



Fiber Optic Whip

Created by Erin St Blaine



<https://learn.adafruit.com/fiber-optic-whip>

Last updated on 2024-06-03 02:12:47 PM EDT

Table of Contents

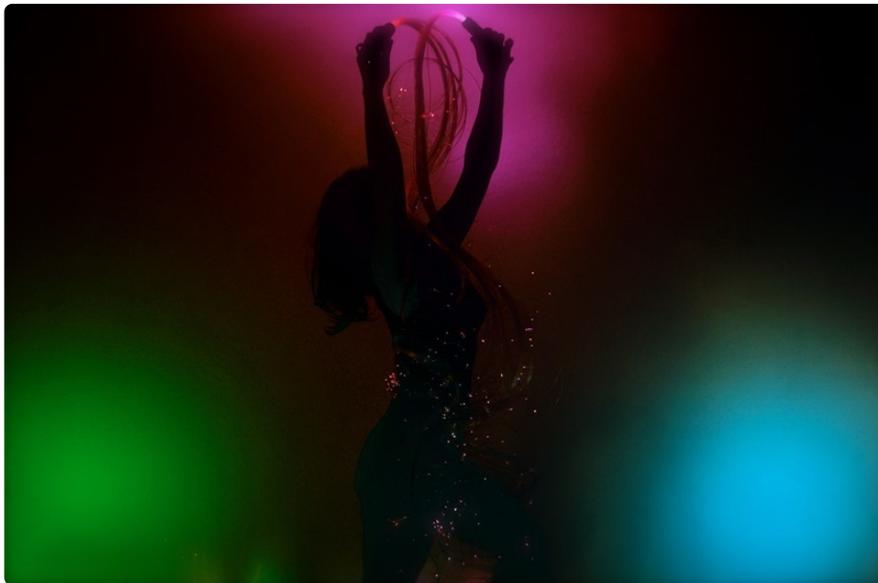
Introduction	3
<ul style="list-style-type: none">• Fiber Optics• Parts & Materials• Tools	
Wiring Diagram	6
<ul style="list-style-type: none">• Wiring	
The Code	7
<ul style="list-style-type: none">• If you encounter trouble...	
3d Printing	12
Assembly	13
<ul style="list-style-type: none">• On/Off Switch• Mode Button• Battery Charger / LiPoly Backpack• Neopixel Jewel• Add the Mode Button• Handle Assembly	
Finishing	18

Introduction

This whip makes me feel like I'm dancing with a thousand fireflies.

Its light source is a neopixel jewel, with seven addressable pixels allowing for an amazing array of color changes and animations. The end-glow fibers dance around in the air and create a whirl of color points that is mesmerizing to watch and incredibly fun to play with. I never want to put it down.

The handle is 3d printed and fits the battery and all the electronics inside. There's a button for changing modes and an on-off switch. It's got an onboard battery charger too, so you can play with it for hours and then recharge by plugging in a USB cable.

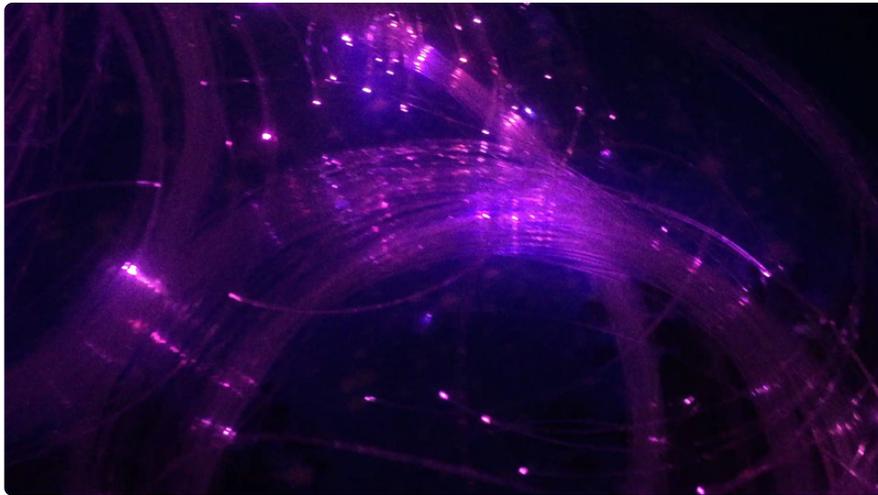


Difficulty Level

This project requires decent soldering skills and the ability to upload code with the Arduino IDE. It's also a little fiddly to get everything situated in the handle correctly.

