

PROIECTAREA MICROPROCESOARELOR

TETRIS

DOCUMENTAȚIE

Student: Nemes Emilia

Grupa: 30238

Profesor îndrumător: Mircea Paul Mureșan

Cuprins

1. Specificația proiectului.....	3
1.1 Prezentarea generală a jocului Tetris.....	3
1.2 Prezentarea versiunii implementate.....	3
2. Proiectare.....	4
3. Implementare.....	6
4. Instrucțiuni de utilizare.....	8
5. Concluzii.....	8
6. Bibliografie.....	8

1. Specificația proiectului

1.1 Prezentarea generală a jocului Tetris

Jocul tetris este un joc video, care a fost creat de inginerul software Alexei Pajitnov în anul 1984. Esența jocului este completarea cât mai multor rânduri, prin așezarea pieselor de diferite forme. Există mai multe versiuni ale jocului, care se găsesc pe platforme diverse, și pot fi jucați online pe internet sau pot fi descărcați ca și aplicații pe anumite dispozitive precum PC-uri, telefoane mobile, ceasuri smart, etc.

Precum s-a menționat, jocul constă din mutarea unor piese, generate în mod aleatoriu, într-un fel încât să se obțină cât mai multe rânduri complete, care vor fi eliminate, astfel reducând numărul pătratelor (fiecare piesă este formată din 4 pătrate) afișate pe display. În momentul în care piesa nou generată ajunge la o poziție în care în rândul următor există un pătrat al unei piese deja așezate, atunci piesa respectivă își păstrează locul și devine o piesă așezată, care nu mai poate fi mutată. După acesta se generează o nouă piesă. Jocul se termină când piesele așezate ajung la rândul cel mai de sus. Scopul jocului este de a ajunge la condiția de terminare cât de târziu se poate, astfel obținând un punctaj cât mai mare.

1.2 Prezentarea versiunii implementate

În proiectul realizat jocul conține șapte piese, cele prezentate în figura 1:

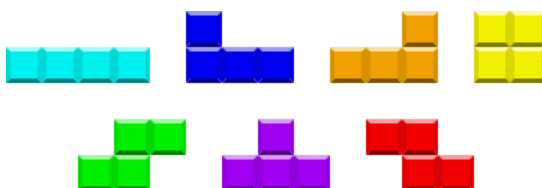


Figura 1. Piesele jocului

Forma pieselor poate fi modificată prin rotirea lor: una dintre piese are aceeași formă și după aplicarea unei rotiri, trei dintre ele au două forme rotite, iar restul au patru forme rotite, după cum se vede în figura 2:

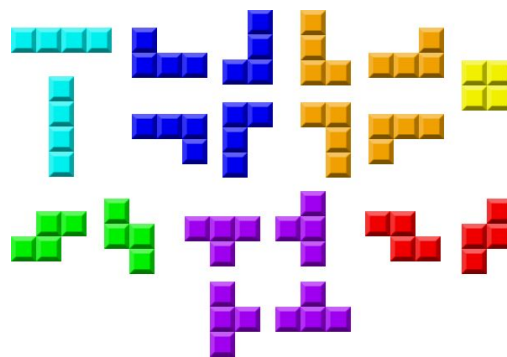


Figura 2. Piesele în formele lor rotite

Regulile pentru calcularea punctajului final sunt următoarele:

- score-ul inițial are valoarea 150
- la completarea unui rând se acord 10 puncte, care se adun la punctajul curent
- în momentul terminării jocului se numără pătratele rămase pe afișor și se scade din punctaj acest număr.

2. Proiectare

Schema bloc al proiectului arată intrările și ieșirile circuitului, singurele lucruri care sunt relevante din punctul de vedere al utilizatorului:

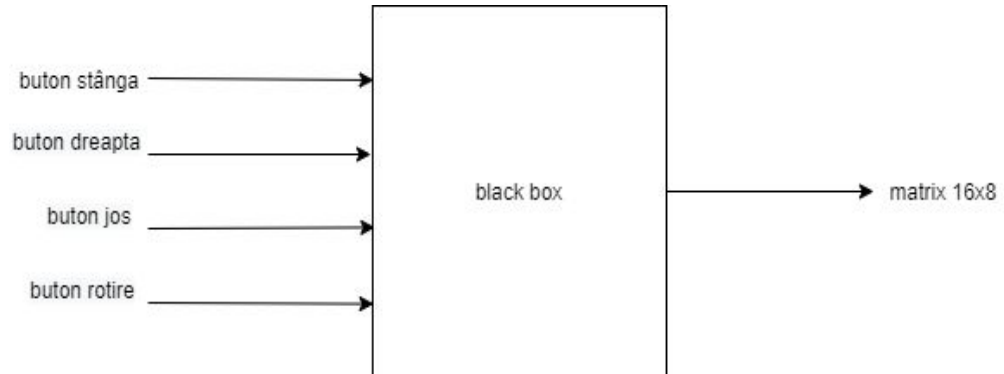


Figura 3. Schema bloc al proiectului

Pentru realizarea proiectului s-a folosit o placă de dezvoltare Arduino Mega la care s-au mai adăugat anumite dispozitive, anume: un breadboard, un set de patru butoane și două LED matrix-uri de dimensiune 8x8. Aceste dispozitive sunt prezentate pe figurile 4-7:

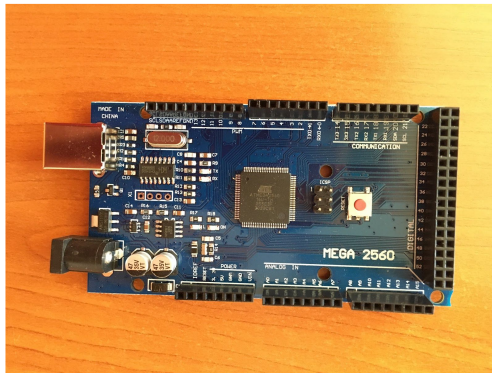


Figura 4. Arduino Mega 2560



Figura 5. Breadboard

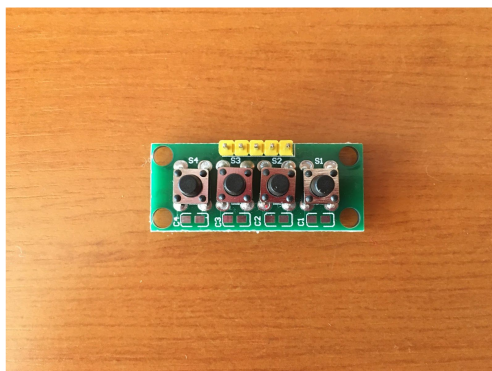


Figura 6. Butoane

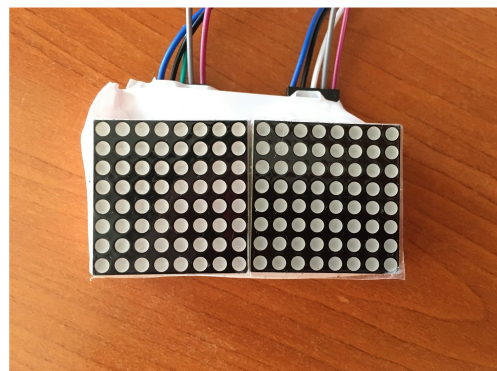


Figura 7. Două LED matrix-uri

Schema de principiu a legăturilor între componente este cea de pe figura 8, iar schema de montaj este cea prezentată pe figura 9.

