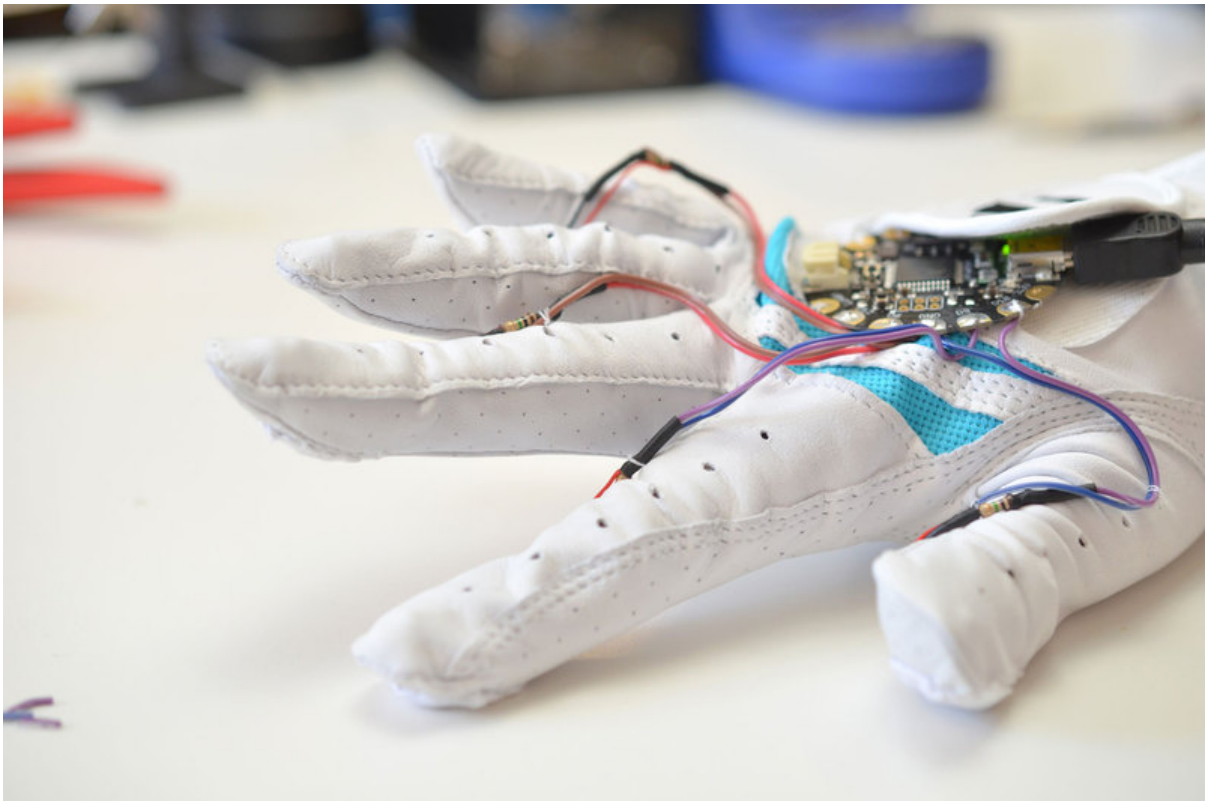
Flora MIDI Drum Glove

Created by Becky Stern



https://learn.adafruit.com/midi-drum-glove

Last updated on 2024-06-03 01:31:53 PM EDT

Table of Contents

Overview

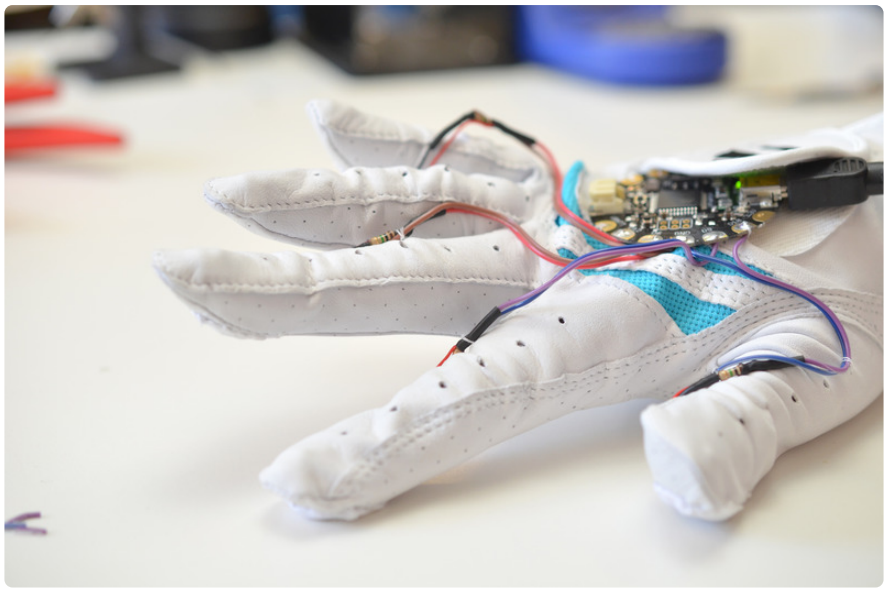
Play your favorite synths by finger drumming! Stitch up four piezos into a glove and use FLORA to transmit signals to your favorite music-making software.

Before you begin, read the Getting Started with FLORA (https://adafru.it/dwi) guide! You'll have installed the Adafruit Arduino software by now, OR you can program your FLORA directly from your browser using Codebender (https://adafru.it/dwj). Fancy!

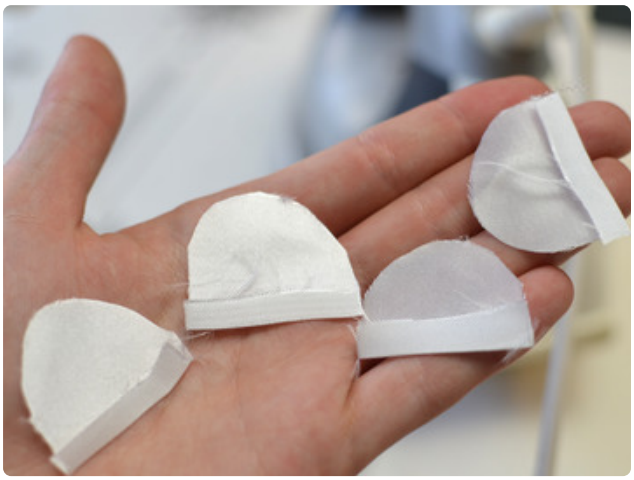
You may also want to brush up on your soldering skillz with our guide (https://adafru.it/drl).

