



Remote Controlled LED Tea Light Sconce

Created by Erin St Blaine



Last updated on 2016-06-09 04:27:12 PM EDT

Guide Contents

| | |
|---------------------------|----|
| Guide Contents | 2 |
| Introduction | 3 |
| Materials Needed | 3 |
| Also Used in this Project | 3 |
| Planning & Testing | 4 |
| Choosing your Lamp | 4 |
| Testing Your LEDs | 4 |
| Power and Measurements | 4 |
| 3d Printing | 5 |
| Wiring & Teensy Setup | 7 |
| Wiring | 7 |
| Teensy Setup | 7 |
| 1. Arduino IDE | 7 |
| 2. Teensyduino Installer | 8 |
| 3. FastLED Library | 8 |
| Code | 9 |
| Assembly | 11 |
| IR Sensor Wiring | 11 |
| Barrel Jack & Power Wires | 13 |
| Dotstar Wiring | 14 |
| Finishing | 18 |
| Use It | 20 |

Introduction

This guide will show you how to transform a simple candelabra wall sconce into a high tech LED lamp. We've kept it simple with just 7 Dotstar LEDs, but this project can easily scale up to make larger lamps or even big LED art installations.

We've included lots of different lighting modes that can be controlled with an infrared remote, and the lamp will remember the last mode you selected even when it's unplugged.

Get creative and add your own design and patterns. Turn your living room, kid's room, or festival tent into a beautiful haven of light.

Materials Needed

- [Teensy 3.2 \(http://adafru.it/2756\)](http://adafru.it/2756)
- [IR Receiver \(http://adafru.it/157\)](http://adafru.it/157)
- [Remote Control \(http://adafru.it/389\)](http://adafru.it/389)
- [Dotstar LED strip \(I used 30/m\) \(http://adafru.it/n4C\)](http://adafru.it/n4C)
- [26awg silicone stranded wire - 4 colors \(http://adafru.it/1881\)](http://adafru.it/1881)
- [5v Power Supply \(I used 2A, but larger projects may want the 10A supply\) \(http://adafru.it/276\)](http://adafru.it/276)
- [Female DC Barrel Jack \(http://adafru.it/373\)](http://adafru.it/373)

Also Used in this Project

- [Tea Light wall sconce \(http://adafru.it/mSE\)](http://adafru.it/mSE)
- [Glow in the Dark PLA 3d printer filament \(http://adafru.it/mSF\)](http://adafru.it/mSF)
- Lots and lots of hot glue
- Fabric backing

Planning & Testing

Choosing your Lamp

Here are a few things to consider when choosing the base lamp to build on:

1. A candelabra with just one or two tea lights won't show off the color modes as well as something with 5 or more lights.
2. Conversely, adding more than 7-10 individual tea lights may become problematic. Longer wire runs add a lot of resistance to your project, which means lights further down the line may flicker and misbehave.
3. Consider whether your choice of sconce will easily hide your wiring. You'll have 8 wires going to / coming from each light, so think about how you plan to hide those.

Testing Your LEDs

If this is your first time using Dotstars, [visit the DotStar tutorial for details on how to install the library, wire the LEDs, power them and test them!](http://adafru.it/kDg) (<http://adafru.it/kDg>)

Before you dive into cutting and soldering and gluing, hook your Dotstar strand up to a microcontroller running the Dotstar Strandtest code, found in the **Adafruit_Dotstar** library under Examples, and make sure all the lights come on and show every color. See the tutorial for details.

I keep a [Gemma](http://adafru.it/duB) (<http://adafru.it/duB>) loaded up with Strandtest code and alligator clips on in a convenient drawer under my work bench. This is a great way to quickly make sure my strand is working without a bunch of soldering and un-soldering.

Don't skip this! You'll save yourself a lot of heartache down the line if you discover any problems now.

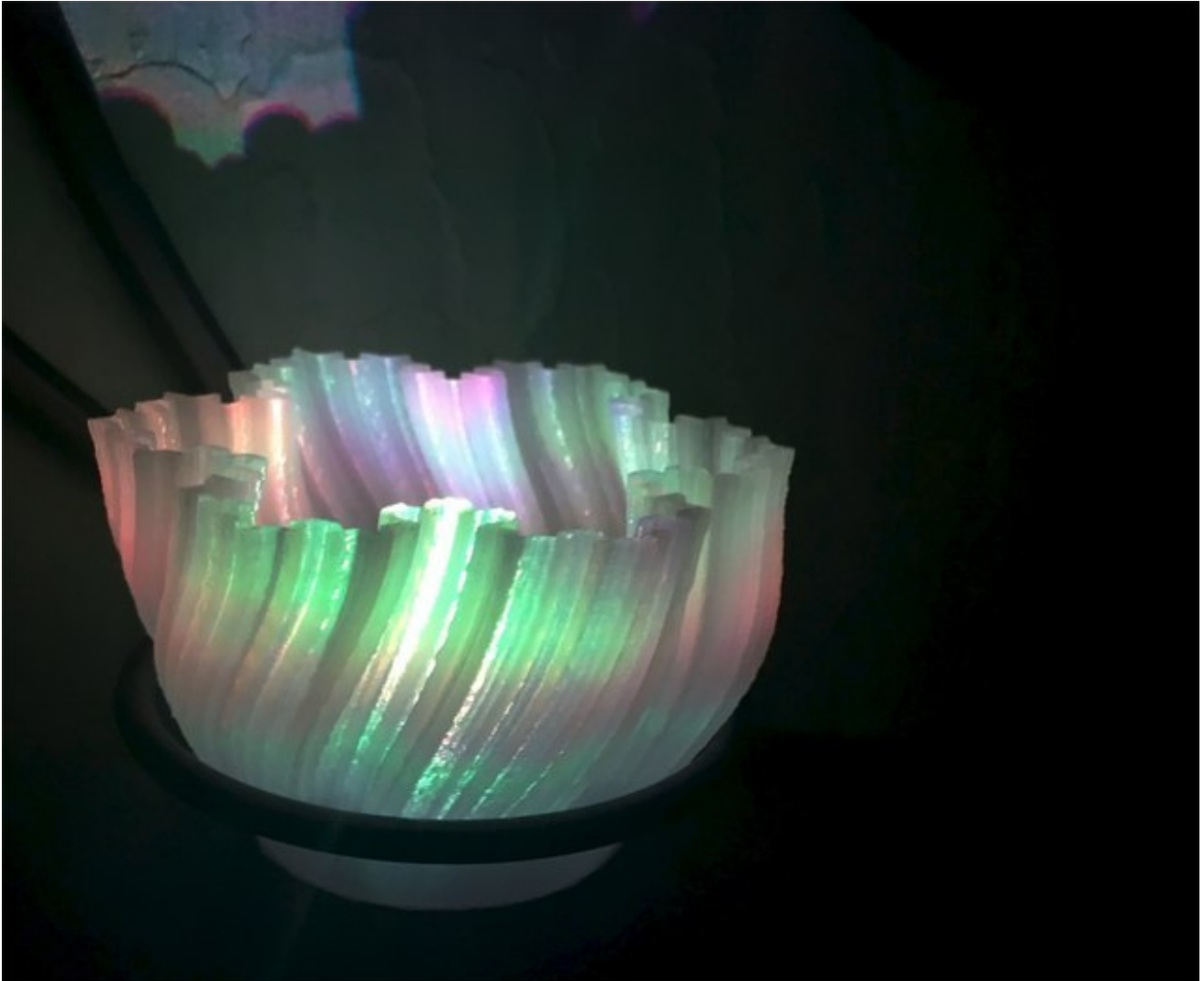
Power and Measurements

For this guide, where we've used just 7 LEDs with a few feet of wire between, our [5V/2A power supply](http://adafru.it/276) (<http://adafru.it/276>) works great. If you have a lot more LEDs, or if your wire runs are any longer than a couple feet, you may need a [beefier power supply](http://adafru.it/658) (<http://adafru.it/658>).

