

# Pong game

Nume: Pelin Raluca

Grupa: 332CC

## Introducere

Proiectul are ca scop realizarea unui joc de pong, în cadrul căruia doi jucători pot concura unul împotriva altuia pentru a da cât mai multe goluri adversarului. Există și modul de joc prin care un singur jucător concurează împotriva controller-ului.

## Descriere generală

- terenul este construit din placi de lemn de traforaj cu grosimea de 4mm, peretii terenului fiind fixati cu ajutorul unor coltare modelate si printate 3D
- crosa ce loveste mingea este fixata pe o piesa modelata si printata 3D, ce se deplaseaza stanga-dreapta prin actionarea unui motor DC, care este angrenat printr-un scripete la o sfoara pe care este fixata piesa cu crosa
- pentru a asigura o deplasare in linie dreapta, piesa este fixata si pe doua bare paralele de aluminiu
- la fiecare capăt al terenului se află butoanele prin care jucătorul controlează direcția crosei de pong, precum și butonul prin care jucătorul lovește mingea de pong
- crosele se pot deplasa doar stânga-dreapta de-a lungul marginii terenului

## Schema bloc:



## Mod de functionare:

În ceea ce privește modul de joc împotriva controller-ului, este necesar că poziția mingii pe teren să fie cunoscută de către acesta, pentru a putea muta crosa la o poziție potrivită. Pentru aceasta, sunt folosiți 3 senzori de distanță IR cu output analog cu ajutorul cărora se poate calcula poziția mingii pe teren, această informație fiind transmisă către Arduino, care apoi acționează motorul DC astfel încât crosa se va alinia cu poziția mingii când ajunge la capătul sau al porții.

În cazul modului de joc între doi jucători, controller-ul așteaptă input de la fiecare jucător pentru a putea acționa cele două motoare DC ce mută crosa de-a lungul lățimii terenului. Jucătorul interacționează prin apăsarea unor butoane, astfel mutând crosa spre stânga sau dreapta. În momentul în care jucătorul dorește să lovească mingea, acesta va trebui să apese alt buton ce va acționa servomotorul care împinge mingea spre poarta oponentului.

