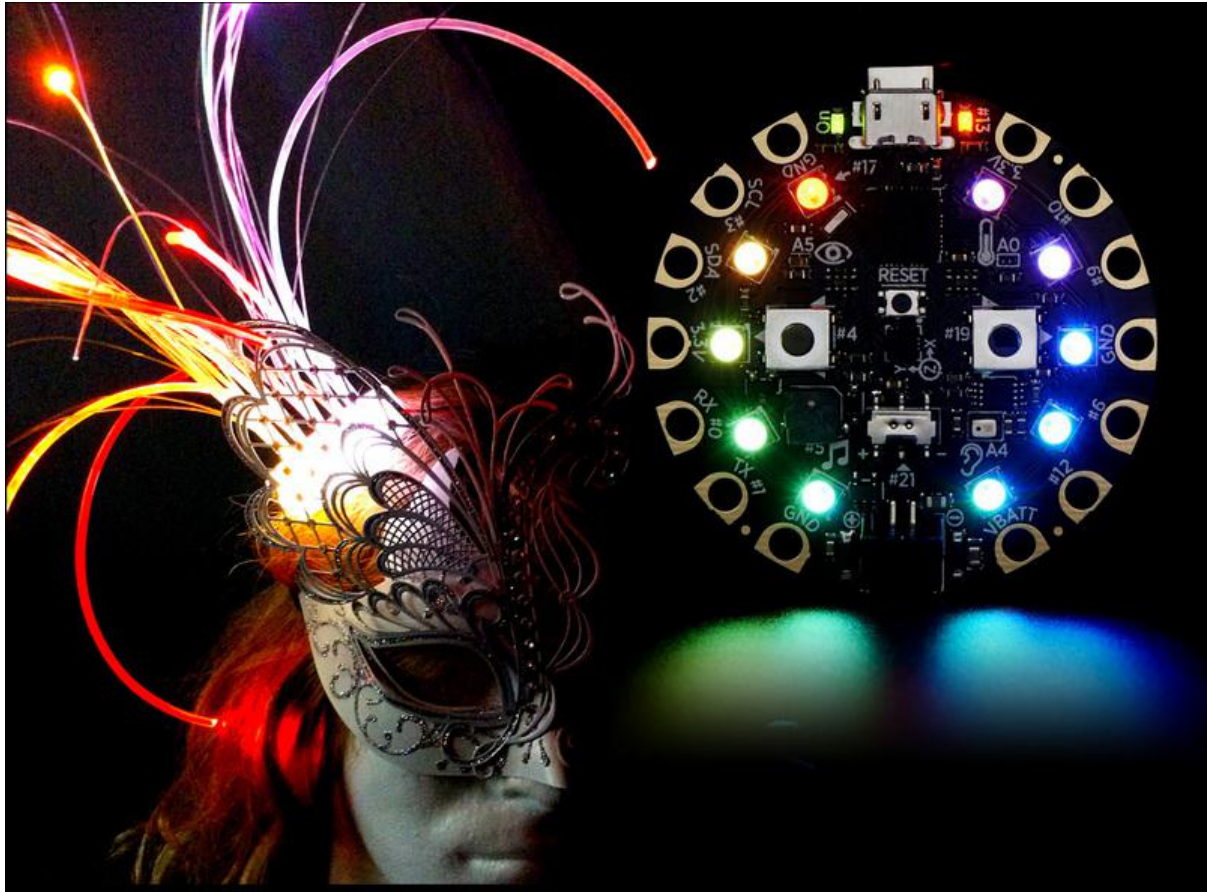




# Circuit Playground & Fiber Optics

Created by Erin St Blaine



<https://learn.adafruit.com/circuit-playground-light-pod>

Last updated on 2023-08-29 03:19:40 PM EDT

# Table of Contents

<b>Intro &amp; Setup</b>	<b>3</b>
<ul style="list-style-type: none"><li>• Materials</li><li>• Tools</li><li>• Before You Start</li></ul>	
<b>Code</b>	<b>4</b>
<ul style="list-style-type: none"><li>• Code Samples</li></ul>	
<b>3d Printing</b>	<b>7</b>
<ul style="list-style-type: none"><li>• Slice Settings</li></ul>	
<b>Assembly</b>	<b>9</b>
<b>Project Ideas</b>	<b>12</b>

