Esercizi di apprendimento

```
c:\pathli~1\eserci~2\es17.asm
;
;Gioco di luci, un'altra temporizzazione.
;
;Si desidera realizzare una rotazione sequenziale nell'accensione di ogni led collegato alla
;porta B sulla scheda Trainer. Se RC0 = 0, la rotazione sarà da destra a sinistra e viceversa.
;Ogni led rimane acceso 0.25 secondi (250 ms)

                list    p=16F870                ;Tipo di processore
                include "P16F870.INC"   ;Definizione dei registri interni

Contatore               equ     0x20    ;Variabile per la temporizzazione

                org     0x00

Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona banco 1
                clrf    TRISB           ;Configura la Porta B come uscita
                movlw   0xFF
                movwf   TRISC           ;Configura la Porta C come ingresso
                movlw   b'00000110'
                movwf   OPTION_REG      ;Prescaler di 128 per il TMR0
                bcf     STATUS,RP0      ;Seleziona banco 0

```

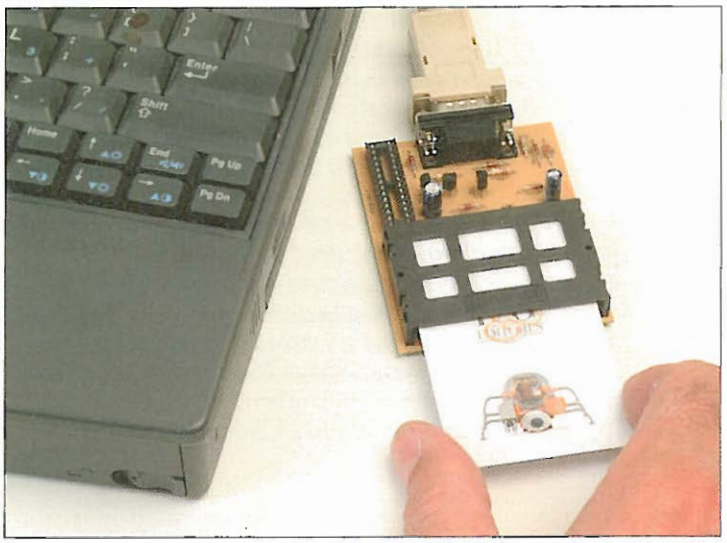
Realizzeremo ora un esercizio con le temporizzazioni, utilizzando il Timer 0 del microcontroller. Questo temporizzatore è già stato utilizzato nell'esercizio es16.asm per realizzare la temporizzazione anti-rimbalzo. Utilizzando il Timer 0 potremo fare in modo che il microcontroller esegua determinate azioni nei momenti che desideriamo. In questo esercizio faremo girare l'accensione di un diodo LED, lungo la barra dei LED della scheda di ingressi e uscite.
Fra un'accensione e l'altra di ogni diodo trascorrerà un periodo di 250 ms.
Con l'interruttore RC0 selezioneremo il verso di avanzamento del diodo LED acceso.

```
c:\pathli~1\eserci~2\es17.asm
Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona banco 1
                clrf    TRISB           ;Configura la Porta B come uscita
                movlw   0xFF
                movwf   TRISC           ;Configura la Porta C come ingresso
                movlw   b'00000110'
                movwf   OPTION_REG      ;Prescaler di 128 per il TMR0
                bcf     STATUS,RP0      ;Seleziona banco 0

                bsf     STATUS,C        ;Attiva il carry
Loop            call    Delay           ;Temporizza 250 ms
                btfsc   PORTC,0         ;RC0 = 0?
                goto    A_Destra        ;No, rotazione a destra
A_Sinistra      rlf     PORTB,F         ;Sì, rotazione a sinistra
                goto    Loop
A_Destra        rrf     PORTB,F         ;Rotazione a destra
                goto    Loop

