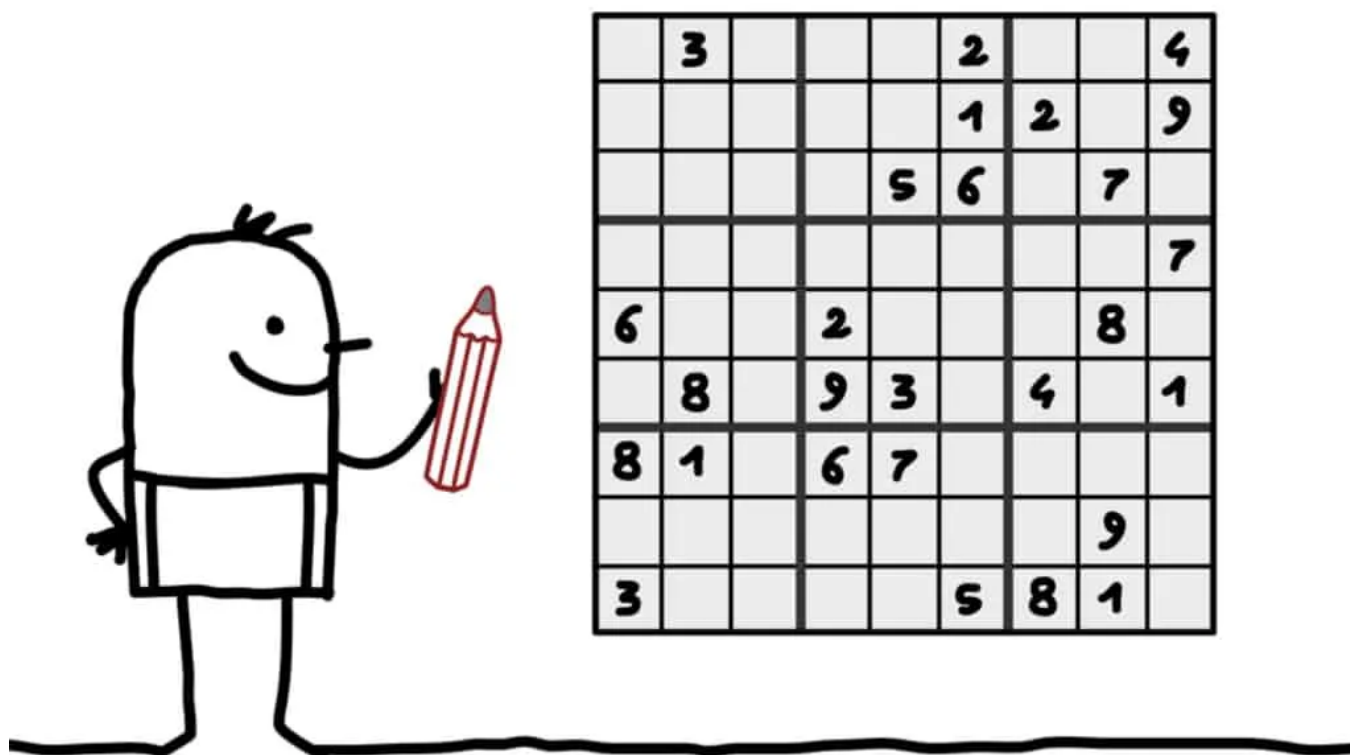


Sudoku

Introducere



Proiectul meu constă în implementarea jocului Sudoku. Jucătorul va avea la dispoziție o telecomandă cu care va completa căsuțele afișate pe un ecran LCD.

Scopul proiectului este fixarea noțiunilor acumulate pe parcursul laboratoarelor de PM, dar poate fi folosit și pentru delectarea personală, fiind nevoie doar de actualizarea periodică a puzzle-urilor din joc.

Obișnuiam să rezolv Sudoku în copilărie așa că ideea mi-a venit natural. Consider că acest proiect poate fi util pentru pasionații de Sudoku deoarece scrierea unor rezolvări ipotetice și necesitatea ulterioară de ștergere a acestora este probabil cea mai mare problemă cu care se comportă practicanții pe suport de hârtie. Pe suport digital valorile din căsuțe pot fi înlocuite/șterse fără să apară mâzgălituri și întinări, jocul păstrându-și astfel claritatea.