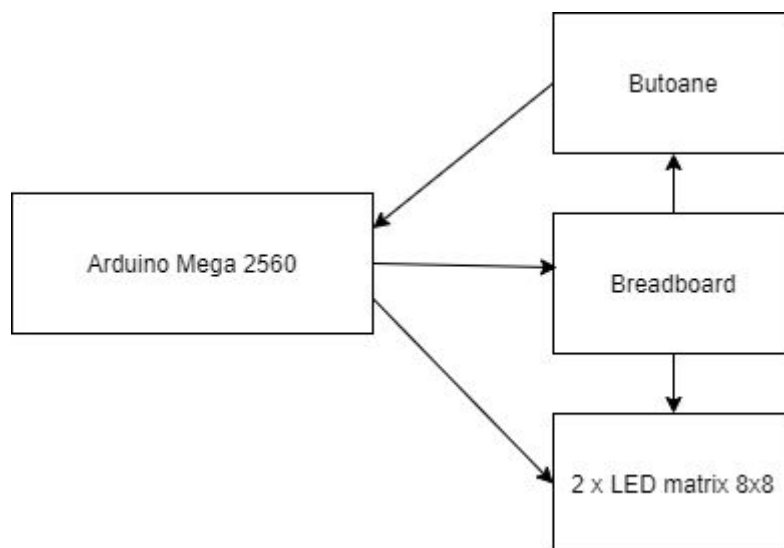


Figura 8. Schema de principiu

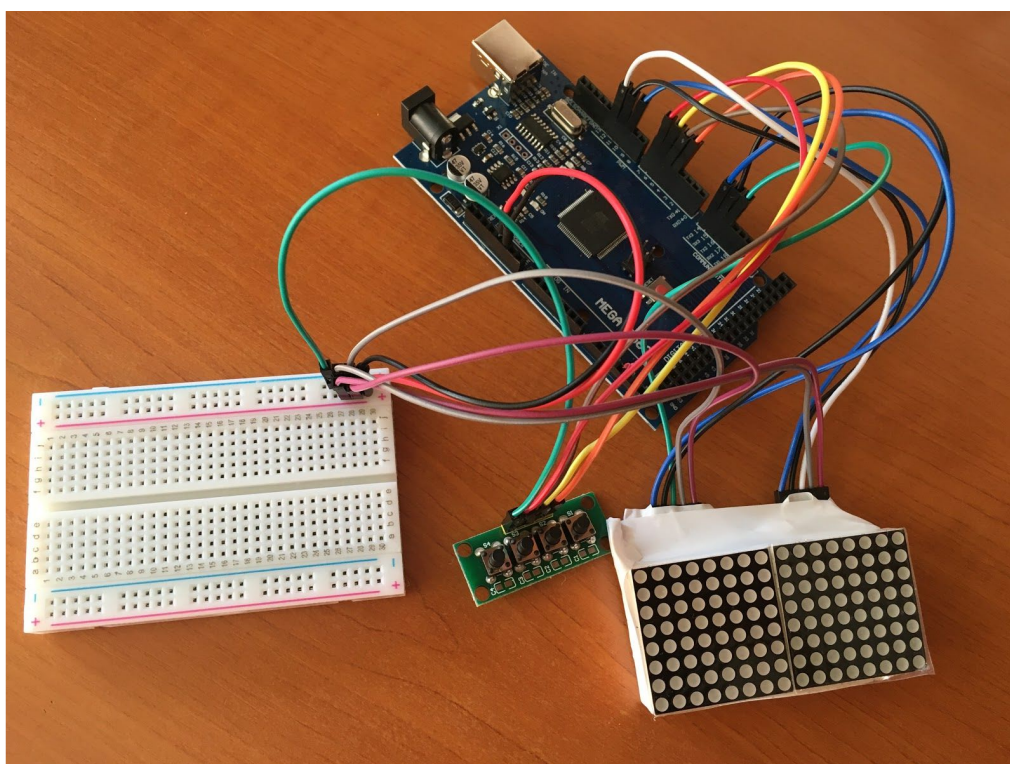


Figura 9. Schema de montaj a proiectului

Placa Arduino Mega alimentează butoanele și cele două LED matrix-uri, prin intermediul legăturilor cu breadboard-ul. La legarea în circuit butoanele necesită rezistențe, acestea însă nu apar între componente, deoarece s-au folosit rezistențele interne ale plăcii Arduino.

Butoanele sunt dispozitivele de input, impulsurile percepute pe ele fiind transferate plăcii Arduino, care modifică datele programului în mod corespunzător semnalelor primite, iar modificarea acestora a datelor va putea fi surprinsă pe afișoarele de LED matrix-uri, care sunt dispozitivele de output ale proiectului.

3. Implementare

Proiectul a fost implementat în mediul de dezvoltare Arduino IDE, iar codul a fost scris în limbajul de programare embedded C.

Codul programului este compus din două părți: una este partea care se ocupă de realizarea comunicării cu dispozitivele hardware (butoane și LED matrix-uri), a doua este partea care implementează logica jocului.

Folosirea LED matrix-urilor necesită folosirea bibliotecii **<LedControl.h>**, și declararea unor semnale pentru realizarea conectării, precum **DIN**, **CS**, **CLK**. Inițializarea acestor dispozitive precum și a butoanelor se face în funcția **setup()** al programului. Citirea datelor de pe butoane se face în funcția **loop()**, iar afișarea datelor pe LED matrix-uri se face în funcțiile **printRows1()** și **printRows2()**.

Pentru implementarea logicii jocului s-au folosit mai multe variabile și funcții. Cele mai importante variabile sunt **currentPiece**, care este reprezentarea piesei curente, care trebuie așezată. Variabila aceasta este un vector bidimensional 16x8. Dimensiunea ei este egală cu dimensiunea display-ului, pentru a simplifica modelarea deplasării ei. Pe pozițiile unde se află pătratele piesei vectorul conține valori de 1, în rest valori de 0. În mod asemănător variabila **blocks** conține piesele deja așezate, iar variabila **displayedData** conține atât piesa curentă cât și piesele așezate, deci toate datele care vor fi afișate pe display. Acestea sunt și ei vectori bidimensionali 16x8.

