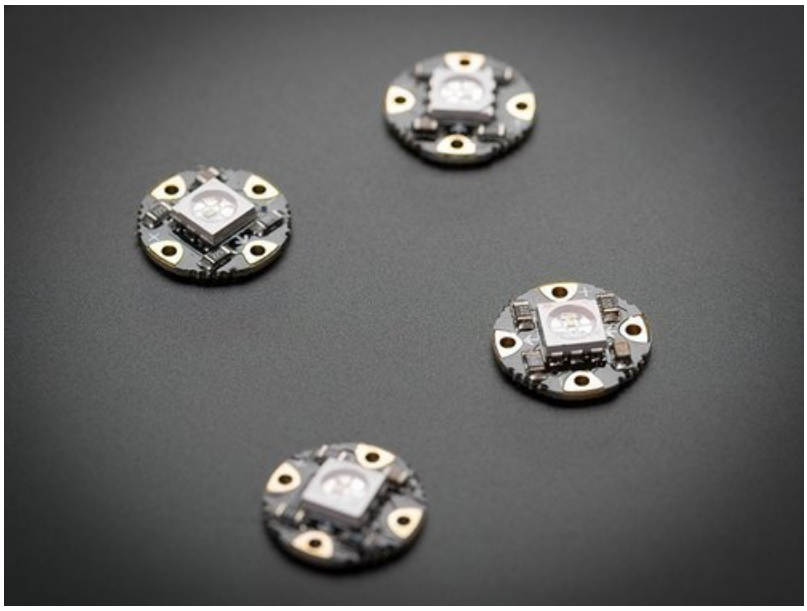
adafruit learning system

Sewable NeoPixels

Created by Justin Cooper



Last updated on 2019-01-25 06:29:06 PM UTC

Guide Contents

Guide Contents 2
Overview 3
- Prerequisite guides 4
- Lots of Pixels? 4

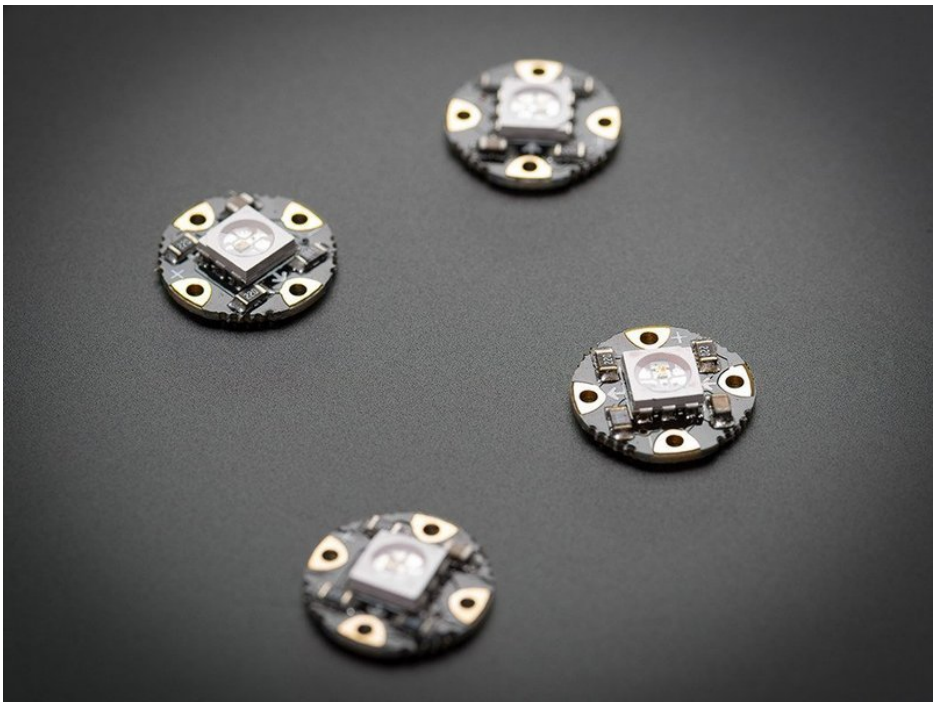
Hook up alligator clips 6
Code with MakeCode 8
Code with Arduino 10
- Install the NeoPixel Arduino library 10
- Now upload! 10

Sewing more pixels 13
Project Ideas 18
Downloads 19
- Files 19
- NeoPixel Schematic 19
- Fabrication Print 19

Overview

What's a wearable project without LEDs? These tiny smart pixels are our favorite thing. Designed specifically for wearables, we found the brightest RGB LEDs available (an eye-blistering ~3800mcd) (http://adafru.it/619) and paired them with a constant-current driver chip. The contacts are easily sewn with conductive thread. Use this guide to test your first pixel and start on a blinding wearables project with Circuit Playground Express, FLORA or GEMMA!

Pictured above: LED Ampli-Tie (https://adafru.it/doN)



Prerequisite guides

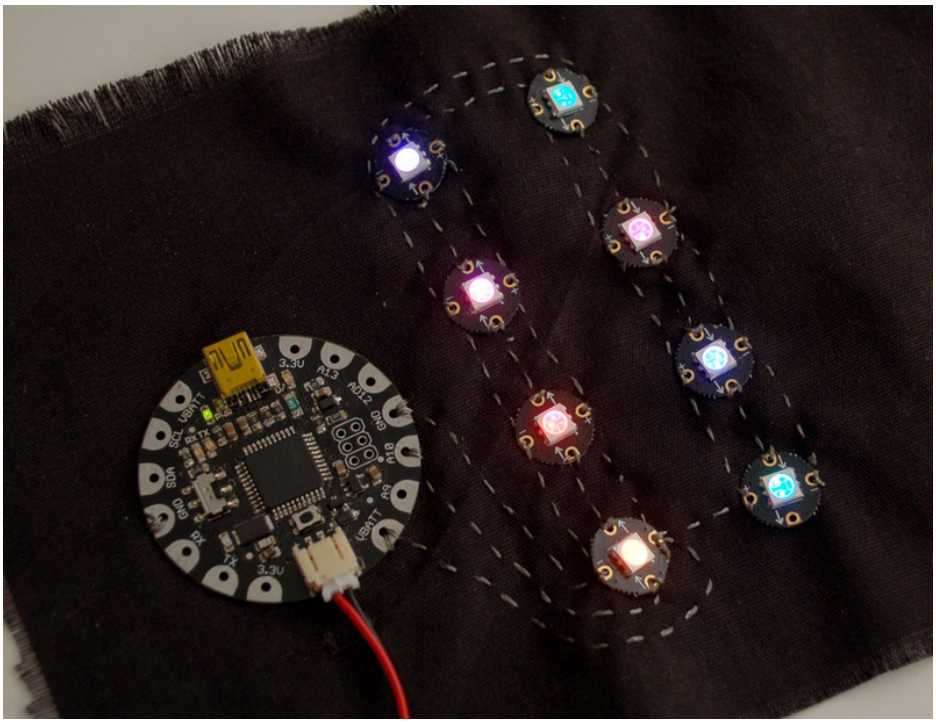
Before you begin, familiarize yourself with the following tutorials:

- Adafruit Circuit Playground Express (https://adafru.it/adafruit-cpx)
- Getting Started with FLORA (https://adafru.it/dwi)
- Introducing GEMMA (https://adafru.it/dgH)
- NeoPixel Uberguide (https://adafru.it/dhw)
- Conductive Thread (https://adafru.it/dn3)

Lots of Pixels?

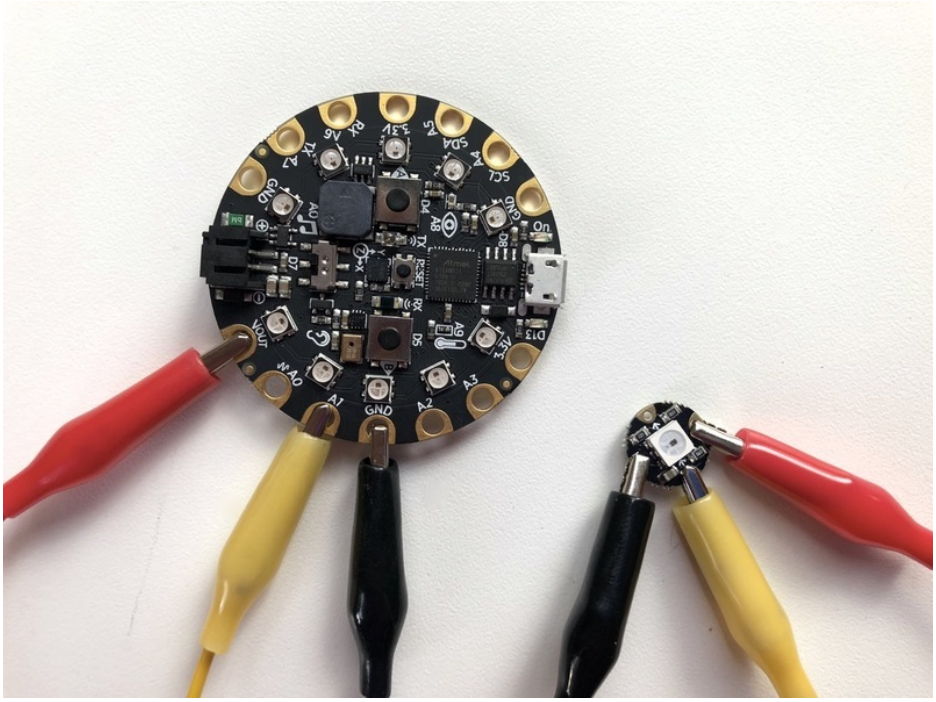
Each pixel draws as much as 60mA (all three RGB LEDs on for full brightness white). In theory, FLORA can drive up to 500 pixels at 30 FPS (above which it will run out of RAM). Circuit Playground Express can drive more than that. However, above about 20 pixels (and/or if the overall length of conductive thread exceeds ~6 feet/2 meters), the nontrivial resistance of the thread adds up and can affect the power supply. For large quantities of pixels over 20 or if you need to insulate your circuit, you should upgrade to silicone coated stranded core wire (https://adafru.it/Cc0), which will provide better conductivity for the pixels - the current draw will add up fast!

> This tutorial will cover the current v2 pixels as well as the older v1 pixels. The sewing part is the same for both versions, but the library code is different because the controlling chipset has been upgraded in v2. Pixels purchased after February 2013 are v2.



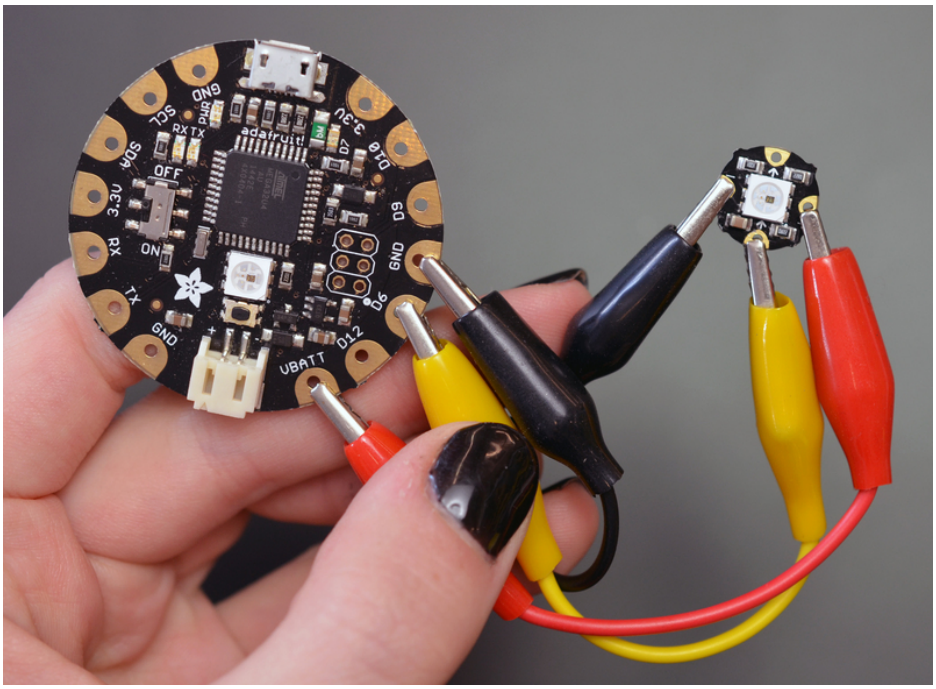
Pictured above: a FLORA prototype with v1 pixels

