

# Sudoku

- DUDU Matei-Ioan
- 333CB

## Introducere

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

aferente.

În cadrul acestui proiect voi implementa (probabil) foarte cunoscutul joc Sudoku pe o placuță Arduino; joc pe care probabil toată lumea l-a jucat măcar o dată pe telefoanele cu taste ale bunicilor lor.

Ideea a fost să creez ceva ce se află mai mult în sfera cunoștințelor mele (software heavy) și totodată interactiv.

## Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>



## Hardware Design

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

- placuță arduino
- buzzer pasiv de 5V
- modul LCD SPI de 1.8" (128×160)
- SD card reader (inclus în modulul LCD)
- joystick
- breadboard
- rezistențe și fire



## Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuieți să le implementați
- (etapa 3) surse și funcții implementate

- Am lucrat în VSCode, folosind extensia PlatformIO
- Am folosit următoarele librării: Ucglib, SD și SPI
- Am implementat algoritmul de verificare a corectitudinii unei rezolvarii făcând maparea numerelor deja folosite pe biti
- Structurile folosite sunt Point (pentru un punct în spațiul 2D) și Color (r, g, b)

Variabile globale folosite:

- constante folosite pentru pinii asociați elementelor precum: buzzer, buton joystick etc.
- Ucglib\_ST7735\_18x128x160\_SWSPI ucg: obiect folosit pentru gestionarea ecranului
- sudoku[9][9]: matrice folosită pentru memorarea configurației puzzle-ului
- selected\_square, selected\_number, and game\_mode: variabile folosite pentru a verifica starea jocului
- no\_placed: variabilă folosită pentru a număra câte numere au fost plasate pe grilă

Funcții:

- check\_if\_sudoku\_is\_valid: Verifică validitatea soluției Sudoku.
- place\_number: Afișează un număr pe grilă.
- select\_square: Evidențiază un pătrat selectat pe grilă.

- `unselect_square`: Deselectează un pătrat.
- `is_selected_number`: Verifică dacă un pătrat este numărul selectat în prezent.
- `draw_guidelines`: Desenează liniile care evidențiază subgrilele de 3×3.
- `set_grid`: Inițializează și desenează grila Sudoku pe LCD.
- `undraw_select_arrow` și `draw_select_arrow`: Gestionează săgeata pentru selecția dificultății.
- `set_difficulty_select`: Afișează ecranul de selecție a dificultății.
- `update_position`: Actualizează poziția selectată pe baza intrării de la joystick.
- `update_number`: Actualizează grila Sudoku pe baza intrării de la buton.

Funcția `setup` inițializează ecranul, pinul pentru buton și comunicarea cu cardul SD

Funcția `loop` verifică în ce stare se află jocul: în ecranul de selecție a dificultății sau în timpul jocului și apelează funcțiile

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

Proiect misto. Mi-a starnit cheful de a mai face lucruri similare.

Din pacate nu mai am timp sa cumpar un arduino nano si sa pun totul pe un pcb universal, dar sigur voi face asta dupa prezentarea proiectului.

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul `:pm:prj20??:c?` sau `:pm:prj20??:c?:nume_student` (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → `:pm:prj2009:cc:dumitru_alin`.

[main.zip](#)

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

- 03.05.2024 - creare pagină
- 11.05.2024 - testare funcționare corectă a ecranului și a joystick-ului
- 14.05.2024 - realizare schema conexiunilor modulelor din proiect
- 15.05.2024 - realizare schema bloc cu modulele din proiect
- 20.05.2024 - modificarea aranjării elementelor hardware
- 21.05.2024 - inceput etapa software
- 23.05.2024 - gata cu software-ul

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/matei\\_ioan.dudu](http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/matei_ioan.dudu)



Last update: **2024/05/24 20:16**