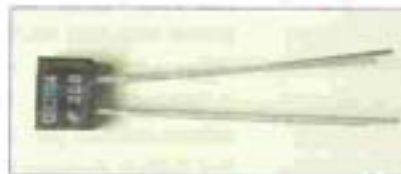




# Con cinque sensori

**C**on il precedente, il quinto sensore monoblocco, allegato a questo fascicolo, completa lo sviluppo del DeA Line Follower. Disponendo di una schiera di più sensori, è ora possibile ottenere un maggior numero di informazioni sul percorso che si vuole seguire e costruire così programmi che permettano al robot di adattarsi

meglio anche a tracciati più complessi. Per poter alloggiare correttamente i nuovi sensori, procedi come visto a pag. 174.



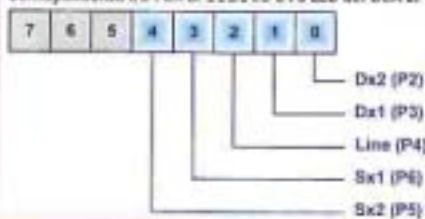
Nelle foto, il sensore (a sinistra) sarà alloggiato in una delle sedi laterali esterne del DeA Line Follower (sopra, indicate dalle frecce). Ripercorrendo le fasi di montaggio **C+D** di pag. 174, alloggerai correttamente entrambi i nuovi sensori.

## Le fasi di programmazione

**P**rima di utilizzare il DeA Line Follower (DeA LF) nella nuova configurazione, è necessario procedere nuovamente alla sua *calibrazione*, per tarare il circuito comparatore in base ai due nuovi sensori. A tal proposito, occorrerà riadattare il programma di test che abbiamo già visto a pag. 176, con le opportune modifiche per il funzionamento del DeA LF con cinque sensori. A destra è riproposta l'intera listato del programma *aggiornato*.

Vediamo ora quali sono le modifiche apportate (a destra) rispetto al programma originale. Innanzitutto, occorre inserire **due nuove costanti**, relative alle porte dei sensori aggiuntivi: **Sx2** e **Dx2**, infatti, fanno riferimento rispettivamente alle porte **P5** e **P2**. Inoltre, la variabile **lfbits**, deputata a contenere il risultato della lettura dei sensori, non può più essere dichiarata di tipo **nib**, bensì **byte**: per contenere le informazioni relative a cinque sensori, infatti, sono necessari **5 bit** (come visualizzato sotto, degli 8 totali 3 rimarranno inutilizzati). In fase di **inizializzazione**, poi, occorre impostare in **modalità output** anche le porte relative ai due nuovi sensori e settarle al valore logico **1** (per mantenere inizialmente spenti tutti i LED). Proseguendo nell'analisi delle modifiche apportate, puoi notare che, nella **subroutine di lettura dei sensori**, le operazioni di lettura e di memorizzazione saranno eseguite **5 volte**: la variabile contatore

Corrispondenza tra i bit di lfbits e i 5 LED del DeA LF



**cont** del ciclo **for**, infatti, varierà tra **0** e **4**; inoltre, nell'istruzione di **lookup**, che rimappa il valore di **cont** su **ledPos**, sono state aggiunte le due nuove costanti dichiarate, **Sx2** e **Dx2**.

```

(*L875k*)
(*OSTATE HDZ)
----- Dichiarazione Costanti -----
LED_ON      con    0
LED_OFF     con    1
LINEA_BIANCA con    0
LINEA_NERA  con    1
MODALITA    con    LINEA_NERA
Sx1         con    6
Sx2         con    5
LISE       con    4
Dx1        con    3
Dx2        con    2
SpostatiIn con    2
----- Dichiarazione Variabili -----
ledPos     var    nib
cont       var    nib
lfbits     var    byte
----- Inizializzazione -----
cont = %d111100
dir1 = %d111100
Debug "Test sensori DeA Line Follower". CP
debug CH
debug CR
debug " "
debug " 0 1 2 3 4 5 6 7 8"
debug "  Dx2 Dx1  Line  Sx1  Dx2  "
debug " "
debug " (+) "
debug " 0 "
Debug " "
----- Programma Principale -----
Main:
  %cmdb Lettura_Sensori
  %cto Main
  %end
----- Subroutine -----
Lettura_Sensori
  lfbits = 0
  for cont = 0 TO 4
    lookup cont, [Dx2 Dx1 LINE Sx1 Sx2] ledPos
    ctrl lowbit(ledPos) = LED_ON
    pause 1
    lfbits_lowbit(cont) = %d9 ~ MODALITA
    ctrl lowbit(ledPos) = LED_OFF
    debug SpostatiIn, [(4-cont)*5], 4, %IND lfbits_lowbit(cont)
  next
  %ctas
  
