# CHAPTER 1
# INTRODUCTION

## 1.1 ABSTRACT

Now day’s we seen that the human being want’s the easier life and every time peoples trying to search the several types of result to break the any problems. Sometimes we're uses the machines that will reduce the sweats as well as needed time. So for that we're enforcing a prototype of a robotic vehicle which is Electric vehicle. According to the report the tradition vehicle contributes the 20- 30% of air pollution. Electric vehicle is eco-friendly[1]. Our proposed system works by using a Wi- Fi module for entering the Wi- Fi command being transferred by the driver. The system apply in this study uses DC motor to move the robotic vehicle to the applicable direction using Wi- Fi commands. This robotic car is solving the major problems which is occurs in traditional vehicles like packing, driving. It has capability to smell the terrain and decide the navigation path without any mortal input. So that, probability of accident is reduces. As we're controlling this robotic car using the android Smart- phone also the handicap people’s can drive this car[3].

The project involves the integration of various technologies, including robotics, mobile computing, wireless communication, and sensor systems. The robot car is equipped with a microcontroller, motor drivers, wheels, sensors, and a camera[2]. The software component of the project consists of two main parts: the Android application and the firmware running on the microcontroller. The Android application provides a user-friendly interface for controlling the car's movements, capturing images or videos through the camera, and receiving sensor data. It utilizes the smartphone's touch screen, accelerometer, and gyroscope to enable intuitive control of the robot car. The firmware on the microcontroller processes the commands received from the smartphone and controls the motors accordingly, enabling the car to move in different directions. Wireless communication plays a crucial role in this project, allowing the Android smartphone and the robot car to establish a connection. Bluetooth or Wi-Fi technology can be used for this purpose, providing a reliable and low-latency communication channel. The smartphone acts as a transmitter, sending control signals to the robot car, while the car acts as a receiver, processing these signals to perform the desired actions. The Android controlled robot car has a wide range of potential applications[4]

## 1.2 INTRODUCTION

The Android controlled robot car is a remarkable project that combines the power of modern technology and robotics. It is designed to be controlled using an Android device, providing a user-friendly interface and enhancing accessibility [2]. This project builds upon the advancements made in the field of robotics, enabling enthusiasts and hobbyists to explore the realm of autonomous vehicles. By integrating Android technology, users can remotely maneuver the robot car, monitor its surroundings, and even engage in real-time video streaming. This project represents a fusion of mobile computing and robotics, pushing the boundaries of innovation and paving the way for future advancements in the field of autonomous vehicles[3].

The concept of remote-controlled vehicles has been prevalent for many years, but the integration of Android technology with robotics has taken it to a whole new level. The Android Controlled Robot Car project builds upon the advancements in mobile computing and wireless communication, making it more accessible and user-friendly.

In the early stages of development, researchers and engineers explored various techniques for wireless communication and control protocols. They focused on optimizing the reliability and responsiveness of the system, allowing for real-time control and feedback. Through extensive testing and iterations, they refined the control algorithms and hardware components to ensure smooth operation and seamless integration with Android devices.

The project also involved developing a custom Android application that serves as the interface for controlling the robot car. This application provides a user-friendly dashboard with intuitive controls, allowing users to maneuver the car effortlessly. It includes features like forward, backward, left, and right movements, as well as the ability to stream live video from the car's onboard camera, providing a first-person perspective.Furthermore, the project team worked on enhancing the robot car's functionality by integrating additional sensors, such as obstacle detection and avoidance systems. These sensors enable the car to navigate autonomously, avoiding collisions with objects in its path.

A robot is a mechanical or virtual artificial agent, usually an electromechanical machine that is guided by a computer program or electronic circuitry. The first digital and programmable Robot was invented by George Devol in 1954 and was named the Unimate. [1]Apps control robot is one where the controlling is done by the smartphone apps using Bluetooth. It is possible to control of different parameters of many applications such as to control the speed, light, direction, sound and temperature. Nowadays smart phones are becoming more powerful with reinforced processors, larger storage capacities, richer entertainment function and more communication methods [2].Recently the Bluetooth technology has become the standard for device-to-device communications for short distance. Bluetooth is an open standard specification for a radio frequency (RF) - based, short-range connectivity technology that promises to change the face of computing and wireless communication. It is designed to be an inexpensive, wireless networking system for all classes of portable devices, such as laptops, PDAs (personal digital assistants), and mobile phones. The controlling device of the whole system is a microcontroller [3-4].

The rapid development of smart phone technology, especially the promotion and application of wireless technology, provides a platform and opportunity for some basic ideas and methods in the control theory to be applied to the car.[11] Automated smooth controlled cars are required for road safety of developing Bangladesh. Still, many traffic situations remain complex and difficult to manage, particularly in urban settings. The driving task belongs to a class of problems that depend on underlying systems for logical reasoning and dealing with uncertainty[12]. So, to move vehicle computers beyond monitoring and into tasks related to environment perception or driving, we must integrate aspects of human intelligence and behaviours so that vehicles can manage driving actuators in a way similar to humans[16]

## 1.3 LITERATURE SURVEY:

Intelligent Transport Systems (ITS) based on Internet of Things (IoT) are getting popular and can be seen as a solution to improve the road safety. One effective technique to reduce traffic hazards and save precious lives could be to reduce the response time after an accident has occurred[5].

Some systems focus on preventive strategy because at the end, goal is to save lives. This system particularly focuses on the safety of two wheelers and checks if the driver is drowsy.

Many of the researchers have worked to bring the automation in the automobile field.Few of them are summarised here.

- The authors have developed a system for the remote controlling of a vehicle using the 8031 microcontroller technology in which author are able to control the car using the android app.

- The authors have developed a system for the Smartphone control robots through bluetooth using the Bluetooth technology in which author are able to control using the bluetooth. In which the user used the Bluetooth module for that project which is used to control the through the Smartphone[2].

- The authors have developed the Bluetooth operated robot vehicle using mobile android app which is used to control the vehicle through the android phone using the android app The authors have developed the Bluetooth based android controlled robot using the bluetooth module[6].

- The authors publish the research paper in which the author is developed one system which is “The DLR lightweight robot: design and control concepts for robots in human environments” .Based on the design and control of robot[5].

- The authors have developed the Obstacle avoidance and Android mobile phone controlled Bluetooth robot using arduino. In which the Author is works on the Arduino board and ultra-Sonic components[2].

There are no. of authors work on this topic of robot which are shows in the below comparison table as following :-

**COMPARISON TABLE:**

**Table 1: Comparison of Existing Systems**

| Sr. No | Authors | Paper Title | Publisher | Year | Method Used |
|---|---|---|---|---|---|
| 1 | Dickmanns E | The development of machine vision for road vehicles in the last decade | IEEE | 2002 | Based on vehicle Control algorithm |
| 2. | Schaffer A A, Haddadin S, Ott Ch, Stemmer A, Wimbock T and Hirzinger G | The DLR lightweight robot: design and control concepts for robots in human environments | Industrial Robot : An Inter national Journal s | 2007 | Based on Design and Control of robot |
| 3. | Hebah H O Nasereddin and Abdelkarim A | Smart phone control robots through bluetooth | IJRRAS | 2010 | Based on Bluetooth |
| 4. | Zi-Yi, Lam, Sew-Kin, Wong, Wai- L eong , Pang , Chee-Pun, Ooi | The Design of DC Motor Driver for Solar Tracking Applications | IEEE | 2012 | Based on Micro-con troller DC-DC buck<br>Converter |

| | | | | | |
|---|---|---|---|---|---|
| 5. | Malik A,Shrivastava A,Singh GK,Shuk A | Remote controlling of a vehicle using android app | Int. J of Res.in Eng. & Adv. Tech. &Sci | 2015 | Based on 8051Microcontroller technology |
| 6. | Parmar D, Tripathi D, Sahni A, Singh P | Bluetooth operated robot vehicle using mobile android app | Int. J of Res.in Eng. &Adv Technol | 2015 | Based on Bluetooth |
| 7. | Ashima, Kumar R, Nikhil T, Singh P | Obstacle avoidance and Anderoid mobile phone controlled Bluetooth robot using arduino | IJEEE | 2015 | Based on Arduino and Ultra-sonic |
| 8. | Eshita R Z, Barua T, Barua A | Bluetooth based android controlled | American Journal of | 2016 | Based on Bluetooth |
| 9. | Gandotra S, Sharma B, Mahajan S, Motup T, Choudhary T and Thakur P | Bluetooth controlled RC car using Arduino | Imp.J of Interdisciplinary Research (IJIR) | 2016 | Based on Arduino Control |

| | | | | | |
|---|---|---|---|---|---|
| 10. | N Kumar , D Acharya and D Lohani | An Iot – based vehical accident detecting on classification on system using senor Fusion | IEEE | 2020 | Based on Iot |
| 11 | KL Narayanan and CRS Ram | IoT based smart accident detection & insurance claiming system | IEEE | 2021 | Based on GUI and Bluetooth |
| 12. | SR Prasath, RS Krishnan and SM Priya | IoT based Smart Accident Detection System for Hit and Run Cases | IEEE | 2022 | Based o n Arduino based control unit |

## 1.4 NEED OF PROJECT

- We know that, Electric vehicles use electricity to charge their batteries instead of using fossil fuels like petrol or diesel. Electric vehicles are more efficient, and that combined with the electricity cost means that charging an electric vehicle is cheaper than filling petrol or diesel for your travel requirements.
- In this project we controlling Electric car using wirelessly through Android smart phone using the Wi-fi module through UART protocol with the Robotic mechanism[16].So, user can control the Electric car from anywhere within the rang of controlled

## 1.5 AIM OF THE PROJECT

- The aim of this project to designing a ROBOTI car that can be operated wirelessly through Wi-fi communication using Android Apps on smart phone.
- To avoid the vehicle/car accident by using the different-different sensor models.

## 1.6 OBJECTIVES OF THE PROJECT

1. To control the car with help of Android phone.
2. To Drive the car/vehicle safely and with the security.
3. To Avoid the Accidental cases.
4. To Maintain the Environmental balance(with ECO-friendly).

