



# MP3 Feather - Gordon Cole

Created by Ruiz Brothers



<https://learn.adafruit.com/mp3-feather-gordon-cole>

Last updated on 2024-06-03 02:14:08 PM EDT

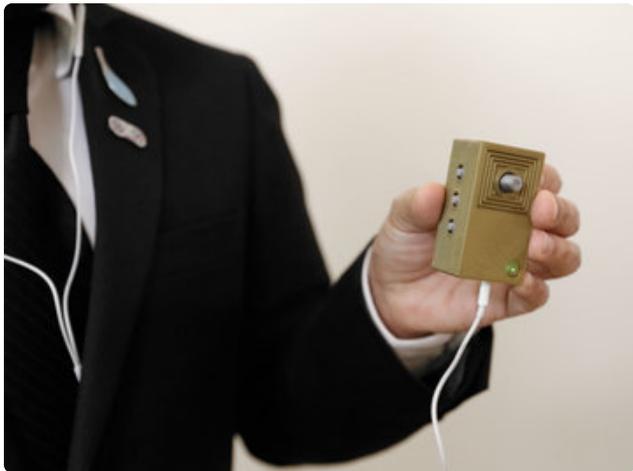
# Table of Contents

<b>Overview</b>	<b>3</b>
<ul style="list-style-type: none"><li>• Parametric Design</li><li>• Wearable</li><li>• Prerequisite Guides</li><li>• Parts, Tool &amp; Supplies</li></ul>	
<b>3D Printing</b>	<b>8</b>
<ul style="list-style-type: none"><li>• Slice Settings</li><li>• Print case without supports</li><li>• Use supports for clip</li><li>• Clean up</li></ul>	
<b>Circuit Diagram</b>	<b>11</b>
<b>Code</b>	<b>12</b>
<ul style="list-style-type: none"><li>• Feather M0 Express Arduino IDE Setup</li><li>• Installing Libraries in Arduino</li><li>• Uploading Code</li></ul>	
<b>Assemble</b>	<b>17</b>
<ul style="list-style-type: none"><li>• Short headers</li><li>• Align headers</li><li>• Solder male headers</li><li>• Check solder</li><li>• Solder female header</li><li>• Check header connections</li><li>• Bend buttons legs</li><li>• Tin Legs</li><li>•</li><li>• Solder ground pins</li><li>• Slide switch</li><li>•</li><li>•</li><li>• Solder switch</li><li>• Insert slide switch</li><li>• Mount Feather</li><li>• Mount Buttons</li><li>• Battery</li><li>• Attach Music Maker Feather</li><li>• Solder button pins</li><li>• Replace Knob</li><li>• Solder wires</li><li>• Solder pins</li><li>• Align lid</li><li>• Mount lid</li><li>• Attach wearable clip</li></ul>	

---

# Overview

In this project we'll build an MP3 Player based on the Feather M0 with a Music Maker wing. It plays a wide range of audio formats like such as MP3, AAC, Ogg Vorbis, WMA, MIDI, FLAC, WAV (PCM and ADPCM)! It also features a removable SD card for storage! Its portable and easy to power with the Feathers built in rechargeable circuit. The tiny size is perfect for installing inside small areas or as a wearable!



## Parametric Design

The player design is based on Gordon Cole's Hearing Aid from the popular show Twin Peaks.

We thought it would be fun to model the design around this characteristic wearable but with a twist.

Instead of a hearing aid, we made an aid for hearing music. ha.

Although a fun design we include the source file to easily modify the design to fit any theme!



## Wearable

You can optionally print the included clip to make an awesome wearable mp3 player!

It can clip onto your front shirt pocket or on a belt clip. Easy to reach, soft tactile buttons let you quickly access the play, pause and next buttons.



The volume knob is easy to control with editable gain adjustments in the code.

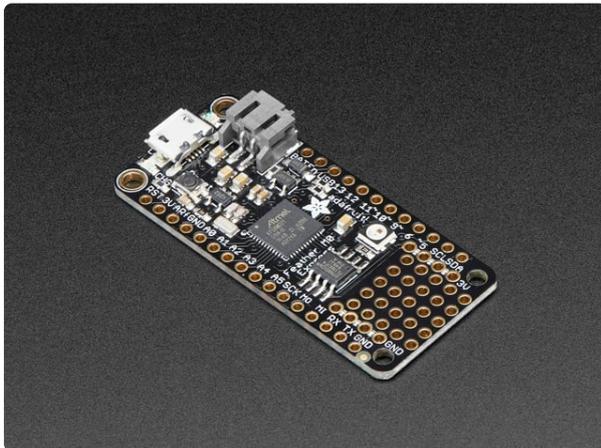
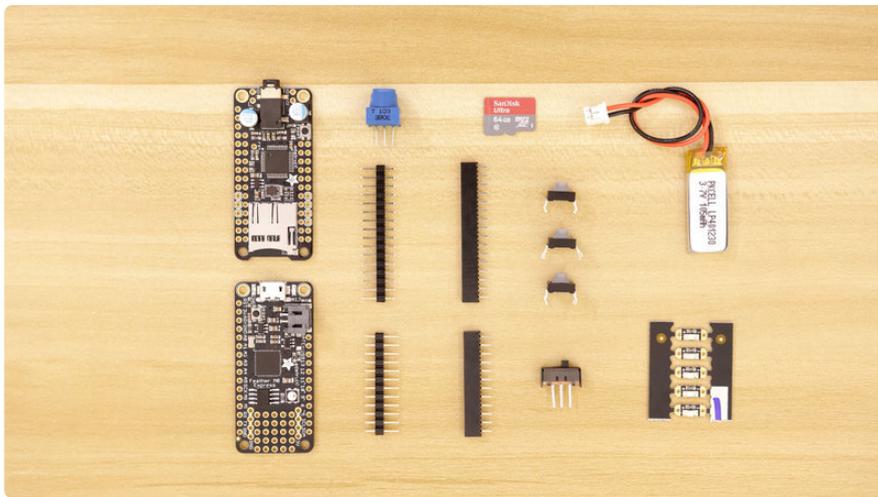
## Prerequisite Guides

Check out the following guide below to get a better understanding of the Feather M0 pin outs.

[Feather M0 Express \(https://adafru.it/BeM\)](https://adafru.it/BeM)

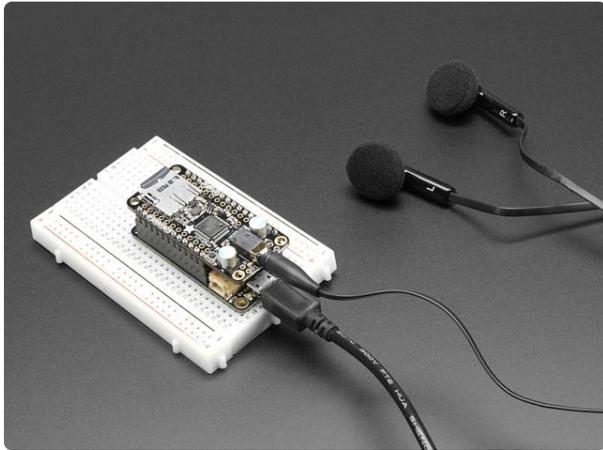
## Parts, Tool & Supplies

If you don't have access to a 3D printer, you can send the files to a service or check with your local hackerspace/library.



### Adafruit Feather M0 Express

At the Feather M0's heart is an ATSAMR21G18 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 Music Maker FeatherWing - MP3 OGG WAV MIDI Synth Player](https://www.adafruit.com/product/3357)

Bend all audio files to your will with the Adafruit Music Maker FeatherWing! It's a fun-size version of our Music...

