GET MOTORISED!

by Mark Brighton

DC Motor

Many of you will have seen or heard about mice and turtles. They whizz through mazes or scuttle around the floor almost like living creatures, providing endless fascination with their antics, especially for the kids.

They are in fact, an excellent means of introduction to computing for young people who will then gain a good understanding of functional programming for robotic control, and develop their concept of artificial intelligence by experimentation, and the Atari 400/800 computer is ideally suited to controlling this type of simple robot, having 16 (8 for XL owners) programmable control lines available at the joystick sockets. Some of these lines could be used to control motors, and some to read information from sensors, etc, which would give the computer information about the robot's surroundings, and so enable it to make a decision regarding what to do next.

Of the different types of motors suitable for control by the Atari joystick ports, the DC motor is the simplest to interface, requiring only a simple external transistor switch to enable the computer to control the motor power supply. The external power supply is necessary because the joystick sockets alone cannot supply the power required to run the motor. Such a circuit is shown in Figure 1, and when plugged into Port 1, the motor may be turned on and off using Listing 1.

Two transistors are used because the first transistor provides the current to switch the motor control transistor hard on from the +12V supply, as the high impedance joystick line could not supply this current alone. However, this circuit has one serious drawback in that it can only make the motor run in one direction, being incapable of reversing the supply to the motor. Where bi-directional operation is required, the bridge circuit of Figure 2 may be used. It may be seen that, depending on which pair of transistors are turned on, the polarity of the supply connected to the motor is controllable. Note that the control inputs should never be high together, as this results in a short circuit of the motor power supply. Listing 2 is a suitable program to drive a DC motor in both directions. The preceding circuits have shown simple ways to connect a DC motor to your Atari.

Listing 1.

```
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CA 40 POKE PORT,0
AC 50 FOR DELAY=1 TO 400
KG 60 NEXT DELAY
CR 70 POKE PORT,1
AF 80 FOR DELAY=1 TO 400
KJ 90 NEXT DELAY
QG 100 GOTO 40
```

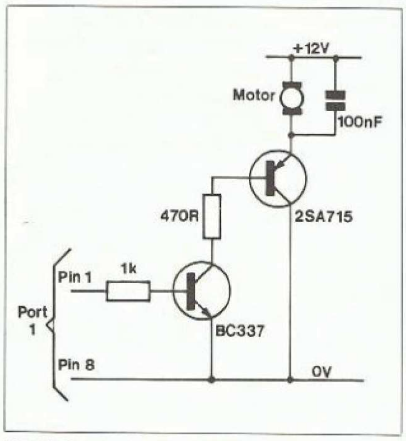

Figure 1.

Servo

However, the DC motor has one major disadvantage in robotic and machine applications, which is that its rotation is only turned on or off for an arbitary period, with no control over the actual position of the rotor, either during rotation or afterwards. This is because a DC motor cannot be controlled down to speeds where *part* of a rotation alone can occur, since it has no means of stopping the rotor in a particular place, or moving it through a precise angle.

