

## Modo ruote: Linea nera (II)

```
c:\pathfinder\Modulo1\ruote\linea\1\Mod.asm
40 org 0x00
41
42 ***** INIZIO DEL PROGRAMMA PRINCIPALE *****
43
44 INIZIO:
45 bcf STATUS, 5 ;Banco 1
46 cld PORTB ;PortB come uscita per i sensori
47 movl b'00000111'
48 movl PORTC
49 movl PORTD ;Prescaler. Mediante esso si sceglie la frequenza
50 bcf STATUS, 5 ;Banco 0
51 bcf INTCON, 7 ;SI disabilitano gli interrupt
52 cld PORTB
53 movl 10000000
54 movl 10000000
55 movl 10000000
56 movl 10000000
57 movl 10000000
58 movl 10000000
59 movl 10000000
60 movl 10000000
61 movl 10000000
62 movl 10000000
63 movl 10000000
64 movl 10000000
65 movl 10000000
66 cld PORTB
67 cld PORTD
68 call DELAY
69
70 POSIZIONE MOTORE CENTRALE:
71 bcf PORTB, 2
72 goto ATTENDI_PISTA
73 call AVANZA_CENTRALE_DESTRA
74 goto POSIZIONE_MOTORE_CENTRALE
75 bcf PORTB, 0 ;Se muove un pù il motore centrale
76 bcf PORTD, 1
77 call DELAY
78
```

Questo primo programma di gestione di Pathfinder, consiste nel seguire un percorso nero su uno sfondo bianco o di colore chiaro. Possiamo costruire la pista utilizzando del nastro isolante. I due sensori ottici dovranno essere posizionati nella parte anteriore del telaio e orientati verso il suolo. Dobbiamo avere tutti i sensori ottici collegati alla scheda di controllo, come è stato spiegato nel fascicolo precedente. All'inizio del programma si realizzano le configurazioni dei pin di ingresso e di uscita e si inizializzano le diverse variabili. La prima operazione consiste nel posizionare le ruote anteriori diritte, operando sul motore centrale.

```
c:\pathfinder\Modulo1\ruote\linea\1\Mod.asm
79 *****
80 ;Ciclo principale del programma
81 ATTENDI_PISTA:
82 cld PORTB
83 btfsc PORTC, 0
84 goto ATTENDI_PISTA
85 btfsc PORTC, 1
86 goto CICLO
87 goto ATTENDI_PISTA
88 ;Sino a che il robot non è correttamente posizionato sulla pista non parte il
89 ;Questo succede quando i due sensori inviano un '1'
90 CICLO:
91 bcf PORTB, 0 ;Si ferma il motore centrale
92 bcf PORTB, 1
93 call MOTORE_TRAZIONE
94 movl PORTC, 0
95 andlw b'00000111'
96 xorlw b'00000111'
97 btfsc STATUS, 2
98 goto AVANZARE
99 movl PORTC, 0
100 andlw b'00000111'
101 xorlw b'00000111'
102 btfsc STATUS, 2
103 goto GIRARE_DESTRA
104 movl PORTC, 0
105 andlw b'00000111'
106 xorlw b'00000111'
107 btfsc STATUS, 2
108 goto GIRARE_SINISTRA
109 goto CICLO
110
```

Il robot non inizierà il suo funzionamento sino a quando non sarà posizionato sulla pista nera, per seguire il percorso il robot utilizzerà i sensori ottici. Nell'immagine possiamo vedere la routine principale del programma, chiamata CICLO, in cui si esegue continuamente il test dei sensori ottici per rilevare se il robot è sulla pista o se è uscito dalla traiettoria. In funzione di quale sensore esce dalla pista verranno eseguite le opportune correzioni con le ruote anteriori.

```
c:\pathfinder\Modulo1\ruote\linea\1\Mod.asm
110 *****
111 ;Se è uscito il sensore di sinistra è necessario girare verso destra
112 GIRARE_DESTRA:
113 decfsz AUXILIARIO_SINISTRA, 1
114 goto CICLO
115 btfsc SX, 0
116 goto CICLO
117 CENTRALE_DIANDO_DX:
118 call AVANZA_CENTRALE_DESTRA
119 btfsc PORTC, 2
120 goto CENTRALE_DIANDO_DX
121 SU_NERO_DX:
122 call AVANZA_CENTRALE_DESTRA
123 btfsc PORTC, 2
124 goto SU_NERO_DX
125 bcf PORTB, 0 ;Si ferma il motore centrale
126 bcf PORTD, 1
127 movl .1
128 movl SX
129 cld PORTB
130 goto CICLO
131
```

Questa routine serve per correggere la posizione del robot quando il sensore ottico sinistro esce dalla pista, muovendo le ruote anteriori verso destra. Si utilizza una variabile ausiliaria per filtrare il possibile rumore introdotto dai sensori. In questo modo solamente quando il segnale del sensore in questione viene rilevato dieci volte consecutive verrà considerata valida l'uscita del sensore stesso. Mediante la variabile SX si controlla anche se le ruote erano già girate in precedenza verso destra o se è necessario realizzare una rotazione completa.

