Talking D20 with RP2040 PropMaker Feather

Created by Phillip Burgess

https://learn.adafruit.com/talking-d20-rp2040

Last updated on 2023-09-05 07:14:32 PM EDT
# Table of Contents

## Overview
- RP2040 PropMaker
- Parts
- Optional Parts

## Circuit Diagram
- Adafruit Library for Fritzing
- Wired Connections
- Optional: Battery Monitor

## CAD Files
- 3D Printed Parts
- Build Volume
- 3D Printing Service
- CAD Assembly
- Dual Extruder/Multi-Material
- Design Source Files

## CircuitPython
- CircuitPython Quickstart
- Safe Mode
- Flash Resetting UF2

## Code the D20
- Upload the Code and Libraries to the RP2040 Prop-Maker Feather
- OPTIONAL: Battery Monitoring
- How the CircuitPython Code Works

## Wiring
- Slide Switch Wire
- Wiring Slide Switch
- Connect Slide Switch
- Connected Slide Switch
- Soldered Slide Switch
- Speaker Wires
- Battery Monitoring (optional)

## Assembly
- Magnets
- Glueing Magnets
- Glueing Magnets Continued
- Tap Standoffs
- Install Speaker
- Adding Counter Weights
- Install Cap
- Speaker Wires
- Install Speaker Wires
- Connect Battery
- Secure Battery
- Install Switch
- Installing Feather
- Secure Feather
• Final Check
• Closing Halves
• Final Build
Overview

Build a D20 gaming die that tells you what number you rolled! Made with 3D printed parts, CircuitPython and an Adafruit RP2040 PropMaker Feather.

RP2040 PropMaker
This is a remake of a classic DIY project () from 2015. This new build simplifies the assembly: instead of several breakout boards, everything is now all-in-one!

As this will come up: "But is it balanced?"©®

No. It’s full of electronics!

There’s a space to add a few pennies or some plumber’s putty as a counterbalance...you can work at it a bit if you like...but this will never be a perfect entropy source. It’s intended for casual gaming, as an “executive decision maker,” or simply a conversation piece to get folks interested in code and electronics.
Parts

Adafruit RP2040 Prop-Maker Feather with I2S Audio Amplifier
The Adafruit Feather series gives you lots of options for a small, portable, rechargeable microcontroller board. By picking a feather and stacking on a FeatherWing you can create...
https://www.adafruit.com/product/5768

Mini Oval Speaker - 8 Ohm 1 Watt
Hear the good news! This wee speaker is a great addition to any audio project where you need 8 ohm impedance and 1W or less of power. We particularly like...
https://www.adafruit.com/product/3923

Lithium Ion Polymer Battery Ideal For Feathers - 3.7V 400mAh
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/3898
**Breadboard-friendly SPDT Slide Switch**
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...

**USB Type A to Type C Cable - approx 1 meter / 3 ft long**
As technology changes and adapts, so does Adafruit. This USB Type A to Type C cable will help you with the transition to USB C, even if you're still...
[https://www.adafruit.com/product/4474](https://www.adafruit.com/product/4474)

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Quantity</th>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neodymium Magnets</td>
<td>8</td>
<td>D42 1/4&quot; dia. x 1/8&quot; thick</td>
<td><a href="https://www.kjmagnetics.com/proddetail.asp?prod=D42">https://www.kjmagnetics.com/proddetail.asp?prod=D42</a></td>
</tr>
<tr>
<td>M2.5 Steel Screws</td>
<td>2</td>
<td>M2.5 x 6mm long screws</td>
<td><a href="https://www.mcmaster.com/92095A458/">https://www.mcmaster.com/92095A458/</a></td>
</tr>
<tr>
<td>Starbond Medium Super Glue</td>
<td>1</td>
<td>2 oz. Medium CA Glue (Premium Cyanoacrylate Super Glue)</td>
<td><a href="https://www.amazon.com/Starbond-EM-150-Medium-Premium-Cyanoacrylate/dp/B00C32MHJU/">https://www.amazon.com/Starbond-EM-150-Medium-Premium-Cyanoacrylate/dp/B00C32MHJU/</a></td>
</tr>
<tr>
<td>M2.5 Screw Tap</td>
<td>1</td>
<td>Metric Machine Taps M2.5 x 0.45mm</td>
<td><a href="https://www.amazon.com/Saipe-Thread-M1-M3-5-Tapping-Adjustable/dp/B0B38WRV9P/">https://www.amazon.com/Saipe-Thread-M1-M3-5-Tapping-Adjustable/dp/B0B38WRV9P/</a></td>
</tr>
</tbody>
</table>
Optional Parts

Through-Hole Resistors - 100K ohm 5% 1/4W - Pack of 25
ΩMG! You're not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you! This is a 25 Pack of 100K...
https://www.adafruit.com/product/2787

Circuit Diagram

The diagram below provides a general visual reference for wiring of the components once you get to the Assembly page. This diagram was created using the software package Fritzing.

