

# **DESIGN OF WHEG ROBOT FOR MULTI TERRAIN MOBILITY**

## **A PROJECT REPORT**

*Submitted by*

**SANYATJEET PAWDE [RA1711004010048]  
ROHAN TEHALYANI [RA1711004010050]  
MITRANSH CHOUBISA [RA1711004010058]  
MAYANK KUMAR [RA1711004010064]**

*Under the guidance of*

**Dr. ESWARAN P, Ph.D**

(Associate Professor, Department of Electronics and Communication Engineering)

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**ELECTRONICS AND COMMUNICATION ENGINEERING**

*of*

**FACULTY OF ENGINEERING AND TECHNOLOGY**



**SRM**  
INSTITUTE OF SCIENCE & TECHNOLOGY  
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

**JUNE 2020**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that this project report titled “**DESIGN OF WHEG ROBOT FOR MULTI TERRAIN MOBILITY**” is the bonafide work of “**SANYATJEET PAWDE[RA1711004010048], ROHAN TEHALYANI[RA1711004010050], MITRANSH CHOUBISA[RA1711004010058], MAYANK KUMAR [RA1711004010064]**, , who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

Dr. ESWARAN P, Ph.D  
**GUIDE**  
Associate Professor  
Dept. of Electronics and Communi-  
cation Engineering

**SIGNATURE**

Mr. SRINATH B, M.E  
**CO-ORDINATOR**  
Assistant Professor(Sr.G)  
Dept. of Electronics and Communi-  
cation Engineering

## **ABSTRACT**

With the advancements of robotics in different fields, there is an on going need for research and development of different types of robots required at different places. The need of today's robots is to be able to run, climb, jump and avoid obstacles. This gives a lot of scope for different designs of a robot's body in modern robotics. Whegged robots are one such robot specially designed to be capable of climbing and avoiding different hindrances because of their unique design. The goal of our project is to control these specially designed robotic legs with the help of available knowledge of electronics and make it more control friendly. The project aims to look for techniques to control such robots wirelessly over radio frequency, so that such robots can further be used in different fields, be it military use or space missions. Many renowned organisations around the world like MIT and NASA are developing such all-terrain robots for different purposes like research or space missions. Our project targets to design one such miniature Whег robot and establish best mode of communication between robot and control station.

## **ACKNOWLEDGEMENTS**

We would like to express our deepest gratitude to our guide, Dr. Eswaran P and to our co-ordinator Mr. Srinath B for their valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

**Project Group**

# TABLE OF CONTENTS

<b>ABSTRACT</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>ABBREVIATIONS</b>	<b>x</b>
<b>LIST OF SYMBOLS</b>	<b>xi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Whegs . . . . .	1
1.1.1 Biologically inspired mechanisms . . . . .	2
1.1.2 Obstacle dependent speed and phase shift . . . . .	3
1.1.3 Comparison and Deduction of whег mobility by insect gait . . . . .	4
1.2 Objectives . . . . .	5
1.3 Applications . . . . .	6
1.4 Motivation . . . . .	6
<b>2 LITERATURE SURVEY</b>	<b>7</b>
2.1 Comparing Cockroach and Whег Body Motions . . . . .	7
2.2 Speed dependent Phase Shifts and Gait changes in Cockroaches running on substrates of different slipperiness . . . . .	7
2.3 Whег Development for Advanced Mobility . . . . .	8
2.4 Design Analysis of a Remote Controlled Robotic Vehicle . . . . .	8
2.5 Design and Analysis of A Miniature Two-Whег Climbing Robot with Robust Internal and External Transitioning Capabilities . . . . .	8
2.6 WHEGSTR: A Multiterrain Robot with C Shaped Whегs, Implementation of Error Minimization Technique and using Artifical Neural Network (ANN) .	9

2.7	Application of Radio Frequency Controlled Intelligent Military Robot in Defense . . . . .	9
<b>3</b>	<b>System Analysis</b>	<b>10</b>
3.1	Robot's Stability Calculation . . . . .	10
3.1.1	Case 1: Inclination . . . . .	10
3.1.2	Case 2: The change in centre of mass and centre of gravity when legs are changing their inclinations with respect to others . . . . .	11
<b>4</b>	<b>System Design</b>	<b>14</b>
4.1	System Integrated Design . . . . .	14
4.2	Design Specifications . . . . .	17
4.2.1	Servo Motors(MG995) . . . . .	17
4.2.2	DC Motor . . . . .	17
4.2.3	Dual Shaft Motor . . . . .	18
4.2.4	MICROCONTROLLER Specifications . . . . .	19
4.2.5	Transceiver Module Specifications . . . . .	20
4.2.6	Motor Driver IC L293D Specifications . . . . .	21
4.2.7	Controller Console Specifications . . . . .	22
4.2.8	Conceptual Sketches using CAD . . . . .	22
<b>5</b>	<b>Coding</b>	<b>26</b>
5.1	CODE . . . . .	26
5.1.1	Receiving user input from controller and sending it the Arduino . . . . .	26
5.1.2	Serially receiving and decoding the input from the controller and sending it to bot . . . . .	29
5.1.3	Receiving the state and actuating it using servo and dc motors . . . . .	29
<b>6</b>	<b>Conclusion</b>	<b>33</b>
<b>7</b>	<b>Future Enhancement</b>	<b>34</b>
<b>A</b>	<b>Materials</b>	<b>35</b>
A.1	Hardware . . . . .	35
A.2	Software . . . . .	36

## LIST OF TABLES

4.1	3D model Specifications . . . . .	16
-----	-----------------------------------	----

## LIST OF FIGURES

1.1	Obstacle clearence using whegs . . . . .	1
1.2	Cockroach climbing an obsatcle . . . . .	2
1.3	Double tripod gait of a cockroach . . . . .	3
1.4	Speed Dependent phase shift in a cockroach . . . . .	4
1.5	Synchronous Movement of Opposite Legs . . . . .	5
1.6	Single Axle response to an obstacle . . . . .	5
1.7	Overall Framework . . . . .	6
3.1	Robot on an inclination . . . . .	10
3.2	Analogy of chassis body with a square in Cartesian system . . . . .	11
3.3	Analogy of one leg in different position . . . . .	12
4.1	System Integrated Design . . . . .	14
4.2	Conceptual Flowchart . . . . .	15
4.3	Required torque at each joint . . . . .	16
4.4	Prototype Model . . . . .	16
4.5	MG995 motor to be used for movement of joints . . . . .	17
4.6	DC Motor to be used for movement of wheels . . . . .	18
4.7	Dual Shaft Motor to be used for movement of Whegs . . . . .	19
4.8	Arduino Uno . . . . .	19
4.9	nRF TX/RX channels . . . . .	20
4.10	nRF module pinout and arduino connection . . . . .	21
4.11	L293D Motor Driver IC . . . . .	21
4.12	Controller . . . . .	22
4.13	Chassis Body/Electronic Cabinet . . . . .	23
4.14	Leg Design . . . . .	24
4.15	Wheel Dimensions . . . . .	24
4.16	Wheg Dimension . . . . .	24



4.17	Whegs Design with Shaft . . . . .	25
4.18	Final Assembly Sketch . . . . .	25
4.19	Final Assembly Design . . . . .	25
5.1	Code Flowchart . . . . .	27
5.2	code for receiving user input . . . . .	28
5.3	code for sending data to bot . . . . .	30
5.4	Code for motor actuation . . . . .	32
6.1	Different Views Of Design . . . . .	33

## **ABBREVIATIONS**

<b>COM</b>	Centre of Mass
<b>UGV</b>	Unmanned Ground Vehicle
<b>ISR</b>	Intelligence Surveillance and Reconnaissance
<b>COG</b>	Centre of Gravity

## LIST OF SYMBOLS

$F$	Force, N
$T$	Torque ,kgcm
$L$	length, cm
$m$	mass, gm
$g$	acceleration due to gravity, m/s <sup>2</sup>
$\theta$	critical angle(angle of elevation), °degrees
$\mu$	coefficient of friction,dimensionless
$h_{cog}$	height of centre of gravity,cm
$w$	width of vehicle,cm
$a$	side length,cm

# CHAPTER 1

## INTRODUCTION

### 1.1 Whegs

Engineering problems can be solved using biological inspiration in varying degrees, from a direct implementation to an abstracted one. Difficulties arise in the direct implementation of biological attributes because, oftentimes, the necessary technology has yet to be developed. The design and testing of this technology can be a slow process, which ultimately causes its employment to be long-term. However, implementing abstracted biological attributes utilizing existing technology enables mission capable solutions in the near-term.

One such solution is provided by the abstraction of arthropodic movement being implemented in mobile robots known as Whegs(wheel-legs or Wing-legs).Whegs are mechanisms for robot locomotion. Whegs use a strategy of locomotion that combines the simplicity of the wheel with the obstacle-clearing advantages of the foot.

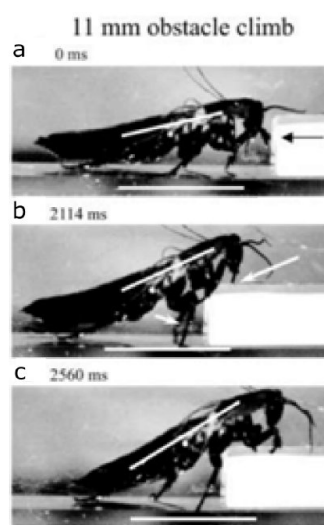


**Figure 1.1:** Obstacle clearence using whegs  
[11]

Whegs were pioneered at the Biologically Inspired Robotics Laboratory at Case Western Reserve University. Whegs development and improvements based on cockroach climbing behavior in several robots has been done in collaboration with the Ritzmann lab in the Biology

department at Case Western Reserve University on cockroach climbing behavior. Whegs robots were inspired by the Prolero robot, designed in 1996 at the European Space Agency, and the RHex robot, developed by a multiuniversity effort funded by the Defense Advanced Research Projects Agency. The mobility system is based on studies on the locomotion of the cockroach. Wing-legs are found on flying robots and are wings dual-purposed as legs for locomotion when the robot is on the ground.