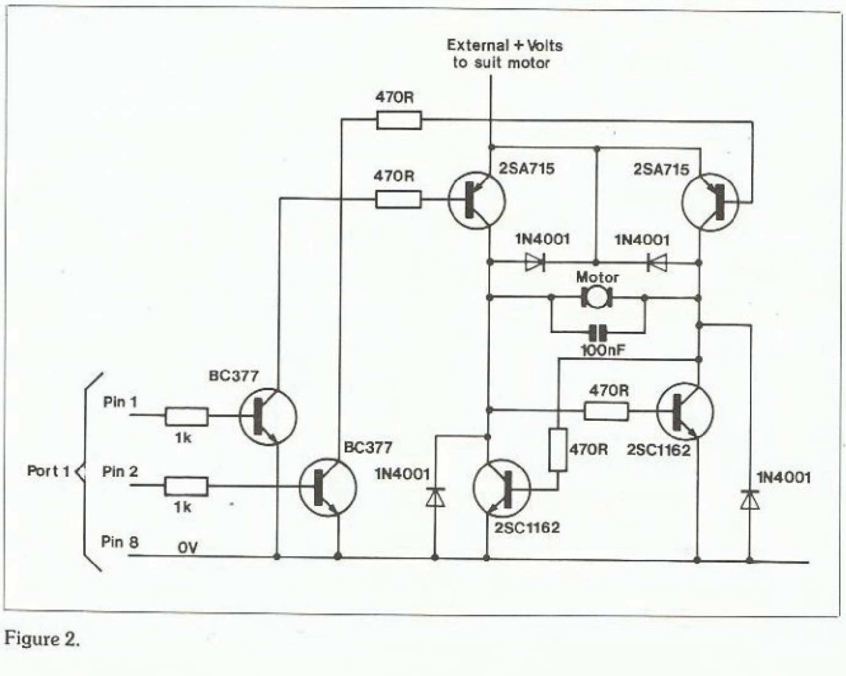

Figure 2.

The traditional answer to this problem has been to use a gearbox to reduce the final drive speed, and to increase both the holding and driving torque, combined with some form of positional sensor, such as a potentiometer. This is the standard set-up for the type of servos used in radio-controlled models, and may be used in a computer controlled set-up, assuming that some means is available to turn the analogue voltage information returned by the potentiometer into a digital format, which your computer can understand. the Atari 400/800 computer is, of course, equipped with a grand total of eight analogue to digital converters (XL owners read as 4), which are ideally suited to this task. An example wiring diagram is shown in Figure 3, and a test program to put a servo through its paces is included, see Listing 3. As before, an external power supply will be necessary to run the servo-motor. A suitable circuit is shown in Figure 4.

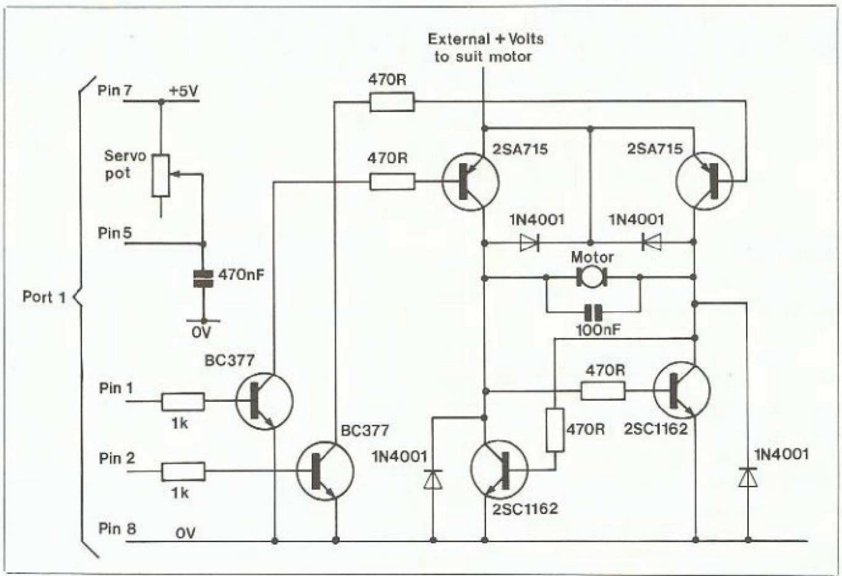

Figure 3.

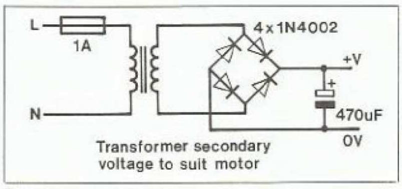

Figure 4.

Stepper Motor

Finally, a motor which is eminently suitable for digital control and small scale robotic/machine designs is the stepping motor. The motor has several separate stator coils which are arranged in such a manner that various combinations of energised coils cause the rotor to 'step' around by a small and definite amount, e.g. 7.5°. For this reason, the stepper motor offers great flexibility in use, being completely controllable both in speed and the angle through which it turns. It also holds each new position it attains strongly, since the stator coils may remain energised even after the rotor has stopped turning. It may therefore, be used directly in most applications, requiring no gearbox and no means of positional feedback, since the computer knows how many steps the motor has taken and can be programmed to calculate the effect so produced.

