

# Bogdan-Ionuț ION (67194) - Ultimate Tic-Tac-Toe

Autorul poate fi contactat la adresa: **Login pentru adresa**

## Introducere

Proiectul constă în implementarea jocului Ultimate Tic-Tac-Toe pe o placă cu LCD.

Scopul proiectului este reprezentat de jocul de X-O între un jucător și A.I. sau jucător vs jucător dar pe un grid de  $9 \times 9$ . Atunci când un jucător pune intr-o casuta într-un  $3 \times 3$  grid din grid-ul de  $9 \times 9$ , jucătorul advers este trimis într-un alt  $3 \times 3$  grid unde va putea pune setă o alta casuta. Atunci când un player câștigă un grid  $3 \times 3$  din cel  $9 \times 9$ , se setează într-un MacroGrid (care este  $3 \times 3$ ) ca și câștigată acea casuta corespunzătoare gridului  $3 \times 3$ . De exemplu, în colțul din stânga-sus al gridului  $9 \times 9$ , am un grid  $3 \times 3$ . În MacroGrid, acel  $3 \times 3$  grid va fi o casuta  $1 \times 1$ , asociată gridului  $3 \times 3$  din cel  $9 \times 9$ . Pentru a câștiga, se verifică MacroGrid-ul în același mod în care se verifică pentru câștigare jocul clasic de Tic-Tac-Toe.

Ideeia de la care am pornit este reprezentată de jucarea unui joc Tic-Tac-Toe la alt nivel, în care se evită draw-urile pe cat de mult posibil, ceea ce în jocul clasic nu se prea poate face.

Este util, deoarece reprezintă o provocare mai mare, care necesită multă atenție și logica.

## Descriere generală



Există 5 butoane. 4 vor fi folosite pentru a ne putea deplasa printre casute, și unul pentru a pune X/0 în acea casută. Pe LCD se va afisa grid-ul impreuna cu niste informatii utile despre joc, precum al carui rand este, sau cine a castigat. Microprocesorul va realiza operatiile necesare de interactiune.

## Hardware Design

Lista de piese:

- Modul LCD TFT SPI de 1.44" (128×128 px) Rosu, bazat pe controller-ul ST7735S
- Placa de baza cu microprocesorul ATMega324
- Fire mama-mama
- Alimentare 3.3V
- 5 Butoane(cele 4 directii si insert X / 0)
- Headere
- Placa de test PCB 9 x 15 cm

Schema electrica:



## Software Design

### Mediu de dezvoltare

- WinAVR
- Programmer's Notepad

### Librarii și surse 3rd-party

Am folosit o librarie care comunica prin SPI cu LCD-ul: [lcd\\_spi\\_st7735.rar](#)

De asemenea, am folosit și librariile de la laborator(avr, util/delay.h, stdlib, stdio).

### Algoritmi și structuri pe care plănuiați să le implementați

Pe ecran va apărea grid-ul 9×9 pe care se va juca. Voi implementa Minimax pentru A.I.. Prin apasarea celor 4 butoane jucatorul se va putea deplasa în grid, iar prin intermediul celui de al 5-lea va putea pune X/0 într-o casută.

### Surse și funcții implementate

Pentru LCD

- void LCD\_init(void);
- void LCD\_end(void);
- void LCD\_set\_range(uint16\_t x1, uint16\_t y1, uint16\_t x2, uint16\_t y2);
- void LCD\_set\_pixel(uint16\_t color);

- void LCD\_draw\_pixel(int16\_t x, int16\_t y, uint16\_t color);
- void LCD\_fill\_screen(uint16\_t color);

Pentru grafica

- void GFX\_init(int16\_t width, int16\_t height);
- void GFX\_draw\_line(int16\_t x1, int16\_t y1, int16\_t x2, int16\_t y2, uint16\_t color);
- void GFX\_draw\_circle(int16\_t x, int16\_t y, int16\_t radius, uint16\_t color, uint8\_t fill);

