

# Joc Sudoku

## Autor

## Dumitrescu Georgiana

## Introducere

Proiectul realizat reprezintă o implementare a jocului clasic Sudoku, într-o versiune ce pune la dispoziția jucătorului o grilă de forma 4 x 4.

## Reguli

Jocul poate fi completat de un singur jucător la un moment dat, care va putea selecta pe rând poziția pe care o va scrie, precum și cifra pe care o va scrie pe această poziție. Scopul jocului presupune completarea întregii grile respectând următoarele limitări:

- Pe o linie nu pot exista cifre duplicate
- Pe o coloană nu pot exista cifre duplicate
- Într-un set din cele 4 ale grilei 4 x 4 nu pot exista cifre duplicate

Progresul jucătorului va putea fi observat la orice moment de timp prin intermediul unui ecran LCD.

## Descriere generală

Jucătorul va avea la dispoziție 4 butoane pentru mișcările pe care le va face în cadrul grilei, pentru a ajunge pe poziția pe care dorește să o completeze (sus, jos, stânga, dreapta). Pe lângă cele acestea, va avea un buton ce incrementează o cifră la fiecare apăsare a acestuia, pornind de la 0, până la valoarea maximă 9. Cifra dorită va putea fi plasată pe grilă prin poziționarea pe căsuța corespunzătoare și, în final, apăsarea butonului SELECT. Acest buton va plasa cifra în grilă dacă aceasta reprezintă o variantă corectă de completare. În caz contrar, cifra nu va fi plasată în grilă și

greșeala jucătorului va fi semnalată prin aprinderea unui led roșu și pornirea unui scurt semnal sonor.

