



# Light-Activated Pixel Heart

Created by Becky Stern



<https://learn.adafruit.com/light-activated-pixel-heart>

Last updated on 2023-08-29 02:29:23 PM EDT

# Table of Contents

<a href="#">Overview</a>	3
<a href="#">Stitch Circuit</a>	4
<a href="#">Arduino Code</a>	9
<ul style="list-style-type: none"><li>• <a href="#">NeoPixel Überguide: Arduino Library Installation</a></li></ul>	
<a href="#">CircuitPython Code</a>	11
<a href="#">Stitch Circuit Part 2</a>	13
<a href="#">Wear it!</a>	16
<a href="#">Circuit Diagram</a>	17

---

# Overview

Wear your heart on your sleeve! Or at least on your lapel. This light-up heart display uses a light-dependent resistor and GEMMA, Adafruit's tiny wearable electronics platform.



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

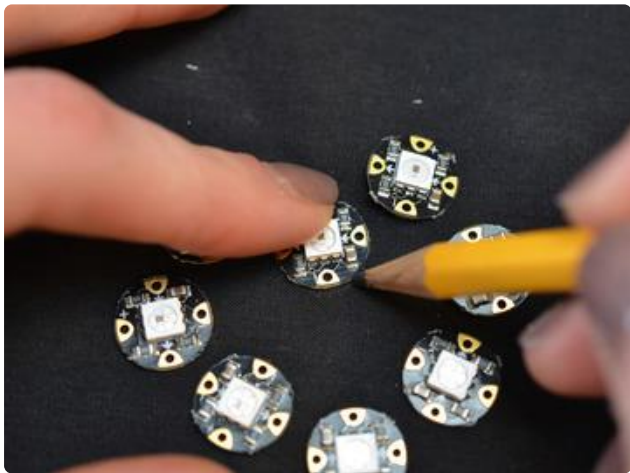
---

## Stitch Circuit

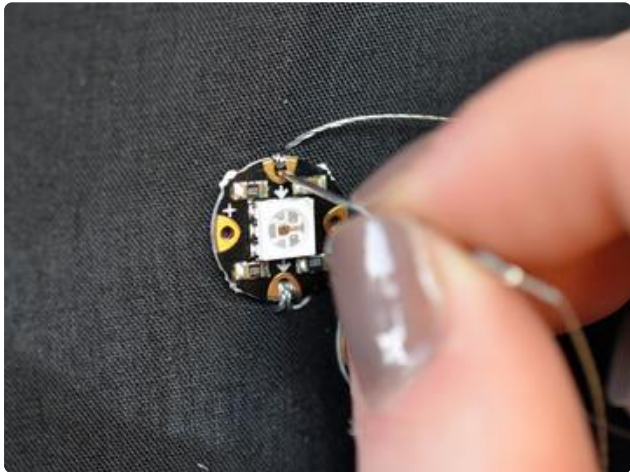
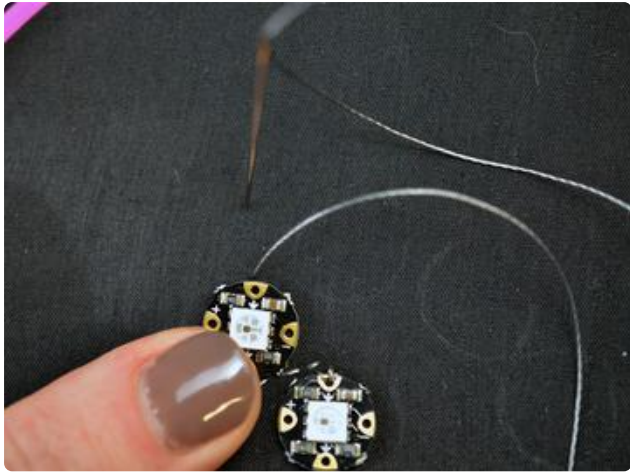


Place a piece of woven fabric in a 6-inch or larger embroidery hoop.

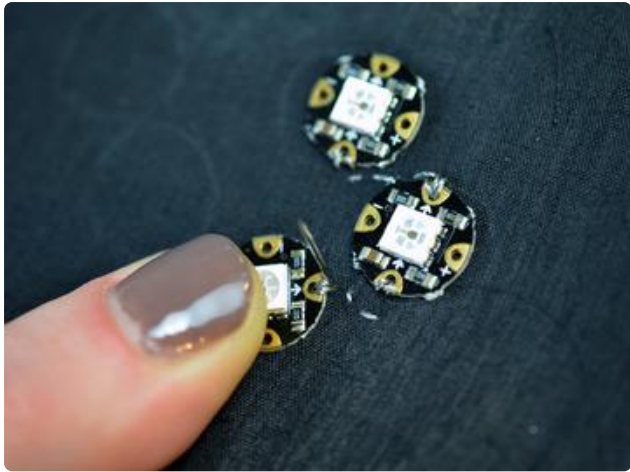
Arrange eight sewable NeoPixels in a heart on your fabric.



