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

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

Introducere

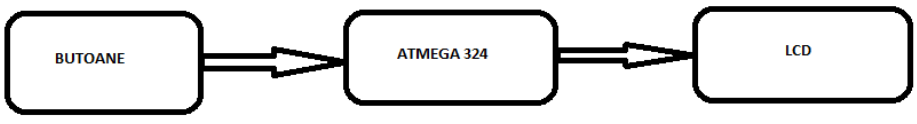
Proiectul consta in implementarea jocului Ultimate Tic-Tac-Toe pe o placa cu LCD.

Scopul proiectului este reprezentat de jocul de X-0 intre un jucator si A.I. sau jucator vs jucator dar pe un grid de 9×9. Atunci cand un jucator pune intr-o casuta intr-un 3×3 grid din grid-ul de 9×9, jucatorul advers este trimis intr-un alt 3×3 grid unde va putea pune seta o alta casuta. Atunci cand un player castiga un grid 3×3 din cel 9×9, se seteaza intr-un MacroGrid(care este 3×3) ca si castigata acea casuta corespunzatoare gridului 3×3. De exemplu, in coltul din stanga-sus al gridului 9×9, am un grid 3×3. In MaroGrid, acel 3×3 grid va fi o casuta 1×1, asociata gridului 3×3 din cel 9×9. Pentru a castiga, se verifica MacroGrid-ul in acelasi mod in care se verifica pentru castigare jocul clasic de Tic-Tac-Toe.

Ideea de la care am pornit este reprezentata de jucarea unui joc Tic-Tac-Toe la alt nivel, in care se evita draw-urile pe cat de mult posibil, ceea ce in jocul clasic nu se prea putea face.

Este util, deoarece reprezinta o provocare mai mare, care necesita multa atentie si logica.

Descriere generală



Exista 5 butoane. 4 vor fi folosite pentru a ne putea deplasa printre casute, si unul pentru a pune X/0 in acea casuta. 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 si surse 3rd-party

Am folosit o librarie care comunica prin SPI cu LCD-ul: lcd_spi_st7735.rar

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

Algoritmi şi structuri pe care plănuiţi să le implementaţi

Pe ecran va aparea 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 in grid, iar prin intermediul celui de al 5-lea va putea pune X/0 intr-o casuta.

Surse şi funcţii 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);

