



# Morning Star POV Double Staffs

Created by Erin St Blaine



<https://learn.adafruit.com/pov-dotstar-double-staff>

Last updated on 2024-06-03 01:47:13 PM EDT

# Table of Contents

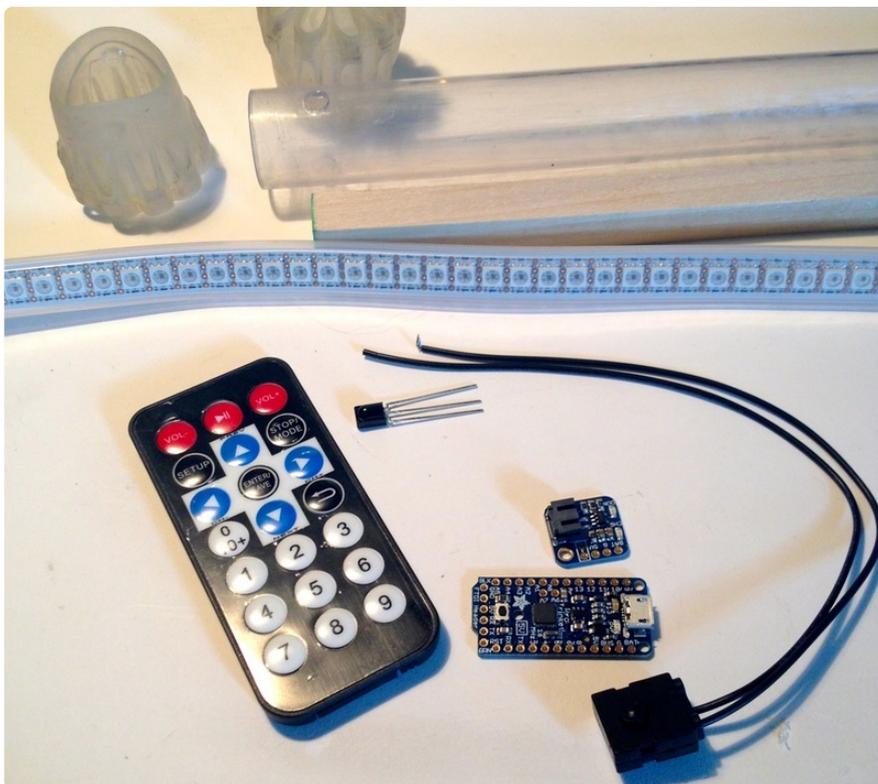
<a href="#">Introduction</a>	<a href="#">3</a>
<a href="#">Code</a>	<a href="#">5</a>
<a href="#">Wiring</a>	<a href="#">6</a>
<a href="#">Layout &amp; Sizing</a>	<a href="#">7</a>
<a href="#">LED Wiring</a>	<a href="#">9</a>
<a href="#">Pro Trinket &amp; Charger Wiring</a>	<a href="#">16</a>
<a href="#">Using the Remote</a>	<a href="#">23</a>
<a href="#">Preparing Images</a>	<a href="#">25</a>

---

# Introduction



Create your own LED persistence-of-vision double staffs to spin and twirl and light up the night. Based on the [Adafruit Genesis Poi \(https://adafru.it/IgE\)](https://adafru.it/IgE) design and code, these double staffs take it to the next level with infrared remote control and more more MORE LED pixels, for bigger and brighter and sharper images, and a juicy 2200mAh battery that lets you outshine the brightest star in the sky.



For **ONE** short staff, you will need:

- Pro Trinket 5v
- LiPoly Backpack Charger
- Tactile on/off switch
- 1 meter of 144/m Dotstar LEDs
- IR Sensor
- Lithium Ion Cylindrical battery (2200mah)
- Mini Remote Control
- 26 gauge wire in various colors

Double the parts for two staffs (except for the remote control — only one is required).

You'll also need:

- 3 foot, 1" diameter polycarbonate tubes with end caps
- 1/2" square wooden dowel
- Glue, heat shrink & soldering accessories

I am using [these ready-made tubes from FlowToys.com \(https://adafru.it/JCS\)](https://adafru.it/JCS), in size 7F. They also sell [wonderful caps \(https://adafru.it/fUI\)](https://adafru.it/fUI) (requires 2 per staff) that snap in place delightfully well even when you spin really fast. Plastics stores will also sell tubing, but be sure to get **polycarbonate** — acrylic tubing will break.

The inner diameter of your tubes should be **at least 7/8"** to ensure that everything fits. And remember to get **two** of everything (except the remote) if you want two staffs to spin!



---

# Code

It's a great idea to get your software all set up and loaded onto your Pro Trinket right away, to make testing your connections easier later on.

If this is your first time using Pro Trinket, follow the [Adafruit Arduino IDE Setup \(https://adafru.it/jDQ\)](https://adafru.it/jDQ) for guidance; a couple extra steps are required compared to typical Arduino Uno programming — the [Introducing Pro Trinket \(https://adafru.it/iPe\)](https://adafru.it/iPe) guide may help.

If you're not already running the [Arduino IDE version 1.6.4 or later \(https://adafru.it/f1P\)](https://adafru.it/f1P), this is a really good time to upgrade. It greatly simplifies installing libraries and support for alternate boards such as Trinket.

Software for the double staff project can be fetched from [GitHub \(https://adafru.it/E-O\)](https://adafru.it/E-O):

**Click to download Kinetic POV  
archive for Arduino**

<https://adafru.it/E-O>

The “dblstaff” folder contains the Arduino sketch for this project. The “convert” folder contains a utility for processing images — we'll cover that on a later page. The other folders can be ignored — they're for other projects that evolved from the same code base.

This project also requires the Adafruit DotStar library for Arduino. Use the Library Manager to install this (Sketch→Include Library→Manage Libraries...), or if you're using an older version of the Arduino IDE, it can be downloaded and [installed manually \(https://adafru.it/dNR\)](https://adafru.it/dNR):

**Click to download Adafruit\_DotStar  
library for Arduino**

<https://adafru.it/eio>

There are two files in the “dblstaff” folder, which will open as two tabs in the Arduino sketch. The second file/tab — graphics.h — contains the bitmaps and color palettes for the different modes. We'll explain how to add different ones later on.

---

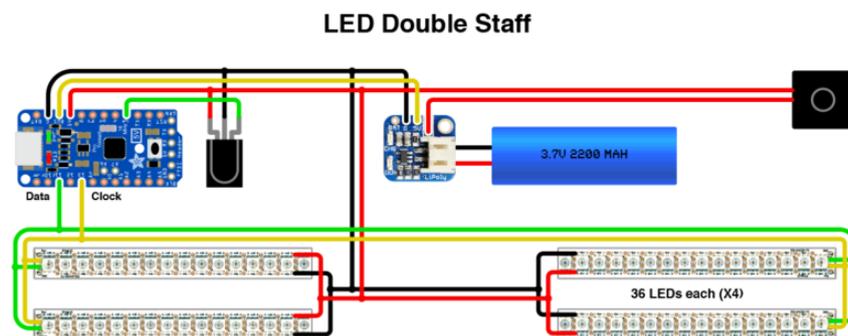
I can compile the code but it won't upload to the Pro Trinket board!

You might have a "charge only" USB cable. Definitely need the normal "charge+data" type for this. Switch it out for a different cable and try again.

Also, be sure you've selected "Pro Trinket 5V/16MHZ" from the tools dropdown.

