

Roxana-Alexandra VOICA (78555) - Snake pe LED-uri

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Doresc implementarea unui joc Snake pe o matrice de 64 de LED-uri (8 x 8). Jucătorul va interacționa cu sistemul folosind 4 butoane, prin intermediul cărora va decide direcția șarpelui.

Scopul proiectului este de a realiza un joc drag din copilărie și de a pune în aplicare cunoștințele învățate la laborator.

Ideea de la care am pornit a fost să fac un joc interactiv, de aceea îmi propun să adaug funcționalități în plus folosind un buzzer, 2 butoane și un LED.

Consider că acest joc este util pentru mine pentru că mă ajută să aprofundez cunoștințele de la laborator și va fi apreciat de colegii mei pentru că reprezintă o modalitate bună de relaxare și distracție.

Descriere generală

Schema bloc:



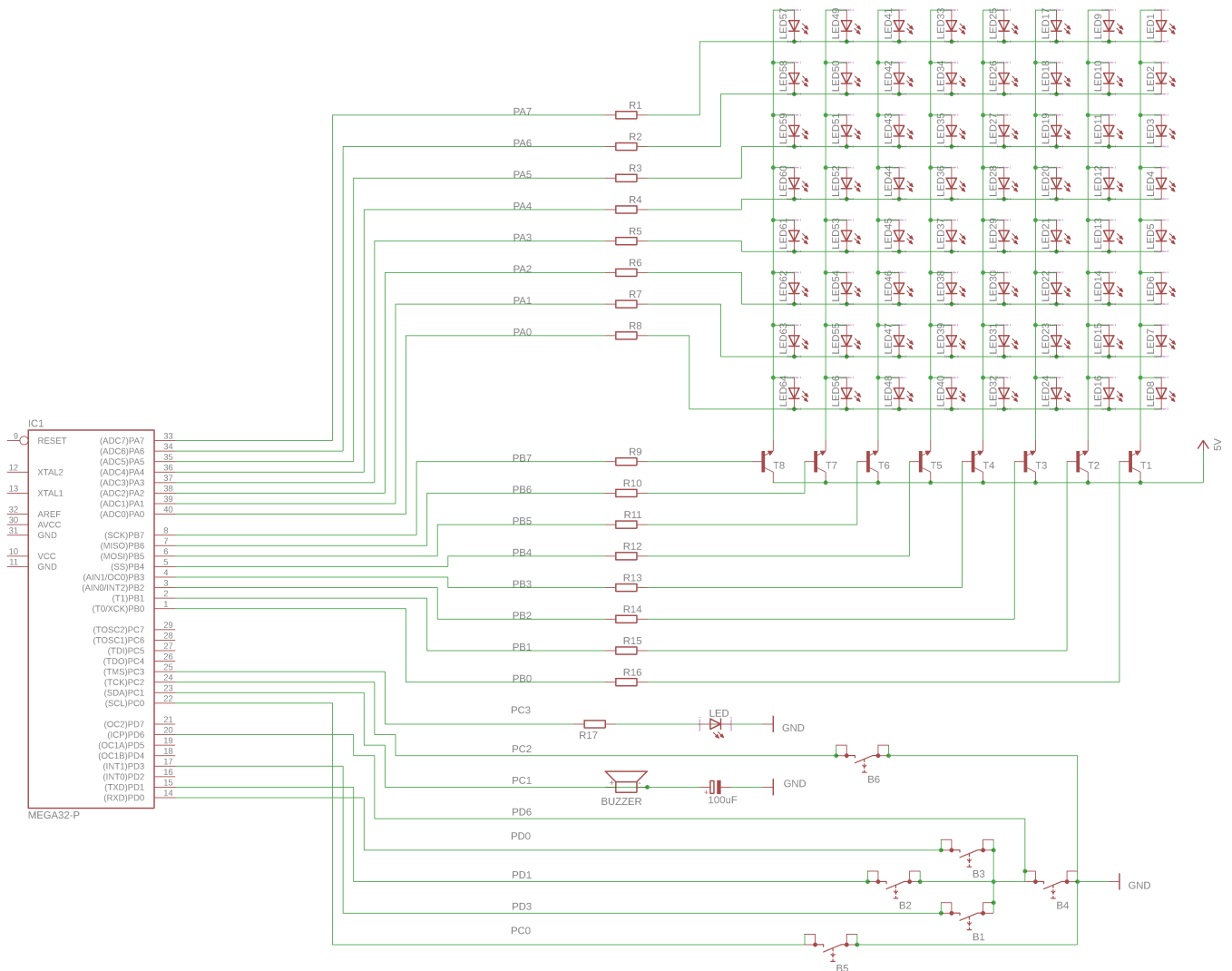
Hardware Design

Lista de componente necesare:

