



NeoPixel Tiara

Created by Becky Stern



<https://learn.adafruit.com/neopixel-tiara>

Last updated on 2022-12-01 02:07:41 PM EST

Table of Contents

Overview	3
Build it!	4
Arduino Code	9
CircuitPython Code	12
Wear it!	14

Overview

You can make a crown of light for prom or your birthday this year! A few free-wired NeoPixels make you the cyber-pageant queen.

For this project you will need:

- [Gemma M0 \(\)](#) or [GEMMA v2 wearable microcontroller \(http://adafru.it/1222\)](http://adafru.it/1222) (or build on the [GEMMA Starter Pack \(http://adafru.it/1657\)](http://adafru.it/1657))
- seven [sewable NeoPixels \(http://adafru.it/1260\)](http://adafru.it/1260) (or any combination of [pixels \(\)](#), NeoPixel rings, or high-density NeoPixel strip) Use "RGB" NeoPixel rings, not "RGBW" type!
- [2x2032 coin cell battery holder \(http://adafru.it/783\)](http://adafru.it/783) and [batteries \(http://adafru.it/654\)](http://adafru.it/654) or [rechargeable lipoly battery \(http://adafru.it/1317\)](http://adafru.it/1317)
- zip ties & hot glue
- [3D printed band, crown \(\)](#), or recycled stiff headband
- [solid core hookup wire \(http://adafru.it/290\)](http://adafru.it/290)
- sharp [wire strippers \(http://adafru.it/527\)](http://adafru.it/527)
- sharp [flush snips \(http://adafru.it/1368\)](http://adafru.it/1368)

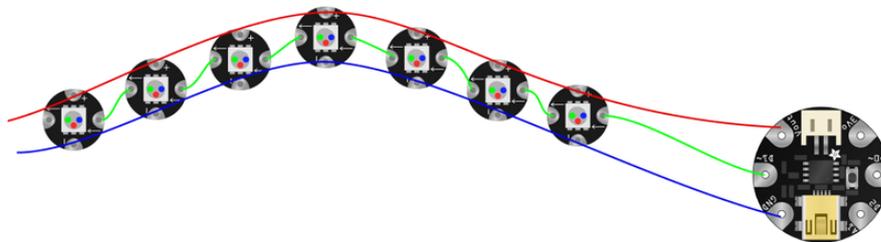
Glamour shots by Andrew Tingle.

This guide was written for the Gemma v2 board, but can be done with either the v2 or Gemma M0. We recommend the Gemma M0 as it is easier to use and is more compatible with modern computers!

Before you get started, follow the [Gemma M0 guide \(\)](#) or the [Classic Introducing GEMMA guide \(\)](#)



Build it!



This diagram uses the original Gemma but you can also use the Gemma M0 with the exact same wiring!

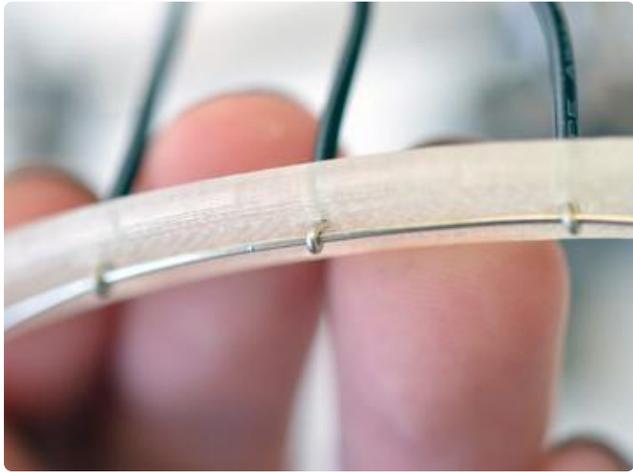
Above is the circuit diagram for the simple tiara. All ground pads are connected to GND on GEMMA, likewise all power pads are connected to Vout on GEMMA, and data lines are chained from D1 to the inward facing arrow on the first pixel, then from each pixel's outward facing arrow to the next pixel's inward facing arrow.



