

Step Sequencer

Autor: [Iulian-Sebastian Docan](#)

Grupa: 331AC

Introducere

Proiectul constă în implementarea unui Step Sequencer ce va produce diverse secvențe de tonuri 8-bit. Acestea vor fi redade cu ajutorul unor potențiometre care vor permite modificarea și alternarea frecvențelor pentru a emite tipare diverse de sunete, similare consolelor Nintendo din anii '70,'80. Pe lângă această funcție de a reda sunete pe un output audio, va exista o altă funcționalitate, care va oferi posibilitatea de modificare a bpm-ului, pentru a crește succesiunea de note cântate, lucru care va fi observabil prin intermediul unor LED-uri care se vor aprinde în ordine.

Descriere generală

Având în vedere obiectivul acestui proiect, utilizarea celor 4 potențiometre pentru modificarea tonalității notelor constituie o modalitate optimă și facilă de implementare. Includerea potențiometrelor a impus și folosirea funcției specifice Arduino numită `map()`, pentru a calibra noul interval de valori care vor fi modificate. Prin urmare, capetele default ale interfaței 0 - 1023 sunt remapate pentru a corespunde intervalului marcat de variabilele `lowFreq` - `highFreq` din codul sursă. De asemenea, redarea notelor muzicale necesită folosirea unor alte funcții specifice, `tone()` și `noTone()`. Cea dintâi va reda un semnal dreptunghiular pe pinul desemnat pentru mini-speaker, cu o frecvență la alegere (nota propriu-zisă) și o durată (poate fi omisă), iar cea de-a doua funcție va înceta redarea pe acel pin.

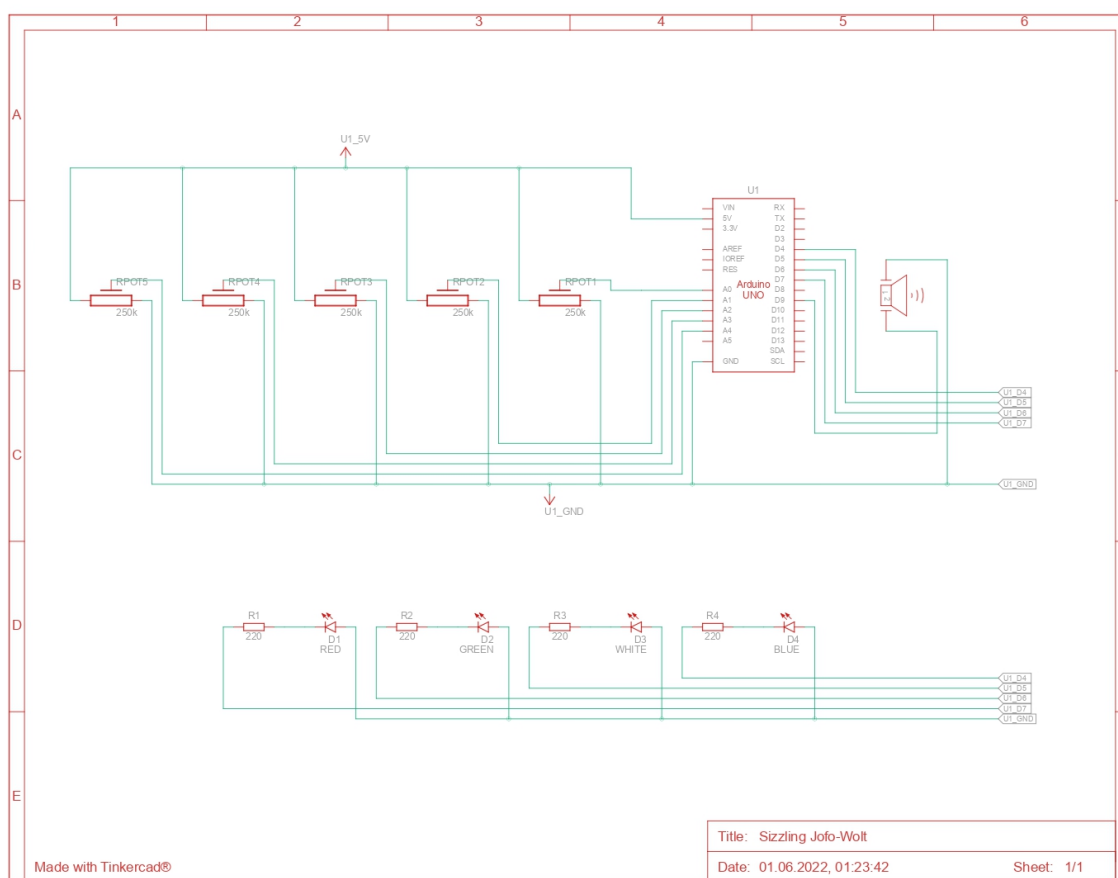
Pe lângă aceste module importante, se numără și includerea LED-urilor care se aprind într-o cadență determinată de BPM-ul setat de al cincilea potențiometru. Aceste LED-uri conectate la patru din pinii digitali ai plăcii se aprind și se sting folosind o funcție definită care realizează apeluri ale modulului `digitalWrite()`, în funcție de valoarea fiecărui potențiometru.



