Table of Contents

Overview 3

• Parts

Build the Blaster Circuit 7

• NeoPixels
• And More NeoPixels
• Connect the Strands
• Quad Board
• Add Amps
• Connect Speakers & Switches
• NeoPixels, USB, & Battery Power

CircuitPython on Feather M4 Express 18

• Set up CircuitPython Quick Start!
• Further Information

Code the Blaster 21

• Text Editor
• Installing Project Code
• Audio Samples
• Make Your Own
• What It Does

Original Lucio Blaster Construction 26
Overview

It's time for a modernized prop blaster! This guide is an overhaul of the Lucio Blaster project from 2017 (The original was a three part guide [here](#), [here](#), and [here](#). You can also use this as a leaping off point for any prop or cosplay element that requires lights, music, and triggered sound effects.

A lot has changed in three years, and the project has been greatly simplified, thanks to advances in hardware and software! We'll now use the Feather M4 Express running CircuitPython and a Prop-Maker FeatherWing.

Thanks to dual-stream, native .mp3 playback in CircuitPython, we can eliminate the need for multiple pieces of hardware previously required to play music and sound effects at the same time, and the power, amplification, and NeoPixel driving circuits have been similarly simplified.

And speaking of NeoPixels -- the LED animation library makes it astonishingly simpler to create highly awesome lighting effects!

This guide shows how to connect all the electronics you need to make your own Sonic Amplifier, or modify them to any prop need.

A future update will show all of the guts fit into a new 808-themed 3D printed blaster! Download the [model files here](#) to make your own.
Here's some demo video of the original version to get a feel for the project:

Parts

Adafruit Feather M4 Express - Featuring ATSAMD51
It's what you've been waiting for, the Feather M4 Express featuring ATSAMD51. This Feather is fast like a swift, smart like an owl, strong like a ox-bird (it's half ox,... https://www.adafruit.com/product/3857

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 NeoPixel Digital RGBW LED Strip - Black PCB 144 LED/m
What is better than smart RGB LEDs? Smart RGB+White LEDs! These NeoPixels now have 4 LEDs in them (red, green, blue and white) for excellent lighting effects. These LED...
[https://www.adafruit.com/product/2848](https://www.adafruit.com/product/2848)

### Adafruit Mono 2.5W Class D Audio Amplifier - PAM8302
This super small mono amplifier is surprisingly powerful - able to deliver up to 2.5 Watts into 4-8 ohm impedance speakers. Inside the miniature chip is a class D controller, able to...
[https://www.adafruit.com/product/2130](https://www.adafruit.com/product/2130)

### Adafruit Quad Side-By-Side FeatherWing Kit with Headers
This is the FeatherWing Quad Side-by-Side - a prototyping add-on and more for all Feather boards. This is similar to our
[https://www.adafruit.com/product/4254](https://www.adafruit.com/product/4254)

<table>
<thead>
<tr>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 x <strong>NeoPixel 1/4 60 Ring</strong></td>
<td><a href="https://www.adafruit.com/product/2875">https://www.adafruit.com/product/2875</a></td>
</tr>
<tr>
<td>5050 RGBW LED w/ Integrated Drivers</td>
<td></td>
</tr>
<tr>
<td>3 x <strong>NeoPixel Ring - 16</strong></td>
<td><a href="https://www.adafruit.com/product/2856">https://www.adafruit.com/product/2856</a></td>
</tr>
<tr>
<td>5050 RGBW LEDs w/ Integrated Drivers</td>
<td></td>
</tr>
<tr>
<td>1 x <strong>NeoPixel Jewel</strong></td>
<td><a href="https://www.adafruit.com/product/2860">https://www.adafruit.com/product/2860</a></td>
</tr>
<tr>
<td>7 x 5050 RGBW LED w/ Integrated Drivers</td>
<td></td>
</tr>
</tbody>
</table>
1 x **Colorful Round Tactile Button Switch Assortment** [https://www.adafruit.com/product/1009](https://www.adafruit.com/product/1009) 15 pack

2 x **Mini Panel Mount SPDT Toggle Switch** [https://www.adafruit.com/product/3221](https://www.adafruit.com/product/3221)  
ON/OFF

1 x **Terminal Block kit for Feather** [https://www.adafruit.com/product/3173](https://www.adafruit.com/product/3173)  
0.1" Pitch

1 x **Lithium Ion Battery Pack** [https://www.adafruit.com/product/353](https://www.adafruit.com/product/353)  
3.7V 6600mAh

3 x **Thin Plastic Speaker w/Wires** [https://www.adafruit.com/product/1891](https://www.adafruit.com/product/1891)  
8 ohm 0.25W

