# CHAPTER I

# INTRODUCTION

Garbage is the major problem not only in cities but also in rural areas of India. It is a major source of pollution. Indian cities alone generate more than 100 million tons of solid waste a year. In 2000, India's Supreme Court directed all Indian cities to implement a comprehensive waste-management program that would include household collection of segregated waste, recycling and composting. These directions have simply been ignored. No major city runs a comprehensive program of the kind envisioned by the Supreme Court. It is not wrong to say that India is on verge of garbage crisis even though 9000crore rupees are allotted for the Swachh Bharath Abhiyan. While India has been grappling with waste management issues, there are not many companies in this space that are utilizing artificial intelligence or robotics. With a continuously towering waste heaps; the dumps at many places are a reason of major outcry. Though there are many companies who are working towards dealing the waste, there aren’t many instances of robotics or AI being at place.

There are already different types of garbage collection robots like Robot-dumpster which mainly aims at collecting garbage from full cans and dispose it designated area and Dust cart which is designed to navigate through urban areas avoiding static and dynamic obstacle and waste door to door. These robots which are in use have various disadvantages like high implementation cost, not user friendly and aims at only collecting filled dustbins but not on collecting mechanism, etc.

Also, Municipal solid waste workers (MSWWs) or refuse collectors, universally expose too many work related health hazards and safety risks, notably allergic and other diseases of the respiratory system.

Health impacts could also entail musculoskeletal, gastro intestinal and infectious diseases as well as injuries caused by work-related accidents.

Hence to overcome this major problem of waste collection Autonomous Garbage Collection robot is developed. It aims at providing automatic control to collect the garbage in a confined space. The robot is designed to work in a region with specific design, for example a laboratory. It differentiates between static and dynamic obstacle and move accordingly as it programmed. It basically consists of sensors at different levels to detect the dynamic obstacle. It also disposes the garbage to a pre-specified place. If the trash bin is filled, it will be detected and the garbage will be disposed.

## 1.1 Aim of the project

Garbage is the major problem not only in cities but also in rural areas of India. It is a major source of pollution. Indian cities alone generate more than 100 million tons of solid waste a year. In 2000, India's Supreme Court directed all Indian cities to implement a comprehensive waste-management program that would include household collection of segregated waste, recycling and composting. These directions have simply been ignored. No major city runs a comprehensive program of the kind envisioned by the Supreme Court. It is not wrong to say that India is on verge of garbage crisis even though 9000crore rupees are allotted for the Swachh Bharath Abhiyan. While India has been grappling with waste management issues, there are not many companies in this space that are utilizing artificial intelligence or robotics. With a continuously towering waste heaps; the dumps at many places are a reason of major outcry. Though there are many companies who are working towards dealing the waste, there aren’t many instances of robotics or AI being at place.

There are already different types of garbage collection robots like Robot-dumpster which mainly aims at collecting garbage from full cans and dispose it designated area and Dust cart which is designed to navigate through urban areas avoiding static and dynamic obstacle and waste door to door. These robots which are in use have various disadvantages like high implementation cost, not user friendly and aims at only collecting filled dustbins but not on collecting mechanism, etc.

Also, Municipal solid waste workers (MSWWs) or refuse collectors, universally expose too many work related health hazards and safety risks, notably allergic and other diseases of the respiratory system.

Health impacts could also entail musculoskeletal, gastro intestinal and infectious diseases as well as injuries caused by work-related accidents.

Hence to overcome this major problem of waste collection Autonomous Garbage Collection robot is developed. It aims at providing automatic control to collect the garbage in a confined space. The robot is designed to work in a region with specific design, for example a laboratory. It differentiates between static and dynamic obstacle and move accordingly as it programmed. It basically consists of sensors at different levels to detect the dynamic obstacle. It also disposes the garbage to a pre-specified place. If the trash bin is filled, it will be detected and the garbage will be disposed.The autonomous trash detection and collection robot aims to be an independent entity that detects waste matter( such as paper, plastic bottles, etc) on its own accord, i.e. detection of waste using image processing and machine learning; and also utilizing the mechanism built on the body of the robot to collect the same. The objective is to make the robot autonomous, i.e.an intelligent robot. So, the question arises "Can machines think?" Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to the prediction of an event. We mainly incorporate object detection and path finding algorithms to the robot.

1) Object detection [1]

Every object class has its own special features that help in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin color and distance between eyes can be found.

Typically, there are three steps in an object detection framework.

1. First, a model or algorithm is used to generate regions of interest or region proposals. These region proposals are a large set of bounding boxes spanning the full image (that is, an object localization component).
2. In the second step, visual features are extracted for each of the bounding boxes, they are evaluated and it is determined whether and which objects are present in the proposals based on visual features (i.e. an object classification component).

3. In the final post-processing step, overlapping boxes are combined into a single bounding box (that is, non-maximum suppression).

Here, the objective is to gather information (i.e. count the number of instances of an object) for specific matter, mainly paper. So the object under consideration is paper (crumpled or otherwise) and the image read by the bot will oriented towards paper only. The bot is put under a testing phase to enable it to detect paper from its surroundings.

2) Path finding [2]

The path planning method describe in this paper aims to calculate Shortest path for a robots, while avoiding obstacles along the way. The autonomous robots are generally based on navigating from a starting point to target point in unknown environment. In these kinds of applications the points that should be visited are unknown. But, for an autonomous robot, consuming less energy and time is very important. To meet these constraints, a shorter path is preferred rather than a longer path. Therefore an intelligent path planning algorithm is always required. Path planning for robots is a complex task. In order to find the shortest path between starting point to goal point in the environment, it is usually needed a map to find shortest the path and follow it. It would be more useful solution for which maps are not available from source to destinations particularly in unknown environment like mars surface . Even more, it would be better, if system able to find the better solution by itself and guide robot to the goal. The path planning method described in this paper aims to calculate the path more effectively. The proposed method is based on stereo images processing which can be captured by the camera set equipped on Base station system (Computer). Obstacles detecting methods are then performed to identify the existence of obstacles within the unknown environment. Shortest path is obtained by A-Star algorithm. Based on this shortest path, Base station system then guides robot to its destination.

3) Collection of the waste:

This section can be further divided into two parts

i. Dust collection [3][4][5]:

The aim is to design a vacuum motor assembly such that it works efficiently and consumes less power. The area of suction is calculated based on the motor that is being used for the operation and accordingly the frame has been designed to accommodate the mounting points for supporting components for the same operation such as side brushes, collecting box. The prime aim of this design is to collect only the dust and dirt.

ii. Solid waste collection:

In the current market there are many machines that focus on the collection of dust and not the collection of solid waste. A mechanism is to be designed to collect the solid waste. Since this is in initial stages, the prototype is considered to work in engineering labs. Labs are known to have lot of paper waste. Keeping this in mind the collection mechanism is to be designed which is discussed in the further section.

## 1.2 Scope of the project

### 1.2.1 Market Analysis

There are many different companies that develop robotic vacuum cleaners. In the appendix under ‘A. Market Analysis’ there is more detailed information about the main competitors. In the following the conclusion of the market analysis are presented. Difference between ‘traditional’ vacuum cleaners and vacuum cleaning robots A vacuum cleaning robot works a lot like a traditional, manual vacuum cleaner. The main difference is that a robotic vacuum cleaner is equipped with brushes, which move the dust to the nozzle. Some robotic vacuum cleaners have extra brushes which collect the dust on both sides of the robot and brush this dust right into the nozzle. This feature allows the robots to sweep along walls and clean thus more effective. The effectiveness of a robotic vacuum cleaner is also determined by the quality of the suction mechanism and the brushes. In comparison with manual vacuum cleaners, the cleaning process of robotic vacuum cleaners takes a longer time. It is slower and through its limited battery life it sometimes has to recharge within its cleaning round. Therefore completing the vacuuming of an entire room takes longer. This is something the consumer is well aware of and since the robot cleans mostly when the consumer is not at home, this should not be a problem.

Pros of robotic vacuum cleaner:

1. The consumer saves time
2. The consumer has to install the robot only once (after the purchase)
3. The robotic vacuum cleaner will automatically vacuum on set times
4. Most models do not require a dust bag

These are very important to keep in mind while the designing process.

Cons of robotic vacuum cleaners

1. The suction power is weaker than in a traditional vacuum cleaner since it works on a battery
2. It has a smaller dust storage

3. The round-shape of the most robotic vacuum cleaners does not allow them to vacuum the corners properly
4. The vacuum cleaning robot is not efficient around objects; the consumer has to tidy up and
5. Removes all small objects from the ground, like toys, shoes etc.

### 1.2.2 Difference between brands

There are many differences between brands. These differences concern mostly the technology. Some robotic vacuum cleaners have many sensors, which makes the robot smarter. The device can determine where he already vacuumed through these sensors. The navigation method differs from zigzag method to spot method. Most of the robotic vacuum cleaners have a docking station which the robot can return to by itself. Notable is, that the Easy Star from Philips is the dumbest2 robot. It does not have a navigation route or docking station in contradiction to Philip’s other robotic vacuum cleaner, the homerun. This one has Navigation, but failed at so many other aspects, that it has been removed from the stores. The Easy star also has a stairs detector to prevent it from falling down the stairs. There are a lot of problems that occurred since the robotic vacuum cleaner is a relative new technologic device. Like mentioned above a robotic vacuum cleaner has to be charged quite often. The average time the robot cleans before it has to be recharged is short. Much brands have the same solution for this problem; a docking station. The robot can return to the docking station by itself. Another problem is the small capacity of the dustbin. The robot is small which means it is easy to maneuver and can drive underneath furniture; concluding, there is little space for a dustbin. The Ecovacs vacuum robot has solved this problem by using the docking station as dust storage. This is a smart solution, because the robot has to return to the docking station when it is out of battery. This method has the advantage that it takes less time to empty the dustbin of the robotic vacuum cleaner. A round shaped vacuum robot cannot vacuum the corners. A solution for this is a square shaped robot. But a square has its problems, too; it cannot find its way back from every place it has driven into. It also gets stuck easily when rotating around its own axis. The LG Hom-Bot Square has solved this problem by rounding the corners. The brushes are positioned on the sides through which still reach the corners. Another advantage of LG robots is that the user can choose the method of vacuum cleaning (zigzag, cel by cell or spot). So each user is able to choose the method they prefer. In this way, many users will be pleased by the product because they can choose their favorite method.

Strengths per vacuum cleaning robot:

1) IRobot Roomba: One does not have to clean the brushes manually any longer.

2) Ecovacs Robot: One of the main features of Ecovacs is the robot automatically empties its dust in a docking station.

3) Samsung SR8895: The robot knows where it has been, because of efficient room mapping.

4) LG Hom-Bot: This device has certain modes to clean the room. One of the modes is cell-by-cell cleaning, this is handy by large areas with many obstacles.

5) LG Hom-Bot Square: Is able to clean the corners with its brushes.

6) Philips Easystar: Is really small, so it can easily drive underneath the couch.

7) Traditional manual cleaner: He can reach under furniture and makes it very easy to vacuum stairs and vertical surfaces. Also the user can change different heads for different tasks.

8) Central vacuum system: These do not require a person to carry a heavy unit from room to room or up and down the stairs. There is furthermore no electrical cord which can tangle up.

### 1.2.3 Problem Identification

Throughout the design process we have tried and identified a total of four persistent problems.

These include the below mentioned:

a. Battery life: Since they do not have a plug and the motor is not as strong as the one of a manual vacuum cleaner, they often have to be charged. This again can lead to extra work and makes the product less desirable.
b. Robotic vacuum cleaner has not the possibility to clean in corners like a manual vacuum cleaner. Companies tried longer brushes and special features but until now they have not found a shape or a feature, which can efficiently clean in the corners.
c. The solid waste such as the paper balls are still left in the room which is not cleaned by the bots.
d. Since the bot is made compact the dust collection box is very small.

From the above Market Analysis and problems we identified we have decided the below mentioned aspects.

The robot aims to clean the area under test, mainly cleaning of paper from the floor of an indoor laboratory. The unique aspect about the project is to inculcate the trash collection of two types of waste, i.e. dust and solid waste.

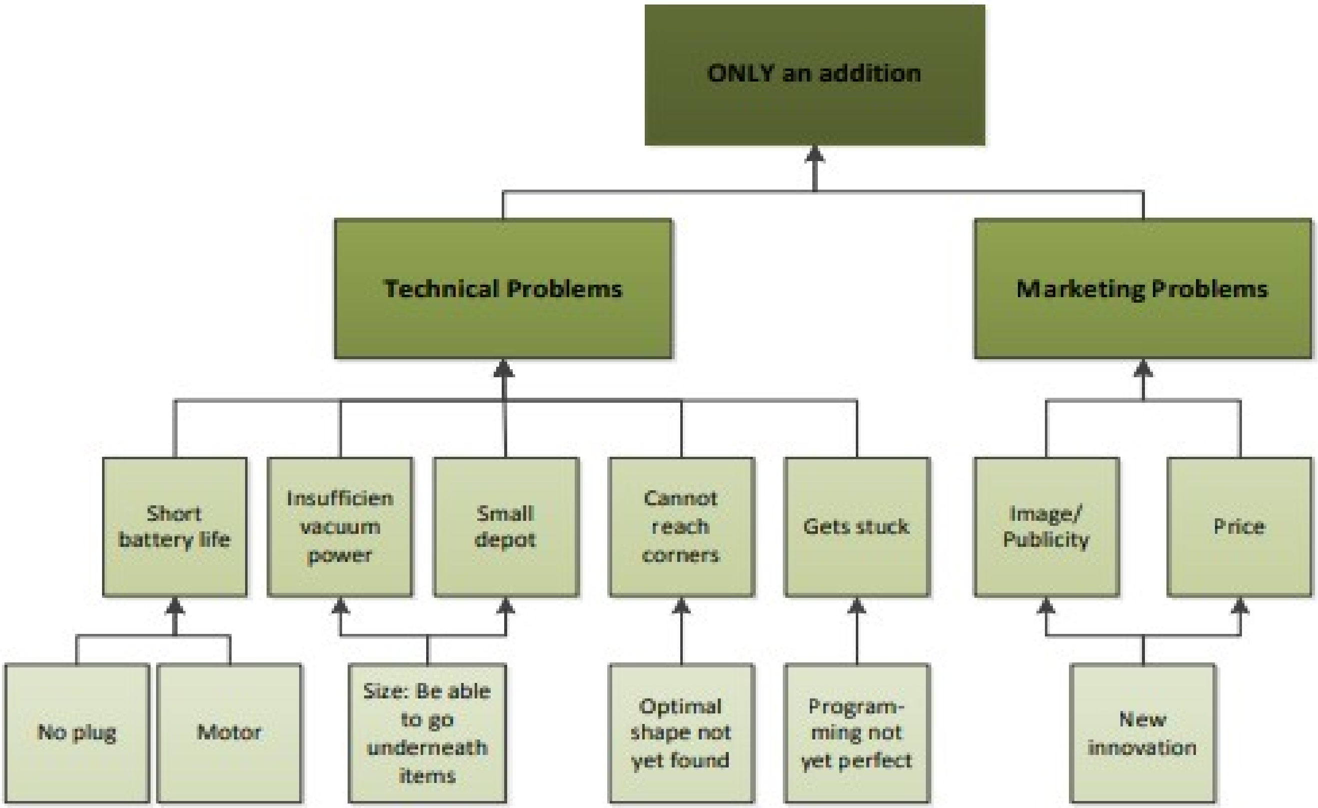



**Fig 1.1:** Problem Identification Flow Chart

Dust particles vary in size from visible to invisible. The smaller the particle, the longer it stays in the air and the further it can travel. So using this principle, the robot is designed to push the dust off the ground. The dust is removed using a vacuum duct and brushes, facing the floor. The brushes are run using 5V DC motors, which are placed on the lower corners of the head of the robot. When these brushes are driven, they agitate the dust particles on the floor and this dust is accumulated using a vacuum and suction duck placed on the body of the robot. The vacuum is also driven with a 5 v DC motor.

Keeping the orientation of the robot as the same, the solid waste collection is done by the flap mechanism. A 2mm aluminum sheet which can rotate 120 degrees, hinged to the wall of the bot is used as the flap. When the bot detects the specimen, the flap is opened upward with the help of a servo motor and while it’s closing it takes in the specimen.

## 1.3 Coverage of the project

The robot has two major aspects, namely:

a) Hardware layer
b) Software layer

**Hardware layer** comprises of mechanical and electronic component that are made to run in tandem with each other to pick up trash. The robot has a specified mechanical design that enables it to collect trash using a flap, which opens upward; when and where it is detected.

The body is fabricated using special care and expertise with the help of CAD software. The exact dimensions of each component of the robot and the chassis were introduced in the designing software and real time simulations were observed to depict its hardware functionality.

To instill the idea of robotics, a special circuit is designs which are infused to the bot so that at the bot becomes mobile. various components such as Arduino Uno, raspberry pi module, Servo Motors ,wheel Motors, step down buck converter, Vacuum motor, Linear voltage regulator And a Raspberry Pi camera are place according to the circuitry depicted in the layout of the robot. A major part of the circuitry is the sensors which are integrated to the robot to access and utilize the data acquired by the Surrounding of the robot. the use of IR and ultrasonic sensors, namely cliff sensors and wall sensors.

An electronic cliff sensor is simply an IR sensor that runs an IR beam along the ground, when a large drop off is encountered - flips a switch or sensor that tells the controller that it has reached a drop off.

Wall sensors are commonly used in platform games but here it works as the name suggests, it detects an obstacle in its path. They consist of four lines, two vertical and two horizontal, fixed to the left, right, top, and bottom of the body. When a sensor detects contact with a wall, a message broadcasted will tell the bot to invert its current velocity, making the wall deflect from its path.

For the **Software part** the Arduino IDE is used to interface the components present on the bot, thereby enabling it to be programmed according to the user's requirements. The robot is controlled manually using a PS3 Bluetooth controller which is interfaced with the Arduino Uno unit and the motor drivers installed in accordance to the bot.

## 1.4 Thesis Organization

This thesis consists of seven chapters out of which, **Chapter 1** gives a proper overview of the project, followed by **Chapter 2** which consists of the technical approach of this project as Literature survey, then comes **Chapter 3** with functional block diagram, flow diagram , the Circuit Block diagram and description of each component. And then the **Chapter 4** explains the Design approach for realizing the implementation beginning with the description of the theory of Arduino, Raspberry Pi and Mecanum wheels. **Chapter5** consists of the Arduino code and steps involves in interfacing the remote controller. **Chapter 6** has the Software tools used in this project. **Chapter 7** consists of the results and **Chapter 8** includes the Conclusions, results and Future Scope of the project.

# CHAPTER II

# LITERATURE REVIEW

People are busy in their many hazards. One of the main problem is garbage collection and its disposal. During the recent times ,it has been evident that the garbage which should be inside the dustbin is actually lying outside the dustbin and causing the pollution of the surrounding environment, which further leads to numerous diseases.

This project proposes the method where a robot professional life and they forget to pay attention to their surrounding problems, which can lead to can be used to clean the polluted areas such as garbage around the dustbin. Robot has the advantages of the powerful image processing and ultrasonic sensor to sense the surrounding area and accordingly action can be taken. An Ultrasonic sensor is used to detect the object and the distance between the object and Robot. The Ultrasonic sensor is also used to employ a movement algorithm for the movement of the robot.

The idea of making a autonomous trash collection and detection robot was developed, keeping the above concept in mind and reading numerous relevant reference papers ,from which the paper on ‘ShobhitKhandare Et.al, “Object Detection Based Garbage Collection Robot (E-Swachh)”, International Research Journal of Engineering and Technology (IRJET) played a decisive role in forming the basis of our project. This paper deals with an autonomous robot which collects garbage using a robotic arm with 3 degrees of freedom, and an image processing aspect that deals with the reading of surroundings of the robot and also the detection of trash [1].

The advancement made by our project is not just the collection of dust, but also the collection of solid waste (such as paper). Solid waste collection is done by the flap mechanism. A 2mm aluminum sheet which can rotate 90 degrees, hinged to the wall of the bot is used as the flap. When the bot detects the specimen, the flap is opened upward with the help of a servo motor and while it’s closing it takes in the specimen. This enhances the scope of garbage collection to a large extent, as the picking mechanism is versatile about the type of garbage to be collected.

The objective of the project was achieved to some extent. The hardware layer was built, whereas the digital image processing layer is soon to be worked on. The circuitry already includes the raspberry pi module, which is used for the image processing phase that holds the autonomy of the robot. Presently, the robot is manually controlled using a controller.

As the trash collection includes collection of dust using a vacuum motor, the design to implement an existing vacuum system to the robot was a challenge. The article by Hsiao Shih-Wen and Yeh, Ting An, ‘Application of Collaborative Design Strategy on Redesign of the Cordless Household Vacuum Cleaner’, International Conference on Organizational

Innovation (ICOI 20170, showed a method to set up the market position of the product through the image scale and subsequently by 5W1H and tree diagram method, and clarifies the problems and the direction of the design goals. The following step, according to Morphological Analysis and Finite Structure method (FSM), arranges the positions of the major components of the product, and shapes the forms for developing the possible alternatives. Then, it is the most suitable for the consumers that using PUGH several times to choose the final solution [2].

the right architecture of the vacuum cleaner is necessary for the various specifications that need to be met , while the operation of cleaning is underway. So to design the fan motor in the vacuum cleaner, an article by Park Changhwan, Jun Sangook, ParkKyunghyun, Lee Sangjong and Chang Kyoungsik, ‘Methodology for SystemLevel Analysis of a Fan-Motor Design for a Vacuum Cleaner’, Part C-Journal of Mechanical Engineering Science, Vol. 231(20) demonstrates the predicted performance for power, flow rate, pressure, and efficiency using the present method agreed well with the experimental results obtained for an equivalent system within 2% difference at best efficiency point. Three models of the fan-motor assembly (S1, S2 and S3) were analyzed at the component level and the decrease in efficiency produced by flow resistance was estimated to be 1% (S1 and S3 models) or 4.7% (S2 model) using the developed method [3].

An article named Effects of Vacuum Cleaner Brushing Action, in the website ‘ristenbatt.com’ tells us the importance of the roller and side brushes .Vacuum cleaners specifically designed to clean carpeted floors have motor or turbine driven revolving brush rolls with rows of bristles and sometimes beater bars as well. These brush rolls revolve at a very high speed, agitating the carpet fibers in the process. The effect of this is to loosen dirt and move it toward the top, aiding in its removal from the carpet fibers. So, an inclusion of 3 D printed brushes has been incorporated, so that the agitation effect on the dust particles is enhanced and thereby the dust collection is done with ease [4].

The idea of implement a vacuum motor driven by the robot architecture was achieved by referring to the article named –vacuum cleaner guide; in the website “learn.compactappliance.com” which helped us understand the various types of vacuum cleaners present in the market and how effectively it can be chosen for a specific need thereby helping us in the selection of the components for a specific need.

The unique aspect of the robot is the wheels that are affixed to it. They are called mecanum wheels. The mecanum wheel is one design for a wheel which can move in any direction. It is sometimes called the Ilon wheel after its Swedish inventor, BengtIlon, who came up with the idea in 1973. The angle between rollers axis and central wheel could have any value but in the case of conventional Swedish wheel it is 45”.The angled peripheral rollers translate a portion of the force in the rotational direction of the wheel to a force normal to the wheel direction .Depending on each individual wheel direction and speed, the resulting combination of all these forces produce a total force vector in any desired direction thus allowing the platform to move freely in the direction of the resulting force vector, without changing of the wheels themselves.Using four of mecanum wheels provides Omni-directional movement for a vehicle without needing a conventional steering system slipping is a common problem in the mecanum wheel as it has only one roller with a single point of ground contact at any one time. Due to the dynamics of the mecanum wheel, it can create force vectors in both the x and y-direction while only being driven in the y-direction. Positioning four mecanum wheels, one at each corner of the chassis (two mirrored pairs), allows net forces to be formed in the x,y and rotational direction.By varying the individual wheel speed we can achieve driving direction along any vector in X-Y axis [6][8][11].

