



Universitatea Tehnică din Cluj-Napoca  
Catedra de Calculatoare

# Proiectare cu Microprocesoare -proiect-

**Student:** Pop Ruxandra Maria

**Grupa:** 30236

Data 03.12.2021

# Cuprins

<b>1.Cerința proiectului .....</b>	<b>3</b>
<b>2.Fundamente teoretice.....</b>	<b>5</b>
<b>3.Proiectare.....</b>	<b>7</b>
<b>4.Concluzii și dezvoltări ulterioare.....</b>	<b>13</b>
<b>5.Bibilografie.....</b>	<b>13</b>

# 1.Cerința proiectului

Proiectul de față reprezintă o aplicație muzicală, care are diverse opțiuni:

- Poate fi folosită ca și un pian, dacă se apasă primul buton mic.
- Poate fi folosită ca și un player muzical, dacă se apasă al doilea buton mic.
- Poate fi folosită ca un memory game, dacă se apasă al treilea buton mic.

În figura 1 se poate observa diagrama de decizie a proiectului, în funcție de apăsarea unui buton din cele 3 mici.

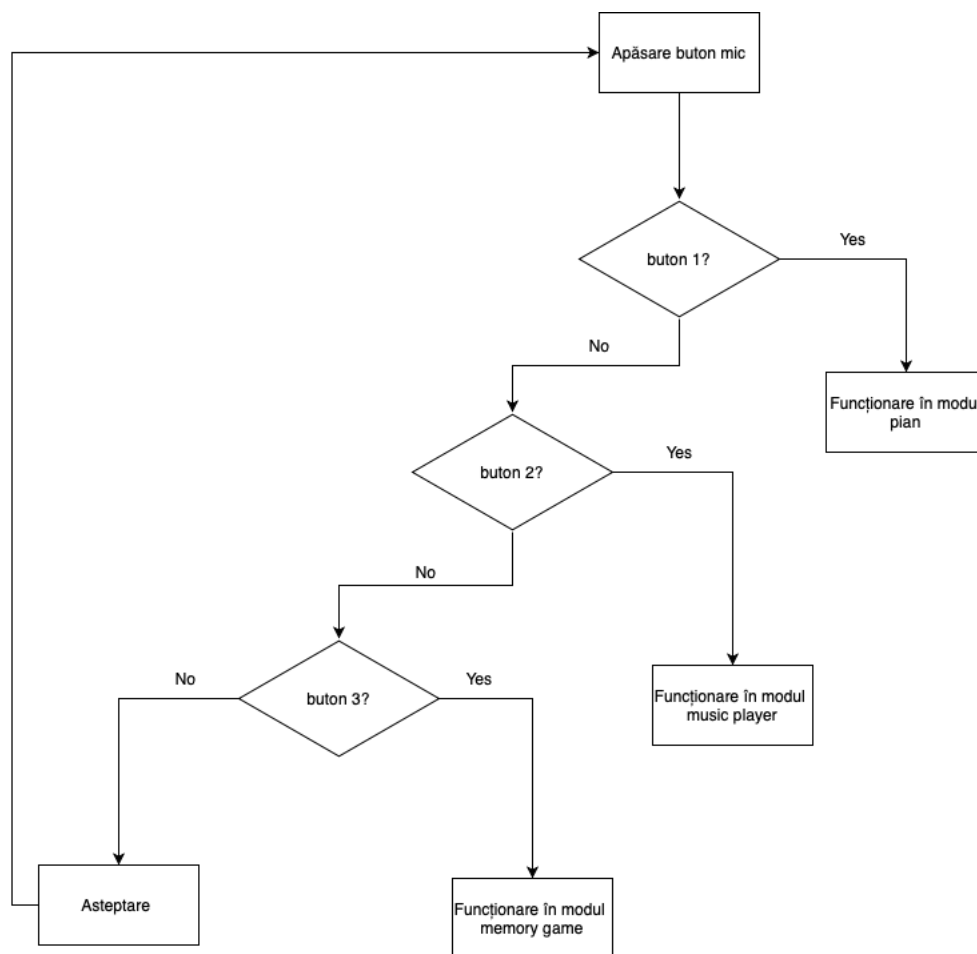


Figura 1 .Diagrama de stare

## ***Modul Pian***

În acest mod , am simulat un mini pian cu 5 taste,astfel utilizatorul are posibilitatea de a cânta la propriul pian ,prin apăsarea oricarui buton din cele 5 , fiecare buton având o nota diferită:

- Butonul 1 – are asociată nota NOTE\_C4 care are o frecvență de 262
- Butonul 2 – are asociată nota NOTE\_D4 care are o frecvență de 294
- Butonul 3 – are asociată nota NOTE\_E4 care are o frecvență de 330
- Butonul 4 – are asociată nota NOTE\_F4 care are o frecvență de 349
- Butonul 5 – are asociată nota NOTE\_G4 care are o frecvență de 392

## ***Modul Music Player***

În acest mod , am simulat un player muzical ,prin care utilizatorul are posibilitatea să asculte una din cele 5 melodii posibile , titlul melodiei fiind afișat pe lcd în timpul rulării.În funcție de ce buton mare e apăsat se pot reda următoarele melodii(a căror linie melodică se găsește în fișerele corespunzătoare):

- Butonul 1 – are asociată melodia din fișierul Song1() *Mario Theme Song*
- Butonul 2 – are asociată melodia din fișierul Song2() *Mario Underworld*
- Butonul 3 – are asociată melodia din fișierul Song3() *Pink Panther*
- Butonul 4 – are asociată melodia din fișierul Song4() *Happy Birthday*
- Butonul 5 – are asociată melodia din fișierul Song5() *We wish you a Merry Christmas*

## ***Modul Memory Game***

În acest mod , am simulat un joc de memorie cu ajutorul celor 4 led-uri și cu ajutorul a 4 butoane mari.Jocul are 10 niveluri de dificultate ,în funcție de numărul nivelului atâtea becuri se aprind ,iar după aia utilizatorul trebuie să apese butoanele în aceeași ordine în care s-au aprins anterior led-urile.Dacă utilizatorul apasă gresit butoanele , se va afișa un mesaj pe lcd de tipul *You Lose* ,și se va auzi un sunet de avertizare cum ca a *pierdut* iar după aia mesajul *Start Game* ,care avertizează ca jocul începe de la început cu nivelul 1.,Dacă utilizatorul a reușit să treacă de toate nivelurile se va afișa un mesaj *You win*, si se va auzi un sunet de avertizare cum ca a *câștigat* iar după aia mesajul *A new game start* ,care avertizează ca jocul începe de la început cu nivelul 1.

Fiecarui buton ii este asociat un led astfel:

- Butonul 2 – are asociat led-ul roșu
- Butonul 3 – are asociat led-ul alb
- Butonul 4 – are asociat led-ul albastru
- Butonul 5 – are asociat led-ul galben

## 2.Fundamente teoretice

Componentele folosite pentru realizarea proiectului sunt:

- **Placa Arduino Uno**

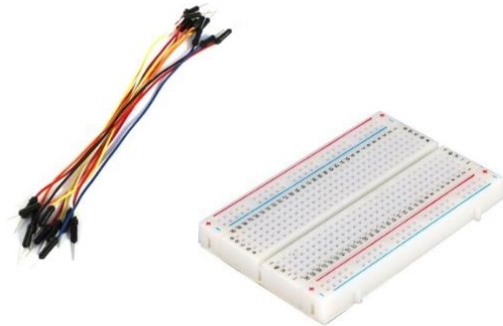
Familia de unelte de dezvoltare Arduino include plăci cu microcontroller ,accesorii și componente software open source ,care permit utilizatorilor să realizeze proiecte folosind o abordare unificată ,de nivel înalt ,care se dorește a fi independent de microcontrollerul folosit. Placa Arduino Uno folosită pentru realizarea acestui proiect, este echipata cu un microcontroller ATmega328, are 14 pini de intrare/ieșire , din care 6 pot fi folosiți ca iesiri PWM , 6 ca intrări analogice , un oscilator de 16MHz ,o conexiune USB, o mufă de alimentare , și un buton de reset.



