

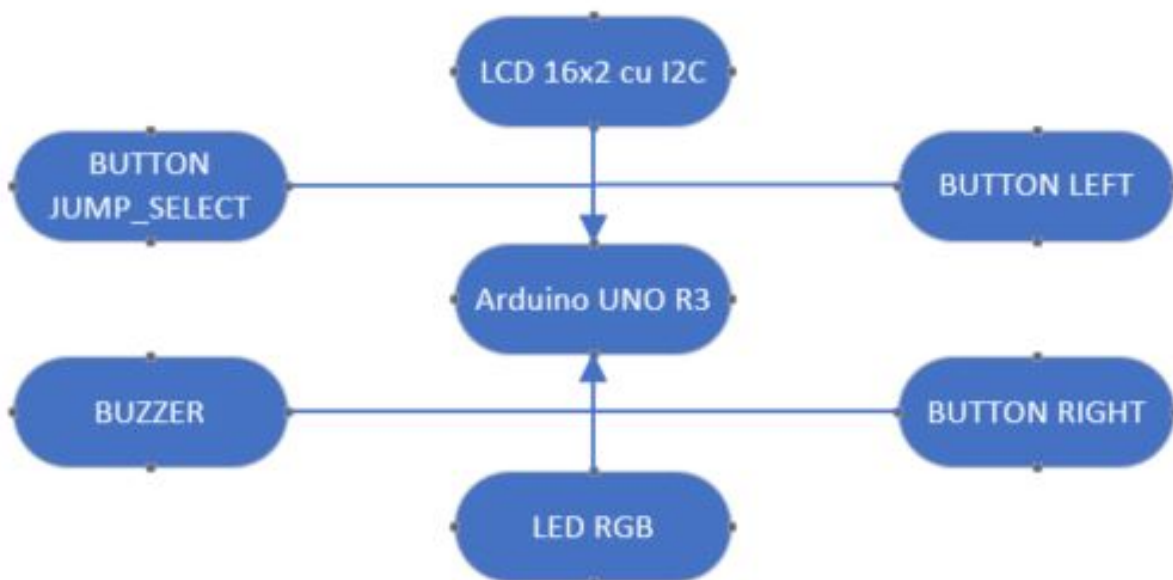
Pixel Runner

Introducere

Pixel Runner este un joc de tip runner inspirat de T-Rex Dino din Chrome, realizat pe Arduino cu LCD, LED RGB, buzzer si mai multe butoane, care testeaza reflexele si atentia utilizatorului prin sarituri peste cactusi ce apar aleatoriu. Ideea proiectului a pornit din dorinta de a transpune o experienta clasica de gaming pe un dispozitiv fizic, dezvoltand atat abilitatile de programare embedded, cat si cele de lucru cu hardware-ul. Jocul este util si distractiv, oferind un mod practic de a invata electronica si programarea, fiind totodata o platforma buna pentru experimente educationale sau extinderi creative de tip meniu, niveluri de dificultate si feedback audio-vizual.

Descriere generala

Implementare joc Pixel Runner pe LCD cu Arduino.



Proiectul folosește o placă Arduino UNO, un afișaj LCD, LED RGB, buzzer și trei butoane pentru control. Utilizatorul poate selecta nivelul de dificultate și controlează un dinozaur care trebuie să sară peste obstacole. LED-ul oferă feedback vizual, iar buzzer-ul redă efecte sonore și melodii de joc. Scorul curent și cel maxim sunt afișate pe ecran, iar meniul interactiv permite navigarea ușoară. Jocul este atât distractiv, cât și educativ, ajutând la dezvoltarea abilităților de programare și lucru cu hardware-ul Arduino.

Hardware Design

Stadiul Actual al Implementarii Circuitul hardware este finalizat si complet functional. Toate componentele din lista de piese au fost interconectate pe un breadboard si conectate la placa de dezvoltare Arduino UNO. Montajul respecta schema electrica si este capabil sa ruleze logica jocului "Pixel Runner", raspunzand la comenzi si afisand elementele vizuale si auditive conform codului.

