



Bluefruit Ouija Board

Created by John Park



<https://learn.adafruit.com/bluefruit-ouija-board>

Last updated on 2024-06-03 03:43:27 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Parts• Optional	
Board Teardown	6
<ul style="list-style-type: none">• Magnet Ring• Access the Innards• Base Electronics• Lid Parts• Linkage• Prep the Motor• Remove LEDs	
CircuitPython for Feather nRF52840	11
<ul style="list-style-type: none">• Set up CircuitPython Quick Start!	
Code the Ouija Board Controller	13
<ul style="list-style-type: none">• Text Editor• Download the Project Bundle• How It Works• Setup• Main Loop• Animation	
Build the Ouija Circuit	19
<ul style="list-style-type: none">• Feather Tripler• Motor FeatherWing• NeoPixel Strand• I2S Audio Amp• Add NeoPixels• Connections and Embedding Electronics	
Use the Ouija Board	24
<ul style="list-style-type: none">• Controller• Control Pad• D-Pad and Numbered Button Controls	

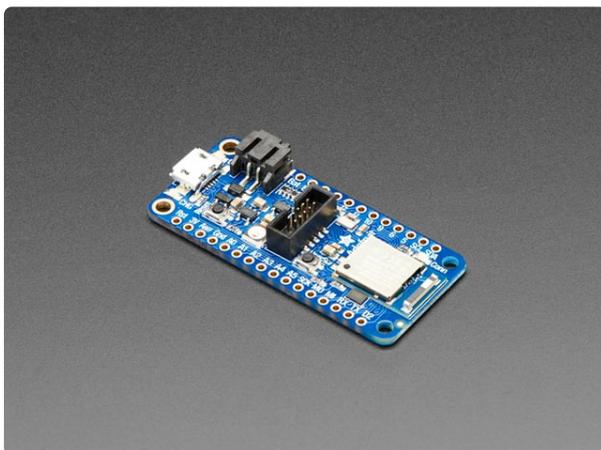
Overview



Upgrade a simple, off-the-shelf Ouija board prop into a wirelessly-controlled spirit board including custom audio clips and RGB NeoPixel lighting.

Using CircuitPython, a Feather nRF52840, and a Motor FeatherWing, plus the Bluefruit app, you can take control of the seance, haunt, or theatrical effects of your choosing!

Parts



[Adafruit Feather nRF52840 Express](https://www.adafruit.com/product/4062)

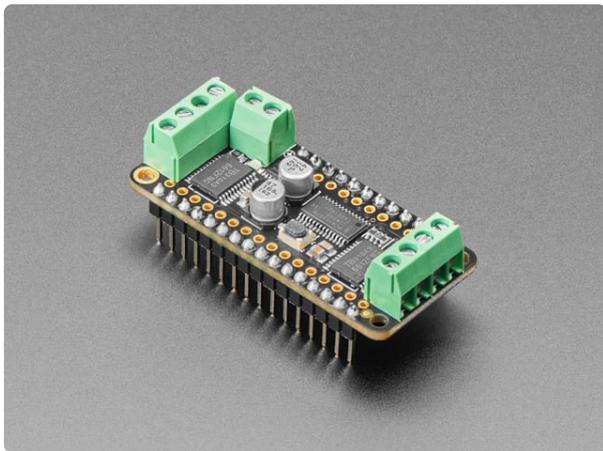
The Adafruit Feather nRF52840 Express is the new Feather family member with Bluetooth Low Energy and native USB support featuring the nRF52840! It's... <https://www.adafruit.com/product/4062>



Adafruit NeoPixel LED Dots Strand - 20 LEDs at 2" Pitch

Attaching NeoPixel strips to your costume can be a struggle as the flexible PCBs can crack when bent too much. So how to add little dots of color? Use these stranded NeoPixel dots!...

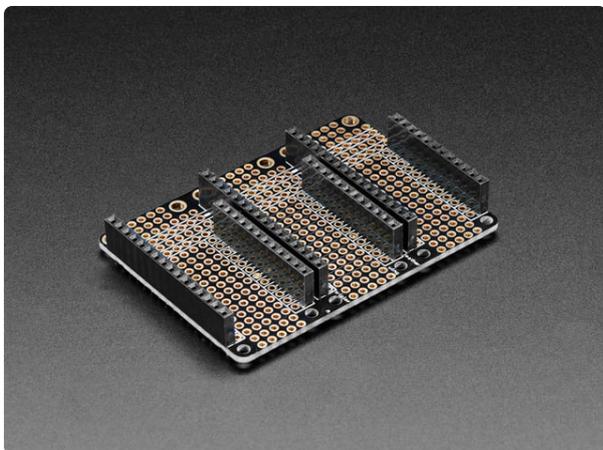
<https://www.adafruit.com/product/3630>



Assembled DC Motor + Stepper FeatherWing Add-on

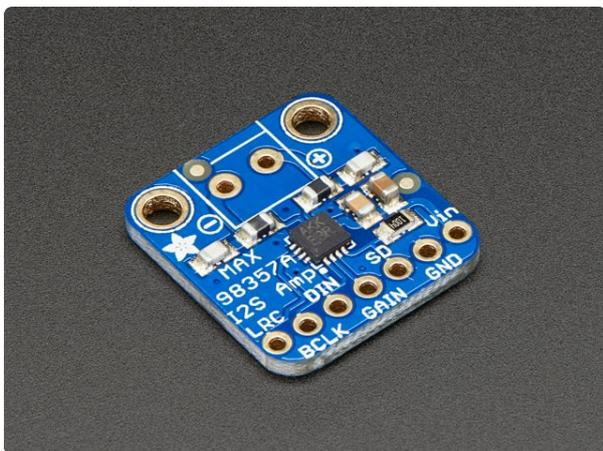
A Feather board without ambition is a Feather board without FeatherWings! This is the Fully assembled (with headers) DC Motor + Stepper FeatherWing which will let...

