# Interior Purse Light

Created by Becky Stern

https://learn.adafruit.com/interior-purse-light

Last updated on 2024-06-03 01:30:15 PM EDT

# Table of Contents

# Overview

Trying to find something in your purse at a dark restaurant or party? What a pain! Make it easy to see what's in your bag with LED sequins, GEMMA, and conductive hook & loop

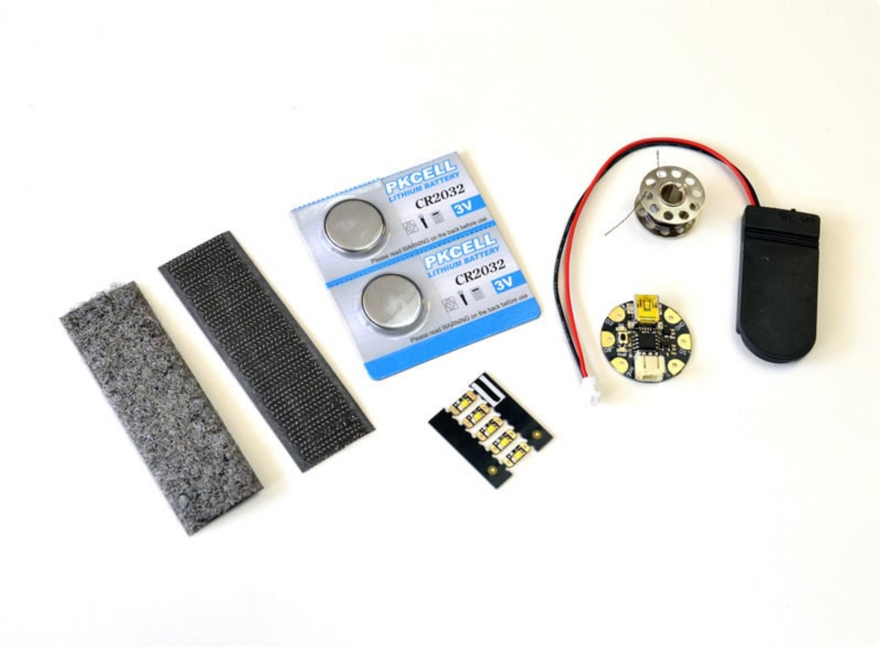
This beginner level e-textiles project has you sewing with conductive thread and loading a simple Arduino program onto the GEMMA microcontroller. This project is perfect for use with the GEMMA Sequin Starter Pack (http://adafru.it/1850)! Just add hook & loop (http://adafru.it/1324). Before you get started, follow the Gemma M0 guide (https://adafru.it/zxE) or the Classic Introducing GEMMA guide (https://adafru.it/e1V)

This guide was written for the Gemma v2 board, but can be done with either the v2 or Gemma M0. We recommend the Gemma M0 as it is easier to use and is more compatible with modern computers!

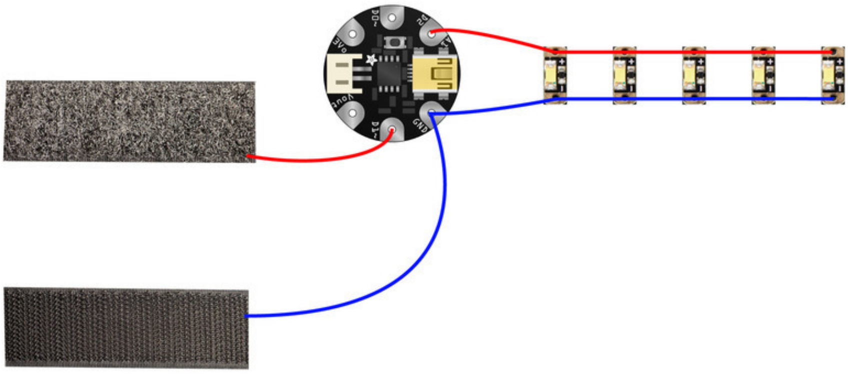
Gather up the following parts & tools:

- Gemma M0 (http://adafru.it/3501) or GEMMA v2 wearable microcontroller (http://adafru.it/1222)
- (Optional): build on the: GEMMA Starter Pack (http://adafru.it/1657) or GEMMA Sequin Starter Pack (http://adafru.it/1850)
- Conductive hook & loop (http://adafru.it/1324)
- Purse or backpack
- Scissors

