Sintetizator MIDI

Introducere

Proiectul este un sintetizator cu input MIDI. Acesta va primi semnale MIDI de la un controller (in cazul meu o claviatura) si va genera unde audio pe baza notei apasate. Totodata sintetizatorul va putea reda fisiere MIDI de pe un card SD. Un ecran va afisa in timp real ce nota este redata si in ce mod de input este dispozitivul.

Descriere generală

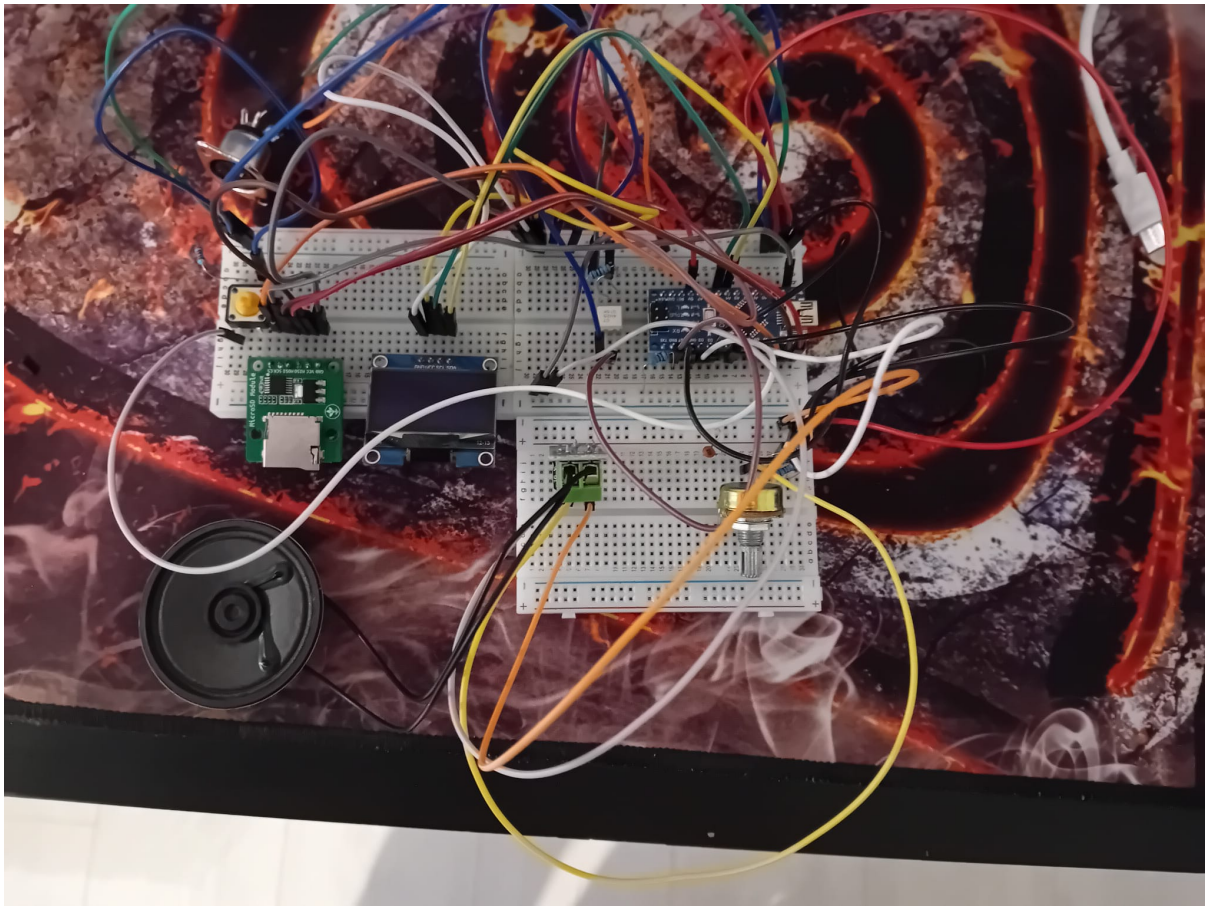
- Clapa va trimite semnale MIDI catre USART.
- Microcontrollerul va procesa aceste semnale, activand un semnal PWM corespunzator notei primite, iar ieșirea va fi transmisa catre un difuzor sau o mufa audio.
- Un buton va fi utilizat pentru a comuta intre doua moduri de input: claviatura live sau citirea unui fisier MIDI de pe cardul SD.
- Ecranul va comunica prin protocolul I2C.
- Cititorul de carduri SD va comunica cu microcontrollerul prin SPI.
- Circuitul de iesire consta intr-un amplificator si un filtru trece-jos RC pentru a reduce zgomotul din semnalul audio si a proteja pinii de PWM.

