

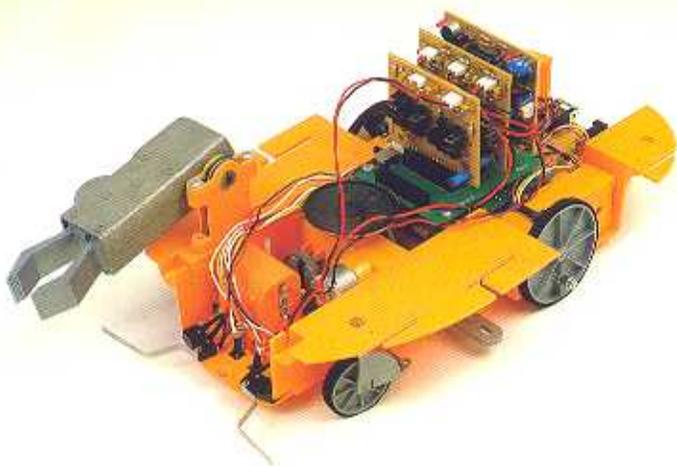
Esercizi con braccio e pinza (I)

```
67:.....
68:;funzione dedicata al movimento di salita del braccio
69 BRACCIO_ALZARE:
70      btfss   INTCOH, 2      ;overflow del timer
71      return
72      bcf     INTCOH, 2
73      decfsz  TEMPO, 1
74      return
75      movf   PORTB, 0
76      andlw  b'11000000'
77      movwf  ANK
78      movlw  0
79      subwf  ANK, 0
80      btfss  STATUS, 2
81      goto  SPEGNERE_BRACCIO
82      goto  ACCENDERE_BRACCIO
83 ACCENDERE_BRACCIO:
84      bcf     PORTB, 7
85      bcf     PORTB, 6
86      movlw  TON_BRACCIO
87      movwf  TEMPO
88      return
89 SPEGNERE_BRACCIO:
90      bcf     PORTB, 6
91      bcf     PORTB, 7
92      movlw  TOFF_BRACCIO
93      movwf  TEMPO
94      return
95
96
```

Questa funzione ha il compito di attivare il motore del braccio per farlo scendere. Si tratta di una routine generica che potremo utilizzare in qualsiasi programma che impieghi il braccio del robot. L'attivazione del motore per far alzare il braccio è realizzata inviando un '1' tramite il pin RB6 del microcontrollore e uno '0' tramite il pin RB7. Per fermare il motore bisognerà impostare a '0' entrambi i pin. Questa funzione utilizza la modulazione di ampiezza degli impulsi per poter controllare meglio la velocità del movimento del braccio.

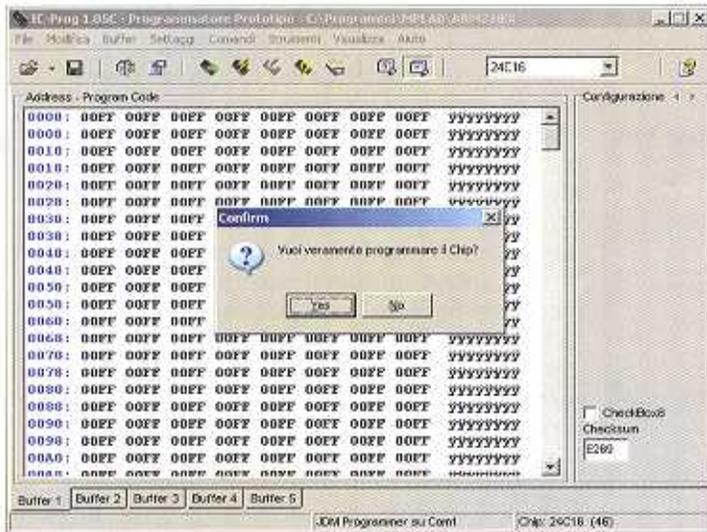
```
66:.....
67:;funzione dedicata al movimento di discesa del braccio
68 BRACCIO_ABISSINO:
69      btfss   INTCOH, 2      ;overflow del timer
70      return
71      bcf     INTCOH, 2
72      decfsz  TEMPO, 1
73      return
74      movf   PORTB, 0
75      andlw  b'11000000'
76      movwf  ANK
77      movlw  0
78      subwf  ANK, 0
79      btfss  STATUS, 2
80      goto  SPEGNERE_BRACCIO2
81      goto  ACCENDERE_BRACCIO2
82 ACCENDERE_BRACCIO2:
83      bcf     PORTB, 7
84      bcf     PORTB, 6
85      movlw  TON_BRACCIO
86      movwf  TEMPO
87      return
88 SPEGNERE_BRACCIO2:
89      bcf     PORTB, 6
90      bcf     PORTB, 7
91      movlw  TOFF_BRACCIO
92      movwf  TEMPO
93      return
94
95      END                      ;fine del programma sorgente
96
```

