

LED Slot Machine

By Podeanu Marius-Bogdan 1222A

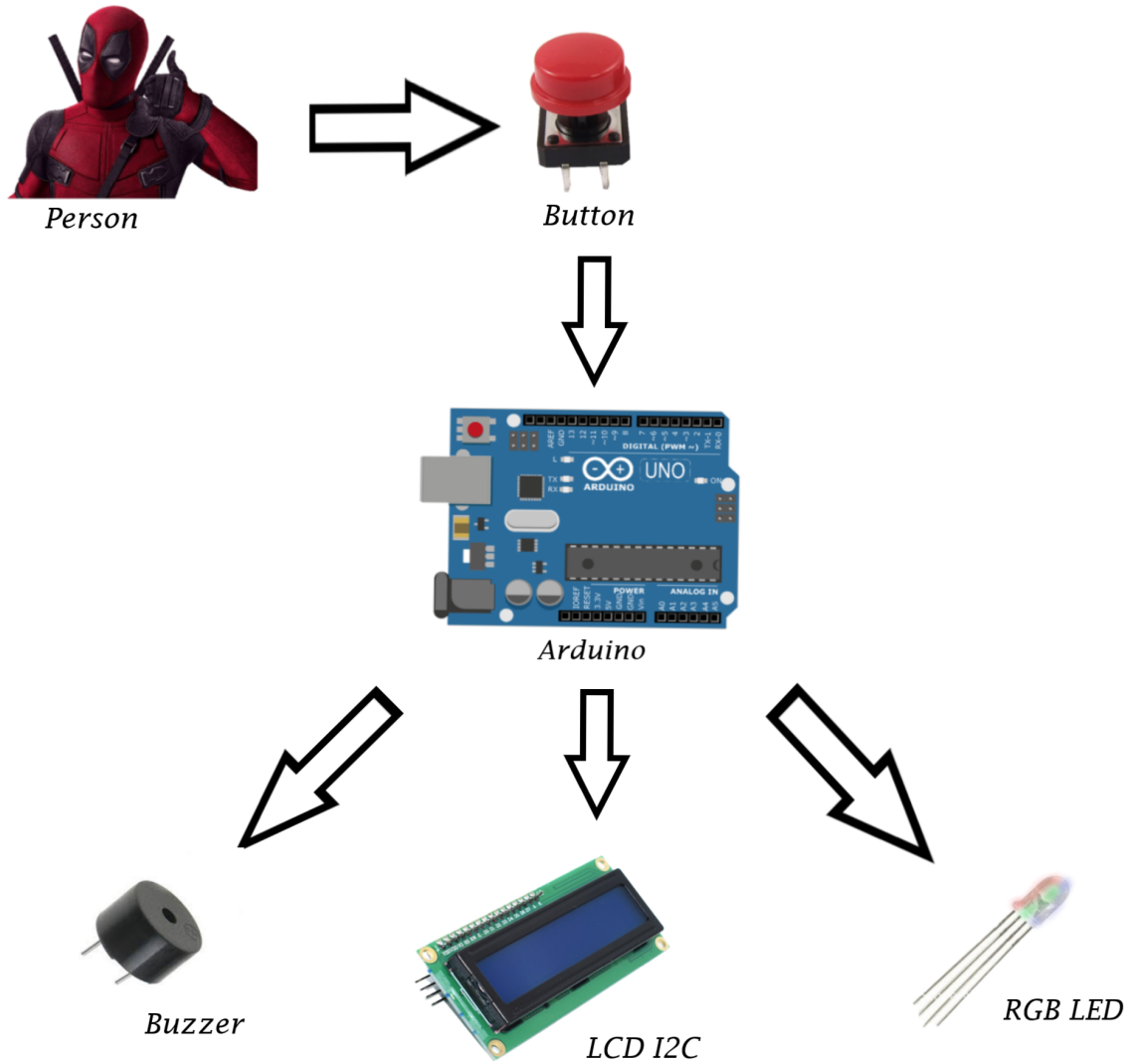
Introduction

LED Slot Machine este un mini joc distractiv in care 3 LED-uri se aprind intr-o culoare la intamplare. Scopul jocului este de a avea toate LED-urile aprinse in aceeasi culoare. In momentul in care castigi, muzica incepe sa cante iar pe ecranul LCD iti apare un mesaj.

