Snake Game on 8x8 LED Matrix

Autor: STAMATIE Mihai-Robert, 333CA

Introducere

Proiectul reprezinta o implementare hardware a jocului clasic **Snake** pe o matrice de LED-uri 8×8, avand ca microcontroller o placa **Arduino Leonardo** (ATmega32U4 @ 16 MHz). Jucatorul controleaza sarpele cu ajutorul unui joystick analogic, iar butonul integrat al joystick-ului (SW — apasarea in jos a manetei) permite resetarea jocului in orice moment.

Firmware-ul este scris in **C bare-metal** (avr-libc + avr-gcc), fara framework-ul Arduino, cu acces direct la registrele microcontrollerului — abordare care permite control fin asupra timpilor si folosirea eficienta a resurselor disponibile pe ATmega32U4.

Logica jocului implementeaza:

- generarea aleatorie a pozitiei hranei pe matrice (LFSR pe 16 biti, seed-uit din zgomotul ADC);
- cresterea progresiva a sarpelui la fiecare hrana consumata;
- detectia coliziunilor cu propriul corp;
- **wrap-around** la marginile matricei (sarpele “iese” pe o margine si apare pe partea opusa);
- cresterea progresiva a vitezei pe masura ce scorul avanseaza;
- afisarea scorului direct pe matricea de LED-uri la finalul partidei (font 3×5 pixeli);
- salvarea high-score-ului in EEPROM, persistent intre resetari;
- animatii pentru boot, death si high-score nou.

Motivatia: mi-am dorit un proiect care sa combine partea de I/O analogic (joystick, citire ADC) cu o componenta de afisare controlata printr-un protocol serial (SPI software, prin MAX7219) si cu stocare persistenta (EEPROM), pentru a acoperi cat mai multe notiuni din curs intr-un singur proiect. In plus, Snake este un joc cu logica simpla dar captivanta, iar provocarea de a-l face sa ruleze pe doar 64 de pixeli este parte din distractie.

Descriere generala

Sistemul este compus din trei parti functionale principale:

- **Input:** joystick-ul analogic furnizeaza doua semnale analogice (axele X si Y) citite de ADC-ul pe 10 biti al microcontrollerului, plus un semnal digital (SW) folosit pentru reset, conectat la PD1 cu pull-up intern.
- **Procesare:** placa Arduino Leonardo (ATmega32U4 @ 16 MHz) ruleaza logica jocului — citeste input-ul, actualizeaza pozitia sarpelui in memorie, genereaza aleator coordonatele hranei, detecteaza coliziuni, calculeaza scorul si gestioneaza stocarea high-score-ului in EEPROM.

- **Output:** matricea de LED-uri 8×8, controlata prin driver-ul MAX7219 via SPI software (bit-banging), afiseaza in timp real starea jocului: pozitia sarpelui, hrana (cu efect de blink), scorul final si animatiile de stare.

Alimentarea se face de la portul micro-USB al PC-ului (5V), care serveste simultan si ca interfata de programare prin USB-ul nativ al ATmega32U4.

snake_game_2026pm-project.png

Hardware Design

Lista de piese (Bill of Materials)

Nr.	Componenta	Cantitate	Descriere / Rol
1	Placa Arduino Leonardo	1	Microcontroller ATmega32U4 @ 16 MHz
2	Matrice LED 8×8 cu driver MAX7219 (LED Dot Matrix Display, Red)	1	Display-ul propriu-zis al jocului
3	Modul joystick analogic (Generic, ex. KY-023)	1	Control directie X/Y + buton SW integrat pentru restart
4	Breadboard mic (Solderless Breadboard Half Size)	1	Suport pentru cablaj
5	Fire dupont tata-mama (Female/Female Jumper Wires)	~10	Conexiuni Arduino — module
6	Fire dupont generice (Jumper wires generic)	~5	Conexiuni pe breadboard
7	Cablu micro-USB	1	Programare si alimentare

