­­

**Tic Tac Toe**

**Arduino Project**

**Specializare: Electronică,** **telecomunicații și tehnologii informaționale**

**Grupa: 4LF631**

**Nume: Bucura Teodora Elena**

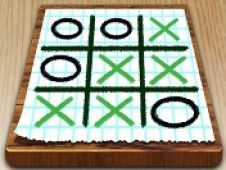
**Cuprins**

1. **Ideea proiectului…………………………………………….….… 3**
2. Ce este Tic Tac Toe…………………………………………………………………… 3
3. Motivarea alegerii temei……………………………………………………………. 4
4. **Componente………………………………………….……………. 5**
5. **Cum functionează?.................................................... 9**
6. Componenta Hardware…………………………………………………………… 9
7. Componenta Software……………………………………….…………….…….. 12
8. Diagramă conectare.…………………………………………………………….… 18
9. **Instrucțiuni de utilizare………………………………………… 23**

Ideea Proiectului

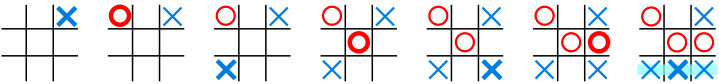
Ce este Tic Tac Toe?

**Tic Tac Toe** ( sau X și 0 ) este un joc clasic de două persoane, care se joacă pe o grilă de 3x3. Scopul jocului este să formezi o linie continuă cu trei simboluri identice — fie un „X”, fie un „O” — pe orizontală, verticală sau diagonală. Fiecare jucător pune pe rând unul dintre simboluri într-o celulă goală a grilei, iar jocul se termină atunci când unul dintre jucători formează o linie de trei simboluri sau când grila este completă, caz în care jocul se poate considera remiză (dacă niciunul dintre jucători nu a câștigat).



Mulți cred că X și 0 este un joc simplu. Dar dacă te apleci asupra istoriei sale, poți afla multe lucruri interesante. Studiul originilor acestui joc evidențiază locul său special în lumea jocurilor.

În Egiptul Antic au fost găsite urme ale unui joc similar, gravate pe acoperișurile templelor. Numele de atunci, "Seega", folosea pietre ca piese, ceea ce poate sugera o semnificație spirituală a jocului. Egiptenii antici au pus bazele pentru jocul X și 0, inventând prima sa versiune. Romanii aveau și ei propria lor versiune a jocului, numit "Terni Lapilli". În locul X și 0 familiare nouă, au folosit simboluri speciale, gravate pe monumentele publice. Și în India Antică exista o variantă a jocului, numit "Pada", unde sarcina era de a crea un anumit desen folosind scoici sau pietricele mici.



Motivarea alegerii temei

Am ales să realizez proiectul „Tic Tac Toe” utilizând Arduino deoarece îmi doream să combin pasiunea pentru tehnologie cu plăcerea de a învăța lucruri noi, într-un mod interactiv și practic. De la început, am simțit că acest proiect îmi va oferi o oportunitate deosebită de a lucra cu componente hardware și de a înțelege mai bine cum funcționează diferitele elemente electronice. Arduino, fiind o platformă accesibilă și ușor de utilizat, a fost alegerea perfectă pentru a implementa acest joc clasic într-o formă digitală.

Unul dintre cele mai interesante aspecte ale acestui proiect a fost integrarea diferitelor componente hardware: butoane, LED-uri și un ecran LCD. Fiecare componentă a avut rolul său esențial în realizarea unui joc funcțional și intuitiv. Folosind Arduino, am reușit să construiesc un sistem interactiv în care fiecare apăsare de buton are un impact direct asupra jocului, iar LED-urile oferă un feedback vizual imediat. Interacțiunea cu aceste componente hardware a fost o experiență captivantă, care mi-a permis să aplic concepte teoretice în mod practic.

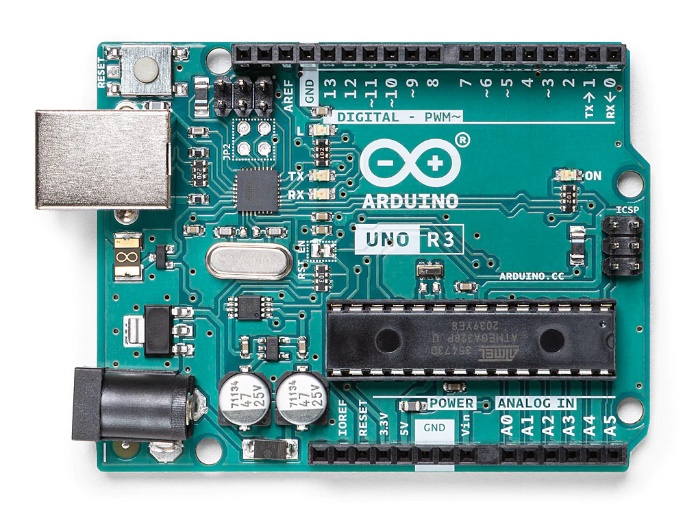
Mai mult, munca cu Arduino mi-a consolidat abilitățile de programare și mi-a arătat cât de importantă este logica în realizarea unui joc de acest tip. Am învățat să scriu cod eficient pentru a gestiona secvențele de joc, a verifica câștigătorul și a răspunde corect la fiecare acțiune a jucătorilor.