Hardware Design

Lista componente

Nr.	Componentă	Cantitate	Info
1	Arduino Nano	1	Microcontroler
2	Conector DIN 5 pini mama	1	Input pentru clapa
3	octocuplor 4n25	1	Pentru a proteja arduino-ul folosit in circuitul de input
4	dioda de protectie	1	Pentru a proteja led-ul octocuplorului
5	rezistente de pull up	2	Pentru buton si inputul midi
6	Modul micro sd	1	interfata cu sd cardul SPI
7	Ecran oled	1	I2C
8	Rezistente 10K ohmi	3	Protejeaza pinii de output PWM in caz de numai unul genereaza semnal

9	Potentiometru 100k ohmi	1	Pentru filtru trece jos
10	Condensator 10nF	1	Pentru filtru trece jos
11	Modul amplificator lm386	1	Amplifica semnalul pentru a proteja pinii de output
12	Speaker 1W	1	-



Schema Electrica

- Circuit midi input

Am folosit un octocuplor pentru a proteja pinii de intrare ai Arduino-ului. Atunci cand clapa trimite semnal LOW, pinul 5 (data line) al mufei este conectat la masa iar pinul 4 (power line) la 5V, ledul octocuplorului se aprinde. Acest lucru face ca fototransistorul din interiorul octocuplorului sa conduca, astfel pinul RX al Arduino-ului va primi un semnal de LOW. In mod similar se intampla si in cazul semnalului HIGH.

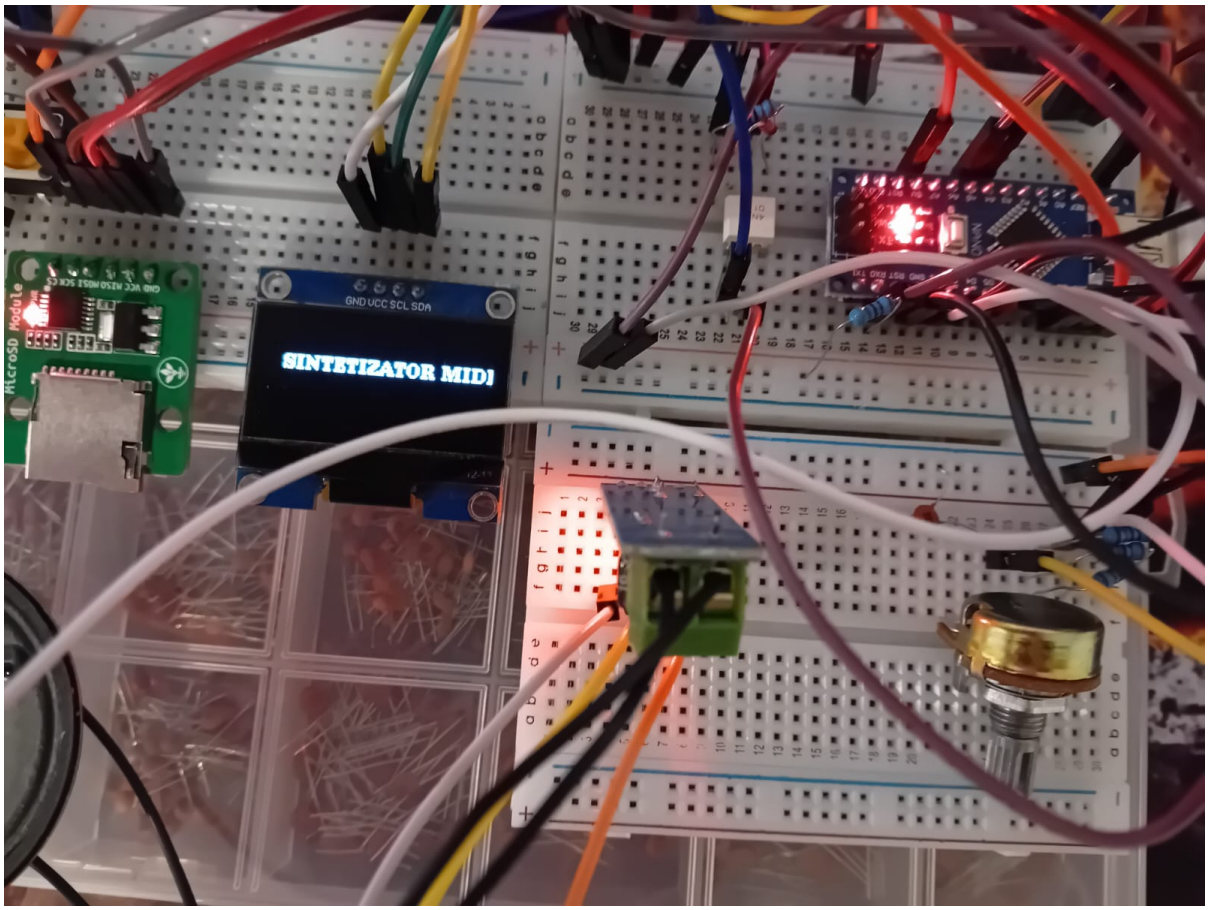
- Circuit PWM out:

Deoarece Arduino-ul are 3 timere, pot folosi 3 pini pentru a genera semnale PWM diferite. Acestia sunt conectati la un filtru trece-jos cu rezistenta variabila prin 3 rezistente de 10K ohmi pentru a proteja pinii de output. Filtrul trece-jos poate netezi o undă pătrată, transformând-o într-o undă triunghiulară și modificând astfel sunetul de la unul aspru la unul mai placut. In final, semnalul intra intr-un amplificator LM386 pentru a putea fi auzi prin difuzor si pentru a proteja pini (fiind un speaker

de 8 ohmi curentul necesar ar fi prea mare pentru pinii arduinoului fara un amplificator).

