

# Lampa Interactivă

- Autor: Neculau Sanda-Elena
- Grupa: 334 CB

## Introducere

Lampă interactivă destinată relaxării, care combină controlul tactil, jocuri de lumini RGB și sunete ambientale pentru a crea o atmosferă plăcută. Utilizatorul poate interacționa cu lampa printr-un senzor capacitiv de atingere, selectând între diverse moduri de iluminare și sunet. Informațiile despre modurile active și a volumului sunt afișate printr-un display. Lampa are și moduri care reacționează la lumina ambientală schimbându-și intensitatea de iluminare și aprinzându-se atunci când e întuneric în cameră.

## Descriere generală



## Hardware Design

Placuta este MEGA 2560, eroare de scriere



## Bill of materials

Nume componenta	Cantitate	Link
Arduino mega2560	1	<a href="#">KIT ARDUINO</a>
Senzor touch TTP223B	1	<a href="#">senzor</a>
led-uri RGB	2	<a href="#">led-uri RGB</a>
LCD+I2C	1	<a href="#">LCD</a>
DFPlayer TF-16P	1	<a href="#">DFPlayer</a>
difuzor 8 ohmi, 0,2W	1	-

rezistoare 220	8	kit arduino
rezistor 10	1	kit arduino
rezistor 1	1	kit arduino
fotorezistor	1	kit arduino
card-microSD	1	card
baterie 9V	1	-
conector baterie	1	kit arduino

