

Infinity Mirror

Student: Pantelimon Andrei

Grupa: 335CA

Introducere

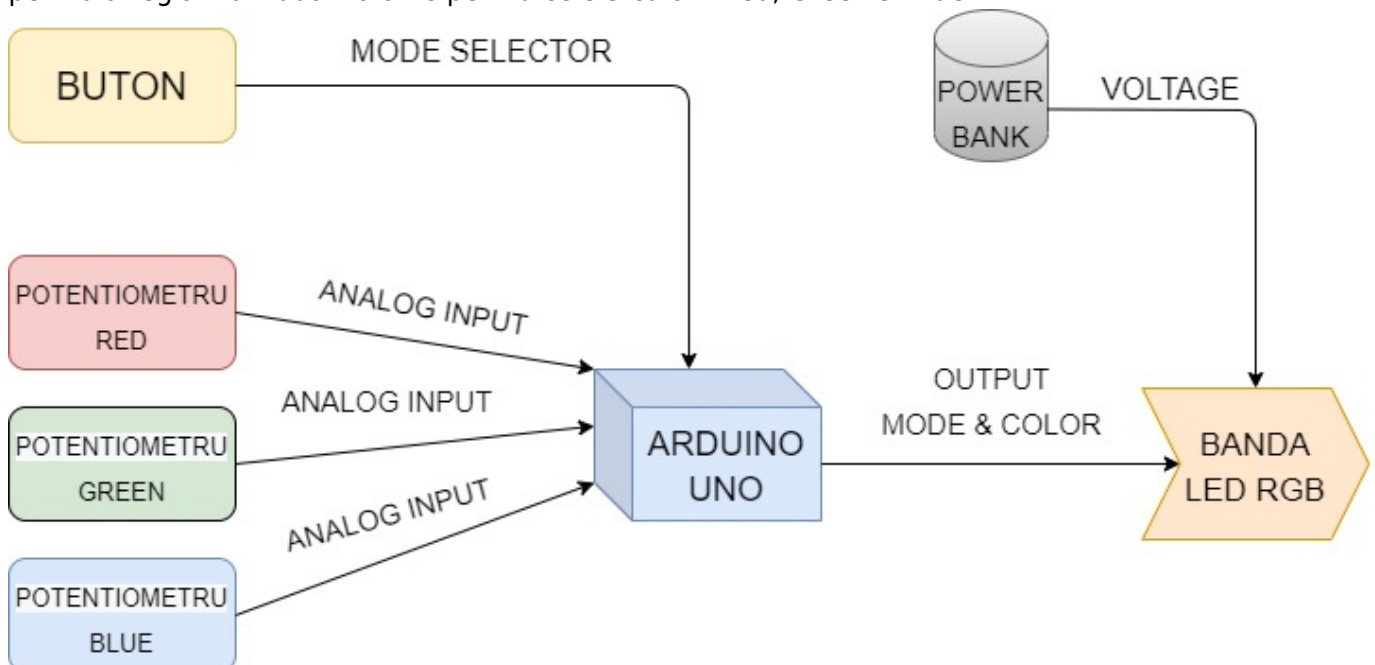
Proiectul consta in realizarea efectului de infinity mirror cu diferite efecte de lumina, folosind o banda LED RGB cu pixeli adresabili individual.

Efectul infinity mirror este unul foarte interesant, fiind realizat doar prin ajutorul a 2 oglinzi puse una in fata celeilalte, una dintre ele fiind de tip one-way.

Descriere generală

Pentru proiect am ales sa folosesc ca resurse hardware 3 potentiometre si un buton. Butonul va face cycle intre diferitele moduri de iluminare.

Pentru modul individual de iluminare se vor folosi potentiometrele pentru a regla individual valorile pentru cele 3 culori: Red, Green si Blue.