<https://www.adafruit.com/product/3243>



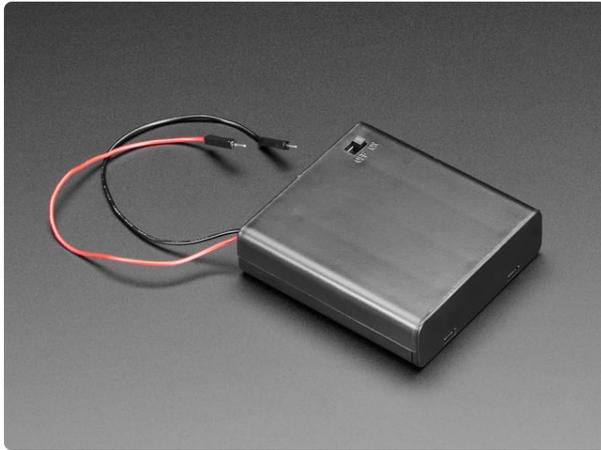
FeatherWing Tripler Mini Kit - Prototyping Add-on For Feathers

This is the FeatherWing Tripler - a prototyping add-on and more for all Feather boards. This is similar to our <https://www.adafruit.com/product/3417>



Adafruit I2S 3W Class D Amplifier Breakout - MAX98357A

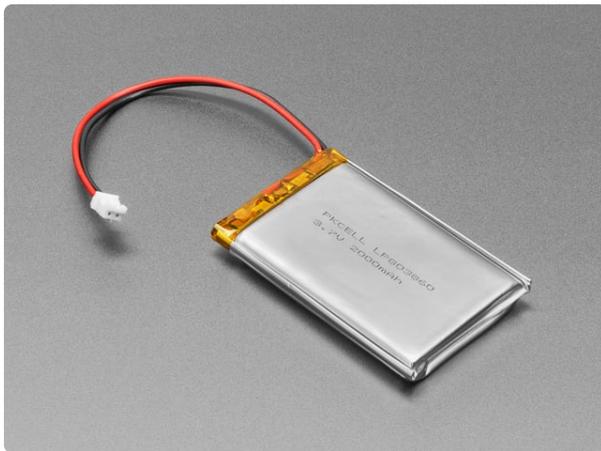
Listen to this good news - we now have an all in one digital audio amp breakout board that works incredibly well with the <https://www.adafruit.com/product/3006>



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

Make a nice portable power pack with this 4 x AA battery holder. It fits any alkaline or rechargeable AA batteries in series. There's a snap on cover and an on/off switch which can...

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



[Lithium Ion Battery - 3.7V 2000mAh](https://www.adafruit.com/product/2011)

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/2011>



[JST 2-pin Extension Cable with On/Off Switch - JST PH2](https://www.adafruit.com/product/3064)

By popular request - we now have a way you can turn on-and-off Lithium Polymer batteries without unplugging them. This PH2 Female/Male JST 2-pin Extension...

<https://www.adafruit.com/product/3064>

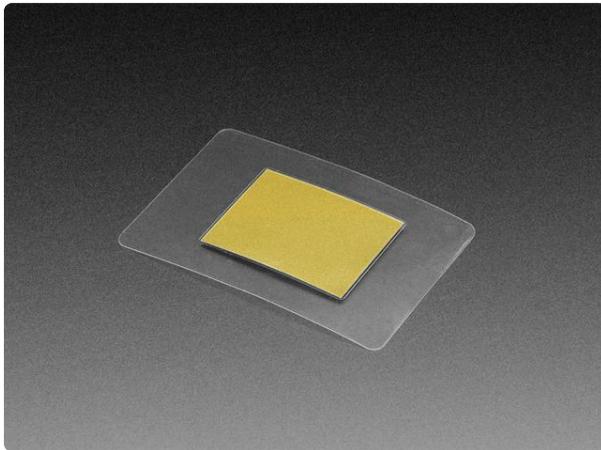


Haunted Ouija Board

You can find these (and similar props) at big box stores and Halloween stores, or search for them online. [Here's one example \(https://adafru.it/18bK\)](https://adafru.it/18bK).

The specific one used in this guide is Item #45079 by Sunstar Industries and can be found in this SKS Novelties [catalog \(https://adafru.it/18bU\)](https://adafru.it/18bU).

Optional



HD Magnet Viewing Film - 50x40mm

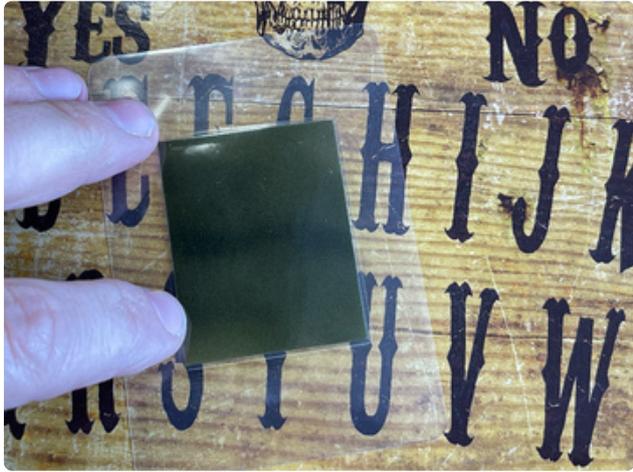
Add extra-sensory powers to your toolkit with this nifty piece of HD Magnet Viewing Film. It's a special film that contains a ferromagnetic slurry of ultra-small...

<https://www.adafruit.com/product/4047>

Board Teardown

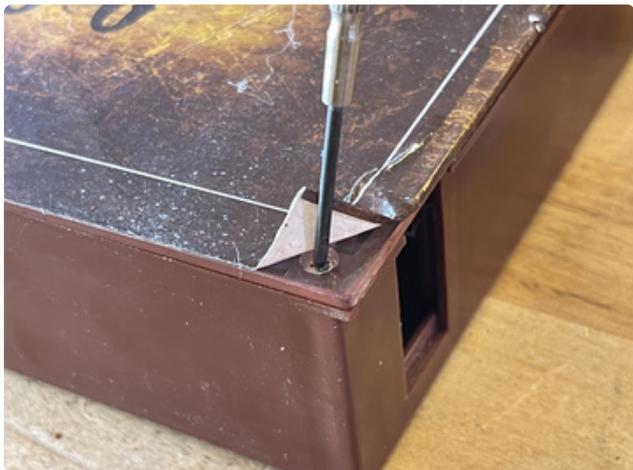
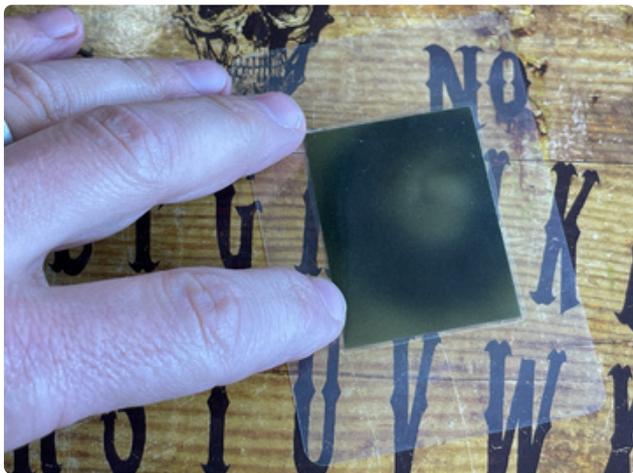


