



Superhero Power Gauntlet

Created by Richard Albritton



<https://learn.adafruit.com/superhero-power-gauntlet>

Last updated on 2024-06-03 01:59:19 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Parts• Tools Used	
3D Prints	5
Electronics	9
First Steps	11
Power	13
NeoPixels	14
Accelerometer	16
Sound FX Board	17
<ul style="list-style-type: none">• Getting the Sound Files• Loading the Sounds• Wiring the Board	
Programming	21
Using the Gauntlet	26
<ul style="list-style-type: none">• Powering up the glove• Using the blast• Disabling the motion control• Charging the battery	

Overview

This project will help you build a super hero glove that lights up, plays sound, and reacts to your movement. The sound effects are easy to change and if you are willing to get into a bit of code the LEDs are easy to adjust as well.

This project involves electronics, 3D printing with post production painting, and some premade Arduino Code.

Now on the to stuff we will need to build our project.

Parts

1 x [Adafruit Feather 32u4 Basic Proto](https://www.adafruit.com/product/2771)

<https://www.adafruit.com/product/2771>

This is the microcontroller brain of the project with plenty of processing power and built-in battery charger.

1 x [Adafruit Audio FX Mini Sound Board](https://www.adafruit.com/product/2342)

<https://www.adafruit.com/product/2342>

This is what adds the sound and makes it easy to change sound FX files.

1 x [Mono Audio Amplifier](https://www.adafruit.com/product/2130)

<https://www.adafruit.com/product/2130>

Adafruit Mono 2.5W Class D Audio Amplifier - PAM8302

1 x [Speaker](https://www.adafruit.com/product/1890)

<https://www.adafruit.com/product/1890>

Mini Metal Speaker w/ Wires - 8 ohm 0.5W

1 x [Accelerometer/Compass Sensor - LSM303](https://www.adafruit.com/product/1247)

<https://www.adafruit.com/product/1247>

Used for detecting the orientation of the glove to trigger effects.

1 x [NeoPixel Ring](https://www.adafruit.com/product/1463)

<https://www.adafruit.com/product/1463>

16 x 5050 RGB LED with Integrated Drivers

1 x [NeoPixel Jewel](https://www.adafruit.com/product/2226)

<https://www.adafruit.com/product/2226>

7 x 5050 RGB LED with Integrated Drivers

1 x [Solder Wire](https://www.adafruit.com/product/1886)

<https://www.adafruit.com/product/1886>

Any solder will do but I recommend 0.5mm Rosin core solder.

Hook-up Wire Spool Set - 22AWG Stranded-Core - 10 x 25ft

1 x Stranded Wire

<https://www.adafruit.com/product/3175>

Hook-up Wire Spool Set - 22AWG Stranded-Core - 10 x 25ft

1 x Battery

<https://www.adafruit.com/product/258>

Lithium Ion Polymer Battery - 3.7v 1200mAh

1 x Slide Switch

<https://www.adafruit.com/product/805>

Breadboard-friendly SPDT Slide Switch

1 x Button

<https://www.adafruit.com/product/1489>

Tactile Switch Buttons (6mm slim) x 20 pack

Tools Used

1 x Soldering Iron

<https://www.adafruit.com/product/180>

Any Soldering Iron can be used, but this is a good one if you don't already have it.

1 x Flush cutter

<https://www.adafruit.com/product/152>

This is used to trim the extra wire ends after soldering.

1 x Helping Third Hand

<https://www.adafruit.com/product/291>

Not a must have, but it will make things easy to have a while you solder things together.

1 x Wire Stripper

<https://www.adafruit.com/product/527>

Basic but effective for stripping back the insulating cover on your wires.

Optional

1 x Automatic Wire Stripper

This makes stripping wires so easy you will never want to use other strippers

https://www.amazon.com/Capri-Tools-20012-Self-Adjusting-Stripper/dp/B01018CYV8/ref=sr_1_17?keywords=auto+stripper&qid=1559593793&refinement=

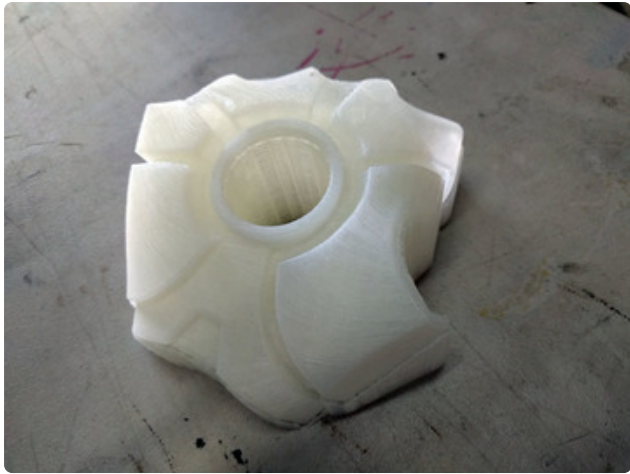
3D Prints



While you can put the electronics into any kind of glove, there is nothing like a nice looking 3D printed part to finish a project. So here is how we did it with a focus on getting a really nice paint job on the external parts.

There are a few models out there that are available for download, so we won't get too far into that part. Here are a few examples we found:

- <https://www.thingiverse.com/thing:3507482> (<https://adafru.it/EYF>)
- <https://www.thingiverse.com/thing:1623321> (<https://adafru.it/DTW>)
- <https://www.thingiverse.com/thing:1124602> (<https://adafru.it/DTX>)



The important part is that you leave enough room to fit all of the electronics into the glove.

If you make a full gauntlet, it will be easy to store things like the battery, but it may be easier to stash the bulk of the electronics inside the glove itself to reduce the number of wires which could break.

Primer/filler (<https://adafru.it/DTZ>) was used on the parts to get a nice, smooth finish. This takes time and a lot of delicate sanding.

Sand the 3D printed parts to knock down most of the ridges.



Apply two coats of the primer/filler with a light sanding between coats.



It takes a lot of time, but the results are incredible. No one would ever guess you 3D printed any of it.



Once you get things rather smooth, apply a generous coat of black spray paint formulated for plastics.



Let that completely dry, then wet sand it until the parts have a dull and smooth appearance.

Repeat this process with your final 2 coats of red, gold, silver or whatever final color paint you will be using.

Now you can add weathering because nothing looks new and shiny for long.

