

# Ioana NEAMȚU (67014) - TicTacToe Ultimate

Autorul poate fi contactat la adresa: **Login pentru adresa**

## Introducere



Ultimate TicTacToe este un joc inspirat din clasicul X si 0 (TicTacToe) insa jucat pe o matrice de 9x9. Jocul are 9 cadrane, dupa cum se poate vedea si in poza de mai sus.

In alegerea proiectului, am pornit de la ideea ca mi-ar placea sa fac un joc pe LED-uri si am incercat sa complic X si 0-ul clasic, ajungand la Ultimate Tic Tac Toe, scopul fiind de relaxare si distractie.

## Reguli

- Pentru a castiga, un jucator trebuie sa castige 3 cadrane pe linie, coloana sau diagonala
- Pentru fiecare cadran sunt aplicate regulile de la X si 0 clasic
- Fiecare casuta din interiorul unui cadran corespunde cu un cadran si la fiecare mutare, adversarul este obligat sa joace in cadranul corespunzator casutei in care a fost facuta mutarea
- In cazul in care unul din jucatori trebuie sa mute intr-un cadran deja inchis, sunt deblocate toate cadranele in care inca se poate juca si poate alege in care sa mute.
- (Exemplu: Cadranul din mijloc este deja inchis si Player1 joaca, punand X in casuta din mijloc a unui cadran. Player2 poate alege orice cadran deschis pentru mutarea sa)

## Functionalitate dorita

- La final, proiectul ar trebui sa ofere o interfata prietenoasa pentru un joc multiplayer
- In momentul in care un utilizator castiga, intreaga matrice se va colora in culoarea corespunzatoare

Cea mai mare problema pe care am intalnit-o in proiectarea aplicatiei, a fost numarul relativ mare de pini de care am nevoie pentru a comanda cele 81 de LED-uri care reprezinta playground-ul, motiv pentru care am ales sa folosesc registrele de shift (comandate prin 3 pini, vor putea comanda 8 linii).

## Descriere generală



**Buttons** - este o matrice de 9 butoane folosita pentru a selecta o noua mutare (casuta unde player-ul vrea sa joace) in cadrul unui cadran deja ales sau a selecta cadranul in care player-ul vrea sa joace, in cazul in care exista mai multe optiuni. Se va indica folosind blocul *Indicator* care din cele 2 mutari vor

fi facute.

**Game Selector** - este o matrice de 3×3 LED-uri care va indica in ce cadrane player-ul curent poate juca. In cazul in care mai multe LED-uri sunt aprinse, deci mai multe cadrane disponibile, player-ul mai intai va selecta cadranul, folosind *buttons*

**9×9 LED Matrix** - este o matrice de 81 de LED-uri bicolore care va reprezenta tabla de joc. Fiecare player va avea asociata o culoare.

**Indicator** - este reprezentat printr-un LED bicolor (sau 2 LED-uri normale) care va comunica daca trebuie aleasa o mutare sau un cadran.

## Hardware Design

Lisa de piese necesare:

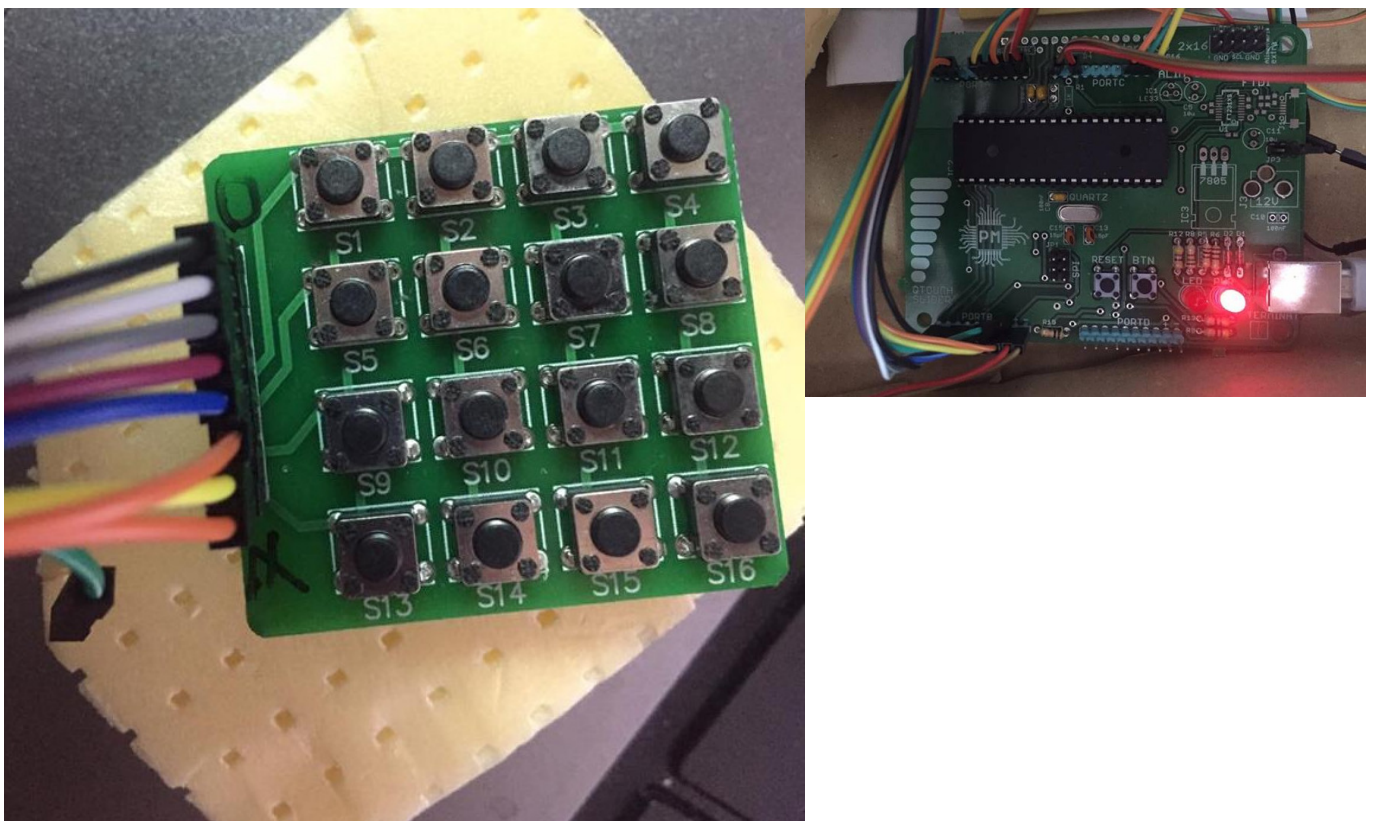
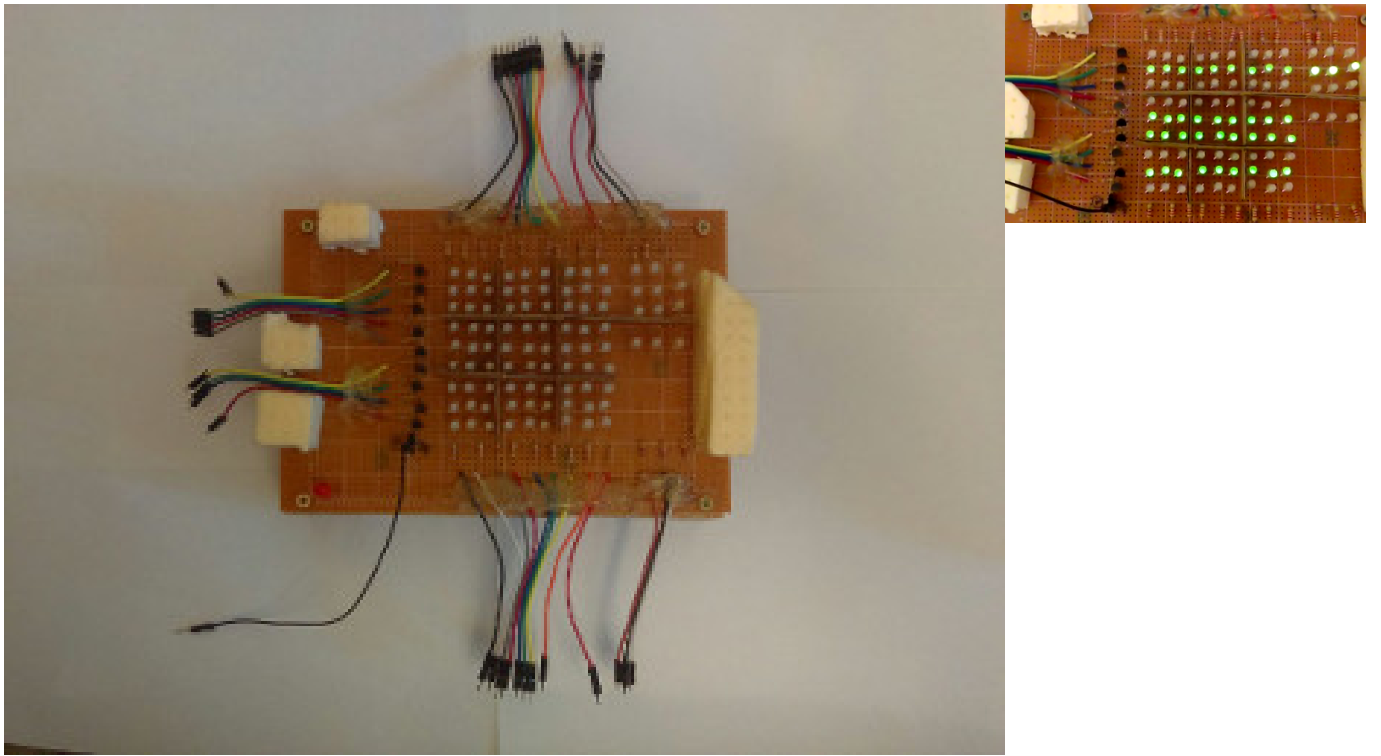
- placa de baza PM
- 91 LED-uri bicolore cu catod comun
- 9 tranzistori NPN2N2222A
- matrice 4×4 butoate
- 6 shift registers 74HC595
- 9 rezistente 1k8 (pe baza tranzistorului)
- 18 rezistentr 2k2 (pe fiecare coloana)
- fire