---

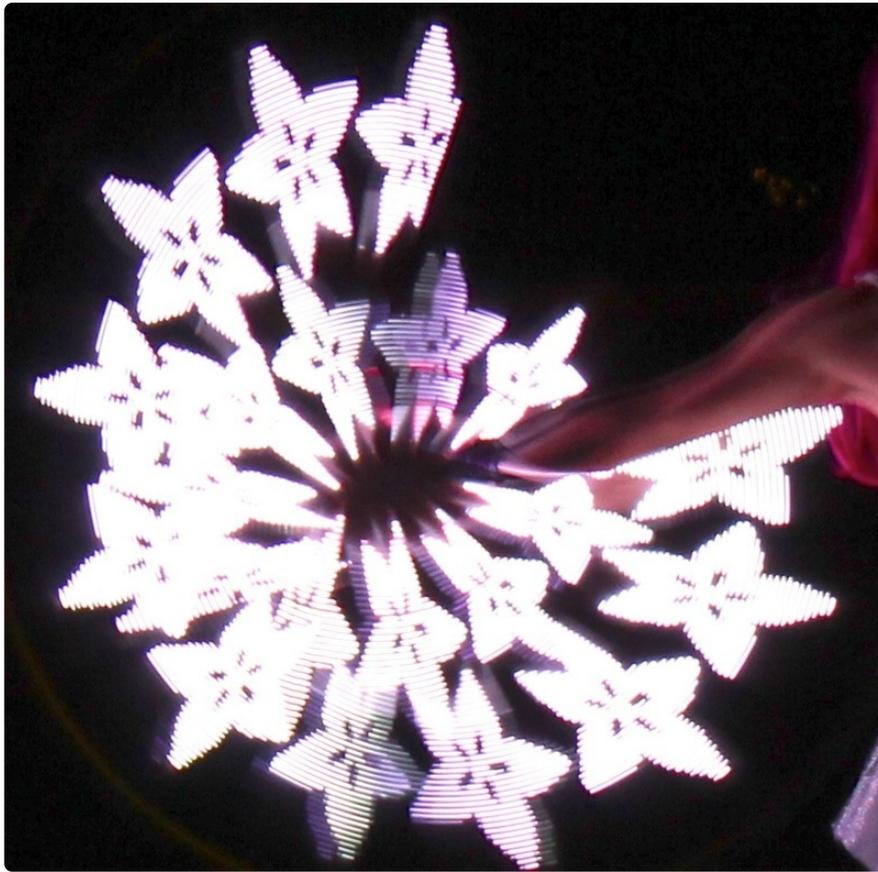
## Wiring



This is a schematic layout of the parts, to clearly show all the connections, and not representative of the actual placement of all the parts or wires. We'll detail that in the following pages.

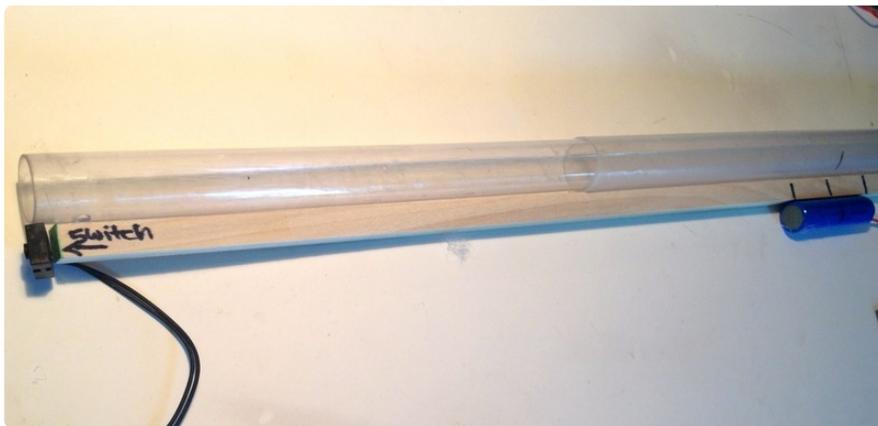
The Pro Trinket, charger, and IR sensor will be at one end of the staff, and the switch will be at the other end.

You'll want to place the battery in the middle, since it's the heaviest part. Placing it at one end would create an unbalanced spin to your staff, so keeping it in the center works best.



---

## Layout & Sizing



Laying out all the pieces ahead of time will make getting wiring lengths right much easier in the long run.

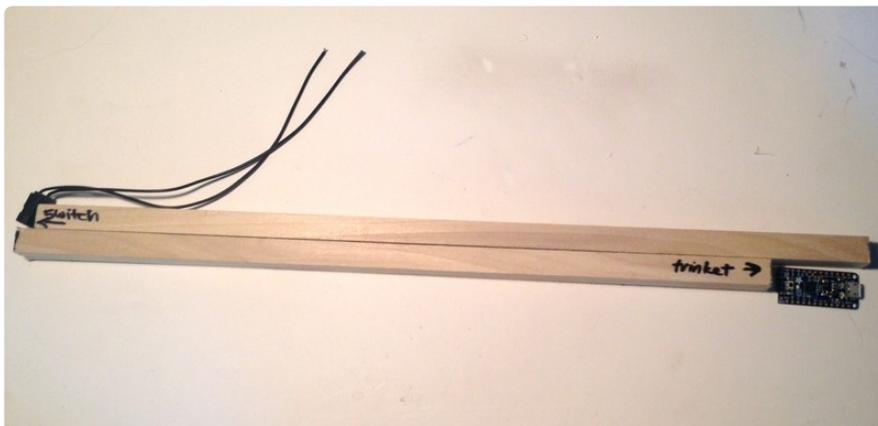
The on/off switch is located at one end of the staff, the battery is in the middle (for balance) and the Pro Trinket and battery charger are at the opposite end from the switch.

If you're using FlowToys caps, remove the insert with a pair of pliers.



Mark the center on your polycarbonate tube. Align the center of the battery with your mark.

Measure and cut two lengths of wooden dowel to fill most of the tube on either side of the battery. Leave enough room on one end for the switch, and on the other end for Pro Trinket. It's helpful to add labels so you remember which dowel goes on which side.



Slide everything into your polycarbonate tube and make sure it fits: Pro Trinket -- shorter dowel -- battery -- longer dowel -- power switch. You want the power switch flush with one end and the Pro Trinket USB port flush with the other end.

Take a minute and be sure this is right and everything fits perfectly.



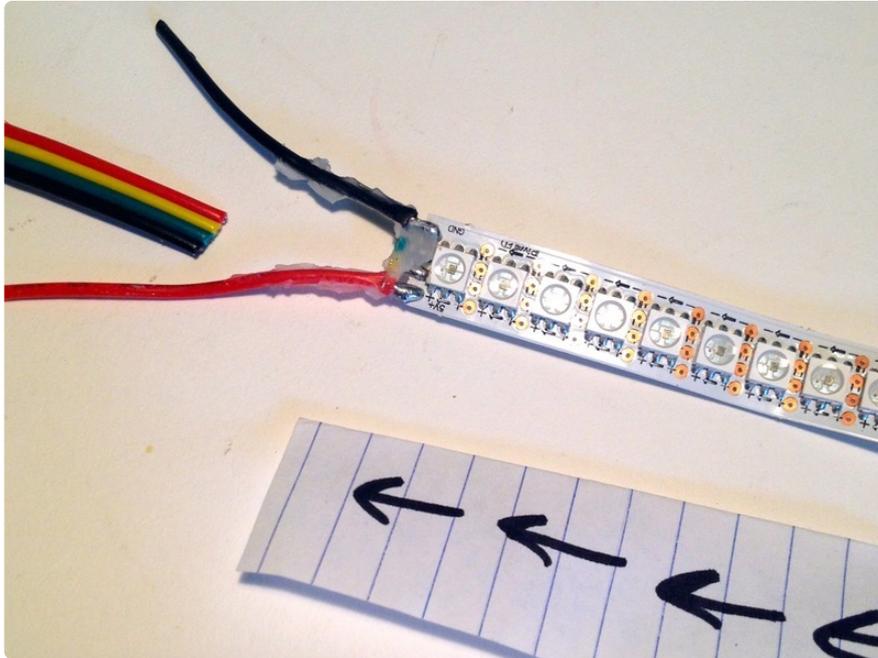
---

## LED Wiring

I'm using [26awg](http://adafru.it/1880006328647588) (power & ground wires) or [30 awg silicone-coated wire](http://adafru.it/2051) (for data lines) for all steps. This wire is super flexible, heat-resistant, easy to use, and very hard to break. It makes the wiring on this project much easier using than traditional wire.

