

## Esercizi con sensori a infrarossi (II)

```
c:\temp\1\1\1\1\1_1.asm
1 |IHRT.asm
2 |Programma che controlla l'accensione e lo spegnimento dei motori posteriori di Pathfinder
3 |in funzione dello stato del pin RA4, controllato dal sensore ad infrarossi della scheda
4 |audio
5
6
7           LIST    p-16F870           ;Tipo di processore
8           include "P16F870.INC"     ;Definizione dei registri interni
9
10          TEMPO EQU    0x20          ;Variabile ausiliaria
11          ANK   EQU    0x21
12
13          ORG    0x00
14          inizio          bsf    STATUS,RP0           ;Selezione il banco 1
15
16          movlw 0x07
17          movwf OPTION_REG           ;Si configura la portaA come digitale
18
19          movlw 0x0000
20          movwf OPTION_REG           ;Porta A come ingresso
21
22          movlw 0xFF
23          movwf PORTA                 ;Porta A si configura come uscita
24
25          movlw 0x00000001
26          movwf TRISB                 ;Porta B come uscita
27
28          movlw 0x0000
29          movwf TRISB                 ;Prescaler per il timer
30
31          bsf    STATUS,RP0           ;Selezione banco 0
32
33          bcf    PORTC,5               ;Si disattiva il sensore di voce
34
35          movlw 0
36          movwf TEMPO
37
38          END
```

Eseguiamo ora un esercizio per il controllo di Pathfinder tramite il segnale del sensore a infrarossi della scheda audio. Il circuito di condizionamento del sensore a infrarossi invia un '1' o uno '0' logico al pin RA4 del microcontroller. Quando su questo pin c'è un '1' logico si attiveranno i due motori posteriori di Pathfinder, invece, se c'è uno '0' logico si fermeranno.

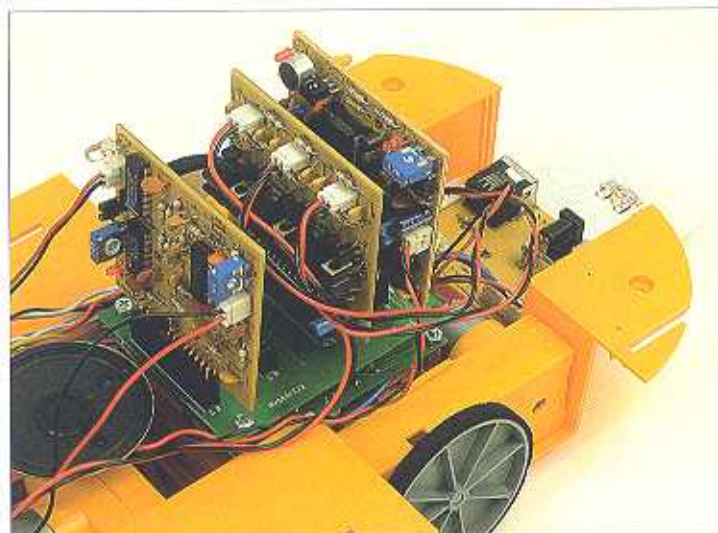
```
c:\temp\1\1\1\1\1_1.asm
39
40
41
42 |Ciclo principale del programma
43
44          LOOP:          btfss  PORTA,4
45
46          goto    PATHFINDER_ON
47
48          btfss  PORTA,4
49
50          goto    PATHFINDER_OFF
51
52          goto    LOOP
53
54
55
56
57 |Si spegno i motori
58
59          PATHFINDER_OFF:  bsf    PORTB,0
60
61          goto    LOOP
62
63
64
65
66 |Si accendono i motori e con il controllo della
67 |velocità di avanzamento
68
69          PATHFINDER_ON:  call   MOTORE_AVANTI
70
71          goto    LOOP
72
73          END
```

Dopo aver configurato la porta A come ingresso e la porta B come uscita per i motori, entriamo nel ciclo principale del programma. Consiste nel testare lo stato del segnale RA4. Se vale '1' saltiamo alla routine Pathfinder\_ON, che attiva i due motori posteriori. Per contro, se leggiamo lo stato '0', entriamo nella routine Pathfinder\_OFF che ferma i due motori posteriori. Il programma si ripete in modo infinito.

