



# Screaming Cauldron

Created by John Park



<https://learn.adafruit.com/screaming-cauldron>

Last updated on 2024-06-03 02:13:41 PM EDT

# Table of Contents

Overview	3
<hr/>	
• Parts and Materials	
Build the Circuit	5
<hr/>	
• Power	
• Main Circuit and Sensor	
• NeoPixel Strip	
Prepare AudioFX Files	10
<hr/>	
Program the Trinket M0 w/ CircuitPython	12
<hr/>	
• Trinket M0 Setup for CircuitPython	
• Installing NeoPixel Library	
• Saving CircuitPython Code	
• Coding the Screaming Cauldron	
• Testing	
Assemble and Use the Cauldron	17
<hr/>	

---

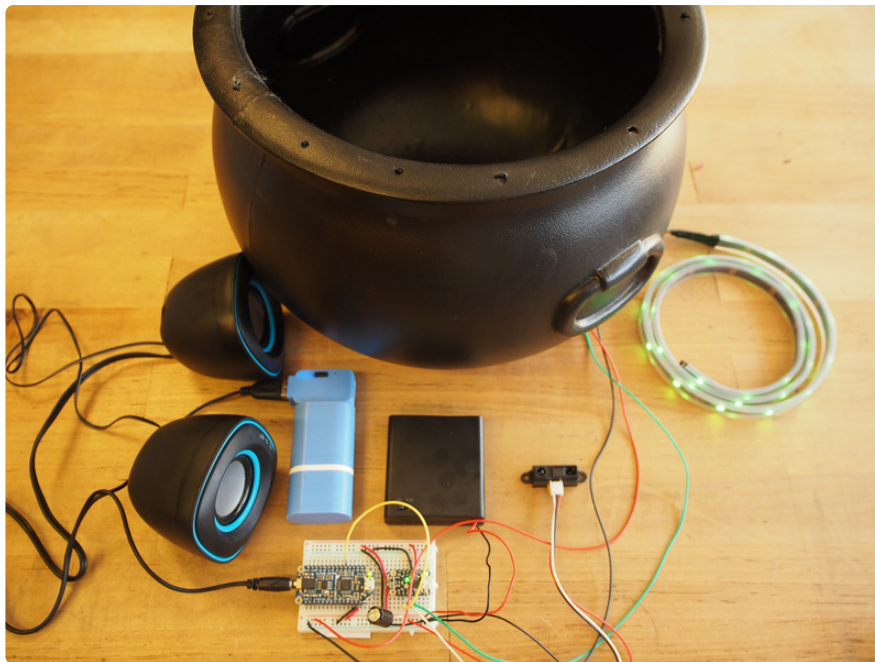
# Overview

Ah, Halloween. A fun time to hand out candy. Also, a very fun time to scare trick-or-treaters!... But how to combine the two? The **Screaming Cauldron** allows you to do both at the same time!

By using an infrared distance sensor to determine hand proximity, the **Trinket MO** microcontroller adjusts the light colors and then lets out a bloodcurdling, amplified scream at just the right moment!



Build your own -- or any other proximity-based sound/light effects -- using this simple project! You'll have it wired up in no time, and the **CircuitPython** code is very simple to adjust to suit your needs.



## Parts and Materials

You can use any kind of container to house the circuit, I chose an inexpensive plastic cauldron. You may also want some black gaffers tape to hold things down, black tissue paper or fabric to hide the circuit, and some white tissue paper to diffuse the NeoPixels.

Here are the parts for the circuit:

**1 x [Adafruit Trinket MO](https://www.adafruit.com/product/3500)**

Microcontroller running CircuitPython

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

**1 x [IR distance sensor](https://www.adafruit.com/product/164)**

Proximity and motion sensing in the 10-80cm range

<https://www.adafruit.com/product/164>

**1 x [Audio FX Sound Board w 2MB Flash](https://www.adafruit.com/product/2133)**

Triggered .WAV audio files

<https://www.adafruit.com/product/2133>

If you have big plans for lots of sound files, check out the [16mb version \(http://adafru.it/2341\)](http://adafru.it/2341). There are also Audio FX Sound Boards with built-in amplifiers in [2mb \(http://adafru.it/2210\)](http://adafru.it/2210) and [16mb variants \(http://adafru.it/2217\)](http://adafru.it/2217), so you have lots of speaker options!

