



Neopixel ring cu senzor de sunet

Proiect atestat

Șerba Ioana Raluca

Profesor Coordonator Silviu Dumitru

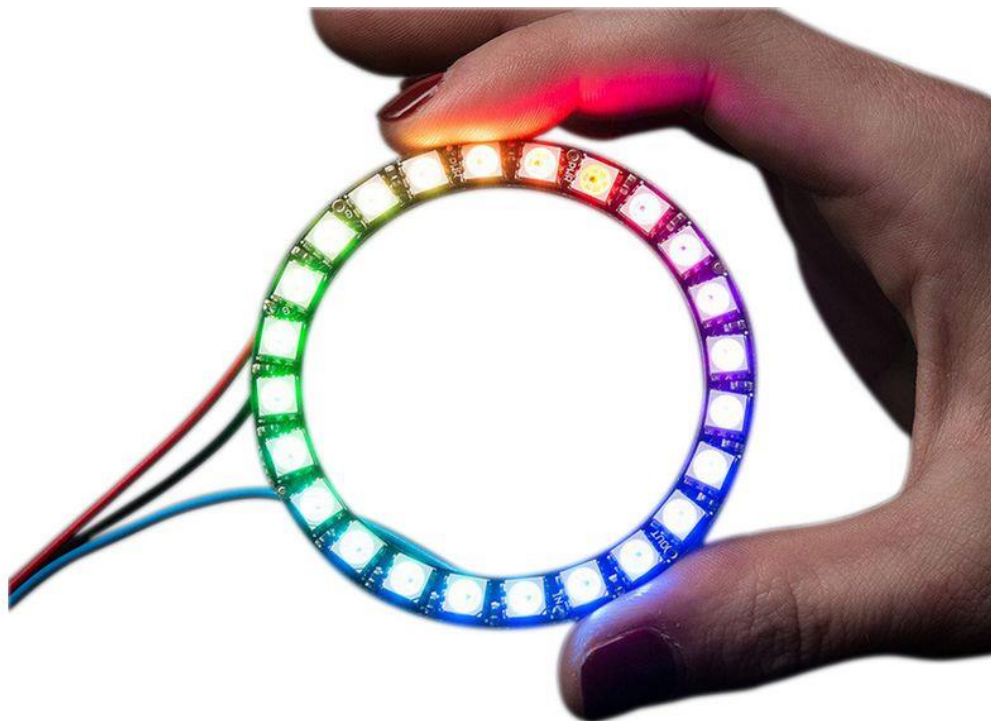


Cuprins

1. Abstract	3
2. Introducere	4
3. Arduino	5
4. Neopixel	8
5. Componente utilizate	10
6. Prezentarea algoritmului	12
7. Schema circuitului	20
8. Dificultati intampinate	21
9. Anexe	22
10. Bibliografie	27

1. Abstract

Neopixel ring este o banda de LED-uri, dispuse într-un cerc, pe care, în cadrul acestui proiect, am dorit să le combin cu muzică și arta culorilor pentru a realiza un spectacol de lumini, prin care să pot transpune muzică într-un mediu vizual.



2. Introducere

Am ales să creez acest proiect în urma descoperirii Neopixel ring-ului, care prin nenumaratele posibilități de programare mi-a atras instant atenția și curiozitatea. Din propria dragoste pentru artele vizuale, dar și artele muzicale, am dorit să integrez acest element, al sunetului, în proiectul meu, ca o metodă de a-l personaliza. Astfel, am adăugat un senzor de sunet. În prima iteratie a codului, după ce am umplut LED-ul cu culoarea dorită, și intensitatea scăzută, dacă valoarea citită de senzorul de sunet era una favorabilă, intensitatea acestuia creștea până aproape de maxim. În a doua iteratie, am utilizat o variabilă care parcurgea cercul independent față de curcubeu, doar atunci când un sunet era perceput de microfon. Această iteratie colorează LED-ul în alb, fiind urmat de 2 LED-uri tot albe, cu intensități mai scăzute, precum o ‘umbră’. Exemple de utilizări din viața cotidiană a unui proiect asemănător ar fi măștile LED, adeseori purtate la concerte sau asistenții virtuali casnici precum Alexa.



3. Ce este Arduino?

Arduino este o companie open-source care produce atât plăcuțe de dezvoltare bazate pe microcontrolere, cât și partea de software destinată funcționării și programării acestora.

Placutele de dezvoltare sunt o sursă de hardware flexibil și simplu de folosit, capabile de a prelua date printr-o serie de senzori și de a efectua acțiuni prin intermediul luminilor, motoarelor, servomotoare, și alte tipuri de dispozitive mecanice. Platforma este extrem de personalizabilă și versatilă, ceea ce o face ideală pentru o gamă largă de aplicații. Procesorul este capabil să ruleze cod scris într-un limbaj de programare care este foarte similar cu limbajul C++.

