Sintetizator

Nume: POPESCU Radu-Andrei

Grupa: 336CC

Introducere

Proiectul meu este un sintetizator care generează semnale audio cu mai multe opțiuni de sunete ce pot fi selectate prin intermediul unui display LCD și a unui potențiometru. Sunetul poate fi redat prin difuzoarele integrate sau prin intermediul unui output de tip Jack. De asemenea, este prevăzut cu un input USB pentru a putea fi conectat la o clapă MIDI externă.

Scopul proiectului este de a crea un instrument muzical portabil care să ofere utilizatorilor o varietate de opțiuni sonore. Este util pentru o gamă largă de utilizatori, de la muzicieni amatori la profesioniști, și poate fi utilizat în diferite contexte, cum ar fi înregistrări sau spectacole live.

Descriere generală

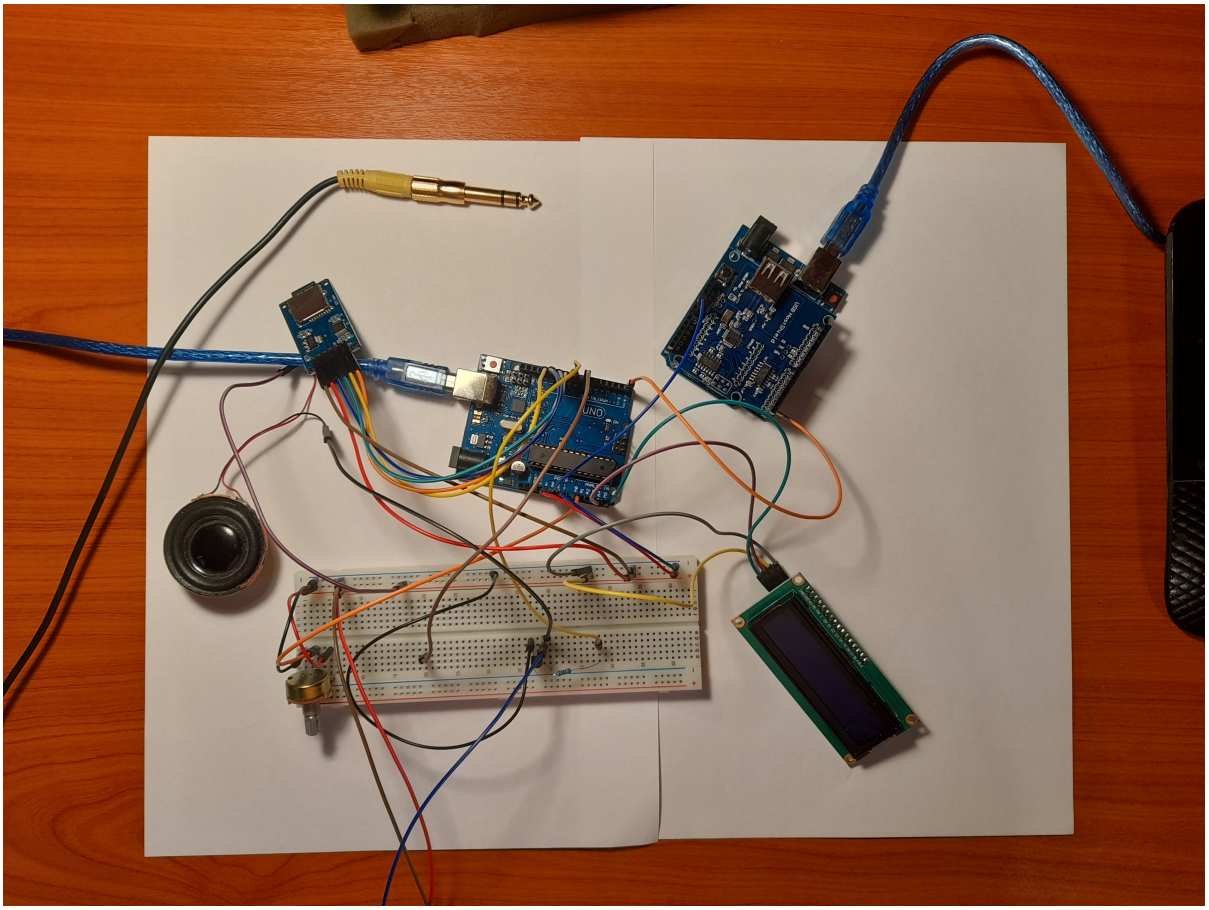
Un instrument muzical portabil, care oferă o varietate de opțiuni sonore și posibilități de personalizare. Este ideal pentru utilizatorii care doresc să experimenteze cu sunete noi și să creeze muzică într-un mod creativ și interactiv.

