

OFICINAS DE INTEGRAÇÃO I



MLV

A horizontal bar composed of three segments: red on the left, green in the middle, and blue on the right.

Music Light Visualizer

CONCEITO

Integrar, de forma responsiva, música com luz, interpretando sinais digitais em tensão elétrica, utilizando de leds para criar o efeito audiovisual.

COMPONENTES

COMPUTADOR

Software

Emissor e
interpretador de
sinais digitais

MICROCONTROLADOR

Arduino UNO

Intermediador entre
dados e tensão elétrica

FITA DE LED

WS2812B

Visualizador de sinais
de forma analógica

FONTE EXTERNA

5V - 2A

Alimentação da fita
de LED

CUSTOS

ARDUINO UNO	R\$ 25
FITA DE LED ENDEREÇÁVEL	R\$ 180
FONTE EXTERNA	R\$ 20
TOTAL	R\$ 225

SOFTWARE

PYTHON

Receptor

Extrai dados de frequência e intensidade sonora e condensa em seu valor mais dominante

ARDUINO

Interpretador

Mapeia as cores baseada na frequência e coordena o comportamento dos leds

HARDWARE

MICROCONTROLADOR

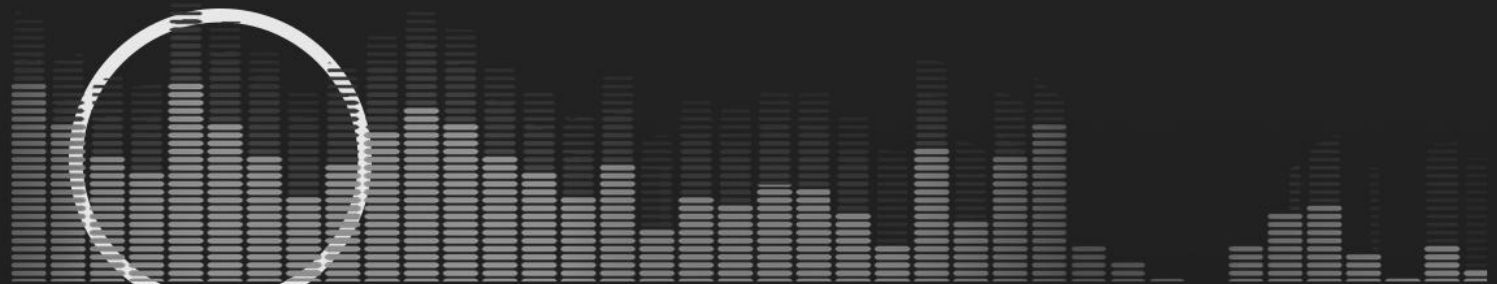
Intermediador

Converte dados em tensão elétrica responsável por controlar a fita de LED

FITA DE LED ENDEREÇÁVEL

Visualizador

Responsável por criar o efeito visual necessário para atingir o objetivo do projeto



