NeoPixel Mini VU Meter

Created by Ruiz Brothers

https://learn.adafruit.com/neopixel-mini-vu-meter

Last updated on 2023-08-29 04:38:44 PM EDT
## Table of Contents

### Overview
- Parts

### Circuit Diagram
- Adafruit Library for Fritzing
- Wiring Connections

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

### Coding the NeoPixel Mini VU Meter
- Upload the Code and Libraries to the QT Py RP2040
- How the CircuitPython Code Works
- PDM Mic Setup
- Defining the NeoPixel Colors
- The Loop
- Updating the NeoPixels

### CAD files
- CAD Parts List
- Build Volume
- Acrylic Template
- SVG File for Laser cutter or CNC Mill
- CAD Assembly
- Design Source Files

### Wiring
- 4-pin Cable
- Wire Mic Cable to QT Py
- Connect Mic Wires to QT Py
- Connect & Check Wiring
- Cut NeoPixel Strip
- Wire for NeoPixel Strip
- Solder Cable to NeoPixel Strip
- Wired NeoPixel Strip
- Connect NeoPixel to QT Py
- Final Wire Check

### Assembly
- Hardware for PDM Mic
- Secure PDM mic
- Connect PDM Mic Cable
- Installing QT Py
- Install QT Py
- Install LED Acrylic to Case
- Install Grid into Case
- Prep NeoPixel Strip
- Install NeoPixel Strip
- Install Strip Cover to Case
- Install Back Cover
• USB C Port
• Final Build
Build a mini VU meter inspired project using NeoPixels and CircuitPython! Powered by the QT Py RP2040, use a PDM microphone to turn audio data into a dazzling display with NeoPixels!

3D print a snap fit case to house the electronics with the mic exposed in the back cover. The grid isolates the 16x NeoPixels and a piece of Black LED acrylic diffuses the LEDs.
Parts

List of parts from the Adafruit shop.

- QT Py RP2040
- PDM Mic
- Mini Skinny NeoPixel Strip 144/meter
- Black LED Acrylic - Sheet
- 10-wire ribbon cable
- Stemma QT Cable
- M2.5 Hardware Kit

**Adafruit QT Py RP2040**
What a cutie pie! Or is it... a QT Py? This diminutive dev board comes with one of our new favorite chip, the RP2040. It's been made famous in the new [https://www.adafruit.com/product/4900](https://www.adafruit.com/product/4900)

**Adafruit PDM Microphone Breakout with JST SH Connector**
An exotic new microphone has arrived in the Adafruit shop, a PDM MEMS Microphone! PDM is the 'third' kind of microphone you can integrate with electronics,... [https://www.adafruit.com/product/4346](https://www.adafruit.com/product/4346)
1 x **STEMMA QT / Qwiic Cable**
JST SH 4-pin to Premium Male Headers Cable - 150mm Long
https://www.adafruit.com/product/4209

1 x **Black Nylon Machine Screw and Stand-off Set**
M2.5 Thread
https://www.adafruit.com/product/3299

---

**Black LED Diffusion Acrylic Panel 12" x 12"**
- 0.1" / 2.6mm thick
A nice whoppin' slab of some lovely black acrylic to add some extra diffusion to your LED Matrix project. This material is 2.6mm (0.1") thick and is made of special cast...
https://www.adafruit.com/product/4594

**Adafruit Mini Skinny NeoPixel Digital RGB LED Strip - 144 LED/m**
So thin. So mini. So teeeeeeeny-tiny. It's the 'skinny' version of our classic NeoPixel strips! These NeoPixel strips have 144 digitally-addressable pixel Mini LEDs...
https://www.adafruit.com/product/2969

**Silicone Cover Stranded-Core Ribbon Cable - 10 Wire 1 Meter Long**
For those who are fans of our silicone-covered wires, but are always looking to up their wiring game. We now have Silicone Cover Ribbon cables! These may look...
https://www.adafruit.com/product/3890
Circuit Diagram

The diagram below provides a visual reference for wiring of the components. This diagram was created using the software package Fritzing.

