

# Tetris

Autor: [Verna Dorian-Alexandru](#)

Grupa: 334CC

## Introducere

### Descriere

Proiectul consta intr-un joc Tetris realizat cu ajutorul unei matrice de LED-uri. Jucatorul trebuie sa aseze piesele care apar in matrice astfel incat aceasta sa nu fie acoperita de ele. Cu cat jucatorul poate sa amane completarea cu piese a spatiului pe care il are la dispozitie, cu atat acesta este mai aproape de a castiga. Jucatorul castiga atunci cand rezista o anumita perioada de timp data de dificultatea jocului. Scorul se afiseaza pe un display, iar dificultatea (viteza, timpul) se regleaza cu ajutorul unui potentiometru. Raspata pentru castig consta intr-o surpriza data catre jucator prin intermediul unui servomotor.

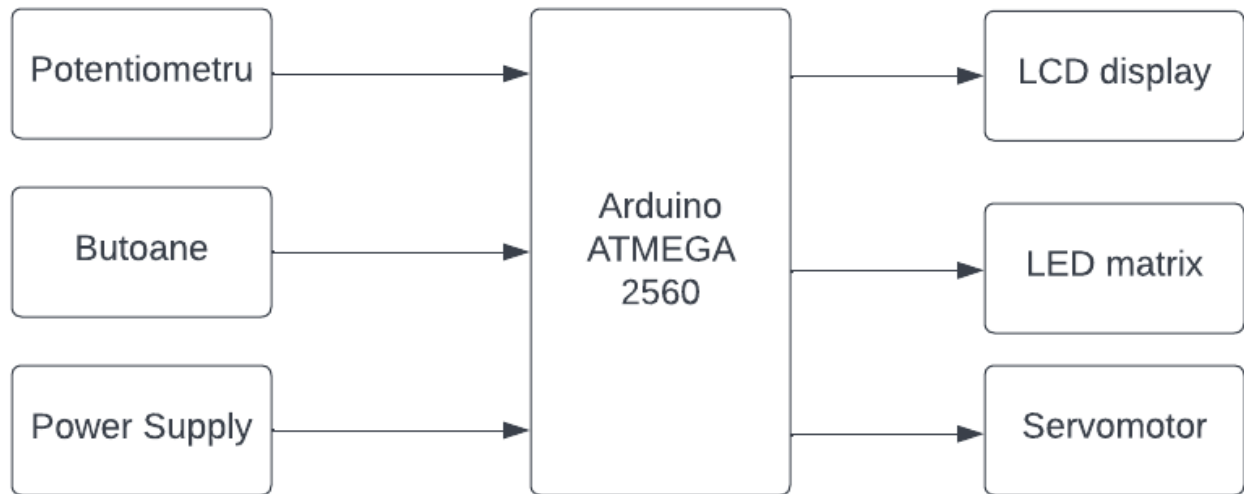
### Scop

Scopul este castigarea premiului si voia buna:)

## Descriere generală

In momentul in care se pune in functiune ansamblul, pe display va aparea un mesaj care va spune ca se poate incepe jocul odata cu apasarea unui buton. Odata inceput jocul, jucatorul trebuie sa seteze dificultatea pe care o va avea pentru joc prin intermediul potentiometrului (timpul cat trebuie sa reziste si viteza cu care coboara piesele), dupa care se poate da start la joc de pe acelasi buton ca inainte. Piesele care trebuie asezate cat mai bine pe matricea de LED-uri sunt reprezentate prin LED-uri aprinse. Jucatorul le poate roti sau misca stanga-dreapta cu ajutorul a 4 butoane. Jocul se incheie atunci cand se termina timpul setat pentru joc sau cand jucatorul a pierdut. Daca acesta a castigat, atunci, prin intermediul servomotorului, jucatorului ii va fi acordat premiul.

