

Player muzical interactiv cu efecte luminoase

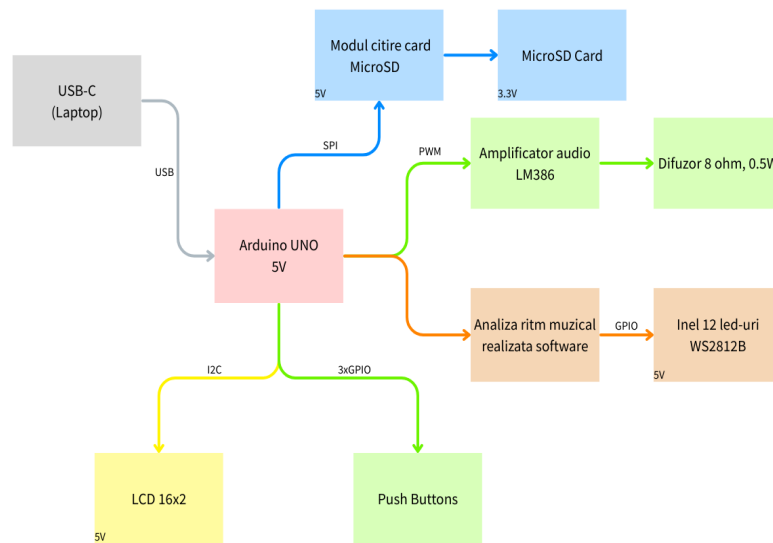
Introducere

Prezentarea pe scurt a proiectului:

- Sistemul propus este un player muzical controlat cu ajutorul unui ecran LCD si butoane fizice, capabil sa redea melodii in format WAV stocate pe un card microSD, utilizand o placa Arduino si un difuzor. Utilizatorul poate naviga intre melodii si moduri de iluminare prin intermediul a trei butoane: Start/Stop, Selectare melodie, Selectare mod iluminare. Melodiile sunt redade folosind biblioteca TMRpcm, iar sistemul utilizeaza pinul analogic A0 pentru a citi semnalul audio preluat direct din difuzor, printr-o punte realizata cu doua rezistente. Pe baza valorilor citite in timp real, este calculat un nivel audio, care este folosit pentru a actualiza banda de LED-uri WS2812B.
- Scopul principal este de a oferi o experienta multimedia interactiva si personalizabila, in care utilizatorul nu doar asculta muzica, ci si vizualizeaza efecte luminoase adaptate atat ritmului, cat si preferintelor sale de moment.
- Ideea de baza a pornit de la dorinta de a crea un sistem care sa combine muzica si luminile intr-un mod sincronizat, inspirat de sistemele moderne de redare muzicala cu efecte vizuale. Pentru a creste interactivitatea si personalizarea, am adaugat posibilitatea de a selecta diferite moduri de iluminare, astfel incat utilizatorul sa poata adapta atmosfera in functie de preferinte sau context.
- Utilitatea proiectului este data de experienta de divertisment personalizabila pe care o ofera, potrivita pentru petreceri, decoruri tematice sau relaxare. De asemenea, proiectul poate fi utilizat si in scopuri didactice, urmarind concepte de baza de electronica, Arduino si programare embedded si avand o posibilitate vasta de extindere. Pentru mine, proiectul este util atat din punct de vedere tehnic, intrucat am ocazia de a aplica o varietate de cunostinte pe care le-am dobandit de-a lungul facultatii, dar si din punct de vedere personal, pentru ca reprezinta o provocare creativa si tehnica.

Descriere generală

Proiectul este un player muzical interactiv care reda fisiere audio WAV de pe un card microSD si sincronizeaza efecte luminoase cu ritmul muzicii. Utilizatorul controleaza redarea si modul de iluminare prin butoane, iar informatiile sunt afisate pe un ecran LCD.



