



**UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA**

4DOF Robotic arm

Proiectare cu microprocesoare

Student:

Cimpeanu Andrei-Vlad

Grupa 30236

Profesor indrumator:

Mircea Paul Muresan

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

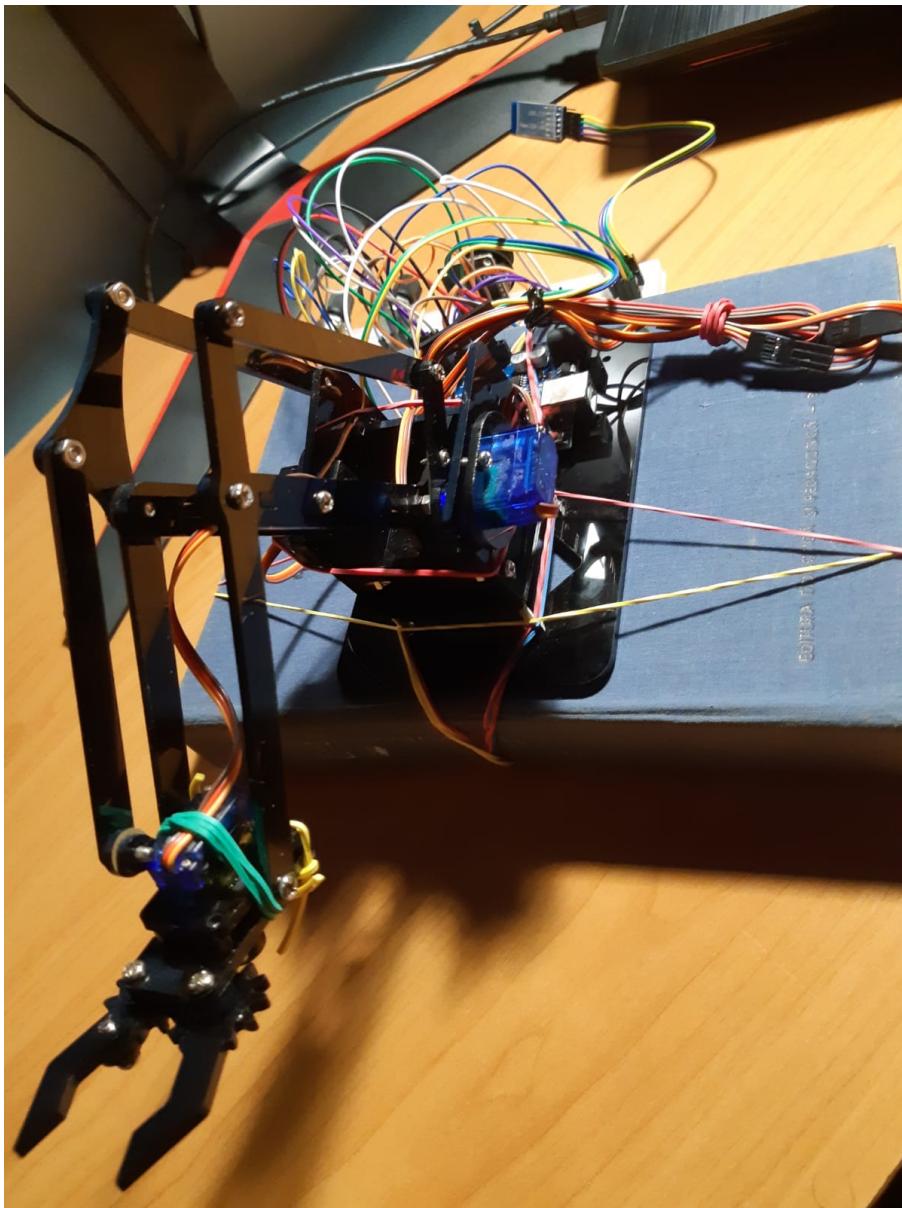
January 10, 2021

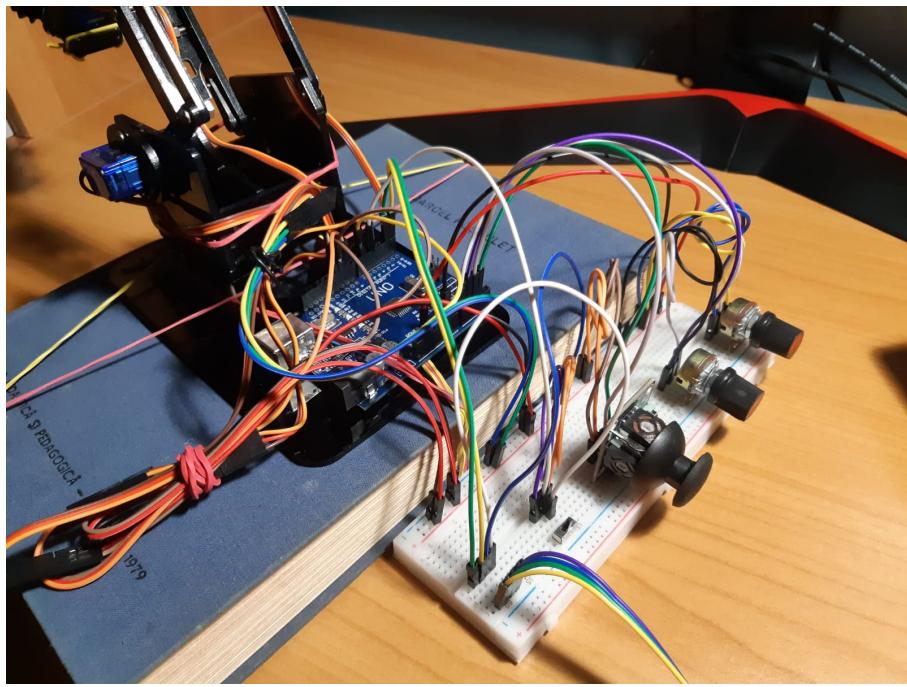
Contents

1 Prezentarea proiectului ales	1
2 Componente necesare si montaj	3
3 Functionalitati si utilizare	4
4 Detalii de implementare	5
4.1 Implementare Arduino	5
4.2 Implementare Aplicatie Android	6
5 Concluzii si dezvoltari ulterioare	6
6 Bibliografie	6