Ideea a pornit de la varianta clasica de slot, dar facuta in asa fel incat sa fie cat se poate de fair.

General Description

Block Scheme:



Hardware Design

Components List:

- 1x Arduino Uno,
- 1x Breadboard,
- 3x LED RGB,
- 1x Push Button,
- 1x Ecran LCD I2C,
- 1x Buzzer

Electrical Schematic:



Software Design

Source Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
```

```
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```

```
int melody[] = {
NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
NOTE_D5, NOTE_D5, NOTE_D5, NOTE_D5,
NOTE_C5, NOTE_C5, NOTE_C5, NOTE_C5,
NOTE_F5, NOTE_F5, NOTE_F5, NOTE_F5,
NOTE_G5, NOTE_G5, NOTE_G5, NOTE_G5,
NOTE_G5, NOTE_G5, NOTE_G5, NOTE_G5,
```

```
NOTE_G5, NOTE_G5, NOTE_G5, NOTE_G5,  
NOTE_C5, NOTE_AS4, NOTE_A4, NOTE_F4,  
NOTE_G4, 0, NOTE_G4, NOTE_D5,  
NOTE_C5, 0, NOTE_AS4, 0,  
NOTE_A4, 0, NOTE_A4, NOTE_A4,  
NOTE_C5, 0, NOTE_AS4, NOTE_A4,  
NOTE_G4,0, NOTE_G4, NOTE_AS5,  
NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,  
NOTE_G4,0, NOTE_G4, NOTE_AS5,  
NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,  
NOTE_G4, 0, NOTE_G4, NOTE_D5,  
NOTE_C5, 0, NOTE_AS4, 0,  
NOTE_A4, 0, NOTE_A4, NOTE_A4,  
NOTE_C5, 0, NOTE_AS4, NOTE_A4,  
NOTE_G4,0, NOTE_G4, NOTE_AS5,  
NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,  
NOTE_G4,0, NOTE_G4, NOTE_AS5,  
NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5  
};
```

```
// note durations: 4 = quarter note, 8 = eighth note, etc.:
```

```
int noteDurations[] = {
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
4,4,4,4,
```

```
};
```

```
//Actual code
```

```
LiquidCrystal_I2C lcd(0x27,20,4);

int ledcolor;
int a = 1; //sets the delay until next color

int red1 = 8; //sets the RED led pin for FIRST RGB
int green1 = 9; //sets the GREEN led pin for FIRST RGB
int blue1 = 10; //sets the BLUE led pin for FIRST RGB

int red2 = 5; //sets the RED led pin for SECOND RGB
int green2 = 6; //sets the GREEN led pin for SECOND RGB
int blue2 = 7; //sets the BLUE led pin for SECOND RGB

int red3 = 11; //sets the RED led pin for THIRD RGB
int green3 = 12; //sets the GREEN led pin for THIRD RGB
int blue3 = 13; //sets the BLUE led pin for THIRD RGB

int button = 3;
int pressed = 0; //checks if the button is pressed

void setup() { //this sets the output pins

pinMode(red1, OUTPUT);
pinMode(green1, OUTPUT);
pinMode(blue1, OUTPUT);

pinMode(red2, OUTPUT);
pinMode(green2, OUTPUT);
pinMode(blue2, OUTPUT);

pinMode(red3, OUTPUT);
pinMode(green3, OUTPUT);
pinMode(blue3, OUTPUT);

pinMode(button, INPUT);

lcd.init();
// Print a message to the LCD.
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Press the button");
lcd.setCursor(4,1);
lcd.print("to play!");

RGB_color1(0, 0, 0);
RGB_color2(0, 0, 0);
RGB_color3(0, 0, 0);

}

void RGB_color1(int red_value, int green_value, int blue_value){
```

```
digitalWrite(red1, red_value);
digitalWrite(green1, green_value);
digitalWrite(blue1, blue_value);
}

void RGB_color2(int red_value, int green_value, int blue_value){
