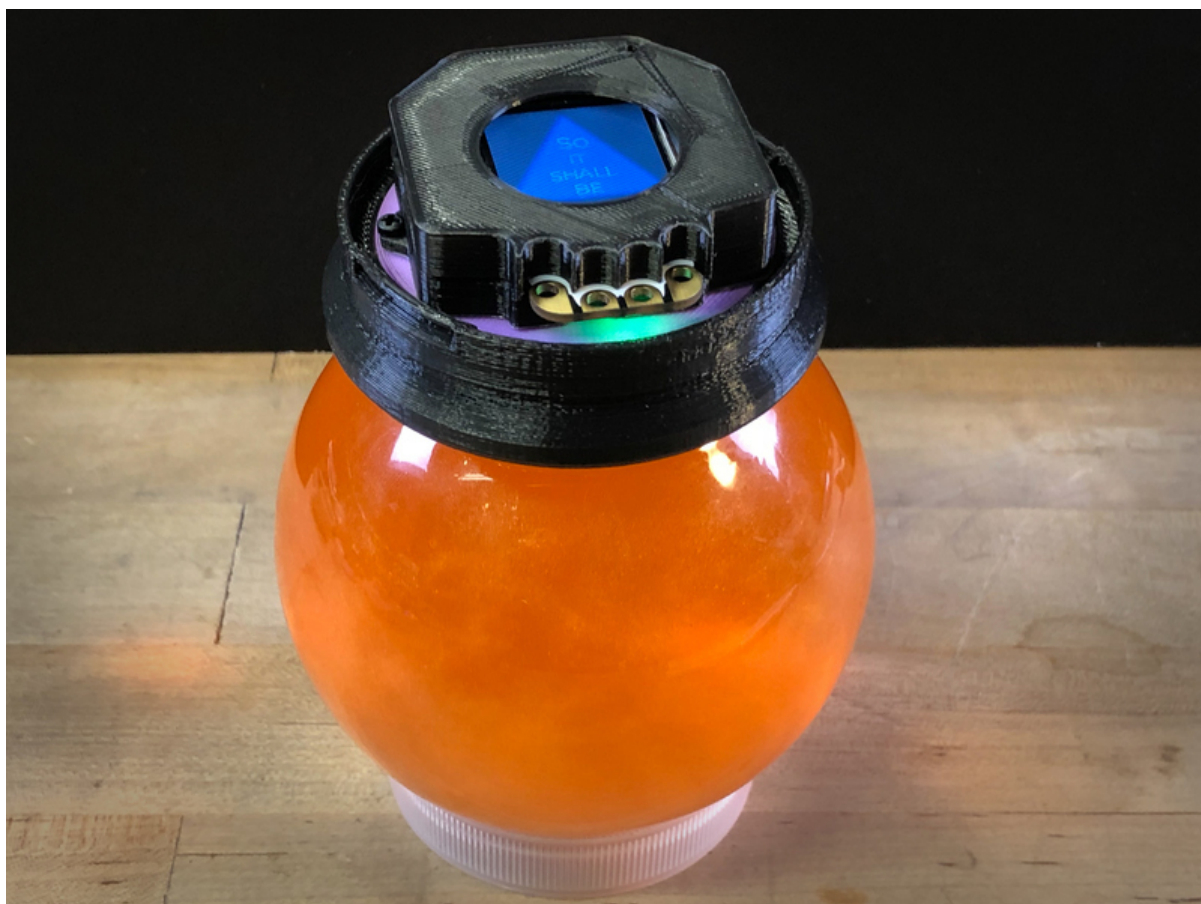




HalloWing Magic 9 Ball

Created by John Park



<https://learn.adafruit.com/hallowing-magic-9-ball>

Last updated on 2024-06-03 02:27:57 PM EDT

Table of Contents

Overview	3
<hr/>	
• Materials & Tools	
Build the Magic 9 Ball	4
<hr/>	
• Mix the Mystery Fluid	
• HalloWing Prep	
• Ready the Battery	
• Prepare the Cap and Mount	
• Assemble	
• Screw Cap onto Globe	
Code with CircuitPython	15
<hr/>	
• Image Player Firmware	
• File Format	
• Magic 9 Ball Code	
• Answer Any Question	
Microcontroller Selection Oracle	21
<hr/>	

Overview

Making difficult decisions can be tricky and stressful. This project will help you take off some of the pressure, instead letting your HalloWing-based Magic 9 Ball do the heavy lifting of your decision making!

By harnessing the HalloWing's accelerometer to detect when the ball is turned upside down and right side up (or shaken), we'll then run a highly mysterious algorithm (OK, it's just a random number generator, sorry) to choose an image file to show on the TFT display.

A stylish housing filled with swirly mystery fluid completes the effect. Will you love building this project?