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.

Wiring Connections

NeoPixels

- NeoPixel GND to QT Py RP2040 GND
- NeoPixel 5V to QT Py RP2040 5V
- NeoPixel DIN to QT Py RP2040 A0

PDM Mic

- PDM Mic GND to QT Py RP2040 GND
- PDM Mic 3V to QT Py RP2040 3V
- PDM Mic CLK to QT Py RP2040 TX
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 NeoPixel Mini VU Meter

Once you've finished setting up your QT Py 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 as a zipped folder.

```python
import time
import array
import math
import board
import audiobusio
import simpleio
import neopixel

# neopixel setup
pixel_pin = board.A0
pixel_num = 16
pixels = neopixel.NeoPixel(pixel_pin, pixel_num, brightness=0.1, auto_write=False)

# function to average mic levels
def mean(values):
    return sum(values) / len(values)

# function to return mic level
def normalized_rms(values):
    minbuf = int(mean(values))
```
samples_sum = sum(
    float(sample - minbuf) * (sample - minbuf)
    for sample in values
)
return math.sqrt(samples_sum / len(values))

# mic setup
mic = audiobusio.PDMIn(board.TX,
    board.A1, sample_rate=16000, bit_depth=16)
samples = array.array('H', [0] * 16)

# variable to hold previous mic level
last_input = 0

# neopixel colors
GREEN = (0, 127, 0)
RED = (127, 0, 0)
YELLOW = (127, 127, 0)
OFF = (0, 0, 0)

colors = [GREEN, GREEN, GREEN, GREEN, GREEN, GREEN, GREEN, GREEN,
          GREEN, YELLOW, YELLOW, YELLOW, YELLOW, YELLOW, YELLOW, RED, RED]

# begin with pixels off
pixels.fill(OFF)
pixels.show()

while True:
    # take in audio
    mic.record(samples, len(samples))
    # magnitude holds the value of the mic level
    magnitude = normalized_rms(samples)
    # uncomment to see the levels in the REPL
    # print((magnitude,))

    # mapping the volume range (125-500) to the 16 neopixels
    # volume range is optimized for music. you may want to adjust the range
    # based on the type of audio that you're trying to show
    mapped_value = simpleio.map_range(magnitude, 125, 500, 0, 16)
    # converting the mapped range to an integer
    input_val = int(mapped_value)

    # if the mic input has changed from the last input value...
    if last_input != input_val:
        for i in range(input_val):
            # if the level is lower...
            if last_input > input_val:
                for z in range(last_input):
                    # turn those pixels off
                    pixels[z] = (OFF)
            # turn on pixels using the colors array
            pixels[i] = (colors[i])
pixels.show()
        # update last_input
        last_input = input_val
        time.sleep(0.01)

Upload the Code and Libraries to the QT Py RP2040

After downloading the Project Bundle, plug your QT Py RP2040 into the computer's USB port. 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 QT Py RP2040's CIRCUITPY drive.

- lib folder
- code.py

Your QT Py RP2040 CIRCUITPY drive should look like this after copying the lib folder and the code.py file.

![CIRCUITPY Drive Content](image)

How the CircuitPython Code Works

First, the strip of 16 NeoPixels are setup.

```python
# neopixel setup
def neopixel_setup:
    pixel_pin = board.A0
    pixel_num = 16
    pixels = neopixel.NeoPixel(pixel_pin, pixel_num, brightness=0.1, auto_write=False)
```

PDM Mic Setup

Then the PDM mic is setup, along with two functions that assist with returning usable data from the PDM mic.

```python
# function to average mic levels
def mean(values):
    return sum(values) / len(values)

# function to return mic level
def normalized_rms(values):
    minbuf = int(mean(values))
```
samples_sum = sum(float(sample - minbuf) * (sample - minbuf) for sample in values)

return math.sqrt(samples_sum / len(values))