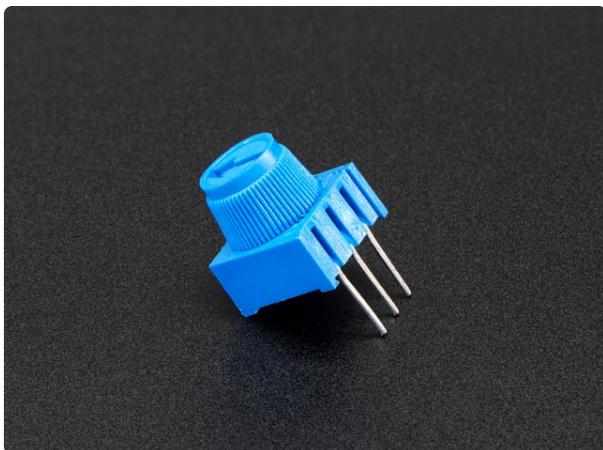
<https://www.adafruit.com/product/3357>



### [Soft Tactile Button \(8mm\) x 10](https://www.adafruit.com/product/3101)

Put your project in stealth mode with these silent Soft Tactile Switch Buttons. Tactile switches are standard input...

<https://www.adafruit.com/product/3101>



### [Breadboard trim potentiometer](https://www.adafruit.com/product/356)

These are our favorite trim pots, perfect for breadboarding and prototyping. They have a long grippy adjustment knob and with 0.1" spacing, they plug into breadboards or...

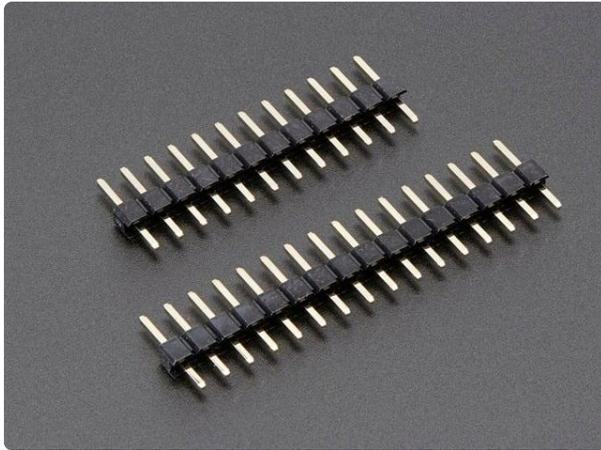
<https://www.adafruit.com/product/356>



### [Breadboard-friendly SPDT Slide Switch](https://www.adafruit.com/product/805)

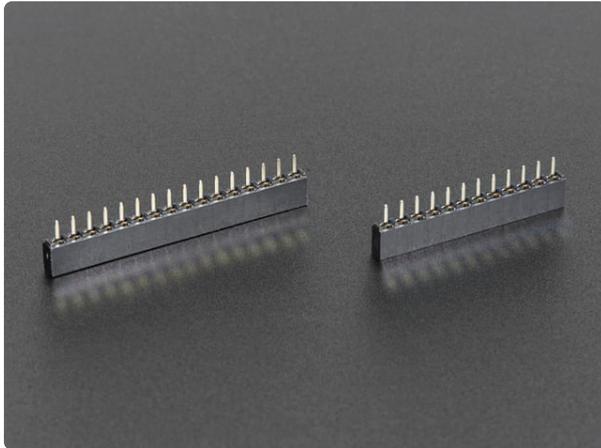
These nice switches are perfect for use with breadboard and perfboard projects. They have 0.1" spacing and snap in nicely into a solderless breadboard. They're easy to switch...

<https://www.adafruit.com/product/805>



### Short Feather Male Headers - 12-pin and 16-pin Male Header Set

These two Short Male Headers alone are, well, lonely. But pair them with any of our <https://www.adafruit.com/product/3002>



### Short Headers Kit for Feather - 12-pin + 16-pin Female Headers

These two Short Female Headers alone are, well, lonely. But pair them with any of our

<https://www.adafruit.com/product/2940>



### Lithium Ion Polymer Battery - 3.7v 100mAh

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/1570>



### Adafruit LED Sequins - Emerald Green - Pack of 5

Sew a little sparkle into your wearable project with an Adafruit LED Sequin.

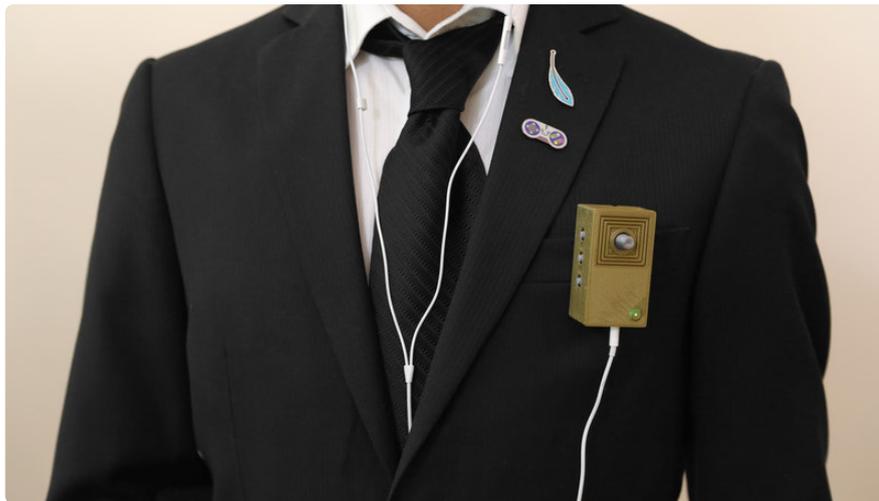
These are the kid-sister to our popular Flora NeoPixel,...

<https://www.adafruit.com/product/1756>



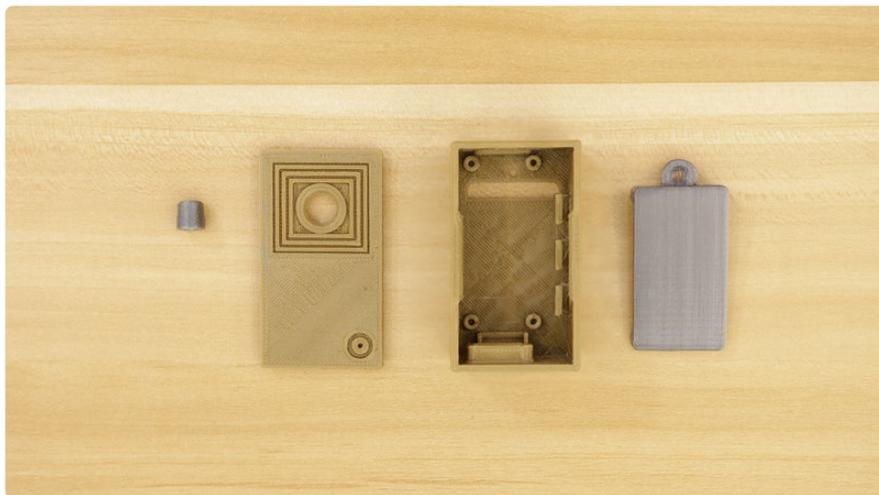
**4GB Blank SD/MicroSD Memory Card**  
Add mega-storage in a jiffy using this 4 GB micro-SD card. It comes with a SD adapter so you can use it with any of our shields or adapters! Preformatted to FAT so it works out of...

<https://www.adafruit.com/product/102>



---

## 3D Printing



The 3D printed parts are fairly easy to make with most common home desktop 3D printers that are on the market.