With reference to the paper published by Motion Control and the Skidding of Mecanum Wheel Vehicles Shih-Liang Wang Department of Mechanical Engineering, North Carolina, A&T State University, Greensboro, North Carolina, USA which identifies the problem of slipping of the wheels and provides a solution to overcome the same. With the cause of the slipping identified, the chassis should be stiffened to minimize the slippage caused by the chassis bending. In the most prevalent movements (forward and backward travels and turning on the spot), the wheel’s tipping moments will be balanced at the chassis, and there is no adverse effect [6][8][11].

# Chapter III

# Bot Description

In this chapter the circuit diagram along with mechanical block description is Discusses and each of the block in the system is discussed in detail along With appropriate pictures

## 3.1 Functional Block Diagram

Below is the functional block diagram of the Project, which includes all the major parts and signal paths. These paths represent the communication type and modes used.

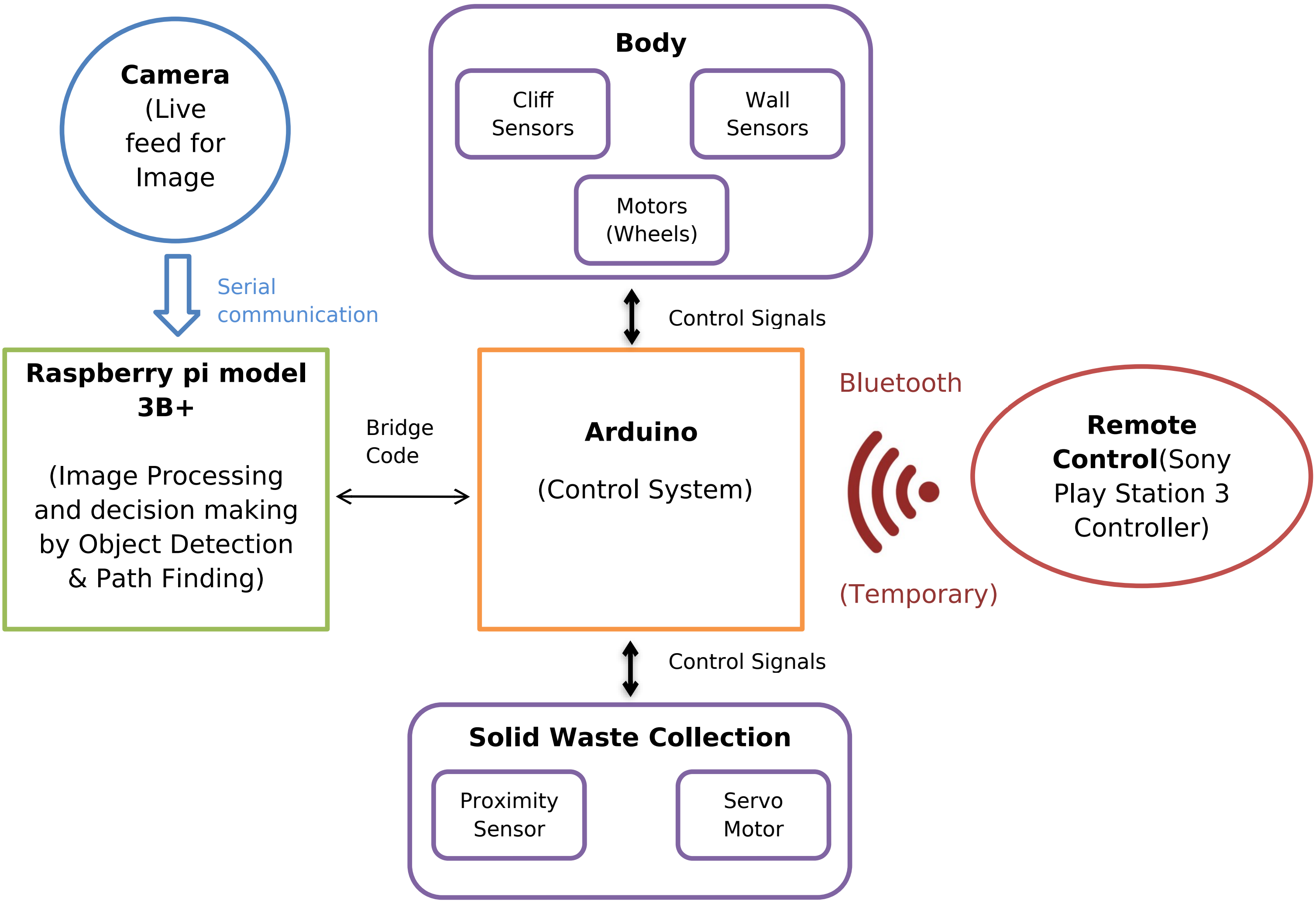



**Fig 3.1**Functional Block Diagram

Each Shape holds a specific purpose and makes the functional block diagram more legible and easy to read and understand.

## 3.2 Signal Flow Chart of the Bot

The Fig 3.2.1 shows the signal flow in the circuit of the bot. It consists of outputs, inputs, microcontroller and the microprocessor along with the end effectors.

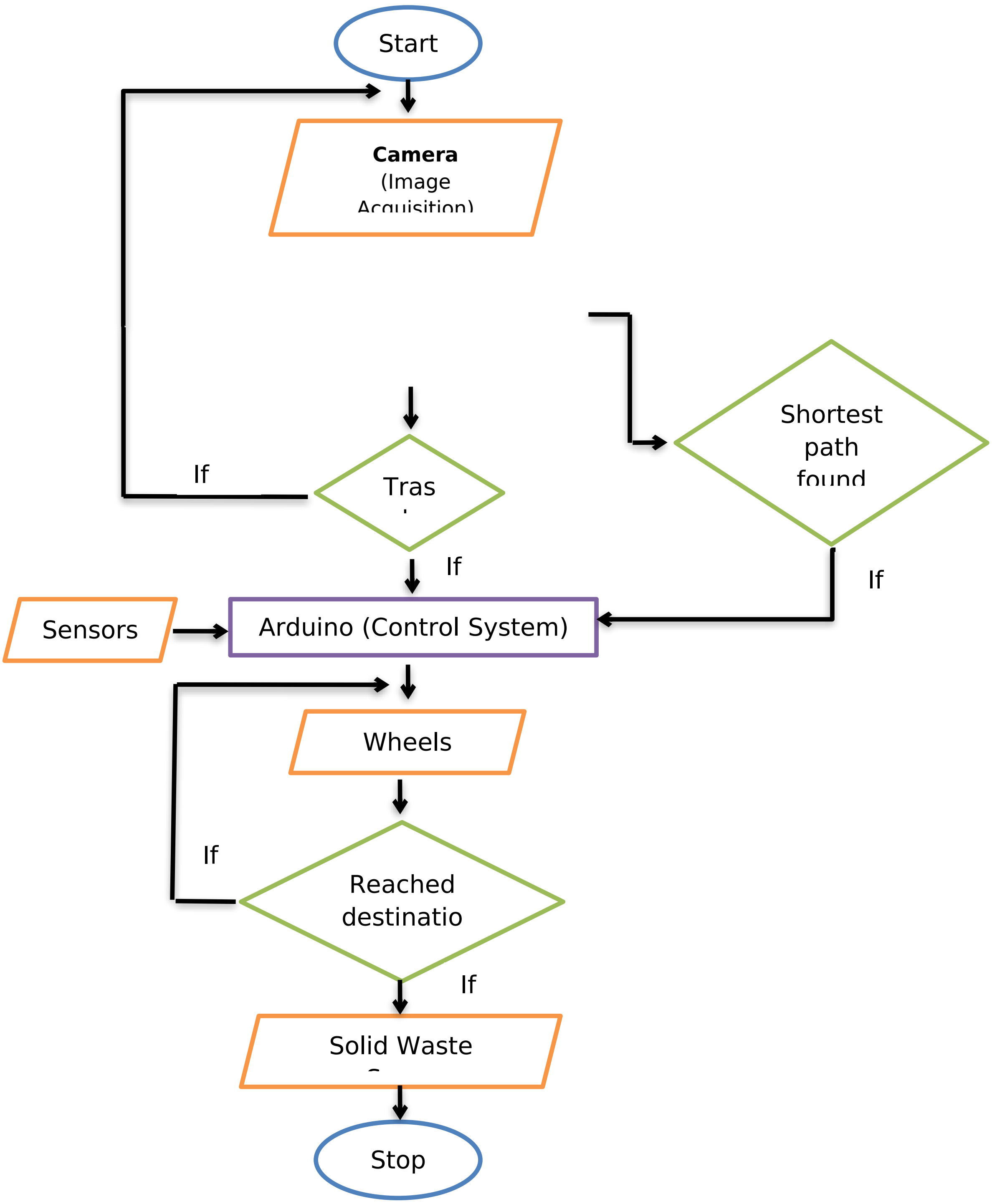
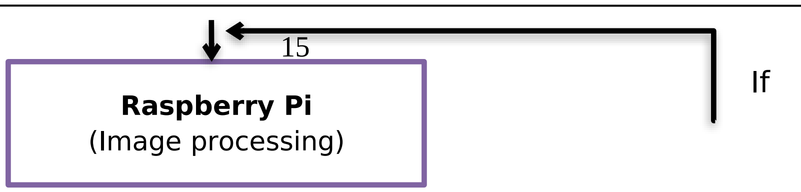



**Fig 3.2** Signal Flow Chart





### 3.3 Circuit Diagram

The fig 3.3.1 shows the circuit diagram designed for the bot. It consists of a combination of electronic and electrical components used in the bot which are listed later in this chapter.

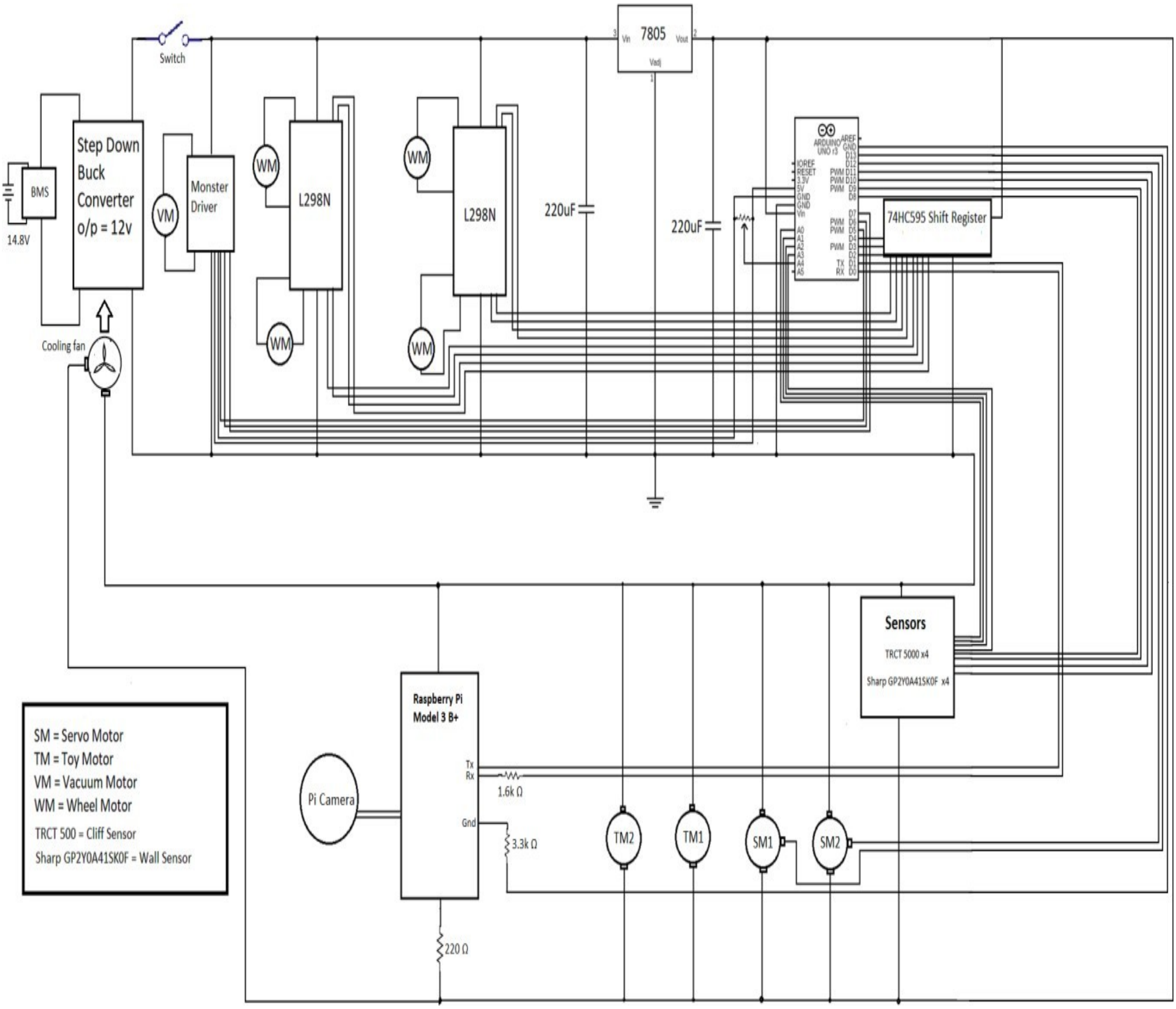



**Fig 3.3:** Circuit diagram of the bot.

The above circuit has been throughly analysed and tested so as to provide efficient output and control.

## 3.4Circuit Description

### 3.4.1 Power Supply

The bot consists of a Vacuum Motor, wheel motors, Arduino Uno, Sensors, Servo motor and Raspberry pi which are the major loads.

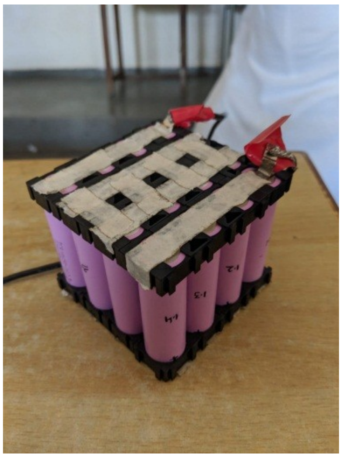


**Fig 3.4:** Battery Pack

### A. Power calculations

The bot and all the components used in the bot consume the following amount of power:

a) Vacuum - 100W
b) Arduino - 0.22W
c) Raspberry Pi- 3.234W
d) Sensors – 360mA x 5
= 1.8W

No. of motors used – 4

e) Power rating of each motor – 300mA

Total power from the motors = 2x 1200 mW
= 14400 mW
= 14.4W

Power Calculations of the Servo

Total Power = 1A x 5v = 5W

Total power consumed by the bot = 100+1.8+0.225+3.234+14.4+5
= 124.655W

**A. Current Drain of the bot**

a) Arduino – 48mA
b) Raspberry Pi – 980mA
c) Sensors – 20mA
d) Vacuum motor – (100W/12V) = 8.333A
e) Wheel Motors – 300mA x 4 = 1200mA
f) Servo Motor – 1A

B. **Total Current Drain** = 48mA+980mA+20mA+8.333A+1200mA+1A = 11.58A

Hence, due to the total power consumed by the bot conventional batteries could not be used in the project therefore a customized battery pack was created of the required specifications

**C. Battery Pack Specifications**

The battery pack designed for the project is made up of 1CR18650- 26K batteries which are manufactured by Samsung. The specifications of individual battery are:

a) Battery Capacity – 2600mAh
b) Nominal Voltage – 3.7V
c) Maximum Charge Voltage – 4.2V
d) Maximum Discharge Current – 5.2A

16 such batteries were used in the battery pack in a 4S4P configuration, i.e. 4 bat tries parallel and 4 such set of parallel connected batteries are then connected in series.

**D. Power consumption of the bot:**

Battery Capacity of the battery pack:

4 x 2600mAh = 10,400 mAh

Total voltage of the battery pack:

4 x 3.7V = 14.8V

Power = 14.8V x 10,400mAh = 153.92Wh

As you can see the battery pack provide more than enough power required for the bot.

**E. Running Time of the bot:**

$$\text{Time of operation} = \frac{powercapacityofthebatterypack}{totalpowerconsumed} = \frac{153.92\,Wh}{124.655\,W} = 1.23\text{hr.}$$

### 3.4.2 Step down Buck Convertor [9]

With power being a key parameter in many designs, step down or "buck" regulators are widely used. We are using the “buck” converter so as to get a stepped down constant voltage of 12V from the 14.8V source battery.

Although a resistor would enable voltage to be dropped, power is lost, and in applications such as the many battery powered items used today, power consumption is a crucial element. As a result step down switch mode converters or as they are more commonly termed, buck regulators are widely used.

The fundamental circuit for a step down converter or buck converter consists of an inductor, diode, capacitor, switch and error amplifier with switch control circuitry.

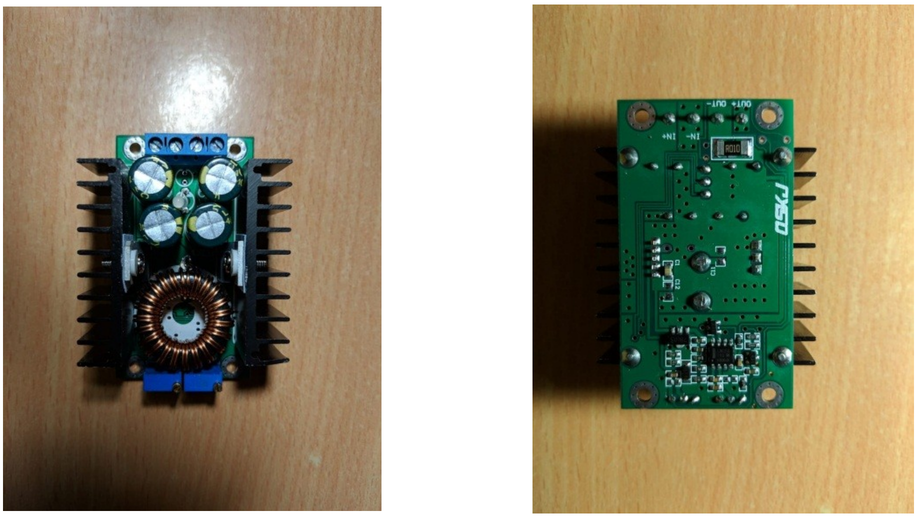


**Fig 3.5**: Front and back view of step down buck convertor

When the switch in the buck regulator is on, the voltage that appears across the inductor is Vin - Vout. Using the inductor equations, the current in the inductor will rise at a rate of (Vin-Vout)/L.

At this time the diode D is reverse biased and does not conduct.

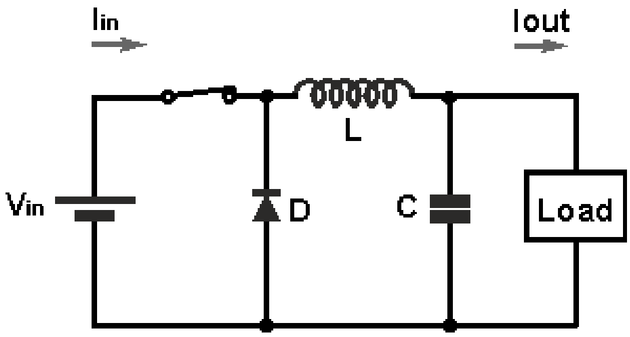



**Fig 3.6**: buck convertor when s/w closed[9]

When the switch opens, current must still flow as the inductor works to keep the same current flowing. As a result current still flows through the inductor and into the load. The diode, D then forms the return path with a current Iodide equal to Iout flowing through it.

With the switch open, the polarity of the voltage across the inductor has reversed and therefore the current through the inductor decreases with a slope equal to -Vout/L.

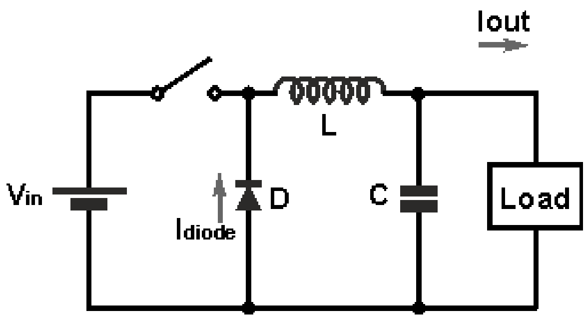



**Fig 3.7**: buck convertor when s/w open[9]

### 3.4.3 Monster Driver

Since the vacuum motor requires a current input of 8.33A, which cannot be fulfilled by a regular motor driver we are using the “Monster Driver” which as the name suggest has the capacity to handle high Currents.

**Fig 3.8**: Monster driver IC

Monster Moto Shield VNH2SP30 Motor Driver 14A is essentially a ramped up version of our Ardumoto motor driver shield. For this Monster Moto Shield, we have replaced the L298 H-bridge with a pair of VNH2SP30 full-bridge motor drivers.

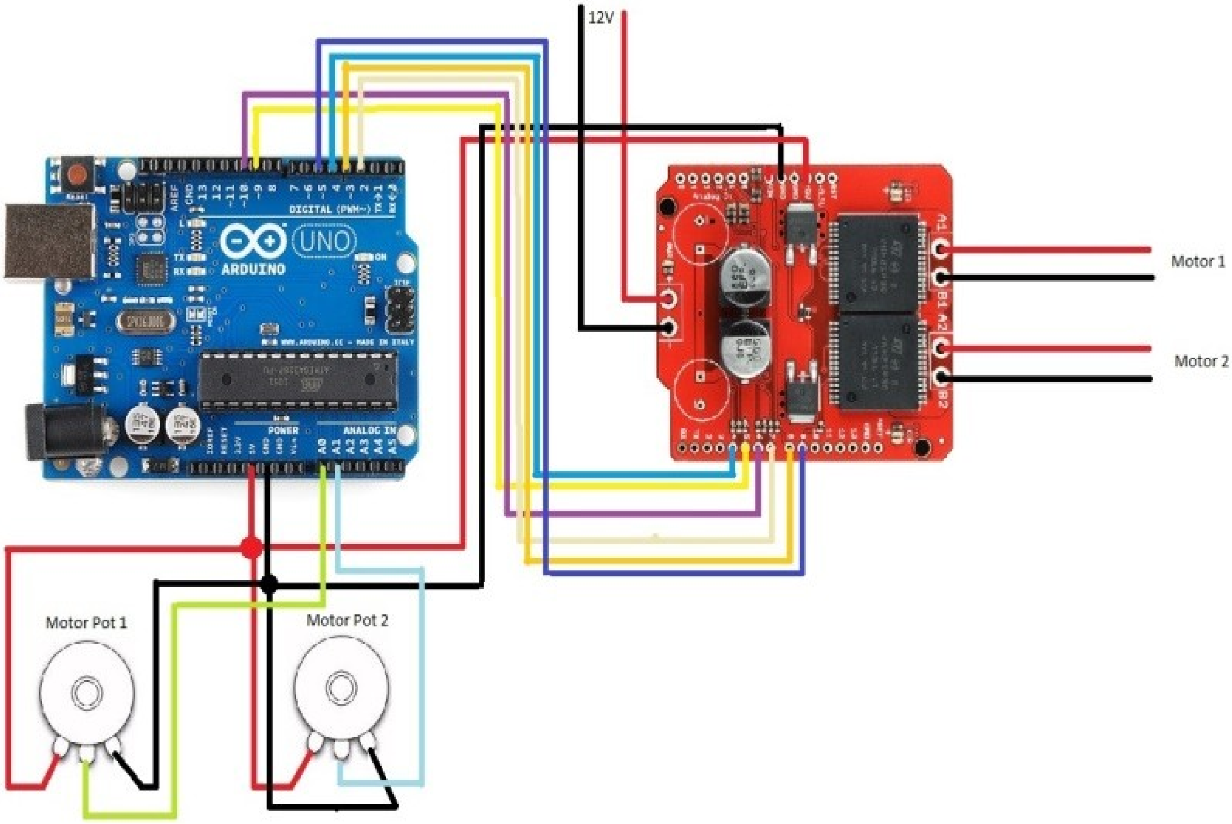



**Fig 3.9:** Monster driver connection circuit [7]

**Features:**

1. Current sensing available to Arduino analog pin
2. MOSFET on-resistance: 19 mΩ (per leg)
3. Maximum PWM frequency: 20 kHz
4. Thermal Shutdown
5. Under voltage and Overvoltage shutdown.

