Animated NeoPixel Glow Fur Scarf

Created by Erin St Blaine

https://learn.adafruit.com/animated-neopixel-gemma-glow-fur-scarf

Last updated on 2023-08-29 02:43:07 PM EDT

Table of Contents

Intro & Materials

Build a fantastic light-up fuzzy scarf with seven different color modes. Easily switch between modes with the press of a button. With a little easy code hacking, you can even create your own color combinations.

Warning: Wearing this scarf in public will get you noticed. Children will follow you around like you're the pied piper. Adults and paparazzi will chase you down the street in order to get a photo with you. Minstrels will sing ballads about the glow you left in their hearts long after your batteries die.

Prepare for the madness of instant glow-fur celebrity.

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

For this project you will need:

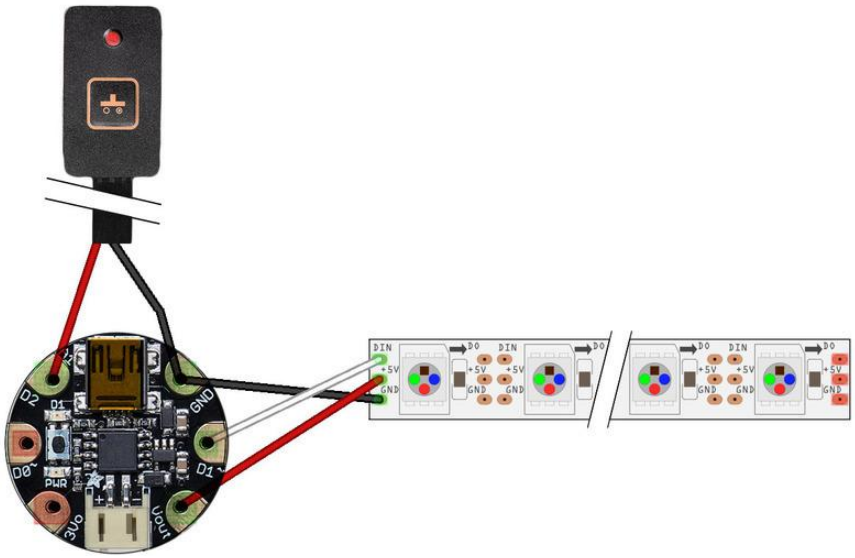
- 1 - 1.5 yds x 9 inches of white fun fur
- 1 - 1.5 meters of White 60 LED Neopixel Strip (http://adafru.it/1138) (no more than 100 pixels will run with the gemma so keep it under 100)

- Gemma M0 () or Original Gemma (http://adafru.it/1222) (M0 type is recommended!)
- 2500mAh Lithium Ion Polymer Battery (http://adafru.it/328) (don't forget the charger (http://adafru.it/1304))
- Tactile On/Off Switch with Leads (http://adafru.it/1092)
- Membrane LED Keypad () (not pictured)
- Soldering iron & solder
- A few pieces of 26g stranded wire ()
- One jumper wire (http://adafru.it/1957)
- Heat shrink
- Needle & thread
- Two heavy duty sew-in snaps
- USB LiIon/LiPoly charger ()

Also helpful to have:

- Hot glue gun
- Heat gun
- 1/2 inch clear heat shrink
- Sewing machine

Wiring



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

LED Strip:

- Gemma Vout -> +5v
- Gemma D1 -> DI (or DIN on some strips)
- Gemma GND -> GND

Membrane LED Keypad:

- Gemma GND -> Center pin
- Gemma D2 -> Left pin

Note: some generations of NeoPixel strip have their 5V/GND/DI pins in a different order. ALWAYS USE THE PIN LABELS PRINTED ON THE STRIP, don't just blindly follow the physical order shown in the wiring diagram!

Arduino Code

The Arduino code presented below works equally well on all versions of GEMMA: v1, 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!

Software Setup

Before you get started, follow the Introducing Gemma M0 guide () or Introducing GEMMA guide ()

FastLED Library

You will also need to install the FastLED library in Arduino (`Sketch > Include Library > Manage Libraries...`)

Libraries? Why? What's a Library?

In a nutshell, Arduino libraries have a lot of pre-written functions that make your neopixels easy to command. You can do fancy stuff without being a code guru. Yay Libraries!

FastLED is a fast, efficient, easy-to-use Arduino library for programming addressable LED strips and pixels. It has a lot of features to get your animations up and running fast -- and it has a lot of code samples available if you're just learning to code.

All about Arduino Libraries () will tell you everything you ever wanted to know about libraries, including more detailed installation instructions.

Once your curiosity is satiated and your library is installed, copy and paste the code into your Arduino window.