- placa de bază, ATmega324
- componente de bază
- placa de test
- matrice LED 8x8
- 6 butoane
- 1 buzzer
- 1 LED
- 3 fire mama-mama 8 pini, 4 fire mama-mama 1 pin

- 8 tranzistori
- 17 rezistori
- 1 condensator

Schema electrică:



Software Design

- Mediul de dezvoltare folosit: Atmel Studio 7.0, bazat pe Visual Studio
- [Bootloader USB](#)

Pentru realizarea jocului am folosit 4 butoane de direcție cu care manevrăm mișcarea șarpelui, iar controlul asupra matricei este făcut prin crearea unui vector bidimensional în care memorez care leduri sunt aprinse, astfel:

- -1 led stins
- 0-4 direcția spre cap (0 - up, 1 - right, 2 - down, 3 - left)
- 4 capul șarpelui
- 5 candy

Memorând direcția spre cap pentru fiecare led care formează șarpele, la fiecare mișcare putem

determina ușor unde este amplasată coada șarpelui. La colectarea unui candy, acela devine capul șarpelui.

Pentru a fi mai interactiv cu utilizatorul, am pus un buton de start/stop și un buzzer care sună atunci când este colectată o bomboană. Am adăugat și funcționalitatea de a nu putea trece prin pereții matricei, apăsând un buton special. Când ne aflăm în modul *blockingWalls* se aprinde ledul verde de pe placă.

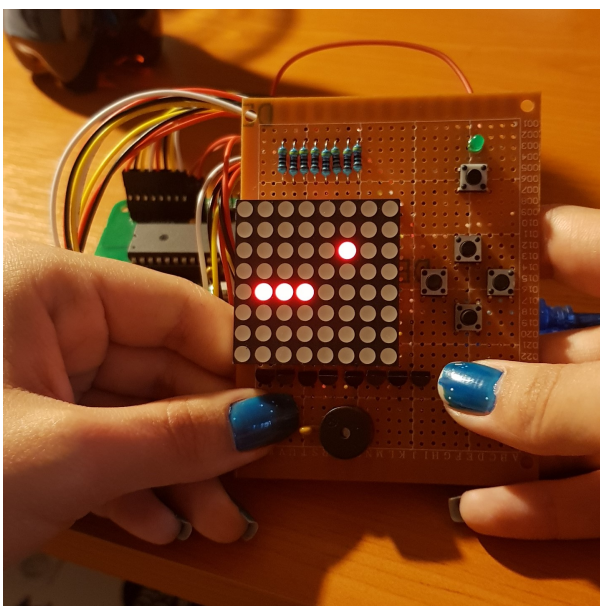
Funcțiile implementate:

- `initGame()` - initializează șarpele la lungimea 3 și poziția default
- `isUpPressed()`, `isRightPressed()`, `isDownPressed()`, `isLeftPressed()` - verifică dacă este apăsat butonul
- `isStopPressed()` - verifică dacă este apăsat butonul de stop/start
- `isBlockingWallsPressed()` - verifică dacă este apăsat butonul de blocare a pereților
- `randomizeCandy()` - generează random un candy în scenă
- `printMatrix()` - afișează șarpele și candy-ul
- `moveHead()` - modifică indecșii capului șarpelui
- `moveTail()` - modifică indecșii cozii șarpelui
- `isGameLost()` - verifică dacă s-a pierdut jocul prin autolovire
- `isGameWon()` - verifică dacă s-a câștigat jocul (șarpele are lungime 12)
- `printOk()` - afișează pe matrice 'OK'
- `getCandy()` - manancă candy
- `wallsEndGame()` - verifică dacă s-a pierdut jocul prin lovirea de pereți în modul *blockingWalls*
- `main()` - controlul general al jocului

