

# LE VARIABILI E IL CICLO FOR

*Prosegue in questo fascicolo lo studio del linguaggio di programmazione RoboBasic, con due nuovi esempi di codice per il ragno RoboSpider.*

**N**elle prossime pagine sono presentati due nuovi esempi in RoboBasic per il ragno robotico (troverai il codice nel quarto CD-Rom, allegato al prossimo fascicolo). Il primo esempio (**BackwardWalk.bas**) è simile a quello presentato nell'uscita precedente, con la differenza che il ragno si muove all'indietro. Anche la struttura del codice sorgente è sostanzialmente invariata. Il secondo esempio (**RotateLeft90.bas**), invece, permette a RoboSpider di ruotare su se stesso di circa 90° in direzione antioraria.

## VARIABILI E CICLI >>>

In quasi tutti i linguaggi di programmazione esiste uno strumento molto importante tramite il quale è possibile gestire in maniera efficace i dati (ad esempio informazioni di tipo numerico): le **variabili**. Si tratta, in sostanza, di 'contenitori' utilizzati per immagazzinare dei dati durante l'esecuzione di un programma. Una variabile può essere usata, ad esempio, per memorizzare un valore numerico, come un parametro misurato da un sensore. Anche in RoboBasic è possibile utilizzare le variabili, ed esistono

alcune istruzioni adatte a tale scopo. Senza entrare nei dettagli, è sufficiente sapere che in RoboBasic esistono due diverse tipologie di variabili, in base al tipo di dati che devono immagazzinare: **BYTE** o **INTEGER**. Un **BYTE** è un numero intero con un valore compreso tra 0 e 255, mentre un **INTEGER** è un numero intero di valore compreso tra 0 e 65535. La scelta su quale tipo utilizzare, ovviamente, è dettata dalle esigenze specifiche di un programma. Prima di utilizzare una variabile in RoboBasic, è necessario 'dichiararla' tramite l'istruzione **DIM**: in questo modo si associa un identificatore (che può essere una lettera o più generalmente una stringa alfanumerica) a un insieme di valori possibili (**BYTE** o **INTEGER**). Una volta dichiarata, la variabile può essere utilizzata all'interno del programma: possiamo assegnarle un valore e modificarlo quando necessario. Nel codice di **RotateLeft90.bas**, incontriamo un importante costrutto: il ciclo **FOR**. Come nel caso del ciclo **GOTO**, incontrato nel fascicolo 49, questo nuovo ciclo permette di iterare una sequenza di istruzioni: a differenza del ciclo



## COMPONENTI

- 1) motore elettrico del decimo servomotore (6N200)

**GOTO**, però, con il ciclo **FOR** possiamo indicare precisamente il numero di iterazioni volute. A pagina 10 è presente un box dettagliato che spiega la sintassi delle nuove istruzioni presentate.

#### CODICE ROBOBASIC: BACKWARDWALK>>>

Dopo aver presentato nello scorso fascicolo un esempio di codice per far muovere in avanti RoboSpider, l'esempio sottostante (**BackwardWalk.bas**) opera in maniera opposta: il ragno si muove in direzione opposta, ossia all'indietro. La struttura del codice è del tutto analoga all'esempio **ForwardWalk.bas**.

```

\=====
\ RoboSpider - BackwardWalk.bas
\=====

\ Settaggio dei servomotori
PTP ALLON
DIR G6A,1,1,1,0,1,1
DIR G6D,0,0,0,1,0,0
GETMOTORSET G6A,1,1,1,1,0,0
GETMOTORSET G6D,1,1,1,1,0,0
MOTOR G6A
MOTOR G6D
SPEED 10

\ Posizione di riposo
MOVE G6A,55,55,55,55,,
MOVE G6D,55,55,55,55,,

backward_walk:
\ Prima posizione
MOVE G6A,90,80,55,80,,
MOVE G6D,55,30,90,30,,

\ Seconda posizione
MOVE G6A,55,80,90,80,,
MOVE G6D,90,80,55,30,,

\ Terza posizione
MOVE G6A,90,30,55,30,,
MOVE G6D,90,80,55,80,,
GOTO backward_walk

