

# Linguaggi di programmazione testuale

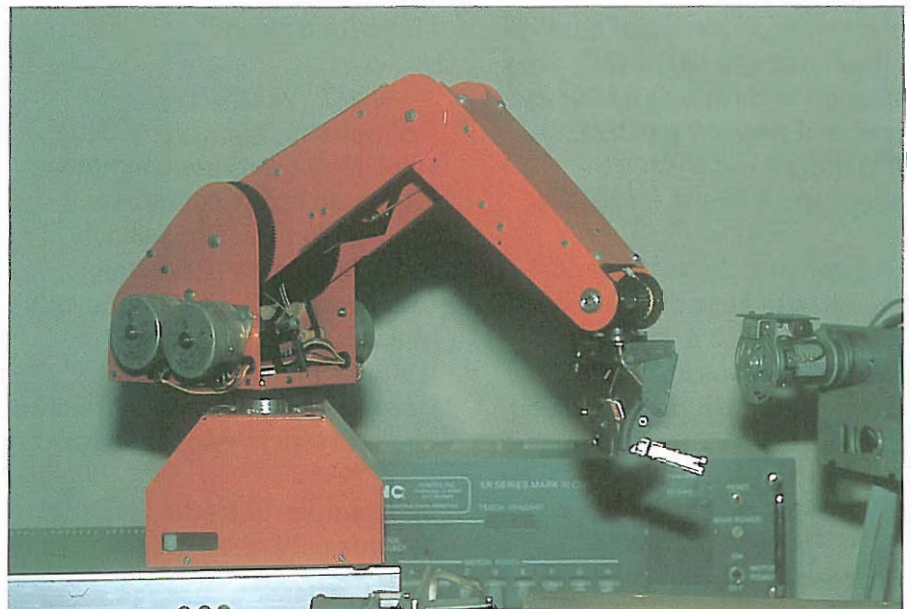
**D**opo aver descritto due banchi manuali per la programmazione gestuale dei robot didattici, vedremo ora un linguaggio più semplice per la programmazione testuale di Armdroid. Il tempo utilizzato nello sviluppo di programmi sul computer principale è un fattore chiave per determinare la produttività. La programmazione testuale non ha bisogno di disporre del robot per la composizione delle istruzioni, e dato che spesso i progettisti dell'applicazione di robotica non sono specialisti in informatica, è importante che i linguaggi siano facili e intuitivi.

All'interno di questi tipi di linguaggi ci sono molti livelli di complessità, e il MIKAN I, sviluppato nel Dipartimento di Architettura dei Computer dell'Università di Deusto, costituisce il primo scalino.

È basato sulla regolazione dei movimenti delle giunzioni del robot: a partire da queste strutture di base le istruzioni si possono complicare per facilitare la realizzazione dei lavori.

### Il linguaggio MIKAN I

È composto da nove istruzioni e due funzioni speciali ed è orientato al controllo dei



Le articolazioni di Armdroid sono governate da motori PAP.

movimenti delle articolazioni del robot Armdroid. È confezionato con il linguaggio Assembler per il microprocessore 6502. Come si può vedere nella figura, ci sono sei motori che governano la rotazione delle articolazioni, uno per la base, l'altro per la spalla, l'altro per il gomito, due per il polso (PITCH e ROLL) e, infine, un sesto motore destinato all'apertura e alla chiusura della pinza.

### Le istruzioni di movimento

Descriveremo ora le istruzioni di MIKAN I che permettono di

controllare i movimenti delle articolazioni e dell'elemento o pinza terminale. Questo sottoinsieme di istruzioni forma il nucleo principale nel confezionamento dei programmi:

#### — MOV (MUOVERE):

Permette di programmare un movimento del manipolatore che comprenda la rotazione simultanea della base, della spalla, del gomito e il PITCH del polso. Il movimento può essere realizzato in qualunque verso e bisogna solamente fornire il numero di passi che deve girare ogni motore. Il formato di questa istruzione è: MOV xxxx, wwww, yyyy, zzzz,

dove xxxx sono il numero dei passi della base, wwww della spalla, yyyy del gomito, e zzzz quelli corrispondenti all'angolo PITCH del polso. Bisogna sempre esprimere con quattro cifre il numero dei passi. Quando la rotazione è in senso orario, il segno è positivo, e quando è in senso antiorario il segno è negativo.

*Esempio:* L'istruzione per muovere la base di Armdroid di 33 passi a destra, la spalla 135 passi in basso, il gomito 21 passi in alto, e il polso (PITCH) 3 passi verso l'alto, sarà:  
MOV 0033, 0135, -0021, -0003

#### — ROT (ROTAZIONE):

Produce la rotazione del polso ROLL. Utilizza quattro cifre per esprimere il numero di passi e se il verso è negativo bisogna anteporre il segno meno al valore.

