

**Facultatea de Automatică și Calculatoare**

**Proiectare cu microprocesoare**

**Joc cu labirint controlat prin joystick**

Profesor îndrumător:

Mircea Paul Mureșan

Student:

Maria-Larisa Radu

Grupa: 30238

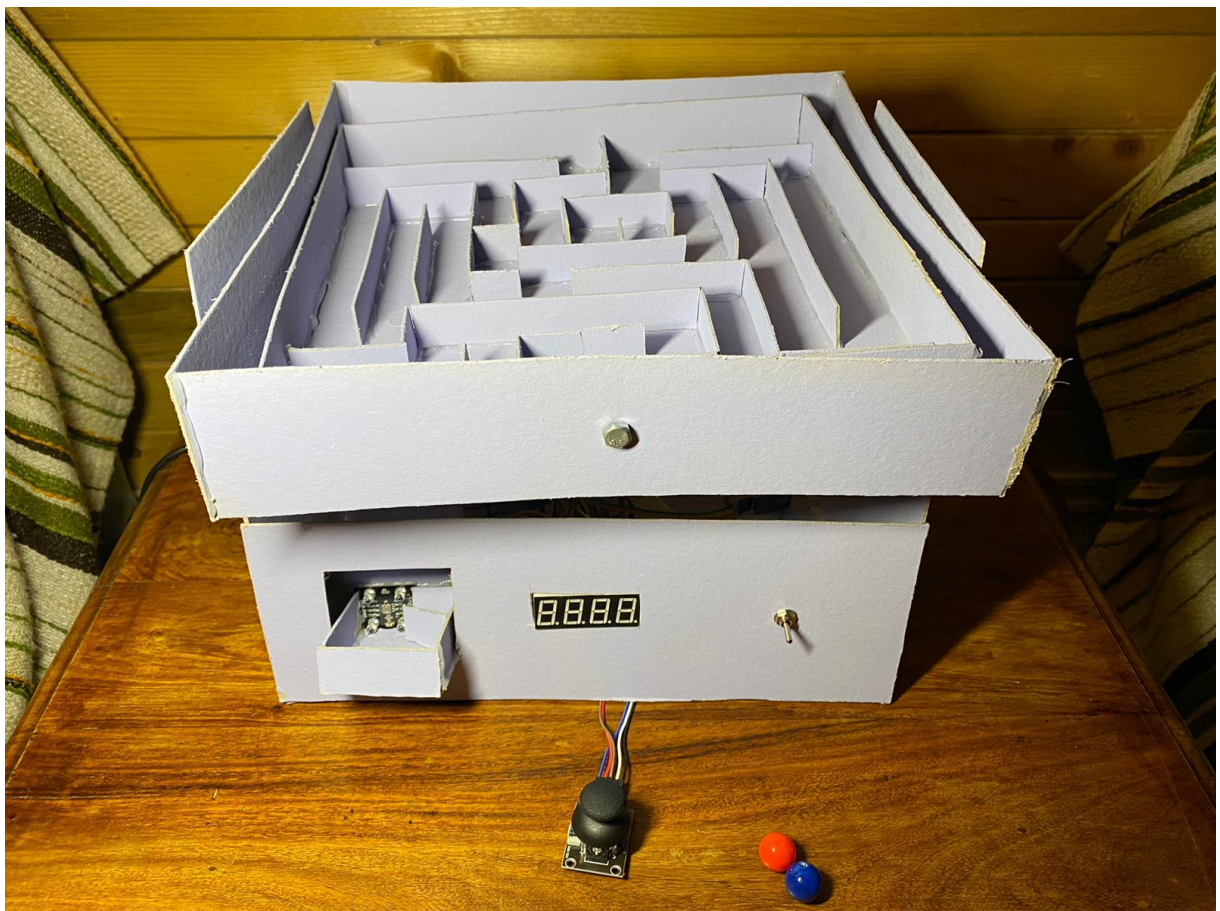
## **Cuprins**

- I.      Specificația proiectului
- II.     Componente utilizate
- III.    Montaj
  - i.      Materiale și ustensile
  - ii.     Construcție si design
  - iii.    Conectarea componentelor
- IV.    Cerințe de sistem
- V.     Implementare
- VI.    Instrucțiuni de utilizare
- VII.   Bibliografie

## I. Specificația proiectului

Jocul cu labirintul și bilele este unul interactiv, destinat unui singur jucător. Scopul acestuia este aducerea bilei de culoare albastră la capătul labirintului pentru a cădea în coș. De asemenea, jocul poate fi pierdut în cazul în care în coș ajunge bila roșie.

Obiectivul acestui proiect este construirea unui astfel de joc din carton, utilizând de asemenea și componente Arduino pentru asigurarea unor funcționalități. Astfel, s-au utilizat numeroase module, îmbinate cu o serie de senzori, toate acestea conectate la un microcontroler programat cu ajutorul unui mediu de dezvoltare numit Arduino IDE.