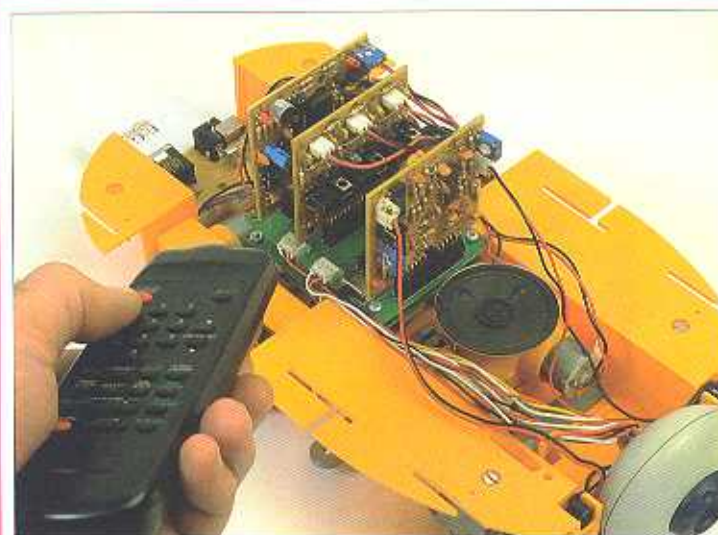
```
c:\temp\1\1\1\1\1_1.asm
74
75
76
77 |funzione che ha il compito di far avanzare i motori posteriori, tramite questa routine
78 |si controlla la velocità di avanzamento del robot mediante modulazione di ampiezza degli impulsi
79 |MOTORI_AVANTI:
80 |controllo della modulazione dell'ampiezza degli impulsi per i motori di avanzamento del robot
81
82          bsf    OPTION_REG,2          ;overclock del timer
83
84          return
85
86          bsf    OPTION_REG,2
87
88          return
89
90          MOTORE_TRAZIONE_ON:
91          decfsz TEMPO, 1
92
93          return
94
95          bsf    PORTB,0
96
97          movlw 0x00000001
98          movwf TRISB
99
100         movlw 0
101         movwf TRISB
102
103         bsf    PORTC,5
104
105         bsf    PORTC,5
106
107         goto    SPEGNERE_MOTORI
108
109         return
110
111         ACCENDERE_MOTORI:
112         bcf    PORTB,2
113
114         bcf    PORTB,3
115
116         bcf    PORTB,4
117
118         bcf    PORTB,5
119
120         movlw 0
121         movwf TEMPO
122
123         return
124
125         SPEGNERE_MOTORI:
126         bcf    PORTB,2
127
128         bcf    PORTB,3
129
130         bcf    PORTB,4
131
132         bcf    PORTB,5
133
134         movlw 0
135         movwf TEMPO
136
137         return
138
139         END                                     ;Fine del programma sorgente
```

L'attivazione dei motori posteriori del robot si realizza sotto forma controllata, con modulazione di ampiezza degli impulsi, il cui codice è riportato nell'immagine. Grazie a questa modulazione, la velocità dei motori si riduce. Utilizzeremo il programma MPLAB per scrivere l'esercizio. Una volta scritto sceglieremo l'opzione Build Node per ottenere il file .hex, che verrà generato se il programma non ha errori. In seguito mediante la scheda di scrittura e il programma ICPROG, caricheremo il programma sulla Smartcard. La Smartcard verrà inserita sulla scheda di alimentazione del robot, rispettando sempre il suo orientamento corretto.

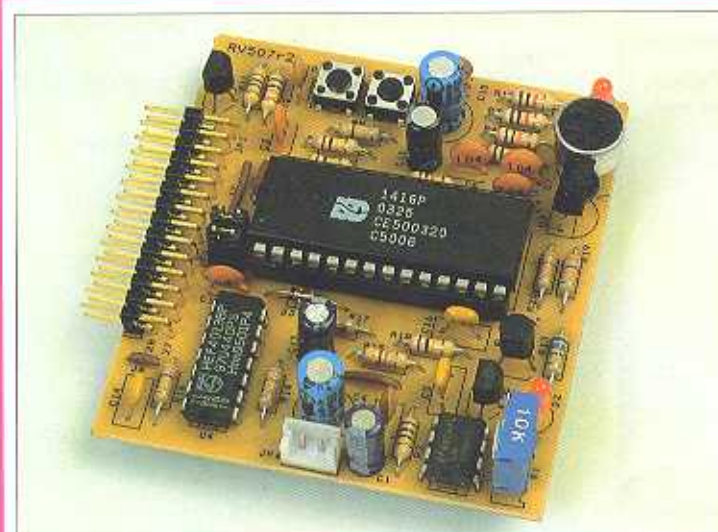
## Esercizi con sensori a infrarossi (II)



Questo esercizio gestisce i due motori posteriori e il sensore a infrarossi. Dobbiamo collegare la scheda di potenza con il connettore JP14 della scheda di interfaccia, così come i motori. La scheda audio rimarrà sul connettore JP13. L'orientamento della scheda deve essere quello corretto, così come mostrato nell'immagine. La scheda dei sensori può anche rimanere inserita nel suo posto, il connettore JP16.

