Voice Controlled Car

A Project Report
Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Saurabh Nitin Kawli
Seat No.: 3039357

Under the esteemed guidance of

Mrs. Varsha Kiranpure

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
SHAILENDRA EDUCATION SOCIETY'S
ARTS, COMMERCE AND SCIENCE COLLEGE
(Affiliated to University of Mumbai)
MUMBAI 400 068
MAHARASHTRA
2018-2019

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No.: …… Seat no.: _____

1. Name of the Student

   _____

2. Title of the Project

   _____

3. Name of the Guide

   _____

4. Teaching experience of the Guide _____

5. Is this your first submission? Yes ☐ No ☐

Signature of the Student Signature of the Guide

Date: …… Date: …….

Signature of the Coordinator

Date: ……

SHAILENDRA EDUCATION SOCIETY'S

ARTS, COMMERCE & SCIENCE COLLEGE

(Affiliated to University of Mumbai)

DAHISAR (E), MUMBAI, MAHARASHTRA-400068

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Voice Controlled Car"**, is bonafied work of **Saurabh Nitin Kawli** bearing Seat No.: **3039357** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: **College Seal**

ABSTRACT

Voice Control Technology is a very efficient way of practicing voice searches and controlling different things. Sustainable IT services require the integration of Voice Control Technology which can help in power management, virtualization, improving technology, and optimization of the IT Technology to meet sustainability requirements. This paper provides a review of the literature on sustainable IT, key areas of focus, and identifies a core set of principles to guide sustainable IT service design.

It's great if you make changes, but you don't want to slip into bad habits, and you also want to stay on top of any trends and changes. This topic is totally based upon Internet of things. Internet of things is space where physical computing collides with Internet, where inspired design combines with hardware, software and the web to create objects that are fun, interactive and practical. Programmed with the ability to sense and respond to online triggers or changes within the object's environment, the Internet of Things brings automation and interactivity to previously inanimate objects that delight and amuse, Or both!

That's what we will be seeing throughout the project.

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I would like to express my deepest appreciation to all those who provided me the possibility to complete project topic on "**Voice Controlled Car***"*. I am extremely thankful to our principal Dr. Swati Pitale. I heartily thank our Coordinator Prof. Hemchandra Kumbhar and internal project guide Prof. Varsha Kiranpure whose valuable guidance and suggestions helped us in accomplishing my project. I also express gratitude towards our Computer Lab Assistant Mr. Swapnil Jadhav and Mr. Jay Galchar who provided us with all the lab facilities. Further, I would like to extend my sincere esteems to all Library staff for providing reference project reports books.

Finally I would like to thank my friends for their cooperation to finalizing this project within the limited time frame.

- Saurabh Nitin Kawli
  Seat No. 3039357

DECLARATION

I hereby declare that the project entitled, "**Voice Controlled Car**" done at Shailendra Education Society's Arts, Commerce and Science College, Dahisar (E) has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Saurabh Nitin Kawli

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Voice control car is a car that runs on the voice commands given by a person. this technology doesn't require any button controls. Means your hands are completely free from the holding a remote and you can do your other work or tasks. Just say a command and it will execute it.

Voice command technology is also used in industry level, and also for home purposes. this technology is possible just because of research and development done by companies, the companies like Microsoft, Apple, Amazon and etc., this company created their own speech recognition technology, like Microsoft has created Cortana, Apple has Siri, Amazon has Alexa. For example, Alexa is a device controlled by voice command and it can control many home appliances which are connected through internet.

Arduino is an open-source device based on easy-to-use hardware and software. Arduino boards are able to connect to sensors read inputs - light on a sensor, a finger on a button, and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board using programming which is saved in a Atmel Processor. To do so you use the Arduino's own Programming Language written on Platform provided Arduino itself known as Arduino IDE (Integrated development environment).

The technology used in this project is based on voice commands which are received by Arduino using Bluetooth module. Bluetooth is a best device for the short-range wireless interconnectivity of smartphones, computers, and other electronic devices which is under the range of Bluetooth device. It is a telecommunications industry specification. It describes how mobile devices, computers and other devices can easily communicate with each other using a short-range wireless connection in modern world.

This project is sort of system in a car to recognize voice of a human language, spoken aloud, and executes the spoken instructions. This could refer to something as basic as recognizing a few commands to run a car via Bluetooth, all the way up to voice commands to exercise some functions in the car itself. Much of the technology that exists today is very limited and inaccurate to the point, but the technology keeps improving.

1.2 OBJECTIVES

The Arduino board connected to various components where Bluetooth is one of the components used in this project. Where Bluetooth receives signal from smartphone and transmits that signal to Arduino board from the Android app as String with help of google translating tool which is automatically done by the android app. There are words pre-programmed (go, back, right, left, round and stop) to the Arduino. Whenever Arduino receives a String it matches with the pre-programmed words in the program & Arduino executes that command and moves the according to command given to it.

The important thing that is covered by this project is how voice controlling can make our life style easy it can not only be implemented in a car but also in various regions. It is totally based upon Internet of things which itself is a very interesting topic to work on Internet of thing is the future of the computing world making things connected to the Internet and controlling them from various region of the globe makes us satisfied and saves a lot of time. Also, sensors play an important role in Internet of Things as it is a part of the Internet of Things equation. Internet of Things equation is the components that are used to consider that thing into Internet of Things. It consists of Physical Object, Sensors, Controllers, Actuators & Internet.

So, in this project the main aim will be to move a car across using only voice commands and using the sensors to sense objects around it and react to the situation according to the programming instructions.

1.3 PURPOSE, SCOPE, AND APPLICABILITY

1.3.1 Purpose

This project consists of a sensor called as ultrasonic sensor, which can detect obstacle in front of it, when it detects an obstacle it stops the car or it can execute any other command like turn left or right based on the programming code written by programmer. It can be effectively implemented in real world like in the night time it may happen in certain places there might be no light available on streets, so sometimes the driver is not able to react quickly if somethings suddenly appear in front of driver. In such case a sensor in the car can react to that situation and saves a huge accident which can save a lot of damages like car crashing, human loss/injury. This ultrasonic sensor works on sonar-based technology that measures the distance to an object using ultrasonic sound waves

This ultrasonic sensor uses a transducer to send & receive ultrasonic pulses that relay back information about an object's nearness. High-frequency sound waves are reflected from boundaries to produce distinct echo patterns. Ultrasonic are independent of light, smoke, dust, color, material (except for soft surfaces, i.e. wool, because the surface absorbs the ultrasonic sound wave and doesn't reflect sound.). Ultrasonic Sensors are best used in the non-contact detection of Presence, Level, Position, Distance Non-contact sensors are also referred to as proximity sensors used to determine the nearness of the object**.** So, an ultrasonic sensor can be used in harsh conditions.

Other than sensors if a driver has a predefined route to his home then he can just give a voice command to the car like "Use Route to Home" and it will search path using google maps and the car will automatically drive us back home safely. Voice searching is mostly popular within younger demographics, with 59% of 14 - 28 years old are using the system on their smartphones. As the use of voice searching is increasing it is becoming better and more accurate day by day, the more people will use them, more it will be updated as per requirement. By 2020, it is predicted that more than 50% of all device interface and interaction will be totally voice-based. Voice searching is a very fast, effective, easy and efficient/convenient hands-free way to operate a device. The voice controlling technology is been used a lot in our smartphones and toys but it lags behind in the usage of many other real-world things. The main purpose of the project is being able to use this voice controlling technology in various fields which can make our way of living very easy.

Voice control technology can change the how we see the world now. It has a power to control very small things to huge machines. Creating something that can change things how they actually work makes us satisfied and looking at the technology being upgraded every single day. Its not far from now that everyone will be using voice controlling technology as frequently as they use their smartphones. Based upon the technology used today many of the programmers are working with Artificial Intelligence which goes hand in hand with voice control is increasing day by day. AI these days compared to back in 90s are very powerful and fast and they are starting to adopt voice-based controlling as they grow faster.

1.3.2 Scope

"Voice Controlling is the future", the world's technology experts are claiming for vital market share, with ComScore projecting that "50% of all searches will be voice searches by 2020."

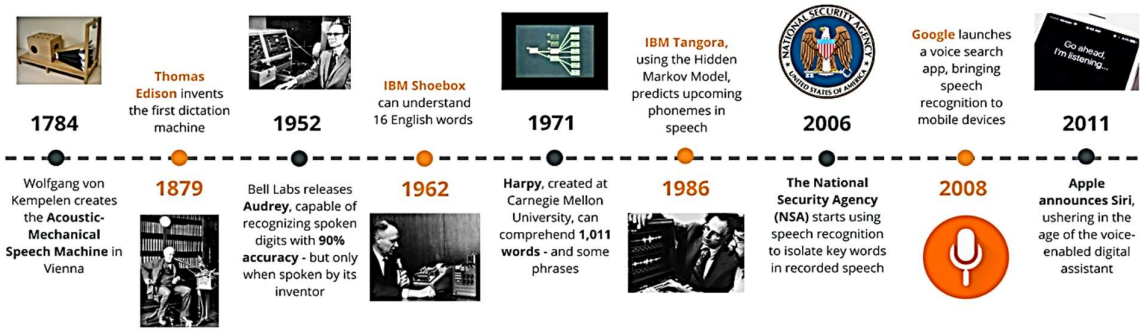


Fig 1.1: Evolution of voice Technology

In our day to day life we use google most of the time then whether it is to search information or to download music, videos, etc. or maybe we use it to send mails or check mails. Over a **40,000** of search queries that are processed by Google every second on an average, around over **3.5 billion** searches per day, and **1.2 trillion** searches per year worldwide as listed by Internet Live Stats. And why is that the people use google so much it's because it is fast and gives more relevant results? maybe but there is also one thing that google is more concerned about and that is improvement in its interface google keeps on adding new ways that can attract more users towards itself.

How Google plays a role in Voice search?

The answer is "Ok Google" another feature of google which is used to do the searches similar to the one we do by typing on google it is done by voice command its listens what has been said by user and then it itself searches it in its search engine and displays us the most relevant accurate result. This made google searching even more effective, attractive and easy.

We don’t even need to touch the device to do so just speak ok google and ask google whatever you need. There’s not only google but there are even more companies which use this technology like Apple has Siri, Amazon has Alexa, Microsoft has Cortana, Samsung has Bixby.

How many times in a day have you set a reminder, opened map to search for some place, sent a message or searched for the answer to a question, simply by speaking into your smartphone? Voice technology, with the combination of built-in AI assistants like Siri, Alexa and Cortana have made these situations common place, and it's changing the way we live our daily lives as it’s upgrading itself day by day.

What Is Google Hummingbird?

Google uses an algorithm that is named as “Hummingbird Algorithm”, and it’s used as a new search platform that Google is using from September 2013, the name “Hummingbird” comes from being “precise and fast” and is designed to focus better on the meaning behind the word Hummingbird.

The role of Hummingbird is to pay more attention to each word in the query, ensuring that the whole query or the whole sentence or conversation or meaning is taken into consideration while searching, rather than particular words. The goal is to achieve pages matching with the meaning do better, instead of pages matching just a few words which can show different results. Google Hummingbird is designed in such a way to apply the meaningful technology to billions of pages from across the world wide web, in addition to Knowledge Graph facts, which may give better results.

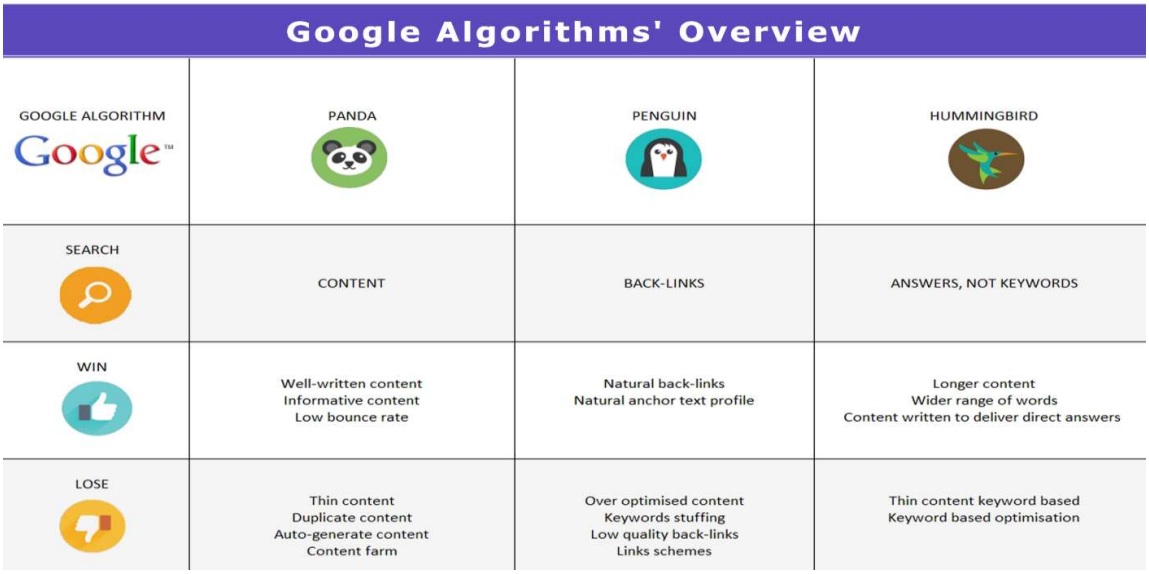


Fig 1.2: Google Algorithm’s Overview

Difference Between Voice Search and Text Search

Here is an interesting fact, an average person can type at least 35-45 words per minute on keyboard, but can speak up to 115-155 words per minute. It is approximately 3.75x faster than typing on a keyboard.

Let's have look in the key differences between voice searching & text searching.

1. Voice search queries are much longer than the text query.

Now a day's People are lazier and when it comes to typing they prefer short phrase to save their time and energy like "good morning" has become "gm".

On smartphones, you talk to a virtual assistant (like Siri, Alexa, etc.) for help in your natural language that is often longer than phrases that you type on the keyboard.

Google's artificial intelligence (AI) is getting smarter every day as it learns with every search query user's ask to it.

2. Voice search queries contains of local search intention

Smartphones voice searching queries consists 3 times more of the local intent than a normal text-based searching. As a matter of fact, to know, two of the cases which are popularly used by most of the users are, when voice searches are mostly performed are asking for direction and looking for a nearby location. Smartphones search for **"near me"** have grown more than 130% year over year says Google's survey. And as google gets this data it also stores it and next time it will automatically set into your daily routine done by AI.

Handicapped/Disables peoples suffer a lot of difficulty of movement inside their home some even can't spend some basic needs and necessities, usually they need another person to help them to do that for them.

Voice assistance can provide them a little help through the design of a special interface for handicapped people to control things around their house by using voice commands. It will save a lot of time and it will also satisfy the person as he won't need much help from others.

1.3.3 Applicability

Applications of implementation of Ultrasonic Sensors:

Ultrasonic Sensor Distance Measurement:

Distance measurement would be applied in a garage parking application, sensing where vehicle will be pulled completely into a garage without damaging it.

Ultrasonic Sensors for water level detection:

Tank level measurement to detect if the tank is full or not if full it will detect and close the water supply to the tank automatically, Fuel gauging, irrigation control.

Ultrasonic Sensors for Obstacle Detection:

Our Unmanned aerial vehicle (UAV) Sensors for Drones as well as our proximity sensors that are used for robots, machines to detect the objects nearness and move accordingly.

Motors:

Another important component used in the project is Motor there are lots of uses of motors in various field across different fields according to different types of motors:

1. Direct Current Shunt Motor

It's a constant speed motor the speed remains almost constant always from no-load to full load where the load has to be driven at a number of speeds and any one of which is nearly constant.

It is used in various Industries like Drills, Boring mills, Shapers, Spinning, Weaving machines, etc.

2. Direct Current Series Motor

It's a variable speed motor the speed of motor is low at high torque at light, the motor speed can go up to a dangerously high speed. This motor can have a very high starting torque. (elevators, electric traction).

It is used in various Industries like Electric traction, Cranes, Elevators, Air compressor, Vacuum cleaner, Hair drier, Sewing machine, etc.

3. Direct Current Compound motor:

These types of compound motors are rarely used because of its very poor torque characteristics.

It is used in various Industries like Presses Shears, Reciprocating machine.

4. AC motor:

An **Alternate Current motor** is an electronic motor used by providing an alternating current (AC). The Alternate Current motor basically consists of two commonly available parts, an outside stator having surrounded coils supplied with alternating current to develop a rotating magnetic field, and an inside rotor that is attached to the output shaft producing a second rotating magnetic field. The rotating magnetic field can be produced by permanent magnets, reluctance saliency, or DC or AC electrical windings.

Alternating Current linear motors operate on a same principle as rotating motors but they have stationary and moving parts that are arranged in a straight-line configuration, producing linear motion instead of rotation.

5. LDR:

Controlling the lights and home appliances is usually done manually and operated in many occasions. But due to the negligence of humans or unusual circumstances, the process of controlling the devices can cause power dissipation. To overcome this problem, we can use light-dependent resistance circuits to control the load based on the intensity of light. An LDR or a photo resistive device is made of high resistance semiconductor material.

1.4 ACHIEVEMENTS

Following are the goals that we achieved during the project being made:

- **Voice Controlling**

  The main aim of this project is to be able to move a car by just giving it voice commands. This project gave us an idea that how we could control things around us just without touching them and by using very small chips which are easily available to us in the market. This involves various other application in our home that can be controlled just like we are talking to them isn't that amazing. Imagine you need to google something and you are bored of typing it so instead we can communicate with google by saying OK Google after which google will listen to you and do the job for you.

- **Saving Time and Energy**

  This project implementation can save our time, money, energy in various ways if we can correctly implement its application in our real life, we also use such things like Siri, Alexa, etc. manually giving a command is a bit more time consuming instead letting the machine do it for you it makes it faster and better just we need to give him commands just like we talk with others.

- **Usage of the technology**

  This technology has been seen very often in our smartphone, toys, etc. But we don't get to see its implementation in real world objects. This project gave an idea that how can we make it happen. All the time we utilized in working on this project we got to learn many new things out of it. Most of the time we implemented it in various things.

- **Relating project with our subjects**

  The project we are implementing has its total base in our core subject Internet of Things but it is not the only subject our project is based upon it also has programming part which we have been learning from first year through different languages, regardless of the different languages the basic algorithm of the code that we learnt is same everywhere so that made its implementation easy. Other than that, it also has a bit of a networking part which we have learnt in second year which helped us in understanding our project faster. Overall the base syllabus learning helped us a lot throughout the project.

- **Making a Prototype**

  While planning this project we decided to make prototype which will exactly work same as the main project. We went through a lot of difficulty doing that but we managed to solve them quickly and made a prototype which actually worked. Making a prototype gave us an idea of where we need to concentrate more which topic is important and which is not. Because of the prototype we already now know where we need to take precaution, how to handle a particular error, and many more things.

- **Using Sensors**

  Sensors have made serious inroads into automotive, medical, industrial, and aerospace applications. Rising concerns for safety, convenience, entertainment, and efficiency factors, coupled with worldwide government mandates, will see sensor usage swell to unprecedented levels. Add to that the predicted explosion in wireless and consumer applications, and one can see why sensor manufacturers anticipate quickly developing huge markets and applications through the end of this decade. Most of these sensors will be of the microelectromechanical-system (MEMS) and microsystem-technology (MST) type, with nano-sensors showing great promise.

1.5 ORGANIZATION OF REPORT

Later in this project report there is a lot more information about projects actual working and a detailed information about each equipment that are being used in this project. The software that is used for programming and the algorithm that will be used for this program as well. There will be charts and diagrams which will make understanding the project easier. Gantt chart will give an overview of how we planned out the working of the project throughout the semester.

All these will be covered in the following ways in the coming chapters:

- **Survey of Technologies**

  It will consist of the detailed information about the technology used in this project covering all the important technology usages in today's time related to the project. It will give a brief information about all the equipment's that are used in the project. It will also discuss the programming tool used to program the code which is software part of the project. This topic will explain why we choose this technology for the completion of the objectives of the project

- **Requirements & Analysis**

  In this topic we will be discussing the component and overview of requirements and analysis. We will be describing the problem definition of the project that we will achieve from the project. And all the hardware requirements for the project. We will also give all the details of conceptual models such as data flow diagram, ER diagram, use case diagrams that are in our project. Planning & scheduling of the project will also be included in this chapter. The Problem Definition, Requirements, Specification, Planning and Scheduling, Hardware Requirement, Software Requirements, Conceptual Models will be covered in this chapter.

- **System Design**

  This topic will consist of various diagrams and algorithm which are needed for the program it will also consist of the basic circuit diagram showing the cars whole connection with details.

CHAPTER 2: SURVEY OF TECHNOLOGIES

Arduino Uno:

Arduino UNO is a micro-controller-based board which uses the ATmega processor. this micro-controller has 14 digital I/O pins in which 6 can be used as PWM outputs, 6 analog inputs from A0 to A5, a 16 MHz quartz crystal, it has a USB connection port which is used to insert a code to the Arduinos memory it can even be used for power supply, and it also have power jack which is actually used for power supply, an ICSP header and a reset button.

Arduino UNO contains everything which is needed to support the microcontroller; simply we need to connect it to a computer via a USB cable. This USB cable Power's it with a AC-to-DC adapter or battery to get started. You can do Research & Development (R&D) with your Arduino UNO without worrying too much about something going wrong with it, if micro-controller chip gets damage you can easily replace that chip for a few rupees and start your project again.

The word "Uno" in Italian it known as the number "one" and it was chosen to mark the release of Arduino Software (IDE) 1.0. The Arduino Uno board version 1.0 of Arduino Software (IDE) were the referred from the versions of Arduino, now evolved to newer releases. The Arduino Uno board is first series of USB Arduino boards, and reference model for the Arduino platform for an extensive list of current, past or outdated boards see the Arduino index of boards.

Front side of Arduino UNO board: **Back side of Arduino UNO board:**

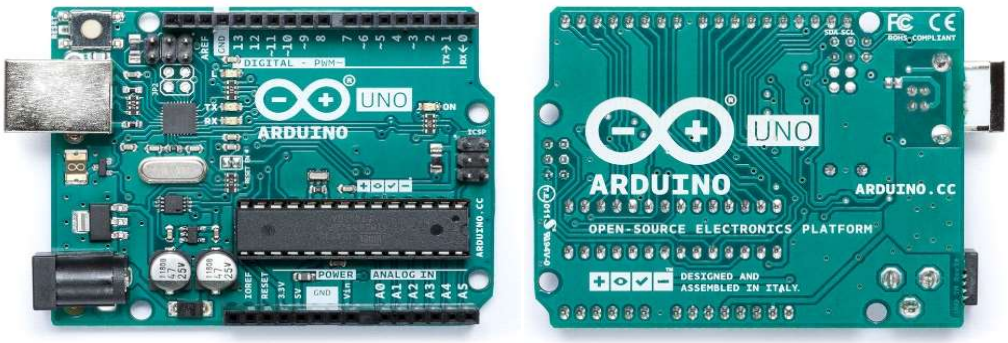


Fig 2.1: Arduino Board

There are also different variants of Arduino available which can have more or less functionality one can use it as per his/her project requirement. Some famous Arduino used for projects are:

- Arduino Nano
- LilyPad Arduino Board.
- RedBoard Arduino Board.
- Arduino Mega (R3) Board.
- Arduino Leonardo Board.

Motor Driver L298N:

L298N dual bidirectional motor driver is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily connect and independently control four motors of up to 2.5A each in both directions. It's ideal for robot applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfered with simple manual switches, TTL logic gates, relays, etc.

Two high – power current drawn on each motor through your microcontroller. An on L298N board user can access 5V regulator and incorporated which can also be used to supply any additional circuits requiring a regulated 5V DC supply of up to about 1A

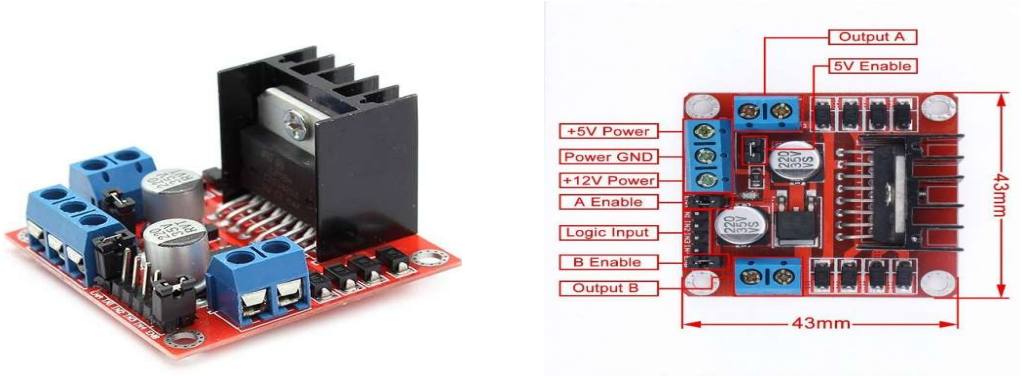


Fig 2.2: L298N Motor Driver

Specifications

- Driver L298 Dual H Bridge is DC Motor Driver IC
- Operating Voltage: 7 to 35V, Peak current: 2A
- power consumption for Driver: 20W (when the temperature T = 75 °C)
- Size of the Driver is 55*49*33 mm (fixed copper pillar and the heat sink height)
- Weight of the Driver Board is 33g

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

Bluetooth Module HC-05:

Bluetooth moduleHC-05 is an easy to use Bluetooth SPP (Serial Port Protocol) module, and it is designed for transparent and wireless serial connection with other devices.

The serial port of the Bluetooth module HC-05 is fully suitable for Bluetooth module V2.0+Enhanced Data Rate (EDR). Bluetooth has 3Mbps Modulation and also comes with a complete 2.4GHz radio transceiver and baseband. It also uses CSR Bluecore 04-External single chip Bluetooth system which comes with CMOS technology and with Adaptive Frequency Hopping Feature (AFH). This will simplify your overall design & development cycle.

Microchip manufacturer delivers the broadest range of Bluetooth certified solutions for embedded designs. All Bluetooth device are Qualified Design Identification (QDID) to ensure conformity for your connected solution. Microchips are highly-integrated module solutions and they are self-contained, low-power, and fully-certified from (QDID) for designers who are seeking to develop a IoT devices without any Bluetooth Low Energy.

The Bluetooth Device which we have used in this project look like this given in below picture.

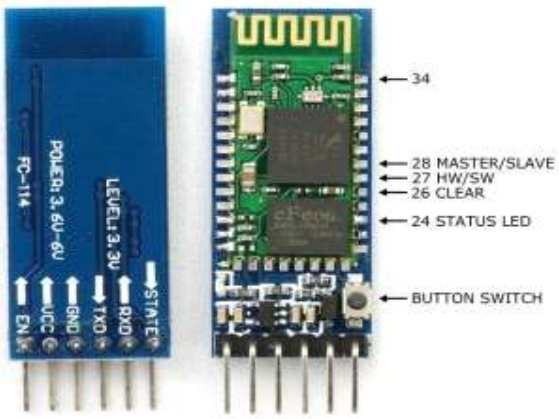


Fig 2.3: Bluetooth Module

Bluetooth 5:

Bluetooth SIG's newest specification, Bluetooth, is now supported by select Microchip devices. Microchips are re-certified the current products to ensure that new and existing designs compatible to this standard. To check Bluetooth are certified products or not.