Componente

Lista componente utilizate:

* 1x Arduino Uno R3

Este o placă de dezvoltare cu un microcontroler ATmega328P. Este folosită pentru a controla toate celelalte componente ale proiectului, procesând semnalele și executând codul programat pentru a controla jocul Tic Tac Toe.



* 9x LED RGB de 5mm 4 pini

Aceste LED-uri pot emite lumini de mai multe culori (roșu, verde, albastru) și sunt folosite pentru a indica starea fiecărei celule din joc (X, O sau liberă). Fiecare LED are 4 pini pentru a controla culorile.



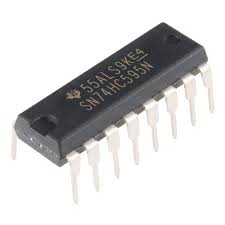
* 20x Rezistente 220 ohmi, si cate o bucata de 100, 150, 200, 220, 300, 470, 560,680,820 ohmi
* 9x Butoane20x

Fiecare buton corespunde unei celule din grila Tic Tac Toe și permite jucătorilor să selecteze locul unde vor pune X sau O. Butoanele sunt folosite pentru a trimite semnale către Arduino.



* 3x Registru de deplasare SN74HC595

Aceste componente permit extinderea numărului de pini de ieșire ai Arduino-ului. Ele sunt folosite pentru a controla mai multe LED-uri cu ajutorul unui număr redus de pini de pe Arduino.



* 1x Ecran LCD 1602 cu modul I2C IIC

Acest ecran este folosit pentru a afișa informații relevante, cum ar fi mesajele de stare (de exemplu, „Jucătorul X a câștigat” sau „Remiză”), într-un format ușor de citit. Modul I2C reduce numărul de pini necesari pentru conectare.

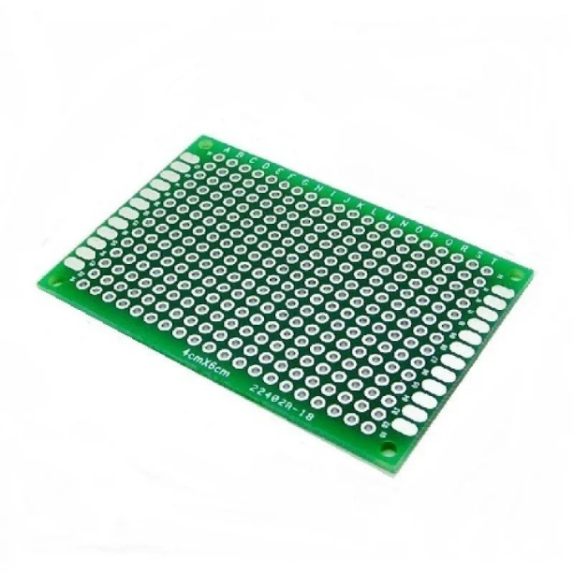


* 1x Placa PCB prototipare ( o față, 5x7cm )

Este o placă pe care sunt montate și conectate componentele electronice. Este utilizată pentru prototipurile electronice, oferind o bază pentru a construi circuite temporare.

* 2x Placa PCB prototipare ( doua fețe, 7x9cm )

Aceste plăci sunt mai mari și permit montarea mai multor componente sau un circuit mai complex. Au linii de conexiune pe ambele fețe pentru o conectivitate mai ușoară.



* 2x Bara 40 pini mama
* 1x Bara 40 pini tata
* 1x Bara 40 pini tata la 90 grade

Bara de pini este folosită pentru a conecta firele Dupont la diferite componente, oferind o soluție ușor de utilizat pentru realizarea circuitelor și pentru a conecta cele 2 plăci de prototipare între ele.



* 10x Fire dupont tata-mama

Aceste fire sunt folosite pentru a face conexiuni între Arduino și diverse componente. Conectorii „tata” și „mama” permit realizarea unui circuit rapid și ușor, fără lipire.

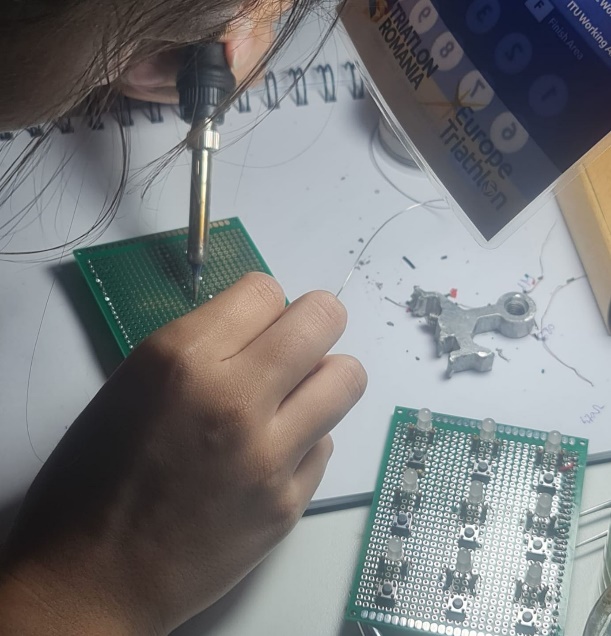
* 1x Cablu conductor electric diverse culori

Cablurile acestea sunt folosite pentru a realiza conexiuni între pinii Arduino și componentele electronice, fiind utili pentru identificarea ușoară a firelor în funcție de culoare.