Adafruit Library for Fritzing

Adafruit uses the Adafruit's Fritzing parts library to create circuit diagrams for projects. You can download the library or just grab individual parts. Get the library and parts from GitHub - Adafruit Fritzing Parts.
Wired Connections

- The speaker wires are connected to pins on the screw block terminal. Red to SP K+, Black to SPK-
- The slide switch is connected to the EN and GND pins on the Feather.
- A 400mAh battery is connected to the battery port on the Feather.

Optional: Battery Monitor

With the addition of two 100KΩ resistors (and a small edit to the code, explained later), the D20 can sense and announce when the battery needs charging.

Through-Hole Resistors - 100K ohm 5% 1/4W - Pack of 25
ΩMG! You’re not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you!This is a 25 Pack of 100K...
https://www.adafruit.com/product/2787

The two resistors are connected in series. One end goes to the board’s BAT pin, opposite end to a GND pin, and the center point to one of the analog input pins such as A3.
The Feather’s main GND pin might already be occupied by a wire for the power switch. If you can’t fit both the wire and resistor through that hole, other ground points will suffice, such as the one on the Servo header.

CAD Files

3D Printed Parts
STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material using PLA filament. Original design source may be downloaded using the links below.

Download STLs.zip
Download CAD Source
Build Volume

The parts require a 3D printer with a minimum build volume.

- 68mm (X) x 60mm (Y) x 36mm (Z)

3D Printing Service

These parts were made using JLCPCB's 3D printing service.

Use the following options for matte black resin parts.

3D Technology: SLA(resin)
Material: Black Resin
Color: Grayish Black
Sanding: Yes

This clear die was made using PCBWAY's 3D printing service. UTR-8100 transparent resin with spray varnish finish option. This is a costlier route, but delivers an injection-molding-like appearance that would be difficult to achieve any other way.

The numbers were hand-painted to make them “pop.” It’s lovely in a 1990s clear electronics craze way.
CAD Assembly
The two halves are joined together with neodymium magnets.

The mini oval speaker is secured to the built-in holder on the Half-A part using the included sticky adhesive.

The 400mAh battery is secured to the mini oval speaker using the other side of the sticky adhesive.

The Feather is secured to half-A using two M2.5x8mm long screws.

The slide switch is press fitted into the built-in holder on half-A.

The cap and counterweights (2x US penny coins) are press fitted into the built-in holder on half-B.

Dual Extruder/Multi-Material
These STLs have been created for FDM 3D printers with multi-material capabilities.

The numbered labels have been separated from the D20 halves.

Each type of machine features their own software for slicing.

Download Multi-Material STLs
Design Source Files

The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more.

Electronic components like Adafruit's boards, displays, connectors and more can be downloaded from the Adafruit CAD parts GitHub Repo.

CircuitPython

CircuitPython is a derivative of MicroPython 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.

CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

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

Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.
To enter the bootloader, hold down the BOOT/BOOTSEL button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the reset button (highlighted in blue above). Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!

If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

A lot of people end up using charge-only USB cables and it is very frustrating! Make sure you have a USB cable you know is good for data sync.
You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

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

Safe Mode

You want to edit your code.py or modify the files on your CIRCUITPY drive, but find that you can't. Perhaps your board has gotten into a state where CIRCUITPY is read-only. You may have turned off the CIRCUITPY drive altogether. Whatever the reason, safe mode can help.
Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in boot.py (where you can set CIRCUITPY read-only or turn it off completely). Second, it does not run the code in code.py. And finally, it does not automatically soft-reload when data is written to the CIRCUITPY drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the CIRCUITPY drive.

**Entering Safe Mode**

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

**In Safe Mode**

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.
Running in safe mode! Not running saved code.
CircuitPython is in safe mode because you pressed the reset button during boot.
Press again to exit safe mode.
Press any key to enter the REPL. Use CTRL-D to reload.
```

You can now edit the contents of the CIRCUITPY drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

**Flash Resetting UF2**

If your board ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try loading this 'nuke' UF2 which will do a 'deep
clean’ on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

Download flash erasing "nuke" UF2

---

Code the D20

Once you've finished setting up your RP2040 Prop-Maker Feather with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the Download Project Bundle button in the window below. It will download to your computer as a zipped folder.

