

Audio Spectrum Visualizer

Autor: Vlad-Alexandru Proteasa

Introducere

Proiectul consta in realizarea unui dispozitiv care va prelua inputul audio si va genera un output vizual pe mai multe benzi led. Numarul de leduri si culoarea de pe fiecare banda va depinde de frecventele prezente in semnalul audio receptionat.

Descriere generala

- Pentru a prelua inputul audio poate sa fie folosit un microfon care preia semnalul sonor din camera sau un cablu jack conectat la un dispozitiv care poate transmite datele necesare.
- Semnalul este preluat de circuitul integrat MSGEQ7 si transmite datele prelucrate catre placuta Arduino UNO, atunci cand este semnalizat sa faca acest lucru.
- La placuta sunt conectate sapte benzi led, fiecare cu cate opt led-uri rgb. Fiecare banda corespunde unei anumite frecvente din semnalul audio prelucrat. Numarul de leduri de pe fiecare banda care o sa se aprinda si culoare lor depinde de valoare returnata de circuit.
- Utilizatorul are la dispozitie mai multe butoane prin care poate configura output-ul vizual:
 - modifica culoare ledurilor
 - sensibilitatea (cat de usor se aprind toate ledurile de pe banda)
 - gama de culori folosita
 - revenirea la setarile initiale

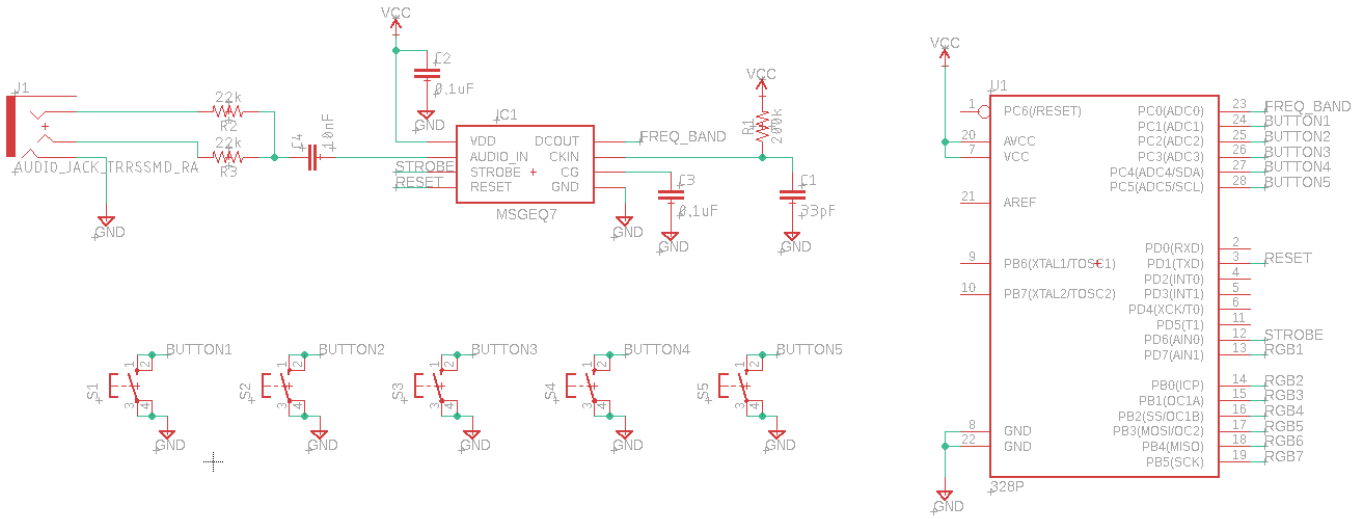


Hardware Design

Lista componente

1. Arduino UNO
2. MSGEQ7
3. Modul microfon
4. Jack input (CJMCU-TRRS)
5. 7 Benzi NeoPixel
6. Rezistori: 1K Ω (6), 10K Ω (6), 56K Ω (3)
7. Condensatori: 33pF (1), 0.1uF (3)
8. Breadboard
9. Fire jumper
10. Cablu audio jack 3.5mm

Schema Electrica



Software Design

Codul sursa esti disponibil aici: [GitHub](#)

Dependinte

Singura dependinta este biblioteca [Adafruit_NeoPixel](#) folosita pentru a controla benzile led adresabile folosite in implementarea hardware a proiectului.

Setup initial

- Seteaza pinii de output pentru leduri si initializeaza vectorul folosit pentru a controla culoriile si luminozitatea benzilor led.
- Sunt initializate variabilele pentru a controla setarile dispozitivului.
- Se activeaza intreruperea pentru unul dintre butoane, celelalte raman doar pentru input. Input-ul este interpretat mai tarziu in functia apelata atunci cand se genereaza intreruperea

Intreruperi

Am configurat o intrerupere pentru butonul din marginea din stanga. Utilizatorul poate sa tina apasate oricare din celelalte patru butoane. Atunci cand butonul asociat intreruperii este apasat input-ul este citit si interpretat. Functia actioneaza asemanator cu un switch case. In functie de combinatia de butoane apasata setarile dispozitivului vor fi modificate asa cum este descris in diagrama de pe acest [link](#).

