

Esercizi di apprendimento

```
c:\p808\1\esercizi\2\test1.asm
1 ;
2 ;Il Display a 7 segmenti. Decodificatore da hex. a 7 segmenti.
3 ;
4 ;tramite i quattro interruttori HD0-HD3 si introduce un valore esadecimale da 4 bita
5 ;che deve essere visualizzato sul display della scheda MicroPIC Trainer
6 ;
7 LIST g=16F87D ;tipo di processore
8 include "16F87D.INC" ;definizione dei registri interni
9 ;
10 org 0x00
11 ;
12 inizio      cllrf PORTB ;cancella i latch di uscita
13             bsf STATUS,RP0 ;seleziona banco 1
14             cllrf TRISA ;configura la Porta A come uscita
15             movlw b'00000111'
16             movwf TRISC ;configura la Porta C come ingresso
17             bsf STATUS,RP0 ;seleziona banco 0
```

Analizzeremo ora un altro esercizio utilizzando il display a 7 segmenti della scheda di ingressi e uscite. In questo esercizio inseriremo il numero binario mediante i primi tre interruttori collegati alla porta C del microcontroller (SW3, SW4 e SW5). Nel display a 7 segmenti vedremo il numero decimale equivalente al numero binario inserito mediante gli interruttori.

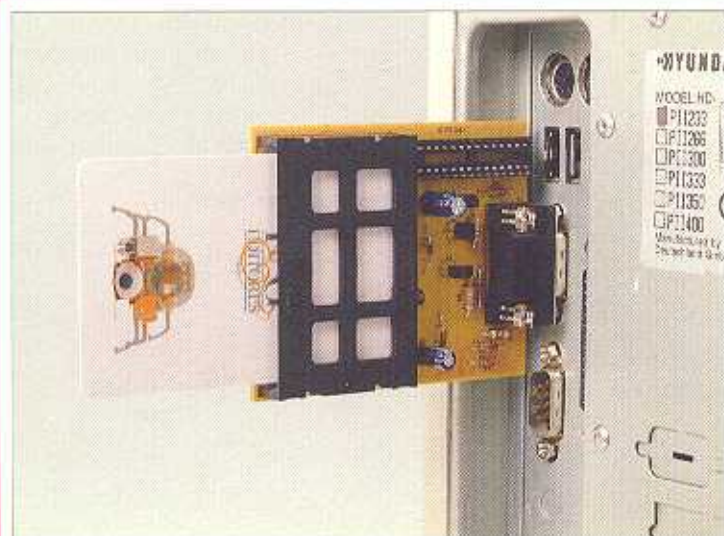
```
c:\p808\1\esercizi\2\test1.asm
9
10 org 0x00
11 ;
12 inizio      cllrf PORTB ;cancella i latch di uscita
13             bsf STATUS,RP0 ;seleziona banco 1
14             cllrf TRISA ;configura la Porta A come uscita
15             movlw b'00000111'
16             movwf TRISC ;configura la Porta C come ingresso
17             bsf STATUS,RP0 ;seleziona banco 0
18 ;
19 loop        cllwlt ;aggiorna il W
20             movwf PORTC,W ;
21             andlw b'00000111' ;legge il codice di 000-000
22             call tabella ;converte a 7 segmenti
23             movwf PORTB ;visualizza sul display
24             goto loop
25
```

All'inizio dell'esercizio realizzeremo le configurazioni. Dobbiamo configurare i primi tre bit della porta C come ingressi per poter leggere lo stato degli interruttori, la porta B si configurerà come uscita per illuminare i diodi LED del display a 7 segmenti. Il programma rimarrà in ciclo infinito denominato loop, in questo ciclo si legge lo stato della porta C e si porta il valore al registro W. Si realizza un'operazione tipo AND per eliminare il resto dei bit della porta C non validi. In seguito verrà chiamata la funzione tabella e il valore risultante si vedrà sul display a 7 segmenti.

