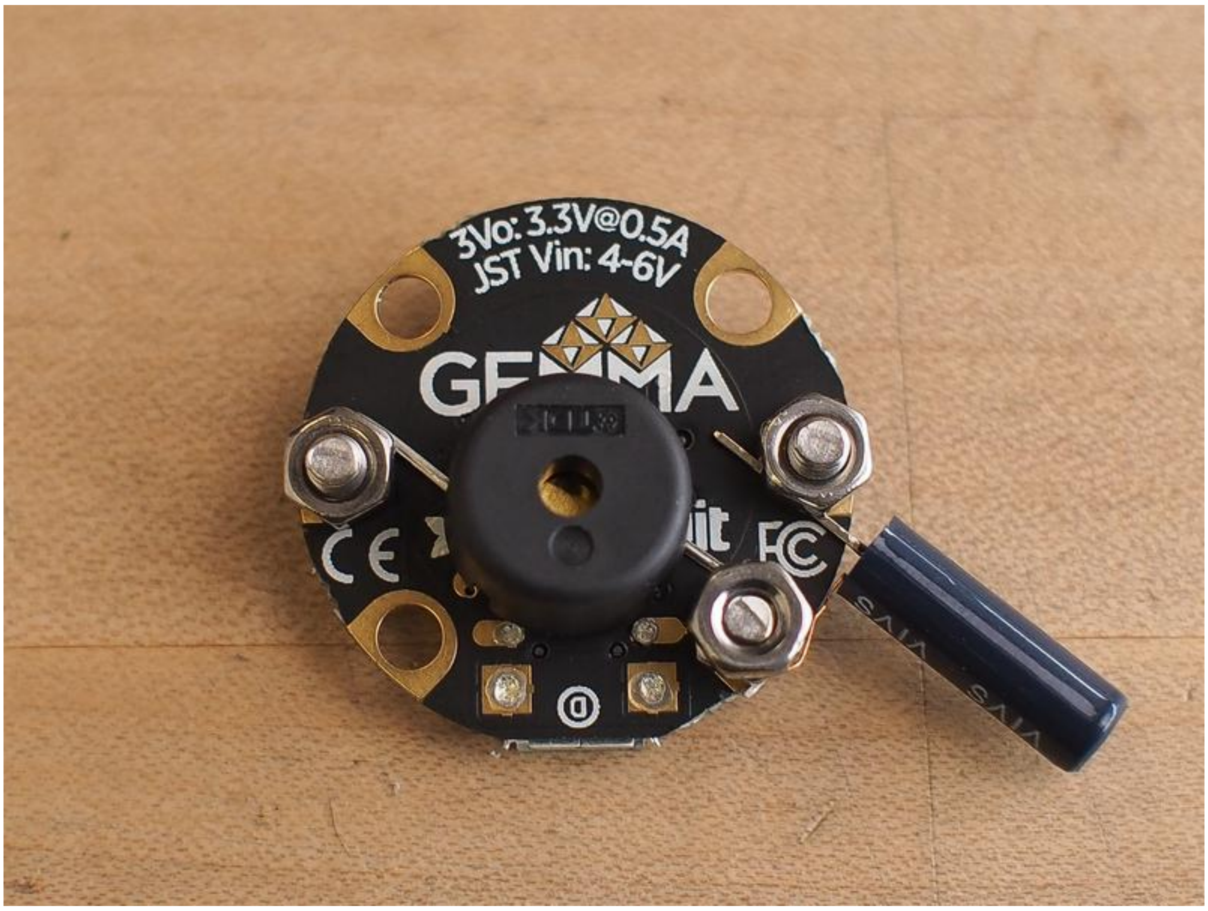
Gemma M0 Vibration Sensor Motion Alarm

Created by John Park

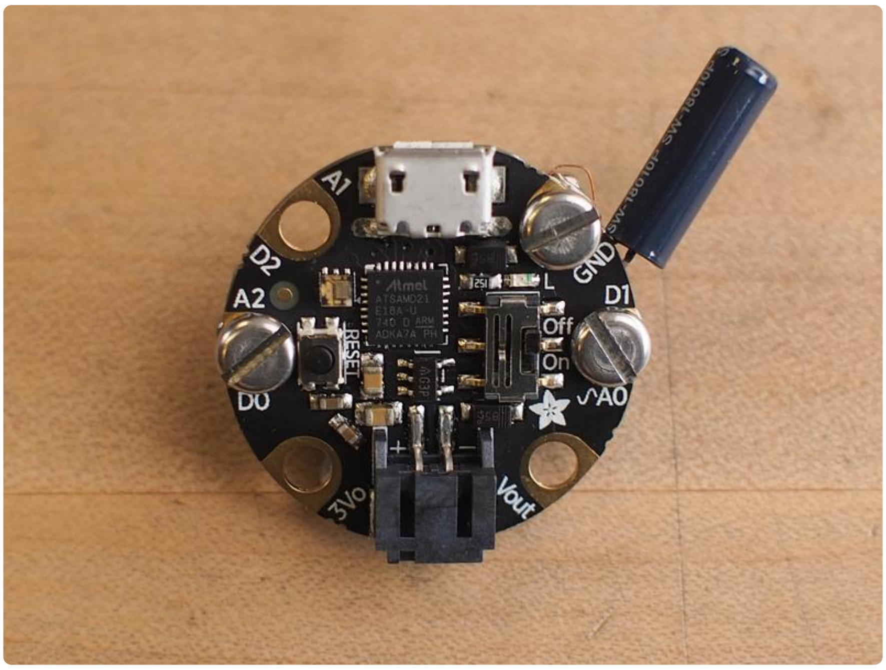


https://learn.adafruit.com/motion-alarm

Last updated on 2023-08-29 03:41:03 PM EDT

Table of Contents

Overview



You can catch enemy spies in the act of touching your stuff by building a motion alarm with a vibration switch, Gemma M0, and a piezo buzzer! Any time it is moved enough to trigger the vibration switch, it will sound the alarm!!

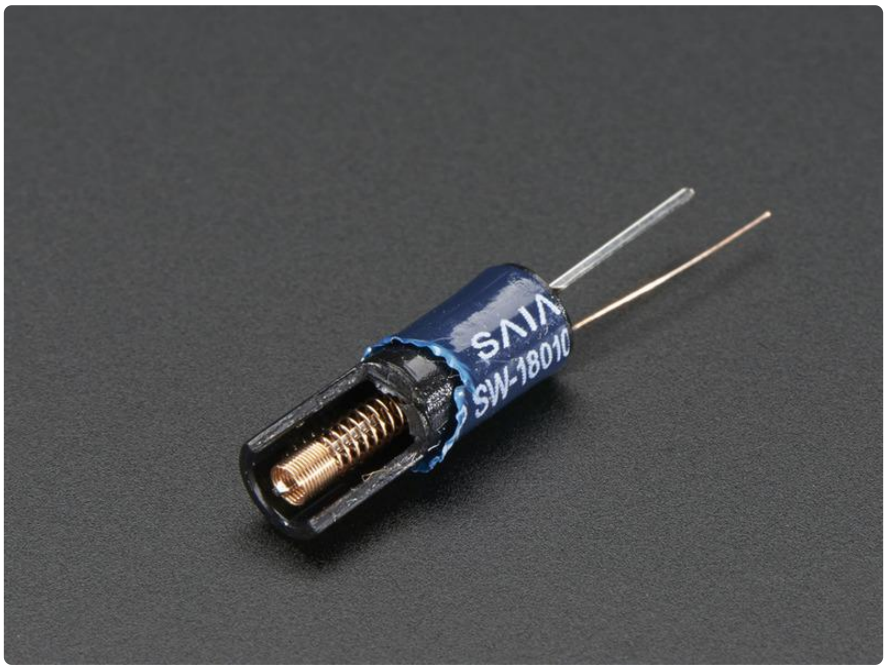
1 x Gemma M0 Small, round microcontroller	https://www.adafruit.com/product/3501
1 x Fast Vibration Sensor Switch Easy to trigger	https://www.adafruit.com/product/1766
1 x Piezo Buzzer Fully enclosed	https://www.adafruit.com/product/1740
1 x 3x AAA Battery Holder with On/Off switch	https://www.adafruit.com/product/727
1 x AAA batteries 3 pack Alkaline power	https://www.adafruit.com/product/3520
1 x USB Cable 6” A/MicroB	https://www.adafruit.com/product/898

You'll also need three M3 x 8mm screws and four nuts to attach parts to the Gemma M0.