## Modo ruote: Linea nera (II)



```
c:\pathfinder\Modulo1\Uscite\linea\1\Mod.asm
153 ;Questo è il movimento di correzione per impostare le ruote dritte
154 AVANZARE:
155     btfsc  SR, 0
156     goto  CORREGGI_SINISTRA
157     btfsc  DS, 0
158     goto  CORREGGI_DESTRA
159     goto  FINE_AVANZAMENTO
160 CORREGGE_DESTRA:
161     call  AVANZA_CENTRALE_DESTRA
162     btfss  PORTB, 2
163     goto  CORREGGE_DESTRA
164     bcf   PORTB, 0 ;Si ruota un pò il motore centrale
165     bcf   PORTB, 1
166     call  DELAY
167     bcf   PORTB, 0 ;Si ferma il motore centrale
168     bcf   PORTB, 1
169     goto  FINE_AVANZAMENTO
170 CORREGGE_SINISTRA:
171     call  AVANZA_CENTRALE_SINISTRA
172     btfss  PORTB, 2
173     goto  CORREGGE_SINISTRA
174     bcf   PORTB, 0 ;Si ruota un pò il motore centrale
175     bcf   PORTB, 1
176     call  DELAY
177     bcf   PORTB, 0 ;Si ferma il motore centrale
178     bcf   PORTB, 1
179 FINE_AVANZAMENTO:
180     clrf  SR
181     clrf  DS
182     movlw -10
183     movwf AUXILIARIO_DESTRA
184     movlw -10
185     movwf AUXILIARIO_SINISTRA
186     goto  CICLO
```

Questa funzione è importante, perché permette al robot di seguire correttamente una traiettoria. Quando un sensore esce, e realizziamo una rotazione per fare in modo che il robot torni all'interno della pista, è necessario che i due sensori tornino a essere posizionati sulla pista stessa e che vengano nuovamente raddrizzate le ruote. In caso contrario, il robot girerebbe sempre verso destra o verso sinistra. Questa routine controlla quale è stata l'ultima rotazione del robot, a destra o a sinistra, quindi quando ritorna sulla pista raddrizza le ruote, realizzando il movimento opportuno del motore centrale.

```
c:\pathfinder\Modulo1\Uscite\linea\3\Mod.asm
187 ;*****
188 ;Funzione che gestisce l'avanzamento dei motori posteriori. Tramite questa routine
189 ;si controlla la velocità di avanzamento del robot mediante modulazione di ang
190 ;*****
191 MOTORI_TRAZIONE:
192 ;Controllo della modulazione di ampiezza degli impulsi per i motori di avanzam
193     btfss  INTCON, 2 ;overflow del timer
194     return
195     bcf   INTCON, 2
196 MOTORI_TRAZIONE_AUX:
197     decfsz  TEMPO, 1
198     return
199     movwf  PORTB, 0
200     andlw  b'00111100'
201     movwf  AUX
202     movlw  0
203     subwf  AUX, 0
204     btfss  STATUS, 2
205     goto  SPEGNERE_MOTORI
206     goto  ACCENDERE_MOTORI
207 ;Mediante il valore caricato sulla variabile TEMPO nelle routine ACCENDERE_MOT
208 ;si può selezionare il Duty Cycle che si desidera per il segnale, la cui frequ
209 ;mediante il prescaler.
210 ACCENDERE_MOTORI:
211     bcf   PORTB, 2
212     bsf   PORTB, 3
213     bsf   PORTB, 4
214     bcf   PORTB, 5
215     movlw  TON_BORGNA
216     movwf  TEMPO
217     return
218 SPEGNERE_MOTORI:
219     bcf   PORTB, 2
220     bcf   PORTB, 3
221     bcf   PORTB, 4
222     bcf   PORTB, 5
223     movlw  TOFF_BORGNA
224     movwf  TEMPO
225     return
```

I motori posteriori di Pathfinder hanno il compito di fornire la velocità di avanzamento del robot. È necessario che questa velocità sia controllata, altrimenti sarebbe troppo elevata e il robot non potrebbe seguire alcun circuito. La routine mostrata dall'immagine, ha il compito di alimentare i motori posteriori mediante un segnale modulato, che ci permetterà di controllare la velocità finale del robot.

```
c:\pathfinder\Modulo1\Uscite\linea\1\Mod.asm
227 ;*****
228 ;Funzione per l'avanzamento del motore centrale quando questo gira verso destra
229 AVANZA_CENTRALE_DESTRA:
230 ;Controllo della modulazione di ampiezza degli impulsi per i motori di avanzam
231     btfss  INTCON, 2 ;overflow del timer
232     return
233     bcf   INTCON, 2
234     call  MOTORI_TRAZIONE_AUX
235     decfsz  TEMPO2, 1
236     return
237     movwf  PORTB, 0
238     andlw  b'00000011'
239     movwf  AUX
240     movlw  0
241     subwf  AUX, 0
242     btfss  STATUS, 2
243     goto  SPEGNERE_MOTORE_CENTRALE_DX
244     goto  ACCENDERE_MOTORE_CENTRALE_DX
245 ;Mediante il valore caricato sulla variabile TEMPO2 nelle routine ACCENDERE_MOT
246 ;si può selezionare il Duty Cycle che si desidera per il segnale, la cui frequ
247 ;mediante il prescaler.
248 ACCENDERE_MOTORE_CENTRALE_DX:
249     bcf   PORTB, 0 ;Si muove un pò il motore centrale
250     bcf   PORTB, 1
251     movlw  TON_CENTRALE
252     movwf  TEMPO2
253     return
254 SPEGNERE_MOTORE_CENTRALE_DX:
255     clrf  PORTB
256     movlw  TOFF_CENTRALE
257     movwf  TEMPO2
258     return
```

Questa routine serve per muovere il motore centrale di Pathfinder verso destra. Esiste anche una routine simile che muove il motore centrale verso sinistra. Questi movimenti del motore centrale sono convertiti in movimenti di traslazione mediante il sistema di ingranaggi e leve collegato alle ruote anteriori. Questa routine ha anche il compito di mantenere la modulazione di ampiezza degli impulsi applicati ai motori posteriori, per fare in modo che il robot non si fermi quando le ruote anteriori cambiano posizione.



