

Minesweeper

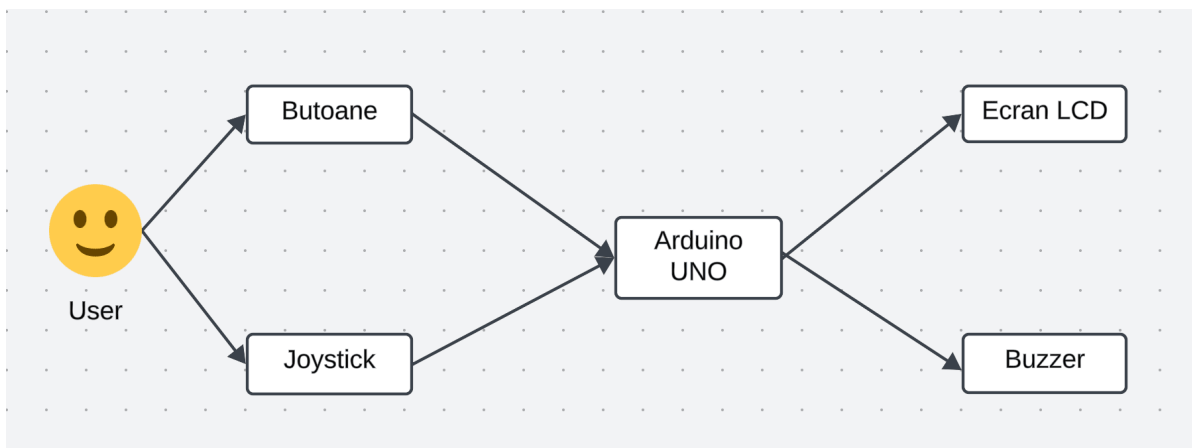
Autor: Alexe Alexandra Alexe

Introducere

Proiectul constă în realizarea jocului Minesweeper, controlat prin butoane cu afișaj pe un ecran LCD. Se va calcula și se va afișa score-ul în fiecare moment, iar la fiecare acțiune a jucătorului se va auzi un sunet corespunzător. La începerea jocului va fi nevoie de introducerea unui username, cu ajutorul unui joystick care ciclează printre litere.

Ideea jocului este de a descoperi toate pătrățelele dintr-o grilă evitându-le pe cele care ascund o mină sub ele. Când un pătrățel este dezvăluit, se afișează un număr care indică numărul de mine învecinate. Acest joc testează abilitățile logice și capacitatea de a face deducții. Ideea jocului a venit din dorința de a realiza un joc cu ajutorul unei plăcuțe și am ales Minesweeper, fiind unul dintre preferatele mele. Așadar consider că jocul este util deoarece antrenează gândirea logică, dar reprezintă și o modalitate de relaxare în același timp.

Descriere generală



Pentru începerea jocului este necesar ca utilizatorul să își introducă numele cu ajutorul joystick-ului. Jocul începe prin inițializarea tablei de joc cu un număr default de mine. Cu ajutorul celor 6 butoane, jucătorul se deplasează pe tablă și descoperă pătrățele. Jocul se termină într-o notă pozitivă, emisă de buzzer, dacă toate pătrățelele au fost descoperite evitându-se minele, altfel se va auzi un sunet corespunzător eșecului.