- Clear nail polish
- USB cable
- Computer with Adafruit Arduino software



## Circuit Diagram

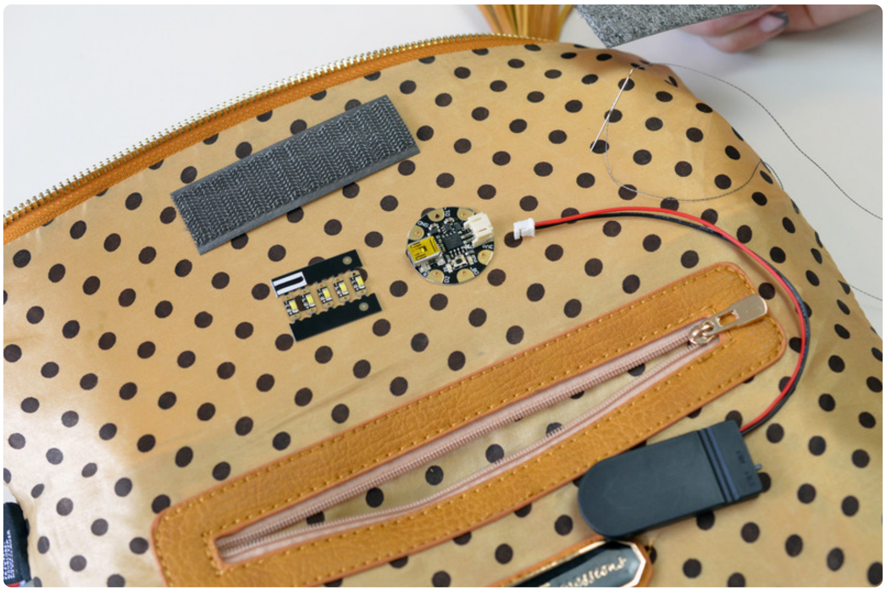


This diagram uses the original Gemma but you can also use the Gemma M0 with the exact same wiring!

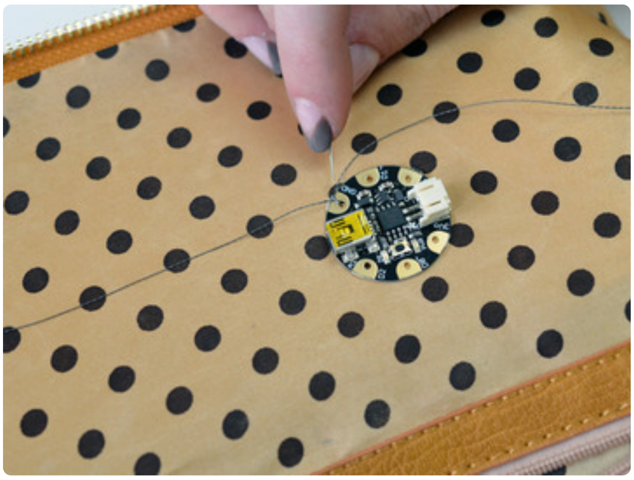
Click to enlarge! Five LED sequins are sewn in parallel to GEMMA's D2 and GND pads.

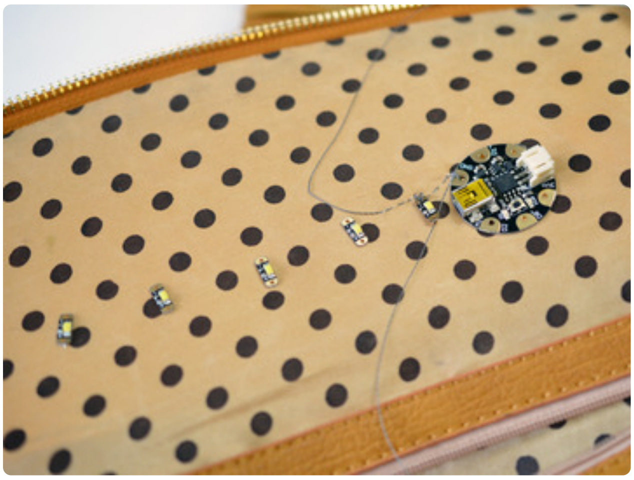
Conductive hook & loop is sewn to GND and D1 pads. In the Arduino program D1's internal pullup resistor is activated.