## Modo ruote: Linea nera (III)

```
c:\pathfinder\1\vedefro\1\ruote\linea\1\lin2.asm
82 :*****
83 :Ciclo principale del programma
84 ATTENDI_PISTA:
85   clrfr PORT0
86   btfss PORTC, 0
87   goto ATTENDI_PISTA
88   btfsc PORTC, 1
89   goto CICLO
90   goto ATTENDI_PISTA
91 :Sino a che il robot non è correttamente posizionato sulla pista non parte il pr
92 :Questo succede quando i due sensori isolano un '1'
93 CICLO:
94   bcf PORT0, 0 ;Si ferma il motore centrale
95   bcf PORT0, 1
96   btfsc PORTA, 1 ;Si testa lo stato del finecorsa
97   goto FINECORSO_ON
98   btfsc PORTA, 2
99   goto FINECORSO_ON
100  call MOTORI_TRAZIONE
101  movf PORTC, 0
102  andlw b'00000011'
103  xorlw b'00000011'
104  btfsc STATUS, 2
105  goto AVANZARE
106  movf PORTC, 0
107  andlw b'00000011'
108  xorlw b'00000011'
109  btfsc STATUS, 2
110  goto CIRARE_DESTRA
111  movf PORTC, 0
112  andlw b'00000011'
113  xorlw b'00000011'
114  btfsc STATUS, 2
115  goto CIRARE_SINISTRA
116  goto CICLO
```

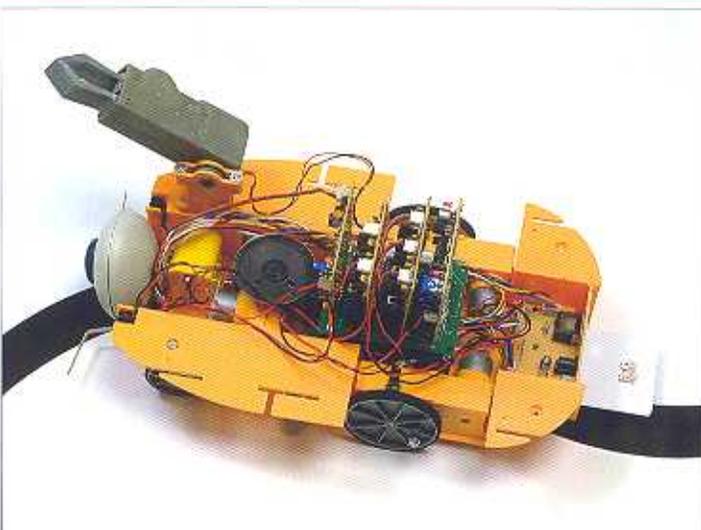
Il ciclo principale del programma segue gli stessi principi di lin1.asm. Si testano i segnali dei sensori ottici e in funzione del loro stato, si realizza la correzione corrispondente sui motori.

Nella routine è compreso il test dei nuovi segnali dei due pin RA1 e RA2, corrispondenti ai finecorsa. Se uno dei due segnali si attiva il robot si dovrà fermare, perché avrà toccato qualche ostacolo sul suo percorso.

```
c:\pathfinder\1\vedefro\1\ruote\linea\1\lin2.asm
186 FINE_AVANZAMENTO:
187   clrfr SX
188   clrfr DX
189   movlw .10
190   movwf AUXILIARIO_DESTRA
191   movlw .10
192   movwf AUXILIARIO_SINISTRA
193   goto CICLO
194
195 :*****
196 :A questa routine si arriva se qualche finecorsa si è attivato, il robot è entra
197 FINECORSO_ON:
198   clrfr PORT0 ;Si fermano i motori
199   goto CICLO
200
201 :*****
202 :Funzione che gestisce l'avanzamento dei motori posteriori. Tramite questa routi
203 :si controlla la velocità di avanzamento del robot mediante modulazione di arpie
204 MOTORI_TRAZIONE:
205 :Controllo della modulazione di ampiezza degli impulsi per i motori di avvanza
206   btfss INTCON, 2 ;Overflow del timer
207   return
208   bcf INTCON, 2
209 MOTORI_TRAZIONE_AUX:
210   decfsz TEMPO, 1
211   return
```

La funzione da eseguire, quando i finecorsa si attivano, si chiama FINECORSO\_ON e a essa si accede mediante l'attivazione di uno qualsiasi dei due sensori.

In questa routine fermeremo il funzionamento di tutti i motori del robot fino a quando il finecorsa è attivo. Quando il finecorsa cessa di segnalare la presenza di un ostacolo, il robot riprenderà la sua marcia sul circuito.



