NeoPixel LED Mickey Ears

Created by Ruiz Brothers

https://learn.adafruit.com/neopixel-led-mickey-ears

Last updated on 2024-03-08 03:11:42 PM EST

Table of Contents

Overview 3

- Light Up Ears
- Flexible Neon NeoPixel Strips
- Shake to Sparkle
- Wearable Ear Hats
- Prerequisite Guides
- Parts List
- Tools

Circuit Diagram 10

MakeCode 11

- MakeCode for Adafruit Boards
- NeoPixel Mickey Ear Code
- GEMMA M0 MakeCode
- Pair Device and Upload
- Testing Code

What is MakeCode Maker? 14

Editing Blocks 16

- Blinky!

Downloading and Flashing 17

- Step 1: Connect your board via USB
- Step 2: Test your code in the simulator
- Step 3: Download and flash your code

Saving and Sharing 20

- Extracting your code from the board
- Sharing

Assemble 21

- Measure the Strip
- Expose Pads
- Prep the Hat
-
- Fit Strips into Slits
- Solder Strip Wires
- Solder Tilt Switch
- Thread and Strip Wires
- Solder the Gemma
- JST Slide Switch
- Heat Shrink
- Mount board
- Mount the Battery
- Glue Strips
- Complete!

Overview

Light Up Ears

In this project we'll show you how to make your own set of light up Mickey Ears.

We made it so they flash like fireworks when you shake your head.

In this learn guide, you'll see how to wire up the electronics to make your own set of NeoPixel Mickey Ears.

Flexible Neon NeoPixel Strips

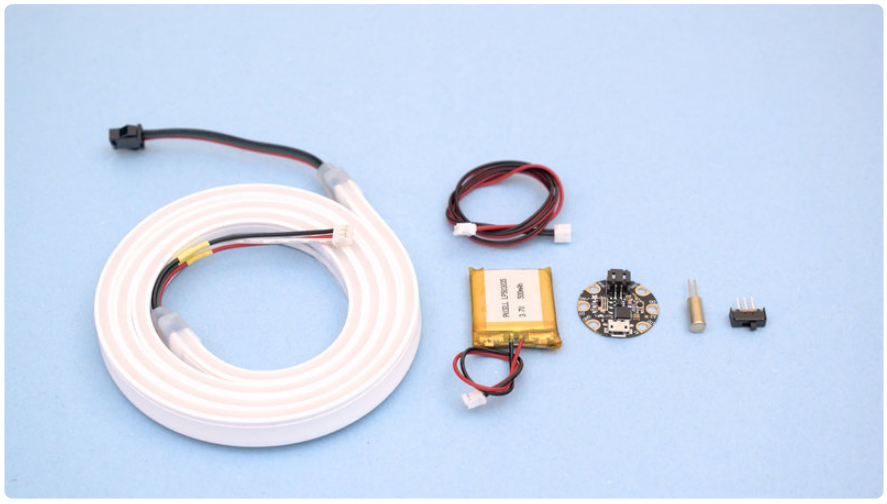
These are really bright LEDs that look a lot like neon lights. They're Adafruit NeoPixel Strips with extra thick silicone so the diffusion looks really nice. They're individually addressable, so you can create really nice animations.

Shake to Sparkle

Motion activated LED animations can be done in a number of different ways. In this guide, a tilt-switch is used to trigger LED animations. This build will use MakeCode to program the hardware!

Wearable Ear Hats

We got these mickey ears from Disney World, but you could use whatever you like. These mickey shorts style of ears feature pre-made holes just behind the ear cups.



Prerequisite Guides

There's resources in these guides that go beyond what's covered in this tutorial. The introduction guides walks you through all of the pinouts and setting up MakeCode with Adafruit's supported board.

- Intro to Adafruit GEMMA M0 (https://adafru.it/FJS)

Parts List

Here's a list of parts. You'll need to choose your NeoPixel LED strip – There's a few different densities and styles to pick from.

NeoPixel LED Strips (Pick one)

- Adafruit Flexible Neon Strip – https://www.adafruit.com/product/4310 (http://adafru.it/4310)

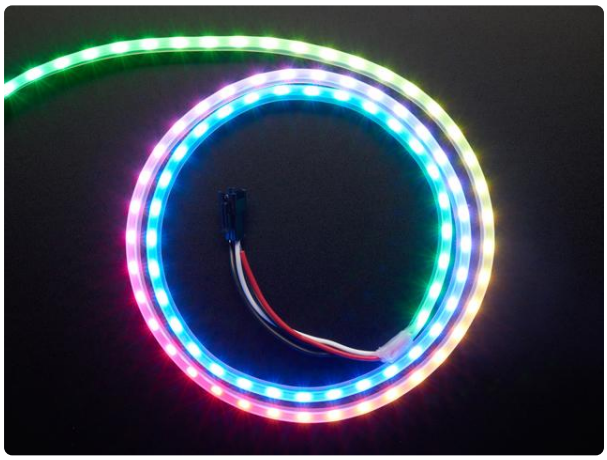
- Adafruit NeoPixel Side Light Strip 90 – https://www.adafruit.com/product/3635 (http://adafru.it/3635)
- Adafruit NeoPixel Side Light Strip 60 – https://www.adafruit.com/product/3636 (http://adafru.it/3636)

Flexible Silicone Neon-like Skinny NeoPixel LED Strip

You love NeoPixels, and you love silicone diffusion? Peep this Flexible Silicone Neon-like Skinny NeoPixel LED Strip! OK it's a bit of a mouthful, but check...
https://www.adafruit.com/product/4310

Adafruit NeoPixel LED Side Light Strip - Black 60 LED

Fancy new side light LED strips are a great alternative for folks who have loved and used Adafruit LED strips for a few years but want gorgeous, glowy light emitting at...
https://www.adafruit.com/product/3636



Adafruit NeoPixel LED Side Light Strip - Black 90 LED

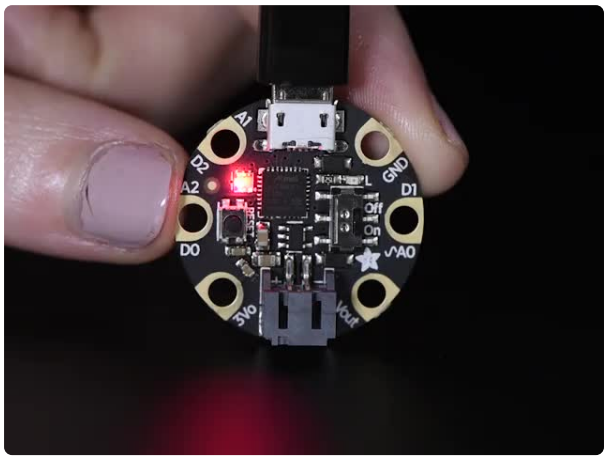
Fancy new side light LED strips are a great alternative for folks who have loved and used Adafruit LED strips for a few years but want gorgeous, glowy light emitting at...
https://www.adafruit.com/product/3635

Adafruit GEMMA M0 – https://www.adafruit.com/product/3501 (http://adafru.it/3501)

