

# Ping Pong Game

PETRE-ANTICA Eduard-Dominic 335CBa

## Introducere

- Proiectul reprezinta implementarea unui **joc de ping pong pentru 2 jucatori**. Acestia isi controleaza fiecare placuta lui de joc folosind **cate 2 butoane** (deplasare stanga, respectiv deplasare dreapta), iar jocul este vazut pe **display**. Jucatorii isi trec numele pe **interfata seriala**, iar cand unul din ei castiga, **LED-ul** corespunzator se aprinde.
- Scopul proiectului este acela de a arata ca notiunile pe care le-am invatat la laborator, desi de baza, pot fi folosite pentru a crea ceva ce nu este doar **functional**, dar si **distractiv**.
- Ideea de la care am plecat in crearea acestui proiect a fost cea de a simula un joc de Ping Pong in *stilul unui joc de tip Arcade*. Mi-a venit in timp ce dezvoltam proiectul ideea de a folosi 2 *mini-breadboard-uri pe post de controllere*, astfel incat jucatorii sa poata controla mai bine piesele lor si sa recreez mai bine sentimentul de joc video.
- Consider ca proiectul este util pentru mine pentru ca m-a ajutat *sa pun in practica notiunile invatate de-a cursul semestrului*, iar pentru colegii mei, sper ca reprezinta un joc de care acestia sa se bucure.

## Descriere generală

Pentru a simula cat mai bine ideea unui joc 2-player, fiecare jucator va folosi un **mini-breadboard** pe post de controller.

- fiecare controller va contine **butoane** cu care jucatorii isi vor putea muta placile

Pentru a scrie cate un nume pentru fiecare jucator, se foloseste comunicarea cu calculatorul prin **interfata USART**. Scriind in Serial Monitor cate un nume atunci cand pe display apare aceasta cerinta, se retin cele 2 nume ce vor fi apoi folosite pentru a arata castigatorul si scorurile.

De asemenea, exista un breadboard principal, ce contine:

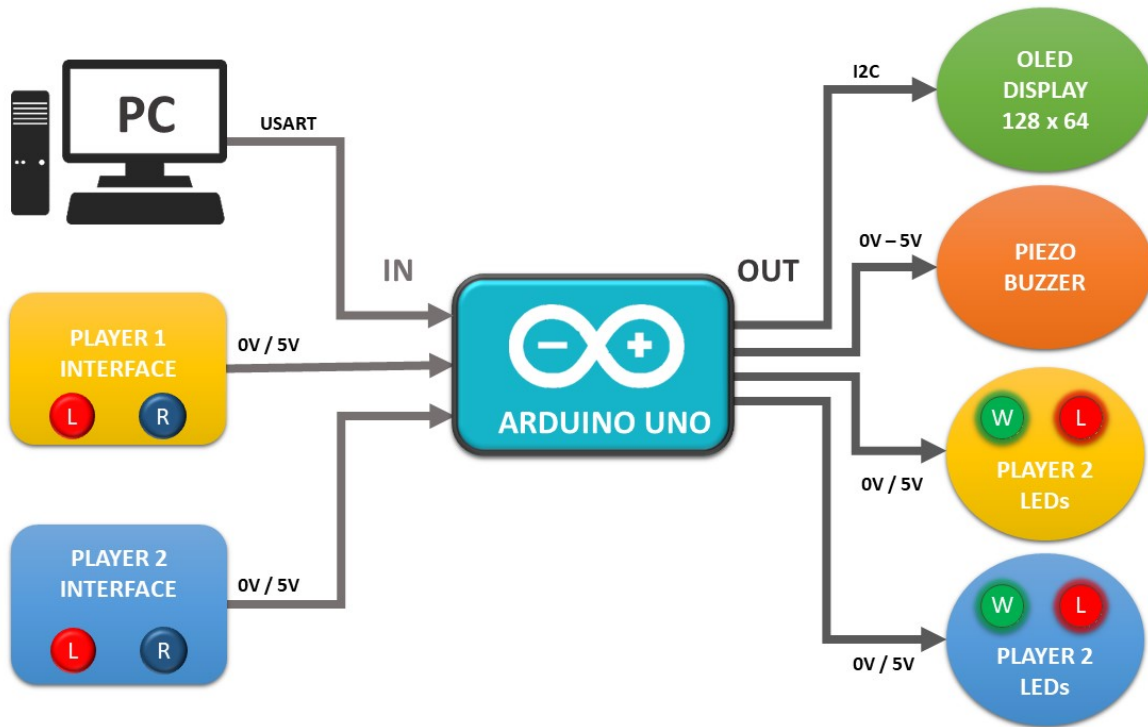
- **display-ul** folosit drept componenta grafica a jocului
- **buzzer-ul piezoelectric** folosit drept componenta auditiva a jocului
- **LED-uri rosii si verzi** folosite pentru a prezenta castigatorul runde curente a jocului