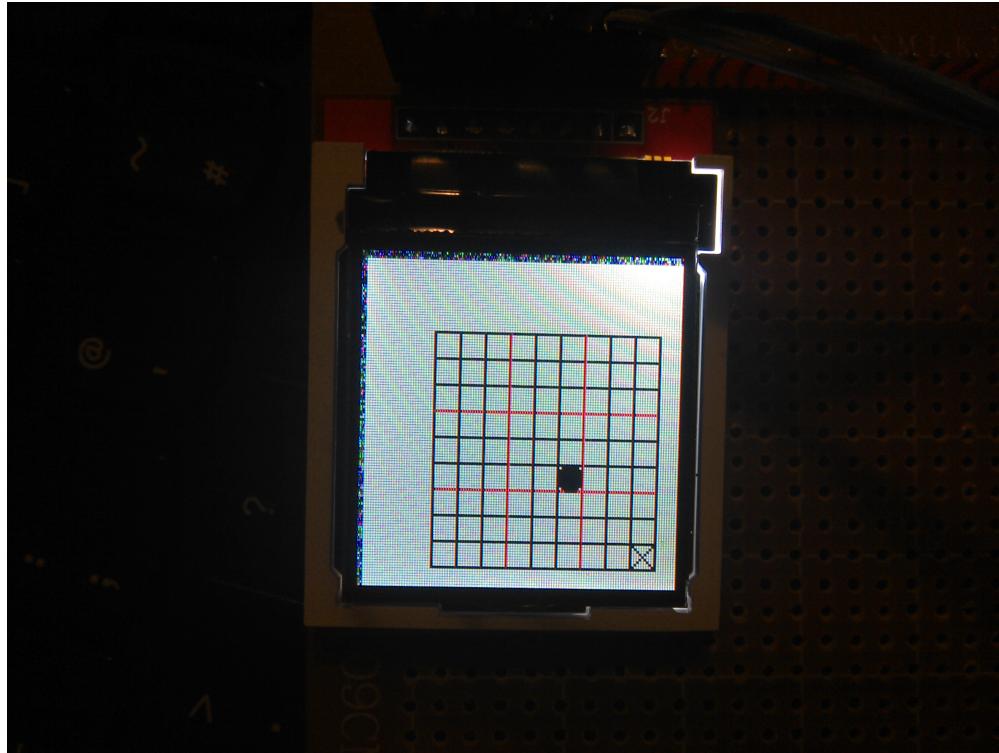
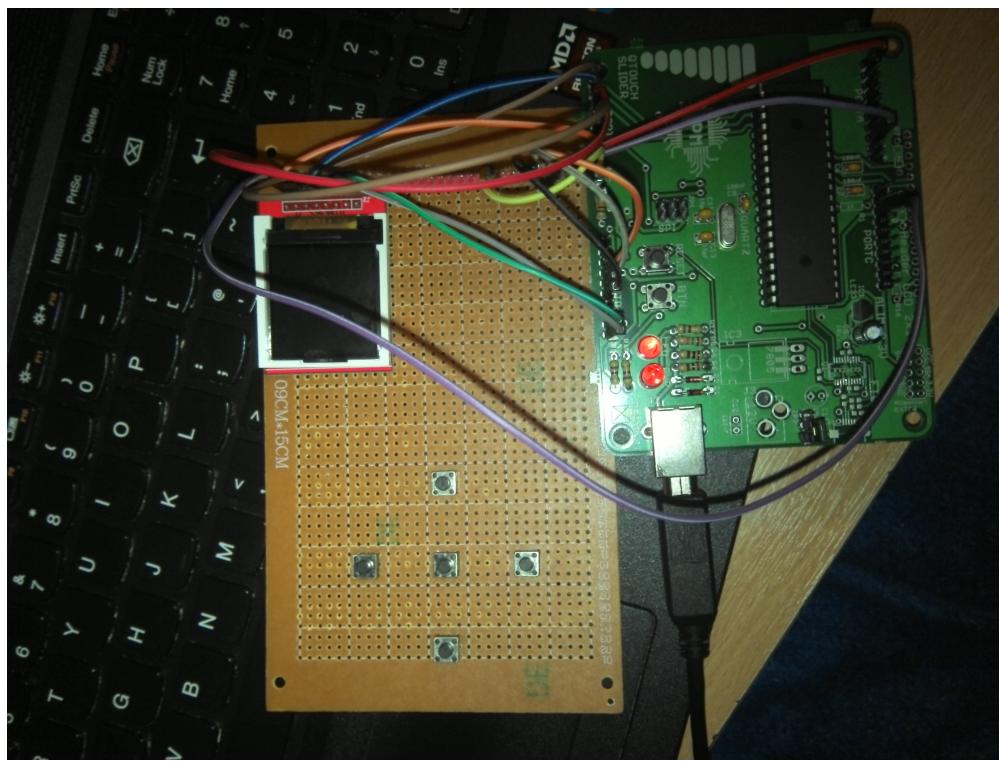
Pentru joc

