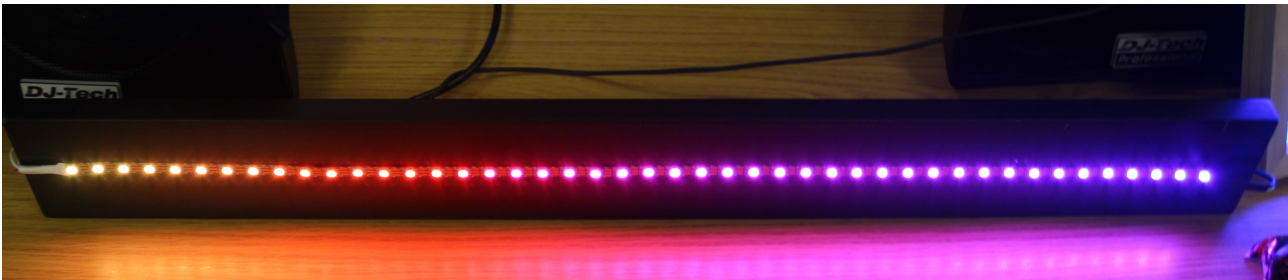


Getting Started with NeoPixels

NeoPixels are individually addressable multicoloured LEDs. In this exercise you will light up two breadboard NeoPixels. These are also available as rings, strings and as a matrix display. The Breadboard NeoPixels are supplied without connecting pins and normally need to have header pins soldered to them first.

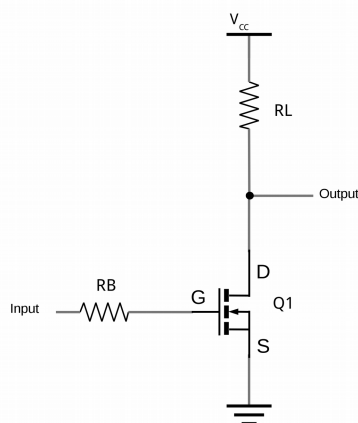
You can learn about soldering at: <http://www.penguintutor.com/electronics/soldering> or you can buy Through-hole NeoPixels which don't need any soldering available on a similar worksheet at <http://www.penguintutor.com/neopixels>



Voltage Level Shifter

To drive NeoPixels from a Raspberry Pi then you will need to increase the output voltage from the GPIO port from 3.3V to 5V. This can be achieved using a MOSFET as a voltage level shifter. A MOSFET is a type of transistor, which is switched on by applying an input voltage to the gate (G) which will allow a current to flow between the drain (D) and source (S).

The circuit diagram is shown below:



Here the MOSFET acts as a digital switch, which is turned on when the input is high and off when the input is low. The output is taken from above the MOSFET so when it is switched on then the output goes low and when the MOSFET is switched off then the output is pulled high through resistor RL. This means that the voltage shifter is inverting (you get the opposite at the output compared with the input). This is something we can handle in the software later.

