

Capitolo 6: Navigazione Sensibile alla Luce con le Fotoresistenze

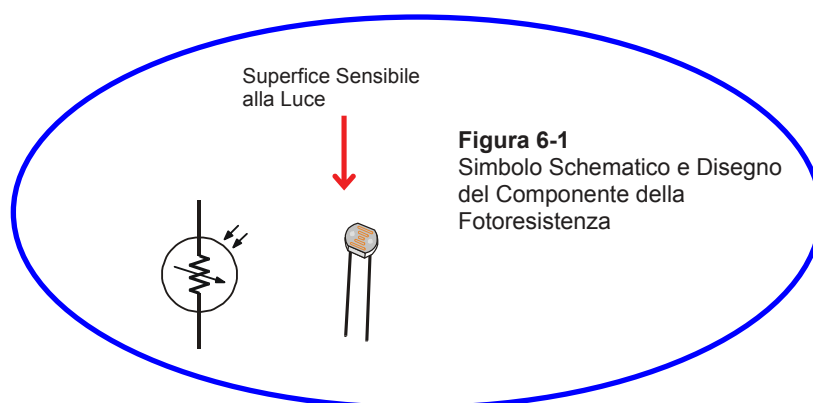
La luce ha molte applicazioni nella robotica e nel controllo industriale. Alcuni esempi includono la rilevazione del bordo dei rotoli di tessuto in una industria tessile, la determinazione dell'ora di accensione dell'illuminazione pubblica nei vari periodi dell'anno, quando scattare una foto o quando irrigare una coltura di piante.

Esistono molti sensori di luce differenti per usi diversi. Il sensore di luce usato nel vostro kit del Boe-Bot è progettato per rilevare la luce visibile, e può essere usato per far rilevare al vostro Boe-Bot variazioni nel livello di illuminazione. Con questa abilità, il vostro Boe-Bot può essere programmato per riconoscere aree con zone di luce o di ombra, registrare la luminosità ed il livello di oscuramento che percepisce, ricercare sorgenti di luce come raggi di una torcia elettrica e porte che lascino filtrare luce in una stanza buia.

6

PRESENTAZIONE DELLA FOTORESISTENZA

Le Resistenze con cui avete lavorato nei capitoli precedenti avevano valori fissi, come ad esempio 220 Ω e 10 k Ω . La fotoresistenza, d'altro canto, è una resistenza dipendente dalla luce (LDR). Questo significa che il suo valore di resistenza dipende dalla brillantezza, o dalla luminosità, della luce incidente sulla sua superficie sensibile. La Figura 6-1 mostra il simbolo schematico ed il disegno del componente della fotoresistenza che userete per mettere in grado il Boe-Bot di rilevare le variazioni dei livelli di luce.





Una Fotoresistenza è una Resistenza Dipendente dalla Luce (LDR) che ha una sensibilità allo spettro luminoso simile all'occhio umano. In altre parole, il genere di luce che viene percepito dai vostri occhi è dello stesso tipo che influisce sulla resistenza della Fotoresistenza. Gli elementi attivi di queste fotoresistenze sono fatti di Solfuro di Cadmio (CdS). La luce penetra nello strato semiconduttore applicato su un substrato ceramico e produce dei portatori di carica liberi. Viene prodotta una determinata resistenza elettrica che è inversamente proporzionale all'intensità luminosa. In altre parole, l'oscurità causa una resistenza elevata, e la luce causa una bassa resistenza.

Luminosità è un nome scientifico per la misura della luce incidente. Un modo di comprendere la luce incidente è di pensare ad un fascio luminoso proiettato su di un muro. Il raggio focalizzato che vedete brillare sul muro è la luce incidente. L'unità di misura della luminosità è comunemente chiamato "foot-candle" nel sistema inglese oppure "lux" nel sistema metrico. Nell'uso corrente delle fotoresistenze non vi interesserebbe dei livelli in lux, ma solamente se la luminosità è maggiore o minore in alcune direzioni. Il Boe-Bot può essere programmato per usare l'intensità relativa come informazione per prendere decisioni sulla navigazione.

ESERCIZIO #1: MONTAGGIO E COLLAUDO DEI CIRCUITI DELLA FOTORESISTENZA

In questo esercizio, assemblerete e collauderete i circuiti con fotoresistenze per rilevare i livelli di luce. I vostri circuiti sensori di livello saranno in grado di rilevare la differenza tra ombra e luce. I comandi PBASIC per la determinazione di una condizione di ombra sulla fotoresistenza saranno molto simili a quelli usati per determinare se un baffo ha toccato un oggetto.

Elenco Componenti:

- (2) Fotoresistenze - CdS
- (2) Resistenze – 2 k Ω (rosso-nero-rosso)
- (2) Resistenze – 220 Ω (rosso-rosso-marrone)
- (4) Ponticelli in filo
- (2) Resistenze – 470 Ω (giallo-viola-marrone)
- (2) Resistenze – 1 k Ω (marrone-nero-rosso)
- (2) Resistenze – 4.7 k Ω (giallo-viola-rosso)
- (2) Resistenze – 10 k Ω (marrone-nero-arancio)

Assemblaggio degli Occhi Fotosensibili

La Figura 6-2 mostra lo schema elettrico e la Figura 6-3 mostra lo schema di cablaggio dei circuiti fotoresistenza che userete in questo e nei due prossimi esercizi.

- ✓ Scollegate l'alimentazione dalla vostra scheda e dai servo.
- ✓ Assemblate il circuito mostrato in Figura 6-2, usando la Figura 6-3 come riferimento.

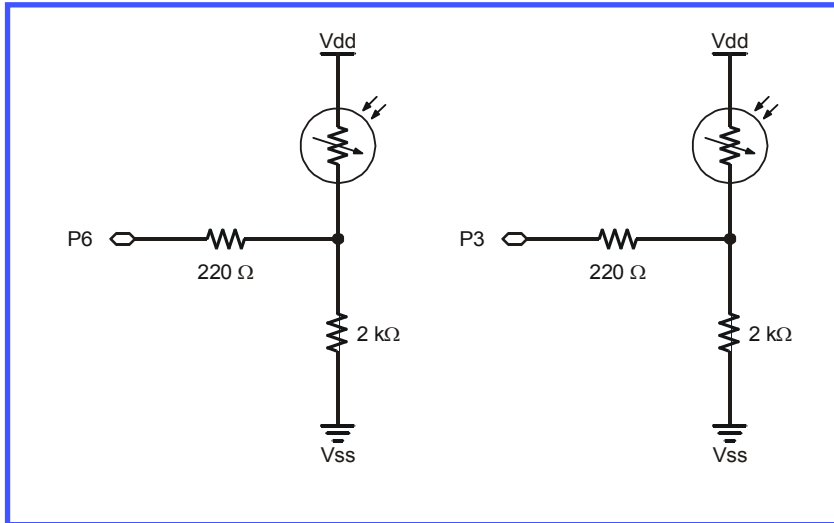


Figura 6-2
Schema
Elettrico del
Primo
Circuito di
Rilevazione
della Luce

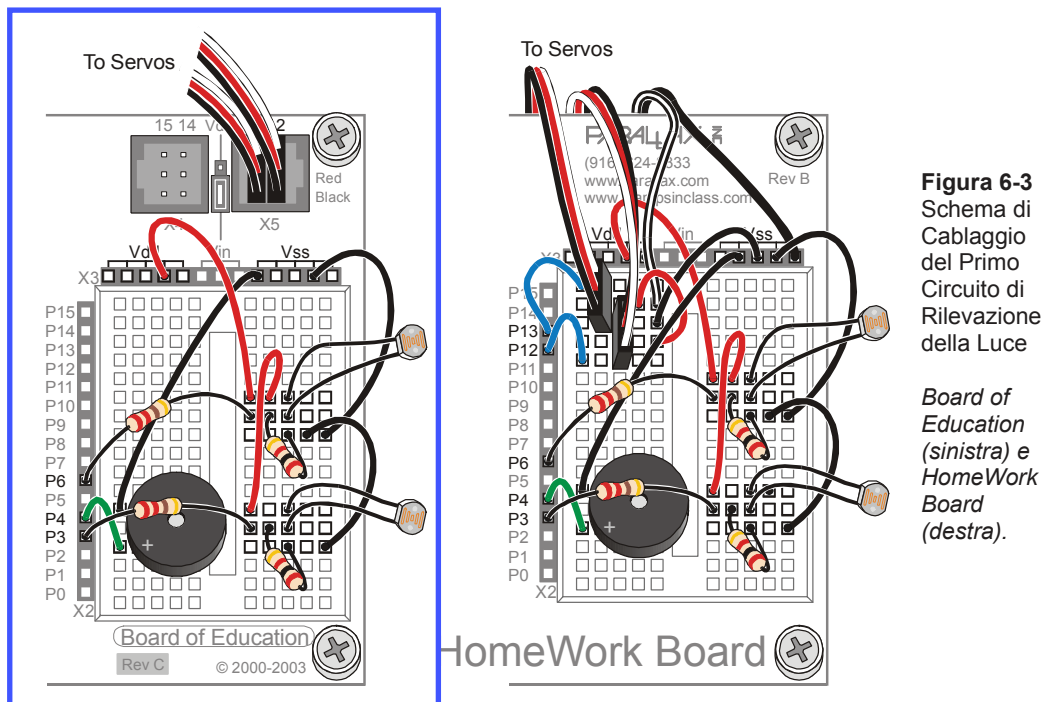


Figura 6-3
Schema di
Cablaggio
del Primo
Circuito di
Rilevazione
della Luce

*Board of
Education
(sinistra) e
HomeWork
Board
(destra).*

Come Funziona il Circuito della Fotoresistenza

Un pin I/O del BASIC Stamp può funzionare come uscita o come ingresso. Come uscita, il pin I/O può inviare un segnale alto (5 V) o basso (0 V). fino a questo punto, i segnali alti e bassi sono stati usati per accendere o spegnere circuiti LED, controllare servo, ed inviare toni ad un cicalino. Un pin I/O del BASIC Stamp può anche funzionare come ingresso. Come ingresso, il pin I/O non applica nessuna tensione ai circuiti a cui è collegato. Invece, ascolta silenziosamente senza alcun effetto sui circuiti. Se il pin I/O sente che la tensione applicata è superiore ad 1.4 V, memorizza un 1 nel suo registro di ingresso. Se invece sente che la tensione applicata è inferiore ad 1.4 V, memorizza uno 0 nel suo registro di ingresso. Nel capitolo precedente, questi registri di ingresso memorizzavano valori che indicavano se un baffo era stato premuto o meno. Per esempio, il registro di ingresso **IN7** memorizzava un 1 quando sentiva 5 V (baffo non premuto), oppure uno 0 quando sentiva 0 V (baffo premuto).



