CHAPTER 8

Rotary Encoders

Rotary encoders are useful for conveying position input and adjustments. For example, as a tuning knob for software-defined radio (SDR), the rotary control can indicate two things related to frequency tuning.

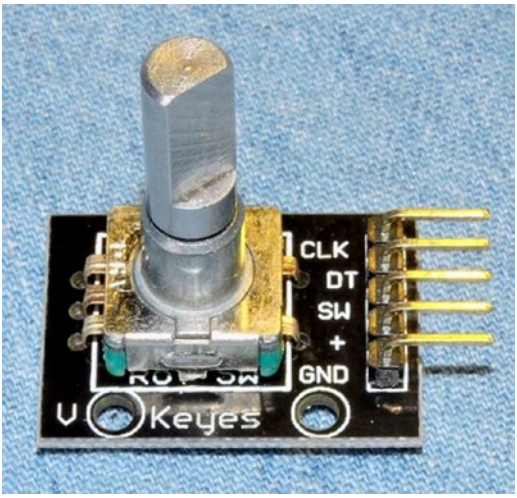
- When a step has occurred
- The direction of that step

Determining these two things, the rotary encoder can adjust the tuning in the frequency band in discrete steps higher or lower. This chapter will examine the rotary encoder with the help of the economical Keyes KY-040 device and the software behind it.

Keyes KY-040 Rotary Encoder

While the device is labeled an "Arduino Rotary Encoder," it is also quite usable in non-Arduino contexts. A search on eBay for *KY-040* shows that a PCB and switch can be purchased (assembled) from eBay for about $1.28. Figure 8-1 shows both sides of the PCB unit I purchased.

W. Gay, *Custom Raspberry Pi Interfaces*, DOI 10.1007/978-1-4842-2406-9_8



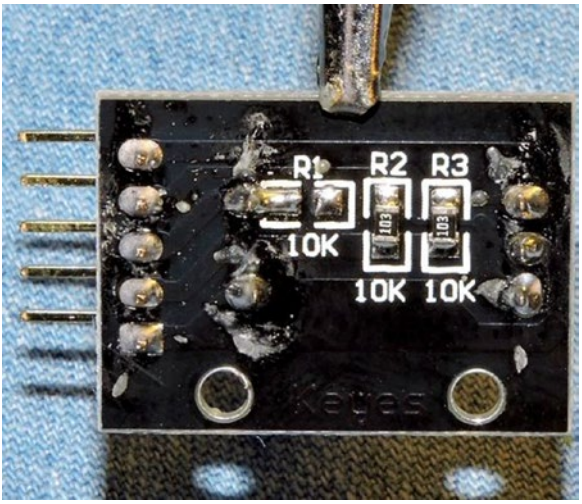


Figure 8-1. *The Keyes YL-040 rotary encoder*

It is also possible to buy the encoder switch by itself on eBay, but I don't recommend it. The PCB adds considerable convenience for pennies more than the cost of the switch. But the main reason to buy the assembled PCB unit is to get a *working* switch. I have given up on buying good rotary switches by themselves from eBay. I suspect that many eBay rotary switch offerings, if not all, are factory rejects and floor sweepings.

The Switch

To help convince you of the need for buying quality, let's take an inside look at how switches are constructed. Figure 8-2 illustrates the contact side of a switch assembly that I did an autopsy on.

Figure 8-2. *Contact side of the rotary encoder*

Figure 8-2 shows that there is a contact pad (lower left) for the wiper arm. The top portion (right side of the figure) shows the other contact points. The smear seen there is some conductive grease. Figure 8-3 illustrates the wiper half of the assembly.

