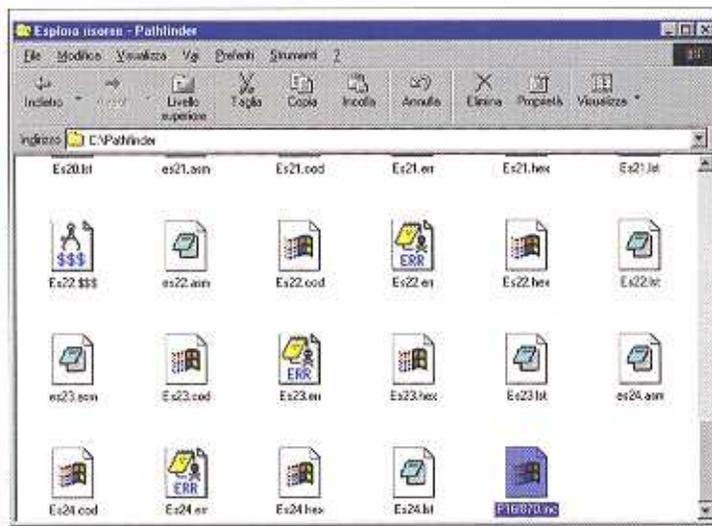


Esercizi di apprendimento

```
c:\pathfinder\source\2\es14.asm
1 ;
2 ;Programma combinatoriale
3 ;
4 ;Al valore impostato tramite gli interruttori RC0-RC2 della porta C, sommare la
5 ;costante 5. Il risultato si visualizza sui leds RB0-RB7 collegati alla porta B
6 ;
7 ;
8 List p16f870 ;Tipo di processore
9 Include "P16F870.INC" ;Definizione dei registri interni
10
11 ORG 0x80
12
13 Inizio
14 clrwf PORTB ;Cancella i latch di uscita
15 bsf STATUS,RP0 ;Seleziona il banco 1
16 cird b'00001111' ;Configura la Porta C come uscita
17 movwf TRISC ;Porta C configurata come ingresso
18 bsf STATUS,RP0 ;Seleziona il banco 0
19
20 Loop:
21 clrwff ;aggiorna il WDT
22 movwf PORTC,W ;Carica lo stato degli ingressi RC0-RC2
23 addlw .5 ;Somma 5
24 movwf PORTB ;Visualizza il risultato su RB0-RB7
25 goto Loop
26
27 end ;Fine del programma sorgente
```

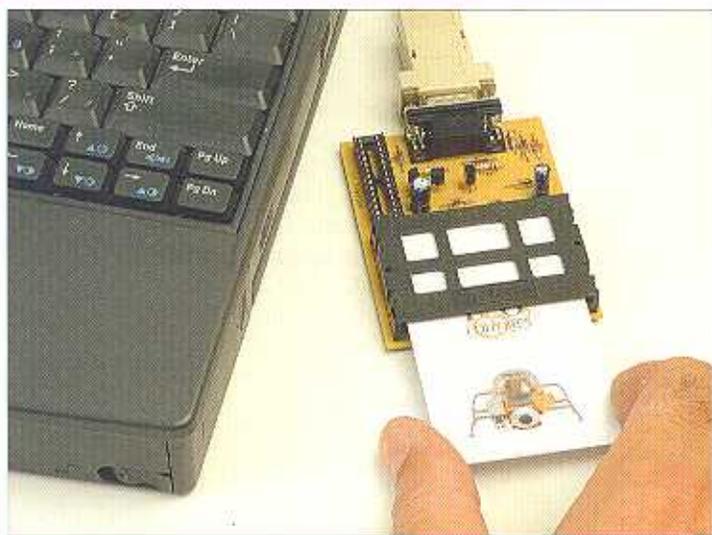
Nell'immagine possiamo vedere il codice di un altro esercizio per la gestione degli ingressi e le uscite (file es14.asm). Viene catturato il dato presente sui primi tre interruttori della porta C (SW3 - SW4 - SW5), si somma il numero 5 a questo dato, e si visualizza il risultato in binario sui diodi LED collegati alla porta B del microcontroller. Il programma configura i primi tre pin della porta C come ingressi, la porta B come uscita, dopodiché entra in ciclo infinito in cui legge continuamente il contenuto della porta C, esegue l'operazione sul dato e invia il risultato alla porta B per la visualizzazione sui diodi LED.



