

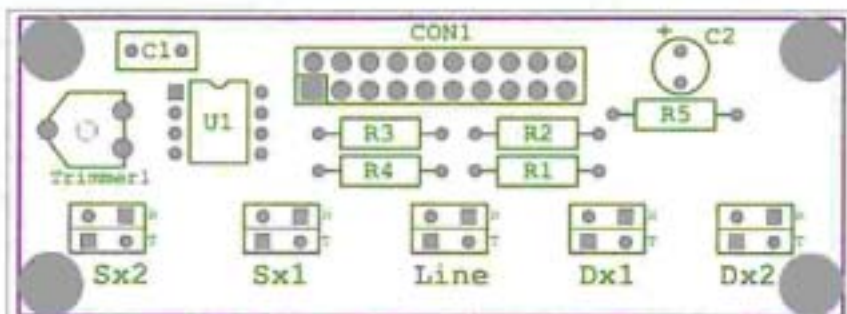


A seconda del tipo di traccia

Ci occuperemo ora della scheda DeA Line Follower, allegata a questo fascicolo. Come avevamo accennato alle pagine 155 e 157, grazie a questo nuovo componente (corredato del chip LM 311N e dei sensori monoblocco in tuo possesso) potrai sviluppare e potenziare ulteriormente le capacità del tuo robot che, equipaggiato con il dispositivo completo (come vedremo nel prossimo fascicolo), sarà in grado di seguire, in modo del tutto autonomo, percorsi anche complessi tracciati con una semplice linea disegnata o applicata sulla superficie da percorrere (sia essa un tavolo, il pavimento ecc.).

DOTI DA SEGUIO

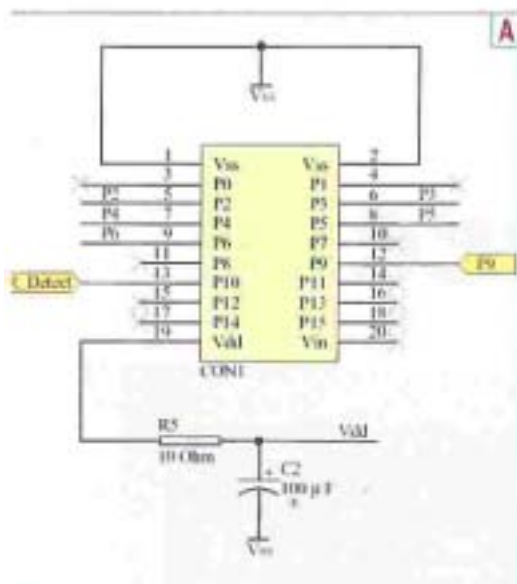
Quello dell'inseguimento automatico di percorsi segnalati da linee o, più in generale, sfruttando particolari riferimenti posti nell'ambiente (i cosiddetti *landmarks*, di cui si è parlato nella sezione Tecno, alle pagg. 148-149) è un aspetto molto importante della robotica mobile. Basti pensare ai robot in grado di muoversi all'interno di fabbriche oppure uffici, ad esempio per consegnare la posta, seguendo opportuni 'indizi' quali linee magnetiche sul pavimento



Sopra. La scheda DeA Line Follower allegata a questo fascicolo e lo schema dei suoi componenti: U1 è la sede per il chip LM 311N, mentre Sx2, Sx1, Line, Dx1 e Dx2 sono sedi predisposte per i sensori IR monoblocco. Gli altri componenti (già presenti) sono invece il trimmer1, da 1 k Ω ; i capacitori C1 (da 100 nF) e C2 (da 100 μ F); i resistori R1 (da 220 Ω), R2-4 (da 10 k Ω) e R5 (da 10 Ω); il connettore CON1 a 20 pin.

oppure adesivi colorati posti sul soffitto, riuscendo anche a evitare eventuali ostacoli presenti lungo il percorso. Inoltre, gruppi di ricercatori hanno trasferito queste tecniche anche alle automobili,

costruendo prototipi dotati di 'pilota automatico' che, senza alcun intervento umano, riesce a seguire correttamente la strada (individuando le linee che delimitano la carreggiata), evitare ostacoli e interagire con gli altri veicoli (mantenendo le opportune distanze per una guida in sicurezza). E ora, grazie al DeA Line Follower, anche tu potrai sviluppare e sperimentare alcune tecniche di guida automatica simili, costruendo diversi percorsi per il tuo robot e, magari, sfidando in gare di navigazione i robot degli amici.



