PyLeap NeoPixel Sound Meter for Circuit Playground Bluefruit

Created by Kattni Rembor

https://learn.adafruit.com/pylein-neopixel-sound-meter

Last updated on 2021-11-17 02:39:08 PM EST
# Table of Contents

| NeoPixel Sound Meter | 3 |

©Adafruit Industries
**NeoPixel Sound Meter**

There is a microphone (sound sensor) and ten NeoPixel LEDs on your Circuit Playground Bluefruit. This example uses the ten LEDs to indicate sound level from the sensor.

Try talking to your board to see the number of NeoPixel LEDs increase.

```
# SPDX-FileCopyrightText: 2021 Kattni Rembor for Adafruit Industries
#
# SPDX-License-Identifier: MIT

Circuit Playground Bluefruit NeoPixel Sound Meter

Talk or make noise close to your Circuit Playground Bluefruit to see the NeoPixels light up.

```
```
from adafruit_circuitplayground import cp

# Choose a color. Defaults to red. This is an RGB value, where (r, g, b) represents red, green, and blue. Each value has a range of 0-255, where 0 is off and 255 is max intensity. You can update these values to change the colors. For example, (0, 255, 0) would be max green. You can combine numbers within the range to make other colors such as (255, 0, 180) being pink.
# Try it out!
color_value = (255, 0, 0)

# This is the sound level needed to light up all 10 NeoPixels. If all the LEDs are lighting up too easily, increase this value to make it more difficult to reach the max. If you are only able to light up a few LEDs, decrease this value to make it easier to reach the max. Full possible sound
```
# range is 0 - 65535.
sound_max = 1500

cp.pixels.auto_write = False
cp.pixels.brightness = 0.3

def scale_range(value):
    """Scale a value from 0-sound_max (chosen sound range) to 0-9 (NeoPixel range).
    Allows remapping sound value to pixel position.
    Full sound range is 0 - 65535. sound_max should be chosen based on testing."
    return round(value / sound_max * 9)

while True:
    peak = scale_range(cp.sound_level)
    for pixel in range(10):
        if pixel <= peak:
            cp.pixels[pixel] = color_value
        else:
            cp.pixels[pixel] = (0, 0, 0)  # Off
    cp.pixels.show()