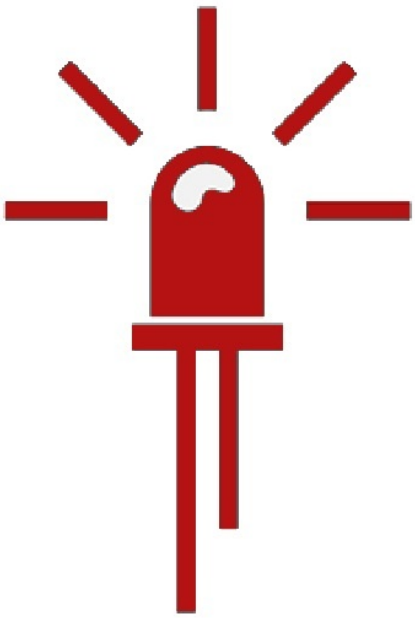
Colored Lights using NeoPixels

In this guide, we will explore different lighting patterns as well as some casing designs that could be used to light a room.

Written By: Davidson Theosmy



© 2024

INTRODUCTION

This tutorial will cover the basics of setting up and using a neopixel with eight pixels to apply different color patterns. For more information about neopixel, please visit the link below.

Adafruit

TOOLS:

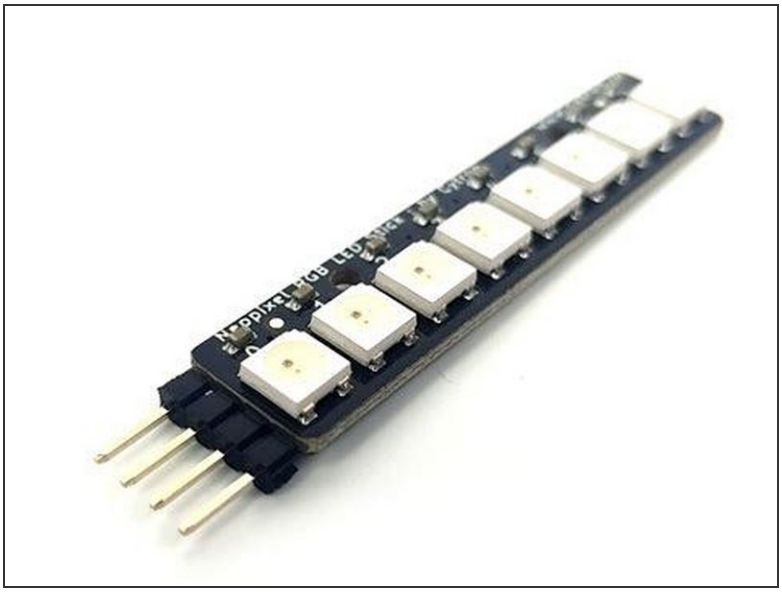
- Micro-USB Cable (1)

PARTS:

- Feather M4 Express (1)
- NeoPixel Stick (1)
- Breadboard Jumper cables (1)

© 2024

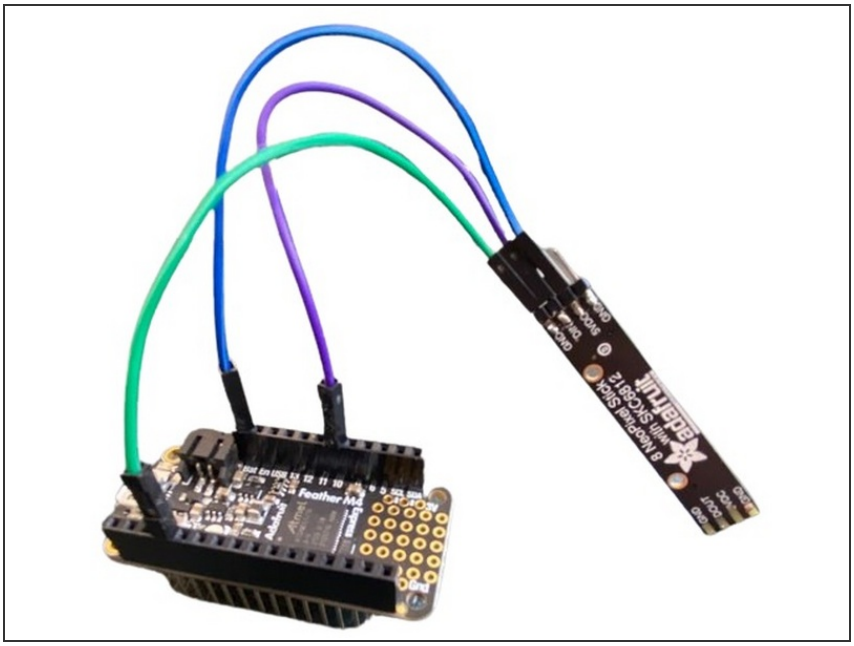
Step 1 — Setup





- Make sure you've previously completed the Feather M4 Express tutorial. We will be using it to address the NeoPixel in this tutorial.
- Additionally, you will need a NeoPixel stick and three (3) male-to-female jumper cables to interact with the Feather.