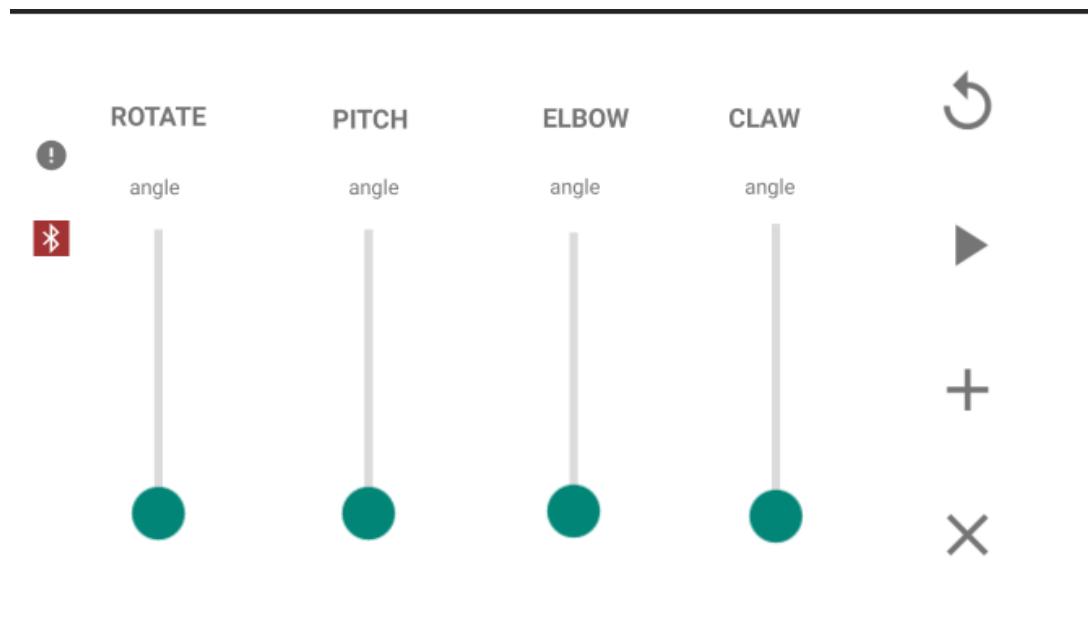
1 Prezentarea proiectului ales

Proiectul presupune implementarea unui brat robotic cu 4 axe de rotatie(de aici si 4DOF - 4 Degrees of freedom). Bratul este conceput pe baza unei placi Arduino UNO si este realizat in asa fel incat sa poata fi controlat fie cu ajutorul potentiometrelor si al unui joystick, fie prin intermediul unei aplicatii Android realizata cu scopul acestui proiect, pe baza conexiunii Bluetooth. Bratul are o structura de plastic acrilic, pentru a mentine o greutate cat mai redusa a componentelor. Miscarea acestuia se face cu 4 servomotoare micro, putand astfel sa fie rotit, inclinat, si sa prinda obiecte cu o