

# Vlad MOISESCU (78558) - Nume proiect

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

## Introducere

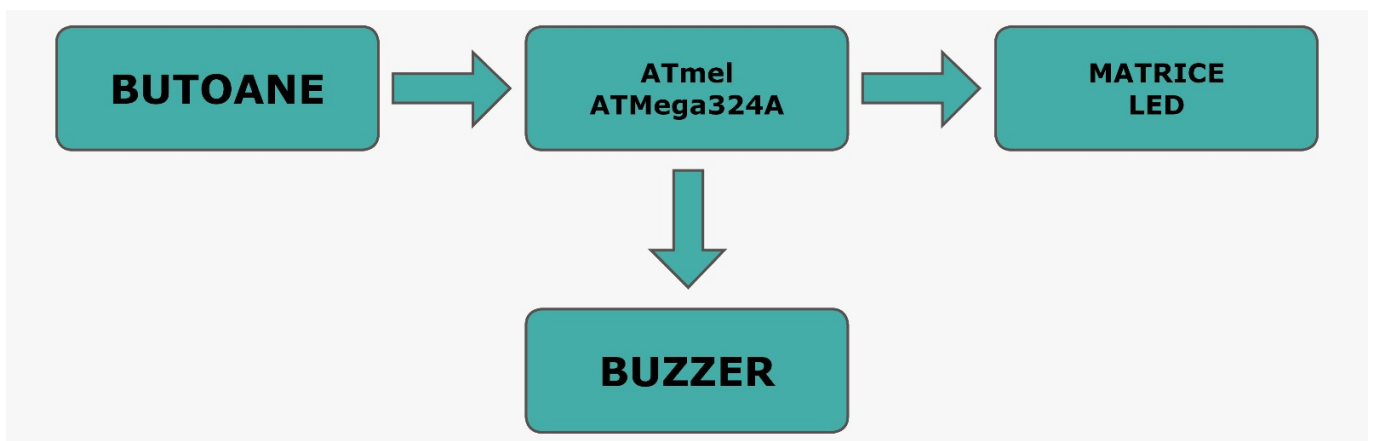
Voi implementa jocul Snake pe o matrice de leduri. Acesta se va controla prin intermediul butoanelor.

Scopul jocului este acela de a acumula un punctaj cat mai mare.

Ideea de la care a pornit jocul este aceea de a face ceva amuzant si distractiv din care pot sa invat. Am vazut o implementare a acestui joc si are un impact vizual impresionant.

Proiectul este util pentru momentele in care ai nevoie de un pic de relaxare in urma unei zile obositoare.

## Descriere generală



Jocul va rula pe o matrice de leduri 8\*8.

Se vor utiliza 6 butoane, 4 pentru miscari, unul pentru pauza si unul pentru joc nou.