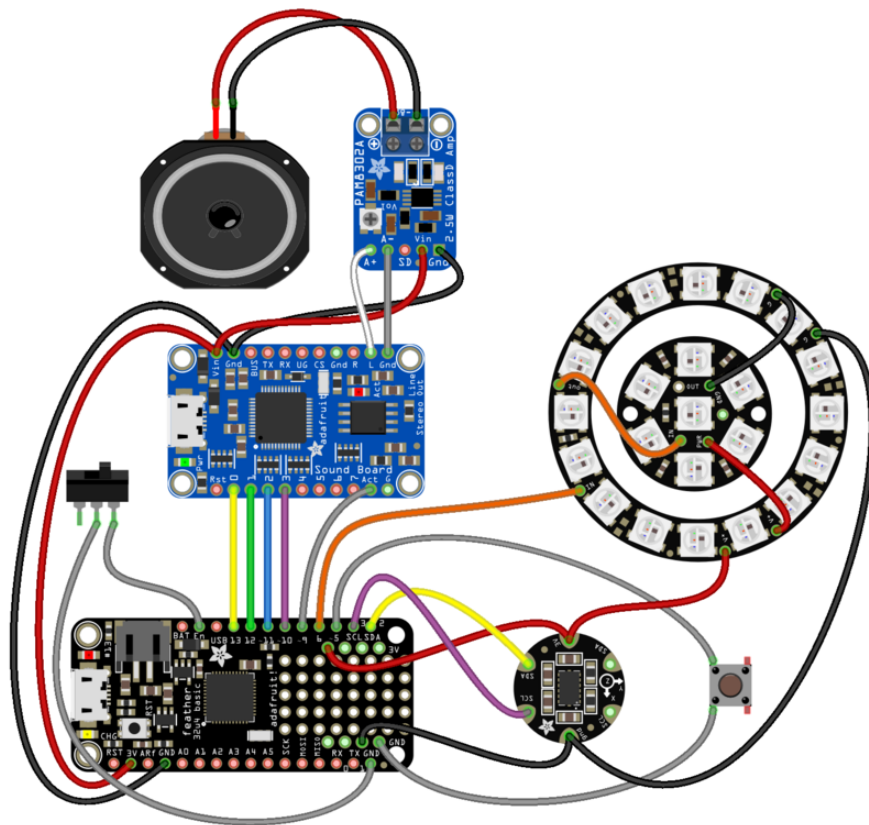


Lastly, cover the parts with a protective spray to lock in that glossy look you are going for.



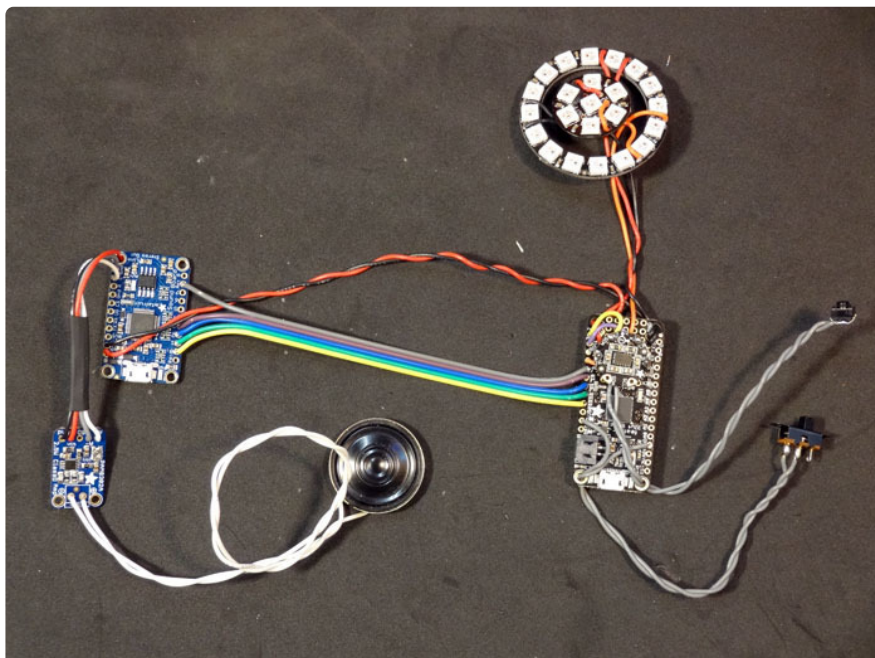
Electronics

Here is a diagram of the wiring used for the gauntlet project.



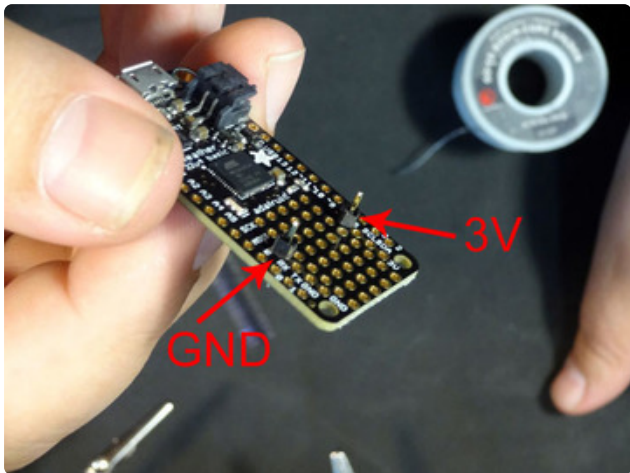
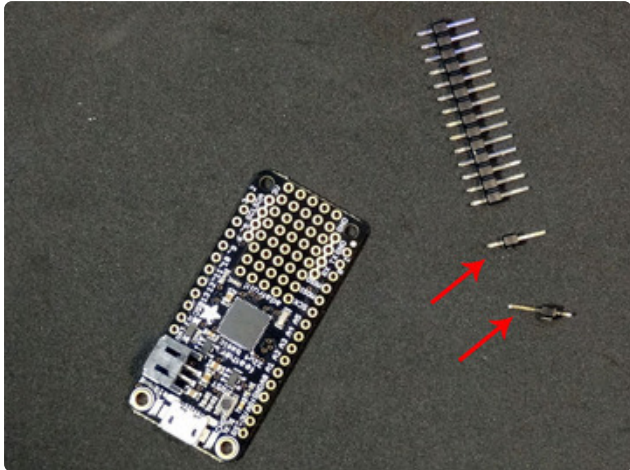
fritzing

I did record a video of this electronics build while taking the build photos. This may help you if the photos and text info is not enough, though it is an hour and a half long.

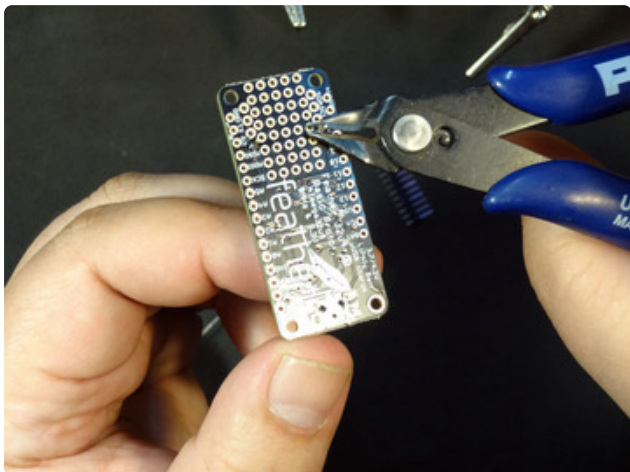


The following pages will go over the electronics build step by step.

