# Table of Contents

**Overview**  
- 3D Printed Darksaber  
- PropMaker Wing  
- Bluetooth NeoPixels LEDs  
- Prerequisite Guides  
- Parts  
- Lightsaber Kit

**Circuit Diagram**  
- Adafruit Library for Fritzing  
- Wired Connections  
- Powering

**3D Printing**  
- CAD Files  
- Slicing Parts  
- Slice Settings  
- Filament for the Blade  
- Hilt Assembly  
- Blade Assembly (2-Part)  
- Parts for All Printers  
- Parts for Small Printers  
- Parts for Big Printers  
- Parts for Giant Printers  
- Parts for Stand

**CircuitPython on Feather M4 Express**  
- Set up CircuitPython Quick Start!  
- Further Information

**CircuitPython for Feather nRF52840**  
- Set up CircuitPython Quick Start!

**CircuitPython Code Setup**  
- Install Libraries  
- BLE Libraries  
- Download the Feather nRF52840 Code from GitHub  
- Download Feather M4 Code from GitHub  
- Sound Effects

**Feather Headers**  
- Feather Headers  
- PropMaker FeatherWing  
- Installing Headers for Bluefruit  
- Solder Bluefruit Headers  
- Installed Headers

**Wiring PropMaker**  
- PicoBalde Cables for PropMaker  
- Solder 2-wire PicoBlade Cable  
- Solder 4-Wire PicoBlade Cable
• Cables for PropMaker
• Solder 4-Wire Cable to PropMaker
• Solder 2-Wire Cable to PropMaker
• Soldered Cables
• Install Feather to PropMaker

Wiring Button
• Cables for RGB LED Button
• Solder 4-Wire PicoBlade Cable
• Solder 2-Wire PicoBlade Cable
• Button Cables
• RGB LED Button Pinouts
• Solder 2-Wire Cable to Button
• Solder 4-Wire Cable to Button
• Wired Button
• Test RGB LED Button

Wiring Speaker
• Speaker Extension Cable
• Solder 2-pin Cable
• Wired Speaker

NeoPixel Strip Assembly
• Reel of NeoPixel LEDs
• Measure and Cut Strip
• Strip of Insulation
• Join Strips
• Double-sided Strips
• Back-to-Back Double-sided Strips
• Assembled NeoPixel Strips

Wiring the NeoPixel Strips
• Wires & Cables for NeoPixel Strip
• Solder Cables to Strips
• Insulate Exposed Pads
• Solder Wires to 3-Pin JST Cable
• Insulate Exposed Wires
• Wired NeoPixel Strips
• Test NeoPixel Strips

Blade Assembly
• Hardware for Blade Parts
• Install Nuts to Blade
• Secure Blade Parts
• Install Nut in Tang
• Install Strip to Blade
• Installed NeoPixel Strip
• Test Blade with NeoPixel Strip
• Installing Emitter
• Install Blade to Emitter
• Installed Blade
• Secure Emitter to Blade
• Installing Covers to Blade
• Join with vinyl or Tape
• Attaching Covers to Blade
• Installed Blade Covers
• Hardware for Coupler
• Install Lock Nuts into Coupler
• Continue Lock Nut Installation
• Install Guard
• Installing Couple Cabling
• Installing Coupler
• Install Coupler to Emitter
• Secure Coupler to Emitter
• Assembled Blade, Emitter, Guard and Coupler
• Test Blade Covers with NeoPixels

**Button Assembly** 64

• Button Hardware
• Install Cables in Hilt
• Install Hex Nut and O-Ring
• Thread Cables in Hilt
• Secured Button to Hilt
• Secured Button
• Hilt Check

**Feather Assembly** 66

• Hardware for Feathers
• Install Screws and Nuts to Feather
• Installing Feather to PCB Holder
• Connect Speaker to PropMaker
• Connect Button to PropMaker
• Connect Battery to Feather
• Test Circuit

**Hilt Assembly** 69

• Installing Blade to Hilt
• Install Strip Cable to Hilt
• NeoPixel Cable Pass Thru Hilt
• Installing Hilt to Coupler
• Secure Hilt to Coupler
• Connect NeoPixel Strip to PropMaker FeatherWing
• Install Speaker into Holder
• Install Battery into Holder
• Install Feather into PCB holder
• Secure and Snip
• Inside the PCB Holder
• Install Speaker to Hilt
• Install Battery to Hilt
• Install PCB Holder to Hilt
• Access to Feather USB Port
• Install Pommeel
• Assembled Hilt

**Usage** 80

• Bluetooth Control
• Using Bluefruit LE Connect App
• Color Picker
• Apple Watch
Overview

3D Printed Darksaber

This is a fully 3D printed prop that we designed to a DIY kit that you can build yourself with electronics from Adafruit. It’s built to be durable so you can throw some heavy swings for those intense photo shoots. All of the electronics are hidden inside the hilt and it features motion activated lights and sound.

PropMaker Wing

For the electronics, the Prop-Maker FeatherWing is designed for making props with lights and sounds. It’s great for portable projects that need motion sensing, buttons and powering lots of NeoPixel LEDs.
Bluetooth NeoPixels LEDs
Combined with the Feather nRF52840 Bluefruit you can use the Bluefruit Connect app to change the color of the NeoPixel LEDs. This is great for cosplay props with need to be remotely controlled.

Prerequisite Guides
Take a moment to walk through the following guides.

- Introducing Feather nRF52840
- Introducing Feather M4 Express
- Prop-Maker FeatherWing
- Welcome to CircuitPython
- Circuit Python Essentials

Parts
Adafruit Feather nRF52840 Express
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 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/3988

Adafruit Mini Skinny NeoPixel Digital RGB LED Strip - 60 LED/m
So thin. So mini. So teeeeeeny-tiny. It's the 'skinny' version of our classic NeoPixel strips! These NeoPixel strips have 60 digitally-addressable pixel Mini LEDs per...
https://www.adafruit.com/product/2959

Lithium Ion Cylindrical Battery - 3.7v 2200mAh
Need a big battery for your project? This lithium-ion battery contains a 2200mAh and a protection circuit that provides over-voltage, under-voltage, and over-current protection. Yet,...
https://www.adafruit.com/product/1781

Rugged Metal Pushbutton with Black Finish - 16mm 6V RGB Latching
By popular demand, we now have latching rugged metal push buttons with a full color RGB LED ring light! These chrome-plated metal buttons are rugged,...
https://www.adafruit.com/product/4660
1 x **3-pin JST Cable**
STEMMA JST PH 3-Pin to Female Socket Cable - 200mm

