

# Robotic Arm

**Student:** *Ulariu Corina-Elena*

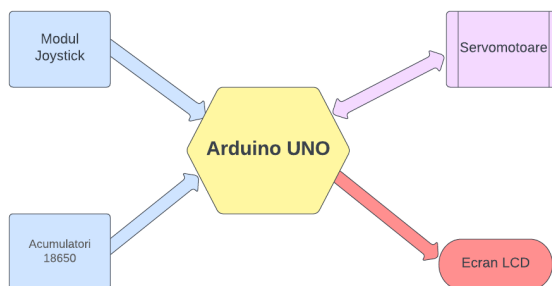
**Grupa:** 331CC

## Introducere

Ne aflăm într-o lume în care roboții sunt din ce în ce mai folosiți pentru simplificarea unor task-uri lungi sau complicate. Așadar, am decis că pentru proiectul meu vreau să construiesc un braț robotic care poate fi considerat punct de pornire pentru mulți dintre roboții prezenți în diferite industrii.

Brațul va ridica și muta obiecte de dimensiuni mici, afișând un mesaj corespunzător dacă nu reușește să apuce sau ridice obiectul dorit.

## Descriere generală



Prin intermediul unuia dintre cele două module joystick utilizatorul poate controla direcția în care se mișcă brațul robotic. O dată ce am adus brațul în poziția dorită, vom folosi celălalt joystick pentru a modifica deschiderea cleștelui în așa fel încât vom putea apuca sau lăsa un obiect.

Pentru mișcarea brațului ne vom folosi de patru servomotoare. Dacă obiectul este prea mare sau prea greu, așadar servomotoarele nu se vor putea mișca, vom afișa un mesaj corespunzător pe ecranul LCD.

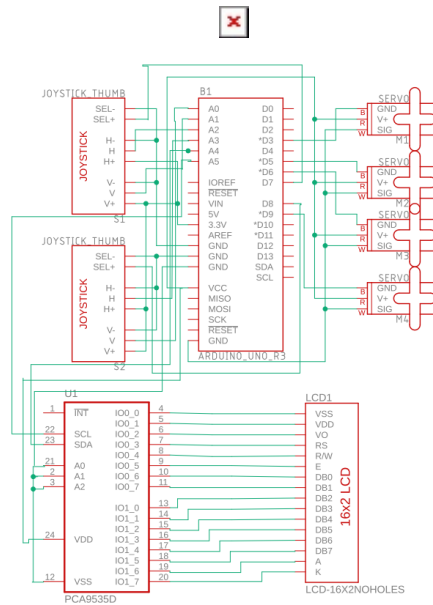
Brațul robotic este alimentat prin intermediul a doi acumulatori. }

## Hardware Design

Lista de piese:

- Arduino UNO R3
- Servo motor MG90S x 4
- Sensor Shield Arduino v5
- Ecran LCD
- Modul Joystick x 2
- Acumulatori 18650 x 2

Schema electrica:



# Software Design

Mediul de dezvoltare folosit pentru realizarea proiectului a fost Arduino IDE. Librăriile folosite au fost:

- **Servo.h** - utilizată pentru controlarea celor 4 servomotoare
- **LiquidCrystal\_I2C.h** - utilizată pentru scrierea pe ecranul LCD folosind I2C

Scurtă descriere a codului implementat:

**Variabile folosite în implementarea codului:**

*LCD*

```
LiquidCrystal_I2C mylcd(0x27, 16, 2);
```

*Servomotoare*

```
Servo myservol;
```

```
Servo myservo2;  
Servo myservo3;  
Servo myservo4;
```

*variabile care rețin pozițiile servomotoarelor*

```
int pos1=90, pos2=90, pos3=90, pos4=90;
```

*variabile care rețin poziția anterioară a servomotoarelor*

```
int prev_pos1=90, prev_pos2=90, prev_pos3=90, prev_pos4=90;
```

*variabile care rețin mișcăriile joystick-urilor*

```
int x1,y1,z1;  
int x2,y2,z2;
```

*variabile care rețin ultima mișcare făcută de joystick-uri*

```
int last_x1, last_x2, last_y1, last_y2;
```

Pentru a executa principalele mișcări ale brațului robotic, am împărțit implementarea în 4 funcții, care se ocupă fiecare de modificarea poziției unui servomotor.

