Table of Contents

Overview
- Feather RP 2040 + Prop-Maker
- Magnetic USB plug
- Parts

Circuit Diagram

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

Coding the LED Rocket Lamp
- Upload the Code and Libraries to the Feather RP2040
- About the Enceladus Hiss mp3 File
- How the CircuitPython Code Works
- The Mixer
- NeoPixels
- Rotary Encoder
- The Loop
- Turn Up the Volume
- Turn Up the Brightness

3D Printing
- Parts List
- Slice with settings for PLA material
- Supports
- Build Plate Adhesion

Assemble
- Solder Headers
- Solder extension wires
- Speaker mount
- Coil wires
- Fan mount
- Feather mount
- LED rings
- LED Ring JST
- Mount fan
- Connect wires
- Thread wires
- Align tabs
- Screw tabs
- DIY USB cables
- Mount pipe
- Attach Lid
- Thread wires
- Rotary assemble
- Mount rotary
• Power switch
• Mount switch
• USB adapter
• Case Lid
• Flame assembly
• Complete
Overview

You can build a SciFi inspired Rocket Lamp with electronics from Adafruit, coded with CircuitPython.

This features NeoPixel LEDs and a speaker to create a rocket-themed white noise generator.

At the base, it’s got an LED toggle switch for powering on the device.

A Stemma QT rotary encoder features two modes for adjusting the brightness of the LEDs and volume of the sound effects.

We think it’s a great project for young makers to inspire them while they’re awake and help soothe them to sleep.
Feather RP 2040 + Prop-Maker

Powered by the Adafruit Feather RP2040 and the Prop-Maker FeatherWing, these two boards are the perfect match for making advanced props with LEDs and sound effects!

This is a great basis for toys or props that require LEDs, sounds, motion sensing, even 3W LEDs, or in this case a fan!

It features a mini fan, speaker, NeoPixels, LED toggle switch and a Stemma QT rotary encoder.
Magnetic USB plug

This USB cable features magnetic tips and provides the lamp with 5V of USB power.

This cable makes managing lots of dev boards or gadgets super easy.

The removable magnetic end that lets you quickly connect or disconnect with magnetic alignment. It's like having a 'MagSafe' connector for everything. Works with any and all devices, for both data-sync and power, for 5V up to 2A.
Parts

Assembled Adafruit Prop-Maker FeatherWing
The Adafruit Feather series gives you lots of options for a small, portable, rechargeable microcontroller board. Perfect for fitting into your next prop build! This FeatherWing will...
https://www.adafruit.com/product/4145

Adafruit Feather RP2040
A new chip means a new Feather, and the Raspberry Pi RP2040 is no exception. When we saw this chip we thought "this chip is going to be awesome when we give it the Feather..."
https://www.adafruit.com/product/4884

Miniature 5V Cooling Fan with Molex PicoBlade Connector
Looking for another way to keep your project cool? Hook up this 5V Mini Cooling Fan and prevent from overheating! Of course, it's also...
https://www.adafruit.com/product/4468
<table>
<thead>
<tr>
<th>1 x Rotary Encoder</th>
<th>Rotary Encoder</th>
<th><a href="https://www.adafruit.com/product/377">https://www.adafruit.com/product/377</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x STEMMA QT / Qwiic JST SH 4-Pin Cable - 300mm long</td>
<td>STEMMA QT / Qwiic JST SH 4-Pin Cable - 300mm long</td>
<td><a href="https://www.adafruit.com/product/5384">https://www.adafruit.com/product/5384</a></td>
</tr>
<tr>
<td>1 x 16 NeoPixel Ring</td>
<td>NeoPixel Ring - 16 x 5050 RGB LED</td>
<td><a href="https://www.adafruit.com/product/1463">https://www.adafruit.com/product/1463</a></td>
</tr>
</tbody>
</table>

**Illuminated Toggle Switch with Cover - Red**
Fire up your engines because these are the hot-rods of toggle switches! Equipped with an aerodynamic red protective casing, and a red LED at the tip of the switch, this...
https://www.adafruit.com/product/3218

