# Dotstar LED and Glass Pebble Floor

Created by Erin St Blaine

https://learn.adafruit.com/neopixel-and-glass-pebble-floor

Last updated on 2022-12-01 02:58:33 PM EST

# Table of Contents

# Introduction

Line your walkway or edge your room with Dotstar LEDs submerged beneath colored glass pebbles. Circuit Playground's onboard buttons give you an easy on/off switch and control of the modes.  Add as many modes as you want! Customize the color palettes and animations to suit your mood.

If you've ever tried installing store-bought LED strips around your house and been disappointed with their annoying color-change modes, this guide is for you.  With just a little coding and soldering the possibilities become as limitless as your imagination.

You can add a sound reactive mode to make your own music-reactive dance floor, use Circuit Playground's onboard light sensor for automatic day/night modes, or use any of its other sensors to make this floor do whatever you can dream up.


# Electronic Parts

1 x Dotstar LEDs
30/m - Get enough so you have a few extra
https://www.adafruit.com/product/2238

1 x Circuit Playground
Circuit Playground Classic
https://www.adafruit.com/product/3000

1 x Power Supply
5v 10A Power Supply
https://www.adafruit.com/product/658

1 x Screw Terminal

I used around 300 pixels to circumnavigate my 2m x 3m dining room.  Larger installations may require additional power or code modifications.

# Other Stuff You'll Need

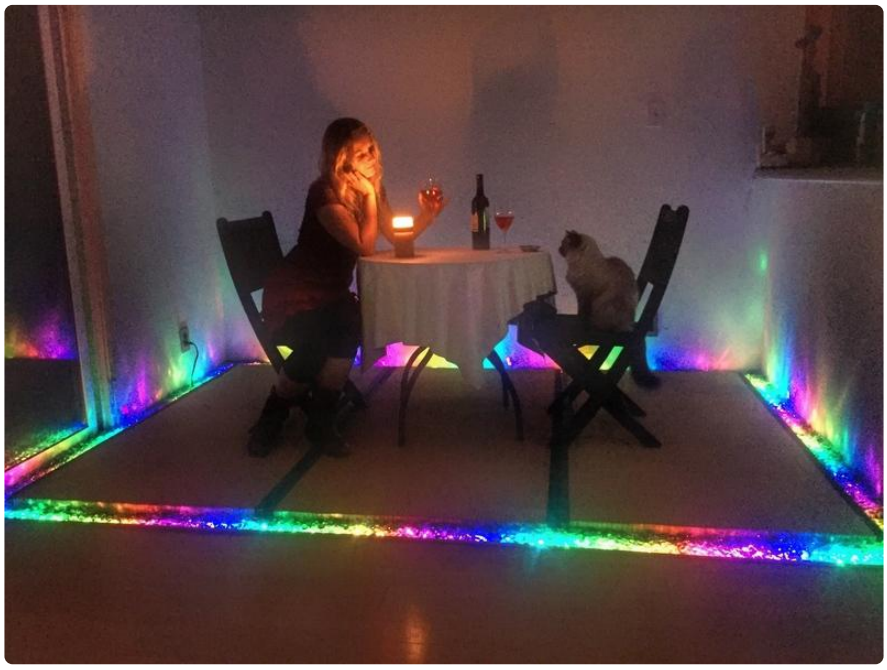
## Glass pebbles

These are sold in small quantities in the floral section of your local craft store, but you'll probably need to find larger quantities.  Shop around to find the best deal.  Search for "fireplace glass" or "glass pebbles in bulk".

Look at local hardware or fireplace supply stores as well as online, since shipping heavy stones can be cost-prohibitive.

## Tools

- Soldering Iron & accessories (starter kit here ())
- 3d printer (optional)
- USB cable () for uploading code



# Planning & Power

My dining room has three Japanese tatami mats centered on the subfloor, creating a 3-4" trench around the perimeter.  The trench is about 2" deep and butts up on the 4th-wall side against the raised bamboo flooring in my living room.

