



Universitatea Politehnica Timișoara
Facultatea de Automatică și Calculatoare
Departamentul de Automatică și Informatică Aplicată



Rover cu braț robotic controlat prin aplicație mobilă

– Proiect de diplomă –

Student:

Bociort Dinu - Iulian

Conducător științific:

Ş.l.dr.ing. Ana Maria Dan

Timișoara

2021

Cuprins

Cuprins	2
1. Introducere	3
1.1 Descrierea temei si incadrarea aplicatiei intr-un domeniu de studiu	4
1.2 Obiectivele lucrării și specificațiile aplicației	5
1.3 Structura lucrării pe capitole	6
2. Prezentarea sistemului si a resurselor utilizate	7
2.1. Mecanica	8
2.2. Componente Hardware	14
2.3. Componente Software	26
3. Implementarea sistemului	31
3.1. Modul 1: Comunicarea master-slave	31
3.2. Modul 2: Comunicarea device mobil - microcontroler	37
4. Utilizarea sistemului	42
5. Concluzii	47
Bibliografie	49

1. Introducere

Un robot este o mașină - în special una programabilă de un computer - capabilă să efectueze automat o serie complexă de acțiuni. Roboții pot fi ghidați de un dispozitiv de control extern sau controlul poate fi încorporat în interior. Roboții pot fi construiți pe linia formei umane, dar majoritatea roboților sunt mașini concepute pentru a îndeplini o sarcină, fără a ține cont de estetica lor. [1]

Un robot este un operator mecanic sau virtual, artificial. Robotul mobil este un sistem compus din mai multe elemente: mecanică, senzori și actuatori precum și un mecanism de direcționare. Mecanica stabilește înfățișarea robotului și mișcările posibile pe timp de funcționare. Senzorii și actuatorii sunt întrebuințați la interacțiunea cu mediul sistemului. Mecanismul de direcționare are grija ca robotul să-și îndeplinească obiectivul cu succes, evaluând, de exemplu, informațiile senzorilor. Acest mecanism reglează motoarele și planifică mișcările care trebuie efectuate. [1]

Roboții pot fi autonomi sau semi-autonomi și pot varia de la umanoizi, precum Advanced Step în Innovative Mobility (ASIMO) și TOSY Ping Pong Playing Robot (TOPIO) la roboți industriali, roboți de operare medicală, roboți de asistență pentru pacienți, roboți de terapie pentru câini, în mod colectiv roboți de roi programăți, drone UAV, precum General Atomics MQ-1 Predator și chiar nano roboți microscopici. [1]

Primii roboți autonomi electronici cu comportament complex au fost creați de William Grey Walter de la Burden Neurological Institute din Bristol, Anglia în 1948 și 1949. Primii săi roboți, numiți Elmer și Elsie (figura 1.1.1), au fost construiți între 1948 și 1949 și au fost adesea descriși ca broaște țestoase datorită formei lor și ritmului lent de mișcare. Roboții de tip broască țestoasă cu trei roți erau capabili de fototaxie, prin care își puteau găsi drumul către o stație de reîncărcare atunci când slăbea puterea bateriei. [1]

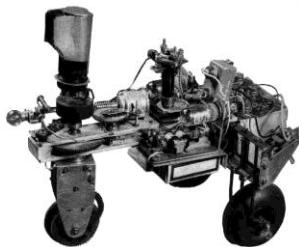


Figura 1.1.1 – Elise[1]

Primul braț robotic programabil și operat digital a fost inventat de George Devol în 1954 și a fost numit în cele din urmă Unimate (figura 1.1.2). Aceasta a pus în cele din urmă bazele industriei moderne a roboticii. Devol a vândut primul Unimate către General Motors în 1960 și a fost instalat în 1961 într-o fabrică din Trenton, New Jersey, pentru a ridica bucăți fierbinți de metal de la o mașină de turnare sub presiune și a le stiva. Brevetul Devol pentru primul braț robot programabil acționat digital reprezintă fundația industriei moderne a roboticii. [2]



Figura 1.1.2 – *Unimate*[2]

1.1 Descrierea temei si incadrarea aplicatiei intr-un domeniu de studiu

Pentru a realiza un robot care se deplasează cu ajutorul roților, sau un braț robotic, se poate utiliza domeniul sistemelor bazate pe microcontrolere. Cu ajutorul acestor sisteme, se pot crea ansambluri care pot îndeplini anumite funcții, cum ar fi mutatul unor obiecte dintr-un loc în altul. După mai multe cercetări, am hotărât ca partea hardware a sistemului meu să aibă la bază plăcuțe de dezvoltare cu microcontroler.

Lucrarea a avut ca și scop final realizarea unui rover cu braț mobil comandat printr-un sistem cu microcontrolere. Ansamblul controlat cu ajutorul unei aplicații mobile printr-o interfață intuitivă și ușor de folosit. Prin intermediul aplicației mobile, robotul va fi capabil de memorarea și reproducerea unor scenarii create de utilizator.

Microcontrolerul este un micro computer comprimat, fabricat pentru a controla funcțiile sistemelor incorporate în mașini de birou, roboți, electrocasnice, autovehicule și o serie de alte dispozitive. Astăzi sunt disponibile pe piață diferite tipuri de microcontrolere cu capacitați diferite, cum ar fi microcontrolere de 4 biți, 8 biți, 64 biți și 128 biți. Un microcontroler cuprinde componente precum - memorie, periferice și cel mai important procesor. Microcontrolerle sunt utilizate practic în dispozitivele care necesită un anumit control pentru a fi aplicat de către utilizatorul dispozitivului. [3]

Originea microcontrolerelor și a microprocesoarelor datează încă din 1959, odată cu crearea tranzistorului MOS de către Mohamed M. Atalla și Dawon Kahng în laboratoarele Bell. Primul microprocesor integrat într-un singur cip a fost Intel 4004, inventat de corporația Intel în 1971. Primul microcontroler este TMS 1000, care combină memorie read-only, memorie read/write, procesor și un clock, integrate într-un singur cip. [3]

Cu timpul, domeniul aplicațiilor bazate pe microcontrolere a avansat și s-a extins la nivel global. Aceste sisteme se regăsesc atât la aparate industriale complexe, cât și la becurile din propriile locuințe. Această dezvoltare accelerată a microcontrolerelor a cauzat nevoie de a le putea comanda într-un mod cât mai simplu și eficient. Astfel a apărut IoT (Internet of Things).

Motivația alegerii acestei teme provine din dorința de a integra cunoștințele dobândite pana acum într-un proiect îndrăzneț, dar totodată și de a mă supune pe mine însumi la încercare. Așadar, bazându-mă pe plăcuțe de dezvoltare Arduino, care au incorporat un microcontroler de tip

Atmega328p, am dorit să creez un robot de tip rover. Acesta este capabil să se deplaseze cu ajutorul a 4 roți și dispune în partea superioară de un braț dotat cu un clește.

Am creat acest sistem bazat pe microcontroler din dorința de a realiza un sistem unic de control și monitorizare la distanță. Acest lucru este relativ ușor de realizat astăzi, mulțumită gamei largi de componente aflate pe piață. Obiectivul principal în construirea unui astfel de sistem este minimizarea costurilor, în timp ce se ajunge la o performanță cât mai ridicată.

1.2 Obiectivele lucrării și specificațiile aplicației

În realizarea acestui sistem bazat pe microcontroler, se vor urmări următoarele obiective principale:

- Realizarea unui model 3D al carcasei/sașifului, în care vor fi depozitate componente hardware, și printarea acestuia împreună cu modelul 3D al brațului robotic.
- Integrarea și fixarea componentelor hardware atât în carcasa roverului cât și pe structura brațului robotic, realizate în obiectivul anterior.
- Realizarea și implementarea unui soft pentru microcontrolere, pentru controlul motoarelor și citirea senzorilor.
- Realizarea aplicației mobile, prin intermediul căreia robotul va putea fi controlat de la distanță.

Pentru realizarea acestor obiective, componentele necesare se împart în 3 categorii:

1) Materiale de construcție:

- Imprimanta 3D cu ajutorul căreia se va printa modelul 3D;
- Filament PLA ca și material de printare pentru imprimantă;
- Șuruburi și piulițe pentru a asambla componentele printate;
- Roți pentru deplasare;
- 3 bare hexagonale lucrate la strung pentru a îmbina roțile cu axul motorului;

2) Componete hardware:

- Motor pas cu pas 17HS4401;
- Servo motor MG90S;
- Modul Arduino Uno (ATmega328p);
- Modul Bluetooth HC-05;
- Modul de expansiune CNC V3;
- Driver motor DRV8825;
- Senzor ultrasonic HC-SR04;
- Cooler;
- Fire;
- Baterie 12V, 8,8Ah;
- Un telefon mobil cu sistem de operare Android și Bluetooth;

3) Componete Software:

- Autodesk Fusion 360 pentru realizarea modelului 3D al carcasei;
- Arduino IDE pentru realizarea algoritmului microcontrolerului;
- Android Studio IDE pentru realizarea aplicației mobile;

1.3 Structura lucrării pe capitole

Lucrarea este structurată în 4 capitole, care acoperă de la funcționarea sistemului la implementarea și înțelegerea folosirii acestuia.

În capitolul 2 este detaliat prezentată atât arhitectura sistemului, cât și specificațiile acestuia, respectiv funcțiile componentelor de natură atât software, cât și hardware. Capitolul 2 redă în 3 subcapitole toate punctele importante ale aplicației realizate, care descriu modul în care s-au folosit diferite elemente în cardul proiectului.

În capitolul 3 se prezintă modulele în care este împărțit întreg sistemul. Acest capitol descrie în 2 subcapitole rolul, funcționalitatea și structura fiecărui submodul. Structura este prezentată cu ajutorul schemelor electrice, iar algoritmul cu ajutorul organigramelor funcționale.

În capitolul 4 este prezentat modul în care utilizatorul ar trebui să folosească sistemul. În acest capitol sunt descrise pe rând toate funcțiile aplicației și modul în care utilizatorul poate interacționa în mod corect cu acestea.

În capitolul 5 sunt prezente concluziile finale. Aici se prezintă problemele întâmpinate pe parcursul conceperii proiectului, alături de soluțiile găsite pentru unele din probleme, dar și perspective noi de dezvoltare ale proiectului.

2. Prezentarea sistemului și a resurselor utilizate

În cadrul proiectului s-a realizat un robot mobil, bazat pe microcontroler, de tip rover cu 4 roți, care dispune în partea superioară de un braț robotic cu clește, capabil de o mișcare pe 3 axe. Comunicarea dintre microcontroler și telefonul mobil se realizează prin intermediul rețelei Bluetooth. Aplicația mobilă permite atât controlul de către utilizator a mișcării robotului mobil, respectiv a brațului, cât și memorarea a diferitelor scenarii. Un scenariu înseamnă un set de comenzi consecutive pe care utilizatorul le introduce în aplicația mobilă și pe care apoi, roverul este capabil să le repete la cerere.

Realizarea proiectului a constat în:

- Crearea unui model 3D și printarea părții mecanice;
- Poziționarea componentelor hardware în carcăsă și efectuarea legăturilor
- Conceperea algoritmilor necesari pentru funcționare și scrierea codului

Componentele care alcătuiesc acest proiect și despre care voi discuta în acest capitol se împart în 3 categorii mari, acestea fiind:

Partea mecanică, care cuprinde:

- Brațul robotic și componente din care acesta este format;
- Carcasă roverului;
- Modulele de îmbinare dintre roți și motoare

Partea hardware, care cuprinde:

- Motor pas cu pas;
- Servo motor
- Modul Arduino Uno
- Modul Bluetooth;
- Modul de expansiune CNC;
- Driver motor;
- Senzor ultrasonic;
- Cooler;
- Fire;
- Baterie;
- Un telefon mobil cu sistem de operare Android și Bluetooth;

Partea software, care cuprinde:

- Algoritmul utilizat pe modulele arduino;
- Algoritmul utilizat pentru aplicația mobilă;

2.1. Mecanica

În acest subcapitol se prezintă mecanismele de funcționare a robotului și procesul decizional prin care robotul a fost construit. În figura 2.1.1, respectiv 2.1.2, se poate observa sistemul construit.

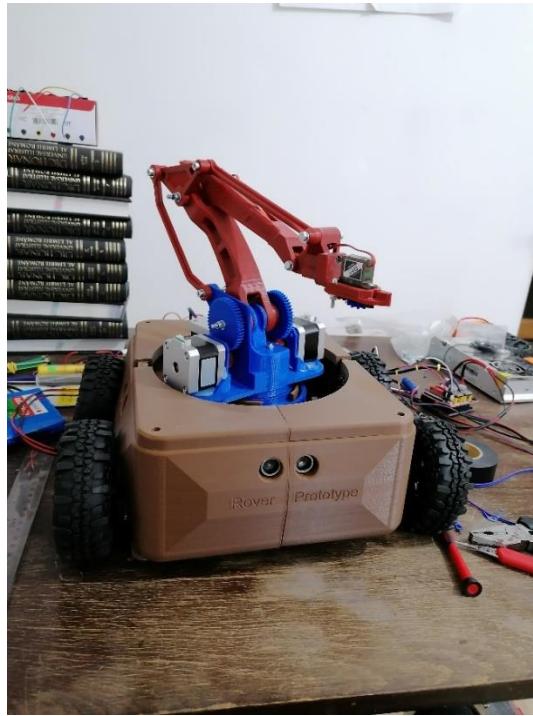


Figura 2.1.1 – *Rover Prototype* vedere din față



Figura 2.1.2 – *Rover Prototype* vedere de sus

Brațul este format din mai multe componente printate, legate între ele prin șuruburi suruburi (figura 2.1.3 -2.3.9). El este capabil de o mișcare pe 3 axe, fiind acționat prin învârtirea roților dințate de către motoarele pas cu pas. Totodată acesta dispune de un clește, acționat de un servo motor.



Figura 2.1.3 – Braț robotic

Acesta are ca și dimensiuni, o înălțime de 27cm și o lungime a deschiderii brațului de 37cm.

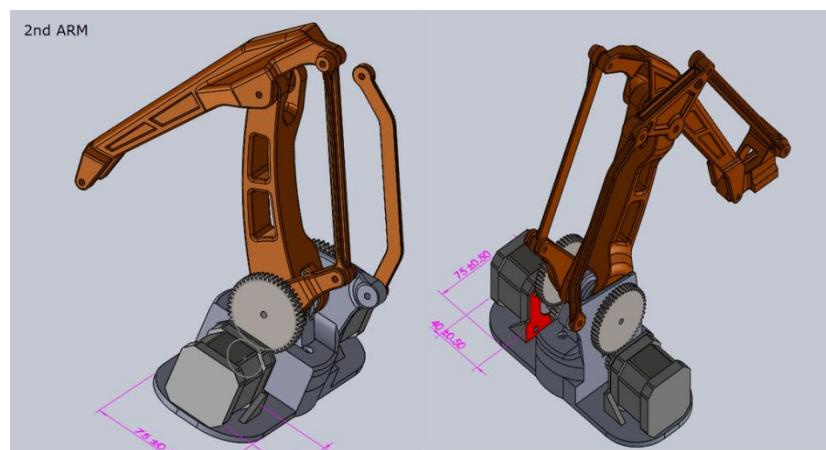
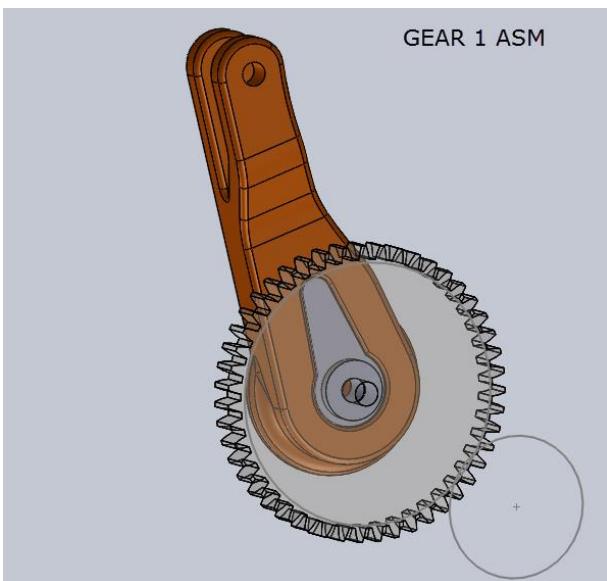
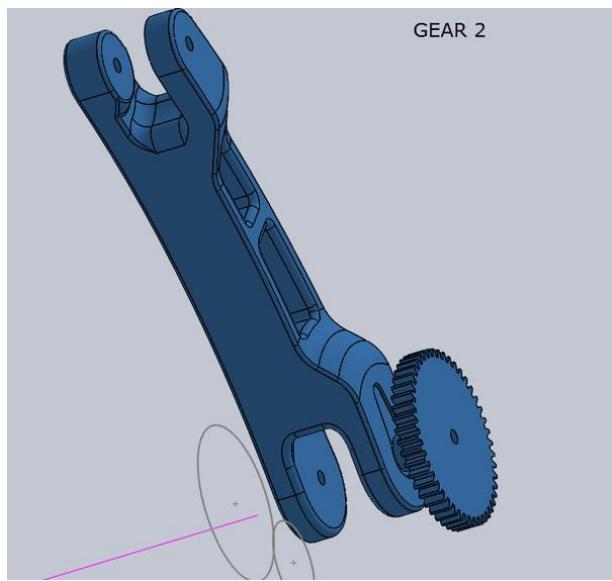


Figura 2.1.4 – Asamblarea brațului [4]

Pentru a se deplasa pe axele X și Y, brațul se folosește de aceste „mini” brațe care acționează ulterior niște pârghii, prin care modelează unghiul de deschidere a brațului (figura 2.1.5 si 2.1.6).