Per farlo, sfruttano le proprietà dei materiali di riflettere o di assorbire i raggi IR emessi per rilevare il percorso da seguire. Proprietà che variano a seconda del colore del materiale di cui è fatta una superficie: le superfici nere, in particolare, tendono ad assorbire la quasi totalità della radiazione; quelle bianche si comportano da specchio, riflettendo i raggi IR. Sfruttando questa peculiarità, si può allora distinguere, ad esempio, la presenza o meno di una linea nera su una superficie bianca (o viceversa, di una linea bianca su una superficie nera) e seguirla.

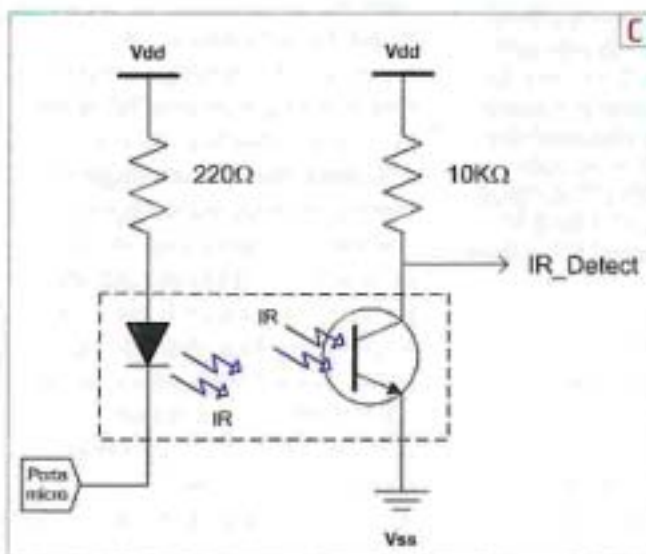
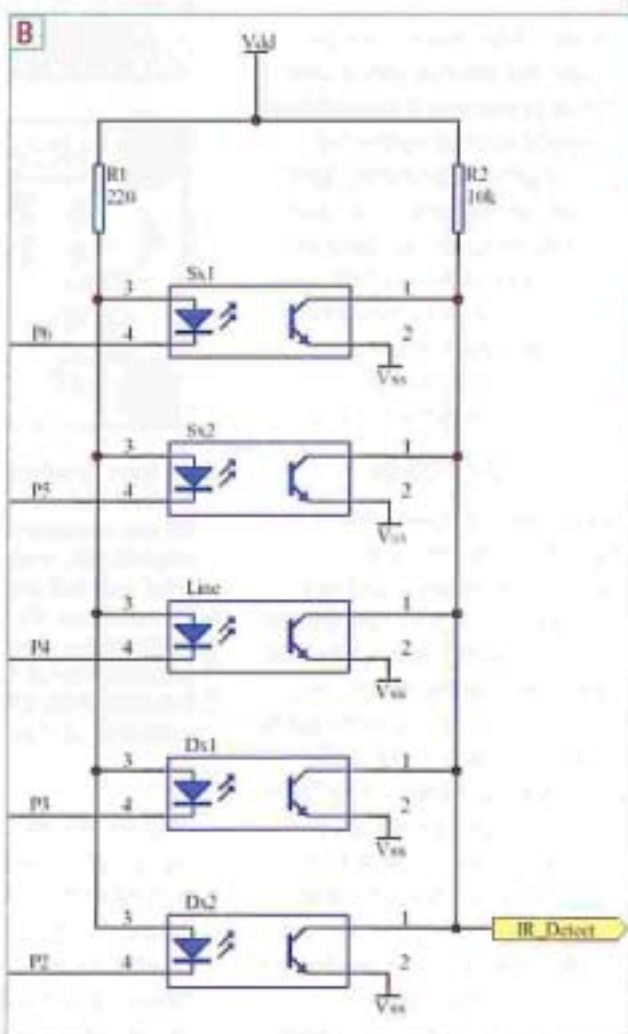
CONNESSIONI