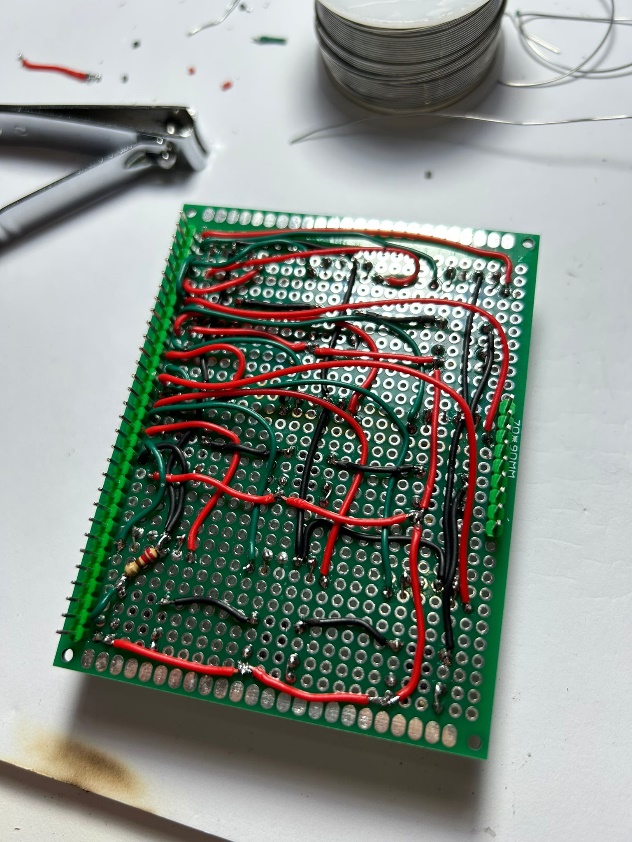
Cum funcționează?

**Componenta Hardware**

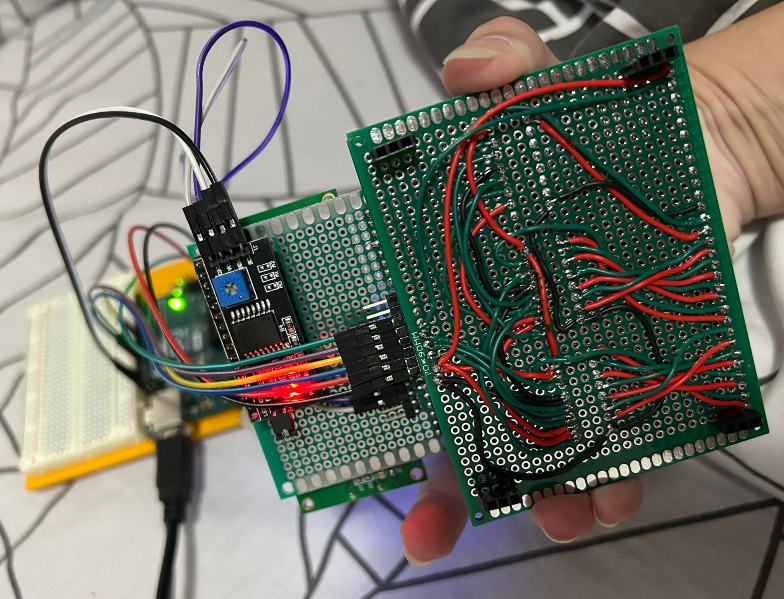
* Pasul 1: Creearea schemelor electrice ( pe site-ul [www.tinkercad.com](http://www.tinkercad.com)).
* Pasul 2: Montarea LED-urilor, butoanelor și a rezistențelor pe primul PCB.



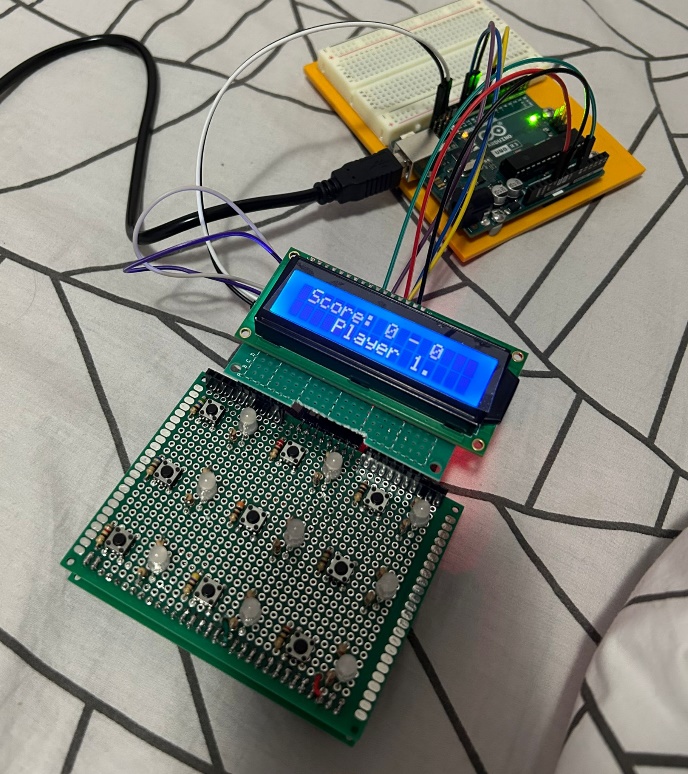
* Pasul 3: Montarea șirurilor de pini tată pe lateralele PCB-ului și lipirea firelor conform schemelor.



