

Music Player

Autor: [Hermina-Maria Matei](#)

Introducere

Descriere

Proiectul reprezintă o implementare a unui Music Player care canta melodiile stocate pe un card SD. Utilizatorul are posibilitatea de a observa numele melodiei pe ecranul LCD, de a trece la melodia urmatoare/precedenta si a pune pe pauza melodia curenta prin intermediul butoanelor.

Utilitate

Ca utilitate, acest Music Player poate fi folosit de orice ascultator de muzica pentru a se relaxa in timpul liber. Faptul ca muzica este stocata pe un card SD ofera un mod rapid de modificare a playlistului si satisface dorinta utilizatorului de a stoca un numar mare melodii.

Descriere generală

Music Player-ul porneste automat cand este conectat la sura de curent si incepe prin a canta prima melodie, in cazul in care cardul SD este introdus in SD Card Reader Module. In cazul in care cardul SD nu este introdus, pe Serial Monitor se afiseaza "SD failed".

Utilizatorul poate pune pe pauza melodia curenta si o poate relua din punctul in care aceasta a fost oprita prin intermediul butonului de pause. Aceasta poate trece la media urmatoare folosind butonul next si se poate intoarce la melodia anterioara folosind butonul prev.

Important de mentionat este faptul ca melodiile care se afla pe cardul SD trebuie sa fie in format ".wav", avand un Sampling Rate de 16000Hz, canalul audio de tip Mono si Bit Resolution setat la 8 biti.

Playlist

- 1) Bruno Mars - Uptown funk
- 2) Luis Fonsi - Despacito

- 3) Don Omar Ft. Lucenzo - Danza Kuduro
- 4) Maluma - Corazon
- 5) Modern Talking - Brother Louie '98
- 6) Mr. President - Coco Jamboo
- 7) Eminem - Godzilla ft. Juice WRLD
- 8) FISHER - Losing It
- 9) Disturbed - Down With The Sickness
- 10) Slipknot - Psychosocial

Hardware Design

Componente

Nume	Numar Piese
Arduino UNO	1
Breadboard	1
LCD with I2C board	1
Push Button	3
SD Card	1
SD Card Module	1
Speaker	1

Schema Bloc



Schema Electrica



Software Design

Descriere mod de functionare

In metoda de setup() se initializeaza pinii folositi, se citeste numarul de melodii care se afla in playlist de pe un fisier stocat pe cardul SD, se afiseaza un mesaj de bun venit, dupa care se porneste prima melodie.

In loop, se verifica daca unul dintre pinii PAUSE_PIN, NEXT_PIN, respectiv PREV_PIN si-a modificat starea. In caz afirmativ, se pune melodia pe pauza, se trece la urmatoarea melodie sau se la melodia anterioara, in functie de pin. In plus, daca nicio melodie nu mai este redata, se trece la melodia urmatoare.

Atunci cand melodia este schimbată, prin incrementarea/decrementarea unui contor care retine numarul melodiei curente se calculeaza numele fisierului ".wav" dupa urmatoarea formula: "music" + <song_idx> + ".wav", unde song_idx este numarul melodiei curente. Numele artistului si numele melodiei sunt citite dintr-un fisier de pe cardul SD, de forma "name" + <song_idx> + ".txt". Ulterior, informatiile despre melodie sunt afisate pe ecranul LCD si melodia este redată.

Alte surse folosite

Intrucat memoria dintr-un program Arduino este limitata, am optat pentru stocarea numelui artistului si a numelui melodiei intr-un fisier pe cardul SD, pentru fiecare melodie in parte. Astfel, chiar daca avem un numar foarte mare de melodii, tot vom putea afisa informatiile cantecului curent pe LCD.

Pentru a realiza fisierele care contin informatii despre fiecare melodie am folosit urmatoarea sursa de C++:

```
ifstream f("songs.in");
ofstream g;
int idx = 0;
string line;
while (getline(f, line)){
    idx++;
    std::size_t found = line.find(" - ");
    string artist = line.substr(0, found - 1);
    if(artist.size() > MAX_LEN)
        artist.resize(MAX_LEN);
    while(artist.size() < MAX_LEN)
        artist += " ";
    string song = line.substr(found + 2);
    if(song.size() > MAX_LEN)
        song.resize(MAX_LEN);
    while(song.size() < MAX_LEN)
        song += " ";
    g.open("name" + to_string(idx) + ".txt", std::ofstream::out);
    g<<artist<<'`n`<<song;
    g.close();
}
g.open("number.txt", std::ofstream::out);
g<<idx;
g.close();
```