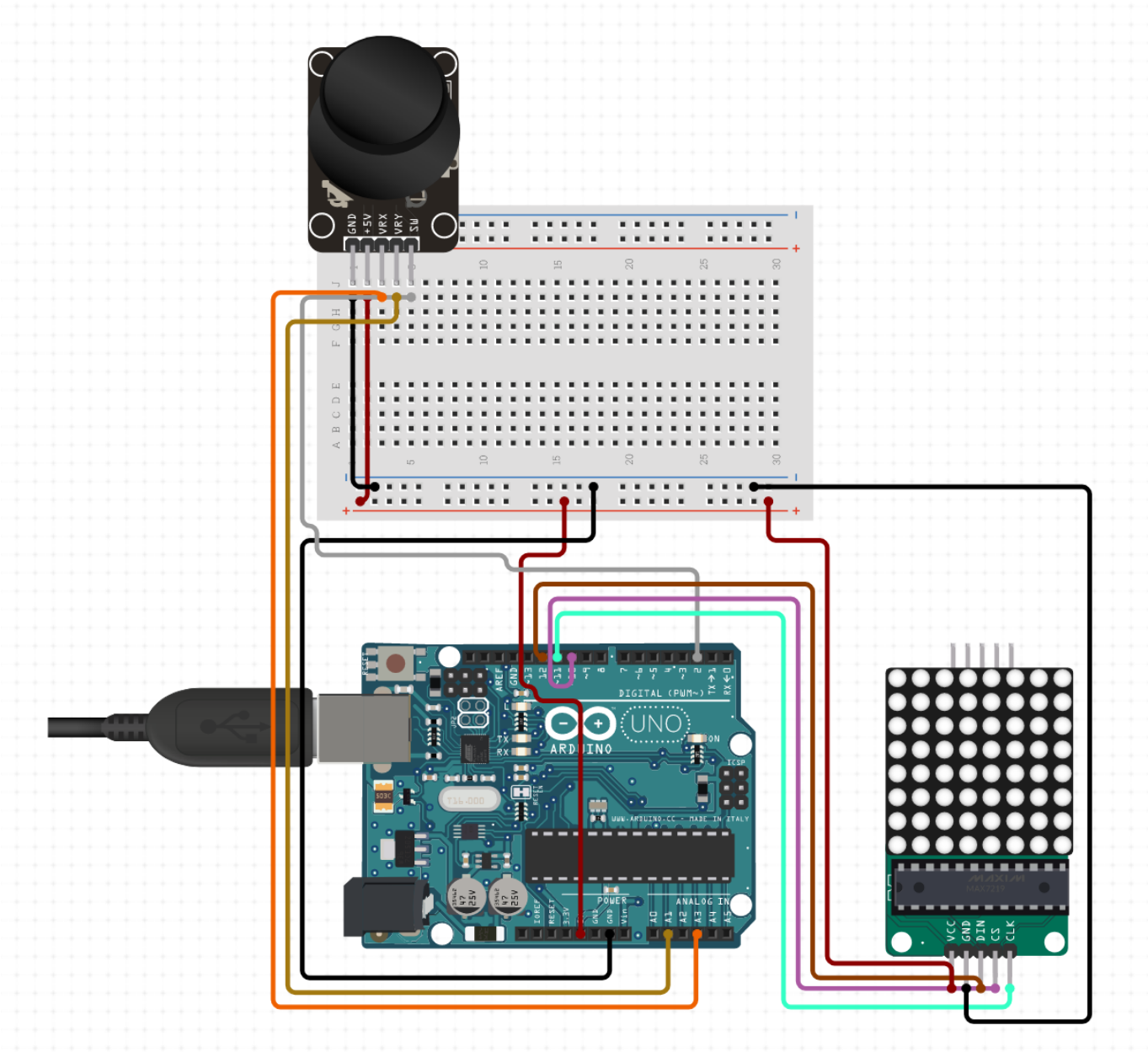
Conexiuni

Componenta	Pin componenta	Pin Arduino Leonardo	Port / Pin AVR
Joystick	VCC	5V	—
Joystick	GND	GND	—
Joystick	VRX	A0	PF7 (ADC7)
Joystick	VRY	A1	PF6 (ADC6)
Joystick	SW (buton integrat)	D2	PD1 (pull-up intern, active LOW — folosit pentru restart)
Matrice MAX7219	VCC	5V	—
Matrice MAX7219	GND	GND	—
Matrice MAX7219	DIN	D11	PB7
Matrice MAX7219	CS	D10	PB6
Matrice MAX7219	CLK	D13	PC7