*Poză de prezentare*

## II. Componente utilizate

Componenta principală utilizată pentru realizarea acestui proiect este plăcuța Arduino Mega 2560. Pe acest microcontroler se realizează atât legăturile între celelalte componente, cât și legătura cu laptopul. Această placă de dezvoltare dispune de numeroși pini de comunicație, analogici și pwm, pentru toate tipurile de componente, astfel: 54 de pini digitali

pentru intrare/ieșire și 16 pini analogici. Frecvența acestui microcontroler este de 16 MHz și este alimentat printr-un cablu USB conectat la un calculator.



*Arduino Mega 2560*

Conectate la acest microcontroler s-au mai utilizat următoarele componente:

→ **Mini servomotor:** acesta este proiectat special pentru aplicații de mică putere. Pentru realizarea proiectului s-au utilizat două servomotoare.

**Funcționare:** de servomotor se prinde o elice care poate fi rotită cu un număr exact de grade, prezentând o precizie ridicată; acesta prezintă un pin pentru alimentare, unul pentru Ground și unul de tip pwm.



*Mini servomotor*

→ **Modul joystick:** la realizarea proiectului s-a utilizat un astfel de modul pentru controlarea înclinării labirintului.

**Funcționare:** modulul se conectează la placa Arduino căreia îi transmite două semnale analogice reprezentate de poziția în spațiul 2D a joystick-ului, și un semnal digital dat de un buton; acesta prezintă un pin de Ground, unul pentru

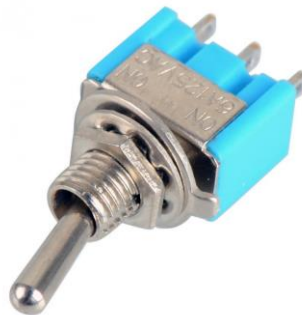
alimentare, un pin analogic pentru X, un pin analogic pentru Y și un pin digital pentru buton.



*Modul joystick*

→ **Întrerupător cu două poziții:** la realizarea proiectului s-a utilizat un astfel de întrerupător pentru a marca începerea jocului.

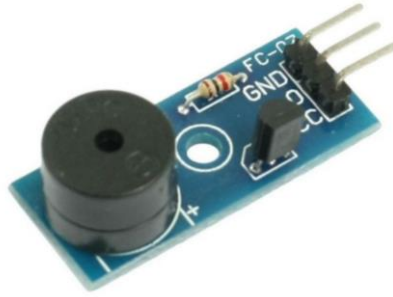
**Funcționare:** modulul se conectează la placă prin 2 pini: unul dintre aceștia se leagă la alimentare, iar celălalt, printr-o rezistență legată la Ground, se leagă la un pin digital.



*Întrerupător*

→ **Modul buzzer:** acesta este un modul care generează un sunet continuu, de frecvență fixă; la realizarea proiectului s-a utilizat un astfel de modul pentru semnalarea atât a începerii jocului, cât și a câștigului sau pierderii acestuia.

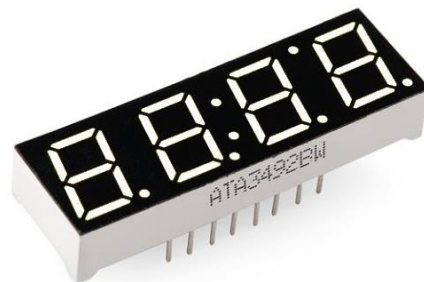
**Funcționare:** modulul se conectează printr-un pin la alimentare, prin altul la Ground și prin ultimul pin la pwm.



*Modul buzzer*

→ **Afișor 7 segmente cu 4 cifre:** această componentă afișează până la 4 cifre sau litere, utilizând un schelet cu 7 segmente pentru care se aprind doar anumite segmente ca să se formeze cifra dorită; la realizarea acestui proiect s-a utilizat această componentă cu scopul afișării scorului.

**Funcționare:** modulul prezintă 12 pini: 7 corespunzători fiecărui segment al unei cifre, 4 corespunzători fiecărei cifre și unul corespunzător punctului dintre cifre. Toți acești pini se pot conecta la pinii digitali ai plăcii.



*Afișor 7 segmente cu 4 cifre*

→ **Modul senzor de culoare:** această componentă detectează culoarea obiectului care este plasat în dreptul ei; la realizarea proiectului, componenta s-a utilizat pentru a detecta care din bile a ajuns la final.

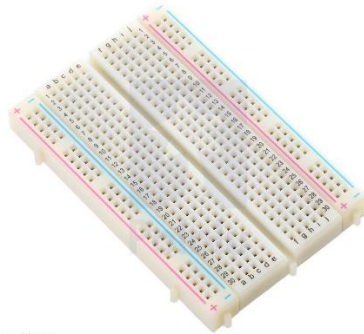
**Funcționare:** senzorul se bazează pe conversia lungimii de undă a luminii într-un semnal dreptunghiular de o anumită frecvență pentru fiecare culoare.



