

# GravitySlide

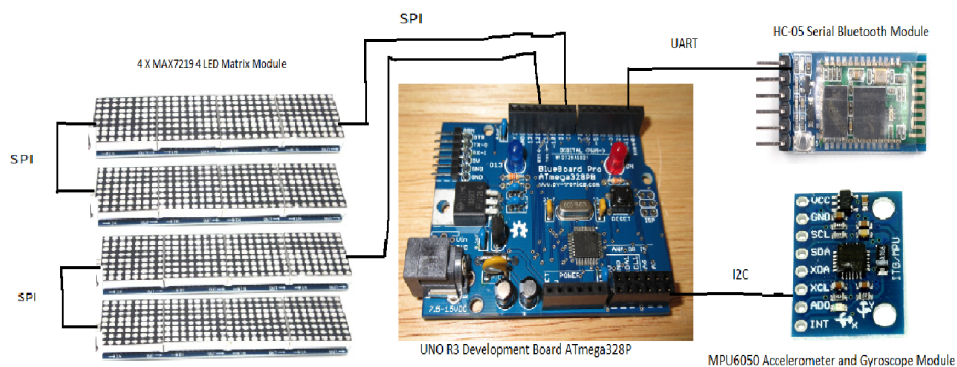
Proiectul presupune realizarea unui display din matrici led 32×32 prin intermediul caruia vom oferi utilizatorului un mod interactiva de a interactiona cu obiectele ce sunt afisate pe ecran influentate de unghiul de inclinare. Obiectele/ formele prezentate pot fi alese de utilizator prin intermediul unui aplicatii mobile. Jocul poate fi un mijloc de relaxare atat pentru adulti cat si pentru cei mici.

## Descriere generală

Design-ul proiectului se bazeaza pe interactiunea dintre placuta principala Arduino UNO ce se conecteaza cu 2 vectori de 2 matrici ce vor insuma un patrat cu latura de 32 de leduri ( $2^5$  leds) conectate initial in serie 2 cate 2, dar care vor fi tratate ca un grid cu diviziunea de 1 led. Un senzor cu modul de acceleratie si giroscop, ce ne va oferi constant date prin care vom afla unghiul de inclinare pe  $O_x$  si  $O_y$ , dupa care vom simula caderea.



Modulul bluetooth se va folosi pentru a primi comenzi de la user printr-o aplicatie pentru a varia elementele afisate.