**Woven USB A Cable with Magnetic Tips - Micro B, Type C, and iOS**
Some days we’re feeling extra fancy here at the ‘fruit warehouse, and we have a big soft spot for woven fabric cables. Like, peep
https://www.adafruit.com/product/5652

**Adafruit I2C Stemma QT Rotary Encoder Breakout with NeoPixel**
Rotary encoders are sooo much fun! Twist em this way, then twist them that way. Unlike potentiometers, they go all the way around and often have little detents for tactile feedback....
https://www.adafruit.com/product/4991
1 x NeoPixel Ring - 24 x 5050 RGB LED
https://www.adafruit.com/product/1586

1 x JST PH 2mm 3-pin Plug-Plug Cable
https://www.adafruit.com/product/4336

1 x DIY USB or HDMI Cable Parts - 30 cm Ribbon Cable
https://www.adafruit.com/product/3562

1 x DIY USB Cable Parts - Straight Type C Plug
https://www.adafruit.com/product/4108

1 x DIY USB Cable Parts - Straight Type A Jack
https://www.adafruit.com/product/4110

1 x USB A to USB C Adapter
https://www.adafruit.com/product/4175

1 x Oval Speaker
https://www.adafruit.com/product/3923

1 x Molex PicoBlade 2 Pin Cable
https://www.adafruit.com/product/4720

8 x M2.5x6mm

6 x M2x6mm Screws
https://www.albanycountyfasteners.com/2MM-x-4-Phillips-Flat-Head-Machine-Screw-p/1011-1000.htm

3 x M2x8mm Screws
https://www.albanycountyfasteners.com/2MM-x-4-Phillips-Flat-Head-Machine-Screw-p/1011-1000.htm

Circuit Diagram

The wiring diagram below provides a visual reference for connecting the components. It is not true to scale, it is just meant to be used as reference. This diagram was created using the Fritzing software package.
Take a moment to review the components in the circuit diagram. This illustration is meant for referencing wired connections - the length of wire, position and size of components are not exact.

---

**Install 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

---

Coding the LED Rocket Lamp

Once you've finished setting up your Feather RP2040 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.

```python
# SPDX-FileCopyrightText: 2023 Liz Clark for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import digitalio
import audiomixer
import audiomp3
from audiopwmio import PWMAudioOut as AudioOut
from adafruit_seesaw import seesaw, rotaryio, digitalio as seesaw_io, neopixel as seesaw_neopixel
from adafruit_led_animation.animation.pulse import Pulse
import neopixel

# wait a little bit so USB can stabilize and not glitch audio
time.sleep(3)

# enable propmaker speaker output
enable = digitalio.DigitalInOut(board.D10)
enable.direction = digitalio.Direction.OUTPUT
enable.value = True

# enable fan pin
fan_pin = digitalio.DigitalInOut(board.D11)
fan_pin.direction = digitalio.Direction.OUTPUT
fan_pin.value = True

# speaker pin on the propmaker
audio = AudioOut(board.A0)

# create mixer instance
mixer = audiomixer.Mixer(voice_count=1, sample_rate=22050, channel_count=1, bits_per_sample=16, samples_signed=True)

# attach mixer to audio playback
audio.play(mixer)
```
# open mp3 audio file
audio_file = audiomp3.MP3Decoder(open("Enceladus_Hiss_NASA.mp3","rb"))
# play audio file in first channel of mixer
mixer.voice[0].play(audio_file, loop=True)
# set mixer channel level
mixer.voice[0].level = 0

# propmaker neopixel pin
pixel_pin = board.D5
num_pixels = 35
pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=0.3, auto_write=False)

# define neopixel colors
RED = (255, 0, 0)
BLUE = (0, 0, 255)

# pulse animation
pulse = Pulse(pixels, speed=0.1, color=RED, period=3)

# i2c setup for rp2040 feather stemma port
i2c = board.STEMMA_I2C()

# rotary encoder
enc0 = seesaw.Seesaw(i2c, addr=0x36)
encoder0 = rotaryio.IncrementalEncoder(enc0)
last_position0 = None

# encoder button
enc0.pin_mode(24, enc0.INPUT_PULLUP)
enc_button = seesaw_io.DigitalIO(enc0, 24)
enc_button_state = False