For this project you will need:

- FLORA main board (http://adafru.it/659)
- 4x small piezos (http://adafru.it/1740)
- USB miniB cable (http://adafru.it/260)
- 4x 1M ohm resistors
- ribbon wire
- glove (https://adafru.it/dyR)
- scrap fabric to match glove
- pencil
- soldering iron (http://adafru.it/180) and solder
- third hand tool (http://adafru.it/291)
- heat shrink tubing (http://adafru.it/344)
- wire strippers (http://adafru.it/527), pliers (http://adafru.it/1368), and flush diagonal cutters (http://adafru.it/152)
- heat gun
- painter's tape or masking tape
- sewing needle (http://adafru.it/615) and thread
- scissors



Sew Piezos



Start by ironing out some fabric to match your glove. Cut four small pieces slightly larger than your fingertips, and iron a small hem on one side.

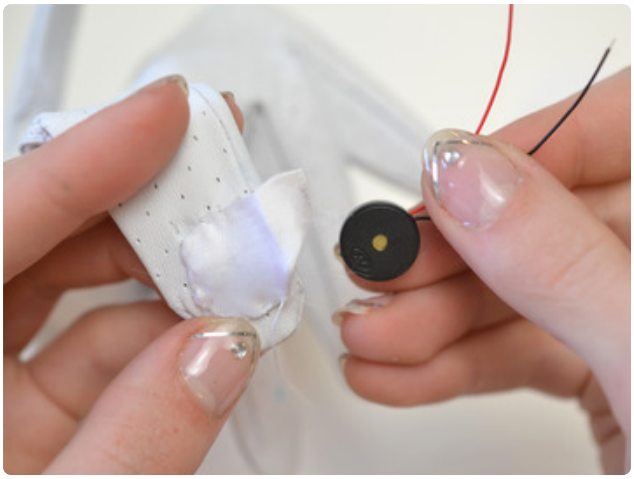
Put your glove on and establish what spots make contact with the table, then mark those spots with a pencil.

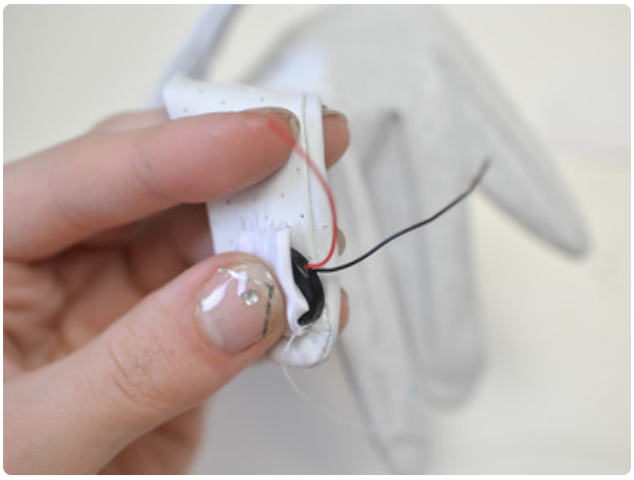
Thread your needle and double the thread over, then tie a knot at the end of the tails.

Stitch through one of your pieces of fabric and affix it to the glove fingertip over the pencil mark with a whip stitch.

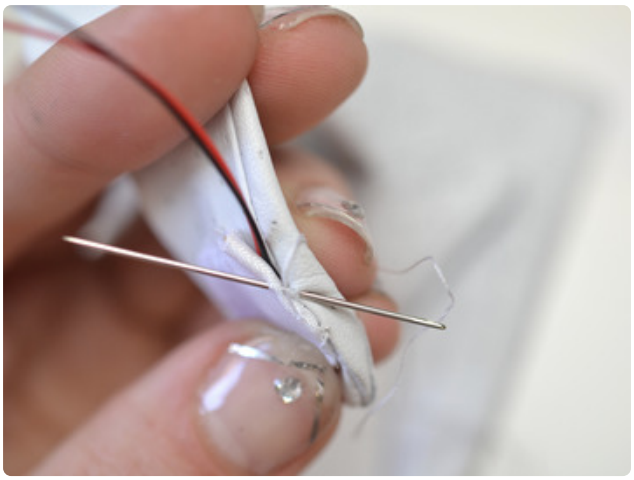
Be careful not to stitch the glove finger closed! Check periodically to be sure your stitches only pierce the intended layer.

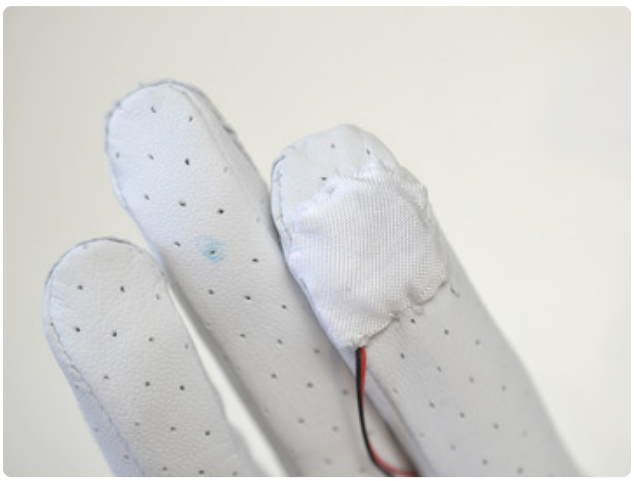
Stitch halfway around the pocket, tucking the seam allowance in a you go.