Ultra-Low Power:

Microchip's are optimized for battery-powered applications with a wide voltage range from 1.8V to 4.3V and TX/RX current (3.6V, 0dBm Pout). This results in ultra-low-power connection events, with the industry's lowest electric charge made from the battery for each event. This extended battery life is 2 to 4x compared to existing BLE solutions.

Ultra-Small Form Factor:

Microchip's modules are highly integrated and require few external components. Microchip manufacturer offers the smallest BLE 4.1 SoC available in the market, the size is only 2.26 x 2.14 mm.

Specifications for Bluetooth device:

Hardware Features:

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna and edge connector

Software Features:

- Default Baud rate is 38400, Data bits are 8, Stop bit are 1, there is No parity.
- Data control also supports baud rates are 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 this pins can be connected to red and blue led individualy.
- When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- It Auto-reconnect in 30 min when it disconnected.

Micro servo motor sg90:

Servo Motors:

This includes a wide range of products like tower pro sg-90 micro servo 9g motors, futaba s3003 servo motor, tower pro mg995 metal gear servo motor, tower pro mg996r servo motors, tower pro sg-5010 - standard servo with servo lever & arm and tower pro mg90d metal gear micro.

Important Notes:

1. Specifing the connector type whenever you buy a servo motor.
2. Please choose correct model for your application.
3. Torque over-loaded will damage the servo's mechanism.
4. Always Keep servo clean and away from dust.

Tower Pro SG-90 Micro Servo 9g Motors:

Description:

- Size:12*23*23
- Speed:0.14sec/60degree at(4.2V)
- 0.12sec/60degree at(5V)
- Torque:06.kg.cm.at(4.2V)
- 0.8kg.cm.at(5V)
- Voltage:4.2V-5V
- Package: One polybag one piece
- Feature: Supports Futaba, JR, Sanwa and Hitec remote system.

Pack usually consists of:

- 9g Servo x 1
- Propeller X 3
- Screw X 3

Fig 2.4: Servo and its components

What's inside the servo?

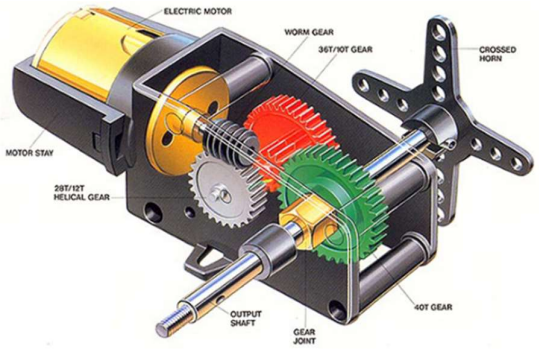


Fig 2.5: Inside the servo

To understand actually how the servo motor works, you should look inside the plastic case. Inside the plastic case there is a simple set-up of a control circuit, a small Direct Current, and potentiometer and the motor is actually attached by gears which controls the wheels.

While the shaft in the motor are in proper position, the power supplied into the motor is being stopped. If the power supplied does not stop, the motor is being turned in an appropriate direction. The proper position of the motor is sent via an electrical pulses via a single wire. The actual motor's speed is proportional to difference between the actual position and the position which we want. Hence, if the motor is near to the position, then it will rotate slowly, or it would rotate faster. It is also known as proportional control. That means that the motor will only run as fast as it is necessary to complete the required task at hand.

How is the servo controlled?

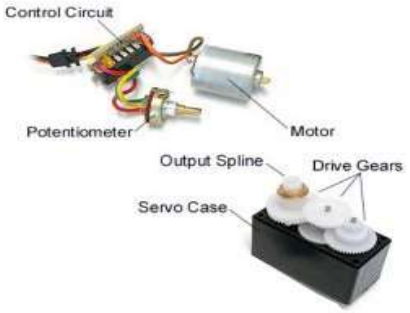


Fig 2.6: Controlling Servo

Strength of the servo motor (L) and the assembled servo (R)

Servo motor is controlled by an electrical pulse with a varying width, or with a Pulse Width Modulation (PWM), with the control wire. There are a repetition rate, a maximum pulse rate, and minimum pulse rate. This servo motor usually can only rotate 90° in either directions for the overall total of 180° movement. First position of motor is defined as the position from where the servo motor have the equal amount of potential rotation in both the directions that is clockwise as well as counter-clockwise direction. The PWM (Pulse Width Modulation) are sent to motors and then it determines the correct position of the shaft which is based on the duration of the pulse which are sent via control wire.

Then the servo motor looks forward to look for the pulse every 20 milliseconds (ms) and the length of the pulse will be determined that how far the motor turns. For example, A 1.5 milliseconds pulse will make the motor rotate to the 90° position.

Shorter than 1.5 millisecond it moves in a counter clockwise direction towards the 0° position, and longer than 1.5 millisecond it will rotates the shaft of servo motor in a clockwise direction towards the 180° position.

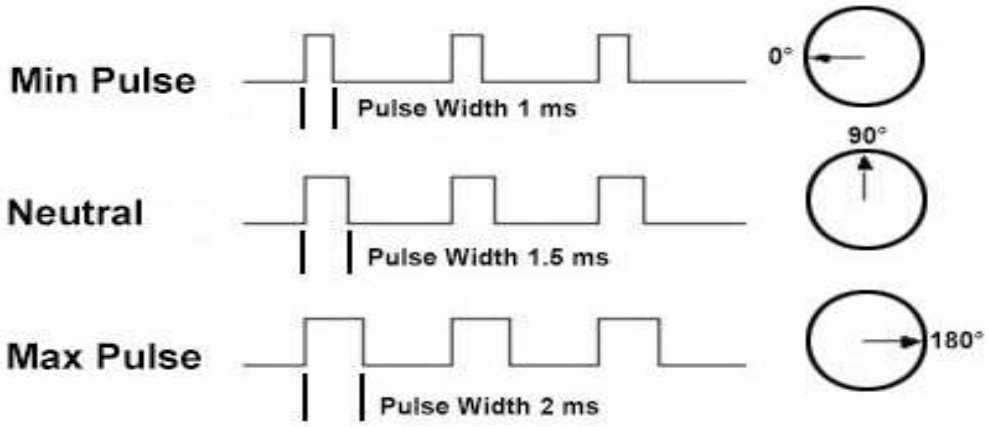


Fig2.7: Variable Pulse width control servo position

When these servo motor get the commands to move, they will move to the position and hold on that position. If external force pushes against the servo motor while the servo motor is holding an position, the servo motor will be against from moving out of that position. The max amount of force the servo motor can pull is called the torque rating of the servo motor. Servos will not hold their position forever. the position of pulse must be repeated to instruct the servo to stay in position.

Types of Servo Motors:

There is two types of servo motors available - AC and DC motors. AC (Alternating Current) servo handles higher current tend to be used in industrial machinery. DC (Direct Current) Servos are not designed for high current and are usually better suited for small applications. Generally, DC motors are less expensive than the AC counterparts. These are also servo motors but that have been built specifically for continuous rotation. The features of two ball bearings on the output shaft for reduced friction and easy to rotation.

Ultrasonic Sensor HC – SR04:

The HC-SR04 ultrasonic sensor uses sonar techniques to measure the distance from any object. This technique especially used by animals like bats or dolphins. It offers non-contact range detects with the high accuracy and stable readings and it is easy-to-use. The range is Distance From 2cm to 400 cm or 1inch to 13 feet. The operation is not affected by sunlight or any other black material like Sharp things. It have complete package of ultrasonic transmitter and receiver module.

The Ultrasonic Sensor HC – SR04 shown in this project, and some features and shared Arduino Project examples that you can follow the integrates in your projects.

Features:

- Power Supply:+5V DC
- Quiescent Current: <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2cm – 400 cm/1″ – 13ft
- Resolution: 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

Pins:

- VCC: +5VDC
- Trig: Trigger (INPUT)
- Echo: Echo (OUTPUT)
- GND: GND

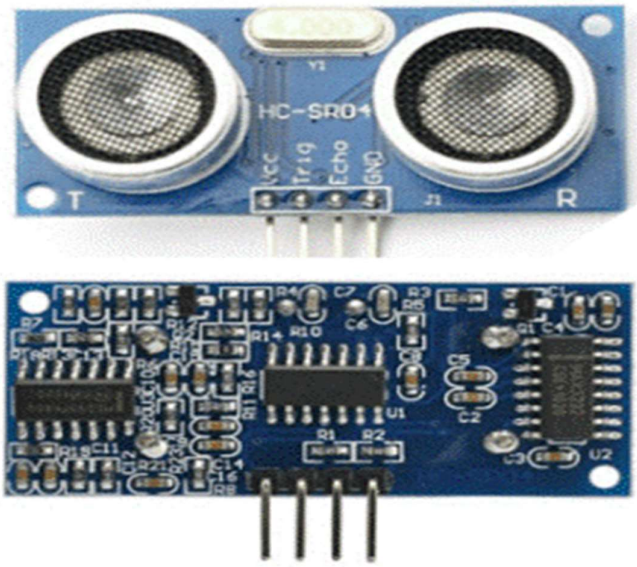


Fig 2.8: Ultrasonic Sensor

Different Types of Ultrasonic Sensors:

There are different types of ultrasonic sensors,

1. Ultrasonic Proximity Sensors
2. Ultrasonic Retro-reflective Sensors
3. Ultrasonic Through Beam Sensors
4. Ultrasonic 2 Point Proximity Switches

How Does it Work?

The ultrasonic sensor uses sonar technology to determine the distance from an object. This technology also used in ships, planes and etc. Here's what happens:

1. The transmitter (trig pin) sends a signal: a high-frequency sound.
2. When the signal finds an object, it is reflected and…
3. The transmitter (echo pin) receives it.

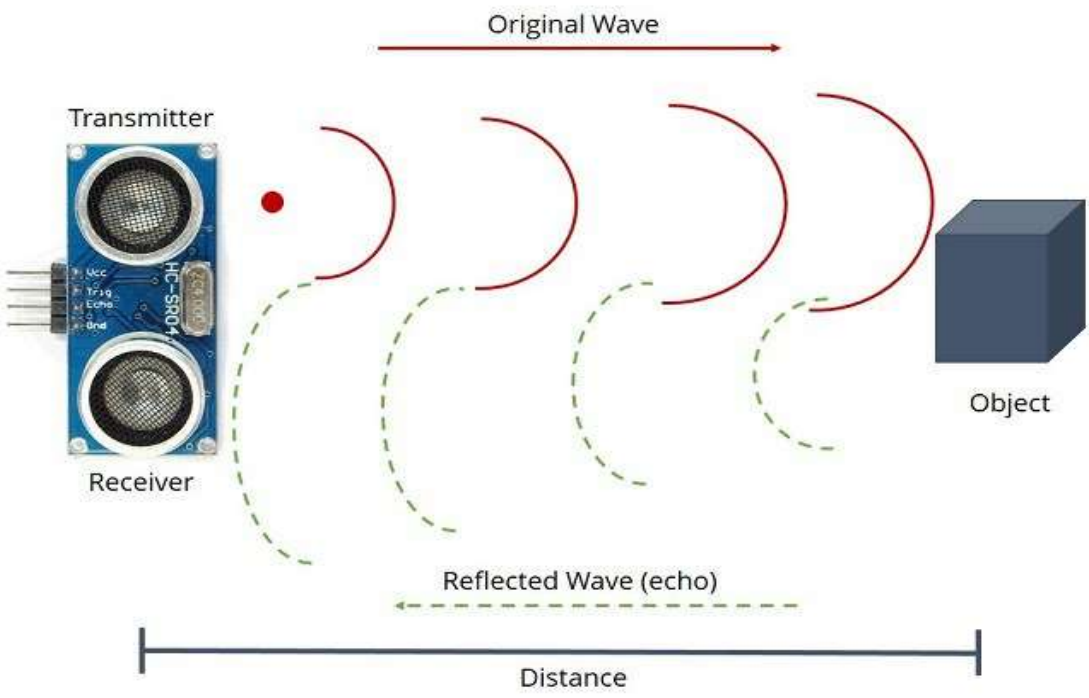


Fig 2.9: Working of ultrasonic sensor

The time between the transmission and reception of the signal allow to know the distance from any object. This is possible because of knowing the sound's velocity in the air. Sensor

Advantages of Ultrasonic sensor:

1. Ultrasonic sensor has a very high frequency it is highly sensitivity and has high curved power so that it can detect external and deep objects very easily.
2. Ultrasonic sensors can easily interface with any microcontroller or any type of other controllers.
3. This ultrasonic sensor has very high accuracy compared to other methods to measure the thickness and the depth of parallel surface.

Disadvantages of Ultrasonic sensor:

1. While using the ultrasonic tester for testing it is very important to know about the operational manual, in other words it requires a careful attention for technicians who are experienced.
2. While the development of the product the inspection procedure of sensor using ultrasonic tester requires an extensive knowledge and testing technicians.
3. While the ultrasonic sensors are used for testing purpose they should be water resistant otherwise they could get damaged.

Motors:

The motor which we have used in this project is DC motor and it comes with a gear box which provides good torque and rpm at lower voltages of DC current. This type of DC motor are run at around 200rpm when it driven by a Dual Hi-Watt cell battery at 6 V and approximately 300 rpm when it get connected to 9V. It is most suitable for DC motor for light weight project which run on small voltage.

Features:

- Working voltage: 3V to 9V
- 30gm weight, light weight geared dc motor
- 1.9Kgf.cm torque

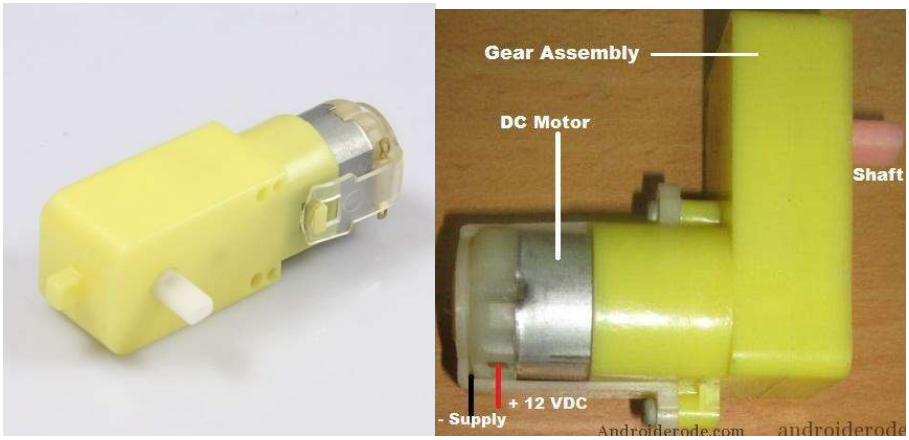


Fig 2.10: Single Shaft Bo Motor

In the project we are using 4 single shaft Bo Motor with tyres.

Fig 2.11: Tyres

Batteries:

The nine-volt battery is a common battery used give power supply to any device which runs on 9V Battery. It has a rectangular shape with two rounded edges and a polarized. This type of battery is commonly used in walkie-talkies, clocks and smoke detectors.

The nine-volt battery are commonly made from carbon-zinc and alkaline chemistry, in primary lithium iron disulfide, and in rechargeable form in nickel-cadmium, nickel-metal hydride and lithium-ion.

Most of nine-volt batteries are constructed of six individual 1.5 V LR61 cells enclosed in a wrapper. These cells are smaller than LR8D425 AAAA cells and it can be used in for some devices, even though they are 3.5 mm shorter.

Types of batteries:

- D
- C
- AA
- AAA
- AAAA
- PP3 (9-volt)

Fig 2.12: 9V Battery

LDR:

An Light Dependent Resistor (LDR) is a component that consists of a (variable) resistance that is changed with the intensity of the light that falls upon it. This allows the LDR to be used in the light sensing circuits.

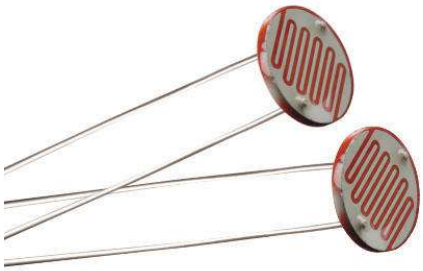


Fig 2.13: An LDR (Light Dependent Resistor)

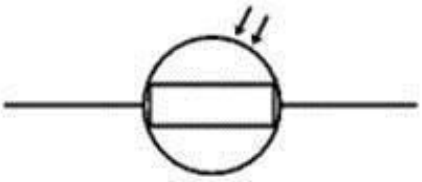


Fig 2.14: LDR Circuit Symbol

Variation in resistance as the light intensity is changing

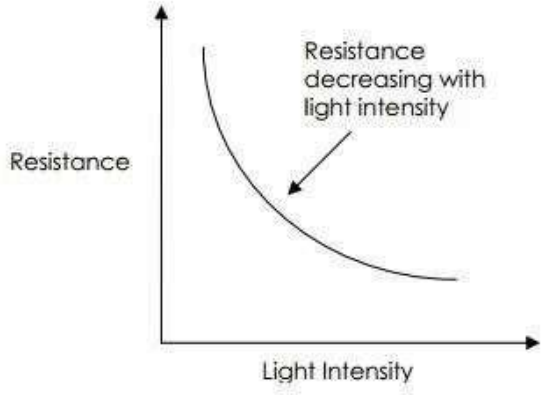


Fig 2.15: LDR resistance vs light intensity graph

The most commonly found LDR type has a resistance that falls with an increase in the light intensity falling upon the device (as shown in the image above). The resistance of an LDR may typically have the following resistances:

Daylight = 5000Ω

Dark = 20000000Ω

You can therefore see that there is a large variation between these figures. If you plotted this variation on a graph you would get something similar to that shown by the graph shown above.

Applications of LDRs

There are many applications for Light Dependent Resistors. These include:

Lighting switch

The most obvious application for an LDR is to automatically turn on a light at a certain light level. An example of this could be a street light or a garden light.

Camera shutter control

LDRs can be used to control the shutter speed on a camera. The LDR would be used to measure the light intensity which then adjusts the camera shutter speed to the appropriate level.

As this there are other various Application such as:

1. The Light Dependent Resistor (LDR) used for automatic contrast and brightness control in television receivers.
2. The LDR's are used in the infrared astronomy.
3. The LDR's are used in optical coding.
4. Used in light activated control circuits.
5. Used in light failure alarm circuits and used in light meter.
6. The LDR's are used in smoke detectors.
7. Used in the security alarm.
8. The LDR's are also used in street light control circuits.
9. It is used in camera light meters.
10. Used in photosensitive relay.
11. It is also used as a proximity switch.

LED:

A **light-emitting diode (LED)** is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. This effect is called electroluminescence. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared light. Infrared LEDs are used in remote-control circuits, such as those used with a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Modern LEDs are available across the visible, ultraviolet, and infrared wavelengths, with high light output.

Early LEDs were often used as indicator lamps, replacing small incandescent bulbs, and in seven-segment displays. Recent developments have produced white-light LEDs suitable for room lighting. LEDs have led to new displays and sensors, while their high switching rates are useful in advanced communications technology.

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper and medical devices.

Unlike a laser, the colour of light emitted from an LED is neither coherent nor monochromatic, but the spectrum is narrow with respect to human vision, and functionally monochromatic.

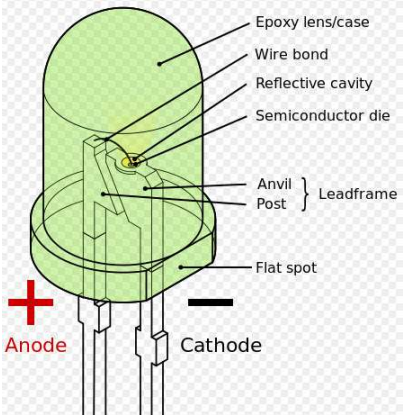


Fig 2.16: LED

Jump wire:

A jumper wire is a wire which is used to transfer electrical signals between two points in a circuit. These wires can either used to modify the circuits or to check the problems within a circuit.

Jumper wires are available in many color and sizes depending on for what are you using. In breadboards, jump wires are used to build a connections between the two micro controller points and other devices such as buttons and sensors.

If it is possible, then jumper wire can be placed on the component side of a circuit board during assembly time. The wires should be routed in the X-Y manner. Jump wires should not be raised more than 1/8 inch above the surface of any circuit board.

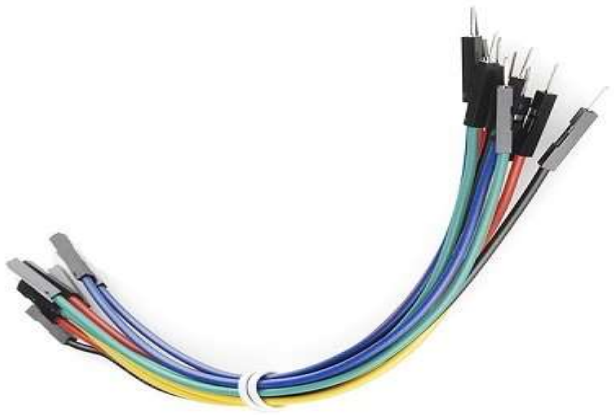


Fig 2.17: Stranded 22AWG jump wires with solid tips.

A jump wire have another name like jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable. It is a group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnection of components on a breadboard or other test circuit board, internally or with other equipment or components, without soldering.

Individual jump wires are connected by inserting their end connectors pin into the slots in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Types of wires used in our Project:

- (Male to Male) M to M wires
- (Male to Female) M to F wires
- (Female to Female) F to F wires

Fig 2.18: Types of Jumper wires

Types of Jumper Wires:

There are different types of jumper wires. Some are similar to this electrical connector at both ends, while others have different connectors. Some common connectors are:

- Solid tips
- Crocodile clips
- Banana connecters
- Register Jack (RJ)
- RCA connectors
- RF connectors

Refrence:

- https://www.arduino.cc/
- https://www.makerfabs.com/L298-Dual-H-Bridge-Motor-Driver.html
- http://microcontrollerslab.com/ultrasonic-sensor-working-applications-advantages/

Arduino IDE:

The Arduino integrated development environment (IDE) is cross platform application for Windows, Mac OS, Linux) that is written in the programming language Java. It is usually used to write and upload programs to Arduino UNO.

The source code for the IDE is released under the GNU General Public License means it is freely available. The Arduino IDE Software supports the languages like C and C++ using standard rules of code structure. The Arduino IDE supplies also a software library from the Writing code for project, which gives many common input and output procedures. User-written code require only two basic functions, for starting the main program loop, which are compiled and linked with a program stub *main()* method into an executive program with the GNU tool chain, it also included with the Arduino IDE distribution. The Arduino IDE programs are converting into the executable code and into a text file in hexadecimal encoding which are loaded into the Arduino board by a loader program in the board software.

Arduino IDE provides a good user interface and also comes with various programs which can be used for test run purpose. The programming in this interface is very simple and can be programmed easily the Arduino program has two methods which it runs by default named as void setup() and void loop(). The void setup() consists of the initialization variables and runs first. The void loop() runs after setup() and it consists the program of expected output.

AMR Voice:

AMR stands for Android Meets Robots Voice Recognition. It uses android mobiles internal voice recognition like Google to pass voice commands to your robot. It is paired with Bluetooth Modules and sends in the recognized voice as a string for example (Hello) the android phone will return a string (Hello) to your Bluetooth module and indicate the start and stop bits.

Following are the platforms for using AMR Voice:

- Arduino
- ARM
- PICAXE
- MSP430
- 8051

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 PROBLEM DEFINITION

The Voice control system covers the various issues:

- User can search for anything using voice search.
- It saves times of the user.
- It is convenient way of searching.
- A person can implement voice search into various different things

Problems in existing system

It is said that human makes mistakes and the computer is always accurate. The system can have many errors with paper-based work. We can see some problems that are been covered in this project. The problems are been categorized into following problems:

- **General Problem**

General problems include that basic systems and data organization in the system for existing system. Some of them are following

- Requires more time for searching old files and details of the products.
- Requires a lot of paper containing invoice for the product.
- Difficult to search for too long sentences.

- **Technical Problem**

The following are some technical issues with existing Voice Controlling system:

- Sometimes the server may be down due to some wires getting damage.
- Host might be unreachable due to some technical issues
- Less Secured
- Takes time to understand what's being said.
- Wires may get damaged by rats or other animals.

- **Programming errors**

There are some programming errors in the existing Voice control system:

- Sometimes there can be an error in the program that can cause error in the output.
- Using an outdated Algorithm.
- Maintaining the flow of the program
- Using very big program can take time for the system to read and execute it.

All the above issues are covered in this system:

- This system is working on a microcontroller, so there are no chances of calculation errors if the program is correct.
- The data is stored in the microcontroller itself so no security issue of manipulating the program once it is written
- It is more secure and can be trusted.
- Program is very short which reduces the response time.
- It is easy to use as small children can also use it.
- The apk that is used has a very user-friendly GUI

Limitations of the Project:

There are some limitations in this project that must be taken into consideration while performing the project as the project is based on Internet of things it requires internet to perform anything and a Bluetooth connection as well without which it won't work. It requires internet to understand what the user is saying and converts that words into a string which can be understood by the program easily as it has been already been implemented in the program. The project shows more of its background in the field of Internet of things which in itself is a very good topic which deals with the changes in the things that can be done when they are connected to the internet so having internet connection is very important in the project.

In India urban areas are seen to be more developed and seemed connected to other countries compared to rural areas so the technology there compared to urban areas is very different. When one visits rural area with an urban origin he can face a lot of problems as he could not connect to the internet which now a days has become an important part of living. So, bringing these things to rural areas always make the people their amaze and the start questioning as they are more curious about how it happened. It always feels great to visit such places and suggest them with new ideas and project which make them grow faster and match up with the urban cities.

3.2 REQUIREMENTS SPECIFICATION

The voice control car is based on embedded system which is combination of hardware and software both. Here the **Hardware components** are Arduino UNO, Bluetooth module HC-05, Motor Driver L298N, Bo Motor, Tires, ultrasonic sensor, servo motor, batteries. All of these are hardware components and software which is used in for this project is Arduino IDE (1.0.6V). Arduino is a very minute part of embedded system, it is application product of embedded system. Arduino is just like any other microcontroller board, with a specially designed API and software which makes easy to program.

Arduino Uno is used because we can mount the motor driver in it. It is easily available and also cheap, and yes easy to use and acquire less space as we have to place everything (other devices) on the car chassis.

The Motor Drive L298N, which is a dual full-bridge driver. This is used to drive motors. It lets you drive four DC motors with your Arduino. We can control the speed and direction of the car.

The HC-05 Bluetooth Module is a class-2 Bluetooth module with Serial Port Profile. which is configured as either Master or Slave.Wheels which are used with dimensions are 7 cm in diameter and 3 cm width of wheel.

Bo-motor is a single shaft DC motor with gear box which gives good torque and rpm at lower voltages. This motor can run at low rpm when driven by a 9v battery. It is most suitable for light weight robot car runs on small amount of voltage. Out of its shafts can be connected to wheel.

Ultrasonic Sensor SRF04 this sensor is based on measuring the properties with the help of sound waves with frequency above the human audible range. This SRF04 sensor provides 2cm to 400cm non-contact measurement function range.

