VOICE CONTROLLED CAR
WITH OBSTACLE DETECTION AND
DISTANCE MEASUREMENT

Submitted by

In the partial fulfillment of the requirements for the award degree of Bachelor of Technology in

Electronics and Communication Engineering

By

ANNAPARTHI SRIVEDA (Reg.No.161FA05214)

RACHARLA VINEETH (Reg.No.161FA05261)

KOTHA SAI PRIYA (Reg.No.161FA05359)

Under the esteemed guidance of

External Guide

Ms. NEHA TIWARI

Senior Software Engineer

Elorce Industries Pvt Ltd

Ameerpet

Hyderabad

Internal Guide

Dr.M. LAAVANYA

Associate Professor

Dept of ECE

VFSTR

Vadlamudi, Guntur

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

(ACCREDITED BY **NAAC** WITH **'A'**GRADE)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH

(Deemed to be university)

VADLAMUDI, GUNTUR – 522213, INDIA

May-2020

CERTIFICATE

This is to certify that the internship report entitled "**VOICE CONTROLLED CAR WITH OBSTACLE DETECTION AND DISTANCE MEASUREMENT''** that is being submitted by **ANNAPARTHI SRIVEDA** bearing **Reg.No.161FA05214, RACHARLA VINEETH** bearing **Reg.no.161FA05261, KOTHA SAIPRIYA** bearing **Reg.no.161FA05359** in partial fulfillment for the award of degree of **Bachelor of Technology in Electronics and Communication Engineering** to Vignan's Foundation for Science Technology and Research University, Guntur during the year 2019-2020 is a record of bonafide work carried out by them at ELORCE under the supervision of **Ms.Neha Tiwari,** senior software engineer, ELORCE Industries Pvt Ltd and the internal guidance of **Dr.M.Laavanya** of ECE Department.

Signature of the Internal guide	Signature of Head of the department
Dr. Laavanya	Dr. T. Pitchaiah, M.E, PhD, MIEEE
Associate Professor	Professor
Department of ECE	Department of ECE
VFSTR	VFSTR
Vadlamudi, Guntur	Vadlamudi, Guntur

DECLARATION

We here by declare that the internship Project entitled **"VOICE CONTROLLED CAR WITH OBSTACLE DETECTION AND DISTANCE MEASUREMENT"** is being submitted to Vignan's Foundation for Science, Technology and Research (Deemed to be University) in partial fulfillment for the award of **B.Tech** degree in **Electronics and Communication Engineering.** The work was originally designed and executed by us under the guidance of our supervisor **Ms. Neha Tiwari ,** Senior Software Engineer, ELORCE Industries Pvt Ltd and **Dr.M.Laavanya** as Internal guide at Department of Electronics and Communication Engineering, Vignan's Foundation for Science Technology and Research (Deemed to be University) and was not a duplication of work done by someone else. We hold the responsibility of the originality of the work incorporated into this thesis.

Date: Signature of the candidate

ANNAPARTHI SRIVEDA (Reg. No.161FA05214)

RACHARLA VINEETH (Reg.No.161FA05261)

KOTHA SAIPRIYA (Reg.no.161FA05359)

ACKNOWLEDGEMENT

It's our privilege to thank **Ms. Neha Tiwari,** senior software engineer, ELORCE Industries Pvt Ltd, who had been a source of inspiration and for their timely guidance in the conduct of our project work. We wish to express our deep sense of gratitude to our Internal Guide, **Dr. Laavanya,** Associate Professor, VFSTR (Deemed to be University) for their help and support during the program.

We would like to specially thank **Dr T. Pitchaiah**, Head of the Department, ECE and **Mr. N.Srikanta** Internship co-ordinator, Department of ECE, VFSTR (deemed to be university) for their help and support during the program.

Words are inadequate in offering our sincere thanks to **Mr.Karthik, Mrs.Manogna HR Manager**, **ELORCE Industries Pvt Ltd** and all the team members for their encouragement, guidance and cooperation during our stay in the company.

We take immense pleasure in thanking **Mr. Avinash ,** Program Director, ELORCE, Hyderabad, for having permitted us to carry out this Internship Project in esteemed organization, Elorce Industries Pvt Ltd. Also, we wish to express our gratitude to **DR.M.Y.S.Prasad,** Vice-Chancellor, VFSTR (Deemed to be University) for providing us the greatest opportunity to have industrial exposure and to carry out the Live Projects.

We would like to thank all my faculty and technical staff for their support to finish my work successfully.

We wish to express our heart full thanks our parents for their support and encouragement throughout our life.

ANNAPARTHI SRIVEDA (Reg. No.161FA05214)

RACHARL VINEETH (Reg.No.161FA05261)

KOTHA SAI PRIYA (Reg.No.161FA05359)

ABOUT THE ORGANIZATION

Founded in 2018, ELORCE Industries PVT LTD located at Hyderabad, is an innovative and reliable technology company. Our offer of custom application development, web design and development, online marketing and rich internet applications come with a strong vertical focus. Also has a background in developing academic student projects, software development and continues its entire attention on achieving transcending excellence in the development and maintenance of software projects and products in many areas.

With an offer of value-driven technology solutions, we aim to enable and synergize the industry verticals in the global market place. Our technology solutions leverage on the power of Web 2.0 applications, latest technologies, and rich internet applications. In today's modern technological competitive environment, students in electronics stream want to ensure that they are getting guidance in an organization that can meet their professional needs. With our well equipped team of solid information systems professionals, who study, design, develop, enhance, customize, implement, maintain and support various aspects of Information Technology, students can be sure.

We understand the students needs, and develop their quality of professional life by simply making the technology readily usable for them. We practice exclusively in software development, network simulation, search engine optimization, customization and system integration.

Our project methodology includes techniques for initiating a project, developing the requirements, making clear assignments to the project team, developing. A dynamic schedule, reporting status to executives and problem solving.

ABSTRACT

This project represents the design and implementation of a Voice controlled car with obstacle detection and measurement to engage surveillance in co-ordination with a human operator. An unmanned ground vehicle is operated without an on-board human presence, while in contact with the ground. The vehicle in this project is based on a four-wheel chassis. The vehicles motion is being controlled through a Bluetooth module which receives navigation commands from an android device. That received commands from Bluetooth module are sent to the Arduino Uno R3 board which process the code and moves the motors accordingly with the help of motor shield. An ultrasonic sensor is being integrated in the vehicle to detect the obstacle and a 16*2 LCD display is used to display the distance between obstacle and the vehicle. It is constructed to operate on an 12v LiPo battery.

LIST OF FIGURES Page No

CONTENTS

CHAPTER-1
INTRODUCTION

1.1 Introduction

The roots of foundation of robotics belong back in 1950s; more than six decades has passed since then, and robotic evolution has been running unparalleled. Today we can feel the presence of robots everywhere and in every field whether it is medical, military, education or governance and so on. Robotics has become a helping hand for humans, and they are making our life easier, better and faster. Robotics is a boon for human kind, because robotic machines are giving alternatives, which is providing a great support to physically impaired people. This proposal focuses in the problems faced by specially abled people who wish to drive their vehicle on their own but they cannot because of the natural cause. In this proposed project, the emphasis is on voice control of robot with automatic braking, speed slow down and avoidance of obstacles automatically or manually (through voicecommand).In this model, we will use an android app to pass on the voice commands to the Arduino through Bluetooth communication using Bluetooth module. Ultrasonic sensor will act as the obstacle detector, which will act as a mediator for Arduino microcontroller (or the CPU) and the proximal environment, and it would eventually lead to slowing down the vehicle or to fully break its motion. An LCD will display the proximity of the obstacle, and if the proximity would reach to a threshold limit, a buzzer will beep up through the processing of the program that Arduino would do. Eventually after that the vehicle will finally stop. It would be a great aid for the safety of people driving it.