Here's a look at the board in it's off-the-shelf mode of operation:



Magnet Ring

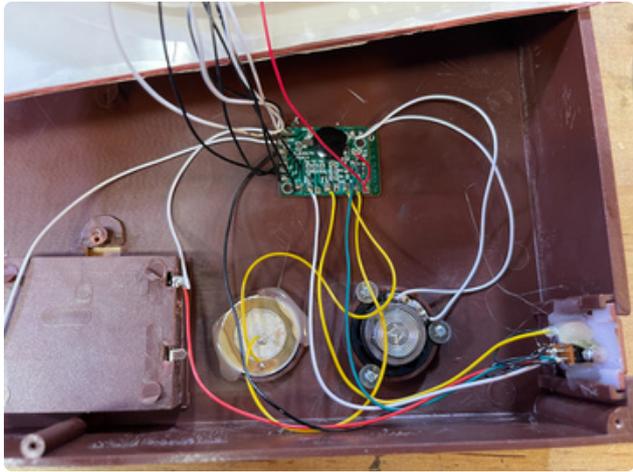
Using some [magnet viewing film \(http://adafru.it/4047\)](http://adafru.it/4047) you can see the magnet that moves under the lid.



Access the Innards

The screws that hold the lid on are hidden under the laminated graphic on the lid.

Unscrew these to open the board.

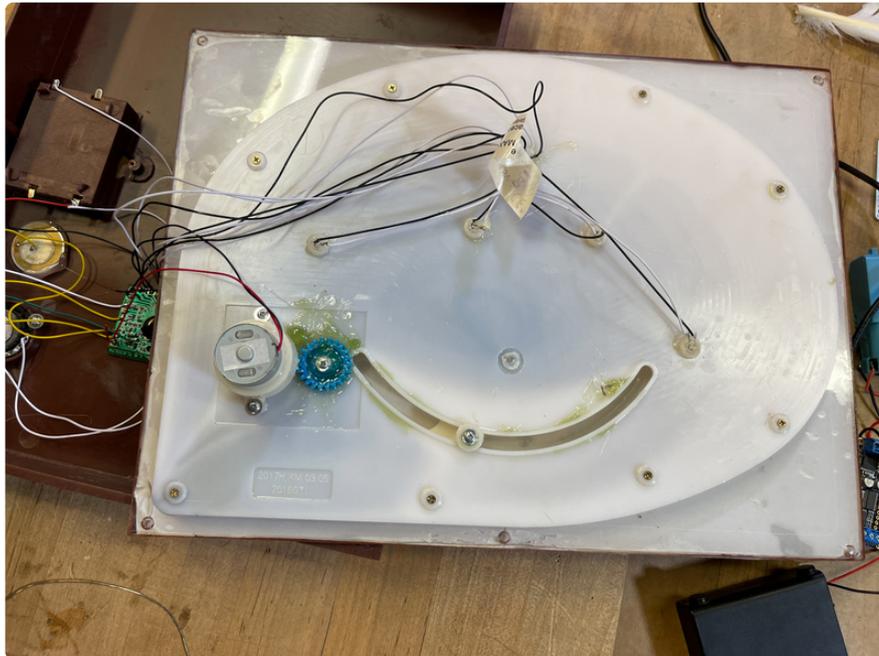


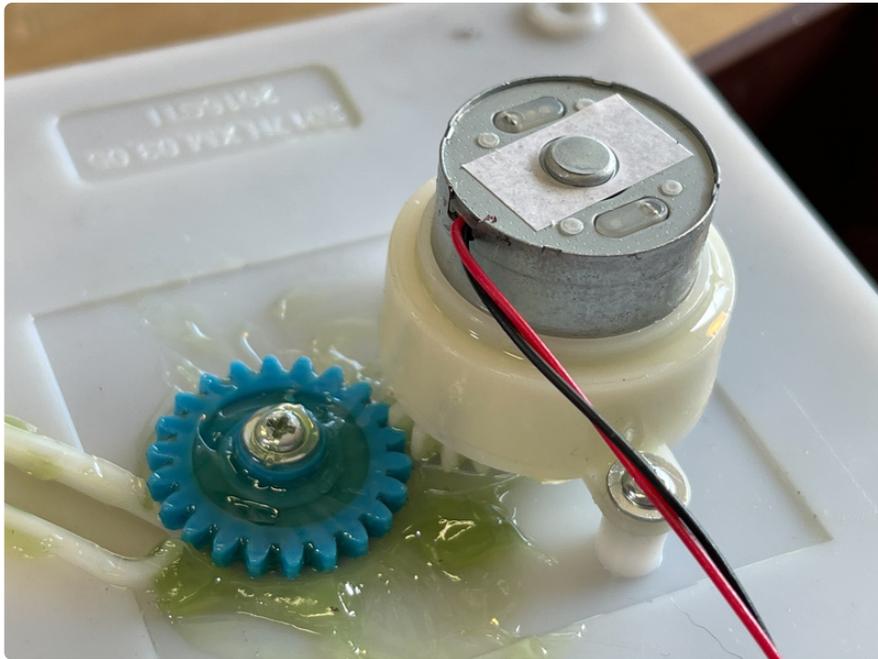
Base Electronics

In the base you'll see the controller board, a piezo-electric pickup (used for knock detection), speaker, molded battery box, and the side switch.