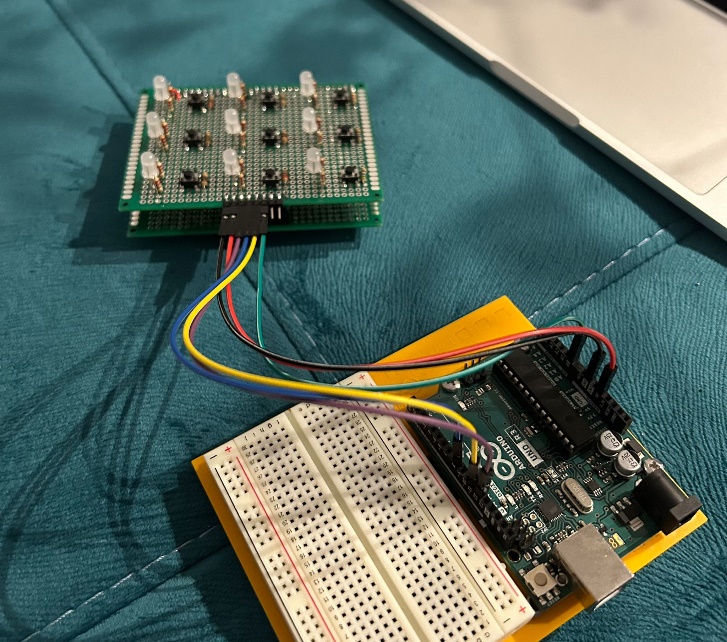
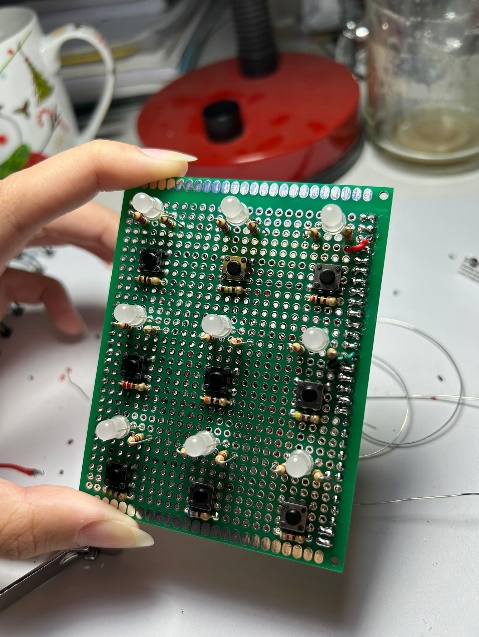
* Pasul 4: Montarea registriilor de deplasare, a piniilor tată-mama pe al doilea PCB și lipirea firelor conform schemelor.



* Pasul 5: Montarea ecranului LCD și a modulului pe al treilea PCB.



* Pasul 6: Verificare cu multimetrul dacă se aprind toate Led-urile, continuitate trasee.



**Componenta Software**

* Pasul 1: Instalare Arduino IDE și conectarea plăcuței la laptop.
* Pasul 2: Descarcarea bibliotecii “LiquidCrystal\_I2C” si rularea unui cod pentru a verifica ce adresă I2C folosește display-ul LCD.

// Biblioteci necesare pentru funcționarea ecranului LCD

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

// Inițializarea ecranului LCD la adresa I2C specifică (0x27) și cu dimensiunea 16x2

LiquidCrystal\_I2C lcd(0x27,16,2);

* Pasul 3: Creearea variabilelor globale, și a functiei “void setup”.

void setup() {

// Configurarea pinilor ca OUTPUT

pinMode(latchPin, OUTPUT);

pinMode(dataPin, OUTPUT);

pinMode(clockPin, OUTPUT);

// Pornirea comunicației seriale

Serial.begin(9600);

// Inițializarea ecranului LCD și afișarea mesajului de start

lcd.init();

lcd.backlight();

lcd.setCursor(2,0);

lcd.print("Tic Tac Toe");

lcd.setCursor(4,1);

lcd.print("Let's play!");

delay(3000);

}

* Pasul 4: Creearea funcției ”void loop” si celorlalte funcții ajutătoare pentru buna funcționare a jocului.

void loop() {

// Afișăm scorul curent pe ecranul LCD

lcd.setCursor(1,0);

lcd.print("Score: ");

lcd.setCursor(8,0);

lcd.print(scor1);

lcd.print(" - ");

lcd.print(scor2);

// Citim butonul apăsat și pornim jocul

button = getButtonPush();

playTicTacToe(); // Funcția principală a jocului

refreshDisplay(); // Actualizăm LED-urile

delay(50); // Pauză pentru stabilitate

}

Funcția “playTicTacToe” este cea care rulează partida de X și O , care verifică după fiecare mutare dacă cineva a câștigat, calculează și afișează câștigătorul, numără mutăriile și partidele.