In questa immagine osserviamo Pathfinder su un percorso del tipo linea nera, che dobbiamo costruire per provare gli esercizi. Possiamo utilizzare del nastro isolante di colore nero su una superficie riflettente, idealmente di colore bianco. Programmeremo la Smartcard con i programmi lin1.hex e lin2.hex, la inseriremo nella scheda di alimentazione di Pathfinder e attenderemo qualche secondo per dare modo al robot di leggere il contenuto del programma e iniziare l'esecuzione.

# Modo ruote: Esploratore (I)

```
1 :=====
2 :Programma per il robot esploratore. Avanza e ogni volta che tocca un ostacolo (finecorsa)
3 :corregge la sua traiettoria.
4 :
5 :
6 :R00 e R01 sono per il motore centrale          MOTORI
7 :R00: 1
8 :R01: 0 -> Robot gira verso destra
9 :R02 e R03 sono per il motore della ruota sinistra
10 :R02: 0
11 :R03: 1 -> Robot avanza
12 :R04 e R05 sono per il motore della ruota destra
13 :R04: 1
14 :R05: 0 -> Robot avanza
15 :
16 :
17 :Quando i sensori rilevano nero inviano un '1'. Se rilevano bianco inviano una '0'.
18 :R09: Sensore di sinistra
19 :R01: Sensore di destra
20 :R02: Sensore centrale
21 :R01 e R02: Finecorsa
22 :R01: Finecorsa sinistro (JP7)
23 :R02: Finecorsa destro (JP8)
24 :=====
25 :
26 list p=16F870 ;definizione del tipo di dispositioin
27 include "16F870.inc" ;libreria con i registri di controllo
28
```

Eseguiamo una nuova esercitazione con Pathfinder nella configurazione con le ruote. Questi esercizi si trovano sul secondo CD, sotto la directory Ruote ed Esploratore. Abbiamo a disposizione tre esercizi: es1.asm, es2.asm ed es3.asm, in cui Pathfinder funzionerà come un esploratore, avanzerà e quando rileverà un ostacolo, grazie all'attivazione dei finecorsa anteriori, lavorerà per correggere la traiettoria ed evitarlo.

```
1 :=====
2 :Programma per il robot esploratore. Avanza e ogni volta che tocca un ostacolo (finecorsa)
3 :corregge la sua traiettoria.
4 :
5 :
6 :R00 e R01 sono per il motore centrale          MOTORI
7 :R00: 1
8 :R01: 0 -> Robot gira verso destra
9 :R02 e R03 sono per il motore della ruota sinistra
10 :R02: 0
11 :R03: 1 -> Robot avanza
12 :R04 e R05 sono per il motore della ruota destra
13 :R04: 1
14 :R05: 0 -> Robot avanza
15 :
16 :
17 :Quando i sensori rilevano nero inviano un '1'. Se rilevano bianco inviano una '0'.
18 :R09: Sensore di sinistra
19 :R01: Sensore di destra
20 :R02: Sensore centrale
21 :R01 e R02: Finecorsa
22 :R01: Finecorsa sinistro (JP7)
23 :R02: Finecorsa destro (JP8)
24 :=====
25 :
26 list p=16F870 ;definizione del tipo di dispositioin
27 include "16F870.inc" ;libreria con i registri di controllo
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 :===== INIZIO DEL PROGRAMMA PRINCIPALE =====
48 :=====
49 INIZIO:
50 bcf STATUS, 5 ;Banco 1
51 cllf PORTA ;Porta0 come uscita per i sensori
52 movlw 0x07
53 movwf ADCON0 ;Porta0 come ingresso digitale
54 movlw 0x0f
55 movwf PORTA
56 movlw b'00000111'
57 movwf PORTC
58 movlw b'00000001'
59 movwf TRIS ;Prescaler. Mediante il quale si sceglie la frequenza del
60 bcf STATUS, 5 ;Banco 0
61 bcf INTCON, 7 ;Si disabilitano gli interrupt
62 cllf PORTB
63 movlw TOFF_AVANZA
64 movwf TEMPO
65 movlw TOFF_CENTRALE
66 movwf TEMPO2
67 call DELAY
68
69 POSIZIONARE MOTORE CENTRALE:
70 btfsc PORTC, 2
71 goto CICLO
72 call AVANZA_CENTRALE_DESTRA
73 goto POSIZIONARE_MOTORE_CENTRALE
74 bcf PORTB, 0 ;Si muove un po' il motore centrale
75 bcf PORTB, 1
76 call DELAY
77
```