This design would also work really well to line the edges of a hallway or walkway, though for outdoor applications the controller and power supply would need to be weather-proofed.

I looked around for glass pebbles in bulk, and found the best deals online sold as "fireplace glass" ().  These appear to be mostly used for diffusion materials in classy restaurants' outdoor gas fire pits.

They come in a good range of colors and a couple different shapes.  The rounder pebbles seem to be much kinder to bare feet, so I went with that shape.  I decided on 80% aqua blue with a few darker green pebbles mixed in for contrast.

They come in 10 pound bags.  For my project I ended up using 10 bags of blue glass and 3 bags of green.  This filled a trench that was approximately 4" wide and 2" deep, and about 9.5m long.

# LED Density

Adafruit sells Dotstar strips in 30/m, 60/m or 144/m densities.  The higher density strips have a couple advantages:

- They are brighter overall: more lights per meter packs more of a punch
- The animations running on the lights look buttery smooth and scintillating.  Mmm.

They also have a couple of drawbacks:

- They cost more
- They need a lot more power
- You may run into mysterious limits with your microcontroller -- it can't always drive thousands of pixels

I wasn't sure which way to go, so I set up three test strands, each about 1/2m long, and loaded the same animations on all three.  Here is the result.

The 144/m lights do indeed look brighter and buttery-smooth.  However, they cost 4x as much as the 30/m, and well.. they just don't look 4x better.  That and the worry about heat and power and programming limits ruled these right out for me.

I thought there would be more difference between the 60/m and 30/m but they look almost identical.  The 30/m are slightly dimmer but are bright enough to look beautiful even during the daytime, and at night they look incredible.  The glass pebbles do a really good job of diffusing the pixels and spreading the light around.

Any of the three varieties will look beautiful.  If you choose to go with the buttery-smooth 144/m, you may want to inject power between each of your strips.

# Neopixels vs Dotstars

You could use either Neopixels or Dotstars for this project.  Neopixels are a little less expensive and require one less wire to hook up.  If you like this project as-is, either type of LED will work fine with just a small change to the code.

I chose to go with Dotstars because then I've got a little more freedom available in terms of coding.  I kept this project fairly simple, but someday I may decide I want to add an IR remote control, or do fancy home automation over Wifi or Bluetooth.  The Dotstar LEDs have a separate clock line, which more-or-less means you can interrupt an animation with a change command much more easily.  I've run into brick walls in my coding when I tried to get fancy with Neopixels in the past, and spending the extra $10 for Dotstars up-front may keep me from a lot of frustration down the line.

# Power

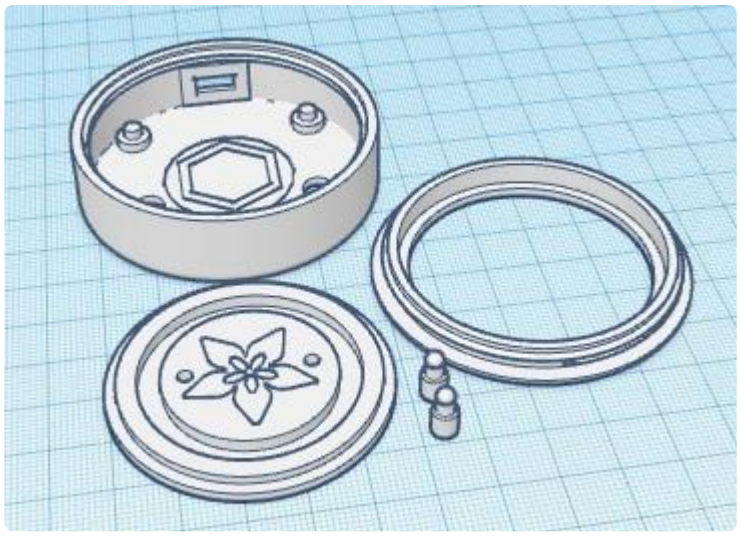
These strips want 5v power.  Using a lower voltage will give you flickering and brown-outs, and using a higher voltage will burn out your LEDs and controller.  Stick with 5v!