Funcția getButtonPush este esențială, deoarece la apăsarea unui buton, cu ajutorul pinului analogic și a rezistențelor diferite puse în circuit aflăm o valoare specifică. În această funcție există 9 condiții “if” care returnează o valoare de la 1 la 9 pentru fiecare apăsare.

// Funcție pentru actualizarea stării LED-urilor

void refreshDisplay() {

// Resetăm stările byte-urilor folosite pentru shift register

byte1 = 0; byte2 = 0; byte3 = 0;

// Setăm LED-urile în funcție de starea lor: roșu (1) sau albastru (2)

if (lights[1] == 2){byte1 = byte1 ^ B00000001;}

if (lights[1] == 1){byte1 = byte1 ^ B00000010;}

if (lights[2] == 2){byte1 = byte1 ^ B00000100;}

if (lights[2] == 1){byte1 = byte1 ^ B00001000;}

if (lights[3] == 2){byte1 = byte1 ^ B00010000;}

if (lights[3] == 1){byte1 = byte1 ^ B00100000;}

if (lights[4] == 2){byte1 = byte1 ^ B01000000;}

if (lights[4] == 1){byte1 = byte1 ^ B10000000;}

if (lights[5] == 2){byte2 = byte2 ^ B00000001;}

if (lights[5] == 1){byte2 = byte2 ^ B00000010;}

if (lights[6] == 2){byte2 = byte2 ^ B00000100;}

if (lights[6] == 1){byte2 = byte2 ^ B00001000;}

if (lights[7] == 2){byte2 = byte2 ^ B00010000;}

if (lights[7] == 1){byte2 = byte2 ^ B00100000;}

if (lights[8] == 2){byte2 = byte2 ^ B01000000;}

if (lights[8] == 1){byte2 = byte2 ^ B10000000;}

if (lights[9] == 2){byte3 = byte3 ^ B00000001;}

if (lights[9] == 1){byte3 = byte3 ^ B00000010;}

setLEDs(); // Apelăm funcția pentru a trimite datele către LED-uri

}

Am declarant global 2 constate “player1=1” și “player2=2” pentru a știi cine face mutarea respectivă. În funcția “refreshDisplay” avem 18 condiții “if”, 9 aprind becul culoarea roșie, iar celelalte albastru, folosiind cei 3 regiștrii de deplasare denumiți “byte1”, “byte2” și “byte3”.

// Funcție pentru a trimite datele către registrul de deplasare

void setLEDs() {

digitalWrite(latchPin, LOW); // Coborâm pinul latch

shiftOut(dataPin, clockPin, MSBFIRST, byte3); // Transmitem byte3

shiftOut(dataPin, clockPin, MSBFIRST, byte2); // Transmitem byte2

shiftOut(dataPin, clockPin, MSBFIRST, byte1); // Transmitem byte1

digitalWrite(latchPin, HIGH); // Ridicăm pinul latch pentru a actualiza LED-urile

}

Iar la final această funcție stinge toate LED-urile de pe tablă pentru a putea incepe o nouă partidă.

// Funcție care curăță tabla de joc (resetează LED-urile la starea inițială)

void clearBoard() {

for (int i = 1; i < 10; i++) {

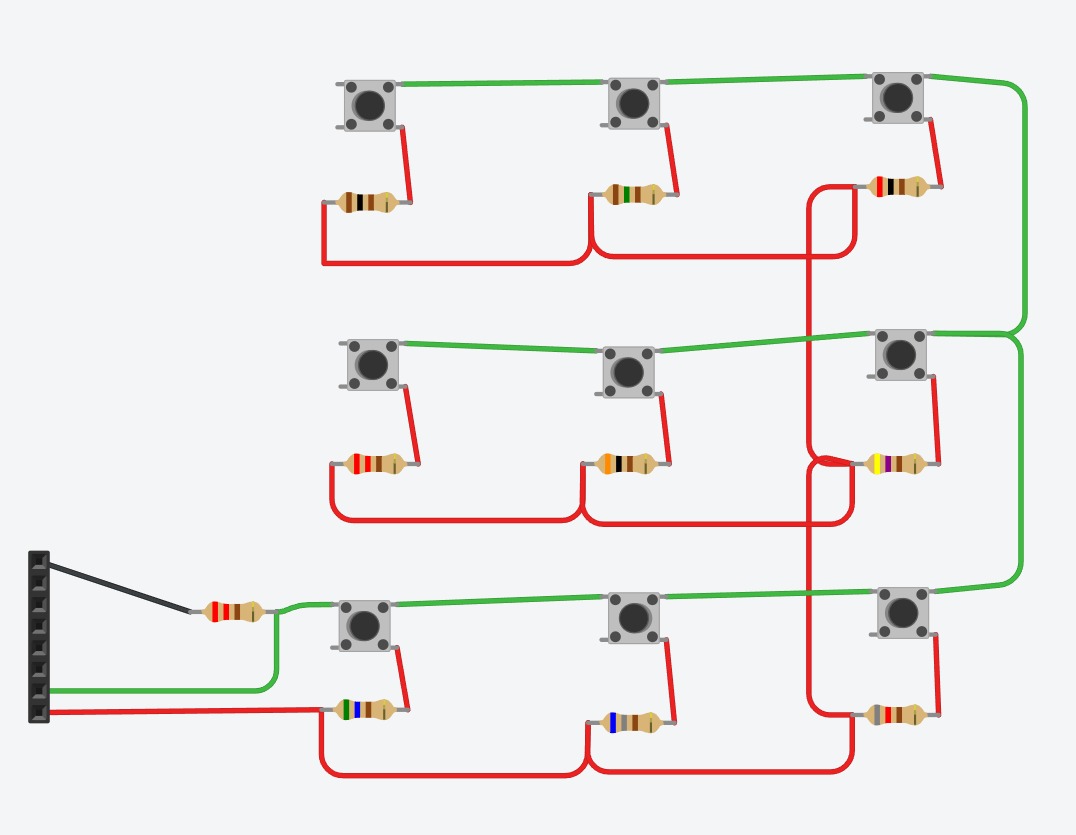
lights[i] = 0; // Resetăm fiecare LED la 0 (stins)

