

Tetris

Autor: [TOADER Sergiu-Cristian](#)

Introducere

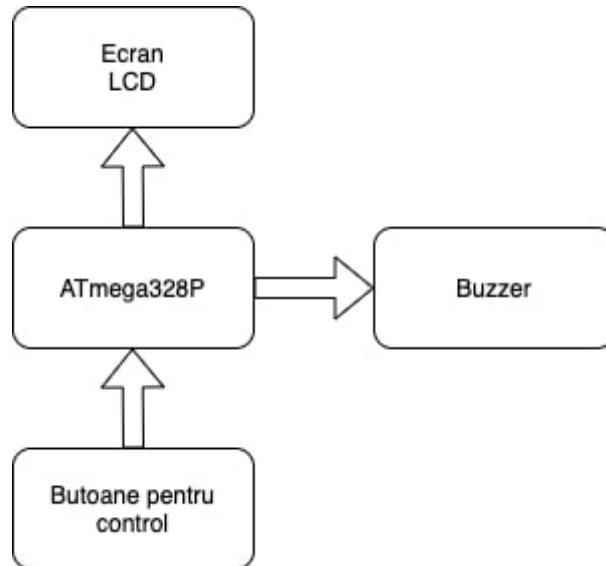
- Proiectul va consta in implementarea jocului clasic de Tetris pe un ecran LCD SPI.
- Jocul consta in pozitionarea pieselor in asemenea mod incat spatiile libere sa fie umplute si sa se genereze astfel linii complete. Liniile complete sunt automat eliminate si se acorda puncte bonus. Pieseile apar din partea de sus a ecranului si coboara pana la contactul cu podeaua sau cu alte piese deja asezate.
- Jocul se incheie in momentul in care piesele asezate au ajuns in partea de sus a zonei de joc, iar alte piese nu mai au loc pentru a fi generate.
- Scopul jocului este de a acumula cat mai multe puncte si de a supravietui cat mai mult timp.

Descriere generala

- Ecranul va fi impartit in doua portiuni: Portiunea in care trebuie pozitionate piesele si portiunea in care este afisat scorul si in care apare urmatoarea piesa generata.
- Jucatorul poate folosi butoanele pentru a deplasa stanga-dreapta piesele si pentru a le roti.
- Exista si un buton de pauza / de incepere a unui joc nou.
- Jocul are si o componenta de sunet. Se vor produce sunete specifice pentru fiecare actiune din joc.

(Rotatii, mutari ale pieselor, completare de linii de patrute, incheiere joc).

Schema bloc



Arduino

- coordonatorul aplicatiei
- primeste semnalele de input de la butoane
- transmite semnale de output catre LCD si buzzer