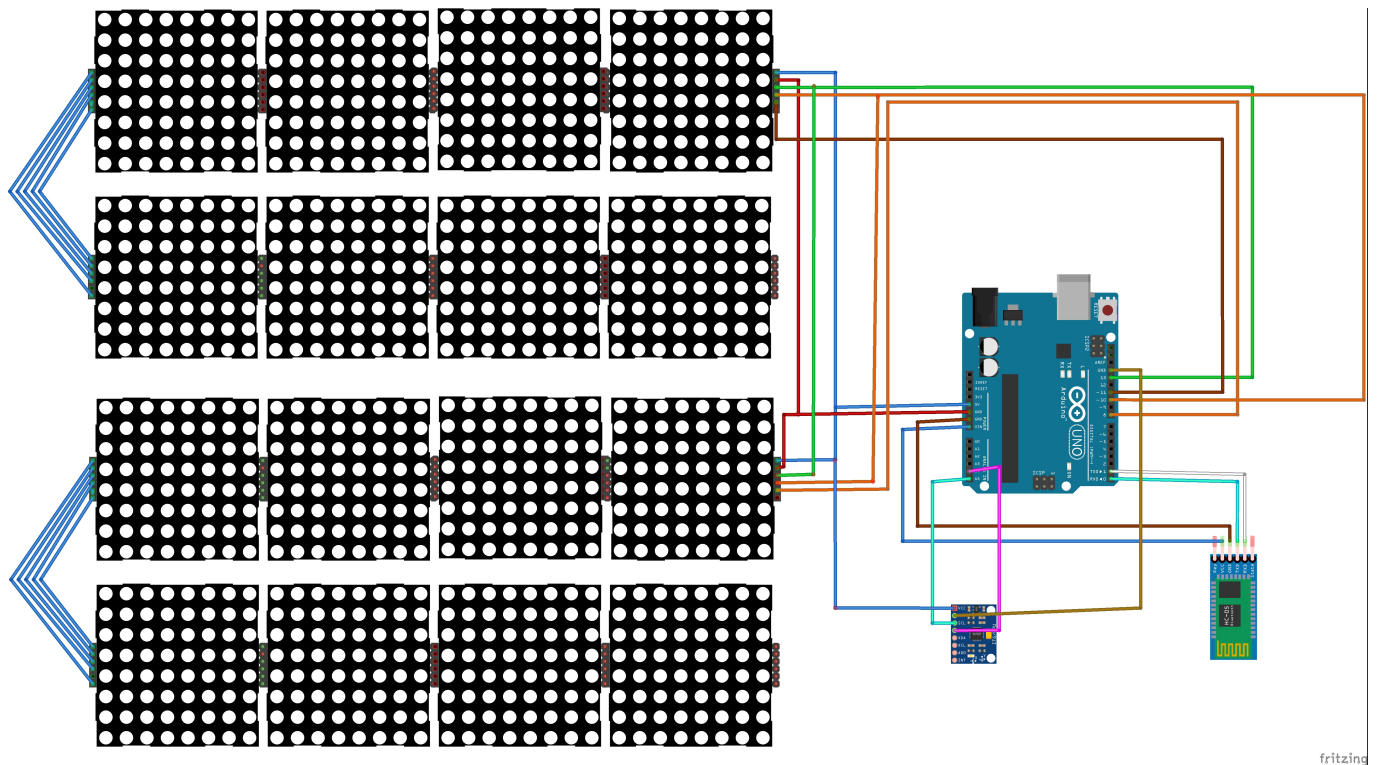


## Hardware Design

Elemente hardware folosite:

- Arduino UNO R3 CH340GO - aici se-ntampla magia

- MAX7219 LED Dot Matrix Module (x4) - va simula un ecran
- Modul bluetooth HC-05 realizeaza comunicarea intre placuta si utilizator
- MPU6050 Accelerometru si giroscop - cu ajutorul datelor vom calcula unghiurile



In practica matricele sunt lipite patru cate patru, rezultand 4 vectori de leduri paraleli.

