ReBoots Animated LED Boot Laces

Created by Erin St Blaine

https://learn.adafruit.com/re-boots-animated-dancing-boot-laces

Last updated on 2023-08-29 02:45:32 PM EDT
# Table of Contents

Overview                                3
Wiring                                   4
LED Assembly                             5
Sensor Assembly                          11
Trinket Assembly                         14
The Code
  • If you encounter trouble...          15
Finishing                                19
Overview

Glowing, animated boot laces. A sensor in each heel, watching you, weighing your every step, waiting for the opportune moment to light your stride and set your dance moves on fire. NeoPixels are glued to light pipe laces and programmed to animate and glow.

Zip into the ReBoots and become the cyborg you've always wanted to be.

2 x Pro Trinket
Pro Trinket 5v

2 x **Battery Connector**  
JST Battery Connector  
https://www.adafruit.com/product/1769

2 x **Neopixels**  
Individual Neopixels  
https://www.adafruit.com/product/1559

2 x **Battery**  
500 mAh Lithium Polymer Battery  
https://www.adafruit.com/product/1578

2 x **FSR**  
Force Sensitive Resistor  
https://www.adafruit.com/product/166

---

**Other Materials**

- Killer Boots with Laces - I used [Demonia's Gothika600UV in Silver](https://www.adafruit.com/product/1559). This project relies on the laces being mostly a decorative feature, not load-bearing...the boots are actually adjusted and closed with buckles and zippers.
- LED Shoelaces or Light Pipe - [I used 3 sets of these shoelaces from Flammi](https://www.adafruit.com/product/1578).
- 30awg silicone wire

---

**Tools Needed**

- Soldering iron & accessories
- 3d Printer (optional)
- Hot glue gun
- Lighter
- Stapler
- Spray glue
- Needle & strong thread (a thimble helps too)

---

**Wiring**

![Wiring Diagram]
**Wiring**

<table>
<thead>
<tr>
<th>Pro Trinket</th>
<th>NeoPixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAT+</td>
<td>+</td>
</tr>
<tr>
<td>G (ground)</td>
<td>–</td>
</tr>
<tr>
<td>Pin 3</td>
<td>Data In</td>
</tr>
</tbody>
</table>

The force-sensing resistor (FSR) connects to Pro Trinket pins A5 and GND (it can connect in either direction).

Easy peezy.

## LED Assembly

**3D Printed Cases**

These little 3d printed cases snap onto your NeoPixels and hold the light pipe shoelaces firmly in place against the LED for maximum brightness.

Print them in white or transparent ABS or PLA.

Every 3D printer is a little different. Before committing to a whole pile of these, start with one and test the fit, adjusting scale as necessary.

Download from Thingiverse
If you don’t have access to a 3D printer, don’t worry! The light pipes and pixels could also be joined with caps of hot glue, Shapelock (aka Instamorph or Friendly Plastic), Sugru or Apoxie Sculpt.

Power and Ground Wires

Grab a tape measure and determine the distance between your shoelace holes.

Double the shoelace-hole-distance measurement. Lay out four 30awg wires (2 for each boot) and make one mark for each NeoPixel's placement. I marked every 2" since my shoelace holes are 1" apart.

Go outside. Grab a lighter with a hot, focused flame and char the shielding at each mark. Let it cool off a bit.

Scrape off the ash and you'll have a nice bit of stripped wire at each mark.
String all the NeoPixels onto your wires, being sure they're all aligned the same way -- all the "+" holes should go along one wire and all the "-" holes along the other.

Slide one NeoPixel to each bare wire spot and twist it around 2-3 times to secure the bare wire to the solder pad. Once you have them all in place, solder each wire to each pad.

Data Wires

Connect all the "out" pads to the "in" pad of the next pixel, making sure your NeoPixel spacing matches up with your lace hole spacing.
Since my boot laces were exactly 1" apart, I cheated a little and simply stapled the data line holes together. If yours are a different distance you'll need to fiddle with stripping and soldering wires to get them to end up the right distance, but for 1" spacing, staples work brilliantly.

However, for the connections that go across the ankle this will NOT work. Any flexing of the strip will eventually break the staples, so only staple the NeoPixels that will be on the shin or the top of the foot. Use flexible stranded wire to connect the pixels that are anywhere near the ankle.

Solder the staples and stranded wires into place.

This is a great time to hook up your pixels with some alligator clips and the NeoPixel Strandtest code, and test to be sure all the pixels are working. I keep a Gemma pre-loaded with this code near my workbench for just this purpose.

Adding the Light Pipe

Get out your LED shoelaces and cut them down so they fit nicely between opposite shoelace holes in your boots. Remember they'll be a little longer for the top holes and shorter for the bottom holes, for most boots.
Pop your 3D printed 'cases' onto your NeoPixels, add a blob of hot glue right in the hole and shove the light pipe in while the glue is wet. When it dries it will hold the light pipe beautifully perpendicular to the LEDs. Hooray! If needed, add another blob of hot glue to the outside of the case for extra support.

Now you have a creepy glowing alien spinal column to play with. Take some artsy photos.

Once the glue is set, thread the light pipes through the shoelace holes, making sure the first NeoPixel in the sequence is at the top. I had to do some disassembly on my boot to get them between the boot facing and boot liner -- each pair of boots is different so get a little creative. Sew or glue them in place.

Bend the pipes across and use hot glue to secure the other end in the opposite lace hole.
Sensor Assembly

Adding the force sensitive resistor (FSR) into the heel of the boots will give you an opportunity to trigger animations, change modes, or just flash to the beat when you step down with each heel.
Hook it Up

Hook up two long 30awg wires to the pads on your force sensitive resistor.

