

Sudoku

Introducere

Introducere Acest proiect are ca scop realizarea unei versiuni fizice a jocului clasic Sudoku. Jocul va fi afișat pe un ecran LCD, navigarea printre celulele grilei se va face cu ajutorul unui joystick, iar introducerea numerelor se va realiza prin intermediul unui keypad.

Obiectivul proiectului este de a crea o interfață interactivă și fizică pentru rezolvarea puzzle-urilor Sudoku, combinând elemente de programare, electronică și logică.

Descriere generală

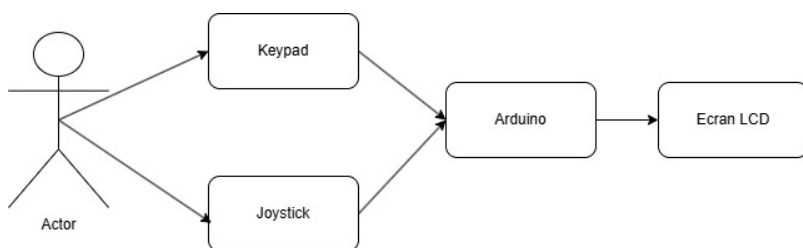
Descriere generală Sudoku este un joc de logică ce presupune completarea unei grile de 9×9 cu cifre de la 1 la 9, respectând anumite reguli stricte:

Fiecare rând trebuie să conțină o singură dată cifrele de la 1 la 9.

Fiecare coloană trebuie să conțină o singură dată cifrele de la 1 la 9.

Fiecare bloc de 3×3 celule (regiune) trebuie să conțină toate cifrele de la 1 la 9, fără repetiții.

În cadrul proiectului, ecranul LCD are rolul de a afișa grila și cifrele corespunzătoare, oferind utilizatorului o experiență vizuală clară a jocului. Introducerea cifrelor se face cu ajutorul keypad-ului, iar joystick-ul permite deplasarea între celule pentru selectarea poziției dorite.



Hardware Design

Lista componente

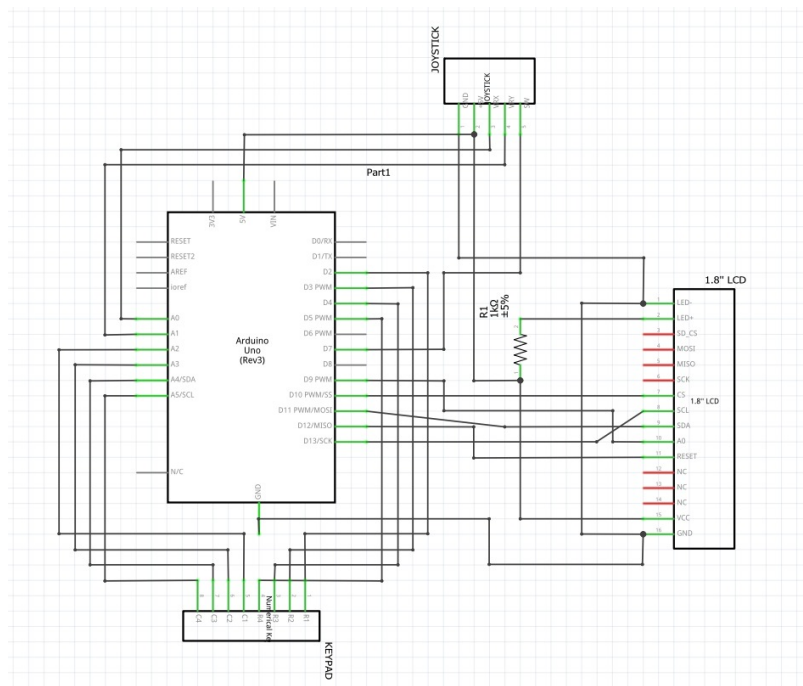
1. [Arduino UNO](#)
2. [Ecran LCD 1.8" SPI](#)
3. [Tastatura numerica 4*4](#)

4. Joystick

5. Breadboard 400 de puncte

6. Rezistente + fire

Schema electrica:



Software Design

Pentru realizarea componentei software am folosit mediul de dezvoltare Arduino IDE. Aplicatia este un joc de **Sudoku** care ruleaza pe un microcontroler si foloseste un **ecran TFT**, un **joystick analogic** si o **tastatura 4x4**.

Biblioteci folosite:

- SPI.h – pentru comunicarea cu ecranul TFT.
- TFT.h – pentru desenarea interfetei grafice.
- Keypad.h – pentru detectarea tastelor apasate pe tastatura matriciala.

Structura codului si functiile folosite:

• 1. Initializare componente (in setup())

Se configureaza joystick-ul, tastatura si ecranul TFT. Se calculeaza dimensiunea fiecarei celule (cellWidth, cellHeight) in functie de rezolutia ecranului. Se apeleaza functia drawMainMenu() pentru a afisa meniul principal.

• 2. Meniu principal (functia drawMainMenu())

Afiseaza pe ecran titlul "SUDOKU" si instructiuni:

- A pentru joc usor,
- B pentru joc greu,

- C pentru super usor (tabla aproape completa),
- * pentru verificarea jocului,
- # pentru revenire in meniu.

• 3. Incarcarea tablei de joc (functia loadBoard(index))

Incarca o tabla predefinita din predefinedBoards[index]. Marcheaza celulele care nu pot fi modificate (fixedCells[]). Initializeaza variabila filledCells. Apeleaza drawSudokuGrid() si drawBoldGridLines() pentru afisare.