```
# SPDX-FileCopyrightText: 2023 Phillip Burgess for Adafruit Industries
#
# SPDX-License-Identifier: MIT

###
Talking D20 for Adafruit RP2040 Prop-Maker Feather.
Required additions:
- 8 Ohm 1 watt speaker (Adafruit #4227)
- 400 mAh LiPoly battery (Adafruit #3898)
- 3D printed enclosure
Optional additions:
- Battery monitoring can be added with two 10K resistors in series.
  One end to BAT, one to GND, and center point to an analog pin (e.g. A3).
  Then set BATT_SENSE in configurables section, e.g. BATT_SENSE = board.A3
###

# pylint: disable=import-error
from random import randint
import time
import adafruit_lis3dh
import analogio
import audiocore
import audiobusio
import board
from digitalio import DigitalInOut, Direction

# CONFIGURABLES

WAV_PATH = "WAVs"
WAV_FILES = (
    "01", # Index 0 (WAV for face 1)
    "02", # Index 1 (WAV for face 2)
    "03", # etc...
    "04",
    "05",
    "06",
    "07",
    "08",
    "09",
    "10",
    "11",
    "12",
    "13",
    "14",
)
BATT_SENSE = None  # Assign analog pin if voltage divider present
BATT_LOW = 3.4  # Voltage for battery warning (if BATT_SENSE)
FREEFALL_THRESHOLD = 8.65  # Near-freefall = 0.3G ^ 2
FREEFALL_MIN_DURATION = 1 / 25  # Time (seconds) roll is in near-freefall
SETTLE_TIME = 0.5  # Time (seconds) to settle on a face
SETTLE_TIMEOUT = 3.0  # If unsettled by this, resume freefall check

FACE_VECTORS = (  # Accelerometer vectors, shouldn't need to edit
    (-3.50, 0.00, 9.16),  # Face 1 (index 0)
    (5.66, -5.66, -5.66),  # Face 2 (index 1)
    (-9.16, 3.50, 0.00),  # 3 etc...
    (9.16, 3.50, 0.00),
    (5.66, -5.66, 5.66),  # 5
    (0.00, 9.16, -3.50),
    (-5.66, -5.66, 5.66),
    (-3.50, 0.00, -9.16),
    (0.00, 9.16, 3.50),
    (-5.66, -5.66, -5.66),  # 10
    (5.66, 5.66, 5.66),
    (0.00, -9.16, -3.50),
    (3.50, 0.00, 9.16),
    (5.66, 5.66, -5.66),
    (0.00, -9.16, 3.50),  # 15
    (-5.66, 5.66, -5.66),
    (-9.16, -3.50, 0.00),
    (9.16, -3.50, 0.00),
    (-5.66, 5.66, 5.66),
    (3.50, 0.00, -9.16),  # 20
)

# HARDWARE SETUP .nextSibling
# Enable power to audio amp, etc.
external_power = DigitalInOut(board.EXTERNAL_POWER)
external_power.direction = Direction.OUTPUT
external_power.value = True

# I2S audio out
audio = audiobusio.I2SOut(board.I2S_BIT_CLOCK, board.I2S_WORD_SELECT, board.I2S_DATA)

# LIS3DH accelerometer
lis3dh = adafruit_lis3dh.LIS3DH_I2C(board.I2C())
lis3dh.range = adafruit_lis3dh.RANGE_4_G

# Battery monitor, if present (requires two 10K resistors)
if BATT_SENSE:
    adc = analogio.AnalogIn(BATT_SENSE)
# FUNCTIONS

```python
def play(index, block=True):
    """Play one WAV file from the WAV_FILES table. Pass in table index (0-n) and optional 'block' flag (if True, function blocks until audio is finished playing)."""
    wave_file = open(f'{WAV_PATH}/{WAV_FILES[index]}.wav', "rb")
    wave = audiocore.WaveFile(wave_file)
    audio.play(wave)
    while block and audio.playing:
        pass

def freefall_wait():
    """Watch for freefall condition (low G for FREEFALL_MIN_DURATION)."""
    start_time = time.monotonic()
    while time.monotonic() - start_time < FREEFALL_MIN_DURATION:
        accel = lis3dh.acceleration
        if accel[0] ** 2 + accel[1] ** 2 + accel[2] ** 2 > FREEFALL_THRESHOLD:
            start_time = time.monotonic()  

# pylint: disable=redefined-outer-name

def settle_wait():
    """Wait for die to stabilize (steady ~1G) on one number. Returns index of corresponding audio file (0.19 for faces 1-20), or -1 if acceleration did not stabilize within SETTLE_TIMEOUT."""
    start_time = time.monotonic()
    prev_face = -1
    while time.monotonic() - start_time < SETTLE_TIMEOUT:
        accel = lis3dh.acceleration
        if 77.89 < mag < 116.35:  # ~1G
            face = -1
            min_dist = 1000000
            for index, vec in enumerate(FACE_VECTORS):
                dist_sq = (
                    (accel[0] - vec[0]) ** 2
                    + (accel[1] - vec[1]) ** 2
                    + (accel[2] - vec[2]) ** 2
                )
                if dist_sq < min_dist:  # New closest match?
                    min_dist = dist_sq  # Save closest distance^2
                    face = index  # Save index of closest match
            if face != prev_face:
                prev_face = face
            elif time.monotonic() - settle_start > SETTLE_TIME:
                return face
        else:
            prev_face = -1
    return -1
```