Functia de generare a culorilor

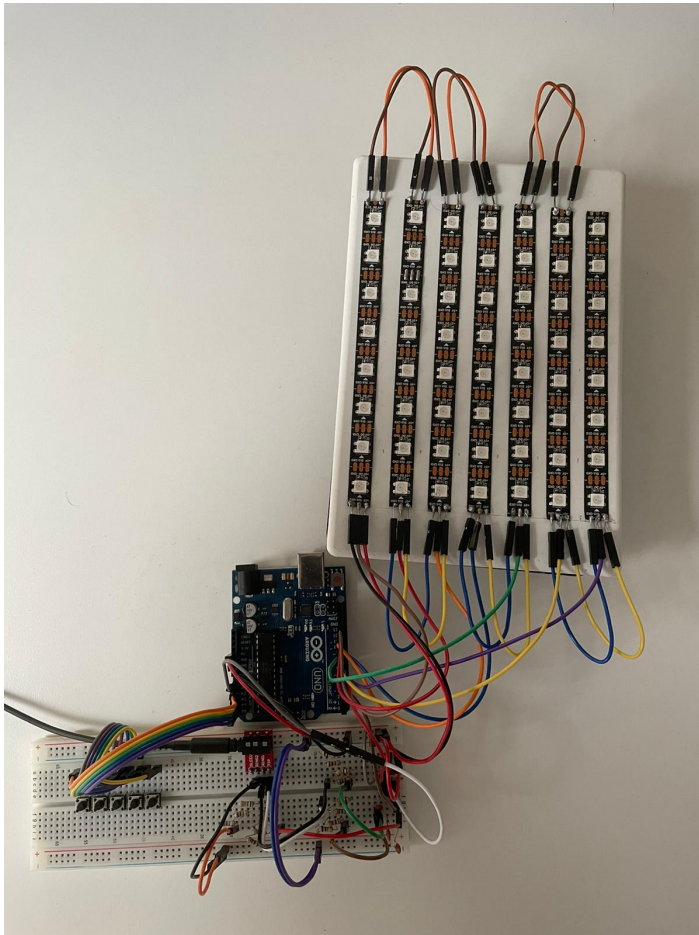
Pentru reprezentarea culorilor am folosit modelul RGB. Functia primeste doi parametri: banda led pe care se aplica modificarile si valoarea returnata de circuitul integrat. Daca indexul ledului este mai mic decat al doilea parametru atunci o sa fie calculata valorile pentru rosu, verde si albastru in functie de setarile specificate de utilizator si index. Led-ul care are index-ul egal cu valoarea celui de-al doilea parametru o sa aiba culoarea alba sau o sa fie complet stins. Toate celelalte leduri o sa fie stinse.

Main Loop

- Semnalizez circuitul integrat MSGEQ7 ca o sa inceapa un nou ciclu de update a valorilor afisate.
- Citesc valoarea returnata pe portul A0.
- Verific daca aceasta este o valoare valida sau zgomot. Aceste valori variaza intre 0 si 1024. In urma testelor am observat ca pentru sistemul meu cea mai buna alegere a fost sa consider zgomot orice valoare mai mica decat 200 (deoarece cablul nu este conectat direct la placa de sunet a calculatorului si este preluat zgomot de la celelalte componente). Aceasta valoare poate sa fie configurata manual prin utilizarea butoanelor.

