

# Music Pad

**Autor:** Bocanu Raul-Teodor

**Grupa:** 336CB

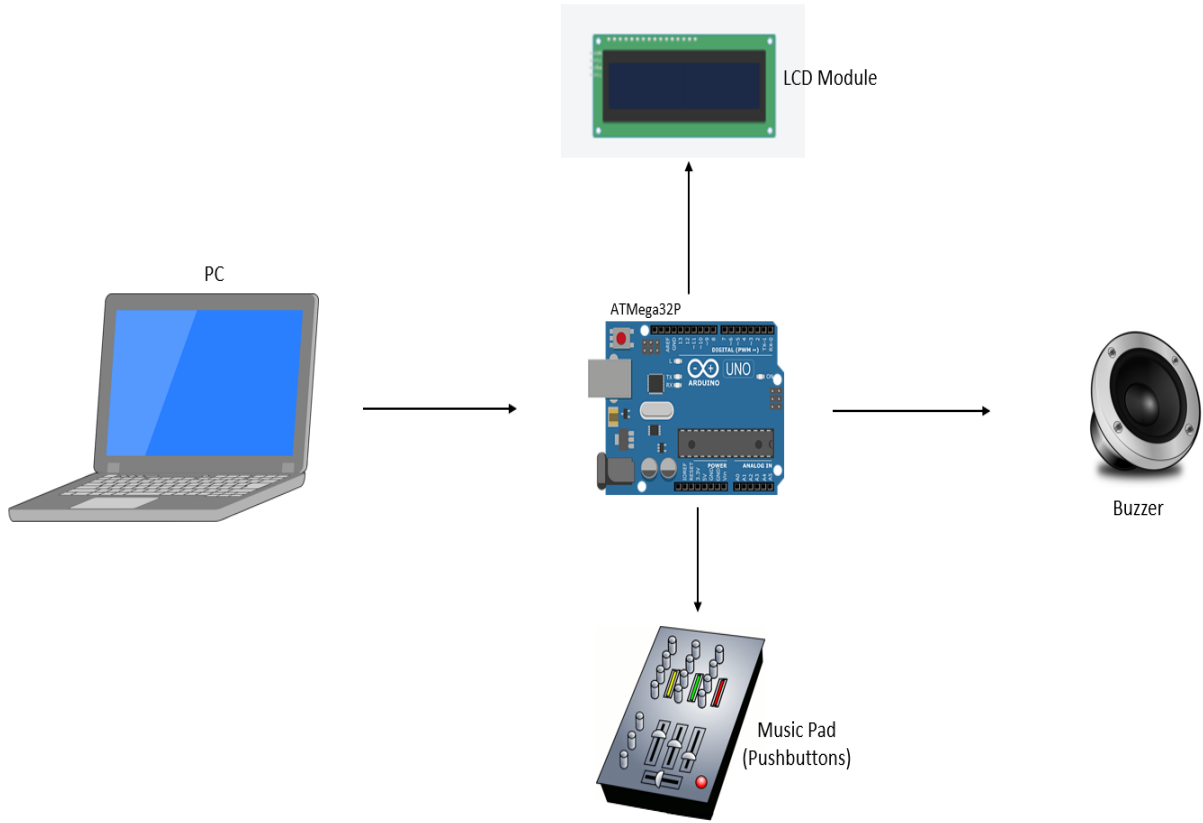
## Introducere

✘ Muzica este cea mai cunoscuta si folosita forma de recreere ce dateaza inca de acum cca. 4,000 ani. Prima forma a unui fragment muzical a fost descoperita pe o tableta din Sumer. De atunci, suntem inconjurati de muzica in toate formele, generatia noastra fiind una care este prezenta la festivaluri, concerte si in fiecare vineri in cluburi.

