Glowing Beehive Hairdo Wig

Created by Erin St Blaine

https://learn.adafruit.com/glowing-beehive-hairdo-wig

Last updated on 2024-06-03 02:12:58 PM EDT

Table of Contents

Introduction

This is the hairstyle of tomorrow! A press-button dream come true for Mrs. Housewife. All sorts of wonders are hers at the push of a button.   Going to an office party? Walk in with the belle of the ball, for a change! Family music hour? Use sound reactive mode and Presto! Now you have something else to pay attention to. And no more stumbling around in the dark after lights-out. Mrs. Housewife will feel like a fairy princess with a magic wand that lights your way.

Skill Level

This project is great for beginners who are looking to level up. Soldering to the Circuit Playground is pretty simple with just 3 wires to hook up. Or this project can be done with no soldering if you skip adding the ambient Neopixel strip and just use the onboard LEDs and light pipe.

We've included several different color modes and a sound reactive mode in the code. You can use our code as-is with a little bit of software installation, or you can code-monkey up and use any of the Circuit Playground's onboard sensors to make this wig into a multi-tasking robotic alien funk machine. Go nuts!

1 x Circuit Playground Circuit Playground Classic	https://www.adafruit.com/product/3000
1 x On/Off Switch On/Off switch with battery cable extension	https://www.adafruit.com/product/3064
1 x 60/m Neopixels White 60/m RGB Neopixels	https://www.adafruit.com/product/1138
1 x Battery 500mAh LiPoly Battery	https://www.adafruit.com/product/1578
1 x Battery Charger USB Battery Charger	https://www.adafruit.com/product/1304

Other Tools & Materials

- White High-Cone Beehive Wig (https://adafru.it/Bsb)
- Light Pipe (https://adafru.it/x2E)
- 1/2" clear Heat Shrink Tubing (https://adafru.it/Bsc)
- 26awg Stranded Wire (in 3 colors) (http://adafru.it/1877)
- Soldering Iron + accessories
- Needle & thread
- Craft Foam
- Super glue
- Hot Glue Gun
- Heat Gun

Code

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 (https://adafru.it/tGC) 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!

FastLED Library

You will also need to install the **FastLED** library in Arduino (`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.

Upload Code

Once you've got everything installed and your computer can talk to the Circuit Playground, it's time to upload the code.

Plug your Circuit Playground into your computer and select the Circuit Plaground under `Tools > Boards`. Then select the Circuit Playground as the Port.