Questa seconda funzione ha il compito di realizzare il movimento di discesa del braccio. Per far abbassare il braccio, bisogna inviare uno '0' tramite il pin RB6 e un '1' tramite RB7. La modulazione di ampiezza degli impulsi di questa funzione si realizza mediante le variabili TON_BRACCIO e TOFF_BRACCIO, definite all'inizio del programma. Tanto maggiore sarà il valore della variabile TON_BRACCIO rispetto a TOFF_BRACCIO, maggiore sarà la velocità del motore. In funzione dello stato delle batterie del robot, potrà essere necessario modificare questi valori per fare in modo che il motore abbia maggiore velocità, ad esempio aumentando il valore di TON_BRACCIO.



Compileremo l'esercizio con MPLAB e lo scriveremo sulla Smartcard mediante il programma ICPROG. Per verificare l'esercizio dovremo avere la scheda di controllo del braccio collegata a JP15 sulla scheda di interfaccia. Il motore del braccio sarà sul connettore JP3 della scheda di controllo del braccio e della pinza; una prima attivazione del finecorsa farà alzare il braccio. Dopo che il braccio sarà stato sollevato, una nuova attivazione del finecorsa lo farà ridiscendere. Il programma si ripete in modo ciclico. Se notiamo che il braccio ha degli attriti e fatica a realizzare i movimenti di salita e discesa, è consigliabile modificare il valore delle variabili all'inizio del programma, ad esempio aumentando il valore della variabile TON_BRACCIO e di IMPULSO_ALZARE di 2 unità.

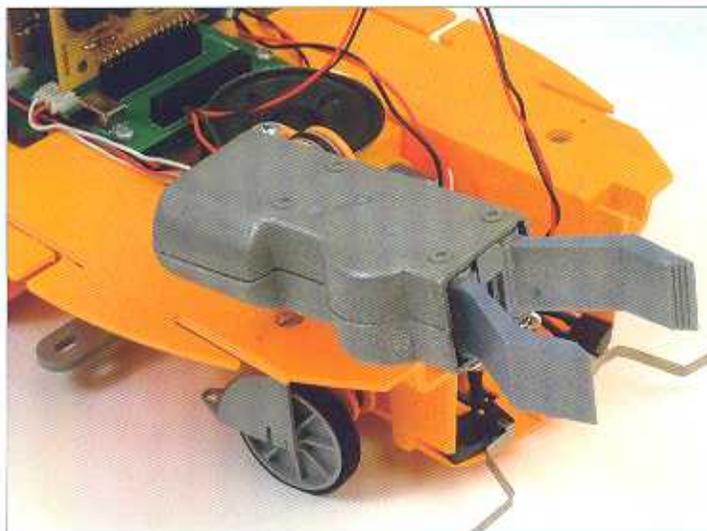
Esercizi con braccio e pinza (II)



Una volta ottenuto il file esadecimale, utilizzeremo ICPROG per la programmazione della Smartcard (dispositivo 24C16), utilizzando la scheda di scrittura collegata al PC. Dopo aver scritto la Smartcard, la inseriremo nella scheda di alimentazione di Pathfinder. Ogni volta che utilizziamo la Smartcard dobbiamo fare attenzione al verso di inserzione, dato che deve essere inserita con l'orientamento corretto.



Per provare l'esercizio, la scheda di controllo del braccio e della pinza dovrà essere collegata sulla scheda di interfaccia. Il ponticello JP2 della scheda di controllo del braccio dovrà essere chiuso, come si può vedere nell'immagine. In questo modo il microcontroller specifico PIC12C508, presente sulla scheda, farà funzionare la pinza in modo temporizzato. Infine, il motore di controllo della pinza dovrà essere collegato al connettore JP4 della scheda.



