

Apollo party music lights

Autor

- Nume: Buracu Ștefan Paul
- Grupa: 331CB
- Îndrumător: Ionuț Oțelea

Introducere

Am ales sa fac acest proiect pentru ca imi place sa ascult muzica si imi place sa merg in cluburi asa ca m-am decis sa imi imbunatatesc viata atunci cand stau acasa si sa imi imaginez macar putin ca ma distrez. Astfel am ales sa imbin utilul cu placutul si sa realizez un sistem de redare audio cu o boxa, ce citeste melodii de pe un card SD si va avea lumini led care se vor anima in functie de melodie. Daca tot nu poti iesi in club poti aduce clubul la tine acasa 😊

Descriere generala

- Arduino -ul va fi alimentat la laptop prin un cablu USB. Cantecele sunt salvate pe un card microSD din care vor fi citite si luate ca date de intrare pentru Arduino. Ca output voi avea o boxa pentru a se auzi melodiile si niste leduri RGB la banda care vor fi prograte de mine. Am ca input dar si output un senzor de sunet care atunci cand va auzi boxa va transmite datele inapoi catre arduino pentru a aprinde led urile in functie de sunet, deci semnalele audio sunt redate prin boxa si interpretate prin senzor. Voi avea si butoane pentru a trece la urmatoarea melodie din playlist, la melodia precedenta sau pentru a pune pe pauza melodia curenta. Banda de led uri va contine in total 10 leduri conectate ce vor reda impreuna efectele dorite, avand o imagine mai clara.
- Schema bloc: Sagetile reprezinta flow-ul executiei, sagetile ce intra in componente reprezentand input-urile si cele ce ies output-urile.



Design Hardware

Lista piese folosite:

- Arduino Uno (ATmega328p)
- Difuzor
- 3 Butoane
- Fire
- Leduri RGB la banda (Neopixels) WS2812B
- Card SD compatibil cu adaptor
- Breadboard
- Modul microfon, senzor sunet
- Amplificator Audio LM386
- Sursa Baterie 9V

Conectarea componentelor: Modul card microSD:

- GND - GND
- VCC - 5V
- MISO - D12
- MOSI - D11
- SCK - D13
- CS - D4

Banda LED:

- GND - GND
- 5V - 5V
- DIN - rezistenta 220ohm - D8

Senzor sunet:

- GND - GND
- 5V - 5V
- OUT - A0

Amplificator Audio LM386:

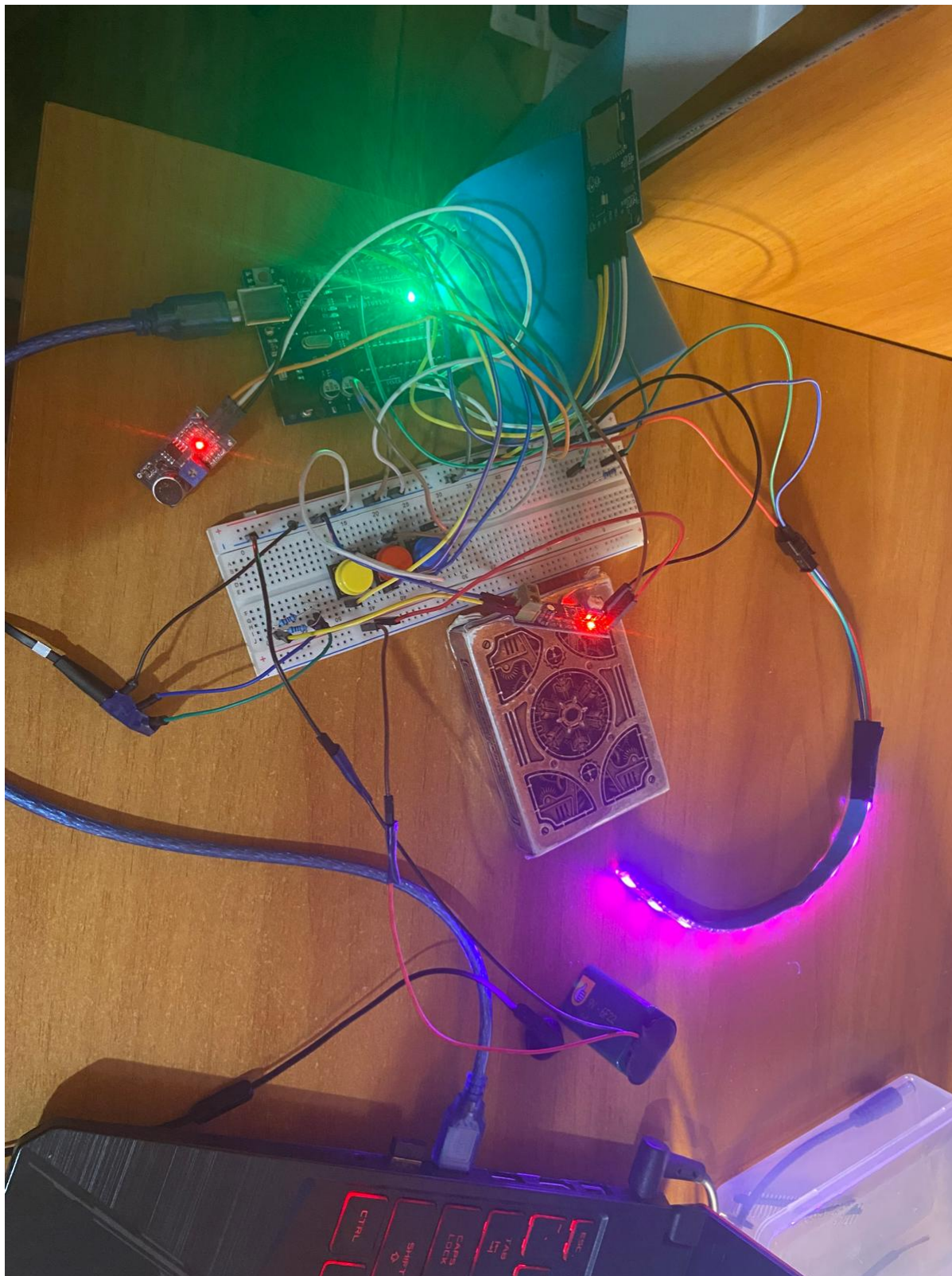
- VCC - + baterie 9V
- GND - GND
- IN - D9

Difuzor:

- - → GND
- casca stanga + rezistenta 220ohm → OUT amplificator
- casca dreapta + rezistenta 220ohm → OUT amplificator

Butoane: conectate cu un picior la GND si celalalt la D5, D6, D7





Design Software

Descrierea codului aplicației:

- Mediul de dezvoltare: Arduino IDE.

- Biblioteci folosite: SD, TMRpcm, FastLED, SPI
- Sursele mele: apollo_party_BuracuStefan331CB

Implementare:

Pentru inceput am incarcat cantece pe cardul microSD in formatul .wav cu specificatiile:

- rate: 16000 Hz
- audio channels: mono
- format: PCM unsigned 8-bit

Variabile globale

```
#include <SD.h> // biblioteca pentru cardul SD
#include <TMRpcm.h> // biblioteca pentru controlul difuzorului
#include <FastLED.h> // biblioteca pentru banda LED
#include <SPI.h>

// LED STRIP
#define LED_PIN 8
#define NUM_LEDS 10
#define LED_TYPE WS2812B
#define COLOR_ORDER GRB
CRGB leds[NUM_LEDS];
int pause_nr = 1;
uint8_t start_rainbow = 0;

// SD
#define SD_ChipSelectPin 4 // pinul de selectie a cipului pentru cardul SD
TMRpcm tmrpcm; // un obiect pentru biblioteca TMRpcm
int temp = 1; // variabila pentru selectarea cantecului
// butoane
int prev = 5;
int pause = 6;
int next = 7;

// senzor sunet
#define sensorPin A0
// valori de calcul
float sensorValue = 0, filteredSignal = 0;
float filteredSignalValues[] = {5.3, 5.2, 5.1, 5.05, 5.0, 4.95, 4.9, 4.85,
4.8, 4.6, 4.4, 4.2, 4.0, 3.7, 3.4, 3.1, 2.7, 2.4, 2.0, 1.6, 1.2, 0.8, 0.4,
0.20, 0.19, 0.18, 0.17, 0.16, 0.15, 0.14, 0.13};
```