- Tilt Ball Switch – https://www.adafruit.com/product/173 (http://adafru.it/173)

Battery, Switch and Wires

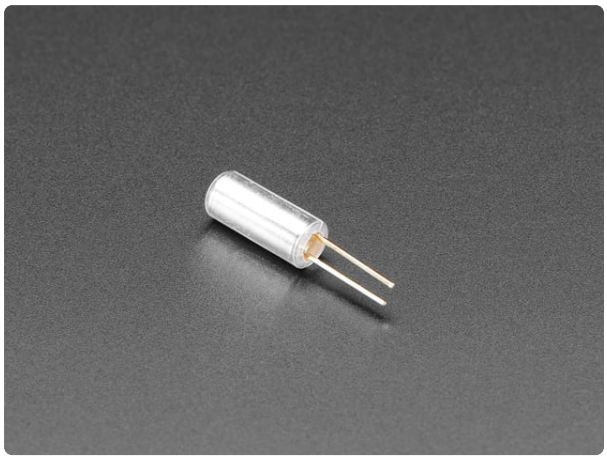
- Slide Switch – https://www.adafruit.com/product/805 (http://adafru.it/805)
- 500mAh Battery – https://www.adafruit.com/product/1578 (http://adafru.it/1578)
- JST Extension Cable – https://www.adafruit.com/product/1131 (http://adafru.it/1131)
- 10 Wire Silicone Cover Ribbon Cable – https://www.adafruit.com/product/3890 (http://adafru.it/3890)



Adafruit GEMMA M0 - Miniature wearable electronic platform

The Adafruit Gemma M0 is a super small microcontroller board, with just enough built-in to create many simple projects. It may look small and cute: round, about the...

https://www.adafruit.com/product/3501



Tilt ball switch

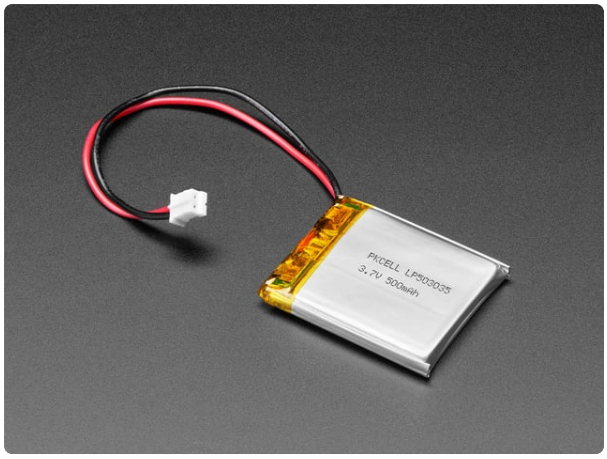
The "poor man's" accelerometer! Tilt sensors are switches that can detect basic motion/orientation. The metal tube has a little metal ball that rolls around in it, when...

https://www.adafruit.com/product/173

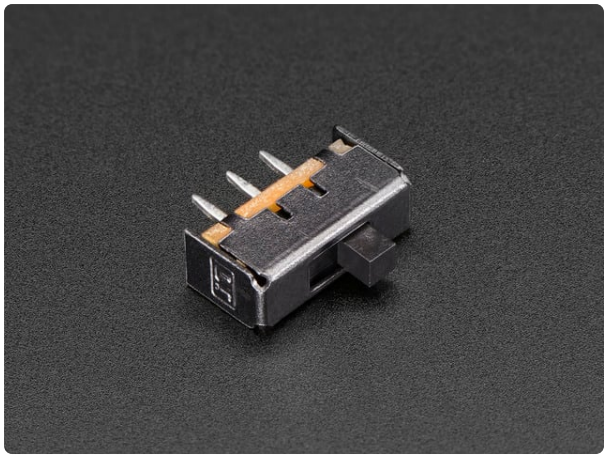
JST-PH Battery Extension Cable - 500mm

By popular demand, we now have a handy extension cord for all of our JST PH-terminated battery packs (such as our LiIon/LiPoly and 3xAAA holders). One end has a JST-PH compatible...

https://www.adafruit.com/product/1131

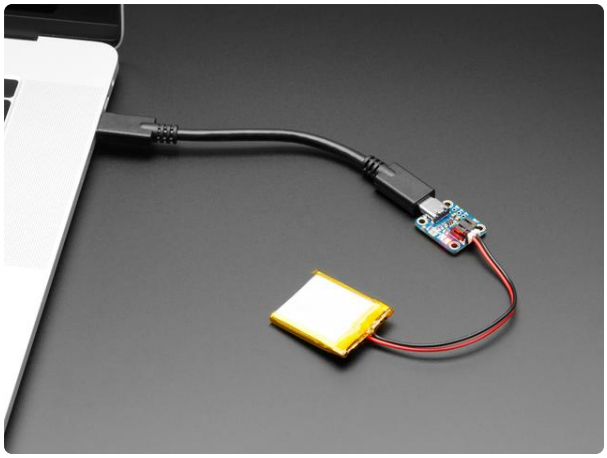


[Lithium Ion Polymer Battery - 3.7v 500mAh]
Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...
https://www.adafruit.com/product/1578



[Breadboard-friendly SPDT Slide Switch]
These nice switches are perfect for use with breadboard and perfboard projects. They have 0.1" spacing and snap in nicely into a solderless breadboard. They're easy to switch...
https://www.adafruit.com/product/805

[Silicone Cover Stranded-Core Ribbon Cable - 10 Wire 1 Meter Long]
For those who are fans of our silicone-covered wires, but are always looking to up their wiring game. We now have Silicone Cover Ribbon cables! These may look...
https://www.adafruit.com/product/3890



[Adafruit Micro-Lipo Charger for LiPoly Batt with USB Type C Jack]
Oh so handy, this little lipo charger is so small and easy to use you can keep it on your desk or mount it easily into any project! Simply plug it via any USB C cable into a USB port...
https://www.adafruit.com/product/4410

Tools

Slice Craft Knife with Ceramic Blade
Discontinued - you can grab Canary Stainless Steel Non-Stick Cardboard Box Cutter
https://www.adafruit.com/product/4306

Helping Third Hand Magnifier W/ Magnifying Glass Tool
The classic 'third hand tool,' as seen on every desk! We have one next to our Panavise jr, they complement each other well. This...
https://www.adafruit.com/product/291

Panavise Jr.
The best mini-vise for working on smaller PCBs (2.875" or less in one dimension)! The jaws are strong plastic that wont damage the PCB and doesn't mind if you hit it with the...
https://www.adafruit.com/product/151

Multi-Colored Heat Shrink Pack - 3/32" + 1/8" + 3/16" Diameters

Heat shrink is the duct tape of electronics which I guess makes this heat shrink the colorful and exciting duct tape they sell at craft stores. This heat shrink comes in six...

https://www.adafruit.com/product/1649

Hakko Professional Quality 20-30 AWG Wire Strippers

These are the finest wire strippers we have used, and if you have to do a lot of wiring, you will agree! They have soft rounded grips - very comfortable to use, and precision ground...

