

Pian Electric

Autor:

- [Megelea Diana-Mihaela](#)

Introducere

În cadrul acestui proiect, am ales să implementez un pian electric ce este capabil să redea gama do major. La apăsarea unui buton, pe lângă redarea notei corespunzătoare din gamă, se va aprinde și un LED care rămâne aprins cât timp este apăsat butonul. De asemenea, pianul dispune de funcțiile de REC și PLAY, care permit înregistrarea și, ulterior, redarea notelor.

Scopul acestui proiect a fost de a crea un mic instrument la care utilizatorul să poată cânta melodii simple sau chiar să își creeze diferite sunete pentru remix-uri. Deși nu poate fi considerat un tool cu adevărat folositor pentru cineva dornic să producă muzică, ar putea fi folosit pentru a experimenta la început de drum sau chiar ca instrument anti-stres pentru cineva care nu este atras de muzică.

Descriere generală

Acest proiect simulează structura unei claviaturi prin cele 8 butoane disponibile pe breadboard. Pe lângă funcționalitatea de bază, aceea de a reda cu ajutorul difuzorului notele aferente fiecărui buton, „pianul” va aprinde câte un LED în funcție de „clapa” apăsată și va permite înregistrarea și redarea melodiilor.

Am dorit să integrez această funcționalitate suplimentară, întrucât mi se pare interesant să poți observa și vizual felul în care se modifică notele. Astfel, pianul nu simulează doar un joc de sunete, cât și de lumini, putând fi o jucărie chiar și pentru cei mai mici (care pot, de altfel, învăța să cânte o melodie dacă cineva o înregistrează în prealabil).



Hardware Design

Componente folosite:

- o placă Arduino
- un breadboard
- 10 butoane
- 10 rezistențe de 10kΩ
- un difuzor

- 10 LED-uri
- 10 rezistențe de 220Ω



Software Design

Proiectul a fost întâi simulat și testat în Tinkercad pentru a mă asigura atât de faptul că au fost conectate bine piesele, cât și de funcționalitatea corectă a codului. Ulterior am folosit Arduino IDE ca mediu de dezvoltare.

Detalii implementare

Pentru funcția de recording am folosit un buffer de dimensiune prestabilită în care salvez frecvența notelor corespunzătoare butoanelor apăstate, crescând un contor. După apăsarea butonului de redare, contorul se resetează la 0 pentru a putea înregistra ulterior o altă piesă.

Întrucât am comandat din neatenție un modul buzzer activ, pentru a-l putea face să funcționeze precum unul pasiv (să redea doar sunetele corespunzătoare butoanelor apăstate, nu să scoată un sunet continuu) am setat pin-ul corespunzător buzzer-ului pe HIGH.

