

# Identificator Note Muzicale

## Introducere

Proiectul este un instrument ce masoara frecventa unor note muzicale inregistrate. El stocheaza ce aude dupa ce este pornit, analizeaza frecventa sunetelor, reda frecventa pe un ecran LCD, apoi trimite frecventa ca nota MIDI pe interfata USB pentru procesare la o statie.

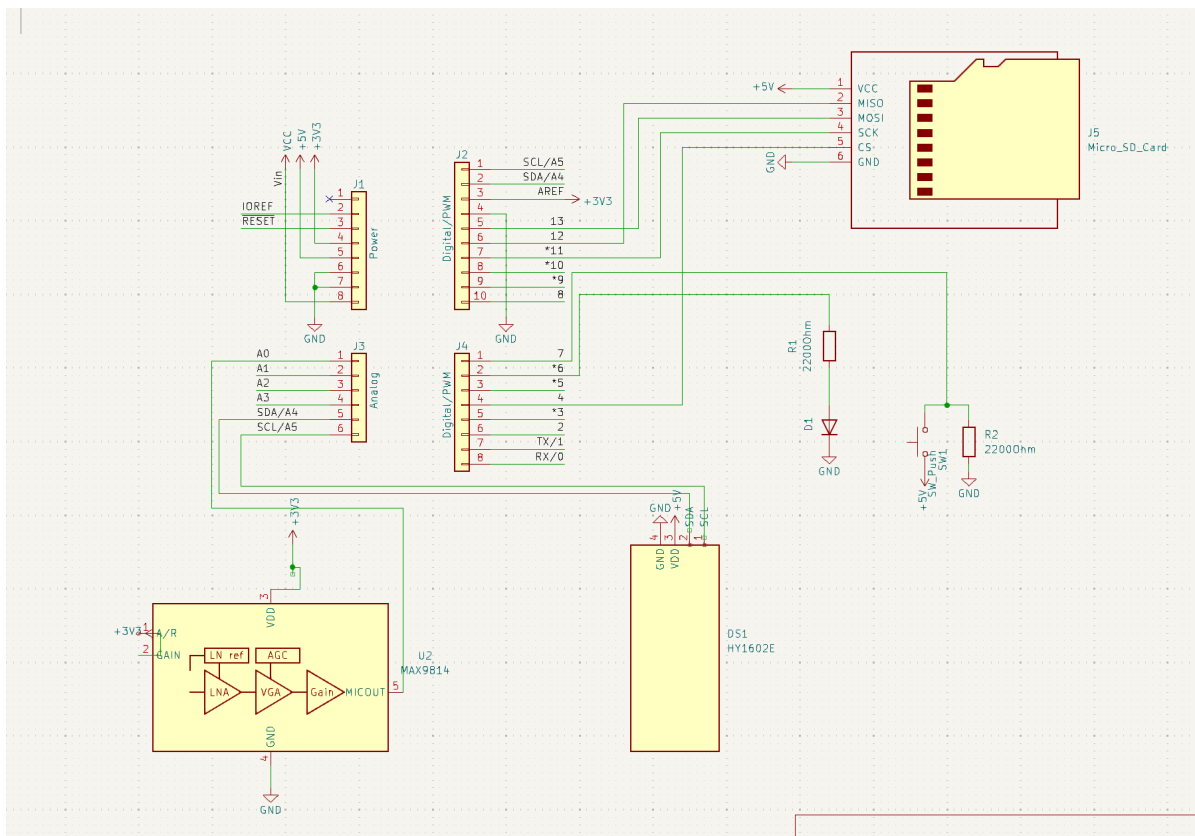
El este util pentru acordarea instrumentelor muzicale, am pornit de la ideea de aparat de acordat o chitara. Poate fi folosit si pentru transmitere live a muzicii cantate pe o aplicatie Desktop, sau in orice aparat ce poate primi evenimente MIDI.

## Descriere generală

Schema bloc:



Schema electrica:

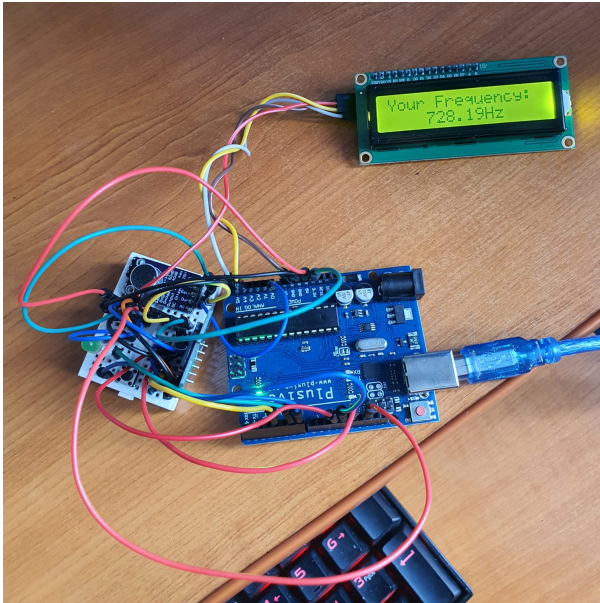


Placa Arduino UNO va calcula frecventa, va controla componentele individuale si le va alimenta. Microfonul capteaza sunetele, ale caror frecvente vor fi stocate pe cardul SD. Rezultatele analizei vor fi afisate pe display-ul LCD. Daca placa este conectata la o interfata USB-MIDI, ea va putea transmite frecventele direct catre un soft de procesare audio. Butonul de pe breadboard trimite un eveniment MIDI ce schimba instrumentul curent.