Micro Servo motor SG90 It is tiny and lightweight with high output power. This servo motor can rotate 180 degrees, and works just like the standard kinds but in smaller. With any servo code, hardware or library to control these servos. A lot of auto manufacturers now supplies this component.

For the programming purpose it requires its own Integrated development environment (IDE). The code is written in Arduino IDE and the code is upload by connecting a USB cable to the Arduino Board and another end to the CPU where the actual code is compiled.

For running this project an apk is required called as AMR_Voice it is connected with the Bluetooth and it is used as a part of communication with the robot.

3.3 PLANNING AND SCHEDULING

Gantt Chart:

Gantt charts are useful for planning and scheduling purpose. The Gantt chart gives a clear overview of all the activities, and which activities will actually be started on what Date and Ends on what Date. Gantt Chart helps to lead in this project. The Activity in Gantt chat are shown in horizontal bar that represents the project activity plan. The following Gantt chart shows the activities that are planned for developing this project. It shows each activity time schedule, activity start date and completion date.

The following Gantt chart is prepared in Microsoft Excel 2016.

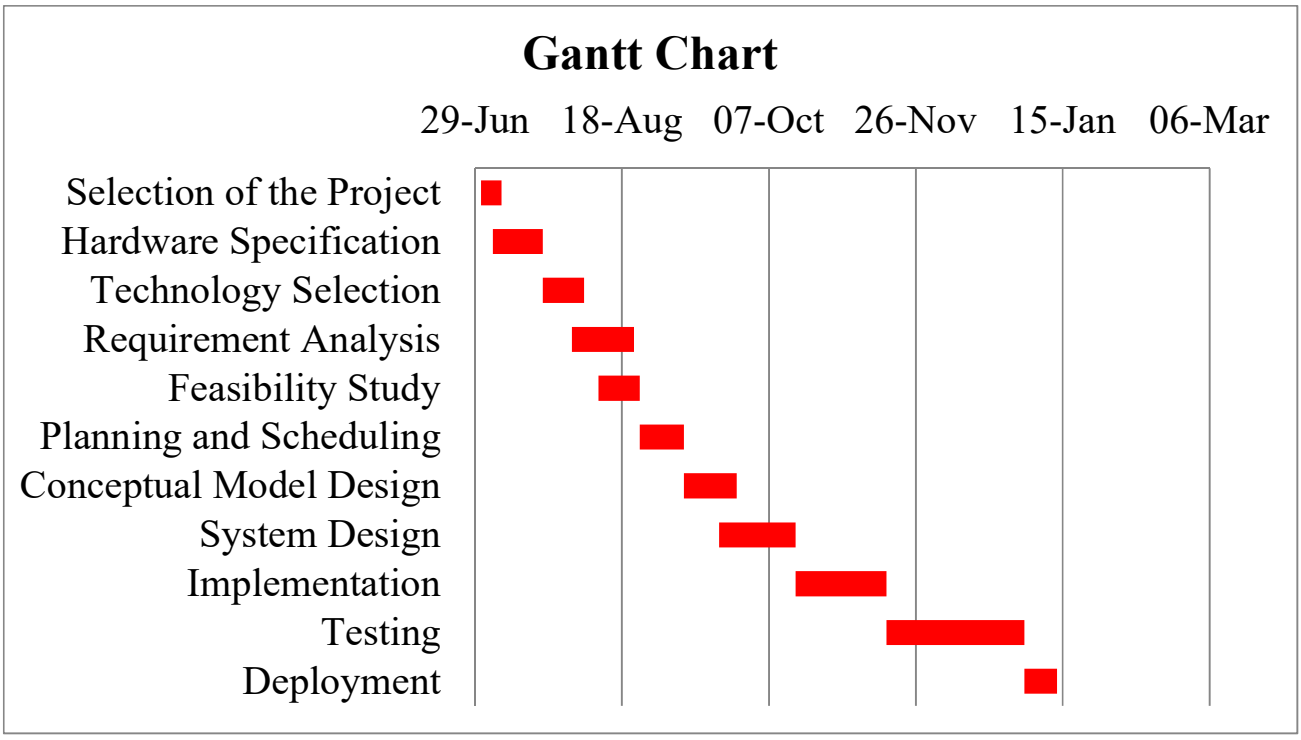


Fig 3.1: Gantt Chart

This project consists of activities given below:

Selection of the Project, Hardware Specification, Requirement Analysis, Feasibility Study, Planning and Scheduling, Conceptual Model Design, System Design, Implementation, Testing, Deployment.

PERT CHART:

Pert charts are useful for planning and scheduling projects. The Pert chart give the clear picture when activities will actually take place and highlights which one will be executed at the same time. Activity network can be misleading in this project. The Pert chat is a horizontal bar that represents the project plan visually. The following Pert chart shows the activities that are planned for developing online shopping system. It shows each activity time schedule, start date, activity completion date.

The following Pert chart is prepared from the website link given below

https://online.visual-paradigm.com/features/pert-chart-tool/.

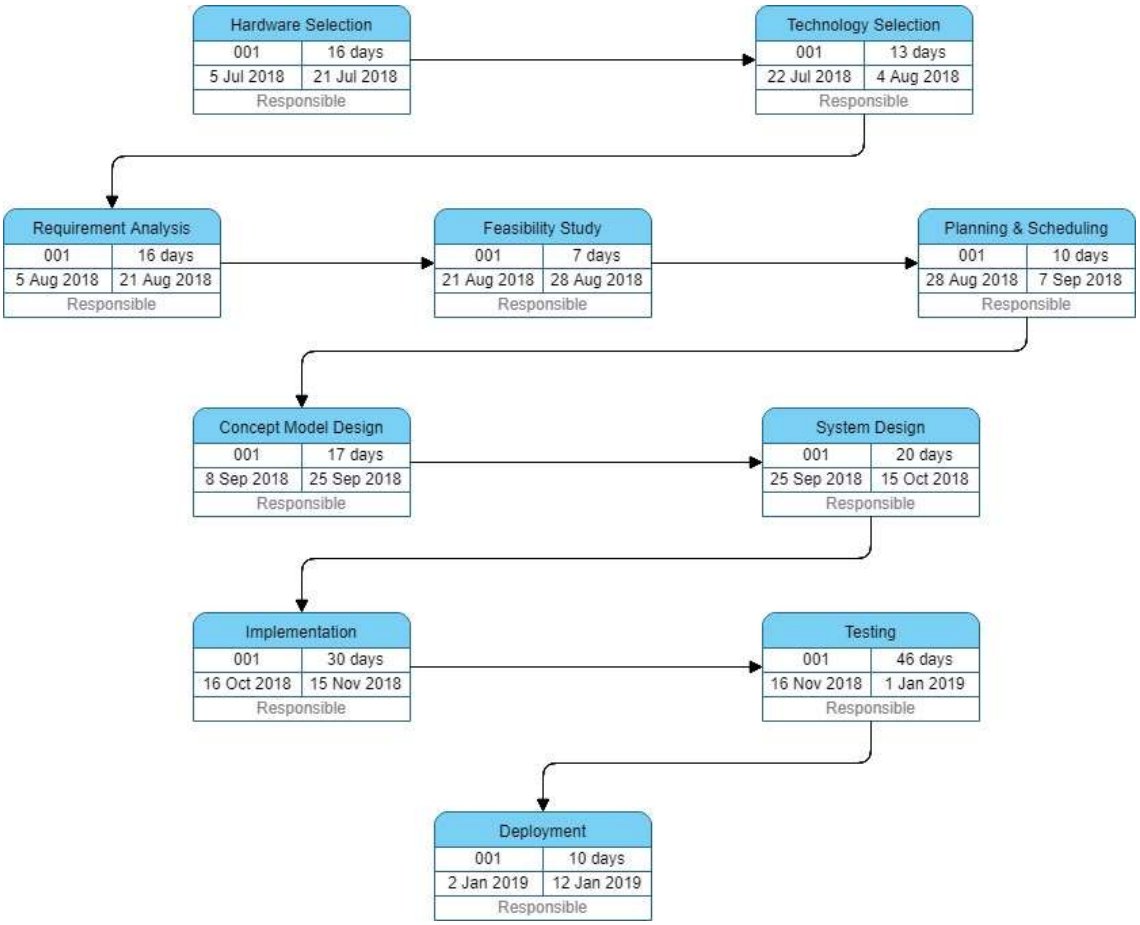


Fig 3.2: PERT Chart

3.4 SOFTWARE AND HARDWARE REQUIREMENTS

3.4.1 Hardware Requirements

The voice control car is based on embedded system which is combination of hardware and software both.

Hardware components are

- Arduino UNO
- Bluetooth module HC-05
- Motor Driver L298N
- Bo Motor
- Ultrasonic Sensors
- Servo Motor
- 9V batteries
- Tires & Wires

All of these are hardware components and software which is used in for this project is Arduino IDE (1.0.6V). Arduino is a very minute part of embedded system, it is application product of embedded system. Arduino is just like any other microcontroller board, with a specially designed API and software which makes easy to program. the logo of Arduino is look like given in below picture

Fig 3.3: Arduino Logo

Arduino is a board, which is combination of microcontroller called AVR. AVR microcontroller is made by Atmel since 1996. Atmel is the extra circuitry needed to make it operate, and a simple programming API built over C++, with a simple way to upload programs via USB bridge.

In AVR microcontroller we can easily program a code with the help of languages like C and C++. these languages are easy to understand by the user which makes it easier to code a program in microcontroller.

Fig 3.4: AVR Microcontroller

Voice controlled car is controlled via **Bluetooth HC-05** through a Smartphone app called AMR_Voice which is available on the internet for free. This smartphone app is developed in such a way that it listens to the voice and first sends request to Google to Convert that voice to a String so that it can be recognized by the program easily.

The Bluetooth connected to the Arduino board where Bluetooth receives signal from smartphone and transmits that signal to Arduino board from the Android app as characters, and stores them as string to the assigned string. There are words pre-programmed (go, back, right, left, round and stop) to the Arduino: whenever the received text matches with the pre-programmed words, the Arduino executes the command that is assigned to the command. Arduino can connect to a laptop to monitor serial communication and check the working process and the words received by the Bluetooth.

This Arduino UNO is an micro-controller board and it is based on ATmega328. To use this Arduino board both things are required a physical board (hardware) and Integrated Development Environment which is IDE (software). The Arduino UNO is and physical board and software which is required that is Arduino IDE(1.0.6V). it means Arduino have its own software. which is use to write a code for Arduino UNO. It works like we first write a code on Arduino IDE(1.0.6V) software which is installed on our system and upload that code in Arduino UNO through the cable bridge. This Arduino UNO is responsible for movement of the car with supportive hardware component Bluetooth module, motor driver, motor, tire, etc. And the second thing is Bluetooth device which named as Bluetooth module hc_05 this help to form a connection between Arduino UNO and Android Phone (Smartphone). It is used to form a wireless connectivity. So, it can be commanded wirelessly and it accepts that command given by user through the android phone and transmits the signal to Arduino. This Bluetooth module hc_05 is pre-programmed no need to require any coding experts for it. This Bluetooth module hc_05 is a master and slave modules

The **L298N** is a motor driver and it is integrated monolithic circuit in which 15-lead Multi watt and PowerSO20 packages are there. This is use for from connection between motor which use to rotate tire and Arduino board. This motor driver has 4 output pins where motor is connected. And this motor driver can also be called as little current amplifier taking low current and converting to high current. which means this L298N motor driver takes low current signal and convert it into high level current signal which drives the motor. 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.

Other than L298N there are also other many different types of motor drivers which are available in the market like L293D, ULN2003, L298, etc. But compared to all of these L298N seemed to be much better, smaller & is easy to use.

To give power supply in our projects we are using components are 9V batteries, jumper wires (male to male, female to male, female to female). Total of three batteries would be used two for supplying power to the motors as they require more power supply and one battery for Arduino is enough as it equally distributes power to each equipment's requirement. Arduino has 2 pins for grounding and 2 different power supply option one is 3.3V and another is 5V.

Ultrasonic sensor is used to prevent the crashing of he car from any obstacles which comes in front of it. Whenever an object comes in front of it the car automatically senses it and stops the car. This ultrasonic sensor is placed on a servo motor which can rotate 180^{o} here it is programmed in such a way that when it receives the command to move right the servo rotates 90^{o} right to check for any obstacles and then moves and it does the same for left command it turns to 90^{o} left and checks for obstacles.

All these components are assembled in a cardboard structure made by a strong and light-weight cardboard so that it can reduce the load on the structure and also it won't affect the speed of the car. The batteries used are of 9V and durable since they are not rechargeable once they are totally used they are of no use and we need to throw them. As an alternative we can buy rechargeable batteries which can be recharged after its battery is fully used. And all the connections made in these components are connected by using Male to Male and Male to Female wires. The motor that are connected with the tyres are called bo motors these are DC motors which are connected to the motor drivers. The speed of these motors can be controlled by the motor driver and programming. The commands are received by the Arduino board from a Bluetooth device connected to it called as HC-05 module.

3.4.2 Software Requirements

Software required for the Project:

- Arduino IDE version 1.0.6V
- AMR_Voice apk

Arduino IDE version 1.0.6V:

The Arduino IDE (1.0.6V) software is open source, so it is freely available on the internet. There is other software (programming tools) also available on internet like makefiles and AVR studio. But Arduino provides its own software for free, so no need to download any other software for programming.

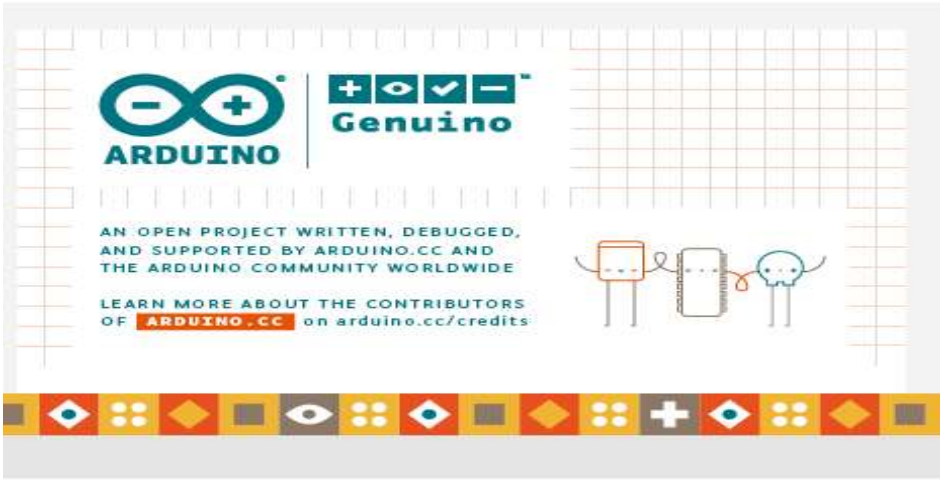


Fig 3.5: Arduino IDE

API stands for Application Programming Interface. it is the way to communication among the various different components. It just provides a GUI (graphical user interface) and design of Application Programming Interface (API) is impact on its usage. Basically, an API specifies how software components should interact with hardware. APIs are used for programming graphical user interface (GUI) components.

API is a software that allows two applications to talk(communicate) to each other. In other words, an API is the messenger that delivers your request and then delivers the response back to you. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together means combine those blocks. these API are available for or may be used in web-based system, operating system, database system, computer hardware, or software library.

- **AMR_Voice apk:**

AMR_Voice this apk is used to give commands to the Arduino board. This apk is freely available it is available on: https://amr-voice.en.aptoide.com/. There are various other apk which are available but they don't have as good user interface as this apk does have

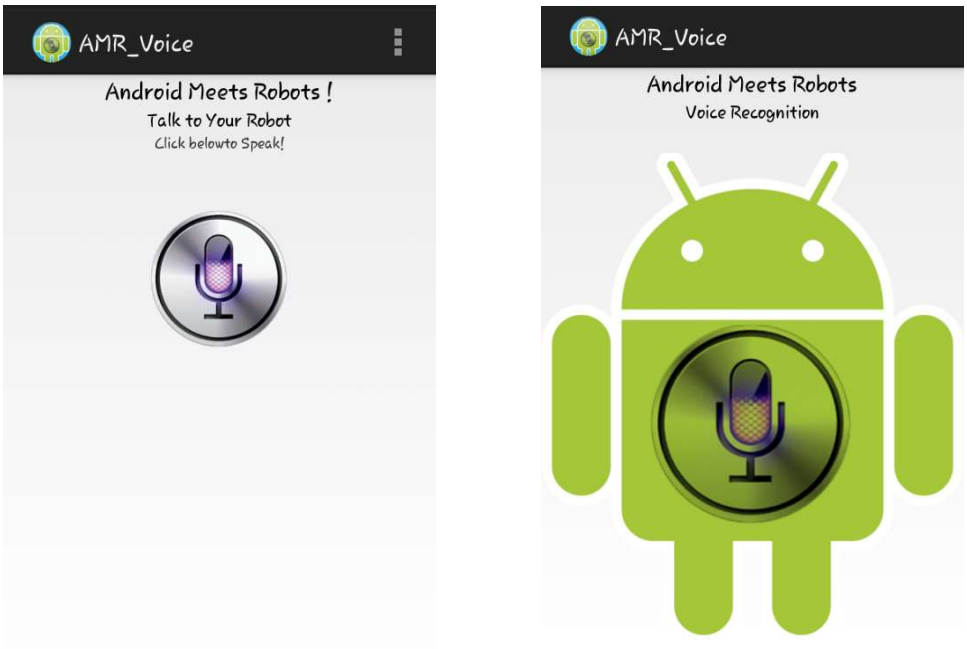


Fig 3.6: AMR_Voice apk

When this apk starts it automatically starts Bluetooth and asks the user to connect the robot using Bluetooth for that we need to first power the Bluetooth HC-05 and after that when we scan for Bluetooth we can see HC-05 and then we connect to it.

This apk requires a voice recognizing software it uses google as a default apk as it is almost present in all android devices. When we click on the mic logo it will start taking our voice command when it receives the voice command it then uses internet and translates this voice command into a string using google translator and then it is sent to the Arduino.

This is how this apk works.

3.5 PRELIMINARY PRODUCT DESCRIPTION

Functions:

1. It listens to the voice commands from the user and runs the car accordingly.
2. Anyone can give voice command as voice recognition is not added.
3. Sensors detect objects around and react accordingly.
4. Bluetooth is used to receive commands from user.
5. Sensors are placed on Servo motor so that they can check for objects before the car turns.
6. The structure gives the car speed and utilizes less electricity.
7. Sensors prevent car from crashing.
8. Sensor check for any obstacle using an echo transmitter and a receiver so it does not depend on any color and works at night as well

Operations:

1. Accepting voice command from the user.
2. Converting the voice command into string using google translator tool.
3. Sending this String to the robot using Bluetooth module.
4. Checking the program for the related string in the program.
5. Executing the program in the related string.
6. Sensors checks for obstacles ahead and controls the car from crashing.

3.6 CONCEPTUAL MODELS

Use case Diagram:

A use case diagram is a simple representation of the user's interaction with the system and shows how the relationship between the user and also other different use cases in which the user is involved. Use case diagram can identify other different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

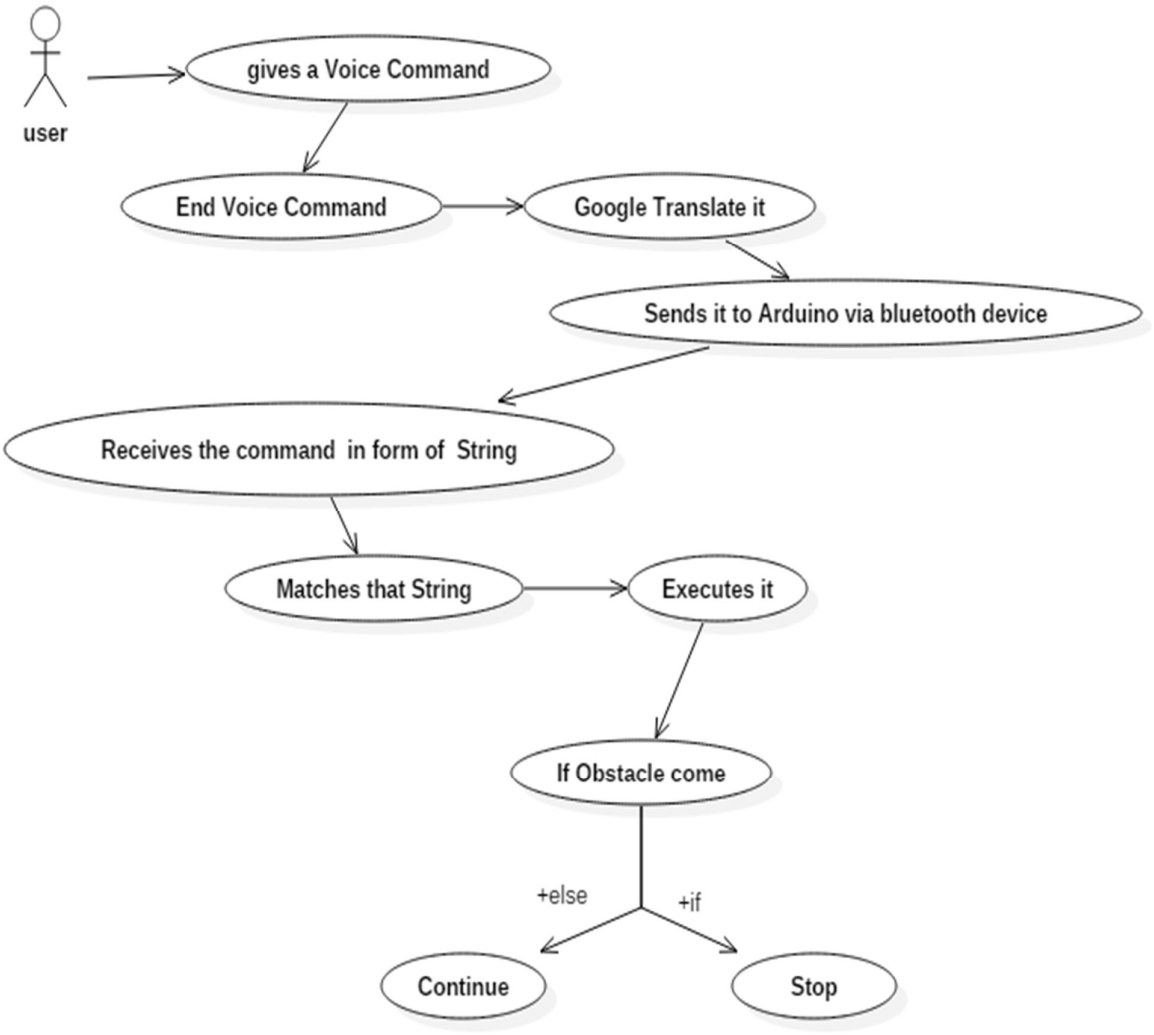


Fig 3.7: Use case Diagram

Activity Diagram:

Activity diagram is an important diagram in UML that describes the dynamic aspects of the working system. Activity diagram is basically used as a flowchart to represent the flow of one activity to another activity. This activity can also be described as an operation of the system.

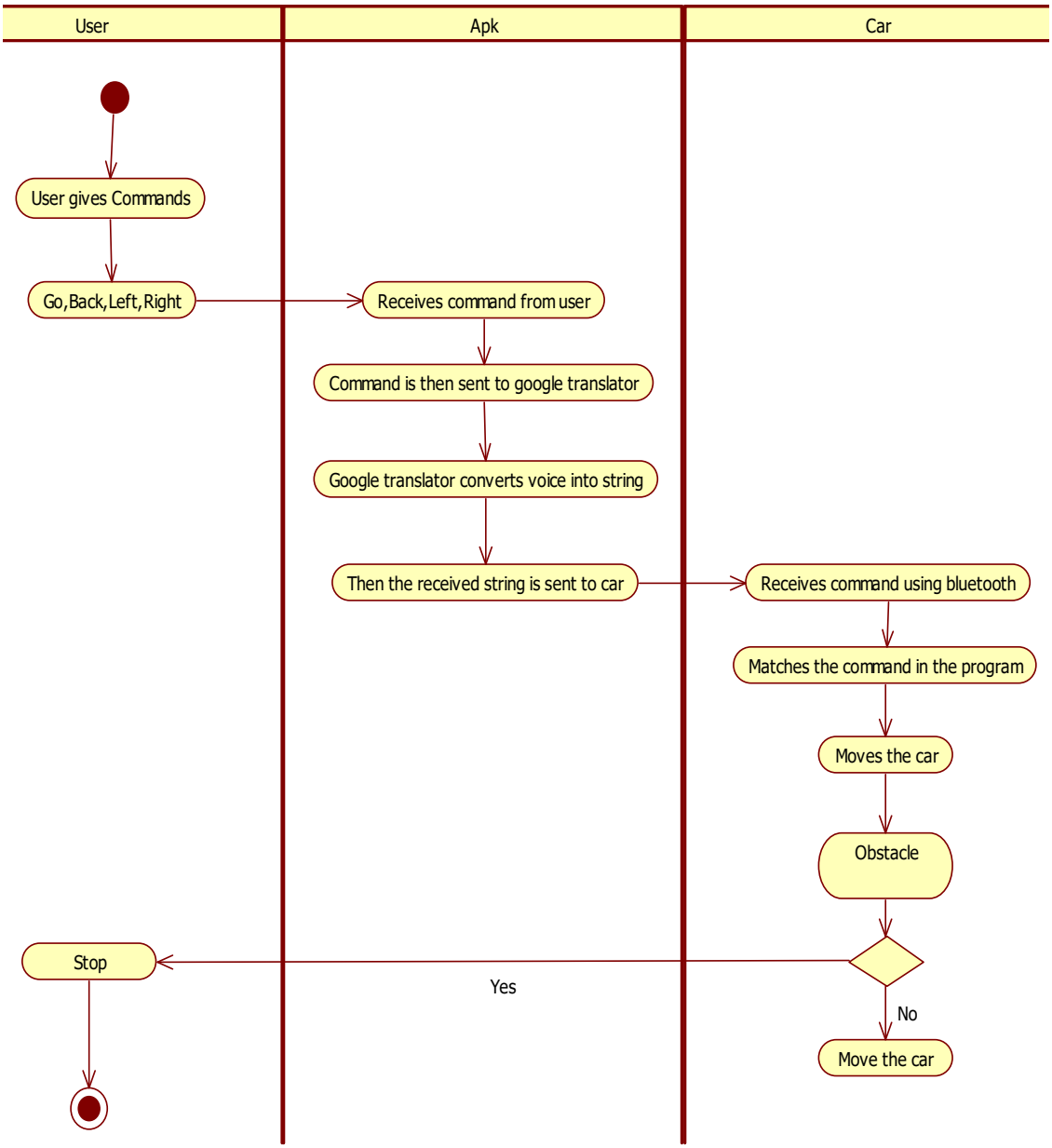


Fig 3.7: Activity Diagram

CHAPTER 4: SYSTEM DESIGN

System Design is basically an overview of how the system will be how the car will actually function. The algorithm used in this particular project and the flow of the program. This chapter will also give an overview of the program's module in flowchart. The data structure used in the program. It will also consist of a circuit diagram which will state all the connections made in this project.