Per provare l'esercizio dobbiamo mantenere la stessa configurazione per i connettori dei motori e i sensori di quella realizzata con la linea nera. All'inizio del programma, si realizzano le configurazioni di base del microcontroller e si centra il motore centrale per fare in modo che le ruote anteriori siano orientate in avanti e il robot avanzi in linea retta.

```
1 :=====
2 :Programma per il robot esploratore. Avanza e ogni volta che tocca un ostacolo (finecorsa)
3 :corregge la sua traiettoria.
4 :
5 :
6 :R00 e R01 sono per il motore centrale          MOTORI
7 :R00: 1
8 :R01: 0 -> Robot gira verso destra
9 :R02 e R03 sono per il motore della ruota sinistra
10 :R02: 0
11 :R03: 1 -> Robot avanza
12 :R04 e R05 sono per il motore della ruota destra
13 :R04: 1
14 :R05: 0 -> Robot avanza
15 :
16 :
17 :Quando i sensori rilevano nero inviano un '1'. Se rilevano bianco inviano una '0'.
18 :R09: Sensore di sinistra
19 :R01: Sensore di destra
20 :R02: Sensore centrale
21 :R01 e R02: Finecorsa
22 :R01: Finecorsa sinistro (JP7)
23 :R02: Finecorsa destro (JP8)
24 :=====
25 :
26 list p=16F870 ;definizione del tipo di dispositioin
27 include "16F870.inc" ;libreria con i registri di controllo
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 :===== INIZIO DEL PROGRAMMA PRINCIPALE =====
48 :=====
49 INIZIO:
50 bcf STATUS, 5 ;Banco 1
51 cllf PORTA ;Porta0 come uscita per i sensori
52 movlw 0x07
53 movwf ADCON0 ;Porta0 come ingresso digitale
54 movlw 0x0f
55 movwf PORTA
56 movlw b'00000111'
57 movwf PORTC
58 movlw b'00000001'
59 movwf TRIS ;Prescaler. Mediante il quale si sceglie la frequenza del
60 bcf STATUS, 5 ;Banco 0
61 bcf INTCON, 7 ;Si disabilitano gli interrupt
62 cllf PORTB
63 movlw TOFF_AVANZA
64 movwf TEMPO
65 movlw TOFF_CENTRALE
66 movwf TEMPO2
67 call DELAY
68
69 POSIZIONARE MOTORE CENTRALE:
70 btfsc PORTC, 2
71 goto CICLO
72 call AVANZA_CENTRALE_DESTRA
73 goto POSIZIONARE_MOTORE_CENTRALE
74 bcf PORTB, 0 ;Si muove un po' il motore centrale
75 bcf PORTB, 1
76 call DELAY
77
78 :=====
79 :Ciclo principale del programma
80 CICLO:
81 bcf PORTB, 0 ;Si ferma il motore centrale
82 bcf PORTB, 1
83 call MOTORI_AVANTI
84 btfsc PORTA, 2
85 goto FINECORSA_DX_ON
86 btfsc PORTA, 1
87 goto FINECORSA_SX_ON
88 goto CICLO
89
90 FINECORSA_DX_ON:
91 call DELAY
92 btfss PORTA, 2
93 goto CICLO
94 call DELAY
95 btfss PORTA, 2
96 goto CICLO
97 goto SENSORE_DESTRA_ON
98
99 FINECORSA_SX_ON:
100 call DELAY
101 btfss PORTA, 1
102 goto CICLO
103 call DELAY
104 btfss PORTA, 1
105 goto CICLO
106 goto SENSORE_SINISTRA_ON
107
```

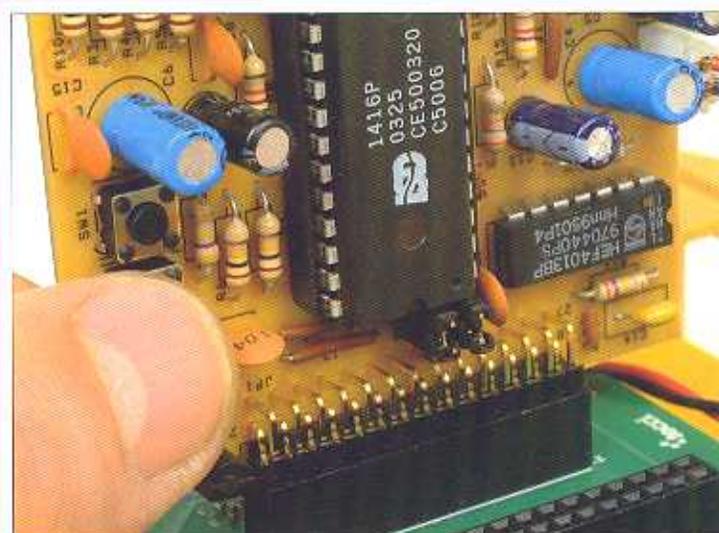
Questo è il ciclo principale del programma. L'avanzamento del robot si ottiene richiamando la funzione MOTORI\_AVANTI, che realizza una modulazione di ampiezza di impulsi applicata ai motori delle ruote superiori. In questo modo potremo controllare la velocità di avanzamento del robot. In questo ciclo principale si testano i finecorsa anteriori, e si rileva se è stato attivato il finecorsa destro o quello sinistro. Esistono due subroutine con ritardi intermedi per filtrare possibili rumori o falsi segnali dei sensori.



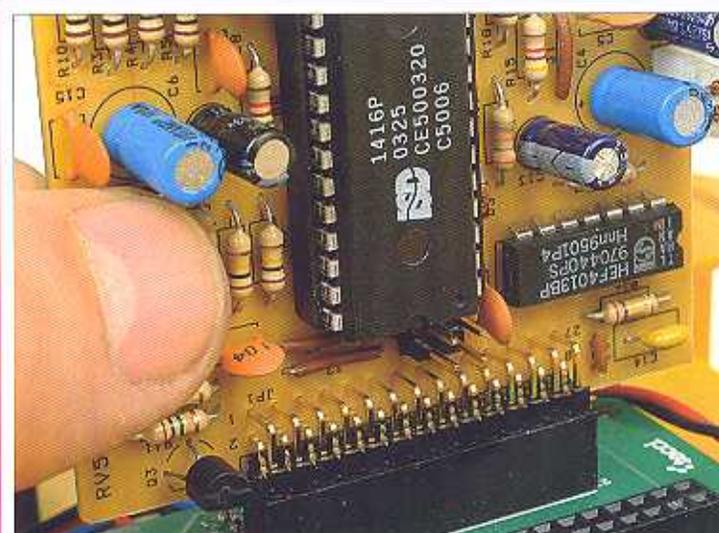
## Modo ruote: Esploratore (II)