Copy and paste this code into a new Arduino window and click "upload".

```
// SPDX-FileCopyrightText: 2017 Erin St Blaine for Adafruit Industries
//
// SPDX-License-Identifier: MIT

// Code by Erin St Blaine for Adafruit.com
// Full tutorial at https://learn.adafruit.com/glowing-beehive-hairdo-wig/

#include <Adafruit_CircuitPlayground.h>  // add this before the FastLED library to 
avoid issues
#include <FastLED.h>

#define LED_PIN     6    //led strand is soldered to pin 6
#define CP_PIN      17   //circuit playground's neopixels live on pin 17
#define NUM_LEDS    12   // number of LEDs in my strand
#define NUM_CP      10   // number of neopixels on the circuit playground

// I have purposefully mixed up the color order on the Neopixel strip to
// quickly and easily add color variety to this code -- This way the strip will 
always
```

```
// show a different color from the light pipe.

#define COLOR_ORDER GRB
#define COLOR_ORDER_STRIP  RBG

uint8_t brightness = 255;  //led strand brightness control
uint8_t cpbrightness = 255;  //circuit playground brightness control

int STEPS = 25;  //makes the rainbow colors more or less spread out
int NUM_MODES = 4;  // change this number if you add or subtract modes

CRGB leds[NUM_LEDS];  //I've set up different arrays for the neopixel strand and
the circuit playground
CRGB cp[NUM_CP];      // so that we can control the brightness separately

CRGBPalette16 currentPalette;
TBlendType    currentBlending;

int ledMode = 0;       //Initial mode
bool leftButtonPressed;
bool rightButtonPressed;

// SOUND REACTIVE SETUP --------------

#define MIC_PIN         A4                                  // Analog port for
microphone
#define DC_OFFSET  0                                        // DC offset in mic
signal - if unusure, leave 0
                                                            // I calculated this
value by serialprintln lots of mic values
#define NOISE     200                                       // Noise/hum/
interference in mic signal and increased value until it went quiet
#define SAMPLES   60                                        // Length of buffer
for dynamic level adjustment
#define TOP (NUM_LEDS + 2)                                  // Allow dot to go
slightly off scale
#define PEAK_FALL 10                                        // Rate of peak
falling dot

byte
  peak      = 0,                                            // Used for falling
dot
  dotCount  = 0,                                            // Frame counter for
delaying dot-falling speed
  volCount  = 0;                                            // Frame counter for
storing past volume data
int
  vol[SAMPLES],                                             // Collection of
prior volume samples
  lvl       = 10,                                           // Current audio
level, change this number to adjust sensitivity
  minLvlAvg = 0,                                            // For dynamic
adjustment of graph low & high
  maxLvlAvg = 512;


// SETUP -------------------

void setup() {
  Serial.begin(57600);
  CircuitPlayground.begin();
  FastLED.addLeds<WS2812B, LED_PIN, COLOR_ORDER_STRIP>(leds,
NUM_LEDS).setCorrection( TypicalLEDStrip );
  FastLED.addLeds<WS2812B, CP_PIN, COLOR_ORDER>(cp, 10).setCorrection(
TypicalLEDStrip );
  currentBlending = LINEARBLEND;
  set_max_power_in_volts_and_milliamps(5, 500);               // FastLED 2.1 Power
management set at 5V, 500mA
```

```
}

void loop()  {

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

  if (leftButtonPressed) {  //left button cycles through modes
    clearpixels();
    ledMode=ledMode+1;
    delay(300);
    if (ledMode > NUM_MODES){
    ledMode=0;
     }
  }
    if (rightButtonPressed) {   // right button turns all leds off
    ledMode=99;

    }

 switch (ledMode) {
       case 0: currentPalette = RainbowColors_p; rainbow(); break;
       case 1: soundreactive(); break;
       case 2: currentPalette = OceanColors_p; rainbow(); break;
       case 3: currentPalette = LavaColors_p; rainbow(); break;
       case 4: currentPalette = RainbowStripeColors_p; rainbow(); break;
       case 99: clearpixels(); break;

}
}

void clearpixels()
{
  CircuitPlayground.clearPixels();
  for (int i = 0; i < NUM_LEDS; i++) leds[i] = CRGB::Black;
  for (int i = 0; i < NUM_CP; i++) cp[i] = CRGB::Black;
  FastLED.show();
}

void rainbow()
{

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

  FillLEDsFromPaletteColors( startIndex);

  FastLED.show();
  FastLED.delay(20);}

//this bit is in every palette mode, needs to be in there just once
void FillLEDsFromPaletteColors( uint8_t colorIndex)
{
  for (int i = 0; i < NUM_LEDS; i++) leds[i] = ColorFromPalette( currentPalette,
colorIndex, brightness, currentBlending);
  for (int i = 0; i < NUM_CP; i++) cp[i] = ColorFromPalette( currentPalette,
colorIndex, cpbrightness, currentBlending);
    colorIndex += 25;

}


void soundreactive() {

  uint8_t  i;
  uint16_t minLvl, maxLvl;
  int      n, height;
  n = analogRead(MIC_PIN);                                    // Raw reading from
mic
  n = abs(n - 512 - DC_OFFSET);                               // Center on zero

  n = (n <= NOISE) ? 0 : (n - NOISE);                         // Remove noise/hum
  lvl = ((lvl * 7) + n) >> 3;                                 // "Dampened" reading
(else looks twitchy)

  // Calculate bar height based on dynamic min/max levels (fixed point):
  height = TOP * (lvl - minLvlAvg) / (long)(maxLvlAvg - minLvlAvg);

  if (height < 0L)       height = 0;                          // Clip output
  else if (height > TOP) height = TOP;
  if (height > peak)     peak   = height;                     // Keep 'peak' dot at
top


  // Color pixels based on rainbow gradient -- led strand
  for (i=0; i<NUM_LEDS; i++) {
    if (i >= height)   leds[i].setRGB( 0, 0,0);
    else leds[i] = CHSV(map(i,0,NUM_LEDS-1,0,255), 255, brightness);  //constrain
colors here by changing HSV values
  }

  // Draw peak dot  -- led strand
  if (peak > 0 && peak <= NUM_LEDS-1) leds[peak] = CHSV(map(peak,0,NUM_LEDS-1,0,
255), 255, brightness);

  // Color pixels based on rainbow gradient  -- circuit playground
  for (i=0; i<NUM_CP; i++) {
    if (i >= height)   cp[i].setRGB( 0, 0,0);
    else cp[i] = CHSV(map(i,0,NUM_CP-1,0,255), 255, cpbrightness);  //constrain
colors here by changing HSV values
  }

  // Draw peak dot  -- circuit playground
  if (peak > 0 && peak <= NUM_CP-1) cp[peak] = CHSV(map(peak,0,NUM_LEDS-1,0,255),
255, cpbrightness);

// Every few frames, make the peak pixel drop by 1:

    if (++dotCount >= PEAK_FALL) {                            // fall rate
      if(peak > 0) peak--;
      dotCount = 0;
    }

  vol[volCount] = n;                                          // Save sample for
dynamic leveling
  if (++volCount >= SAMPLES) volCount = 0;                    // Advance/rollover
sample counter

  // Get volume range of prior frames
  minLvl = maxLvl = vol[0];
  for (i=1; i<SAMPLES; i++) {
    if (vol[i] < minLvl)      minLvl = vol[i];
    else if (vol[i] > maxLvl) maxLvl = vol[i];
  }
  // minLvl and maxLvl indicate the volume range over prior frames, used
  // for vertically scaling the output graph (so it looks interesting
  // regardless of volume level).  If they're too close together though
  // (e.g. at very low volume levels) the graph becomes super coarse
  // and 'jumpy'...so keep some minimum distance between them (this
  // also lets the graph go to zero when no sound is playing):
  if((maxLvl - minLvl) < TOP) maxLvl = minLvl + TOP;
  minLvlAvg = (minLvlAvg * 63 + minLvl) >> 6;                 // Dampen min/max
levels
  maxLvlAvg = (maxLvlAvg * 63 + maxLvl) >> 6;                 // (fake rolling
average)
```

```
show_at_max_brightness_for_power();                         // Power managed
FastLED display
  Serial.println(LEDS.getFPS());
}
```

