



Il kit per il cicalino

Il cicalino (o cicala) è un avvisatore elettroacustico, in grado di emettere diversi toni (simili a un ronzio) modulabili e temporizzabili. Tramite un opportuno circuito realizzato sulla breadboard, potrai presto programmarne il funzionamento, dotando così il robot di una sorta di voce.

Il pulsante del cicalino, essendo un generico selettore ON/OFF, può essere utilizzato anche con altri dispositivi.

Nella foto. Allegati a questo fascicolo, il cicalino 1, il pulsante ALPS 2, due resistori da 10 K Ω 3, uno da 10 Ω 4 e un transistor BC 337 5.



Le fasi di programmazione

Come abbiamo visto a pagina 116 e 124, per ottenere la calibrazione dei servomotori si applicano loro impulsi di durata diversa in grado di compensare le differenti velocità di rotazione. Una volta trovati i valori di *pulsout* che consentono al robot di mantenere traiettorie rettilinee, possiamo passare alla fase successiva, ossia quella che riguarda il controllo delle distanze. Come sai, rispetto ai motori a spazzola i servomotori offrono

il vantaggio di un maggior controllo e precisione nell'esecuzione dei movimenti. Utilizzandoli per la locomozione del robot, in particolare, puoi programmarlo in modo da ottenere gli spostamenti desiderati, determinando inoltre la distanza effettivamente percorsa di volta in volta. Per farlo è necessario ricordare alcune semplici relazioni geometriche ed effettuare qualche esperimento sui movimenti del robot.

Per ottenere una stima della distanza percorsa dal robot possiamo utilizzare un metodo molto semplice: calcolarla in base al numero di giri effettuati dalle ruote e alle dimensioni di queste ultime. Quando le ruote del robot compiono un giro completo, infatti, il robot sarà avanzato (o indietro) per un tratto pari alla misura della circonferenza della ruota stessa. In particolare, le ruote del tuo robot hanno un **diametro complessivo** (considerando anche il copertone in gomma) approssimabile a 4,6 cm. Dal momento che la circonferenza di un cerchio si calcola moltiplicandone il diametro per π greco ($\pi = 3.14159\dots$), la ruota del tuo robot ha una **circonferenza di circa 14,5 cm** ($4,6 \cdot \pi$), il che significa che, **a ogni giro completo delle ruote, il robot percorre una distanza pari a 14,5 cm**. Per sfruttare questa informazione, però, è necessario fissare un riferimento che ti permetta di determinare il numero di giri effettivi compiuti di volta in volta dalla ruota. A tal proposito puoi costruire un semplice programma di test (a destra) con cui effettuare una serie di esperimenti volti a ricavare una relazione tra il numero di giri effettuati dalle ruote e i valori utilizzati nel programma. Prima, però, sarà utile marcare un riferimento 'fisico' (ad esempio un segno tracciato con un pennarello) sulle ruote stesse, in modo da verificare empiricamente l'entità della rotazione eseguita dalla ruota e poter decidere le eventuali variazioni dei valori del listato.

```

wd_8rbc2 |
' (STAMP B52)
' Programma di test per il controllo delle distanze percorse
' ----- Variabili -----
contatore var word
FBin var in2
' ----- Costanti -----
SERVO_SX con 12
SERVO_DX con 13
ORARIO_DX con 500
ANTIORARIO_SX con 1000
' ----- Inizializzazione -----
low SERVO_SX
low SERVO_DX
' ----- Check tasto di Start/Stop -----
Aspetta:
if FBin = 1 then Aspetta
' ----- Programma principale -----
for contatore = 1 to 100
  pulsout SERVO_SX, ANTIORARIO_SX
  pulsout SERVO_DX, ORARIO_DX
  pause 20
next
end