Pe scurt, placuta Arduino UNO foloseste drept input-uri:

- **interfata USART** pentru comunicarea cu calculatorul
- **cele 4 butoane** de pe cele 2 mini-breadboard-uri,

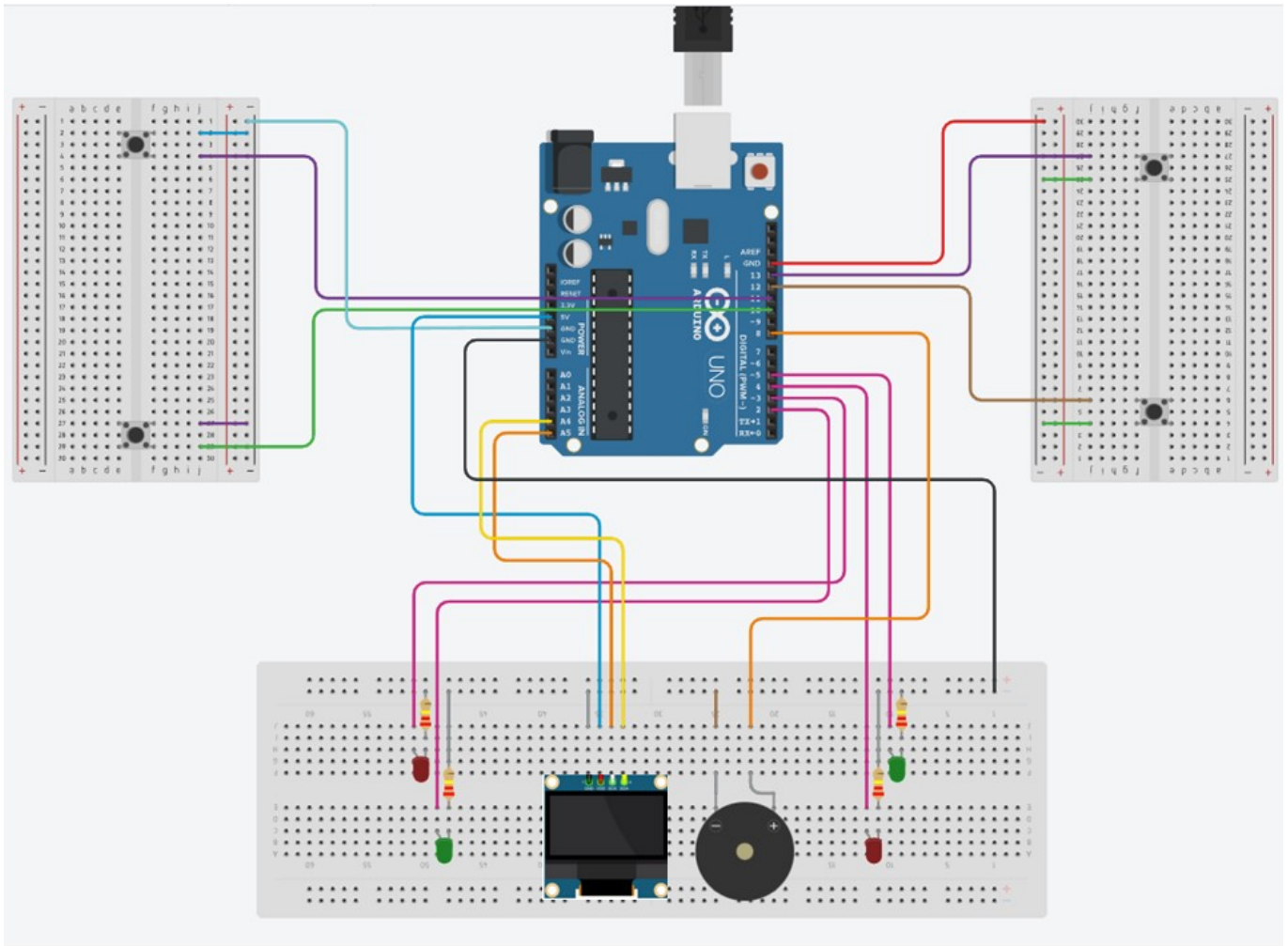
iar drept output:

- **display-ul OLED**
- **buzzer-ul piezoelectric**
- **LED-urile**



## Hardware Design

[Schema TINKERCAD:](#)



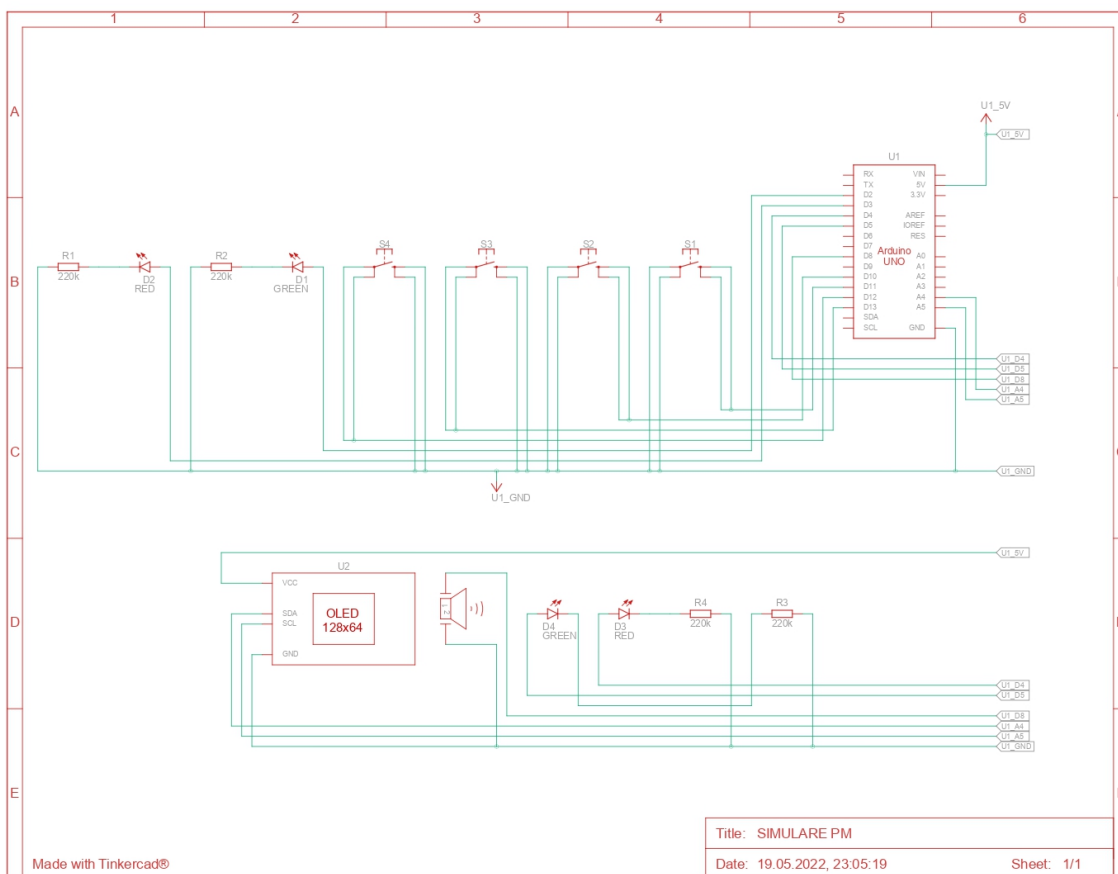
Pentru a asigura faptul ca LED-urile nu se vor arde daca vor fi conectate direct la sursa, acestea au fost legate in serie cu **rezistente de  $R=220k\Omega$** .