## Stitch Sequins



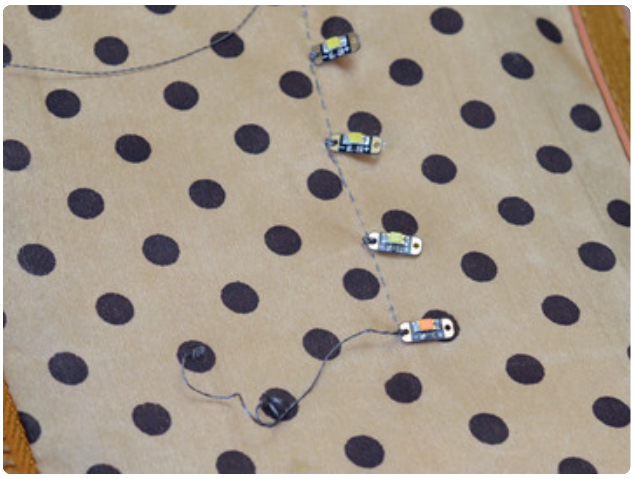
Clean off your work space and turn your bag inside out. Arrange the components on your bag to get an idea of where you'd like to position everything, referring to the circuit diagram.

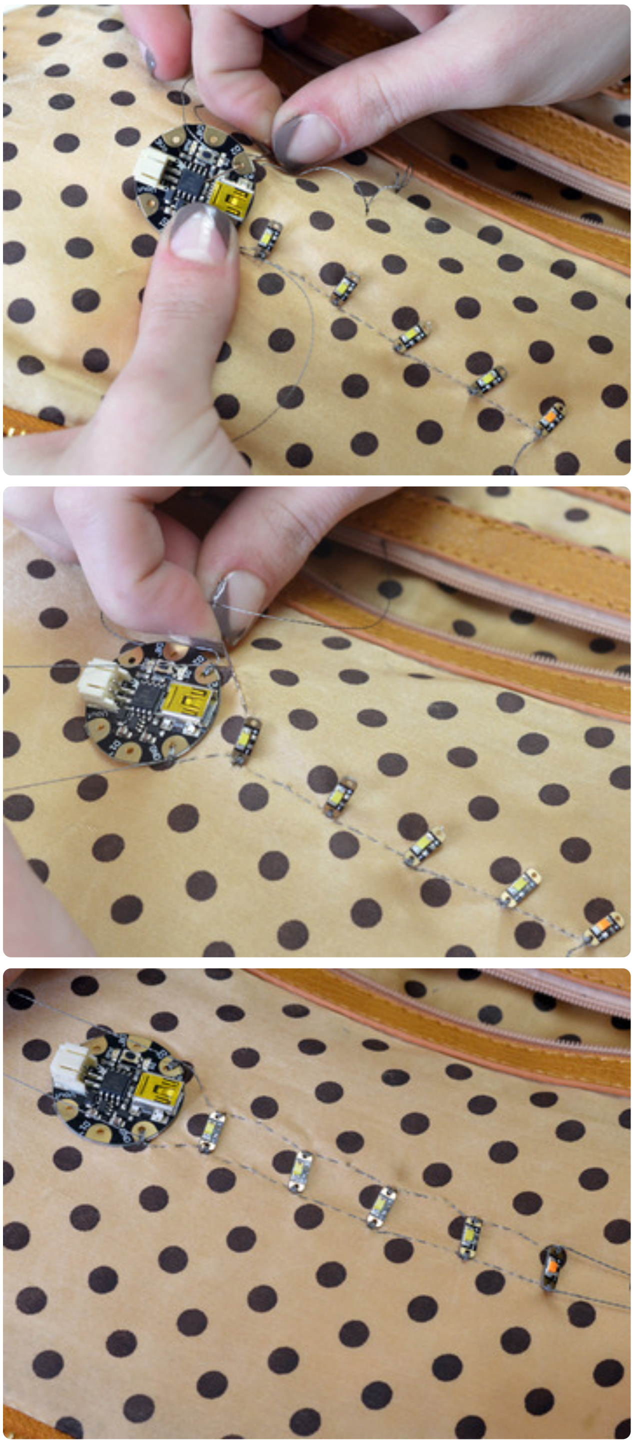




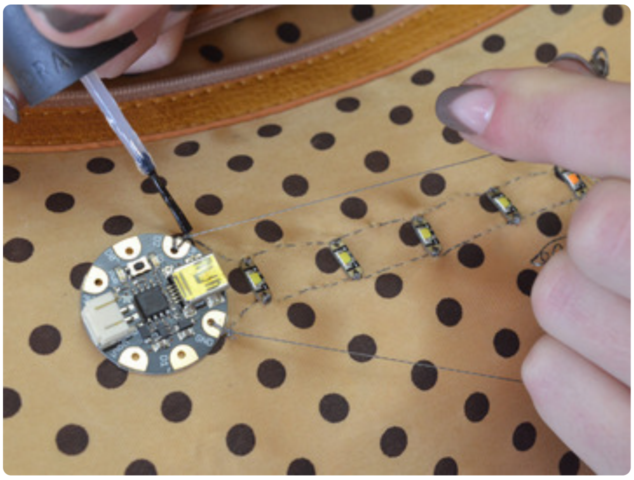
Thread a needle with conductive thread and stitch around GND on GEMMA. Tie a knot and continue stitching toward the first sequin. Stitch around its negative pad (marked "-") a few times, then continue down the line, stitching to each sequin.

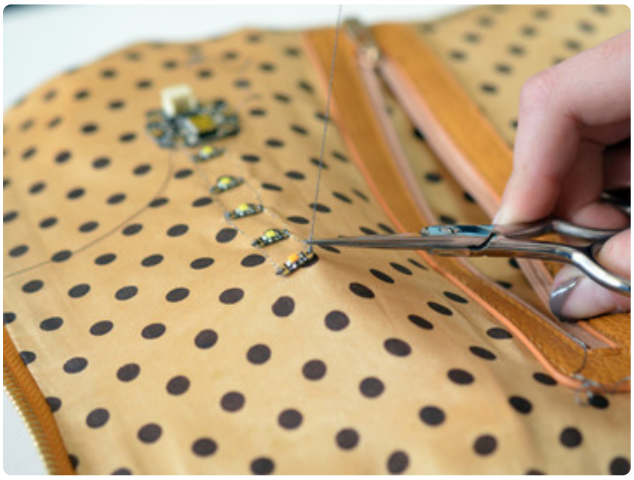