Pini Folositi

Pin Arduino	Conectat la	Descriere
5V	Vcc	Alimentare
GND	Toate GND-urile	Masă comună
D0 (RX)	Iesire Octocuplor	Input semnal midi
D1 (TX)	—	
D3	R4	Semnal PWM către difuzor
D5	R5	Semnal PWM către difuzor
D9	R6	Semnal PWM către difuzor
D2	Buton S1	Intrare digitală
D10	CS (Pin 6 MicroSD)	SPI chip select
D11	MOSI (Pin 4 MicroSD)	SPI MOSI
D12	MISO (Pin 3 MicroSD)	SPI MISO
D13	SCK (Pin 5 MicroSD)	SPI Clock
A4 (SDA)	SDA OLED	Comunicare I2C (date)
A5 (SCL)	SCL OLED	Comunicare I2C (ceas)



Dovada functionare ecran.

Aici puneţi tot ce ţine de hardware design:

- listă de piese
- scheme electrice (se pot lua şi de pe Internet şi din datasheet-uri, e.g. http://www.captain.at/electronic-atmega16-mmc-schematic.png)
- diagrame de semnal
- rezultatele simulării

Software Design

Am folosit Arduino IDE pentru mediul de dezvoltare.

Biblioteci folosite:

- **SdFat.h**, **MD_MIDIFile.h**:

Pentru a citi si procesa fisiere MIDI de pe cardul SD.

- **SSD1306Ascii.h**, **SSD1306AsciiWire.h**:

Pentru a afisa informatii pe ecranul OLED.

Structura cod:

Sistemul are 3 stari:

1. **MANUAL** - sintetizatorul primeste comenzi MIDI de la claviatura si ecranul OLED afiseaza informatii despre notele primite.
2. **LOADING** - sintetizatorul citeste fisierul MIDI de pe cardul SD si il incarca in memorie.
3. **PLAYING_SD** - sintetizatorul reda fisierul MIDI incarcate de pe card.

Initialiare: In Setup() se initializeaza timerul pentru PWM, interuptul de pe pinul 2, cardul SD, obiectul MIDI (pentru redare din fisier MIDI) si ecranul. De asemenea, se seteaza starea initiala a sistemului la MANUAL.

Loop:

1. **MANUAL**: se citesc 3 bytes pe usart si functia handleMidi() seteaza frecventa timerul si afiseaza informatiile pe ecran.
2. **LOADING**: se citeste fisierul MIDI de pe cardul SD si se incarca in obiectul MIDI, dupa programul trece in starea PLAYING_SD automat.
3. **PLAYING_SD**: se apeleaza functia getNextEvent() pentru a reda urmatorul eveniment MIDI din fisierul incarcat. Astfel se simuleaza primirea de comenzi MIDI de la claviatura.

Trecerea intre stari se face prin apasarea butonului de pe pinul 2, care declanseaza un interupt. Aceasta trece de la starea MANUAL la LOADING sau de la starea LOADING, PLAYING_SD la MANUAL. Este implementata si debouncing pentru a evita treceri multiple intre stari la o singura apasare a butonului. Totodata la apasarea de 2 ori a butonului sistemul trece in **teaching mode** atunci cand reda fisiere, reducand viteza de redare a fisierului si afisand informatii despre note pe ecran.

Functia **showNoteName()** calculeaza numele si octava notei primite si le afiseaza pe ecran.

Notiuni din laborator aplicate:

- PWM: pentru a genera semnalul audio.
- Interupt: pentru a schimba starea sistemului la apasarea butonului.
- USART: pentru a primi comenzi de la clapa.

cod sursa demo

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării şi surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi şi structuri pe care plănuiţi să le implementaţi
- (etapa 3) surse şi funcţii implementate

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

Desi nu am putut implementa de la 0 citirea fisierelor midi, proiectul m-a ajutat sa invat mai mult despre acest protocol si despre cum functioneaza clapa digitala. Am învatat cum sunt interpretate comenzile MIDI și cum acestea pot fi folosite pentru a genera sunete în timp real. De asemenea, am avut ocazia sa experimentez cu un filtru trece-jos RC si sa reusesc sa trec peste bug-uri hardware.

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