https://www.adafruit.com/product/527

Flush diagonal cutters

These are the best diagonal cutters, large super-comfortable grip to use and have strong nippers for perfect trimming of wires and leads. I've used my pair every day for years.

https://www.adafruit.com/product/152

Digital Genuine Hakko FX-888D (936 upgrade)

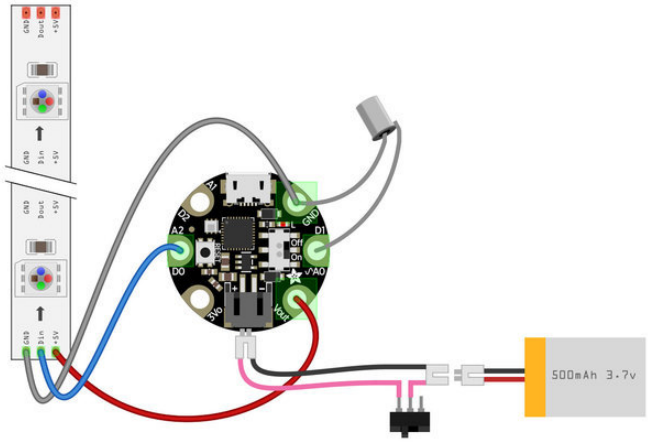
Known by engineers for making excellent quality tools & soldering irons! This is a genuine Hakko FX-888D with digital temperature control! We worked hard to get...

https://www.adafruit.com/product/1204

Circuit Diagram

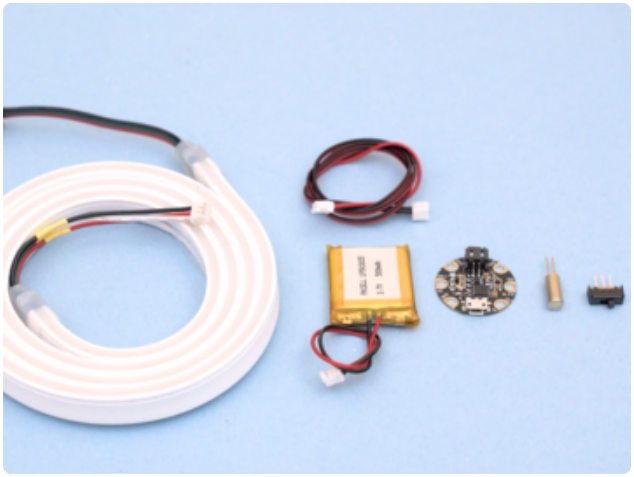
The wiring diagram below provides a visual reference for connecting the components. It is not true to scale, it is just meant to be used as reference. This diagrams was created using the Fritzing software package (https://adafru.it/oEP).

Take a moment to review the components in the circuit diagram. This illustration is meant for referencing wired connections - the length of wire, position and size of components are not exact.



Wires are measured and cut to have enough slack to reach each component.

Silicone ribbon wire is used to make them easier to coil and manage each wire inside the tight spacing. The wire lengths used are listed below:



LED strip for Ears: 310mm (remove 1.75mm of sheath)

Gemma to LED Strip connector: 90mm

Strip to Strip Connection: 160mm

JST Extension: 240mm

Add Slide Switch about 60mm from the female connector

MakeCode

MakeCode for Adafruit Boards

MakeCode Maker is an easy-to-use block-based programming editor that runs in the Google Chrome web browser. It's has an intuitive interface and features lots of examples and demos.

The Maker version works with supported Adafruit boards, so you can make interactive projects with a wide selection of microcontroller boards.

NeoPixel Mickey Ear Code

Click the links below to launch the code in maker.makecode.com. The GEMMA M0 version uses the tilt-ball switch.

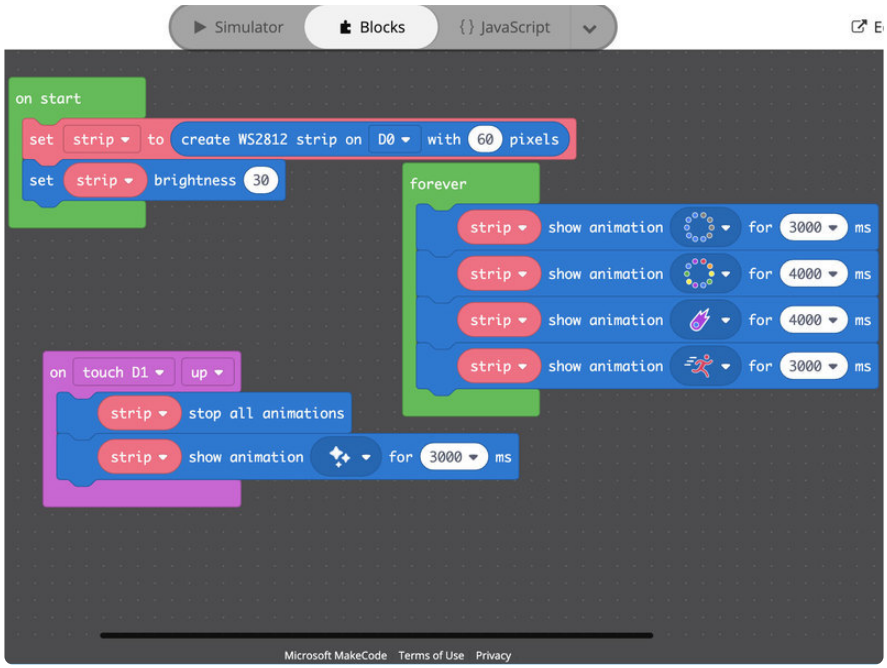
Edit Gemma Code

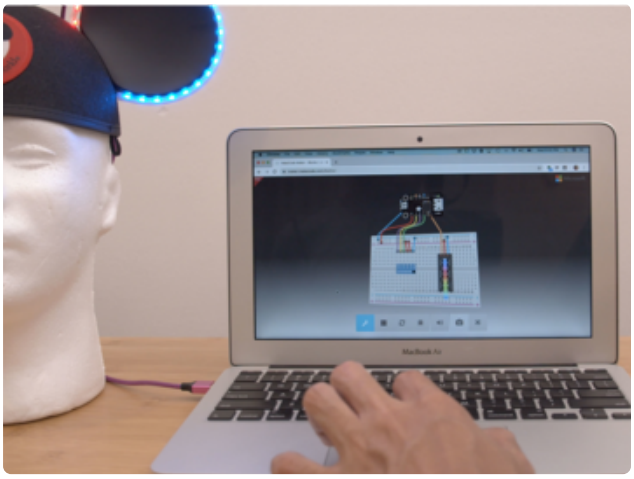
https://adafru.it/FK2

GEMMA M0 MakeCode

A NeoPixel strip is created with the **set strip** block. It's set to pin **D0** with **60** pixels. The **strip brightness** block tells the strip how bright to glow on start. An **on touch** block is used to detect when a pin is triggered.

When the tilt-ball switch is triggered on pin **D1**, the LED flash animation will run for a few seconds. In the **forever** loop, several light blocks are setup to cycle through animations for a few seconds each.