# mic setup
mic = audiobusio.PDMIn(board.TX,
    board.A1, sample_rate=16000, bit_depth=16)
samples = array.array('H', [0] * 16)

Defining the NeoPixel Colors

Four colors for the NeoPixels are setup. The `colors` array holds the colors in the order that they will appear on the VU meter.

```python
# neopixel colors
GREEN = (0, 127, 0)
RED = (127, 0, 0)
YELLOW = (127, 127, 0)
OFF = (0, 0, 0)

# array of colors for VU meter
colors = [GREEN, GREEN, GREEN, GREEN, GREEN, GREEN, GREEN, GREEN,
          GREEN, YELLOW, YELLOW, YELLOW, YELLOW, YELLOW, RED, RED]

# begin with pixels off
pixels.fill(OFF)
pixels.show()
```

The Loop

In the loop, data is received by the PDM mic and is held in the variable `magnitude`. `magnitude` is scaled to fit into the range of 0 to 16 using `map_range()`. This allows the 16 NeoPixels to represent the range of data coming from the PDM mic.

```python
# take in audio
mic.record(samples, len(samples))
# magnitude holds the value of the mic level
magnitude = normalized_rms(samples)
# uncomment to see the levels in the REPL
# print((magnitude,))

# mapping the volume range (125-500) to the 16 neopixels
# volume range is optimized for music. you may want to adjust the range
# based on the type of audio that you're trying to show
mapped_value = simpleio.map_range(magnitude, 125, 500, 0, 16)
# converting the mapped range to an integer
input_val = int(mapped_value)
```
Updating the NeoPixels

The NeoPixels are updated if the data from the PDM mic changes. This is checked by comparing `last_input` and `input_val`.

A `for` statement is used to iterate through the range of the current input data from the PDM mic. The index, represented with `i`, is used to turn on the correct number of pixels and the defined colors from the `colors` array.

```python
# if the mic input has changed from the last input value...
if last_input != input_val:
    for i in range(input_val):
        # if the level is lower...
        if last_input > input_val:
            for z in range(last_input):
                # turn those pixels off
                pixels[z] = (OFF)
            # turn on pixels using the colors array
            pixels[i] = (colors[i])
            pixels.show()
        # update last_input
        last_input = input_val

time.sleep(0.01)
```
CAD files

CAD 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:

case
grid
strip-cover
back-cover

Download CAD source
Download STLs.zip

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

120mm (X) x 30mm (Y) x 50mm (Z)
Acrylic Template
Use the template to cut out the piece of black LED acrylic. The dimensions are listed in the template. Print the PDF and do not scale.

Cut inside the purple outline. The light gray color resembles the actual piece of acrylic needed.

SVG File for Laser cutter or CNC Mill

CAD Assembly
The QT Py snap fits into the built-in holder on the strip-cover. The PDM mic is secured to the strip cover using M2.5 screws and hex nuts. The NeoPixel strip is placed over the grip into the recess. The piece of black LED acrylic is press fitted into the case. The grid is fitted into the case. The strip cover is press fitted into the case.

Design Source Files
The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more. Electronic components like Adafruit's board, displays, connectors and more can be downloaded from the Adafruit CAD parts GitHub Repo. 
Wiring

4-pin Cable
Use the 4-pin cable to connect the PDM mic to the QT Py. This cable features header pins that are not needed.

Wire Mic Cable to QT Py
Cut the header pins from the wires so the cable has a length of about 114mm (4.5in).

Using wire stripper, remove a bit of insulation from the tips of each wire. Tin the tips of the wires by adding a bit of solder. This helps prevent the strands of wire from fraying.

Connect Mic Wires to QT Py
Solder the four wires from the cable to the pins on the QT Py. Follow the list to reference which wires go where.

