



ADVAL(0) performs two different functions. The least significant two bits correspond to the fire buttons on joystick 1 and joystick 2. X=ADVAL(0) AND 3 will return a value of one if joystick 1's fire button is pressed. X=ADVAL(0) DIV 256 will give the number of the channel that last completed an A-to-D conversion.

As conversion of each analogue input channel takes about 10 milliseconds, then to process each of the four channels will take 40 milliseconds. In our application we use channels 1 and 2 only. We can cut down on wasted conversion time by specifying the channels that require conversion. This can be done by using *FX16,2, which enables channels 1 and 2 but disables channels 3 and 4.

The following program combines all this information to control a twin-motor Lego car.

```
10 REM BBC JOYSTICK CONTROL
20 DDR=&FE62:DATREG=&FE60
30 ?DDR=255:REM ALL OUTPUT
40 REM ENABLE A-D CHANNELS 1&2
50 *FX16,2
60 REPEAT
70 PROCtest_joystick
```

```
80 UNTIL fire=1
90 END
100 :
110 DEF PROCtest_joystick
120 REPEAT
130 channel=ADVAL(0) DIV 256
140 UNTIL channel<>0:REM WAIT FOR CONVERT
150 IF channel=1 THEN PROCleft_right
160 IF channel=2 THEN PROCup_down
170 ENDPROC
180 :
190 DEF PROCleft_right
200 REPEAT
210 joyval=ADVAL(1)
220 IF joyval<100 THEN ?DATREG=9
230 IF joyval>64000 THEN ?DATREG=6
240 fire=ADVAL(0) AND 3
250 PRINT?DATREG,channel,joyval
260 UNTIL(joyval>100 AND joyval<64000) OR fire=
270 ?DATREG=0
280 ENDPROC
290 :
300 DEF PROCup_down
310 REPEAT
320 joyval=ADVAL(2)
330 IF joyval<100 THEN ?DATREG=10
340 IF joyval>64000 THEN ?DATREG=5
350 fire=ADVAL(0) AND 3
360 PRINT?DATREG,channel,joyval
370 UNTIL(joyval>100 AND joyval<64000) OR fire=
380 ?DATREG=0
390 ENDPROC
```

Exercise Answers

1) Calibration of your vehicle can be done by timing the period taken to travel various distances, typically 10 cm, 20 cm, 50 cm, 100 cm and 150 cm. By calculating the speed over each distance and averaging, a good estimate can be made for the distance travelled in a second. This value can then be used to control the vehicle over measured distances. A similar approach could be adopted for turning, selecting a number of angles and making timings. Controlling motors by switching them on and off over measured time periods can present many difficulties, not least that the structure of the controlling program is such that time intervals must be measured as accurately as possible. Differences of a few hundredths of a second in timing can produce large discrepancies in distances travelled or angles turned through. These problems can be reduced substantially by introducing reduction gearing between the motor and the driving wheels.

2) The program listing given on page 613 will allow you to steer the vehicle through the obstacle course. Retracing the pattern in reverse is a little more tricky. We must first assign a pair of variables for each direction together with its inverse. So, for example, forward and reverse are paired together.

Commodore 64

```
17 A(1)=5:B(1)=10:A(2)=10:B(2)=5
18 A(3)=6:B(3)=9:A(4)=9:B(4)=6
```

The following routines can then be added to play back the recorded sequence in reverse.

```
92 GOSUB2000:REM REVERSE REPLAY
2000 REM REVERSE REPLAY S/R
2010 FOR I=0 TO 1 STEP -1
2020 FOR J=1 TO 4
2030 IF OR(I,1)=R(J) THEN POKE DATREG,B(J):J=4
2040 NEXT J
2050 T=T1
2060 IF (T1-T)<OR(I,2) THEN 2060
```

BBC Micro

```
2070 NEXT J
2080 STOP
2090 RETURN
1020 DIM DR(100,2),A(10),B(10)
1030 A(1)=5:B(1)=10:A(2)=10:B(2)=5
1040 A(3)=6:B(3)=9:A(4)=9:B(4)=6
1115 PROCreverse_replay
2000 DEF PROCreverse_replay
2010 FOR I=0 TO 1 STEP -1
2020 FOR J=1 TO 4
2030 IF OR(I,1)=R(J) THEN?DATREG=B(J):J=4
2040 NEXT J
2042 TIME#9
2045 REPEAT UNTIL TIME#DR(I,2)
2047 ?DATREG#A
2050 NEXT I
2055 PRINT"IT 0 TO CONTINUE"
2060 REPEAT AB=GET#9
2070 UNTIL AB=0
2080 ENDPROC
3000 FOR I=1 TO 4:PRINTOR(I,1),DR(I,2)
3010 NEXT
```

3) Assuming that line 7 is forward, line 6 is reverse, line 5 is left and line 4 is right:

```
10 REM BBC EXTERNAL SWITCHES
20 DDR=&FE62:DATREG=&FE60
30 ?DDR=15:REM LINES 4-7 INPUT
40 ?DATREG#0
50 PROCtest_line#6
60 GOTO50
70 :
80 DEF PROCtest_line#6
90 IF ?DATREG AND 240 =240 THEN ?DATREG#5
100 IF ?DATREG AND 120 =0 THEN ?DATREG#4
110 IF ?DATREG AND 60 =0 THEN ?DATREG#3
120 IF ?DATREG AND 30 =0 THEN ?DATREG#2
130 IF ?DATREG AND 16 =0 THEN ?DATREG#1
140 ENDPROC
10 REM CBM 64 EXTERNAL SWITCHES
20 DDR=56578:DATREG=56577
30 POKEDDR,15:REM LINES 4-7 INPUT
40 POKEDATREG,0:REM MOTORS OFF
50 REM L7 FORWARD,L6 REV,LS LEFT,LR RIGHT
55 REM LS ONE OF THE INPUT LINES LOW
60 IF (PEEK(DATREG)AND240) < 240 THEN GOSUB 1000:GOTO60
70 POKEDATREG,0:GOTO60
80 :
1000 REM SCAN INPUT LINES S/R
1005 IF (PEEK(DATREG)AND120)=0 THEN POKE DATREG,5
1010 IF (PEEK(DATREG)AND64)=0 THEN POKE DATREG,10
1020 IF (PEEK(DATREG)AND32)=0 THEN POKE DATREG,15
1030 IF (PEEK(DATREG)AND16)=0 THEN POKE DATREG,20
1040 RETURN
```