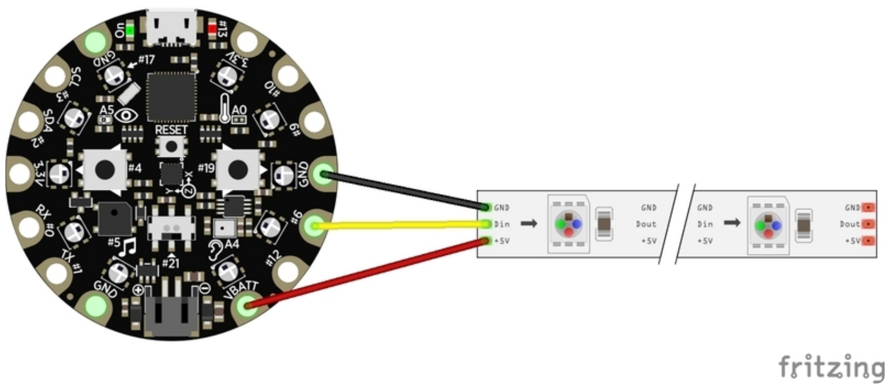
If all goes well, the lights on the Circuit Playground will come on. Press the right side button and the lights will go off. Press the left side button to toggle between modes:

1. Rainbow Colors
2. Sound Reactive
3. Ocean Colors
4. Lava Colors
5. Rainbow Stripe Colors

Press the button once to enter Sound Reactive mode. Make some noise!

Add your own modes or customize the modes that are already there. If you add more modes, remember to change the NUM_MODES variable to reflect the total number of modes you have.

Wiring Diagram

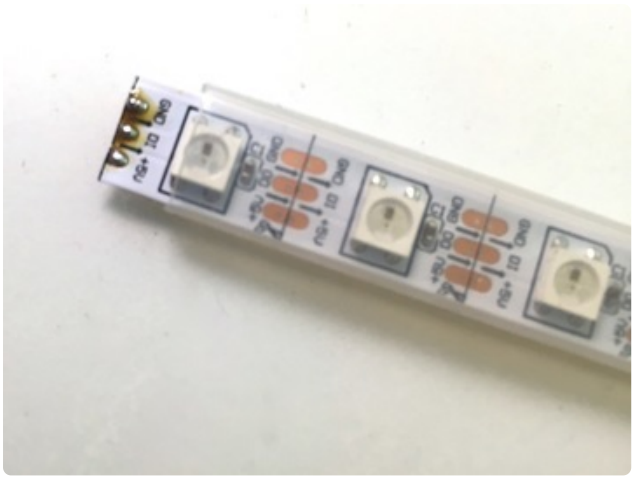


The wiring for this project is fairly simple:

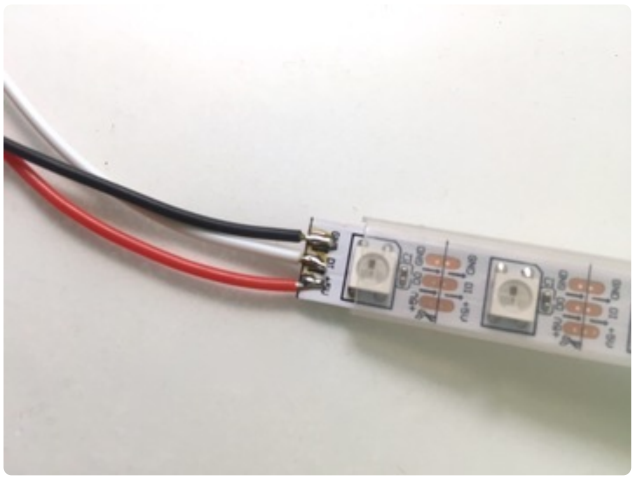
- Circuit Playground VBATT --> Neopixel +5V
- Circuit Playground 6 --> Neopixel DIN
- Circuit Playground GND --> Neopixel GND

The battery extension cable will plug into the JST connector on the Circuit Playground, and the battery will plug into the extension cable.

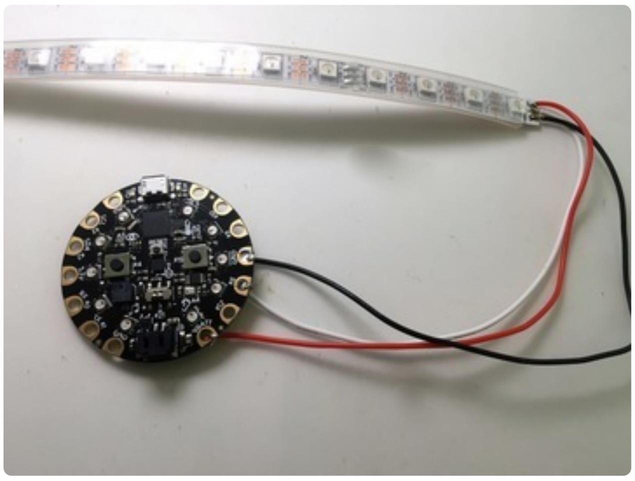
Connect Neopixels



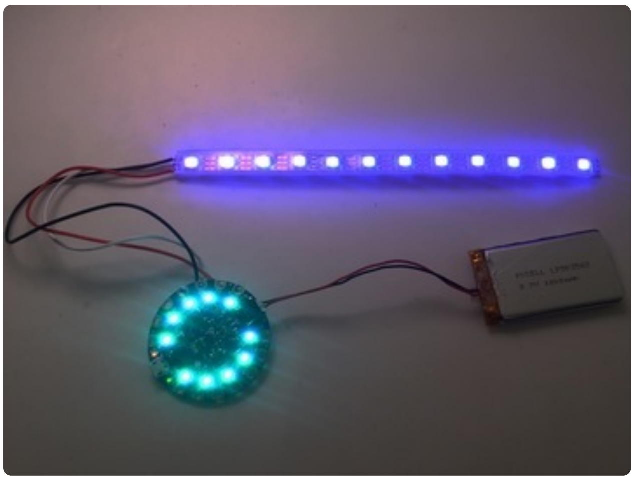
Slide the neopixels a little way out of their silicone sleeve. Find the "in" end of the strip (arrows pointing away from you) and tin the three solder pads with your soldering iron.



Cut three 6" pieces of stranded wire. Strip a teeny bit off one end and tin the exposed wire. Solder red to the + pad, black to G, and white to the DIN pad.



Solder the other ends of the wires to the Circuit Playground: red to VBAT, white to pin 6, and black to GND.