Unul dintre lucrurile grozave despre Arduino este cât de accesibil este. Cu câteva componente de bază și cunoștințe de programare, puteți începe imediat să explorați posibilitățile acestei platforme. Comunitatea online este, de asemenea, incredibil de susținătoare și activă, cu o mulțime de resurse disponibile pentru a vă ajuta să începeți.



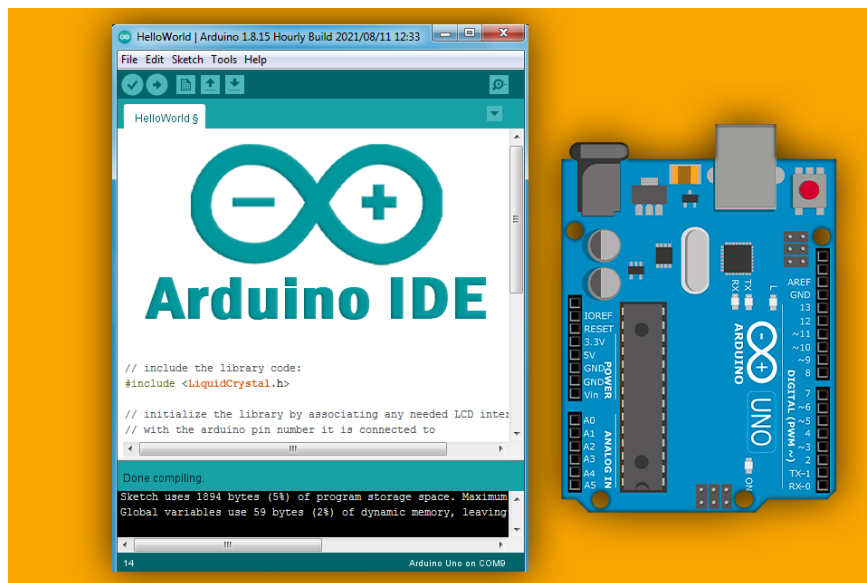
Arduino IDE (Integrated Development Environment) este o platformă software concepută pentru a facilita scrierea, compilarea și încărcarea codului pe microcontrolere Arduino. IDE-ul este open-source și poate fi descărcat gratuit de pe site-ul Arduino.

Acesta suportă limbajele de programare C și C++, însă folosește reguli speciale de organizare a codului și include un editor de cod, un compilator și un monitor de output. Codurile tipice Arduino au în general 2 părți: partea de `setup()`, care se rulează o singură dată, când se initializează setările, și partea de `loop()`, o funcție care este apelată în mod repetat până la oprirea alimentării cu energie a plăcuței. De asemenea, vine cu o serie de biblioteci încorporate și exemple pentru a ajuta utilizatorii să înceapă cu proiectele lor. IDE-ul acceptă o gamă largă de plăci Arduino și poate fi utilizat atât cu sistemele de operare Windows, cât și cu Mac.



Unul dintre cele mai mari avantaje ale IDE-ului Arduino este simplitatea acestuia. Chiar și cei cu puțină experiență în programare pot începe să creeze proiecte rapid folosind exemplele și tutorialele încorporate. IDE-ul oferă, de asemenea, o interfață ușor de utilizat pentru a se conecta la placa Arduino, facilitând încărcarea codului și monitorizarea ieșirii.

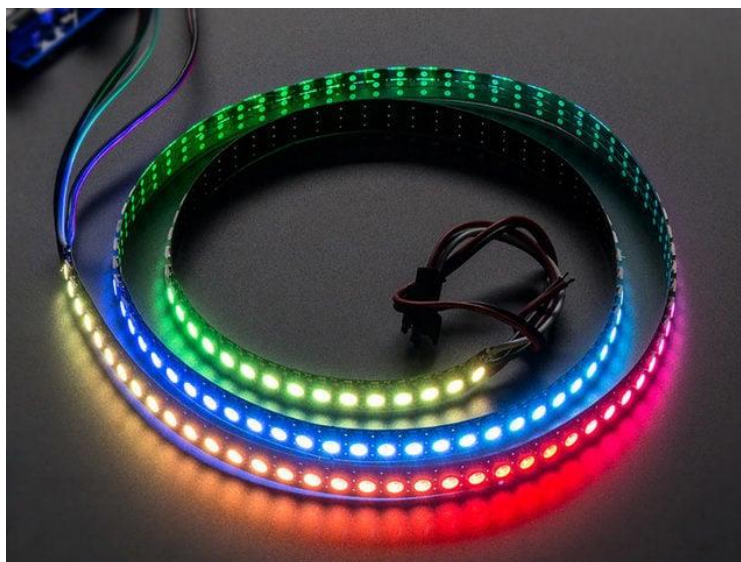
O altă caracteristică excelentă a IDE-ului Arduino este flexibilitatea sa. Platforma poate fi extinsă cu biblioteci și instrumente terțe, permițând utilizatorilor să își personalizeze mediul de dezvoltare pentru a se potrivi nevoilor lor specifice. Acest lucru îl face o alegere populară atât pentru începători, cât și pentru dezvoltatorii experimentați.