Tie a knot at the last pixel (we chose pink for extra flair), and cut the thread tail long.





Repeat the last step on the other side of the pixels, stitching them all to D2 on GEMMA.





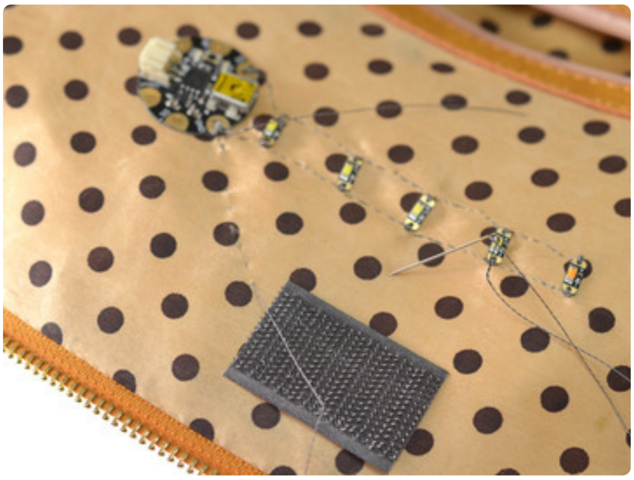
Pull your thread tails taught and dab on a small amount of clear nail polish-- it should only touch the knot, not the pads of GEMMA or the sequins.

For more tips on working with conductive thread, check out our Conductive Thread guide (https://adafru.it/dn3).

When dry, snip the thread tails short.

Double check you don't have any rogue bits of thread hanging out anywhere, and that any stray bits have been cleaned up.