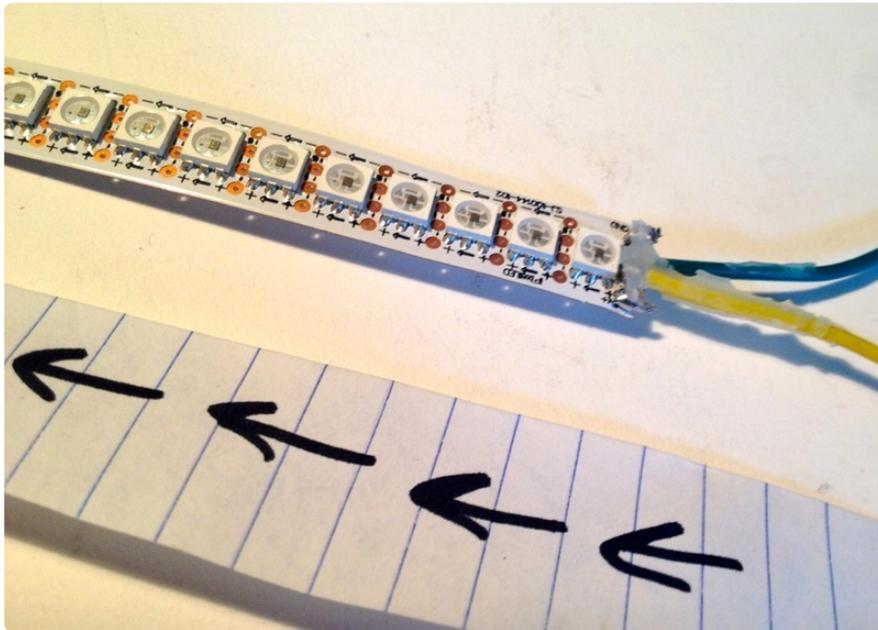
Color-coding the wire is very helpful too, as the wiring can get a little crazy otherwise. I'm going with the color-coding as I found it on my Dotstar strips. Manufacturing sometimes changes these colors around, so notice if your wiring colors are different and adjust accordingly.

For me: Power (5v) is red, G is black, Clock is yellow and Data is green.



Use a utility knife to carefully remove the silicone sheathing and most of the covering and glue from the ends of your 1M Dotstar LED strip where the connector is soldered on. Get the ends as clean and tidy as you can.

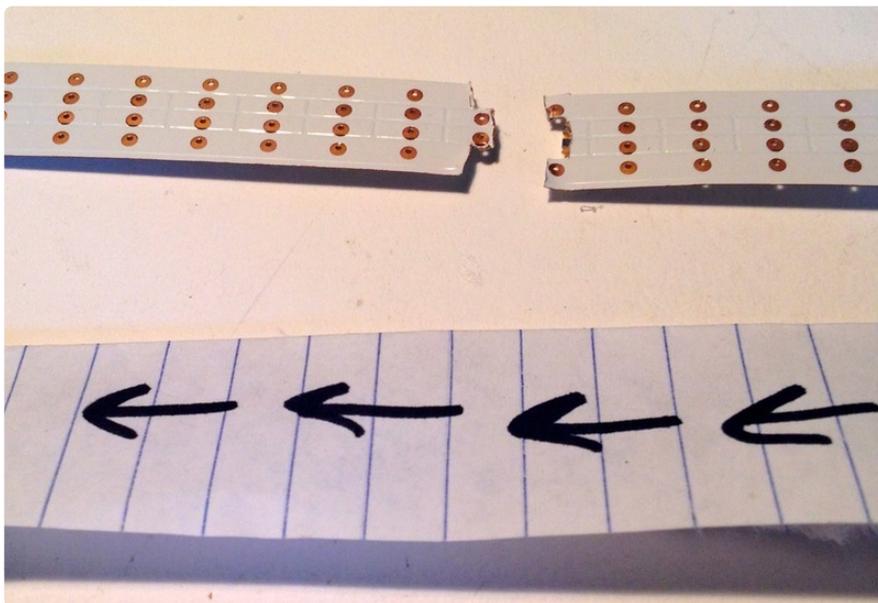
On the "in" end of the strip (arrows pointing away from that end), clip off the red and black (5v & g) wires, leaving the two center wires. On the "out" end, do the opposite -- clip off the data & clock wires (yellow and green on my strip) leaving the power wires intact.



Go back to the "in" end and count out 36 pixels. Count them once more. Then, with tiny scissors or a utility knife, VERY CAREFULLY cut between the pixels as shown. Leave the two center pads on the "in" end, and the two outer pads on the "out" end of the segment. (hint: this is easier if you do it from the back of the strip)

Repeat with the rest of the pixels until you have 4 equal strips, with clock and data pixel pads on the "input" ends and power and ground pads on the "output" ends.

You may not need this fancy cut in the middle of the strip, if your strip has a solder joint there -- just unsolder and you've got four full pads on each strip you can use.

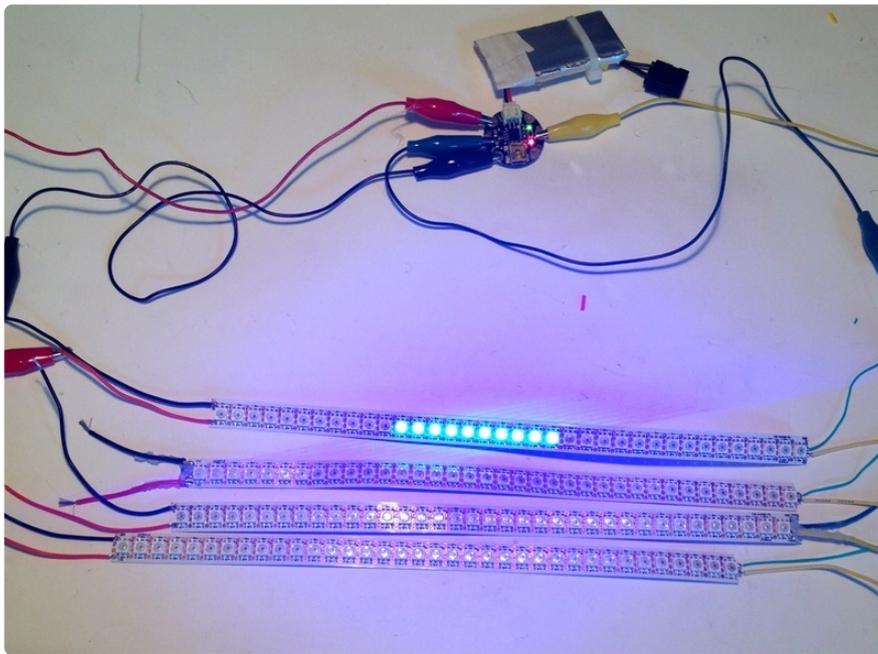


Cut 3 short (2-3 inch) pieces of yellow & green wire. (You only need 3 because one of your strips already has the wires soldered on, fancy that!) Solder the yellow wires to

the clock pin and the green wire to the data pin on each of the 3 LED strips, copying the layout from the 4th strip.

Cut 3 short (2-3 inch) pieces of red & black wire. Solder the red wires to the + and the black wires to the - pads on the other ends of the LED strips, copying the layout from the pre-soldered one again.

This is a great time to hook up some alligator clips and a gemma or flora running the [Dotstar strandtest code](https://adafru.it/kDg) to be sure all your LEDs turn on and your connections are solid. Once you're happy, cover the ends of the LED strips in hot glue to reinforce these tiny solder points.