INDICATIONS SAY YES

1 x [Adafruit HalloWing M0 Express](https://www.adafruit.com/product/3900)

<https://www.adafruit.com/product/3900>

Skull-shaped ATSAMD21 board w 1.44" 128x128 TFT display

1 x [Lithium Ion Polymer Battery](https://www.adafruit.com/product/1578)

<https://www.adafruit.com/product/1578>

3.7v 500mAh

1 x [DIY Snow Globe Kit](https://www.adafruit.com/product/3722)

<https://www.adafruit.com/product/3722>

108mm Diameter

1 x [Black Nylon Screw and Stand-off Set](https://www.adafruit.com/product/3299)

<https://www.adafruit.com/product/3299>

M2.5 Thread

Materials & Tools

In addition to the above parts, you will want the following:

- Food dye
- Water
- Glycerin (optional)
- Super Pearl Dust (used in baking to add shimmer, [thanks to Epbot for the tip \(https://adafru.it/Cz7\)!](https://adafru.it/Cz7))
- E6000 glue (optional)
- Double stick foam tape

- Optional: 3D printed cap, mounting plate, and HalloWing case (if you don't have a 3D printer, you can send the files to be printed, or simply double-stick tape your HalloWing to the regular snow globe cap)
-

Build the Magic 9 Ball



Mix the Mystery Fluid

The Magic 9 Ball derives much of its power from the mysterious roiling fluid contained within the orb. Here, for the very first time, we shall reveal the secrets of the unfathomable fluid!



Get ready for the secrets to be revealed! -- We'll mix water with super pearl dust (also called shimmer dust), food coloring, and optional glycerin. The dye gives the mix its color, while the pearl dust provides the moving, glittering view into the complex fluid dynamics going on in there while it is thinking.



Add a very small amount of the pearl dust to the empty globe.



Fill halfway with water, then add dye and swirl it around. I used five drops of yellow to two drops of red here.

Fill the rest of the way with water, up to the level of the stopper cap.



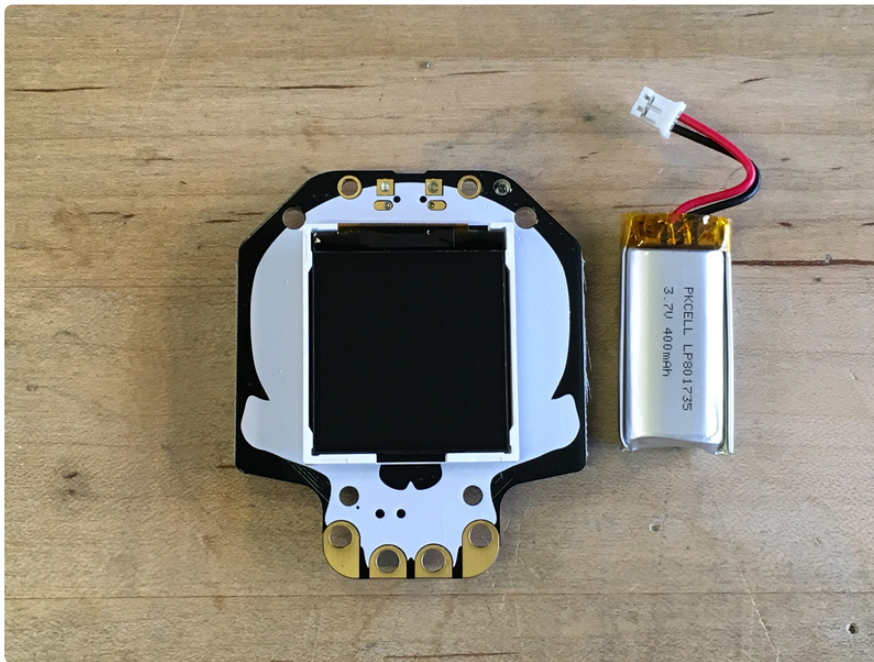
Press the stopper cap into place. You can glue this in with silicone sealant or E6000 glue if you like, for extra safety.