GEAR 1 ASM



GEAR 2

Figura 2.1.5 – Angrenaj braț 1[4]

Figura 2.1.6 – Angrenaj braț 2[4]

Pe axa Z, el se rotește pe 10 rulmenți din metal (figura 2.1.7), fiind capabil de o mișcare de rotație de 180 de grade cu ajutorul suportului din figura 2.1.8.



Figura 2.1.7 – Angrenaj braț cu rulmenți



Figura 2.1.8 – Angrenaj braț cu suport

Pentru ca roțile dințate corespunzătoare fiecărei axe de mișcare să fie învărtite de către motorul pas cu pas, i s-a aplicat un adaptor tot de tip roată dințată, pe axul acestuia.

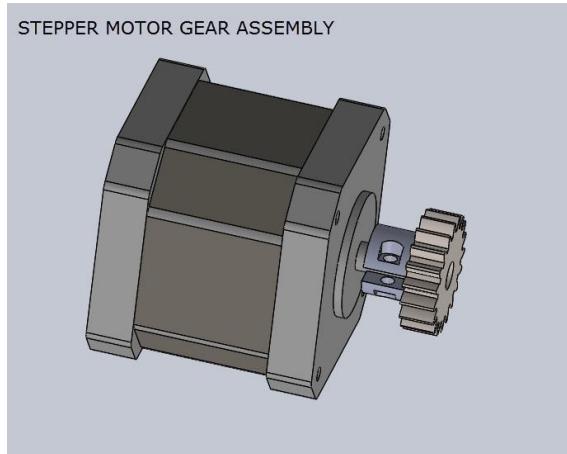


Figura 2.1.9 – Angrenaj motor[4]

Ca și sursă de inspirație pentru crearea brațului, am folosit site-ul „Thingiverse.com”, de unde am luat modelul 3D al brațului și sugestii de implementare.

Carcasa din figura 2.1.10 are o dimensiune de 22 pe 27 în interior și 27 pe 29 în exterior. Am ajuns la aceste dimensiuni prin calcularea spațiului necesar pentru fiecare componentă hardware care urma a fi încastrată în carcăsa.



Figura 2.1.10 – Carcasă rover

Carcasa a fost concepută în softul de modelare 3D creat de Autodesk, numit Fusion360. În acesta am măsurat dimensiunea fiecărei componente hardware și i-am calculat un loc specific în carcasă. În această carcasă încap: 4 motoare pas cu pas, 1 baterie de 12v, 3 plăcuțe Arduino Uno, 4 senzori ultrasonici, un modul Bluetooth și cabluri. Carcasă, având aceste dimensiuni, pentru a fi printată a trebuit să fie secționată în 5 componente, cele 4 colțuri ale carcasei și pilonul central, care s-au îmbinat mai apoi ca și niște piese de puzzle.



Figura 2.1.11 – Componenta carcsei 1

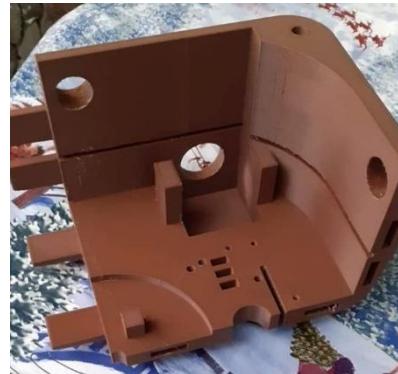


Figura 2.1.12 – Componenta carcsei 2

Atât brațul cât și carcasă, au fost printate cu ajutorul unei imprimante 3D, care folosește tot motoare pas cu pas pentru deplasarea pe axe (figura 2.1.13). Ca și material utilizat la printare, am folosit filament PLA, cu o grosime a filamentului de 1.75mm.

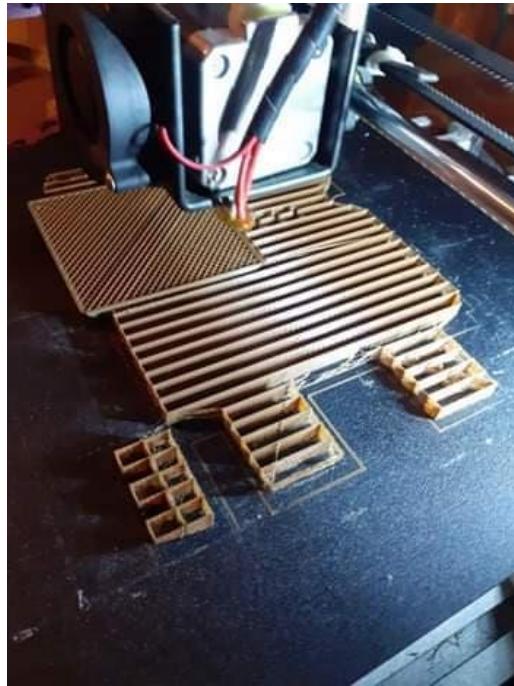


Figura 2.1.13 – Prinarea carcsei

Pentru deplasarea roverului am folosit 4 roți, care au un înveliș dintr-un cauciuc subțire, fiind umplute cu un burete presat de o jantă de plastic, acestea având 13 cm în diametru.



Figura 2.1.14 – Roată rover 1



Figura 2.1.15 – Roată rover 2

Aceste roți sunt acționate separat de către un motor pas cu pas, iar ca elemente de legătură și de prindere între roți și motoarele pas cu pas, am folosit bare hexagonale de metal, care au fost lucrate la strung.

Aceste bare au o lungime de 5cm și un hexagon de 1,2cm (figura 2.1.16 – 2.1.17).



Figura 2.1.16 – Profil hexagonal 1

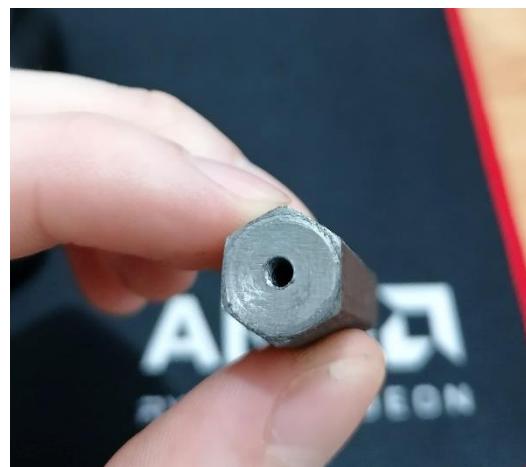


Figura 2.1.17 – Profil hexagonal 2

Aceste bare hexagonale au fost găndite astfel încât axul motorului să intre în interior până la jumătate, apoi să fie imobilizat cu ajutorul unui șurub de 3mm, care i se opune perpendicular pe scobitura axului. Pe cealaltă jumătate a barei se află un filet pentru un șurub, care odată înfiletat unește roata cu bara hexagonală. Îmbinarea poate fi observată în figura 2.1.18.



Figura 2.1.18 – Îmbinarea axului motorului cu roata

2.2. Componente Hardware

Schema informațională din Figura 2.2.1 este compusă din totalitatea componentelor hardware folosite. Se regăsesc atât componente de intrare, precum senzorii ultrasonici, și componente de ieșire, precum motoarele pas cu pas sau servo motorul, cât și componente mixte, precum modulul bluetooth sau modulul arduino.

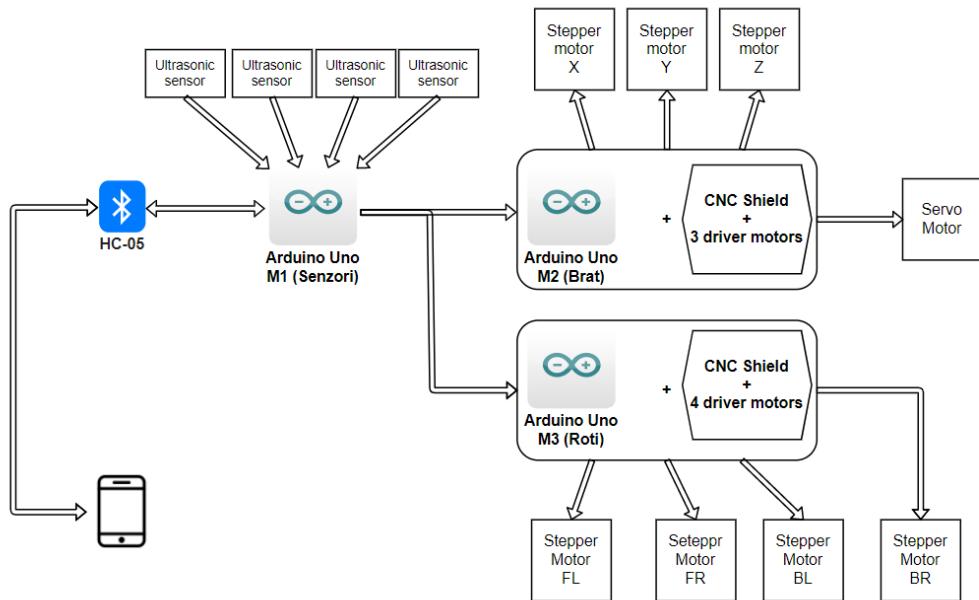


Figura 2.2.1 – Arhitectura sistemului

Sistemul realizat este unul bazat pe microcontrolerul ATmega328p, aflat în componența plăcii de dezvoltare Arduino Uno. Modelul plăcii a fost ales din mai multe motive.

Unul dintre aceste motive este faptul că satisfacă toate condițiile necesare realizării proiectului, placă fiind compatibilă cu placa de expansiune CNC, care este montată deasupra, reușind să facă mai ușoară conexiunea cu driverele motoarelor pas cu pas.

Un alt motiv de care am ținut cont în alegerea plăcii este prețul. Acest tip de placă, având un cost variabil între 20 și 40 de lei, are un preț foarte avantajos comparativ cu competitorii săi. Datorită popularității acesteia, dobândite printr-un raport performanță-preț foarte bun, accesul la informații despre Arduino Uno este foarte ușor de realizat.

În Figura 2.2.2 se poate observa schema de principiu a ansamblului hardware în care se regăsesc toate elementele utilizate în asamblarea sistemului.

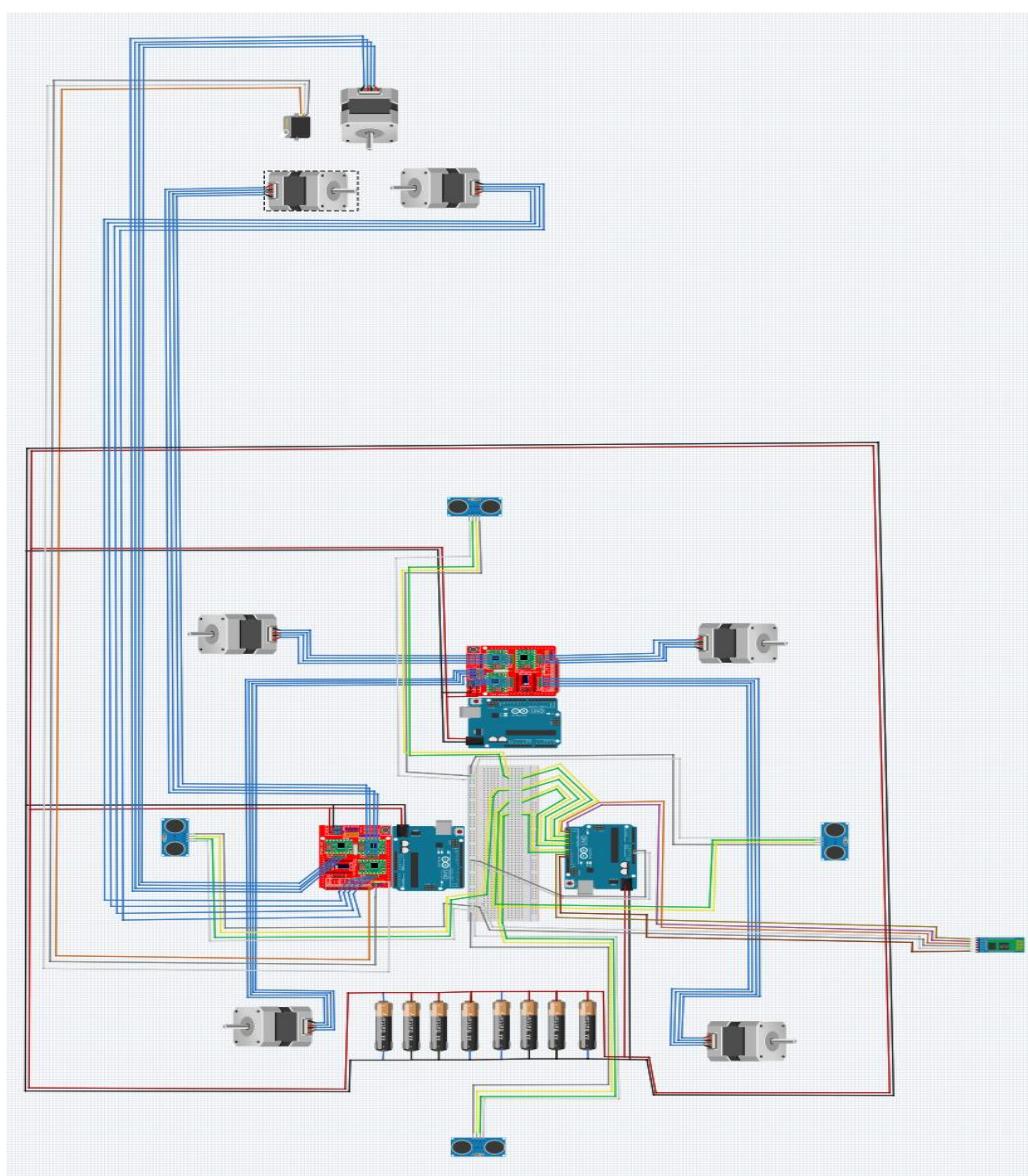


Figura 2.2.2 – Schema electrică a sistemului

În continuare sunt prezentate componentele sistemului, alături de caracteristicile fiecăreia.

➤ Arduino Uno

Arduino este o companie open-source bazată pe componente hardware ușor de folosit, cât și componente software. Plăcile de dezvoltare de tip Arduino sunt capabile să citească intrări declanșate de diferite evenimente cum ar fi lumina pe un senzor, un deget apăsând un buton, sau chiar un mesaj Twitter. Acestea sunt capabile să răspundă la aceste intrări cu evenimente provocate de ieșirile lui, cum ar fi aprinderea unui bec sau pornirea unei alarme. [5]

Termenul de arduino cuprinde atât placa de dezvoltare, cât și mediul de programare complet accesibil și gratuit oricărei persoane dornice să realizeze diverse aplicații. Se enumeră o mulțime de domenii de aplicabilitate a modulului Arduino, cum ar fi: securitatea (crearea de sisteme de supraveghere video, cu senzori de mișcare, căldură etc.; carduri de acces în clădiri), automatizarea unei locuințe (închidere/deschidere jaluzele, pornirea sistemului de irigații, aprinderea becurilor, detectarea gazelor și oprirea acestora, etc.), medicină (un robotel care ține evidență și livrează pastilele pacienților, un pat reglabil), militar (robot care poate dezamorsa bombe). Pe lângă acestea, se pot face și lucruri simple precum apăsarea unui buton, să aprindă un bec sau să pornească un ventilator. [5]

Un mare plus este că pe internet se poate găsi absolut orice legat de această platformă, de la arhitectura diverselor blocuri, până la detalii tehnice amănunțite. Este atât de popular deoarece dezvoltatorul a lăsat informația la îndemâna oricărei persoane interesate, în prezent fiind gratuită.

Platforma Arduino utilizează protocoale de comunicație standard, ca de exemplu:

- SPI – Serial Peripheral Interface;
- I2C – Inter-Integrated Circuit;
- USB – Universal Serial Bus;
- UART – Universal Asynchronous Receiver/Transmitter.

În cadrul proiectului, se folosește protocolul de comunicare I2C.