You don't need mad coding skills -- you can copy and paste the code into your IDE and use it as-is.

Also don't worry if you don't have a 3d printer! There are services available that will print the handle and ship it to you.



Fiber Optics

Fiber optics are fiddly beasts. You can order them in bulk online but lining them up with the light source is hard -- to do it right you need specialty tools like a hot knife and fancy connectors.

I left that part of it to the pros, and ordered [this pre-assembled fiber optic bundle from Weidamark](https://adafru.it/zfR) (<https://adafru.it/zfR>). Shop around on the site -- there are several different options for cable length and number of fibers. The 12 foot, 144 strand bundle is what I used.

Parts & Materials

1 x Pro Trinket Pro Trinket 5v	https://www.adafruit.com/product/2000
1 x Neopixel Jewel Neopixel Jewel	https://www.adafruit.com/product/2226
1 x LiPoly Backpack Charger Add-On Battery Charger	https://www.adafruit.com/product/2124
1 x Momentary Switch Metal mode-change switch	https://www.adafruit.com/product/3347
1 x Slide Switch On/off Switch	https://www.adafruit.com/product/805

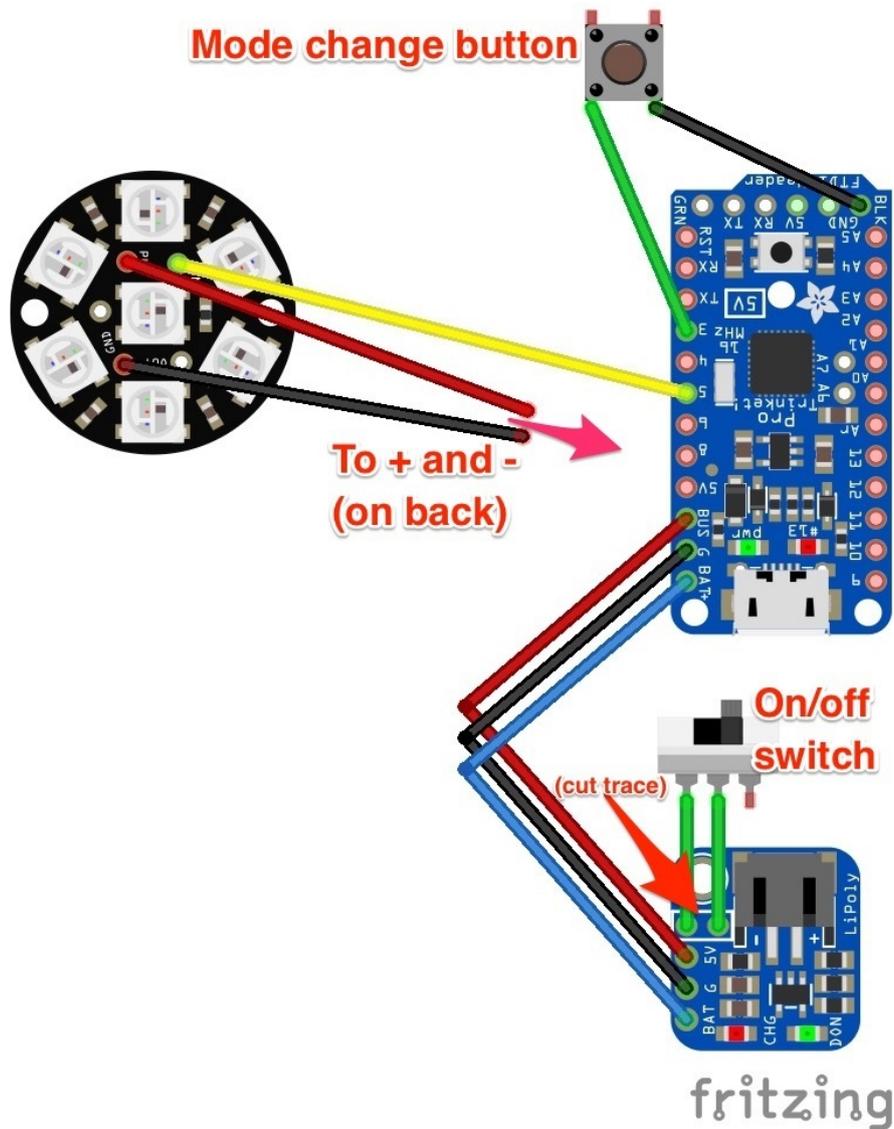
You'll also need the [fiber optic bundle \(https://adafru.it/zfS\)](https://adafru.it/zfS) and a 3d printed handle. I also added a [wrist strap \(https://adafru.it/zfT\)](https://adafru.it/zfT) to minimize dropping.