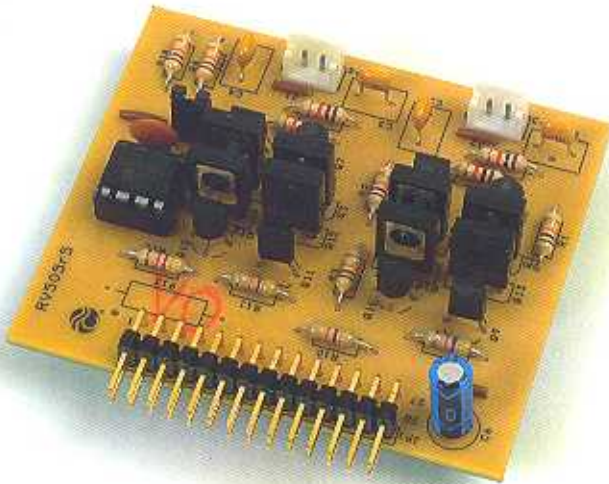


Per verificare l'esercizio alimenteremo il robot e utilizzeremo un normale telecomando. Il microcontroller impiegherà qualche secondo per leggere il contenuto della Smartcard. A ogni impulso del telecomando potremo veder cambiare lo stato del LED D2 della scheda audio. Quando il LED sarà acceso, i due motori posteriori del robot, si attiveranno, e quando il LED rimarrà spento, i motori si fermeranno.

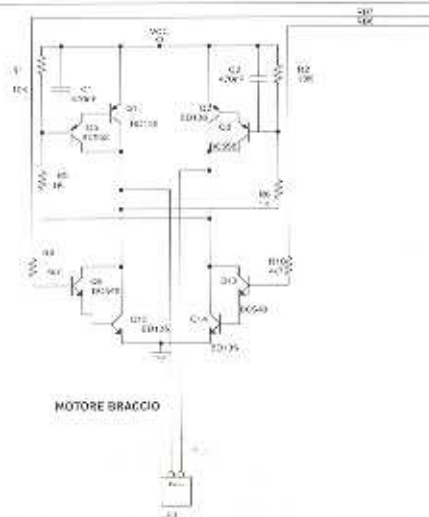


Con questo e alcuni esercizi dimostrativi del suo funzionamento, terminiamo l'analisi della scheda audio. È facile realizzare esercizi che combinino audio e infrarossi e includere queste funzioni nel controllo dei motori per il resto delle periferiche di Pathfinder. Mediante la scheda audio e il microfono che essa contiene acquisiremo il suono proveniente dall'ambiente, che sarà poi inviato verso il PC per poterlo udire negli altoparlanti del computer.

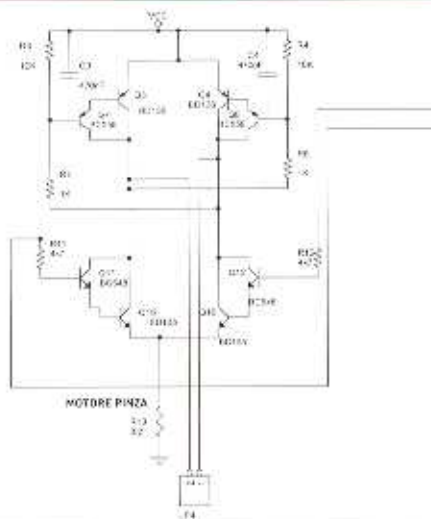
## Scheda di controllo del braccio (I)



Analizzeremo ora l'ultima scheda di Pathfinder, la scheda di controllo del braccio. Questa scheda gestirà il movimento di salita e discesa del braccio di Pathfinder e inoltre l'apertura e chiusura della sua pinza. Sarà in comunicazione con il microcontroller tramite la scheda di interfaccia. Possiede un microcontroller PIC12C508 che ha il compito di controllare i movimenti della pinza.

