

# Vizualizator de muzică

Autor : Tudorache Alina  
Grupa : 336CB

## Introducere

- **Ce face ?**
- Afiseaza pe ecranul lcd frecventele identificate in fisierul audio care este redat
- **Care este scopul lui ?**
- Un mod placut de a reda si vizualiza muzica redata.
- **Care a fost ideea de la care ați pornit ?**
- Utilizez de obicei aplicatie Android de tip vizualizator pentru a putea reda fisiere audio, in principal muzica. Nu am gasit sa cumpar un dispozitiv de acest tip, asa ca am decis sa construiesc unul.
- **De ce credeți că este util pentru alții și pentru voi ?**
- Exista destul de multi utilizatori care folosesc un vizualizator de muzica si probabil ar dori sa detina un astfel de device in casa.

## Descriere generală



## Hardware Design

Componente

Nume Piesa	Cantitate
Display OLED 128x64 - 0.96" SPI	1
Arduino UNO	1
Breadboard	1
Buton	1
Microfon	1
Fire	11



## Software Design

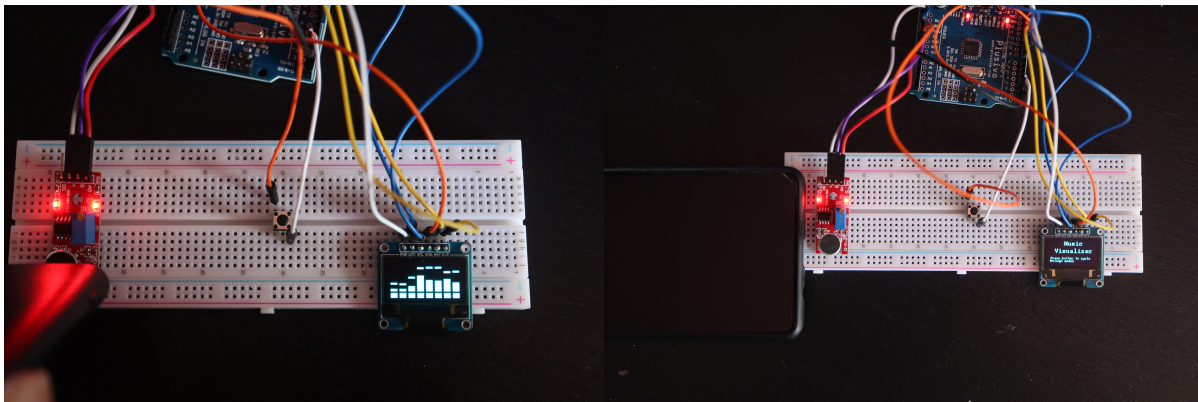
In crearea vizualizatorului am folosit :