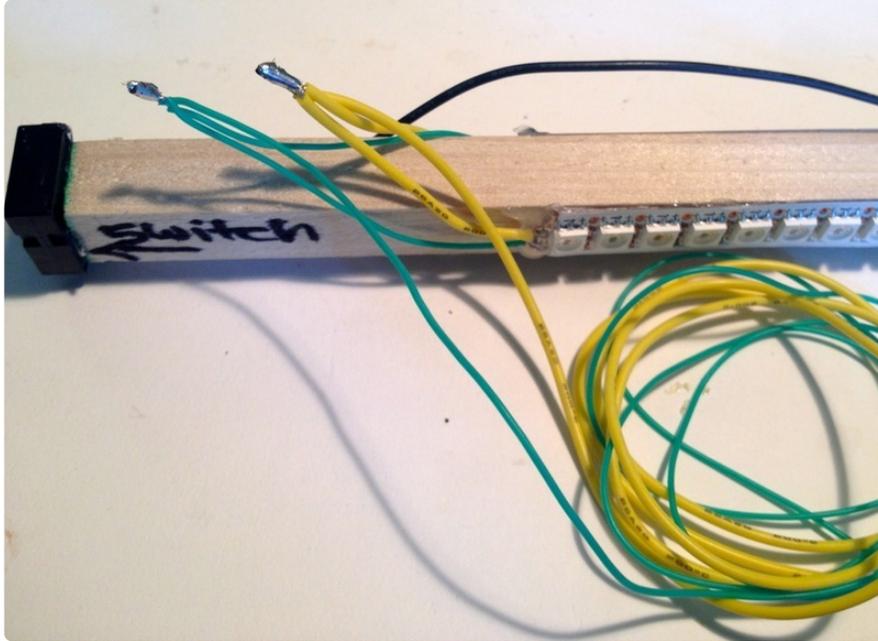
Lay the LED strips along the wooden dowels, making sure they're an equal distance from the \*battery\* end (the middle) -- remember, these two dowels are different lengths, so aligning them with the outer edges of the dowels will give you an off-center staff. Leave an inch or two of dowel at the outer edge, so your LEDs don't get blocked by converging wires later on.

Use a thin layer of glue (hot glue works great) to secure the LED strips to the dowels. Be sure not to pile it on -- you want a low profile here so the whole thing fits nicely into the tube.



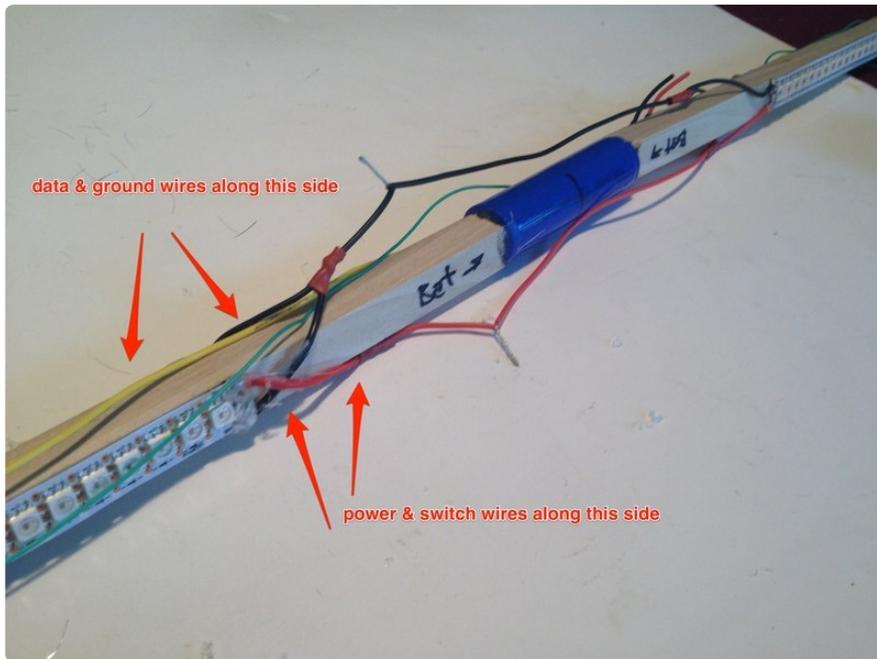
Twist the two data wires and the two clock wires together at each end of the staff.

Find the end with the switch. Splice a long (3 foot) piece of color-coordinated wire to each twisted pair. This wire needs to reach the corresponding twisted pair at the other end. Run this wire down the opposite side of the dowel from the switch wires.

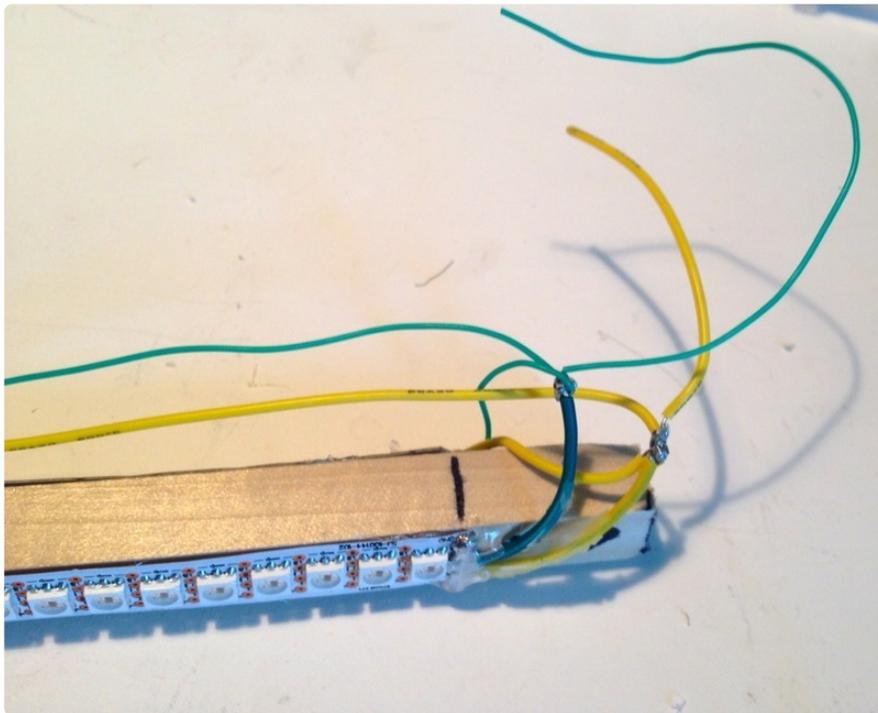


Pull the two red wires from the opposing LED strips to the same side as the switch wires and twist into pairs.

Pull the black wires to the other side, and twist them into pairs too. Splice another short wire onto each pair, long enough to meet near the middle as another twisted pair. (Mine are off-center so the joins don't compete with the battery for tube space) Leave these alone for now, we'll hook them up a little later.



At the Pro Trinket end, connect the green & yellow clock & data wires together, splicing one more short wire in to each side for the run to the Pro Trinket.

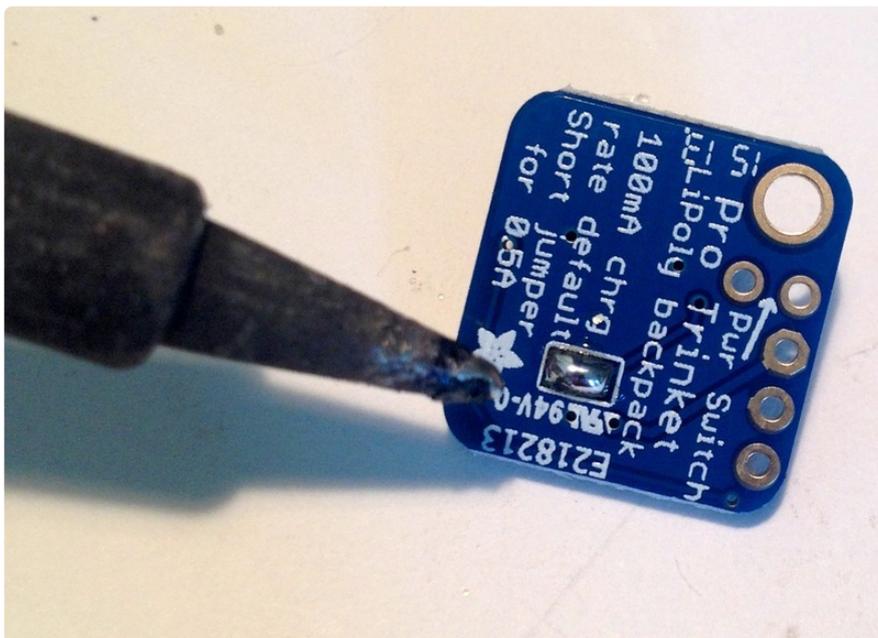


Whew! Our LED strips and switch are mostly wired up. Set this assembly aside. Maybe have some chocolate.

