

Karaoke machine

Nume: Oțetea Andrei

Grupa: 331CA

Introducere

Proiectul constă în realizarea unui sistem de karaoke în care utilizatorii primesc puncte. Utilizatorul va putea încărca pe un card microSD melodiile și versurile, urmând ca să vadă versurile pe un ecran LCD, respectiv să audă melodiile într-o boxă legată la un amplificator. De asemenea, utilizatorul va putea primi puncte, dacă cântă cât mai tare în timp ce apar versurile. Utilizatorul va putea trece prin lista de melodii și să selecteze melodia pe care și-o dorește. Acest sistem poate fi atât practic, prin dimensiunile sale reduse, cât și distractiv, putând concura cu alte persoane pentru a obține un scor mai mare. Proiectul este inspirat din jocurile de felul Just Sing sau asemănătoare.

Descriere generală

Schema bloc



Utilizatorul stochează melodiile în formatul WAV și versurile în formatul LRC. La introducerea cardului de memorie în slot, utilizatorul vede pe ecranul LCD melodia curentă și o poate schimba folosind butonul next. De asemenea, cu butonul play/pause poate continua reluarea melodiei, respectiv poate pune pauză la melodia curentă. În plus, butonul on/off mic pornește/oprește microfonul.

Pe ecranul LCD o să apară versurile curente, iar utilizatorul va putea folosi microfonul pentru a obține puncte, dacă cântă în timp ce apar versurile. Scorul este cu atât mai mare cu cât cântă mai tare. Utilizatorul poate obține puncte dacă rulează o melodie și dacă acesta are microfonul pornit.

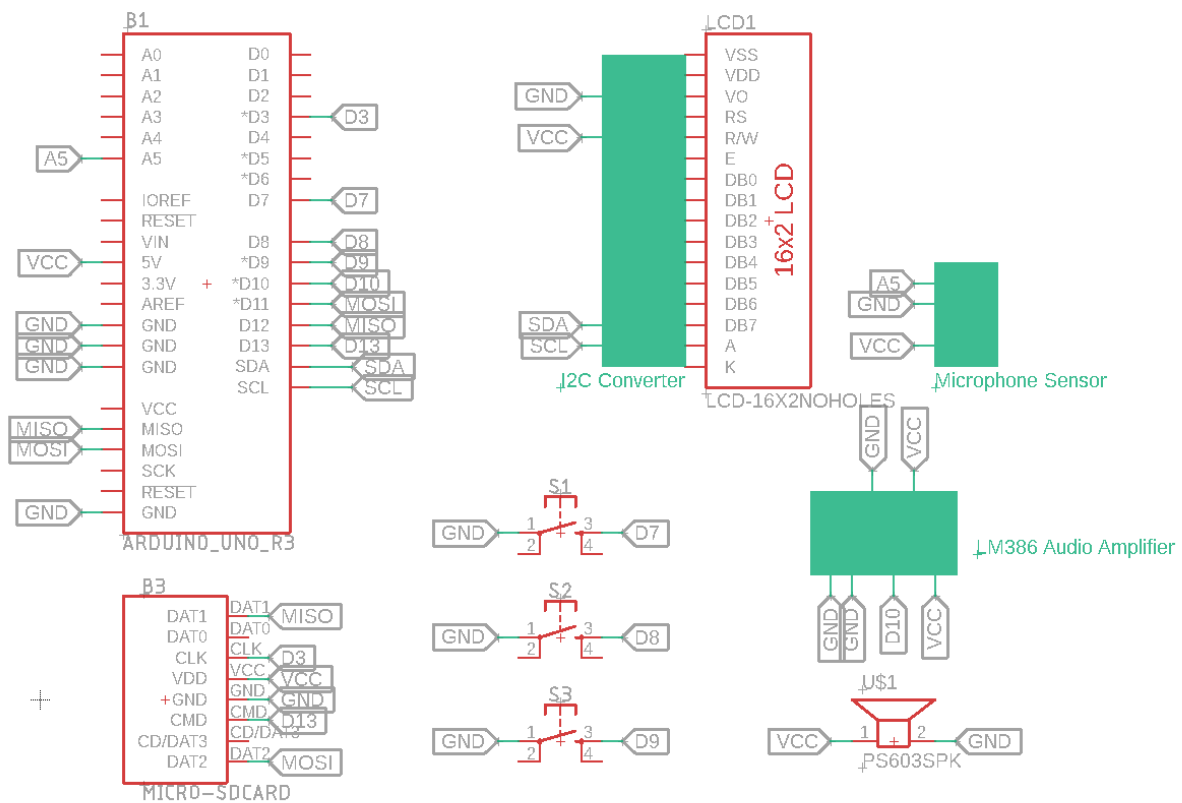
Melodia se va auzi pe o boxă, sunetul fiind amplificat.