Trace around your pixels with a pencil or a disappearing-ink embroidery pen.



Start stitching the data lines between each pixel according to the circuit diagram. Begin by bringing the thread from back to front, leaving a 6-inch tail, and stitching around the inward facing arrow of one pixel, piercing through the fabric from front to back. Repeat a few times to secure the pixel to the fabric. Tie the tail in a knot with the working thread at the back.



Use a running stitch to make your way to the outward-facing arrow on the next pixel, using your trace marks for guidance in pixel placement.



After you've stitched around the pixel's pad a few times, tie the thread in a knot at the back.

Apply tension to thread tails and dab clear nail polish onto the knots.

Allow to dry, periodically tugging on the thread tails to keep the knots tight.

Only clip thread tails short when completely dry to prevent unraveling!

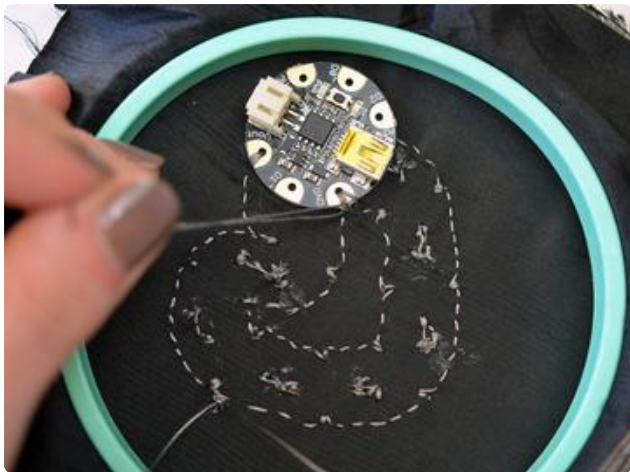


Continue making short connections between each pixel's data connections, making sure the arrows all point in the same direction along the chain. Refer to the circuit diagram.





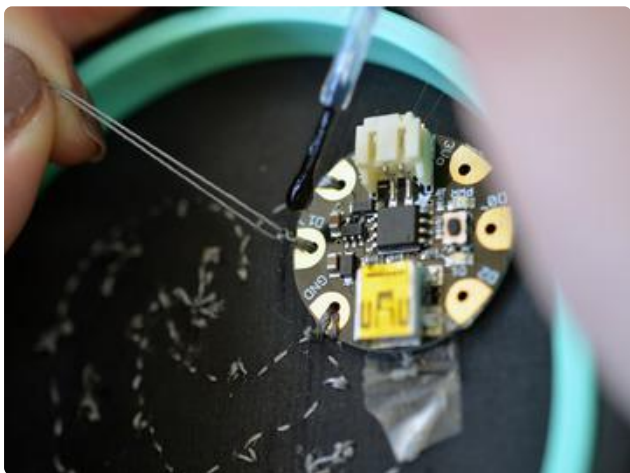
Use a running stitch to trace around the perimeter of the heart, connecting to each pixel's power (+) pad as you go. Connect this thread to Vout on GEMMA at the back.



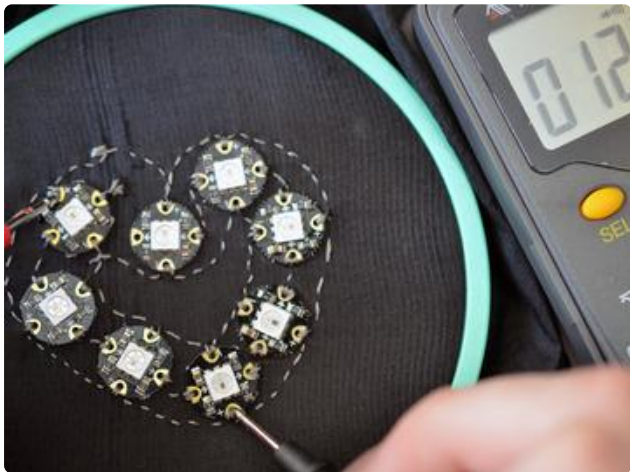
Use another thread and running stitch to connect up all the ground (-) pads of the pixels, circling the inner perimeter of the heart. Connect this thread to GND on GEMMA at the back.