## Software Design

### Medii de dezvoltare

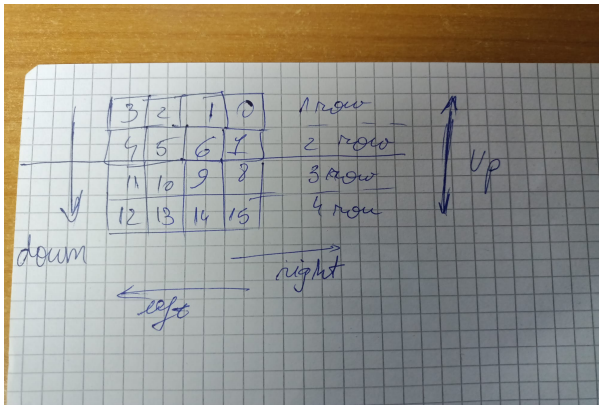
- VSCode si Arduino IDE

### Blioteci

- Wire - giroscop si accelerometru
- LedControl - pentru setul de matrici

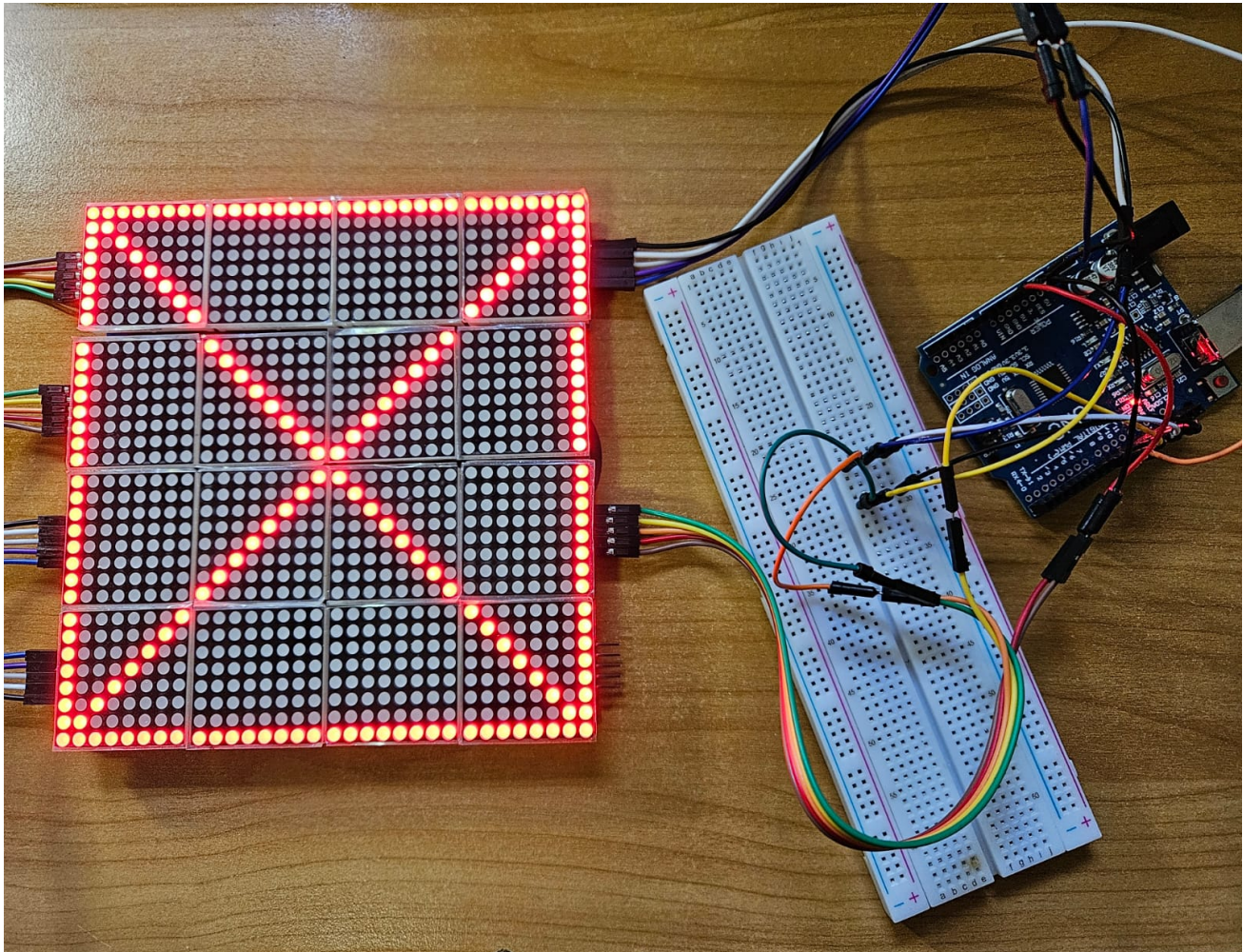
**Cod** Pentru a intelege ce a dus la forma software ului, trebuie prezentate cateva probleme aparute:

- biblioteca producatorului, ([https://github.com/MajicDesigns/MD\\_MAX72XX](https://github.com/MajicDesigns/MD_MAX72XX)), nu permite conectarea a mai mult de 8 matrici bloc in serie, eu avand 16, motiv pentru care a trebuit sa calculez de mana indicii fiecarei bucle si afisari. Matricele trebuie considerate paralel desi nu sunt, folosind doi pini pentru CS\_PIN.
- Pinii de Vcc si Ground de la matricii nu pot fi utilizati, intrucat nu mai dau acelasi voltaj la iesire. :(



Codul a fost impartit in functii pentru modularizare, apelul lor bazanduse pe cele doua apeluri de setup si loop. Formele generate sunt pe un set de leduri de 8x8. Aceasta structura este un bitfield de dimensiune de 1 byte in care am retinut ce led trebuie aprins si stins.