1.2 Voice Command Input

An android mobile will work as the agent for voice command given to Arduino microcontroller through the help of a specific voice-control app**.**

1.3 Obstacle Detection

The purpose of this part is to secure the driver and vehicle from different hurdles. To achieve this, ultrasonic sensor is used in the front side of the vehicle. The sensor consists of one transmitter and receiver. The transmitter of the sensor will continuously transmit the signals. When these signals will collide with any object and will be reflected back, the receiver of sensor will get these reflected signals and forward them to the Arduino microcontroller. To find the object distance, the signal is

sent to processing unit through different ports. In ultrasonic sensor, the power is directly proportional to range of ultrasonic sensor. $P \propto$ Range.

CHAPTER – 2
LITERATURE SURVEY

2.1 Literature Survey

Various researches have been made by different researchers in developing this project. However, they Serve a different application and have different technologies implemented. Some of those papers are mentioned below stating their technology and application.

Jorge Kazacos Winter has developed android controlled robot automation. Main aim of his project was the transfer of information wirelessly between a smart phone and the robot and developing the robot and its communication system underneath a low price and open source philosophy. He used 3D Design technique to style the structure of the robot with the facilitation of parametrical modeling software.

The style, when fed to the 3D printer can print the parts of the robot in a layered manner one by one and can then use these parts to assemble the robot simply. He has used Arduino micro-controller and Wi-Fi technology in this robot.

M.Selvam in his paper has projected design to develop a robotic system which has a wireless camera attached to it for surveillance. Bluetooth was implemented in his project for providing connection between robot and smart phone. Wireless night vision camera was used for providing remote surveillance. The video which is recorded by camera is then transmitted to TV unit through Radio Frequency signal. He used 8051 micro controller for the robotic unit.

Vito M Guardi has evolved the method of Bluetooth technology by developing an android app for a robot which is driven by a microcontroller. The central idea of his work is to show that one android app can be operated using totally different electronic devices. Vito M Guardi has invented a communication protocol for android smart phone and robotic platform over a Bluetooth.

Ranjith Kumar Goud and B.Santhosh Kumar have invented a pick and drop robot. They wanted it to be used for diffusing a bomb remotely with safety. For the robotic arm, they used a pair of motors and another pair as the wheels of the robot for controlling the movement. Connectivity is established using Bluetooth. The micro-controller used is LPC2148. They had also attached a wireless camera for remote surveillance. They have worked on this project mainly for industrial and military applications.

Xiao Lu, Wenjun Liu, Haixia Wang, Qia Sun have published a paper based on a project in which the smart phone is capable of IFLYTEK voice as well as handwritten input. The design is therefore robust, suitable, and practical for use and it also ensures the reliability of the full system. For connectivity between the smart phone and robot, Wi-Fi is used. Use of Wi-Fi makes it easy and absolutely convenient for controlling the robot so that it can act according to the commands.

Arpit Sharma, Reetesh Verma, Saurabh Gupta, Sukhdeep Kaur Bhatia have configured an android smartphone which can control a robot via Bluetooth technology. The phone uses motion sensors and records the gestures sent via an android mobile phone. It also has an inbuilt accelerometer and Bluetooth.

Some of the defects identified in the previous papers are that Wi-Fi module is little expensive, by using microcontroller it leads to complexity of implementation.

2.2 problem identification

A car or automobile is a wheeled motor vehicle used for transportation. Some major problems have been identified.

- In this project, there is an additional specification that is, the user sets the destination and should give the voice commands like start and stop.
- Object detection is included in this so that it automatically deviates the direction by avoiding the obstacles.
- It shows the distance between the vehicle and obstacle. By this accidents can be avoided.
- This technology can only be used where similar technology vehicles are running.

CHAPTER-3
PROPOSED SYSTEM

3.0 proposed system

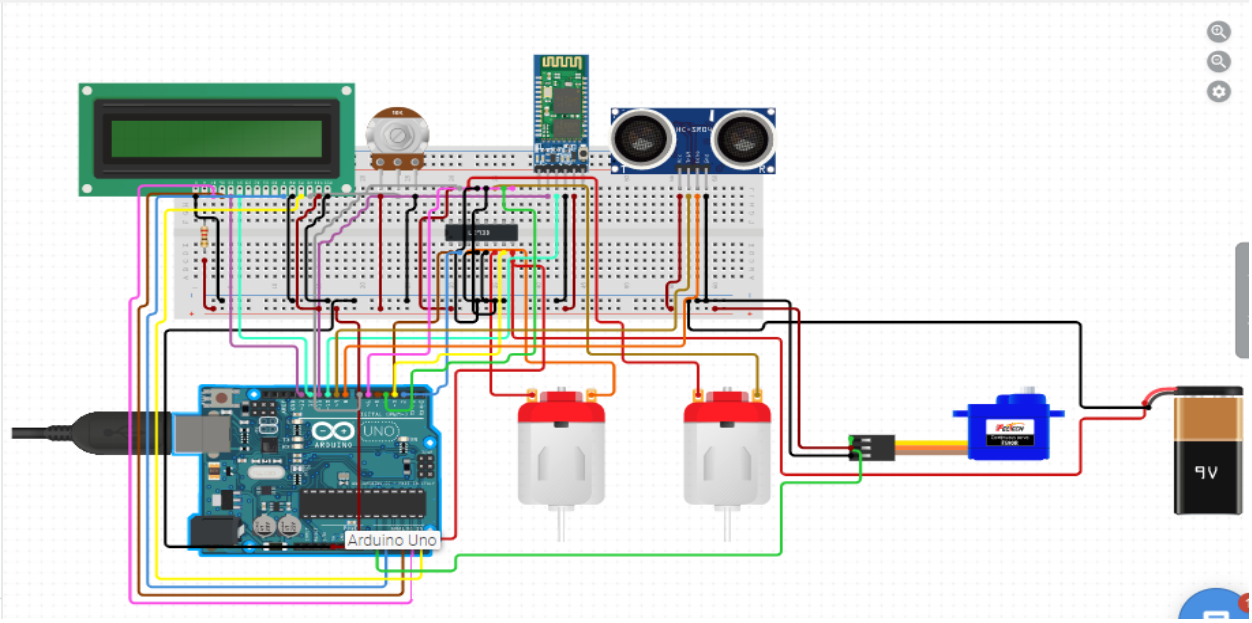


Fig.3.1 The proposed System

3.1 Components Used and Their Description

The list of components of used to build the voice controlled car is:

- Arduino uno R3
- L239D motor driver
- HC05 Bluetooth module
- LCD
- Ultrasonic sensor
- Mini bread board
- Servo motor
- Robotic kit

- Ultrasonic sensor holder
- Jumper wires

3.2 Arduino Uno R3

Arduino is a rapid electronic prototyping platform composed by the Arduino board and the Arduino IDE. Arduino is a popular programmable board used to create projects. It consists of a simple hardware platform as well as a free source code editor which has a “one click compiles or upload” feature. Hence it is designed in way that one can use it without necessarily being an expert programmer (Kushner 1987). Arduino offers an open-source electronic prototyping platform that is easy to use and flexible for both the software and hardware. Arduino is able to sense the environment through receiving input from several sensors.

3.2.1 Why Arduino?

It is an open-source project, software/hardware is extremely accessible and very flexible to be customized and extended. It is flexible, offers a variety of digital and analog inputs, SPI and serial interface and digital and PWM outputs. It is easy to use, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers. It is inexpensive, around 30 euros per board and comes with free authoring software. Arduino is backed up by a growing online community, lots of source code is already available Due to its open-source environment, one is able to easily write and upload codes to the I/O board. It is also worth to note that Arduino can be run on Linux, Mac OSX and Windows as its environment is written in Java.