```

#### CODICE ROBOBASIC: ROTATELEFT90>>>

In questo esempio (**RotateLeft90.bas**) si utilizzano per la prima volta le variabili e il ciclo **FOR**. Proprio in questo ciclo si fa uso della variabile **i**, dichiarata per mezzo dell'istruzione **DIM**. Il blocco di istruzioni di movimento contenuto nel ciclo viene ripetuto tre volte. Al termine della rotazione RoboSpider viene riportato nella posizione di riposo.

```

\=====
\ RoboSpider - RotateLeft90.bas
\=====

\ dichiarazione della variabile i
DIM i AS BYTE
\ Settaggio dei servomotori
PTP ALLON
DIR G6A,1,1,1,0,1,1
DIR G6D,0,0,0,1,0,0
GETMOTORSET G6A,1,1,1,1,0,0
GETMOTORSET G6D,1,1,1,1,0,0
MOTOR G6A
MOTOR G6D
SPEED 10

\ Posizione di riposo
MOVE G6A,55,55,55,55,,
MOVE G6D,55,55,55,55,,

FOR i = 1 TO 3
\ Prima posizione
MOVE G6A,145,55,55,55,,
MOVE G6D,55,55,145,55,,
\ Seconda posizione
MOVE G6A,145,110,55,55,,
MOVE G6D,55,55,145,110,,
\ Terza posizione
MOVE G6A,55,110,145,55,,
MOVE G6D,145,55,55,110,,
\ Quarta posizione
MOVE G6A,55,55,145,55,,
MOVE G6D,145,55,55,55,,
NEXT i

\ Posizione di riposo
MOVE G6A,55,55,55,55,,
MOVE G6D,55,55,55,55,,

```

## ISTRUZIONI ROBOBASIC

L'esempio `RotateLeft90.bas` introduce alcune nuove istruzioni per la gestione delle variabili e dei cicli. Vediamo in dettaglio come utilizzarle correttamente.

### DIM x AS type

Prima di poter utilizzare una variabile in RoboBasic è necessario dichiararla, ossia specificare un identificatore che la caratterizzi e denoti la classe di valori che può assumere. Il parametro *x* denota l'identificatore (che può essere una lettera o, più generalmente una stringa, alfanumerica), mentre il parametro *type* (che può essere **BYTE** o **INTEGER**) specifica la classe di valori. Una variabile di tipo **BYTE** può assumere valori interi compresi tra 0 e 255, mentre una variabile di tipo **INTEGER** può assumere valori interi compresi tra 0 e 65535.

#### ▶ Esempi:

`DIM k AS BYTE` - La variabile 'k' viene dichiarata come **BYTE**.

`DIM sensore AS INTEGER` - La variabile 'sensore' viene dichiarata come **INTEGER**.

### FOR i = v1 TO v2 [blocco di istruzioni] NEXT i

Il ciclo **FOR** è un costrutto di programmazione che permette di iterare una sequenza di istruzioni per un numero di volte stabilito. Si compone di due parti, tra le quali viene inserita la sequenza di istruzioni. Il parametro *i* è una variabile (deve essere stata dichiarata in precedenza) che serve per tenere il conteggio delle iterazioni. I parametri *v1* e *v2* denotano, rispettivamente, il valore iniziale e finale della variabile contatore. All'inizio del ciclo la variabile contatore viene inizializzata con il valore *v1*, quindi l'iterazione ha inizio: viene eseguito il blocco di istruzioni e, una volta raggiunta l'istruzione **NEXT**, il valore della variabile contatore viene incrementato di 1 e il blocco di istruzioni contenuto nel ciclo viene eseguito nuovamente. Il processo si ripete, fino a quando il valore della variabile contatore raggiunge il valore indicato dal parametro *v2*.