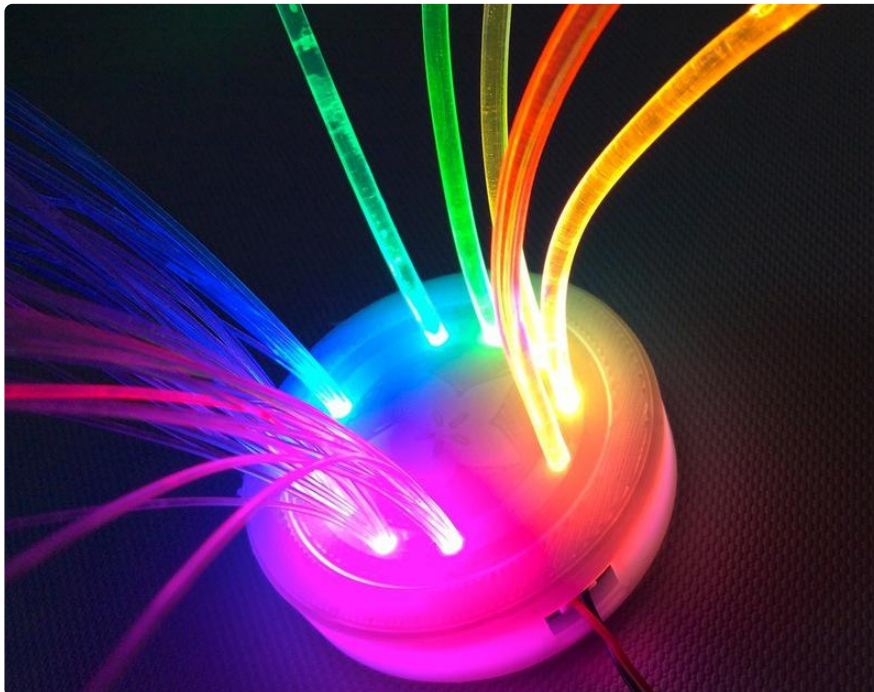
---

# Intro & Setup

Circuit Playground is a really fun, really easy to use micro controller that is a great way to add sensors and interactivity to your project without a whole lot of fuss and soldering.

It's got a built in motion sensor, a sound sensor, a light sensor, and a temperature sensor, as well as a built in speaker and 10 neopixels, ready to use right out of the box. I'm just starting to explore the fun things this board can do.

Today we're going to play with the onboard neopixels and add fiber optics or light pipe to make your project dance and shine.



## Materials

- Circuit Playground
- Battery (LiPoly, AAA or USB) - optional
- Light Pipe or Fiber Optics
- Hot Glue or Superglue

## Tools

- 3d Printer
- Sharp scissors or snips

Note: If you don't have a 3d Printer, you can easily just order the case from [Shapeways](#) ([s \(\)](#)).

## Before You Start

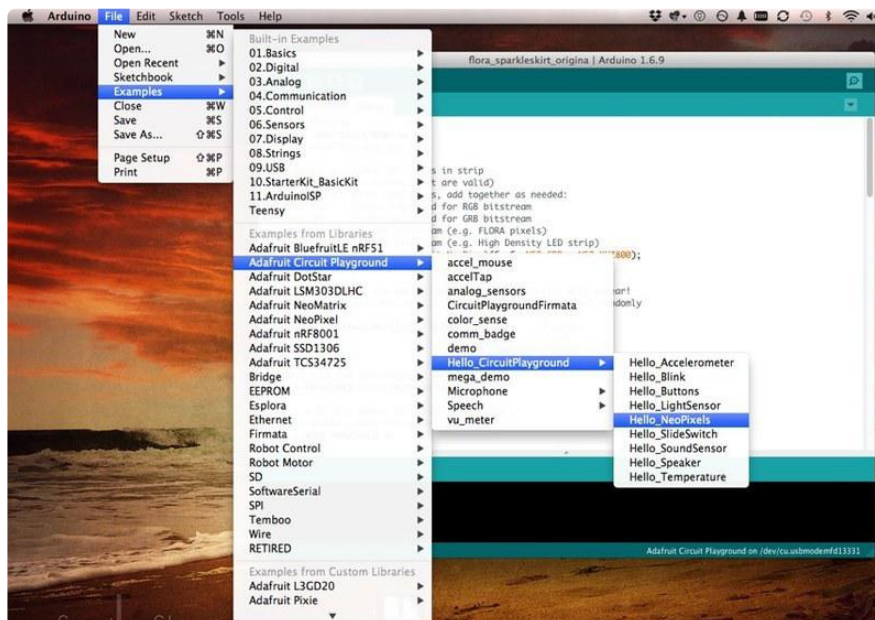
If this is your first foray into the world of arduino-based microcontrollers, you'll need to install some software first. Head over to the [Circuit Playground Lesson 0 guide \(\)](#) for detailed installation and setup instructions.