Tools

- Soldering Iron & accessories
- Heat shrink tubing
- Hot glue gun
- Gaffer's tape or duct tape
- USB cable



Wiring Diagram



Wiring

The LiPoly backpack sits neatly on top of the Pro Trinket with the three pins (BAT, G, 5v/BUS) aligning.

To enable the on/off switch capability, cut the trace between the two switch pads with a utility knife.

Power the Neopixel Jewel by soldering to the two power pads on the **back** of the Pro Trinket.

Pro Trinket BAT	Charger BAT
Pro Trinket G	Charger G
Pro Trinket BUS	Charger 5V
Pro Trinket 5	Neopixel Jewel IN
Pro Trinket + (on back)	Neopixel Jewel +
Pro Trinket - (on back)	Neopixel Jewel G
Pro Trinket GND	Momentary Switch
Pro Trinket 3	Momentary Switch
On/Off Switch	Charger Switch Pads

The Code

Software Setup

If this is your first time using Pro Trinket, take a look at [Introducing Pro Trinket \(https://adafru.it/iPe\)](https://adafru.it/iPe) to get a guided tour.

Once you've got your Pro Trinket up and running with [Arduino \(https://adafru.it/dgt\)](https://adafru.it/dgt), you'll need to install the FastLED library. (**Sketch > Include Library > Manage Libraries...**)

Libraries? Why? What's a Library?

In a nutshell, Arduino libraries have a lot of pre-written functions that make your neopixels easy to command. You can do fancy stuff without being a code guru. Yay Libraries!

FastLED is a fast, efficient, easy-to-use Arduino library for programming addressable LED strips and pixels. It has a lot of features to get your animations up and running fast -- and it has a lot of code samples available if you're just learning to code.

[All about Arduino Libraries \(https://adafru.it/dNR\)](https://adafru.it/dNR) will tell you everything you ever wanted to know about libraries, including more detailed installation instructions.

Once your curiosity is satiated and your library is installed, copy and paste the code into your Arduino window.

Go to your Tools menu and select "Pro Trinket 5V USB" from the list of boards. Plug your Pro Trinket into your computer via the onboard USB port. Press the "reset" button on your Pro Trinket and wait for the blinky red light, then click the upload button in Arduino.

```
//Code by Erin St. Blaine for Adafruit.com

#include <FastLED.h>

#define LED_PIN    5
#define COLOR_ORDER GRB
#define NUM_LEDS 7

int HUE = 0;
int SATURATION = 255;
int BRIGHTNESS = 255;
int STEPS = 7;

const byte BUTTON_PIN = 3;
uint8_t gHue = 0; // rotating "base color" used by confetti

CRGB leds[NUM_LEDS];
TBlendType currentBlending;
CRGBPalette16 currentPalette;

//BUTTON SETUP STUFF
byte prevKeyState = HIGH; // button is active low

unsigned long keyPrevMillis = 0;
const unsigned long keySampleIntervalMs = 25;
byte longKeyPressCountMax = 80; // 80 * 25 = 2000 ms
byte longKeyPressCount = 0;

int ledMode = 0; //FIRST ACTIVE MODE

//-----SETUP-----
void setup()
{
```

```

    delay( 3000 ); // power-up safety delay
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    FastLED.addLeds<WS2812B, LED_PIN, COLOR_ORDER>(leds,
NUM_LEDS).setCorrection( TypicalLEDStrip );
    FastLED.setBrightness( BRIGHTNESS );
    currentBlending = LINEARBLEND;
}

#define NUM_MODES 6
//-----MAIN LOOP-----
void loop() {
    switch (ledMode) {
        case 999: break;
        case 0: confetti(); 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: currentPalette = RainbowColors_p; rainbow(); break;
        case 6: currentPalette = RainbowStripeColors_p; rainbow(); break;
        case 888: ledMode=0; break;
    }

    // key management section
    if (millis() - keyPrevMillis >= keySampleIntervalMs) {
        keyPrevMillis = millis();

        byte currKeyState = digitalRead(BUTTON_PIN);

        if ((prevKeyState == HIGH) && (currKeyState == LOW)) {
            keyPress();
        }
        else if ((prevKeyState == LOW) && (currKeyState == HIGH)) {
            keyRelease();
        }
        else if (currKeyState == LOW) {
            longKeyPressCount++;
        }

        prevKeyState = currKeyState;
    }

    EVERY_N_MILLISECONDS( 20 ) { gHue++; } // slowly cycle the "base color" through
the rainbow
    FastLED.show();
    FastLED.delay(20);
}

//BUTTON CONTROL STUFF
// called when button is kept pressed for less than 2 seconds
void shortKeyPress() {
    Serial.println("short");
    ledMode++;
    if (ledMode > NUM_MODES){
        ledMode=0; }
}

// called when button is kept pressed for more than 2 seconds
void longKeyPress() {
    Serial.println("long");
    ledMode=888;
}

// called when key goes from not pressed to pressed