• 4. Afisarea tablei Sudoku:

- drawSudokuGrid() Deseneaza fiecare celula. Daca are o valoare, o afiseaza (text alb). Evidentiaza celula selectata cu un chenar rosu. - drawBoldGridLines() Deseneaza liniile groase pentru a marca fiecare subgrila 3x3.

• 5. Navigare si interactiune (in loop())

Citeste pozitia joystick-ului (analogRead(X) si analogRead(Y)) si modifica selectedRow / selectedCol. Daca utilizatorul apasa o tasta intre 1 si 9, se apeleaza writeToCell(key).

• 6. Introducere cifre (functia writeToCell(key))

Verifica daca celula este editabila (fixedCells). Daca da, salveaza cifra si coloreaza celula pentru a arata ca a fost modificata.

• 7. Validarea jocului (functia validateBoard())

Daca tabla nu este completata (filledCells < TOTAL_CELLS), afiseaza mesajul "NOT FINISHED!". Daca este completata, verifica: toate liniile, toate coloanele, toate subgrilele 3x3.

Daca totul e corect, afiseaza "YOU WON!". Daca sunt greseli, afiseaza "TRY AGAIN!". In toate cazurile, dupa afisarea mesajului se redeseneaza tabla (drawSudokuGrid() si drawBoldGridLines()).

• 8. Revenire in meniu (functia resetToMenu())

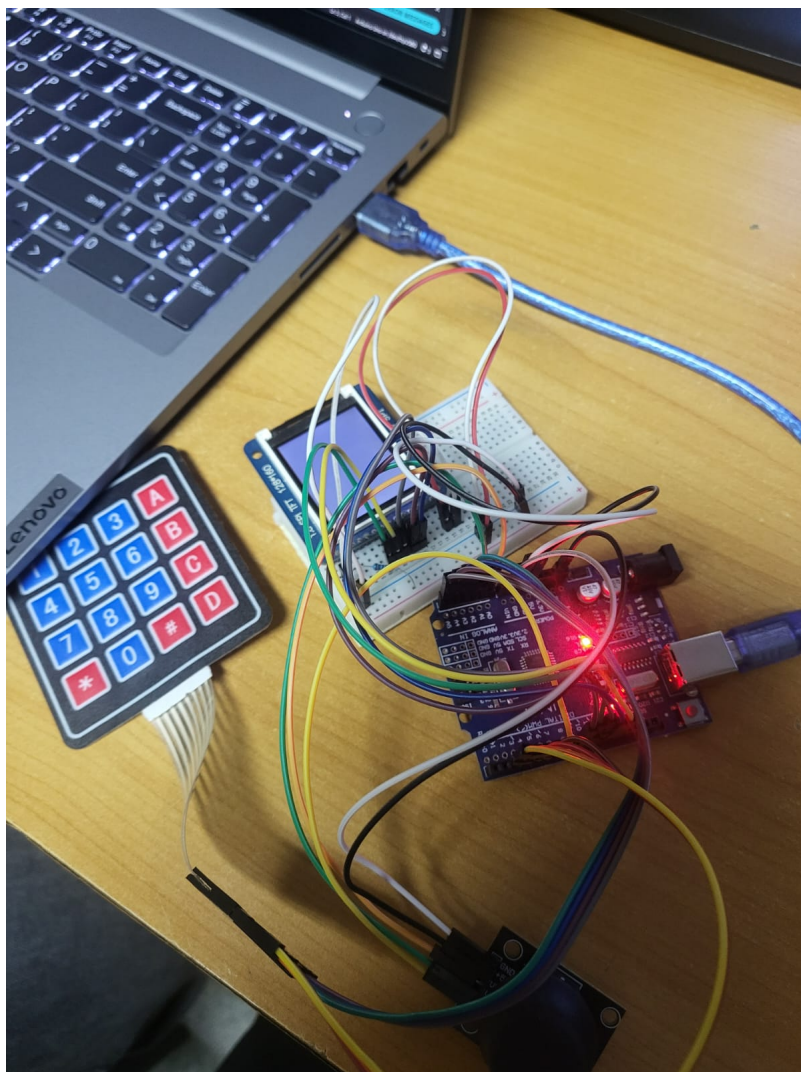
Reseteaza valorile de stare (isInMenu, selectedRow, filledCells, etc.). Curata ecranul si afiseaza din nou meniul (drawMainMenu()), unde codul este deja optimizat:

Optimizari posibile: Debounce pentru joystick - ar reduce miscarile involuntare. Butonul joystick-ului (joystickButtonPin) nu este folosit - ar putea fi adaugata o functie de stergere celula. Feedback vizual pentru greseli - se pot colora celulele incorecte. Verificare in timp real - pentru a preveni introducerea cifrelor care incalca regulile. Oprea mai devreme la validare - algoritmul verifica tot jocul chiar daca gaseste o greseala; se poate optimiza.

Notiuni din laborator aplicate:

- SPI - pentru comunicarea cu ecranul TFT.
- ADC - pentru joystick (citirea valorilor analogice).
- GPIO - pentru tastatura si butoane.

Rezultate Obținute



Concluzii

- Proiectul arata ca poti construi o aplicatie interactiva completa pe un microcontroler, combinand afisaj grafic, control cu joystick si tastatura.
- Jocul ofera o experienta placuta si intuitiva.
- Sudoku-ul este verificat automat, ceea ce implica o logica bine gandita si o coordonare clara intre input si afisaj.
- M-am distrat :)

Cod sursă

[Link download](#)

[Link Demo](#)[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/diana_ioana.tudor



Last update: **2025/05/25 17:31**