

# Mini Player Audio cu Volum Reglabil și Feedback Vizual

## Introducere

Proiectul meu este un **Mini Player Audio** care redă fișiere `.wav` cu melodii de pe un card microSD, permite reglarea volumului printr-un potențiomtru și oferă feedback vizual al nivelului sonor printr-un VU-meter format din LED-uri.

Scopul proiectului este de a integra conceptele învățate în laboratoare (PWM, ADC, SPI) într-o aplicație practică, ușor de înțeles și relevantă în viața reală.

Fiind pasionat de partea de muzica / producție muzicala etc. ideea a pornit de la dorința de a învăța cum funcționează redarea audio la nivel de microcontroler și cum putem controla semnalul atât din punct de vedere al intensității, cât și al afișării vizuale.

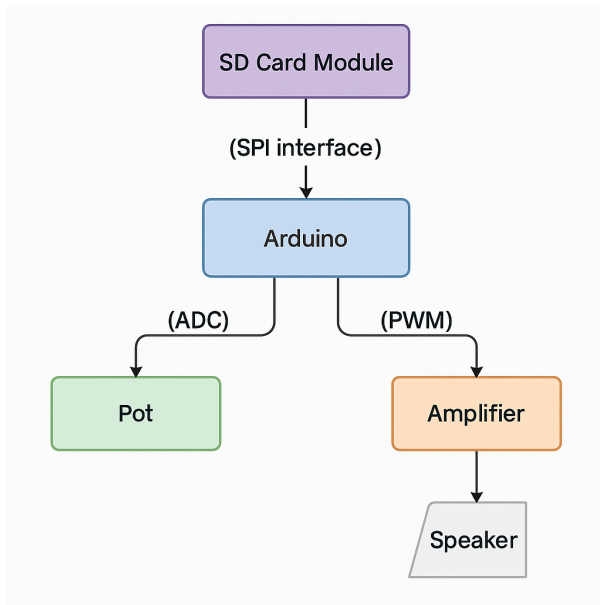
Considerăm că este un proiect util atât pentru învățare, cât și ca o improvizație de boxă portabilă — o alternativă low-cost și ușor de replicat.

## Descriere generală

Schema bloc ilustrează principalele componente hardware și fluxul de date din proiect. Modulul SD comunică cu Arduino prin interfața SPI, trimițând fișierul audio `.wav` către microcontroler. Acesta procesează datele și generează un semnal PWM proporțional, transmis către amplificatorul audio care alimentează difuzorul.

Potențiometrul conectat la un pin analogic este citit de Arduino prin conversie ADC, iar valoarea sa este folosită pentru a controla volumul audio și pentru a aprinde un număr corespunzător de LED-uri, simulând vizual un VU-meter.

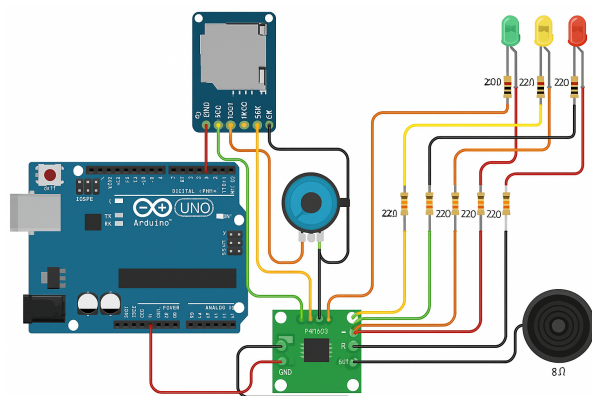
La nivel software, sunt utilizate librăriile `SD.h` și `TMRpcm.h` pentru gestionarea fișierelor audio, iar interacțiunile cu componentele externe sunt realizate prin funcțiile `analogRead()`, `analogWrite()` și `digitalWrite()`.

