

Music Visualizer

Introducere

Nume autor: Grigorescu Ana-Maria

Grupa: 332CC

Ce face?

Music visualizer-ul reprezinta un dispozitiv prin care putem vedea exact frecventa/ritmul melodiei pe care o ascultam in momentul respectiv, prin intermediul jocului de lumini al matricei de led-uri. De asemenea, device-ul va avea si un vu-meter care va arata printr-un lcd audio level-ul melodiei, in dBu.

Scopul

Scopul este, din punctul meu de vedere, unul estetic (mereu mi s-a parut interesant sa ma uit cum se misca luminile pe ritmul muzicii), dar si informativ (nivelul de db afisat).

Ideea + Utilitatea

Ideea mi-a venit de la boxe de tipul celei JBL Pulse, sau a programelor online prin care se poate vizualiza spectrul frecventelor atinse. Mereu mi-au atras atentia. Din punct de vedere estetic/decorativ mi se pare ca schimba toata experienta de a asculta muzica.

Descriere generală

Odata ce placuta Arduino UNO R3 este econectata la alimentare, microfonul capteaza undele sonore din apropierea sa (nu merge de la distanta intrucat modulul audio nu este unul profesional). In acest timp, matricea de led-uri incepe sa se miste in functie de intensitatea muzicii. Pe ecranul LCD este afisat nivelul de dB, iar la apasarea butonului, LCD-ul va afisa nivelul de presiune al sunetului (SPL-ul).

SCHEMA BLOC:



Varianta de baza a schemei.

Hardware Design

LISTA DE PIESE:

- 1 x 3.5 mm Stereo Audio Jack Module; -> modificat in modul microfon, senzor sunet
- 1 x Breadboard HQ + 1 x Breadboard mini;
- DuPont wires (male-male, male-female, female-female);
- 1 x LCD I2C 16x2;
- 2 x LED Dot Matrix Module;
- 1 x Arduino Uno R3;
- 1 x Button.

Am realizat in Fritzing schema proiectului (varianta inainte de a fi pusa in cutie). Nu a fost nevoie sa folosesc rezistente sau condensatori pana la urma, iar microfonul am observat ca se comporta mai bine atunci cand e conectat la sursa de 3V3 a UNO-ului.

SCHEMA CIRCUIT:



Fisierul .fzz se va putea gasi in arhiva proiectului.

Software Design

Mediul de dezvoltare utilizat pentru proiect a fost Arduino IDE. Pentru a folosi modulele Arduino precizate in componente, am folosit urmatoarele biblioteci, importate prin Library Manager:

- pentru matricea MAX7219 : LedControl;
- pentru ecranul LCD 1602 x I2C : LiquidCrystal_I2C.

Initial am intentionat sa folosesc FFT (**F**ast **F**ourier **T**ransform) pentru a interpreta input-ul audio. Dupa aproape doua saptamani de incercari si apelatul la asistentul de laborator, am concluzionat amandoi ca se misca mult prea greu si nu face ce trebuie niciuna din bibliotecile importate, cel putin in combinatie cu microfonul si toata configuratia mea. In aceste circumstante, am convenit cu Bogdan sa interpretez in mod liber nivelul sonor captat de microfon si jocul de lumini generat de matrice.

Dupa multiple teste in care am pus o boxa langa microfon si am pus muzica la un volum divers (urmarind la Serial Monitor input-ul de la microfon) am dedus niste valori intre care pot incadra numarul de led-uri aprinse pe o coloana. Adaug random si niste valori diferite pentru a oferi un efect vizual mai placut, la o perioada tot random de timp. Toate acestea se intampla in functia **displayLed**.

In **loop** tratez apasarea de buton care imi schimba afisajul de pe LCD din dB Volts (cei uzuali) in dB SPL (**S**ound **P**ressure **L**evel). Tratez si situatia de debounce, masurand timpul dintre apasarea actuala si cea anterioara, punand restrictia de a fi > 500ms. Fac un loop de dimensiunea inputWindow-ului

pentru a capta muzica, afland minimul maximul si media. Urmeaza partea de calcule matematice pentru a obtine dB-ii. Codul este segmentat prin comentarii pentru lizibilitate.