6 x **JST PH 3-Pin to Male Header Cable** [https://www.adafruit.com/product/4046](https://www.adafruit.com/product/4046)  
200mm

6 x **JST PH 3-Pin to Socket Cable** [https://www.adafruit.com/product/4046](https://www.adafruit.com/product/4046)  
200mm

1 x **Molex Pico Blade 2-pin Cable** [https://www.adafruit.com/product/3922](https://www.adafruit.com/product/3922)  
200mm

---

**Silicone Cover Stranded-Core Wire - 26AWG in Various Colors**  
Silicone-sheathing wire is super-flexible and soft, and it's also strong! Able to handle up to 200°C and up to 600V, it will do when PVC covered wire wimps out. We like this...  
We'll build out the circuit starting with the NeoPixels. These will plug into the JST port on the Propmaker FeatherWing which can drive them with more power than a bare microcontroller on its own.

A pair of toggle switches will connect the Propmaker for powering the project on and off via the Enable pin, and for switching song/color modes.

To drive the four speakers we'll connect one to the amp built into the Propmaker and then add three 2.5W class D amps for the remaining speakers.

We'll plug the Feather and Propmaker into a quad FeatherWing kit which gives us plenty of prototyping area to connect the amps and the trigger buttons. Adding Feather terminal headers make it easy to connect the buttons.

Here's a Fritzing diagram of the complete circuit:
NeoPixels
We'll create a single circuit of all the NeoPixel strands, rings, and jewel so that they share a single data line, ground, and power.

First, solder together the four quarter segments into a single ring. When doing so, be sure to solder all of the power and ground pads, and three of the four Data In/Data Out pairs.

On the un-soldered Data In segment, solder a JST wiring interconnect cable so that the ring can be connected to the rest of the NeoPixels -- this will be the last set of NeoPixels so it doesn't need an out cable.
And More NeoPixels
Solder a pair of in/out interconnect cables to each ring, jewel, and strand so they may all be joined.

It's important to be consistent with polarity of interconnects so all of the Data In sides can only be connected with Data Out sides.
Connect the Strands
I decided to wire the two VU meter strips directly to each other, and use the interconnects at either end for connecting to the Propmaker FeatherWing and the rest of the circuit.

Plug them all together and they're ready for use!

You may choose to wire each segment directly to the next, which will work just fine. Interconnects are a bit of extra effort, but give you added freedom during integration with your prop blaster parts.
Quad Board

Solder female headers into the top two positions of the Quad board.

Solder male headers to the underside of the Feather and the Propmaker so they can be attached to the Quad.

Then, solder the terminal blocks to the bottom quad area.
Add Amps

Use three sections of male header pin rows to solder the three amps to the prototyping area as shown, pushing the long ends of the pins down into the board so they can be used as posts for soldering the wires.

We'll wire the circuit connections as they're shown in the Frizing diagram, except we'll do it underneath the board.

Solder a set of wires from the Quad board GND to the GND posts of the two bottom amps as shown in the first wiring picture here.

This is followed by running a circuit from the +V rail to the appropriate posts.

Then, run the A0 pin to the A+ (audio in) pin on both bottom amps and the A1 pin to the A+ of the top amp.
Connect Speakers & Switches

Connect two of the small speakers and the large front speaker to their respective amps as shown using the terminal blocks.

The third small speaker will connect to the speaker output on the Propmaker. Do this by grafting in a Molex "Pico-Blade" connector ().

You’ll use the Feather terminal headers to connect the trigger buttons to ground, A5 and A6.

The toggle switches are connected to the En & GND pins on the Propmaker wing and the SW & GND pins, also on the Propmaker. You can solder these directly or add a couple of 0.1” terminal blocks first.
NeoPixels, USB, & Battery Power

You can now plug in the LiPoly battery as well as the USB panel mount to extend the Feather's USB port to make it accessible from the exterior of the prop. The USB will be used both for programming the Feather as well as charging the battery.

Plug the NeoPixel mega strand into the JST socket on the Propmaker FeatherWing and the blaster is ready to go!
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 :)

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.

Code the Blaster

Text Editor

Adafruit recommends using the Mu editor for using your CircuitPython code with the Feather boards. You can get more info in this guide.

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

Installing Project Code

To use with CircuitPython, you need to first install a few libraries, into the lib folder on your CIRCUITPY drive. Then you need to update code.py with the example script.

Thankfully, we can do this in one go. In the example below, click the Download Project Bundle button below to download the necessary libraries and the code.py file in a zip file. Extract the contents of the zip file, open the directory Lucio_Blaster_2020 / and then click on the directory that matches the version of CircuitPython you're using and copy the contents of that directory to your CIRCUITPY drive.