4. Ce este Neopixel?

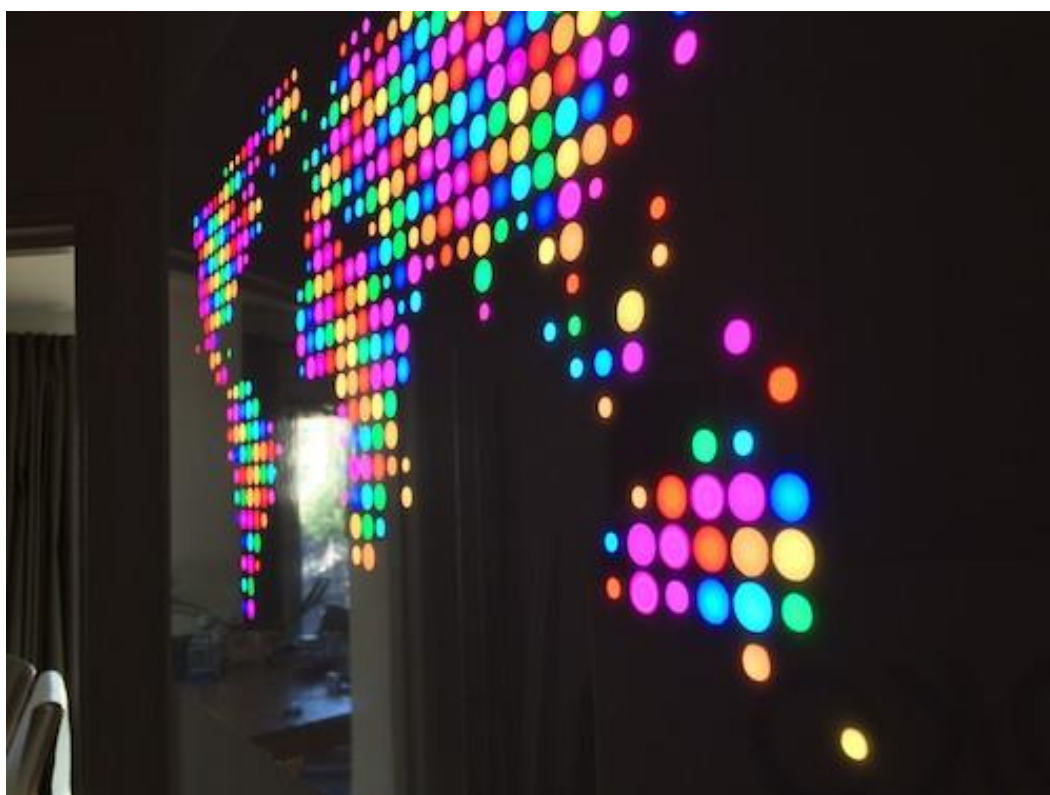
Neopixel este o marcă populară de lumini LED adresabile. Aceste lumini LED pot fi controlate individual, permițând o mare varietate de efecte creative de iluminare. Luminile LED Neopixel sunt utilizate în mod obișnuit în proiectele de bricolaj, tehnologia purtabilă și iluminatul scenei. De asemenea, sunt o alegere populară pentru crearea luminii ambientale în medii de acasă sau de birou, datorită capacității lor de a afișa o gamă largă de culori și animații.

Luminile Neopixel necesită un microcontroler pentru a funcționa, cum ar fi un Arduino sau Raspberry Pi. Ele vin într-o varietate de forme și dimensiuni, inclusiv benzi, inele și pixeli individuali. Benzile Neopixel sunt deosebit de populare pentru crearea de instalații de iluminat personalizate, deoarece pot fi tăiate la dimensiune și ușor atașate pe suprafețe.



Una dintre caracteristicile unice ale luminilor Neopixel este capacitatea lor de a fi conectate în lanțuri. Aceasta înseamnă că mai multe benzi sau pixeli Neopixel pot fi conectate împreună, permițându-le să fie controlate ca o singură unitate. Acest lucru facilitează crearea de afișaje complexe de iluminare, cum ar fi textul derulat sau modelele animate.

În general, luminile Neopixel sunt o soluție de iluminat versatilă și creativă, care a devenit o alegere populară atât pentru producători, cât și pentru designeri.



5. Componente utilizate

În realizarea proiectului, am folosit următoarele componente:

- o [placă de dezvoltare compatibilă cu Arduino UNO](#).



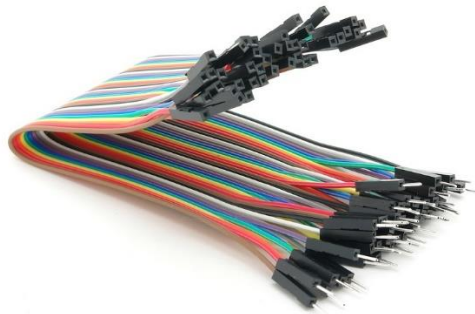
- [Neopixel ring](#) cu 24 de LED-uri, care este un cerc cu diametrul de aprox. 7cm, cu 24 de LED-uri programabile, dispuse în acest cerc.