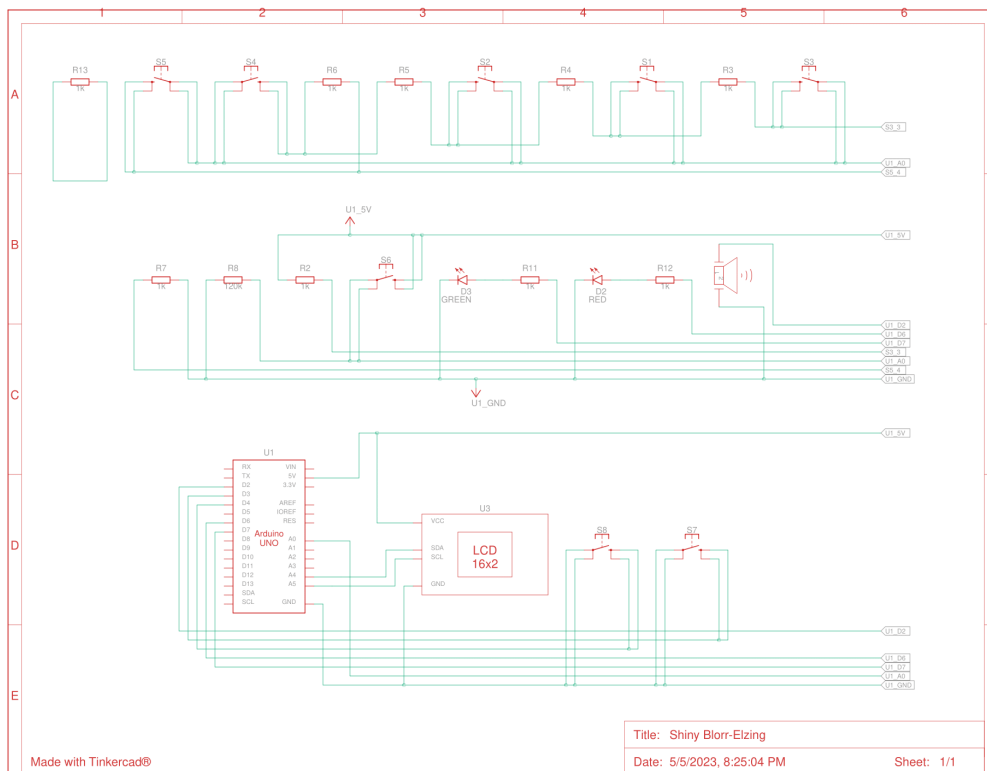
O inovatie recenta in muzica sunt MusicPad-urile. Cum si eu, la randul meu, sunt pasionat de muzica, am decis sa creez un proiect interactiv, dar totodata si amuzant.

## Descriere generală

Proiectul meu va simula un MusicPad simplist, cu 6 tipuri de sunete diferite. Acestea vor fi redade de 6 butoane diferite, conectate in paralel la acelasi pin de pe Arduino. Voi determina ce buton a fost apasat cu ajutorul tensiunii si rezistentelor, tensiune ce va avea un output diferite pentru fiecare buton. Un buzzer va reda sunetele, dar si secventa formata de acestea prin cele doua butoane adiacente, unul pentru functia de "Play", care va reda secventa de pana acum, si inca unul pentru "Reset", care va reseta toata secventa. Cele doua LED-uri vor functiona ca timere, unul verde care va permite apasarea butoanelor, respectiv generarea sunetelor, cat timp este aprins, si unul rosu care va permite doar utilizarea butoanelor de "Play" si "Reset", practic semnaland momentul in care o secventa se opreste si poate fi ascultata. Daca ultima secventa nu va fi resetata, se vor adauga sunete in continuarea acesteia. Prin LCD-ul cu comunicare I2C voi reda ce butoane/sunete au fost redade.