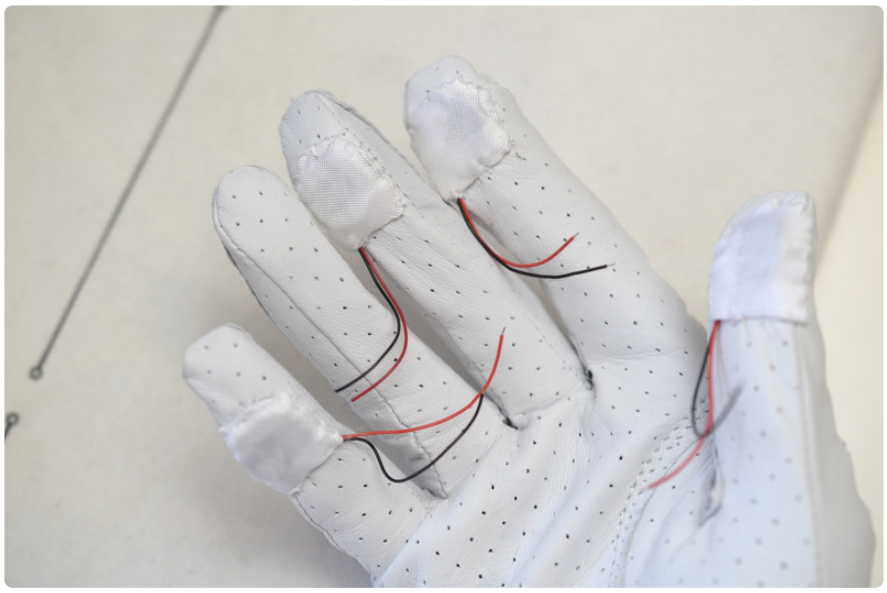


Stick the piezo in the pocket, then finish stitching it shut, leaving the wire sticking out towards the back of the hand. Tie off and cut your thread.

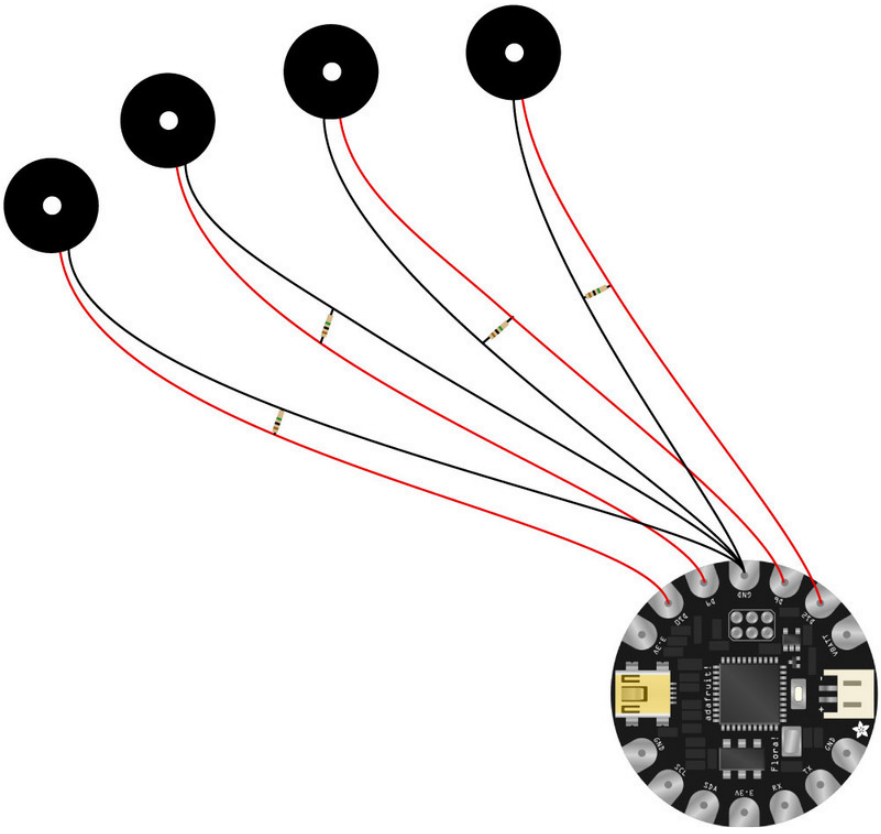




Repeat for the other three piezo pockets, and put your glove on to double check they are tapped when you finger drum. We found the best placement was not necessarily on the pad of the finger, for instance the thumb is around to the side and the pinky is across the first knuckle.



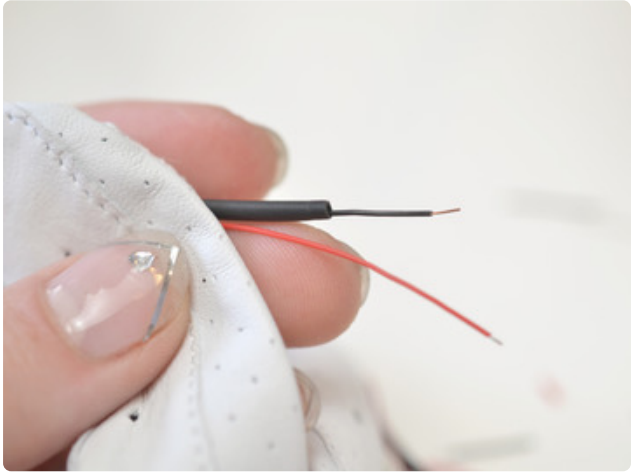
Circuit Diagram

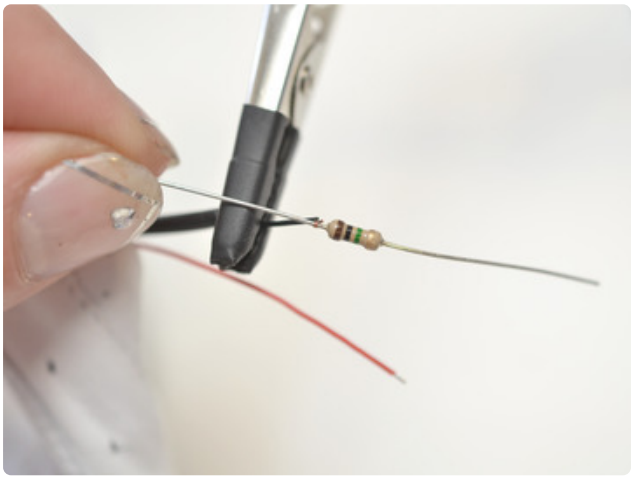


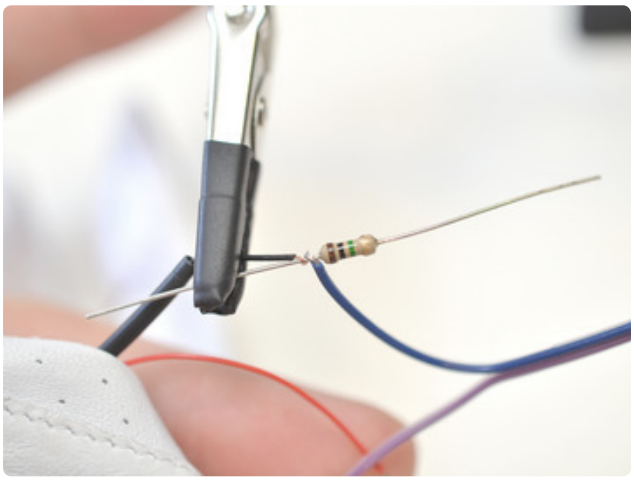
Click to enlarge.

