WEIGHING IN

Having completed the construction of the robot motors and microswitch sensors, we must now calibrate the robot to allow us accurate control when moving it through measured distances and angles. We also look at software that makes use of the interaction between sensors and motors.

Stepper motors are ideal for control by digital devices as they turn through a precise step every time the motors receive a pulse. In order to relate digital stepper motor control to the real world of distance and angle, we must carry out some initial experiments on our robot. These are designed to determine the number of pulses required to move the robot through various distances and angles. Having performed these experiments we should be able to determine average pulse/distance and pulse/angle ratios that we can enter as constants to programs. In future instalments of Workshop we shall be designing software, that will, in addition to other applications, allow the robot to probe and build up digital representations of solid objects. To enable the robot to function accurately, we shall require the ratio values obtained from carrying out the experiments in this section.

LINEAR CALIBRATION

We can make a guess at the pulse/distance ratio of our robot by using some elementary mathematics. As one pulse induces a 7.5° turn in the motors, putting the motor output through a 25:2 gear ratio means that one pulse will induce a turn of 7.5×2/25 = 0.6° turn at the axle. As the Lego wheel has a radius of 30mm the linear movement per pulse can be calculated as follows: 1 pulse causes 0.6/360×2×PI×30mm movement. Breaking this expression down shows us that 1 pulse causes 0.1×PImm movement. Inverting this figure gives us a theoretical pulse/distance ratio: p/d ratio = 3.183.

The calibration program that follows allows you to run your robot through trials over various distances. On each run the number of pulses and the theoretical distances that should be travelled are displayed on the screen. Using two 30cm rulers end-to-end, you can record the actual distance travelled over each trial. The program then displays a table of the number of pulses, the actual distances recorded, and the theoretical estimates. An average p/d ratio is also calculated. This figure is important so make a separate note of it. The sample output from this program shows that our prototype robot tends to travel slightly further for a given number of pulses than theory would suggest. The importance of the exercise is that you find the p/d ratio for your robot and use it in future programs as required.

```
 10 REM **** BBC CALIBRATION ****
 20 DDR=&FE62:DATREG=&FE60
 30 ?DDR=15:REM LINES 0-3 OUTPUT
 50 forwards=4:backwards=2:DIM MD(12)
 60 FOR CC=500 TO 1700 STEP 100
 70 ?DATREG=0
 80 ?DATREG=(?DATREG OR 1) OR forwards
 90 PRINT CC,INT(CC*PI)/10
100 A$=GET$
110 FOR I=1 TO CC
120 PROCpulse
130 NEXT I
140 INPUT"MEASURED DISTANCE IN MM";MD((CC-500)/100)
150 NEXT CC
160 ?DATREG=0:T=0
180 PRINT"    PULSES","  MEASURED","  THEORET."
190 PRINT
200 FOR CC=500 TO 1700 STEP 100
210 PRINT CC,MD((CC-500)/100),INT(CC*PI)/10
220 T=T+CC/MD((CC-500)/100)
230 NEXT CC
240 PRINT:PRINT "PULSE TO DISTANCE RATIO:",T/12
260 END
270 DEF PROCpulse
280 ?DATREG=(?DATREG OR 8)
290 ?DATREG=(?DATREG AND 247)
300 ENDPROC
```

```
10 REM **** CBM 64 CALIBRATION ****
20 DDR=56579:DATREG=56577
30 POKE DDR,15:REM LINES 0-3 OUTPUT
50 FW=4:BW=2:DIM MD(12)
60 FOR CC=500 TO 1700 STEP 100
70 POKE DATREG,0
80 POKE DATREG,(PEEK(DATREG)OR 1)OR FW
90 PRINT CC,INT(CC*π)/10
100 GET A$:IF A$="" THEN 100
110 FOR I=1 TO CC
120 GOSUB 270:REM PULSE
130 NEXT I
140 INPUT"MEASURED DISTANCE IN MM";MD((CC-500)/100)
150 NEXT CC
160 POKE DATREG,0:T=0
170 REM ** LINES 180-260 AS BBC VERSION **
175 REM ** BUT REPLACE PI BY π IN LINE 210 **
270 REM **** PULSE S/R ****
280 POKE DATREG,PEEK(DATREG)OR 8
290 POKE DATREG,PEEK(DATREG)AND 247
300 RETURN
```

ANGLE CALIBRATION

We can calculate a pulse/angle ratio as follows: if the wheelbase is 140mm, the turning circle circumference = 140×PI. A pulse causes a turn of 360×0.1×PI/(140×PI) degrees, and the p/a ratio is therefore 3.846.

One of the major problems involved in performing angular calibration is the accurate measurement of angles. As for most applications the robot will be turning through angles of 90° (or multiples thereof), our theoretical p/a ratio tells us that 346 pulses should be required for a right-angled turn.

Mark on a piece of paper a pair of perpendicular lines. On one of the lines make two small marks on either side of the point at which the lines cross to designate the starting points for the robot wheels. Run the accompanying program to turn the robot through 90°. The `FOR...NEXT` loop at line 70 dictates the number of pulses passed to the motors. The figure 371 is the experimental value required to turn our prototype robot through 90°. Edit the program, altering the upper limit of