# encoder neopixel
pixel0 = seesaw_neopixel.NeoPixel(enc0, 6, 1)
pixel0.brightness = 0.2
pixel0.fill(BLUE)

# variables
volume = 0 # volume
pixel_level = 25 # pixel brightness for rings
last_pos0 = 0 # position of the encoder
ctrl_mode = 0 # control mode set with encoder

while True:
    # run neopixel animation
    pulse.animate()
    # get encoder position
    pos0 = -encoder0.position

    # if the encoder button is pressed..
    if not enc_button.value and not enc_button_state:
        enc_button_state = True
        # switch between control modes
        if ctrl_mode == 0:
            ctrl_mode = 1
        else:
            ctrl_mode = 0
        print("ctrl_mode is %d" % ctrl_mode)
    # button debounce
    if enc_button.value and enc_button_state:
        enc_button_state = False
    # if control mode is 0..
    # control the volume of the white noise
    if ctrl_mode == 0:
        # encoder neopixel is blue
        pixel0.fill(BLUE)
        # if the encoder moves..
        if pos0 != last_pos0:
            ...
# if you increase the encoder
# increase value by 0.1
# maxed out at 1
if pos0 > last_pos0:
    volume = volume + 0.1
    if volume > 1:
        volume = 1
# if you decrease
# decrease value by 0.1
# minimum value of 0
if pos0 < last_pos0:
    volume = volume - 0.1
    if volume < 0:
        volume = 0
print(volume)
# reset the position
last_pos0 = pos0

# if control mode is 1..
# control the brightness of the neopixel rings
# actually controlling the % of red, not brightness directly
if ctrl_mode == 1:
    # set the encoder neopixel to red
    pixel0.fill(RED)
    # if you increase the encoder
    # increase value by 10
    # max out at 255
    if pos0 != last_pos0:
        if pos0 > last_pos0:
            pixel_level = pixel_level + 10
            if pixel_level > 255:
                pixel_level = 255
        # if you decrease the encoder
        # decrease value by 10
        # minimum level of 25
        if pos0 < last_pos0:
            pixel_level = pixel_level - 10
            if pixel_level < 25:
                pixel_level = 25
        print(pixel_level)
    # reset the position
    last_pos0 = pos0
# set the neopixel ring color
pulse.color = (pixel_level, 0, 0)
# set the audio volume
mixer.voice[0].level = volume

Upload the Code and Libraries to the Feather RP2040

After downloading the Project Bundle, plug your Feather RP2040 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 Feather RP2040's CIRCUITPY drive.

- lib folder
- code.py
- Enceladus_Hiss_NASA.mp3
Your Feather RP2040 CIRCUITPY drive should look like this after copying the lib folder, Enceladus_Hiss_NASA.mp3 file and the code.py file.

![CIRCUITPY folder contents]

### About the Enceladus Hiss mp3 File

The .mp3 file used for the white noise is a recording from NASA of Saturn's radio emissions:

Saturn is a source of intense radio emissions, which were monitored by the Cassini spacecraft. The radio waves are closely related to the auroras near the poles of the planet. These auroras are similar to Earth's northern and southern lights.

This recording is available from NASA's Soundcloud site and was featured in their Spooky Space Sounds collection. It is distributed with a CC BY-NC 3.0 license.

### How the CircuitPython Code Works

Pins **D10** and **D11** and set to **True** to enable hardware on the PropMaker FeatherWing to work properly. **D10** enables the speaker pin and **D11** turns on the R pin for the RGB LED, which is used by the fan since it has a 5V output.

```python
# enable propmaker speaker output
enable = digitalio.DigitalInOut(board.D10)
enable.direction = digitalio.Direction.OUTPUT
enable.value = True

# enable fan pin
fan_pin = digitalio.DigitalInOut(board.D11)
fan_pin.direction = digitalio.Direction.OUTPUT
fan_pin.value = True
```
The Mixer

