

Bianca-Mihaela CAUC (78494) - Pian electronic

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Proiectul care se dorește a fi implementat constă în realizarea unui pian electronic capabil să redea notele muzicale din mai multe octave, permițând și opțiuni pentru înregistrarea și redarea melodiilor.

Astfel, se dorește ca proiectul să se apropie cât mai mult de modul de funcționare al unui pian adevărat și poate reprezenta o jucărie pentru cei pasionați de muzică, indiferent de vârsta lor.

Descriere generală

Pianul va conține butoane pentru a reprezenta clapele dintr-o octavă:

- 7 butoane care vor reprezenta clapele albe: *Do, Re, Mi, Fa, Sol, La, Si*
- 5 butoane pentru clapele negre



MOD DE FUNCȚIONARE

Pianul are 3 moduri de funcționare: **Normal Mode**, **Recording Mode**, **Play Mode** și permite emiterea notelor muzicale din primele 6 octave. Pentru schimbarea octavei curente se folosesc 2 butoane: incrementare octavă și decrementare octavă. Modul de funcționare în care se deschide programul este Normal Mode, având setată octava 1. Octava curentă poate fi identificată prin culoarea LED-ului corespunzător modului normal de funcționare (octava curentă poate fi văzută doar dacă pianul se află în Normal Mode sau Recording Mode). Astfel, codificările celor 6 octave sunt următoarele:

- **Octava 1:** <fc #0000FF>**Albastru**</fc>
- **Octava 2:** <fc #00FF00>**Verde**</fc>
- **Octava 3:** <fc #00FFFF>**Cyan**</fc>
- **Octava 4:** <fc #FF00FF>**Magenta**</fc>
- **Octava 5:** <fc #FFFF00>**Galben**</fc>
- **Octava 6:** Alb

- **Normal Mode:** Permite utilizatorului să cânte prin apăsarea butoanelor corespunzătoare notelor muzicale. În cazul în care acest mod este cel activ, LED-ul va avea una dintre culorile corespunzătoare celor 6 octave, altfel va fi `<fc #FF0000>roșu</fc>`.
- **Recording Mode:** acest mod permite utilizatorului să înregistreze melodia pe care o cântă (în cazul în care există deja o melodie înregistrată, aceasta va fi suprascrisă). Înregistrarea va porni în momentul în care se apasă butonul *Record* și se va opri când se apasă din nou pe buton. Starea în care se află pianul în cadrul Recording Mode va fi dată de un led RGB: dacă acest mod de funcționare nu este selectat, LED-ul va fi `<fc #FF0000>roșu</fc>`, altfel în momentul în care nu se înregistrează ledul va fi `<fc #0000FF>albastru</fc>` (indicând că este pregătit să înregistreze), iar în momentul în care se află în procesul de înregistrare, acesta va avea culoarea `<fc #00FF00>verde</fc>`. În plus, clapele pianului vor emite sunete doar când LED-ul acestui mod de funcționare este verde (nu se emit sunete dacă se află în starea READY), iar octava curentă va fi indicată de LED-ul corespunzător modului normal de funcționare.
- **Play Mode :** acest mod permite utilizatorului să redea ultima melodie înregistrată. În timpul redării melodiei, apăsarea butoanelor corespunzătoare clapelor nu va avea niciun efect (nu se va emite niciun sunet). Pentru a opri redarea melodiei, se va apăsa din nou butonul corespunzător modului *Play*. Ca și în cazul modului anterior, LED-ul va indica starea: `<fc #FF0000>roșu</fc>` indică faptul că acest mod de funcționare nu este selectat, culoarea `<fc #00FF00>verde</fc>` a LED-ului va semnifica faptul că o melodie este redată în acel moment, iar `<fc #0000FF>albastru</fc>` indică faptul că este pregătit să redea o melodie înregistrată.

