

BattleShip

Introducere

Proiectul consta intr-un joc minimal de Battleship cu 2 jucatori (umani), realizat folosind minimul de piese necesar(doua joysitck-uri, doua ecrane, si un buzzer).

Descriere generală

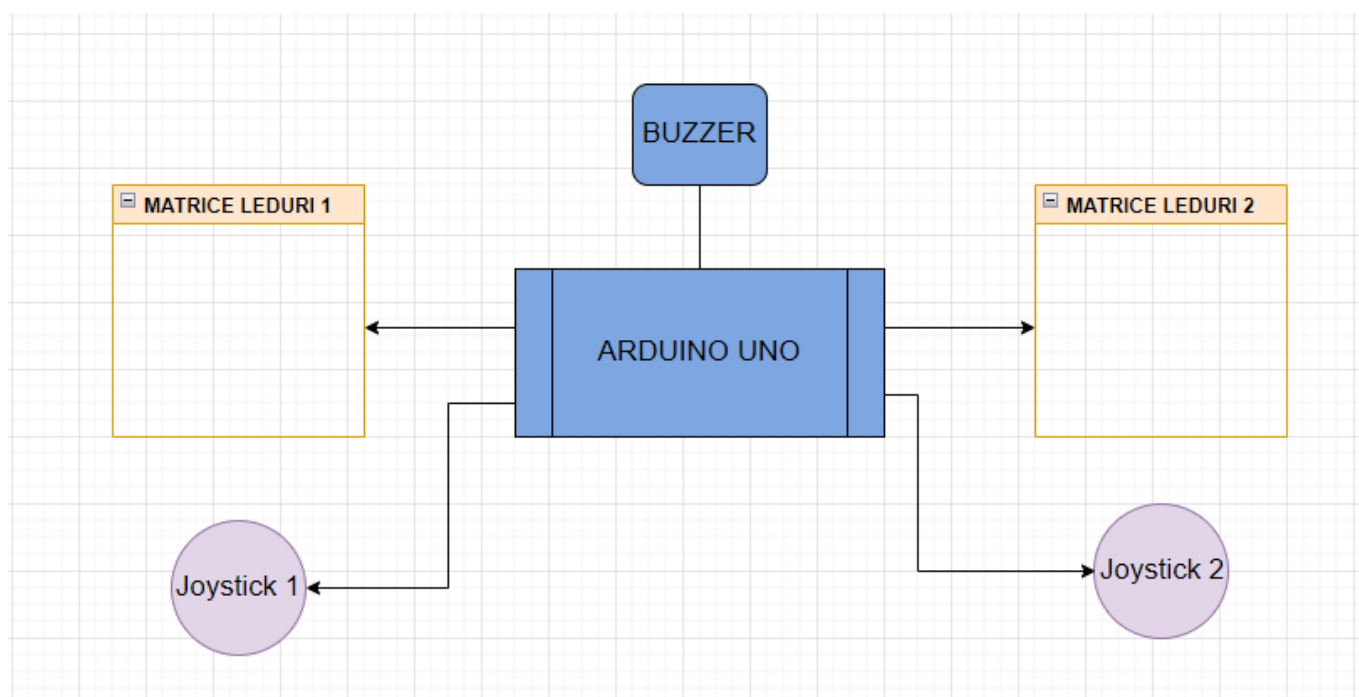
Playerii primesc o aranjare random a pieselor pe care o pot vedea pe propriul ecran.

Turn-ul player-ului este semnalat de un led care face switch rapid on/off pe ecran ce reprezinta cursorul lui. Cursorul poate fi mutat prin joystick si prin apasarea acestuia se selecteaza tinta.

In functie daca a lovit barca jucatorului sau nu, led-ul tintit isi va schimba culoarea in mod specific.

Apoi vine turn-ul player-ului 2, si ciclul se va repeta pana ce unul din playeri a doborat toate piesele inamicului.

Buzzer-ul va semnala prin diferite sunete: tinta lovita, ratarea si finalul de joc (specific pentru fiecare player).

