A Homebrew Shaft Encoder

Build a shaft encoder using parts you can salvage from dead computer mice.

By Doug Smith, KF6DX

Equipment designers often find that digital shaft encoders are more expensive than they would like. Encoders also may not be available in the angular resolutions they desire. Here is how I designed and built a tuning knob for my computer-controlled transceiver using readily available components and a little persistence.

Digital Rotation Sensors: Principles of Operation

When it comes to digital rotation sensors, the idea is to develop digital signals that indicate both direction and rate of rotation. The most common arrangement is a pair of digital signals in quadrature. That is, two digital signals having a 90° phase relationship (see Fig 1). When the shaft is rotating in one direction, signal A leads signal B; when rotating in the opposite direction, signal B leads signal A.

It is easy to determine the direction of rotation using an edge-clocked D flip-flop (see Fig 2). Applying signal A to the flip-flop's clock input and signal B to its data input, signal B is low at signal A's rising edge when rotation occurs in one direction; it is high when rotation is in the opposite direction. The flip-flop's output signal, Q, therefore indicates the direction of rotation. Such a direction signal, combined with the rate of one of the signals, A or B, yields the direction and speed of rotation.

Of Mice and Men

John Steinbeck is a better "read," but the method described above is exactly what happens inside your computer mouse or trackball. Movement of the

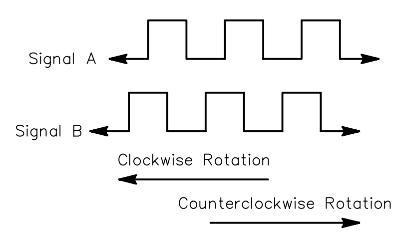

Fig 1—Desired shaft encoder output signals A and B.

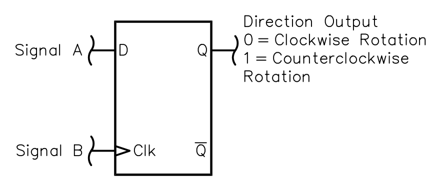

Fig 2—A rising-edge-clocked D flip-flop decodes rotation direction.

225 Main St
Newington, CT 06111-1494
kf6dx@arrl.org

pointing device causes two sets of direction and speed information to be conveyed to the computer. One set corresponds to movement left and right (X axis), the other to movement back and forth (Y axis). For radio tuning applications, only one set of information is needed, corresponding to clockwise or counterclockwise dial rotation.

It is possible to perform the function of the D flip-flop in software running on a microprocessor. Signals A and B are fed to two discrete inputs of the processor and code is written to compare their states. Problems usually arise, though, when the rotation rate exceeds the processor's ability to keep up. Backward steps and other strange things can occur when the processor is unable to read each and every transition of both signals. For that reason, mouse makers and embedded-control system designers have stayed away from decoding in software and have relied on hardware to do the job.

For a computer-controlled transceiver, a mouse chip seems an excellent choice. It does the decoding and delivers serial data commands that correspond to shaft rotation. Power consumption is so low that power may be derived from the host computer. I chose the HM8350A from Hualon Microelectronics.[1]

To get quadrature signals, a slotted wheel is generally used. Optical emitters and sensors are mounted on opposite sides of the wheel to generate signals A and B. A single infrared emitter is common, accompanied by a pair of phototransistors (see Fig 3). The distance between the phototransistors is chosen to be the product of an odd integer and one-quarter of the distance between transparent slots in the wheel so that light intensity varies in quadrature at the sensors. When the signals from the phototransistors are squared (hard-limited) inside the mouse chip, they resemble signals A and B as shown in Fig 1.

HM8350A chips, slotted wheels and photo-electronic devices may be salvaged from discarded mice that failed because of mechanical reasons. One alternate way of making a slotted wheel is described below.

The Slotted Wheel and Shaft Bearing

I wanted to use a 1/4-inch shaft for my tuning knob and the slotted wheels from mice were too small. I reasoned that with modern CAD software, it should not be too hard to create a pattern of radial segments that were alternately light and dark. Armed with a laser printer and some laser-compatible transparency film, I made my own photo-interrupter disk, shown in Fig 4.

I made the disk about 1 inch in diameter and with pairs of light and dark segments. I punched a hole through the center of the disk so that I could attach it to the end of the shaft using a regular fastener. The end of the shaft was drilled and tapped for #4-40 hardware to accept the disk (see Fig 5). A little touch-up with a fine permanent marker made the disk usable.

The shaft bearing is a critical part of tuning-knob design because any play, either back-and-forth or side-to-side, is quite undesirable. The shaft-to-bearing fit must be quite close and it was not easy finding the right shaft material. Fortunately, rod stock is available in very fine gradations of diameter. A generous blob of silicone grease helped eliminate any remaining play.