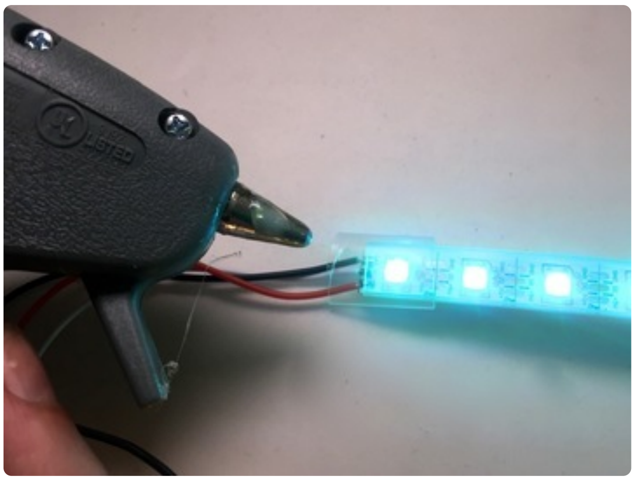
Plug your battery in to your Circuit Playground and be sure all the lights come on.   Press the button and see the modes change.

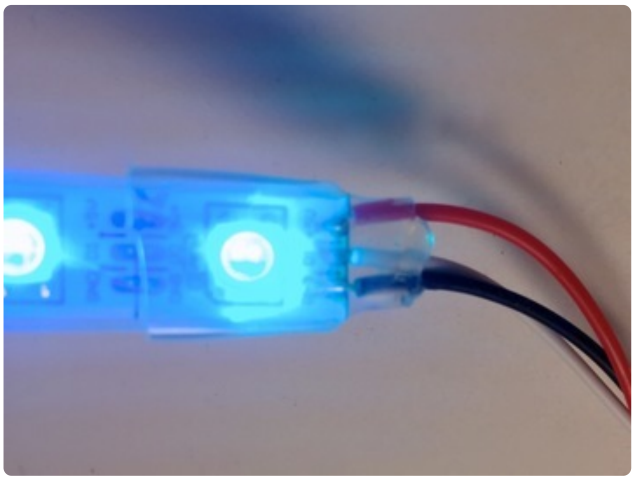
Troubleshooting

If the lights don't come on on both the Circuit Playground and the LED strip, here are some things to try:

- Re-upload the code to your Circuit Playground
- Check to be sure you soldered to the "in" end of the pixel strip -- it won't work if you solder to the "out" end
- Be sure you're soldered to pin 6 on the Circuit Playground
- Make sure there are no shorts; i.e. your wires or solder joints aren't touching each other where they shouldn't be

Once you've got everything working, it's time to seal off the neopixel strips to keep them nice and safe from sweat or bumps in the night.





Cut a small piece of 1/2" clear heat shrink and slide it over your solder joint. Squirt some hot glue inside, then quickly grab your heat gun and shrink down the heat shrink while the glue is still wet. This will create a solid plastic seal around your connection and keep it safe from just about everything.

Connect Light Pipe

There are lots of different sources for light pipe, and light pipe from different sources acts differently. This attachment method requires light pipe that melts at a fairly low temperature. These LED shoelaces (https://adafru.it/x2E) work wonderfully for this.

I also tried using good quality LED light pipe from AliExpress (https://adafru.it/zAn). This stuff worked beautifully in that it carries light really far -- this light pipe looks gorgeous for a couple meters whereas the shoelaces really only look good for about 2 feet. But.. the good stuff doesn't melt the same way.

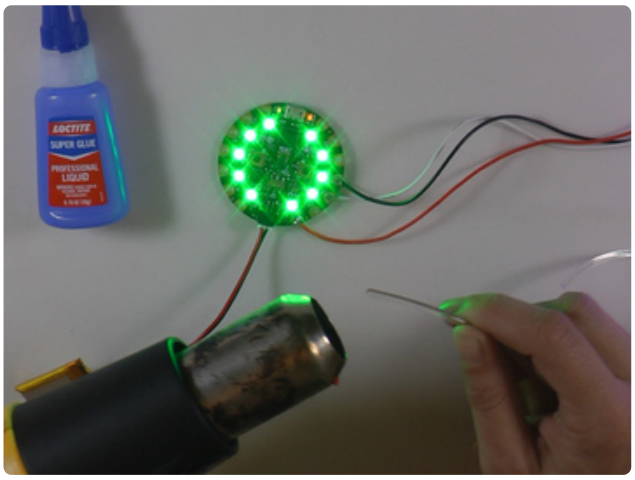
If your light pipe doesn't want to melt, you can 3d print a case that will work just about as well:

Download .stl from Thingiverse

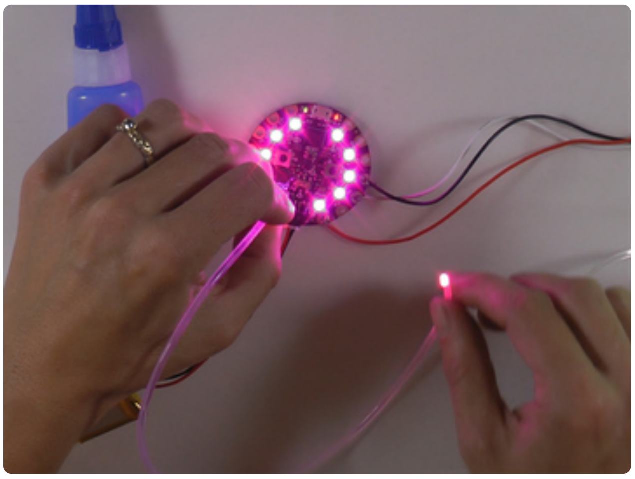
https://adafru.it/zBA

.. or you can order one printed for you from Shapeways (https://adafru.it/zBB).

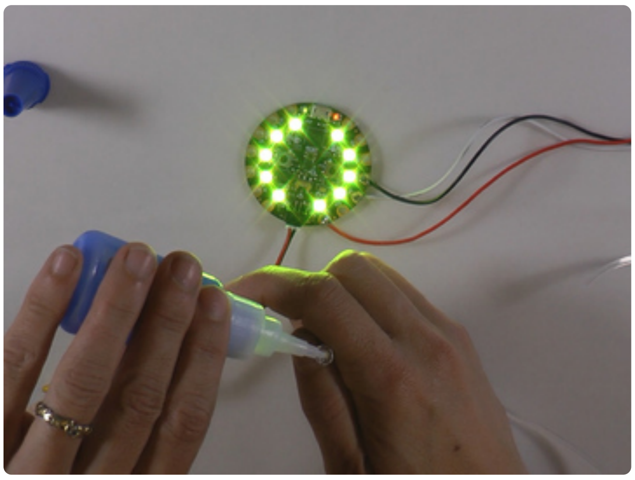
If you're going this route, check out this guide about making a 3d printed case that holds the light pipe in place. (https://adafru.it/zAo) If you've got the melty kind, read on!



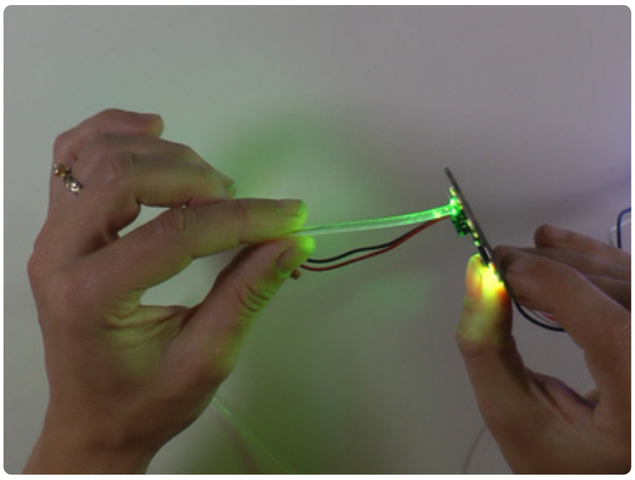
Trim your light pipe to the desired length (I cut each shoelace in half). Gently heat the end of each piece with a heat gun until it just barely liquefies.



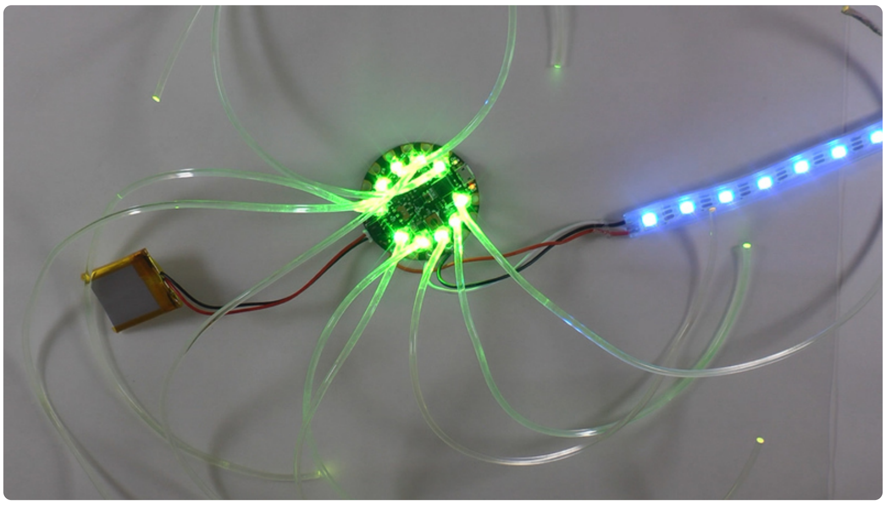
Position the light pipe over one of your neopixels and press down so about 1/4" of the light pipe melts over and around the pixel. Make sure the pipe is glowing brightly, so you know you're centered on the light.