Riguardo alla connessione della scheda DeA Line Follower al microcontrollore, essa avverrà tramite un cavo piatto che permette la comunicazione tra il connettore a 20 pin e il bus di espansione sulla scheda madre. Lo schema della loro connessione è riportato nella figura A, a sinistra; le fasi di montaggio, invece, saranno illustrate prossimamente. Vediamo ora più in dettaglio la struttura della scheda. Da un punto di vista funzionale, è possibile individuare due porzioni distinte: la serie delle

L'HARDWARE

Vediamo ora come si presenta il DeA Line Follower e di quali elementi è costituito. La scheda dispone di una serie di cinque sedi predisposte per l'alloggiamento dei sensori monoblocco IR che, come sai, sono costituiti da una coppia LED-ricevitore IR ciascuno. In questo caso, i tre sensori di cui disponi non saranno utilizzati per rilevare ostacoli, ma per distinguere i contrasti di colore.

Nelle figure. Lo schema della corrispondenza dei 20 pin del connettore con le porte logiche (in alto a sinistra, fig. A) e il circuito delle cinque sedi per i sensori monoblocco (a destra, fig. B). Sotto (fig. C), il particolare generalizzato valido per ciascuna sede e relativo sensore (la parte tratteggiata).



sedi per i sensori IR, da un lato, e il 'circuitto comparatore a soglia', dall'altro. Nello schema circuitale proposto nella pagina accanto (in figura B), sono rappresentati i collegamenti tra le sedi dei sensori e le rispettive porte del microcontrollore.

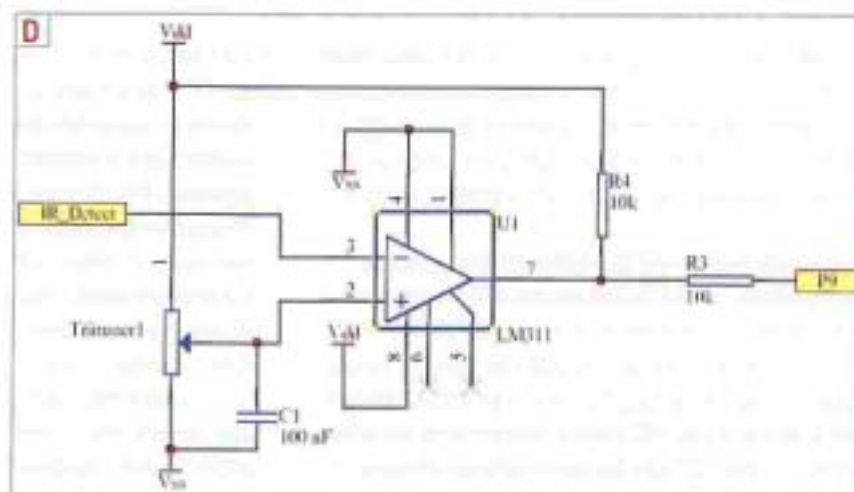
Noterai che questo collegamento andrà a impegnare le porte comprese tra P2 e P6: questo significa che, quando userai il DeA Line Follower, non potrai invece usare, contemporaneamente, i motori a spazzola (collegati a P3 e P5), i baffi (a P4 e P6) e il tasto di reset (a P2). Come mostrato nella figura C (che è il particolare generalizzato di uno dei collegamenti), ogni porta logica comanda un emettitore IR (il LED nel riquadro tratteggiato, sulla sinistra): settando la porta a un valore basso, l'emettitore corrispondente risulterà acceso; viceversa, con valore logico alto il LED rimane spento. Quando è acceso, la radiazione riflessa dalla superficie sottostante viene 'catturata' dal ricevitore (il transistor nel riquadro, sulla destra), generando un flusso di corrente proporzionale alla quantità di radiazione catturata.