Intrucat am dorit ca jucatorii sa aiba suficient spatiu pentru a controla mini-breadboard-urile, am folosit fire tata-tata legate la fire mama-tata.

Componente hardware utilizate:

Nume piesa	Numar
Arduino UNO	1
OLED Display 128x64	1
Buzzer piezoelectric	1
LED-uri rosii	2
LED-uri verzi	2
Breadboard	1
Breadboard mini	2
Fire tata-tata	18
Fire mama-tata	6

[Schema electrica:](#)



## Software Design

Biblioteci utilizate:

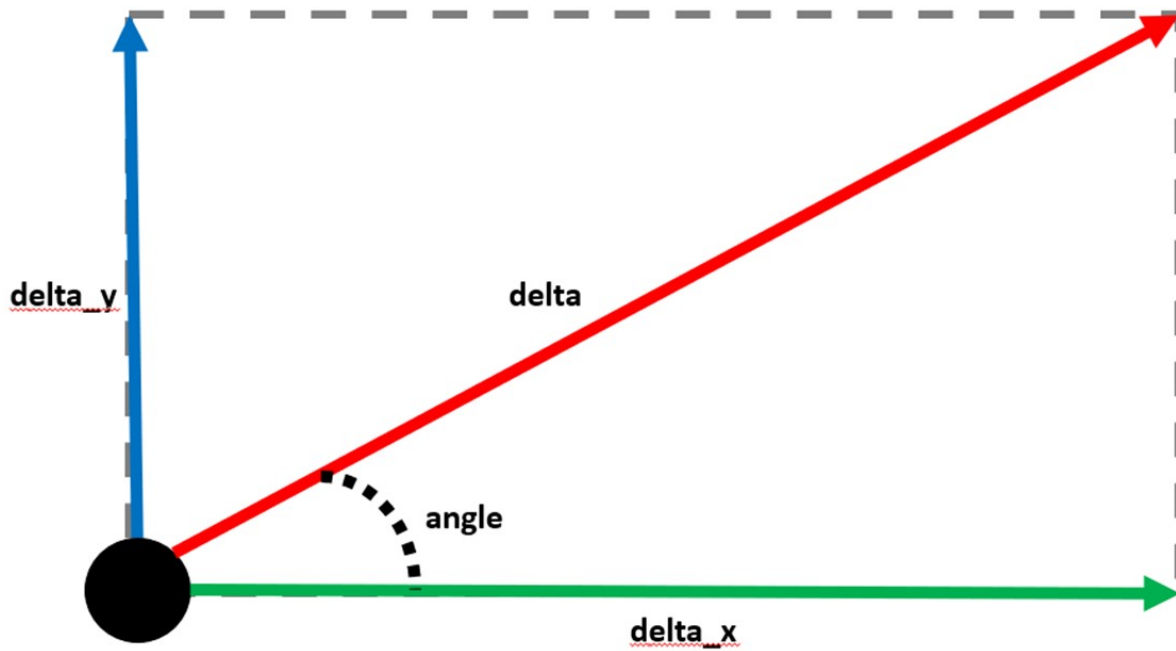
1. **SPI.h** - biblioteca folosita pentru comunicarea pe interfata seriala cu perifericele
2. **Wire.h** - biblioteca folosita pentru comunicarea cu perifericele ce folosesc I2C/TWI. Implicit, aceasta foloseste pinul A4 drept SDA si A5 drept SCL
3. **Adafruit\_GFX.h** - biblioteca ce ofera un API pentru LCD/OLED displays
4. **Adafruit\_SSD1306.h** - biblioteca ce ofera un API specific pentru display-ul OLED 128x32 si 128x64 monocromatic

Pentru a afla distantele la care se deplaseaza obiectele de pe display (minge si placi), am folosit o implementare ce se foloseste de timpul dintre 2 rulari consecutive ca un framerate dupa care sa se afle distantele. Astfel, chiar daca intre 2 rulari consecutive framerate-ul difera cu cateva milisecunde, piesele vor fi percepute ca miscandu-se cu aceeasi viteza.