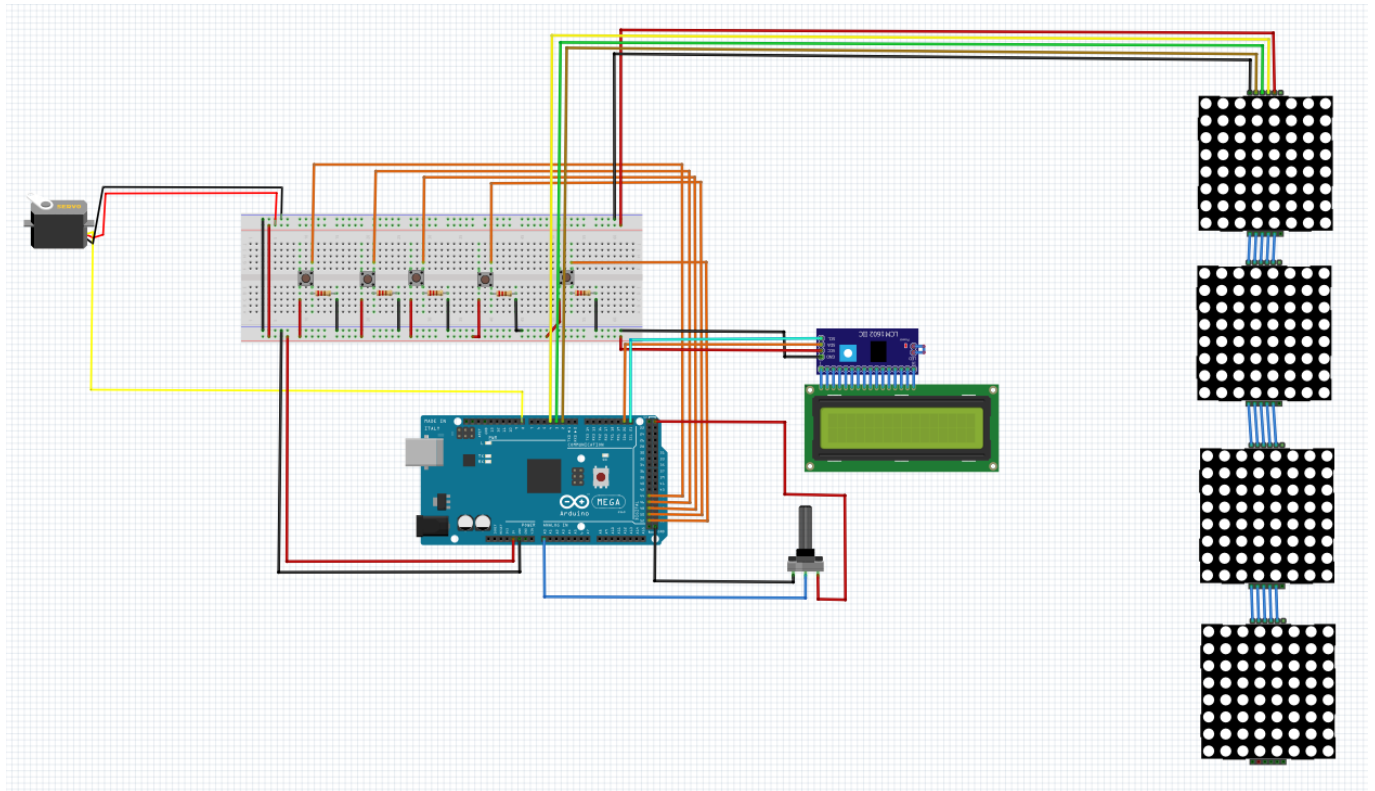
## Schema bloc



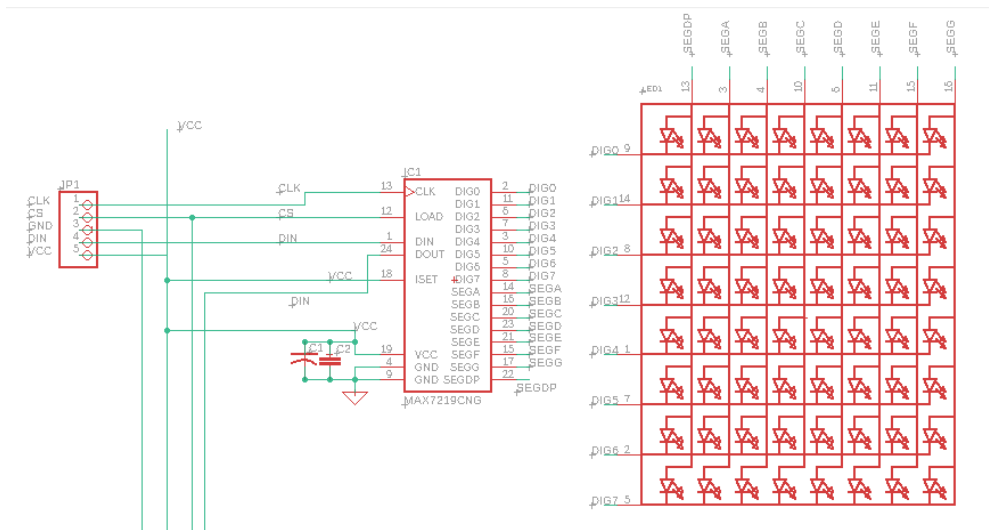
## Hardware Design

### Componente

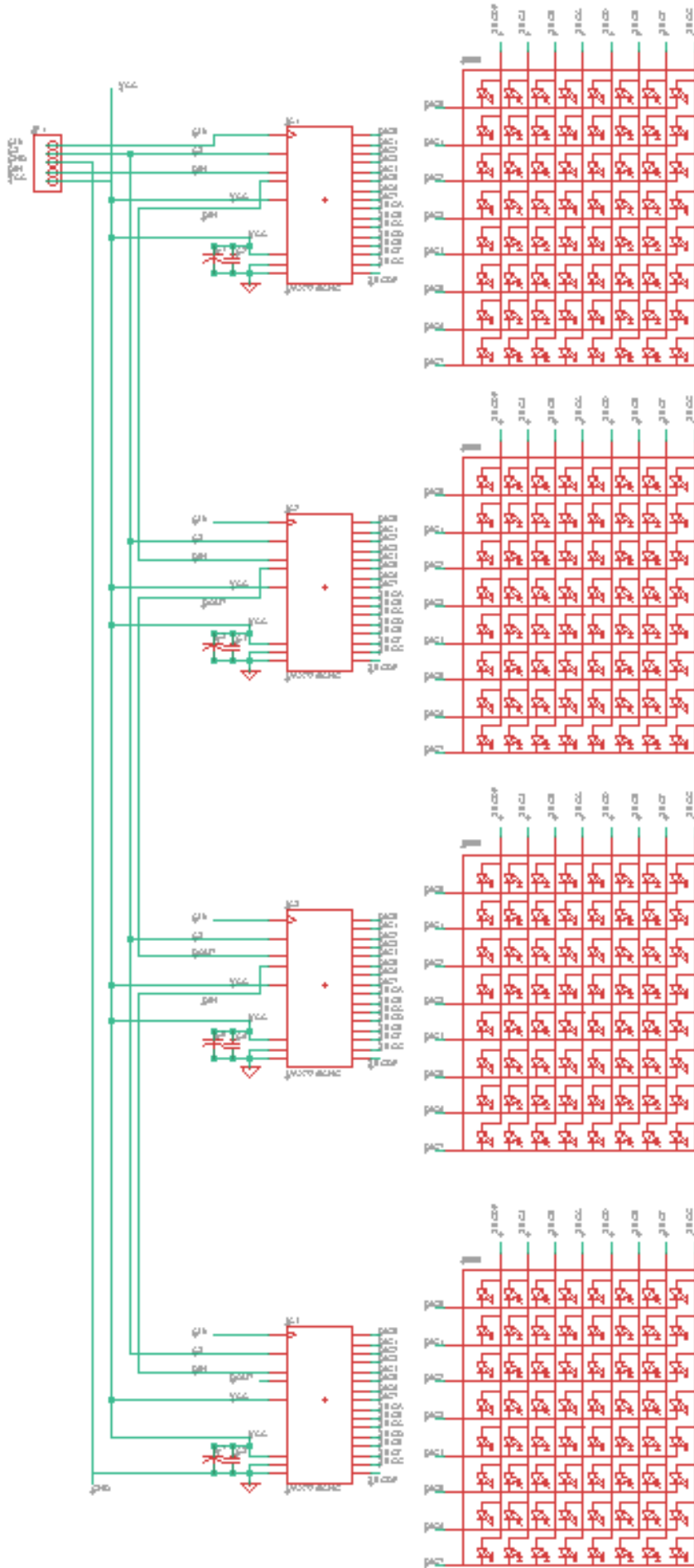
- Arduino ATMEGA 2560
- Breadboard
- 5 X Buton
- Modul LCD 1602
- 4 X Matrice LED 1088AS 8×8
- Potentiometru
- Rezistori
- Fire
- Servomotor