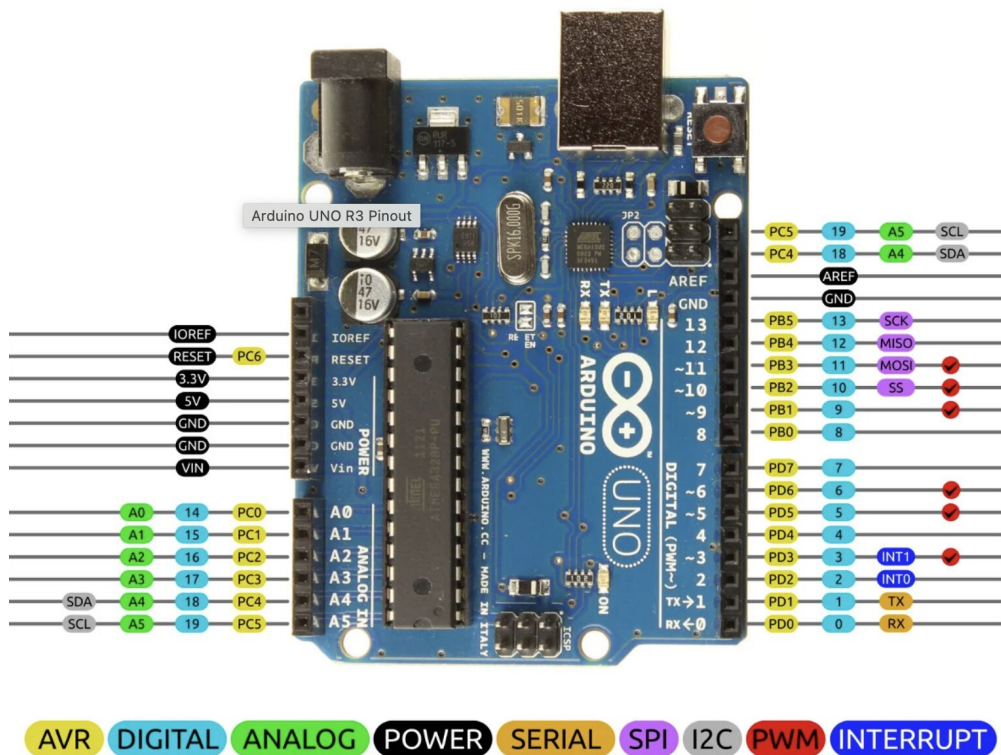
Hardware Design

Componente:

- Arduino UNO R3
- modul de LCD SPI 1.8
- modul de joystick
- modul de buzzer activ
- 6 butoane: sus, jos, stângă, dreapta, descoperire, marcare mina
- fire

Pinii de dezvoltare Arduino:

Arduino Uno R3 Pinout



Schema electrica a proiectului

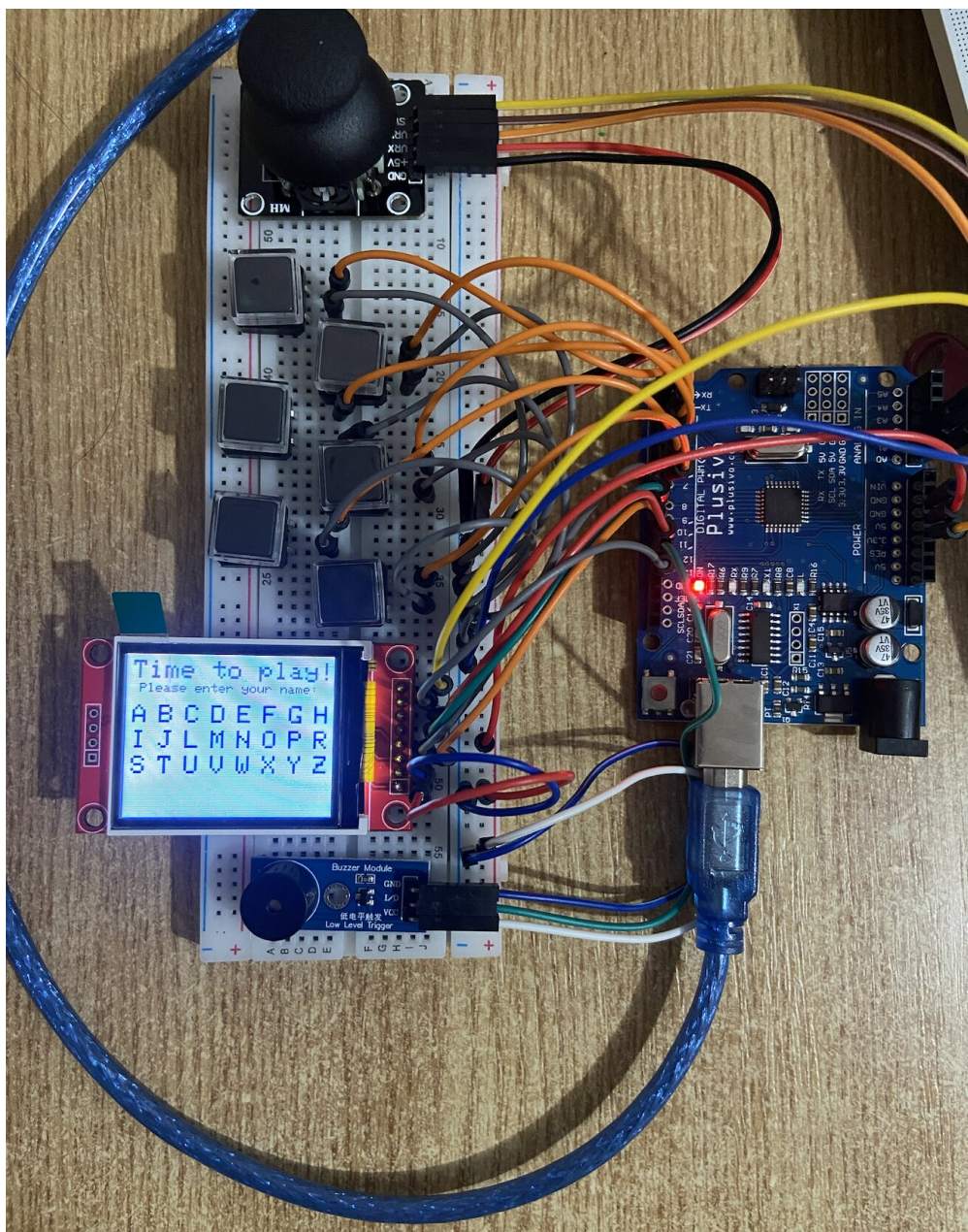
Am folosit:

- pinii digitali 2-7 pentru controlarea celor 6 butoane
- pinii digitali 8-13 sunt corelati cu pinii ecranului
- buzzer-ului se folosește de pinul A3
- controlul joystick-ului pe axele Ox si Oy se transmite prin pinii analogici A0 si A1
- fiindcă la final nu mai aveam niciun pin digital disponibil pentru pinul SW al joystick-ului, m-am folosit de unul analog, A2

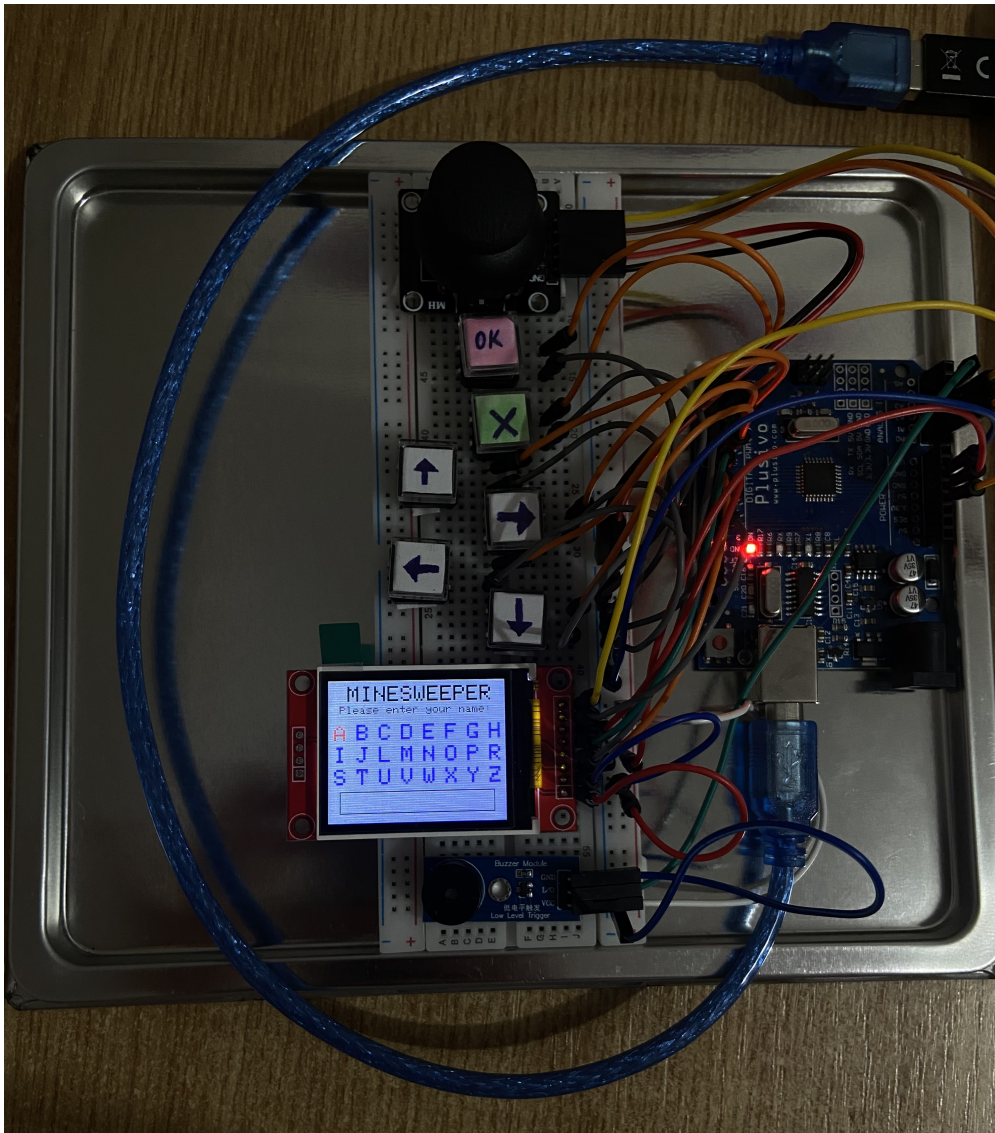


Următoarea poză ilustrează implementarea hardware a proiectului la momentul deadline-ului. Așa

cum se poate observa, am testat funcționalitatea ecranului LCD.



Adaug o poza a hardware-ului final, în care am stabilizat totul pe o placa de metal și am adăugat imagini sugestive pentru folosirea butoanelor.