*Modul senzor de culoare*

→ **Breadboard:** acesta permite plasarea modulelor, senzorilor, ledurilor și a altor componente fără a fi necesară lipirea acestora, permițând schimbarea rapidă a oricărei configurații de circuit; la realizarea proiectului, breadboardul a fost folosit pentru alimentarea multiplelor componente, adăugarea de rezistențe și crearea circuitului necesar funcționării corecte a proiectului.

**Funcționare:** pentru utilizare este suficientă introducerea unui pin într-un spațiu din breadboard și utilizarea celorlalți pini de pe același rând cu acesta.



*Breadboard*

→ **Fire jumpers:** acestea permit atât conectarea componentelor cu plăcuța, cât și conectarea cu breadboardul; pentru acest proiect s-au utilizat numeroase astfel de fire (de tipul tată-tată, tată-mamă și mamă-mamă) pentru conectarea tuturor componentelor.

**Funcționare:** aceste fire trebuie conectate la pinii componentelor în funcție de tipul potrivit.





*Fire jumpers*

→ **Rezistență 250 ohm:** aceasta se utilizează la legarea ieșirii întrerupătorului la Ground



*Rezistență*

### III. Montaj

i. **Pentru realizarea montajului a fost nevoie de utilizarea unor materiale și ustensile.**

Acestea sunt:

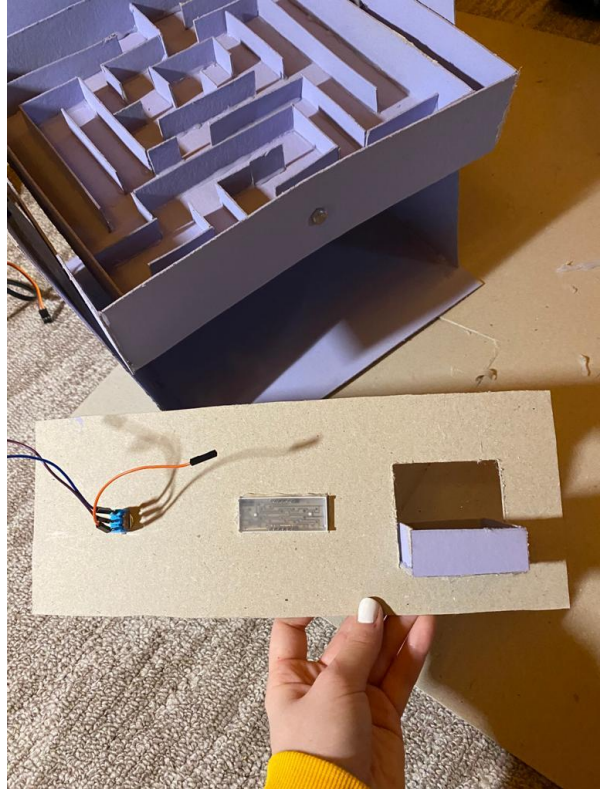
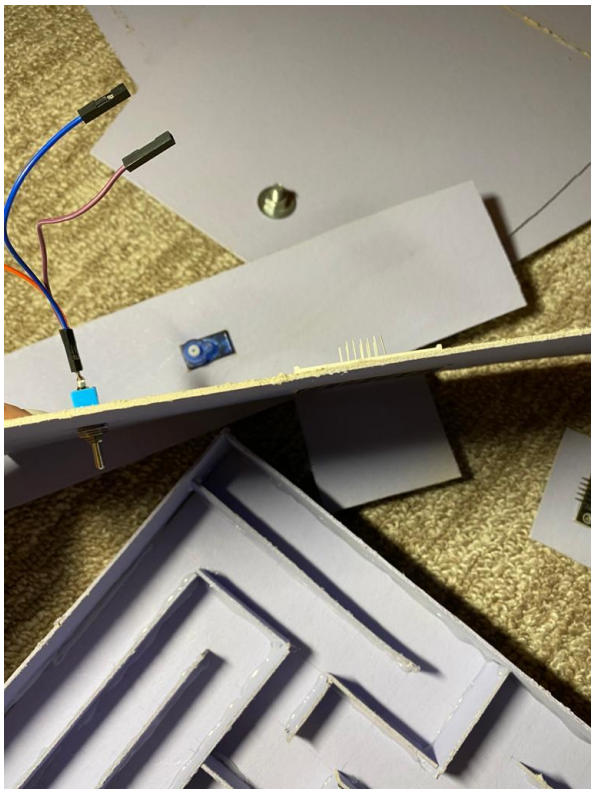
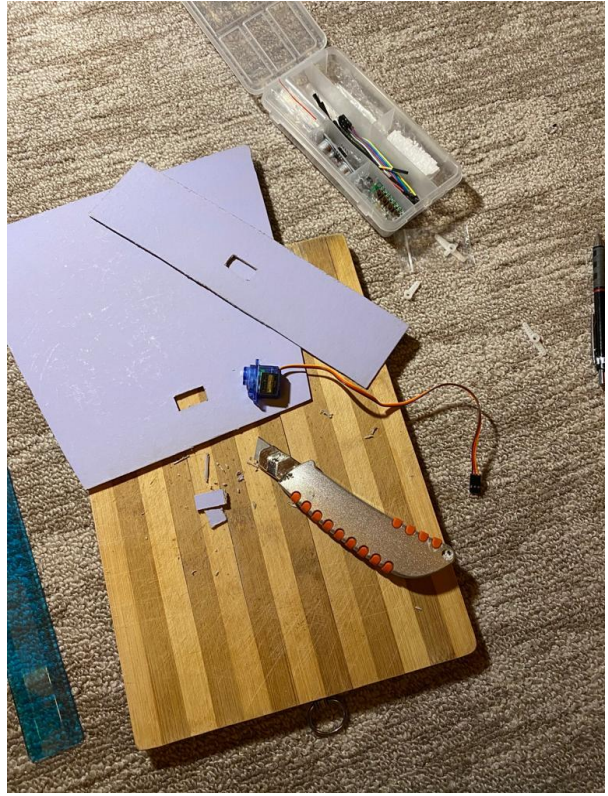
- Carton cu grosime de 2mm: cu acest carton s-a realizat întreaga construcție a proiectului. Grosimea acestuia este importantă deoarece, pentru a se putea mișca fără probleme, cartonul utilizat trebuie să fie puțin dur pentru a nu se îndoi sau rupe, dar cu toate acestea greutatea lui nu trebuie să fie mare deoarece servomotoarele utilizate sunt pentru proiecte mici.
- Creion pentru desenarea conturului pe carton
- Foarfecă pentru decuparea cartonului
- Pistol de lipit componentele decupate de carton