Arduino board has been used for making different engineering projects and different applications. The Arduino software is very simple to use for beginners, yet flexible adequate for advanced users. Teachers and students in the schools utilize it to design low cost scientific instruments to verify the principles of physics and chemistry.

There are numerous other microcontroller platforms obtainable for physical computing. Arduino also makes simpler the working process of microcontroller, but it gives some advantages over other systems for teachers, students and beginners.

- Inexpensive
- Cross-platform
- Simple, clear programming environment
- Open source and extensible software
- Open source and extensible hardware

3.2.2 What can we do with Arduino ?

Arduino is a great tool for developing interactive objects, taking inputs from a variety of switches or sensors and controlling a variety of lights, motors and other outputs. Arduino projects can be stand-alone or they can be connected to a computer using USB. The Arduino will be seen by the computer as a standard serial interface There are serial communication APIs on most programming languages so interfacing Arduino with a software program running on the computer should be pretty straight forward. The Arduino board is a microcontroller board, which is a small circuit (the board) that contains a whole computer on a small chip (the microcontroller). There are different versions of the Arduino board: they are different in components, aim and size, etc. Some examples of Arduino boards are: Arduino Diecimila, Arduino Duemilanove, Freeduino, Arduino NG and lot more. Arduino schematics are distributed using an open license so anyone is free to build his own Arduino compatible board. The Arduino name is a registered trademark so you won't be able to call your hacked board Arduino. The founder of arduino is "Massimo Banzi".

3.2.3 Arduino Uno R3

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages

between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Fig.3.2 Arduino Uno R3

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases.

The ATmega328 on the board comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

3.2.4 Technical specifications:

- Microcontroller: Microchip ATmega328P [7]
- Operating Voltage: 5 Volts

- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 can provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by boot loader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

3.2.5 Pins:

General pin functions

LED: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.

VIN: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source).

You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

3.3V : A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND: Ground pins.

I/O REF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.

Reset: Typically used to add a reset button to shields that block the one on the board.

3.2.6 Special pin functions

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pin Mode(), digital Write(), and digital Read() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analog Reference() function.

In addition, some pins have specialized functions:

- **Serial / UART**: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
- **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM (pulse-width modulation):** pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analog Write() function.
- **SPI (Serial Peripheral Interface):** pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI (two-wire interface) / I²C:** pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
- **AREF (analog reference):** Reference voltage for the analog inputs..

3.2.7 Automatic (software) reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line

of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. This setup has other implications. When the Uno is connected to a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

3.2.8 Pin Description:

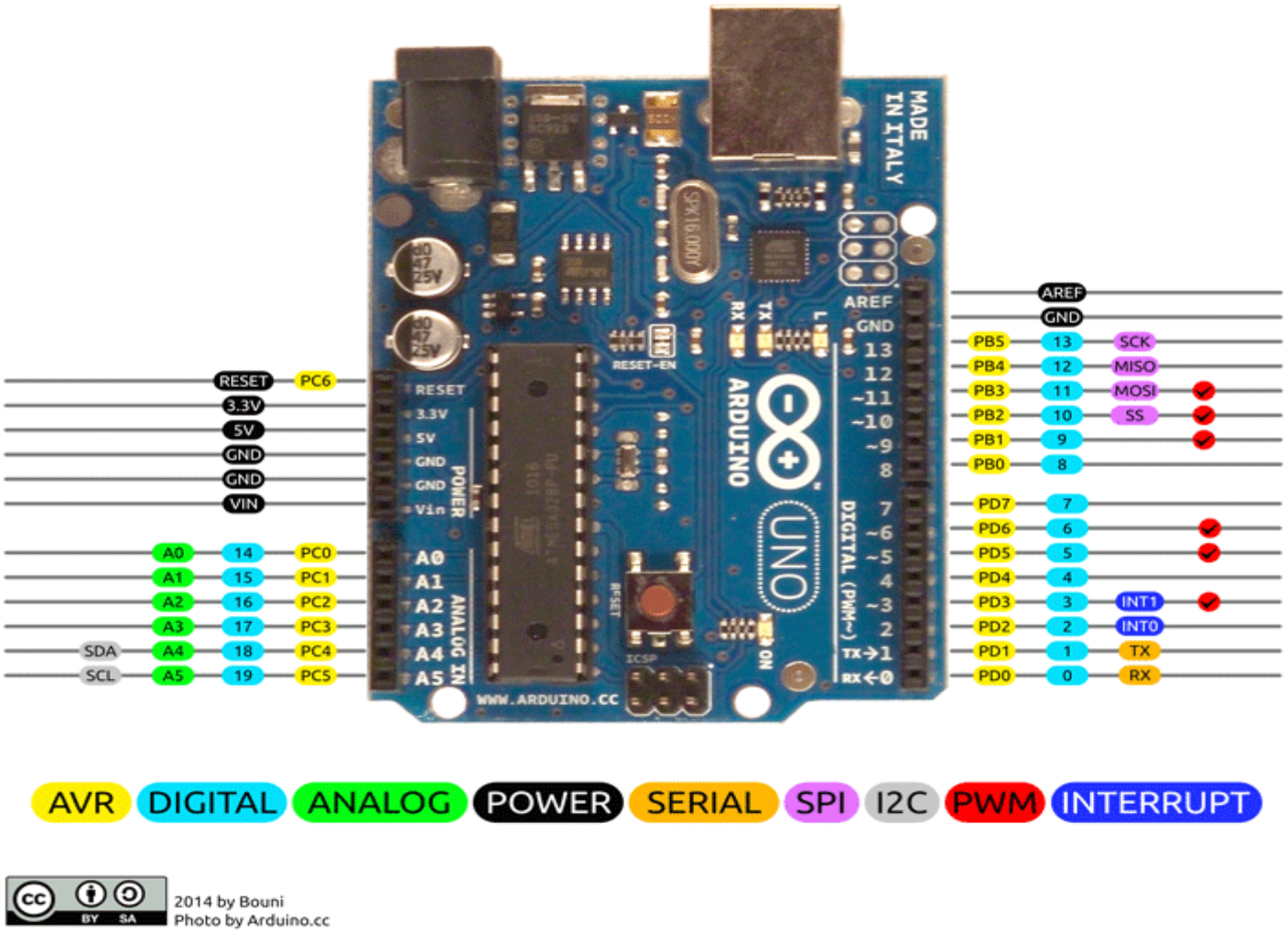


Fig.3.3 Pin Description of Arduino Uno R3

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

Arduino Uno to ATmega328 Pin Mapping

When ATmega328 chip is used in place of Arduino Uno, or vice versa, the image below shows the pin mapping between the two