EXTRAÇÃO DE AMOSTRA
DE DADOS

TRATAMENTO DE DADOS

PROGRAMA EM PYTHON



MAPEAMENTO DE CORES
•
ALGORITMO DE RESPOSTA
DOS LEDS

PROGRAMA DO ARDUINO



CONVERSÃO DE DADOS
EM TENSÃO ELÉTRICA

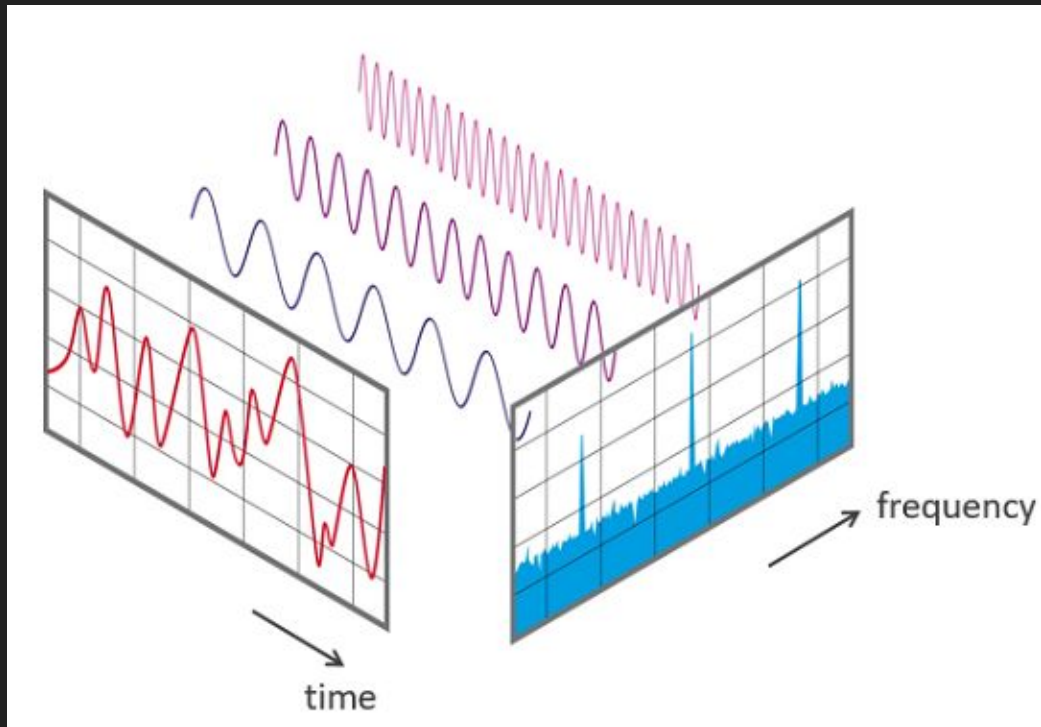
5V
2A

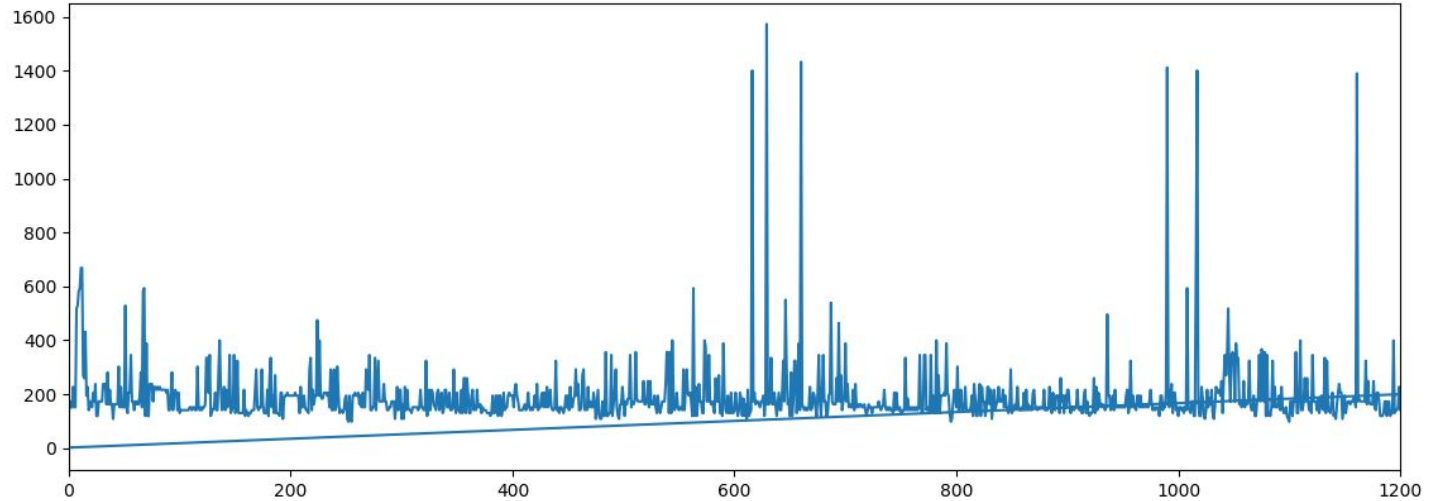


PYTHON

```
30
31 while True:
32     data2 = np.fromstring(stream_freq.read(CHUNK2), dtype=np.int16)
33     data2 = data2 * np.hanning(len(data2)) # smooth the FFT by windowing data
34     fft = abs(np.fft.fft(data2).real)
35     fft = fft[:int(len(fft) / 2)] # keep only first half
36     freq = np.fft.fftfreq(CHUNK2, 1.0 / RATE)
37     freq = freq[:int(len(freq) / 2)] # keep only first half
38     freqPeak = freq[np.where(fft == np.max(fft))[0][0]] + 1
39
```


FFT



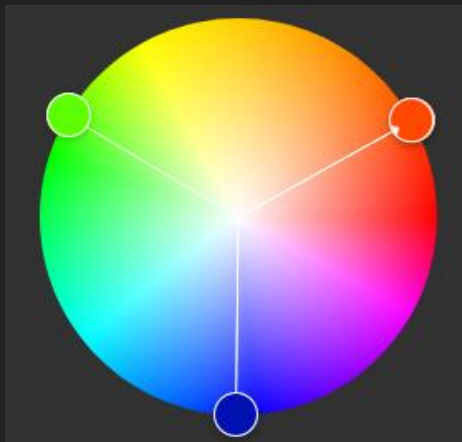


```
41     #  
42     freqPeak = freqPeak*0.255 - 18  
43     if freqPeak < 0:  
44         arduino.write(struct.pack('>B', int(0)))  
45     elif freqPeak > 255:  
46         arduino.write(struct.pack('>B', int(255)))  
47     else:  
48         arduino.write(struct.pack('>B', int(freqPeak)))  
49
```

```
49
50 for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
51     data = stream_vol.read(CHUNK)
52     rms = audioop.rms(data, 2)    # here's where you calculate the volume
53
```

```
61
62     #
63     rms = int(rms * 0.0255)
64     if rms > 255:
65         arduino.write(struct.pack('>B', int(255)))
66
67     elif rms <= 255:
68         arduino.write(struct.pack('>B', int(rms)))
69
70
```

