

Tablă de șah inteligentă

Conducător lucrare

Conf. Dr. Ing. Horia CUCU

Absolvent

Victor-Valentin MOCANU

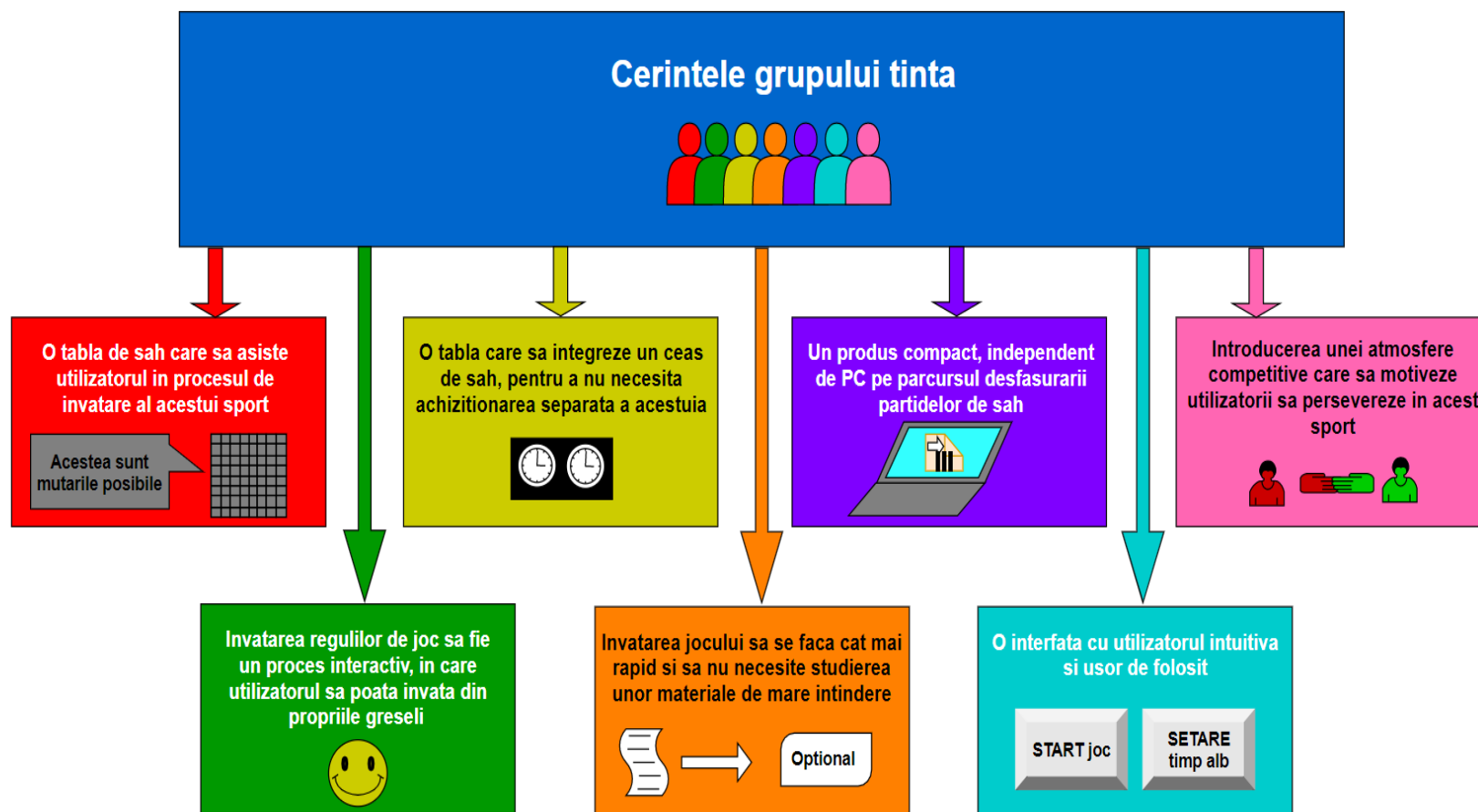
Cuprins

- Obiectivele lucrării
- Cerințele grupului țintă
- Gestionarea funcționalităților parțiale ale tablei de șah inteligente
- Schema bloc a tablei de șah inteligente
- Realizarea fizică a tablei de șah inteligente
- Structura software simplificată
- Organizarea aplicației software
- Abordarea regulilor jocului de șah
- Teste efectuate pe parcursul realizării proiectului
- Contribuții personale
- Concluzii

Obiectivele lucrării

- Şah – joc al minţii, sport;
- Regulile jocului de şah – multe, complexe, necesită timp;
- Jocuri de societate inteligente – şahul, rezolvarea cubului Rubik, jocul de go;
- Tablă de şah inteligentă – sistem incorporat, determină mutarile posibile ale pieselor ridicate şi contorizează timpul de joc.