Hook up alligator clips

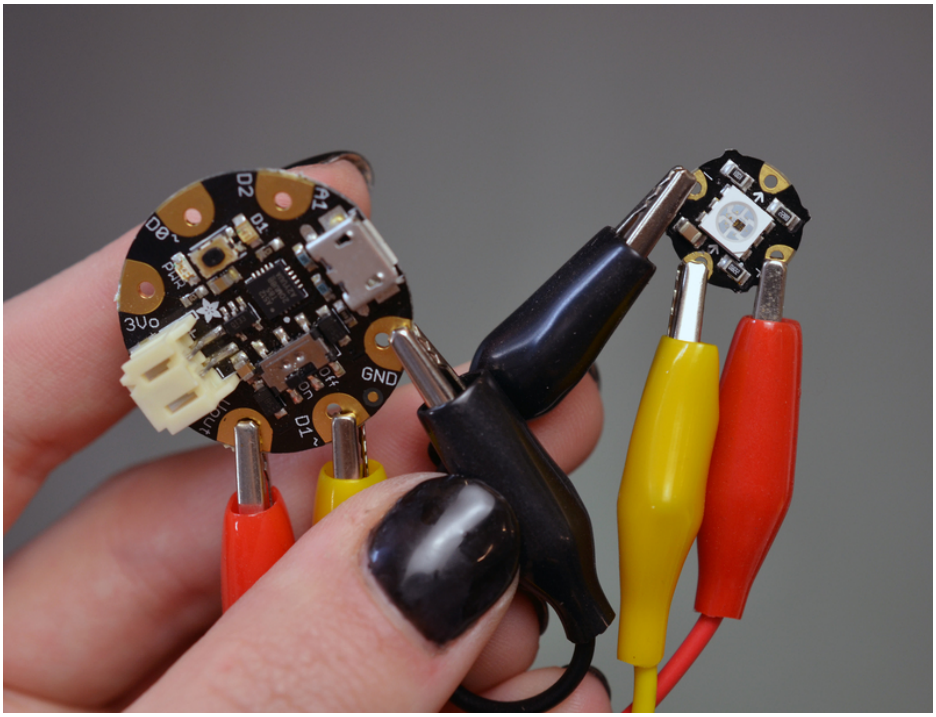


Hook up three alligator clips to your Circuit Playground Express as shown: one to **VOUT**, one to **A1**, and one to **GND**. I used a red wire for power and a black wire for ground for clarity, but it doesn't matter what color you use.

Hook up the other ends of your alligator clips to a single pixel. **VOUT** (red) connects to the + on the pixel, **GND** (black) to the **-**, and **A1** (yellow) to the pad marked with an arrow pointing towards the LED on the tiny board (not away from it).

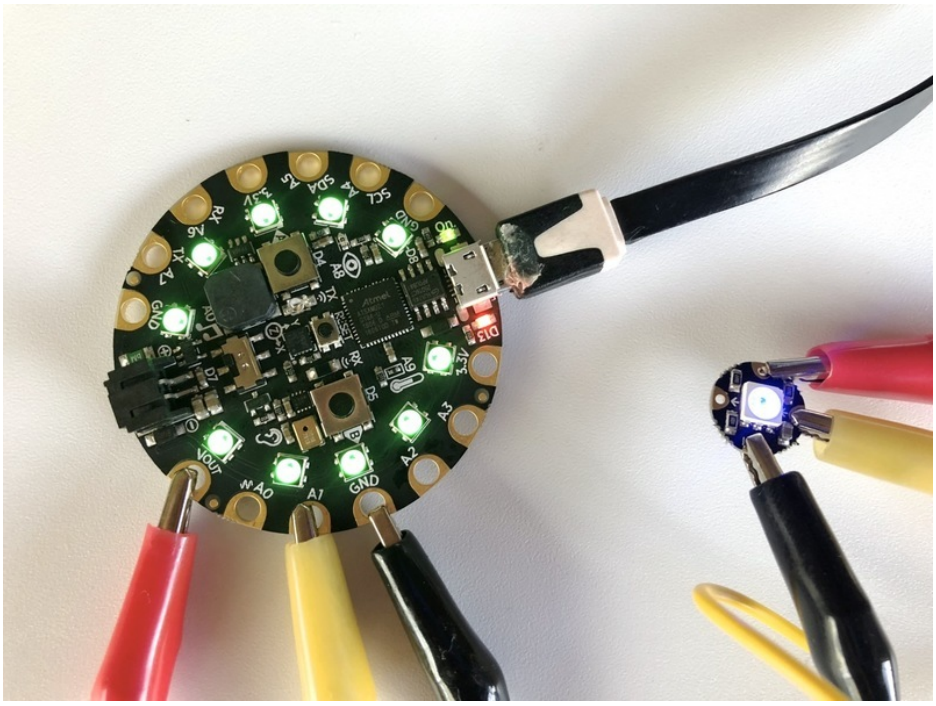


Wiring on a FLORA is very similar. Just wire to **D6** instead of **A1**. **VBATT** and **VOUT** are more or less the same, for our purposes.



On GEMMA or Gemma M0, the wiring is very similar. Hook up **VOUT** (red in the picture above) to **+**, **GND** to **-** (black above), and **D1** to the inward facing arrow (yellow above).

Code with MakeCode



MakeCode (https://adafru.it/Dyi) is Microsoft's drag-and-drop code editor, and it's about the easiest way to get your pixels showing color.

Click the appropriate button below to go to the MakeCode **strandtest** project. Click the Download button on that page, and a file called **circuitplayground-strandtest.uf2** (for Circuit Playground Express) or **maker-strandtest.uf2** (for Gemma M0) will download to your computer.

Plug your Circuit Playground Express into your computer via a USB port and double-click the tiny "reset" button in the center of the board. All the onboard lights will turn green, and a drive will appear on your computer called **CPLAYBOOT** or **GEMMABOOT**. Drag the file you just downloaded onto this drive.

The pixel you have connected will light up in a succession of colors followed by a rainbow animation.

