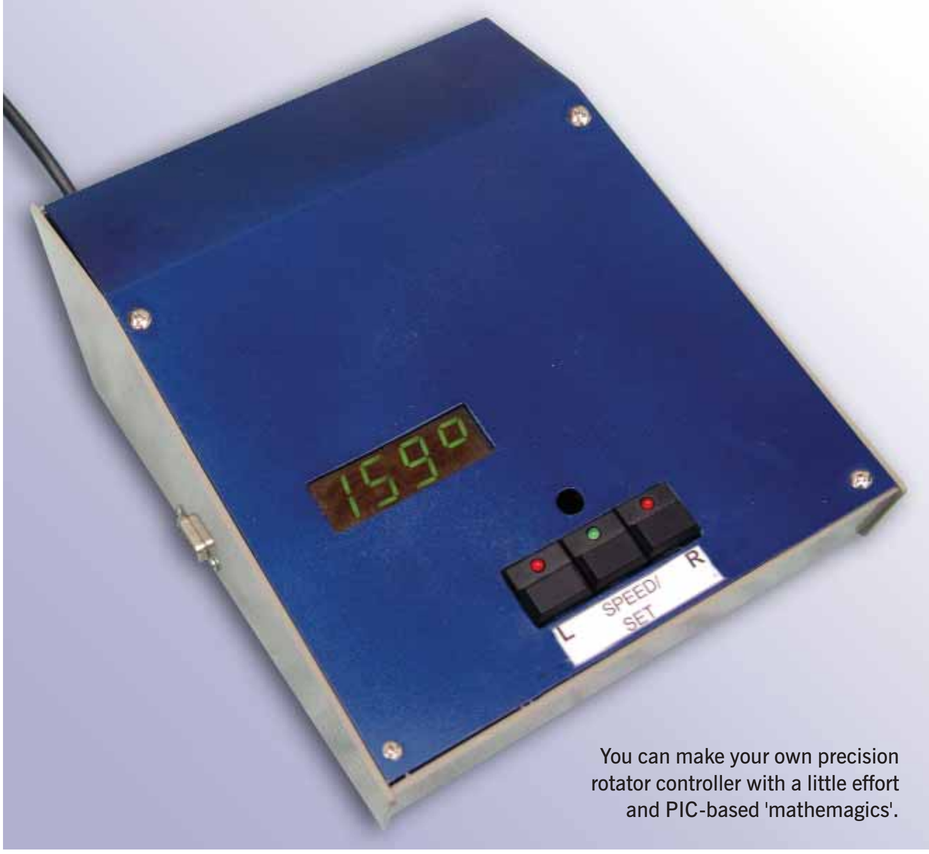
Short Circuits

G4JNT proposes a practical PIC-powered precision pointer.



You can make your own precision rotator controller with a little effort and PIC-based 'mathemagics'.

PRECISE ROTATOR CONTROL. This month we will look at the design concepts for an after-market add-on controller for antenna rotators giving digital readout, slow / fast speed, in-field calibration and remote control over a serial link. There are many antenna rotators with mechanical hardware capable of pointing an antenna to a precision of better than one degree and a popular design, often seen on the surplus stands, is the 'bell' type, as shown in Photo 1. Most of them are spoiled by having a simple controller with imprecise bearing readout. While fine for HF or small VHF beams, this readout is totally unsuitable for positioning a microwave dish to a bearing accuracy of one degree.

The head unit must provide positional feedback via a variable resistor or potentiometer; low cost units with pulsed feedback are not suitable for this modification.

This unit measures the voltage from the feedback resistor with an analogue to digital converter (ADC) in a PIC microcontroller, then calculates the bearing based on a set of calibration values specific to the rotator and displays this on a three digit LED display. Three pushbutton switches control left/right motion and fast/slow rotation speed; they also serve for in-field calibration.

Positional feedback is via a two wire variable resistor in the head unit whose value varies smoothly over the complete 360° of rotation. In many units, the mechanical design allows for accurate readout over significantly more than 360° with a resistor track having overlapping ends. This may have maximum and minimum resistance values of some arbitrary value at the rotator's physical end stops - the design will cope with such units.

READING THE POSITION. The first stage is to turn the resistance value into a voltage that can be measured with an ADC. Provided we can perform simple arithmetic in the microcontroller, there is a straightforward way to measure R by making it part of a potential divider with a fixed resistor R_{top}. Refer to Figure 1.