Go to your Tools menu and select Gemma from the list of boards. Plug your Gemma into your computer via the onboard USB port. Press the "reset" button on your Gemma and wait for the blinky red light, then click the upload button in Arduino.

```
// SPDX-FileCopyrightText: 2018 Mikey Sklar for Adafruit Industries
//
// SPDX-License-Identifier: MIT

#include <FastLED.h>

#define LED_PIN      1   // which pin your pixels are connected to
#define NUM_LEDS    78   // how many LEDs you have
#define BRIGHTNESS 200   // 0-255, higher number is brighter.
#define SATURATION 255   // 0-255, 0 is pure white, 255 is fully saturated color
#define SPEED       10   // How fast the colors move.  Higher numbers = faster
motion
#define STEPS        2   // How wide the bands of color are.  1 = more like a
gradient, 10 = more like stripes
#define BUTTON_PIN   2   // button is connected to pin 2 and GND

#define COLOR_ORDER GRB  // Try mixing up the letters (RGB, GBR, BRG, etc) for a
whole new world of color combinations
```

```
CRGB leds[NUM_LEDS];
CRGBPalette16 currentPalette;
CRGBPalette16 targetPalette( PartyColors_p );
TBlendType    currentBlending;
int ledMode = 0;


//FastLED comes with several palettes pre-programmed.  I like purple a LOT, and so
I added a custom purple palette.

const TProgmemPalette16 PurpleColors_p PROGMEM =
{
  CRGB::Purple,
  CRGB::Purple,
  CRGB::MidnightBlue,
  CRGB::MidnightBlue,

  CRGB::Purple,
  CRGB::Purple,
  CRGB::BlueViolet,
  CRGB::BlueViolet,

  CRGB::DarkTurquoise,
  CRGB::DarkTurquoise,
  CRGB::DarkBlue,
  CRGB::DarkBlue,

  CRGB::Purple,
  CRGB::Purple,
  CRGB::BlueViolet,
  CRGB::BlueViolet
};


unsigned long keyPrevMillis = 0;
const unsigned long keySampleIntervalMs = 25;
byte longKeyPressCountMax = 80;    // 80 * 25 = 2000 ms
byte longKeyPressCount = 0;

byte prevKeyState = HIGH;         // button is active low

void setup() {
  delay( 2000 ); // power-up safety delay
  FastLED.addLeds<WS2812B, LED_PIN, COLOR_ORDER>(leds, NUM_LEDS).setCorrection(
TypicalLEDStrip );
  FastLED.setBrightness(  BRIGHTNESS );
  currentBlending =LINEARBLEND;
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {

  byte currKeyState = digitalRead(BUTTON_PIN);

  if ((prevKeyState == LOW) && (currKeyState == HIGH)) {
    shortKeyPress();
  }
  prevKeyState = currKeyState;

  static uint8_t startIndex = 0;
  startIndex = startIndex + 1; /* motion speed */

  switch (ledMode) {

  case 0:
    currentPalette = HeatColors_p;    //Red & Yellow, Fire Colors
    break;
  case 1:
```

```
    currentPalette = ForestColors_p;    //Foresty greens and yellows
    break;
  case 2:
    currentPalette = OceanColors_p;  //Oceans are pretty and filled with mermaids
    break;
  case 3:
    currentPalette = PurpleColors_p;  //My custom palette from above
    break;
  case 4:
    currentPalette = RainbowColors_p;  //All the colors!
    break;
  case 5:
    currentPalette = RainbowStripeColors_p;   //Rainbow stripes
    break;
  case 6:
    currentPalette = PartyColors_p; //All the colors except the greens, which make
people look a bit washed out
    break;
  }

  FillLEDsFromPaletteColors( startIndex);
  FastLED.show();
  FastLED.delay(1000 / SPEED);
}

void FillLEDsFromPaletteColors( uint8_t colorIndex) {
  for( int i = 0; i < NUM_LEDS; i++) {
    leds[i] = ColorFromPalette( currentPalette, colorIndex, BRIGHTNESS,
currentBlending);
    colorIndex += STEPS;
  }
}

void shortKeyPress() {
  ledMode++;
  if (ledMode > 6) {
    ledMode=0;
  }
}
```

Taking a look at the code, you'll see that there are a few variables at the top you can change.

#define NUM_LEDS 78 -- change this number to match the number of LEDs in your scarf.

#define BRIGHTNESS 200 -- change this number to make your scarf brighter (woo hoo!) or dimmer (longer battery life). 0 is totally dark, max bright is 255.

#define SATURATION 255 -- lower number to make the colors more pastel / less saturated. Max is 255.

#define SPEED 10 -- 10 is kind of a slow crawl of color. Higher numbers make it move faster.

#define STEPS 2 -- How wide the bands of color are. 1 = more like a gradient, 10 = more like stripes

You'll also see a custom purple color palette. You can play around with this and change it to whichever colors you like. All the supported color names can be found in the keywords.txt document you downloaded along with the FastLED library.

If you encounter trouble...

Any time you hit a roadblock with a neopixel project, we'll usually ask that you start with the "strandtest" example from our own Adafruit_NeoPixel library. This helps us narrow down whether it's a hardware or software issue. The library is installed similarly to FastLED or any other in Arduino (`Sketch > Include Library > Manage Libraries...`)

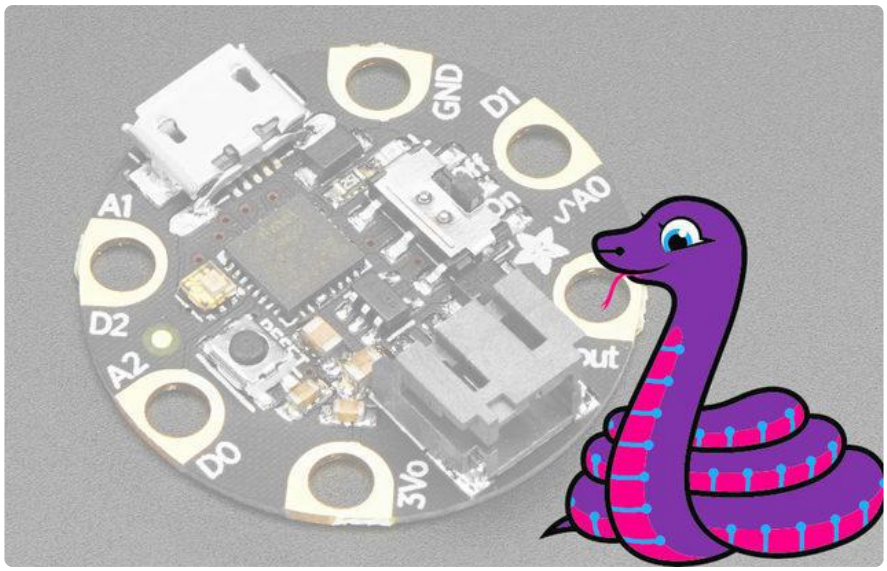
You'll find the strandtest example under File→Sketchbook→Libraries→Adafruit_NeoPixel→strandtest

If strandtest fails to run, this suggests a hardware issue...for example, connecting to the wrong Gemma pin.