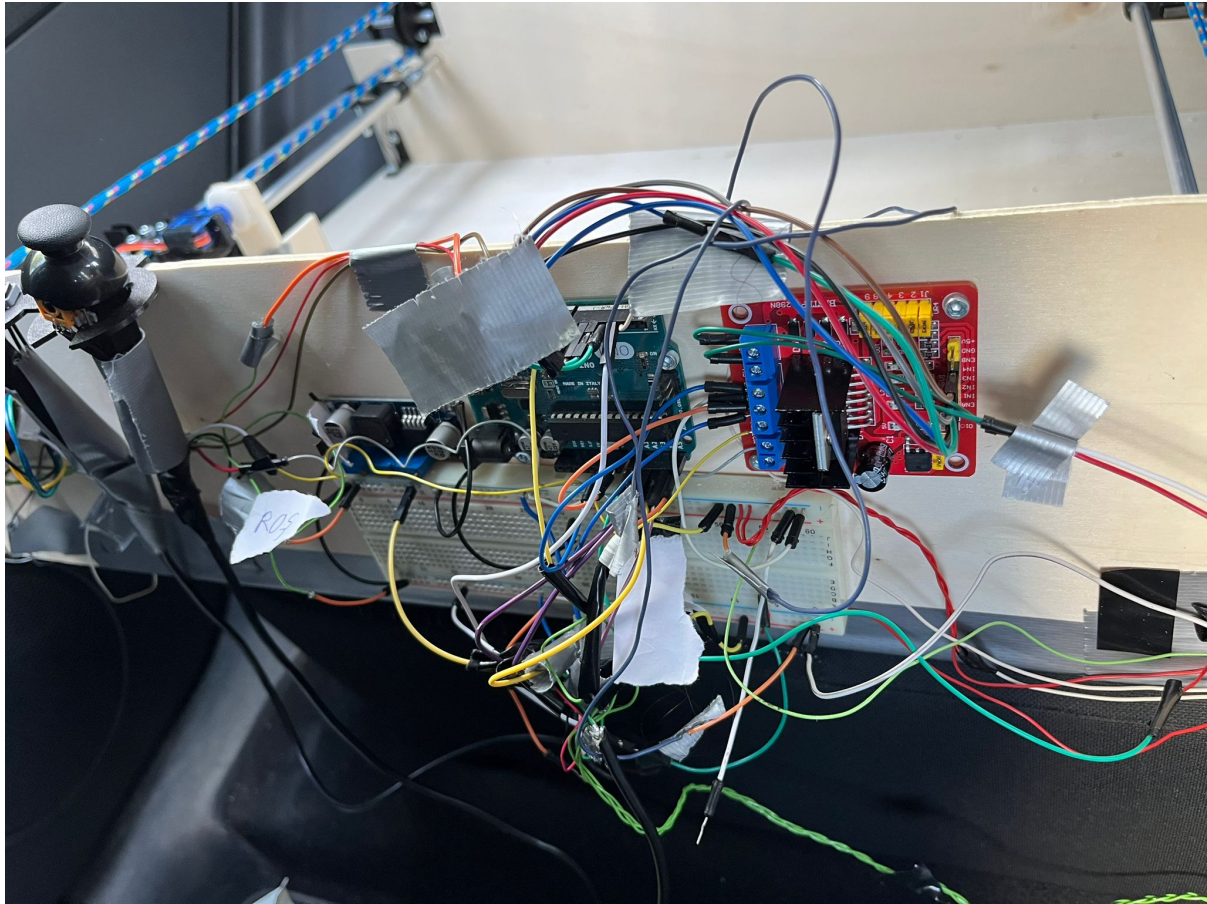
## Hardware Design

### Piese necesare:

Piesa	Cantitate
Arduino UNO	1
motor JGA25-370 cu reductor și encoder	2
servomotor SG90	2
senzor IR distanță Sharp	2
sursă de tensiune 12V	1
L298N driver dual motoare	1
Modul coborâre tensiune LM2596	1
Condensator 100uF	2
Limitator cu lamela	4
holder motoare DC	2
placi lemn traforaj	2

### Schema electrică:





## Modelare și printare 3D

Pentru a putea realiza jocul de pong în modul în care mi-am propus, am folosit piese printate 3D pentru a obține rapid elemente customizate pentru nevoile proiectului, și anume holdere pentru motoare, servomotoare și senzori, scripeți, holder pentru croșele din joc, holder pentru joystick-uri.

Astfel, am modelat în AutoDesk Fusion 360 piesele menționate anterior și apoi le-am printat la o imprimantă 3D Ender Creality, folosind programul de slicing PrusaSlicer. Alegerea de a printa 3D aceste componente ale proiectului s-a dovedit foarte utilă, întrucât timpul petrecut modelând un obiect de forma și dimensiunile dorite, precum și printarea acestuia, au durat mult mai puțin decât ar fi durat improvizarea unei piese similare din alte materiale.

Inițial, planul era ca cele două servomotoare să se alimenteze la Arduino UNO, iar cele două motoare DC de la sursa de tensiune de 12V. Din păcate trebuiau efectuate niște calcule legate de urmările unui asemenea format de cablare, întrucât a dus la arderea plăcii Arduino și a driver-ului de motoare. Soluția pentru această situație este utilizarea unui modul de coborâre tensiune LM2596, care convertește tensiunea de 12V primită de la sursa de tensiune într-o tensiune de 5V, care va fi folosită pentru alimentare servomotoarelor. Astfel, nu mai există componente de mare consum care să se alimenteze de la placa Arduino, aceste alimentându-se de la sursa de tensiune de 12V.