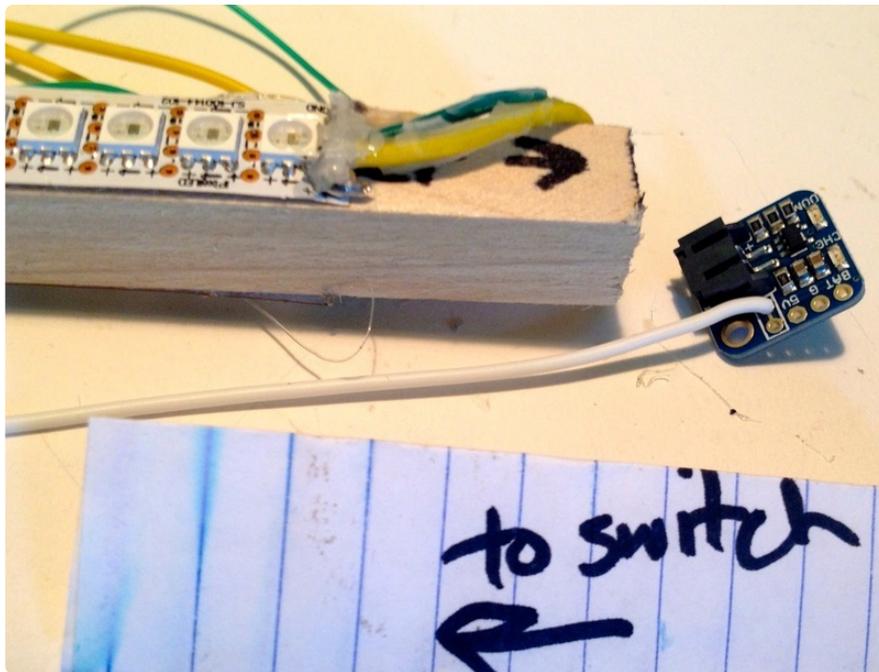


## Pro Trinket & Charger Wiring

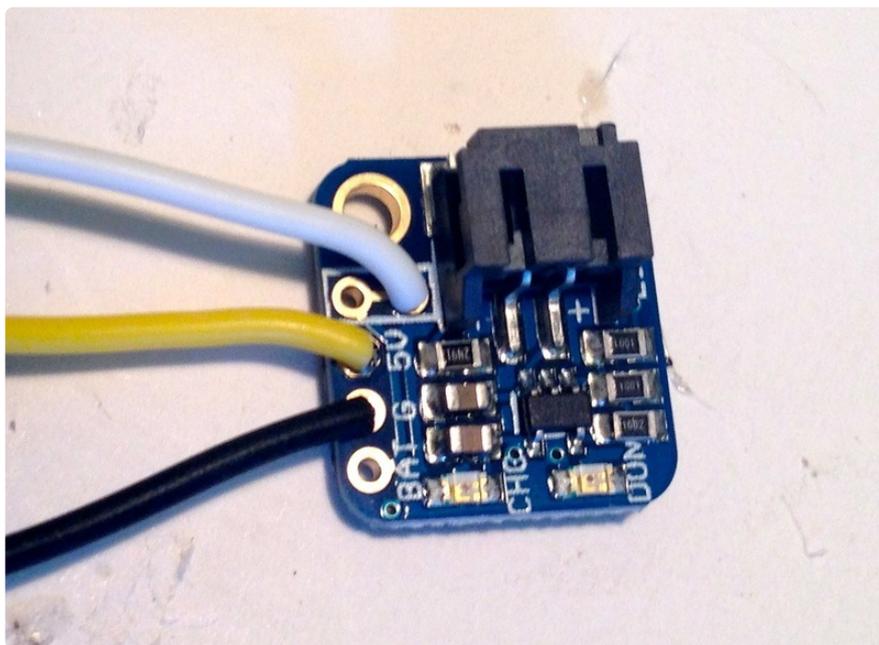
Flip the LiPoly backpack charger over and look at the back. There's a little silvery patch. Take your soldering iron and bridge the two pads here -- this will make your staff charge a lot faster.



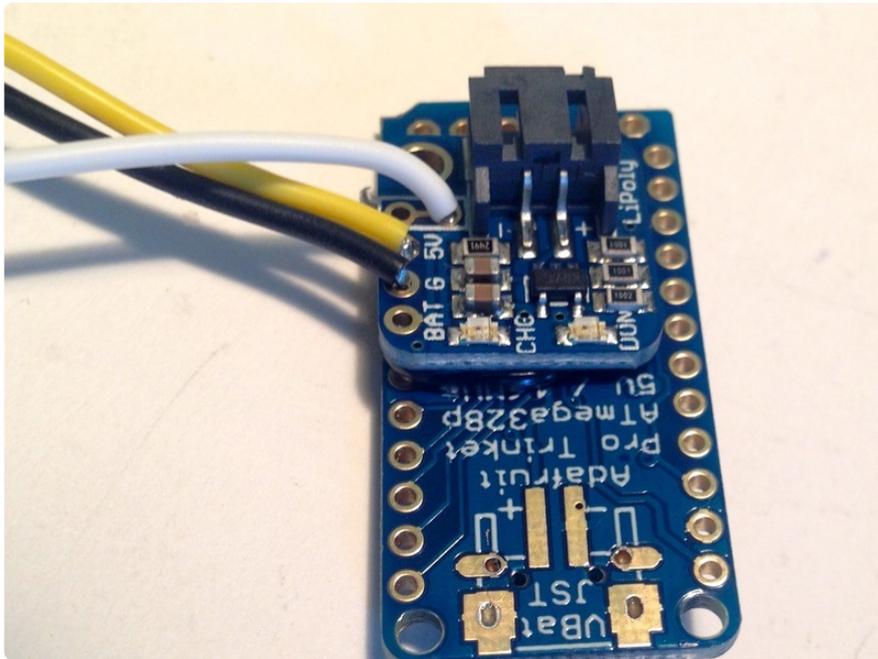
Find the long wire from the switch. Measure it out to a good length and then solder this into the inner power switch hole on the LiPoly Backpack.



Solder a short yellow wire into the 5v pad, and a short black wire into the G pad on the LiPoly backpack.



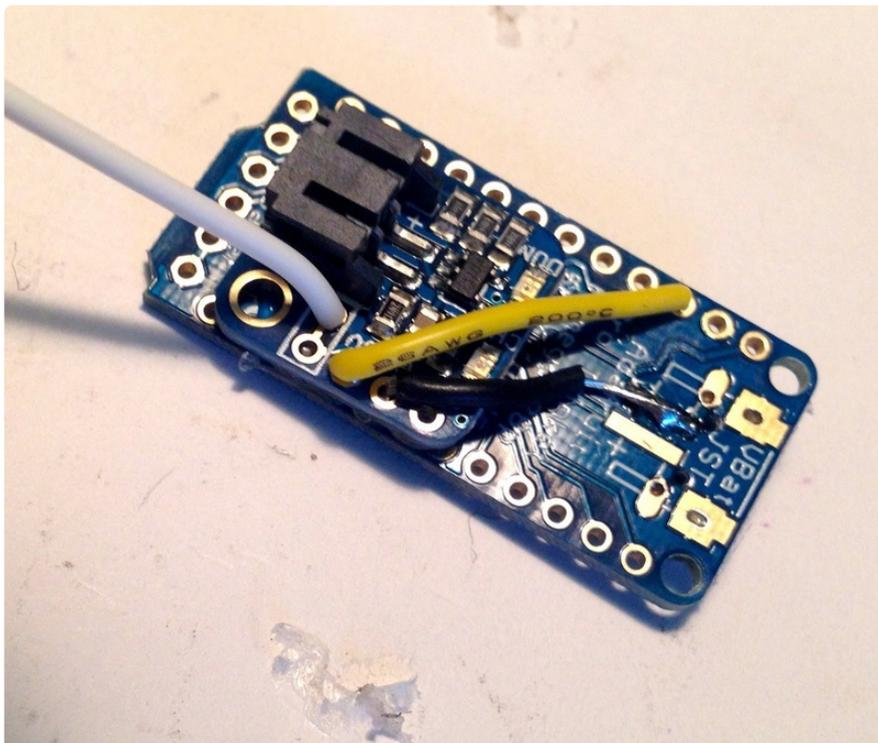
Place the LiPoly backpack on the back of the Pro Trinket, making sure it doesn't block pin 3 or 13. Use a big dollop of hot glue to glue it in place, making sure the glue insulates the back of the backpack from shorting out on the back of the Pro Trinket.