```

```

void keyPress() {
    Serial.println("key press");
    longKeyPressCount = 0;
}

// called when key goes from pressed to not pressed
void keyRelease() {
    Serial.println("key release");
    if (longKeyPressCount >= longKeyPressCountMax) {
        longKeyPress();
    }
    else {
        shortKeyPress();
    }

    // other code goes here
}

// SOLID -----
void solid()
{
    fill_solid(leds, NUM_LEDS, CHSV( HUE, SATURATION, BRIGHTNESS));
    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);
        colorIndex += STEPS;
    }
}

void confetti()
{
    // random colored speckles that blink in and fade smoothly
    fadeToBlackBy( leds, NUM_LEDS, 10);
    int pos = random16(NUM_LEDS);
    leds[pos] += CHSV( gHue + random8(64), 200, 255);
    FastLED.show();
}

```

If you encounter trouble...

Any time you hit a roadblock with a neopixel project, we'll usually ask that you start with the "strandtest" example from our own Adafruit_NeoPixel library. This helps us narrow down whether it's a hardware or software issue. The library is installed similarly to FastLED or any other in Arduino (**Sketch > Include Library > Manage Libraries...**)

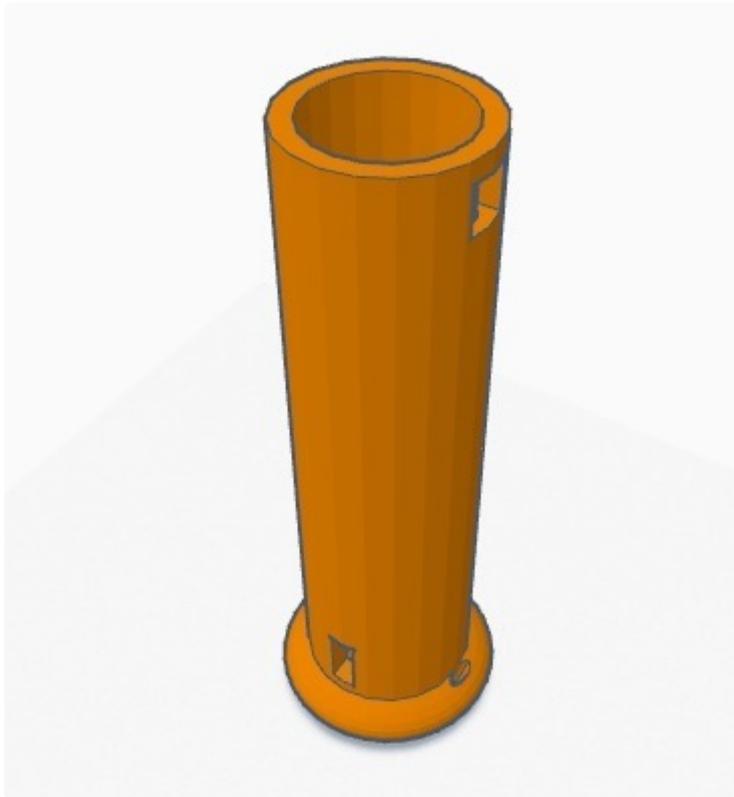
You'll find the strandtest example under **File→Sketchbook→Libraries→Adafruit_NeoPixel→strandtest**

If strandtest fails to run, this suggests a hardware issue...for example, connecting to the wrong pin on the Pro Trinket.