4.1 BASIC MODULES

- **User Module**

  User in this module is any person who can give voice commands to the car to make that car move. He basically can control the car.

  - User gives voice command through the apk.
  - The apk uses an in-built voice recognizer to get the voice command.
  - The apk usually uses google voice to get command.
  - It then let's google translator to convert that voice to string.
  - The conversion process requires internet as its not an in-built feature.
  - After the apk gets the voice command in string format and it then passes it to the car.
  - The command is passed using the Bluetooth connection.
  -

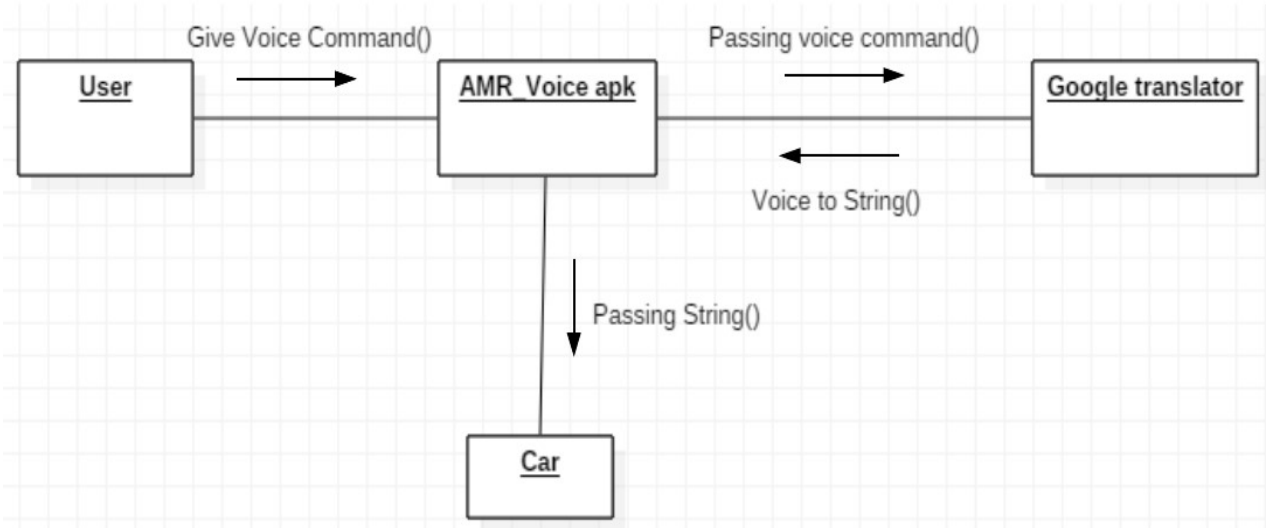


Fig 4.1: User Module

- **Car Module**

  Car Module is the working of the car that how it handles a voice commands and shows the output to the user by moving it according to the command.

  - The Arduino receives the string from the apk using a Bluetooth module connected to it called HC-05.
  - After receiving the string, it compares the string in the program.
  - It then executes the required the satisfied condition as per the voice command from the user.
  - The sensors are also used to avoid the obstacles that will come in between.
  - The sensors always keeps on sending and receiving signals if an obstacle is present or not if any it stops the car.

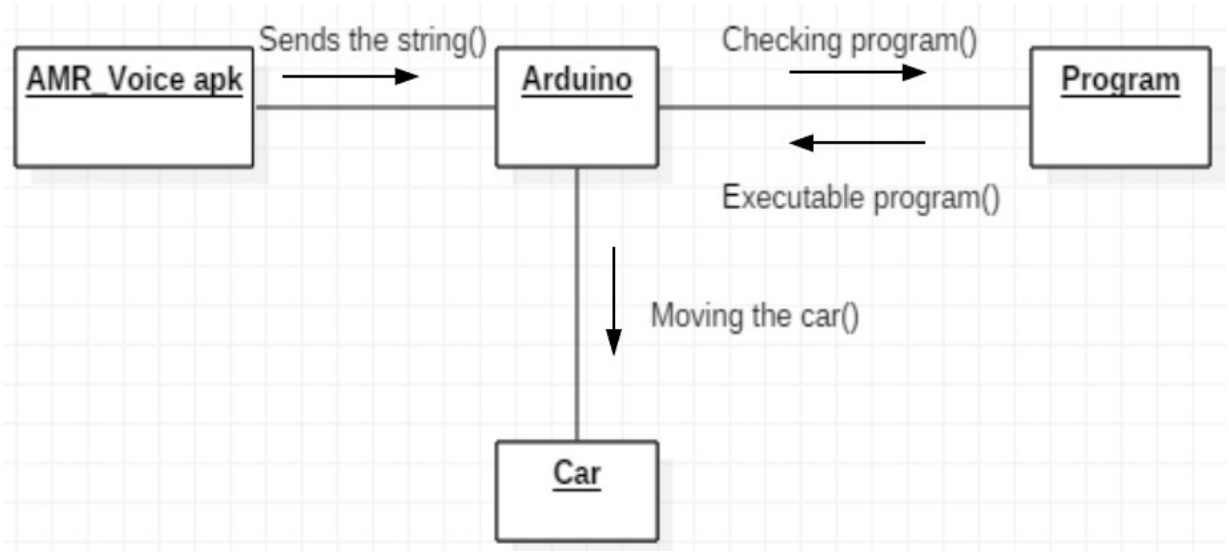


Fig 4.2: Car Module

4.2 DATA DESIGN

This is how the program is designed for the project just used for prototype:

```
#include <SoftwareSerial.h>
SoftwareSerial BT(0, 1); //TX, RX respetively just in arivu
String readvoice;
int In1=7;
int In2=8;
int In3=9;
int In4=10;
int ENA1=5;
int ENA2=6;
int SPEED=255;
int trigPin = 11;
int echoPin = 12;
void setup()
{
  BT.begin(9600);
  Serial.begin(9600);
     pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(In1,OUTPUT);
  pinMode(In2,OUTPUT);
  pinMode(In3,OUTPUT);
  pinMode(In4,OUTPUT);
  pinMode(ENA1,OUTPUT);
  pinMode(ENA2,OUTPUT);
}
void loop()
{
  long duration,distance;
    digitalWrite(trigPin,HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW);
```

```
duration=pulseIn(echoPin, HIGH);
distance =(duration/2)/29.1;
Serial.print(distance);
Serial.println("CM");
delay(10);
analogWrite(ENA1,SPEED);
analogWrite(ENA2,SPEED);
while (BT.available())
{
delay(10); //Delay added to make thing stable
char c = BT.read(); //Conduct a serial read
readvoice += c; //build the string- "forward", "reverse", "left" and "right"
} if(distance<=10)
  {
    digitalWrite (In1, LOW);
  digitalWrite (In2, LOW);
  digitalWrite (In3, LOW);
  digitalWrite (In4, LOW);
  delay (100);
  digitalWrite (In2, HIGH);
  digitalWrite (In4, HIGH);
  delay(500); }
if (readvoice.length() > 0)
{
if(readvoice == "*go#")
{
  digitalWrite(In1,HIGH);
  digitalWrite(In2,LOW);
  digitalWrite(In3,HIGH);
  digitalWrite(In4,LOW);
  delay(100); }
else if(readvoice == "*back#")
{
  digitalWrite (In1, LOW);
```

```
  digitalWrite (In2, HIGH);
  digitalWrite (In3, LOW);
  digitalWrite (In4, HIGH);
  delay (100);
}else if (readvoice == "*left#")
{
  digitalWrite (In1, LOW);
  digitalWrite (In2, HIGH);
  digitalWrite (In3, HIGH);
  digitalWrite (In4, LOW);
  delay (1100);
  digitalWrite (In2, LOW);
  digitalWrite (In3, LOW);
  delay (100);
  }else if (readvoice == "*right#")
{
  digitalWrite (In1, HIGH);
  digitalWrite (In2, LOW);
  digitalWrite (In3, LOW);
  digitalWrite (In4, HIGH);
  delay (1100);
  digitalWrite (In1, LOW);
  digitalWrite (In4, LOW);
  delay (100);
}else if (readvoice == "*stop#")
{
  digitalWrite (In1, LOW);
  digitalWrite (In2, LOW);
  digitalWrite (In3, LOW);
  digitalWrite (In4, LOW);
  delay (100);
}  readvoice="";  } }
```

4.2.1 Circuit Design

A circuit diagram (electrical diagram) is a graphical representation of any electrical circuit. A graphical circuit diagram uses simple images of components, while a schematic diagram shows the components and interaction of the circuit using standard symbolic representation. In the schematic diagram, the presentation of the interaction between the circuit components is not compatible with the physical system in the device.

Connections:

In this project we have shown the circuitry diagram of our project means circuit diagram for Voice Control Car. And we have shown connection between those components which we have used in this project. All the components connection description are given below for our project. Arduino UNO have total pin in which some pins connected the other components.

The pin which is connected to other components is:

The Arduino UNO is connected to Bluetooth module, motor driver, servo motor, and ultrasonic sensor.

The Arduino has more than thirteen pins which are used for connection in this project.

But pins which we are using are given below:

- VCC
- GND
- PB4/MISO
- PB3/MOSI/OC2A
- PB/SS/OC1B
- PB1/OC1A
- PB0/ICP1/CLKO
- PD7/AIN1
- PD6/AIN0
- PD3/T1
- PD1/TXD
- PD0/RXD

The pins in Bluetooth are VCC, GND, RXD, and TXD. VCC is connected to VCC of Arduino. And GND in connected to GND of Arduino, the RXD is connected to TXD and TXD is connected to RXD of Arduino.

Ultrasonic sensor has four pin which are +5V, GND, Trigger, and Echo. +5V is connected to VCC and GND is connected to GND of Arduino. Trigger pin is connected to PB4/MISO pin of Arduino and Echo pin is connected to PB3/MOSI/OC2A pin.

Pins in ultrasonic sensor:

- VCC for +5VDC
- Trig for Trigger (INPUT)
- Echo for Echo (OUTPUT)
- GND for Grounding extra current

Servo motor SG90 have three pin which is VCC, GND, PWM. Same as above VCC is connected to VCC and GND is connected to GND. PWM pin is connected to PD4/T0/XCK pin.

Motor Driver L298N has input pins as well as output pins. Pins which are connected to Arduino are ENA, IN1, IN2, IN3, IN4, and ENB. ENA and ENB pin is connected to PD6/AIN0 and PD3/T1 of Arduino. Others pins like IN1, IN2, IN3, IN4 are connected to pins called are PB/SS/OC1B, PB1/OC1A, PB0/ICP1/CLKO, and PD7/AIN1. Motor driver have also four pins giving output to any component.

Bo Motor have positive and negative pin which is connected to output pins of Motor Driver L298N. this motor are connected to output pin1, output pin2, output pin3, and output pin4.

For power supply Arduino has power input pin where batteries are connected to give power supply to the Car for movement.

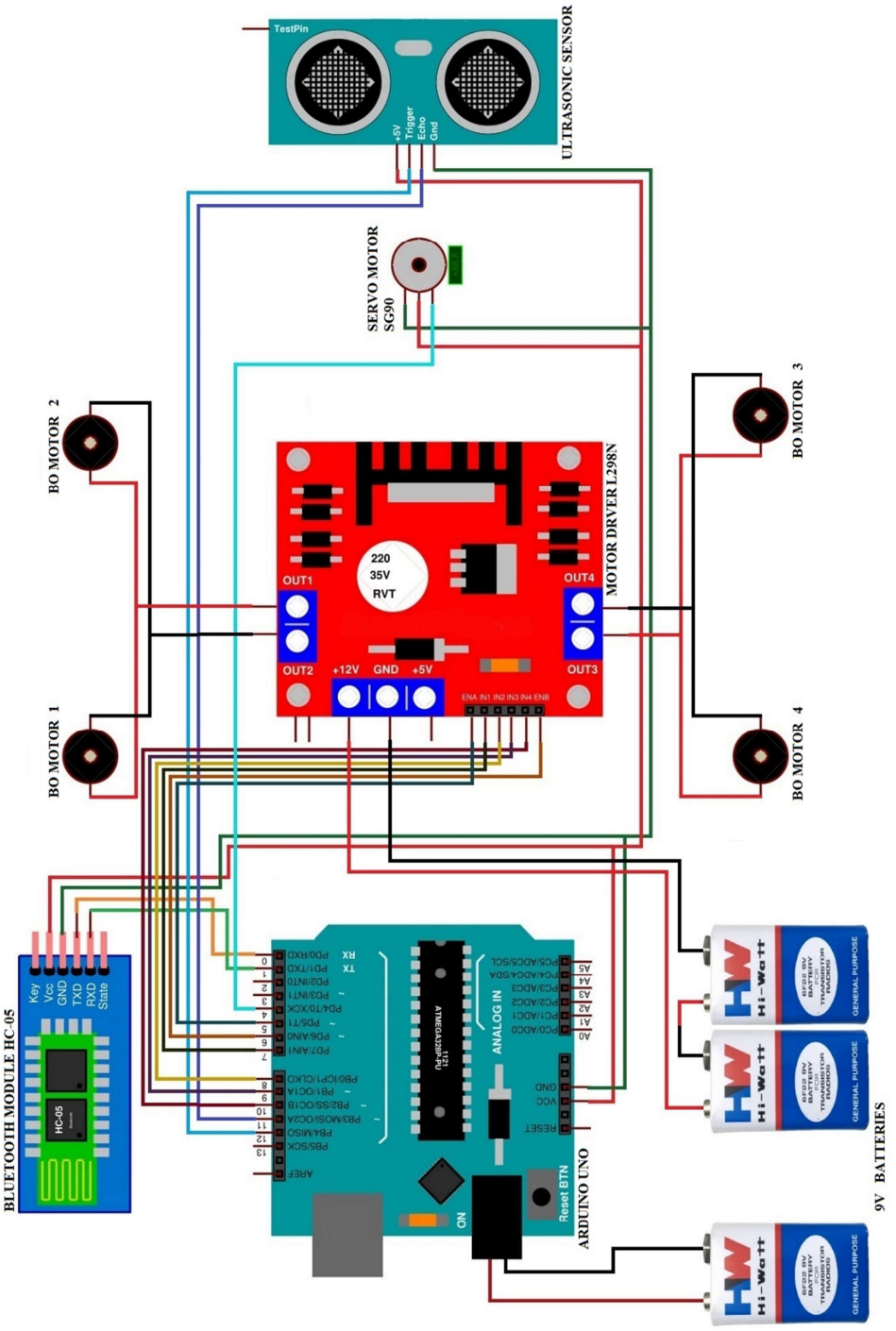


Fig 4.3: Circuit Diagram

4.3 PROCEDURAL DESIGN

Procedural design is classified as a computational come up to relying upon a set of instructions that, when used in a particular sequence, are the generators of form. While within this framework certain methods may be iterative and cyclical, procedural design often denotes the construction, conceptually, of a linear solver. The work which documented in this section, shows a important evolution of this approach. Intelligent systems are formed in which computation is given the freedom to absorb, interpret, and respond within the sequential set of procedures, thus shifting from linear logics to networked ones. This is address through papers that discuss the language from which such processes are enacted and explores the appearance of a built architecture through dynamic logics of design computation.

Traditionally, the procedural design is offered means of testing the relationships of parameters, but the work which is shown in this section are demonstrates a development of this approach. Procedural processes become an active agent for resolving the relationships of systems. In Gerber's "Multi-Agent System for Design" and Savov's utilization of gameplay, logics of fabrication shift from defining constraints to being exploratory agents for design ideation and the construction of architectural systems. Human transformation becomes an operational process in Johnson's work with SIFT algorithms and Sanchez's "Combinatorial Design". While injecting the transformational characteristic of human intuition, both of us take advantage of the repetitive aspect of the feedback mechanism to scan data caps on a large scale.

Ludwig von Bertalanffy established the sequencing of a feedback system as a part of General Systems Theory. It did basic work for structure of procedural design. In essence, it is a methodology that is used to test the relationships of parameters through iteration. Bertalanffy classified the components of a feedback system by count, species, and association. They have been exaggerated as the design space metric, because procedural operations allow the discovery, testing and finishing of ideal parametric relationships. In this application, the reaction system design is an active agent of exploration.

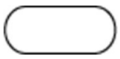
Refrence : http://papers.cumincad.org/data/works/att/acadia16_10.pdf

4.3.1 Logic Diagrams

A flowchart is a diagram which is used for representing an algorithm, workflow or process. Flowchart shows the steps and their order in the form of various boxes by adding boxes with arrows. This is explained in diagrammatic representation illustrates a solution model to a given problem. Flowchart Diagram is especially used for analyzing, designing, documenting or managing a process or program in various fields.

Symbols for flow Chart

- **Terminal (Terminator)**

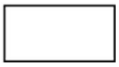


The terminator shows the start and the end of a process.

- **Data (Input Output)**

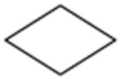
The Parallelogram represents the data object shows the input to and output from a process.

- **Process (Rectangle)**



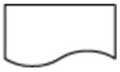
The Rectangle symbol is used to represent a process or an operation.

- **Decision (Conditional)**



The Diamond represents a decision. This shape is used to ask a question and it has two outputs, true or false and/or yes or no.

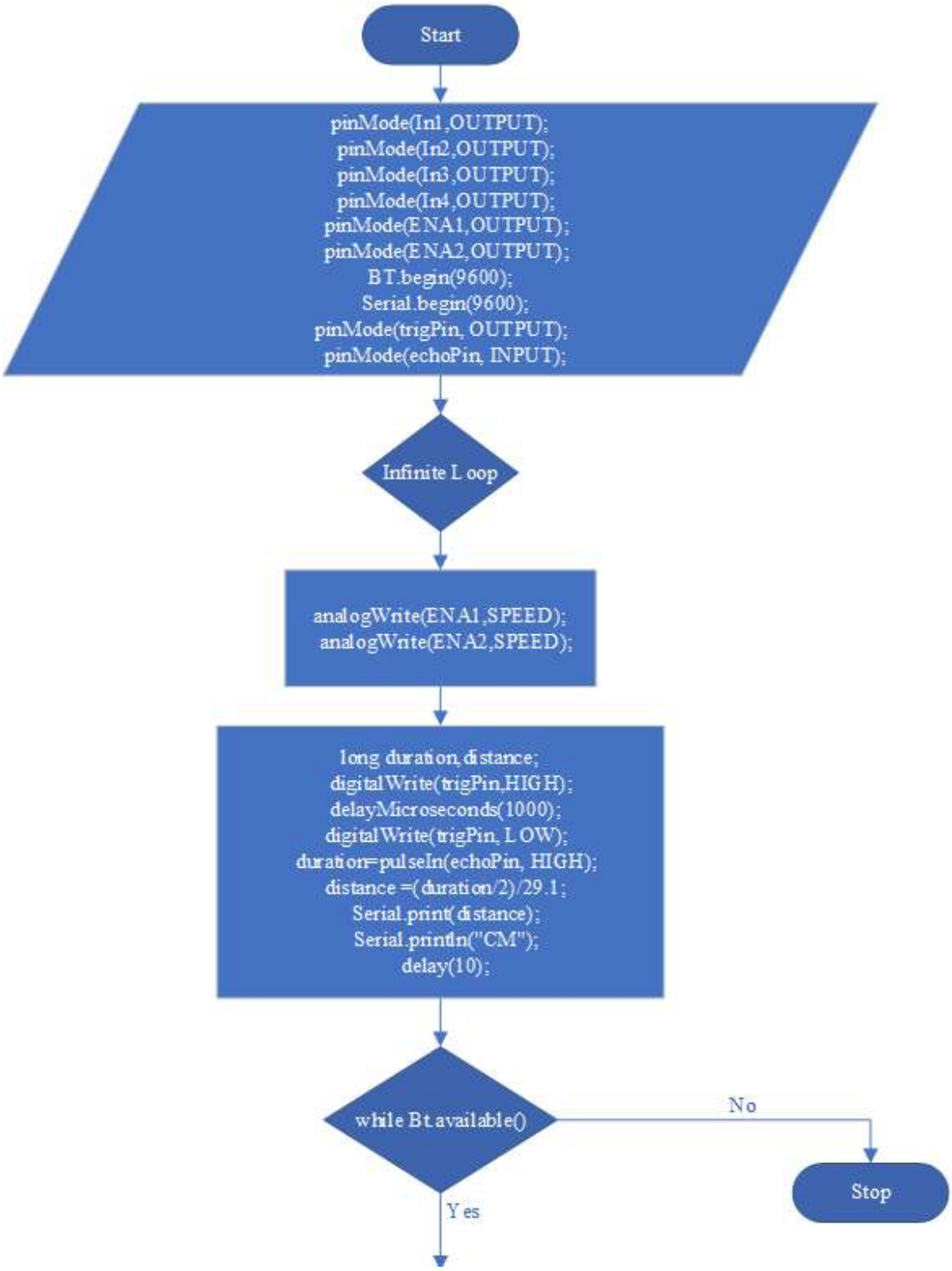
- **Document**

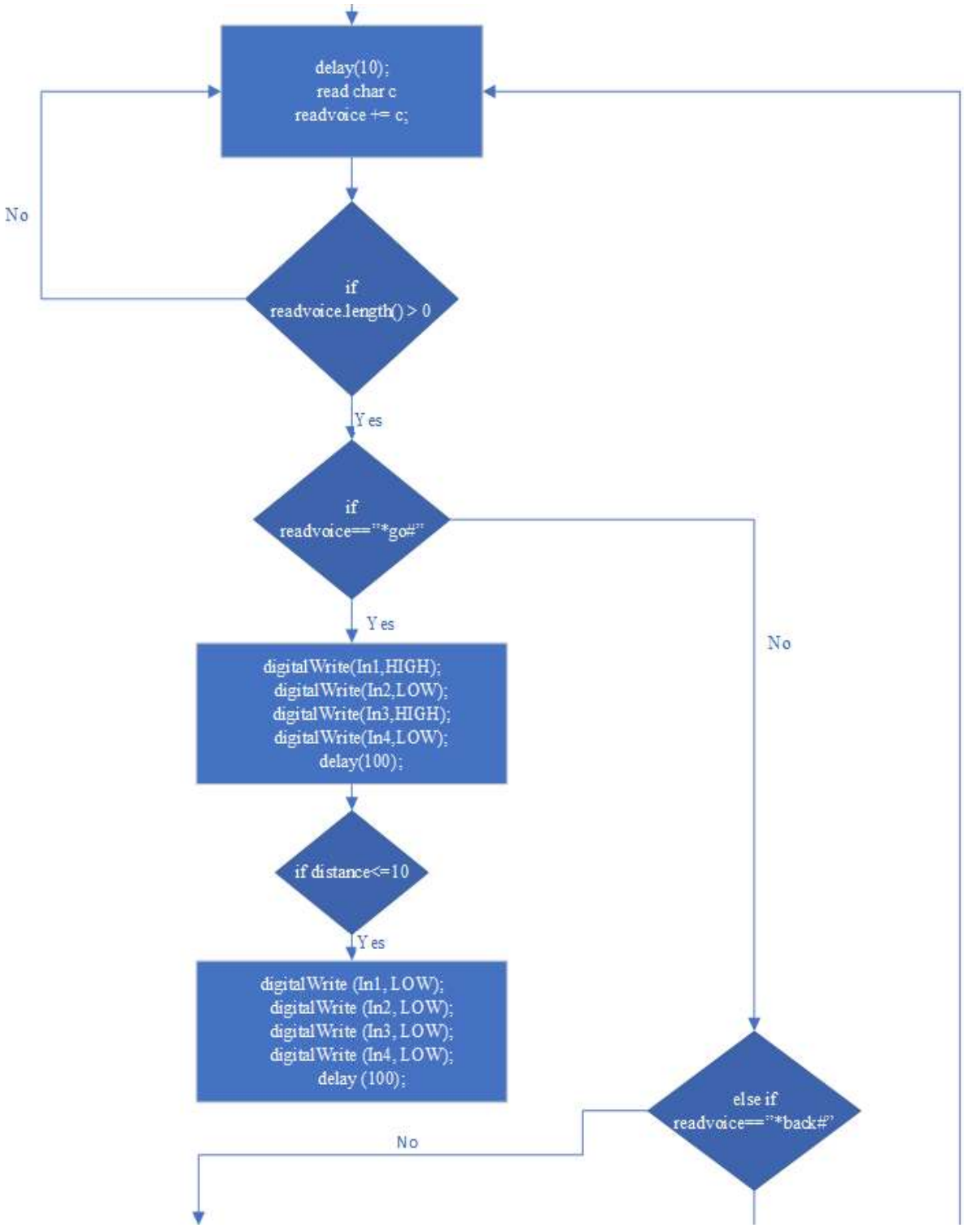


Document object is a rectangle with a wave shape in the base. It is used to represent a document in a process flow.

- **Subroutine (Predefined process)**

This means that the flowchart is already to be drawn for the predefined process, and you should reference the flowchart for more information.


