



MOTOR CONTROL

In the last part of the Workshop series we looked at the construction of an interface that will enable us to control low voltage devices and considered the principles of motor control. Now we look at the software required to switch motors and develop a simple feedback control system.

Microprocessor control of external devices is now common in industry, its applications ranging from counting bottles on a conveyor belt to welding car bodies. The principles of all control systems are, firstly to input data from the outside world in a form acceptable to a microprocessor system; secondly to analyse that data; and thirdly to instigate external actions based on that analysis. If these three activities are repeated in a continuous cycle then we have a system known as *feedback control*.

To illustrate the principle of feedback control let us take the example of simmering a pan of soup on a stove. In order to cook the soup, the heat supplied must be sufficient to make it bubble, without making it boil over the lip of the pan. If we were to carry out this task ourselves, we might initially apply maximum heat to the pan until the soup started to bubble and then turn it down until the soup was simmering. If at any time the soup started to boil we would lower the heat further. In doing this sort of operation we monitor the state of the soup visually; analyse what we see; and take the appropriate action. We would repeat these actions until the soup was cooked. A microcomputer could control the cooking of the soup in a similar, but not identical way. The main difference would be in the way that the soup's state was monitored. Whereas we are able to look at the soup and make an assessment using our experience, a computer system would have to use another method based upon easily determined physical properties, such as temperature. Before the soup-monitoring was computerised, someone would need to carry out initial experiments to determine what temperature corresponded to an even simmer, and at which temperature the soup would boil over. From then on however, the computer could take over the job, provided that the appropriate devices were available to monitor temperature and regulate the heat.

Motors can be controlled in several ways using the buffer box (see page 546) and low voltage output box (see page 574) that we have constructed. A Lego or electric train set motor is ideal to test the software that we shall design. We

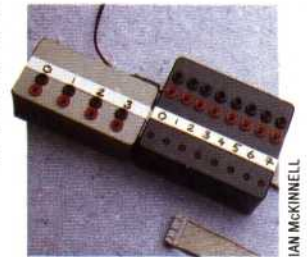
need only to ensure that the motor we connect to the output box has a voltage rating equal to or greater than the input voltage from the transformer. Connecting the two motor terminals to line 0 of the output box, we can switch the motor on and off using the 'Z' and 'X' keys on the keyboard.

BBC MICRO

```
10 REM BBC SIMPLE MOTOR
20 DDR=&FE62:DATREG=&FE60
30 ?DDR=255:REM ALL LINES OUTPUT
40 ?DATREG=0:REM MOTOR OFF
50 REPEAT
60 AS=INKEY$(1):REM KEYPRESS?
70 IF AS="Z" THEN ?DATREG=1:REM TURN ON
80 UNTIL AS="X"
90 ?DATREG=0:REM TURN OFF
```

COMMODORE 64

```
10 REM CBM64 SIMPLE MOTOR
20 DDR=56579:DATREG=56577
30 POKEDDR,255:REM ALL LINES OUTPUT
40 POKEDATREG,0:REM MOTOR OFF
50 GETAS:IFAS<>"Z" ANDAS<>"X" THEN
50:REM AWAIT KEY
60 IF AS="Z" THEN POKEDATREG,1:GOTO50
70 IF AS="X" THEN POKEDATREG,0:END
```



Put Out More Buffers

The buffer box, our first construction, is intended to protect the computer's circuitry from excessive input or output currents. Its independent power supply drives the output box — the more recent project — and allows software-controlled switching of the 12v output

Exercises

We can now design some elegant control experiments that rely on input and output using the buffer box and low voltage output box. The sensors referred to in the following experiments can be pressure sensors or 'reed' switches operated by a small magnet. Such devices are easily obtainable from electrical and electronics suppliers at low cost. Heat-sensitive switches operate by making an internal contact between the terminals when a certain temperature is reached and can be similarly obtained.

- 1) Write a program to make a vehicle run backwards and forwards between two sensors laid in its path.
- 2) Write a program to make a vehicle stop directly over a sensor, reversing back to it if necessary.
- 3) Write a program to keep a beaker of water between 70° and 100°C using a low voltage water heater and two heat-sensitive switches.
- 4) Write a program to calculate the speed in metres per second of a vehicle travelling between two points. (You will need to know the distance between the points and record the time taken to travel the distance.)