Your CIRCUITPY drive should now look similar to the following image:
# SPDX-FileCopyrightText: 2020 John Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

# Lucio 2020
# Feather M4 + Propmaker + amps + lots of neopixels
import board
import busio
from digitalio import DigitalInOut, Direction, Pull
import audioio
import audiomixer
import audiomp3
import adafruit_lis3dh
import neopixel
from adafruit_led_animation.animation.solid import Solid
from adafruit_led_animation.animation.comet import Comet
from adafruit_led_animation.animation.pulse import Pulse
from adafruit_led_animation.helper import PixelSubset
from adafruit_led_animation.group import AnimationGroup
from adafruit_led_animation.color import RED, ORANGE, WHITE

ORANGE_DIM = 0x801400  # half value version
RED_DIM = 0x800000

# ---Set Volume Max Here---
VOLUME_MULT = 0.65  # 1 = full volume, 0.1 is very quiet, 0 is muted

# ---SWITCH/BUTTON SETUP---
mode_switch = DigitalInOut(board.D9)
mode_switch.switch_to_input(pull=Pull.UP)
mode_state = mode_switch.value
trig_button = DigitalInOut(board.A4)
trig_button.switch_to_input(pull=Pull.UP)
alt_button = DigitalInOut(board.A5)
alt_button.switch_to_input(pull=Pull.UP)

# ---ACCELEROMETER SETUP---
# Set up accelerometer on I2C bus, 4G range:
i2c = busio.I2C(board.SCL, board.SDA)
int1 = DigitalInOut(board.D6)
accel = adafruit_lis3dh.LIS3DH_I2C(i2c, int1=int1)

# ---SPEAKER SETUP---
enable = DigitalInOut(board.D10)
enable.direction = Direction.OUTPUT
enable.value = True
# Set up speakers and mixer. Stereo files, where music has empty right channel, FX
# empty left
speaker = audioio.AudioOut(board.A0, right_channel=board.A1)
mixer = audiomixer.Mixer(channel_count=2, buffer_size=2304, sample_rate=22050)

# ---NEOXPIXEL SETUP---
pixel_pin = board.D5
pixel_num = 154
pixels = neopixel.NeoPixel(
    pixel_pin, pixel_num, brightness=0.6, auto_write=False,
    pixel_order=neopixel.GRBW
)
# ^ change pixel_order depending on RGB vs. RGBW pixels

# ---Pixel Map---
# this is the physical order in which the strips are plugged
pixel_stripA = PixelSubset(pixels, 0, 18)  # 18 pixel strip
pixel_stripB = PixelSubset(pixels, 18, 36)  # 18 pixel strip
pixel_jewel = PixelSubset(pixels, 36, 43)  # 7 pixel jewel
pixel_ringsAll = PixelSubset(pixels, 43, 151)  # all of the rings
# or use rings individually:
# pixel_ringA = PixelSubset(pixels, 43, 59)  # 16 pixel ring
# pixel_ringB = PixelSubset(pixels, 59, 75)  # 16 pixel ring
# pixel_ringC = PixelSubset(pixels, 75, 91)  # 16 pixel ring
# pixel_ringD = PixelSubset(pixels, 91, 151)  # 60 pixel ring

# ---BPM---
BPM = 128
BEAT = 60 / BPM  # quarter note beat
b16TH = BEAT / 4  # 16TH note beat
b64TH = BEAT / 16  # sixty-fourth note beat

# ---Anim Setup---
# heal color mode
# Pulse 'speed' = smoothness
pulse_rings_m0 = Pulse(pixel_ringsAll, speed=0.01, color=ORANGE, period=BEAT)
pulse_jewel_m0 = Pulse(pixel_jewel, speed=0.01, color=ORANGE, period=BEAT)
comet_stripA_m0 = Comet(
    pixel_stripA, speed=b64TH, color=ORANGE, tail_length=9, bounce=False
)
comet_stripB_m0 = Comet(
    pixel_stripB, speed=b64TH, color=ORANGE, tail_length=9, bounce=False
)

# speed color mode
pulse_rings_m1 = Pulse(pixel_ringsAll, speed=0.02, color=RED, period=BEAT / 2)
pulse_jewel_m1 = Pulse(pixel_jewel, speed=0.02, color=RED, period=BEAT / 2)
comet_stripA_m1 = Comet(
    pixel_stripA, speed=b64TH, color=RED, tail_length=9, bounce=False
)
comet_stripB_m1 = Comet(
    pixel_stripB, speed=b64TH, color=RED, tail_length=9, bounce=False
)

solid_white = Solid(pixel_ringsAll, color=WHITE)