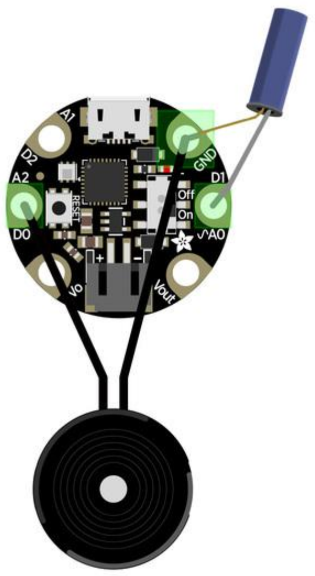
Make the Alarm

How It Works

A vibration switch is a very small cylindrical sensor that has a stiff piece of metal surrounded by a spring. The spring normally doesn't touch the stiff leg, so the circuit is open. When it is moved, the spring will vibrate a bit, touching the stiff leg, and closing the circuit. You can see the copper leg with its coiled spring surrounding the silver colored leg in this cutaway photograph.

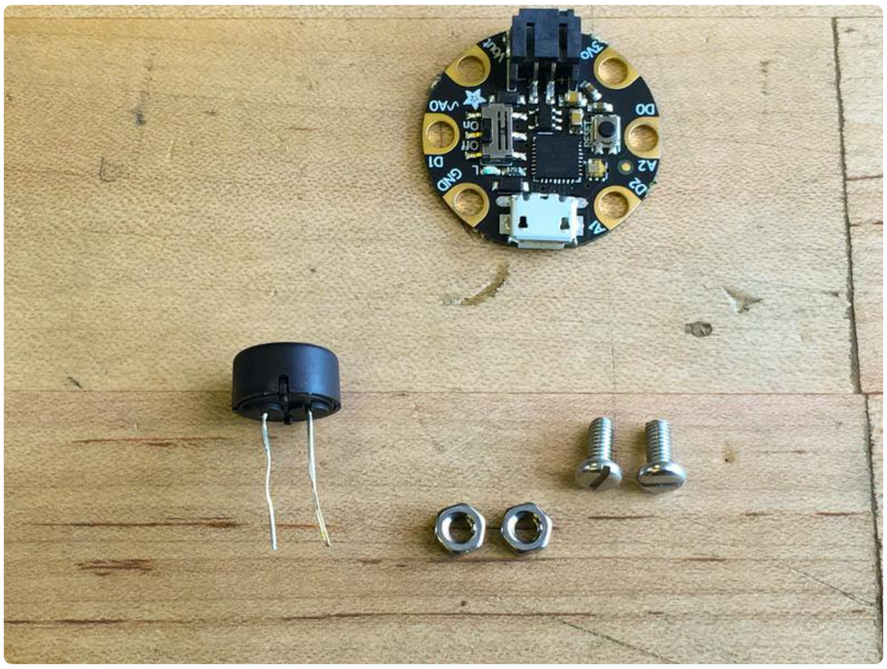


By connecting one leg to analog pin A0 and the other to GND, we can code the Gemma M0 so that it monitors the analog value of the A0 input pin — when the spring vibrates it will send the analog pin to ground, and so the analog voltage value will decrease. When the value goes below a certain threshold, the code will send a pulse width modulated (PWM) tone signal to the piezo buzzer on D0 to make it sound the alarm.