## Software Design

Mediu de dezvoltare: Arduino IDE

Biblioteci utilizate: Servo.h

## Întreruperi

Pentru a preveni ca un player să distrugă integritatea structurală a mesei de joc prin acționarea necorespunzătoare a crosei de joc, au fost utilizate limitatoare cu lamelă în capetele axelor pe care se deplasează crosa.

Cu ajutorul întreruperilor hardware, dacă un player acționează motorul prea mult timp într-o anumită direcție, iar crosa se lovește de unul dintre limitatoare, motorul se va opri instant și nu se va mai putea deplasa decât în direcția opusă capătului de care s-a lovit.

```
void setup(){
  attachInterrupt(digitalPinToInterrupt(swPin1), stop_motor_right1, CHANGE);
  // rest of setup content
}
```

Întrucât întreruperile de tipul *attachInterrupt* sunt disponibile doar pe pinii 2 și 3 pe Arduino UNO, doar 2 dintre cele 4 limitatoare folosesc acest tip de întreruperi, iar pentru celelalte două limitatoare se vor actualiza stările aferente acestora în cadrul funcției de loop. Se puteau utiliza și întreruperi de tipul PCINT, însă implementarea curentă este mai rapidă și atinge același scop.

```
void loop(){
  stop_left_2 = !digitalRead(swPin3);
  // rest of loop content
}
```

## PID

Pentru a putea ajunge cu crosa controlată de Arduino la distanța dorită, am folosit un algoritm simplu de PID. La un capăt al terenului am plasat un senzor IR, iar pe holder-ul unde se află servomotorul a fost plasată o plăcuță de lemn ca referință pentru senzorul IR. După o serie de calibrări, am constatat că algoritmul de PID plasează crosa la distanța dorită doar prin utilizarea proporționalei și a integralei, fără a fi nevoie de diferențială, valorile acestor constante fiind determinate printr-un proces de trial and error.