Consider this carefully -- your sconce may be small but after you weave the wire back and forth in order to hide it, that resistance really starts to add up when the current is over an Ampere

Also, consider where you intend to place the sconce when it's finished. Is there a power outlet nearby? Now is a great time to figure out if you'll need to add extra wire to the project so you can reach the outlet without a bunch of pesky extension cords.

3d Printing



My wall scone originally came with clear glass tea light cups. I wanted something a little more unique, and something that would diffuse the light better, so I replaced them with 3d printed cups.

I printed them in [glow in the dark PLA \(http://adafru.it/mSF\)](http://adafru.it/mSF). This PLA has a beautiful rainbowy sheen when you shine light through it, and of course it glows softly for a while after I turn the lights off.

Lovely. I love the scallop pattern that's thrown onto the wall by the ruffled edge.

This is a remix from [Marco Alici's Koch tealight design \(http://adafru.it/mTa\)](http://adafru.it/mTa) on Thingiverse. I made it a little smaller -- the 4cm height fit better into my scone.

[Download tealight.stl](#)

<http://adafru.it/mTb>

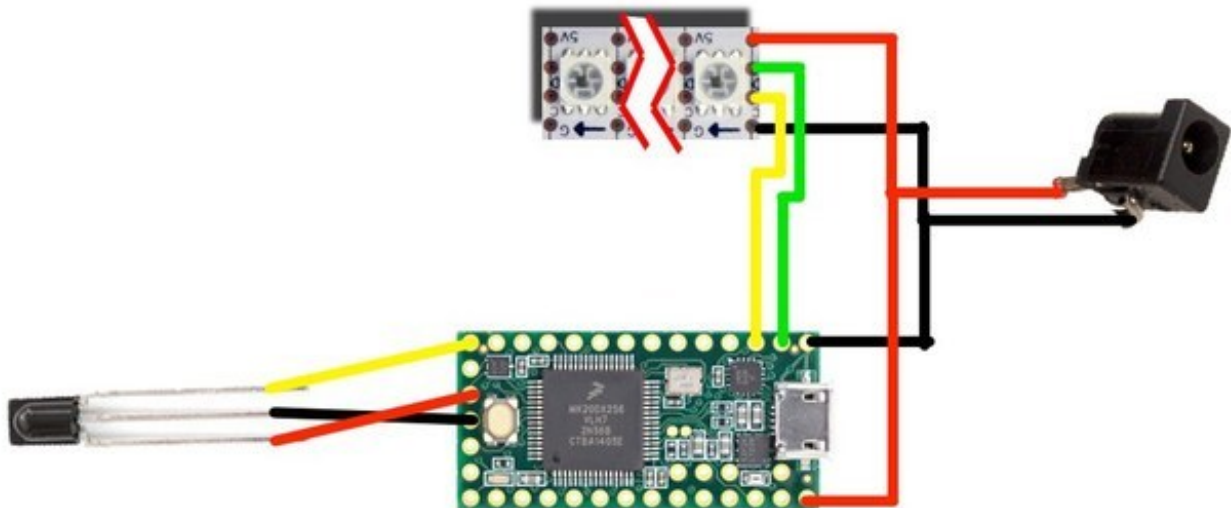
[Order on Shapeways](#)

<http://adafru.it/mTc>



Or, [find a design on Thingiverse \(http://adafru.it/mTd\)](http://adafru.it/mTd) for your own creative delight.

Wiring & Teensy Setup



Wiring

- Barrel Jack + to Teensy Vin and Dotstar 5v
- Barrel Jack - to Teensy G and Dotstar G
- Teensy 0 to Dotstar Data In
- Teensy 1 to Dotstar Clock In

With the bump on the IR sensor facing you:

- IR Pin 1 to Teensy 3v
- IR Pin 2 to Teensy G
- IR Pin 3 to Teensy 12

Teensy Setup

To get the code running on the Teensy you'll need:

1. Arduino IDE (1.6.5 or newer preferred)
2. Teensyduino Installer
3. FastLED Library

1. Arduino IDE

If you're not using a recent version of the Arduino IDE (1.6.5 or newer), this would be a [good time to upgrade](http://adafru.it/fvm) (<http://adafru.it/fvm>).

2. Teensyduino Installer

Once you have that software installed and working, download and run the [Teensyduino installer \(http://adafru.it/iwF\)](http://adafru.it/iwF), which adds support for the full line of Teensy microcontroller boards in the Arduino IDE.

In the Arduino IDE, from the **Tools** menu, select **Board**→**Teensy 3.2** and **CPU Speed**→**72 MHz (Optimized)**. Confirm that you can compile and upload the classic “blink” sketch to the Teensy board. Be sure you can blink the light before continuing.

3. FastLED Library

Use the Library Manager in the Arduino IDE to install this (Sketch→Include Library→Manage Libraries...).

Code

We are using a fabulous code base called React, by Jason Coon. This code requires installing a few custom Teensy Arduino libraries, and comes ready to go with over a dozen different amazing LED modes, many based on contributions from Mark Kriegsman and the FastLED community.

[More about Jason's creations and EvilGeniusLabs \(http://adafru.it/mTe\)](http://adafru.it/mTe)

So let's get these libraries installed!

Arduino versions 1.6.5 and above make it easy to install libraries from within the IDE, but since these are custom libraries, we need to do it the old-fashioned way -- **the library installer in Arduino will not work.**

1. Download all the .zip files
2. Unzip the files and rename the resulting folders, getting rid of the "-master" -- i.e. "react-master" becomes "react"
3. Quit Arduino
4. Move the React code base into your Sketchbook
5. Move the renamed library folders into your Arduino > Libraries folder.
6. Restart Arduino

Note: These library versions have been modified to work with the Teensy 3.2. If you have a different, or more "official" version of these libraries you will need to replace those libraries with the downloads found here.