## 1.7 PLANNING

**Table 2: Planning**

| **Sr. no.** | **Month** | **Task** |
|---|---|---|
| 1 | July 2022 | • Formed the group.<br>• Did the survey on problems related to renewable energy sources.<br>• Found out the problems faced by the people.<br>• Discussed different ideas with Guide related to Agriculture, Robotics, and Embedded.<br>• We submitted 3 project ideas.<br>1) Android Controlled Robot Car<br>2) IOT based Green House Farming.<br>3) Women safety with GPS tracking and alerts using arduino. |
| 2. | August 2022 | • Given the presentation on the above three project topics<br>• Final topic was selected : Android Controlled Robot Car.<br>• Gave presentation on final topic.<br>• Suggestions are given by the teachers. |
| 3. | September 2022 | • Literature survey<br>• Block diagram implementation<br>• Finalisation of components, downloading of datasheet of each component used for the project. |
| 4. | October 2022 | • Circuit diagram design.<br>• Designed the Flowchart. |

| | | |
|---|---|---|
| 5. | November 2022 | • Simulation on TinkerCAD simulation software.<br>• To find the solutions for generated problems like how to avoid the obstacle in front of car, controlling issue of the caret<br>• Layout |
| 6. | December 2022 | • Preparation of Synopsis report. |
| 7. | January 2023 | • Testing |
| 8. | February 2023 | • Real time Programming<br>• Faults finding |
| 9. | March 2023 | • Troubleshooting and modification if necessary |
| 10. | April 2023 | • Preparation of Report |

# CHAPTER 2
# HARDWARE DESIGN

## 2.1 INTRODUCTION:

### I. Robotic Car Hardware:

- **Chassis:** The physical structure of the car that houses the motors, wheels, and other components.
- **Motor Control**: Motor drivers or controllers that regulate the speed and direction of the car's motors.
- **Sensors:** Various sensors such as proximity sensors, ultrasonic sensors and gas sensor to detect gases the car's surroundings.
- **Microcontroller/Controller Board:** A microcontroller or controller board that acts as the brain of the robotic car, receiving commands from the Android app and controlling the car's hardware components.

### II. Android Application:

- **User Interface:** The Android app provides a user-friendly interface through which users can control the robotic car and access various functionalities
- **Wireless Communication:** The app establishes a wireless connection (e.g., Bluetooth or Wi-Fi) with the robotic car to send control commands and receive feedback.
- **Control Commands:** Users can send commands to the robotic car via the app to control its movements, speed, and other actions.
- **Feedback and Visualization**: The app receives feedback from the car's sensors, such as obstacle detection information or video streaming, and presents it to the user through visual elements.

## 2.2 BLOCK DIAGRAM:

Following block diagram show the actual working of the android controlled robot car. We can control the robot car by the help of android phone. As the connection shows in fig. The android phone is connected to the microcontroller by the help of wireless module to send the instruction to the microcontroller to control the speed and direction of car.

And the microcontroller is connected to the motor driver ic and then motor driver IC is connected to the motor.

This block represents the physical robot car that performs various actions based on the commands received from the Android device. It consists of a microcontroller or a dedicated control board, motor drivers, and various sensors or actuators.

Motor Driver IC: The Motor Driver IC controls the movement of the robot car's motors. It receives control signals from the control system and regulates the motor speed and direction accordingly. The Motor Driver IC is typically used to interface with the motor drivers or directly drive the motors.

The power supply block provides electrical power to all the components of the system, including the ESP8266 module, control system, motor driver IC, ultrasonic sensor, and other peripherals. It ensures a stable and regulated power output to ensure reliable operation of the entire system.

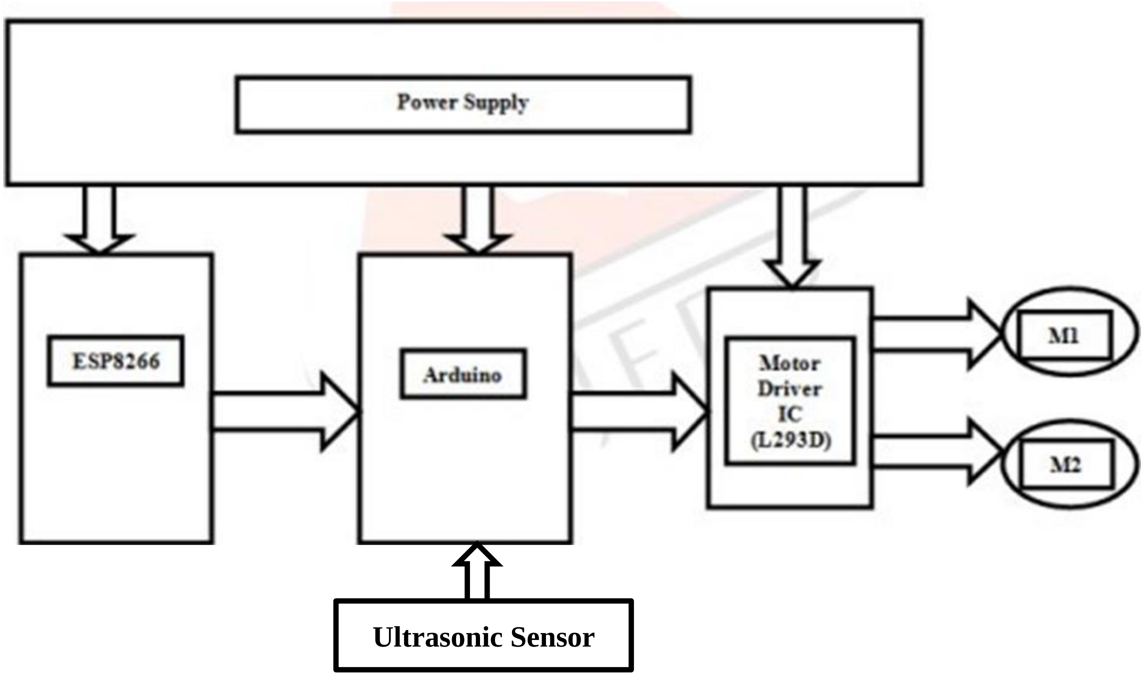


**Fig. 1: BLOCK DIAGRAM ROBOTIC CAR**

## 2.3 COMPONENTS REQUIRED

### 2.3.1 Node MCU:

The ESP8266 NodeMCU is a widely used development board based on the ESP8266 Wi-Fi module. It combines a microcontroller unit (MCU) with built-in Wi-Fi connectivity, making it ideal for IoT (Internet of Things) projects. The NodeMCU board is equipped with a powerful 32-bit Tensilica L106 MCU, offering ample processing power and storage space for embedded applications. It supports Lua scripting language, enabling quick and easy programming. The onboard Wi-Fi module provides seamless wireless communication capabilities, allowing the board to connect to the internet and interact with other devices. With its compact size and extensive community support, the ESP8266 NodeMCU is a popular choice for prototyping IoT solutions[8].

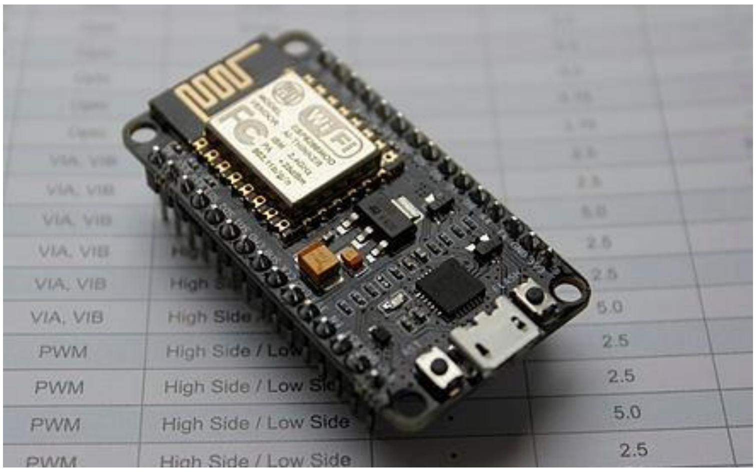


**Fig. 2: Node MCU**

**Feature:**

| | |
|---|---|
| Microcontroller: | ESP-8266 32bit |
| Node MCU Model: | Amica |
| USB Connector: | Micro USB |
| Operating Voltage: | 3.3V |
| Input Voltage: | 4.5V- 10V |
| Flash Memory: | 4 MB |
| Digital I/O Pins: | 11 |
| Analog In Pin: | 1 |

## 2.3.2 L293D MOTOR DRIVER:

The L293D motor driver is a popular integrated circuit used for controlling DC motors and stepper motors in a wide range of applications. It features four high-current half-H drivers, allowing bidirectional control of two DC motors or a single stepper motor. Each motor channel can handle a continuous current of up to 600mA and a peak current of 1.2A, making it suitable for driving small to medium-sized motors. The L293D also offers built-in protection features such as thermal shutdown and output clamp diodes to prevent damage to the driver and the motors. With its straightforward pin configuration and compatibility with microcontrollers, the L293D motor driver provides a reliable and convenient solution for motor control.

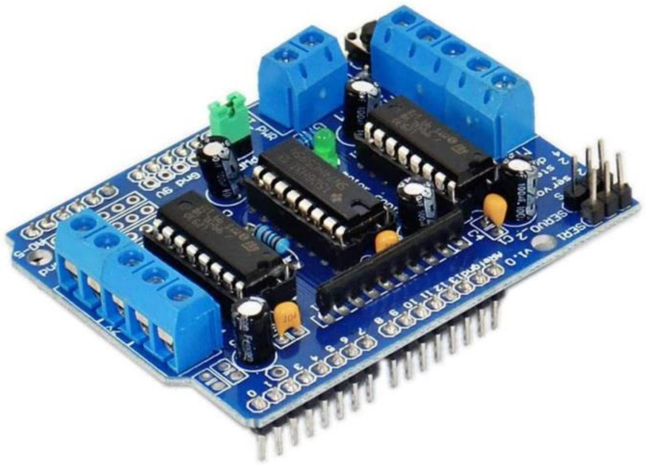


**Fig. 3: L293D MOTOR DRIVER**

**Specification:**

| | |
|---|---|
| • Input Voltage: | 4.5 - 36V |
| • Number of channel: | 4(Two H-bridges) |
| • Output Voltage: | 1.2V - Vcc |
| • Current: | 16mA(typical) |
| • Package Type: | 16-pin DIP |
| • Thermal shutdown Protection: | yes |

| **Pin Name** | **Description** |
|---|---|
| VCC1 | Power supply for the logic circuitry (5V) |
| VCC2 | Power supply for the motor (can range from 4.5V to 36V) |
| \| GND | Ground connection |
| Enable 1, 2 | Enable pins for motor channel 1 and 2 respectively. A logic high on these pins enables the motor. |
| Input 1, 2 | Input pins for controlling the direction of motor rotation for channel 1 and 2 respectively. |
| Output 1, 2 | Output pins for motor channel 1 and 2 respectively. Connect these pins to the motor terminals. |
| Motor 1, 2 | Motor supply pins for channel 1 and 2 respectively. Connect these pins to the motor power supply. |
| Enable 3, 4 | Enable pins for motor channel 3 and 4 respectively. A logic high on these pins enables the motor. |
| Input 3, 4 | Input pins for controlling the direction of motor rotation for channel 3 and 4 respectively. |
| Output 3, 4 | Output pins for motor channel 3 and 4 respectively. Connect these pins to the motor terminals. |
| Motor 3, 4 | Motor supply pins for channel 3 and 4 respectively. Connect these pins to the motor power supply. |

### 2.3.3 DC Motor:

A DC motor is any of a class of rotary electrical motors that converts direct current (DC) electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic. To periodically change the direction of current in part of the motor. DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, andappliances.

The Universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. Working principle of DC motor is that When kept in a magnetic field, a current-carrying conductor gains torque and develops a tendency to move. In short, when electric fields and magnetic fields interact, a mechanical force arises. This is the principle on which the DC motors work.