If you're new to Arduino programming and LEDs, we usually suggest starting with the Adafruit_NeoPixel library...it's pretty basic, the strip declaration is more conventional, and we can stay on top of keeping it compatible with our own products and the most mainstream Arduino boards.

As FastLED is a more "bleeding edge" third-party library, we can't always guarantee compatibility across versions or with specific boards. You can find help through their community on Google+ (). This is potent stuff, written by people with a deep appreciation for LED art, and we wanted to showcase it.

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: 2018 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import adafruit_fancyled.adafruit_fancyled as fancy
import board
import neopixel
from digitalio import DigitalInOut, Direction, Pull
```

```
led_pin = board.D1  # which pin your pixels are connected to
num_leds = 78  # how many LEDs you have
brightness = 1.0  # 0-1, higher number is brighter
saturation = 255  # 0-255, 0 is pure white, 255 is fully saturated color
steps = 0.01  # how wide the bands of color are.
offset = 0  # cummulative steps
fadeup = True  # start with fading up - increase steps until offset reaches 1
index = 8  # midway color selection
blend = True  # color blending between palette indices

# initialize list with all pixels off
palette = [0] * num_leds

# Declare a NeoPixel object on led_pin with num_leds as pixels
# No auto-write.
# Set brightness to max.
# We will be using FancyLED's brightness control.
strip = neopixel.NeoPixel(led_pin, num_leds, brightness=1, auto_write=False)

# button setup
button = DigitalInOut(board.D2)
button.direction = Direction.INPUT
button.pull = Pull.UP
prevkeystate = False
ledmode = 0  # button press counter, switch color palettes

# FancyLED allows for assigning a color palette using these formats:
# * The first (5) palettes here are mixing between 2-elements
# * The last (3) palettes use a format identical to the FastLED Arduino Library
# see FastLED - colorpalettes.cpp
forest = [fancy.CRGB(0, 255, 0),  # green
          fancy.CRGB(255, 255, 0)]  # yellow

ocean = [fancy.CRGB(0, 0, 255),  # blue
         fancy.CRGB(0, 255, 0)]  # green

purple = [fancy.CRGB(160, 32, 240),  # purple
          fancy.CRGB(238, 130, 238)]  # violet

all_colors = [fancy.CRGB(0, 0, 0),  # black
              fancy.CRGB(255, 255, 255)]  # white

washed_out = [fancy.CRGB(0, 0, 0),  # black
              fancy.CRGB(255, 0, 255)]  # purple

rainbow = [0xFF0000, 0xD52A00, 0xAB5500, 0xAB7F00,
           0xABAB00, 0x56D500, 0x00FF00, 0x00D52A,
           0x00AB55, 0x0056AA, 0x0000FF, 0x2A00D5,
           0x5500AB, 0x7F0081, 0xAB0055, 0xD5002B]

rainbow_stripe = [0xFF0000, 0x000000, 0xAB5500, 0x000000,
                  0xABAB00, 0x000000, 0x00FF00, 0x000000,
                  0x00AB55, 0x000000, 0x0000FF, 0x000000,
                  0x5500AB, 0x000000, 0xAB0055, 0x000000]

heat_colors = [0x330000, 0x660000, 0x990000, 0xCC0000, 0xFF0000,
               0xFF3300, 0xFF6600, 0xFF9900, 0xFFCC00, 0xFFFF00,
               0xFFFF33, 0xFFFF66, 0xFFFF99, 0xFFFFCC]


def remapRange(value, leftMin, leftMax, rightMin, rightMax):
    # this remaps a value fromhere original (left) range to new (right) range
    # Figure out how 'wide' each range is
    leftSpan = leftMax - leftMin
    rightSpan = rightMax - rightMin

    # Convert the left range into a 0-1 range (int)
    valueScaled = int(value - leftMin) / int(leftSpan)
```

```
    # Convert the 0-1 range into a value in the right range.
    return int(rightMin + (valueScaled * rightSpan))


def shortkeypress(color_palette):
    color_palette += 1

    if color_palette > 6:
        color_palette = 1

    return color_palette


while True:

    # check for button press
    currkeystate = button.value

    # button press, move to next pattern
    if (prevkeystate is not True) and currkeystate:
        ledmode = shortkeypress(ledmode)

    # save button press state
    prevkeystate = currkeystate

    # Fire Colors [ HEAT ]
    if ledmode == 1:
        palette = heat_colors

    # Forest
    elif ledmode == 2:
        palette = forest

    # Ocean
    elif ledmode == 3:
        palette = ocean

    # Purple Lovers
    elif ledmode == 4:
        palette = purple

    # All the colors!
    elif ledmode == 5:
        palette = rainbow

    # Rainbow stripes
    elif ledmode == 6:
        palette = rainbow_stripe

    # All the colors except the greens, washed out
    elif ledmode == 7:
        palette = washed_out

    for i in range(num_leds):
        color = fancy.palette_lookup(palette, offset + i / num_leds)
        color = fancy.gamma_adjust(color, brightness=brightness)
        strip[i] = color.pack()
    strip.show()