1 x **Silicone Ribbon Cable**
Silicone Cover Stranded-Core Ribbon Cable - 10 Wire 1 Meter Long - 28AWG Black
[https://www.adafruit.com/product/3890](https://www.adafruit.com/product/3890)

1 x **4-pin pico Cable**
.25mm Pitch 4-pin Cable Matching Pair - 40cm long - Molex PicoBlade Compatible
[https://www.adafruit.com/product/4722](https://www.adafruit.com/product/4722)

1 x **2-Pin picoBlade Cable**
1.25mm Pitch 2-pin Cable Matching Pair - 40cm long - Molex PicoBlade Compatible
[https://www.adafruit.com/product/4720](https://www.adafruit.com/product/4720)

1 x **Short Feather Headers**
Short Headers Kit for Feather - 12-pin + 16-pin Female Headers
[https://www.adafruit.com/product/2940](https://www.adafruit.com/product/2940)

1 x **Heat Shrink Tubing**
Pre-Cut Multi-Colored Heat Shrink Pack Kit - 280 pcs
[https://www.adafruit.com/product/4559](https://www.adafruit.com/product/4559)

**Tools**

1 x **Stick Vise**
PCB stick vise
[https://www.adafruit.com/product/3197](https://www.adafruit.com/product/3197)

1 x **Pana Vise Jr**
Pana Vise Jr
[https://www.adafruit.com/product/151](https://www.adafruit.com/product/151)

1 x **Helping Third Hands**
Helping Third Hands

---

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](https://www.adafruit.com/product/3923)
Lightsaber Kit

The **lightsaber kit** does not come with the **Feather nRF52840 Bluefruit** , but instead the **Feather M4 Express** .

The code for the Darksaber is written to work with both Feathers. If you'd like to substitute the Feather nRF52840 Bluefruit for the Feather M4 Express, use the **lightsaber code** .

The speaker in the lightsaber kit will not fit the Darksaber hilt (**mini oval speaker** is used).

The RGB LED button in the Lightsaber kit is a momentary switch. The Darksaber uses a **latching RGB LED button** instead.

Here's a list of parts from the Lightsaber kit that will work with the Darksaber build.

- PropMaker FeatherWing
- 2200mAh battery
- Headers
- Cables
- Heatshrink

Adafruit Prop-Maker Starter Kit - Make your own Lightsaber!
Szoom woom woom! Use the force and Adafruit Feather to build a battle-grade lightsaber! If you can't make the hyperspeed jump to Ilum for your Jedi...
https://www.adafruit.com/product/4860

Circuit Diagram

The diagram below provides a visual reference for wiring of the components. This diagram was created using the software package [Fritzing](https://fritzing.org).
Adafruit Library for Fritzing

Use Adafruit's Fritzing parts library to create circuit diagrams for your projects. Download the library or just grab individual parts. Get the library and parts from [GitHub - Adafruit Fritzing Parts](https://github.com/adafruit/Fritzing-Parts).

Wired Connections

NeoPixels to PropMaker FeatherWing

- DIN from NeoPixel strip to NeoPixel Port on PropMaker
- 5V from NeoPixel strip to NeoPixel Port on PropMaker
- GND from NeoPixel strip to NeoPixel Port on PropMaker

RGB LED Button to PropMaker

- Common LED (pin c) to V+ pin on PropMaker
- Red LED (pin a) to Red pin on PropMaker
- Green LED (pin b) to Green pin on PropMaker
- Blue LED (pin d) to Blue pin on PropMaker
- Common pin to G pin on PropMaker
- NC (normally closed) pin to EN pin on PropMaker

The speaker and battery plug directly into the ports on the PropMaker.

Powering

In this project, a 3.7V 2200mAh battery is used. The battery can be recharged over the USB port on the Feather.
3D Printing

CAD Files

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.

- CAD-source.zip
- STLs.zip

Slicing Parts

No supports are required. Slice with setting for PLA material.

Place the blade parts diagonally across the bed of the printer.

Rotating parts at 45 degrees allows long narrow parts to fit.
Slice Settings

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

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

Filament for the Blade

A translucent white colored filament works best for the blade parts. Often called "Natural Clear" this filament features a white pigment but slightly transparent. This diffuses the NeoPixel LEDs quiet nicey.

1 x Natural Clear PLA 1.75mm
Purchase Natural Clear Filament from Amazon

https://www.amazon.com/gp/product/B00MB3CV6K

Hilt Assembly

The coupler is fitted into the hand of the hilt. The guard is fitted into the coupler. The emitter is fitted over the coupler and keeps the guard secured in place. Screws secure the hilt and emitter to the coupler.

Blade Assembly (2-Part)

Hex nuts are inserted into the cavities in the blade parts. M3 screws secure the two blade parts together. Pockets allow the screws to be fastened together.
Parts for All Printers

These parts are designed to fit printers with small or large build volumes.

- coupler
- button-oring (For TPU filament)
- emitter
- pommel
- speaker-holder
- pcb-holder
- battery-holder
- guard

Parts for Small Printers

These parts are specifically designed for printers with smaller build volumes (150x150x140). The blade and hilt sections do not feature connection tabs and will need to be glued together with a strong adhesive.

- blade-cover-1
- blade-cover-2
- blade-cover-3
- blade-cover-4
- blade-base-1
- blade-mid-2
- blade-upper-3
- blade-tip-4
- hilt-top
- hilt-bottom

Parts for Big Printers

These parts are specifically designed for printer with large build volumes (300x300x400)

- hilt
- blade-tip
- blade-base
- blade-cover-tip
Parts for Giant Printers

The blade and covers can be fully printed in one piece with printers that feature extra large build volumes.

Full size blade: 670mm (26.4in) long.

- blade-full
  - blade-cover-full

Parts for Stand

These parts are specifically for building the stand for the prop.

- stand
  - stand-top-holder
  - stand-bottom-holder (Fits the Pommel)
  - stand-bottom-charger (Fits without Pommel - USB Charging mode)

CircuitPython on Feather M4 Express

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.

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 :)

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

Click the link above and download the latest UF2 file.

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

Plug your Feather M4 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. 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 FEATHERBOOT.

Drag the adafruit_circuitpython_etc.uf2 file to FEATHERBOOT.