I pin I/O del BASIC Stamp per default sono ingressi. Quando un programma BASIC Stamp inizia, tutti i pin I/O partono come ingressi. Quando usate comandi come **HIGH**, **LOW**, **PULSOUT** o **FREQOUT**, il pin I/O viene cambiato da ingresso ad uscita in modo che il BASIC Stamp possa inviare i segnali alto o basso.

Quando un pin I/O del BASIC Stamp è un ingresso, il circuito si comporta come se non siano presenti né il pin I/O né la resistenza da 220 Ω . La Figura 6-4 mostra il circuito equivalente. La resistenza della fotoresistenza viene indicata con la lettera **R**. può andare da pochi Ω se la luce è molto intensa, fino a circa 50 k Ω nell'oscurità completa. In una stanza ben illuminata da plafoniere fluorescenti sul soffitto, la resistenza può essere dell'ordine di 1 k Ω (piena esposizione alla luce) oppure circa 25 k Ω (ombra proiettata dagli oggetti presenti). Come la resistenza della fotoresistenza cambia con l'esposizione alla luce, così fa la tensione al punto V_o ; se la **R** aumenta, V_o diminuisce, e se **R** diminuisce, V_o aumenta. V_o è ciò che il pin I/O del BASIC Stamp rileva quando funziona come ingresso. Se questo circuito è collegato a **IN6**, quando la tensione al punto V_o sale sopra 1.4 V, **IN6** memorizzerà un 1. Se V_o scende sotto ad 1.4 V, **IN6** memorizzerà uno 0.

6

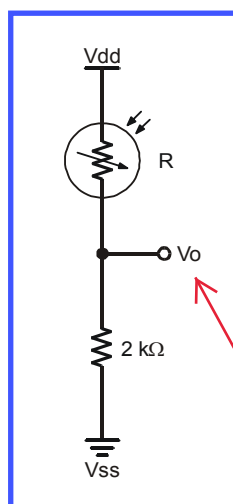


Figura 6-4
Schema Elettrico
del circuito del
Partitore di
Tensione

Ad un ingresso digitale



Quando le resistenze sono collegate terminale con terminale come mostrato nella Figura 6-4 si dice che sono **collegate in serie**, e ci si può riferire ad esse come a resistenza serie.

Quando due resistenze sono collegate in serie per impostare una tensione al punto V_o , il circuito si chiama **Partitore di Tensione**. In questo circuito, il valore di V_o può essere qualsiasi tra V_{dd} e V_{ss} . Il valore esatto di V_o è determinato dal rapporto tra R e $2\text{ k}\Omega$. Quando R è maggiore di $2\text{ k}\Omega$, V_o sarà più vicina a V_{ss} . Quando R è inferiore di $2\text{ k}\Omega$, V_o sarà più vicino a V_{dd} . Quando R è uguale a $2\text{ k}\Omega$, V_o sarà 2.5 V . Se misurate uno dei due valori (R o V_o), potete calcolare l'altro valore usando una di queste due equazioni.

$$V_o = 5V \times \frac{2000\Omega}{2000\Omega + R} \quad R = \left(5V \times \frac{2000\Omega}{V_o} \right) - 2000\Omega$$

1.4 V viene chiamata **tensione di soglia** dei pin I/O del BASIC Stamp, conosciuta anche come **soglia logica** dei pin I/O. Quando la tensione sentita da un pin I/O è superiore a quella soglia, il registro di ingresso dei pin I/O memorizza un 1. Se è inferiore memorizza uno 0.

Rivelare Ombre

Ombreggiare la Fotoresistenza fa aumentare la sua resistenza (R), che a sua volta rende V_o minore. Le Resistenze da $2\text{ k}\Omega$ sono state scelte per rendere il valore di V_o leggermente superiore alla soglia di 1.4 V dei pin I/O del BASIC Stamp in una stanza ben illuminata. Quando con la vostra mano create un'ombra, V_o dovrebbe scendere sotto la soglia di 1.4 V .

In una stanza ben illuminata, ambedue **IN6** ed **IN3** memorizzeranno il valore 1. Se ombreggiate la fotoresistenza del partitore collegato a P6, verrà memorizzato uno 0. Similmente, se ombreggiate la fotoresistenza del partitore collegato a P3, **IN3** sarà forzato a memorizzare uno 0.

Programma Esempio: TestFotoresistenzeDividers.bs2

Questo programma esempio è TestWhiskers.bs2 adattato ai partitori con fotoresistenza. Invece di tenere sotto controllo P5 e P7 come avete fatto con i baffi, terrete ora sotto controllo P3 e P6, che sono collegati ai circuiti dei partitori con fotoresistenza. Questo programma dovrebbe visualizzare il valore 1 in ambedue i circuiti in una stanza ben illuminata. Quando ombreggiate una o tutte e due le fotoresistenze, i loro valori corrispondenti dovrebbero diventare 0.

- ✓ Rialimentate la vostra scheda ed i servo.
- ✓ Digitate, Salvate e Lanciate TestFotoresistenzaDividers.bs2.

- ✓ Verificate che senza ombra, ambedue **IN6** ed **IN3** memorizzino il valore 1.
- ✓ Verificate che ombreggiando con una mano ciascuna fotoresistenza il registro di ingresso corrispondente cambi da 1 a 0.
- ✓ Se siete in difficoltà, sia nel far cambiare a 0 il registro di ingresso ombreggiando una fotoresistenza, o se i registri di ingresso memorizzano uno 0 senza tener conto dell'effettiva ombreggiatura delle fotoresistenze, vedere la sezione Risoluzione dei Problemi del Partitore con Fotoresistenza. Lavorateci fino a quando l'ombreggiatura creata con la mano non generi una variazione affidabile dello stato da 1 a 0.

```
' Robotica con il Boe-Bot - TestFotoresistenzaDividers.bs2
' Visualizza ciò che viene percepito dal pin I/O collegato al partitore
' di tensione con fotoresistenza.

' {$STAMP BS2}
' {$PBASIC 2.5}

DEBUG "STATI delle FOTORESISTENZA", CR,
      " Sinistra      Destra      ", CR,
      " -----      -----      "

DO
  DEBUG CR$RXY, 0, 3,
        "P6 = ", BIN1 IN6,
        "      P3 = ", BIN1 IN3
  PAUSE 100
LOOP
```

Risoluzione dei Problemi del Partitore con Fotoresistenza

Cose Generali da Verificare:

- ✓ Controllate il cablaggio ed il programma per eventuali errori.
- ✓ Assicuratevi che ciascun componente sia fermamente inserito nel suo connettore.
- ✓ Controllate il codice colori delle resistenze. Le resistenze collegate tra Vss e le fotoresistenze devono essere da 2 k Ω (rosso-nero-rosso). Le resistenze collegate da P6 e P3 alle fotoresistenze devono essere da 220 Ω (rosso-rosso-marrone).

Se il registro IN3 ed il registro IN6 non mostrano uno 0 pur ombreggiando le fotoresistenze:



- ✓ Può darsi che la stanza non sia illuminata a sufficienza, considerate l'aggiunta di altre lampade. In alternativa, potete sostituire le resistenze da 2 k Ω con resistenze da 4.7 k Ω (Giallo Viola Rosso). Questo darà ai partitori di tensione un rendimento migliore in condizioni di scarsa illuminazione. Per condizioni di illuminazione veramente basse, potete perfino usare resistenze da 10 k Ω (marrone-nero-arancio).

Se il registro IN3 ed il registro IN6 mostrano un 1 senza reagire all'ombreggiatura:

- ✓ Se la stanza è molto luminosa, e vi trovate a dover avvolgere completamente la superficie sensibile delle fotoresistenze per far commutare da 1 a 0, potreste dover sostituire le resistenze da 2 k Ω con resistenze di valore più basso. Provate con resistenze da 1 k Ω (marrone-nero-rosso), o anche con resistenze da 470 Ω (giallo-viola-marrone) se siete all'aperto.

Il vostro Turno – Sperimentare con Partitori di Tensione Differenti

In relazione con le condizioni di illuminazione dell'area robotica in cui operate, maggiori o minori valori di resistenza serie al posto della resistenza da 2 k Ω possono migliorare le prestazioni dei vostri rivelatori di ombra.