It's possible to solder these wires to the pads securely, but it is NOT easy to do. Be quick and neat, or the plastic tabs may melt.

More connection options & info about FSRs()

For all that it's difficult, soldering is probably your best bet since this will go under your heel. Nobody loves a bulky connector stabbing them with every step.

Placement

Open up the liner inside the heel of your boot and find a good spot for the sensor. (If there is any metal in there, screws or whatnot, cover it up with electrical tape first)

Poke a little hole in the boot right at heel level and thread the wires through to the outside. This will keep them safe from pulling and tearing when you're dancing or putting the boots on. (A needle threader works great for getting small wires through a very small hole.)

Feed the wires up the side of the boot to the cuff area, where they will attach to the Pro Trinket.
Later you'll want to use some spray glue to secure the sensor and liner back in place, but first let's test and make sure it all works.
Trinket Assembly

Battery Connector
Solder a JST battery connector to the pads on the back of your Pro Trinket.

Hook it All Up

Decide where on your boots the Pro Trinket will live. You may want yours on the inside (for slickness) or the outside (for rocking your cyber style). Be sure the battery port and USB port are accessible, and be sure the reset button is safe from accidental squeezing when your boots are on.

Poke the wires from the NeoPixels and the sensor through the boots so they all happily meet at the Pro Trinket. Flex the boot a bit and make sure you have enough slack in the wires that they won’t break if you dance too hard.

Solder the wires on according to the wiring diagram. Take your needle and thread and stitch the Pro Trinket to the boot so it stays securely in place.
The Code

Software Setup

If this is your first time using Pro Trinket, take a look at [Introducing Pro Trinket](#) to get a guided tour.

Once you've got your Pro Trinket up and running with [Arduino](#), you'll need to install the FastLED library.

FastLED Library

You will also need to install the FastLED library in Arduino ([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 () 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.

```
#include &lt;FastLED.h&gt;
#define LED_PIN 3
#define NUM_LEDS 16 //change this to reflect the number of neopixels in your boots
#define SENSOR_LOW 35 //these numbers will change the sensitivity of the sensor. Lower numbers = harder step to trigger
#define SENSOR_HIGH 65

CRGB leds[NUM_LEDS];
CRGBPalette16 currentPalette;
TBlendType currentBlending;
uint8_t thishue = 0;
double deltahue = -3; //controls how quickly the rainbow fades from one color to another. Negative numbers make the rainbow // appear to crawl "down" and positive numbers make it crawl "up"

const int STEP_PIN = A5; // Analog input pin that the force sensitive resistor is attached to
int sensorValue = 0; // value read from the pot
int ledMode = 0;

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  pinMode(STEP_PIN, INPUT_PULLUP);
  FastLED.addLeds<WS2812B, LED_PIN, GRB>(leds, NUM_LEDS).setCorrection( TypicalLEDStrip );
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(STEP_PIN);
  // print the results to the serial monitor:
  Serial.print("sensor = ");
  Serial.println(sensorValue);
  switch (ledMode) {
  case 999: break;
  case 0: rainbow(); break; //mode that's running all the time
  case 1: fire(); break; //mode that runs when sensor is triggered
  }
```
if (sensorValue < SENSOR_LOW) {
    Serial.println("leds triggered");
    ledMode = 1;
}
if (sensorValue > SENSOR_HIGH) {
    Serial.println("leds not triggered");
    ledMode = 0;
}

//RAINBOW EFFECT

void rainbow() {
    byte thishue = thishue + 1;
    fill_rainbow(leds, NUM_LEDS, thishue, deltahue);
    FastLED.show();
    FastLED.delay(100); //change delay number to speed up or slow down the animation
}

//FIRE EFFECT by Mark Kriegsman

void fire() {
    currentPalette = LavaColors_p;
    Fire2012WithPalette(); // run simulation frame, using palette colors
    FastLED.show(); // display this frame
    FastLED.delay(100);
}
#define COOLING 55 // Less cooling = taller flames. Default 55, suggested range 20-100
#define SPARKING 120 //Higher chance = more roaring fire. Default 120, suggested range 50-200

void Fire2012WithPalette() {
    random16_add_entropy( random());
    static byte heat[NUM_LEDS];
    for( int i = 0; i < NUM_LEDS; i++) {
        heat[i] = qsub8( heat[i], random8(0, ((COOLING * 10) / NUM_LEDS) + 2));
    }
    for( int k = NUM_LEDS - 3; k > 0; k--) {
        heat[k] = (heat[k - 1] + heat[k - 2] + heat[k - 2]) / 3;
    }
    if( random8() < SPARKING ) {
        int y = random8(7);
        heat[y] = qadd8( heat[y], random8(160,255) );
    }
    for( int j = 0; j < NUM_LEDS; j++) {
        byte colorindex = scale8( heat[j], 240);
        leds[ (NUM_LEDS-1) - j ] = ColorFromPalette( currentPalette, colorindex); // upside down fire!! reverse it: leds[j] = ...
    }
}

Plug in your LiPo battery and the boots will come on in Rainbow mode. Step or press down on the sensor in the boot heel, and they will switch to Fire mode.
You may need to tweak the SENSOR_HIGH and SENSOR_LOW numbers until you have a satisfying rainbow-to-fire ratio.

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 Gemma pin.

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+ (. This is potent stuff, written by people with a deep appreciation for LED art, and we wanted to showcase it.

©Adafruit Industries
Once you're happy with your sensor placement and the code is all working, use some spray glue to glue down the sensor and the heel pad inside the boot.

Plug in your battery and tuck it someplace safe.

Try them on, stomp around, and set the place on fire!