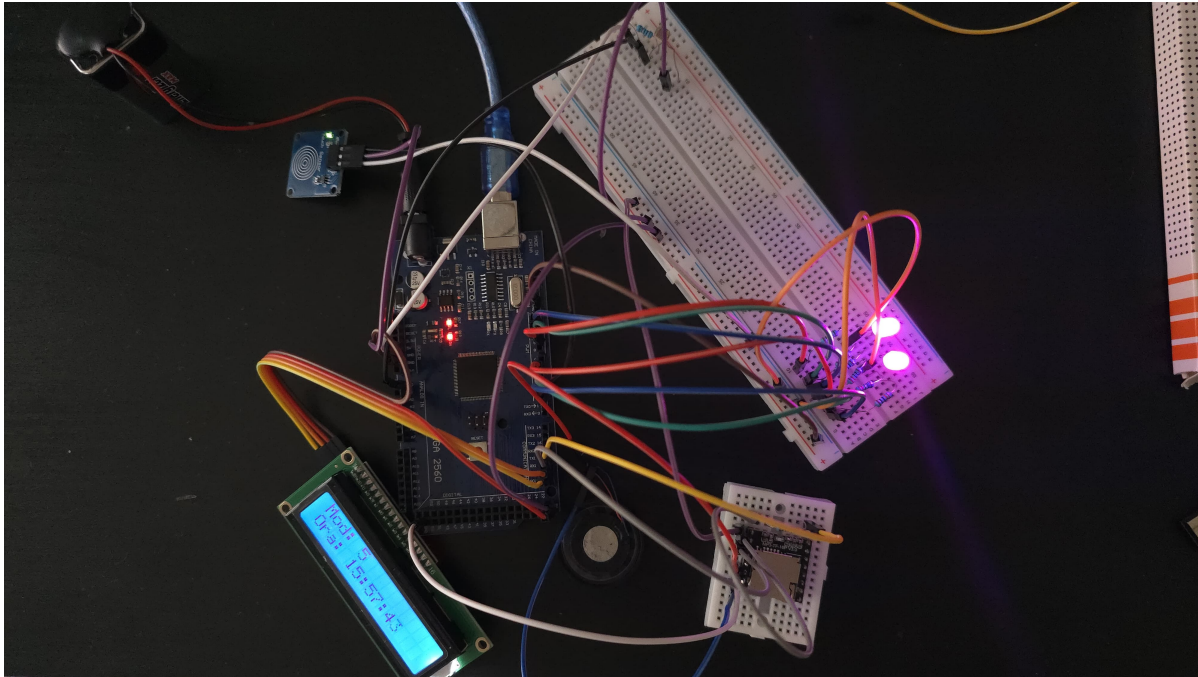
## Pini folositi

Componentă	Pin Arduino Mega	Motiv
<b>Senzor Touch TTP223B</b>	pin 2	Este un pin care permite întrerupere externă (`attachInterrupt`)
<b>LED RGB 1</b>	pin 3 (PWM) - Albastru, pin 5 (PWM) - Verde, pin 6 (PWM) - Roșu	PWM pentru controlul intensității culorilor LED-ului 1
<b>LED RGB 2</b>	pin 9 (PWM) - Roșu, pin 10 (PWM) - Verde, pin 11 (PWM) - Albastru	PWM pentru LED 2. Pe Mega, aceste pini suporta analogWrite()
<b>LCD 1602 I2C</b>	SDA: pin 20, SCL: pin 21	Pe Arduino Mega, I2C-ul este pe pinii 20 (SDA) si 21 (SCL)
<b>DFPlayer Mini</b>	RX: pin 18, TX: pin 19	UART hardware serial (Serial2).
<b>Difuzor</b>	Conectat la <b>SPK_1</b> și <b>SPK_2</b> pe DFPlayer	Nu se leagă direct la Arduino, este controlat de DFPlayer
<b>Fotorezistor + rezistor de 10k</b>	A0 (pin analogic)	Citire lumina ambientală din camera prin analogRead()

## Biblioteci

- LiquidCrystal\_I2C.h
- DFRobotDFPlayerMini.h
- stream.h

## Componente conectate



## Software Design

### Mediu de dezvoltare

- Arduino IDE

### Librări

- LiquidCrystal\_I2C.h : comunicarea cu lcd-ul
- DFRobotDFPlayerMini.h : comunicarea cu dfplayerul
- stream.h : m-am folosit de ea pentru serialul de pe dfplayer, intruca dfplayerului trebuie sa fie un obiect de tip stream ca sa putem utiliza read(), available(), peek()

### Structura software

- **Serial 0 manual**
  - Initializare serial0

```
void manualSerialInit(long baud) {  
  // Setare registri pentru UART0 (Serial)  
  // calculare viteza de comunicare, F_CPU(viteza procesorului)  
  uint16_t ubrr = F_CPU/16/ baud-1;  
  // impartim valoare pe 2 registri primi 8 biti si restul
```

```
UBRR0H = (unsigned char)(ubrr>>8);
UBRR0L = (unsigned char)ubrr;
// Activeaza receptorul (RXEN0) si transmisorul (TXEN0) pentru UART0
UCSR0B = (1<<RXEN0)|(1<<TXEN0);
// seteaza marimea datelor transmise la 8 biti
UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);
}
```

- Scriere date pe UART0

```
void manualSerialWrite(unsigned char data) {
    // Asteptam ca buffer-ul UDRE0 de transmisie sa se goleasca
    // verifica daca bitul UDRE0 e activ
    while (!(UCSR0A & (1<<UDRE0)));
    // Scrie un caracter in registrul de date pentru al transmite
    UDR0 = data;
}
```

- trimitere caractere

```
// trimite caracter cu caracter pana ajunge la final
void manualSerialPrint(const char* str) {
    while (*str) {
        manualSerialWrite(*str++);
    }
}
```

- transformarea numerelor in caractere

```
// transforma numar intreg in caracter ascii si il transmite caracter cu
caracter
void manualSerialPrintNumber(int num) {
    if (num == 0) {
        manualSerialWrite('0');
        return;
    }

    char buffer[10];
    int i = 0;
    bool negative = false;

    if (num < 0) {
        negative = true;
        num = -num;
    }
    ...
}
```