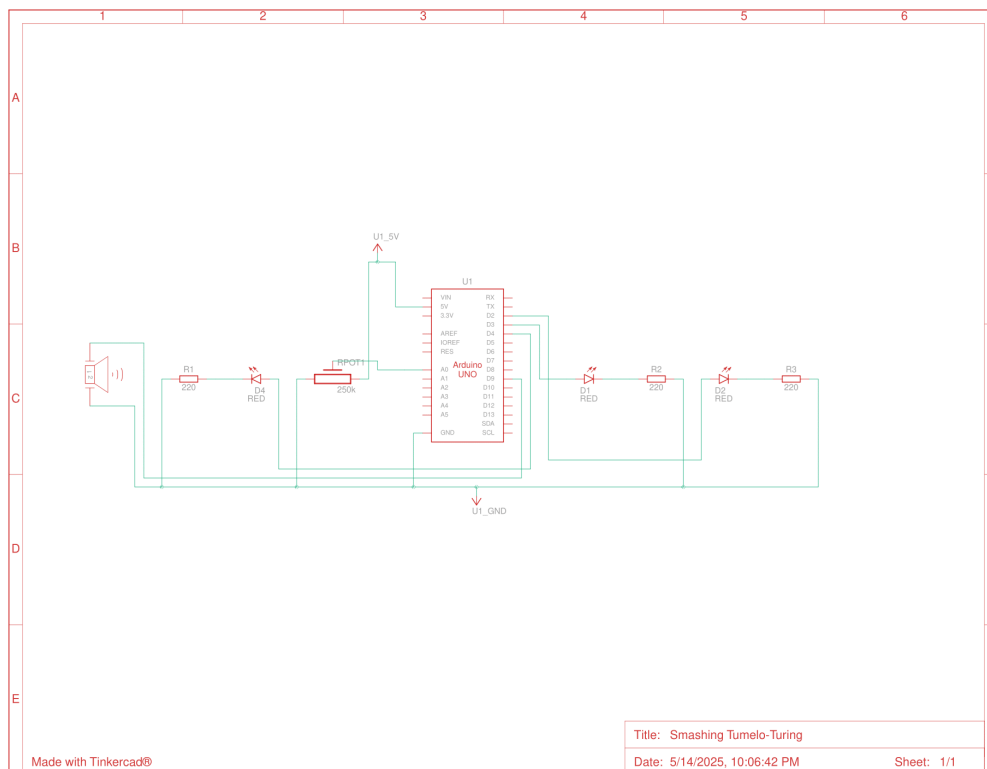


## Hardware Design

Listă componente hardware utilizate:

- **\*\*Arduino Uno R3\*\***
- **\*\*Modul cititor microSD (interfață SPI)\*\***
- **\*\*Card microSD (format FAT32) cu fișier .wav\*\***
- **\*\*Amplificator audio PAM8403\*\***
- **\*\*Difuzor pasiv 8Ω / 0.5W\*\***
- **\*\*Potențiometru 10kΩ (linear)\*\***
- **\*\*4× LED-uri (roșu/galben/verde)\*\***
- **\*\*4× rezistențe 220Ω pentru LED-uri\*\***
- **\*\*Breadboard și fire de conexiune\*\***





## Software Design

### ► Mediu de dezvoltare:

Proiectul a fost dezvoltat în Arduino IDE, folosind platforma Arduino UNO. Testarea și încărcarea s-au realizat prin conexiune USB, iar pentru debugging s-a folosit Serial Monitor.

### ► Librării și surse 3rd-party utilizate:

- TMRpcm.h – pentru redarea fișierelor .wav de pe cardul SD folosind PWM audio.
- SD.h – pentru comunicarea cu cardul microSD prin interfața SPI.
- SPI.h – necesară pentru comunicația SPI între Arduino și modulul SD.

### ► Algoritmi și structuri implementate:

- Inițializare SD și detecția fișierelor .wav numerotate secvențial (short2.wav – short6.wav).
- Redare audio secvențială: aplicația redă automat fiecare fișier pe rând, trecând la următorul doar după ce s-a terminat cel precedent.
- Control volum cu potențiometrul: semnalul analogic de la potențiometrul conectat la A0 este citit periodic și mapat pe intervalul 0–4 pentru a seta volumul audio.

- Afișare vizuală a volumului prin LED-uri: 3 LED-uri conectate la D6, D7, D8 se aprind progresiv în funcție de valoarea citită de pe potențiomtru.
- Comandă de tip skip prin serial: apăsarea tastei s în Serial Monitor oprește melodia curentă și trece la următoarea.

► Surse și funcții implementate:

Fișierul principal Arduino (.ino) conține următoarele funcții și secțiuni:

- `setup()`:
- Inițializarea pinii pentru LED-uri.
- Inițializarea cardului SD și verificarea fișierelor.
- Setarea inițială a volumului și pornirea primei melodii.
- `loop()`:
- Verifică dacă s-a apăsat s pentru a sări la următoarea melodie.
- La finalul unei melodii, încarcă automat următorul fișier .wav.
- La fiecare 100 ms:
- Actualizează volumul audio în funcție de poziția potențiometrului.
- Aprinde LED-urile corespunzător pragurilor (ex: peste 200, 400, 600).

## Rezultate Obținute


Proiectul a fost finalizat cu succes, atingând toate obiectivele propuse:

- Redare audio funcțională direct de pe cardul SD, folosind fișiere .wav în format compatibil (8-bit, mono, 16 kHz), prin modulul TMRpcm.
- Control dinamic al volumului printr-un potențiomtru conectat la pinul A0, cu reglare în timp real a intensității sunetului în intervalul 0–4.
- Indicație vizuală a volumului prin aprinderea progresivă a 3 LED-uri în funcție de poziția potențiometrului.
- Redare secvențială a fișierelor audio (short2.wav până la short6.wav), automatizat fără intervenție manuală.
- Funcție de skip implementată prin Serial Monitor (comanda s), pentru a trece la următorul track și funcție de back (comanda b).

## 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/apredescu/rares.carbunaru> 

Last update: **2025/05/28 08:32**