Dopo aver alimentato il robot attenderemo qualche secondo per fare in modo che il microcontroller di Pathfinder legga il contenuto della Smartcard. Dopo aver letto il programma, inizierà la sua esecuzione. Attiveremo il fincorsa e la pinza si chiuderà. Passato un certo tempo, la pinza si aprirà automaticamente. La pinza di Pathfinder farà in modo che il robot possa prendere oggetti poco pesanti e trasportarli.

Esercizi con braccio e pinza (III)

```
c:\progra-1\mplab\asm1.asm
1:.....
2:;Programma di gestione della pinza di Pathfinder in modo controllato
3:
4:          LIST      p=16F87B          ;Tipo di processore
5:          include "16F87B.INC"        ;Definizione dei registri interni
6:
7:  TEMPO    EQU     0x70
8:
9:          ORG      0x00
10:
11:.....
12:;inizio del programma. Configurazioni
13:inizio    bcf      STATUS,RPD        ;Seleziona il banco 1
14:          movlw   0x0F                ;Porta digitale
15:          movwf   ANCONO              ;Porta digitale
16:          movlw   b'00000111'        ;Configurazione della Porta
17:          movwf   PORTO               ;Configurazione della Porta
18:          movlw   0x0F                ;Prescaler 256 per il timer
19:          movwf   TMR0               ;Seleziona il banco 0
20:          bcf      STATUS,RPD        ;Seleziona il banco 0
21:          bcf      PORTO, 0           ;Pinza disattivata
22:
23:.....
```

Realizzeremo un secondo esercizio di controllo della pinza di Pathfinder. In questo caso gestiremo la pinza nel modo controllato; attraverso questo metodo di funzionamento si chiuderà la pinza mediante un fronte di salita inviato sul pin RA3 del microcontroller, e grazie a un secondo fronte inviato sullo stesso pin, la pinza si aprirà. Come nel precedente esercizio utilizzeremo il finecorsa collegato su JP7 della scheda di interfaccia per controllare l'apertura e la chiusura della pinza.

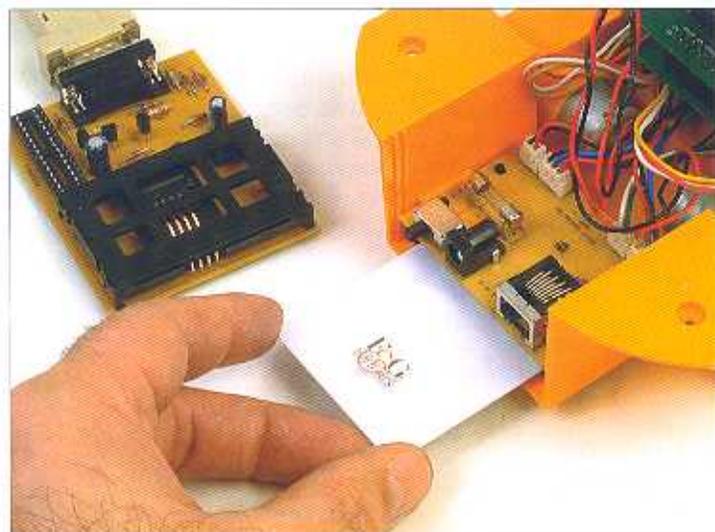
```
c:\progra-1\mplab\asm2.asm
22:.....
23:;Ciclo principale del programma
24:;Ciclo principale del programma
25:WAIT_FINECORSA: btfsz PORTA, 1
26:                goto  WAIT_FINECORSA
27:                bcf      PORTO, 1    ;Si invia un fronte che provocherà
28:                nop                    ;l'apertura e la chiusura della pinza
29:                nop
30:                nop
31:                bcf      PORTO, 1
32:                call    DELAY         ;Si eliminano i rimbalzi del finecorsa
33:                goto  WAIT_FINECORSA
34:
35:.....
```

Il ciclo principale del programma, verifica in continuazione lo stato del pin RA1 su cui è collegato il finecorsa. Ogni volta che si attiva il finecorsa; verrà inviato un fronte di salita sul pin RA3 del microcontroller. Se la pinza in quel momento è chiusa, il fronte di salita la farà aprire, e se è aperta l'impulso la farà chiudere. Il programma si ripete in modo infinito.