**1 x [Perma-Proto Half-sized Breadboard PCB](https://www.adafruit.com/product/1609)**

PCB for permanent assembly

<https://www.adafruit.com/product/1609>

Black 30 LED strip

1 x [NeoPixel RGB LED strip](https://www.adafruit.com/product/1460)

<https://www.adafruit.com/product/1460>

Black 30 LED strip

---

1 x [4 x AA Battery Holder w On/Off Switch](https://www.adafruit.com/product/830)

<https://www.adafruit.com/product/830>

To power the project using 4 rechargeable AA battery

---

You'll need to supply:

- 4 AA rechargeable batteries for ~5V of power (Alkaline batteries will be a bit too high, around 6V)
- Hookup wire
- soldering iron and solder (if using the perma proto board)

You can use any portable powered speaker you like, or, pick up one of these and a portable USB power supply:

1 x [USB powered speakers](https://www.adafruit.com/product/1363)

<https://www.adafruit.com/product/1363>

With 1/8" stereo audio cable

---

1 x [USB Battery Pack](https://www.adafruit.com/product/1959)

<https://www.adafruit.com/product/1959>

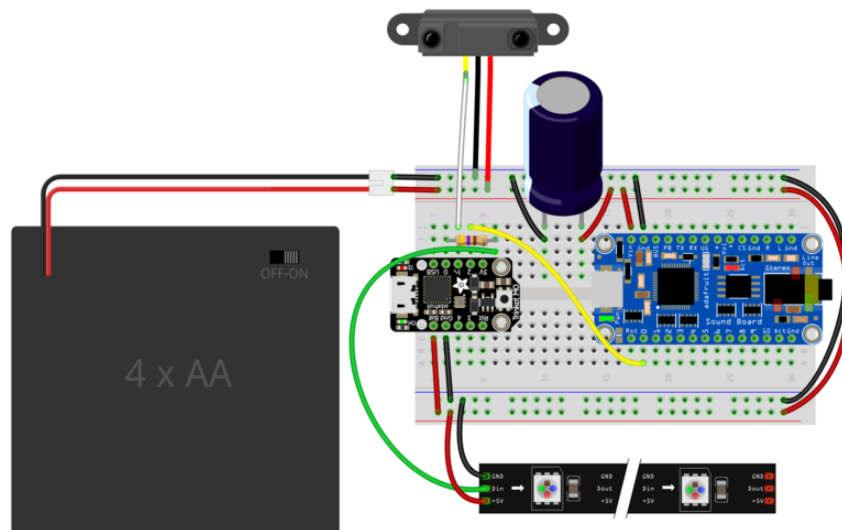
2200mAh Capacity Battery, 5V 1A Output

---

If you'd like to watch a full build of this project, check out this **John Park's Workshop Live** episode:

---

## Build the Circuit



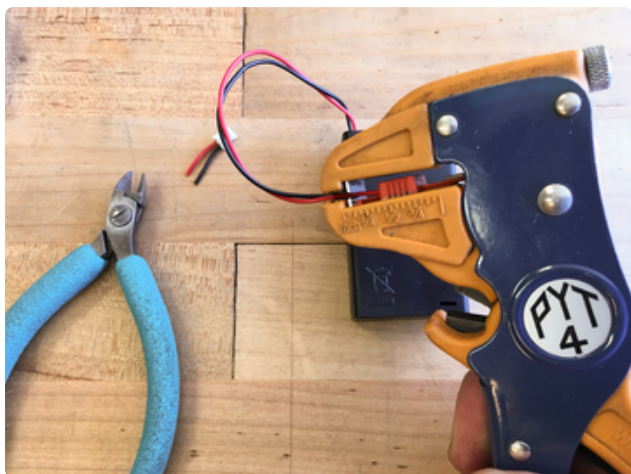
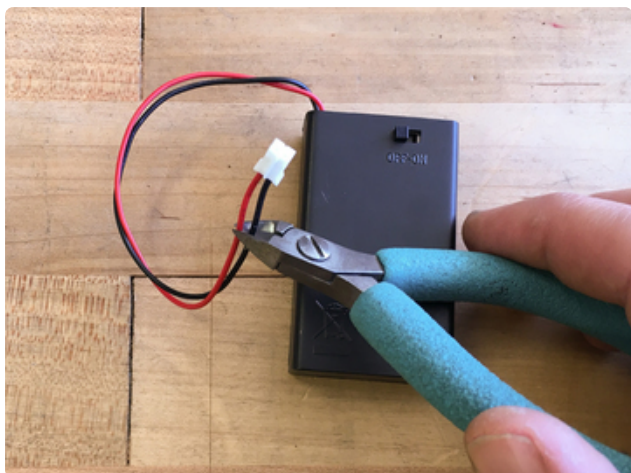
The circuit for the Screaming Cauldron consists of these major components:

- **IR distance sensor** to detect candy grabbing hand proximity
- **AudioFX Sound Board** to play terrifying sound samples through a powered speaker
- **NeoPixel** strip for interactive colored lighting
- **Trinket M0** to act as the brain, telling the NeoPixels to change color and the AudioFX board to play sounds

You'll assemble the circuit as shown in the diagram, either on a breadboard, or soldered to a perma proto board. The breadboard is recommended for making sure everything works properly at first. Then, you can transfer to a perma proto board for a more permanent connection.