Fig 3.4.4 shows us the proper way of connecting the monster driver to the motors and Arduino Uno. We also use 5V potentiometers at the input so as to control the voltage supply and also the PWM connection enables speed control.

### 3.4.4 L298N Motor Driver IC

Coming to the Wheels, they are connected to the bot with reduction gear 60rmp motors of 12V DC voltage requirement. These do not require a high current output so the optimum motor driver was L298N as it can control 2 motors at a time along with direction changing capability.

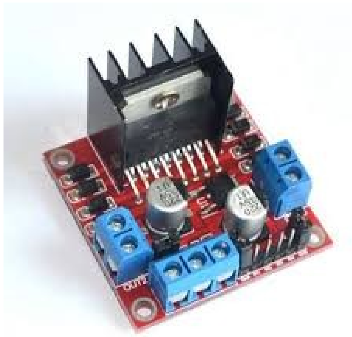


**Fig 3.10**: L298N motor driver IC

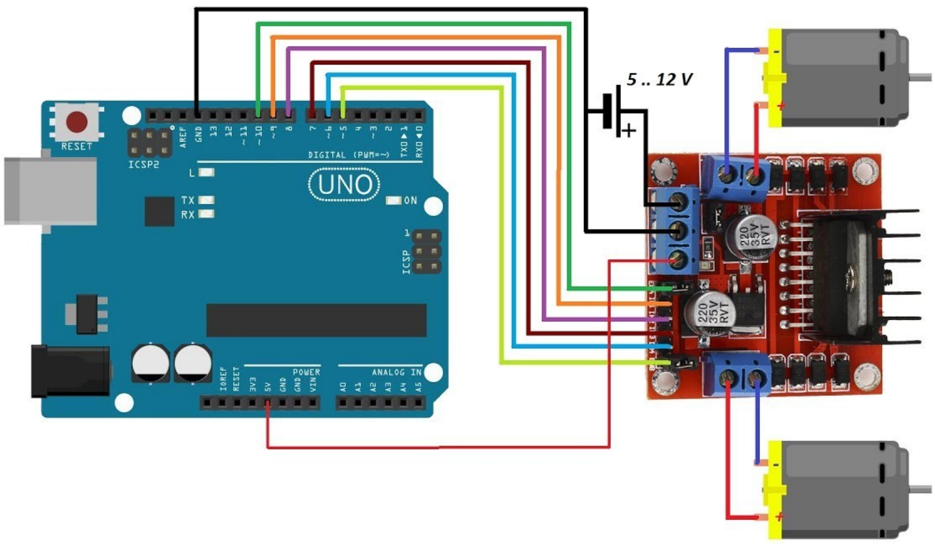



**Fig 3.11**: L298N motor driver IC connections

The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input

signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.
The above circuit diagram explains how to connect the motors to the L298N driver. It basically has 2 pins as control pins which are connected to the Arduino for direction control.

### 3.4.5 IC 7805 – Voltage regulator

The Sensors, Arduino Uno, Raspberry Pi and the Servo motor all operate under 5V and hence to reduce the bot voltage from a constant 12V DC to constant 5V DC we are using the LM7805 IC with a low pass filter.

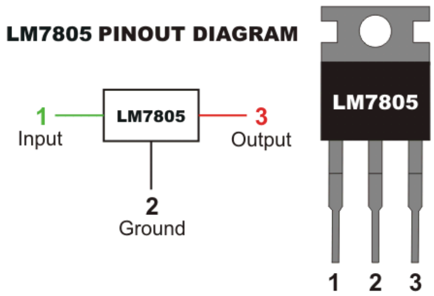



**Fig 3.12**: Voltage Regulator IC 7805

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

### 3.4.6 74HC595 Shift Register

The Arduino Uno come equipped with 14 digital pins and 6 analog pins, but for our problem statement and circuit diagram, we required more than the provided number of pins and hence we are using a 3x8 shift register for pin expansion.
This IC, i.e. 74HC595Shift Registeris connected to the Arduino with 3 pins and can give the equivalent of an 8pin Pin out. If need be, we can further extend the pins of Arduino Uno by connecting multiple shift registers in a daisy-chain chan configuration.

This enables us to have virtually as many pin in/outs as needed.

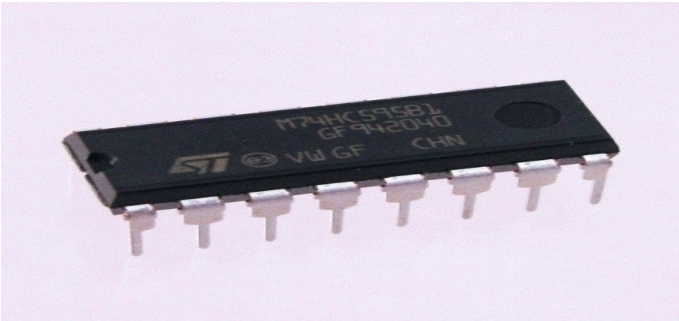



**Fig 3.13**: 74HC595 Shift Register

The shift register holds what can be thought of as eight memory locations, each of which can be a 1 or a 0. To set each of these values on or off, we feed in the data using the 'Data' and 'Clock' pins of the chip.The clock pin needs to receive eight pulses. At the time of each pulse, if the data pin is high, then a 1 gets pushed into the shift register. Otherwise, it is a 0. When all eight pulses have been received, then enabling the 'Latch' pin copies those eight values to the latch register. This is necessary; otherwise the wrong LEDs would flicker as the data was being loaded into the shift register.

The chip also has an OE (output enable) pin, this is used to enable or disable the outputs all at once. You could attach this to a PWM capable Arduino pin and use 'analog Write' to control the brightness of the LEDs. This pin is active low, so we tie it to GND.

### 3.4.7 Analog Distance Sensors

**a) Sharp IR Sensor (GP2Y0A21SK0F)**

We are using the Sharp IR Sensor (GP2Y0A41SK0F) as wall sensors i.e. for obstacle detection. We have chosen these as they have greater efficiency than regular quality IR sensors and are more accurate with the distance measurement range of 4-30 cm. We are using a total of 4 of them on the 4 sides of the bot for optimum coverage.

The Sharp distance sensors are a popular choice for many projects that require accurate distance measurements. This IR sensor is more economical than sonar rangefinders, yet it provides much better performance than other IR alternatives.

Interfacing to most microcontrollers is straightforward: the single analog output can be connected to an analog-to-digital converter for taking distance measurements, or the output can be connected to a comparator for threshold detection.

**Fig 3.14**: Sharp IR Sensor (GP2Y0A21SK0F)

**b) TRCT 5000 Sensor**

Coming to the part of ‘Cliff Sensor’ i.e. the sensors used to detect bumps and holes/stairs on the floor we are using the TRCT 5000. It is comparatively cheaper and less accurate than the Sharp IR Sensor (GP2Y0A41SK0F), but serves us the purpose as drastic height changes in an indoor floor happen very rarely and often only while encountering stairs.

**Fig 3.15:** TRCT 5000 Sensor

. TCRT5000 is a reflective sensor which includes an infrared emitter and phototransistor in a leaded package which blocks visible light. Its distance measuring range is 15mm.

### 3.4.8 Servo motor

We are using a servo motor to drive the flap mechanism of the solid waste collection part of the bot. It is connected to the Arduino Uno which sends the appropriate control signal as a PWM signal so as to open the flap at a predefined angle and closes it when needed. To properly establish a contact between the flap and the Servo we have designed an appropriate end effector.

A **servo motor** is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through **servo mechanism**.

If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

**Fig 3.16:** S3003 Servo Motor

### 3.4.9 Fish Eye Camera

A **fisheye lens** is an ultra wide-angle lens that produces strong visual distortion intended to create a wide panoramic or hemis pherical image.145 fisheye lenses achieve extremely wide angles of view. Instead of producing images with straight lines of perspective (rectilinear images), fisheye lenses use a special mapping (for example: equisolid angle), which gives images a characteristic convex non-rectilinear appearance.

**Fig 3.17**: Raspberry Pi Fish eye camera

We are using a fisheye lens with Raspberry pi so as to achieve a wide angle view of the surrounding so as to cover larger ground in terms of the bots visibility.

### 3.4.10 Battery Management System

**Fig: 3.18: 4s BMS Chip**

A **battery management system** (**BMS**) is any electronic system that manages a rechargeable battery (cell or battery pack), such as by protecting the battery from operating outside its safe operating area, monitoring its state, calculating secondary data, reporting that data, controlling its environment, authenticating it and / or balancing it.

## 3.5 Factors to be considered in product design

There are many factors to be considered while solving a design problem. Some of the factors to be considered are as follows :

**Material**: Material plays a very important role while designing. If a wrong material is selected, it tends to fail. Material should be chosen based on how the manufacturing has to be carried out and also depends on the budget.

**Load:** Stresses must be determined accurately as it is responsible for the internal stresses. Different types of loads are static load, dynamic load, etc.

**Size, Shape, Space Requirements and Weight:** Size, shape, space and weight are very crucial factors that should be taken care of. If the size of the component is too large than required, it will occupy more work space area and might also result in a high cost. Weight mainly depends on the selected material.

**Manufacturing**: A feasible manufacturing process suitable to the design of the component must be implemented.

**Operation:** The operation of any component should be as simple as possible. For example, usage of knob or a switch easily starts the machine. In some of the cases number of operations has to be done to get the machine started. The sequence of operating a machine should be designed in such a way that it is user-friendly.

**Reliability and Safety:** The designed component should work effectively without facing any failure while operating it. Proper analysis of the machine should be done prior to check for its reliability. Excessive heat generation, overloading, wear of elements should be avoided for the proper functioning of the component. Safety is the other important thing that should be taken care of.

**Maintenance, Cost and Aesthetics:** Maintenance and safety are linked to each other. If the machines are maintained well in a timely fashion, they will be safe to use resulting in low maintenance future costs possible. Cost of the machine should not be too high or too low. The overall cost of the machine depends on the selected material, machine design and components, and manufacturing process. Aesthetic features are not necessary for machine unless they serve a purpose.

## 3.6 Layout of the bot

The Fig 3.21 is the top view of the bot. Considering various parameters like surroundings where the bot is used, compatibility, human comfort, automatic bots in the current market etc.

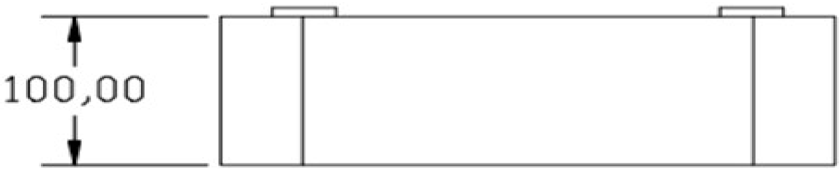



**Fig 3.19:** Flap Mechanism

The dimensions were fixed to 50cm*30 cm. The positioning of the four mecanum wheels, which supports the bot, was decided considering the dynamic stability of the same. Since battery is the heaviest of all the components it was placed exactly at thecentre of the frame. The remaining place in the bot can accommodate all the electrical components.Lets divide the bot as the front part i.e. Above the battery and bottom part i.e. below the battery.

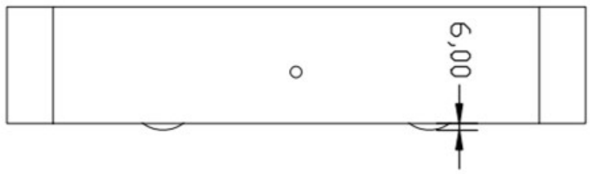



**Fig 3.20:** Side view of the bot.

The front part has the vacuum motor assembly and two holes for the motors which are coupled to the side brushes used for agitating the dust. Based on the suction pressure of the 100W vacuum motor the area for the vacuum suction was decided. The figure below shows the dimensions of the vacuum suction opening. A duct of suitable dimensions is welded to the frame on the opening on top of which the dust collecting box is placed which are connected together with a hosepipe of 2.5cm diameter. This box has the mounting points of the vacuum motor fan.

The wheels are coupled to the motors and are attached to the bot with clamps of suitable radius. Rubber sheets of suitable thickness are used to reduce the vibrations caused by the mo The special ability of these wheels are that the bot can be moved in different direction without any change in its orientation.

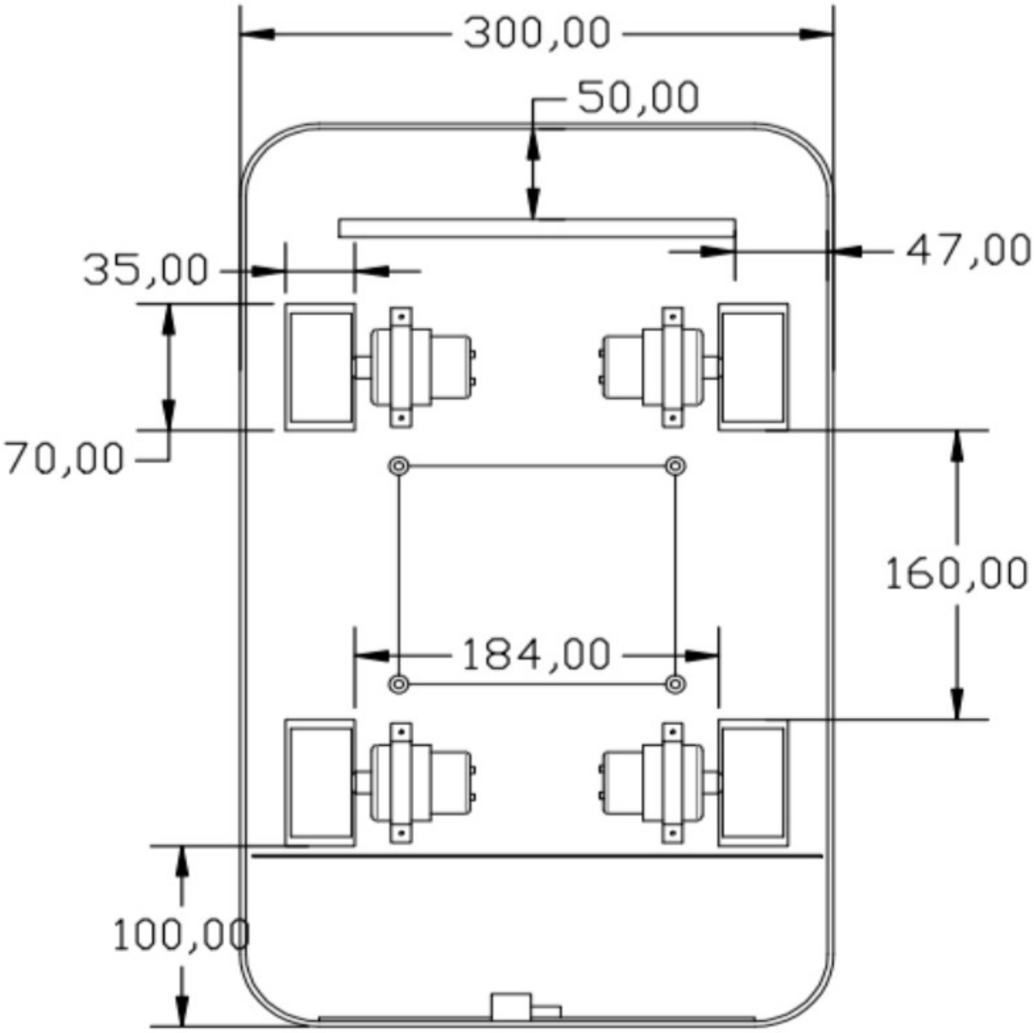



**Fig 3.21:** Top view of the bot.

These wheels offer ten degrees of motion thus giving us zero turning radius which is desirable in spaces. The motion control of this vehicle is introduced from the resolved force control. vent of the bot. The clearance given to the bot is 0.6cm.

The tractive force at each wheel will determine the resultant tractive force that propels the vehicle to any direction. This is achieved with the help of logics given to the motor drivers that control the wheels.

**Table 3.1**: Control Logic of mecanum wheels

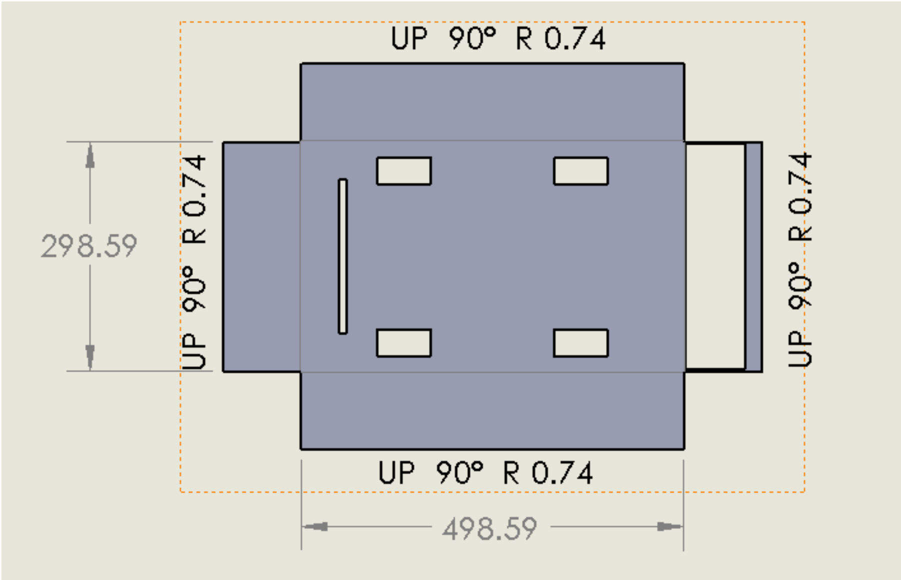
| BOT | FRONT WHEELS | | REAR WHEELS | | L298N | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOVEMENT (DIRECTION) | LEFT (1, 2) | RIGHT (3,4) | LEFT (1’, 2’) | RIGHT (3’, 4’) | FRONT | | | | REAR | | | |
| | | | | | 1 | 2 | 3 | 4 | 1’ | 2’ | 3’ | 4’ |
| FORWARD(↑) | ↑ | ↑ | ↑ | ↑ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| BACKWARD(↓) | ↓ | ↓ | ↓ | ↓ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| RIGHT(→) | ↑ | ↓ | ↓ | ↑ | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| LEFT(←) | ↓ | ↑ | ↑ | ↓ | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| NORTHEAST(↗) | ↑ | - | - | ↑ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| NORTHWEST(↖) | - | ↑ | ↑ | - | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| SOUTHEAST(↘) | - | ↓ | ↓ | - | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| SOUTHWEST(↙) | ↓ | - | - | ↓ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| CLOCKWISE(↻) | ↑ | ↓ | ↑ | ↓ | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| ANTICLOCKWISE(↺) | ↓ | ↑ | ↓ | ↑ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

The bottom part has the assembly for solid waste collection run by the servo motors. A rectangular cut-out of size 22cm *10cm is provided for the collection of solid waste. The flap is mounted on the chassis with two hinges run by a servo motor which is used to push the solid waste inside. The edge of the chassis is tapered to $45^0$ which eases the action of pushing.

## 3.7 Fabrication process:

One of the major challenges is to design the bot as light and compact as possible without compromising on its strength. Keeping this in mind, aluminum sheet of thickness 2mm has been selected as the base material of the bot.

Initially a sheet of 740mm*540mm is taken and cut-out to the section as shown in figure(fig no) followed by fabricating the slots for wheels, vacuum suction area and solid waste




collection using a cutting machine.

**Fig 3.22:** Cross section of the chassis.

The edges are 120mm from the top at each side are bent to 90 degrees and are welded at the intersection using an electrode grade 6403 such that these form the side walls of the bot. Now buffering operation is carried out at the welded portions to give it a smooth and neat finish. A vacuum duct with dimensions 210*50*20mm is welded to the bot over the dust collection slot which has a hole of 25mm drilled (on its to surface) to accommodate a pipe of same size.

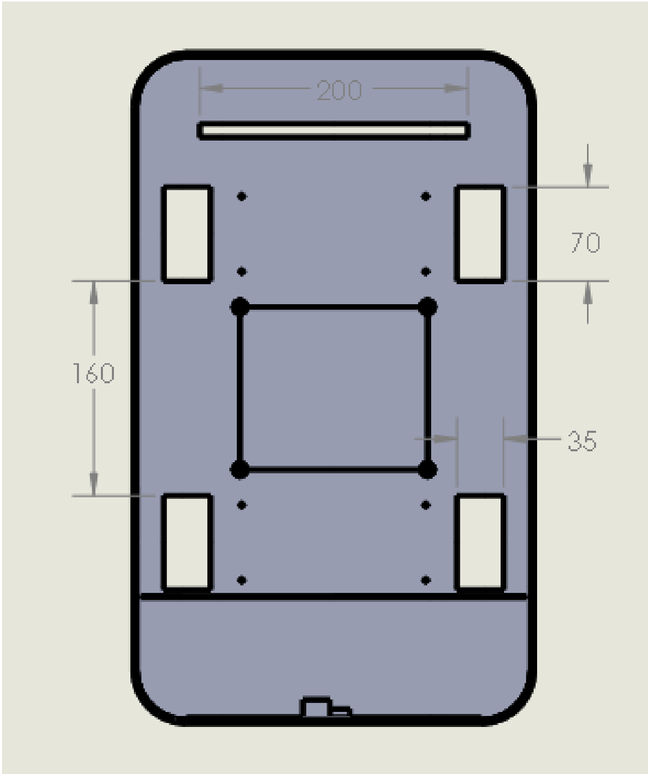



**Fig 3.23:** Top view of the bot.

The hosepipe used forms a connection between the vacuum duct and the dust collecting box which is held at its position by using glands of suitable size at each junction. On the other side of the dustbin a vacuum motor is mounted which creates the suction.