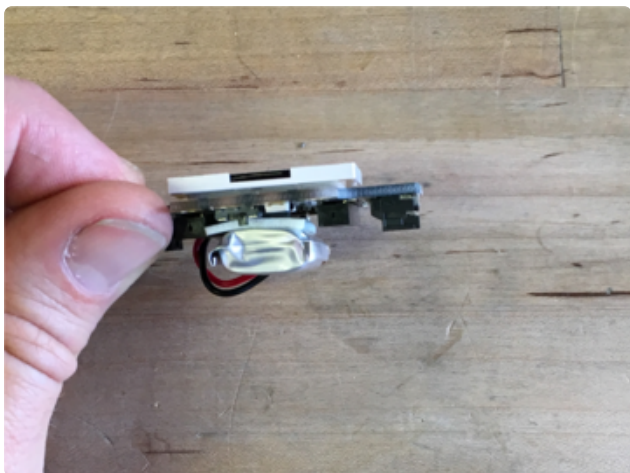
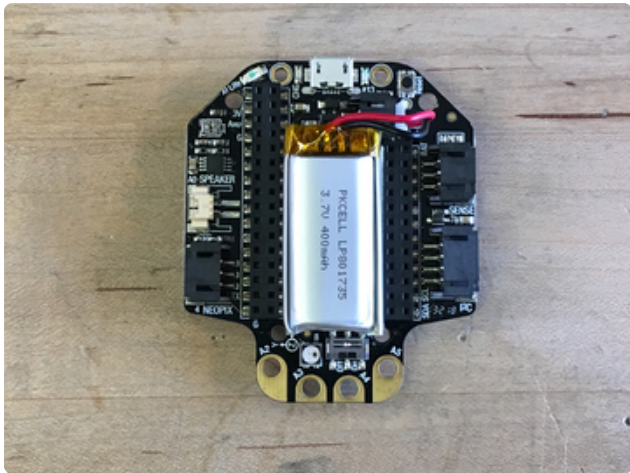
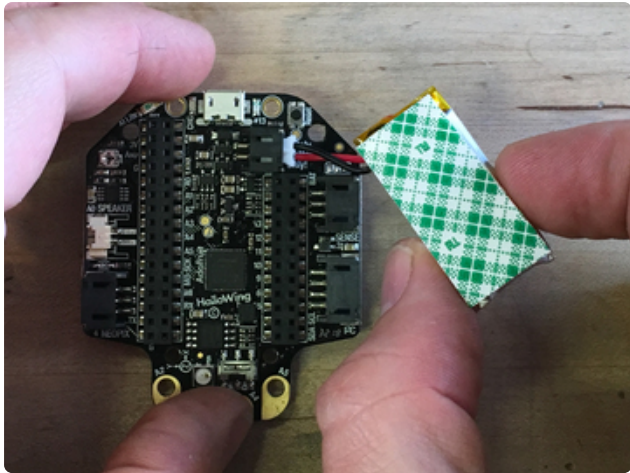
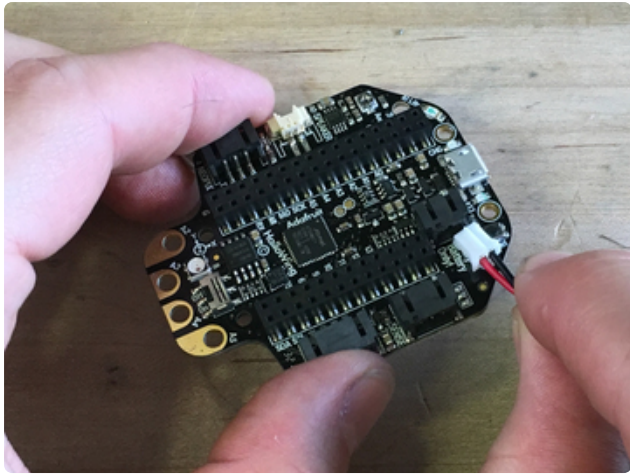
This is a fun time to test it out -- screw on the original cap and swirl the orb to see the intricate fluid motion!



HalloWing Prep



The HalloWing will be the brains and display of our Magic 9 Ball.

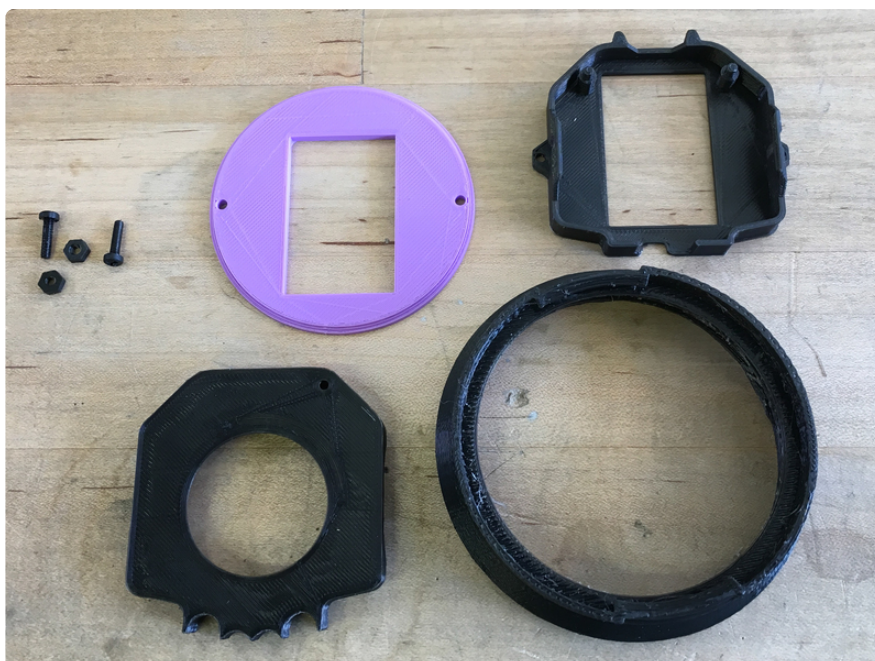


Ready the Battery

Plug your Lipoly battery's JST plug into the battery port on the HaloWing.

Use double-stick foam tape to affix the battery to the board as shown.

Prepare the Cap and Mount



The cap, spacer and mount are all 3D printed. I modified these parts that were created by the Ruiz Bros. More info on the [HaloWing M0 Express Wearable Case here \(https://adafru.it/CzV\)](https://adafru.it/CzV), and the threaded ring came from their excellent [Automatic Dice Roller project \(https://adafru.it/CzW\)](https://adafru.it/CzW).