Just a few more wires to hook up!

Connect the yellow wire from 5v on the LiPoly Backpack to the BUS pin on the Pro Trinket.

Connect the black wire from G on the LiPoly Backpack to the - pad on the BACK of the Pro Trinket.

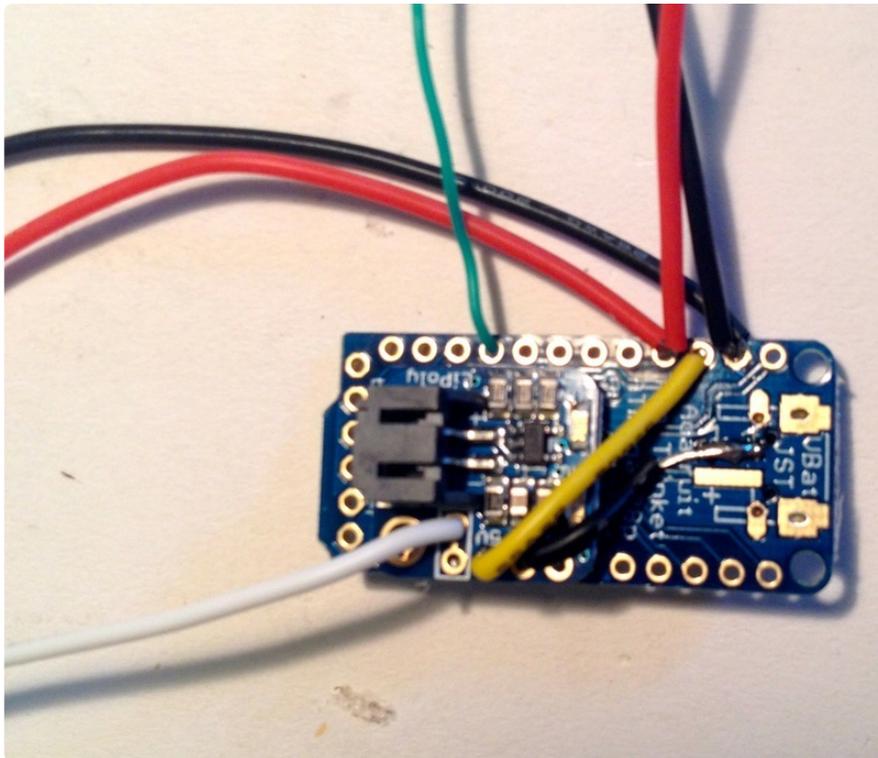


Twist together a long (2 ft) and a short (3 in) red wire, and solder both into the 5V pin on the Pro Trinket.

Twist together a long (2 ft) and a short (3 in) black wire, and solder both into the G pin on the Pro Trinket.

Solder a short (3in) green wire into the 3 pin on the Pro Trinket. These three short wires will later attach to the IR sensor.

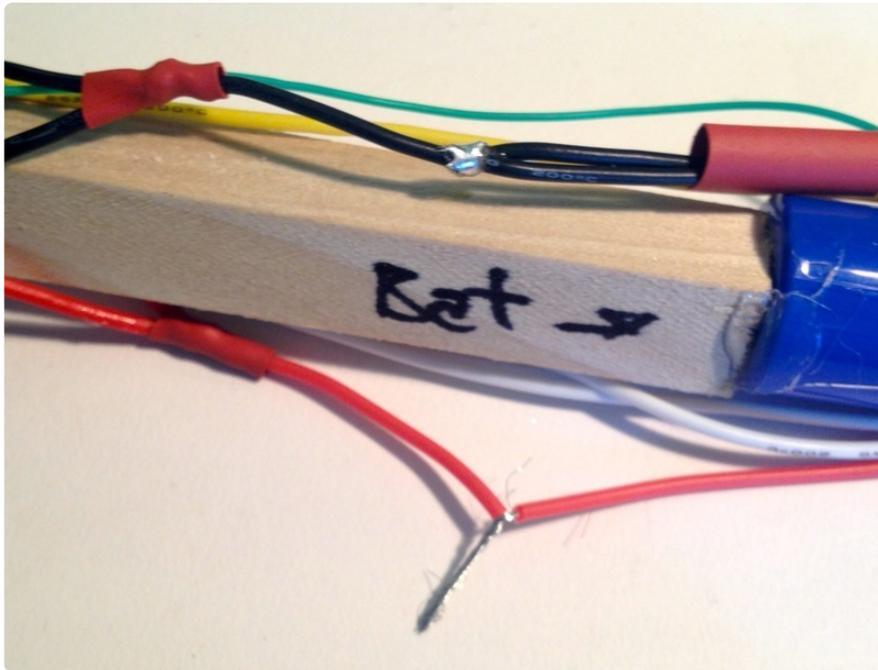
The long wires will go to the LEDs and power switch.



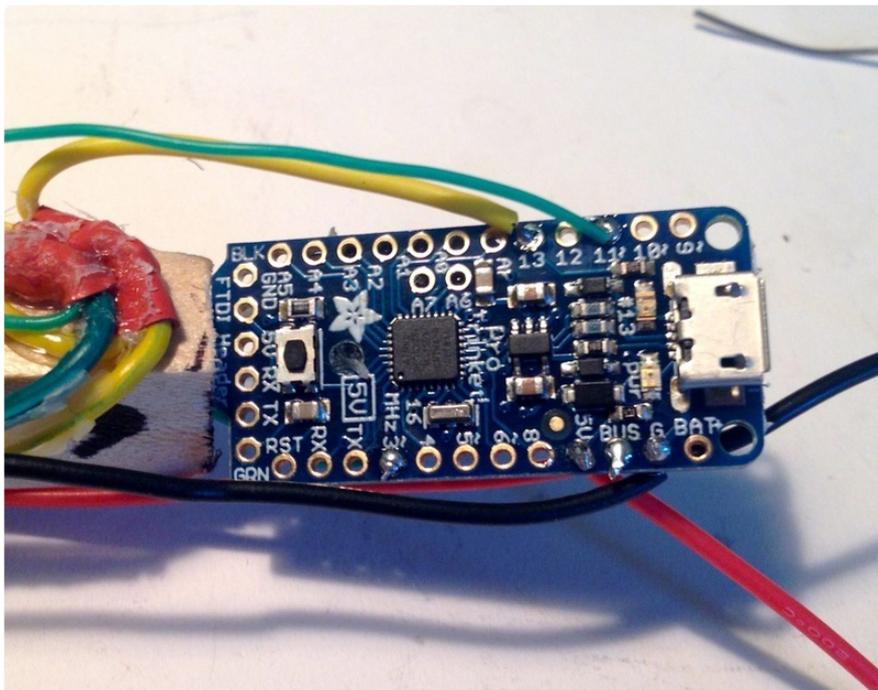
Take the long black wire and run it down to the middle of the staff. Find your twisted pair of black wires coming from the LEDs, and splice it in.

Red is a tiny bit trickier. It's the same idea, but you need to incorporate the other wire coming from the switch as well.

Run the long red wire down toward the battery until you find the loose wire coming from the on/off switch and twist these two together. Slide some heat shrink on and splice this pair together with the red pair coming from the LEDs.