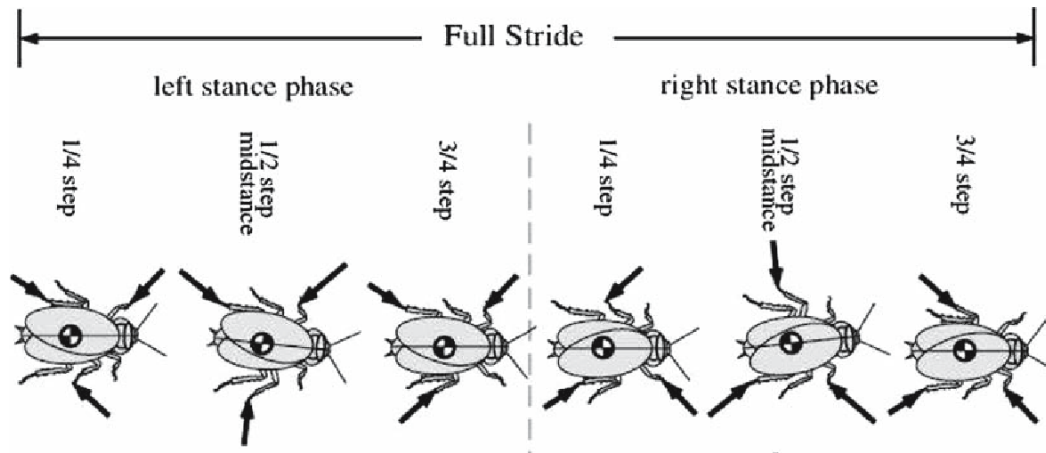
### 1.1.1 Biologically inspired mechanisms



**Figure 1.2:** Cockroach climbing an obstacle  
[9]

Cockroaches have remarkable locomotion abilities. One solution to the problem of producing highly mobile amphibious robots is to design a vehicle with the mechanisms responsible for the mobility of a cockroach. In studies of cockroach movement, we have noted the following locomotion principles. A cockroach has six legs, which support and move its body. It typically walks and runs in a tripod gait where the front and rear legs on one side of the body move in phase with the middle leg on the other side. The front legs swing head-high during normal walking so that many obstacles can be surmounted without significant gait changes. However, its gait changes when it encounters larger barriers.[1]

The cockroach turns by generating asymmetrical motor activity in legs on either side of its body as they extend during stance. A cockroach enhances its climbing abilities by changing its body postures before and during a climb over an obstacle. It uses its middle legs to pitch its



**Figure 1.3:** Double tripod gait of a cockroach  
[7]

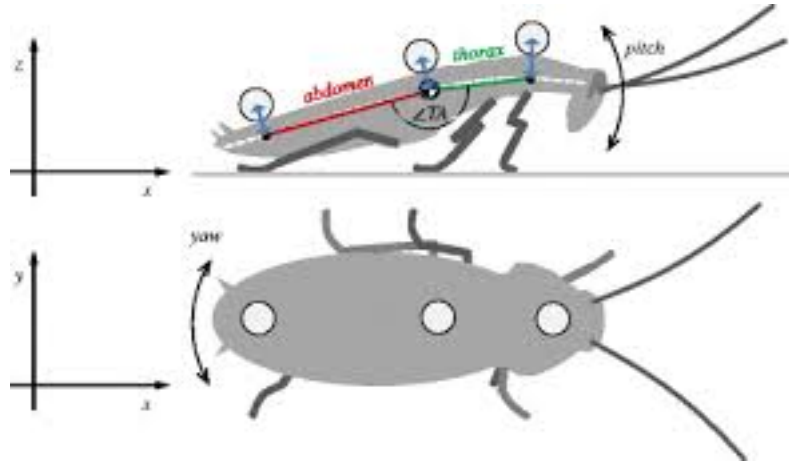
body up, prior to climbing obstacles that are higher than its head. This behaviour enables its front legs to reach higher. Also, during a climb it uses its body flexion joints to bend the front half of its body down to avoid high centering.

A cockroach enhances its climbing abilities by changing its body postures before and during a climb over an obstacle. It uses its middle legs to pitch its body up prior to climbing obstacles that are higher than its head. This behaviour enables its front legs to reach higher. Also, during a climb it uses its body flexion joints to bend the front half of its body down to avoid high centering.

### 1.1.2 Obstacle dependent speed and phase shift

Many legged animals change gaits when increasing speed. In insects, only one gait change has been documented so far, from slow walking to fast running, which is characterised by an alternating tripod. Studies on some fast-running insects suggested a further gait change at higher running speeds. Apart from speed, insect gaits and leg co-ordination have been shown to be influenced by substrate properties, but the detailed effects of speed and substrate on gait changes are still unclear. Here we investigate high-speed locomotion and gait changes of the cockroach *Nauphoeta cinerea*, on two substrates of different slipperiness.

Temporally distributed leg force application as resulting from metachronal leg coordination at high running speeds may be particularly useful in animals with limited capabilities for elastic energy storage within the legs, as energy efficiency can be increased without the need for



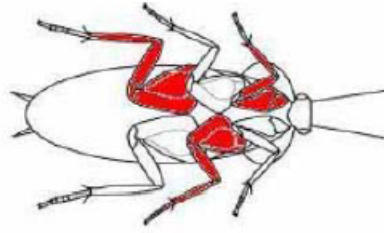
**Figure 1.4:** Speed Dependent phase shift in a cockroach [10]

elasticity in the legs. It may also facilitate locomotion on slippery surfaces, which usually reduce leg force transmission to the ground. Moreover, increased temporal overlap of the stance phases of the legs likely improves locomotion control, which might result in a higher dynamic stability.

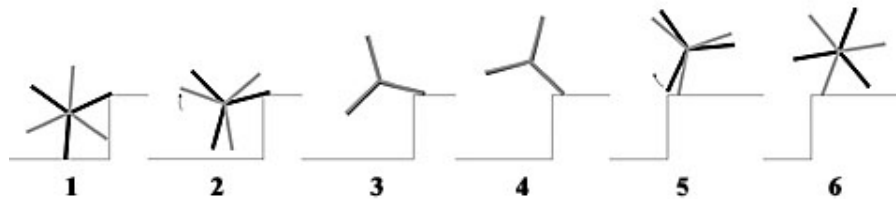
### 1.1.3 Comparison and Deduction of wheg mobility by insect gait

The nominal locomotion pattern found in cockroaches and most other insects is the tripod gait in which the front and rear legs on one side move in approximate synchrony with the middle leg on the opposite side. This tripod alternates with a second tripod made up of the remaining legs. Each leg moves rapidly through the swing phase and then slows down as it enters the stance phase and pushes the body forward. The front legs move through an arc on the order of the animal's height in order to reach the top of small barriers without the need for any change in leg movement. The highlighted legs push forward while the remaining legs are lifted and moving forward preparing for the next step, then the highlighted legs lift and move forward while the remaining legs are now used to push. The Whegs design incorporates the six-legged locomotion (two wheel-legs on each of three axles) as well as compliance in the wheel-legs to enable the platform to passively go in and out of the tripod gait through the use of a modified limited slip differential.

During normal travel, the left and right wheel-legs (shown as black and gray, respectively) are 60 degrees out of phase. As the axle strikes the obstacle, the left wheel-leg is applying



**Figure 1.5:** Synchronous Movement of Opposite Legs  
[8]



**Figure 1.6:** Single Axle response to an obstacle

most of the force to climb up, while at the same time, due to the compliance that is designed in, is rotating until it is in phase with the right wheel-leg and the force of lifting the axle over the obstacle is distributed for even support and maximum lifting power. Once the obstacle is overcome, the left wheel-leg quickly rotates back into phase allowing the platform to resume its tripod gait.

## 1.2 Objectives

To design a wheg and wheel transformable robot

Control it using a RC console

Program it using arduino ide to control the movement of the wheels, joints and whegs

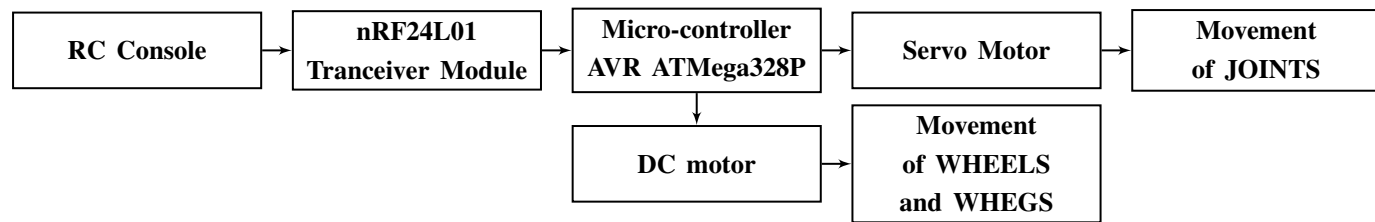
Automate it.

To take inspiration from arthropodic movement and develop a model with similar ability of motion over various terrains.

Thereby making a transformable and convertible robot rather than a normal Whег robot for more mobility and obstacle clearance. and develop a radio frequency based communication system for better accessibility of the robot and to achieve mobility over both rough and



smooth surfaces, i.e., plain terrains, sand dunes, craters, multiterrain.



**Figure 1.7:** Overall Framework

## 1.3 Applications

Some of the fields in which our proposed design would prove to be an efficient aid are

Space Exploration

Archaeological Excavation

Military Reconnaissance

Underwater Exploration

Amphibious Operations

Rescue operations

Biomedical Designs

Automobile Designs

## 1.4 Motivation

To experiment with the already existing technology and try to make it more efficient and feasible for long range and all terrain purposes, such as a better communication system using radio frequency and a wheel cum whег design for better mobility.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 Comparing Cockroach and Wheg Body Motions**

In this paper it is found that abstracting cockroach locomotion principles with reduced actuation can lead to a simple yet effective cockroach-like vehicle able to traverse irregular terrain. One such hexapod robot is Whegs VP, which achieves a cockroach-like nominal tripod gait using only a single DC motor. Whegs VP uses compliant mechanisms in its axles to passively adapt its gait to the terrain such that it can climb obstacles 175 percent of its leg height. High speed video analysis of walking cockroaches and Whegs VP illustrates that Whegs VP walks with cockroach-like body motions. The experiments indicate that stepping patterns play a major role in an animal's or robot's overall body motions.[7]

#### **2.2 Speed dependent Phase Shifts and Gait changes in Cockroaches running on substrates of different slipperiness**

In this paper the analysis of leg co-ordination and body oscillations for straight and steady escape runs revealed that at high speeds, blaberid cockroaches changed from an alternating tripod to a rather metachronal gait, which to our knowledge, has not been described before for terrestrial arthropods. Despite low duty factors, this new gait is characterised by low vertical amplitudes of the Centre of Mass (COM), low vertical accelerations and presumably reduced total vertical peak forces. However, lateral amplitudes and accelerations were higher in the faster gait with reduced leg synchronisation than in the tripod gait with distinct leg synchronisation.[10]