Proveremo il secondo esercizio di esplorazione, che si trova nella cartella Esploratore del secondo CD, sotto il titolo esp2.asm. Questo programma realizza una funzione simile a quella di esp1.asm, permette cioè al robot di esplorare il terreno evitando gli ostacoli, aggiungendo però la riproduzione di messaggi tramite l'altoparlante, ogni volta che il robot entra in collisione con qualcosa. La prima operazione da eseguire per poter provare questo esercizio sarà caricare i messaggi sulla scheda audio.



Dobbiamo registrare due messaggi. Il primo di questi verrà riprodotto quando si attiva il finecorsa destro del robot, il secondo quando si attiva il finecorsa sinistro. Ogni messaggio avrà una durata di 4 secondi. Per memorizzare premeremo il pulsante SW2 e lo manterremo premuto mentre registriamo il messaggio. Il messaggio avrà una durata di 8 secondi, dei quali, i primi 4 saranno per il finecorsa destro e i rimanenti per il sinistro. I jumper JP2 e JP3 della scheda audio durante la registrazione del messaggio dovranno essere inseriti.



Dopo aver registrato il messaggio potremo utilizzare il pulsante SW1 della scheda audio per ascoltare ciò che abbiamo registrato. Se la registrazione effettuata è di nostro gradimento, toglieremo i jumper JP2 e JP3 dalle loro posizioni sulla scheda audio, lasciando il controllo della riproduzione dei messaggi al microcontrollore. A questo punto possiamo provare l'esercizio di esplorazione.

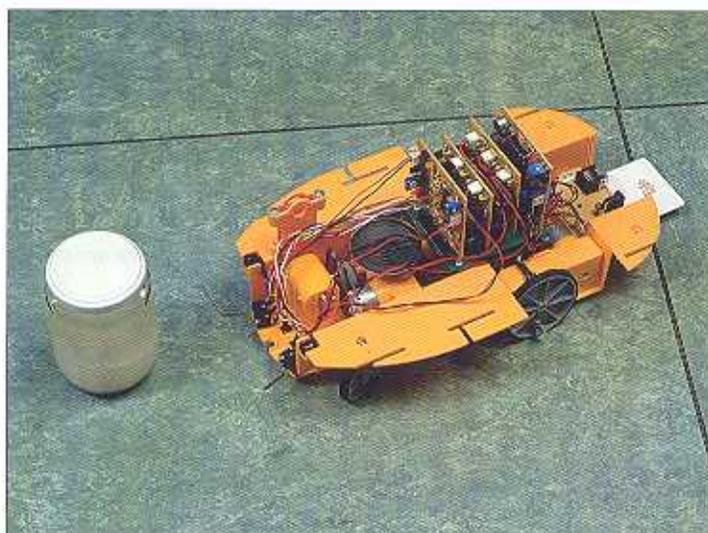
## Modo ruote: Esploratore (II)

```
c:\pathfinder\1\robot\esp2\1\esp2.asm
113 ;*****
114 ;Routine da eseguire quando si attiva il finecorsa di destra
115 SENSORE_DESTRA_ON:
116   bsf   PORTC, 6
117   bcf   PORTC, 7
118   bcf   PORTC, 5
119   clrF  PORTD ;Si fermano i motori e si attende che si riproduca il me
120   call  DELAY2
121   bsf   PORTC, 5
122   clrF  TEMPO1
123   clrF  TEMPO2
124   movl  .3
125   movw  TEMPO3
126 ATTENDI_RETROMARCIA_DESTRA:
127   call  MOTORI_RETROMARCIA
128   decfsz TEMPO1,1
129   goto  ATTENDI_RETROMARCIA_DESTRA
130   decfsz TEMPO2,1
131   goto  ATTENDI_RETROMARCIA_DESTRA
132   decfsz TEMPO3,1
133   goto  ATTENDI_RETROMARCIA_DESTRA
134 CENTRALE_SU_BIANCO_SX:
135   call  AVANZA_CENTRALE_SINISTRA
136   btfsc PORTC, 2
137   goto  CENTRALE_SU_BIANCO_SX
138 SU_NERO_SX:
139   call  AVANZA_CENTRALE_SINISTRA
140   btfsc PORTC, 2
141   goto  SU_NERO_SX
```