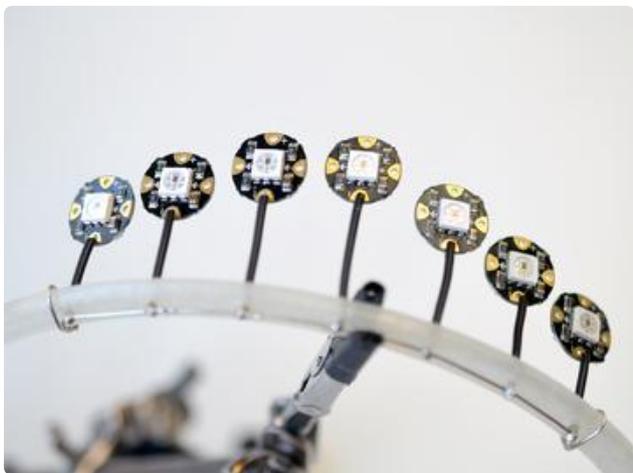
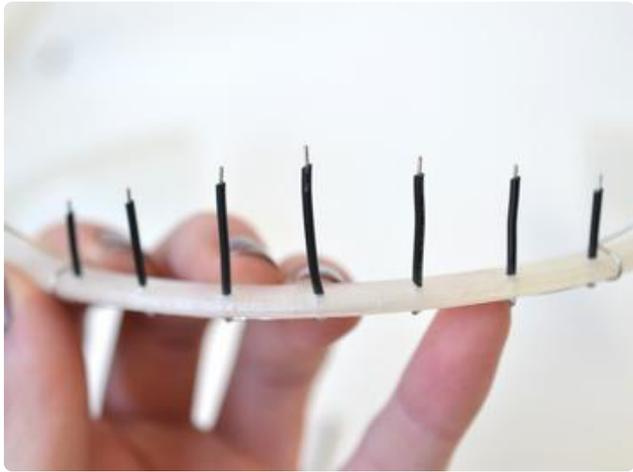
Strip the insulation from a long piece of solid-core hookup wire and clip it to one side of your headband. This will be the pixels' ground bus.

Insert perpendicular wires through holes in the headband and curl their stripped ends around the long ground bus wire.

Crimp wire connections with pliers.



Be sure wire connections are tidy enough to nest snugly against the hair band, and solder connections in place.



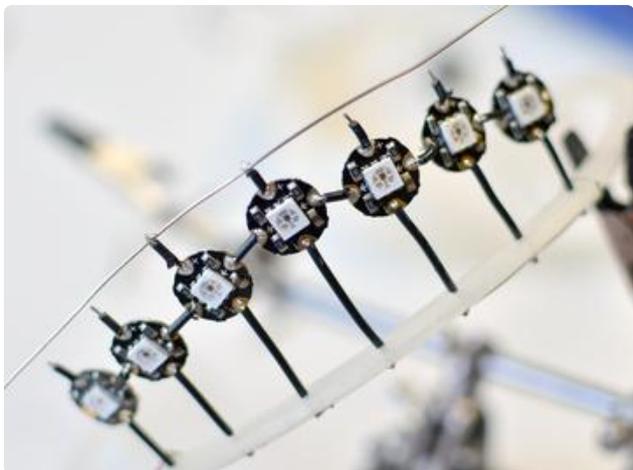
Trim wires to the silhouette you like, then strip the ends. Bend the stripped ends over and hand the NeoPixels from them (through the pad marked -). Solder all NeoPixels, making sure they're facing the same way.