You'll only need to do all this once, so be a little patient if it seems like a lot! This is actually the hardest part.

## Code

Accessing the onboard neopixels is easy to do. There is some example code to get you started in the Circuit Playground library. In Arduino, open:

Examples > Adafruit Circuit Playground > Hello\_CircuitPlayground > Hello\_NeoPixels



This code shows two different ways to send color data to your neopixels, and an easy way to blink them on and off. Take a look at these five lines:

```
CircuitPlayground.setPixelColor(0, 255, 0, 0);  
CircuitPlayground.setPixelColor(1, 128, 128, 0);  
CircuitPlayground.setPixelColor(2, 0, 255, 0);
```

```
CircuitPlayground.setPixelColor(3, 0, 128, 128);  
CircuitPlayground.setPixelColor(4, 0, 0, 255);
```

The first argument in each line -- in this case, 0-4 -- is the pixel number. (The pixels are numbered 0-9, since counting that way makes computers unreasonably happy.)

When looking at the board with the USB port facing up, pixel 0 is located at 11:00, just to the left of the port. You can light up any pixel by calling its pixel number here.

The next three numbers correspond to values for red, green, and blue. 0 is off, and 255 is full brightness for that color. The way neopixels work is by mixing different amounts of light from the red, green, and blue LEDs hidden inside each pixel. Play with changing these numbers within a range of 0-255 to get the color you want.

```
CircuitPlayground.setPixelColor(6, 0x808000);  
CircuitPlayground.setPixelColor(7, 0x00FF00);  
CircuitPlayground.setPixelColor(8, 0x008080);  
CircuitPlayground.setPixelColor(9, 0x0000FF);
```

The next 5 lines show a slightly different way to set the color. The first argument still corresponds to the pixel number, and the next part is a Hexidecimal code -- you can always recognize these because you'll see the prefix "0x".

Hex codes are an easy way to get any color you want. Look for online [color pickers](#) () or desktop widgets to make your life easy. However, the colors will not always look the same on your neopixels as they do on your screen, especially with darker colors, so be ready for some experimentation.

These are just a couple of quick ways to get your lights up and running. Neopixels are also wonderful for running animations or reacting to some of the Circuit Playground's onboard sensors. Here are a few other guides and code samples that are fun to play with.

## Code Samples

- [Motion Sensor Response](#) ()
- [Analog Input Response](#) ()
- [Simon Game](#) ()
- [Adalight for Circuit Playground](#) ()
- [Light Paintbrush](#) ()
- [Magic Wand & Sword](#) ()

You'll also find a lot more samples to play with in the Examples folder in the Circuit Playground library, including `vu_meter` which lights up the pixels in response to sound.

Here's one more sample that gently animates the pixels in a rainbow or one of several other color palettes. The left button will change modes and the right button will turn the LEDs off.

NOTE: You will need to install the FastLED library in Arduino first (Sketch > Include Library > Manage Libraries...)