- ✓ Ricordate di scollegare l'alimentazione della vostra scheda prima di ogni modifica circuitale.
- ✓ Provate a sostituire le resistenze da 2 k Ω (rosso-nero-rosso) con ciascun valore delle resistenze che avete raccolto: 470 Ω , 1 k Ω , 4.7 k Ω , e 10 k Ω .
- ✓ Provate ciascuna combinazione di partitore di tensione con il programma TestFotoresistenzaDividers.bs2 e determinate quali resistenze funzionano meglio con le condizioni di illuminazione del vostro ambiente. La migliore combinazione è quella non ipersensibile ma anche che non vi richiede di schermare completamente la fotoresistenza.
- ✓ Per i prossimi due esercizi usate la combinazione di resistenze che pensate funzioni meglio.

ESERCIZIO #2: IL MOVIMENTO EVITANDO LE OMBRE DEGLI OGGETTI

Siccome i partitori con fotoresistenza si comportano in maniera simile ai baffi, è bene esaminare che cosa comporta adattare `RoamingWithWhiskers.bs2` in modo che funzioni con i partitori con le fotoresistenze.

Adattamento di `RoamingWithWhiskers.bs2` per i Partitori con le Fotoresistenze

Tutto quello che in realtà dovete fare è aggiustare le istruzioni **IF...THEN** in modo che controllino `IN6` ed `IN3`, invece di `IN7` ed `IN5`. La Figura 6-5 vi dimostra come fare questi cambiamenti.

Figura 6-5: Modifiche di `RoamingWithWhiskers.bs2` per l'uso con i Partitori con Fotoresistenza

6

	' Modifiche per
	' <code>RoamingWithFotoresistenza</code>
' Da <code>RoamingWithWhiskers.bs2</code>	' <code>Dividers.bs2</code>
IF (IN5 = 0) AND (IN7 = 0) THEN	IF (IN6 = 0) AND (IN3 = 0) THEN
GOSUB Back_Up	GOSUB Back_Up
GOSUB Turn_Left	GOSUB Turn_Left
GOSUB Turn_Left	GOSUB Turn_Left
ELSEIF (IN5 = 0) THEN	ELSEIF (IN6 = 0) THEN
GOSUB Back_Up	GOSUB Back_Up
GOSUB Turn_Right	GOSUB Turn_Right
ELSEIF (IN7 = 0) THEN	ELSEIF (IN3 = 0) THEN
GOSUB Back_Up	GOSUB Back_Up
GOSUB Turn_Left	GOSUB Turn_Left
ELSE	ELSE
GOSUB Forward_Pulse	GOSUB Forward_Pulse
ENDIF	ENDIF

Programma Esempio– `RoamingWithFotoresistenzaDividers.bs2`

- ✓ Aprite il programma `RoamingWithWhiskers.bs2` della pagina 175, e salvatelo con il nome `RoamingWithFotoresistenzaDividers.bs2`.
- ✓ Fate le modifiche mostrate nella Figura 6-5.
- ✓ Ricollegate l'alimentazione alla vostra scheda ed ai servo.
- ✓ Lanciate il programma e collaudatelo.



Ombreggiare ambedue le Fotoresistenze allo stesso momento può essere difficoltoso. Quando il Boe-Bot sta procedendo in avanti, controlla le fotoresistenze circa 40 volte/secondo. Vi dovrete muovere velocemente per ombreggiare ambedue le fotoresistenze tra un impulso ed un altro. Agitare le vostre mani rapidamente davanti alle fotoresistenze per ombreggiarle può essere un modo. In alternativa, ombreggiate con le mani le fotoresistenze mentre il Boe-Bot sta eseguendo una manovra. Quando ha finito la manovra e controlla di nuovo lo stato delle fotoresistenze, dovrebbe riconoscere che ambedue le fotoresistenze sono in ombra. □

- ✓ Verificate che il Boe-Bot eviti le ombre usando le vostre mani per proiettare un'ombra sulle fotoresistenze. Provate con le seguenti combinazioni; nessuna ombra, ombra sul partitore a fotoresistenza di destra (circuito collegato a P3), ombra sul partitore a fotoresistenza di sinistra (circuito collegato a P7), ed infine ombra su tutte e due i partitori a fotoresistenza.
- ✓ Aggiornate i commenti in modo che il titolo e le descrizioni della reazione della pressione dei baffi rifletta invece il comportamento dei circuiti con la fotoresistenza. Quando avete finito dovrebbe assomigliare al programma sottostante.

```
' -----[ Titolo ]-----
' Robotica con il Boe-Bot - RoamingWithFotoresistenzaDividers.bs2
' Il Boe-Bot rileva le ombre con il partitore di tensione a fotoresistenza
' ed effettua una curva per allontanarsene.

' {$STAMP BS2}                ' Direttiva Stamp.
' {$PBASIC 2.5}              ' Direttiva PBASIC.

DEBUG "Il Programma è in Funzione!"

' -----[ Variabili ]-----
pulseCount    VAR    Byte      ' Contatore del ciclo FOR...NEXT.

' -----[ Inizializzazione ]-----
--

FREQOUT 4, 2000, 3000          ' Segnale di Start/restart del programma.

' -----[ Routine Principale ]-----
-----

DO
IF (IN6 = 0) AND (IN3 = 0) THEN ' Rileva ambedue le Fotoresistenze
  GOSUB Back_Up                ' Ombra, Retromarcia e curva ad U
  GOSUB Turn_Left              ' (a sinistra due volte).
  GOSUB Turn_Left
```

```

ELSEIF (IN6 = 0) THEN      ' Rileva la Fotoresistenza sinistra
  GOSUB Back_Up            ' Ombra, Retromarcia e curva a destra.
  GOSUB Turn_Right
ELSEIF (IN3 = 0) THEN      ' Rileva la Fotoresistenza destra
  GOSUB Back_Up            ' Ombra, Retromarcia e curva a sinistra.
  GOSUB Turn_Left
ELSE                        ' Nessuna Fotoresistenza rilevata
  GOSUB Forward_Pulse      ' Ombra, applica un impulso in avanti.
ENDIF

LOOP

' -----[ Subroutine ]-----

Forward_Pulse:              ' Invia un singolo Impulso in Avanti.
  PULSOUT 12,650
  PULSOUT 13,850
  PAUSE 20
  RETURN

Turn_Left:                  ' Curva a Sinistra, circa 90-gradi.
  FOR pulseCount = 0 TO 20
    PULSOUT 12, 650
    PULSOUT 13, 650
    PAUSE 20
  NEXT
  RETURN

Turn_Right:                 ' Curva a Destra, circa 90-gradi.
  FOR pulseCount = 0 TO 20
    PULSOUT 12, 850
    PULSOUT 13, 850
    PAUSE 20
  NEXT
  RETURN

Back_Up:                    ' Retromarcia.
  FOR pulseCount = 0 TO 40
    PULSOUT 12, 850
    PULSOUT 13, 650
    PAUSE 20
  NEXT
  RETURN

```

Il Vostro Turno– Migliorare le Prestazioni

Potete migliorare le prestazioni del vostro Boe-Bot commentando (e quindi escludendo dal programma) alcune delle chiamate alle subroutine che erano state progettate per aiutare il Boe-Bot ad evitare gli ostacoli. La Figura 6-6 mostra un esempio dove le due chiamate delle subroutine **Turn_Left** sono state commentate nell'istruzione **IF...THEN**

quando la condizione è che ambedue le fotoresistenze rilevano un'ombra. Poi, quando una sola fotoresistenza rileva un'ombra, è stata commentata la chiamata alla subroutine **Back_Up** in modo che il Boe-Bot effettui solamente la curva in risposta ad un'ombra.

Figura 6-6: Modifiche a RoamingWithFotoresistenzaDividers.bs2

' Estratto da	' Estratto Modificato da
' RoamingWithFotoresistenza	' RoamingWithFotoresistenza
' Dividers.bs2	' Dividers.bs2
IF (IN6 = 0) AND (IN3 = 0) THEN	IF (IN6 = 0) AND (IN3 = 0) THEN
GOSUB Back_Up	GOSUB Back_Up
GOSUB Turn_Left	' GOSUB Turn_Left
GOSUB Turn_Left	' GOSUB Turn_Left
ELSEIF (IN6 = 0) THEN	ELSEIF (IN6 = 0) THEN
GOSUB Back_Up	' GOSUB Back_Up
GOSUB Turn_Right	GOSUB Turn_Right
ELSEIF (IN3 = 0) THEN	ELSEIF (IN3 = 0) THEN
GOSUB Back_Up	' GOSUB Back_Up
GOSUB Turn_Left	GOSUB Turn_Left
ELSE	ELSE
GOSUB Forward_Pulse	GOSUB Forward_Pulse
ENDIF	ENDIF

- ✓ Modificate RoamingWithFotoresistenzaDividers.bs2 come mostrato nella parte destra della Figura 6-6.
- ✓ Lanciate il programma, e confrontate le prestazioni.

ESERCIZIO #3: UNA MIGLIORE REATTIVITÀ PER UN BOE-BOT CONTROLLATO DALLE OMBRE

Eliminando i cicli **FOR...NEXT** nelle subroutine di navigazione, potete rendere il Boe-Bot maggiormente reattivo. Questo non era possibile con i baffi, perché il Boe-Bot doveva retrocedere prima di curvare in quanto aveva già toccato fisicamente l'ostacolo. Quando usate le ombre per guidare il Boe-Bot, si può controllare tra un impulso e l'altro per vedere se l'ombra viene ancora rilevata non importa se si sta muovendo in avanti o se sta effettuando una manovra.

