

# Mini Game Console

## Introducere

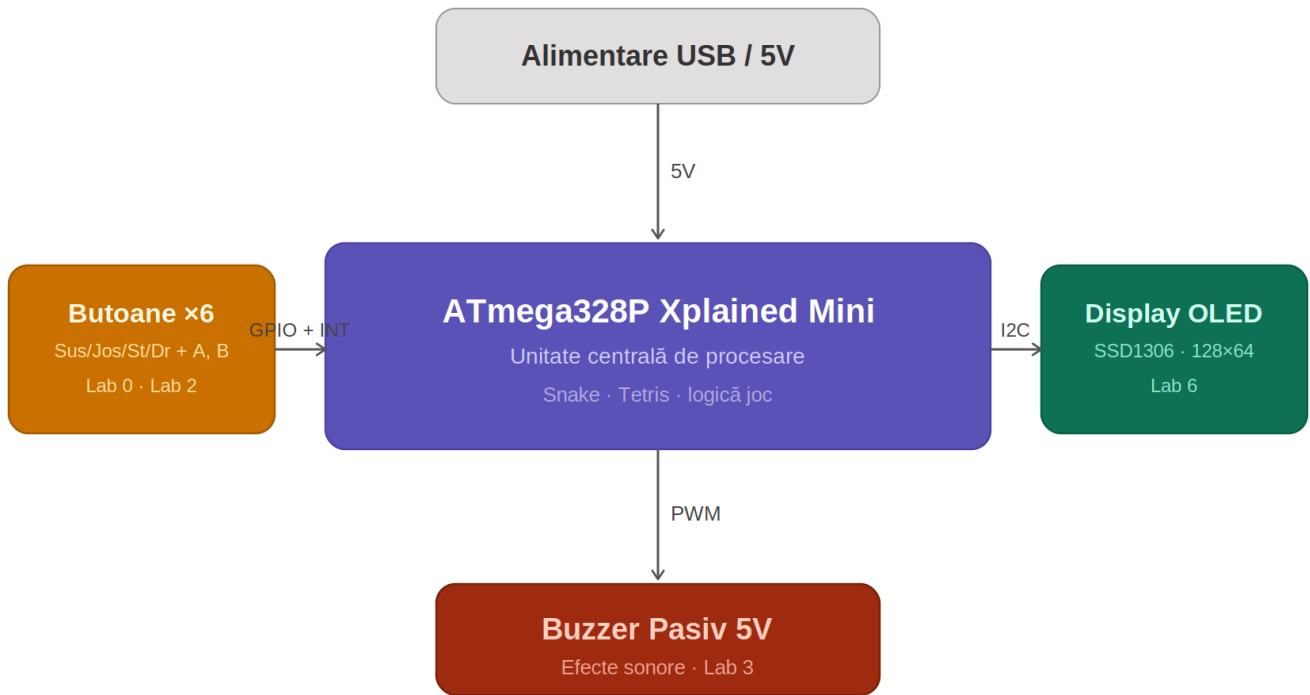
**Mini Game Console** este o consolă de jocuri portabilă de dimensiuni reduse, construită în jurul microcontrollerului ATmega328P Xplained Mini.

- **Ce face:** Dispozitivul rulează două jocuri clasice — **Snake** și **Tetris** — afișate pe un ecran OLED de 0.96". Jucătorul controlează jocul prin șase butoane fizice (direcții + acțiuni), iar efectele sonore sunt generate printr-un buzzer pasiv controlat prin PWM.
- **Scopul proiectului:** Realizarea unui dispozitiv de joacă embedded complet funcțional, care să integreze conceptele fundamentale studiate la laboratoarele de Proiectare cu Microprocesoare: GPIO, întreruperi, timere și comunicație I2C.
- **Ideea de pornire:** Proiectul a pornit din dorința de a combina nostalgia jocurilor clasice cu provocarea tehnică a programării low-level pe AVR, demonstrând că un microcontroller simplu poate rula aplicații interactive în timp real.
- **Utilitate:** Proiectul demonstrează în mod practic modul în care componentele hardware și software interacționează într-un sistem embedded complet — de la citirea intrărilor fizice până la afișarea graficii pe ecran și generarea sunetului.

## Descriere generală

Schema bloc a proiectului este prezentată mai jos. Microcontrollerul ATmega328P Xplained Mini reprezintă unitatea centrală care coordonează toate celelalte module.

**Schemă bloc:**



Mini Game Console — ATmega328P Xplained Mini

### Module și interacțiuni:

- **ATmega328P Xplained Mini** — unitatea centrală de procesare. Rulează logica jocurilor (Snake și Tetris), gestionează intrările de la butoane, actualizează afișajul și controlează buzzerul.
- **Display OLED SSD1306 (128×64)** — afișează starea curentă a jocului (grila, scorul, meniurile). Comunicația se face prin protocolul **I2C** (pini SDA și SCL).
- **Butoane x6** — permit controlul jocului: Sus, Jos, Stânga, Dreapta (direcție) și A, B (acțiune/pauză). Apăsarea butoanelor este detectată prin **întreruperi externe**, nu prin polling.
- **Buzzer Pasiv 5V** — generează efecte sonore (mâncare colectată în Snake, plasare piesă în Tetris, game over). Controlat printr-un semnal **PWM** generat de Timer.

