

Tetris

- Student: Giuglan Denisa-Cristina
- Grupa: 334CD

Introducere

- Proiectul meu aduce jocul Tetris din lumea virtuala intr-un format fizic, construit cu ajutorul unei placi Arduino. Ideea de baza este simpla: piese de diferite forme cad pe o matrice verticala de LED-uri, iar jucatorul trebuie sa le potriveasca perfect pentru a crea linii orizontale. Atunci cand o linie este completa, ea dispare, lasand loc pentru noi piese.
- Am pornit de la dorinta de a vedea cum pot transforma niste componente electronice obisnuite intr-o consola de jocuri functionala. Mi-am dorit sa vad daca pot sa fac toate aceste piese sa se miste fluid, sa afisez scorul in timp real si sa adaug muzica de fundal, totul dintr-un singur cod.
- Pentru mine, acest proiect este cea mai buna metoda de a vedea cum teoria de la cursuri se aplica in realitate. Am invatat cum sa conectez corect senzorii (joystick, buton) si cum sa organizez codul software pentru a controla hardware-ul.

Descriere generală

- **Control (Input):** Jucatorul foloseste un joystick pentru a muta piesele la stanga, la dreapta sau in jos si un buton separat pentru a le roti. Aceste miscari sunt trimise ca semnale catre placa Arduino.
- **Procesare (Logica):** Placa Arduino Uno este "creierul" care ruleaza jocul. Ea decide cat de repede cad piesele, verifica daca s-a format o linie si calculeaza scorul.
- **Afisare (Output):** Jocul se vede pe un modul format din 4 matrice de LED-uri (8x32 puncte in total), iar informatiile suplimentare, precum scorul, sunt afisate separat pe un ecran OLED.
- **Sunet (Feedback):** Un buzzer reda muzica de fundal si scoate sunete scurte atunci cand jucatorul castiga puncte sau pierde jocul.



Hardware Design

Componente folosite:

- **Arduino Uno:** Placa de baza pentru controlul intregului sistem.
- **Modul 4 matrice LED MAX7219 (SPI):** Suprafata principala de joc (8x32 pixeli).
- **Display OLED (I2C):** Pentru afisarea scorului si a mesajelor.
- **Joystick:** Pentru controlul directiei de deplasare a pieselor.
- **Buton (Push-button):** Pentru functia de rotire a pieselor.
- **Buzzer (PWM):** Pentru feedback sonor si melodia Tetris.
- **Rezistenta 220 Ω:** Pentru protectia circuitului buzzer-ului.
- **Breadboard si fire de legatura:** Pentru realizarea conexiunilor intre componente.



!Display-ul de ceas din imaginea atasata mai sus reprezinta matricea de LED-uri, iar potentiometrul reprezinta joystick-ul!

In tabelul de mai jos este prezentata configuratia pinilor utilizati, alaturi de justificarea tehnica a alegerii acestora:

Componenta	Pin Componenta	Pin Arduino	Justificare Tehnica
OLED	VCC	5V	Alimentare modul ecran.
	GND	GND	Masa comuna a circuitului.
	SDA	A4	Serial Data pentru I2C. Pin dedicat hardware pe Arduino Uno.
	SCL	A5	Serial Clock pentru I2C. Sincronizeaza transferul de date.
Matrice LED	VCC	5V	Alimentare module MAX7219.
	GND	GND	Masa comuna.
	DIN	D11	MOSI (SPI). Transfer rapid de date catre cele 256 de LED-uri.
	CS / LOAD	D10	Chip Select. Activeaza scrierea pe modulele LED.
	CLK	D13	SCK (SPI Clock). Sincronizeaza bitii in protocolul SPI.
Joystick	VCC	5V	Alimentare pentru potentiometrele interne.
	GND	GND	Masa comuna.
	VRX	A0	Intrare Analogica. Citeste axa X pentru deplasare stanga/dreapta.
	VRX	A1	Intrare Analogica. Citeste axa Y pentru cadere rapida.
	SW	D4	Intrare Digitala pentru butonul intern al joystick-ului.
Push Button	P1	D2	Intrare Digitala. Suporta intreruperi hardware (Interrupts).

	P2	5V	Sursa de semnal pentru logica de tip Active High.
Buzzer	(+)	D3	Pin PWM. Genereaza frecventele audio.
	(-)	GND	Masa comuna.

Software Design

IDE folosit

Am folosit **Arduino IDE** pentru scrierea, compilarea si incarcarea firmware-ului direct pe placa de dezvoltare.

Biblioteci utilizate

Urmatoarele biblioteci 3rd-party si native au fost incluse pentru a asigura controlul perifericelor:

- **Wire.h** — Biblioteca nativa din Arduino, utilizata pentru gestionarea protocolului de comunicare I2C cu ecranul OLED.
- **Adafruit_GFX.h** — Biblioteca de baza Adafruit pentru functiile grafice (afisare text, linii si forme geometrice).
- **Adafruit_SSD1306.h** — Driver software dedicat controlului si maparii pixelilor pe ecranul OLED de 128×64.
- **LedControl.h** — Biblioteca utilizata pentru controlul celor 4 module de matrice LED (MAX7219) legate in cascada prin interfata SPI hardware.