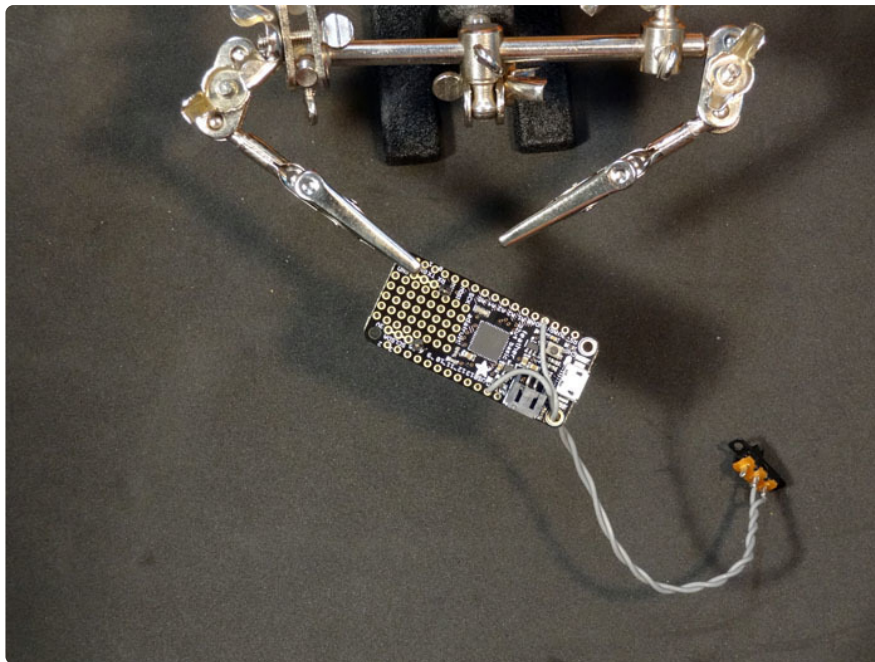
First Steps



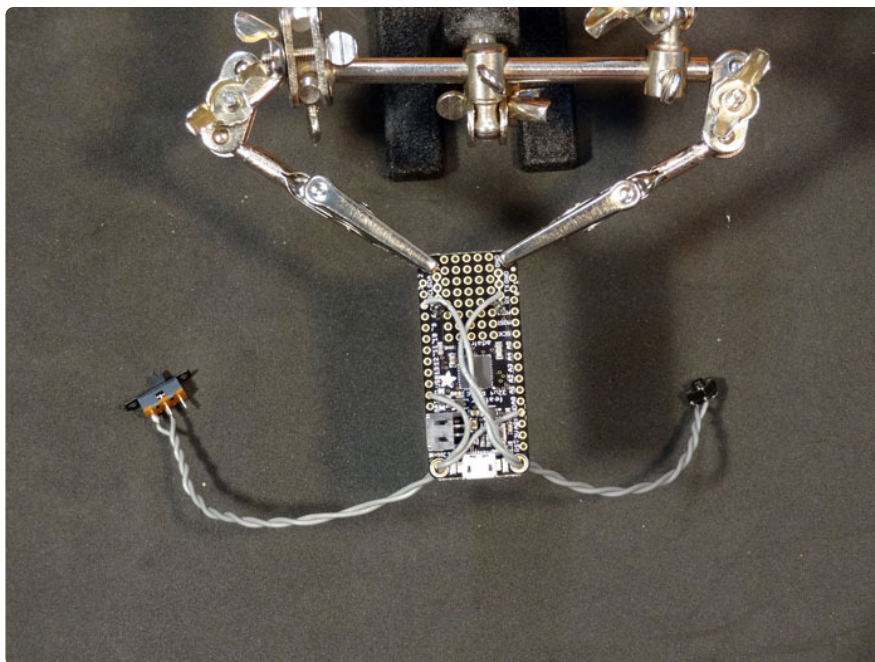
First we will solder two header pins to the Feather board. One gets attached to 3V and the other to GND. This will help to mount the Accelerometer later.



Now add a switch with one end connected to the **En** and the other end to **GND**. This will let you turn the glove off but still charge the battery.

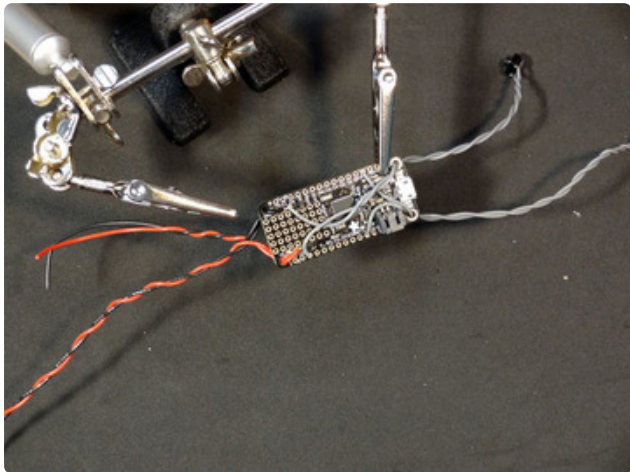
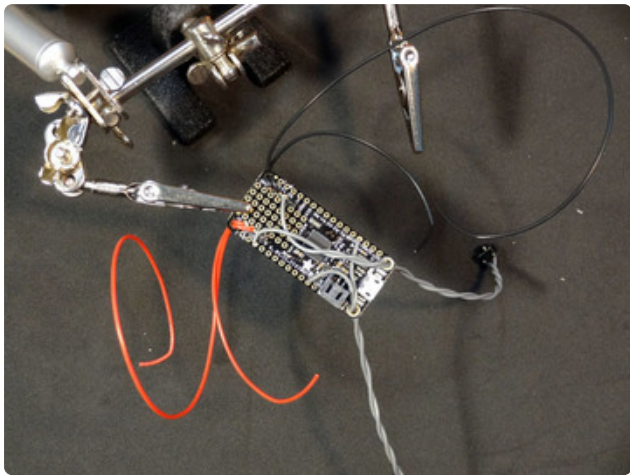
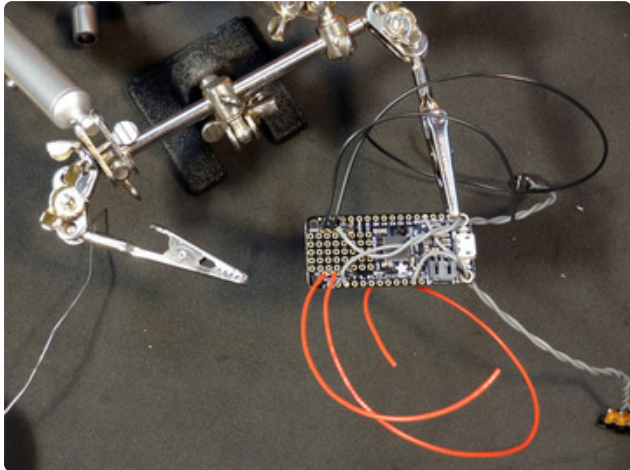


Connect a button from **Pin 5** to **GND**. This will act as a mode button.