Schema electrica



Comandarea coloanelor se face similar cu cea a liniilor, folosind 2 shift register pe culoare.

### Matricea de LED-uri



## Software Design

Pentru dezvoltarea software-ului am folosit

- BootloaderHID
- avr-gcc
- Sublime

Pentru inceput, am implementat comandarea matricii de led-uri folosind registrele de shift:

- Matricea se aprinde prin balearea liniilor aprinse: se seteaza valorile de pe linia 0 si se aprinde, aplicand o tensiune HIGH(+5V) pe baza tranzistorului corespunzator liniei 0
- Se stringe linia si se continua aprinderea celorlalte linii
- Am adaugat delay-uri astfel incat LED-urile stinse sa nu lumineze si cele aprinse sa nu clipeasca

La fiecare front pozitiv pe intrearea CLOCK, datele sunt deplasate spre bitii mai semnificativi, iar valoarea de la intrare este pusa pe bitul 0. Pe frontul pozitiv al LATCH-ului, informatia este copiata in zona de memorare si de aici ajunge pe iesiri.

In implementarea mea, semnalul de CLOCK si LATCH este comun pentru grupuri de 2 circuite (linii, coloane rosii, coloane verzi). Pentru linii am o valoare de 1 care se deplaseaza pe bitii corespunzatori fiecărei linii, activand astfel linia. Pe coloane, valorile sunt transmise serializat.