## ii. Construcție și design:

Pașii urmați în realizarea construcției din carton:

- Se decupează un pătrat cu latura de 40 cm pentru baza construcției
- Se decupează două trapeze isoscele cu baza inferioară de 35 cm, baza superioară de 30 cm și înălțimea de 30 cm
- Se lipesc aceste 2 trapeze de 2 laturi opuse ale bazei, cu baza inferioară jos
- Se decupează 4 dreptunghiuri cu lungimea de 35 cm și lățimea de 5 cm
- Se lipesc aceste dreptunghiuri pentru a crea un contur de pătrat
- Se decupează un pătrat cu latura de 30 cm pentru baza labirintului propriu-zis
- Se decupează 4 dreptunghiuri cu lungimea de 30 cm și lățimea de 5 cm
- Se lipesc aceste dreptunghiuri pentru a crea un contur de pătrat, la care se lipește baza labirintului
- Se decupează mai multe dreptunghiuri de lungimi diferite și lățimea de 3 cm
- Se lipesc aceste dreptunghiuri pe baza labirintului pentru a crea diferite căi de a traversa labirintul
- Se decupează un pătrat într-un colț de pe baza labirintului
- Se decupează unul din trapeze pentru a se introduce un servomotor, și se lipește elicea acestuia de conturul exterior al pătratului, iar în partea opusă, între celălalt trapez și contur exterior se introduce un șurub pentru a permite mișcarea
- Se decupează o latură a conturului exterior (adiacentă cu trapezul cu servomotorul) pentru a se introduce celălalt servomotor, și se lipește elicea acestuia de conturul interior al pătratului, iar în partea opusă, se introduce din nou un șurub pentru a permite mișcarea
- Se decupează un dreptunghi cu lungimea de 35 cm și lățimea de 10 cm
- Se face o gaură în acest carton pentru fixarea întrerupătorului
- Se decupează un contur în același carton pentru fixarea afișorului
- Se decupează un pătrat mic și se introduce un coș creat din carton în spațiul decupat
- Pe spatele coșului și în lateralul acestuia se lipesc 2 dreptunghiuri de carton pentru a asigura căderea bilei în coș
- Se lipește acest dreptunghi de suporturile laterale astfel încât gaura din labirint să fie deasupra coșului și servomotorul prins de labirintul exterior și cel interior să fie așezat în spate
- De coșul de carton se lipește senzorul de culoare pentru a cădea bila în fața acestuia
- Joystick-ul se va așeza în fața construcției din carton
- Plăcuța, breadboard-ul și buzzerul se vor așeza în interiorul construcției, pentru a nu fi la vedere