# STARTUP & MAIN LOOP

```python
play(29, False)  # Play greeting (non-blocking)

# pylint: disable=invalid-name, used-before-assignment
while True:
    freefall_wait()  # Wait for roll
    face = settle_wait()  # Wait for landing (or timeout)
    if face >= 0:  # Not timeout...
        if face == 2:  # If '3' face
            if randint(0, 9) == 0:  # 1-in-10 chance of...
                face = 30  # Alternate 'face 3' track
        play(randint(20, 22))  # One of 3 random announcements
```

© Adafruit Industries
play(face)  # Face number
if face != 30:  # If not the alt face...
    if face <= 3:  # Index 0-3 (face 1-4) = bad
        play(randint(23, 25))  # Random jab
    elif face >= 16:  # Index 16-19 (face 17-20) = good
        play(randint(26, 28))  # Random praise
elif BATT_SENSE:
    volts = adc.value / 65535 * 3.3 * 2
    if volts < BATT_LOW:
        time.sleep(0.5)
        play(randint(31, 32), False)

Upload the Code and Libraries to the RP2040 Prop-Maker Feather

After downloading the Project Bundle, plug your RP2040 Prop-Maker Feather into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called CIRCUITPY. Unzip the folder and copy the following items to the RP2040 Prop-Maker Feather's CIRCUITPY drive.

- lib folder
- WAVs folder
- code.py

Your RP2040 Prop-Maker Feather CIRCUITPY drive should look like this after copying the lib folder, WAVs folder and the code.py file. Other files on the drive like boot_out.txt and settings.toml can be ignored, they're just part of how CircuitPython works.
OPTIONAL: Battery Monitoring

If you've added two 100K resistors as explained on the “Circuit Diagram” page, the die can monitor and announce when the battery is due for recharging. To enable, look for this line in the code (around line 67 or so):

```python
BATT_SENSE = None  # Assign analog pin if voltage divider present
```

Change this to reference an analog input pin where the resistors are connected. Our wiring diagram showed pin A3, so the change would be:

```python
BATT_SENSE = board.A3  # Assign analog pin if voltage divider present
```

How the CircuitPython Code Works

Sensing a die roll is hard. The code uses a similar approach to the earlier Arduino version of this project(): it watches for a free-fall state, where readings from the accelerometer will all be near zero. You can see this in action in the `freefall_detect()` function.
This means the die is triggered by dropping a few inches, or a light toss upward. Gently rolling off an inclined hand won’t suffice.

Once a roll is detected, it then waits for the accelerometer to settle on a single face at around 1 G (9.8 meters/sec²). It then plays a WAV file to announce which face, and a little jab or praise for low or high values.

The code uses a classic math trick to speed things along so detection is more responsive. Getting the magnitude (length) of a vector (such as from an accelerometer) usually involves a square root...one of the costlier mathematical operations you could ask a microcontroller to do, especially one with no FPU:

\[
magnitude = \sqrt{x^2 + y^2 + z^2}
\]

Since we’re just comparing this magnitude to certain thresholds (<0.3G for free-fall, and 0.9–1.1G for settling), not using the actual value in further operations, we eliminate the square root and instead square the values we’re comparing against.

\[
magnitude^2 = x^2 + y^2 + z^2
\]

Free-fall detect is looking for 0.3 G (2.94 m/s²) or less. To use the above trick, the square root operation is skipped and we instead compare against 2.94², or about 8.65:

```
FREEFALL_THRESHOLD = 8.65 # Near-freefall = 0.3G ^ 2
```

Similarly, that’s why the “around 1G detect” uses such large numbers, much more than the 9.8 ±10% one would expect. These are magnitudes squared:

```
if 77.89 < mag < 116.35: # ~1G
```

It’s probably not a huge performance difference with the D20, but some day when you’re short on CPU cycles this trick might save your tail.
Wiring

Slide Switch Wire
Use the silicone ribbon cable to create a 2-pin cable. Measure and cut the wire to be 7.5cm in length.