Pair Device and Upload

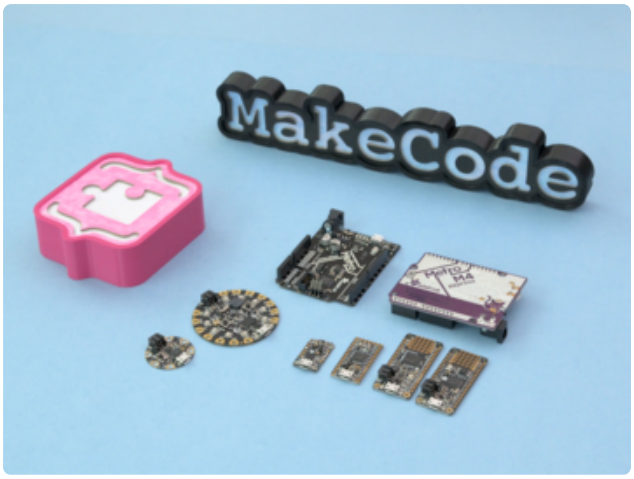
Once you have the MakeCode program open, click the **edit** button on the top right to open it in a new MakeCode project.

The GEMMA M0, code blocks and simulator will be ready to modify. Let's get the device paired with WebUSB.

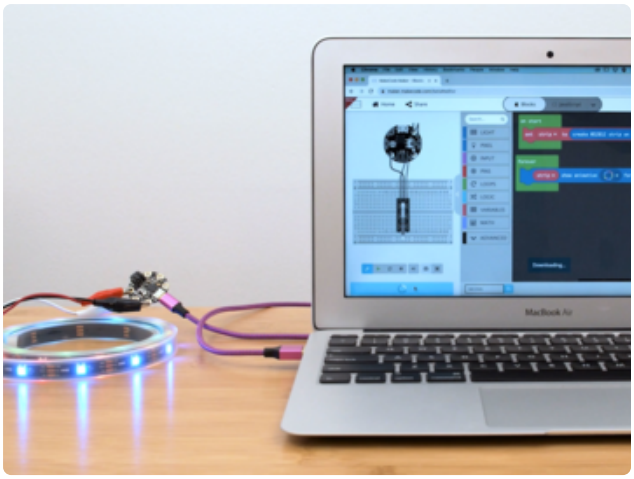
1. Click on the **Gear icon**, at the top right.
2. Select **Pair Device** from the dropdown.
3. Click on **GEMMA M0** and **Connect.**
4. Click the big blue **Download** button to upload the code.

Testing Code

The NeoPixel LED strips and tilt-ball switch should be wired up to the GEMMA M0. If you haven't already, take a moment to do so now. Shaking the tilt-ball should trigger the flashing LED animation. The rainbow and chasing animations are programmed to loop until touch is detected on Pin **D1**.

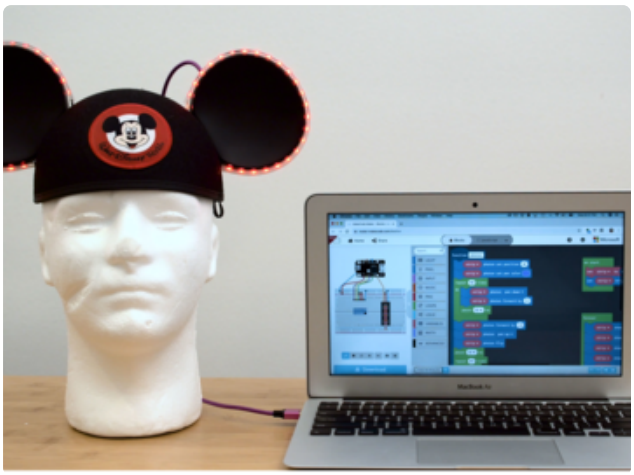


The Maker version of MakeCode (https://adafru.it/C9N) supports lots of boards from Adafruit and the open source community. In your MakeCode program, click on the **gear** icon and select **change board**. Take a moment to browse the support boards, there's a great selection!



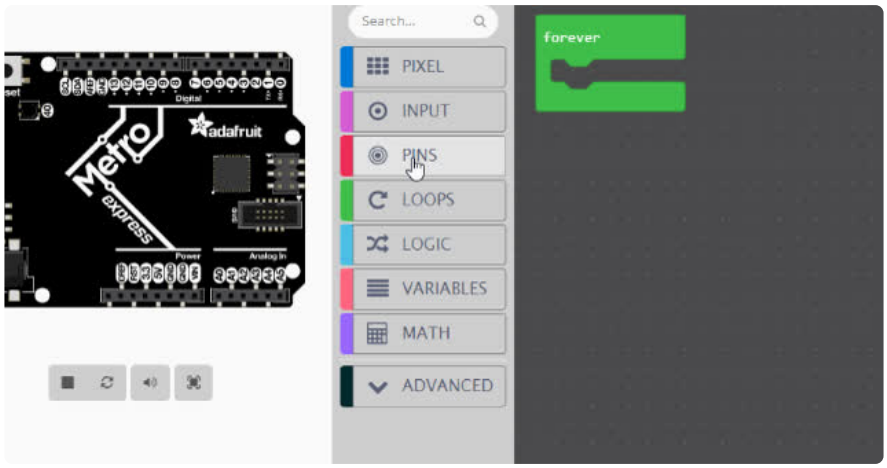
Any supported board from Adafruit can pair with the device via WebUSB so it's easy to connect.

You can also share the code and download it from any computer using the Chrome browser.



With the GEMMA M0, you can hook up a strip of NeoPixels using alligator clips which is nice because you don't need to solder.

What is MakeCode Maker?



MakeCode Maker, https://maker.makecode.com, is a web-based code editor for physical computing. It provides a block editor, similar to Scratch or Code.org, and also a JavaScript editor for more advanced users.

Some of the key features of MakeCode are:

- **web based editor:** nothing to install
- **cross platform:** works in most modern browsers from tiny phone to giant touch screens
- **compilation in the browser:** the compiler runs in your browser, it's fast and works offline
- **blocks + JavaScript:** drag and drop blocks or type JavaScript, MakeCode let's you go back and forth between the two.
- **works offline:** once you've loaded the editor, it stays cached in your browser.
- **event based runtime:** easily respond to button clicks, shake gestures and more

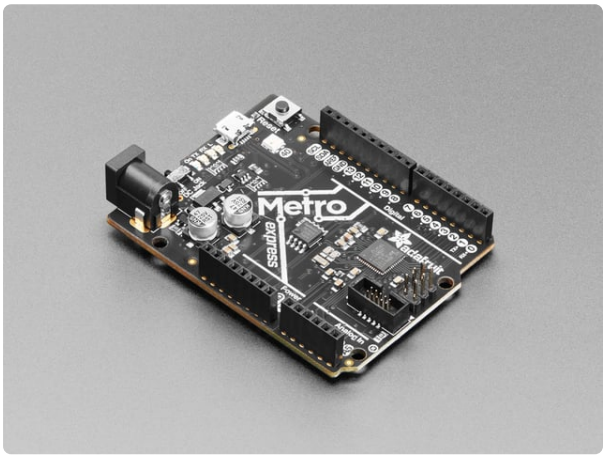
How is it related to makecode.adafruit.com ?

makecode.adafruit.com (https://adafru.it/wmd) and maker.makecode.com (https://adafru.it/C9N) are editors built using the MakeCode (https://adafru.it/Bhz) project. In both editors, one can use drag-and-drop blocks or JavaScript to program micro-controllers.