Plăcile de dezvoltare pot fi achiziționate fie de la producătorii oficiali, precum "Adafruit Industries", fie de la alți dezvoltatori, care au construit placă compatibilă cu Arduino. Oricine poate să-și construiască propriul modul Arduino, deoarece detaliile tehnice sunt disponibile. [5]

Există mai multe variante standard de placi Arduino, fiindcă uneori există proiecte care nu necesită prea multe componente interconectate, nefiind necesari mulți pini, iar altele este nevoie de mai mulți pini pentru proiectele mari cu multe componente programabile. [5]

Cea mai des întâlnită și totodată cea care este folosita în acest proiect, se numește Arduino UNO.



Figura 2.2.3 – Arduino Uno[5]

Caracteristici tehnice:

- Microcontroller ATmega328
- Tensiune de operare: 5V
- Tensiune de alimentare recomandată: 7-12V
- Limită de tensiune: 6-20V
- Pini intrare/iesire digitali: 14 (D0, ..., D13)
 - D0 (Rx_D) , D1(Tx_D), D3 (întreruperi)
 - D3, D5, D6, D9, D10, D11 (pentru PWM)
 - D13 (pentru LED)
- Pini analogici de intrare: 6 (A0, ..., A5)
 - Citesc valori între 0 și 1023
 - A4 (I2C – SCA) , A5 (I2C – SCL)
- Memorie Flash 32KB
- SRAM 2KB
- EEPROM 1KB
- Frecvență de lucru: 16 MHz

➤ CNC Shield V3

CNC Schield V3 poate fi utilizat ca placă de expansiune pentru mașini de gravat, imprimante 3D și alte dispozitive similare. Aceasta dispune de 4 sloturi pentru 4 module de acționare a motorului pas cu pas. Fiecare motor pas cu pas are nevoie de doar două porturi IO. Se alimentează separat cu tensiunea necesară funcționării motoarelor, shield-ul fiind prevăzut cu câte un condensator de filtraj suplimentar pentru fiecare driver.

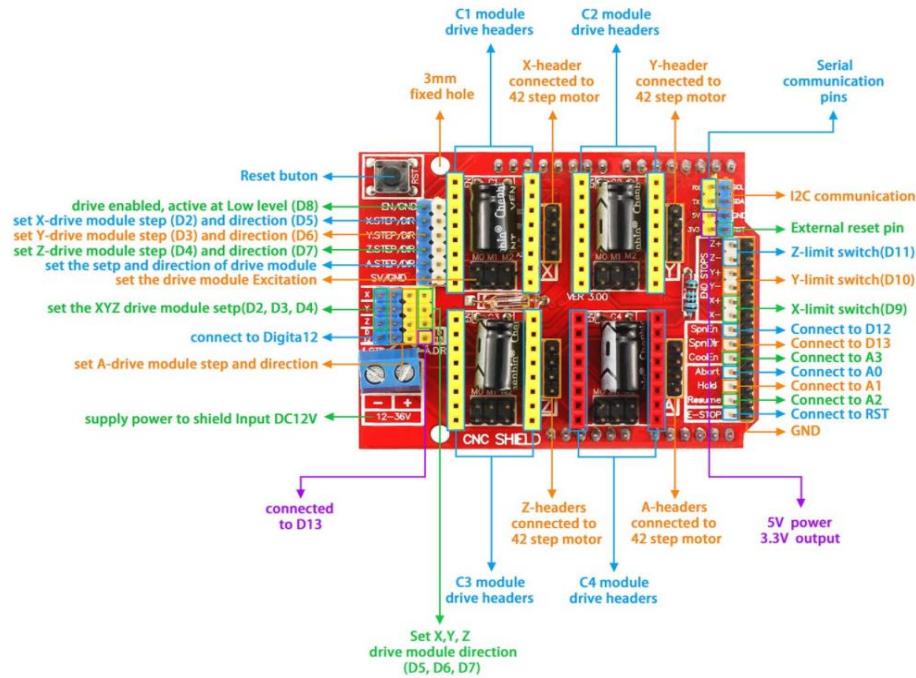


Figura 2.2.4 – CNC Shield V3[6]

Cu ajutorul unor jumperi, se poate regla puterea maximă curentă, astfel încât să se obțină o rată mai mare a pasului. Acest lucru este posibil prin conectarea pinilor M1, M2, M3, conform tabelelor de mai jos, în funcție de driver-ul folosit.



Figura 2.2.5 – Schema de reglare[6]

Caracteristici tehnice:

- Tensiune circuite logice: 3 - 5.5V
- Tensiune motor: 8 - 35V
- 5 / 6 moduri pentru pași
- Protecție la subtensiune, supracurent și temperatură ridicată
- Current maxim 2A

➤ Modul driver motor pas cu pas DRV8825

Un driver de motor este un dispozitiv sau un grup de dispozitive, care coordonează într-o manieră prestativă performanța unui motor electric. Driver-ul DRV8825 oferă o soluție integrată pentru imprimante, scannere și alte aplicații automate. Dispozitivul dispune de două punți H-bridge și un indexator microstepping și este construit astfel încât să poată conduce un motor pas cu pas bipolar. Acesta este capabil să conducă până la 2,5A de curent de la fiecare ieșire. Având o interfață simplă STEP/DIR, permite controlul acestuia cu ușurință. Inelele de moduri permit configurarea motorului de la mod complet până la modurile de 1/32-pas. [7]

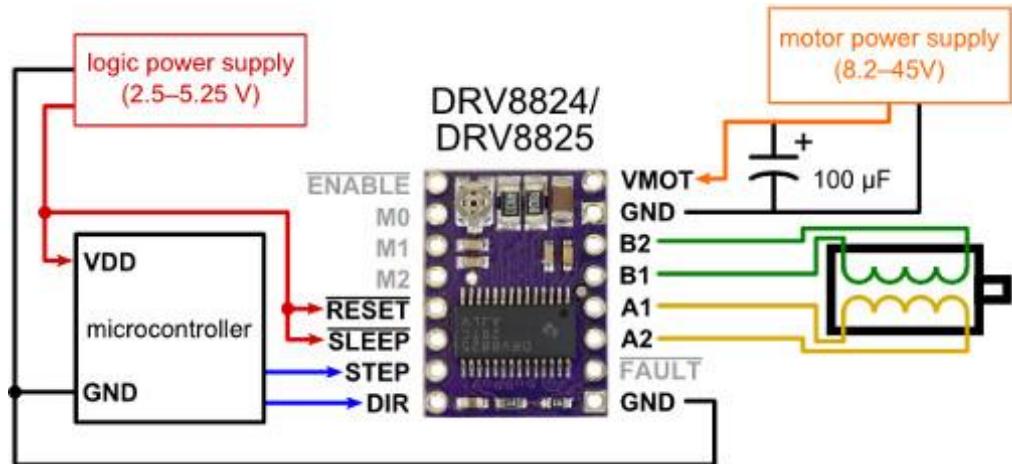


Figura 2.2.6 – Motor driver DRV8825[8]

Pentru a folosi corespunzător driver-ul DRV8825, acesta necesită calibrarea tensiunii. Calibrarea se efectuează în funcție de motorul folosit, pentru ca acesta să nu fie sub alimentat, astfel încât să nu fie capabil să asigure cuplul ideal, sau supra alimentat, supraîncălzindu-se și deteriorând componente.

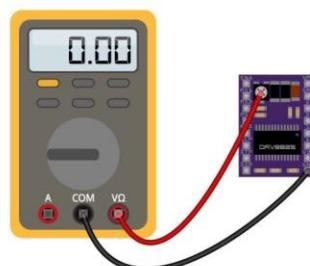


Figura 2.2.7 – Poziționare multimetru pentru calibrare[8]

Acum reglaj de tensiune se calculează după formula:

$$V_{ref} = (\text{Curentul nominal al motorului}) / 2 ;$$

Pentru funcționarea rover-ului, se folosesc 7 drivere de acest tip, care sunt setate pe un mod de lucru cu 1/32-pas, acestea controlând fiecare câte un motor pas cu pas. Motoarele folosite având un curent nominal egal cu 1.7A, driverele au fost setate conform formulei prezentate anterior cu un V_{ref} egal cu 0.85V

Caracteristici tehnice:

- Rezoluții diferite ale pașilor: 1-pas, 1/2-pas, 1/4-pas, 1/8-pas, 1/16-pas, 1/32-pas
- Interfață CAN direct cu sisteme de 3,3V și 5V
- 4 straturi de cupru, pentru a îmbunătăți disiparea căldurii
- Curentul maxim este de 2,5A
- Este potrivit pentru motoarele pas cu pas până la 8.2V – 45V

➤ Motor pas cu pas 17HS4401

Motorul pas cu pas este un motor de curent continuu fără perii, în care rotația este divizată într-un anumit număr de pași, care rezultă din construcția motorului. În mod standard, o rotație completă a arborelui cu 360° este divizată în 200 de pași, ceea ce înseamnă că arborele efectuează un nou pas la fiecare $1,8^\circ$. Ca urmare a faptului că o rotație completă a arborelui este împărțită în mai multe segmente discrete, motorul pas cu pas nu se rotește în mod continuu, ci prin pași, trecând prin stări intermediare. De aceea, funcționarea unui asemenea motor este însoțită de un sunet caracteristic sau de vibrații. [9]



Figura 2.2.8 – Motor pas cu pas[9]

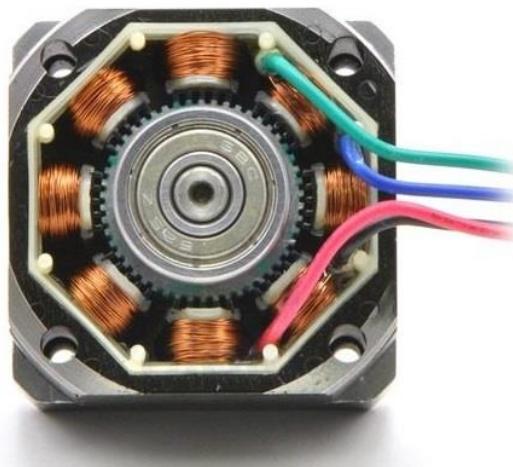


Figura 2.2.9 – Mecanism motor pas cu pas[9]

Funcționarea motorului pas cu pas, ca și cea a oricărui alt motor electric, constă în crearea unui câmp magnetic care, acționând asupra unei bobine, determină rotirea rotorului. Spre deosebire de motoarele electrice "obișnuite", rotorul motorului pas cu pas, după aplicarea impulsului de alimentare, execută o rotație cu un unghi strict determinat. Acesta fiind un motor pas cu pas bipolar, în interior sunt prezente două bobine, separate între ele, care revin câte unei faze de lucru. După cum se vede în Figura 2.2.10, pentru a comanda un motor bipolar trebuie asigurată modificarea direcției de trecere a curentului prin întreaga bobină, iar acest lucru necesită un comutator dublu, realizat cel mai des ca punte de tranzistoare. În fiecare fază de funcționare, curentul trece prin întreaga înfășurare. [9]

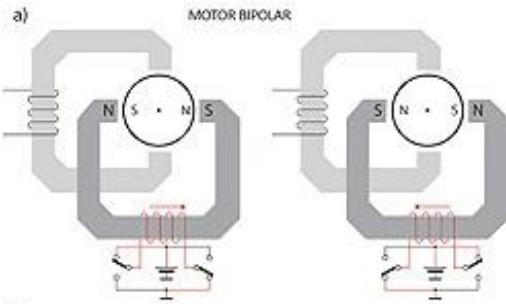


Figura 2.2.10 – Schemă de funcționare motor pas cu pas[9]

Funcționare în modul pas întreg:

Principiul de funcționare al motorului pas cu pas, în modul de pas întreg, este ilustrat în figura 2.2.11. În acest mod, motorul se rotește cu unghiul rezultat din construcția acestuia, care poate fi, de exemplu, de $1,8^\circ$. Așa cum putem calcula cu ușurință, în acest caz, pentru o rotație completă, este necesară efectuarea a 200 de pași ($200 \times 1,8^\circ = 360^\circ$).

Pasul arborelui este efectuat după alimentarea uneia sau a două bobine. Funcționarea cu alimentarea unei singure bobine necesită o putere minimă a driver-ului. În cazul unei funcționări bifazate, cu alimentarea bobinelor aflate una în fața celeilalte, este nevoie de o putere de două ori mai mare a curentului de alimentare, dar, astfel, cresc și viteza și cuplul motor.[9]

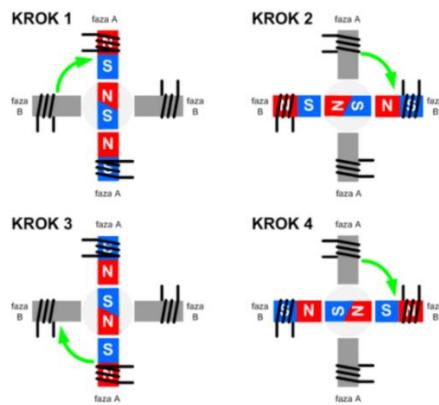


Figura 2.2.11 – Schemă de funcționare pas întreg [9]

Funcționare în modul semi pas:

Principiul de funcționare al motorului în modul de semi pas este ilustrat în figura 2.2.12. Așa cum arată și numele, în acest mod, pasul discret al rotorului este împărțit la 2 și, atunci când efectuează un pas, se rotește cu jumătatea unghiului nominal. Dacă raportăm acest lucru la exemplul menționat mai sus, un singur pas se va realiza la fiecare $0,9^\circ$, iar numărul de pași la o rotație completă va crește la 400.

În cazul funcționării în modul de semi pas este necesară alimentarea alternativă a două faze (bobine). Efectul îl reprezintă creșterea cupplului motor față de funcționarea cu alimentare monofazată, o funcționare „mai lină” a motorului și dublarea a rezoluției unghiulare. [9]

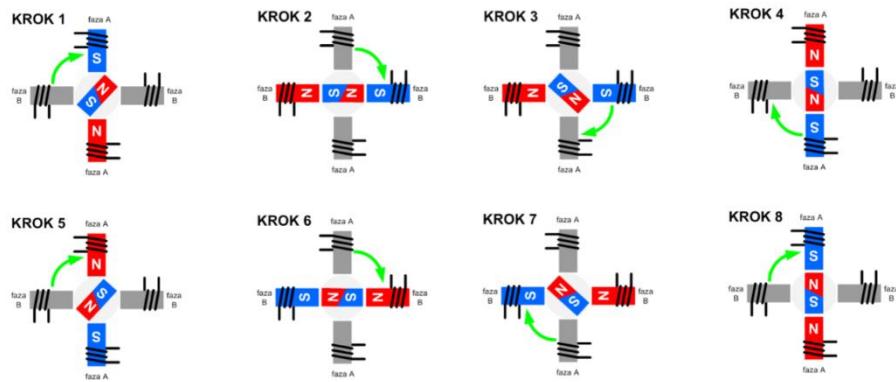


Figura 2.2.12 – Schemă de funcționare semi pas[9]

În cadrul lucrării, se folosesc 7 motoare pas cu pas. 4 dintre ele se folosesc pentru controlul individual al fiecărei roți a rover-ului, iar 3 dintre ele se folosesc pentru controlul fiecărei din cele 3 axe pe care brațul este capabil să se deplaseze.

Caracteristici tehnice:

- Unghi pas: 1.8 grade
- Lungime motor: 40mm
- Current 1.7A la 12V
- Rezistență fază: 1.5ohm
- Inductanță fază: 2.8mH
- Cuplu: 40N x cm
- Precizie pas: +/- 5%
- Acuratețe rezistență: +/- 10%
- Acuratețe inductanță: +/- 20%
- Temperatură maximă: 80°C
- Masa: 280g

➤ Servo motor MG90S

Un servomotor este un dispozitiv electric autonom, care rotește părțile unei mașini cu eficiență ridicată și cu mare precizie. Servomotoarele sunt alcătuite dintr-un motor de curent continuu, un potențiometru acționat de axul motorului ce măsoară unghiul la care acesta se rotește, un circuit ce compară semnalul provenit de la potențiometru cu comanda primită de la utilizator și un mecanism cu roți dințate ce reduce turația motorului, dar crește cuplul acestuia. [10]



Figura 2.2.13 – Servo motor MG90S[10]

Pentru a face acest motor să se rotească, trebuie să alimentăm motorul cu + 5V, folosind firul roșu și maro și să trimitem semnale PWM către firul de culoare portocalie.

Pozitia „0” (impuls de 1.5ms) este în mijloc, „90” (impuls de 2ms) este tot drumul spre dreapta, „-90” (impuls de 1ms) este tot drumul spre stânga.

