# Table of Contents

**Introduction**
- Parts Needed 4
- Also Needed 4
- Tools 5

**Wiring Diagram** 6

**Software** 7

**3d Printing the Horn**
- Modifications 10
- Slicer Settings 11

**Electronics Assembly**
- Battery Charger 14

**Final Assembly** 19

**Calibration** 23
Introduction

Legends agree that Unicorn horns have magical healing and purification properties. Unicorns do this naturally -- as soon as their horn encounters scummy pond water or a festering wound, the horn's magic kicks in and does its thing. How cool is Nature?

For a cosplay Unicorn, it's not quite as intuitive to invoke the horn's magic. This horn solves that problem with another kind of magic: capacitive touch.

Decorate the horn with a beautiful vein of copper and let the purification process begin with a simple touch. The horn glows with color shifting rainbows all the time, and when someone touches the copper vein, the rainbow will fade and the horn's tip will glow a bright white, purifying everything in its vicinity. Like magic!
## Parts Needed

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x Gemma M0</td>
<td>Gemma M0 Microcontroller</td>
<td><a href="https://www.adafruit.com/product/3501">https://www.adafruit.com/product/3501</a></td>
</tr>
<tr>
<td>1 x Neopixel Ring</td>
<td>12 Neopixel Ring</td>
<td><a href="https://www.adafruit.com/product/1643">https://www.adafruit.com/product/1643</a></td>
</tr>
<tr>
<td>1 x Battery</td>
<td>LiPoly 350mah Battery</td>
<td><a href="https://www.adafruit.com/product/2750">https://www.adafruit.com/product/2750</a></td>
</tr>
<tr>
<td>1 x Battery Charger</td>
<td>USB Charger</td>
<td><a href="https://www.adafruit.com/product/1304">https://www.adafruit.com/product/1304</a></td>
</tr>
<tr>
<td>1 x JST Connector</td>
<td>Battery Charger connector</td>
<td><a href="https://www.adafruit.com/product/261">https://www.adafruit.com/product/261</a></td>
</tr>
<tr>
<td>1 x White LED</td>
<td>Diffused white 3mm LED</td>
<td><a href="https://www.adafruit.com/product/778">https://www.adafruit.com/product/778</a></td>
</tr>
</tbody>
</table>

## Also Needed

- 1 x 51Ω resistor (or similar)
- Copper foil tape [https://adafruit.it/Cje](https://adafruit.it/Cje) in your desired width
- Heat shrink tubing
You will also need a unicorn horn -- you can 3d print one or make your own out of worbla (https://adafru.it/Cjh), resin, or any other translucent material. Marlinspike Auger seashells (https://adafru.it/Cjm) make great unicorn horns too. The sky's the limit!

This guide includes a printable unicorn horn file as well as a printable mounting plate, so you can remove the horn for switching it on/off and battery charging.

**Tools**

- 3d Printer (optional)
- Soldering iron & accessories
- Utility knife
- Needle & thread for mounting
Wiring Diagram

This project won't work with a 'classic' Gemma because we use the Gemma M0's built-in capacitive touch!

<table>
<thead>
<tr>
<th>Gemma 3V</th>
<th>Neopixel Ring 5V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gemma A0</td>
<td>Neopixel Ring IN</td>
</tr>
<tr>
<td>Gemma G</td>
<td>Neopixel Ring G</td>
</tr>
<tr>
<td>Gemma A2</td>
<td>51 ohm resistor</td>
</tr>
<tr>
<td>Gemma A1</td>
<td>Copper Tape</td>
</tr>
</tbody>
</table>
Battery Charger

To add charging capability, splice a second JST connector inline with the battery's wires, and plug in a USB charger.

Why solder NeoPixels to the 3V pad instead of the Vout pad?

The Gemma M0’s Vout pad could give us 3.7V straight from the battery or 5V from USB, and the NeoPixels are brightest at these higher voltages. However, we've designed this project to leave the battery plugged in at all times, and even with the Gemma switched off, anything connected to the Vout pad will slowly drain the battery. Wiring to 3V will fix this problem!