Below are the model files you'll need. These were printed in PLA plastic with 0.2mm layer height.

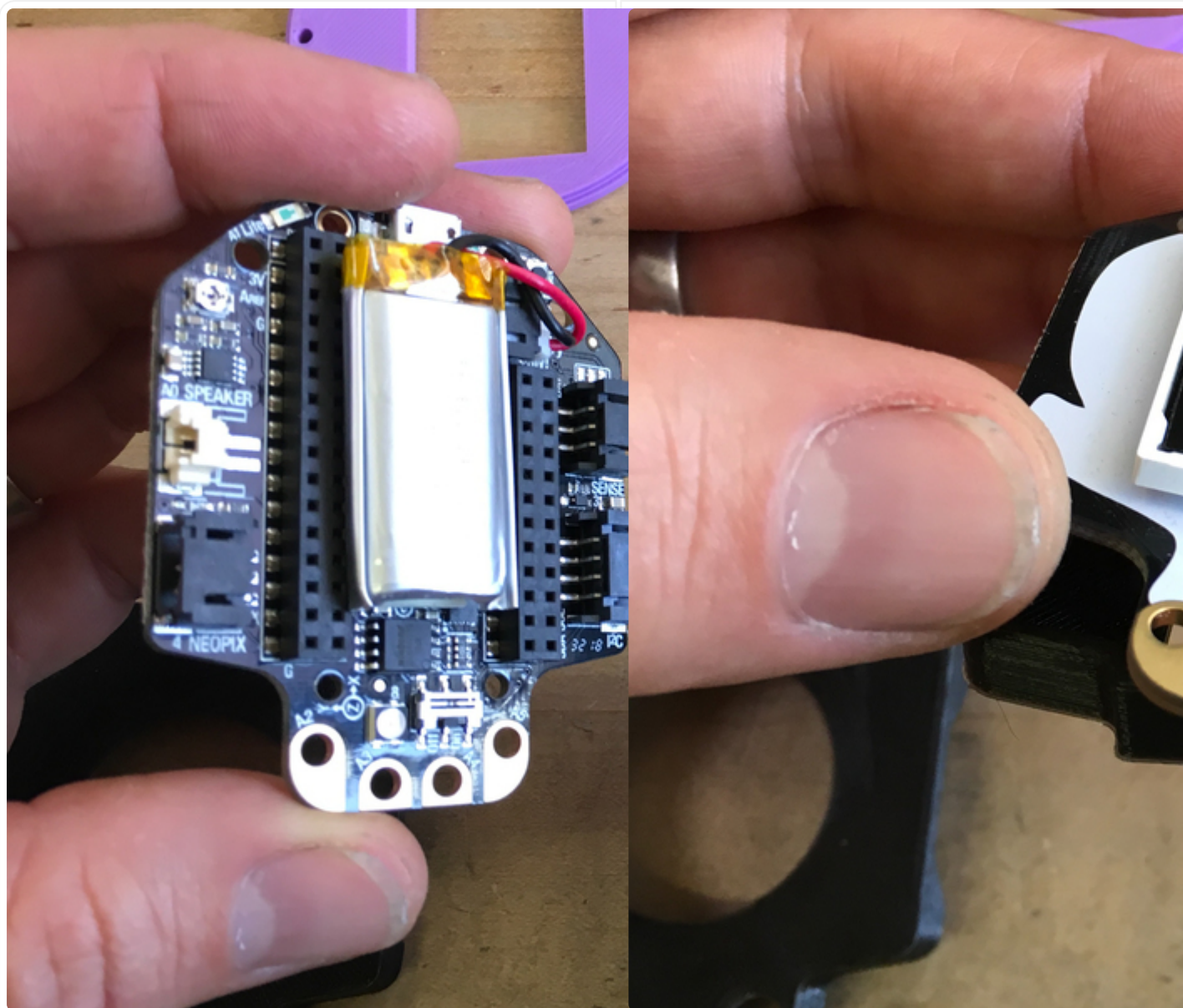
nine_ball_models.zip

<https://adafru.it/CzU>

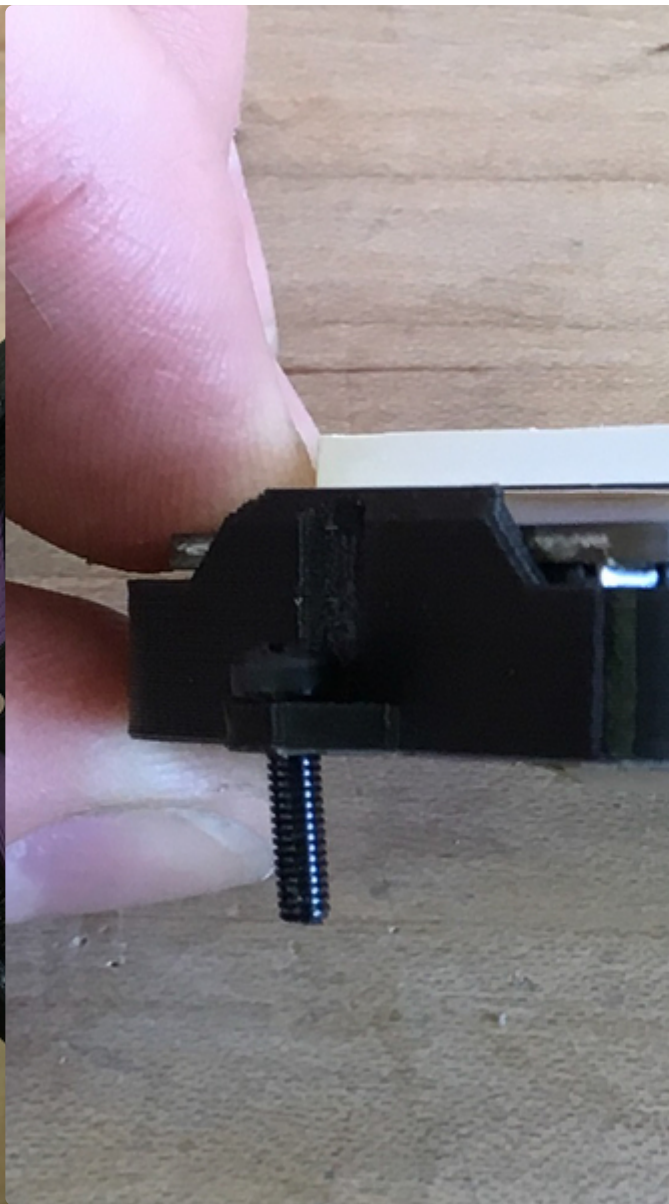