React Code Base

<http://adafru.it/mTf>

IR Remote for Teensy

<http://adafru.it/mTA>

Audio.h for Teensy

<http://adafru.it/mTB>

SerialFlash for Teensy

<http://adafru.it/mTC>

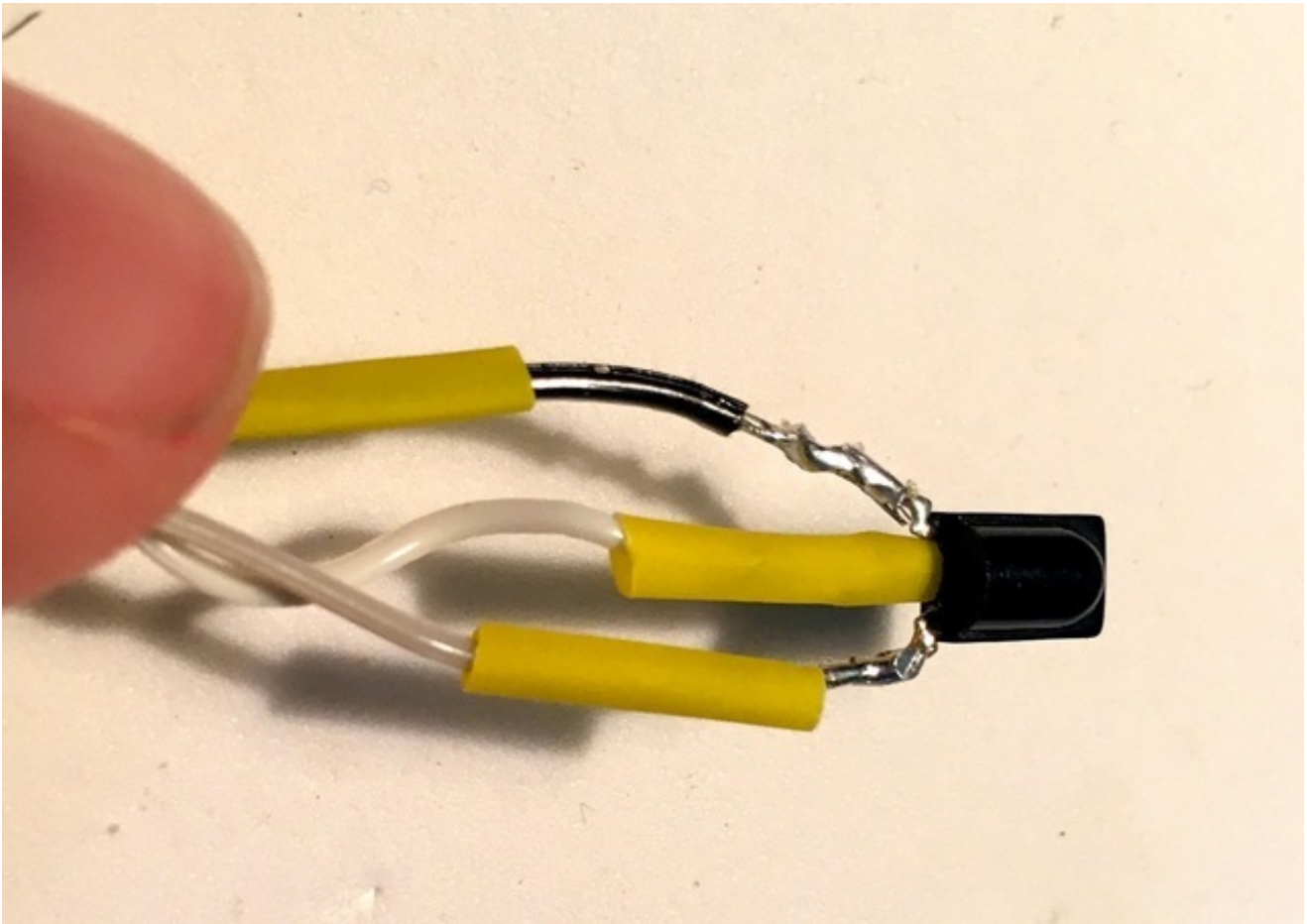
Open the react.ino file in Arduino and make sure it compiles. If it doesn't, go back and be sure all the libraries above are installed correctly.

Take a look at the code. Near the top, change LED_PIN to 0 and CLOCK_PIN to 1. Change NUM_LEDS to reflect the number of dotstars you're using in your project.

(I have just 7 LEDs but have changed this number to 8, since it seems to make the patterns run a little more smoothly for me)

```
#define LED_PIN 0
#define CLOCK_PIN 1
#define IR_RECV_PIN 12
#define COLOR_ORDER GBR
#define CHIPSET APA102
#define NUM_LEDS 8
```

Assembly



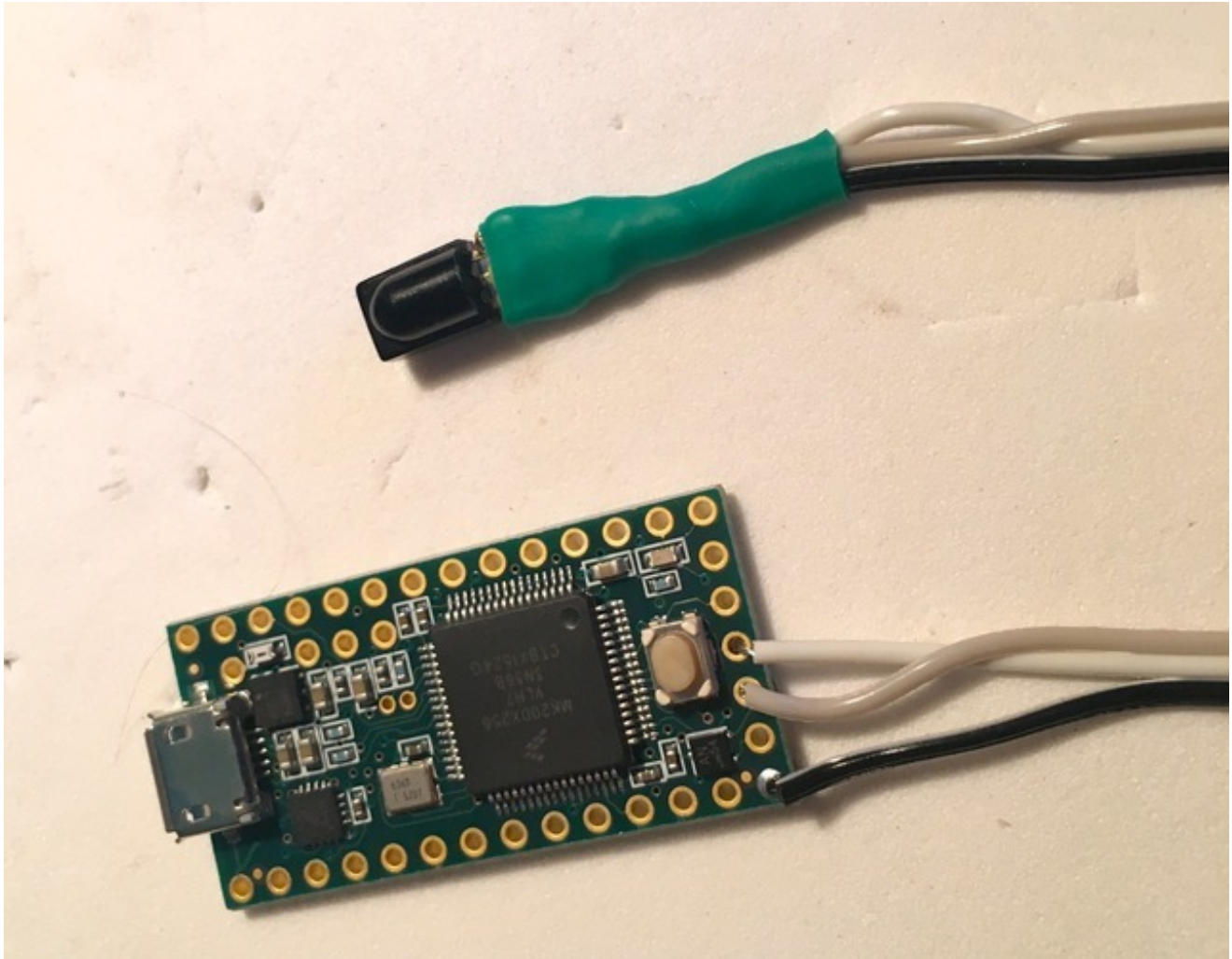
IR Sensor Wiring

As you look at the IR sensor with the bump facing you:

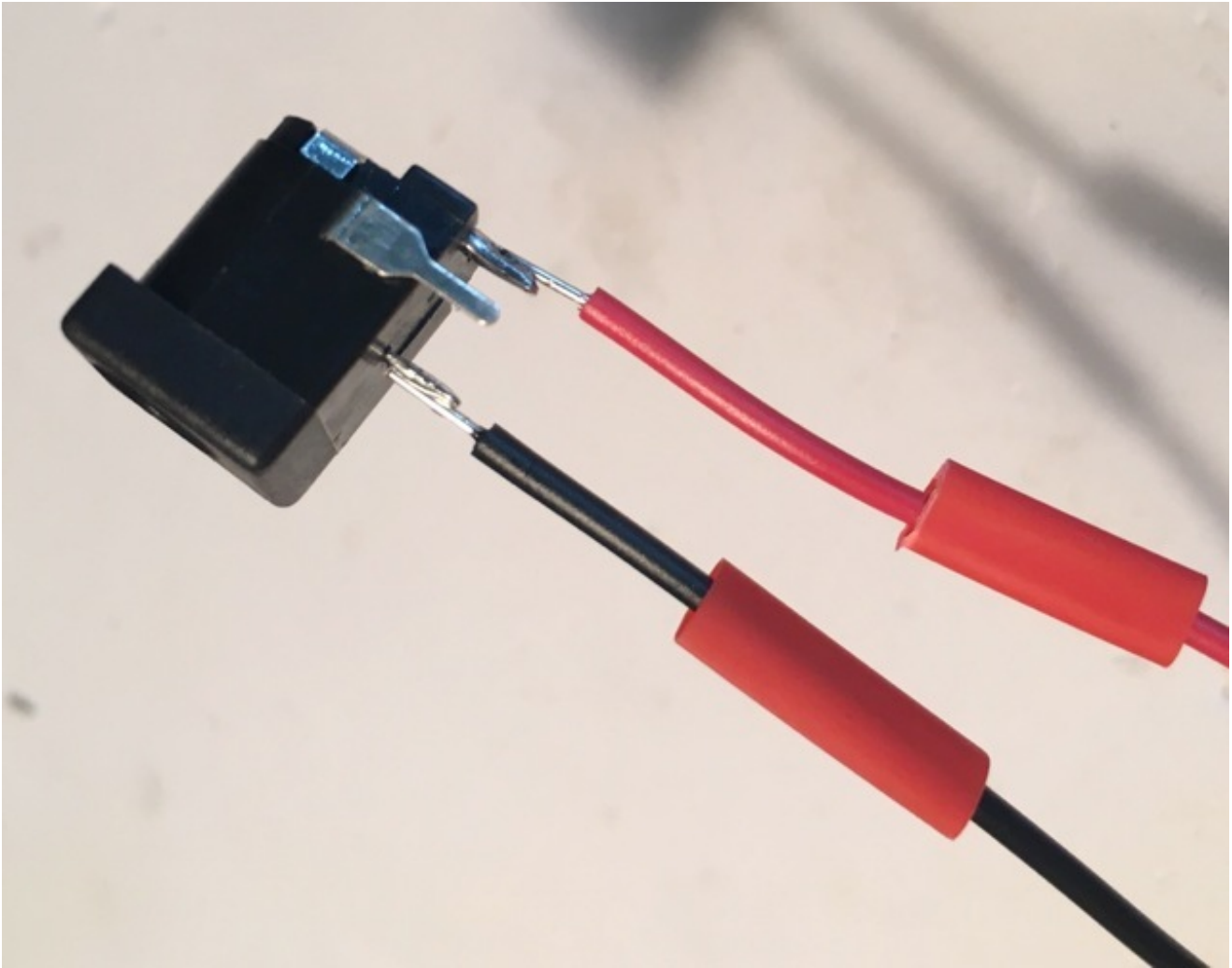
- Pin 1 (left) is Power
- Pin 2 (middle) is Ground
- Pin 3 (right) is the Signal pin, and goes to Teensy 12

Trim the leads a bit and splay out the pins gently so you have some room to work. Strip a bit of stranded wire and spiral around each leg, then solder in place and cover with heat shrink. Slip a bigger piece of heat shrink over all 3 legs together and secure.

Solder the other ends of the wire to the Teensy. (Note: I used a piece of leftover ribbon cable wire here so in the photo, my black wire is actually the Signal wire, not Ground as you may expect)



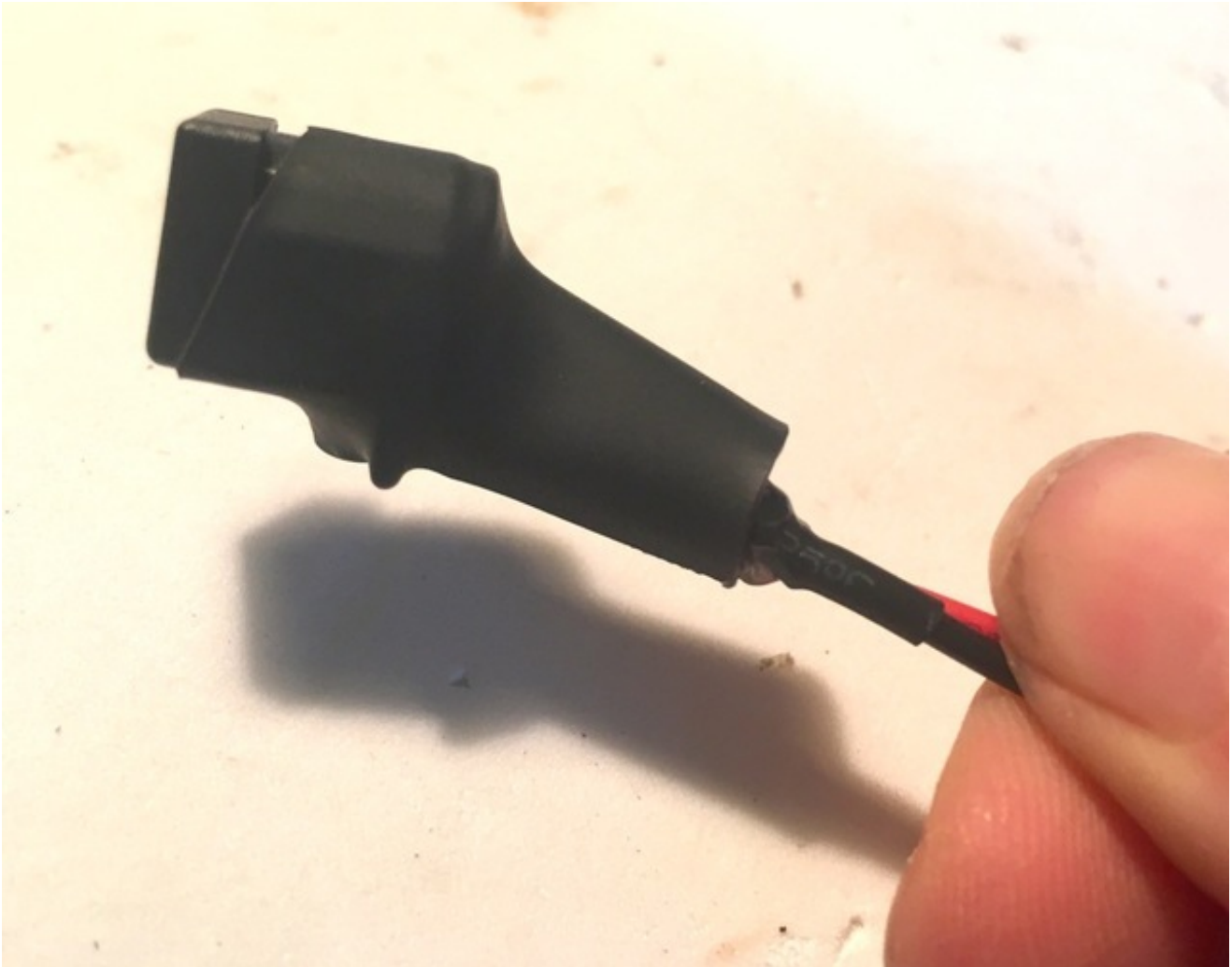
If you've uploaded the React code to the Teensy already, you can test the sensor by plugging in a USB cable and pressing some buttons on your remote. If the Teensy's onboard LED lights up when you press a button, your sensor is working correctly.