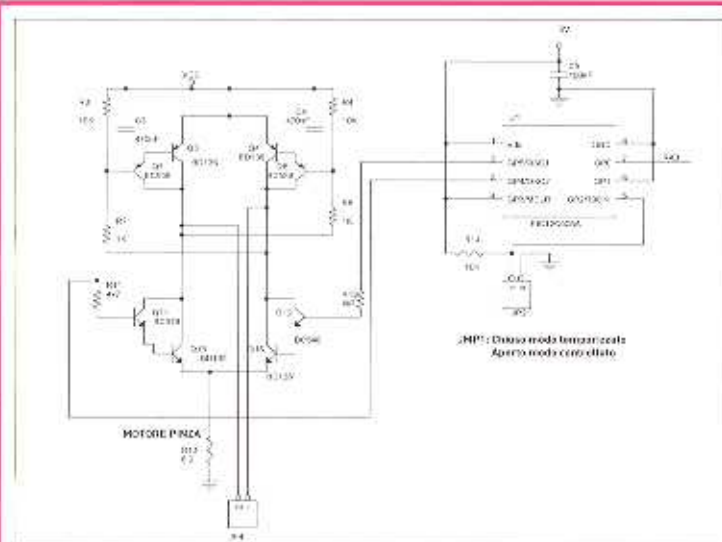


Nell'immagine possiamo vedere la prima parte dello schema della scheda di controllo del braccio. Si tratta di un ponte ad H, basato su transistor in configurazione Darlington, in modo simile ai ponti ad H della scheda di potenza che controlla i tre motori del sistema delle ruote e delle zampe di Pathfinder. Questo ponte ad H controlla il motore che gestisce il movimento di salita e discesa del braccio. Le linee RB6 e RB7 del microcontroller avranno il compito di controllare questo motore.



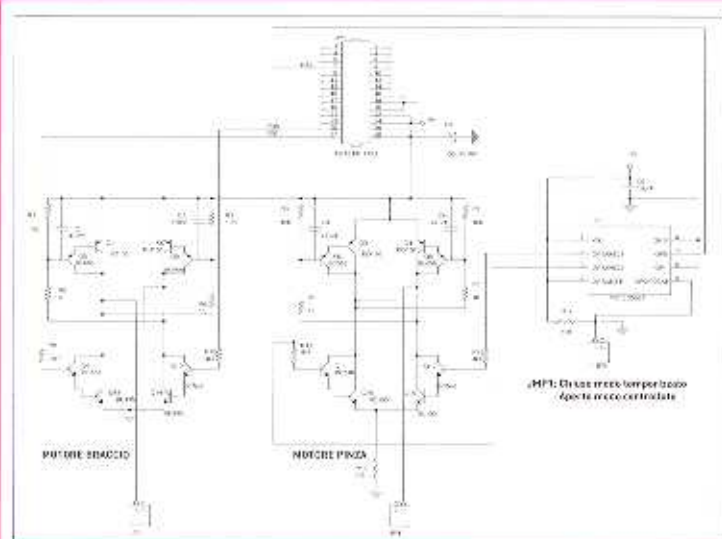
L'ultimo motore di cui è composto Pathfinder è il motore di controllo del movimento di apertura e chiusura della pinza. Per questo lavoro, è necessario un nuovo ponte ad H sulla scheda di controllo del braccio. Tuttavia, questo ponte non sarà gestito dal microcontroller principale, poiché i segnali di controllo che vanno alla base dei transistor arrivano da un microcontroller specifico modello PIC12C508.

## Scheda di controllo del braccio (I)

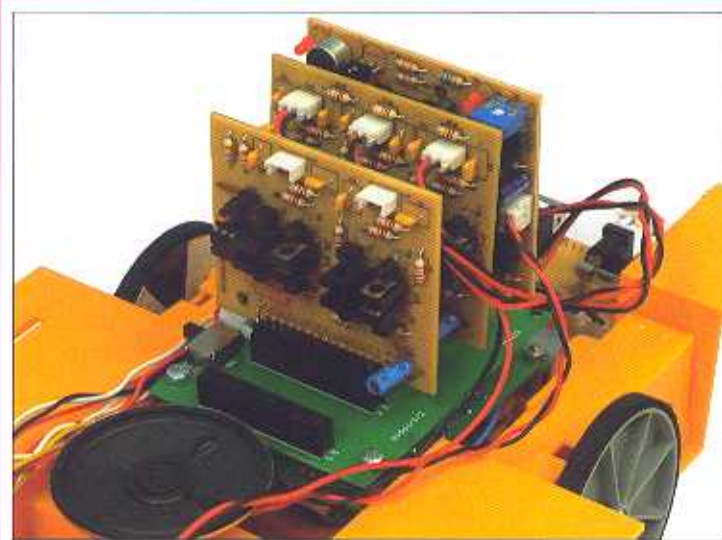


In quest'altro schema possiamo vedere il ponte ad H di controllo per il motore della pinza collegato con il microcontroller dedicato PIC12C508.