}

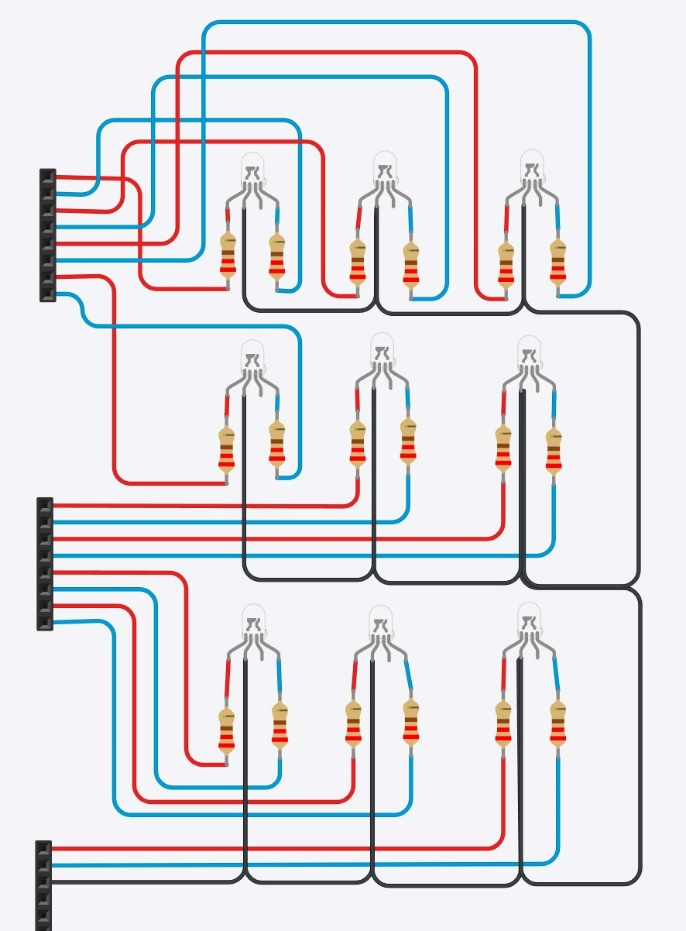
**Diagramă conectare**

Pe primul PCB am legat butoanele din schema 1 și LED-urile din schema 2 în felul următor:

* Ground-ul din ambele scheme (firul negru sau minusul) merg la pinul aflat in dreptul liniei numărul 19 a PCB-ului ( pin19 ).
* VCC-ul din schema 1 (firul roșu sau plusul) merge la pin31, iar A0 (firul verde) merge la pin30. Din acest fir verde pleacă GND-ul, cu o rezistență între ele.
* Firele roșii din schema 2 merg la pinii impari de la 1 la 18, iar cele verzi merg la pinii pari, reprezentând pinul 1 al LED-ului (culoarea roșie) trasat cu roșu si pinul 4 al LED-ului (culoarea albastra) trasat cu albastru.

