# Table of Contents

- **Overview** 3
- **Tools & Supplies** 4
- **Circuit Diagram** 8
- **Solder Components** 9
- **Arduino Code** 10
  - NeoPixel Überguide: Arduino Library Installation
- **CircuitPython Code** 12
- **Affix Jewelry Findings** 14
- **Wear 'em!** 17
Overview

GEMMA jewelry! The bitty board fits perfectly in the center of a NeoPixel ring for flashy hoop earrings or a charming pendant. Read on to build your own!

Before you get started, follow the Introducing GEMMA guide or Introducing Gemma M0 guide.

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!
Tools & Supplies

Bill of materials (for one pendant-- double this for a pair of earrings):

- Gemma M0 (or Original Gemma) (http://adafruit.it/1222) (M0 type is recommended!)
- NeoPixel ring (http://adafruit.it/1463)
- tiny lipoly battery (http://adafruit.it/1317) with charger (http://adafruit.it/1304)
- pendant hanger or ear wires/jump rings
- stranded wire
- clear thread
- double stick tape
- E6000 adhesive (if using glue-on pendant hanger)

Any entry level 'all-in-one' soldering iron that you might find at your local hardware store should work. As with most things in life, you get what you pay for. Upgrading to a higher end soldering iron setup, like the Hakko FX-888 that we stock in our store (http://adafruit.it/180), will make soldering fun and easy.

Do not use a "ColdHeat" soldering iron! They are not suitable for delicate electronics work and can damage the boards (see here ()).

Click here to buy our entry level adjustable 30W 110V soldering iron. (http://adafruit.it/180)

Click here to upgrade to a Genuine Hakko FX-888 adjustable temperature soldering iron. (http://adafruit.it/303)

Learn how to solder with tons of tutorials! ()
You will want rosin core, 60/40 solder. Good solder is a good thing. Bad solder leads to bridging and cold solder joints which can be tough to find.

Click here to buy a spool of leaded solder (recommended for beginners). (http://adafru.it/145)

Click here to buy a spool of lead-free solder. (http://adafru.it/734)

You will need a good quality basic multimeter that can measure voltage and continuity.

Click here to buy a basic multimeter. (http://adafru.it/71)

Click here to buy a top of the line multimeter. (http://adafru.it/308)

Click here to buy a pocket multimeter. (http://adafru.it/850)

Don't forget to learn how to use your multimeter too! ()

Sharp scissors are a must!
Don’t forget your wire strippers (http://adafruit.it/527), pliers (http://adafruit.it/146), and flush snips (http://adafruit.it/152)! Tweezers (http://adafruit.it/421) can help manipulate the wires connecting components in your circuit.
Use needles (http://adafru.it/615) to secure threads to the circuit.

A helping third hand tool really makes this project a joy to build.

Click here to buy a helping third hand tool. (http://adafru.it/291)
This diagram uses the original Gemma but you can also use the Gemma M0 with the exact same wiring!

The NeoPixel ring's Data In pin connects to D0 on GEMMA, GND to Gnd, and Vcc to Vout.
Solder Components

Solder stranded wires to three spots on the NeoPixel ring according to the circuit diagram: power, ground, and data input.

Arrange GEMMA in the center of the ring, holding everything in place with a pair of helping hands.

Cut to length and strip the ends of the wires in order to connect to GND, Vout, and D0" on GEMMA (referencing the circuit diagram). Tweezers can help get tiny wires in position.
Flip the assembly over and solder the wires to the back of the GEMMA pads. Be careful not to fill in the hold with solder, so that you can still thread a needle through it later.

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:

```cpp
#include <Adafruit_NeoPixel.h>
#define PIN       0
#define NUM_LEDS 16

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUM_LEDS, PIN);

uint8_t  mode   = 0,        // Current animation effect
          offset = 0;        // Position of spinner animation
uint32_t color  = 0xFF8000; // Starting color = amber
uint32_t prevTime;          // Time of last animation mode switch

void setup() {
    pixels.begin();
    pixels.setBrightness(60); // ~1/3 brightness
    prevTime = millis();      // Starting time
}

void loop() {
    uint8_t  i;
    uint32_t t;

    switch(mode) {

    case 0: // Random sparkles - just one LED on at a time!
        i = random(NUM_LEDS);           // Choose a random pixel
        pixels.setPixelColor(i, color); // Set it to current color
        pixels.show();                  // Refresh LED states
        // Set same pixel to "off" color now but DON'T refresh...
        // it stays on for now...both this and the next random
        // pixel will be refreshed on the next pass.
        pixels.setPixelColor(i, 0);
        delay(10);                      // 10 millisecond delay
        break;

    case 1: // Spinny wheel (4 LEDs on at a time)
        for(i=0; i<NUM_LEDS; i++) {    // For each LED...
            uint32_t c = 0;           // Assume pixel will be "off" color
            if(((offset + i) & 7) < 2) { // For each 8 pixels, 2 will be...
                c = color;                 // ...assigned the current color
            }
            pixels.setPixelColor(i, c); // Set color of pixel 'i'
        }
        pixels.show();                // Refresh LED states
        delay(50);                    // 50 millisecond delay
        offset++;                     // Shift animation by 1 pixel on next frame
        break;

    // More animation modes could be added here!
    }
```