The voltage out is given by

$$V_{out} = V_{in} \times R / (R + R_{top})\ V_{out}$$

and does not vary linearly with position. ADCs have a measurement resolution based on the number of bits used for the conversion. Many PICs include a 10 bit ADC for a resolution of 1 in 2^{10}, or 1024 levels, which is about 0.1% accuracy; precise enough for angle readout to 1° resolution while allowing some overhead for calculation accuracy. ADCs always give a digital output, N, that is the fraction of a reference voltage, V_{ref}, multiplied by the full scale digital value. For a 10 bit ADC:

$$N = V_{in}/V_{ref} \times 1024.$$

If V_{ref} = 5V, an input voltage of 2.5V would result in N = 512. An input of 1.5V would give N = 307.

Taking the equations for the potential divider and the ADC together we get:

$$N = V_{in} \times R / (R + R_{top}) \times 1024 / V_{ref}.$$

There are two voltages defined here and they both need to be generated in our hardware, so let us generate V_{in} by amplifying V_{ref} (the 5V supply used for the PIC) by a fixed value K in an op-amp. $V_{in} = K \times V_{ref}$ and the equation now becomes:

$$N = 1024 \times K \times R / (R + R_{top})$$

Notice the *actual* voltage has disappeared

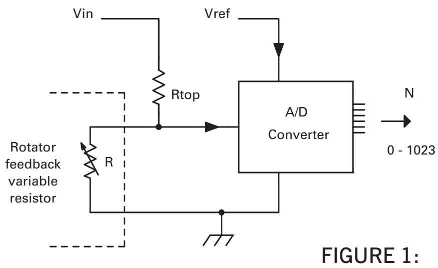


FIGURE 1: Simple method of interfacing a position feedback pot to the digital world using an ADC.

TABLE 1: Serial output data after the control unit is turned on and activated.

G4JNT Rotator Controller
Stored Constants -
Z 1807
M -15897
C 3758
K 2048
Commands # + Z/C/K/M + (-)nnnn(n)
b059.3,p1415
b058.6,p1422
b057.8,p1430
b057.8,p1430
b056.7,p1441
b054.3,p1466
b052.5,p1485
b052.5,p1485
b049.9,p1511
b051.0,p1500
b052.2,p1488
b053.9,p1470

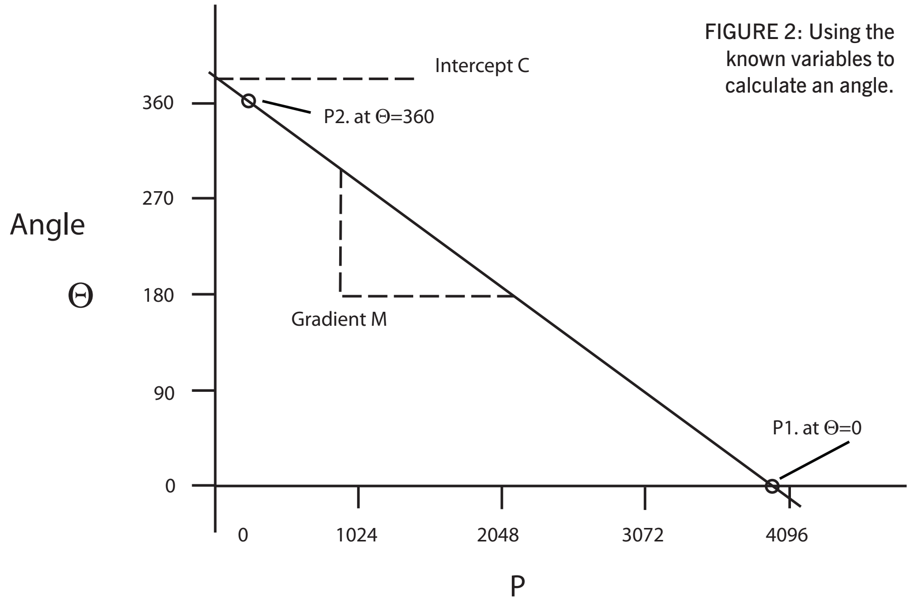


FIGURE 2: Using the known variables to calculate an angle.