    if fadeup:
        offset += steps
        if offset >= 1:
            fadeup = False
    else:
        offset -= steps
        if offset <= 0:
            fadeup = True
```

This code requires two libraries be installed:

- neopixel
- adafruit_fancyled

A factory-fresh board will have the neopixel library already installed. If you've just reloaded the board with CircuitPython, create the "lib" directory and then copy in the neopixel.mpy and adafruit_fancyled folder from the latest release of the Adafruit_CircuitPython_Bundle ().

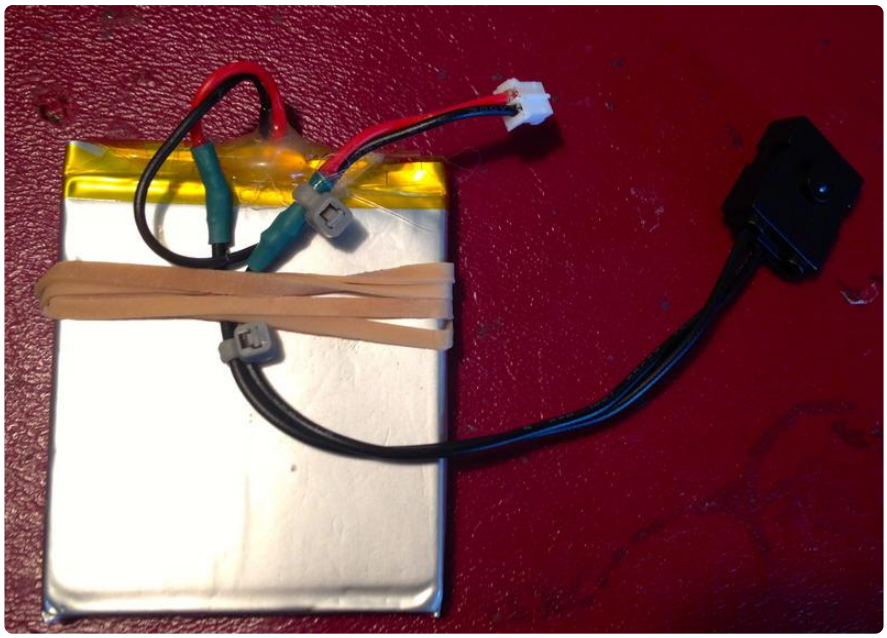
The FancyLED library () being using in this CircuitPython example is not the same as the FastLED ()used for Arduino. FancyLED has a subset of FastLED features and some different syntax. The FancyLED tutorial provides an excellent overview ().

The file system layout on your gemma M0 should look like this:

```
$ pwd
/Volumes/CIRCUITPY
$ find .
.
./boot_out.txt
./.fseventsd
./.fseventsd/fseventsd-uuid
./lib
./lib/neopixel.mpy
./lib/adafruit_fancyled
./lib/adafruit_fancyled/adafruit_fancyled.mpy
./lib/adafruit_fancyled/fastled_helpers.mpy
./main.py
```

Battery & Switch

The gemma has a tiny on/off switch onboard. You can skip this step and just use that switch, but adding a clicky switch you can feel without opening up the scarf and looking inside can be really helpful.



Lithium Polymer batteries are wonderful things. They are small, compact and light and carry a lot of power, and will keep your LEDs running for a nice long time.

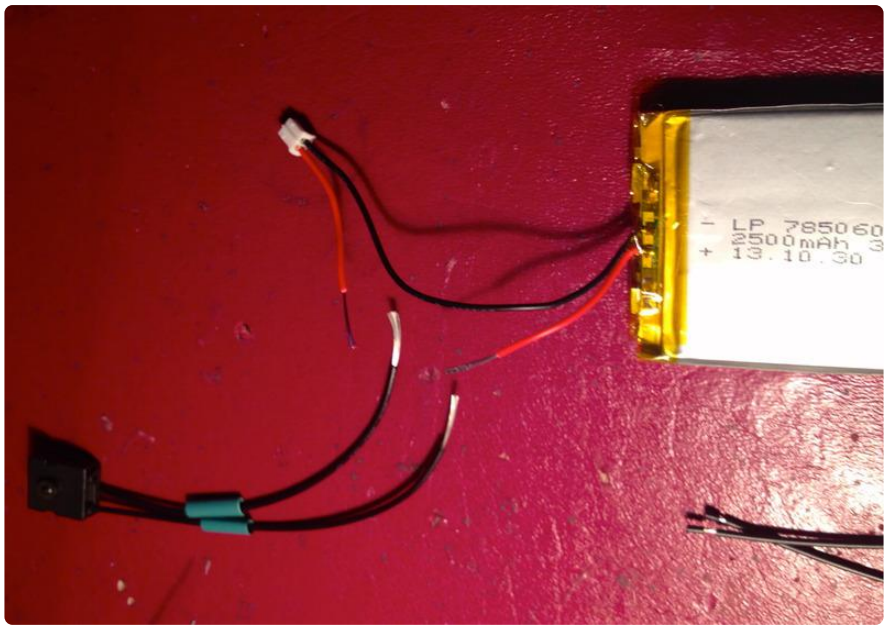
However, they do have their drawbacks -- it's good to know what the dangers are before you commit.

LiPo batteries are pretty volatile. If they get wet or get punctured, they can catch fire.. not a good thing when you've got them involved with your clothing.

They are also rather delicate. The batteries come with a handy JST connector, but the solder points going into the battery terminal are small and brittle. To keep them from breaking or doing anything dangerous, we need to add some strain relief.

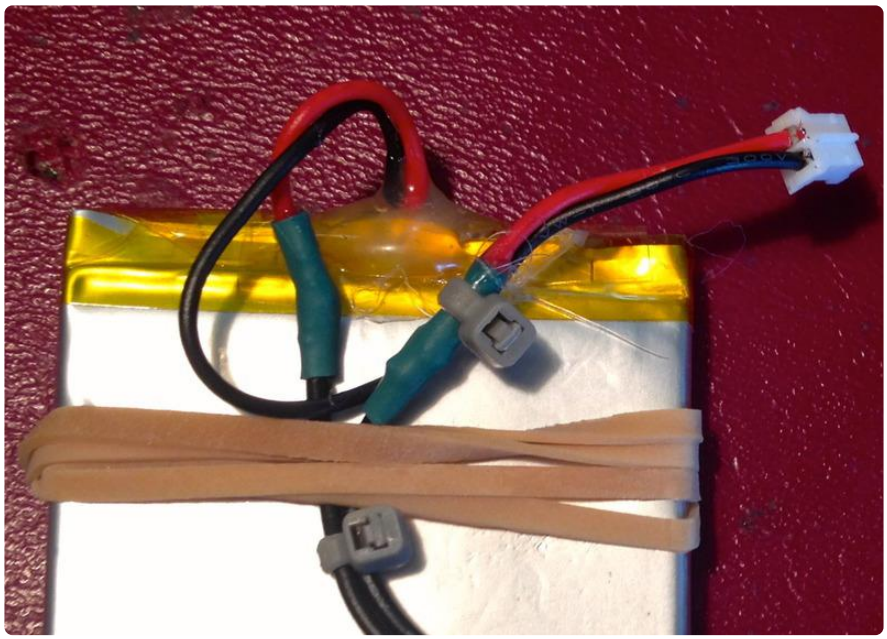
I also like to add a switch to these batteries, as pulling the connector in and out of your microcontroller places wear and tear on the wires. Having a switch is easier and safer all around.

If you're particularly safety-minded (e.g. if the scarf will be worn by a child), the LiPo cell can be replaced by a 3xAA battery holder (http://adafru.it/771), with single-use alkaline cells. The battery holder already has a switch built in. Add a JST connector by cutting and splicing one end of a JST battery extension cable (http://adafru.it/1131). You'll need to edit the code to limit the brightness to a level the alkaline cells can sustain. Try a brightness value of 50 to start and work up or down from there.



Cut the red power wire in the middle and strip about 1/2" of insulation from each side. Strip the same amount from both leads of your switch, and slide some heat shrink on to each one.