Software Design

Mediul de dezvoltare folosit a fost Arduino IDE, iar librăria folosită este TFT.h, care include intern și librăria SPI.h, ambele fiind necesare pentru gestionarea comunicarea cu LCD-ul.

FLOW-UL JOCULUI

- Așa cum este prezentat și în schema de mai jos, jocul începe prin introducerea numelui utilizatorului folosindu-se de joystick. Selectarea unei litere se face apăsând butonul joystick-ului. Cursorul care ciclează printre litere este evidențiat prin culoarea roșie. Este posibilă ștergerea unei litere greșite folosind butonul marcat cu X, iar pentru a începe jocul trebuie apăsat butonul OK.
- Următorul ecran ilustrează tabla de joc, reprezentată de o matrice de 9 x 6. Mutarea cursorul se face folosind cele 4 butoane, evidențiate prin săgeți. Descoperirea unei celule este posibilă prin butonul OK, iar marcarea cu flag (F) se face prin butonul X, acesta reprezentând existența unei mine acolo. Pe măsură ce jocul se desfășoară, celulele descoperite se disting față de celelalte prin o altă culoare de fundal (roz).
- În partea de sus a tablei de joc este însemnat scorul jocului, acesta modificându-se în timp real în funcție de alegerile din joc.

- Am ales să însoțesc fiecare apăsare de buton printr-un sunet scurt. Nu am făcut cea mai bună alegere în cumpărarea acestui tip de butoane deoarece sunt momente în care deși butonul este apăsat, nu face contact și acțiunea nu se îndeplinește. Astfel am considerat ca un sunet care validează fiecare acțiune este de ajutor.



IMPLEMENTAREA

Am scris destul de multe linii de cod astfel ca în explicațiile de mai jos, voi insera cele mai importante funcții. În schimb încarc arhiva cu întreaga implementare la secțiunea Download.

1. Definirea pinilor, constantelor și a variabilelor

Configurarea pinilor pentru butoane, joystick, ecran și buzzer am realizat-o folosindu-mă de schema electrică. Am avut nevoie de 2 matrici pentru evidența tablei de joc:

```
int stateBoard[table_rows][table_cols] = { 0 };
int board[table_rows][table_cols] = { 0 };
```

Matricea board reprezintă matricea în starea inițială, așa cum este văzută de către utilizator, în timp ce stateBoard reține pozițiile exacte ale minelor și reține valorile numerice din celule. Tot aici am declarat alte variabile necesare de-a lungul implementării.

2. Funcția setup

În funcția de setup am apelat funcțiile de inițializare pentru ecran, buzzer, joystick și butoane, având grijă să activez rezistențele de pullup din cod. De asemenea, m-am folosit de 2 funcții proprii pentru desenarea literelor și a chenarului în care va fi afișat numele.

```
void setup() {
  TFTscreen.begin();
  clean_background();
  // initializare joystick-ul
  pinMode(SW, INPUT);
  digitalWrite(SW, HIGH);
  // initializare buzzer-ul
  pinMode(buzzerPin, OUTPUT);
  digitalWrite(buzzerPin, HIGH);
  // initializare butoane
  pinMode(BUTTON_PIN_1, INPUT_PULLUP);
  pinMode(BUTTON_PIN_2, INPUT_PULLUP);
  pinMode(BUTTON_PIN_3, INPUT_PULLUP);
  pinMode(BUTTON_PIN_4, INPUT_PULLUP);
  pinMode(BUTTON_PIN_5, INPUT_PULLUP);
  pinMode(BUTTON_PIN_6, INPUT_PULLUP);
  // desenare initiala tabla
  drawAlphabetScreen();
  drawBorder(enteredName);
  randomSeed(analogRead(3));
}
```

3. Funcția loop

Această funcție este împărțită în 2 părți, prima fiind responsabilă de ecranul care permite introducerea numelui jucătorului. Se citesc datele de la joystick și de la butoanele OK și X, modificându-se astfel indicii pentru cursor. Cursorul indica litera selectată și se evidențiază prin culoarea roșu. Accesarea literelor se face cu ajutorul matricei `alphabetMatrix[cursorRow][cursorCol]`. Butonul OK modifică string-ul `enteredName` care reține numele, iar butonul X apelează funcția `clean_background()`; prin care este eliminată ultima litera introdusă. Cea de-a doua parte se declanșează prin modificare variabilei `startPage` în `false` și reprezintă jocul propriu-zis de Minesweeper. Mă asigur de faptul că funcțiile de inițializare a matricilor și a variabilelor necesare sunt setate prin codul de mai jos:

```
if (!gameStarted) {
    score = 0;
    initializeMinesweeperBoard();
    drawScore(enteredName);
    gameStarted = true;
    prevCursorRow = tableCursorRow;
    prevCursorCol = tableCursorCol;
    redrawCursor();
}
```

În fiecare moment primesc date de la cele 4 butoane ce indică direcțiile și modific variabilele ce rețin indicii cursorului. De această dată cursorul este indicat printr-un chenar de culoare distinctă. Butonul OK este sugestiv pentru descoperirea celulelor și prin apăsarea lui este apelată funcția `revealCell(..)` prin care se face descoperirea și funcția `drawScore(enteredName)` prin care este updatat scorul. Apăsarea butonului X duce la apelarea `toggleFlag` și `drawFlag` care sunt responsabile cu marcarea celului cu F (flag) sugerând prezența unei mine acolo. Nu uit să verific cu ajutorul funcției `checkWin` dacă tabla de joc a fost completată corect, ceea ce ar duce la câștigarea jocului.