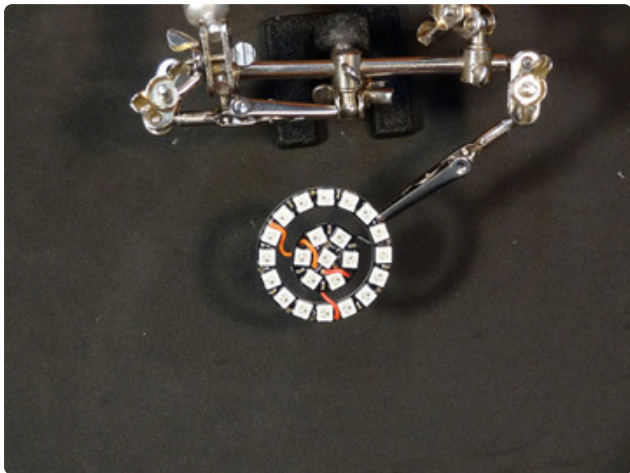
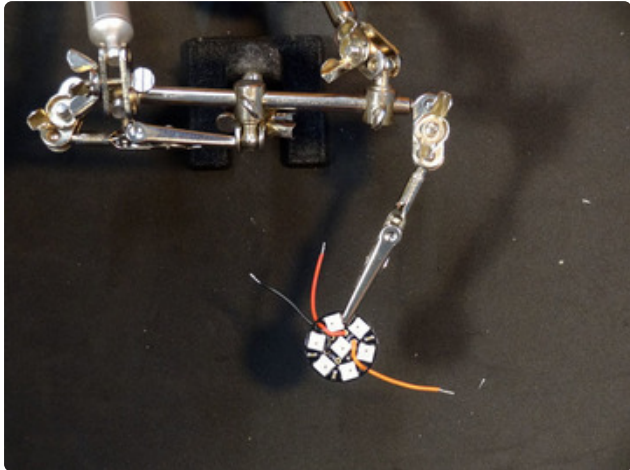
I like to feed some of this wires through the mounting holes to take the stress off of the solder joints.

Power



Install two power and ground wires onto the **3V** and **GND** parts of the proto board. One set of wires will give power to the FX Sound Board. The other set will power the NeoPixels.

NeoPixels

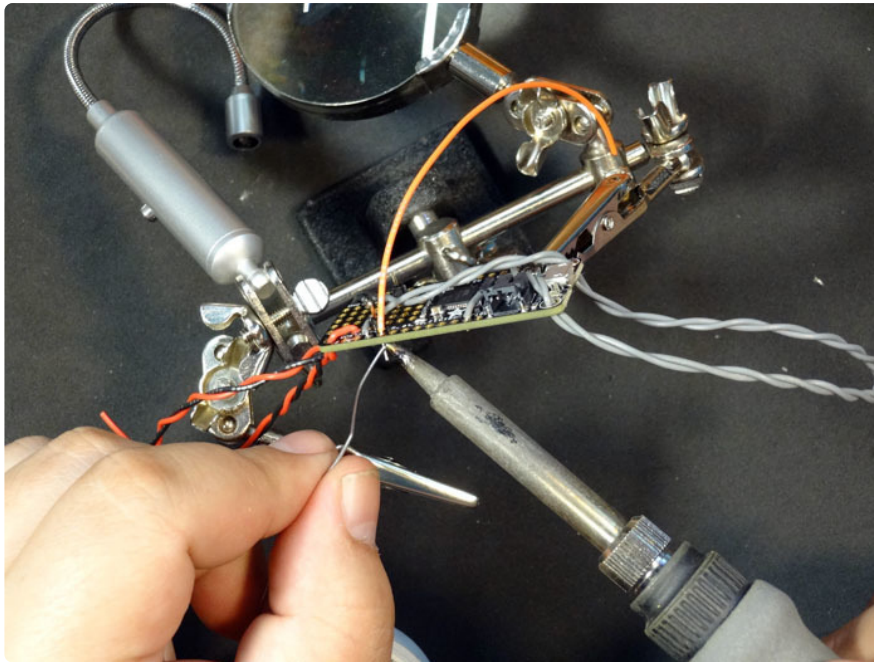


Solder short wires to **V+**, **G**, and **In** on the NeoPixel Jewel.

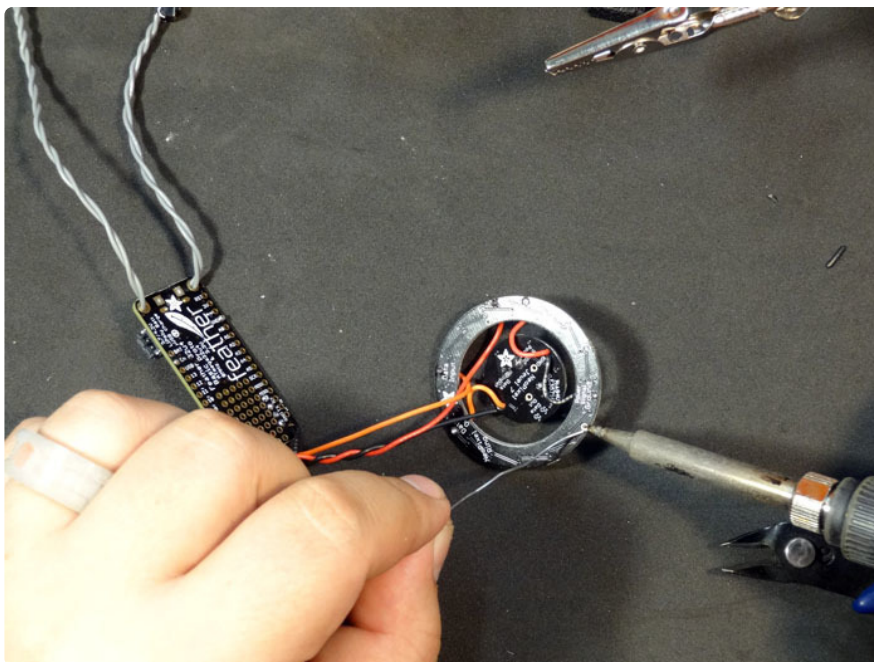
Then connect the Jewel wires to **V+**, **G**, and **Out** on the NeoPixel ring.



