

Proiect Arduino

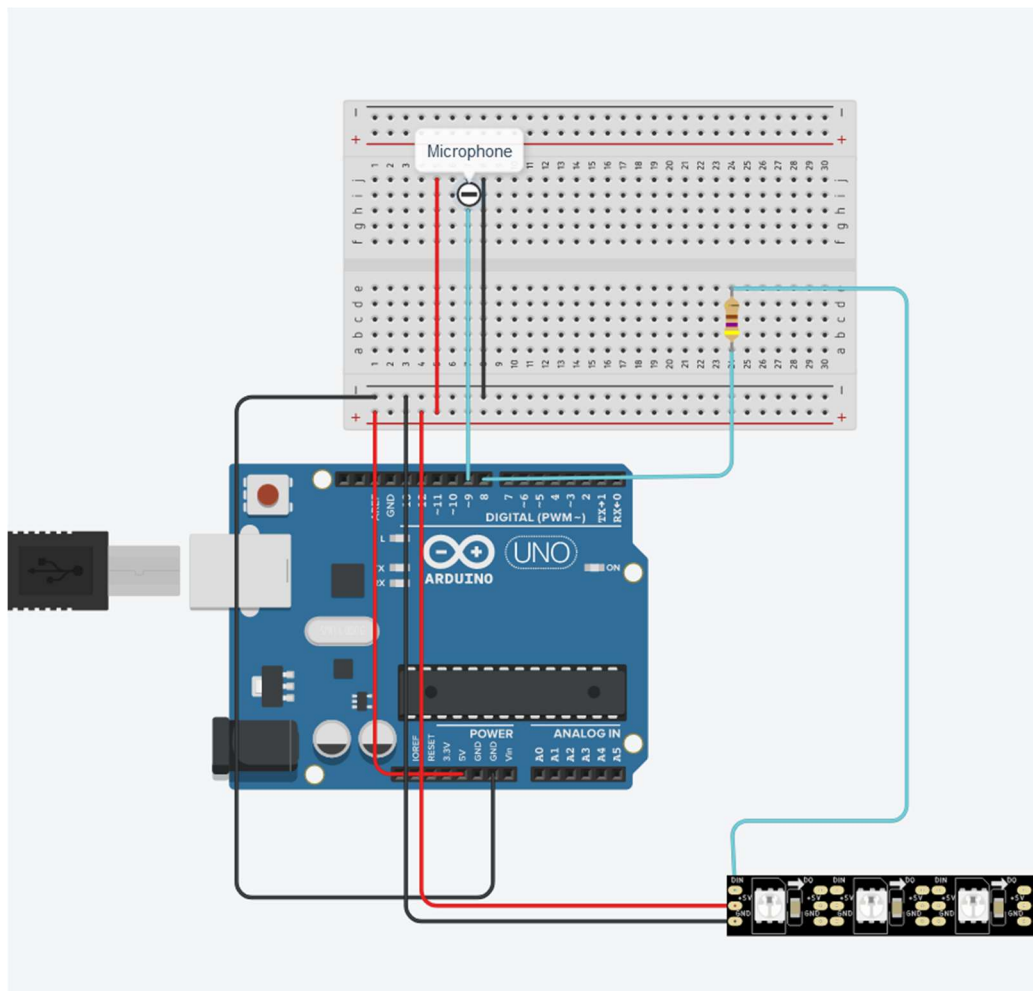
Vizualizator audio

Prezentare generala

In cadrul acestui proiect am realizat o banda LED ce raspunde prin aprinderea sau stingerea succesiva a luminilor la volumul sunetului dintr-o camera.

Prezentarea montajului

Montajul este compus din un microfon ce da un semnal binar, de tip HIGH sau LOW in functie de nivelul ales pe rezistenta variabila. Acest semnal este trimis la un Arduino UNO unde este folosit intr-un algoritm pentru a determina numarul de lumini ce trebuie aprinse. Pe conexiunea de date a benzii de 30 LED-uri este necesara o rezistenta de 470 ohm pentru limitarea curentului. Alimentarea de face direct din Arduino prin intermediul unui breadboard.



Prezentarea codului

Pe o iteratie a loop-ului, senzorul ofera un semnal de tip HIGH sau LOW, ceea ce nu este sufficient pentru masurarea volumului. Ne multumim astfel cu adunarea valorilor pe un interval de timp mic. In cazul in care pe loop avem un semnal de tip HIGH adunam la un buffer o anumita valoare pozitiva. Cand timpul intervalului expira, aprindem ledurile si resetam buffer-ul. Astfel obtinem o diferenta intre valori obtinute pe 2 intervale de timp consecutive, in functie de numarul de semnale de tip HIGH.

```
#include <FastLED.h>
const int SOUND_PIN = 8;
const int SAMPLE_TIME = 40;
unsigned long current;
unsigned long last = 0;
unsigned long elapsed = 0;
const int OUT_PIN = 9;

class Sound_lvl
{
    float old_lvl;
    float new_lvl;

public:
    Sound_lvl():old_lvl(0), new_lvl(0){};
    Sound_lvl& operator+=(float a)
    {
        new_lvl = new_lvl + a;
        return *this;
    };
    void reset()
    {
        old_lvl = new_lvl;
        new_lvl = 0;
    };
    float getold()
    {
        return old_lvl;
    }
    float getnew()
    {
        return new_lvl;
    }
};

CRGB leds[30];

void setup() {
    Serial.begin(9600);
    FastLED.addLeds<NEOPIXEL, OUT_PIN>(leds, 30);
}
```

```
Sound_lvl level = Sound_lvl();

void loop() {
    current = millis();
    elapsed = current - last;

    if (digitalRead(SOUND_PIN) == HIGH){
        level+=0.025;
    }

    if (elapsed > SAMPLE_TIME) {
        Serial.println( level.getnew() );
        if( level.getnew() > level.getold() )
        {
            for(int i = level.getold(); i < level.getnew() ; i++)
            {
                leds[i] = CRGB::Green;
                FastLED.show();
            }
        }
        else
        {
            for( int i = level.getold(); i >= level.getnew(); i--)
            {
                leds[i] = CRGB::Black;
                FastLED.show();
            }
        }
        level.reset();
        last = current;
    }
}
```

Valoarea de sample_time a fost aleasa in urma testarii. Valoarea de crestere a bufferului a fost aleasa pentru a mentine valoarea acestuia in intervalul 0-30, corespunzator numarului de led-uri.