## **2.3 Wheg Development for Advanced Mobility**

There is a significant amount of interest in autonomous, amphibious robots capable of operating in the surf zone. Such robots would be capable of preparing amphibious landing zones by locating, classifying and mapping potential threats to a landing force such as obstacles and mines as well as performing general reconnaissance of the beachhead. In order to accomplish this, the vehicle would have to be rugged, highly mobile and capable of navigation in possibly extreme environments. In order to achieve the level of mobility required, recent research has focused on the use of legged or crawling platforms.[8]

## **2.4 Design Analysis of a Remote Controlled Robotic Vehicle**

In this paper the design of a Remote Controlled Robotic Vehicle has been completed. A prototype was built and confirmed functional. This system would make it easier for man to unrivalled the risk of handling suspicious objects which could be hazardous in its present environment and workplace. Complex and complicated duties would be achieved faster and more accurately with this design.[6]

## **2.5 Design and Analysis of A Miniature Two-Wheg Climbing Robot with Robust Internal and External Transitioning Capabilities**

In this paper the climbing robots have many benefits such as a highly expanded workspace and the ability to reach or accomplish otherwise impossible spots or tasks for ground robots. When climbing robots are used collaboratively with Unmanned Ground Vehicle (UGV) for indoor autonomous Intelligence Surveillance and Reconnaissance (ISR) missions, they can be a powerful swarm that collectively completes tasks such as mapping, detection, monitoring and tracking. Specifically, the climbing robot can be used to provide an overall image of the area and as a bridge for communication for robots on different floors. To complete such tasks, the robot may be required to climb obstacles and transition from one plane to another, both

internally (concave angle) and externally (convex angle).[2]

## **2.6 WHEGSTR: A Multiterrain Robot with C Shaped Whegs, Implementation of Error Minimization Technique and using Artificial Neural Network (ANN)**

From this paper it was deduced that this typical design will provide maneuverability and stability according to the terrain. With its intrinsic mobility, robot will have the ability to adapt according to the terrain and will maintain its balance using algorithms based on differential equations. An unstrapped robot will easily traverse the complexity and diversity of the terrain with compliant legs. The extraordinary nature of the ladder gymnastics brings itself to a solid platform, whereas mountain platforms can be very difficult to stairs. Tracked vehicles must be able to overcome three steps to overcome these obstacles in a stable fashion, and still avoiding due to low traction of steps. Due to its rigid body and strong build, the robot can withstand impacts without compromising its performance. It will also transmit live broadcast of the surrounding area. Moreover, it will give a continuous feedback about the physical conditions i.e. temperature, pressure, altitude, humidity etc. of the nearby area.[3]

## **2.7 Application of Radio Frequency Controlled Intelligent Military Robot in Defense**

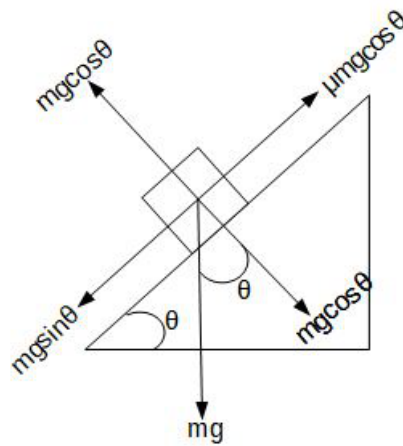
In the present paper the authors tried to explore how a radio frequency controlled robot can be used in defense and in real war field. The military robot will be able to substitute the real human soldier in the battle field. The robot can be controlled from a base station by means of radio frequency. It also has the ability to re-establish contact with the base station in case of a signal failure by retracing its path back for some distance. [5]

## CHAPTER 3

### SYSTEM ANALYSIS

#### 3.1 Robot's Stability Calculation

##### 3.1.1 Case 1: Inclination



**Figure 3.1:** Robot on an inclination

Now robot can either climb an inclination or be moving forward on an inclined plane. In either case there is a chance for it to topple.

So, for the stability of robot we must calculate the critical angle for inclination, i.e., the angle of inclination at which the robot may topple.

The basis of our calculation is **Newton's law of motion**. We are considering the free body diagram of robot with a mass ' $m$ ', and a gravitation acceleration acting on it as ' $g$ '.

From the figure we can see the force  $mg$  acting towards ground and its cosine component acting towards inclination. Now from the figure

$$mg \sin \theta = mg \cos \theta \quad (3.1)$$

$$\tan\theta = \mu \quad (3.2)$$

$$\theta = \tan^{-1}(\mu) \quad (3.3)$$

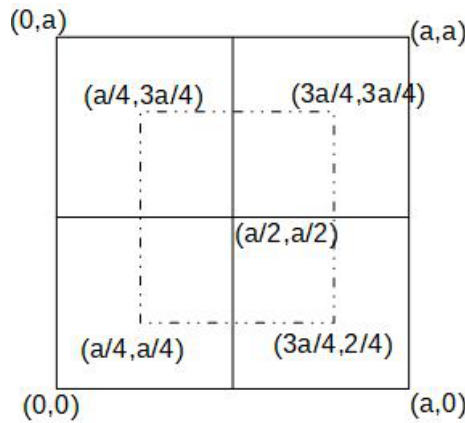
The other case of our robot in which it may topple is when the robot's legs are not parallel to the body. This can be deduced by calculating the height of centre of gravity of the bot. This can be calculated from the critical angle. We are taking the assumption of the static stability factor. **In eq<sup>n</sup>(3.3) we calculated critical angle.**

$$\tan\theta = \frac{w}{2 * h_{cog}} \quad (3.4)$$

$$h_{cog} = \frac{w}{2 * \tan\theta} \quad (3.5)$$

### 3.1.2 Case 2: The change in centre of mass and centre of gravity when legs are changing their inclinations with respect to others

Now this theory is assumed by creating an analogy. We consider the body of the robot, i.e., the robot's chassis to be a square for simplification. Now this square is divided into four more equal squares.

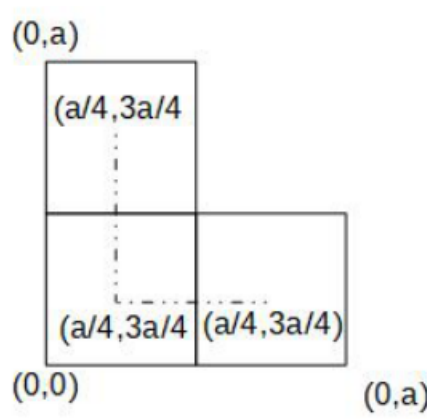


**Figure 3.2:** Analogy of chassis body with a square in Cartesian system

From the figure we can understand how the assumptions are made. The very first analogy is that

of chassis being a square, the second is considering this square to be placed on the Cartesian coordinate system.

Now we are assuming that whenever a leg of body is changing its alignment with respect to other legs than the cog and com of the body will move towards the opposite direction. This theory is based on the fact that centre of gravity of square is at the same place as centre of mass, i.e., in the centre of the square. Thus, if the leg changes direction this will be assumed as a square going missing from the set. Let us see one case.



**Figure 3.3:** Analogy of one leg in different position

Here we are assuming that when one leg is not aligned with other ones than we may remove one square to calculate the shift of centre of mass or centre of gravity, using this calculation we will be able to understand the change in the servo motor's angles.

The calculation is as follows,

$$X = \frac{m * ((a/4) + (a/4) + (3a/4))}{3m} \quad (3.6)$$

$$X = \frac{5a}{12} \quad (3.7)$$

$$Y = \frac{m * ((a/4) + (a/4) + (3a/4))}{3m} \quad (3.8)$$

$$Y = \frac{5a}{12} \quad (3.9)$$

In the above equations,  $m$  is mass of each square body,  $a$  is side length.

From eq (3.7) and (3.8), we see the shift in the Centre of Gravity (COG) and COM of the system. This analogy helps us to understand the shift of balance and mass in the robot as well. If we consider the robot as the same system, then these assumptions help us to understand the shift in angle when the legs are moving independently.

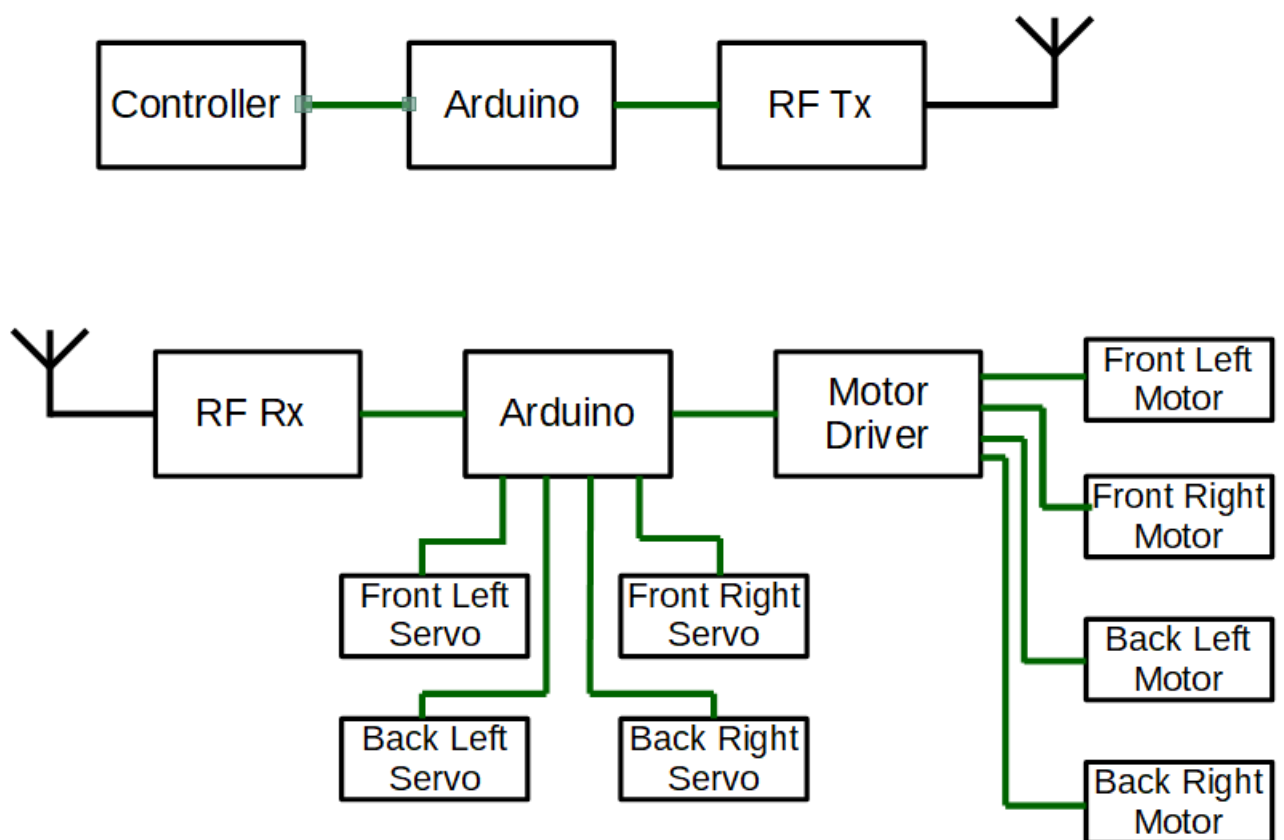
This was one such case where the system is changing, there will be other cases as well but this basis of calculations will easily help us to deduce a system for them as well.