- makecode.adafruit.com (https://adafru.it/wmd) specifically applies to the **Adafruit Circuit Playground Express only**
- maker.makecode.com (https://adafru.it/C9N) aims at supporting the **Adafruit Express** boards (and more boards from different manufacturers), with an emphasis on breadboarding support.

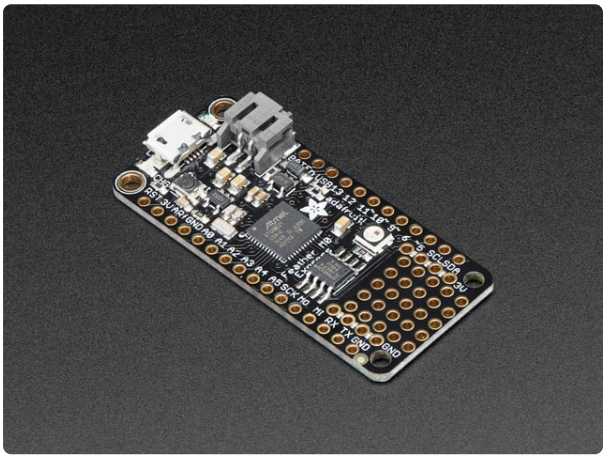
Is it open source?

Yes, Maker is open source under MIT at https://github.com/Microsoft/pxt-maker (https://adafru.it/ND5).



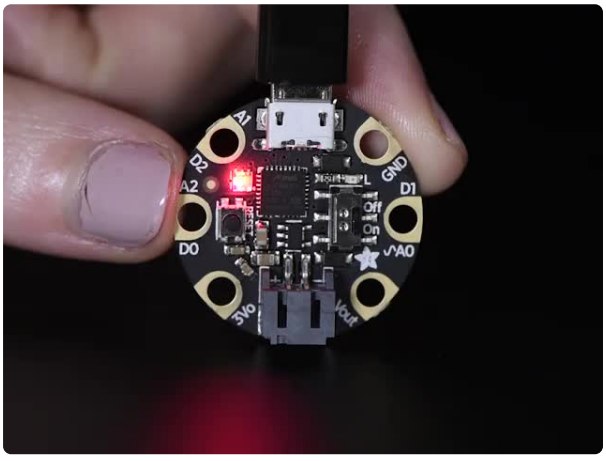
Adafruit METRO M0 Express - designed for CircuitPython

Metro is our series of microcontroller boards for use with the Arduino IDE. This new Metro M0 Express board looks a whole lot like our https://www.adafruit.com/product/3505



Adafruit Feather M0 Express

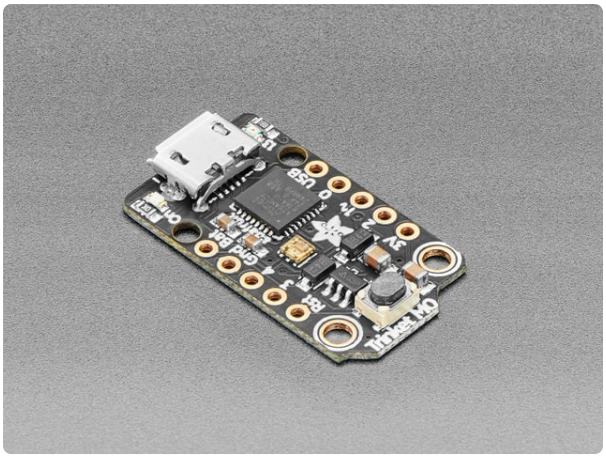
At the Feather M0's heart is an ATSAMD21G18 ARM Cortex M0+ processor, clocked at 48 MHz and at 3.3V logic, the same one used in the new https://www.adafruit.com/product/3403



Adafruit GEMMA M0 - Miniature wearable electronic platform

The Adafruit Gemma M0 is a super small microcontroller board, with just enough built-in to create many simple projects. It may look small and cute: round, about the...

https://www.adafruit.com/product/3501



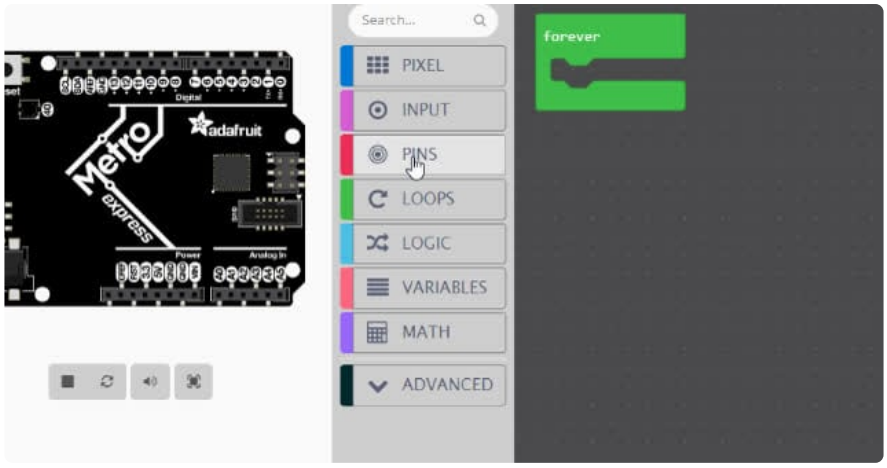
Adafruit Trinket M0 - for use with CircuitPython & Arduino IDE

The Adafruit Trinket M0 may be small, but do not be fooled by its size! It's a tiny microcontroller board, built around the Atmel ATSAMD21, a little chip with a lot...

https://www.adafruit.com/product/3500

Editing Blocks

The block editor is the easiest way to get started with MakeCode Maker. You can drag and drop blocks from the category list. Each time you make a change to the blocks, the simulator will automatically restart and run the code. You can test your program in the browser! The simulator will also generate the wiring for your breadboard for simple programs.