digitalWrite(red2, red_value);
digitalWrite(green2, green_value);
digitalWrite(blue2, blue_value);
}

void RGB_color3(int red_value, int green_value, int blue_value){
digitalWrite(red3, red_value);
digitalWrite(green3, green_value);
digitalWrite(blue3, blue_value);
}

void loop() {
int a = random(6); //randomly selects a number between 0 and 6
int b = random(6);
int c = random(6);
int ledcolor1 = a;
int ledcolor2 = b;
int ledcolor3 = c;
pressed = digitalRead(button);

if (pressed == LOW) {
delay(300);

switch (a) {
case 0: //if ledcolor equals 0 then the led will turn red
RGB_color1(255, 0, 0);
delay(a);
break;

case 1: //if ledcolor equals 1 then the led will turn green
RGB_color1(0, 255, 0);
delay(a);
break;

case 2: //if ledcolor equals 2 then the led will turn blue
RGB_color1(0, 0, 255); // Blue
delay(a);
break;

case 3: //if ledcolor equals 3 then the led will turn yellow
RGB_color1(255, 255, 0);
delay(a);
break;

case 4: //if ledcolor equals 4 then the led will turn cyan
RGB_color1(255, 255, 255);
delay(a);
```

```
break;

case 5: //if ledcolor equals 5 then the led will turn magenta
RGB_color1(255, 0, 255); // Magenta
delay(a);
break;
}

switch (b) {
case 0: //if ledcolor equals 0 then the led will turn red
RGB_color2(255, 0, 0);
delay(a);
break;

case 1: //if ledcolor equals 1 then the led will turn green
RGB_color2(0, 255, 0);
delay(a);
break;

case 2: //if ledcolor equals 2 then the led will turn blue
RGB_color2(0, 0, 255); // Blue
delay(a);
break;

case 3: //if ledcolor equals 3 then the led will turn yellow
RGB_color2(255, 255, 0);
delay(a);
break;

case 4: //if ledcolor equals 4 then the led will turn cyan
RGB_color2(255, 255, 255);
delay(a);
break;

case 5: //if ledcolor equals 5 then the led will turn magenta
RGB_color2(255, 0, 255); // Magenta
delay(a);
break;
}

switch (c) {
case 0: //if ledcolor equals 0 then the led will turn red
RGB_color3(255, 0, 0);
delay(a);
break;

case 1: //if ledcolor equals 1 then the led will turn green
RGB_color3(0, 255, 0);
delay(a);
break;

case 2: //if ledcolor equals 2 then the led will turn blue
RGB_color3(0, 0, 255); // Blue
delay(a);
```



```
break;

case 3: //if ledcolor equals 3 then the led will turn yellow
RGB_color3(255, 255, 0);
delay(a);
break;

case 4: //if ledcolor equals 4 then the led will turn cyan
RGB_color3(255, 255, 255);
delay(a);
break;

case 5: //if ledcolor equals 5 then the led will turn magenta
RGB_color3(255, 0, 255); // Magenta
delay(a);
break;
}
}

if (ledcolor1 == ledcolor2) {
if (ledcolor2 == ledcolor3) {
if (pressed == LOW){

lcd.init();
// Print a message to the LCD.
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Congratulations!");
lcd.setCursor(1,1);
lcd.print(" You won 10k!");

for (int thisNote = 0; thisNote < 112; thisNote++) {

int noteDuration = 750 / noteDurations[thisNote];
tone(2, melody[thisNote], noteDuration);

int pauseBetweenNotes = noteDuration * 1.3;
delay(pauseBetweenNotes);

noTone(2);
}

lcd.clear();
setup();

}
}
}
}
```

Libraries Used:

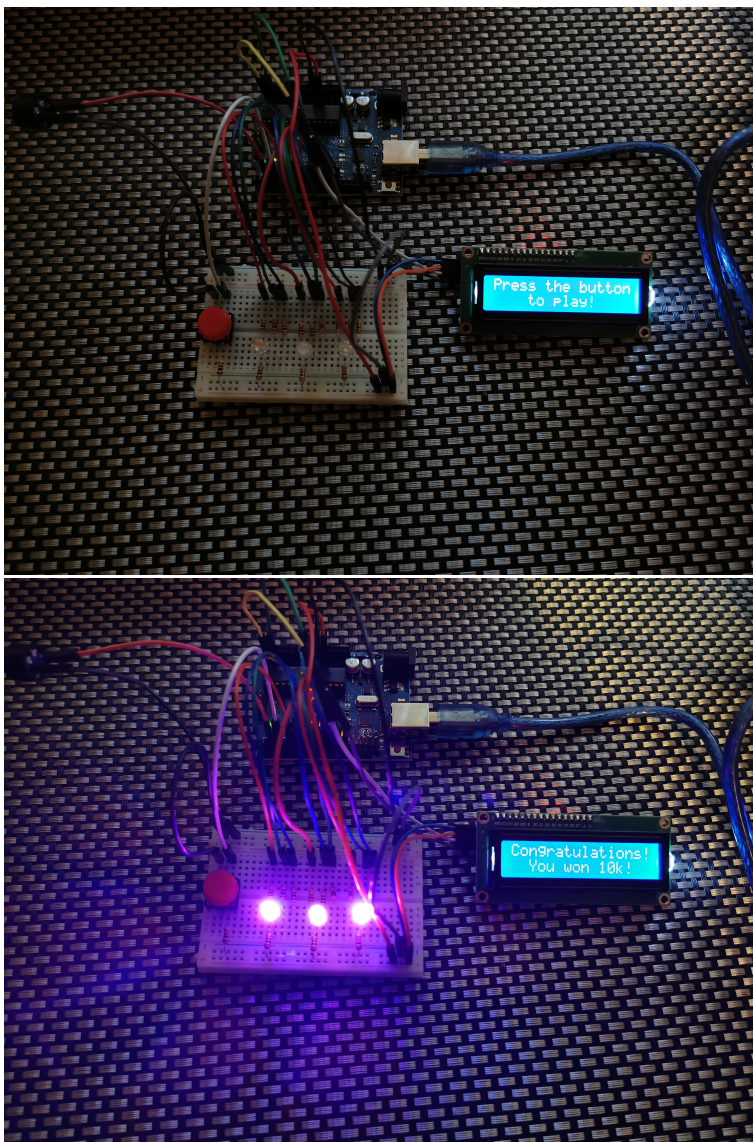
The only library needed was the one for the I2C LCD.

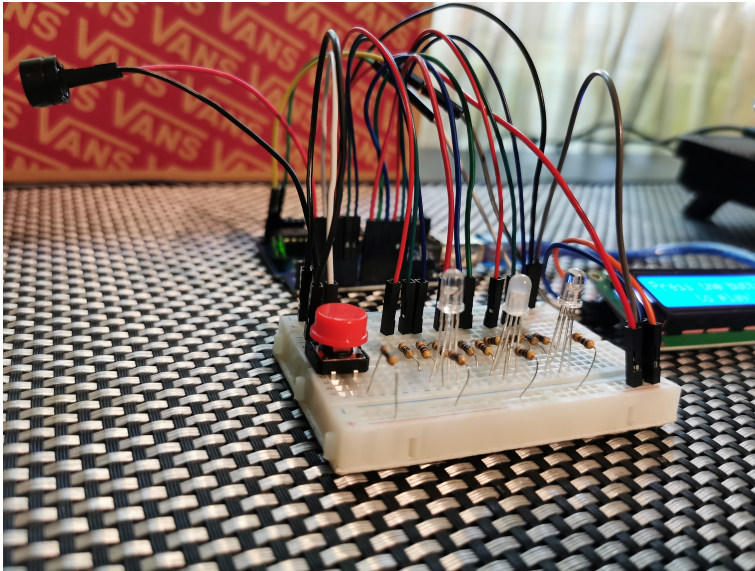
[liquidcrystal_i2c.rar](#)

Obtained Results

After many tries, I made it so that the game works. Now, when you press the button, the RGB's turn on into a random color.