Modul de functionare:

Sistemul este organizat in jurul unui Arduino UNO, care coordoneaza toate modulele hardware si software. Semnalul audio este citit de pe un card microSD, procesat si trimis sub forma de semnal PWM catre un amplificator audio, care reda sunetul printr-un difuzor. In paralel, Arduino analizeaza semnalul pentru a extrage ritmul si controleaza un inel de LED-uri WS2812B, generand efecte luminoase sincronizate. Utilizatorul interactioneaza cu sistemul prin butoane pentru a selecta melodia si modul de iluminare, iar un ecran LCD afiseaza informatii relevante despre starea redarii. Toate modulele comunica direct cu Arduino, care gestioneaza logica si sincronizarea in timp real.

Interactiunea intre module:

- Utilizatorul interactioneaza cu sistemul prin butoane pentru:
 - Selectare melodie
 - Alegere mod de iluminare
 - Pornire/Oprire redare
- Arduino UNO:
 - Citeste fisiere WAV de pe cardul microSD prin SPI
 - Trimite semnal PWM audio catre amplificatorul audio → difuzor reda sunetul
 - Analizeaza semnalul audio in timp real pentru a detecta ritmul muzicii
 - Controleaza LED-urile WS2812B in functie de beat-uri si modul de iluminare
 - Actualizeaza LCD-ul cu informatii despre melodia curenta si starea sistemului
- Amplificatorul (LM386) primeste semnal PWM si il reda ca sunet prin difuzor

- LED-urile WS2812B afiseaza efecte sincronizate cu ritmul muzicii si stilul ales
- LCD 16×2 afiseaza informatii utile pentru utilizator (melodie, mod, status)

Hardware Design

Lista de componente:

Componenta	Descriere	Link
Placa de dezvoltare Arduino UNO	Microcontroler ATmega328P, tensiune de operare 5V, alimentare recomandata 7-12V DC, curent maxim per pin I/O: 40 mA	link
Inel cu 12 LED-uri RGB5050 WS2812	LED-uri RGB adresabile individual, alimentare 5V DC, consum curent per LED: ~60 mA	link
Modul amplificator audio LM386, 4-12V	Amplificator audio mono, tensiune de alimentare 4-12V DC, putere de iesire pana la 1W, impedanta minima difuzor: 4Ω	link
Modul citire card microSD, compatibil Arduino	Interfata SPI, alimentare 5V DC, regulator de tensiune integrat 3.3V, curent DC pini: 0.2-200 mA	link
Card de memorie Micro-SDHC, 16GB, Class 10	Capacitate 16GB, viteza de scriere minima 10 MB/s, tensiune de operare 2.7-3.6V	link
Boxa ultra subtire 40mm 8 ohms 0.5W	Diametru 40mm, impedanta 8Ω, putere nominala 0.5W, ideala pentru iesiri audio de mica putere	link
Breadboard 830 puncte MB-102	830 puncte de conectare, compatibila cu fire de 0.8mm, permite alimentare 3.3V/5V prin modul dedicat	link
3 x Butoane Mini 6x6x5, 4 pini	Dimensiuni 6x6x5 mm, curent maxim recomandat: 50 mA	link
LCD 16×2 cu interfata I2C	Afisaj 16×2 caractere, alimentare 5V DC, interfata I2C, consum curent: ~2 mA	link
Rezistente 330 ohm	Valoare 330Ω, utilizate pentru limitarea curentului prin LED-uri, putere nominala 0.25W	
Rezistente 10K ohm	Valoare 10kΩ, utilizate pentru divizorul de tensiune	
Fire Tata-Tata	Fire de conectare cu pini tata la ambele capete, pentru conexiuni pe breadboard	
Fire Tata-Mama	Fire de conectare cu un pin tata si un pin mama, pentru interconectarea modulelor	

Schema electrica:

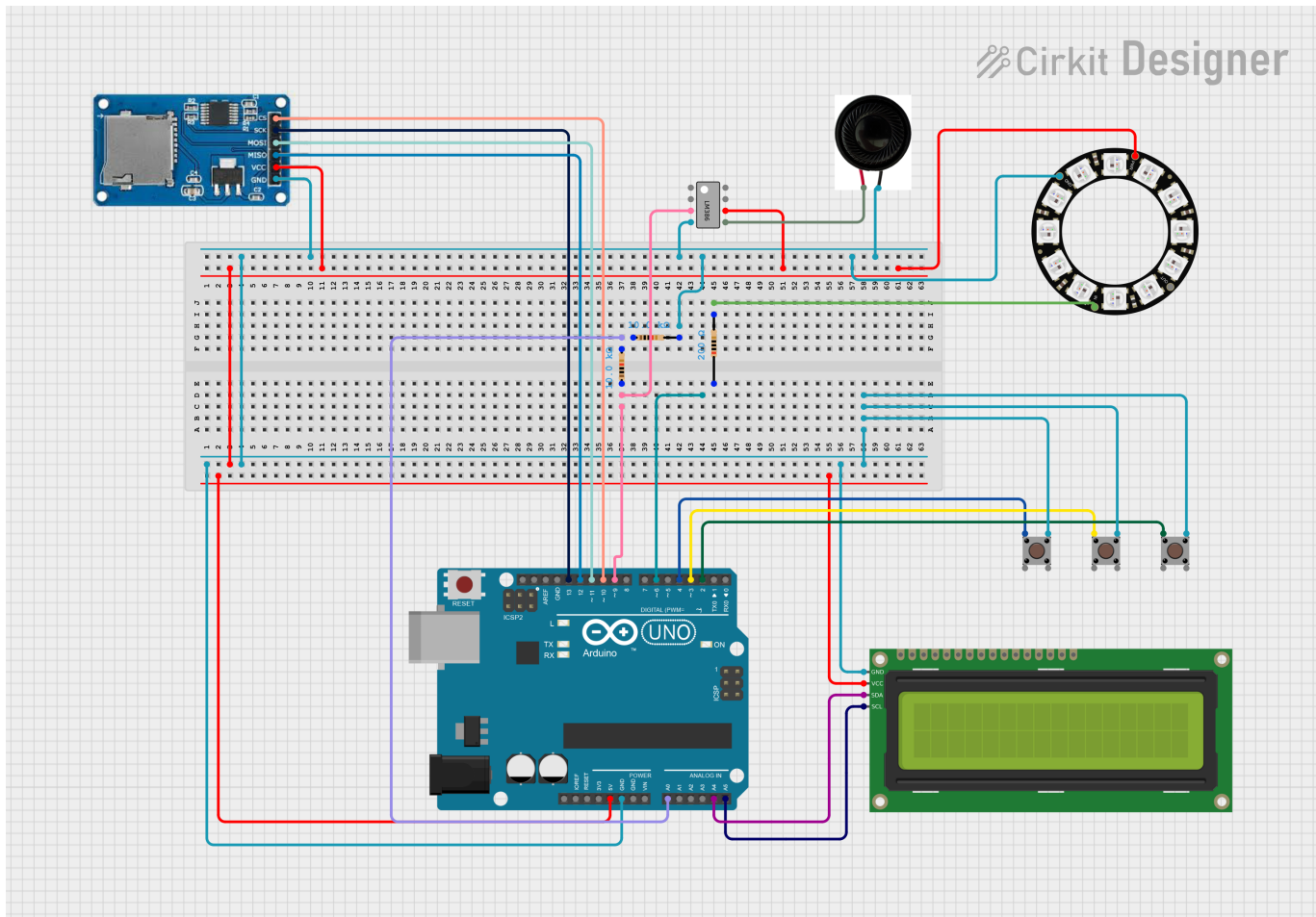


Diagrama realizata in [Cirkuit Designer IDE](#)



Schema electrica realizata in [EasyEDA](#)

Pini utilizati:

Componenta	Pin componenta	Pin Arduino (Registru)	Descriere	Tehnologie utilizata
Modul SD Card	MISO	D12 (PB4)	Citire date de pe card	SPI
	MOSI	D11 (PB3)	Transmitere date catre card	
	SCK	D13 (PB5)	Clock SPI	
	CS	D10 (PB2)	Selectare chip activ	
	+3.3V	-	Alimentare	
	GND	GND	Masa	
WS2812 Inel LED	DIN	D6 (PD6)	Semnal date catre LED-uri	PWM (protocol WS2812)
	+5V	-	Alimentare 5V	
	GND	GND	Masa	
Rezistenta 330Ω	-	D6 (PD6)	Limitare curent pentru LED-uri	-