Abbiamo un jumper JP2 mediante il quale possiamo selezionare i due modi di funzionamento per la pinza. Un modo sarà quello temporizzato, in cui la pinza eseguirà un movimento di chiusura e di apertura automatico. La seconda opzione ci permetterà di controllare, a nostra scelta, sia l'apertura che la chiusura della pinza con segnali che arrivano dal microcontroller principale della scheda di controllo.

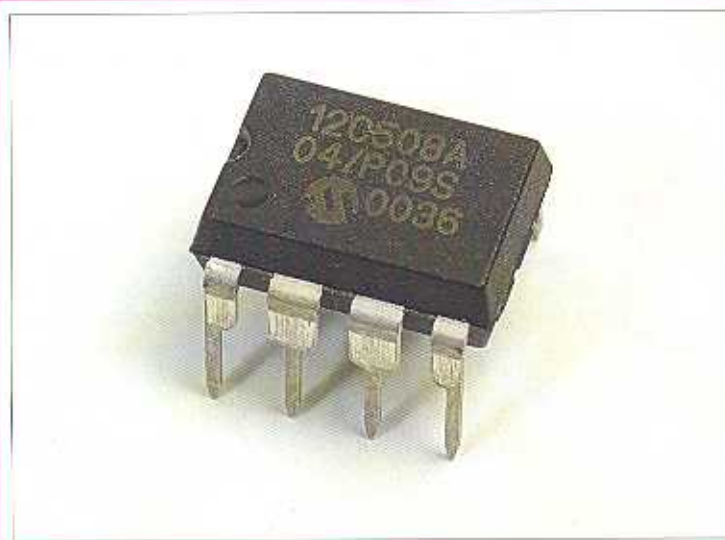


Questo è lo schema completo della scheda di controllo del braccio. Mediante i segnali RB6 e RB7 del microcontroller, gestiremo il movimento necessario per alzare e abbassare il braccio. Tramite il segnale RA3 invieremo ordini al microcontroller dedicato PIC12C508 che ha il compito di controllare l'apertura e la chiusura della pinza nei suoi due modi di funzionamento. Possiamo notare come i due motori del braccio prendano l'alimentazione direttamente dell'alimentatore, allo stesso modo dei motori delle ruote e delle zampe.

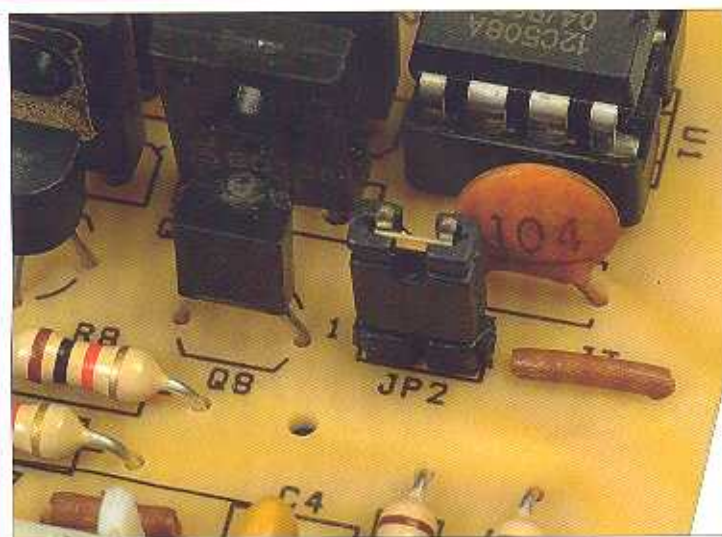


La scheda di controllo del braccio verrà inserita sul connettore femmina JP15 della scheda di interfaccia di Pathfinder, con lo stesso orientamento mostrato dall'immagine. Per essere sicuri di introdurre la scheda sempre con l'orientamento corretto e per fare in modo che non sia possibile inserire in questo connettore un'altra scheda, diversa da quella del braccio, possiamo tagliare il pin 4 del connettore JP1 della scheda del braccio, e inserirlo nel foro 4 del connettore femmina JP15 della scheda di interfaccia.