4. Funcții auxiliare

Următoarele funcții sunt utilizate pentru partea inițială în care se cere introducerea numelui. Trebuie menționat faptul că prin apăsarea directă a butonului OK, se poate evita acest pas.

- `clean_background()`
- `drawAlphabetScreen()`
- `drawBorder(String enteredName)`

Următoarele funcții sunt responsabile de sunetele emise de buzzer pe parcursul jocului:

- `positiveSound()`
- `negativeSound()`
- `oneSound()`

Următoarele funcții ajută la pregătirea jocului propriu-zis:

- `initializeMinesweeperBoard()`
- `redrawCursor()`
- `drawScore(String enteredName)`
- `placeMines(int board[][9], int rows, int cols)`
- `calculateNeighborMines(int board[][9], int rows, int cols)`

Următoarele funcții sunt utilizate în marcarea unei celule cu F(flag):

- toggleFlag(int row, int col)
- drawFlag(int row, int col)

Următoarele funcții ajută la descoperirea unei celule sau la descoperire în masă:

- revealCell(int row, int col)
- revealAdjacentCells(int row, int col)

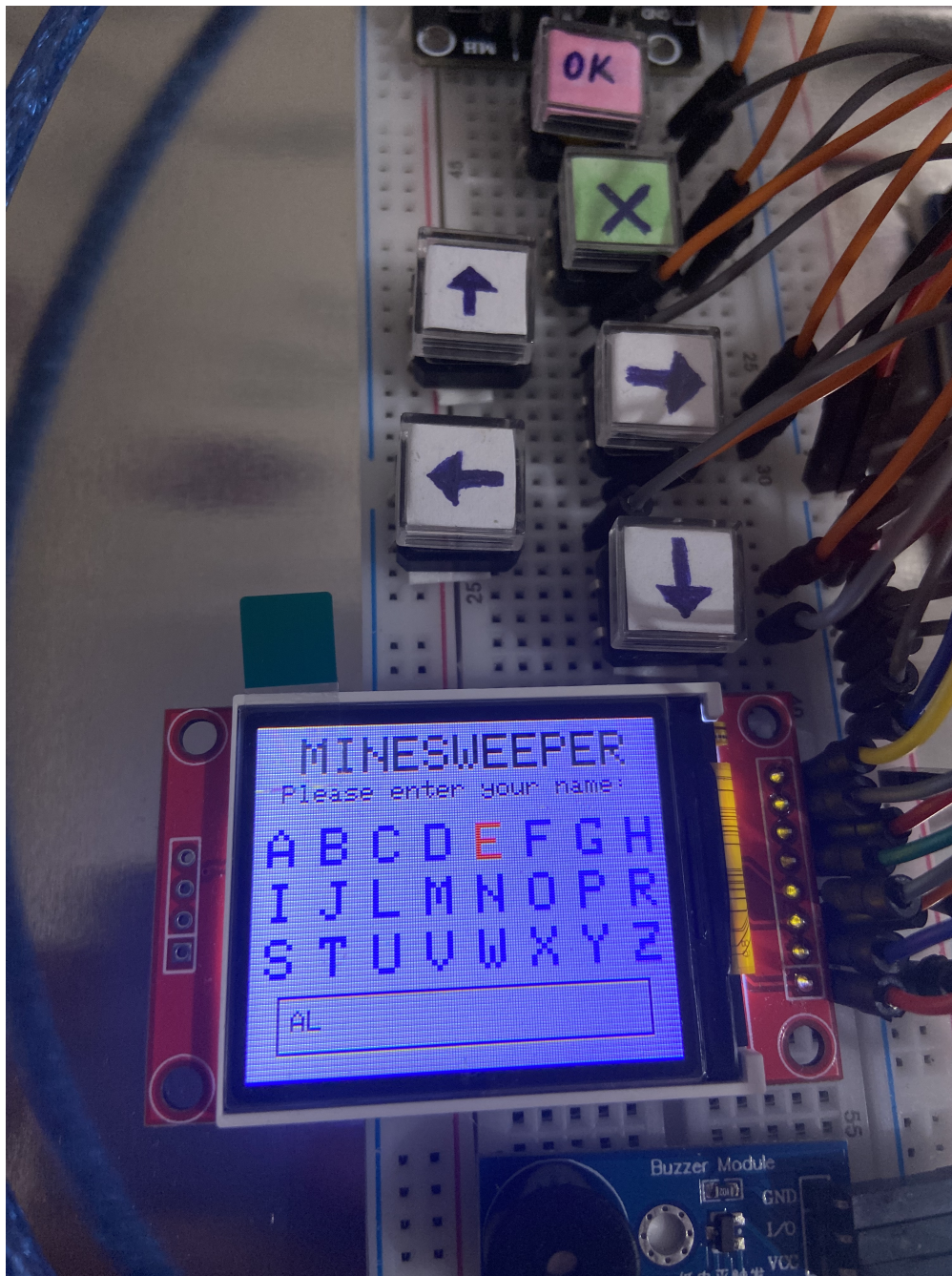
Următoarele funcții sunt utile în verificarea finalului jocului și în afișajele finale, în funcție de caz:

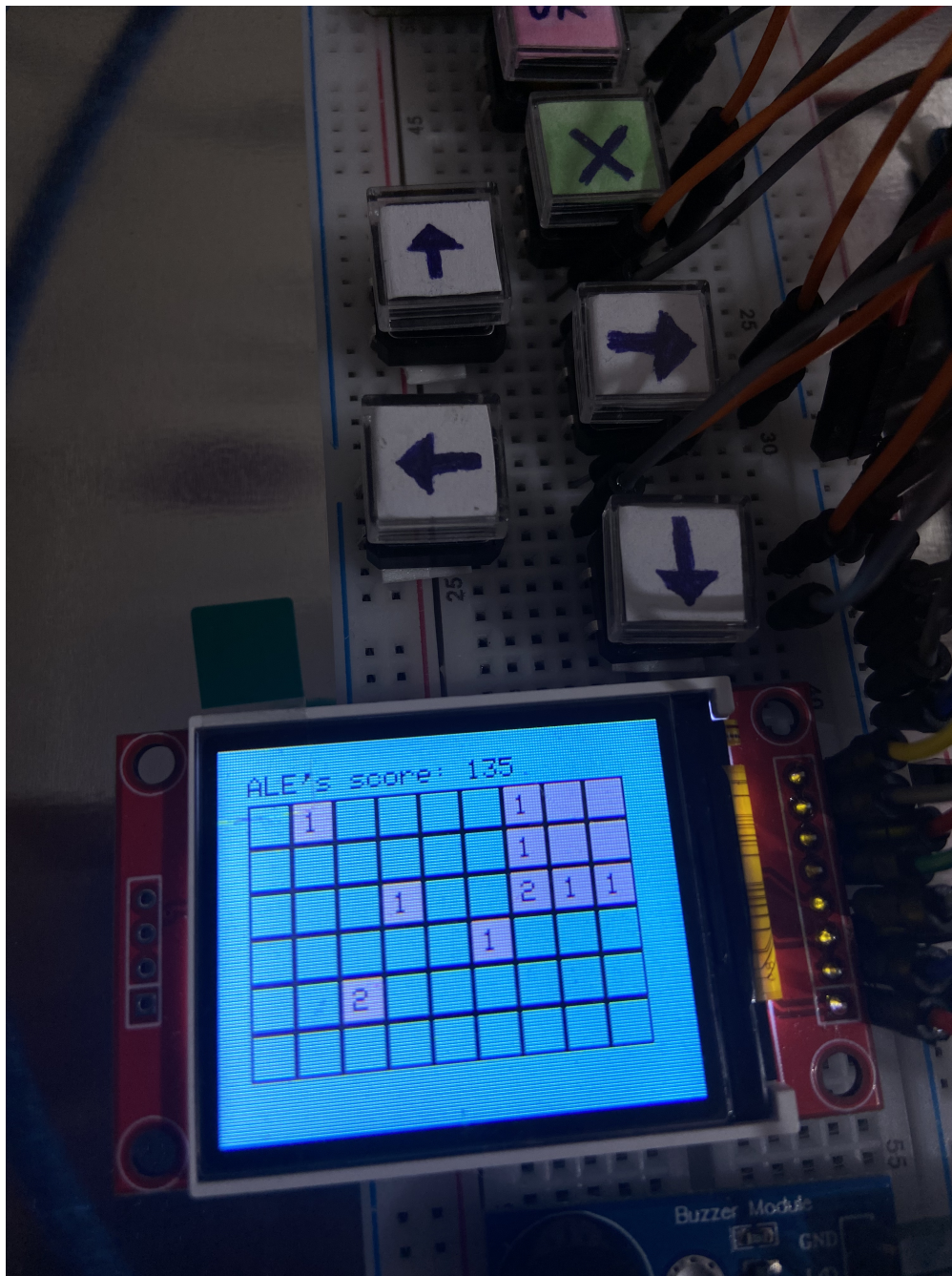
- checkWin()
- displayWinMessage()
- displayGameOverMessage()

