

Acordor pentru chitara acustica

Introducere

Un acordor de chitara acustica pe baza de senzor piezoelectric. Acesta va da feedback pentru fiecare coarda prin intermediul unui display LCD ce comunica prin I2C cu un microcontroller ESP32 C6. Acest acordor va citi informatia cu senzorul direct de pe suprafata instrumentului si va fi configurat pentru acordajul standard (Standard E tuning). Conectivitate bluetooth va fi incorporata pentru a trimite datele primite la un terminal pe telefon/calculator.

Functionalitati cheie:

- Acordaj prin tensiuni piezoelectrice (ADC)
- Feedback prin LCD (I2C)
- Comunicare prin terminal Bluetooth (Serial)

Descriere generală

Proiectul are la baza folosirea protocoalelor studiate la laborator (ADC, I2C, Serial) pentru a oferi o solutie pentru acordarea chitarelor acustice care, de obicei, nu au metode incorporate pentru acordaj sa de conectare la o unitate de procesare audio. In cazul acesta, o doza piezoelectrica poate fi lipita de suprafata instrumentului si sa preia semnalul mecanic, trimitandu-l la microcontroller. Prin software, ideea de baza este descompunerea semnalului in frecventa si compararea frecventei dominante cu una de referinta, precizata de acordajul standard.

Schema bloc a proiectului este urmatoarea:



Hardware Design

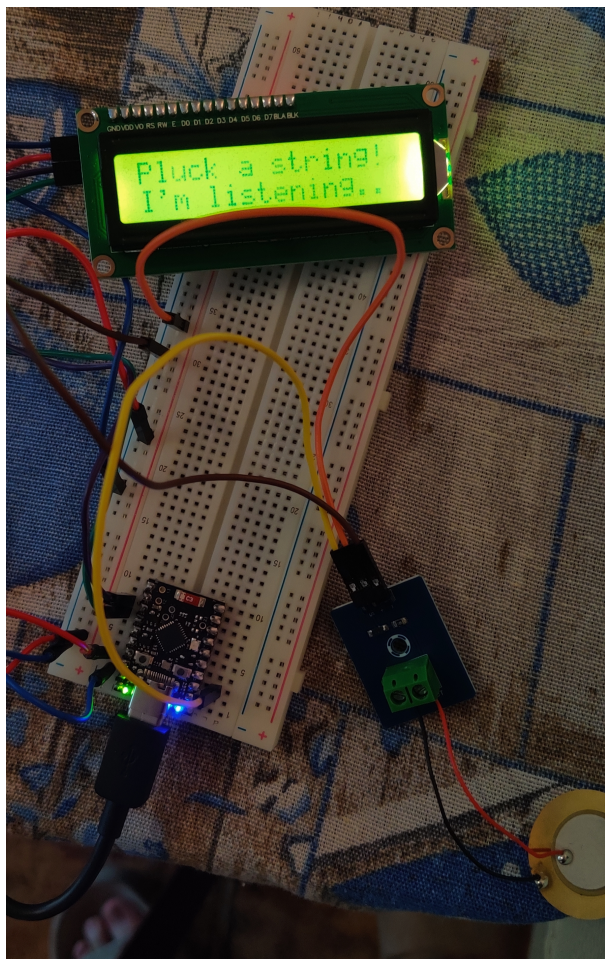
Lista de piese:

- ESP32 C6
- Ecran LCD I2C

- Senzor Piezo
- Breadboard

Partea de hardware a acestui proiect se bazeaza intens pe functionalitatile microcontrollerului si a senzorului piezo electric. Prin intermediul pinului 0 (Analog 0), controllerul citeste o valoarea analogica de la iesirea tip signal a modului de piezo. Rezultatul este printat pe ecranul LCD comunicand prin I2C. Pentru a evita probleme de conflict cu intrariile esp-ului (pin 8 shared pentru led builtin si pin 9 folosit pentru bootstrapping) am decis sa renunt la pinii default pentru I2C si sa folosesc pinii GPIO 14 si 15. Atat senzorul, cat si display-ul sunt alimentate prin intermediul placutei folosind pin-ul de 3V3. Configuratia bluetooth se va face prin intermediu software, deci nu este relevant sa mentionez comunicarea in aceasta sectiune. Nicio alta sursa nu este folosita de-a lungul proiectului.

