Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

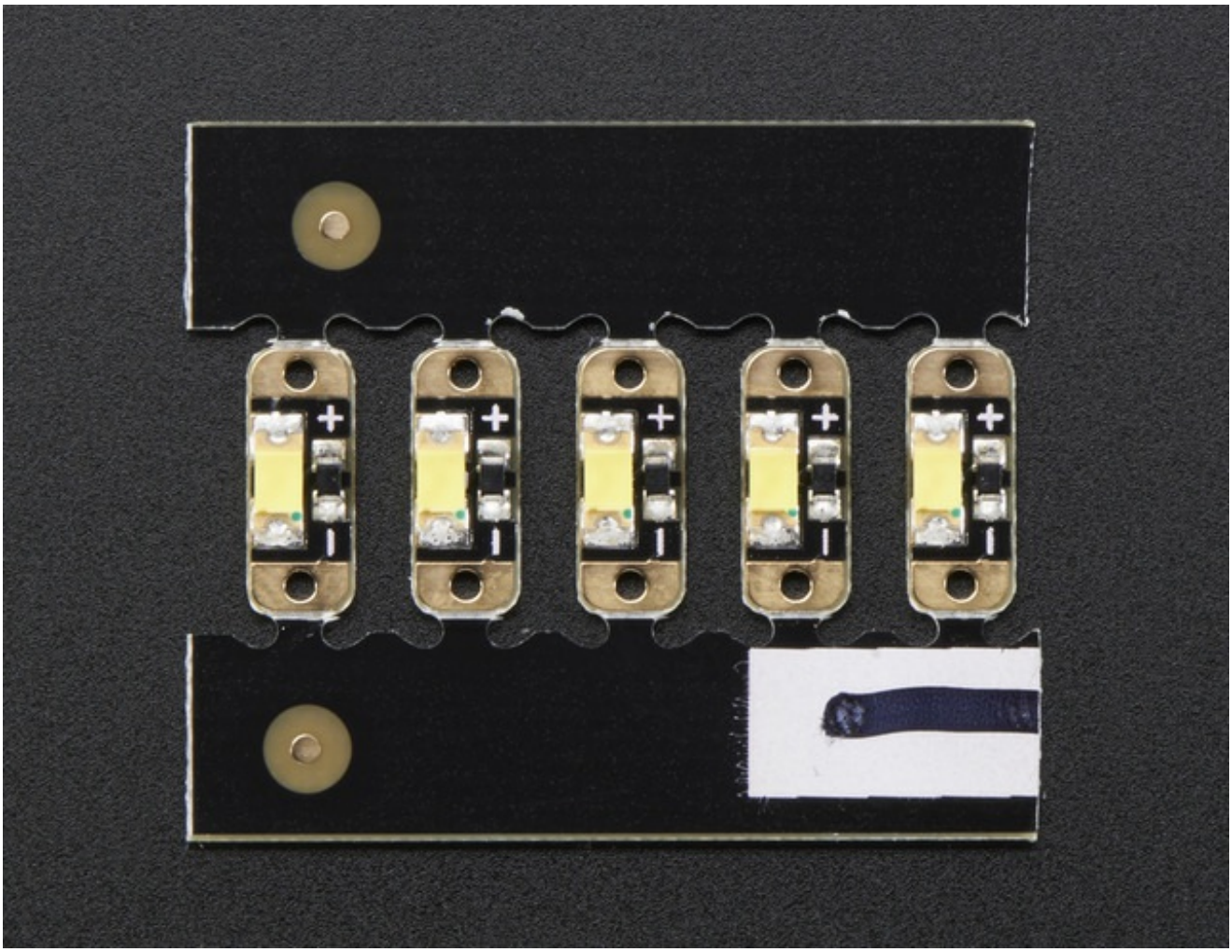
Adafruit LED Sequins

Created by Becky Stern

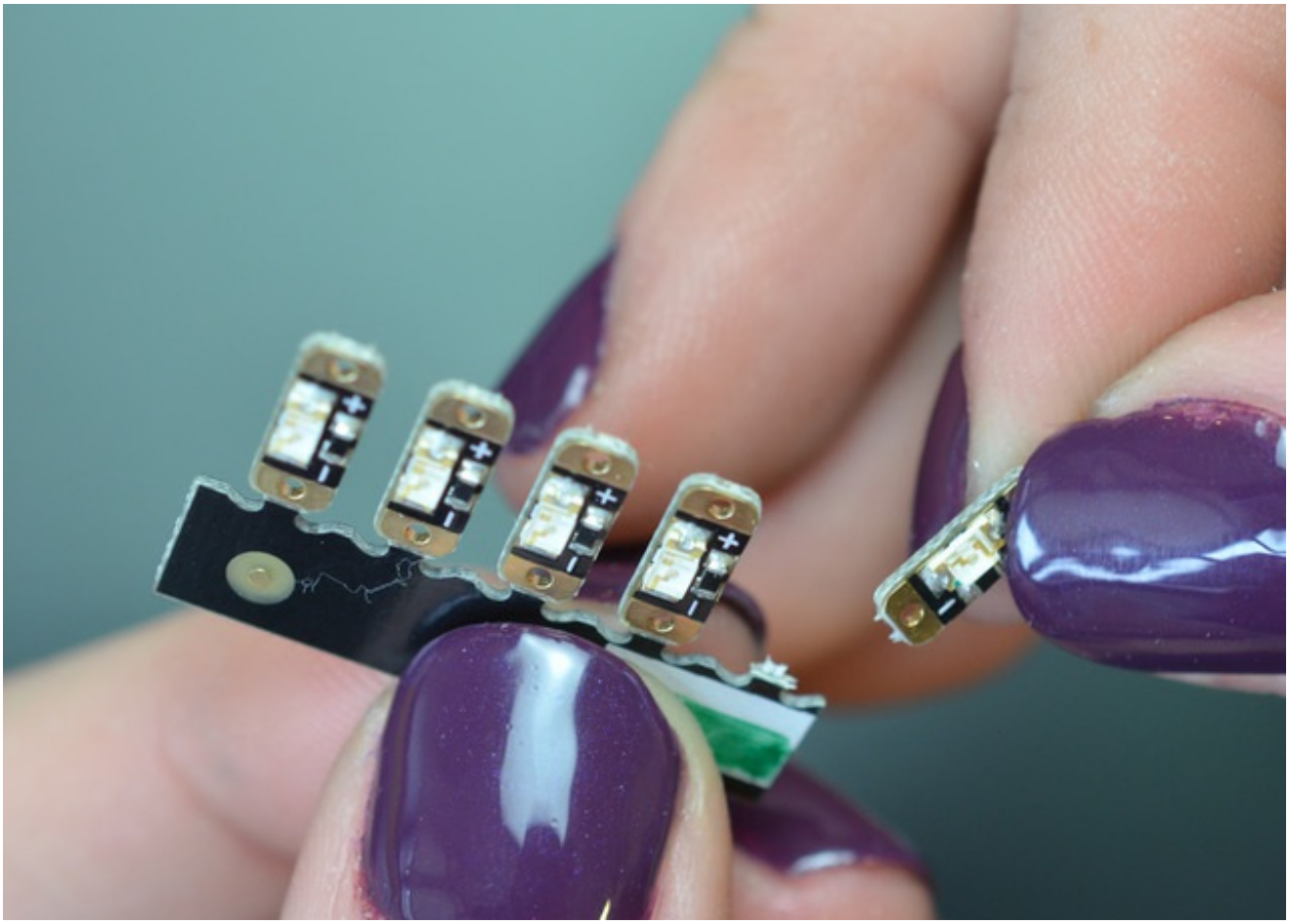
Last updated on 2016-09-12 06:59:51 PM UTC