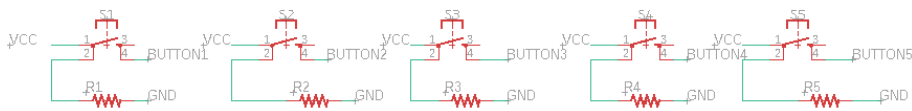
### Schema pentru o matrice



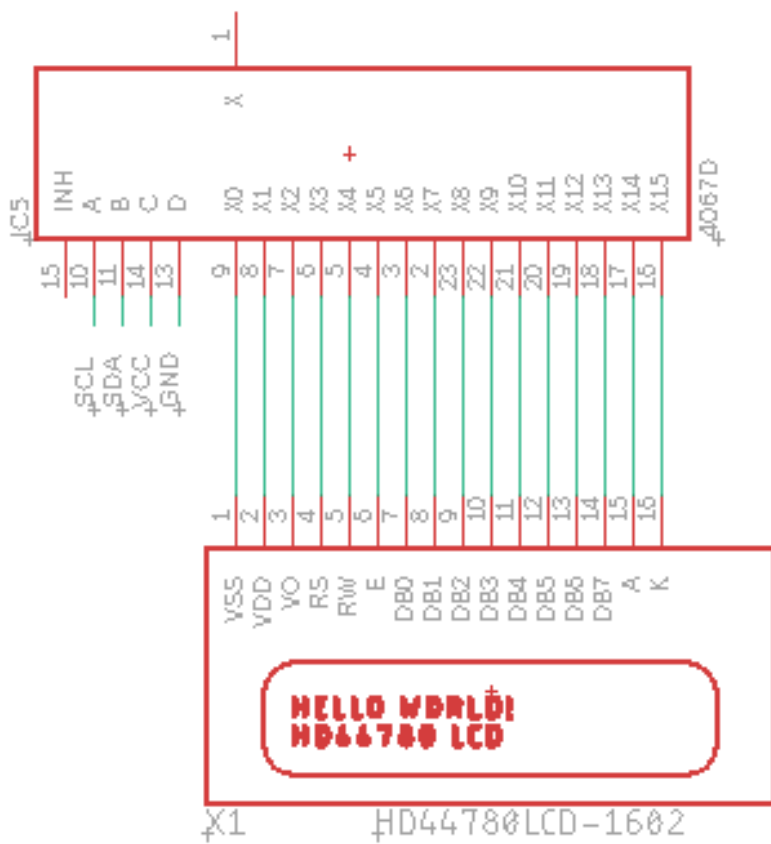
### Schema pentru tot ansamblul de 4 matrice