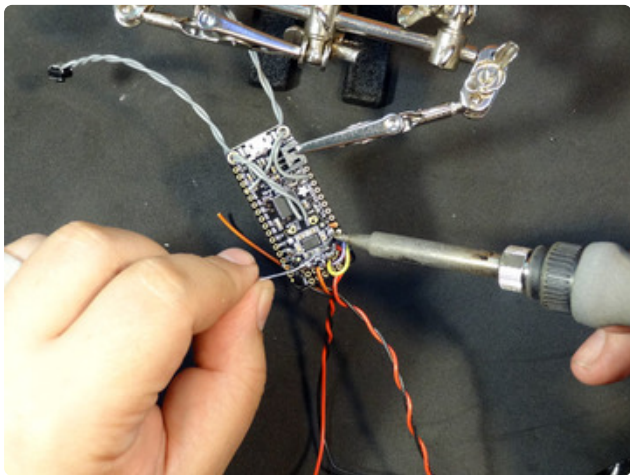
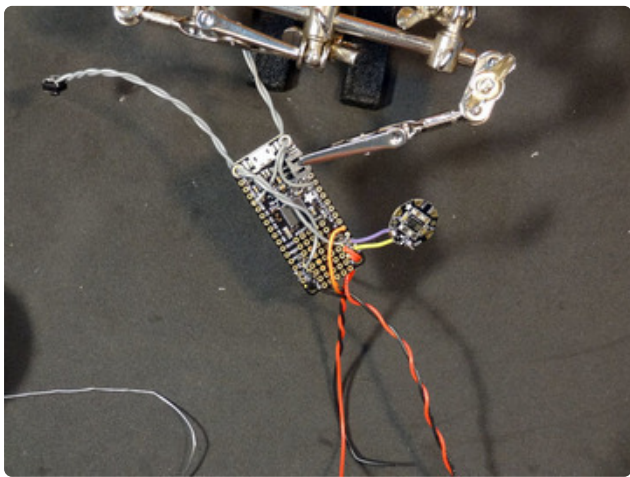
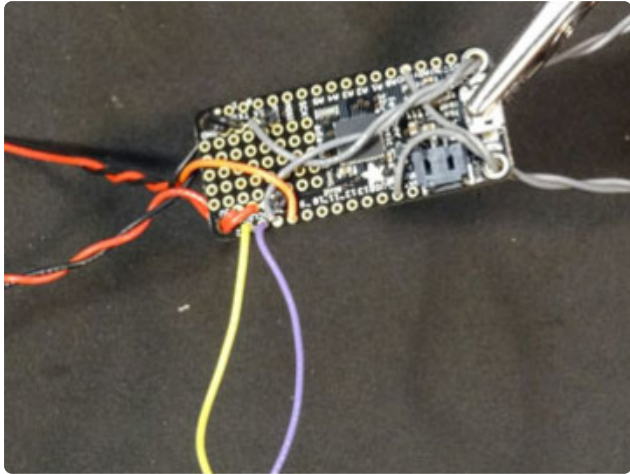
Solder **Pin 6** to the **In** pin of the NeoPixel ring.



Then solder the **3V** and **GND** wires to the ring as well.



Accelerometer



Solder Wires to **Pin 3** and **Pin 2** on the Feather Board.

Connect the **Pin 3** wire to **SCL** and the **Pin 2** wire to **SDA** on the Accelerometer.

Remember those two header pins we put on a while ago? Now let's solder the **3V** and **GND** pins from the Accelerometer onto the header pins on the **3V** and **GND** pins of the Feather Board.

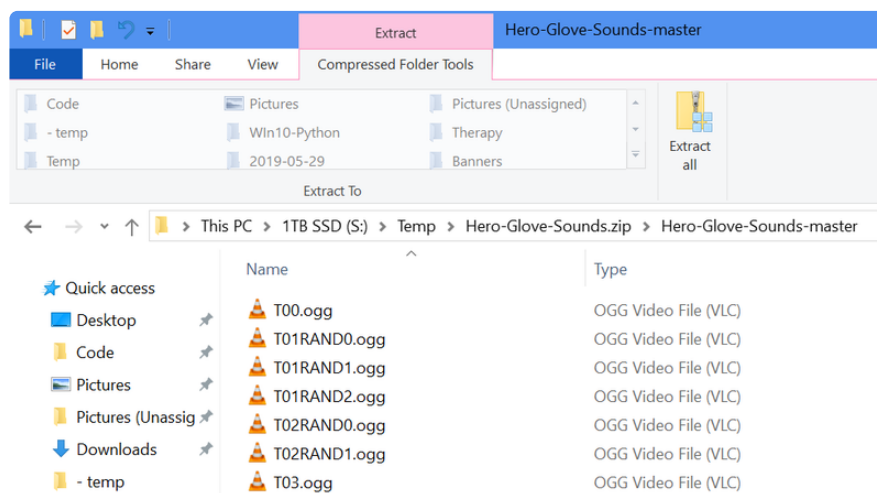
Sound FX Board

Getting the Sound Files

Click the green button below to download the sound files used for this project. In the folder you download the file to, use your file explorer/finder to open the zip file and extract the 7 sound files which all have the .ogg file extension.

Hero-Glove-Sounds.zip

<https://adafru.it/EYG>



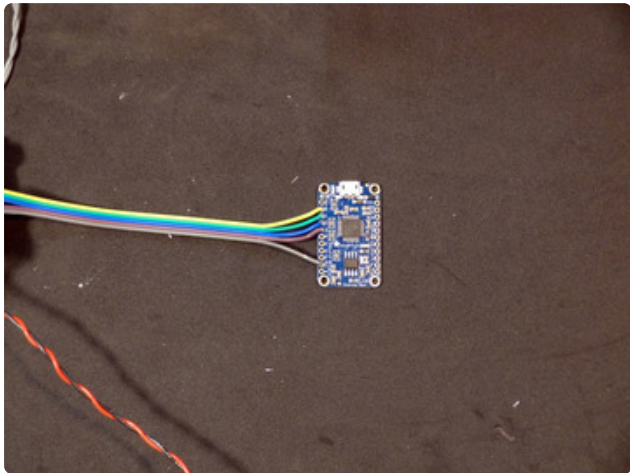
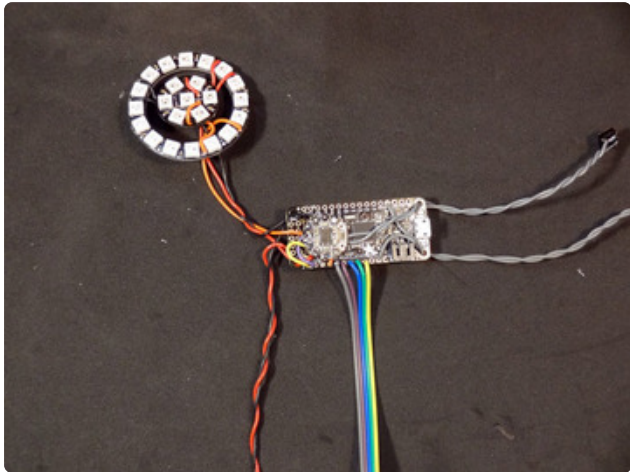
Loading the Sounds

Plug the FX sound board into your computer via a known, good data USB cable. Your computer should indicate a new removable drive was attached. Using your file explorer/finder, copy the 7 sound files from your downloads area onto the removable drive.