Cod sursă

```
int buton1 = 2; // do
int buton2 = 3; // re
int buton3 = 4; // mi
int buton4 = 5; // fa
int buton5 = 6; // sol
int buton6 = 7; // la
int buton7 = 12; // si
int buton8 = 8; // do
int butonREC = 10; // rec
int butonPLAY = 11; // play
int buzzer = 9;

int on = 0; // should rec or not
int i = 0;

int frequencies[] = {262, 294, 330, 349, 392, 440, 494, 523};
int sounds[100];

void setup() {
```

```
pinMode(buton1,INPUT);
pinMode(buton2,INPUT);
pinMode(buton3,INPUT);
pinMode(buton4,INPUT);
pinMode(buton5,INPUT);
pinMode(buton6,INPUT);
pinMode(buton7,INPUT);
pinMode(buton8,INPUT);
pinMode(butonREC,INPUT);
pinMode(butonPLAY,INPUT);
pinMode(buzzer,OUTPUT);
digitalWrite(buzzer, HIGH);
}

void loop() {
  int b1 = digitalRead(buton1);
  int b2 = digitalRead(buton2);
  int b3 = digitalRead(buton3);
  int b4 = digitalRead(buton4);
  int b5 = digitalRead(buton5);
  int b6 = digitalRead(buton6);
  int b7 = digitalRead(buton7);
  int b8 = digitalRead(buton8);
  int bREC = digitalRead(butonREC);
  int bPLAY = digitalRead(butonPLAY);

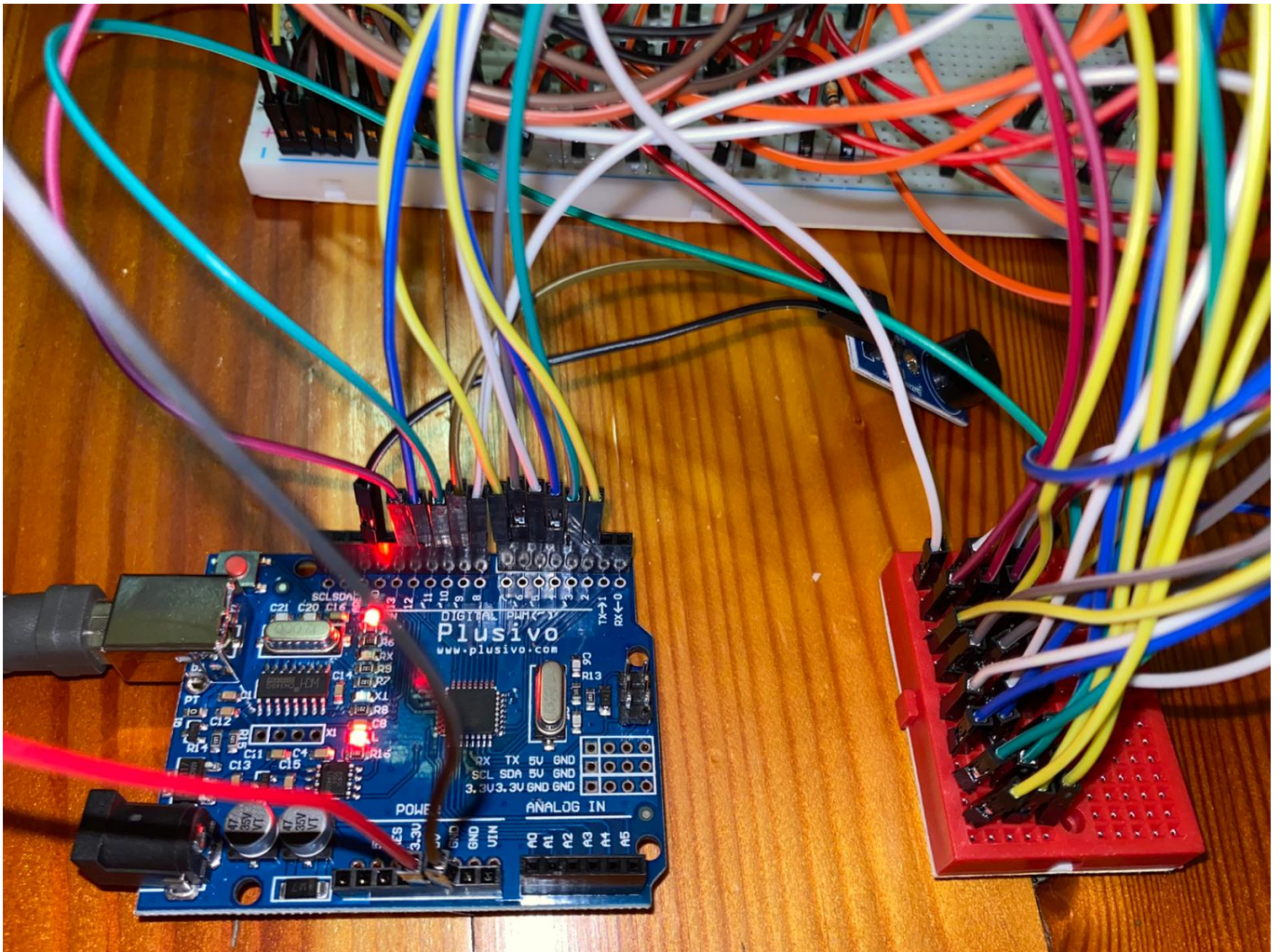
  if (bREC) {
    if (on == 0) {
      on = 1;
    } else if (on == 1) {
      on = 0;
    }
  }

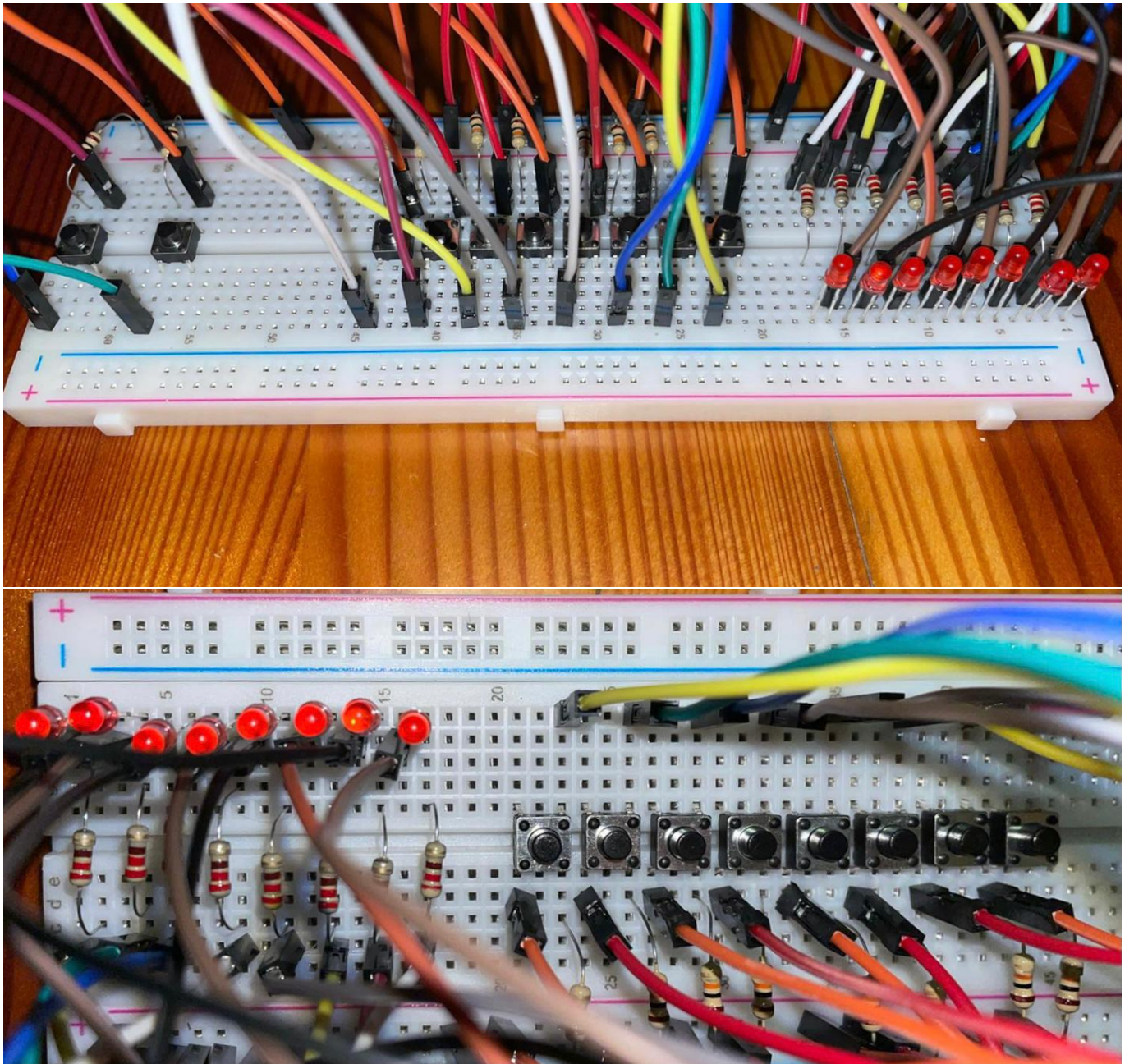
  if (b1) {
    tone(buzzer,frequencies[0],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (on) {
      sounds[i] = frequencies[0];
      i = i + 1;
    }
  } else if (b2) {
    tone(buzzer,frequencies[1],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (on) {
      sounds[i] = frequencies[1];
      i = i + 1;
    }
  } else if (b3) {
```