Come Funziona ShadowGuidedBoeBot.bs2

L'istruzione **IF...THEN** nel ciclo **DO...LOOP** attende per il verificarsi di una delle quattro condizioni possibili: Ambedue, Sinistra, Destra, Nessuna. In relazione a quale condizione viene rilevata, i comandi **PULSOUT** inviano impulsi per una delle seguenti manovre: Avanzamento, Rotazione Destra, Rotazione Sinistra, Oppure stà Fermo. A prescindere dalla condizione, verrà inviato uno dei quattro gruppi di impulsi ad ogni iterazione del ciclo **DO...LOOP**. Dopo l'istruzione **IF...THEN**, è importante ricordare di includere la **PAUSE 20** per assicurare il tempo di pausa tra ciascun paio di impulsi per i servo.

Il Vostro Turno– Condensare il Programma

Questo programma non necessita della condizione **ELSE** e dei due comandi **PULSOUT** che seguono. Se voi non inviate impulsi, il Boe-Bot starà fermo, proprio come se gli inviaste impulsi di **PULSOUT** con l'argomento *Duration* di 750.

- ✓ Provate a cancellare (o commentare) questo blocco di codice.

```
ELSE
    PULSOUT 13, 750
    PULSOUT 12, 750
```
- ✓ Lanciate il programma modificato.
- ✓ Potete notare qualche differenza nel comportamento del Boe-Bot?


ESERCIZIO #4: OTTENERE MAGGIORI INFORMAZIONI DALLE VOSTRE FOTORESISTENZE

La sola informazione che il BASIC Stamp è stato in grado di ottenere dal circuito del partitore con la fotoresistenza è se il livello luminoso era sopra o sotto una soglia. Questo esercizio presenta un circuito diverso che il BASIC Stamp può controllare, ottenendo anche sufficienti informazioni per determinare livelli di luminosità relativi. Il valore che il BASIC Stamp ottiene dal circuito spazierà da piccoli numeri che indicano luce brillante, a grandi numeri che indicano bassa luminosità. Questo evita la sostituzione manuale delle resistenze in base al livello di luminosità. Invece, potrete regolare il vostro programma per avviare a differenti gamme di valori.

Presentazione del Condensatore

Un condensatore è un dispositivo che immagazzina cariche elettriche, ed è un blocco fondamentale per la costruzione di molti circuiti. La quantità di carica che un condensatore è in grado di immagazzinare si misura in farad (F). Un farad è un valore

molto grande che non è pratico usare con il Boe-Bot. I condensatori che userete in questo esercizio immagazzinano frazioni di milionesimi di farad. Un milionesimo di farad si chiama microfarad, e si abbrevia con il simbolo μF . Il condensatore che userete in questo esercizio immagazzina un centesimo di un milionesimo di farad. Che è $0.01 \mu\text{F}$.



Le Capacità sono misurate Comunemente in:

- Microfarad: (milionesimi di Farad), abbreviato μF $1 \mu\text{F} = 1 \times 10^{-6} \text{ F}$
- Nanofarad: (miliardesimo di Farad), abbreviato nF $1 \text{nF} = 1 \times 10^{-9} \text{ F}$
- Picofarad: (millimiliardesimo di Farad), abbreviato pF $1 \text{pF} = 1 \times 10^{-12} \text{ F}$

Il 103 sul corpo del condensatore da $0.01 \mu\text{F}$ è la misura in picofarad o (pF). 103 sta per 10, con l'aggiunta di tre zeri, che è 10,000. ecco come si relaziona 103 a $0.01 \mu\text{F}$.

$10,000 \text{ is } 10 \times 10^3.$

$(10 \times 10^3) \times (1 \times 10^{-12}) \text{ F} = 10 \times 10^{-9} \text{ F}$

Che è anche $0.01 \times 10^{-6} \text{ F}$

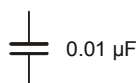
Che è $0.01 \mu\text{F}$. □

6

La Figura 6-7 mostra il simbolo schematico del condensatore da $0.01 \mu\text{F}$ ed il disegno del componente fornito nel vostro kit del Boe-Bot. La scritta 103 sul corpo del condensatore indica il suo valore.

Elenco Componenti:


- (2) Fotoresistenze - CDS
- (2) Condensatori – $0.01 \mu\text{F}$ (103)
- (2) Resistenze - 220Ω (rosso-rosso-marrone)
- (2) Jumper wires



$0.01 \mu\text{F}$



Figura 6-7
Simbolo
Schematico e
disegno del
componente



Ci possono essere anche condensatori da $0.1 \mu\text{F}$ marcati 104 nel vostro kit. Non li usate in questi esercizi.

✓ Per questo esercizio assicuratevi di aver selezionato i condensatori da $0.01 \mu\text{F}$ (marcati 103).

I condensatori da $0.1 \mu\text{F}$ possono essere usati in zone molto luminose od all'aperto, ma interferiscono con le prestazioni del Boe-Bot per le attività in condizioni di luce normale e di bassa illuminazione.

Riassemblare gli Occhi Fotosensibili

Il circuito che il BASIC Stamp può usare per determinare i livelli di luce si chiama circuito a resistenza e condensatore (RC). La Figura 6-8 mostra lo schema elettrico dei circuiti di rilevamento della luce del Boe-Bot e la Figura 6-9 mostra schemi di cablaggio esempio per la Board of Education e per la HomeWork Board.

- ✓ Scollegate l'alimentazione dalla vostra scheda e dai vostri servo.
- ✓ Assemblate i circuiti RC mostrati in Figura 6-8 usando la Figura 6-9 come riferimento.

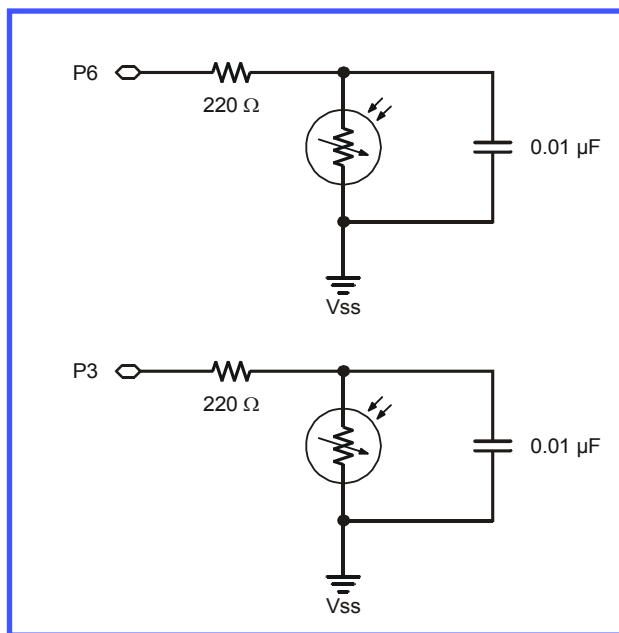


Figura 6-8
Schema Elettrico –
Due Circuiti RC a
Fotoresistenza

*Per la misura di
resistenza che varia
con la luce.*

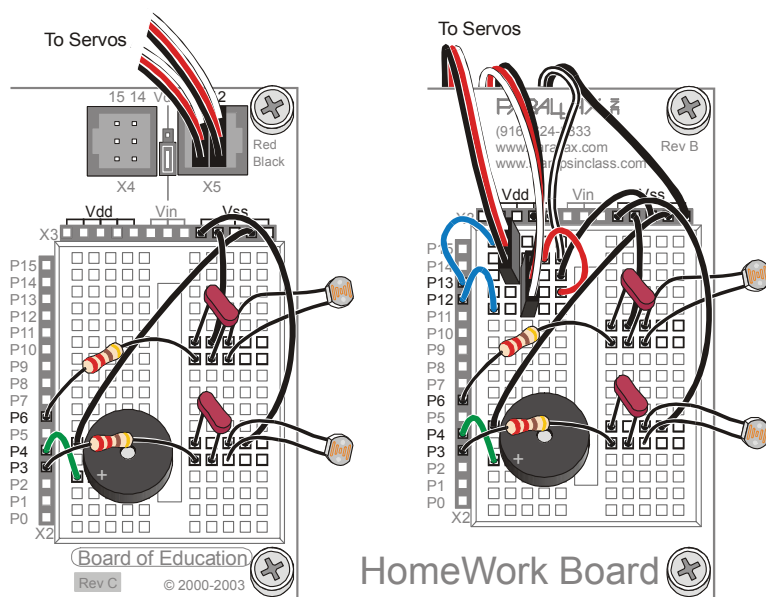


Figura 6-9
Schema di
Cablaggio per i
circuiti RC a
Fotoresistenza

*Board of
Education
(Sinistra) e
HomeWork
Board (Destra).*

6

sul Tempo di Decadimento del Circuito RC Decay e della Fotoresistenza

Pensate ad un condensatore nel circuito mostrato nella Figura 6-10 come ad una minuscola batteria ricaricabile. Quando P6 invia un segnale alto, essenzialmente stà caricando questo condensatore-batteria aplicandogli 5 V. dopo alcuni ms, il condensatore si è caricato circa a 5 V. se il programma del BASIC Stamp cambia il pin I/O in maniera che si metta in ascolto, il condensatore perde la sua carica attraverso la fotoresistenza. Man mano che il condensatore perde la sua carica, la sua tensione decade, diminuendo man mano che perde carica. La quantità di tempo impiegata perché la tensione sentita da **IN6** cada sotto la soglia di 1.4 V dipende da quanto fortemente la fotoresistenza “resiste” al flusso di corrente elettrica fornita dal condensatore. Se la fotoresistenza ha un alto valore di resistenza a causa di condizioni di poca luce, il condensatore impiega più tempo per scaricarsi. Se la fotoresistenza ha un basso valore di resistenza perché la luce incidente sulla sua superficie è molto intensa, non opporrà molta “resistenza”, ed il condensatore perderà la sua carica molto rapidamente.