The stepper motor requires a repeating sequence of several (usually) 4-bit words to cause it to rotate continuously in one direction which makes it a little more difficult to use than conventional DC motors. For this reason, Mullard have developed a chip which greatly simplifies the use of stepper motors, requiring only a single pulse to step the motor and a high/low logic level to determine the direction of rotation. A kit containing the IC and external components including a motor, together with a small pcb is available from Maplin Electronic Supplies Ltd, stock code LK76H at the meagre price of £15.95 (Phone 0702-552911 for details). This board will require the simple interface circuit, shown in Figure 5, to convert the 5V TTL pulses from the Atari joystick sockets into 12V pulses required by the motor driver chip.

A test program which demonstrates the degree of control obtainable over a stepper motor is shown in Listing 4, the reader should also bear in mind that this circuit is only suitable for uni-polar stepper motors. If in doubt, Maplin sell a reasonably priced motor that fits the bill nicely, stock code FT73Q price £9.95, but remember, this is supplied in the kit if you decide to buy it.

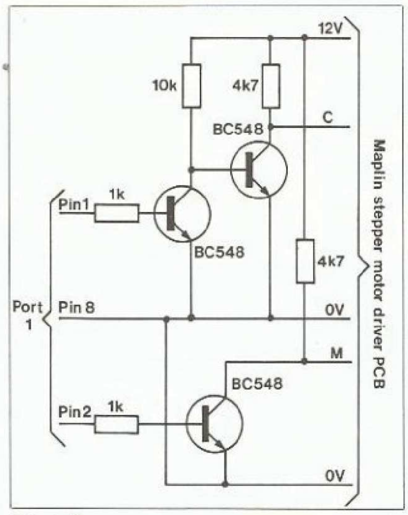

Figure 5.

Listing 2.

```
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CA 40 POKE PORT,0
AC 50 FOR DELAY=1 TO 400
KG 60 NEXT DELAY
CR 70 POKE PORT,1
AF 80 FOR DELAY=1 TO 400
KJ 90 NEXT DELAY
MT 100 POKE PORT,2
XR 110 FOR DELAY=1 TO 400
SP 120 NEXT DELAY
QM 130 GOTO 40
```

Listing 3.

```
ZS 5 REM ** HIGH AND LOW ARE THE
JF 6 REM ** MAX AND MIN VALUES READ
RQ 7 REM ** AT THE PADDLE PORT FOR
HW 8 REM ** THE EXTREMES OF TRAVEL
KL 9 REM ** OF YOUR SERVO.
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CO 40 POKE PORT,1
KC 50 IF PADDLE(0)<HIGH THEN 50
DF 70 POKE PORT,2
LN 80 IF PADDLE(0)>LOW THEN 80
LP 100 POKE PORT,0
XR 110 FOR DELAY=1 TO 400
SP 120 NEXT DELAY
QM 130 GOTO 40
```

Listing 4.

```
WQ 10 PORT=54016:REM J1
FF 20 POKE 54018,56:POKE PORT,255
QG 30 POKE 54018,60
CA 40 POKE PORT,0
IR 50 FOR STEP=100 TO 200
QD 60 POKE PORT,1+X:POKE PORT,0+X
WC 70 FOR DELAY=1 TO 200-STEP
KI 80 NEXT DELAY
GU 90 NEXT STEP
ZX 100 FOR STEP=1 TO 200
PK 110 POKE PORT,1+X:POKE PORT,0+X
NB 120 NEXT STEP
ZK 130 FOR STEP=1 TO 100
PQ 140 POKE PORT,1+X:POKE PORT,0+X
JL 150 FOR DELAY=1 TO STEP
SX 160 NEXT DELAY
NL 170 NEXT STEP
DG 180 IF X=0 THEN X=2:GOTO 50
```