Există un număr mai mare de funcții, care implementează diferite operații:

- a. **initializeCurrentPiece()** și **initialize()**
 - prima inițializează variabila **currentPiece**, iar a doua variabilele **displayedData** și **blocks** cu valori de 0
- b. **generateNewPiece()**
 - aceasta generează o piesă în mod aleatoriu din cele șapte piese implementate, și o poziționează în primele rânduri
- c. **changeLEDs()**
 - schimbă valoarea variabilei **displayedData** pe baza valorilor curente ale variabilelor **currentPiece** și **blocks**
- d. **moveLeft()**, **moveRight()** și **moveDown()**
 - implementează deplasarea piesei curente cu o poziție la stânga, la dreapta respectiv cu o poziție în jos

- e. **isSpaceBelow(), isTopReached(), isRowFull(), isBlock()**
 - testează anumite condiții, precum: dacă mai există spațiu dedesubt de piesă, întrucât aceasta să se deplaseze cu o poziție în jos; dacă s-a ajuns la rândul cel mai de sus, întrucât nu se mai poate genera o piesă nouă; dacă există rând complet, care trebuie eliminat; dacă există pătrat pe o anumită poziție
- f. **moveDownOnFullRow()**
 - mută rândurile care se află deasupra rândului complet cu un rând în jos, și elimină rândul complet
- g. **generateRotatedPiece()**
 - cu ajutorul unui contor se numără de câte ori a fost apăsat butonul de rotire, și se generează următoarea formă rotită (acestea se generează în buclă) a piesei curente
 - se memorează la fiecare deplasare în jos, la stânga sau la dreapta a piesei curente poziția unde se află (cu câte poziții la dreapta și cu câte poziții în jos a ajuns, față de poziția inițială), astfel se afișează pe display pe poziția corectă
- h. **gameOver()**
 - se apelează când o piesă a fost așezată pe primul rând
 - de aici se apelează funcția care calculează și afișează punctajul final
- i. **computeScore()**
 - calculează punctajul final
- j. **displayNumber()**
 - generează formele cifrelor punctajului final care vor fi afișate pe display
- k. **displayScore()**
 - afișează punctajul final
 - acest număr trebuie să fie mai mic decât o mie, deoarece dimensiunea celor două LED matrix-uri nu permite afișarea unui număr care are mai mult de trei cifre zecimale
- l. **printRows1() și printRows2()**
 - afișează valoarea variabilei **displayData** pe cele două LED matrix-uri
- m. **setup()**
 - inițializează LED matrix-urile și butoanele
- n. **loop()**
 - percepe dacă s-a apăsat vreun buton, și apelează metodele corespunzătoare butoanelor apăstate
 - deplasează piesa curentă la fiecare secundă cu o poziție în jos

4. Instrucțiuni de utilizare

Descrierea cazurilor de utilizare:

- programul se încarcă pe placa de dezvoltare Arduino Mega
- cu ajutorul celor patru butoane se deplasează piesa curentă: cu butonul cel mai din stânga se deplasează cu o poziție la stânga, cu butonul următor se deplasează cu o poziție la dreapta, cu butonul următor se deplasează cu o poziție în jos (astfel crește viteza de deplasare în jos a piesei), iar cu butonul cel mai din dreapta se realizează rotirea piesei curente
- dacă piesa curentă ajunge la coloana cel mai din stânga sau cel mai din dreapta deplasarea în direcția respectivă nu se mai poate efectua (adică piesele nu poate să iasă din spațiul display-ului)
- dacă forma rotită a piesei nu poate fi reprezentată, din cauză că nu încapă în zona unde se află (este la margine, sau ar întrepătrunde cu piesele deja așezate) atunci rotirea nu va avea efect
- când jocul se termină, se afișează score-ul timp de 10 secunde, după care se începe jocul din nou
- acest comportament se repetă până când placa este alimentată

5. Concluzii

Proiectul prezentat realizează implementarea jocului tetris pe placa de dezvoltare Arduino, având și un dispozitiv de afișare. Utilizarea jocului este simplă, făcându-se prin utilizarea a patru butoane. Versiunea implementată este cea standardă cunoscută, însă ar putea fi îmbunătățită de exemplu cu implementarea a mai multor tipuri de piese sau prin adăugarea mai multor butoane, care ar putea să introducă alte operații, precum punerea pe pauză a jocului sau resetarea forțată.

6. Bibliografie

[1] Prezentarea pieselor jocului Tetris - de pe website-ul:

<https://tetris.fandom.com/wiki/Tetromino>

accesat la data de: 30.11.2020.

[2] Folosirea LED matrix-urilor - de pe website-ul:

<https://www.instructables.com/Multiple-LED-Matrixes-with-Arduino/>

accesat la data de: 18.12.2020.