Nota despre SPI pe Leonardo: pinii hardware SPI ai placii Leonardo (MOSI / MISO / SCK) sunt

expusi doar pe header-ul ICSP, nu pe D11/D12/D13 ca la UNO. In proiect este folosit **SPI software** (bit-banging direct pe registrele PORTB / PORTC), asa ca alegerea pinilor D10 / D11 / D13 este pur conventionala. Pinul D13 este partajat cu LED-ul on-board, dar acest lucru nu afecteaza comunicatia cu MAX7219.

Schema electrica



Schema de mai sus prezinta cablajul complet realizat pe breadboard: joystick-ul analogic conectat la pinii A0 (VRX), A1 (VRY) si D2 (SW pentru restart), iar matricea LED 8×8 cu driver MAX7219 conectata la pinii D11 (DIN), D10 (CS), D13 (CLK). Alimentarea de 5V este distribuita prin rail-urile breadboard-ului direct de la pinii Arduino.

Software Design

Mediu de dezvoltare

- **Toolchain:** `avr-gcc` + `avr-libc` + `avr-objcopy`
- **Flashing:** `avrdude` (protocol `avr109` pentru Leonardo, baud 57600)
- **Build system:** Makefile clasic (vezi sectiunea Download)
- **Limbaj:** C pur (C99), fara framework Arduino — acces direct la registrele MCU

Arhitectura codului

Codul este organizat modular in mai multe grupuri de functii, toate in `main.c`:

- **Driver MAX7219 (SPI software):** `spi_init()`, `spi_tx()`, `mx_wr()`, `mx_init()` — toggle direct pe PORTB / PORTC pentru DIN, CS, CLK.
- **Framebuffer:** `fb_clr()`, `fb_dot()`, `fb_push()` — un array `uint8_t fb[8]` tine starea matricei (un byte per linie, fiecare bit = un LED). Push pe matrice se face cu `mx_wr(r+1, fb[r])`.
- **Driver ADC:** `adc_init()`, `adc_rd(ch)` — citire pe canalele ADC6 / ADC7 (A1 / A0), prescaler 128.
- **Driver buton:** `btn_init()`, `btn_rst()` — pull-up intern pe PD1, debounce software 30 ms.
- **EEPROM:** `ee_load()`, `ee_save()` — high-score-ul e salvat la adresa 0x00, cu un byte “guard” (0xD4) la adresa 0x01 pentru a detecta EEPROM neinitializat la primul boot.
- **Logica joc:** `read_joy()` (citeste joystick + previne intoarcerea pe directia opusa), `game_step()` (actualizeaza pozitia cu wrap-around la margini, detecteaza coliziuni, mananca hrana), `place_apple()` (random pana se gaseste o celula libera), `cur_steps()` (calculeaza viteza curenta in functie de scor).
- **Rendering:** `draw()` (deseneaza sarpe + hrana blinking), `show_score()` (afiseaza scorul cu font 3×5), `anim_boot()` / `anim_dead()` (animatii).

Algoritmi si structuri principale

- **Reprezentarea sarpelui:** array de coordonate `Pt snake[MAXLEN]`, unde `MAXLEN = 64`. Capul e `snake[0]`. La fiecare pas, toate elementele sunt shift-uite cu o pozitie si capul nou e scris la index 0.
- **Detectia coliziunilor:** se parcurge `snake[]` si se verifica daca noul cap se suprapune cu vreun segment existent.
- **Random (LFSR):** generator pseudo-aleator pe 16 biti cu polinom `0xB400`, seed-uit la boot din zgomotul ADC al ambelor axe — nu necesita hardware suplimentar.
- **Anti-reverse:** input-ul de la joystick e filtrat pentru a nu permite intoarcerea direct pe directia opusa (ar duce la coliziune instant).
- **Viteza adaptiva:** `cur_steps()` returneaza numarul de tick-uri (de 20 ms fiecare) intre deplasari — scade liniar cu scorul, de la 14 (start) la 4 (rapid).
- **Font scor:** `DIG[10][5]` — fiecare cifra ocupa 3×5 pixeli, codificata ca 5 byti (low 3 biti activi per linie).

