Jocul Nim

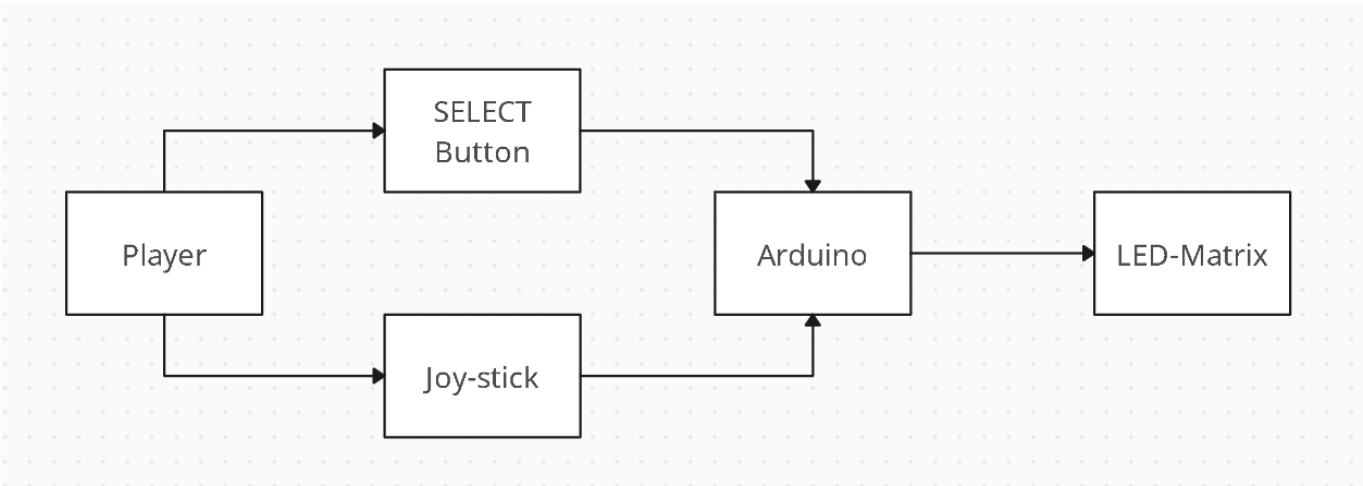
Introducere

Acesta este o platforma pentru a juca jocul Nim atat cu un prieten ,cat si contra un calculator. Acest joc este foarte popular in “Game-theory” si sta la baza jocurilor de tipul “impartial game”. Acest joc este util pentru a intelege bazele acestor concepte intr-un mod mai interactiv.

Descriere generală

Cand placuta este pornita, jucatorul va fi intampinat de un meniu in care va selecta modul de joc, single-player sau two-player. Acest lucru se va face prin utilizare joystick-ului stanga dreapta, pentru a schimba modul de joc (care va aparea pe matricea de LED-uri), si intr-un final apasarea butonului pentru a selecta modul ales. Pentru modul single-player se va putea folosi joy-stick-ul sus jos pentru selectarea dificultatii. Dupa aceasta alegere, jucatorul va putea alege formatul jocului (modul in care sunt distribuite betele) folosind joy-stick-ul stanga dreapta si butonul de SELECT

Odata ales jocul, jucatorul aflat la miscare fi indicat de marginea matricii de LED-uri. Acesta va putea folosi joystick-ul pentru alegerea miscarii dorite. Betele afectate vor fi vizibile prin aprinderea si stingerea acestora. Miscarea este executata cu ajutorul butonului SELECT. Intr-un final cand se termina jocul va fi afisat castigatorul jocului, iar placuta se va intoarce in meniul de selectie al modului de joc. De asemenea jocul se mai termina dupa trecerea unui interval de timp fara ca jucatorul sa faca vre-o miscare, jocul fiind declarat castigat de celalt jucator(Fie ca acesta este alta persoana sau calculatorul).