The shaft bearing I used has a 3/8-inch threaded bushing for panel mounting. A lever is used to provide variable friction on the shaft. The housing was made for this accessory from a track-lighting enclosure (see Fig 6). A heavy base was necessary to prevent the assembly from traveling across the desk when in use. The base is made from a roughly circular slug that was cut from thick steel plate at

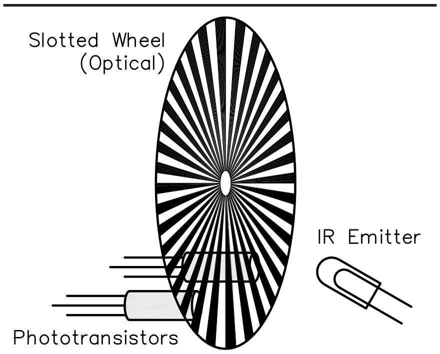

Fig 3—The general arrangement for generating signals A and B.

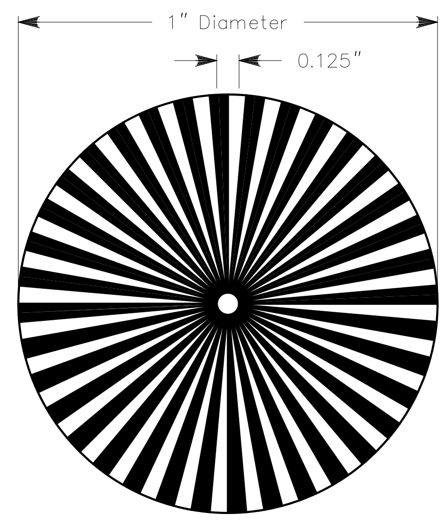

Fig 4—An opto-interrupter disk made from laser-printer transparency film.

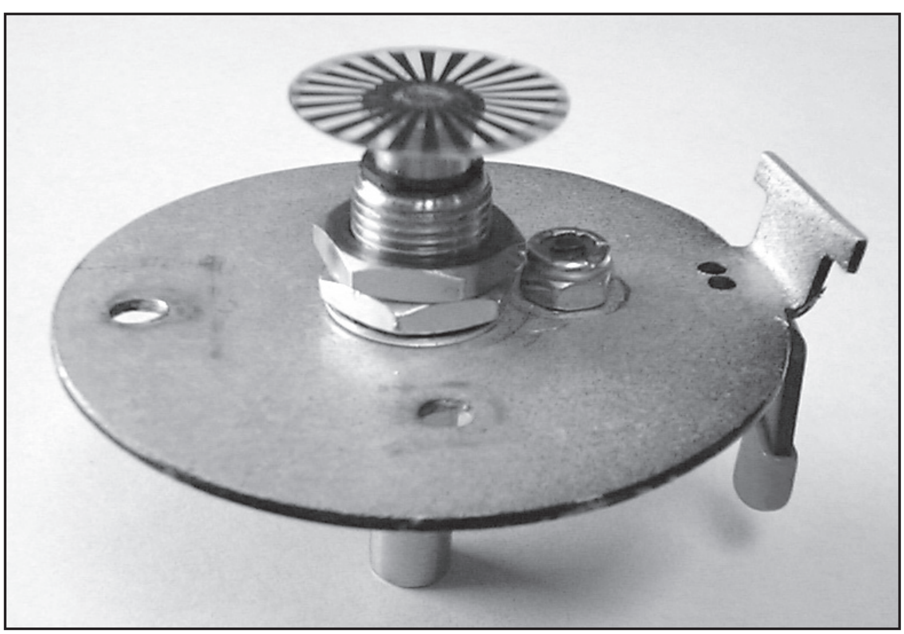

Fig 5—The disk attached to the shaft.

[1]Notes appear on page 00.

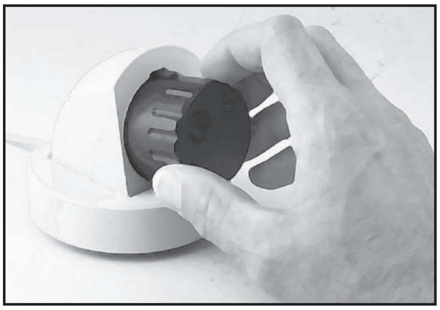

Fig 6—Completed tuning knob assembly.