Descriere generală



Hardware Design

Lista de piese:

- * Placă de dezvoltare Mega 2560 CH340 + cablu USB – Compatibilă Arduino
- * Breadboard 400 puncte
- * LED RGB, 10mm, catod comun
- * Modul Buzzer pasiv
- * Buton fara retinere, PBS-110, 7mm, NO
- * Kit rezistori 30 valori, 600 bucati, 1/4W, 10R-1M
- * 40 Fire Dupont 20cm, Tata-Tata
- * Modul IR Receptor plus telecomanda
- * Display 3.5, 320x480, compatibil Arduino, ILI9486, 8Bit



Software Design

Biblioteci folosite:

- IRremote - pentru telecomanda si senzor IR
- LCDWIKI_GUI - pentru LCD
- LCDWIKI_KBV - pentru LCD

Am construit diferite functii pentru tratarea interactiunii cu buzzer-ul, senzorul IR, butonul, led-ul RGB sau pentru testarea corectitudinii solutiei.

Jucatorul apasa butonul pentru a incepe jocul si apoi foloseste o telecomanda pentru a muta cursorul de la o casuta la alta si pentru a introduce cifrele. Daca jocul este completat, un led rgb va lumina rosu sau verde in functie de corectitudinea solutiei. De asemenea, o solutie corecta va fi acompaniata de un sunet victorios scos de buzzer.

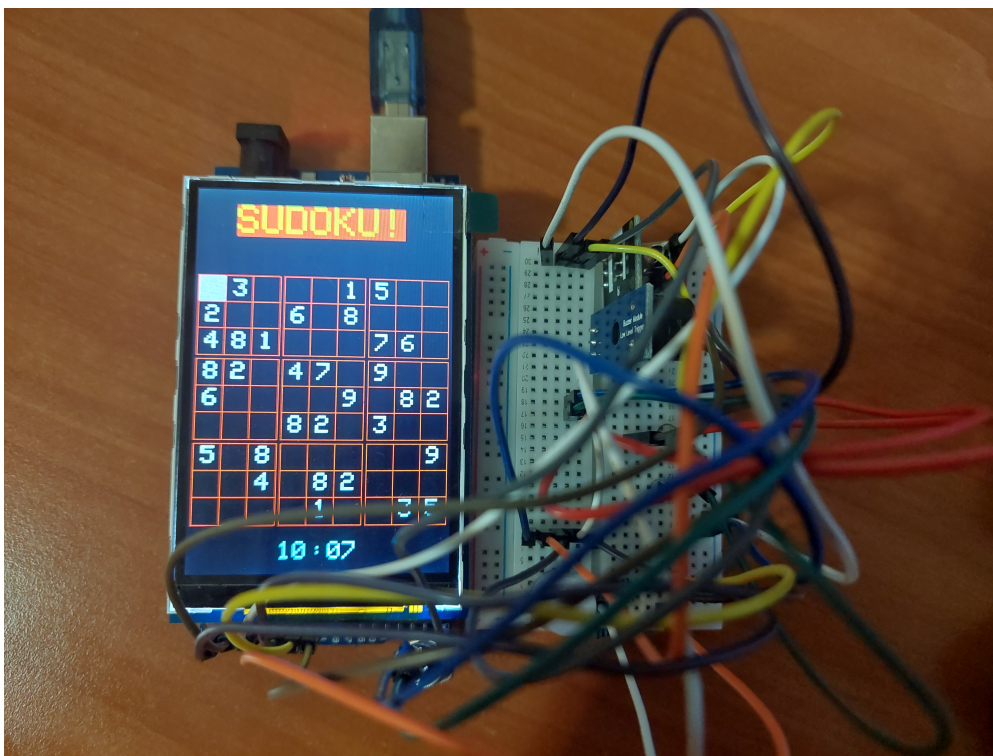
Functii implementate:

- `get_grid_and_numbers_positions()` → calculeaza pozitia pe ecran a grid-ului si a cifrelor
- `configurare_menu()` → construiește meniul initial
- `manage_LED()` → controleaza starea ledului
- `RGB_color()` → seteaza culoarea ledului
- `turn_off()` → opreste ledul
- `get_input()` → interpreteaza inputul primit de la telecomanda