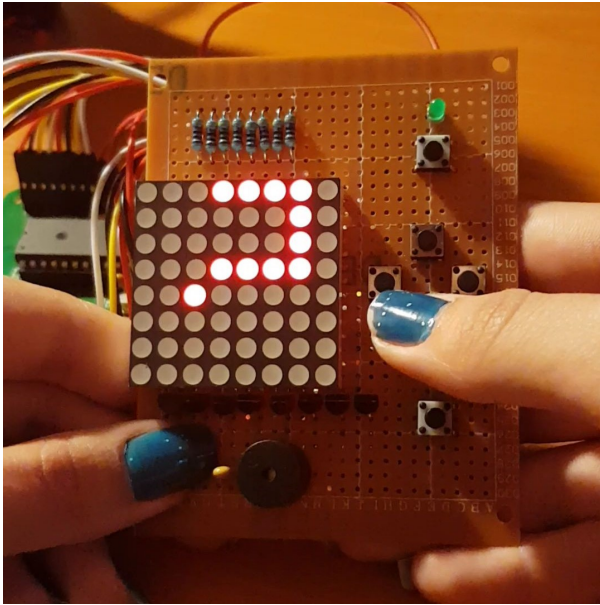
Rezultate Obținute

Proiectul a rezultat prin efectuarea cu succes a unui joc Snake colorat (matrice cu led-uri roșii), care funcționează ca un joc standard de Snake, însă cu câteva funcționalități în plus, care ajută la interacțiunea cu utilizatorul, după cum se poate vedea:

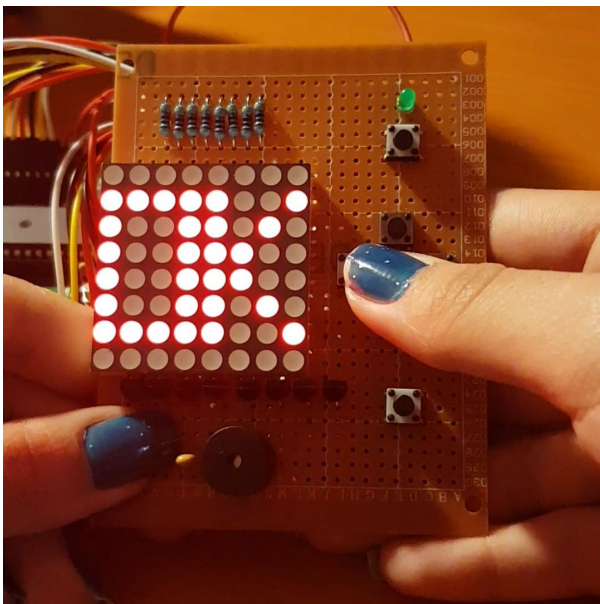
Jocul inițial:



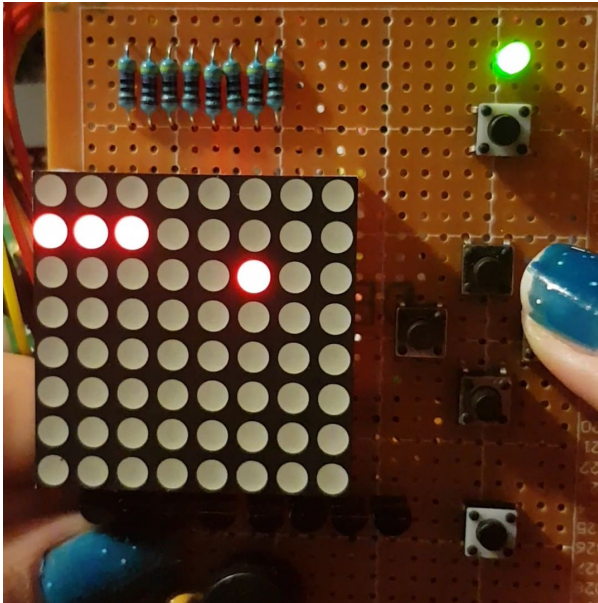
În timpul jocului:



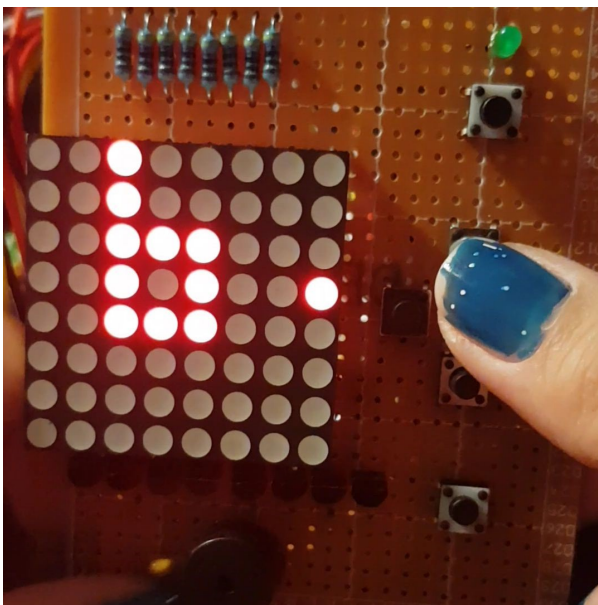
Jocul câștigat:



Jocul pierdut prin lovire de margini (se observă ledul verde aprins):



Jocul pierdut prin autolovire:



Demo video: [demo.zip](#)

Concluzii

Realizarea acestui proiect a fost o adevărată provocare, deoarece înainte nu știam să lipesc și să dezlipesc elemente pe o placută, iar acum am avut de făcut zeci de lipituri și m-am obișnuit. Proiectul mi-a adus o satisfacție foarte mare în momentul în care am reușit să îl fac perfect funcțional și mă bucur de el și în timpul liber, împreună cu colegii. Per total, mi-a plăcut mai mult partea de programare software decât cea hardware:)

Download

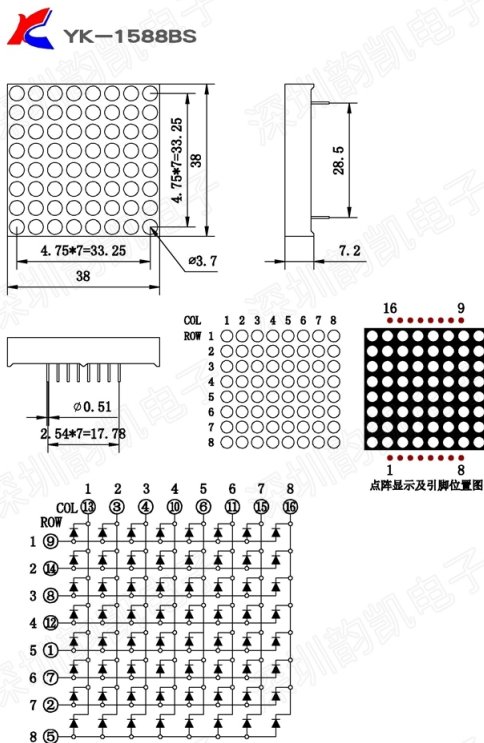
Arhiva: [335cb_roxanaalexandravoica_snake.zip](#)

Jurnal

- Saptămâna 1: lipirea componentelor pe placa de bază (2 ore)
- Saptămâna 2: realizarea schemei electrice (3 ore)
- Saptămâna 3: achiziționarea componentelor necesare proiectului (2 ore)
- Saptămâna 4: lipirea componentelor pe placa de test (6 ore)
- Saptămâna 5: realizarea soft-ului (6 ore)
- Saptămâna 6: testarea proiectului (1 ora)

Bibliografie/Resurse

- Documentația în format [PDF](#)
- Laboratoare: <http://cs.curs.pub.ro/wiki/pm/>
- Datasheet [ATMega324](#)
- <https://www.optimusdigital.ro/ro/optoelectronice-matrice-de-led-uri/944-matrice-de-led-uri-roie-8x8-375-mm.html>



- <https://www.optimusdigital.ro/ro/audio-buzzere/634-buzzer-pasiv-de-5-v.html>

- https://www.optimusdigital.ro/ro/cabluri-cabluri-usb/3146-cablu-albastru-usb-am-la-bm-30-cm-pentru-arduino-mega-i-uno.html?search_query=cablu+usb+AM+BM&results=157
- <https://www.optimusdigital.ro/ro/fire-fire-mufate/627-fir-mama-mama-8p-de-20-cm.html>
- <https://www.optimusdigital.ro/ro/prototipare-cablaje-de-test/232-cablaj-de-test.html>

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2018/rbarbascu/snake_rv



Last update: **2021/04/14 15:07**