## Schema bloc

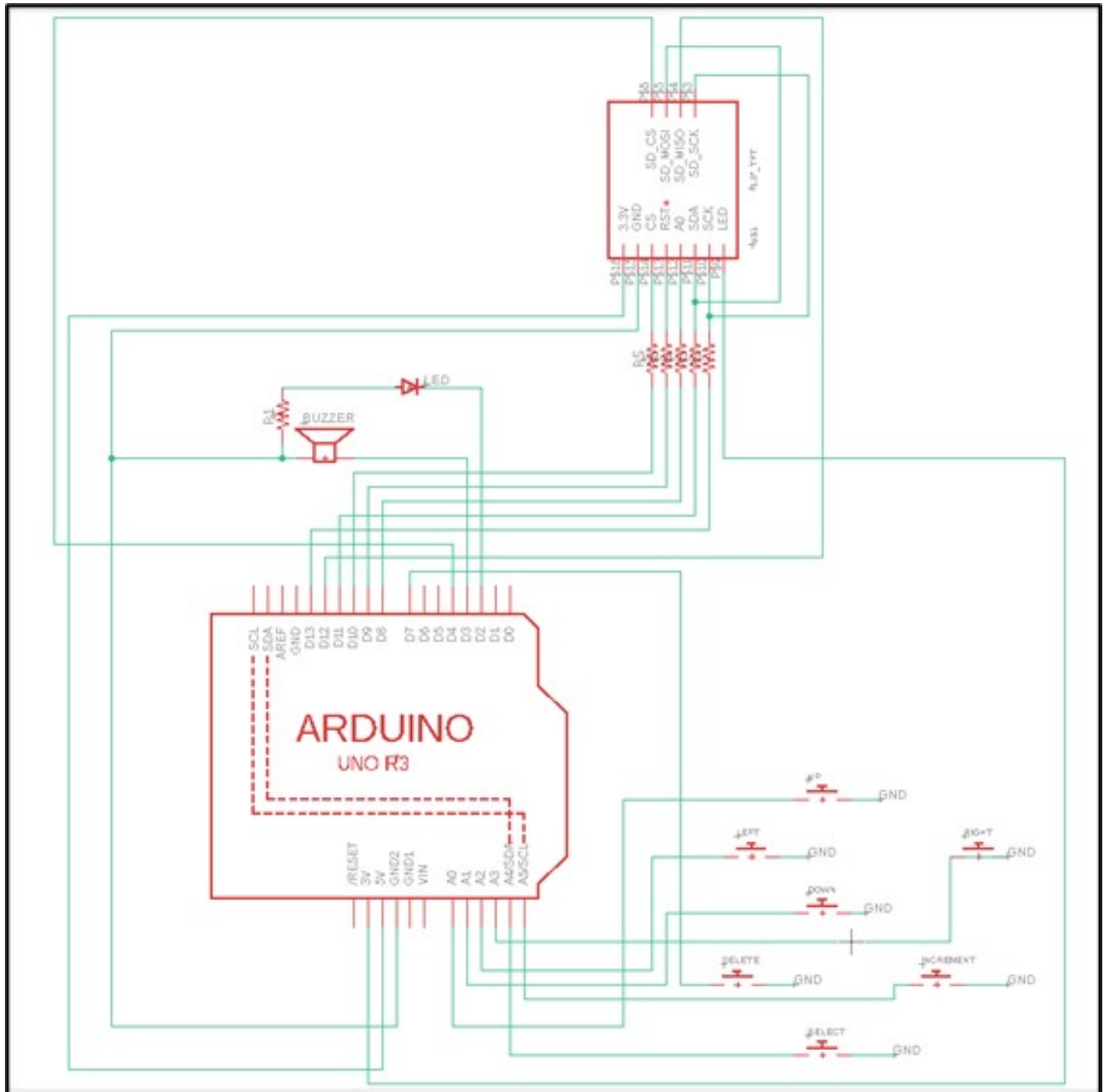


## Hardware Design

### Piese

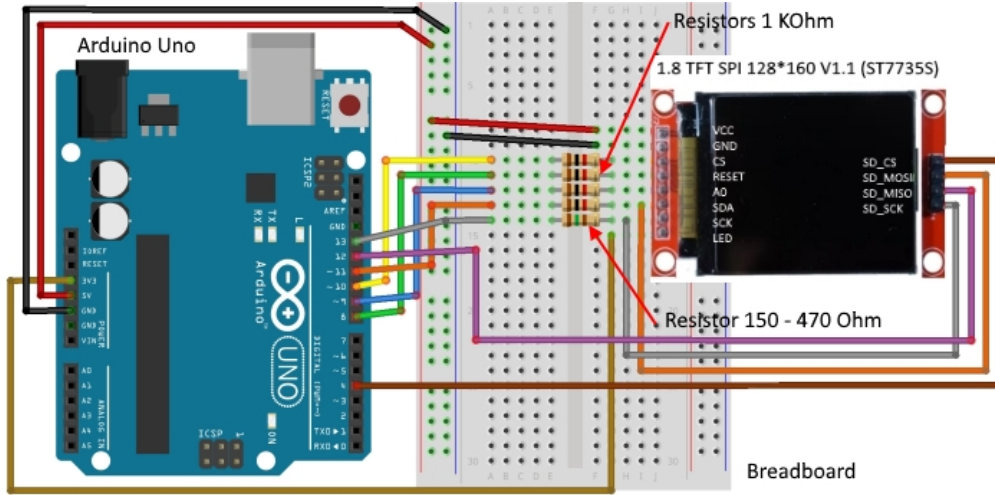
Componenta	Numar	Link
Arduino Uno	1	<a href="#">Arduino</a>
Cablu USB	1	<a href="#">Cablu</a>
Breadboard	1	<a href="#">Breadboard</a>
LED	1	<a href="#">LED</a>
Ecran LCD	1	<a href="#">Ecran LCD</a>
Buzzer	1	<a href="#">Buzzer</a>
Butoane	7	<a href="#">Butoane</a>
Rezistente	5	<a href="#">Rezistente</a>
Fire	30	<a href="#">Fire</a>