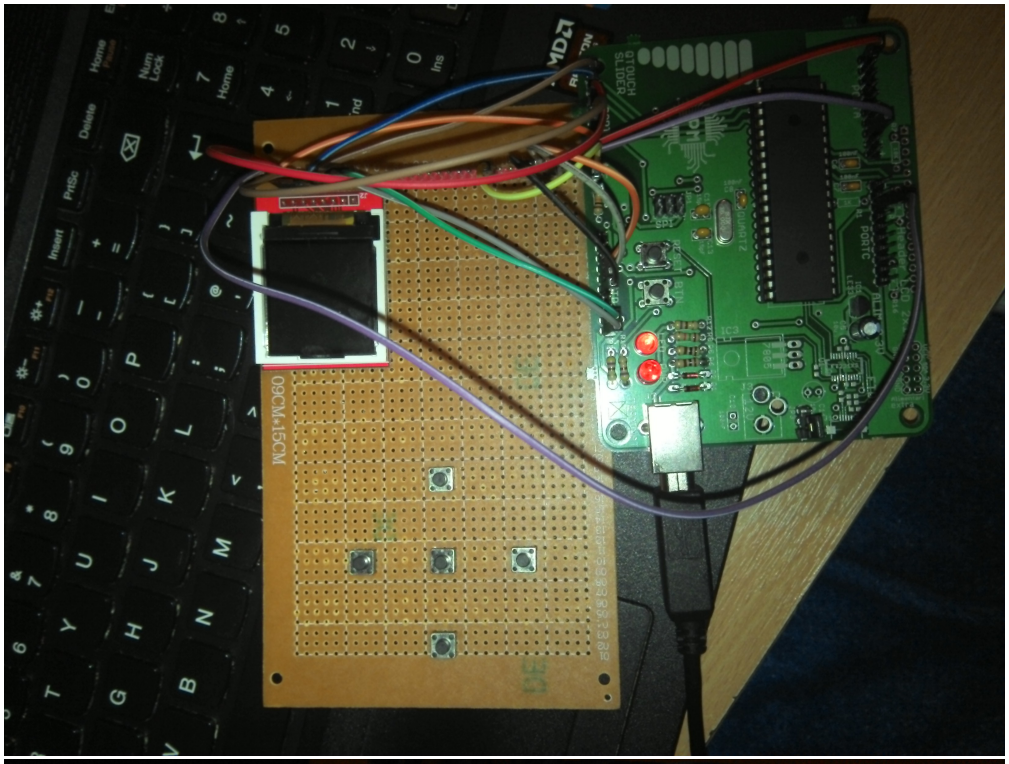
Pentru joc

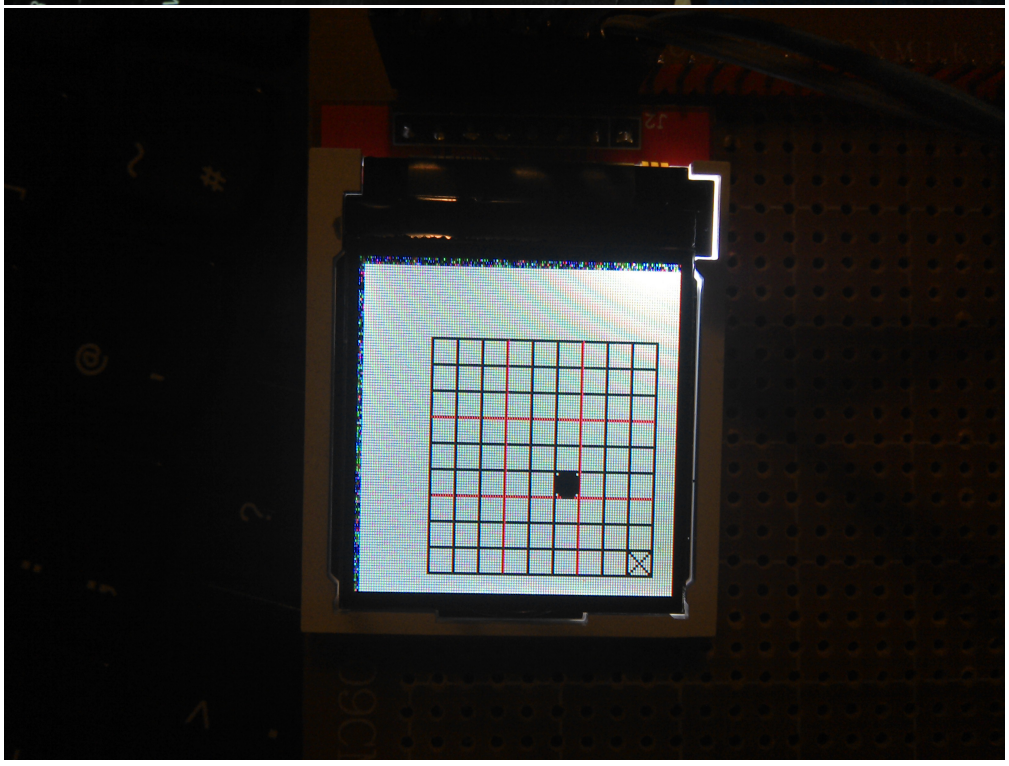
- void setup_buttons();
- void make_move();
- void draw_grid();
- void draw_X(int16_t x, int16_t y, uint16_t color);
- void draw_O(int16_t x, int16_t y, int16_t radius, uint16_t color);
- void draw_curr_position(int16_t x, int16_t y, uint16_t color);
- void convert_mat_xy_to_lcd_xy(int16_t x, int16_t 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);