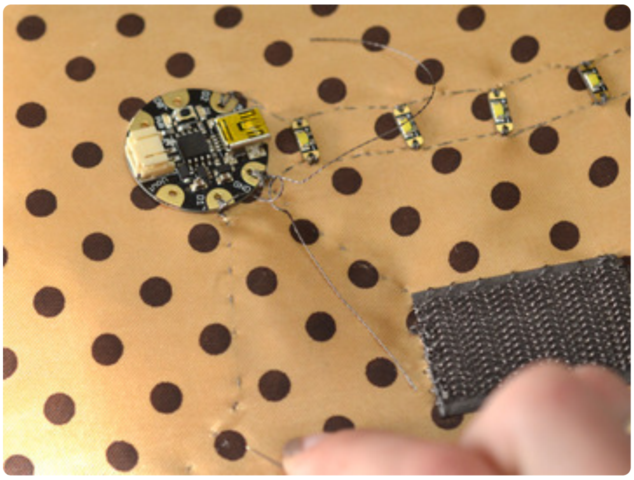
## Add Tape

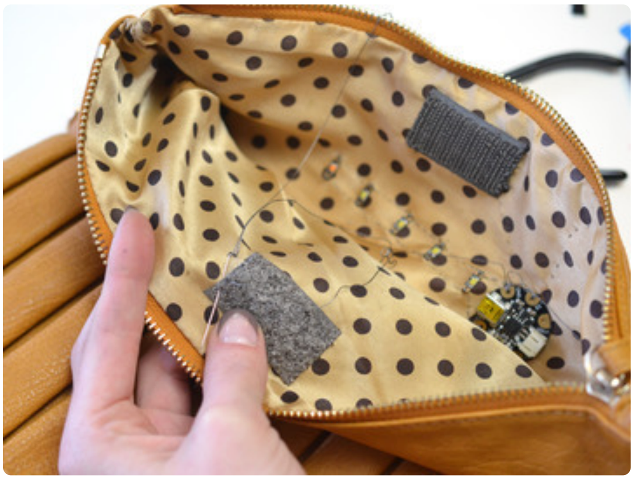


Thread your needle with another length of conductive thread, and stitch to GND on GEMMA again. Tie a knot, then stitch over to one piece of conductive hook & loop.

This stuff can be difficult to sew through, so you may want to have a pair of pliers handy for helping to grasp the needle.

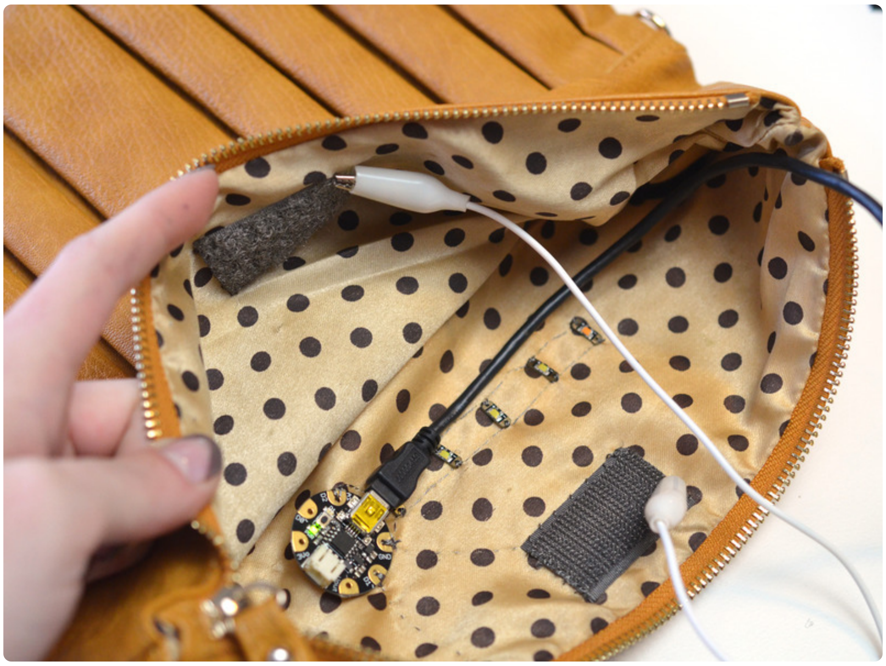
Since the hook&loop will pull at it's mating piece, stitch it very securely to the bag, around all four edges. Then tie it off and cut the tail.





Stitch around pad D1 on GEMMA with a new piece of conductive thread, then stitch all the way around the opening of the bag to the other side. Repeat the hook & loop securing on this side, making sure the two pieces are aligned.

Jump to the CircuitPython Code (https://adafru.it/Cd4) or Arduino Code (https://adafru.it/Cd5) section for programming your Gemma.



Now your bag should light up when it's open! If you can't see into the bag when it's shut, use an alligator clip or piece of wire to connect the two pieces of hook & loop to see if the light shut off. When you've verified the sensor is working properly, you can disconnect the USB cable and connect up a coincell battery pack for taking your project portable!

# Arduino Code

The Arduino code presented below works equally well on all versions of GEMMA: v2 and M0. But if you have an M0 board, consider using the CircuitPython code on the next page of this guide, no Arduino IDE required!