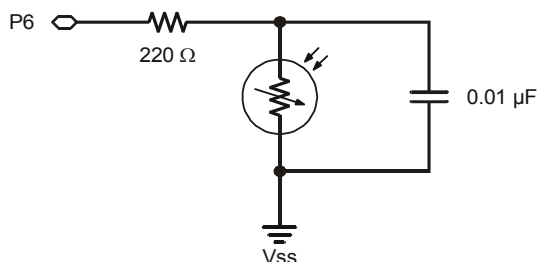


Figura 6-10
Circuito RC Collegato
al Pin I/O



Collegamento in Parallelo

La fotoresistenza ed il condensatore mostrati nella Figura 6-10 sono collegati in parallelo. Perché due componenti si dicano connessi in parallelo, ciascuno dei suoi terminali devono essere connessi a terminali comuni (anche chiamati nodi). La fotoresistenza ed il condensatore hanno uno dei terminali collegato a Vss. Essi hanno anche l'altro terminale collegato al medesimo terminale della resistenza da 220 Ω . □

Misurare il Tempo di Decadimento RC con il BASIC Stamp

Il BASIC Stamp può essere programmato per caricare il condensatore e quindi per misurare il tempo che impiegato perché la sua tensione scenda ad 1.4 V. La misura di questo tempo di decadimento può essere usata per indicare quanto è realmente intensa la luce rilevata dalla fotoresistenza. Questa misura richiede una combinazione di comandi **HIGH** e **PAUSE** insieme con un nuovo comando chiamato **RCTIME**. Il comando **RCTIME** è stato progettato per misurare il tempo di decadimento di un circuito RC in un circuito simile a quello mostrato in Figura 6-10. segue la sintassi per il comando **RCTIME**:

RCTIME Pin, State, Duration

L'argomento **Pin** è il numero del pin I/O su cui volete effettuare la misura. Per esempio, se volete fare la misura su P6, l'argomento **Pin** sarà 6. l'argomento **State** può essere sia 1 che 0. dovrebbe essere 1 se la tensione ai capi del condensatore è superiore a 1.4 V e diminuisce. Dovrebbe invece essere 0 se la tensione ai capi del condensatore è inferiore ad 1.4 V ed aumenta. Per i circuiti della Figura 6-10, la tensione ai capi del condensatore inizierà vicino ai 5 V e diminuirà verso 1.4 V, detto questo l'argomento **State** dovrà essere 1. l'argomento **Duration** deve essere una variabile dove memorizzare la misura del tempo, ed è espressa in unità di 2 μ s. In questo prossimo Programma Esempio, misurerete il tempo di decadimento RC del circuito a fotoresistenza collegato a P6, che è la fotoresistenza sinistra del Boe-Bot. Userete quindi, una variabile chiamata **timeLeft**.

Per misurare il tempo di decadimento RC, la prima cosa che dovrete fare è assicurarvi di aver dichiarato la variabile per la memorizzazione della misura del tempo:

```
timeLeft    VAR    Word
```

queste tre righe di codice caricano il condensatore, misurano il tempo di decadimento RC e quindi lo memorizzano nella variabile **timeLeft**.

```
HIGH 6
PAUSE 3
RCTIME 6,1,timeLeft
```

Per arrivare alla misura, il codice implementa questi tre passi:

1. inizia caricando il condensatore collegando il circuito a 5 V (usando il comando **HIGH**).
2. Usa **PAUSE** per dare al comando **HIGH** tempo sufficiente per la carica del condensatore del circuito RC.
3. Esegue il comando **RCTIME**, che imposta il pin /IO come ingresso, misura il tempo di decadimento (da circa 5 V ad 1.4 V), e lo memorizza nella variabile **timeLeft**.

Programma Esempio: TestP6Fotoresistenza.bs2

- ✓ Rialimentate la vostra scheda.
- ✓ Digitate, Salvate e Lanciate TestP6Fotoresistenza.bs2.
- ✓ Proiettate un'ombra sulla fotoresistenza collegata a P6 e verificate che la misura del tempo aumenti man mano che l'ambiente della fotoresistenza diviene più buio.
- ✓ Puntare la superficie sensibile della fotoresistenza verso una luce sovrastante, oppure illuminatela con una torcia elettrica. Il tempo misurato dovrebbe divenire molto piccolo. Questo tempo dovrebbe crescere gradualmente man mano che orientate la fotoresistenza lontano dalla fonte luminosa.

```
' Robotica con il Boe-Bot - TestP6Fotoresistenza.bs2
' Collauda il circuito della Fotoresistenza del Boe-Bot collegato a P6
' e visualizza il tempo di decadimento.

' {$STAMP BS2}                                ' Direttiva Stamp.
' {$PBASIC 2.5}                                ' Direttiva PBASIC.
```

```
timeLeft      VAR      Word  
  
DO  
  
    HIGH 6  
    PAUSE 2  
    RCTIME 6,1,timeLeft  
  
    DEBUG HOME, "timeLeft = ", DEC5 timeLeft  
    PAUSE 100  
  
LOOP
```

Il vostro Turno

- ✓ Salvate TestP6Fotoresistenza.bs2 con il nome TestP3Fotoresistenza.bs2.
- ✓ Modificate il programma in modo che effettui la misura del tempo di decadimento RC sulla Fotoresistenza di destra, quella collegata a P3.
- ✓ Ripetete la prova di ombreggiamento ed illuminazione con il circuito RC P3 e verificate che funzioni correttamente.

ESERCIZIO #5: UN BOE-BOT CHE SEGUE LA LUCE DI UNA TORCIA

In questo esercizio, collauderete e calibrerete i fotosensori del vostro Boe-Bot in modo che riconoscano la differenza tra la luce ambientale ed il raggio di luce diretta di una torcia. Quindi programmerete il Boe-Bot per seguire il raggio della torcia puntato sulla superficie fotosensibile sulla fronte del Boe-Bot.

Attrezzatura Aggiuntiva

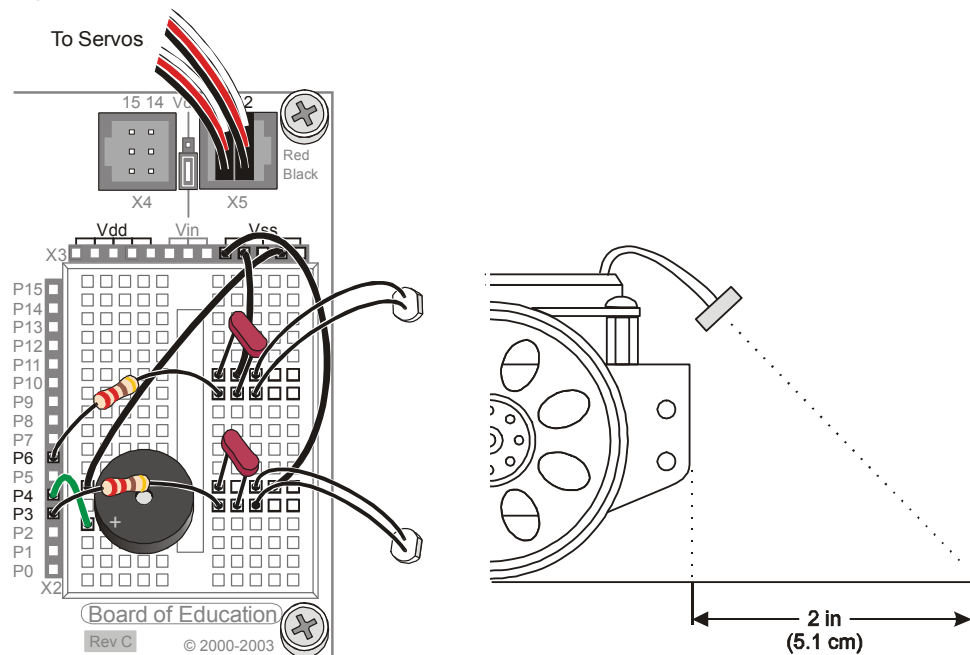
(1) Torcia Elettrica a Pile

Regolazione dei Sensori per la ricerca del raggio luminoso

Questo esercizio funziona meglio se le superfici fotosensibili delle fotoresistenze vengono posizionate per puntare verso il basso su punti separati circa 2 in (5.1 cm) in sul lato anteriore del Boe-Bot.

- ✓ Orientate la superficie fotosensibile delle vostre fotoresistenze verso il pavimento davanti al Boe-Bot come mostrato in Figura 6-11.

Figura 6-11: Orientamento delle Fotoresistenze



6

Collaudo della Risposta del Sensore al Raggio di Luce

Prima che possiate programmare il Boe-Bot per inseguire il raggio luminoso di una torcia, dovete conoscere la differenza tra le letture di luminosità del percorso del Boe-Bot con e senza la luce della torcia.