Lid Parts

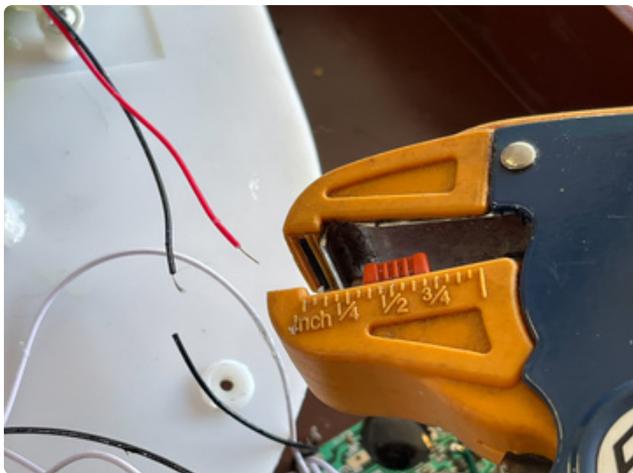
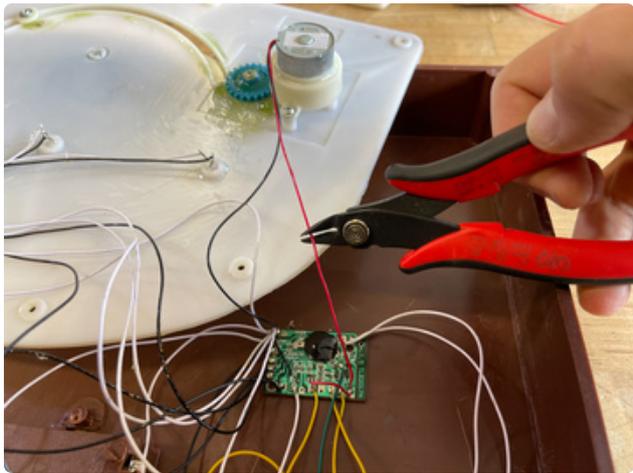
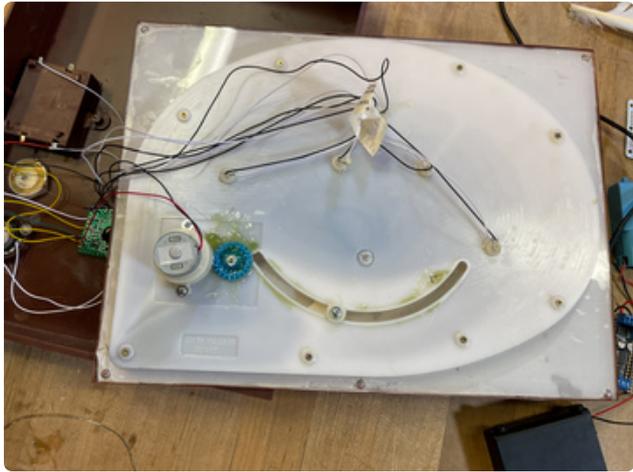
The lid contains a DC motor, linkage arm for moving the magnet ring, and four LEDs.





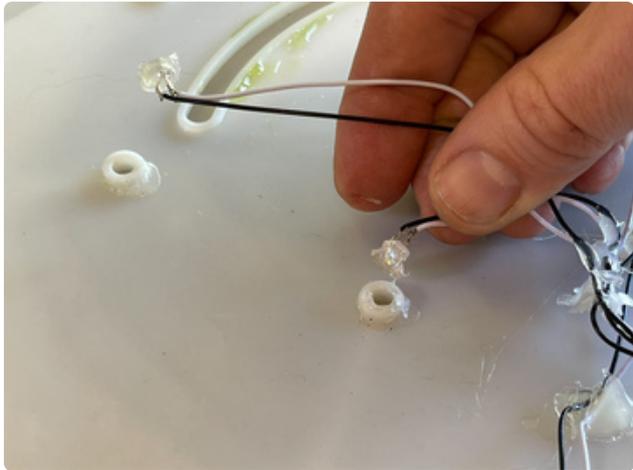
Linkage

Here you can see the very clever linkage that converts the rotation of the motor into the figure eight pattern of the circular magnet.



Prep the Motor

Carefully flip the lid over. The motor is connected to the driver board with two wires. You can snip these with some edge cutters and then trim away a bit of insulation so they can be connected to the motor FeatherWing later.



Remove LEDs

Pull out the old LEDs - you'll be replacing them with NeoPixels! You can use a bit of isopropyl alcohol to assist in releasing the hot glue bond.

Now you're ready to begin adding in the new electronics.

CircuitPython for Feather nRF52840

[CircuitPython \(https://adafru.it/tB7\)](https://adafru.it/tB7) is a derivative of [MicroPython \(https://adafru.it/BeZ\)](https://adafru.it/BeZ) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** drive to iterate.

The following instructions will show you how to install CircuitPython. If you've already installed CircuitPython but are looking to update it or reinstall it, the same steps work for that as well!

Set up CircuitPython Quick Start!

Follow this quick step-by-step for super-fast Python power :)

Download the latest version of
CircuitPython for this board via
CircuitPython.org

<https://adafru.it/FxJ>



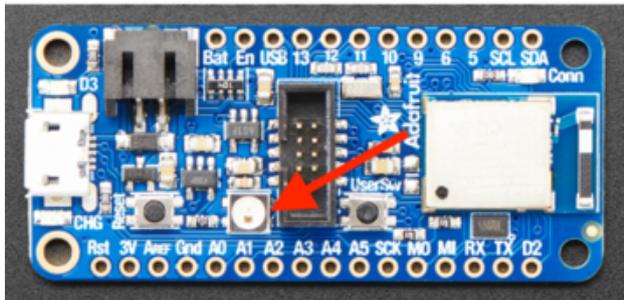
adafruit-circuitpython-
feather_nrf52840_exp...en_US-4.0.0-beta.1.uf2

Click the link above to download the latest UF2 file.

Download and save it to your desktop (or wherever is handy).

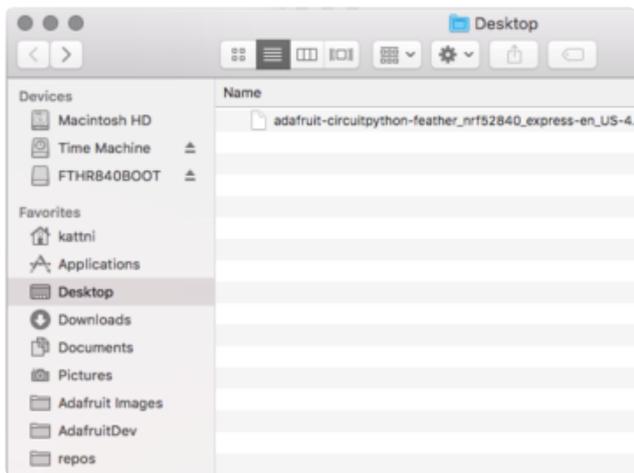
Plug your Feather nRF52840 into your computer using a known-good USB cable.

A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.

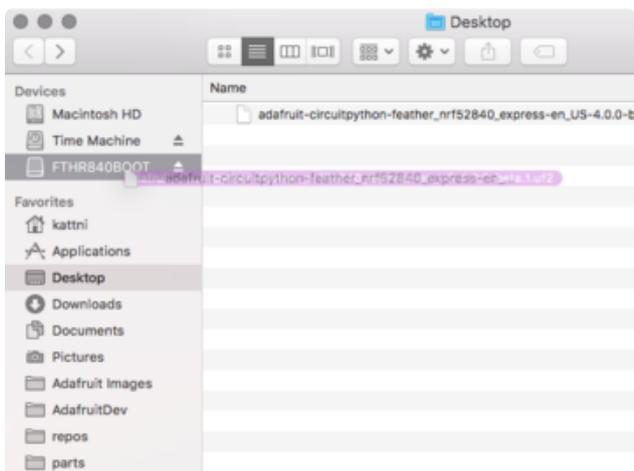


Double-click the **Reset** button next to the USB connector on your board, and you will see the NeoPixel RGB LED turn green (identified by the arrow in the image). If it turns red, check the USB cable, try another USB port, etc. **Note:** The little red LED next to the USB connector will pulse red. That's ok!