If you're new to Arduino programming and LEDs, we usually suggest starting with the Adafruit_NeoPixel library...it's pretty basic, the strip declaration is more conventional, and we can stay on top of keeping it compatible with our own products and the most mainstream Arduino boards.

As FastLED is a more "bleeding edge" third-party library, **we can't always guarantee compatibility across versions or with specific boards.** You can find help through their [community on Google+ \(https://adafru.it/ebn\)](https://adafru.it/ebn). This is potent stuff, written by people with a deep appreciation for LED art, and we wanted to showcase it.

3d Printing



This handle was created in Tinkercad. It has a port for the button and one for the switch, and it matches the exterior dimension of the fiber optic bundle while leaving the right amount of space inside for the Neopixel Jewel and the electronics.

I added a ring around the base to keep it from slipping out of my hand, and a hole in the ring to attach a wrist strap for extra safety. [Add your own details \(https://adafru.it/zdr\)](https://adafru.it/zdr) -- every Jedi's lightsaber handle should be unique.

It prints with no supports needed. The largest dimension is the height -- it is 119mm high, which was just under the size limit of my Up! Mini 3d printer.

I found it printed best using the "fast draft" print setting. The higher quality settings overloaded the printer, possibly because this model is so close to the size limit for the bed. Your results may vary.

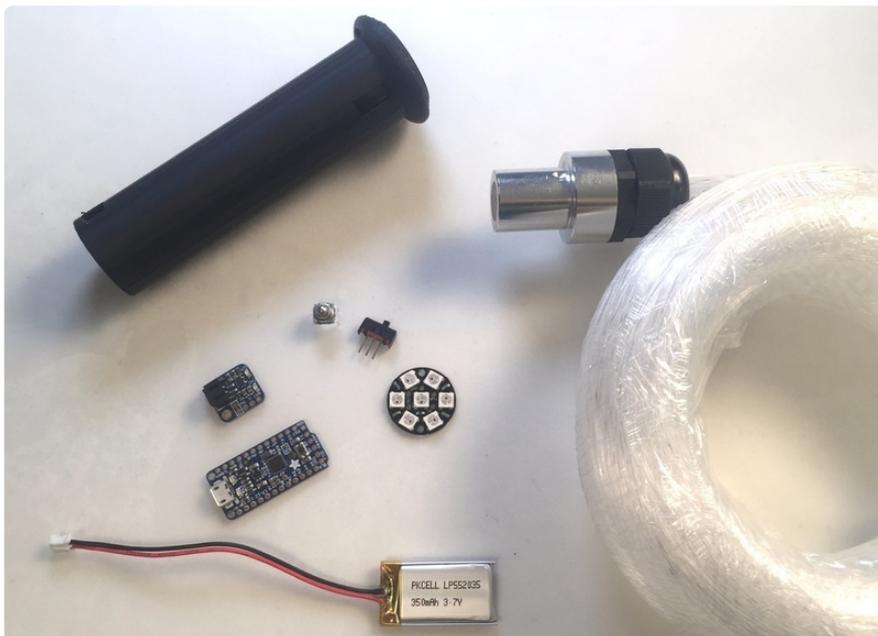
[Download .stl file from Thingiverse](https://adafru.it/zds)

<https://adafru.it/zds>

If you don't have access to a 3d printer, you can [order a pre-printed handle from Shapeways. \(https://adafru.it/zdt\)](https://adafru.it/zdt)