## CHAPTER 4

### SYSTEM DESIGN

#### 4.1 System Integrated Design



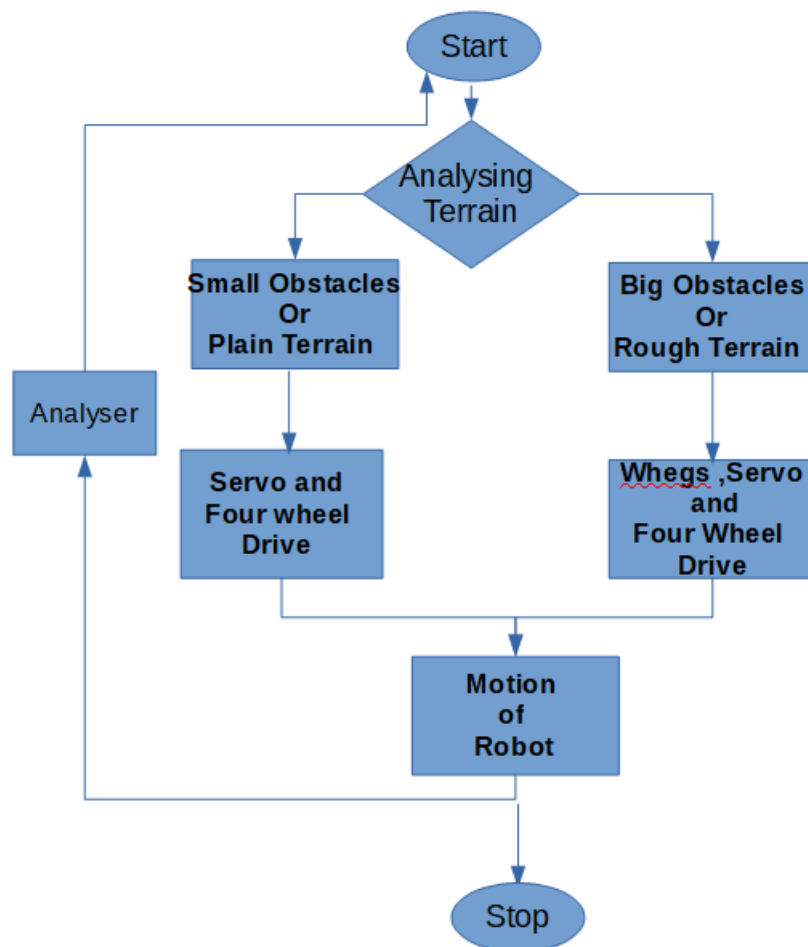
**Figure 4.1:** System Integrated Design

By considering the several approaches, to avoid many technical challenges such that of a shaft and axle design which was inefficient over inclines as the whole chassis of the robot would topple[4], we have proposed a conceptualised integrated design of both wheels and whegs. The robot would be controlled by a simple gaming console by the user which is feasible while using radio frequencies to communicate with the bot. Our design incorporates a chassis which would act as a cabinet to house the electronic circuitry and components.

The whole body of the bot would be made of **Polyacrylonitrile** also known as Acrylic material which is light and durable polymer, to reinforce the body we would add sheets of **Aluminium** metal. The legs would be joined to the servo motors on four corners of the chassis and subsequently the wheels would be joined to the legs via DC motors. The wheels would be integrated at the **Thoracic** region of the bot as compared to cockroach and their motion would be controlled by DC motors.

To enable a long range and less attenuated communication channel between user console and the bot we would use **nRF** module, which is Radio Frequency Transceiver.

By the aforementioned concepts we would be able to overcome the technical challenges in already existing technologies.



**Figure 4.2:** Conceptual Flowchart

Table 4.1: 3D model Specifications

Name of the part	Motion	Dimensions
Leg	extension/linear	15*3cm
Body(Base)		18*12cm
Body(Top plate)		18*4cm
Wheels	rotation	Diameter: 4cm
Whegs	revolute(swoop)	Diameter:3.5cm

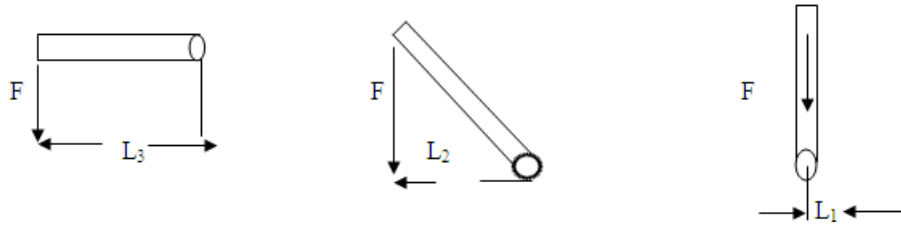


Figure 4.3: Required torque at each joint

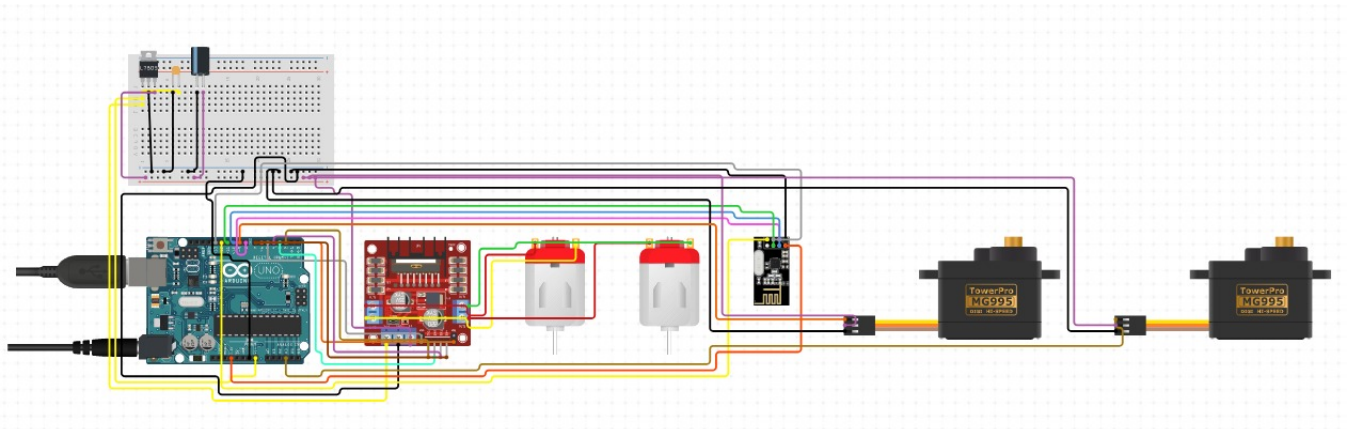
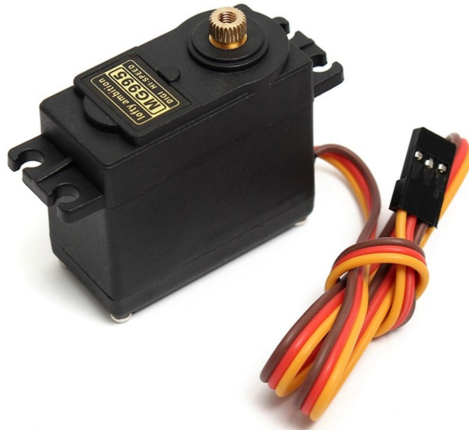


Figure 4.4: Prototype Model

## 4.2 Design Specifications

### 4.2.1 Servo Motors(MG995)



**Figure 4.5:** MG995 motor to be used for movement of joints

1. **Size** 40.7 x 19.7 x 42.9mm
2. **Weight** 55g (Include a cable and a connector)
3. **Speed** 0.2sec/60° (4.8V) 0.16sec/60° (6.0V)
4. **Torque** 9.4kgf-cm (4.8V) 11kgf-cm (6.0V)
5. **Voltage range** 4.8V-7.2V
6. **Operating Temperature** -30° to 60° C
7. **Rotational Degree** 180°
8. **Dead Band Width** 1 micro-sec
9. **Connector type** JR type '(Yellow: Signal, Red: VCC, Brown:GND)'

### 4.2.2 DC Motor

1. **Gearbox Diameter** 37mm
2. **Motor Diameter** 25mm



**Figure 4.6:** DC Motor to be used for movement of wheels

3. **Motor Length** 75mm
4. **Shaft Length** 22mm
5. **Shaft Diameter** 06mm
6. **Weight** 80g
7. **Rated RPM** 300
8. **Stall Torque** 3.5kgf-cm
9. **Rated Torque** 1.2kgf-cm
10. **Voltage range** 4-12V(DC)
11. **Load Current** 0.3A
12. **No load current** 60mA

#### **4.2.3 Dual Shaft Motor**

1. **Operating Voltage** 3-12V(DC)
2. **Rated Speed** 100RPM
3. **Rated Torque** 0.35kg-cm
4. **Shaft Length** 8.5mm
5. **Shaft Diameter** 5.5mm
6. **No Load Current** 40-180mA



**Figure 4.7:** Dual Shaft Motor to be used for movement of Whegs

#### 4.2.4 MICROCONTROLLER Specifications

##### **MICROCONTROLLER- Arduino UNO board with 8 bit ATmega328p AVR**

1. **Power** Vin,3.3V,5V,GND pins for power I/O. operating voltage of 5V and recommended input of 7-12 volts. 50mA current is drawn.
2. **I/O pins** digital pins 0-13, analog pins(A0-A5 0-5V),PWM pins(3,5,6,9,11) for 8 bits pwm I/O.
3. **Serial pins** Rx(0);Tx(1); for transmission and reception of TTL serial data
4. **Microcontroller** ATmega 328p AVR 8 bit microcontroller high performance ,low power controller from Microchip previously manufactured by atmel.
5. **CPU** AVR CPU - Advanced Virtual RISC CPU RISC - reduced instruction set computer Alf-Egil Bogen Vegard Wollan RISC .