În realizarea proiectului m-am folosit de 4 pini analogi pentru butoane , 6 pini digitali pentru LCD , un pin digital pentru buzzer ,și încă 4 pini pentru led-uri.

- **BreadBoard și fire**

Breadboard-ul este folosit la extinderea planului de proiectare ,mai exact un semnal poate fi folosit în mai multe locuri, deoarece fiecare pin se poate conecta prin cele 4 găuri așezate perpendicular pe circuit. Firele sunt folosite la conectarea componentelor între ele și la transmiterea semnalelor.



- **Led-uri**

Led-urile sunt ieftine ,consumă o putere semnificativ mai mică decât alte opțiuni de iluminat .  
Am folosit 4 led-uri de culoare : roșu , galben, albastru ,alb ; sunt folosite la jocul de memorie.



- **Rezistențe**

Rezistențele sunt componente care restricționează cantitatea de curent care trece prin circuit. Pot avea diferite valori. În cazul proiectului, toate rezistențele au o valoare de 220 Ohm.



- **Potențiometru**

Potențiometrul oferă o rezistență variabilă pe care o putem citi pe placa Arduino ca valoare analogică. Am folosit un potențiometru de 10k ohm pentru a controla calitatea afisării datelor pe LCD.



- **Buzzer**

Buzzer-ul este folosit pentru a genera sunete la o anumită frecvență.

Am folosit această componentă pentru a reda sunetele necesare pentru pian, pentru a reda cele 4 melodii, și pentru a genera un sunet în cazul în care utilizatorul a câștigat/pierdut.



- **LCD**

LCD este folosit pentru a oferi o interfață cu utilizatorul. Va afișa mesaje pe ecran, în funcție de stadiu în care se află aplicația la un moment dat. Permite afisarea a maxim 32 de caractere în același timp.



- **Butoane**

Butonul este o componentă care conectează două puncte dintr-un circuit când e apăsat. Prezintă rezistențe de PULL UP interne, astfel când e apăsat va avea valoarea LOW, iar când nu e apăsat va avea valoarea HIGH.

Am folosit 8 butoane în total: 3 mici conectate fiecare la propriul pin analogic (il folosesc ca și un pin digital) și 5 butoane mari conectate la un singur pin analogic (conectare în lanț/serie cu ajutorul a 6 rezistențe).

Această conectare va fi prezentată mai în detaliu în următorul capitol.



### 3.Proiectare

În continuare voi prezenta schema generală a circuitului în care se vor putea observa toate componentele și conexiunile dintre acestea.

Schema a fost realizată cu *Circuit Designer*.