Start
pinMode(In1,OUTPUT);
pinMode(In2,OUTPUT);
pinMode(In3,OUTPUT);
pinMode(In4,OUTPUT);
pinMode(ENA1,OUTPUT);
pinMode(ENA2,OUTPUT);
BT.begin(9600);
Serial.begin(9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Infinite Loop
analogWrite(ENA1,SPEED);
analogWrite(ENA2,SPEED);
long duration,distance;
digitalWrite(trigPin,HIGH);
delayMicroseconds(1000);
digitalWrite(trigPin, LOW);
duration=pulseIn(echoPin, HIGH);
distance =(duration/2)/29.1;
Serial.print(distance);
Serial.println("CM");
delay(10);
while Bt.available()
No
Stop
Yes


delay(10);
read char c
readvoice += c;
No
if
readvoice.length() > 0
Yes
if
readvoice=="*go#"
Yes
No
digitalWrite(In1,HIGH);
digitalWrite(In2,LOW);
digitalWrite(In3,HIGH);
digitalWrite(In4,LOW);
delay(100);
if distance<=10
Yes
digitalWrite (In1, LOW);
digitalWrite (In2, LOW);
digitalWrite (In3, LOW);
digitalWrite (In4, LOW);
delay (100);
else if
readvoice=="*back#"
No

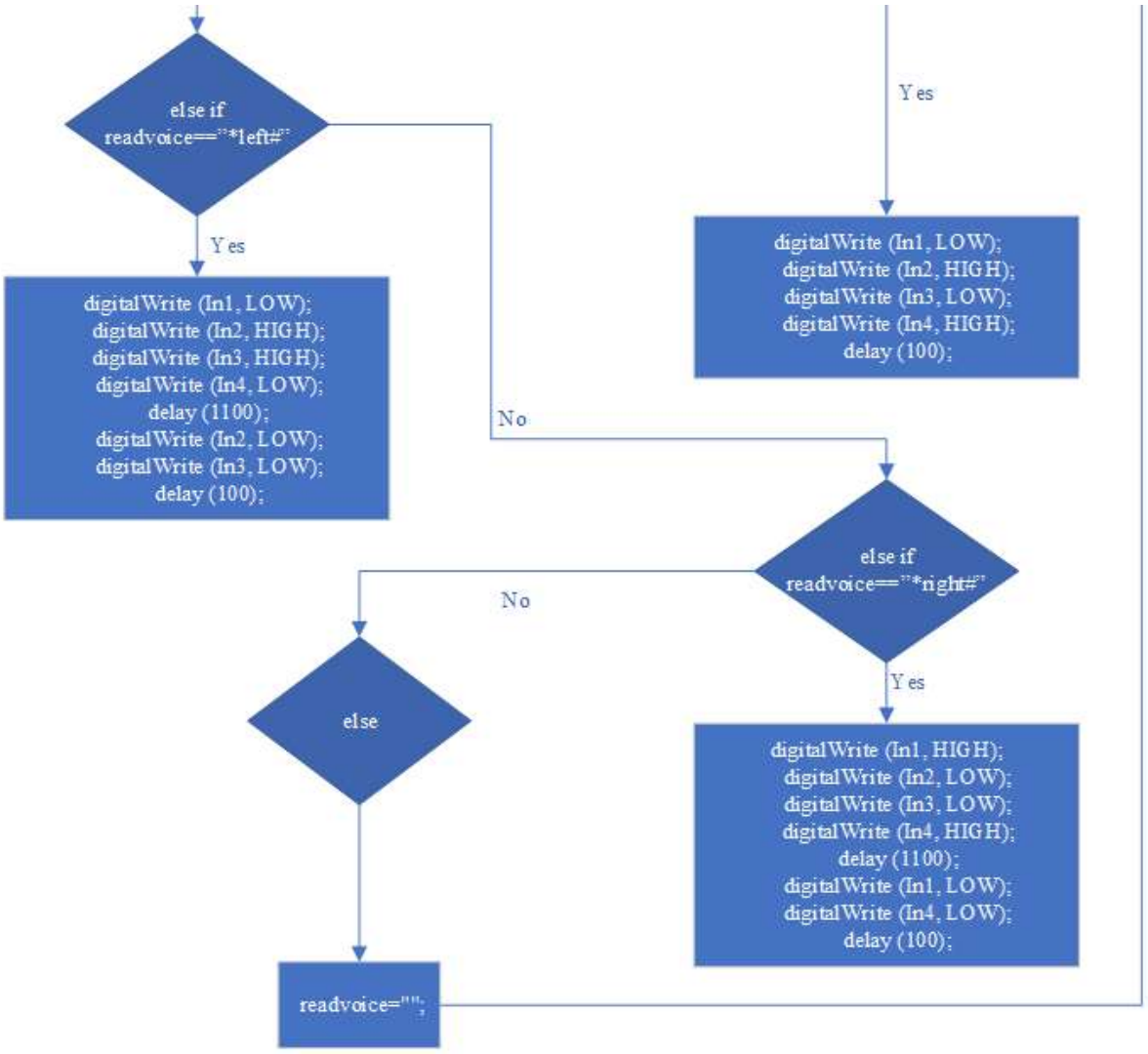


Fig 4.4: Data Flow Diagram

4.3.2 Data Structures

The Arduino software is open-source. Source code has been released for Java Environment under GPL and under the C / C ++ microcontroller library LGPL.

Structure:

Arduino programs can be divided into three main parts structure, value (variable and constant), and function. We will see about Arduino software programs, step-by-step, and writing programs without any syntax or compilation error.

Software structure consist of two main functions −

- Setup( ) function
- Loop( ) function

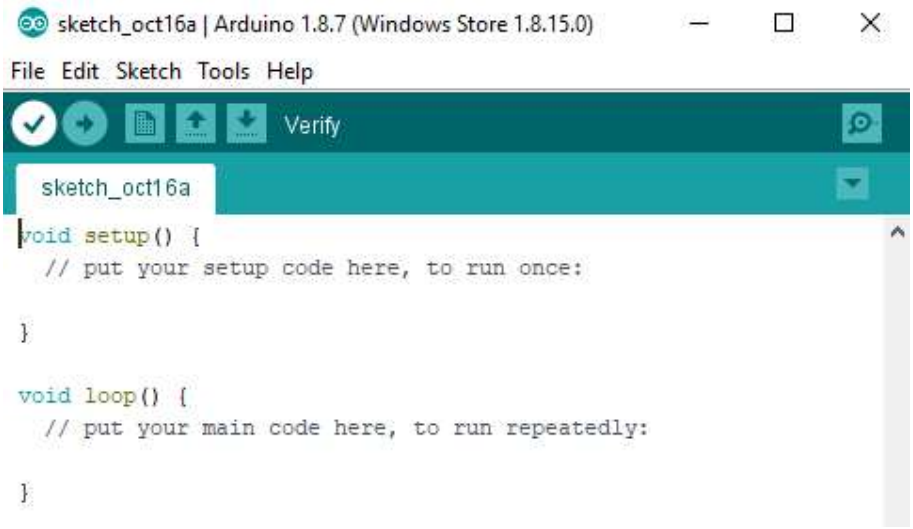


Fig 4.5: Sketch of Arduino

Void setup():

```
Void setup () {


}
```

- **PURPOSE** − When the sketch starts, the setup () function is called. Use it to start using variables, pin mode, libraries. After the reset of each power-up or Arduino board, the setup function will run only once.
- **INPUT** – Takes input
- **OUTPUT** – Provides an output
- **RETURN** – Return type

Void Loop():

```
Void Loop () {

}
```

- **PURPOSE** − After creating a setup () function, which initializes and sets the initial values, the loop () function correctly states its name, and continuously loops, to change its program and to get feedback. Use Arduino Board
- **INPUT** – Takes input
- **OUTPUT** – Provides an output
- **RETURN** – Return type

Arduino Data Types:

In C language the data type refers to a comprehensive system that is used to declare variable or different types of functions. The type of a variable determines how much space it is in storage and how the stored bit pattern is used.

The following table provides all the data types used during Arduino programming:

void	Boolean	char	Unsigned char	byte	int	Unsigned int	word
long	Unsigned long	short	float	double	array	String-char array	String-object

Reference: https://www.tutorialspoint.com/arduino/arduino_program_structure.html

4.3.3 Algorithms Design

i. Set pin Mode In1, In2, In3, In4, ENA1, ENA2, ENA3, ENA4, Trig pin, echo pin
ii. Analog Write (ENA1, Speed)
(ENA1, Speed)
iii. Long duration, distance.
iv. Set trig pin to high
v. delay (10)
vi. set trig pin to low
vii. read duration=echo pin
viii. distance= (duration/2)/29.1
ix. delay (10)
x. while Bt. available ()
delay (10)
read char c
readvoice +=c
if readvoice.lenght()>0
if readvoice==”*go#”
car move’s ahead
if distance<10
car stops
else if readvoice ==”*back#”
car move’s back
else if readvoice ==”*left#”
car move’s left
else if readvoice ==”*right#”
car move’s right
readvoice=””

4.4 USER INTERFACE DESIGN

User interface design gives us a simplified view of how the user interacts with the main module. In this UI design we can see how user interacts with the car for controlling it.

User gives voice command through the apk then that apk is translated by google using google translator. The apk receives the translated apk in the form of apk from google. Then the apk sends this string value to the Arduino using Bluetooth command. And then just user need to watch the output result.

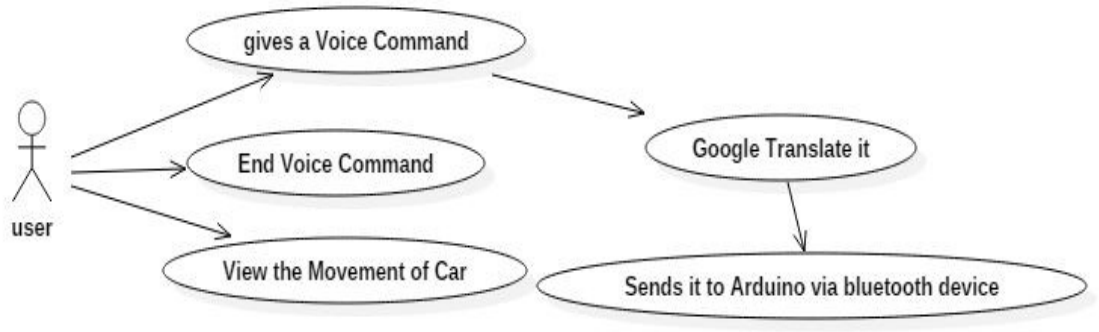


Fig 4.6: User Interface

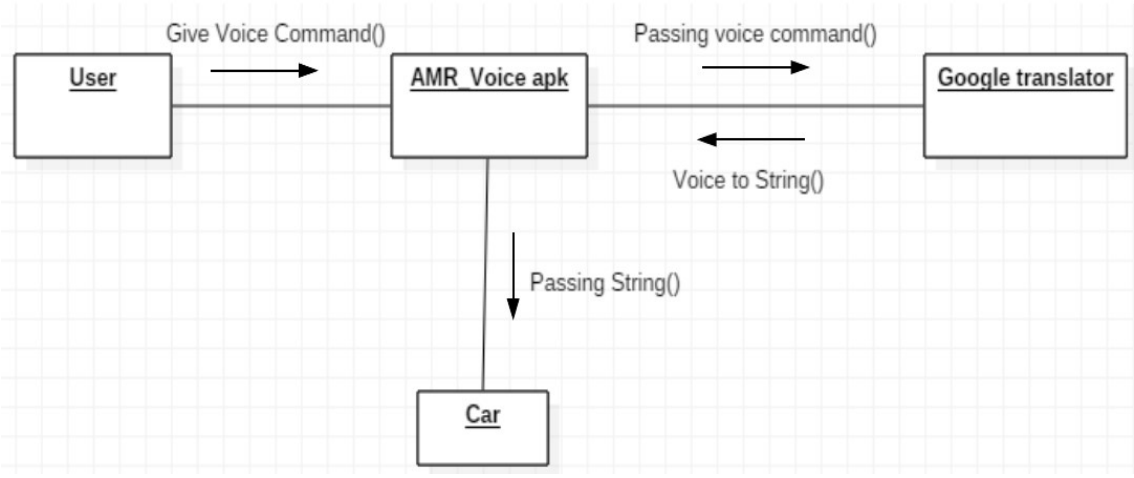


Fig 4.7: User Interface module

4.5 SECURITY ISSUES

These concerns lead us back to the important issue of security. Though we emphasized this when talking about software prototyping, it is critical to consider the consequences of a mass-market product being attacked in this way.

There are other issues related to encryption via HTTPS, and the requirement or otherwise for this may have driven our choice of prototyping or production platform. As well as securing the communication between client device and web server, you should consider the server itself; as an always-on, always- connected device visible to the whole Internet, it is the most obvious target. These considerations are not specific to the Internet of Things, but they should be well known to your system administration and development team. Following are some of the more important guidelines:

- Make sure that servers are kept up-to-date with the latest security patches, are hardened with the appropriate firewalls, and detect and mitigate against password hacking attempts and rootkit attacks.
- User passwords should never be stored in plain text, if your database were ever compromised, an attacker could easily log in as any userbase touched upon in the sidebar on "Hashes" passwords should be encrypted with a secure algorithm which is not known to be trivially cracked, and "salted" for additional security.
- Never simply trust user input. Check that anything that is entered into a web application fits the type of data we expect, and refuse or clean anything which doesn't. Although we can think input from our connected devices would be okay (because we wrote the code), it is possible that it has been compromised or an attacker may be "spoofing" it. In particular, be wary of passing user input to our database without checking it (otherwise, we risk an SQL injection attack were it to include SQL commands), or including unfiltered user input in our HTML pages, as this could allow a cross-site scripting (XSS) attack Strip out all HTML tags (or allow only a limited selection of acceptable ones for formatting) or escape the output.
- Be aware of cross-site request forgery (CSRF) attacks from other malicious or compromised websites. For example, if one of the users browses a bad site which uses JavaScript to open http://some.example.com/heating?switcheoff on our site and the user is already logged in, he may come home to a cold house.

4.6 TEST CASES DESIGN

For test case we have made a fully working prototype which will be some what similar to our final model. Its most of the functions are similar to our final working model. The sensor works perfectly. The code runs correctly.

But as it's just a prototype there were many trial and error which we went through like:

- The tyres where not proper so we needed to change it.
- Battery voltage was less so we increased the batteries.
- There where many connections problems that we faced like connecting wires wrong.
- We had to change the program several times through out the process which resulted in a final working model

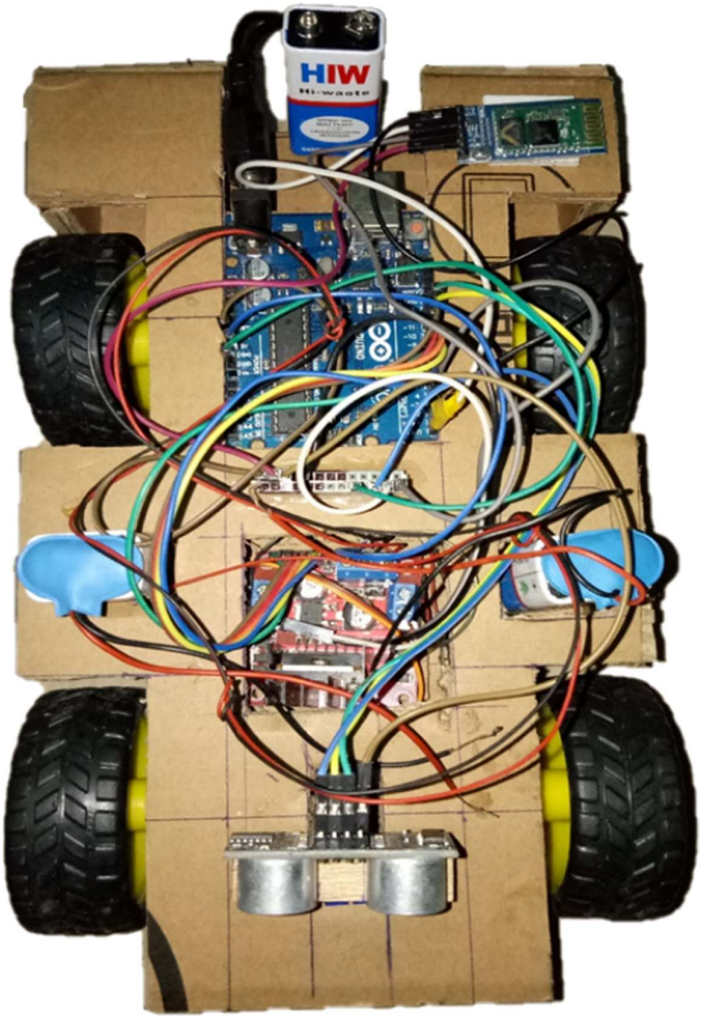


Fig 4.8: Top View

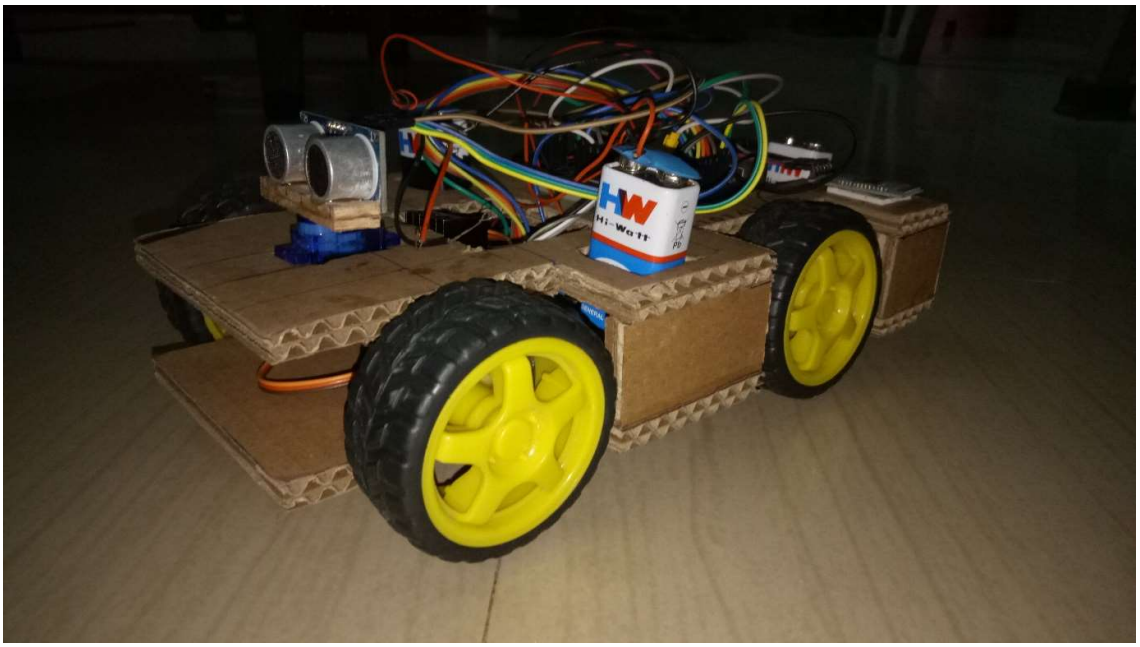


Fig 4.9: Side View

This is how the test model looks like it's made of hard cardboard which keeps everything in one place. The motors and motor drivers are place inside as they are fixed and not need to change again. Other things which are used moreover are kept above like sensors Arduino board.

The structure is solid and light weight which gives less load on the motors and hence it's fast and amazing.

CHAPTER 5: IMPLEMENTATION AND TESTING

Implementation: Implementation is the practice, execution, or practice of a plan, a method, or any design, specification, idea, model, standard or policy. As such, implementation is a verb that should actually follow any preliminary thinking for something. Project implementation is a phase where an idea becomes reality.

Testing: Testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. For example, in software development, product objectives are sometimes tested by product user representatives. When the design is complete, coding follows and the finished code is then tested at the unit or module level by each programmer; at the component level by the group of programmers involved; and at the system level when all components are combined together. At early or late stages, a product or service may also be tested for usability.

5.1 IMPLEMENTATION APPROACHES

. There are various approaches for implementing our project plan and strategy into action. A product software implementation method is a systematically structured approach to effectively integrate a software-based service or component into the workflow of an organizational structure or an individual end-user. Implementation approach focuses on the process modelling side of the implementation of large product software, using the implementation planning systems as the main component.

A product software implementation method is a blueprint to get users and/or organizations running with a specific software product. The method is a set of rules and views to cope with the most common issues that occur when implementing a software product: business alignment from the organizational view and acceptance from human view. The implementation of product software, as the final link in the deployment chain of software production, is in a financial perspective of a major issue.

Implementation methods can on the one hand be used as a guiding principle, indicating that the method serves as a global idea about how the implementation phase of any project should run. This choice leaves more room for situational factors that are not taken into account in the chosen method, but will result in ambiguity when questions arise in the execution of the implementation process.

On the other hand, methods can be used as a profession, meaning that the method should be taken strict and the usage of the method should be a profession, instead of a guiding principle. This view is very useful if the implementation process is very complex and is very dependent on exact and precise acting. Organizational and quality management will embrace this view, as a strict usage of any method results in more clarity on organizational level.

We have used DSDM method of implementation for our project. The power of dynamic systems development method is that the method uses the principles of iteration and incremental value, meaning that projects are carried out in repeating phases where each phase adds value to the project. the possibility to execute different phases of the implementation process iteratively enables the process to be executed by incrementally aligning the product to be implemented with the end-user. In this way implementation phases can be carried out incrementally and add value to important project aspects such as the degree of acceptance, awareness and skills within every increment. We have designed our project in increments. The incremental approach consists of successively realizing the functional elements of the product that are directly usable. The project is divided into a number of deliverable elements, which are called increments. Each increment is an operational product, partially in the beginning of the cycle (with a reduced perimeter) and totally in the end of the cycle (with a full perimeter).

An increment is a coherent functional part of the final product. It is characterized by the fact that:

- Each increment is like making a small part of big system.
- Each increment adds new features to the system.
- Each increment is tested as a final product.

The incremental approach is only possible if we have a clear idea of the software application's architecture and interfaces (we must not return to the parts that are developed at each increment) and the expected features are separable (ability to be developed independently).

So, first we gathered the information regarding our project. We did a lot of research of how can we implement our project into real world system. We were very much aware about the voice control technology emerging a lot this day so we decided to make a car which can run on voice commands. We started to gather requirements based on the information we had. So first we had to buy components for making the car them we need to check if each component is working or not based on which we can proceed further with our work.

After testing each module we then started to integrate each module into one car for that we needed a platform so there were many option in the market we could have brought a readymade module but we decided not to buy instead we made one our self so as it also saved our money as well as we modified it as per our requirements we first started with very basic module we just used Aluminium board on which we just kept everything and checked if its working or not.

Then we made a model with single sheet of cardboard which was reliable but it was too heighted so there was difficulty for some components like Ultrasonic sensor it couldn't detect things easily and the structure wasn't that strong either so then we decided to make a strong model with double sheet of cardboard and also big that all the components will fit perfectly in it. Later we added LED to give it a real car look. We then added a hood so that it can protect the crucial parts of the car that is its components. We then added some extra features like LDR sensors which were helpful in the dark. And after that we performed final testing and the car was ready.

As we changed the car models and added more features into the car we also did a lots of changes in the coding part as that is also important. That's what keeps our car performing well and that's how we implemented our project.

5.2 CODING DETAILS AND CODE EFFICIENCY

- **Program Code**

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial BT(0, 1);
String readvoice;
//L298N Driver pins
const int In1 = 7;
const int In2 = 6;
const int In3 = 4;
const int In4 = 5;
//Sensor pins
#define trigPin 11
#define echoPin 12
//LED pins
int red = A0;
int frontled = A1;
int rightled = A2;
int leftled = A3;
const int ldr = A4;  // LDR pin
int distance, duration;
Servo servo;  //servo pin
void setup()
{
 pinMode(In1, OUTPUT);
 pinMode(In2, OUTPUT);
 pinMode(In3, OUTPUT);
 pinMode(In4, OUTPUT);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(red, OUTPUT);
 pinMode(frontled, OUTPUT);
 pinMode(rightled, OUTPUT);
 pinMode(leftled, OUTPUT);
//initialize the LDR pin as an input
 pinMode(ldr, INPUT);
 servo.attach(10);
 servo.write(90);
 delay(1000);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 distance = readPing();
 delay(100);
 BT.begin(9600);
 Serial.begin(9600);
}
void loop()
{
 while (BT.available())
  {
  //Delay added to make thing stable
   delay(10);
    //Conduct a serial read
   char c = BT.read();
   readvoice += c; //build the string-
"go", "back", "left" and "right"
   //Serial.print("Available!!");
  }
```

```
//read the status of the LDR value
int stat = analogRead(ldr);
 if (stat <=400)
 {
  glowfront(); //turn LED on
Serial.println("Lights are off, LED is
ON");
 }
 else
 {
  closefront();//turn LED off
  Serial.println("Lights are on");
 }
 int distanceRight = 0;
 int distanceLeft = 0;
 delay(50);
 distance = readPing();
 delay(100);
 Serial.print("Distance ");
 Serial.print(distance);
 Serial.println(" cm");

 //Code for Voice Commands
 if (readvoice.length() > 0)
 {
  //Serial.print("Here "+readvoice);
  Serial.println(readvoice);
  if (readvoice == "*go#")
  {
   ahead();
   again:
   while (BT.available())
   {
    delay(50);
    char c = BT.read();
    readvoice += c;
   }
   distance = readPing();
   delay(50);
   Serial.print("Distance in go ");
   Serial.print(distance);
   Serial.println(" cm");
   delay(50);
   if (distance <= 20 || readvoice ==
"*stop#")
   {
    moveStop();
    delay(100);
   }
   else
   {
    readvoice = "";
    goto again;
   }
  }
  else if (readvoice == "*back#")
  {
   back();
   delay(4000);
   moveStop();
  }
  else if (readvoice == "*go back#" ||
readvoice == "*backup#")
  {
   back();
  }
  else if (readvoice == "*left#")
  {
```

```
  digitalWrite(leftled, HIGH);
  distanceLeft = lookLeft();
  delay(300);
  if (distanceLeft <= 20)
  {
   digitalWrite(red, HIGH);
   delay(500);
   digitalWrite(red, LOW);
   delay(500);
   digitalWrite(red, HIGH);
   delay(500);
   digitalWrite(red, LOW);
   delay(500);
   digitalWrite(red, HIGH);
   delay(500);
   digitalWrite(red, LOW);
   delay(50);
  }
  else
  {
   left();
  }
  digitalWrite(leftled, LOW);
}
else if (readvoice == "*right#")
{
  digitalWrite(rightled, HIGH);
  distanceRight = lookRight();
  delay(300);
  if (distanceRight <= 20)
  {
   digitalWrite(red, HIGH);
   delay(500);
   digitalWrite(red, LOW);
   delay(500);
   digitalWrite(red, HIGH);
   delay(500);
   digitalWrite(red, LOW);
   delay(500);
   digitalWrite(red, HIGH);
   delay(500);
   digitalWrite(red, LOW);
   delay(50);
  }
  else
  {
   right();
  }
  digitalWrite(rightled, LOW);
}
else if(readvoice == "*keep watch#"
|| readvoice == "*keepwatch#")
{
  anotheragain:
  while (BT.available())
  {
   delay(50);
   char c = BT.read();
   readvoice += c;
  }
  if (distance <= 20)
  {
   moveStop();
   delay(300);
   back();
   delay(400);
   moveStop();
   delay(300);
```

```
  distanceRight = lookRight();
  delay(300);
  distanceLeft = lookLeft();
  delay(300);
  if (distance >= distanceLeft)
  {
   right();
   moveStop();
  }
  else
  {
   left();
   moveStop();
  }
}
else
{
 ahead();
}
distance = readPing();
Serial.print("I'm Watching ");
Serial.print(distance);
Serial.println(" cm");
delay(1000);
if (readvoice == "*stop#")
{
 moveStop();
 delay(100);
}
else
{
 readvoice = "";
 goto anotheragain;
   }
  }
  else if (readvoice == "*round#")
  {
   circle();
  }
  //LIGHTS
  else if(readvoice == "*frontlight#" ||
readvoice == "*front light#" )
  {
   delay(50);
   glowfront();
  }
  else if(readvoice == "*close front#"
|| readvoice == "*close light#" )
  {
   delay(50);
   closefront();
  }
  else if (readvoice == "*night#")
  {
   glowlights();
  }

  else if (readvoice == "*morning#")
  {
   closelights();
  }
  else if (readvoice == "*red on#" ||
readvoice == "*redon#")
  {
   digitalWrite(red, HIGH);
  }
```

```
    else if(readvoice == "*red of#" ||
readvoice == "*redoff#" || readvoice
== "*red off#" || readvoice ==
"*redof#")
    {
      digitalWrite(red, LOW);
    }
    else if (readvoice ==
"*systemcheck#" || readvoice ==
"*system check#")
    {
      ahead();
      delay(1000);
      moveStop();
      delay(200);
      back();
      delay(1000);
      moveStop();
      delay(50);
      lookRight();
      delay(500);
      lookLeft();
      delay(500);
      digitalWrite(frontled, HIGH);
      delay(500);
      digitalWrite(frontled, LOW);
      delay(200);
      digitalWrite(red, HIGH);
      delay(500);
      digitalWrite(red,LOW);
      delay(200);
      digitalWrite(rightled, HIGH);
      delay(500);
      digitalWrite(rightled, LOW);
      delay(200);
      digitalWrite(leftled, HIGH);
      delay(500);
      digitalWrite(leftled, LOW);
      delay(200);
      glowlights();
      delay(1000);
      closelights();
      delay(200);
      glowlights();
      delay(1000);
      closelights();
      delay(200);
      glowlights();
      delay(1000);
      closelights();
      delay(200);
      glowlights();
      delay(1000);
      closelights();
      delay(200);
    }
    //STOP
    else if (readvoice == "*stop#")
    {
      moveStop();
    }
    readvoice = "";
  }
}
int lookRight()
{
  servo.write(0);
  delay(500);
```