One other note: if you're using FastLED with Circuit Playground, be sure to include the Circuit Playground library FIRST and the FastLED library second, or you may run into problems.

```
#include <Adafruit_CircuitPlayground.h>;
#include <FastLED.h>;

#define LED_PIN    17
#define NUM_LEDS   10
#define COLOR_ORDER GRB

int BRIGHTNESS =255;
int STEPS      =25;
int NUM_MODES  =5;

CRGB leds[NUM_LEDS];

CRGBPalette16 currentPalette;
TBlendType    currentBlending;

int ledMode = 0;
bool leftButtonPressed;
bool rightButtonPressed;

void setup() {
  CircuitPlayground.begin();
  FastLED.addLeds<WS2812B, LED_PIN, COLOR_ORDER>(leds,
NUM_LEDS).setCorrection( TypicalLEDStrip );
  currentBlending = LINEARBLEND;
}

void loop() {

  leftButtonPressed = CircuitPlayground.leftButton();
  rightButtonPressed = CircuitPlayground.rightButton();

  if (leftButtonPressed) {
    ledMode=ledMode+1;
    delay(300);
    if (ledMode > NUM_MODES){
      ledMode=0;
    }
  }
  if (rightButtonPressed) {
    ledMode=99;
  }
}
```

```

switch (ledMode) {
  case 0: currentPalette = RainbowColors_p; rainbow(); break;
  case 1: currentPalette = OceanColors_p; rainbow(); break;
  case 2: currentPalette = ForestColors_p; rainbow();
break;
  case 3: currentPalette = PartyColors_p; rainbow(); break;
  case 4: currentPalette = HeatColors_p; rainbow(); break;
  case 5: currentPalette = RainbowStripeColors_p; rainbow(); break;
  case 99: CircuitPlayground.clearPixels(); break;

}
}
void rainbow()
{

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

  FillLEDSFromPaletteColors( startIndex);

  FastLED.show();
  FastLED.delay(10);}

//this bit is in every palette mode, needs to be in there just once
void FillLEDSFromPaletteColors( uint8_t colorIndex)
{
  uint8_t brightness = BRIGHTNESS;

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

```

---

## 3d Printing

This is a remix of the [3d Printed Circuit Playground Yo Yo project \(\)](#) by the Ruiz Brothers.

There are 4 different pieces to print: the base, the top, the buttons, and the cover which holds the fiber optics or light pipe.

If you don't have access to a 3d printer, you can order the whole case ready-to-go from Shapeways.



[Download from Thingiverse](#)

[Order from Shapeways](#)

## Slice Settings

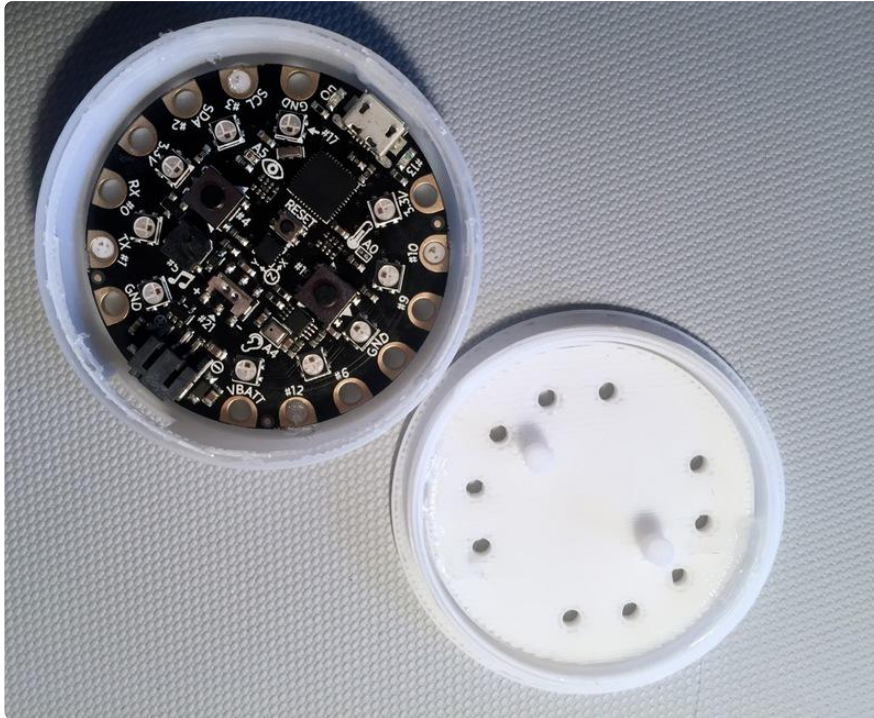
Below are some recommended slice settings. These parts were printed on an Up! Mini.

- .15mm layer height
- 220C extruder / PrintInZ Skin
- 20% infill
- 4 top/bottom layers
- 2 shells / parameters
- 40mm/s printing speed



---

# Assembly



Place the Circuit Playground inside the base, with the USB port facing the smaller hole and the battery connector lined up with the larger hole. Secure it with a little dab of hot glue on the case's pillars.

Set the 3d printed button pieces aside for a minute.

Place the cover into the top and tape it in place from the outside. Screw the top onto the base. Once the top is tight, adjust the cover's orientation so the holes line up perfectly with the neopixels and buttons.