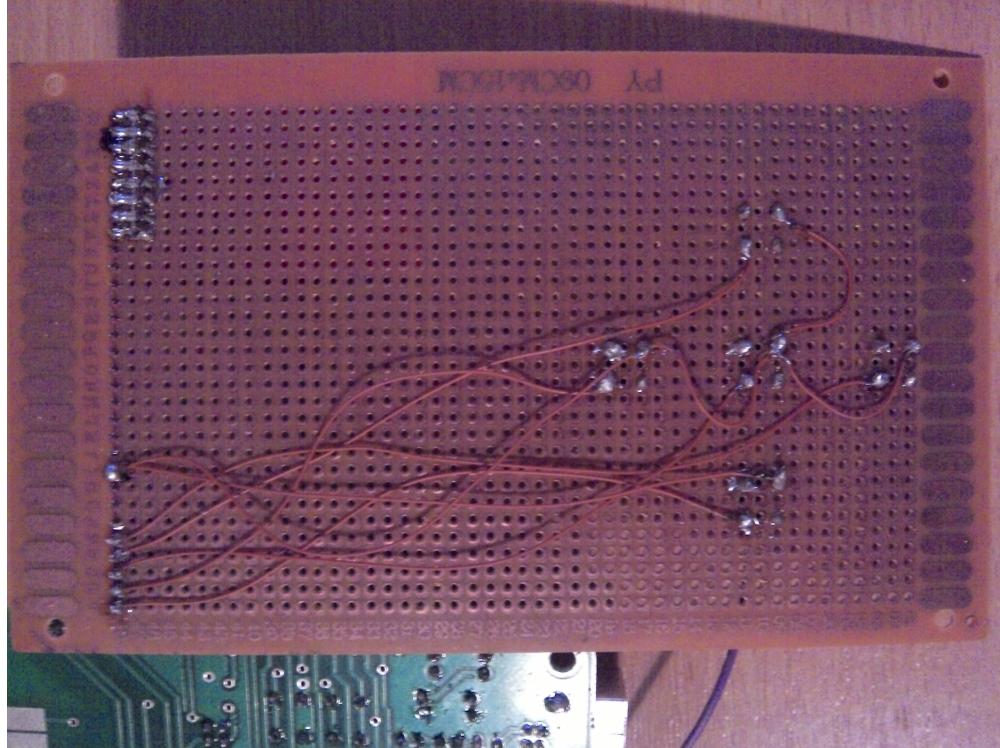
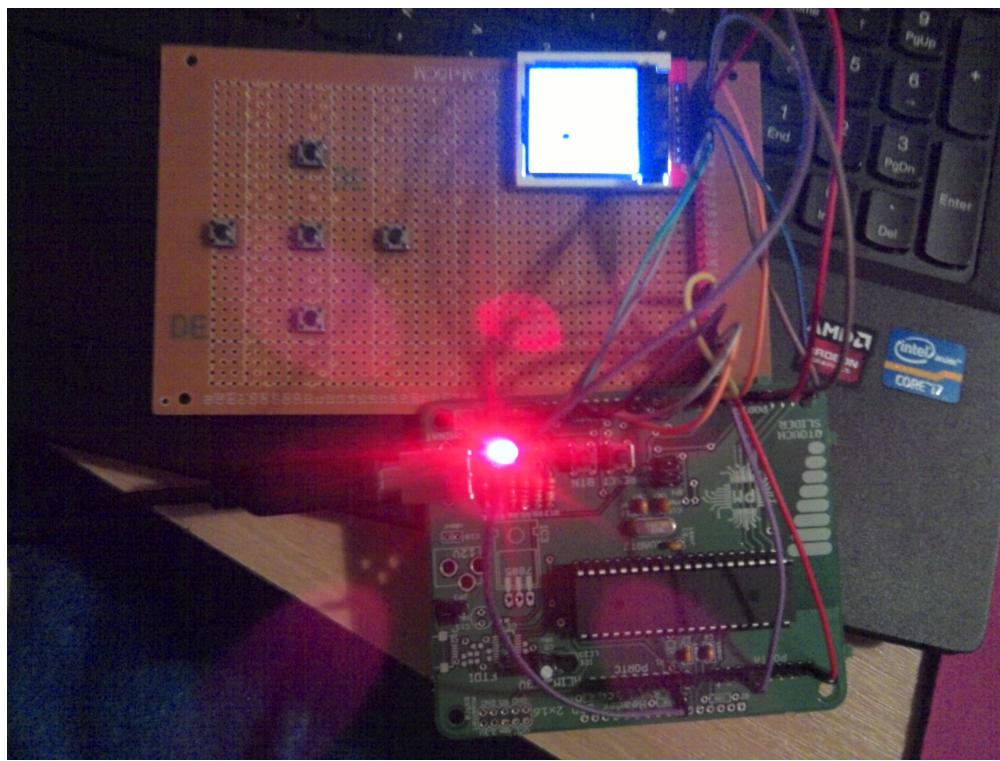
- void setup\_buttons();
- void make\_move();
- void draw\_grid();
- void draw\_X(int16\_t x, int16\_t y, uint16\_t color);
- void draw\_O(int16\_t x, int16\_t y, int16\_t radius, uint16\_t color);
- void draw\_curr\_position(int16\_t x, int16\_t y, uint16\_t color);
- void convert\_mat\_xy\_to\_lcd\_xy(int16\_t x, int16\_t y, int16\_t offset);
- int16\_t convert\_lcd\_x\_to\_mat\_y(int16\_t offset);
- int16\_t convert\_lcd\_y\_to\_mat\_x(int16\_t offset);