*Realizarea construcției de carton*

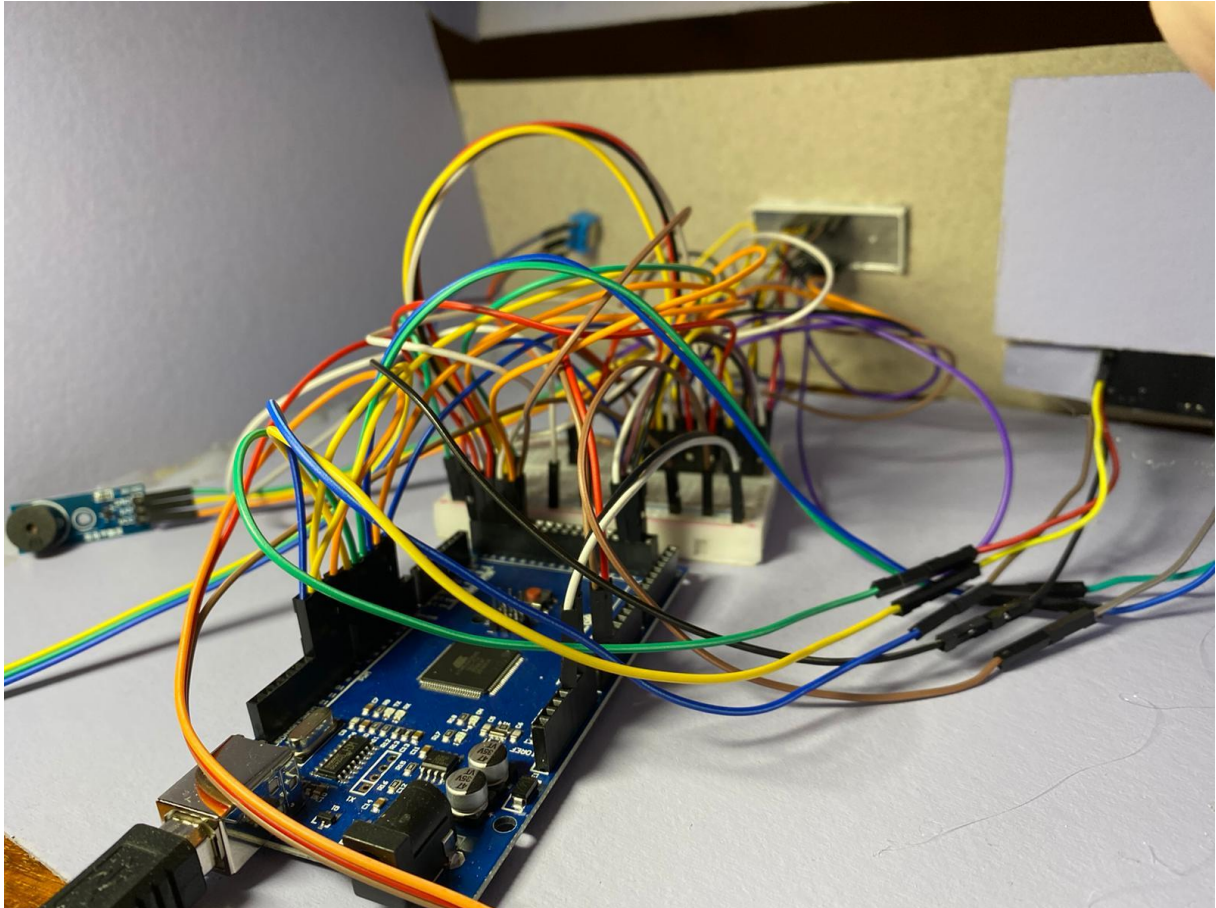
### **iii. Conectarea componentelor:**

Placa de dezvoltare Arduino Mega a fost conectată prin USB la calculator, iar restul componentelor s-au conectat la pinii plăcii, prin fire jumper și eventual trecerea prin breadboard, astfel:

- Servomotorul responsabil cu rotația în jurul axei X: pinul Signal a fost conectat direct la pinul pwm 8 de pe placă
- Servomotorul responsabil cu rotația în jurul axei Y: pinul Signal a fost conectat direct la pinul pwm 9 de pe placă
- Joystick-ul: pinii VRx și VRy au fost conectați la pinii analogici A0 respectiv A1 de pe placă, iar SW a fost conectat la pinul pwm 2.
- Întrerupătorul: unul dintre cei 2 pini laterali (oricare) a fost conectat la pinul pmw 3 de pe placă. Această conexiune a fost trecută prin breadboard și printr-o rezistență, care apoi s-a legat la Ground
- Buzzerul: pinul I/O a fost conectat la pinul pwm 10 de pe placă
- Afișorul 7 segmente cu 4 cifre: pinii corespunzători celor 7 segmente au fost conectați la pinii digitali ai plăcii: 22, 23, 24, 25, 26, 27, 28, 29; iar pinii corespunzători celor 4 cifre au fost conectați la pinii digitali: 50, 51, 52, 53. Întreaga conexiune cu afișorul a fost trecută prin breadboard