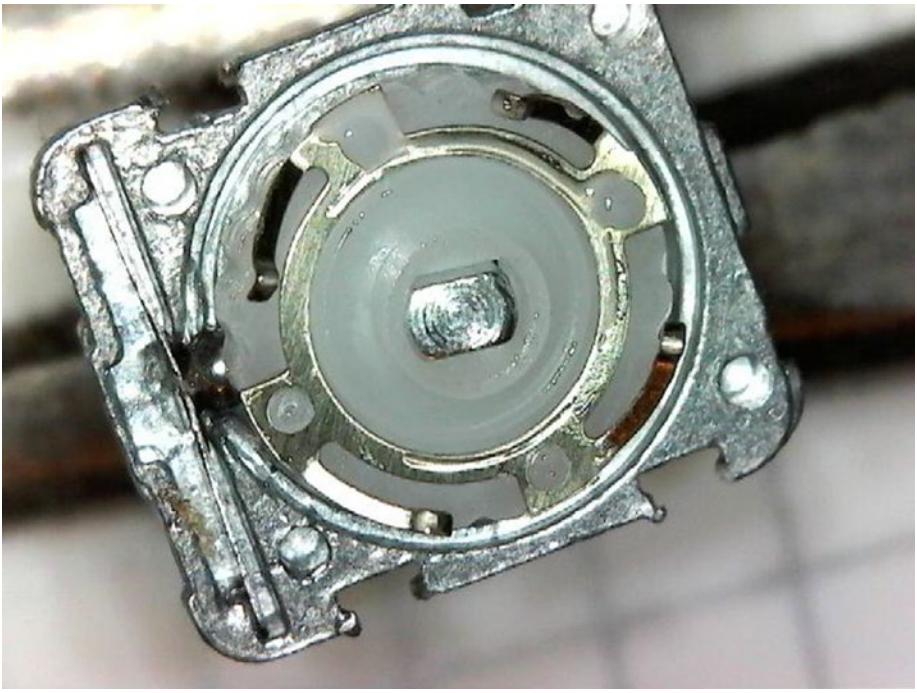


Figure 8-3. *The wiper assembly of the rotary encoder*

The photos illustrate the cheap nature of the rotary encoders' construction. They must go through some sort of quality control at the factory, but it wouldn't take much more than a bent wiper arm to render one faulty.

Figure 8-4 shows the KY-040 schematic without the optional push button switch. The units with the optional push switch have another connection labeled "SW" on the PCB. The return side of that switch connects to ground (GND).

The schematic shown focuses on the rotary switches that connect switches A and B to the common point C. The KY-040 PCB includes two 10kΩ pull-up resistors so that when either switch is open, the host will read a high (1 bit). When the A or B switch is closed, this brings the signal level low (0 bit). Finally, note that the KY-040 PCB labels the connections as CLK and DT. According to reference [1], switch A is the CLK connection, while B is the DT connection. These specifics are unimportant to its operation.

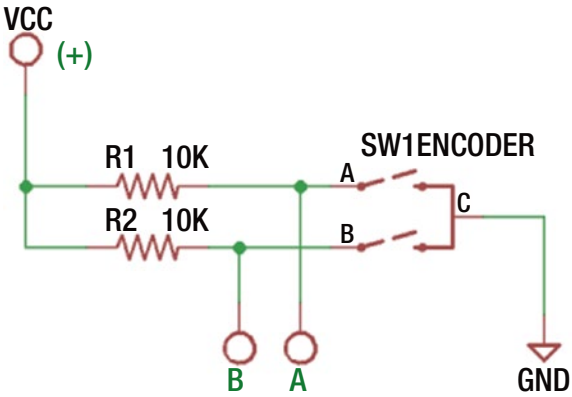


Figure 8-4. *The KY-040 schematic (excluding optional push switch)*

Operation

The rotary encoder opens and closes switches A and B as you rotate the knob. As the shaft is turns, both switches alternate between on and off at the detents. The magic occurs *in between* the detents, however, allowing you to determine the direction of travel. Figure 8-5 illustrates this effect.

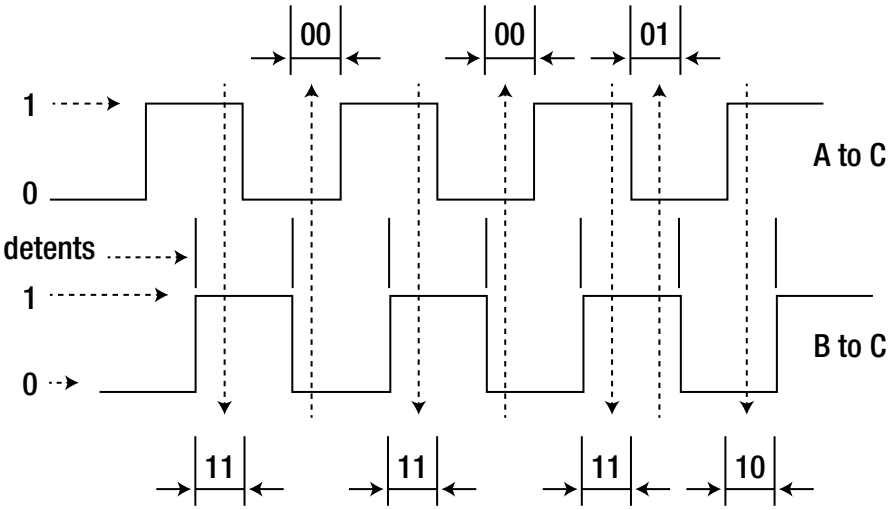


Figure 8-5. *Rotary encoder signals*

Starting from the first detent shown in Figure 8-5, switches A and B are open, reading 11 in bits (both read high). As the shaft is rotated toward the next detent, switches A and B become both closed (reading as 00). *In between* the detents however, you can see that switch A closes first (reading as 01), followed by switch B later (now reading 00). The timing of these changes allows you to determine the direction of travel. Had the shaft rotated in the reverse direction, switch B would close first (reading as 10), followed by A.

Voltage

As soon as you see the word *Arduino*, you should question whether the device is a 5V device because many Arduinos are 5V-based. For the KY-040 PCB, it doesn't matter. Even though the reference [1] indicates that the supply voltage is 5 volts, it can be supplied the Raspberry Pi level of 3.3 volts instead. The rotary encoder is simply a pair of mechanical switches and has no voltage requirement.