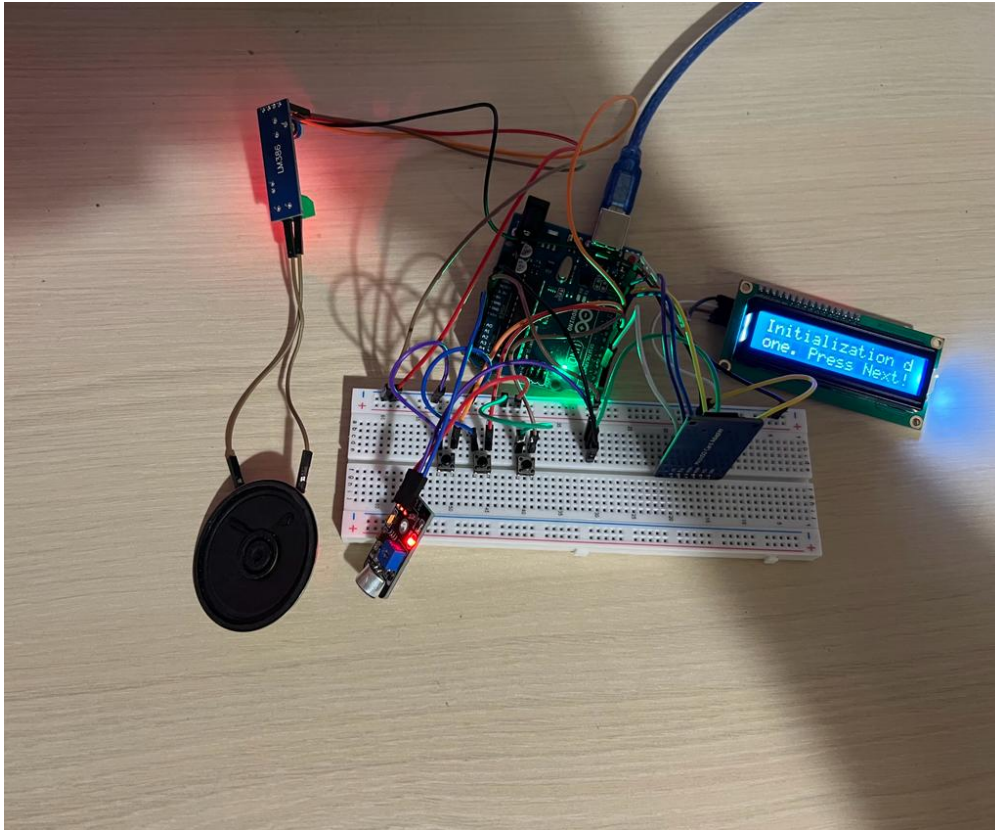
Hardware Design

Listă piese:

- 1 x Arduino Uno
- 1 x Ecran LCD 1602 IIC/I2C
- 1 x Breadboard 830 puncte MB102
- 1 x Amplificator audio LM386
- 1 x Boxă 1W
- 1 x Modul microfon
- 1 x Modul microSD
- 3 x butoane

Schema electrică:





Software Design

Mediul de dezvoltare folosit în realizarea proiectului: Arduino IDE.