SCHEMA BLOC



Hardware Design

LISTĂ PIESE

Pe lângă componentele plăcii de bază, se vor folosi:

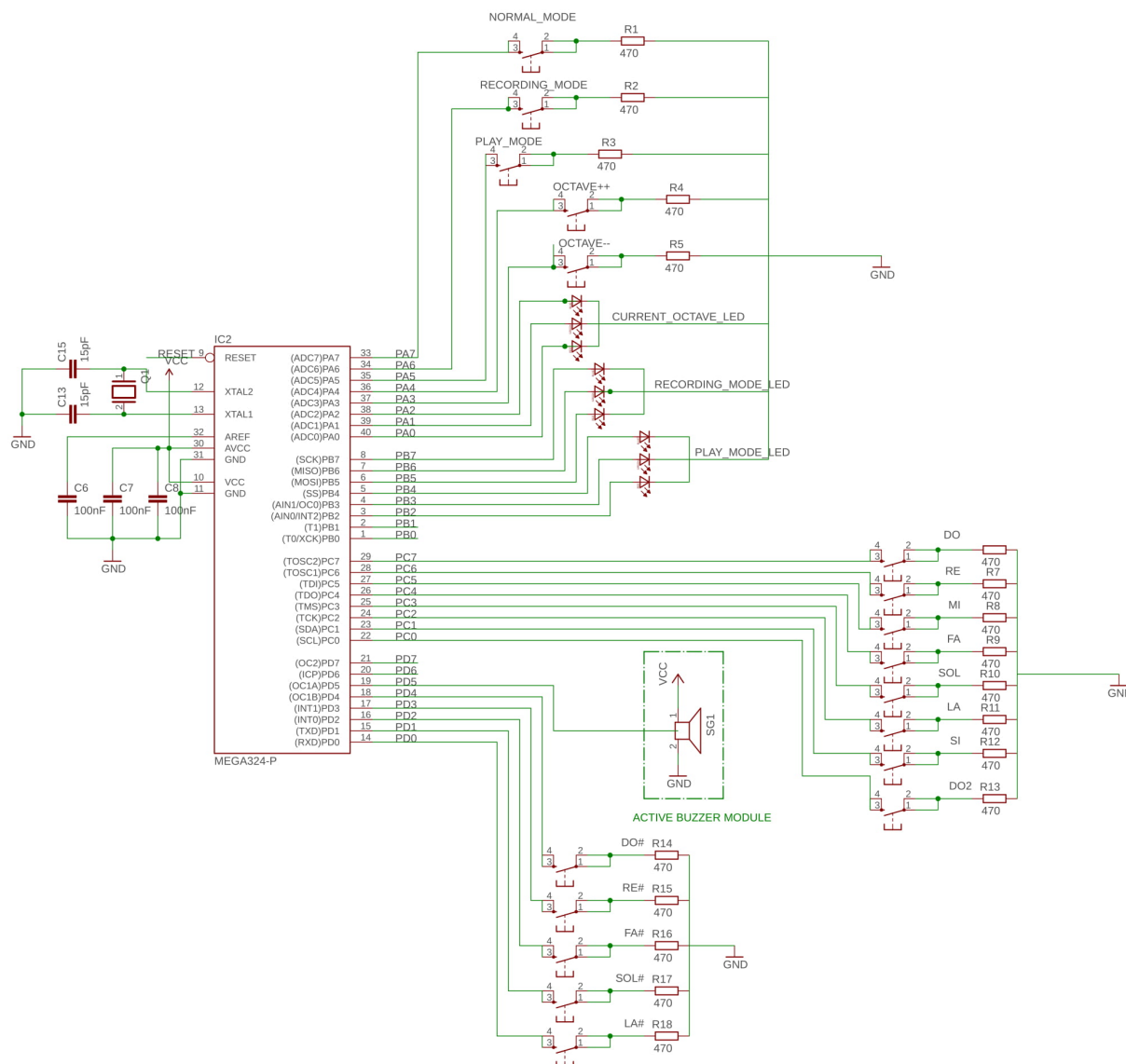
- **Buzzer**
- **12 butoane** care corespund clapelor: 7 butoane pentru clapele albe și 5 butoane pentru clapele negre
- **2 butoane** pentru schimbarea octavei curente: un buton pentru incrementare și unul pentru decrementare
- **3 butoane** pentru schimbarea modului de funcționare: un buton pentru setarea *Normal Mode*, un buton pentru *Recording Mode* (start/stop recording) și unul pentru *Play Mode* (start/stop melodie înregistrată)
- **3 LED-uri RGB** folosite pentru indicarea stărilor modurilor de funcționare și a octavei curente
- Alte componente adiționale