Componente Folosite si Rolul Lor:

- Arduino UNO R3: Reprezinta unitatea centrala a proiectului. Ruleaza codul sursa, proceseaza logica jocului, citeste starea butoanelor si controleaza toate componentele externe (LCD, LED, Buzzer).
- Afisaj LCD I2C 16x2: Este interfata vizuala principala. Afiseaza personajul, obstacolele, scorul, vietile ramase si meniurile de navigare. Modulul I2C simplifica mult conexiunile.
- 3 x Butoane: Sunt dispozitivele de intrare pentru jucator.
- * Buton Start/Saritura: Porneste jocul si controleaza actiunea de saritura a personajului.
- * Butoane Stanga/Dreapta: Permit navigarea prin meniurile jocului (selectare jucator, dificultate, etc.).
- LED RGB: Oferă feedback vizual colorat despre starea jocului:
- * Verde: Joc in desfasurare normala.
- * Albastru: Personajul este in timpul unei sarituri.
- * Rosu: Sfarsitul jocului (Game Over).
- Buzzer Pasiv: Componenta audio a proiectului. Genereaza efecte sonore (saritura, coliziune) si melodia din timpul jocului si de la finalul acestuia.
- Rezistenta 1.0kΩ: Este folosita ca rezistenta de limitare a curentului pentru a proteja pinii digitali ai placii Arduino, de obicei in serie cu LED-ul sau buzzer-ul.

Descrierea Detaliata a Pinilor Afisaj LCD (I2C):

- Pin SDA → Pin A4 (Arduino): Linia de date (Serial Data).
- Pin SCL → Pin A5 (Arduino): Linia de ceas (Serial Clock).
- Motivatie: A4 si A5 sunt pinii hardware dedicati pentru comunicatia I2C pe placa Arduino Uno. Utilizarea I2C reduce numarul de pini necesari pentru controlul ecranului de la peste 6 la doar 2.

Butoane:

- Start/Saritura → Pin Digital 2:
- Motivatie: Pinul 2 este unul dintre pinii de intrerupere externa pe Arduino Uno. Folosirea unei intreruperi (attachInterrupt) asigura detectarea apasarii butonului instantaneu, fara intarziere, ceea ce este critic pentru o actiune de saritura precisa.
- Stanga → Pin Digital 3:
- Dreapta → Pin Digital 4:
- Motivatie: Acesti pini digitali sunt folositi ca intrari simple. Sunt configurati ca INPUT_PULLUP, eliminand necesitatea rezistentelor externe pentru a mentine o stare logica stabila.