Programma Esempio: TestBothFotoresistenze.bs2

- ✓ Digitate, Salvate e Lanciate TestBothFotoresistenze.bs2.
- ✓ Posizionate il Boe-Bot sulla superficie dove dovrà seguire il raggio luminoso. Assicuratevi che sia ancora collegato con il cavo seriale e che le misure siano visualizzate sul Terminale di Debug.
- ✓ Scrivete i valori di tutte e due le misure di tempo nella prima riga della Table 6-1.
- ✓ Accendete la vostra torcia, e proiettate il fascio davanti al Boe-Bot.

- ✓ Le vostre misure dovrebbero essere significativamente inferiori rispetto alla prima coppia di misure. Scrivete questi nuovi valori di tutte e due le misure di tempo nella seconda riga della Table 6-1.

Table 6-1: Misurazioni del tempo RC con e senza il Fascio Luminoso		
Valori di Duration		Descrizione
timeLeft	timeRight	
		Misure del Tempo senza il fascio luminoso (luce ambiente).
		Misure del Tempo con il fascio luminoso proiettato davanti al Boe-Bot.

```
' Robotica con il Boe-Bot - TestBothFotoresistenze.bs2
' Collaudo dei Circuiti della Fotoresistenza del Boe-Bot RC.

' {$STAMP BS2}                                ' Direttiva Stamp.
' {$PBASIC 2.5}                                ' Direttiva PBASIC.

timeLeft      VAR      Word                    ' Dichiarazione delle Variabili.
timeRight     VAR      Word

DEBUG "VALORI DELLA FOTORESISTENZA", CR,        ' Inizializzazione.
      "timeLeft  timeRight", CR,
      "-----  -----"

DO                                                    ' Routine Principale .

  HIGH 6                                           ' Misure Del Tempo RC Sinistra.
  PAUSE 3
  RCTIME 6,1,timeLeft

  HIGH 3                                           ' Misure Del Tempo RC Destra.
  PAUSE 3
  RCTIME 3,1,timeRight

  DEBUG CRSRXY, 0, 3,                             ' Visualizzazione delle Misure.
        DEC5 timeLeft,
        "      ",
        DEC5 timeRight

  PAUSE 100

LOOP
```

Il vostro Turno

- ✓ Provate a posizionare il Boe-Bot in differenti direzioni, e ripetete le vostre misurazioni.
- ✓ Per migliori risultati, potete mediare le vostre misure con “torcia accesa” e “torcia spenta” e sostituire i valori della Table 6-1 con i valori mediati.

Sequire il Fascio di Luce

A questo punto avete già usato la dichiarazione delle variabili. Per esempio, **counter** **VAR Nib** assegna il nome **counter** ad una particolare locazione di memoria nella RAM del BASIC Stamp. Dopo aver dichiarato le variabili, ogni volta che usate **counter** in un programma PBASIC, usa il valore memorizzato in quella particolare locazione della RAM del BASIC Stamp.

6

Potete anche dichiarare le costanti. In altre parole, se avete un numero che pensate di usare nel vostro programma, assegnategli un nome adatto. Invece della direttiva **VAR**, usate la direttiva **CON**. Le seguenti sono alcune direttive **CON** tratte dal prossimo programma esempio:

```
LeftAmbient    CON    108
RightAmbient   CON    114
LeftBright     CON    20
RightBright    CON    22
```

Ora, ogni volta che nel programma viene usato il nome **LeftAmbient**, il BASIC Stamp verrà usato il numero 108. Ognivolta che è usato **RightAmbient**, il BASIC Stamp userà il valore 114. Similmente, ovunque appare **LeftBright**, in realtà è il valore 20, e **RightBright** è 22. Prima di lanciare il programma sostituite a quelli mostrati sopra, i vostri valori dalla Table 6-1.

Le costanti possono anche essere usate peer calcolare altre costanti. Il prossimo è un esempio di due costanti, chiamate **LeftThreshold** e **RightThreshold** che vengono calcolate usando le quattro costanti appena trattate. Le costanti **LeftThreshold** e **RightThreshold** sono usate nel programma per calcolare se è stato rilevato o meno il raggio di luce di una torcia.

```
'                               Mediato                Fattore di Scala

LeftThreshold  CON    LeftBright + LeftAmbient / 2    * 5 / 8
```

`RightThreshold CON RightBright + RightAmbient / 2 * 5 / 8`

Le operazioni matematiche eseguite su queste costanti sono una media, e poi una scalatura. Il calcolo della media per `LeftThreshold` è `LeftBright + LeftAmbient / 2`. Questo risultato è moltiplicato per 5 e diviso per 8. Questo significa che `LeftThreshold` è una costante il cui valore è $\frac{5}{8}$ della media di `LeftBright` e di `LeftAmbient`.



Le espressioni Matematiche in PBASIC sono eseguite da sinistra a destra. Primo, `LeftBright` è sommato a `LeftAmbient`. Questo valore poi è diviso per 2. Il risultato è quindi moltiplicato per 5 e diviso per 8.

Fate una prova: `LeftBright + LeftAmbient = 20 + 108 = 128`.

`128 / 2 = 64`.

`64 * 5 = 320`

`320 / 8 = 40`

In una riga di codice PBasic per forzare il calcolo di numeri più a destra che devono essere completati prima, potete usare le parentesi. Per esempio, potreste riscrivere questa linea di codice PBASIC:

`pulseRight = 2 - distanceRight * 35 + 750`

In questo modo:

`pulseRight = 35 * (2 - distanceRight) + 750`

In questa espressione, 35 è moltiplicato per il risultato di `(2 - distanceRight)`, il prodotto quindi è sommato a 750.

Programma Esempio: FlashlightControlledBoeBot.bs2

- ✓ Digitate FlashlightControlledBoeBot.bs2 nell'Editor del BASIC Stamp.
- ✓ Sostituite la vostra misura `timeLeft` senza fascio luminoso (dalla Table 6-1) al posto di 108 nella direttiva `CON` di `LeftAmbient`.
- ✓ Sostituite la vostra misura `timeRight` senza fascio luminoso al posto del valore 114 nella direttiva `CON` di `RightAmbient`.
- ✓ Sostituite la vostra misura `timeLeft` con il fascio luminoso al posto del valore 20 nella direttiva `CON` di `LeftBright`.
- ✓ Sostituite la vostra misura `timeRight` con il fascio luminoso al posto del valore 22 nella direttiva `CON` di `RightBright`.
- ✓ Ridate alimentazione alla vostra scheda ed ai servo.
- ✓ Salvate e quindi lanciate FlashlightControlledBoeBot.bs2.

- ✓ Sperimentate e calcolate esattamente dove dirigere il raggio luminoso della torcia per far eseguire le manovre di avanzamento, curva a sinistra, curva a destra.
- ✓ Usate il fascio luminoso per guidare il vostro Boe-Bot intorno a vari ostacoli, percorsi e manovre.

```
' -----[ Titolo ]-----
' Robotica con il Boe-Bot - FlashlightControlledBoeBot.bs2
' Il Boe-Bot segue il fascio luminoso proiettatogli davanti.

' {$STAMP BS2}                                ' Direttiva Stamp.
' {$PBASIC 2.5}                                ' Direttiva PBASIC.

DEBUG "Il Programma è in Funzione!"

' -----[ Costanti ]-----

' SOSTITUIRE QUESTI VALORI CON I VALORI DA VOI DETERMINATI ED INSERITI NELLA
' TABELLA 6.1.

LeftAmbient    CON    108
RightAmbient    CON    114
LeftBright      CON    20
RightBright     CON    22

'                               Media                Fattore di Scala
LeftThreshold   CON    LeftBright + LeftAmbient / 2 * 5 / 8
RightThreshold  CON    RightBright + RightAmbient / 2 * 5 / 8

' -----[ Variabili ]-----

' Dichiarazione delle Variabili per la memorizzazione dei valori RC misurati
' per le Fotoresistenze sinistra e destra.

timeLeft        VAR    Word
timeRight       VAR    Word

' -----[ Inizializzazione ]-----

FREQOUT 4, 2000, 3000

' -----[ Routine Principale ]-----

DO

  GOSUB Test_Fotoresistenze
  GOSUB Navigate
```

```

LOOP

' -----[ Subroutine - Test_Fotoresistenze ]-----

Test Fotoresistenze:

HIGH 6                                ' Misura del Tempo RC Sinistro.
PAUSE 3
RCTIME 6,1,timeLeft

HIGH 3                                ' Misura del Tempo RC Destro.
PAUSE 3
RCTIME 3,1,timeRight

RETURN

' -----[ Subroutine - Navigate ]-----

Navigate:

IF (timeLeft < LeftThreshold) AND (timeRight < RightThreshold) THEN
  PULSOUT 13, 850                      ' tutte e due rilevano fascio luminoso,
  PULSOUT 12, 650                      ' Marcia Avanti Massima Velocità.
ELSEIF (timeLeft < LeftThreshold) THEN ' Sinistra rileva fascio luminoso,
  PULSOUT 13, 700                      ' Rotazione Sinistra.
  PULSOUT 12, 700
ELSEIF (timeRight < RightThreshold) THEN ' Destra rileva fascio luminoso,
  PULSOUT 13, 800                      ' Rotazione Destra.
  PULSOUT 12, 800
ELSE
  PULSOUT 13, 750                      ' Nessun fascio luminoso, resta fermo.
  PULSOUT 12, 750
ENDIF

PAUSE 20                              ' Pausa tra gli impulsi.

RETURN

```

Come Funziona FlashlightControlledBoeBot.bs2

Queste sono le quattro dichiarazioni di costante che avete usato con i vostri valori della Table 6-1.