Connect one lead to each power wire. Wind the wires around a whole bunch of times before your solder -- you want these connections really solid. Once the connections are good, slide the heat shrink down and shrink it in place.



Using a rubber band, secure the wires to the battery in such a way that tugging on the JST connector or on the switch doesn't pull on the wires where they connect into the battery. Don't use cable ties around the battery itself, or pull it too tight! You do NOT want to puncture this battery! We're trying to make it safer, not make it explode.

For good measure, add a blob of hot glue to the point where the wires connect to the battery to discourage them even more from disconnecting.

LED Assembly

Cut LEDs

Decide how long you want your scarf to be. Mine is just shy of 1.5 yds long, but if you're tall you may want it longer.

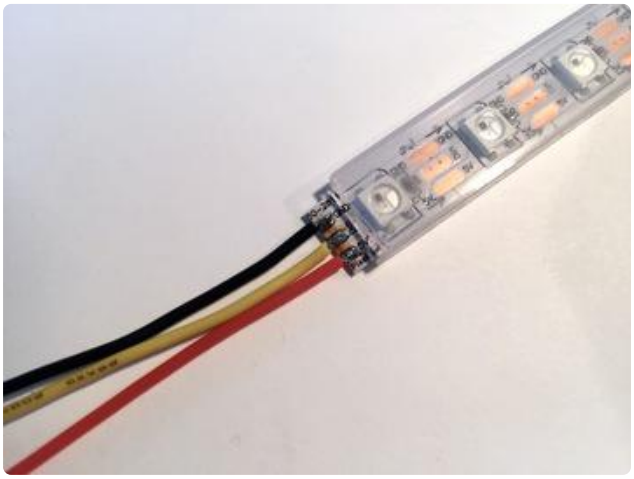
The gemma board will run up to around 100 pixels. It doesn't have the processing power for more, so if you have more than that in your scarf you may want to use a Flora () or a Circuit Playground ().

Lay out the fun fur face down and mark your scarf length on the back. Then lay your LED strip on top of the fur and cut it carefully to the same length along the cut line.

LED Wiring

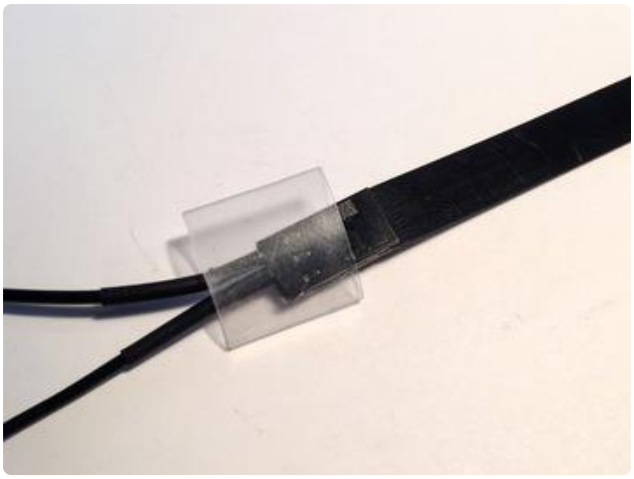
Look at the LED strip and figure out which is the "in" end (the arrows will be pointing down the strip away from this end).

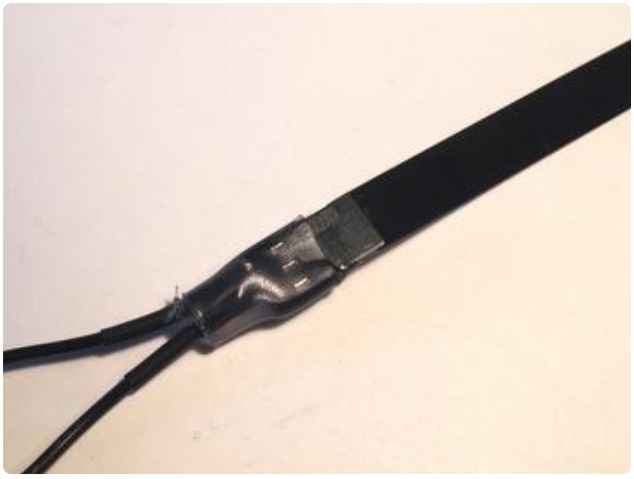
If you're using a brand-new strip and there's a connector soldered on, your job is easy. Cut off the connector so you have bare wires to solder onto the gemma.



If there's no connector soldered on, you'll need to add your own wires.

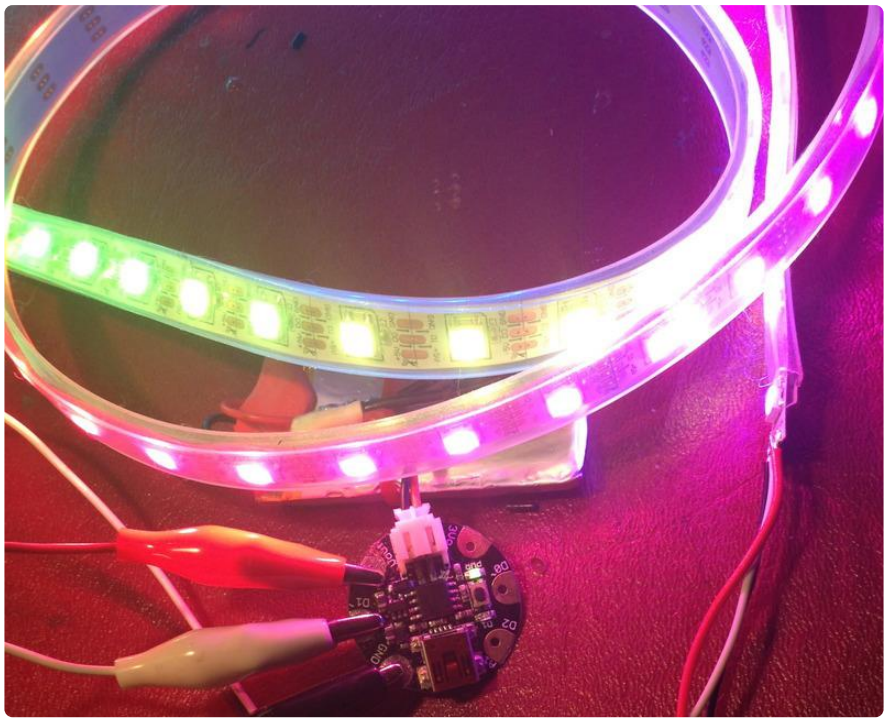
Tin the three solder pads with your soldering iron. Strip a tiny bit of insulation off the end of three wires and solder them on to the three pads. I used red for 5V, yellow for DI, and black for G.





