

# Documentație Proiect

## Joc "Stack" controlat de Arduino

Tutelea Aurel Emanuel

25 ianuarie 2026

### Rezumat

Acest document prezintă implementarea unui joc de tip "Stack", realizat folosind platforma Arduino. Proiectul integrează matrice LED pentru afișaj grafic al blocurilor, un ecran LCD pentru interfață, cât și feedback auditiv.

### Cuprins

<b>1</b>	<b>Descrierea Proiectului</b>	<b>2</b>
<b>2</b>	<b>Componente Hardware</b>	<b>2</b>
<b>3</b>	<b>Funcționalități și Implementare Software</b>	<b>3</b>
3.1	Biblioteci Utilizate . . . . .	3
3.2	Interacțiunea cu Utilizatorul și Controlul . . . . .	3
3.3	Algoritmul de "Tăiere" (Chopping Logic) . . . . .	3
3.4	Implementarea Modulului Infinit . . . . .	4
3.5	Sistemul de Feedback . . . . .	4
<b>4</b>	<b>Schema Electrică</b>	<b>5</b>
<b>5</b>	<b>Bibliografie și Resurse</b>	<b>6</b>

# 1 Descrierea Proiectului

Jucătorul trebuie să clădească un turn din blocuri luminoase pe o serie de matrice LED suprapuse. Obiectivul este suprapunerea perfectă a blocurilor aflate în mișcare peste cele anterioare. Orice eroare de aliniere duce la „tăierea” blocului (pierderea LED-urilor care nu se suprapun), iar jocul continuă cu piese tot mai mici până la atingerea vârfului (Victorie) sau pierderea completă a blocului (Game Over).

## 2 Componente Hardware

Pentru realizarea acestui sistem au fost utilizate următoarele componente:

- **Placă de dezvoltare:** Arduino Uno
- **Afișaj Grafic:** 4 x Module Matrice LED 8x8 (legate în cascadă).
- **Afișaj Text:** LCD cu adaptor I2C (pentru afișarea scorului și a timpului).
- **Sunet:** Buzzer Activ (Low Level Trigger - declanșare pe nivel logic 0).
- **Interacțiune:** 2 x Butoane (Push buttons) pentru controlul jocului și al meniului.
- **Conectică:** Breadboard și fire.
- **Alimentare:** Sursă externă 5V sau USB.

## 3 Funcționalități și Implementare Software

Codul este organizat în funcții separate pentru a fi ușor de înțeles. Pentru temporizare (viteză), folosim funcția `millis()` în loc de `delay()`. Aceasta permite ca jocul să răspundă instantaneu la apăsarea butoanelor, chiar și în timp ce blocurile se mișcă.

### 3.1 Biblioteci Utilizate

Proiectul folosește biblioteci standard pentru a comunica ușor cu componentele:

- **LedControl.h:** Folosită pentru controlul matricelor de LED-uri (aprinderea și stingerea punctelor luminoase).
- **LiquidCrystal\_I2C.h:** Folosită pentru afișarea textului (meniu, scor, timp) pe ecranul LCD.
- **Wire.h:** Bibliotecă standard necesară pentru comunicarea cu ecranul LCD.

### 3.2 Interacțiunea cu Utilizatorul și Controlul

Sistemul de control este implementat folosind două butoane digitale, fiecare având un rol specific în funcție de starea jocului (Meniu sau Joc Activ):

- **Butonul Stânga (Pin 8 - Selecție):** Acest buton este activ doar în meniul principal. Are rolul de a naviga ciclic prin cele 4 niveluri de dificultate disponibile. La fiecare apăsare, cursorul de pe LCD se mută la următoarea opțiune (1, 2, 3 sau Infinit).
- **Butonul Dreapta (Pin 9 - Start/Acțiune):** Acesta este butonul principal și are dublă funcționalitate:
  1. *În Meniu:* Confirmă nivelul selectat și pornește jocul efectiv.
  2. *În Timpul Jocului:* Comandă plasarea blocului. La apăsarea sa, blocul aflat în mișcare se oprește instantaneu pe poziția curentă pentru a fi verificat de algoritmul de tăiere.

### 3.3 Algoritmul de "Tăiere" (Chopping Logic)

Aceasta este componenta centrală a jocului, implementată matematic prin intersecția vectorilor. La apăsarea butonului, sistemul compară poziția blocului curent cu cea a rândului anterior (referința).

Procesul logic este următorul:

1. **Calculul limitelor:** Se determină punctele de start și de final ale suprapunerii folosind funcțiile de minim și maxim:

$$Start_{nou} = \max(Start_{curent}, Start_{anterior})$$

$$Stop_{nou} = \min(Stop_{curent}, Stop_{anterior})$$

2. **Validarea:** Se calculează lățimea noii piese:

$$Latime = Stop_{nou} - Start_{nou}$$

Dacă  $Latime \leq 0$ , înseamnă că blocurile nu se ating, rezultând în **Game Over**.

3. **Actualizarea stării:** Dacă există suprapunere, LED-urile din afara intervalului calculat sunt stinse logic și vizual. Noua lățime devine referința pentru nivelul următor.