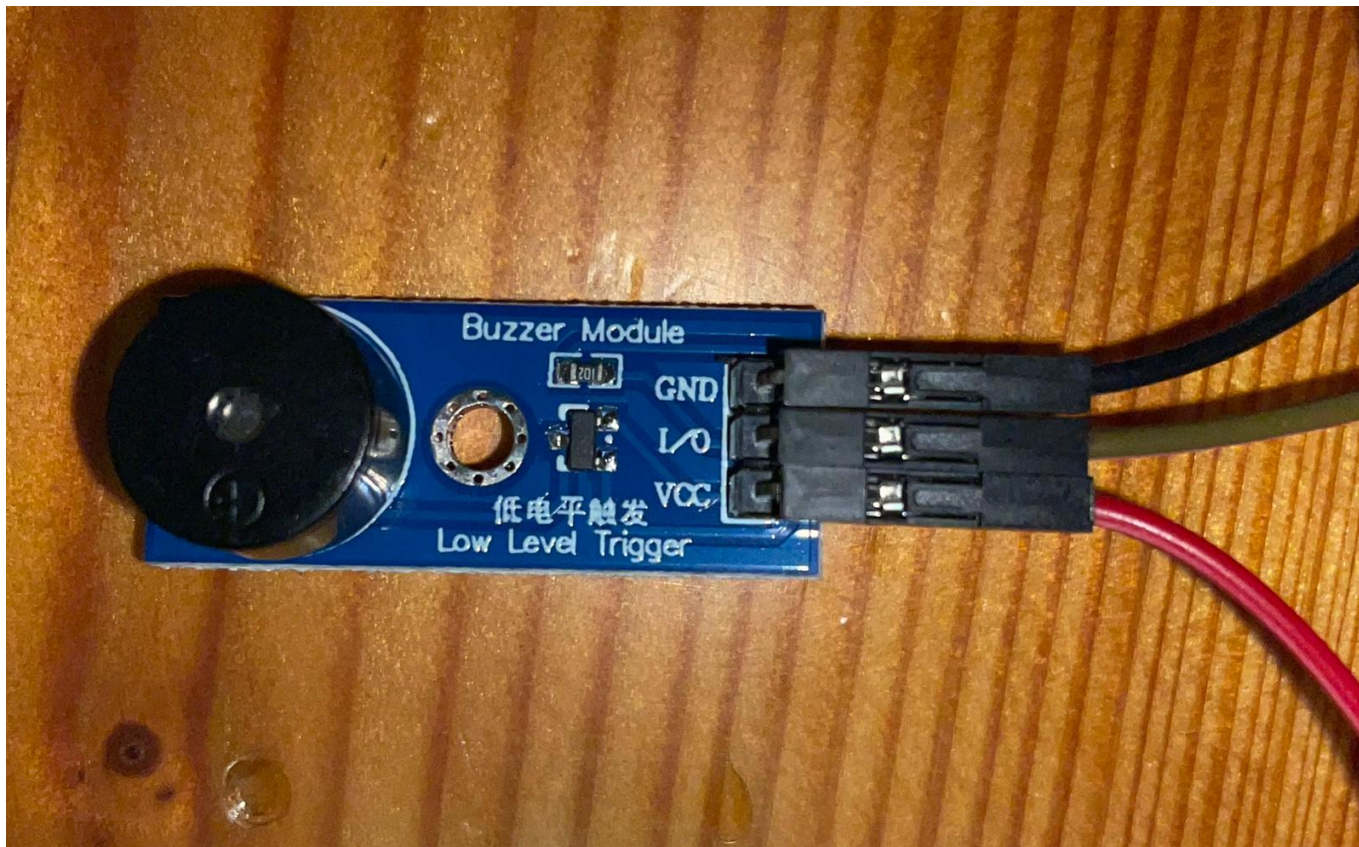
```
tone(buzzer,frequencies[2],100);
delay(100);
digitalWrite(buzzer, HIGH);
if (on) {
    sounds[i] = frequencies[2];
    i = i + 1;
}
} else if (b4) {
    tone(buzzer,frequencies[3],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (on) {
        sounds[i] = frequencies[3];
        i = i + 1;
    }
} else if (b5) {
    tone(buzzer,frequencies[4],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (on) {
        sounds[i] = frequencies[4];
        i = i + 1;
    }
} else if (b6) {
    tone(buzzer,frequencies[5],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (on) {
        sounds[i] = frequencies[5];
        i = i + 1;
    }
} else if (b7) {
    tone(buzzer,frequencies[6],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (bREC) {
        sounds[i] = frequencies[6];
        i = i + 1;
    }
} else if (b8) {
    tone(buzzer,frequencies[7],100);
    delay(100);
    digitalWrite(buzzer, HIGH);
    if (on) {
        sounds[i] = frequencies[7];
        i = i + 1;
    }
} else if (bPLAY) {
    for (int j = 0; j < i; j = j + 1) {
        tone(buzzer, sounds[j], 300);
        delay(300);
    }
}
```

```
digitalWrite(buzzer, HIGH);  
delay(200);  
}  
i = 0;  
}  
}
```