If all 3 RGB's have the same color, the buzzer will start singing and the LCD will display a different message.





Soon, I will put it to use and make my friends pay in order to play with it.

Conclusion

The project was very interesting. I learned a lot of new things, how to use Arduino, and how to search things better on the internet.

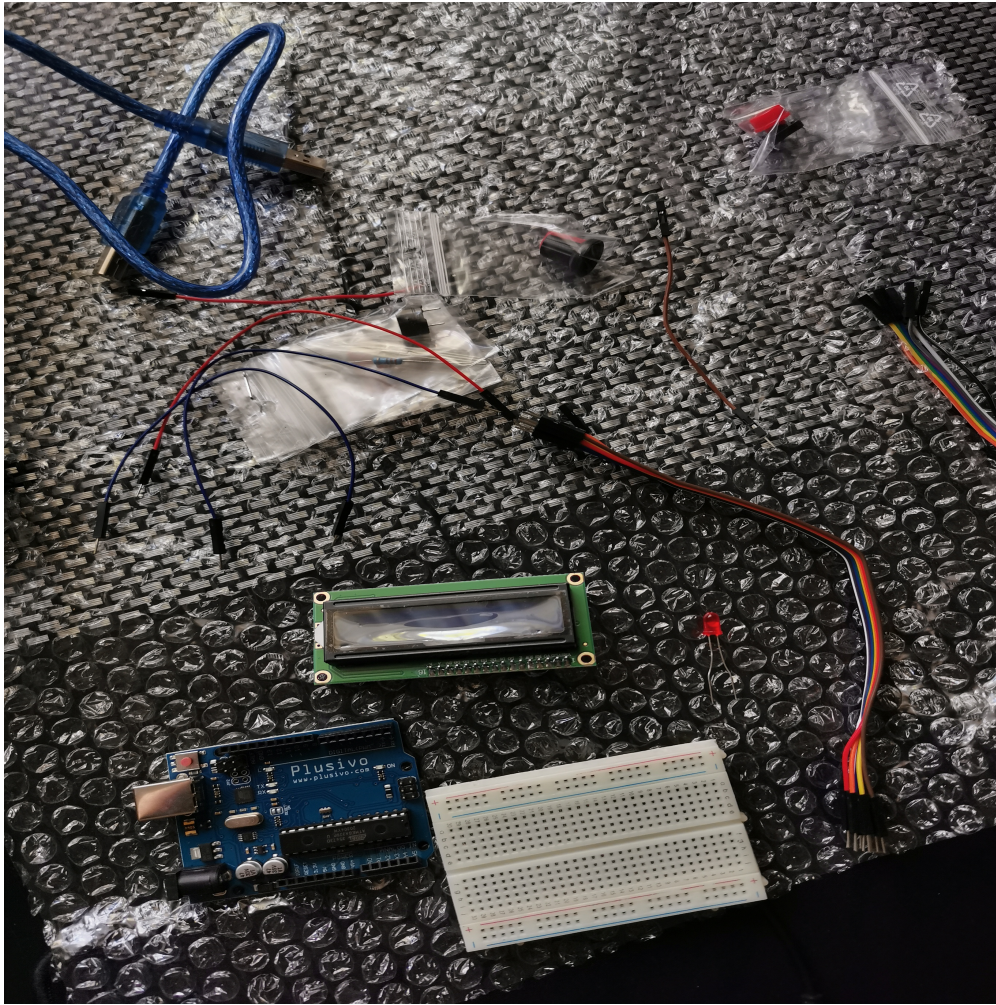
After this project, I wish to do more Arduino projects in the near future.