Hardware Design

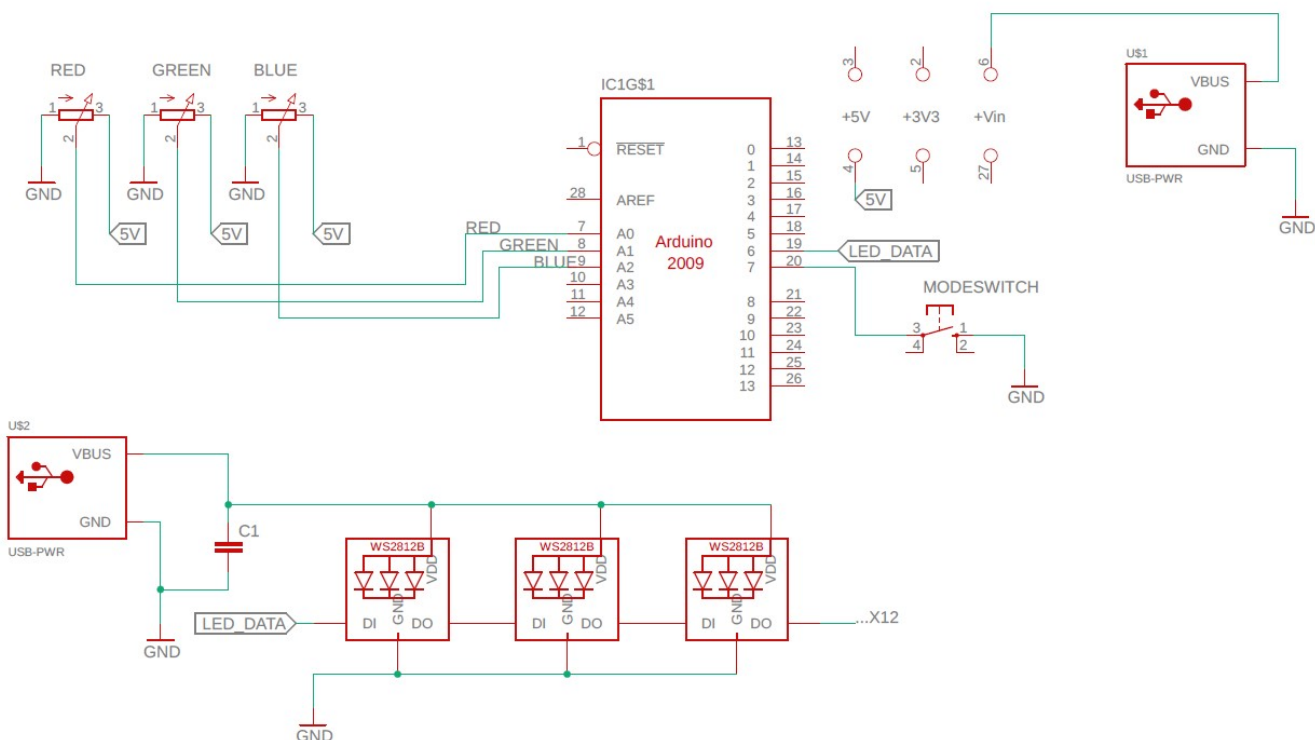