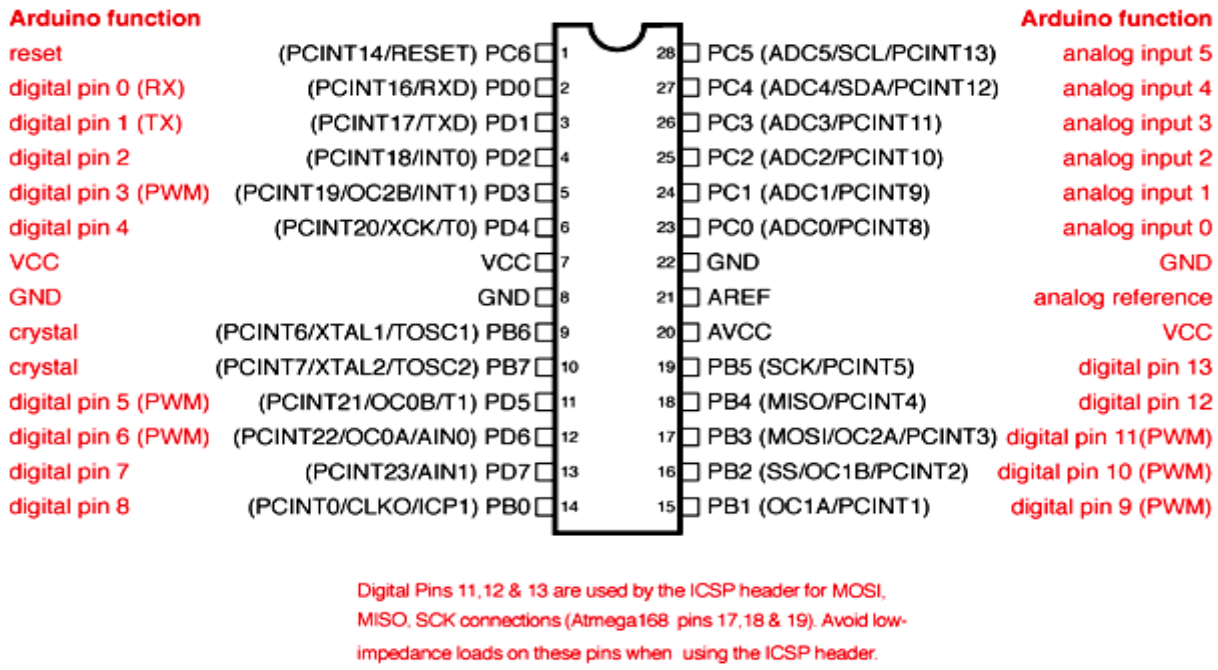


Fig.3.3 Arduino Uno to ATMEGA328 pin mapping

Applications

- Prototyping of Electronics Products and Systems
- Multiple DIY Projects.
- Easy to use for beginner level DIYers and makers.
- Projects requiring Multiple I/O interfaces and communications.

3.3 L293D Motor Driver

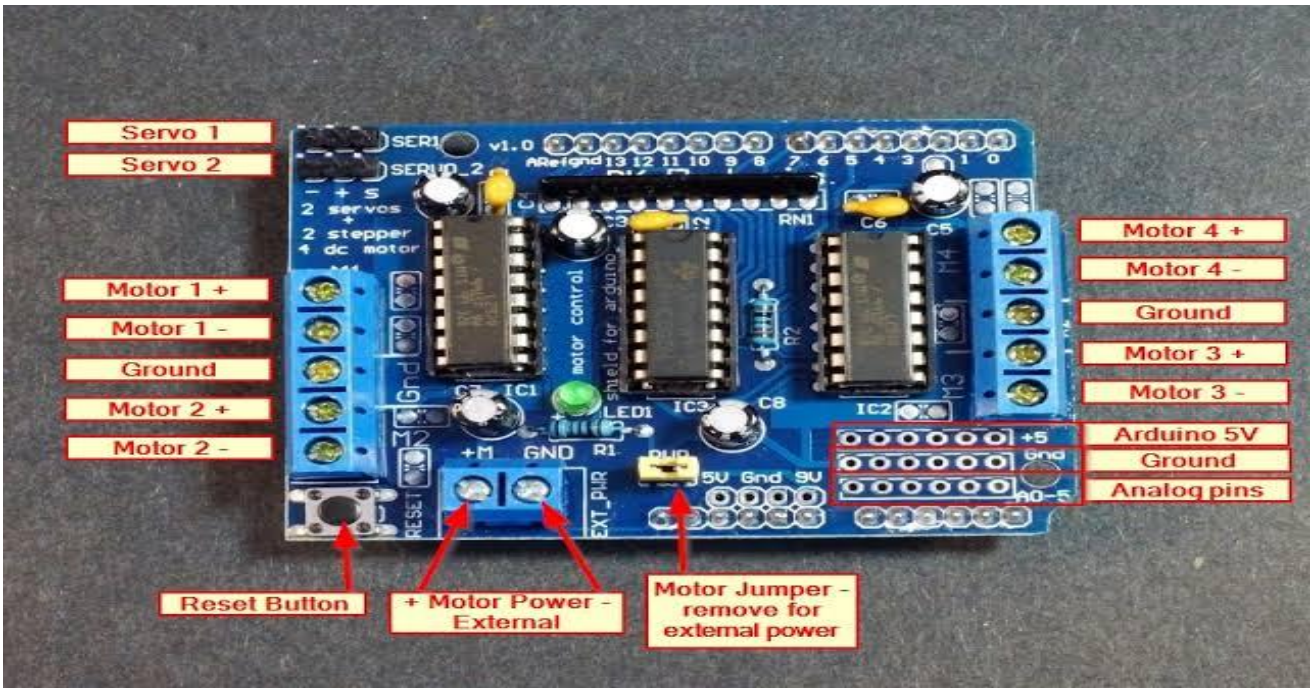


Fig.3.4 L293D Motor Driver

Sun Founder L293D is a monolithic integrated, high voltage, high current, 4-channel driver. Basically this means using this chip you can use DC motors and power supplies of up to 16 Volts, that's some pretty big motors and the chip can supply a maximum current of 600mA per channel, the L293D chip is also what's known as a type of H-Bridge. The H-Bridge is typically an electrical circuit that enables a voltage to be applied across a load in either direction to an output, e.g. motor.

The schematic diagram is as follows

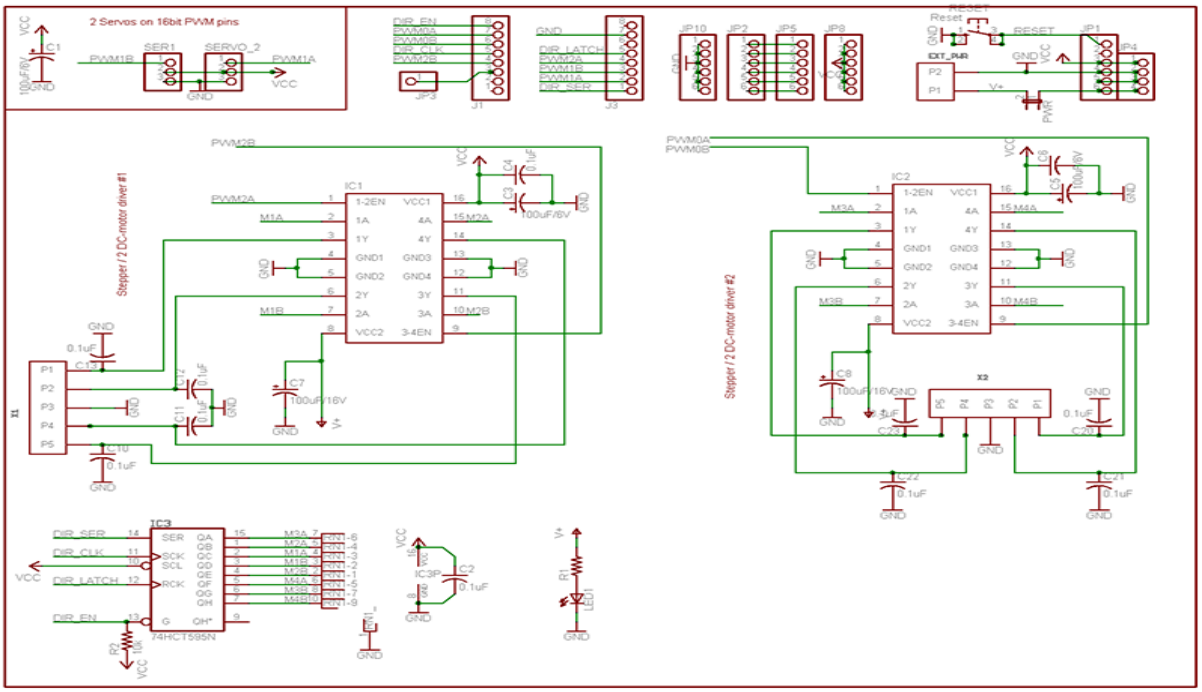
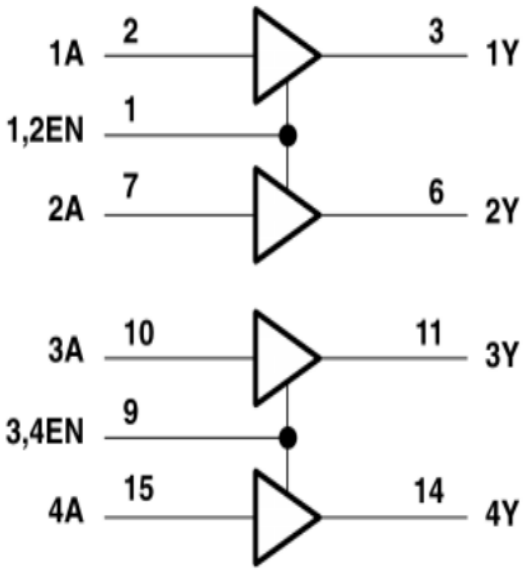