LCD I2C	SDA	A4 (PC4)	Linie date I2C	I2C
	SCL	A5 (PC5)	Linie ceas I2C	
	VCC	-	Alimentare 5V	
	GND	GND	Masa	
Amplificator Audio LM386	IN+	D9 (PB1)	Intrare audio de la Arduino spre amplificator	GPIO
	VCC	-	Alimentare 5V	
	VOUT	SP1 (difuzor)	iesire audio catre difuzor	
	GND	GND	Masa	
Divizor tensiune audio (2x10kΩ)	-	A0 (PC0)	Citire semnal audio analogic (detectare ritm)	ADC
	R1 capat 1	D9 (PB1)	Legat in serie cu IN de la LM386	
	R1-R2 comun	A0 (PC0)	Punctul de prelevare semnal	
	R2 capat 2	GND	Referinta masa	
Difuzor	+	VOUT (U1)	Redare sunet	-
	-	GND	Masa	
Buton 1	-	D2 (PD2)	Da play la melodie	GPIO
Buton 2	-	D3 (PD3)	Selecteaza melodia	GPIO
Buton 3	-	D4 (PD4)	Selecteaza modul de iluminare	GPIO

Sincronizarea LED-urilor cu semnalul audio:

Pentru a putea analiza semnalul audio generat in timpul redarii unei melodii si a aprinde LED-urile in functie de intensitatea acestuia, am realizat un divizor de tensiune format din doua rezistente de 10kΩ conectate in serie. Un capat al primului rezistor este conectat la pinul D9 al placii Arduino, pin utilizat pentru transmiterea semnalului audio catre intrarea amplificatorului LM386. Capatul celalalt al celui de-al doilea rezistor este conectat la GND, iar punctul comun dintre cei doi rezistori este conectat la pinul analogic A0. Aceasta configuratie permite preluarea unei versiuni atenuate a semnalului audio (prin divizorul de tensiune), astfel incat Arduino-ul sa poata citi nivelul semnalului cu ajutorul convertorului analog-digital (ADC) fara a-l suprasolicita electric. In acest mod, se poate monitoriza variatia semnalului in timp real si controla efectele luminoase generate de banda de LED-uri, sincronizandu-le cu ritmul muzicii.



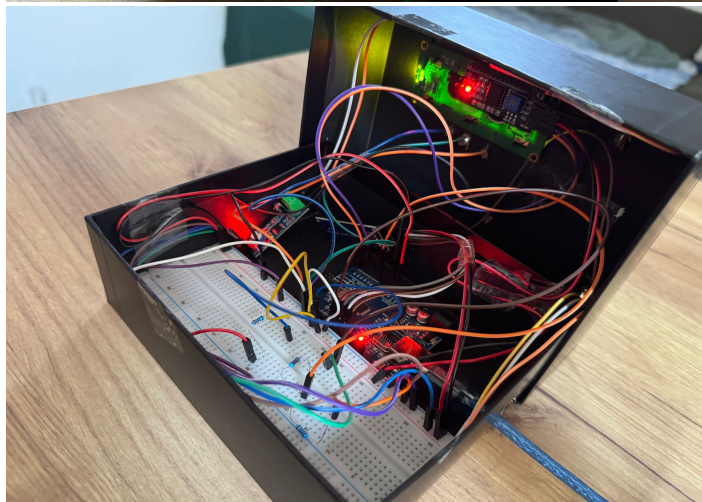
Calculare efectuate:

Banda de LED-uri WS2812 este compusa din 12 LED-uri adresabile individual, fiecare LED avand trei componente de culoare (rosu, verde, albastru) care pot consuma pana la aproximativ 20 mA fiecare. Pentru a aprinde un LED in culoarea alba, toate cele trei componente se activeaza simultan la intensitate maxima, deci un singur LED poate consuma pana la circa 60 mA ($20 \text{ mA} \times 3$). Astfel, pentru 12 LED-uri aprinse simultan in alb, curentul total necesar este aproximativ $60 \text{ mA} \times 12 = 720 \text{ mA}$. Placa Arduino UNO, alimentata prin USB sau prin pinul de 5V, poate furniza un curent maxim mult

mai mic, in jur de 500 mA in total pentru toata placa, iar porturile sale I/O nu sunt proiectate sa suporte curenti atat de mari. Prin urmare, alimentarea directa a tuturor celor 12 LED-uri WS2812 in culoarea alba simultan fara o sursa externa de curent dedicata nu este posibila, deoarece risca sa suprasolicite regulatorul de tensiune al placii si sa provoace resetari sau chiar deteriorarea componentelor. Din acest motiv, proiectul pe care l-am dezvoltat nu prevede aprinderea tuturor led-urilor simultan in alb, avand moduri de iluminare prestabilite.

Stadiul implementarii hardware:

Proiectul este implementat complet si contine toate modulele si componentele pe care le-am enumerat mai sus. Am utilizat o cutie, pe a carei parte frontala am lipit difuzorul, banda de leduri care il inconjoara, un mini breadboard pe care sunt conectate cele 3 butoane si ecranul LCD pentru care am decupat o portiune din cutie, intrucat nu putea fi lipit din cauza modulului I2C atasat pe spatele lui.



Imaginile de mai sus ilustreaza proiectul final in primul stadiu al functionarii: "Sistem Pornit". Pentru a aprinde banda de leduri si a reda muzica, trebuie apasat butonul 1: "Start/Stop". Am considerat adecvat sa lipesc 3 post-it-uri, numerotate de la 1 la 3, care sugereaza functionalitatea fiecarui buton de la stanga la dreapta.

Software Design

Mediu de dezvoltare:

Sistemul software este dezvoltat in ArduinoIDE, folosind limbajul C++ specific platformei Arduino, cu acces direct la registrele microcontroller-ului ATmega328P.

Biblioteci utilizate

- **LiquidCrystal_I2C** - Permite controlul afisajului LCD 16x2 prin protocolul I2C, afisand informatii precum titlul melodiei, modul de iluminare sau starea sistemului. Economiseste pini GPIO prin utilizarea magistralei I2C.
- **FastLED** - Oferă control precis asupra benzii de LED-uri WS2812B. Permite generarea de efecte luminoase dinamice, sincronizate cu intensitatea semnalului audio, in functie de modul selectat (Energic, Unda, Disco).
- **TMRpcm** - Permite redarea fisierelor audio in format WAV direct de pe cardul microSD prin semnal PWM generat pe pinul 9 (OC1A). Nu necesita DAC extern si este compatibila cu ATmega328P.
- **SD.h** - Gestioneaza accesul la fisierele stocate pe cardul microSD folosind protocolul SPI. Asigura initializarea cardului si permite redarea fisierelor audio prin TMRpcm.
- **SPI.h** - Biblioteca nativa pentru comunicatia SPI. Este folosita de SD.h pentru a facilita schimbul de date intre Arduino si cardul SD (pinii MOSI, MISO, SCK, CS).
- **Wire.h** - Suporta comunicatia I2C cu dispozitive precum afisajul LCD 16x2. Este necesara pentru utilizarea bibliotecii LiquidCrystal_I2C.

Formatarea cardului SD si specificatii pentru fisierele audio

Cardul SD utilizat este formatat in sistemul de fisiere FAT32, asigurand compatibilitatea si accesul rapid la fisierele audio prin biblioteca SD a Arduino.

Fisierele audio au fost pregatite cu urmatoarele proprietati pentru a fi compatibile cu modulul de redare:

- Format: WAV (Waveform Audio File Format)
- Canal audio: mono
- Rata de esantionare: 16.000 Hz
- Dimensiune esantion audio: 8 biti
- Rata de biti: 128 kbps
- Dimensiune maxima fisier: 4 MB

Aceste setari au fost aplicate folosind programul Audacity, pentru a asigura o redare optima si stabila

pe platforma Arduino UNO.