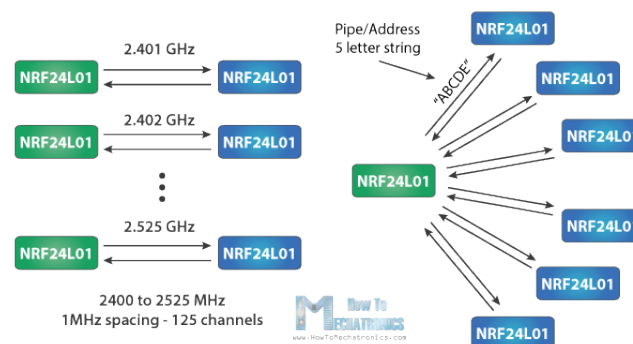
**Figure 4.8:** Arduino Uno

6. **Crystal Oscillator** used for square pulse generation as clock with 16MHz frequency. as it takes (1/16micro seconds) for 1 state
7. **EEPROM** 1Kb
8. **SRAM** 2Kb
9. **Flash Memory** 32Kb for storage of input code

## 4.2.5 Transceiver Module Specifications

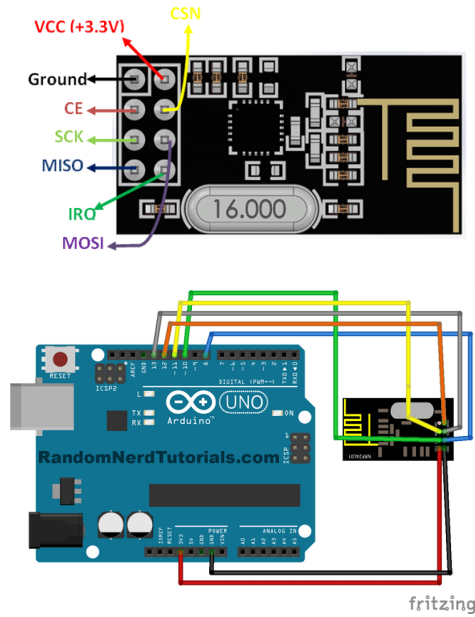
### nRF24L01 Module

1. **Power** (Vcc) Powers the module. Connect to +5V Supply voltage
2. **TX** Transmitter Transmits Serial Data.
3. **RX** Receiver Receive Serial Data.
4. **Operating Voltage:** 3.3V
5. **ChannelRange** :125
6. **Maximum pipeline/node:** 6



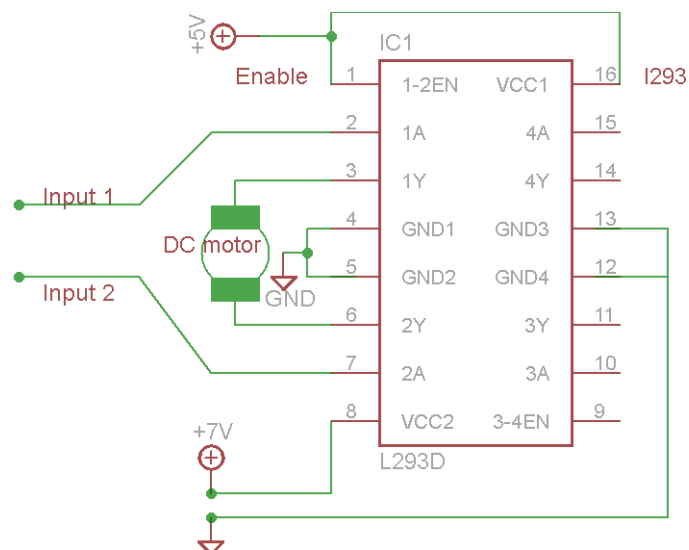
**Figure 4.9:** nRF TX/RX channels

7. **Frequency:**2.4GHz ISM band
8. **Operating Current:** 50mA-250mA(maximum)
9. **Range:** 50-200feet
10. Works with Serial communication (USART) and TTL compatible
11. SPI communication protocol.
12. Can operate in Master, Slave or Master/Slave mode
13. Supported **baud rate:** 250kbps to 2mbps.



**Figure 4.10:** nRF module pinout and arduino connection

#### 4.2.6 Motor Driver IC L293D Specifications



**Figure 4.11:** L293D Motor Driver IC

1. **Supply voltage,  $V_{CC1}$  36V(MAX)**
2. **Output supply voltage,  $V_{CC2}$  36V(MAX)**
3. **Input voltage,  $V_I$  7V(MAX)**
4. **Output voltage,  $V_O$  -3 to  $V_{CC2} + 3$  V**
5. **Peak output current,  $I_O$  (nonrepetitive,  $t < 100\text{microsec}$ ) -1.2 to 1.2 A**



6. Continuous output current,  $I_O$ : L293D -600 to 600mA
7. Maximum junction temperature,  $T_J$  150°
8. Storage temperature,  $T_{stg}$  -65° to 150°C

#### 4.2.7 Controller Console Specifications

##### Logitech F310 Gamepad

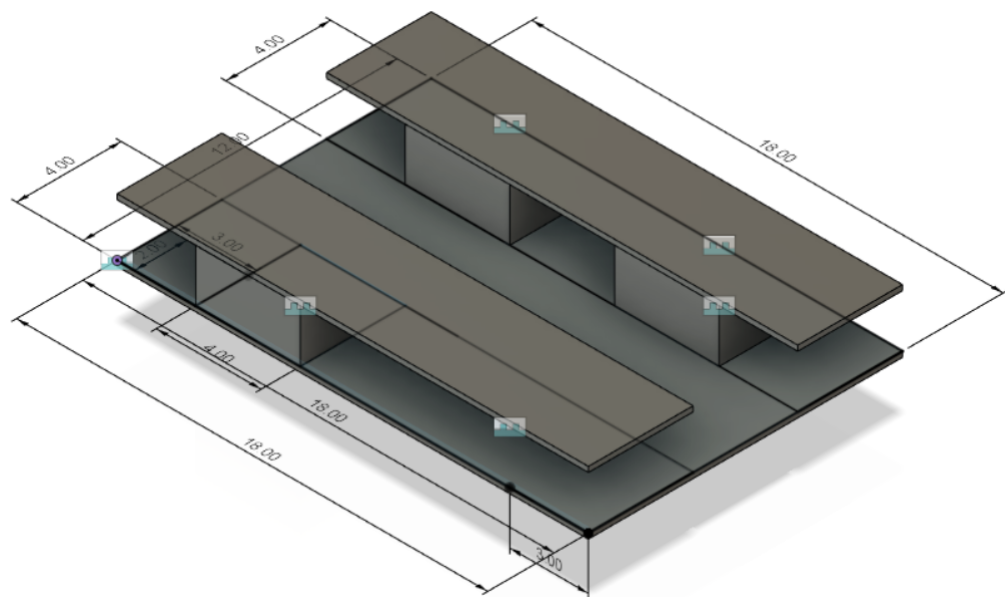
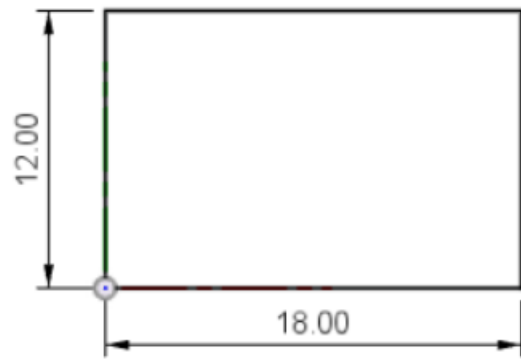
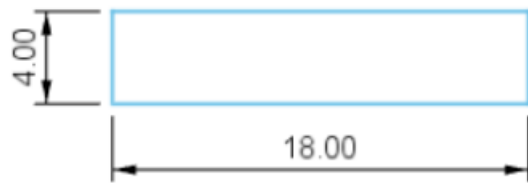


**Figure 4.12:** Controller

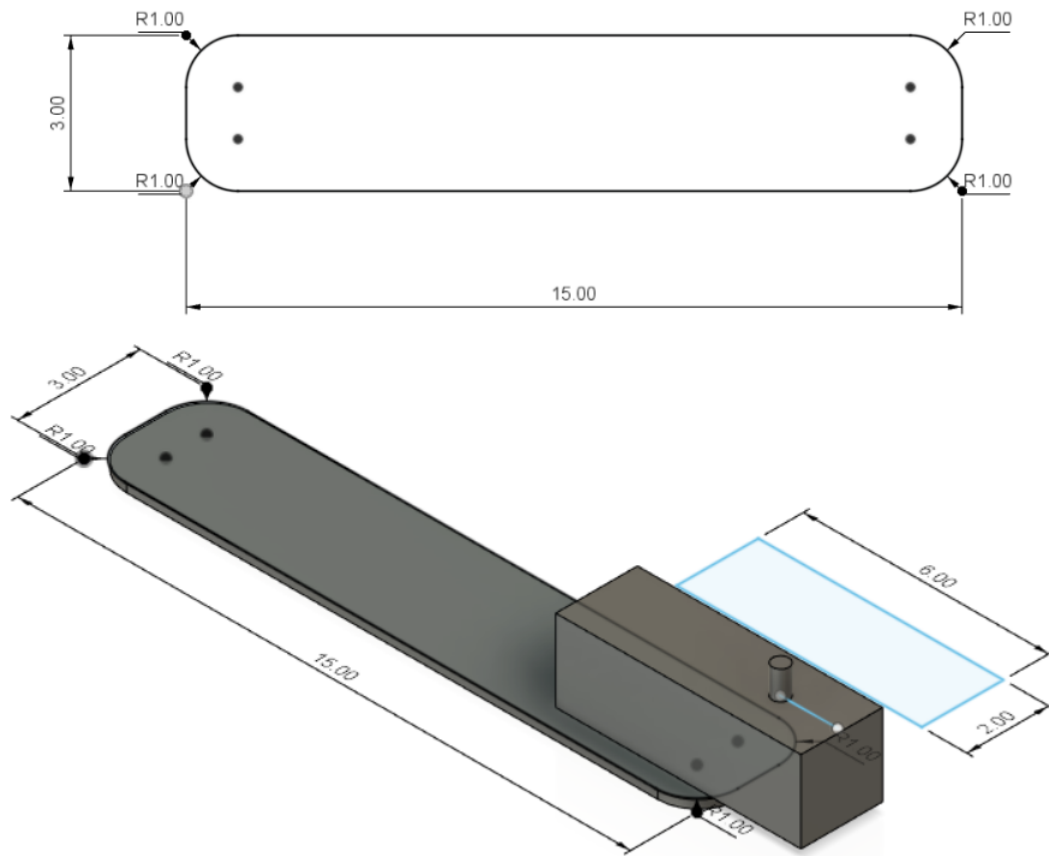
1. 8-Way Programmable D-pad
2. Two Programmable Analog Mini-Joysticks
3. 10-Programmable Buttons
4. Programmable Left and Right Analog Triggers
5. Product Dimensions 17.2 x 7.5 x 20.5 cm
6. Batteries: 2 AAA batteries required.
7. Item Weight 295 g

#### 4.2.8 Conceptual Sketches using CAD

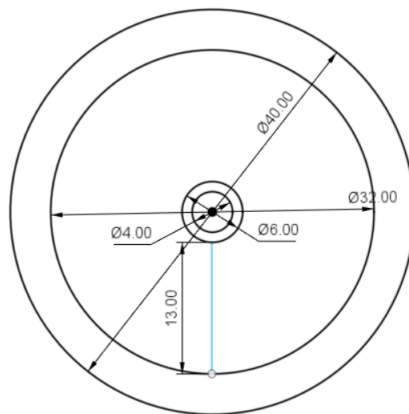
These sketches are made using **Fusion360** depicting various stages of conceptual design.