Journal

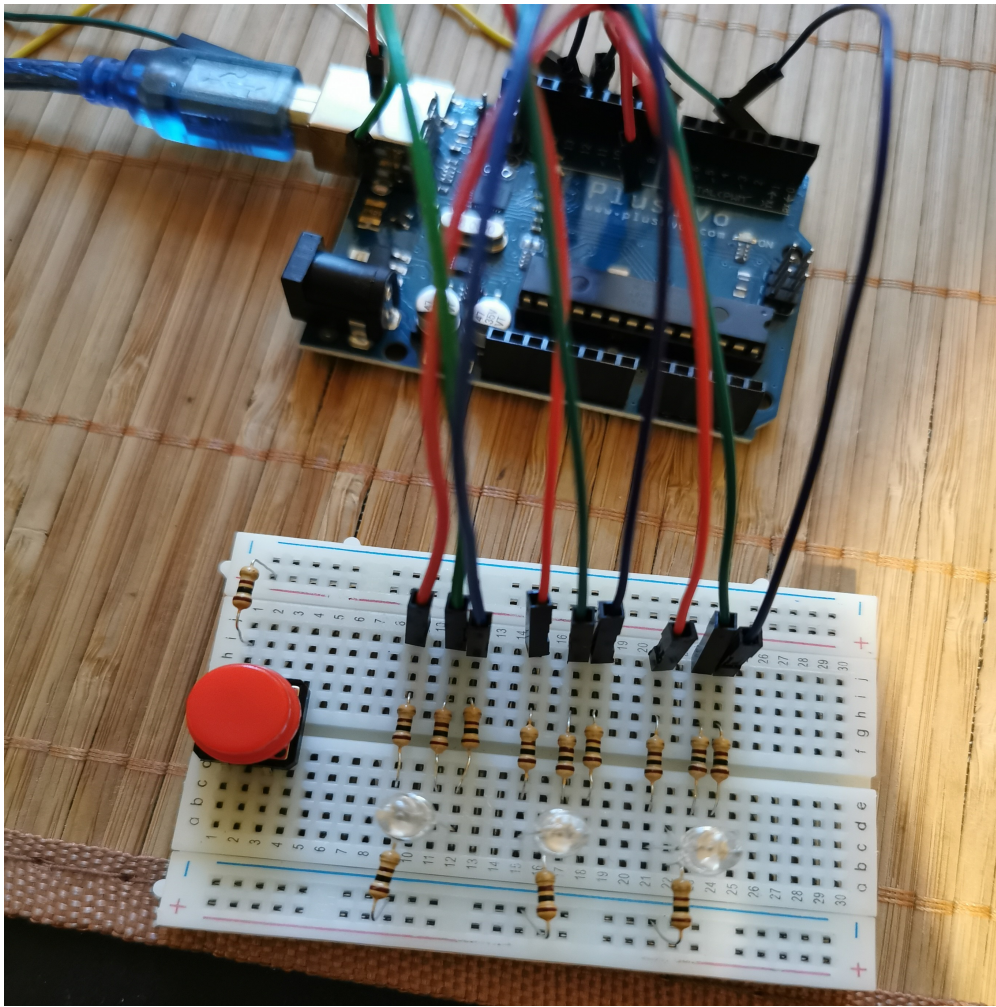
I started by doing the project in Tinkercad, so that I can do trial and error there.



