

# UN SEMPLICE PROGRAMMA

*In questo numero vedrai come creare un semplice programma per muovere il braccio robotico RoboArm partendo da zero, grazie agli strumenti messi a disposizione dal software RoboScript.*

**N**ello scorso fascicolo hai iniziato a prendere confidenza con la programmazione in **RoboScript**, testando l'esempio per RoboArm contenuto nel secondo CD-Rom (allegato all'uscita 19). In questo nuovo fascicolo, invece, ti sarà mostrato come creare un breve programma partendo da zero, grazie alle istruzioni passo a passo che trovi nelle prossime pagine. Come al solito, prima di creare e testare questo nuovo

esempio, dovrai collegare il pacco batterie e i servomotori di RoboArm alla scheda **PC Servo Control I**, quindi connettere quest'ultima al PC utilizzando l'apposito cavo seriale. Per ulteriori dettagli circa queste operazioni fai riferimento a quanto già spiegato nei fascicoli precedenti (19 e 21). Creare un programma in RoboScript è molto semplice: è sufficiente fare assumere a RoboArm la configurazione desiderata,

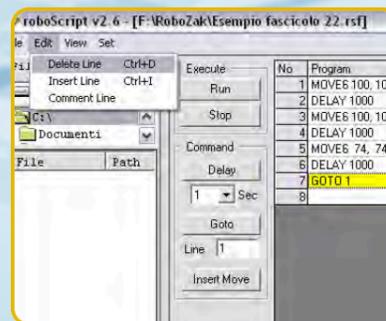
usando gli speciali cursori messi a disposizione dall'applicazione per muovere i tre servomotori, quindi memorizzare la posizione cliccando sul pulsante **'Insert Move'**. In questo esempio utilizzerai anche una nuova istruzione messa a disposizione dal software, chiamata **GOTO**. Questo speciale comando permette di 'saltare' da un punto a un altro del codice, specificando come parametro il numero di riga al quale si vuole 'passare'. Ogni riga di codice RoboScript, infatti, è contrassegnata da un numero intero crescente: la prima dal numero 1, la seconda dal numero 2 e così via. Se durante la programmazione sbagli a

⚡ *Se desideri eliminare una riga di codice è sufficiente cliccare sopra di essa e selezionare la voce 'Delete line' dal menu Edit.*



## COMPONENTI

- ◀1▶ 4 fascette in plastica per il raggruppamento dei cavi
- ◀2▶ nastro biadesivo
- ◀3▶ sensore di luce
- ◀4▶ 8 viti tipo T-2 da 2x4 mm



**WARNING**

In questo nuovo esempio di programmazione in RoboScript muoveremo anche il servomotore con identificativo numerico 0, ossia quello posto nella base di appoggio di RoboArm. A causa della struttura del braccio robotico, il range



di movimento di questo servo è molto ridotto. Pertanto, quando realizzerai i tuoi programmi, evita di impostare con l'istruzione **MOVE** delle posizioni angolari non raggiungibili dai servo (per ulteriori dettagli circa i parametri di funzionamento dell'istruzione **MOVE** e le posizioni angolari dei servo consulta il fascicolo numero 21, pagina 7). Quando RoboArm è in movimento, fai attenzione a non posizionare le tue dita vicino ai giunti mobili: la coppia dei servomotori è infatti molto elevata (oltre 7 kg/cm) e potresti ferirti.

inserire un'istruzione, puoi cancellare la riga di codice selezionando la voce **'Delete line'** dal menu **Edit**. Per testare l'esempio di programmazione, non dovrai cambiare gli **ID numerici** dei servomotori ma lasciarli settati come abbiamo specificato nel fascicolo numero 21. Dopo aver creato il codice, potrai salvarlo sul tuo personal computer e caricarlo in seguito. In questo fascicolo hai trovato altri nuovi elementi

Un dettaglio del sensore di luce. Potrai utilizzarlo quando riceverai la scheda di controllo MR-C3024.

che dovrai utilizzare più avanti per proseguire la costruzione delle nuove configurazioni robotiche. Tra questi vi è il **sensore di luce**, un sofisticato

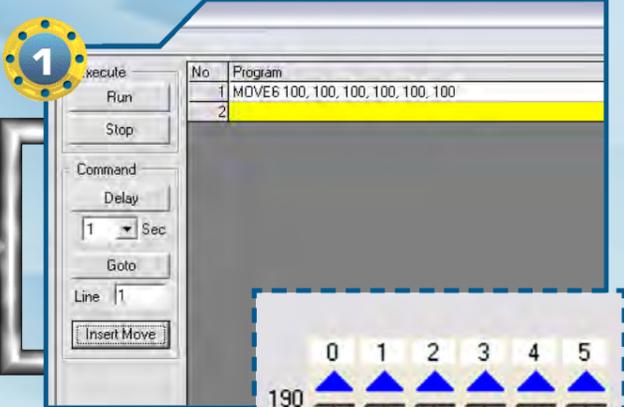
componente elettronico che potrai utilizzare una volta che avrai ricevuto la scheda madre MR-C3024 e con il quale potrai sviluppare nuovi programmi.