You'll also need two 10mm long M2.5 nylon screws and nuts to affix the HaloWing case bottom to the spacer.

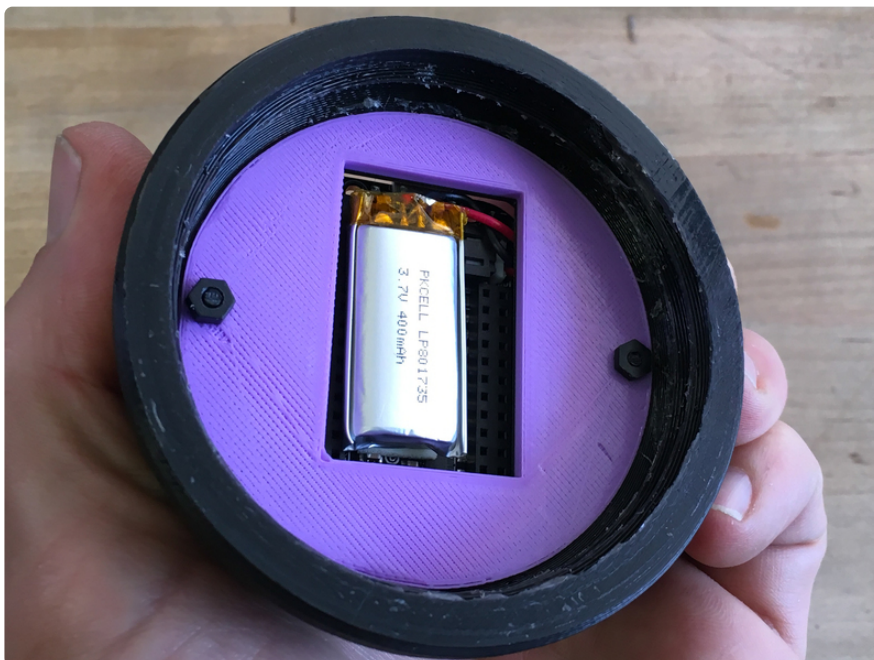
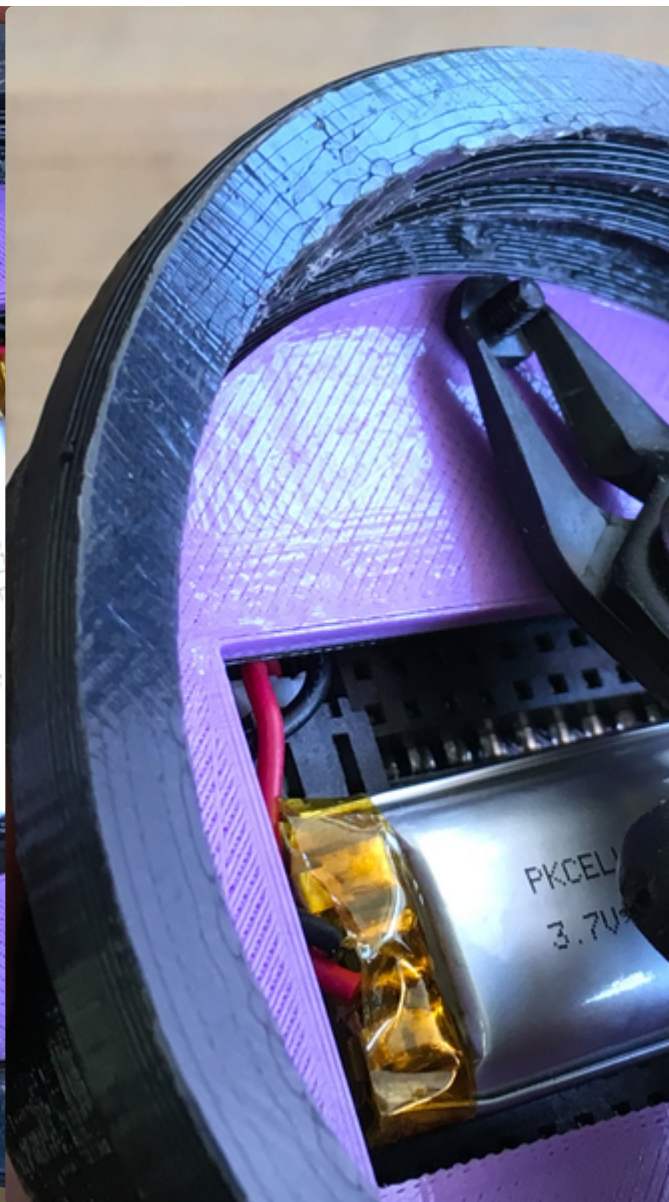
Assemble