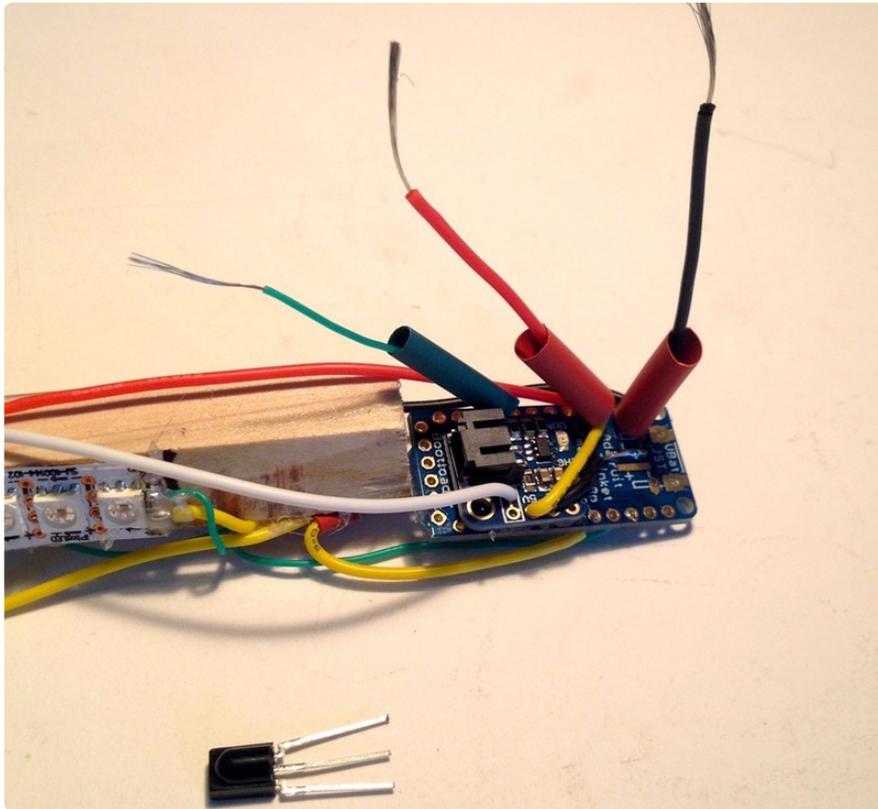
Attach the green (data) pin coming from the LEDs near the Pro Trinket to pin 11.  
Attach the yellow (clock) wire to pin 13.



## IR Sensor

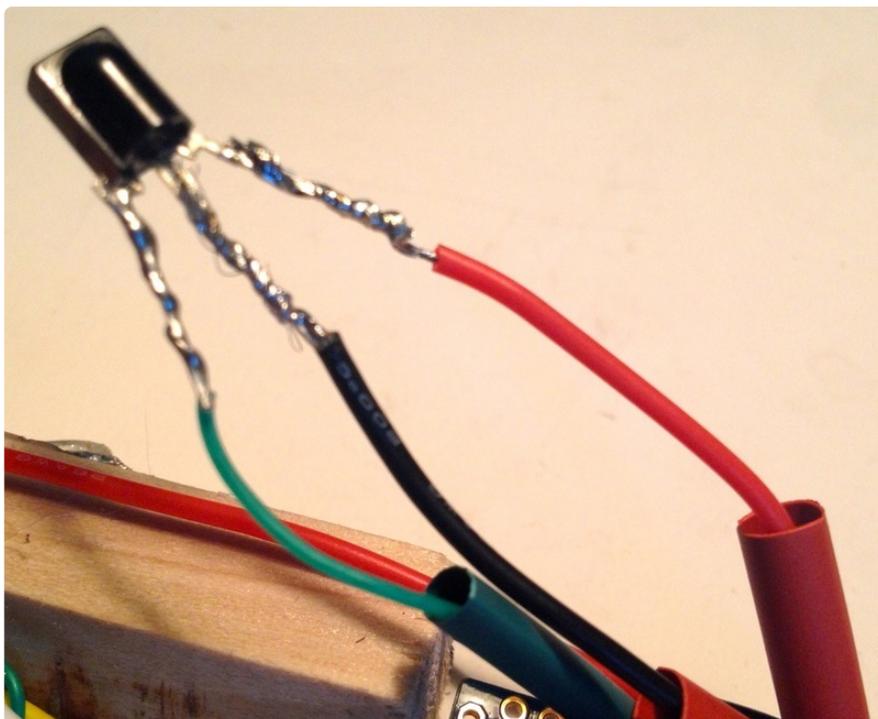
Trim the leads on the IR sensor down to about half their length.

Strip about 1/2 inch (yes, that's a lot!) of shielding from each of the three remaining wires coming from the Pro Trinket and slip a large piece of heat shrink onto each wire.

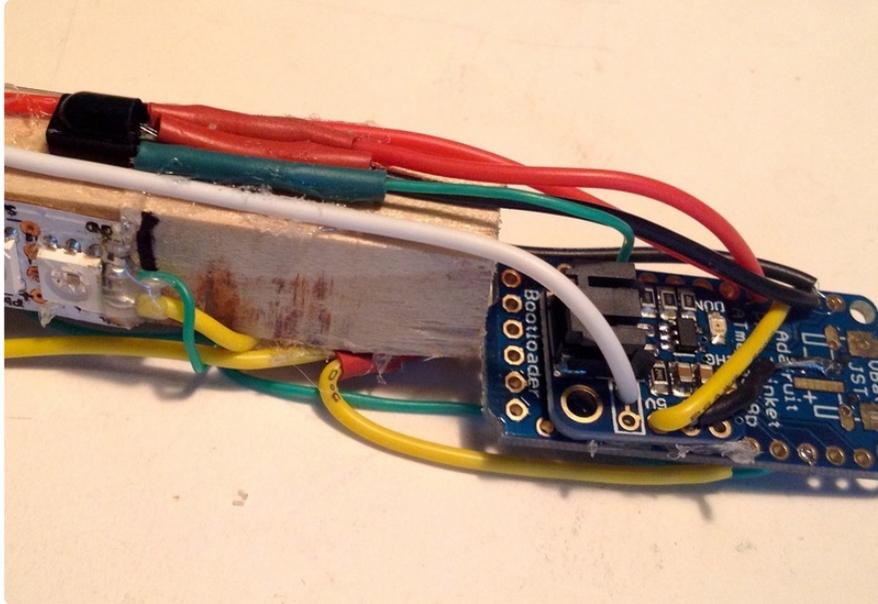


As you're looking at the sensor with the bump facing you and the legs down, coil the green wire (to pin 3) around the left leg, the black wire (to ground) around the center leg, and the red wire (to 5v) around the right leg. Solder and secure with heat shrink.

Be sure you get this right! This sensor will quietly break if you mix up the wires so double check that you're doing it right.



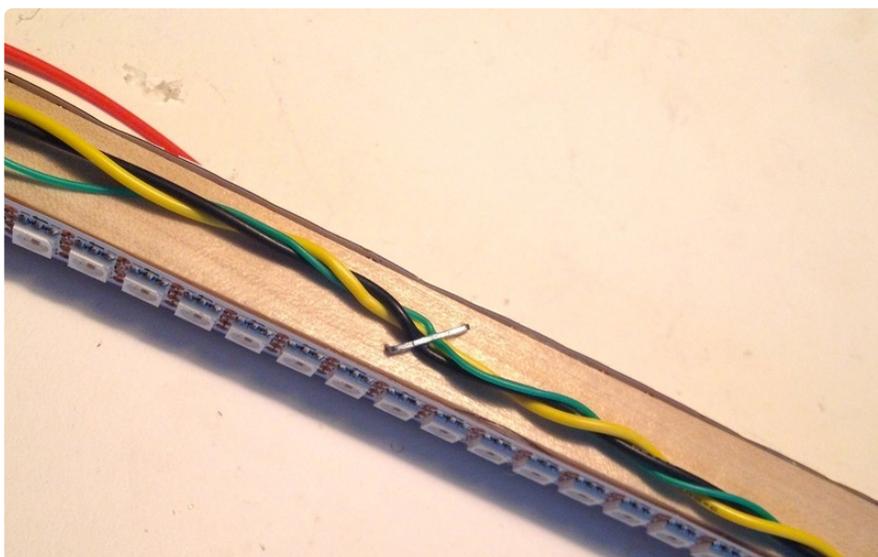
Glue the IR sensor to the side of the dowel with no LEDs and make sure it'll fit nicely inside the tube, without being covered by the cap.