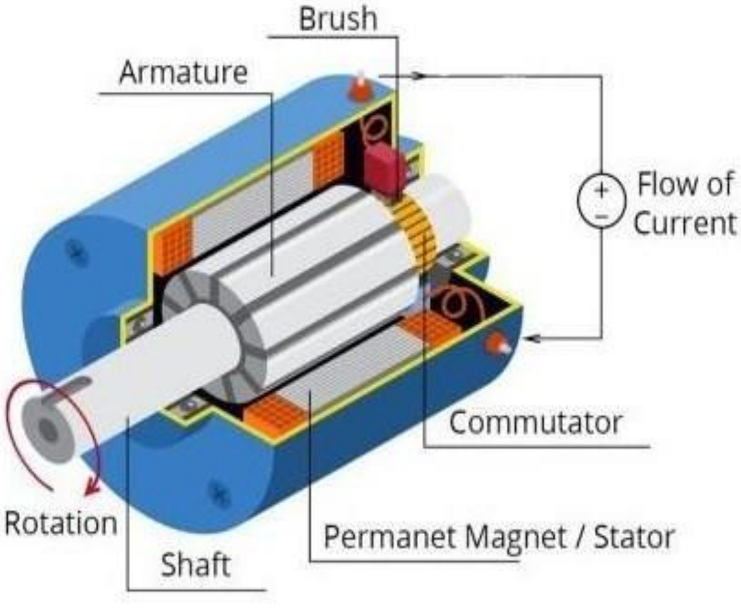


**Fig.4: DC Motor.**

### 2.3.4 Ultrasonic Sensor:

The ultrasonic sensor is a popular electronic device used for distance measurement and object detection in various applications. It utilizes ultrasonic waves, which are sound waves with frequencies above the human hearing range, typically around 40 kHz. The sensor emits ultrasonic pulses and measures the time it takes for the waves to bounce back after hitting an object. By calculating the time difference, the distance to the object can be determined accurately. Ultrasonic sensors are commonly employed in robotics, automation, parking systems, security systems, and even in medical devices. They offer non-contact sensing capabilities, high accuracy, and reliable performance, making them an essential component in many projects.

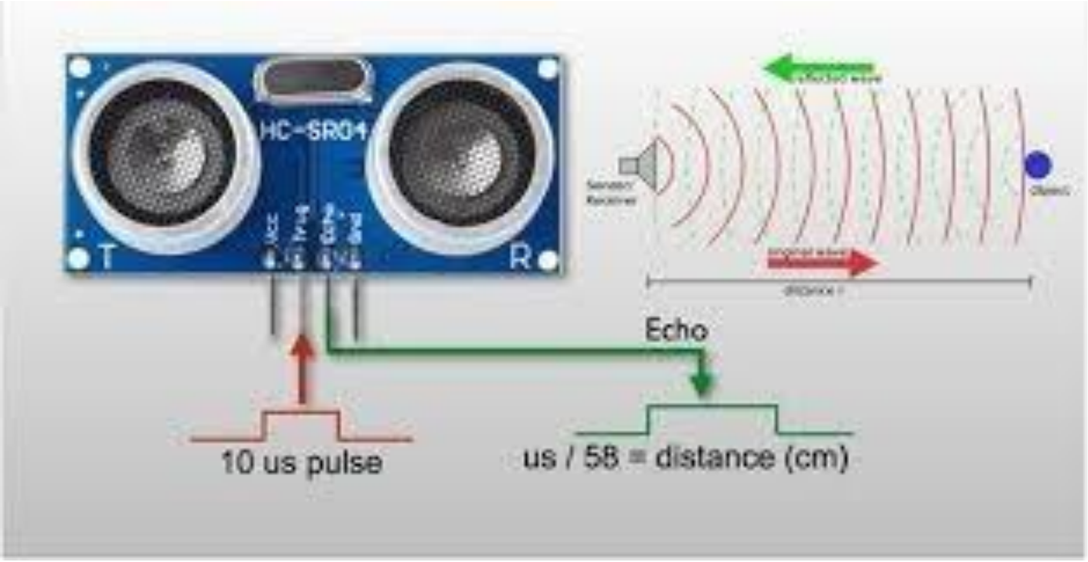


**Fig. 5(A): Ultrasonic sensor**

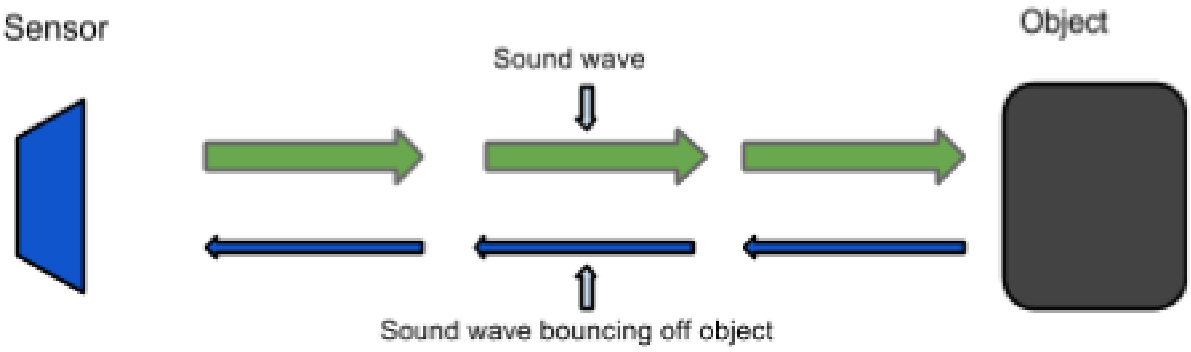
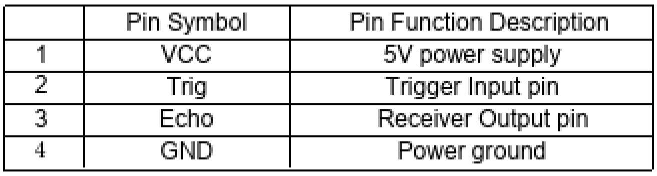


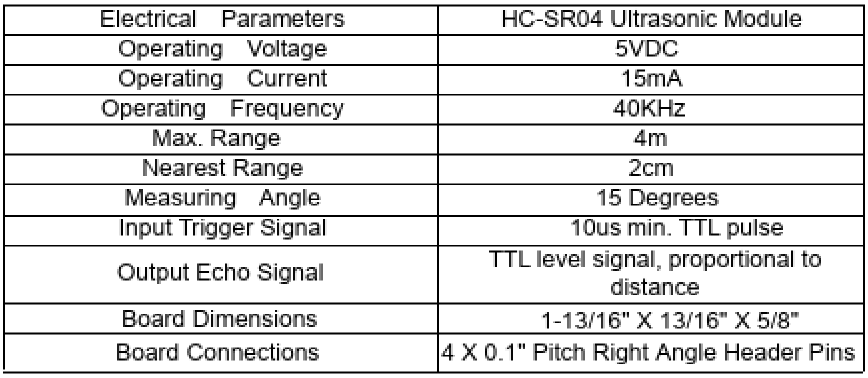
**Fig. 5(B): Actual Working of Ultrasonic sensor**

## Module Pin Assignments:



| | Pin Symbol | Pin Function Description |
|---|---|---|
| 1 | VCC | 5V power supply |
| 2 | Trig | Trigger Input pin |
| 3 | Echo | Receiver Output pin |
| 4 | GND | Power ground |

## Specification:



| Electrical Parameters | HC-SR04 Ultrasonic Module |
|---|---|
| Operating Voltage | 5VDC |
| Operating Current | 15mA |
| Operating Frequency | 40KHz |
| Max. Range | 4m |
| Nearest Range | 2cm |
| Measuring Angle | 15 Degrees |
| Input Trigger Signal | 10us min. TTL pulse |
| Output Echo Signal | TTL level signal, proportional to distance |
| Board Dimensions | 1-13/16" X 13/16" X 5/8" |
| Board Connections | 4 X 0.1" Pitch Right Angle Header Pins |

2

### 2.3.5 Gas Sensor:

The MQ3 gas sensor is a popular sensor used to detect alcohol vapor concentrations in the air. It is commonly employed in breathalyzer devices and alcohol detection systems. The sensor operates based on the principle of a tin dioxide (SnO2) sensing element, which changes its resistance when it comes into contact with alcohol vapors. The MQ3 sensor has a high sensitivity to alcohol, allowing it to detect even small amounts of alcohol in the air. It is compatible with various microcontrollers and can be easily integrated into electronic projects. The sensor provides an analog output that can be measured and processed to determine the alcohol concentration level.

**Fig. 6: Gas Sensor MQ3**

**Specification:**

| | |
|---|---|
| Sensing Element: | Tin dioxide (SnO2) |
| Detection Gas: | Alcohol vapor |
| Operating Voltage: | 5V DC |
| Heater Voltage: | 5V DC |
| Load Resistance (RL): | Adjustable, typically around 5K ohm |
| Sensitivity: | High sensitivity to alcohol vapor |
| Response Time: | <10 seconds |
| Operating Temperature: | 10°C to 50°C |
| Humidity Range: | 95% RH (non-condensing) |

### 2.3.6 9V Battery:

A 9V battery is a compact and portable power source commonly used in various electronic devices and small-scale projects. It typically consists of six smaller 1.5V cells connected in series, providing a total voltage of 9 volts. These batteries are popular in applications such as smoke detectors, remote controls, guitar effects pedals, and small electronic circuits[17]. The 9V battery's rectangular shape and snap connector make it easy to install and replace. It offers a reliable and consistent power supply, allowing devices to operate efficiently. However, due to its relatively small capacity, the 9V battery is best suited for low-power devices with intermittent usage or short-duration tasks[18].

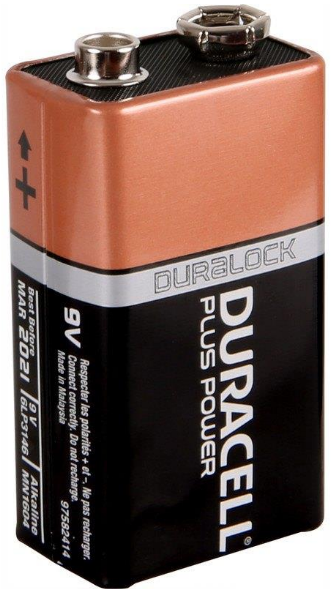


**Fig. 7: 9V Battery**

## Features:

- Can be mounted in any orientation.
- Computer designed lead, calcium tin alloy grid for high power density.
- Long service life, float or cyclic applications.
- Maintenance-free operation.
- Low self- discharge.