Four piezos each have a 1M ohm resistor across them, with the red wires each going to a different analog input (labeled D6, D9, D10, and D12).

Solder FLORA circuit





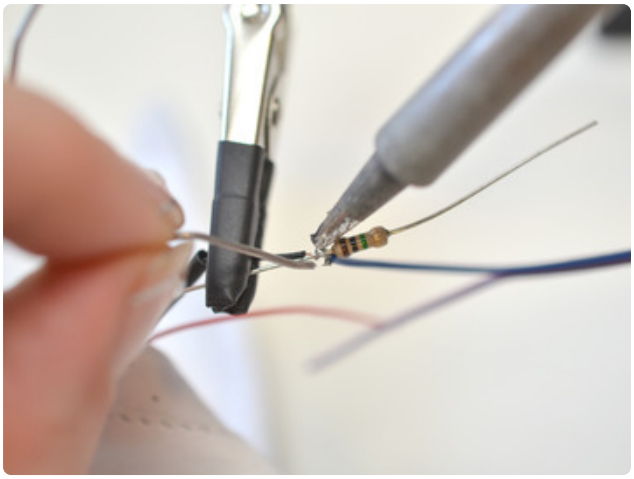


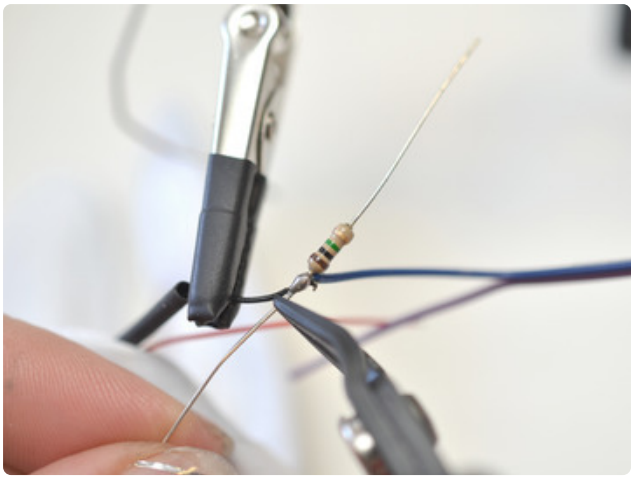
We're going to extend the piezo wires and also attach a pull-down resistor.

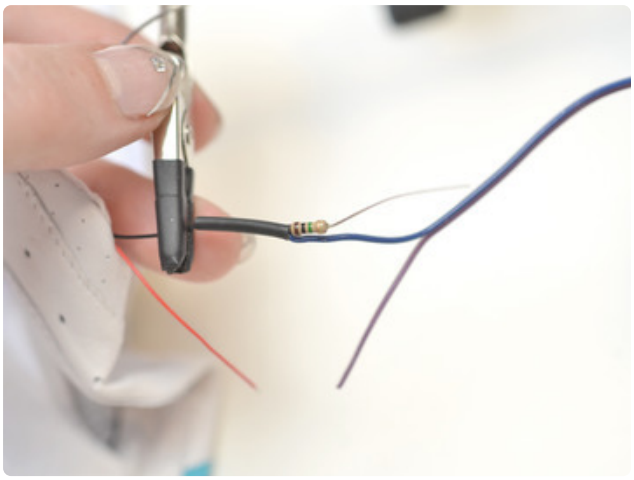
Start with one piezo and trim the black wire a little shorter than the red wire. Slide on a piece of heat shrink tubing and strip the end of the wire.

Wrap the wire end around one of the resistor's leads (doesn't matter which one), and situate the two in the grasp of your third hand tool.

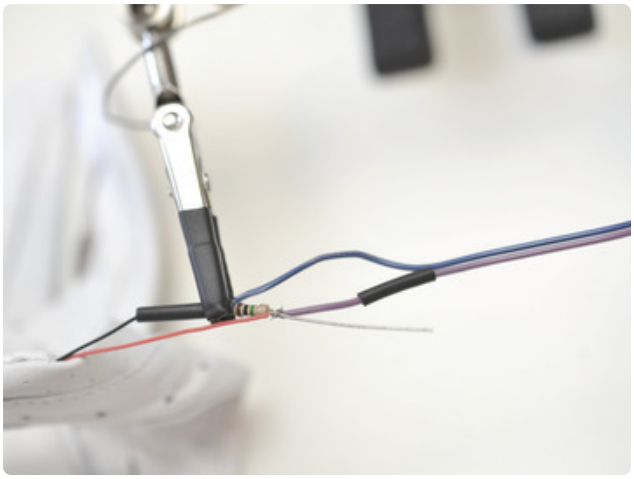
Strip one end on a piece of wire, and wrap this end around the same resistor lead. Solder the two wires to the resistor lead.