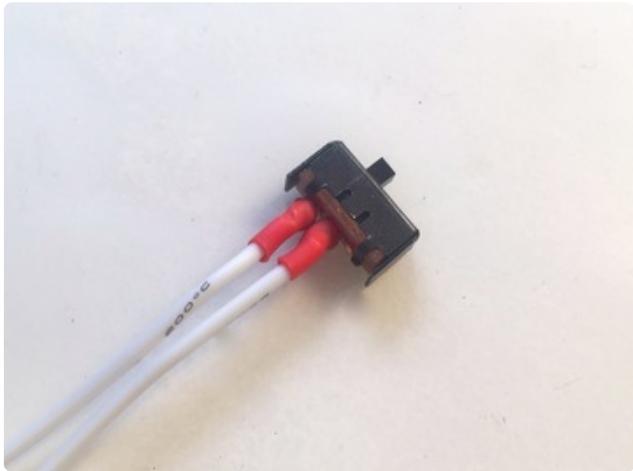


Assembly



Be sure you've uploaded the code to your Pro Trinket before assembly. That will make it a lot easier to test at each step and be sure things are working.

On/Off Switch



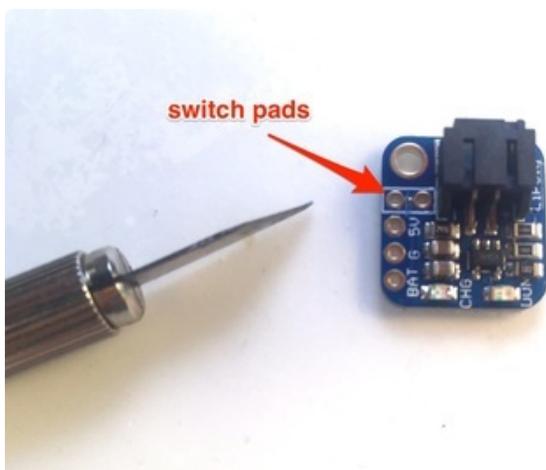
Trim one of the side legs off your switch and trim the other legs so they're a little shorter. Solder a 4" white wire to both remaining legs and cover with heat shrink.

Mode Button

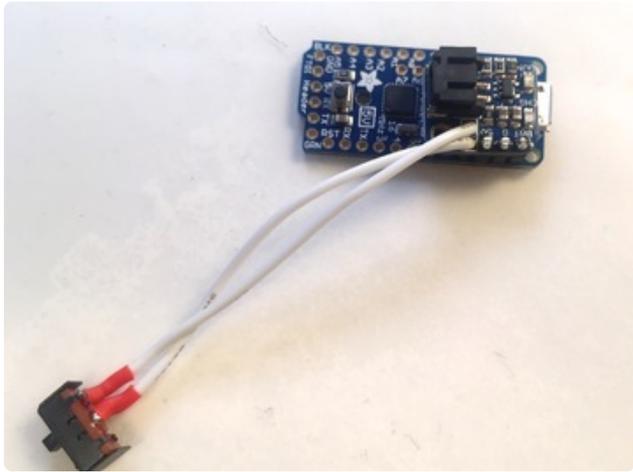


Trim the side legs off one side of your switch button. Flatten the remaining two legs out so the switch lies flat on your table. Solder a 7" white and black wire to these two legs and cover with heat shrink.

Battery Charger / LiPoly Backpack



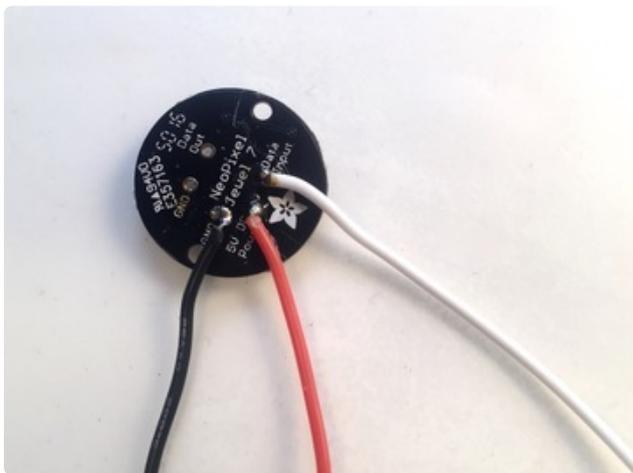
Cut the trace between the two switch pads on the LiPoly backpack with a utility knife. Solder the two white wires coming from your switch into these two holes.