### 2.3.7 Light-Emitting Diode:

LED, short for Light Emitting Diode, is a semiconductor device that emits light when an electric current passes through it. LEDs are widely used in various applications due to their efficiency, durability, and versatility. These small, solid-state devices offer numerous advantages over traditional light sources .LEDs are available in a range of colors, including red, green, blue, and white, allowing for a wide spectrum of lighting possibilities. They have a long operational life, typically lasting tens of thousands of hours, which makes them highly reliable and cost-effective. LEDs also consume significantly less power compared to incandescent or fluorescent bulbs, resulting in energy savings.

One of the key advantages of LEDs is their ability to produce directional light, meaning they emit light in a specific direction. This directional nature makes them ideal for applications such as indicator lights, automotive lighting, and display screens.

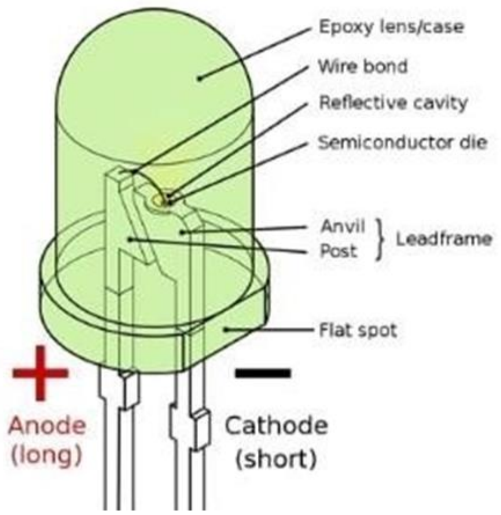


**Fig. 8: Light-Emitting Diode**

**Specification:**

| | | |
|---|---|---|
| • | Light Output: | Brightness or intensity of emitted light (lm/cd) |
| • | Color Temperature: | Warmth or coolness of the light (Kelvin) |
| • | Power Consumption: | Energy consumed by the LED (W) |
| • | Operating Voltage: | Voltage range at which the LED operates (V) |
| • | Environmental Impact | Eco-friendliness & absence of hazardous materials |

### 2.3.8 JUMPER WIRE:

A jump wire (also known as jumper, jumper wire, DuPont wire) is an electrical wire, or 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 interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. These wires typically consist of a thin conductor, such as copper, that is encased in an insulating material, such as plastic or silicone. The insulation helps prevent short circuits and electrical interference between adjacent wires or components. The ends of the jumper wires are often fitted with connectors, such as pin headers, alligator clips, or banana plugs. Jumper wires come in various lengths, color and types.

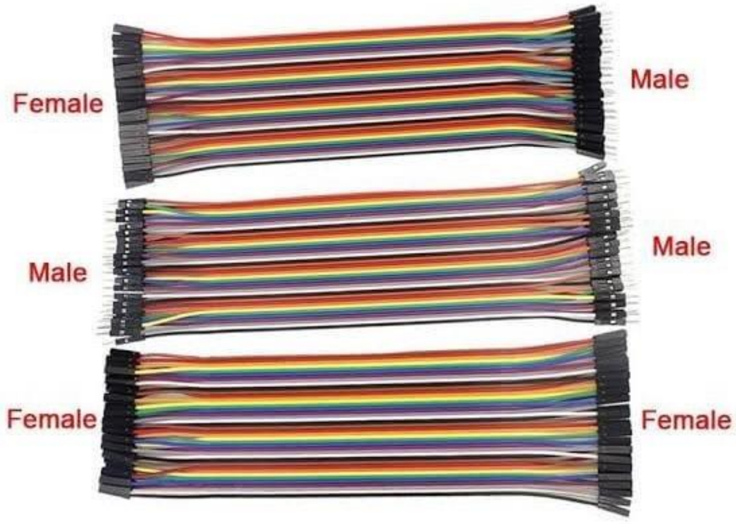


**Fig. 9: Jumper wire**

| Specification | Description |
|---|---|
| Length | Length of the jumper wire (e.g., 10cm, 20cm, etc.) |
| Wire Gauge | Thickness of the wire (e.g., 22 AWG, 26 AWG, etc.) |
| Conductor Material | Material used for the wire conductor (e.g., copper) |
| Insulation Material | Material used to insulate the wire (e.g., PVC) |
| Color | Color of the wire insulation (e.g., red, black, etc.) |
| Flexibility | Degree of flexibility or rigidity of the wire |

### 2.3.9 DIODE:

A diode is an electronic device that allows the flow of electric current in only one direction. It is a fundamental component in electronic circuits and finds applications in a wide range of fields, including telecommunications, power electronics, and signal processing.At its core, a diode consists of a semiconductor material, typically silicon or germanium, with two terminals: an anode and a cathode. These terminals determine the direction of current flow. The anode is the positive terminal, while the cathode is the negative terminal.The primary function of a diode is to enforce a one-way flow of current, allowing it to act as a rectifier. When a voltage is applied across the diode in the forward direction (anode positive, cathode negative), it conducts current with very low resistance. This allows the current to flow freely through the diode. However, when the voltage is reversed (anode negative, cathode positive), the diode blocks the flow of current and acts as an insulator.

In addition to rectification, diodes have other important properties. For instance, they exhibit a reverse breakdown voltage, beyond which the diode starts conducting in reverse. This characteristic is utilized in applications such as voltage clamping and voltage regulation.

**Fig. 10: Diode**

## Features:

**1. One-Way Current Flow:** Diodes allow current to flow in only one direction, from the anode (positive terminal) to the cathode (negative terminal).

**2. Rectification:** Diodes are commonly used as rectifiers to convert alternating current (AC) to direct current (DC) by blocking the reverse current flow.

**3. Voltage Drop:** Diodes have a forward voltage drop, typically around 0.6 to 0.7 volts for silicon diodes, which occurs when current flows through them in the forward direction.

**4. Reverse Voltage Protection:** Diodes offer reverse voltage protection by blocking the flow of current when a reverse bias voltage is applied, safeguarding circuits from potential damage.

**5**. **Nonlinear Current-Voltage Characteristic:** Diodes exhibit a nonlinear relationship between current and voltage. They have a low resistance (forward bias) and act as an insulator (reverse bias) based on the applied voltage polarity.

**6. Switching Speed:** Diodes have fast switching speeds, allowing them to quickly turn on and off in response to changes in the applied voltage or current.

**7. Temperature Dependence:** The performance of diodes is influenced by temperature changes. The forward voltage drop decreases with increasing temperature, while the reverse leakage current increases.

### 2.3.10 CAPACITOR:

A capacitor is an electronic component that stores and releases electrical energy. It consists of two conductive plates separated by a dielectric material. When a voltage is applied across the plates, an electric field is created, causing positive and negative charges to accumulate on each plate. The capacitance of a capacitor, measured in Farads (F), determines its ability to store charge. A higher capacitance means the capacitor can store more charge for a given voltage. Capacitors are available in a wide range of capacitance values, from picofarads (pF) to farads (F), depending on the application. Capacitors have various uses in electronic circuits. They can act as energy reservoirs, smoothing out voltage fluctuations and providing stable power to sensitive components. Capacitors are commonly employed in power supply filters, decoupling circuits, and voltage regulators. In addition to their energy storage and timing functions, capacitors are vital in signal processing and coupling applications. They can block direct current (DC) while allowing alternating current (AC) to pass through, enabling them to couple signals between different parts of a circuit without disturbing the DC bias.

**Fig. 11: CAPACITOR**

| **Specification** | **Discription** |
|---|---|
| Capacitance | The measure of a capacitor's ability to store charge, expressed in Farads (F). |
| Voltage Rating | The maximum voltage that can be applied across the capacitor without causing damage, specified in volts (V). |
| Tolerance | The allowable deviation from the stated capacitance value, typically given as a percentage (%). |
| Leakage Current | The small amount of current that flows through the dielectric, even when the capacitor is fully charged. It is usually specified in microamps (μA). |
| Lifespan | The expected operational lifetime of the capacitor, typically given in hours or cycles. |

# CHAPTER 3

## MRTHODOLOGY

### 3.1 STEPS IN PROJECT IMPLEMENTATION:

The steps taken in the implementation process are :

1. Circuit development and design.

2. Writing and developing code on ARDIUNO IDE.

3. WI FI testing and configuration IP Address.

4. Programming the Microcontroller.

5. Bread board testing of circuit.

6. Building and soldering circuit.

7. Troubleshooting and testing.

8. Writing code to control the Car.

9. Writing code to add the Ultrasonic Sensor and Gas Sensor.

10. Simulating circuit with the new code.

11. Reprogramming the microcontroller.

12. Further troubleshooting and testing.

13. Casing design and construction.

14. Final testing of circuit.

# CHAPTER 4

## SOFTWARE DEVELOPMENT

### 4.1 SELECTION OF MICROCONTROLLER DEVELOPMENT TOOLS:

Once microcontroller is selected, selecting a perfect development tools is most important. For develop every microcontroller based system, a set of software and hardware tools are required. Software tools for editing and debugging and troubleshooting the microcontroller program. W9/hile hardware tools for burning computer code into microcontroller and testing microcontroller hardware. A good development tools must have following properties:

1. Simple to use.
2. Not many steps execution.
3. Inexpensive.
4. Must include basic functions like editor, debugger, compiler.
5. Must include power supply and basic hardware required and I/O pins connector facility.
6. Cross-platform development.
7. Must support different programming language and computer operating system.

### 4.2 EMBEDDED C

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc

## ARDUINO IDE SOFTWARE:

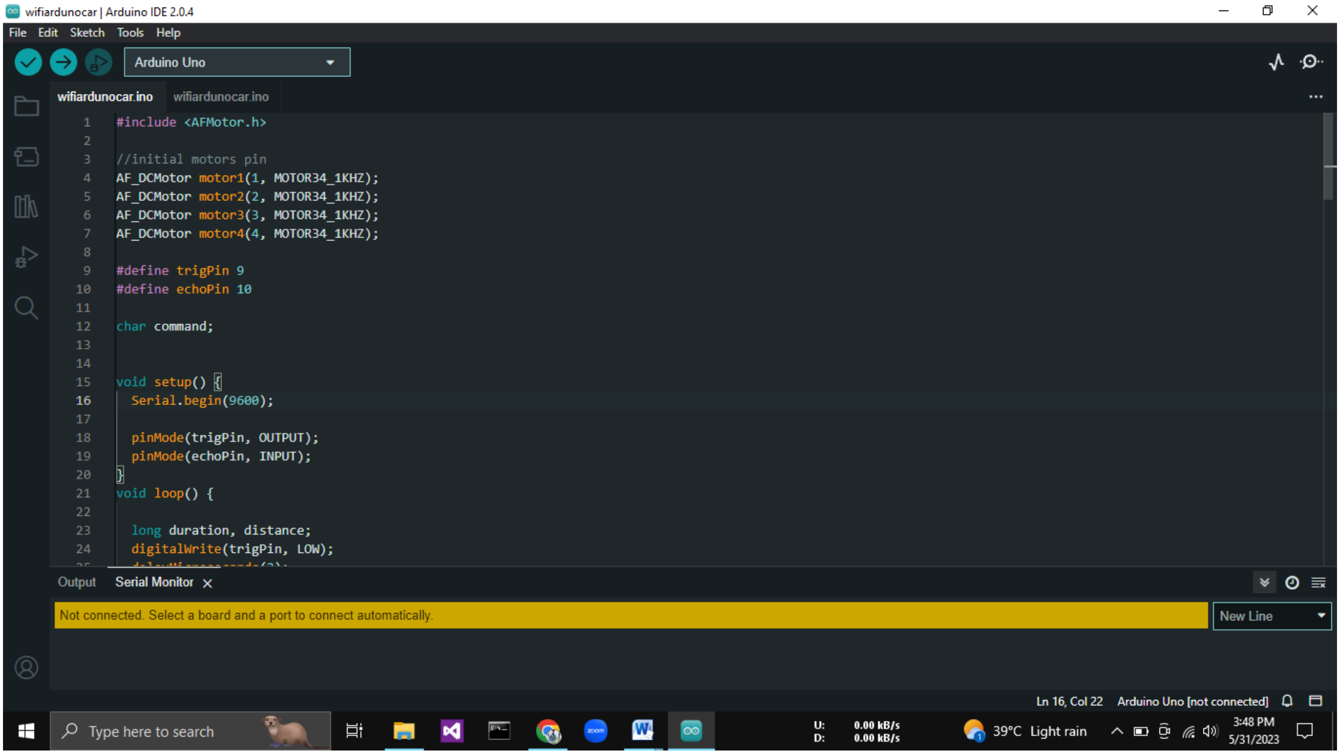
1. Overview: The Arduino IDE is an open-source software that supports the Arduino programming language, which is based on C and C++. It is compatible with various Arduino boards, including the popular Arduino Uno, Arduino Mega, and Arduino Nano.
2. Features: The Arduino IDE offers several features to facilitate the development of Arduino projects:

- Code Editor: It provides a text-based editor with syntax highlighting and auto-completion features to assist in writing code.
- Library Manager: It includes a library manager that allows users to easily search, install, and manage third-party libraries, expanding the functionality of Arduino projects.
- Serial Monitor: The IDE includes a built-in Serial Monitor tool that enables real-time communication between the Arduino board and the computer, facilitating debugging and data exchange.
- Board Manager: It supports a wide range of Arduino boards and allows users to install additional board definitions to work with different models and variants.
- Examples: The IDE comes with a collection of example sketches that showcase various functionalities of Arduino boards, providing a starting point for beginners and reference for more advanced users.
- Tools: The IDE integrates tools for compiling and uploading code to Arduino boards seamlessly.
- Debugging: While the Arduino IDE doesn't have an extensive debugging feature, users can incorporate serial print statements or use external debuggers to monitor code execution and troubleshoot issues.

3. Cross-platform Compatibility: The Arduino IDE is available for Windows, macOS, and Linux operating systems, making it accessible to a wide range of users across different platforms.
4. Community and Resources: Arduino has a large and active community of users and developers who share their projects, code, and expertise. This vibrant community provides forums, online tutorials, documentation, and a wealth of resources that can help beginners get started and support advanced users in tackling complex projects.
5. Extensibility: The Arduino IDE is highly extensible, allowing users to add custom libraries, boards, and tools. This flexibility makes it possible to adapt the IDE to suit specific project requirements or to work with non-Arduino compatible hardware.
6. Integration with Arduino Ecosystem: The Arduino IDE is an integral part of the larger Arduino ecosystem, which includes the Arduino hardware platform and a vast array of shields, sensors,

and actuators. The tight integration with Arduino boards and components ensures a seamless development experience.

In conclusion, the Arduino IDE is a user-friendly, open-source software platform designed specifically for programming Arduino microcontrollers. It offers features such as a code editor, library manager, serial monitor, and board manager, making it easy to write, compile, and upload code to Arduino boards. With cross-platform compatibility, a thriving community, and extensive resources, the Arduino IDE provides an accessible and powerful tool for creating a wide range of electronic projects.



```
#include <AFMotor.h>

//initial motors pin
AF_DCMotor motor1(1, MOTOR34_1KHZ);
AF_DCMotor motor2(2, MOTOR34_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

#define trigPin 9
#define echoPin 10

char command;


void setup() {
  Serial.begin(9600);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {

  long duration, distance;
  digitalWrite(trigPin, LOW);
```

Not connected. Select a board and a port to connect automatically.

**Fig 12. ARDUINO IDE SOFTWARE**

## STEPS OF ARDUINO IDE USE:

1. Install Arduino IDE: Download and install the latest version of the Arduino IDE from the official Arduino website (https://www.arduino.cc/en/software).
2. Connect the Arduino board: Connect your Arduino board to your computer using a USB cable. Ensure that the board is properly connected and recognized by your computer.
3. Launch Arduino IDE: Open the Arduino IDE that you installed in Step 1.
4. Select the board: From the "Tools" menu, navigate to the "Board" submenu, and select the appropriate Arduino board you are using. For example, if you have an Arduino Uno, select "Arduino/Genuino Uno."
5. Select the port: From the same "Tools" menu, navigate to the "Port" submenu, and select the port to which your Arduino board is connected. The specific port name may vary depending on your operating system.
6. Open the sketch: Either create a new sketch or open an existing one from the "File" menu. A new sketch is a blank canvas for your code.
7. Verify the sketch: Click on the "Verify" button (checkmark icon) to compile your code. This step ensures that there are no syntax errors in your program.
8. Upload the sketch: Click on the "Upload" button (right-arrow icon) to upload the compiled code to your Arduino board. The IDE will compile the code again and then transfer it to the board. You can monitor the progress in the status bar at the bottom of the IDE.
9. Wait for the upload to complete: Once the upload process starts, the Arduino IDE will display messages about the progress. Wait for the process to complete successfully. You may see a "Done uploading" message when it finishes.
10. Verify the upload: After the upload is complete, the program will start running on your Arduino board. You can check the behavior of your program by observing any connected sensors, LEDs, or other outputs.

