

# Battleship

---

**Autor:** Miruna-Dalia ȚÎRCĂ [mailto:miruna\_dalia.tirca@stud.acs.upb.ro]

## Introducere

---

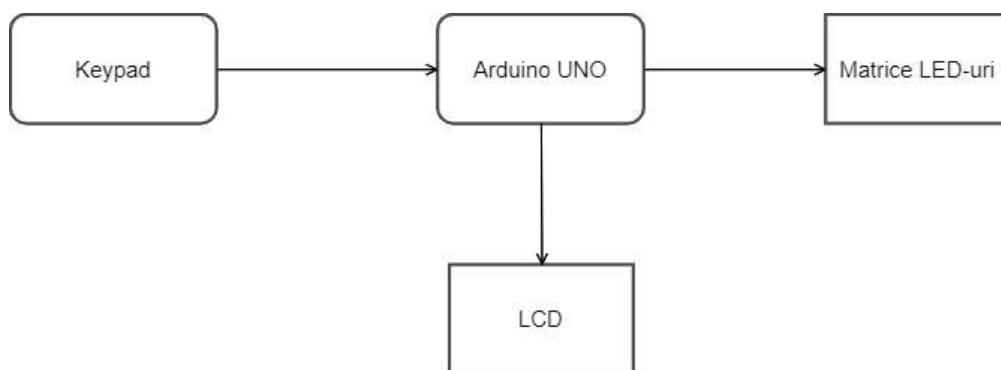
Implementarea jocului Battleship pentru 2 jucatori, fiecare jucator trebuind sa gaseasca navele adversarului.

## Descriere generală

---

Fiecare jucator are o harta cu 3 nave de lungimi diferite dispuse random pe o matrice 8×8. Pe rand, fiecare jucator va ataca navele de pe harta adversarului, prin alegerea unui pozitii folodind tastele de pe keypad. Interfata cu jucatorii se face prin intermediul unui ecran LCD - pe care se va afisa al cui este randul si un mesaj atunci cand un jucator a atacat o pozitie - daca este hit sau miss. In timpul jocului, matricea de LED-uri este folosita pentru a-i arata jucatorului curent pe ce pozitie se afla (prin aprinderea unui LED), iar la castigarea jocului, va fi afisata harta cu navele doborate ale adversarului.

Schema Bloc



1. Arduino Uno R3
  - Mediaza fluzul de date intre keypad, matricea de LED-uri si LCD 16×2.
2. Membrane 4×4 Keypad
  - Jucatorul foloseste tastele 2 (sus), 8(jos), 4(stanga), 6(dreapta) pentru a alege pozitia de pe harta ce poate fi atacata folosind tasta 5.
3. Matrice de LED-uri
  - In timpul jocului, fiecare jucator poate vedea pe matricea de LED-uri pozitia pe care este prin luminarea LED-ului corespunzator (si poate vizuliza cum se deplaseaza pe harta prin apasarea tastelor numerice). Daca decide sa atace o celula, pozitia aratata pe matrice este mapata la harta adversarului si se decide daca este hit sau miss.
  - La finalul jocului, pentru castigator, va fi afisata pe matricea de LED-uri harta doborata a adversarului.
4. LCD-I2C
  - Afiseaza care este jucatorul curent (jucatorul 0 sau 1) si, la fiecare apasare a unei taste, este afisata directia deplasarii.
  - La apasarea tastei 5(atac) este afisat pe ecran daca a fost hit sau miss.
  - La sfarsitul jocului (cand un jucator a doborat toate navele adversarului), este afisat pe ecran mesajul "Player 0/1 won".