Connect the first pixel's inward-facing arrow to D1 on GEMMA.

Tie off and seal all knots before trimming thread tails.



Put a piece of tape or fabric behind GEMMA where it overlaps any stitches it shouldn't touch.



Use a multimeter in continuity-testing mode to test out your circuit. Make sure power isn't connected to ground, and verify that all points that should be connected are, indeed, connected.





Plug in the USB cable and test out the pixels using the strandtest sketch included in the Adafruit NeoPixel library for Arduino (change the value of PIN to 1 before compiling).

Are they all working? Great! If not, unplug and re-evaluate your circuit. Clean up any frayed or loose bits of thread, re-stitch any loose spots and double-seal your knots. Probe again with your multimeter before reconnecting via USB.

If you're happy with the flashing lights, you're done! Sew your new circuit swatch into a brooch or onto your Valentine's outfit, and rock your new flashy accessory.

But if you want your heart to turn on only when the moment's right, continue on to add a light sensor, perfect for revealing your heart from under a jacket, hat, or hemline.

The circuit isn't finished yet, but let's get the final code onto the board before continuing, so we can test it along the way...

---

## Arduino Code

The Arduino code presented below works equally well on all versions of GEMMA: v1, 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!

If this is your first time using GEMMA, work through the [Introducing GEMMA](#) guide first; you need to customize some settings in the Arduino IDE. Once you have it up and running (test the 'blink' sketch), then follow the instructions on the following page for installing the NeoPixel library:

## NeoPixel Überguide: Arduino Library Installation ()

Plug in your circuit and load up the sketch below:

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

#include <Adafruit_NeoPixel.h>

#define NUM_LEDS 8 // This many NeoPixels...
#define LED_PIN 1 // are connected to this DIGITAL pin #
#define SENSOR 2 // Light sensor to this DIGITAL pin #

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN);

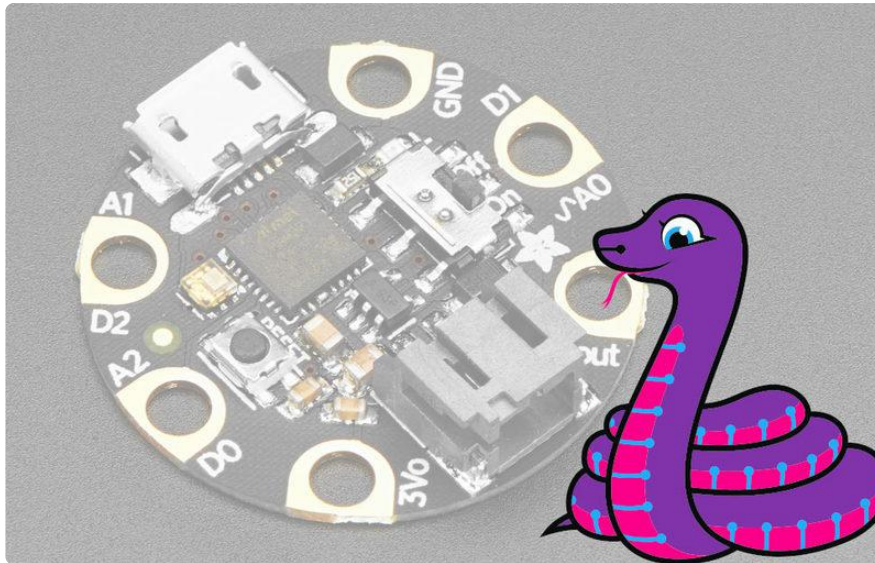
void setup() {
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
  pinMode(SENSOR, INPUT_PULLUP); // Enable pull-up resistor on sensor pin
}

void loop() {
  // The LDR is being used as a digital (binary) sensor, so it must be
  // COMPLETELY dark to turn it off, your finger is not opaque enough!
  if(!digitalRead(SENSOR)) { // Sensor exposed to light?
    colorWipe(strip.Color(255, 0, 255), 50); // Animate purple
  } else { // else sensor is dark
    colorWipe(strip.Color(0, 0, 0), 50); // Animate off
  }
  delay(2); // Pause 2 ms before repeating
}

// Fill pixels one after the other with a color
void colorWipe(uint32_t c, uint8_t wait) {
  for(uint16_t i=0; i<strip.numPixels(); i++) {
    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}
```

---

# 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 \(\)](#).

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.

This code requires the neopixel.py library. A factory-fresh board will have this already installed. If you’ve just reloaded the board with CircuitPython, create the “lib” directory and then [download neopixel.py from Github \(\)](#).

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

```

# SPDX-License-Identifier: MIT

import time

import board
import digitalio
import neopixel

numpix = 8 # Number of NeoPixels
ledpin = board.D1 # Digital pin # where NeoPixels are connected
sensorpin = board.D2 # Digital pin # where light sensor is connected
strip = neopixel.NeoPixel(ledpin, numpix, brightness=1.0)

# Enable internal pullup resistor on sensor pin
pin = digitalio.DigitalInOut(sensorpin)
pin.direction = digitalio.Direction.INPUT
pin.pull = digitalio.Pull.UP

while True: # Loop forever...