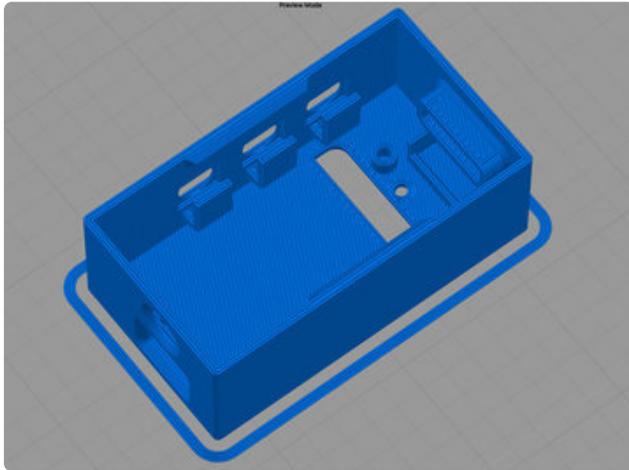
And if you don't have access a 3D printer, you can order our parts by visiting our Thingiverse page and have someone local 3D print the parts and ship them to you.

Download CAD source

<https://adafru.it/1a0s>

Download from Thingiverse

<https://adafru.it/D2N>

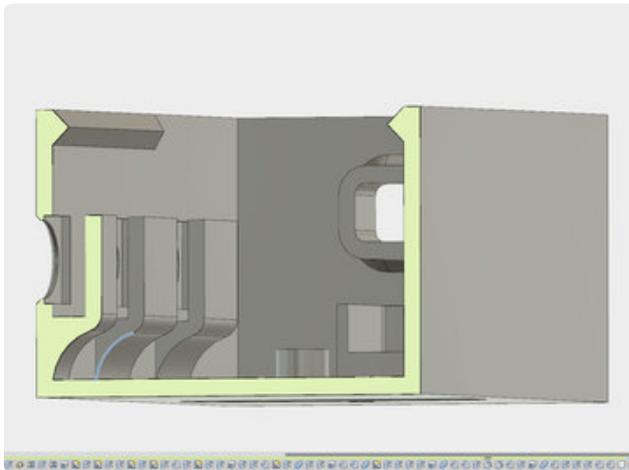


## Slice Settings

Download the STL file and import it into your 3D printing slicing software. You'll need to adjust your settings accordingly if you're using material different than PLA.

230C Extruder Temp  
No heated bed (65C for heated)  
1.0 Extrusion Multiplier  
.4mm Nozzle  
0.48 Extrusion Width  
.2mm Layer Height  
30% infill  
No Supports  
90mm/s | 120mm travel speed

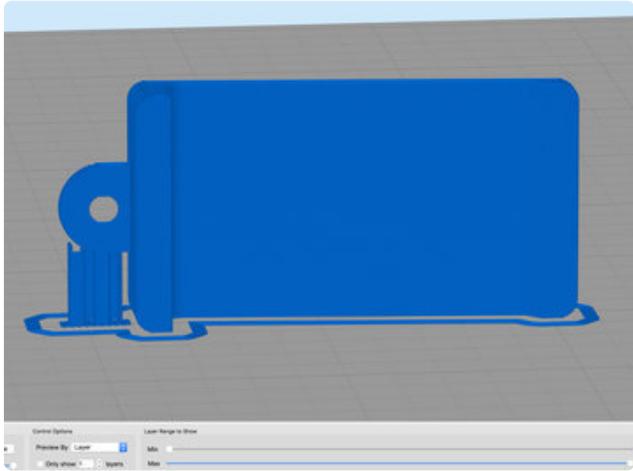
## Print case without supports



The enclosure features mounts on both sides with a cutout through the model for mounting the battery. To avoid adding supports, we can orient the model to print on its side. This worked out really well as the overhangs start to catch themselves after a couple of layers!

You can about 6 skirts (brims) the help adhere the enclosure to the build plate

## Use supports for clip



The screw mount for the clip may need a small amount of support as shown in the picture.

Orient the clip on its side to have the maximum amount strength.

To keep the supports from fusing with the part, position the supports away from the part, position the supports away from the main body and add a **horizontal offset of about .8mm** from the part.

## Clean up

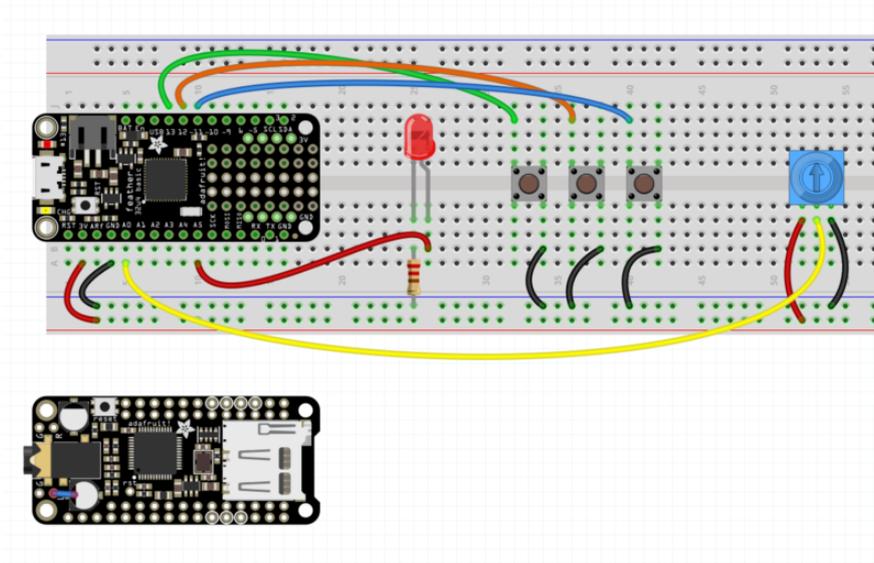


We used a flush diagonal cutter to clean up any stringing around the port openings and around the standoffs inside the enclosure.

Make sure the openings for the slide switch and USB ports are cleaned before mounting components. Use a hobby knife to help cut away stringing that could block components from mounting.

---

# Circuit Diagram



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 to the diagram.

Below are the wires lengths need for each component:

**Buttons:**

57mm

**Knob:**

83mm

**Slide switch:**

49mm

**LED:**

3v: 82mm

Ground: 70mm

The Music Maker wing fits on top of the Feather M0 board.

Play, Pause/ Resume and Next buttons are connected to pins, **13, 12 and 11**. Negative is connected to the ground rail on the prototyping area on the Feather M0.

The status LED is connects power to **A5** and negative to **GND**.

Potentiometer connects power to **3v**, negative to **GND** and analog to **A0**.

Battery connects to the **JST** port next to the USB port on the Feather M0.

---

## Code

### Feather M0 Express Arduino IDE Setup

We'll upload the code to the Adafruit Feather M0 Express using the Arduino IDE. We'll need to have the Adafruit board profile and libraries installed before uploading the code. If you're new to Arduino, let's first walkthrough the board profile setup using the [Adafruit Feather M0 Express guide \(https://adafru.it/zBo\)](https://adafru.it/zBo).

Feather M0 Arduino Setup

<https://adafru.it/zBo>

Music Maker Wing

<https://adafru.it/wbk>

### Installing Libraries in Arduino

With the board profile, we can then install the dependencies. We'll use Arduino's built-in Library Manager to install the libraries. Goto **Sketch > Include Library** and select **Manage Libraries**. Here, we'll search for the Adafruit VS1053 Library and install the latest version.

### Uploading Code

With the Feather board profile and Arduino library installed, download the **mp3player.ino** file save them to a new folder named "**mp3player**". You'll want to place the new folder into arduino's sketches folder (ie. ~/HD/Documents/Arduino/sketches).

Open the mp3player.ino file in the Arduino IDE. Select the **Adafruit Feather M0 Express** board under the **Tools > Board** menu. Connect the Feather board to your computer via microUSB cable and select **"/dev/cu.modem..."** under the **Tools > Port** menu (Option might be named different using Windows OS).