The amperage requirement will change depending on how many LEDs you have. Head over to the Neopixel Uberguide () to learn more about power requirements.  For up to around 300 LEDs I found the 5v 10A power supply to be totally sufficient.

---

# 3d Printing

The Circuit Playground will act as my on/off switch and mode controller, so I wanted to make it look pretty for mounting on my wall.



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.  There's also an optional belt clip you can glue to the back of the case, just in case that works with your mounting solution.

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

---

# 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 () 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".

```
// Code by Erin St Blaine for Adafruit.com, based on FastLED animations by Mark
Kriegsman

#include &lt;Adafruit_CircuitPlayground.h&gt;
#include &lt;FastLED.h&gt;
#include &lt;Wire.h&gt;
#include &lt;SPI.h&gt;
#include &lt;math.h&gt;

// tell FastLED all about the Circuit Playground's layout

#define CP_PIN      17   //circuit playground's neopixels live on pin 17
#define DATA_PIN     9    //LED data on pin 9
#define CLK_PIN     6     // Clock pin on pin 6
#define NUM_LEDS     300   // total number of LEDs in your strip
#define COLOR_ORDER BGR    // Dotstar color order -- change this if your colors are
coming out wrong
#define COLOR_ORDER_CP GRB  // Circuit Playground color order

uint8_t brightness = 250;  // Set brightness level

int STEPS = 6;  //makes the rainbow colors more or less spread out
int NUM_MODES = 5;  // change this number if you add or subtract modes
int CYCLETIME = 60;  // number of seconds on each mode, for mode cycling

CRGB leds[NUM_LEDS];      // set up an LED array

CRGBPalette16 currentPalette;
TBlendType    currentBlending;

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


// SETUP ----------------------------------------------------
```

```
void setup() {
  Serial.begin(57600);
  CircuitPlayground.begin();
  FastLED.addLeds&lt;WS2812B, CP_PIN, COLOR_ORDER_CP&gt;(leds,
10).setCorrection( TypicalLEDStrip );
  FastLED.addLeds&lt;DOTSTAR,DATA_PIN,CLK_PIN,COLOR_ORDER&gt;(leds,
NUM_LEDS).setCorrection(TypicalLEDStrip); // Use this line for if using dotstars
  //FastLED.addLeds&lt;WS2812B, DATA_PIN, COLOR_ORDER&gt;(leds,
NUM_LEDS).setCorrection( TypicalLEDStrip ); // Use this line if using neopixels
  currentBlending = LINEARBLEND;
  set_max_power_in_volts_and_milliamps(5, 5000);               // FastLED 2.1 Power
management set at 5V, 5000mA


}

void loop()  {

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

  if (leftButtonPressed) {  //left button cycles through modes
    clearpixels();
    ledMode=ledMode+1;
    delay(300);
    if (ledMode &gt; NUM_MODES){
    ledMode=0;
     }
  }
    if (rightButtonPressed) {  // on off button
    ledMode=99;

    }

 switch (ledMode) {
       case 0: modeCycle(); break;
       case 1: currentPalette = RainbowColors_p; rainbow(); break;
       case 2: currentPalette = OceanColors_p; rainbow(); break;
       case 3: currentPalette = LavaColors_p; rainbow(); break;
       case 4: currentPalette = ForestColors_p; rainbow(); break;
       case 5: currentPalette = PartyColors_p; rainbow(); break;
       case 99: clearpixels(); break;

}
}
void clearpixels()
{
  CircuitPlayground.clearPixels();
   for( int i = 0; i &lt; NUM_LEDS; i++) {
   leds[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 &lt; NUM_LEDS; i++) {
    leds[i] = ColorFromPalette( currentPalette, colorIndex, brightness,
currentBlending);
    colorIndex += STEPS;
  }
}


int cycleMode=0;

void modeCycle()
{
   switch (cycleMode) {
       case 0: currentPalette = RainbowColors_p; rainbow(); break;
       case 1: currentPalette = OceanColors_p; rainbow(); break;
       case 2: currentPalette = LavaColors_p; rainbow(); break;
       case 3: currentPalette = ForestColors_p; rainbow(); break;
       case 4: currentPalette = PartyColors_p; rainbow(); break;
       case 5: cycleMode=0; break;
}

   EVERY_N_SECONDS(CYCLETIME) {
    cycleMode++;
   }

}
```