Barrel Jack & Power Wires

Solder some beefy wires (at least 26awg) onto your barrel jack as shown -- the power wire goes to the pin at the rear, and the pin on the middle of the bottom goes to ground. We are not using the pin on the side.

Secure with heat shrink, then cover the whole assembly with more large heat shrink. I filled this larger heat shrink with hot glue to be sure these wires stay tight, since this jack is pretty tricky to solder a solid connection to.



Take the two wires coming from the barrel jack split them by splicing two wires onto each. One power wire will go to the LEDs and the other will go to the Teensy.

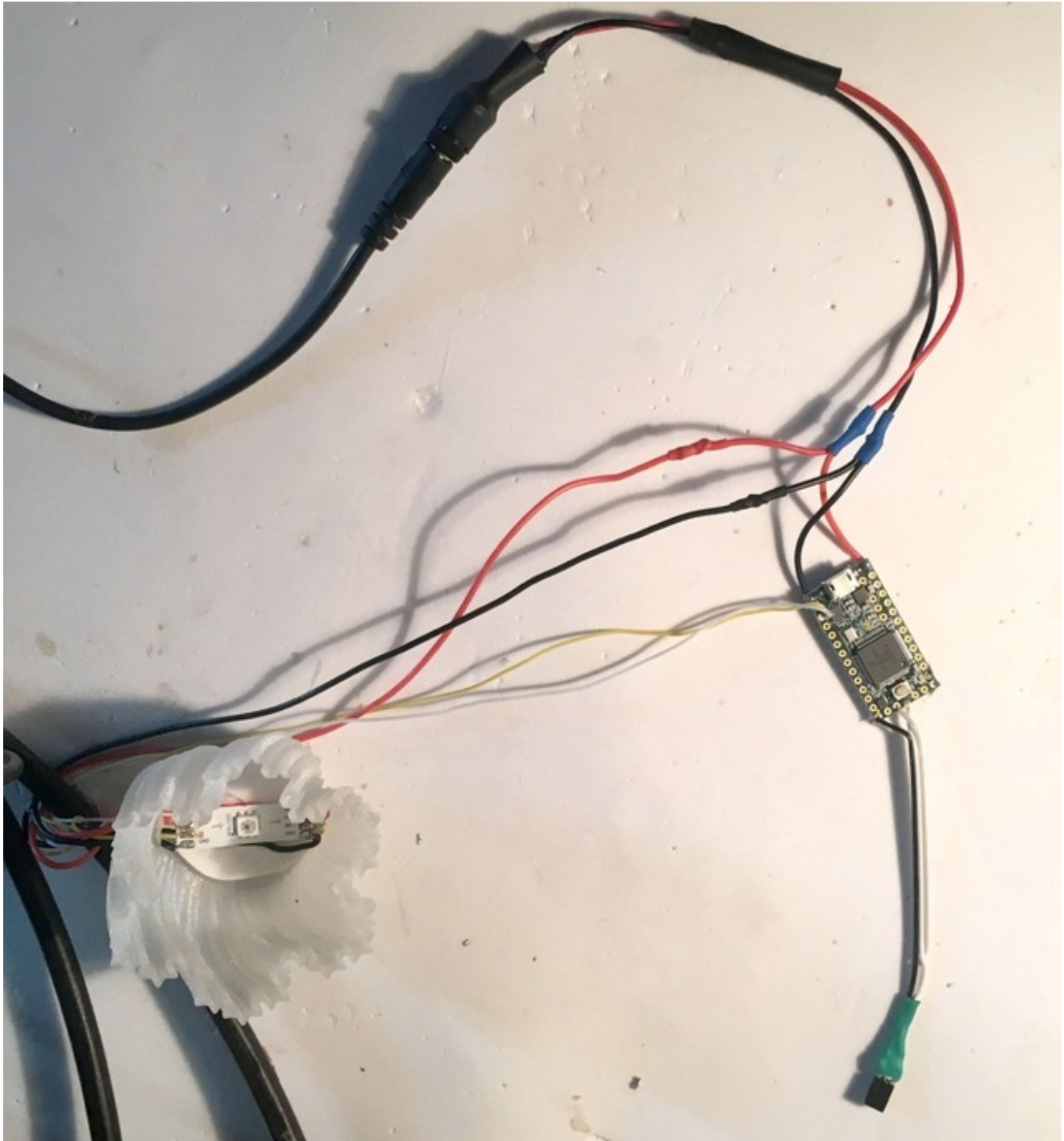
Dotstar Wiring

Pull your Dotstar strip out of its silicone sheath and carefully cut each light apart on the cut lines.

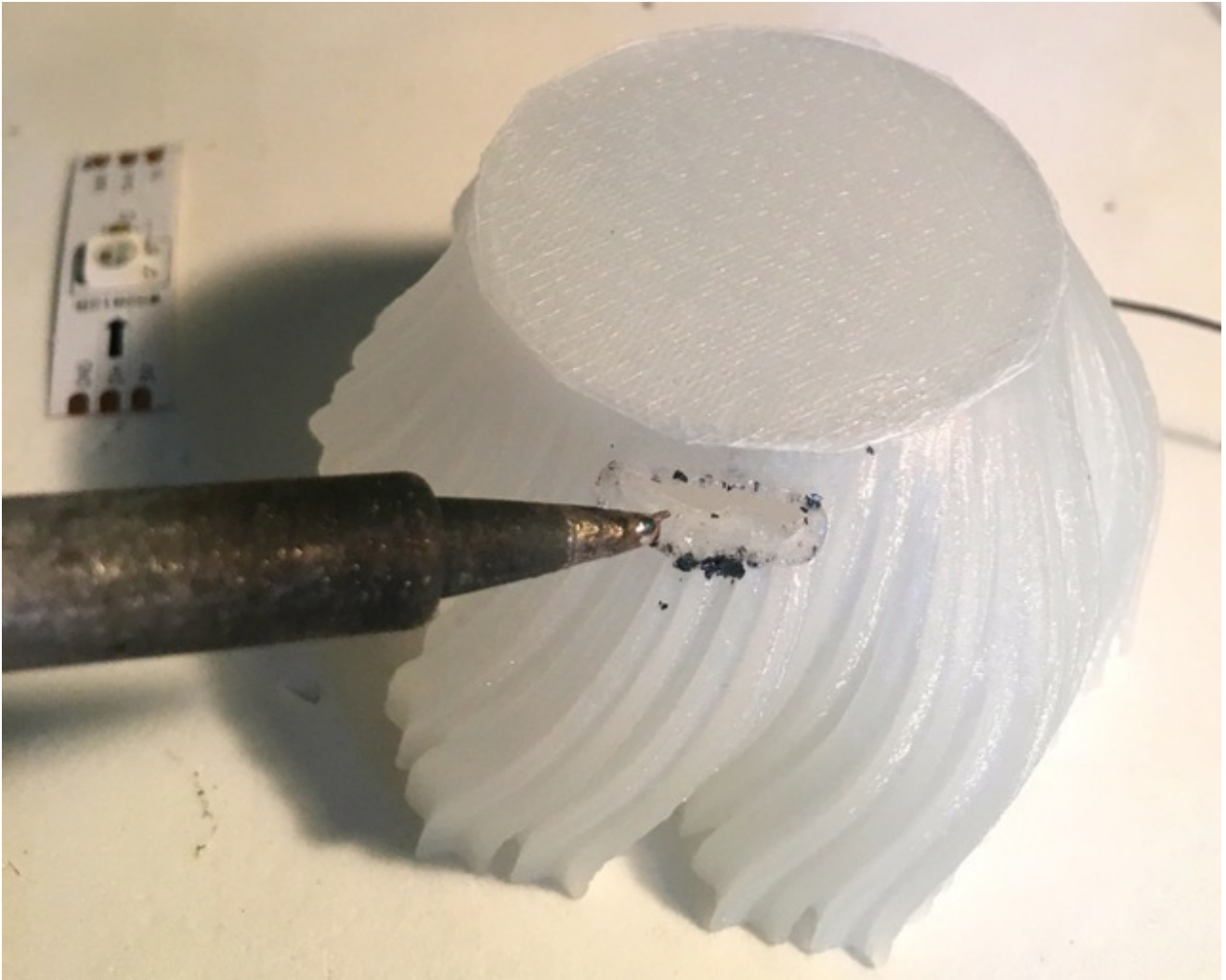
Solder a wire from pin 0 on the Teensy to Dotstar's Data pin, and a wire from pin 1 to Dotstar's Clock pin.