Placa Arduino UNO este alimentata de la o baterie de 9V, sau prin USB. Componentele sunt alimentate de catre pinii de 5V si 3.3V de pe Arduino.

Microfonul transmite output pe pinul A0. Cititorul de card SD este conectat la pinii SPI. CS la 4, SCK la 11, MOSI la 13 si MISO la 12. Ecranul LCD 1602 foloseste un modul I2C. Pinii SDA/SCL de pe modul sunt conectati la pinii A4/A5 de pe Arduino. Pinul 7 citeste butonul SW1, si pinul 6 alimenteaza un led cand butonul este apasat.

Proiectul in stadiul curent:



Captura output microfon:



## Hardware Design

- Placa Arduino UNO Rev3
- Microfon cu amplificare Adafruit MAX9814
- Cititor card SD ce comunica prin SPI
- Card SD 16GB
- Ecran LCD 1602 cu modul I2C

-Cablu USB

-Baterie 9V

-Buton

-Dioda Verde

-2 X Rezistente 2200 Ohm

-Mini Breadboard

Proiectul este menit sa fie compact, am folosit un breadboard mic alaturi de o placa mica si portabila.

## Software Design

M-am folosit de laboratorul de GPIO pentru pinii de la led si buton, de laboratoarele de timer si ADC pentru masuratori ale pinului A0, si de concepte din laboratoarele de SPI si I2C pentru cititorul de card SD si ecranul LCD cu modul I2c.

Pentru transmitere pe interfata USB-MIDI, am folosit proiectul dualMocoLUFA de pe github, cu care am rescris firmware-ul de pe cipul Atmega 16u2 ce controleaza interfata USB. Am pus placa in mod DFU, si cu Atmel FLIP am instalat un sistem de dual boot. Daca pinii MOSI si GND de la pinii ICSP ai Atmega 16u2 sunt conectati cand placa este alimentata, interfata USB poate fi folosita pentru scris cod pe placa. In caz contrar, placa va porni in mod USB-MIDI, si va putea trimite MIDI events.

Pe parcursul proiectului am incercat sa minimizez consumul de memorie, folosind inttypes pentru flexibilitate.

Aparatul analizeaza in continuu sunetele pe care microfonul le capteaza. Acesta este legat la pinul A0, pe care se face o conversie DAC cu precizie pe 8 biti. Conversia se face la interrupt cu o frecventa de 9615Hz.

Dupa ce se capteaza un numar setat de valori, acestea sunt trimise la o functie ce calculeaza frecventa fundamentala pe baza valorilor. Algoritmul este unul de autocorelatie cu interpolare si normalizare. Am ales acest alogritm datorita limitarilor de memorie, nu este la fel de precis dar consuma mult mai putina memorie decat analiza spectrala cu transformata Fourier.

Funcția de autocorelatie:

```
int32_t autocorrelation(int8_t *samples, uint_least16_t t)
{
    int32_t r = 0;
    for (uint_least16_t k = 0; k < no_samples - t; k++)
        r += ((int16_t)samples[k] * samples[k + t]);
    return r;
}
```

Dupa ce se detecteaza o frecventa se va salva in cardul SD. Pentru comunicare se vor folosi SPI.h si SD.h. Frecventele detectate vor fi afisate pe LCD. Comunicare cu LCD va folosi biblioteca LiquidCrystal

I2C. Apoi, notele sunt trimise prin interfata USB la statia conectata.

## Rezultate Obținute

Proiectul identifica bine notele cu frecvente in intervalul 900 - 1600Hz, cu o rata mica de eroare. Acesta detecteaza si sunete mai joase, pana la ~100Hz, dar rezultatele au deseori erori sub 500Hz. Detectia functioneaza mai bine pentru instrumente cu un sunet mai curat, cum ar fi cele de suflat, decat instrumente precum pianul.

Proiectul transmite notele catre statia audio, dar, in lipsa unui algoritm de segmentare a notelor, nu pastreaza mereu corect ritmul sunetelor.

## Concluzii

SRAM-ul mic (2048 bytes) al arduino UNO nu este suficient pentru a efectua o analiza performanta a frecventei. Desi avem o plaja de valori corecte, nu sunt suficiente pentru un aparat de acordat instrumente muzicale.

Este, de asemenea, nevoie de un algoritm ce poate segmenta notele in mod mai corect, pentru a putea trimite catre statie piese muzicale mai complexe.

Ideea este promitatoare, proiectul functioneaza in modul asteptat pentru unele date de intrare, dar este nevoie de mai multa putere pentru a obtine un produs performant.

## Download

[proiect\\_pm\\_jipa\\_alexandru\\_333cc.zip](#)

## Jurnal

## Bibliografie/Resurse

<https://github.com/kuwatay/mocolufa>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/tdicu/alexandru.jipa2803>



Last update: **2024/05/27 09:42**