

# Mario Game

## Introducere

Prezentarea pe scurt a proiectului vostru:

- Realizarea unui joc cu un joystick
- Fiecare dintre noi am jucat ceva asemanator cand eram mici: nostalgia vibes.
- Utilitatea acestui proiect consta in oportunitatea de a invata, de a pune in practica si de a programa un arduino uno in scopul implementarii unui joc

## Descriere generală

Proiectul va fi dezvoltat pe o placuta compatibila Arduino UNO. Acesta va avea ca interfata un Modul LCD.

Utilizatorul va putea controla caracterul din joc cu ajutorul unui joystick. Acesta il va misca in sus pentru a se putea apara de blocurile aparute random in scena. Blocurile vor aparea din partea dreapta si se vor deplasa liniar spre stanga.

Pentru a fi mai interactiv vom avea un fisier in care vom putea sa vedem rezultatele unei sesiuni de joc, citite de pe un card SD.

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.



## Hardware Design

Piese necesare:

- Breadboard
- Arduino uno

- Modul Joystick
- Modul LCD
- Buzzer pasiv
- Modul cititor carduri SD

Schema hardware:



Schema Arduino:



## Software Design

Mediu de dezvoltare :Arduino IDE

Bibliotecile Arduino utilizate in dezvoltarea proiectului:

- LiquidCrystal\_I2C
- SD
- SPI
- TMRpcm

Ca sa imi stochez "ecranul" am folostii doua array-uri de char-uri in care stochez ce am:  
**terrainUpper, terrainLower.**

Mi-am creat functii pentru initializare a display-ului, pentru a-mi crea caracterul si pentru a putea avansa cu blocurile generate spre stanga.

Pentru a trimite comenzi caracterului de pe display am folosit o intrerupere pentru joystick, care sa nu interfereze cu display-ul. De asemenea am folosit protocoalele SPI pentru utilizarea cardului SD si I2C pentru utilizarea display-ului. Oparte mai grea a fost desenarea caracterului, pentru a putea sa-l afisez in joc.

Blocurile pe care le generez sunt random, atat ca dimensiune cat si ca amplasare(jos/sus).

[proiect.zip](#)

## Rezultate Obținute

Rezultatele au fost satisfacatoare. <https://www.youtube.com/watch?v=444QGZqInNs>

## Concluzii

A fost un proiect interesant care mi-a dezvoltat gândirea și mi-a indicat diverse greseli în abordarea problemelor și anumite probleme de organizare sau tehnice. Dezvoltarea codului a fost interesant și provocator, mai ales din cauza dificultăților tehnice care m-au obligat să rescriu codul, astfel ca am avut multe de învățat și de descoperit datorită contruirii și dezvoltării acestui proiect.

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

<https://docs.arduino.cc/learn/programming/sd-guide/>

[https://projecthub.arduino.cc/arduino\\_uno\\_guy/i2c-liquid-crystal-displays-5eb615](https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/fstancu/ioan.predescu>



Last update: **2024/05/27 14:18**