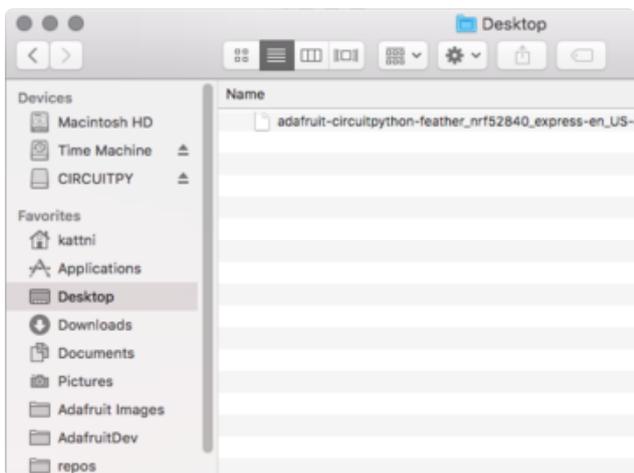
If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!



You will see a new disk drive appear called **FTHR840BOOT**.



Drag the `adafruit_circuitpython_etc.uf2` file to **FTHR840BOOT**.



The LED will flash. Then, the **FTHR840BOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

That's it, you're done! :)

Code the Ouija Board Controller

You may need to update the bootloader on your nRF52840 board to install the latest CircuitPython! Use the link below for more info:

<https://learn.adafruit.com/introducing-the-adafruit-nrf52840-feather/update-bootloader> (<https://adafru.it/18fQ>)

Text Editor

Adafruit recommends using the **Mu** editor for editing your CircuitPython code. You can get more info in [this guide \(https://adafru.it/ANO\)](https://adafru.it/ANO).

Alternatively, you can use any text editor that saves simple text files.

Download the Project Bundle

Your project will use a specific set of CircuitPython libraries, **.wav** files, and the **code.py** file. To get everything you need, click on the **Download Project Bundle** link below, and uncompress the **.zip** file.

Drag the contents of the uncompressed bundle directory onto your board's **CIRCUITPY** drive, replacing any existing files or directories with the same names, and adding any new ones that are necessary.

```
# SPDX-FileCopyrightText: 2022 John Park for Adafruit Industries
# SPDX-License-Identifier: MIT

# BLE Ouija Board

import time
import random
import board
import digitalio
import neopixel
from adafruit_motorkit import MotorKit
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService
from adafruit_bluefruit_connect.packet import Packet
from adafruit_bluefruit_connect.button_packet import ButtonPacket
import audiocore
import audiomixer
import audiobusio

# Prep the status LEDs on the Feather
blue_led = digitalio.DigitalInOut(board.BLUE_LED)
red_led = digitalio.DigitalInOut(board.RED_LED)
blue_led.direction = digitalio.Direction.OUTPUT
red_led.direction = digitalio.Direction.OUTPUT

ble = BLERadio()
uart_service = UARTService()
advertisement = ProvideServicesAdvertisement(uart_service)

num_motors = 1 # up to 4 motors depending on the prop you are driving

i2c = board.I2C() # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C() # For using the built-in STEMMA QT connector on a
microcontroller
motorwing = MotorKit(i2c=i2c)
motorwing.frequency = 122 # tune this 50 - 200 range
max_throttle = 0.65 # tune this 0.2 - 1 range
```

```

# # make arrays for all the things we care about
motors = [None] * num_motors

motors[0] = motorwing.motor1

# # set motors to "off"
for i in range(num_motors):
    motors[i].throttle = None

# - Audio setup
audio = audiobusio.I2SOut(bit_clock=board.TX, word_select=board.MISO, data=board.RX)
mixer = audiomixer.Mixer(voice_count=2, sample_rate=11025, channel_count=1,
                        bits_per_sample=16, samples_signed=True)
audio.play(mixer) # attach mixer to audio playback
wav_files = (('spooky_ouija.wav', 0.07, True), ('lars_ouija.wav', 0.09, False))
# open samples
for i in range(len(wav_files)):
    wave = audiocore.WaveFile(open(wav_files[i][0], "rb"))
    mixer.voice[i].level = 0 # start up with level down
    mixer.voice[i].play(wave, loop=wav_files[i][2])

# - NeoPixels
fire_color = 0xcc6600
fade_by = -1
num_leds = 7
max_bright = 0.5
led_pin = board.D6
leds = neopixel.NeoPixel(led_pin, num_leds, brightness=0.0, auto_write=False)
leds.fill(fire_color)
leds.show()

last_time = 0
next_duration = 0.2

print("BLE Ouija board")
print("Use Adafruit Bluefruit app to connect")

while True:
    blue_led.value = False
    ble.name = 'Ouija'
    ble.start_advertising(advertisement)

    while not ble.connected:
        # Wait for a connection.
        pass
    blue_led.value = True # turn on blue LED when connected
    while ble.connected:
        if uart_service.in_waiting:
            # Packet is arriving.
            red_led.value = False # turn off red LED
            packet = Packet.from_stream(uart_service)
            if isinstance(packet, ButtonPacket) and packet.pressed:
                red_led.value = True # blink to show a packet has been received
                if packet.button == ButtonPacket.RIGHT: # > button pressed
                    print("forward")
                    motors[0].throttle = max_throttle
                    time.sleep(0.1) # wait a moment

                elif packet.button == ButtonPacket.LEFT: # < button
                    print("reverse")
                    motors[0].throttle = max_throttle * -1
                    time.sleep(0.1)

                elif packet.button == ButtonPacket.DOWN: # v button
                    print("stop")
                    motors[0].throttle = None
                    time.sleep(0.1)