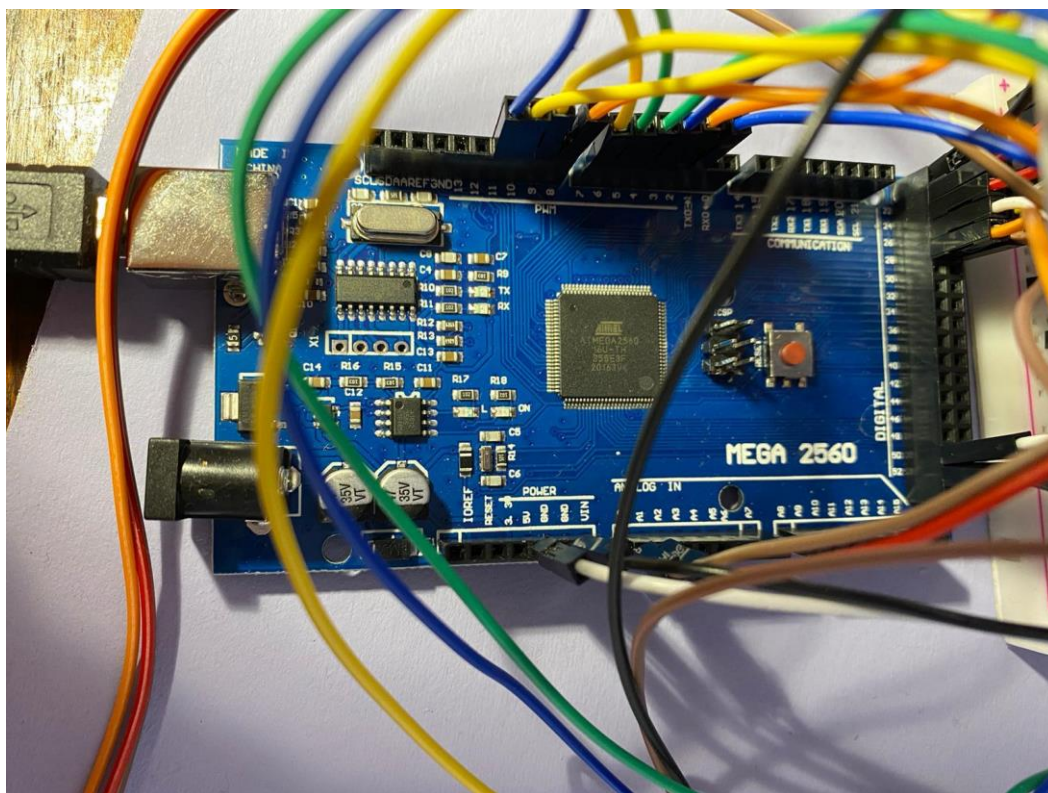
→ Senzorul de culoare: pinii s0, s1, s2, s3 au fost conectați la pinii pwm ai plăcii: 4, 5, 6, 7; iar pinul de ieșire a fost conectat la pinul 11 al plăcii

Toți pinii de VCC s-au conectat pe rândul de VCC de la breadboard, unde s-a conectat un fir adus de la un pin VCC de la placa Mega. De asemenea, toți pinii de Ground s-au conectat pe rândul de Ground de la breadboard, unde s-a conectat un fir adus de la un pin Ground de la placa Mega.

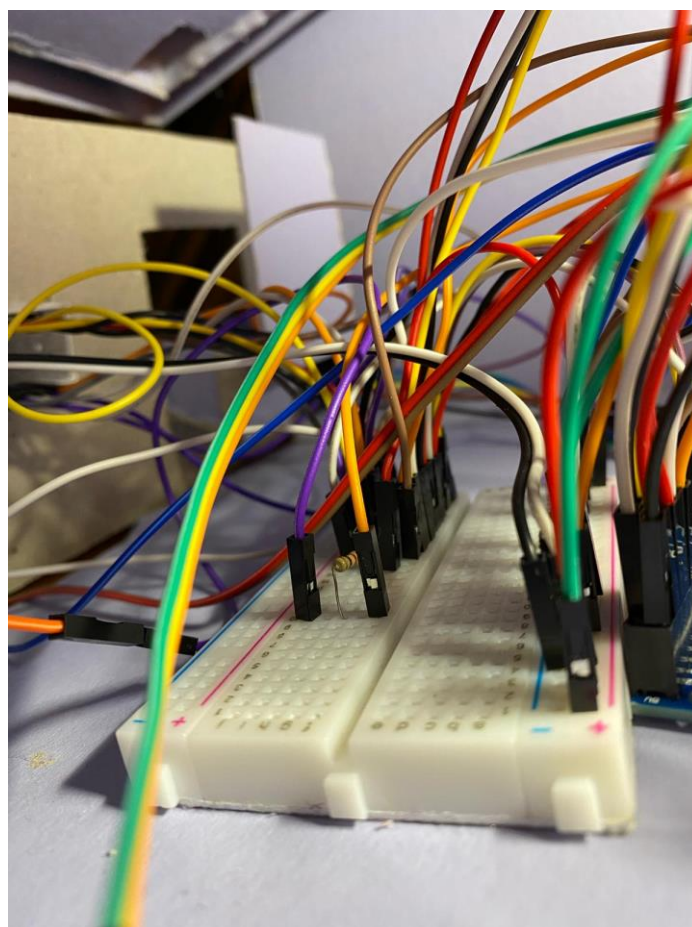


*Poză generală cu conexiunile dintre componente*





*Poză în detaliu cu legăturile făcute pe Mega 2560*



*Poză în detaliu cu legăturile făcute pe breadboard*

## IV. Cerințe de sistem

Pentru crearea acestui proiect s-a folosit mediul de dezvoltare Arduino IDE, descărcat de pe site-ul oficial Arduino. Codul este scris în limbajul C.