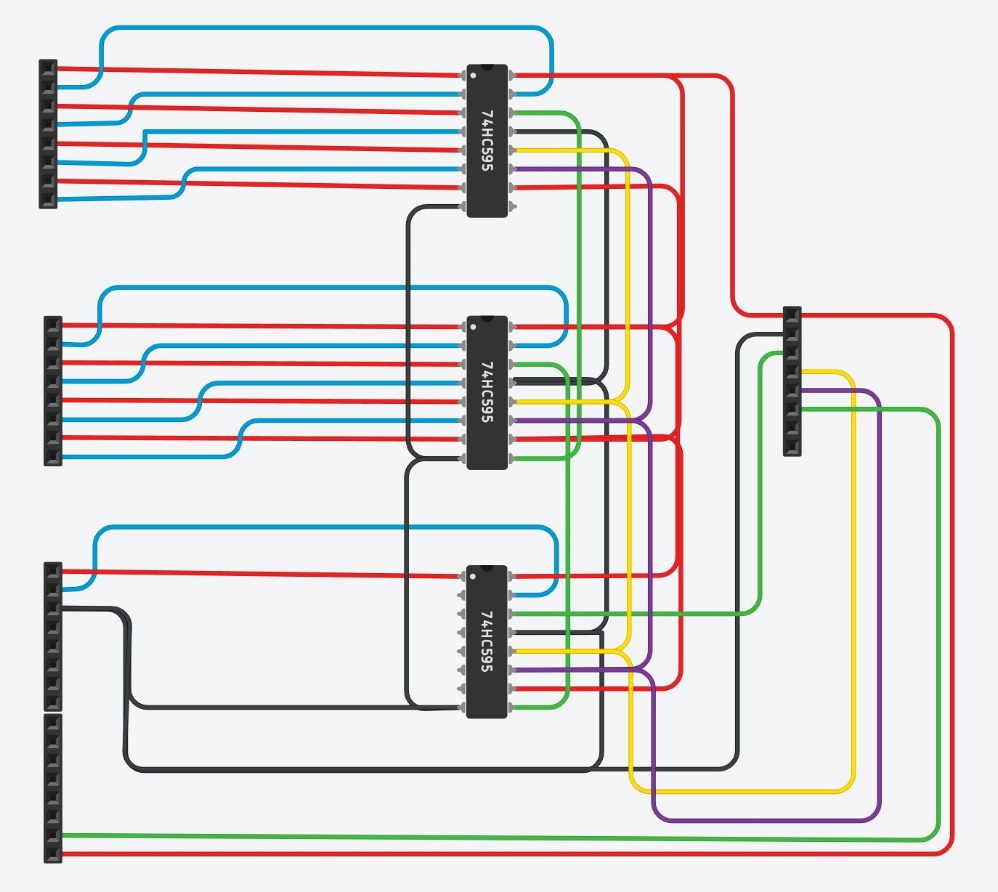


Schema 1



Schema 2

Pe al 2-lea PCB am legat cei 3 regiștrii de deplasare din schema 3 și am dus toate firele într-o bară de pini finală ce duce spre Arduino.

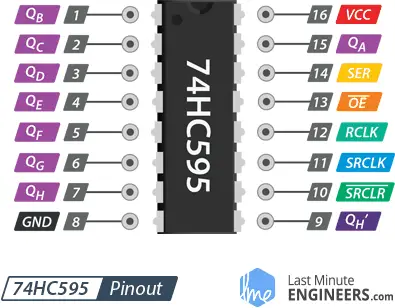


Schema 3

Firele de la 1 la 18 duc spre pinii Q0-Q7 ai registriilor, GND-ul de pe pin19 este legat de pinii de GND ai regiștriilor, iar mai apoi de al doilea pin din dreapta.