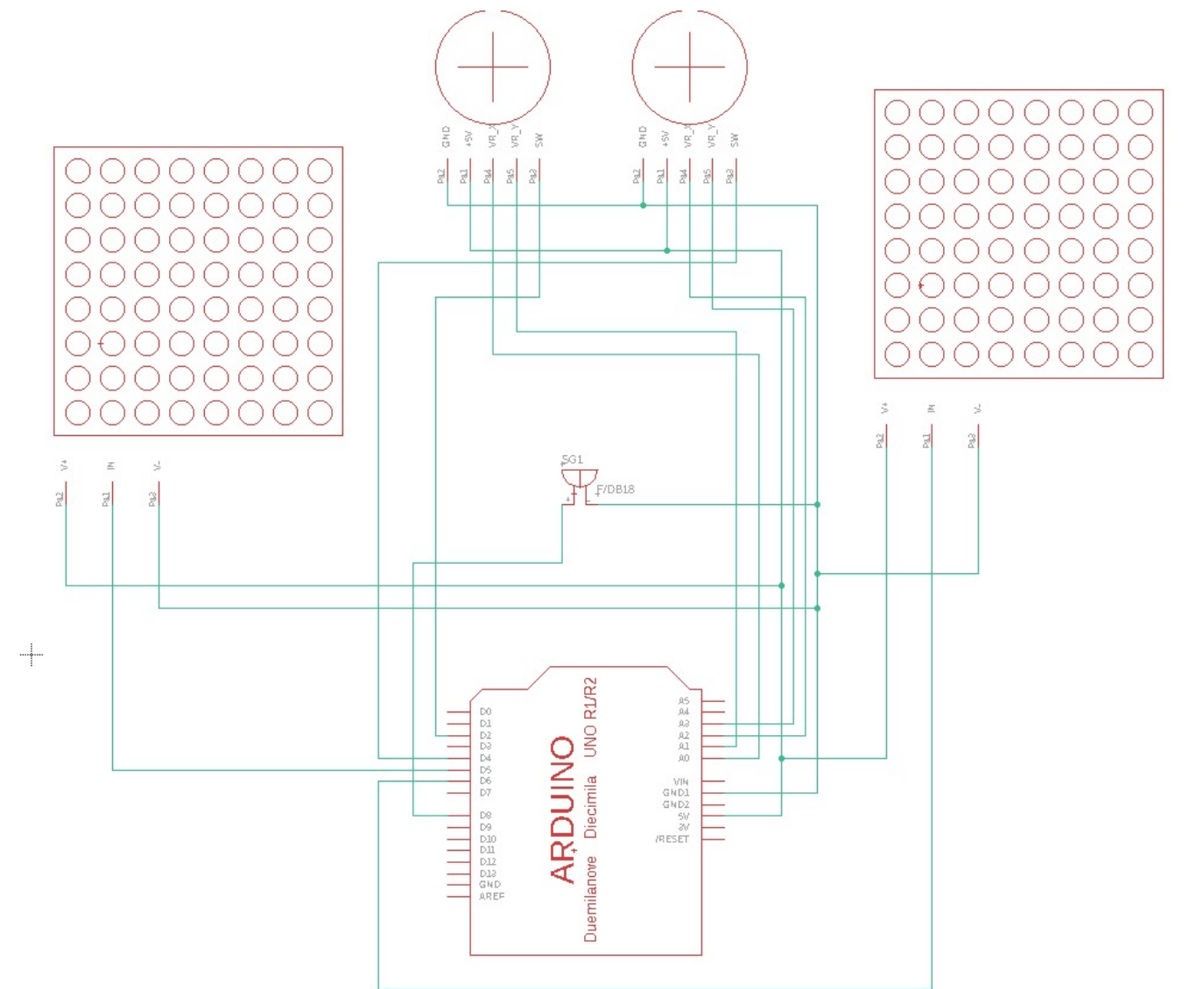


Hardware Design

Lista Piese:

- Arduino Uno
- 2x Matrice 64 RGB Led WS2812B (https://cleste.ro/matrice-64-led-uri-rgb-8x8.html?utm_medium=GoogleAds&utm_campaign=ShoppingAds&utm_source=&gclid=Cj0KCQjw9_mDBhCGARIsAN3PaFMau1WxpPFX1apSHNrZwz-AAqLUkziskS Ua0af5rQ4ECZ_UUttYWMaAvdAEALw_wcB)
- 2x Modul Joystick PS2 Aarduino (<https://cleste.ro/modul-joystick-ps2-compatibil-arduino.html>)
- 1x piezzo buzzer (<https://cleste.ro/modul-buzzer-pasiv.html>)

Schema Electrica



Software Design

Mediu de dezvoltare

- **Arduino IDE** - pentru programarea placutei Arduino
- **EAGLE** - pentru schema electrica

Librarii si surse 3rd-party

- **Biblioteca FastLED** - pentru programarea matricei de leduri

Descrierea codului

Initializari

Folosesc o serie de variabile globale ce vor fi initializate in setup si folosite mai tarziu in joc: Pinii folositi pentru componente:

- SW_pin_joystick0
- X_pin_joystick0
- Y_pin_joystick0
- SW_pin_joystick1
- X_pin_joystick1
- Y_pin_joystick1
- buzzerPin

Variabile ce retin pozitia cursorului (curenta si precedenta):

- cursorX
- cursorY
- lastCursorX
- lastCursorY

Variabila ce retine playerul ce se afla la turn: **player**

Matrici de leduri liniarizate ce vorfi folosite pe parcursul jocului:

- ledsP1Boats
- ledsP2Boats
- ledsP1Hits
- ledsP2Hits
- ledsP1
- ledsP2

Variabile ce reprezinta entitatile ce pot aparea pe matrice:

- boat
- empty
- hit
- miss
- cursor
- cPlayer0
- cPlayer1

Variabile pentru contorizarea timpului la intreruperi:

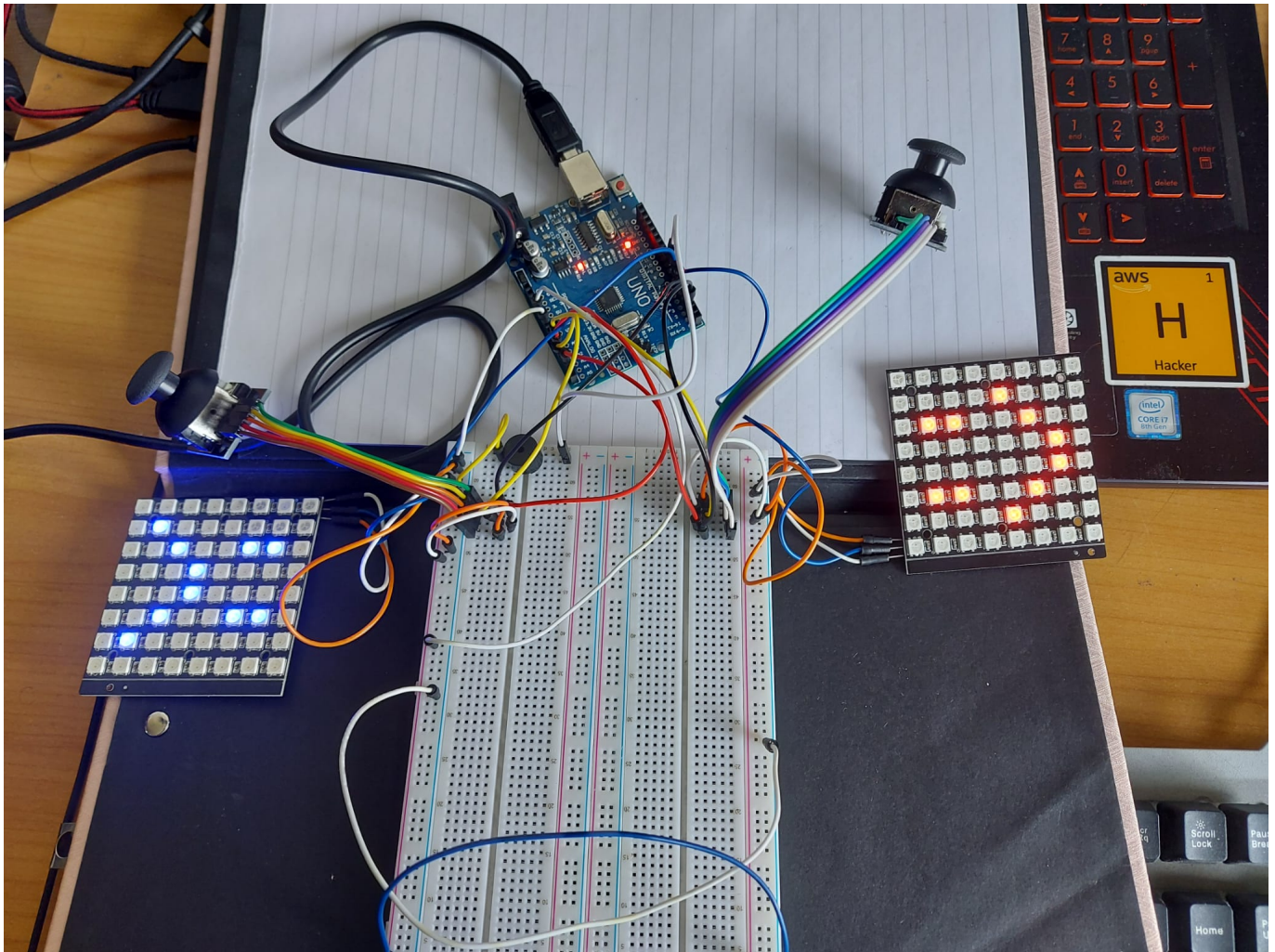
- timerPD2
- timerPD4

Functii folosite

- **resetGame** - elibereaza matricele de joc, seteaza random jocul pentru cei 2 playeri, apoi anunta prin buzzer startul de joc
- **debugBoats** - functie folosita in debugging, printeaza matricele de barcute ale celor 2 jucatori pe interfata seriala
- **setDisplay** - matricele liniarizate ledsP1 si ledsP2 sunt folosite pentru a afisa barcile, respectiv punctele lovite de cei doi jucatori, prin setDisplay, se face switch in functie de jucatorul la turn intre cele doua matrici pentru vizualizarea jocului
- **setup** - setez pinii dispozitivelor folosite, pornesc interfata seriala, initializez jocul prin resetGame() si setez cele doua matrici de leduri la pinii corespunzatori
- **play** - in aceasta functie se citesc miscarile joystick-ului de la playerul ce se afla la turn si se muta corespunzator cursorul acestuia. Apasarea joystick-ului anunta faptul ca jucatorul a tintit un anume punct din arena adversarului, ce urmeaza sa fie evaluat la hit or miss
- **start** - functia ce va fi folosita in loop-ul programului, aceasta face switch intre matricele de joc prin setDisplay si apeleaza functia play cu parametrii necesari jucatorului la turn
- **win** - anunta prin buzzer finalul jocului, pune pe display-ul castigatorului un smiley face si pe cel al pierzatorului un sad face, apoi prin apasarea oricarui buton al joystick-ului reincepte jocul prin functia resetGame
- **checkEnded** - verifica daca vreunul dintre jucatori a castigat (a distrus toate "barcutele" adversarului) si apeleaza functia win daca unul din jucatori a castigat
- **ISR(INT0_vect)** - functie ce controleaza intreruperea INT pe PD2, daca butonul a fost tinut apasat mai mult de 2.5 secunde, se resetaza jocul la urmatoarea mutare
- **ISR(PCINT2_vect)** - functie ce controleaza intreruperea PCINT pe PD4, acelasi efect ca cel de pe PD2

Rezultate Obținute

Demo: <https://www.youtube.com/watch?v=ci0xViEkQDU>



Concluzii

Am bifat un achievment pe care il am inca din copilarie: sa fac jocuri :)

Cred ca am sumarizat destul de bine continutul laboratoarelor de PM.

Download

[333cb_anica_alexandru-gabriel_battleship.zip](#)

Bibliografie/Resurse

Laboratoarele de PM

Biblioteca pentru control led-uri: <https://github.com/FastLED/FastLED>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/imacovei/alexandru.anica>



Last update: **2022/05/17 14:41**