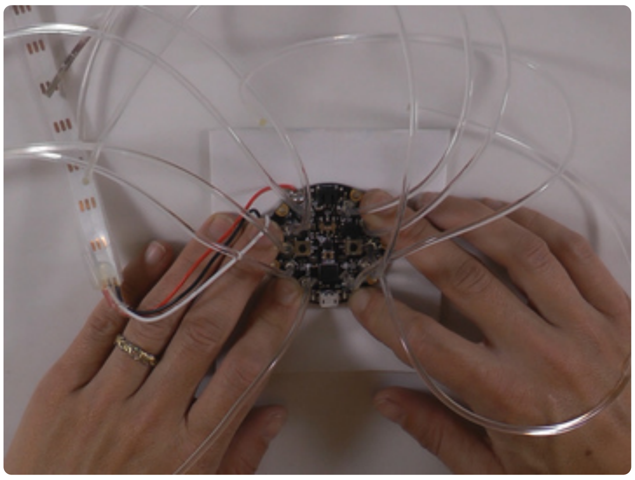
Once the light pipe is cooled and solidified, gently pull it off the neopixel. Add a dab of superglue to the inside and press it back down, lining up the indentation so you get full contact with the LED.



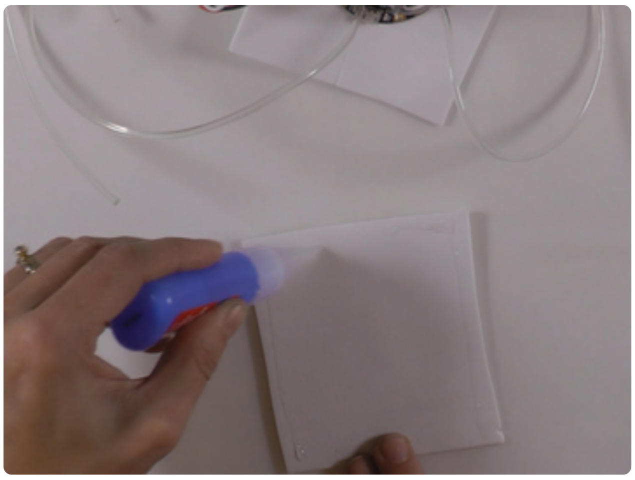
Once the superglue is dry, tug on the light pipe gently to be sure it won't come loose. The superglue holds it really well! I tried to remove a piece of light pipe and ended up breaking the neopixel before the glue gave way.



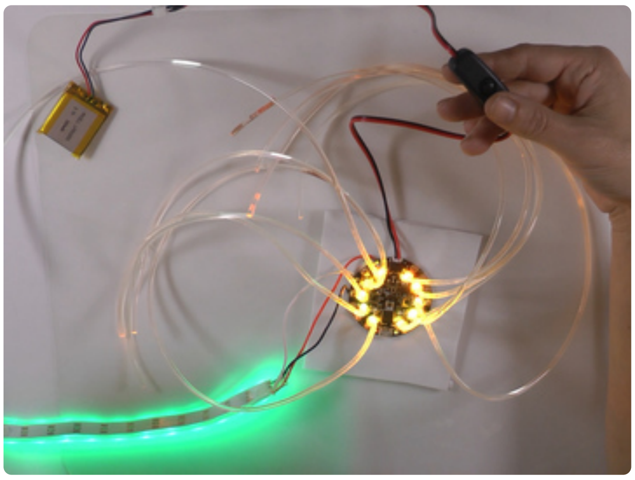
Final Assembly

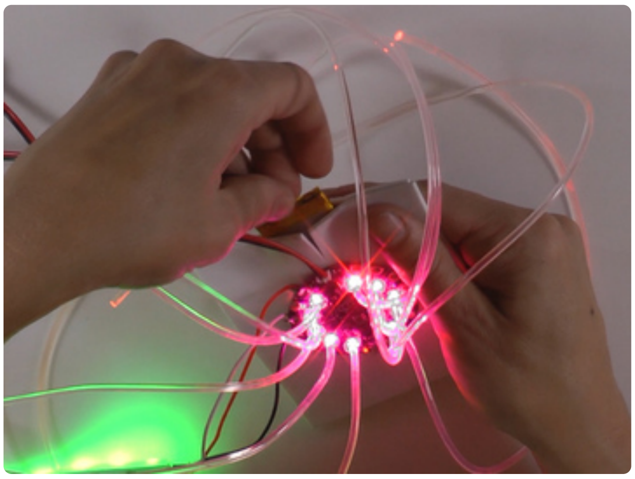


Cut two squares of craft foam approximately 4" across. Use superglue to attach the Circuit Playground in the center of one square.