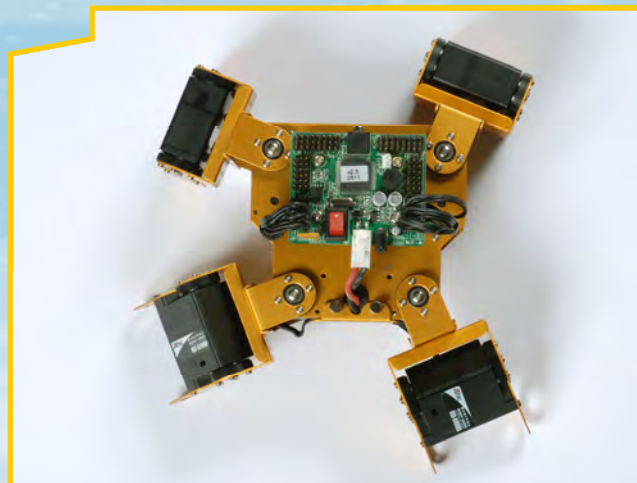
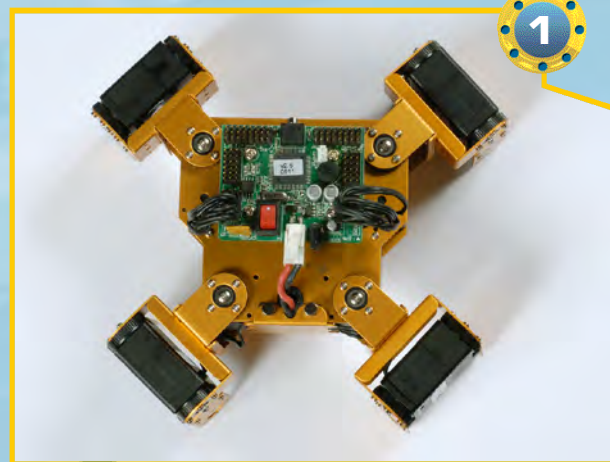
#### ▶ Esempio:

```
FOR B = 1 TO 10
  DELAY 1000
NEXT B
```

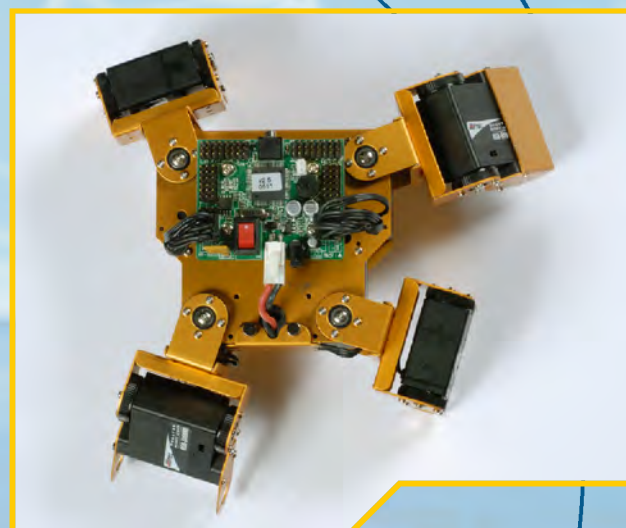
- L'istruzione 'DELAY 1000' viene eseguita 10 volte. In questo esempio la variabile *B* funge da contatore. Il suo valore iniziale è impostato a 1, quello finale a 10.

◀1▶ Vediamo insieme le posizioni che RoboSpider assume durante l'esecuzione del programma 'BackwardWalk.bas'. Il ragno robotico si pone inizialmente nella posizione di riposo, quindi vengono ripetute ciclicamente tre coppie di movimenti. Il codice di questo esempio sarà presente sul quarto CD-Rom; è comunque possibile creare un nuovo programma in RoboBasic ed editare il codice sorgente manualmente.

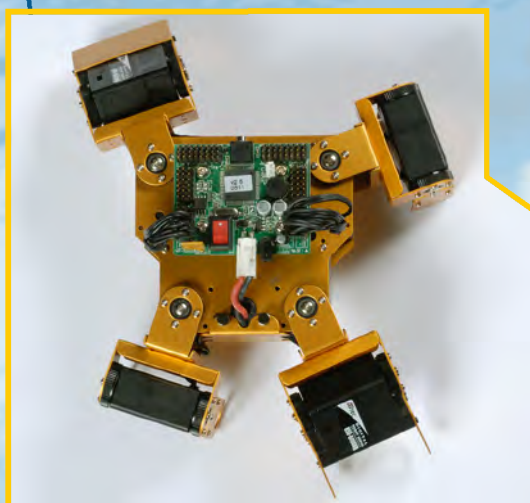
Posizione di riposo:  
 MOVE G6A, 55, 55, 55, 55, ,  
 MOVE G6D, 55, 55, 55, 55, ,