Buzzer

- produce semnale audio / cantece in diverse stadii ale jocului

Butoane

- reprezinta modul prin care utilizatorul interactioneaza cu aplicatia

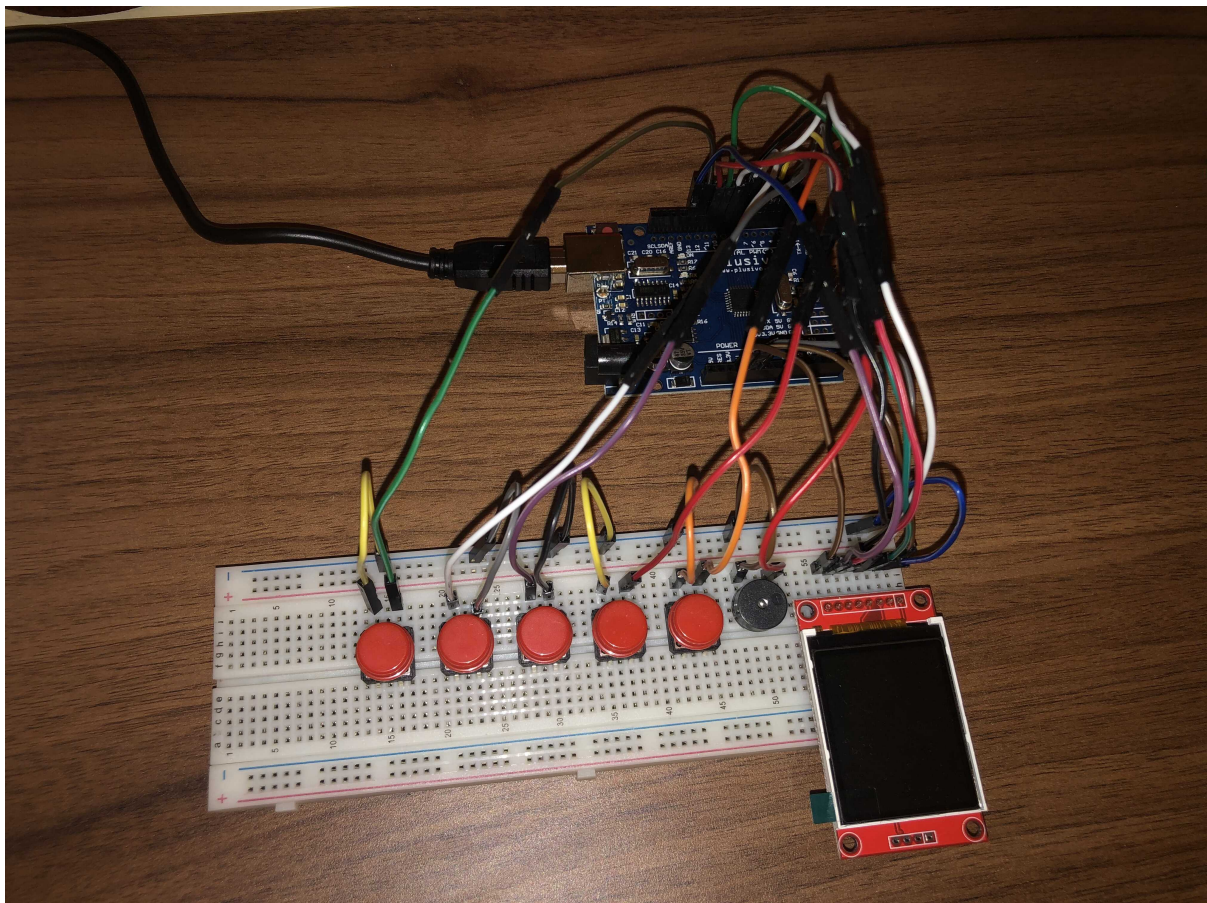
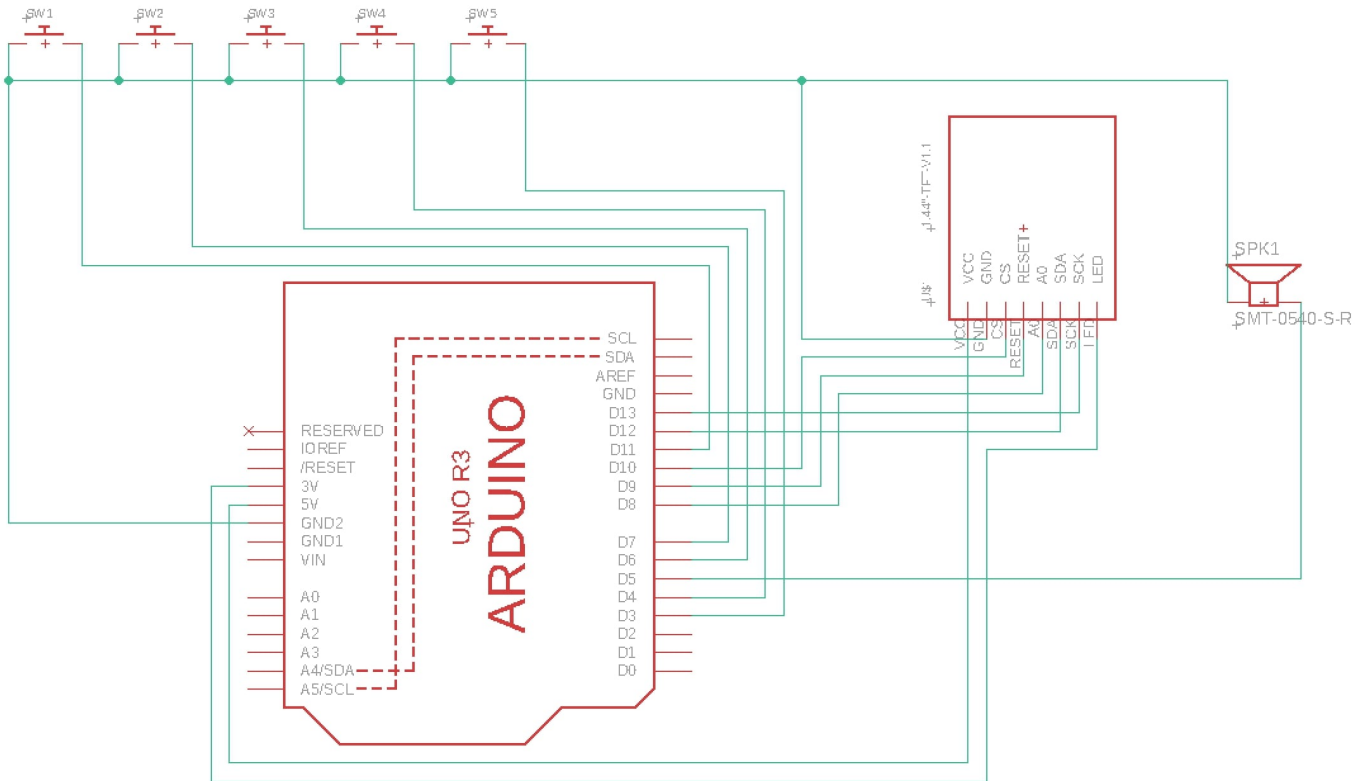
LCD