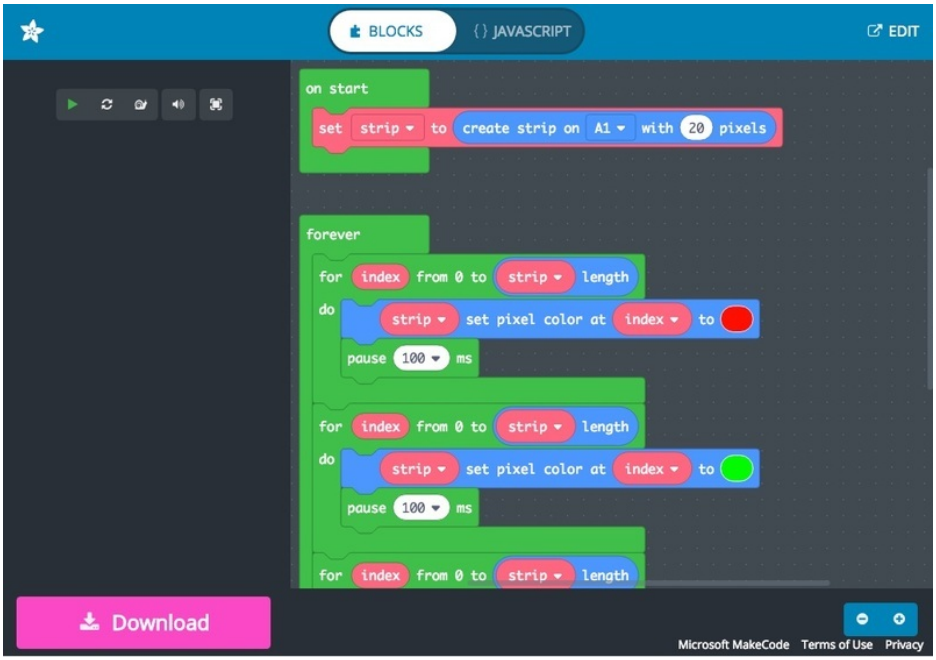
https://adafru.it/DIZ

https://adafru.it/DIZ

https://adafru.it/DI-

https://adafru.it/DI-

Note: MakeCode will NOT work on a FLORA or GEMMA v1 board. For those older boards you must use the Arduino code method. However, Arduino will work on Circuit Playground Express or Gemma M0 as well as the older boards.


BLOCKS
{ } JAVASCRIPT
EDIT
on start
set strip to create strip on A1 with 20 pixels
forever
for index from 0 to strip length
do
strip set pixel color at index to
pause 100 ms
for index from 0 to strip length
do
strip set pixel color at index to
pause 100 ms
for index from 0 to strip length
Download
Microsoft MakeCode
Terms of Use
Privacy

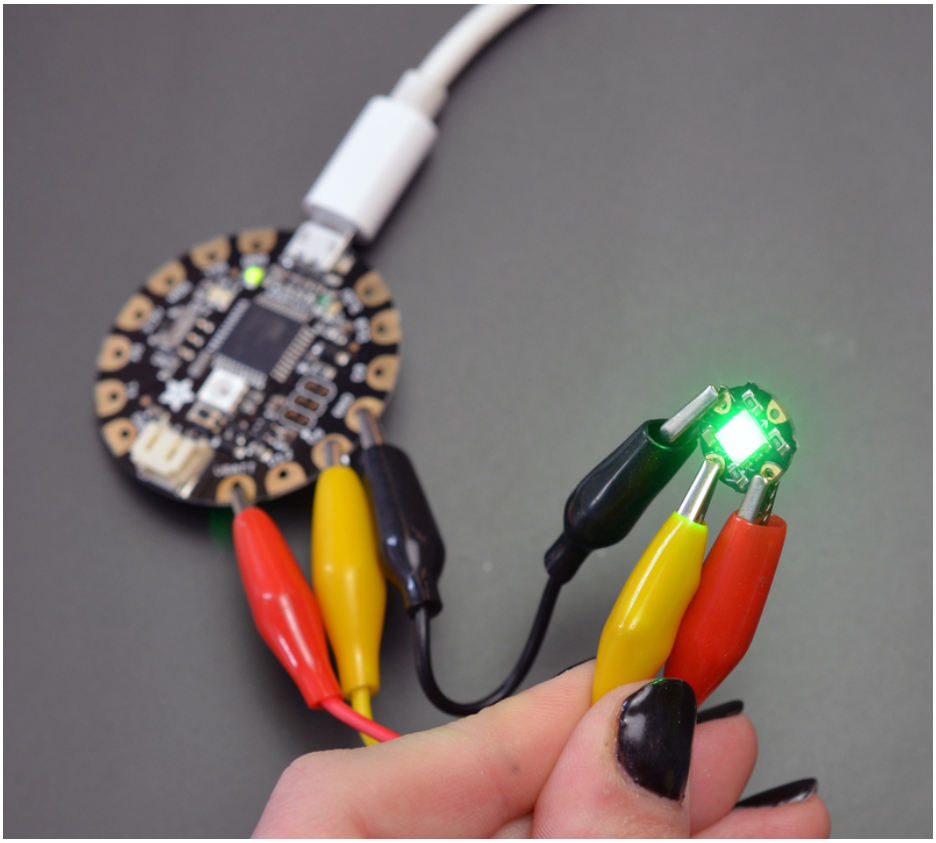
Code with Arduino

Install the NeoPixel Arduino library

Installation of the library is as follows:

1. Visit the Adafruit_NeoPixel library page (https://adafru.it/aZU) at Github.com.
2. Select the “Download ZIP” button, or simply click this link (https://adafru.it/cDj) to download directly.
3. Uncompress the ZIP file after it’s finished downloading.
4. The resulting folder should contain the files “Adafruit_NeoPixel.cpp”, “Adafruit_NeoPixel.h” and an “examples” sub-folder. Sometimes in Windows you’ll get an intermediate-level folder and need to move things around.
5. Rename the folder (containing the .cpp and .h files) to “Adafruit_NeoPixel” (with the underscore and everything), and place it alongside your other Arduino libraries, typically in your (home folder)/Documents/Arduino/Libraries folder. Libraries should not be installed alongside the Arduino application itself.
6. Re-start the Arduino IDE if it’s currently running.