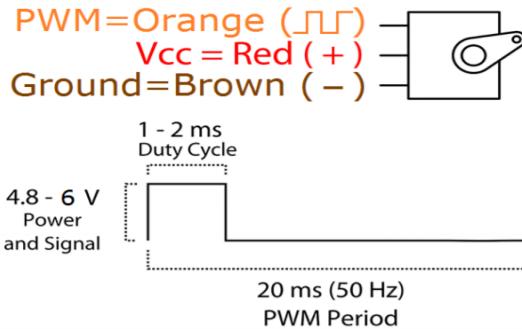


Figura 2.2.14 – Principiu de functionare servo motor[11]

În cadrul proiectului, servomotorul este cel care acționează deschiderea și închiderea cleștelui, aflat pe brațul roverului.

Caracteristici tehnice:

- Tensiune de alimentare: 4.8 – 6v
- Curent în gol: 8 - 10mA;
- Cuplu: 4.8V - 1.8 kgf*cm; 6V - 2.2 kgf*cm;
- Viteză de rotație: 4.8V - 0.1s/60°; 6V - 0.08s/60°;
- Frecvență PWM: 50Hz

➤ Modul Bluetooth HC-05

Modulul Bluetooth HC-05 este un modul Bluetooth SPP (Serial Port Protocol) ușor de utilizat, proiectat pentru configurarea conexiunii wireless în mod transparent. Acesta poate fi utilizat într-o configurație de tip Master-Slave, făcându-l o soluție bună pentru comunicația wireless. Acest modul comunică prin intermediul interfeței USART, la o rată baud de 9600. De asemenea, se pot configura valorile predefinite, folosind comenzi specifice. [12]

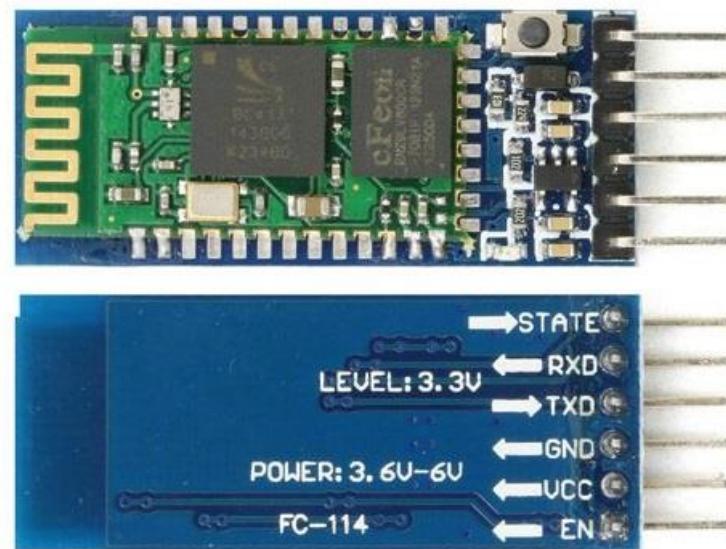


Figura 2.2.15 – Modul bluetooth HC-05

Modulul Bluetooth este utilizat pentru transmiterea și recepționarea wireless de informații la și de la alt modul Bluetooth cu care este conectat. În cadrul proiectului, modulul Bluetooth asigură comunicarea wireless dintre telefonul mobil și Arduino, permitând transmisia și receptia comenziilor.

Descrierea pinilor:

- ENABLE/Key: acest pin este folosit pentru a comuta între modul de date (setat LOW) sau modul AT de comandă (setat HIGH)
- Vcc: Alimentează modulul. Conectat la 5V
- Ground: Pinul de ground, conectat la ground-ul sistemului
- Tx: Pinul de transmisie serială a datelor. Toate datele primite prin transmisie Bluetooth vor fi trimise serial prin intermediul acestui pin
- Rx: Pinul de recepție serială a datelor. Toate datele primite în mod serial prin acest pin vor fi distribuite în mod Broadcast prin transmisie Bluetooth
- STATE: Pinul de stare este conectat la un led al plăcuței, și poate fi folosit ca un răspuns primit în urma funcționării corecte al modulului

Caracteristici tehnice:

- Tensiune alimentare: 3.6V - 6V
- Senzitivitate -80dBm (decibel-miliwatt)
- Până la +4dBm putere de transmisie prin radiofrecvență
- PIO(Programmable Input/Output) control;
- Rază acoperire: max. 100m
- Interfață UART cu rata de transfer configurabilă
- Antenă integrată

➤ Senzor ultrasonic HC-SR04

Senzorul de distanță cu ultrasunete HC-SR04 folosește sonar pentru a determina distanța față de un obiect, cu citire stabilă și precizie ridicată de 3 mm. Modulul include transmițător cu ultrasunete, receptor și circuit de control. Pentru a genera ultrasunete, declanșarea trebuie setată pe o stare ridicată pentru 10 microsecunde. Pinul de declanșare trimite explozia sonică, care se deplasează cu viteza sunetului. Unda sonară va fi recuperată odată ce interacționează cu un solid sau lichid și va fi recepționat în pinul echo. [13]



Figura 2.2.16 – Senzor ultrasonic HC-SR04

Distanța dintre senzor și obiect va fi calculată folosind timpul de călătorie. Una dintre caracteristicile deosebite ale acestui senzor de distanță este faptul că acesta poate detecta nu doar distanța dintre el și un obiect solid, ci poate detecta și distanța dintre el și un lichid. [13]

Rover-ul folosește acești 4 senzori ultrasonici poziționați față-spate și stânga-dreapta, pentru a determina dacă există obstacole pe traseul său.



Figura 2.2.17 – Principiu de funcționare senzor ultrasonic [13]

Formula de calcul este: $\text{Distanță} = \text{Viteza} \times \text{Timp}$

Caracteristici tehnice:

- Tensiune de alimentare: 5V
- Current consumat: 15mA
- Distanță de funcționare: 2cm - 2m
- Unghi de măsurare: 15 grade
- Eroare de doar 3mm
- Durată semnal input: 10us
- Frecvență ultrasunete : 40 kHz

Descrierea pinilor:

- Vcc : Alimentează modulul, conectat la 5 V
- Trigger : Pinul de declanșare este un pin de intrare. Acest pin trebuie să fie menținut sus pentru 10 μ s pentru a iniția măsurarea prin trimiterea undei ultrasonice
- Echo : Pinul de ecou este un pin de ieșire. Acest pin se ridică pentru o perioadă de timp, care va fi egală cu timpul necesar pentru ca una ultrasonică să se întoarcă înapoi în senzor;
- Ground : Pinul de masă, conectat la masa sistemului ;

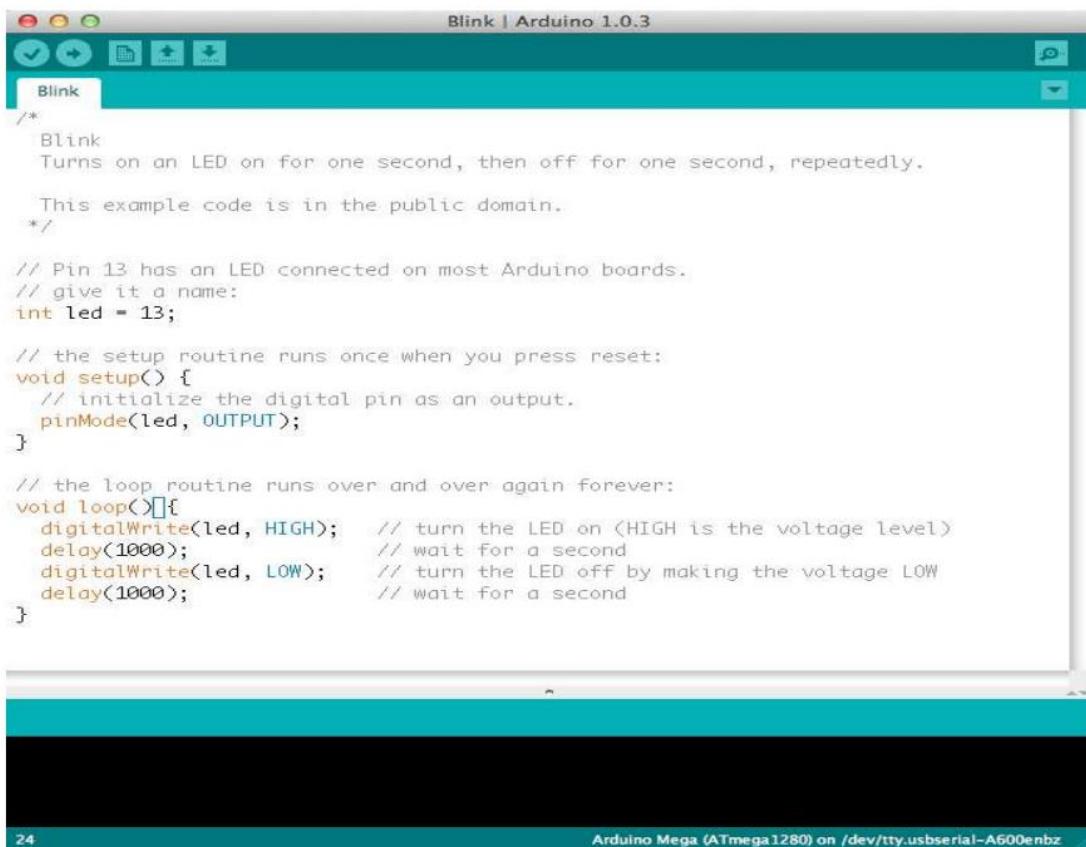
2.3. Componete Software

Pentru realizarea acestui proiect, au fost folosite 2 medii de programare, un program de modelare 3D și un program pentru realizarea schițelor.

➤ Arduino IDE [4]

Arduino IDE (Integrated Development Environment) este o aplicație de tip cross-platform, scrisă în C, C++ și Java. Este folosită în scopul scrierii și încărcării programelor în plăcile de dezvoltare compatibile Arduino, dar și pentru a sprijini dezvoltarea de alte plăci, prin faptul că se poate utiliza și pentru acestea, cu ajutorul unor nucleu terțe.

Arduino IDE suportă limbajele C și C++, acestea trebuind să urmeze un set de reguli speciale și o structură predefinită a codului. Arduino IDE oferă o bibliotecă software a proiectului Wiring, care vine în ajutor cu multe proceduri de intrare/ieșire uzuale. Codul definit de utilizator necesită 2 funcții de bază, una pentru inițializarea programului, și o funcție tip loop, care sunt compilate și conectate cu un cod main(), într-un program executabil ciclic cu ajutorul toolchain-ului GNU, de asemenea inclus în IDE. Programul creat în acest IDE a fost salvat cu extensia “.ino”. Un exemplu de cod scris în Arduino IDE se poate observa în figura 2.3.1.



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0.3". The central area displays the following C++ code for the "Blink" example:

```

/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
    // initialize the digital pin as an output:
    pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}

```

The status bar at the bottom indicates "24" and "Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz".

Figura 2.3.1 – Exemplu de cod scris în Arduino IDE[4]

Odată cu creșterea popularității Arduino ca platformă software, alți furnizori au început să implementeze compilatoare și instrumente (nuclee) de tip “open source”, care pot construi și încărca schițe către alte unități MCU, care nu sunt acceptate de linia oficială a MCU-urilor Arduino.

Utilizând acest IDE și limbajul C, am dezvoltat codul sursă, destinate celor 3 plăci de dezvoltare compatibile Arduino. Scopul acestor aplicații este de a inițializa setările motoarelor și de a le menține într-o stare neutră până la primirea unei comenzi de la telefonul mobil, executând comanda respectivă, apoi revenind la starea neutră. Alte obiective ale aplicațiilor realizate sunt de a citi valorile senzorilor.

➤ Android Studio [14]

Android Studio este bazat pe software-ul IntelliJ IDEA de la JetBrains, este instrumentul oficial de dezvoltare al aplicațiilor Android. Acest mediu de dezvoltare este disponibil pentru Windows, OS X și Linux. Pe 17 mai 2017, la conferința anuală Google I / O, Google a anunțat asistența pentru limbajul Kotlin, utilizat de Android Studio ca limbaj de programare oficial pentru platforma Android, pe lângă Java și C++.

Android Studio este o platformă esențială în dezvoltarea aplicațiilor mobile Android deoarece include un întreg ecosistem tehnic ce permite compatibilitatea cu telefoanele mobile, tabletele, televizoarele și o serie de accesorii, precum ceasurile smart și multe altele. Mai mult, versatilitatea aplicațiilor Android determină ca acestea să poată servi mai multor scopuri odată ce au fost instalate.

Fiind un sistem open source, Android-ul facilitează la maximum opțiunile de individualizare, deoarece aplicațiile mobile pot fi integrate într-o mulțime de instrumente multimedia, precum aplicații audio/video, programe de educație și training, programe bazate pe inteligență artificială și multe altele.

IDE-ul Android Studio arată exact ca în Figura 2.3.2.

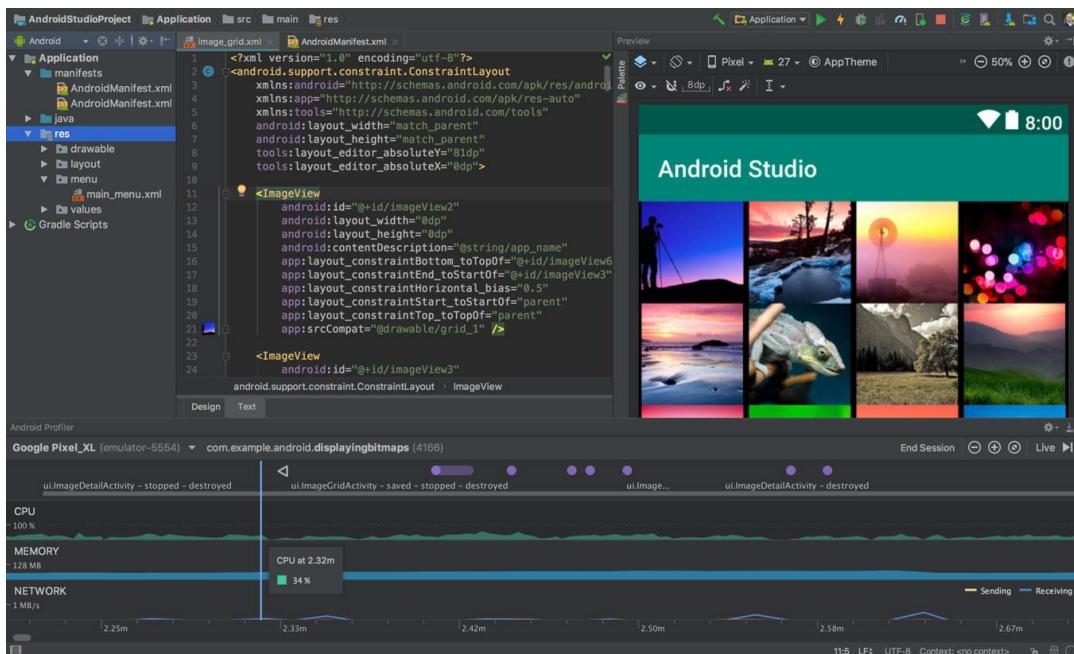


Figura 2.3.2 – Exemplu de cod scris in Androi Studio[14]

În cadrul lucrării, aplicația mobilă a rover-ului a fost dezvoltată cu ajutorul Android Studio. Aplicația mobilă se dorește a fi o interfață prietenoasă și ușor de folosit, cu care utilizatorul să interacționeze cât mai intuitiv. Aceasta oferă posibilitatea de a controla atât rover-ul, cât și brațul de care acesta dispune prin intermediul unor săgeți, dar și de a crea scenarii pe care rover-ul să le

repete ulterior. Totodată, aplicația asigură comunicarea bidirectională cu modulul bluetooth al rover-ului. Mai multe detalii se vor regăsi în capitolul următor.

➤ Fritzing [15]

Fritzing este un program de tip open-source, cu scopul de a oferi dezvoltatorilor de aplicații amatori un mediu pentru a proiecta și a realiza arhitecturi hardware. A fost creat la Universitatea de Științe Aplicate Potsdam. Acest program vine preinstalat cu o multitudine de componente, dar oferă și posibilitatea importării de noi componente create de comunitate.

De la crearea unui model cât mai realistic al proiectului, până la crearea schițelor electrice utilizând simboluri clasice, Fritzing vine cu o multitudine de ușor de folosit. Software-ul este creat în spiritul limbajului de programare Processing și al microcontrolerului Arduino și permite unui proiectant, artist sau cercetător să-și documenteze prototipul bazat pe Arduino și să creeze un layout PCB pentru fabricație.

Un exemplu de circuit simplu creat cu ajutorul Fritzing, se poate observa în Figura 2.3.3.

