

Modo zampe: Esapodo (III)

```

c:\pathfinder\vedelo\1\zampe\esap3.asm
1      list p=16F07B
2
3      include "16F07B.inc"
4
5  CONTATORE      EQU    0x20
6  RD1            EQU    0x21
7  STATUS        EQU    0x25
8  I2CIP         EQU    0x30
9  ;La variabile retroscandica indica quanti passi indietro farà il robot prima di girare a destra
10 RETROSCANDICA EQU    5
11 ;Le variabili gira_sinistra e gira_destra indicano quanti passi esegue il robot per girare uno
12 20 verso destra dopo una collisione.
13 GIRA_SINISTRA EQU    4
14 GIRA_DESTRA  EQU    4
15
16 ;RD2 e RD3 -> ZAMP1 (JP4) (sinistra) con Sensore RD0 (JP12)
17 ;RD4 e RD5 -> ZAMP2 (JP2) (destra) con Sensore RD1 (JP11)
18 ;RD6 e RD1 -> ZAMP3 (JP3) (destra) con Sensore RD2 (JP13)
19 ;Quando i sensori rilevano nera inviamo un '1'. Se rilevano bianco inviamo un '0'
20 ;RD3: Finecorsa sinistra (JP7)
21 ;RD2: Finecorsa destra (JP8)
22
23 ;STATI IN CUI PASSA IL ROBOT NEL SUO MUOVIMENTO
24 ;STATUS 1
25 :      OFF_ZAMP1
26 :      OFF_ZAMP2
27 :      ON_ZAMP3
28 : Sensore RD2 su Bianco

```

Vediamo l'ultimo esercizio di esempio della gestione di Pathfinder in modo esapodo. Si chiama esap3.asm e si trova nella stessa directory Zampe del programma precedente. Con questo programma, Pathfinder funziona come un esploratore, avanzerà e ogni volta che toccherà un ostacolo con i finecorsa anteriori, retrocederà e girerà a destra o a sinistra per schivare l'ostacolo e proseguire il suo percorso.

```

c:\pathfinder\vedelo\1\zampe\esap3.asm
95
96 ;MACRO UTILIZZATE NEL PROGRAMMA
97 ;*****
98 SX_ZAMP1:      macro
99                bcf     PORTB, 0
100               bsf     PORTB, 1
101               ENDM
102 DX_ZAMP1:      macro
103                bcf     PORTB, 0
104               bsf     PORTB, 1
105               ENDM
106 OFF_ZAMP1:     macro
107                bcf     PORTB, 0
108               bsf     PORTB, 1
109               ENDM
110 ;*****
111 RX_ZAMP2:      macro
112                bcf     PORTB, 5
113               bsf     PORTB, 4
114               ENDM
115 DX_ZAMP2:      macro
116                bcf     PORTB, 5
117               bsf     PORTB, 4
118               ENDM
119 OFF_ZAMP2:     macro
120                bcf     PORTB, 5
121               bsf     PORTB, 4
122               ENDM
123 ;*****

```

La configurazione dei sensori ottici e dei finecorsa è la stessa di quella utilizzata nel precedente esercizio, esap2.asm; anche le configurazioni dei pin di ingresso e di uscita sono le stesse. Le macro che si trovano all'inizio del programma sono state generate per facilitare la gestione delle zampe del robot e sono identiche a quelle utilizzate nei due programmi precedenti.

```

c:\pathfinder\vedelo\1\zampe\esap3.asm
211
212 ;*****
213 ;Inizializzazione
214 CICLO:
215     movf     PORTA, W
216     movwf   TEMP
217     movf     TEMP, W
218     andlw   b'00000110'
219     xorlw   b'00000000'
220     btfsc   STATUS, Z
221     goto    AVANTI
222     movf     TEMP, W
223     andlw   b'00000110'
224     xorlw   b'00000010'
225     btfsc   STATUS, Z
226     goto    FINECORSO_SINISTRO
227     movf     TEMP, W
228     andlw   b'00000110'
229     xorlw   b'00000100'
230     btfsc   STATUS, Z
231     goto    FINECORSO_DESTRO
232

```

Questo è il ciclo principale del programma, la cui struttura è simile a quella degli altri esercizi che abbiamo già realizzato. Mediante una maschera si testano i finecorsa del robot e in funzione del loro stato si entra nelle diverse routines. Quando i due sensori sono disattivati il robot avanzerà, ma se qualcuno di essi si attiva, si passerà alla routine che farà retrocedere e girare a destra o a sinistra per schivare l'ostacolo.