Il transistor, inoltre, è posto in serie con un resistore da 10 k Ω : dunque, il valore che verrà utilizzato come misura (in uscita su IR_Detect) sarà quello della tensione presente alla base del resistore. In questo modo quanta piú corrente passa per il transistor (cioè quanto piú questo può essere approssimato a un interruttore chiuso), minore sarà il valore di tensione misurato

Il DeA Line Follower registra la misura effettuata da un sensore per poi compararla con una soglia

(che, infatti, tenderà ad assumere la tensione di massa Vss). Viceversa, minore sarà la corrente (per cui il transistor può essere visto come interruttore aperto), maggiore sarà la tensione alla base del resistore (che tenderà a Vdd=5V). In ogni caso, il valore di tensione misurato (IR_Detect) viene trasmesso alla seconda porzione del circuito, atta alla comparazione. Di fatto, il circuito comparatore (illustrato sopra, in figura D) è composto dal chip LM311N, dal trimmer da 1 k Ω , da due resistori da 10 k Ω e dal capacitore da 100 nF. La sua

uscita è collegata alla porta P9 e può assumere valore logico basso (0) o alto (1), in base ai due ingressi (sopra, indicati con + e -) del chip LM311N. Se l'ingresso -, cui è connesso IR_Detect, ha un valore di tensione maggiore dell'altro (+), l'uscita del circuito avrà un valore basso (0) e viceversa. L'utilizzo del trimmer permette di calibrare



Sopra. Lo schema del circuito comparatore a soglia (fig. D) che riceve in input IR_Detect da uno dei sensori IR.

in maniera fine l'intero circuito, in modo da tener conto sia del diverso grado di riflessione delle superfici sia delle condizioni di illuminazione dell'ambiente. Infatti, variando il valore di resistenza del trimmer varia anche il valore di tensione all'ingresso + del chip e, quindi, la soglia in base alla quale valutare la misura di IR_Detect. Ad esempio, su una superficie non molto riflettente (scura), IR_Detect ha generalmente un valore di tensione maggiore: per adeguare il valore di tensione dell'ingresso +, alzandolo, basterà diminuire il valore di resistenza del trimmer. Resta solo da notare che, poiché ciascun sensore registra la propria misura e la invia al circuito comparatore e poiché tutti sono collegati in parallelo allo stesso (e unico) circuito comparatore, perché questo possa funzionare correttamente è necessario che i sensori lavorino uno alla volta.

Le fasi di programmazione

Prima di passare alla programmazione del DeA Line Follower, concludiamo il discorso relativo all'uso del microfono (iniziato a pag. 167). Implementato il circuito corrispondente sulla breadboard, collega il robot al PC con il cavo seriale ed esegui il programma (sotto).

