

Assembler per PIC

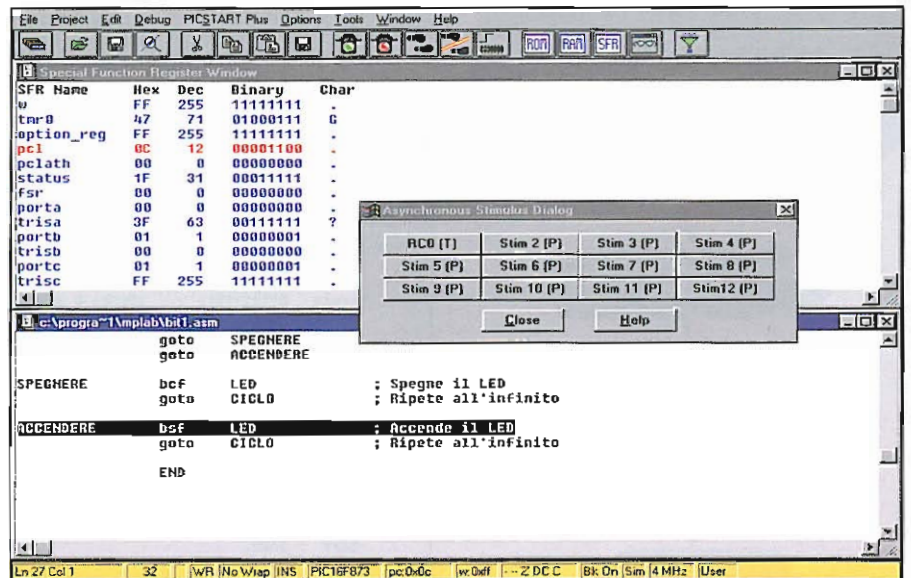
questo momento abbiamo fatto eseguire le istruzioni una ad una, ora sceglieremo l'opzione "Run>Animate", altrimenti non viene abilitato l'ingresso degli impulsi esterni che ci interessa. In questo modo il programma viene eseguito in modo continuo, e vengono visualizzati i registri e le loro modifiche. Agendo sugli elementi di ingresso (nel nostro caso la linea RC0) vedremo i cambiamenti su quelli di uscita. Nell'esempio della figura RC0 ha valore 1 quindi il valore di RB0 sarà anch'esso 1. Mentre il programma è in esecuzione in questa modalità, sulla parte bassa del video deve apparire una riga gialla, e la simulazione si può fermare premendo il tasto "F5".

Attesa dell'attivazione di un tasto

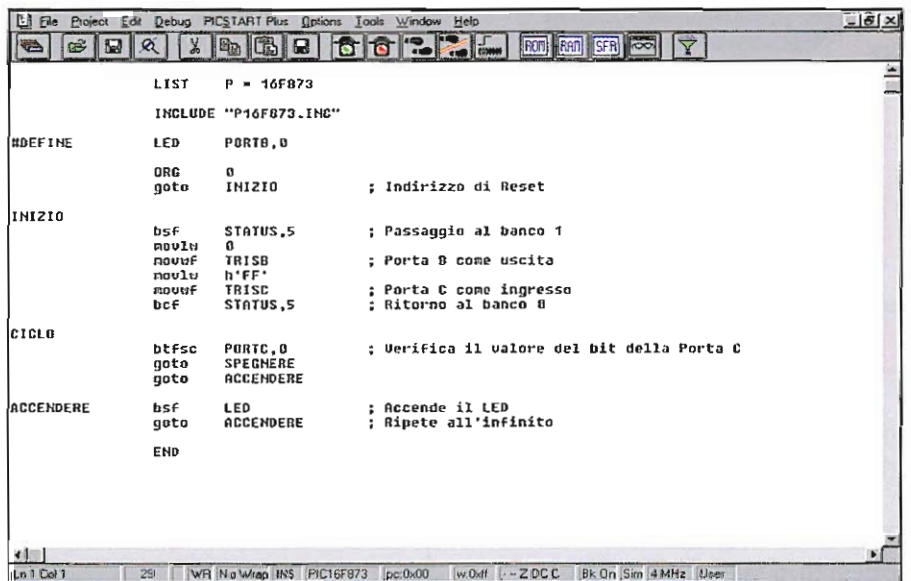
Per attendere l'attivazione di un tasto, come succede anche con il LetPicBasicPlus, non esiste alcuna istruzione speciale, ma lo si può fare in modo semplice e molto simile all'esempio precedente. Nell'esempio si utilizza l'istruzione "btfsc" per rimanere all'interno del ciclo sino a quando RC0 non avrà valore 1; in quel momento si accenderà un LED, e anche se andremo nuovamente a modificare il valore di RC0, l'uscita non cambierà più.