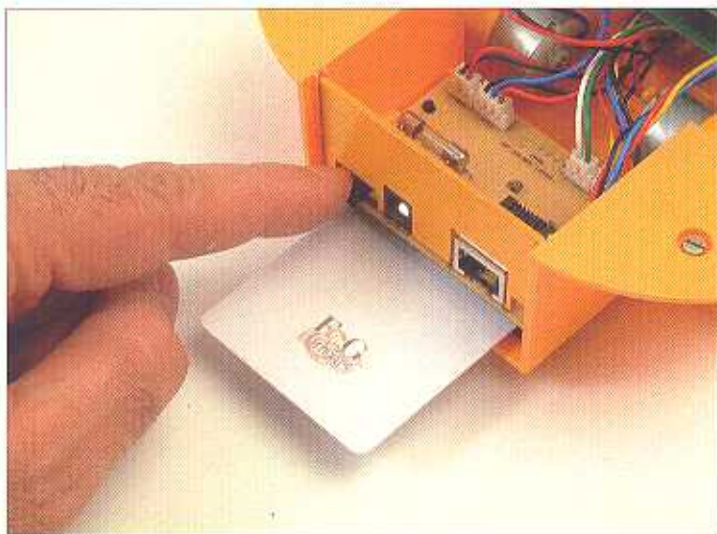
```
c:\p808\1\esercizi\2\test1.asm
26 ;=====
27 ;Tabella: questa routine converte il codice binario presente sui 4 bit meno significativi
28 ;del reg. W nel suo equivalente a 7 segmenti. Il codice a 7 segmenti viene scritto anche
29 ;nel reg. V
30 ;
31 tabella:    aduof PCL,F ;spostamento sulla tabella
32             retlw b'00111111' ;digit 0
33             retlw b'00001110' ;digit 1
34             retlw b'01011011' ;digit 2
35             retlw b'01001111' ;digit 3
36             retlw b'01100110' ;digit 4
37             retlw b'01101101' ;digit 5
38             retlw b'01111101' ;digit 6
39             retlw b'00000111' ;digit 7
40             retlw b'01111111' ;digit 8
41             retlw b'01100111' ;digit 9
42             retlw b'01110111' ;digit A
43             retlw b'01111100' ;digit B
44             retlw b'00111001' ;digit C
45             retlw b'01011110' ;digit d
46             retlw b'01111001' ;digit E
47             retlw b'01110001' ;digit F
48 ;
49             end ;Fine del programma sorgente
```

Questa è la funzione tabella che ha il compito di convertire un numero esadecimale in un codice decimale per il display a 7 segmenti. Quando si entra nella funzione, in W abbiamo il valore binario che vogliamo convertire. Questo valore binario viene sommato al contatore del programma PCL, in modo che si salteranno un numero di istruzioni pari al valore binario presente su W. In ogni posizione della tabella troviamo un'istruzione di ritorno retlw per ritornare al ciclo principale loop, lasciando nel contempo sul registro di lavoro W, il codice decimale corrispondente al valore binario di partenza.

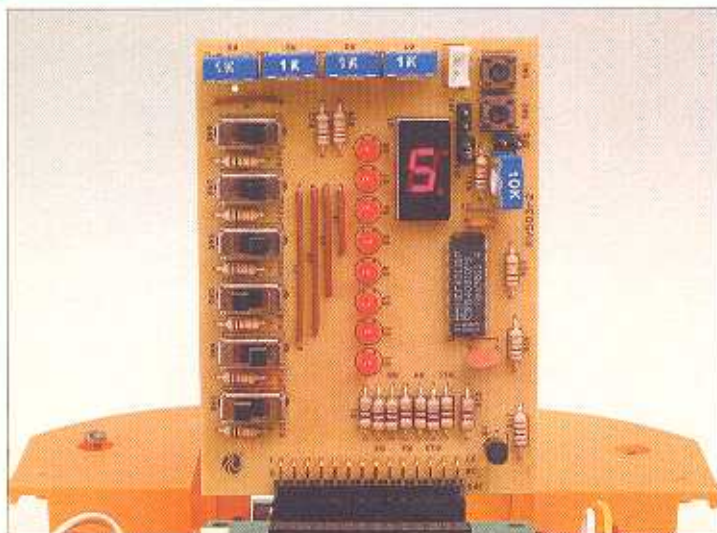
Esercizi di apprendimento



Questo esercizio si trova sul CDRom sotto il nome di es19.asm. Dobbiamo copiare il file sulla nostra directory di lavoro dell'hard disk e procedere alla sua compilazione utilizzando il programma MPLAB. In seguito eseguiremo il software di scrittura ICPROG e mediante la scheda di scrittura caricheremo il file es19.hex nella Smartcard.



Dopo aver scritto la Smartcard la inseriremo sul robot. Per eseguire questo esercizio la scheda di ingressi e uscite deve essere collegata sulla scheda di interfaccia sul connettore JP13. Sulla scheda di ingressi e uscite dovremo attivare il display a 7 segmenti, quindi dovremo chiudere il jumper JP4, il resto dei jumper di questa scheda rimarrà aperto, quindi disabilitaremo la barra dei diodi LED e l'altoparlante.



Ora utilizzeremo gli interruttori SW3, SW4 e SW5 per inserire i numeri binari che verranno visualizzati sotto forma di numeri decimali sul display a 7 segmenti. Quando i tre interruttori sono a zero, vedremo uno "0" sul display, quando i tre interruttori sono a uno vedremo il numero "7", pari al valore binario più alto che possiamo inserire mediante tre interruttori.