Fig.3.5 Schematic Diagram of L293D motor driver

The L293D is quadruple high-current half-H drivers. It is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications



Logic diagram

FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)
† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

Function Table

Fig.3.6 Logic Diagram and function table

3.3.1 L293D Motor Driver Pin description

Enable1,2	Enable pin to control 1,2 driver
Input 1A	Input to control 1Y
Output 1Y	Output,connect to motor
GND	Ground and heat sink
GND	Ground and heat sink
Output 2Y	Output,connect to motor

Input 2A	Input to control 2Y
Vcc2	Output supply voltage
Enable3,4	Enable pin to control 3,4 driver
Input 3A	Input to control 3Y
Output 3Y	Output,connect to motor
GND	Ground and heat sink
GND	Ground and heat sink
Output 4Y	Output,connect to motor
Input 4A	Input to control 4Y
Vcc1	Supply voltage(7 max)

3.3.2 Features

- Wide Supply-Voltage Range: 4.5 V to 36
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Output Current 600 mA Per Channel
- Peak Output Current 1.2 A Per Channel

- 8-Bit Serial-In, Parallel-Out Shift
- Wide Operating Voltage Range of 2 V to 6 V
- High-Current 3-State Outputs Can Drive Up To 15 LSTTL Loads
- Low Power Consumption, 80-µA Max ICC
- Typical tpd = 13 ns
- Low Input Current of 1
- Shift Register Has Direct Clear

3.3.3 Specification of L293D Motor Driver Shield:

- Operating Voltage : 5V to 12V.
- Motor controller: L293D, Drives 2 DC motors or 1 stepper motor.
- Max current: 600mA per channel.
- Peak Output Current : 1.2 Amp.

Applications of L293D Motor Driver Shield:

- It is used by arduino users.
- Multiple DIY Projects.

H-Bridge L293D Dual H-Bridge Motor Driver :

L293 series of chips are power H-bridge motor drivers. The L293D chip is in 16-pins dip packages, and has two h-bridge drivers. An H bridge is typically capable of running two DC motor bidirectional (forward, backward), or two separate motors bidirectional. Thus a L293 chip can run two motors bidirectional, or 4 unidirectional

3.3.4 Component description of L293D Motor Driver Shield:

Motor indicator:

In this shield we connected four led as a motor indicator, which shows the rotation of motor either in clockwise direction or anticlockwise direction. Led 0 and led 1 is connected with motor M1 and led 2 and 3 is connected to motor M2. Motors and led are connected to the same pin. LED L0 & L1 0R MOTOR M1- PIN 9 & PIN10 LED L2 &L3 OR MOTOR M2- PIN 11& PIN 12

Power led:

In our shield we use five led in which one is used for power indicator when supply is connected then led will glows.

Analog I/O port:

Port having 6 pins. We can use this port as an input and output for analog devices, all the analog devices or sensor should be connected at this port.

Digital I/O Port:

In our shield there are 14 pin for interfacing the digital devices at that digital port. We can use this port as a input and as a output.

Power port:

In this port we have a six pin. One pin for Vcc supply by the circuit two pin used for ground, and next two for external power supply for 5 volt and 3 volt. Last pin of power port is reset.

Sensor connector:

There are two sensor connectors to connect the sensor. Connector having 3pins. First pin used to connect the sensor with the power supply, second pin is used to connect it with the ground, and third pin is used to transfer the data. - Left sensor is connected to analog port A pin no 0 i.e. (A0) -Right sensor is connected to analog port A pin no 3 i.e. (A3)

Motor connector:

In this circuit there are two motor connectors. So we can drive two motor by this circuit. Each motor connector having a set of four pins, first two pins used to connect positive wire of motor and other two pins are used to connect the negative wire of motor. By this connector pins we can drive our each motor either in clockwise direction or in anticlockwise.

Motor selector jumper:

As per the jumper application it is used to short the two connection. Motor selector jumper short the motor with the 5v supply or we can short the motor to the USB power supply.

Reset switch:

It is used to reset the program when reset switch is pressed then program routine will stop and execute from the beginning. Sometimes errors are introduced during the program execution so to remove unwanted routine we used reset switch.

3.4 HC-05 Bluetooth Module:

- Bluetooth is used for wireless headset, game controllers, wireless mouse and many consumer applications.
- Its range is less than 100m but depends on transmitter & receiver, atmosphere conditions.
- IEEE 802.15.1 protocol for Bluetooth / WPAN.
- Uses frequency hopping spread spectrum to send data over air
- FHSS is used to transmitter signals by changing carrier frequency among many different frequencies of large band.
- Serial communication to communicate with devices it means it transmits data bit one by one.
- This can be used in a master and slave configuration (in this one device controls one or more other devices for communication)

Why we use bluetooth?

It is easy to use Bluetooth Serial Port Protocol (it transfers in or out bit by bit)Which makes an easy way to interface with controller or device.

3.4.1 Bluetooth Versions:

Specification:

- Bluetooth v1.0 to v1.08
- Mandatory Bluetooth hardware device and address
- Bluetooth v1.1 - IEEE standard 802.15.1-2002
- Bluetooth v1.2 - Faster connection
- Bluetooth v2.0+EDR - Enhanced data rate
- Bluetooth v2.1 - Secure simple pairing
- Bluetooth v3.0 - High-speed data transfer

- Bluetooth v4.0 - Low energy consumption recently used in apple I – phone 4s

The Bluetooth version we used here is v2.0+edr (enhanced data rate) which means it provides faster data rate and possibly improves battery life (increases data rate potential up-to 3mbps)

3.4.2 Types of Bluetooth modules:

- HC05
- HC06
- BLE link bee
- BLE mini
- Blue SMiRF
- JY MCU
- ITEAD BT
- Shield Bluetooth RL 42
- Blue fruit EZ-Link

HC05

- The HC-05 module is cheap and designed for transparent wireless serial connection set-up with Bluetooth SPP (Serial Port Protocol).
- If u r at the beginning and want to use Bluetooth module.
- Choosing Bluetooth depends upon role of operation of the module and this can be used as both master and slave role while others work in only role.
- It provides faster data rate when compared remaining modules.