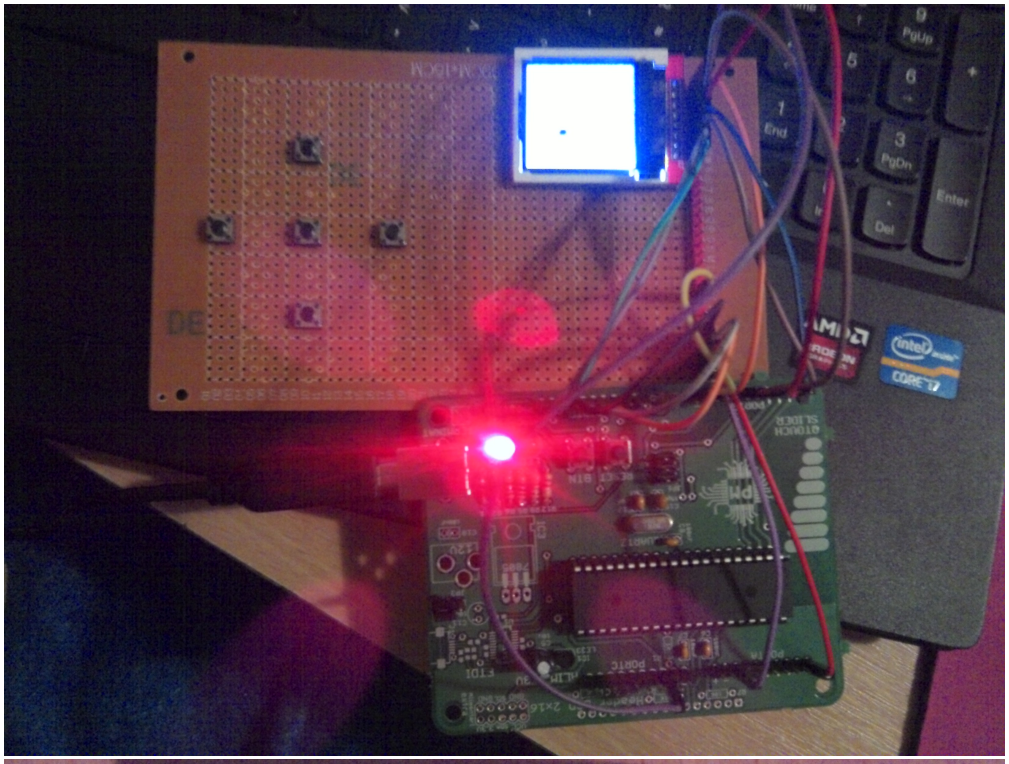
Metode pentru AI si gestionare grid

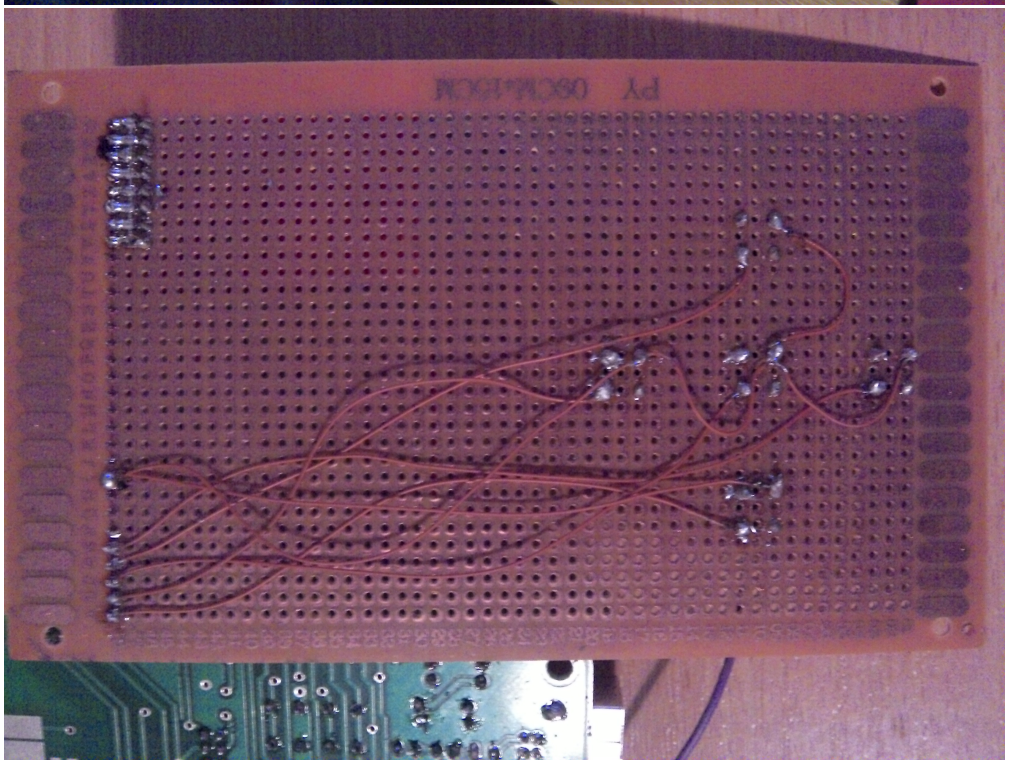
- bool isInActiveMicroboard(int16_t x, int16_t y);
- void getAvailableMoves(Move *moves, int16_t *size);
- bool checkMacro();
- int16_t checkForVictory(int16_t lx, int16_t hx, int16_t ly, int16_t hy, bool flag);
- void setWinnerMicroSquare(int16_t x, int16_t y, int16_t winner);
- void clearMacroBoardWhenSentToAWonSquare();
- 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 ();