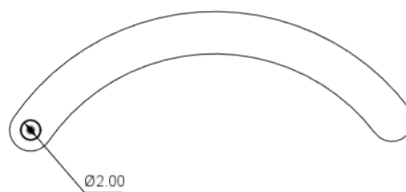
**Figure 4.13:** Chassis Body/Electronic Cabinet



**Figure 4.14: Leg Design**



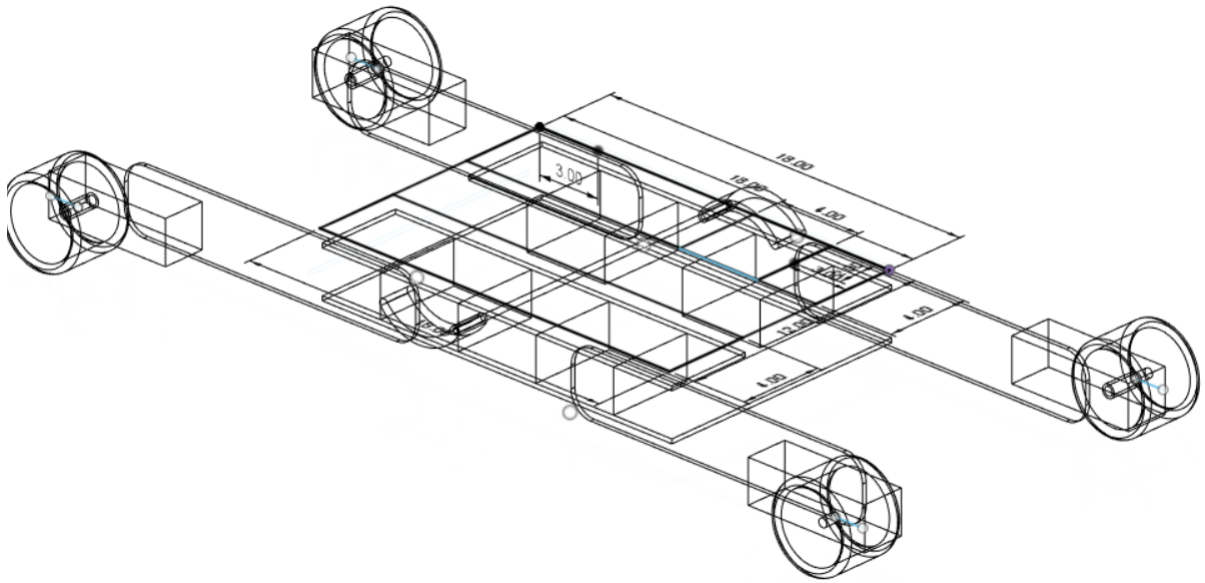
**Figure 4.15: Wheel Dimensions**



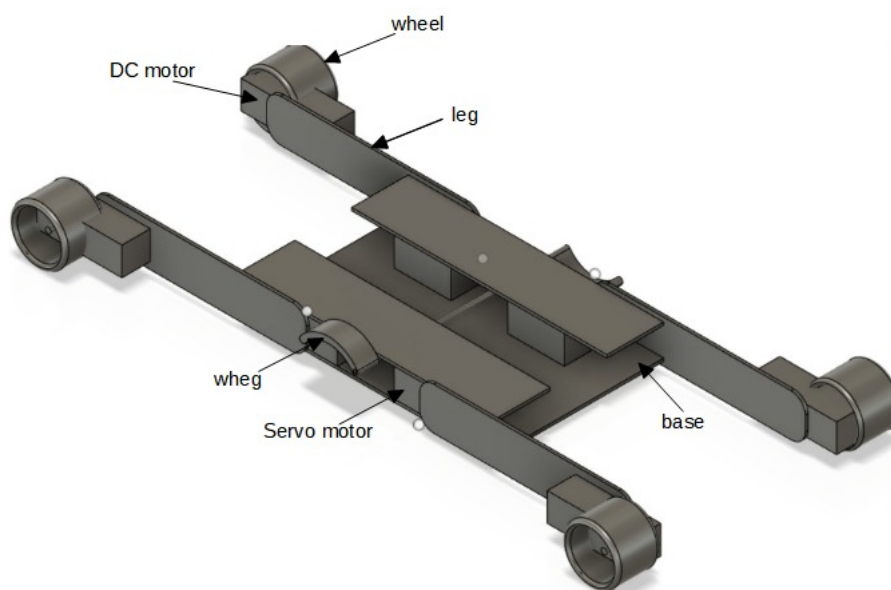
**Figure 4.16: Whег Dimension**



**Figure 4.17:** Whegs Design with Shaft



**Figure 4.18:** Final Assembly Sketch



**Figure 4.19:** Final Assembly Design

# CHAPTER 5

## CODING

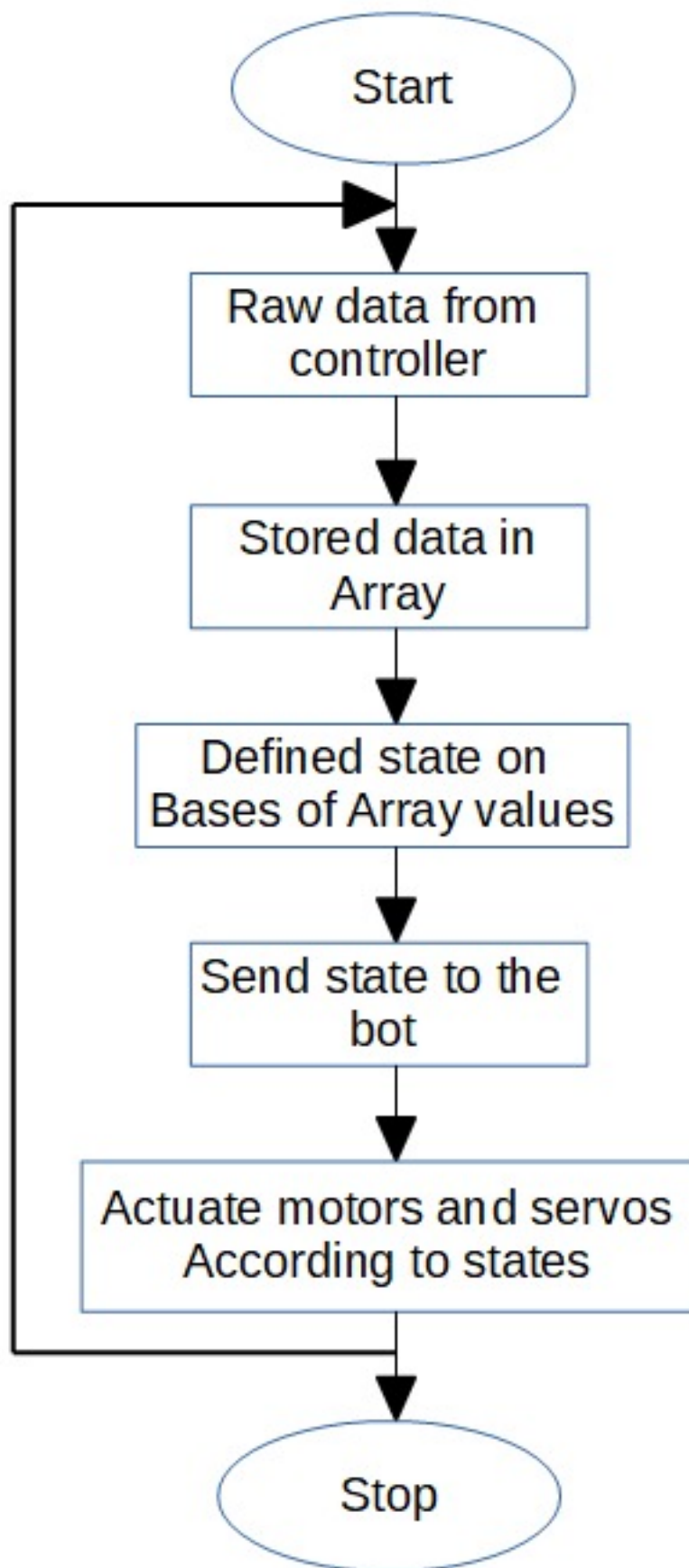
### 5.1 CODE

#### 5.1.1 Receiving user input from controller and sending it the Arduino

The task is to take user input from a gaming console and use that data to control the robot's movement. The controller used here is Logitech f310 gamepad. It has two analog inputs and rest digital push buttons. The data from the controller can be taken by using the library "pygame". Pygame is a Python wrapper module for the SDL multimedia library. It contains python functions and classes that will allow you to use SDL's support for playing cdroms, audio and video output, and keyboard, mouse and joystick input. Thus, by using "pygame" module we are taking the input values from the controller. The input is taken in the form of an array of 17 elements.

Now the controller has two analog joysticks which send a value of -10 to 10 to the computer, other than this the push buttons are used to send either 1 or 0. From 17 elements for the array four are dedicated for analog inputs and the rest for push buttons. The data received is read and processed accordingly for controlling the robot's movements. The task to control the robots is supposed to be done using atmega328p chip. The data must be sent to the IC from the controller. This is done using serial communication. The best way to communicate between python and atmega IC is via a serial medium. The library used for this is "serial" library. This establishes a connection between the python and microcontroller over the comport it is attached to the computer. This microcontroller IC will be used to transmit the data to the robot.