*Esempio:* L'istruzione che fa girare il polso in senso negativo di ROLL 58 passi è:  
ROT -0058

#### — CIE (CHIUDERE):

Chiudi la pinza fino a quando un sensore (SW2) genera un livello logico attivo. Questo livello si produce sia se la pinza si chiude senza aver preso un oggetto, sia che ne abbia preso uno di qualsiasi dimensione. Il suo formato non ha bisogno di operandi.

#### — ABR (APRIRE):

Produce l'apertura della pinza. Per fare in modo che abbia effetto è necessario che quest'ultima sia stata chiusa

in precedenza, in caso contrario viene ignorata. Il numero di passi con cui si apre la mano, è lo stesso utilizzato per chiuderla l'ultima volta con l'istruzione CIE. Manca di operandi.

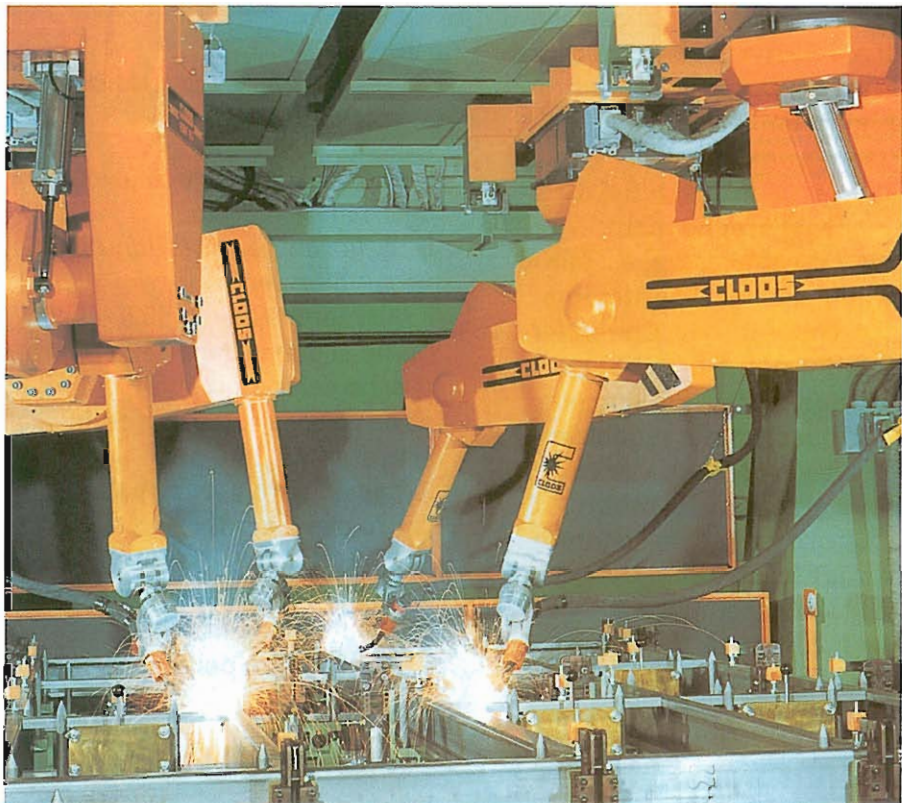
#### — VEL (VELOCITÀ):

Permette la regolazione della velocità di rotazione dei motori PAP in un range compreso tra 1 e 120 rpm. Dopo aver eseguito questa istruzione tutte le articolazioni si muovono alla velocità specificata. Il suo formato è VEL xxx. Dove xxx è un numero di tre cifre che esprime la velocità in rpm.

*Esempio:* Si tratta di confezionare un programma di

lavoro per Armdroid con il linguaggio MIKAN I. Aprire la mano e muovere il gomito a una velocità di 40 rpm, 400 passi verso l'alto. Dopo, muoverlo 400 passi verso il basso a una velocità di 30 rpm. Quindi, ruotare la base di 1.000 passi a sinistra a 120 rpm. Termina il lavoro chiudendo la pinza a 40 rpm.

```
ABR
VEL 040
MOV 0, 0, 0400, 0
VEL 030
MOV 0,0, -0400, 0
VEL 120
MOV 1000, 0, 0, 0
VEL 040
FIN
```



Nei lavori complessi, come quelli che realizzano questi quattro robot ROMAT 76 saldando elementi di case prefabbricate, la programmazione è uno dei fattori determinanti nella produttività finale.