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

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

Further Information

For more detailed info on installing CircuitPython, check out [Installing CircuitPython ()](#).

CircuitPython for Feather nRF52840

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.

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

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! :)

---

**CircuitPython Code Setup**

Take your Feather and plug it into your computer via a known good data + power USB cable. Have your Feather Bluefruit handy as you'll be performing most of the same steps for each. Your operating system will show a drive named CIRCUITPY when a board is plugged in. If you get a drive named FTHR840BOOT or FEATHERBOOT you'll likely need to install CircuitPython.

**Install Libraries**

You'll need a few CircuitPython libraries in the lib folder on the Feather CIRCUITPY drive for the code to work. Head to [https://circuitpython.org/libraries](https://circuitpython.org/libraries) to download the latest library bundle matching the major version of CircuitPython now on your board (6 for CircuitPython 6.x, etc.).

Once you've downloaded the libraries bundle, add these libraries to the lib folder on the Feather:

- `adafruit_bus_device`
- `adafruit_led_animation`
- `adafruit_lis3dh.mpy`
- `neopixel.mpy`
BLE Libraries

Install the bluetooth libraries for the Feather nRF52840.

- adafruit_ble
- adafruit_bluefruit_connect

Your Feather nRF52840 CIRCUITPY drive should look like this after you load the code below:

![CircuitPython Drive](image)

Your Feather M4 Express CIRCUITPY drive should look like this after you load the code below:

![CircuitPython Drive](image)

Download the Feather nRF52840 Code from GitHub

Once your Feather nRF52840 is all setup with CircuitPython and the necessary libraries, you can click on the Download: Project Zip link above the code to get the files.
Prop-Maker based Darksaber
Adapted from the Prop-Maker based Master Sword code
by Kattni Rembor & Limor Fried
Adafruit invests time and resources providing this open source code.
Please support Adafruit and open source hardware by purchasing
products from Adafruit!
Written by Liz Clark for Adafruit Industries
Copyright (c) 2021 Adafruit Industries
Licensed under the MIT license.
All text above must be included in any redistribution.

```python
import time
import random
import board
from digitalio import DigitalInOut, Direction
import neopixel
import adafruit_lis3dh
from adafruit_led_animation.animation.solid import Solid
from adafruit_led_animation.animation.pulse import Pulse
from adafruit_led_animation.animation.comet import Comet
from adafruit_bluefruit_connect.packet import Packet
from adafruit_bluefruit_connect.color_packet import ColorPacket
from adafruit_ble import BLERadio
from adafruit_ble.advertising.standard import ProvideServicesAdvertisement
from adafruit_ble.services.nordic import UARTService

# BLE setup
ble = BLERadio()
uart_service = UARTService()
advertisement = ProvideServicesAdvertisement(uart_service)

# CUSTOMISE SENSITIVITY HERE: smaller numbers = more sensitive to motion
HIT_THRESHOLD = 250
SWING_THRESHOLD = 150

# Set to the length in seconds of the "on.wav" file
POWER_ON_SOUND_DURATION = 1.7

# NeoPixel setup
NUM_PIXELS = 34  # Number of pixels used in project
NEOPIXEL_PIN = board.D5
POWER_PIN = board.D10

enable = DigitalInOut(POWER_PIN)
enable.direction = Direction.OUTPUT
enable.value = False

strip = neopixel.NeoPixel(NEOPIXEL_PIN, NUM_PIXELS, brightness=.5, auto_write=False)
strip.fill(0)  # NeoPixels off ASAP on startup
strip.show()

# default NeoPixel color is white
COLOR = (255, 255, 255)

# NeoPixel animations
pulse = Pulse(strip, speed=0.05, color=COLOR, period=3)
solid = Solid(strip, color=COLOR)
comet = Comet(strip, speed=0.05, color=COLOR, tail_length=40)

#audio
try:
    from audiocore import WaveFile
except ImportError:
    from audioio import WaveFile
```
try:
    from audioio import AudioOut
except ImportError:
    try:
        from audiopwmio import PWMAudioOut as AudioOut
    except ImportError:
        pass  # not always supported by every board!

audio = AudioOut(board.A0)  # Speaker
wave_file = None

# 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 STEMMA QT connector on a
# microcontroller
accel = adafruit_lis3dh.LIS3DH_I2C(i2c)
accel.range = adafruit_lis3dh.RANGE_4_G

def play_wav(name, loop=False):
    
    Play a WAV file in the 'sounds' directory.
    :param name: partial file name string, complete name will be built around
    this, e.g. passing 'foo' will play file 'sounds/foo.wav'.
    :param loop: if True, sound will repeat indefinitely (until interrupted
    by another sound).
    
    global wave_file  # pylint: disable=global-statement
    print("playing", name)
    if wave_file:
        wave_file.close()
    try:
        wave_file = open('sounds/' + name + '.wav', 'rb')
        wave = WaveFile(wave_file)
        audio.play(wave, loop=loop)
    except OSError:
        pass  # we'll just skip playing then

def power_on(sound, duration):
    
    Animate NeoPixels with accompanying sound effect for power on.
    :param sound: sound name (similar format to play_wav() above)
    :param duration: estimated duration of sound, in seconds (>0.0)
    
    start_time = time.monotonic()  # Save audio start time
    play_wav(sound)
    while True:
        elapsed = time.monotonic() - start_time  # Time spent playing sound
        if elapsed > duration:  # Past sound duration?
            break  # Stop animating
            comet.animate()

    # List of swing wav files without the .wav in the name for use with play_wav()
    swing_sounds = [
        'swing1',
        'swing2',
        'swing3',
        'swing4',
    ]

    # List of hit wav files without the .wav in the name for use with play_wav()
    hit_sounds = [
        'hit1',
        'hit2',
        'hit3',
        'hit4',
    ]
mode = 0  # Initial mode = OFF