The data from the python serial monitor is retrieved and send to the com port where the IC is connected to the computer. It is sent in the form of an array itself. This helps the user to assign each index for specific purpose. So as soon as the code starts running the data from the controller is directly sent to the microcontroller without any delay.



**Figure 5.1:** Code Flowchart

```

import serial
import time
import pygame

pygame.init()
j = pygame.joystick.Joystick(0)
j.init()
print ('Initialized Joystick : %s' % j.get_name())

def get():
    out = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    it = 0 #iterator
    pygame.event.pump()
    #Read input from the two joysticks
    for i in range(0, j.get_numaxes()):
        out[it] = round(j.get_axis(i),1)
        it+=1
    #Read input from buttons
    for i in range(0, j.get_numbuttons()):
        out[it] = j.get_button(i)
        it+=1
    if(abs(out[1])>abs(out[0])):
        out[0]=0.0
    else:
        out[1]=0.0
    if(out[1]==0.1 or out[1]==-0.1):
        out[1]=0.0
    if(out[0]==0.1 or out[0]==-0.1):
        out[0]=0.0
    if(out[3]==0.1 or out[3]==-0.1):
        out[3]=0.0
    if(out[4]==0.1 or out[4]==-0.1):
        out[4]=0.0
    out[1]=out[1]*10
    out[0]=out[0]*10
    out[3]=out[3]*10
    out[4]=out[4]*10
    #s=str(out).strip('[]')
    #data = pickle.dumps(s)
    #return data
    return out

arduino=serial.Serial('COM4', 9600)
time.sleep(2)

print("Enter 1 to turn ON LED and 0 to turn OFF LED")

while 1:

    '''datafromUser=input()
    datafromUser=datafromUser+'\r'
    arduino.write(datafromUser.encode())'''
    x = get()
    for i in range(0,17):
        x[i]=107+int(x[i])
        x[i]=chr(x[i])

    ch=''
    for i in range(0,17):
        ch=ch+(x[i])
    ch=ch+'\r'
    print(ch)
    arduino.write(ch.encode())

```

**Figure 5.2:** code for receiving user input

### 5.1.2 Serially receiving and decoding the input from the controller and sending it to bot

First, we need to include the SPI standard library for the serial peripheral interface with the **nrf-24l01** transceiver module. Then we have used an external library **RF24.h** for handling SPI communication and other functionalities of the transceiver module. Radio object of RF24 class is created with chip select and chip enable connected to the 7<sup>th</sup> and 8<sup>th</sup> pins of Arduino, respectively. The five bit address for transmission is set to "00001". The serial data is received in an array of 17 elements which indicates the state of different buttons of the controller

1. In the **setup** section first the radio is initialized for all the rf communication. Then the pipe is set up using **openWritingPipe()** function to write the data to a particular address. All the reading pipes are closed using the **stopListening()** function. Serial communication is initialized at 9600 baud rate.
2. Next, in the loop section, the received data is decoded and proper combinations are made for selecting the required action to be taken by the bot, which is referred to as particular states of the bot. There are in total 17 states of the bot which refers to particular actions it can perform when a combination of buttons of the controller is pressed.
3. After deciding the state, the data is stored in the form of a number and sent via nrf-24l01 module using the write() function.

### 5.1.3 Receiving the state and actuating it using servo and dc motors

In this also we include the SPI standard library for the serial peripheral interface with the nrf-24l01 transceiver module of the Blue-tooth module as well as the servo library. Then we have used an external library RF24.h for handling SPI communication and other functionalities of the transceiver module. Radio object of RF24 class is created with chip select and chip enable connected to the 7<sup>th</sup> and 8<sup>th</sup> pins of Arduino, respectively. The five bit address for reception is set to "00001".

**In addition Servo library is included and four servo instances are created and required global variables are declared**

1. In the setup section first the radio is initialized for all the rf communication. Then the reading pipe is set up using **openReadingPipe()** function to receive data from a particular



File Edit Sketch Tools Help



```

Controller_to_arduinoFinal $
1 #include <SPI.h>
2 #include <RF24.h>
3
4 RF24 radio(7, 8); // CE, CSN
5 const byte address[6] = "00001";
6
7 //int datafromUser = 0;
8 int data;
9 int i = 0;
10 int d[17] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
11 /*initialise the pins to be used here
12
13 */
14 int state=0;
15 void setup() {
16   radio.begin();
17   radio.openWritingPipe(address);
18   radio.setPALevel(RF24_PA_MIN);
19   radio.stopListening();
20   Serial.begin(9600);
21 }
22
23
24 void loop() {
25   while(Serial.available() > 0) {
26     if(i<15){
27       data = Serial.read();
28       d[i]=data;
29       i++;
30     }
31     else{
32       i=0;
33       break.
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60   else if(d[11]==1&&d[9]==1){
61     state=8;
62   }
63   else if(d[11]==1&&d[10]==1){
64     state=9;
65   }
66   else if(d[11]==1){
67     state=10;
68   }
69   else if(d[12]==1){
70     state=11;
71   }
72   else if(d[1]>0){
73     state=12;//wheels forward
74   }
75   else if(d[1]>0&&d[13]==1){
76     state=13;//whegs forward
77   }
78   else if(d[2]>0){
79     state=14;//wheels back
80   }
81   else if(d[2]>0&&d[13]==1){
82     state=15;//whegs back
83   }
84   else if(d[3]>0){
85     state=16;//wheels left
86   }
87   else if(d[4]>0){
88     state=18;//wheels right
89   }
90   radio.write(state, sizeof(state));
91 }
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
255
```

address. All the writing pipes are closed using the **startListening()** function. Serial communication is initialized at 9600 baud rate. All the four servo instances are attached to the required pins of Arduino. We have used two motor controllers for controlling four dc motors for leg wheels and two dc motors for the whegs. Then all the servos are brought to the initial position

2. We have defined a custom function `moveWheelsDC()` to control the forward ,backward ,left and right motion of wheels as well as the whegs.
3. Next, in the loop section, dc motors and servo motors are actuated based on the received state which are described as follows:

State 0: Reset all the servo motors by bringing them to the initial position

State 1: Put the bot only on whegs by raising all the legs using servo motor

State 2 to 5: Increment individual servos

State 6 to 9: Decrement individual servos

State 10: Increment all the servos together

State 11:Decrement all the servos together

State 12,14,16,18: Move the bot forward, back, left, right respectively

State 15 and 17:Move the whegs front and back

ardMotorControlRohan

```

1 #include <SPI.h>
2 #include <RF24L01.h>
3 #include <RF24.h>
4 #include <Servo.h>
5 RF24 radio(7, 8); // CE, CSN
6
7 const byte address[6] = "00001";
8 int state=0;
9 int i=0;
10 int pos=0;
11 Servo servolf;
12 Servo servorf;
13 Servo servolb;
14 Servo servorb;
15 int servolfp,servorfp,servolbp,servorbp;
16
17 /* declare all the pins to be used here
18 *
19 */
20 /* STATES
21 0-Arms down .All servo at 45 degree
22 1-All arms up.All servo at 135 degrees
23 2-Increment servo S1
24 3-Increment servo S2
25 4-Increment servo S3
26 5-Increment servo S4
27 6-Decrement servo s1
28 7-Decrement servo s2
29 8-Decrement servo s3
30 9-Decrement servo s4
31 10-Increment all servos
32 11-Decrement all servos
33 12-Wheels DC motor forward

```

ardMotorControlRohan

```

40 19-Wheels DC motor right
41 */
42 void setup() {
43   Serial.begin(9600);
44   radio.begin();
45   radio.openReadingPipe(0, address);
46   radio.setPALevel(RF24_PA_MIN);
47   radio.startListening();
48   /* declare the functioning of pins here
49   *
50   */
51   servolf.attach(2);
52   servorf.attach(3);
53   servolb.attach(4);
54   servorb.attach(5);
55   pinMode(10,OUTPUT);
56   pinMode(11,OUTPUT);
57   pinMode(12,OUTPUT);
58   pinMode(13,OUTPUT);
59   pinMode(14,OUTPUT);
60   pinMode(15,OUTPUT);
61   delay(100);
62   int pos=0;
63   servolf.write(pos);
64   servorf.write(pos);
65   servolb.write(pos);
66   servorb.write(pos);
67   servolfp=pos;servorfp=pos;servolbp=pos;servorbp=pos;
68 }
69
70
71
72 void moveWheelsDC(String motion){

```

ardMotorControlRohan

```

70
71
72 void moveWheelsDC(String motion){
73   if(motion == "Forward")
74   {
75     digitalWrite(10,HIGH);
76     digitalWrite(11,LOW);
77     digitalWrite(12,HIGH);
78     digitalWrite(13,LOW);
79     Serial.println("Forward");
80   }
81   if(motion == "Back")
82   {
83     digitalWrite(10,LOW);
84     digitalWrite(11,HIGH);
85     digitalWrite(12,LOW);
86     digitalWrite(13,HIGH);
87     Serial.println("Backward");
88   }
89   if(motion == "Right")
90   {
91     digitalWrite(10,LOW);
92     digitalWrite(11,HIGH);
93     digitalWrite(12,HIGH);
94     digitalWrite(13,LOW);
95     Serial.println("Right");
96   }
97   if(motion == "Left")
98   {
99     digitalWrite(10,HIGH);
100    digitalWrite(11,LOW);
101    digitalWrite(12,LOW);
102    digitalWrite(13,HIGH);
103    Serial.println("Left");
104   }
105 }

