

Esercizi con braccio e pinza (I)

```
c:\progr-1\mqab\arm1.asm
1 ;Programma che alza e abbassa il braccio di Pathfinder. Un primo impulso su finecorsa
2 ;JP7 (RA1) provoca la salita del braccio. Un secondo impulso fa sì che il braccio si
3 ;abbassi
4
5
6
7 AUX EQU 8c20 ;tipo di processore
8 TEMP EQU 8c21 ;definizioni dei registri interni
9 TEMP2 EQU 8c22
10 TON_BRACCIO EQU .5
11 TOFF_BRACCIO EQU -.10
12 IMPULSO_ALZARE EQU .3
13 IMPULSO_ABBASSARE EQU .3
14
15 DMS 0x00
16
```

Realizzeremo un primo esercizio con il braccio di Pathfinder. Questo programma servirà a far alzare o abbassare il braccio al robot ogni volta che viene rilevata l'attivazione del finecorsa del robot collegato sul connettore JP7 della scheda di interfaccia (pin RA1 del microcontroller). All'inizio del programma troviamo quattro variabili con cui potremo configurare i tempi applicati al motore del braccio. Mediante le variabili TON_BRACCIO e TOFF_BRACCIO si controlla la modulazione di ampiezza degli impulsi del motore. Le variabili IMPULSO_ALZARE e IMPULSO_ABBASSARE controllano rispettivamente il tempo in cui rimane acceso il motore nei movimenti di salita e di discesa. Tanto maggiore sarà il valore, più tempo rimarrà acceso il motore.

```
c:\progr-1\mqab\arm1.asm
17 INIZIO bsf STATUS_0P4 ;Selezione banco 1
18 movwf 0x00
19 movwf 0x00
20 movwf 0x00
21 movwf PORTA ;Porta A come ingresso
22 clrwf TRISA ;Porta B si configura come uscita
23 movwf 0'10000001'
24 movwf TMR0 ;Prescaler per il Timer0
25 bcf STATUS_0P0 ;Selezione banco 0
26 movwf TOFF_BRACCIO
27 movwf TEMP
28 clrwf PORTB ;Si spengono i motori
29 movwf 0'00100001'
30 movwf TMR1
31 clrwf TMR0L
32 clrwf TMR1L
33 bcf PIR1, INT1IF
34
```

Il programma utilizzerà la porta A come ingresso digitale, dato che impiegheremo il finecorsa collegato su RA1. Configureremo anche la porta B come uscita, dato che il motore del braccio è gestito mediante i pin RB6 e RB7 del microcontroller. Si assegna il prescaler al Timer 0 che verrà utilizzato per controllare la modulazione di ampiezza degli impulsi applicata al motore. Configureremo anche il Timer 1, dato che sarà il temporizzatore utilizzato per il controllo del tempo di attivazione del motore, sia per il movimento di salita del braccio che per quello di discesa.

```
c:\progr-1\mqab\arm1.asm
35 ;-----
36 CICLO1: clrwf PORTB ;ciclo principale del programma
37 bcfss PORTA, 1 ;Si attende il segnale su RA1
38 goto CICLO1
39 movwf IMPULSO_ALZARE
40 movwf TEMP2
41 ALZARE_BRACCIO: clrwf TMR0L
42 clrwf TMR1H
43 bcf PIR1, INT1IF
44 ATTESA_ALZARE_BRACCIO: call BRACCIO_AVANTI ;Si alza il braccio del robot
45 btfss PIR1, INT1IF
46 goto ATTESA_ALZARE_BRACCIO
47 decfsz TEMP2, 1
48 goto ALZARE_BRACCIO
49
50 CICLO2: clrwf PORTB ;Si attende il secondo segnale su RA1
51 btfss PORTA, 1
52 goto CICLO2
53 movwf IMPULSO_ABBASSARE
54 movwf TEMP2
55
56 ABBASSARE_BRACCIO: clrwf TMR0L
57 clrwf TMR1H
58 bcf PIR1, INT1IF
59 ATTESA_ABBASSARE_BRACCIO: call BRACCIO_INDIETRO ;Si abbassa il braccio del robot
60 btfss PIR1, INT1IF
61 goto ATTESA_ABBASSARE_BRACCIO
62 decfsz TEMP2, 1
63 goto ABBASSARE_BRACCIO
64
65 goto CICLO1
```