Guide Contents

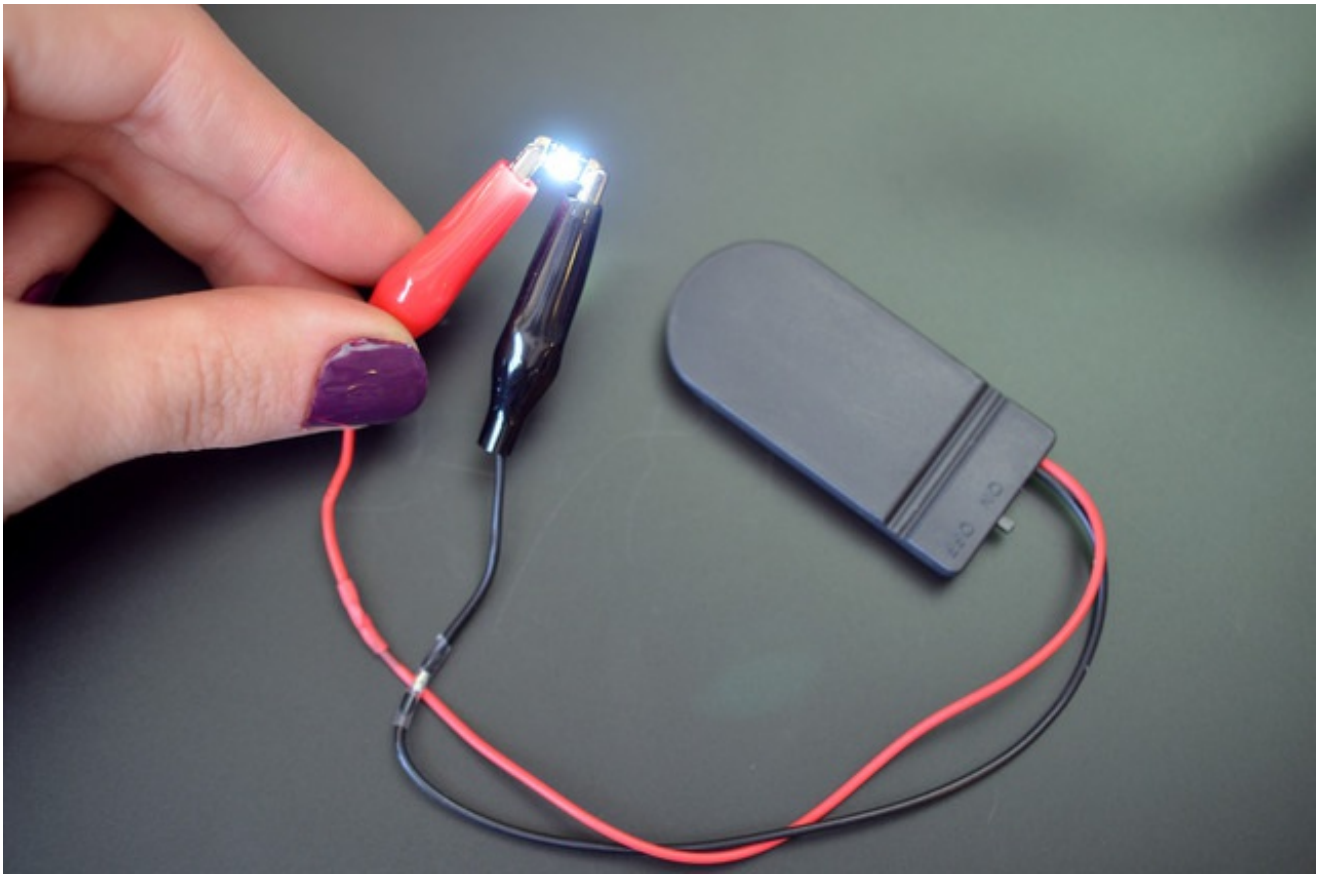
Overview



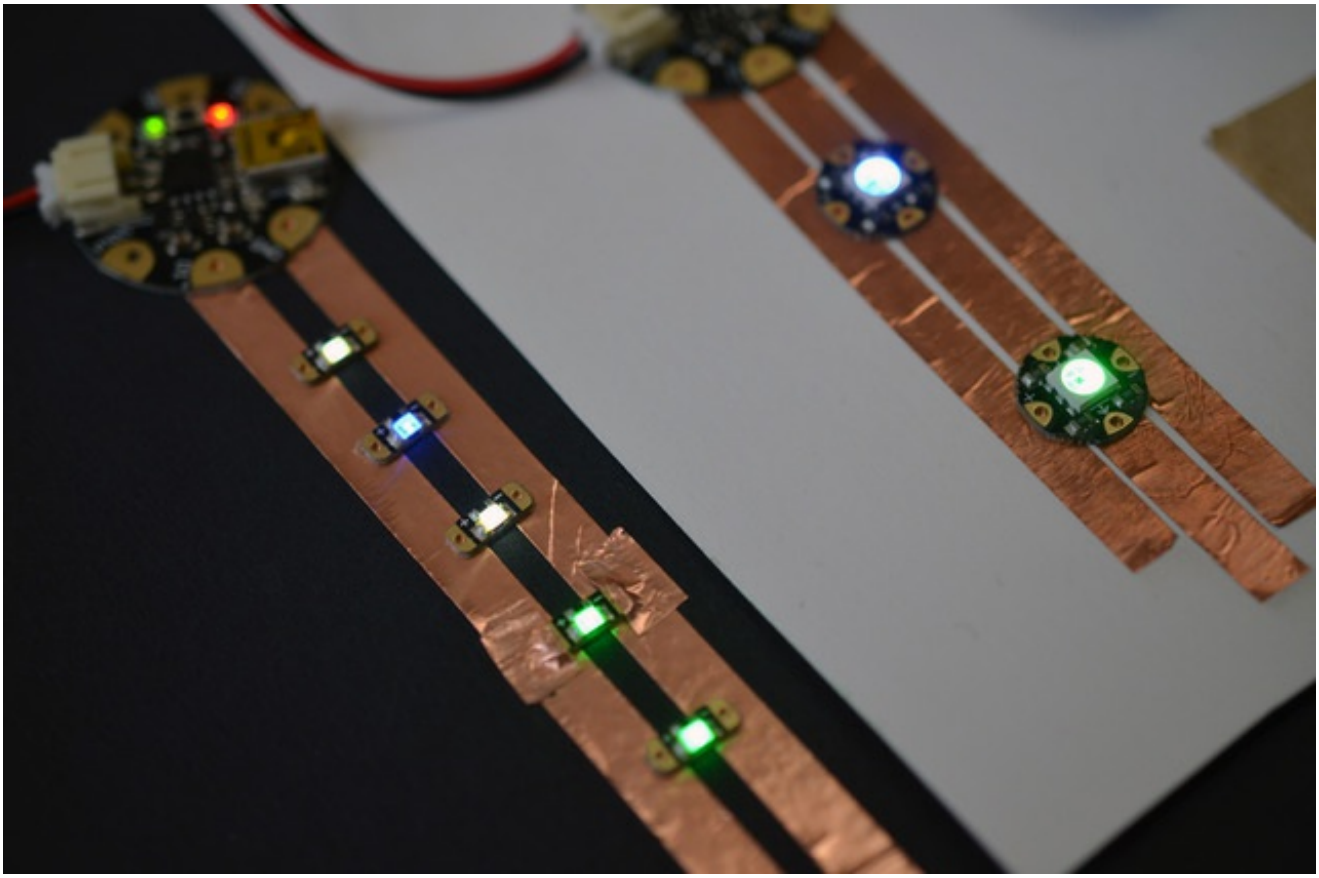
Sew a little sparkle into your wearable project with an Adafruit LED Sequin. These are the kid-sister to our popular Flora NeoPixel-- they only show a single color and they aren't addressable, but they are our smallest sewable LEDs ever and very easy to use!



They come in packs of five with breakaway tabs-- you can separate them one at a time as you build your project.