Upload the code to the Feather using the **Upload** command, **Cmd+U** or clicking the arrow icon.

```
// SPDX-FileCopyrightText: 2017 Carter Nelson for Adafruit Industries
//
// SPDX-License-Identifier: MIT

//
// Gordon Cole MP3 Player
//

#include <SPI.h>
#include <SD.h>
#include <Adafruit_VS1053.h>

// Guide is based on Feather M0 Express
// ARDUINO_SAMD_FEATHER_M0 defines only
// VS1053 Pins
#define VS1053_RESET -1 // VS1053 reset pin (not used!)
#define VS1053_CS 6 // VS1053 chip select pin (output)
#define VS1053_DCS 10 // VS1053 Data/command select pin (output)
#define CARD_CS 5 // Card chip select pin
#define VS1053_DREQ 9 // VS1053 Data request, ideally an Interrupt pin
// Button Pins
#define BUTTON_PLAY 13 // PLAY / STOP button
#define BUTTON_PAUSE 12 // PAUSE / RESUME button
#define BUTTON_NEXT 11 // NEXT button
// Status LED
#define LED_STATUS 19 // status LED
#define BLINK_RATE 500 // blink rate in ms
// Volume Control
#define KNOB_VOLUME 0 // volume knob
#define KNOB_MIN 0 // min ADC value
#define KNOB_MAX 1023 // max ADC value
#define VOL_MIN 0 // min volume (most loud)
#define VOL_MAX 50 // max volume (most quiet)
#define VOL_UPDATE 250 // update rate in ms
#define VOL_SAMPLES 10 // number of reads for average
#define VOL_SAMPLE_RATE 5 // ms delay per sample
#define VOL_THRESHOLD 20 // vol must change by this many counts
// Maximum number of files (tracks) to load
#define TRACKS_MAX 100
// Player behavior
#define AUTO_PLAY_NEXT true // true to automatically go to next track

unsigned long currentMillis;
unsigned long previousBlinkMillis, previousVolMillis;
int currentKnob, previousKnob;
int volume;
int currentTrack, totalTracks;
char trackListing[TRACKS_MAX][13] = { ' ' };
enum mode {
  PLAYING,
  PAUSED,
  STOPPED
} currentMode = STOPPED;

Adafruit_VS1053_FilePlayer musicPlayer =
  Adafruit_VS1053_FilePlayer(VS1053_RESET, VS1053_CS, VS1053_DCS, VS1053_DREQ,
  CARD_CS);

//-----
void setup() {
  Serial.begin(9600);
  // Leave commented for standalone operation, uncomment for troubleshooting
  //while (!Serial) ;
```

```

// Initialize pins
pinMode(BUTTON_PLAY, INPUT_PULLUP);
pinMode(BUTTON_PAUSE, INPUT_PULLUP);
pinMode(BUTTON_NEXT, INPUT_PULLUP);
pinMode(LED_STATUS, OUTPUT);

// Initialize status LED
previousBlinkMillis = millis();
digitalWrite(LED_STATUS, LOW);

Serial.println("\n\nGordon Cole MP3 Player");

// Initialize the music player
if (!musicPlayer.begin()) {
  Serial.println(F("Couldn't find VS1053, do you have the right pins defined?"));
  while (1) {
    digitalWrite(LED_STATUS, !digitalRead(LED_STATUS));
    delay(100);
  }
}
Serial.println(F("VS1053 found"));
musicPlayer.softReset();

// Make a tone to indicate VS1053 is working
musicPlayer.sineTest(0x44, 500);

// Set volume for left, right channels. lower numbers == louder volume!
previousVolMillis = millis();
previousKnob = analogRead(KNOB_VOLUME);
volume = map(previousKnob, KNOB_MIN, KNOB_MAX, VOL_MIN, VOL_MAX);
Serial.print("Volume = "); Serial.println(volume);
musicPlayer.setVolume(volume, volume);

// Initialize the SD card
if (!SD.begin(CARD_CS)) {
  Serial.println(F("SD failed, or not present"));
  while (1) {
    digitalWrite(LED_STATUS, !digitalRead(LED_STATUS));
    delay(100);
  }
}
Serial.println("SD OK!");

// Load list of tracks
Serial.println("Track Listing");
Serial.println("=====");
totalTracks = 0;
loadTracks(SD.open("/"), 0);
currentTrack = 0;

// Setup interrupts (DREQ) for playback
musicPlayer.useInterrupt(VS1053_FILEPLAYER_PIN_INT);
}

//-----
void loop() {
  // Check and set volume
  updateVolume();

  // Update status LED
  updateStatusLED();

  // Auto play next track if feature enabled
  if (AUTO_PLAY_NEXT) {
    if (currentMode==PLAYING && musicPlayer.stopped()) {
      currentTrack = ++currentTrack < totalTracks ? currentTrack : 0;
      Serial.print("Next ");
      Serial.print(currentTrack); Serial.print("=");
    }
  }
}

```

```

        Serial.println(trackListing[currentTrack]);
        musicPlayer.startPlayingFile(trackListing[currentTrack]);
        currentMode = PLAYING;
    }
}

// Start / Stop
if (!digitalRead(BUTTON_PLAY)) {
    if (musicPlayer.stopped()) {
        Serial.print("Start ");
        Serial.print(currentTrack); Serial.print("=");
        Serial.println(trackListing[currentTrack]);
        musicPlayer.startPlayingFile(trackListing[currentTrack]);
        currentMode = PLAYING;
    } else {
        Serial.println("Stopped.");
        musicPlayer.stopPlaying();
        currentMode = STOPPED;
    }
}
delay(250);

// Pause / Resume
if (!digitalRead(BUTTON_PAUSE)) {
    if (!musicPlayer.stopped()) {
        if (musicPlayer.paused()) {
            Serial.println("Resumed");
            musicPlayer.pausePlaying(false);
            currentMode = PLAYING;
        } else {
            Serial.println("Paused");
            musicPlayer.pausePlaying(true);
            currentMode = PAUSED;
        }
    }
}
delay(250);

// Next
if (!digitalRead(BUTTON_NEXT)) {
    if (!musicPlayer.stopped()) {
        Serial.println("Stopping current playback.");
        musicPlayer.stopPlaying();
    }
    currentTrack = ++currentTrack < totalTracks ? currentTrack : 0;
    Serial.print("Next ");
    Serial.print(currentTrack); Serial.print("=");
    Serial.println(trackListing[currentTrack]);
    musicPlayer.startPlayingFile(trackListing[currentTrack]);
    currentMode = PLAYING;
    delay(250);
}

}

//-----
void updateVolume() {
    // Rate limit
    currentMillis = millis();
    if (currentMillis - previousVolMillis < VOL_UPDATE) return;
    previousVolMillis = currentMillis;
    // Get an average reading
    currentKnob = 0;
    for (int i=0; i<VOL_SAMPLES; i++) {
        currentKnob += analogRead(KNOB_VOLUME);
        delay(VOL_SAMPLE_RATE);
    }
    currentKnob /= VOL_SAMPLES;
    // Only update if it's changed
    if (abs(currentKnob-previousKnob) > VOL_THRESHOLD) {