Questo è il ciclo principale del programma. All'inizio si attende l'attivazione del finecorsa, e quando questo succede si richiama la funzione che ha il compito di attivare il motore del braccio in senso ascendente. Il motore rimarrà attivato tanti secondi quanti indicati nella variabile IMPULSO_ALZARE all'inizio del programma. Dopo aver completato il primo ciclo, si attende una seconda attivazione del finecorsa nel cui caso si realizza la discesa del braccio seguendo lo stesso procedimento utilizzato nella salita. In questo caso il motore rimarrà acceso tanti secondi quanti indicati nella variabile IMPULSO_ABBASSARE.

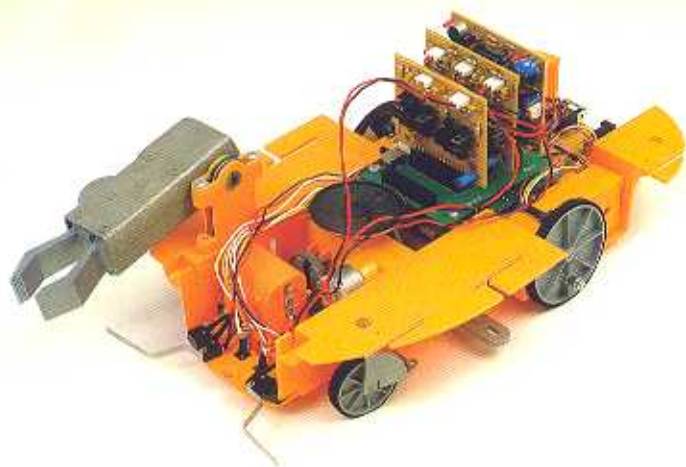
Esercizi con braccio e pinza (I)

```
67:.....
68:;funzione dedicata al movimento di salita del braccio
69 BRACCIO_ALZARE:
70      btfsz   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      btfsz  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      btfsz   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      btfsz  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)

```
c:\progra-1\mplab\am2.asm
1 ;-----
2 ;Programma di controllo della pinza di Pathfinder in modo temporizzato
3
4         LIST    p-16F870          ;Tipo di processore
5         include "P16F870.INC"    ;Definizione dei registri interni
6
7         ORG    0x00
8
9 ;-----
10 ;inizio del programma. Configurazioni
11         bsf    STATUS,RP0        ;Selezione il banco 1
12         movlw 0x07
13         movwf ADCON0            ;Porta A digitale
14         movlw b'00000111'
15         movwf PORTA            ;Configurazione della Porta A
16         bcf    STATUS,RP0        ;Selezione il banco 0
17
```

Continuiamo gli esercizi con il braccio di Pathfinder. In questo programma controlleremo la pinza del braccio in modo temporizzato. In questo modo di funzionamento, controlliamo la chiusura della pinza, dato che l'apertura della stessa si realizza automaticamente dopo un certo tempo. Il controllo della pinza è realizzato mediante il pin RA3 del microcontroller, quindi questo pin verrà configurato come uscita. Utilizzeremo il finecorsa collegato su JP7 della scheda di interfaccia (pin RA1 del microcontroller) per inviare il segnale che attiva la pinza.