Detalii componente

Denumire	Număr	Preț	Link către produs
Placă de bază	x1	8 RON	-
Componente placă de bază	x1	24 RON	-
Cablu USB AM → USB BM	x1	3.49 RON	https://goo.gl/KAVn2a
Placă test 12cm x 18 cm	x1	10 RON	-
Modul Buzzer activ	x1	5.49 RON	https://goo.gl/DFwUk8
Butoane	x17	16.83 RON	https://goo.gl/21rLGo
LED RGB cu catod comun	x3	2.85 RON	https://goo.gl/cfvpTb
Rezistențe 470R	x9	3.24 RON	https://goo.gl/TZjNWh
Cablu mamă-mamă 20cm	x1	3.99 RON	https://goo.gl/RPRNd1
Cablu mamă-mamă 30cm	x3	22.35 RON	https://goo.gl/QFUKud
Cablu mamă-tată 10cm	x1	2.99 RON	https://goo.gl/H5CFcR
Header de pini	x1	0.99 RON	https://goo.gl/qvzbGK
Ojă albă	x1	-	-
Total			104.22 RON

SCHEMĂ ELECTRICĂ

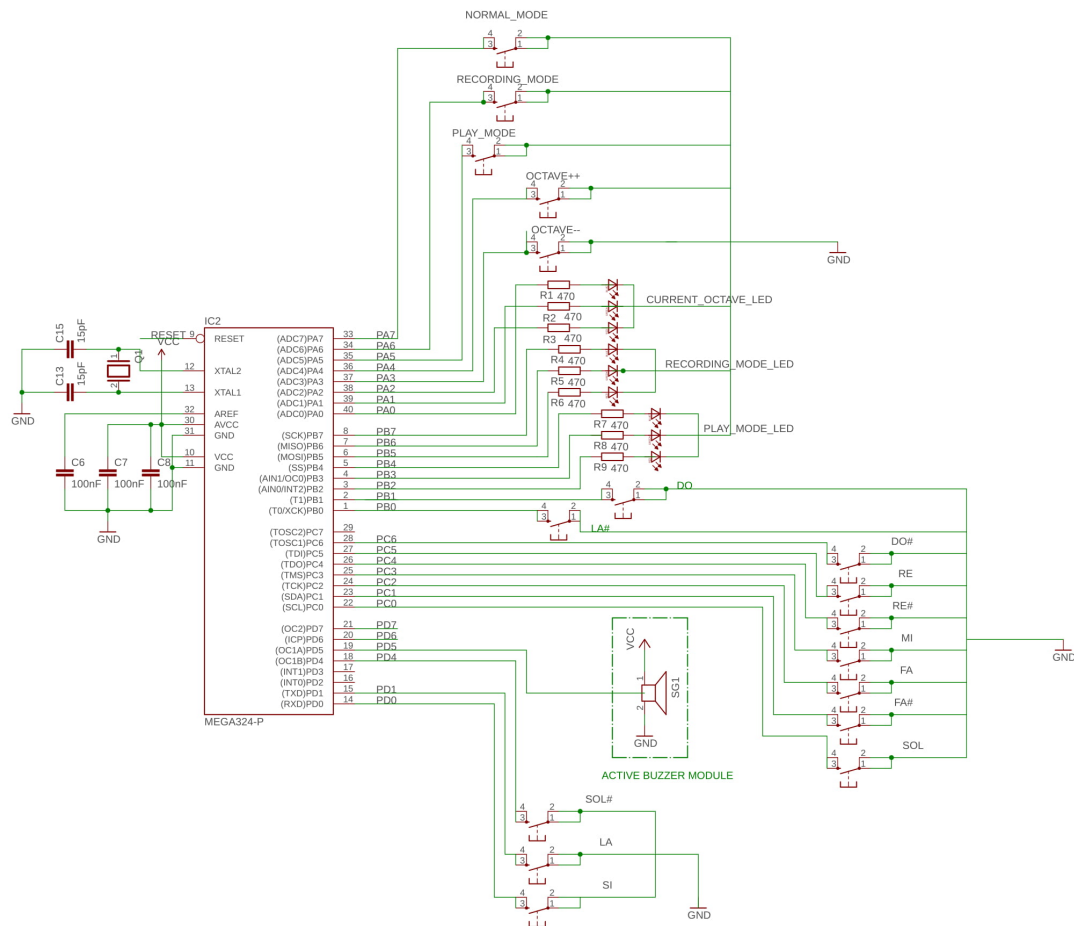
Inițial, am dorit să implementez următoarea schemă electrică:



Totuși, în momentul în care m-am apucat de implementarea hardware, am întâmpinat câteva probleme, deoarece programul nu funcționa așa cum ar fi trebuit când legam butoane la următorii pini:

- **PC7** (se detecta butonul conectat la acest pin ca fiind mereu apăsat)
- **PD2** și **PD3** (nu mai intra în bootloader)

Din aceste motive, a trebuit să renunț la folosirea lor și să mă folosesc de cei rămași liberi (**PB1** și **PB0**) și din cauză că nu mai aveam încă un pin liber, am decis să elimin butonul corespunzător notei DO2, întrucât aceasta poate fi redată prin alegerea lui DO din octava superioară a celei curente. De asemenea, deși inițial am dorit să pun toate butoanele corespunzătoare clapelor albe la un port, iar cele corespunzătoare clapelor negre la un alt port , mi-am dat seama că acest lucru îmi va cauza probleme când lipesc, deoarece s-ar fi încurcat firele între ele foarte mult. Schema electrică finală este următoarea:



- Butoanele pentru cele 3 moduri de funcționare: **PA5** (Play mode), **PA6** (Recording mode), **PA7** (Normal mode)
- Cele 2 butoane pentru setarea octavei curente: **PA3** (Octava-), **PA4** (Octava++)
- LED RGB pentru indicarea octavei curente: **PA0** (Red), **PA1** (Green), **PA2** (Blue)
- LED RGB pentru Recording Mode: **PB5** (Red), **PB6** (Green), **PB7** (Blue)
- LED RGB pentru Play Mode: **PB2** (Red), **PB3** (Green), **PB4** (Blue)
- Butoane corespunzătoare clapelor albe: **PB1** (DO), **PC5** (RE), **PC3** (MI), **PC2** (FA), **PC0** (SOL), **PD1** (LA), **PD0** (SI)
- Butoane corespunzătoare clapelor negre: **PC6** (DO#), **PC4** (RE#), **PC1** (FA#), **PD4** (SOL#), **PB0** (LA#)
- Buzzer: **PD5**

SOFTWARE DESIGN

MEDIUL DE DEZVOLTARE

Aplicația a fost dezvoltată în Programmer's Notepad, iar pentru încărcarea acesteia pe microcontroller s-a folosit HID Boot Flash.

DESCRIEREA CODULUI SURSĂ AL APLICAȚIEI

Am început prin a defini în `my_utils.h` o serie de variabile globale și de macro-uri cu scopul de a avea un coding style cât mai frumos.