**ESEMPIO DI PROGRAMMAZIONE>>>**

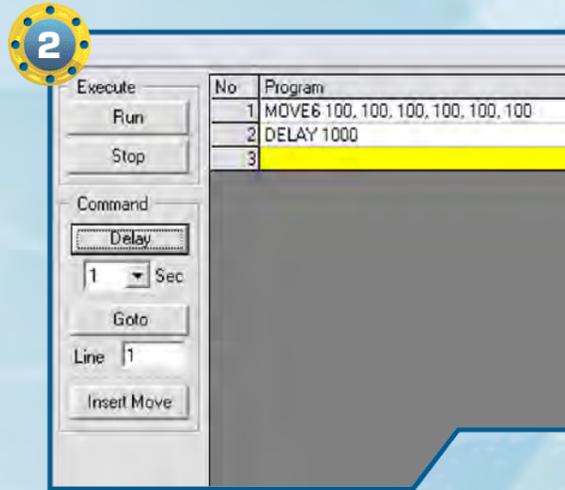
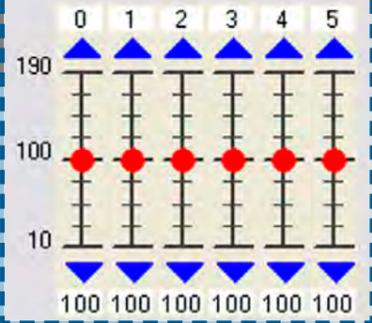
Questo nuovo esempio di programmazione fa assumere a RoboArm tre diverse posizioni, specificate dalle tre istruzioni **MOVE** (di cui ci siamo occupati in dettaglio nel fascicolo 21). Anche in questo caso dopo ogni istruzione **MOVE** è presente un'istruzione **DELAY**, il cui scopo è quello di sospendere temporaneamente l'esecuzione del programma per consentire ai servomotori di completare il comando di movimento. L'ultima istruzione, **GOTO**, permette di 'saltare' alla riga di codice specificata come parametro (in questo caso 1). Pertanto, dopo l'esecuzione del comando **GOTO 1**, la sequenza di movimento riparte dall'inizio (l'istruzione corrispondente alla prima riga di codice, infatti, è **MOVE 100, 100, 100, 100, 100, 100**). Poiché in questo modo il programma prosegue ciclicamente, per interrompere la sua esecuzione è necessario cliccare sul pulsante **'Stop'** presente nell'interfaccia della finestra del software RoboScript.

```
MOVE6 100, 100, 100, 100, 100, 100
DELAY 1000
MOVE6 100, 100, 10, 100, 100, 100
DELAY 1000
MOVE6 74, 74, 10, 100, 100, 100
DELAY 1000
GOTO 1
```

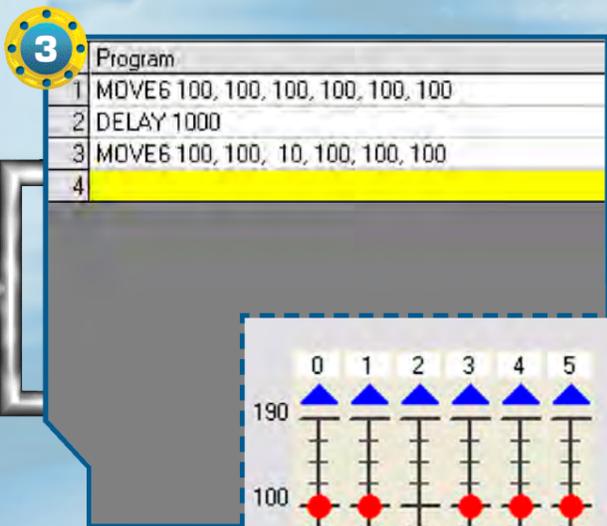




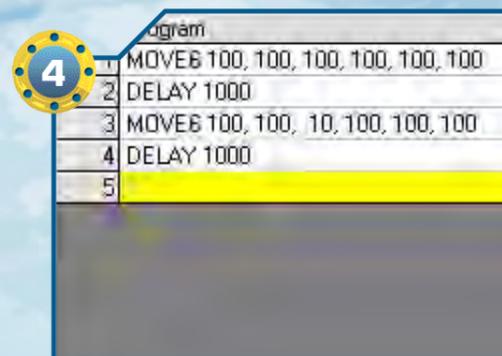
1 Dopo aver effettuato tutti i cablaggi e aver avviato l'applicazione RoboScript, inseriamo la prima istruzione MOVE. Per fare questo dobbiamo posizionare i controlli dei servomotori di RoboArm come mostrato nel riquadro tratteggiato (100, 100, 100, 100, 100, 100), quindi cliccare sul pulsante 'Insert Move' per memorizzare la posizione.