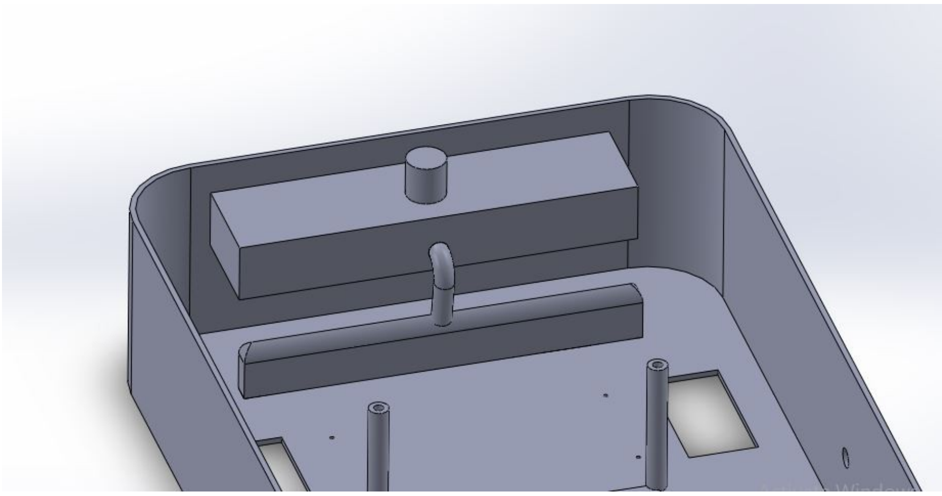


**Fig 3.24:** Vacuum suction assembly.

The bot is designed such dust collection is at the front end of the bot and at the rear end the solid waste collection is done. For collecting the solid waste, a flap mechanism, riveted to the frame using two 11⁄2 inch hinges is designed. Two servo motors are used to open-close the

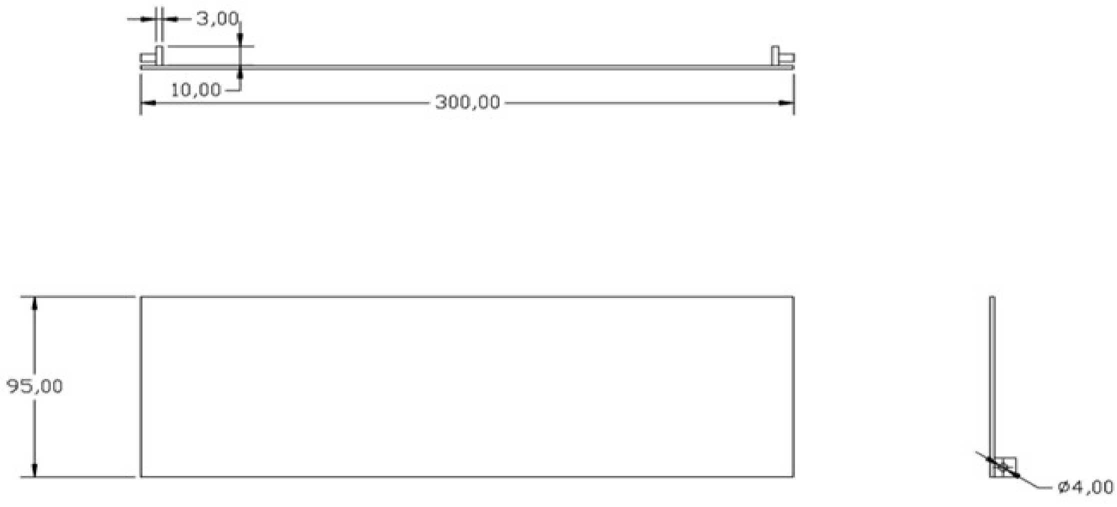



**Fig 3.25:** Flap.

Flap which can cover an angle of 90-95 degrees. The rivet used is of length 10mm and 3mm diameter.

Mecanum wheels facilitate the motion of the bot. Each wheel is coupled (made of aluminum) to a 12 v 60rpm reduction gear motor.

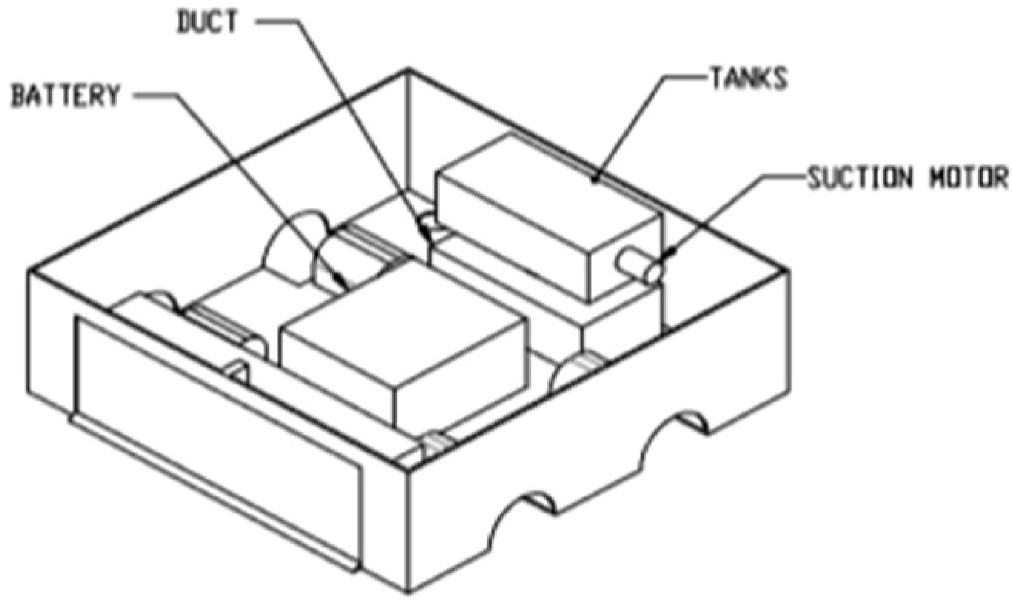



**Fig 3.26:** Flap Mechanism.

To fabricate the coupling an aluminum rod is made to undergo lathe operations like turning, facing and knurling. One side of the coupling (lesser diameter) is press-fit into the wheel. The other side is drilled along its axis to press-fit the motor shaft and is screwed together.

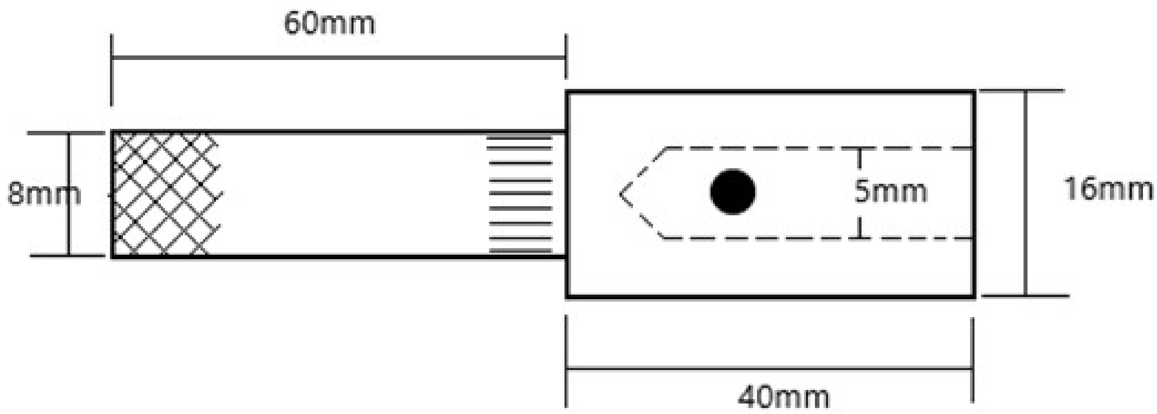



**Fig 3.27:** Wheel Coupling.

The wheel -motor assembly is mounted to the frame using clamps of suitable size. A clearance of 6mm is achieved for efficient vacuum suction and solid waste collection.

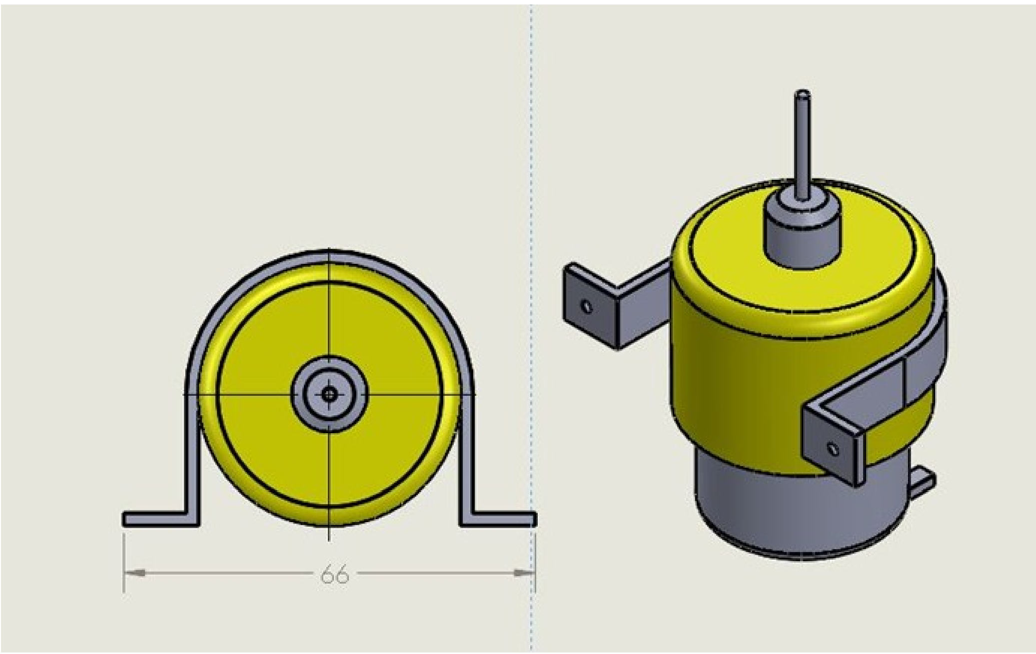


**Fig 3.28:** Dc motor with 'C' clamp.

Two cut-outs are made for mounting 2 side brushes operated by 3.3V motors at the front corner of the bot. These help to agitate the dust at the wall corners and along the edges of the wall.

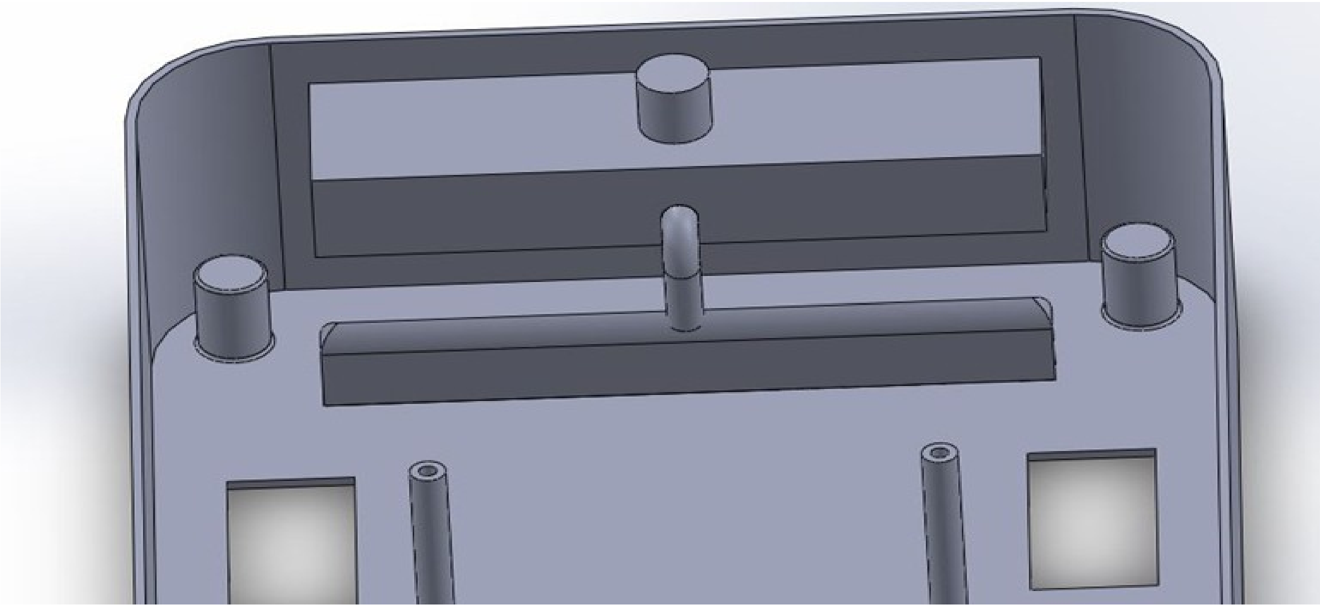


**Fig 3.29:** Vacuum suction assembly with side sweeper brushes

Battery being the heaviest among all the components it is placed at the Centre for better dynamic stability.

Battery being the heaviest among all the components it is placed at the Centre for better dynamic stability.

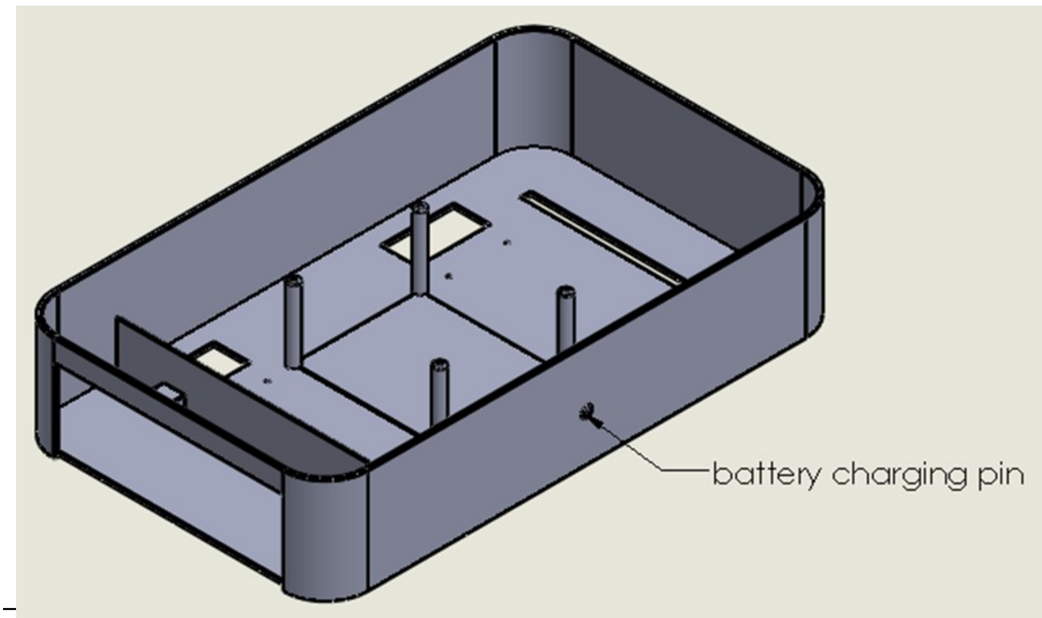


**Fig 3.30:** Battery Charging Port.

A special provision is given at this area to increase the stiffness of the sheet and avoid bending. A hole is drilled on the side wall for plug-in charge.

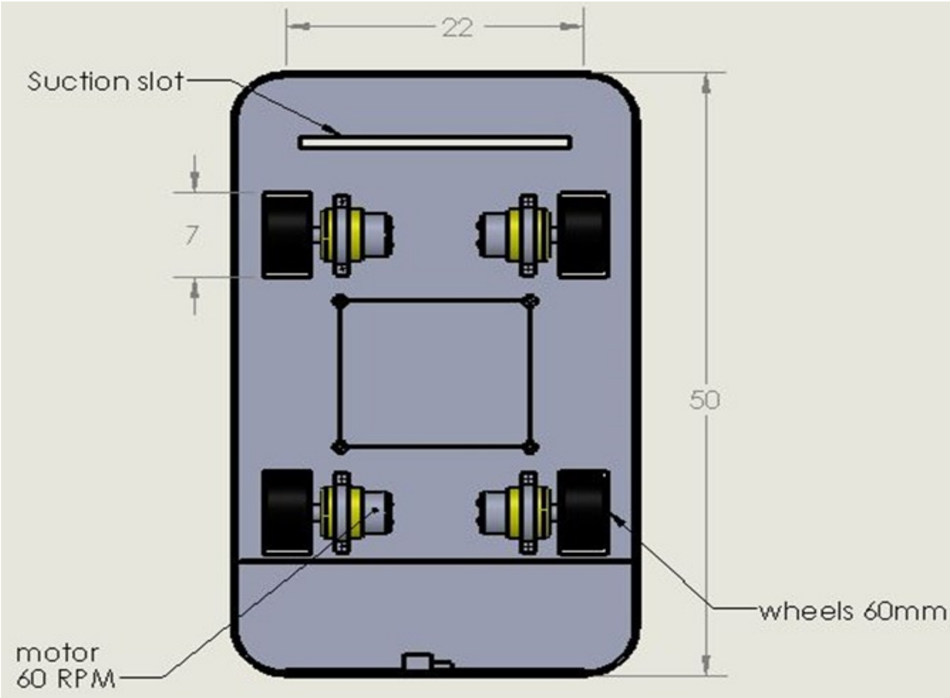



**Fig 3.31:** The assembled bot

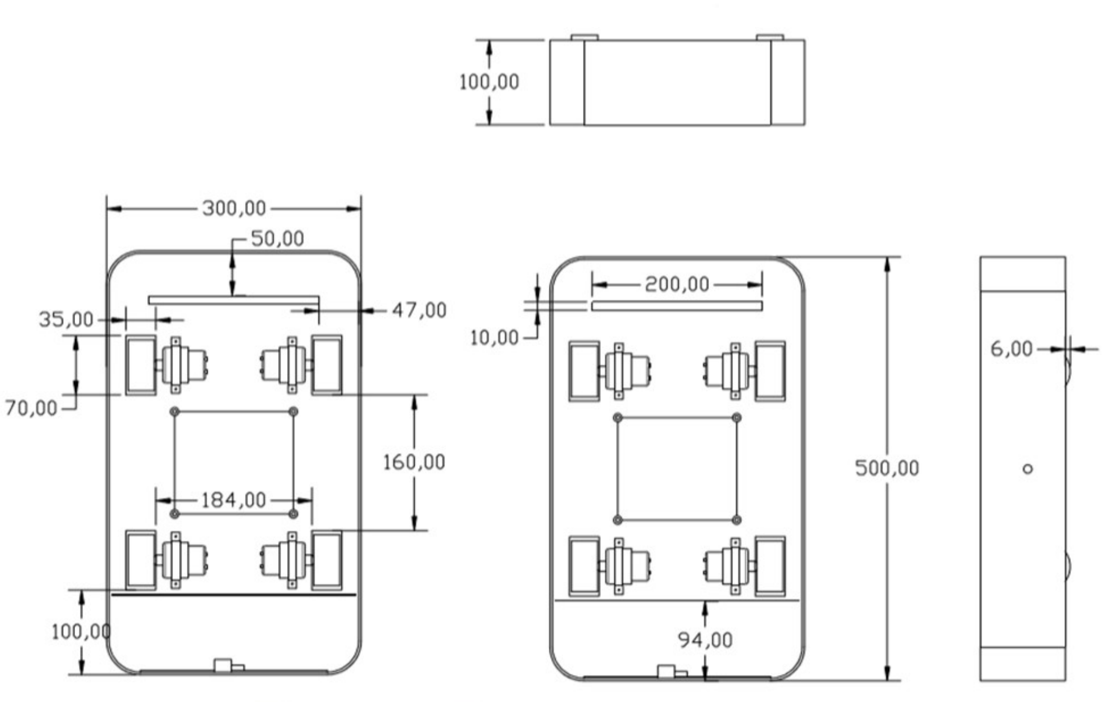


**Fig 3.32:** Layout of the assembled bot.

**Powder coating** is a type of coating that is applied as a free-flowing, dry powder. The main difference between a conventional liquid paint and a powder coating is that the powder coating does not require a solvent to keep the binder and filler parts in coating and is then cured under heat to allow it to flow and form a "skin".

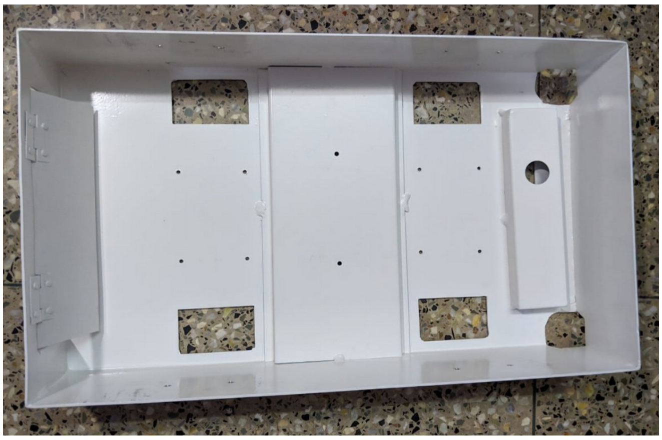


**Fig 3.33:** The bot after powder coating

The powder may be a thermoplastic or a thermoset polymer. It is usually used to create a hard finish that is tougher than conventional paint. Powder coating is mainly used for coating of metals, such as household appliances, aluminum extrusions, drum hardware and automobile and bicycle parts. Newer technologies allow other materials, such as MDF (medium-density fiberboard), to be powder coated using different methods.

## 3.8 Mechanical components

### 3.8.1 Side Sweeper brushes

The **spinning side brush** sticks out past the Roomba shell to reach spots the underside can't access. It spins along walls to kick up dirt and direct it into the vacuum area.

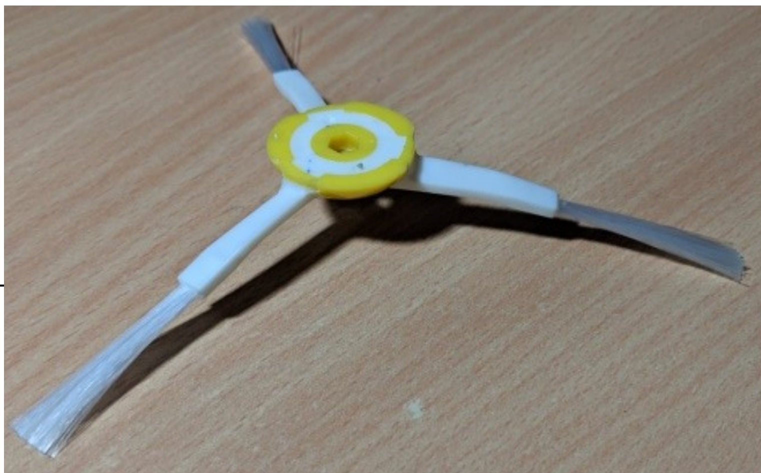


**Fig 3.34:** Side Sweeping brush

The brush on the opposite side of the Roomba directs any wayward dirt back under the unit to be sucked up. We have accommodated two brushes near the head of the bot on either side of the suction point that rotate in counter clock-wise direction. These are coupled to 5V DC motor.

### 3.8.2Vacuum pump

It is mechanical machinery that creates negative pressure which helps in sucking air. Vacuum pump exchanges the mechanical input power rotating shaft into pneumatic or hydraulic power by evacuating the air or liquid contained in a system. The pressure levels thus become lowered than the outside atmospheric pressure.

The amount of power produced solely depends on the volume of air evacuated and the pressure difference being produced.

**Fig 3.35:** Vacuum Motor

**Fig 3.36:** Vacuum fan.

In mechanical vacuum pumps the mechanism is so designed that air or liquid is sucked from closed area and being thrown to atmosphere. The major specialties of vacuum pumps are that the pressure here is below atmospheric pressure. The low pressure is achieved by moving a cycle of blades by a motor. We are using a 12V DC Motor with the power capacity of 100W as our Vacuum Motor. It consumes 8.33A of current and has a suction pressure of about 4300-4500PA. We are using this because it has high suction capacity at 12V DC and is pretty efficient. It produces low noise at around 75db.

### 3.8.3 Vacuum System

The assembly comprises of a flexible hose pipe and a dust collection box. The diameter of the hose pipe is 25mm. The dimensions of the dust collection box are 90mm in height and 80mm in diameter. The box has been divided into two compartments separated by a filter.

**Fig 3.37:**

### 3.8.4 5V DC Motor

5V DC motors are used for the Side rolling brush. It provides 2000 rpm which is enough to agitate the dust near the vacuum suction area. We are using this so as to reach the edges ofthe desired area to clean and the small size of the motor is optimum for this purpose.

**Fig 3.38:** 5V DC Motor.

It does not require any control signal and runs continuously until the power is cut down. These motors are pretty efficient and have a very long life which helps us with our purpose. This is being connected to the LM7805 IC as it provides a constant 5V DC output. The gear head on top of it is removable and it makes the motor a bit modular. Generally a 220Ω resistor is connected in between its two terminals while connecting to a 9V battery, which is eliminated in our circuit diagram due to the usage of LM7805 IC.