Terza posizione:  
 MOVE G6A, 90, 30, 55, 30, ,  
 MOVE G6D, 90, 80, 55, 80, ,



Prima posizione:  
 MOVE G6A, 90, 80, 55, 80, ,  
 MOVE G6D, 55, 30, 90, 30, ,

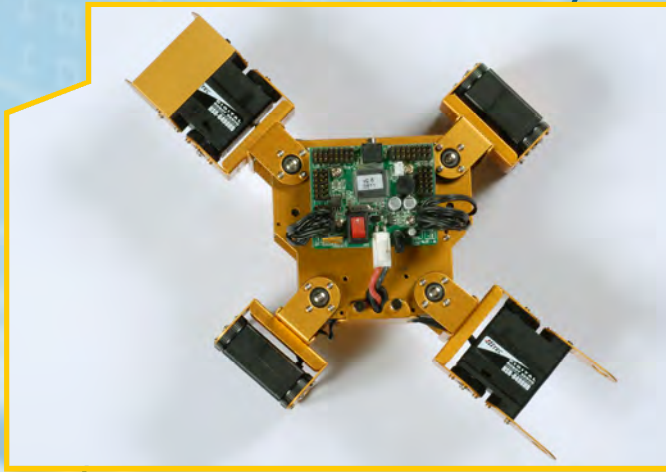
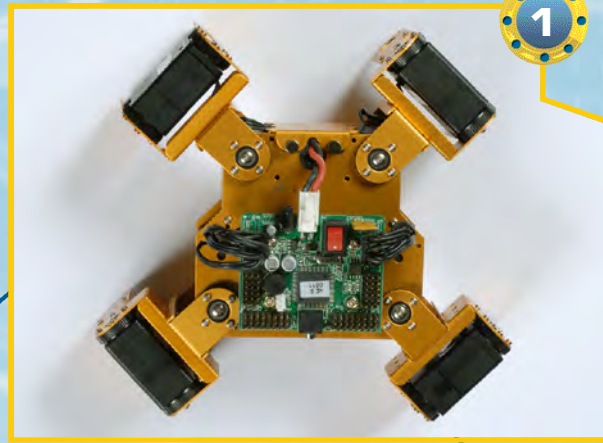


Seconda posizione:  
 MOVE G6A, 55, 80, 90, 80, ,  
 MOVE G6D, 90, 80, 55, 30, ,

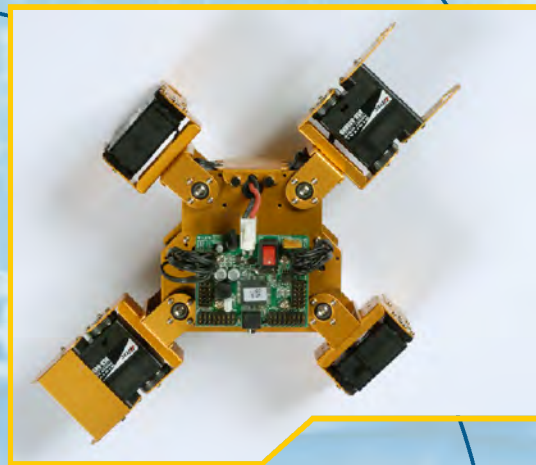


◀1▶ Nell'esempio 'RotateLeft90.bas', RoboSpider si pone nella posizione di riposo, quindi esegue per 3 volte la sequenza di istruzioni che gli permette di ruotare di 90°. Infine torna nella posizione di riposo.