Modo zampe: Esapodo (III)

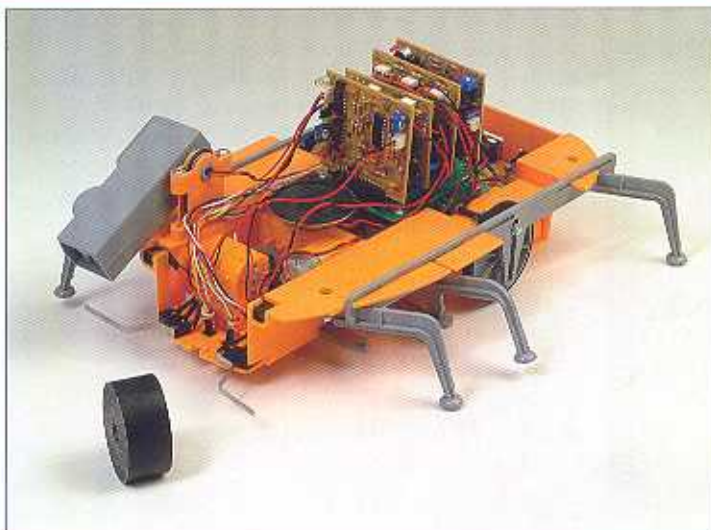


```
c:\pathfinder\1\modello\1\zampe\esap3.asm
232
233 FINECORSO_SINISTRO:
234     movlw RETROMARCIA
235     movwf AUX
236 DX_CICLO:
237     call INDIETRO
238     decfsw AUX, 1
239     goto DX_CICLO
240     movlw GIRA_SINISTRA
241     movwf AUX
242 DX_CICLO_SINISTRA:
243     call SINISTRA
244     decfsw AUX, 1
245     goto DX_CICLO_SINISTRA
246     goto CICLO
247
248 FINECORSO_DESTRO:
249     movlw RETROMARCIA
250     movwf AUX
251 SX_CICLO:
252     call INDIETRO
253     decfsw AUX, 1
254     goto SX_CICLO
255     movlw GIRA_DESTRA
256     movwf AUX
257 DX_CICLO_DESTRA:
258     call DESTRA
259     decfsw AUX, 1
260     goto DX_CICLO_DESTRA
261     goto CICLO
```

Queste sono le funzioni che realizza il robot quando si attiva il finecorsa sinistro o destro. Mediante la variabile RETRO_MARCIA, dichiarata all'inizio del programma, possiamo controllare quanti passi indietro vogliamo fare eseguire al robot prima di iniziare la rotazione a destra o a sinistra. Con le variabili GIRA_SINISTRA e GIRA_DESTRA controlliamo quanti passi esegue il robot quando gira in entrambe le direzioni. Questa funzione permetterà a Pathfinder di adattarsi a terreni diversi.

```
c:\pathfinder\1\modello\1\zampe\esap3.asm
263 .....
264 ;Routine da eseguire quando il robot cammina in avanti
265 AVANTI:
266     OFF_ZANPA1
267     OFF_ZANPA2
268     OFF_ZANPA3
269     movlw -1
270     movwf v_STATUS
271     call SCRIVERE
272 ;STATUS 1
273 AV_GIRO_DX:
274     OFF_ZANPA1
275     OFF_ZANPA2
276     DX_ZANPA3
277     btfsc PORTC,2
278     goto AV_GIRO_DX
279     call DELAY
280     btfsc PORTC,2
281     goto AV_GIRO_DX
282     movlw -2
283     movwf v_STATUS
284     call SCRIVERE
285 ;STATUS 2
286 AV_GIRO_DX2:
287     OFF_ZANPA1
288     OFF_ZANPA2
289     DX_ZANPA3
290     btfsc PORTC,2
291     goto AV_GIRO_DX2
292     call DELAY
293     btfsc PORTC,2
```

Le funzioni di avanzamento, retromarcia e rotazione del robot a sinistra o a destra sono le stesse che abbiamo già utilizzato nel programma esap2.asm, dato che le sequenze da realizzare con le zampe per ottenere questi movimenti nel robot sono sempre le stesse. Possiamo utilizzare le stesse funzioni per qualsiasi altra applicazione per la quale vorremo programmare il robot.



Per provare l'esercizio dobbiamo scrivere il programma esap3.hex sulla Smartcard e inserirla sulla scheda di alimentazione di Pathfinder. Dopo aver alimentato il robot, quest'ultimo inizierà la sua marcia e rileverà la presenza di ostacoli con i suoi finecorsa anteriori. Ogni volta che un finecorsa si attiva, il robot inizierà la sequenza per schivare l'ostacolo e continuare la sua marcia.