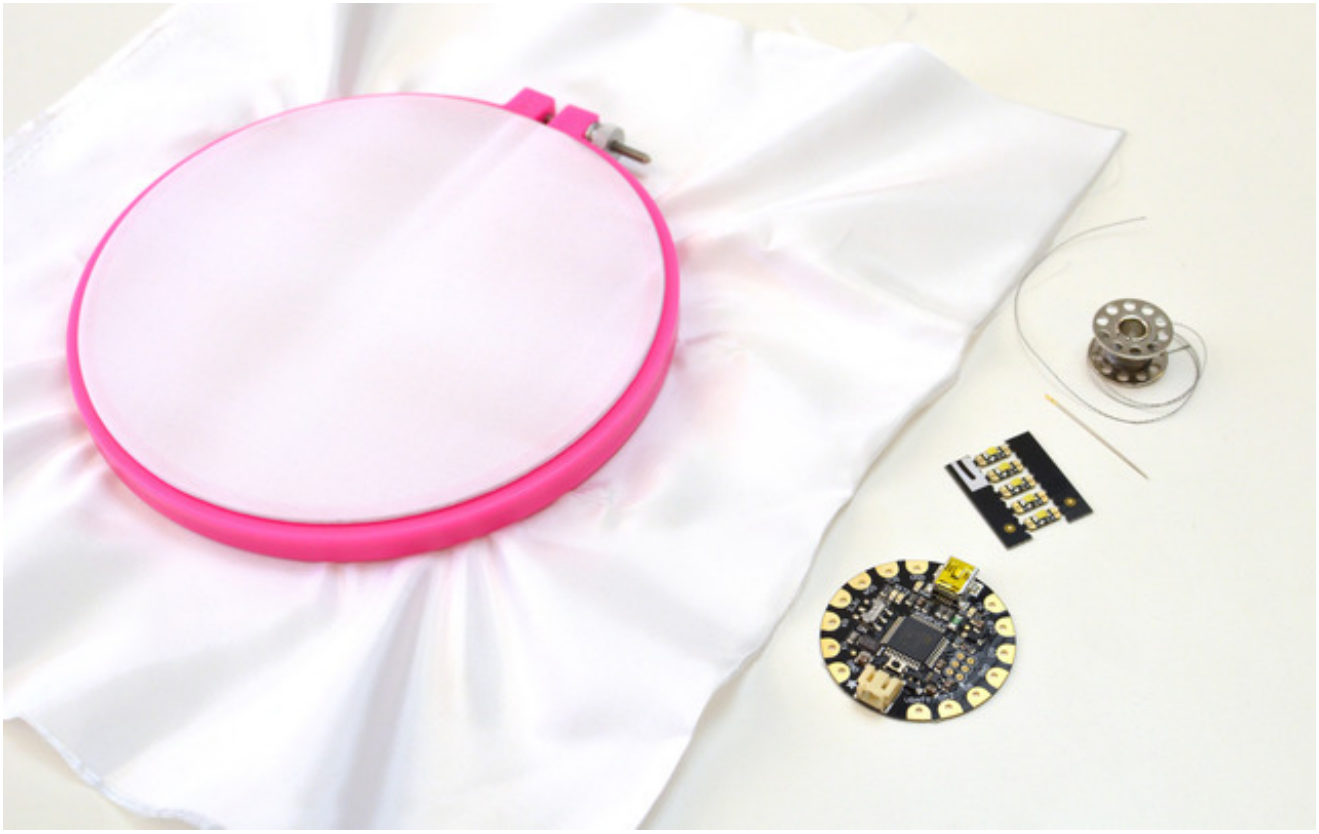
Simply connect 3 to 6VDC to the + pin and ground to the - pin, and the LED on the board will light up. In the photo above we've soldered alligator clips to a 2x2032 coincell battery holder (~6V). You could also use a single CR2032 sewable coincell holder-- you do not need a microcontroller to drive these sequins, unless you want them to blink or fade.



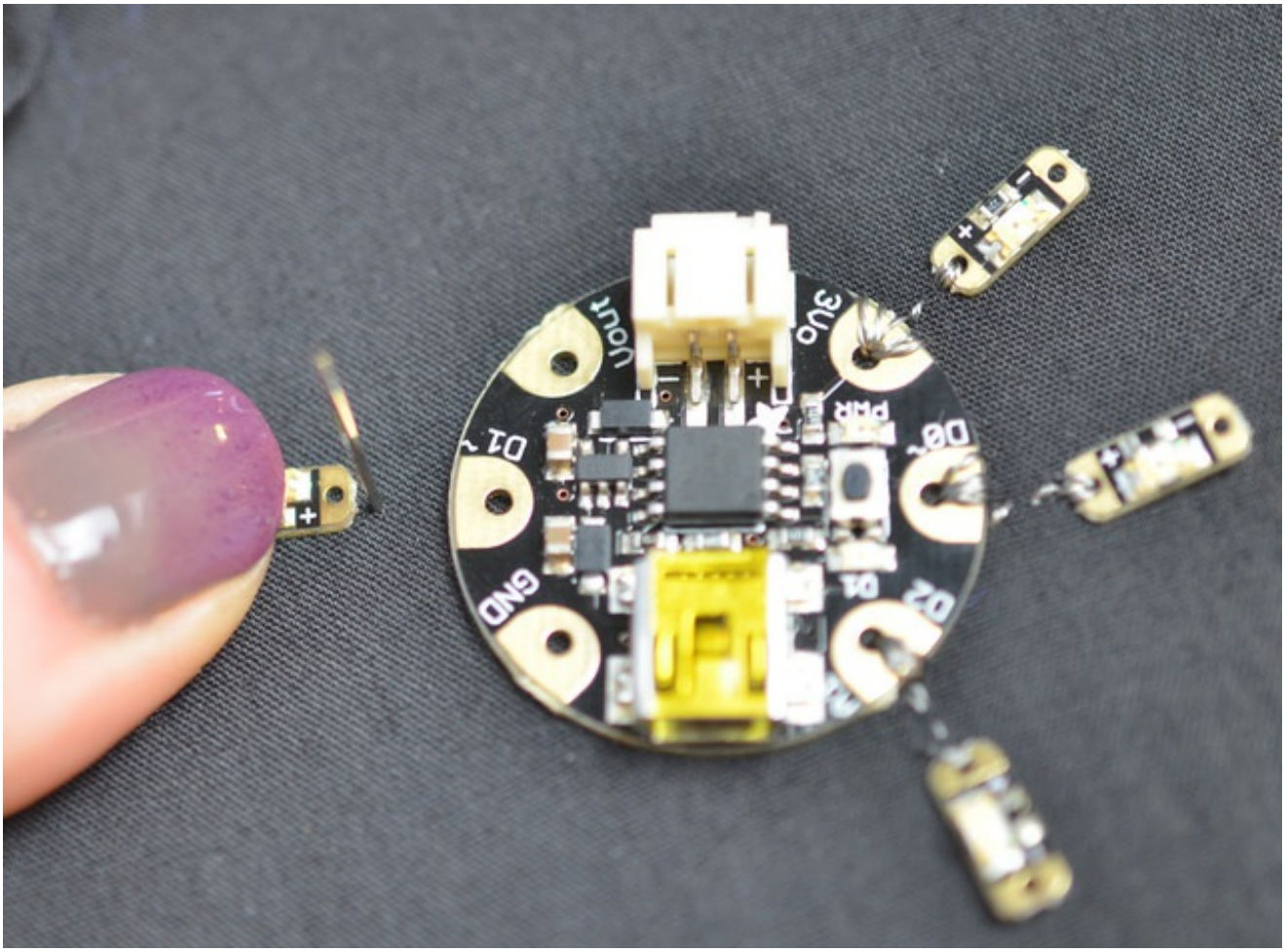
When powered from 3.3V they draw about 5mA so you can put up to 4 or 5 in parallel on a single microcontroller pin. We currently have these sequins in warm white, emerald green, ruby red, and royal blue.