Posizione di riposo:  
MOVE G6A, 55, 55, 55, 55, ,  
MOVE G6D, 55, 55, 55, 55, ,

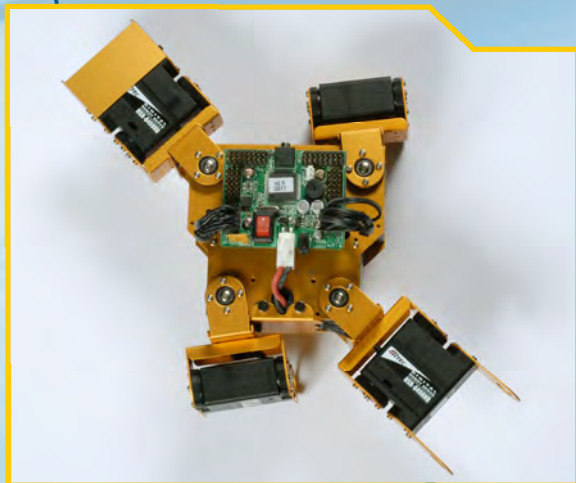


Quarta posizione:  
MOVE G6A, 55, 55, 145, 55, ,  
MOVE G6D, 145, 55, 55, 55, ,

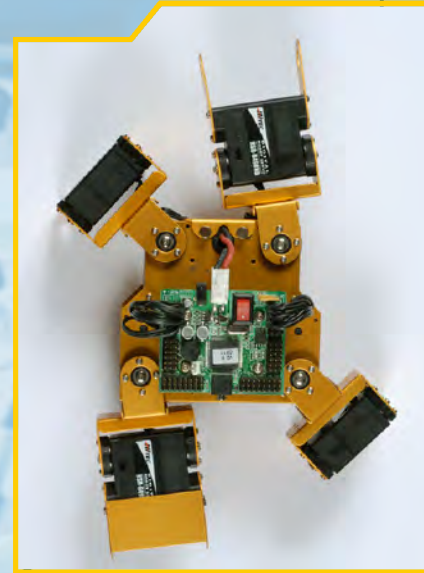


Prima posizione:  
MOVE G6A, 145, 55, 55, 55, ,  
MOVE G6D, 55, 55, 145, 55, ,

x3



Terza posizione:  
MOVE G6A, 55, 110, 145, 55, ,  
MOVE G6D, 145, 55, 55, 110, ,



Seconda posizione:  
MOVE G6A, 145, 110, 55, 55, ,  
MOVE G6D, 55, 55, 145, 110, ,

# RIEPILOGO COMPONENTI

*In questo elenco trovi tutte le tipologie di pezzi che ti sono state fornite a partire dal primo fascicolo: puoi consultarlo quando devi affrontare le fasi di montaggio, in modo da avere un riferimento immediato per i componenti che dovrai utilizzare e per quelli che hai a disposizione.*

- ▶ armatura del dorso
- ▶ armatura del torace
- ▶ base inferiore per servo A
- ▶ base inferiore per servo B
- ▶ base inferiore per servo C
- ▶ base superiore per servo A
- ▶ base superiore per servo B
- ▶ base superiore per servo C
- ▶ bullone da 3x4 mm
- ▶ caricabatterie
- ▶ cavo di prolunga per pacco batterie
- ▶ cavo seriale
- ▶ circuito con LED
- ▶ coperchio in metallo per il vano batterie
- ▶ copertura in plastica del piede sinistro e destro
- ▶ cuscinetto a sfera
- ▶ distanziatore da 3x5 mm
- ▶ elementi plastici della mano
- ▶ fascetta di fissaggio dei cavi
- ▶ fascetta in plastica per il raggruppamento dei cavi
- ▶ guaina in plastica proteggi cavo
- ▶ intelaiatura metallica del dorso
- ▶ intelaiatura metallica del piede
- ▶ intelaiatura metallica del polso
- ▶ intelaiatura metallica del torace
- ▶ motore elettrico cavo 200 mm (6N200 - Servo C)
- ▶ motore elettrico cavo 300 mm (4N300 - Servo A)
- ▶ motore elettrico cavo 400 mm (5N400 - Servo B)
- ▶ nastro biadesivo
- ▶ pacco batterie ricaricabili
- ▶ parte anteriore della testa
- ▶ parte posteriore della testa
- ▶ perno da 1,6x14 mm
- ▶ perno da 1,6x9 mm
- ▶ protezione per scheda MR-C3024
- ▶ rondella da 6x2,2x0,5 mm
- ▶ rondella da 7,6x2,8x0,5 mm
- ▶ ruota dentata di tipo 1
- ▶ ruota dentata di tipo 2
- ▶ ruota dentata di tipo 3
- ▶ ruota dentata di tipo 4
- ▶ scheda MR-C3024
- ▶ scheda PC Servo Control
- ▶ sensore di contatto
- ▶ sensore di luce
- ▶ sostegno per potenziometro
- ▶ squadretta circolare di tipo 1
- ▶ squadretta circolare di tipo 2
- ▶ squadretta circolare di tipo 3
- ▶ squadretta circolare di tipo 4
- ▶ squadretta circolare per il fissaggio della testa
- ▶ squadretta metallica a I
- ▶ squadretta metallica a U (16 fori)
- ▶ squadretta metallica a U (22 fori)
- ▶ squadretta metallica ad H
- ▶ tubetto di grasso
- ▶ visiera
- ▶ vite di tipo M da 2,6x4 mm
- ▶ vite di tipo M da 2x4 mm
- ▶ vite di tipo M da 3x4 mm
- ▶ vite di tipo T-2 da 2,6x6 mm
- ▶ vite di tipo T-2 da 2x12 mm
- ▶ vite di tipo T-2 da 2x18 mm
- ▶ vite di tipo T-2 da 2x21 mm (nera)
- ▶ vite di tipo T-2 da 2x26 mm (nera)
- ▶ vite di tipo T-2 da 2x4 mm
- ▶ vite di tipo T-2 da 2x5 mm
- ▶ vite di tipo T-2 da 2x8 mm