Rezultate Obținute







În implementarea fizică, față de cea de pe Tinkercad prezentată la început, am folosit un mini breadboard adițional pentru a permite conectarea mai multor fire la același pin de pe plăcuța Arduino. De asemenea, butoanele de REC și PLAY nu mai au LED-uri asociate întrucât din comandă mi-a lipsit unul dintre LED-uri și am decis în final să nu mai pun la niciunul dintre aceste 2 butoane (făcând vizibilă conexiunea buton-LED la partea de claviculă).

Funcționalitățile proiectului pot fi vizualizate în următorul [demo](#). La început, redau fiecare notă din gama do major, aprinzându-se LED-ul corespunzător fiecărei note muzicale. Ulterior, pornesc funcția de recording și cânt primele 2 versuri din *Twinkle, twinkle*, iar apoi le redau oprind înregistrarea și apăsând butonul de play.

De menționat ar fi faptul că filmarea s-a făcut din spatele "pianului" (adică invers față de cum este prezentat pianul în schema Tinkercad) întrucât, din cauza numărului mare de fire, nu ar fi putut fi observate LED-urile și nici butoanele nu s-ar fi văzut bine (sorry for the confusion 😞)

Concluzii

Consider că acest proiect a reprezentat cea mai interesantă temă pe care am avut-o, întrucât a fost prima în care am îmbinat hardware-ul și software-ul. Deși nu este ceva complicat, a fost interesant să mă informez cum să folosesc o plăcuță Arduino și cum să poziționez butoane, LED-uri și un difuzor pe un breadboard și să le interconectez. Tinkercad-ul a fost foarte util în cazul acestui proiect, făcând din acesta un lucru ușor aplicabil în practică odată ce reușești să îl simulezi.

Resurse

- [Demo proiect](#)
- Codul sursă poate fi descărcat de aici: [proiectpm.zip](#)
- Documentul în format PDF: [pian_electric.pdf](#)

Bibliografie

- Laboratoare
- [Documentația oficială](#)
- [Conectare butoane și buzzer](#)
- [Folosire buzzer activ](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/abasoc/pianelectric>

Last update: **2021/06/03 01:30**