©Adafruit Industries
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.

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 “code.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).

This project is designed to work with RGB NeoPixel rings, not RGBW. The code will not work with RGBW.

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

```python
# SPDX-FileCopyrightText: 2017 Phillip Burgess for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# NeoPixel earrings example. Makes a nice blinky display with just a few LEDs on at any time...uses MUCH less juice than rainbow display!

import time
from rainbowio import colorwheel
import board
import neopixel
import adafruit_dotstar

try:
    import urandom as random  # for v1.0 API support
except ImportError:
    import random

dot = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1, brightness=0.2)
dot[0] = (0, 0, 0)

numpix = 16  # Number of NeoPixels (e.g. 16-pixel ring)
pixpin = board.D0  # Pin where NeoPixels are connected
strip = neopixel.NeoPixel(pixpin, numpix, brightness=.3, auto_write=False)

mode = 0  # Current animation effect
offset = 0  # Position of spinner animation
hue = 0  # Starting hue
color = colorwheel(hue & 255)  # hue -> RGB color
prevtime = time.monotonic()  # Time of last animation mode switch

while True:  # Loop forever...
    if mode == 0:  # Random sparkles - lights just one LED at a time
        i = random.randint(0, numpix - 1)  # Choose random pixel
        strip[i] = color  # Set it to current color
        strip.show()  # Refresh LED states
        # Set same pixel to "off" color now but DON'T refresh...
        # it stays on for now...but this and the next random
        # pixel will be refreshed on the next pass.
        strip[i] = [0, 0, 0]
        time.sleep(0.008)  # 8 millisecond delay
    elif mode == 1:  # Spinnny colorwheel (4 LEDs on at a time)
        for i in range(numpix):  # For each LED...
            if ((offset + i) & 7) < 2:  # 2 pixels out of 8...
                strip[i] = color  # are set to current color
            else:
                strip[i] = [0, 0, 0]  # other pixels are off
        strip.show()  # Refresh LED states
        time.sleep(0.04)  # 40 millisecond delay
        offset += 1  # Shift animation by 1 pixel on next frame
        if offset >= 8:
            offset = 0

    mode = 1  # Additional animation modes could be added here!
```

©Adafruit Industries
```
t = time.monotonic()  # Current time in seconds
if (t - prevtime) >= 8:  # Every 8 seconds...
    mode += 1  # Advance to next mode
    if mode > 1:  # End of modes?
        mode = 0  # Start over from beginning
        hue += 80  # And change color
        color = colorwheel(hue & 255)
        strip.fill([0, 0, 0])  # Turn off all pixels
        prevtime = t  # Record time of last mode change
```

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](https://github.com/adafruit/Adafruit_CircuitPython_NeoPixel).

---

**Affix Jewelry Findings**

You shouldn’t rely solely on the wires to hold GEMMA in place. Secure it in four spots with clear thread-- we’re using purple so you can see it better.

Thread a needle and pass it through a hole on GEMMA.
Tie the thread in a knot around the NeoPixel ring, aligning the thread between pixels. Do this in four spots around GEMMA to secure it in the center of the earring.
You can either glue a pendant hanger on the back with E6000 adhesive (hot glue is INSUFFICIENT), or attach an ear wire with a jump ring and two pairs of pliers.
Our tiny li-poly battery can be affixed to the back with double-stick tape. Secure it further with more clear thread if desired.

Wear 'em!

Enjoy your precious jewels! Each circuit weighs a measly 11.39 grams! Keep them out of the rain, and switch off when not in use.

The first version of the GEMMA board did not include a power switch — if using one of these, you’ll need to unplug the battery to switch the circuit off.