

# Snake Game

## Autor

Nume: Olaru Alexandru

Grupa: 336CA

## Introducere

Fiind pasionat de jocuri video și de console, m-am gândit să retrăiesc bucuria din copilărie atunci când mă jucam Snake pe diversele mini-console de la acea vreme. Așadar, proiectul reprezintă un joc Snake, ce încearcă să fie cât mai asemănător cu un joc pe o consolă portabilă.

## Descriere generală

Schema bloc este compusă din următoarele componente: un Arduino Uno, o matrice de leduri 8×8, un potențiomtru, un buzzer și un joystick. Matricea de leduri stă în locul ecranului, pe care m-am gândit să îl înlocuiesc, oferindu-i jocului un aspect mai retro. Ledurile aprinse vor semnifica atât șarpele, cât și merele la care acesta va trebui să ajungă. Joystick-ul reprezintă singura modalitate de input a utilizatorului, prin care acesta poate mișca șarpele în 4 direcții: sus, jos, stânga și dreapta. Nu în ultimul rând, prin intermediul potențiometrului se poate regla viteza de joc, iar buzzer-ul este post de difuzor.



## Hardware Design

Lista de piese folosite:

- Arduino Uno
- Breadboard
- Modul joystick compatibil Arduino
- Matrice 8×8 leduri OKY3523

- Buzzer activ OKY0151
- Potentiometru rotativ OKY0107

Schema electrica:



Matrice Led:

- VCC → 5V
- GND → GND
- DIN → D12
- CS → D11
- CLK → D10

Modul Joystick:

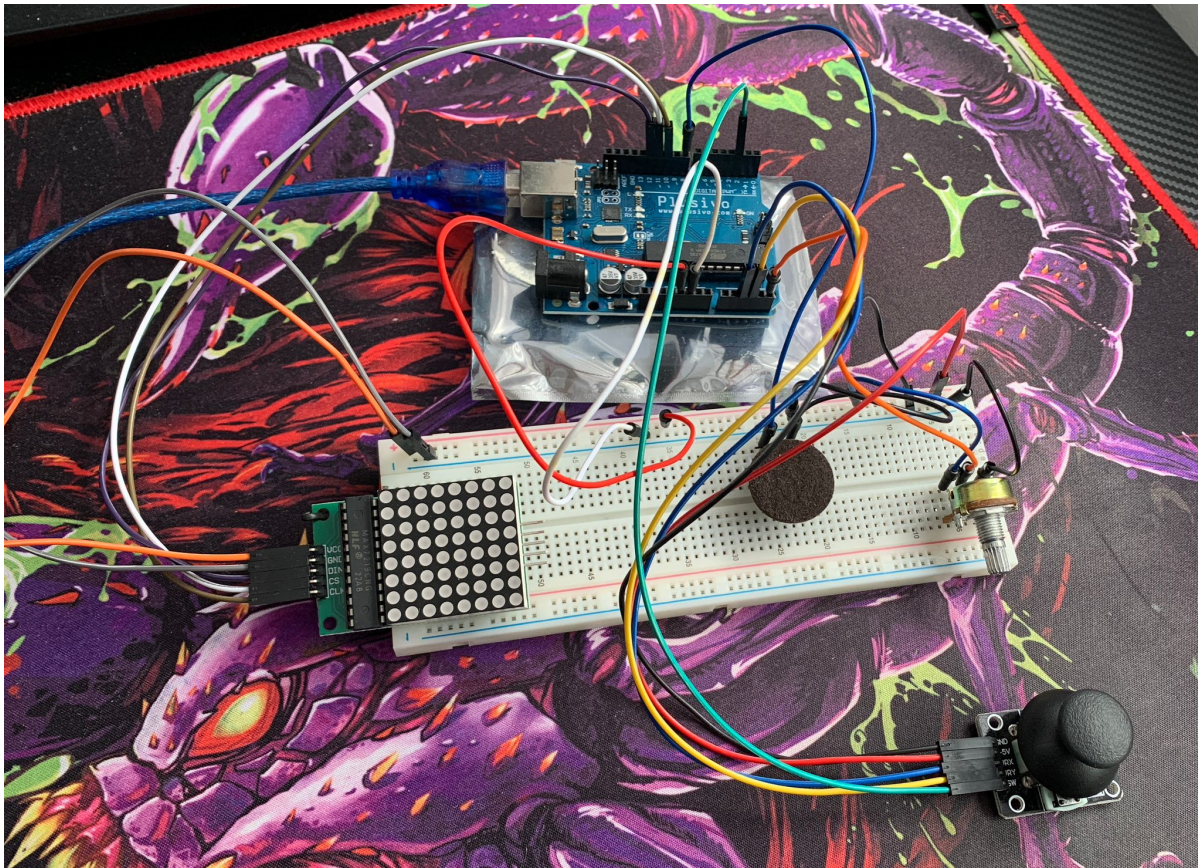
- GND → GND
- VCC → 5V
- VRX → A2
- VRY → A3
- SW → D2

