

MIDI CONTROLLER

Autor: Al Erksousi Mohamad

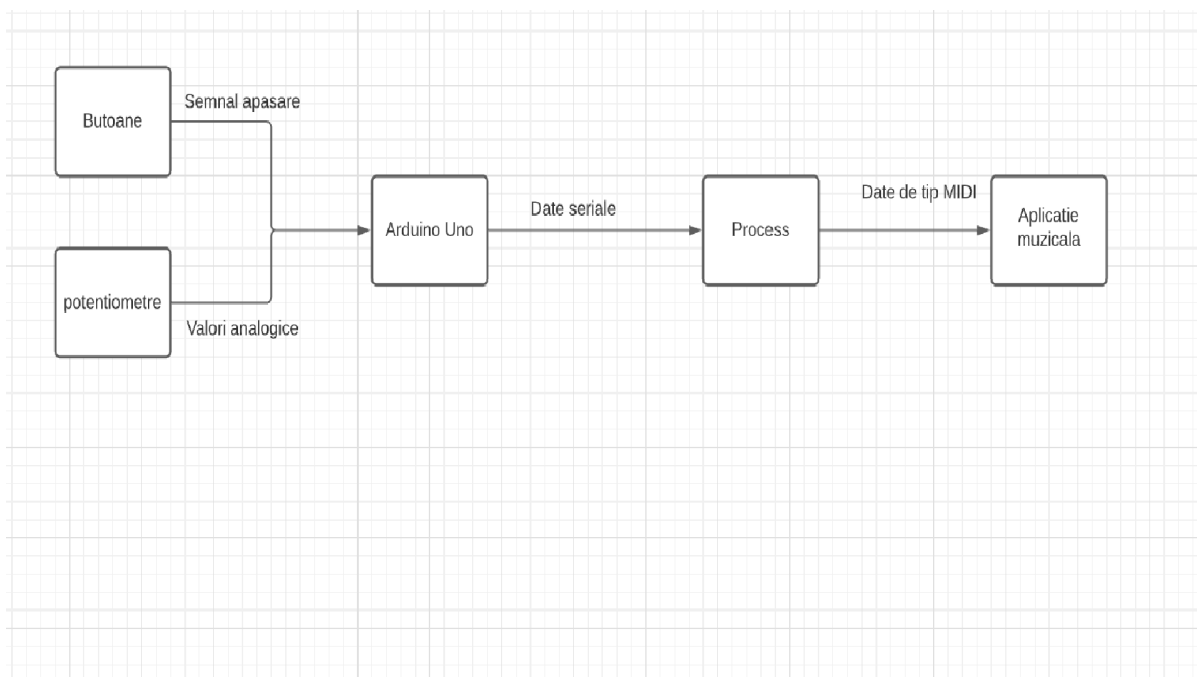
Grupa: 336CA

Introducere

Proiectul consta in realizare unui MIDI(Musical Instrument Digital Interface) Controller, un aparat ce trimite semnale unui sintetizator/alt instrument muzical asemanator, oferind posibilitatea atat de a declansa diferite tonuri, cat si de a schimba parametrii sunetului(Intensitate, frecventa, etc..) prin intermediul unor potentiometre.

Descriere Generala

Pe post de instrument muzical voi folosi o aplicatie de tip sintetizator, cel mai probabil loopBe sau FL studio. Declansarea unui sunet se va face prin intermediul butonaelor, facandu-se o mapare intre inputul primit si sunetul care va fi declansat in aplicatie. Pe langa butoane, aparatul va avea si un numar de potentiometre care vor putea fi folosite pentru a regla diferite efecte sonore si proprietati precum volumul si frecventa.