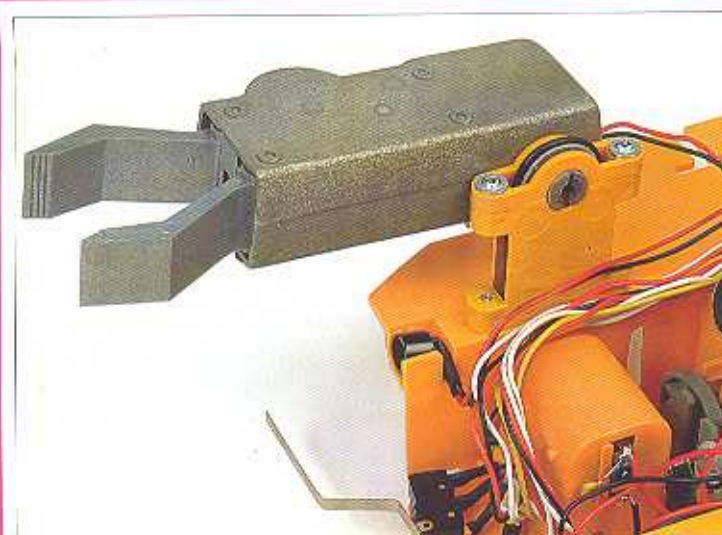
## Scheda di controllo del braccio (II)



Continuiamo l'analisi della scheda di controllo del braccio. In questa immagine possiamo vedere il microcontroller PIC12C508. Si tratta di un microcontroller OTP, con un programma già scritto che non si può modificare. Questo microcontroller ha una particolarità, non richiede oscillatore esterno, dato che possiede un clock interno da 4 MHz, che è la frequenza di lavoro del microcontroller PIC16F870 della scheda di controllo. Grazie ad esso, dei suoi otto piedini, due sono di alimentazione, i rimanenti si possono utilizzare come pin di ingresso e segnali di uscita.

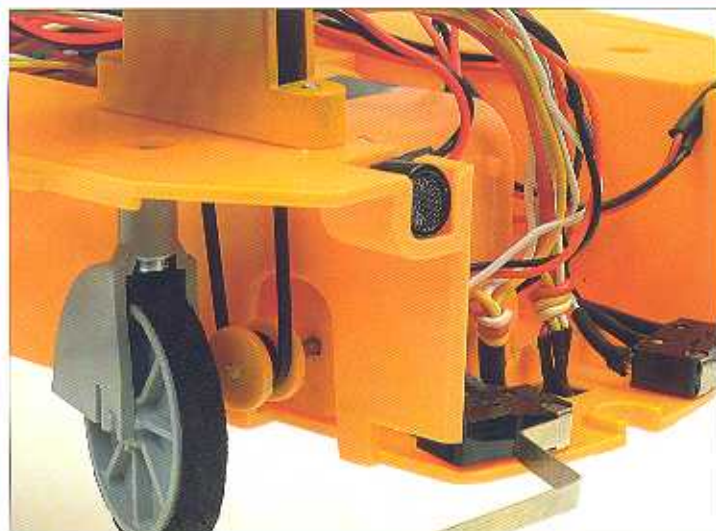


Il jumper JP2 della scheda di controllo del braccio, servirà per gestire due modi di funzionamento. Se il jumper è inserito, il funzionamento della pinza è temporizzato. Un fronte di salita, inviato tramite il pin RA3 del microcontroller principale, farà in modo che la pinza si chiuda e trascorso un determinato tempo si riapra. Se il jumper è tolto controlleremo il movimento. Mediante un fronte di salita inviato tramite RA3, faremo chiudere la pinza, e con un secondo fronte di salita la pinza si riaprirà. Tutti i movimenti di controllo del motore sono gestiti dal microcontroller PIC12C508.



In questa immagine possiamo vedere il braccio e la pinza di Pathfinder montati sul telaio. Il braccio avrà un movimento di salita e discesa fra due posizioni fisse. All'interno del braccio si trova il motore di controllo della pinza e tutto il meccanismo necessario per convertire i movimenti di rotazione del motore in movimento di apertura e chiusura della pinza.

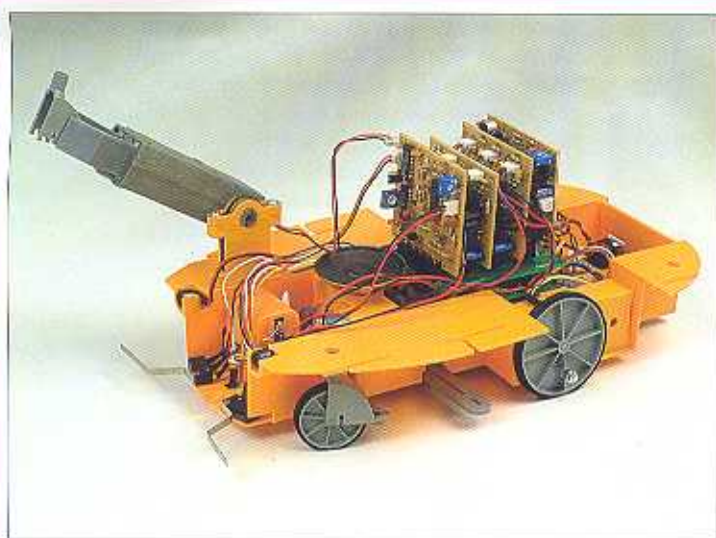
## Scheda di controllo del braccio (II)