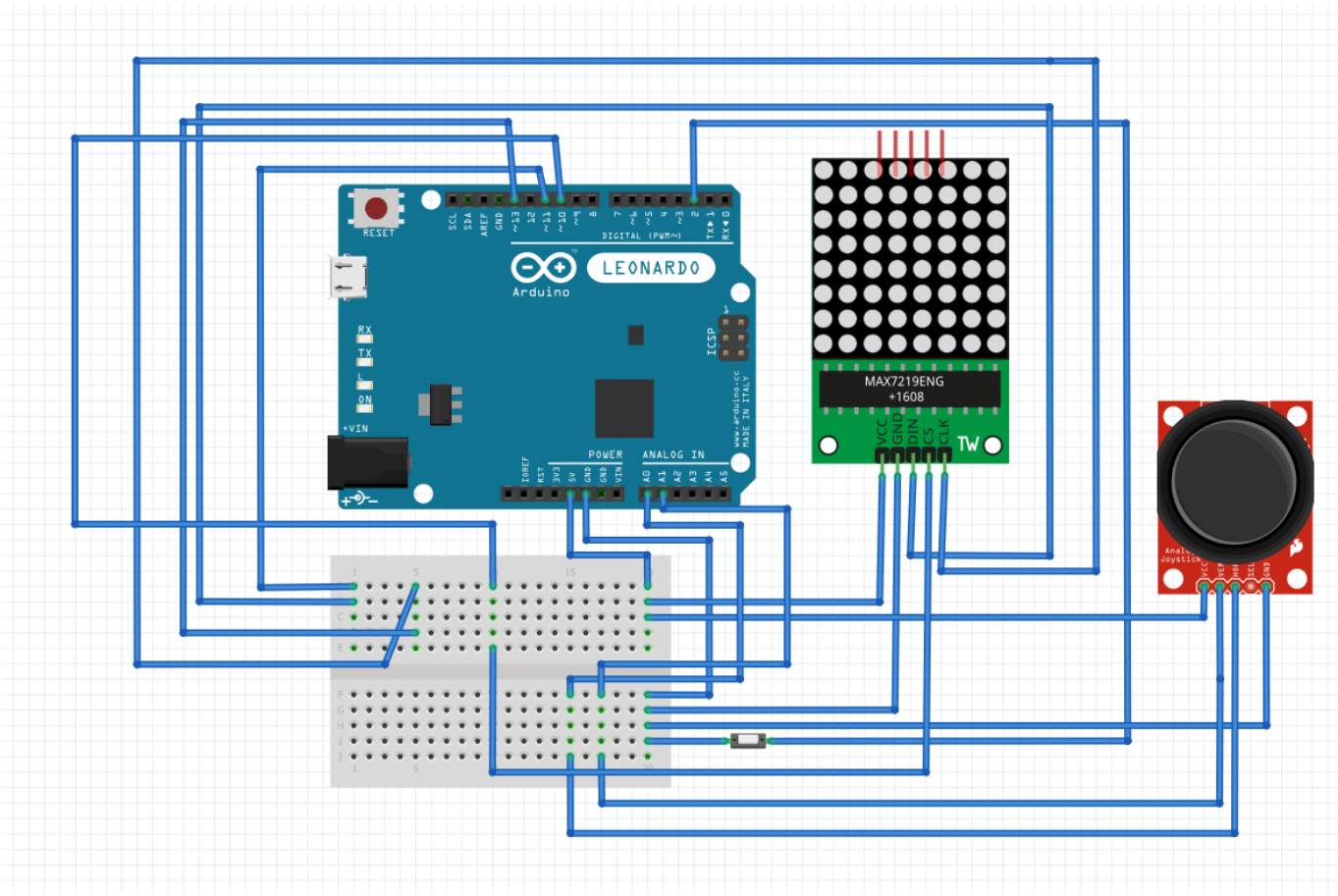


Concepte folosite:

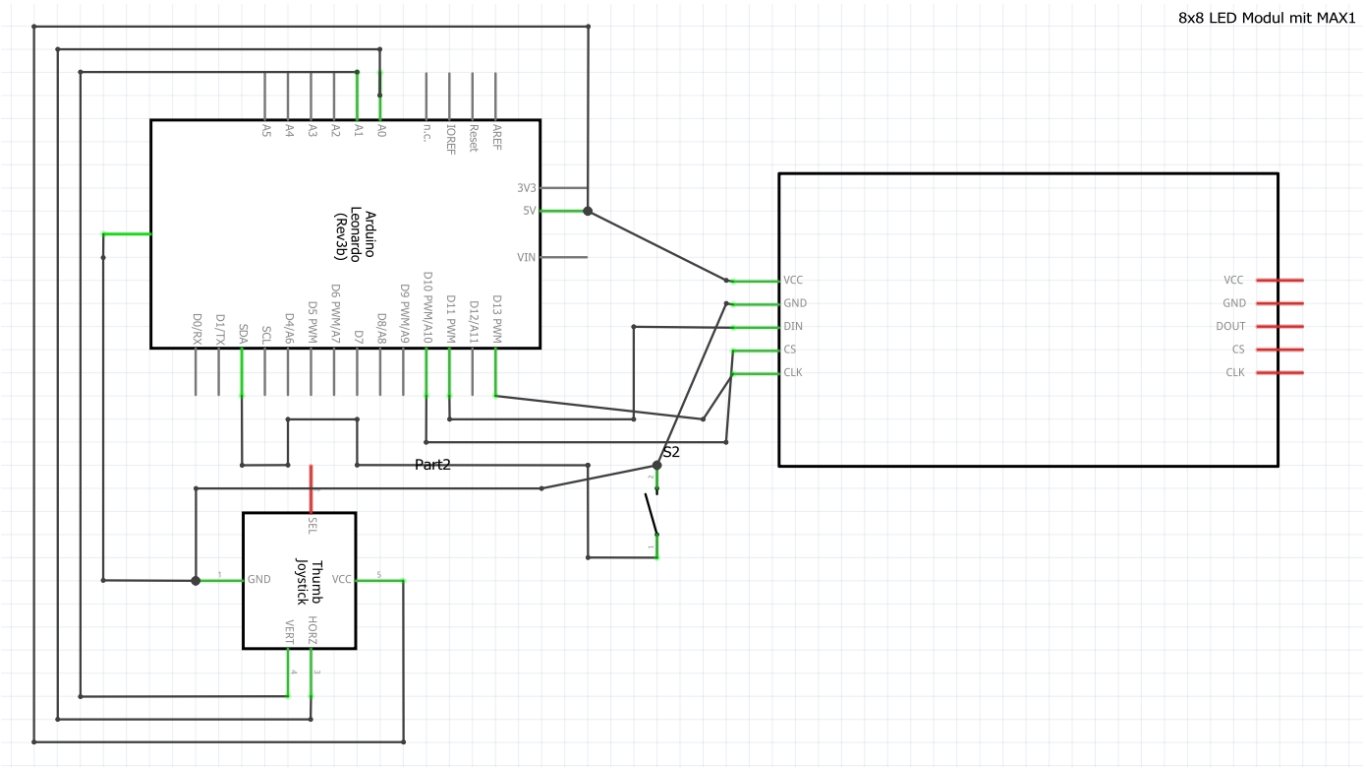
- **GPIO** - Majoritatea proiectului afiseaza tot ce se intampla cu ajutorul matricei de LED-uri
- **Întreruperi** - Sunt folosite pentru toate apasariile de buton.
- **ADC** - Folosit la citirea input-urilor de pe joystick.
- **SPI** - Biblioteca LedControl transimite matricei de LED-uri pe interfata SPI

Hardware Design

Schema circuit:



Schema electrica:



Lista de componente:

- Arduino Leonardo
- Breadboard
- Matrice LED-uri 8×8 MAX7219
- Modul Joy-stick
- Buton
- Fire de legatura

Modul de utilizare al componentelor:

Arduino Leonardo Microcontroller care proceseaza codul, are rolul de a lega toate functionalitatiile fiecarei componente. El se ocupa de citirea input-urilor de la joystick si buton, procesarea lor pentru a calcula starea de joc (sau starea in meniu)si afisarea unui rezultat pentru utilizator pe matricea de led-uri

Matrice LED-uri 8×8 MAX7219

Acesta este folosit pentru afisarea jocului si pentru testarea functionarii corecte a programului.