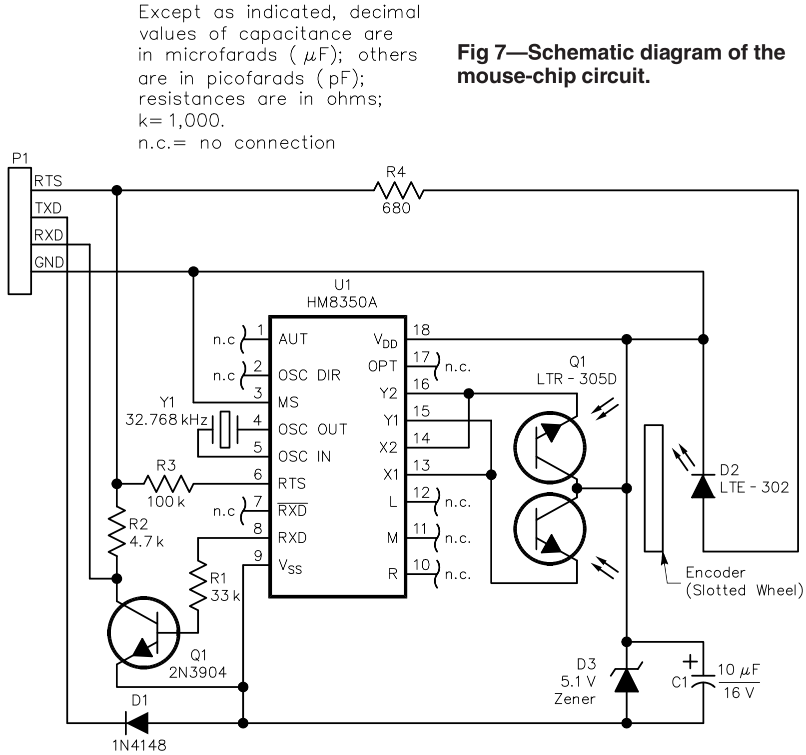

Fig 7—Schematic diagram of the mouse-chip circuit.

a construction site and machined until presentable, then painted.

A Serial-Mouse Circuit

Refer to Fig 7, a schematic diagram. The mouse chip gets its power from the computer's EIA-232 TxD line, since that line is not doing anything else in this case. Only the RxD line carries data. D1 protects against reverse polarity in case the TxD line goes to the positive rail. D3 sets the supply voltage to the mouse chip at about 5 V dc. The power for the LED comes from the RTS line, which also does not toggle.

Phototransistors Q1 are tied directly to the chip. Tuning-fork resonator Y1 provides a clock for the chip. Transistor Q2 level-shifts the R×D output to EIA-232 levels. Output comes at 1200 baud and standard mouse driver software running on a PC may be used to detect knob rotation commands from either the X or Y detector. Pins L, M and R would normally be hooked to the three SPST switches on a mouse or trackball. Those could have been used to change the tuning rate or to implement some other function, but they were left unconnected here.

Fig 8 is a close-up of the circuit assembly. The encoder disk fits between the LED and the phototransistors when assembled. The LED is an LTE-302; the phototransistor pair is a single, three-pin unit: LTR-305D.[2]

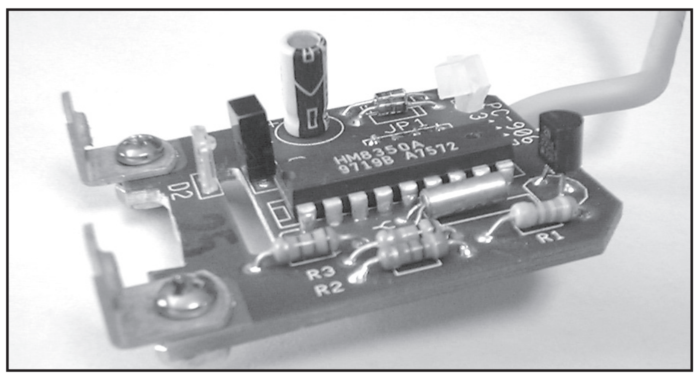

Fig 8—A close-up view of the PC assembly.

Note: Modern PC operating systems tend to interrogate devices connected to serial ports at boot time. Since my knob circuit looks just like a mouse to the PC, it is wise to connect a serial PC mouse or trackball to a lower-numbered COM port than that of the knob. Otherwise, the PC will install the knob as the pointing device and you will find the icon traveling diagonally across the screen as you rotate the knob; your mouse will be tuning your radio!

The Knob Itself

In these days of push-button, menu-driven machinery, decent tuning knobs are becoming difficult to obtain "off the shelf." On the other hand, regular milling equipment makes it easy to make a custom knob from readily available rod stock.

Armed with some three-inch black Delrin stock[3] and a ball-end milling tool, I was able to produce the massive symmetrical tuning knob shown in Fig 7. I used Delrin because it is an easy-to-machine plastic that is already black in color. Aluminum may be used and subsequently anodized if you worry about marring the Delrin under heavy use. The ball-end milling tool made it easy to put a finger hole into the front surface of the knob and to make flutes that taper along the knob's length.

Zack Lau, W1VT, has more to say about making knobs. Watch for the topic in an upcoming RF column.

Notes

[1]Hualon Microelectronics (HMC), **www.hualon.com.tw**.

[2]LiteOn, **www.liteon.com**. These particular parts may no longer be available, but equivalents are. Scrounge them from dead mice!

[3]Delrin rod stock is available from McMaster-Carr Supply Co, **www.mcmaster.com**.