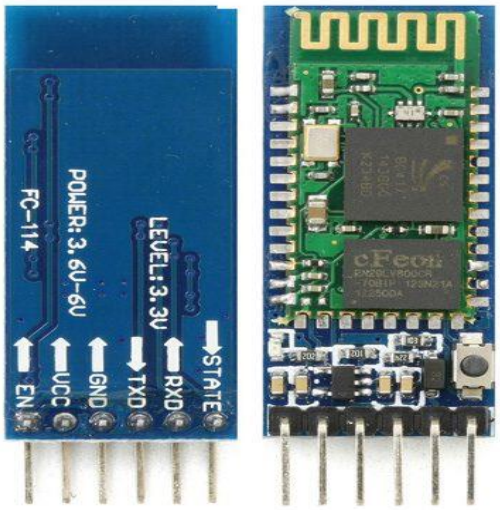


Fig.3.7 Bluetooth Module

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

It has 6 pins:

- **Key/EN:** It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode

HC-05 module has two modes:

1. Data mode: Exchange of data between devices.
2. Command mode: It uses AT commands which are used to change setting of HC- 05. To send these commands to module serial (USART) port is used.

- **VCC:** Connect 5 V or 3.3 V to this Pin.
- **GND:** Ground Pin of module.
- **TXD:** Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
- **RXD:** Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
- **State:** It tells whether module is connected or not.

HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.This module

works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.

- Send data from Smartphone terminal to HC-05 Bluetooth module and see this data on PC serial terminal.
- To communicate smartphone with HC-05 Bluetooth module, smartphone requires Bluetooth terminal application for transmitting and receiving data.

3.5 LCD(Liquid Crystal Display)

- LCD is an electronic display module which uses liquid crystal to produce a visible image.
- LCD 16x2 display is a very basic module commonly used in circuits.
- This LCD displays 16 characters per line in 2 such lines. And it is displayed in a 5x7 pixel matrix.

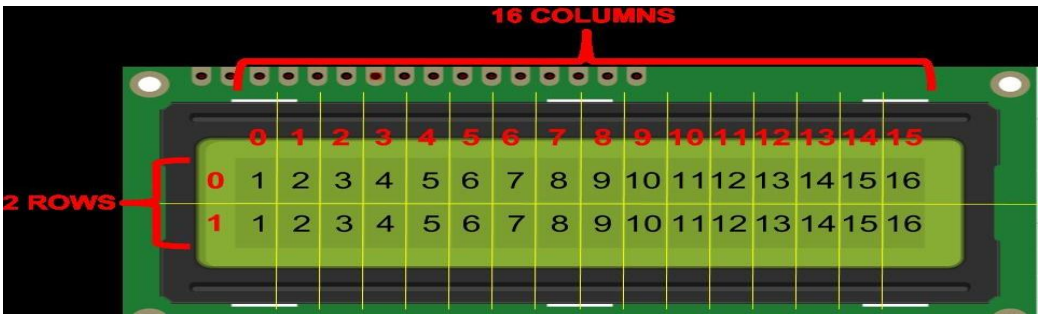


Fig.3.8 Liquid Crystal display

Why we use LCD display:

- LCD uses liquid crystal to produce a visible image.
- And these are super thin technology display screen they are generally used in laptop, computer screens, TV's etc.
- This technology allows displays to be much thinner when compared to cathode ray tube technology.

3.5.1 LCD pin description:

Pin number	Function	Name
1	Ground (0v)	Ground
2	ıpply voltage (4.7v-5.3v)	Vcc
3	ıntrast adjustment; the best way is to use a variable resistorsuch as a potentiometer. The output of the potentiometer is connected to this pin. Rotate the potentiometer knobforward and backwards to adjust the LCD contrast.	V0/VEE
4	lects command register when low, and data register when high	Register Select
5	w to write to the register; High to read from the register	Read/Write
6	nds data to data pins when a high to low pulse is given; Extra voltage push is required to execute the instruction and EN(enable) signal is used for this purpose. Usually, we make it en=0 and when we want to execute the instruction we make it high en=1 for some milliseconds. After this we again make it ground that is, en=0.	Enable
7		DB0
8		DB1
9		DB2
10	8-Data pins	DB3
11		DB4
12		DB5
13		DB6

14		DB7
15	Black light VCC(5V)	Led+
16	Black Ground(0V)	Led-

3.5.2 Advantages of LCD when compared to other display devices:

1.Sharpness:

Image is perfectly sharp at the native resolution of the panel. LCDs using an analog input require careful adjustment of pixel tracking/phase.

2.Geometric Distortion:

Zero geometric distortion at the native resolution of the pixel. Minor distortion for other resolution because image must be rescaled.

3.Brightness:

High peak intensity produces very bright images. Best for brightly lit images.

4.Screen Shape:

Screens are perfectly flat.

5.Physical:

Thin, with a small foot print. Consume little electricity and produces little heat. This usually the first display most people use when they first start using the Arduino board. It displays 16 characters in 2 rows (there are also other sizes available). These displays come with a blue or green background and with a backlight.

3.5.3 Types of Arduino compatible LCD displays:

- TFT LCD Display
- TFT LCD Touch screen Display
- Dot matrix
- White OLED Display
- 16x2 character LCD Display
- Bits Digital Tube LED display

3.6 Ultrasonic Sensor

Ultrasonic transducers or ultrasonic sensors are a type of acoustic sensor divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions.

Fig.3.9 Ultrasonic Sensor

The design of transducer can vary greatly depending on its use: those used for diagnostic purposes, for example the range-finding applications listed above, are generally lower power than those used for the purpose of changing the properties of the liquid medium, or targets immersed in the liquid medium, through chemical, biological or physical (e.g. erosive) effects. The latter class include ultrasonic probes and ultrasonic baths, which apply ultrasonic energy to agitate particles, clean, erode, or disrupt biological cells, in a wide range of materials.

3.6.1 Applications and performance:

Ultrasound can be used for measuring wind speed and direction (anemometer), tank or channel fluid level, and speed through air or water. For measuring speed or direction, a device uses multiple detectors and calculates the speed from the relative distances to particulates in the air or water. To measure tank or channel liquid level, and also sea level (tide gauge), the sensor measures the distance

(ranging) to the surface of the fluid. Further applications include: humidifiers, sonar, medical ultrasonography, burglar alarms, non-destructive testing and wireless charging.

Systems typically use a transducer which generates sound waves in the ultrasonic range, above 18 kHz, by turning electrical energy into sound, then upon receiving the echo turn the sound waves into electrical energy which can be measured and displayed.This technology, as well, can detect approaching objects and track their positions.

Ultrasound can also be used to make point-to-point distance measurements by transmitting and receiving discrete bursts of ultrasound between transducers. This technique is known as Sonomicrometry where the transit-time of the ultrasound signal is measured electronically (ie digitally) and converted mathematically to the distance between transducers assuming the speed of sound of the medium between the transducers is known.

This method can be very precise in terms of temporal and spatial resolution because the time-of-flight measurement can be derived from tracking the same incident (received) waveform either by reference level or zero crossing. This enables the measurement resolution to far exceed the wavelength of the sound frequency generated by the transducers.

3.6.2 Transducer

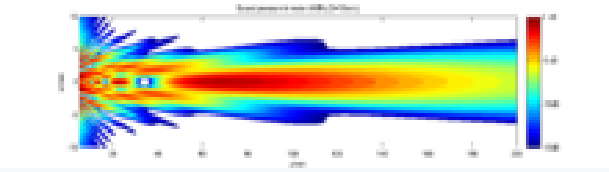


Fig.3.10 Transducer

Sound field of a non focusing 4 MHz ultrasonic transducer with a near field length of N = 67 mm in water. The plot shows the sound pressure at a logarithmic db-scale.