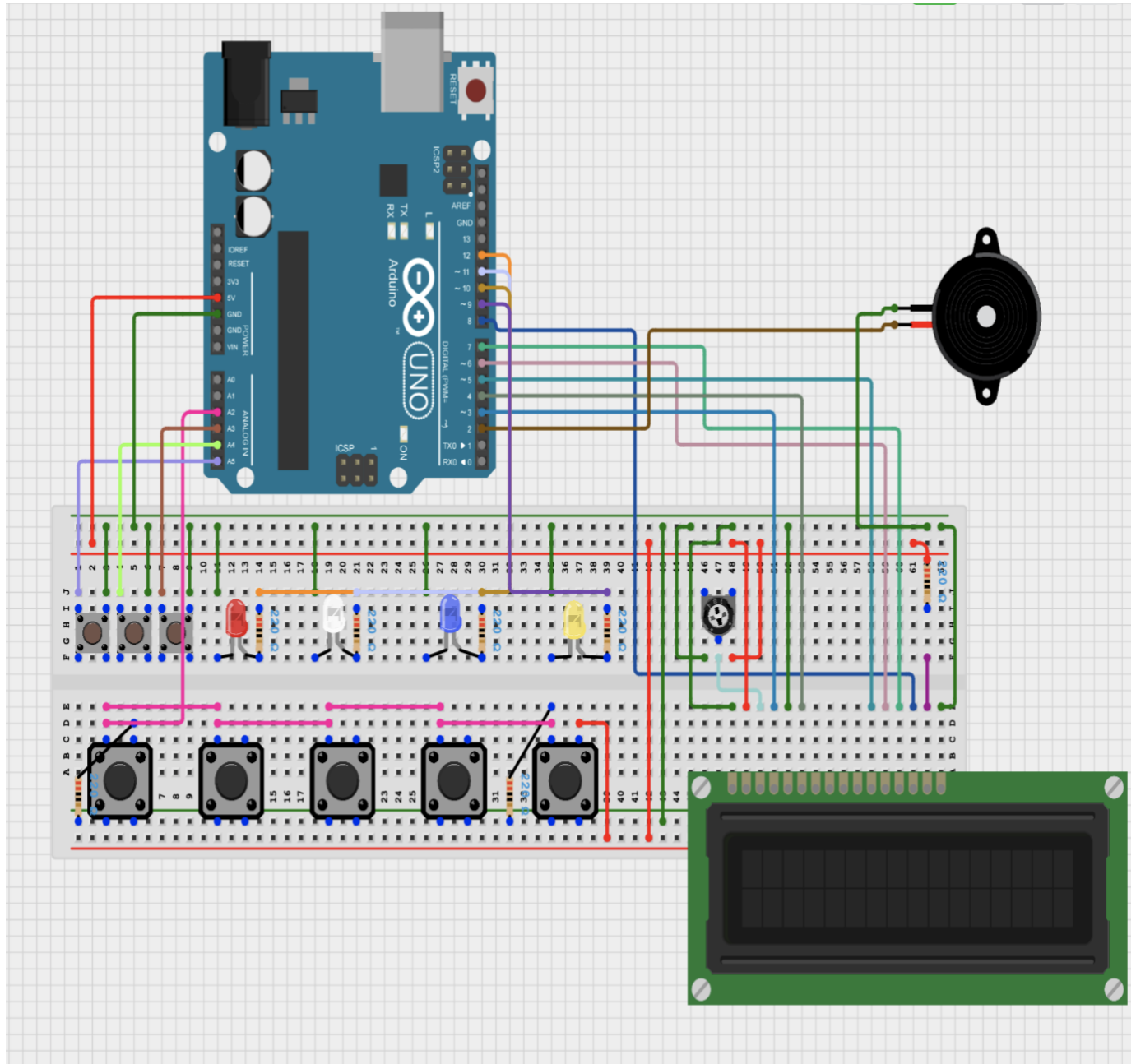


Figura 3.1. Schema completă a proiectului



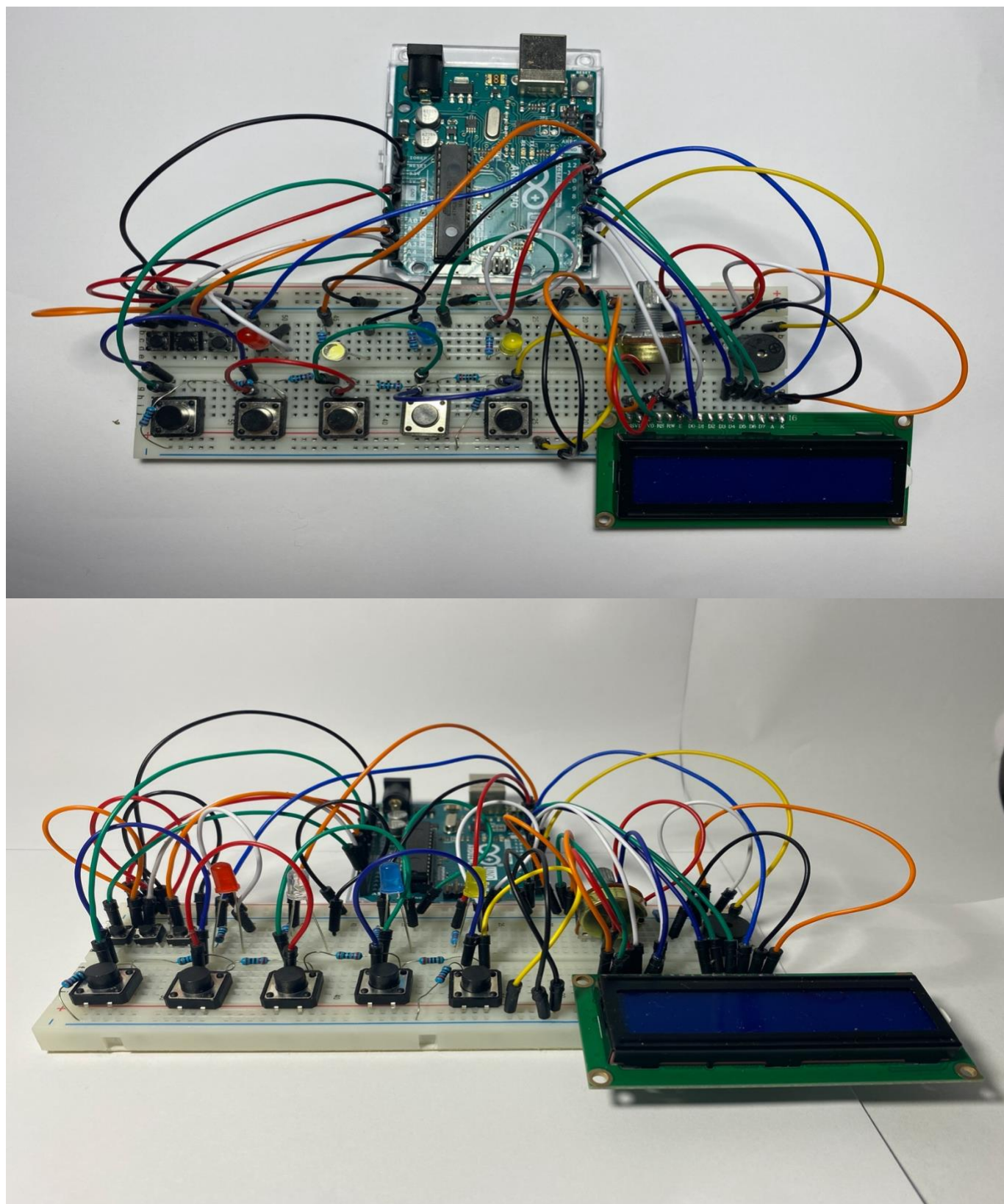


Figura 3.2. Interfața cu utilizatorul



Pentru că placa Arduino Uno are un număr redus de pini , am ales să folosesc o conexiune în serie pentru cele 5 butoane mari, putând astfel să le controlez pe toate doar printr-un singur pin analogic . Astfel se creează un divizor de tensiune ,pentru care fiecare buton aduce un alt rezistor , permițându-ne să facem diferența între ele. Pentru a fi mai ușor a calcula valoarea la care se va activa un buton , toate rezistențele folosite la montare sunt de 220 Ohm.

Butonul 1 – se activează atunci când valoarea citită de pe intrarea analogică este între 100 și 150

Butonul 2 – se activează atunci când valoarea citită de pe intrarea analogică este între 150 și 250

Butonul 3 – se activează atunci când valoarea citită de pe intrarea analogică este între 250 și 350

Butonul 4 – se activează atunci când valoarea citită de pe intrarea analogică este între 350 și 850

Butonul 5 – se activează atunci când valoarea citită de pe intrarea analogică este între 850 și 1023

În figura 3.4 se poate observa cum sunt conectate butoanele în funcție de rezistențe.