Then, connect two of the power wires from your barrel jack to Dotstar's + and - pads.

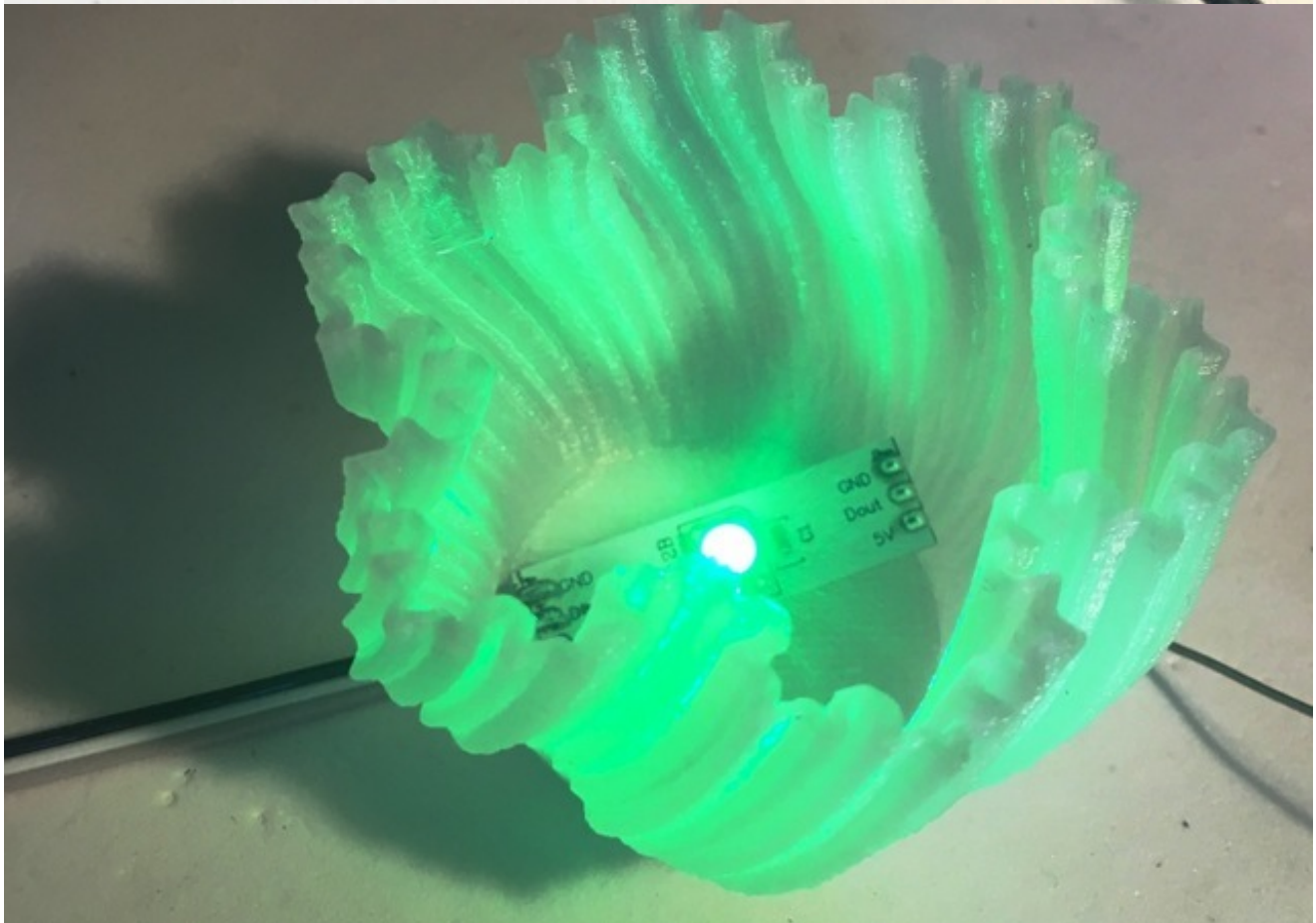
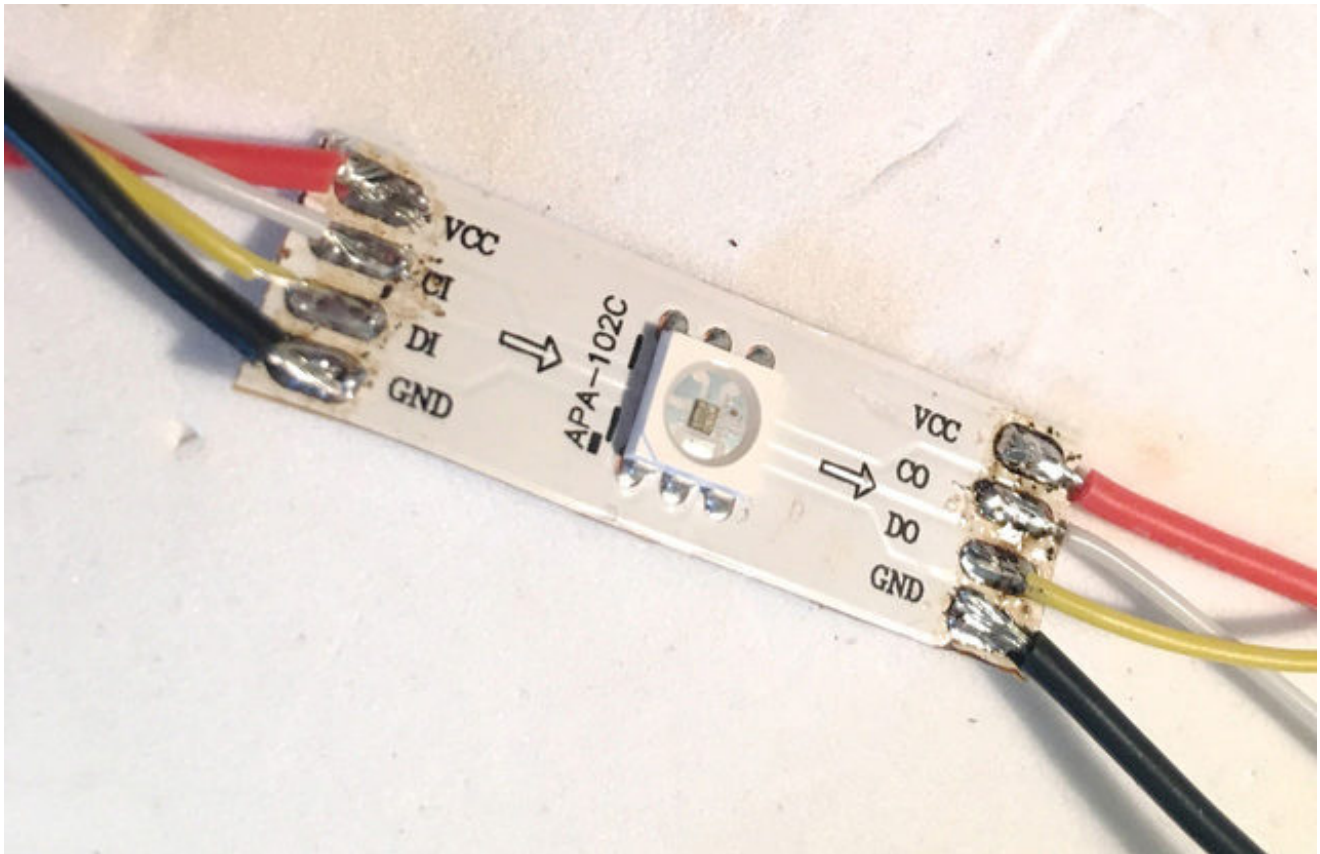
Connect the other two wires from the barrel jack to Teensy's G and Vin pins. Plug in your power supply and be sure the light comes on. You can also test out the remote control at this point and be sure things happen when you push the buttons.