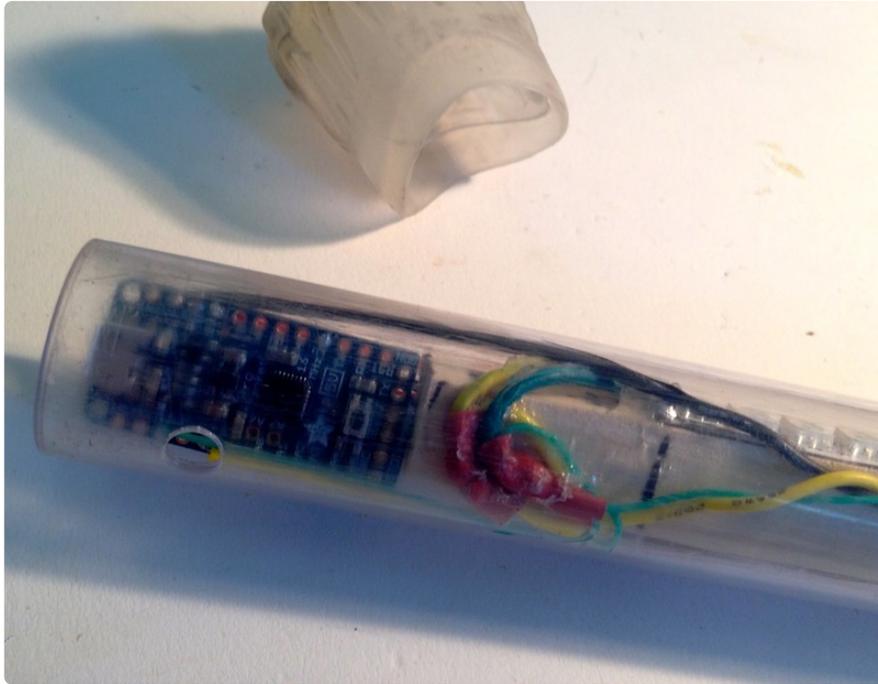
## Battery Wiring

The battery wires are very straightforward. ONE AT A TIME, carefully cut each battery wire, and splice in enough wire to extend the connector to the LiPoly backpack so that it plugs in happily. (Don't ever cut both battery wires at once, in case your cutter accidentally causes a battery short)

Plug that bad boy in and press the switch to make the lights come on! Grab the remote and press some buttons to be sure it's all responding.



Once you're sure everything is working, take a minute to manage your wires so they're comfortably running down the sides of the dowel, and not blocking the LEDs or bunching up anywhere.



Finish up by sliding everything into the tube -- this works best if you start with the Pro Trinket end since the switch is a nice tight fit.



---

## Using the Remote

This remote can be reconfigured in the code to work however you want. Here's how I did it:



Note: The STOP/MODE button will turn all the LEDs off, but it does not turn the Pro Trinket off. If you leave your staff in this mode, without pressing your On/Off switch, your battery will slowly drain. Always use the power switch at the end of the staff to fully power off.

To charge your staff, just plug a USB cable into the Pro Trinket.



---

## Preparing Images

The staff can display **GIF** images with up to **16 colors** (though two-color images — bitmaps — are much smaller and you can store more of them). Images are 36 pixels tall (or however many LEDs your staff has), and a maximum of 255 pixels wide (but can be as narrow as 1 pixel).

The process for converting images is a little gritty right now, requiring a command-line tool written in Python. It also requires the Python Imaging Library (PIL).

Probably the least-bothersome way to do this right now is on a Raspberry Pi computer, where most of the tools are already built-in, though this requires some familiarity with the Linux operating system.

Phil says he'd like to make a more user-friendly tool for this in the future. But for the time being these steps remain a bit technical.

Installing and using Python varies from system to system. On the Raspberry Pi, Python is already installed by default, though PIL must be added manually:

## GitHub repo for the conversion software

<https://adafru.it/E-R>

```
sudo apt-get install python-imaging
```

Things will be entirely different on Windows or Mac or even on other Linux distributions. Unfortunately setting up Python is way beyond the scope of this guide, so you might Google 'round for tutorials elsewhere. If this gets too dry and technical, don't fret...I suspect that given time other users will post some good poi-ready images to the [Adafruit Forums \(https://adafru.it/cer\)](https://adafru.it/cer).

Inside the "convert" folder included with the project is a file called "convert.py" — the Python script which readies images for the staff.

As written, this code is designed for the [Genesis Poi \(https://adafru.it/lgE\)](https://adafru.it/lgE) project, which use fewer LEDs and a much smaller battery. For maximum eye-searing potential on our staffs, we need to make some small changes to the script. Open the file in a text editor and look for these three lines near the beginning:

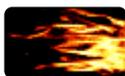
```
batterySize    = 150 # Battery capacity, in milliamp-hours (mAh)
runTime        = 1.1 # Est. max run time, in hours (longer = dimmer LEDs)
parallelStrips = 2   # Same data is issued to this many LED strips
```

Change these to:

```
batterySize    = 2200 # Battery capacity, in milliamp-hours (mAh)
runTime        = 2.5  # Est. max run time, in hours (longer = dimmer LEDs)
parallelStrips = 4    # Same data is issued to this many LED strips
```

Save the changes to the file. But if you'll ever be converting images for the Genesis Poi in the future, you'll need to change those lines back.

So, let's suppose we have this little flames image, which is 36 pixels tall (the same number as my LED count on each end):



The top of the image will correspond to the tip of the poi. To convert this using the Python script, you'd type:

```
python convert.py images/flames.gif &gt; graphics.h
```

Or you can convert a whole list of images:

```
python convert.py images/*.gif > graphics.h
```

The “> graphics.h” redirects the output of the convert.py script to the plain-text file graphics.h, which can then be incorporated into an Arduino sketch.

Inside the file you’ll see one or more sections like this:

```
// usa.gif -----  
  
const uint8_t PROGMEM palette04[][3] = {  
  { 56, 56, 56 },  
  { 56,  0,  0 },  
  {  0,  0,  0 },  
  {  0,  3, 56 } };  
  
const uint8_t PROGMEM pixels04[] = {  
  0X22, 0X22, 0X22, 0X22, 0X22, 0X22, 0X22, 0X22,  
  0X22, 0X22, 0X22, 0X22, 0X22, 0X22, 0X22, 0X22,  
  0X10, 0X10, 0X10, 0X10, 0X10, 0X10, 0X12, 0X22,  
  0X33, 0X33, 0X33, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X30, 0X30, 0X30, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X33, 0X03, 0X03, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X30, 0X30, 0X30, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X33, 0X03, 0X03, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X30, 0X30, 0X30, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X33, 0X03, 0X03, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X30, 0X30, 0X30, 0X30, 0X10, 0X10, 0X12, 0X22,  
  0X33, 0X33, 0X33, 0X30, 0X10, 0X10, 0X12, 0X22 };
```

Above is the data for an American flag...a four-entry color palette (white, red, black, blue) followed by the pixel data (packed two pixels per byte).

Then, near the bottom of the file, you’ll see a block like this:

```
const image PROGMEM images[] = {  
  { PALETTE1 , 100, (const uint8_t *)palette00, pixels00 },  
  { PALETTE4 ,  48, (const uint8_t *)palette01, pixels01 },  
  { PALETTE4 ,  54, (const uint8_t *)palette02, pixels02 },  
  { PALETTE4 ,   1, (const uint8_t *)palette03, pixels03 },  
  { PALETTE4 ,  24, (const uint8_t *)palette04, pixels04 },  
  { PALETTE4 ,   9, (const uint8_t *)palette05, pixels05 },  
  { PALETTE4 ,  26, (const uint8_t *)palette06, pixels06 }  
};
```

This table holds references to all of the images in the file, along with their widths in pixels (height is always 36) and format (bitmap or up to 16 colors).