- Stările posibile ale unui mod de funcționare:

```
#define DISABLED 0
#define READY    1
#define ACTIVE   2
```

- Reținerea pinilor pentru fiecare culoare a fiecărui LED din cele 3 posibile:

```
/* Pini pentru LED-ul corespunzator modului normal de functionare + octava
curenta */
#define NORMAL_MODE_RED      PA2
#define NORMAL_MODE_GREEN    PA1
#define NORMAL_MODE_BLUE     PA0

/* Pini pentru LED-ul corespunzator modului de inregistrare */
#define RECORDING_MODE_RED    PB5
#define RECORDING_MODE_GREEN  PB6
#define RECORDING_MODE_BLUE   PB7

/* Pini pentru LED-ul corespunzator modului de redare */
#define PLAY_MODE_RED         PB2
#define PLAY_MODE_GREEN       PB3
#define PLAY_MODE_BLUE        PB4
```

- Reținerea pinilor unde sunt conectate butoanele de control și buzzer-ul

```
/* Pini pentru butoanele de control */
#define NORMAL_MODE_BUTTON    PA7
#define RECORDING_MODE_BUTTON PA6
#define PLAY_MODE_BUTTON      PA5
#define OCTAVE_PLUS           PA4
#define OCTAVE_MINUS          PA3

/* Pin pentru modulul de buzzer activ */
#define BUZZER                 PD5
```

- Reținea pinilor unde sunt conectate butoanele corespunzătoare notelor muzicale

```
#define D0      PB1
#define D0_DIEZ PC6
#define RE      PC5
#define RE_DIEZ PC4
#define MI      PC3
#define FA      PC2
```

```
#define FA_DIEZ    PC1
#define SOL        PC0
#define SOL_DIEZ   PD4
#define LA         PD1
#define LA_DIEZ    PB0
#define SI         PD0
```

De asemenea, se folosesc următoarele variabile globale:

- Frecvențele fiecărei note muzicale pentru primele 6 octave:

```
double DO_FREQUENCY[NOF_OCTAVES]      = { 16.352, 32.703, 65.406, 130.813,
261.626, 523.251 };
double DO_DIEZ_FREQUENCY[NOF_OCTAVES] = { 17.324, 34.648, 69.296, 138.591,
277.183, 554.365 };
double RE_FREQUENCY[NOF_OCTAVES]      = { 18.354, 36.708, 73.416, 146.832,
293.665, 587.333 };
double RE_DIEZ_FREQUENCY[NOF_OCTAVES] = { 19.445, 38.891, 77.782, 155.563,
311.127, 622.254 };
double MI_FREQUENCY[NOF_OCTAVES]      = { 20.602, 41.203, 82.407, 164.814,
329.628, 659.255 };
double FA_FREQUENCY[NOF_OCTAVES]      = { 21.827, 43.654, 87.307, 174.614,
349.228, 698.456 };
double FA_DIEZ_FREQUENCY[NOF_OCTAVES] = { 23.125, 46.249, 92.499, 184.997,
369.994, 739.989 };
double SOL_FREQUENCY[NOF_OCTAVES]     = { 24.500, 48.999, 97.999, 195.998,
391.995, 783.991 };
double SOL_DIEZ_FREQUENCY[NOF_OCTAVES] = { 25.957, 51.913, 103.826, 207.652,
415.305, 830.609 };
double LA_FREQUENCY[NOF_OCTAVES]      = { 27.500, 55.000, 110.000, 220.000,
440.000, 880.000 };
double LA_DIEZ_FREQUENCY[NOF_OCTAVES] = { 29.135, 58.270, 116.541, 233.082,
466.164, 932.328 };
double SI_FREQUENCY[NOF_OCTAVES]      = { 30.868, 61.735, 123.471, 246.942,
493.883, 987.767 };
```

- Variabile pentru reținerea stării în care se găsește fiecare mod de funcționare și octava curentă

```
static unsigned int normal_mode = ACTIVE;
static unsigned int recording_mode = DISABLED;
static unsigned int play_mode = DISABLED;
static unsigned int current_octave = 1;
```

Algoritmul propriu-zis

Programul folosește tehnica de busy-waiting, verificând la fiecare 250ms ce butoane au fost apăsată (am încercat să implementez cu întreruperi, însă din a numite motive pe care nu am reușit să le descopăr, programul nu funcționa așa cum îmi doream 😞)

```
int main(void) {

    set_environment();
```

```
    while (1) {  
        check_buttons();  
        _delay_ms(250);  
    }  
  
    return 0;  
}
```