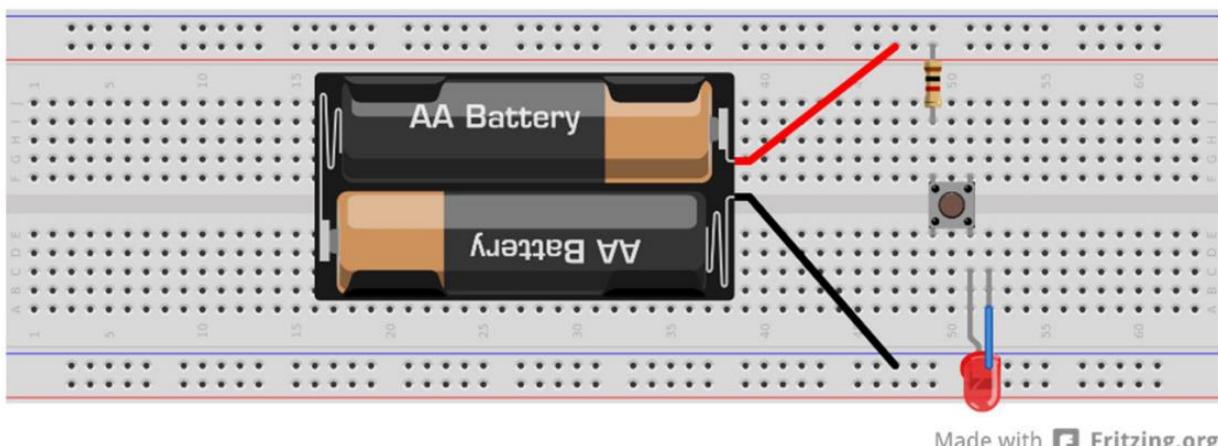


Figura 2.3.3 – Exemplu de circuit realizat in Fritzing[15]

➤ Fusion 360 [16]

Fusion 360 este un instrument CAD / CAM / CAE, bazat pe cloud, pentru dezvoltarea colaborativă a produsului. Aceasta combină modelarea organică, rapidă și ușoară, cu modelarea solidă precisă, ajutând astfel la creația unor modele fabricabile.

Fusion 360 conectează întregul proces de dezvoltare a produsului într-o singură platformă bazată pe cloud pentru Mac și PC. Explorează și rafinează forma designului utilizatorului cu instrumentele de sculptură, modelare și design generativ.

Deoarece proiectele din Fusion 360 sunt stocate și partajate cu mai mulți utilizatori dintr-o eventuală echipă în cloud, se pot itera idei de design în timp real, ceea ce crește productivitatea echipei. Se poate optimiza și valida designul creat cu ansambluri, studii comune și de mișcare și simulări.

În cele din urmă, se poate imprima un prototip rapid al designului creat, pe o imprimantă 3D sau se pot genera căi de instrumente CAM pentru mașini CNC pentru a fabrica designul.

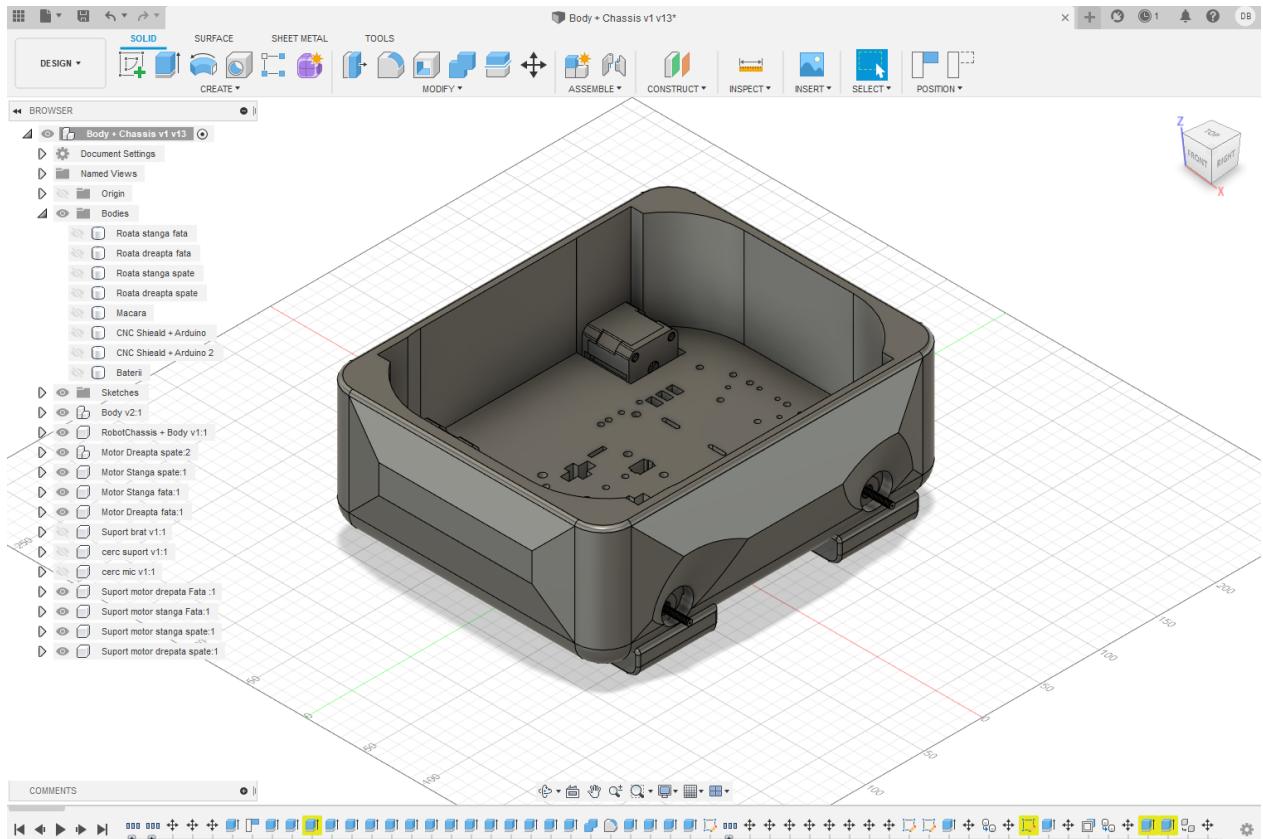


Figura 2.3.4 – O versiune a carcasei roverului

În cadrul proiectului, Fusion 360 m-a ajutat în procesul de modelare 3D al carcasei rover-ului. Interfața acestui program, și totodată o versiune inițială a carcasei, se pot observa în Figura 2.3.4.

3. Implementarea sistemului

Sistemul prezentat în capitolul 2, ale cărui componente au fost evidențiate în schema informațională din figura 2.1.1, este împărțit la nivel abstract în 2 module. Modulul 1 este responsabil de comunicarea de tip master-slave dintre plăcile Arduino, iar modulul 2 este responsabil de comunicarea utilizatorului cu rover-ul și implicit cu microcontrolerele prin interfață aplicației mobile.

Mai departe, voi prezenta cele 2 module mai detaliat.

3.1. Modul 1: Comunicarea master-slave

Modulul de comunicare master-slave, prezentat în figura 3.1.1 are 3 componente (submodule):

- Placa Arduino - „Slave Roti”, este cea care execută comenziile primite pentru mișcarea roțiilor
- Placa Arduino – „Slave Braț”, este cea care execută comenziile primite pentru mișcarea brațului
- Placa Arduino – „Master Senzori”, este cea care interpretează comenziile primite de la aplicația mobilă și le trimită mai departe modulului căreia i se adresează; totodată este modulul responsabil de citirea senzorilor

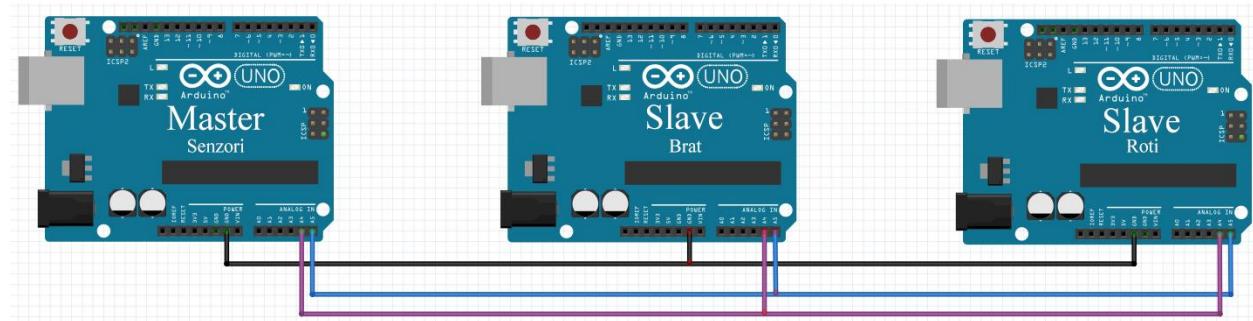


Figura 3.1.1 – Comunicare master-slave

Master - slave este un model de comunicare sau control asimetric în care un dispozitiv sau proces „master” controlează unul sau mai multe alte dispozitive sau procese „slavii” și servește ca hub de comunicație al acestora. În unele sisteme, un master este selectat dintr-un grup de dispozitive eligibile, celelalte dispozitive acționând în rolul de sclavi. [4]

Acest model de comunicare folosește un protocol de comunicare / interfață numit I²C.

Interfața I²C sau IIC (*Inter-Integrated Circuit*) este un tip de magistrală pentru transmisie de date serială master-slave, utilizată intensiv între circuite integrate digitale (microcontrolere, memorii, convertoare) și a fost inițial dezvoltată de către firma olandeză Philips (în prezent NXP) în anul 1982. [4]

Prin această soluție, s-a redus numărul de linii la două (SDA – date și SCL – ceas). Popular, această interfață este cunoscută și sub numele de „comunicare pe două fire” sau „two wire interface”.

I2C presupune :

- Rata de transfer: 10 Kb/s – 100Kb/s
- SDA – Serial Data line
- SCL – Serial Clock line
- 128 de adrese posibile
- 16 adrese rezervate
- 112 dispozitive se pot conecta pe I2C
- Dispozitivele conectate trebuie să aibă masa comună (GND)

În continuare, voi prezenta funcționalitățile fiecărei componente individual.

➤ Master Senzori

Acest submodul, are o arhitectură identică cu cea din figura 3.1.2 și are în componență sa 4 senzori ultrasonici HC-SR04 și un modul bluetooth HC-05.

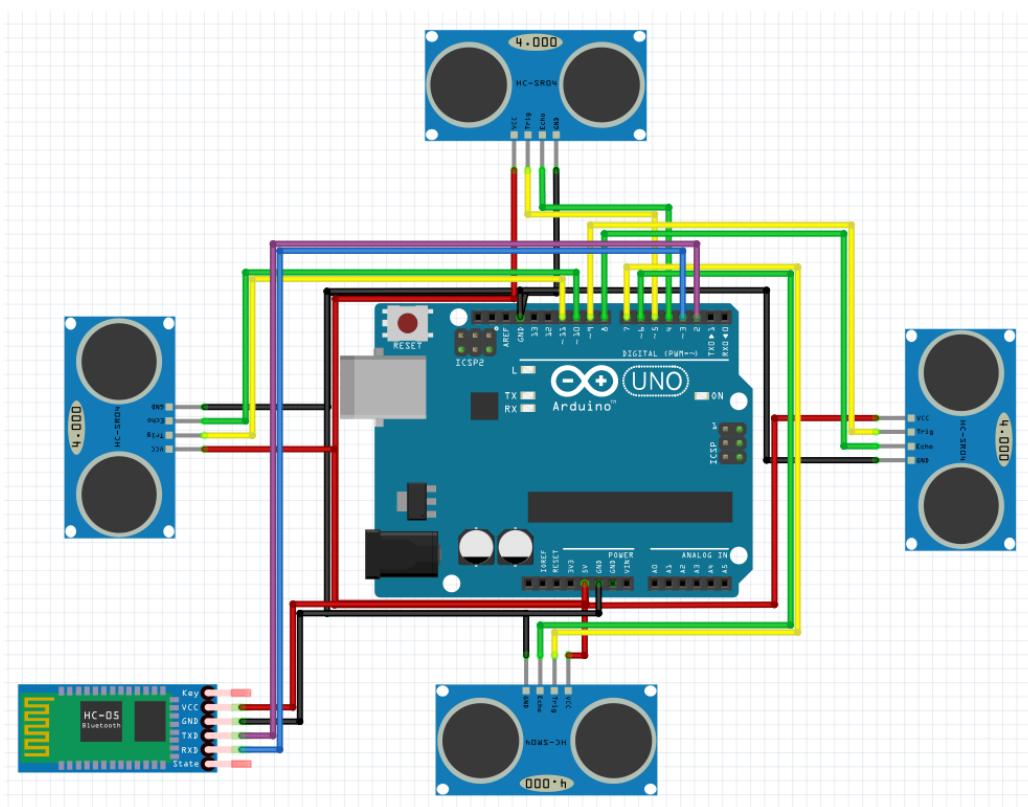


Figura 3.1.2 – Schema electrică a submodulului Master-Senzori

Componenta „Master Senzori” este responsabilă pentru interpretarea comenzielor primite prin serial de la modulul bluetooth. Aceasta folosește un algoritm proiectat astfel încât să satisfacă doar o singură comandă la un moment dat.

Procesul care are loc în momentul pornirii sistemului este reprezentat în organograma funcțională din figura 3.1.3

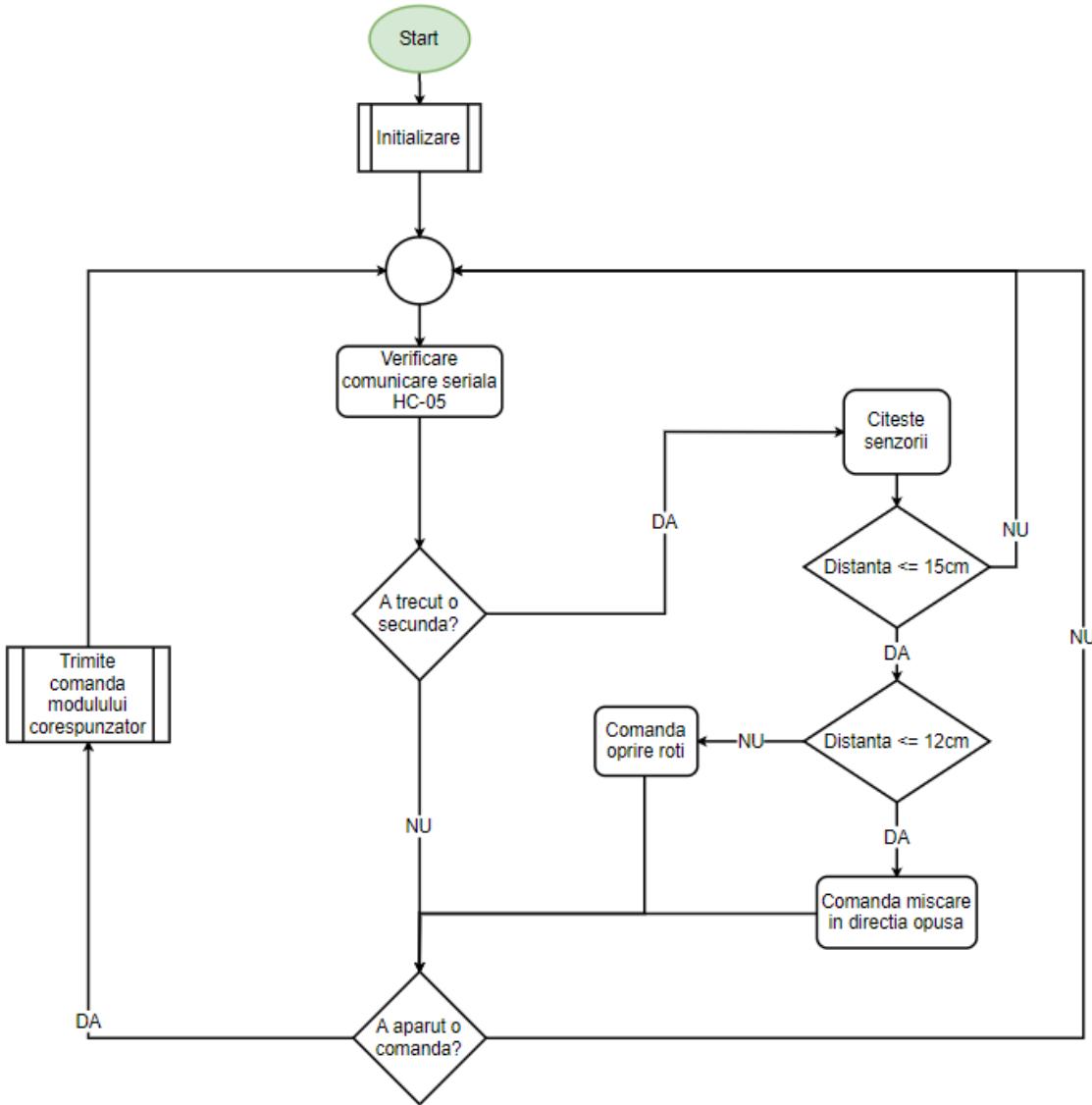


Figura 3.1.3 – Organograma funcțională a submodulului Master-Senzori

Algoritmul gestionează atât traficul de date dintre Arduino și modulul bluetooth, cât și transmiterea datelor, respectiv comenziilor, mai departe, următoarelor module, cu scopul de a fi executate (vezi figura 2.2.1). Totodată, algoritmul este capabil să transmită datele citite de la fiecare senzor ultrasonic în parte, către aplicația mobilă, prin intermediul modulului bluetooth. În același timp, datele citite de la senzori sunt interpretate de către microcontroler. Astfel, dacă robotul se apropie de un obstacol și acesta ajunge la o distanță de 15cm, microcontrolerul trimite o comandă spre modulul responsabil de controlul roților și oprește rotația acestora, iar dacă în continuare obstacolul se apropie și distanța scade la 12cm, microcontrolerul trimite comanda de a se deplasa în direcția opusă.

