Superhero Power Plant
Created by Becky Stern

https://learn.adafruit.com/superhero-power-plant

Last updated on 2021-11-15 06:04:54 PM EST
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Laser it</td>
<td>5</td>
</tr>
<tr>
<td>Circuit Diagram</td>
<td>6</td>
</tr>
<tr>
<td>Build Circuit</td>
<td>8</td>
</tr>
<tr>
<td>Arduino Code</td>
<td>10</td>
</tr>
<tr>
<td>• NeoPixel Überguide: Arduino Library Installation</td>
<td>10</td>
</tr>
<tr>
<td>CircuitPython Code</td>
<td>12</td>
</tr>
<tr>
<td>Affix Faceplate</td>
<td>14</td>
</tr>
<tr>
<td>Wear it!</td>
<td>18</td>
</tr>
</tbody>
</table>
Overview

Planning an epic Iron Man costume for Halloween or Comic Con, or looking for that iconic piece that turns a plain t-shirt into Tony Stark? Look no further, for in this guide we’ll show you how to make your own electronic glowing reactor with a cool pulsing effect.

You can even customize it once complete, go for red, purple, green, pink: Whatever color will power you up! Or change the pulse rate or effects to add a special touch.

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!

This is a simple soldering+crafts project, but before you begin, please review the following prerequisite guides:

- [Introducing GEMMA](https://adafru.it/cLp) or [Introducing Gemma M0](https://adafru.it/yeq)
- [Adafruit NeoPixel Überguide](https://adafru.it/cEz)

Vector designs by Dano Wall. Vice photo by John de Cristofaro.

Parts list:

- [GEMMA v2](http://adafru.it/1222) or [GEMMA M0](https://adafru.it/ytb) wearable microcontroller
- [NeoPixel ring](http://adafru.it/1463)
- single [FLORA NeoPixel](http://adafru.it/1260)
- [solid-core hookup wire](http://adafru.it/1311)
- [JST extension cable](http://adafru.it/1131)
- [3xAAA battery pack](http://adafru.it/727) with [batteries](http://adafru.it/617)
- 2x laser-cut/etched acrylic in clear or white ([files on Thingiverse](https://adafru.it/cLn))
- elastic
- safety pin or velcro
- E6000 craft glue or hot melt glue
- needle and thread
- tracing/printer paper
Tools:

- [soldering iron](http://adafruit.it/180) & [solder](http://adafruit.it/145)
- [helping third hand tool](http://adafruit.it/291)
- [pliers](http://adafruit.it/146)
- [wire strippers](http://adafruit.it/527)
- [flush snips](http://adafruit.it/152)
- scissors
Laser it

Download the [vector files for this project from Thingiverse (https://adafruit.it/cLn)](https://adafruit.it/cLn) (Iron Man 3-inspired version available, too). Inking the engraved portion with a dry erase marker really brings out the detail. We layer two of these together for a neat 3D effect then later we'll wrap wire through the provided holes and indents to hold them together.
Hairline features are vector cuts, everything else is engraved.

Our settings, using a 60 Watt Epilog:
Raster speed 50%, raster power 50%, vector speed 40%, vector power 100%, vector frequency 5000
Have fun!

Circuit Diagram

This diagram uses the original Gemma but you can also use the Gemma v2 or M0 with the exact same wiring!
GEMMA Vout -> NeoPixel Ring Vcc and single FLORA pixel +
GEMMA D1 -> NeoPixel Ring Data IN
GEMMA GND -> NeoPixel Ring Gnd and single FLORA pixel -
NeoPixel Ring Data OUT -> single FLORA pixel inward-facing arrow
Build Circuit

Using solid-core wire for stiffness, solder short wires to all four junctions on the NeoPixel ring. Orient all wires to stick down through the pixel ring, and solder its output to the inward-facing arrow on a FLORA NeoPixel.

This center pixel will sit back in space more than the pixel ring, adding a bit more space for diffusion behind the faceplate.
Carefully solder the rest of the wire connections according to the circuit diagram (https://adafruit.it/cLo). The single pixel will sit back to back with GEMMA, not quite on the same plane as the pixel ring.
If this is your first time using GEMMA, work through the Introducing GEMMA (original) (https://adafru.it/cHH) or Introducing Gemma M0 guide (https://adafru.it/yeq) 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 (https://adafru.it/nBF)

Plug in your circuit via USB and test that all LEDs are functioning properly with the NeoPixel example sketch 'strandtest.' Please refer to Introducing GEMMA (https://adafru.it/cHH) and the NeoPixel Überguide (https://adafru.it/cEz) if you haven't before.