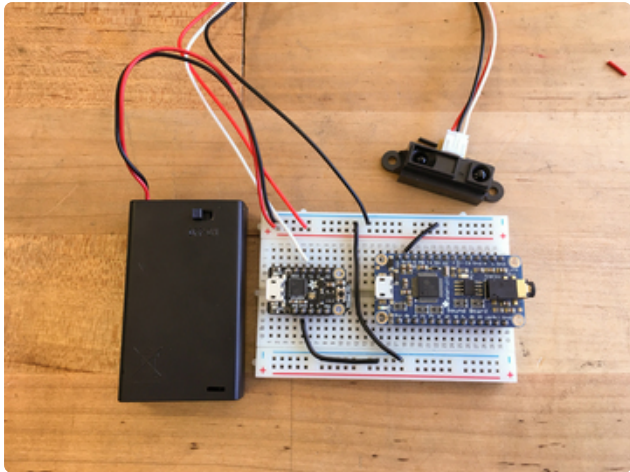


## Power



You can start by snipping the JST connector off of the battery pack so you can attach it directly to the board.

## Main Circuit and Sensor



Solder the header pins on both the Trinket M0 and Audio FX Sound Board as shown here and here, then connect them to the board. Run these connections:

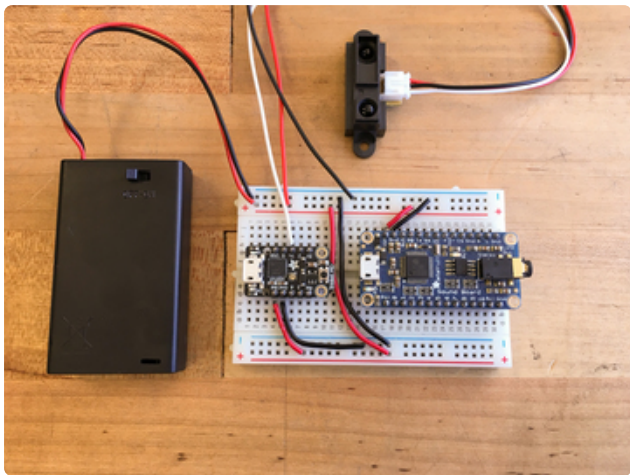
Trinket M0 **GND** to ground rail

Trinket M0 **BAT** to power rail

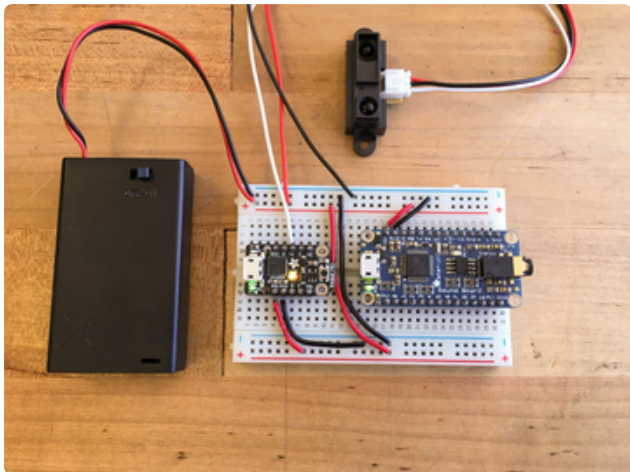
Audio FX **GND** to ground rail (do not use the Audio FX GND at the top right of the board, this is an audio line out ground)

Audio FX **Vin** to power rail

Attach power to the board's power rails, and run jumpers to connect the positive rails and ground rails to their partners across the board.