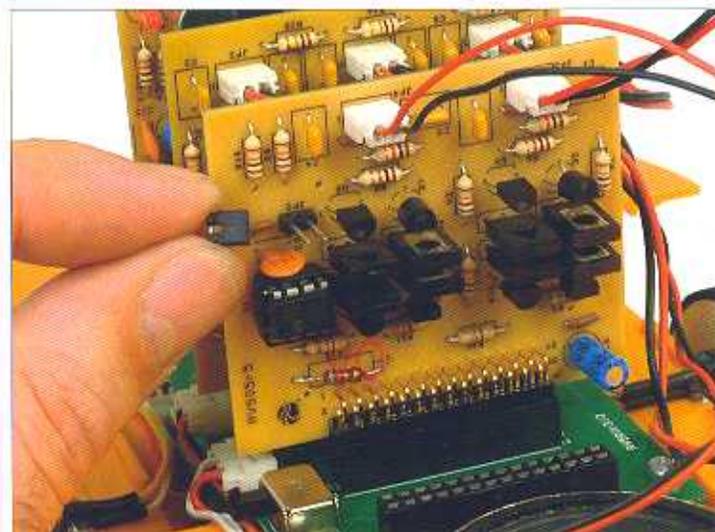
```
c:\progra-1\mplab\asm3.asm
35:.....
36:;Rutine di temporizzazione di ts
37:DELAY:        movlw   .100
38:              movwf  TEMPO
39:              movlw   0x01          ;temporizzazione da 100
40:              movwf  TMR0
41:              bcf      INTCOM, 2
42:              decfsz  INTCOM, 2
43:              goto   DEL
44:              decfsz  TEMPO, 1
45:              goto   DEL_10        ;Si ripete il ciclo 100 volte
46:              return
47:
48:.....
```

Allo scopo di eliminare i rimbalzi del finecorsa, e lasciare il tempo necessario all'apertura e alla chiusura della pinza, fra gli impulsi del finecorsa è stata inserita una routine di temporizzazione di un secondo, controllata con il Timer 0 del microcontroller. Questa routine viene chiamata ogni volta che si attiva il finecorsa, per poter aspettare un secondo, prima di ritornare a testare lo stato del sensore.

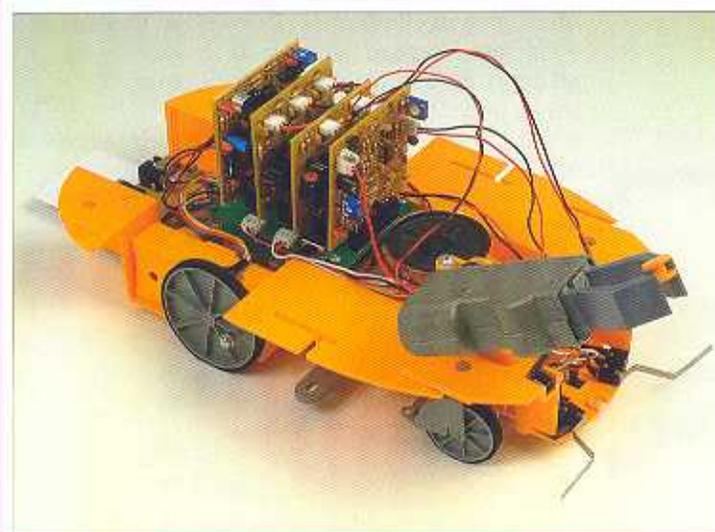
Esercizi con braccio e pinza (III)



Scriveremo l'esercizio con il programma MPLAB e lo compileremo per ottenere il file esadecimale. Utilizzeremo il software di scrittura ICPROG per trasferire il file alla Smartcard mediante la scheda di scrittura. Dopo aver programmato la Smartcard, la inseriremo sulla scheda di alimentazione di Pathfinder con l'orientamento adeguato.



Per fare in modo che la pinza del braccio di Pathfinder funzioni in modo controllato bisogna togliere il jumper JP2 della scheda di controllo del braccio e della pinza. Grazie a questo, il microcontroller PIC12C508 di questa scheda interpreterà i segnali inviati sul pin RA3 del microcontroller principale, come segnale di controllo sia dell'apertura che della chiusura della pinza. Per verificare l'esercizio, la scheda di controllo del braccio e della pinza deve essere inserita sul connettore JP15 della scheda di interfaccia e il motore della pinza collegato sul terminale JP4 di questa scheda.



Un'ulteriore verifica dell'esercizio, dovrà essere fatta alimentando il robot e attendendo qualche secondo in modo che il microcontroller possa leggere il programma dalla Smartcard. Grazie all'attivazione del finecorsa controlleremo sia l'apertura che la chiusura della pinza. Un primo impulso farà chiudere la pinza, la seconda attivazione farà aprire la pinza rilasciando l'oggetto che aveva in precedenza afferrato.