Grab the modified "Button" example sketch below and load it onto GEMMA:

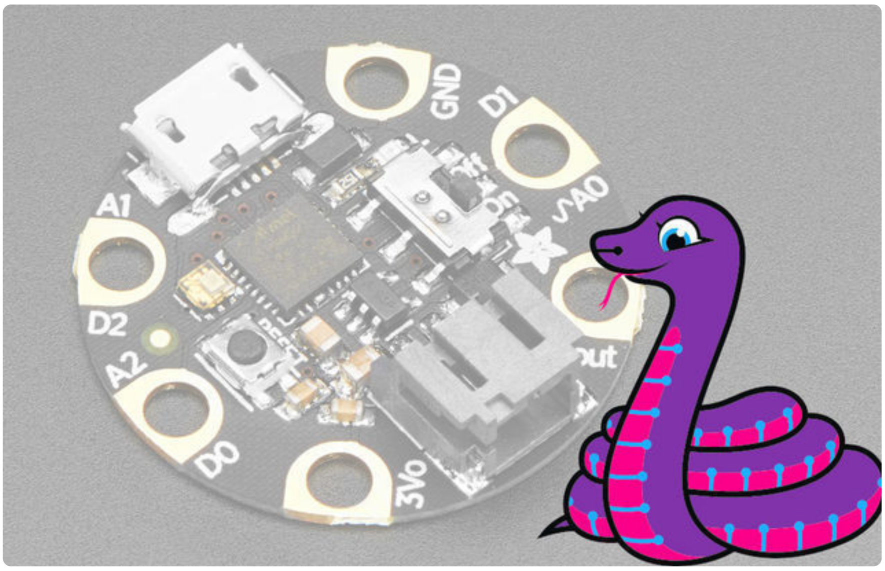
```
// SPDX-FileCopyrightText: 2017 Mikey Sklar for Adafruit Industries
//
// SPDX-License-Identifier: MIT

// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 1;     // the number of the pushbutton pin
const int ledPin =  2;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

From the **Tools→Board** menu, select the device you are using:

- **Adafruit Gemma M0**
- **Adafruit Gemma 8 MHz**

Connect the USB cable between the computer and your device. The original Gemma (8 MHz) need the reset button pressed on the board, then click the upload button (right arrow icon) in the Arduino IDE. You do not need to press the reset on the newer Gemma M0.

# CircuitPython Code



**GEMMA M0** boards can run **CircuitPython** — a different approach to programming compared to Arduino sketches. In fact, **CircuitPython comes factory pre-loaded on GEMMA M0**. If you've overwritten it with an Arduino sketch, or just want to learn the basics of setting up and using CircuitPython, this is explained in the **Adafruit GEMMA M0 guide** (https://adafru.it/z1B).

These directions are specific to the "M0" GEMMA board. The original GEMMA with an 8-bit AVR microcontroller doesn't run CircuitPython…for those boards, use the Arduino sketch on the "Arduino code" page of this guide.

Below is CircuitPython code that works similarly (though not exactly the same) as the Arduino sketch shown on a prior page. To use this, plug the GEMMA M0 into USB…it should show up on your computer as a small **flash drive**…then edit the file "**main.py**" with your text editor of choice. Select and copy the code below and paste it into that file, **entirely replacing its contents** (don't mix it in with lingering bits of old code). When you save the file, the code should **start running almost immediately** (if not, see notes at the bottom of this page).

**If GEMMA M0 doesn't show up as a drive, follow the GEMMA M0 guide link above to prepare the board for CircuitPython.**

```
# SPDX-FileCopyrightText: 2017 Limor Fried for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time

import board
from digitalio import DigitalInOut, Direction, Pull
```

```
button = DigitalInOut(board.D1)
button.direction = Direction.INPUT
button.pull = Pull.UP

led = DigitalInOut(board.D2)
led.direction = Direction.OUTPUT

while True:
    if button.value:
        led.value = True  # check if the pushbutton is pressed.
    else:
        led.value = False  # turn LED off

    time.sleep(0.01)  # debounce delay
```

# Use It!

Take your purse to a dark club or party, but always be able to see what's inside! Avoid getting the circuit wet while the battery is connected, and remove the battery pack for laundering.

Depending on your circuit placement, you'll also have to watch out for metal objects that could short your GEMMA. Happy stitching!