We can control the direction of the motor by connecting the motor terminals to adjacent lines on the output box. The diagram shows these connections. We shall also use a simple make/break type switch, connected to line 7 of the buffer box to control the direction.

In the following program a value of 1 in the data register causes current to flow one way through the motor. Placing a value of 2 in the data register will cause the current to flow in the reverse direction. The program repeatedly tests line 7 and only places a 2 in the data register when the line is set low (i.e. the switch is closed). In this way, the closing and opening of the switch controls the motor's direction. This is a very simple example of a feedback control system.

BBC MICRO

```
10 REM BBC DIRECTED MOTORS
20 DDR=&FE62:DATREG=&FE60
30 ?DDR=127:REM LINE 7 INPUT
40 ?DATREG=0:REM TURN OFF
50 A$=GET$:REM AWAIT KEYPRESS
60 REPEAT
70 A$=INKEY$(1)
80 IF (?DATREG AND 128)=0 THEN DIR=2 ELSE DIR=1
90 ?DATREG=DIR
100 UNTIL A$="X":REM PRESS X TO END
110 ?DATREG=0:REM TURN OFF
```

COMMODORE 64

```
10 REM CBM64 DIRECTED MOTORS
20 DDR=56579:DATREG=56577
30 POKEDDR,127:REM LINE 7 INPUT
40 POKEDATREG,0:REM ALL OFF
50 GETA$:IFA$=" "THEN50:REM AWAIT KEYPRESS
60 GETA$
70 IF (PEEK(DATREG)AND128)=0THENPOKEDATREG,
   2:GOTO90
80 POKEDATREG,1
90 IFA$<>"X"THEN60
100 POKEDATREG,0:REM TURN OFF
```

In addition to being able to control the direction of motors, we can also control their speed directly from the output box. This does not require complicated devices, such as digital-to-analogue converters to control the supply to the motors. Instead we can send pulses to the motor, turning it on and off in rapid succession. If we do this fast enough, the motor will appear to rotate continuously; the interval between each pulse determining the speed at which the motor turns. In order to program this, all we require is a pair of delay loops of adjustable length, within a larger repetitive structure, to determine the length of time that the motor is on and off during each cycle.

BBC MICRO

```
10 REM BBC VARIABLE MOTOR CONTROL
20 DDR=&FE62:DATREG=&FE60:SPEED=30
30 ?DDR=255:REM ALL OUTPUT
40 ?DATREG=0:REM ALL OFF
50 A$=GET$:REM AWAIT KEYPRESS
60 REPEAT
70 A$=INKEY$(1)
80 ?DATREG=0:REM TURN OFF
90 FORI=1TO(100-SPEED):NEXT:REM DELAY1
100 ?DATREG=1:REM TURN ON
110 FORI=1TO SPEED:NEXT:REM DELAY 2
120 IF A$="D"THEN SPEED=SPEED-5
130 IF A$="Z"THEN SPEED=SPEED+5
140 UNTIL A$="X"
150 ?DATREG=0:REM TURN OFF
```

COMMODORE 64

```
10 REM CBM64 VARIABLE MOTOR CONTROL
20 DDR=56579:DATREG=56577:SPEED=30
30 POKEDDR,255:REM ALL LINES OUTPUT
40 POKEDATREG,0:REM TURN OFF
50 GETA$:IFA$=" "THEN50:REM AWAIT KEY
60 GETA$
70 POKEDATREG,0:REM TURN OFF
80 FORI=1TO(100-SPEED):NEXT:REM DELAY1
```


Four Wheel Drive
The two boxes share the data bus and power supply through their minicon ports; if each box is built with a socket and a plug, then boxes can be connected in chains, 'piggyback' fashion.

The car has one 12v DC motor, supplied from the output box, and switched by bit 7 of the buffer box. The polarity of the outputs can be reversed under software control, driving the car forward and backward