➤ Slave Braț

Arhitectura sistemului de control al brațului poate fi observată în figura 3.1.4. Aceasta cuprinde o placă Arduino, peste care este poziționată o placă de expansiune de tip „CNC Shield V3”, 3 drive pentru controlul a 3 motoare bipolare pas cu pas corespondente a 3 axe și un servo motor.

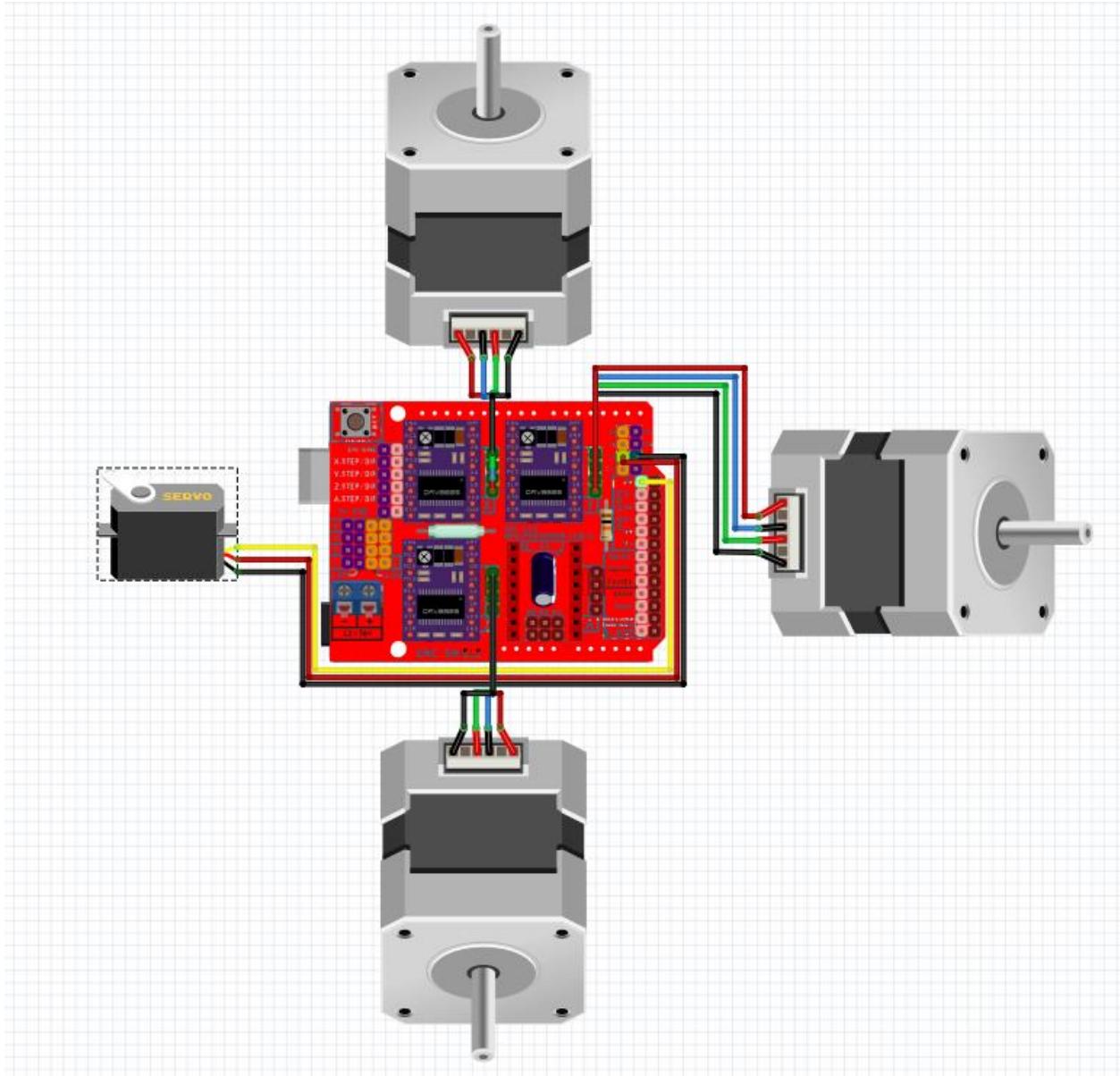


Figura 3.1.4 – Schema electrică a submodulului Slave-Braț

Submodul „Slave Braț” se ocupă cu executarea comenzielor primite prin serial, folosind ca și protocol de comunicare I2C bus. Algoritmul folosit pe această componentă este proiectat astfel încât să execute câte o comandă pe rând, dar totodată o comandă să poată fi întreruptă de o alta.

Organograma algoritmului care gestionează primirea comenzi și executarea acestora, poate fi observată în figura 3.1.5.

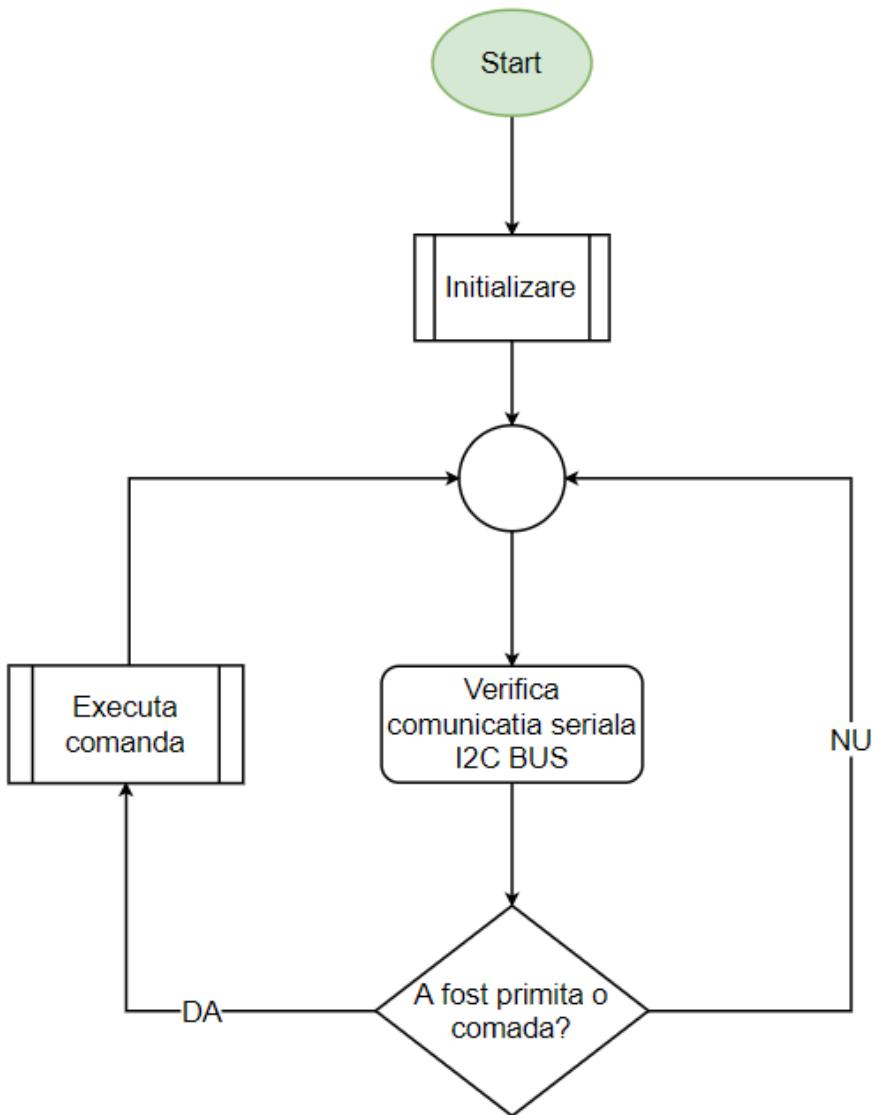


Figura 3.1.5 – Organograma funcțională de primire și executare a comenzi

Acest algoritm, prin executarea comenzi, asigură capacitatea brațului de a se deplasa pe cele 3 axe de mișcare X, Y și Z. Totodată, această mișcare este una obturată de niște limite impuse, neexistând posibilitatea ca brațul să se distrugă de unul singur în momentul deplasării. Aceste limite sunt concepute să se autoregleze în timpul deplasării pe axe, în funcție de punctul în care se află brațul în momentul respectiv, luând în calcul faptul că brațul are întotdeauna un punct „0” de plecare, la fiecare repornire a sistemului.

➤ Slave Roti

Schema din figura 3.1.6 reprezintă arhitectura submodulului responsabil de controlul roților. Acesta are în componență să o placă Arduino, peste care este poziționată o placă de expansiune de tip „CNC Shield V3”, 4 drivere pentru controlul a 4 motoare bipolare pas cu pas, corespondente a celor 4 roți ale roverului.

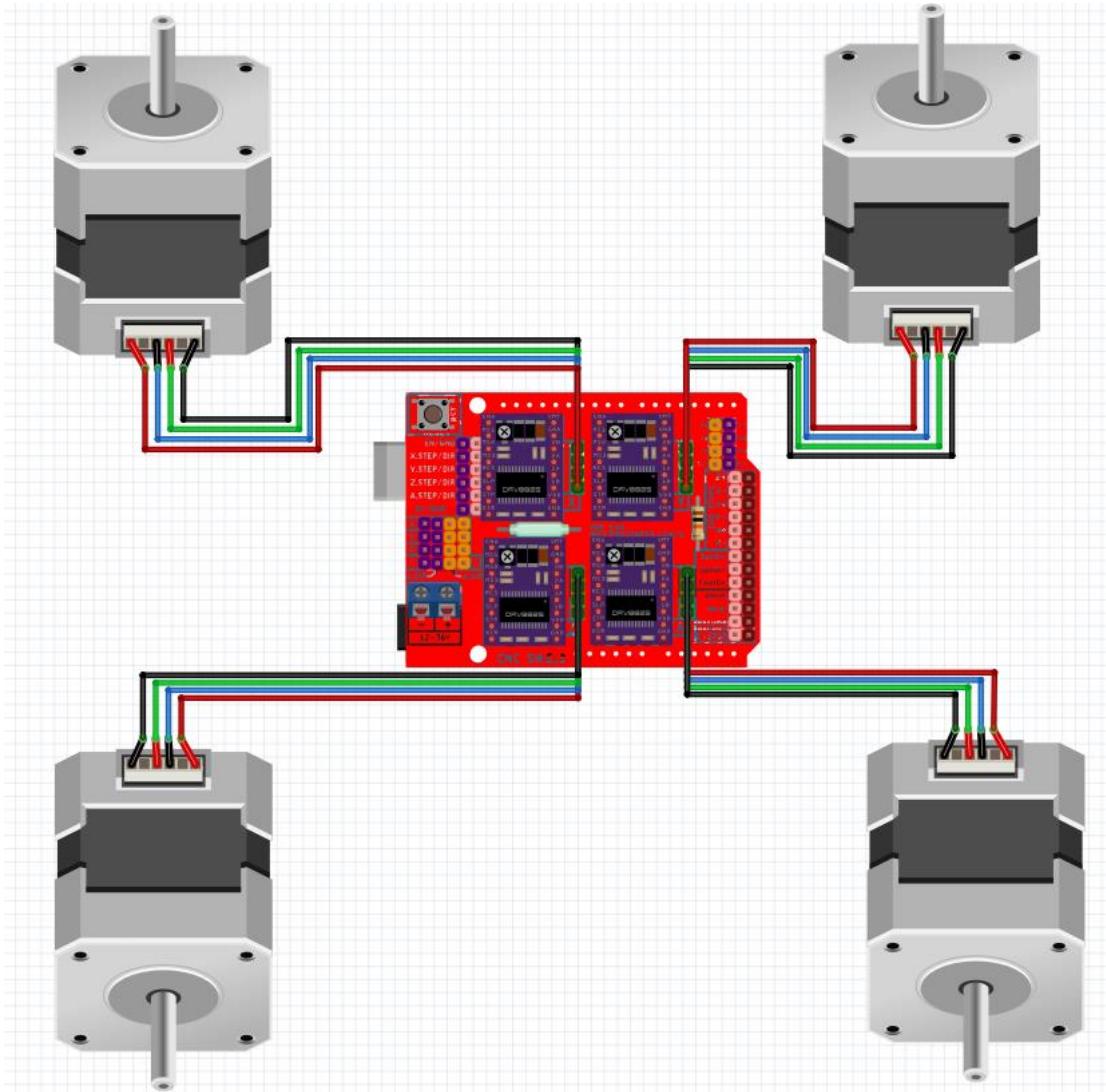


Figura 3.1.6 – Schema electrică a submodulului Slave-Roti

Submodulul „Slave Roți”, exact ca și componenta prezentată anterior, se ocupă cu executarea comenzielor primite prin serial, folosind ca și protocol de comunicare I2C bus. Algoritmul folosit pe această componentă este proiectat astfel încât să execute câte o comandă pe rând, dar totodată o comandă să poată fi întreruptă de o alta.

Organograma algoritmului care gestionează primirea comenzielor și executarea acestora, este identică cu cea a componentei anterioare și poate fi observată în figura 3.1.5.

Algoritmul utilizat pe această componentă a fost conceput astfel încât rover-ul să se poată deplasa cu ajutorul roților montate pe acesta. Algoritmul convertește comenziile primite în rotații relevante ale motoarelor.

Pentru o deplasare înainte, respectiv înapoi, algoritmul învârte toate cele 4 roți înainte, respectiv înapoi. Pentru o rotație pe loc la dreapta, algoritmul învârte roțile de pe partea dreaptă înapoi și roțile de pe partea stânga înainte, iar pentru o rotație la stânga, în sens contrar.

3.2. Modul 2: Comunicarea device mobil - microcontroler

Modulul reprezentat în figura 3.2.1 are la bază 3 componente principale:

- Device-ul mobil cu bluetooth
- Modulul Bluetooth HC-05
- Placa de dezvoltare Arduino

Modulul bluetooth realizează conexiunea dintre device-ul mobil și Arduino. Comunicația fiind una bidirectională, aceasta se desfășoară în felul următor:

- **Comunicația device mobil → Arduino:** are ca și scop transmiterea comenziilor spre microcontrolerul modulului „Master Senzori”, care le transmite mai departe componentei corespunzătoare prin protocolul I2C. Dispozitivul mobil fiind conectat la modulul HC-05, transmite informațiile wireless către acesta, care le transmite mai departe către Arduino, prin intermediul comunicației seriale.
- **Comunicația Arduino → device mobil:** are ca și scop transmiterea în timp real a valorilor citite de către microcontrolerul Arduino de la senzorii ultrasonici HC-SR04. Arduino transmite valorile senzorilor prin comunicația serială, către modulul bluetooth, care le transmite mai departe device-ului mobil, în mod wireless.

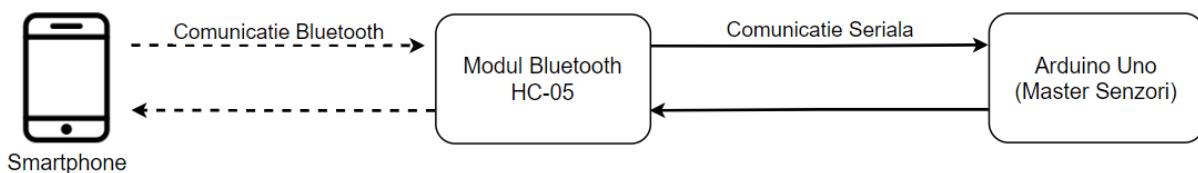


Figura 3.2.1 – Schema bloc modul 2

Ca și interfață cu utilizatorul, prin care să se asigure comunicarea de date între dispozitivul mobil și modulul Bluetooth, a fost realizată o aplicație mobilă. Pentru ca aplicația să poată fi instalată și să funcționeze în parametrii optimi, este necesar un dispozitiv mobil care dispune de Android ca și sistem de operare, cu o versiune minimă „Android Jelly Bean” (API 16) și un modul bluetooth, incorporat cu o versiune mai mare sau egală cu Bluetooth 2.0.