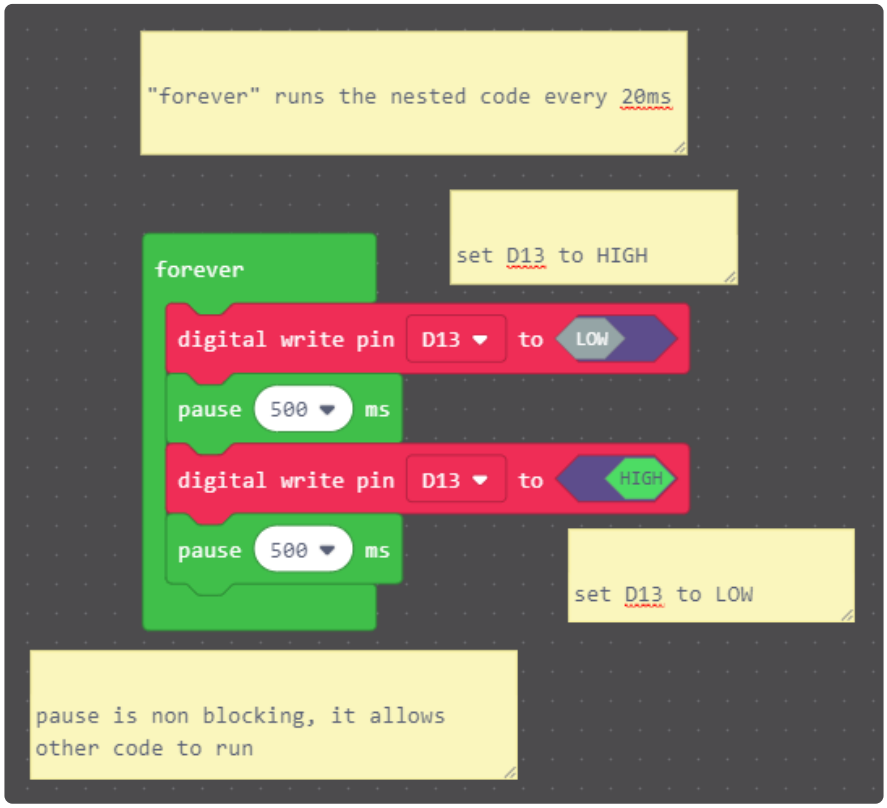
On the maker home screen, click on "New Project", then select which board you want to use (you can change board later too).

Blinky!

The animation above shows to use the blocks to create a program that blinks an LED.

Creating a blink effect is done by setting the pin HIGH, **pause** for a little, then set the pin LOW, pause for a little, then repeat **forever**.

- **forever** runs blocks in a loop with a 20ms pause in between (it is similar to Arduino **loop**).
- **digital write pin** sets the pin to high or low
- **pause** blocks the current thread for 100ms. If other events or forever loops are running, they have the opporunity to run in parallel.



Downloading and Flashing

Getting your code into your device is very easy with MakeCode. You do not need to install any software on your machine and the process takes two steps:

- Step 1: Connect your board via USB
- Step 2: **Compile and Download** the .uf2 file into your board drive

We are going to go through these two steps in detail.

Step 1: Connect your board via USB

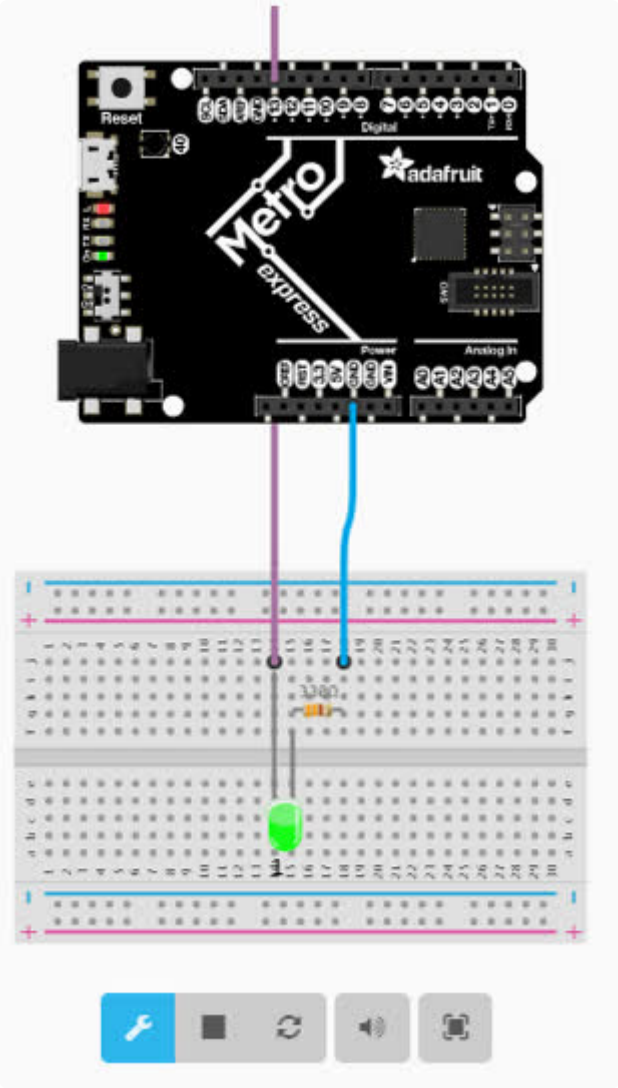
Connect your board to your computer via a USB cable. You should see a **MAKECODE** drive appear in your file explorer/finder. If your board is in bootloader mode, you will see drive names like **METROBOOT** or **GEMMABOOT**. We will call those **boardnameBOOT**.

If it is your first time running MakeCode or if you have previously installed Arduino or CircuitPython, you may need to double press the reset button to get your board into bootloader mode.

Step 2: Test your code in the simulator

Let's first verify that our code compiles properly in MakeCode.

MakeCode has a built-in simulator that re-loads and re-runs code when restarted. This is an easy way to both ensure that our code compiles and simulate it before moving it onto the board. The refresh button re-loads the simulator with your latest version of block code.



If you receive a "we could not run this project" error, please check over your code for errors.

Step 3: Download and flash your code

If your board is working in the simulator, it's time to download it to your actual board! Click the **Download** button. It will generate a .uf2 file and download it to your computer. UF2 (https://adafru.it/vPE) is a file format designed by Microsoft to flash microcontrollers over USB.

General Steps to copy over your program (not specific to any Operating system)

* Ensure your board is connected via USB.
* Find the .uf2 file generated by MakeCode in your file explorer. Copy it to the **MAKECODE** or **boardnameBOOT** volume.

* The status LED on the board will blink while the file is transferring. Once it's done transferring your file, the board will automatically reset and start running your code (just like in the simulator!)

On a Mac, you can safely ignore the "Disk Not Ejected Properly" notification that may appear after copying your .uf2 file.

Saving and Sharing

Extracting your code from the board

The .uf2 file you created by clicking on the Compile button in MakeCode also contains the source code of your program!

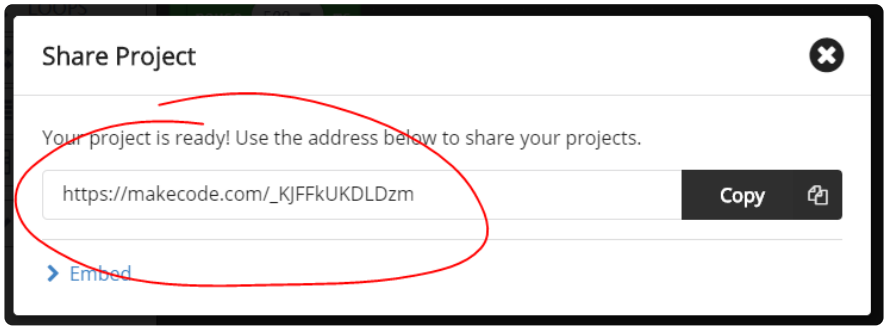
You can open this file in MakeCode by dragging and dropping it into the browser to edit it.