Concluzii

Proiectul a fost finalizat cu succes — jocul Snake este complet functional pe placa Arduino Leonardo, ruleaza fluid pe matricea LED 8×8 si raspunde rapid la input-ul de la joystick. Implementarea in **C bare-metal** s-a dovedit a fi o alegere potrivita pentru acest tip de proiect, oferind control direct asupra timpilor de executie si o intelegere mai profunda a modului de functionare a microcontrollerului ATmega32U4 la nivel de registre.

Aspecte care au functionat bine

- Driver-ul **SPI software** pentru MAX7219 s-a dovedit suficient de rapid si stabil pentru rate-ul de refresh dorit, fara a fi necesar SPI hardware (care oricum nu este disponibil pe pinii D11/D12/D13 pe Leonardo).
- Generatorul **LFSR pe 16 biti**, seed-uit din zgomotul ADC al ambelor axe la pornire, ofera o distributie rezonabila a pozitiilor hranei fara a necesita hardware suplimentar (timer extern, RTC).
- Salvarea high-score-ului in **EEPROM** cu un byte de guard (`0xD4`) previne afisarea de valori reziduale la primul boot dupa flashing.
- **Wrap-around-ul** la marginile matricei face jocul mai accesibil decat varianta clasica cu coliziuni la pereti, fiind o decizie de design care creste durata medie a partidelor.
- Folosirea **butonului SW integrat** in joystick (in loc de un buton extern separat) reduce numarul de componente si simplifica cablajul.

Provocari intampinate

- Maparea pinilor pe **ATmega32U4 difera semnificativ de ATmega328P** (Arduino UNO) — in special faptul ca SPI hardware nu este pe pinii digitali standard, ci doar pe header-ul ICSP. Solutia adoptata (SPI software) este insa mai flexibila si independenta de placa folosita.
- **Calibrarea zonelor moarte** ale joystick-ului (`JOY_LO = 300`, `JOY_HI = 700`) a necesitat experimentare — prea mica genereaza schimbari de directie false, prea mare face controlul lent si imprecis.
- Vitezele de joc (`STEPS_SLOW = 14`, `STEPS_FAST = 4`) au necesitat ajustare pentru un gameplay placut: prea rapid in faza incipienta, jocul devine frustrant; prea lent, devine plictisitor.

Posibile extensii viitoare

- Adaugarea unui **buzzer pasiv** pentru efecte sonore (PWM via timer hardware) la consumarea hranei si la game over.
- Implementarea unui **meniu de selectie** a dificultatii initiale (slow / normal / hard).
- Optimizarea consumului prin utilizarea **modurilor sleep** ale MCU intre tick-uri.
- Portare pe alta placa AVR (Mega, Uno, Nano) cu modificari minime la nivel de pini.

Download

Arhiva contine **varianta finala a proiectului**, gata de compilat si rulat pe Arduino Leonardo. Include

sursele complete (`main.c`), Makefile pentru build si flash, plus un fisier README cu instructiuni de utilizare.

- snake_game.zip — varianta finala a proiectului

Pentru a rula proiectul:

1. Conectati placa Arduino Leonardo prin micro-USB
2. Dezarhivati `snake_game.zip`
3. In folder-ul dezarhivat, rulati `make` pentru a compila
4. Rulati `make flash` (sau scriptul `flash.ps1` pe Windows) pentru a incarca firmware-ul

Bibliografie/Resurse

Resurse Hardware

- Datasheet ATmega32U4
- Datasheet MAX7219
- Documentatie Arduino Leonardo

Resurse Software

- AVR Libc Reference Manual
- avrdude (flashing tool)

Export to PDF