You can make the LEDs fade and twinkle by using the PWM (a.k.a. analogWrite) functionality of your Gemma or Flora, or just connect directly to a digital I/O pin of a microcontroller to turn on and off (digitalWrite).

Sewing with conductive thread

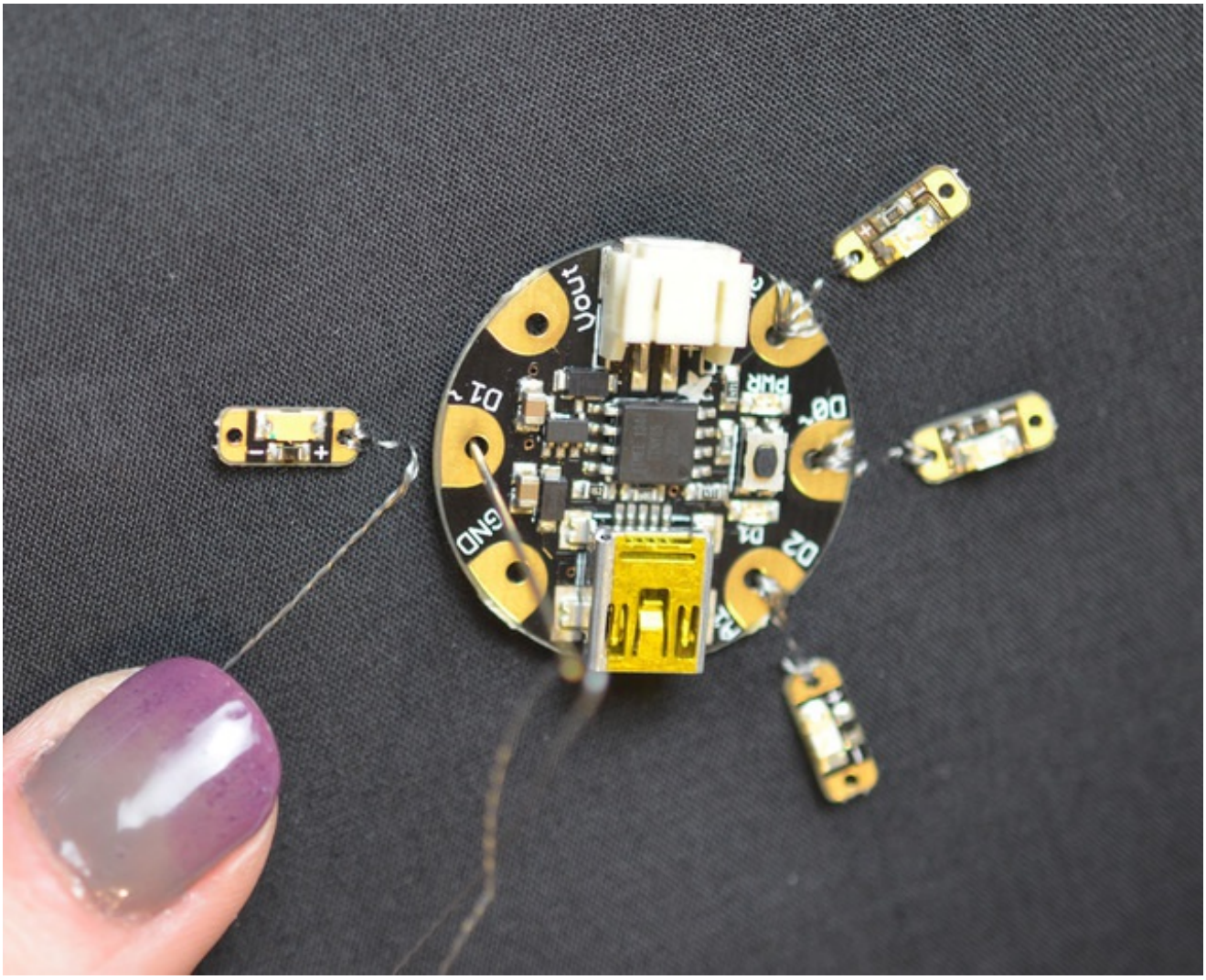


Grab a needle and test it's small enough to pass through the holes in the sequins. Thread up your needle with conductive thread, our 2-ply and 3-ply both are great for this purpose. Check out our conductive thread guide(http://adafru.it/aVx) to learn more about this stainless steel fiber.

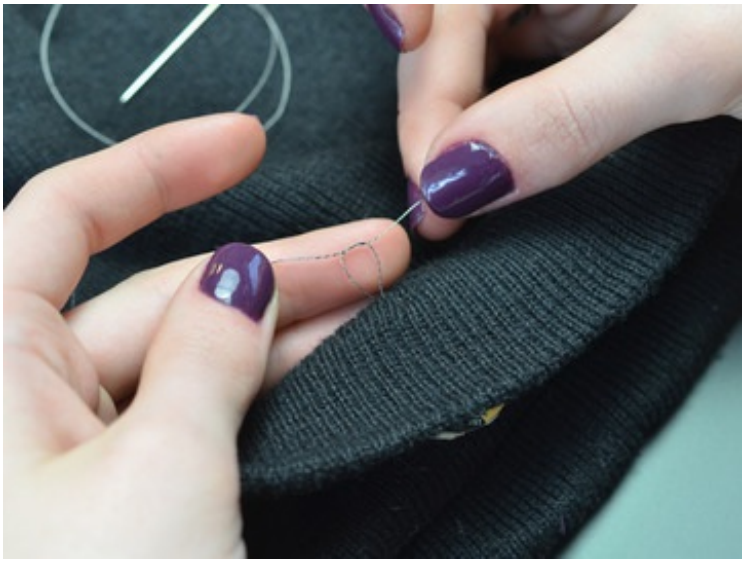


We find it easiest to load up your fabric in an embroidery hoop to keep it taught during stitching, especially for beginners.

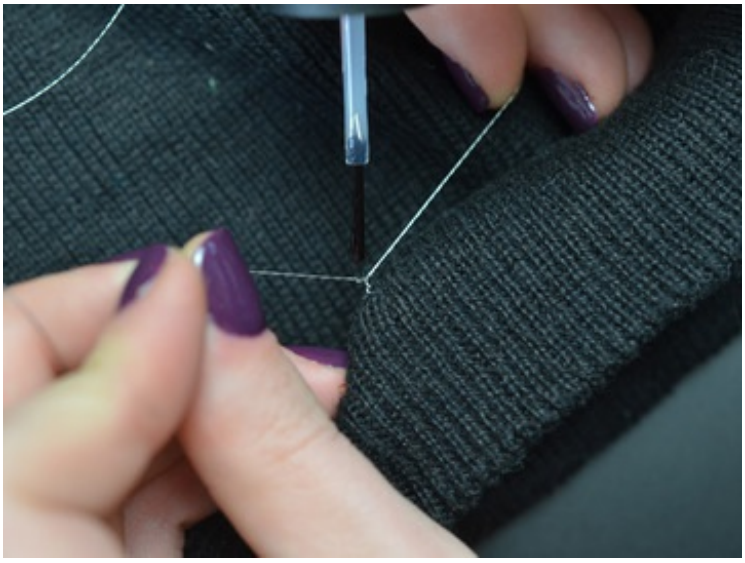
Begin by piercing the needle through the fabric from back to front, near the + on the pixel leaving a six-inch thread tail at the back. Then affix the pixel to the fabric by piercing the needle down through the hole marked + and through to the back of the fabric. Repeat to make a few stitches around the pixel's + connection.



Stitch over to a digital or analog output on your microcontroller, and repeat the stitching process around the circuit board's pad.