Next, prepare the first of your tea light 3d printed cups. You'll need to make a hole that's the right size to slide the Dotstar and the wires in. A Dremel tool is ideal, but if you keep an old spare "dirty jobs" soldering iron around, that can work too. Don't do this with your nice iron.



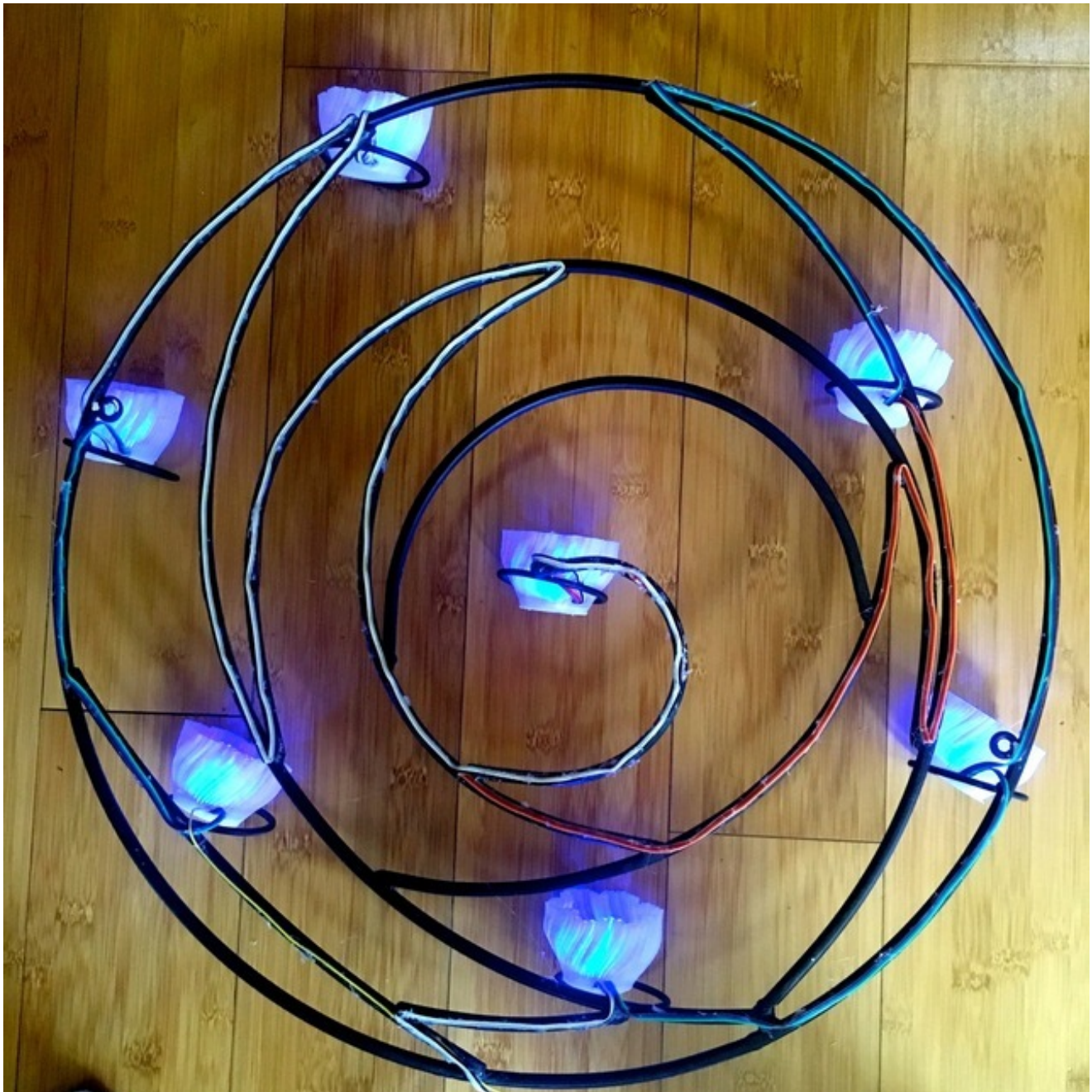
Solder 4 wires to the "out" pads of your Dotstar LED, then fold the wires over and slip the light into the cup through the hole you just made. Secure it down with some hot glue.



From here on out, the artistry is up to you. Solder as many more Dotstars in line as your finished piece demands.

Keep in mind that longer wire runs will add a lot of resistance to your circuit, so you'll need to find a balance between hiding the wires and being sure you're getting enough power to the lights. Test each new connection as you go to minimize troubleshooting later on.