### 3.8.5 Mecanum wheels

One of the common Omni-directional wheel designs is Mecanum Wheel or Ilon wheel [8]. Mecanum wheel was design and invented in Sweden in 1975 by BengtLlon, an engineer with Swedish company Mecanum AB. Mecanum wheel is based on the principle of a central wheel with a number of roller placed at an angle around the periphery of the wheel. The angled peripheral roller translates a portion of the force in the rotational direction of the wheel to force normal to the wheel directional.

Depending on each individual wheel direction and speed, the resulting combination of all these forces produces a total force vector in any desired direction thus allowing the platform

to move freely in direction of resulting force vector, without changing the direction of the wheel. Figure 1 shows a traditional Mecanum wheel design by Ilon with the peripheral roller with 45° degree slope held in place from the outside.

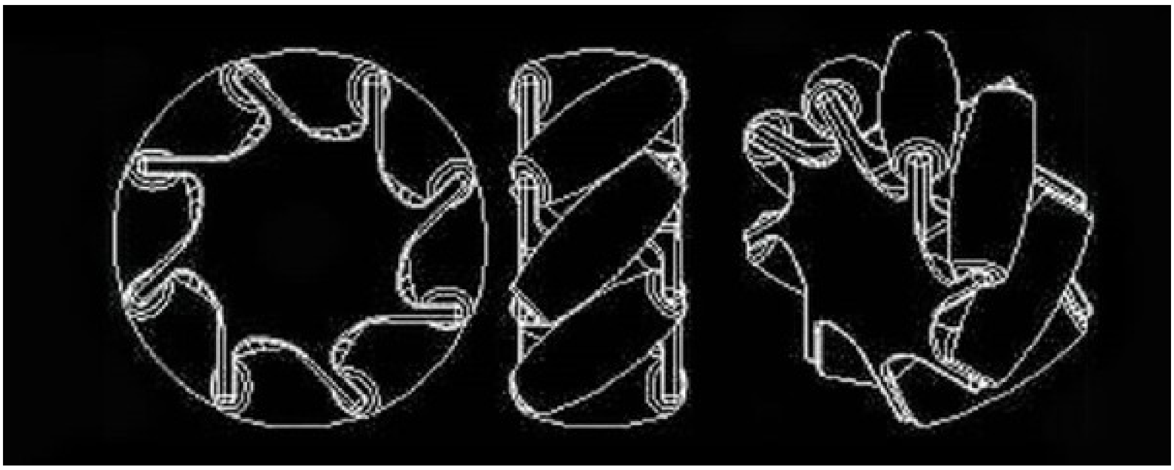


**Fig 3.39:** Mecanum wheel based on Ilon’s concept.

This design only can operate in even work surface [6]. When encountering an inclined or an uneven work surface, the rim of the wheel can make contact with the surface instead of the roller, thus preventing the wheel from operating correctly.

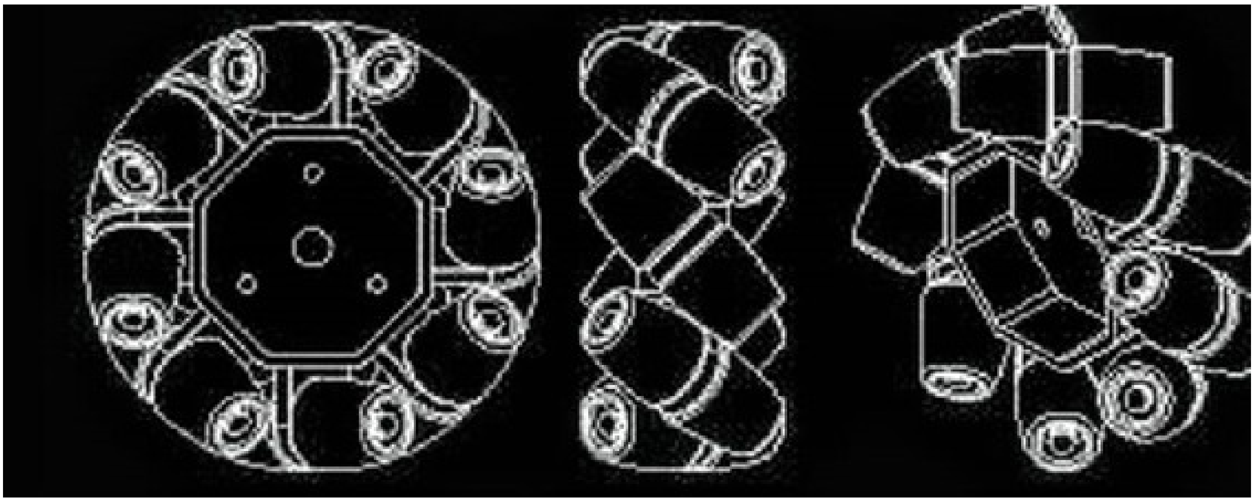


**Fig 3.40:** Mecanum wheel with centrally mounted rollers.

To encounter this problem a simple alternative design, also proposed by Ilon, which consist two spilt roller mounted centrally on the periphery of the wheel.

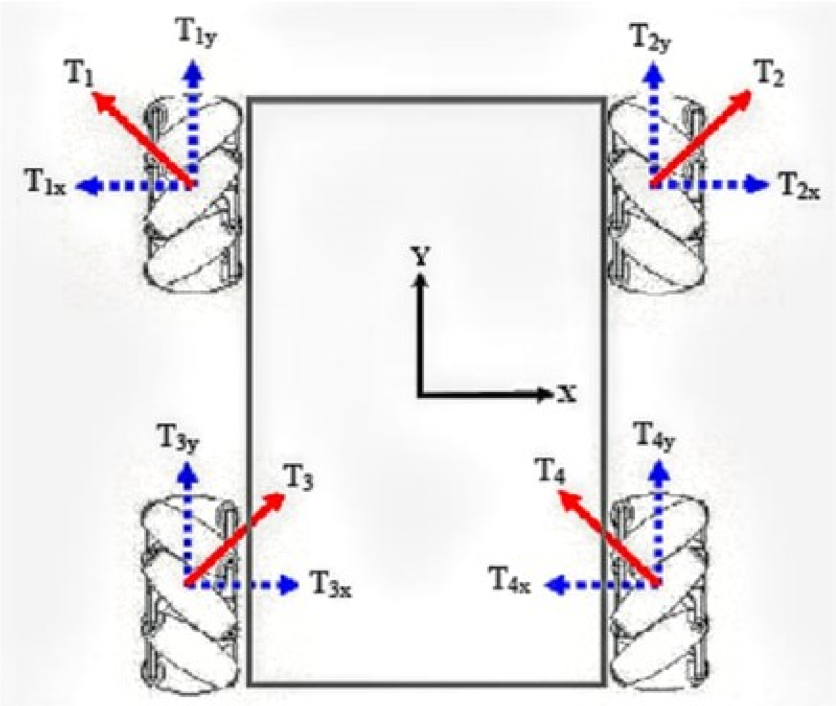



**Fig 3.41:** Mecanum wheel vector diagram.

This design ensures that the rollers are always in contact with the work surface, thus allowing better performance on uneven surfaces. Using four of Mecanum wheels provides Omni-directional movement for a vehicle without needing a conventional steering system. Slipping is a common problem in the Mecanum wheel as it has only one roller with a single point of ground contact at any one time. Due to the dynamics of the Mecanum wheel, it can create force vectors in both the x and y-direction while only being driven in the y-direction. Positioning four Mecanum wheels, one at each corner of the chassis (two mirrored pairs), allows net forces to be formed in the x, y and rotational direction. Refer to Figure 3. A difficulty with this strategy is that there are four variables to control three degrees-of-freedom. In this case the system is said to be over determined and it is possible to create conflicts in the actuation. As a result of the constraints associated with the Mecanum wheel some form of controller is required to produce satisfactory motion.

To get the most out of a Mecanum drive system you will need to have the following information available to control it:

- Desired Angle – What angle the robot needs to translate at
- Desired Magnitude – What speed the robot must move at
- Desired Rotation – How quickly to change the direction the robot faces

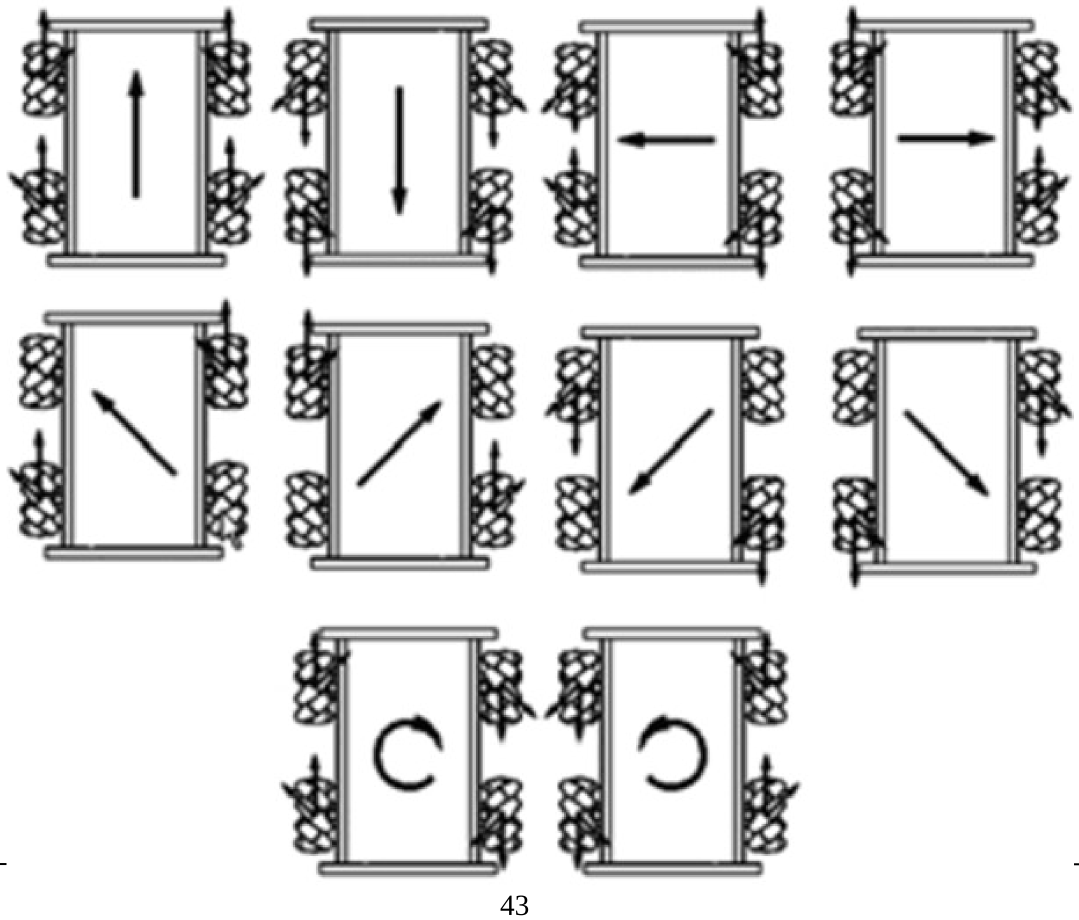


**Fig 3.42:** Mecanum wheel direction vectors.

The control of the robot is possible in two ways: using infrared remote control or using a wireless PS2 joystick. Achievement of the main movement directions is possible by changing the wheels rotation directions. For example the forward movement is possible when all four wheels are driven in the same direction. In Fig 3.5.4.3 are specified the rotation direction for each wheel and the driving directions for each combination. In the Fig 3.5.4.4 is presented the disposition of the wheels and the frames Σ0, Σiw (i = 1, 2, 3, 4), respectively. Viw (i = 1,2,3,4) is the velocity vector corresponding to wheel revolutions, where Viw = ωiwRw, RW, is the radius of wheel, ωiw is the revolution velocity of the wheel, and Viw (i = 1,2,3,4) is the tangential velocity vector of the free roller touching the floor. The velocity of the robot vx, vy and ωzar given by the relation [8]

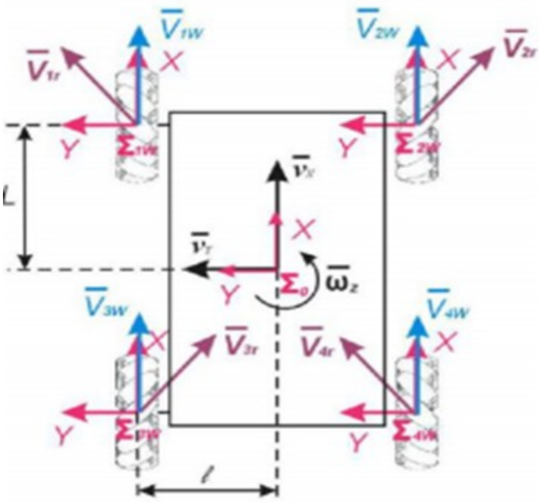



**Fig 3.43:** Mecanum wheel direction vectors for calculation.

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega_z(t) \end{bmatrix} = \frac{R_W}{4} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{L+l} & \frac{1}{L+l} & -\frac{1}{L+l} & \frac{1}{L+l} \end{bmatrix} \cdot \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix}$$

The angular velocity of the wheel is given by the relation:

$$\begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{pmatrix} = \frac{1}{R_W} \cdot \begin{bmatrix} 1 & -1 & -(L+l)/2 \\ 1 & 1 & (L+l)/2 \\ 1 & 1 & -(L+l)/2 \\ 1 & -1 & (L+l)/2 \end{bmatrix} \cdot \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix}$$

### 3.8.5 Traction in Mecanum wheels

It has been proved that a Mecanum Wheel has less traction than an ordinary standard wheel. To prove, the theory has been described below. Figure 3.43 below shows the bottom view of he front port-side of the Mecanum Wheel roller which is in contact with the floor.

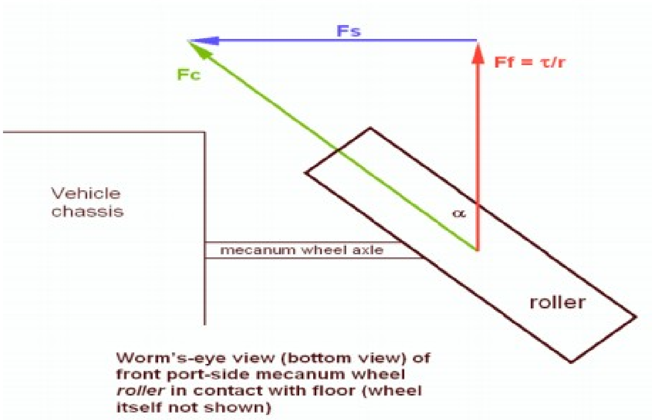



**Fig 3.44:** Bottom view of Mecanum wheel roller

Fc= Reaction force produced by the floor on the bottom of the Mecanum Wheel roller which is in contact with the floor. This force should also be aligned along the roller axis, assuming that roller bearings do not produce friction.

Fc also has two components as shown in Figure 3.44, which are Fs and Ff. Where, Fs is the force which is counterbalanced by an equal but opposite force from the wheel on the other side of the vehicle when that wheel is being driven with the same torque. Ff is the forward component which is in the plane of the Mecanum Wheel. When wheel is not in motion component Ff should be equal to $\tau/r$. $\tau$ is the driving torque applied to the Mecanum Wheel and r is the radius of the Mecanum Wheel. Forward force produced by the Mecanum Wheel is same as the force produced by an ordinary wheel which has the same diameter and driven with the same torque. The only difference is that the force of the floor acting on the roller is greater than an ordinary wheel by factor $1/\cos\alpha$. This reaction force is high which breaks the friction and slip associated with Mecanum Wheel with the floor [11].

Considering the case where the vehicle is moving forward, the rear port side of the wheel is driven with the same magnitude as the front wheel but in backwards direction. Ff wheel is counterbalanced by equal and opposite force which is due to the back wheel of the vehicle. Fs remains I n the same direction for both front and rear wheels. Considering no bearing friction the pushing force of the vehicle remains same for both fore and sideways direction [11].

The Figure 3.45 below describes the vectors in dotted lines which are for reference

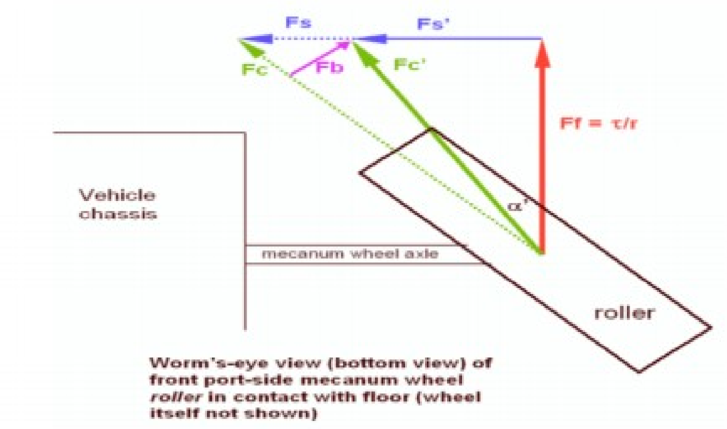



**Fig 3.45:** Change in Vector Components due to Forward Motion

When the wheel moves forward, the forward force vector remains unchanged. As seen in Figure 3.44 new vector fb reduces the angle and magnitude of the vector Fc which is now Fc′. Reduction of magnitude Fc′ shows that the greater forward force can be applied by increasing the driving torque before the roller breaks friction with ground. 74 This is the reason why Mecanum Wheel can have more traction and that is why roller bearings are used which improves the traction of the Mecanum Wheel. As shown in Figure 3.44 Fs magnitude is also reduced to Fs′ which is due to the introduction of the reduced. To obtain this, the motor torque should be increased to obtain the same sideways driving force. Increase in sideway force by increasing the motor force also gradually increases the reaction force Fc″ which is shown in Figure 3.45 below. Hence the roller bearing friction is the main reason behind the reduction of friction and available force in sideways direction [11].

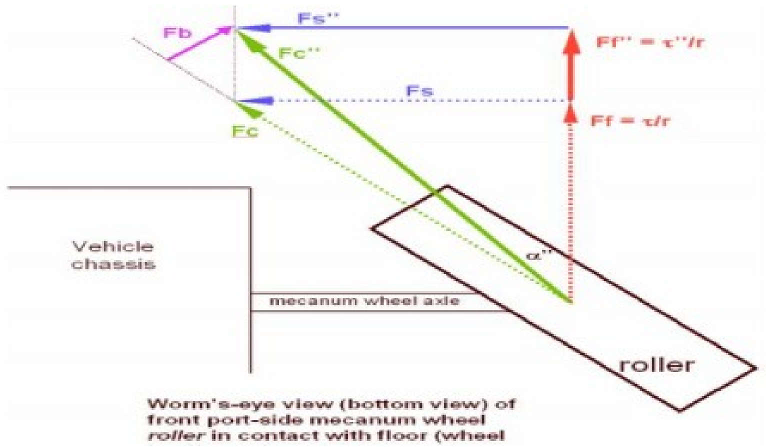



**Fig 3.46**: Increase Noticed in Motor Torque

The above case shows that Mecanum Wheel has low traction and it is due to the roller bearing. Also, it has been proven that more weight on the tire generates more traction.

. CHAPTER IV

# ARDUINO AND RASPBERRY Pi

## 4.1 Arduino

### 4.1.1 Introduction to Arduino

**Arduino** is open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. Its products are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards (*shields*) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

The name *Arduino* comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduino of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

### 4.1.2 Arduino Architecture: Harvard Architecture [10].

The processor of the Arduino board uses the **Harvard architecture**

The Harvard architecture is a term for a computer system that contains two separate areas for commands or instructions and data. In the Harvard architecture, the media, format and nature of the two different parts of the system may be different, as the two systems are represented by two separate structures.

Some examples of Harvard architectures involve early computer systems where programming input could be in one media, for example, punch cards, and stored data could be in another media, for example, on tape. More modern computers may have modern CPU processes for both systems, but separate them in a hardware design.

The Harvard architecture, with its strict separation of code and data processes, can be contrasted with a modified Harvard architecture, which may combine some features of code and data systems while preserving separation in others. One example is the use of two caches, with one common address space.

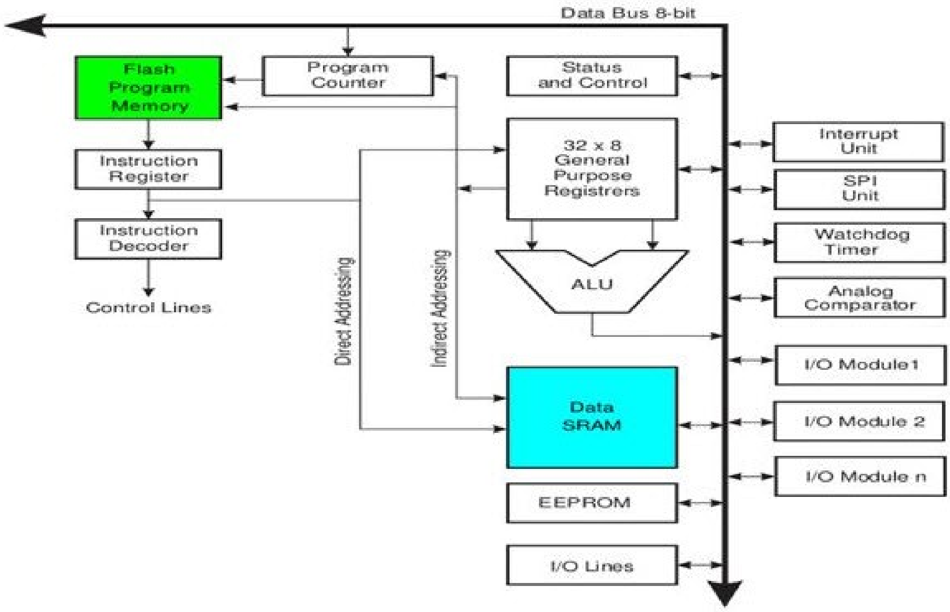



**Fig 4.1:** Architecture of Arduino ATmega328P microcontroller [10]

In Arduino the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Wherein the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed.

### 4.1.3Arduino Uno (R3)

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

**Fig 4.2:** Arduino Uno

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter

### 4.1.4 Tmega328P microcontroller

The **ATmega328** is a single-chip microcontroller created by Atmel in the megaAVR family. It has a modified Harvard architecture 8-bit RISC processor core. The Atmel AVR core combines a rich instruction set with 32 general purpose working registers

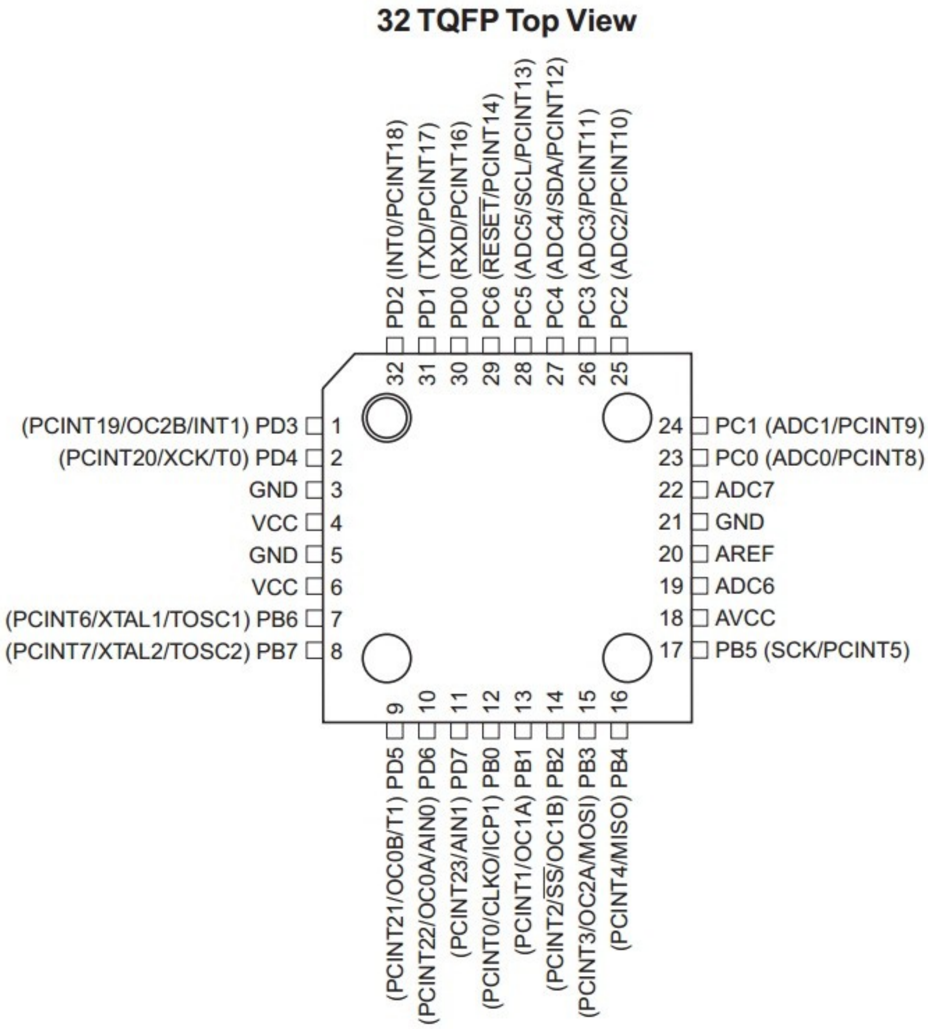



**Fig. 4.3:** Pinout diagram of ATmega328

. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle.