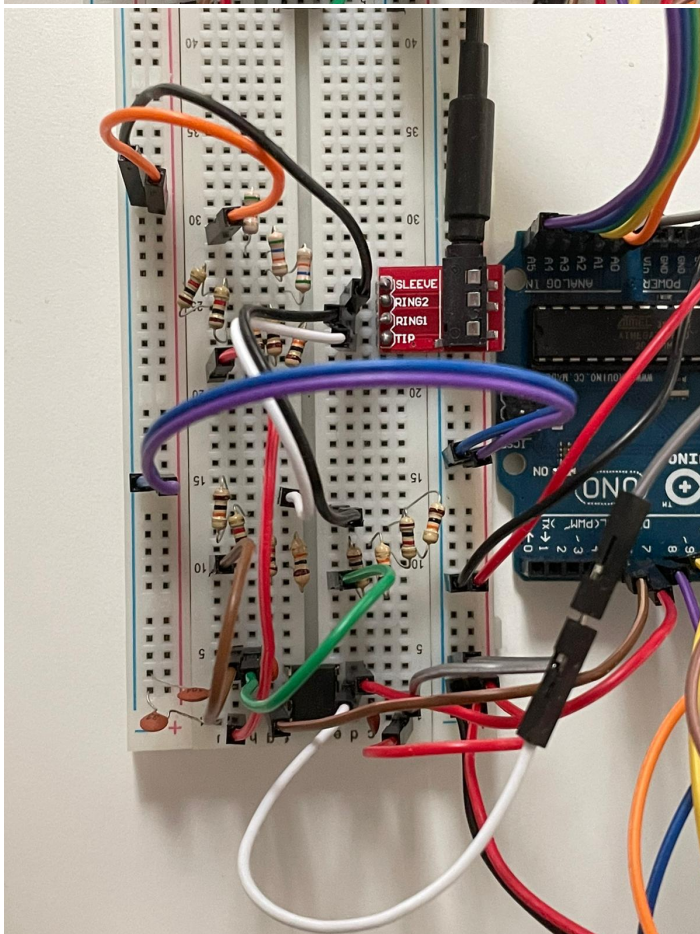
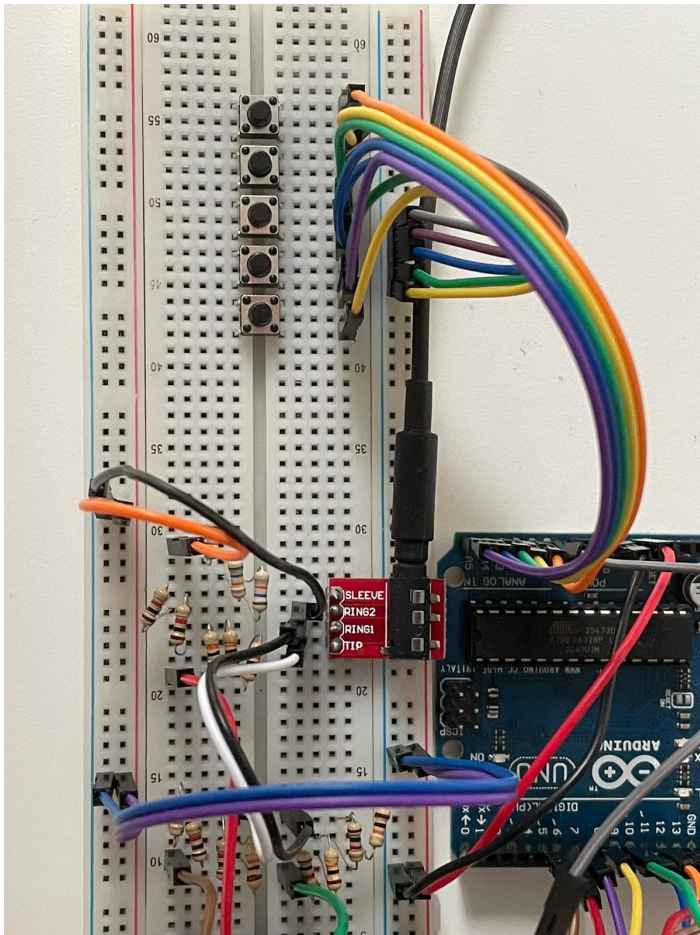
Rezultate Obtinute

In urmatoarele doua poze este prezentata implementarea hardware:



Breadboard-ul pe care se afla butoanele pentru setari, mufa jack pentru conectarea cu o sursa de la

care se preia inputul audio si circuitul MSGEQ7:



Folosind butoanele de pe breadboard pot fi ajustate urmatoarele setari:

- Luminozitatea
- Culoarea
- Led-ul alb din varful fiecarei benzi on/off
- Ajustarea filtrului de zgomot
- Sensitivitatea
- Revenire la setarile initiale

Inregistrari video in care este demonstrata functionalitatea:

1. [Testare frecvente 0-20.000Hz](#)
2. [Test Martin Garrix - Animals](#)
3. [Ajustare setari](#)

Jurnal

- 25 aprilie - alegere tema proiect si componente necesare
- 4 mai - sosesc componentele
- 8 mai - testarea componentelor
- 11 mai - implementare hardware
- 14 mai - implementare software

Bibliografie/Resurse

1. [Export to PDF](#)
2. <https://www.sparkfun.com/datasheets/Components/General/MSGEQ7.pdf>
3. <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>
4. https://ardushop.ro/ro/home/1314-modul-microfon-senzor-sunet.html?gclid=CjwKCAjwhYOFBhBkEiwASF3KGfQ1NW1dzJCDF-H5T_eVcdyS1Gfm6zImNeJkqesQr1wOgd7uVjSXDxoCIYEQAvD_BwE
5. https://ardushop.ro/ro/home/686-jack-audio-stereo-cjmcu-trrs-35mm.html?search_query=trrs&results=1

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/audio_spectrum_visualizer



Last update: **2021/06/02 08:59**