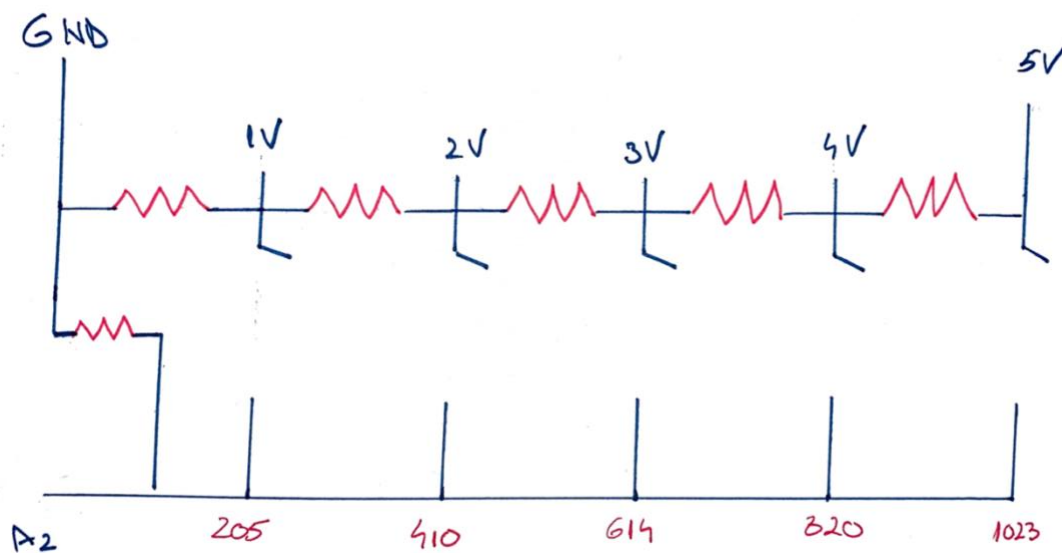


Figura 3.4. Legarea rezistențelor la butoane

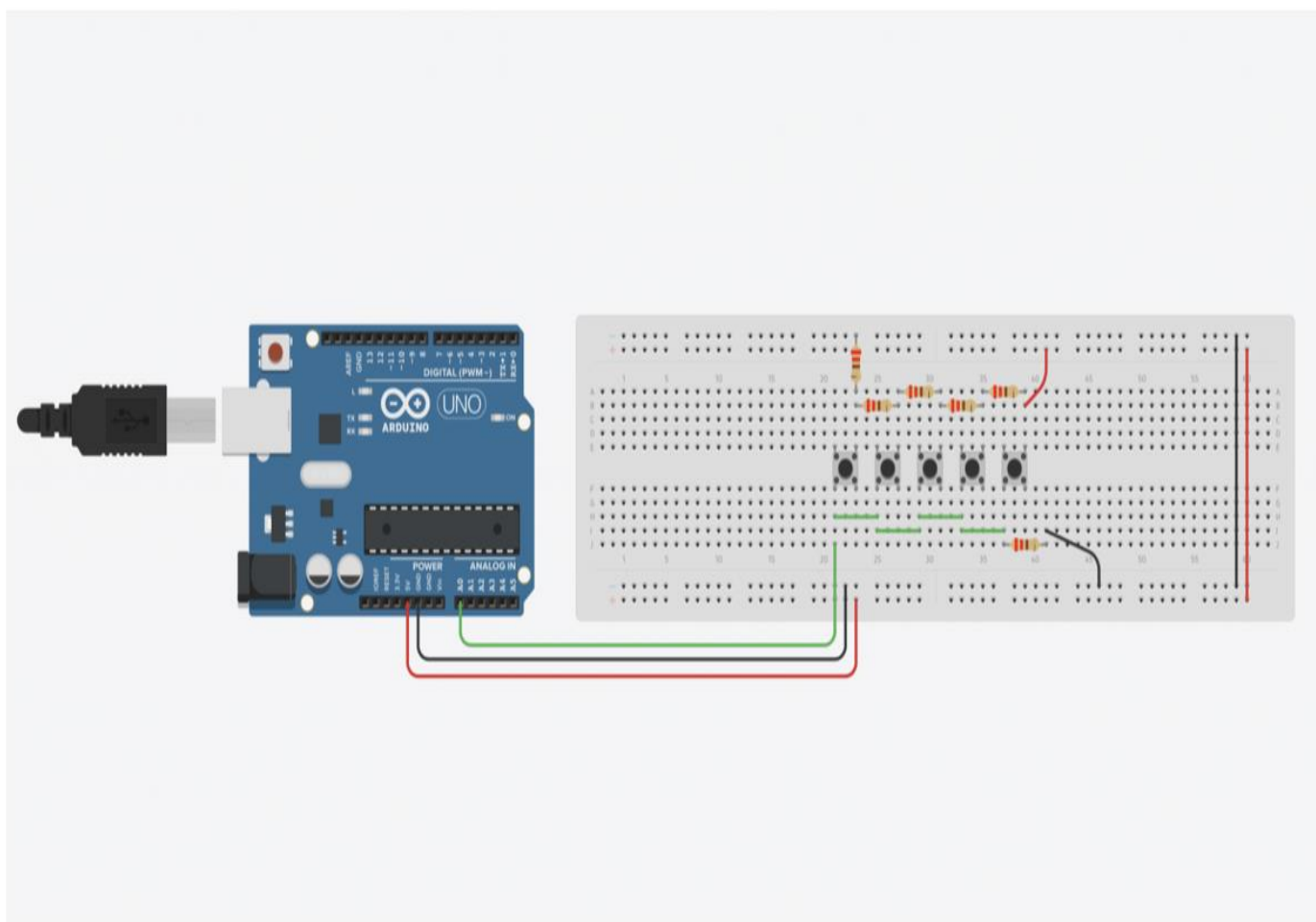


Figura 3.5. Schema de conectare în mediu Arduino a butoanelor

Funcțiile folosite în implementarea proiectului sunt:

- void configPin() – este funcția care leagă toate semnalele la pinii plăci Arduino UNO
  - am folosit for-uri pentru butoane și led-uri
- void configLcd() – este funcția care inițializează Lcd-ul și afișează primul mesaj din aplicație
- void playPiano () – este funcția principală pentru modul Pian ,în funcție de valoarea recepționată de la pinul analog A2 se va reda nota corespunzătoare
- void musicStart() -este funcția principală pentru modul Music Player, și conține cele 5 funcții ,care se află în fișiere diferite , pentru fiecare din cele 5 melodii , și în funcție de butonul apăsător se va reda una din ele.
- void gameStart() – este funcția principală pentru modul Music Player .Această funcție conține automatul de stare prezentat în figura 3 ,precum și următoarele funcții care depind de starea în care se afla automatul la un moment dat :
  1. STATE\_START\_GAME : void generateRandom() – care selectează care led-uri se vor aprinde , numărul led-urilor aprinse este restricționat de nivelul la care se află utilizatorul