Since we've only got 12 pixels, the 3V pin will give us plenty of power. Don't use this trick if you're trying to power a whole lot of NeoPixels! The Gemma’s 3V regulator won't handle a larger strand very well.

Software

It's a great idea to get your software all set up and loaded onto your board right away, to make testing your connections easier later on.

To get the code running you'll need:

1. Arduino IDE (1.8 or newer preferred)
1. Arduino IDE

If you're not using a recent version of the Arduino IDE (1.8.5 or newer), this would be a good time to upgrade (https://adafru.it/fvm). If this is your first time using Arduino, head over to this guide to get it installed (https://adafru.it/jDQ). It's free and fairly simple to get set up.

2. Board Support

You'll need to tell the Arduino IDE which board you're using. This takes just a few minutes of setup, and you'll only need to do it once.

Here is a step-by-step tutorial for setting up Gemma M0 (https://adafru.it/Cib)

3. Libraries

All three libraries can be installed using the Arduino Library Manager — use Sketch→Include Library→Manage Libraries... and search for all or part of the library's name, then click “Install.”

Look for:

- FastLED
- Adafruit_FreeTouch
- Adafruit_NeoPixel

Adafruit_NeoPixel isn't absolutely required for this project, but it's handy to have installed in case you have problems with FastLED. Troubleshooting the basics is a little easier with Adafruit_NeoPixel.

Upload the Code

Plug your microcontroller into your computer with a USB cable. In the Arduino IDE, go to Tools > Boards and select the name of the board. Then go to Tools > Port and select the board there too. (If it's not showing up there, be sure your microcontroller is plugged into your computer via USB)
#include "Adafruit_FreeTouch.h"
#include "FastLED.h"

#define CAPTOUCH_PIN A1
#define NEOPIXEL_PIN 1
#define LED_PIN 0
#define NUM_LEDS 12
#define LED_TYPE WS2812
#define COLOR_ORDER GRB

CRGB leds[NUM_LEDS];

int BRIGHTNESS=150;
int touch = 500; // Change this variable to something between your capacitive touch serial readouts for on and off

long oldState = 0;
int gHue=0;

Adafruit_FreeTouch qt_1 = Adafruit_FreeTouch(CAPTOUCH_PIN, OVERSAMPLE_4, RESISTOR_50K, FREQ_MODE_NONE);
//Adafruit_FreeTouch qt_2 = Adafruit_FreeTouch(A2, OVERSAMPLE_4, RESISTOR_50K, FREQ_MODE_NONE);

void setup() {
  Serial.begin(115200);
  if (!qt_1.begin())
    Serial.println("Failed to begin qt on pin A1");

  pinMode(LED_PIN, OUTPUT); //initialize the LED pin

  FastLED.addLeds<WS2812, NEOPIXEL_PIN, COLOR_ORDER>(leds, NUM_LEDS); // Set up neopixels with FastLED
  FastLED.setBrightness(BRIGHTNESS);
  FastLED.setMaxPowerInVoltsAndMilliamps(3,350); //Constrain FastLED's power usage
}

void loop() {
  Serial.print(qt_1.measure());
  Serial.write(' ');
  checkpress();
  delay(20);
}

void checkpress() {
  // Get current button state.

  long newState = qt_1.measure();
  Serial.println(qt_1.measure());
  if (newState > touch && oldState < touch) {
    // Short delay to debounce button.
    delay(20);
    // Check if button is still low after debounce.
    long newState = qt_1.measure();
  }

  if (newState > touch ) {
    dark();
    digitalWrite(LED_PIN, HIGH);
delay(20);
}

else {
  rainbow();
  digitalWrite(LED_PIN, LOW);
  delay(20);
}

// Set the last button state to the old state.
oldState = newState;

// do some periodic updates
EVERY_N_MILLISECONDS(20) { gHue++; } // slowly cycle the “base color” through the rainbow