- **Serial 1**

- asemanator ca la serial 0 (UART1), insa la registri se schimba numarul in 1
- initializare

```
void manualSerial1Init(long baud) {
  // Setare registri pentru UART1 (Serial1)
  // calculare viteza de comunicare, F_CPU(viteza procesorului)
  uint16_t ubrr = F_CPU/16/ baud-1;
  // impartim pe 2 registri valoarea
  UBRR1H = (unsigned char)(ubrr>>8);
  UBRR1L = (unsigned char)ubrr;
  // Activeaza receptorul (RXEN1) si transmisorul (TXEN1) pentru UART0
  UCSR1B = (1<<RXEN1)|(1<<TXEN1);
  // setez marimea datelor la 8 biti
  UCSR1C = (1<<UCSZ11)|(1<<UCSZ10);
}
```

```
void manualSerial1Write(unsigned char data){
  ...
}
```

- Citire de date pe UART1

```
unsigned char manualSerial1Read() {
  // Asteapta sa se primeasca date pe uart1 apoi il citeste, RXC1 = activ =>
  date disponibile
  while (!(UCSR1A & (1<<RXC1)));
  return UDR1;
}
```

- Verificare dacă sunt date primite

```
bool manualSerial1Available() {
  // Verifica daca exista date primite in buffer-ul UART1
  return (UCSR1A & (1<<RXC1));
}
```

- **clasa pentru serial1 care extinde stream**

- **setup()**

1. Configurare pini: Setează LED-urile ca OUTPUT
2. Întrerupere: attachInterrupt() pentru detectarea atingerii
3. Inițializare comunicații: Serial pentru debug, Serial1 pentru DFPlayer
4. Configurare LCD: Pornește backlight-ul și afișează interfața
5. Configurare DFPlayer: Setează volumul și începe redarea

```
void setup() {
  // setare pini rgb pentru iesire
  int pins[] = {led1R, led1G, led1B, led2R, led2G, led2B};
  for (int i = 0; i < 6; i++) pinMode(pins[i], OUTPUT);

  // pinul touch(2) ca input
  pinMode(touchPin, INPUT);
  // Seteaza o intrerupere hardware pe pinul conectat la senzorul de
  atingere (touchPin),
```

```
// care va executa automat functia schimbareMod() atunci cand semnalul de
pe pin trece
// de la LOW la HIGH (adica o atingere pe senzor).
attachInterrupt(digitalPinToInterrupt(touchPin), schimbareMod, RISING);

// initializare comunicatii seriale manual
manualSerialInit(9600);
manualSerial1.begin(9600); //dfplayer manual

// pornire lcd
lcd.begin();
lcd.backlight();

if (!player.begin(manualSerial1)) {
    manualSerialPrintln("DFPlayer nu a fost gasit!");
    while (true);
}

// setam pe dfplayer volumul si melodia
player.volume(volum);
player.play(mode);

// afisam pe lcd
lcd.setCursor(0, 0); lcd.print("Mod: ");
lcd.setCursor(0, 1); lcd.print("Volum: ");
afiseazaMod();
afiseazaVolum();
}
```

## loop()

- Detectare atingere multiplă a senzorului touch. Atunci când s-a detectat o atingere se porneste flag-ul care anunță ca o să se i-a o decizie. Decizia se i-a într-un interval de 1,5s. Dacă în acest interval de timp s-a făcut o singură atingere atunci se schimbă doar **modul de luminare**, dacă s-au făcut 2 atingeri în intervalul acesta de timp atunci se **mareste volumul** și dacă s-au făcut 3 atingeri atunci se scade volumul. În această parte m-am folosit de **millis()** pentru a calcula timpul, întrucât dacă utilizăm delay() atingerile nu s-ar fi detectat la timp din cauza lui, trebuia să aștept până când trecea perioada de timp.
- Dacă s-a detectat funcția de schimbare a modului, se redă modul de luminat, melodia specifică modului, afisarea pe lcd a modului și a volumului. Pentru asta utilizez un contor al modurilor. În total sunt 6 moduri de luminat.
- Dacă s-a detectat schimbarea volumului atunci se scade/mareste volumul și se afisează pe lcd.
- După se termină melodia specifică modului se repune din nou (redare continua).