Functii

- 1. `setup()`:

Defineste ledurile, configureaza pinii pentru butoane, pinul pentru difuzor, face verificarile pentru cardul microSD, seteaza volumul pentru muzica, verifica daca exista primul cantec si da play la acesta.

```
void setup() {
  delay(2000);
  // initializare banda LED
  FastLED.addLeds<LED_TYPE, LED_PIN, COLOR_ORDER>(leds, NUM_LEDS);
  FastLED.setBrightness(50);
  // butoane
  pinMode(pause, INPUT_PULLUP);
  pinMode(next, INPUT_PULLUP);
  pinMode(prev, INPUT_PULLUP);

  tmrpcm.speakerPin = 9; // pin pentru difuzor
  Serial.begin(9600);    // serial communication

  Serial.print("Initializing SD card...");
  if (!SD.begin(SD_ChipSelectPin)) { // incerc sa citesc cardul SD
    Serial.println("Card failed, or not present");
    return; // return daca nu il gaseste
  }
  Serial.println("SD card initialized.");
  char song[] = "song1.wav";
  // verifica daca am cantec
  if (SD.exists(song)) {
    Serial.println("File exists.");
  } else {
    Serial.println("File does not exist.");
  }

  // setez volum (0 to 7)
  tmrpcm.setVolume(6);

  // play song
  tmrpcm.play(song);
  if (tmrpcm.isPlaying()) {
    Serial.println("The sound is playing.");
  } else {
    Serial.println("The sound is not playing.");
  }
}
```

- 2. *FilterSignal(float sensorSignal):*

Calculeaza valoarea semnalului in functie de un semnal de la senzor

```
// functie calculat semnal
void FilterSignal(float sensorSignal) {
  filteredSignal = (0.8 * filteredSignal) + (0.04995 * sensorSignal);
}
```

- 3. *CompareSignal(float filteredSignal):*

Compara semnalul si decide in ce interval se afla acesta pentru a aprinde ledurile cu o culoare corespunzatoare

```
// functie colorare banda LED in functie de semnal
void CompareSignal(float filteredSignal) {
    if (filteredSignal > filteredSignalValues[i]) {
        fill_solid(leds, NUM_LEDS, CRGB::COLOR_i);
        FastLED.show();
        delay(50);
    }
}
```

- 4. *song (void):*

Alege la ce cantec sa dea play in functie de variabila temp

```
// functie alegere cantec in functie de variabila temp
void song (void)
{
    if (temp == i)
    {
        char song[] = "song_i.wav";
        tmrpcm.play(song);
    }
}
```

- 5. *loop():*

Aceasta functie are in vedere functionalitatea principala a programului dupa cum urmeaza:

- extrage semnalul senzorului in functie de ce aude
- foloseste functia de filtrare a semnalului
- verific daca butonul de pauza este apasat si aprind ledurile in model de curcubeu ce se deplaseaza atata timp cat e muzica oprita
- daca nu am pauza apelez functia de comparare a semnalului pentru a si aprinde ledurile dupa acest semnal
- daca am apasat butonul de pauza se opreste muzica
- daca am apasat butonul de next se incrementeaza temp si se apeleaza functia song pentru a da play la urmatorul cantec
- daca am apasat butonul de prev se decrementeaza temp si se apeleaza functia song pentru a da play la cantecul precedent

```
void loop() {
    // calculez o valoare pentru senzor
    sensorValue = (float) analogRead(sensorPin) * (5.0 / 1024.0);
    Serial.println(sensorValue);
    FilterSignal(sensorValue);
    // apelez functia pentru a aprinde banda LED in functie de semnal
    if (pause_nr % 2 == 0) {
        fill_rainbow(leds, NUM_LEDS, start_rainbow, 255/NUM_LEDS);
    }
}
```



```
FastLED.show();
delay(10);
start_rainbow ++;
} else{
  CompareSignal(filteredSignal);
}
// verific daca am apasat un buton
while (digitalRead(pause) == 0 || digitalRead(next) == 0 || digitalRead(
prev) == 0)
{
  // am dat pauza
  if (digitalRead(pause) == 0)
  {
    pause_nr ++;
    tmrpcm.pause();
    while (digitalRead(pause) == 0);
  }
  // am dat next
  else if (digitalRead(next) == 0)
  {
    if (temp < 20) {
      temp = temp + 1;
    } else {
      temp = 1;
    }
    while (digitalRead(next) == 0);
    song();
  }
  // am dat prev
  else if (digitalRead(prev) == 0)
  {
    if (temp > 1) {
      temp = temp - 1;
    } else {
      temp = 20; // Set temp to the last song
    }
    while (digitalRead(prev) == 0);
    song();
  }
}
}
```

Rezultate

Am reusit sa imi acopar toate planificarile pentru acest proiect folosind concepte spi pentru card SD, transformare din semnal analogic in digital cu senzor dar si pwm cu leduri 😊

Am atasat clipul:

- https://drive.google.com/file/d/13v_hs0RTp1iZvAQ_Inq0mK0wcsWTKSM7/view?usp=drive_link
- <https://www.youtube.com/watch?v=v2Cs-0qUORs&t=7s>

Concluzii

Mi-a placut sa lucrez la acest proiect, chiar daca la inceput am avut dificultati legate de ce piese sa cumpar si cum sa le leg impreuna, dupa mult stres a aparut si satisfactia cand am vazut ca totul functioneaza si ma pot juca cu melodiile si ledurile. Desi am ars un arduino si un difuzor, nu m-a oprit nimic sa continui proiectul si sa merg mai departe 🤔. Am avut de munca cu tema culorilor alese pentru a reinterpretata globul original din apollo, dar eu spun ca am facut o treaba buna si acum il am si acasa 😊.

Download

Arhiva cu codul: [apollo_party_buracustefan331cb.zip](#)

Jurnal

- 4 mai: Creare pagina
- 14 mai: Detalii in plus
- 15 mai: Comandat alte piese
- 17 mai: Hardware
- 19 mai: Am scris cod pentru cardSD si cumva am reusit sa ard arduinoul 🤔
- 20 mai: Am luat alt arduino si am confectionat castile pentru proiect in loc de difuzorul pe care il aveam initial in plan
- 21 mai: Am continuat codul si am reusit sa redau melodiile pe casti si sa adaug butoane
- 22 mai: Am lucrat cu banda led si sensorul si le-am facut sa se aprinda in functie de acesta
- 23 mai: Am adaugat aprinderea ledurilor in curcubeu atunci cand muzica e data pe pauza
- 24 mai: Am mai retusat pagina de pe ocw

Resurse

Pieșe:

- <https://ardushop.ro/ro/home/1314-modul-microfon-senzor-sunet.html>
- https://ardushop.ro/ro/electronica/331-banda-leduri-rgb-neopixels-ws2812b.html#/125-leduri_metru-60
- https://www.optimusdigital.ro/ro/electronica-de-putere-amplificatoare-audio/45-amplificator-audio-lm386.html?search_query=lm386&results=4
- <https://www.optimusdigital.ro/ro/memorii/1516-modul-slot-card-microsd.html>

Videoclipuri utile:

- <https://www.youtube.com/watch?v=zj3sa5HV2Bg>
- <https://www.youtube.com/watch?v=gi9mqlha8n0&t=155s>

Export to PDF

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/iotelea/stefan_paul.buracuLast update: **2024/05/26 22:17**