La struttura del programma esp2.asm è simile a quella di esp1.asm, spiegato nel capitolo precedente. La differenza consiste nell'inizio delle routine SENSORE\_DESTRA\_ON e SENSORE\_SINISTRA\_ON. All'inizio di ogni funzione si riproduce il messaggio corrispondente a ognuno dei due finecorsa e si attende qualche secondo per fare in modo che si possa ascoltare bene il messaggio, prima che il robot inizi nuovamente a muoversi.

```
c:\pathfinder\1\robot\esp2\1\esp2.asm
167 ;*****
168 ;Routine da eseguire quando si attiva il finecorsa di sinistra
169 SENSORE_SINISTRA_ON:
170   bsf   PORTC, 6
171   bcf   PORTC, 7
172   bcf   PORTC, 5
173   clrF  PORTD ;Si fermano i motori e si attende che si riproduca il me
174   call  DELAY2
175   bsf   PORTC, 5
176   clrF  TEMPO1
177   clrF  TEMPO2
178   movl  .3
179   movw  TEMPO3
180 ATTENDI_RETROMARCIA_SINISTRA:
181   call  MOTORI_RETROMARCIA
182   decfsz TEMPO1,1
183   goto  ATTENDI_RETROMARCIA_SINISTRA
184   decfsz TEMPO2,1
185   goto  ATTENDI_RETROMARCIA_SINISTRA
186   decfsz TEMPO3,1
187   goto  ATTENDI_RETROMARCIA_SINISTRA
188 CENTRALE_SU_BIANCO_DX:
189   call  AVANZA_CENTRALE_DESTRA
190   btfsc PORTC, 2
191   goto  CENTRALE_SU_BIANCO_DX
192 SU_NERO_DX:
193   call  AVANZA_CENTRALE_DESTRA
194   btfsc PORTC, 2
195   goto  SU_NERO_DX
```

In questa immagine vediamo le modifiche della routine SENSORE\_SINISTRA\_ON. I pin RC6 e RC7 servono per la selezione dei messaggi. In questo caso, RC6 vale '1', questo valore seleziona il secondo messaggio di 4 secondi registrato sulla scheda audio. Impostando a '0' il pin RC5 inizia la riproduzione del messaggio, che terminerà quando questo pin passa a valore '1'.



Possiamo eseguire delle prove con Pathfinder in diverse abitazioni o terreni per verificare come risolve le diverse situazioni e trova il percorso di uscita dopo aver schivato gli ostacoli. Oltre alla selezione dei messaggi che può riprodurre il robot, possiamo anche modificare la sua velocità di funzionamento mediante le variabili TON e TOFF dichiarate all'inizio del programma.

## Modo ruote: Esploratore (III)



Nella cartella esploratore, all'interno della directory ruote troviamo un terzo file chiamato esp3.asm che comprende un miglioramento rispetto i due programmi di tipo esploratore commentati in precedenza. Anche questo programma permette a Pathfinder di rilevare gli ostacoli con i finecorsa e di schivarli, riproducendo messaggi vocali ogni volta che entra in collisione con essi. Dispone inoltre del controllo di marcia e arresto del robot mediante un comando a distanza, che sfrutta il sensore a infrarossi di cui dispone il robot.

```
c:\pathfinder\luol\luole\esploratore\luole\esp3.asm
53 :*****
54 INIZIO:
55 bcf STATUS, 5 ;Banco 1
56 cird PORTB ;PortaB come uscita per i sensori
57 movlw 0x07
58 movwf ADCON0 ;PortaB come ingresso digitale
59 movlw 0x0f
60 movwf PORTA
61 movlw b'00000111'
62 movwf PORTC
63 movlw b'00000001'
64 movwf TRISB
65 bcf STATUS, 5 ;Prescaler, Mediante il quale si sceglie la freq
66 movlw 0'00100000'
67 movwf PCONC
68 bcf INTCON, 7 ;Si disabilitano gli interrupt
69 bcf PORTC, 5
70 cird PORTH
71 movlw TOFF_AVANZA
72 movwf TEMPO
73 movlw TOFF_CENTRALE
74 movwf TEMPO2
75 call DELAY
76
77 POSIZIONARE MOTORE CENTRALE:
78 btfsc PORTC, 2
79 goto CICLO
80 call AVANZA_CENTRALE_DESTRA
81 goto POSIZIONARE_MOTORE_CENTRALE
82 bcf PORTB, 0 ;Si muove un pò il motore centrale
83 bcf PORTB, 1
84 call DELAY
```

All'inizio del programma si realizzano le configurazioni. Il sensore a infrarossi è sul pin RA4 del microcontroller, quindi è necessario configurare questo pin come ingresso digitale. I motori sono sulla porta B, quindi questa porta è configurata come uscita. I pin RC5, RC6 e RC7 si configurano anch'essi come uscita per controllare il chip audio che ha il compito di riprodurre messaggi. La funzione POSIZIONA\_MOTORE\_CENTRALE, che si trova dopo la configurazione iniziale, ha il compito di raddrizzare le ruote del robot prima che questo inizi il suo avanzamento.

