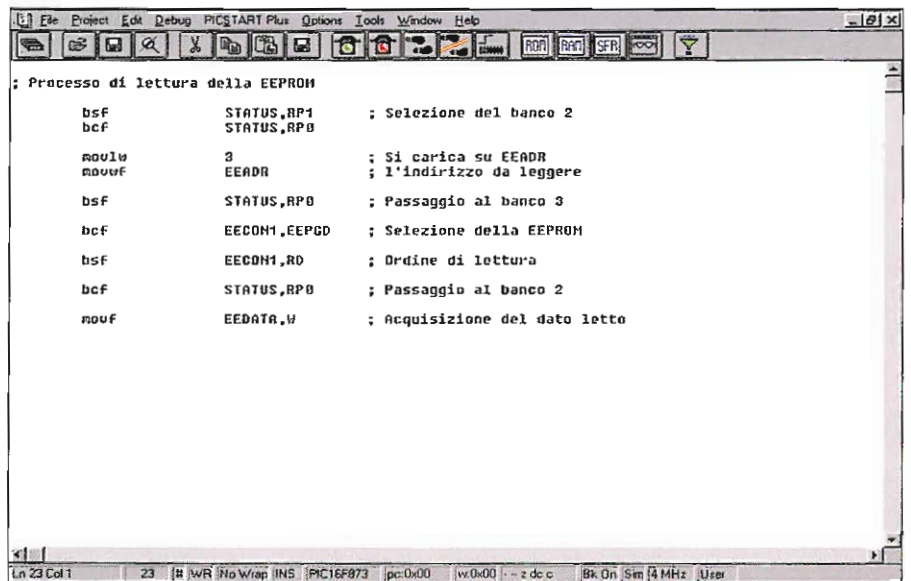


# Assembler per PIC

Con l'incremento di FSR si accede alla posizione successiva della memoria.

## Letture della EEPROM dei dati

Nemmeno la EEPROM dei dati dispone di istruzioni specifiche, però abbiamo già visto come la sua gestione sia un po' più complessa che nel caso della RAM. Fate riferimento agli organigrammi e ai passaggi che già conoscete, per seguire i programmi che vi presentiamo di seguito. I due programmi che presentiamo per la lettura della EEPROM realizzano la stessa funzione, con una sola differenza: mentre il primo è specifico per la lettura di un determinato indirizzo, in questo caso il "3", il secondo è proposto come una subroutine che si può richiamare tutte le volte che si vuole per differenti indirizzi, quindi risulta più generale. In nessuno dei due casi si tratta di programmi completi, quindi la loro compilazione produrrebbe errore. Tuttavia li possiamo memorizzare come file con estensione ".ASM" che



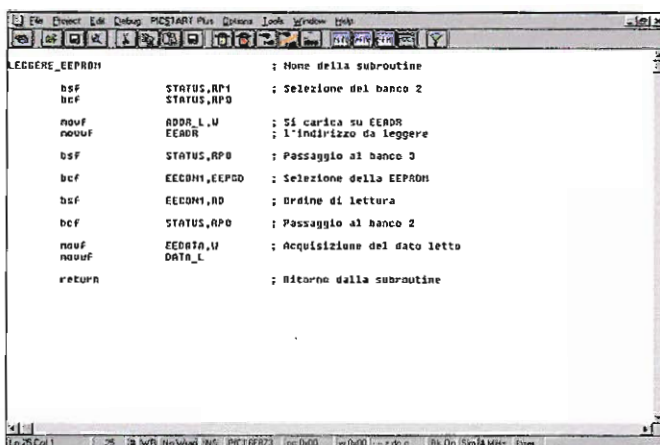
```
Processo di lettura della EEPROM
    bsf     STATUS,RP1      ; Selezione del banco 2
    bcf     STATUS,RP0
    movlw  3                ; Si carica su EEAR
    movwf  EEAR            ; l'indirizzo da leggere
    bsf     STATUS,RP0      ; Passaggio al banco 3
    bcf     EECOH1,EEPGR    ; Selezione della EEPROM
    bsf     EECOH1,RO       ; Ordine di lettura
    bcf     STATUS,RP0      ; Passaggio al banco 2
    movf   EEDATA,W        ; Acquisizione del dato letto
```

Processo di lettura della EEPROM dei dati.

verranno inseriti in seguito in altri programmi più completi. Nel caso della subroutine, ADDR\_L e DATA\_L, sono registri generali che devono essere definiti a partire dalla posizione 20h, e si utilizzano per il passaggio dei parametri alla subroutine. Notate che si seguono i passi di lettura così come sono stati spiegati, anche se si sono dovute introdurre le istruzioni di cambio di banco per ogni registro.