- 

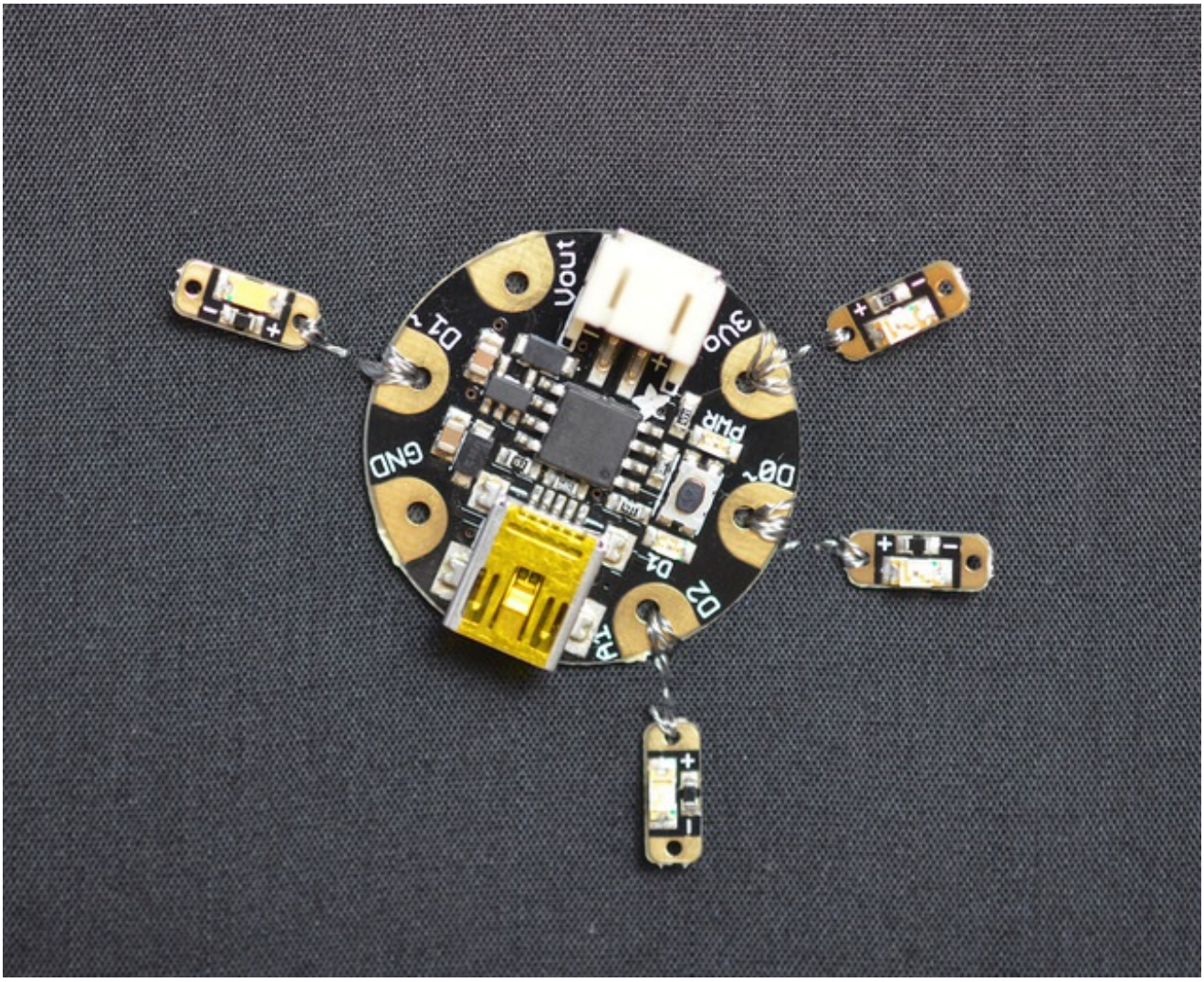
  Stitch back over to your thread tail at the back and tie the two threads in a double knot.

- 

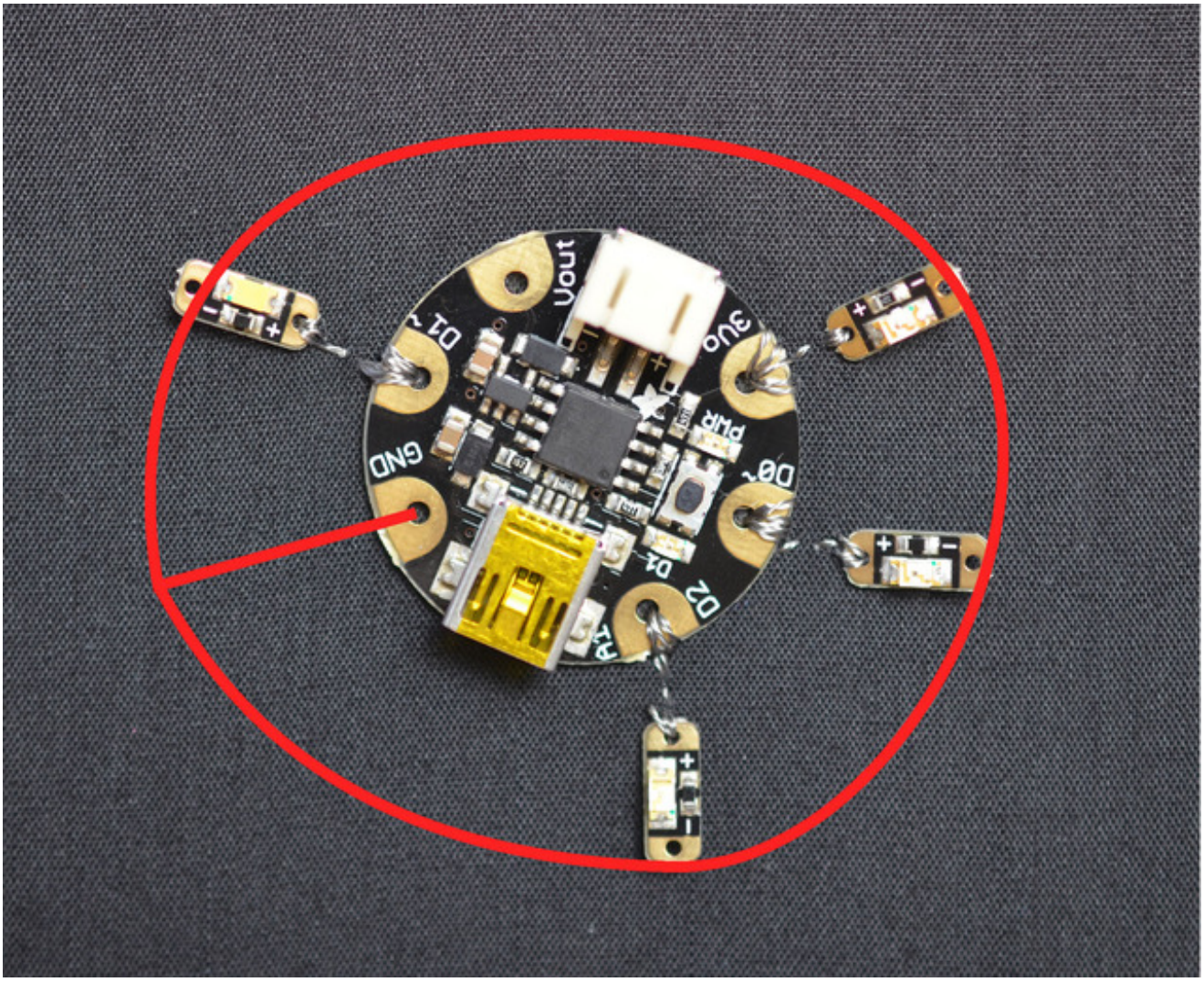
  Seal the knot from springing loose with some clear nail polish or other adhesive.

  Snip the thread tails short.