An `audiomixer.Mixer()` object is instantiated to allow for volume control of the audio file in software. The value of `mixer.voice[0].level` can be a float from 0.0 to 1.0 and later in the code will be set by the rotary encoder.

```
# speaker pin on the propmaker
audio = AudioOut(board.A0)
# create mixer instance
mixer = audiomixer.Mixer(voice_count=1, sample_rate=22050, channel_count=1, bits_per_sample=16, samples_signed=True)
# attach mixer to audio playback
audio.play(mixer)

# open mp3 audio file
audio_file = audiomp3.MP3Decoder(open("Enceladus_Hiss_NASA.mp3","rb"))
# play audio file in first channel of mixer
mixer.voice[0].play(audio_file, loop=True)
# set mixer channel level
mixer.voice[0].level = 0
```

NeoPixels

The NeoPixel pin on the PropMaker FeatherWing (D5) is instantiated as a `NeoPixel()` object. The `Pulse()` animation is used from the LED Animations library.

```
# propmaker neopixel pin
pixel_pin = board.D5
num_pixels = 35
pixels = neopixel.NeoPixel(pixel_pin, num_pixels, brightness=0.3, auto_write=False)

# define neopixel colors
RED = (255, 0, 0)
BLUE = (0, 0, 255)

# pulse animation
pulse = Pulse(pixels, speed=0.1, color=RED, period=3)
```

Rotary Encoder

The rotary encoder and its NeoPixel are setup over I2C. Since `digitalio` and the standard NeoPixel library are being used alongside the `adafruit_seesaw` library, the `adafruit_seesaw digitalio` and `neopixel` portions of the library are imported as `seesaw_io` and `seesaw_neopixel` to avoid conflicts.

```
# i2c setup for rp2040 feather stemma port
i2c = board.STEMMA_I2C()

# rotary encoder
enc0 = seesaw.Seesaw(i2c, addr=0x36)
encoder0 = rotaryio.IncrementalEncoder(enc0)
```
last_position0 = None

# encoder button
c0.pin_mode(24, c0.INPUT_PULLUP)
cenc = seesaw_io.DigitalIO(c0, 24)
cenc_button_state = False

# encoder neopixel
pixel0 = seesaw_neopixel.NeoPixel(c0, 6, 1)
pixel0.brightness = 0.2
pixel0.fill(BLUE)

The Loop

In the loop, the Pulse animation is shown on the NeoPixels and the encoder button switches between `ctrl_mode 0` and `1`.

```python
# run neopixel animation
pulse.animate()
# get encoder position
pos0 = -encoder0.position

# if the encoder button is pressed..
if not cenc_button.value and not cenc_button_state:
cenc_button_state = True

# switch between control modes
if ctrl_mode == 0:
    ctrl_mode = 1
else:
    ctrl_mode = 0
```

Turn Up the Volume

If `ctrl_mode` is `0`, then the volume of the audio file is affected by twisting the rotary encoder knob. Each position move either increases or decreases the audio level by `0.1`.

```python
# if control mode is 0..
# control the volume of the white noise
if ctrl_mode == 0:
    # encoder neopixel is blue
    pixel0.fill(BLUE)

    # if the encoder moves..
    if pos0 != last_pos0:
        # if you increase the encoder
        # increase value by 0.1
        # maxed out at 1
        if pos0 > last_pos0:
            volume = volume + 0.1
            if volume > 1:
                volume = 1

        # if you decrease
        # decrease value by 0.1
        # minimum value of 0
        if pos0 < last_pos0:
            volume = volume - 0.1
            if volume < 0:
                volume = 0
```
Turn Up the Brightness

If `ctrl_mode` is 1, then the "brightness" of the NeoPixels are affected. The value affected by the rotary encoder is actually the red value of the RGB color value. Each position change by the rotary encoder increases or decreases the red value by 25.

```python
# if control mode is 1..
    # control the brightness of the neopixel rings
    # actually controlling the % of red, not brightness directly
    if ctrl_mode == 1:
        # set the encoder neopixel to red
        pixel0.fill(RED)
        # if you increase the encoder
        # increase value by 10
        # max out at 255
        if pos0 != last_pos0:
            if pos0 > last_pos0:
                pixel_level = pixel_level + 10
                if pixel_level > 255:
                    pixel_level = 255
            # if you decrease the encoder
            # decrease value by 10
            # minimum level of 25
            if pos0 < last_pos0:
                pixel_level = pixel_level - 10
                if pixel_level < 25:
                    pixel_level = 25
        print(pixel_level)
        # reset the position
        last_pos0 = pos0
```

At the bottom of the loop, the NeoPixel color value is set, followed by the audio mixer level value.

```python
# set the neopixel ring color
    pulse.color = (pixel_level, 0, 0)
# set the audio volume
    mixer.voice[0].level = volume
```
3D Printing

Parts List

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. Original design source may be downloaded using the links below.