[See this guide \(https://adafru.it/pqc\)](https://adafru.it/pqc) for details on copying over the sound files.

Eject the drive and unplug from USB. Now you can complete the wiring.

Wiring the Board



Connect the Feather Board to the FX Sound Board.

Pin 13 -> 0

Pin 12 -> 1

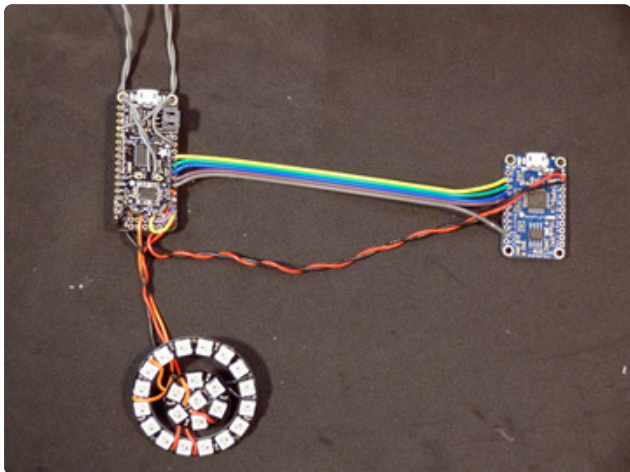
Pin 11 -> 2

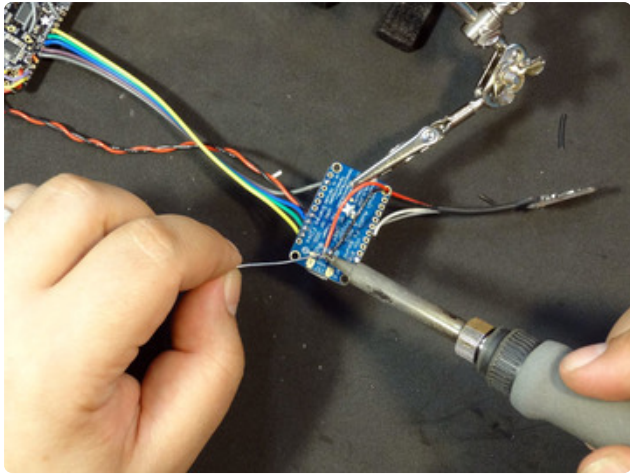
Pin 10 -> 3

Pin 9 -> Act

GND -> Gnd

3V -> Vin





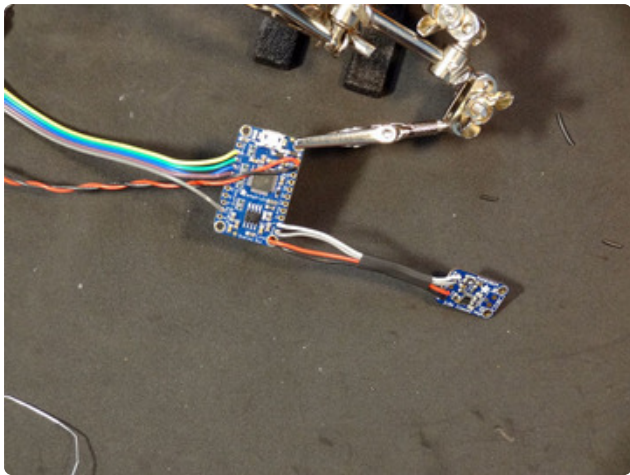
Connect the Amplifier to the audio output of the FX Sound Board.

L -> A+

Gnd -> A-

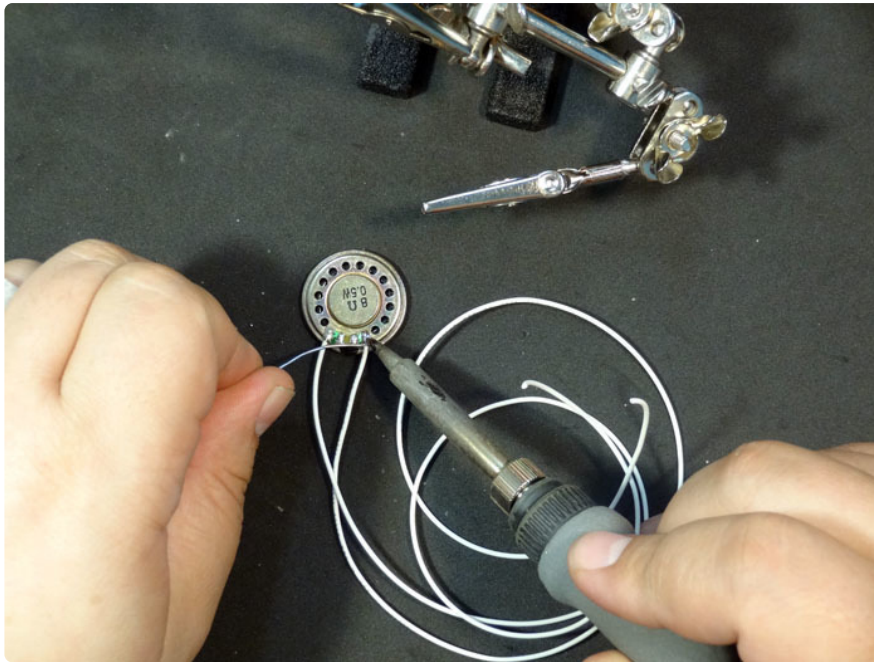
Vin -> Vin

Gnd -> Gnd

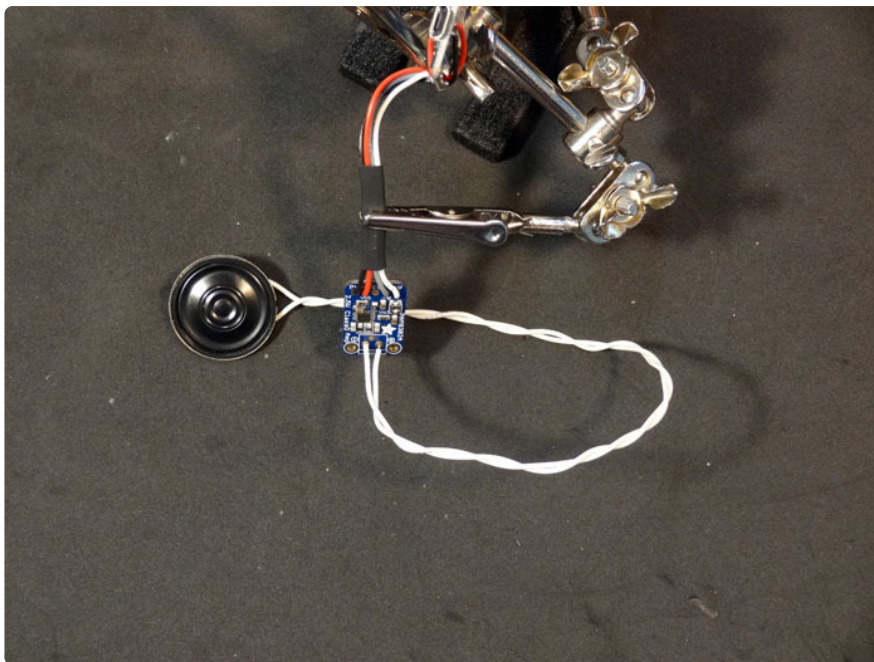


Connect the Amplifier to the audio output of the FX Sound Board.