Cerințele grupului țintă



Gestionarea funcționalităților parțiale ale tablei de șah inteligente

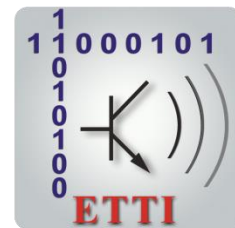


Tabla de sah inteligenta - sistem incorporat

Componente HARDWARE

Element ce determina
modificarea pozitiei
pieselor

Elemente ce permit
utilizatorului sa vizualizeze
campurile posibile de joc ale
pieselor

Elemente ce permit
utilizatorului sa distinga diferite
tipuri de evenimente ce pot
aparea intr-un joc de sah
(roca, transformarea
pionului, amenintarea de mat)

Element ce determina
inceperea si resetarea
partidei de sah

Componente HIBRIDE

Element care verifica respectarea
regulilor impuse

Elemente ce compun
ceasul de sah

Componente SOFTWARE

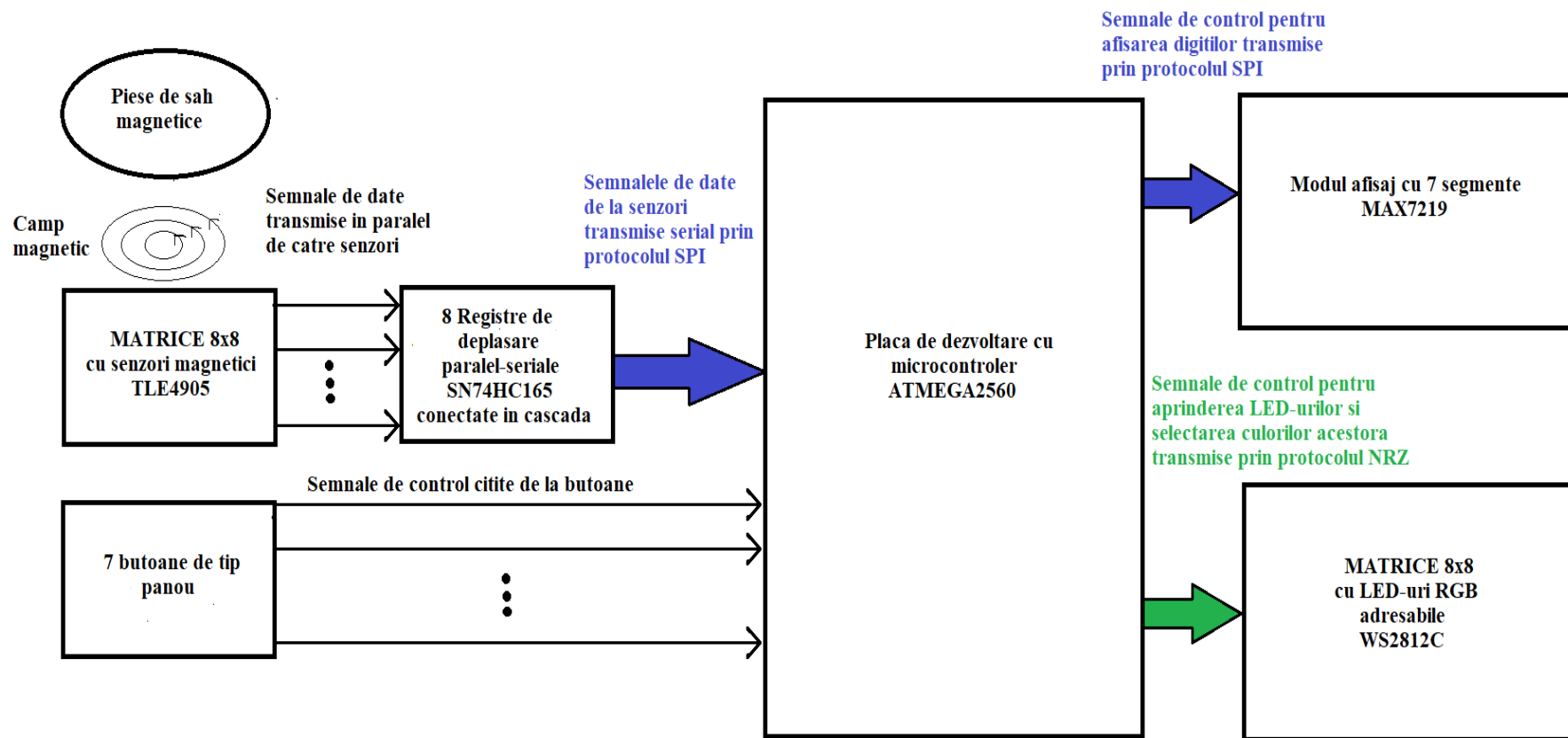
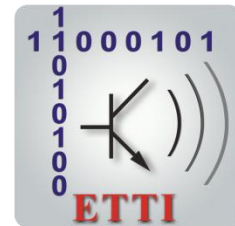
Elemente pentru detectarea tipului si
al culorii piesei de sah