The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

### 4.1.5 Pin Description of Arduino

The pin configuration of the Arduino Uno board is shown in the above. It consists of 14-digital i/o pins.

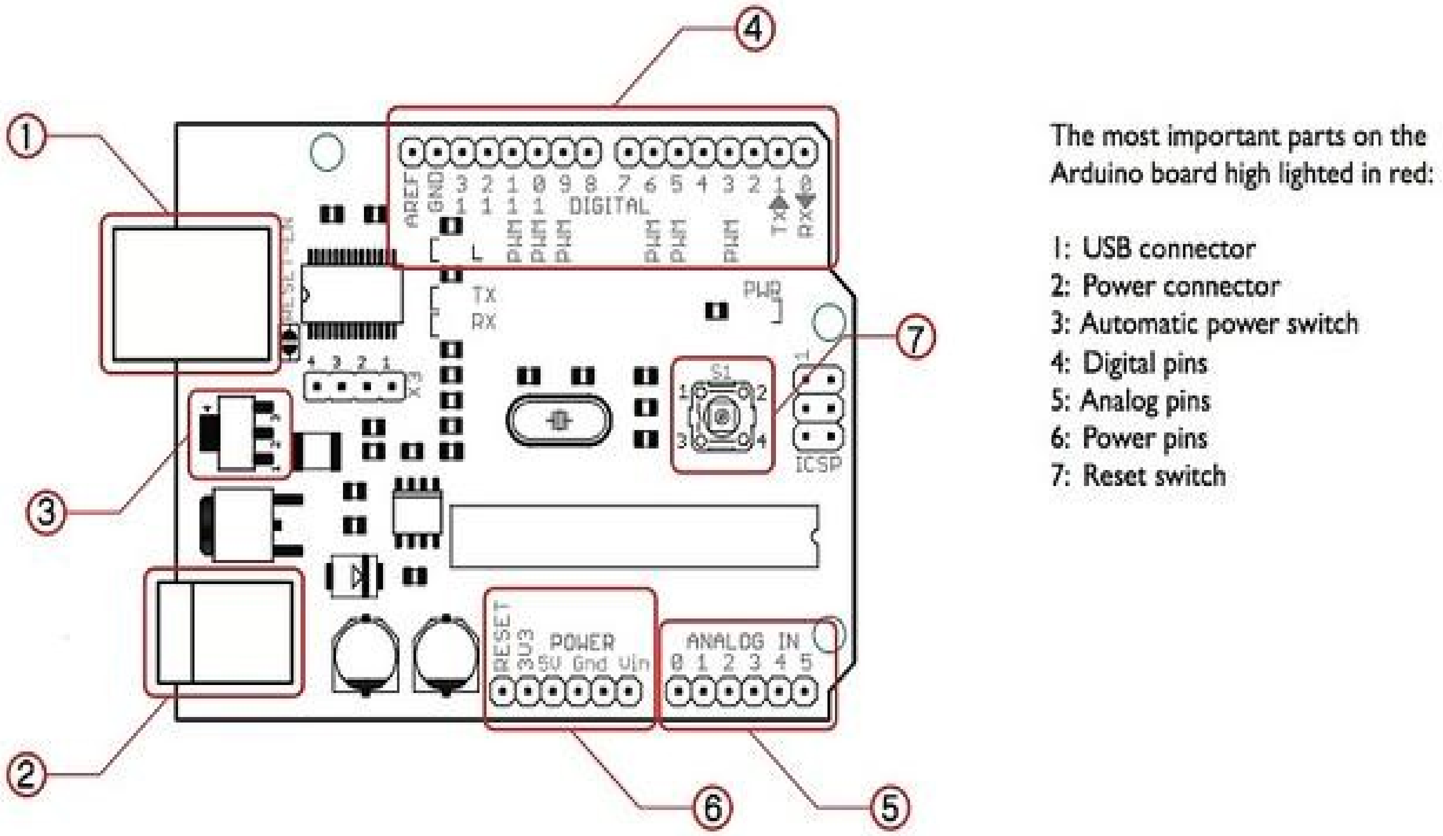



**Fig 4.4:** Pin Description of Arduino

Wherein 6 pins are used as pulse width modulation o/ps and 6 analog i/ps, a USB connection, a power jack, a 16MHz crystal oscillator, a reset button, and an ICSP header. Arduino board can be powered either from the personal computer through a USB or external source like a battery or an adaptor. This board can operate with an external supply of 7-12V by giving voltage reference through the IORef pin or through the pin Vin.

### 4.1.6 Input and Output Pins

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead()functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

In addition, some pins have specialized functions:

a) **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
b) **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
c) **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
d) **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
e) **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

**TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

a) **AREF.** Reference voltage for the analog inputs. Used with analogReference().
b) **Reset.** Bring this line LOW to reset the microcontroller. Typically used to
c) add a reset button to shields which block the one on the board

### 4.1.7 Advantages of Arduino Technology

a) It is cheap
b) It comes with an open supply hardware feature that permits users to develop their own kit

c) The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.

d) It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programing language libraries and may be extended and changed.

e) For beginners, it is very simple to use.

## 4.2 RASPBERRY PI

### 4.2.1 Introduction to Raspberry Pi

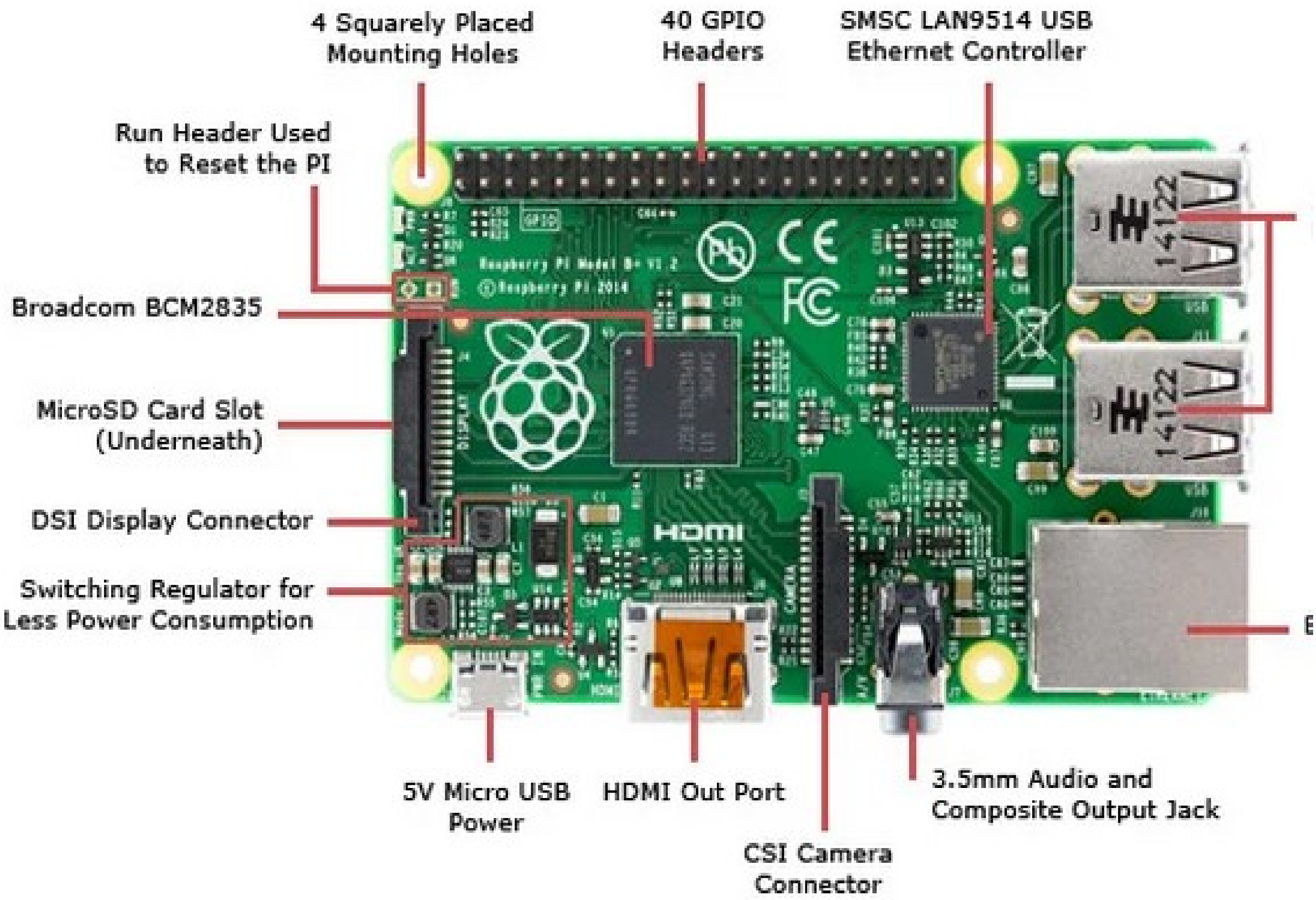
A Raspberry Pi is a credit card-sized computer originally designed for education, inspired by the 1981 BBC Micro. Creator Eben Upton's goal was to create a low-cost device that would improve programming skills and hardware understanding at the pre-university level. But thanks to its small size and accessible price, it was quickly adopted by tinkerers, makers, and electronics enthusiasts for projects that require more than a basic microcontroller (such as Arduino devices).

The Raspberry Pi is slower than a modern laptop or desktop but is still a complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level.

The Raspberry Pi is open hardware, with the exception of the primary chip on the Raspberry Pi, the BroadcommSoC (System on a Chip), which runs many of the main components of the board–CPU, graphics, memory, the USB controller, etc. Many of the projects made with a Raspberry Pi are open and well-documented as well and are things you can build and modify yourself.

There are a two Raspberry Pi models, the A and the B, named after the aforementioned BBC Micro, which was also released in a Model A and a Model B. The A comes with 256MB of RAM and one USB port. It is cheaper and uses less power than the B. The current model B comes with a second USB port, an ethernet port for connection to a network, and 512MB of RAM.

**Fig 4.5:** Raspberry Pi 3 Model B+




There were over 100 000 Raspberry Pi system boards sold on its first day of launch and almost 3 million units sold worldwide to date.Initially, Raspberry Pi boards were manufactured in China and Taiwan but today's models being sold are manufactured in the United Kingdom.The system board is overclockable by running the raspi-config tool or by editing the boot time parameters.Raspberry Pi does not come with a Mpeg-2 decoder but can be added for an increased price of about ten percent.

The system board runs on an ARM6 processor and therefore you can not run windows 8 or Ubuntu Linux as it requires ARM7 processing power. The newer Raspberry Pi 3 model runs on an ARM 8 processor.

The name Raspberry Pi originates from the fruit-based naming traditions in the old days for microcomputers.

### 4.2.2 FEATUES OF RASPBERRY PI

a) CPU: Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
b) GPU: 400MHz VideoCore IV multimedia
c) Memory: 1GB LPDDR2-900 SDRAM (i.e. 900MHz)
d) USB ports: 4
e) Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
f) Network: 10/100Mbps Ethernet and 802.11n Wireless LAN
g) Peripherals: 17 GPIO plus specific functions, and HAT ID bus
h) Bluetooth: 4.1
i) Power source: 5 V via MicroUSB or GPIO header
j) Size: 85.60mm × 56.5mm
k) Weight: 45g (1.6 oz)

### 4.2.3 INPUT AND OUTPUT PORTS

**Voltages**

Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are not configurable. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant.

**Outputs**

A GPIO pin designated as an output pin can be set to high (3V3) or low (0V).

**Inputs**

A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors.

Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software.

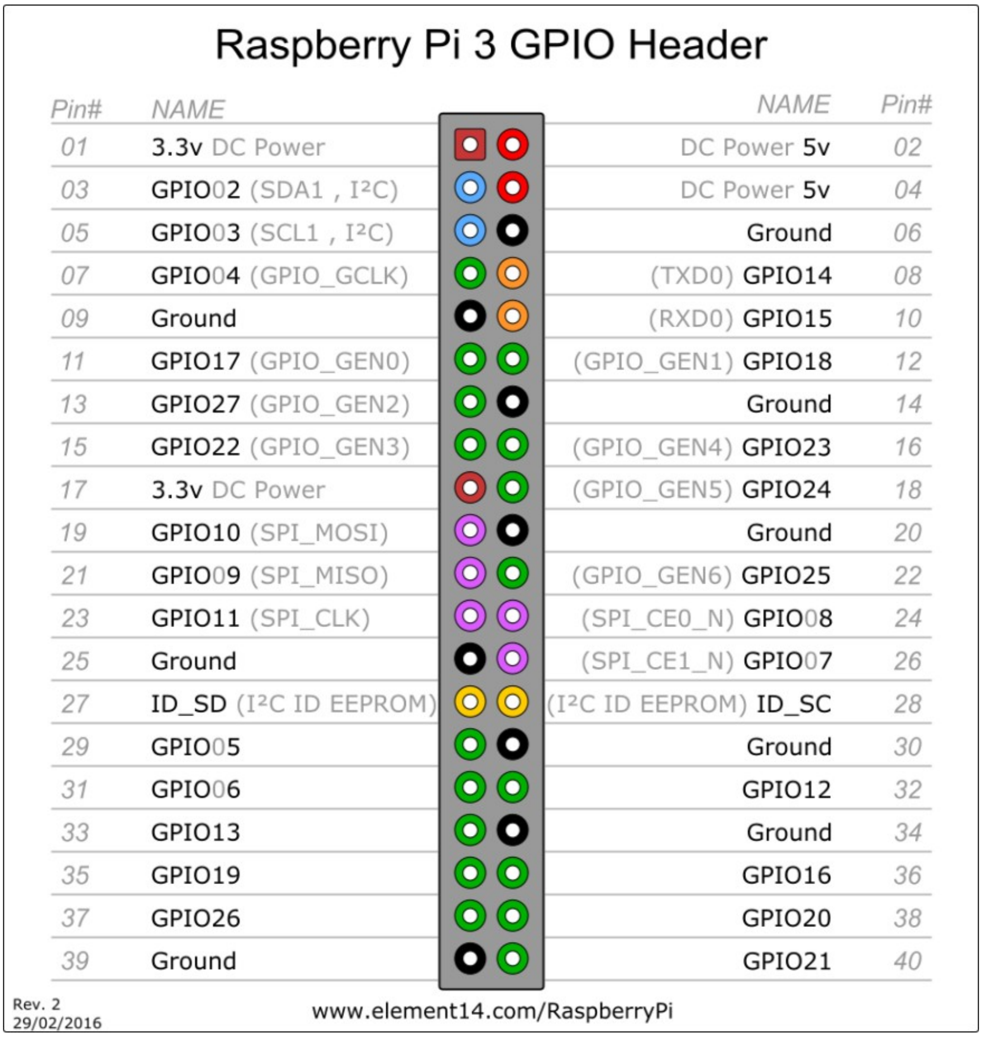



**Fig 4.6:** Raspberry pi Pin-out Diagram

As well as simple input and output devices, the GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on specific pins.

**PWM (pulse-width modulation)**

Software PWM available on all pins

Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19

**SPI**

**SPI0**: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)

**SPI1**: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

**I2C**

**Data: (GPIO2); Clock (GPIO3)**

**EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)**

**Serial**

**TX (GPIO14); RX (GPIO15)**

## 4.3 Interfacing Arduino and Raspberry Pi

One of the main aim of this project is to Program the bot in such a way that it much autonomously be able to go around the room and simultaneously be able to pick up the solid waste and clean the dust. This can be done by interfacing the Arduino (which controls the bot) to the Raspberry Pi (which does the image processing).

There are four basic ways to connect Arduino to Raspberry Pi:

1. Buy an add-on board like the Gertboard which has an Arduino compatible IC on it. Pricey.
2. Plug a standard Arduino like an Uno or Nano into the USB port of the RPi. This is by far the easiest method and minimises wiring and hassle. However it requires the more expensive Arduinos.
3. Use a USB to Serial adapter with a cheaper/smaller Arduino like a Pro Mini or a self-made Shrimp. This is the best DIY option and has the same advantage of method 2 that you can power the Arduino/Shrimp from USB. For a Model B RPi, I’d recommend this route.
4. Use the Serial Pins on the Raspberry Pi to connect to a cheaper/smaller Aruduino like a Pro Mini or a self-made Shrimp. This is theoretically the cheapest method but by far the most hassle. This is also the best method if you are using the cheaper Raspberry Pi Model A and its single USB port is being used for Wi-Fi.

The easiest and most efficient way of interfacing both the arduino and Pi is done by using the USB RPi and connecting it to an Arduino through a Standard USB cable, which can be done by following these steps:

a) First step is to connect the 3.3V/GND/TX/RX pins on the Raspberry Pi via a level converter to 5V/GND/RX/TX pins on an Arduino. Alternatively you buy a 3.3V Arduino and avoid the need for a level converter.

b) RPi software changes involve commenting out this line in /etc/inittab with a #

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

and removing the following parts of the one line in /boot/cmdline.txt

```
console=ttyAMA0,115200 kgdboc=ttyAMA0,115200
```

You also need to create a link to the serial port so that the Arduino IDE recognises it:

```
sudoln -s /dev/ttyAMA0 /dev/ttyUSB9
```

That last step has to be done after every reboot.

c) Connect Pin 11 (GPIO 17) of the RPi to the DTR Pin on the Arduino Pro Mini via the level converter

d) Run the following commands to download and configure avrdude-rpi:

```
sudo apt-get update

sudo apt-get install python-dev

sudo apt-get install python-rpi.gpio
```

```
wgethttps://raw.github.com/deanmao/avrdude-rpi/master/autoreset

wgethttps://raw.github.com/deanmao/avrdude-rpi/master/avrdude-autoreset

sudocpautoreset /usr/bin

sudocpavrdude-autoreset /usr/bin

sudo mv /usr/bin/avrdude /usr/bin/avrdude-original

sudoln -s /usr/bin/avrdude-autoreset /usr/bin/avrdude

sudochmod 755 /usr/bin/avrdude-autoreset

sudochmod 755 /usr/bin/autoreset
```

# CHAPTER V

## Steps Involved in interfacing the remote control to Arduino

The main method of controlling the bot is through a remote controller. The remote controller used in this project is a Genuine Wireless Dualskock 3 PS3 controller which uses Bluetooth to wirelessly communicate with the wireless Bluetooth dongle connected to the arduino. Since, the Arduino board used in the project is an Arduino Uno which does not have a built-in USB port we need to use a USB Host shield in order to use the Bluetooth dongle with the Arduino

### Step 1

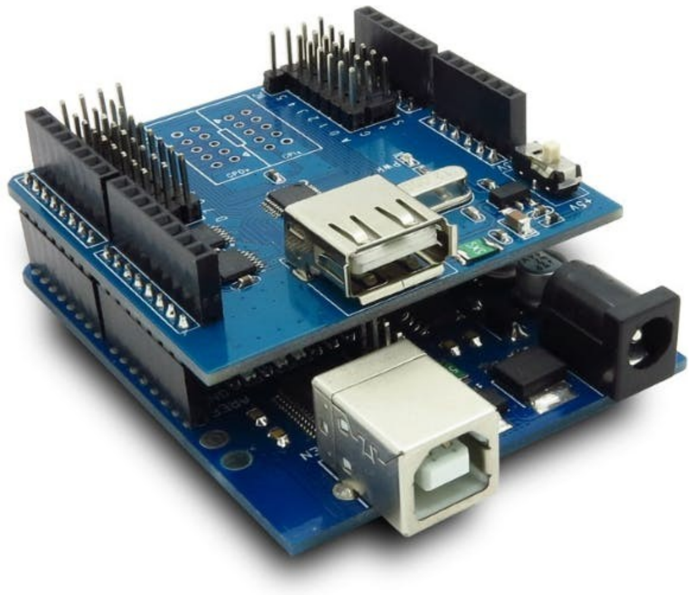


**Fig 5.1**: Arduino With USB Host Shield

Put the USB Shield on the top of the Arduino.

### Step 2

Download the USB Host shield libraries

### Step 3

Upload the code to your Arduino.

```
#include<SPI.h>
#include<PS3BT.h>                                   //Include the necessary libraries.

#include<Servo.h>

USB Usb;
```

```
BTD Btd(&Usb);

PS3BT PS3(&Btd);



Servo servo1;                                    //Create instances of type
Servo.servo1 is the steering servo and servo2 is the ESC.

Servo servo2;



voidsetup() {

Serial.begin(115200);



if (Usb.Init() == -1) {

Serial.print(F("\r\nOSC did not start"));

while (1); //halt

  }

Serial.print(F("\r\nPS3 Bluetooth Library Started"));



servo1.attach(9);                                //Steering servo on digital pin 5

}

voidloop()

{

Usb.Task();



if (PS3.PS3Connected || PS3.PS3NavigationConnected) {



servo1.write(map(PS3.getAnalogHat(RightHatX), 0, 255, 0, 180));
```

```
servo2.write(map(PS3.getAnalogHat(LeftHatY), 0, 255, 180, 0));

Serial.println(map(PS3.getAnalogHat(RightHatX), 0, 255, 0, 180));

  }

if (PS3.getButtonClick(UP)) {

digitalWrite(Led, HIGH);


Serial.print(F("\r\nUp"));


if (PS3.PS3Connected) {

PS3.setLedOff();

PS3.setLedOn(LED1);


    }

  }

}
```

## Step 4

Plug the Bluetooth dongle in the USB shield.

## Step 5

Open the serial and wait until the Arduino will read the Bluetooth dongle then unplug it and connect your PS3 control to the USB shield and wait until the Arduino will change the mac address of the PS3 controller then unplug it an plug the Bluetooth dongle again and press the center button in the PS3 controller (HOME BUTTON(power on)) and the PS3 will be connected by seeing the number 1 on the controller is on.