Il braccio dispone di un motore esterno che servirà per gestire il suo movimento. Tramite un sistema di ingranaggi, riusciremo ad applicare una riduzione al movimento di rotazione del motore e aumentare la sua coppia, in modo da poter muovere il braccio. Sarà necessario un impulso sul motore per sollevarlo e un altro impulso per farlo scendere. Il braccio avrà due posizioni fisse, una con il braccio alzato e l'altra con il braccio in linea orizzontale.



Questo è il meccanismo di controllo della pinza. Come potete osservare, il movimento del motore è collegato a una serie di corone sino ad arrivare a una cremagliera, che avrà il compito di trasmettere il movimento alla pinza. I movimenti di temporizzazione necessari per l'attivazione di questo meccanismo, li controlla il microcontroller PIC12C508, quindi dovremo solamente inviare gli impulsi tramite il piedino RA3 del microcontroller della scheda di controllo per iniziare il movimento.



La pinza permetterà a Pathfinder di raccogliere gli oggetti. Grazie ai movimenti del braccio potremo anche alzarli per trasportarli e depositarli in un altro luogo. Il braccio aumenta il numero di applicazioni che può realizzare Pathfinder, dato che potrà afferrare e trasportare oggetti. Dobbiamo ricordare che gli oggetti non potranno essere eccessivamente pesanti, perché in caso contrario, la pinza non riuscirà a sollevarli.

# 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 0x001
19 movwf 0x000
20 movwf 0x00f
21 movwf PORTA ;Porta A come ingresso
22 cld PORTB ;Porta B si configura come uscita
23 movwf 0'0000001'
24 movwf TMR0 ;Prescaler per il Timer0
25 bcf STATUS_0P0 ;Selezione banco 0
26 movwf TOFF_BRACCIO
27 movwf TEMP
28 cld PORTB ;si spengono i motori
29 movwf 0'0010001'
30 movwf TMR0
31 cld TMR0
32 cld TMR1
33 bcf PIR1, INT1F
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: cldf 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: cldf TMR0L
42 cldf TMR0H
43 bcf PIR1, INT1F
44 ATTESA_ALZARE_BRACCIO: call BRACCIO_AVANTI ;si alza il braccio del robot
45 btfss PIR1, INT1F
46 goto ATTESA_ALZARE_BRACCIO
47 decfsz TEMP2, 1
48 goto ALZARE_BRACCIO
49
50 CICLO2: cldf 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: cldf TMR0L
57 cldf TMR0H
58 bcf PIR1, INT1F
59 ATTESA_ABBASSARE_BRACCIO: call BRACCIO_INDIETRO ;si abbassa il braccio del robot
60 btfss PIR1, INT1F
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.

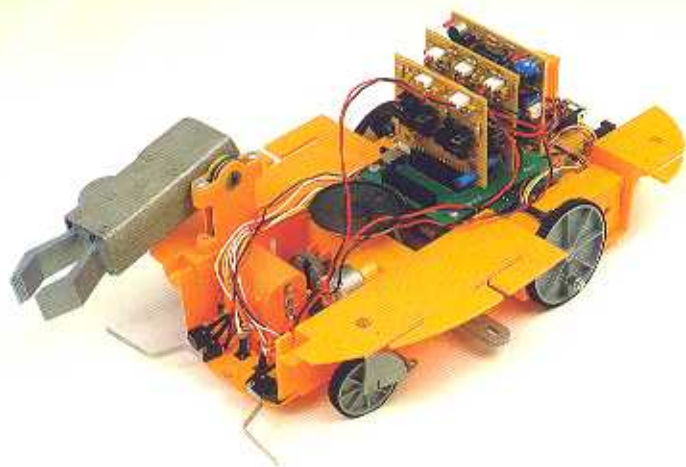
# Esercizi con braccio e pinza (I)

```
67:.....
68:;funzione dedicata al movimento di salita del braccio
69 BRACCIO_ALZARE:
70      btfss   INTCOH, 2      ;overflow del timer
71      return
72      bcf     INTCOH, 2
73      decfsz  TEMPO, 1
74      return
75      movwf  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
```