## Moduri de luminare

### • Mod 1 - staticColors()

- Funcție: Reacționează la lumina ambientală, în funcție de valorile primite de la fotorezistor.
- Logică: Dacă valoarea scade sub 250 (întuneric) → ciclează Roșu/Verde/Albastru
- Peste 250 (luminos) → LED-urile se sting

### • Mod 2 - fadingColors()

- Efect: Tranziție rainbow smooth, se joaca si cu intensitatea luminii
- Faze: Roșu→Galben→Verde→Cyan→Albastru→Magenta→Roșu
- **Mod 3 - blinkRGB()**
  - Efect: Clipire rapidă în alb
  - Frecvență: 200ms ON/OFF
- **Mod 4 - fadeWhiteBlue()**
  - Efect: Tranziție relaxantă alb↔albastru
  - Utilizare: Mod de relaxare/somn
- **Mod 5 - rainbowFade()**
  - Efect: Rainbow dinamic cu direcții independente pentru R/G/B
  - Complexitate: Fiecare culoare se mișcă independent
- **Mod 6 - pingPong()**
  - Efect: Alternare între LED1 și LED2
  - Culoare: Fuchsia (roșu + albastru)

### **schimbareMod()**

- Funcția de întrerupere (ISR)
- Setează doar flag-ul, procesarea se face în loop()

### **setAll()**

- Setează ambele LED-uri la aceeași culoare
- Folosește analogWrite() pentru PWM


### **Cod github**

[Proiect IP](#)

## **Rezultate Obținute**

Dupa terminarea constructiei hardware si software, am realizat prototipul la lampa(fara exterior prietenos vizual care sa arate ca o lampa). La pornirea ei se va porni automat modul 1 de luminat care este afisat si pe lcd impreuna cu volumul difuzorului. La primul mod nu o sa porneasca luminile automat daca in camera lumina este puternica, daca in camera este intuneric atunci va incepe jocul de lumini, melodia este redada in fundal indiferent de lumina de afara. Celelalte moduri nu depind de lumina ambientala, fiecare mod are o melodie proprie si joc de lumini propriu. Pentru a schimba modul trebuie facut o singura apasare pe senzor, pentru a marii volumul 2 apasari pe senzor, iar pentru a scade volumul 3 apasari.

## **Concluzii**

A fost un proiect interesant de implementat si mi-a facut placere sa lucrez la el. Pe parcursul lui am intampinat si mici probleme cum ar fi lipirea pinilor de la adaptorul I2C la lcd (nu mi-a iesit si am cumparat altul gata lipit ) , dar si probleme de afisare a scrisului pe lcd (solutia a fost o alta biblioteca decat cele gasite pe arduino IDE, se află in arhiva cu proiectul), probleme cu citirea

dfplayer-ului, dupa ore de stat si de analizat problema, mi-am pus intrebarea daca poate sa fie de la pinul care a venit lipit prost. Asa ca am mai cumparat alt dfplayer si inca un card microsd cu o clasa mai inferioara (de preferat clasele 1-4 si maxim 32gb) si pana la urma a functionat, iar in cod mi-a placut sa ma joc cu modurile de luminat.

## Download

Arhiva contine: poze cu schema, biblioteca de la lcd+i2c si codul.

[lampa\\_interactiva\\_neculau\\_sanda.zip](#)

## Bibliografie/Resurse

Pentru a afla registri pentru scrierea seriala m-am documentat din urmatoarele resurse [documentatie placuta artmega 2560](#), pagina 400(tabel cu registri descrisi)

Pentru a implementa functiile de scriere citire si transmitere pe serial [datasheet](#), paginile 149, 150, 151, 152 [Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2025/atoader/sanda\\_elena.neculau](http://ocw.cs.pub.ro/courses/pm/prj2025/atoader/sanda_elena.neculau)



Last update: **2025/05/27 19:15**