De asemenea, fisierele audio sunt denumite cu nume scurte, fara spatii si cu extensia ".wav" (exemplu: 001.wav), respectand cerintele sistemului de fisiere FAT32.

Elemente de noutate:

Sistemul propus aduce o abordare moderna si interactiva in redarea de continut audio prin sincronizarea sunetului cu efecte vizuale dinamice. Utilizatorul beneficiaza de o experienta personalizata prin controlul direct al melodiilor si al stilurilor de iluminare, intr-o interfata simpla si intuitiva. In plus, sistemul functioneaza autonom, fara nevoie de aplicatii externe sau dispozitive inteligente, fiind ideal pentru proiecte educationale, demonstratii vizuale sau ambient personalizat. Combinatia dintre redare audio si iluminare reactiva transforma interactiunea clasica cu sunetul intr-o experienta captivanta si vizuala.

Legatura cu laboratoarele:

Proiectul propus valorifica in mod direct conceptele studiate in mai multe laboratoare parcurse pe parcursul semestrului:

- **Laboratorul 0: GPIO**

Utilizat pentru citirea starii butoanelor care controleaza redarea melodiei, selectarea piesei si alegerea modului de iluminare.

- **Laboratorul 3: Timere. PWM**

Folosit pentru generarea semnalului PWM care transmite sunetul catre amplificatorul audio si pentru controlul LED-urilor WS2812B sincronizate cu ritmul muzicii.

- **Laboratorul 5: SPI**

Esential pentru comunicarea cu cardul microSD de pe care sunt citite fisierele WAV.

- **Laboratorul 6: I2C**

Utilizat pentru interfatarea cu ecranul LCD 16x2, care afiseaza informatii despre melodie, mod de iluminare si starea sistemului.

Functii implementate

Functii de initializare

- **initButtons()** - Configureaza pinii PD2, PD3 si PD4 ca intrari digitale, activand rezistentele de pull-up pentru a detecta apasarea celor trei butoane (Play, Next, Mod).

```
void initButtons() {
  DDRD &= ~(BUTTON_PLAY_MASK | BUTTON_NEXT_MASK | BUTTON_MODE_MASK);
  PORTD |= BUTTON_PLAY_MASK | BUTTON_NEXT_MASK | BUTTON_MODE_MASK;
}
```

- **lcd_init()** - Initializeaza afisajul LCD I2C (16x2), porneste iluminarea de fundal si curata ecranul.

```
void lcd_init() {
  lcd.init();
  lcd.backlight();
  lcd.clear();
}
```

- **lcd_set_cursor(uint8_t col, uint8_t row)** - Pozitioneaza cursorul la coloana si randul specificate pe ecranul LCD.
- **lcd_print(const char* str)** - Afiseaza un text simplu pe LCD, incepand de la pozitia curenta a cursorului.
- **setup()** - Initializeaza comunicari seriale, LCD-ul, butoanele, banda de LED-uri WS2812B, intrarea analogica pentru senzorul audio, magistrala SPI si cardul SD. Seteaza pinul audio si volumul pentru redare.

```
// Initializare SPI pentru cardul SD
DDRB |= (1 << PB3) | (1 << PB5); // MOSI (PB3), SCK (PB5) ca iesire
DDRB &= ~(1 << PB4); // MISO (PB4) ca intrare
SPCR = (1 << SPE) | (1 << MSTR) | (1 << SPR0); // Activeaza SPI, modul
master

if (!SD.begin(SD_CS)) {
  lcd.clear();
  lcd.print("Eroare SD card");
  while (true);
}
```

Functii audio

- **playAudio()** - Reda melodia curent selectata din array-ul `songs[]` folosind biblioteca TMRpcm. Actualizeaza ecranul LCD cu titlul melodiei si seteaza flag-ul `isPlaying` pe `true`.
- **stopAudio()** - Opreste redarea curenta a fisierului audio si reseteaza flag-ul `isPlaying` pe `false`.
- **selectNextSong()** - Opreste redarea curenta, selecteaza urmatoarea melodie din lista, iar apoi o porneste automat.