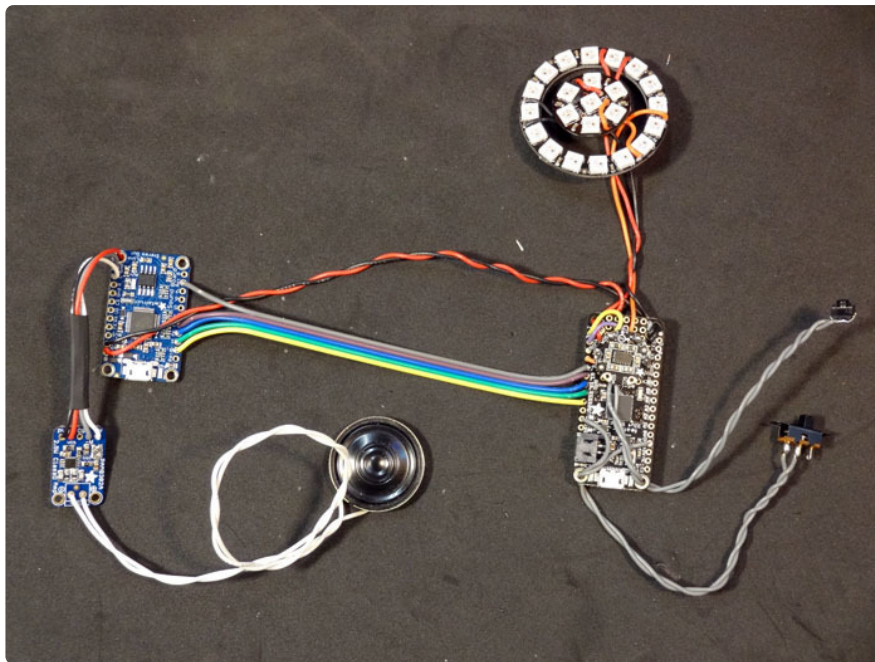
- **L -> A+**
- **Gnd -> A-**
- **Vin -> Vin**
- **Gnd -> Gnd**



Connect the speaker wires to the + and - sides of the Amplifier.



This is what the final electronics looks like. Load in the code that runs it all on the next page.



Programming

If you have never used the Adafruit Feather with the Arduino IDE, there are a few things that you will need to do before you can program the hardware.

Install the latest version of the IDE software using the following guide:

**Install and Setup the Arduino IDE
with Adafruit AVR Board support**

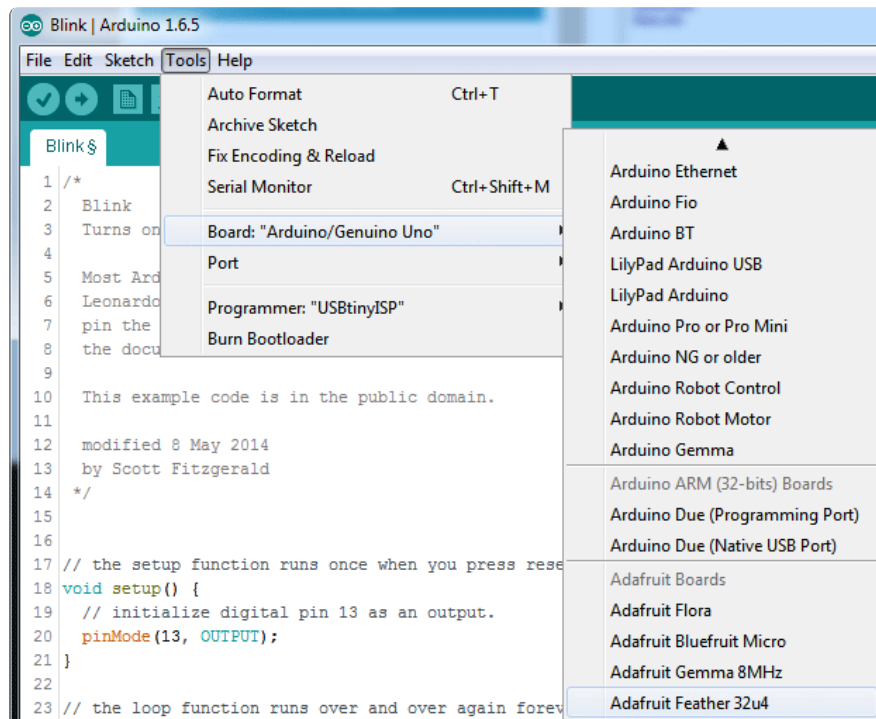
<https://adafru.it/DU1>

Add support for NeoPixel, Adafruit Sensors and LSM303 Libraries.

1. From the Arduino IDE, click the **Sketch** menu.
2. Move to **Include Library**.
3. Click on **Manage Libraries...**
4. When the new window opens, search for **NeoPixel**
5. Click the listing for **Adafruit NeoPixel**, then click Install or Update.
6. Once finished, search for **Adafruit LSM303**.
7. Click and install the listing for **Adafruit LSM303DLHC**.
8. Now search for **Adafruit Unified Sensor**.
9. Scroll to the listing for **Adafruit Unified Sensor**, this may be at the bottom of the listings, and install.

Now you should be ready to download the code.

Be sure that you have the **Adafruit Feather 32u4** board selected under the **Tools** menu.