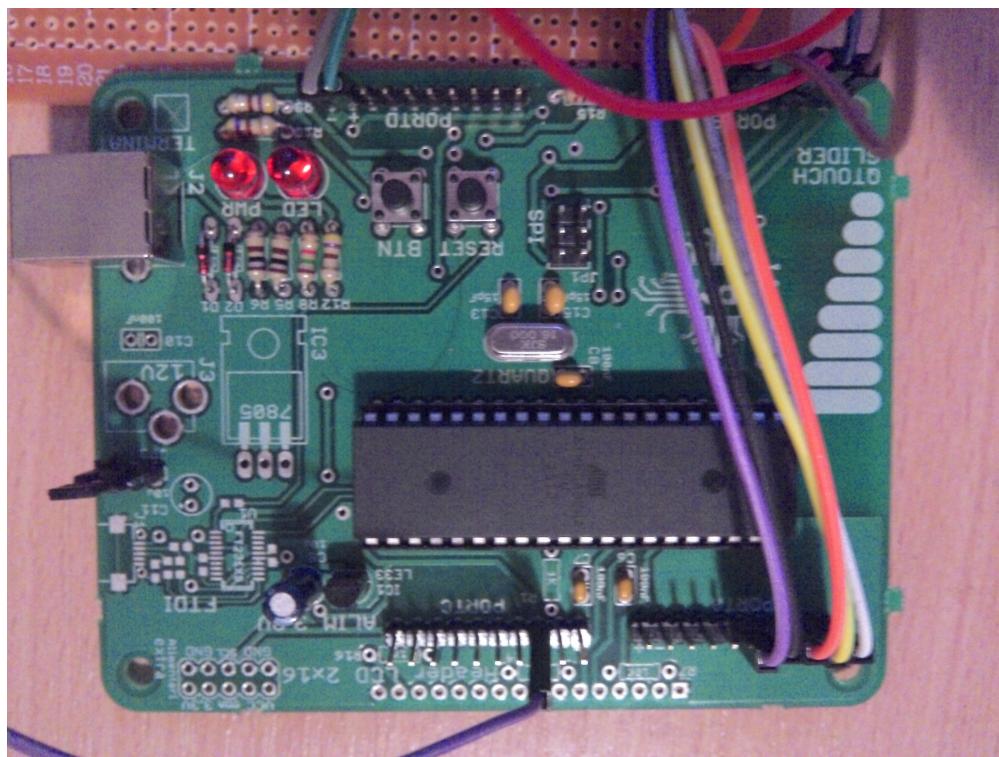
Metode pentru AI si gestionare grid

- bool isInactiveMicroboard(int16\_t x, int16\_t y);
- void getAvailableMoves(Move \*moves, int16\_t \*size);
- bool checkMacro();
- int16\_t checkForVictory(int16\_t lx, int16\_t hx, int16\_t ly, int16\_t hy, bool flag);
- void setWinnerMicroSquare(int16\_t x, int16\_t y, int16\_t winner);
- void clearMacroBoardWhenSentToAWonSquare();
- void unClearMacroBoardWhenSentToAWonSquare();
- void setMove(int16\_t x, int16\_t y, int16\_t player);
- void unMove(int16\_t x, int16\_t y, int16\_t player);
- int16\_t getPlayerIdFromMacro(int16\_t row, int16\_t column);
- void checkX(Move move);
- void boundaries(Move move);
- void setMicroWins();
- int16\_t calculateMacroBoardScore(int16\_t player);
- int16\_t calculateBoardScore(int16\_t lowX, int16\_t lowY, int16\_t player);
- int16\_t evaluate(int16\_t player);
- Move minimax(int16\_t player, int16\_t depth, int16\_t alpha, int16\_t beta);
- bool checkMacroVictory();

## Rezultate Obținute







## Concluzii

A fost un proiect interesant care mi-a oferit sansa sa lucrez si pe partea de hardware, oferindu-mi o mica experienta din care am invatat multe. AI-ul nu am reusit sa il fac sa mearga pe microcontroller, deoarece algoritmul Minimax este recursiv, iar stiva uc-ului este mica, dar algoritmul este corect si are o euristica buna. De asemenea, nu am mai putut adauga alte functionalitati jocului, deoarece fisierul .hex ar depasi cei 32k de flash.

## Download

Surse: [332cb\\_ion\\_bogdan.rar](#)

## Jurnal

1. Milestone 1: Introducere, Schema Bloc, Lista de Piese
2. Milestone 2: Realizare PCB
3. Milestone 3: Schema Electrica
4. Am cumparat LCD, butoane, fire.
5. Am cumparat placă de test.
6. Am lipit LCD și butoane pe placă și am conectat LCD și butoanele la pini.
7. Am inceput să lucrez la software.

## Bibliografie/Resurse

- [1.44" SPI LCD Datasheet](#)
- [1.44" SPI LCD PCB](#)
- [1.44" SPI LCD Schematic](#)
- [ST7735S TFT Controller Datasheet](#)
- [Adafruit-ST7735-Library](#)
- [Adafruit-GFX-Library](#)
- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/ddragomir/1591>



Last update: **2021/04/14 15:07**