```

```

elif packet.button == ButtonPacket.BUTTON_1: # 1 button
    print("on BG music")
    mixer.voice[0].level = wav_files[0][1]
    time.sleep(0.1)

elif packet.button == ButtonPacket.BUTTON_3: # 3 button
    print("off BG music")
    mixer.voice[0].level = 0.0
    time.sleep(0.1)

elif packet.button == ButtonPacket.BUTTON_2: # 2 button
    print("on led & Lars")
    leds.brightness = max_bright
    mixer.voice[1].play(wave, loop=False)
    mixer.voice[1].level = wav_files[1][1]
    time.sleep(0.1)

elif packet.button == ButtonPacket.BUTTON_4: # 4 button
    print("off led & Lars")
    leds.brightness = 0
    mixer.voice[1].level = 0.0
    time.sleep(0.1)

# fade down all LEDs
leds[:] = [[min(max(i+fade_by, 0), 255) for i in l] for l in leds]
leds.show()

# add new fire to leds
if time.monotonic() - last_time > next_duration:
    last_time = time.monotonic()
    next_duration = random.uniform(0.95, 1.95) # tune these nums
    # for i in range( 1):
    c = fire_color # get our color
    c = (c >> 16 & 0xff, c >> 8 & 0xff, c & 0xff) # make it a tuple
    leds[random.randint(0, num_leds-1)] = c

```

How It Works

The code imports a number of libraries needed to use the board pins, NeoPixels, BLE, Motor FeatherWing, and audio.

```

import time
import random
import board
import digitalio
import neopixel
from adafruit_motorkit import MotorKit
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService
from adafruit_bluefruit_connect.packet import Packet
from adafruit_bluefruit_connect.button_packet import ButtonPacket
import audiocore
import audiomixer
import audiobusio

```

Setup

Next, the code does the setup for LED indicators, BLE, motor driving, audio, and NeoPixel animation.

```
blue_led = digitalio.DigitalInOut(board.BLUE_LED)
red_led = digitalio.DigitalInOut(board.RED_LED)
blue_led.direction = digitalio.Direction.OUTPUT
red_led.direction = digitalio.Direction.OUTPUT

ble = BLERadio()
uart_service = UARTService()
advertisement = ProvideServicesAdvertisement(uart_service)

num_motors = 1 # up to 4 motors depending on the prop you are driving

motorwing = MotorKit(i2c=board.I2C())
motorwing.frequency = 122 # tune this 50 - 200 range
max_throttle = 0.65 # tune this 0.2 - 1 range

# # make arrays for all the things we care about
motors = [None] * num_motors

motors[0] = motorwing.motor1

# # set motors to "off"
for i in range(num_motors):
    motors[i].throttle = None

# - Audio setup
audio = audiobusio.I2SOut(bit_clock=board.TX, word_select=board.MISO, data=board.RX)
mixer = audiomixer.Mixer(voice_count=2, sample_rate=11025, channel_count=1,
                        bits_per_sample=16, samples_signed=True)
audio.play(mixer) # attach mixer to audio playback
wav_files = (('spooky_ouija.wav', 0.07, True), ('lars_ouija.wav', 0.09, False))
# open samples
for i in range(len(wav_files)):
    wave = audiocore.WaveFile(open(wav_files[i][0], "rb"))
    mixer.voice[i].level = 0 # start up with level down
    mixer.voice[i].play(wave, loop=wav_files[i][2])

# - NeoPixels
fire_color = 0xcc6600
fade_by = -1
num_leds = 7
max_bright = 0.5
led_pin = board.D6
leds = neopixel.NeoPixel(led_pin, num_leds, brightness=0.0, auto_write=False)
leds.fill(fire_color)
leds.show()
```

Main Loop

The main loop of the program establishes the BLE advertisement and waits for a connection with the Bluefruit program.

Once connected, the UART service waits to receive packets of button presses. These are translated into motor movement in both directions, motor stop, and enabling or disabling the audio tracks and NeoPixel animations.

```
while True:
    blue_led.value = False
    ble.name = 'Ouija'
    ble.start_advertising(advertisement)

    while not ble.connected:
        # Wait for a connection.
        pass
    blue_led.value = True # turn on blue LED when connected
    while ble.connected:
        if uart_service.in_waiting:
            # Packet is arriving.
            red_led.value = False # turn off red LED
            packet = Packet.from_stream(uart_service)
            if isinstance(packet, ButtonPacket) and packet.pressed:
                red_led.value = True # blink to show a packet has been received
                if packet.button == ButtonPacket.RIGHT: # > button pressed
                    print("forward")
                    motors[0].throttle = max_throttle
                    time.sleep(0.1) # wait a moment

                elif packet.button == ButtonPacket.LEFT: # < button
                    print("reverse")
                    motors[0].throttle = max_throttle * -1
                    time.sleep(0.1)

                elif packet.button == ButtonPacket.DOWN: # v button
                    print("stop")
                    motors[0].throttle = None
                    time.sleep(0.1)

                elif packet.button == ButtonPacket.BUTTON_1: # 1 button
                    print("on BG music")
                    mixer.voice[0].level = wav_files[0][1]
                    time.sleep(0.1)

                elif packet.button == ButtonPacket.BUTTON_3: # 3 button
                    print("off BG music")
                    mixer.voice[0].level = 0.0
                    time.sleep(0.1)

                elif packet.button == ButtonPacket.BUTTON_2: # 2 button
                    print("on led & Lars")
                    leds.brightness = max_bright
                    mixer.voice[1].play(wave, loop=False)
                    mixer.voice[1].level = wav_files[1][1]
                    time.sleep(0.1)

                elif packet.button == ButtonPacket.BUTTON_4: # 4 button
                    print("off led & Lars")
                    leds.brightness = 0
                    mixer.voice[1].level = 0.0
                    time.sleep(0.1)
```

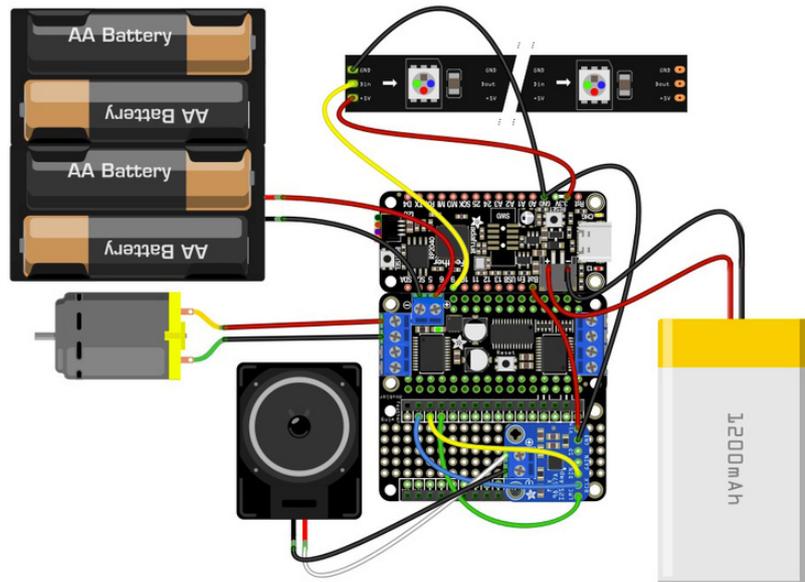
Animation

The fade animation on the NeoPixels is run at all times, with the BLE packets only controlling the brightness.

```
# fade down all LEDs
leds[:] = [[min(max(i+fade_by, 0), 255) for i in l] for l in leds]
leds.show()

# add new fire to leds
if time.monotonic() - last_time > next_duration:
    last_time = time.monotonic()
    next_duration = random.uniform(0.95, 1.95) # tune these nums
    # for i in range( 1):
    c = fire_color # get our color
    c = (c >> 16 & 0xff, c >> 8 & 0xff, c & 0xff) # make it a tuple
    leds[random.randint(0, num_leds-1)] = c
```