Element ce stocheaza
setul de reguli al jocului
de sah

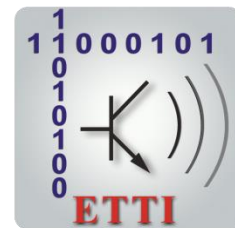
Element pentru sincronizarea
ceasului de sah cu mutarile
efectuate pe tabla

Elemente ce determina finalizarea
unei partide de sah

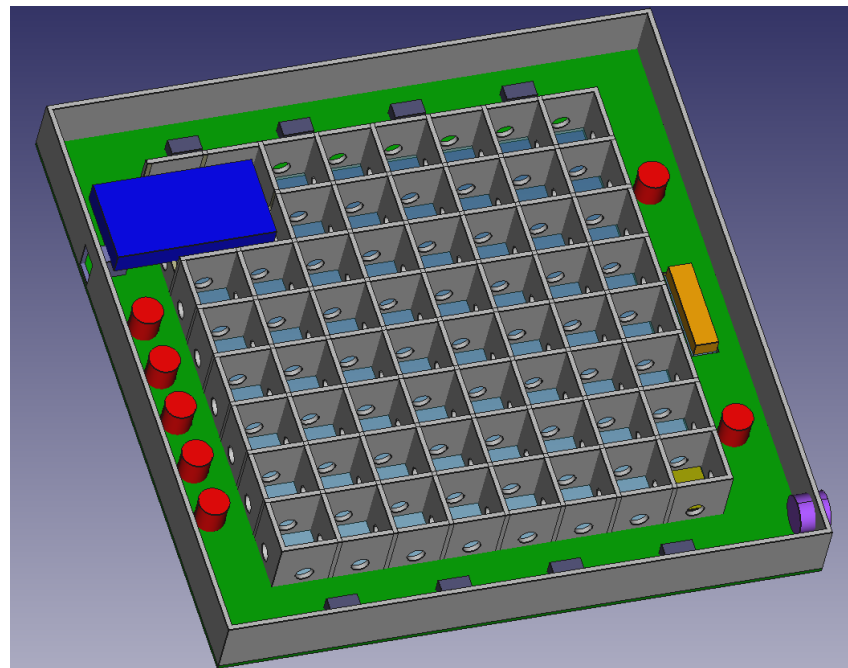
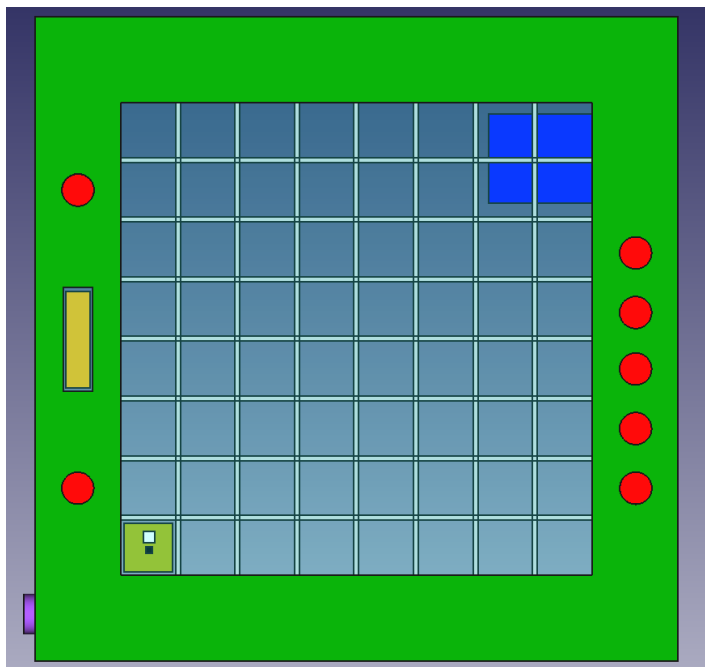
Schema bloc a tablei de șah inteligente