Edit Design

Download STLS

Slice with settings for PLA material

The parts were sliced using CURA using the slice settings below.

PLA filament 220c extruder
0.2 layer height
10% gyroid infill
60mm/s print speed
60c heated bed

Supports
Support Extrusion Width: .2
Support Density: 4%
Support Overhang Angle: 50
Support Z Height: .21
Interface: On
Support Roof: On
Support Pattern: Zig Zag

Build Plate Adhesion
Type: brim
Line Count: 6
Assemble

Solder Headers

Align socket header pins to the Prop-Maker wing. Use a breadboard to help solder the short side of the headers to the Prop-Maker.

Solder socket headers to the top side of the Feather RP2040 board. Then, solder pin headers to the bottom side of the PropMaker FeatherWing so it can be plugged into the Feather.

Solder extension wires

Solder extension wires for the fan and power switch to the Prop-Maker. The fan plugs into a Molex PicoBlade socket extension cable.

LEDs and speakers wires plug into the ports on the board.

NeoPixel LEDs plug in via the 3 Pin JST cable on the Prop-Maker.
Speaker mount

Speaker press fits into the printed mount, facing up. Use M2.5x6mm long screws to attach to the top of the Prop-Maker board.

Coil wires

Coil wires for the fan and power switch under the Prop-Maker board.
Fan mount

Sandwich the fan between the printed mount and the Feather mount. Use the included fan screws and nuts to secure the mounts.
Feather mount

Attach the Feather board to the fan mount with M2.5x6mm long screws.

Join the Prop-Maker headers to the Feather board socket headers.
LED rings

Solder wires to the back side of the LED rings.

Lay the 16 NeoPixel ring facing down on the printed mount.

Lay the 24 NeoPixel ring facing up and solder wires from the 16 ring to 24 ring.
LED Ring JST

Solder a 3 pin JST PH cable to the 12 ring LED, this will connect to the port on the Prop-Maker.

Mount fan

Use M2x12mm long screws to secure the LED ring mount to the Prop-Maker / Feather sandwich.

Connect wires

Plug in the wires for the LEDs, fan, switch and battery extension cable.

Thread wires

Pass all wires through the USB port opening on the Rocket body.
Align tabs

Pass the Prop-Maker circuit from the bottom (larger) end of the Rocket body. Gently press on the one side of the Rocket body to fit the mounting tabs inside the body.

Screw tabs

Tabs are aligned and secured with M2x6mm long screws
DIY USB cables

Use 30mm long DIY USB ribbon cable with a socket USB A on one side and USB C plug on the other side.
Mount pipe

Press wires into the mounting pipe. Use M2x6mm long screws to secure the Rocket body.

Attach Lid

Align the Pipe lid and press fit to the pipe opening.
Thread wires

Thread wires into the center opening on the case. Use M2x6mm screws to secure the Pipe to the case.
Rotary assemble

Align pins on the rotary encoder and solder to the board.

Mount rotary

 Identify the LED on the Rotary board and align it to the slit opening on the case. Press the Rotary Stem through the opening on the case.

Fasten with the included nut and attach the cap to the stem.
Power switch

Solder spade connectors to a 2 Pin JST socket to easily connect the toggle switch to an extension cable.

Mount switch

Press fit the stem into the case. Use the include nut to secure the switch cover to the case.
USB adapter

Attach a USB A to USB C adapter to the DIY USB socket.

The adapter press fits to the port opening on the case.

Use a magnetic USB C tip to easily attach a charging cable.

Connect an optional LiPo battery to the JST extension cable.

Case Lid

Align the opening on the lid to the USB port on the case. Press fit to secure.

Flame assembly

Trace and cut out a flame design out of tissue paper.

Use double stick tape to adhere the ends to the bottom of the ring mount.
Complete

Attach the top Rocket Cap and blast off!