# CHAPTER 5

# CONNECTION DIAGRAM

## 5.1 CIRCUIT DIAGRAM:

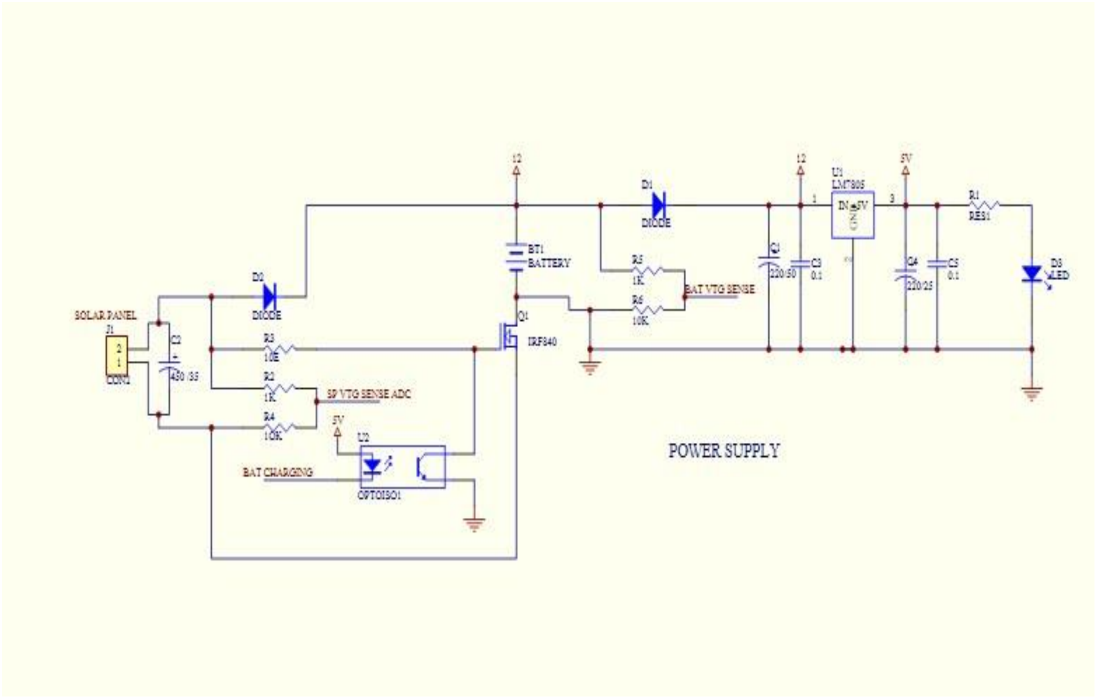


Fig. 12: Circuit Diagram of Power Supply

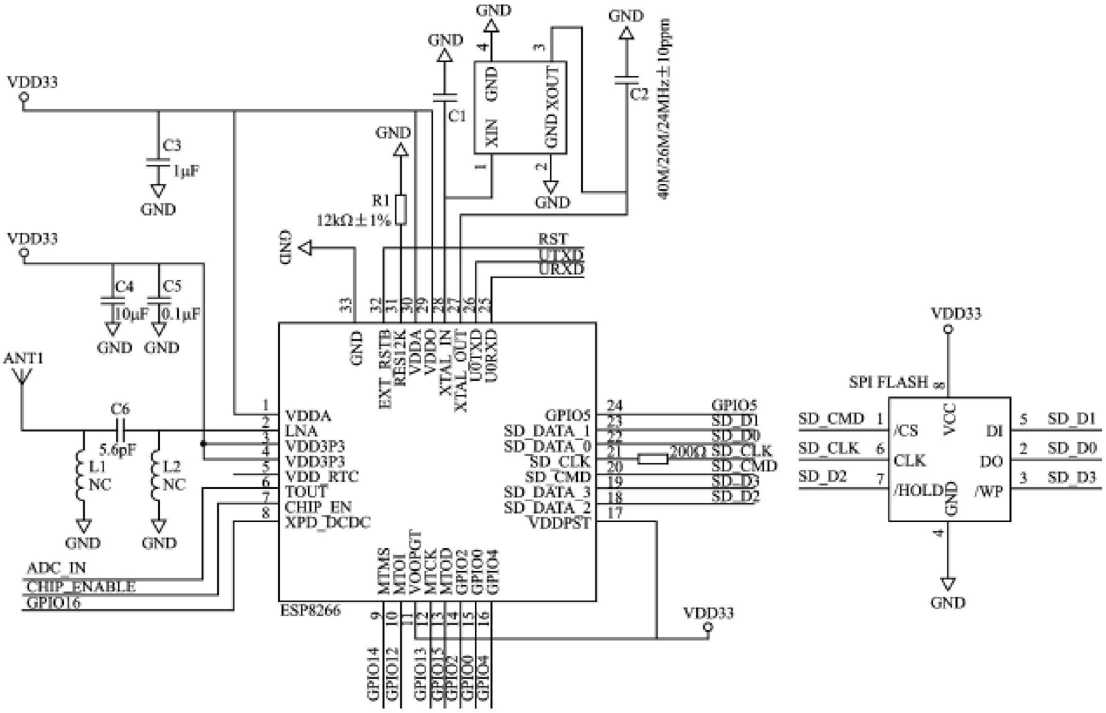


Fig. 13: Circuit Diagram of controller

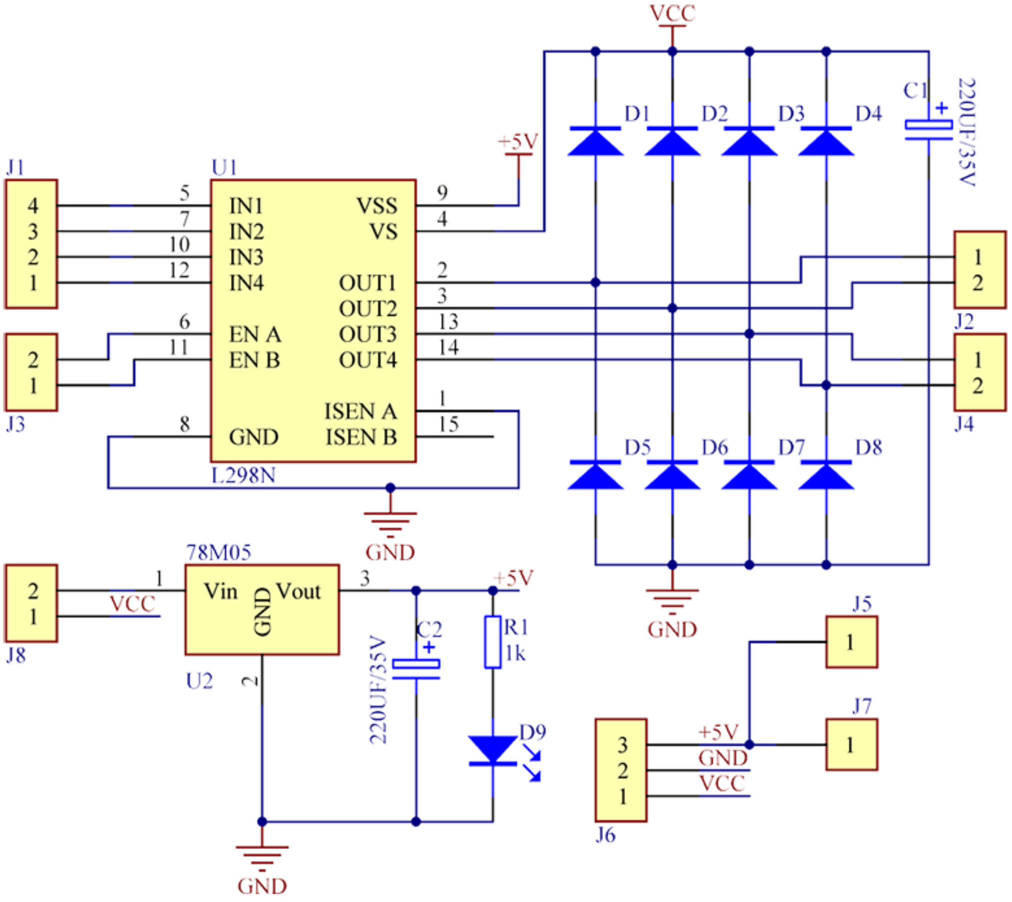


**Fig. 14: Circuit Diagram of L293D Motor Driver**

# CHAPTER 6

## 6.1 ALGORITHM

### 6.1.1 Algorithm of Nodemcu

Step 1: Start.

Step 2: Initialize connection.

Step 3: get the input from Android app.

Step 4: send the signal to the motor driver ic.

Step 5: received the input from ultrasonic sensor.

Step 6: stop

### 6.1.2 Algorithm for motor Driver IC

Step 1: Start.

Step 2: receive the input from node mcu.

Step 3: send the power to the motor.

Step 4: control the rotation speed.

Step 5: control the motor direction ex. Backward or forward

Step 6: stop.

### 6.1.3 Algorithm for ultrasonic sensor

Step 1: Start.

Step 2: send the ultrasonic waves.

Step 3: received the waves after some micro time.

Step 4: change in resistance.

Step 5: send the signal to the node-mcu.

Step 6: Stop.

## 6.2 ALGORITHM FOR ALL PROJECT

Step 1: Start.

Step 2: Initialize connection of node-mcu.

Step 3: get the input from Android app.

Step 4: send the signal to the motor driver ic.

Step 5: received the input from ultrasonic sensor.

Step 6: if (1) stop car.

Step 7: else send signal to the motor driver ic

Step 8: control motor direction speed .

Step 9: drive the car using the android .

Step 10: stop car.

Step 11: disable connection.

Step 12: Stop.

## 6.3 FLOWCHART

### 6.3.1 Flowchart of Motor Driver IC

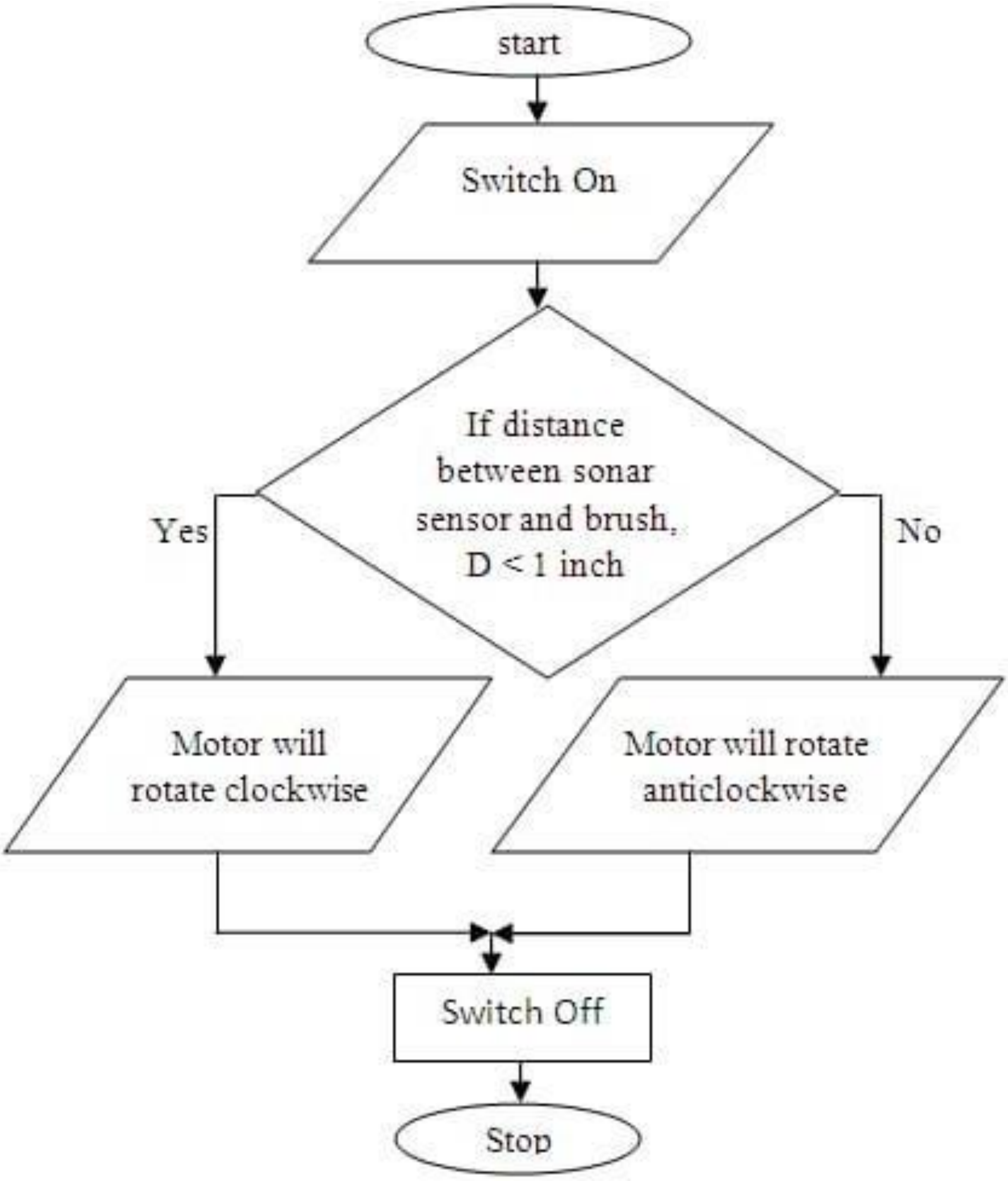

start
Switch On
If distance between sonar sensor and brush, D < 1 inch
Yes
No
Motor will rotate clockwise
Motor will rotate anticlockwise
Switch Off
Stop

