

Tuner Chitara Electrica

Introducere

Acest proiect isi propune realizarea unui tuner pentru chitari electrice, electro-acustice si electro-clasice. Utilizatorul va avea nevoie doar de un cablu jack-jack pentru a conecta chitara la acordor. Pentru o experienta de utilizare usoara, el va avea la dispozitie 2 butoane de selectie (previous si next string), un ecran LCD, unde vor fi afisate coarda selectata si frecventa tinta, si un LED, care va indica daca coarda este acordata corespunzator.

Descriere generală

Microcontrolerul Arduino va fi componenta principala a proiectului. Acesta va primi input de la chitara prin mufa jack, amplitudinea semnalului fiind marita cu ajutorul amplificatorului operational. Utilizatorul va trebui sa aleaga ce coarda vrea sa acordeze, folosind butoanele si urmarind mesajele de pe ecranul LCD, pentru a stabili frecventa de referinta a semnalului. In functie de cat de cat de dezacordata este coarda, LED-ul se va comporta astfel:

- coarda este acordata corect (frecventa input = frecventa referinta) \Rightarrow LED verde
- coarda trebuie stransa (frecventa input < frecventa referinta) \Rightarrow LED rosu
- coarda trebuie slabita (frecventa input > frecventa referinta) \Rightarrow LED albastru

Se va folosi acordajul standard in Mi:

- Mi - 329.6Hz
- Si - 246.9Hz
- Sol - 196Hz
- Re - 146.8Hz
- La - 110Hz
- Mi - 82.4Hz



Hardware Design



Flow si explicatii:

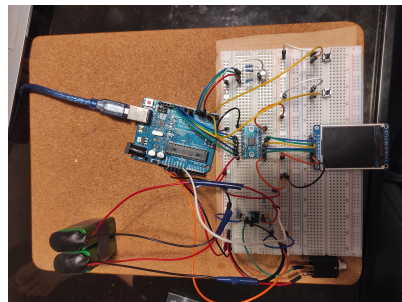
1. Signal input:

- Semnalul de la chitara este preluat prin mufa jack si transmis mai departe la amplificatorul operational.
 - Amplificatorul este alimentat cu 2 baterii de 9V si este montat intr-o configuratie non-inverting (1KOhm pentru gain, 10KOhm pentru feedback), astfel incat semnalul sa fie amplificat de la +/-200mV la +/-2.5V. In continuare, se foloseste un condensator si un divizor de tensiune pentru a centra semnalul in 2.5V in loc de 0V, astfel incadrandu-se in intervalul de 0-5V acceptat de Arduino
2. SPI LCD connection:
- Pentru a realiza conexiunea dintre Arduino si modulul LCD a fost nevoie de un 8bit level shifter, pentru a face conversia de la cei 5V produsii de pini GPIO ai Arduino in 3V3 suportati de LCD

Lista piese:

- Arduino Uno R3
- Mufa Jack 6.3mm mama
- [Op-Amp OP07](#)
- 2 Butoane
- Rezistente
- Condensatoare
- LED RGB
- [Modul SPI LCD ST7735](#)
- [8bit Level Shifter TXS0108E](#)
- Baterii

[Cablaj:](#)



Software Design

[Source Code](#)

Mediu de dezvoltare: PlatformIO (VSCode)

Surse folosite: [Arduino-Frequency-Detection](#)

Codul este structurat pe mai multe fisiere sursa, astfel incat sa exista o separare a componentelor. In mare, tuner-ul este compus din 2 parti software importante.

1. Preluarea input-ului de la utilizator prin butoane, astfel setand si afisand frecventa de referinta si generarea output-ului la ecran
2. Preluarea semnalului analogic de la chitara, determinarea frecventei si compararea acesteia cu frecventa de referinta

Input utilizator:

- Pentru a prelua input-ul de la butoane se folosesc intreruperile INT0 si INT1. Ele modifica o variabila in care se tine minte ce frecventa de referinta se foloseste
- Din `loop()` se scrie pe ecran, folosind biblioteca `Adafruit_ST7735.h` pentru comunicarea prin SPI, ce frecventa (deci ce coarda) este selectata.

Input chitara:


- Implementarea este aproape identica cu cea din sursa mentionata mai sus. Pentru a mari sampling rate-ul de la 8000Hz la ~40KHz, s-a renuntat la folosirea functiei `analogRead()` si s-a scazut rezolutia de la 1023 la 255
- Atunci cand o intrerupere de conversie ADC este generata, se citeste semnalul din ADCH si apoi se calculeaza perioada si amplitudinea semnalului primit, care sunt salvate in variabile dedicate
- In `loop()`, daca amplitudinea semnalului este peste un `threshold` dat, atunci se va calcula frecventa si se va compara cu referinta. In functie de rezultatul comparatiei, led-ul RGB se va aprinde appropriately.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

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

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele 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** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/mdinica/gabriela.limberea>



Last update: **2024/05/26 17:18**