<http://www.penguintutor.com/neopixels>

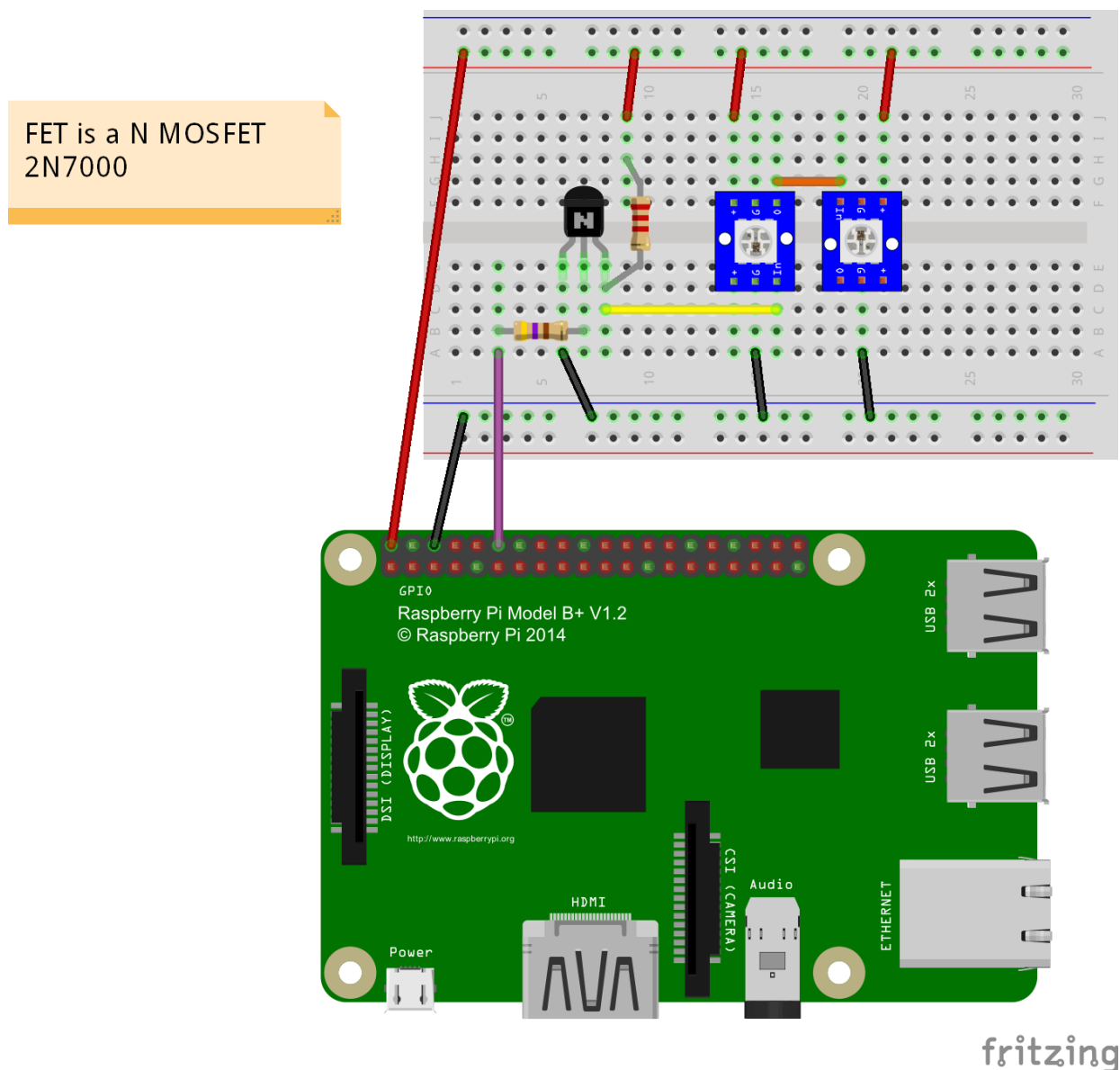
Creating the Electronic Circuit

In addition to the Raspberry Pi the hardware components used are as follows:

- RL – 2.2k Ω (Red, Red, Red, Gold)
- RB – 470 Ω (Yellow, Violet, Brown, Gold)
- Q1 - 2N7000
- 2 x Breadboard NeoPixel modules

You will also need a breadboard and some jumper wires, using male-to-male for the connections within the breadboard and male-to-female for the connections to the Raspberry Pi.

You need to wire the circuit up as shown in the diagram below. Note that this uses the power from the Raspberry Pi, which will work when using only one or two NeoPixels, but an external supply is required if more NeoPixels are used.



The circuit diagram is based on Breadboard NeoPixels from AdaFruit. Other manufacturers use a different pin layout, in which case the wires may need to be connected differently. The input to the NeoPixel may be labelled as Data In, DI or In, the output to the next stage may be labelled Data Out, DO or out. The supply voltage may be labelled as +V, +5V or 5V, the ground connection as -V, 0V, G or Gnd. Please check that you wire the appropriate pins up correctly.

The two NeoPixels are mounted the opposite way around from each other. This is so that the output from the first is close to the input for the second NeoPixel.