Osservando il listato, noterai che il programma non introduce alcuna nuova istruzione. Il suo funzionamento è molto semplice: **il robot, infatti, reagirà a ogni segnale acustico (un rumore o un suono scelto come comando, purché sufficientemente intenso), avanzando per un intervallo di tempo pari alla durata del segnale stesso.** Come si diceva a pag. 168, inoltre, intervenendo sul valore di resistenza del trimmer RV1 che hai posto sulla breadboard (ruotandone la vite), potrai osservare direttamente il variare del range entro cui il robot può percepire i rumori e, dunque, reagire: in assenza di rumori 'significativi' (rispetto alla soglia fissata), infatti, **il transistor Q1 rimane chiuso e il valore della porta P2 (intesa come input) è basso.** Se un rumore supera la soglia, invece, il transistor si apre per la durata di persistenza del rumore, il che equivale, per P2, a un impulso alto (rilevabile perciò tramite l'istruzione `pulsin`). Il punto cruciale del programma è contenuto nel `Main` e riguarda, in particolare, la coppia di istruzioni `pulsin mic_in, alto, time` e `If time > soglia then Avanza` (separate solo dalla istruzione di `debug`). Come dichiarato in partenza, la variabile `mic_in` è associata al valore in input su P2 (`in2`) ed è di tipo `var`, mentre `time` è di tipo `word`; la costante `alto` vale 1, mentre `soglia` vale 10. La coppia di istruzioni comporta che, se su P2 si rileva una pulsazione positiva (1) di durata superiore al valore di `soglia * 2 μs` (in base alla sintassi di `pulsin`, infatti, l'unità di misura della variabile `time` è 2 μs), allora il robot avanza (grazie al ciclo `for` contenuto nella routine `Avanza`). Il test sulla durata (`if... then`) è stato introdotto per evitare che rumori o suoni di durata molto breve (spesso casuali e poco controllabili) attivino il robot inavvertitamente. Detto questo, avrai notato che **i parametri di soglia che incidono sulla sensibilità del robot sono due e dipendono, uno, dalla resistenza variabile**

del trimmer Q1; l'altro, dal valore della costante soglia, scelto in fase di programmazione. A seconda dell'intervento, si potrà dunque variare la sensibilità del sistema rispetto all'intensità dei segnali acustici (con il trimmer), oppure rispetto alla loro durata (con la costante). Per sfruttare la prima possibilità, è importante conoscere il range di rendimento del trimmer: **prova dunque a portare la sua resistenza al minimo** (ruotando la vite, come spiegato a pag. 168) e a produrre suoni sempre più deboli, finché uno lo sarà a tal punto da non essere rilevato (il robot, infatti, rimarrà fermo). Ora **inverti il test: mantieni il suono a quell'intensità e aumenta gradualmente la resistenza del trimmer (vale a dire la sua sensibilità)**; a patto che il suono non sia troppo debole, dovresti arrivare a un punto in cui il robot, rilevandolo, si muove. In questo modo, verificato il rendimento del trimmer, potrai calibrare il robot perché non venga disturbato da rumori ininfluenti: ad esempio, in un ambiente che presenti un costante rumore di fondo (purché non così elevato da impedire l'utilizzo del microfono), potrai tarare la sensibilità del sistema in modo da escludere la rilevazione di tale rumore, a favore di quella di comandi acustici 'significativi' (purché più intensi). **La soglia imposta con la costante, invece, ti permette di escludere altri rumori, in base alla loro durata.** In alcuni casi, potrebbe esserti utile fare a meno di questo parametro, settandolo a 0: in tal caso, il robot reagirebbe a qualunque rumore superiore alla soglia fissata con il trimmer e un semplice battito di mani potrebbe attivarne il moto, a patto però che si operi in un ambiente per il resto silenzioso. Nel caso di brevi rumori di disturbo, invece, converrà settare la soglia a un valore alto, in modo che il robot reagisca solo ai comandi desiderati (purché sufficientemente lunghi da superare la soglia).

testModuloServo02

```
(*STAMP BS2)
*Programma per testare il funzionamento del circuito con microfono
*tramite il servomotori

'dichiarazione variabili
mic_in    var    in2
time      var    word
ciclo     var    byte

'dichiarazione costanti
motore0x  con    12
freq0x    con    850
motore0x  con    13
freq0x    con    650
alto      con    1
soglia    con    10

*Programma
-----
Main:
  pulsIn mic_in, alto, time      'rileva sulla porta 0 il tempo per cui un'eventuale
                                pulsazione si sentiva sul fronte alto

  debug cin.doc ? time         'mostra nella finestra di debug il tempo rilevato
  If time > soglia then Avanza 'se il tempo supera un certo valore soglia (10) allora il motore
                                è da considerarsi sufficientemente per attivare il robot
                                variando questo parametro si può calibrare il funzionamento del robot

  goto Main

Avanza:
  For ciclo =1 to 100
    pulsOut motore0x, freq0x
    pulsOut motore0x, freq0x

  next
  goto Main
```