```
c:\progra-1\mplab\am2.asm
17
18 ;-----
19 ;Ciclo principale del programma
20 WAIT_FINECORSO: bcf    PORTA, 3    ;Pinza disattivata
21                btrss   PORTA, 1
22                goto    WAIT_FINECORSO
23                bsf    PORTA, 3
24                nop
25                nop                ;Invio di un fronte di salita tramite RA3
26                nop
27                goto    WAIT_FINECORSO
28
29                END                ;Fine del programma sorgente
```

Nel modo temporizzato dobbiamo solo inviare un fronte di salita sul pin RA3 del microcontroller per fare in modo che la pinza si chiuda. Dopo un certo tempo, la pinza si aprirà. Il ciclo principale del programma rimane in attesa dell'attivazione del finecorsa. Quando questo avviene, invia un fronte su RA3 che provoca la chiusura della pinza. Il programma si ripete in modo infinito.

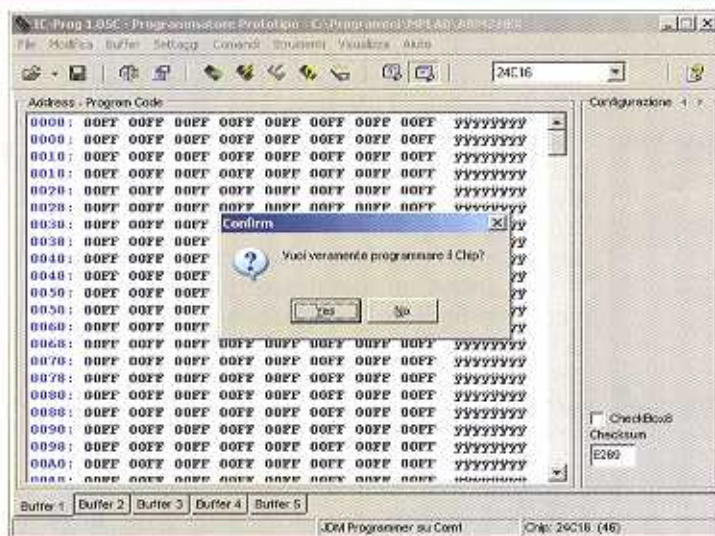
Description	On	Off	Data
Define	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Hex Format	<input checked="" type="checkbox"/> INHX8M	<input checked="" type="checkbox"/> INHX8S	INHX32
Error File	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ERR
List File	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Cross-reference File	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Warning level	<input checked="" type="checkbox"/> all	<input checked="" type="checkbox"/> warn+err	err
Case sensitivity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Macro expansion	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Default radix	<input checked="" type="checkbox"/> HEX	<input checked="" type="checkbox"/> DEC	OCT
Tab size	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Command Line
/s+ /p- /c- /p16F870

Additional Command Line Options

Scriveremo il programma mediante MPLAB. Dopo averlo scritto, procederemo alla compilazione. Il microcontroller selezionato sarà il PIC16F870, come in tutti gli esercizi con il robot. Dobbiamo compilare l'esercizio fino a quando otterremo il file con estensione .hex, cosa che succederà quando il programma sarà privo di errori.

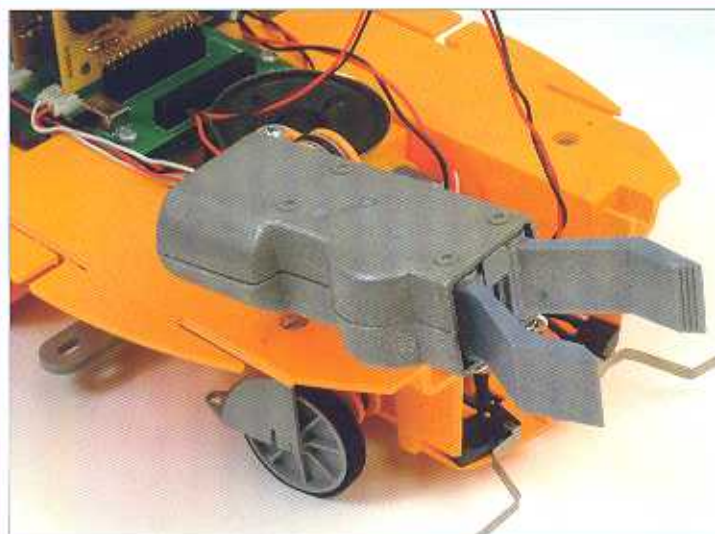
Esercizi con braccio e pinza (II)



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'
17:          movwf   PORTB            ;Configurazione della PortaB
18:          movlw   0x0F
19:          movwf   TRISA            ;Prescaler 256 per il timer
20:          bcf      STATUS,RPD        ;Seleziona il banco 0
21:          bcf      PORTB, 0          ;Pinza disattivata
22:
```

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.