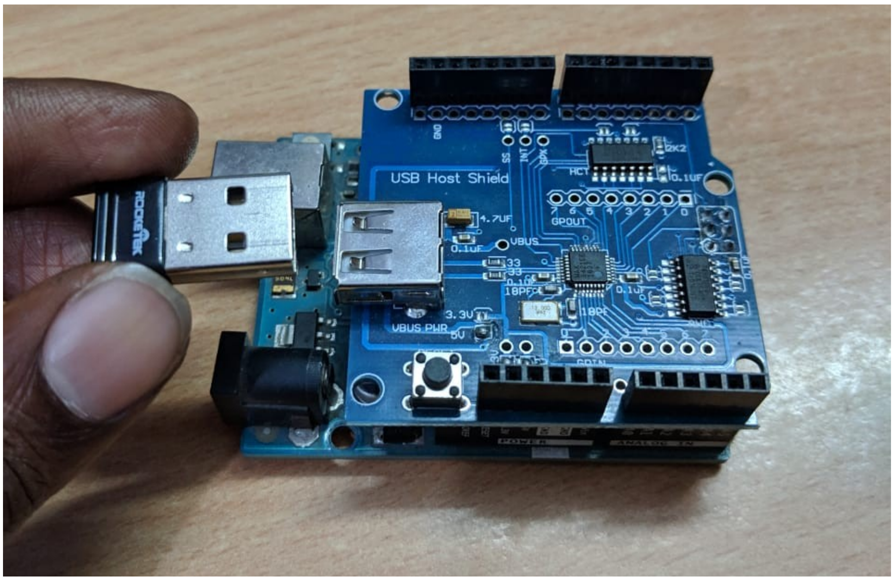



**Fig 5.2:** USB Dongle with the USB Host Shield

And on the serial it will tell you that the PS3 controller is connected and ready to use. Then the angle of the servo will be showing in the serial.

After completion of the above steps we can check whether the connection has been established or not by checking the serial monitor.

# Chapter VI

# Software Implementation

## 6.1 ARDUINO IDE

The **Arduino integrated development environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.**Raspbian** is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie.Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

The Arduino board is connected to a computer via USB, where it connects with the Arduino development environment (IDE). The user writes the Arduino code in the IDE, then uploads it to the microcontroller which executes the code, interacting with inputs and outputs such as sensors, motors, and lights.

### 6.1.1About the Software

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

### 6.1.2Arduino Coding Environment and basic tools

**What language is Arduino?**

Arduino code is written in C++ with an addition of special methods and functions, which we’ll mention later on. C++ is a human-readable programming language. When you create a ‘sketch’ (the name given to Arduino code files), it is processed and compiled to machine language.

**Sketches**

The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. It is where you’ll be typing up your code before uploading it to the board you want to program. Arduino code is referred to as **sketches**.

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers).

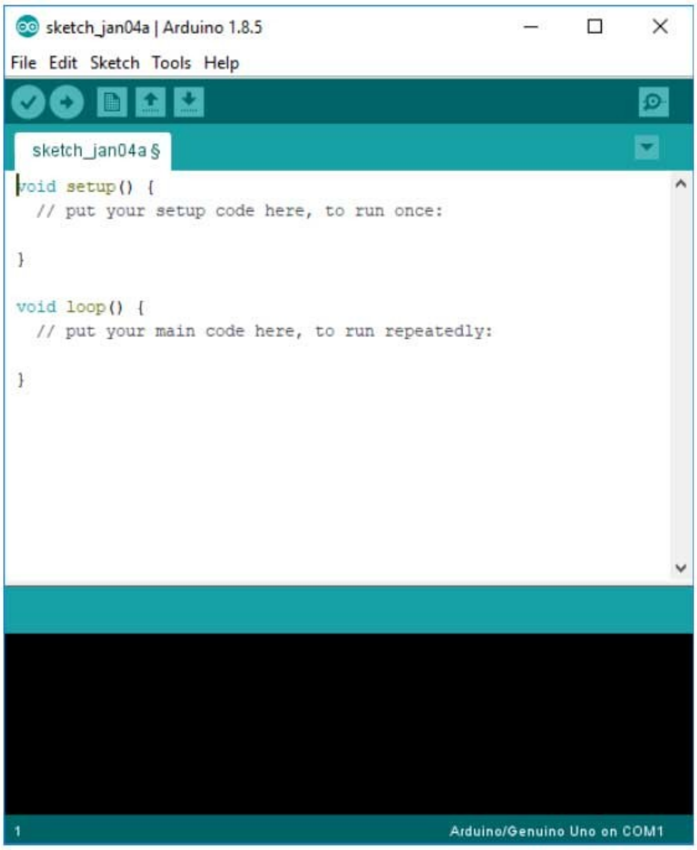



**Fig 6.1:** Arduino in its default state

This sometimes confuses users who think Arduino is programmed in an “Arduino language.” However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.

As you can see, the IDE has a minimalist design. There are only 5 headings on the menu bar, as well as a series of buttons underneath which allow you to verify and upload your sketches. Essentially, the IDE translates and compiles your sketches into code that Arduino can understand. Once your Arduino code is compiled it’s then uploaded to the board’s memory.

All the user has to do to start compiling their sketch is press a button (a guide to this can be found below).


**Fig 6.2:** Arduino code example

If there are any errors in the Arduino codea warning message will flag up prompting the user to make changes. Most new users often experience difficulty with compiling because of Arduino’s stringent syntax requirements. If you make any mistakes in your punctuation when using Arduino, the code won’t compile and you’ll be met with an error message.

### Serial Monitor and Serial Plotter

**Arduino serial monitor** can be opened by clicking on the magnifying glass icon on the upper right side of the IDE or under tools. The serial monitor is used mainly for interacting with the Arduino board using the computer, and is a great tool for real-time monitoring and debugging. In order to use the monitor, you’ll need to use the Serial class.

The code you download from circuito.io has a test section that helps you test each components using the serial monitor, as you can see in the screenshot below:

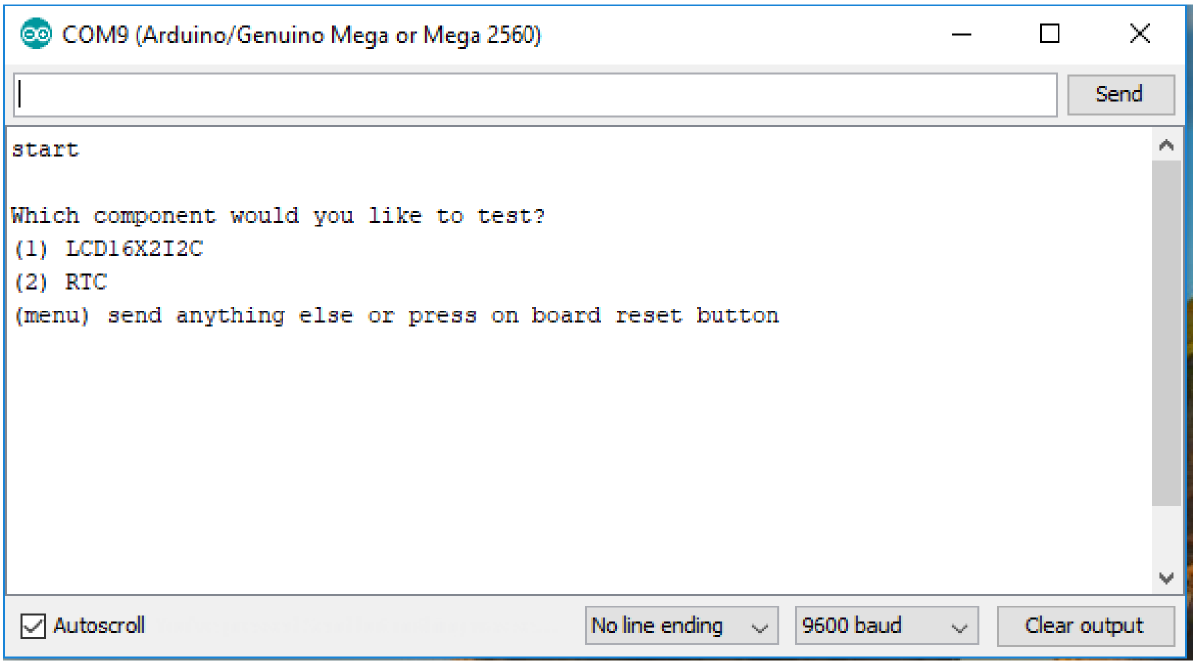



**Fig 6.3**: Arduino Serial Monitor

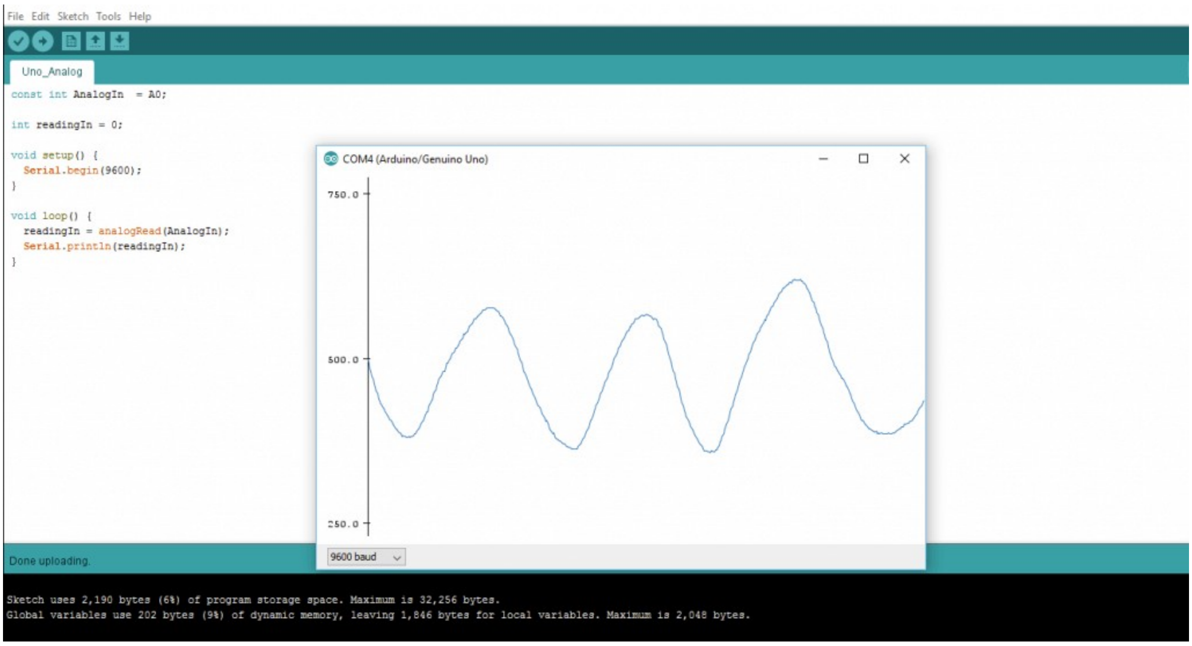



**Fig 6.4:** Arduino Serial Plotter

**Arduino serial plotter** is another component of the Arduino IDE, which allows you to generate a real-time graph of your serial data. The serial plotter makes it much easier for you to analyze your data through a visual display. You're able to create graphs, negative value graphs, and conduct waveform analysis.

**Debugging Arduino Code and Hardware**

Unlike other software programming platforms, Arduino doesn't have an onboard debugger. Users can either use third-party software, or they can utilize the serial monitor to print Arduino's active processes for monitoring and debugging.

By using the Serial class, you can print to the serial monitor, debugging comments and values of variables. On most Arduino models, this will be using serial pins 0 and 1 which are connected to the USB port.

**The 6 Buttons**

While more advanced projects will take advantage of the built-in tools in the IDE, most projects will rely on the six buttons found below the menu bar.


**Fig 6.5**: The button bar

1. The **check mark** is used to verify your code. Click this once you have written your code.
2. The **arrow** uploads your code to the Arduino to run.
3. The **dotted paper** will create a new file.
4. The **upward arrow** is used to open an existing Arduino project.
5. The **downward arrow** is used to save the current file.
6. The far right button is a **serial monitor**, which is useful for sending data from the Arduino to the PC for debugging purposes.

## 6.1.3 Conclusion

There are plenty of other features available to consider on the IDE. But, having used many different types of microcontrollers and having been involved in multiple programming environments, it is shocking how simple the Arduino and its IDE is! In less than two minutes, you can get a simple C++ program uploaded onto the Arduino and have it running.

## 6.2 RASPBIAN

### 6.2.1About the Software

The Raspberry Pi operates in the open source ecosystem: it runs Linux (a variety of distributions), and its main supported operating system, Raspbian, is open source and runs a suite of open source software. The Raspberry Pi Foundation contributes to the Linux kernel and various other open source projects as well as releasing much of its own software as open source.

### 6.2.2 Advantages

The Raspberry Pi is tiny computer with a 700 MHz ARM11 processor and 256MB of RAM. While it won't run Windows, it can run a variety of Linux-based operating systems including Debian, Fedora, and even Raspbmc

But the Raspberry Pi's ARM-based processor isn't exactly a speed demon, and while those operating systems all *run* on the computer, they don't necessarily run very quickly.
Raspbian, on the other hand, is an upcoming operating system that looks like it could run circles around the others.
The operating system is based on Debian Linux, but it includes support for "hard float" code that takes advantage of the Raspberry Pi's hardware. That means software that use floating point operations should run much more quickly on Raspbian than they do on existing versions of Debian for the Raspberry Pi.

## 6.3 CAD Software

**Computer-aided design** (**CAD**) is the use of computers (or workstations) to aid in the creation, modification, analysis, or optimization of a design. CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing.CAD output is often in the form of electronic files for print, machining, or other manufacturing operations.

We designed the bot in SolidWorks and structural analysis was done in ANSYS.

### 6.3.1 SolidWorks

SolidWorks is a solid modeling computer-aided design and computer-aided engineering computer program that runs on Microsoft Windows. SolidWorks is published by DassaultSystèmes. According to the publisher, over two million engineers and designers at more than 165,000 companies were using SolidWorks as of 2013.

Advantage:

1. User friendly (Easy to learn)
2. Integrated to different other software's like SOLID CAM, ANSYS. (Direct import and export of the file can be done easily)
3. Complex motion analysis can be done.
4. Realistic rendering can be obtained but not that good as Catia.
5. Complex surfaces can be Designed and their patterns can be viewed.
6. A proper estimation of Cost report and durability of the given product can be obtained.
7. Weldments is an additional added feature then other software's

Disadvantage:

1. Few modules like Ergonomics is still missing.
2. Drawing and drafting particular product is still to basic level.

## 6.4 FINITE ELEMENT ANALYSIS

After producing the prototype, it undergoes certain analysis to check for its certain properties. It can be applied to many different areas. For example, it can be applied to different structures such as beam, bridge, etc. It can also be applied to mechanics such as gear, powertrain, etc. Finite element analysis is carried to check displacement, stresses and strain at each material point when force is applied. Various functions such as mass, volume, temperature, stress, strain, displacement etc. can be 16 measured by using FEA software. Various types of analysis can be done with the help of FEA software such as, structural, vibrational, fatigue and heat transfer analysis.

### 6.4.1 ANSYS

**Ansys Inc.** is an American public company based in Canonsburg, Pennsylvania. It develops and markets engineering simulation software. Ansys software is used to design products and semiconductors, as well as to create simulations that test a product's durability, temperature distribution, fluid movements, and electromagnetic properties.

Ansys was founded in 1970 by John Swanson. Swanson sold his interest in the company to venture capitalists in 1993. Ansys went public on NASDAQ in 1996. In the 2000s, Ansysmade numerous acquisitions of other engineering design companies, acquiring additional technology for fluid dynamics, electronics design, and other physics analysis.

With ANSYS software, you can achieve:

a) Innovative, reliable and high-quality products and processes
b) Faster ROI, due to fewer physical prototypes and test setups
c) A more flexible and responsive information-based development process, enabling modifications of design at later stages
d) A seamless working exchange of data, regardless of location, industry, CAD environment, etc.

# Chapter 7

# Results

The robot is controlled manually using a PS3 Bluetooth controller that is interfaced with the Arduino Uno module through a UBB Bluetooth dongle.

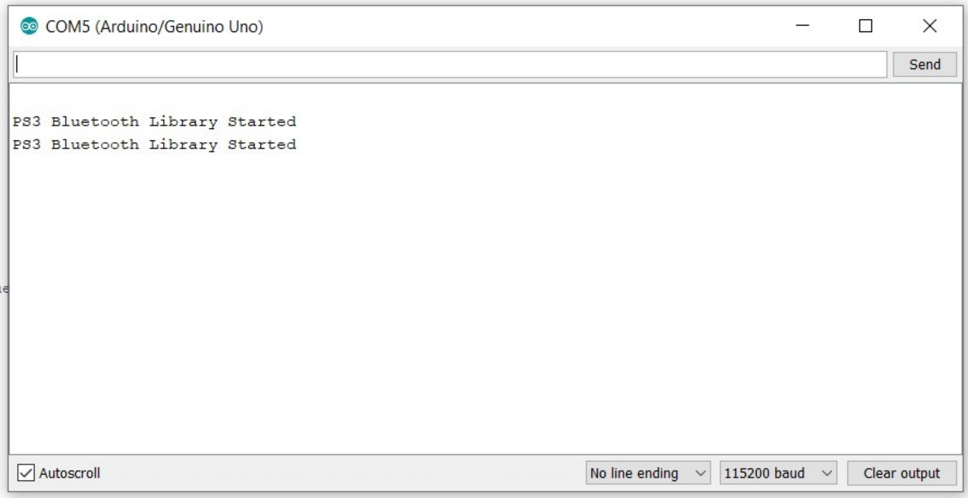



**Fig 7.1** dongle interface initiation

To initiate the address association, connect the Bluetooth host shield with the Arduino module. The MAC address of the USB dongle is first recognized, then the dongle is disconnected and the controller is connected to the Arduino using the USB cable. The controller is now assigned this MAC address and then the controller is disconnected, and the USB dongle is connected again, this causes the controller to be connected to the Arduino module.

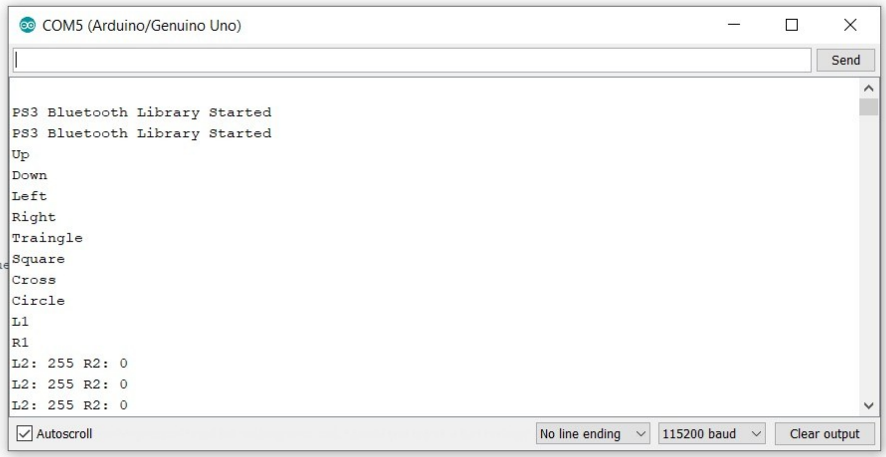



**Fig 7.2:** Button configuration

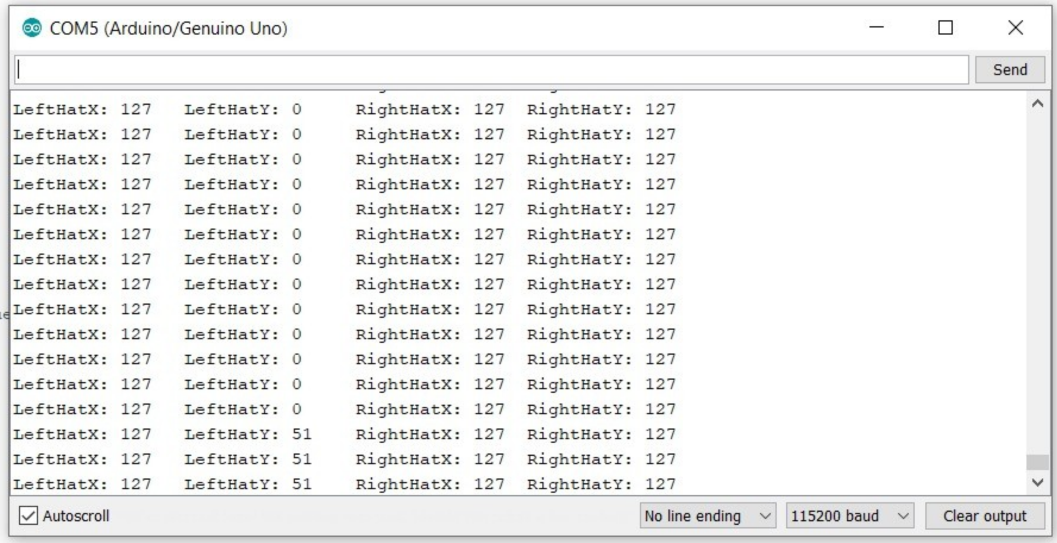



**Fig 7.3** the interface of the Bluetooth controller

The toggle switches used in the control of the robot is read by the Arduino using the above notations. The mecanum wheels are successfully controlled to move the robot in all directions, with the help of the mapping technique that is condensed in the tabular columns below.