Copy and paste the code below into a new Sketch or download the code.

```
// SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
//
// SPDX-License-Identifier: MIT

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_LSM303_Accel.h>
#include <Adafruit_NeoPixel.h>

#define LEDpin 6

#define SFX_start 13
#define SFX_PowerUp 12
#define SFX_Fire 11
#define SFX_PowerDown 10

#define SFX_Playing 9

#define button 5

Adafruit_NeoPixel strip = Adafruit_NeoPixel(19, LEDpin, NEO_GRB + NEO_KHZ800);

/* Assign a unique ID to this sensor at the same time */
Adafruit_LSM303_Accel_Unified accel = Adafruit_LSM303_Accel_Unified(54321);

// #####Light Flicker
int alpha; // Current value of the pixels
int dir = 1; // Direction of the pixels... 1 = getting brighter, 0 = getting dimmer
int flip; // Randomly flip the direction every once in a while
int minAlpha = 50; // Min value of brightness
int maxAlpha = 75; // Max value of brightness
int alphaDelta = 1; // Delta of brightness between times through the loop
```

```

int Coffset;

void setup()
{
#ifdef ESP8266
  // while (!Serial);      // will pause Zero, Leonardo, etc until serial console
  opens
#endif
  Serial.begin(9600);
  Serial.println("Accelerometer Test"); Serial.println("");

  /* Initialise the sensor */
  if(!accel.begin())
  {
    /* There was a problem detecting the ADXL345 ... check your connections */
    Serial.println("Ooops, no LSM303 detected ... Check your wiring!");
    while(1);
  }

  /* Display some basic information on this sensor */
  displaySensorDetails();

  pinMode(SFX_Playing, INPUT_PULLUP);
  pinMode(button, INPUT_PULLUP);

  pinMode(SFX_start, OUTPUT);
  pinMode(SFX_PowerUp, OUTPUT);
  pinMode(SFX_Fire, OUTPUT);
  pinMode(SFX_PowerDown, OUTPUT);
  delay(500);
  digitalWrite(SFX_start, HIGH);
  digitalWrite(SFX_PowerUp, HIGH);
  digitalWrite(SFX_Fire, HIGH);
  digitalWrite(SFX_PowerDown, HIGH);

  strip.begin();
  strip.show(); // Initialize all pixels to 'off'

  strip.setBrightness(255);
}

int ReadX;
int PowerUp = 0;
int DotVal = 10;

int ButtonPress = 1;

void loop()
{
  while (!digitalRead(button)){
    if(ButtonPress){
      ButtonPress = 0;
      digitalWrite(SFX_PowerDown, LOW);
      while(digitalRead(SFX_Playing));
      digitalWrite(SFX_PowerDown, HIGH);
      while(!digitalRead(SFX_Playing));
    }else{
      ButtonPress = 1;
      digitalWrite(SFX_start, LOW);
      while(digitalRead(SFX_Playing));
      digitalWrite(SFX_start, HIGH);
      while(!digitalRead(SFX_Playing));
    }
  }
  flip = random(32);
  if(flip > 20) {

```



```

    dir = 1 - dir;
}
// Some example procedures showing how to display to the pixels:
if (dir == 1) {
    alpha += alphaDelta;
}
if (dir == 0) {
    alpha -= alphaDelta;
}
if (alpha < minAlpha) {
    alpha = minAlpha;
    dir = 1;
}
if (alpha > maxAlpha) {
    alpha = maxAlpha;
    dir = 0;
}
//DotVal = alpha/8

Coffset = random(10);
colorWipe(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
colorWipeDot(strip.Color(10, 10, alpha/2));

sensors_event_t event;
accel.getEvent(&event);
Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print(" ");
Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print(" ");
Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print("
");Serial.println("m/s^2 ");
ReadX=event.acceleration.y;
if(ReadX < 10.5 && ReadX > 8 && ButtonPress){
    PowerUp = 1;
    digitalWrite(SFX_PowerUp, LOW);
    while(digitalRead(SFX_Playing));
    digitalWrite(SFX_PowerUp, HIGH);
    while(!digitalRead(SFX_Playing)){
        alpha = alpha+1;
        if(alpha<254){
            colorWipe(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
            colorWipeDot(strip.Color(alpha-Coffset/2, alpha-Coffset/2, alpha/2));
        }
    }
    sensors_event_t event;
    accel.getEvent(&event);
    Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print(" ");
    Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print(" ");
    Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print("
");Serial.println("m/s^2 ");
    ReadX=event.acceleration.y;
    delay(50);
    if(ReadX < 10.5 && ReadX > 8 ){
        digitalWrite(SFX_Fire, LOW);
        while(digitalRead(SFX_Playing)){
            alpha = alpha-8;
            if(alpha>10){
                colorWipe(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
                colorWipeDot(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
            }
        }
        digitalWrite(SFX_Fire, HIGH);
        alpha = 255;
        while(!digitalRead(SFX_Playing)){
            alpha = alpha - 2;
            if(alpha>10){
                colorWipe(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
                colorWipeDot(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
            }
        }
    }
}

```

```

        colorWipeDot(strip.Color(alpha-Coffset/2, alpha-Coffset/2, alpha/2));
        delay(1000);
    }else{
        digitalWrite(SFX_PowerDown, LOW);
        while(digitalRead(SFX_Playing));
        digitalWrite(SFX_PowerDown, HIGH);
        while(!digitalRead(SFX_Playing)){
            alpha = alpha - 1;
            DotVal = alpha - 8;
            if(alpha>10){
                colorWipe(strip.Color(alpha-Coffset, alpha-Coffset, alpha));
                colorWipeDot(strip.Color(DotVal-Coffset, DotVal-Coffset, DotVal));
            }
        }

        delay(300);

    }
}
//digitalWrite(SFX_PowerUp, HIGH);

//delay(50);
}

// Fill the dots one after the other with a color
void colorWipe(uint32_t c) {
    for(uint16_t i=0; i<16; i++) {
        strip.setPixelColor(i, c);
        strip.show();
    }
}

// Fill the dots one after the other with a color
void colorWipeDot(uint32_t c) {
    for(uint16_t i=16; i<19; i++) {
        strip.setPixelColor(i, c);
        strip.show();
    }
}

void displaySensorDetails()
{
    sensor_t sensor;
    accel.getSensor(&sensor);
    Serial.println("-----");
    Serial.print ("Sensor:      "); Serial.println(sensor.name);
    Serial.print ("Driver Ver:  "); Serial.println(sensor.version);
    Serial.print ("Unique ID:   "); Serial.println(sensor.sensor_id);
    Serial.print ("Max Value:   "); Serial.print(sensor.max_value);
    Serial.println(" m/s^2");
    Serial.print ("Min Value:   "); Serial.print(sensor.min_value);
    Serial.println(" m/s^2");
    Serial.print ("Resolution:  "); Serial.print(sensor.resolution);
    Serial.println(" m/s^2");
    Serial.println("-----");
    Serial.println("");
    delay(500);
}

```

Click here for more help with
uploading to the Feather
microcontroller

Using the Gauntlet

Powering up the glove

The slide switch will connect the battery power to the electronics and play the "Importing preferences ..." sound effect. The lights should flicker randomly.

Using the blast

You will want to hold your palm up at about a 90° angle and hold it when you start to hear the power up sound. If you keep holding your palm in that position, the blast will happen after the power up is finished. Sometimes this will be short and sometimes long, the power up sounds are random to mix things up.

If you move your palm out of position before the power up is finished, the glove will play the power down sequence rather than firing.

Disabling the motion control

As cool as it is to blast people with the glove, things can get awkward when raising your hand at a panel or giving someone a high five. For this reason you can press the small button to disable the sensor so that the palm just flickers and will not shoot. Press the button again and it will rearm the sensor and let you shoot the blast as usual.

Charging the battery

The battery should last you for most of the day, but it will need to be recharged. Thankfully the Adafruit Feather has a built in charger via the same USB port that you used to program it with. Simply connect it to any 5V power outlet and charge just like your cellphone.