Matricea are 5 pini care sunt conectati astfel:

- VCC - conectat la pinul de 5V de la Arduino
- GND - conectat la toate celelalte GND
- DIN - conectat la pinul digital ~11 (MOSI)
- CS - conectat la pinul digital ~10 (SS)

- CLK - conectat la pinul digital ~13 (SCK)

Modul Joystick

Acesta este folosit pentru citirea input-ului jucator.

Modulul are 5 pini care sunt conectati astfel:

- GND - conectat la toate celelalte GND
- VCC - conectat la pinul de 5V de la Arduino
- VRx- conectat la pinul analog A1, citeste input-urile pe axa Ox
- VRy - conectat la pinul analog A0, citeste input-urile pe axa Oy
- SW - este pinul pentru butonul de pe joystick dar nu este utilizat

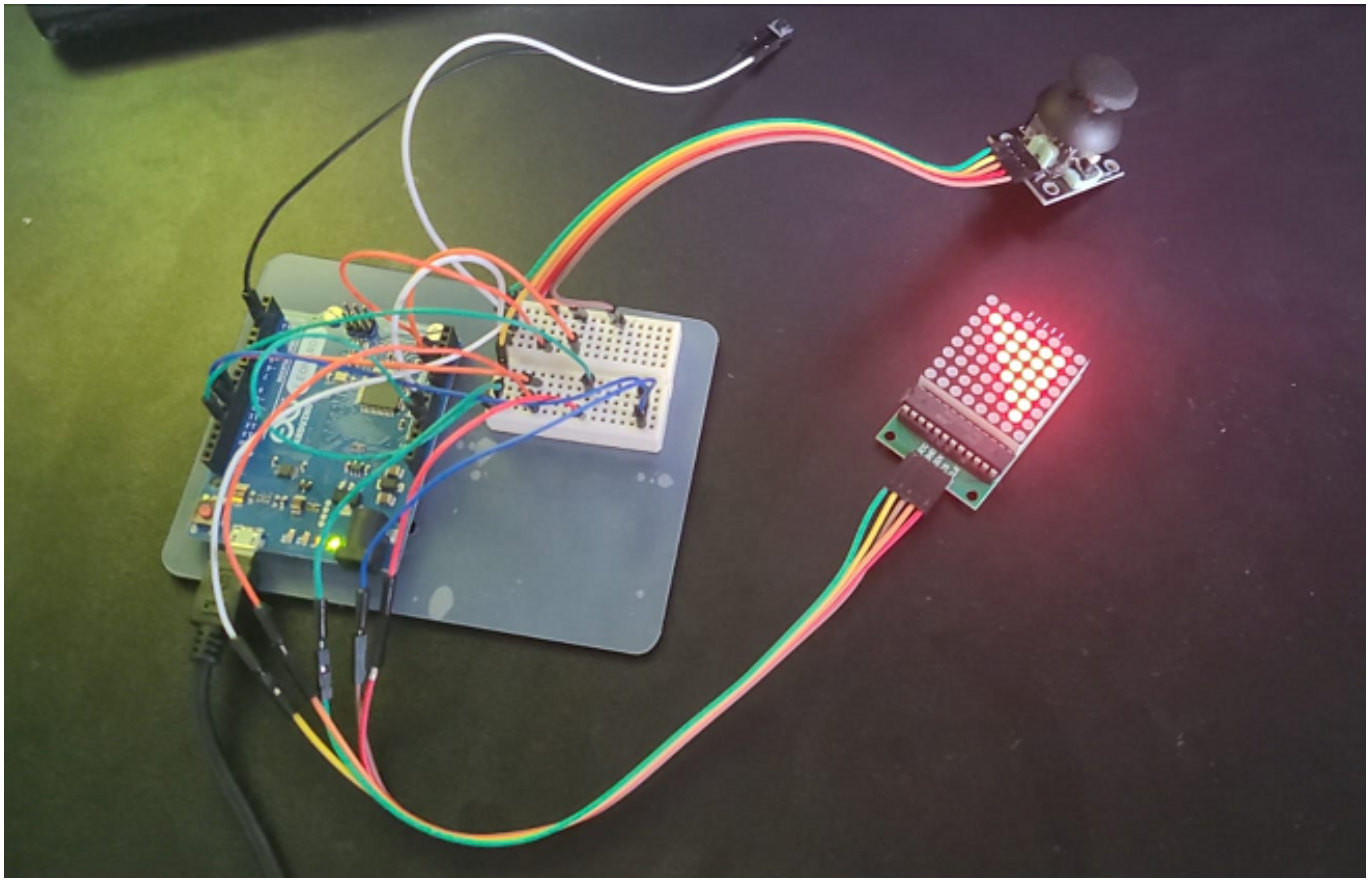
Buton

Acesta este folosit pentru citirea input-ului jucator.

Butonul are 2 pini:

- GND - conectat la toate celelalte GND
- BTN - conectat la pinul digital 2, citeste daca butonul e apasat

Poza cu proiectul:



Demo

Software Design

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare: ArduinoIDE
- librării: LedControl.h

Github: Nim Game

Biblioteci si Constante

```
#include <LedControl.h>

#define NEUTRAL 510
#define MAX 1023
#define ERR 10
#define SWITCH_PLAYER 3
#define RANDOM 3
#define NUM_MAPS 4
```

```
#define MAP_SIZE 6
#define DEBOUNCE_DELAY 300
#define CPU_MOVE 300
#define G2P 2
#define GCPU 1
#define HARD 2
#define RESET_TIME 120000
```

Biblioteca LedControl este utilizata pentru a controla matricea de LED-uri prin intermediul interfetei SPI.

Definitii de pini

```
int DIN = 11; // Pinul pentru date
int CS = 10;  // Pinul pentru selectarea cipului
int CLK = 13; // Pinul pentru ceas
LedControl lc = LedControl(DIN, CLK, CS, 0);

int xPin = A0;
int yPin = A1;
int buttonPin = 2;
```

Imagini pentru afisare

```
