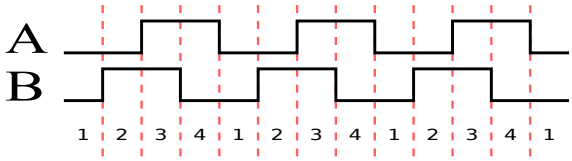
Seattle Raspberry Jam
A Raspberry Pi Meetup for Beginners to Experts

RASPBERRY JAM

Hardware Project #2: Rotary Encoder

Assembly

After ensuring that the Raspberry Pi is not powered, plug the rotary encoder circuit board into a Raspberry Pi on pins 13, 14, 21 and 22 (see the picture to the left; on the ciruit board, the side with three pins from the rotary encoder should go closest to the SD card slot). See the box to the right for more information about the rotary encoder. **WARNING! Incorrectly installing the rotary encoder could damage the Raspberry Pi. Please have one of the organisers check that the circuit board is plugged in correctly before applying power.**



How does a Rotary Encoder Work?

A rotary encoder has two pairs of contacts (called A and B), which open and close out of phase as the shaft is rotated. Depending on whether A or B changes level (rises/falls) first, the direction of rotation can be established. The diagram above shows the rotary encoder output pulses on A and B on the Y axis and phases on the X axis. In the EC11 rotary encoder (the one in this project), one detent ("click") is four phases. Other rotary encoders have a different number of phases per detent. If the rotary encoder is turned in the opposite direction to the diagram above, A would rise (and fall) before B. The EC11 also has a switch that is activated by pushing down on the shaft.

Software Setup

Now we need to setup the software on the Raspberry Pi. First, we need to configure the Raspberry Pi to read the rotary encoder and switch. We will do this by adding the some parameters to the /boot/config.txt file. Open /boot/config.txt in the nano text editor with the following command:

```
sudo nano /boot/config.txt
```

Add the following lines to the bottom of the file:

```
dtoverlay=rotary-encoder,pin_a=25,pin_b=27,steps=20
dtoverlay=gpio-key,gpio=9,keycode=30
```

Continued overleaf

Tip: For more information on a command, type man COMMAND. This opens a page of information about the command and its paramters and uses. Use the up and down arrow keys to move through the file. Type q to exit.

The first line sets up the rotary encoder overlay. The parameter *pin_a* is the BCM pin number of the A line from the rotary encoder while *pin_b* is the B line. The *steps* parameter will be explained later. The second line sets the rotary encoder's switch to output a character to the screen every time it is pressed. The parameter *gpio* is the pin number of the switch and the parameter *keycode* sets the key that will be printed, in this case, the letter a. To exit out of the editor, type Ctrl-X, then type y and finally press Enter. The software setup is now complete. Reboot the Raspberry Pi (command sudo reboot) for the changes to take effect.

Reading the Rotary Encoder

The rotary encoder can be read with evtest, which is a program that reads input devices and watches for events (in this case, the rotation of the encoder). Note: evtest is not installed by default on some versions of Raspbian, so you may need to install it with sudo apt-get install evtest -y. Run evtest with the following command.

```
evtest
```

This will list several devices and ask for which device to read events from. If you have other devices such as a keyboard or mouse attached, they will appear on the list. Find the encoder device, type the corresponding number, and press Enter (ask one of the organisers for assistance if you need it). Now rotate the rotary encoder's shaft. Output should appear on the screen showing which way the shaft is turning. Also, a counter will be increased (or decreased, depending on the direction). Note that the counter will only range from 0 to 20. This is from the *steps* parameter in /boot/config.txt. You can adjust that parameter to change the range of the counter (remember that a reboot is required for the changes to take effect). When you want to exit the evtest command, type Ctrl-C. Now we will test the button on the rotary encoder. If you **not** connected over SSH, the character "a" will be printed when you press the button. If you are connected over SSH, you can use evtest again, but this time selecting the button option.

What is SSH?

Secure SHell, or SSH for short, is a protocol for remotely accessing and controling a computer. It uses a secure channel over TCP network port 22. The command for SSH is ssh USER@IP_ADDRESS, where USER is the user you are logging in under and IP_ADDRESS is the IP address of the remote computer. The ssh application is installed by default in Raspbian, but needs to be enabled with raspi-config (under the *Interfacing Options* section). Apple Macs also already have SSH by default, but Windows users will have to use services like Putty for SSH.

```
evtest
```

Press the button to trigger the event. Congratulations! You have completed the rotary encoder tutorial.

Bonus: See if you can figure out how to control the mouse with the rotary encoder.
Hint: the parameter *relative_axis* for the *rotary-encoder* overlay in /boot/config.txt sets up the rotary encoder as a mouse. You no longer need the *steps* parameter. Note: if you are not running a graphical user interface or are accessing the Raspberry Pi over SSH, you will not see the mouse. You can use evtest to visualize the output instead.