```


LE FASI DI PROGRAMMAZIONE

Il programma di pag. 129 è molto semplice ed è costituito da una **routine di movimento** che, utilizzando come valori di **pulsout** quelli calcolati con la calibrazione (nel listato sono riportati i valori di default 500 e 1000, da sostituire con quelli reali), fa avanzare a velocità massima il robot. Nel programma sono inoltre dichiarate la variabile **contatore** di tipo **word** e l'aliasing **PBin** relativo alla porta **P2** del tasto start/stop. Riconoscerai facilmente l'inizializzazione delle porte dedicate ai servomotori (**low SERVO_SX/DX**) e la parte relativa alla gestione del tasto start/stop (**Aspetta... then Aspetta**). Il programma principale è invece costituito da un **ciclo for**, all'interno del quale sono inserite le due istruzioni **pulsout** e quella **pause** per la gestione dei servomotori; questo comporta che la sequenza delle tre istruzioni del ciclo (**pulsout... pulsout... pause...**) venga ripetuta fino a quando il valore della variabile **contatore** assume il valore limite, nell'esempio 100 (**for contatore = 1 to 100**). Gli impulsi vengono quindi applicati ai due servomotori per 100 volte, a intervalli di **20 ms**, ottenendo un movimento uniforme delle due ruote. Dal momento che, aumentando o diminuendo il numero delle ripetizioni si modifica la durata del movimento delle due ruote e, quindi, dello spazio percorso dal robot, con questo programma di test è possibile determinare il numero di iterazioni del corpo del ciclo **for** affinché la ruota compia esattamente un giro completo. Vediamo come. Digita il listato (avendo cura di sostituire ai valori delle due costanti **ORARIO_DX** e **ANTIORARIO_SX** i valori reali calcolati con la precedente calibrazione) e mandalo in esecuzione premendo il tasto **Run** della barra degli strumenti; quindi scollega il cavo seriale e premi il tasto **start/stop** per avviare il robot. Quando si sarà fermato, controlla la ruota marcata con il pennarello in precedenza e verifica la posizione del segno: **se la ruota è tornata esattamente nella posizione iniziale, significa che il numero (100) di iterazioni del ciclo for corrisponde a un giro completo alla ruota**. In caso contrario, invece, modifica il valore finale della variabile **contatore**: **diminuendolo** se la ruota ha compiuto più di un giro, **aumentandolo** nel caso opposto. Ripeti l'esperimento più volte, così da determinare un valore sufficientemente preciso: puoi cominciare con il valore 100 proposto nel listato, che dovrebbe essere abbastanza vicino a quello reale, quindi variarlo a seconda dei casi. **Una volta determinato il numero di iterazioni necessario per ottenere un giro completo, potrai stabilire a priori le distanze da far percorrere al robot**. Se, ad esempio, vuoi che percorra un tratto rettilineo in avanti di 40 cm, sapendo che la ruota percorre 14.5 cm a ogni giro, dovrai far compiere alle ruote 2.76 giri (40/14.5); sapendo inoltre che un giro completo della ruota corrisponde al numero di iterazioni del ciclo **for** trovato, nell'esempio 100, saranno necessarie 276 iterazioni (2.76*100). Per realizzare questo movimento (procedere per 40 cm), dunque, dovrai modificare opportunamente il ciclo **for** (sotto).

In pratica, mantenendo inalterate le tre istruzioni di gestione dei servomotori, dovrai modificare solo il valore finale che deve assumere la variabile **contatore** (non più 100, ma 276). Come avrai notato, **la relazione tra numero di giri delle ruote e numero di iterazioni del ciclo for è di tipo lineare**: per realizzare n giri, infatti, basta moltiplicare per n il numero di iterazioni trovato che realizza il singolo giro. Quindi ripeti la stessa misurazione per il movimento del robot all'indietro, avendo cura di sostituire le costanti **ORARIO_DX** e **ANTIORARIO_SX** con le costanti **ORARIO_SX** e **ANTIORARIO_DX** accoppiate ai rispettivi valori precedentemente calibrati (sotto).

```

test_dista2
(*STAMP B57)
' Programma di test per il controllo delle distanze percorse
'----- Variabili -----
contatore var word
PBin      var in2
'----- Costanti -----
SERVO_SX   con 12
SERVO_DX   con 13
ORARIO_SX  con 500
ANTIORARIO_DX con 1000
'----- Inizializzazione -----
low SERVO_SX
low SERVO_DX
'----- Check tasto di Start/Stop -----
Aspetta
if PBin = 1 then Aspetta
'----- Programma principale -----
for contatore = 1 to 100
    pulsout SERVO_SX, ORARIO_SX
    pulsout SERVO_DX, ANTIORARIO_DX
    pause 20
next
end

```

Data la possibilità di avere differenti velocità per i motori o diversa calibrazione effettiva per i due movimenti (avanti e indietro), anche il numero di iterazioni richieste potrà essere leggermente diverso. I valori che troverai, inoltre, sono legati alla particolare struttura del codice utilizzato per gestire i due motori: saranno cioè calibrati per impulsi impartiti ogni 20 ms (**pause 20**). **Se la frequenza degli impulsi viene cambiata, le relazioni trovate non sono più valide; in tal caso, dunque, sarà necessario procedere a nuove misurazioni per ricavare i relativi valori effettivi**. Se ad esempio volessi utilizzare una pausa minore, poniamo 10 ms, i due motori avranno una velocità e uno spunto (cioè uno 'slancio') leggermente superiori. Gli impulsi, però, sono distribuiti su un arco di tempo minore; perciò, a parità di iterazioni, si otterrà un numero di giri minore (dunque meno 'strada').

```

'----- Programma principale -----
for contatore = 1 to 276
    pulsout SERVO_SX, ANTIORARIO_SX
    pulsout SERVO_DX, ORARIO_DX
    pause 20

```