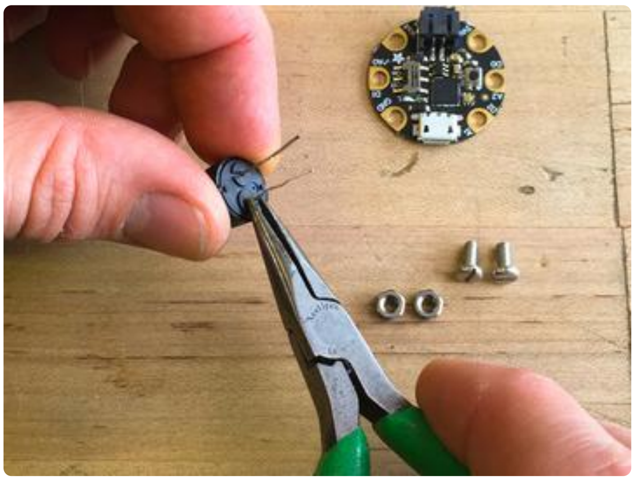
Assembly

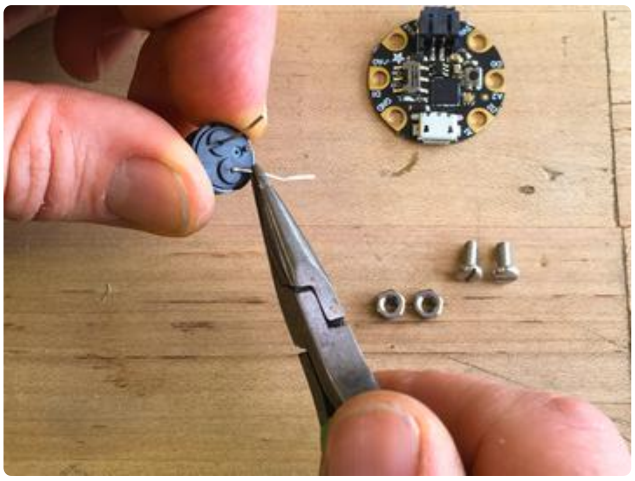
The Gemma M0 can't make any sounds on it's own, so let's connect the piezo buzzer to it so we can hear it!

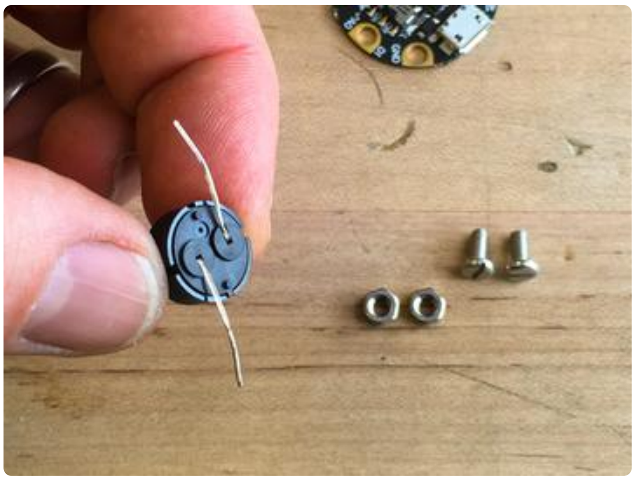


Bending Legs

We'll use the two M3 screws and nuts to connect the buzzer's legs to the D0 and GND pads of the Gemma M0.







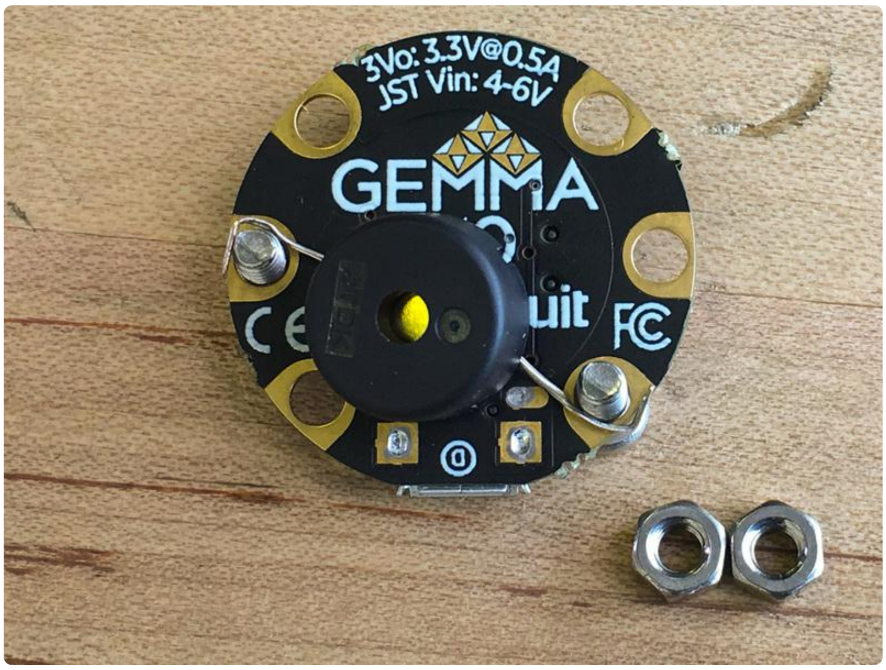
Use pliers (or your fingers) to bend the legs down, carefully, making sure not to break them off.