**Fig. 16: Flowchart of Motor Driver IC**

### 6.3.2 Flowchart for Ultrasonic Sensor

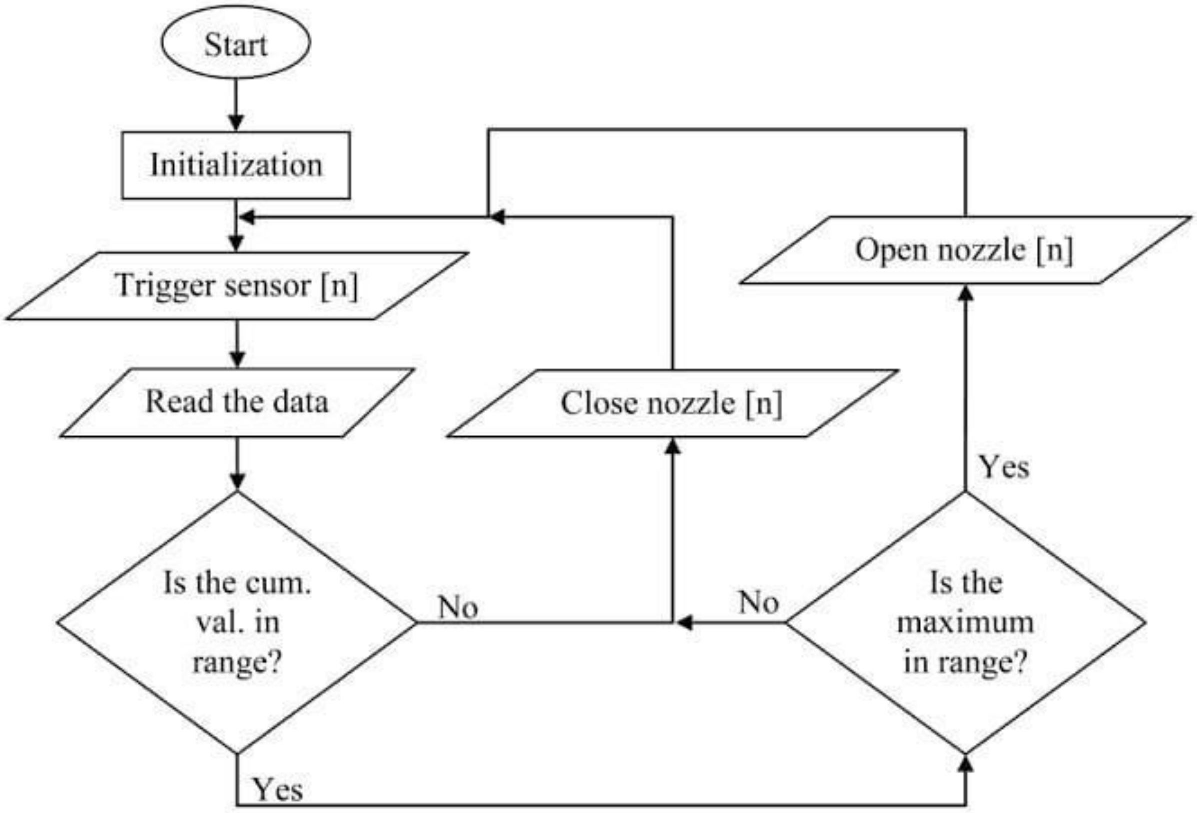


**Fig. 17: Flowchart of Ultrasonic Sensor**

### 6.3.3 Flowchart of Robotic Car

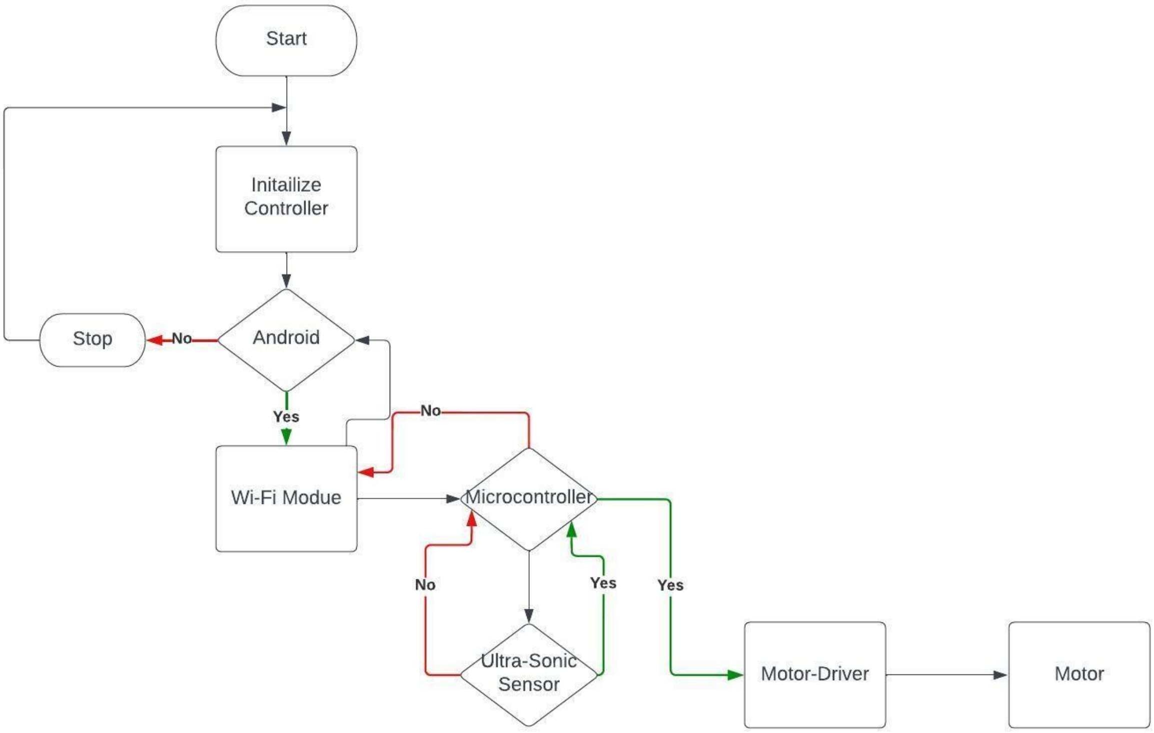


**Fig. 18: Flowchart of Robotic Car**

# CHAPTER 7
# PROGRAMING

## 7.1 PROGRAM

### Arduino UNO  Code:

```
#include <AFMotor.h>

//initial motors pin
AF_DCMotor motor1(1, MOTOR34_1KHZ);
AF_DCMotor motor2(2, MOTOR34_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

#define trigPin 9
#define echoPin 10

char command;



void setup() {
  Serial.begin(9600);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {

  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration / 2) / 29.1;
if(distance < 20)
{
  Stop();
}
if (Serial.available() > 0 && distance > 20 ) { // Check if there is data in the serial buffer
  String data = Serial.readStringUntil('\n'); // Read the data until the newline character
  Serial.print("Received data: ");
  Serial.println(data);
    Serial.println(distance);
   // Print the received data to the serial monitor
 // if(data=="R") {
       // right();
 // } else {
    command = data[0]; // Get the first character of the received string
    switch (command) {
      case 'F':
        forward();
        break;
      case 'B':
        back();
        break;
      case 'L':
        left();
        break;
      case 'R':
        right();
        break;
      default:
        Stop();
        break;
    }
    }
 // }
}

void forward(){
    motor1.setSpeed(255); //Define maximum velocity
   motor1.run(FORWARD);  //rotate the motor clockwise
   motor2.setSpeed(255); //Define maximum velocity
   motor2.run(FORWARD);  //rotate the motor clockwise
   motor3.setSpeed(255); //Define maximum velocity
    motor3.run(FORWARD);  //rotate the motor clockwise
    motor4.setSpeed(255); //Define maximum velocity
    motor4.run(FORWARD);  //rotate the motor clockwise
}

void back()
{

  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD); //rotate the motor anti-clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void left()
{

  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD); //rotate the motor anti-clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD);  //rotate the motor clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD);  //rotate the motor clockwise
}

void right()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void Stop()
{
  motor1.setSpeed(0);  //Define minimum velocity
  motor1.run(RELEASE); //stop the motor when release the button
  motor2.setSpeed(0);  //Define minimum velocity
  motor2.run(RELEASE); //rotate the motor clockwise
  motor3.setSpeed(0);  //Define minimum velocity
  motor3.run(RELEASE); //stop the motor when release the button
  motor4.setSpeed(0);  //Define minimum velocity
  motor4.run(RELEASE); //stop the motor when release the button
}
```

**ESP8266 CODE:**

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>

const char* ssid = "Vivo Y35";
const char* password = "Abhigade";
const char* serverIP = "192.168.1.100"; // IP address of the ESP 8266
const int serverPort = 9600; // serial port baud rate

SoftwareSerial espSerial(2, 3); // create a software serial port on pins 2 and 3

void setup() {
  Serial.begin(9600); // initialize serial communication at 9600 baud
  espSerial.begin(9600); // initialize software serial communication at 9600 baud
  WiFi.begin(ssid, password); // connect to WiFi network