You can also find the current .uf2 file running on the **MAKECODE** or **boardnameBOOT** drive.

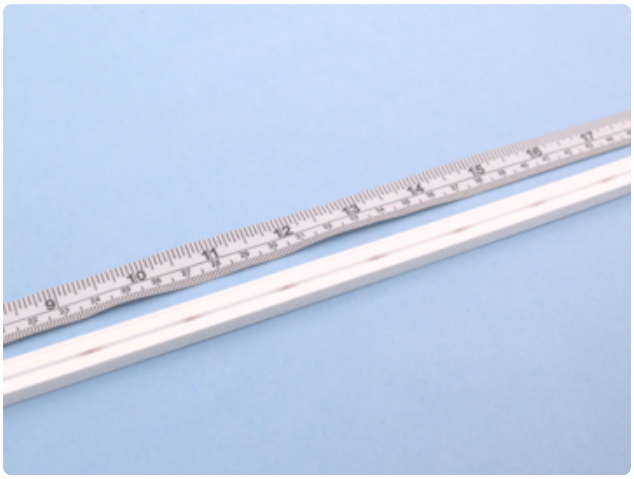
Sharing

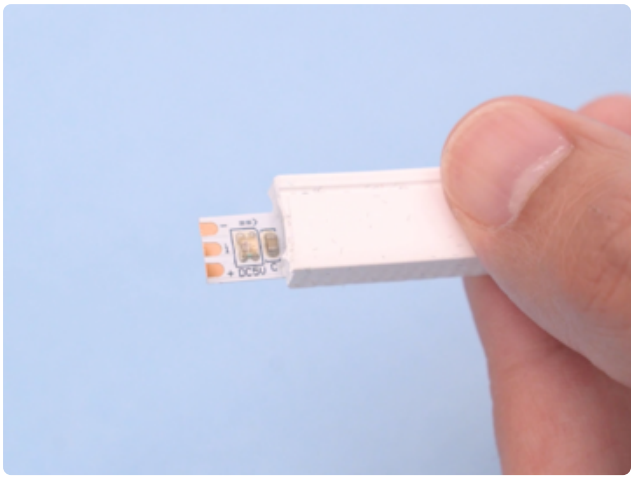
You can share your code by clicking on the share button. After confirmation, MakeCode will create a short unique URL for your code. Anyone with that URL will be able to reload the code.

These URLs can also be used to embed the editor your blog or web pages! Just copy paste the URL in your text editor and (if it supports oEmbed) it will automatically load it in your page.



Assemble





Measure the Strip

To start I'll measure the length LED strip required to fit over each ear.

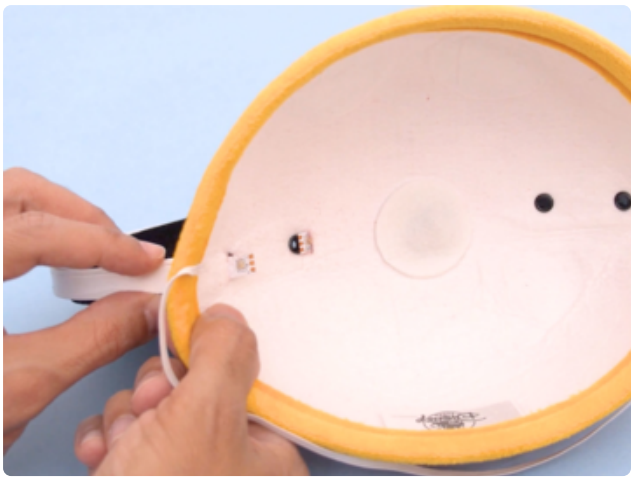
I cut two Neon strips that measured in at 310mm each.

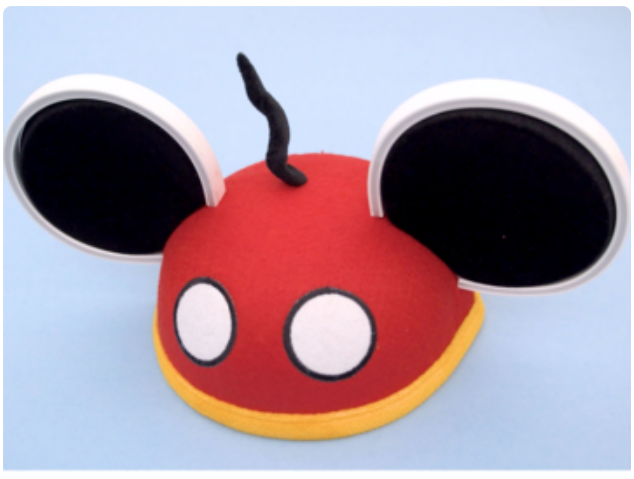
If you choose to use regular NeoPixel strip, the size will measure in at about 280mm.

Use scissors to cut between the pads as shown in the pictures.

Expose Pads

I removed about 17mm of sheathing from the strip so we can fit the strip into slits to be cut on the hat.





Prep the Hat

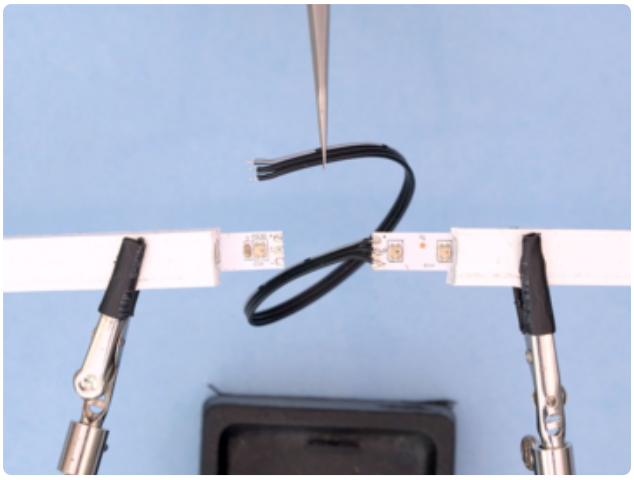
I used Adafruit ceramic blades (http://adafru.it/4306) to cut four slits on each side of both ears.

It only needs to be the width of the strip, which happens to be the same width as the ears.

Fit Strips into Slits

Take note of the arrow directions of the LED strips.

Fit the ends of the strips into the slits and **make sure the arrows all go in the correct direction**.