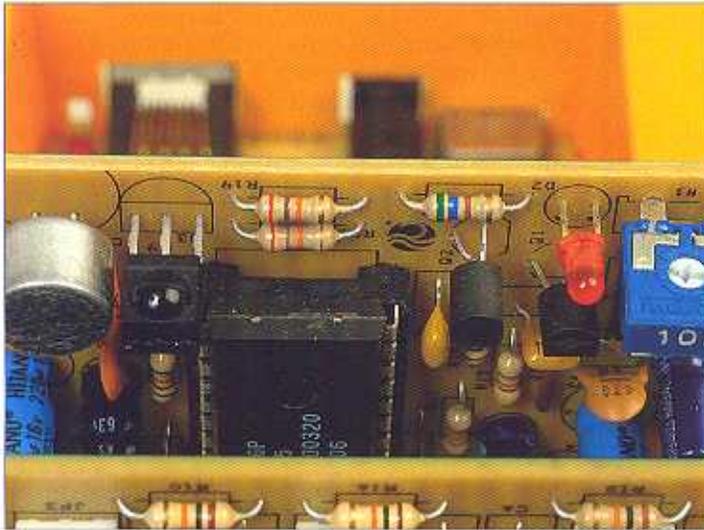
```
c:\pathfinder\luol\luole\esploratore\luole\esp3.asm
85
86 ;*****
87 ;Ciclo principale del programma
88 CICLO: bcf PORTB, 0 ;Si ferma il motore centrale
89 bcf PORTB, 1
90 btfss PORTA, 4
91 goto INFRAROSSO_ATTIVATO
92 call MOTORI_AVANTI
93 btfsc PORTA, 2
94 goto FINECORSA_DX_ON
95 btfsc PORTA, 1
96 goto FINECORSA_SX_ON
97 goto CICLO
98
```

Questo è il ciclo principale del programma, in cui possiamo vedere il test sui due finecorsa per rilevare gli ostacoli. È stato inserito il test del sensore a infrarossi, il cui cambio di stato si rileva mediante il pin RA4. Se il sensore si attiva passiamo alla routine INFRAROSSO\_ATTIVATO, in cui si fermeranno i motori del robot, e questi non inizierà il suo avanzamento fino a quando il sensore a infrarossi tornerà ad attivarsi.

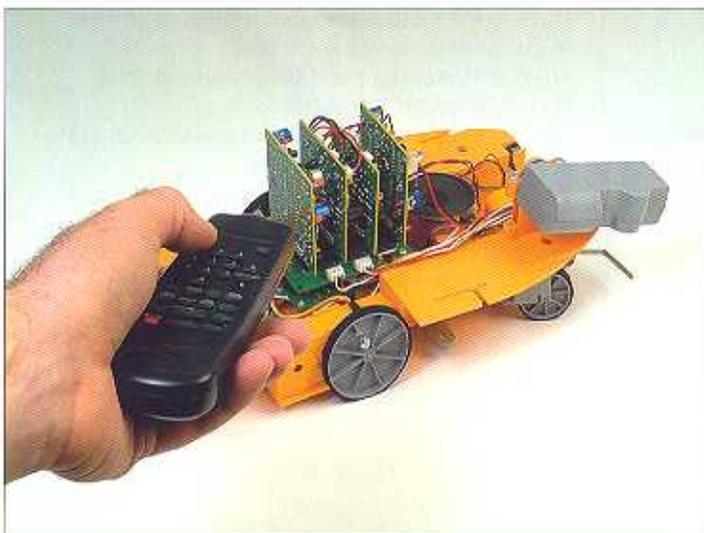
## Modo ruote: Esploratore (III)

```
c:\pathfinder\1\cdeko\1\ruote\explor\1\esp3.asm
29 .....
30 .....
31 list p=16F870 ;Definizione del tipo di dispositivo
32 include "P16F870.inc" ;Libreria con i registri di controllo
33 .....
34 ;Variabili di programma che saranno utilizzate come registri (memoria RAM de
35 TEMPO EQU 0x20 ;Variabile ausiliaria per la modulazione di ampiezza
36 AUX EQU 0x21 ;Variabile ausiliaria per operazioni matematiche
37 DELAY_VAR EQU 0x24
38 TEMPO2 EQU 0x25
39 TEMPO1 EQU 0x26
40 TEMPO_2 EQU 0x27
41 TEMPO_3 EQU 0x28
42 ;Definizioni di valori per alcuni parametri del programma
43 TON_AVANZA EQU .5
44 TOFF_AVANZA EQU .8
45 TON_CENTRALE EQU .5
46 TOFF_CENTRALE EQU .10
47 CORREGGI_CENTRALE EQU .10
48 .....
49 org 0x80
50 .....
```

Nei tre programmi di esplorazione abbiamo a disposizione alcune variabili all'inizio del programma, che serviranno per realizzare alcune configurazioni. Le variabili TON\_AVANZA e TOFF\_AVANZA servono per regolare la velocità dei motori posteriori e quindi la velocità di avanzamento del robot. Se aumentiamo il valore di TON in relazione a TOFF il robot andrà più veloce. Con le variabili TON\_CENTRALE e TOFF\_CENTRALE regoliamo i giri che eseguono le ruote anteriori per schivare gli ostacoli. Tanto maggiore sarà TON\_CENTRALE maggiore sarà la curva realizzata dal robot.



Il sensore a infrarossi si trova sulla scheda audio del robot, nella sua parte superiore. I sensori a infrarossi sono direzionali, quindi dovremo puntare verso di esso il telecomando per fare in modo che possa ricevere le onde a infrarossi e attivarsi. Il diodo LED D2 della scheda audio, visualizzerà lo stato del sensore a infrarossi, di conseguenza potremo verificare se lo stiamo attivando.



Per provare questa terza versione del programma esploratore in modo ruote, dobbiamo aver registrato il messaggio sulla scheda audio, e avere a disposizione un comune telecomando. Pathfinder rileverà gli ostacoli mediante i suoi fincorsa anteriori e riprodurrà messaggi ogni volta che entrerà in collisione, inoltre manovrerà per schivarli. Mediante il telecomando controlleremo l'avviamento e l'arresto del robot.