gheara careia i se poate regla independent miscarea si deschiderea acesteia. Tranzitia de la un mod de control la altul se face printr-un switch, initial fiind setat pentru controlul fizic, putand fi schimbat pentru a fi controlat prin telefon.





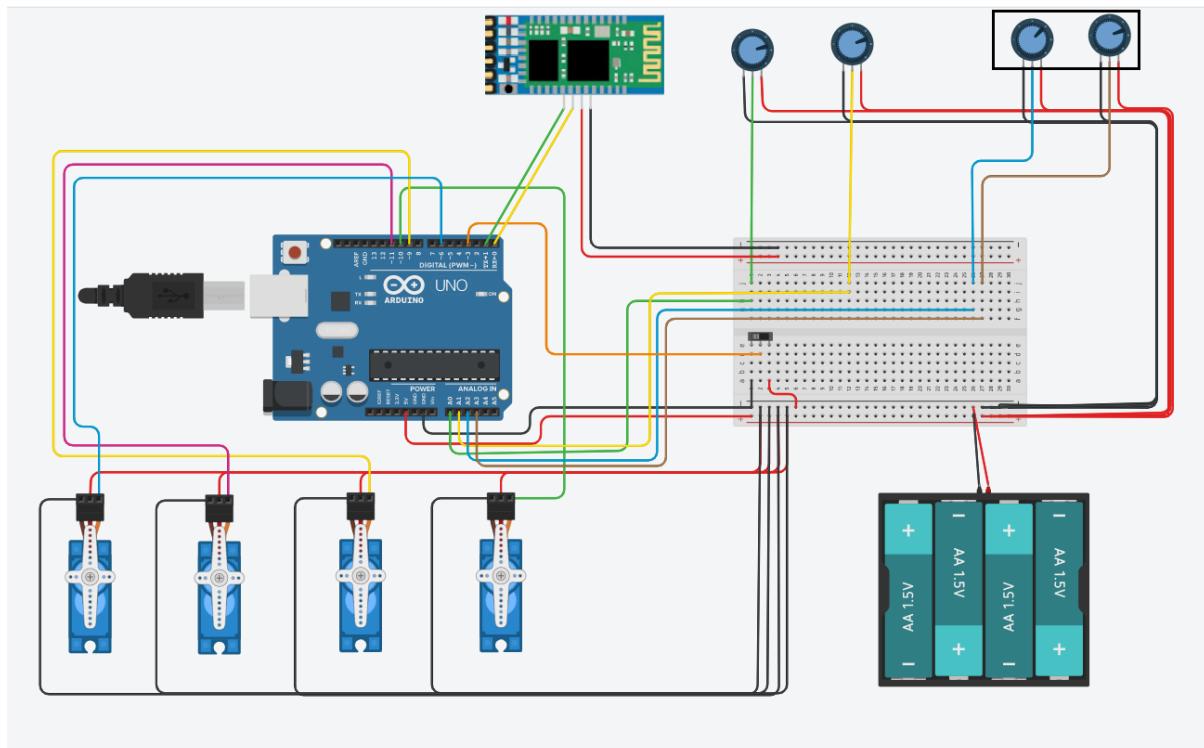
Aplicatia de telefon a fost implementata in Android Studio si a fost gandita strict pentru acest proiect, dar cu optiuni multiple de extindere a functionalitatilor. Designul folosit este unul simplu, avand doar cele 4 slide pentru controlul servo-urilor, butonul de conectare prin Bluetooth, cel de instructiuni de folosire(info) si butoanele laterale pentru setarile automatizarii.