Pentru conectarea plăcii, s-a selectat în Arduino, la Instrumente, la Placă: “Arduino Mega or Mega 2560”, și portul a fost selectat la Instrumente, la Port.

Pentru funcționarea corectă și eficientă a proiectului s-au utilizat 3 biblioteci. Acestea vor fi precizate în cele ce urmează:

- “*pitches.h*” : în acest fișier s-au definit notele și valorile numerice corespunzătoare lor pentru interpretarea corectă a sunetului de către buzzer și pentru înfrumusețarea și ordonarea codului
- “*SevSeg.h*” : această bibliotecă a fost descărcată de pe internet; este responsabilă de controlul afișorului cu 7 segmente; funcționează atât pentru anod comun cât și pentru catod comun, și, de asemenea, poate adăuga rezistențe pe segmente
- <*Servo.h*> : această bibliotecă este responsabilă cu controlul servomotoarelor utilizate

## V. Implementare

În cele ce urmează, va fi explicat codul utilizat pentru funcționarea jocului.

Semnificația variabilelor utilizare:

- *melodyGameStart[], melodyGameWon[], melodyGameLost[]* (liniile 6-8) : aceste variabile sunt vectori de int-uri cu notele muzicale descrise în fișierul “*pitches.h*”, fiecare dintre acestea conținând un cântec pentru câte o situație
- *durationsGameStart[], durationsGameWon[], durationsGameLost[]* (liniile 11-13): aceste variabile sunt vectori de int-uri cu valorile duratelor celor 3 cântece declarate anterior
- *songLengthGameStart, songLengthGameWon, songLengthGameLost* (liniile 16-18): acești întregi reprezintă lungimile celor 3 melodii
- *buzzer\_pin* (linia 21): reprezintă pinul de pe microcontroler la care este conectat buzzerul
- *sing\_start\_buzzer* (linia 23): reprezintă un flag care indică dacă se poate sau nu porni cântecul pentru începutul jocului
- *servoX, servoY* (liniile 26-27): sunt instanțe ale obiectului Servo, reprezentând cele două servomotoare utilizate în acest proiect
- *pinServoX, pinServoY* (liniile 30-31): reprezintă pinii de pe placă corespunzători controlării celor două servomotoare
- *VRx, VRy, SW* (liniile 33-36): reprezintă pinii de pe placă corespunzători joystick-ului
- *switch\_pin* (linia 39): reprezintă pinul de pe placă corespunzător switch-ului
- *xPos, yPos, SWState* (liniile 42-44): reprezintă variabile folosite pentru citirea informațiilor trimise de joystick
- *mapX, mapY* (liniile 47-48): reprezintă variabile folosite pentru stocarea valorii returnate de maparea unei valori de la un anumit interval la altul

- *servoPosX*, *servoPosY* (liniile 51-52): reprezintă înclinația de început a elicelor servomotoarelor (motivul pentru care valorile nu sunt 90, 90 este faptul că la lipire nu au fost îndreptate perfect elicele, așa că, pentru stabilirea unei poziții de repaus perfect drepte, a fost nevoie de devierea puțin a unghiului de 90 de grade așteptat)
- *sevseg* (linia 55): reprezintă o instanță a obiectului SevSeg, folosită pentru controlarea afisorului BCD 7 segmente cu 4 cifre
- *nr* (linia 57): este un vector de 5 caractere și reprezintă scorul care va fi afișat pe afișor; acesta este de tip char și nu int deoarece pentru precizarea culorilor (adica a literelor corespunzătoare) nu s-a putut utiliza int
- *s0*, *s1*, *s2*, *s3*, *out\_sensor* (liniile 60-64): reprezintă pinii de pe placă corespunzători senzorului de culoare
- *frequencyR*, *frequencyB*, *frequencyG* (liniile 67-69): reprezintă variabile în care se vor stoca frecvențele percepute de către senzorul de culoare cu privire la cele 3 culori principale

Funcțiile *singGameStart()*, *singGameWon()* și *singGameLost()* (liniile 252-283) sunt funcții care activează cele 3 cântece și așteaptă terminarea acestora înainte de parcurgerea codului rămas.

În funcția *setup()* (liniile 71-116) se inițializează și se pregătesc diferite variabile sau componente: se începe transmiterea serială, se atașează motoarele servo la pinii corespunzători, se setează poziția de început a motoarelor, se setează pinii pentru joystick, se setează pinul pentru întrerupător, se setează pinii pentru senzorul de culoare, se setează frecvența de selecție a senzorului la 20%, se inițializează și setează o serie de variabile necesare instanțierii obiectului *sevseg*.