byte DIFFICULTY[3][8] = { ... };
byte WIN[2][8] = { ... };
byte EMPTY[8] = { ... };
byte P2[8] = { ... };
byte P1[8] = { ... };
byte QUESTION_MARK[8] = { ... };
byte NUM_PLAY[8][3];
byte Disp[8];
```

Tabele reprezentand diferite stari ale afisajului: niveluri de dificultate, ecrane de castig, indicatori de jucatori, etc.

Configuratiile hartii

```
int map_layout[3][6] = { ... };
int current_map[6];
```

Definirea configuratiilor stariilor initiale de joc si o variabila pentru a tine harta curenta.

Functii ale jocului

- **reset_game()**: Reseteaza variabilele jocului la starile lor initiale.
- **setup()**: Initializeaza pinii, comunicarea seriala, matricea LED si configureaza intreruperea pentru buton.
- **show_display()**: Actualizeaza matricea LED cu datele actuale ale afisajului.
- **difficulty_select()**: Gestioneaza intrarea de la joystick pentru selectarea nivelului de dificultate.
- **menu_org()**: Gestioneaza meniul principal pentru selectarea numarului de jucatori.
- **convert_map_to_disp()**: Convertește un array de harta in formatul de afisare.
- **random_map()**: Genereaza o configuratie aleatorie a hartii.
- **reset_cursor()**: Reseteaza pozitia cursorului la prima coloana nevida.
- **map_select()**: Gestioneaza selectarea unei configuratii de harta.
- **player_move()**: Gestioneaza miscarea jucatorului bazata pe intrarea de la joystick.
- **make_move()**: Efectueaza o mutare in joc a CPU-ului.
- **CPU_h_move()**: Logica pentru mutarile CPU pe modurile HARD si MEDIUM.
- **CPU_e_move()**: Logica pentru mutarile CPU pe modul EASY.
- **win_game()**: Verifica daca jocul a fost castigat si afiseaza animatia de castig.
- **game_manager()**: Gestioneaza logica generala a jocului si alterneaza intre mutarile jucatorilor si cele ale CPU-ului.