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

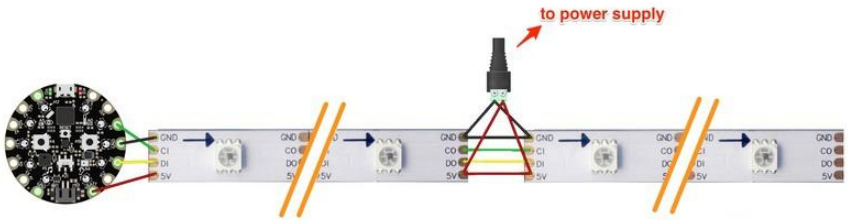
The initial mode will cycle through five different color modes, changing every 60 seconds.  Change this duration in the code with the CYCLETIME variable.

The other modes use FastLEDs built in color palettes for a variety of different looks.

Add your own modes or customize the modes that are already there.  If you want to add your own color palettes, check out this Neopixel Parasol guide () for ideas and instructions on how to do that with FastLED.

---

# Wiring



# Data Connections

The Circuit Playground must be connected to the "in" end of the LED strip.  Data flows one way through the strip, from "in" to "out".

| | |
|---|---|
| Circuit Playground G | Dotstar G |
| Circuit Playground 9 | Dotstar CI |
| Circuit Playground 6 | Dotstar DI |
| Cirucit Playground VBATT | Dotstar 5V |

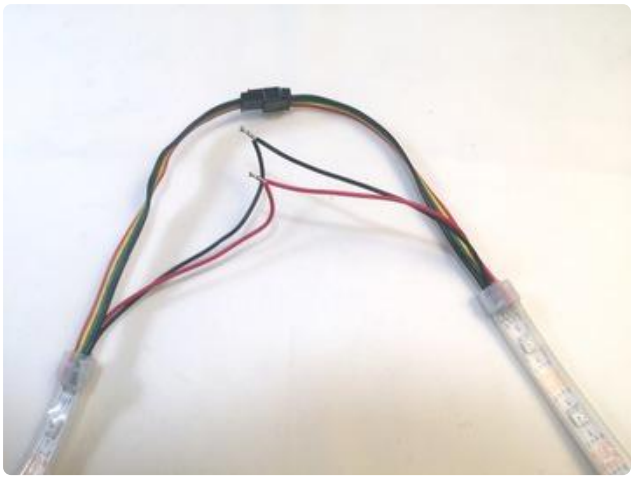
# Power Connections

The power supply will be connected in the middle of the strip.  Power flows happily either up or down the strip so we don't need to connect it at the "in" end like we did with the Circuit Playground.

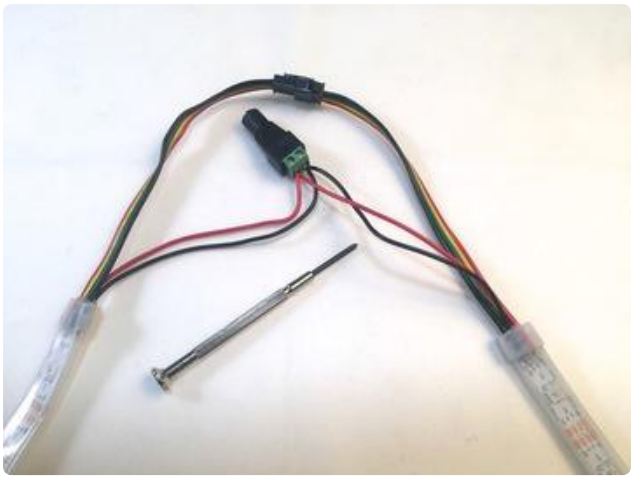
Putting it in the middle will minimize the number of pixels the power needs to flow through. Too many pixels can cause a brown-out down the line. If this is a problem with your installation, run an extra set of + and – wires from the power supply to points further down the strip, using heavy gauge wire (18 AWG or thicker). You can inject power into the strips wherever you need it.