Funcția **claw()** se ocupă de cel de-al patrulea servomotor, care va deschide și închide cleștele de la capătul brațului robotic, în funcție de mișcările pe orizontală ale joystick-ului din stânga. Dacă joystick-ul este împins în stânga, cleștele se va închide, iar dacă este împins în dreapta se deschide. Pentru a se asigura că cleștele nu se va deschide sau închide mai mult decât poate, se limitează unghiul de rotire al servomotorului.

Funcția **turn()** va controla primul servomotor, transmițându-i acestuia unghiul la a cărui poziție va trebui să ajungă pentru a roti corpul brațului robotic, în funcție de mișcările pe orizontală ale joystick-ului din dreapta. Brațul se va roti în direcția în care este împins joystick-ul. Pentru a se asigura că rotația este limitată, unghiul de rotire al servomotorului va fi în intervalul [1, 180].

Funcția **upper\_arm()** va determina mișcarea de coborâre și urcare a brațului, fiind controlată prin cel de-al treilea servomotor, mișcările căruia vor fi preluate din citirea mișcării pe verticală a modulului joystick din stânga. Dacă joystick-ul este împins în jos, brațul coboară, urcând dacă e împins în sus.

Funcția **lower\_arm()** stabilește mișcarea de întindere și retragere a brațului, fiind controlată prin cel de-al doilea servomotor, mișcările căruia vor fi preluate din citirea mișcării pe verticală a modulului joystick din dreapta. Dacă joystick-ul este împins în jos, brațul se va trage înapoi, întinzându-se dacă e împins în sus.

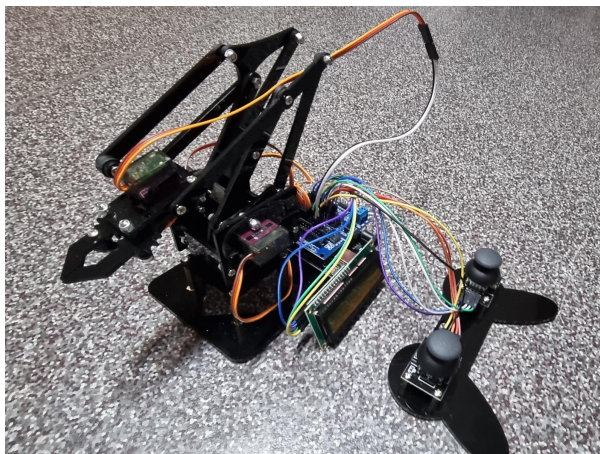
În funcția de **setup()** se setează lcd-ul și se aduce robotul în poziția inițială, afișându-se pe ecranul LCD un mesaj de început.

În funcția de **loop()** se apelează funcțiile de mișcare ale brațului, și se afișează mesaje corespunzătoare dacă mișcăriile nu pot avea loc.

Cod proiect: [robotic\\_arm.zip](#)

## Rezultate Obținute

- Rezultate obținute în urma ansamblării componentelor hardware și a scheletului brațului robotic



## Concluzii

Realizarea acestui proiect m-a făcut să realizez câtă muncă se află în spatele oricărui mic obiect care are în spatele lui un circuit electric.

Cea mai grea parte a reprezentat-o cel mai probabil asamblarea scheletului și legarea firelor, deoarece a necesitat o foarte mare atenție, dar și scrierea codului a fost destul de complicată întrucât nefiind foarte familiarizată cu librăriile am întâmpinat câteva bug-uri.

Mi-ar plăcea să lucrez și la alte proiecte mai complicate, dar consider că este o muncă destul de

dificilă atât proiectarea, cât și asamblarea unui asemenea proiect.

## Download

Arhiva conține:

- codul sursă
- schema bloc
- diagrama electrică
- imagini cu brațul robotic

[331cc\\_ulariu\\_corina-elena.zip](#)

## Jurnal

- 07.04.2023 - alegere proiect
- 13.04.2023 - comandare piese
- 05.05.2023 - scriere documentație
- 18.05.2023 - finalizarea părții hardware
- 23.05.2023 - finalizarea părții software
- 24.05.2023 - finalizarea wiki

## Bibliografie/Resurse

1. Resurse Hardware:
  - <https://www.hackster.io/Hack-star-Arduino/learn-to-use-lcd-1602-i2c-parallel-with-arduino-uno-f73f07>
  - <https://microbotlabs.com/arduino-joystick-servo-motor-robot-arm-control-tutorial.html>
2. Resurse Software:
  - <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2022>
  - <https://ocw.cs.pub.ro/courses/pm/lab/lab5-2022>
  - <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2022>
  - [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
  - <https://reference.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/ncaroi/robotic-arm>



Last update: **2023/05/29 16:20**