```

```

Serial.print("["); Serial.print(currentKnob);
Serial.print(","); Serial.print(previousKnob);
Serial.print("] ");
previousKnob = currentKnob;
volume = map(currentKnob, KNOB_MIN, KNOB_MAX, VOL_MIN, VOL_MAX);
Serial.print("Volume set to: "); Serial.println(volume);
musicPlayer.setVolume(volume, volume);
}
}

//-----
void updateStatusLED() {
  if (musicPlayer.paused()) {
    // Blink it like a polaroid
    currentMillis = millis();
    if (currentMillis - previousBlinkMillis > BLINK_RATE) {
      previousBlinkMillis = currentMillis;
      digitalWrite(LED_STATUS, !digitalRead(LED_STATUS));
    }
  } else if (!musicPlayer.stopped()) {
    // It's so on again
    digitalWrite(LED_STATUS, HIGH);
  } else {
    // It's so off again
    digitalWrite(LED_STATUS, LOW);
  }
}

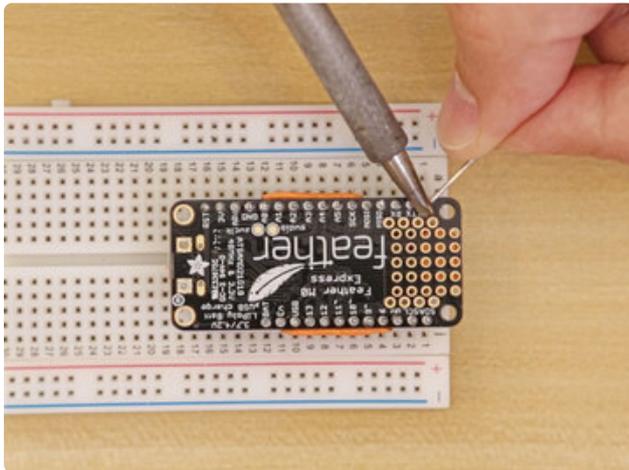
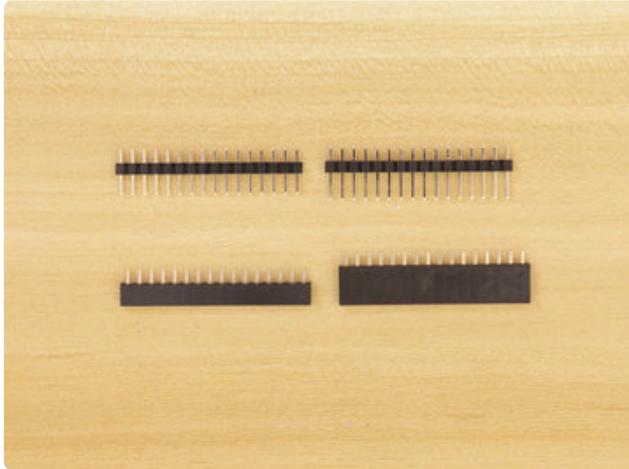
//-----
void loadTracks(File dir, int level) {
  while (true) {
    File entry = dir.openNextFile();
    if (!entry) return;
    if (entry.isDirectory()) {
      // Recursive call to scan next dir level
      loadTracks(entry, level + 1);
    } else {
      // Only add files in root dir
      if (level == 0) {
        // And only if they have good names
        if (nameCheck(entry.name())) {
          strncpy(trackListing[totalTracks], entry.name(), 12);
          Serial.print(totalTracks); Serial.print("=");
          Serial.println(trackListing[totalTracks]);
          totalTracks++;
        }
      }
    }
    entry.close();
    // Stop scanning if we hit max
    if (totalTracks >= TRACKS_MAX) return;
  }
}

//-----
bool nameCheck(char* name) {
  int len = strlen(name);
  // Check length
  if (len <= 4) return false;
  // Check extension
  char* ext = strchr(name, '.');
  if (!(
    strcmp(ext, ".MP3") == 0 ||
    strcmp(ext, ".OGG") == 0
  )) return false;
  // Check first character
  switch(name[0]) {
    case '_': return false;
  }
}

```

```
return true;  
}
```

## Assemble



### Short headers

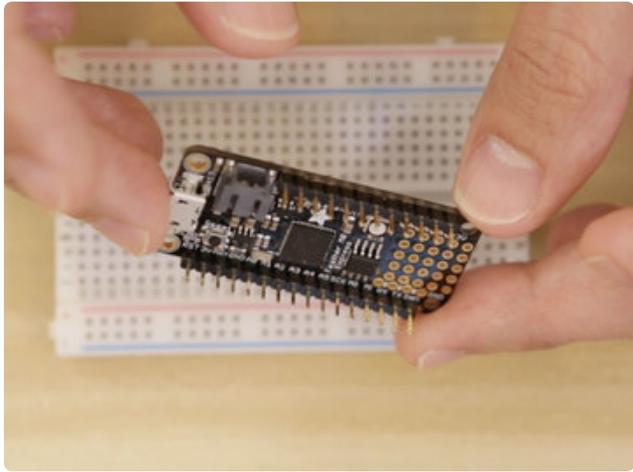
To save as much space as we could, we'll use the short male and female header pins to connect the Feather M0 and Music Maker wing.

### Align headers

To keep the head aligned, we use a bread board and then added small pieces of sticky tac to help hold the Feather in place while soldering.

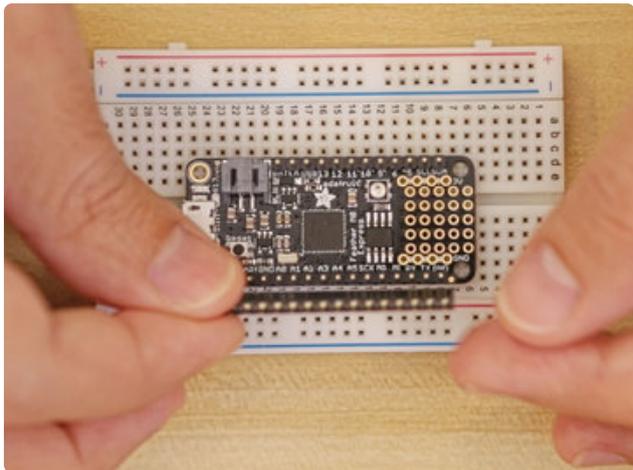
### Solder male headers

Position the long ends of the male headers on the breadboard as shown in the picture. We'll lay the Feather over the pins and then solder the shorter side of the headers to the back of the board.

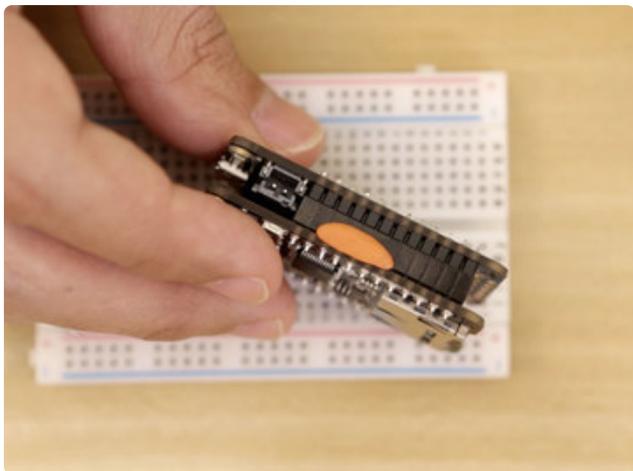


## Check solder

Check that all of the the pins have enough solder applied to each pin. Now we'll turn the Feather board over so the back side is laying on the breadboard.