Snap the HaloWing into place in the bottom mount, then place the spacer into the cap as shown.



Screw the mount into place, thread on the nuts and tighten them. Then trim the screws flush with diagonal cutters. This prevents the screws from interfering with the fluid cap.



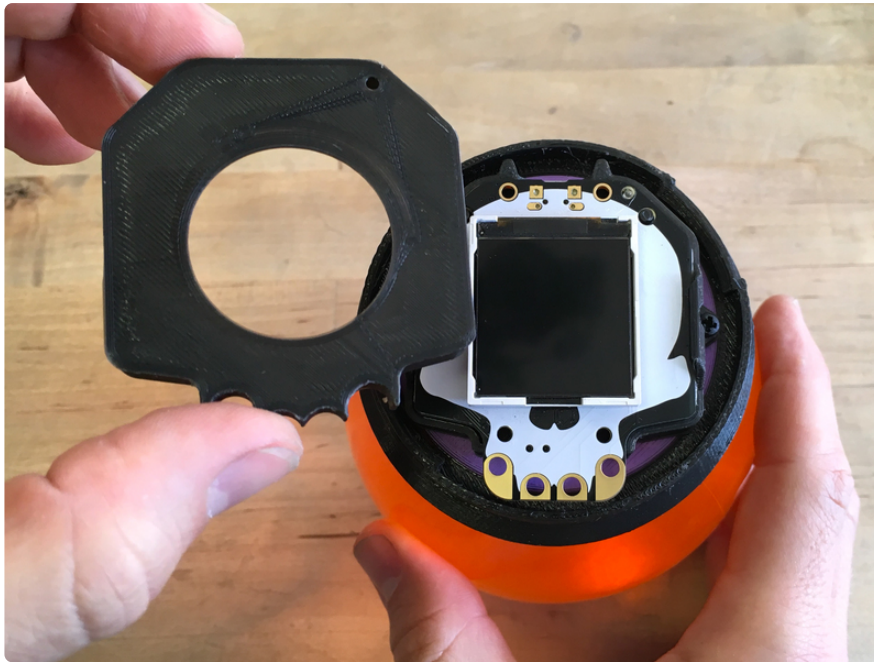


Screw Cap onto Globe

Now, we can screw the cap onto the globe -- just make sure the stopper is securely in place!



Finally, snap the HaloWing case cover into place.



Plug in a USB cable and next we'll program it!



Code with CircuitPython

Image Player Firmware

This project requires CircuitPython version 4.0 or higher. Refer to the [HalloWing CircuitPython page \(https://adafru.it/CmJ\)](https://adafru.it/CmJ) for ensuring you upload the HalloWing version of CircuitPython 4.0 or greater.

To load the CircuitPython firmware, plug in the HalloWing over a known good USB cable to your computer and then double-click the reset button on the HalloWing. This will put it into bootloader mode, allowing you to change the firmware. You'll see a USB drive appear called **HALLOWBOOT**.

Drag the .uf2 file you downloaded onto the **HALLOWBOOT** drive. Once it copies over, it will automatically restart and show a flash drive named **CIRCUITPY**. Then you are ready to load a CircuitPython program that can take advantage of the display capabilities.

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

<https://adafru.it/Em7>

File Format

To prepare your graphics for the HalloWing, you'll need to size them to 128 x 128 pixels. You can do this in nearly any graphics program, such as Photoshop, GIMP, Preview on osx, ImageMagik, MS Paint, you name it!

Save your file as a 24-bit .bmp file. Here you can see some example graphics.