## Schema electrica

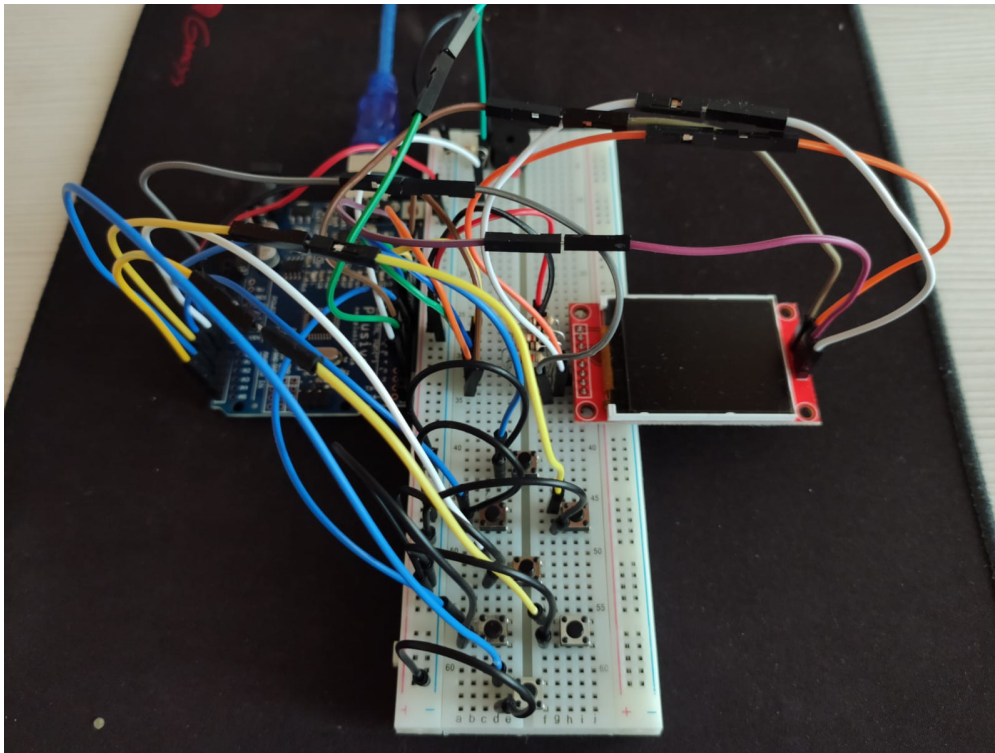


## Conectare ecran LCD

Modulul LCD SPI 128x160 vine cu 8 pini, conectați astfel:



## Conectare hardware completă



## Software Design

## Mediul de dezvoltare

Arduino IDE (1.8.15)

# Biblioteci

[Ucglib](#) → Folosită la afișarea pe ecran LCD

## Descriere implementare

### Funcția `setup()`

- Stabilește rolurile butoanelor, a buzzer-ului și a LED-ului : OUTPUT/INPUT\_PULLUP
- Afișează regulile de acordare a scorului (menținute pe ecran pentru o perioadă scurtă de timp)
- Definește grila de joc
- Scrie numerele presetate în joc (acestea nu pot fi alterate pe perioada jocului)
- Setează valoarea inițială a scorului la 0

Funcția `setup()` generează grila de joc prin apelarea funcției **`resetGame()`**, care:

- Desenează grila de joc
- Printează numerele inițiale
- Inițializează matricea jocului (`sudoku_matrix`), utilizată în verificări ulterioare pentru respectarea regulilor
- Setează poziția inițială a jucătorului în grila de joc

Grila este desenată din 4 linii verticale și 4 linii orizontale de culoare albă, suprapuse unui fundal negru. Numerele presetate în joc sunt scrise cu o culoare roșie pentru a se diferenția de restul și nu pot fi șterse pe parcursul jocului. Deplasarea pe căsuțele grilei este indicată prin colorarea în albastru a celei curente.

**Funcția `loop()`** verifică starea fiecărui buton (LOW = apăsat/ HIGH = liber) și decide pe baza acestuia, acțiunea executată.

- Butonul **UP** - Decrementează poziția pe axa Oy a jucătorului
- Butonul **DOWN** - Incrementează poziția pe axa Oy a jucătorului
- Butonul **LEFT** - Decrementează poziția pe axa Ox a jucătorului
- Butonul **RIGHT** - Incrementează poziția pe axa Ox a jucătorului

Jucătorului nu i se permite ieșirea din grila de joc, în niciuna dintre cele 4 direcții.

- Butonul **INCREMENT** - Incrementează valoarea numărului

Jucătorului nu i se permite incrementarea numărului peste valoarea maximă admisă.

- Butonul **DELETE** - Șterge numărul de pe poziția curentă

Jucătorului nu i se permite ștergerea oricăruia dintre numerele presetate.