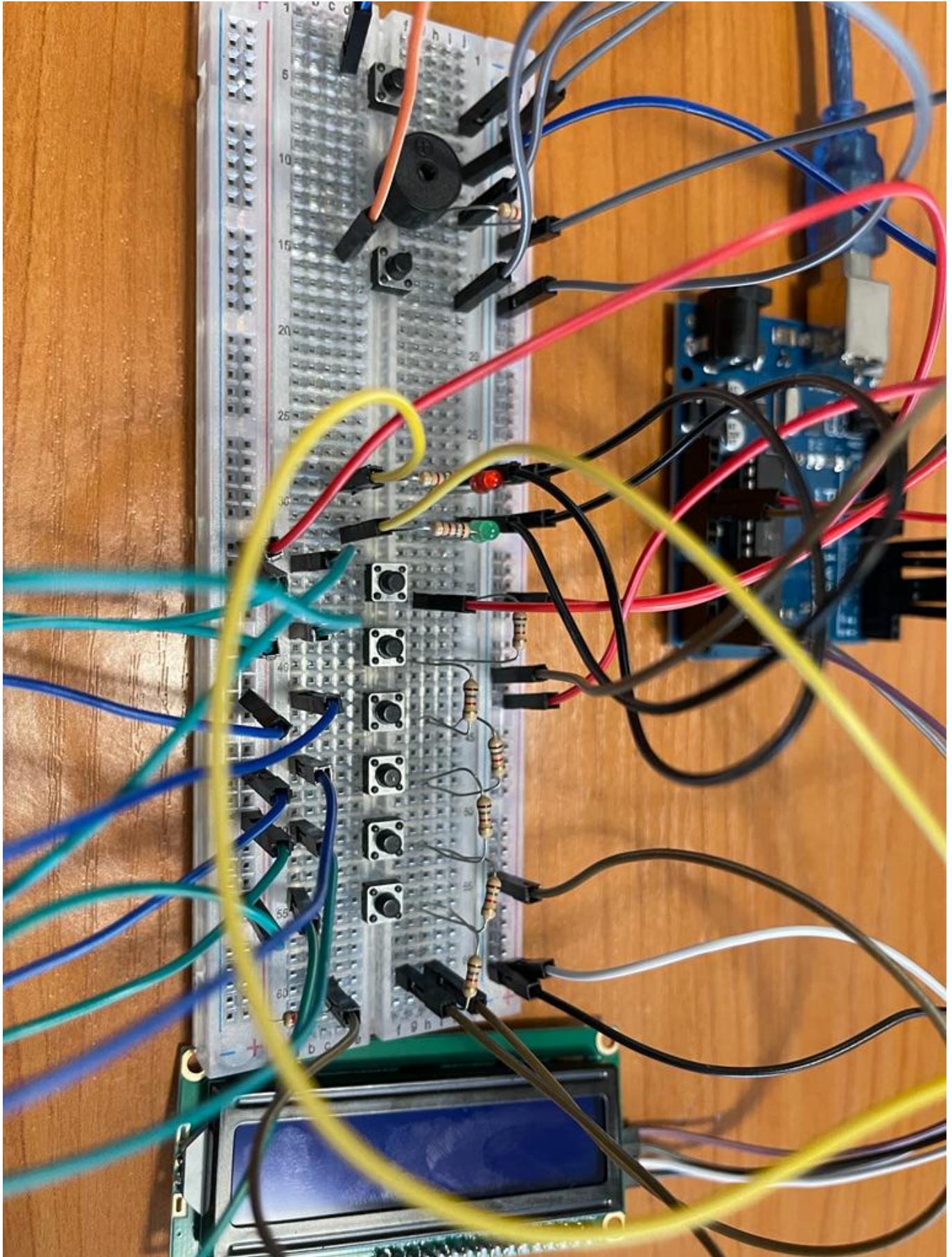


## Hardware Design



Componente folosite:

1. Arduino UNO
2. Breadboard
3. LCD 16x2 I2C
4. 8x Pushbutton
5. 8x Rezistor 1.8kOhm
6. 1x Rezistor 220kOhm
7. Buzzer Pasiv
8. LED Verde
9. LED Rosu
10. Cabluri tata-tata/mama-tata



Pentru a implementa partea de hardware am respectat schema realizata anterior pentru a conecta toate componentele. In partea stanga a breadboard-ului se afla Buzzer-ul si butoanele de "Play" si "Reset". In partea dreapta, avem butoanele pentru redarea sunetelor, inseriate cu o rezistenta de 1.8KOhm, si conectate paralel unul fata de altul. Rezistenta de 220KOhm se afla intre ansamblul de

butoane si pinul A0 de pe placuta Arduino. De asemenea, pentru cele doua led-uri am folosit cate o rezistenta de 1.8KOhm, la fel si pentru Buzzer. Ecranul LCD 16x2 este separat de breadboard, fiind conectat direct la placuta Arduino prin cei 4 pini specifici.