Red Wire – 3V pin on QT Py
Black Wire – GND pin on QT Py
Yellow Wire – TX pin on QT Py
Blue Wire – A1 pin on QT Py
Connect & Check Wiring
Plug in the JST connector from the cable to the JST port on the PDM mic. Double check the wires are properly soldered to the pins on the QT Py.

Cut NeoPixel Strip
Locate the end of the strip with the Data In arrow marking.
Count 16 LEDs from the Mini Skinny NeoPixel strip and cut in between the marked pads.
Double check the strip has the correct number of NeoPixels.

Wire for NeoPixel Strip
Use the 10-wire ribbon cable to create a 3-wire cable for connecting the NeoPixel strip to the QT Py.
Measure and cut a piece of ribbon cable so it's about the length of the strip of NeoPixels.
Using wire stripper, remove a bit of insulation from the tips of each wire. Tin the tips of the wires by adding a bit of solder.
Solder Cable to NeoPixel Strip
Locate the end of the NeoPixel strip with the DATA IN pad.

Solder the wires from the cable to the three pads on the back of the NeoPixel flexible PCB.

Reference the labels for soldering the wires to the correct pads.

Wired NeoPixel Strip
Double check the cable has been properly soldered to the pads on the NeoPixel strip.

Get the QT PY ready to solder the NeoPixel three wired connections.
Connect NeoPixel to QT Py
Solder the three wires from the NeoPixel strip to the pins on the QT Py.

5V from NeoPixel – 5V on QT Py
GND from NeoPixel – GND on QT Py
DIN from NeoPixel – A0 on QT Py

Final Wire Check
Double check the wires from the NeoPixel strip has been properly soldered to the QT Py.

Test the circuit by plugging a USB cable to a 5V power source.

Assembly

Hardware for PDM Mic
Use the following hardware to secure the PDM mic to the 3d printed strip cover.

2x M2.5 x 6mm screws
2x M2.5 hex nuts
Secure PDM mic
Insert the M2.5 screws through the two holes on the strip cover. Place the PDM mic over so the screws fit through the two mounting holes. Insert and fasten the two hex nuts to secure the PDM mic to the strip cover.

Connect PDM Mic Cable
Ensure the 4-pin cable is connected to the PDM microphone.

Installing QT Py
Get the QT Py ready to install onto the strip cover. Orient the strip cover so the built-in holder is lined up with the USB port on the QT Py.
Install QT Py
Insert the QT Py into the built-in holder at an angle with the corners of the PCB fitting under the clips.

Double check the USB-C port is facing the correct orientation.

Slightly flex the strip cover to allow the QT Py PCB to fit into the corners of the built-in holder.
Install LED Acrylic to Case

The black LED acrylic is press fitted into the case with the shinny side facing up and the matte side facing down.

Press the acrylic so it's flush with the bottom surface of the case.
Install Grid into Case
Place the grid into the case with the closed bottom frame going in first and the recesses facing up.

Firmly press the grid into the case so it sits flush with the LED acrylic.

Prep NeoPixel Strip
Place the NeoPixel strip over the back of the strip cover.

The cable should be tucked under the PDM mic for a nice installation.
Install NeoPixel Strip
Begin to lay the NeoPixel strip over the recesses on the grid.

The NeoPixel strip should be placed with the LEDs fitting into the slots.

The strip cover will be placed over the NeoPixel strip and press fits into the case.

Install Strip Cover to Case
Double check the USB C port from the QT Py is orientated correctly with the case.

Firmly press the strip cover into the case with the QT Py and PDM mic facing up.

Install Back Cover
Get the back cover ready to install over the case.

Orient the back cover so the notch end is lined up with the USB C port on the QT Py.

Firmly press fit the back cover over the case and snap fit to close it shut.
USB C Port
Double check the USB C Port from the QT Py is accessible.

Final Build
Congratulations on your build! It's ready for some audio input.

Use a USB cable and a 5V power source to power the VU meter.