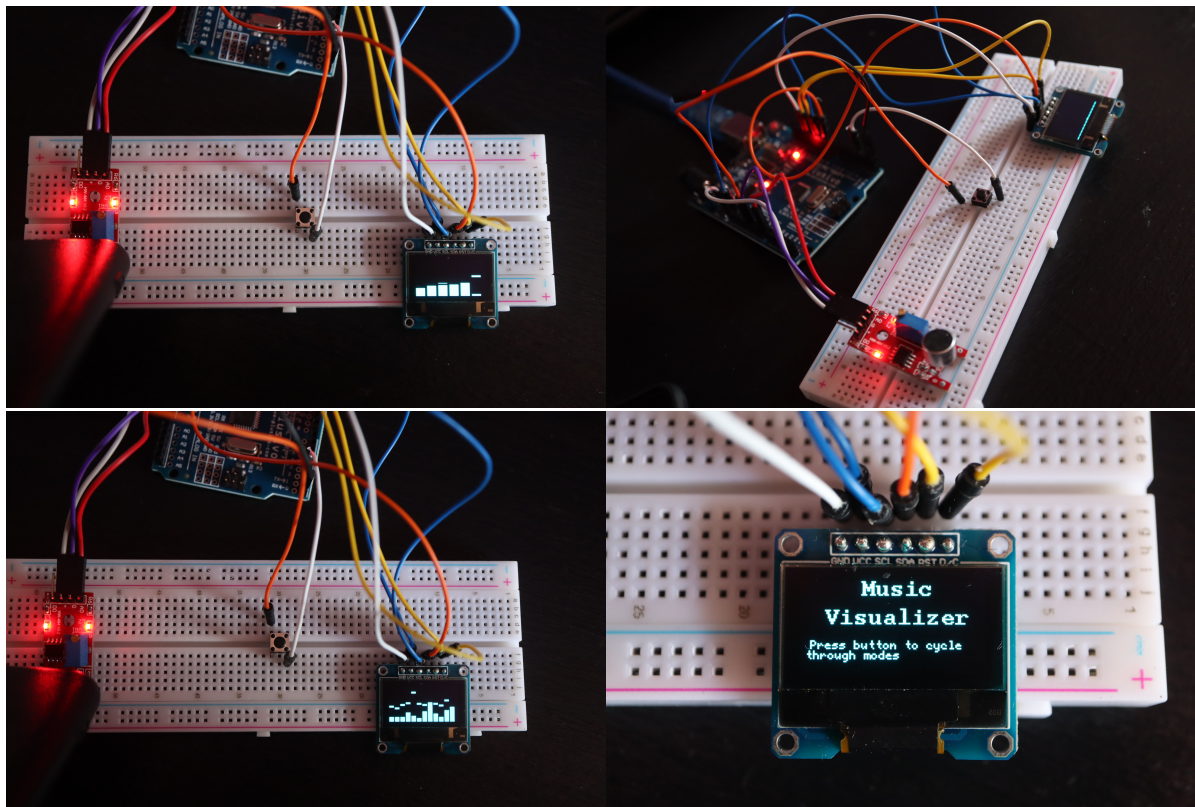
- **Arduino IDE** - pentru scrierea si testarea codului
- **Biblioteca <U8glib.h>** - pentru afisarea pe display
- **Biblioteca <PinChangeInt.h>** - pentru a detecta schimbarea de mod la apasarea butonului
- **Biblioteca <FHT.h>** - pentru a prelucra inputul audio, care este mai rapida ca FFT

Am definit toti pinii folositi de catre program, cat si obiectul de tip u8g (conform tipului de display folosit) si variabilele necesare pentru implementarea programului. Se preiau 256 de sample-uri pentru a putea obtine rezultate cat mai exacte la aplicarea FHT pe ele si apoi afisarea rezultatelor obtinute sub forma de blocuri in modul selectat in functie de frecventele identificate.

## Rezultate Obținute

Produsul obtinut este un vizualizator de muzica cu 3 moduri posibile - 6, 8 sau 12 segmente






Video Demo : <https://youtu.be/eXeO9W5kKG0>

## Concluzii

A fost destul de dificil, intrucat nu stiam exact cum as putea prelua inputul audio si sa il prelucrez pentru a putea obtine o reprezentare grafica. Initial am incercat bibliotecile <Adafruit\_SSD1306.h> impreuna cu <Adafruit\_GFX.h> pentru display si FFT pentru prelucrarea inputului audio, dar am descoperit ca este mai usor cu cele mentionate anterior (chiar si mai eficient). In plus, am descoperit o biblioteca noua <PinChangeInt.h>, care simplifica procesul de a trece de la un mod la altul prin apasarea butonului, folosind intreruperi.

Dupa ce am citit documentatia bibliotecilor, am inteles ce am de facut si proiectul a iesit destul de repede. A fost o experienta foarte placuta (mai ales ca am si ars cate ceva in proces)  Sunt multumita de cum am iesit, nu ma asteptam.

## Download

Implementarea folosita in cadrul proiectului : [vizualizator\\_muzica.zip](#)

## Bibliografie/Resurse

- [Fast Hartley Transformation Library for AVR microcontrollers](#)
- [U8glib](#)
- [PinChangeInt](#)
- [arduinoFFT library](#)
- [Tutorial folosire display OLED SPI](#)
- [Arduino Audio Input](#)
- [Tutorial folosire microfon](#)
- [Laboratorul 0](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/amocanu/ioana.tudorache2507>



Last update: **2022/05/23 20:56**