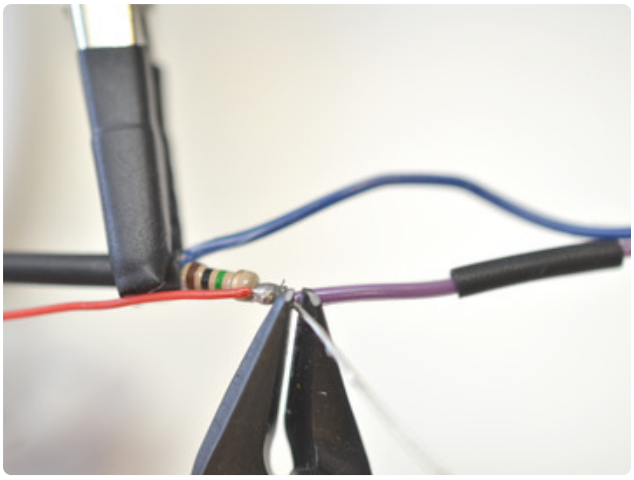






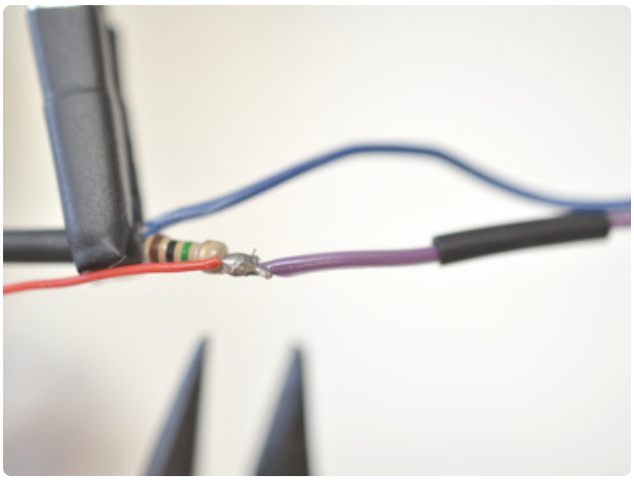
Once soldered, clip the resistor lead short and shield the junction with the heat shrink tubing you added earlier.

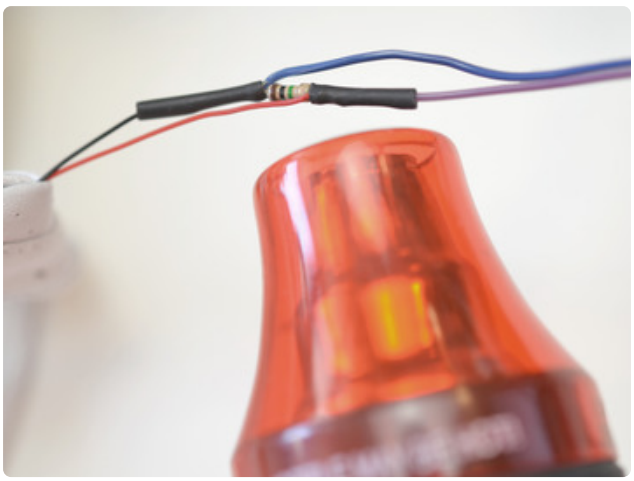




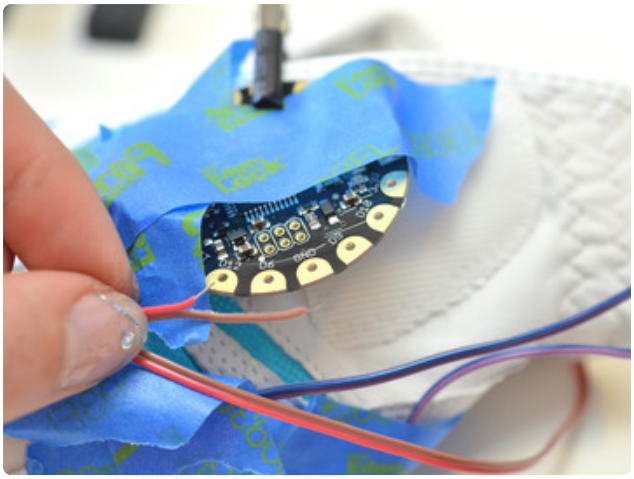
Now it's time for the other wire-- wrap the piezo's red wire and the other wire around the other side of the resistor, remembering to add a piece of heat shrink tubing before soldering.

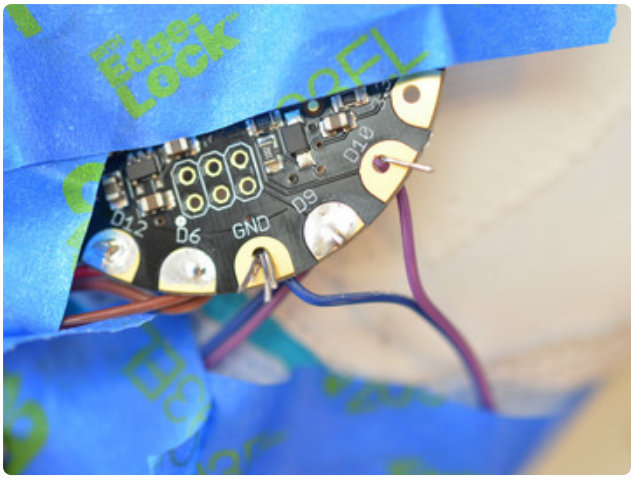
Use a heat gun to shrink both pieces of heat shrink and repeat this process for the three other piezos.





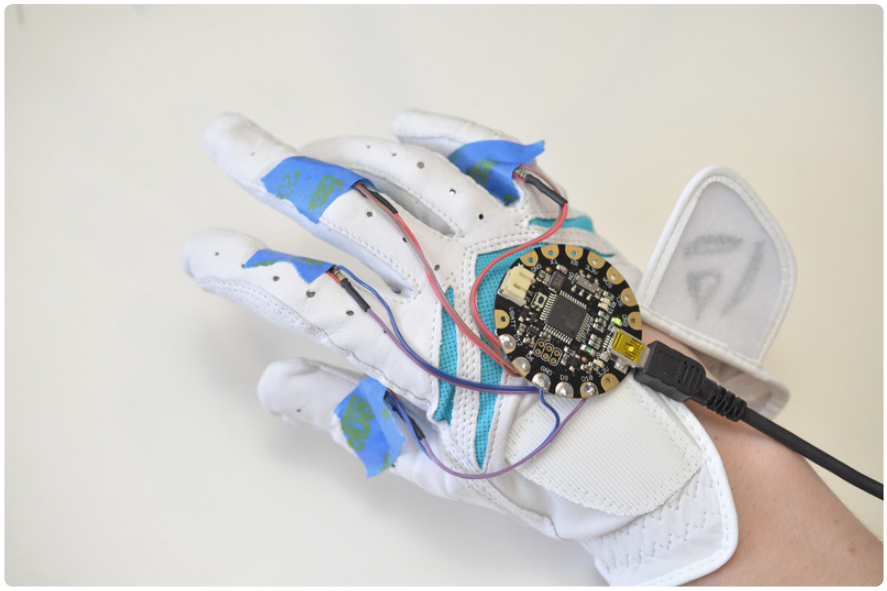
Put the glove on and finesse the wires into a position that reduces strain and lets your hand move. Use painter's tape or masking tape to hold the wires in place as you route the ends towards the analog inputs on FLORA.