Main loop

```
void loop() {
    if (menu > 0) {
        menu_org();
    } else {
        game_manager();
    }
    show_display();
    for (int i = 0; i < 8; i++) Disp[i] = EMPTY[i];
}
```

Bucla principala care comuta intre meniu și modurile de joc si actualizeaza afisajul.

Rezultate Obţinute

Implementarea acestui proiect a dus la crearea unui joc complet funcțional pe o matrice de LED-uri, controlat printr-un joystick și un buton. Am reușit să gestionăm corect intrările analogice și digitale, să implementăm o logică de joc pentru două jucătoare și să integrăm un oponent CPU cu diferite niveluri de dificultate. Proiectul a demonstrat capacitatea de a utiliza biblioteca LedControl pentru a controla o matrice LED 8×8, de a debuga și testa eficient codul pentru a elimina erorile și de a crea o experiență interactivă de joc. Acest proiect a oferit o bază solidă pentru dezvoltarea unor proiecte mai complexe, implicând automatizarea și controlul hardware-ului prin programare.

Concluzii

Implementarea acestui proiect a implicat gestionarea intrărilor analogice și digitale, cum ar fi joystick-ul și butonul, pentru a controla jocul. De-a lungul procesului de dezvoltare, s-au întâlnit diverse erori și probleme, necesitând testare și depanare constantă pentru asigurarea funcționării corecte a codului. Cu toate acestea am obținut o introducere în crearea unui proiect folosind automatizările și cred că acum sunt pregătit să întâmpin proiecte și mai dificile.

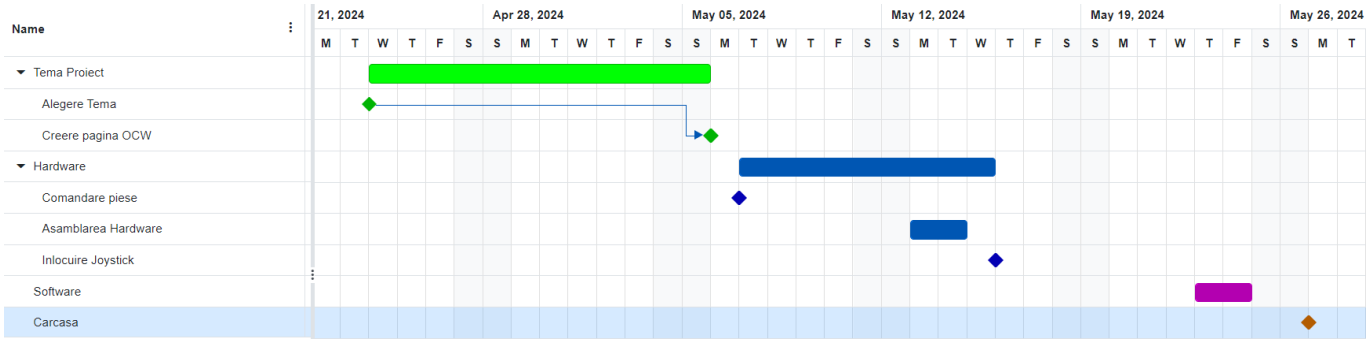
Download

ZIP cu proiectul:

nim-game.zip

Jurnal

- 05/05/2024 - Creare pagină wiki.
- 13/05/2024 - 14/05/2024 - Construire prima versiune de circuit.
- 16/05/2024 - Inlocuire Joystick defect
- 23/05/2024 - 24/05/2024 - Implementare Software
- 25/05/2024 - Update final documentatie



Bibliografie/Resurse

Guide intreruperi Arduino

Datasheet Arduino Leonardo

Guide Max7219 Matrice 8x8

Guide folosit pentru conectarea corecta a pinilor la matrice

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/matei.costescu

Last update: **2024/05/25 20:49**