

Minesweeper

Introducere

Proiectul meu constă într-o simulare a jocului Minesweeper pe un display de dimensiuni reduse, în care este inclus un buzzer pentru a marca începutul și finalul jocului în funcție de verdictul acestuia (câștigător/necâștigător).

Scopul acestui proiect este să recreez experiența clasică a jocului Minesweeper pe un dispozitiv compact, oferind posibilitatea de experimenta acest joc captivant, chiar și într-un mediu mai restrâns.

Cred că acest proiect este util pentru alții deoarece oferă o alternativă de petrecere a timpului liber și stimulează logica și gândirea strategică. Pentru mine, acest proiect reprezintă o oportunitate de a-mi dezvolta abilitățile în programare și de a oferi o soluție creativă și interactivă pentru utilizatori.

Descriere generală

1. Inițializarea jocului:

- Se inițializează matricea reprezentând tabla de joc pe placa Arduino cu dimensiunile prestabilite și se plasează un număr aleator de bombe în diverse poziții în matrice.
- Se afișează pe display un mesaj de început și se activează buzzer-ul pentru a marca începutul jocului.

2. Interacțiunea utilizatorului:

- Utilizatorul selectează o poziție din matrice, reprezentând o celulă din tabla de joc, folosindu-se de butoane (pentru selectare) și joystick (pentru mutare).
- Dacă poziția aleasă este o bombă, se declanșează finalul jocului și se activează buzzer-ul pentru a indica pierderea.
- Dacă poziția aleasă nu este o bombă, se afișează numărul de bombe adiacente (0, 1, 2, 3 etc.) și se continuă jocul.

3. Descoperirea zonei:

- Dacă poziția selectată are valoarea 0 (adică nu are bombe adiacente), se descoperă automat o zonă de celule conectate care nu conțin bombe, extinzând astfel aria jocului.
- Pentru aceste celule, se afișează numărul de bombe adiacente sau, în cazul în care sunt și ele 0, se continuă descoperirea zonei.

4. Verificarea câștigului:

- După fiecare selectare a unei poziții, se verifică dacă toate celulele fără bombe au fost descoperite.

Dacă da, jocul este câștigat.

- În cazul câștigului, se declanșează finalul jocului și se activează buzzer-ul pentru a indica câștigul.

5. Finalizarea jocului:

- Dacă utilizatorul a selectat o poziție cu bombă sau a descoperit toate celulele fără bombe, jocul se încheie.
- Se afișează un mesaj corespunzător pe display și se dezactivează buzzer-ul pentru a marca sfârșitul jocului.



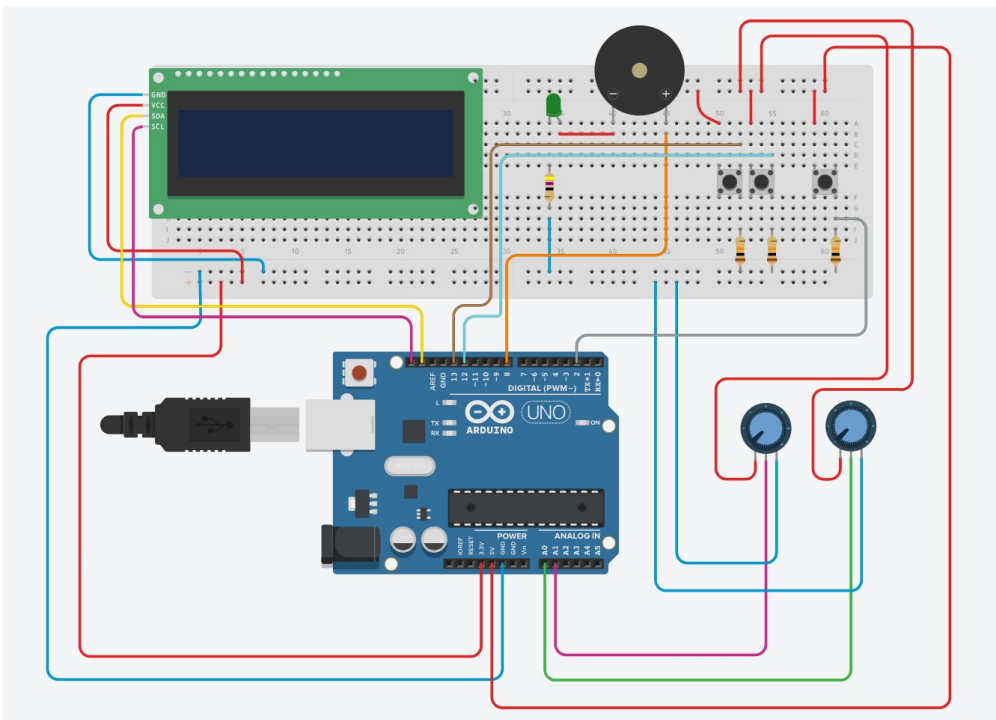
Hardware Design

Listă componente necesare:

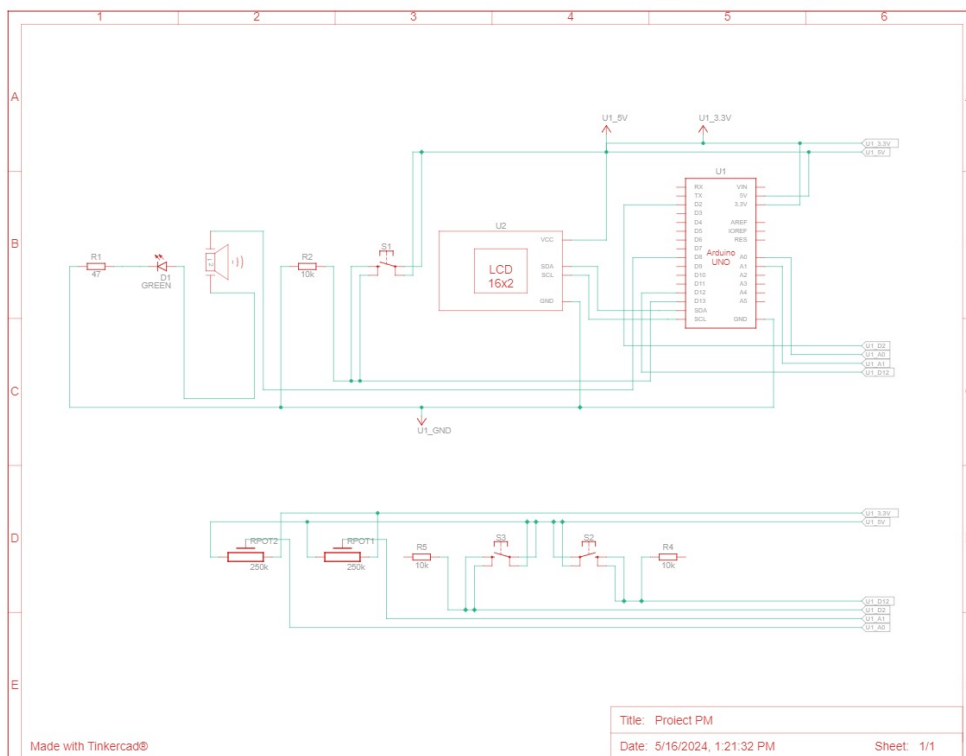
- Arduino UNO (sau echivalent) x 1
- Display (I2C) x 1
- Buzzer x 1
- LED-uri x 2
- Butoane x 4
- Joystick analog x 1
- Cabluri
- Breadboard x 1
- Rezistori

În imaginile ce urmează, am folosit 2 potențiometre în locul joystick-ului care se ocupă de mișcările pe sistemul de axe xOy.

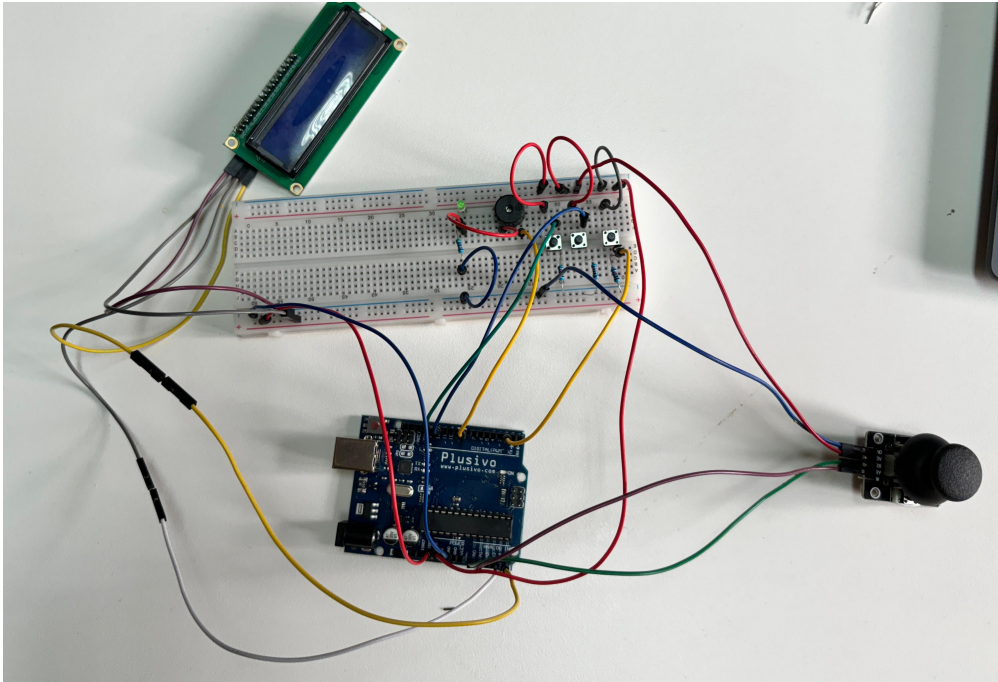
Model Tinkercad:



Schematic:



Cum arată acum:



Software Design

Mediu de Dezvoltare

- **Platformă de Dezvoltare:** Arduino IDE

Biblioteci folosite

1. **Wire Library** (`Wire.h`):

1. **Descriere:** Bibliotecă standard care facilitează comunicarea I2C între Arduino și alte dispozitive I2C.
2. **Link:** [Wire Library](#)

2. **LiquidCrystal_I2C Library** (`LiquidCrystal_I2C.h`):

1. **Descriere:** Utilizată pentru a controla afișajele LCD I2C.
2. **Link:** [LiquidCrystal I2C Library](#)

Algoritmi și Metode Implementate

1. `setup()`

- **Descriere:** Inițializează toate componentele utilizate în joc, inclusiv LCD-ul, butoanele, LED-urile și buzzer-ul. Afișează un mesaj de întâmpinare pe LCD și pornește sunetul de start. Setează timerul inițial și pregătește tabla de joc.

2. *loop()*

- **Descriere:** Bucla principală a programului care verifică starea jocului. Actualizează afișajul și poziția cursorului în funcție de intrările joystick-ului și butoanelor. Dacă jocul nu s-a terminat, permite selectarea celulelor și resetarea jocului prin butoane. Dacă jocul s-a terminat, afișează un mesaj de final și permite resetarea jocului.

3. *generate_board()*

- **Descriere:** Inițializează tabla de joc cu celule acoperite (`#`). Aceasta pregătește tabla pentru jocul nou.

4. *initialize_game()*

- **Descriere:** Inițializează jocul cu o configurație predefinită a minelor pe tablă. Copiază configurația minelor într-o variabilă globală.

5. *display_board()*

- **Descriere:** Afișează tabla de joc și timpul curent pe LCD. Afișează două rânduri din tablă și timpul de joc în format mm:ss în partea dreaptă a ecranului.

6. *position_cursor()*

- **Descriere:** Poziționează cursorul pe baza intrării joystick-ului. Permite deplasarea cursorului în sus, jos, stânga și dreapta pe tablă.

7. *select_cell()*

- **Descriere:** Selectează o celulă și dezvăluie conținutul acesteia. Dacă celula conține o mină, jocul se termină. Dacă celula conține `0`, dezvăluie și celulele adiacente care conțin `0`.

8. *reveal_connected_zeros()*

- **Descriere:** Descoperă celulele adiacente care conțin `0` și celulele învecinate fără mine. Utilizează o coadă pentru a explora recursiv celulele adiacente.

9. *check_win()*

- **Descriere:** Verifică dacă toate celulele non-mină au fost descoperite. Dacă toate celulele non-mină sunt descoperite, jucătorul câștigă.

10. *display_win_message()*

- **Descriere:** Afișează mesajul de câștig pe LCD pentru 2 secunde.

11. *display_game_over_message()*

- **Descriere:** Afișează mesajul de game over pe LCD pentru 2 secunde.

12. *scroll_message()*

- **Descriere:** Afișează un mesaj pe LCD care se derulează pentru a încuraja jucătorul să reînceapă jocul. Este utilizat după terminarea jocului.

13. *reveal_final_state()*

- **Descriere:** Dezvăluie starea finală a jocului cu bombele clipind timp de 3 secunde. Aceasta este utilizată atât pentru câștig, cât și pentru pierdere.