Solder each piezo's red wire (or whatever color your red wire became when you extended it) to a different analog input according to the circuit diagram.

Solder all the piezo's black wires to ground.



Before sewing FLORA to the glove, plug in via USB and test your rig using the code on the next page. You'd hate to have to unstitch anything if anything needs re-soldering!

Code

There are many ways to trigger events on your computer with FLORA. The following simple sketch generates configurable keyboard presses suitable for use with things like NanoStudio (https://adafru.it/dz4) or Garage Band:

```
/*
  Piezo Keyboard glove

  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!

  Written by Limor Fried &amp; Becky Stern for Adafruit Industries.
  BSD license, all text above must be included in any redistribution

*/
const int indexFinger = A9; // the piezo is connected to analog pin 9 (aka D9)
const int middleFinger = A7; // the piezo is connected to analog pin 7 (aka D6)
const int thumb = A10; // the piezo is connected to analog pin 10 (aka D10)
const int pinkyFinger = A11; // the piezo is connected to analog pin 11 (aka D12)

const int pins[] = {thumb, indexFinger, middleFinger, pinkyFinger};

char Keys[] =   {'z','x','c','v'};

boolean currentPressed[] = {false, false, false, false};

const int threshold = 40;  // threshold value to decide when the detected sound is
a knock or not

void setup()
{
  //while (!Serial)
  Serial.begin(115200);
  Serial.println("start");
  Keyboard.begin();
}


void loop()
{
  for (int i=0;i&lt;4;i++) {
    delay(1);
    long total1 = 0;
    long start = millis();
    long total =  analogRead(pins[i]);

    // check if we are sensing that a finger is touching
    // and that it wasnt already pressed
    if ((total &gt; threshold) &amp;&amp; (! currentPressed[i])) {
      Serial.print("Key pressed #"); Serial.print(i);
      Serial.print(" ("); Serial.print(Keys[i]); Serial.println(")");
      currentPressed[i] = true;

      Keyboard.press(Keys[i]);
    }
    else if ((total &lt;= threshold) &amp;&amp; (currentPressed[i])) {
      // key was released (no touch, and it was pressed before)
      Serial.print("Key released #"); Serial.print(i);
      Serial.print(" ("); Serial.print(Keys[i]); Serial.println(")");
      currentPressed[i] = false;

      Keyboard.release(Keys[i]);
```

```
    }

    delay(5);
  }
}
```

Adding MIDI support to Flora

Advanced users might want to talk straight USB MIDI-- you can do that with FLORA too! You'll have to mod your Adafruit Arduino 1.0.5 installation. The following instructions are derived from PhilB's upcoming OONTZ tutorial (https://adafru.it/dz5).

To start, download and install PJRC's Teensyduino package **(https://adafru.it/dy3)**, an add-on for the Arduino IDE. This brings support for theirTeensy line of microcontroller boards, including excellent MIDI support. A graphical installer guides you through setup. Make sure to do the Teensification to the Adafruit Arduino IDE so you get both Flora & Teensy support

Installing George Werner's TeeOnArdu package then lets us use the Teensy MIDI library on the FLORA. Ladyada made some changes so this package can use the stock Leonardo/Flora bootloader, saving headaches.

Click to download TeeOnArdu

https://adafru.it/dy4

Installing TeeOnArdu is a little different from other software or libraries. Start by unzipping the **TeeOnArdu-master.zip** file that you downloaded, that much is normal. The unzipped folder that results should contain two items: a "README" file and a sub-folder called "TeeOnArdu."

Locate your Arduino sketchbook — the folder where your own Arduino code and related files go (not the Arduino application, but the folder containing your personal work). The location is a little different for each operating system:

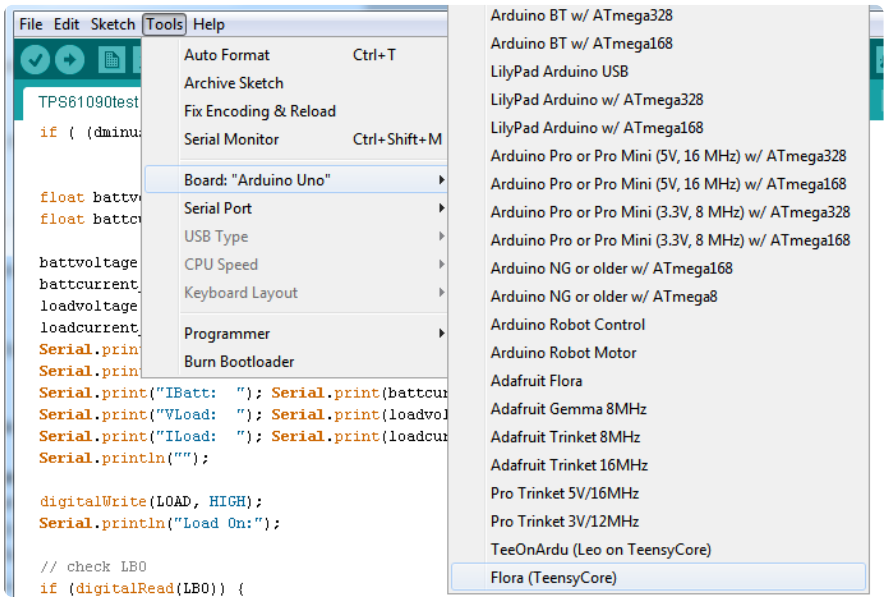
- **Windows:** (home)\My Documents\Arduino
- **Mac:** (home)/Documents/Arduino
- **Linux:** (home)/sketchbook

Inside the sketchbook, create a new folder called "hardware" if one does not already exist.

Move the TeeOnArdu folder (the one alongside the README, not its parent folder) into this hardware folder.

Start the Arduino IDE. Or, if it's already running, exit and restart it.

On the **Tools→Board** menu there should be a new item labeled "Flora (TeensyCore)." Select this.