- Funcția **set_environment** inițializează toate intrările și ieșirile aplicației. De asemenea, inițializează pianul ca fiind în modul normal de funcționare cu octava 1 activă, prin aprinderea primului LED albastru, iar a celorlalte roșii.
- Funcția **check_buttons** verifică pe rând toate tipurile de butoane pentru a vedea care este apăsat: inițial se verifică dacă s-a apăsat unul dintre butoanele care schimbă modul de funcționare al pianului. În cazul în care nu s-a întâmplat acest lucru, se verifică dacă s-a apăsat unul dintre butoanele care schimbă octava curentă. În cazul negativ, se verifică ce mod de funcționare este activ și se execută operația specifică (cântă, înregistrează melodie, redă melodie înregistrată)
- Funcția **check_changed_mode** verifică dacă s-a apăsat unul dintre butoanele specifice schimbării modului de funcționare al pianului. Astfel, se testează pe rând fiecare dintre cele 3 butoane, se actualizează stările tuturor modurilor și ulterior se apelează o altă funcție care actualizează culoarea LED-urilor. În plus, în cazul în care se activează modul de înregistrare, se reinițializează indexul curent din vector pentru a putea suprascrie melodia deja existentă (în cazul în care există). Dacă se dezactivează modul de înregistrare (stop recording), se adaugă la finalul vectorului -1 pentru a marca sfârșitul melodiei.
- Funcția **check_changed_octave** verifică dacă a fost apăsat unul dintre butoanele care schimbă octava curentă. În caz afirmativ, se incrementează/decrementează și se actualizează culoarea primului LED în funcție de octava aleasă. În plus, schimbarea octavei este posibilă doar în momentul în care modul activ de funcționare este cel normal sau cel de înregistrare.
- Funcția **change_colors** actualizează culoarea celor 3 LED-uri în funcție de stările modurilor de funcționare și de octava curentă. Astfel, inițial este stins LED-ul și ulterior este verificată starea fiecărui mod de funcționare. În cazul în care acestea sunt dezactivate, culoarea este roșie, iar în caz contrar: LED-urile corespunzătoare pentru Recording Mode și Play Mode sunt galbene în momentul în care se află în starea READY și verzi în momentul în care se află în starea ACTIVE, iar LED-ul corespunzător Normal Mode indică culoarea octavei curente dacă ne aflăm în modul normal de funcționare sau în modul de înregistrare, altfel este roșu.
- Funcția **change_octave_colors** actualizează culoarea primului LED în funcție de valoarea variabilei care memorează indicii octavei curente (conform codificării de mai sus)
- Funcția **check_pressed_note** verifică dacă a fost apăsat vreun buton corespunzător unei clape. În caz afirmativ se redă nota muzicală folosind play_sound care primește ca parametrii frecvența notei (corespunzătoare octavei curente în acel moment) și durata sunetului. După redarea sunetului, dezactivez buzzer-ul întrucât aceasta este activ pe 0 și rămâne blocat. Funcția returnează valoarea notei apăsată, deoarece este folosită și în cadrul funcției de înregistrare și este necesară frecvența acesteia pentru a putea fi memorată într-un vector.
- Funcția **record_music** are ca scop înregistrarea notelor muzicale: se folosește funcția menționată anterior pentru a determina frecvența notei muzicale alese de utilizator și se reține într-un vector.