  2. STATE\_SHOW\_LED : void showRandom() – care va aprinde led-urile selectate în starea precedentă
  3. STATE\_READ: int readOption() – care va returna 0 dacă utilizatorul vrea să oprească jocul sau 1 dacă s-au citit toate intrările date de utilizator. Aici se adaugă în vectorul playerRead ,pini led-urilor pe care utilizator le consideră ca au fost aprinse în stare anterioară.Pinii led-urile sunt selectate cu ajutorul butoanelor asociate pentru fiecare led separat.
  4. STATE\_VERIFY : bool isCorrect () – care verifică dacă intrările date de utilizator corespund cu vectorul care conține valorile corecte.
  5. STATE\_CORRECT: void soundCorrect() – prin care se va reda pe buzzer un sunet care să sugereze că utilizatorul a trecut de toate nivelurile,și implicit că a câștigat.
  6. STATE\_AGAIN : void soundInccorect () prin care se va reda pe buzzer un sunet care să sugereze că utilizatorul nu a trecut de nivelul curent , și implicit că e nevoie să reînceapă jocul de la nivelul 1
- void Song1()- conține notele pentru melodia *Mario Theme Song*
- void Song2()- conține notele pentru melodia *Mario Underworld*
- void Song3()- conține notele pentru melodia *Pink Panther*
- void Song4()- conține notele pentru melodia *Happy Birthday*
- void Song5()- conține notele pentru melodia *We wish you a Merry Christmas*