```
  int distance = readPing();
  delay(100);
  servo.write(90);
  Serial.print("Right Distance ");
  Serial.print(distance);
  Serial.println(" cm");
  return distance;
  delay(100);
}
int lookLeft()
{
  servo.write(180);
  delay(500);
  int distance = readPing();
  delay(100);
  servo.write(90);
  Serial.print("Left Distance ");
  Serial.print(distance);
  Serial.println(" cm");
  return distance;
  delay(100);
}
float readPing()
{
  delay(70);

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, HIGH);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) * 0.0343;
  //distance = distance + 10;
  return distance;
}
void ahead()
{
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);
  delay(100);
}
void back()
{
  digitalWrite (In1, LOW);
  digitalWrite (In2, HIGH);
  digitalWrite (In3, LOW);
  digitalWrite (In4, HIGH);
  delay (100);
}
void circle()
{
  digitalWrite (In1, HIGH);
  digitalWrite (In2, LOW);
  digitalWrite (In3, LOW);
  digitalWrite (In4, LOW);
  delay (100);
}
//LED LGHTS
void glowlights()
{
  digitalWrite(frontled, HIGH);
  digitalWrite(red, HIGH);
  digitalWrite(rightled, HIGH);
  digitalWrite(leftled, HIGH);
}
```

```
void closelights()
{
 digitalWrite(frontled, LOW);
 digitalWrite(red,LOW);
 digitalWrite(rightled, LOW);
 digitalWrite(leftled, LOW);
}
void glowfront()
{
 digitalWrite(frontled, HIGH);
}
void closefront()
{
 digitalWrite(frontled, LOW);
}
void right()
{
 delay(500);
 digitalWrite (In1, HIGH);
 digitalWrite (In2, LOW);
 digitalWrite (In3, LOW);
 digitalWrite (In4, HIGH);
 delay (1100);
 digitalWrite (In1, LOW);
 digitalWrite (In4, LOW);
 delay (100);
}
void left()
{
 delay(1000);
 digitalWrite (In1, LOW);
 digitalWrite (In2, HIGH);
 digitalWrite (In3, HIGH);
 digitalWrite (In4, LOW);
 delay (1100);
 digitalWrite (In2, LOW);
 digitalWrite (In3, LOW);
 delay (100);
}
void moveStop()
{
 digitalWrite(red, HIGH);
 digitalWrite (In1, LOW);
 digitalWrite (In2, LOW);
 digitalWrite (In3, LOW);
 digitalWrite (In4, LOW);
 delay (500);
 digitalWrite(red, LOW);
}
```

5.2.1 Code Efficiency

Code efficiency plays a significant role in applications in a high-execution-speed environment where performance and scalability are paramount. In simple word, code efficiency is how fast your system is working and is capable of taking high load. Well-developed programming codes should be able to handle complex algorithms.

We wrote the code in such a way that it could also be understood by the any other user so that it makes it easy for him also to understand the code and make it easy to change. We used function as we needed to use a lot of code again and again. We also made us of go to statement which required a lot of research because it was not common in Arduino programming.

Our project code is efficient and reusable. Following are some points shows that our project is code efficient-

- We have kept reusable code separate in a function or in a file.
- We have used enough variable to balance the memory uses.
- We have created different methods to store data.
- Flow of code is efficient.
- We removed unnecessary code that goes to redundant processing
- We developed a programming code that is compliant with the flow and design logic.
- We used keywords, data types and variables, and other available programming concepts to implement the related algorithm.

5.3 TESTING APPROACH

Testing is basically a process of verifying a computer system/program to decide whether it meets the specified requirements and produces the desired results. There are various approaches for testing our project. Following are some of them-

5.3.1 Unit Testing

Before starting with unit testing, first we need to understand the meaning of unit. So, unit is the smallest part of software or hardware which includes codes, classes, methods or hardware modules and can be tested individually for correctness. Unit testing is a validation technique using black box methodology which mainly concentrates on requirements of the system. In unit testing, individual components and units are tested to ensure that they work as an individual as defined in design.

We have selected register module for unit testing. The registration module field has following fields –

- **Arduino UNO**
- **Bluetooth HC-05**
- **Motor Driver L298n**
- **BO Motor**
- **Servo Motor**
- **LED lights**

- **Arduino UNO**

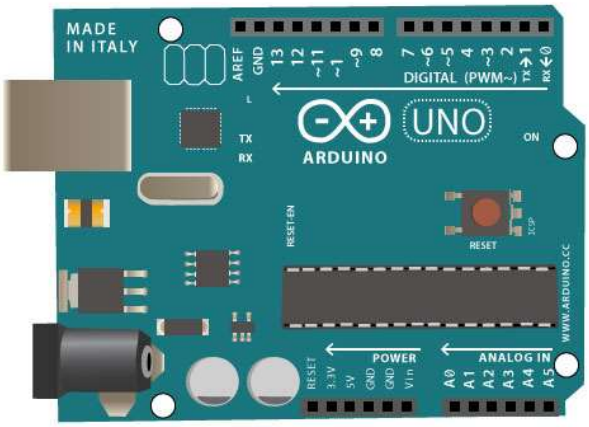


Fig. 5.1 Arduino UNO

This is an Arduino UNO, it has 13 PWM pins which are used to connect other components with Arduino Board. It also has 6 Analog pins which can also be used for connecting other devices. This board can provide two types of power supply namely 5volts and 3.3volts. It has three ground (GND) pins. The PWM pins 0 and 1 are used for Receiving (RX) and Transmitting (TX) signals respectively. It has reset button which is used for resetting the embedded code in the Arduino UNO.

All the code is uploaded in ATMEGA microchip. the code is uploaded by the user through a DATA Cable which connect with the PC (CPU). There are two ways of providing power supply to Arduino board first way is through the Data Cable which is connected with the PC and another way is connecting batteries to the power jack of Arduino board through the power jack pin.

This Arduino UNO is main component of our project because all other components are connected to this board and it is controlled by a code which is embedded in it by the user.To perform the unit testing on the Arduino UNO Board, we Simply connect the Arduino Board with the PC using a Data Cable after connecting it the lights on the Arduino Board turns on which indicates that the Arduino Board is ON and it is working. There is predefined code in the Arduino UNO, which is used to blink the LED at PWM pin 13.

This is how we tested our Arduino UNO Circuit Board is working properly.

- **Bluetooth HC-05**

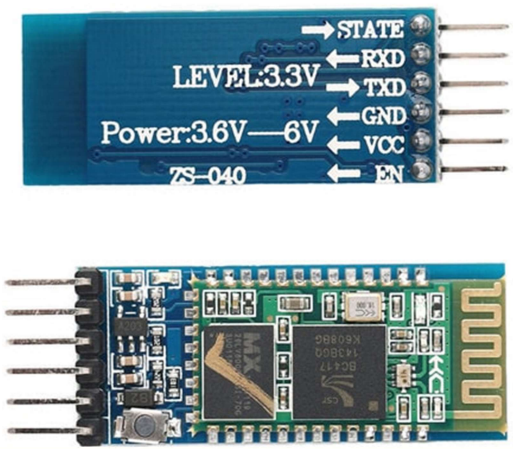


Fig. 5.2 Bluetooth HC-05

HC-05 module Bluetooth SPP (Serial Port Protocol) module is easy to use, which is designed for transparent wireless serial connection setup.The serial port Bluetooth module is fully qualified Bluetooth V2.0 + EDR (Enhanced Data Rate) with a full 2.4GHz radio transceiver and baseband with 3Mbps modulation. It uses CSR Bluecore 04-External Single Chip Bluetooth system with CMOS technology and AFH (Adaptive Frequency Hoping Feature). It has a small footprint of 12.7mmx27mm. Hope this will simplify your overall design / development cycle. The latest specification of Bluetooth SIG, Bluetooth, is now supported by select microchip devices. Microchips current products are again certified to ensure new and existing designs in line with this standard. Whether or not certified products are available for Bluetooth check.BT Bluetooth module is a stainless shield with serial port based on HC-06. The gradient for wireless communication can be directly linked to the Arduino UART port. Without hindrance or other interference, Bluetooth shields can communicate in the range of 10 meters (32 feet).

So to test this device we connected 3volt battery to HC-05 Bluetooth module in order to give it power supply. As we gave the power supply the led on the Bluetooth Module started blinking. So we with LED Blinking we can consider that the Bluetooth Module is working. To confirm that Bluetooth Module is working we connected this Module with Mobile Bluetooth. First we paired the Mobile Bluetooth with Bluetooth Module, to do so the Bluetooth Module asked for a password that ensures that the Bluetooth Module is and the password is “1234”. the password can be change by going into the Bluetooth AT mode.

This is how we ensured that the Bluetooth module is working.

- **Motor Driver L298n**

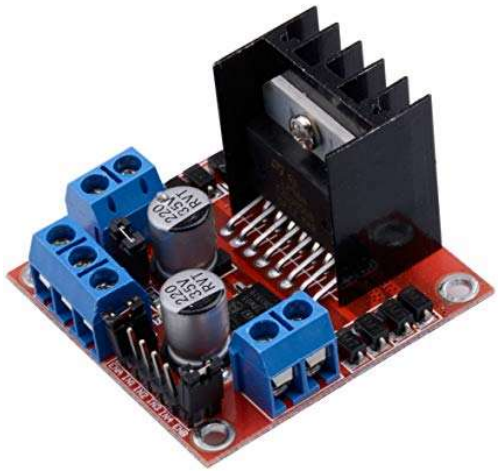


Fig. 5.3 Motor Driver L298n

The L298N double bidirectional motor driver is based on the very popular L298 Dual H-Bridge motor driver integrated circuit. The circuit will allow you to easily connect and independently control four motors of 2.5 A in each of the two directions. This robot is ideal for applications and requires a pair of well motor control lines, which is suitable for a microcontroller connection. It can also be interfered with simple manual switches, TTL logic gates, relays, etc.

Two high-power currents are drawn on each motor through your microcontroller. A user on the L298N board can use the 5V regulator and can be included in it, which can be used to supply any additional circuit, for which approximately 5A DC supply is required for approximately 1A. The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

To test this module we gave it power supply twice once with +5V and other with +12V as the motor driver has two options for providing power supply. +5V provides only 5V to the motor whereas giving power supply in +12V one can provide upto 35V of voltage to the motors. Both the techniques work with an LED blinking that states that the motor driver is working.

- **BO Motor**

Fig. 5.4 BO Motor

The motor we used in this project is DC motor and it comes with the gear box which provides good torque and rpm in the lower voltage of DC current. This type of DC motor is run at approximately 200rpm, when it runs on a double V-Watt cell battery at 6 V and up to 300 rpm, when it connects to 9V. This light weight project is best suited for DC motor which runs on low voltage.

There are types of Bo motor available in market. They come's in different shapes, size, speed, power, single shaft and double shaft. we used single shaft Bo motor means one sided shaft Bo motor. The RPM of this motor is 300 which mean the shaft of Bo motor will rotate 300 times in one minute. This DC motor have a plastic gear box which gives good torque and rpm at lower voltages. It is most suitable DC motor for light weight robotic type project on lower voltage.

- **Servo Motor**

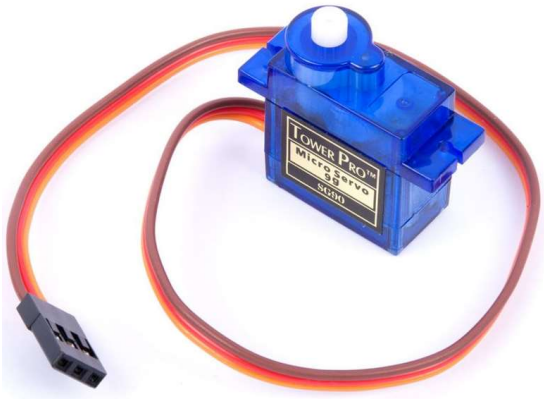


Fig. 5.5 Servo Motor

This is TowerPro SG90 9g Mini Servo and it is 180° rotational servo. It is a Digital Servo Motor which receives and processes PWM signal faster and better. It equips 3-wire JR servo plug which is compatible with Futaba connectors too.

It is servo motor like Bo motor but it having more functionality than Bo motor. For example, we can control the speed, degree, torque and we can read that shaft of servo motor that which side or at which degree it is.

This motor also has a gear set in it and along that motor also. Which a plastic container or plastic case hold the motor and gears together. It is hold in a such a way that motor inside it connected to gear set. Which reduces or slowing the speed of motor in it. It has three wire with female jumper wire connector which we can easily connect to Arduino.

Inside this servo there is potentiometer and chip which gives the reading of motor that what is the degree of shaft.

To test the servo, we wrote a set of codefor testing the servo motor. To test the motor, we connected the wire of servo motor to Arduino with jumper wires. The code which we wrote gets executed and gives Command to servo motor to rotate the motor shaft which will rotates gear inside servo. And the servo motor was working properly.

- **LED lights**

Fig. 5.6 LED lights

The LED which we used in our project is 5mm LED's. They are available in different colours, but which used are Red, Yellow, and White as you can see in picture given above. This LED's is used in our project for front headlight and for back light. The voltage requirement for this LED's is 3.3 voltage.

To test the LED's, there is inbuild code for LED's in the Arduino it does not require any extra code for testing the LED. To test the LED's, we connected the positive wire to Arduino power pin which gives 3.3volt power supply to led and negative wire GND (ground) pin of Arduino. To figure out the LED positive pin and Negative pin is easy the positive pin of LED will be longer than Negative pin. And negative pin will be shorter than positive pin.

5.3.2 Integrated Testing

Integration testing involves integration of units to make a module / integration or modules to make system integration. Integration testing may start at module level, where different units and components come together to form a module, and go up to system level.

Integration testing has following types –

- **Bottom-Up Testing**- In this type of testing, it focuses on testing individual units and modules first and then goes upward by integrating tested and working units and modules for system testing and intersystem testing.
- **Top-Down Testing-** In this type of testing, the top level of application is tested first and then it goes downward until it reaches the final component of the system.

We have used bottom up approach for testing our project. First, we start with testing the lower units viz. are register, login and logout module. Then, we integrate them to test whether these components functioning correctly or not.

Following tests have been performed for integration of these modules –

- Testing Arduino with a simple LED code
- Testing Bluetooth Module HC-05
- Testing LDR sensor
- Testing Ultrasonic sensor
- Testing Motor Driver and Motors
- Testing Servo Motor

In the following tests we used a common module i.e. Arduino which is the main component of the model and it integrates everything together the code is written in Arduino to be executed and through that we can control the whole system. So before performing integration we checked the Arduino Board and then we proceeded with the testing.

- **Testing Arduino with a simple LED code:**

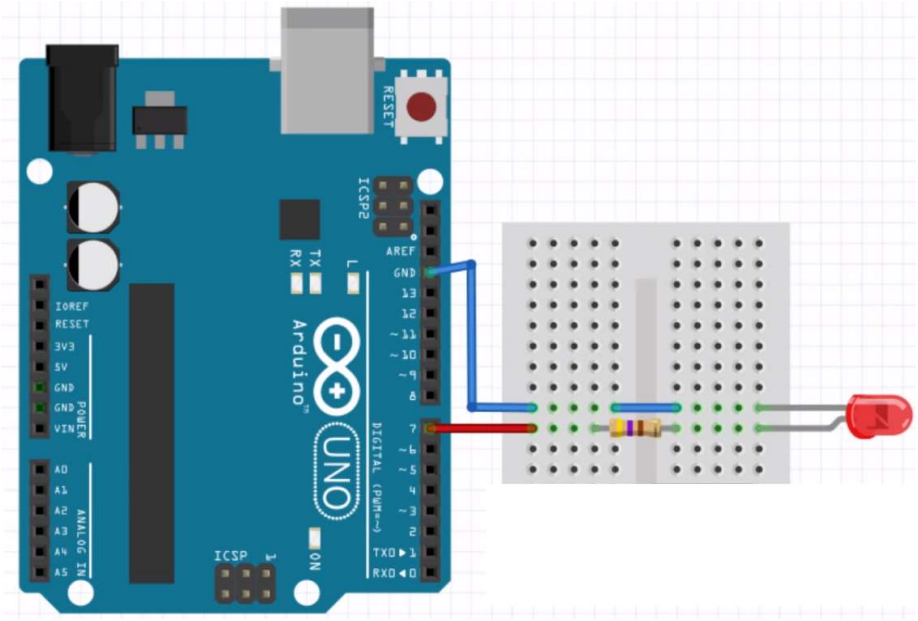


Fig. 5.7 Arduino with LED

The Code:

```
void setup() {
   pinMode(7, OUTPUT);     }
void loop() {
  digitalWrite(7, HIGH);
  delay(2000);
digitalWrite(7, LOW);
  delay(1000);
}
```

Output:

Arduino and LED works properly here in the output LED which is connected to Arduino port 7 blinks on and off continuously with a 2 second delay.

- **Testing Bluetooth Module HC-05:**

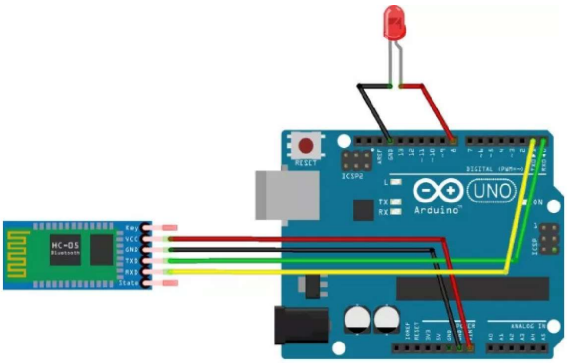


Fig. 5.8 Arduino with Bluetooth Module

The Code:

```
void setup() {
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}
void loop() {
  while(Serial.available())
 { delay(10);
    char c=Serial.read();
    voice += c;
}
if (voice.length() > 0) {
  Serial.println(voice);
  if (voice == "*led on#")
  { digitalWrite (8, HIGH); }
  else if (voice == "*led off#")
  {
digitalWrite (8, LOW);
}
voice="";
}
```

Output:

Here an LED is connected to Arduino pin 8 and the Tx (Transmission) & Rx (Receiving) pins of the HC-05 module are connected to 0 & 1 respectively. So, after we give voice command through an android app via Bluetooth it responds to it. On giving led on command led glows then after we give led off command the led light goes off. That is how we tested our HC-05 module as well as tested the voice command.

- **Testing LDR sensor:**

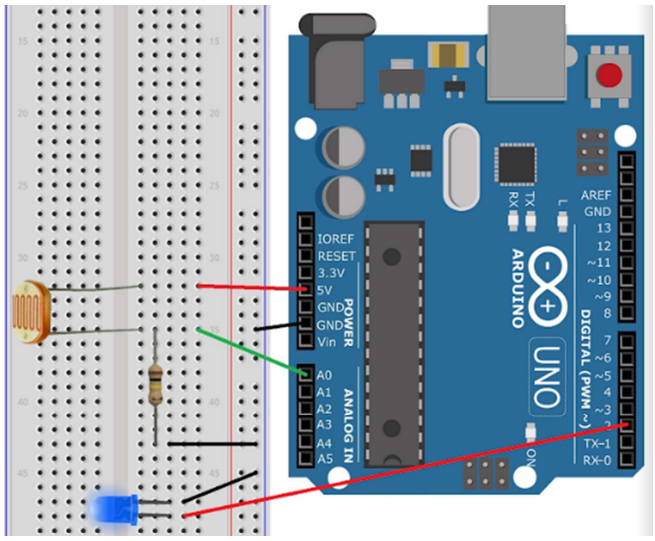


Fig. 5.9 Arduino with LDR

The Code:

```
int sensPin = A0;
int sens = 0;
void setup()
{
 pinMode(2, OUTPUT);
 Serial.begin(9600);
 }
void loop()
{
 sens = analogRead(sensPin);
 Serial.println(sens);
 if(sens< 400)
        digitalWrite(2,HIGH);
else
        digitalWrite(2,LOW);
delay(100);
```

Output:

With this test we tested the LDR sensor which detects the surrounding lights that falls on the surface of LDR and detects the intensity of that light based on it if the intensity of light is low that means its dark outside so it glows the LED else the LED is kept off.

- **Testing Ultrasonic sensor:**

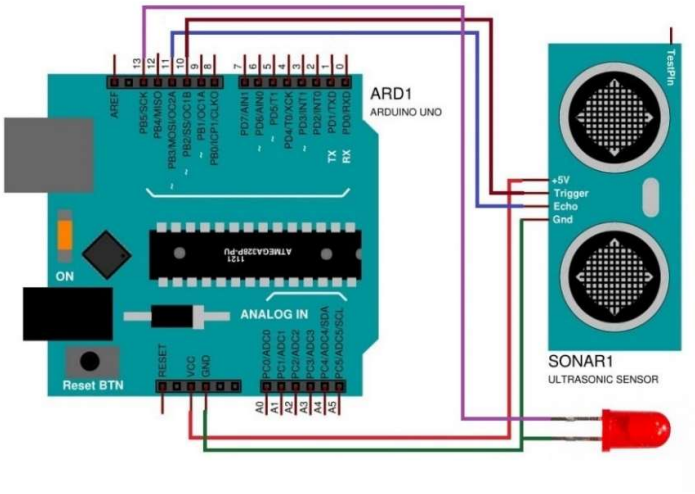


Fig. 5.10 Arduino with Ultrasonic Sensor

```
const int trigPin = 10;

const int echoPin = 11;

const int ledPin = 13;

long duration;

int distance;

int safetyDistance;

void setup()

{

pinMode(trigPin, OUTPUT);  //Trigger output pin

pinMode(echoPin, INPUT);  //Echo input pin

pinMode(ledPin, OUTPUT);

Serial.begin(9600);

}
```

```
void loop()
{digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance= duration*0.034/2;
safetyDistance = distance;
if (safetyDistance <= 20)
{         digitalWrite(ledPin, HIGH);              }
else{
  digitalWrite(ledPin, LOW);  }
 Serial.print("Distance: ");
Serial.println(distance);          }
```

Output:

With this test we tested the Ultrasonic sensor which detects the surrounding obstacles which come's in front of the Sensor, actually what it does is that it sends echo when the echo falls on an object it reflects the echo which is catch by the Sensor based on the duration of sending and receiving the echo it calculates the distance between itself and the object.

In this testing we connected the LED to pin 13 and we connected the echo and trigger pin to pin 11 and 10 respectively to Arduino Board. Accordingly, we programmed the code as you can see above and as per the code the ultrasonic sensor detects the object in front of it. If the distance between the object and sensor is less than 20cm then the pin 13 on Arduino will activate and the LED will glow.

This is how we tested our Ultrasonic sensor is working.

- **Testing Motor Driver and Motors:**

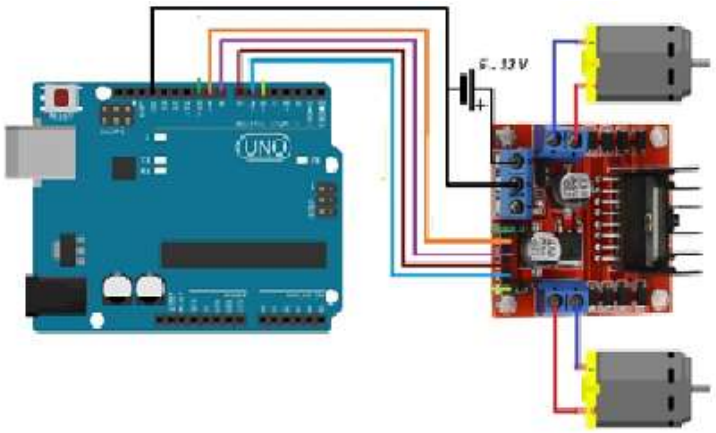


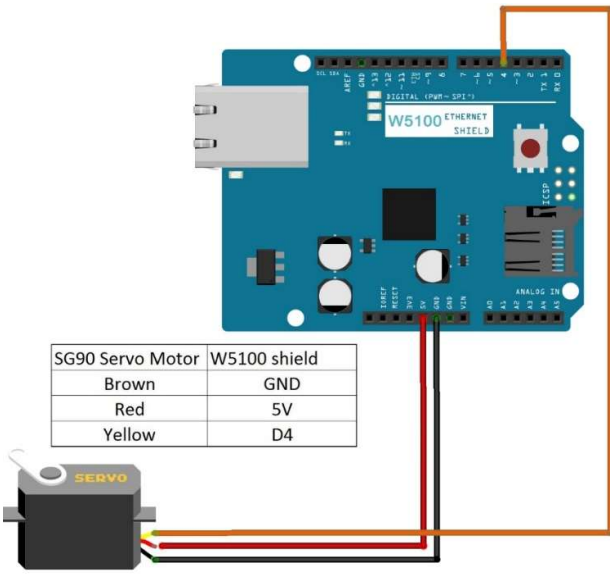
Fig. 5.11 Arduino with Motor Driver

```
int In1=9;
int In2=8;
int In3=7;
int In4=6;
void setup()
{
  pinMode(In1,OUTPUT);
  pinMode(In2,OUTPUT);
  pinMode(In3,OUTPUT);
  pinMode(In4,OUTPUT);
}
void loop()
{
   digitalWrite(In1,HIGH);
   digitalWrite(In2,LOW);
   digitalWrite(In3,HIGH);
   digitalWrite(In4,LOW);
   delay(5000);
   digitalWrite (In1, LOW);
   digitalWrite (In2, LOW);
   digitalWrite (In3, LOW);
   digitalWrite (In4, LOW);
   delay (100);
}
```

Output:

With this test we tested the Motor Driver Module as well as the motors we powered Motor driver with +5 volts through Arduino we programed it so that the car will move for 5 seconds in forward direction and then it will stop it worked correctly.

- **Testing Servo Motor:**



SG90 Servo Motor	W5100 shield
Brown	GND
Red	5V
Yellow	D4

Fig. 5.12 Arduino with Servo