Funcționarea proiectului începe cu selectarea opțiunii de sunet de pe display-ul LCD prin intermediul potențiometrului. Aceasta poate include selectarea unui sunet presetat sau crearea unui sunet personalizat prin ajustarea parametrilor de sunet, cum ar fi volumul, tonul. Odată selectat sunetul, semnalul audio este generat și reprodus prin difuzoarele integrate sau prin intermediul output-ului de tip Jack, care poate fi conectat la un amplificator sau alt dispozitiv audio. Pentru a “cânta” la acest instrument, trebuie conectată o clapă MIDI externă în inputul USB pus la dispoziție.

Hardware Design

Listă de piese:

- Placă Arduino Uno R3 (microcontroller board)
- Modul USB Host Shield (pentru conectarea la clapa MIDI prin USB)
- Difuzoare (pentru redarea sunetului)
- Modul MicroSD Card (pentru a stoca sunetele)
- Potențiometre (pentru controlul volumului și al altor parametri de sunet)
- Jack audio de ieșire (pentru conectarea difuzorului sau a altor dispozitive audio)
- Display LCD (pentru afișarea opțiunilor de sunet și a altor informații relevante)



Software Design

Mediul de dezvoltare

- Arduino IDE

Biblioteci folosite

- Arduino.h

- usbh_midi.h
- usbhub.h
- SD.h
- TMRpcm.h
- SPI.h
- Wire.h
- LiquidCrystal_I2C.h
- input_converter.h (bibliotecă proprie)

Implementare

- Pentru citirea input-ului MIDI:

```
...
void MIDI_poll();

void setup()
{
  _MIDI_SERIAL_PORT.begin(31250);

  if (Usb.Init() == -1) {
    while (1);
  }
  delay( 200 );
}

void loop()
{
  Usb.Task();

  if ( Midi ) {
    MIDI_poll();
  }
}

// Poll USB MIDI Controler and send to serial MIDI
void MIDI_poll()
{
  uint8_t outBuf[ 3 ];
  uint8_t size;

  do {
    if ( (size = Midi.RecvData(outBuf)) > 0 ) {
      if (outBuf[0] == 144) { // 144 reprezintă codul pentru când se apasă
clapa respectivă
        _MIDI_SERIAL_PORT.print(outBuf[1], DEC);
        _MIDI_SERIAL_PORT.println();
      }
    }
  } while (size > 0);
}
```

```
...
```

- Pentru citirea input-ului de pe cealaltă placă Arduino:

```
...
if (Serial.available() > 0) {
    // Read the incoming data
    c = Serial.read();
    if (c != '\n'){
      note[i++] = c;
    } else {
      note[i] = c;
      i = 0;
      note_int = atoi(note);
      Serial.print(note_int);
}
...
```

- Pentru a cânta nota primită ca input:

```
...
if (note_int != 0){
    sprintf(file, "%s%d.wav", MIDI_to_note(note_int), curr_preset);
    Serial.print(file);
    Serial.println();
    tmrpcm.play(file);
}
...
```

- Pentru a citi valoarea potențiometrului și pentru a afișa preset-ul ales:

```
...
void loop() {
  val = analogRead(A0);
  val = map(val, 0, 1020, 1, 6);

  if (val != curr_preset && val != 6){
    curr_preset = val;
    printPreset();
  }
}
...
void printPreset(){
  clearCharacters();
  lcd.setCursor(0,1);
  if (curr_preset == 5)
    lcd.print("Drums");
  else lcd.print(curr_preset);
}
...
void clearCharacters()
{
```

```
    for (int i = 0; i < 10; i++)
    {
    lcd.setCursor (i,1); //
    lcd.write(254);
    }

}
...
```

De asemenea, pentru a converti valoarea input-ului MIDI într-o notă muzicală, am creat propria bibliotecă. **input_converter.cpp:**

```
#include "input_converter.h"

char* MIDI_to_note(int note) {
    if (note == 60)
        return "C1";
    if (note == 61)
        return "C#1";
    if (note == 62)
        return "D1";
    if (note == 63)
        return "D#1";
    if (note == 64)
        return "E1";
    if (note == 65)
        return "F1";
    if (note == 66)
        return "F#1";
    if (note == 67)
        return "G1";
    if (note == 68)
        return "G#1";
    if (note == 69)
        return "A1";
    if (note == 70)
        return "A#1";
    if (note == 71)
        return "B1";
    return "";
}
```

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună .

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** şi **Resurse Hardware**.

Export to PDF