2 Componente necesare si montaj

Pentru implementarea acestui proiect au fost folosite urmatoarele componente, in modul in care au fost legate in schema de mai jos. Deoarece pe tinkercad.com[1] nu erau disponibile toate componentele necesare, joystick-ul a fost reprezentat printr-un chenar cu 2 potentiometre si modulul Bluetooth are doar cei 4 pini utilizati. Pentru legarea servomotorilor am folosit 4 pini digitali de tip PWM, cu Ground si VCC legate la cele 4 baterii asezate in serie, pentru extinderea tensiunii. Potentiometrele si Joystick-ul au fost legate la pinii analogici, pentru a putea citi valorile distincte din spectrul disponibil la care le reglam, Ground-ul si VCC-ul fiind legate la placa, prin breadboard. Butonul de selectie al modului de operare este legat la un pin digital si are nevoie de sursele de curent pentru valorile de 0 si de 1 logic. Modulul Bluetooth a fost alimentat de pe breadboard, iar pinul RX al acestuia a fost legat la pinul TX al placii si invers, pentru transmiterea de date in mod serial intre ele.

- 3 servomotoare SG90, 1 servomotor MG90S
- 2 potentiometre 10 kOhm
- 1 joystick
- 1 placă Arduino UNO
- 1 breadboard cu 830 de puncte
- 1 switch
- 1 modul Bluetooth HC-05
- fire mama-mama si tata-tata
- 4 baterii AA + suport
- kit din plastic acrilic pentru brat robotic



3 Functionalitati si utilizare

Pentru explicarea functionalitatilor si utilizarea acestora, voi incepe cu cele legate de controlul fizic al bratului robotoc. Fiecare din cele 4 servomotoare va controla bratul pe cate o axa, respectiv va avea un rol, precum clestele. Joystick-ul controleaza 2 axe (X si Y), ele putand fi controlate simultan prin miscarea intre acestea. Folosind joystick-ul, vom putea controla rotatia bratului (axa X) si inclinarea bratului (pe axa Y). Pentru a inchide sau deschide clestele vom folosi potentiometrul din mijloc, iar pentru controlul inalitimii clestelui si ajustari mai precise pe axa Y vom folosi potentiometrul din marginea breadboard-ului. Unghiurile servomotoarelor au fost limitate, pentru a nu bloca bratul in zone care nu vor fi folosite, sau care ar misca gresit structura de plastic, astfel creand o piedica. Unghiul de rotatie al bratului a fost inversat, deoarece este mai natural sa se respecte o linie dreapta de la pozitia joystick-ului spre pozitia curenta a sistemului - schimbarea se poate face din cod, de la mapare.

Pentru a face trecerea de la controlul fizic analog la cel de pe telefon, se va folosi switch-ul, care initial este setat pe 1 (respecta directia aranjarii pe breadboard a componentelor). Acest lucru va debloca transmisia dintre modulul Bluetooth si aplicatia creata. Pentru conectarea la modul, va fi nevoie de imperecherea modulului cu telefonul. Acest lucru se va face tinand apasat pe butonul cu semnul Bluetooth, care este rosu daca este deconectat. Dupa ce se accepta permisiunea de pornire a Bluetooth-ului si de conectare, va aparea un Toast care va anunta succesul conectarii. Odata conectat, telefonul va transmite

semnale cand se ridica degetul de pe fiecare slider, putand vedea live schimbarea valorii unghiului selectat. In cazul in care se doreste memorarea pozitiei curente, se apasa pe butonul "+" pentru a o adauga in lista, iar dupa ce a fost creata o serie de pozitii, se apasa butonul de Start (simbolul "Play" din aplicatie) si vor fi efectuate automatizarile, miscarile repetandu-se pana cand se apasa din nou pe butonul de Start. Butonul cu semnul "Replay" va reseta bratul in pozitia initiala si va actualiza sliderele si unghiul afisat in cazul in care automatizarea este oprita, iar in caz contrar va afisa un mesaj pentru atentionare. Daca butonul "Clear" este tinut apasat, pozitiile memorate pentru procesul de automatizare vor fi sterse si nu sa va putea porni un nou proces de automatizare fara adaugarea unor pozitii noi.

4 Detalii de implementare

In continuare va fi prezentata implementarea codului, atat al partii de Arduino, cat si cea a aplicatiei Android. Mediile de dezvoltare folosite sunt Arduino IDE si Visual Studio Code cu extensii pentru fisiere .ino pentru compilare, adaugarea codului pe placa si monitorizare in Serial Monitor, respectiv pentru editarea acestuia si Android Studio pentru realizarea aplicatiei de telefon, codul fiind scris in Java. Pentru documentarea legata de cod si metode de implementare am folosit paginile oficiale de documentatie Arduino[2] si Android Developers[3].

