

Modo ruote: Traiettorie

```

c:\pathfinder\Modulo\Traiettorie\Traiettorie.asm
1:
2: ***** PROBLEMA CHE FA PERCORRERE A PATHFINDER UNA TRAIETTORIA PREDEFINITA. *****
3: Il robot avanzerà di un metro, girerà verso sinistra e percorrerà un
4: altro metro in questa direzione, dopo girerà verso destra, percorrerà un altro metro in questa
5: direzione ed infine si fermerà.
6: Il perimetro della ruota del robot è circa 10 cm; quindi,
7: se vogliamo che il robot percorra un metro, la ruota dovrà compiere circa 5
8: giri. Si utilizzano i sensori neri delle ruote come encoder, per rilevare
9: quanti giri compie la ruota.
10:
11: ***** MOTORI *****
12: RB0 e RB1 sono per il motore centrale
13: RB0: 1
14: RB1: 1 -> Robot gira verso destra
15: RB2 e RB3 sono per il motore della ruota sinistra
16: RB2: 1
17: RB3: 1 -> Robot avanza
18: RB4 e RB5 sono per il motore della ruota destra
19: RB4: 1
20: RB5: 0 -> Robot avanza
21:
22: ***** SENSORI *****
23: Quando i sensori rilevano nero inviano un '1'. Se rilevano bianco inviano uno '0'.
24: RB0 (JP12): Sensore encoder (Sensore nero di una ruota)
25: RB2 (JP18): Sensore centrale
26: RB4: Il orientamento sinistra-destra è riferito al centro vista della parte posteriore.
27:

```

Nella cartella Traiettorie, all'interno della directory Ruote, troviamo l'ultimo programma di esempio di Pathfinder in modo ruote. Questo programma mostra come possiamo programmare il robot per fargli realizzare traiettorie predefinite. Per realizzare programmi di traiettorie abbiamo bisogno di sapere quanta distanza ha percorso il robot. Utilizzeremo un sensore ottico CNY70 montato nella parte interna delle ruote. Questo sensore rileverà il passaggio della striscia nera presente sugli adesivi incollati all'interno delle ruote. Grazie a esso, controlleremo la rotazione delle ruote e di conseguenza l'avanzamento che realizza il robot.

```

c:\pathfinder\Modulo\Traiettorie\Traiettorie.asm
20: list p=16080 ;definizione del tipo di dispositivo
21: include "16080.inc" ;libreria con i registri di controllo
22:
23: ***** Variabili del programma che saranno utilizzate come registri (necessa 800 del PIC) *****
24: TEMPO EQU 0x20 ;variabile ausiliaria per la modulazione di ampiezza degli impulsi con il TMR0
25: SX EQU 0x21 ;variabile ausiliaria per operazioni matematiche
26: DX EQU 0x22 ;variabile che si imposta a 1 quando il robot muove verso sinistra, si è
27: DELAY_WD EQU 0x23 ;variabile che si imposta a 1 quando il robot muove verso destra, si è at
28: TEMPO EQU 0x24
29: CONTATORE_SINISTRA EQU 0x25
30: CONTATORE_AVANTI EQU 0x26
31: CONTATORE_DESTRA EQU 0x27
32: ***** Definizione di valori per alcuni parametri del programma *****
33: TEM_SINISTRA EQU 5
34: TEM_AVANZO EQU 10
35: TEM_DESTRA EQU 5
36: TEM_CENTRALE EQU 10
37: ***** Due variabili che servono per impostare il numero di giri che si vuole far compiere alla
38: ***** ruota, quando il robot si muove in avanti e verso sinistra, ogni giro della
39: ***** ruota equivale approssimativamente ad uno spostamento di 20 centimetri. *****
40: GIRI_SINISTRA EQU 5
41: GIRI_DESTRA EQU 5
42:
43: org 0x00

```

Per provare l'esercizio sono necessari solamente due sensori ottici. Il sensore ottico che rileva l'encoder delle ruote sarà collegato su JP12 della scheda di controllo, e il sensore ottico per il controllo di rotazione delle ruote anteriori rimarrà collegato su JP18, come per il resto degli esercizi che abbiamo realizzato. Questo programma realizza una traiettoria definita in cui si avanza di un metro in avanti e uno a sinistra, e infine un altro a destra. Nell'intestazione del programma disponiamo di alcune variabili che possiamo modificare per cambiare la distanza percorsa dal robot in ogni direzione.

```

c:\pathfinder\Modulo\Traiettorie\Traiettorie.asm
58: ***** INIZIO DEL PROGRAMMA PRINCIPALE *****
59: *****
60: INIZIO:
61: bsf STATUS, 5 ;banco 1
62: movlw 0x20
63: movwf DECBASE
64: movlw 0x21
65: movwf INCRBASE
66: clrf PRATE ;porta con uscita per i sensori
67: movlw b'00000111'
68: movwf PRATE
69: movlw b'00000001'
70: movwf TMR0 ;prescaler. Mediante esso si sceglie la frequenza del segnale di mo
71: bcf STATUS, 5 ;banco 1
72: bcf INTRCN, 7 ;51 disabilitano gli interrupt
73: clrf PRATE
74: movlw TOF_AVANZO
75: movwf TEMP0
76: movlw TOF_DESTRA
77: movwf TEMP1
78: movlw GIRI_SINISTRA
79: movwf CONTATORE_SINISTRA
80: movlw GIRI_DESTRA
81: movwf CONTATORE_DESTRA
82: movlw GIRI_DESTRA
83: movwf CONTATORE_DESTRA
84: call DELAY