14. *playStartSound()*

- **Descriere:** Redă un sunet de început specific atunci când jocul începe. Utilizează un buzzer pentru a reda o melodie predefinită.

15. *playWinningSound()*

- **Descriere:** Redă un sunet specific de câștig atunci când jucătorul câștigă jocul. Utilizează un buzzer pentru a reda o melodie predefinită de câștig.

16. *playLosingSound()*

- **Descriere:** Redă un sunet specific de pierdere atunci când jucătorul pierde jocul. Utilizează un buzzer pentru a reda o melodie predefinită de pierdere.

17. *reset_game()*

- **Descriere:** Resetează toate variabilele și starea jocului pentru a începe un joc nou. Afișează un mesaj de inițializare pe LCD și redă sunetul de început. Resetează timerul, LED-urile și cursorul.

Surse și Funcții Implementate

Codul Principal: [GitHub link](#)

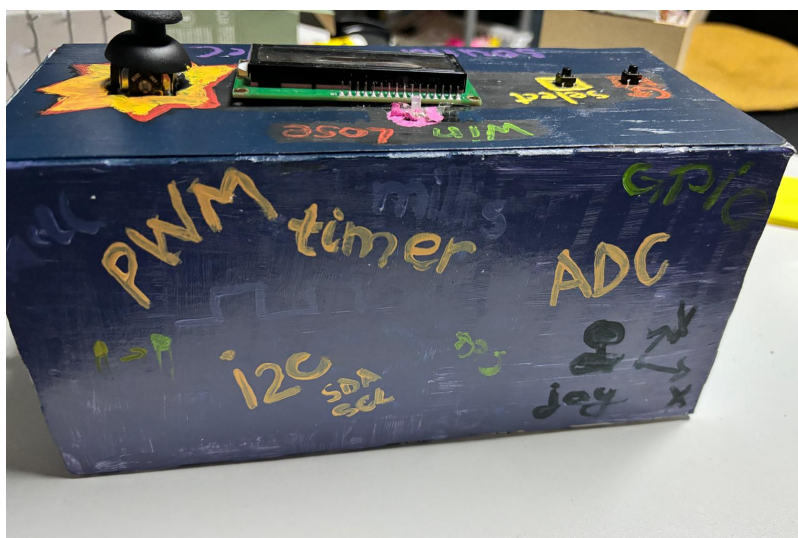
Laboratoare utilizate

1. [Laboratorul 3: Timere, Pulse Width Modulation \(PWM\)](#)
2. [Laborator 4: Analog Digital Convertor \(ADC\)](#)
3. [Laboratorul 6: I2C \(Inter-Integrated Circuit\)](#)

Rezultate Obținute

Demo Youtube: [Link](#)

Câteva poze cu carcasa finală:



Concluzii

Proiectul a fost o oportunitate excelentă de a învăța și de a aplica cunoștințe despre programarea microcontrolerelor, integrarea componentelor hardware și dezvoltarea unui joc complet funcțional. A fost un exemplu clar de cum se poate combina hardware-ul și software-ul pentru a crea o experiență interactivă și captivantă.

Consider că acest proiect m-a ajutat să vizualizez mai bine ce anume vreau să fac pe viitor.

Per total, proiectul este unul mai mult bazat pe software, dar care a avut și dificultăți hardware pentru a obține rezultatele corecte.

Download

Descărcare arhivă: [andra-ioana_stoica_332ca.zip](#)

Bibliografie/Resurse

Software Related

1. [Stack Overflow](#)
2. [LCD I2C Display Questions - Arduino Forum](#)
3. [Bi-color LED \(red/green and yellow/green\) - Arduino Forum](#)
4. [Interfacing RGB LED with Arduino - Project Hub](#)
5. [Piezo Buzzer Win and Fail Sound - Arduino Forum](#)
6. [Testing Joystick - Arduino Forum](#)
7. [How to Use Active and Passive Buzzers on the Arduino - Circuit Basics](#)
8. [Analog Joystick with Arduino - Explore Embedded](#)
9. [I2C LCD with Arduino Tutorial - Last Minute Engineers](#)
10. [C++ Implementation of Minesweeper Game - GeeksforGeeks](#)

Hardware Related

1. I2C Controller - Optimus Digital
2. LCD Display - Optimus Digital
3. Placa de Dezvoltare - Optimus Digital
4. Joystick - Optimus Digital

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/andra_ioana.stoica



Last update: **2024/05/27 21:30**