Evaluation Circuit

The evaluation circuit can serve as an educational experiment and double as a test of the component. It is simple to wire up on a breadboard, and if the unit is working correctly, you will get a visual confirmation from the LEDs. Figure 8-6 illustrates the full circuit. The added components are shown at the left, consisting of resistors R_3, R_4 and the pair of LEDs.

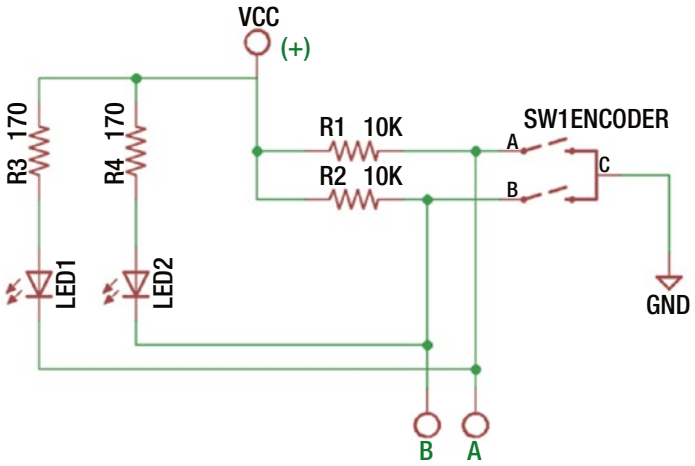


Figure 8-6. *KY-040 evaluation circuit*

The dropping resistors R_3 and R_4 are designed to limit the current flowing through the LEDs. Depending upon the LEDs you use, these should work fine at 3.3 or 5 volts. Assuming a forward voltage of about 1.8 volts for red LEDs and the circuit powered from 3.3 volts, this should result in approximately:

$$\frac{3.3-1.8volts}{170\Omega}=\frac{1.5}{170}=8.8mA$$

If you want to use less current for smaller LEDs, adjust the resistor higher than 170Ω.

Figure 8-7 illustrates my own simple breadboard test setup. You can see the dropping resistors R_3 and R_4 hiding behind the yellow LEDs that I had on hand at the moment. This test was performed using a power supply of 3.3 volts. The photo was taken with the shaft resting at a detent, which had both switches A and B closed, illuminating both LEDs.

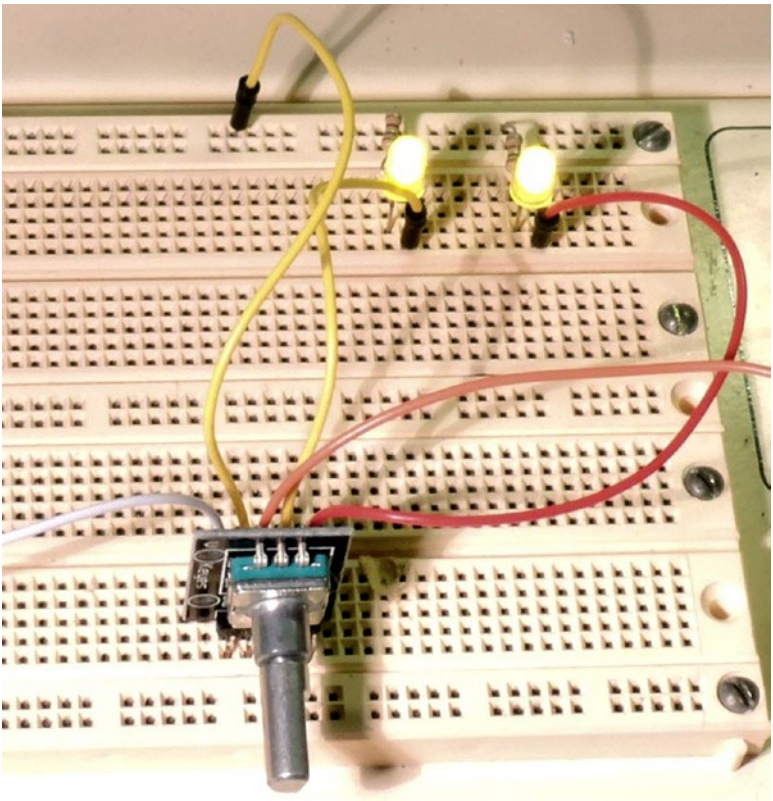


Figure 8-7. *Evaluation test of the rotary encoder*

When rotating the shaft, be sure to hold the shaft (a knob is recommended) securely so that you can see the individual LEDs turn on and off before the shaft snaps to the next detent. You should be able to see individual LED switching as you rotate.

If you see "skips" of LED on/off behavior for LED 1 and/or 2, this may indicate that you have a faulty switch. The only thing you can do for it is to replace it. But do check your wiring and connections carefully before you conclude that.