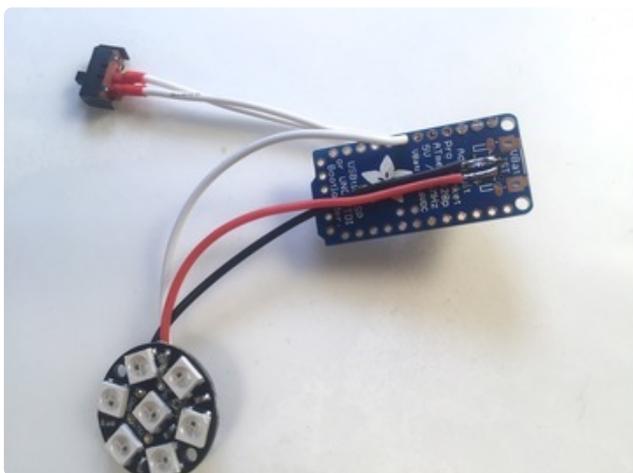
Stack the LiPoly backpack on top of the Pro Trinket using the included headers, aligned as shown. Solder the headers in place on both boards and trim the header legs close to the boards.

Plug your battery into your charger and flip the switch. The lights on the Pro Trinket should come on so you know it's working.

Neopixel Jewel



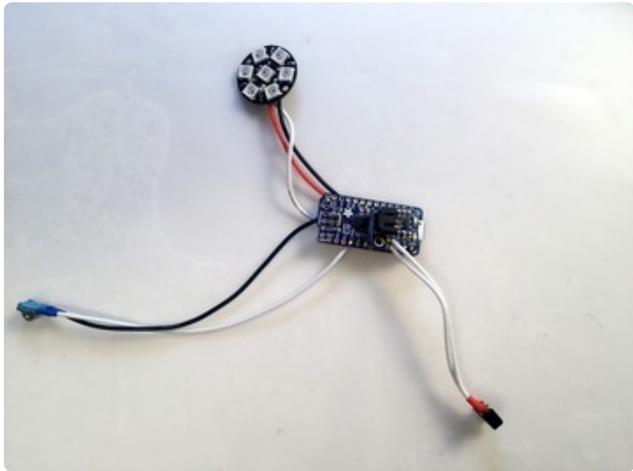
Solder a 7-8" red wire to +, black wire to G and white wire to "IN" on the back of the Neopixel Jewel.



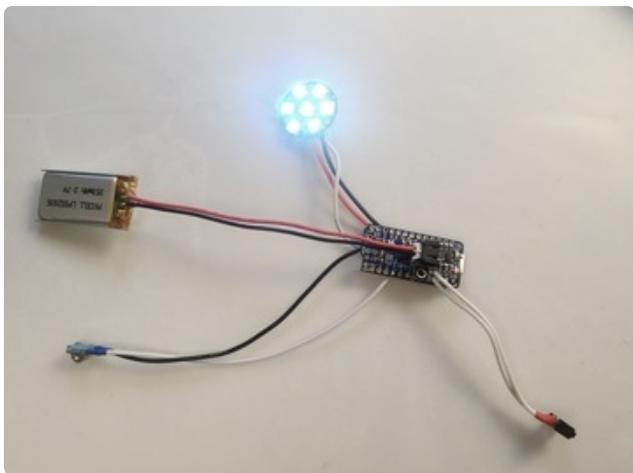
Solder the white wire to pin 5 on the Pro Trinket. Solder the power and ground wires to the + and - pads on the back of the Pro Trinket.

Plug your battery in again and test to be sure the Neopixel Jewel comes on.

Add the Mode Button



Solder the black wire from the mode button to the G pin on the bottom end of the Pro Trinket. Solder the white wire to pin 3.



Plug your battery in once more and click the mode button to see the Neopixel Jewel change color modes. Tug on all your connections and wiggle things around to make sure all your solder joints are tight.

Handle Assembly

Getting everything into the handle is a little tricky, but it all fits neatly with a little bit of patience and care. Needle-nose pliers are very helpful here.



Feed the electronics into the handle from the top. Place the on-off switch first. It helps to pull it all the way through while getting the Pro Trinket in place and then press it back so it's flush with the outside of the case. Glue it in place with hot glue.