| | | |
|---|---|---|
| LED Strip "out" connector | LED Strip "in" connector | |
| LED Strip Red Wire | Screw Terminal + | |
| LED Strip Black Wire | Screw Terminal - | |

# Assembly



Plug your LED strips into each other using the included connectors.



Decide which of your strip connections will host your power supply -- use the one closest to the middle of your entire run of LEDs.

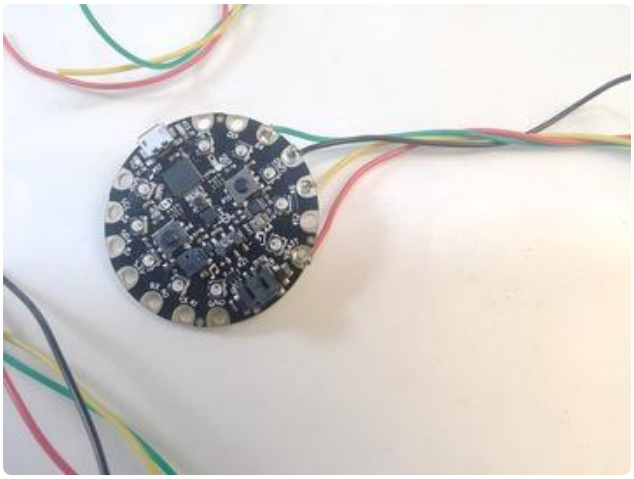
Use a small screwdriver to connect the red wires to + and the black wires to - on your screw terminal.

Note: if your screw terminal doesn't want to grab the wires, try again with a different screw terminal.  These parts seem to be pretty delicate and I went through a few of them before I found one that grabbed adequately.

## Connect the Circuit Playground

Decide where you'd like your Circuit Playground controller to be mounted and measure the distance from that point to the placement of your first LED strip.  Cut four wires to this measurement (plus a couple inches for slack) to connect to the Circuit Playground.

It's easiest to match the colors on the LED strip connectors, so we're using black, green, yellow, and red.

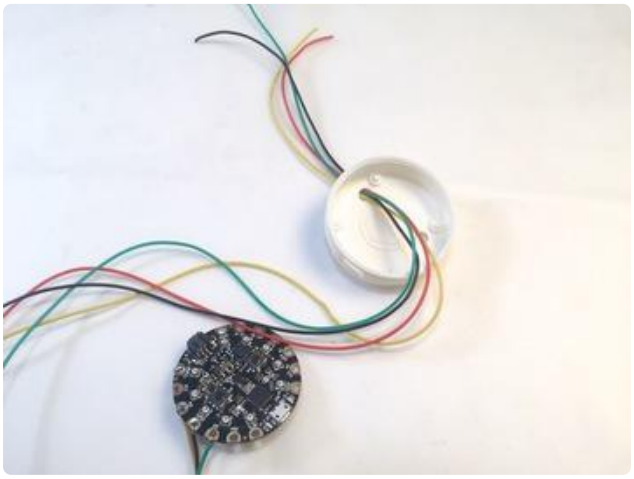


Solder the four wires to the Circuit Playground:

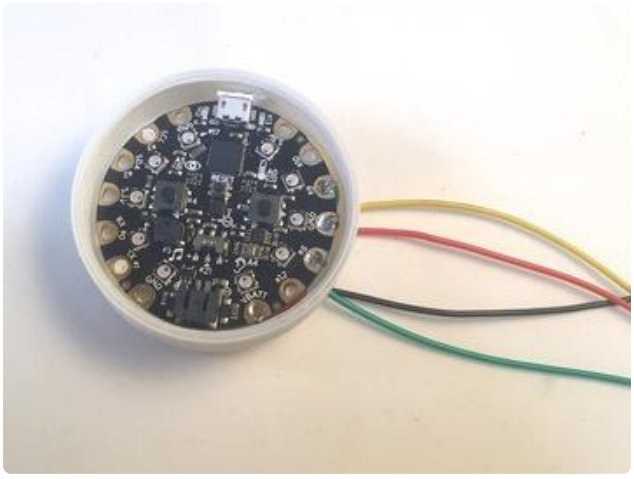
Black to G
Red to VBATT
Green to 9
Yellow to 6

Pro Tip:

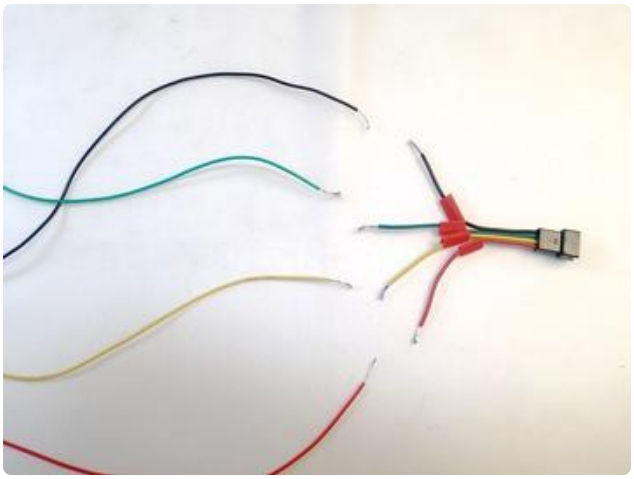
Bring the wires in from the back of the Circuit Playground rather than from the front. This will make the board fit snugly in the case without having the wires in the way.



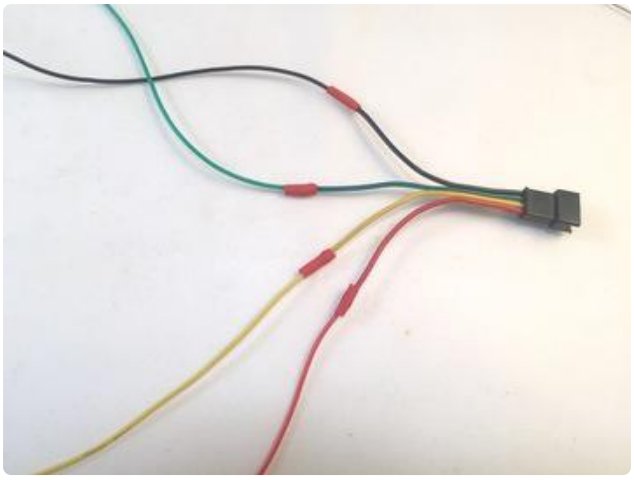
Thread the other end of the four wires through the hole in the bottom of the 3d printed Circuit Playground case.



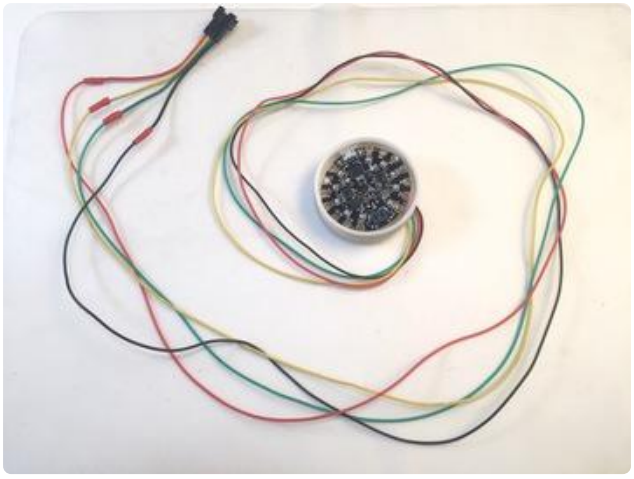
Settle the Circuit Playground in the case and press down until it clicks.