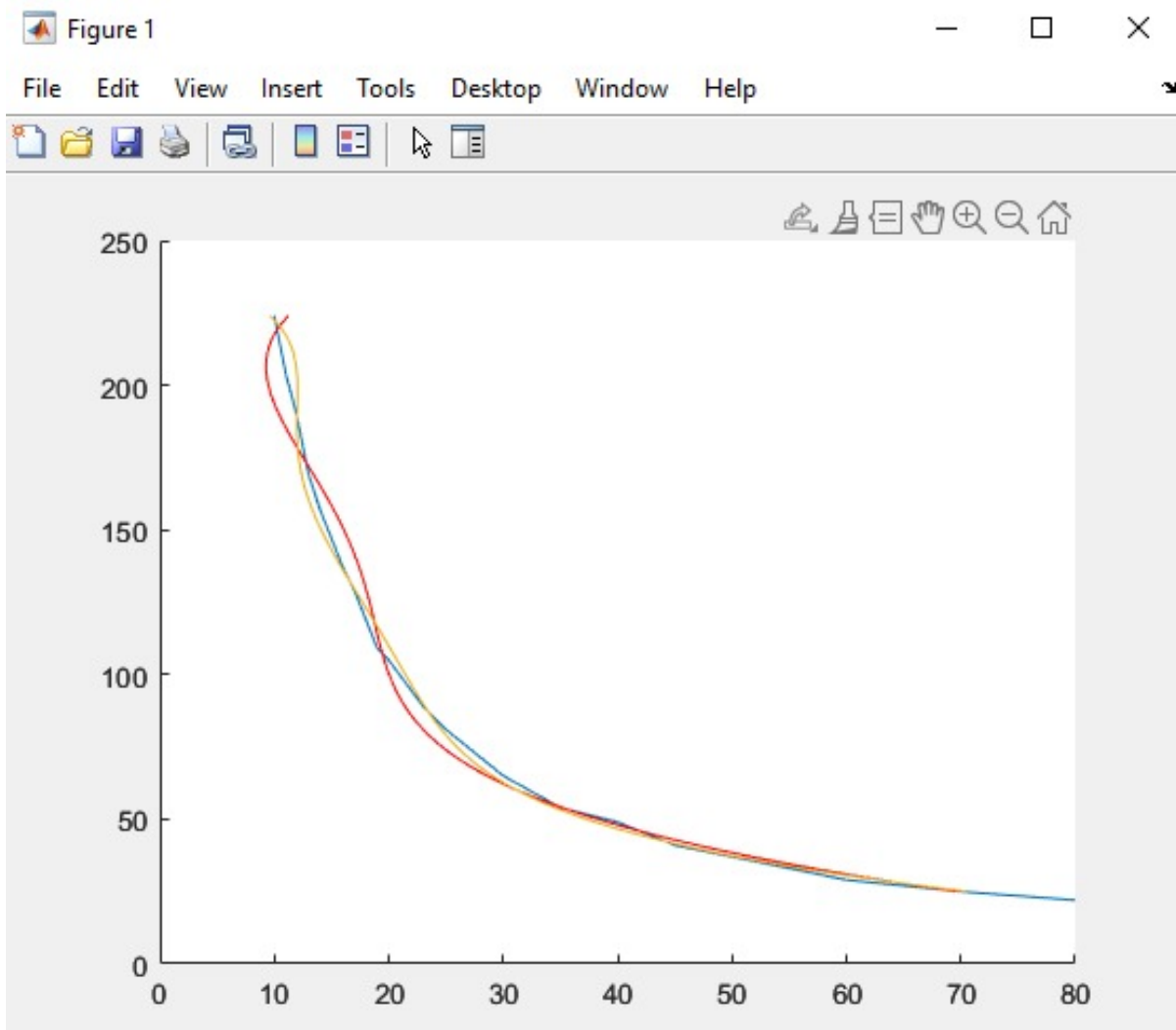
Astfel, implementarea de bază a algoritmului stă la baza acestor formule:

```
double input = getSensorInput(A0);
double err = ref - input;
sum += err * dt;
double out = err * Kp + sum * Ki;
```

## Funcție de transfer senzor IR

Pentru măsurarea distanței în cadrul jocului sunt folosiți doi senzori IR Sharp GP2Y0A41SK0F. Acest senzor este unul de tip analog, însemnând că output-ul său reprezintă un voltaj (un număr între 0 și 1024, de fapt) ce poate fi apoi mapat cu distanța măsurată de senzor.

Deși pe Internet există resurse și exemple de cod pentru măsurarea distanței cu ajutorul acestor senzori, prin compararea output-ului de la senzor și măsurarea distanței reale am constatat că funcția de transfer din exemplele existente nu se aplică pe senzorii mei și trebuie să implementez eu funcția bazându-mă pe output-ul de la senzori.



Astfel, am realizat măsurători pentru aproximativ 30 de distanțe din range-ul [10, 80] cm, acesta fiind range-ul de funcționare normală al senzorului. Aceste informații le-am pasat apoi într-un script MatLab, în cadrul căruia am folosit funcția *polyfit()* pentru a obține cel mai apropiat polinom care să reprezinte adecvat output-ul senzorilor. Am ales să folosesc un polinom de gradul 5, acesta fiind cel mai complex polinom ce se putea obține în MatLab pornind de la datele furnizate din măsurătorile mele, ajungând la următoarea implementare a funcției de transfer pentru senzor:

```
float getDistanceA41(int x){  
    return -2.8883e-09 * pow(x,5) + 2.0301e-06*pow(x,4) - 5.4998e-04 * pow(x,3)  
+ 0.0723 * pow(x,2) - 4.7854*x + 152.759;  
}
```

}

## Activități simultane

Întrucât flow-ul unui joc de pong implică petrecerea în același timp a mai multe activități - input de la un jucător și de la celălalt, acționarea servomotoarelor, acționarea motoarelor DC, a trebuit implementată o soluție care să gestioneze aceasta problemă fără să interfereze cu experiența jucătorilor.