Buzzer:

- - → GND
- + → D8

Potentiometer:

- VCC → 5V
- VOUT → A5
- GND → GND



## Software Design

Descrierea codului aplicației:

- Mediul de dezvoltare: Arduino IDE
- Bibliotecile și sursele 3rd-party folosite: <https://github.com/wayoda/LedControl>
- Surse proprii: fișierul "Snake\_Game.ino" și header-ul "Snake.h"

Structura codului arată astfel:

```
void setup() {  
  initialize();  
  display_snake_message(message_snake);  
}
```

```
void loop() {  
  /* Generate food on the display */  
  generate_food();  
  /* Scan joystick input and move the snake */  
  scan_input();  
  /* Snake movement */  
}
```

```
    move_snake();
    /* Check and handle game over situation */
    check_game_over();
}
```

Implementare:

- Funcția `initialize()` inițializează variabilele ce se ocupă cu stările jocului, configurează ADC (Analog-to-Digital Converter), ce va fi folosit pentru a citi inputul de la joystick, precum și matricea de led-uri.
- Funcția `display_snake_message(message_snake)` va afișa un text dinamic atunci când "consola" va fi pornită. Acest text va dispărea atunci când se înregistrează un input de la joystick (fie o mișcare a manetei, fie o apăsare a butonului acesteia).
- Funcția `generate_food()` generează mâncarea șarpelui într-un loc random (unul din cele 64 led-uri de pe matrice), care este valid și care nu este ocupat de șarpe. Aceasta se va genera, schimbându-și locul, doar după ce șarpele o va mânca.
- Funcția `scan_input()` va citi input-ul de la joystick, calculând direcția de deplasare a șarpelui. Aceasta se va ocupa și de viteza jocului, care poate fi setată cu ajutorul potențiometrului (cu cât viteza este mai mică, cu atât funcția se termină mai greu, dictând intervalul de timp în care se aprind și se sting led-urile).
- Funcția `move_snake()` este cea care controlează locația în care se află șarpele, în funcție de direcția acestuia. Aceasta tratează cazurile în care șarpele își mușcă propria coadă, mănâncă mărul, trece prin marginile "display-ului" și apare din părțile opuse, și se ocupă de aprinderea corespunzătoare a led-urilor în funcție de poziția corpului șarpelui. Atunci când șarpele va mânca mărul, un sunet scurt va fi emis de către buzzer.
- Funcția `check_game_over()` verifică dacă jocul este încheiat, caz în care va afișa un mesaj de "GAME OVER". Buzzer-ul va emite un sunet ceva mai lung, pe o tonalitate mai joasă, iar variabilele de joc vor fi resetate, așteptând input de la joystick pentru ca jocul să poată fi jucat din nou.

