

# Audio Spectrum Analyzer

## Introducere

Partea principala a proiectului consta in preluarea frecventelor audio din jur si afisarea lor pe o matrice de led-uri, oferind si alte moduri de afisare precum temperatura din mediul ambiental, umiditatea, ora curenta, dar si anumite jocuri de lumini predefinite.

Interschimbarea se face prin apasarea singurului buton de pe carcasa.

De asemenea luminozitatea matricii este setata automat in functie de mediu in care functioneaza proiectul, in zone mai intunecate aceasta fiind mult redusa pentru a nu deveni deranjanta.

Conceptul pentru acest proiect a plecat de la dorinta mea de a adauga un feature nou televizorului, initial folosind o banda de LED-uri ambientala de unde a pornit si ideea de a amplasa sub acesta pe perete (langa difuzoare) dispozitivul, avand astfel un input cat mai bun pentru microfon.

## Descriere generală

Doua moduri de functionare:

1. Idle - sunt afisate anumite modele / informatii (PacMan, Mesaje, Temperatura, Umiditate, Ora)
2. Receptionare frecvente - se afiseaza pe matrice frecventele captate de microfon sub forma de armonice

Exista un singur buton de modificare al modului care cicleaza prin acestea.



## Hardware Design

Lista componente:

- Arduino UNO
- Matrice LED-uri 8\*32 - MAX7219
- Modul microfon - LM393
- Modul senzor temperatura - DHT11
- Modul RTC - DS1302
- Buton

- Rezistentă 10k \* 2
- Fotorezistor
- Cablu alimentare
- Adaptor priză 5V

## Software Design

MAX7219 Matrice LED-uri 32×8:

- Pentru acest modul este disponibilă o varietate largă de biblioteci capabile să ofere posibilitatea de a afișa diverse imagini pe acesta.
- Pentru afișajul principal (Audio Spectrum Analyzer) am folosit biblioteca “MD\_MAX72xx” care îmi permite controlarea individuală a coloanelor, având astfel posibilitatea de a seta fiecare coloană a matricei pe verticală în funcție de frecvențele captate de microfon. De asemenea pentru a avea un rezultat cât mai plăcut vizual am folosit “arduinoFFT” care preia semnalele și stabilește armonicile corespunzătoare.
- Pentru celelalte moduri de funcționare am folosit “MD\_Parola” care oferă funcții mult mai potrivite și mai ușor de folosit pentru jocuri de lumini decât biblioteca menționată anterior, dar nu este de o flexibilitate foarte mare de aceea nu am folosit-o și la modul de la punctul anterior.
- Pentru a nu seta manual fiecare led pe o coloană am folosit un vector pentru fiecare amplitudine posibilă.

Luminozitate reglată automat:

- Pentru acest feature m-am bazat pe conceptul de ADC învățat la laborator, folosind un fotorezistor asupra căruia am aplicat un factor de “smoothing”.

Schimbare moduri afișare:

- Am folosit o simplă întrerupere cuplată la PIN-ul 2 de pe Arduino pe care am făcut un debouncer în software.

Moduri secundare de afișare:

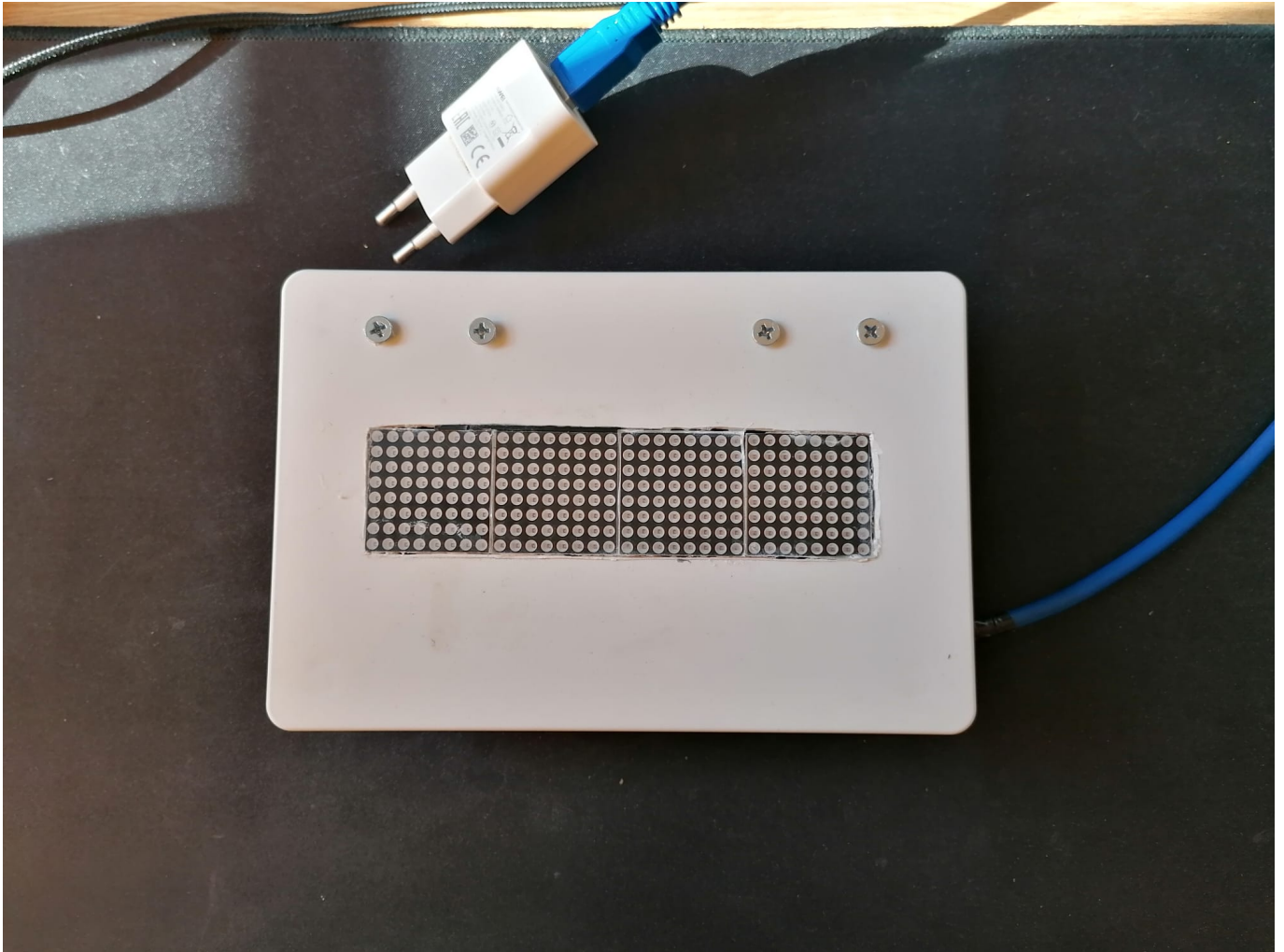
- Pentru modurile care nu implică “VU Metrul” am preluat datele de la senzori prin bibliotecile corespunzătoare (ex. sezon și temperatura → “DHT”), iar pentru jocurile de lumini am folosit conceptul de “SpriteData” dat în MD\_Parola.
- La afișare modelul l-am pastrat simplu cu PA\_CENTER pentru a avea totul centrat pe matrice.