Am continuat cu citirea inputului de la tastatura. Pentru identificarea butonului apăsat se procedeaza astfel:

- La initializare, toti pinii portului la care e conectata tastatura sunt considerati intrari (stare de inalta impedanta) si pentru pinii corespunzatori randurilor, se activeaza rezistenta de pull-up
- Pentru a indentifica butonul apasat, pinii corespunzatori coloanelor se trec pe rand in starea LOW (se transforma in iesiri) si se verifica daca pe vreo linie am 0 → buton apasat
- Returnez butonul apasat sub forma unui octet, cu primii 4 biti setati la valoarea liniei, si ultimii la cea a coloanei
- Daca nu s-a apasat nicio tasta, se returneaza 0xFF

Pasul final a constat in implementarea logicii jocului care functioneaza ca un automat cu 4 stari:

1. Choose Board - jucatorul trebuie sa aleaga cadranul in care sa joace, in limita cadranelor libere (aprinse in boardul din dreapta)
2. Choose Move - jucatorul trebuie sa aleaga miscarea in cadranul indicat de led-ul aprins in cadranul auxiliar din dreapta
3. Game Done - jocul curent a fost castigat board-ul se coloreaza in culoarea castigatorului. Apasarea butonului 13 va declansa inceperea unui joc nou
4. Set Done - un jucator a castigat setul curent. Pe matricea de led-uri se va afisa litera W in culoarea corespunzatoare invingatorului. Butonul 14 duce la resetarea scorului si a intregului set

Am implementat si o stare de eroare, in cazul unei mutari invalide: tabela de scor se coloreaza in rosu. Jucatorul trebuie sa refaca mutarea.

