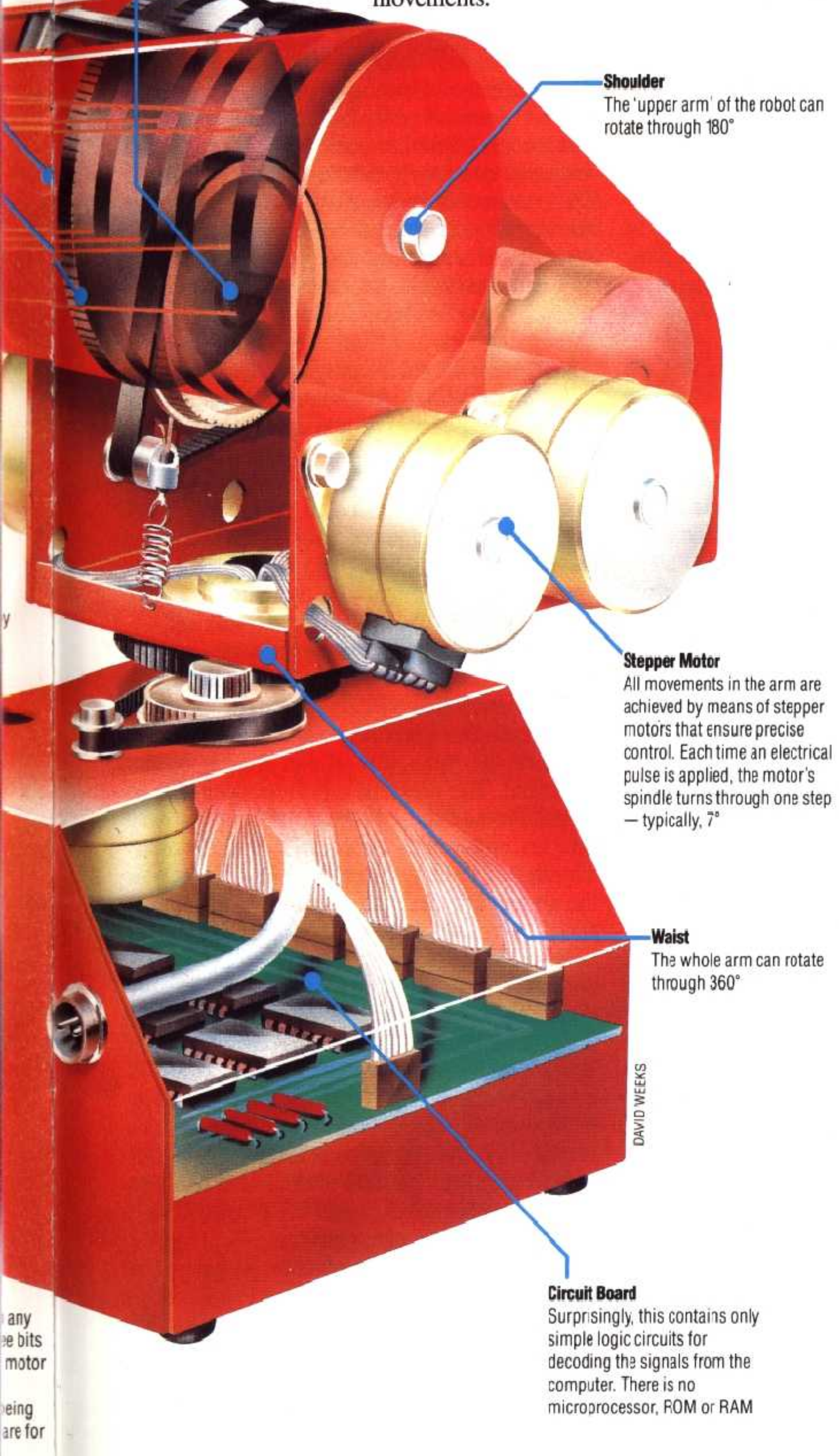
In order to make the arm move into position and grip an object, it is first necessary to divide the overall movement into a set of simple steps. Each motor will need to be instructed separately in a precise movement that will together compose the total motion of the robot arm. This information is then stored in the computer's memory and the arm can be made to repeat the operation as many times as required. Most robot arms currently available are supplied with programs to drive them that include routines to 'learn' sequences of movements.