Unscrew the top, being careful to keep the orientation right, and glue the cover and top together securely.



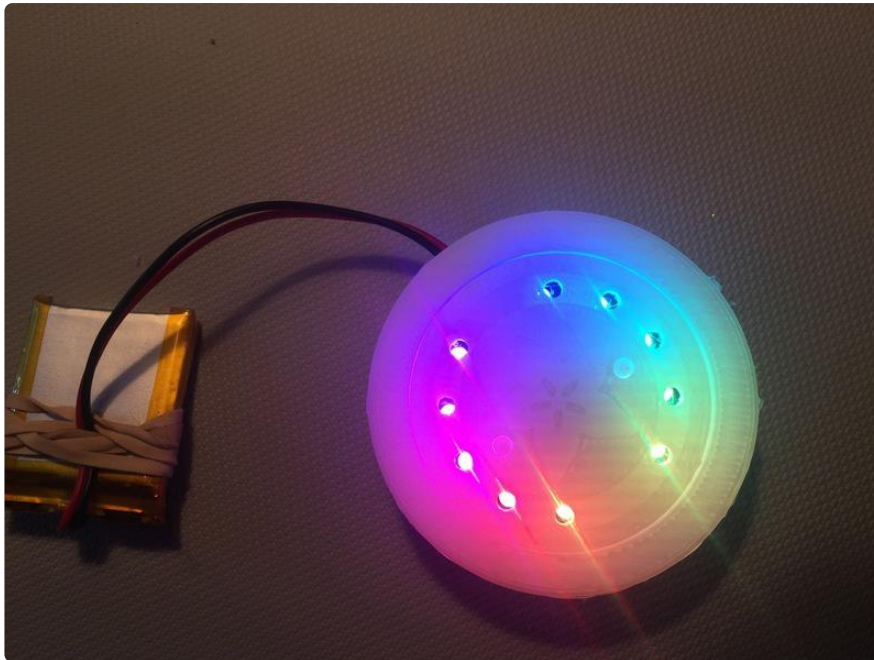
Turn the lid upside down and set the buttons small-side-down into the button holes. Screw or press the base with the Circuit Playground inside into place so the neopixels are aligned and the buttons contact the Circuit Playground's buttons.

My 3d printed threads didn't hold very well, so I secured the top with hot glue.

Plug your battery or USB cable in and make sure the lights are lined up perfectly through the holes and the buttons are satisfyingly clicky.

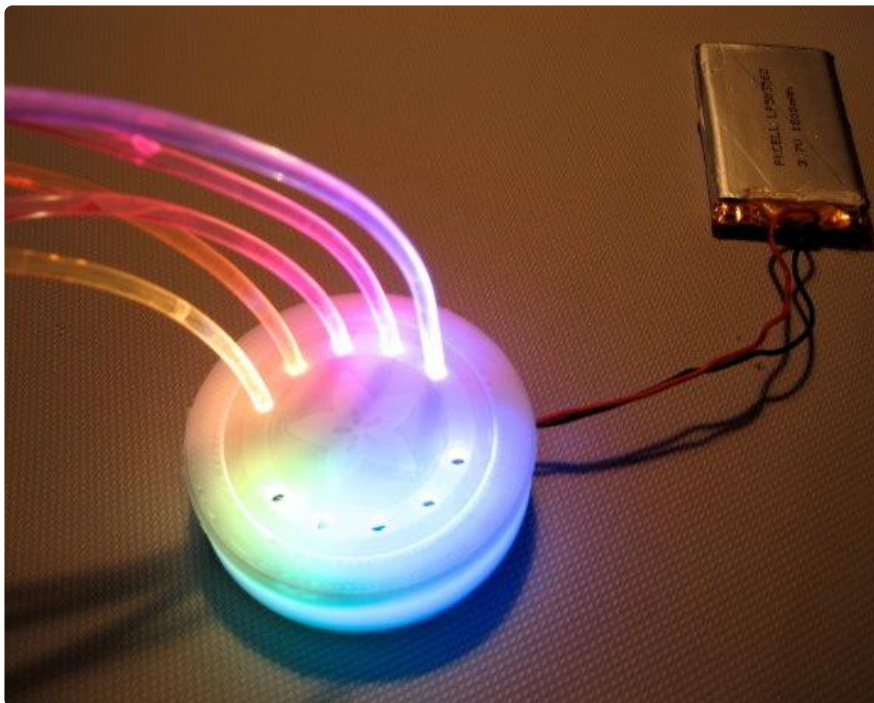
The left button will cycle through color modes and the right button will turn the LEDs off.