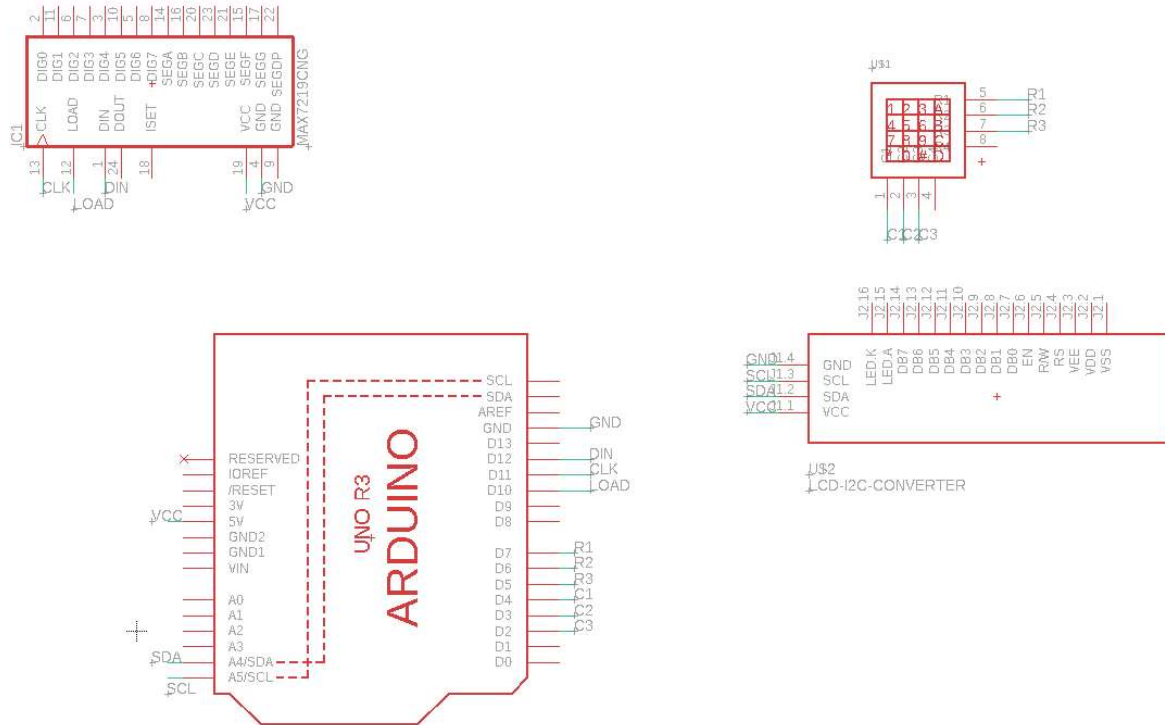
## Hardware Design

---

### Lista de piese

Nume	Numar piese
Arduino Uno R3	1
Keypad	1
Matrice de LED-uri	3
LCD	1
Fire tata-tata	
Fire mama-tata	

## Schema electrica



### Conectare Membrane 4x4 Keypad

Folosesc doar 6 pini din cei 8 ai Keypad-ului (Keypad-ul are cate un pin pentru fiecare rand si pentru fiecare coloana, din care folosesc doar primele 3 randuri si primele 3 coloane - corespunzatoare cifrelor.) Am conectat pinii 8, 7, 6, 4, 3, 2 de la keypad la pinii digitali 7, 6, 5, 4, 3, 2 de la Arduino.

### Conectare modul matrice de LED-uri

Am folosit o matrice de LED-uri 8x8 cu driver-ul MAX7219. Am conectat pinii VCC, GND, DIN, CS, CLK la pinii corespunzatori de pe Arduino.

### Conectare LCD 16x2 I2C

Am folosit un LCD cu modulul I2C conectat si astfel am avut de conectat doar pinii GND, VCC, SDA, SCL la pinii corespunzatori de la Arduino.

## Software Design

- Limbaj de programare folosit: C
- Mediu de dezvoltare si compilare: Arduino IDE

## Implementare

### Biblioteci folosite

- Wire.h - pentru interfata seriala I2C
- LiquidCrystal\_I2C.h - pentru modulul LCD
- Keypad.h - pentru Keypad
- LedControl.h - pentru matricea de LED-uri

### Program

- setup()

- Initializarea variabilelor: hartile, pozitia de inceput de atac (stanga sus), WON(id-ul jucatorului castigator, initial -1).
- Initializarea modulelor: LCD, matricea de LED-uri, keypad.

