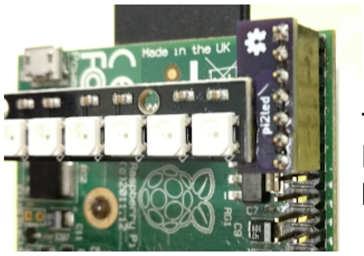
Seattle Raspberry Jam
A Raspberry Pi Meetup for Beginners to Experts

RASPBERRY JAM

Hardware Project #3: RGB LED Strip



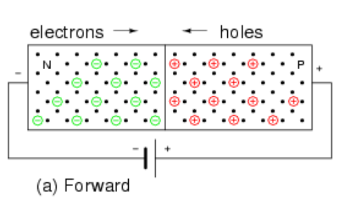
Assembly

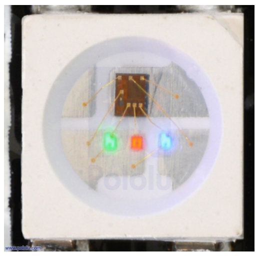
Place the LED circuit board on top of a Raspberry Pi as shown in the picture to the left. **WARNING! Incorrectly installing the LEDs could damage the Raspberry Pi. Please have one of the organisers check that the circuit board is plugged in correctly before applying power.**

Software Setup

We now need to setup the software on the Raspberry Pi. Refer to the text boxes for more information on the LEDs. The method we are using in this tutorial to control the LEDs is to utilise the Raspberry Pi's DMA controller to control the Pulse Width Modulation (PWM) peripheral to send bytes of data to the LEDs. DMA, which stands for Direct Memory Access, does block-to-block memory transfers, and therefore has the ability to directly control the Raspberry Pi's peripherals, such as PWM. Refer to the *BCM2835 ARM Peripherals* document for more information about PWM and DMA on the Raspberry Pi. This method Is better than other methods for controlling WS2812 LEDs because it leaves the Raspberry Pi's CPU free for other tasks. First, we need to disable audio output on both the HDMI port and 3.5mm audio jack on the Raspberry Pi to free the PWM

How do LEDs work?





LED is an acronym that stands for Light Emitting Diode. Diode is the keyword. Principally, diodes are designed to pass current in one direction: from the anode to the cathode. An LED is no different except for that it also emits light. To understand why LEDs emit light, we need to take a closer look at diodes. Essentially, a diode is a PN junction, which is a two-layer semiconductor made out of a single crystal (typically silicon or germanium). The two halves are the P layer, which has a surplus of electron holes (a.k.a. a positive charge, hence the "P"), and the N layer, which has a surplus of electrons, creating a negative charge. The P side is the anode, whereas the N side is the cathode. As positive voltage increases on the anode relative to the cathode, the electrons and electron holes are forced together. Once the voltage reaches a certain level, current will flow through the diode. If the current is reversed, the electrons and electron holes will be repelled from each other, and subsequently no current will flow. LEDs emit light when the electrons and electron holes combine. As each electron goes into an electron hole, energy is released as a photon. The amount of energy released determines the wavelength, and therefore the color, of the LED. A complete technical description of LEDs is beyond the scope of this article, so if you would like further information, we would recommend reading resources such as *Make: Encyclopedia of Electronic Components Volume 2.*

Continued overleaf

peripheral. We will do this by adding a line of text to /boot/config.txt. Open /boot/config.txt in the nano text editor with the following command:

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

Add the following line to the bottom of the file:

```
dtparam=audio=off
```

This tells the Raspberry Pi to not enable audio when booting. Exit nano by typing Ctrl-X, then y and finally Enter. Reboot the Raspberry Pi with the command sudo reboot for the changes to take effect.

WS2812 Addressable RGB LEDs

The LEDs we are using for this tutorial are WS2812 addressable RGB LEDs. Inside each LED chip there are three LEDs (one red, one green and one blue) and a controller. These chips have four (or six, depending on the exact model) pins. Essentially, each LED has a data input, a data output and power pins. When these chips are placed in a strip, such as the one used in this tutorial, they are chained together with the data output from one chip going to the data input from the next chip. Data from a controller (a Raspberry Pi, for example) is passed into the data input pin on the first LED. Each LED then takes the first 24 bits of the data (8 bits for green, 8 bits for red and 8 bits for blue, in that order) and outputs the remaining data on its data output pin. This process repeats until all of the LEDs are lit. For more information, we would recommend the page on Tim's Blog about WS2812 LEDs.

More Software Setup...

We have written some code to control the LEDs. The files you will need are called *pi2ws2812pwm.c, mailbox.c, mailbox.h and Makefile*. The first file is our code; it controls DMA and PWM to write data to the LEDs. The next two were created by Broadcom, and serve some functions for our code such as communication between programs. The last file, Makefile, was written by one of our members. It makes compiling the code easier. You can download these files from sites.google.com/site/mincepi/pi2ws2812/files . Before we can run the program to turn on the LEDs, these files need to be compiled. Run the command below to compile the files:

```
make
```

This will create an executable file called *pi2ws2812pwm*. If you want to remove this file after you are done, run make clean. Now run the following command to setup a device file for the LEDs:

```
sudo ./pi2ws2812pwm 8 &
```

This command creates a file called *pi2ws2812pwm* in the */dev* directory and waits for data to be written to it. The number after the command is the number of LEDs to control, in this case 8. The amperstand makes the program run in the background, so you are returned to the prompt right after running the command.

Lighting the LEDs

Finally, we can make some light! First, run this command to gain root priveleges:

```
sudo su
```

Now run the following command to light the LEDs in a rainbow pattern:

```
echo "0 255 0 165 255 0 255 255 0 128 0 0 0 0 255 0 75 130 130 238 238 190 250 190" > /dev/pi2ws2812pwm
```

The LEDs should immediately light up. The format of the above command is echo "LED1$_{green}$ LED1$_{red}$ LED1$_{blue}$ LED2$_{green}$ LED2$_{red}$..." > /dev/pi2ws2812. echo is a command that prints text to a file or to the terminal. The > operator in the command above redirects the output of the command before it, echo, to the file after it, */dev/ws2812*. Congratulations! You have completed the RGB LED strip tutorial.

Photo Credits: Diode Diagram: Allaboutcircuits, WS2812B LED: Pololu