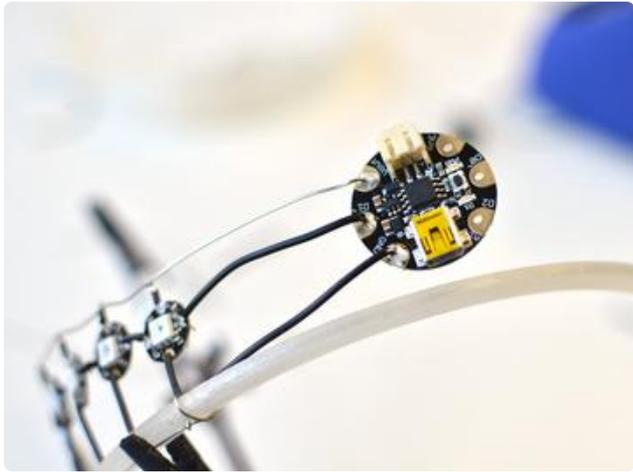


Next use bits of wire in the shape of staples to connect up the data lines.

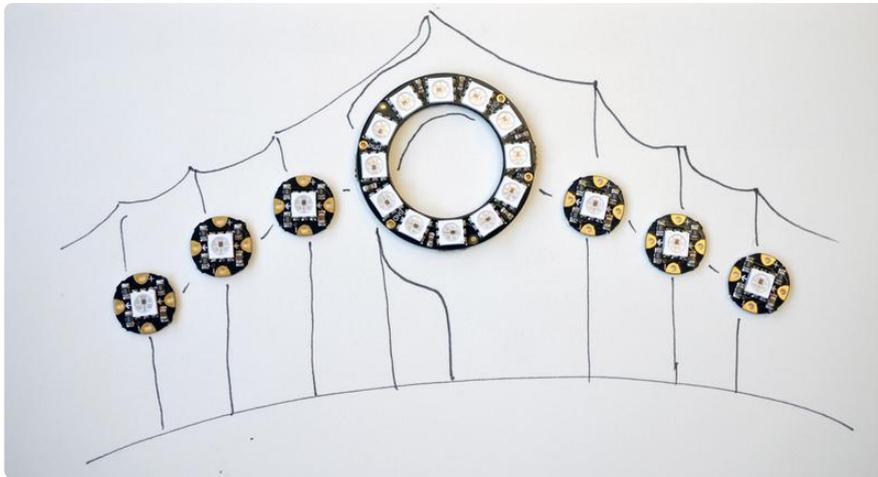
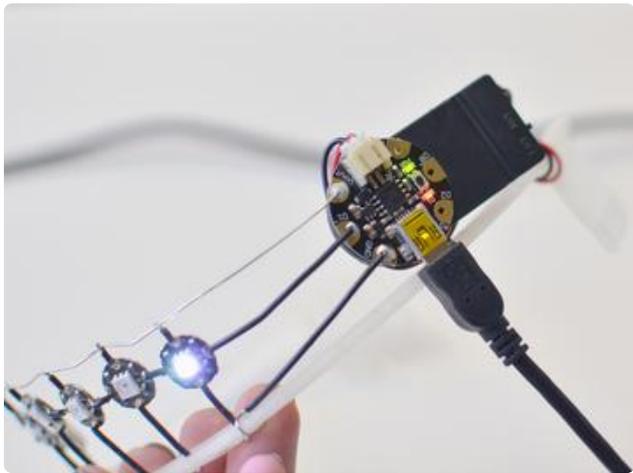


Solder wires from the + side of each pixel, and cut to length. Strip the ends and solder to one long power bus.





Connect power, ground, and data lines to GEMMA at the input end of the pixel chain. Plug in over USB and program it up! This basic sketch flashes the seven pixels randomly:



Get creative with your tiara design! This one uses a NeoPixel ring in the center, and the wiring is barely different than above.

Arduino Code

The Arduino code presented below works equally well on all versions of GEMMA: v2 and M0. But if you have an M0 board, consider using the CircuitPython code on the next page of this guide, no Arduino IDE required!