- 

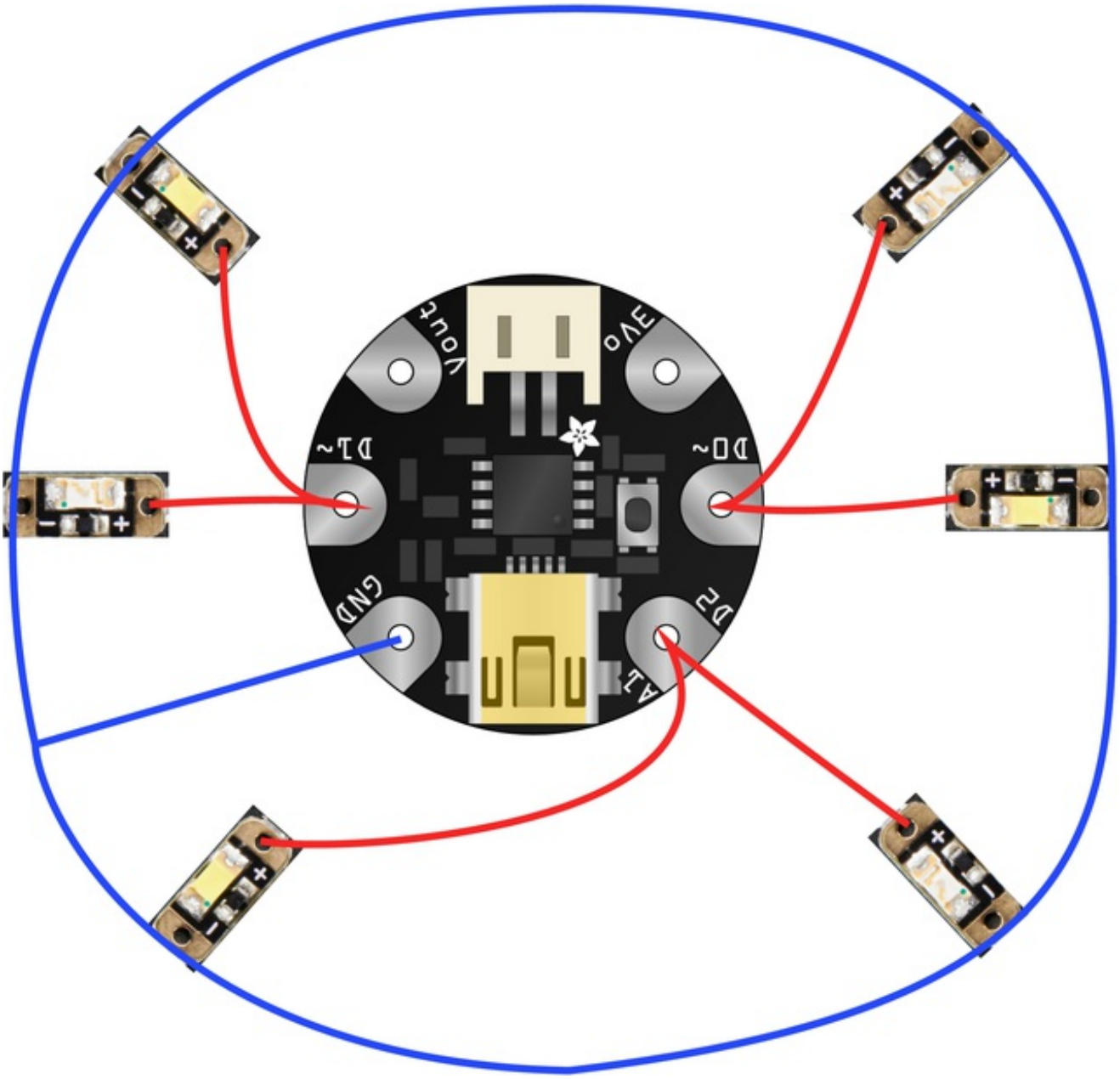


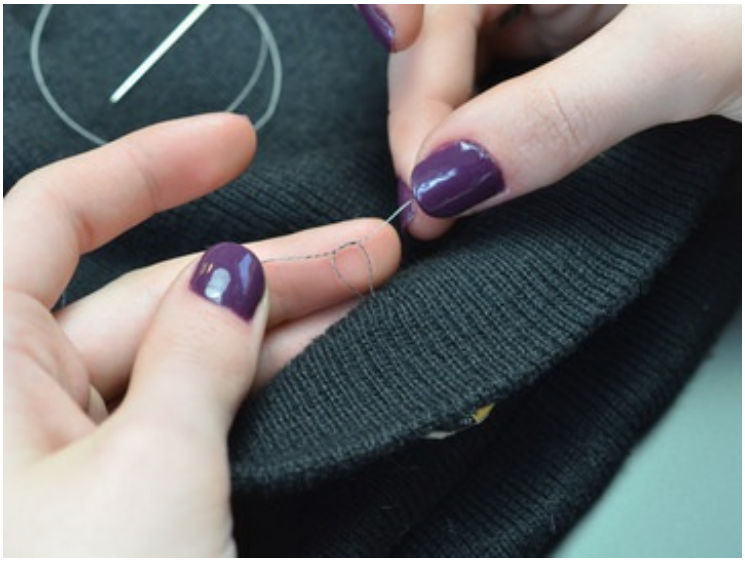
After sewing the positive leads, stitch a line from GND on your microcontroller to connect all sequins' grounds (marked with a -), as illustrated below.

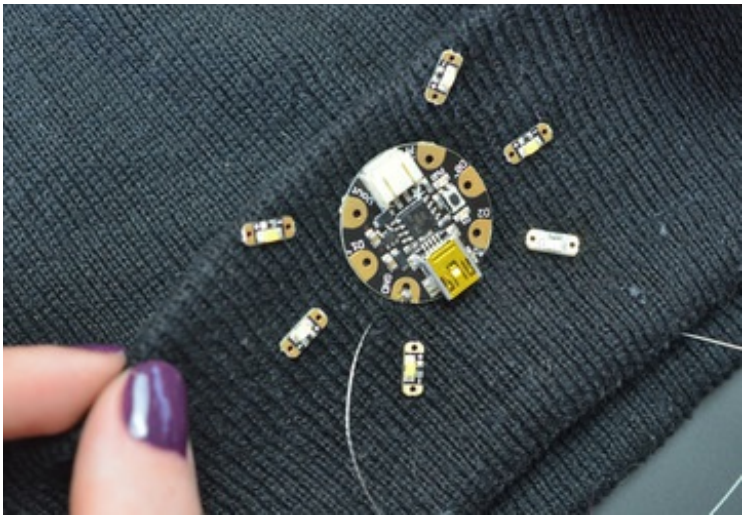
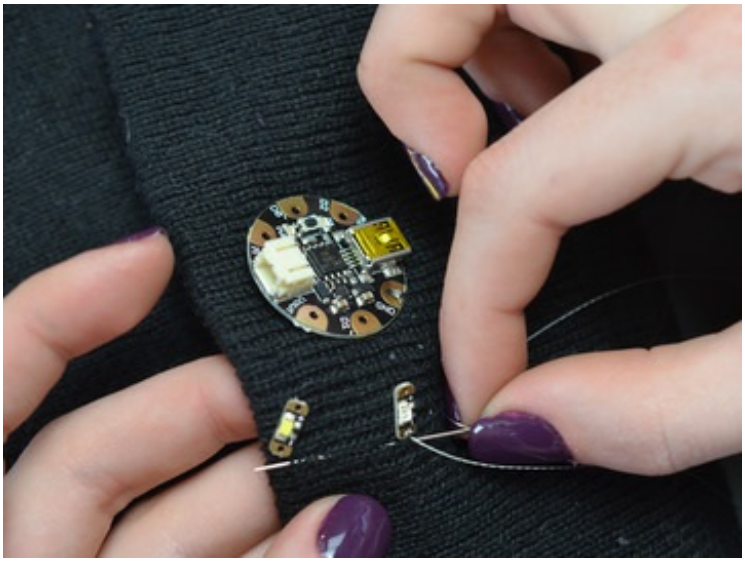

Vout
3V3
PWR
D0
D1
GND
D2
D4
A3

GEMMA sequin hat

LED sequins are great for clothing and accessories! Here's a simple hat project to build with GEMMA.