Acest cod de C++ citeste informatiile despre toate melodiile din playlist dintr-un fisier si determina numarul de melodii, precum si fisierele ce contin informatiile specifice fiecarei melodii. Important de mentionat este ca LCD-ul perminte afisearea a maximum 16 caractere pe linie. Din acest motiv, daca numele artistului sau daca numele melodiei este mai lung de 16 caractere, doar primele 16 caractere se vor lua in considerare. Pentru a face citirea din fisier cat mai usoara, numele artistului este afisat pe prima linie din fisier, la care se adauga un padding de spatii pana se ajunge la limita de 16 caractere. Apoi, se trece pe urmatorul rand si se procedeaza identic si pentru numele melodiei.

Fisierul de intrare folosit pentru sursa de C++ este urmatorul:

Bruno Mars - Uptown funck
Luis Fonsi - Despacito
Don Omar - Danza Kuduro
Maluma - Corazon
Modern Talking - Brother Louie
Mr. President - Coco Jamboo
Eminem - Godzilla
FISHER - Losing It
Disturbed - Down With The Sickness
Slipknot - Psychosocial

Astfel, adaugarea informatiilor despre o melodie noua este foarte rapida. Trebuie adaugate detaliiile melodiei in fisierul care contine toate melodiile, in forma <nume_artist> - <nume_melodie> si rulat executabilul.

Biblioteci folosite

- LiquidCrystal_I2C.h
- SPI.h
- SD.h
- TMRpcm.h

Rezultate Obtinute

Link la video:

https://drive.google.com/file/d/1C5zzNqLmtb_GaGvuVCp3kN1-kn8zb0Y3/view?usp=sharing

Concluzii

Proiectul a fost foarte distractiv de implementat, intrucat m-am putut bucura de melodiile mele preferate. Consider ca realizarea acestui m-a ajutat sa fixsez diverse notiuni pe care le-am dobandit in cadrul laboratorului de PM.

In plus, am fost fortata sa gandesc outside de box in momentul in care mi-am dat seama ca stocarea in memorie a informatiilor melodiei este posibila doar pentru un numar limitat de melodii. Citirea informatiilor de pe fisiere dintr-un card SD este mult mai eficienta din punct de vedere al memoriei. Am ales sa fac fisiere separate pentru fiecare melodie in parte intr-un mod automat, folosind o sursa de C++, deoarece functia `read()` oferita de biblioteca `<SD.h>`, pe care o folosesc in codul de Arduino vine cu anumite limitari: poate citi un singur caracter o data / un numar predefinit de caractere.

Din punct de vedere electronic, circuitul functioneaza in modul in care mi-am dorit. Cu toate acestea, modificarea playlistului este destul de neplacuta, intrucat numele fisierelor ".wav" trebuie modificate in cazul in care o melodie este adaugata/stearsa. Sursa de C++ pe care am implementat-o faciliteaza adaugarea unei melodii noi la finalul playlist-ului.

In continuare, imi doresc sa mai implementez o sursa de C++ care sa faca adaugarea si stergerea de melodii mult mai rapida pentru un user.

Download

[331ca_matei_hermina_pm_project.zip](#)

Jurnal

20.04.2022 - Alegere tema proiect

21.04.2022 - Publicare descriere proiect

24.05.2022 - Editare descriere proiect

25.05.2022 - Adugare Hardware design

26.05.2022 - Adaugare Software desgin + Adaugare video + Adaugare arhiva

Bibliografie/Resurse

1. <https://circuitdigest.com/microcontroller-projects/arduino-audio-music-player>
2. <https://github.com/TMRh20/TMRpcm/wiki>
3. [https://create.arduino.cc/projecthub/muhammad-aqib/arduino-button-tutorial-using-arduino-digitalread-function-08adb5](https://create.arduino.cc/projecthub/muhammad-aqib/arduino-button-tutorial-using-arduino-digital-read-function-08adb5)
4. https://create.arduino.cc/projecthub/Arnov_Sharma_makes/lcd-i2c-tutorial-664e5a
5. <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2022>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/avaduva/musicplayer>



Last update: **2022/05/26 21:27**