While the legs will bend easily to their new position, trying to move them back to their original position may be more stress than they can take.

Each leg can fit nicely into a small groove molded into the plastic case.

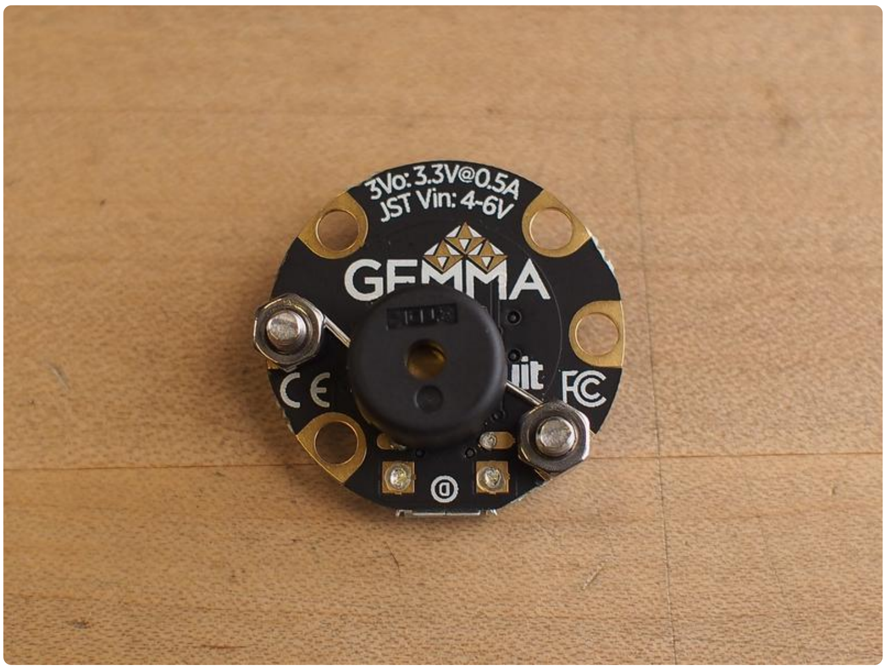
Make Feet

Use the pliers or the screws themselves to form small hook-like feet at the end of each leg, as shown here, so they will connect to D0 and GND. (You can check the other side of the board to see the pad names on the silkscreen.)



It doesn’t matter which leg of the buzzer goes to which pad, it is not polarized.

Thread the nuts on and tighten them, being careful that the legs don't touch any copper pads on the board other than their respective pin assignments.