- reprezinta compomenta vizuala a aplicatiei
- afiseaza toate ecranele jocului

Componente

Nume	Numar
Arduino Uno	1
Breadboard	1
Modul LCD SPI 128×160	1
Push button	5
Fire mama-tata	10
Fire tata-tata	18
Buzzer	1

Hardware Design



Conectarea LCD-ului a presupus conectarea celor 8 pini ai LCD-ului la pinii Arduino. Pini LCD-ului au fost conectati la cei ai placii Arduino in felul urmato:

- VCC la pinul 5V
- GND la GND

- CS la pinul digital 10
- RESET la pinul digital 9
- A0 la pinul digital 8
- SDA la pinul digital 11
- SCK la pinul digital 13
- LED la 3.3V

Pinii puteau fi configurati si in alte moduri, aceasta a fost insa configuratia aleasa de mine. Initializarea display-ului a fost facuta cu definirea pinilor astfel: `ucg(8,10,9)` Pentru conectarea butoanelor s-au folosit rezistente de PULLUP. Fiecare buton este conectat cu un pin la GND si cu celalalt la pinii placii in felul urmatoar:

- Butonul de START → Pin 12
- Buton miscare stanga → Pin 4
- Buton miscare dreapta → Pin 3
- Buton rotatie stanga → Pin 7
- Buton rotatie dreapta → Pin 6

Buzzerul s-a conectat cu un pin la GND si cu un pin la pinul 5 al placii Arduino.

Conectarea componentelor a fost usurata de utilizarea unui Breadboard, Astfel, semnalul GND a fost adus aproape de componentele de input output pentru compactarea si reducerea numarului de fire utilizate.

Software Design

Mediul de dezvoltare

Arduino IDE

Biblioteci Folosite

- [Ucglib.h](#): Folosit pentru afisarea ecranului LCD, desenarea componentelor, desenarea elementelor de text pe ecran
- [pitches.h](#): din aceasta biblioteca au fost utilizate anumite note muzicale predefinite pentru realizarea sunetelor din joc