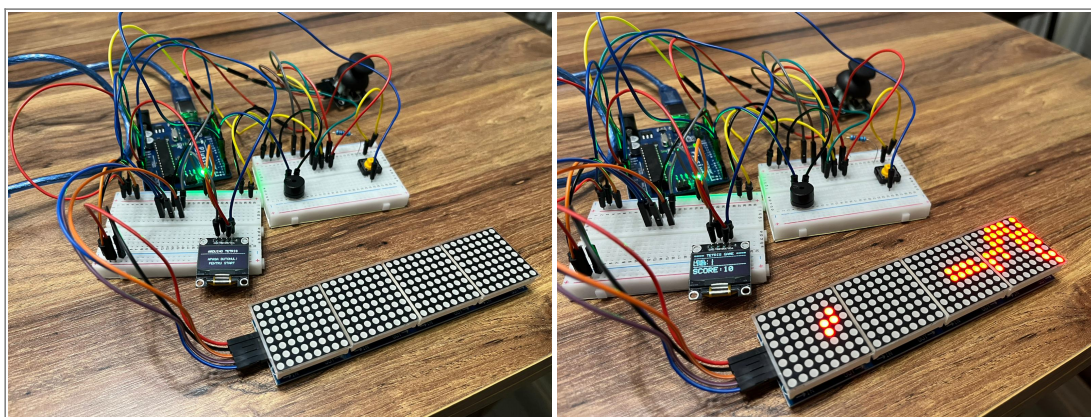
Structuri si algoritmi implementati

- **Salvarea starii pe biti (Bit-mapping):** Pentru optimizarea memoriei RAM limitate, tabla de joc este stocata intr-un vector compact de tip `uint8_t board[32]`. Fiecare bit dintr-un byte reprezinta starea unui pixel (0 = stins, 1 = aprins) de pe latimea de 8 coloane a jocului.
- **Logica non-blocking (Asincrona):** Functia audio si cea de control al timpului folosesc functia `millis()` in loc de `delay()`. Acest lucru permite rulara concomitenta a melodiei prin buzzer, citirea joystick-ului si caderea automata a pieselor fara blocarea microcontrolerului.

Surse si functii implementate

Nume Functie	Descriere Functionalitate
<code>void setup()</code>	Initializeaza ecranul OLED si matricele LED, configureaza pinii pentru joystick/buton si porneste meniul principal.
<code>void loop()</code>	Gestioneaza starile jocului, apeleaza functia de muzica, citeste intrarile si controleaza automat caderea pieselor.
<code>void resetGame()</code>	Goleste tabla de joc (reseteaza vectorul de biti la 0), aduce scorul si nivelul la 1 si genereaza prima piesa.
<code>void drawMenu()</code>	Afiseaza ecranul de start pe display-ul OLED, asteptand apasarea butonului pentru a incepe jocul.
<code>void updateOLED()</code>	Actualizeaza in timp real datele despre scor, nivelul curent de dificultate si numarul total de linii sterse pe ecranul OLED.
<code>void spawnPiece()</code>	Alege aleatoriu una dintre cele 7 forme de piese, o plaseaza in partea de sus si verifica conditia de Game Over.
<code>void handleInput()</code>	Citeste axele joystick-ului pentru miscari si verifica starea butonului pentru a roti piesa activa (transpunere matrice 3x3).
<code>bool canMove(...)</code>	Verifica daca o piesa se poate muta sau roti in siguranta fara sa iasa din ecran sau sa loveasca piese deja fixate.
<code>void lockPiece()</code>	Copiază piesa activa in vectorul permanent al tablei de joc (board) prin aplicarea unei masti pe biti.
<code>void checkLines()</code>	Scaneaza tabla pentru a gasi linii complete (byte egal cu 0xFF). Daca gaseste, le sterge, coboara liniile de deasupra si creste scorul.
<code>void manageMusic()</code>	Citeste vectorul cu notele muzicale ale temei Tetris si le reda consecutiv prin buzzer, calculand pauzele in mod asincron.
<code>void render()</code>	Preia buffer-ul tablei de joc, suprapune piesa activa si trimite datele prin SPI catre cele 4 drivere MAX7219.

Rezultate Obținute



Demo joc: https://youtu.be/ZB1cEshHryI?si=_HgvfmvWA-34vG1v

Concluzii

Rezultate obtinute:

- **Sistem de control precis:** Utilizarea joystick-ului analogic si a unui buton extern ofera o experienta de joc fluida si intuitiva, permitand miscarea rapida si rotirea exacta a pieselor in timp real.
- **Afisare duala eficienta:** Integrarea matricelor LED 8×32 ofera o suprafata de joc clara si luminoasa, in timp ce ecranul secundar OLED afiseaza in mod constant statisticile esentiale (scorul, nivelul si liniile sterse), imbunatatind experienta utilizatorului.
- **Multitasking software (Non-blocking):** Prin eliminarea functiilor de tip `delay()` si utilizarea exclusiva a contorizarii timpului via `millis()`, firmware-ul ruleaza asincron. Consola reuseste sa redea melodia de fundal prin buzzer, sa scaneze intrarile si sa gestioneze caderea pieselor in mod simultan, fara intreruperi.

Concluzii personale:

- Proiectul a reprezentat o oportunitate excelenta de a combina cunostintele de electronica digitala cu cele de programare software pe microcontrolere. Cele mai mari provocari au fost optimizarea memoriei RAM (prin maparea tablei de joc direct pe masti de biti) si implementarea algoritmului de transpunere si rotire a matricelor software pentru a se potrivi cu orientarea fizica a hardware-ului MAX7219.
- In final, toate obiectivele propuse in faza de design au fost atinse cu succes, rezultand un dispozitiv hardware stabil si un joc complet functional.

Download


Arhiva contine codul Arduino al proiectului: [tetris.zip](#)

Bibliografie/Resurse

- <https://www.optimusdigital.ro/>
- <https://www.tinkercad.com/>

[Export to PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2026/theodor_ioan.buliga/denisa.giuglan 

Last update: **2026/05/27 00:53**