

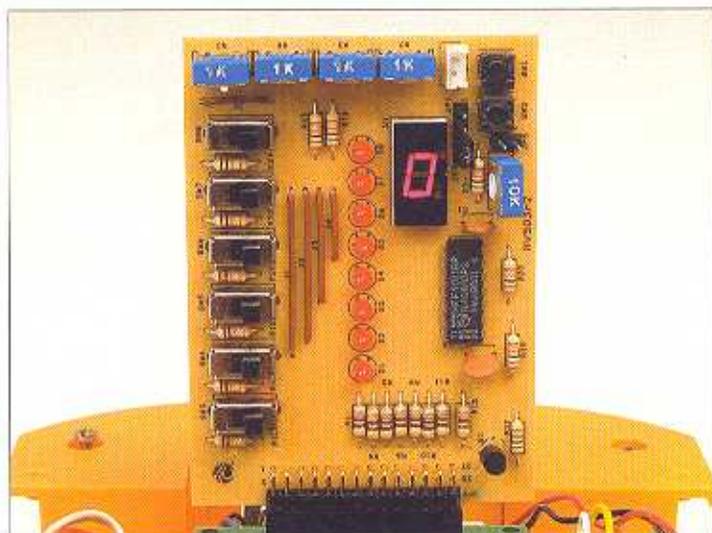
Questo esercizio visualizzerà sul display a 7 segmenti un numero decimale che corrisponderà con lo stato logico, 0 o 1, dell'interruttore RC0. Allo stesso tempo, mediante l'interruttore RC1, potremo modificare lo stato del punto decimale del display. All'inizio del programma si configura la porta B del microcontroller come uscita per la gestione del display, e la porta C come ingresso, per la lettura dello stato degli interruttori.

```
c:\spdm\lavoro\25018.asm
19
20 Loop      clrwdt      ;Raggiorna il MDT
21           btfsc  PORTC,0 ;Testa RC0
22           goto  RC0_1 ;if a livello "1"
23           movlw 0x01111111 ;
24           movwf PORTB ;Visualizza il digit 0
25           goto  Test_RC1
26
27 RC0_1     movlw 0x00001110 ;Visualizza il digit 1
28           movwf PORTB
29
30 Test_RC1 btfsc  PORTC,1 ;Testa RC1
31           goto  RC1_1 ;if a "1"
32           bcf   PORTB,7 ;Spegni punto decimale
33           goto  Loop
34 RC1_1     bcf   PORTB,7 ;Attiva punto decimale
35           goto  Loop
36
37          end          ;Fine del programma sorgente
```

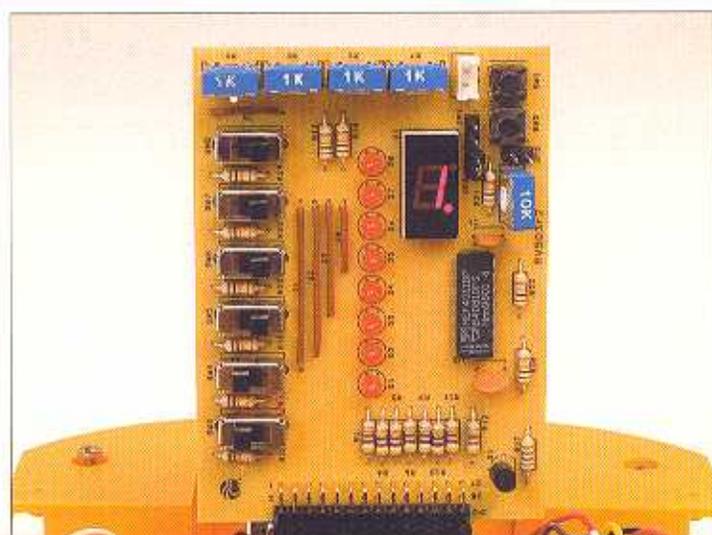
A questo punto il programma entra in un ciclo infinito denominato Loop. All'inizio del ciclo si testa lo stato di RC0. Se lo stato è "0" si visualizza il numero "0" in modo decimale sul display, questo richiederà l'accensione dei segmenti a-b-c-d-e-f. Se lo stato dell'interruttore è "1", verrà visualizzato un "1" sul display, accendendo i segmenti b-c. Dopo aver testato RC0, si aggiorna lo stato di RC1, per accendere o spegnere il punto decimale del display.



Questo esercizio si trova sul CDROM sotto il nome di es18.asm. Dobbiamo compilare l'esercizio per ottenere il file es18.hex che sarà caricato sulla Smartcard. Utilizzeremo la scheda di scrittura e il software ICPROG per scrivere il file esadecimale sulla Smartcard. In seguito, inseriremo la Smartcard sul robot ed eseguiremo la prova dell'esercizio sulla scheda di ingressi e uscite.



Per provare questo esercizio dobbiamo attivare il display a 7 segmenti e disattivare la barra dei diodi LED e l'altoparlante. Per fare questo, chiuderemo il jumper JP4 e apriremo i jumper JP1 e JP5. Nell'immagine possiamo vedere il display a 7 segmenti che mostra il numero decimale "0", dato che l'interruttore RC0 (SW3) ha questo valore. Il punto decimale del display rimane spento, dato che RC1 (SW4) ha il valore logico "0".



In quest'altra immagine appare sul display a 7 segmenti il numero "1", perché l'interruttore RC0 ora è a livello alto. Possiamo anche osservare come si sia illuminato il punto decimale del display, grazie al fatto che anche RC1 è passato a livello alto. In questo modo, potremo visualizzare qualsiasi numero decimale che desideriamo con il display, sarà sufficiente selezionare i segmenti del display che si devono illuminare, tenendo conto del loro ordine di collegamento alla porta B del microcontroller.