## Software Design

Pentru a putea realiza redarea unor sunete specifice zonei de Beat Making a trebuit sa-mi definesc toate notele muzicale in functie de frecventa pe care acestea o redau. Un sunet va fi compus dintr-un tuplu de forma (nota, durata), astfel am reusit sa-mi definesc **6 sound-uri**, precum **Snare Drum**, **Hi-Hat**, **Bassline** si alte sunete specifice instrumentelor muzicale. Desigur, fiind limitat la folosirea unui Buzzer care poate doar sa redea anumite frecvente, aceste sunete sunt relativ similare intre ele si fac reproducerea unui beat/sound fluent destul de grea.

Pentru fiecare sound a trebuit sa asociez un tempo specific si sa calculez durata unei note care este compusa din doua valori: **pitch** si **duration**. Prin functia **play()** am redat sunetele prin Buzzer, iterand prin toate notele. Aceasta functie primeste un parametru "i" care este asociat pozitiei sound-ului respectiv din array-ul "all-sounds".

Cea mai interesanta parte a proiectului a fost conectare a celor sase butoane specifice celor sase sunete la **un singur pin**. Fiecare buton avand inseriat o rezistenta si fiind conectat la restul butoanelor in paralel va determina o valoare diferita a tensiunii cand este apasat. Aceasta valoare a fost citita de pe pinul analog **A0**. Cu ajutorul functiei **readPushButtons()** si unor calcule destul de elementare de fizica, voi determina ce buton a fost apasat, actualizand cu valoarea "true" la pozitia aferenta din vectorul "pushbuttons\_value". Pentru fiecare buton, am adaugat si o functie de **debounce** pentru a inregistra apasarea unui buton o singura data, chiar daca este tinut apasat in continuare.

Pentru butoanele de **Play** si **Reset**, conectate pe pinii PD3, respectiv PD2, am atasat cate o intrerupere care sa seteze un flag la valoare true cand acestea au fost apasate. Buton-ul de **Play** va reda secventa cumulata pana in acel moment. Daca un buton din cele sase care redau sound-urile este apasat, acel sound va fi automat adaugat in secventa. Daca se doreste stergerea secventei curente si inceperea uneia noi, acest lucru se realizeaza prin butonul de **Reset**, care goleste vectorul "sequence". Cand unul din cele doua butoane este apasat, se apeleaza functia **printSequenceState()** care afiseaza un mesaj corespunzator si, in functie de parametrul dat, apeleaza functia de **playSequence**, respectiv goleste vectorul.

De asemenea, am setat un timer folosind registrii. Timer-ul este pornit prin apelarea functiei **startTimer()**, functie apelata odata ce unul din cele sase butoane este apasat, si aprinde LED-ul verde care, cat timp este aprins, permite redarea celor sase sound-uri. Odata ce timpul a expirat (18 secunde), se va aprinde LED-ul rosu timp de 5 secunde. Aprinderea LED-ului rosu permite redarea/resetarea secventei, sau, in cazul in care nu se doreste niciuna din cele doua optiuni, dupa ce acesta se stinge, se poate relua mixing-ul sound-urilor care va deschide un nou ciclu, pornind un nou timer. Timer-ul se foloseste de o intrerupere in care se intra odata la 0.5 secunde, asa ca am contorizat numarul de intrari in intrerupere pentru a putea verifica timpilor doriti.

