

Music Player

Autor: [Hermina-Maria Matei](#)

Introducere

Descriere

Proiectul reprezintă o implementare a unui Music Player care cântă melodiile stocate pe un card SD. Utilizatorul are posibilitatea de a observa numele melodiei pe ecranul LCD, de a trece la melodia următoare/precedentă și a pune pe pauză melodia curentă prin intermediul butoanelor.

Utilitate

Ca utilitate, acest Music Player poate fi folosit de orice ascultător de muzică pentru a se relaxa în timpul liber. Faptul că muzica este stocată pe un card SD oferă un mod rapid de modificare a playlistului și satisface dorința utilizatorului de a stoca un număr mare de melodii.

Descriere generală

Music Player-ul porneste automat când este conectat la sursă de curent și începe prin a cânta prima melodie, în cazul în care cardul SD este introdus în SD Card Reader Module. În cazul în care cardul SD nu este introdus, pe Serial Monitor se afișează "SD failed".

Utilizatorul poate pune pe pauză melodia curentă și o poate relua din punctul în care aceasta a fost oprită prin intermediul butonului de pauză. Acesta poate trece la melodia următoare folosind butonul next și se poate întoarce la melodia anterioară folosind butonul prev.

Important de menționat este faptul că melodiile care se află pe cardul SD trebuie să fie în format ".wav", având un Sampling Rate de 16000Hz, canalul audio de tip Mono și Bit Resolution setat la 8 biți.

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 redada, se trece la melodia urmatoare.

Atunci cand melodia este schimbata, 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 carul SD, de forma "name" + <song_idx> + ".txt". Ulterior, informatiile despre melodie sunt afisate pe ecranul LCD si melodia este redada.

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++ citește informațiile despre toate melodiile din playlist dintr-un fișier și determină numărul de melodii, precum și fișierele ce conțin informațiile specifice fiecărei melodii. Important de menționat este că LCD-ul permite afișarea a maximum 16 caractere pe linie. Din acest motiv, dacă numele artistului sau dacă numele melodiei este mai lung de 16 caractere, doar primele 16 caractere se vor lua în considerare. Pentru a face citirea din fișier cât mai ușoară, numele artistului este afișat pe prima linie din fișier, la care se adaugă un padding de spații până se ajunge la limita de 16 caractere. Apoi, se trece pe următorul rând și se procedează identic și pentru numele melodiei.

Fișierul de intrare folosit pentru sursa de C++ este următorul:

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

Astfel, adăugarea informațiilor despre o melodie nouă este foarte rapidă. Trebuie adăugate detaliile melodiei în fișierul care conține toate melodiile, în forma <nume_artist> - <nume_melodie> și rulat executabilul.

Biblioteci folosite

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

Rezultate Obținute

Link la video:

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

Concluzii

Proiectul a fost foarte distractiv de implementat, întrucât m-am putut bucura de melodiile mele preferate. Consider că realizarea acestui m-a ajutat să fixez diverse noțiuni pe care le-am dobândit în cadrul laboratorului de PM.

În plus, am fost forțată să găsesc o soluție de stocare în momentul în care mi-am dat seama că stocarea în memorie a informațiilor melodiei este posibilă doar pentru un număr limitat de melodii. Citirea informațiilor de pe fișiere dintr-un card SD este mult mai eficientă din punct de vedere al memoriei. Am ales să fac fișiere separate pentru fiecare melodie în parte într-un mod automat, folosind o sursă de C++, deoarece funcția `read()` oferită de biblioteca `<SD.h>`, pe care o folosesc în codul de Arduino vine cu anumite limitări: poate citi un singur caracter o dată / un număr predefinit de caractere.

Din punct de vedere electronic, circuitul funcționează în modul în care mi-am dorit. Cu toate acestea, modificarea playlistului este destul de neplăcută, întrucât numele fișierelor ".wav" trebuie modificate în cazul în care o melodie este adăugată/ștearsă. Sursa de C++ pe care am implementat-o facilitează adăugarea unei melodii noi la finalul playlist-ului.

În continuare, îmi doresc să mai implementez o sursă de C++ care să facă adăugarea și ștergerea de melodii mult mai rapidă 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 - Adăugare Hardware design

26.05.2022 - Adăugare Software design + Adăugare video + Adăugare arhivă

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>
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:

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

Permanent link:

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



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