```

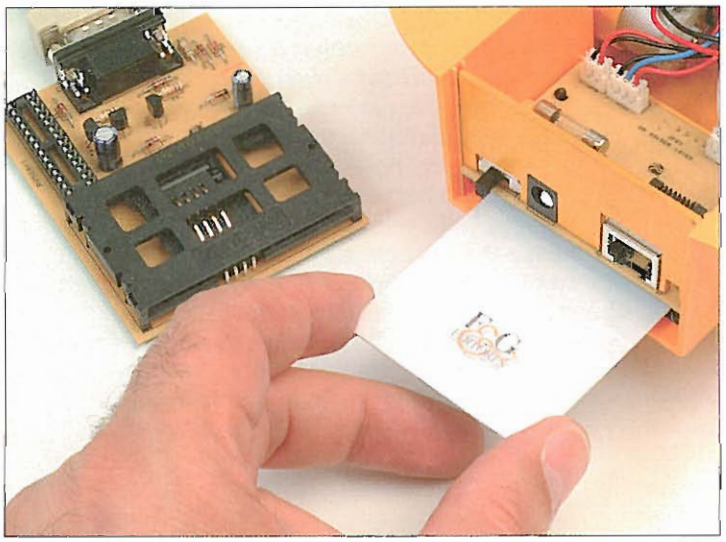
Come per tutti gli esercizi, il codice del programma inizia a partire dalla posizione 0x0B della memoria di programma, dato che nelle posizioni precedenti della memoria di programma risiede il software Uploader del microcontroller, che ha il compito di leggere il contenuto della Smartcard. Il primo passo consiste nel configurare RC0 come ingresso per poter leggere il dato inserito mediante l'interruttore, e la porta B come uscita per illuminare la barra di diodi LED. Viene anche configurato il prescaler a 128 per il Timer 0, impostando il valore adeguato sul registro OPTION.

```
c:\pathli~1\eserci~2\es17.asm
;*******************************************************************************
;Delay è una routine che realizza una temporizzazione di 250 ms, che è il tempo che devono
;rimanere accesi i leds.
;
;Se il PIC lavora ad una frequenza di 4 MHz, il TMR0 si aggiorna ogni µS. Se si desidera tempo-
;rizzare 25000 µS (25 ms) con un prescaler da 128, il TMR0 dovrà contare 195 eventi
;(195 * 128 = 24960). Il valore 195 equivale a 0xC3 e, dato che  il TMR0 è ascendente, lo
;dovremo caricare con il suo complemento (0x3c). Questa temporizzazione la dobbiamo ripetere
;10 volte per ottenere il totale desiderato (250000)

Delay           movlw   .10
                movwf   Contatore       ;Carica il contatore con 10
Delay_0         bcf     INTCON,T0IF     ;Resetta il flag di overflow del TMR0
                movlw   0x3c
                movwf   TMR0            ;Carica il TMR0
Delay_1         clrwdt                  ;Aggiorna il WDT
                btfss   INTCON,T0IF     ;Overflow del TMR0?
                goto    Delay_1         ;Non sono ancora passati i 25 ms
                decfsz  Contatore,F     ;Decremento il contatore. È stata ripetuta 10 volte?
                goto    Delay_0         ;Non ancora, temporizza 25 ms
                return                  ;Ora sì

                end                     ;Fine del programma sorgente
```

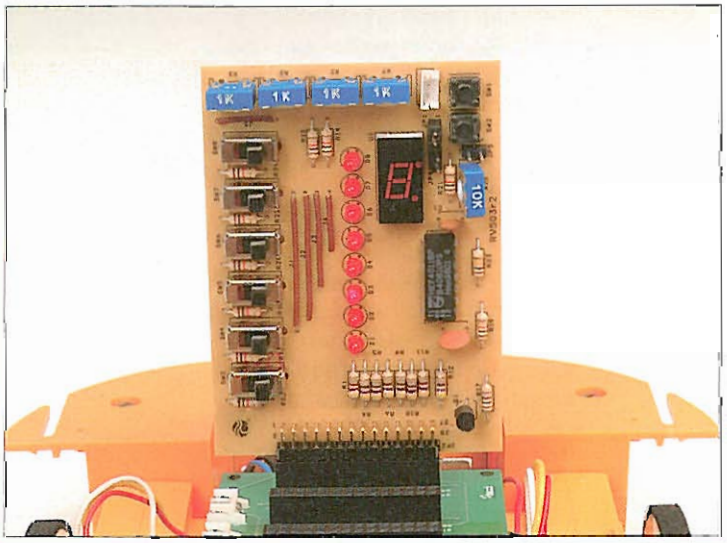
A questo punto il programma attiva il bit di carry del registro di stato, che sarà il bit utilizzato per le rotazioni e farà in modo che si attivino i diodi LED della porta B. Il programma permane in un ciclo infinito chiamato loop. All'inizio del ciclo si realizza la chiamata alla routine di temporizzazione da 250 ms. In seguito si legge lo stato di RC0 mediante un'istruzione di salto. In funzione dello stato di RC0 si entra in un ciclo in cui si ruota il carry verso destra (rrf) o verso sinistra (rlf).
In questo modo, il diodo LED si illuminerà lungo tutta la barra dei diodi.



Questo esercizio si trova sul CDROM sotto il nome di es17.asm. Copieremo il file in una directory dell'hard disk, insieme al file di definizione dei registri P16F870.inc. Compileremo l'esercizio utilizzando il software MPLAB. Dopo aver compilato e ottenuto il file es17.hex, lo scriveremo sulla Smartcard. Collegheremo la scheda di scrittura al computer, inseriremo la Smartcard con l'orientamento adeguato ed eseguiremo il programma ICPROG.



Dopo aver eseguito il software ICPROG, selezioneremo il tipo di dispositivo 24C16. In seguito, apriremo il file es17.hex e procederemo alla scrittura della Smartcard. Dopo averla scritta, la estrarremo dalla scheda di scrittura e la inseriremo nella scheda di alimentazione del robot. Alimenteremo il robot mediante pile o con un alimentatore a corrente continua. La scheda di ingressi e uscite dovrà essere inserita sulla scheda di interfaccia e il microcontroller dovrà essere sulla scheda di controllo con il software Uploader scritto.



Per provare l'esercizio, il jumper JP1, che attiva la barra dei diodi LED della scheda di ingressi e uscite, dovrà essere chiuso. I restanti diodi della scheda dovranno rimanere spenti. Dopo aver alimentato il robot, il microcontroller impiegherà qualche secondo a leggere il contenuto della Smartcard. Quindi procederà all'esecuzione dell'esercizio. Grazie all'interruttore SW3 controlleremo l'accensione progressiva dei diodi LED da destra a sinistra o viceversa.