PHOTO 1: The bell type rotator that is often found on surplus stands at rallies.

and we are now only working with ratios. To get the position of the wiper we rearrange the equation so we can derive R from N, leading to:

$$R / R_{top} = N / (1024 \times K - N)$$

For convenience we make R_{top} equal near-enough to the full scale value of the feedback resistor so the ratio of the variable and fixed resistors on the left hand side varies from about zero to about one. Then we accurately make K = 2, feeding 10V to the potential divider to utilise as much of the range of N from 0 to 1023 that is possible. As PIC processors can only do integer arithmetic, we will also scale the result by a convenient binary number, 4096, to maintain sufficient calculation accuracy without making the PIC code too complicated. We will call the scaled value P, (for Position). It will vary from around 0 to around 4096 *and is linearly related to the position of the rotator*. Gathering all the terms together, the equation the PIC has to calculate becomes:

$$P = 4096 \times N / (2048 - N)$$

We have to know K accurately to maintain linearity, but the rest of the calculation is ratiometric and errors can be taken out in calibration.

DERIVING BEARING FROM POSITION.

Now we need to turn the position *P* of the rotor around its track (0 to about 4096) into a bearing of 0 to 360°. We don't at this stage know the relationship between P and a rotator angle that, with overlap, could vary from, say, -10° to 370° about the physical end stop, with P varying linearly from around 0 to around 4096. We also don't know the bearing the physical end stop will be placed.

The PIC sends the value of *P* via the serial port to a host PC for calibration and setup purposes. To relate the angle to *P*, two calibration points must now be determined. The rotator is moved to each end of its span, to the spot just before the end stop where the anti-clockwise and clockwise positions overlap. The resulting values P_1 and P_2 transmitted on the serial output are noted. These become the 0 and 360˚ reference positions. The angular position of the rotator from this reference point is given the name θ. Note that the actual value of θ may vary from something like -10° to 370° if a decent overlap is allowed by the mechanics. Also, to allow for future enhancements and to make PIC arithmetic more precise, θ is measured in units of 0.1° so could potentially range in value from –100 to +3700.

The two values P1 and P2 are plotted on a straight line graph against angle θ, as shown in Figure 2. By obtaining the gradient, M, and the intercept, C, of this graph, it becomes possible to calculate any angle given *P*. The calculation of M and C only needs to be made once for any rotator, and does not have to happen within the PIC controller. A calibration routine running on a PC can do the job automatically. We now have a value of θ giving the angle of the rotator around its travel to a resolution of 0.1°. In this example the gradient is negative as the resistance decreases with increasing angle, but either direction is acceptable to the PIC software.

All that now remains is to define the actual bearing that corresponds to the overlap reference. This depends how the rotator is bolted to its mast and needs to be made easily adjustable. The PIC software allows a user to enter a bearing for this zero position that is then stored in non-volatile memory. The offset is added to the calculated θ value, remembering to correct the value by adding or subtracting 360° to keep the resulting bearing within the range 000° to 359˚.

THE REST OF THE HARDWARE. This is shown in the abbreviated circuit diagram given in Figure 3. Filtering components around the feedback lines are necessary to prevent motor-switching transients from upsetting or damaging the circuitry. Apart from the voltage measurement, the PIC reads two pushbuttons corresponding to left / right demand and generates control signals for the motor. A third button, when pressed with either of the turn buttons, pulses the output at 2Hz to step the motor slowly.

Many rotators use a three-wire AC motor with a capacitor to give a phase shift to force the motor to travel in either direction. The capacitor is usually a non-polarised electrolytic with a value of several microfarads and rated at typically 50 – 100V depending on the power supply. This capacitor and the supply transformer present the only special components needed for this rotator controller and can most easily be obtained by salvaging those in the original control box. The PIC outputs switch one of two thyristors to control the direction of travel, via MC3021 opto-isolated thyristor drivers to give isolation between the PIC and the high voltage AC. The CR networks across the thyristors are essential to prevent transients causing spurious triggering. Relays could be used instead, but with the slow option relays are not ideal for the job.

The PIC also drives a multiplexed seven segment LED display showing bearing to 1° resolution, which is also transmitted to 0.1° resolution along with the *P* value on an RS232 serial link. The link can also accept a number of commands. Simple go-left and go-right commands pulse the rotator in either direction. All the values needed for the calculations, that are normally stored in non-volatile memory in the PIC, can be updated by entering the values into the controller from a PC running Hyperterm or similar.