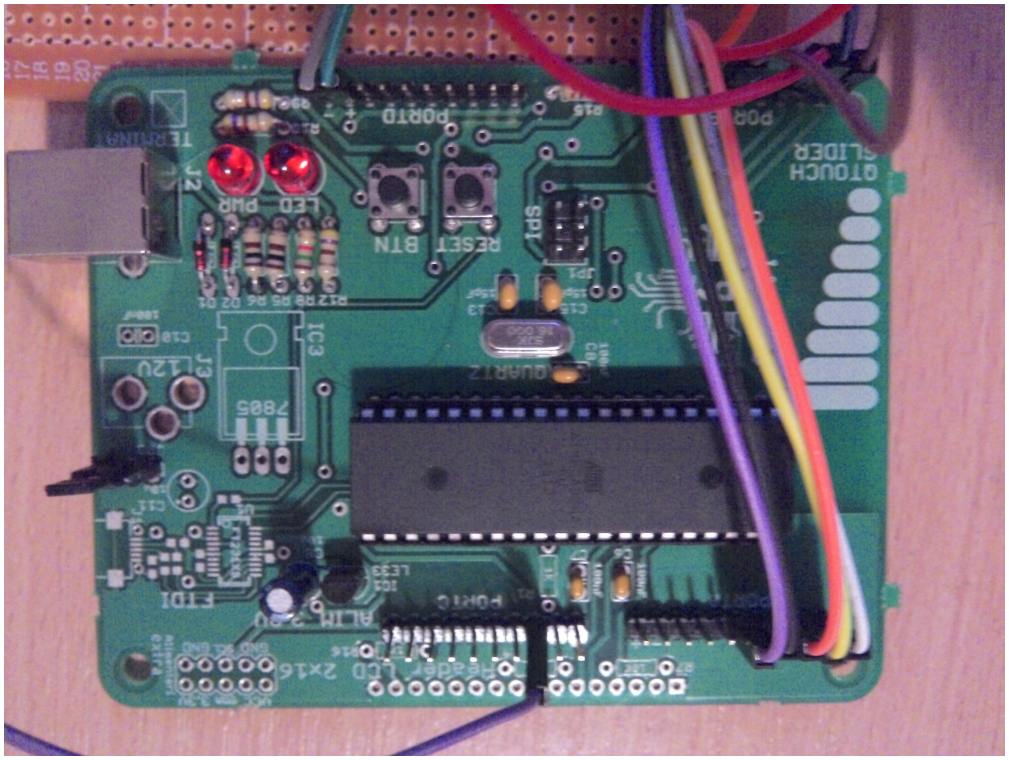
Rezultate Obţinute











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 placa de test.
6. Am lipit LCD si butoane pe placa si am conectat LCD si butoanele la pini.
7. Am inceput sa 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:
https://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
https://ocw.cs.pub.ro/courses/pm/prj2017/ddragomir/1591

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