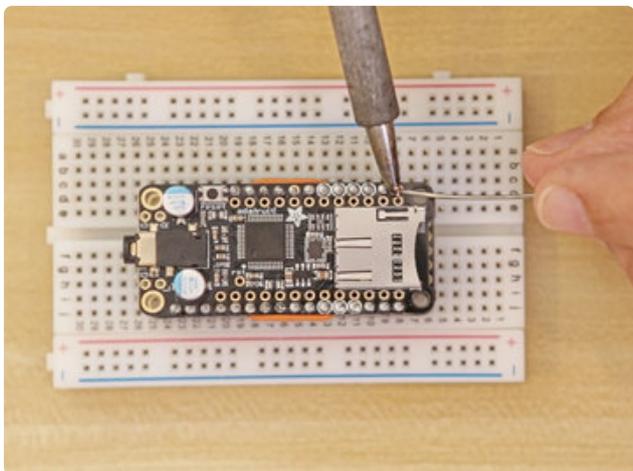


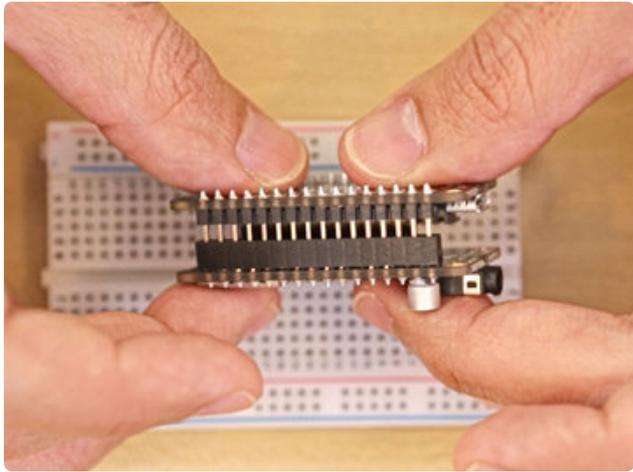
Fit the female headers onto the male headers and then lay the Music Maker Feather on top. Apply a small amount tac on the sides of the Music Maker to hold and align the female headers in place.



## Solder female header

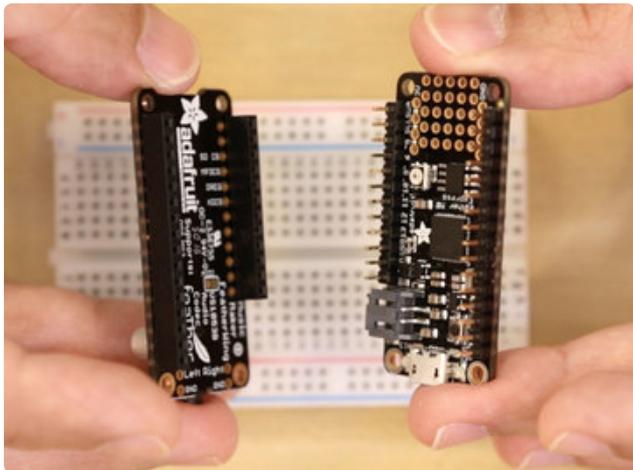
Carefully solder each female header pin without moving the boards. The enclosure port holes won't properly align if the headers are soldered on crooked.





## Check header connections

Remove the tac and check that all of your solder points are connected. Make sure there isn't excess solder bridging other pins.

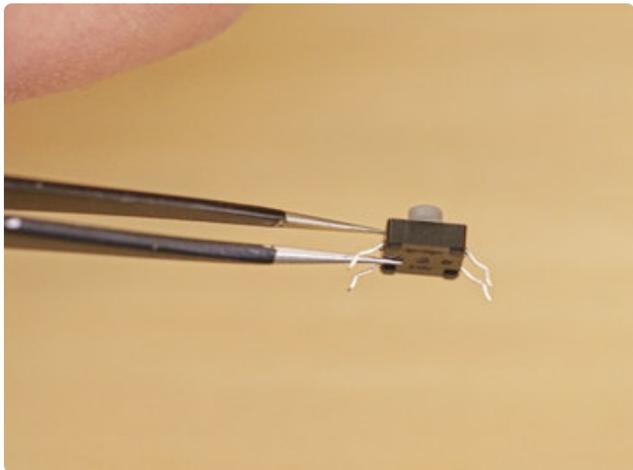


Gently remove the Music Maker wing and set it a side as we'll move on to soldering the buttons.



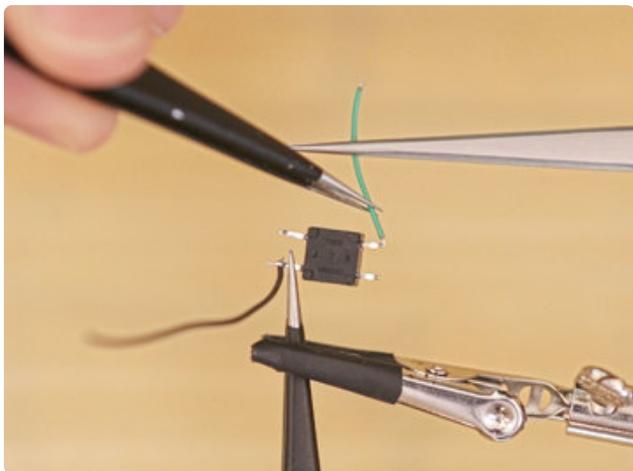
## Bend buttons legs

Slightly bend the legs outward with a pair of flat pliers. We'll need to gently pull them back far enough to fit through the button mounts inside the enclosure.



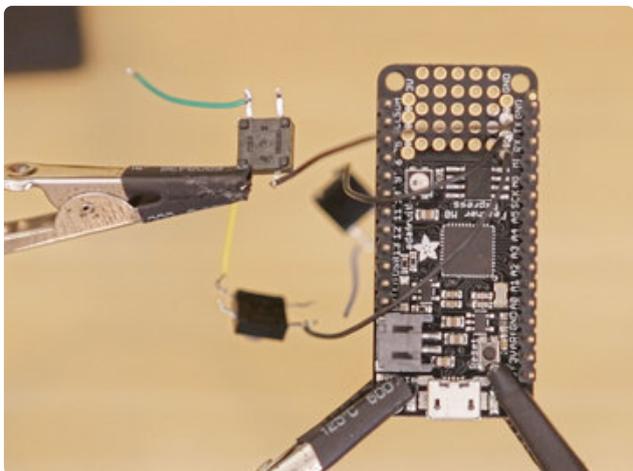
## Tin Legs

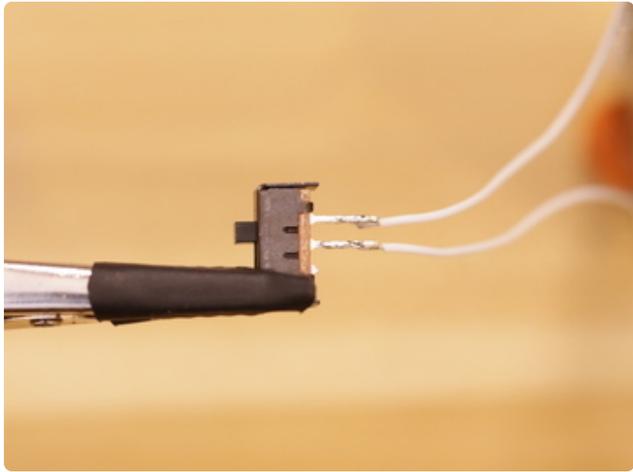
We'll need to solder a wire to two of the legs that are diagonally across from each other. Apply a small amount of solder to both legs and then cut wires for each. Cut six wires **57mm** long, for each io pin and ground connection needed.



## Solder ground pins

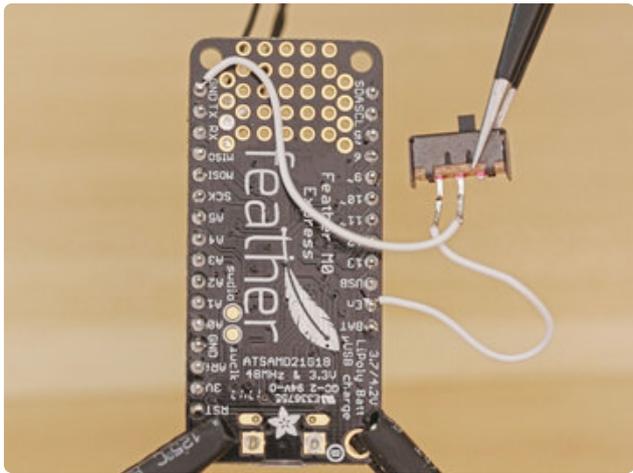
First, we'll solder the ground connections to all three buttons along the ground rail on the prototyping are of the Feather M0.