Programul este initializat cu un X cu un contor al matricei, dupa care i-a nastere minijocul nostru. Fiecare led isi schimba intensitatea intre 0 si 16.



## Funcții

- `init_TO_DO` - initializeaza imaginea pe diplay, seteaza intensitatea luminoasa la un interval de 0.5s, initializeaza conexiunea bluetooth si asteapta date pe seriala, initializeaza giroscopul-accelerometrul, prindeaza forma pe display-ul din pozitia curenta.
- functii de calcul pentru indexi si starea ledurilor, precum si nr mini display-ul pe care trebuie incarcate.
- blocuri de cod timer si preluarea si prelucrarea mesajelor prinmrite prin bluetooth (se utilizeaza o

aplicatie ca interfata -

<https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor&hl=en> )

Documentatia pentru cele doua librari utilizate poate fi gasita pe site-ul oficial, cu o multime de exemple, modele grafice, imagini, cod sursa, API-uri si multe alte informatii pentru o mai buna intelegere a modului de lucru si a potentialului pe care il are proiectul.

Protocoalele de comunicatie utilizate: I2C, SPI, UART. Pentru o buna intelegere, utilizati laboratoarele de PM.

### Calcul unghiuri de inclinatie

- $AccX = Wire.read() \ll 8 | Wire.read();$
- $AccY \dots, AccZ \dots;$
- $AccX = (float)AccX / 4096; \dots$
- $AngleRoll = atan(AccY / \sqrt{AccX * AccX + AccZ * AccZ}) * 1 / (3.14159265 / 180);$
- $AnglePitch = -atan(AccX / \sqrt{AccY * AccY + AccZ * AccZ}) * 1 / (3.14159265 / 180);$

Valorile se afla intre -90 si +90 de grade inclinatie. Pe baza acestora si a unui thrashhold de 15-30, are loc deplasarea elementelor.

## Rezultate Obținute

**Demo** [https://drive.google.com/file/d/1bflazC9i1ql\\_rWyKbnhYPqLdTY1luT5F/view?usp=sharing](https://drive.google.com/file/d/1bflazC9i1ql_rWyKbnhYPqLdTY1luT5F/view?usp=sharing)

## Concluzii

Lipsa unei biblioteci, mareste substantial volumul de munca al unui programator, dar si imposibilitatea modificarii uneia este un proces tot la fel de anevoios. Proiectul poate si merita imbunatatiri multiple, motiv pentru care vreau continui cu el. Daca mi-a placut ce am facut ? Cu siguranta, am dobandit cunostinte foarte utile, am realizat si un proiect, ce PRINDE VIATA :). Este foarte util sa lucrezi si sa intelegi ce presupune un hardware, limitat, o memorie mica, mult prea mica ( inial am si umplut-o in primele zile - uhh... ). Codul a trebuit sa suferie multe imbunatatiri si sa respecte conditii peste conditii si ma bucur ca le-am avut, te fac sa te adaptezi si sa cauti solutii.

## Download

### Check the repo

- [gravityslide.zip](#)

## Bibliografie/Resurse

Informatiile sunt multiple, dar vreau sa impartasesc cu voi cateva ce le-am gasit eu ca fiind mai esentiale.

- [https://www.youtube.com/watch?v=7VW\\_XVbtu9k&t=697s](https://www.youtube.com/watch?v=7VW_XVbtu9k&t=697s)
- <https://mschoeffler.com/2017/10/05/tutorial-how-to-use-the-gy-521-module-mpu-6050-breakout-board-with-the-arduino-uno/>
- <https://lastminuteengineers.com/max7219-dot-matrix-arduino-tutorial/>
- <https://atadiat.com/en/e-towards-understanding-imu-basics-of-accelerometer-and-gyroscope-sensors/>
- <https://w3.braude.ac.il/wp-content/uploads/2021/03/HC-051.pdf>
- <https://www.arduino.cc/reference/en/language/functions/communication/wire/>

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/mihai\\_brezniceanu](http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/mihai_brezniceanu)



Last update: **2024/05/27 07:20**