Esercizi con braccio e pinza (IV)

```
1 ;Programma di controllo del braccio e della pinza di Pathfinder. Il finecorsa collegato a RA1
2 ;preziona la salita e la discesa del braccio. Il finecorsa collegato su RA2 preziona
3 ;l'apertura e la chiusura della pinza
4
5 LIST p=16F87B ;Tipo di processore
6 INCLUDE "16F87B.INC" ;Definizione dei registri interni
7
8 ORG 0
9 TEMPO EQU 0x20
10 TEMPO2 EQU 0x22
11 TEN_BRACCIO EQU -5
12 TOFF_BRACCIO EQU -18
13 IMPULSO_ALZARE EQU -2
14 IMPULSO_ABBASSARE EQU -3
15
16 ORG 0x000000
17
```

Effettueremo ora un altro esercizio con il braccio di Pathfinder; in esso combineremo diversi esercizi che abbiamo già utilizzato in precedenza. Con questo programma controlleremo sia la salita che la discesa del braccio, e l'apertura e la chiusura della pinza. Utilizzeremo il finecorsa collegato su JP7 per il controllo del braccio, e il finecorsa su JP8 per il controllo dell'apertura e della chiusura della pinza. All'inizio del programma imposteremo i parametri di configurazione TON_BRACCIO, TOFF_BRACCIO, IMPULSO_ALZARE e IMPULSO_ABBASSARE, che compiono le stesse funzioni di controllo del motore del braccio dell'esercizio già realizzato arm1.asm.

```
18 ;Inizializza
19 ;Inizializza
20 movlw 0x07
21 movwf ACC0H
22 movlw 0x00001111
23 movwf PORTA
24 clrf TRISB ;Porta B si configura come uscita
25 movlw 0x00000001
26 movwf TMR0 ;Prescaler per il Timer0
27 bcf STATUS,RPB ;Selezione banca 0
28 movlw TOFF_BRACCIO
29 movwf TEMPO
30 clrf PORTB
31 movlw 0x00100001
32 movwf TCON ;Si spegnono i motori
33 clrf TRISB
34 bcf PORTB,0
35 bcf PORTA,1
```

Questo programma utilizzerà i pin RA1 e RA2 come ingresso per finecorsa, e RA3 come uscita per il segnale di controllo della pinza. La porta B del microcontroller verrà configurata come uscita per la gestione dei motori. L'esercizio utilizza il Timer 0 e il Timer 1 per il controllo della modulazione dell'ampiezza degli impulsi dei motori e per i tempi di salita e di discesa del braccio del robot.

```
37 ;
38 ;
39 CICLO1: clrf PORTB ;Ciclo principale del programma
40 btfsc PORTA, 2 ;SI testa il finecorsa RA2
41 call PINZA ;per l'apertura/chiusura della pinza
42 btfsc PORTA, 1 ;SI attende il segnale su RA1
43 goto CICLO1 ;per alzare il braccio
44 movlw IMPULSO_ALZARE
45 movwf TEMPO2
46 clrf TRISB
47 bcf PORTB,0
48 ;
49 ATTESA_ALZARE_BRACCIO:
50 call BRACCIO_MUOVE ;SI alza il braccio del robot
51 btfsc PORTB,0
52 goto ATTESA_ALZARE_BRACCIO
53 decfsz TEMPO2,1
54 goto ALZARE_BRACCIO
55 ;
56 CICLO2:
57 btfsc PORTA, 2 ;SI testa il finecorsa RA2
58 call PINZA ;per l'apertura/chiusura della pinza
59 btfsc PORTA, 1 ;SI attende il secondo segnale su RA1
60 goto CICLO2 ;per abbassare il braccio
61 movlw IMPULSO_ABBASSARE
62 movwf TEMPO2
63 clrf TRISB
64 bcf PORTB,0
65 ;
66 ATTESA_ABBASSARE_BRACCIO:
67 call BRACCIO_MUOVE ;SI abbassa il braccio del robot
68 btfsc PORTB,0
69 goto ATTESA_ABBASSARE_BRACCIO
70 decfsz TEMPO2,1
71 goto ABBASSARE_BRACCIO
72
```

Questo è il ciclo principale del programma. Si testa lo stato dei segnali RA1 e RA2 che contengono lo stato dei finecorsa. Un '1' su RA2 richiamerà la routine, che ha il compito di inviare un impulso per RA3 per provocare l'apertura o la chiusura della pinza. L'attivazione del finecorsa RA1 fa iniziare il movimento di salita del braccio e una seconda attivazione di questo finecorsa provocherà la discesa del braccio.