Hardware Design

Componente:

- Arduino UNO
- Breadboard
- Fire
- 4 x Rezistor 220Ohm
- 5 x Potențiometru Stereo 10k
- 4 x LED colorat
- Speaker 1W, 80hm

**Software Design**

A fost utilizat mediul de dezvoltare Arduino 1.8.19. Pentru realizarea schemei de circuit și a schemei electrice a fost utilizată platforma [Tinkercad](https://www.tinkercad.com/).

În designul acestui program sunt incluse două funcții definite de utilizator, void switchLEDs(int) ce

realizează aprinderea și stingerea LED-urilor într-o manieră succesivă și `int getNote(int)` pentru determinarea frecvenței notei ce va fi redată în urma schimbării valorilor unuia dintre cele 4 potențiometre.

```
void switchLEDs(int pattern)
{
    switch (pattern)
    {
        case 0:
            digitalWrite(LED4, LOW);
            digitalWrite(LED1, HIGH);
            break;
        case 1:
            digitalWrite(LED1, LOW);
            digitalWrite(LED2, HIGH);
            break;
        case 2:
            digitalWrite(LED2, LOW);
            digitalWrite(LED3, HIGH);
            break;
        case 3:
            digitalWrite(LED3, LOW);
            digitalWrite(LED4, HIGH);
            break;
    }
}
```

```
int getNote(int pattern)
{
    if (pattern == 0)
    {
        note = map(analogRead(A0), 0, 1023, lowFreq, highFreq);
    }
    else if (pattern == 1)
    {
        note = map(analogRead(A1), 0, 1023, lowFreq, highFreq);
    }
    else if (pattern == 2)
    {
        note = map(analogRead(A2), 0, 1023, lowFreq, highFreq);
    }
    else if (pattern == 3)
    {
        note = map(analogRead(A3), 0, 1023, lowFreq, highFreq);
    }

    return note;
}
```