2 Inseriamo ora un'istruzione DELAY, cliccando sul pulsante Delay. Come parametro manteniamo l'impostazione di default, ossia un secondo.



3 Impostiamo i controlli dei servomotori come in figura (100, 100, 10, 100, 100, 100) e clicchiamo nuovamente sul pulsante 'Insert Move' per inserire la nuova posizione.

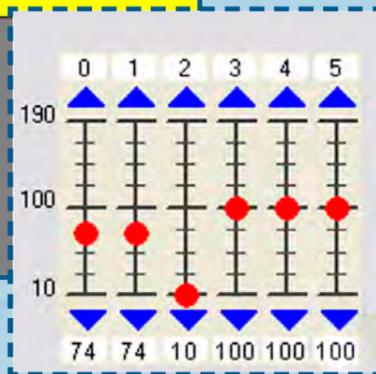


4 Clicchiamo sul pulsante Delay per inserire un nuovo 'ritardo' nel nostro programma RoboScript.

**5**

No	Program
1	MOVE6 100, 100, 100, 100, 100, 100
2	DELAY 1000
3	MOVE6 100, 100, 10, 100, 100, 100
4	DELAY 1000
5	MOVE6 74, 74, 10, 100, 100, 100
6	

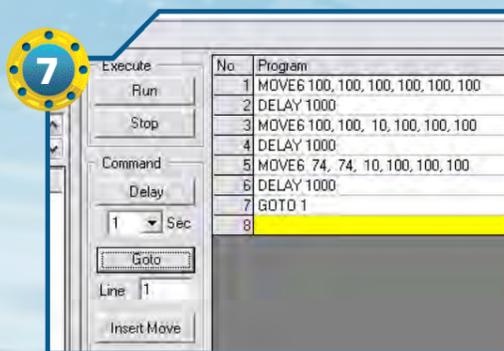
«5» Dobbiamo ora inserire la terza e ultima posizione. Dopo aver posto i controlli dei servomotori come mostrato nel riquadro (74, 74, 10, 100, 100, 100), clicchiamo sul pulsante 'Insert Move'.



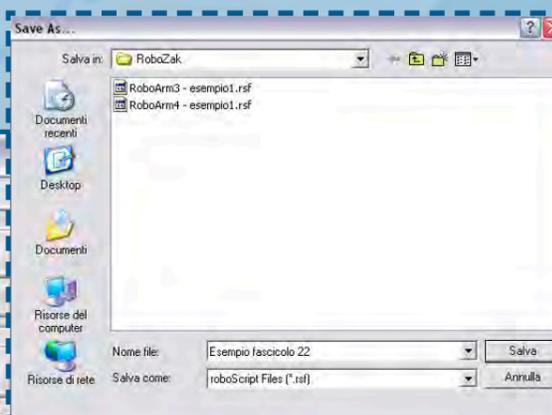
**6**

No	Program
1	MOVE6 100, 100, 100, 100, 100, 100
2	DELAY 1000
3	MOVE6 100, 100, 10, 100, 100, 100
4	DELAY 1000
5	MOVE6 74, 74, 10, 100, 100, 100
6	DELAY 1000
7	

«6» Inseriamo un nuovo ritardo, come già visto in precedenza negli step 2 e 4.



«7» Non ci resta che inserire l'ultima istruzione, cliccando sul pulsante Goto. Nella finestra del codice apparirà il comando GOTO 1.



«8» Possiamo salvare il programma appena creato, in modo da poterlo caricare nuovamente in futuro. Dal menu File selezioniamo la voce 'Save Program File'. Comparirà una nuova finestra, come quella

mostrata nel riquadro, in cui dovremo inserire il nome con cui salvare il programma. Una volta scelto il nome con cui memorizzare il nostro esempio, confermiamo l'operazione di salvataggio cliccando sul pulsante 'Salva'.

9

Execute	No	Program
Run	1	MOVE6 100, 100, 100, 100, 100, 100
Stop	2	DELAY 1000
	3	MOVE6 100, 100, 10, 100, 100, 100
	4	DELAY 1000
Command	5	MOVE6 74, 74, 10, 100, 100, 100
Delay	6	DELAY 1000
1 Sec	7	GOTO 1
Goto	8	
Line 1		

«9» A questo punto possiamo finalmente testare il programma, cliccando sul pulsante 'Run'. Per interrompere l'esecuzione del codice, clicchiamo su Stop.



«10» In queste tre immagini è mostrata la sequenza di movimenti (dall'immagine a sinistra, in senso orario) di RoboArm durante l'esecuzione del programma.