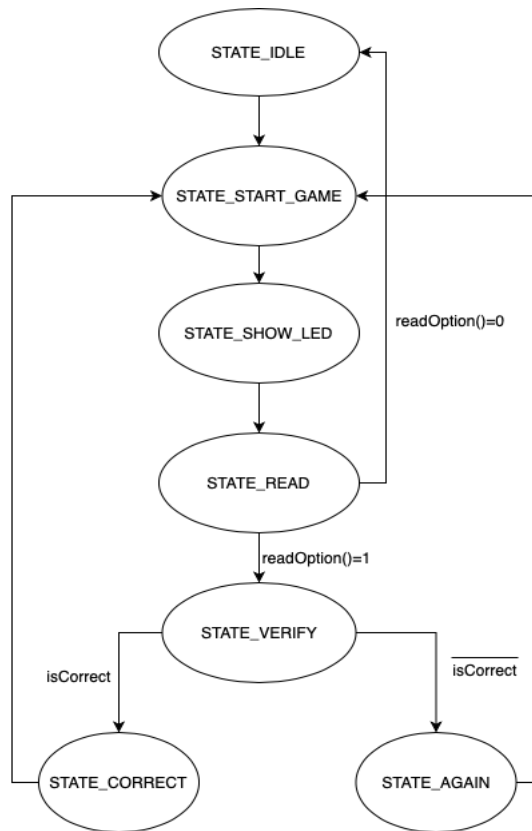


Figura 3. Automatul de stare pentru modul Memory Game

- STATE\_IDLE : este starea de început a automatului , unde se vor afișa pe ecran secunde până când va începe jocul.
- STATE\_START\_GAME : este starea unde se va genera secvența de led-uri ca vor urma să se aprindă
- STATE\_SHOW\_LED : este momentul în care se vor aprinde led-urile alese în starea anterioară
- STATE\_READ : este starea unde se vor citi intrările date de utilizator .Dacă readOption este 0 adică utilizatorul vrea sa iasă din modul Memory Game , stare următoare se va reseta ca fiind starea inițială a automatului
- STATE\_VERIFY : este starea unde se verifică dacă intrările utilizatorului corespund cu secvența aleatorie de led-uri generata în starea STATE\_START\_GAME
- STATE\_CORRECT : se ajunge în această stare dacă verificarea din starea anterioara este corectă. În această stare se incrementează numărul nivelului la care se află utilizatorul
- STATE\_AGAIN : se ajunge în această stare dacă verificarea din starea anterioara este greșită. Numărul nivelului la care se află utilizatorul se resetează la 1.

*Am implementat acest automat folosind funcția switch.*

## 4.Concluzii și dezvoltări ulterioare

În concluzie ,dezvoltarea unui asemenea proiect a făcut mult mai ușoară înțelegerea folosirii unei plăcuțe Arduino și a componentelor acesteia ,precum și a funcțiilor predefinite. Mai mult de atât am învățat cum să codific notele unei melodii în cod Arduino. Ca și dezvoltări ulterioare , as propune utilizarea unui număr mai mare de butoane pentru pian, mai multe efecte și caractere afișate pe LCD,precum și utilizarea unui buzzer cu volum de redare mai puternic ,sau utilizarea simultana a mai multor buzzere.

## 5.Bibilografie

1. <https://create.arduino.cc/projecthub/rowan07/make-a-simple-led-circuit-ce8308>
2. <https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld>
3. <https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45>