### 3.4 Implementarea Modului Infinit

Pentru a depăși limita fizică a celor 4 matrice, s-a implementat o logică de resetare ciclică a stivei.

- Când jucătorul atinge vârful turnului (Modulul 3, Rândul 0), ecranul este curățat complet (`lc.clearDisplay`).
- Blocul rezultat din ultima tăiere este "copiat" automat la baza primei matrice (Rândul 7, Modulul 0).
- Variabila de scor este actualizată, viteza crește (timpul de reacție scade cu 10ms), iar jocul continuă fără întrerupere, simulând un turn infinit.

### 3.5 Sistemul de Feedback

- **Audio:** Deoarece buzzer-ul este de tip *Low Level Trigger*, funcțiile de sunet inversează logica standard: semnalul LOW activează membrana piezoelectrică, iar HIGH o dezactivează.
- **Scor și Timp:** Scorul este calculat prin sumarea lățimii blocurilor plasate cu succes (`scor += blocLatime`), iar durata jocului este determinată la final prin diferența  $T_{final} - T_{start}$ .

## 4 Schema Electrică

Conexiunile dintre componente și placa Arduino sunt realizate astfel:

### Matrice LED :

VCC → 5V, GND → GND, DIN → Pin 12, CLK → Pin 11, CS → Pin 10.

### LCD 1602 (I2C):

VCC → 5V, GND → GND, SDA → Pin A4, SCL → Pin A5.

### Butoane:

Configurate cu rezistență internă (INPUT\_PULLUP).

- Buton Selectare Nivel → Pin 8
- Buton Start/Acțiune → Pin 9

### Buzzer:

Pin 7 (Logică inversă: LOW = Activ).

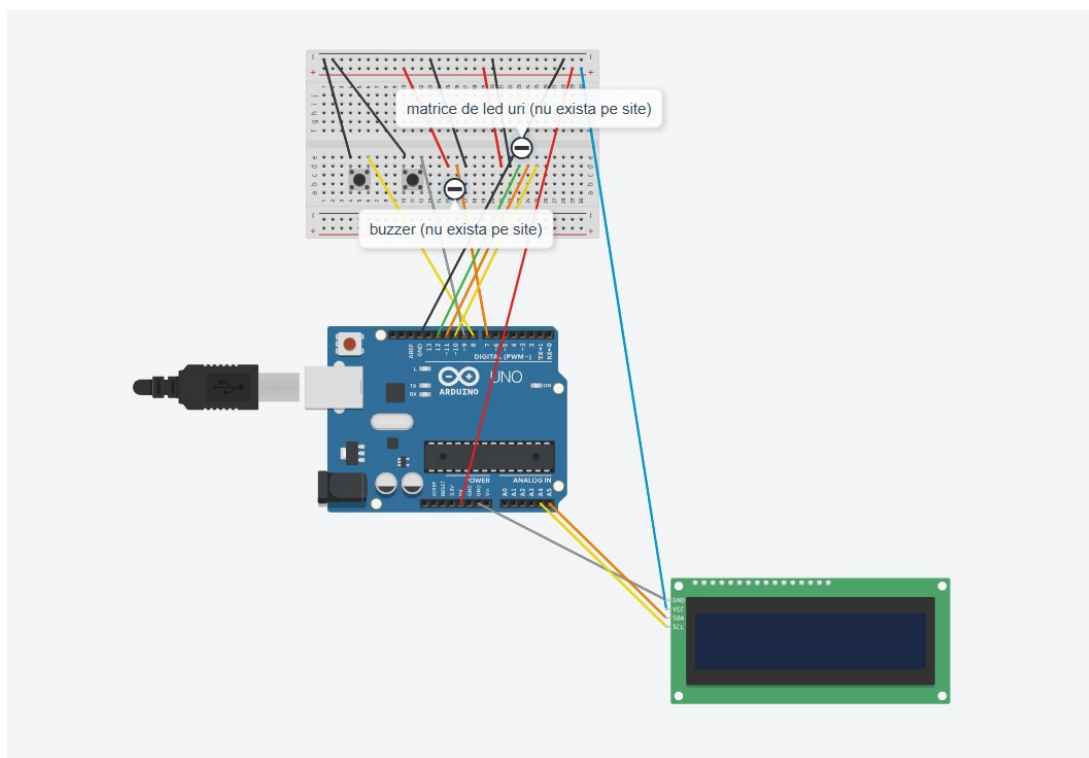


Figura 1: Diagrama de conexiuni a proiectului

## 5 Bibliografie și Resurse

Pentru realizarea acestui proiect au fost consultate următoarele surse:

1. **Controlul Matricei LED:**

Tutorial și documentație pentru biblioteca LedControl:

<https://www.makerguides.com/max7219-led-dot-matrix-display-arduino-tutorial/>

2. **Resurse Academice:**

Cursuri și materialele de laborator pentru restul componentelor.

3. **Schema electrică:**

<https://www.tinkercad.com/dashboard/tutorials/>

4. **Asistență:**

*Google Gemini* – Utilizat pentru idei și pentru generarea formatului pentru documentația în format LaTeX.