DUAL CONTROL

In the last section of this series we wrote the software to control a Lego car with one motor. We were able to move the car in two directions by simply switching the current from the output box. Now we extend these principles to control a Lego car powered by two motors.

If we use two motors of equal power to drive a vehicle, we can gain computer control over all directions of movement by combining the forward and reverse movements of each motor. This allows us to turn the vehicle as well as direct it forwards or backwards. There are, in fact, two methods of turning a twin motor vehicle; the first of these is simply to stop one motor while turning the other. This will cause the vehicle to turn in an arc, pivoting about the stationary wheel(s). The second method involves turning one motor backwards as the other turns forward, improving the manoeuvrability of the vehicle as, in turning, the vehicle will pivot about its central axis.

We can control each motor bi-directionally by using the four red outputs on the low voltage output box (built on page 574) and connecting the right-hand motor to terminals 0 and 1, and the left-hand motor to terminals 2 and 3 (positive and negative, respectively).

Each motor is connected across a pair of adjacent positive output terminals so that we can

Tandem Turtle
Having discovered how to drive the Lego vehicle backwards and forwards, we can now link two of them to make a buggy. The motors can be switched individually, making the new vehicle far more manoeuvrable than the old

have independent directional control of each motor. By placing the appropriate number in the data register we can now make the vehicle move forwards or backwards, or turn to the left or right. The right-hand (RH) motor will go forward if line 0 is set high and line 1 is set low, and go backwards if line 1 is set high and line 0 is set low. Similarly, the left-hand (LH) motor will go forwards if line 2 is high and line 3 is low, and backwards if the converse is true. By combining these movements we can control the motion of the whole vehicle:

Vehicle Motion	LH Motor	RH Motor	Bit Pattern	No In DATREG
OFF	OFF	OFF	0000	0
FORWARD	FORWARD	FORWARD	0101	5
REVERSE	REVERSE	REVERSE	1010	10
PIVOT LEFT	FORWARD	REVERSE	0110	6
PIVOT RIGHT	REVERSE	FORWARD	1001	9
ARC LEFT	FORWARD	OFF	0100	4
ARC RIGHT	OFF	FORWARD	0001	1

The following program allows us to control the vehicle directly from the keyboard using 'T' for forwards, 'B' for reverse, 'F' to turn left, and 'H' to turn right. If no key is pressed then the vehicle will stop.

BBC MICRO

```
  10 REM BBC TWIN MOTORS
  20 DDR=&FE62:DATREG=&FE60
  30 ?DDR=255
  40 REPEAT
  50 A$=INKEY$(10)
  60 PROCtest_keyboard
  70 UNTIL A$="X"
  80 ?DATREG=0
  90 END
1000 DEF PROCtest_keyboard
1010 IF A$=" " THEN ?DATREG=0
1020 IF INKEY(-36) = -1 THEN ?DATREG=5
1030 IF INKEY(-101) = -1 THEN ?DATREG=10
1040 IF INKEY (-68) = -1 THEN ?DATREG=6
1050 IF INKEY (-85) = -1 THEN ?DATREG=9
1060 ENDPROC
```

COMMODORE 64

```
  10 REM CBM 64 TWIN MOTORS
  20 DDR=56579:DATREG=56577
  25 POKE650,128: REM REPEAT KEY MODE
  30 POKE DDR,255
  40 GETA$
  50 GOSUB1000:GOTO70
  60 POKEDATREG,0
  70 IF A$<>"X" THEN FOR I=1TO100:NEXT:GOTO40
  80 POKE DATREG,0
  90 END
1000 REM TEST INPUT S/R
1005 IFA$=" " THEN POKE DATREG,0
1010 IFA$="T" THEN POKE DATREG,5
1020 IFA$="B" THEN POKE DATREG,10
1030 IFA$="F" THEN POKE DATREG,6
1040 IFA$="H" THEN POKE DATREG,9
1050 RETURN
```


IAN McKINNELL