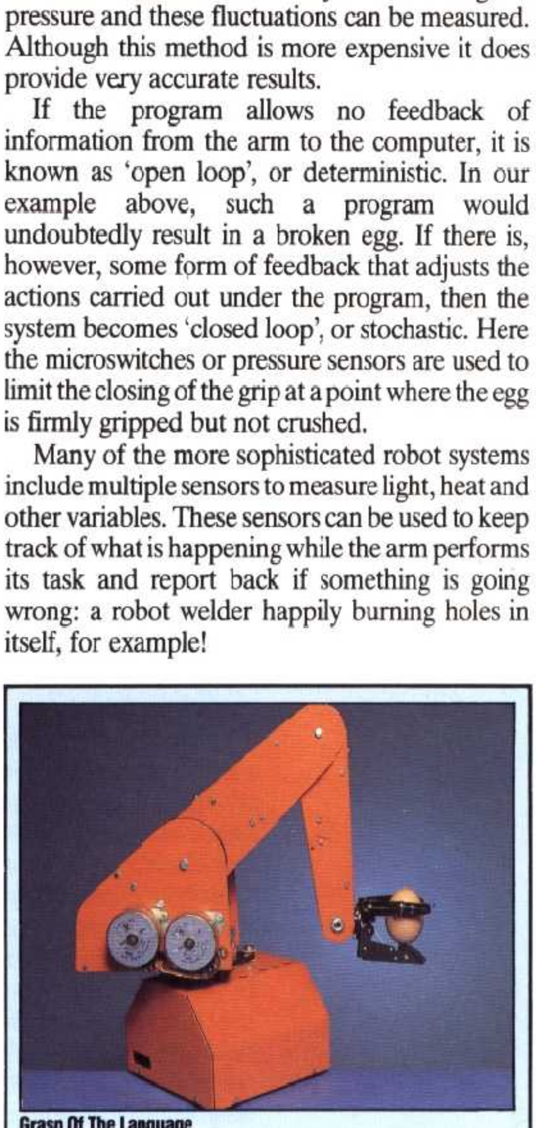


If the arm is handling delicate objects — the normal testpiece is an egg — the computer must be made to monitor the pressure of the grip. If it is too light the egg will fall; if it is too tight the shell will be broken. Various methods are used to convey information from the arm to the computer, but the most common involve simple microswitches. These can be fitted to set the limits of travel of the arm (most low-cost arms don't include sensors), or they can be built into the grip to detect a pre-set pressure limit.

The main alternative system to microswitches, used on most of the bigger arms, is based on pressure sensing. Certain materials alter their electrical resistance when subjected to change of pressure and these fluctuations can be measured. Although this method is more expensive it does provide very accurate results.

If the program allows no feedback of information from the arm to the computer, it is known as 'open loop', or deterministic. In our example above, such a program would undoubtedly result in a broken egg. If there is, however, some form of feedback that adjusts the actions carried out under the program, then the system becomes 'closed loop', or stochastic. Here the microswitches or pressure sensors are used to limit the closing of the grip at a point where the egg is firmly gripped but not crushed.

Many of the more sophisticated robot systems include multiple sensors to measure light, heat and other variables. These sensors can be used to keep track of what is happening while the arm performs its task and report back if something is going wrong: a robot welder happily burning holes in itself, for example!



IAN McKINNELL

Grasp Of The Language
It is relatively simple to write a program that will control a robot arm. In BASIC the main task would be to enable your computer to accept control commands from the keyboard and pass these to the arm through the port using POKE. Similarly, input from the arm could be read from the associated port by the PEEK function. If speed is required above all else then machine code programming is essential. FORTH is a language that offers the programming ease of BASIC and most of the speed associated with machine code. This language is becoming available for an increasing number of home computers. Sections of a program, rather like subroutines or procedures, are given names that can be incorporated into the command set of the language. This makes it highly efficient for specialised applications such as robot arm control programs. A complete gripping operation, for example, could be controlled by the single command GRIP