#RGB LED
red_led = DigitalInOut(board.D11)
green_led = DigitalInOut(board.D12)
blue_led = DigitalInOut(board.D13)

red_led.direction = Direction.OUTPUT
green_led.direction = Direction.OUTPUT
blue_led.direction = Direction.OUTPUT

blue_led.value = True
red_led.value = True
green_led.value = True

# Darksaber start-up before loop
if mode == 0:  # If currently off...
    enable.value = True
    power_on('on', POWER_ON_SOUND_DURATION)  # Power up!
    play_wav('idle', loop=True)  # Play idle sound now
    mode = 1  # Idle mode

while True:
    # begin advertising BLE
    ble.start_advertising(ADVERTISEMENT)
    # if no BLE connection...
    # allows it to be used without the bluefruit app connection
    while not ble.connected:
        if mode >= 1:  # If not OFF mode...
            x, y, z = accel.acceleration  # Read accelerometer
            accel_total = x ** 2 + z ** 2
            # (Y axis isn't needed, due to the orientation that the Prop-Maker
            # Wing is mounted. Also, square root isn't needed, since we're
            # comparing thresholds...use squared values instead.)
            if accel_total > HIT_THRESHOLD:  # Large acceleration = HIT
                TRIGGER_TIME = time.monotonic()  # Save initial time of hit
                play_wav(random.choice(hit_sounds))  # Start playing 'hit' sound
                solid.animate()  # NeoPixels are solid on with a hit
                mode = 3  # HIT mode
            elif mode == 1 and accel_total > SWING_THRESHOLD:  # Mild = SWING
                TRIGGER_TIME = time.monotonic()  # Save initial time of swing
                play_wav(random.choice(swing_sounds))  # Randomly choose from
                available swing sounds
                while audio.playing:
                    pass  # wait till we're done
                mode = 2  # we'll go back to idle mode
            elif mode == 1:
                # pulse animation when idling or swinging
                pulse.animate()
            elif mode > 1:  # If in SWING or HIT mode...
                if audio.playing:  # And sound currently playing...
                    blend = time.monotonic() - TRIGGER_TIME  # Time since triggered
                    if mode == 2:  # If SWING,
                        blend = abs(0.5 - blend) * 2.0  # ramp up, down
                    else:  # No sound now, but still SWING or HIT modes
                        play_wav('idle', loop=True)  # Resume idle sound
                        mode = 1  # Return to idle mode
            ble.stop_advertising()

        # if BLE is connected...
        while ble.connected:
            # color picker from bluefruit app
            if uart_service.in_waiting:
                packet = Packet.from_stream(uart_service)
                # if a color packet is received...
                if isinstance(packet, ColorPacket):
                    print(packet.color)
                    # color for the different animations are updated
comet.color = packet.color
solid.color = packet.color
pulse.color = packet.color
solid.animate()

# repeat of the above code
if mode >= 1:  # If not OFF mode...
x, y, z = accel.acceleration  # Read accelerometer
accel_total = x * x + z * z
# (Y axis isn't needed, due to the orientation that the Prop-Maker
# Wing is mounted. Also, square root isn't needed, since we're
# comparing thresholds...use squared values instead.)
if accel_total > HIT_THRESHOLD:  # Large acceleration = HIT
    TRIGGER_TIME = time.monotonic()  # Save initial time of hit
    play_wav(random.choice(hit_sounds))  # Start playing 'hit' sound
    # NeoPixels are solid on with a hit
    solid.animate()
    mode = 3  # HIT mode
elif mode == 1 and accel_total > SWING_THRESHOLD:  # Mild = SWING
    TRIGGER_TIME = time.monotonic()  # Save initial time of swing
    play_wav(random.choice(swing_sounds))  # Randomly choose from
available swing sounds
    while audio.playing:
        pass  # wait till we're done
    mode = 2  # we'll go back to idle mode
elif mode == 1:
    # pulse animation when idling or swinging
    pulse.animate()
elif mode > 1:  # If in SWING or HIT mode...
    if audio.playing:  # And sound currently playing...
        blend = time.monotonic() - TRIGGER_TIME  # Time since triggered
        if mode == 2:  # If SWING,
            blend = abs(0.5 - blend) * 2.0  # ramp up, down
        else:  # No sound now, but still SWING or HIT modes
            play_wav('idle', loop=True)  # Resume idle sound
            mode = 1  # Return to idle mode
    elif mode >= 1:  # If not OFF mode...
        x, y, z = accel.acceleration  # Read accelerometer
        accel_total = x * x + z * z
        # (Y axis isn't needed, due to the orientation that the Prop-Maker
        # Wing is mounted. Also, square root isn't needed, since we're
        # comparing thresholds...use squared values instead.)
        if accel_total > HIT_THRESHOLD:  # Large acceleration = HIT
            TRIGGER_TIME = time.monotonic()  # Save initial time of hit
            play_wav(random.choice(hit_sounds))  # Start playing 'hit' sound
            # NeoPixels are solid on with a hit
            solid.animate()
            mode = 3  # HIT mode
        elif mode == 1 and accel_total > SWING_THRESHOLD:  # Mild = SWING
            TRIGGER_TIME = time.monotonic()  # Save initial time of swing
            play_wav(random.choice(swing_sounds))  # Randomly choose from
available swing sounds
            while audio.playing:
                pass  # wait till we're done
            mode = 2  # we'll go back to idle mode
        elif mode == 1:
            # pulse animation when idling or swinging
            pulse.animate()
        elif mode > 1:  # If in SWING or HIT mode...
            if audio.playing:  # And sound currently playing...
                blend = time.monotonic() - TRIGGER_TIME  # Time since triggered
                if mode == 2:  # If SWING,
                    blend = abs(0.5 - blend) * 2.0  # ramp up, down
                else:  # No sound now, but still SWING or HIT modes
                    play_wav('idle', loop=True)  # Resume idle sound
                    mode = 1  # Return to idle mode

Download Feather M4 Code from GitHub

Once your Feather M4 Express is all setup with CircuitPython and the necessary libraries, you can click on the Download: Project Zip link above in the code to get the file.

# SPDX-FileCopyrightText: Kattni Rembor for Adafruit Industries
# SPDX-FileCopyrightText: Limor Fried for Adafruit Industries
# SPDX-FileCopyrightText: 2019 Liz Clark for Adafruit Industries
#
# SPDX-License-Identifier: MIT

###

Prop-Maker based Darksaber
Adapted from the Prop-Maker based Master Sword code
by Kattni Rembor & Limor Fried

Adafruit invests time and resources providing this open source code. Please support Adafruit and open source hardware by purchasing products from Adafruit!
Written by Liz Clark for Adafruit Industries
Copyright (c) 2021 Adafruit Industries
Licensed under the MIT license.
All text above must be included in any redistribution.
###

import time
import random
import board
from digitalio import DigitalInOut, Direction
import neopixel
import adafruit_lis3dh
from adafruit_led_animation.animation.solid import Solid
from adafruit_led_animation.animation.pulse import Pulse
from adafruit_led_animation.animation.comet import Comet