```
#include<Servo.h>
Servo servo;
int pos;
void setup() {
  servo.attach(10);
  servo.write(90);
  delay(2000);
}
void loop() {
  servo.write(0);
  delay(1000);
  servo.write(90);
  delay(1000);
  servo.write(180);
  delay(1000);
  servo.write(90);
  delay(1000);
}
```

Output:

With this test we tested the Servo motor we supplied with +5 Volts through the Arduino and we connected the data pin to Arduino pin 10 and programmed it to move 90^{o} right and wait for 1 second and then turn 180^{o} left and then after 1 second back to original position that's our test for servo motor.

5.3.3 Beta Testing

Beta testing is a part of user acceptance testing and done by the end user to validate the usability, functionality, compatibility and reliability of system. Beta testing adds value to the software development life cycle as it allows the "real" customer an opportunity to provide inputs into the design, functionality, and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively. Also, the user checks that the software is working proper as they had suggested.

To perform Beta testing we integrated the whole system together. We wrote a short code for Beta testing which included commands like “go”, “back”,” left”, and “right”. On giving those command following action was done by the car

- **go command:**

  On giving this command the car move’s in forward direction. If there is any obstacle in front of the car then car will instantly stop and the back LED will glow which indicates that the car has stopped.

- **back command:**

  On giving this command the car move’s in backward direction. If we need to stop the car we need to give the stop command and the car will instantly stop and the back LED will glow which indicates that the car has stopped.

- **left command:**

  On giving this command the car move’s in left direction but first the servo will rotate the ultrasonic sensor 90 degree left and check if there is any obstacle in front of the car if there is then car won’t turn left, else is there is no obstacle then the car will turn left.

- **right command:**

  On giving this command the car move’s in right direction but first the servo will rotate the ultrasonic sensor 90 degree right and check if there is any obstacle in front of the car if there is then car won’t turn right, else is there is no obstacle then the car will turn right.

This is what we tested in Beta testing and it worked properly.

5.4 MODIFICATIONS AND IMPROVEMENTS

Our project named voice-controlled car, is just the title of our project which we planned to make it. Basically, it is car which runs on voice command given to it.

The car model which we made for first time for our project is shown below. In our first car model we used only few components which were Arduino, Bluetooth, Motor Driver and Motor. These components we used to check and test the car if it would run or not.

In our first car model we used Aluminium plastic composite sheet as the structure of car where we placed the components on it and connect those components with jumper wires for the connection. In this basic designed car there is Arduino, Bluetooth, Motor Driver and Motor where a command received by the Bluetooth and transmitted to the Arduino and Arduino does processing and sends the command to motor driver that which motor has to work in which direction to go ahead, back, right, and left. This model was made very easily as we just needed to cut the aluminium sheet and place the components on it and just make it run with the code. At first it seemed easy but later we realized that it would take a lot of hard work and it did.

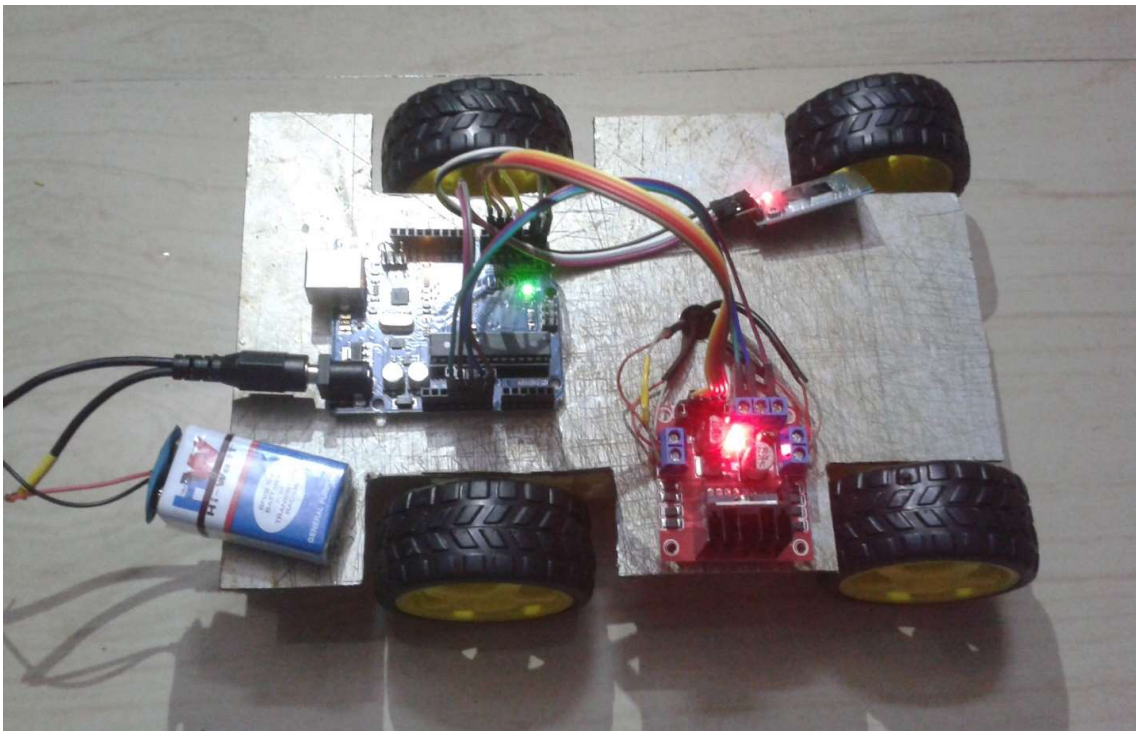


Fig. 5.13 First Car Module

And the component which provides it electricity is the battery, without battery it won't work so we did various research on which battery to choose as there were many options available to choose from. So, the battery that we used is 9volt battery as you can see in picture given above it is HW (HI – WATT) battery which gives power supply to Arduino through a power-jack on Arduino board.

The wheel which we have used in our project is a toy car wheel. The dimension of wheel is 7cm in diameter and 3cm in width as you can see in picture given above. Which perfectly fits with the Bo motor shaft.

For connection of components we jumper wires and normal wire. A jumper wire is very easy to connect components together. We used jumper wire to connect Arduino to Bluetooth and Arduino to Motor Driver, but for connection Motor Driver to Bo motor we use normal wire which we have taken from broken phone charger (charger adapter). We first soldered the one end of wire to the Bo motor and other end of wire connected to Motor Driver with screw which was there in Moto Driver.

To stick the Bo motor with Aluminium plastic composite sheet we use hot glue gun. And to place other components we use double sided foam tape. We use double sided foam tape because it's easy remove and change the place of components on car, for example Bluetooth module placed front side of car we want in back it just needs to pull off from there and place in back side of car.

This was the very basic model that we prepared we still remember the very first time giving the command "go" and guess what of course it didn't run! But later after a lot of research we found out that the power we supplied was not enough. Hence as a result we needed to add another battery and after that we were happy seeing the car move for very first time.

This is how we make our first car model project but, in this car, we cannot do anything else rather then move in front, back, left and right. Then we decided to add more things in it, like ultrasonic sensor which detects the object coming in front. And we use that ultrasonic sensor in our car to detect object as a obstacle. But problem was that it can only detect the object which in front of car, it cannot detect the object which is in left side or right side. This was because we didn't add anything that can make the ultrasonic sensor move around and we had to search for something that can move the ultrasonic sensor in 180^{o}

So to solve that problem, we use servo motor which rotates the shaft slower than a normal motor, and servo motor create a torque which was useful for us. And also rotates according to our application and also, we can control the rotating speed of servo motor.

To solve the ultrasonic side detection problem, we created a small platform for ultrasonic sensor from piece of wood. And that wooden piece placed on servo motor's shaft and on the wooden piece we placed an ultrasonic sensor. So, when shaft of servo motor rotates then ultrasonic sensor also rotates in that direction.

Then we added the object detection system which was working properly. The object detection system detects the object and react to that situation. The action taken by was like if the Object come in front of sensor the car should stop and move in other direction, it can be left or right. It depends on command given to the car.

One problem that occurred in changing the direction of the car was that the car height was higher so the object detection system was also on height. Which was a problem for our project. The object detection system was not able to detect the object which is lower. It was able to detect the object which was on height not a lower object.

To solve the detection problem, we created a new base structure for car which was made from cardboard. Because cardboard material is very light weight material to make project. It is very easy to make structure from cardboard and this carboard also known as corrugated cardboard which is made by corrugated board on a corrugating line. One roll of cardboard is corrugated and then glued between two other layers.

We used this because we can easily create any structure from it. And it is strong enough to make car structure for our project. So, we made a new structure of a car. We cut the shape on cardboard according to structure of car and stick the cardboard shape together, and car was ready. After completing the car structure, we started to place the components on a car. According where it should be placed. We When car was ready it looks like in a picture given below.

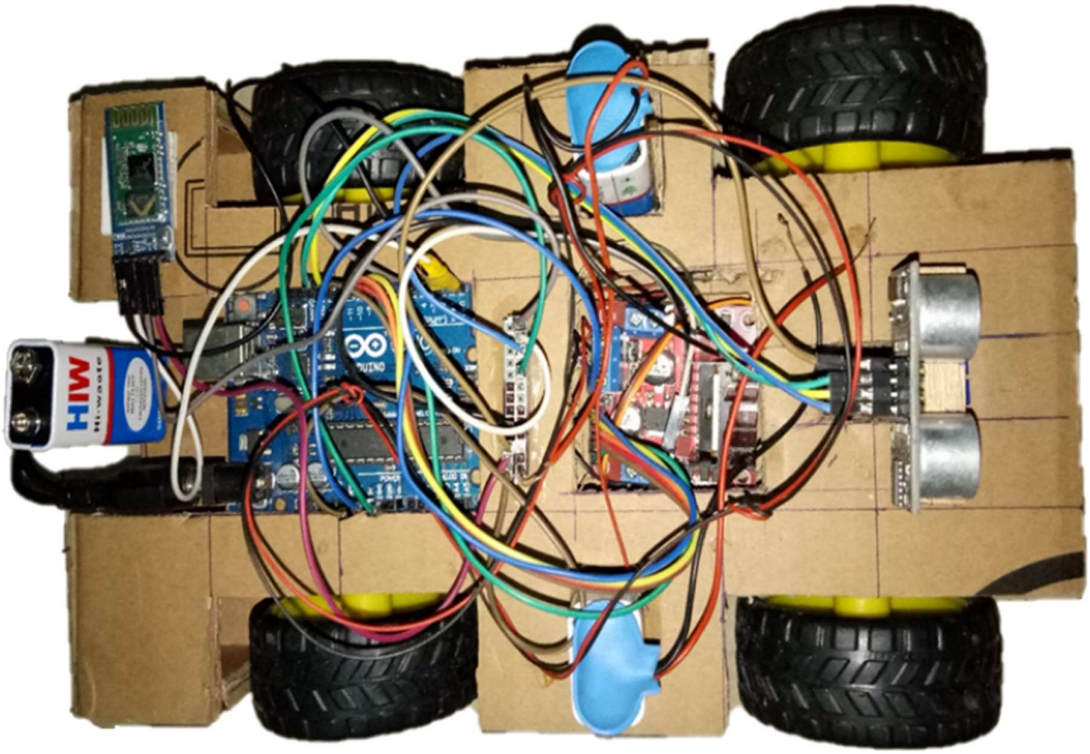


Fig. 5.14 Second Car Module

But in this car, there was still some limitation to detect object. It was not detecting object in left and right because of tyre. It was blocking the object and tyre was detected as object and car was not moving in left and right direction.so we made new car which able to do everything or fulfilling our needs that what we want in a car. For example, sensor should detect the obstacle in left, right and in front of car and also detect the object which is lower to ground that could be a small object or big object that doesn't matter. A car should react the situation when object is detected.

The new car which we make was better than previous car structures. Previous car structure was having certain limitation which was in first car structure like there was no object detection system, not a proper structure, structure material was heavy, we can't add any extra component in it. When we make a second car structure its was able solve previous car structure limitation which was object detection system that we added in second car structure and also, we added a battery holder mean we can place a battery in car. And yes, there was a battery problem in first car structure that we have stick a battery with a double-sided sticky tape which was not durable.

When we make third new car structure. The design of a car structure was better than previous car structures because in this car we ca add more thing like LED's, LDR system. The work of LED is simple, just we attached the positive pin to positive terminal of battery and negative pin goes to negative one. LDR (Light Dependent Resistor) is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits. Its mean's that an LDR (Light Dependent Resistor, or Photo resistor) is basically a resistor that change resistance depending on light. If there is more light the resistance will be less. Is there is Less light than resistance will be more. It is very cheap to use LDR in any project. To use LDR we need to connect the LDR to Arduino.

The LDR is a special type of resistor which allows a lower voltage to pass through it (high resistance) whenever its dark and higher voltages to pass (low resistance) whenever there is a high intensity of light.

Connect the A0 pin of the Arduino to one pin the LDR and resistor. And the other pin of LDR will connect to 3.3v pin Arduino. Now connect the other end of the resistor to the GND (ground pin of Arduino). Now take an LED and connect positive pin to the resistor pin and other pin will be connected to pin 13 of Arduino. And the negative pin of LED will connect to GND (ground pin of Arduino).

The code in Arduino chip will be executed. When there will be darkness in around the LDR won't detect any light, and as LDR comes to know its dark it sends this information to Arduino, then Arduino automatically turns on the LED to overcome the darkness. The Led is on until the LDR detects any Light around it so with the help of this sensor we don't need to worry if its dark or sunny outside because we will be always having brightness. This is how we used the LDR system in our project. There are various applications of LDR they can be used in street light system where there is manual need of starting the lights the sensors can do that automatically if they detect darkness it can trigger the switch on and of accordingly.

In our car we did lots modification as we mentioned above. After removing previous limitation. We created a completely new car design. In which we added a small platform in front of car structure for object detection system. And cut a small rectangle shape in front platform where we can attach the servo motor. We also made a modified platform for Arduino which kept it in place as well as safe. We arranged the wires so that it didn't looked messy and arranged everything so that it also looks good from inside as it is from inside.

In this car we add a hood which open closed. And also, we added a LED headlight below the side of hood. In the middle of the car we left the place for batteries where we will put batteries and from power will be supplied to the components (Arduino, Motor Driver, Bluetooth module and etc.). In the back of the car we added a backlight. And also, we made place for Arduino. To remove the Arduino, we have to slide it to back, and to slide back we need remove back panel of car where back LED is attached. We also made a place for Motor Driver in center of the front tyre where it goes inside the car (base of car). After completing all this thing, we made a roof for a car which will cover the car top. It covers the bunch of wire which connected one component to another component which looks untidy from top.

This is how we did all modification in our car. Along with car modification we were modifying the code as well.

Fig. 5.15 3D Car Model

5.5 TEST CASES

Test Case ID	Test Case	Pre-conditions	Expected	Actual	Result (Pass/ Fail)
TC001	Testing Arduino	Buying Arduino UNO	Arduino must not be damage	Arduino was not damaged	Pass
		Uploading Code in Arduino	The Code must be Uploaded	The Code Uploaded successfully	Pass
		Testing all the ports	All the ports should work properly	All ports are working perfectly	Pass
TC002	Testing Bluetooth Module	Buying Bluetooth Module	Bluetooth Module must not be damaged	Bluetooth Module was not damaged	Pass
		There must be password for Bluetooth Module	Bluetooth must ask password when pairing for first time	Bluetooth is connected properly	Pass
		Connecting Bluetooth to Arduino	Bluetooth should be connected	Bluetooth connected perfectly	Pass
		Connecting Bluetooth Module with mobile Bluetooth	Bluetooth should pair mobile Bluetooth	Bluetooth Module connected with mobile Bluetooth	Pass
TC003	Testing Motor Driver	Buying Motor Driver	Motor Driver must not be damaged	Motor Driver was not damaged	Pass
		Ports of Motor Driver	All ports of Motor Driver must work	All ports of Motor Driver is working	Pass

		Power supply to Motor Driver	Power supply to Motor Driver must be constant	Power supply to Motor Driver is constant	Pass
TC004	Testing Ultrasonic	Buying Ultrasonic	Ultrasonic must not be damaged	Ultrasonic was not damaged	Pass
		Sensor must detect objects	Sensor must detect object within the given range	Sensor detects object	Pass
TC005	Testing Servo Motor	Buying Servo Motor	Servo Motor must not be damaged	Servo Motor was not damaged	Pass
		Rotation of Servo Motor shaft	Shaft of Servo Motor must Rotate 180 degree	Shaft of Servo Motor Rotates 180 degree	Pass
TC006	Testing LED	Buying LED	LED must not be damaged	LED was not damaged	Pass
		Powering LED without resistor	LED must work	LED pigments burns	Fail
		Powering LED with resistor	LED must work	LED worked perfectly	Pass
TC007	Testing LDR	Buying LDR	LDR must not be damaged	LDR was not damaged	Pass
		Detection of light	LDR must detect light if there is light	LDR detects the light	Pass

TC008	Car testing with Aluminum body	Movement of car	The car must move in forward, backward, left and right direction	The car moves in forward, backward, left and right direction	Pass
		Placing the equipment's	All the components should be placed on the platform	All the components are placed on the platform	Pass
		Fitting the components in place	The components should be placed in one fixed position	The components were not placed in one fixed position	Fail
		Car hood	There must be hood to protect the component	Making a hood was not possible	Fail
		Height of car	Aluminum body must be on proper height	Height was perfect	Pass
		Place for servo motor	There must be enough space for servomotor to rotate ultra-sonic sensor	There was no space for servomotor to rotate ultra-sonic sensor	Fail
		Placing LED's	There must be place for LED's	There was no place for LED's	Fail
		Wiring	Wires must be properly place	Wires were not properly placed	Fail

TC009	Car testing with single corrugated cardboard sheet	Movement of car	The car must move in forward, backward, left and right direction	The car moves in forward, backward, left and right direction	Pass
		Placing the equipment's	All the components should be placed on the platform	All the components are placed on the platform	Pass
		Fitting the components in place	The components should be placed in one fixed position	The components were placed in one fixed position	Pass
		Car hood	There must be hood to protect the component	Making a hood was not possible	Fail
		Height of car	Car body must be on proper height	Height was too big	Fail
		Place for servo motor	There must be enough space for servomotor to rotate ultra-sonic sensor	There was not enough space for servomotor to rotate ultra-sonic sensor	Partially Passed
		Placing LED's	There must be place for LED's	There was no place for LED's	Fail
		Wiring	Wires must be properly placed	Wires were not properly placed	Fail

TC010	Car testing with double corrugated cardboard sheet	Movement of car	The car must move in forward, backward, left and right direction	The car moves in forward, backward, left and right direction	Pass
		Placing the equipment's	All the components should be placed on the platform	All the components are placed on the platform	Pass
		Fitting the components in place	The components should be placed in one fixed position	The components were placed in one fixed position	Pass
		Car hood	There must be hood to protect the component	Making a hood was not possible	Pass
		Height of car	Car body must be on proper height	Height was too big	Pass
		Place for servo motor	There must be enough space for servomotor to rotate ultra-sonic sensor	There was not enough space for servomotor to rotate ultra-sonic sensor	Pass
		Placing LED's	There must be place for LED's	There sufficient was place for LED's	Pass
		Wiring	Wires must be properly placed	Wires were properly placed	Pass

CHAPTER 6: RESULTS AND DISCUSSION

6.1 TEST REPORTS

1. Purpose

The purpose of this document is to test the various activities that can be performed using Voice Control System.

2. Overview

Voice Control System is a very popular method these days for controlling a thing with just giving voice commands. If any individual wants to do a particular work he/she can just give voice command through his/her mobile phone which will be connected to that device through the internet. In some case it's not even necessary to have a phone if the devices come with preinstalled mic. This concept is mainly related to Internet of things and has a huge future scope. Another important thing that I must mention is using the sensors this are also an important part of IoT we have used sensors to sense various factors such as light objects. Using a voice control system will save a lot of our time and hard work that we can spend in doing other things. Controlling something on voice not only just it helps us but it also don't let us feel alone.

3. Testing Scope

The scope of the Voice Control System is to improve the technology used to make a better system. And the goal was to maintain an end to end connection with the car using only voice commands. We also made the code short and simple so as it could be understood by anyone. We used functions in our program so that we don't have to write the same thing again and again.

i. Scope:

Functional Testing for the following modules is in Scope of Testing

- Testing Motor Driver
- Testing Sensor
- Data transfer through Bluetooth
- Working Motors
- Connection with other devices

ii. Out of scope

Voice Recognition was not added in the system.

4. Metrics

This will help to understand the test execution results, status of test cases and defects etc. Following are matrices of testing Voice Control Car

a) **No. of test cases planned vs executed**
b) **No. of test cases passed/failed**

Test cases planned	Test cases Executed	Test Cases Passed	Test Cases Failed
10	10	7	3

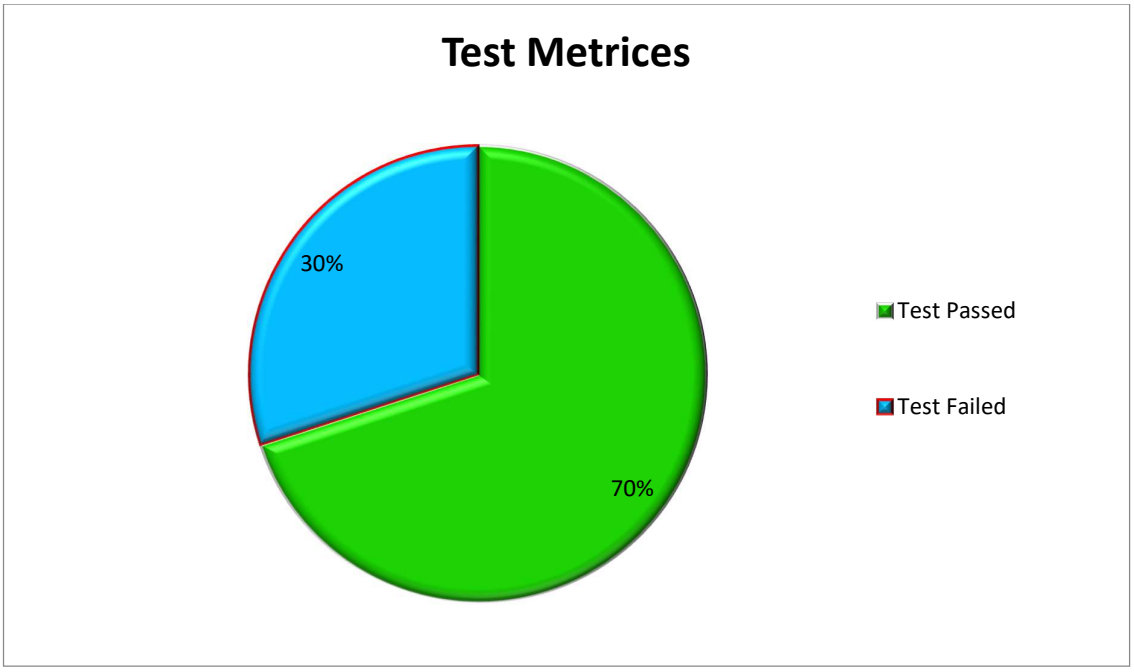


Fig. 6.1 Test Metrics

30% of our test failed which we later on changed as any failure won't give the correct output. We changed the body of the car twice since it was important for it to be placed in a rigid body which can hold everything together. We also failed in testing the Led where the LED burnt since it was getting extra voltage so to overcome that we used Resistors which reduces the current that goes in any device.

5. Types of testing performed

a) Unit Testing

In unit testing, individual components and units are tested to ensure that they work as an individual as defined in design. We have selected each of the module individually to ensure the functionality of it.

We have tested following module as unit testing –

- Arduino Uno
- Bluetooth HC-05
- Motor Driver L298n
- BO Motor
- Servo Motor
- LED lights

b) System Integration Testing

Integration testing involves integration of units to make a module / integration or modules to make system integration.We have selected Combination of modules to test the flow and working of module when they are integrated with other modules.

Following are the combinations of components –

- Register, Login, Logout
- Testing Arduino with a simple LED code
- Testing Bluetooth Module HC-05
- Testing LDR sensor
- Testing Ultrasonic sensor
- Testing Motor Driver and Motors
- Testing Servo Motor

c) Beta Testing

Beta testing is a part of user acceptance testing and done by the end user to validate the usability, functionality, compatibility and reliability of system. So for Beta testing we integrated the whole system together and tested it with commands like “go”, “back”, “left” and “right”.

6.2 USER DOCUMENTATION

User documentation for Car

- **GO command**

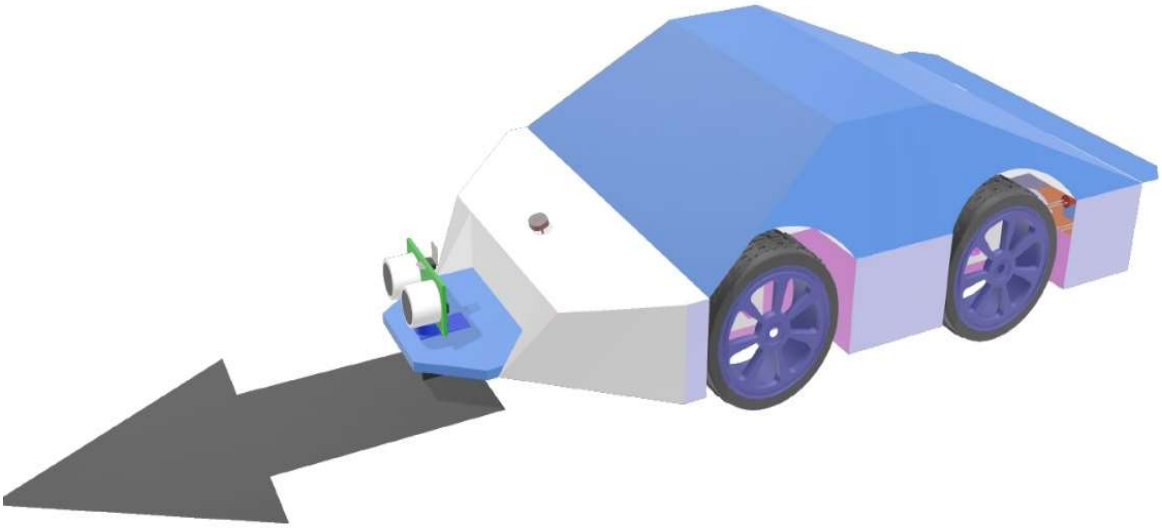


Fig. 6.2 Car Moving Ahead

The “GO” command in the program moves the car in forward direction. Once the user gives this command the car starts the move in forward direction. There is not any specific speed that it runs on as a result the speed of the car depends on the battery that supplies the voltage to the motors if the voltage is high the car will run faster or else it will slow down.

To stop the car, we just need to give the STOP command after giving this command the car stops instantly while stopping it glows the red light. If any obstacle comes in front of the car the ultrasonic sensor is always active and it detects if anything in front of the car if that object is within 20 cm radius of the car the car will automatically stop with the red light glowing in the back.

Fig. 6.3 GO

- **BACK command**

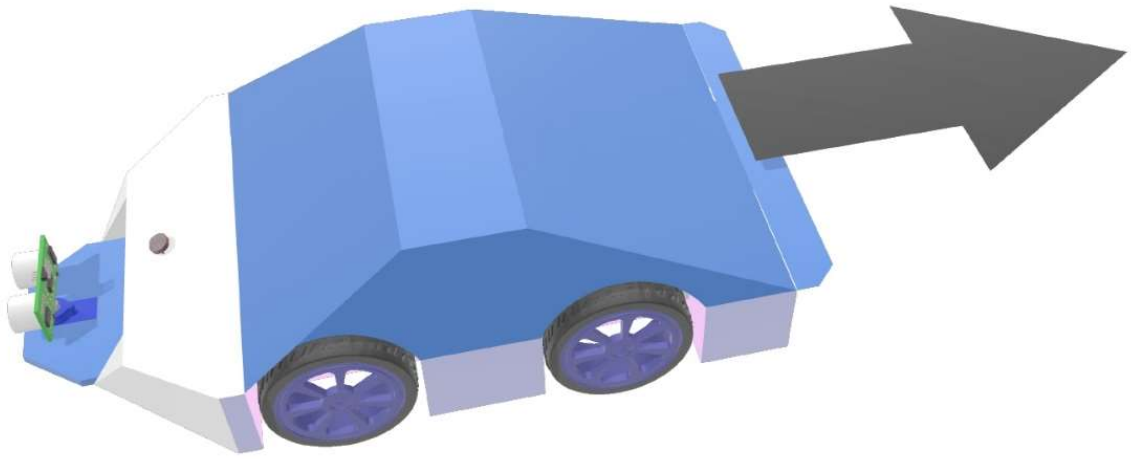


Fig. 6.4 Car Moving Backward

The “BACK” command in the program moves the car in backward direction. Once the user gives this command the car starts the move in backward direction. There is not any specific speed that it runs on as a result the speed of the car depends on the battery that supplies the voltage to the motors if the voltage is high the car will run faster or else it will slow down.

To stop the car, we just need to give the STOP command after giving this command the car stops instantly while stopping it glows the red light. If any obstacle comes in the back of the car the ultrasonic sensor won’t detects it because the servo can’t rotate 360^{o} so the ultrasonic sensor won’t be able to detect what is at its back while moving backward.

Fig. 6.5 Back

- **LEFT command**

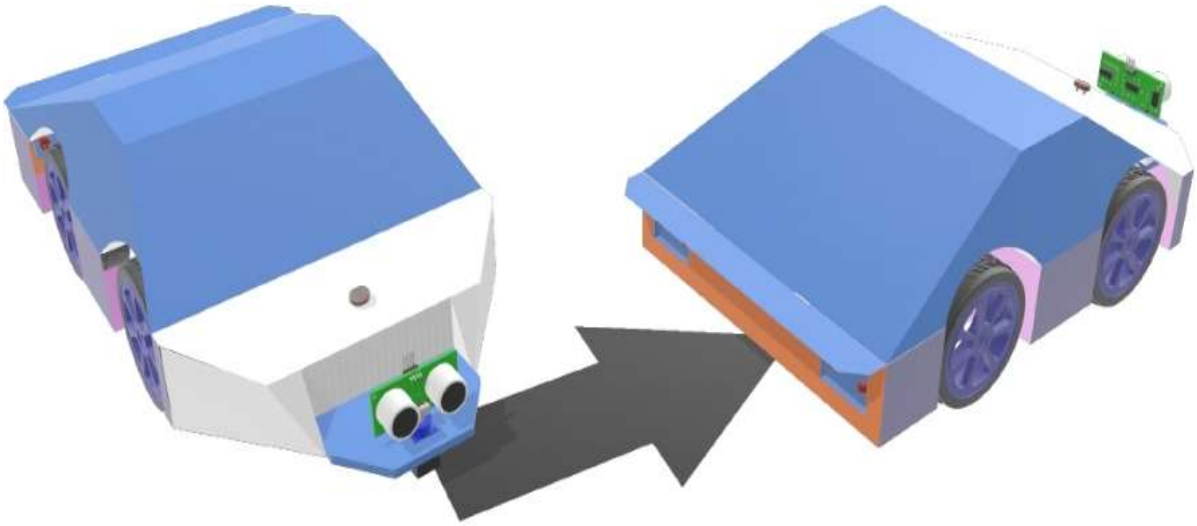


Fig. 6.6 Car Moving Left

The “LEFT” command in the program moves the car in left direction. Once the user gives this command the car moves towards left direction. In this process right side tyres move in forward direction where as the left side tyres move in backward direction this helps the car in moving in the left direction.

In this process first left led glows indicating that the car will move in the left direction servo motor turns 90^{o} left which makes the ultrasonic sensor on the servo motor move towards left direction for checking any obstacle in left direction. If there is any object between 20 cm radius of the car will not move in left direction instead it will on the red back light showing that there is obstacle in the left direction. If there is nothing within 20cm radius the car will turn left.

Ultrasonic sensor helps the device in changing its direction which is placed on servo motor as servo motor can turn only in 180^{o} so it is initially placed at 90^{o} when the car has to turn left the servo moves towards 180^{o} that is to the left side this makes the servo attached to it move towards the left direction to check if there is any obstacle or not before turning.

This is how the car takes left turn.

- **RIGHT command**

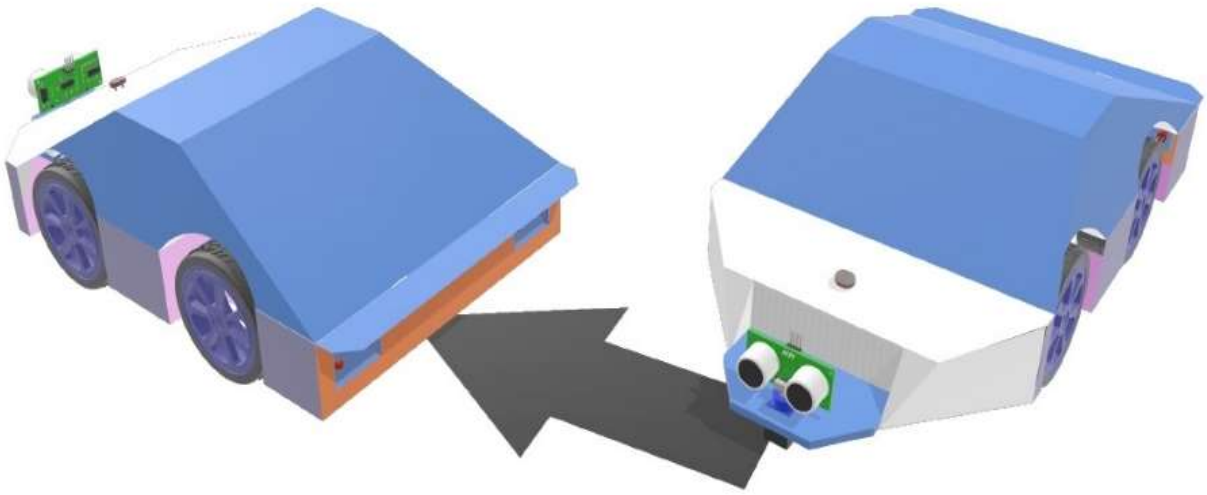


Fig. 6.7 Car Moving Right

The “RIGHT” command in the program moves the car in right direction. Once the user gives this command the car moves towards right direction. In this process left side tyres move in forward direction whereas the right-side tyres move in backward direction this helps the car in moving in the right direction.

In this process first right led glows indicating that the car will move in the right direction servo motor turns 90^{o} right which makes the ultrasonic sensor on the servo motor move towards right direction for checking any obstacle in right direction. If there is any object between 20 cm radius of the car will not move in right direction instead it will on the red back light showing that there is obstacle in the right direction. If there is nothing within 20cm radius the car will turn right.

Ultrasonic sensor helps the device in changing its direction which is placed on servo motor as servo motor can turn only in 180^{o} so it is initially placed at 90^{o} when the car has to turn right the servo moves towards 0^{o} that is to the right side this makes the servo attached to it move towards the right direction to check if there is any obstacle or not before turning.

This is how the car takes right turn.

- **System check command**