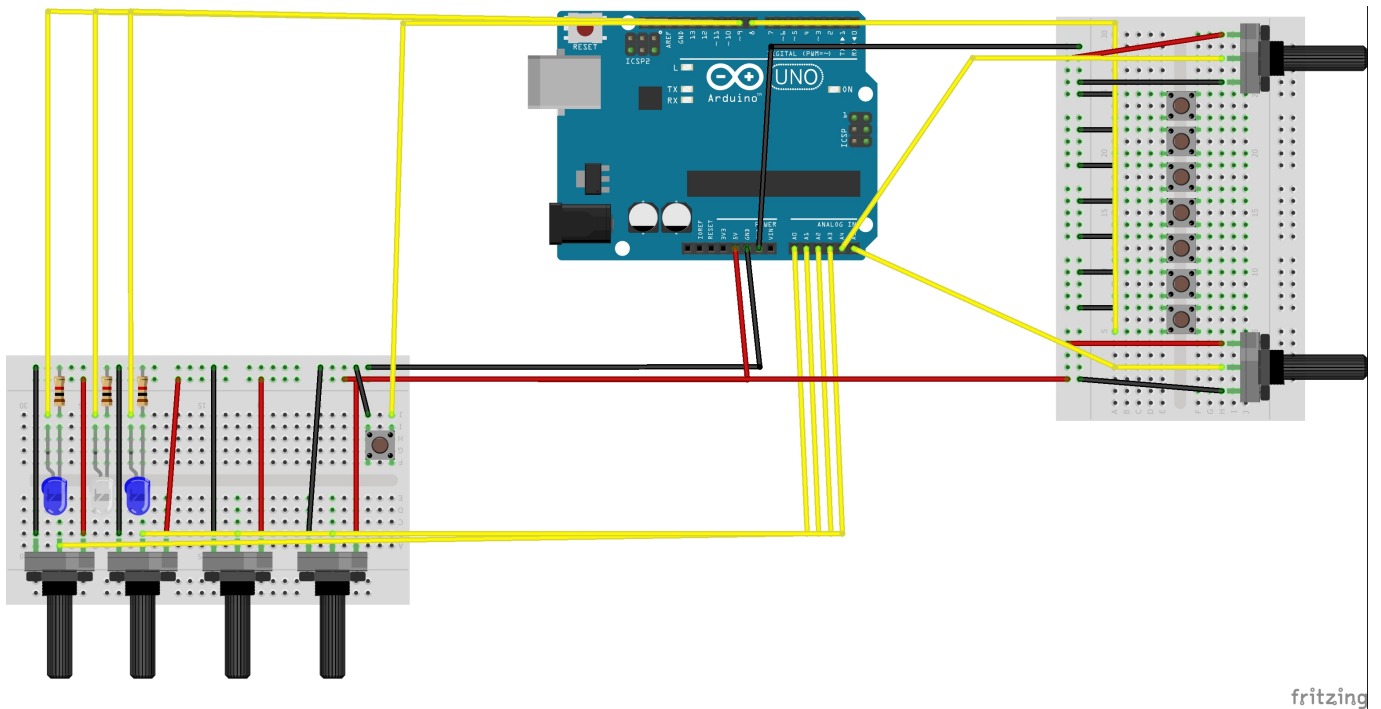


Schema bloc

Hardware Design

Componentele hardware necesare sunt următoarele:

- Arduino Uno
- butoane
- potentiometre
- fire
- Led-uri
- rezistente(pentru conectarea led-urilor)



Schema electrica

Software Design

Codul este scris in Arduino IDE, iar pe langa bibilotecile standard Arduino acesta foloseste biblioteca MIDI.h, biblioteca folosita pentru a trimite datele intr-un format MIDI-convertibil. Codul este relativ simplu, acesta procedeaza astfel:

1- citeste datele de la butoane cu digitalRead si datele de la potentiometre cu analogRead