Wiring Slide Switch
Cut one of the three pins on the slide switch, either the far left or right but not the middle. The, cut the two remaining pins to half their length.

Solder the 2-pin cable to the pins on the slide switch.

Connect Slide Switch
Get the slide switch ready to connect to the Feather.
Connected Slide Switch
The two wires are soldered to the pins on the bottom of the Feather.

Solder one wire to the ground pin (G) and the other to the enable pin (EN). It doesn't matter which wire goes to which pin.

Soldered Slide Switch
Double check the wires have been properly soldered to the pins on the RP2040 PropMaker Feather.

Speaker Wires
Cut the cable from the speaker so the wires are 8cm in length.

Remove a bit of insulation from the two wires and tin them using a bit of solder.

Optionally save the 2-pin connector for another project.
Battery Monitoring (optional)
This die has the optional 100K resistors installed for battery monitoring. These can reach point-to-point across the board, using some heat-shrink tube to prevent making contact with other components as the die bounces about. One pin of the servo header is borrowed as a ground point.

Assembly

Magnets
Gather up the eight neodymium magnets needed for the two halves.
Glueing Magnets
To permanently adhere the magnets, we suggest using Starbond's medium super glue.

Use the included fine precision micro tip to apply small drops of glue to the magnet cavities.

Carefully place the four magnets into the glue applied cavities of one half of the D20.

Allow the glue to fully set before proceeding to the second half.

Glueing Magnets Continued
Once the glue has fully set, add the second set of magnets to the permanently bonded magnets. This ensures the polarities are matching.

Apply small drops of glue to the four cavities on the second half of the D20.

Carefully fit the second half onto the magnets so the cavities are properly lined up.

Press the two halves together and let the glue dry.

Only use a few drops of glue to avoid squeeze-out. Allow glue to fully set for at least two hours before handling.
It’s easiest to assemble the die with magnets all “north up” on one half, and “south up” on the opposite half…but that does create a situation where the die could close wrong.

If you’re very patient and carefully label the polarity of your magnets, a better approach is to have one polarity facing up on the magnets near the 3, 4, 17 and 18 faces, and the opposite polarity on faces 2, 7, 14 and 19. The magnets will naturally resist improper closing when assembled this way.

**Tap Standoffs**
Use an M2.5 size screw tap to create the proper threads for the two standoffs.

**Install Speaker**
Once the glue has fully set on both halves, pull them apart.

Remove the sticker backing from the front of the speaker.

Orient the speaker with the half of the D20 with the oval shaped flat surface. Then, press the speaker onto the surface.
Adding Counter Weights
To balance the two halves, we suggest using two US pennies as counterweights.

Place the two pennies into the cavity, they should have a loose fit.

Install Cap
Press the cap onto the counter weight cavity. It should have a tight tolerance.

Optionally glue the pennies in place for permanently bonded counter weights.
Speaker Wires
Get the Feather ready to connect the speaker wires.

Install Speaker Wires
Insert the two wires from the speaker into the corresponding terminals on the Feather.

Use a screwdriver with a 2.5 sized flat head to secure the speaker wires.

Connect Battery
Get the 400mAh battery. Connect the cable from the battery to the 2-pin JST connector on the Feather.

If the Feather powers on, use the slide switch to turn it off.
Secure Battery
The speaker features a second sticky side on surface of the magnet driver. Remove the protective cover by peeling it off.

Then, position the 400mAh battery and press it onto the sticky surface of the speaker.

Install Switch
Position the slide switch with the built-in holder on the half of the D20.

Insert and press the slide switch in between the walls of the built-in switch holder.
Installing Feather
Orient the Feather with the D20 so the mounting holes line up with the built-in standoffs.

Secure Feather
Insert and fasten the two M2.5 x 6mm long screws to secure the Feather to the half of the D20.

Do not over tighten screws! If threads become loose, try using a longer M2.5 sized screw (max length: 10mm).
Final Check
Take a moment to inspect the components are properly secured in place.

Use the slide switch to power on the Feather before joining the two halves together.

Closing Halves
The halves can only be joined one way.

Orient the two halves so the magnets are matching.

Slowly bring the two halves together and keep your fingers away from the edges.

Keep finger tips away from the edges when closing! The magnets are really strong and can pinch your fingers!

Final Build
Congratulations on your build!

Go ahead and give it a roll. For best results, the accelerometer readings work better when it's free falling at least 6 inches (15cm) away from the surface.