Once you've verified your wiring is correct, load your desired color code or the sketch below that gently pulses the LEDs blue.

```cpp
//Superhero Power Plant
//fades all pixels subtly
//code by Tony Sherwood for Adafruit Industries
#include <Adafruit_NeoPixel.h>
#define PIN 1
```
// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
// NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
// NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
// NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
// NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(17, PIN, NEO_GRB + NEO_KHZ800);

int alpha; // Current value of the pixels
int dir = 1; // Direction of the pixels... 1 = getting brighter, 0 = getting dimmer
int flip; // Randomly flip the direction every once in a while
int minAlpha = 25; // Min value of brightness
int maxAlpha = 100; // Max value of brightness
int alphaDelta = 5; // Delta of brightness between times through the loop

void setup() {
    strip.begin();
    strip.show(); // Initialize all pixels to 'off'
}

void loop() {
    flip = random(32);
    if(flip > 20) {
        dir = 1 - dir;
    }
    // Some example procedures showing how to display to the pixels:
    if (dir == 1) {
        alpha += alphaDelta;
    } else if (dir == 0) {
        alpha -= alphaDelta;
    }
    if (alpha < minAlpha) {
        alpha = minAlpha;
        dir = 1;
    }
    if (alpha > maxAlpha) {
        alpha = maxAlpha;
        dir = 0;
    }
    // Change the line below to alter the color of the lights
    // The numbers represent the Red, Green, and Blue values
    // of the lights, as a value between 0(off) and 1(max brightness)
    // EX:
    // colorWipe(strip.Color(alpha, 0, alpha/2)); // Pink
    colorWipe(strip.Color(0, 0, alpha)); // Blue
}

// Fill the dots one after the other with a color
void colorWipe(uint32_t c) {
    for(uint16_t i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
    }
}
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://adafruit.it/z1B).

Below is CircuitPython code that works similarly to 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).

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

```python
import board
import neopixel

try:
    import urandom as random  # for v1.0 API support
except ImportError:
    import random
```
num_pix = 17  # Number of NeoPixels
pix_pin = board.D1  # Pin where NeoPixels are connected
strip = neopixel.NeoPixel(pix_pin, num_pix)

min_alpha = 0.1  # Minimum brightness
max_alpha = 0.4  # Maximum brightness
alpha = (min_alpha + max_alpha) / 2  # Start in middle
alpha_delta = 0.01  # Amount to change brightness each time through loop
alpha_up = True  # If True, brightness increasing, else decreasing

strip.fill([0, 0, 255])  # Fill blue, or change to R,G,B of your liking

while True:  # Loop forever...
    if random.randint(1, 5) == 5:  # 1-in-5 random chance
        alpha_up = not alpha_up  # of reversing direction
    if alpha_up:  # Increasing brightness?
        alpha += alpha_delta  # Add some amount
        if alpha >= max_alpha:  # At or above max?
            alpha = max_alpha  # Limit to max
            alpha_up = False  # and switch direction
    else:  # Else decreasing brightness
        alpha -= alpha_delta  # Subtract some amount
        if alpha <= min_alpha:  # At or below min?
            alpha = min_alpha  # Limit to min
            alpha_up = True  # and switch direction

    strip.brightness = alpha  # Set brightness to 0.0 to 1.0
    strip.write()  # and issue data to LED strip

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://adafruit.it/yew).
Affix Faceplate

Strip your copper wire and insert through one set of slots in the stacked acrylic pieces. Hide the ends to the back side of the faceplate and use pliers to squeeze the wires into a tight fit. Our thick wire couldn’t be wrapped multiple times around, so we cut each loop at the back and started with a fresh loop. The wire ends will be hidden so the back doesn’t have to look pretty!
Continue this process to place five or six loops in each slot until they are all filed with faux "coils."
The arc reactor in Iron Man 3 has a different design without wires. We used white acrylic and a blue dry-erase marker in the etchings to achieve the above effect.

Dab some E6000 adhesive on the back of the coils, and gently glue the circuit assembly to it. Don't press so hard that the metal wire comes in contact with the pixel ring PCB-- the glue acts as an insulator. E6000 dries clear and flexible in about 24 hours.
If you're in a rush you can use hot melt glue. It's inferior to E6000 in almost every way, but it will get the job done fast.
Wear it!

Thread a piece of elastic long enough to go around your chest through the circuit, between GEMMA and the NeoPixel Ring.

To keep it from obscuring the center pixel, cinch the elastic in one spot with a few stitches, then align that spot with the circuit.

Use a safety pin or velcro to secure the strap.
Use a JST extension cable (http://adafruit.it/1131) and 3xAAA battery pack (http://adafruit.it/727) tucked in your pocket for many hours of glow time. We experimented with using a coincell battery pack (http://adafruit.it/783) as well, but it only lasted for about an hour.