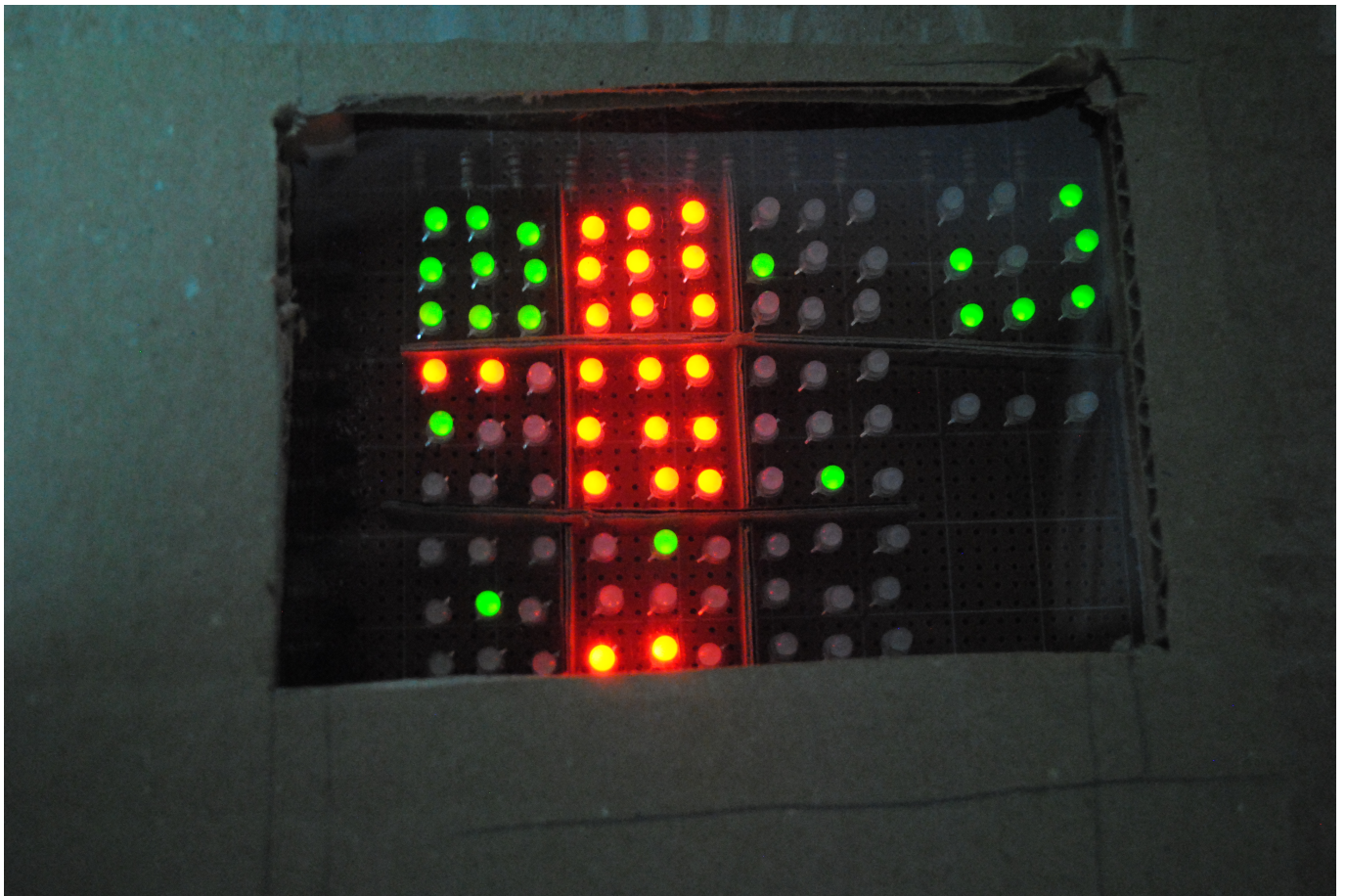
## Rezultate Obținute

Am terminat proiectul, implementand jocul Ultimate Tic Tac Toe pentru 2 jucatori, invingatorul urmand sa fie desemnat dupa castigarea a 2 meciuri din 3. Acest lucru poate fi observat dupa tabela de scor, dar si dupa W(inner)-ul afisat pe matricea de LED-uri in culoarea corespunzatoare player-ului