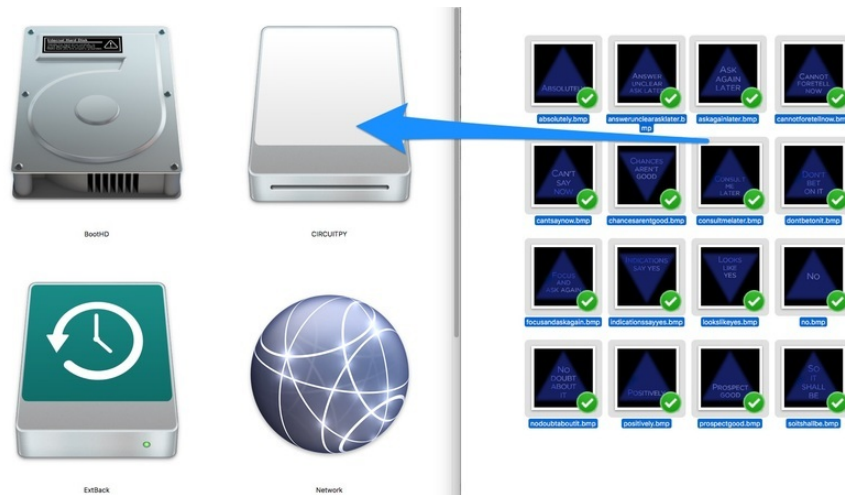


Here is the full set of answers for your Magic 9 Ball. Download the file and then unzip it.

Drag the .bmp files onto your **CIRCUITPY** drive as shown.

nine_ball_answers.zip

<https://adafru.it/CDi>



Tip: Images inside of a folder on your HalloWing won't be seen by this image player code below, only files at the root level of the HalloWing are seen. So, you can swap files in and out of folders to quickly change your image set!

Magic 9 Ball Code

This program is pretty simple in what it does -- it loads a random image onto the screen using `displayio.TileGrid` while fading the backlight up with `board.DISPLAY.brightness`

Then, whenever the HalloWing's accelerometer detects the board has been turned upside down or shaken, it dims the backlight, selects and displays a new image, and brings the backlight up again.

Adafruit really likes using the Mu editor to edit the CircuitPython code. [See this guide on loading and using Mu](https://adafru.it/ANO) (<https://adafru.it/ANO>).

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

# Magic 9 Ball
# Turn HalloWing face down and then face up to change images at random
# place 128 x 128 pixel 24-bit .bmp images at root level of HalloWing

import time
import os
import random
import board
import displayio
import adafruit_lis3dh

splash = displayio.Group()
board.DISPLAY.root_group = splash

SENSITIVITY = 5 # reading in Z direction to trigger, adjustable
```



```

images = list(filter(lambda x: x.endswith("bmp"), os.listdir("/")))

i = random.randint(0, (len(images)-1)) # initial image is randomly selected

# Set up accelerometer on I2C bus, 4G range:
i2c = board.I2C() # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C() # For using the built-in STEMMMA QT connector on a
# microcontroller
ACCEL = adafruit_lis3dh.LIS3DH_I2C(i2c, address=0x18)

ACCEL.range = adafruit_lis3dh.RANGE_4_G

while True:
    shaken = False

    print("Image load {}".format(images[i]))
    # CircuitPython 6 & 7 compatible
    try:
        f = open(images[i], "rb")
        odb = displayio.OnDiskBitmap(f)
    except ValueError:
        print("Image unsupported {}".format(images[i]))
        del images[i]
        continue
    face = displayio.TileGrid(odb, pixel_shader=getattr(odb, 'pixel_shader',
displayio.ColorConverter()))

    # # CircuitPython 7+ compatible
    # try:
    #     odb = displayio.OnDiskBitmap(images[i])
    # except ValueError:
    #     print("Image unsupported {}".format(images[i]))
    #     del images[i]
    #     continue
    # face = displayio.TileGrid(odb, pixel_shader=odb.pixel_shader)

    splash.append(face)
    # Wait for the image to load.
    try:
        board.DISPLAY.refresh(target_frames_per_second=60)
    except AttributeError:
        board.DISPLAY.wait_for_frame()

    # Fade up the backlight
    for b in range(101):
        board.DISPLAY.brightness = b / 100
        time.sleep(0.01) # default (0.01)

    # Wait until the board gets shaken
    while not shaken:
        try:
            ACCEL_Z = ACCEL.acceleration[2] # Read Z axis acceleration
        except OSError:
            pass
        # print(ACCEL_Z) # uncomment to see the accelerometer z reading
        if ACCEL_Z > SENSITIVITY:
            shaken = True

    # Fade down the backlight
    for b in range(100, 0, -1):
        board.DISPLAY.brightness = b
        time.sleep(0.005) # default (0.005)

    splash.pop()