În funcția *loop()* (118-250) se află codul propriu-zis pentru funcționarea jocului. Astfel: se începe prin a verifica dacă jocul este pornit (linia 121). În cazul în care este oprit, se parcurge secvența de cod (prezentă la liniile 241-249): se printează pe monitorul serial un mesaj cu privire la starea jocului, se setează variabila de cântat pe 0, se oprește afișarea pe afișor și se resetează scorul. În cazul în care jocul este pornit, se verifică dacă buzzerul a mai fost activat (linia 121). În caz negativ, se activează și apoi se setează indicatorul pe 1, iar în caz pozitiv, se sare peste acea secvență de instrucțiuni. Următorul pas este verificarea dacă s-a apăsător butonul de pe joystick (linia 127). În caz afirmativ, se pune un delay sub forma de for loop (din cauză ca biblioteca utilizată pentru afișare nu permite utilizarea întârzierilor, așa că în această buclă se reîmprospătează informația afișată), apoi se setează combinația pentru detectarea frecvenței roșu, urmată de citirea acesteia și încă o întârziere. Se repetă procesul pentru toate 3 culorile, după care se trimit informațiile obținute spre monitorul serial (liniile 150-155). Apoi începe verificarea datelor obținute pentru interpretarea acestora, astfel încât se verifică dacă culoarea detectată corespunde frecvenței roșu sau albastru (liniile 158-179). În oricare din cazuri, se incrementează scorul pentru culoarea detectată și se activează cântecul corespunzător, iar în caz că s-a depășit pragul de 5 puncte, se resetează tabelul. Apoi urmează citirea informațiilor cu privire la poziția joystick-ului (liniile 182-184). Folosind acestea, se mapează informațiile la un interval mai mic, simetric față de origine, după care se trimit valorile spre monitorul serial. Apoi urmează modificarea pozițiilor servomotoarelor în funcție de valorile mapate (liniile 200-233) astfel: se compară cu referința -1 (deoarece intervalul în care s-a mapat este considerabil mai mic decât cel inițial, și în acest interval, mijlocul celui mare se mapează la valoarea -1, nu 0) atât x cât și y și se procedează astfel: în cazul în care valoarea mapată este



mai mare decât referința: se crește poziția motorul servo cu un grad, se trimite informația la componentă și apoi se realizează o întârziere print-un for loop. În caz contrar, se scade poziția motorul servo cu un grad, se trimite informația la componentă și apoi se realizează din nou o întârziere printr-un for loop. La finalul funcției loop() se mai realizează o întârziere, după care ciclul se reia.

## VI. Instrucțiuni de utilizare

Jocul se constituie dintr-un labirint ale cărui înclinații sunt controlate de un joystick. Acesta mai prezintă și două bile de culori diferite: albastru și roșu, așezate în două colțuri diferite ale labirintului. De asemenea, există și un afișor pe care se va nota scorul și un întrerupător care va marca începerea acestuia. Scopul jocului este aducerea bilei albastre în coșulețul cu senzorul de culoare, prin gaura făcută într-un colț al labirintului, iar jocul se poate pierde în cazul în care ajunge bila de culoare roșie în coșuleț.

Stare inițială: cele două bile se află în două colțuri diferite ale labirintului și întrerupătorul este oprit.

Utilizare: începerea jocului se semnalează prin pornirea întrerupătorului. La pornire se va auzi un cântec generat de buzzer și se va aprinde afișorul. Pe afișor se va putea vedea mesajul: b0r0. Acest mesaj transmite faptul că bila de culoare albastră (b) are scorul 0 și bila de culoare roșie (r) are tot scorul 0. Apoi se poate mișca joystickul și se poate observa că se înclină labirintul conform acestuia. Se aduce bila albastră la gaura din labirint după care va cădea în coșuleț. În momentul în care o bilă se află acolo, se poate apăsa butonul pus la dispoziție de joystick care semnalează începerea detectării culorii de către senzorul de culoare. În cazul în care culoarea detectată este albastră, se adaugă un punct pe afișor pentru “b” și se pornește un sunet de marcare a câștigului. În caz contrar, se adaugă un punct pe afișor pentru “r” și se pornește un sunet de marcare a pierderii. După ce una dintre bile ajunge la scorul 5, afișorul se resetează și se poate reîncepe jocul. La terminarea jocului se închide întrerupătorul.

## VII. Bibliografie

[1] Mureșan Mircea Paul, Danescu Radu, Itu Răzvan, “Proiectare cu Microprocesoare – Îndrumător de laborator”, Cluj-Napoca, 2018;

[2] Atmega 2560 datasheet, website: [https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\\_datasheet.pdf](https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf)

[3] Arduino color detection, website: <https://create.arduino.cc/projecthub/SurtrTech/color-detection-using-tcs3200-230-84a663>

[4] SevSeg Arduino library, website: <https://www.arduino.cc/reference/en/libraries/sevseg/>

[5] *Arduino Marble Maze* – for design, website:

<https://create.arduino.cc/projecthub/AhmedAzouz/arduino-marble-maze-labyrinth-bd9ea6>