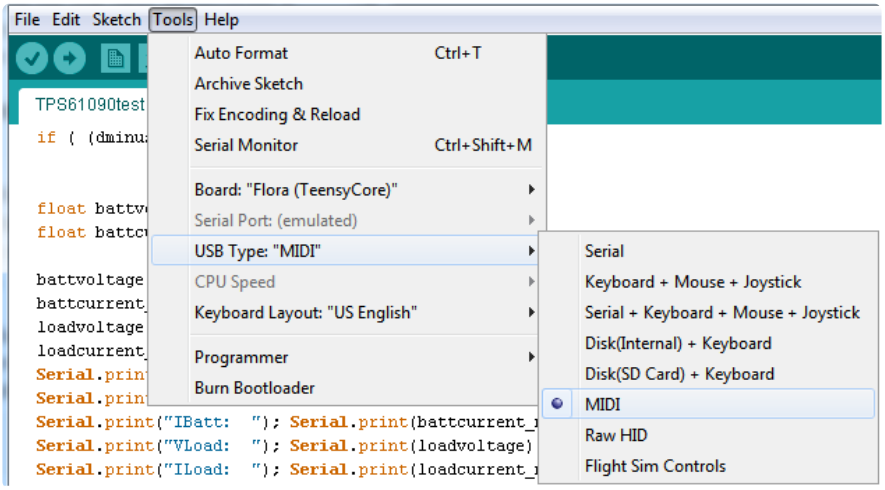


There's still a little song and dance to be done, but it's easier than MIDI The Old Way™.

You should have previously selected **Flora (TeensyCore)** from the **Tools→Board** menu.

Now, from the **Tools→USB Type** menu, select **Serial**. Then unplug and re-plug the USB cable. This resets the FLORA bootloader, and for the next 10 seconds it appears to the system as a serial device. Quickly now...from the **Tools→Serial Port** menu, select the port corresponding to the Arduino board.

From the **Tools→USB Type** menu, now select **MIDI.** It's necessary when compiling MIDI code.



Load the code below into your Arduino IDE. Before uploading the code to the board, **press the mini reset button on the Flora** to launch the bootloader again. There's a 10 second window to then click Upload. If you miss it on the first try, just try again!

If you return to MIDI after working on other Arduino projects and have problems uploading code, remember to do this Serial-then-MIDI selection rigmarole. (You don't need to do this every time you upload code, just when switching back to MIDI from a different Arduino project.)

```
/*
  Piezo MIDI glove

  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!

  Written by Limor Fried, Phillip Burgess, &amp;amp; Becky Stern for Adafruit
Industries.
  BSD license, all text above must be included in any redistribution

*/
#define LED     7 // Pin for heartbeat LED (shows code is working)
#define CHANNEL 1  // MIDI channel number

int note;
const int indexFinger = A9; // the piezo is connected to analog pin 9 (aka D9)
const int middleFinger = A7; // the piezo is connected to analog pin 7 (aka D6)
const int thumb = A10; // the piezo is connected to analog pin 10 (aka D10)
const int pinkyFinger = A11; // the piezo is connected to analog pin 11 (aka D12)

const int pins[] = {thumb, indexFinger, middleFinger, pinkyFinger};

//char Keys[] =   {'z','x','c','v'};

boolean currentPressed[] = {false, false, false, false};

const int threshold = 40;  // threshold value to decide when the detected sound is
a knock or not

void setup()
{
   pinMode(LED, OUTPUT);
  // We dont have Serial or Keyboard available in MIDI mode!!!
  //while (!Serial)
  //Serial.begin(115200);
  //Serial.println("start");
  //Keyboard.begin();
}


void loop()
{
  for (int i=0;i&amp;lt;4;i++) {
    delay(1);
    long total1 = 0;
    long start = millis();
    long total =  analogRead(pins[i]);

    // check if we are sensing that a finger is touching
    // and that it wasnt already pressed
    if ((total &amp;gt; threshold) &amp;amp;&amp;amp; (! currentPressed[i])) {
      //Serial.print("Key pressed #"); Serial.print(i);
```

```
      //Serial.print(" ("); Serial.print(Keys[i]); Serial.println(")");
      currentPressed[i] = true;
      //Keyboard.press(Keys[i]);
       //note = LOWNOTE + (i &amp; 0x0F) * 8;
      usbMIDI.sendNoteOn(60+i, 127, CHANNEL);
    }
    else if ((total &lt;= threshold) &amp;&amp; (currentPressed[i])) {
      // key was released (no touch, and it was pressed before)
      //Serial.print("Key released #"); Serial.print(i);
      //Serial.print(" ("); Serial.print(Keys[i]); Serial.println(")");
      currentPressed[i] = false;
      //Keyboard.release(Keys[i]);
      usbMIDI.sendNoteOff(60+i, 0, CHANNEL);
    }

    delay(5);
  }
  while(usbMIDI.read()); // Discard incoming MIDI messages
}
```

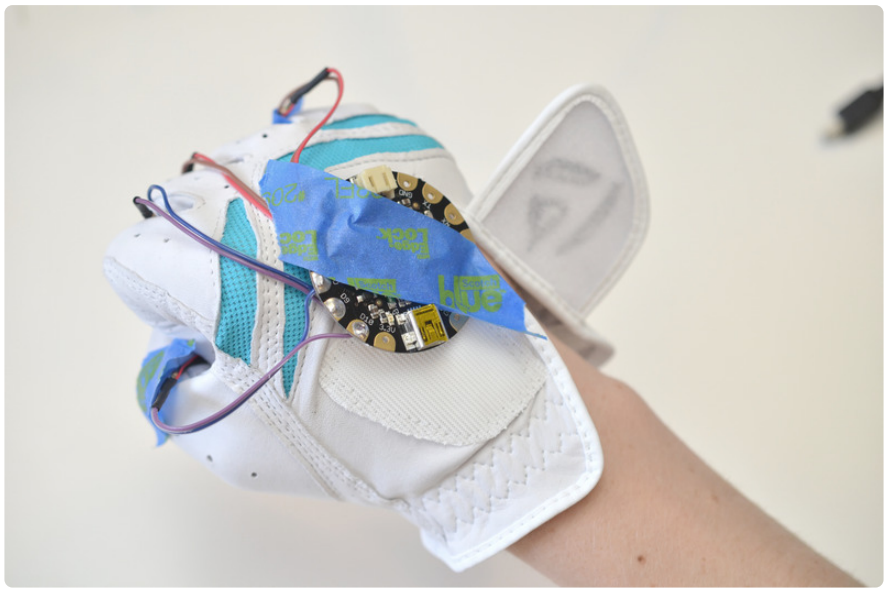
Now you'll need software on the computer at the other end of that cable.

This requires either a software synthesizer — a program that generates sounds using your computer's audio hardware — or a MIDI patchbay utility, which forwards the MIDI data to a hardware synthesizer.