Pentru aceasta, a trebuit să renunț la utilizarea apelurilor funcției `delay()` în cadrul programului, cum ar fi în momentul acționării servomotoarelor, pentru a nu împiedica rularea altor componente ale jocului, cum ar fi motoarele DC. În schimb, am simulat funcționalitatea unui timer cu ajutorul funcției `millis()` în felul următor:

```
current_time = millis();
if (current_time - start_time >= THRESHOLD){
    do_action = 0;
}
if (do_action) {
    start_time = millis();
    // do stuff until threshold passes
}
//do other stuff as usual
```

Dacă se dorește utilizarea bibliotecii `Servo.h` pentru controlarea servomotoarelor, trebuie luat în considerare faptul că această bibliotecă dezactivează funcționalitatea PWM pentru pinii 9 și 10 din Arduino UNO.

În cazul configurației mele, pinii de enable pentru motoarele DC erau chiar pinii 9 și 10, motiv pentru care acestea nu mai funcționau odată ce am introdus servomotoarele și a fost nevoie de debugging. După identificarea acestei probleme, am mutat pinii de enable pentru motoarele DC încât toate componentele funcționează corespunzător.

## Detectarea poziției mingii

Pentru a detecta poziția mingii pe teren, se utilizează un senzor IR plasat aproape la jumătatea terenului. Senzorul măsoară distanța mingii față de acesta atunci când mingea trece prin fața sa, apoi putând determina pe ce poziție ar trebui să se afle crosa de joc pentru a putea pasa mingea mai departe, cros fiind deplasată la poziția dorită folosind algoritmul de PID descris mai sus.

Această strategie ar fi funcționat dacă senzorul utilizat nu ar fi avut un noise foarte sporit, însă, datorită spike-urilor ce apar în urma citirii valorii senzorului, nu se poate determina o poziție la care sa trebuiască mutată crosa, motiv pentru care am renunțat la input-ul din partea acestui senzor și am ales să ofer crosei o poziție random din 4 posibile poziții astfel încât să fie acoperită toată distanța porții din joc, fiecare poziție acoperind un cadran din poartă, poarta având în total 4 cadrane.



[2players\\_final.zip](#)

## Jurnal

- **23 Aprilie** - Realizare prototip masă de joc din carton, modelare și printare scripeți, suport crosă și holdere motoare DC
- **25 Aprilie** - Realizare funcție transfer senzor IR + testare
- **29-30 Aprilie** - Realizare masă de joc din lemn de traforaj + montare bare aluminiu
- **7-8 Mai** - Montare motor DC pe o parte a mesei de joc + sfoara scripeți, testare mișcare liniară
- **18 Mai** - Realizare mecanism mișcare liniară și pe partea opusă a mesei de joc + printare holdere servomotoare
- **19 Mai** - Montare servomotor pe suport crosă + montare switch-uri la capetele barelor de aluminiu
- **20 Mai** - Printare holder senzor IR + montare senzor, implementare și testare PID motor DC
- **21 Mai** - Montare servomotor simetric, s-a ars placa Arduino + driver motoare
- **22 Mai** - Înlocuire Arduino UNO și modul L298N, refacere cablaj + adăugare convertor 12V-5V
- **24 Mai** - Retestare sistem, lipire fire pentru joystick-uri
- **25 Mai** - Implementare flow pentru joc între 2 playeri

## Bibliografie/Resurse

1. [https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk\\_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf)
2. <https://docs.arduino.cc/learn/electronics/servo-motors>
3. <https://forum.arduino.cc/t/demonstration-code-for-several-things-at-the-same-time/217158>

[Export to PDF](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2023/alucaci/pong-game>



Last update: **2023/05/30 09:08**