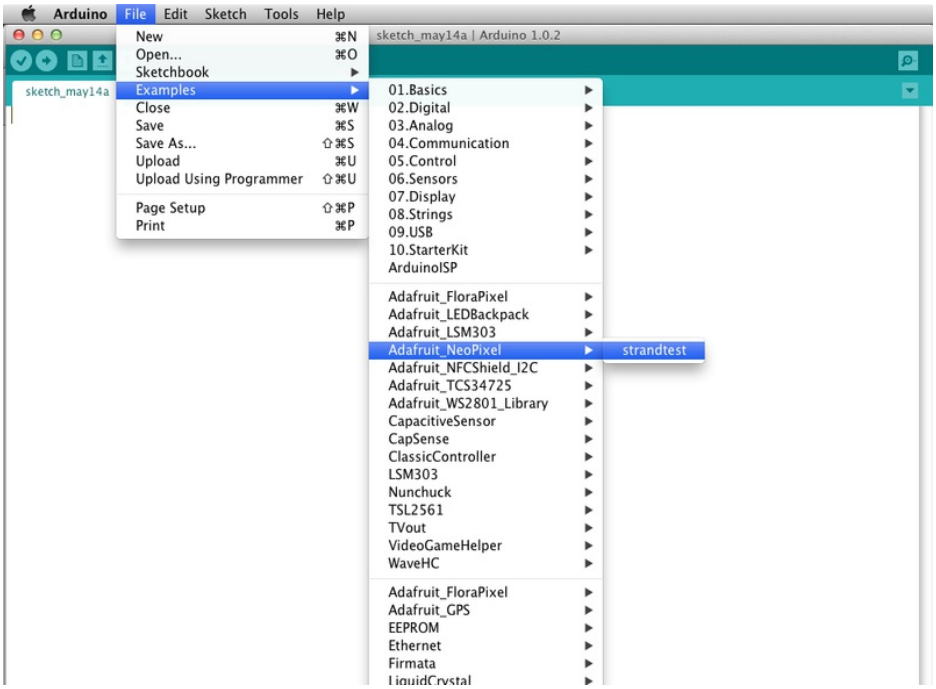
Here’s a tutorial (https://adafru.it/aYM) that walks through the process of correctly installing Arduino libraries.



Connect your FLORA and computer with a USB cable.

Now upload!

Install the NeoPixel library (https://adafru.it/aZU) and go to **File --> Examples --> Adafruit_NeoPixel --> strandtest.**

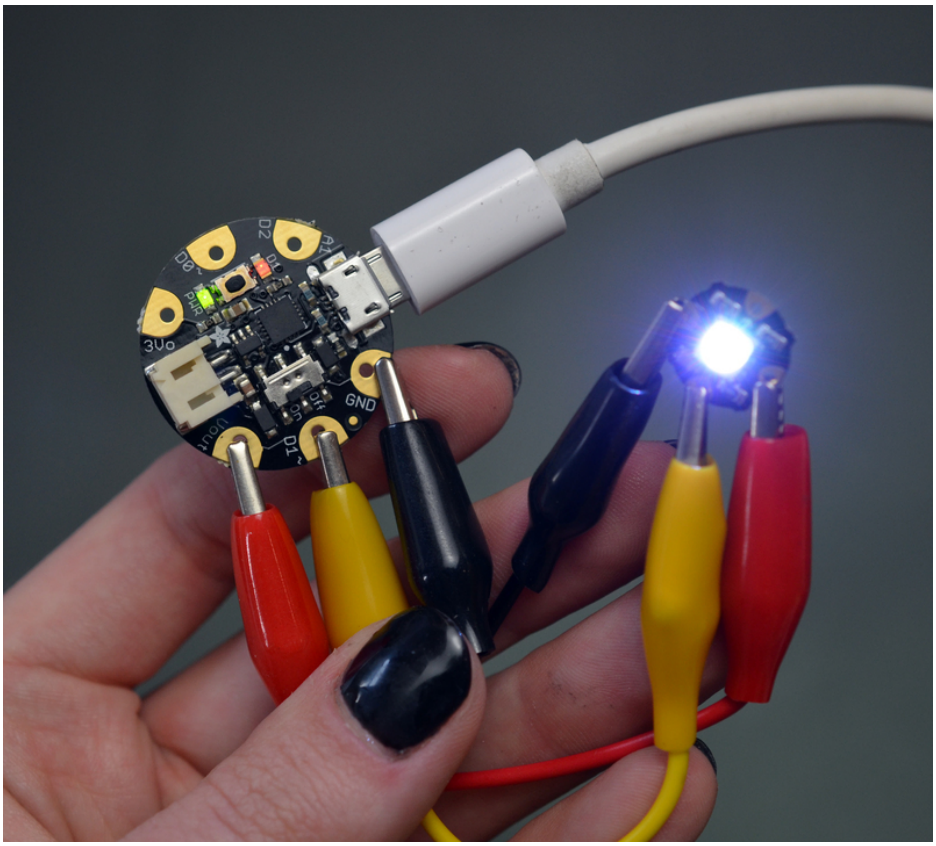


If you cannot find that file in the dropdown menu go back up and reinstall the library.

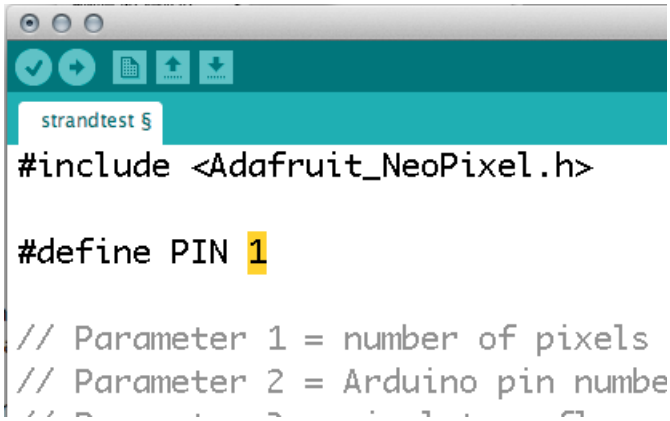
This will launch a window containing the sample sketch. Upload it to your FLORA or GEMMA with the Upload button.

Once successfully uploaded to your FLORA or GEMMA, the pixel will flash and fade different colors. Congrats, you've tested your pixel!

We do test each pixel in the factory, but you should use this method to test that all three colors (red, green, blue) work on each of your pixels before sewing them into your project!