Configuratia functioneaza cum trebuie, senzorul piezo trebuie bine prins de pragusul chitarei si in momentul lovirii unei corzi citeste o frecventa aproximativa si o printeaza. Se poate vedea comunicarea intre senzor si display (in cazul de mai jos, nicio frecventa relevanta nu este citita) aici:



Schema electrica:



Bill of materials:

- [ESP32 C6 super mini](#)
- [LCD I2C](#)

- [Senzor piezo](#)
- [Breadboard 830 de puncte \(inclus in kit plusivo\)](#)

Software Design

Codul din spatele acestui proiect are la baza citiri repetate de pe pin-ul analog al placii si procesarii intrarii. Fiind un acordor de chitara, nu este intuitiv sau util sa folosim valoarea returnata de analogRead (intreg 0-4095). Pentru a transforma tensiunea citita, este utilizat algoritmul de **Fast Fourier Transform**, cu care putem extrage o frecventa. Pentru a efectua aceasta transformare, la fiecare interval de timp se extrage un batch de sample-uri de tensiune (256 numarul optim) la o frecventa prestabilita (5 kHz), fiecare sample fiind salvat intr-o lista. Asupra listei se face descompunerea rezultata transformarii. Semnalul citit are un DC offset care este extras din fiecare componenta astfel:

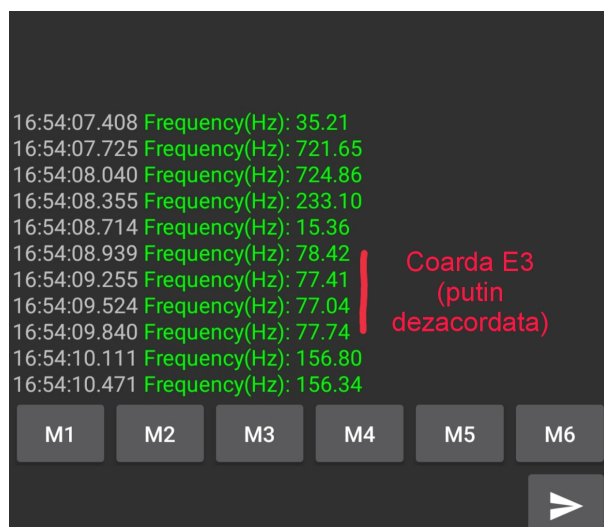
```
vReal[i] -= medie vReal / 256
```

Rezultatul este supus transformarii Fourier, iar frecventa corzii este data de peak frequency.

Pentru a interfata in mod placut frecventa, este folosit un vector de frecvente pentru fiecare coarda, alaturi de un vector cu numele corzilor. se va calcula minimul distantei intre frecventa gasita si toate elementele vectorului. Pozitia gasita ne va da coarda dorita. Daca o frecventa se regaseste la o distanta de $\pm 3\text{Hz}$ este considerata acordata. Altfel, este indicata directia de acordare (higher/lower). Frecventele sub 62Hz sau peste 440Hz sunt considerate zgomot si nu vor fi considerate in procesul de cautare.

Rezultatul este afisat prin I2C catre un ecran LCD.

In cazul in care dorim sa facem debugging pe tuner sau sa intelegem mai bine frecventele in joc, esp32 genereaza un server de BLE prin care utilizatorul se poate conecta remote prin telefon sau laptop. Cu ajutorul unui terminal fiecare frecventa va fi printata la momentul ciupirii unei corzi. Se poate observa comunicarea aici:



```
16:54:07.408 Frequency(Hz): 35.21
16:54:07.725 Frequency(Hz): 721.65
16:54:08.040 Frequency(Hz): 724.86
16:54:08.355 Frequency(Hz): 233.10
16:54:08.714 Frequency(Hz): 15.36
16:54:08.939 Frequency(Hz): 78.42
16:54:09.255 Frequency(Hz): 77.41
16:54:09.524 Frequency(Hz): 77.04
16:54:09.840 Frequency(Hz): 77.74
16:54:10.111 Frequency(Hz): 156.80
16:54:10.471 Frequency(Hz): 156.34
```

Coarda E3
(putin
dezacordata)

M1 M2 M3 M4 M5 M6

Acest lucru se realizeaza prin generarea unui server folosind modulul incorporat din placa pentru a transmite datele in mod eficient si rezonabil din punct de vedere al consumului. Comunicarea este realizata prin UART, datele fiind transmise doar in cazul conexiunii unui utilizator. Comunicarea este

unidirectionala, am restrictionat trimiterea de mesaje catre placuta din partea observatorilor.

Aceasta etapa a fost implementata folosind urmatoarele biblioteci si resurse:

- [ArduinoFFT Library](#)
- [I2C Liquid Crystal Display Library](#)
- [ESP32 BLE UART Communication](#)


Demo-ul proiectului:

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/prj2025/aluca/vlad_andrei.muntean



Last update: **2025/05/24 16:16**