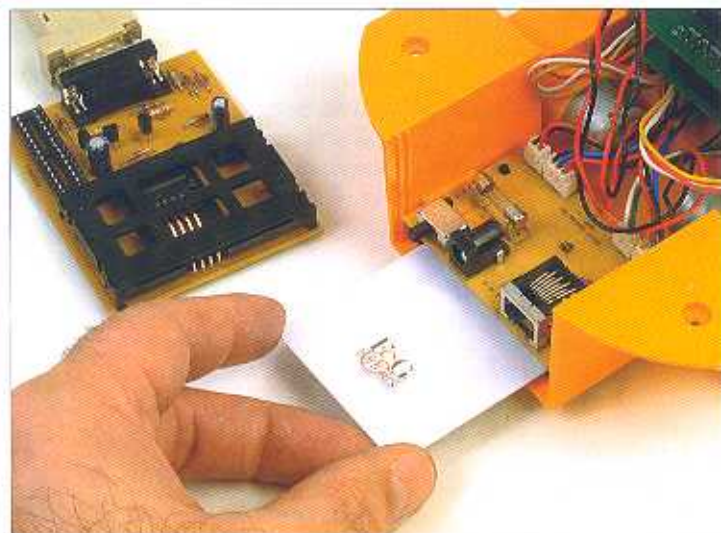
```
c:\progra-1\mplab\asm2.asm
22:.....
23:;Ciclo principale del programma
24:;Ciclo principale del programma
25:WAIT_FINECORSA: btfsz PORTA, 1
26:                goto  WAIT_FINECORSA
27:                bcf      PORTB, 0    ;Si invia un fronte che provocherà
28:                nap      PORTB, 0    ;l'apertura e la chiusura della pinza
29:                nap
30:                nap
31:                bcf      PORTB, 0
32:                call    DELAY        ;Si eliminano i rimbalzi del finecorsa
33:                goto    WAIT_FINECORSA
34:
```

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.

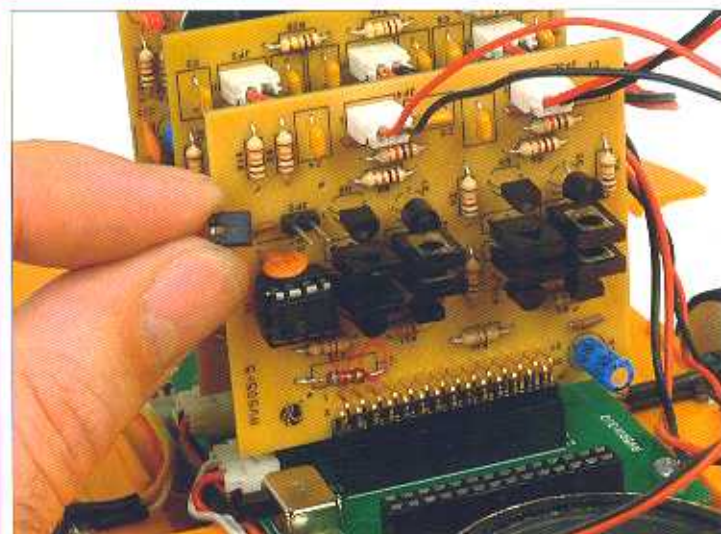
```
c:\progra-1\mplab\asm3.asm
35:.....
36:;Rutine di temporizzazione di ts
37:DELAY:        movlw   .100
38:              movwf  TEMPO
39:              movlw   0x01          ;temporizzazione da 100ms
40:              movwf  TRIS0
41:              bcf      INTCOM, 2
42:              decfsz  INTCOM, 2
43:              goto    DEL
44:              decfsz  TEMPO, 1
45:              goto    DEL_10        ;Si ripete il ciclo 100 volte
46:              return
47:
```

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.