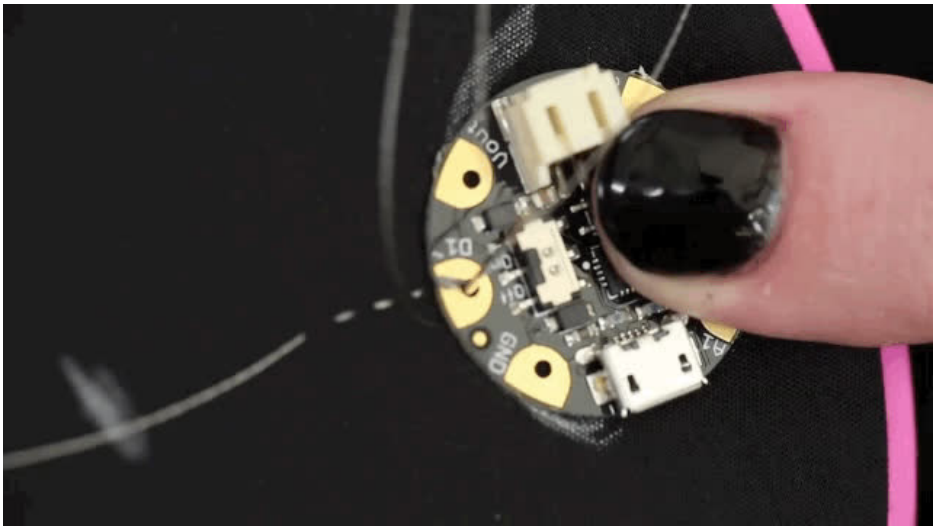


For GEMMA, you'll need to change the pin number in the Arduino sketch before uploading. Change "6" to "1":



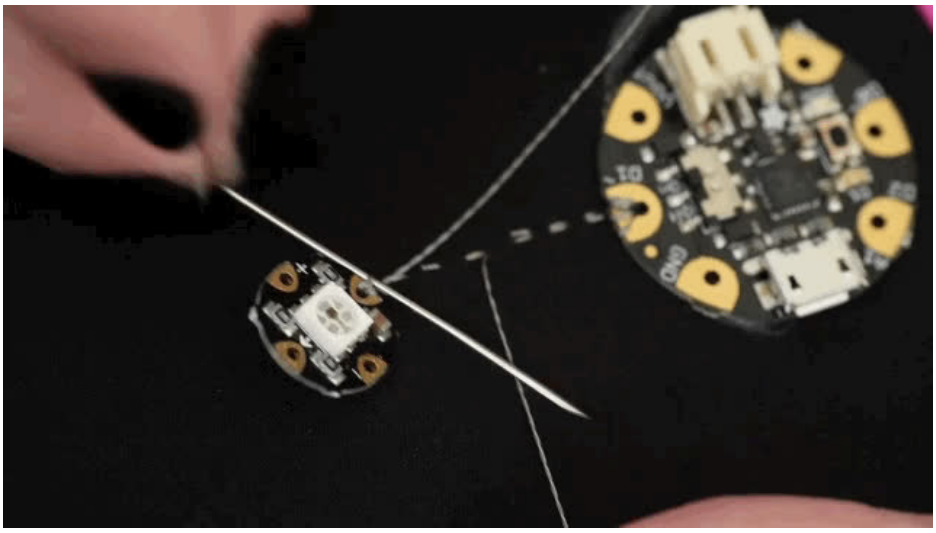
Then upload the sample code to GEMMA by pressing the reset button (the red LED should start pulsing) and clicking the "upload" button in the Arduino software.

Sewing more pixels

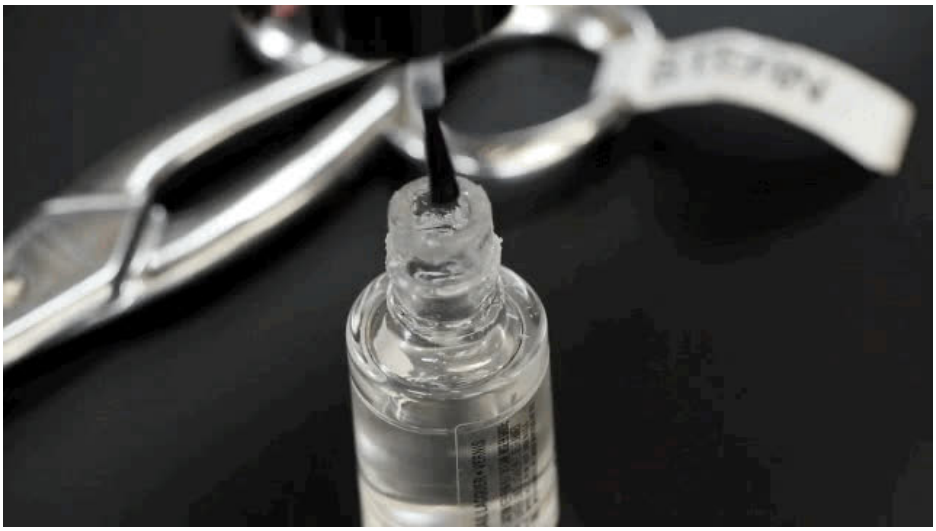


The pixels are chainable - so you only need 1 pin/wire to control as many LEDs as you like. They're easy to sew, and the chainable design means no crossed conductive threads (http://adafru.it/641). The output of one pixel connects directly to the input of the next.

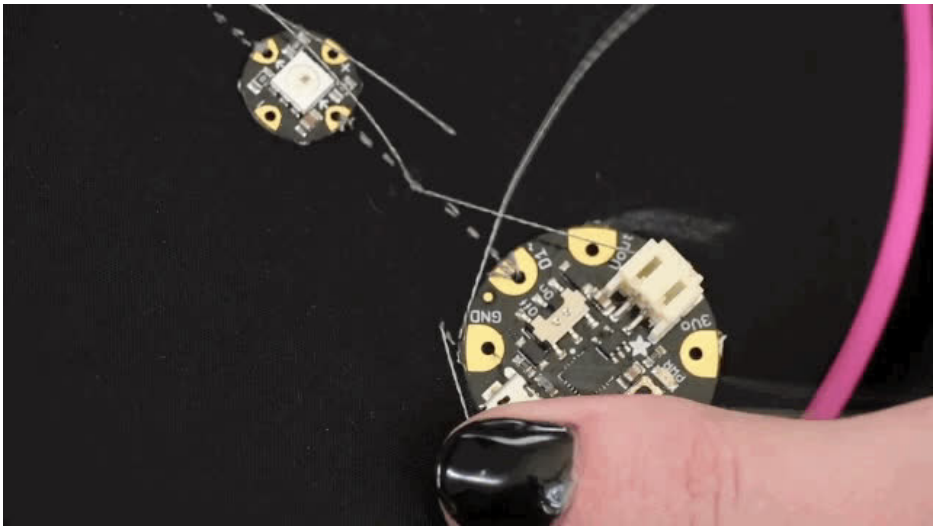
To begin sewing, stitch around the data pin of your microcontroller (**A1** on Circuit Playground, **D1** on GEMMA or Gemma M0 and **D6** on FLORA are ideal because they're right between power and ground), and stitch over to your first pixel.