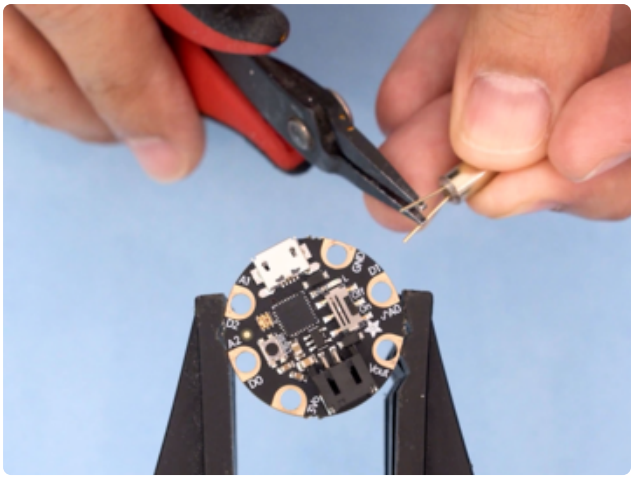
Solder Strip Wires

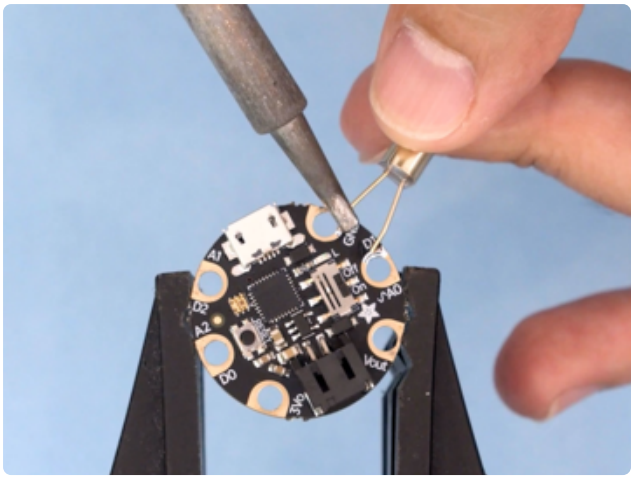
At this point, I proceeded to tin and solder the wires to each side of the first strip.

I measured and cut a silicone ribbon (http://adafru.it/3927) cable to connect the Gemma to the first strip, about 90mm long.

The second strip will connect to the first with a 160mm long wire.

The second strip will get soldered once it is fitted on the ears.

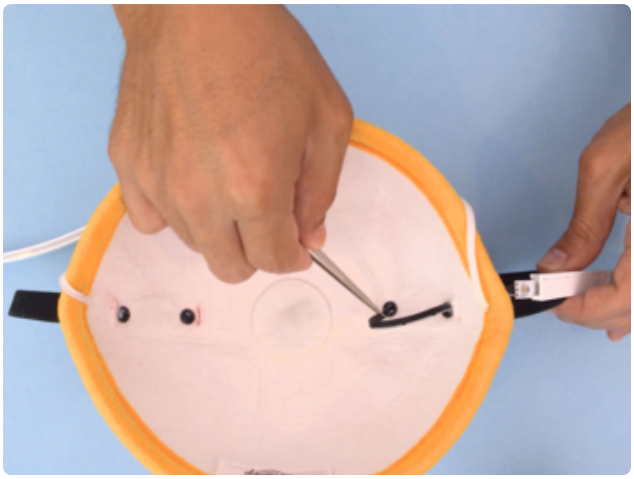




Solder Tilt Switch

Use a small amount of solder to tin the GND and **A0** pads on the Gemma.

To fit the tilt switch over the **GND** and **A0**, I used pliers to gently bend the two pins apart so they have enough distance to reach the pads.



Thread and Strip Wires

I'll first thread the wires that will connect the Gemma to the first strip.

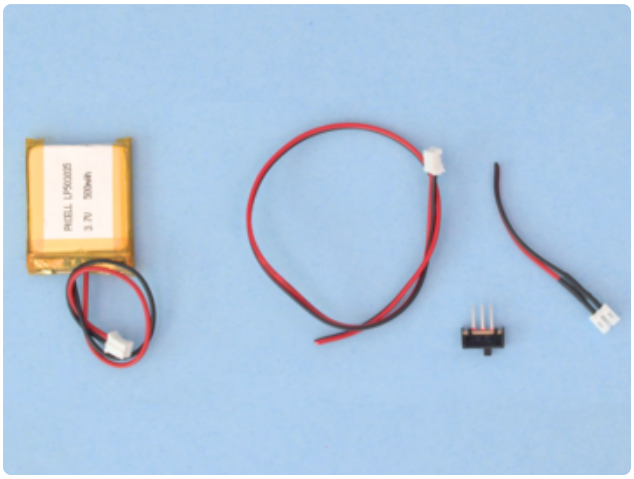
Pass the wires through the slit and then route the wires into the ear through the rivet eyelets.

Pass the wires through until the LED strip in taught around the ear.

Solder the Gemma

I used a third helping hand to hold the Gemma while soldering the LED strip connections.

I'll have to share the **GND** pad, so we can carefully tin the opposite side of the pad to connect the LED strip.

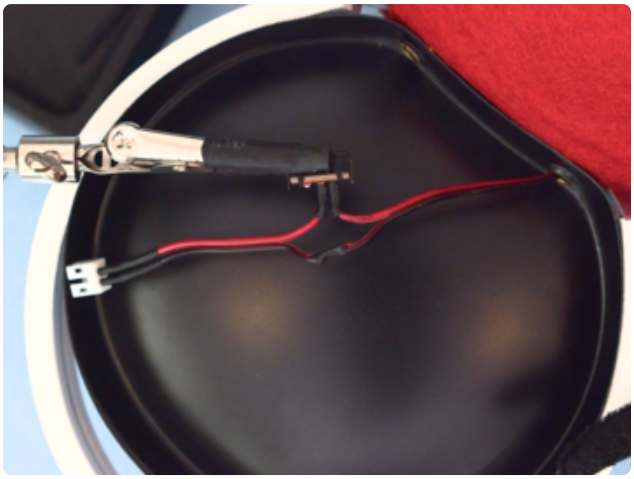


JST Slide Switch

I used a 2-pin JST Extension cable so we can easily disconnect the battery. Measure a cut the cable to 240mm.

To power the circuit on and off, you'll need to build a slide switch adapter.

I then measured and cut the JST Extension wire at about 60mm from the female connector to solder a slide switch in line with the voltage wire.



Heat Shrink

I used heat shrink tubing to insulate the two pins on the slide switch connections and one the ground wire.

Arrange Board and LiPo

Take note of the orientation of the tilt switch. I arranged it so the ball is tilted away from the base. If we don't angle it away from the base, the firework animation will keep triggering.

Mount board

Next I'll test out the circuit to make sure the tilt switch tiggers correctly.

I used a piece of black duck tape to match the color of the ears.

Take note of the tilt switch orientation and then apply a strip of duck tape over the Gemma and tilt switch.

I placed the Gemma with the USB port on the edge of the tape to allow a USB cable to easily plug in for reprogramming.

Apply pressure on the tape edges to adhere it to the ear.

Mount the Battery

Place the Slide switch on the edge of the duck tape to have easy access to it.

I carefully placed the battery close the to slide switch and then oriented the switch close to the edge on the ear. This way I can easily turn the circuit on and off.

Glue Strips

Now we can finally start to glue down the LED strips to the ears.

Double check the circuit still functions properly.

I used Dap 00694 Silicone Rubber Sealant (https://adafru.it/FK7) to adhere the strips to the ears. Use small binder clips to hold each glued section.

Allow the glue to dry for about 24 hours.

Complete!

Test the circuit by shaking to active animations. You can adjust the tilt switch if needed.

Test that the strip is properly glued down. I had to go back and reapply additional drops of glue to sections. The most important parts to check are the sections were the the strips are close the hat slits.