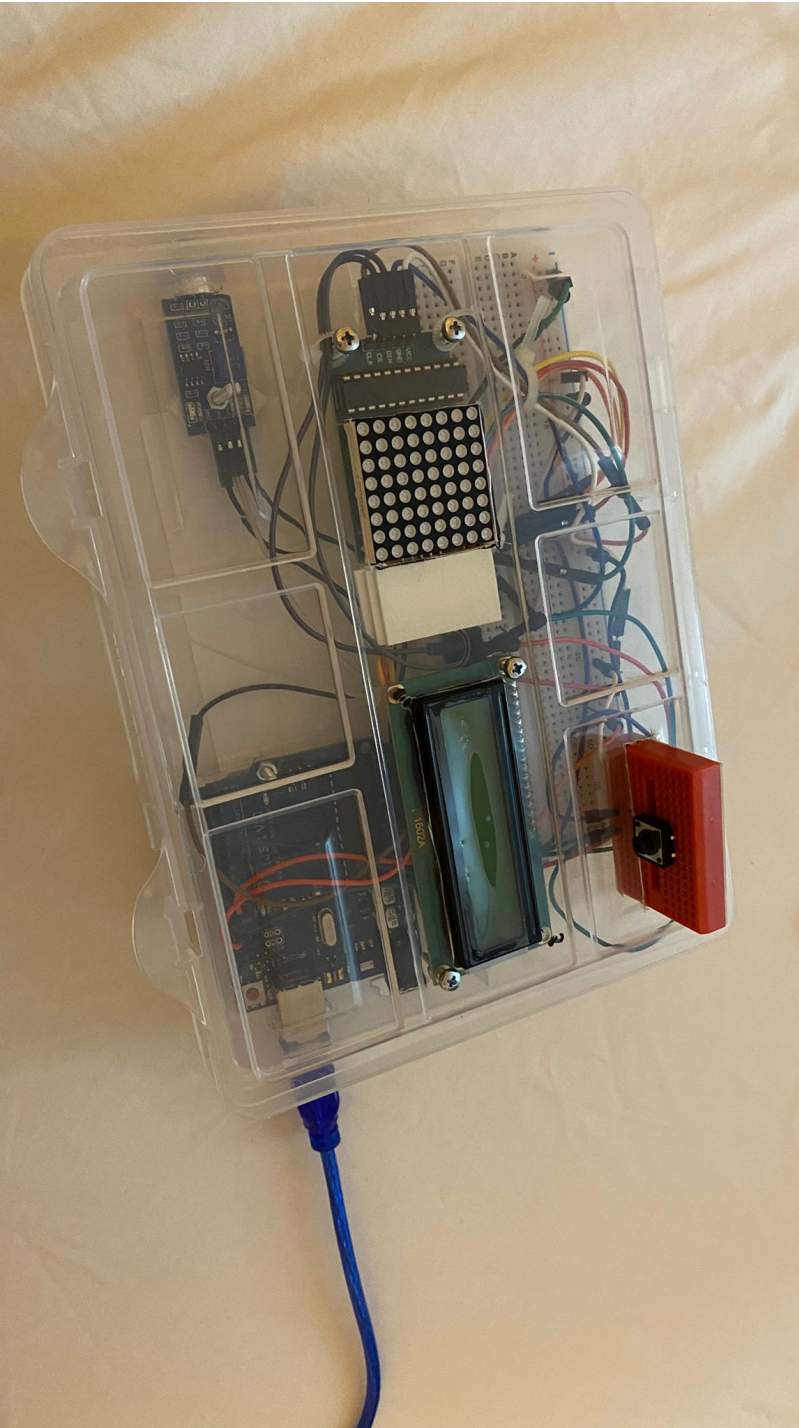
Rezultate Obținute

Proiectul functioneaza cum trebuie (desi microfonul lasa de dorit, insa dupa achizitionarea a trei modele diferite mi-am acceptat soarta), cu restrictia ca sursa de muzica (boxa) trebuie pozitionata langa microfon. De asemenea, microfonul nu poate capta sonor mult prea puternic, volumul maxim la care poate ajunge fiind in jur de 86dB (o situatie de genul 3.6, not great not terrible, m-am adaptat la neajunsurile modulului sonor). Restul modulelor functioneaza foarte bine. O sa atasez

Link demo (before mutat in cutie) : <https://youtube.com/shorts/3r7MCkrnHD8?feature=share>

Link demo (pus ordonat in cutie) : <https://youtube.com/shorts/7NgdvLogclQ?feature=share>

Pentru a sta compact si a arata mai bine, am pus proiectul intr-o cutie de plastic, in care am masurat, taiat, si prins in suruburi componentele (cu tata :), plasticul e mult mai tare decat pare). Am prins firele cu zip-ties si lipit in interiorul cutiei pentru a sta ordonate. *Probabil* voi infasura cutia in banda adeziva pentru a nu mai fi transparenta, insa adaug poze cu ea in aceasta varianta semi-gata pentru a se vedea clar ce si cum:



Jurnal

- **12.05.2022:** Created wiki page, completed introduction, block diagram, components and supplies;
- **13-15.05.2022:** Research amanuntit, atat despre proiect cat si componentele sale.
- **16-20.05.2022:** Incercari folosind modul Jack -> renuntare la acesta deoarece nu functiona corespunzator;
- **22-27.05.2022:** Trecut la utilizarea microfonului, schimbat microfonul deoarece primul nu functiona cum trebuie, renuntat la utilizarea FFT dupa sfatuirea cu Bogdan;
- **27-28.05.2022:** Finalizat proiect;
- **29.05.2022:** Mutat proiect in cutie si ordonarea firelor + finalizare documentatie

Bibliografie/Resurse

Resurse Hardware:

- [Datasheet matrice de led-uri \(MAX7219\)](#);
- [Datasheet LCD 1602 cu I2C](#);
- [Datasheet Arduino Uno R3](#);
- Modul microfon -asemanator cu KY-037 (nu am gasit datasheet-ul, m-am folosit de descrierea de pe site-ul de unde l-am luat).

Resurse Software:

- Inspiratie initiala proiect: [Visualizer](#), [vu-meter](#), [alt visualizer cu FFT](#) :(;
- [Documentatie biblioteca LedControl](#);
- [Tutorial LCD](#).

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2022/bogdanc/music-visualizer>



Last update: **2022/05/30 19:04**