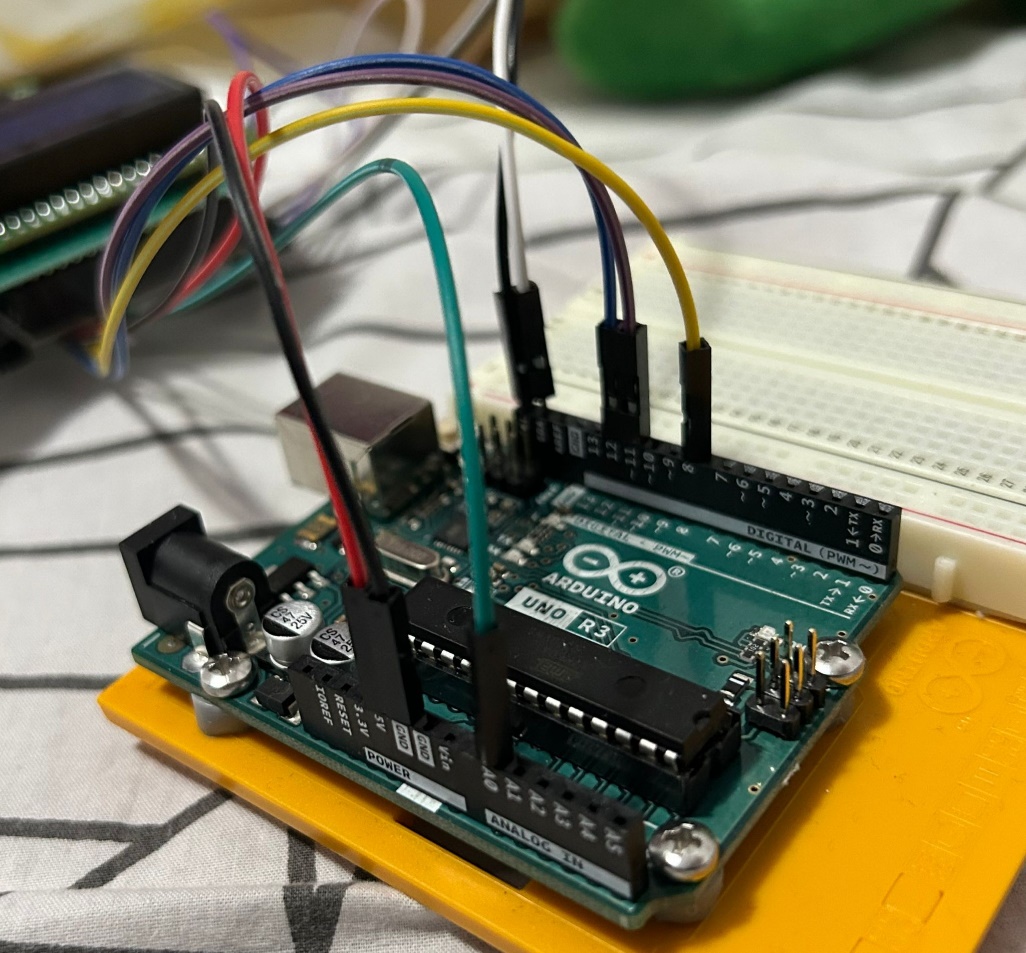
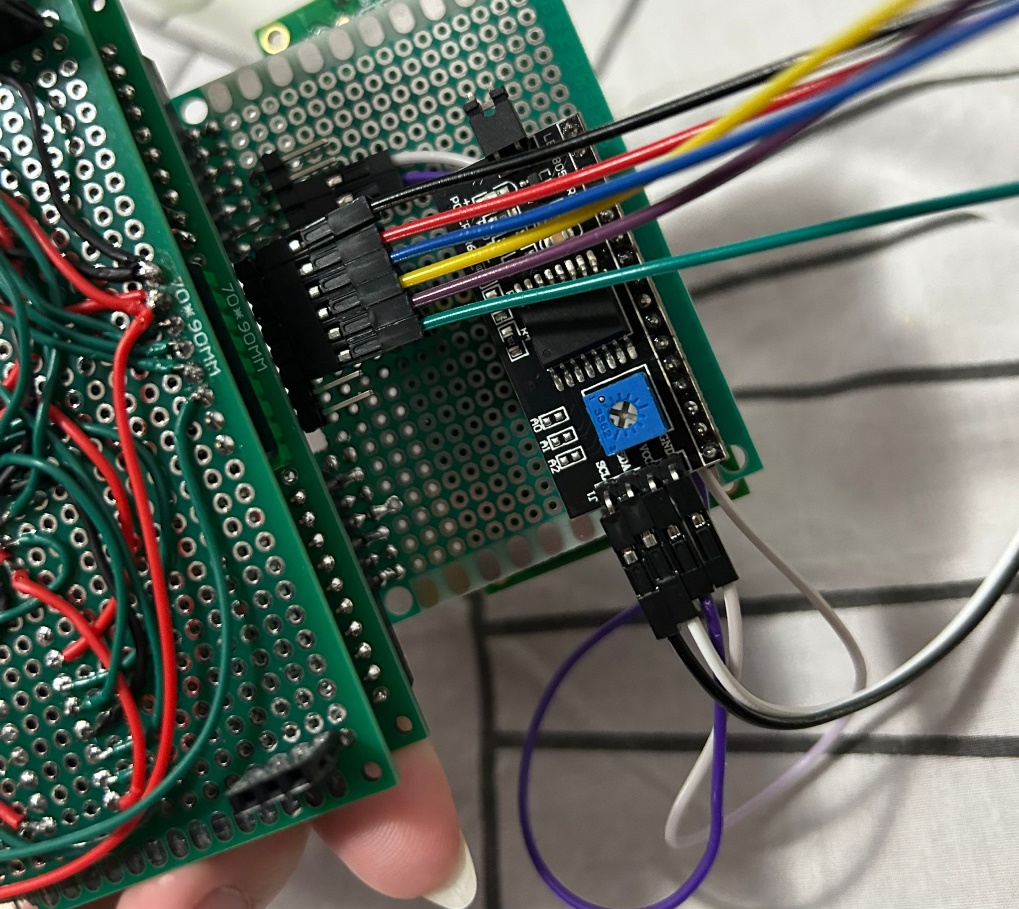
Firul de la pin 31 se duce pe primul pin din dreapta unde se duc și rstul de plusuri de la regiștrii și anume VCC și SRCLR.

Cu galben avem RCLK. Cu mov avem SRCLK. Cu verde avem Q7’ și SER.



Din bara de pini din dreapta am dus spre Arduino firele in ordinea aceasta: (vezi următoarele 2 poze)

1. GND – negru
2. VCC – roșu
3. D12 – albastru
4. D8 – galben
5. D11 – mov
6. A0 – verde



Display-ul LCD cu ajutorul modulului I2C folosește doar 4 pini: GND, VCC, SDA și SCL. În pinul 1 și 2 din dreapta (vezi schema 3) am adus si VCC-ul și GND-ul display-ului ajutându-mă de al treilea PCB pe care am fixat display-ul.

Cu alb am legat SDA-ul de pe modul la SDA-ul Arduino-ului, iar cu negru SCL-ul.

Instrucțiuni de utilizare

1. Se deschide aplicația Arduino IDE și se incarcă codul pe Arduino, dacă încă nu s-a făcut acest lucru.

2. Se fac toate legaturiile intre pinii de pe PCB-uri și Arduino folosiind cele 10 fire dupont tata-mama.

3. Se conectează Arduino-ul ori la laptop, ori la altă sursă de alimentare folosind cablul potrivit.

4. Pe ecran va apărea mesajul de bun-venit, apoi se va afișa scorul de 0-0. Player1 joacă mereu cu culoarea roșu, iar player2 cu albastru. Când unul dintre jucători câștigă se sting toate LED-urile, mai putin linia/coloana/diagonala câștigătoare, iar mai apoi și aceasta. Scorul se modifică, toate LED-urile se sting si poate incepe o rundă nouă. Pentru restetarea scorului se resetează Arduino-ul