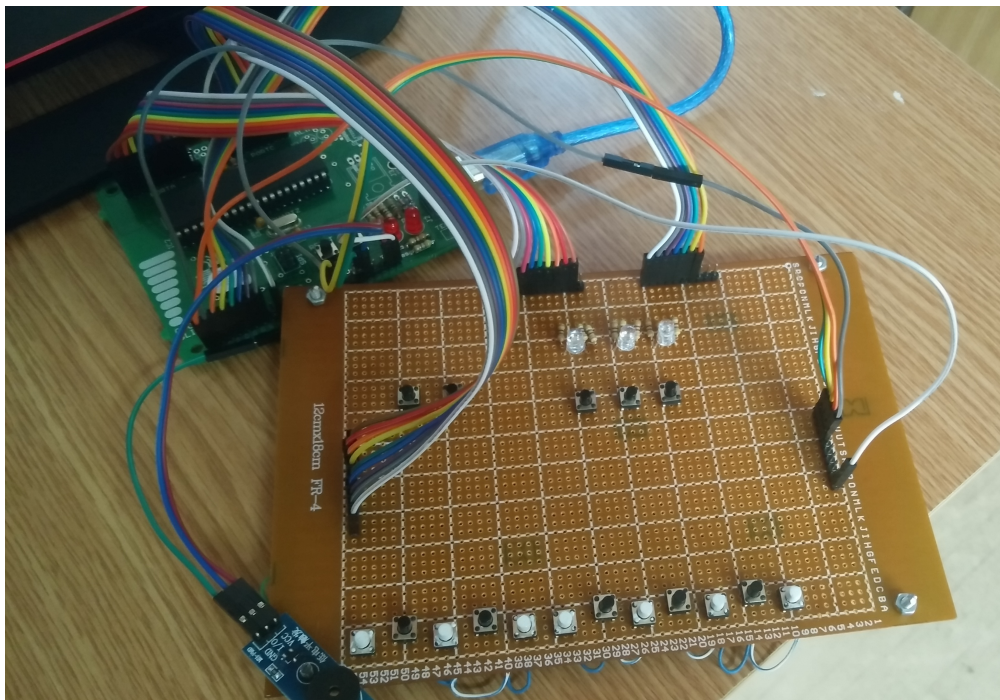
- Funcția **play_recorded** redă ultima melodie înregistrată de utilizator prin redarea succesivă a tuturor notelor a căror frecvență a fost salvată în vector de către funcția `record_music`. Înainte de a reda nota următoare, se verifică de fiecare dată dacă utilizatorul a apăsat butonul pentru oprirea melodiei în curs de redare.