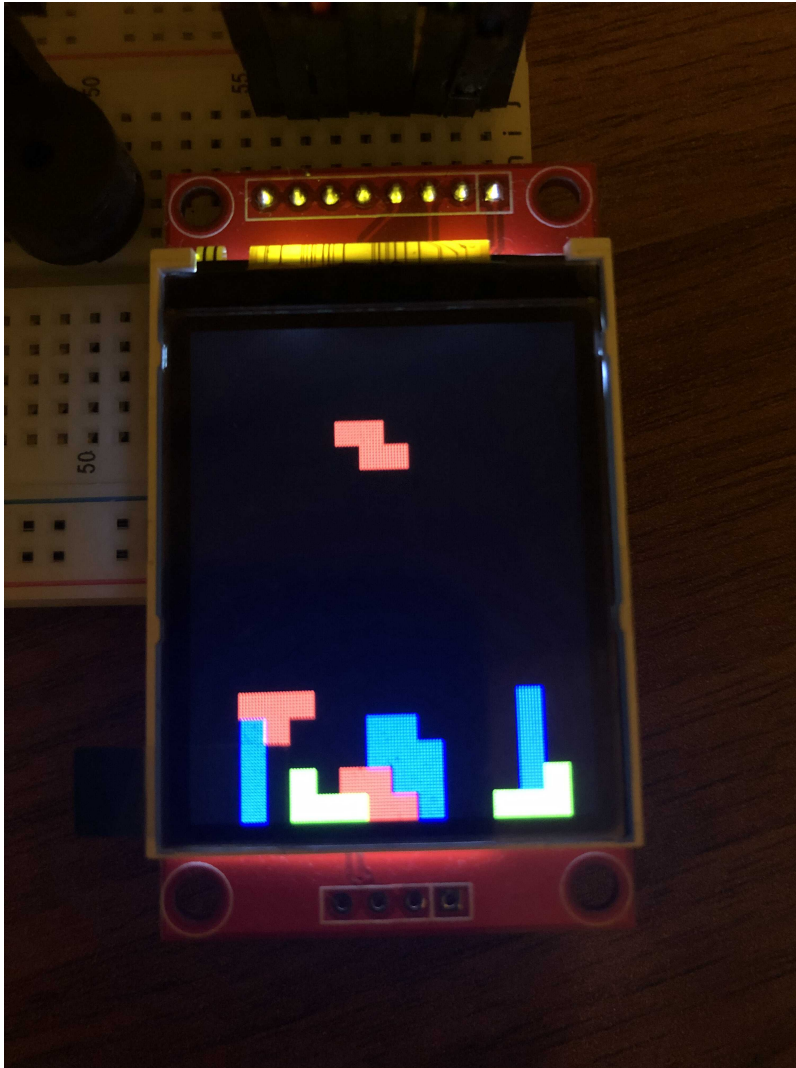
Ecrane principale ale programului

Ecranul de start:



- Ecranul care apare la alimentarea placii. Contine textul "Tetris Game", urmat de "Press Start". Ecranul indica jucatorului butonul pe care trebuie sa apese pentru a incepe jocul. Ecranul apare din nou dupa incheierea unui joc.

Ecranul principal de joc:



- Este ecranul in care se desfasoara jocul. In acest mod, apar continuu piese din partea de sus a ecranului, piese care vor fi deplasate si rotite de catre jucator prin intermediul butoanelor de control. Se poate iesi imediat din acest ecran folosind butonul de START care va incheia imediat jocul si va duce din nou utilizatorul la ecranul de start.

Ecranul de sfarsit de joc:



- Ecranul de sfarsit de joc. Acest ecran apare in momentul in care ecranul principal de joc este umplut de piese, generarea altor piese noi fiind imposibila din lipsa de spatiu. Ecranul contine textul "Game over". Buzzerul produce sunetul unei melodii in timpul afisarii ecranului. La scurt timp, utilizatorul este trimis inapoi automat la ecranul de start de unde poate incepe un joc nou.