In functia **setup()**, declar modul de folosire al piniilor (INPUT/OUTPUT), pornesc interfata seriala si LCD-ul, afisand un mesaj de inceput. In plus, se apeleaza trei functii auxiliare de setup pentru setarea registrilor corespunzatori intreruperilor si timer-ului si calculul specific fiecarui sound, mentionat la inceputul sectiunii. Pentru LCD am folosit biblioteca **LiquidCrystal\_I2C.h**, astfel comunicand I2C cu

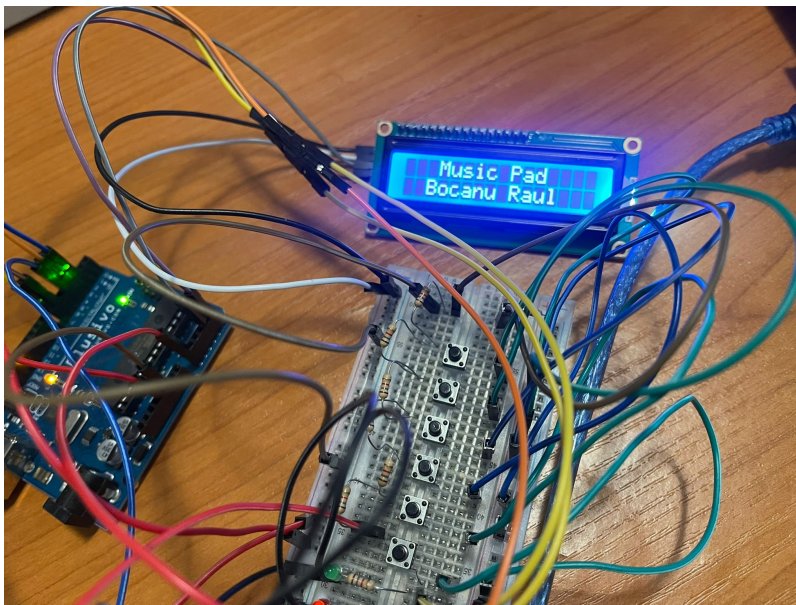
acesta.

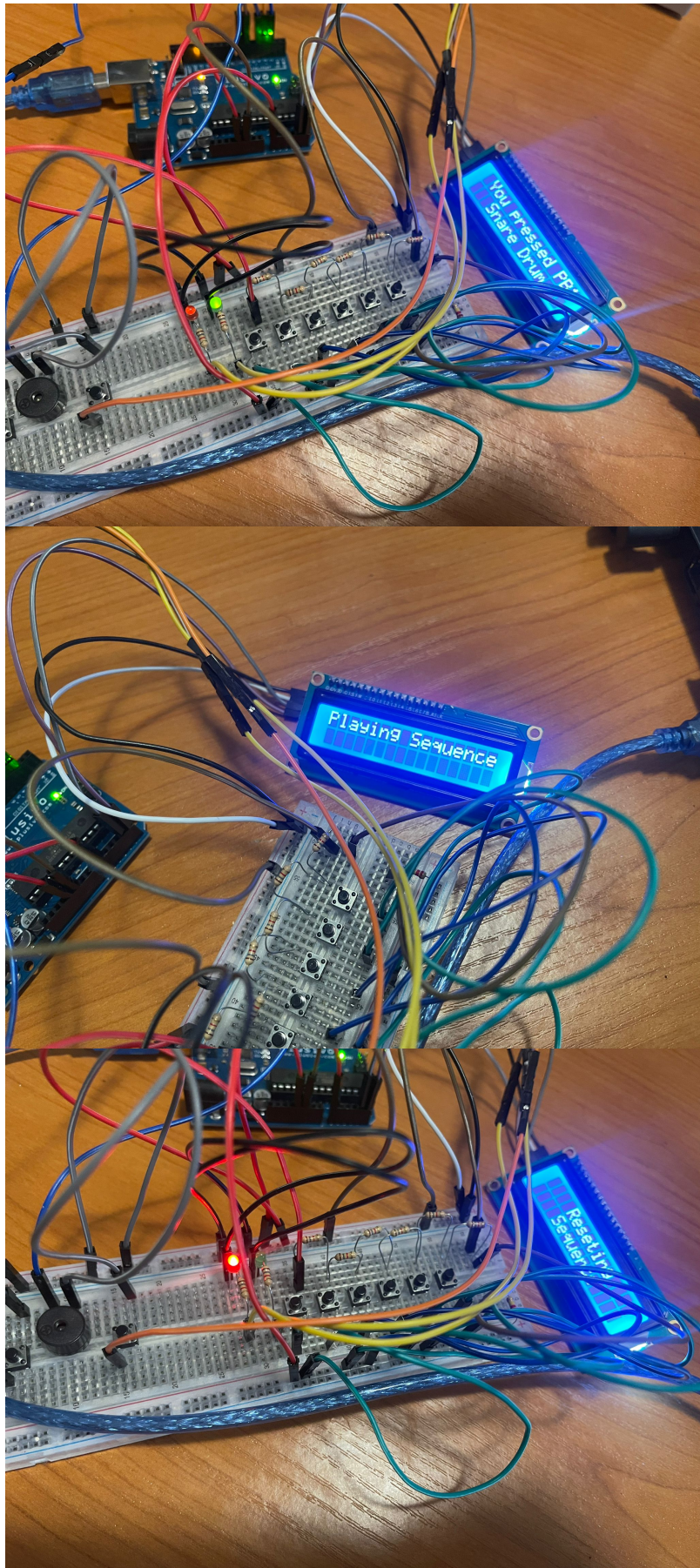
In functia **loop()**, verific care dintre cele trei functionalitati a fost folosita: "Play", "Reset", sau unul din cele 6 butoane pentru sound-uri.