```

## LE FASI DI PROGRAMMAZIONE

Infine, per visualizzare correttamente le informazioni all'interno dello **schema grafico** riprodotto nella finestra di debug, è necessario modificare l'espressione parametrica per il calcolo della prima coordinata del cursore: al precedente valore [9] viene sostituito **4**, corrispondente alla colonna relativa alla prima posizione [1], ovvero **Sx2**; inoltre, il valore 2 della precedente espressione (2-cont) viene sostituito con il valore **4**. Ovviamente, analoghe modifiche dovranno essere apportate anche agli altri programmi relativi al DeA LF, cosicché siano aggiornati alla nuova configurazione. A questo punto, dunque, potrai procedere alla **calibrazione** vera e propria, ripetendo per tutti e cinque i sensori i passi descritti in dettaglio alle pagg. 176-177. Facciamo ora un passo avanti. Come si è accennato, due sensori in più, posti di lato esternamente, forniscono più informazioni sul tracciato da seguire. Tali informazioni possono essere utilizzate per migliorare la navigazione del robot sia in termini di precisione sia in termini di velocità di percorrenza. Ipotizzando per il momento di escludere i casi 'misti' (di cui si è detto a pag. 182), se la linea del percorso venisse rilevata sotto uno dei due sensori laterali esterni, il robot non potrà che trovarsi in una delle due situazioni seguenti: in fase di rettilineo, ma molto decentrato rispetto alla linea guida, oppure in fase di **esecuzione di una curva particolarmente stretta**. Il primo caso è molto infrequente: la tenuta di strada su rettilineo, infatti, è garantita dai due sensori laterali più interni. Il secondo caso, invece, è più interessante ed è quello che andremo ad analizzare nel dettaglio. Riconsideriamo l'algoritmo di navigazione visto nei precedenti programmi e, in particolare, l'associazione tra la lettura dei sensori e il movimento eseguito (sempre a pag. 182): quando la linea veniva rilevata al di sotto di uno dei due sensori laterali **Sx1** e **Dx1**, il robot rispondeva con una **rotazione**, ottenuta mantenendo una ruota **ferma**, mentre l'altra continuava a ruotare. In questo modo, la rotazione del robot è centrata sulla ruota ferma e ha un **raggio** pari alla lunghezza dell'asse delle due ruote. Questo movimento consente di assecondare la maggior parte delle curve; tuttavia, in presenza di un raggio di curvatura molto stretto (molto più piccolo dell'asse delle due ruote) il robot può facilmente perdere il tracciato, salvo continuare a ruotare fino a ritrovarlo (come ricorderai, se la lettura dei sensori non è tra quelle previste, il robot continua a eseguire il movimento associato all'ultimo rilevamento riconosciuto). Una possibile soluzione a questo inconveniente è quella di prevedere due tipi di svolta diversi (Ruota e Gira, nel frammento di codice in alto a destra), grazie all'informazione dei sensori laterali esterni Sx2 e Dx2. Infatti, rispetto a Gira, la routine Ruota permette di eseguire una rotazione con un raggio minore:

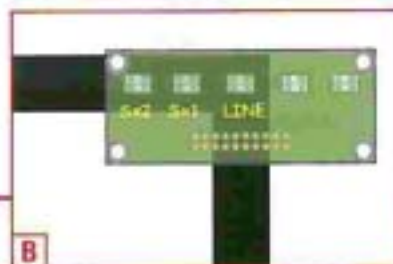
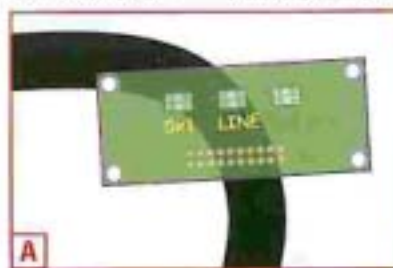
```
Programma Principale
-----
Ruota
-----
goto Lettura_Sensori

Decodifica
lookdown 142bit, [00001 100010 100100 10100 110000] azimut
branch azimut, [RuotaDestra GiraDestra Avanti GiraSinistra RuotaSinistra]

RuotaDestra:
polonost SERRO_SX SERRO_STIP + VELOCITA100
polonost SERRO_SX SERRO_STIP + VELOCITA100
goto Ruota

GiraDestra:
polonost SERRO_SX SERRO_STIP
polonost SERRO_SX SERRO_STIP + VELOCITA100
goto Ruota
```

invece di tenere ferma una delle due ruote, le fa **ruotare entrambe**, una nel senso opposto all'altra. Si ottiene così una rotazione centrata nel punto **mediante** dell'asse delle ruote, dunque, con raggio pari alla **metà** della lunghezza dell'asse stesso. Di conseguenza, la risposta del robot è più precisa e permette di assecondare meglio il tracciato. Osservando il frammento di codice, è importante notare che questo è possibile perché, in prima istanza, i vari tipi di letture dei sensori che si intendevano gestire sono stati aggiunti nell'istruzione **lookdown**, tenendo conto della nuova gestione dei bit di **ffBits**; di conseguenza, nel **branch** è stato possibile aggiungere le due nuove routine (**RuotaDestra** e **RuotaSinistra**), per la gestione dei sensori laterali esterni **Sx2** e **Dx2** (a titolo d'esempio, il frammento mostra solo l'implementazione di **RuotaDestra**). Tornando al discorso generale, considera che, disponendo ormai di cinque sensori, è possibile far navigare il robot su circuiti più complessi: ora, infatti, possiamo rilassare alcuni vincoli inizialmente posti sulle caratteristiche del tracciato e sfruttare anche le letture 'miste' dei sensori. Consideriamo, ad esempio, le curve 'a gomito' (ossia quelle a 90°). Con la configurazione a tre sensori, erano difficilmente gestibili. In quel caso, infatti, il DeA LF avrebbe rilevato la linea con due sensori, quello centrale e uno dei due laterali. Tale lettura, tuttavia, non permetteva, allora, un'interpretazione univoca del tracciato: come appare evidente dalla figura **A** sotto, anche l'esecuzione di una curva normale avrebbe potuto ottenere la stessa lettura sensoriale. I casi **%011** o **%110**, cioè, fornivano informazioni ambigue che potevano comportare alcuni errori di valutazione in curva. Con la configurazione a cinque sensori, invece, i sensori che rileverebbero la curva a gomito sono tre (il centrale e i laterali, interno ed esterno, a destra o sinistra). Di conseguenza si avrebbero le configurazioni **%00111** oppure **%11100**



(a seconda che la curva sia a destra o a sinistra). Una rilevazione di questo tipo (la figura **B** illustra quella relativa alla curva a sinistra) non comporta ambiguità, giacché non può essere confusa con quella di una normale curva (il sensore laterale esterno non rileverebbe la linea). In quanto distinta, dunque, tale rilevazione può essere gestita con un'apposita routine: ad esempio, **Ruota** (precedentemente discussa), ma anche un'altra, magari con parametri di velocità diversi. Con una rotazione più veloce, ad esempio, il robot tornerà ad allinearsi al tracciato rettilineo più rapidamente.