## Slide switch

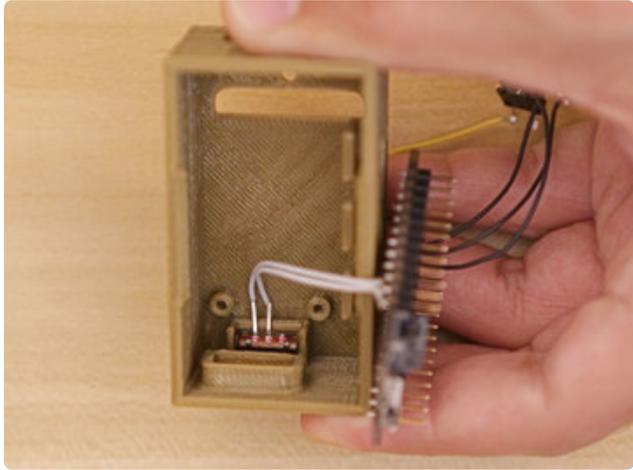
Use a third helping hand to hold a slide switch and tin two pins and then remove the third pin. Don't remove the middle pin. Cut two wires **49mm** long and solder them each of the pins on the slide switch.



## Solder switch

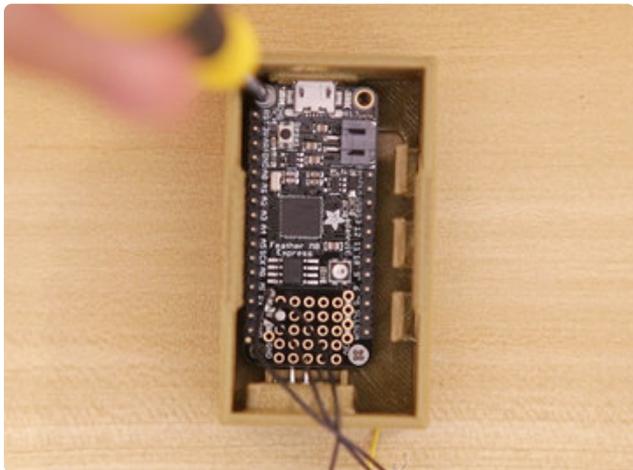
Turn the Feather M0 over and solder one of the slide switch pins to the **EN pin** and the other to the **GND pin**.

## Insert slide switch



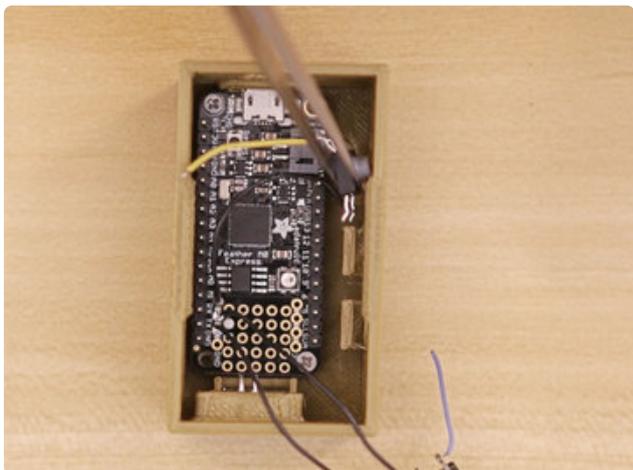
Insert the slide switch at an angle between the two walls inside the enclosure as shown in the picture. Make sure the port opening for the switch are clean. Use tweezers to help push the switch through the port opening.

## Mount Feather



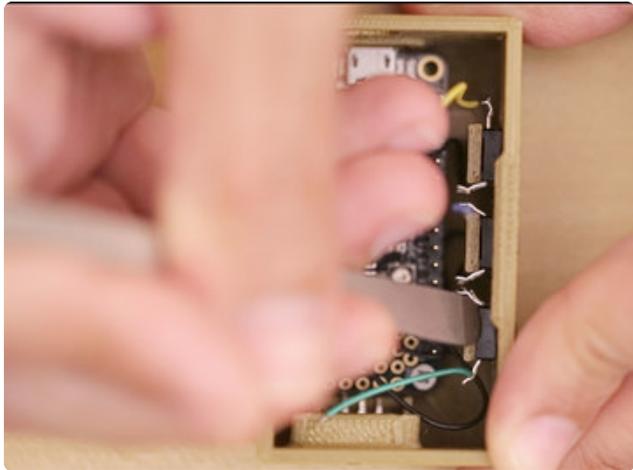
Now we can carefully mount the Feather M0 inside the enclosure. Align the port holes and use **M2.5x5mm** long screws to mount the board.

## Mount Buttons



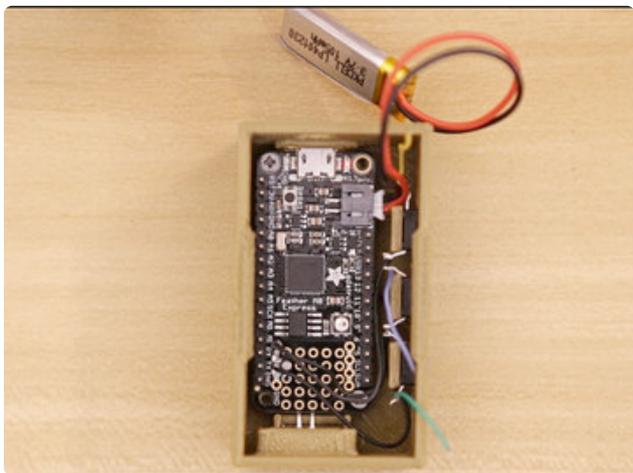
Insert the buttons at an angle. You can use tweezers to help push the buttons between the walls and into the cut outs for each.

After inserting each button, gently bend the leg back inward to make room for the next button.



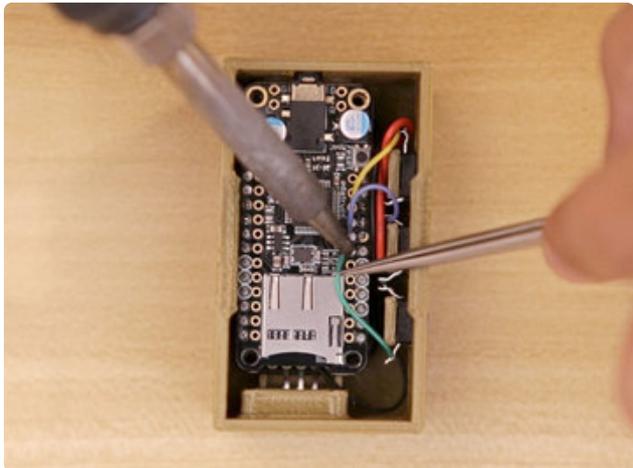
## Battery

Now we can mount the battery on top of the Feather M0. Plug the battery into the JST connection on the boards and then arrange the wires into the corner of the enclosure.



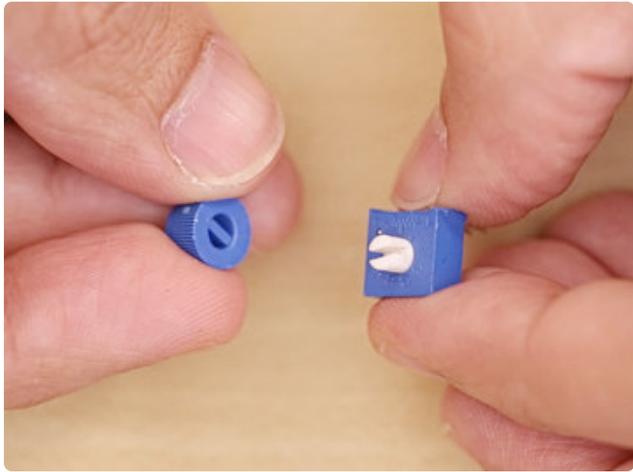
## Attach Music Maker Feather

Align and mount the Music Maker Feather on top of the M0. Make sure to maneuver the ground wires so the headers don't kink them when mounting.