### Laboratoare utilizate:

- **Lab 0 — GPIO** — configurarea pinilor pentru butoane (INPUT cu pull-up intern) și pentru buzzer (OUTPUT)
- **Lab 2 — Întreruperi** — detectarea apăsării butoanelor prin întreruperi externe (INT0, INT1) și pin-change interrupts (PCINT), eliminând necesitatea polling-ului continuu
- **Lab 3 — Timere și PWM** — Timer1 pentru game loop la frecvență fixă (~10 FPS), Timer2 pentru generarea semnalului PWM către buzzer (frecvențe diferite pentru sunete diferite)
- **Lab 6 — I2C** — comunicarea cu display-ul OLED SSD1306 prin protocolul I2C pentru afișarea graficii jocului

## Hardware Design

## Listă de piese

Nr.	Componentă	Cantitate	Rol în proiect
1	ATmega328P Xplained Mini	1 buc.	Microcontrollerul principal, creierul întregului sistem
2	Ecran OLED SSD1306 0.96" (I2C, 128x64)	1 buc.	Afișarea jocului (Snake, Tetris), scorului și meniurilor
3	Butoane tactile PCB 6x6x5mm	6 buc.	Control joc: Sus, Jos, Stânga, Dreapta, A (acțiune), B (pauză)
4	Buzzer pasiv 5V	1 buc.	Efecte sonore: mâncare, plasare piesă, game over
5	Rezistențe 10kΩ	6 buc.	Pull-down pentru butoane (opțional — se poate folosi pull-up intern)
6	Rezistență 100Ω	1 buc.	Limitare curent buzzer
7	Condensatori 100nF	2 buc.	Decuplare alimentare (stabilitate tensiune)
8	Breadboard 830 puncte	1 buc.	Prototipare circuit fără lipire
9	Fire jumper M-M	1 set	Conexiuni între componente pe breadboard
10	Fire jumper M-F	1 set	Conexiuni de la placă la module externe
11	Cablu Micro-USB	1 buc.	Alimentare 5V și programare ATmega de la laptop

## Conexiuni hardware

### Display OLED SSD1306 → ATmega328P:

- SDA → PC4 (SDA, pin I2C)
- SCL → PC5 (SCL, pin I2C)
- VCC → 3.3V / 5V
- GND → GND

### Butoane → ATmega328P:

- Buton Sus → PD2 (INT0)
- Buton Jos → PD3 (INT1)
- Buton Stânga → PB0 (PCINT0)
- Buton Dreapta → PB1 (PCINT1)
- Buton A → PB2 (PCINT2)
- Buton B → PB3 (PCINT3)

### Buzzer Pasiv → ATmega328P:

- Pin + → PB3 (OC2A — ieșire PWM Timer2)
- Pin - → GND (prin rezistență 100Ω)

## Software Design

### Mediu de dezvoltare

- **IDE:** Microchip Studio (fost Atmel Studio) / VS Code cu extensie AVR
- **Compiler:** avr-gcc (AVR GNU Toolchain)
- **Programare:** AVRDUDE prin bootloader USB al plăcii Xplained Mini
- **Limbaj:** C (programare la nivel de registre, fără Arduino)

## Librării și surse 3rd-party

- **avr/io.h** — acces la registrele ATmega328P
- **avr/interrupt.h** — gestionarea întreruperilor
- **util/delay.h** — delay-uri software (folosit minim, doar la inițializare)
- **Librărie I2C custom** (bazată pe laboratorul 6) — comunicație cu OLED
- **Librărie SSD1306** (adaptată pentru AVR, fără Arduino) — desenare pixeli, caractere și forme pe OLED

## Algoritmi și structuri planificate

### Structura generală a firmware-ului:

1. **Inițializare** — configurare GPIO, I2C, Timer1 (game loop), Timer2 (PWM buzzer), activare întreruperi globale
2. **Meniu principal** — selecție joc (Snake / Tetris) prin butoane, afișat pe OLED
3. **Game loop** — declanșat de Timer1 la ~100ms (10 FPS): citire stare butoane → actualizare logică joc → redesenare ecran
4. **ISR butoane** — rutine de întrerupere pentru fiecare buton, cu debounce software (~20ms)
5. **ISR Timer2** — generare ton PWM pentru buzzer (frecvență variabilă în funcție de eveniment)

### Joc Snake:

- Șarpele reprezentat ca listă circulară de coordonate (cap + coadă)
- Generare aleatoare a poziției mâncării (folosind valoarea Timer ca seed)
- Detecție coliziune cu pereții și cu propriul corp
- Creștere viteză pe măsură ce scorul crește (reducere perioadă Timer1)
- Afișare scor curent în colțul ecranului OLED

### Joc Tetris:

- Grila de joc 10×16 celule, stocată ca matrice de biți
- 7 tipuri de piese (tetromino) cu rotație în 4 poziții
- Cădere automată controlată de Timer1, cădere rapidă prin buton
- Detecție și ștergere linii complete
- Afișare piesă următoare și scor


## Rezultate Obținute

Această secțiune va fi completată după finalizarea proiectului.

## Concluzii

Această secțiune va fi completată după finalizarea proiectului.

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## Jurnal

- **Săptămâna 1:** Alegerea temei, documentare componente, comandă piese

## Bibliografie/Resurse

### Resurse Hardware

### Resurse Software

- [Laborator 0 — GPIO](#)
- [Laborator 2 — Întreruperi](#)
- [Laborator 3 — Timere și PWM](#)
- [Laborator 6 — I2C](#)

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/david\\_gabriel.rusu](http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/david_gabriel.rusu)



Last update: **2026/05/09 21:00**