PADRÕES DE CORES



R *red*

hue **H**

G *green*

saturation **S**

B *blue*

valor **V**



PROGRAMA DO ARDUINO

```
#include "FastLED.h"

#define LED_DT      7
#define COLOR_ORDER RGB
#define LED_TYPE     WS2812B
#define NUM_LEDS    300
#define BRIGHTNESS  100

CRGB leds [NUM_LEDS];

void setup() {
  Serial.begin(115200);
  delay(1000);
  LEDS.addLeds<LED_TYPE, LED_DT, COLOR_ORDER>(leds, NUM_LEDS);
  FastLED.setBrightness(BRIGHTNESS);
  FastLED.clear();
  FastLED.show();
}
```

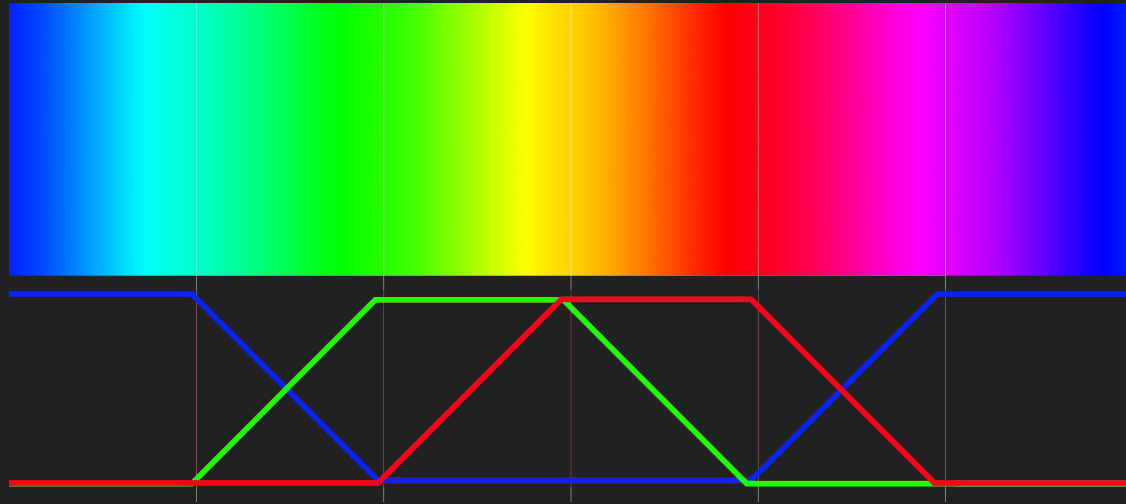
```
void piano()
{
  int i;
  for ( i = 0 ; i < vol/10 ; i++)
  {
    leds[NUM_LEDS/2 + sensorvalue/2 + i] = CHSV (sensorvalue-80, vol / 5 + 200, vol);
    leds[NUM_LEDS/2 - sensorvalue/2 - i] = CHSV (sensorvalue-80, vol / 5 + 200, vol);
  }
  FastLED.show();

  for ( i = 0 ; i < NUM_LEDS ; i++)
  {
    leds[i] -= CHSV (0,0,20);
  }

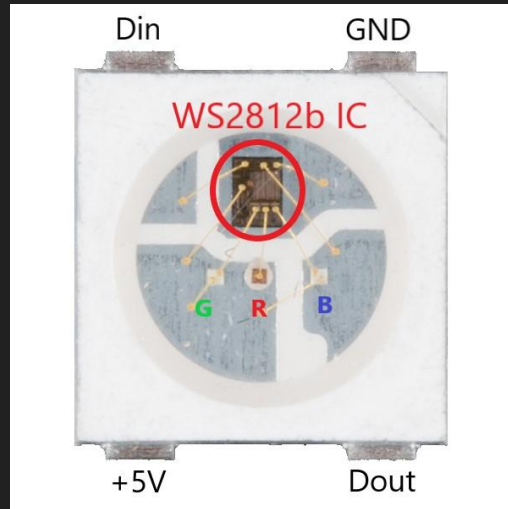
  //FastLED.clear();
}
```



MAPEAMENTO DE CORES



FITA DE LED



OBRIGADO !

youtu.be/i0ZS1kjcP40

AUGUSTO FREITAS

GUILHERME LIBÉRIO

HENRIQUE VANIN