Descrierea codului aplicației (firmware)

- **void setup()**- Setez seriala pe 9600 baud, inițializez obiectele TMRpcm și SdFat, inițializez LCD-ul și configurez întreruperile pentru Timer2 la 10ms.
- **ISR(TIMER2_COMPA_vect)** - o dată la 10ms, atunci când e cazul, cresc timpul trecut din melodie și actualizez scorul
- **void loop()**- verific dacă oricare dintre butoane sunt apăstate, dacă nu e pauză verific dacă ar trebui modificat textul de pe LCD și citesc noul text care ar trebui afișat pe LCD, și, dacă primesc comanda **S** pe serială afișez scorul curent
- **void displayTextOnLCD(const char* text)** - afișează textul primit pe ambele linii ale LCD-ului
- **bool isWAV(const char* filename)** - verifică dacă un fișier are extensia .wav.
- **void listFiles()** - afișează fișierele din rădăcina cardului SD
- **void readLyricsLine()** - citește o linie din fișierul de lyrics și extrage versul și timpul când ar trebui să apară; nu citesc tot fișierul din prima, îl citesc pe parcurs fiindcă nu aș avea destul SRAM să-l salvez pe tot
- **unsigned long parseTimestamp(const char* timestampStr)** - transformă un string ce conține ora în formatul MM:SS.MM în milisecunde.
- **void performAction(int buttonIndex)** - în funcție de butonul apăsat, execută acțiunea asociată lui

- mic on/off
- pause/play
- next - de fiecare dată când apăs next trec la următoarea melodie de pe cardul microSD. Dacă am epuizat toate melodiile, o iau de la început. De asemenea, deschid și fișierul cu lyrics, folosind același basename, schimbând doar extensia și afișez pe LCD titlul melodiei curente

Ca să pot folosi biblioteca TMRpcm a trebuit să modific în fișierul de configurare **pcmConfig.h** să decommentez următoarele:

- **#define SD_FULLSPEED** - ca să citesc cât mai repede
- **#define SDFAT** - ca să pot folosi împreună cu biblioteca SdFat
- **#define buffSize 128** - am crescut buffer-ul

Setup date microSD

Pentru a stoca fișierele audio am ales formatul WAV și am comprimat fișierele cu următoarele setări pe site-ul <https://audio.online-convert.com/convert-to-wav>:

- Rezoluție biți: 8 Bit
- Rata de eșantionare: 16000 Hz
- Canale audio: mono
- Format PCM: PCM U8

Pentru a stoca lyrics am folosit fișiere LRC; acestea stochează versurile în acest format:

```
[00:33.95]We are the champions, my friends  
[00:41.45]And we'll keep on fighting till the end
```

unde 00:33.95 e timpul când ar trebui să apară, 0 minute, 33 de secunde și 95 de milisecunde, respectiv We are the champions, my friends e versul. Parsarea versului se face în **readLyricsLine**.

Rezultate Obținute

Concluzii

Proiectul în mare a ieșit cum mi-am dorit, sunt sigur că mai pot fi aduse îmbunătățiri. Lucrând la el mi-am însușit mai bine lucrurile învățate la lab.

Singura problemă pe care am întâmpinat-o a fost atunci când am încercat să fac întreruperi pe **Timer0** fiindcă mergea în jur de un minut, după se bloca orice interacțiune cu placa (nu mai mergea niciun buton, nu se mai afișa nimic pe serială sau pe LCD, millis() făcea overflow etc.), așa că am folosit **Timer2**.

Download

[proiect_pm_karaoke_machine_andrei_otetea.zip](#)

Jurnal

- 07.05 - Crearea paginii de proiect și scrierea documentației
- 21.05 - Adăugare schemă electrică
- 28.05 - Adăugare descriere software
- 29.05 - Retușare pagină wiki

Bibliografie/Resurse

Datasheet ATmega328P:

https://ocw.cs.pub.ro/courses/_media/pm/atmel-7810-automotive-microcontrollers-atmega328p_datasheet.pdf

Serial: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>

LiquidCrystal_I2C: <https://reference.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>

SdFat: <https://reference.arduino.cc/reference/en/libraries/sdfat/>

TMRpcm: <https://reference.arduino.cc/reference/en/libraries/tmrpcm/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/karaoke-machine>



Last update: **2023/05/29 22:33**