4.1 Implementare Arduino

Codul Arduino este relativ simplu, continand instantierile componentelor, legarea la pinii destinati lor si miscarea servomotoarelor prin scrierea valorilor obtinute. Sectiunea de setup cuprinde setarea Baud Rate-ului la 9600 pentru caile seriale, neavand nevoie de setari suplimentare pentru modulul Bluetooth, atasarea servo-urilor la pinii de PWM declarati in antet si pozitionarea acestora in pozitiile initiale, care vor ramane neschimilate daca in urmatoarea sectiune, loop, nu vor primi alte valori. In sectiunea loop prima data se verifica daca toate servomotoarele au fost conectate cu succes: in cazul in care nu au fost va aparea un mesaj in Serial Monitor, iar daca toate sunt conectate, vor putea fi modificate pozitiile acestora, in functie de modalitatea selectata. Daca este aleasa metoda de control prin Bluetooth se asteapta transmiterea datelor si maparea acestora la servomotorul corespunzator, conform cifrei miilor din fiecare valoarea primita, restul reprezentand unghiul dorit. Cealalta metoda de control va citi de pe pinii analogici valorile(intre 0 si 1024) si le va mapa proportional cu unghiul setat pentru fiecare motor. Indiferent de modalitatea aleasa, cate o valoare va fi mereu scrisa pe servo-uri, chiar daca aceasta este pozitia initiala, nemodificata.

Pentru flexibilitatea codului au fost scrise functii ajutatoare: una dintre ele va scrie toate cele 4 valori calculate pe servouri, iar cealalta va verifica daca toate servomotoarele sunt conectate.

4.2 Implementare Aplicatie Android

Aceasta parte a proiectului are 2 parti de implementare: una este cea de design grafic a aplicatiei, folosind XML pentru adaugarea componentelor si aranjarea acestora cu ajutorul codului si cu editorul grafic pus la dispozitie, iar cea de a doua este implementarea de back-end a aplicatiei. Pentru partea de functionare am folosit threadul principal al aplicatiei cu metoda `onCreate`, in care am adaugate cate un listener pentru fiecare element vizual: sliderele vor actualiza pozitiile fizice ale bratului si valorile in grade scrise deasupra acestora, butonul de Bluetooth va cere permisiuni de pornire a adaptorului si asocierea acestuia cu modulul legat la placa Arduino, iar celelalte butoane vor controla partea de automatizare a proiectului. Unele dintre butoane au fost programate in asa fel incat sa actioneze doar la apasare, iar altele, pentru a evita apasarea lor nedorita, au fost programate sa actioneze doar dupa ce au fost apasate mai mult timp. Cea mai diferita functionalitate legata de aplicatie este cea de automatizare, deoarece va avea nevoie de crearea unui nou thread care va parcurge lista de pozitii memorate, dar fara a bloca controlul altor butoane sau slider, care sunt esentiale pentru a opri sau pentru a efectua alte operatii concomitent.

5 Concluzii si dezvoltari ulterioare

Tema proiectului a fost una foarte interesanta si interactiva, tocmai motivul pentru care am ales sa implementez acest brat robotic care poate fi automatizat. Consider ca este una dintre cele mai utile teme pentru dezvoltarea unui inginer si initierea acestuia in acest domeniu in plina expansiune. Unele dezvoltari pe care ar fi dorit sa le mai adaug ar fi fost folosirea unui cadru metalic mai solid si mai bine gandit decat acest kit fragil de plastic, dar acesta implica si folosirea de servomotoare mult mai puternice si mai rezistente, avand probleme cu unele dintre cele folosite si fiind nevoie sa le desfac sa le repar. O alta dezvoltare semnificativa pe care ar trebui sa o mentionez este adaugarea de optiuni de automatizare prin componente fizice si un ecran care sa afiseze unghiurile pozitiei curente in timp real, dar si optimizarea si finisarea aplicatiei Android pentru a putea fi folosita si pentru alte proiecte asemănătoare, lucru pe care l-am considerat cand am realizat-o, astfel fiind o dezvoltare usor de realizat.

6 Bibliografie

- [1] Autodesk Tinkercad
- [2] Arduino Documentation
- [3] Android Developers