# CUSTOMISE SENSITIVITY HERE: smaller numbers = more sensitive to motion
HIT_THRESHOLD = 250
SWING_THRESHOLD = 150

# Set to the length in seconds of the "on.wav" file
POWER_ON_SOUND_DURATION = 1.7

# NeoPixel setup
NUM_PIXELS = 34  # Number of pixels used in project
NEOPIXEL_PIN = board.D5
POWER_PIN = board.D10

enable = DigitalInOut(POWER_PIN)
enable.direction = Direction.OUTPUT
enable.value = False

strip = neopixel.NeoPixel(NEOPIXEL_PIN, NUM_PIXELS, brightness=.5, auto_write=False)
strip.fill(0)  # NeoPixels off ASAP on startup
strip.show()

# default NeoPixel color is white
COLOR = (255, 255, 255)

# NeoPixel animations
pulse = Pulse(strip, speed=0.05, color=COLOR, period=3)
solid = Solid(strip, color=COLOR)
comet = Comet(strip, speed=0.05, color=COLOR, tail_length=40)

# audio
try:
    from audiocore import WaveFile
except ImportError:
    from audioio import WaveFile
try:
    from audiioio import AudioOut
except ImportError:
    try:
        from audiopwmio import PWMAudioOut as AudioOut
    except ImportError:
        pass  # not always supported by every board!

audio = AudioOut(board.A0)  # Speaker
wave_file = None

# 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 STEMMA QT connector on a
# microcontroller
accel = adafruit_lis3dh.LIS3DH_I2C(i2c)
accel.range = adafruit_lis3dh.RANGE_4_G

def play_wav(name, loop=False):
    """Play a WAV file in the 'sounds' directory.
    :param name: partial file name string, complete name will be built around
                 this, e.g. passing 'foo' will play file 'sounds/foo.wav'.
    :param loop: if True, sound will repeat indefinitely (until interrupted
                 by another sound).
    """

©Adafruit Industries
global wave_file  # pylint: disable=global-statement
print("playing", name)
if wave_file:
    wave_file.close()
try:
    wave_file = open('sounds/' + name + '.wav', 'rb')
    wave = WaveFile(wave_file)
    audio.play(wave, loop=loop)
except OSError:
    pass  # we'll just skip playing then

def power_on(sound, duration):
    """
    Animate NeoPixels with accompanying sound effect for power on.
    :param sound: sound name (similar format to play_wav() above)
    :param duration: estimated duration of sound, in seconds (>0.0)
    """
    start_time = time.monotonic()  # Save audio start time
    play_wav(sound)
    while True:
        elapsed = time.monotonic() - start_time  # Time spent playing sound
        if elapsed > duration:  # Past sound duration?
            break  # Stop animating
        comet.animate()

# List of swing wav files without the .wav in the name for use with play_wav()
swing_sounds = [
    'swing1',
    'swing2',
    'swing3',
    'swing4',
]

# List of hit wav files without the .wav in the name for use with play_wav()
hit_sounds = [
    'hit1',
    'hit2',
    'hit3',
    'hit4',
]

mode = 0  # Initial mode = OFF

# RGB LED
red_led = DigitalInOut(board.D11)
green_led = DigitalInOut(board.D12)
blue_led = DigitalInOut(board.D13)

red_led.direction = Direction.OUTPUT
green_led.direction = Direction.OUTPUT
blue_led.direction = Direction.OUTPUT

blue_led.value = True
red_led.value = True
green_led.value = True

# Main loop
while True:
    if mode == 0:  # If currently off...
        enable.value = True
        power_on('on', POWER_ON_SOUND_DURATION)  # Power up!
        play_wav('idle', loop=True)  # Play idle sound now
        mode = 1  # Idle mode
    elif mode >= 1:  # If not OFF mode...
        x, y, z = accel.acceleration  # Read accelerometer
        accel_total = x * x + z * z
# (Y axis isn't needed, due to the orientation that the Prop-Maker # Wing is mounted. Also, square root isn't needed, since we're # comparing thresholds...use squared values instead.)
if accel_total > HIT_THRESHOLD:  # Large acceleration = HIT
    TRIGGER_TIME = time.monotonic()  # Save initial time of hit
    play_wav(random.choice(hit_sounds))  # Start playing 'hit' sound
    solid.animate()
    mode = 3  # HIT mode
elif mode == 1 and accel_total > SWING_THRESHOLD:  # Mild = SWING
    TRIGGER_TIME = time.monotonic()  # Save initial time of swing
    play_wav(random.choice(swing_sounds))  # Randomly choose from available swing sounds
    while audio.playing:
        pass  # wait till we're done
        mode = 2  # we'll go back to idle mode
    else:  # If SWING or HIT mode...
        pulse.animate()
        if audio.playing:  # And sound currently playing...
            blend = time.monotonic() - TRIGGER_TIME  # Time since triggered
            if mode == 2:  # If SWING,
                blend = abs(0.5 - blend) * 2.0  # ramp up, down
            else:  # No sound now, but still SWING or HIT modes
                play_wav('idle', loop=True)  # Resume idle sound
                mode = 1  # Return to idle mode

Sound Effects

The audio files are hosted on the [BLE code github repo](https://github.com/). You can make your own audio files or use the royalty-free ones we've provided. Be sure to create a new folder named "sounds" on to CIRCUITPY drive and drop in the audio files.

Adafruit CircuitPython supports 16-bit, Mono, 22.050kHz .wav audio format. Additionally, the looping sample idle.wav should be a multiple of 512 samples long.
Feather Headers

Use short 12 pin and 16pin female headers for the Feather. Use regular 12pin and 16pin male headers for the PropMaker FeatherWing.

PropMaker FeatherWing

Install the 12pin and 16pin male headers onto the bottom of the PropMaker PCB. Use a breadboard to help keep the headers straight while solder all of the pins.
Installing Headers for Bluefruit

Insert the 12pin and 16pin female headers onto the male headers that were soldered to the PropMaker. Then, place the Feather onto the pins of the female headers. This helps keep the headers on the Feather nice and straight while soldering.

The bottom of the Feather should be facing up when soldering.

Solder Bluefruit Headers

Use a vise to keep the Feather board steady while soldering all of the pins.
Installed Headers
Double check to ensure all of the solder joints are solid. The Feather and PropMaker FeatherWing can be separated by carefully pulling them apart.

Wiring PropMaker

PicoBalde Cables for PropMaker
Use the following wires and cables for the PropMaker FeatherWing.