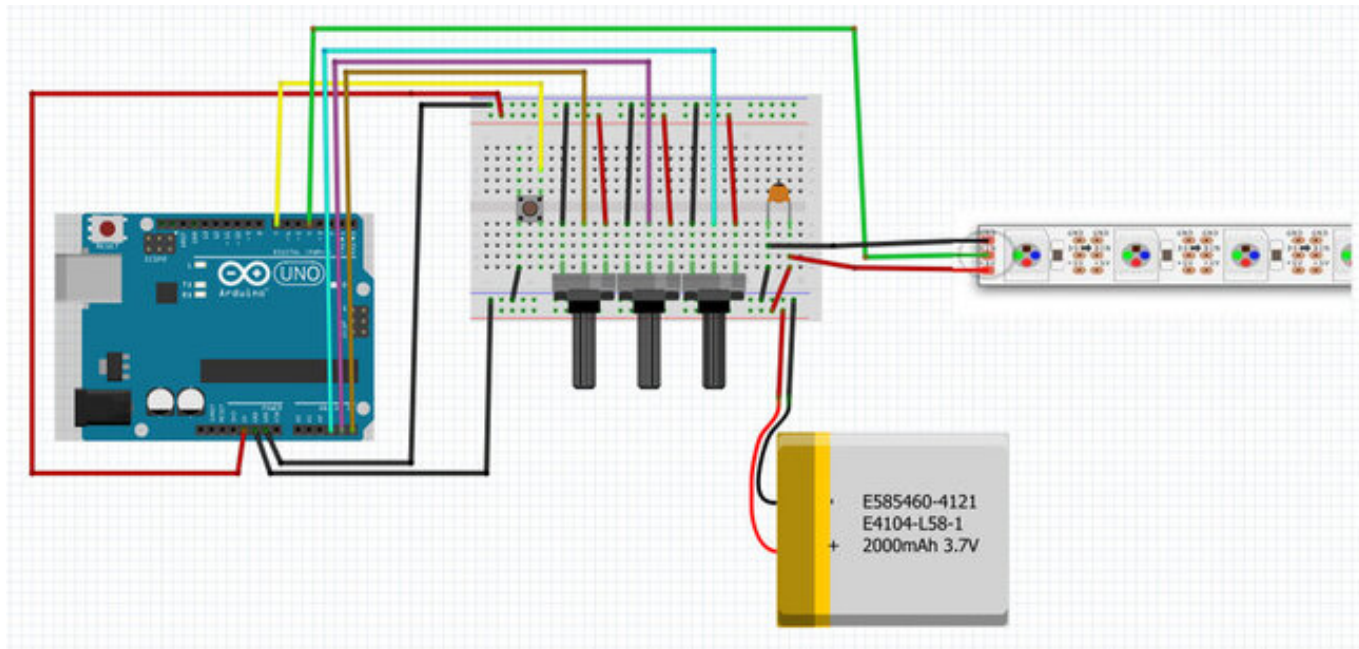
Lista componente:

- Arduino UNO
- Breadboard
- Fire
- 1 Buton
- 3 Potentiometre
- Banda LED Neopixels RGB
- PowerBank
- 1 Condensator

Pentru realizarea partii hardware am parcurs urmatoarele etape:

1. Am alimentat o parte din breadboard de la bateria externa si am legat banda LED

2. Am legat cele trei potentiometre pentru a realiza modul de reglaj manual al culorii
3. Am adaugat si butonul pentru a schimba intre modurile de operare alese
4. Am lipit banda LED in interiorul cercului de plastic
5. In spatele acestuia am fixat o oglinda taiata rotund
6. In partea din fata am lipit cu banda dublu adeziva un plexiglas peste care am aplicat o folie de tip one-way mirror
7. La final, am asezat tot proiectul pe un gratar de metal ca suport



Software Design

Am realizat partea de software folosind Arduino IDE. Dupa ce am testat toate piesele am realizat partea de schimbare a modului prin apasarea unui buton si apoi m-am decis la cele patru moduri de operare, si anume:

1. Modul Manual, in care banda lumineaza static pe o singura culoare ce se controleaza folosind cele trei potentiometre ce reprezinta canalele de red, green si blue
2. Modul Strobe, ce realizeaza efectul de stroboscop prin stingerea si aprinderea foarte rapida a intregii benzi cu culoarea alba
3. Modul Chase, in care ledurile se aprind pe rand si realizeaza 'un tur' al benzii.
Cand ajung de unde au plecat, alt sir porneste si realizeaza acelasi tur, sub alta culoare
4. Modul Twinkle, in care ledurile se aprind random, unul cate unul, si cu o culoare random generata

```
#include <Adafruit_NeoPixel.h>
#define PIN 6
#define N_LEDS 36

Adafruit_NeoPixel strip = Adafruit_NeoPixel(N_LEDS, PIN, NEO_GRB +
NEO_KHZ800);
uint32_t greenishwhite = strip.Color(0, 64, 0, 64);
int countColor = 0;
int previous = 0;

const int POT1 = A0;
const int POT2 = A1;
const int POT3 = A2;

const int BUTTON = 7;

int red = 0, green = 0, blue = 0;
int outputRed = 0, outputGreen = 0, outputBlue = 0;

int count = 0;

void setup() {
  strip.begin();
  pinMode(BUTTON, INPUT_PULLUP);
}
void loop() {
  int val = digitalRead(BUTTON);

  if (val == 0) {
    delay(1000);
    count++;
  }

  if (count > 3) {
    count = 0;
  }

  red = analogRead(POT1);
  green = analogRead(POT2);
  blue = analogRead(POT3);

  outputRed = map(red, 0, 1023, 0, 255);
```

```
outputGreen = map(green, 0, 1023, 0, 255);
outputBlue = map(blue, 0, 1023, 0, 255);

if (count == 0) {
  uint32_t currentColor = strip.Color(outputRed, outputGreen,
outputBlue);
  strip.fill(currentColor, 0, 36);
  strip.show();
}

if (count == 1) {
  Strobe(0xff, 0xff, 0xff, 1, 45, 0);
}

if (count == 2) {
  chase(strip.Color(255, 0, 0)); // Red
  chase(strip.Color(0, 255, 0)); // Green
  chase(strip.Color(0, 0, 255)); // Blue
}

if (count == 3) {
  TwinkleRandom(20, 100, false);
}
}
static void chase(uint32_t c) {
  for(uint16_t i=0; i<strip.numPixels()+4; i++) {
    strip.setPixelColor(i , c);
    strip.setPixelColor(i-4, 0);
    strip.show();
    delay(30);
  }
}

void Strobe(byte red, byte green, byte blue, int StrobeCount, int
FlashDelay, int EndPause){
  for(int j = 0; j < StrobeCount; j++) {
    setAll(red,green,blue);
    showStrip();
    delay(FlashDelay);
    setAll(0,0,0);
    showStrip();
    delay(FlashDelay);
  }
  delay(EndPause);
}

void setPixel(int Pixel, byte red, byte green, byte blue) {
  strip.setPixelColor(Pixel, strip.Color(red, green, blue));
}

void setAll(byte red, byte green, byte blue) {
```

```
    for(int i = 0; i < N_LEDS; i++ ) {
        setPixel(i, red, green, blue);
    }
    showStrip();
}

void showStrip() {
    strip.show();
}

void TwinkleRandom(int Count, int SpeedDelay, boolean OnlyOne) {
    setAll(0,0,0);

    for (int i=0; i<Count; i++) {
        setPixel(random(N_LEDS), random(0,255), random(0,255), random(0,255));
        showStrip();
        delay(SpeedDelay);
        if(OnlyOne) {
            setAll(0,0,0);
        }
    }

    delay(SpeedDelay);
}
```

Rezultate obtinute

Proiectul este 100% functional. Initial acesta se porneste pe modul manual, iar la apasarea butonului cicleaza prin diferitele moduri.

In Demo trec prin toate modurile, in modul manual modificand culorile benzii din potentiometre.



Demo proiect

Concluzii

Fiind primul proiect de acest gen, mi-a luat mai mult decat ma asteptam, mai ales la partea hardware pentru a obtine montajul dorit. In final mi-a iesit si sunt bucuros ca efectele sunt exact cum mi-am dorit. Pe viitor este posibil sa mai alterez modurile de functionare pentru a obtine si alte efecte. 😊

Jurnal

25.04.2021 : Alegere tema proiect si completare wiki cu schema initiala si componentele necesare

27.04.2021 : Sosirea pieselor si inceperea realizarii proiectului

29.04.2021 : Finalizarea proiectului

26.05.2021 : Realizare Demo si completare Wiki

Bibliografie/Resurse

[Ideea generala](#)

[Efecte NeoPixel](#)

[Wiki Biblioteca NeoPixels](#)

[infinity_mirror.pdf](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/infinity-mirror>

Last update: **2021/05/26 18:29**