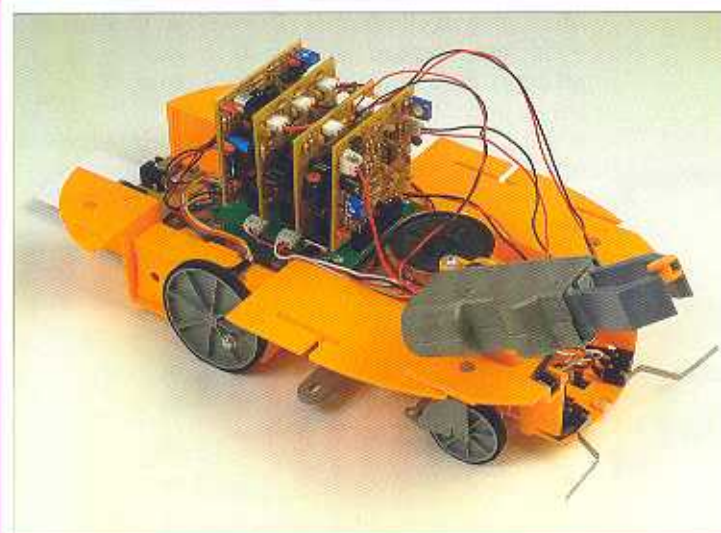
Esercizi con braccio e pinza (III)



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=16F478 ;Tipo di processore
6 include "16F478.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 0x0000
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 inizia bcf STATUS,RPB ;Selezione banca 1
19 movlw 0x07
20 movwf ACC0H
21 movlw 0'00000111'
22 movwf PORTA
23 clrf TRISB ;Porta B si configura come uscita
24 movlw 0'10000001' ;Prescaler per il Timer0
25 movwf TMR0 ;Selezione banca 0
26 bcf STATUS,RPB
27 movlw TOFF_BRACCIO
28 movwf TEMPO
29 clrf PORTB ;Si spengono i motori
30 movlw 0'00100001'
31 movwf TMR1H
32 clrf TMR1L
33 clrf TMR1H
34 bcf PIR1, INR1IF
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 CICLO1: clrf PORTB ;Ciclo principale del programma
39 btfsc PORTA, 2 ;SI testa il finecorsa RA2
40 call PINZA ;per l'apertura/chiusura della pinza
41 btfsc PORTA, 1 ;SI attende il segnale su RA1
42 goto CICLO1 ;per alzare il braccio
43 movlw IMPULSO_ALZARE
44 movwf TEMPO2
45 clrf TRISB
46 clrf PORTB
47 bcf PIR1, INR1IF
48 ATTESA_ALZARE_BRACCIO:
49 call BRACCIO_SALITA ;SI alza il braccio del robot
50 btfsc PIR1, INR1IF
51 goto ATTESA_ALZARE_BRACCIO
52 decfsz TEMPO2, 1
53 goto ALZARE_BRACCIO
54 CICLO2: clrf PORTB
55 btfsc PORTA, 2 ;SI testa il finecorsa RA2
56 call PINZA ;per l'apertura/chiusura della pinza
57 btfsc PORTA, 1 ;SI attende il secondo segnale su RA1
58 goto CICLO2 ;per abbassare il braccio
59 movlw IMPULSO_ABBASSARE
60 movwf TEMPO2
61 clrf TRISB
62 clrf PORTB
63 bcf PIR1, INR1IF
64 ATTESA_ABBASSARE_BRACCIO:
65 call BRACCIO_SALITA ;SI abbassa il braccio del robot
66 btfsc PIR1, INR1IF
67 goto ATTESA_ABBASSARE_BRACCIO
68 decfsz TEMPO2, 1
69 goto ABBASSARE_BRACCIO
70 goto CICLO1
```

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:\proga\lupia\lupia.asm
72:.....
73:;Restino 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:\proga\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:            movlw b'11000000'
92:            movwf 0
93:            movlw 0
94:            subwf 0, 0
95:            btfss STATUS, 2
96:            goto  SPEDIRE_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:SPEDIRE_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:\proga\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:           movlw b'11000000'
121:           movwf 0
122:           movlw 0
123:           subwf 0, 0
124:           btfss STATUS, 2
125:           goto  SPEDIRE_BRACCIO
126:           goto  ACCENDERE_BRACCIO
127:ACCENDERE_BRACCIO:
128:           bcf   PORTA, 7
129:           bcf   PORTA, 6
130:           movlw TON_BRACCIO
131:           movwf TEMPO
132:           return
133:SPEDIRE_BRACCIO:
134:           bcf   PORTA, 6
135:           bcf   PORTA, 7
136:           movlw TOFF_BRACCIO
137:           movwf TEMPO
138:           return
139:           END                    ;fine del programma sorgente
140:.....
```

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.