Disabling the Raspberry Pi Audio Driver

Before you get around to installing the software there is an issue when trying to control NeoPixels from a Raspberry Pi where sound is enabled. This is because the hardware PWM feature of the Raspberry Pi has to be manipulated by the software library, but this is also used by the sound driver. The solution is to disable the sound driver, which unfortunately means not being able to use sound.

To disable the sound driver create a new blacklist file using

```
..... sudo nano /etc/modprobe.d/snd-blacklist.conf
```

Add an entry

```
..... blacklist snd_bcm2835
```

Then save and exit (Ctrl-O Ctrl-X)

Adding

The next step is to install the python module (software library) that allows us to control the NeoPixels from the Raspberry Pi. It is recommended to use a version of the NeoPixel library that has been modified so that it works on all versions of the Raspberry Pi (the original AdaFruit library only works with the original Raspberry Pi 1). The Python module is not included in the Raspberry Pi repositories and needs to be compiled from source code.

You will need some developer libraries which will allow you to compile the software. This is installed using the normal Raspberry Pi installer as below:

```
sudo apt-get install scons swig
```

Next download the ws281x library code from github using the clone command, which copies all the source code to your local computer.

```
git clone https://github.com/jgarff/rpi\_ws281x.git
```

Change to that directory and run scons to compile the software.

```
cd rpi_ws281x
```

scons

You then need to change to the python directory and install the Python module from there:

cd python

Next install the Python 3 module file using

sudo python3 setup.py install

There are some test programs in that directory which you can try, but I suggest you use the following simple test script using Python 3. I recommend you use IDLE 3 for creating the file (which includes syntax highlighting), but you should run it from the command-line rather than in IDLE.

```
#!/usr/bin/python3
from neopixel import *
import time

LEDCOUNT = 2          # Number of LEDs
GPIOPIN = 18
FREQ = 800000
DMA = 5
INVERT = True         # Invert required when using inverting buffer
BRIGHTNESS = 255

strip = Adafruit_NeoPixel(LEDCOUNT, GPIOPIN, FREQ, DMA, INVERT, BRIGHTNESS)
# Initialize the library (must be called once before other functions).
strip.begin()

while True:
    # First LED white
    strip.setPixelColor(0, Color(255,255,255))
    strip.setPixelColor(1, Color(0,0,0))
    strip.show()
    time.sleep(0.5)
    # Second LED white
    strip.setPixelColor(0, Color(0,0,0))
    strip.setPixelColor(1, Color(255,255,255))
    strip.show()
    time.sleep(1)
    # LEDs Green
    strip.setPixelColor(0, Color(255,0,0))
    strip.setPixelColor(1, Color(255,0,0))
    strip.show()
    time.sleep(0.5)
    # LEDs Red
    strip.setPixelColor(0, Color(0,255,0))
    strip.setPixelColor(1, Color(0,255,0))
    strip.show()
    time.sleep(0.5)
    # LEDs Blue
    strip.setPixelColor(0, Color(0,0,255))
    strip.setPixelColor(1, Color(0,0,255))
    strip.show()
    time.sleep(1)
```

Save this as file neopixel.py
Set executable permission using:

```
chmod +x neopixel.py
```

It needs to be run with root permission using sudo

```
sudo ./neopixel.py
```

In the code the first part of the program sets up the appropriate values for the Raspberry Pi and the NeoPixels, this may need to be updated particularly if there are more NeoPixels than the two we have specified. Note that the INVERT option is set to True which is required due to the inverting nature of the circuit we have created.

One final thing is that for some NeoPixels the red and green are handled in a different order. The above code is based on GRB (Green Red Blue) ordering which is used by the breadboard NeoPixels I used, but many NeoPixels instead use a more traditional RGB (Red Green Blue) ordering in which case the colour ordering needs to be changed.

A graphical user interface is available from:
<https://github.com/penguintutor/neopixel-gui>

For more details see:
<http://www.penguintutor.com/neopixels> or follow @penguintutor

This worksheet is based on a project from *Learn Electronics with Raspberry Pi*,
by Stewart Watkiss
Available as printed book or ebook from www.apress.com or through book stores.

ISBN 978-1-4842-1897-6