- Butonul **SELECT** - Scrie în celula curentă, numărul selectat

Scrierea în celulă este permisă doar prin respectarea regulilor clasice ale jocului:

- Nu este permisă dublarea unei cifre pe același rând
- Nu este permisă dublarea unei cifre pe aceeași coloană
- Cifrele dintr-un set de 4 celule trebuie să fie distincte

#### **Precizări scor:**

- Încercarea de a scrie o cifră ce nu respectă regulile jocului va eșua, iar jucătorul va fi avertizat prin pornirea LED-ului roșu și a alarmei BUZZER-ului, această acțiune rezultând în scăderea a 5 puncte din scor
- Ștergerea unei cifre scrise anterior este penalizată prin scăderea a 2 puncte din scor
- Scrierea unei cifre ce respectă condițiile jocului va adăuga 10 puncte la scor

La fiecare execuție a funcției loop, se verifică dacă jocul a ajuns la final. Dacă această condiție este îndeplinită, pe ecranul LCD se afișează mesajul "FELICITĂRI", urmat de scorul final calculat pe parcursul sesiunii. În urma afișării acestui mesaj, jocul să se reia în formatul inițial prin apelarea funcției **resetGame()**.

## **Rezultate obținute**

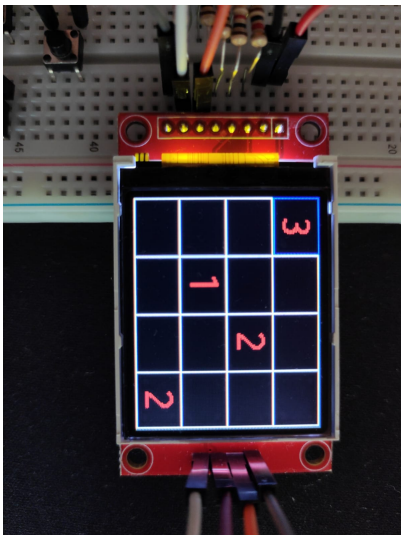
Combinarea design-ului hardware cu implementarea software a rezultat în obținerea unui joc de Sudoku, care respectă design-ul clasic și regulile jocului original. Acesta pune la dispoziția jucătorului toate funcționalitățile necesare (deplasarea în grila de joc, selectarea și scrierea numărului dorit) și, în plus față de jocul clasic, acesta dispune de un semnal de alarmă care se activează la încercarea de încălcare a regulilor. Pe parcursul jocului, se calculează scorul conform regulilor afișate inițial și se afișează la final rezultatul obținut.

[Demo](#)

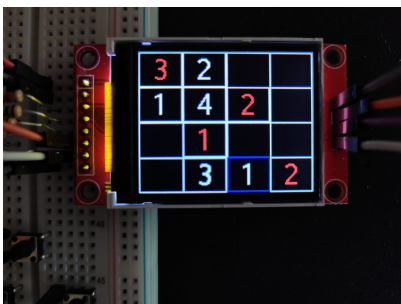
## **Ecran de start**



## Grid inițial



## Ecran joc



# Ecran final



## Concluzii

- Proiectul este realizat integral și implementează toate funcționalitățile necesare
- Implementarea tratează și cazurile excepționale; jucătorul nu poate ajunge într-o stare necunoscută a jocului
- Jocul se aseamănă cu cel clasic și utilizează majoritatea componentelor hardware discutate la laborator
- Pe viitor, acestuia i se pot adăuga funcționalități suplimentare dacă se dorește, codul putând fi actualizat cu ușurință

## Download

[Fișiere sursa & Schema Electrica](#)

## Jurnal

- ✓ 25.aprilie - Alegere temă proiect
- ✓ 26.aprilie - Creare pagină wiki & schema bloc
- ✓ 27.aprilie - 5 mai - Achiziționare piese necesare
- ✓ 6.mai - Conectare ecran LCD
- ✓ 7.mai - 20 mai - Creare & Testare design software

✓ 22.mai

- Adăugare buton DELETE
- Adăugare eveniment la finalul jocului
- Introducerea resetării jocului

✓ 23.mai - 28.mai - Finalizare documentație wiki

✓ 29.mai

- Adăugare scor
- Adăugare ecran inițial pentru stabilire punctaj

## Bibliografie/Resurse

[Conectare ecran LCD](#)

[Biblioteca Ucglib](#)

[Formatare ecran LCD](#)

[Fonturi & Desene ecran LCD](#)

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/joc\\_sudoku](http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/joc_sudoku)



Last update: **2021/05/31 02:02**