Sound pressure field of the same ultrasonic transducer (4 MHz, N = 67 mm) with the transducer surface having a spherical curvature with the curvature radius R = 30 mm

Ultrasonic transducers convert AC into ultrasound, as well as the reverse. Ultrasonics, typically refers to piezoelectric transducers or capacitive transducers. Piezoelectric crystals change size and shape when a voltage is applied; AC voltage makes them oscillate at the same frequency and produce ultrasonic sound. Capacitive transducers use electrostatic fields between a conductive diaphragm and a backing plate.

The beam pattern of a transducer can be determined by the active transducer area and shape, the ultrasound wavelength, and the sound velocity of the propagation medium. The diagrams show the sound fields of an unfocused and a focusing ultrasonic transducer in water, plainly at differing energy levels.

Since piezoelectric materials generate a voltage when force is applied to them, they can also work as ultrasonic detectors. Some systems use separate transmitters and receivers, while others combine both functions into a single piezoelectric transceiver.

Ultrasound transmitters can also use non-piezoelectric principles such as magnetostriction. Materials with this property change size slightly when exposed to a magnetic field, and make practical transducers.

A capacitor ("condenser") microphone has a thin diaphragm that responds to ultrasound waves. Changes in the electric field between the diaphragm and a closely spaced backing plate convert sound signals to electric currents, which can be amplified.

The diaphragm (or membrane) principle is also used in the relatively new micro-machined ultrasonic transducers (MUTs). These devices are fabricated using silicon micro-machining technology (MEMS technology), which is particularly useful for the fabrication of transducer arrays. The vibration of the diaphragm may be measured or induced electronically using the capacitance between the diaphragm and a closely spaced backing plate (CMUT), or by adding a thin layer of piezo-electric material on diaphragm (PMUT). Alternatively, recent research showed that the vibration of the diaphragm may be measured by a tiny optical ring resonator integrated inside the diaphragm (OMUS).

Use in medicine

Medical ultrasonic transducers (probes) come in a variety of different shapes and sizes for use in making cross-sectional images of various parts of the body. The transducer may be passed over the surface and in contact with the body, or inserted into a body opening such as the rectum. Clinicians

who perform ultrasound-guided procedures often use a probe positioning system to hold the ultrasonic transducer.

Air detection sensors are used in various rolesNon-invasive air detection is for the most critical situations where the safety of a patient is mandatory. Many of the variables, which can affect performance of amplitude or continuous-wave-based sensing systems, are eliminated or greatly reduced, thus yielding accurate and repeatable detection.

One key principle of in this technology is that the transmit signal consists of short bursts of ultrasonic energy. After each burst, the electronics looks for a return signal within a small window of time corresponding to the time it takes for the energy to pass through the vessel. Only signals received during this period will qualify for additional signal processing. This principle is similar to radar range gating.

Use in industry

Ultrasonic sensors can detect movement of targets and measure the distance to them in many automated factories and process plants. Sensors can have an on or off digital output for detecting the movement of objects, or an analog output proportional to distance. They can sense the edge of material as part of a web guiding system.

Ultrasonic sensors are widely used in cars as parking sensors to aid the driver in reversing into parking spaces. They are being tested for a number of other automotive uses including ultrasonic people detection and assisting in autonomous UAV navigation.

Because ultrasonic sensors use sound rather than light for detection, they work in applications where photoelectric sensors may not. Ultrasonic is a great solution for clear object detection and for liquid level measurement, applications that photoelectric struggle with because of target translucence. As well, target colour or reflectivities do not affect ultrasonic sensors, which can operate reliably in high-glare environments.

Passive ultrasonic sensors may be used to detect high-pressure gas or liquid leaks, or other hazardous conditions that generate ultrasonic sound. In these devices, audio from the transducer (microphone) is converted down to human hearing range.

High-power ultrasonic emitters are used in commercially available ultrasonic cleaning devices. An ultrasonic transducer is affixed to a stainless steel pan which is filled with a solvent (frequently water

or isopropanol). An electrical square wave feeds the transducer, creating sound in the solvent strong enough to cause cavitation.

Ultrasonic technology has been used for multiple cleaning purposes. One of which that is gaining a decent amount of traction in the past decade is ultrasonic gun cleaning.

Ultrasonic testing is also widely used in metallurgy and engineering to evaluate corrosion, welds, and material defects using different types of scans.

3.7 Mini Bread Board

In the early days of radio, amateurs nailed bare copper wires or terminal strips to a wooden board (often literally a board to slice bread on) and soldered electronic components to them. Sometimes a paper schematic diagram was first glued to the board as a guide to placing terminals, then components and wires were installed over their symbols on the schematic. Using thumbtacks or small nails as mounting posts was also common.

Breadboards have evolved over time, with the term now being used for all kinds of prototype electronic devices. For example, US Patent 3,145,483 was filed in 1961 and describes a wooden plate breadboard with mounted springs and other facilities. US Patent 3,496,419 was filed in 1967 and refers to a particular printed circuit board layout as a Printed Circuit Breadboard. Both examples refer to and describe other types of breadboards as prior art.

The breadboard most commonly used today is usually made of white plastic and is a pluggable (solderless) breadboard. It was designed by Ronald J. Portugal in 1971.

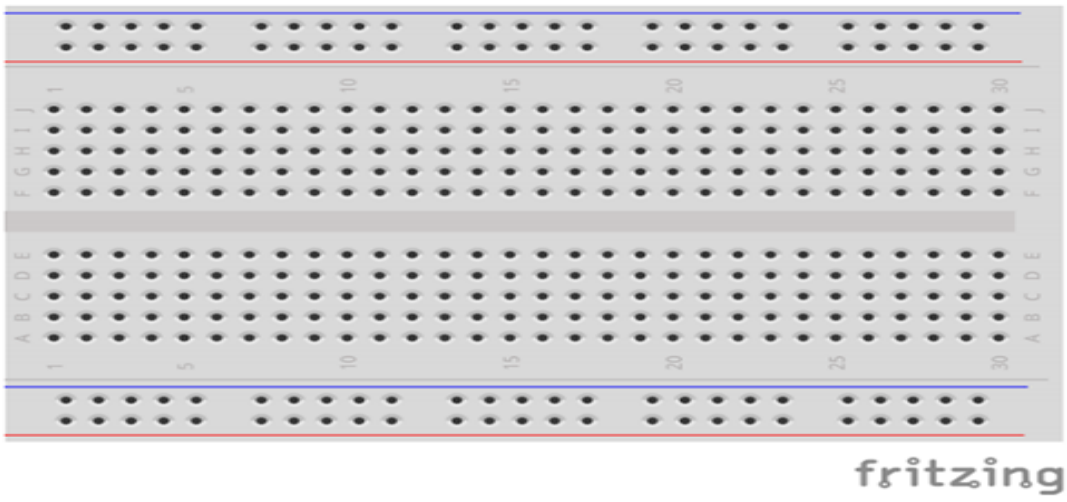


Fig.3.11 Mini Bread Board

3.7.1 Limitations

Due to relatively large parasitic capacitance compared to a properly laid out PCB (approx 2pF between adjacent contact columns), high inductance of some connections and a relatively high and not very reproducible contact resistance, solderless breadboards are limited to operation at relatively low frequencies, usually less than 10 MHz, depending on the nature of the circuit. The relatively high contact resistance can already be a problem for some DC and very low frequency circuits. Solderless breadboards are further limited by their voltage and current ratings.

Very complex circuits can become unmanageable on a solderless breadboard due to the large amount of wiring required. The very convenience of easy plugging and unplugging of connections also makes it too easy to accidentally disturb a connection, and the system becomes unreliable. It is possible to prototype systems with thousands of connecting points, but great care must be taken in careful assembly, and such a system becomes unreliable as contact resistance develops over time. At some point, very complex systems must be implemented in a more reliable interconnection technology, to have a likelihood of working over a usable time period.