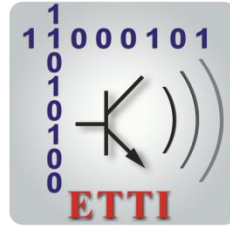
Realizarea fizică a tablei de șah inteligente



Proiectare în mediul FreeCAD



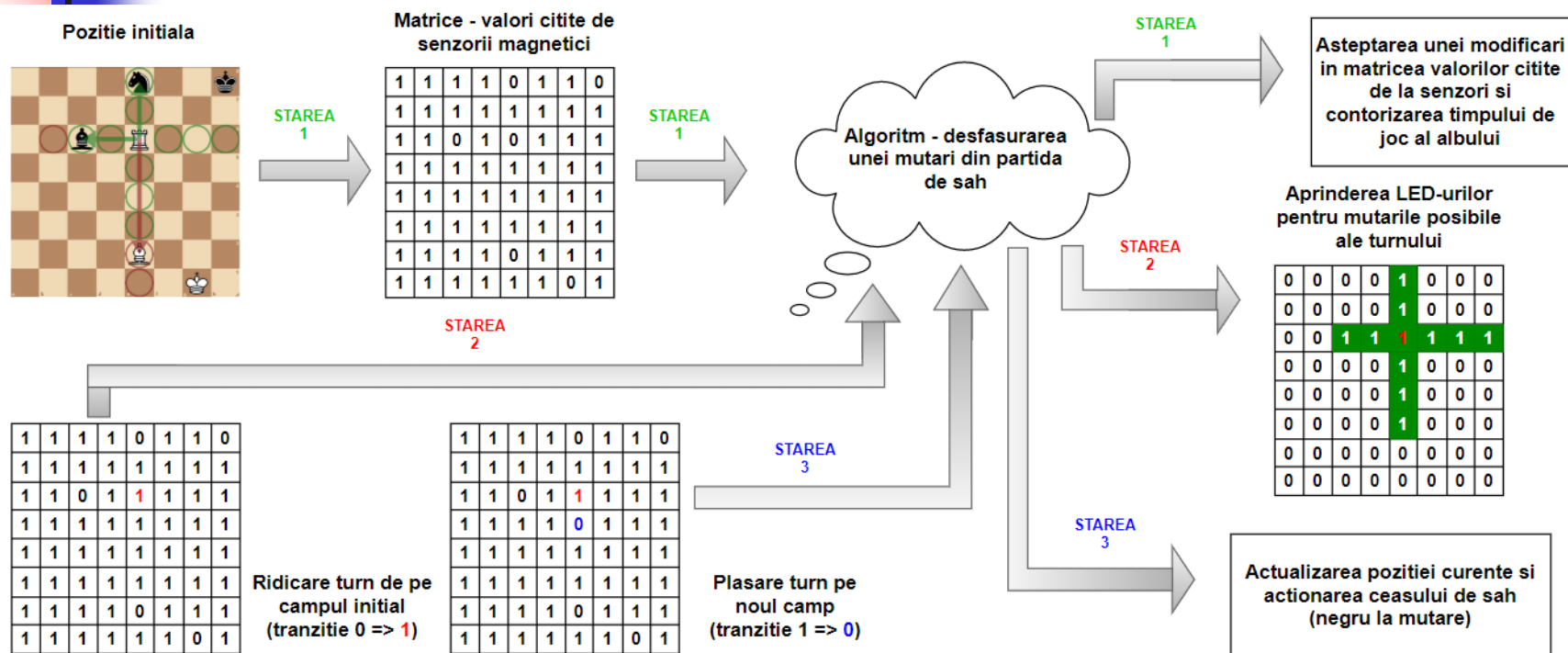
Realizarea fizică a tablei de șah inteligente - continuare