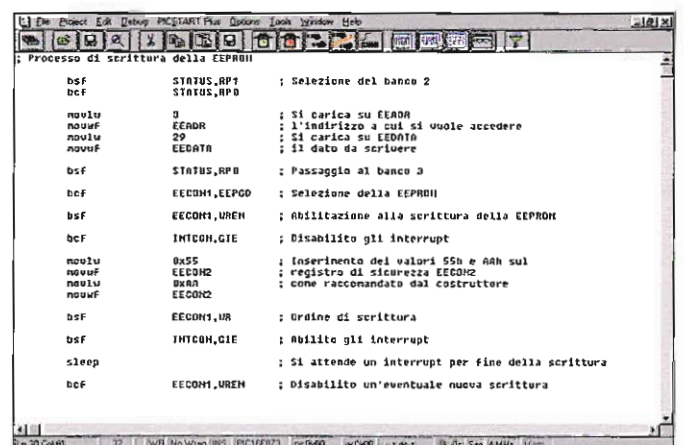
## Scrittura della EEPROM dei dati

Vediamo ora i programmi per la scrittura della EEPROM. Nella scrittura della EEPROM dei dati, succede lo stesso che nella lettura, si può fare come una subroutine più generale o dando ogni volta i dati specifici, in questo caso di indirizzo e dato da scrivere; anche



```
LEGGERE_EEPROM      ; None della subroutine
    bsf     STATUS,RP1      ; Selezione del banco 2
    bcf     STATUS,RP0
    movlw  ADDR_L,W        ; Si carica su EEAR
    movwf  EEAR            ; l'indirizzo da leggere
    bsf     STATUS,RP0      ; Passaggio al banco 3
    bcf     EECOH1,EEPGR    ; Selezione della EEPROM
    bsf     EECOH1,RO       ; Ordine di lettura
    bcf     STATUS,RP0      ; Passaggio al banco 2
    movf   EEDATA,W        ; Acquisizione del dato letto
    movwf  DATA_L
    return              ; Ritorno dalla subroutine
```

Subroutine di lettura della EEPROM dei dati.



```
Processo di scrittura della EEPROM
    bsf     STATUS,RP1      ; Selezione del banco 2
    bcf     STATUS,RP0
    movlw  3                ; Si carica su EEAR
    movwf  EEAR            ; l'indirizzo a cui si vuole accedere
    movlw  29              ; Si carica su EEDAT
    movwf  EEDATA          ; il dato da scrivere
    bsf     STATUS,RP0      ; Passaggio al banco 3
    bcf     EECOH1,EEPGR    ; Selezione della EEPROM
    bsf     EECOH1,UREN     ; Abilitazione alla scrittura della EEPROM
    bcf     INTCON,GIE      ; Disabilito gli interrupt
    movlw  0x55             ; Inserimento del valori 55h e AAh sul
    movwf  EECOH2          ; registro di sicurezza EECOH2
    movlw  0xA5            ; come raccomandato dal costruttore
    movwf  EECOH2
    bsf     EECOH1,WR       ; Ordine di scrittura
    bcf     INTCON,GIE      ; Abilito gli interrupt
    sleep                  ; Si attende un interrupt per fine della scrittura
    bcf     EECOH1,UREN     ; Disabilito un'eventuale nuova scrittura
```

Processo di scrittura della EEPROM dei dati.