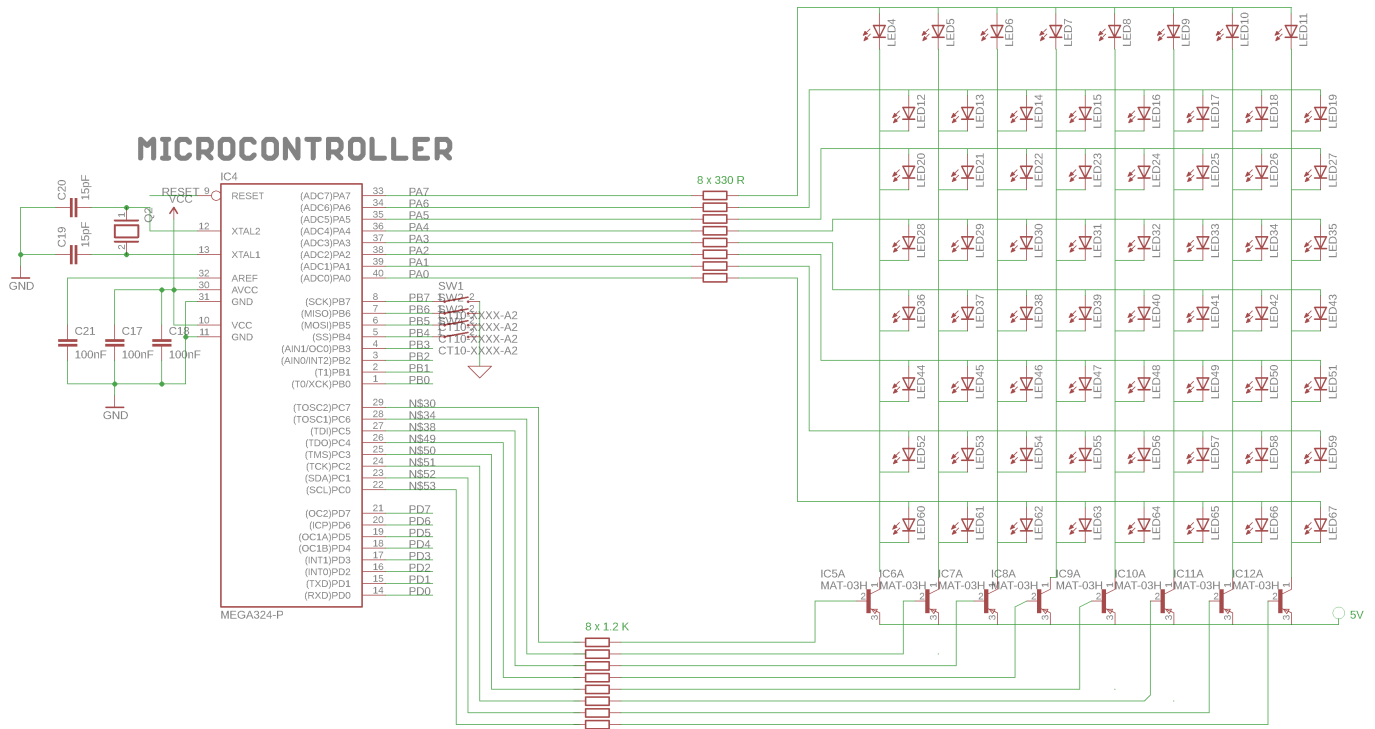
In momentul in care jucatorul este in pericol de a pierde, buzzerul va porni un semnal de atentionare.

## Hardware Design

Lista de piese necesare:

- placa de bază, ATmega324
- componente de bază
- placa de test
- matrice LED 8x8
- 4 butoane
- 1 buzzer
- fire mama-mama,
- fire tata-tata

Schema electrica:



## Software Design

Pentru realizarea părții software a proiectului am folosit Atmel Studio 7.0, din care am generat un fisier .hex pe care l-am încărcat pe plăcuță cu ajutorul aplicației puse la dispoziție: HIDBootFlash.

Pentru aconstrui sarpele am utilizat un vector de coordonate. Initial sarpele are lungime 2, iar restul elementelor din vector care nu fac parte din sarpe sunt initializate cu -1. La fiecare miscare a sarpelei schimb valoarea capului in functie de directie, iar restul elementelor iau valoarea elementului din fata lor. Coliziunea o verific doar cu capul, deoarece numai acesta poate sa produca pierderea jocului. Mancarea sarpelei este tot un element de tip coordonata si o generez folosind functia rand(). Toate acestea le retin pentru afisare intr-un vector cu dimensiunea de 8 si elemente de tip uint8\_t pe care il reinitializez la fiecare pas. Comenzile de miscare ale sarpelei sunt implementate folosind intreruperi pentru a evia dezavantajele delay-ului.

Functii implementate:

- coord generate\_coord(void)
- void init\_snake(uint32\_t seed)

- void update\_game(void)
- void detect\_collision(void)
- void spi\_send(uint8\_t data)
- void max7219\_writec(uint8\_t high\_byte, uint8\_t low\_byte)
- void max7219\_clear(void)
- void max7219\_init(void)
- void update\_display(void)
- void image(const uint8\_t im[8])
- void set\_pixel(uint8\_t r, uint8\_t c, uint8\_t value)

## Rezultate Obținute

## Concluzii

A fost un proiect interesant din care am învățat multe lucruri ce țin de partea hardware.

## Download

[vlad\\_moisescu.zip](#)

## Jurnal

- lipirea componentelor pe placa de bază
- realizarea schemei electrice
- achiziționarea componentelor necesare proiectului
- realizarea hard-ului
- realizarea soft-ului
- testarea proiectului

## Bibliografie/Resurse

- Laboratoare: <http://cs.curs.pub.ro/wiki/pm/>
- Datasheet ATmega324
- proiecte din anii trecuti
- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/asolot/moiescu-vlad>



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