

Aceasta este structura recomandată pentru documentația de proiect de pe paginile voastre de wiki. Este o structură minimală, sunteți liberi să adăugați capitole/subcapitole dacă considerați că este nevoie. Vă rugăm ca toată documentația să fie scrisă în limba **Engleză**

Bop IT

Introducere

BopIt este un joc interactiv conceput pentru a testa și antrena reflexele și viteza de reacție. Mecanica jocului presupune executarea rapidă a unor acțiuni simple, precum apăsarea unui buton, rotirea unui element, imediat ce acțiunea apare pe ecran.

Scopul jocului este ca jucătorul să răspundă corect și cât mai rapid la comenzile primite. Jocul oferă două moduri de joc: unul normal, în care jucătorul are la dispoziție 5 secunde pentru a realiza orice acțiune, și un mod rapid, în care jucătorul are 10 secunde pentru a efectua cât mai multe acțiuni posibil.

Descriere generală

Proiectul este dezvoltat pe o placuta Arduino Uno, la care sunt conectate toate componentele necesare. Atunci când butonul de start este apăsător, pe ecranul LCD este afisată acțiunea pe care jucătorul trebuie să o realizeze, împreună cu scorul curent.

Pe un afișaj cu 7 segmente este indicat timpul rămas pentru a acționa- acesta începe de la 5 secunde pentru modul normal și 10 pentru modul rapid

Imediat ce o acțiune este afisată pe LCD, jucătorul trebuie să o îndeplinească corect și în intervalul de timp alocat. Dacă nu reușește, pierde runda, lucru semnalizat printr-un sunet emis de buzzer.



Hardware Design

Hardware:

- Placuta ARDUINO UNO R3
- 2 butoane
- LCD
- 7 segment display
- Potentiometru
- 8-bit shift register 74HC595

- Passive Buzzer

Schema electrica:



Schema hardware:



Descriere Hardware

placă de dezvoltare compatibilă cu Arduino UNO R3:

- 5V (folosit pentru alimentarea componentelor care necesită tensiune constantă)
- GND (comun tuturor componentelor, asigurând referința de tensiune)

LCD 16x2 (cu modul I2C): display pentru afișarea meniului

- SDA (Serial Data) -> A4
 - linia de date pentru protocolul I2C (permite transferul datelor între Arduino și LCD)
- SCL (Serial Clock) -> A5:
 - linia de ceas pentru transmiterea sincronizată a datelor între dispozitivele I2C

2 x (push-button + rezistență): implementarea de comenzi pentru meniu

- conexiune la pini digitali (A1-A2):
 - folosiți pentru a detecta apăsările de butoane
- folosire de rezistențe 10kΩ:
 - folosite ca pull-down pentru a menține starea LOW în absența apăsării

Passive buzzer: sunete trist la pierderea jocului

- conexiune la pin digital (D8):
 - controlul sunetului prin generarea de tonuri cu funcția `tone()` din Arduino

Potentiometru: folosit pe post de rotita in joc

- conexiune la pin digital (D8)

7 segment display: Folosit pentru a afisa cate secunde mai avem pentru a actiona

- display cu catod comun

74HC595: multiplexeaza

- controlează display-ul cu 7 segmente folosind doar 3 pini de la Arduino
- conexiune la pinii digitali:
 - latchPin (D10) transfera datele din registrul de deplasare catre pinii de iesire
 - clockPin (D9) puls de ceas pentru deplasarea bitilor in registru
 - dataPin (D11) transmiterea datelor bit cu bit

Mapare cu 7-segment display Q0 → Segment DP

Q1 → Segment G

Q2 → Segment F

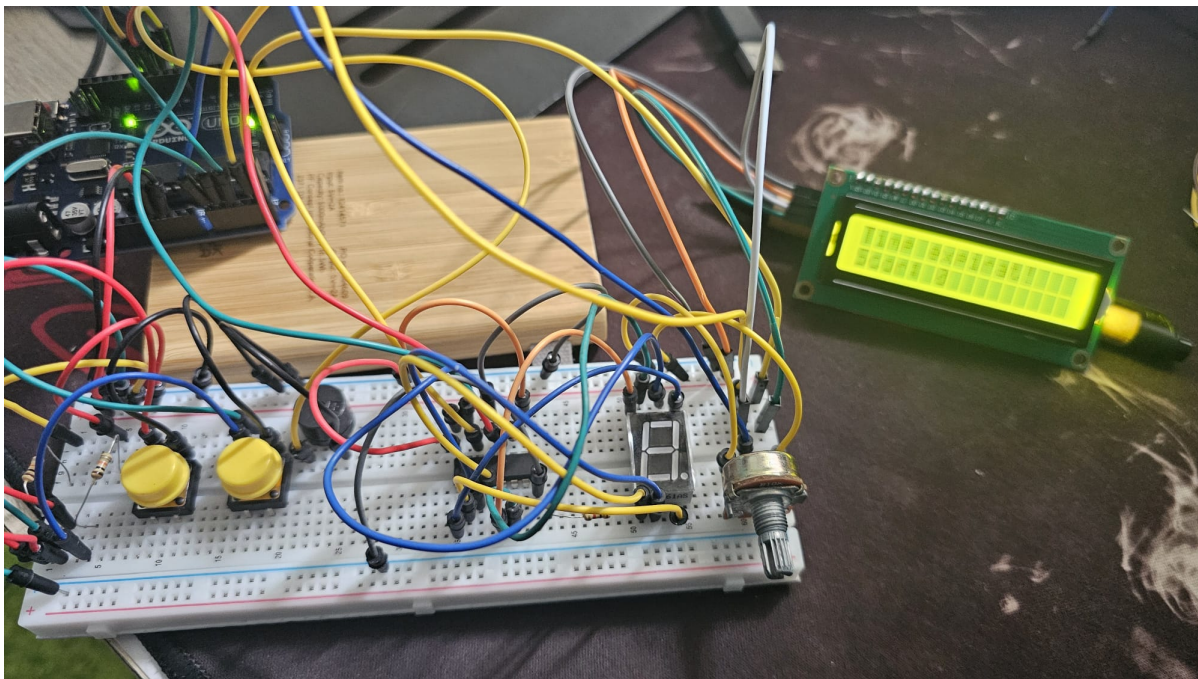
Q3 → Segment E

Q4 → Segment D

Q5 → Segment C

Q6 → Segment B

Q7 → Segment A



Software Design

Mediu de dezvoltare:

Pentru dezvoltarea proiectului, am folosit editorul de text VS Code împreună cu extensia PlatformIO, configurată pentru a utiliza placa Arduino Uno

Librarii:

Wire : Aceasta este o librerie standard pentru comunicatia I2C pe Arduino, folosita pentru a interactiona cu dispozitivele conectate pe magistrala I2C.

LiquidCrystal_I2C : O librerie pentru controlul afisajelor LCD prin intermediul protocolului I2C, oferind functionalitati pentru setarea cursorului, stergerea ecranului si afisarea de text. Am folosit-o pentru a afisa diverse mesaje pe ecranul LCD.

Algoritm: Codul este impartit in 3 bucati

- partea de inițializare a datelor constante
- main-ul pornirea și inițializare
- funcții de control al jocului

Partea de inițializare include:

- Configurarea pinilor pentru senzori, butoane și afișaje
- Pattern-urile pentru afișajul cu 7 segmente
- Enum-urile pentru stările jocului
- Variabilele globale pentru scor, timpi, etc.

enum GameState {

```
IDLE,  
PLAYING,
```

```
}
```

Main-ul (setup și loop):

- Inițializarea hardware-ului (LCD, pini)
- Bucla principală care gestionează stările jocului prin switch-case
- Starea IDLE:
 1. afișează meniul de alegere pe LCD
 2. așteapta alegerea modului de joc
 3. resetează scorul și trece în starea PLAYING
- Starea PLAYING:
 1. afiseaza numaratoarea inversa pe display-ul cu 7-segmente
 2. verifică expirarea timpului
 3. detectează acțiunile greșite
 4. alege o noua actiune

Funcții:

- showNumber(int num)
 1. afișează cifrele(0-9) pe display-ul cu 7-segmente
- checkAction(int action)
 1. verifică dacă utilizatorul a executat acțiunea aleasa
 2. pentru butoane: folosește funcția buttonPressed()
 3. pentru potențiometrul verifică rotirea completă (de la minim la maxim sau invers)
- buttonPressed(int pin)

1. detectează apăsarea unui buton (detectează când butonul trece de la starea HIGH la LOW)

- wrongAction(int correctAction)

1. verifică dacă s-a executat orice altă acțiune în afară de cea corectă

- showGameOver(String msg)

1. gestionează sfârșitul jocului

2. afișează mesajul final și scorul

3. stinge display-ul 7-segmente

4. reda sunet de pierdere

- newAction()

1. generează o nouă acțiune aleatorie:

Utilizarea laboratoarelor în cadrul proiectului:

- Laboratorul 1

1. am configurat pinii butoanelor, potentiometrului și buzzer-ului cu pinMode() (INPUT / OUTPUT)

2. am citit stările pinilor butoanelor cu digitalRead() pentru a determina acțiunea

- Laboratorul 2

1. am folosit funcția tone(<pin>, <frecvență>) pentru a genera sunete

- Laboratorul 3

1. am citit valoarea potentiometrului cu analogRead()

- Laboratorul 4

1. am utilizat biblioteca LiquidCrystal_I2C.h pentru controlul ecranului LCD

COD: <https://github.com/Lex-Stud/PM/blob/main/COD.c++>

Proiect final: <https://youtu.be/vwC6Ghq6aGw>

Rezultate Obținute

Un joc interactiv și competitiv.

Demo:

Mod normal: <https://youtu.be/xgpH6ndhMmk>

Mod rapid: <https://youtu.be/3KyweplOT1g>

Concluzii

Proiectul a fost unul interesant și plăcut. M-a ajutat să înțeleg mai bine materia predată la laborator și mi-a sporit dorința de a realiza pe cont propriu și alte proiecte în viitor. Implementarea nu a fost lipsită de greutăți și probleme, dar la final am putut realiza ceea ce mi-am dorit.

Proiectul este unul bun și funcțional, însă poate fi îmbunătățit în viitor:

- pot fi introduse noi acțiuni
- se poate utiliza piese hardware mai performant și mai rezistent

Cod sursă și alte resurse ce trebuie să existe pe GitHub

Proiect: <https://github.com/Lex-Stud/ProiectPM>

Jurnal

06.05.2025 - am creat pagina si am adăugat descrierea generală a proiectului

12.05.2025 - am asamblat hardware-ul și am testat piesele

16.05.2025 - am început dezvoltarea software-ului

20.05.2025 - am obținut un proiect functional

Bibliografie/Resurse

[Export to PDF](#)

Piese:

https://www.emag.ro/kit-compatibil-uno-r3-24-componente-electronice-latauar-dianjixuexitaojianliying/pd/D16TKJYBM/?ref=profiled_categories_home_base_first_3_3&provider=rec&recid=rec_93_516904b2a52f8e38c1bce4991000bb50f4d0d298b4f93e5bb03eeddb2865640e_1745098167&scenario_ID=93

From:

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

Permanent link:

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



Last update: **2025/05/28 18:55**