- [Modul senzor sunet](#) care trimite un output analog, atunci cand sunetul preluat de microfon atinge o anumita intensitate, care poate fi reglata din potentiometrul modulului.



- [cabluri mama-tata](#), pentru a realiza conexiunea dintre piesele mentionate anterior



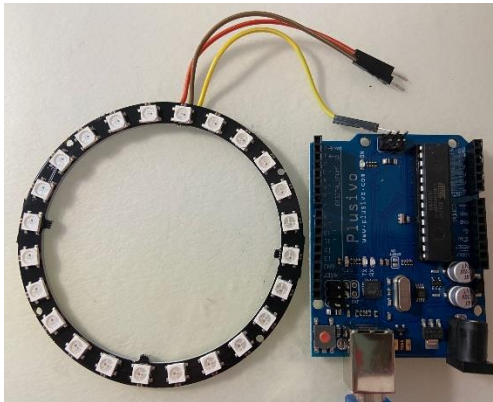
- [cablu albastru USB AM la BM](#), prin care am conectat PC-ul la placuta Arduino



6. Prezentarea algoritmului

Codul a fost scris în software-ul open-source Arduino IDE 1.8.19.

Înainte să încep scrierea programului dorit, am testat piesele. Am conectat Neopixel ring-ul și am rulat un cod simplu de aprindere a tuturor LED-urilor cu alb. Înainte de cadrul secțiunii de set-up, am inclus biblioteca „[Adafruit_NeoPixel.h](#)”, pentru a avea acces la instrucțiunile cu ajutorul cărora pot programa LED-urile. Am declarat apoi inelul sub numele de „[strip](#)”, împreună cu numărul de LED-uri, pin-ul pe care este conectat cercul și tipurile de LED-uri. În secțiunea de set-up, cu funcțiile [begin\(\)](#) și [show\(\)](#) am initializat LED-urile cu 'off', pentru a nu avea valori restante, și am setat intensitatea luminii la maxim, pentru a avea posibilitatea să o modific mai târziu.



```
#include <Adafruit_NeoPixel.h> // biblioteca specifica Neopixel

Adafruit_NeoPixel strip(24, 4, NEO_GRB + NEO_KHZ800); // declararea ring-ului meu sub denumirea de strip
// alaturi de nr de LED-uri si pin-ul pe care se afla

void setup() {
    strip.begin(); // initializarea LED-urilor cu 'off'
    strip.show();
    strip.setBrightness(255); // setarea intensitatii luminii la maxim
}

void loop()
{
    int i;
    for(i = 0; i < 24; i++)//parcurerea LED-urilor
        strip.setPixelColor(i, 50, 50, 50); // atribuirea valorilor egale de
        // rosu, verde si albastru
        // pentru a creea alb

    strip.show();// afisare
}
```

Pentru a crea un curcubeu, m-am decis sa folosesc valori de culori HSV, decat sa ofer valori nivelului de culoare rosu, verde, albastru separat, RGB, deoarece in HSV, culorile sunt deja in ordinea dorita, avand valori de la 0 la 65536. Intr-o variabila de tip long, f_hue, memorez valori in acest interval, cu incrementari de 500. Acest 500 determina cat de rapid si cat de lin se face tranzitia dintre culori. Alterior, cu o variabila i care ia valori de la 0 la 23, pentru fiecare LED in parte, folosesc functia `setPixelColor()` pentru a-i oferi culoarea. Deoarece doresc sa am intensitatea la 50 din 255, este nevoie sa folosesc functia `ColorHSV()`, in care primul argument reprezinta nuanta, al doilea saturatia culorii, pe care am lasat-o la valoarea maxima de 255, si al treilea intensitatea luminii, 50.

Pentru nuanța, ca să am un curcubeu, am luat valoarea maximă 65536 și am împărțit-o la 24, numărul de LED-uri, ca apoi să înmulțesc rezultatul cu i pentru a dispune curcubeul pe toate LED-urile. Ca acest curcubeu să se miste, am scăzut sau adăugat valoarea menționată din `f_hue`, care se schimbă după un ciclu de 24. Scăderea sau adunarea determină direcția în care se mișcă culorile, iar dacă valoarea scade sub 0 sau trece peste maxim, funcția folosită cunoaște cum să o pună în interval, precum valorile ar fi într-un cerc. Funcția `gamma32()` ajută la rafinarea valorii returnate de `ColorHSV()`.