Tablă realizată fizic



Structura software simplificată



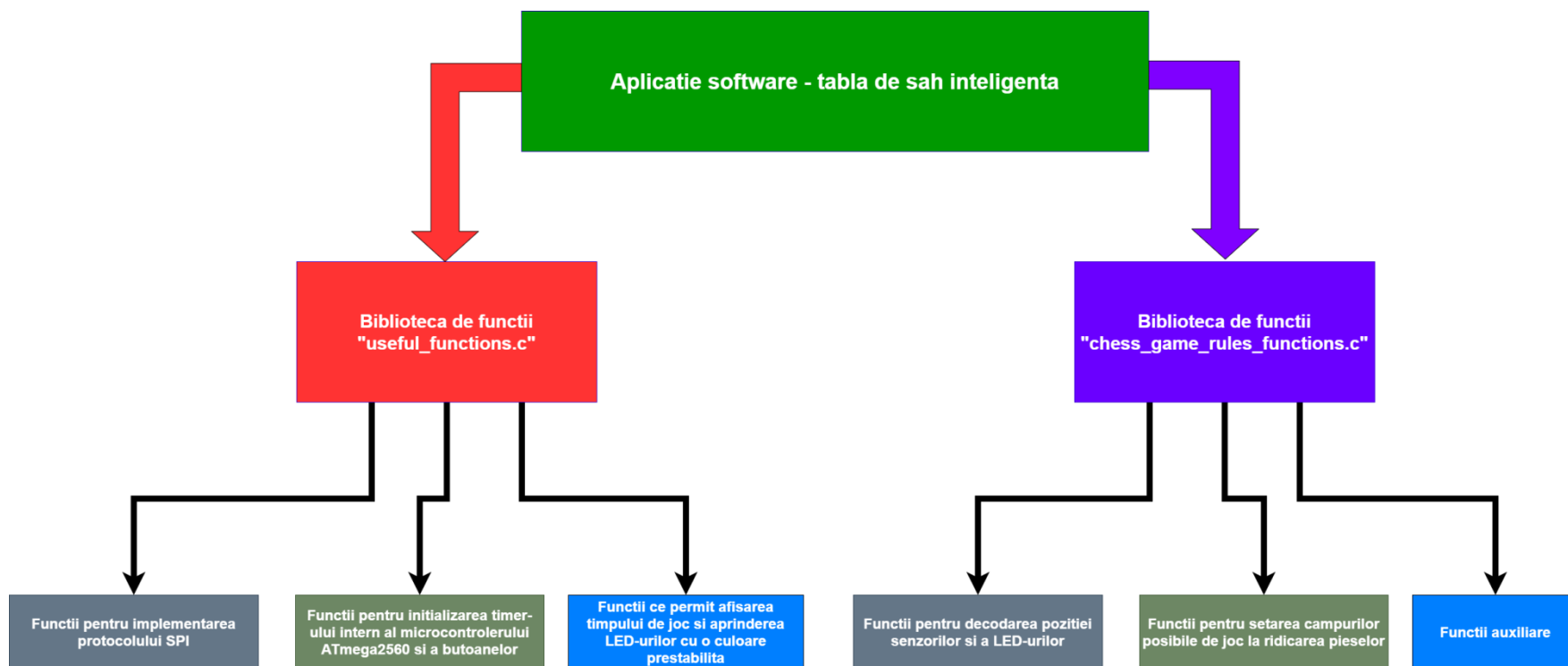
Legenda:

STAREA 1 - Comportamentul tablei de sah in lipsa efectuarii unei mutari

STAREA 2 - Jucatorul aflat la mutare ridica o piesa

STAREA 3 - Piesa ridicata este plasata pe un camp indicat de catre LED-uri

Organizarea aplicației software



**Matricea mutarilor posibile ale
regelui care nu tine cont de
amenintarile de sah**

Matricea mutarilor posibile ale
regelui care nu tine cont de
amenintarile de sah

**Matricea mutarilor posibile ale
turnului**

**Matricea mutarilor posibile ale
regelui care tine cont de
amenintarile de sah**

X

Prelucrare
date

**Matricea mutarilor posibile ale
regelui care tine cont de
amenintarile de sah**

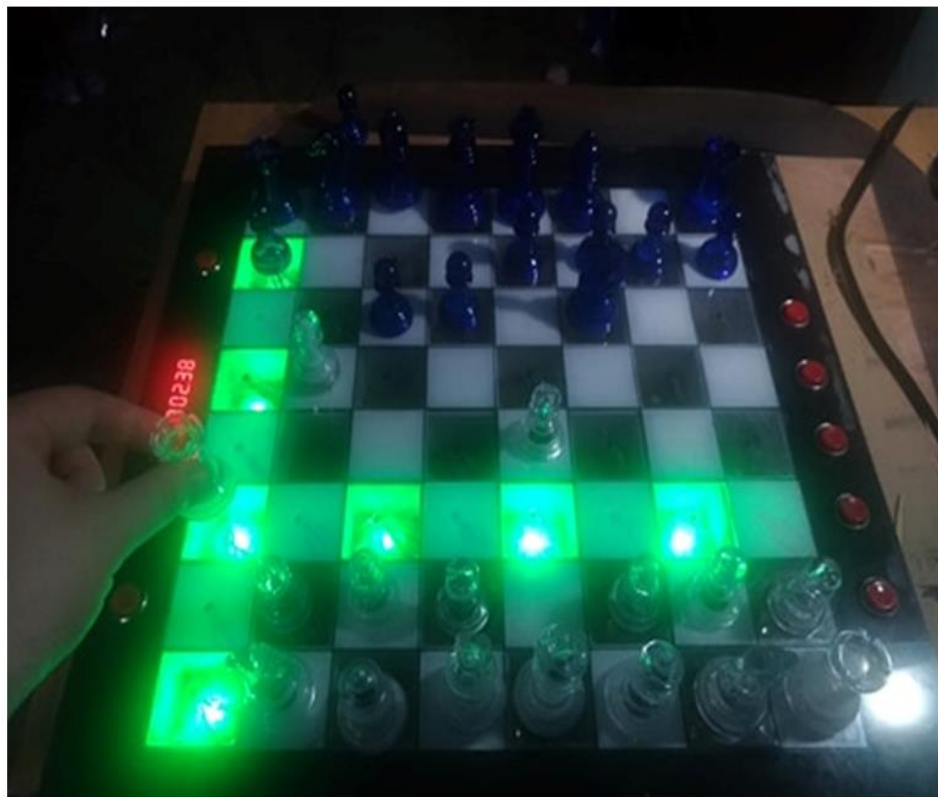
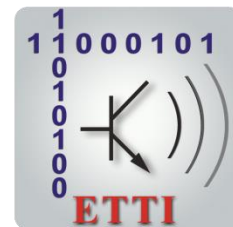
Observatiji:

- Campurile goale ale matricilor corespund valorii "0".
- Rezultatul scaderii 1 - 103 este 154, deoarece toate valorile sunt reprezentate pe 8 biti, fara semn.
- Campurile a caror valori sunt indicate cu culoarea rosie corespund campurilor curente unde sunt pozitionate piesele.
- Toate operatiile aplicate sunt operatii de tip "element cu element".

Pozitia curenta pe tabla de joc



Teste efectuate pe parcursul realizării proiectului



Citirea celor 64 de senzori: sensorValues = 3B1355633F333633

nrOfPiecesHeld = 1

sensorValues = 3B1355633F333633

newLedMatrix values:

0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
103	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

piecesPositionMatrix values:

14	12	13	15	16	13	0	14
11	0	0	0	11	11	11	11
0	0	11	11	0	12	0	0
0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0
0	1	1	1	0	1	1	1
0	2	3	5	6	3	2	4