- loop()

- Cat timp jocul nu este castigat, astept ca jucatorul curent sa aleaga ce pozitie va ataca: afisez pe matricea de LED-uri pozitia selectata de jucator prin apasarea tastelor (si deplasarea) pana cand este apasata tasta 5 - cand se va verifica pe harta adversarului (o matrice de 0 si 1) daca atacul este hit sau miss.
- Daca a fost hit, se pastreaza acelasi player, iar daca a fost miss, se marcheaza pe harta adversarului si se trece la player-ul urmator.
- Jocul a fost castigat atunci cand harta adversarului este complet nula, moment in care este afisat un mesaj pe LCD si pe matricea de LED-uri apare harta "doborata" a adversarului.

Funcțiile folosite pentru logica jocului:

- generateMap

- Genereaza o matrice de 0 si 1 ce reprezinta o harta cu nave de lungime 2, 3 si 4 aseazate random (navele sunt marcate cu 1 in matrice).
- Pentru fiecare noua nava, folosesc functia random pentru a gasi pozitia de start si directia. Pe baza acestor date verific daca nava nu intersecteaza cu alte nave deja amplasate.

- checkHit

- Verifica pentru player-ul curent daca pe harta adversarului, la pozitia atacata se gaseste o nava si returneaza 1 la hit, 0 la miss.
- La fiecare hit, se marcheaza acea pozitie pe harta.

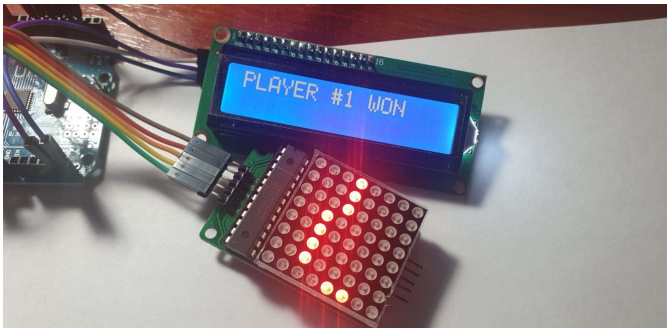
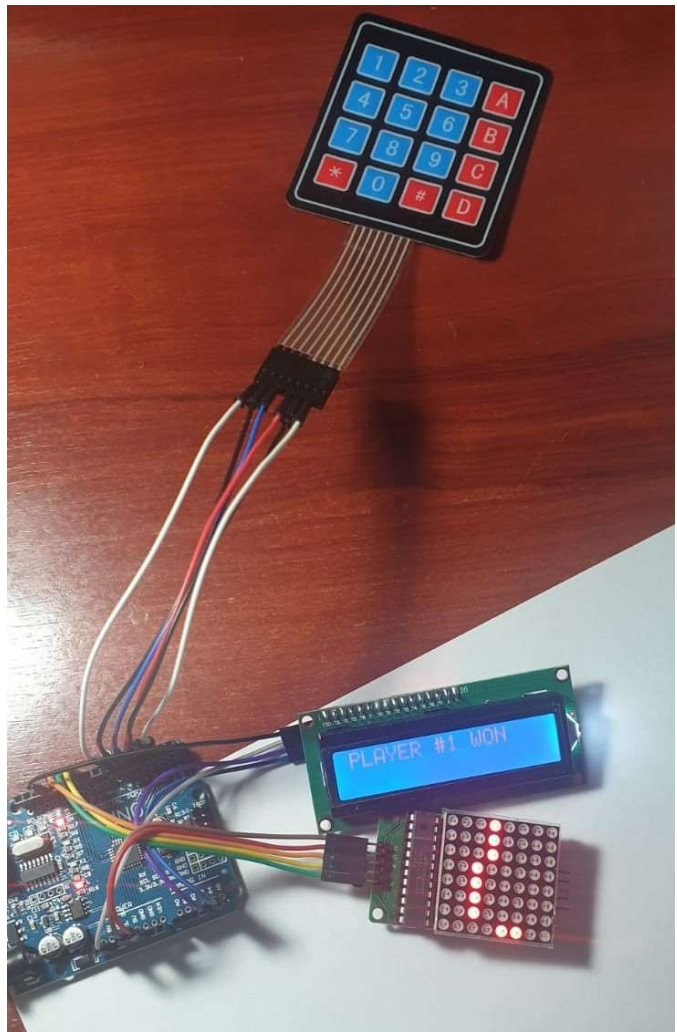
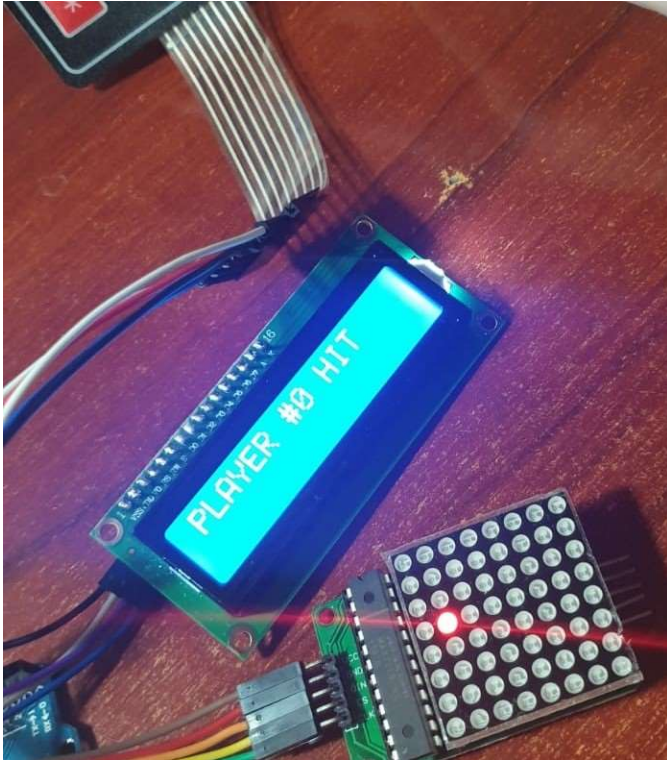
- checkWin