Ca LED-urile să crească în intensitate la un sunet, am reglat potentiometrul astfel încât să se aprindă becul modulului atunci când un sunet de intensitate dorită este perceput de microfon. Am afișat aceste valori într-un program separat, și am constatat că la preluarea sunetului, valorile sunt sub 10.

```
void setup() {
}

void loop() {
    int i;
    i = (analogRead(A0)); //atribuirea valorii citite de senzor
                          // care se afla pe pin-ul analog A0
    Serial.print(i);      //afisarea in Serial Monitor
    Serial.print("\n");
    delay(10);            // cu un delay de 10 milisecunde
}
```



Înapoi la programul principal, am instructat astfel încât valoarea citită de pe pin-ul analog A0, la care este conectat modulul, dacă este sub 10, LED-ul primește aceeași culoare și saturatie, dar intensitatea luminii este setată la 200. După ce se parcurg cele 24 de LED-uri odată, cercul este pus la curent cu funcția `show()`, alături de o întârziere de 50 de milisecunde, astfel încât curcubeul să nu se miste prea rapid.



```
#include <Adafruit_NeoPixel.h> // libraria specifica Neopixel

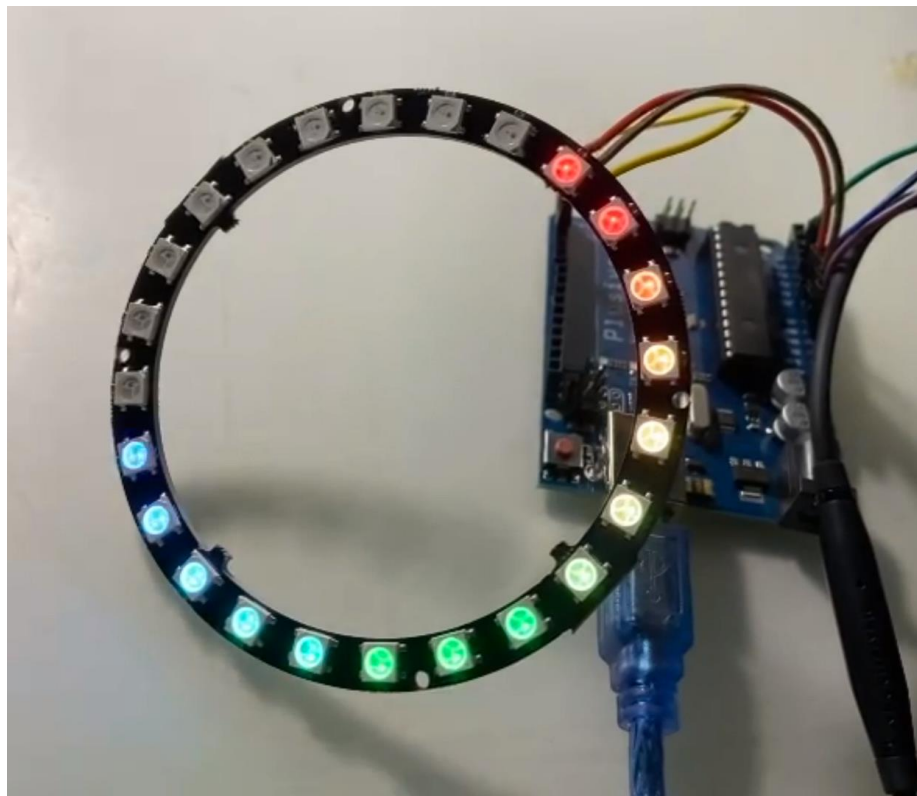
Adafruit_NeoPixel strip(24, 4, NEO_GRB + NEO_KHZ800); // declararea ring-ului meu sub denumirea de strip
// alaturi de nr de LED-uri si pin-ul pe care se afla

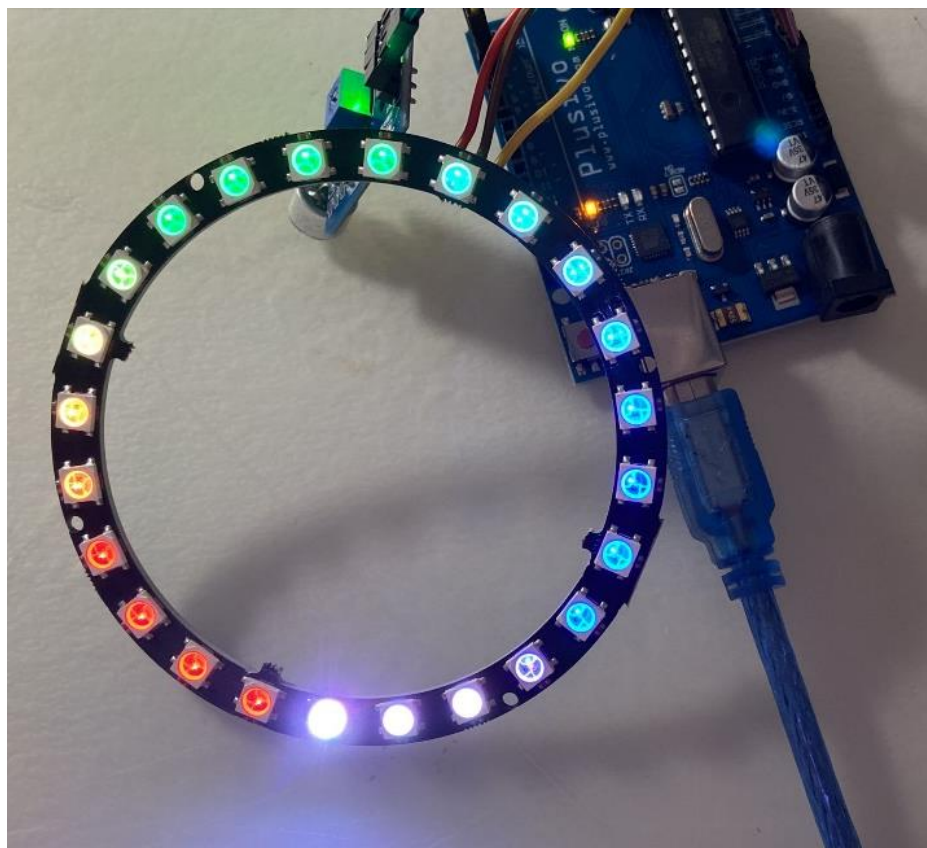
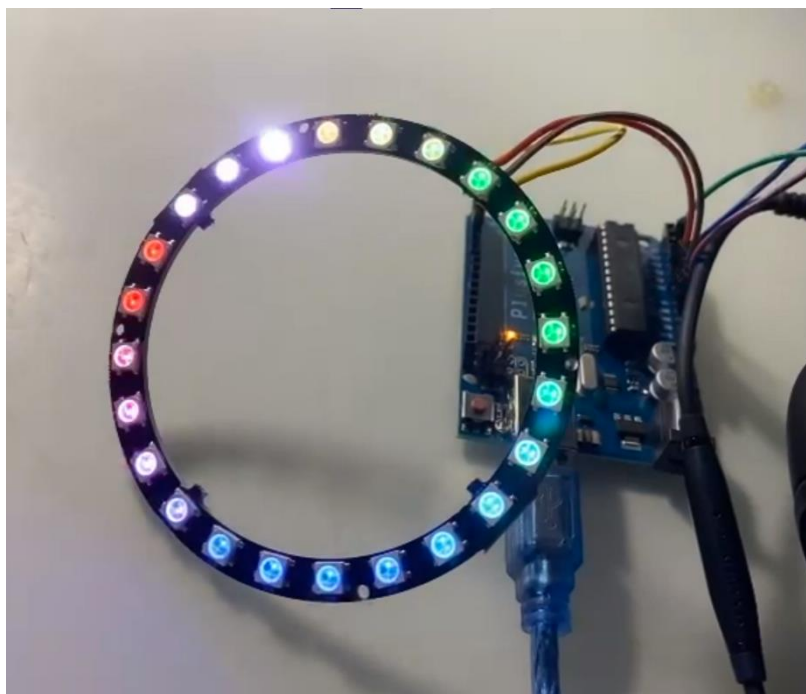
void setup() {
    strip.begin(); // initializarea LED-urilor cu 'off'
    strip.show();
    strip.setBrightness(255); // setarea intensitati luminii la maxim
}

void loop() {
    int i;
    for(long f_hue = 0; f_hue < 65537; f_hue += 500)
    { for( i = 0; i < 24; i++) // parcurgerea fiecarui LED in parte
        {
            strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue + i*2730, 255, 50))); // atribuirea culorii
            if(analogRead(A0)<10)strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue +i*2730, 255, 200 )));
            //^ cresterea intensitatii cand valoarea oferita de microfon e favorabila
        }

        strip.show(); // afisarea schimbarilor realizate anterior
        delay(50); // cu un delay de 50 de milisecunde
    }
}
```