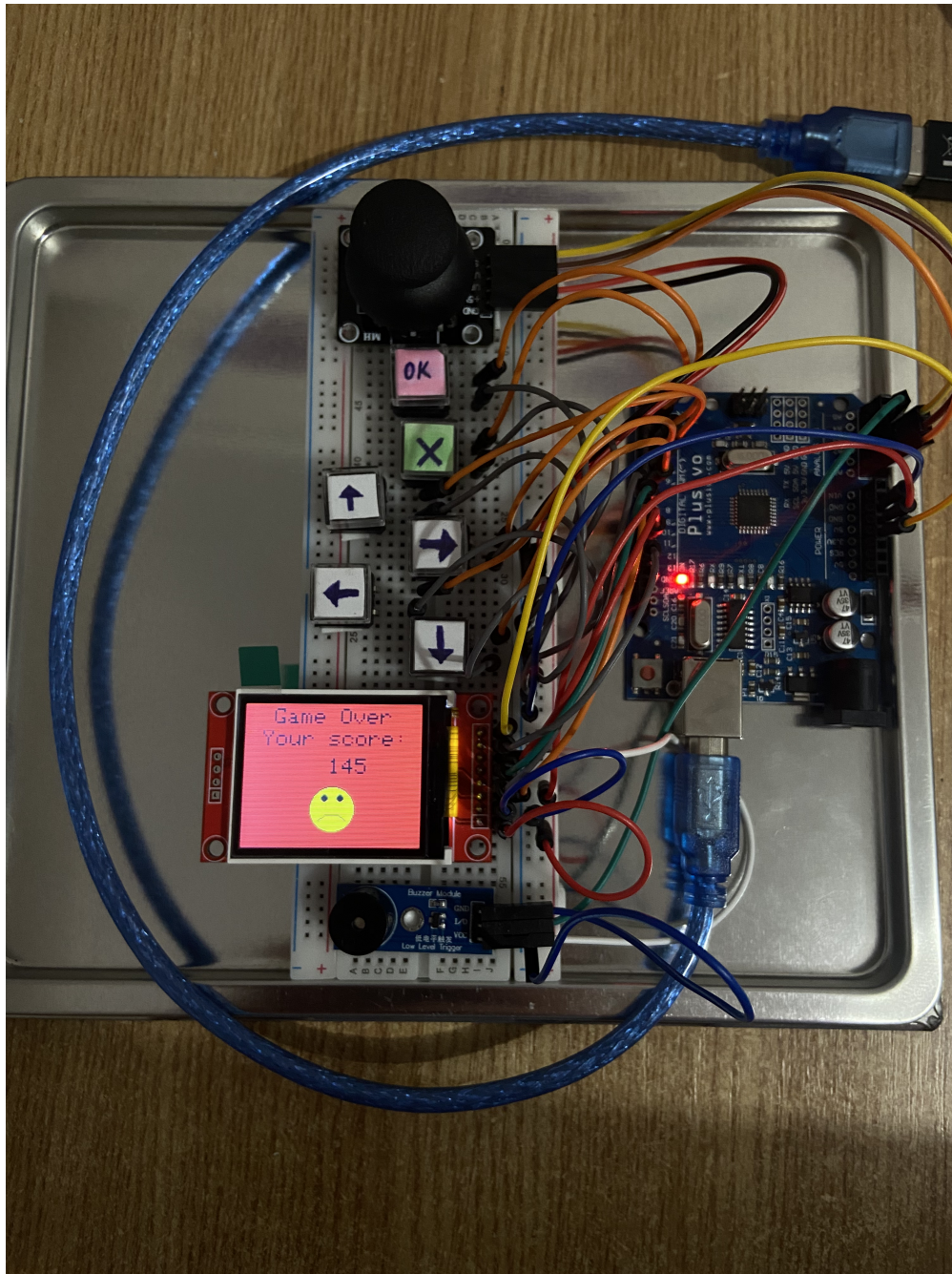
Rezultate Obținute

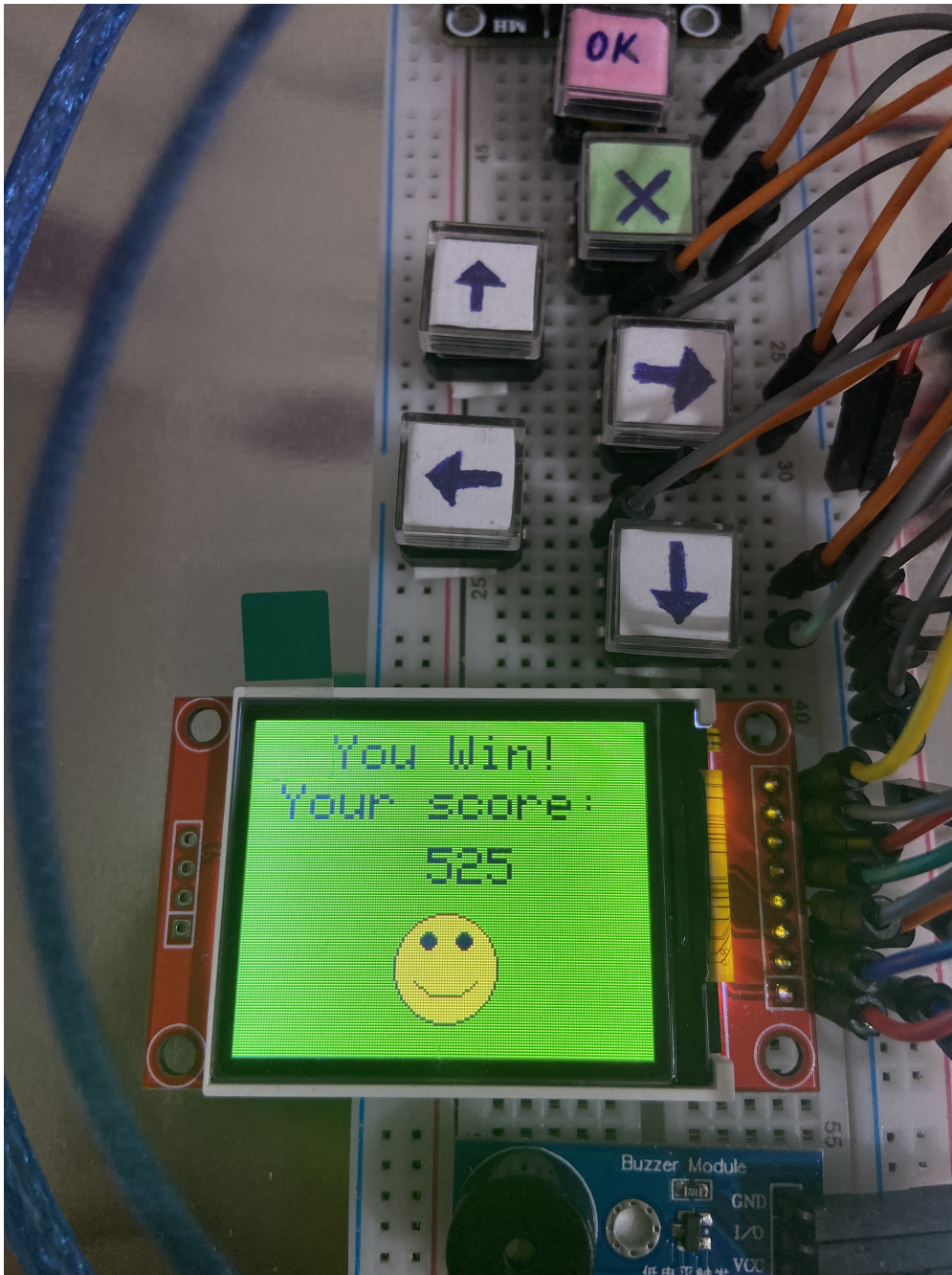
Am realizat tot ceea ce mi-am propus, cu precizarea că dacă aș schimba ceva ar fi buzzer-ul activ. Lucrând la acest proiect am învățat diferențele între un buzzer activ și unul pasiv, astfel că unul pasiv mi-ar fi permis să personalizez mai mult sunetul ce însoțește jocul. Atașez poze în diferite momente ale jocului și un video demonstrativ:

- [Lost Game](#)
- [Won Game](#)









Concluzii

Acesta a fost un proiect din care am învățat noțiuni teoretice despre proiectarea cu microprocesoare, pe care la-am pus în practică, dar am învățat și ce de fapt nimic nu e ceea ce pare: deși am fost temătoare la început cu privire la realizarea acestui proiect, s-a dovedit a fi o plăcere să lucrez la el!

Download

[alexe_alexandra.zip](#)

Jurnal

26.04.2024 - Comandarea pieselor.

5.05.2024 - Realizarea documentației inițiale.

17.05.2024 - Realizarea hardware-ului

23.05.2024 - Finalizarea implementarii software

Bibliografie/Resurse

Resurse Hardware

- <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- <https://www.instructables.com/How-to-Interface-With-Active-Buzzer-Sensor-Module/>
- https://arduinogetstarted.com/tutorials/arduino-joystick?utm_content=cmp-true

Resurse Software

- <https://www.arduino.cc/reference/en/language/functions/random-numbers/random/>
- <https://www.instructables.com/HOW-TO-INTERFACE-AN-ACTIVE-BUZZER-WITH-ARDUINO/>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/tdicu/alexandra.alexe1305>



Last update: **2024/05/25 23:13**