Click to Download the NeoPixel Library

Installing Arduino libraries is a frequent stumbling block. If this is your first time, or simply needing a refresher, please read the [All About Arduino Libraries \(\)](#) tutorial. ()If the library is correctly installed (and the Arduino IDE is restarted), you should be able to navigate through the “File” rollover menus as follows:

File→Sketchbook→Libraries→Adafruit_NeoPixel→strandtest

Connect up your NeoPixels in a solderless breadboard and use alligator clips to attach to GEMMA, referring to the circuit diagram if necessary.

You’ll need to change a few lines in the code regarding the data pin (1), type of pixels (RGB vs GRB), and number of pixels (5). The resulting (and slightly simplified) code is below:

```
// SPDX-FileCopyrightText: 2017 Dano Wall for Adafruit Industries
// SPDX-FileCopyrightText: 2017 Becky Stern for Adafruit Industries
//
// SPDX-License-Identifier: MIT

//Random Flash animation for Neopixel circuits
//by Dano Wall and Becky Stern for Adafruit Industries
//based on the Sparkle Skirt, minus the accelerometer
#include <Adafruit_NeoPixel.h>

#define PIN 1

// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(7, PIN, NEO_GRB + NEO_KHZ800);

// Here is where you can put in your favorite colors that will appear!
// just add new {nnn, nnn, nnn}, lines. They will be picked out randomly
//           R   G   B
uint8_t myColors[][3] = {{232, 100, 255}, // purple
                        {200, 200, 20},  // yellow
                        {30, 200, 200},   // blue
                        };

// don't edit the line below
#define FAVCOLORS sizeof(myColors) / 3

void setup() {
  strip.begin();
  strip.setBrightness(40);
  strip.show(); // Initialize all pixels to 'off'
}

void loop() {
  flashRandom(5, 1); // first number is 'wait' delay, shorter num == shorter twinkle
```

```

flashRandom(5, 3); // second number is how many neopixels to simultaneously light
up
flashRandom(5, 2);
}

void flashRandom(int wait, uint8_t howmany) {

  for(uint16_t i=0; i<howmany; i++) {
    // pick a random favorite color!
    int c = random(FAVCOLORS);
    int red = myColors[c][0];
    int green = myColors[c][1];
    int blue = myColors[c][2];

    // get a random pixel from the list
    int j = random(strip.numPixels());

    // now we will 'fade' it in 5 steps
    for (int x=0; x < 5; x++) {
      int r = red * (x+1); r /= 5;
      int g = green * (x+1); g /= 5;
      int b = blue * (x+1); b /= 5;

      strip.setPixelColor(j, strip.Color(r, g, b));
      strip.show();
      delay(wait);
    }
    // & fade out in 5 steps
    for (int x=5; x >= 0; x--) {
      int r = red * x; r /= 5;
      int g = green * x; g /= 5;
      int b = blue * x; b /= 5;

      strip.setPixelColor(j, strip.Color(r, g, b));
      strip.show();
      delay(wait);
    }
  }
  // LEDs will be off when done (they are faded to 0)
}

```

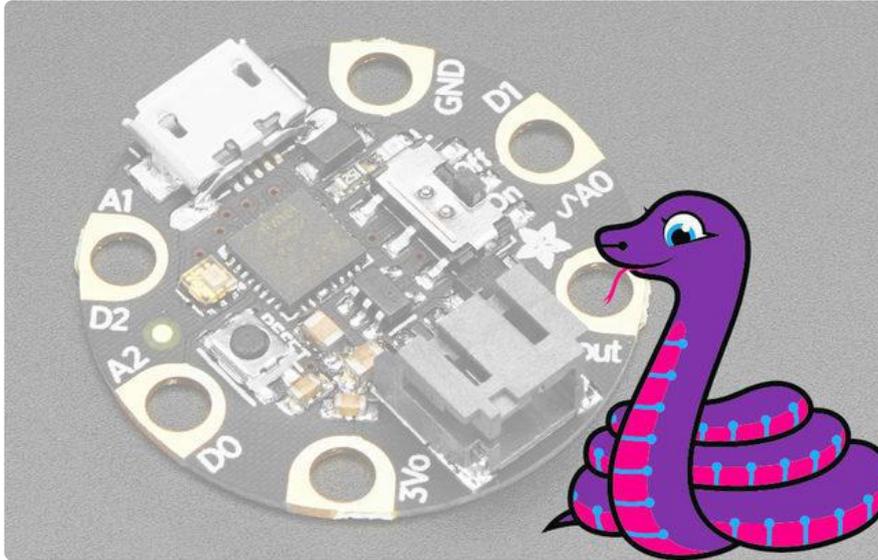
From the Tools→Board menu, select the device you are using:

- Adafruit Gemma M0
- Adafruit Gemma 8 MHz

Connect the USB cable between the computer and your device. The original Gemma (8 MHz) need the reset button pressed on the board, then click the upload button (right arrow icon) in the Arduino IDE. You do not need to press the reset on the newer Gemma M0.

When the battery is connected, you should get a light show from the LEDs. All your pixels working? Great! You can take apart this prototype and get ready to put the pixels in the collar. Refer to the [NeoPixel Uberguide \(\)](#) for more info.

CircuitPython Code



GEMMA M0 boards can run CircuitPython — a different approach to programming compared to Arduino sketches. In fact, CircuitPython comes factory pre-loaded on GEMMA M0. If you’ve overwritten it with an Arduino sketch, or just want to learn the basics of setting up and using CircuitPython, this is explained in the [Adafruit GEMMA M0 guide \(\)](#).

These directions are specific to the “M0” GEMMA board. The original GEMMA with an 8-bit AVR microcontroller doesn’t run CircuitPython...for those boards, use the Arduino sketch on the “Arduino code” page of this guide.

Below is CircuitPython code that works similarly (though not exactly the same) as the Arduino sketch shown on a prior page. To use this, plug the GEMMA M0 into USB...it should show up on your computer as a small flash drive...then edit the file “code.py” with your text editor of choice. Select and copy the code below and paste it into that file, entirely replacing its contents (don’t mix it in with lingering bits of old code). When you save the file, the code should start running almost immediately (if not, see notes at the bottom of this page).

If GEMMA M0 doesn’t show up as a drive, follow the GEMMA M0 guide link above to prepare the board for CircuitPython.

```
# SPDX-FileCopyrightText: 2017 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time

import board
```

```

import neopixel

try:
    import urandom as random
except ImportError:
    import random

numpix = 7 # Number of NeoPixels
pixpin = board.D1 # Pin where NeoPixels are connected
strip = neopixel.NeoPixel(pixpin, numpix, brightness=1, auto_write=True)
colors = [
    [232, 100, 255], # Purple
    [200, 200, 20], # Yellow
    [30, 200, 200], # Blue
]

def flash_random(wait, howmany):
    for _ in range(howmany):

        c = random.randint(0, len(colors) - 1) # Choose random color index
        j = random.randint(0, numpix - 1) # Choose random pixel
        strip[j] = colors[c] # Set pixel to color

        for i in range(1, 5):
            strip.brightness = i / 5.0 # Ramp up brightness
            time.sleep(wait)

        for i in range(5, 0, -1):
            strip.brightness = i / 5.0 # Ramp down brightness
            strip[j] = [0, 0, 0] # Set pixel to 'off'
            time.sleep(wait)

while True:
    # first number is 'wait' delay, shorter num == shorter twinkle
    flash_random(.01, 1)
    # second number is how many neopixels to simultaneously light up
    flash_random(.01, 3)
    flash_random(.01, 2)

```

This code requires the `neopixel.py` library. A factory-fresh board will have this already installed. If you've just reloaded the board with CircuitPython, create the "lib" directory and then [download neopixel.py from Github](#) ().

[Download neopixel.py from GitHub](#)

Wear it!



Plug in a coin cell battery holder or rechargeable lipoly battery! You can pin the battery up into your hairdo or attach to the tiara with glue.

Glamour shots by Andrew Tingle.