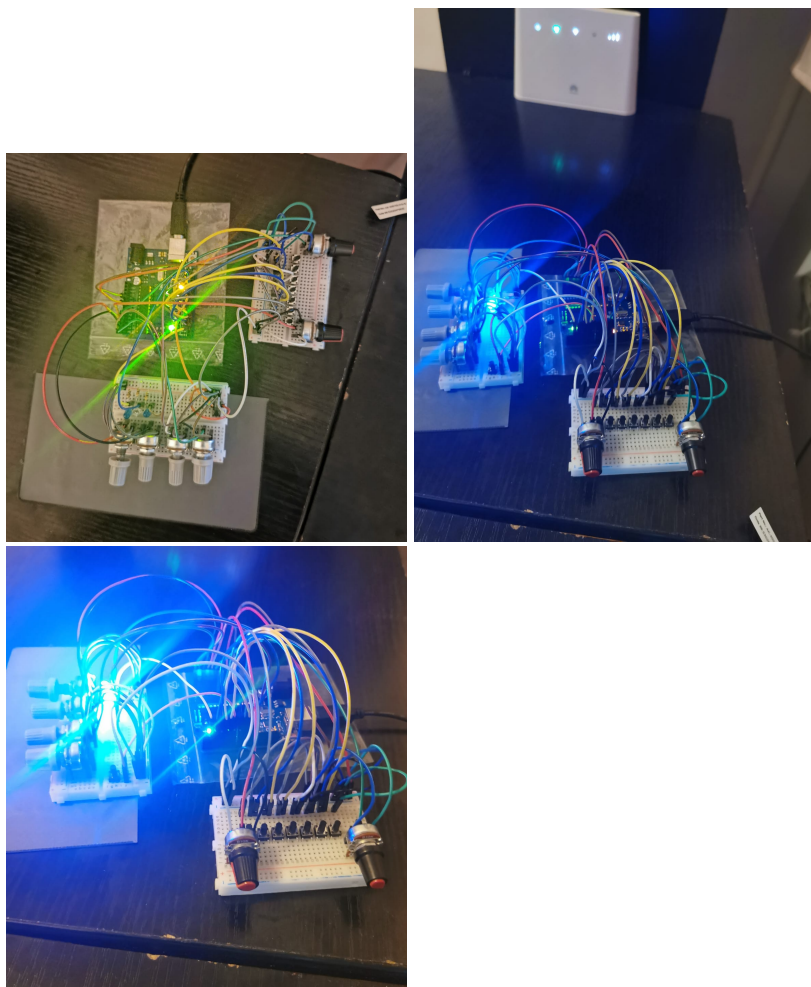
2- pentru butoane va verifica inputurile la fiecare trecere a unei perioade de delay si le va trimite ca note folosind functia sendNoteOn(). daca se apasa butonul de pauza si ledurile sunt active se vor dezactiva, atfel daca se apasa alt buton si ledurile sunt inactive se vor activa

3- pentru potentiometre se va folosi acelasi principiu de verificare dupa delay, diferenta este ca datele nu se vor trimite folosind sendNoteOn(), ci folosind sendControlChange() pentru a putea modifica controale precum efectele si filterele. Datele trimise de la potentiometru vor trebui scalate pentru a se incadra in intervalul 0-127 deoarece acesta este intervalul notelor MIDI

Pe langa scrierea codului am avut nevoie de diverse aplicatii pentru functionarea proiectului. In primul rand, un controller MIDI nu isi are rostul daca nu trimite datele unui instrument muzical. Neavand instrumente pe care sa le pot folosi, am optat la folosirea unei aplicatii de tip sintetizator apeland la programul Ableton Live pentru generarea muzicii. Am ales Ableton Live in principal deoarece este cel mai MIDI-friendly dintre toate aplicatiile de acest tip.

Deoarece datele trimise de program sunt date seriale, nu date MIDI, acestea nu puteau fi trimise in mod direct aplicatiei, fara sa treaca printr-un convertor care sa le transforme in date MIDI. Pentru a face aceasta conversie am folosit Hairless MIDI, o aplicatie care primeste date pe un port serial si le converteste pentru a le transmite mai departe pe un port MIDI. Windows nu are implementate optiuni pentru crearea unui port MIDI, deci am folosit aplicatia LoopMIDI pentru a crea un astfel de port.

Rezultate Obtinute



!Router sold separately!

Concluzii

Pe langa experienta dobandita in lucrul cu Arduino, am invatat multe despre datele MIDI si modul in care acestea sunt transmise unui mediu muzical. Deasemenea am invatat putin despre lucrul cu potentiometre si ce tipuri de potentiometre sunt potrivite pentru anumite tensiuni.

Demo

<https://youtu.be/MG1ahkrgh0Y>

Download

https://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/midi_controller?do=export_pdf

Bibliografie

https://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/midi_controller

https://github.com/FortySevenEffects/arduino_midi_library/blob/master/src/MIDI.h

<https://youtu.be/BzLKtor-6Vg>

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/midi_controller



Last update: **2021/06/03 13:48**