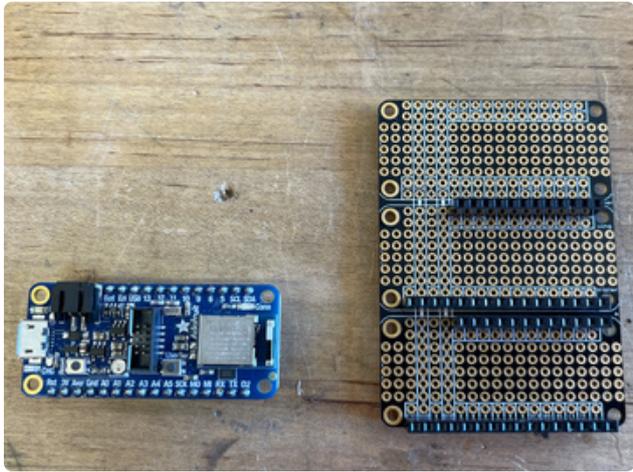
Build the Ouija Circuit



You can use the Feather Tripler to connect the Feather, Motor FeatherWing, and I2S amplifier together. This makes it nice and modular and simplifies the connections to the motor, speaker, batteries, and NeoPixels.

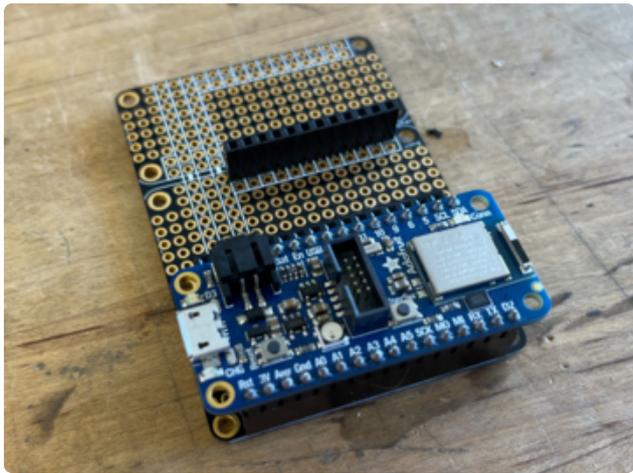
Power

You'll use a 4x AA battery pack for motor power and a separate LiPo battery for the Feather.

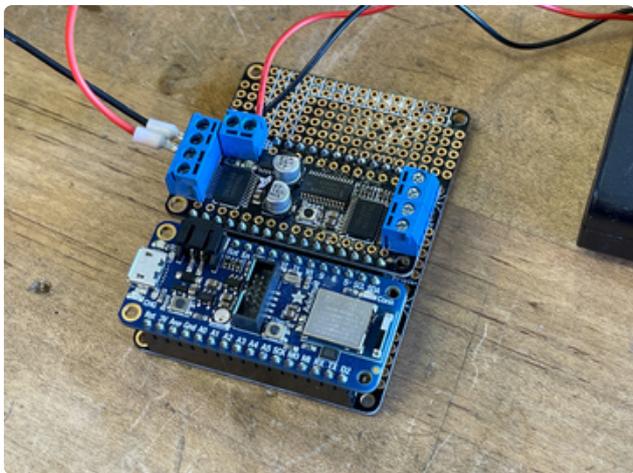
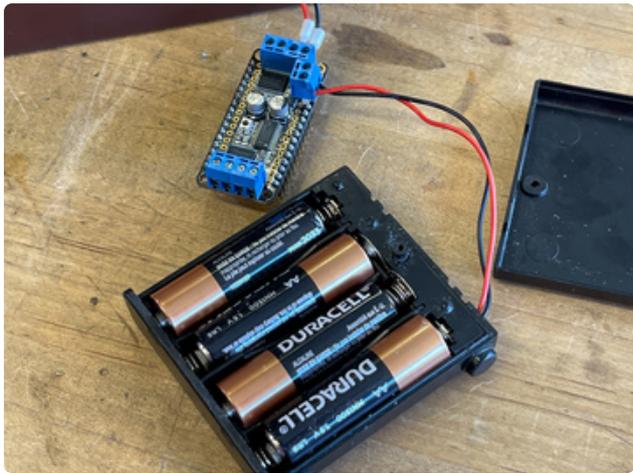
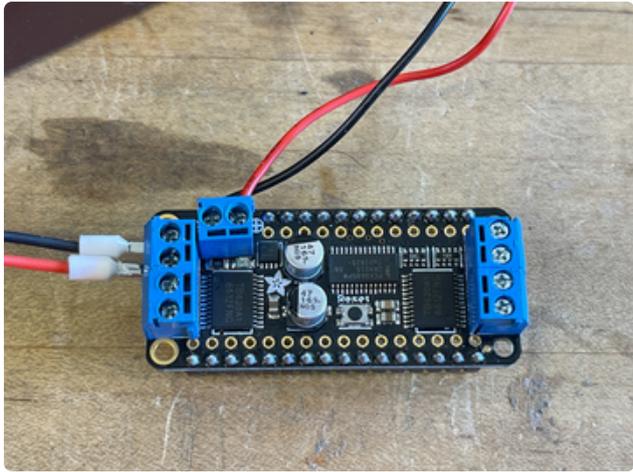


Feather Tripler

Solder two sets of socket headers on the Tripler as shown.