whoMoves = 1

chessClockPressEnable = 0

whiteCastleInProgress = 0

blackCastleInProgress = 0

first_white_king_move = 1

first_a_column_white_rook_move = 0

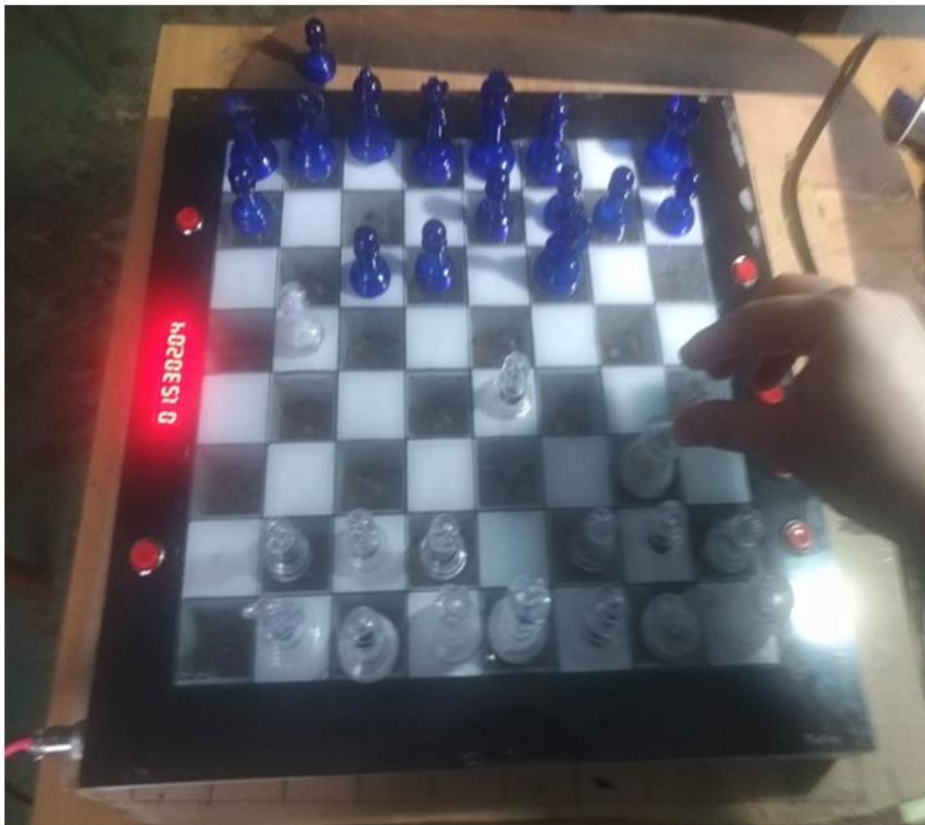
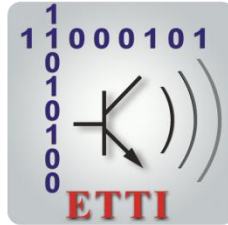
first_h_column_white_rook_move = 1

first_black_king_move = 1

first_a_column_black_rook_move = 1

first_h_column_black_rook_move = 1

Teste efectuate pe parcursul realizării proiectului - continuare



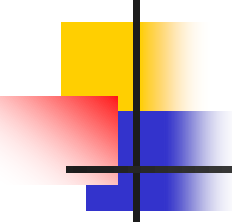
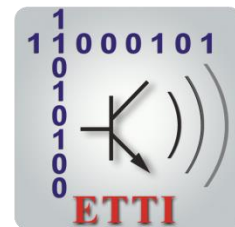
```
Citirea celor 64 de senzori: sensorValues = 3B1355633F333631
nrOfPiecesHeld = 1
it's legal; i = 2, j = 6, currentPieceRow = 2, currentPieceColumn = 0
Black king safety!
sensorValues = 3B1355633F333631
newLedMatrix values:
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
piecesPositionMatrix values:
14 12 13 15 16 13 0 14
11 0 0 0 11 11 11 11
0 0 11 11 0 12 0 0
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 4 0
0 1 1 1 0 1 1 1
0 2 3 5 6 3 2 4
whoMoves = 0
chessClockPressEnable = 1
whiteCastleInProgress = 0
blackCastleInProgress = 0
first_white_king_move = 1
first_a_column_white_rook_move = 0
first_h_column_white_rook_move = 1
first_black_king_move = 1
first_a_column_black_rook_move = 1
first_h_column_black_rook_move = 1
```

Contribuții personale

- Proiectarea și realizarea schemei hardware a sistemului;
- Scrierea bibliotecilor de funcții pentru configurarea modulelor și implementarea regulilor jocului de șah, pornind de la indicațiile prezentate în foaia de catalog a microcontrolerului ATmega2560 și folosind o bibliotecă existentă pentru controlul LED-urilor WS2812C;
- Proiectarea și realizarea structurii fizice care încapsulează circuitul electronic al tablei de șah;
- Efectuarea testelor asupra montajului hardware (cu ajutorul multimetrului și al simulatorului Proteus) și introducerea unei metode folosite pentru depanarea codului (prezentată în diapozitivele anterioare).

Concluzii

- Toate funcționalitățile tablei de șah sunt realizate în timp real;
- Structura de tip fagure – nu permite pătrunderea radiației optice decât pe câmpul destinat;
- Depanarea codului – facilitată prin reprezentarea informațiilor într-o manieră similară cu cea dorită a fi obținută pe tabla fizic realizată.



Vă mulțumesc!!!