For operation in the field the zero offset frequently has to be changed. After the rotator has been installed on a mast, the bearing of the zero reference is determined manually –

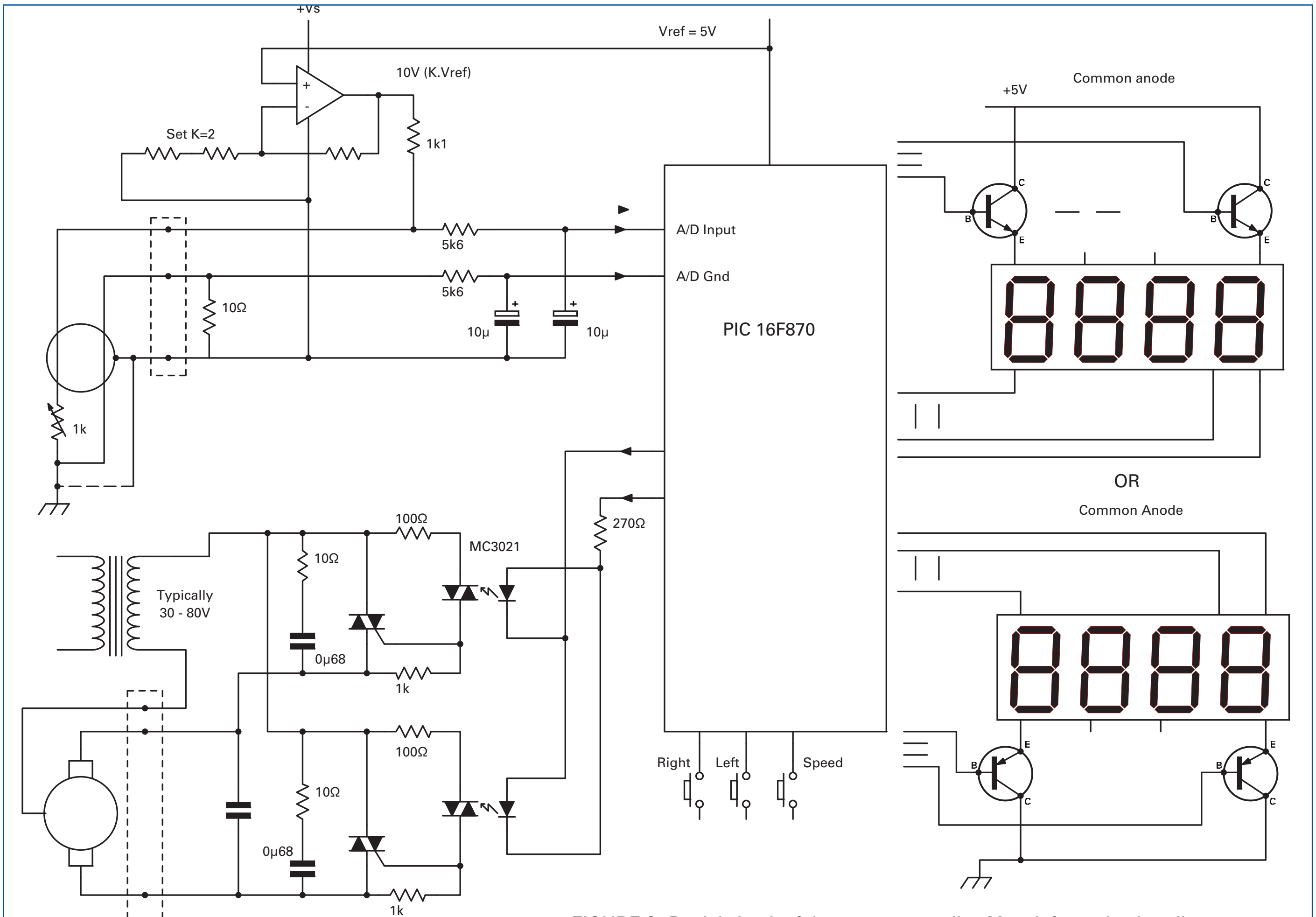


FIGURE 3: Partial circuit of the rotator controller. More information is online, see text.

perhaps by rotating to the endstop and using a compass. Then, the controller is switched on while simultaneously holding down both left and right buttons. The LED display shows the Z value used for the calculation that can be changed with the L/R buttons to that measured. The value is stored in non volatile memory by pressing the 'slow' button, whereupon rotator operation returns to normal with the new value of Z used.

This is not the place to describe the PIC routines, but full PIC source and binary code is available. PIC code, a fuller description of the hardware and a set of controller commands can be found at www.scrbg.org/g4jnt/RotatorController.zip.

The PIC software should detect automatically whether common anode or common cathode displays are used and cope with either, provided drive transistors are chosen appropriately as in Figure 3.

Table 1 shows the output from the serial interface when the rotator controller is turned on, then rotated. The calibration constants used for the internal calculations are listed, eg Z = 180.7°, and a summary of the command set for changing these. More details are supplied in the online documentation. Then, as it is turned, bearing and the *P* value are transmitted.