LeftAmbient	CON	108
RightAmbient	CON	114
LeftBright	CON	20
RightBright	CON	22

Ora che le quattro costanti sono state dichiarate, le prossime due righe effettuano la media e scalano i valori per ottenere i valori di soglia da usare nel programma. Questi valori di soglia possono essere confrontati con le misure correnti di **timeLeft** e **timeRight** per determinare se le fotoresistenze stanno rilevando la luce ambientale od un fascio luminoso.

```
'
                                Media                               Scala
LeftThreshold  CON             LeftBright + LeftAmbient / 2      * 5 / 8
RightThreshold CON             RightBright + RightAmbient / 2    * 5 / 8
```

Queste variabili sono usate per memorizzare le misure **RCTIME**.

```
timeLeft      VAR      Word
timeRight     VAR      Word
```

6

questo è l'indicatore di reset che è stato usato nella maggior parte dei programmi di questo testo.

```
FREQOUT 4, 2000, 3000
```

La sezione Routine Principale contiene solamente due chiamate a subroutine. Tutto il lavoro effettivo nel programma viene svolto in queste due subroutine. La subroutine **Test_Fotoresistenze** effettua la misura **RCTIME** per tutti e due i circuiti RC delle fotoresistenze, e la subroutine **Navigate** prende le decisioni ed invia gli impulsi ai servo.

```
DO

    GOSUB Test_Fotoresistenze
    GOSUB Navigate

LOOP
```

Questa è la subroutine che effettua la misura **RCTIME** per tutti e due i circuiti RC delle fotoresistenze. La misura del circuito sinistro è memorizzata nella variabile **timeLeft**, e la misura del circuito destro è memorizzata nella variabile **timeRight**.

```
Test_Fotoresistenze:

    HIGH 6
    PAUSE 3
    RCTIME 6,1,timeLeft
```

```
HIGH 3
PAUSE 3
RCTIME 3,1,timeRight

RETURN
```

La subroutine `Navigate` usa un'istruzione **IF...THEN** per confrontare la variabile `timeLeft` con la costante `LeftThreshold` e la variabile `timeRight` con la costante `RightThreshold`. Ricordate, quando la misura `RCTIME` è piccola, significa che è stata rilevata una luce intensa, e quando è grande, significa che la luce non è così intensa. In questo modo, quando una delle variabili che memorizzano una misura `RCTIME` è minore della costante di soglia, significa che il fascio luminoso è stato rilevato; all'opposto, il fascio luminoso non è stato rilevato. In base a quale condizione è rilevata da questa subroutine (ambidue, sinistra, destra o nessuna), vengono applicati gli impulsi adeguati per la navigazione, seguiti da una **PAUSE** prima che il comando **RETURN** esca dalla subroutine.

`Navigate:`

```
IF (timeLeft<LeftThreshold) AND (timeRight<RightThreshold) THEN
  PULSOUT 13, 850
  PULSOUT 12, 650
ELSEIF (timeLeft < LeftThreshold) THEN
  PULSOUT 13, 700
  PULSOUT 12, 700
ELSEIF (timeRight < RightThreshold) THEN
  PULSOUT 13, 800
  PULSOUT 12, 800
ELSE
  PULSOUT 13, 750
  PULSOUT 12, 750
ENDIF

PAUSE 20

RETURN
```

Il Vostro Turno— Calibrazione delle Prestazioni e Variazione del Comportamento

Potete calibrare le prestazioni del programma regolando il fattore di scala in questa dichiarazione di costante:

	Media	Fattore di Scala
LeftThreshold CON	LeftBright + LeftAmbient / 2	* 5 / 8
RightThreshold CON	RightBright + RightAmbient / 2	* 5 / 8

Se cambiate il fattore di scala da $\frac{5}{8}$ a $\frac{1}{2}$, questo renderà il Boe-Bot meno sensibile al fascio luminoso, il che può (oppure no) portare ad un migliore controllo del fascio luminoso.

- ✓ Provate fattori di scala differenti, come $\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{2}{3}$, e $\frac{3}{4}$ ed annotate qualsiasi differenza nel modo in cui il Boe-Bot reagiva al fascio luminoso.

Modificando l'istruzione **IF...THEN** nel Programma Esempio, potete cambiare il comportamento del Boe-Bot in modo che, per esempio, tenti di togliersi il fascio luminoso dagli occhi.

- ✓ Modificare l'istruzione **IF...THEN** in modo che il Boe-Bot retroceda quando rileva il fascio luminoso con ambedue i circuiti a fotoresistenza e curvi in direzione opposta se rileva il fascio luminoso con uno soltanto dei suoi circuiti a fotoresistenza.

6

ESERCIZIO #6: MUOVERSI VERSO LA LUCE

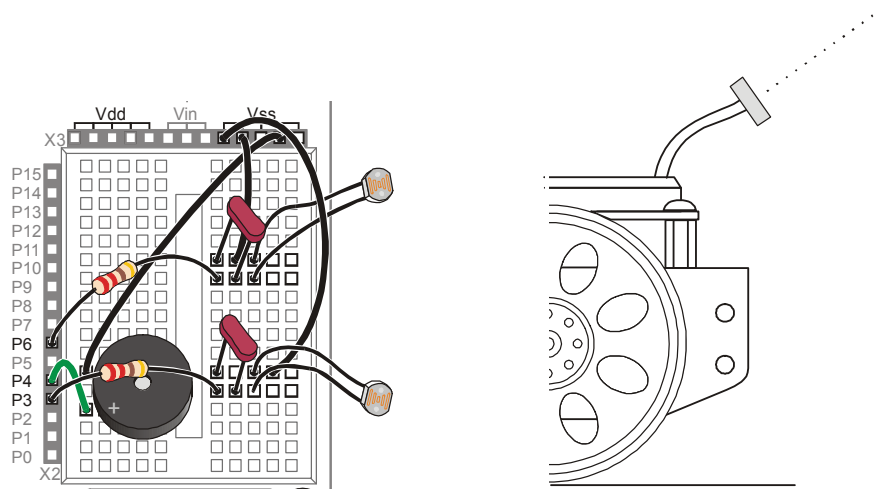
Il programma esempio in questo esercizio può essere usato per guidare il Boe-Bot da una camera sostanzialmente buia verso una porta che lasci filtrare una luminosità più intensa. Permetterà inoltre un controllo migliore del movimento del Boe-Bot proiettando un'ombra sulle fotoresistenze con la vostra mano.

Ricalibrazione delle Fotoresistenze

Questo esercizio funziona meglio se la superficie fotosensibile delle fotoresistenze punta verso l'alto e verso l'esterno.

- ✓ Orientate le superfici sensibili delle vostre fotoresistenze verso l'alto e verso l'esterno come mostrato nella Figura 6-12.

Figura 6-12: Orientamento delle Fotoresistenze



Programmazione del Comportamento per Procedere verso la Luce

La strategia chiave del movimento verso le sorgenti di luce più intensa è di andare avanti se le differenze della misura di intensità luminosa tra le fotoresistenze sono piccole, e girare verso la fotoresistenza con la misura di intensità minore quando c'è una grande differenza tra le due misure. In effetti, questo significa che il Boe-Bot avanzerà verso la luce più intensa.

Inizialmente questo sembra un compito per un programma abbastanza semplice; il ragionamento **IF...THEN** come nell'esempio sotto riportato dovrebbe funzionare. Il problema è, che non funziona perché il Boe-Bot si blocca nel curvare a sinistra ed a destra di nuovo perché le variazioni di **timeLeft** e di **timeRight** sono troppo grandi. Ogni volta che il Boe-Bot curva un pò, le variabili **timeRight** e **timeLeft** cambiano in maniera così grande che il Boe-Bot tenta di correggere e curva nell'altro verso. Non tenta mai di inviare qualche impulso in avanti.

```
IF (timeLeft > timeRight) THEN           ' Curva a Destra.
  PULSOUT 13, 850
  PULSOUT 12, 850
ELSEIF (timeRight > timeLeft) THEN       ' Curva a Sinistra.
  PULSOUT 13, 650
```

```

        PULSOUT 12, 650
    ELSE
        PULSOUT 13, 850
        PULSOUT 12, 650
    ENDIF

```

' Avanzamento.

Questo è un blocco di codice che funziona un po' meglio. Questo blocco di codice risolve il problema della curva a destra e sinistra in certe condizioni. La variabile `timeLeft` deve essere maggiore della variabile `timeRight` con un margine di 15 prima che il Boe-Bot applichi un impulso a sinistra. Similmente, `timeRight` deve essere maggiore di `timeLeft` di un fattore 15 prima che il Boe-Bot reagisca verso left. Questo dà al Boe-Bot l'opportunità di applicare abbastanza impulsi in avanti prima di correggere con una curva, ma solo se si supera un certo livello di luce.

```

    IF (timeLeft > timeRight + 15) THEN
        PULSOUT 13, 850
        PULSOUT 12, 850
    ELSEIF (timeRight > timeLeft + 15) THEN
        PULSOUT 13, 650
        PULSOUT 12, 650
    ELSE
        PULSOUT 13, 850
        PULSOUT 12, 650
    ENDIF

```

' Curva a Destra.
' Curva a Sinistra.
' Avanzamento.