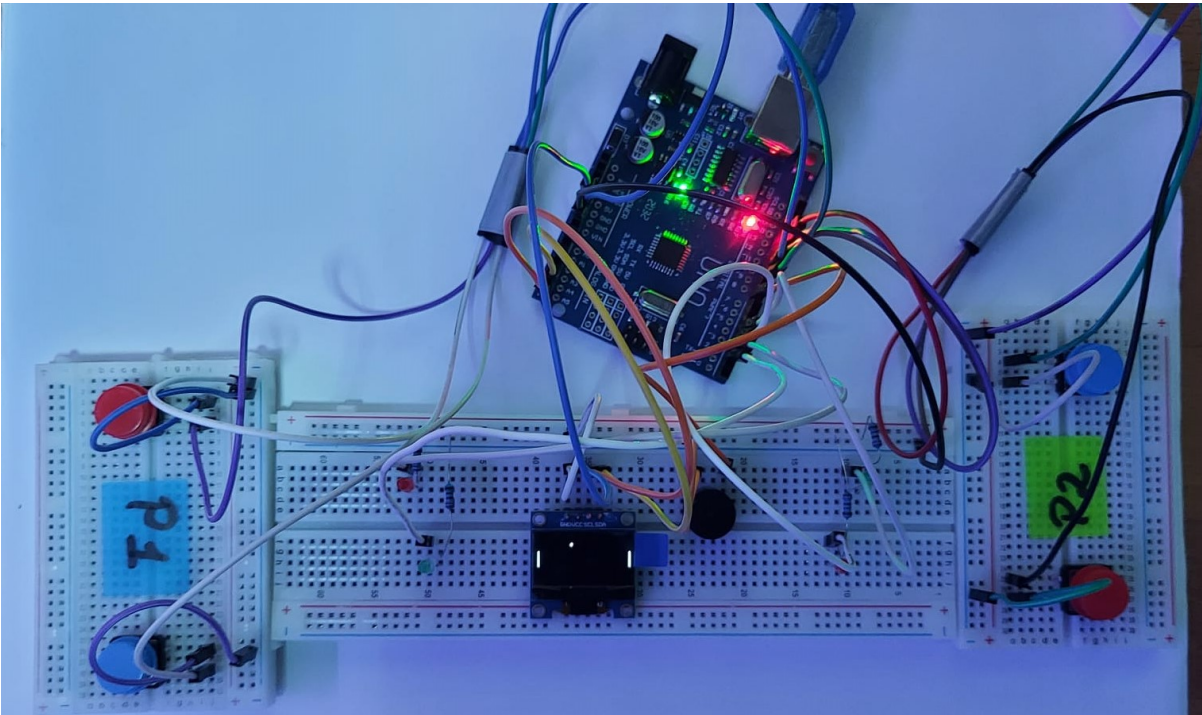
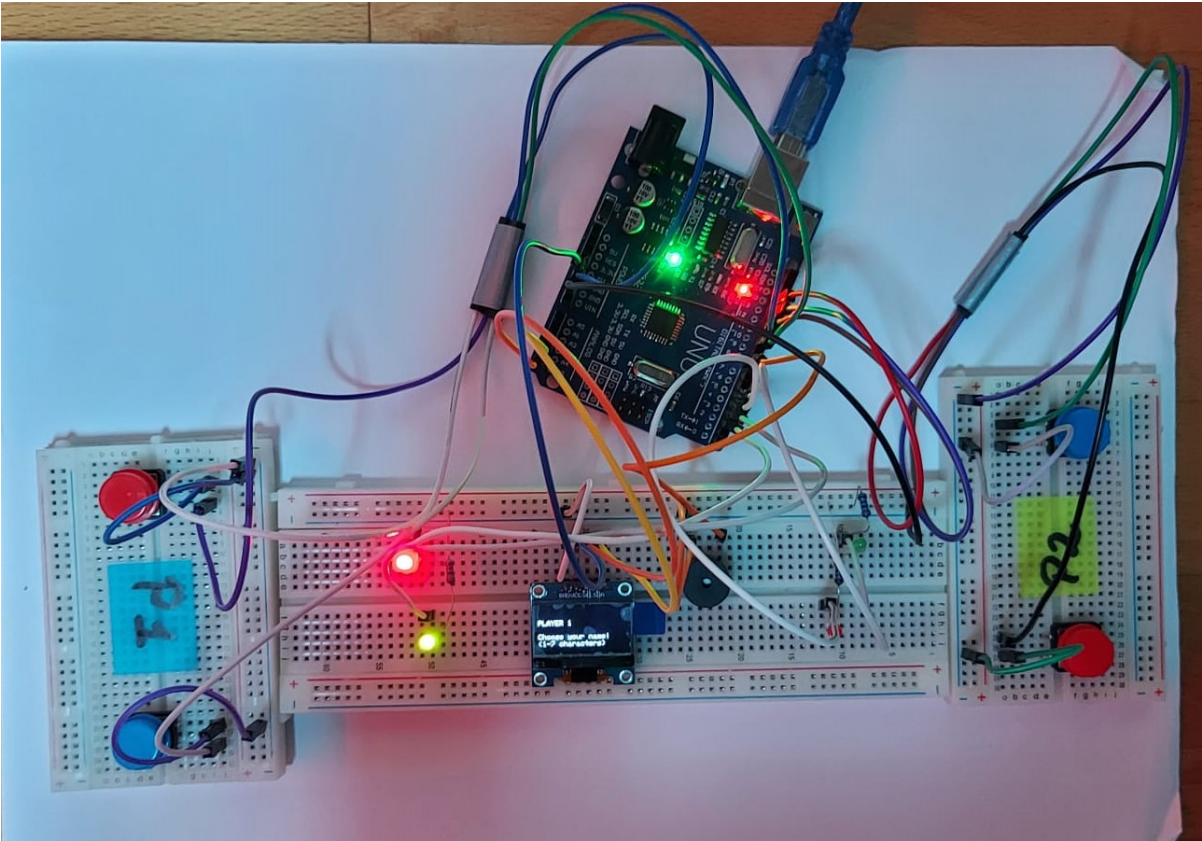
Pentru a afla cum se deplaseaza obiectele de pe display, trebuie sa aflam pozitia lor relativa fata de cea curenta pentru urmatorul "frame"; astfel, pentru deplasarea pe axa Ox avem:

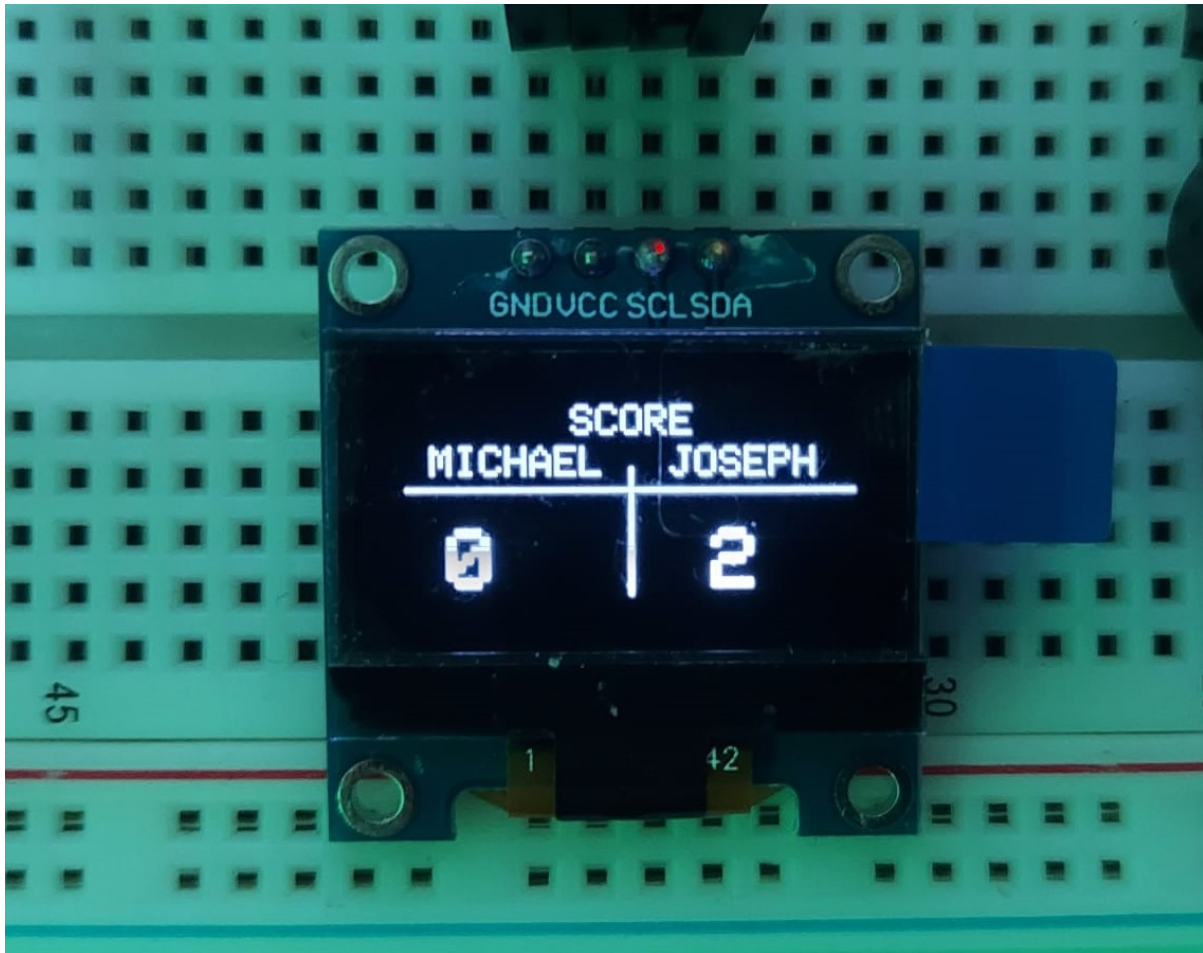
```
new_pos_x = pos_x + a * delta_x
```

```
delta_x = cos(angle) * delta * velocity
```