Make sure that each arrow points away from the microcontroller in a line. Stitch around the input pad tightly, even knotting the thread here to form an extra sturdy connection.

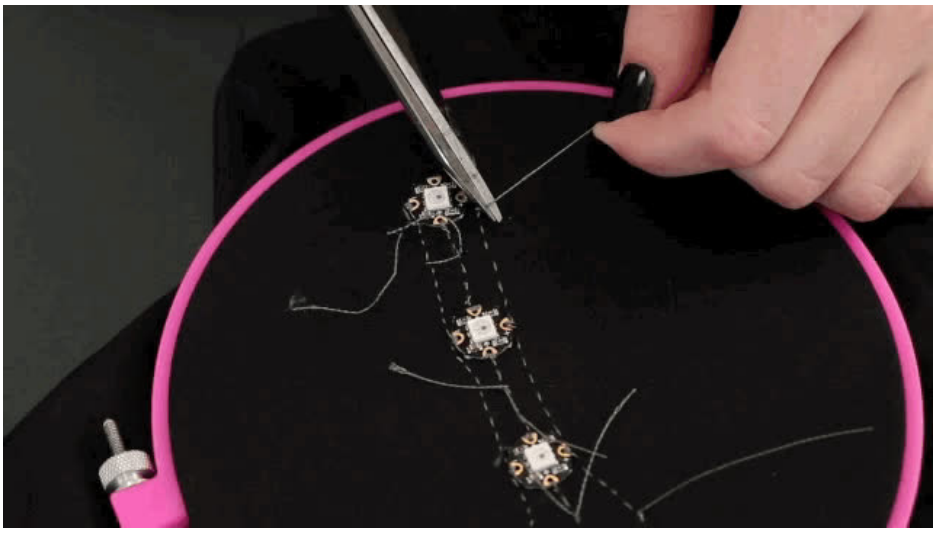


Stitch back to the thread origin and tie the two ends in a square knot. Use clear nail polish to seal this knot and pull the ends tight until it dries. Do not clip the thread tails until later on.

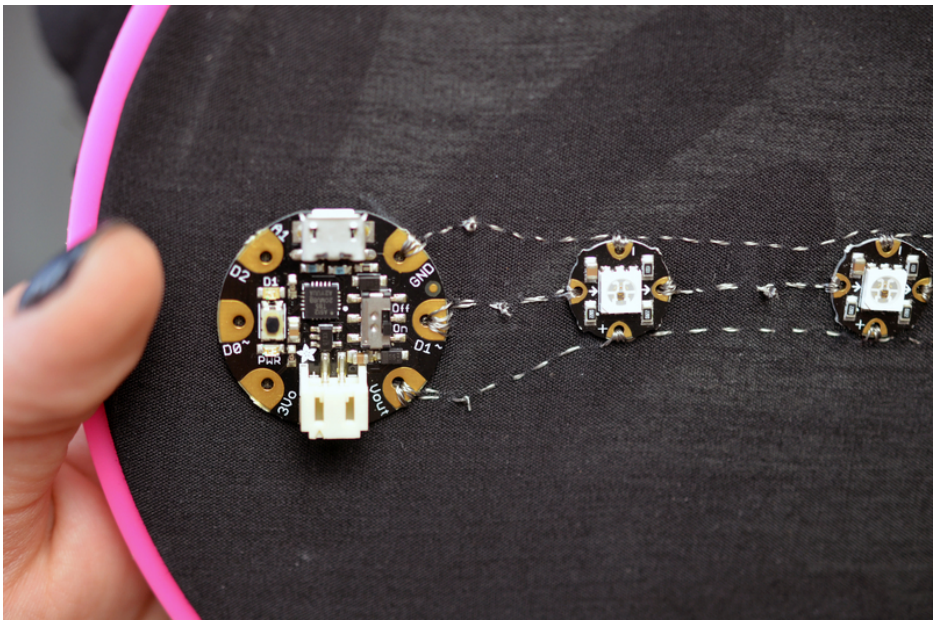


All the positive pads (marked with a +) connect together to form one power bus. Likewise all the negative pads (marked with a -) connect together to form one ground bus.

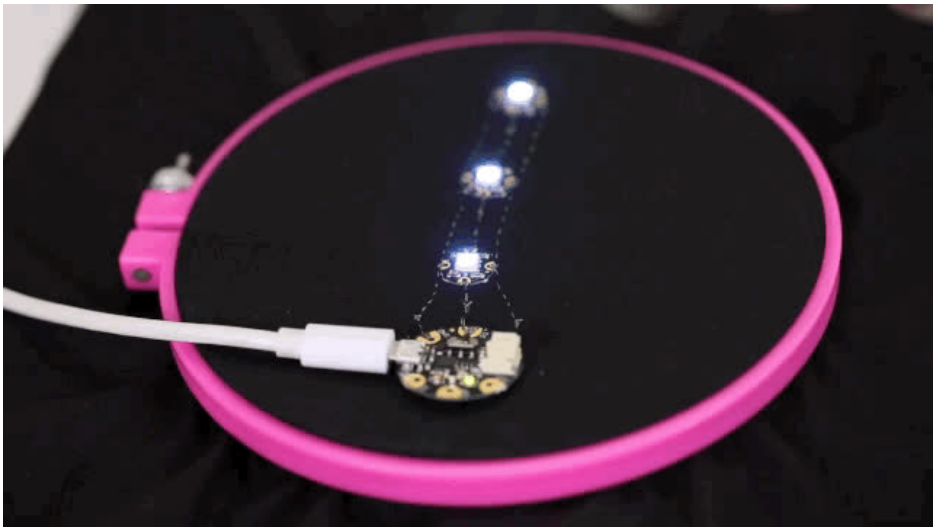
Here's what a three-pixel circuit looks like from the back. See the individual data connections in the center and the continuous power and ground buses above and below?