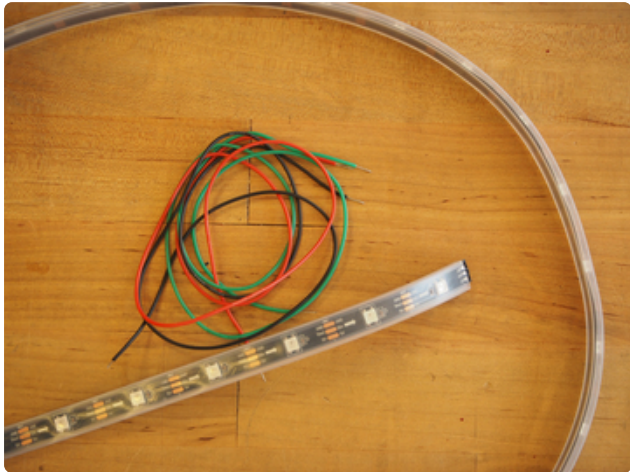
Then, connect the IR sensor's power and ground wires to their respective rails, and the white signal wire to pin D1 on the Trinket M0.



We haven't yet written the code or uploaded audio samples to the FX board, but you can turn on the battery power now to test things out and make sure they're getting powered on, lighting their green LEDs.



## NeoPixel Strip



Now you'll prepare the NeoPixel strip for connection to the circuit.

First, cut and strip some ~8" lengths of red, black, and green wire. Tin the ends, and then solder these connections:

Black to **GND**

Green to **Din**

Red to **5V**

You can use some heat shrink tubing to strengthen the connection point, but this is an optional step.

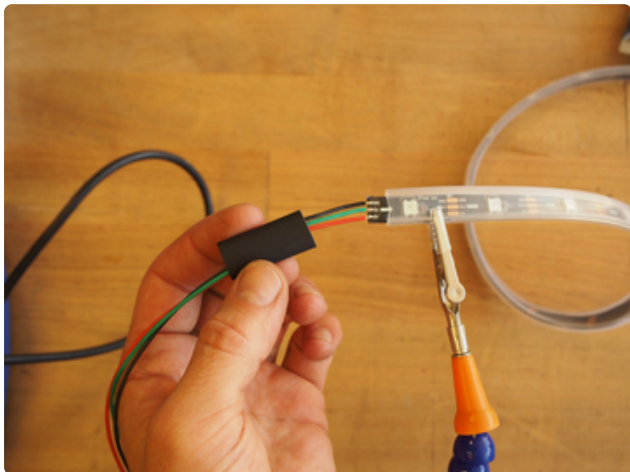
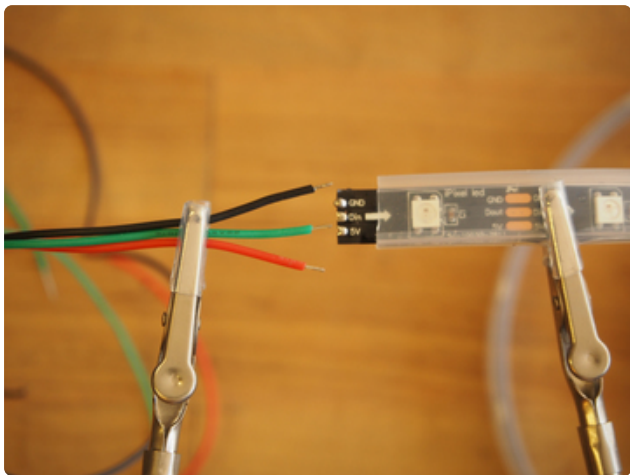
Connect the NeoPixel strip to the board, with these connections:

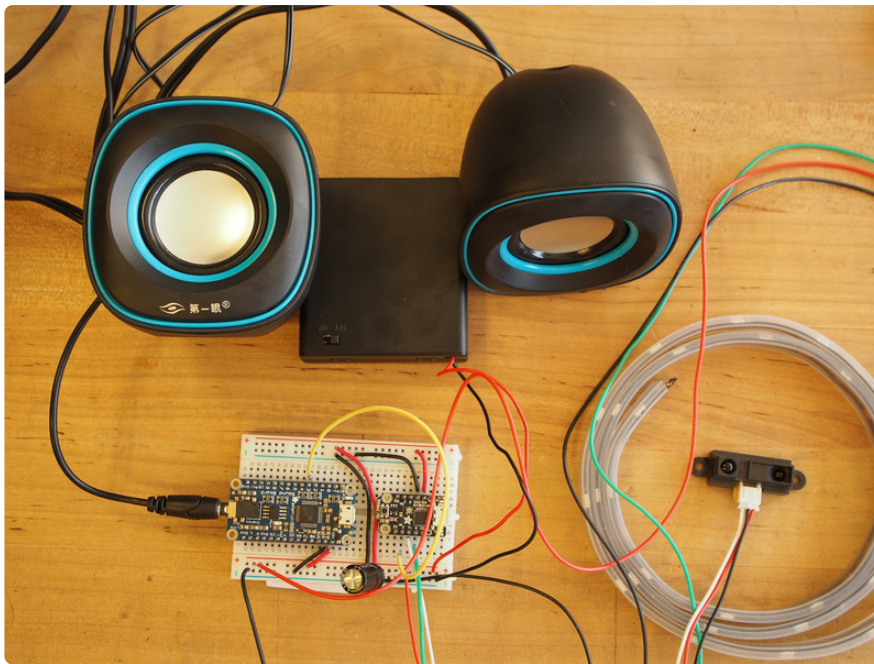
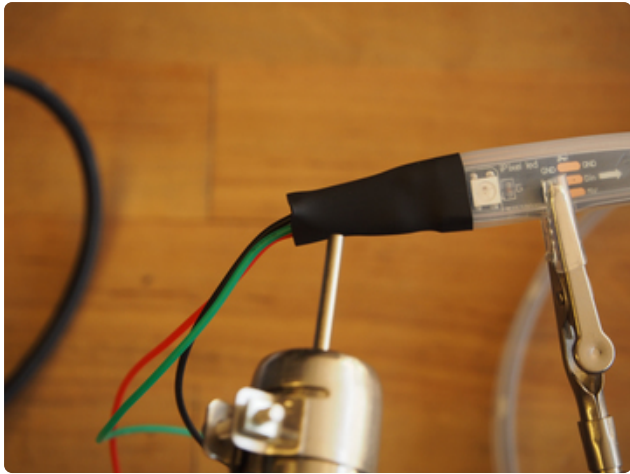
Red to power rail

Black to ground rail,

Green data input to Trinket M0 pin **D0** (ideally through a 470 Ohm resistor)

You will also connect a 1000uF capacitor to the ground and power rails [as per this guide \(https://adafruit.it/ukF\)](https://adafruit.it/ukF), in order to prevent any power issues on the NeoPixel strip.





As shown here and in the circuit diagram, connect the Trinket M0 **D2** pin to the AudioFX Sound Board's **pin 0**. This is how the Trinket M0 will be able to trigger the sounds.

You can now connect the speaker input to the AudioFX Sound Board and get ready to program things!

---

## Prepare AudioFX Files

[This guide will teach you \(https://adafru.it/pqb\)](https://adafru.it/pqb) everything you need to know about triggering audio on the AudioFX board. Each of the eleven input pins can be used to trigger different sound samples with different buttons and sensors, but we'll only be using one of them.

Since we're triggering over a single pin, we can use Play Next Trigger mode or Play Random Trigger mode for multiple sound files. Everything is based upon a naming convention. Download the files below and unzip them to use the four sounds I selected.

**screamingCauldronWavs.zip**

<https://adafru.it/zKF>

This naming convention is the key:

**T00NEXT0.WAV**

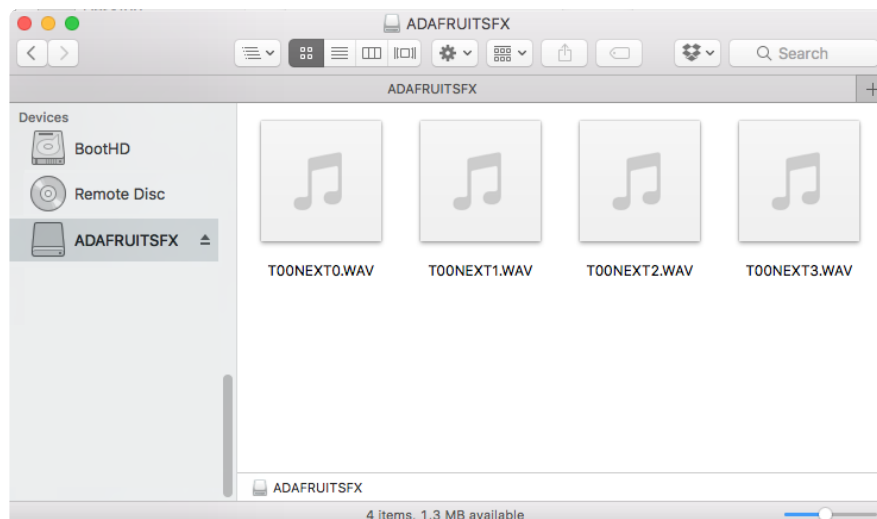
**T00NEXT1.WAV**

**T00NEXT2.WAV**

**T00NEXT3.WAV**

This naming means that the sounds are triggered by pin 0, and will play one after another sequentially by number each time the pin 0 is grounded.