Deși eram destul de mulțumita de rezultatele obținute, am dorit să realizez și varianta următoare: cu ajutorul unei variabile `j` de tip `int`, atunci când se preluază un sunet, LED-ul de pe poziția `j` se colorează în alb la intensitate maximă, iar cele 2 LED-uri din urma lui tot alb, la intensități de 20, respectiv 10. Acest `j` parcurge toate cele 24 de LED-uri doar când se înregistrează un sunet, separat de `i` și de curcubeu. Pentru ca variabila `i` să nu îl suprascrie, am creat și o structură `if()`, care cere unui LED să fie pe poziție diferită de `j` și cei 2 din urma lui și să nu se înregistreze un sunet, înainte să se coloreze LED-ul în curcubeu. Aceste instrucțiuni nu interferează cu restul cercului, curcubeul mișcându-se în mod normal. Ca și bonus, am realizat și o etapă de „`inițiere`” a cercului, atunci când codul este rulat pentru prima oară, se afișează cum se aprinde fiecare LED în parte.





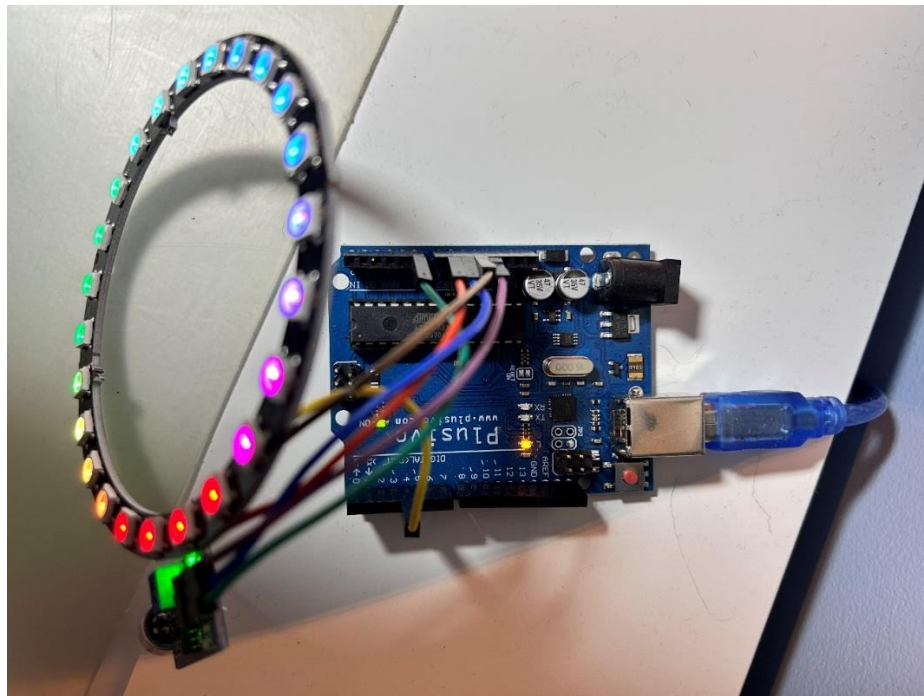
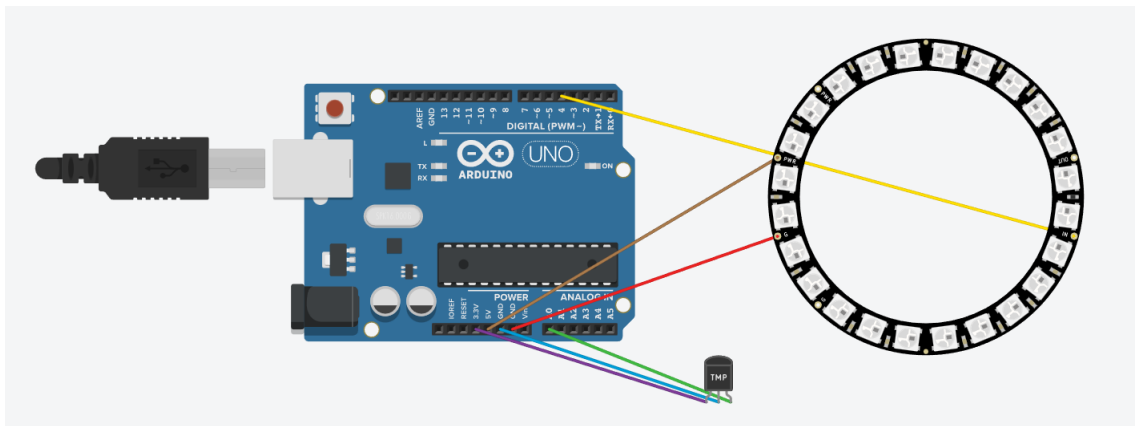


```
void loop() {
  int i;
  for(long f_hue = 0; f_hue < 65537; f_hue += 500)
  { for( i = 0; i < 24 && ok; i++)
    {
      if(j1 != i && j2 != i && j2-1 != i && analogRead(A0)>10) // verificarea necesare pentru a evita suprascrierea
        strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue +i*2730, 255, 50 )));
      if(analogRead(A0)<10 && ok) // daca output-ul de la senzor este favorabil si animatia de initere s-a realizat deja
      { strip.setPixelColor(j, 250, 250, 250); // atribuirea culorii alb la intensitate maxima
        strip.setPixelColor(j1, 20, 20, 20); // prima 'umbra'
        strip.setPixelColor(j2, 10, 10, 10); // a doua 'umbra'
        j++;
        j1 = j-1; j2 = j-2; // mutarea pe urmatorul LED
        if(j == 24 ) { j = 0; j1 = 23; j2 = 22;} // atribuirile necesare astfel incat variabilele sa ramana in cerc
        if(j==1) { j1 = 0; j2 = 23;}
      }
    }
  }
  strip.show(); // afisare cu un delay de 25 milisecunde
  delay(25);

  if(!ok) // verificarea ca animatia de 'initiere' nu s-a realizat deja
  for( i = 0; i < 24; i++)
  {
    strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue +i*2730, 255, 50 )));
    strip.show(); // afisarea dupa fiecare atribuire
    delay(50); // cu un delay de 50 de milisecunde
    if(i == 23) ok = 1;
  }
}
```