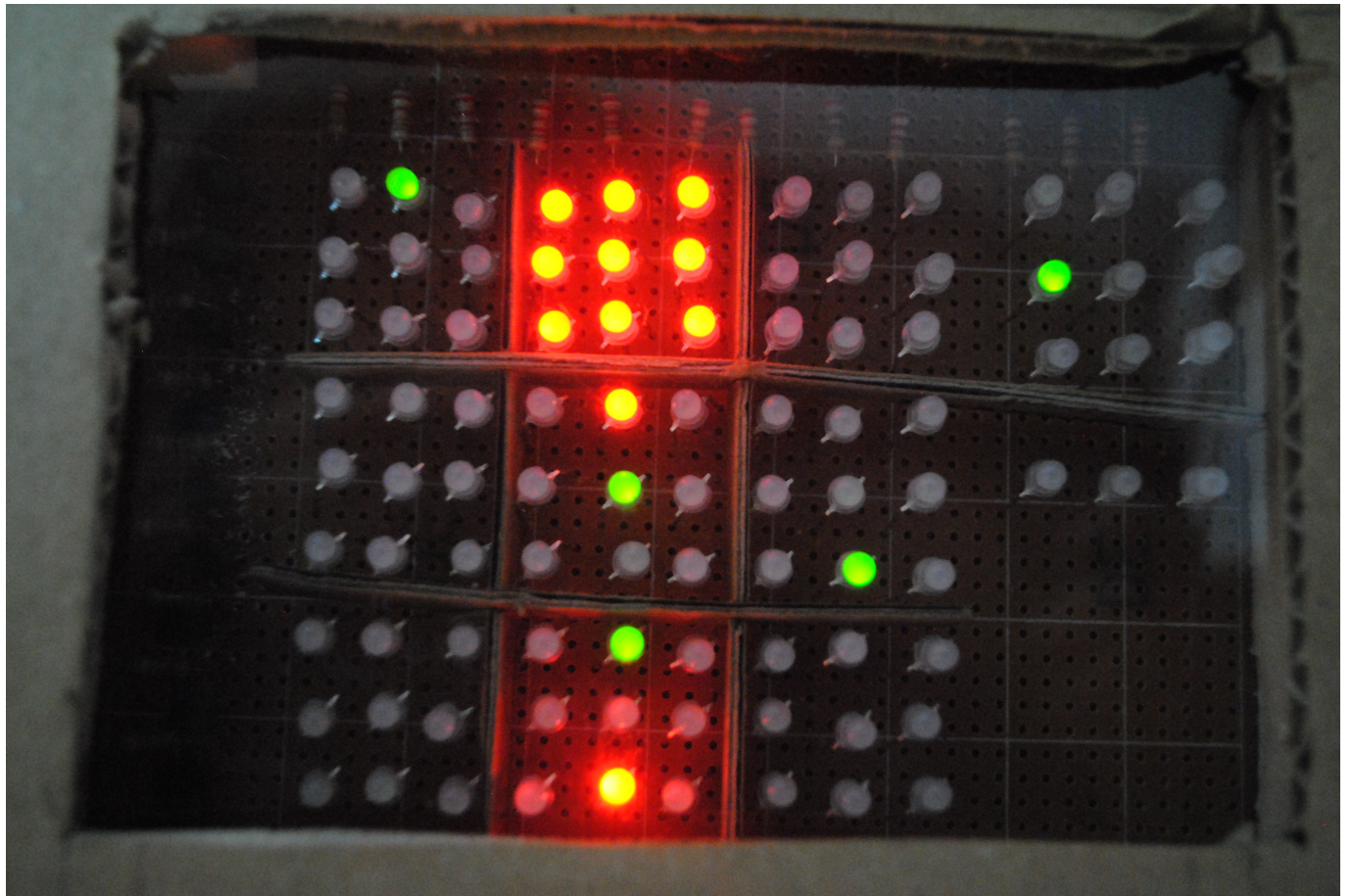
care a castigat.

Mai jos, puteti vedea cateva imagini cu proiectul:

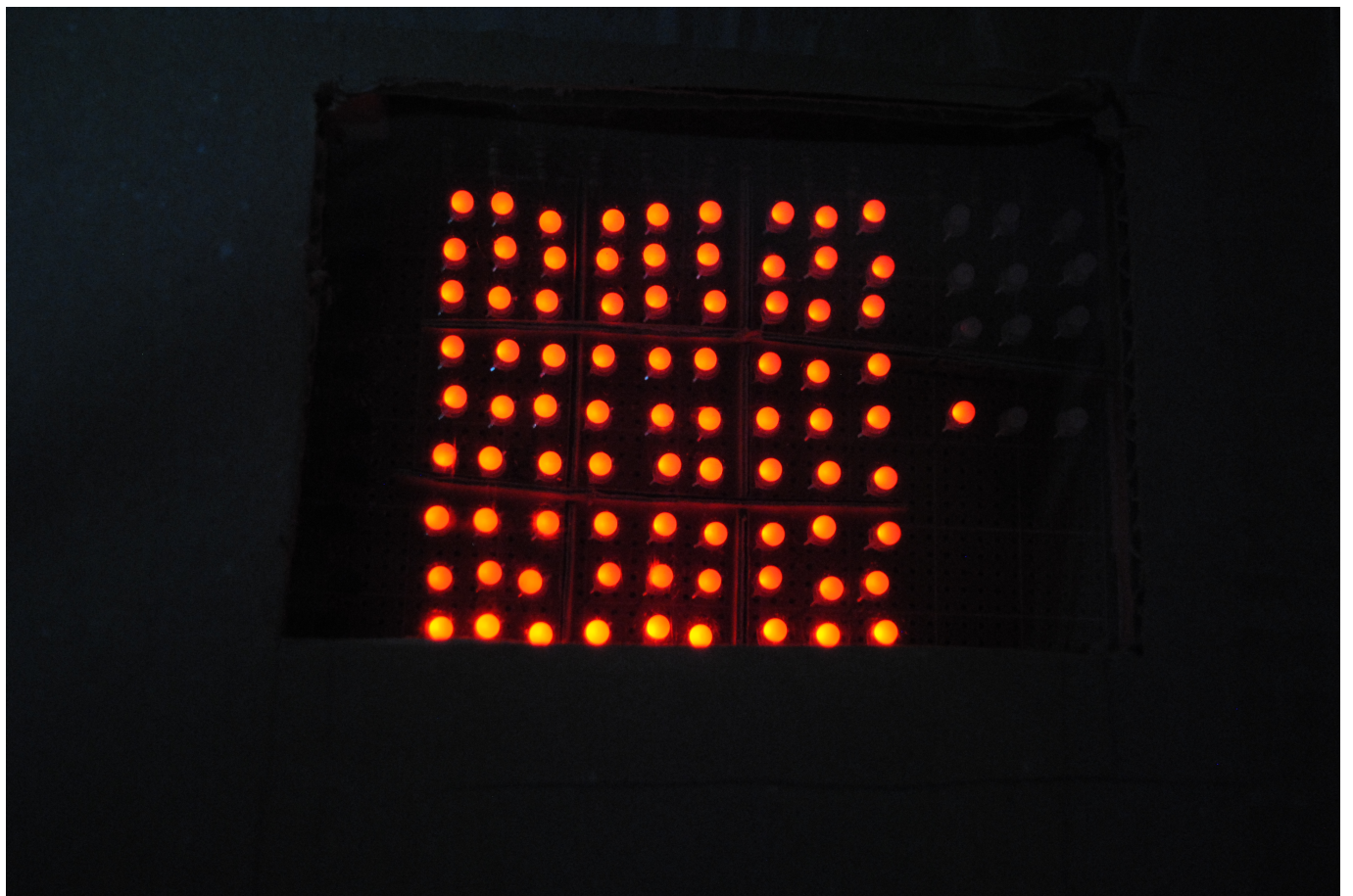
- **CHOOSE BOARD STATE**



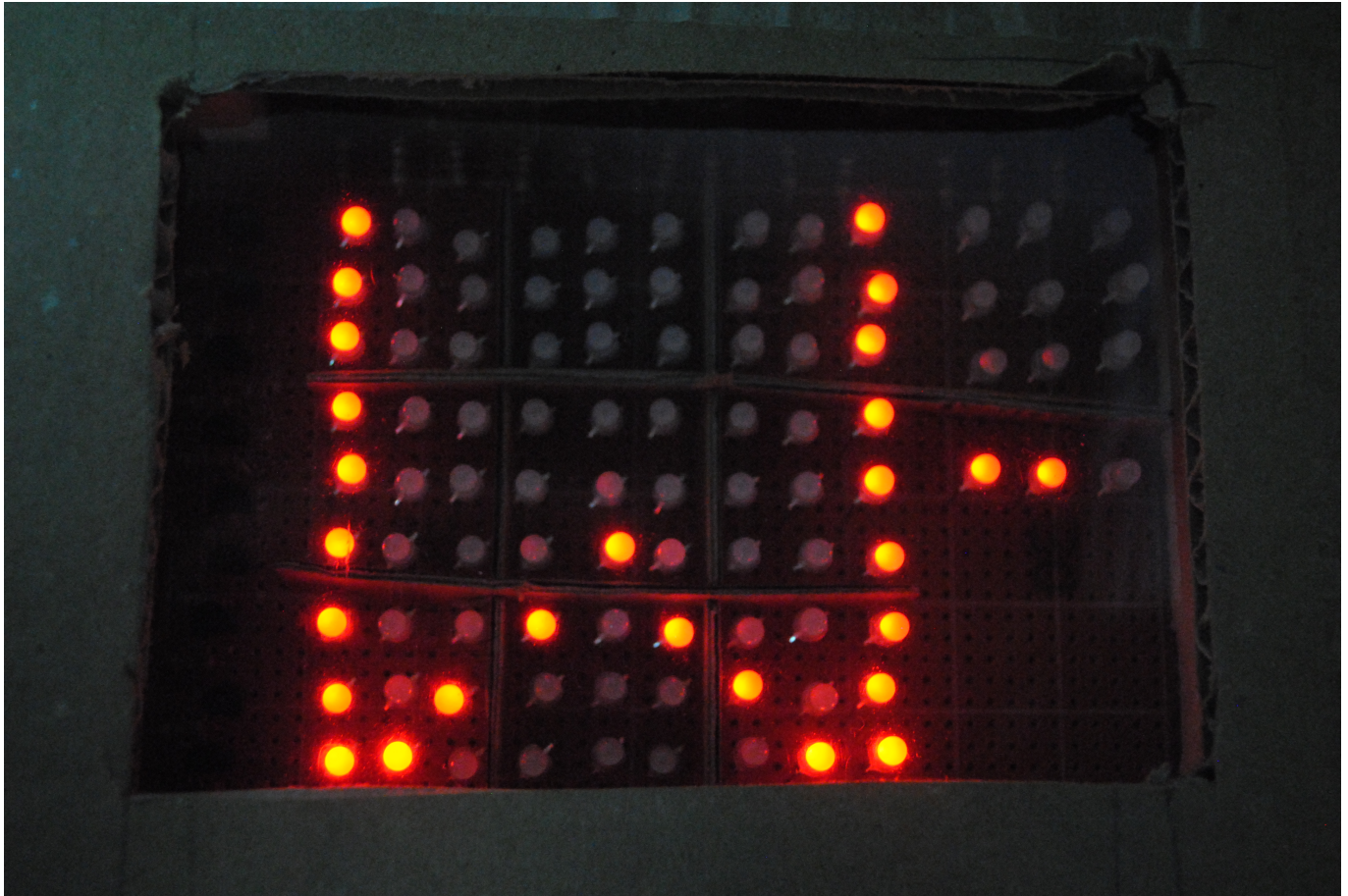
- **CHOOSE MOVE STATE**



• RED won the game



• RED won the set



Video: [watch](#)

## Concluzii

A fost placut si interesant sa lucrez la acest proiect, mai ales la partea hardware, cu toate ca am avut foarte mult de lipit. Cea mai mare provocare a fost sa lucrez cu shift registers, deoarece nu mai avusem contact cu aceste circuite pana acum.

## Download

Fisiere .sch [ioana\\_neamtu\\_sch.zip](#)

Arhiva cod [ioana\\_neamtu\\_sourcecod.zip](#)

## Jurnal

- Saptamana 9: Alegerea temei de proiect
- Saptamana 10-11: Lipirea pieselor pe placuta de test si realizarea schemei bloc a proiectului
- Saptamana 12-13: Lipirea placii de test si scrierea unor functii auxiliare pentru registrele de shift

- Saptamana 14: Implementare software si testare

## Bibliografie/Resurse

DataSheet ATmega324 [atmega.pdf](#)

DataSheet 74HC595 [shiftreg.pdf](#)

DataSheet 2N2222A [2n2222.pdf](#)

DataSheet LED [led.pdf](#)

Link-uri utile:

<https://learn.adafruit.com/adafruit-arduino-lesson-4-eight-leds/the-74hc595-shift-register>

<https://www.arduino.cc/en/Tutorial/ShiftOut>

<http://extremeelectronics.co.in/avr-tutorials/4x3-matrix-keypad-interface-avr-tutorial/>

- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/imatesica/tictactoeultimate>



Last update: **2021/04/14 15:07**