- 4-Wire Silicone Cable - 6cm
- 4-Pin picoBlade Cable - 3cm
- 2-Wire Silicone Cable - 6cm
- 2-Pin picoBalde Cable - 3cm

The PicoBlade cables are cut short to better fit inside the hilt of the Darksaber.
Solder 2-wire PicoBlade Cable

The 2-wire silicone cable is soldered to the 2-pin picoBlade cable. The 2-pin picoBlade cable is shortened to 3cm in length.

Use pieces of heat shrink tubing to insulate the exposed wire.
Solder 4-Wire PicoBlade Cable
The 4-wire silicone cable is soldered to the 4-pin picoBlade cable. The 4-pin picoBlade cable is shortened to 3cm in length.

Use pieces of heat shrink tubing to insulate the exposed wire.

Helping third hands can help keep wires sturdy while soldering.

Cables for PropMaker
Double check each wire to ensure they’re properly soldered and any exposed wire is insulated with heat shrink tubing.
Solder 4-Wire Cable to PropMaker
Solder the 4-wire cable to the PropMaker. Use the following list of wires and pins for reference.

- Black Wire to V+
- Red Wire to Red Pin
- White Wire to Green Pin
- Yellow Wire to Blue Pin

Solder 2-Wire Cable to ProMaker
Solder the 2-wire cable to the PropMaker. Use the EN and GN pins. Polarity does not matter for these connections.

Soldered Cables
Double check to ensure all of wires has been properly soldered to the pins on the PropMaker FeatherWing.
Install Feather to PropMaker
The PropMaker FeatherWing can be installed back onto the Feather.

Wiring Button

Cables for RGB LED Button
Use the following wires and cables for the RGB LED Button.

- 4-Wire Silicone Cable - 20cm
- 4-Pin picoBlade Cable - 5cm
- 2-Wire Silicone Cable - 20cm
- 2-Pin picoBalde Cable - 5cm

The wires are long in order to cover the length of the hilt.
Solder 4-Wire PicoBlade Cable

The 4-wire silicone cable is soldered to the 4-pin picoBlade cable. The 4-pin picoBlade cable is shortened to 5cm in length.

Use pieces of heat shrink tubing to insulate the exposed wire.
Solder 2-Wire PicoBlade Cable
The 2-wire silicone cable is soldered to the 2-pin picoBlade cable. The 2-pin picoBlade cable is shortened to 3cm in length.

Use pieces of heat shrink tubing to insulate the exposed wire.

Button Cables
Double check each wire to ensure they're properly soldered and any exposed wire is insulated with heat shrink tubing.
RGB LED Button Pinouts
Reference the chart for the pin labels. Read the label markings on the RGB LED button to determine which pins to use.

Solder 2-Wire Cable to Button
Tin the terminals on the button before soldering wires. This helps make soldering the wires to the terminals easier.

Solder the 2-wire cable to the following pins

C (common)
NC (normally closed)

Solder 4-Wire Cable to Button
Attach the 4-wire cable to the following terminals on the button.

Black Wire – Common LED (pin c)
Red Wire – Red LED (pin a)
White Wire – Green LED (pin b)
Yellow Wire – Blue LED (pin d)
Wired Button
Double check the wires are properly soldered to the RGB LED button.

Test RGB LED Button
The RGB LED button can be tested by connecting the 4-wire cable to the 4-wire cable on the PropMaker. Plug in the battery to the Feather to power on the Circuit. The code, libraries and sounds should be uploaded to the Feather board.

Wiring Speaker

Speaker Extension Cable
The stock cable on the mini oval speaker is too short to reach the circuit once inside the hilt. Use a piece of 2-wire silicone cable to make the speaker cable longer.

2-Wire Silicone Cable - 8cm long
Solder 2-pin Cable
Cut the speaker cable in half and use wire stripper to remove a bit of insulation from each wire. Tin the strands of wire by adding a bit of solder. Use helping third hands to keep the wires sturdy while soldering. Use pieces of heat shrink tubing to insulate any exposed connections. Repeat this process for the other send of the cable.

Wired Speaker
Double check the wires are properly soldered.
NeoPixel Strip Assembly

Reel of NeoPixel LEDs
This project uses a two strips of NeoPixels. Remove the strip from the plastic reel. Count and measure each strip. Each strip is cut using scissors. The strips should contain 34 NeoPixel LEDs each. This makes each strip measure to 1.13 meters (44.5in) in length.

Measure and Cut Strip
Use the length of the the blade to determine a proper length of LEDs. The LED strips will be fitted inside the center channel. If you're using a higher density NeoPixel strip, you'll need to measure it in order to fit inside the blade.

Remove the silicone sheathing from the strips once they're measured and cut.
Strip of Insulation

The two NeoPixel strips are placed back-to-back and fitted inside the blade. In order to prevent the two strips from shorting each other out, a strip of insulation is used.

This can be a thin, long strip (or several short strips) of non-conductive material, such as poster board. Another option is to cover up the pads on the strips with kapton tape.

A cutting matt, hobby knife and t-square was used to cut a strip from a 22x28 sheet of clear poster board.
Join Strips
Use double-sided scotch tape to join the strip of insulation to the back of one of the NeoPixel LED strips. This makes joining the two strips together easier as it will keep them together.

Ensure the flow of data is consistent on each strip. Check the strip for an arrow label, make they are pointing in the same direction.

The two strips will share the same DIN, 5V and GND pins.

Double-sided Strips
The thin strip of insulation is sandwiched in between the two NeoPixel LED strips. Additional scotch tape was wrapped over the two LED strips to secure them together. Avoid apply tape over the LEDs and capacitors on the NeoPixel strips.
Back-to-Back Double-sided Strips
Apply additional tape along the strip as needed. Carefully handle the NeoPixel strips when apply taping to avoid any damage.

Assembled NeoPixel Strips
Double check the length of the strip is capable of fitting inside the center of the blade. Ensure the flow of data is the same for both NeoPixel LED strips.
Wiring the NeoPixel Strips

Wires & Cables for NeoPixel Strip
Use the following wires and cables for the NeoPixel strips.

- 2x 3-Wire Silicone Cable – 25cm long
- 2x 3-Pin JST-PH Cable – 17cm long

Solder Cables to Strips
Tin the pads on the first pixel in the NeoPixel strip. Attach the first 3-wire silicone cable to the first pixel in the strip.