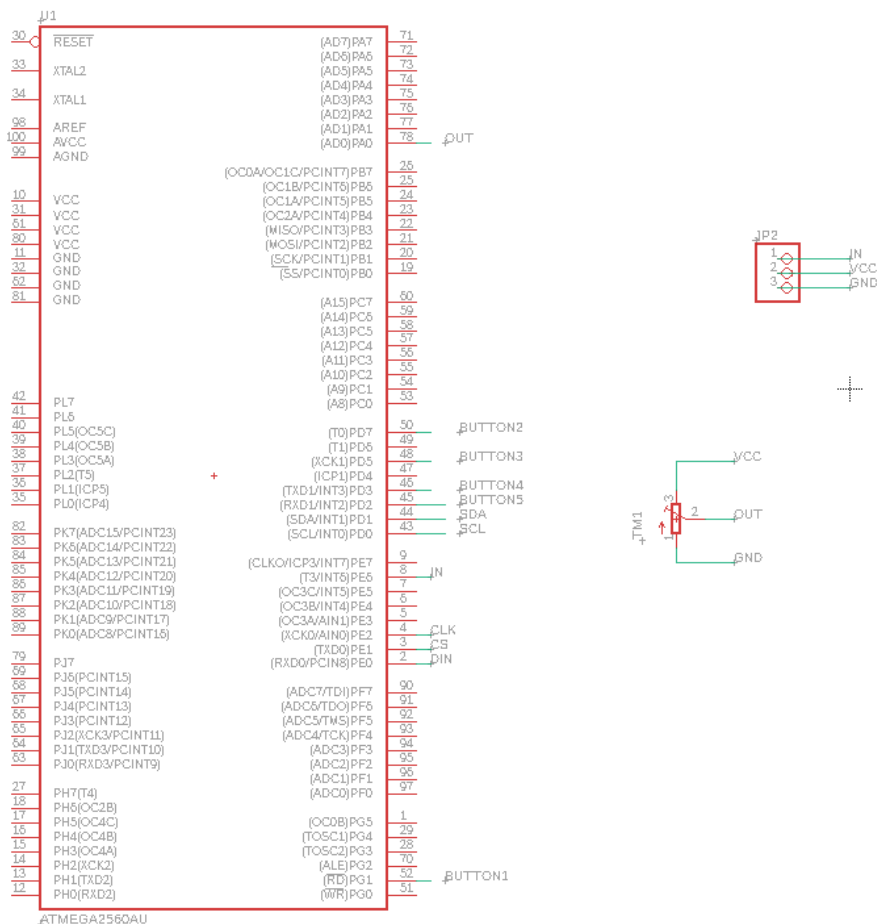
### Schema pentru butoane



### Schema pentru display



### Schema pentru microcontroller, potentiometru si servomotor



## Software Design

- Pentru dezvoltarea aplicatiei am folosit Arduino IDE
- Bibliotecile pe care le-am folosit sunt: MD\_MAX72xx.h si SPI.h pentru matricele de LED-uri, LiquidCrystal\_I2C.h si Wire.h pentru LCD
- Am folosit o matrice de 8x32 pentru a reprezenta matricele de LED-uri. Am folosit o structura de date pentru a retine pozitiile din matrice unde se regaseste o piesa care se afla in miscare.

## Detalii Implementare

Am inceput proiectul cu pasi mici, primul a fost generarea random a pieselor. Dupa acest pas, am inceput sa implementez modul in care coboara acestea pe matricea de LED-uri, urmand sa se opreasca atunci cand ajung la baza matricei. Am folosit o structura (moving\_piece) pentru a retine pozitiile pe care le detine o piesa in miscare pe matricea mare de LED-uri (practic, pentru a face piesa sa coboare pe matrice, incrementez coloanele aferente campurilor din moving\_piece). Pe matricea mare care simuleaza matricea de LED-uri, 0 inseamna spatiu liber, 1 inseamna piesa care s-a oprit, iar 2 inseamna piesa care inca se misca. Matricea aceasta este afisata tot timpul in loop pe matricea fizica. Pe matricea fizica pot sa setez doar cate un rand intreg pe fiecare din cele 4 matrice care intra in componenta sa.

Urmatorul pas in implementare consta in implementarea utilitatii butoanelor. Am 5 butoane in total, dintre care 4 le folosesc pentru miscarea pieselor. Primele 2 butoane de pe breadboard, cele pentru miscarea stanga-dreapta a piesei, apeleaza, atunci cand sunt apasate, functiile aferente care verifica daca se poate realiza miscarea stanga-dreapta a piesei, iar in caz afirmativ, decrementez si incrementez randul elementelor din `moving_piece`.