Detalii implementare

Flow principal:

- In setup este initializat obiectul ucg care va realiza manipularea ecranului, apoi este desenat ecranul de inceput de joc. Sunt configurati pinii de input si output. Este initializata si prima piesa generata.
- Functia loop va defini doua stari posibile: Daca jocul nu este in desfasurare, se asteapta apasarea butonului de start, caz in care se va trece in starea 2. Starea 2 reprezinta starea jocului in desfasurare. Se salveaza pozitia veche a piesei, se asculta pinii de input de la butoabe, se realizeaza orice tip de miscare, in functie de starea pinilor de intrare, se deseneaza piesa pe noua pozitie si se "sterge" (de fapt se realizeaza o desenare cu negru) piesa de pe pozitia anterioara. La apasarea butonului de start se va trece inapoi in starea 1.
- Proprietatile pieselor sunt incapsulate in clasa Piese (culoare, vector de patrute, pozitie, tip de piesa, rotatie curenta). Clasa contine functii pentru rotatie, miscare, desenare, a unei piese.
- Generarea piesei se face in constructorul clasei. Se genereaza un tip de piesa random si o culoare

random folosind functia predefinita random() ce foloseste ca seed valoarea citita de pe pin-ul analogic 0. Pentru culoare, se genereaza un numar pe 3 biti diferit de 0, fiecare bit existent in numar reprezentand valoarea pentru rosu, verde respectiv albastru (se genereaza astfel 6 culori posibile diferite: rosu, verde, albastru, galben, violet, turcoaz).

- Miscarea pieselor in jos se realizeaza la fiecare jumatate de secunda folosind intreruperi, declansate la fiecare jumatate de secunda (am setat pentru asta Timer1 sa declanseze un semnal de 2Hz). Intreruperea verifica daca piesa poate cobori, si realizeaza miscarea ei. In cazul in care aceasta s-a asezat pe sol sau pe alte piese, se va genera alta piesa.
- Pentru coliziunile dintre piese se foloseste o matrice de booleeni square_map reprezentand harta de patrute deja asezate. Coliziunea piesei din aer cu patratele deja asezate se face prin verificarea existentei valorilor din square_map egale cu true pe tot parcursul miscarii piesei in aer. In momentul in care piesa isi incheie coborarea, se seteaza alte noi valori in matrice. In momentul in care o linie este completata, se elimina o linie si din aceasta matrice.

Rezultate obtinute

- Rezultatul este un joc dinamic, amuzant si care iti poate capta atentia minute bune incercand de a rezista cat mai mult fara a pierde.
- Este un joc care se aseamana foarte mult cu cele de pe gameboy-ul cu care petreceam timpul in copilarie.

Demo

- [Demo Youtube](#)

Concluzii

- Acest proiect m-a ajutat sa dobandesc foarte multe cunostiinte noi legate de lucrul cu microcontrollere, in special lucrul cu SPI, cu timere si intreruperi, utilizarea unor biblioteci Arduino si mai ales importanta eficientizarii codului datorita resurselor limitate pe care le are placa Arduino.
- Jocul este perfect functional, functioneaza fara probleme la oricate rulari succesive.
- Ca feature-uri posibile care mai pot fi adaugate pot enumera:
 - Incadrarea structurii hardware intr-o carcasa de plastic, cu ecranul si butoanele in exterior
 - Adaugarea unui buton pentru accelerarea miscarii in jos

Download

- [Soft & Schematic](#)

Jurnal

- 25.04: Realizare diagrama bloc, realizare pagina wiki initiala
- 26.04 - 08.05: - Achizitionare componente, realizarea partii hardware, testarea conexiunii cu ecranul LCD
- 09.05 - 15.05: - Constructia si generarea pieselor, implementarea miscarii corecte, realizarea tuturor rotatiilor pentru piese.
- 16.05 - 22.05: - Coliziuni, eliminari de linii, verificare incheiere a jocului, ecran de sfarsit de joc, sunete buzzer
- 23.05: Finalizare pagina de wiki

Bibliografie/Resurse

- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody/>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2021>
- https://www.youtube.com/watch?v=TQ-Z1alBUME&ab_channel=learnelectronics
- https://github.com/electricmango/Arduino-Music-Project/blob/master/A_Theme__Tetris_with_Bass/A_Theme__Tetris_with_Bass.ino
- <https://github.com/olikraus/ucglib>
- <https://gist.github.com/mikeputnam/2820675>
- Pagina aceasta in format PDF

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/tetris>

Last update: **2021/05/24 01:10**