```

ardMotorControlRohan

```

133
134 void loop() {
135   if (radio.available()) {
136     radio.read(&state, sizeof(state));
137     //On the basis of various states control different motors
138     if(state==0)
139     {
140       for(int pos=0;pos<45;pos=pos+2)
141       {
142         servolf.write(pos);
143         servorf.write(pos);
144         servolb.write(pos);
145         servorb.write(pos);}
146       servolfp=pos;servorfp=pos;servolbp=pos;servorbp=pos;
147     }
148     else if(state==1){
149       for(int pos=0;pos<135;pos=pos+2)
150       {
151         servolf.write(pos);
152         servorf.write(pos);
153         servolb.write(pos);
154         servorb.write(pos);}
155       servolfp=pos;servorfp=pos;servolbp=pos;servorbp=pos;
156     }
157     else if(state==2){
158       servolfp=servolfp+5;
159       servolf.write(servolfp);
160     }
161     else if(state==3){
162       servorf=servorf+5;
163       servorf.write(servorf);
164     }
165   }

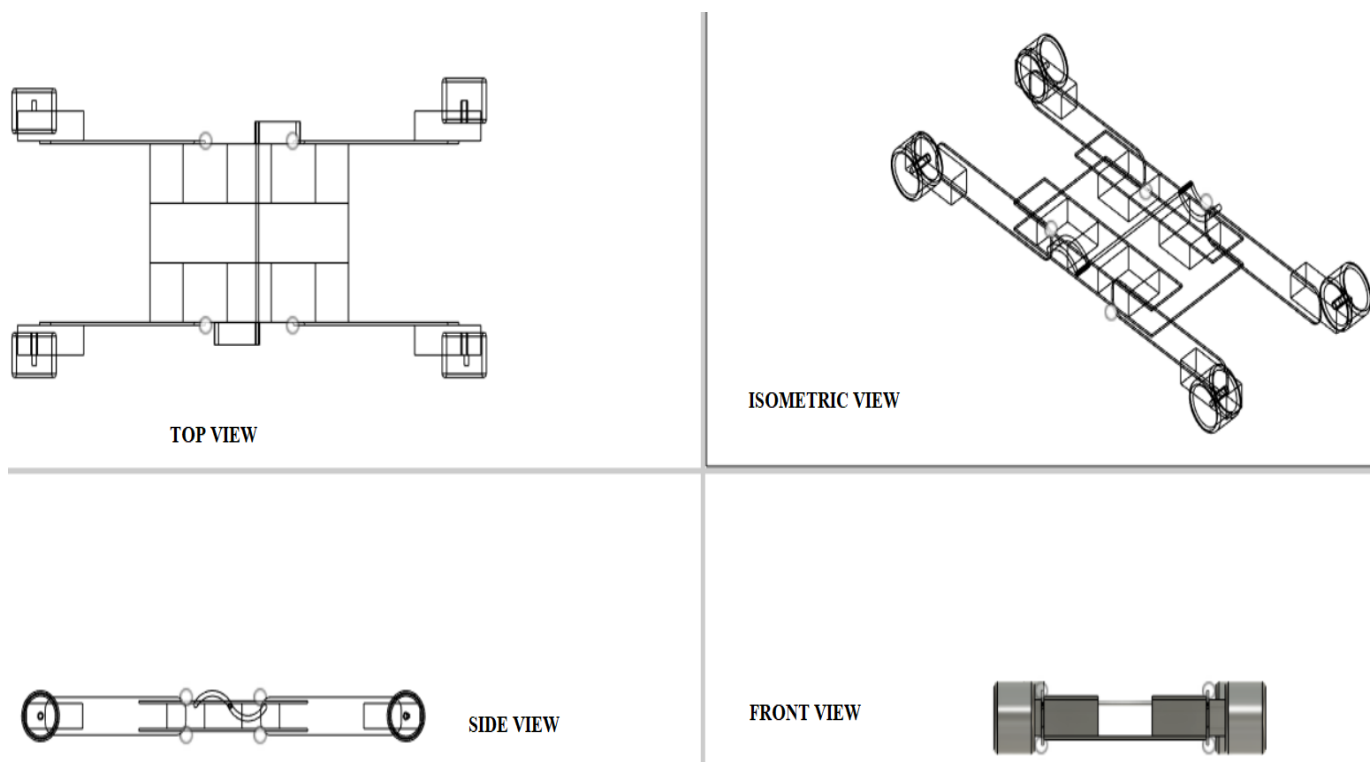
```

Figure 5.4: Code for motor actuation

# CHAPTER 6

## CONCLUSION

1. Our model of multi terrain bot employs both movable arms as well as whegs mechanism which makes it suitable for almost any kind of rough terrain , unlike the previously existing models which generally employs either of the two.
2. Our model would be suitable for terrain with small obstacle like stones, stair like incline, terrain with depressions or mild holes and normal slopes.
3. Wheels attached to the arms can move the bot in forward ,backward, left and right directions ,while the whegs can clear the obstacle in forward and backward direction.
4. While encountering a rough surface the robot adjusts its structure by either raising or lowering the four arms to efficiently use both whegs and wheels.
5. At present these adjustments in the structure are made manually through a controller using RF communication which can be modified after proper testing and be made semi autonomous or even autonomous using certain sensors



**Figure 6.1:** Different Views Of Design

## CHAPTER 7

### FUTURE ENHANCEMENT

1. This prototype can be implemented for vast area of exploration sciences and military purposes by replacing the current transceiver by a more efficient **Radio Frequency Technology** module with larger range of communication and also replacing servo motors by more efficient **stepper motors**
2. It can be made amphibious by protecting the housing cabinet and other modifications.
3. It can be made autonomous by feeding into GPS way-point pattern and using Artificial Intelligence.
4. In medical fields the bot can be modified into wheelchair designs.
5. In automobile industry , military vehicles.

# APPENDIX A

## MATERIALS

### A.1 Hardware

1. **Arduino Microcontroller:**Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started
2. **Polyacrylonitrile Acrylic Polymer/Aluminium Metal Sheets**
3. **nRF24L01 transceiver module:**nRF24L01 transceiver module it uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space and with lower baud rate its range can reach up to 100 meters.The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time.The power consumption of this module is just around 12mA during transmission, which is even lower than a single LED. The operating voltage of the module is from 1.9 to 3.6V, but the good thing is that the other pins tolerate 5V logic, so we can easily connect it to an Arduino without using any logic level converters.Three of these pins are for the SPI communication and they need to be connected to the SPI pins of the Arduino, but note that each Arduino board have different SPI pins. The pins CSN and CE can be connected to any digital pin of the Arduino board and they are used for setting the module in standby or active mode, as well as for switching between transmit or command mode. The last pin is an interrupt pin which doesn't have to be used.
4. **L293D Motor Driver IC:**L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. As you know voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, hence H-bridge IC are ideal for driving a DC motor. In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. There are two Enable pins on l293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge you need to enable pin 1 to high. And for right H-Bridge you need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend

working. It's like a switch. VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply). L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply. The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d.

5. DC motor: The metal gears have better wear and tear properties. Gearbox is sealed and lubricated with lithium grease and requires no maintenance. Although motor gives 300 RPM at 12V, motor runs smoothly from 4V to 12V and gives the wide range of RPM, and torque. The shaft has a hole for better coupling.
6. Servomotors: The term servomotor does not refer to one single kind of motor. Instead it refers any type of motor that receives a command signal from a controller. In this same respect, any closed loop system can be referred to as a servo system. This flexibility allows for several suitable types of electric motors to be used in servo systems. These electric motors include:

Permanent Magnet DC Motor

Brushless DC Motor

Induction AC Motor

Electromagnetic motors operate based on the principle that the magnetic force on an electrical conductor in a magnetic field is perpendicular to that field. This force is defined by,

$$F = qv \cdot B$$

Where:

F is the vector describing the magnetic force

q is the magnitude of the electrical charge

v is the vector magnitude of the charged particles velocity

B is the vector describing the magnetic field

However, in the case of an electric motor the force can be quantified as a scalar

$$F = I \cdot L \cdot B$$

Where: F is the magnetic force

I is the electric current in the coil

L is the length of the coil contained within the magnetic field

B is the strength of the magnetic field

## A.2 Software

1. Arduino IDE: Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development

Environment) that runs on your computer, used to write and upload computer code to the physical board.

2. IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has the following features:

- coded in 100% pure Python, using the tkinter GUI toolkit

- cross-platform: works mostly the same on Windows, Unix, and macOS

- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages

- multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features

- search within any window, replace within editor windows, and search through multiple files (grep)

- debugger with persistent breakpoints, stepping, and viewing of global and local namespaces configuration, browsers, and other dialogs

3. Fusion360: Fusion360 is a solid modeling computer-aided design (CAD) and computer-aided engineering (CAE) computer program that runs on Microsoft Windows. Fusion360 is published by Autodesk.



## REFERENCES

- [1] Harkins, R., Ward, J., Vaidyanathan, R., Boxerbaum, A. X., and Quinn, R. D. (2005). “Design of an autonomous amphibious robot for surf zone operations: part ii - hardware, control implementation and simulation.” *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 1465–1470.
- [2] Koh, D. C., Dharmawan, A. G., Hariri, H. H., Soh, G. S., Foong, S., Bouffanais, R., Low, H. Y., and Wood, K. L. (2019). “Design and analysis of a miniature two-wheg climbing robot with robust internal and external transitioning capabilities.” *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 9740–9746.
- [3] Kousar, H., Atif, M., Ali, H., and Kundi, M. H. (2019). “Whegstr: A multiterrain robot with c-shaped whegs, implementation of error minimization technique and using artificial neural network (ann).” *2019 International Conference on Green and Human Information Technology (ICGHIT)*, IEEE, 139–141.
- [4] Nakajima, S. (2009). “Development of four-wheel-type mobile robot for rough terrain and verification of its fundamental capability of moving on rough terrain.” *2008 IEEE International Conference on Robotics and Biomimetics*, IEEE, 1968–1973.
- [5] Naskar, S., Das, S., Seth, A. K., and Nath, A. (2011). “Application of radio frequency controlled intelligent military robot in defense.” *2011 International Conference on Communication Systems and Network Technologies*, 396–401.
- [6] Omijeh, B., Uhunmwangho, R., and Ehikhamenle, M. (2014). “Design analysis of a remote controlled pick and place robotic vehicle.” *International Journal of Engineering Research and Development*, 10(5), 57–68.
- [7] Schroer, R. T., Boggess, M. J., Bachmann, R. J., Quinn, R. D., and Ritzmann, R. E. (2004). “Comparing cockroach and whegs robot body motions.” *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, Vol. 4, 3288–3293 Vol.4.
- [8] Ward, J. L. (2005). “Design of a prototype autonomous amphibious whegs robot for surf-zone operations.” Ph.D. thesis, Monterey, California. Naval Postgraduate School, Monterey, California. Naval Postgraduate School.
- [9] Watson, J. T., Ritzmann, R. E., Zill, S. N., and Pollack, A. J. (2002). “Control of obstacle climbing in the cockroach, *blaberus discoidalis*. i. kinematics.” *Journal of Comparative Physiology A*, 188(1), 39–53.
- [10] Weihmann, T., Brun, P.-G., and Pycroft, E. (2017). “Speed dependent phase shifts and gait changes in cockroaches running on substrates of different slipperiness.” *Frontiers in zoology*, 14(1), 54.
- [11] Wile, G. D., Daltorio, K. A., Palmer, L. R., Witushynsky, T. C., Southard, L., Ahmad, M. R., Malek, A. A., Gorb, S. N., Boxerbaum, A. S., Ritzmann, R. E., and Quinn, R. D.

(2008). “Making orthogonal transitions with climbing mini-whegs.” *2008 IEEE International Conference on Robotics and Automation*, 1775–1776.