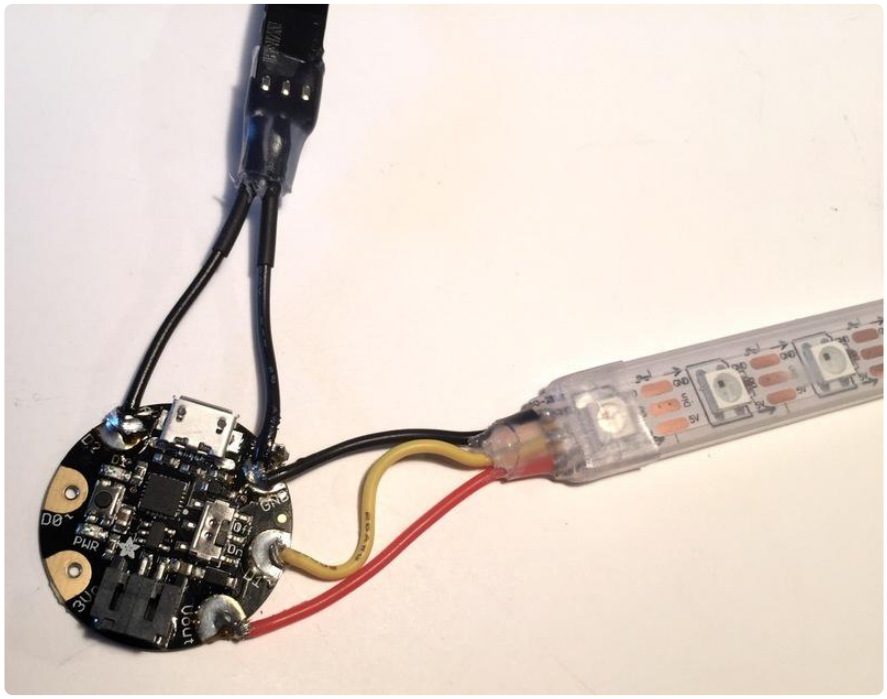
Membrane Switch Wiring

Take your jumper wire and snip it in half. Slide one pin into the center slot and the other into the left-hand slot of the membrane switch. They don't like to stay put, so slide a piece of heat shrink over the whole connector, fill the heat shrink with hot glue, and shrink it down with your heat gun.



Testing

Plug your battery into your gemma. Secure the wires with alligator clips (or just wind them tightly in place). Your lights will come on -- this is a great time to test your switch and make sure all your lights are working correctly before you solder everything together.

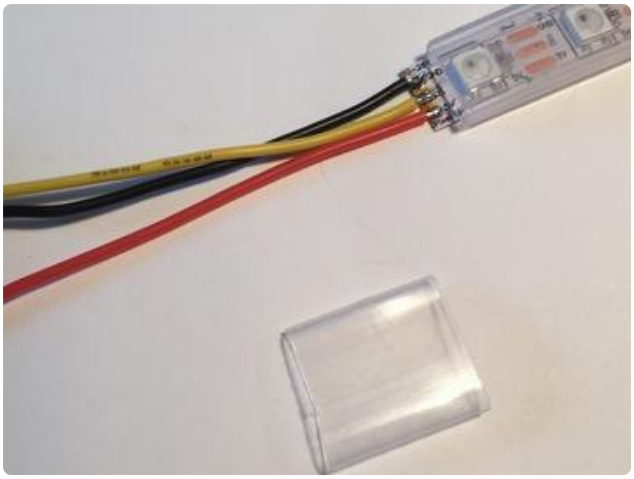


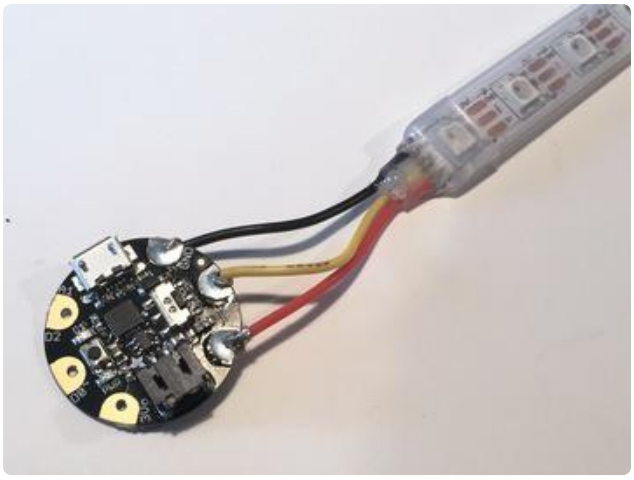
Gemma Wiring

Solder all 5 wires to the Gemma:

- Switch left pin to Gemma D2
- Switch center pin to Gemma GND
- LEDs 5v to Gemma Vout
- LEDs GND to Gemma GND
- LEDs DIN to Gemma D1

Notice that you have two wires both going to the GND pad on the Gemma! Be sure they're both in place before you solder.





Finishing

Now it's time to seal up the LED strip to keep it weatherproof and minimize breakage.

Slide a piece of clear 1/2' heat shrink over the LED strip. With the heat shrink positioned over the solder connections, fill up the area inside the heat shrink with hot glue.

Then, take a heat gun or hair dryer and point it at the heat shrink to shrink it down. It's best to do this quickly, before the glue sets up.

This will create a plastic seal that will greatly increase the strength of your connections.

Do the same at the other end of the LED strip.

Create the Scarf

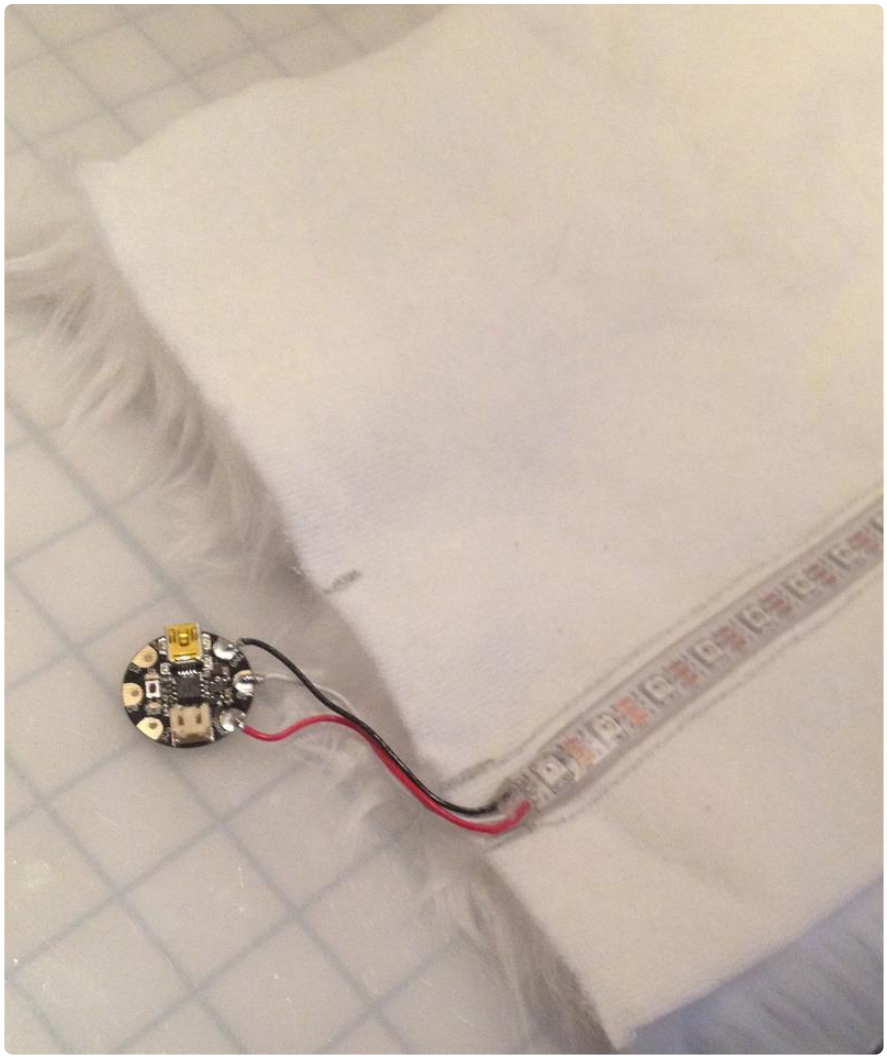
Pattern & Cutting

The scarf pattern is a simple rectangle:

- Length: 1.5 to 2 yds
- Width: 8.5 inches

Lay out your fur face down and measure. Mark your cut lines on the back with a marking pen.

Fun fur doesn't like to be cut with scissors! It does, however, very much like to be ripped with your bare hands. Make a small incision with scissors at the beginning of your 1.5 yard cut line, then grab the fur and tear it down the line. (Be sure to make your best Incredible Hulk face and enjoy the delicious tearing sound.) Repeat for the 8.5" side.