  while (WiFi.status() != WL_CONNECTED) { // wait until connected to WiFi
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
}
```

# CHAPTER 8

## ADVANTAGES, DISADVANTAGES, APPLICATIONS

### 8.1 ADVANTAGES

1. Intuitive Control Interface: Using an Android device as the control interface provides a familiar and user-friendly experience. The touch-based controls and graphical user interface make it easy for users to navigate and operate the robot car.
2. Wireless Control: The Android-controlled robot car eliminates the need for physical connections, allowing for wireless control. This enhances flexibility and mobility, as the user can control the car from a distance without any restrictions.
3. Accessibility: Android devices are widely available and accessible to a large number of users. This makes it convenient for anyone with an Android smartphone or tablet to control the robot car, increasing its reach and potential audience.
4. Advanced Functionality: Android devices offer a wide range of features and capabilities that can be harnessed in the robot car project. For example, users can utilize the device's built-in sensors, such as accelerometers and gyroscopes, to control the car's movements or enable features like tilt-based steering.
5. Integration with other Android Apps: The Android platform allows for seamless integration with other applications. This opens up possibilities for incorporating additional functionalities, such as GPS navigation, voice control, or live video streaming from the robot car.

## 8.2 DISADVANTAGES:

1. Device Compatibility: Android-controlled robot cars may face compatibility issues with certain Android devices due to variations in hardware specifications, operating system versions, or device capabilities. Ensuring compatibility across a wide range of devices can be challenging.
2. Reliance on Battery Life: Both the Android device and the robot car require power sources to operate. Depending on the battery life of the Android device and the robot car's power supply, the runtime may be limited, requiring frequent recharging or replacement of batteries.
3. Connectivity Limitations: The wireless control of the robot car relies on a stable and reliable connection between the Android device and the car. Connectivity issues, such as signal interference or limited range, can affect the control experience and responsiveness.
4. Programming Complexity: Developing the Android application and the necessary firmware for the robot car requires a certain level of programming knowledge and expertise. This can be a barrier for beginners or individuals without programming skills who want to engage with the project.
5. Security Risks: As with any connected device, there is a potential risk of unauthorized access or hacking when using an Android-controlled robot car. Implementing appropriate security measures and keeping software up to date is essential to mitigate these risks.

## 8.3 APPLICATIONS:

1. Education and Learning: The project can be used as a practical learning tool for students, allowing them to understand the fundamentals of robotics, electronics, and programming. It provides a hands-on experience in designing, building, and controlling a robotic system.
2. Home Automation: The robot car can be employed as a part of a home automation system. It can be used to remotely monitor and control various devices and appliances, such as lights, security systems, and entertainment systems. Users can use their Android devices to navigate the robot car through the house and perform desired tasks.
3. Surveillance and Security: The robot car can serve as a mobile surveillance system for monitoring and securing a specific area. Equipped with a camera, it can capture real-time video footage and transmit it to the user's Android device, enabling remote surveillance and monitoring capabilities.
4. Exploration and Mapping: The robot car can be utilized for exploration purposes, especially in environments that are difficult or unsafe for humans. It can navigate through unknown terrains, capture images or videos, and create a map of the surroundings. This application can be valuable in search and rescue operations or in hazardous environments.
5. Entertainment and Gaming: The Android controlled robot car can be incorporated into interactive gaming systems. Users can control the car to participate in races, obstacle courses, or other gaming scenarios. This application adds an element of fun and excitement to the project.
6. Research and Development: The project can be utilized in research and development activities related to robotics, artificial intelligence, and human-robot interaction. It provides a platform for testing and experimenting with new algorithms, sensors, and control systems.
7. STEM Outreach: The Android controlled robot car project can be used in educational outreach programs to inspire and engage students in science, technology, engineering, and mathematics (STEM). It can be demonstrated at schools, science fairs, and community events to showcase the potential of robotics and encourage young minds to pursue STEM fields.

# CHAPTER 9

## COST SHEET

### 9.1 COMPONENTS, SPECIFICATIONS AND THEIR COSTS

| COMPONENTS | SPECIFICATION | QUANTITY | COST | TOTAL COST |
|---|---|---|---|---|
| Microcontroller | ESP 8266 | 1 | 450 | 450 |
| | Ardiuno Uno v3 | 1 | 750 | 750 |
| IC | Motor Driver IC | 1 | 250 | 250 |
| Sensors | Ultrasonic Sensor | 1 | 150 | 150 |
| | Gas Sensor | 1 | 250 | 250 |
| Other | DC Gear Motor | 4 | 90 | 360 |
| | Jumper wire | 30 | 2 | 60 |
| | Wheels | 4 | 40 | 160 |
| | chasis | 1 | 160 | 160 |
| | Battery | 10 | 30 | 300 |

# CHAPTER 10

## SYSTEM OVERVIEW

### 10.1 Hardware Overview:

1. ESP8266 Microcontroller: The main control unit for the robotic car, providing Wi-Fi connectivity and processing capabilities.

2. Motor Driver IC: Interfaces between the microcontroller and the motors, supplying power and control signals for movement.

3. Motors: Drive the wheels of the robotic car and determine its speed and direction.

4. Power Supply: Provides the necessary voltage and current to power the ESP8266, motor driver IC, motors, and sensors.

5. Gas Sensor: Detects the presence and concentration of specific gases in the surrounding environment. It can be used for gas leakage detection or air quality monitoring. The gas sensor is typically connected to the ESP8266 through digital or analog pins, depending on the sensor's interface requirements.

6. Ultrasonic Sensor: Measures the distance between the robotic car and obstacles in its path using sound waves. It emits ultrasonic pulses and calculates the time it takes for the pulses to bounce back after hitting an object. The ultrasonic sensor is typically connected to the ESP8266 through digital pins, using separate trigger and echo pins for distance measurement.

## 10.2 Software Overview:

1. Arduino IDE: The programming environment used for coding and uploading firmware to the ESP8266 microcontroller.

2. ESP8266Wifi Robot CAR Library: A specialized library for controlling the robotic car using the ESP8266 and Arduino. It simplifies Wi-Fi connectivity, command reception, motor control, and sensor interfacing.

3. Firmware: The software code uploaded to the ESP8266 microcontroller, written in Arduino programming language. The firmware code handles Wi-Fi communication, interprets incoming commands, controls the motor driver IC, and interfaces with the sensors.

4. Wi-Fi Communication: The ESP8266 connects to a Wi-Fi network, either as a client or an access point, enabling wireless communication with a remote device. This allows the user to control the car remotely and receive sensor data.

5. Motor Control: The firmware code interprets the received commands and sends appropriate signals to the motor driver IC to control the motors' speed, direction, and rotation based on the user's input.

6. Gas Sensor Integration: The firmware code includes logic to read data from the gas sensor. This involves configuring the sensor's interface, reading sensor values, and performing any necessary data processing or calibration.

7. Ultrasonic Sensor Integration: The firmware code includes logic to trigger the ultrasonic sensor and measure the time it takes for the sound waves to return. This information is used to calculate the distance to obstacles, which can be used for collision avoidance or navigation purposes.

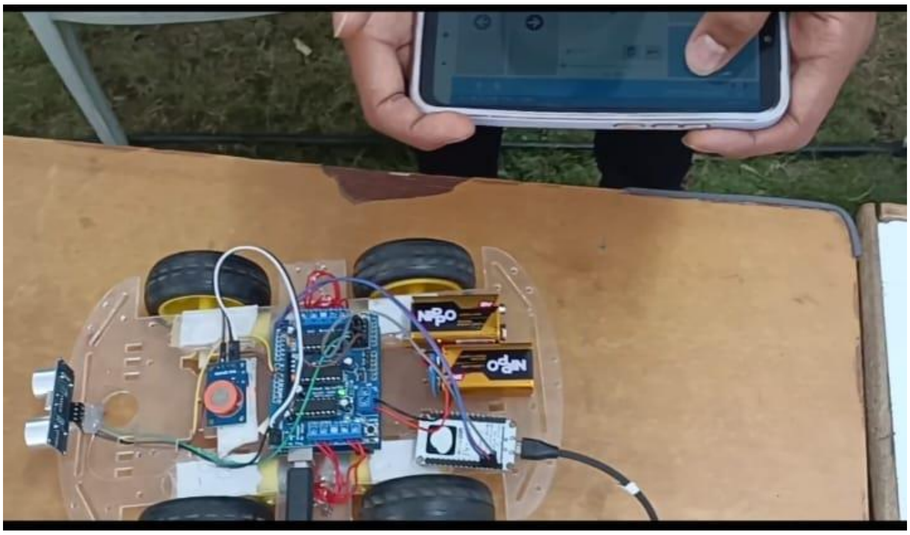
By incorporating the gas sensor and ultrasonic sensor into the robotic car system, you can enhance its functionality by adding gas detection and obstacle avoidance capabilities. The firmware code and sensor integration ensure that the sensors' data is properly collected, processed, and utilized in controlling the car's behaviour.

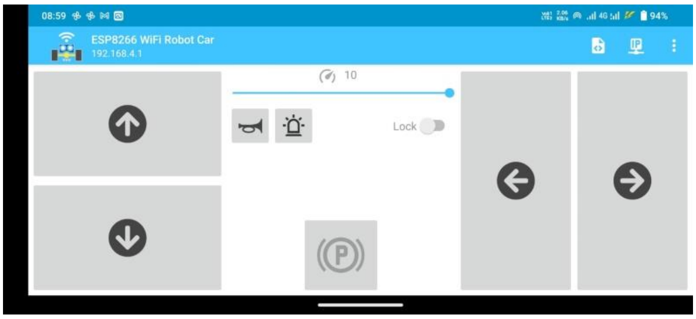
# CHAPTER 11

## 11.1 RESULTS :

The Android Controlled Robot Car project has achieved the following results:

1. Successful integration of Android technology: The project successfully integrated Android smartphones or tablets as the control interface for the robot car, leveraging the power of mobile computing and touch-based interaction.
2. Remote control capabilities: Users can control the robot car wirelessly using the Android application, providing them with the ability to navigate the car in different directions, such as forward, backward, left, and right.
3. Autonomous navigation: By incorporating obstacle detection and avoidance systems, the robot car can navigate autonomously, detecting and avoiding obstacles in its path.
4. Seamless control experience: The control interface provided by the Android application offers intuitive and user-friendly controls, enabling users to operate the robot car effortlessly.
5. Potential for future development: The Android Controlled Robot Car project lays the foundation for further enhancements and advancements in the field of remote-controlled vehicles and robotic systems. It showcases the possibilities of merging Android technology with robotics, fostering innovation and exploration in this domain.





# CHAPTER 12

## CONCLUSION

### 12.1 CONCLUSION:

- The Android Controlled Robot Car project successfully combines the power of Android technology with robotics to create a versatile and interactive vehicle. Through the integration of an Android device and a custom-built application, users can remotely control the robot car, providing a seamless and intuitive control experience.
- Throughout the development process, the project team focused on optimizing wireless communication, control algorithms, and hardware components. Extensive testing and iterations were performed to ensure reliable and responsive control, allowing users to maneuver the car effortlessly. The addition of sensors, such as obstacle detection and avoidance systems, further enhanced the functionality of the robot car, enabling autonomous navigation and collision avoidance.
- The project's results demonstrate the potential of merging Android technology with robotics. The Android Controlled Robot Car opens up possibilities for various applications, ranging from entertainment and hobbyist projects to practical applications in areas like surveillance, exploration, and automation.
- By leveraging the capabilities of Android smartphones or tablets, users can control the robot car from a distance, enabling them to explore and interact with their surroundings in new and exciting ways. The integration of a live video stream from the onboard camera provides a first-person perspective, enhancing the user's immersion and control experience.
- The Android Controlled Robot Car project represents a successful convergence of mobile computing, wireless communication, and robotics, showcasing the advancements made in these fields. It serves as a stepping stone for further innovation and development in remote-controlled vehicles and robotic systems.
- Overall, this project demonstrates the potential of combining Android technology with robotics to create engaging and interactive experiences, fostering curiosity, exploration, and learning in the field of robotics.

# CHAPTER 13

## REFERENCES

### 13.1 REFERENCES:

1. Malik A, Shrivastava A, Singh G K and Shuk A (2015). Remote controlling of a vehicle using android app. using 8051 microcontroller, Int. J of Res. in Eng. & Adv. Tech. & Sci, Vol. 2, Issue-3.
2. Hebah H O Nasereddin and Abdelkarim A A (2010). Smartphone control robots through bluetooth, Source: IJRRAS, Vol. 4.
3. Parmar D, Tripathi D, Sahni A, Singh P (2015). Bluetooth operated robot vehicle using mobile android app. Int. J of Res. in Eng. & Adv Technol, Vol. 3, Issue 2, 45-48.
4. Eshita R Z, Barua T, Barua A and Dip A (2016). Bluetooth based android controlled robot. American Journal of Engineering Research (AJER). Vol. 5, Issue 3, 195-199.
5. Schaffer A A, Haddadin S, Ott Ch, Stemmer A, Wimbock T and Hirzinger G (2007). The DLR lightweight robot: design and control concepts for robots in human environments. Industrial Robot: An International Journal, Vol. 34, No. 5, 376–385.
6. Gandotra S, Sharma B, Mahajan S, Motup T, Choudhary T and Thakur

   P (2016). Bluetooth controlled RC car using Arduino. Imp. J of Interdisciplinary Research (IJIR) Vol. 2, Issue. 9, 144-147.
7. Zeroday. "A lua based firmware for wifi-soc esp8266". *Github*. Retrieved 2 April 2015.
8. Hari Wiguna. "NodeMCU LUA Firmware". *Hackaday*. Retrieved 2 April 2015.
9. Systems, Espressif. "Espressif Systems". *Espressif-WikiDevi*. Archived from the original on 1 December 2017. Retrieved 3 June 2017.
10. Brian Benchoff (2 January 2015). "A DEV BOARD FOR THE ESP LUA INTERPRETER". *Hackaday*. Retrieved 2 April 2015.
11. Espressif system (December 30, 2013). "IoT Wi-Fi 802.11b/g/n integrated SoC implementation of volume production". Archived from the original on 2 April 2015. Retrieved 2 April 2015.
12. Hong. "First commit of NodeMCU Firmware". *Github*. Retrieved 2 April 2015.
13. Huang R. "Initial design of NodeMCU devkit". *Github*. Retrieved 2 April 2015.
14. Tuan PM. "MQTT client library for ESP8266". *Github*. Retrieved 2 April 2015.

15. Olikraus; Daniel Sittig. "Universal Graphics Library for 8 Bit Embedded Systems". *Google code*. Retrieved 2 April 2015.
16. Devsaurus. "U8glib for esp8266". *Github*. Retrieved 2 April 2015.
17. "9-Volt Battery Safety". National Fire Protection Association. 2016. Archived (PDF) from the original 2016-07-27. Retrieved 2022-05-01.
18. "Non-Rechargeable Batteries (filtered for 9V)". CPC Farnell.Retrieved 2022-04-22. 9-volt batteries supplied by a typical distributor: PP3 (or CRV9), PP6, PP7, and PP9 are the PP sizes; A10 and A29 the only other 9v types.