Il problema con il blocco di codice su esposto è che funziona solamente in condizioni di oscurità medie. Se portate il Boe-Bot in un area molto più oscura, ricomincerà a curvare a destra ed a sinistra, e non accennerà ad avanzare. Se al contrario, lo mettete in una zona molto più luminosa, il Boe-Bot andrà solamente in avanti, e non accennerà a nessuna correzione a sinistra od a destra.

Perché accade tutto questo?

La risposta è: Quando il Boe-Bot è nella zona oscura di una stanza, la misura per ciascuna fotoresistenza sarà grande. Perché il Boe-Bot decida di curvare verso una sorgente di luce, la differenza tra queste due misure di luminosità dovrà essere grande. Quando il Boe-Bot è in una zona molto più luminosa, le misure di ogni fotoresistenza saranno minori. Perché il Boe-Bot decida di effettuare una curva, la differenza tra le due misurazioni delle fotoresistenze dovranno anch'esse essere molto più piccole di quando era nella zona più scura. Il modo di rendere questa differenza proporzionale alle condizioni di illuminazione, è di creare una variabile che sia una frazione di della media

di **timeRight** e **timeLeft**. In questo modo, sarà sempre il valore corretto, senza tener conto se la stanza è oscura o luminosa.

```
average = timeRight + timeLeft / 2
difference = average / 6
```

Ora, la variabile **difference** può essere usata in questa istruzione **IF...THEN**, e sarà un valore grande se la luminosità è bassa, ed un valore piccolo quando la luminosità è alta.

```
IF (timeLeft > timeRight + difference) THEN ' Curva a Destra.
  PULSOUT 13, 850
  PULSOUT 12, 850
ELSEIF (timeRight > timeLeft + difference) THEN ' Curva a Sinistra.
  PULSOUT 13, 650
  PULSOUT 12, 650
ELSE ' Avanzamento.
  PULSOUT 13, 850
  PULSOUT 12, 650
ENDIF
```

Programma Esempio– RoamingTowardTheLight.bs2

A differenza di **RoamingWithFotoresistenzaDividers.bs2** di pagina 196, questo programma sarà molto reattivo alla vostra mano che proietta un'ombra sulla fotoresistenza, senza influenza delle condizioni di illuminazione della stanza. In questo programma non è necessaria la sostituzione delle resistenze per adeguarsi alle condizioni di luminosità. Invece, tiene conto delle condizioni di luminosità e la sensibilità viene regolata via software usando le variabili **average** e **difference**.



Perché questo programma funzioni bene, le vostre fotoresistenze dovrebbero reagire in modo simile a livelli di luce simili. Se i circuiti RC sono grandemente disallineati, le vostre misurazioni della Table 6-1 saranno molto diverse nelle stesse condizioni di luce. Potete correggere queste misurazioni alterate usando le tecniche discusse nell' Appendice F: Bilanciamento delle Fotoresistenze.

Questo programma misura la media generale di **timeLeft** e **timeRight** ed usa questo valore per impostare la **difference** tra le misure di **timeLeft** e di **timeRight** necessarie perché venga inviato un impulso di curva.

✓ Digitate, Salvate e Lanciate **RoamingTowardTheLight.bs2**

- ✓ Posizionate il Boe-Bot in varie zone, lasciatelo girovagare, e verificate che potete cambiare il suo percorso proiettando l'ombra su uno dei circuiti RC a fotoresistenza, senza influenze da parte delle condizioni di illuminazione della stanza.
- ✓ Provate anche a mettere il vostro Boe-Bot in una stanza che sia fiocamente illuminata, ma che abbia una porta che faccia entrare luce da una stanza maggiormente illuminata. Vedere se il Boe-Bot riesce con successo a trovare la sua strada verso la porta.

```
' -----[ Titolo ]-----
' Robotica con il Boe-Bot - RoamingTowardTheLight.bs2
' Il Boe-Bot vagabonda, evitando le aree scure preferendo zone più luminose.

' {$STAMP BS2}                                ' Direttiva Stamp.
' {$PBASIC 2.5}                                ' Direttiva PBASIC.

DEBUG "Il Programma è in Funzione!"

' -----[ Variabili ]-----

' Dichiarazione delle Variabili per la memorizzazione delle misure di tempo RC
' delle Fotoresistenze Sinistra & Destra.

timeLeft      VAR    Word
timeRight     VAR    Word
average       VAR    Word
difference    VAR    Word

' -----[ Inizializzazione ]-----

FREQOUT 4, 2000, 3000

' -----[ Routine Principale ]-----

DO

  GOSUB Test Fotoresistenze
  GOSUB Average_And_Difference
  GOSUB Navigate

LOOP

' -----[ Subroutine - Test Fotoresistenze ]-----

Test Fotoresistenze:

  HIGH 6                                ' Misura del Tempo RC di Sinistra.
  PAUSE 3
```

```

RCTIME 6,1,timeLeft

HIGH 3                                ' Misura del Tempo RC di Destra.
PAUSE 3
RCTIME 3,1,timeRight

RETURN

' ----[ Subroutine - Average_And_Difference ]-----
Average_And_Difference:

    average = timeRight + timeLeft / 2
    difference = average / 6

    RETURN

' ----[ Subroutine - Navigate ]-----
Navigate:

    ' Shadow significantly stronger on left detector, turn right.
    IF (timeLeft > timeRight + difference) THEN
        PULSOUT 13, 850
        PULSOUT 12, 850
    ' Shadow significantly stronger on right detector, turn left.
    ELSEIF (timeRight > timeLeft + difference) THEN
        PULSOUT 13, 650
        PULSOUT 12, 650
    ' Shadows in same neighborhood of intensity on both detectors.
    ELSE
        PULSOUT 13, 850
        PULSOUT 12, 650
    ENDIF

    PAUSE 10

    RETURN

```



Perché PAUSE 10 invece di PAUSE 20? Perché la subroutine **Test_Fotoresistenze** ha due comandi **PAUSE** che aggiungono 6 ms di tempo extra per l'esecuzione dei comandi **RCTIME**. Ambedue questi fattori aumentano il tempo tra gli impulsi servo, per questo la **PAUSE** nella subroutine **Navigate** deve essere ridotta. Dopo alcuni esperimenti pratici, è sembrato che **PAUSE 10** desse ai servo le prestazioni più attendibili per un'ampia gamma di livelli di luminosità ambientale.

Il Vostro Turno—Regolare la Sensibilità alle Differenze della Luminosità

Fino ad ora, la variabile **difference** è la media **average** divisa per 6. Potete dividere **average** per un valore inferiore se volete che il Boe-Bot sia meno sensibile alle differenze di luminosità o dividerla per un valore maggiore se volete che il Boe-Bot sia più sensibile alle differenze dei livelli di luce.

- ✓ Invece del valore 6, provate a dividere la variabile **average** per i valori 3, 4, 5, 7, e 9.
- ✓ Lanciate il programma e provate l'abilità del Boe-Bot di uscire da una stanza oscurata con ciascun valore per il denominatore.
- ✓ Decidete quale valore è quello ottimale.

6

```
Average_And_Difference:
```

```
average = timeRight + timeLeft / 2
difference = average / 6
```

```
RETURN
```

Potete anche dichiarare il denominatore con una costante come in questo caso:

```
Denominator CON 6
```

Quindi, nella subroutine **Average_And_Difference**, potete sostituire 6 (od il valore ottimale che avete determinato) con la costante **Denominator**, in questo modo:

```
Average_And_Difference:
```

```
average = timeRight + timeLeft / 2
difference = average / Denominator
```

```
RETURN
```

- ✓ Digitate i cambiamenti appena discussi, e verificate che il programma funzioni ancora correttamente.

In questo programma potete anche fare a meno di una variabile. Notate che la sola volta che la variabile **average** viene usata è per memorizzare temporaneamente il valore medio, quindi lo divide per il **Denominator** e lo memorizza nella variabile **difference**. La variabile **difference** vi servirà in seguito, non così per la variabile **average**. Un

modo per risolvere questo problema sarebbe di usare semplicemente la variabile **difference** al posto della variabile **average**. Funzionerà a meraviglia, e non avreste più bisogno della variabile **average**. Qui viene mostrato come funziona la subroutine:

```
Average_And_Difference:

    difference = timeRight + timeLeft / 2
    difference = difference / Denominator

RETURN
```

Ma c'è un modo migliore.

✓ Lasciate la routine **Average_And_Difference** in questo modo:

```
Average_And_Difference:

    average = timeRight + timeLeft / 2
    difference = average / Denominator

RETURN
```

✓ Poi, digitate queste variazioni nella dichiarazioni di variabili:

Figura 6-13: Modificate RoamingTowardTheLight.bs2 per salvare una Word di RAM

' Codice non cambiato			' Cambiato per Salvare 1 Word di RAM		
average	VAR	Word	average	VAR	Word
difference	VAR	Word	difference	VAR	average

La variabile **average** in realtà non serve, ma il programma avrà più senso per qualcuno che tenti di capirlo se usiamo la parola **average** nella prima riga e la parola **difference** nella seconda riga. Questo è il modo di creare un alias di nome **difference** per la variabile **average**.

```
difference    VAR    average
```

Ora, ambedue **average** e **difference** si riferiscono alla stessa word di RAM.

- ✓ Collaudate il vostro programma modificato ed assicuratevi che funzioni ancora correttamente.