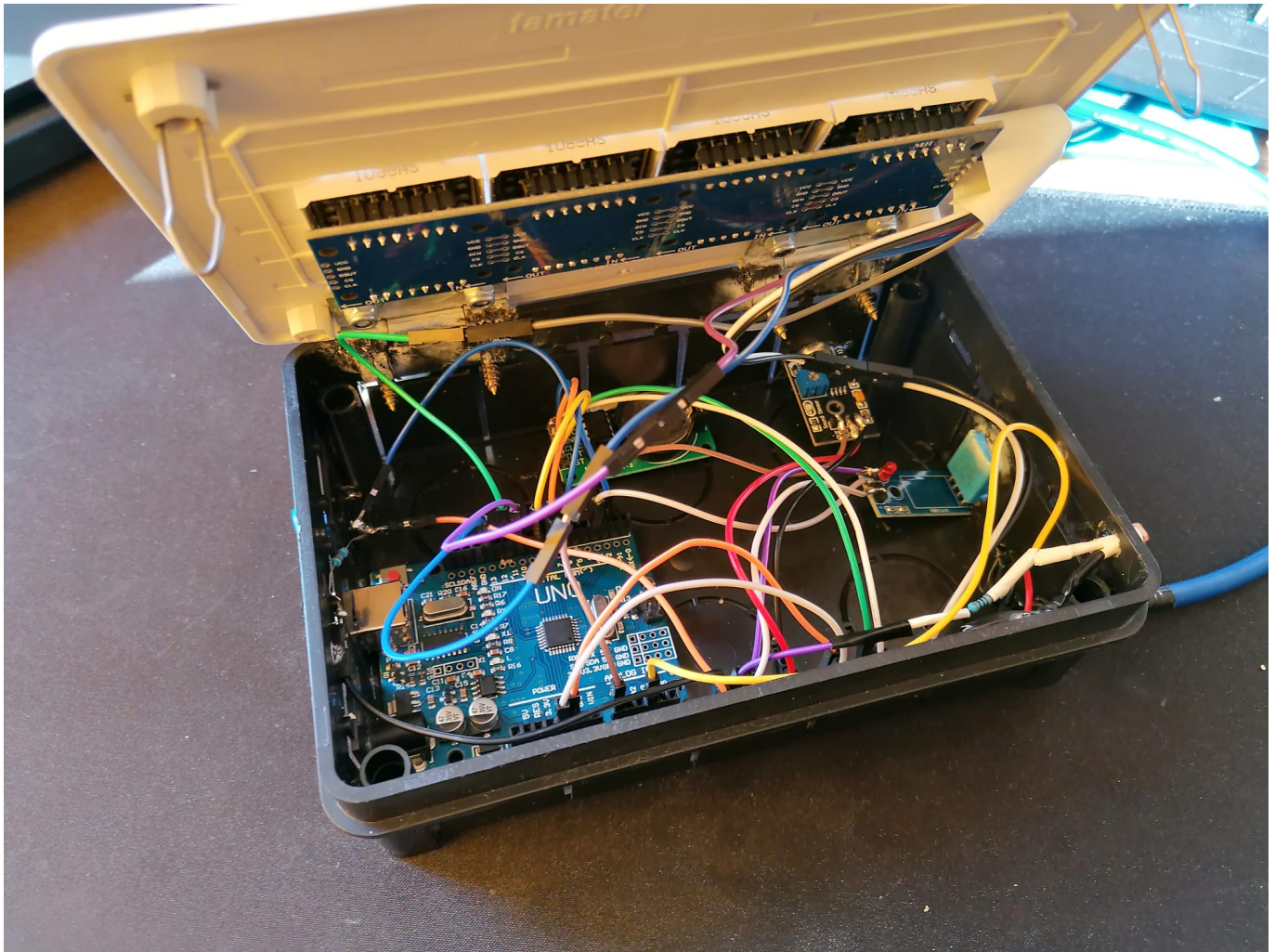
Optimizări:

- Având un număr destul de mare de biblioteci și o memorie limitată pe Arduino acest lucru a fost o problemă din start.
- Jocurile de lumini se bazează pe matrici care setează starea ledurilor, matrici care pot ocupa destulă memorie de aceea am folosit cuvântul cheie “PROGMEM” care îndrumă compilatorul să pună aceste date pe memoria flash, nu pe SRAM unde ar fi fost puse.
- Mi-aș fi dorit să pot renunța la una din bibliotecile pentru MAX7219 deoarece ocupă destulă memorie, dar MD\_Parola este dependentă de MD\_MAX72xx ceea ce mi-a îngreuna procesul de implementare.
- Memoria utilizată în momentul actual este de 86%.

## Rezultate Obținute

Am reușit să implementez absolut tot ce mi-am propus, inclusiv o carcasă funcțională pentru acesta. Pe lângă conceptul inițial am adăugat și feature-ul de reglare automată a luminozității.





## Concluzii

A fost un proiect care in primul rand chiar mi-a placut sa-l realizez, desi a fost destul de multa munca, dorind sa aiba si atat o utilitate cat si un aspect decent. Cu siguranta se pot face imbunatatiri la acesta (un RTC mai precis precum DS3231, o carcasa printata 3D etc.).

Ca si timp de implementare majoritatea a fost ocupat de research pentru a nu strica anumite piese \*din nou\* si pentru a nu pierde timpul cu incercari inutile, in sine asamblarea hardware + design nefiind atat de costisitoare ca timp.

In schimb pentru carcasa/lipit/asezat fire timpul alocat a fost destul de mult incat sa fie ceva cat de cat rezistent.

Sincer nu ma asteptam sa-mi placa sa lucrez atat de multe ore la proiectul acesta, dar tot am ajuns la concluzia ca imi place sa dorm noaptea in loc sa dezizolez/lipesc fire.

## Download

Link repository github:

<https://github.com/AlexANDrei33/PM-project-VU-Metru.git>

## Bibliografie/Resurse

Mai jos se afla link-uri catre cele mai frecventate site-uri pentru acest proiect.

<https://www.instructables.com/>

<https://docs.arduino.cc/>

<https://forum.arduino.cc/>

<https://how2electronics.com/>

<https://lastminuteengineers.com/>

<https://projecthub.arduino.cc/abhilashpatel121/f1b6ba36-bd96-4f7f-9a83-85c4925ff545>

Download: [Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/gpatru/audiospectrum>



Last update: **2023/05/29 17:09**