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      movwf  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      END      ;fine del programma sorgente
95
```

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.



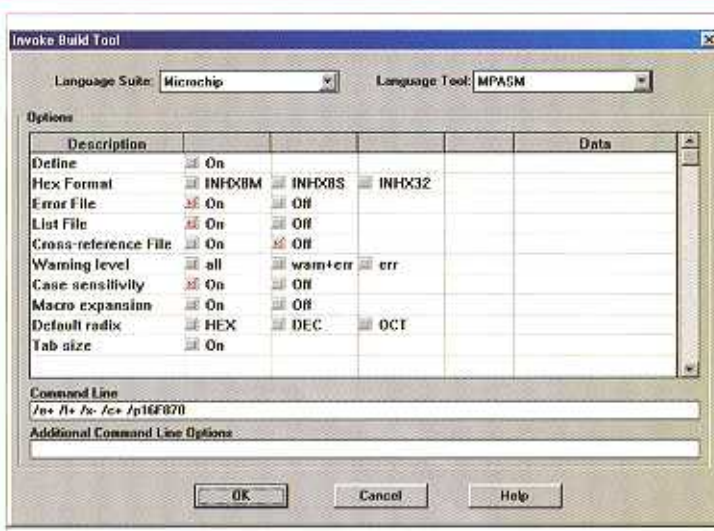
## Esercizi con braccio e pinza (II)

```
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 ;Seleziona 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 ;Seleziona 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.

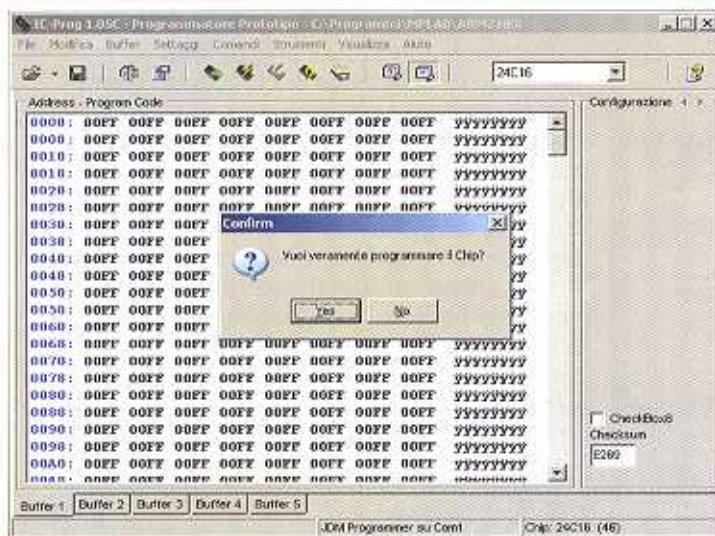
```
c:\progra-1\mplab\am2.asm
17
18 ;-----
19 ;Ciclo principale del programma
20 WAIT_FINECORSO: bcf PORTA, 3 ;Pinza disattivata
21 btfss PORTA, 1
22 goto WAIT_FINECORSO
23 bsf PORTA, 3
24 nop ;Invio di un fronte di salita tramite RA3
25 nop
26 goto WAIT_FINECORSO
27
28 END ;Fine del programma sorgente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

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.



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.

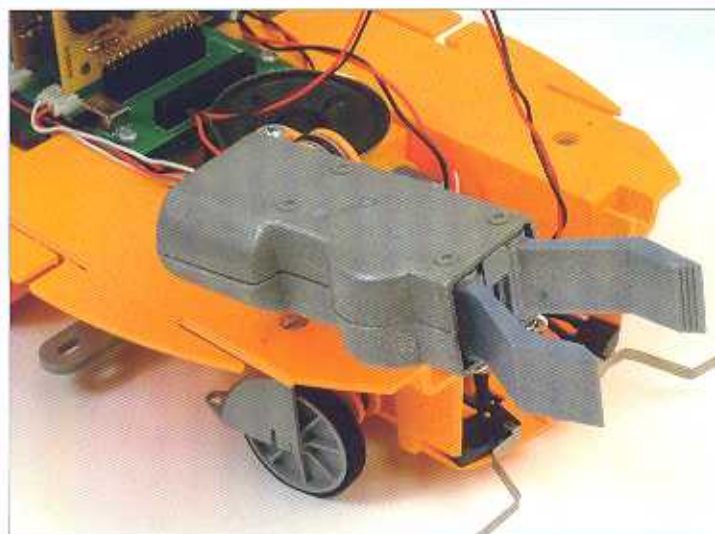
## Esercizi con braccio e pinza (II)



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.