// FastLED’s built-in rainbow generator
void rainbow() {
  fill_rainbow(leds, NUM_LEDS, gHue, 7);
  FastLED.show();
  delay(20);
}

void dark() {
  for(int i = 0; i < NUM_LEDS; i++) {
    leds[i] = CRGB::Black;
    FastLED.show();
    delay(20);
  }
}

3d Printing the Horn
I created this design in Fusion360, based on this fabulous tutorial by Noe Ruiz. I'm printing on a dual extruder Sigma BCN3D ([https://adafru.it/xFR](https://adafru.it/xFR)):

**Modifications**

Following the video will create a horn that's slightly too small to fit the neopixel ring inside. Start with a sketch that's around 42mm across instead of 38mm across to be sure it will fit.

I also added a 4mm x 13mm hole near the base for the USB charger, and added tabs right at the bottom for sewing or for attaching to the mounting plate.

Since the on/off switch is on the Gemma, and that might be hard to get to, I created a slide-in mounting plate that can be sewn to a headdress or hood. This way the horn is easily removable for charging or for turning on and off.

For non-dual extruder printers, here's an .stl file of the horn and spiral merged together.

- **horn.stl**
  - [https://adafru.it/Ax7](https://adafru.it/Ax7)

- **spiral.stl**
  - [https://adafru.it/Ax8](https://adafru.it/Ax8)

- **Platform.stl**
  - [https://adafru.it/Ax9](https://adafru.it/Ax9)

For non-dual extruder printers, here's an .stl file of the horn and spiral merged together.

- **horn_with_spiral.stl**
  - [https://adafru.it/Axa](https://adafru.it/Axa)
This horn was printed with dual extruders for two-color goodness. I printed the horn in glow-in-the-dark ABS and the spiral in purple ABS that changes to pink when it gets warm (https://adafru.it/Cjn). Magic!

To print the unicorn horn, the 3D printer will need to be capable of printing a minimum of 150mm tall, and the base of the horn is around 42x48mm, which should fit on most beds.

### Slicer Settings

Download and import the .fff files below into Simplify3D. These settings are for dual extruders, with 3mm ABS. The horn will print with the left extruder and the spiral with the right.

**Horn Profile:**
DualExtruder3mmABS.fff
https://adafru.it/Axb

**Platform Profile:** Platform_Settings.fff
https://adafru.it/Axc

**Horn Print Settings**
- 0% infill
- No supports needed
- No raft
• Skirt/brim can be helpful

Platform Print Settings

• 20% infill
• With suports
• With raft
• Skirt/brim can be helpful
Electronics Assembly

Cut a red and a black wire, slightly longer than your horn.

Grab a coin cell battery and test your LED. Make sure you know which is the anode leg (power) and which is the cathode (ground).

99+ percent of the time, the longer leg is the anode, but occasionally a “trickster” LED with two same-sized legs slips by. No harm in double-checking regardless.
Wind one of your resistor’s legs to the anode pin of your LED and solder it in place. Solder the red wire to the other resistor leg, and then solder your black wire to the cathode leg of the LED.

Test again with your coin cell to make sure the LED is still working. If it all looks good, slip some heat shrink over the connections to cover them and keep them safe from shorting.
Grab your neopixel ring. Solder a red wire to 5V, a black wire to G, and a white wire to IN.

Note: it's easiest to slip the wire in from the neopixel side, then make the solder joint on the back of the ring.

Pull the wires through to the inside of the ring and trim to about 2".

Twist the black wire from the neopixel ring together with the black wire coming from the LED and solder both into the G pin on your Gemma M0.

Solder the red wire from the neopixel ring to the 3V pin on the Gemma, and solder the white wire to pin D1.
Finally, solder the remaining red wire from the LED to pin D0 on the Gemma.

Plug your battery into the Gemma and be sure it's switched on. If you haven't done so yet, upload the code. Test to be sure your neopixels come on.

Touch pad A1 with your finger. The neopixels should go off and the LED should light up. Yay!