Flip the strip over and solder the second cable to the NeoPixel strip.

Leave some space in between the two LED strips to avoid bridging solder.
Insulate Exposed Pads
Add pieces of heat shrink tubing to the insulate the exposed pads and wire. Slip the pieces over the cables and use heat to shrink the tubing. Repeat this process for the second NeoPixel strip.

Solder Wires to 3-Pin JST Cable
Measure, cut and tin the wires on the 3-pin JST cable.

Add pieces of heat shrink to the JST cable before soldering wires.

The wires from the two silicone cables are soldered to the ends of the wires on the 3-pin JST cable.

Use two helping third hands to keep the wires together while soldering.

Solder the two voltage wires from the strips to the red wire on the 3-pin JST cable. Solder the ground wires to the black wire. Then, solder the data wires to the white wire.
Insulate Exposed Wires
Double check the solder joints are solid in each set of wires. Slip the pieces of heat shrink over the exposed connected. Then, apply heat to the tubing to shrink them.

Wired NeoPixel Strips
The double-sided back-to-back NeoPixel strip is now ready to test out.

Test NeoPixel Strips
Connect the 3-pin JST cable from the strip to the NeoPixel port on the PropMaker FeatherWing. Apply power to the Feather by connecting the battery.

The code, libraries and sound files should already uploaded to the Feather board.
Blade Assembly

Hardware for Blade Parts
Use the following hardware to secure the two blade parts together.

- 2x M3 x 10mm pan head screws
- 2x M3 lock nuts with nylon inserts
Install Nuts to Blade
The M3 nuts are installed into cavities inside the lower blade part. The nuts have a very tight fit, so they must be installed by forcing them into place.

Insert an M3 x 10mm screw into the screw hole. Fasten the screw while holding the hex nut in place (use needle nose pliers). Fastening the screw will force the hex nut into the cavity.

Repeat process for the second screw hole. Unfasten the screw from the hex nut when it has been installed.
Secure Blade Parts

Insert the two M3 x 10mm screws into the screw holes in the second blade part. Join the two blade pieces together. Fasten the screws into the hex nuts until fully tightened.
Install Nut in Tang
An M3 lock nut is installed in to the screw hole near the tang of the blade. The installation process is similar to joining the two blade parts together. Use flat needle nose pliers to hold the nut in place while fastening an M3 screw. Fasten the screw until the hex nut is fully pressed into the cavity. Remove the screw when complete.
Install Strip to Blade

Alright, it's time to install the double-sided back-to-back mini skinny strip into the blade.

Carefully fit the NeoPixel strips into the slot in the center of the blade. Press the cabling into the tang.
Installed NeoPixel Strip
Position the strips so it's reaching the very tip of the blade.

The NeoPixel strips are bent slightly just before reaching the tang.

The silicone cables are routed through the hole in the bottom of the tang.

Test Blade with NeoPixel Strip
With the NeoPixel strips now fitted inside the blade, they can be tested with the PropMaker FeatherWing. Plug in the battery to the Feather to power on the circuit.
Installing Emitter
Insert the 3-pin JST cable from the strip through the top of the emitter part. Pull the cable all the way through the emitter and out the other end.

Reference the photo for installing the cable in the emitter in the correct orientation.

Install Blade to Emitter
Begin to fit the tang of the blade into the tip of the emitter. Orient the blade so the shape fits into the opening in the emitter.
Installed Blade
The blade should slide into the emitter with ease. Ensure the wiring is not being kinked when installing the blade into the emitter.

Secure Emitter to Blade
Use an M3x16mm long screw to secure the emitter to the blade. The screw should be long enough to reach the lock nut that was installed into the tang of the blade.

Tightly fasten the M3 screw to secure the emitter to the blade.
Installing Covers to Blade
The 4 cover pieces are attached to both sides of the blade. A piece of dark tape (vinyl with adhesive backing or electrical tape) is used to join the covers together. This also prevents any light leaking from the seam.
Join with vinyl or Tape

Join the set of covers together. Note the chamfer near the edge where the two parts meet. This helps join the two parts together.

Carefully stick the piece of vinyl or electrical tape to the mating cover pieces. Use a hobby knife to trim away any access.
Attaching Covers to Blade
Double-sided nitto tape is used to attach the covers to the blade. Optionally use glues or other adhesives.

Using double-sided tape is nice because it allows the covers to be removed in case the LEDs ever need to be serviced.
Installed Blade Covers
Carefully place the cover onto the blade making sure the parts are lined up nicely. Start to lay the cover down near the tang and work towards the tip.

Apply a good amount of pressure to the covers to make the tape adhere to the surfaces.

Hardware for Coupler
The coupler is used to secure the blade emitter to the hilt. Use the following hardware to install the coupler.

- 2x M3 x 10mm pan head screws
- 2x M3 lock nuts with nylon inserts
Install Lock Nuts into Coupler
The two lock nuts are installed into cavities inside the coupler. Insert an M3 screw into one of the two holes. Use twisters to help fasten the hex nuts onto the thread of the screw. Fasten the screw until the lock nut is fully seated inside the cavity.
Continue Lock Nut Installation
Repeat this process for the second lock nut and screw hole.

An M3 x 16mm long screw is used in order to pass through the coupler.

The installation process is similar to the blade assembly.

Install Guard
The guard is installed onto the coupler. Orient the guard with the coupler and slip the guard over the coupler. Push the guard down so it sits flush with the coupler.
Installing Couple Cabling
Begin to insert the 3-pin JST cable from the NeoPixel strips through the coupler. Reference the photo for the correct orientation.

Installing Coupler
Pull the 3-pin JST cable all the way through the coupler.

Install Coupler to Emitter
Insert the coupler into the bottom of the emitter. Ensure the wires are not being kinked when fitting the coupler. Push the coupler into the emitter until it’s fully seated.
Secure Coupler to Emitter
Use an M3x10mm long screw to secure the emitter to the coupler.

Fully tighten the screw to secure the emitter to the coupler.

Assembled Blade, Emitter, Guard and Coupler
Double check the blade assembly is complete.

Test Blade Covers with NeoPixels
With blade now assembled, it's ready for another round of testing!

Plug in the cable from the strip into the NeoPixel port on the PropMaker FeatherWing. Then connect a battery to the Feather to power on the circuit.
Button Assembly

Button Hardware
Use the hex nuts that came with the RGB LED button. An additional o-ring is 3d printed in flexible filament. The o-ring helps to secure the hex nut onto the threading of the button. Without it, the hex nut cannot be fully seated into the hilt to panel mount the button.