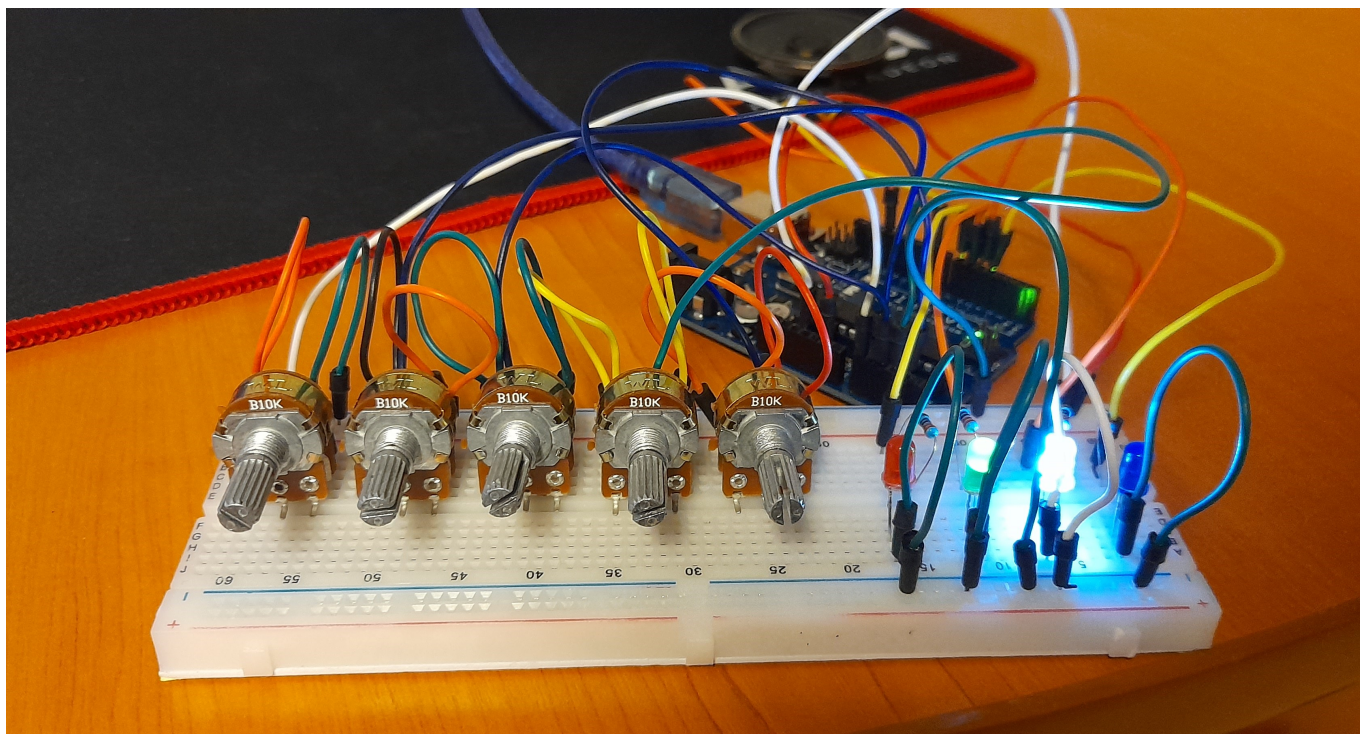
În interiorul funcției `void setup()` se realizează inițializarea pinilor digitali la care sunt conectate LED-urile.

```
pinMode(LED1, OUTPUT);  
pinMode(LED2, OUTPUT);  
pinMode(LED3, OUTPUT);  
pinMode(LED4, OUTPUT);  
digitalWrite(LED1, LOW);  
digitalWrite(LED2, LOW);  
digitalWrite(LED3, LOW);  
digitalWrite(LED4, LOW);
```

În cadrul funcției void loop() se verifică valoarea BPM-ului, setată de potențiometru, ce poate lua valori între 80 și 4000, după care se apelează funcțiile switchLEDs și getNote pentru a realiza funcționalitățile principale de redare a sunetului și a jocului de lumini.

```
switchLEDs(pattern);  
note = getNote(pattern);  
  
pattern = (pattern + 1) % 4;  
if (note > lowFreq + 10)  
{  
    tone(speaker, note);  
}
```

Rezultate Obținute



Download

Arhiva ce conține codul sursă al proiectului, precum și imaginile schemelor de cricuit poate fi descărcată de [aici](#).

Bibliografie/Resurse

Resurse Software:

<https://www.arduino.cc/reference/en/>

Resurse Hardware:

<https://www.optimusdigital.ro/ro/56-toate-produsele>

<https://create.arduino.cc/projecthub/costantinorizzuti/step-sequencer-19a9d7>

<https://create.arduino.cc/projecthub/eEdizon/arduino-step-sequencer-afea0e>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/avaduva/iulian.docan>

Last update: **2022/06/01 01:40**