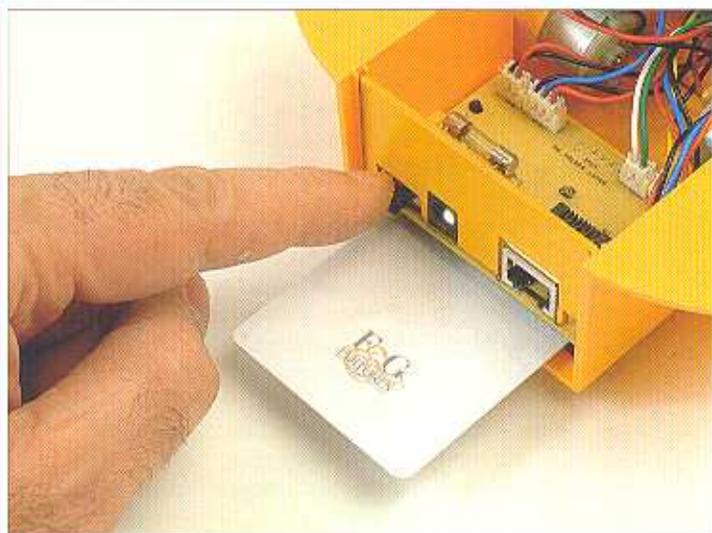
Tutti gli esercizi di Pathfinder devono essere compilati partendo dall'indirizzo 0x80 della memoria di programma. Questa funzione è realizzata dalla direttiva ORG 0x80. L'impostazione di questo indirizzo di partenza è dovuto al fatto che i primi indirizzi della memoria di programma sono già occupati dal programma scritto in precedenza sul microcontroller, cioè il programma uploader, che ha il compito di leggere il contenuto della memoria EEPROM e autoprogrammarlo nella memoria del microcontroller. Il file P16F870.inc, incluso nel programma, contiene tutte le definizioni dei registri e i bit di controllo del microcontroller. Possiamo aprirlo con un qualsiasi programma di editor per vederne il contenuto.

```
c:\pathfinder\source\2\p16f870.inc
47 ----- Register Files -----
48
49 INDF EQU H'0000'
50 IIR0 EQU H'0001'
51 PC EQU H'0002'
52 STATUS EQU H'0003'
53 FSR EQU H'0004'
54 PORTA EQU H'0005'
55 PORTB EQU H'0006'
56 PORTC EQU H'0007'
57 PORTD EQU H'0008'
58 INTCON EQU H'0009'
59 PIR1 EQU H'000C'
60 PIR2 EQU H'000D'
61 TRISL EQU H'000E'
62 TRISH EQU H'000F'
63 T1CON EQU H'0010'
64 INR2 EQU H'0011'
65 T2CON EQU H'0012'
66 CDFR1L EQU H'0015'
67 CDFR1H EQU H'0016'
68 CDFR2CON EQU H'0017'
69 ROSTA EQU H'0018'
70 TRSER EQU H'0019'
71 RSR EQU H'001A'
72 ADDRESS EQU H'001E'
73 ADDR0 EQU H'001F'
74
75 OPT10K_REG EQU H'00B1'
```

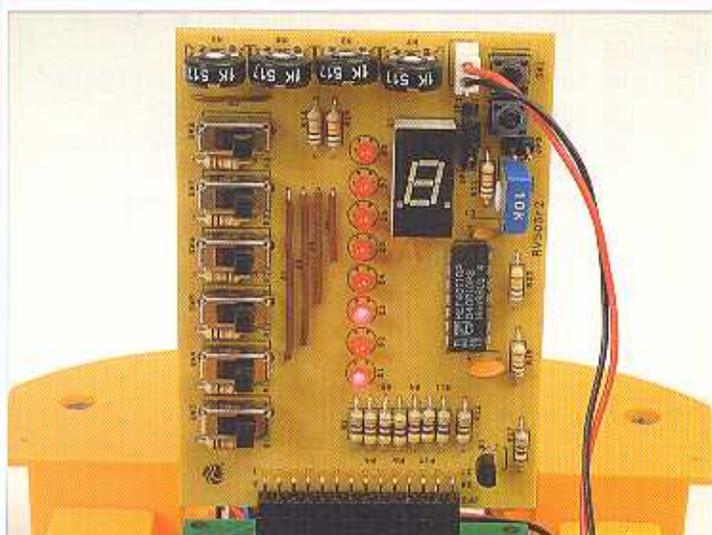
L'immagine mostra una parte del contenuto del file dove possiamo vedere la definizione dei registri di controllo del Banco 0 del microcontroller. Includendo il file P16F870.inc nel nostro programma, eviteremo di dover definire manualmente i registri di controllo, ma sarà sufficiente assegnare un nome ai registri di lavoro che potremo generare a partire dall'indirizzo 0x20 della memoria RAM, la prima posizione libera per i registri di lavoro.



Dopo aver compilato il file es14.asm e aver ottenuto il file esadecimale es14.hex, passeremo alla scrittura della Smartcard. La dovremo inserire nella scheda di scrittura con l'orientamento adeguato. Selezioneremo nel programma ICPROG la memoria modello 24C16, apriremo il file es14.hex e daremo il via alla programmazione della Smartcard (opzione Programma Tutto). Vi consigliamo di verificare la programmazione mediante l'opzione Verifica del menù Comandi.



Dopo aver scritto la Smartcard, è necessario inserirla nella scheda di alimentazione del robot, con l'orientamento adeguato. Per fare in modo che il programma venga eseguito dal microcontroller, quest'ultimo dovrà essere presente sulla scheda di controllo, ed essere stato precedentemente programmato con il file uploader.hex. Dopo aver alimentato il robot, il microcontroller impiegherà alcuni secondi a leggere il contenuto della Smartcard e a iniziare l'esecuzione del programma. La scheda di ingressi e uscite dovrà essere inserita sul connettore JP13 della scheda di interfaccia.



Per eseguire il programma è necessario chiudere il jumper JP1 e lasciare il resto dei jumper della scheda aperti, in questo modo abiliteremo solamente i diodi LED. A questo punto inseriremo i dati binari mediante i primi tre interruttori e vedremo sui diodi LED il dato inserito sommato a cinque, la quantità costante. Il programma rimarrà in esecuzione per un tempo indefinito.