Double check your knots are secure before clipping all your thread tails. Clean up your work space so there aren't any stray bits of conductive thread hanging around.

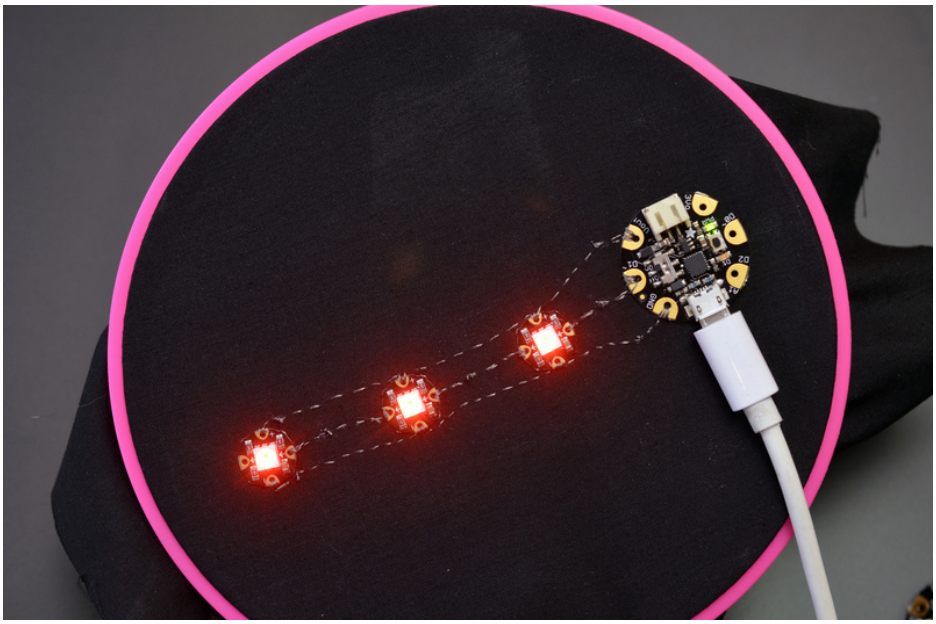


Visually inspect your circuit to check for shorts or stray threads.



Plug your microcontroller into your computer with a USB cable. Change the number of pixels in the Arduino sketch or MakeCode project to match the number of pixels in your projects, and make sure the pin number matches what you sewed to. Upload the strandtest sample code as you did when testing a pixel earlier in this guide. Your pixels should light up and animate different colors and patterns.

If they don't all come on or some later in the chain are flickering, your stitches might not be snug enough against the pads of the circuit board. Double check your sewing and reinforce it where necessary (with the circuit off/unplugged).



The library for these pixels is very similar to our Adafruit_WS2801 (https://adafru.it/aRL) library for other types of RGB pixels (https://adafru.it/aRL).

Project Ideas

Light up Angler Fish Embroidery (https://adafru.it/qzd)

NeoPixel Coat Buttons (https://adafru.it/qze)

Sparkle Skirt with FLORA Motion Sensor (https://adafru.it/qzf)

Textile Potentiometer Hoodie (https://adafru.it/qzA)

NeoPixel Tiara (https://adafru.it/rza)

Close Encounters Hat (https://adafru.it/rzb)

Light-Activated Pixel Heart (https://adafru.it/rzc)

Space Face LED Galaxy Makeup (https://adafru.it/rzd)

Pac Man Pixel Suspenders (https://adafru.it/rze)

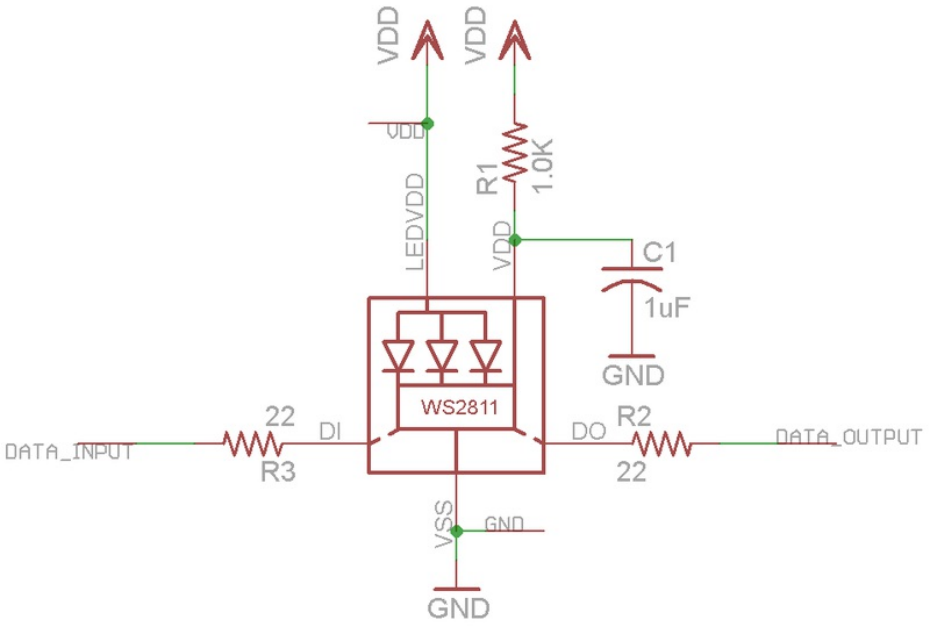
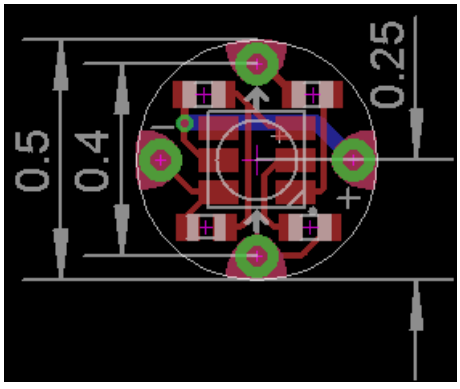
Chameleon Scarf with FLORA Color Sensor (https://adafru.it/dNd)

Downloads

Files

- Fritzing object in Adafruit Fritzing library (https://adafru.it/aP3)
- EagleCAD PCB files on GitHub (https://adafru.it/rzf)
- WS2812 Datasheet (https://adafru.it/qta)
- SK6812 Datasheet (https://adafru.it/Cc1)

NeoPixel Schematic



Fabrication Print