Find the very last LED in your series, and cut the male connector off the end of the strip.



Slip some heat shrink tubing over the wires and solder the other end of your four color-coded wires to this connector.



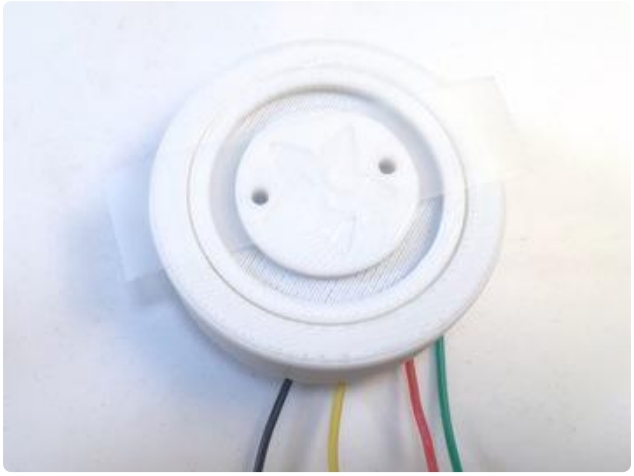
You're all wired up!  Time for testing.  Plug in your connector to the very first LED in your series, plug in your power strip and be sure all the lights work.  Press the buttons on the Circuit Playground to cycle the LEDs on/off and through their different modes.

## Troubleshooting

If your lights don't come on and act as expected, here are a few things to try:

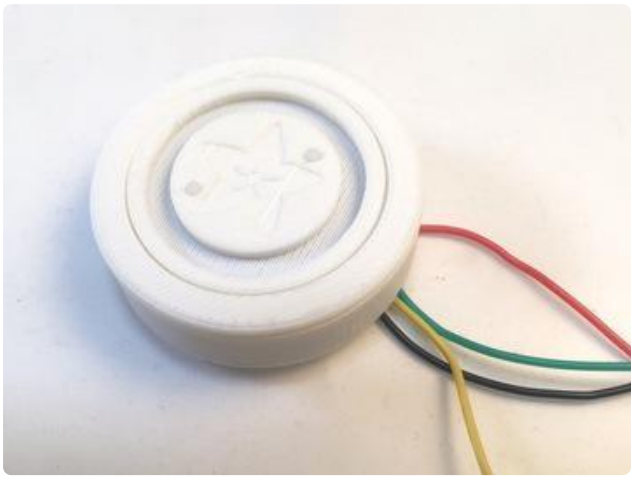
1. Make sure the wires going into your screw terminal are still tightly attached. These like to pull themselves out!
2. Check to be sure you've connected the Circuit Playground to the "IN" end of the LED strips. If you're connected to the "out" end you'll need to cut off your connectors and switch everything around.
3. Be sure your power supply is adequate for the number of LEDs you're driving. If you're getting a brown-out in part of the strip (the LEDs look dim and the colors are off), you can "inject" power into any of the exposed power wires between the strips by adding more power wires into your screw terminal and connecting the power in several places.

## 3d Case Assembly



Set the cover inside the lid and use some scotch tape to hold it in place temporarily.

Screw the lid onto the base and then adjust the cover until the holes line up perfectly with the Circuit Playground's buttons.