I treni di impulsi

Nemmeno in questo caso abbiamo il lavoro facilitato da



Una fase dell'esecuzione del programma.

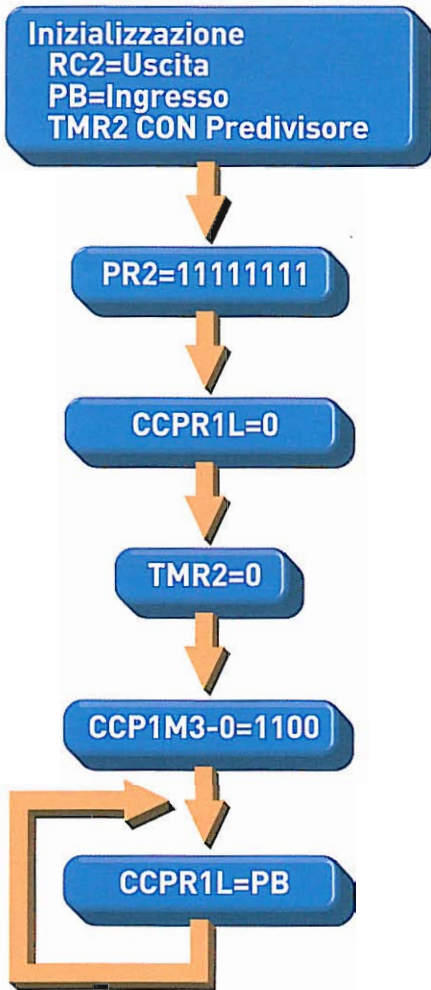


Per attendere l'attivazione di un tasto si utilizza l'istruzione "btfsc" oppure "btfss".

istruzioni particolari, però abbiamo già visto che il PIC16F873 dispone di due moduli CCP (Capture/Compare/Modulazione d'Amplitude degli Impulsi) il cui compito è esattamente quello

di acquisire o generare impulsi da alcuni piedini specifici.

Prenderemo ad esempio l'organigramma visto per la Modulazione dell'Amplitude degli Impulsi (PWM). I piedini di uscita devono essere obbligatoriamente RC2



Organigramma di un programma proposto per il lavoro con il PWM.

oppure RC1 (la seconda del modulo utilizzato).
 Dato che il predivisore del TMR2 non è specificato possiamo scegliere quello che vogliamo. I registri da definire sono quelli che impostano il periodo e l'ampiezza dell'impulso, a cui abbiamo assegnato un valore X. Il valore di CCPR1L lo prenderemo dagli interruttori, perché la simulazione dovrà essere realizzata in modo simile a quello che

```

LIST P = 16F873 ;Definiamo il nostro PIC
INCLUDE "P16F873.INC" ;File dei registri interni

ORG 0
goto INIZIO
ORG 5

;Programma principale
INIZIO
bsf STATUS,RP0 ;Passaggio al banco 1
movlw b'11111111' ;Porta B come ingresso
movwf PORTB

movlw b'11111011' ;RC2 uscita
movwf TRISC

movlw h'FF'
movwf PR2 ;carichiamo il PR2 con il periodo

bcf STATUS,RP0 ;Ritorno al banco 0

movlw b'00001100' ;Attiviamo il TMR2 e configuriamo il suo predivisore
movwf T2CON

clrf CCPR1L ;Inizializziamo a 0 i registri
clrf TMR2

movlw b'00001100' ;CCP1CON<5:4>=0
  
```

Prima parte del programma proposto.

```

clrf CCPR1L ;Inizializziamo a 0 i registri
clrf TMR2

movlw b'00001100' ;CCP1CON<5:4>=0
movwf CCP1CON ;Attiviamo il modo PWM

CICLO
movwf PORTB ;Carichiamo il CCPR1L con la
movwf CCPR1L ;durata dell'impulso
goto CICLO ;Ciclo che aggiorna il CCPR1L

END
  
```

Seconda parte del programma proposto.

abbiamo visto negli esempi precedenti, anche se dovremo definire più linee di ingresso. Il passaggio

da organigramma a programma è semplice, dato che si utilizzano istruzioni che già conosciamo.