  The “System Check” command in the program is used to check all the components are working properly or not. After giving this command the car first moves little forward and little back to check motors. And then to check servo it turns the servo left and right. To check LED it starts each LED one by one and closes and at the end it blinks all LED for 5 times.

- **Round command**

  The “ROUND” command in the program moves the car in a circular direction. Once the user gives this command the car moves in a circular direction. In this process left side tyres move in forward direction whereas the right-side tyres don’t move in any direction this helps the car in moving in circular direction keeping one side moving and the other side stopped

- **Keep watch command**

  The “Keep Watch” command in the program moves the car in straight direction. While the car is moving in straight direction it checks for any obstacles with the help of ultrasonic sensors that come in between if any obstacles comes the car stops in comes a little back then the servo first turns left checking for any obstacles then turns right for checking obstacles with the ultrasonic sensor on it the direction where there is no obstacle the car starts moving in that direction. And the process continues until stop command is given.

- **stop command**

  The “STOP” command in the program simply stops the car moving. Once the user gives this command the car stops and the red LED glows. This command is used to stop the car which is performing any given commands.

- **Glow lights command**

  The “Glow lights” command in the program Glows the LED in the car. In the car there are a total of 12 LED in which 6 are white in the front 3 on each side (i.e. Left and Right) there are two orange LED one on each side they are used while changing direction. And there are 4 Red LED at the back two on each side. This are used to denote that the car has stopped. Glow lights glows all the LED to close those use the “Close Lights” command.

CHAPTER 7: CONCLUSIONS

7.1 CONCLUSION

In this rapidly growing world, everyone is becoming so busy that they find shortcuts in everything they can so whether it is sending a message to friend or finding the shortest way to reach home. This not only saves their time but also it saves lot of efforts he/she would have invested in it. So off course who likes typing those long documents on the pc whole day when you can do that just with your voice command one can use Speechnotes website just to do that. One can just ask google to search for an information by saying "Ok Google" and it will listen to you There are various other system that can-do similar things this system is improving day by day and showing lot more ideas and can change the way we do things.

Voice Control System has a big potential in becoming an important factor of interaction between human and machine in the near future. A speaker independent Voice Control system has been proposed to combine the advantages of the dictation ability and easy of controlling devices. The Voice Control System is more popular now a day. This Voice Control System is used in many sectors like industrial sector, business sector, and also used in home automation system. The Future Scope of the Voce Control System is very good. This technology is Beneficial for public and private sectors of business community. The technology is easy and safe to use by children's and adults.

The results show reasonably good success in recognizing continuous speech from various speakers, for a large vocabulary. The different modules were analysed in their respective order and were successfully verified for different speech input files. We designed the 2 stages of speech recognition - the general processing and Text Recognition in software. The speaker independent speech recognition systems were successfully trained to recognize speech 101 inputs that were recorded using a microphone as well as speech samples obtained from various platforms. Key research challenges for the future are acoustic robustness, use of multiple word pronunciations and efficient constraints for the access of a very large lexicon and well-organized methods for extracting conceptual representations from word hypotheses. The recognition system presented in our proposed method performs stage mode speaker independent speech recognition. The obtained results can be improved by fine tuning the system with larger training databases. The next step would be to recognize live speech, which would require more resources including larger speech databases, acoustic models and exhaustive vocabularies to produce good recognition results.

IT and control systems manufacturers are seizing the opportunity of having new novel hardware devices as the "Internet of Things" begins to scale up in the market. As the number of devices used continues to increase, more and more automation will be required for both the consumer (e.g. home and car) as well as industrial environments. The main benefits of autonomous capabilities in the future IoT is to extend and complement human performance. Robotic manufacturing and medical nanobots may be useful.

The IoT has the potential to dramatically increase the availability of information, and is likely to transform companies and organizations in virtually every industry around the world. As such, finding ways to leverage the power of the IoT is expected to factor into the strategic objectives of most technology companies, regardless of their industry focus. The number of different technologies required to support the deployment and further growth of the IoT places a premium on interoperability, and has resulted in widespread efforts to develop standards and technical specifications that support seamless communication between IoT devices and components. Collaboration between various standards development groups and consolidation of some current efforts will eventually result in greater clarity for IoT technology companies.

7.1.1 Significance of the System

Voice or speaker recognition is the ability of a machine or program to receive and interpret dictation or to understand and carry out spoken commands. Voice recognition has gained prominence and use with the rise of AI and intelligent assistants, such as Amazon's Alexa, Apple's Siri and Microsoft's Cortana.

Voice recognition systems enable consumers to interact with technology simply by speaking to it, enabling hands-free requests, reminders and other simple tasks.

How voice recognition works:

Voice recognition software on computers requires that analog audio be converted into digital signals, known as analog-to-digital conversion. For a computer to decipher a signal, it must have a digital database, or vocabulary, of words or syllables, as well as a speedy means for comparing this data to signals. The speech patterns are stored on the hard drive and loaded into memory when the program is run. A comparator checks these stored patterns against the output of the A/D converter -- an action called pattern recognition.

In practice, the size of a voice recognition program's effective vocabulary is directly related to the random-access memory capacity of the computer in which it is installed. A voice recognition program runs many times faster if the entire vocabulary can be loaded into RAM, as compared with searching the hard drive for some of the matches. Processing speed is critical, as well, because it affects how fast the computer can search the RAM for matches.

Voice recognition systems, such as Amazon Alexa, serve as digital assistants that respond to simple user requests.

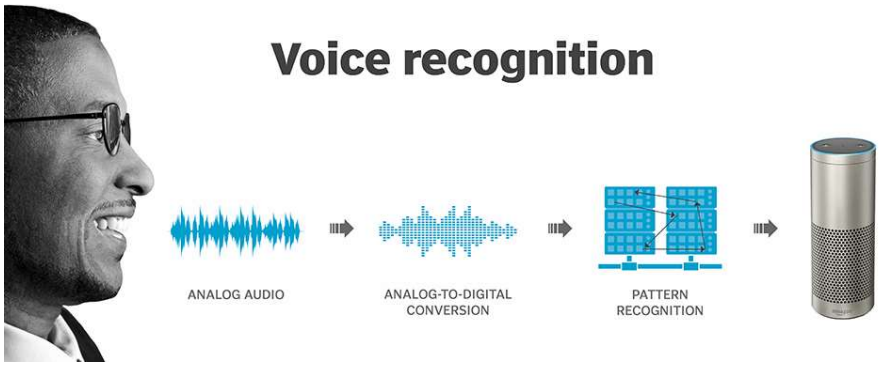


Fig. 7.1 Voice Recognition

Some of the most popular voice recognition systems function as virtual assistants to answer questions about weather or perform simple tasks, such as adding items to an online shopping cart.

While voice recognition technology originated on PCs, it has gained acceptance in both business and consumer spaces on mobile devices and in-home assistant products. The popularity of smartphones opened up the opportunity to add voice recognition technology into consumer pockets, while home devices, like Google Home and Amazon Echo, brought voice recognition technology into living rooms and kitchens. Voice recognition, combined with the growing stable of internet of things sensors, has added a technological layer to many consumer products that previously lacked any smart capabilities.

As uses for voice recognition technology grow and more users interact with it, the companies implementing voice recognition software will have more data and information to feed into the neural networks that power voice recognition systems, thus improving the capabilities and accuracy of the voice recognition products.

Voice recognition uses:

The uses for voice recognition have grown quickly as AI, machine learning and consumer acceptance have matured. In-home digital assistants from Google to Amazon to Apple have all implemented voice recognition software to interact with users. The way consumers use voice recognition technology varies depending on the product, but it can include transcribing voice to text, setting up reminders, searching the internet, and responding to simple questions and requests, such as playing music or sharing weather or traffic information.

The government is also looking for ways to use voice recognition technology for security purposes. The National Security Agency has used voice recognition systems dating back to 2004.

7.2 LIMITATIONS OF THE SYSTEM

Voice recognition software turns speech into text you talk into the system and it transcribes what you say. This is useful for people with visual impairments and those with physical problems that make typing on a keyboard difficult. Others may use a system because they find talking easier than typing or simply because it's fun. Voice recognition software can certainly be attractive as it enables you to retain files in-house rather than sending them out to a professional transcription service.

Programs cannot understand the context of language the way that humans can, leading to errors that are often due to misinterpretation. Homonyms, complex deixis, and even complete omission of entire words or phrases seldom prevent us from understanding one another. While modern AI-powered voice-control systems are much better than the technology from 10 years ago, true natural communication with real-time feedback is still impossible.

With errors also comes the necessity to invest more time to correct them. This can turn a quick Google search into a minute-long order, which isn’t all that bad unless you add up how much extra time it takes you to get things done over a long period of time.

Another major disadvantage of voice control over graphical user interfaces is background noise interference. For voice control systems to work properly, you need to be in a quiet environment, undisturbed by ambient noise and people talking. Such conditions may not always be possible to achieve, although headphones with noise-cancelling microphones do help to some extent.

Voice recognition technology is not perfect, however, and comes with a few disadvantages.

- **Lack of Accuracy and Misinterpretation**

Voice recognition software won't always put your words on the screen completely accurately. Programs cannot understand the context of language the way that humans can, leading to errors that are often due to misinterpretation. When you talk to people, they decode what you say and give it a meaning. Voice recognition software can do this but may not be capable of choosing the correct meaning. For example, it cannot always differentiate between homonyms, such as "their" and "there." It may also have problems with slang, technical words and acronyms.

- **Time Costs and Productivity**

You might think that computerizing a process speeds it up, but this isn't necessarily true of voice recognition systems, and you may have to invest more time than you expected into the process. You'll have to factor in time to review and edit to correct errors. Some programs adapt to your voice and speech patterns over time; this may slow down your workflow until the program is up to speed. You'll also have to learn how to use the system. For example, you must find the right pace and tone -- if you talk too fast or indistinctly, you'll increase spelling and grammar errors. Getting used to using a system's commands and speaking punctuation out loud is not always easy. This can affect the flow and speed of your speech.

- **Accents and Speech Recognition**

Voice recognition systems can have problems with accents. Even though some may learn to decode your speech over time, you have to learn to talk consistently and clearly at all times to minimize errors. If you mumble, talk too fast or run words into each other, the software will not always be able to cope. Programs may also have problems recognizing speech as normal if your voice changes, say when you have a cold, cough, sinus or throat problem.

- **Background Noise Interference**

To get the best out of voice recognition software, you need a quiet environment. Systems don't work so well if there is a lot of background noise. They may not be able to differentiate between your speech, other people talking and other ambient noise, leading to transcription mix-ups and errors. This can cause problems if you work in a busy office or noisy environment. Wearing close-talking microphones or noise-cancelling headsets can help the system focus on your speech.

- **Physical Side Effects**

If you use voice recognition technology frequently, you may experience some physical discomfort and vocal problems. Talking for extended periods can cause hoarseness, dry mouth, muscle fatigue, temporary loss of voice and vocal strain. The fact that you aren't talking naturally may make this worse and you may need to learn how to protect your voice if you'll use a program regularly.

7.3 FUTURE SCOPE OF THE PROJECT

- **Voice Control Technology**

We are still some distance from realizing the true potential of speech recognition technology. This applies both to the sophistication of the technology itself and to its integration into our lives. The current digital assistants can interpret speech very well, but they are not the conversational interfaces that the technology providers want them to be. Moreover, speech recognition remains limited to a small number of products.

The rate of progress, compared to the earliest forays into speech recognition, is really quiet phenomenal nonetheless.

As such, we can look into the near future and envisage a vastly changed way of interacting with the world around us. Amazon's concept of "ambient computing" seems quite fitting.

The smart speaker market has significant room left to grow, with 75% of US homes projected to have at least one by the end of 2020.

Now that users are getting over the initial awkwardness of speaking to their devices, the idea of telling Alexa to boil the kettle or make an espresso does not seem so alien.

Voice is becoming an interface of its own, moving beyond the smartphone to the home and soon, to many other quotidian contexts.

We should expect to see more complex input-output relationships as the technology advances, too. Voice-voice relationships restrict the potential of the response, but innovations like the Amazon Echo Show and Google's support for smart displays will open up a host of new opportunities for engagement. Apple and Google will also incorporate their AR and VR applications when the consumer appetite reaches the required level.

Challenges remain, however. First of all, voice search providers need to figure out a way to provide choice through a medium that lends itself best to short responses. Otherwise, how would it be possible to ensure that a user is getting the best response to their query, rather than the response with the highest ad budget behind it?

Modern consumers are savvy and have access to almost endless information, so any misjudgements from brands will be documented and shared with the user's network.

A new study from Google has shown that there is an increasing acceptance among consumers that brands will use smart speakers to communicate with them. A sizeable number revealed a willingness to receive information about deals and sales, with almost half wanting to receive personalized tips:

Fig. 7.2 Most voice search

- **Sensors**

Sensors have made serious inroads into automotive, medical, industrial, and aerospace applications. Rising concerns for safety, convenience, entertainment, and efficiency factors, coupled with worldwide government mandates, will see sensor usage swell to unprecedented levels. Add to that the predicted explosion in wireless and consumer applications, and one can see why sensor manufacturers anticipate quickly developing huge markets and applications through the end of this decade. Most of these sensors will be of the microelectromechanical-system (MEMS) and microsystem-technology (MST) type, with nano-sensors showing great promise.

Mention automotive systems, and sensor manufacturers can easily see a host of sensing possibilities for measuring not only pressure, but also inertia, position, proximity, temperature, flow rate, force, strain, torque, vibration, and tilt. And the sensing technologies used to measure these parameters are just as varied. According to Alex Cade, a technical fellow at the General Motors Technology Centre, "sensing needs for automobiles are growing by leaps and bounds." He cited several growth areas for chassis controls, vehicle positioning/location, object detection, vision enhancement, auto environment heating, ventilation, and air conditioning, as well as engine and transmission controls. Vehicle stability enhancement was just one of the many examples he cited.

The future of the role of sensor technology in the world is limitless. Several research projects are currently looking at various ways to manipulate this technology for future purposes such as enhancing the performance of energy sources like solar-powered batteries and fuel cells, improving the capability of these systems to monitor health, safety and security, as well as a potential use for environmental monitoring.

Furthermore, futuristic intelligent systems are looking towards incorporating sensor technology to recognize visual, tactile, olfactory, audiological and physical senses. Sensors are capable of carrying out each of these actions through its transducer ability, which relies upon the application of electrical or optical signals to induce a response of the system. Through this conversion of various types of physical phenomena into measurable data, sensors often exhibit accurate, reliable, efficient and robust capabilities that allow for this technology to be used in an increasing number of industries around the world.

The application of sensor technology has been found in almost every aspect of our daily lives, including safety, security, monitoring, awareness and surveillance products. As a technology that can be utilized in both micro- and nano-settings, sensor technology systems have found a particularly useful application within the field of medicine as a diagnostics and drug monitoring tool that has enhanced critical care for patients.

To quantify the global market for sensor technology, BCC Research recently conducted a report of the current and future applications of sensor technology around the world. The main purpose of this study was to understand the rate of growth that sensor technology is expected to have, while also providing prospective manufacturers, distributors and users of this technology information on the potential opportunities sensor technology may have for their future endeavours.

Once the preliminary research regarding the function of sensors and the history of the applications of this technology since its introduction into society was conducted, the overall market size was quantified and analysed. Revenue estimates from the total market was achieved by collecting data from engineers, marketing executives, product sales engineers, international sales managers and other necessary personnel of sensor companies. From this gathered information, it has been estimated that the global market for sensors will rise from $110.4 billion in 2015 to an anticipated $240.3 billion in 2022. Cities around the world such as Montreal, Quebec in Canada, San Jose, California in the United States and Wuhan, China are already obtaining the benefits of this type of smart sensor system.

Smart sensor technology is looking to revolutionize the future of the world within the next few years. As compared to a typical sensor, smart sensors are equipped with a microprocessor that is capable of customizing outputs and providing interpretive data that significantly improves the capability and performance of the sensor system. By ensuring the integrity and reliability of the results created by the smart sensor, its application in medicine, consumer products, communication devices, home sensors, transportation and military is inevitable.

One example in how smart sensor technology is already advancing the field of diagnostic medicine is clear in the PillCam Capsule Endoscopy; a device that has been developed by Israeli company Given Imaging that is changing the way in which the human intestinal tract is studied. The PillCam is an ingestible pill-sized camera that allows physicians to visualize the oesophagus, small bowel and colon, in order to avoid the need for sedation and as invasive endoscopic procedures that are traditionally used for these investigative purposes

Within the communications industry, current smartphones are equipped with several sensors such as a microphone, camera, digital compass, accelerometer and gyroscope; however, the future of smartphones is expected to contain even more sensors. Of these future innovative sensors include barometers and accelerometers that can be used to study the activity in which the user of the smartphone is doing. In fact, some researchers are studying how specific sensors can be applied to smartphones in order to detect the ambient pollution present in their surrounding environment[4]. Researchers estimate that by the year 2050, smart sensors and similar sensor-based systems will form the foundation of most of the work performed by the world's ever-growing economy.

The future of both traditional and smart sensor applications to be implemented around the world is unimaginable. Current research trends are anticipating that, aside from the previously stated industries that have already found use in sensor technology, "smart cities" are also on the rise. Defined as cities that are either in the process of or have already employed various information, communication and technology solutions to be integrated across three or more functional areas of the city, smart cities offer a resolution to the massive global urbanization that is already taking place.

A United Nations report has recently postulated that by the year 2050, 66% of the world's population is estimated to live in cities, which will also lead to an overall population growth of another 2.5 billion people. One way to handle such an exponential growth is through the implementation of smart cities, in which the ability to monitor the activity of millions of people to adequately manage and improve the efficiency of the city's assets and services can meet the population needs at an impressive rate.

REFERENCES

Books and Authors

- Designing the Internet of Things
  -By Adrian McEwen & Hakim Cassimally
- Learning Internet of Things
  -By Peter Waher
- Programming Arduino: Getting Started with Sketches
  -By Simon Monk

Reference Websites

- https://create.arduino.cc/projecthub/Yug_Ajmera/voice-controlled-car-983ef1
- https://www.arduino.cc/en/Guide/Introduction
- https://www.arduino.cc/en/Main/Products
- https://www.arduino.cc/en/Main/Software
- https://create.arduino.cc/projecthub/andriy-baranov/from-bt-to-wifi-creating-wifi-controlled-arduino-robot-car-09b7c1?ref=tag&ref_id=car&offset=4
- https://create.arduino.cc/projecthub/samanfern/bluetooth-controlled-car-d5d9ca?ref=tag&ref_id=cars&offset=5
- https://www.youtube.com/watch?v=QYa02BYWD14&t=60s
- https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/
- http://wiki.sunfounder.cc/index.php?title=Motor_Driver_Module-L298N
- https://www.makerspaces.com/arduino-uno-tutorial-beginners/
- https://www.youtube.com/watch?v=1n_KjpMfVT0
- https://www.youtube.com/channel/UCTJ9288yPA-hkSWpuAzcr5g
- https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/
- https://www.elprocus.com/ultrasonic-detection-basics-application/
- https://components101.com/ultrasonic-sensor-working-pinout-datasheet
- https://www.youtube.com/watch?v=WslzsHDYuF0
- https://www.youtube.com/watch?v=6F1B_N6LuKw
- https://www.youtube.com/watch?v=OhnxU8xALtg
- https://www.youtube.com/watch?v=dZi9fM-X2jU

GLOSSARY

Implementation	:	The process of applying decision or plan
Resistor	:	A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element
Anode	:	Anode is an electrode through which the traditional current enters a polarized electrical device
Cathode	:	A cathode is an electrode from which a conventional current leave a polarized electrical device
Prototype	:	A first or preliminary version of a device or vehicle from which other forms are developed
Shunt	:	A shunt is a device which creates a low-resistance path for electric current.
Subroutine	:	Set of instructions designed to perform a frequently used operation within a program
Interface	:	A point where two systems meet & interact.
Efficiency	:	The comparison of what is actually produced or performed with what can be achieved with the same consumption of resources
Integration	:	To put together parts or elements and combine them into a whole.
Corrugated	:	Combination of parallel ridges & grooves that gives rigidity & strength to an object.
Recognition	:	Identification of someone or something or person from previous encounters or knowledge
Interference	:	The action of interfering or the process of being interfered with a system.
Significance	:	A quality of being worthy of attention; importance.
Accent	:	A typical way of pronouncing the language, especially related to a particular country, region or social class