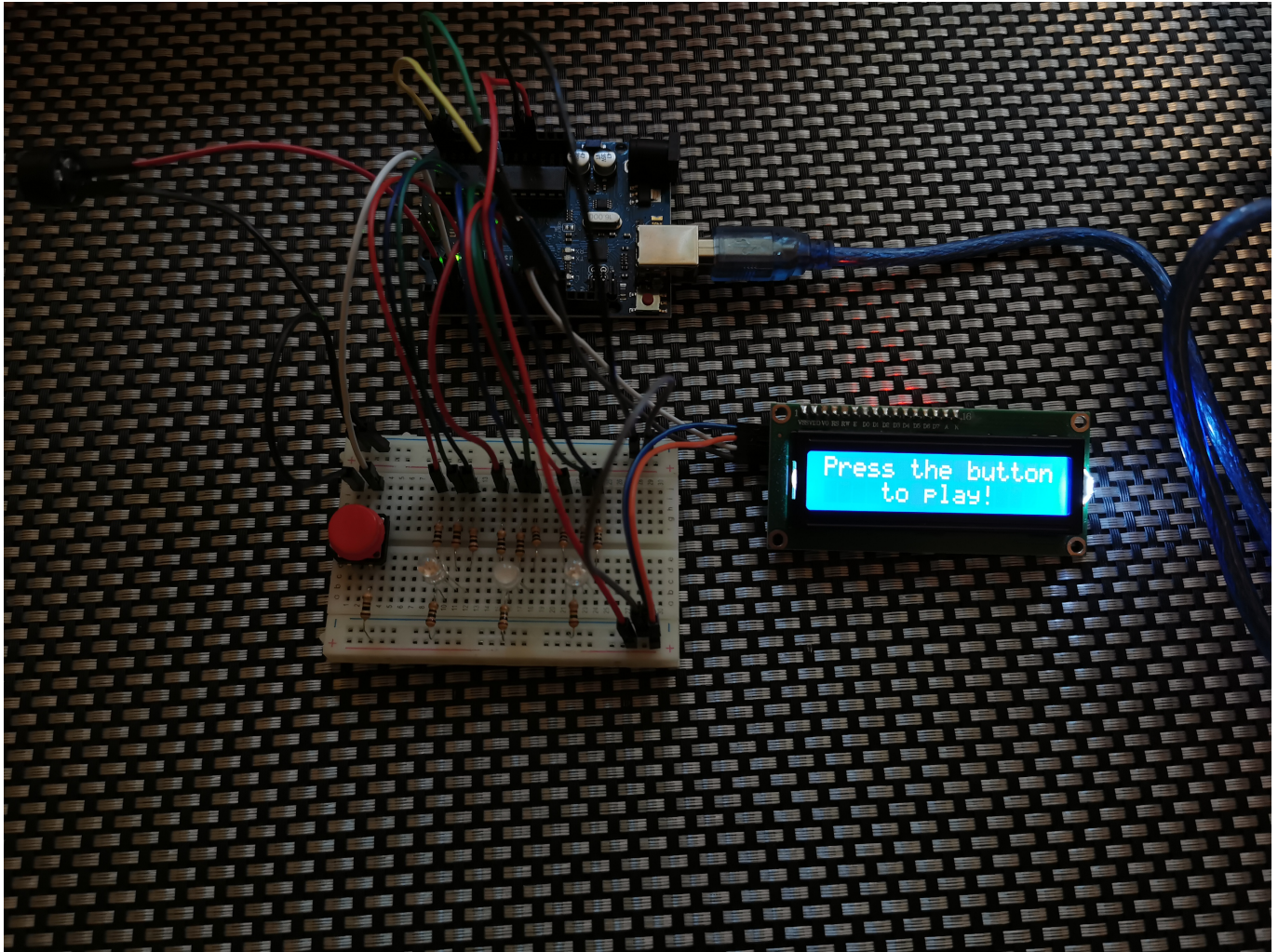
After I was done with it in Tinkercad, and I saw that it worked, I ordered my components and I got them.



I started thinking how should I arrange it so that it's not messy and I found the solution.



Without realising it, the project was already done.



Download

The code:

[led_slot_machine.rar](#)

The libraries:

[liquidcrystal_i2c.rar](#)

Bibliography / Resources

Link for buzzer melody:

[Buzzer Song](#)

Link for random RGB inspiration:

[Random RGB](#)

Download a PDF of this page:

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/cstan/led-slot-machine>



Last update: **2022/05/27 20:03**