Step 2 — Connection



- To connect your NeoPixel to your Feather, first plug in the female end of your jumpers to the male prongs that should be soldered on one side of the NeoPixel. Check the pinouts of the Feather.

- The **GND** pins on either end of the NeoPixel go to ground. You only need to plug in to one or the other, not both. Connect one jumper wire to a **GND** male connector and use

© 2024

the male end of your jumper wire to plug into a header marked **Gnd** on the board.

- The **5VDC** pin is where the NeoPixel gets its power. Connect a jumper wire between the **5VDC** pin and the header marked **Bat** on the Feather. **Bat** supplies 5 volt power, while **USB** and **3V** do not supply the correct voltage.
- The **DIN** pin is where the NeoPixel receives its instructions from the Feather. It can be connected to any digital output pin on the Feather, simply marked 13-5 on the PCB. For the example code, we will use the header marked **10** on the board and **D10** in code.

Step 3 — Code

```
import time
import board
import digitalio
import neopixel

num_pixels = 8
pixels = neopixel.NeoPixel(board.D10, num_pixels)
pixels.fill((0, 0, 0))
while True:
    pixels.fill((255,0,0))
    time.sleep(.5)
    pixels.fill((0,0,0))
    time.sleep(.5)
```

- Open the MU editor and import the necessary libraries, for this project the required libraries are linked here. Make sure the version you download matches the one that is installed on your Feather m4

- Set the number of pixels by assigning it to a variable like **num_pixels = 8**. Next, create a variable that will store the values of the neopixel stick. It will look similar

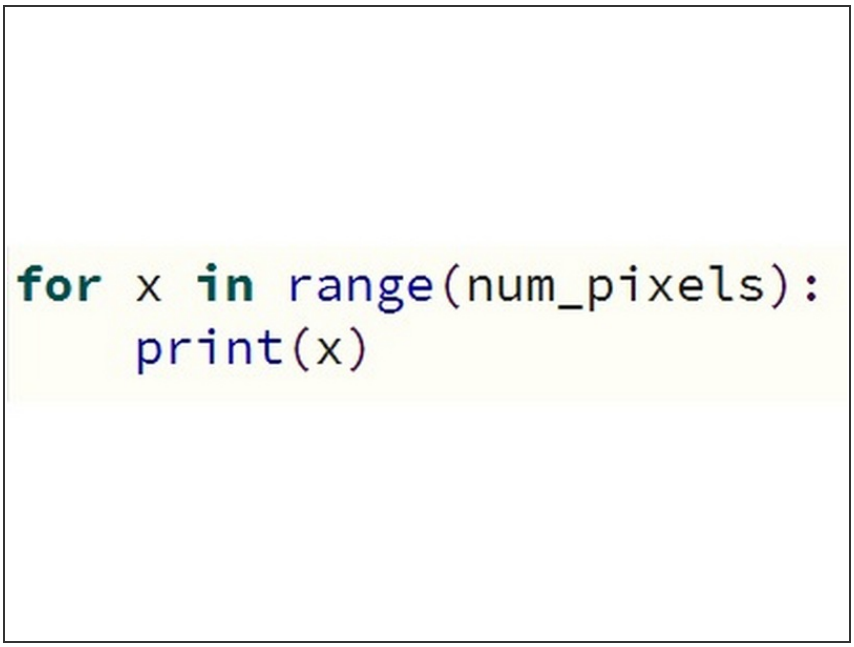
© 2024

to **pixels = neopixel.NeoPixel(board.D10, num_pixels)**.

- Now the neopixels can be set using the variable **pixels** that you created.
- You can address a single pixel by using its index, which counts up to the total number of pixels. Remember that this is in Python, so the counting starts at 0. For a stick of 8 pixels, the index will be 0-7. To address a single pixel, use the form **pixels[*index*] = (*R*,*G*,*B*)**
- You can also address every pixel at once, using the form **pixels.fill((*R*,*G*,*B*))**.
- Use the sample code in the image to apply this knowledge and test that your NeoPixel is hooked up correctly. For more functions available with the NeoPixel, check out this link.

© 2024

Step 4 — Challenge



- To earn a badge for the NeoPixel, create a scrolling colored pixel that clears behind itself. You will need the same time functions and while loop that were included in the Feather tutorial. You may also likely use a for loop.
- The form for a loop as such is **for *x* in *variable*:** This format will let you refer to *x* inside the loop and automatically iterate *x* by 1 each time the loop occurs.
- For the final challenge, use the NeoPixel to create a bouncing pattern with LEDs. Remember this will need to have some changed variables. If you need more information or documentation, visit Adafruit's guide to NeoPixels.

The NeoPixel has many different applications and can be used to create lighting in versatile situations. Check out some of the projects we have in Technology Kits for some ideas of what you can do.