- Daca harta adversarului este nula, atunci player-ul curent a castigat

## Rezultate Obținute

---

Am obtinut un joc Battleship functional pentru 2 jucatori. Fiecare jucator alege pozitia de atac prin apasarea tastelor de la keypad. Interfata cu utilizatorul este intuitiva, folosind o matrice de LED-uri pe care, in timpul jocului, jucatorul poate alege ce pozitie va ataca si un ecran LCD ce va arata rezultatul atacului pentru player-ul curent. La sfarsitul jocului, pe matricea de LED-uri va fi afisata harta doborata a adversarului. Pe LCD se poate vizualiza de asemenea al cui este randul, informatii despre tastele apasate si, la sfarsitul jocului, ce player a castigat.



Video: [https://www.youtube.com/watch?v=DhTAVobn\\_Z4](https://www.youtube.com/watch?v=DhTAVobn_Z4) [[https://www.youtube.com/watch?v=DhTAVobn\\_Z4](https://www.youtube.com/watch?v=DhTAVobn_Z4)]

## Concluzii

---

Proiectul este realizat in intregime si functional. Mi-a placut sa lucrez la acest proiect, pentru ca am putut sa vad si o implementare in hardware a unui joc. Am invatat sa folosesc keypad-ul, modulele de LCD si matricea de LED-uri, folosind in partea de software functiile din bibliotecile corespunzatoare.

## Download

---

Cod sursa: tircamiruna\_cod\_sursa.zip

## Jurnal

---

- 30 aprilie: alegere tema proiect, creare pagina de documentatie
- 4 mai: achizitionare piese

- 10 mai: asamblare piese hardware, testare componente
- 17 mai: implementarea jocului
- 30 mai: schema electrica
- 4 iunie: finalizare pagina de documentatie

## Bibliografie

---

<https://bitbucket.org/nerdfirst/arduino-8x8-basics/src/master/> [<https://bitbucket.org/nerdfirst/arduino-8x8-basics/src/master/>]

[https://www.youtube.com/watch?v=q9YC\\_GVHy5A](https://www.youtube.com/watch?v=q9YC_GVHy5A) [[https://www.youtube.com/watch?v=q9YC\\_GVHy5A](https://www.youtube.com/watch?v=q9YC_GVHy5A)]

<https://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/> [<https://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/>]

pm/prj2021/apredescu/battleship.txt · Last modified: 2021/06/04 18:45 by miruna\_dalia.tirca