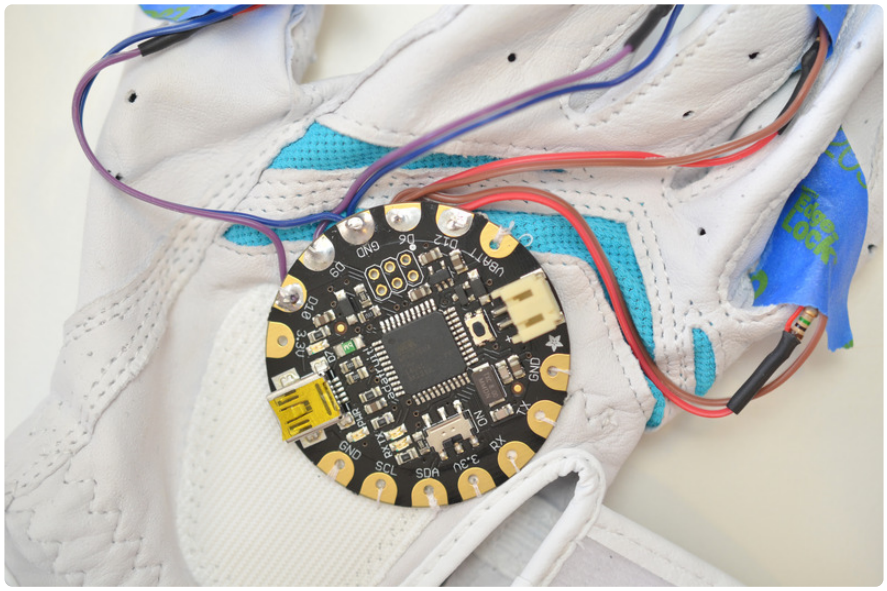
If you've done music on your computer before, you probably already have one or both of these tools. Otherwise, there are a ton of options out there, many of them free. For example, on Windows there's the freeware Firebird2. Mac has the very basic SimpleSynth or the soup-to-nuts GarageBand. Even Linux has options. Google around for "software synthesizer", "MIDI", "free" and "Windows", "Mac" or "Linux" and see what you find.

There are so many options, we can't walk you through this for every app or operating system. It's usually pretty straightforward though, we're confident you can find something and get it installed and running.

Finishing Touches



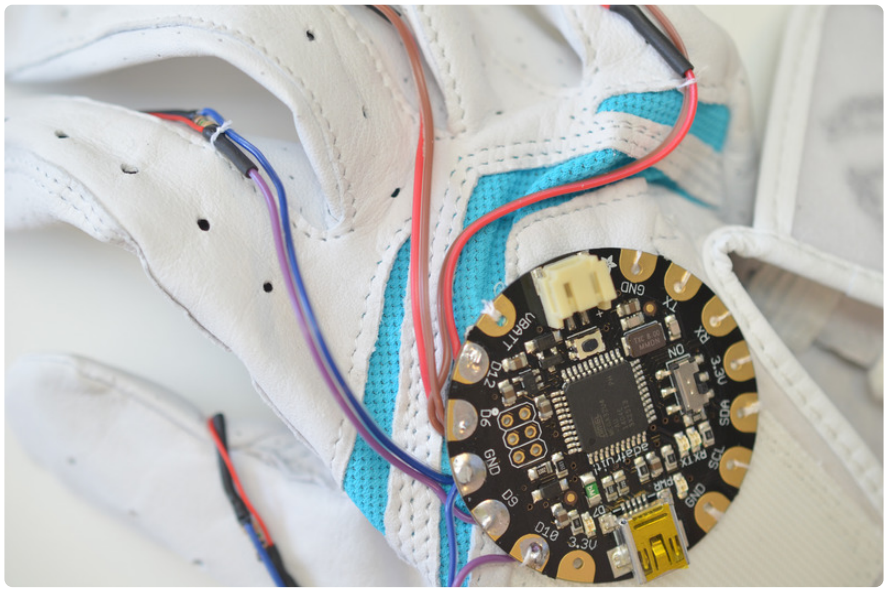
Once your glove is functioning properly, it's time to tack everything down. Put the glove on and position FLORA so that the wires don't tug when you make a fist. Tape it down so it stays put before stitching.



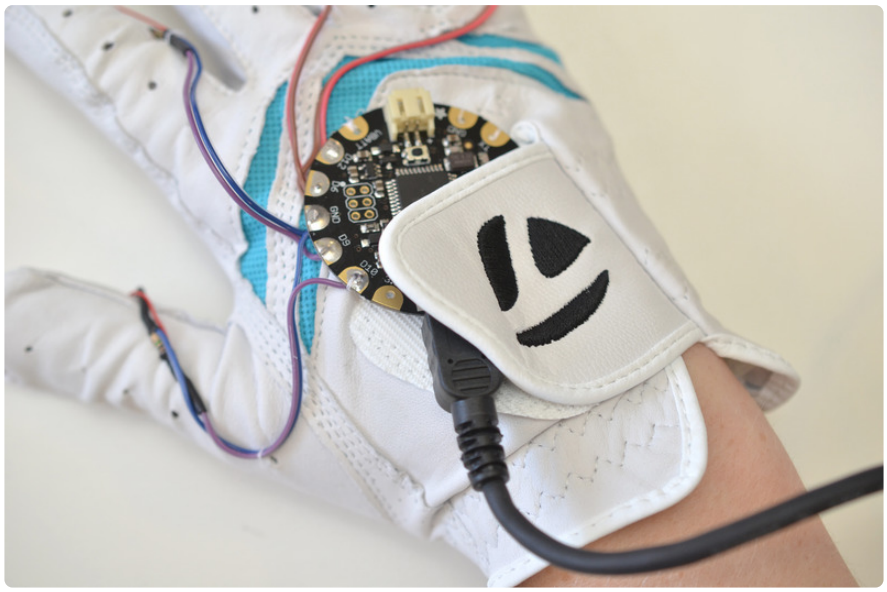
Use plain thread to stitch FLORA's unused pads to the glove. On the side where all the wires come in, stitch around the wires instead of through the pads.

Tack the wires in place with strategic stitches along their lengths.

Remove the tape and try on your completed drum glove!



Use It!



Your USB cable will be plugged in while you use the drum glove. You may want to grab a USB extension cable for more freedom of movement! For a similar, yet wireless, project, try the 3D Printed Wireless MIDI Controller Guitar (https://adafru.it/dz3).