Once you have the switch in its hole, settle the Pro Trinket at the bottom of the case so you can access the USB port. Glue it in place, being careful not to get any glue inside the USB port.



Slide the battery down next to the Pro Trinket, then set the mode button in place. I found it easiest to tape the button and wires against the inner wall of the handle with gaffer's tape. The handle has a shallow track inside below the switch hole where the wires feed through past the Neopixel Jewel.



Lastly, settle the Neopixel Jewel in place. There's a tiny ledge that will hold it at just the right spot. It's a little tricky to get it settled -- if you're having trouble, be sure the switch wires are settled neatly into the wire channel.

TEST AGAIN AT THIS POINT! Make sure you haven't knocked any wires loose with all that fiddling.



Wrap a few layers of gaffer's tape around the fiber bundle until it fits snugly inside the handle. Gently press it in as far as it will go, being careful not to dislodge the mode button.

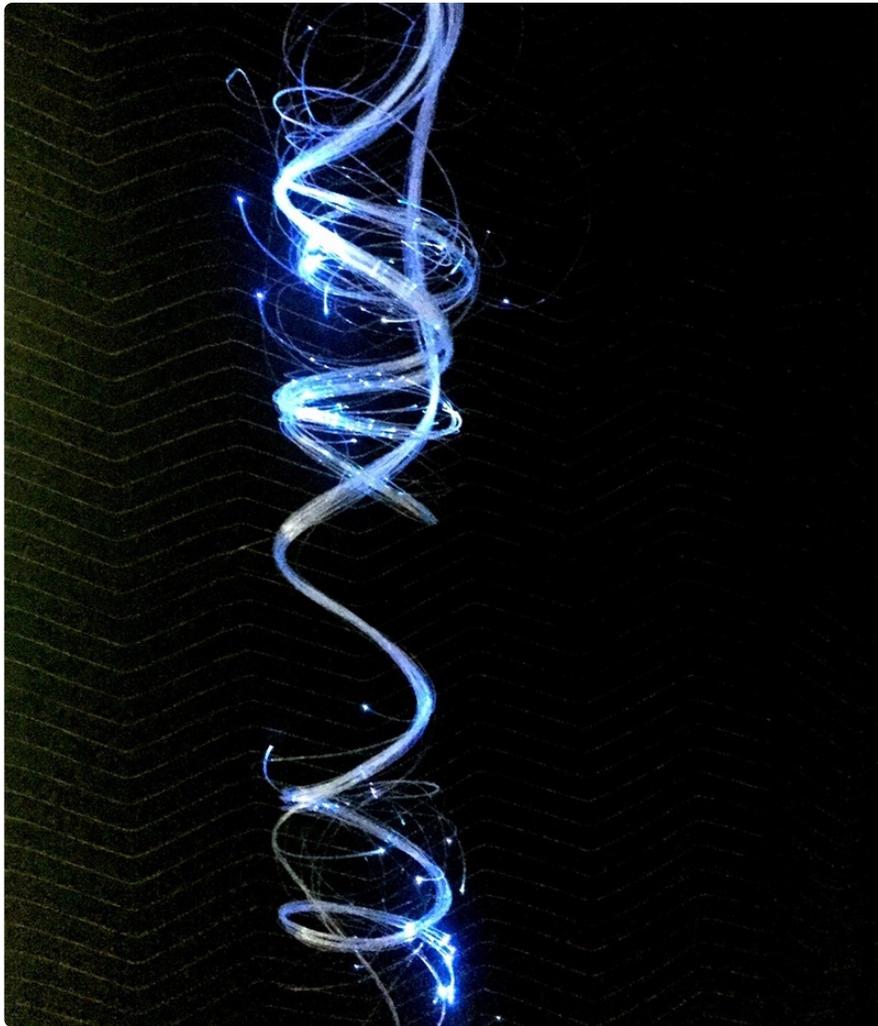


Add a wrist strap to the hole at the bottom of the handle. Swing it around to be sure the fibers won't come out!

Finishing



Hang the whip up near the ceiling so the fibers spill down. Take a sharp pair of scissors and trim the fibers at all different lengths so the points of light at the end of the fibers are scattered across the length of your whip. This will create a gorgeous swarm-of-fireflies effect.



You're finished! Take it someplace dark and dance your heart out.