```

All'inizio del programma si realizzano le configurazioni dei pin di ingresso e uscita. In questo caso dobbiamo leggere lo stato dei due sensori ottici e inviare il segnale di controllo ai motori collegati sulla porta B. Si gestisce anche il Timer 0 con il prescaler, dato che il controllo della velocità di avanzamento del robot si realizza mediante modulazione di ampiezza degli impulsi, con le stesse funzioni che sono già state utilizzate nel resto degli esercizi di controllo di Pathfinder in modo ruote.

Modo ruote: Traiettorie

```
c:\spahil\Ivodebo\Ivodebo\traiet\Traiet.asm
95 ;Ciclo principale del programma
96 CICLO1: bcf PORTB, 0 ;SI ferma il motore centrale
97         bcf PORTB, 1
98         call MOTORI_TRAZIONE ;il robot comincia a muoversi
99         btfscc PORTC, 0
100        goto CICLO1
101
102 CICLO2: bcf PORTB, 0 ;SI ferma il motore centrale
103         bcf PORTB, 1
104         call MOTORI_TRAZIONE ;il robot comincia a muoversi
105         btfscc PORTC, 0
106        goto CICLO2
107         decfsz CONTATORE_AVANTI,1
108        goto CICLO1
109
110 CENTRALE_SU_BIANCO_SX:
111         call AVANZA_CENTRALE_SINISTRA
112         btfscc PORTC, 2
113        goto CENTRALE_SU_BIANCO_SX
114
115 SU_NERO_SX:
116         call AVANZA_CENTRALE_SINISTRA
117         btfscc PORTC, 2
118        goto SU_NERO_SX
119
120 CICLO3: bcf PORTB, 0 ;SI ferma il motore centrale
121         bcf PORTB, 1
122         call MOTORI_TRAZIONE ;il robot comincia a muoversi
123         btfscc PORTC, 0
124        goto CICLO3
```

Qui comincia il ciclo principale del programma. In esso possiamo vedere che il robot inizia a muoversi con le ruote dritte fino a percorrere la distanza di un metro, che viene definita tramite la variabile `CONTATORE_AVANTI`. Dopo aver percorso un metro il robot girerà le ruote verso sinistra, e percorrerà un altro metro in questa direzione. In questo caso la distanza che percorre verso sinistra si controlla con il valore della variabile `CONTATORE_SINISTRA`.

```
c:\spahil\Ivodebo\Ivodebo\traiet\Traiet.asm
122 CICLO4: bcf PORTB, 0 ;SI ferma il motore centrale
123         bcf PORTB, 1
124         call MOTORI_TRAZIONE ;il robot comincia a muoversi
125         btfscc PORTC, 0
126        goto CICLO4
127         decfsz CONTATORE_SINISTRA,1
128        goto CICLO3
129
130 CENTRALE_SU_BIANCO_DX:
131         call AVANZA_CENTRALE_DESTRA
132         btfscc PORTC, 2
133        goto CENTRALE_SU_BIANCO_DX
134
135 SU_NERO_DX:
136         call AVANZA_CENTRALE_DESTRA
137         btfscc PORTC, 2
138        goto SU_NERO_DX
139
140 CICLO5: bcf PORTB, 0 ;SI ferma il motore centrale
141         bcf PORTB, 1
142         call MOTORI_TRAZIONE ;il robot comincia a muoversi
143         btfscc PORTC, 0
144        goto CICLO5
145
146 CICLO6: bcf PORTB, 0 ;SI ferma il motore centrale
147         bcf PORTB, 1
148         call MOTORI_TRAZIONE ;il robot comincia a muoversi
149         btfscc PORTC, 0
150        goto CICLO6
151         decfsz CONTATORE_DESTRA,1
152        goto CICLO5
153
154 FINE: goto FINE
```

Quando il robot avrà percorso un altro metro verso sinistra girerà completamente verso destra per terminare il programma, percorrendo un altro metro in questa direzione. In questo caso il controllo dell'avanzamento verso destra si realizza tramite la variabile `CONTATORE_DESTRA`. Il valore di queste tre variabili `CONTATORE_GIRI_AVANTI`, `CONTATORE_GIRI_SINISTRA` e `CONTATORE_GIRI_DESTRA`, che si trovano nell'intestazione del programma. Ogni unità di valore contenuta in una delle variabili equivale a un percorso di 20 cm nella direzione corrispondente.



Il file che contiene il codice sorgente si chiama `traiet.asm` e il programma esadecimale `traiet.hex`. Scriveremo questo file sulla scheda di memoria e la inseriremo sul robot per iniziare l'esecuzione del programma. I sensori ottici inseriti nel lato interno delle ruote ci forniranno il controllo sulla distanza che percorre il robot, quindi potremo realizzare applicazioni in cui il robot segue delle traiettorie definite e assolve compiti specifici.