Comunicarea device mobil → microcontroler :

Prima funcție a aplicației mobile este stabilirea conexiunii Bluetooth dintre device-ul mobil și modulul HC-05. În urma verificărilor efectuate în schimbul de date dintre mobil și modul, se stabilește conexiunea bluetooth, urmată de mesajul validator „Connected to device”. În cazul în care conexiunea eșuează, se va afișa un mesaj de eroare, cu textul erorii întâlnite. Organograma procesului este prezentată în figura 3.2.2.

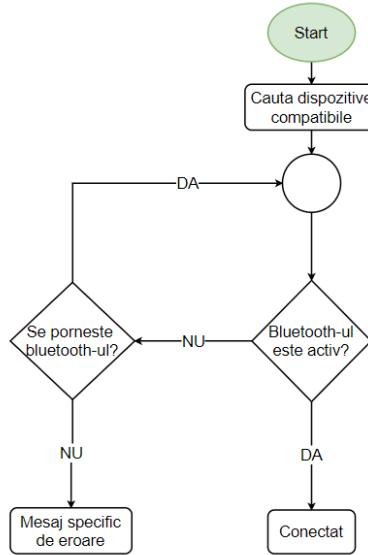


Figura 3.2.2 – Organograma funcțională de conectare bluetooth

Principala funcție a aplicației mobile este controlul în timp real al întregului rover. Comenzi pentru controlul roverului se dau prin apăsarea unor săgeți sau simboluri așezate intuitiv pe ecran.

În urma apăsării uneia dintre săgeți, se apelează algoritmul de transmiterea a comenzi. Acesta poate fi observat în organograma funcțională a procesului de execuție a comenziilor din figura 3.2.3.

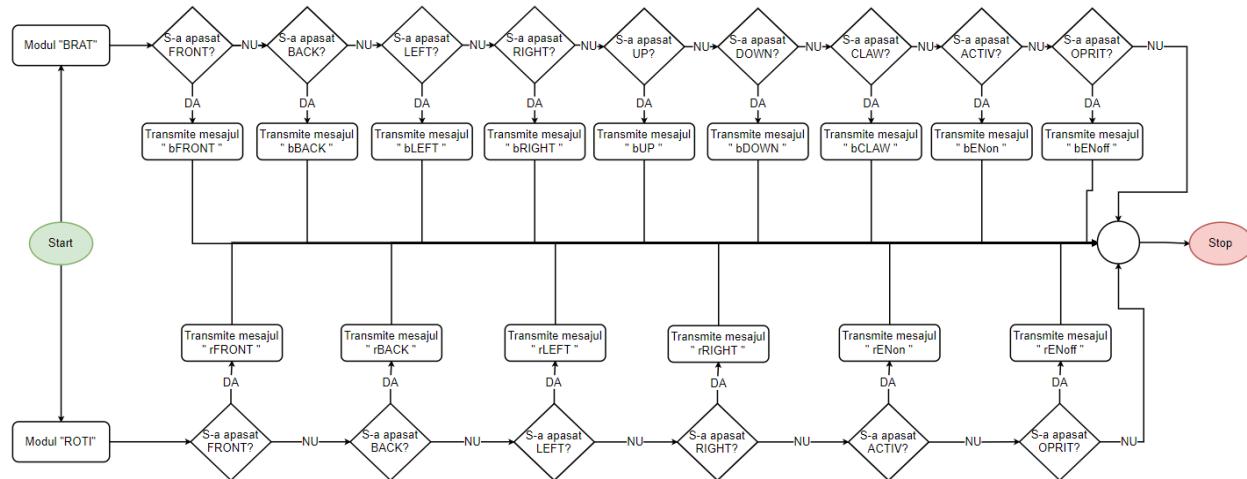


Figura 3.2.3 – Organograma funcțională a procesului de transmitere a comenziilor

Algoritmul de transmitere a comenziilor constă într-o clasa RoverCommands, care poate fi observată în figura 3.2.4, și o funcție care transmite comanda prin intermediul bluetooth la modulul „Master Senzori” pentru a fi interpretată și transmisă mai departe prin comunicația serială la submodulul corespunzător.



Figura 3.2.4 – Clasa RoverCommands

În fiecare atribut al clasei RoverCommands, este stocată câte o comandă de tip „String”. Aceste comenzi stocate, în funcție de denumirea fiecăreia, decid distanța de deplasare a rover-ului pe axă, respectiv direcția dorită. Valorile distanței parcurse la o comandă dată fiecărui motor, pot fi modificate prin intermediul interfeței „Setări”, care se poate accesa prin intermediul pictogramei aflate în figura 3.2.5, din meniul principal. Meniul principal poate fi observat în figura 4.1.6, din capitolul următor.



Figura 3.2.5 – Pictograma Setări

Interfața „Setări” îți pune la dispoziție 3 tipuri de distanțe pentru deplasarea pe fiecare axă, sau în fiecare direcție: Fullstep, Microstep și Picostep. Fiecare dintre aceste setări corespunde unei distanțe în centimetri, și sunt distanțe divizibile cu 16 (1cm – 16cm – 32cm). Prin apăsarea butonului „Apply”, aflat în partea de jos a ecranului, toate atributele din clasa RoverCommands vor fi suprascrisе cu noile setări.

O altă funcție importantă a aplicației mobile, este crearea de scenarii. Pentru a intra în meniul de creare a scenariilor, trebuie să fie accesată pictograma din partea dreaptă-sus a ecranului, regăsită și în figura 3.2.6.



Figura 3.2.6 – Pictograma Scenario

În acest meniu, ai la dispoziție 4 butoane, cu ajutorul cărora poți crea și reda 4 scenarii diferite. Scenariile create sunt o succesiune de comenzi consecutive, care se dă rover-ului, salvate într-un fișier text, în memoria telefonului mobil. Ce stă la baza capacitatei rover-ului de a reproduce comenzi, este, de fapt, un algoritm care citește din fișierul text câte o comandă, pe rând, și o transmite mai departe prin intermediul bluetooth.

În figura 3.2.7, este reprezentată organograma procesului de funcționare a unui buton din interfața „Scenario”. În schemă se poate observa atât procesul de execuție, cât și procesul de creare a unui scenariu.

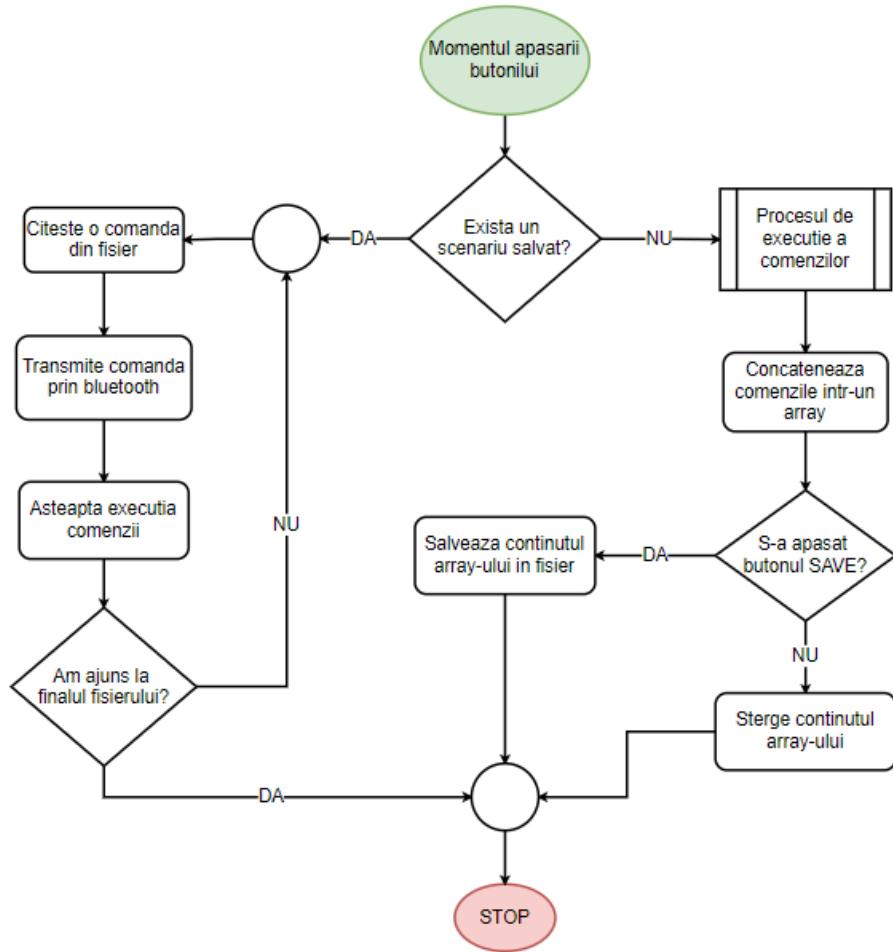


Figura 3.2.7 – Organograma funcțională a procesului de funcționare a unui buton din interfața „Scenario”

Deoarece în momentul terminării execuției unei comenzi nu am un răspuns validator din partea modulului responsabil cu executarea acesteia, partea de execuție succesiva a comenziilor citite din fișier, am implementat-o astfel.

După citirea și transmiterea fiecărei comenzi, am blocat thread-ul de execuție a algoritmului, un anumit timp necesar pentru executarea comenzi efective. Acest timp a fost dedus în mod experimental și se poate regăsi în tabelul din figura 3.2.8

Brat					
Comanda (S*1)	Tim(s)	Comanda (S*16)	Tim(s)	Comanda (S*32)	Tim(s)
bUP1	1s	bUP16	2,5s	bUP32	4,2s
bDOWN1	1s	bDOWN16	2,5s	bDOWN32	4,2s
bFRONT1	1s	bFRONT16	2,5s	bFRONT32	4,2s
bBACK1	1s	bBACK16	2,5s	bBACK32	4,2s
bRIGHT1	1s	bRIGHT16	2,5s	bRIGHT32	4,5s
bLEFT1	1s	bLEFT16	2,5s	bLEFT32	4,5s
bCLAW	1s	bCLAW	1s	bCLAW	1s
bENon	1s	bENon	1s	bENon	1s
bENoff	1s	bENoff	1s	bENoff	1s

Roti					
Comanda (S*1)	Tim(s)	Comanda (S*16)	Tim(s)	Comanda (S*32)	Tim(s)
rFront1	1s	rFront16	2,5s-3s	rFront32	5s
rBACK1	1s	rBACK16	2,5s-3s	rBACK32	5s
rLEFT1	1s	rLEFT16	4,5s	rLEFT32	8,5s
rRIGHT1	1s	rRIGHT16	4,5s	rRIGHT32	8,5s
rENon	1s	rENon	1s	rENon	1s
rENoff	1s	rENoff	1s	rENoff	1s

Figura 3.2.8 – Tabelul cu timpul de execuție al fiecărei comenzi

Comunicarea microcontroler → device mobil :

Ultima funcție de care aplicația mobilă dispune, este aceea de a primi informații în timp real, de la modulul „Master Senzori”. Aceste informații constau în valorile citite de la senzorii aflați în jurul rover-ului, cât și poziția fiecărui senzor în parte (POZITIE : valoare).

Interfața poate fi accesată prin intermediul pictogramei aflate în centru-sus al ecranului. Aceasta dispune de un bloc text, în care se vor afișa informațiile primite, de un buton „Read”, care odată apăsat va porni citirea informațiilor, și un buton „Clear Input”, la apăsarea căruia se va șterge informația aflată în momentul respectiv în blocul de text (vezi figura 4.1.13).

4. Utilizarea sistemului

Pentru a putea utiliza sistemul, este necesară instalarea fișier de tip APK(Android Package), care conține aplicația mobilă. Android Package este formatul standard de comprimare al fișierelor folosite și recunoscute de sistemul Android, dar și de alte sisteme de operare care au la bază Android pentru distribuirea și instalarea de aplicații mobile și jocuri.

Icoană aplicației este prezentată în figura 4.1.1

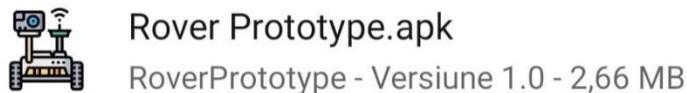


Figura 4.1.1 – Icoană aplicației

Odată instalată aplicația, este necesar să va asigurați că butonul de ON/OFF al sistemului fizic, aflat în partea din spate, este în poziția ON, astfel încât acesta să fie alimentat de baterie, și poziția brațului se află în poziția „0”. Poziția „0” a brațului este reprezentată în figura 4.1.2.



Figura 4.1.2 – Brațul în poziția „0”

În urma instalării aplicației mobile, aceasta poate fi identificată în sistem după numele „Rover Prototype”. După instalare și accesare, aceasta se va deschide în activitatea de conectare la rover.

Conexiunea bluetooth:

Interfața de conectare cu sistemul fizic conține un buton de „Search”, un buton de „Connect” și o listă (figura 4.1.3). Pentru a putea efectua conexiunea, este necesară apăsarea butonului „Search”, din interfață. Această comandă, dacă nu este deja activat, va cere dreptul de a activa bluetooth-ul device-ului, sub formă de pop-up window, exact ca în figura 4.1.4 și figura 4.1.5.

Este necesar să permiteți ca bluetooth-ul dumneavoastră să fie activat sau să îl activați dumneavoastră, pentru a putea crea o conexiune.



Figura 4.1.3 – Fereastra conectare Bluetooth

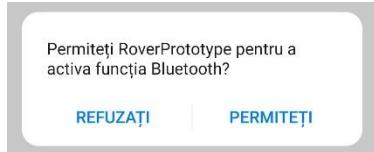


Figura 4.1.4 – Interogare conectare



Figura 4.1.5 – Fereastră conectare

După ce ați activat bluetooth-ul și ați apăsat pe butonul „Search”, veți dispune de o listă de dispozitive bluetooth, care se află în jurul dumneavoastră. Pentru a putea efectua conexiunea cu rover-ul, trebuie să selectați dispozitivul cu numele „HC-05”, care este numele modulului bluetooth de care dispune roverul, și, apăsând butonul „Connect”, veți efectua conexiunea.

Panoul de control:

Odată efectuată conexiunea, aplicația mobilă se deschide direct în meniul principal, unde se află și panoul de control al rover-ului (figurile 4.1.6 și 4.1.7).

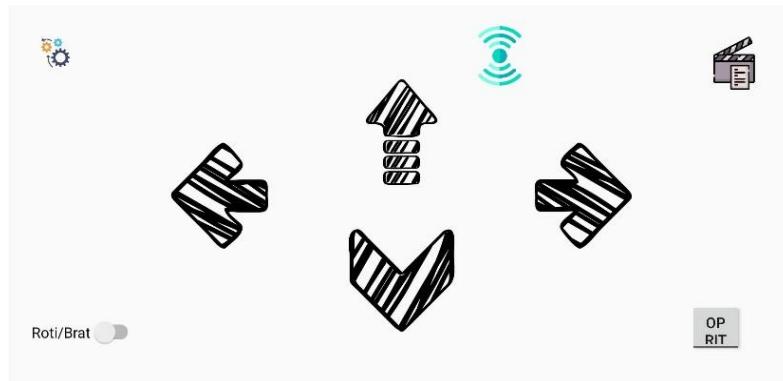


Figura 4.1.6 – Panoul de control al roților

Panoul de control al roverului este împărțit în 2 panouri separate, cel care se ocupă cu controlul roților (figura 4.1.6) și cel care se ocupă cu controlul brațului (figura 4.1.7). Între aceste 2 panouri se poate naviga cu ușurință folosind butonul de tip „switch” din partea stânga-jos, care este marcat intuitiv cu numele „Roti/Brat”.

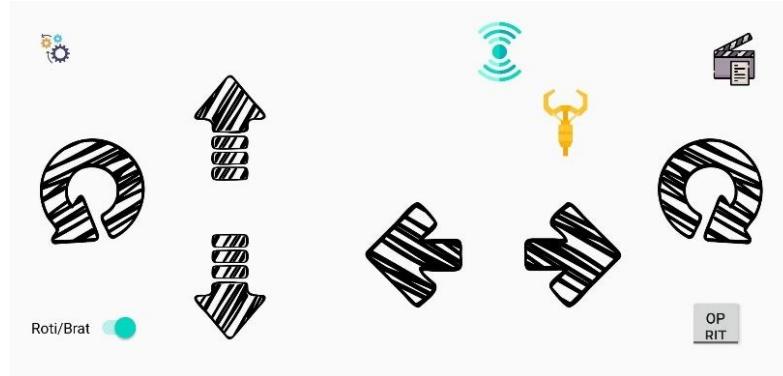


Figura 4.1.7 – Panoul de control al brațului

Pentru ca rover-ul să fie capabil să execute comenziile, trebuie alimentate motoarele acestuia. Puteți face acest lucru acționând butonul „OPRIT/PORNIT” din dreapta-jos a meniului principal. După ce acesta se află în poziția „PORNIT”, se pot folosi săgețile pentru controlul roților sau a brațului.