O problema peste care am dat a fost declansarea unei intreruperi, din alt motiv decat cel pentru care a fost declarata. Astfel, am observat ca apasarea unuia dintre cele sase butoane care redau cele sase sunete diferite, declansa si intreruperea pentru butonul de "Play", conectat la pinul PD3. Motivele nu-mi sunt cunoscute pe deplin, eventuale cauze putand fi zgomotul sau fluctuatii de tensiune neasteptate. Pentru a rezolva aceasta problema, am dezactivat intreruperile la inceputul functiei **play()**, activand-ule inainte de iesirea din aceasta functie. Si pentru ca acest lucru nu a fost suficient, in cadrul intreruperii pentru butonul de play, am folosit functia Arduino **bit\_is\_clear()** pentru a verifica exclusiv schimbarea valorii de pe pinul PD3.

## Rezultate Obținute

La final, am reusit sa creez un Music Pad minimalist si perfect functional, fiind usor de folosit si totodata fun si relaxant.





Demo1

Demo2

## Concluzii

Realizare proiectului nu a fost o munca usoara neaparat, lovindu-ma de multe probleme care m-au adus des in pragul in care sa vrea sa renunt la tot. Cu toate acestea, de fiecare data cand am reusit sa implementez o functionalitate, ma bucura maxim sa vad un efect imediat, un efect tangibil, realizat de mine. Acest proiect mi-a dovedit ca pot sa dobor niste bariera pe care mi le-am pus initial, neasteptandu-ma sa duc totul la capat.

Acest tip de proiect este primul, si probabil ultimul, din cadrul facultatii si s-a dovedit a fi o experienta foarte placuta, care m-a ajutat sa inteleg anumite aspecte, sa vad ce inseamna lucrul practic, fiind obisnuiti in general cu experiente mult mai putin palpante, si nu in ultimul rand, sa ma distrez.

## Download

Am atasat codul sursa intr-un fisier .txt:

musicpad\_cod.zip

## Jurnal

- 3.05.2023: Am creat pagina proiectului si am descris, pe scurt, functionalitatea
- 18.05.2023: Am adăugat partea de Hardware Design si am conectat toate componentele fizice
- 20.05.2023: Am realizat testarea componentelor si cateva functionalitati de baza pentru proiect
- 23.05.2023: Am implementat redarea sunetelor prin butoane si functiile de Play si Reset
- 26.05.2023: Am terminat implementarea Software a proiectului si am finisat pagina OCW

## Bibliografie/Resurse

- [Lab PM 2](#)
- [Lab PM 3](#)
- [Lab PM 6](#)
- [Exemplu conectare mai multe butoane la acelasi pin](#)
- [Arduino Songs](#)
- [ChatGPT pentru a genera sound-urile ca note](#)
- [music\\_pad-1.pdf](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/iotelea/musicpad>



Last update: **2023/05/29 18:10**