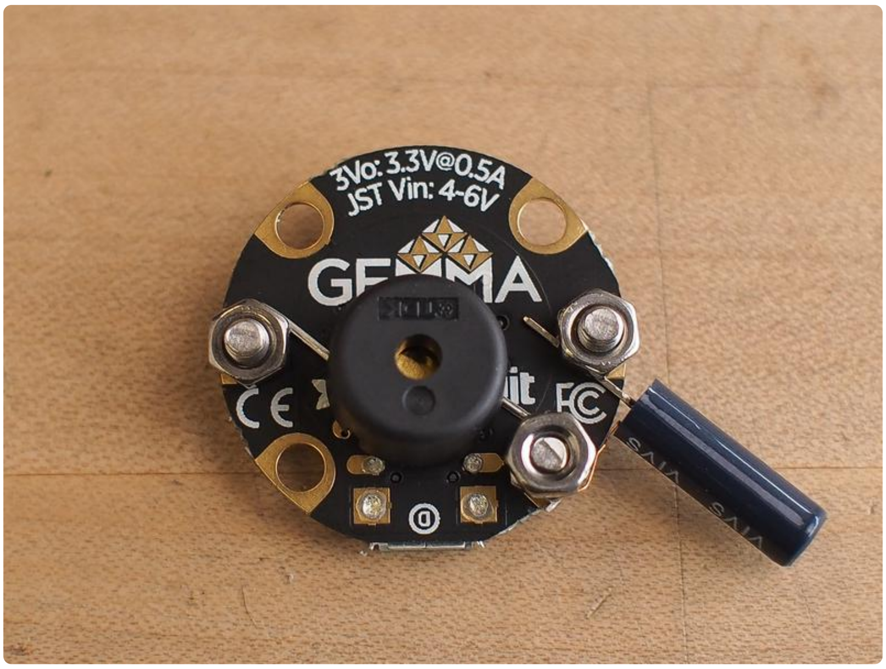


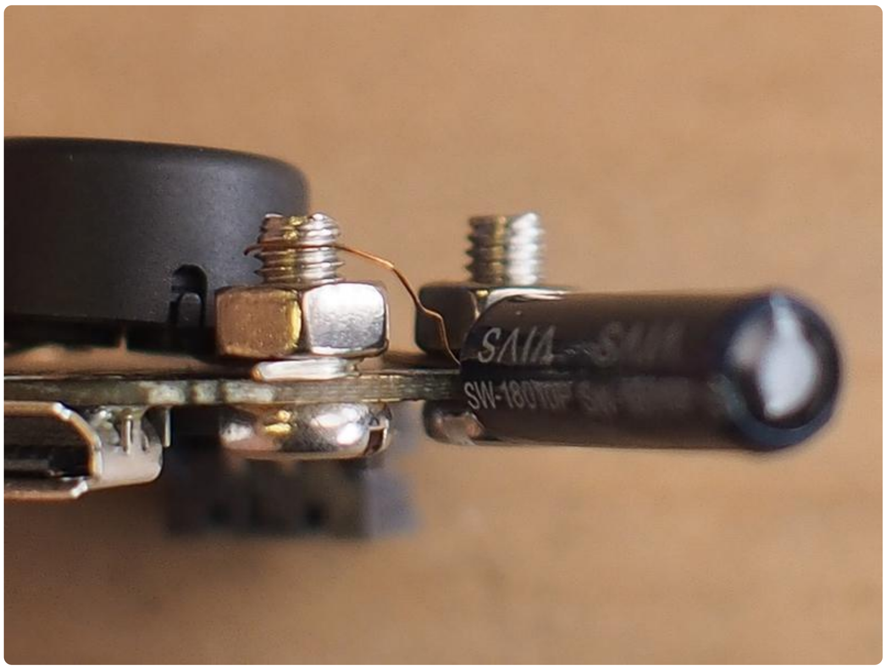
Fast Vibration Sensor

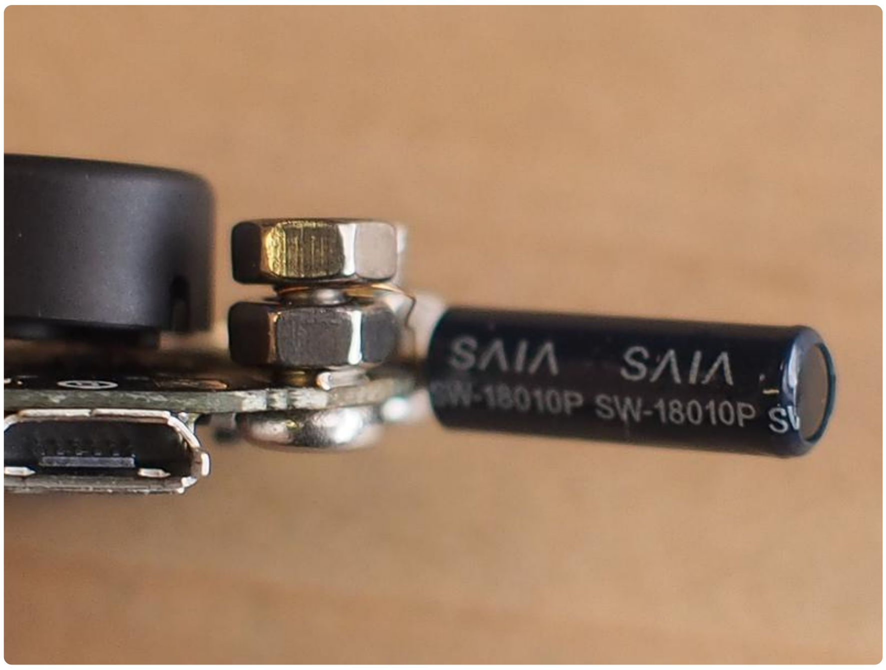
OK, now that the piezo is connected, we'll add the vibration switch.

We’ll connect the vibration switch to A0 with the thicker, silver colored leg. Use a screw and nut to connect it to the Gemma M0 as shown.