Plug in your Audio FX Sound Board, and then drag the .wav files you unzipped onto the ADAFRUITSFX disk drive that shows up!



If you want to use your own sound files, you can! Record, sample, remix, or simply download files from a sound file sight, such as [freesample.org](https://freesample.org). Then, to make sure you have the files converted to the proper specifications, [check out this guide here \(https://adafru.it/s8f\)](https://adafru.it/s8f) that'll show you how! Spoiler alert: you'll need to make a small, 22Khz (or lower), 16 bit PCM, mono .wav file!

---

# Program the Trinket M0 w/ CircuitPython

## Trinket M0 Setup for CircuitPython

Now it's time to set up the **Trinket M0** for use with [CircuitPython](https://adafru.it/zB0) (<https://adafru.it/zB0>) and our Screaming Cauldron code.

If you haven't already, follow [this guide](https://adafru.it/ABS) (<https://adafru.it/ABS>) to preparing the **Trinket M0**, including updating it with the latest version of **CircuitPython**.

After prepping the **Trinket M0** to run **CircuitPython** we'll also need to add a **NeoPixel** library. [This guide tells you how](https://adafru.it/ABU) (<https://adafru.it/ABU>), as well as providing a good primer on using **NeoPixels** on the **Trinket M0** with **CircuitPython**.

## Installing NeoPixel Library

Download the latest `adafruit-circuitpython-bundle-xxxx.zip` (or newer) from the [releases directory](https://adafru.it/zB-) (<https://adafru.it/zB->) and then unzip it somewhere easy to find, such as your desktop. Then, copy the `neopixel.mpy` file to your CIRCUITPY's `lib` directory on the **Trinket M0**.

Finally, if you had a `neopixel.py` file that was already in that same `lib` directory you can delete it (the mpy is a compressed version)

## Saving CircuitPython Code

Once you've got things working, you can edit the `main.py` file on the **Trinket M0** to adjust what it actually does. No need for a software IDE, complaining tools, or flashing the chip -- when you code with **CircuitPython** all you need is a text editor. Edit the code, save it to the **Trinket M0**, and it immediately runs!

## Coding the Screaming Cauldron

The Trinket M0 will be responsible for a few things:

- Checking the IR distance sensor to see when a hand is getting close and then breaking a threshold
- Controlling the NeoPixel color as the hand gets closer

- Triggering the AudioFX Sound Board to play sounds when a distance threshold is met

Here's the code written in CircuitPython to do all of those things:

```
# SPDX-FileCopyrightText: 2017 John Edgar Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Screaming Cauldron
# for Adafruit Industries Learning Guide

import time

import board
import neopixel
from analogio import AnalogIn
from digitalio import DigitalInOut, Direction

led = DigitalInOut(board.D13) # on board red LED
led.direction = Direction.OUTPUT

aFXPin = DigitalInOut(board.D2) # pin used to drive the AudioFX board
aFXPin.direction = Direction.OUTPUT

aFXPin.value = False # runs once at startup
time.sleep(.1) # pause a moment
aFXPin.value = True # then turn it off

analog0in = AnalogIn(board.D1)
neoPin = board.D0 # pin int0 which the NeoPixels are plugged
numPix = 30 # number of NeoPixels
pixels = neopixel.NeoPixel(neoPin, numPix, auto_write=0, brightness=.8)
pixels.fill((0, 0, 0))
pixels.show()

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

def remap_range(value, left_min, left_max, right_min, right_max):
    # this remaps a value from original (left) range to new (right) range
    # Figure out how 'wide' each range is
    leftSpan = left_max - left_min
    rightSpan = right_max - right_min

    # Convert the left range into a 0-1 range (int)
    valueScaled = int(value - left_min) / int(leftSpan)

    # Convert the 0-1 range into a value in the right range.
    return int(right_min + (valueScaled * rightSpan))

while True:
    distRaw = analog0in.value # read the raw sensor value
    print("A0: %f" % distRaw) # write raw value to REPL
    distRedColor = remap_range(distRaw, 0, 64000, 0, 255)
    distGreenColor = remap_range(distRaw, 0, 64000, 200, 0)
    if distRaw > 40000: # at about 4 inches, this goes off!
        led.value = True
        aFXPin.value = False
        time.sleep(.35)
    else:
        led.value = False
```



```
aFXPin.value = True

# change pixel colors based on proximity

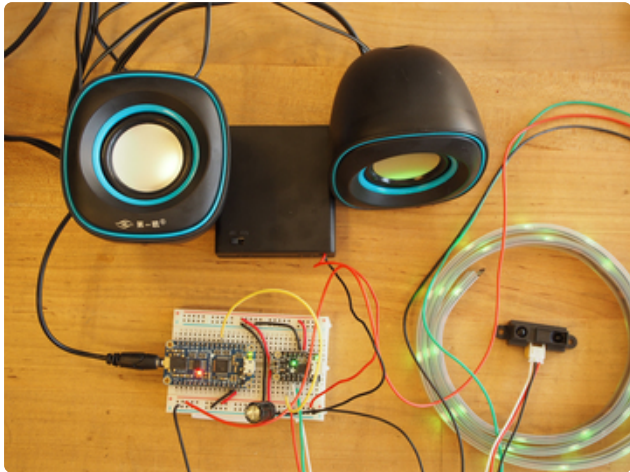
print(distRedColor, distGreenColor, (distRedColor + distGreenColor))
pixels.fill((distRedColor, distGreenColor, 0))

pixels.write()

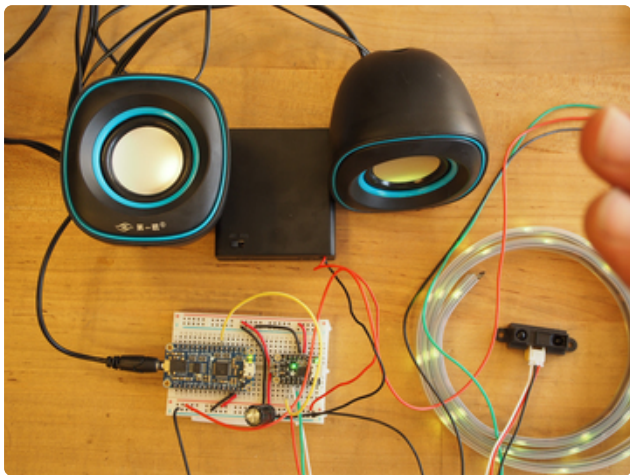
time.sleep(0.1)
```