    # LDR is being used as a digital (binary) sensor. It must be
    # completely dark to turn it off, a finger may not be opaque enough!
    if pin.value:
        color = (0, 0, 0) # Off
    else:
        color = (255, 0, 255) # Purple

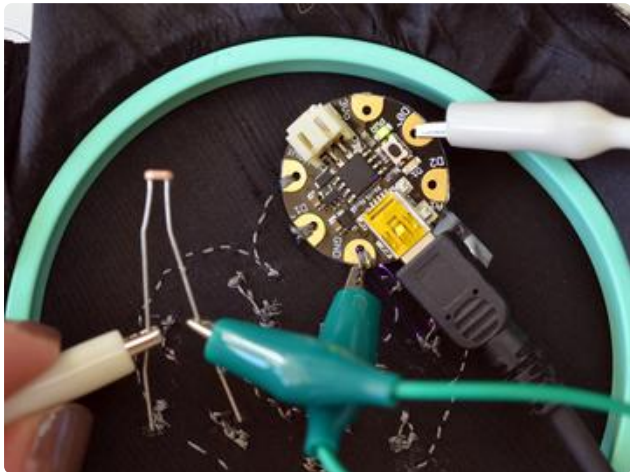
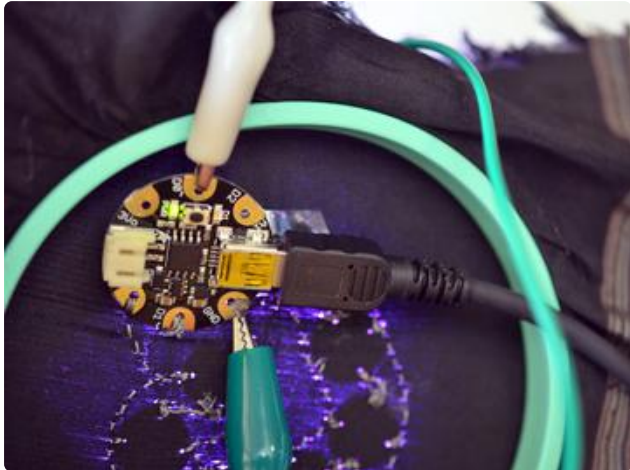
    for i in range(numpix): # For each pixel...
        strip[i] = color # Set to 'color'
        strip.write() # Push data to pixels
        time.sleep(0.05) # Pause 50 ms