Citirea input-ului de la joystick (valorile X și Y) este făcută folosind ADC:

```
uint8_t adc_read(uint8_t pin) {
    /* Select ADC channel (pin) for conversion */
    ADMUX = (ADMUX & 0xF0) | (pin & 0x0F);
    /* Start the conversion */
    ADCSRA |= (1 << ADSC);
    /* Wait for the conversion to complete */
    while (ADCSRA & (1 << ADSC));
    /* Return the result */
    return ADCH;
}
```

Pentru a aprinde și a stinge led-urile matricei, am folosit funcții din biblioteca <https://github.com/wayoda/LedControl>, după mai multe încercări de a implementa această funcționalitate "de mână", folosind SPI (Serial Peripheral Interface), care este de asemenea folosit și în funcțiile bibliotecii, într-o modalitate asemănătoare cu cea prezentată mai jos:

```
void send_data(uint8_t address, uint8_t value) {
    digitalWrite(CS, LOW);
    /* Send address */
    SPI.transfer(address);
}
```

```
/* Send value */
SPI.transfer(value);
/* Finish transfer */
digitalWrite(CS, HIGH);
}
```

Astfel, pinul Chip-Select (CS) este configurat, trimițându-se mai apoi prin SPI adresa registrului lui MAX7219 al matricei de leduri, precum și valoarea dorită. Dacă se dorește aprinderea sau stingerea led-urilor, conform specificațiilor din datasheet (<https://www.analog.com/media/en/technical-documentation/data-sheets/max7219-max7221.pdf>), adresa va fi un număr de la 1 la 8, reprezentând linia din matrice pe care se vor aplica modificări, iar valoarea un număr de la 0 la 255, indicând ce led-uri vor fi aprinse, respectiv stinse.

## Rezultate Obținute

Rezultatele obținute în urma realizării proiectului se pot observa în următorul demo: [https://www.youtube.com/watch?v=BKTVWNCKRaY&t=9s&ab\\_channel=AlexandruOlaru](https://www.youtube.com/watch?v=BKTVWNCKRaY&t=9s&ab_channel=AlexandruOlaru). Fiind primul proiect de acest tip pentru mine, în care am folosit componente hardware fizice și le-am asamblat, pot spune că mi-a făcut plăcere să lucrez la el și că mi-a oferit o satisfacție mai mare decât majoritatea proiectelor la care am lucrat până acum.

**UPDATE: Versiunea finală a proiectului o puteți observa în următorul demo:**

[https://www.youtube.com/watch?v=qB7Kzn-Cgeg&t=4s&ab\\_channel=AlexandruOlaru](https://www.youtube.com/watch?v=qB7Kzn-Cgeg&t=4s&ab_channel=AlexandruOlaru)

## Concluzii

În concluzie, am reușit să creez ceva asemănător unei mini-console portabile mai vechi cu manetă, pe care se poate juca doar jocul Snake. Viteza de joc, adică de deplasare a șarpelui, poate fi setată cu ajutorul rotiței potențiometrului, în acest sens putând fi schimbată dificultatea jocului. Este un proiect care va putea fi folosit și în viitorul îndepărtat, atunci când îți se face dor să rejoci unul din jocurile alături de care ai copilărit.

## Download

Arhiva proiectului: [snakeolaru.zip](#)

# Jurnal

- 3 Mai 2023: M1 - Introducere, Descriere generală, Schemă bloc, Listă componente hardware
- 14 Mai 2023: M2 - Schemă electrică, Hardware Design
- 21 Mai 2023: M3 - Implementare software, Software design
- 22 Mai 2023: Rezultate obținute

## Bibliografie/Resurse

### Resurse Hardware:

- Data Sheet MAX7219 (matrice LED-uri): <https://www.analog.com/media/en/technical-documentation/data-sheets/max7219-max7221.pdf>
- Buzzer: <https://www.ardumotive.com/how-to-use-a-buzzer-en.html>
- Modul Joystick: <https://arduinogetstarted.com/tutorials/arduino-joystick>

### Resurse Software:

- Matrice LED-uri: <https://projecthub.arduino.cc/mdraber/0c417a04-ec3f-405a-a383-b2d66e889e7a>
- Matrice LED-uri: <https://github.com/wayoda/LedControl>
- ADC: [https://electronoobs.com/eng\\_arduino\\_tut156.php](https://electronoobs.com/eng_arduino_tut156.php)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/snakegame>



Last update: **2023/05/28 17:23**