# Assembler per PIC

```

SCRIVERE_EEPROM      : Nome della subroutine
bsf      STATUS,RP1      ; Selezione del banco 2
bcf      STATUS,RP0

movwf   ADDR_L,U        ; Si carica su EEDR
movwf   ADDR_L,U        ; Si carica su EEDATN
movwf   ADDR_L,U        ; il dato da scrivere

bsf      STATUS,RP0      ; Passaggio al banco 3
bcf      EECMD1,EEPCD    ; Selezione della EEPROM
bcf      EECMD1,UREN     ; Abilitazione alla scrittura della EEPROM
bcf      INTCON,GIE      ; Disabilito gli interrupt
movlw   0x55             ; Inserimento del valore 55h e AAh sul
movwf   EECMD2          ; registro di sicurezza EECMD2
movlw   0x0A             ; come raccomandato dal costruttore
movwf   EECMD2

bsf      EECMD1,WR      ; Ordine di scrittura
bcf      INTCON,GIE     ; Abilito gli interrupt
sleep   ; Si attende un interrupt per fine della scrittura
bcf      EECMD1,UREN    ; Disabilito un'eventuale nuova scrittura
return  ; Ritorno dalla subroutine
    
```

Subroutine di scrittura della EEPROM dei dati.

```

LIST      P = 16F073

RADIX     HEX

INCLUDE   "P16F073.INC"

ADDR_L    EQU    20      ; Var. aux. dell'indirizzo
DATA_L    EQU    21      ; Var. aux. del dato

goto     INIZIO
ORG      4
goto     INTERRUPT
ORG      5

INCLUDE   <eeprom.asm>      ; Routine di lettura e scrittura eeprom
INCLUDE   <eeprom.asm>

INIZIO    bcf      INTCON,GIE      ; Abilito gli interrupt
          bcf      INTCON,PEIE     ; Abilito gli interrupt secondari
          bcf      STATUS,RP0      ; Passaggio al banco 1
          bcf      PIE2,ECIE      ; Abilito l'Interrupt di fine scrittura
          bcf      STATUS,RP0      ; Passaggio al banco 0

          movlw   0                ; Indirizzo da leggere
          movwf   ADDR_L           ; Lettura della EEPROM
          call    LEGGERE_EEPROM

          movlw   1                ; Indirizzo da scrivere
          movwf   ADDR_L           ; Scrittura della EEPROM
          call    SCRIVERE_EEPROM

          goto    FINE
    
```

Prima parte del programma di prova delle subroutines trattate.

qui sarà necessario cambiare continuamente di banco.

## Prove delle espressioni in MPLAB

Il programma che presentiamo per la prova e la scrittura dei dati nella EEPROM è fra i più semplici, se avete capito le routines corrispondenti presentate in precedenza, dato che si tratta solamente di richiamare queste routines. Prima si legge un dato dall'indirizzo 0, poi lo stesso dato

letto è scritto all'indirizzo 1. La parte più complicata risiede da un lato nell'utilizzo dell'interrupt che dobbiamo abilitare all'inizio del programma, e il cui trattamento, una volta entrati nell'interrupt, consiste nel cancellare il flag che si sarà attivato; dall'altra parte, siccome i registri trattati corrispondono a banchi diversi, bisogna adattare continuamente i bit RP0 e RP1. Notate che sia i registri generali del PIC che le subroutines sono state incluse all'interno del programma con l'istruzione "INCLUDE". Per

la simulazione possiamo aprire la finestra dello stack, oltre a quella dei registri specifici e dei registri generali, per vedere come si producono le chiamate e la memoria EEPROM; essa si può modificare tramite Window→Modify per verificare che il programma funzioni correttamente.

Vedrete che eseguendo il programma (Debug→Run→Step) entreranno in gioco le videate dei tre differenti file che abbiamo scritto, secondo i passaggi delle loro istruzioni.

```

movwf   ADDR_L          ; Indirizzo da scrivere
call    SCRIVERE_EEPROM ; Scrittura della EEPROM

goto    FINE

INTERRUPT
bcf     STATUS,RP0      ; Passaggio al banco 0
bcf     STATUS,RP1
bcf     PIR2,EIF        ; Azzeramento del flag
bcf     STATUS,RP0
bcf     STATUS,RP1
retfie

nop
goto   FINE
END
    
```

Seconda parte del programma di prova delle subroutines.

Finestre per la simulazione.