Funcții LED / efecte vizuale

- **updateLEDEffects()** - Actualizează efectele de iluminare în funcție de modul selectat (`currentMode`) și de nivelul semnalului audio analogic:
- Mod 0 (Energic): LED-urile sunt colorate în nuanțe vibrante (HSV) cu saturatie mare, generate aleator pentru fiecare ciclu de actualizare.

```
int audioLevel = analogRead(AUDIO_SENSE_PIN);
int intensity = map(audioLevel, 0, 1023, 0, 255);
intensity = constrain(intensity, 0, 200);

switch (currentMode) {
  case 0: {
    uint8_t hueBase = random(0, 255);
    for (int i = 0; i < NUM_LEDS; i++) {
      leds[i] = CHSV(hueBase + i * 10, 240, intensity);
    }
    break;
  }
}
FastLED.show();
```

- Mod 1 (Unda): Creează o undă de lumină sinusoidală animată pe LED-uri, cu culori gradate și strălucire dependentă de intensitatea sunetului.
- Mod 2 (Disco): Iluminează aleator LED-uri cu culori intense și variabile la intervale scurte, imitând un efect disco dinamic.

Funcții pentru butoane

- **buttonPressed(uint8_t mask)** - Verifică dacă un buton conectat la un pin digital este apăsător, returnând `true` dacă starea este LOW.

```
bool buttonPressed(uint8_t mask) {
  return !(PIND & mask); // LOW inseamna apasat
}
```

Funcții din loop()

- **loop()** - Verifică apăsările de butoane (Play, Next, Mod), execută acțiunile corespunzătoare (redare, schimbare melodie, schimbare mod iluminare) și, dacă o melodie este în redare, actualizează efectele LED-urilor în timp real.

Atat codul final, cat si codurile pentru verificarea functionarii modulelor se gasesc pe [GitHub](#)

Rezultate Obținute

Proiectul realizat permite redarea fișierelor audio WAV de pe un card microSD și controlul unei benzi de LED-uri WS2812B în funcție de semnalul audio și modul selectat.

[Videoclip demonstrativ](#)

Următoarele funcționalități sunt complet implementate și prezentate în videoclipul demonstrativ:

- Navigarea între modurile de redare prin apăsarea butoanelor fizice
- Afisarea informațiilor pe un ecran LCD 16x2: numele melodiei curente și modul activ
- Redarea fișierelor audio
- Iluminare dinamică a LED-urilor în funcție de semnalul audio (efecte sincronizate)
- Selectarea modurilor de iluminare

Aceste rezultate demonstrează funcționalitatea completă a sistemului audio-luminos proiectat și integrarea cu succes a componentelor hardware și software.

Concluzii

Ideea de la care a pornit acest proiect a fost una simplă: să creez un player muzical care să redea melodii și, în același timp, să ofere o experiență vizuală prin efecte de lumină sincronizate cu ritmul muzicii. La final, pot spune că obiectivul a fost atins. Sistemul funcționează corect, permite selectarea melodiei și a modului de iluminare, iar rezultatul este o experiență interactivă pentru utilizator. Proiectul poate fi îmbunătățit în viitor prin adăugarea unei telecomenzi cu infraroșu, integrarea unui senzor de sunet pentru o reacție mai dinamică a luminilor sau chiar conectivitate Bluetooth pentru selectarea melodiilor direct de pe telefon. Sunt multumită de rezultatul obținut și cred că are potențial să evolueze în direcții și mai creative.

Bibliografie/Resurse

Resurse Hardware

- **Arduino UNO R3** : [Datasheet oficial](#)
- **WS2812B LED Ring** : [Datasheet WS2812B](#)

- **Micro SD Card Module** : [Modul SD Card + Pinout & Conectare](#)
- **LCD 16x2 cu interfață I2C** : [LCD1602 + I2C Datasheet](#)

Resurse Software

- **Biblioteca SD** : [Arduino SD Library Reference](#)
- **Platforma Arduino IDE** : [Arduino IDE oficial](#)

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/ajipa/bianca_ana.iliescu



Last update: **2025/05/30 05:26**