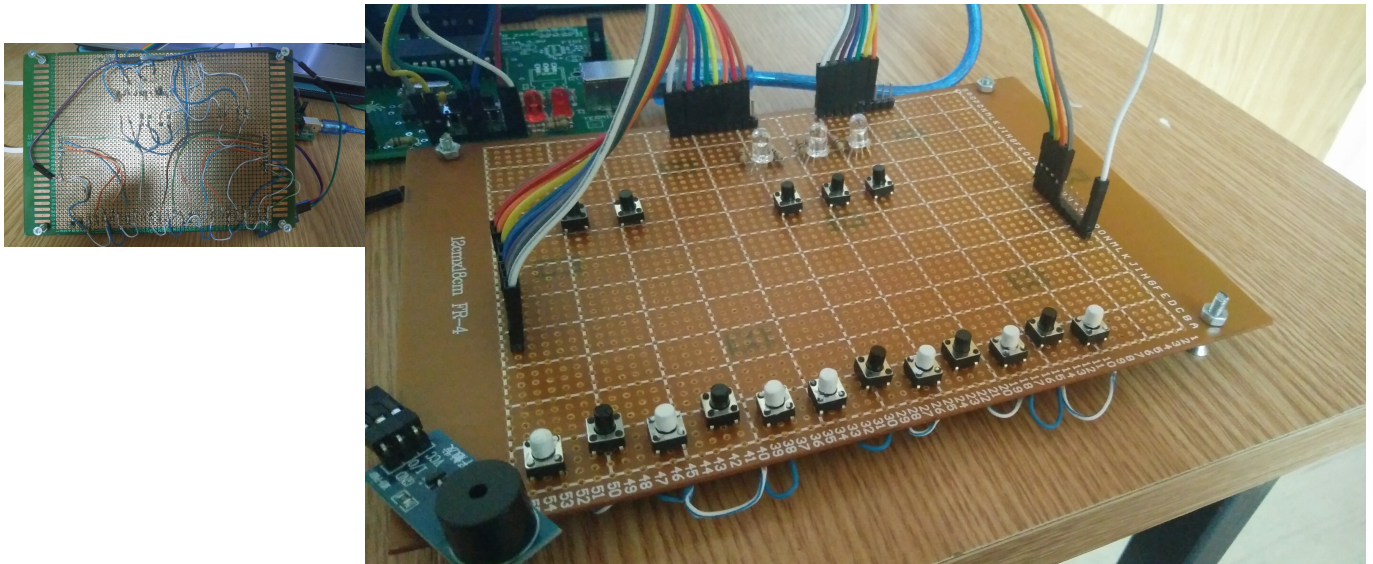
Rezultate Obținute

Demo movies

- Demo Movie: <https://youtu.be/pHx7MKh8L3w>
- Happy Birthday to you Recording: <https://youtu.be/6XZMZ0pbKGU>

Fotografii





Concluzii

Ideea proiectului este una interesantă și este foarte plăcut în momentul în care chiar vezi că merge ce ai implementat. Totuși, a fost destul de scurt timpul de care l-am avut la dispoziție și a fost destul de dificil la început până am înțeles exact cum trebuie să procedez și de ce piese am nevoie. M-am ales cu câteva arsuri de la lipituri și am cedat nervos de câteva ori în momentele în care brusc nu mai mergea nimic, însă până la urmă sunt mulțumită de rezultatul final 😊 Problemele care mi-au cauzat cei mai mulți nervi:

- Credeam că am stricat plăcuța de bază, deoarece refuza să intre în bootloader, însă problema a fost că eu țineam mâna pe spatele plăcuței și cumva se creaa un contact între anumite pini și nu mai mergea 🤔
- În momentul în care am pus firele pe portul D nu a mai intrat în bootloader și a durat ceva să îmi dau seama că problema este cauzată de PD2 și PD3 (ulterior s-a menționat și pe cs.curs acest lucru)
- Butonul conectat inițial la PC7 este mereu apăsat și am crezut inițial că poate am lipit greșit butonul
- În încercarea de a avea un coding style cât mai drăguț, am folosit foarte multe constante și variabile globale (inclusiv un vector foarte mare pentru înregistrare) șiiii.... nu mai făcea absolut nimic programul 😞 După ceva timp, am observat că dacă scot din variabile merge și am restructurat

Download

- Schemă electrică: [schema_electrica_pian.sch](#)
- Implementare software: [pian_electronic.zip](#)

Jurnal

- **21.04.2018:** Stabilirea obiectivelor minime pentru proiect și actualizarea paginii de wiki cu informațiile pentru Milestone1 (temă proiect, descriere, mod de funcționare, schemă bloc, piese necesare)

- **04.05.2018:** Finalizarea plăcii de bază
- **06.05.2018:** Actualizarea paginii de wiki prin adăugarea schemei electrice și a detaliilor componentelor cumpărate
- **17.05.2018 - 20.05.2018:** Lipirea tuturor componentelor și refacerea schemei electrice
- **21.05.2018 - 22.05.2018:** Implementarea software-ului (fără partea de înregistrare și redare a ultimei melodii înregistrate)
- **23.05.2018:** Actualizarea paginii de Wiki cu toate informațiile despre proiect până în momentul curent
- **24.05.2018:** Rezolvarea bug-urilor de la înregistrare și redare + actualizare wiki cu modificările făcute

Bibliografie/Resurse

- https://ro.wikipedia.org/wiki/Gam%C4%83_muzical%C4%83
- <http://www.sengpielaudio.com/calculator-notenames.htm> (pentru frecvențele specifice fiecărei octave)
- Wiki PM
- Proiectele din anii trecuți
- Documentația în format [PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2018/adraghici/bianca_mihaela.cauc

Last update: **2021/04/14 15:07**