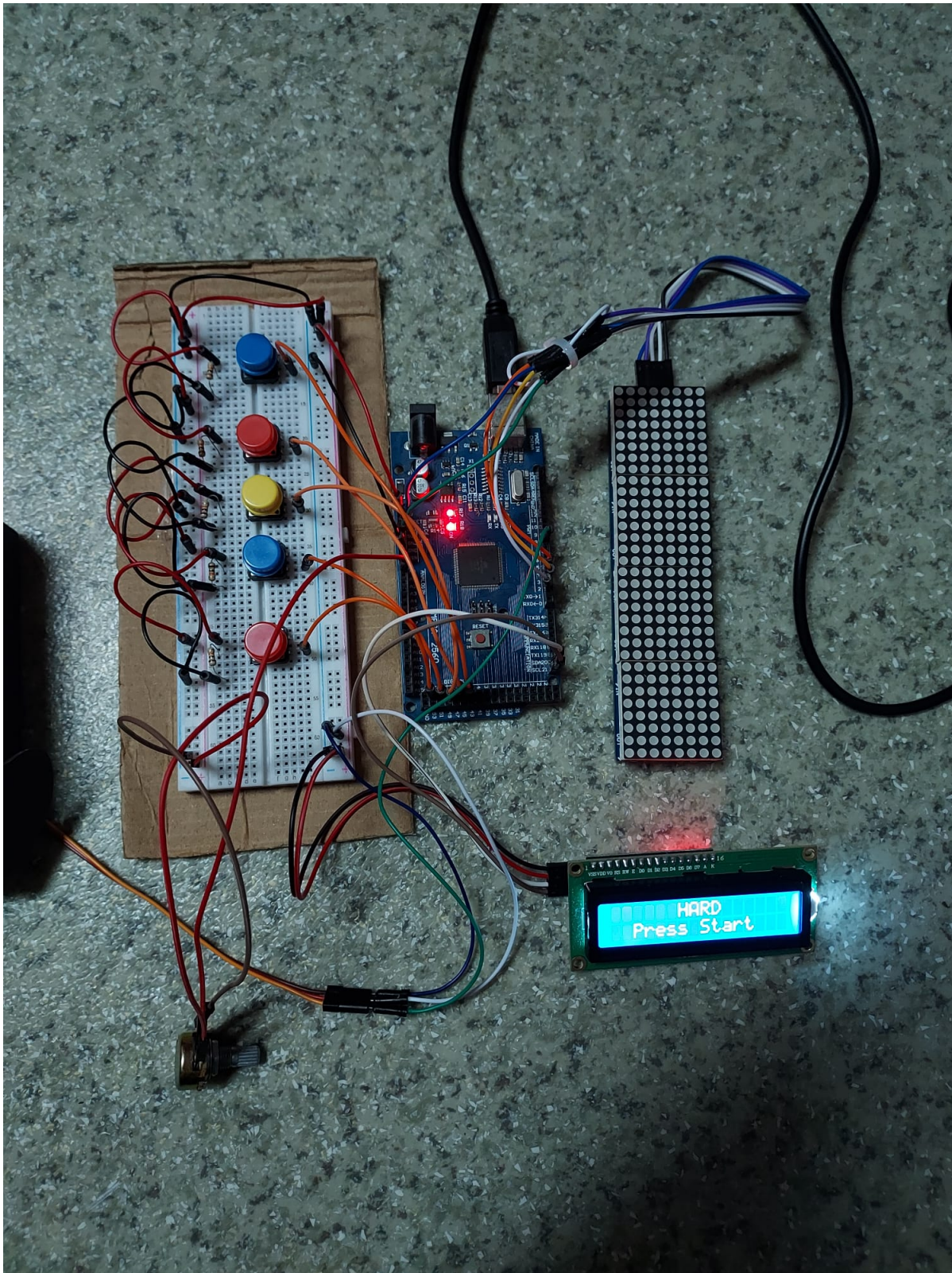
Dupa aceea, a trebuit a implementez rotirea pieselor pentru urmatoarele butoane (rotire stanga, respectiv rotire dreapta). Ca dificultate, aceasta a fost cea mai grea parte din tema. Practic, aici folosesc o matrice auxiliara unde efectuez rotirea, iar daca aceasta este posibila, atunci translatez pozitiile care vor fi ocupate dupa rotire de piesa in matricea originala. Apoi am implementat stergerea unor linii si shiftarea in jos a liniilor de deasupra.

Dupa aceea, am conectat display-ul LCD si am facut posibila afisarea unor mesaje precum cel care afiseaza scorul pe ecran, urmand ca, mai apoi, sa introduc si potentiometrul care sa faca posibila alegerea dificultatii (numar de linii care trebuie sterse si viteza piese).

La sfarsit de tot am introdus si servomotorul pe care l-am pus intr-o cutie. In momentul in care se castiga jocul, servomotorul deschide cutia si jucatorul isi poate lua premiul.

## Rezultate Obținute

Proiectul arata in felul urmator:



Mai jos este un demo:

- <https://drive.google.com/file/d/1kuXiZmQBcJujScksjTUPsMBVEaROCct0/view?usp=sharing>

## Concluzii

Am intalnit o serie de concepte noi in timpul realizarii proiectului. Am invatat cum sa conectez cele 4 matrice la microcontroller si sa realizez operatii cu servomotorul si display-ul. Mare parte din proiect a constat in programarea jocului, acolo am intampinat si cele mai mari dificultati, dar per total consider ca a fost o experienta reusita:)

## Download

Mai jos este link-ul catre repo-ul meu de github unde am realizat implementarea proiectului:

- <https://github.com/dorianverna17/Tetris>

## Jurnal

- 12.05 - alegere tema
- 15.05 - 16.05 - cumparare piese
- 15.05 - realizare miscare piese pe matrice
- 17.05 - completare joc
- 18.05 - conectare display, potentiometru
- 19.05 - realizare mecanism servomotor si legare la microcontroller

## Bibliografie/Resurse

- <https://ocw.cs.pub.ro/courses/pm>
- <https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24>
- <https://docs.arduino.cc/learn/electronics/servo-motors>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/dene/tetris>



Last update: **2022/05/27 21:49**