Install Cables in Hilt
Start by insert the two cables from the button through the hole for the button in the hilt.

Pull the two cables through the top opening of the hilt.

Install Hex Nut and O-Ring
First, insert the two cables through the o-ring. Then, slip the hex nut over.

Double check the order is correct. First the o-ring, then the hex nut. Got it? Ok!
Thread Cables in Hilt
Begin to insert the two cables back into the hilt.

Ensure the hex nuts are positioned so they can be fitted onto the button.

Use both hands to maneuver the cables through the hilt while maintaining the o-ring and hex nut.

Secured Button to Hilt
Begin to panel mount the button by pushing it into the hole in the hilt. Insert the o-ring onto the threading. Then, fasten the hex nut onto the button until it has a tight fitting.

Tweezers can be used to help grab hold of the hex nut.
Secured Button
Press the button a few times to see if it has a tight fit on the hilt.

Hilt Check
The two cables should stick out the bottom of the hilt. These are rather lengthy in order to connect to the PropMaker FeatherWing.

Feather Assembly

Hardware for Feathers
Use the following hardware to secure the Feather to the PCB holder.

4x M2.5 x 10mm screws
8x M2.5 hex nuts
Install Screws and Nuts to Feather
Insert the 4x M2.5 x 10mm screws through the Feathers mounting holes. Then, insert and fasten hex nuts onto the thread of the screws.

The nuts serve as very-low profile standoffs. This helps to elevate the Feather when installed to the holder.

Installing Feather to PCB Holder
The speaker and battery must be connect BEFORE securing the Feather to the PCB holder.

Please note: Once the Feather is mounted to the PCB holder, the speaker and battery connection will not be accessible.

Connect Speaker to PropMaker
Plug in the speaker to the audio port on the PropMaker FeatherWing.
Connect Button to PropMaker
Grab the 4-pin picoBlade connector from the button and the Feather. Connect the 4-pin picoBalde cables together.

Connect Battery to Feather
Plug in the battery to the battery port on the Feather board.

Test Circuit
Let's test the circuit before installing the Feather board to the PCB holder.
Use the button to turn on the circuit.
Hilt Assembly

Installing Blade to Hilt
Get ready to install the blade to the hilt.

Install Strip Cable to Hilt
Insert the 3-pin JST cable from the strips through the top of the hilt.

NeoPixel Cable Pass Thru Hilt
Pull the 3-pin JST cable all the way through the hilt and out the other end. It should have enough length to reach the PropMaker FeatherWing.
Installing Hilt to Coupler
Pull the 3-pin JST cable so the hilt can begin to first into the coupler.

Fit the coupler into the hilt. Reference the photo for the correct orientation.

Make sure the cable is not being kinked when fitting the hilt into the coupler.

Secure Hilt to Coupler
Insert and fasten an M3 x 10mm pan head machine screw. Fasten until fully tightened to secure the hilt.

Connect NeoPixel Strip to PropMaker FeatherWing
Plug in the 3-pin JST cable from the strip to the PropMaker FeatherWing.
Install Speaker into Holder
The speaker is press fitted into the speaker holder. Reference the photo for best placement.

The speaker can be removed by popping it out using the hole on the other side.

 Optionally peel off the protective backing if you’d like to stick it to the speaker holder. Note: Speaker will be harder to remove once the adhesive is stuck to the surface.
Install Battery into Holder
Press the battery through the holder from the ends. Do not attempt to insert the whole battery through the clip (it has a very tight fit and could break). Once the battery is fitted into the holder it can slide into place.
Install Feather into PCB holder

With the speaker, battery and NeoPixel strips now connected, the Feather is ready to be secured to the PCB holder.

Gather up the 4x M2.5 hex nuts for securing the Feather board to the PCB holder.

Place the Feather board into the PCB holder with the screws going through the mounting holes.
Secure and Snip
Insert and fasten the M2.5 hex nuts onto the threads of the screws.

Tightly fasten the hex nuts onto the screws.

The threads must be trimmed short in order to fit through the hilt. Use flush diagonal cutters to snip the screws short.

Inside the PCB Holder
The battery cable should have plenty of wiggle room. A look through the PCB holder shows there's enough space for additional connections.
Install Speaker to Hilt
The speaker is inserted through the bottom of the hilt and pushed up inside. Use a screw driver or similar to push the speaker all the through the hilt.
Install Battery to Hilt

The wiring from the speaker, strips and button are press fitted through the slits in the battery holder. This helps manage the wires and prevent them from kinking when fitted inside the hilt.

Carefully insert the battery holder into the bottom of the hilt.

Push the battery holder all the way into the hilt.
Install PCB Holder to Hilt

Begin to insert the PCB holder into the hilt. Push the PCB holder into the hilt until its flush with the hilt.

The ledge on the end of the PCB holder allow it to be remove from the hilt by pulling it out.

The extra lip allows you to grab hold of the PCB holder.
Access to Feather USB Port

The USB port on the Feather board is accessible through the bottom of the hilt.

Use the USB port to recharge the battery when it gets low.
Install Pommel
The pommel is press fitted over the bottom of the hilt.

Pull the pommel out to remove it from the hilt when it's time to recharge or reprogram.

Assembled Hilt
And there we have it! The darksaber is fully assembled and ready for cosplaying.

Congratulations on your build, this is the way, and May The Force Be With You!
Usage

Bluetooth Control
This project uses the Adafruit Bluefruit LE connect app (available free for Android and iOS) to trigger the lights and sounds. It uses the control pad to trigger up to 8 different colors and sound effects. If you haven't downloaded the app yet, use the button below to install it on your mobile device.

Download BLE Connect App iOS
Download BLE Connect App for Android

Using Bluefruit LE Connect App

Turn on the Feather by either connecting it via USB to your computer or with the RGB LED button.

Open the Bluefruit LE Connect app and locate the device named CIRCUITPY and tap the connect button. Locate and tap on Controller. Under module, tab on Color Picker.

Color Picker
You can change the color of the NeoPixel LEDs by using the Color Picker. Tap connect, select Controller. Under module, tap on Color Picker. Use the color wheel to select a color and the slider to adjust the brightness. Tap the send color button to trigger the color.
Apple Watch
If you have an Apple Watch, you can use the Bluefruit LE Connect App just like the mobile app. The Apple Watch app is included with the iOS app download.