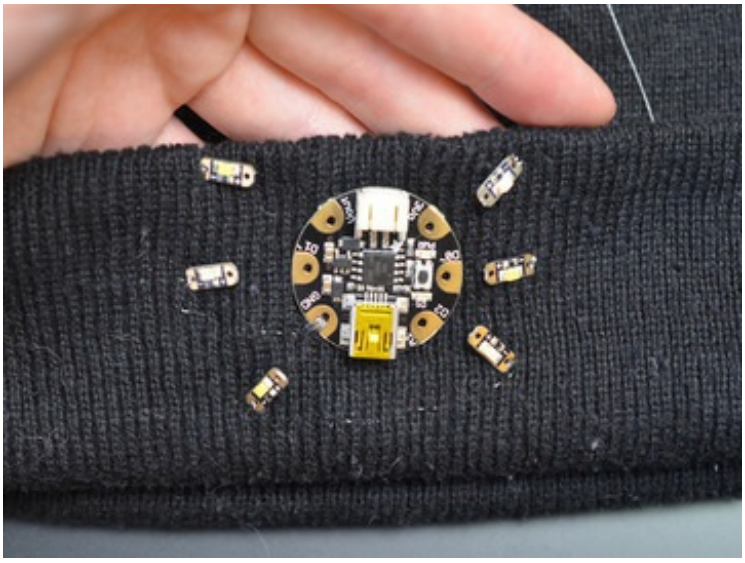

Vout
3Vo
D1~
D0~
GND
D2
A1

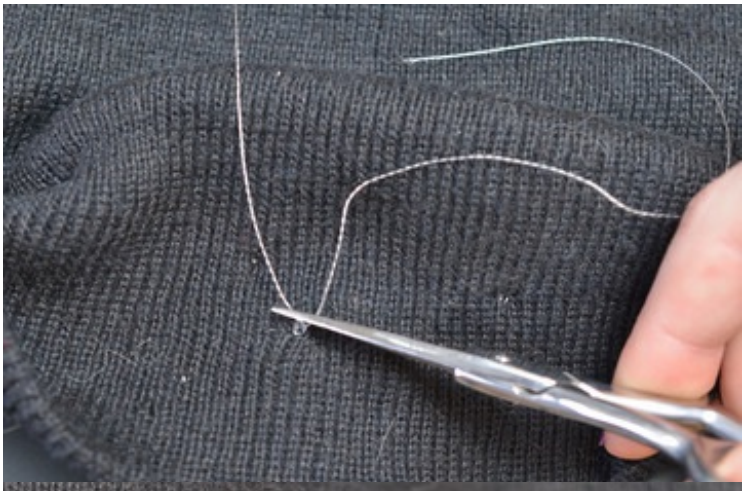
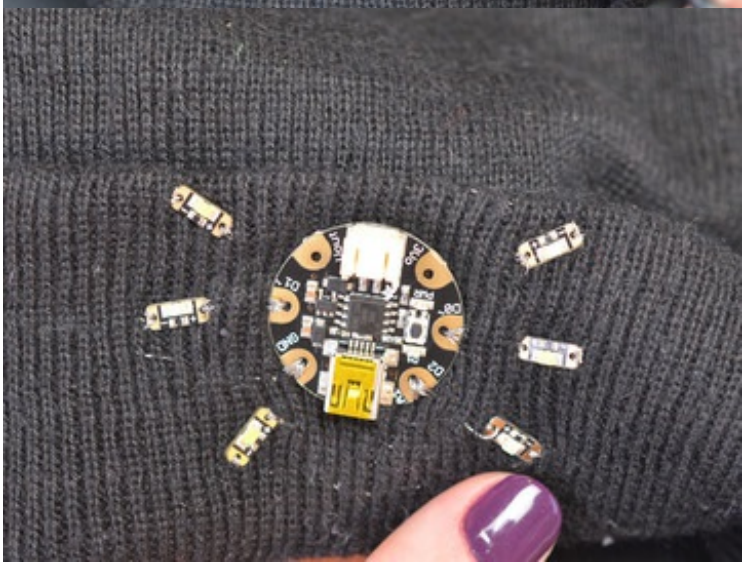
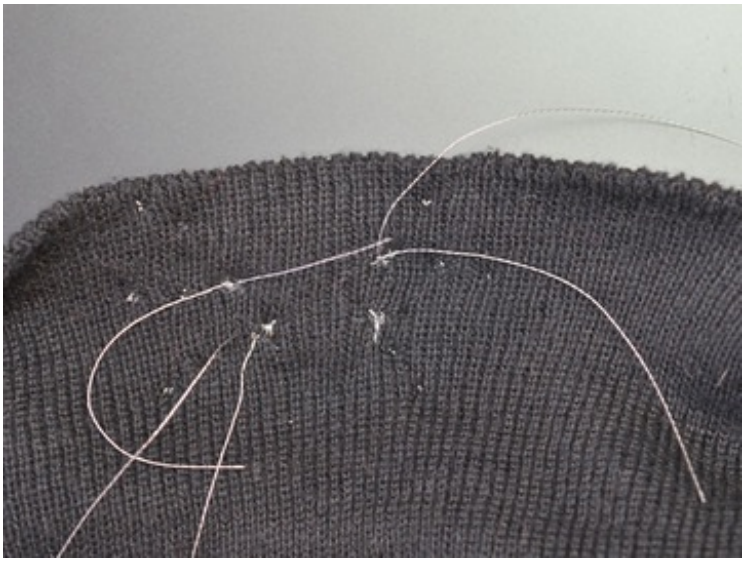
- 
  You can start with the pixel + or -. For this hat we chose to start with the shared ground line that hooks up all the sequins. Stitch around the GND pad on GEMMA at the edge of a knit cap and knot/seal at the back.
- 
- 
- 

- 
- 

  Lay out your own design or use our circuit diagram above, with the sequins' + sides facing the microcontroller.