Warning! this button does not turn the Circuit Playground board off entirely, so if you're running from a battery, keeping it semi-off like this will drain your battery eventually -- unplug your battery if you want it off completely.



The holes are sized to snugly fit 2.75mm light pipe. The best source I've found for this is these cheesy [light up shoelaces](#) () (available everywhere!)

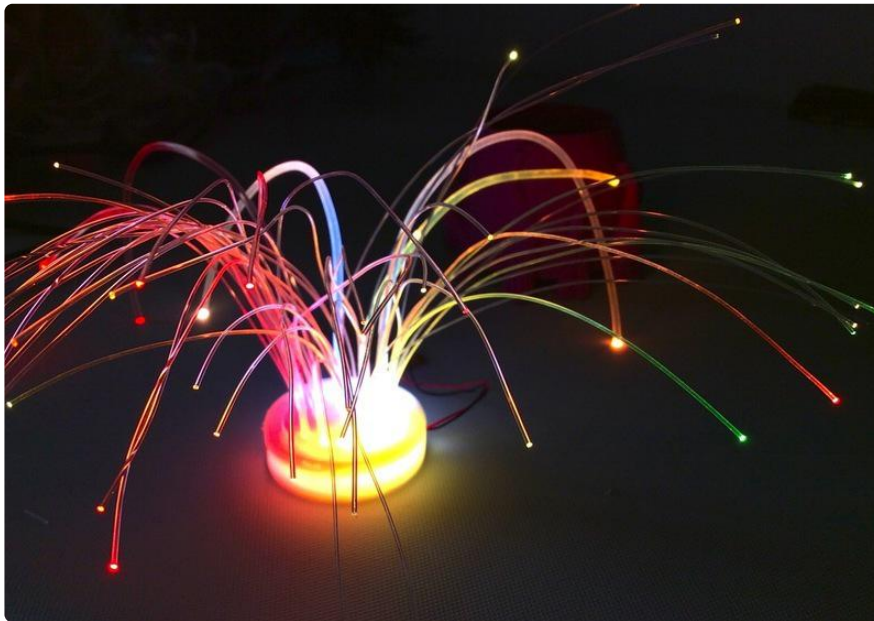
With your LEDs on full brightness these will glow for up to 3 feet, though the light does get quite a bit dimmer toward the end. I find they're most satisfying if kept to a length of 1-1.5 feet.



Or you can use fiber optics. There are two different kinds: end glow and side glow. The side glow are my favorite since they glow along their entire length. End glow are useful if you just want a single floating point of light at the end of each strand, also a beautiful effect.

My favorite source for fiber optic cable is [Weidamark lighting \(\)](#). They've got loads of options and sell the fibers by the foot. (Note: fiber optic solid core side glow appears to be the same stuff as the LED shoelace light pipe mentioned above). Order in any size bundles; you can easily separate them down to the right size bunches.

I can fit about 10 fibers in each 3d printed hole, so if you use all 10 holes you'll want around 100 fibers. Like the light pipe, these seem to be brightest and cheeriest in the first 1.5 feet or so. After that they do still glow but get a little dimmer. If you need more length than that, I'd recommend going with end-glow.



Trim the fibers sharply and cleanly with a pair of snips or a hot knife. The cleaner the cut, the brighter the fiber will be. Start by slipping around 8 fibers into the hole, and then keep adding fibers to the middle of the bundle until they're nice and snug.

If this will be going on a costume or art car, or something that will move a lot, it doesn't hurt to add some extra hot glue around the outside of the fiber holes to secure them.

**DO NOT USE SUPERGLUE HERE.** Most superglues will degrade the fiber optic cable and it'll stop glowing and break. Sad Christmas.

---

## Project Ideas

Well that was easy. But.... what do I DO with this thing?

- Mardi Gras Mask accessory
- Top Hat Topper

- Kid's science fair volcano
- Glowing puppet hair
- Flower Bouquet Enhancer
- Pot Holder
- Night Light
- Showgirl Headdress

You can also use end-glow light pipe to put glowing eyes or spots of color into hard-to-reach areas of your costume or project, without messing around with breakable wires. Post your project in the [Adafruit Forum!](#)