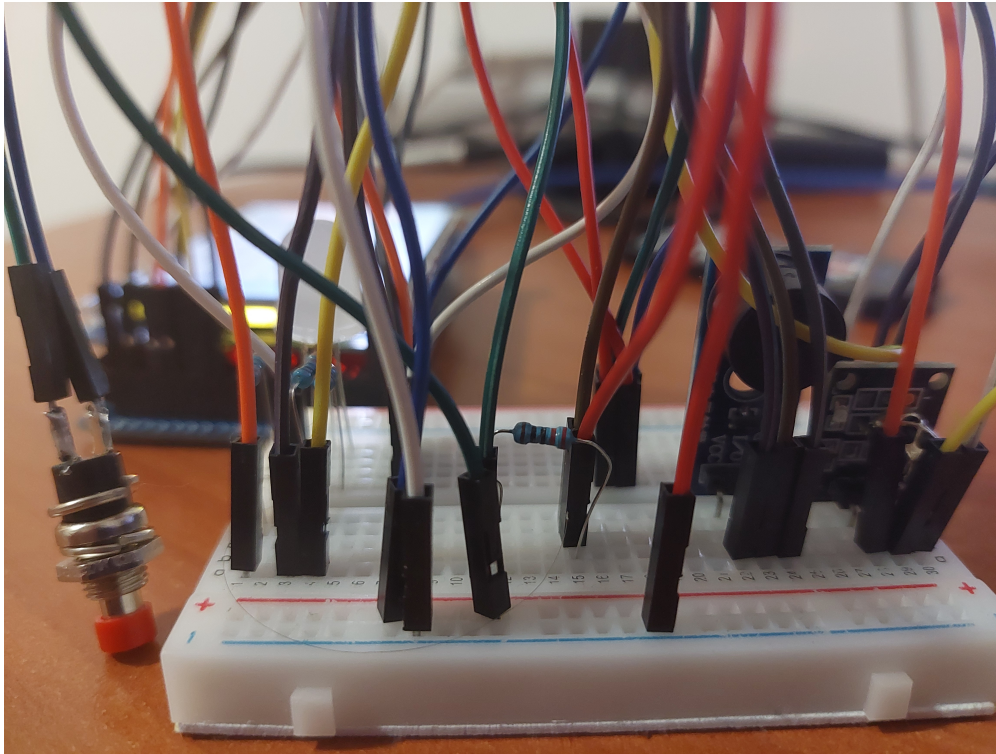
- `print_grid()` → afiseaza grid-ul
- `print_numbers()` → afiseaza cifrele
- `display_timer()` → construiește și afisează cronometrul
- `check_solution()` → verifică corectitudinea soluției
- `game_over()` → afisează mesajul "GAME OVER", porneste o melodie redată de buzzer și oferă posibilitatea întoarcerii la meniul inițial
- `congratulations()` → afisează mesajul "CONGRATULATIONS!", porneste o melodie redată de buzzer și oferă posibilitatea întoarcerii la meniul inițial

Rezultate Obținute

Demo: [Sudoku_demo](#)







Concluzii

A fost interesant sa descopar mai multe despre functionarea unui senzor IR si a unei telecomenzi in general, precum si interactiunea cu un display LCD. Limitarile de pini ale placutei Arduino Uno ce m-au determinat sa achizitionez o noua placuta Arduino Mega 2560, arderea unui senzor IR , tresarirea la primul zgomot asurzitor scos de buzzer, fumul scos de componentele incinse si socurile electrice resimtite au facut din acest proiect o experienta aparte.

Download

[sudoku.rar](#)

Jurnal

- 07.05.2022 - Descrierea proiectului si fixarea listei de componente necesare
- 14.05.2022 - Conectarea si testarea tuturor componentelor
- 18.05.2022 - Desenarea grid-ului si facilitarea introducerii numerelor in casute
- 24.05.2022 - Implementarea functiei de verificare a solutiei si a interactiunii cu utilizatorul (buton, buzzer, led)
- 27.05.2022 - Finalizarea documentatiei

Bibliografie/Resurse

- 3.5inch_Arduino_8BIT_Module_ILI9486_MAR3501_V1.1
- KY-022 Infrared receiver module Datasheet
- https://github.com/gitcnd/LCDWIKI_GUI
- <https://github.com/robsoncouto/arduino-songs>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/ionuto/sudoku>



Last update: **2022/06/01 23:00**