    i = random.randint(0, (len(images)-1)) # pick a new random image
    # print("shaken")
    faceup = False
    i %= len(images) - 1

```


Copy the code and then paste it into Mu. Save the file to your **CIRCUITPY** board as **code.py**.

The board will restart and then play an image. Try turning the Magic 9 Ball upside down and then back up -- you'll see a new image appear!

Answer Any Question

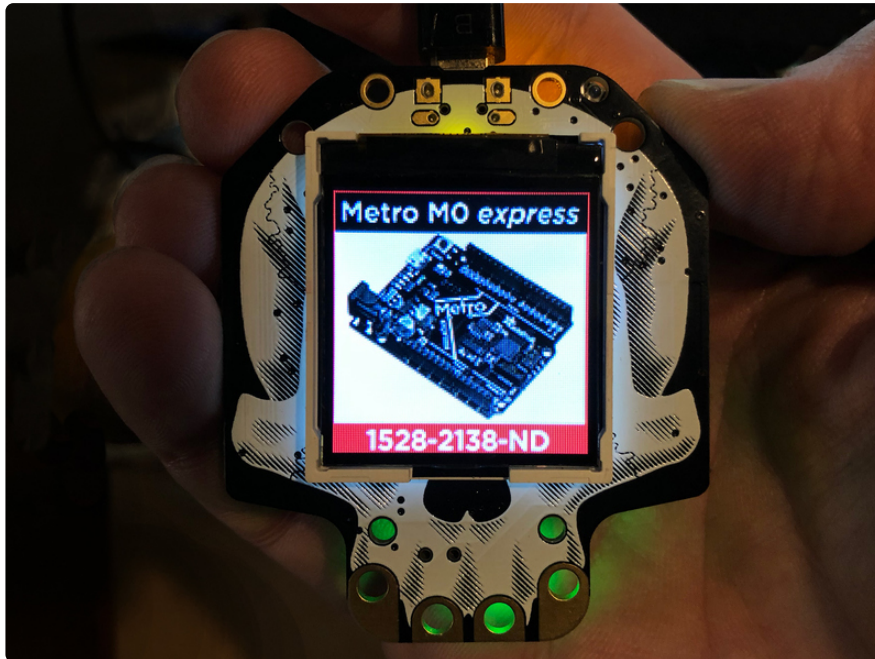
You can now unplug the USB cable (plug it back in any time to charge the battery) and use the Magic 9 Ball!

Ask it a yes or no question, such as "Have we reached the mother ship yet?", then turn the ball over (or give it a shake), then turn it right side up and the answer will present itself in all of its wisdom!





Microcontroller Selection Oracle



Here's another interesting use of the Magic 9 Ball code -- you can create a random microcontroller selector for all those times you just aren't sure which amazing CircuitPython-capable Adafruit board to get from [Digi-Key](https://www.digikey.com), who are going all-in on [CircuitPython](https://adafru.it/CAj) (<https://adafru.it/CAj>)!

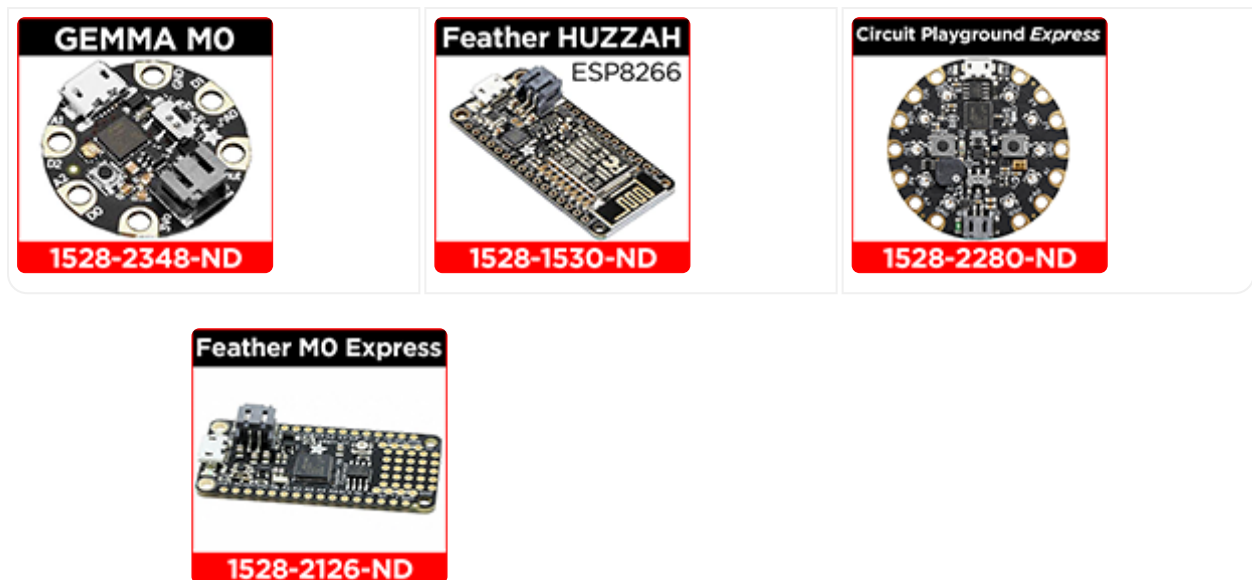
This will display a randomly selected board and show the product name, image, and Digi-Key product number!

Simply load these images linked below onto the board (you can place the Magic 9 Ball images into a folder you make on the HalloWing and they won't be selected for viewing).

dk_quest.zip

<https://adafru.it/CAk>





This is just an example we created for our friends at Digi-Key, but we hope it also serves as a leaping off point for you to come up with your own creative uses for the image randomization of the Magic 9 Ball!