Copy that code, and then paste it into a new text document in your favorite text/coding editor. Then, save it to your Trinket M0's CIRCUITPY drive as **main.py**

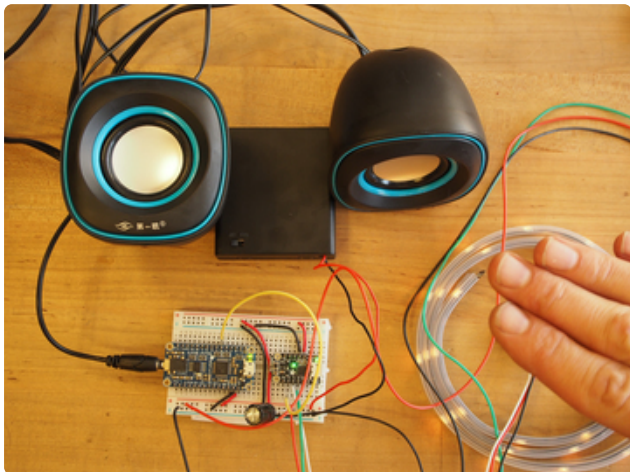
## Testing

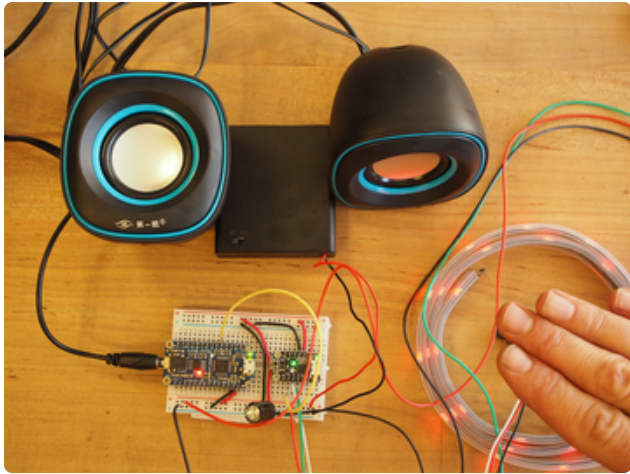


Your circuit is now ready for testing! Make sure the battery pack is turned on, then press the reset button once on the Trinket M0. you should hear the first scream sound trigger upon reboot, and then it will light the NeoPixels green, waiting to detect an unlucky hand!



Move your hand up and down over the IR sensor and watch the color shift from green to red. When you get too close (about 3-4" away) a sound will trigger, in this case the T00NEXT0.WAV sound. Do it again (if you dare!) and you'll trigger T00NEXT1.WAV and so on, until it loops back around to the first sound.