**Table 7.1** Mapping of controller buttons

| **BOT** | **FRONT WHEELS** | | **REAR WHEELS** | | **L298N** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **MOVEMENT (DIRECTION)** | **LEFT (1,2)** | **RIGHT (3,4)** | **LEFT (1',2')** | **RIGHT (3',4')** | **FRONT** | | | | **REAR** | | | |
| | | | | | **1** | **2** | **3** | **4** | **1'** | **2'** | **3'** | **4'** |
| **FORWARD( )** | | | | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| **BACKWARD( )** | | | | | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| **RIGHT( )** | | | | | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| **LEFT( )** | | | | | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| **NORTHEAST( )** | | - | - | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| **NORTHWEST( )** | - | | | - | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **SOUTHEAST( )** | - | | | - | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **SOUTHWEST( )** | | - | - | | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **CLOCKWISE( ↻ )** | | | | | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| **ANTICLOCKWISE ( ↺ )** | | | | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

**Table 7.2**: Control logic of the robot

| Button name | Button description |
|---|---|
| Up arrow | Upward movement |
| Down arrow | Downward movement |
| Left arrow | Leftward movement |
| Right arrow | Rightward Movement |
| Triangle Button | Movement towards Northeast direction |
| Circle button | Movement towards Northeast direction |
| X Button | Movement towards Northeast direction |
| Square  button | Movement towards Northeast direction |
| L1 | Rotation in anti-clockwise direction |
| R1 | Rotation in clockwise direction |
| L2 | Servo motor anti-clockwise motion |
| R2 | Servo motor clockwise motion |

# ANSYS RESULTS:

Results:

Case-1: Load due to battery on its mount considering its weight.

a.) Stress on the application of load

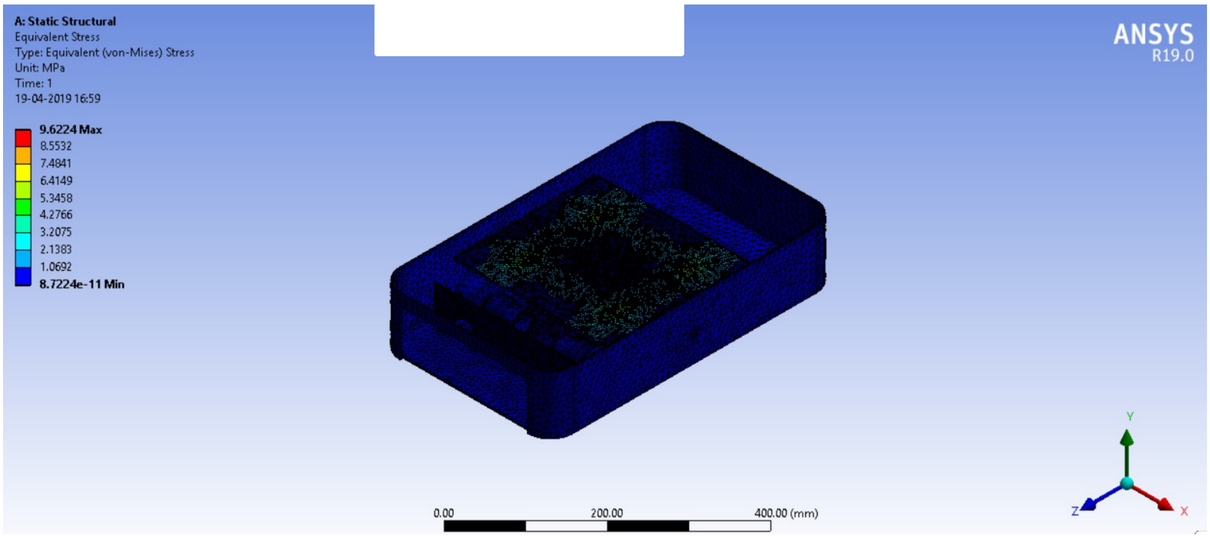



**Fig 7.4**

Maximum Stress occurred: **9.62MPa**

b.) Deformation on the application of load

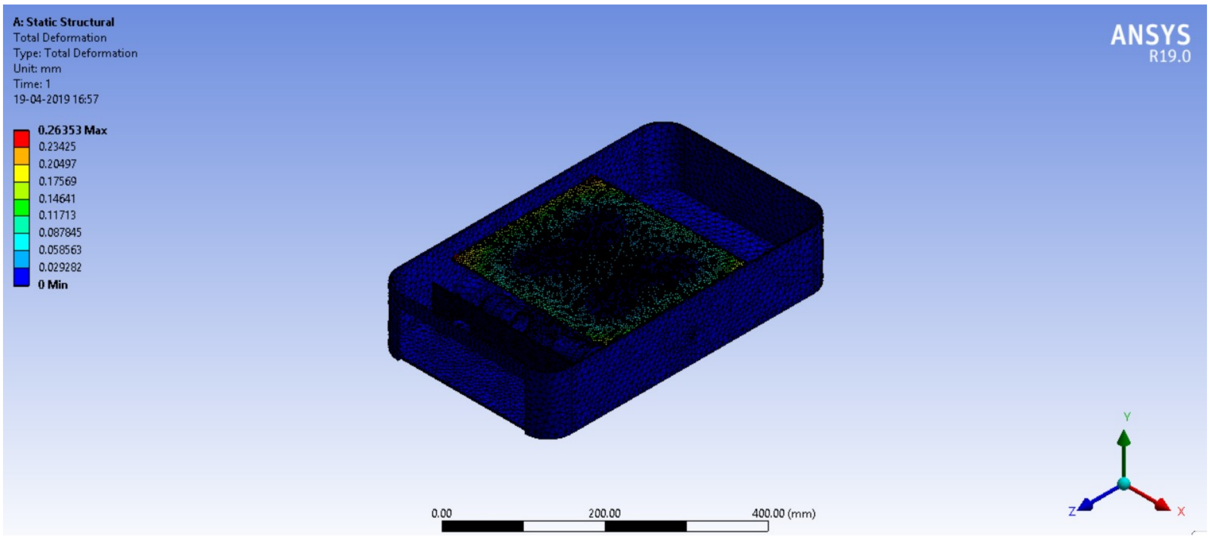



**Fig 7.5**

Maximum Deformation occurred: **0.263mm**

Case-2: Load considered during front impact

a.) Stress on application of load

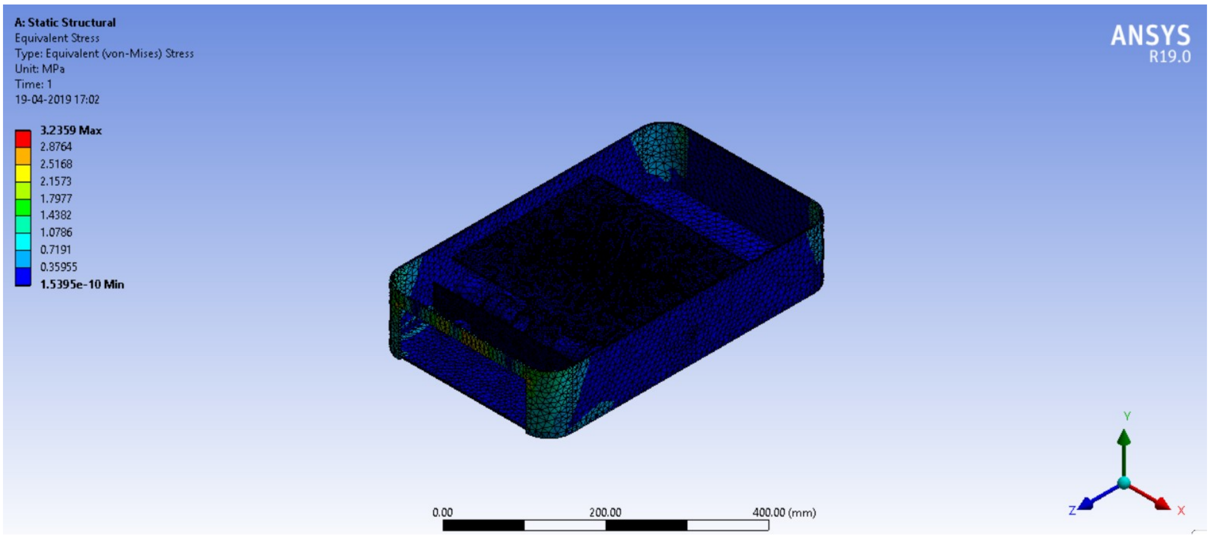



**Fig 7.6**

Maximum stress occurred: **3.23MPa**

b.) Deformation on application of load

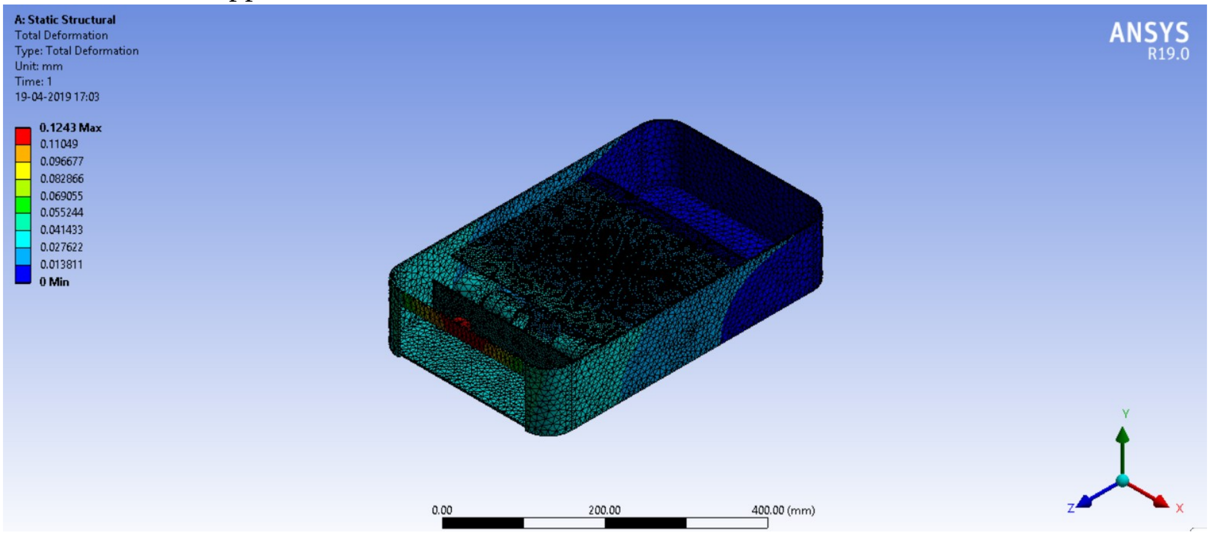



**Fig 7.7**

Maximum Deformation occurred: **0.1243mm**

# Chapter 8

# Conclusions and Future Scope

## 8.1 SCOPE FOR FUTURE WORK

After the initial implementation of our project, we can further develop it to make it more effective. Firstly, we have considered the inclusion of neural network based artificial intelligence into the garbage collector, to replace the present method we have of finding the path, which consists of IR sensors and a manual PS3 controller . This would help to overcome the difficulties we faced about human interaction with the robot and making it exclusively autonomous. Also, we have an idea to introduce our project to customizable cleaning mechanism such as wiping the floor, disposal of garbage autonomously and inculcating an automatic docking system when the battery of the robot gets completely discharged, instead of being limited to cleaning of dust using a vacuum, collecting and disposal of paper only. We can broaden the scope of the detectable specimen, which is presently only paper; to a variety of solid waste materials. As the robot is put to multiple testing phases, it can be more fast and equipped to collect and detect trash. As the robot is presently built for a specific area namely an indoor laboratory, it can further be expanded to the outdoors. Once the robot is exposed to a higher level of machines learning and image processing techniques, it can be further evolved to be used over a larger area, with more efficiency and quickness.

This would greatly expand thescope of use of our project. Finally, as a step towards improving the environment-friendly aspect of our project, we could arrange a way to separate different types of garbage into different compartments within the garbage collector. This can be done according to the 5S system, and would make the task of managing the garbage more easy and tidy5S is a systematic form of visual management utilizing everything from floor tape to operations manuals. It is not just about cleanliness or organization; it is also about maximizing efficiency and profit. 5S is a framework that emphasizes the use of a specific mindset and tools to create efficiency and value. It involves observing, analyzing, collaborating, and searching for waste and also involves the practice of removing waste. An introduction to a robotic arm can greatly influence the operability of the robot, as it is possible to attach a robotic arm which is customizable and thereby, can greatly improve its versatility in the cleaning processes. Collectively, these are the possible further implementations of our project.

We also have a blueprint of a specialized unique docking system for the bot with wireless charging designed into it for hands free charging and disposal of dust. This docking system is equipped with a set of electrical, electromechanical and electronic components working together efficiently.

## 8.2 Conclusion

In previous chapters, we have given a description about the operation of the robot. The robot is presently run manually using a Bluetooth PS3 controller device, so that the disposal of garbage can be automated to some extent thus reducing the labor requirement, health hazard and increasing the productivity, efficiency of the robot. The robot has a unique quality of maintaining the orientation of the body as the same, using special kind of wheels known as Mecanum wheels. These wheels are designed to move a vehicle in any direction. These rollers typically each have an axis of rotation at 45° to the plane of the wheel and at 45° to a line through the center of the roller parallel to the axis of rotation of the wheel. The primary aim for doing our project was to introduce a way in which solid garbage could be collected and disposed in an efficient and effective manner. By implementing the automated garbage collector, we were able to achieve our targets largely. By developing this system mainly focused in an indoor laboratory. In addition, the cleanliness of the surroundings is maintained with less effort. The become more conscious about disposing garbage in the right way, at the right time, by getting used to this automated system. Therefore, both effectiveness and efficiency garbage disposal is improved. The rechargeable battery is custom-tailored made using a special technique called spot welding, which is made using a confutation of 4S4P network; It is a combination of individual rechargeable cells named, ICR18650 – 26 F, individually holding 2600mah of energy. In total, it keeps the power to 10,400 mAH. The cells have are Lithium Ion rechargeable batteries a nominal voltage of 3.7 V each, and a maximum discharge current of 5.2 A.

# REFERENCES


[1] ShobhitKhandare Et.al, “Object Detection Based Garbage Collection Robot (E-Swachh)”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03 | Mar-2018.

[2] Hsiao Shih-Wen and Yeh, Ting An, “Application of Collaborative Design Strategy on Redesign of the Cordless Household Vacuum Cleaner”, International Conference on Organizational Innovation (ICOI 2017), Vol. 131, pp. 211-222, 2017.

[3] Park Changhwan, Jun Sangook, Park Kyunghyun, Lee Sangjong and Chang Kyoungsik , “Methodology for SystemLevel Analysis of a Fan-Motor Design for a Vacuum Cleaner”, Part C-Journal of Mechanical Engineering Science, Vol. 231(20) pp. 3840-3854, October 2017.

[4] An article named Effects of Vacuum Cleaner Brushing Action, in the website “ristenbatt.com” entails the importance of brushes in the design of the project.

[5] The article named –vacuum cleaner guide; in the website “learn.compactappliance.com” explains the choice of vacuum cleaner, according to the project specification.

[6] Shih-Liang (Sid) Wang “Motion Control and the Skidding of Mecanum Wheel Vehicles”, Department of Mechanical Engineering, North Carolina A&T State University, Greensboro, North Carolina, USA.

[7] From the Image posted by the user wokz from the Arduino forums website, “forum.arduino.cc/index.php?topic=332159.0”.

[8] MihaiOlimpiu Tătar1,a, Claudiu Cirebea1,b, Dan Mândru1,c 1Technical University of Cluj-Napoca, “Structures of the Omnidirectional Robots with Swedish Wheels” Department of Mechatronics and Machine Dynamics, Muncii Blvd. 103-105, Cluj-Napoca, România.

[9] An article named Buck (step down) converter/regulator, in the website “onmyphd.com” entails the workings of a step down buck convertor.

[10] An article named Arduino (ATmega328/p) Architecture Explained, in the website “engineering.eckovation.com” explains the architecture of the arduinouno board.

[11] Kang, Shao Hua, Xue Ling Zhang, and Ya Xie. 'Mechanics Analysis of OmniDirectional Mecanum-Wheel Based On FEM'. AMM 271-272 (2012): 1012-1016.

# APPENDIX

## SOFTWARE CODE FOR ARDUINO

```
#include<PS3BT.h>
#include<usbhub.h>

// Satisfy the IDE, which needs to see the include statment in the ino too.
#ifdefdobogusinclude
#include<spi4teensy3.h>
#endif
#include<SPI.h>

USBUsb;
//USBHubHub1(&Usb);
BTDBtd(&Usb);
PS3BTPS3(&Btd); // This will just create the instance
//PS3BT PS3(&Btd, 0x00, 0x15, 0x83, 0x3D, 0x0A, 0x57); // This will also
store the bluetooth address - this can be obtained from the dongle when
running the sketch

boolprintTemperature,printAngle;

//Motordriver1
constint M11 = 0;
constint M12 = 1;
constint M13 = 2;
constint M14 = 3;

//Motordriver2
constint M21 = 4;
constint M22 = 5;
constint M23 = 6;
constint M24 = 7;


voidsetup() {
  Serial.begin(115200);
#if!defined(__MIPSEL__)
  while (!Serial); // Wait for serial port to connect - used on Leonardo,
Teensy and other boards with built-in USB CDC serial connection
#endif
  if (Usb.Init() ==-1) {
    Serial.print(F("\r\nOSC did not start"));
    while (1); //halt
  }
  Serial.print(F("\r\nPS3 Bluetooth Library Started"));

pinMode(M11,OUTPUT);
pinMode(M12,OUTPUT);
pinMode(M13,OUTPUT);
pinMode(M14,OUTPUT);

pinMode(M21,OUTPUT);
pinMode(M22,OUTPUT);
pinMode(M23,OUTPUT);
pinMode(M24,OUTPUT);
```

```
}
voidloop() {
  Usb.Task();

  if (PS3.PS3Connected|| PS3.PS3NavigationConnected) {
    if (PS3.getAnalogHat(LeftHatX) > 137 || PS3.getAnalogHat(LeftHatX) <
117 || PS3.getAnalogHat(LeftHatY) > 137 || PS3.getAnalogHat(LeftHatY) < 117
|| PS3.getAnalogHat(RightHatX) > 137 || PS3.getAnalogHat(RightHatX) < 117
|| PS3.getAnalogHat(RightHatY) > 137 || PS3.getAnalogHat(RightHatY) < 117)

    {
      Serial.print(F("\r\nLeftHatX: "));
      Serial.print(PS3.getAnalogHat(LeftHatX));
      Serial.print(F("\tLeftHatY: "));
      Serial.print(PS3.getAnalogHat(LeftHatY));
      if (PS3.PS3Connected) { // The Navigation controller only have one
joystick
        Serial.print(F("\tRightHatX: "));
        Serial.print(PS3.getAnalogHat(RightHatX));
        Serial.print(F("\tRightHatY: "));
        Serial.print(PS3.getAnalogHat(RightHatY));
      }
    }

    // Analog button values can be read from almost all buttons
    if (PS3.getAnalogButton(L2) || PS3.getAnalogButton(R2)) {
      Serial.print(F("\r\nL2: "));
      Serial.print(PS3.getAnalogButton(L2));
      if (PS3.PS3Connected) {
        Serial.print(F("\tR2: "));
        Serial.print(PS3.getAnalogButton(R2));
      }
    }

    if (PS3.getButtonClick(PS))
    {
      Serial.print(F("\r\nPS"));
      PS3.disconnect();
    }
    else {
      if (PS3.getButtonClick(TRIANGLE))
      {
        Serial.print(F("\r\nTraingle"));

// Right Forward

digitalWrite(M11,LOW);
digitalWrite(M12,HIGH);
digitalWrite(M13,LOW);
digitalWrite(M14,LOW);

digitalWrite(M21,LOW);
digitalWrite(M22,LOW);
digitalWrite(M23,LOW);
digitalWrite(M24,HIGH);

      }
      if (PS3.getButtonClick(CIRCLE))
      {
        Serial.print(F("\r\nCircle"));
```

```
      //left forward

digitalWrite(M11,LOW);
digitalWrite(M12,LOW);
digitalWrite(M13,LOW);
digitalWrite(M14,HIGH);

digitalWrite(M21,LOW);
digitalWrite(M22,HIGH);
digitalWrite(M23,LOW);
digitalWrite(M24,LOW);


      }
      if (PS3.getButtonClick(CROSS))
      {
        Serial.print(F("\r\nCross"));

        //back left

digitalWrite(M11,HIGH);
digitalWrite(M12,LOW);
digitalWrite(M13,LOW);
digitalWrite(M14,LOW);

digitalWrite(M21,LOW);
digitalWrite(M22,LOW);
digitalWrite(M23,HIGH);
digitalWrite(M24,LOW);

      }
      if (PS3.getButtonClick(SQUARE))
      {
        Serial.print(F("\r\nSquare"));

        //back right

digitalWrite(M11,LOW);
digitalWrite(M12,LOW);
digitalWrite(M13,HIGH);
digitalWrite(M14,LOW);

digitalWrite(M21,HIGH);
digitalWrite(M22,LOW);
digitalWrite(M23,LOW);
digitalWrite(M24,LOW);

      }
      if (PS3.getButtonClick(UP))
      {
        Serial.print(F("\r\nUp"));
        if (PS3.PS3Connected) {
          PS3.setLedOff();
          PS3.setLedOn(LED4);
        }
      }
      if (PS3.getButtonClick(RIGHT)) {
        Serial.print(F("\r\nRight"));
        if (PS3.PS3Connected)
        {
```

```
        PS3.setLedOff();
        PS3.setLedOn(LED1);
      }
    //Forward

digitalWrite(M11,LOW);
digitalWrite(M12,HIGH);
digitalWrite(M13,LOW);
digitalWrite(M14,HIGH);

digitalWrite(M21,LOW);
digitalWrite(M22,HIGH);
digitalWrite(M23,LOW);
digitalWrite(M24,HIGH);


    }
    if (PS3.getButtonClick(DOWN))
    {
      Serial.print(F("\r\nDown"));
      if (PS3.PS3Connected)
      {
        PS3.setLedOff();
        PS3.setLedOn(LED2);
      }

      //Backward

digitalWrite(M11,HIGH);
digitalWrite(M12,LOW);
digitalWrite(M13,HIGH);
digitalWrite(M14,LOW);

digitalWrite(M21,HIGH);
digitalWrite(M22,LOW);
digitalWrite(M23,HIGH);
digitalWrite(M24,LOW);
    }
    if (PS3.getButtonClick(LEFT))
    {
      Serial.print(F("\r\nLeft"));
      if (PS3.PS3Connected)
      {
        PS3.setLedOff();
        PS3.setLedOn(LED3);
      }

//left
digitalWrite(M11,HIGH);
digitalWrite(M12,LOW);
digitalWrite(M13,LOW);
digitalWrite(M14,HIGH);

digitalWrite(M21,LOW);
digitalWrite(M22,HIGH);
digitalWrite(M23,HIGH);
digitalWrite(M24,LOW);

    }

    if (PS3.getButtonClick(L1))
```

```
    {
        Serial.print(F("\r\nL1"));

    //anti clockwise

digitalWrite(M11,HIGH);
digitalWrite(M12,LOW);
digitalWrite(M13,LOW);
digitalWrite(M14,HIGH);

digitalWrite(M21,HIGH);
digitalWrite(M22,LOW);
digitalWrite(M23,LOW);
digitalWrite(M24,HIGH);

    }
      if (PS3.getButtonClick(L3))
        Serial.print(F("\r\nL3"));
      if (PS3.getButtonClick(R1))
      {
        Serial.print(F("\r\nR1"));
     //clockwise

digitalWrite(M11,LOW);
digitalWrite(M12,HIGH);
digitalWrite(M13,HIGH);
digitalWrite(M14,LOW);

digitalWrite(M21,LOW);
digitalWrite(M22,HIGH);
digitalWrite(M23,HIGH);
digitalWrite(M24,LOW);

      }
      if (PS3.getButtonClick(R3))
        Serial.print(F("\r\nR3"));

      if (PS3.getButtonClick(SELECT)) {
        Serial.print(F("\r\nSelect - "));
        PS3.printStatusString();
      }
      if (PS3.getButtonClick(START)) {
        Serial.print(F("\r\nStart"));
        printAngle=!printAngle;
      }
    }
#if 0 // Set this to 1 in order to see the angle of the controller
    if (printAngle) {
      Serial.print(F("\r\nPitch: "));
      Serial.print(PS3.getAngle(Pitch));
      Serial.print(F("\tRoll: "));
      Serial.print(PS3.getAngle(Roll));
    }
#endif
  }
#if 0 // Set this to 1 in order to enable support for the Playstation Move
controller
  elseif (PS3.PS3MoveConnected) {
    if (PS3.getAnalogButton(T)) {
      Serial.print(F("\r\nT: "));
      Serial.print(PS3.getAnalogButton(T));
```

```
    }
    if (PS3.getButtonClick(PS)) {
      Serial.print(F("\r\nPS"));
      PS3.disconnect();
    }
    else {
      if (PS3.getButtonClick(SELECT)) {
        Serial.print(F("\r\nSelect"));
        printTemperature=!printTemperature;
      }
      if (PS3.getButtonClick(START)) {
        Serial.print(F("\r\nStart"));
        printAngle=!printAngle;
      }
      if (PS3.getButtonClick(TRIANGLE)) {
        Serial.print(F("\r\nTriangle"));
        PS3.moveSetBulb(Red);
      }
      if (PS3.getButtonClick(CIRCLE)) {
        Serial.print(F("\r\nCircle"));
        PS3.moveSetBulb(Green);
      }
      if (PS3.getButtonClick(SQUARE)) {
        Serial.print(F("\r\nSquare"));
        PS3.moveSetBulb(Blue);
      }
      if (PS3.getButtonClick(CROSS)) {
        Serial.print(F("\r\nCross"));
        PS3.moveSetBulb(Yellow);
      }
      if (PS3.getButtonClick(MOVE)) {
        PS3.moveSetBulb(Off);
        Serial.print(F("\r\nMove"));
        Serial.print(F(" - "));
        PS3.printStatusString();
      }
    }
    if (printAngle) {
      Serial.print(F("\r\nPitch: "));
      Serial.print(PS3.getAngle(Pitch));
      Serial.print(F("\tRoll: "));
      Serial.print(PS3.getAngle(Roll));
    }
    elseif (printTemperature) {
      Serial.print(F("\r\nTemperature: "));
      Serial.print(PS3.getTemperature());
    }
  }
#endif
}
}
```