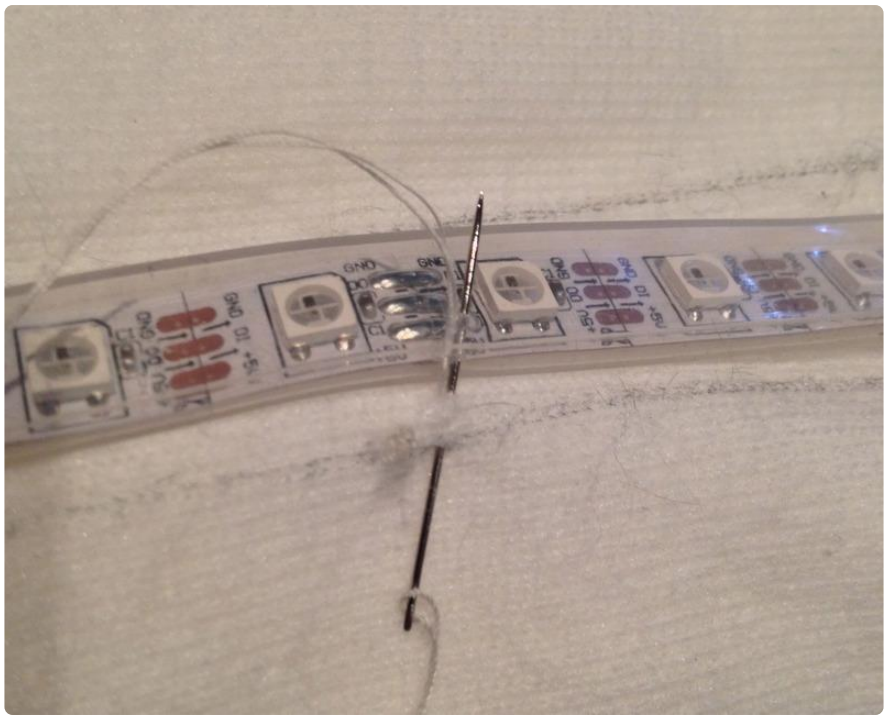


Placing the LEDs

Fold the scarf in half lengthwise and make a crease, then unfold and mark the center line on your crease.

Fold the edge in to the center mark and make another crease, to find the 1/4 line, and mark that.

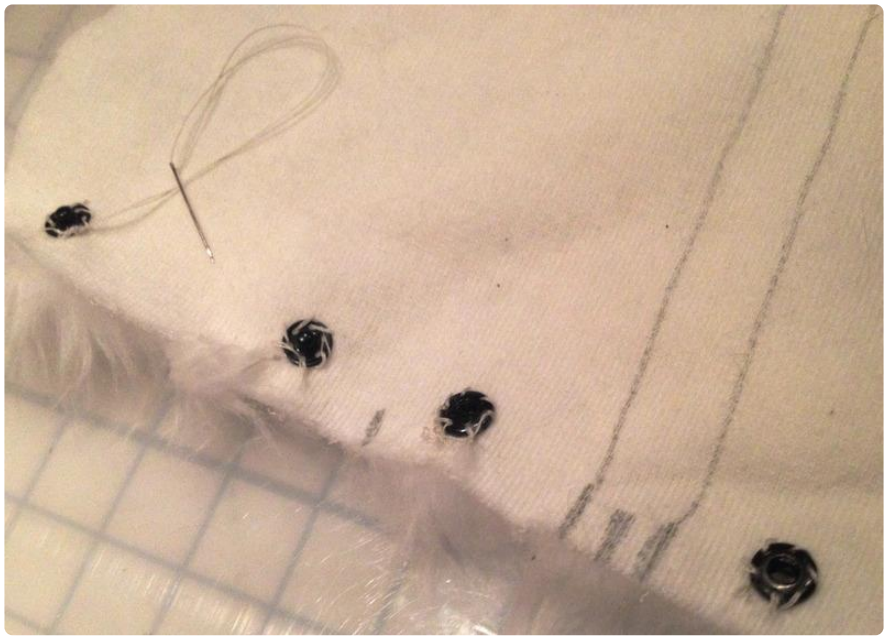
Place your LED strand along the 1/4 mark, making sure the ends don't extend past the fur. With a needle and thread, hand-stitch the fun fur to the silicone sleeve of the LEDS every 6-8 inches to hold it in place. Be careful not to stick the needle through the LED strand, just through the silicone.



Secure the Electronics

Once your LEDs are secure, stitch the Gemma to the fun fur through the two unusued pads. Secure the membrane switch right along the short edge of the fur by stitching over it several times.

Sew two heavy-duty snaps along the edge of the scarf, folding it along the center line to be sure the snaps line up.



Sew It Together

Once all your components feel secure, fold the scarf in half lengthwise, fur-side in (inside-out).

Stitch carefully along the short side opposite the Gemma and snaps, being careful not to hit the LED strand. Stitch lengthwise along the raw edge, pushing the fur inside as much as possible as you sew. A sewing machine is helpful here, but you can do it by hand with a little patience. Backstitch to secure at the end of the long side.

Carefully turn your scarf right-side out through the open end. Be careful not to kink the LED strip...though “flexible,” it doesn’t take kindly to sharp bends.

Plug in the battery, snap the snaps closed, and switch your scarf on.

Care & Feeding

You can hand-wash this scarf, or spot clean it if it gets dirty. Just be sure to remove the battery before washing and let it get completely and totally dry before turning it on again.

Be prepared for lots of attention while wearing this scarf!