    time.sleep(0.002) # Pause 2 ms

```

---

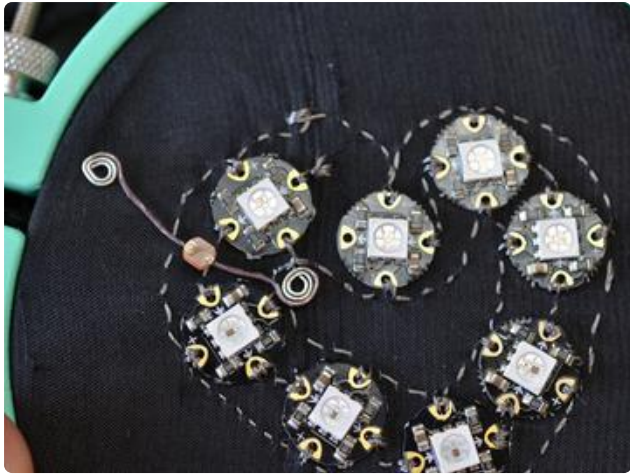
## Stitch Circuit Part 2



Let's test before finishing the sewing. Use alligator clips to connect one leg of the photocell to D0 on GEMMA and the other to GND.

The pixels should animate pink when the sensor is exposed to light. Likewise they should turn off when the sensor is covered and completely dark.





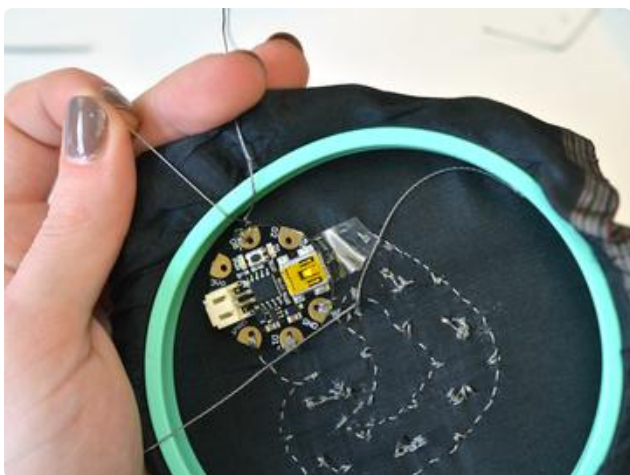
To attach the LDR permanently, twist the ends of the leads into spirals as shown and paint the straight parts with nail polish-- it will insulate the LDR from the traces of conductive thread.

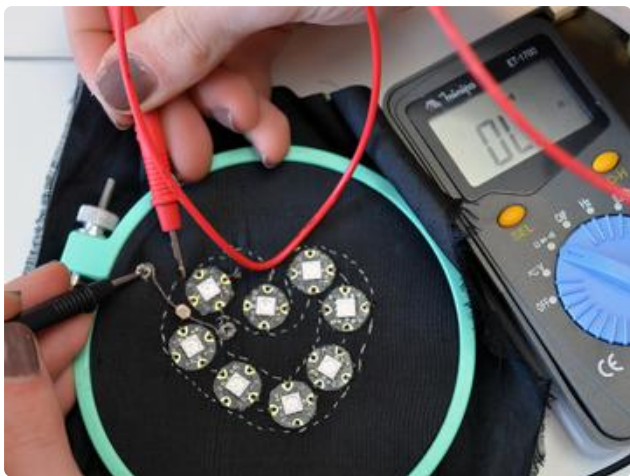


Stitch one spiral to the GND line around the inner perimeter of the heart (all GNDs are connected), and the other to D0 on GEMMA.



Toe knots and seal as before, then clip tails.





Test your new components' connections to GEMMA with a multimeter. If everything checks out, plug in your battery and pat yourself on the back!

---

## Wear it!

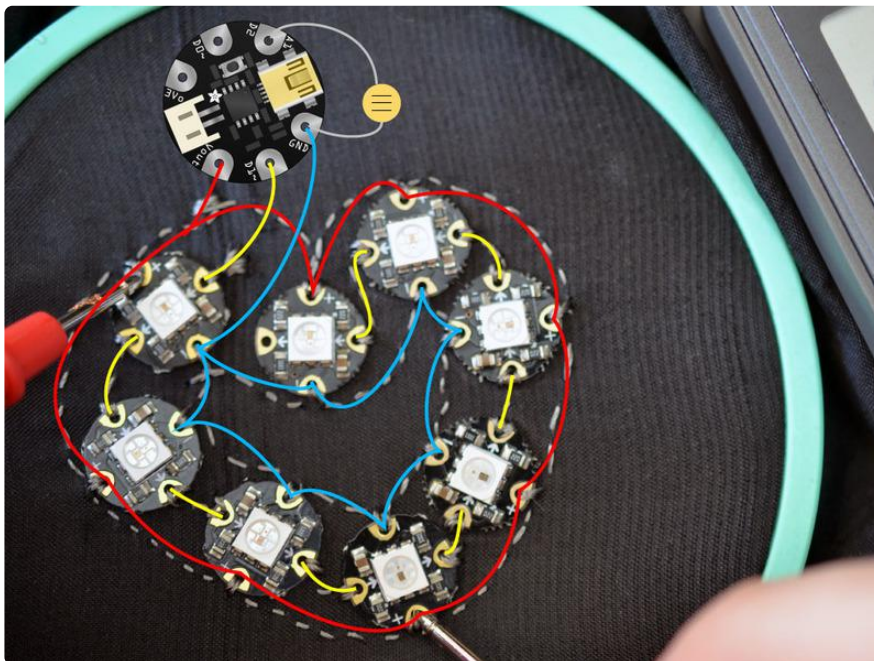
Take your circuit out of the embroidery hoop and finish the edge however you like. We sewed on a backing piece of fabric to enclose the circuit, and attached a pinback to wear it as a brooch.

Under a jacket the circuit remains off, then when you reveal it, the heart animates on!





## Circuit Diagram



NeoPixels are chained together with data input coming from GEMMA D1, all + to Vout and all - to GND.

Photoresistor is attached to D2 and GND.