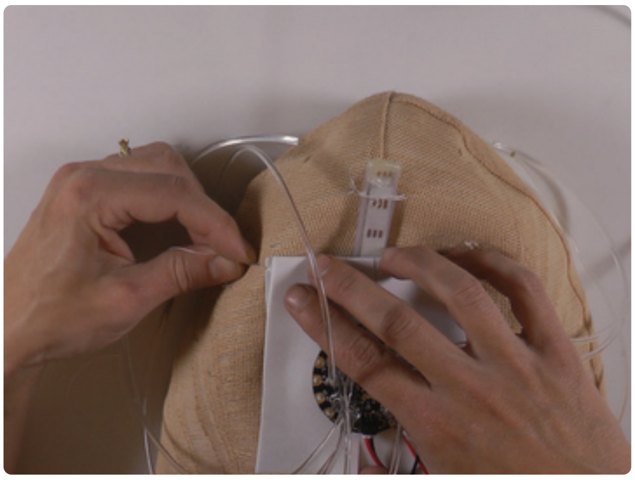
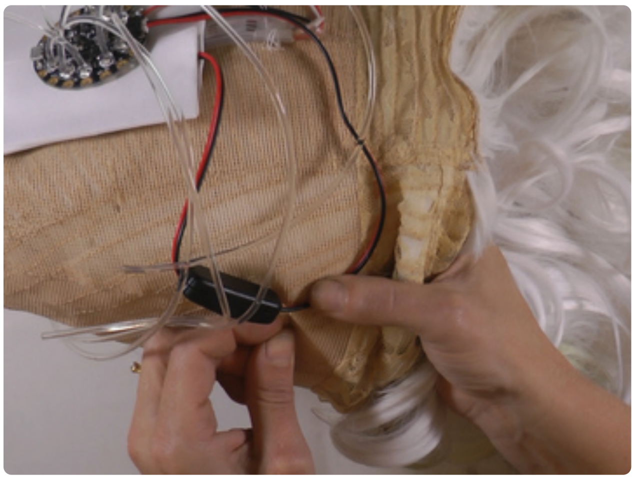


Add superglue around 3 sides of the other square and glue it to the back of the first square to create a pocket for the battery.

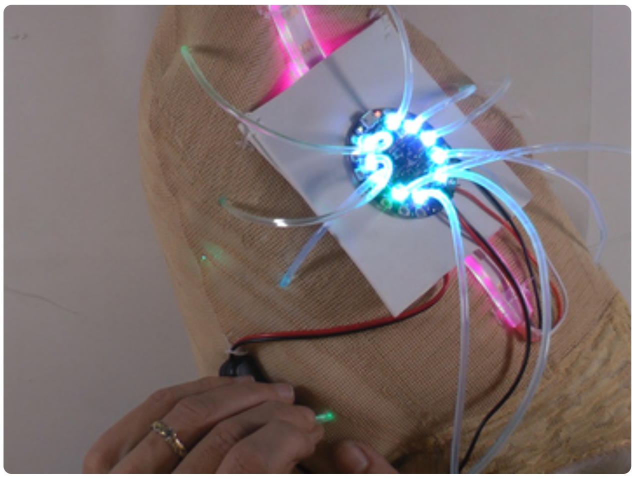




Plug your battery extension cable / switch into the Circuit Playground and your battery into the other end. Make sure everything's working, and then slip the battery and any extra wire inside the craft foam pocket.

Turn the wig inside out. Lay the neopixel strip face down (so the lights are pointing toward the outside of the wig) and sew it in place through the silicone sleeve. Sew the craft foam in place over the back of the strip. Sew the switch in place along one of the side seams of the wig.



Clip small holes in the fabric of the wig and slip the individual light pipe strands through so they end up on the outside of the wig when it's right-side out.

Pick your favorite mode and bring on the atompunk revolution.