If your touch pad is too sensitive or doesn't seem to do anything, you may need to calibrate the code. We'll go over that in the next section.

Battery Charger

Cut and solder ONE wire at a time! DO NOT cut through both at once. LiPoly batteries can catch fire if treated incorrectly!
Trim your JST connector to about 2". Carefully cut the red wire to your LiPoly battery. Slip on some heat shrink and splice in the red wire from your JST connector.

Repeat with the black wire. You now have two plugs connected to the battery: one for the Gemma and one for the charger.
Final Assembly

Cut a piece of copper tape that's a little longer than you'll need. Trim it to shape and coil it around the horn. I found that coiling it right above the spiral works best -- that way when I'm wearing the horn I can find it easily with my fingers without looking.

Use a blunt tool to press and smooth the copper tape down. It doesn't want to spiral neatly so spend some time getting it to look nice. Use a dab of superglue at the tip to be sure it stays stuck.

Leave a long tail at the end of the tape for connecting to your Gemma.

Drop a little superglue in the tip of the horn and slide the LED in so it stays put.

Slide the battery, then the neopixel ring into the horn. Set the charger in place with the JST connector facing downwards towards the horn's base.

Then, attach the copper tape to Gemma's A1 pad. You can do this by slipping the tape through the hole and soldering, or if that's too fiddly, solder a short wire to the pad and the tape to make the connection.
Secure the charger with a dab of hot glue.

Tuck the Gemma inside the base of the horn, being careful to keep the capacitive touch pad away from any other electronics, as much as possible.

At this point, my horn went haywire. The capacitive touch was either way too sensitive, or not sensitive enough. Or, all the other pads on the Gemma would set it off but pad A1 did nothing.

I spent a good amount of time calibrating the software, testing, shielding, and reconfiguring the copper tape. The calibration here is pretty tricky. As soon as I unplugged the USB cable, the horn reacted totally differently -- without that ground from the USB port, the copper tape's sensitivity would freak out.

Don't give up! Check the calibration page in this guide for some tips.
To make a headband, measure around your head from your forehead to the nape of your neck and cut a piece of stiff fabric or vinyl to approximately that size.

Sew a pretty shiny fabric casing and attach a hair elastic to the ends for a nice stretch fit.
Sew the 3d printed platform base onto the center of the headband using the mounting holes.

Decorate your headband with jewels, flowers, seashells, beads, or whatever else strikes your fancy.
After everything is assembled, you may find that your horn is too sensitive: you get within a few inches and the LED tip comes on. Or it may be not sensitive enough and no amount of touching will make the horn's tip glow. Luckily this is easy to adjust in the code.

Plug your Gemma into your computer and be sure it's selected under Tools > Port > Adafruit Gemma M0.

Open the Serial monitor and move your hands away from the Gemma. You should see numbers scrolling by. This is your baseline number, so write it down.

Now touch the copper tape. This is your active number. Write this down too.

Go find this line in the code:

```c
int touch = 500; // Change this variable to something between your capacitive touch serial readouts for on and off
```
The "touch" variable is what sets the sensitivity. Change the number to something near the median point between your baseline number and your active number. Upload the code again and test to see if it's behaving better.

Adding the copper tape will change the active number wildly. The more tape you add, the greater the change, so re-calibrate when your horn is fully assembled for best results.

The sensitivity will also change in different situations. If the Gemma is plugged into your computer, your results will be different from when it's not plugged in. Since you can't get readouts when it's not plugged in, there will be some trial and error to get the horn calibrated just right.

The sensitivity will also change if it's on your head, not on your head, on the headdress or not on the headdress. Or if the wind is blowing. It can be a little frustrating! So be sure you're testing with the horn in its final location (on your head).

If all else fails, ground yourself. Take three deep breaths, wiggle your toes, and connect your chakras to the Universal Ground. Electricity runs through us all, and frustration can change your results. Breathe.

Then, go prance around like the Magical Sparkly Unicorn you are.