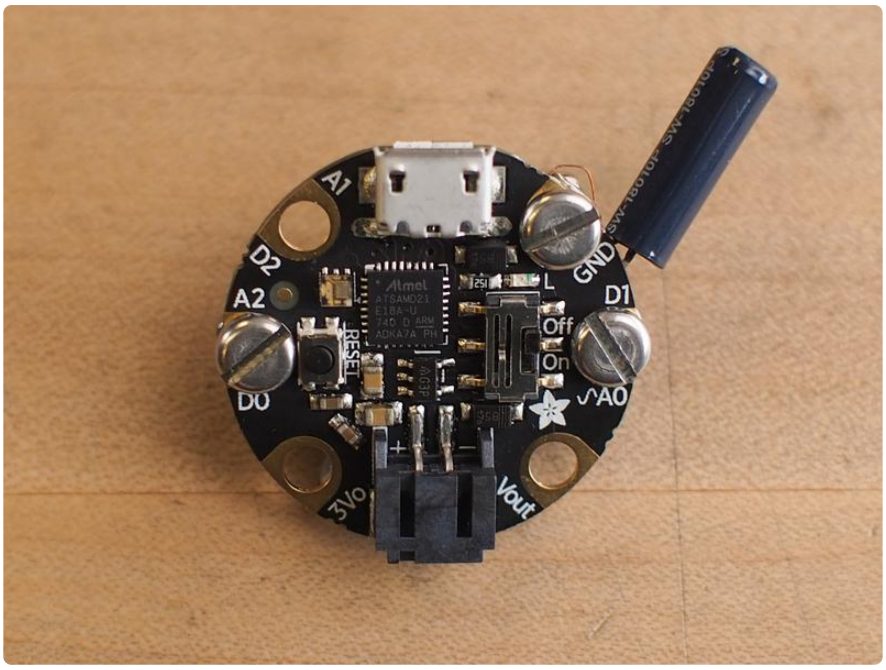
You'll use an extra nut to connect the thin wire of the vibration sensor to the same GN D screw that's being used for the piezo buzzer.






SAIA SAIA
SW-18010P SW-18010P

SAIA SAIA
SW-18010P SW-18010P



That’s all there is to it — next we’ll program it!

Code the Alarm with CircuitPython

CircuitPython Prep

Follow this guide https://learn.adafruit.com/adafruit-gemma-m0/circuitpython () to get started with coding the Gemma M0 in CircuitPython. Install the latest release version of CircuitPython on the board. You may also want to install the Mu editor https://learn.adafruit.com/adafruit-gemma-m0/installing-mu-editor () for your coding needs.

Once you can successfully code in Mu and upload to the board, return here.

Alarm Code

The alarm code will do two fundamental things: measure the analog voltage value on pin A0 to see when it drops below a certain threshold, and use PWM (pulse width modulation) to play alarm tones on the piezo buzzer over pin D0.

Copy the code below, paste it into Mu, and then save it to your Gemma M0 as code.py.

```
# SPDX-FileCopyrightText: 2018 John Edgar Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Motion Sensor Alarm
# uses Gemma M0, vibration sensor on A0/GND, & piezo on D0/GND
import time
import pwmio
from analogio import AnalogIn
import board

piezo = pwmio.PWMOut(board.D0, duty_cycle=0, frequency=440,
                       variable_frequency=True)

vibrationPin = AnalogIn(board.A0)


def get_voltage(pin):
    return (pin.value * 3.3) / 65536


while True:
    print((get_voltage(vibrationPin),))
    vibration = get_voltage(vibrationPin)

    if vibration < 1:  # the sensor has been tripped, you can adjust this value
        # for sensitivity
        for f in (2620, 4400, 2620, 4400, 2620, 4400, 2620, 4400):
            piezo.frequency = f
            piezo.duty_cycle = 65536 // 2  # on 50%
            time.sleep(0.2)  # on 1/5 second
            piezo.duty_cycle = 0  # off
            time.sleep(0.02)  # pause
    time.sleep(0.01)  # debounce delay
```

Test the alarm by touching or tapping it. As soon as it senses a bit of movment, BEEP BEEP BEEP BEEP BEEP!!!

Use the Alarm

Now, you’re ready to deploy the alarm. Hook up the battery pack, set the alarm in your bag or secret hollow book, and then turn it on. When your overly curious enemy rummages around in your things, the piercing alarm will let you catch them in the act!