# ---Anim Modes---
vu_strip_animations_mode0 = AnimationGroup(comet_stripA_m0, comet_stripB_m0,
    sync=True)
vu_strip_animations_mode1 = AnimationGroup(comet_stripA_m1, comet_stripB_m1,
    sync=True)

# ---Audio Setup---
if mode_state:
    BGM = "/lucio/bgmheal.mp3"
else:
    BGM = "/lucio/bgmspeed.mp3"
sample0 = audiomp3.MP3Decoder(open(BGM, "rb"))
FX = "/lucio/shoot.mp3"
sample1 = audiomp3.MP3Decoder(open(FX, "rb"))
speaker.play(mixer)
mixer.voice[0].play(sample0, loop=True)
mixer.voice[0].level = 0.3 * VOLUME_MULT
mixer.voice[1].level = 0.7 * VOLUME_MULT

© Adafruit Industries
while True:
    if mode_state:  # heal mode on startup
        vu_strip_animations_mode0.animate()
        pulse_rings_m0.animate()
        pulse_jewel_m0.animate()
    else:  # speed mode on startup
        vu_strip_animations_mode1.animate()
        pulse_rings_m1.animate()
        pulse_jewel_m1.animate()

    # Change modes
    if mode_switch.value:
        if mode_state == 0:  # state has changed, toggle it
            BGM = "/lucio/bgmheal.mp3"
            sample0.file = open(BGM, "rb")
            mixer.voice[0].play(sample0, loop=True)
            vu_strip_animations_mode0.animate()
            pulse_rings_m0animate()
            pulse_jewel_m0.animate()
            mode_state = 1
        else:
            if mode_state == 1:
                BGM = "/lucio/bgmspeed.mp3"
                sample0.file = open(BGM, "rb")
                mixer.voice[0].play(sample0, loop=True)
                vu_strip_animations_mode1.animate()
                pulse_rings_m1.animate()
                pulse_jewel_m1.animate()
                mode_state = 0
    else:
        x, _, _ = accel.acceleration  # get accelerometer values

    if not mixer.voice[1].playing:
        if not trig_button.value:  # trigger squeezed
            FX_sample = "/lucio/shoot.mp3"
            sample1.file = open(FX_sample, "rb")
            mixer.voice[1].play(sample1)
            if mode_state:
                solid_white.animate()
            else:
                solid_white.animate()
        if not alt_button.value:  # alt trigger squeezed
            FX_sample = "/lucio/alt_shoot.mp3"
            sample1.file = open(FX_sample, "rb")
            mixer.voice[1].play(sample1)
            if mode_state:
                solid_white.animate()
            else:
                solid_white.animate()
        if accel.acceleration.x > 8:  # reload
            FX_sample = "/lucio/reload.mp3"
            sample1.file = open(FX_sample, "rb")
            mixer.voice[1].play(sample1)
            if mode_state:
                solid_white.animate()
            else:
                solid_white.animate()
        if accel.acceleration.x < -8:  # Ultimate
            FX_sample = "/lucio/ultimate.mp3"
            sample1.file = open(FX_sample, "rb")
            mixer.voice[1].play(sample1)
            if mode_state:
                solid_white.animate()
Audio Samples

You can get started by using the Lucio music and sound effects .mp3 files. (Of course, you're free to use any files that you like, see below for more info on that.)

Download the .zip file linked below and then uncompress the file. Drag the /lucio directory onto the root level of your Feather's CIRCUITPY drive.

[link to download: lucio.zip]

Make Your Own

To make your own .mp3, use audio software such as [Audacity](https://audacity.sourceforge.net) to save them with these settings:

- bit rate: anywhere from 16 to 320 kb/s (lower will be smaller, higher is better quality)
- sample rate: 22050Hz or 44100Hz are recommended, although 16kHz, 24kHz, and 48kHz should also work
- channels: mono or stereo
- must be DRM free

What It Does

These are the main features of the code:

- Play one of two background music song files depending on the mode switch position
- Play NeoPixel LED animation to the beat, the base color changed depending on mode switch position
- Trigger button plays shooting sound effect file and flashes all NeoPixels white
- Alt trigger button plays alt shooting sound effect file and flashes all NeoPixels white
- Tilt "up" of the unit (Propmaker FeatherWing's accelerometer) plays reload sound effect
- Tilt "down" of the unit plays "ultimate" sound effect
Look through the code and you’ll see places to tweak and modify the behavior. You can use the led_animation library's `PixelSubset` code to send entirely different animation and color to the different logical groupings of pixels, for example.

Original Lucio Blaster Construction

Original Lucio Blaster Construction ()