Customizarea distanțelor de deplasare:

Pentru a modifica/customiza distanțele pe care le parurge rover-ul, se poate accesa butonul „Setări”, aflat în stânga-sus, în meniul principal și având o pictogramă intuitivă.

Odată accesată, la fel ca și la partea de control al rover-ului, veți putea naviga între setările brațului și ale roților, printr-un buton de tip „switch”.

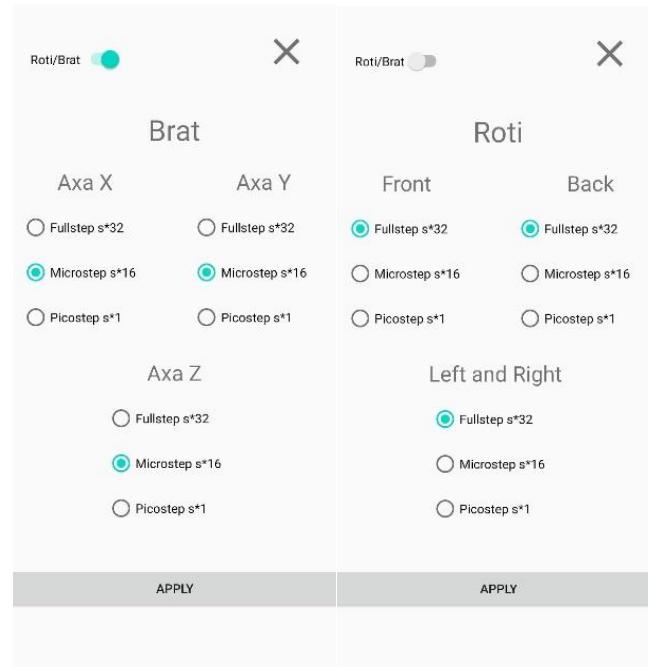


Figura 4.1.8 – Setări braț Figura 4.1.9 – Setări Roți

Mișcarea diferitelor motoare se pot customiza prin setarea a 3 niveluri de precizie: Fullstep, Microstep și Picostep. Acești pași înseamnă diferite lucruri pentru fiecare din componente.

Pentru „Roți”, aceste niveluri de precizie se referă la distanța efectivă pe care rover-ul o înaintează într-o direcție, 1cm – 16cm – 32cm.

Pentru „Braț”, aceste niveluri de precizie se referă la numărul de pași pe care îl efectuează brațul din numărul total de pași, sau, în caști pași împarte distanța maximă pe care o poate parurge.

În cazul brațului, un Fullstep reprezintă distanța maximă pe care brațul o poate parurge, Microstep reprezintă jumătate, iar Picostep reprezintă 1/32 din distanța maximă.

Crearea scenariilor:

Pentru a crea un scenariu, pe care rover-ul este capabil să îl reproducă, se poate accesa interfața „Scenario”, prin apăsarea pictogramei din dreapta-sus din meniul principal.

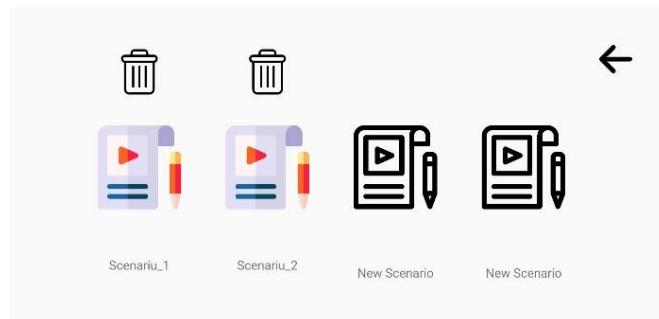


Figura 4.1.10 – Meniul Scenario

O dată accesată, interfața dispune de 4 butoane în care se pot salva 4 scenarii diferite. Scenariile deja salvate pot fi șterse cu ajutorul pictogramei „Trash”, din dreptul fiecărui buton. Pentru a reveni la meniul principal, trebuie accesată pictograma „arrow-back” din dreapta-sus.

Pentru a crea un scenariu, se poate accesa butonul de tip „New Scenario”, în care nu există salvat deja un scenariu. Prin accesarea acestui buton, se lansează o nouă interfață de control, ca în figura 4.1.11 și figura 4.1.12.

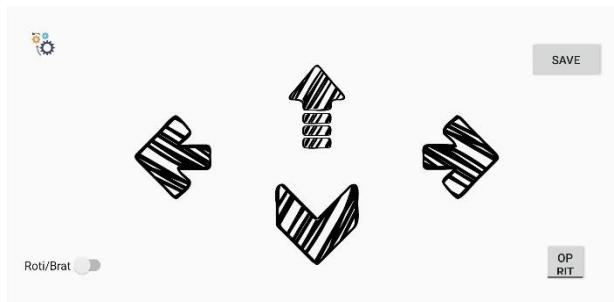


Figura 4.1.11 – Control roti Scenario

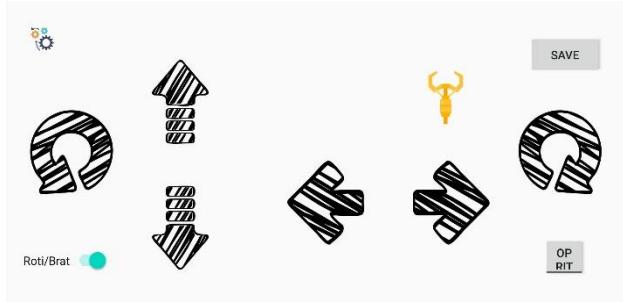


Figura 4.1.12 – Control braț Scenario

În acest panou de control, puteți crea scenariul pe care doriți ca rover-ul să îl execute autonom, prin ghidarea acestuia, folosind săgețile și pictogramele de care dispune interfața.

La finalul scenariului, pentru ca acesta să se salveze, trebuie să apăsați pe butonul „Save”, aflat în partea dreapta-sus a interfeței. Butonul „Save”, odată apăsat, închide panoul de control și sunteți redirectionat în interfață „Scenario” inițială, dar cu posibilitatea de a reda scenariul creat anterior.

Scenariul nu are o limită de comenzi pe care le suportă, astfel încât poate fi cât de lung se dorește.

Citirea senzorilor:

Interfața de citire a senzorilor poate fi accesată prin intermediul pictogramei din partea de sus-centru a meniului principal. Aceasta dispune de un bloc text, în care se vor afișa informațiile primite, de un buton „Read”, care odată apăsat va porni citirea informațiilor, și un buton „Clear Input”, la apăsarea căruia se va șterge informația aflată în momentul respectiv în blocul de text.

După cum poate fi observat în figura 4.1.13, informațiile venite de la senzori în timp real sunt sub forma „POZITIE : valoare”.

Senzorii folosiți au o capacitate de acțiune cuprinsă între 2cm și 2m, asta însemnând că dacă distanța nu se află în acest interval, se va afișa textul „Out of range”.

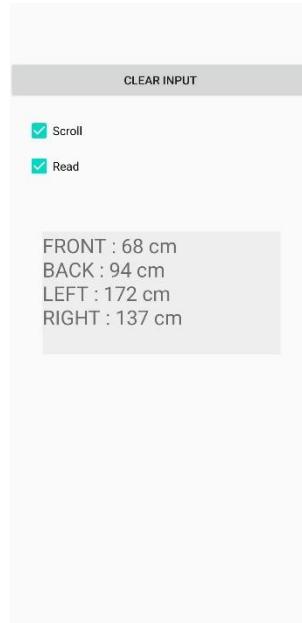


Figura 4.1.13 – *Interfața de citire a senzorilor*

5. Concluzii

Am ales această tematică pur și simplu din dorința de a crea un robot. Inițial, nu aveam idee ce fel de robot să construiesc, dar după câteva cercetări, am ales 3 caracteristici pe care aş dori să le aibă robotul meu. Aceste 3 caracteristici fiind: să se poată deplasa pe roți, să dispună în partea superioară de un braț dotat cu un clește și să poată fi controlat de la distanță.

Primul pas în construcția robotului pe care am decis să îl fac în acel moment, a fost să găsesc un model 3D al unui braț robotic, care să semene cât mai mult cu cea ce îmi imaginam eu. După câteva zeci de modele, m-am hotărât la unul, acesta numindu-se „MK2 Plus”, de pe site-ul „<https://www.thingiverse.com/>”. Acest model de braț robotic are un design creat pentru fi acționat de motoare pas cu pas, pentru axele X,Y,Z și un servo motor pentru controlul cleștelui. După ce am printat și asamblat componentele brațului cu ajutorul unor șuruburi, împreună cu componente hardware necesare, a urmat o perioadă de experimentare, în care am observat mișcarea brațului, utilizând algoritmi pentru acționarea motoarelor. În urma diferitelor rezultate obținute, s-a realizat un algoritm care impune niște limitări în rotația motoarelor, făcând dependentă axa X de Y și invers, astfel încât brațul să nu se autodistrugă.

O problemă a apărut în momentul fixării motoarelor pe platforma brațului și cuplarea acestora la angrenajul de roți dințate. Deoarece axul motorului, pentru a angrena roata dințată care mișcă brațul, are nevoie de un adaptor dințat, care intră pe ax, acesta trebuia lipit de ax. În momentul rotației motoarelor, se mai întâmplă uneori să aibă scăpări (se învârte în gol) din cauza lipiciului, care nu reușește să reziste.

Un alt pas important în construcția robotului a fost crearea carcasei. După căutări îndelungate ale unui model 3D pentru carcăsă, am renunțat și am decis să îl creez singur. Pentru crearea carcasei robotului, am utilizat Fusion 360, care este un program de modelare 3D creat de Autodesk, fiind recomandat de un prieten. După săptămâni de tutoriale și încercări, am ajuns la o variantă de carcăsă care să mă mulțumească. Am calculat-o astfel încât să îmi încapă în ea toate componente hardware necesare pentru funcționarea robotului. Carcasa dispune de un pilon exact în centrul de greutate al acesteia, pe care se montează brațul robotic.

O altă problema a apărut atunci când trebuit să printez carcăsa, deoarece aceasta era prea mare, comparativ cu ce dimensiuni maxime putea să printeze imprimata 3D. Decizia din direcția rezolvării problemei a fost să secționez carcăsa în 4 bucăți egale, suficient de mici, ca să poată fi printate, iar ulterior să se poată îmbina ca și piesele de lego.

După printarea carcasei, integrarea componentelor hardware, atașarea roților și asamblarea întregului sistem, a urmat, din nou, o serie de experimente pentru conceperea algoritmului. Din cauza unor experimente eşuate, am reușit să îmi ard laptopul și câteva componente esențiale. În urma experimentelor, algoritmul a fost optimizat, astfel încât să se opreasca la o anumită distanță de un obstacol, iar dacă obstacolul se apropie, roverul să se deplaseze în direcția opusă.

O problema care și-a făcut apariția în urma unor experimente de deplasare a rover-ului, a fost schimbarea direcției de mers a acestuia. Rover-ul a fost gândit ca având 4 roți, comandate 2 câte 2, pentru a lua virajul, să fie comandat să învârtă 2 roți (de pe partea stânga) într-o direcție și 2 roți

(de pe partea dreaptă) în celalaltă direcție. Din păcate, din cauza materialului și construcției roților, acest lucru se poate întâmpla doar în niște condiții speciale. Deoarece roțile sunt dintr-un cauciuc moale, rover-ul poate vira doar pe suprafețe cu un coeficient mare de alunecare.

Următorul și ultimul pas în terminarea proiectului, a fost dezvoltarea aplicației mobile. Aceasta a fost dezvoltată pe platforma celor de la Google, numită Android Studio, folosind limbajul Java. Aplicația a fost gândită să disponă de o interfață intuitivă cu utilizatorul, rover-ul fiind controlat cu ajutorul unor săgeți.

Robotul rezultat în urma construcției nu este unul perfect, dar reflectă cunoștințele deja deținute și cele dobândite pe parcursul realizării acestuia. Părerea mea este că rover-ul poate fi îmbunătățit din mai multe puncte de vedere.

Una dintre îmbunătățirile care se pot aduce rover-ului ar fi înlocuirea unor componente din braț cu unele puțin mai lungi. Aceasta ar fi o îmbunătățire deoarece, în configurația actuală, roverul nu poate ridica obiecte aflate pe sol în fața acestuia, iar cu un braț mai lung, s-ar putea face acest lucru.

O altă îmbunătățire a brațului rover-ului ar putea fi înlocuirea cleștelui cu unul mai mare. Brațul, având cleștele actual, este capabil să apuce doar obiecte de mici dimensiuni. Cu un design diferit al cleștelui, acesta ar putea fi mult mai util.

Consider că înlocuirea roților actuale ale rover-ului, cu unele care au un design mult mai inovator, ar fi o ameliorare a eficienței acestuia. Acest tip de roți se numește „Roți Mecanum”, iar ele pot fi observate în figura 5.1.1.



Figura 5.1.1 – Roată mecanum[17]

Figura 5.1.2 – Model de sășiu care utilizează roți mecanum[17]

Aceste roți, prin construcția lor, sunt mult mai eficiente în luarea virajelor decât roțile din cauciuc folosite de mine în proiect. Din cauza construcției și poziționării lor, ca și în figura 5.1.2, ele sunt capabile de o deplasare atât la stânga și la dreapta, cât și de o deplasare spre unghiul de 45° în fiecare direcție, fără ca platforma să își schimbe orientarea.

În concluzie, după o investiție de 1600 de lei, s-a creat un prototip de rover cu braț robotic, controlat printr-o aplicație mobilă.

Bibliografie

- [1] <https://en.wikipedia.org/wiki/Robot>
- [2] <https://www.automate.org/a3-content/joseph-engelberger-unimate>
- [3] <https://en.wikipedia.org/wiki/Microcontroller>
- [4] <https://www.instructables.com/Robot-Arm-MK2-Plus-Stepper-Motor-Used/>
- [5] <https://www.arduino.cc/>
- [6] https://wiki.keyestudio.com/Ks0160_keyestudio_CNC_Shield_V3
- [7] <https://www.ti.com/lit/ds/symlink/drv8825.pdf>
- [8] <https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/>
- [9] <https://www.tme.eu/ro/news/library-articles/page/41861/Motorul-pas-cu-pas-in-salturi-tipuri-i-exemplu-de-utilizare-a-motoarelor-pas-cu-pas/>
- [10] <https://components101.com/motors/mg90s-metal-gear-servo-motor>
- [11] https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf
- [12] <http://users.utcluj.ro/~rdanescu/pmp-lab8-Bluetooth-RO.pdf>
- [13] <https://sites.google.com/site/arduinoelectronicasiprogramare/arduino-si-senzori/1>
- [14] <https://developer.android.com/studio>
- [15] <https://fritzing.org/>
- [16] <https://www.autodesk.com/products/fusion-360>
- [17] https://en.wikipedia.org/wiki/Mecanum_wheel

**DECLARAȚIE DE AUTENTICITATE A
LUCRĂRII DE FINALIZARE A STUDIILOR ***

Subsemnatul BOCIORT DINU-IULIAN

legitimat cu C. i. seria AR nr. 846878,

CNP 1981218020101

autorul lucrării ROVER CU BRAT ROBOTIC CONTROLAT
PRIN APPLICATIE MOBILA

elaborată în vederea susținerii examenului de finalizare a studiilor de

LICENȚA organizat de către Facultatea

AUTOMATICA SI CALCULATOARE din cadrul Universității

Politehnica Timișoara, sesiunea IUNIE a anului universitar

2020-2021, coordonator SL.DR.ING. ANA MARIA DAN, luând în

considerare conținutul art. 34 din *Regulamentul privind organizarea și desfășurarea examenelor de licență/diplomă și disertație*, aprobat prin HS nr. 109/14.05.2020 și cunoscând faptul că în cazul constatării ulterioare a unor declarații false, voi suporta sancțiunea administrativă prevăzută de art. 146 din Legea nr. 1/2011 – legea educației naționale și anume anularea diplomei de studii, declar pe proprie răspundere, că:

- această lucrare este rezultatul propriei activități intelectuale,
- lucrarea nu conține texte, date sau elemente de grafică din alte lucrări sau din alte surse fără ca acestea să nu fie citate, inclusiv situația în care sursa o reprezintă o altă lucrare/alte lucrări ale subsemnatului.
- sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.
- această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență/diplomă/disertație.

Timișoara,

Data

09.06.2021

Semnătura



* Declarația se completează „de mână” și se inserează în lucrarea de finalizare a studiilor, la sfârșitul acesteia, ca parte integrantă.