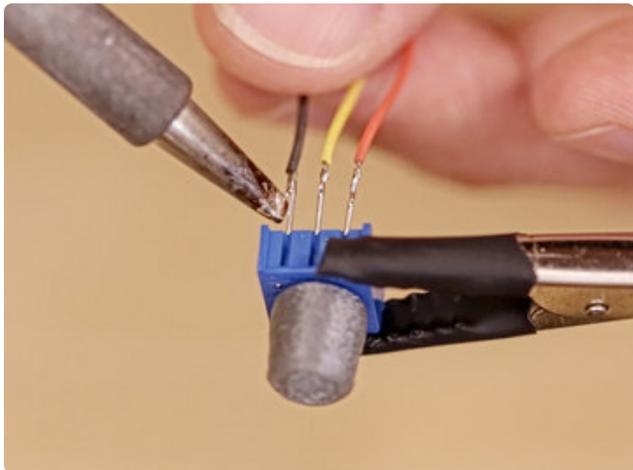
## Solder button pins

Now we can solder each button to pins 13, 12 and 11.



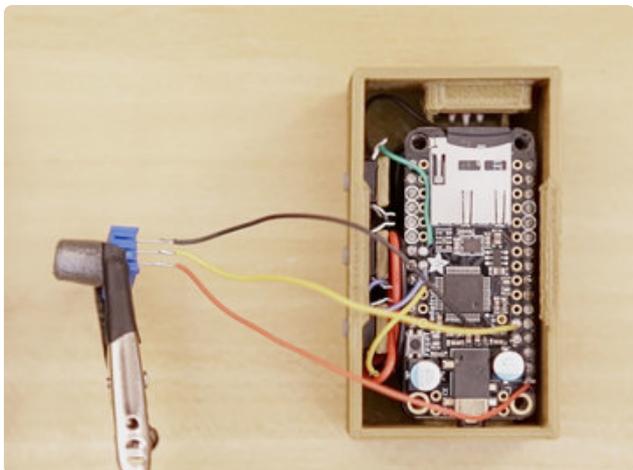
## Replace Knob

Gently remove the knob on the potentiometer by carefully wedging your fingernail between it and the base. The printed replacement fits into the white plastic nub.



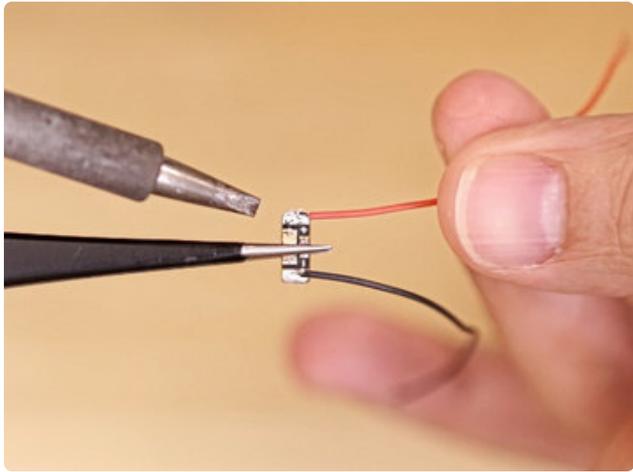
## Solder wires

Solder pins as shown in the circuit diagram. Tin all three pins on the potentiometer, cut three wires **83mm** long and then solder to each wire.



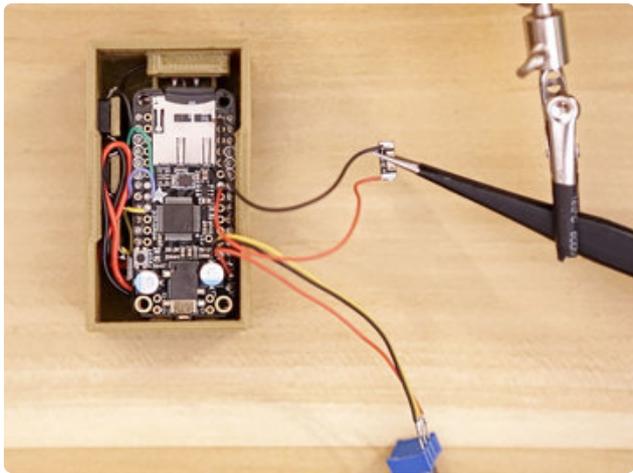
## Solder pins

Now we can solder each pin to **A0**, **GND** and **3V**. You can solder wires to the secondary pin or right on top of the header pins to avoid soldering so close to other components on the Music Maker board.



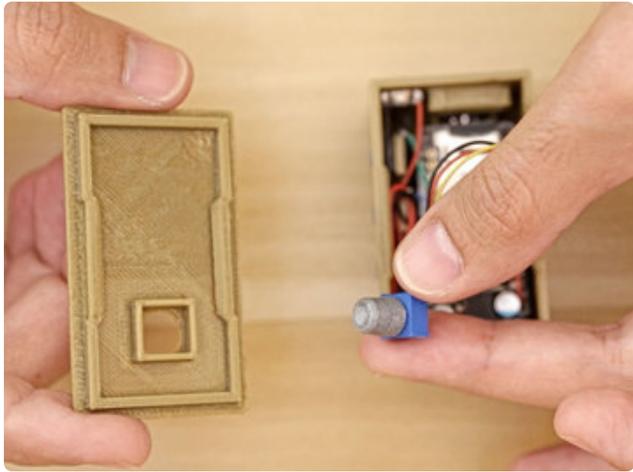
## LED

Use a third helping hand to hold the tiny Sequin led in place while tinning a soldering wires for power and ground.



Cut wires **82mm** long and solder them to the + and - pads on the sequin.

We can mount the sequin on the corner, over top of the Music Maker wing as shown in the picture.



## Align lid

Orient the potentiometer so the pins are pointing towards the long side of the lid and gently push it inside the port opening walls. Gently bend the pins straight to fit it inside the enclosure.



## Mount lid

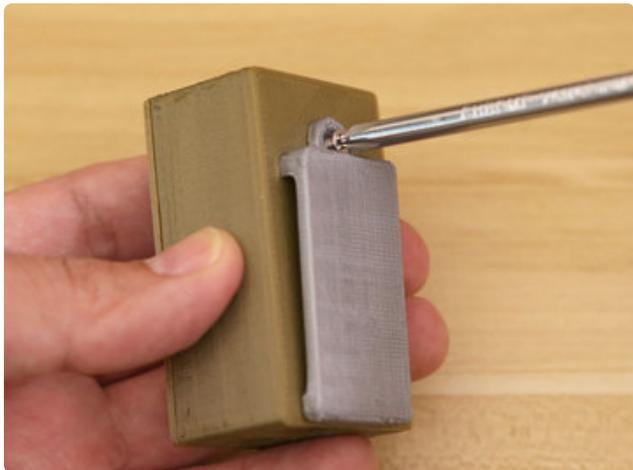
Align the lid so the potentiometer is over the SD card slot. Press the lid so the snap fit nubs on the side snap fit together.





## Attach wearable clip

Next we can go ahead and push the printed clips through the back opening on the enclosure.



Align the two the screws mounts so the back flat side is facing away from the enclosure as shown in the picture.

Align the clip to the slot on the enclosure. Press the protrusion on the clip into the slot until the screw mount pushes up against the back of the enclosure.



Use **M2.5x5mm** long screws to secure the clips to the enclosure.

Make sure the protrusion has a tight fit into the slot or the clip will have a weak hold when wearing.