Esercizi con braccio e pinza (IV)

```
c:\progr\lupia\lupia.asm
72 :.....
73 :Funzione di apertura e chiusura della pinza
74 PINZA:      bcf   PORTA, 3      ;! Isola un fronte che provocherà
75             nop             ;l'apertura e la chiusura della pinza
76             nop
77             nop
78             nop
79             bcf   PORTA, 3
80             return
81
```

La funzione che vediamo nell'immagine ha il compito di inviare un fronte di salita tramite il pin RA3 del microcontroller che provocherà l'apertura o la chiusura della pinza. Questo esercizio gestisce la pinza in modo controllato, dato che il jumper JP2 della scheda di controllo del braccio e della pinza dovrà essere tolto dalla sua posizione. Se il jumper rimane montato la pinza funzionerà in modo temporizzato, cioè un'attivazione del finecorsa RA2 provocherà un ciclo completo di apertura e chiusura.

```
c:\progr\lupia\lupia.asm
82 :.....
83 :Funzione dedicata al movimento di salita del braccio
84 BRACCIO_ALZARE:
85             btfss  IMCON, 2      ;overflow del timer
86             return
87             bcf   IMCON, 2
88             decfsz TEMPO, 1
89             return
90             movf  PORTA, 0
91             movl  b'11000000'
92             movwf 0
93             movl  0
94             subwf 0, 0
95             btfss STATUS, 2
96             goto  SPENDERE_BRACCIO
97             goto  ACCENDERE_BRACCIO
98 ACCENDERE_BRACCIO:
99             bcf   PORTA, 7
100            bcf   PORTA, 6
101            movlw TON_BRACCIO
102            movwf TEMPO
103            return
104 SPENDERE_BRACCIO:
105            bcf   PORTA, 6
106            bcf   PORTA, 7
107            movlw TOFF_BRACCIO
108            movwf TEMPO
109            return
110
```

Questa funzione ha il compito di controllare il motore del braccio nel movimento di salita. Possiamo modificare la velocità di salita e di discesa del braccio con le variabili che si trovano all'inizio del file. Se aumentiamo il valore della variabile TON_BRACCIO, il motore si muoverà più rapidamente e il braccio avrà più forza. Modificando il valore della variabile IMPULSO_ALZARE e IMPULSO_ABBASSARE controlliamo quanti secondi rimane attivato il motore del braccio sia nel movimento di salita che in quello di discesa.

```
c:\progr\lupia\lupia.asm
111 :.....
112 :Funzione dedicata al movimento di discesa del braccio
113 BRACCIO_ABBASSARE:
114             btfss  IMCON, 2      ;overflow del timer
115             return
116             bcf   IMCON, 2
117             decfsz TEMPO, 1
118             return
119             movf  PORTA, 0
120             movl  b'11000000'
121             movwf 0
122             movl  0
123             subwf 0, 0
124             btfss STATUS, 2
125             goto  SPENDERE_BRACCIO2
126             goto  ACCENDERE_BRACCIO2
127 ACCENDERE_BRACCIO2:
128             bcf   PORTA, 7
129             bcf   PORTA, 6
130             movlw TON_BRACCIO
131             movwf TEMPO
132             return
133 SPENDERE_BRACCIO2:
134             bcf   PORTA, 6
135             bcf   PORTA, 7
136             movlw TOFF_BRACCIO
137             movwf TEMPO
138             return
139
140             END                ;fine del programma sorgente
```

Per provare l'esercizio, la scheda di controllo del braccio e della pinza di Pathfinder dovrà essere inserita sul connettore JP15 della scheda di interfaccia. Il motore del braccio sarà collegato sul connettore JP3 della scheda di controllo del braccio e il motore della pinza sul connettore JP4. Dopo avere scritto l'esercizio con MPLAB lo compileremo e lo scriveremo sulla Smartcard utilizzando il software ICPROG. Per verificare l'esercizio, attiveremo i finecorsa anteriori del robot, controllando il braccio e la pinza.