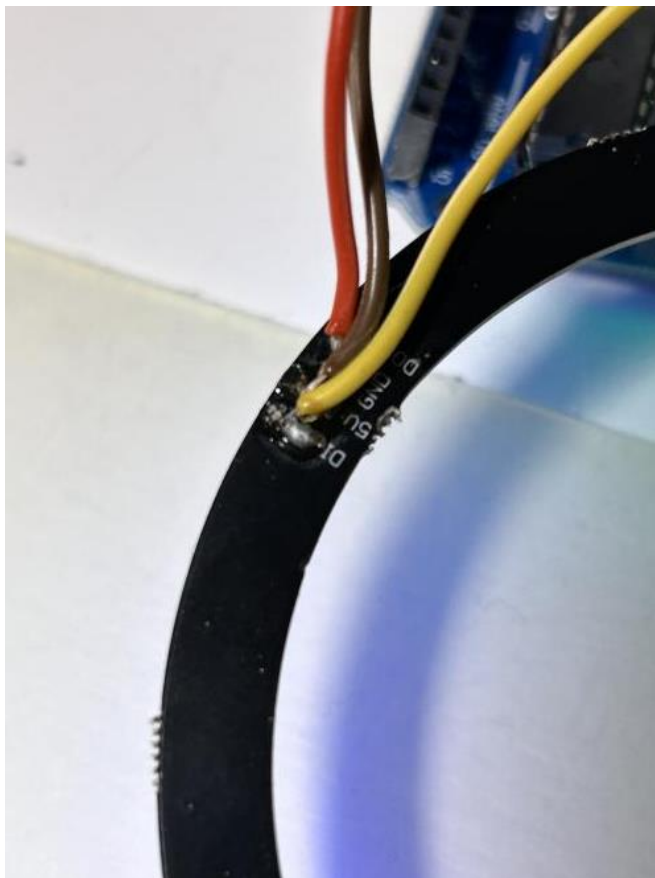
7. Schema circuitului

Schema digitală a circuitului vizualizată înainte de a achiziționa piesele necesare. Aceasta a fost realizată în aplicația Tinkercad, în care am utilizat un senzor de temperatură în loc de un senzor de sunet, deoarece programul nu oferea această posibilitate



8. Dificultati intampinate

- Lipirea cablurilor de Neopixel ring, deoarece ledcon-ul folosit era puțin vechi, îngreunând utilizarea acestuia.





9. Anexe

- Prima varianta a codului

```
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel strip(24, 4, NEO_GRB + NEO_KHZ800);

void setup() {

    strip.begin();

    strip.show();

    strip.setBrightness(255);

}

void loop() {

    int i;
```



```
for(long f_hue = 0; f_hue < 65537; f_hue += 500)

{ for( i = 0; i < 24; i++)

{

    strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue + i*2730,
255, 50 )));

    if(analogRead(A0)<10)strip.setPixelColor(i,
strip.gamma32(strip.ColorHSV(f_hue +i*2730, 255, 200 )));

}

strip.show();

delay(50);

}}
```



- A doua varianta a codului

```
#include <Adafruit_NeoPixel.h>
```

```
Adafruit_NeoPixel strip(24, 4, NEO_GRB + NEO_KHZ800);
```

```
void setup() {
```

```
    strip.begin();
```

```
    strip.show();
```

```
    strip.setBrightness(255);
```

```
}
```

```
int j = 0, j1 = -1, j2 = -1, ok = 0;
```

```
void loop() {
```

```
    int i;
```

```
    for(long f_hue = 0; f_hue < 65537; f_hue += 500)
```

```
    { for( i = 0; i < 24 && ok; i++)
```

```
        {
```

```
            if(j1 != i && j2 != i && j2-1 != i && analogRead(A0)>10)
```

```
                strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue +i*2730, 255, 50 )));
```




```
if(analogRead(A0)<10 && ok)
{
    strip.setPixelColor(j, 250, 250, 250);
    strip.setPixelColor(j1, 20, 20, 20);
    strip.setPixelColor(j2, 10, 10, 10);

    j++;
    j1 = j-1; j2 = j-2;

    if(j == 24 ) { j = 0; j1 = 23; j2 = 22;}
    if(j==1) { j1 = 0; j2 = 23;}

}

}

strip.show();

delay(25);
```



```
if(!ok)
    for( i = 0; i < 24; i++)
    {
        strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(f_hue +i*2730, 255,
50 )));
        strip.show();
        delay(50);
        if(i == 23) ok = 1;
    }

}

}
```



10. Bibliografie

11. <https://www.hackerearth.com/blog/developers/arduino-programming-for-beginners/>
12. https://www.optimusdigital.ro/ro/optoelectronice-altele/5623-inel-cu-24-led-uri-rgb-adresabile-ws2812.html?search_query=inel&results=61
13. <https://create.arduino.cc/projecthub/electropeak/how-to-use-ky-037-sound-detection-sensor-with-arduino-a757a7>
14. <https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use>
15. <https://forum.arduino.cc/t/serial-print-how-do-i-do-strings-and-variables-all-in-one-line/167239>
16. <https://www.arduino.cc/reference/en/libraries/colorconverter/>
17. <https://www.tinkercad.com/dashboard>
18. <https://forums.adafruit.com/viewtopic.php?f=8&t=73593>