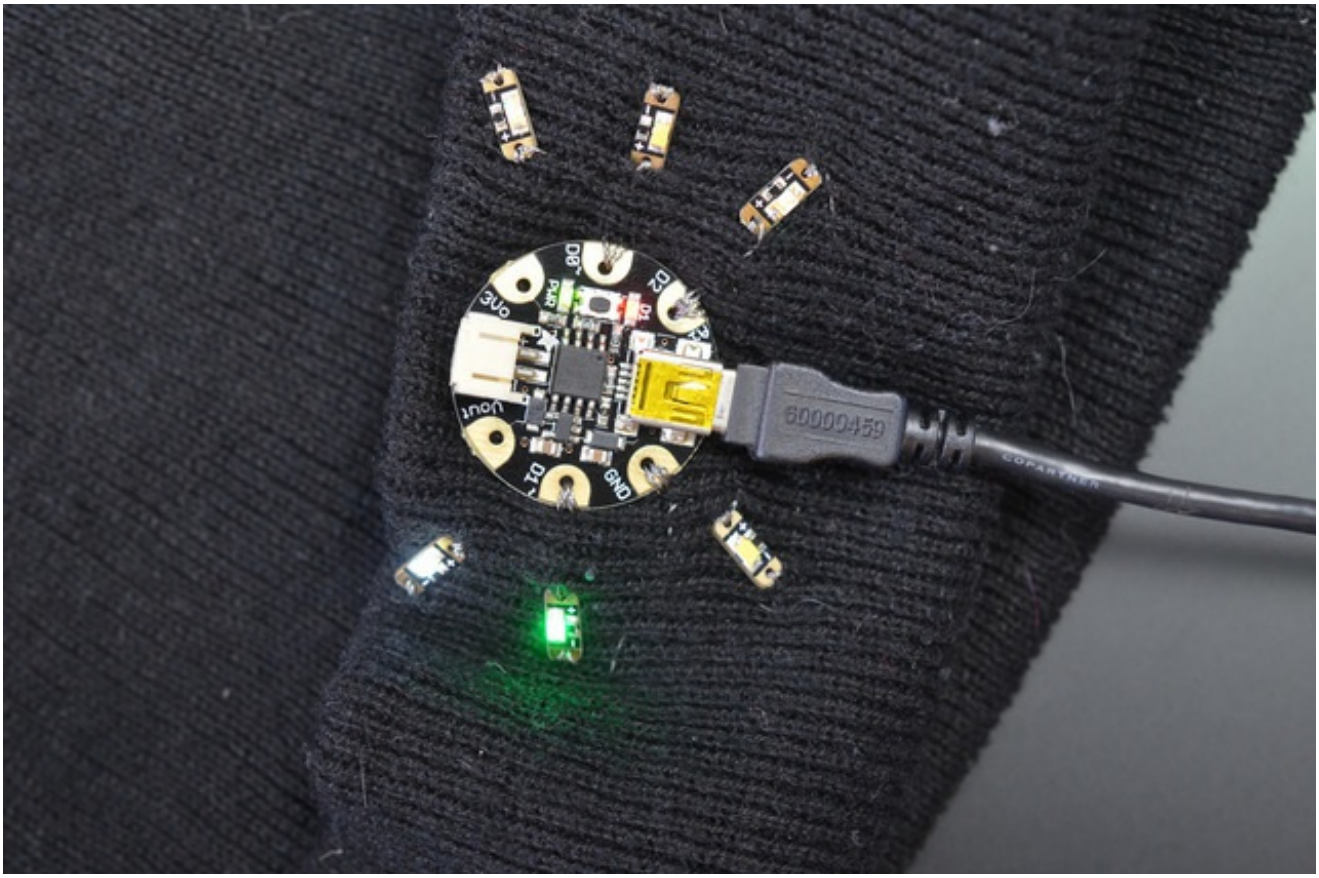
  Continue stitching the ground line all the way around the perimeter of your sequin design, stitching to every sequin's - pad as you go.

  Knot to your original knot, seal the knot, and trim the ends once dry.
- 

- 
- 
- 

Next hook up the + connections from each of GEMMA's outputs to two pixels. We stitched from one sequin to the GEMMA, then over to another sequin and back to the first, making a sort of triangle that results in the thread tails being in the same place.

Knot these thread tails, seal and trim short.

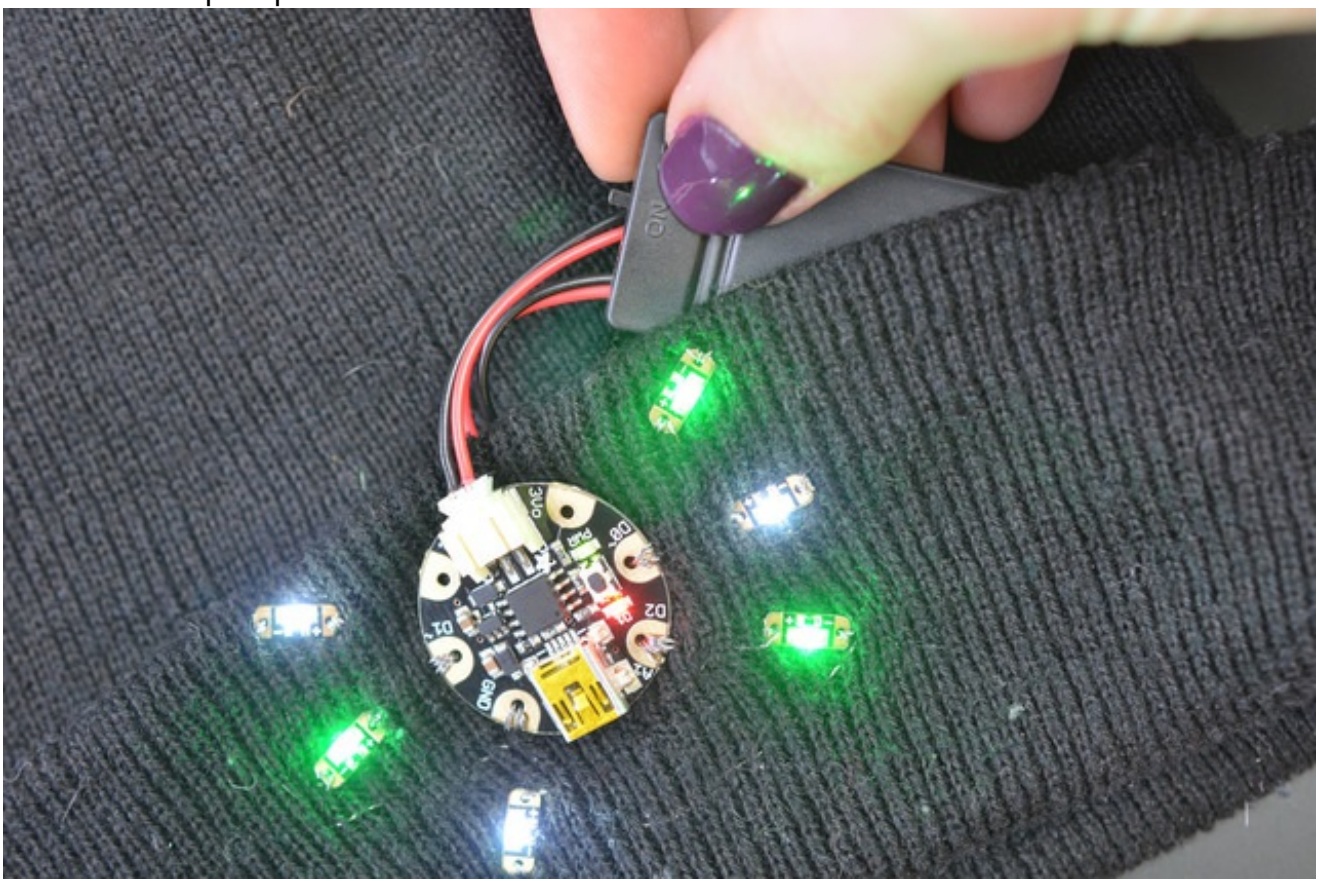


Plug in GEMMA over USB and load the following code into your Adafruit Arduino IDE. If you haven't before, check out our Introducing GEMMA guide (http://adafru.it/cHH) to get started with the software.

```
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by
int counter = 0;       // counter to keep track of cycles

// the setup routine runs once when you press reset:
void setup()  {
  // declare pins to be an outputs:
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop()  {
  // set the brightness of the analog-connected LEDs:
  analogWrite(1, brightness);
  analogWrite(0, 255-brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
```

```
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount;
    counter++;
  }
  // wait for 15 milliseconds to see the dimming effect
  delay(15);

// turns on the other LEDs every four times through the fade by
  // checking the modulo of the counter.
  // the modulo function gives you the remainder of
  // the division of two numbers:
  if (counter % 4 == 0) {
    digitalWrite(2, HIGH);
  } else {
   digitalWrite(2, LOW);
  }
}
```

This sketch is a mash-up of two very basic Arduino examples: blink and fade. You can modify it to display the patterns of light you like best or code up your own sketch starting with the examples provided from inside Arduino.



Once your program is doing what you like, unplug the USB cable and plug in a battery pack like our 2x2032 holder with on/off switch.

Take the batteries out if you get stuck in a rainstorm and for washing. Enjoy your new light-

up hat!

Downloads

- Datasheet for blue 1206 LED (http://adafru.it/qle)
- Datasheet for warm white LED (http://adafru.it/qlf)
- Datasheet for red 1206 LED (http://adafru.it/qlA)
- Datasheet for green LED (http://adafru.it/rla)
- Datasheet for pink LED (http://adafru.it/rlc)
- EagleCAD PCB files on GitHub (http://adafru.it/rlb)
- Fritzing object in Adafruit Fritzing library (http://adafru.it/aP3)