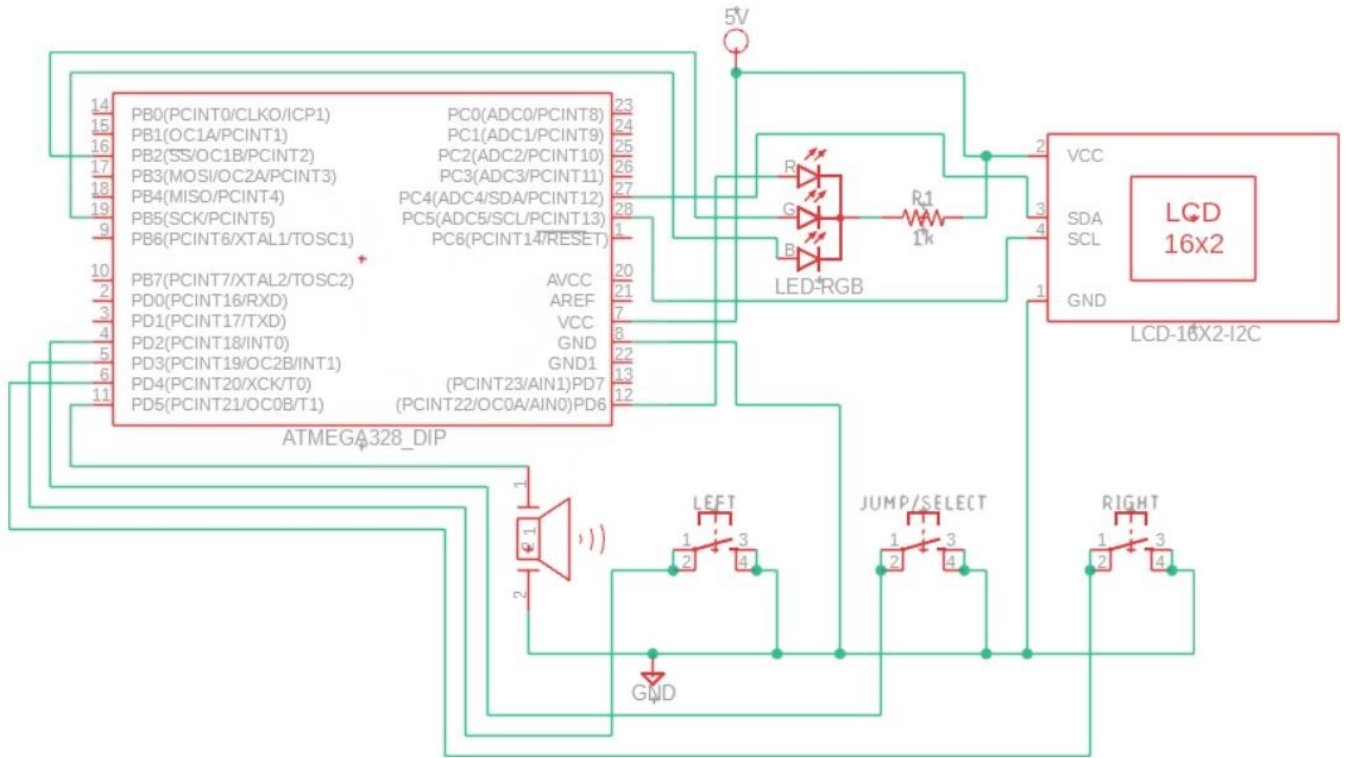
LED RGB:

- Pin Rosu → Pin Digital 9:
- Pin Verde → Pin Digital 10:
- Pin Albastru → Pin Digital 11:
- Motivatie: Pinii 9, 10 si 11 sunt pini PWM (Pulse Width Modulation), necesari pentru a controla intensitatea fiecărei culori prin functia analogWrite.

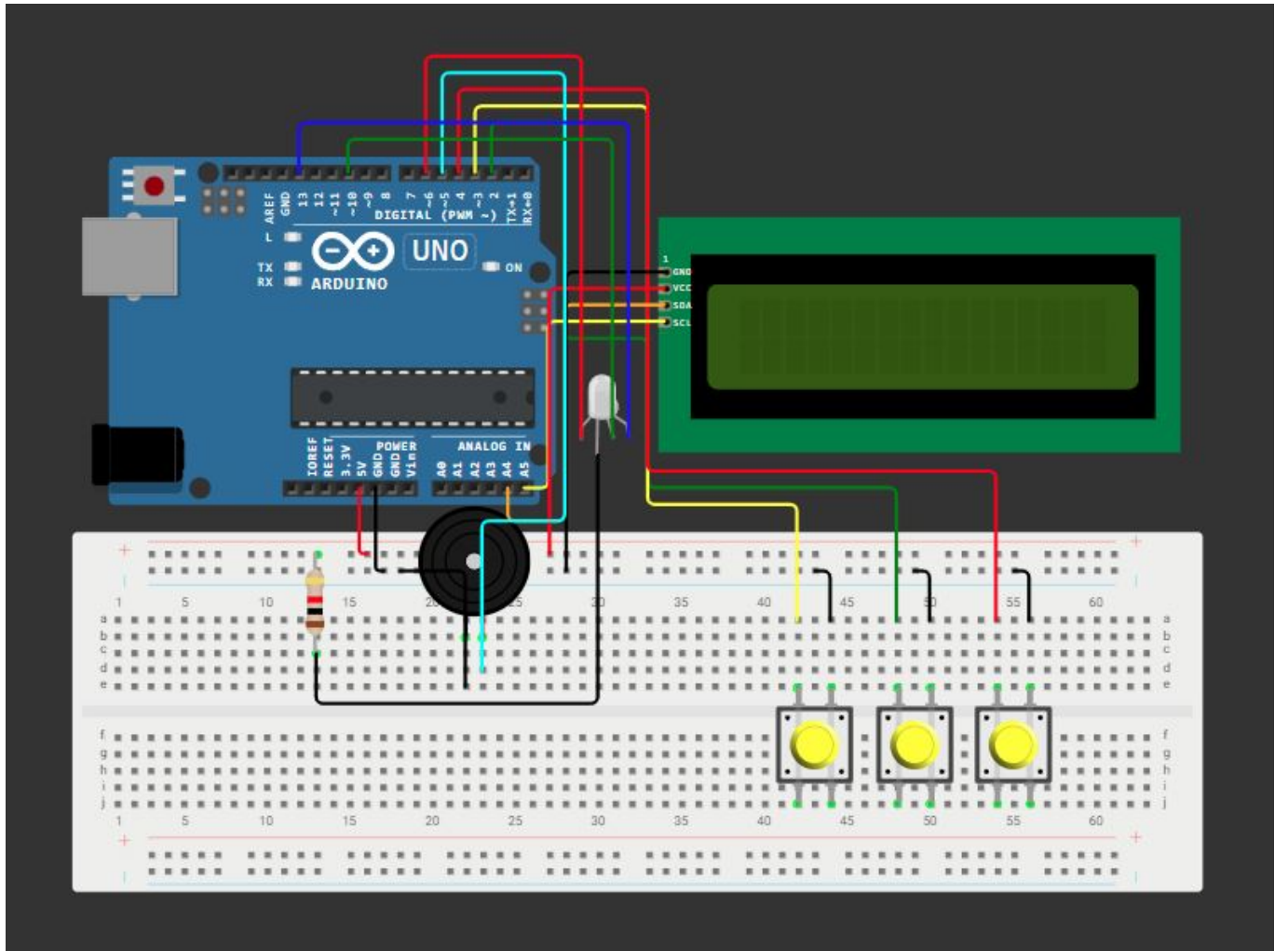
Buzzer Pasiv:

- lesire → Pin Digital 5:
- Motivatie: Pinul 5 este un pin PWM, ceea ce il face ideal pentru a genera tonuri de diferite frecvente si durate folosind functia tone(), necesara pentru melodiile si efectele sonore din joc.

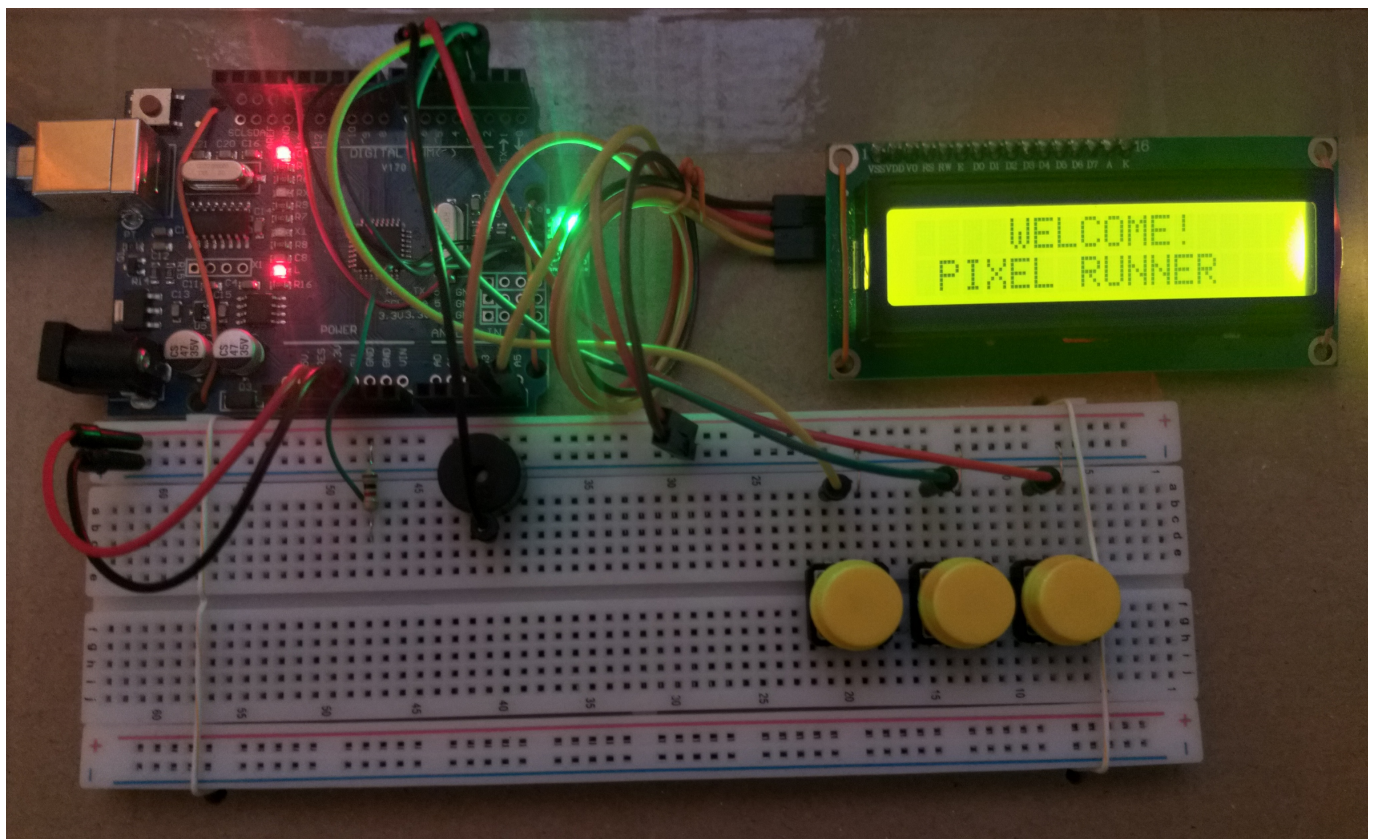
Schema electrica:



Circuit design:



Hardware:



Software Design

Stadiul Actual al Implementarii Software: Implementarea software, realizata in mediul de dezvoltare Arduino IDE, este completa, testata si stabila. Toate functionalitatile si algoritmi descriși in proiect sunt implementati si functioneaza conform asteptarilor, de la meniul interactiv si mecanicile de joc, pana la sistemul de salvare a scorurilor si modul de economisire a energiei.

Motivarea Alegerii Bibliotecilor Pentru proiect, am folosit doua biblioteci esentiale, alese strategic pentru eficienta:

- LiquidCrystal_I2C.h: Am ales aceasta biblioteca pentru a simplifica drastic conexiunile hardware. Prin protocolul I2C, controlul afisajului se realizeaza cu doar doi pini de date (SDA si SCL), eliberand ceilalti pini ai microcontrolerului pentru alte componente, cum ar fi butoanele sau LED-ul.
- EEPROM.h: Aceasta biblioteca standard a fost cruciala pentru a implementa o functionalitate cheie: persistenta datelor. Prin utilizarea memoriei non-volatile, scorurile maxime ale jucatorilor sunt salvate permanent, chiar si dupa intreruperea alimentarii, oferind o experienta de joc completa si motivanta.

Elementul de Noutate al Proiectului Elementul de noutate consta in integrarea unui set complex de sisteme software pe o platforma hardware cu resurse foarte limitate. Proiectul nu este doar un simplu joc, ci o aplicatie completa care demonstreaza eficienta software prin imbinarea urmatoarelor sisteme avansate:

- O masina de stari pentru gestionarea meniurilor.
- Un algoritm dinamic pentru generarea obstacolelor, care se adapteaza la dificultate.
- Un sistem de animare a caracterelor prin alternarea de frame-uri.
- Managementul scorurilor pentru multiple profiluri de jucatori.
- Un mecanism de economisire a energiei (sleep mode).
- Scheletul Proiectului, Interactiune si Validare

Scheletul si Interactiunea Functionalitatilor Fluxul software este structurat logic pentru a asigura claritate si eficienta:

- setup(): La pornire, aceasta functie ruleaza o singura data pentru a initializa componentele hardware (pini, LCD), a incarca scorurile din EEPROM si a afisa meniul principal.
- Functiile de Meniu (showMainMenu, selectPlayer, etc.): Acestea actioneaza ca o masina de stari. Programul ramane in aceasta bucla de navigare, permitand utilizatorului sa configureze jocul. Setarile alese (dificultate, personaj) sunt stocate in variabile globale.
- loop(): Odata ce jocul este pornit din meniu, bucla principala devine activa. Aceasta constituie motorul de joc, unde se gestioneaza in timp real miscarea personajului, generarea obstacolelor, detectarea coliziunilor si actualizarea scorului, folosind variabilele setate anterior.
- onJumpButtonPress(): Aceasta functie de intrerupere ruleaza independent de loop(). Ea detecteaza instantaneu apasarea butonului de salt si seteaza un indicator (flag), pe care bucla loop() il verifica si actioneaza imediat, asigurand un raspuns rapid.

Validarea Implementarii Validarea a fost realizata in doua etape:

- Testare modulara: Fiecare functie a fost testata individual pentru a-i verifica corectitudinea (ex: saveHighScore salveaza corect, resetGame reseteaza toate variabilele).

- Testare integrata: Am rulat scenarii complete de utilizare – navigare prin toate meniurile, jucarea unei partide pe fiecare nivel de dificultate, atingerea starii de “Game Over” si verificarea salvarii corecte a scorului. Astfel, am confirmat ca toate modulele interactioneaza corect.

Calibrarea Elementelor de Senzorica Singurele “elemente de senzorica” sunt butoanele. Calibrarea acestora a insemnat implementarea unui mecanism de debouncing software pentru a preveni inregistrarea unor apasari multiple la o singura actionare fizica. Acest lucru a fost realizat printr-o scurta pauza (delay) adaugata in cod imediat dupa detectarea unei apasari, asigurand astfel ca fiecare apasare este contorizata o singura data, facand navigarea prin meniuri precisa si fiabila.

Optimizari Realizate Pentru a asigura o experienta de joc superioara si o utilizare eficienta a resurselor, am implementat urmatoarele optimizari:

- Reactivitate Maxima la Comenzi: Cea mai importanta optimizare este utilizarea unei intreruperi hardware (attachInterrupt) pentru butonul de salt. Astfel, comanda de salt este preluata si procesata instantaneu, eliminand orice intarziere (lag) si fiind esentiala pentru un gameplay precis.
- Eficienta Grafica si Animatie Fluida: In loc sa redesenam caracterele in bucla principala, acestea sunt create o singura data la pornire si stocate in memoria CGRAM a ecranului LCD. Animatia se realizeaza prin alternarea rapida a acestor caractere pre-incarcate. Aceasta tehnica reduce drastic incarcarea procesorului si asigura o animatie fluida, fara sacadari.
- Economisirea Energiei: Am implementat un mod sleep care, dupa un minut de inactivitate, stinge iluminarea ecranului si LED-ul RGB. Aceasta optimizare reduce semnificativ consumul de energie si este activata/dezactivata inteligent prin functiile enterSleepMode() si wakeUp().

Cod sursa: [Pixel Runner GitHub Repository](#)

Rezultate Obtinute


Am realizat cu succes un joc complet functional si stabil pe platforma Arduino, care integreaza corect toate componentele hardware. Sistemul ofera o experienta de joc fluida, cu meniuri intuitive, scoruri salvate in EEPROM si un mod de economisire a energiei.

Concluzii

Acest proiect a demonstrat integrarea practica si eficienta a tehnologiilor studiate (GPIO, I2C, PWM, Intreruperi) intr-o aplicatie reala. Codul rezultat este modular si robust, indeplinind toate cerintele initiale si constituind o baza solida pentru dezvoltari viitoare.

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**.

Bibliografie/Resurse

Proiectul utilizează concepte studiate în laborator, precum GPIO (interfatarea cu butoane și LED), I2C (comunicarea cu LCD-ul), întreruperi hardware (pentru butonul de salt), PWM (pentru controlul ledului RGB și buzzer), precum și timere software (gestionarea timpului de joc și a animațiilor) – integrând astfel cerințele pentru laboratoare PM.

<https://docs.arduino.cc/learn/built-in-libraries/eeprom/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/mmahmoudi> 

Last update: **2025/05/24 07:58**