## Rezultate Obținute





## Concluzii

Folosind notiunile invatate la laborator, precum si datasheet-urile echipamentului hardware folosit, am reusit sa creez un joc simplu, dar care ofera toate functionalitatile pe care mi le-am propus sa le aiba.

Imbunatatiri ce ar putea fi aduse proiectului:

- folosirea unui display mai mare, care sa permita jucatorilor sa vada mai bine ceea ce se intampla
- lipirea componentelor pe placi de prototipare cablaj, nu pe un breadboard, pentru a asigura faptul ca firele si butoanele nu se vor desprinde in timp ce sunt folosite

## Download

Arhiva ce contine fisierul sursa pentru rulare.

OBS: Pentru rulara cu succes a programului, trebuie ca local sa se instaleze bibliotecile:

- **Adafruit\_GFX.h**
- **Adafruit\_SSD1306.h**

[ping\\_pong.zip](#)

## Bibliografie/Resurse

Resurse Hardware (datasheets):

- [DISPLAY - OLED 128x64 I2C/TWI](#)
- [PIEZO BUZZER - LOUDITY LD-BZEN-1205](#)
- [ATmega328p](#)

Resurse Software:

- [Laborator 0 PM - GPIO](#)
- [laborator 1 PM - USART](#)
- [Laborator 6 PM - I2C](#)
- [Biblioteca Adafruit\\_GFX](#)
- [Biblioteca Adafruit\\_SSD1306](#)
- [Introducere in Adafruit\\_GFX](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/rtilimpea/pingponggame>



Last update: **2022/05/30 20:04**