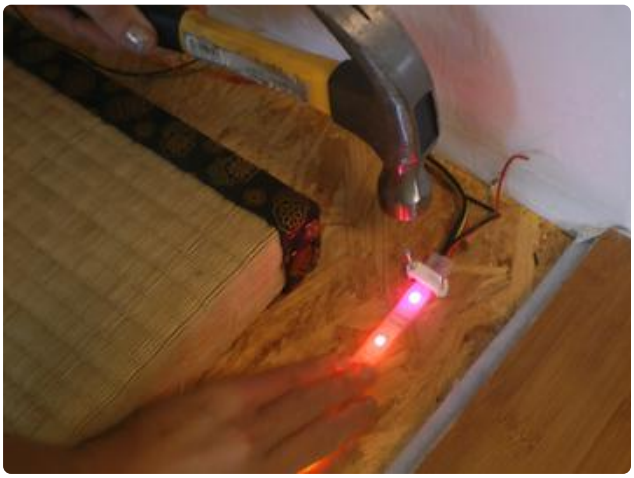


Remove the lid and add some hot glue to secure the lid and cover together.  Place the buttons in the holes and carefully screw the lid onto the base.

This is a little trickier than it looks, so take your time!  It's easiest to screw the lid on when the case is upside down so the buttons stay in place.

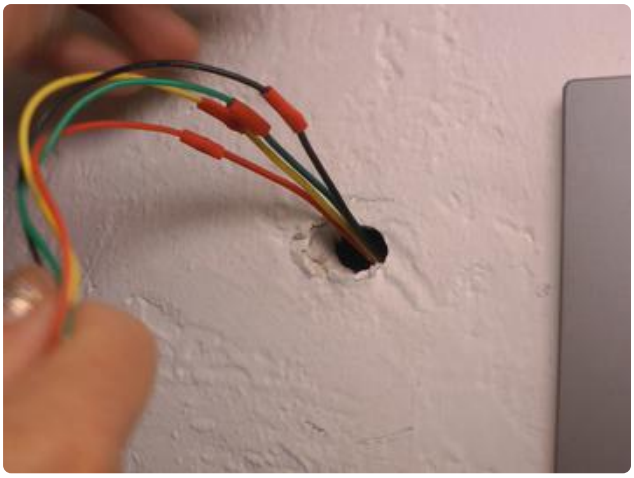
# Installation

Each installation will be different due to the nature of space and materials.  Here is what worked for me.

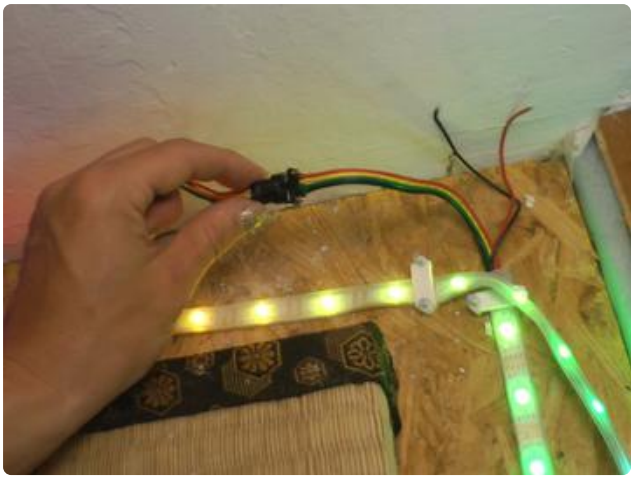



Secure the LED strip down in the center of your channel using 3/4" cable staples.  Give the corners a fairly rounded arc to avoid breaking the LED strip.

Secure the power cord down in the same manner.



Drill two adjacent 1/4" holes in the wall behind where you plan to mount the Circuit Playground, and two more where the connector will emerge to connect with your first LED.  I found it easiest to mount the Circuit Playground directly above the first LED in the series, so gravity and a repurposed coat hanger were the only tools I needed for threading the wires through the wall.



Fasten the Circuit Playground to the wall with mounting tape.  Plug the connector into your LED strip.

Fill your trough with glass pebbles and enjoy the light show!