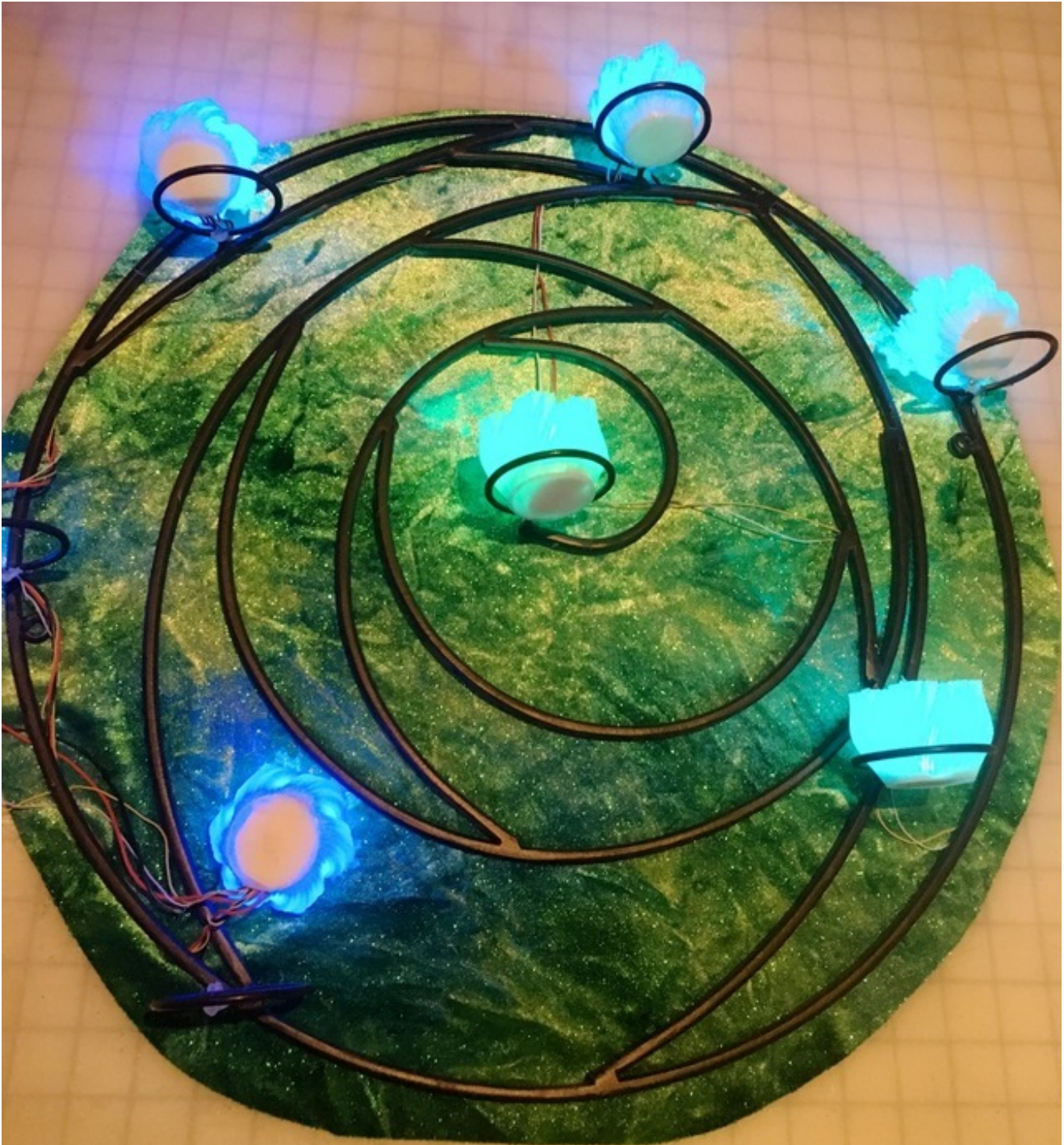
Another option is to “home-run” the power wires, but this can be more complex to visualize and build. + and – all lead back to the barrel jack (perhaps using something like Perma-Proto board to distribute power), while data and clock are chained from light to light as normal.



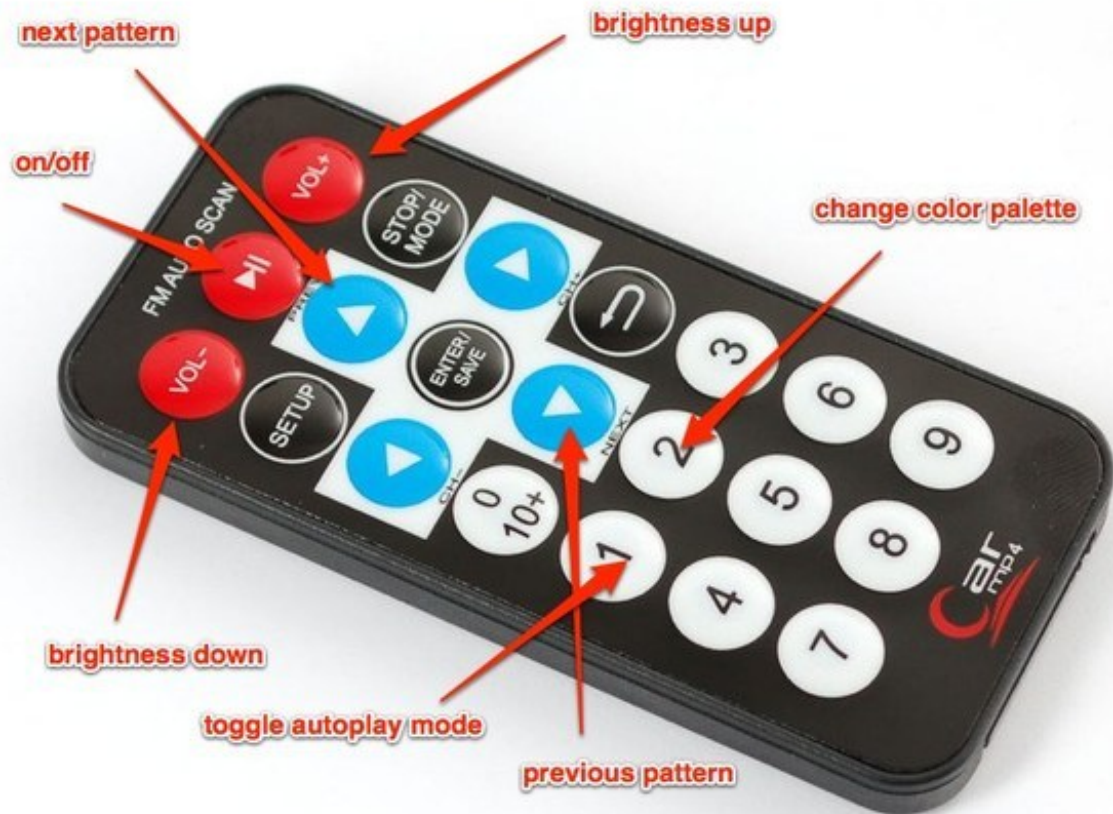
Finishing

I added a fabric backing to my wall scone to help hide some of the wiring and to provide a place to mount the Teensy -- I just glued it to the back of the fabric.

Mount the IR sensor someplace inconspicuous on the front of your lamp with the bump facing outwards. Make sure it picks up signals from your remote at a comfortable distance.



Use It



As written, the React code allows you to change patterns and brightness with the Adafruit remote control. There is more functionality available with other [commercially available remotes](http://adafru.it/mTD) (<http://adafru.it/mTD>), or it's not too hard to add your own functions for the unused buttons.

Take some time and look through the React code. You'll find a "playlist" section near the top, where you can choose the modes you like and reorder them as you see fit.

Since we haven't included audio or interactivity in this lamp design, some of the modes won't appear to do anything at all. Some may look better on your finished lamp than others. You can customize your modes simply by editing this playlist.

```
const PatternList: Pattern[] = [
  waves,
  colorWaves,
  prds,
  softTwinkles,
  colorTwinkles,
];
```

```
//ire2012 with alien  
sinelon;  
lightning2014;  
//bouncingBalls2014;  
spectrumBar;  
//spectrumDots;  
//juggle;  
bpm;  
//confetti;  
rainbow;  
//rainbowWithGlitter;  
hueCycle;  
//discoStrobe;  
//simon;  
//colorInvaders;  
showSolidColor;  
};
```