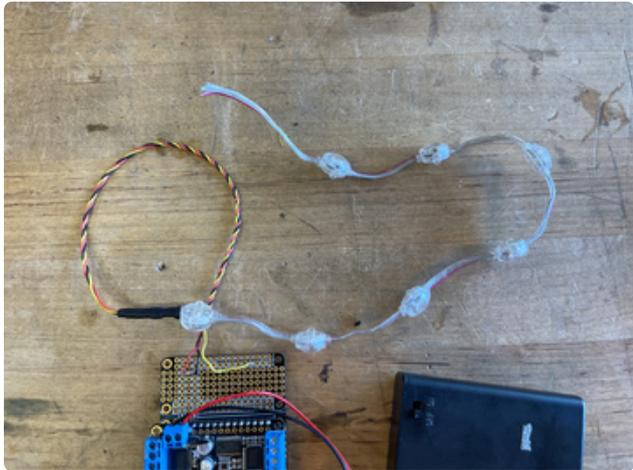
Solder header pins underneath the Feather and connect it to the Tripler.



Motor FeatherWing

Assemble the Motor FeatherWing, connect it to the AA battery pack, and then connect it to the FeatherWing Tripler.

Connect the motor wires to the M1 motor output. Polarity doesn't matter, you'll be able to reverse direction in code either way, since it's a dual H-bridge driver.



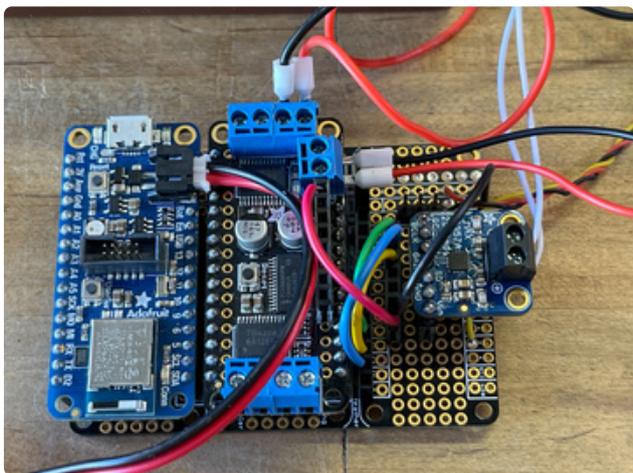
NeoPixel Strand

Connect the NeoPixel strand to the Feather (via the Tripler) with these connections:

Feather 3V to NeoPixel V+

Feather GND to NeoPixel GND

Feather D6 to NeoPixel Din



I2S Audio Amp

Using the prototyping area of the Tripler, connect the I2S audio amplifier to the following pins:

amp LRC to Feather MI

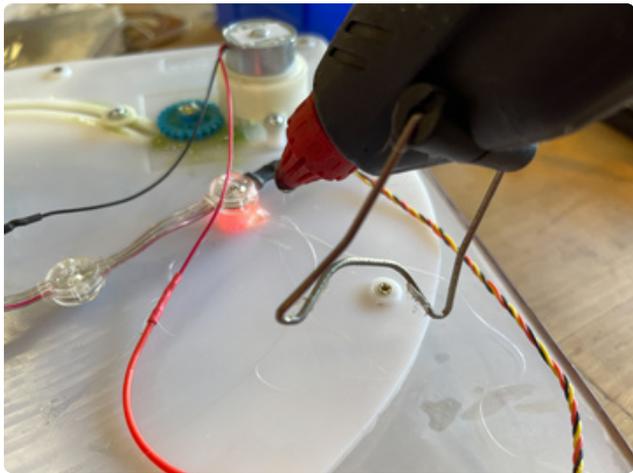
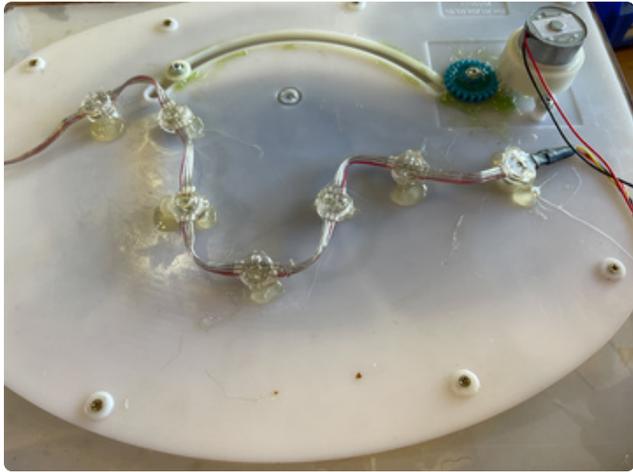
amp BCLK to Feather TX

amp DIN to Feather RX

amp GND to Feather GND

amp Vin to Feather BAT

Then connect the speaker wires to the amp output (polarity doesn't matter here).



Add NeoPixels

You can use a hot glue gun or other adhesive to add the NeoPixel strand.

Please be careful with hot glue guns - avoid burns from the device and hot glue. Help younger makers if needed.

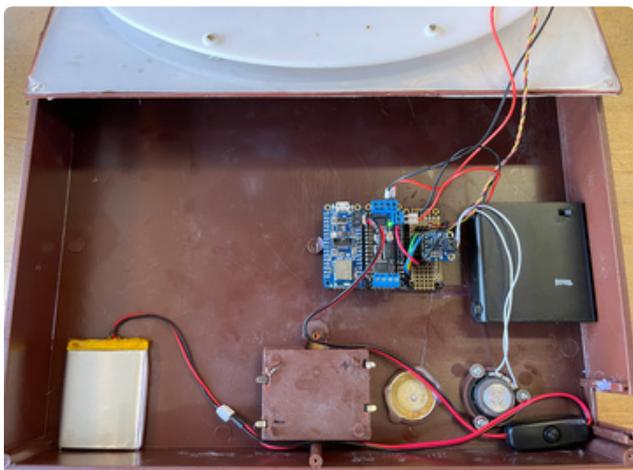
Connections and Embedding Electronics

Plug the LiPo battery to the Feather via the JST extender switch, then place the electronics inside the prop as shown.

Test the fit with the lid in place, then secure everything with hot glue or double stick adhesive foam tape.

You can turn on the Feather power when ready for use.

Close the lid and you're ready to hold a high-tech seance!





Use the Ouija Board



Launch Bluefruit Connect on your mobile device, making sure Bluetooth is enabled.



With the Ouija board Feather turned on, you'll see a device named 'Ouija' show up in the app. Click that device name.



Controller

Click on the **Controller** item in the **Modules** list.





Control Pad

Choose **Control Pad** from the **Module** list.



D-Pad and Numbered Button Controls

Now, you can use the buttons to control the Ouija board motor, sound, and lights!

Here's what they do:

Right arrow = motor forward

Left arrow = motor backward

Down arrow = motor stop

1 = play background music (loop)

3 = stop background music

2 = play voiceover sound file (one shot) and LED animation

4 = stop LED animation