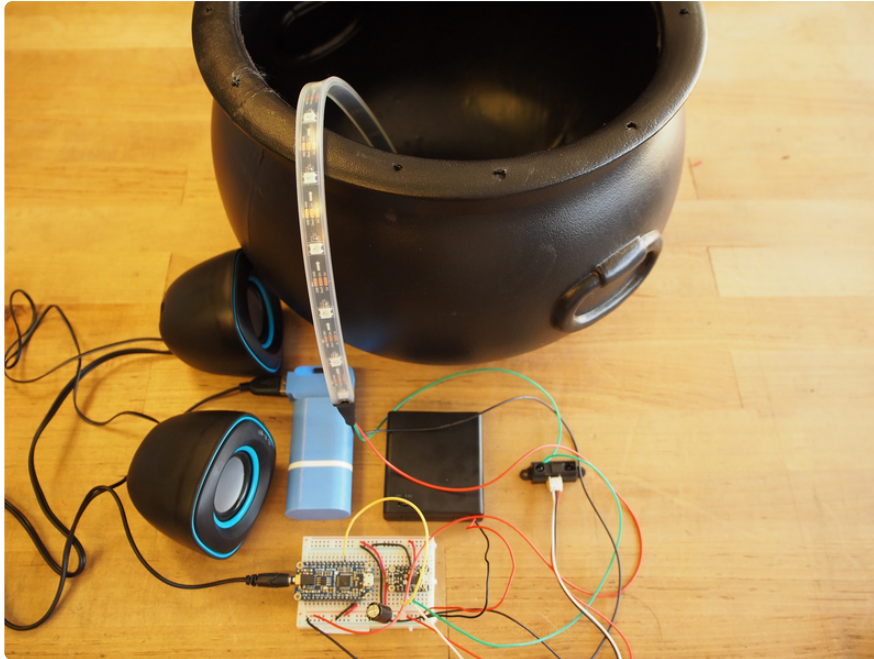
Next, we'll hide the circuit inside the cauldron!





---

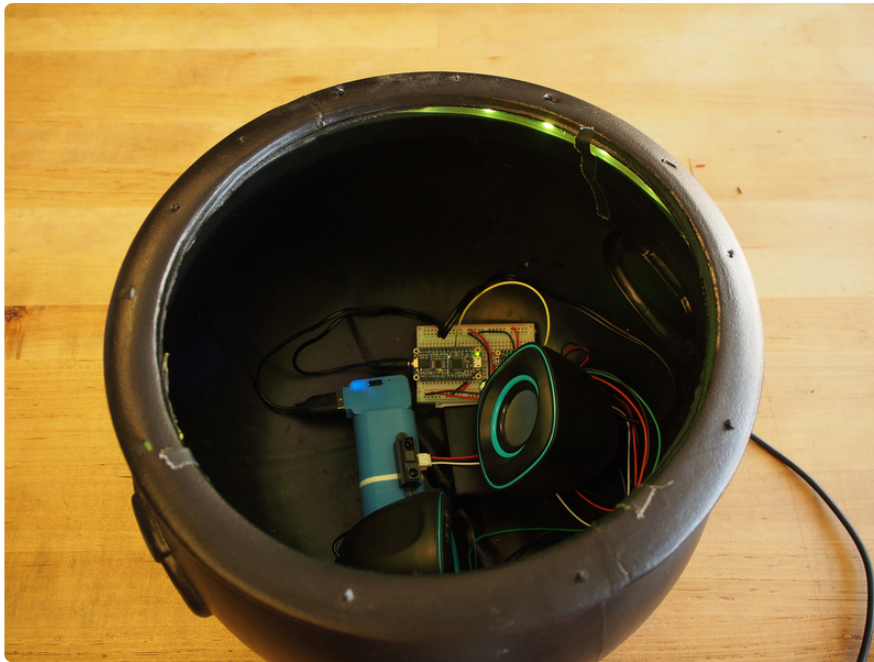
## Assemble and Use the Cauldron



Wrap the NeoPixel strip around the inside of the cauldron's rim, and secure it with more tape if needed.



You can place all of the parts inside the cauldron, and use gaffers tape or double stick foam tape to secure things. It's also OK to leave things loose in there if you like.



Cover the circuit and speakers with some black tissue paper or fabric

Secure the IR sensor to the side of the bowl's rim, pointing in toward the sensor. You can adjust until you feel the placement is best, then tape the sensor to the inside.







You can also diffuse the lighting a bit with some tissue paper if you like.



Time to fill the bowl with candy. You're ready to give out some scary (and delicious) surprises!!