3.8 Jumper wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. Though jumper wires come in a variety of colors, the colors don't actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power.

3.8.1 Types of Jumper Wires

Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can

plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you'll need.

Fig.3.12 Jumper wires

3.9 Servo Motor

A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

Fig.3.13 Servo Motor

Servos are extremely useful in robotics. The motors are small, have built-in control circuitry, and are extremely powerful for their size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, does not consume much energy.

The guts of a servo motor is shown in the following picture. You can see the control circuitry, the motor, a set of gears, and the case. You can also see the 3 wires that connect to the outside world. One is for power (+5volts), ground, and the white wire is the control wire.

3.9.1 Working of a Servo Motor:

The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) connected to the output shaft. In the picture above, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor.

If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor until it is at a desired angle. The output shaft of the servo is

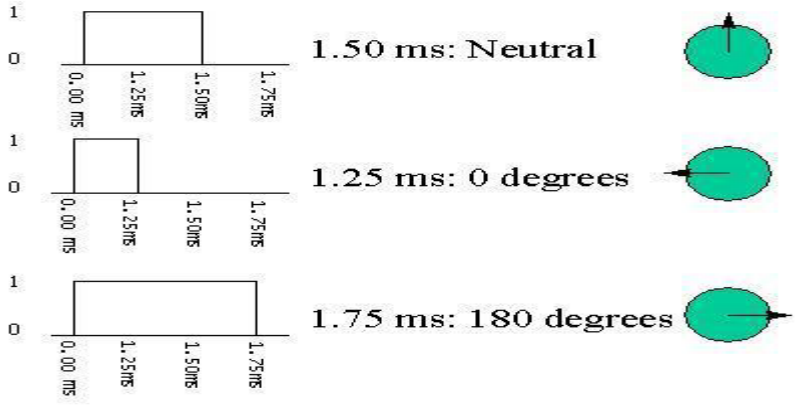
capable of traveling somewhere around 180 degrees. Usually, it is somewhere in the 210-degree range, however, it varies depending on the manufacturer.

A normal servo is used to control an angular motion of 0 to 180 degrees. It is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

The power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called **proportional control**.

How to Communicateto the Angle at Which the Servo Should Turn?

The control wire is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called **Pulse Coded Modulation**. The servo expects to see a pulse every 20 milliseconds (.02 seconds). The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90-degree position (often called as the neutral position). If the pulse is shorter than 1.5 milliseconds, then the motor will turn the shaft closer to 0 degrees. If the pulse is longer than 1.5 milliseconds, the shaft turns closer to 180 degrees.



3.10 Robotic Kit

A **robot kit** is a special construction kit for building robots, especially autonomous mobile robots.

Toy robot kits are also supplied by several companies. They are mostly made of plastics elements like Lego Mindstorms, rero Reconfigurable Robot kit, the Robotics Bioloid, Robo builder, the ROBO-BOX-3.0 (produced by Inex), and the lesser-known KAI Robot (produced by Kaimax), or aluminium elements like Lynx motion's Servo Erector Set and the qfix kit. Some robots, such as Ebdot, come ready-assembled.

Fig.3.14 Robotic Kit

3.11 Ultrasonic Sensor Holder

It is used to hold the sensor and is used to connect ultrasonic sensor and servomotor as a support. It helps the servo motor to rotate with ease. So, with the help of this holder ultrasonic sensor and servomotor can perform their functions with great flexibility.

Fig.3.15 Ultrasonic sensor Holder

CHAPTER-4

PROGRAM

To design voice controlled obstacle detection and distance measurement robotic car, the programming has to be done to execute three features of robotic car. The programming to the arduino is done in embedded C language it can also be done in python. The programming is done in arduino software.

4.1 Brief introduction to arduino software

The open-source Arduino Software (IDE) Integrated Development Environment makes it easy to write a code and upload it to the board.It runs on Windows, Mac OS X, and Linux.The environment is written in java, C, C++ and based on processing and other open source software.This Arduino software can be used with any arduino board.

A program for arduino hardware maybe written in any programming language with compilers that produce binary machine code for the target processor.Atmel provides a development environment for their 8-bit AVR and 32-bit ARM cortex-M based microcontrollers.The open source nature of the arduino project has facilitated the publication of many free software libraries that other developers used to augment their projects.

4.2 The program to execute the voice controlling car is as follows:

```
#include <AFMotor.h>
#include <Servo.h>
String voice;
AF_DCMotor motor1 (1, MOTOR12_1KHZ);
AF_DCMotor motor2 (2, MOTOR12_1KHZ);    Servo my Servo;
Void setup()
{
Serial.begin(9600);
myServo.attach(10);
```

```
  myServo.write(90);                    //servo position is 90 degrees
  }
  void loop()
  {
  while (Serial.available())

  {
                                        //Check if there is an available byte to read
delay(10);                              //Delay added to make thing stable
  char c = Serial.read();               //Conduct a serial read
  if (c == '0X0A') {break;}             //Exit the loop when the # is detected after
  the word
  voice += c;                           //Shorthand for voice = voice + c
    }
  if (voice.length() > 0)
  {
  if(voice == "forward")
  {
  forward_car();
    }
  else if(voice == "backward")
  {
  back_car();
   }
  else if(voice == "right")
  {
  right_car();
   }
  else if(voice == "left")
  {
  left_car();
```

```
 }
else if(voice == "stop")
{
stop_car();
 }
voice="";                                   //Reset the variable after initiating
 }
}
 Void forward_car()
{

motor1.run(FORWARD);
motor1.setSpeed(500);
motor2.run(FORWARD);
motor2.setSpeed(500);
delay(10000);
motor1.run(RELEASE);
motor2.run(RELEASE);
}
 Void back_car()
{
motor1.run(BACKWARD);
motor1.setSpeed(500);
motor2.run(BACKWARD);
motor2.setSpeed(500);
delay(10000);
motor1.run(RELEASE);


motor2.run(RELEASE);
}
```

```
voidright_car()
{
myServo.write(0);
delay(500);
myServo.write(90);
delay(500); motor1.run(BACKWARD);
motor1.setSpeed(250);
motor2.run(FORWARD);
motor2.setSpeed(250);
delay(10000);
motor1.run(RELEASE);
motor2.run(RELEASE);
}
 voidstop_car ()
{
motor1 .run(RELEASE);
motor2.run(RELEASE);
```

4.3 Program to execute obstacle detection feature of robotic car is as follows:

```
#include <AFMotor.h>
#include <Servo.h>
#include <NewPing.h>
#define TRIG_PIN 12
#define ECHO_PIN 13
#define MAX_DISTANCE_POSSIBLE 1000
#define MAX_SPEED 150 //
#define MOTORS_CALIBRATION_OFFSET 3
#define COLL_DIST 20
#define TURN_DIST COLL_DIST+10
```

```
NewPingsonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE_POSSIBLE);
AF_DCMotorleftMotor(1, MOTOR12_1KHZ);
AF_DCMotorrightMotor(2, MOTOR12_1KHZ);
Servo neckControllerServoMotor;
intpos = 0;
intmaxDist = 0;
intmaxAngle = 0;
intmaxRight = 0;
intmaxLeft = 0;
intmaxFront = 0;
int course = 0;
intcurDist = 0;
String motorSet = "";
intspeedSet = 0;
void setup() {
neckControllerServoMotor.attach(10);
delay(2000);
checkPath();
motorSet = "FORWARD";
neckControllerServoMotor.write(90);

moveForward();
}
 void loop() {

checkForward();
checkPath();
}
 voidcheckPath() {
intcurLeft = 0;
intcurFront = 0;
```

```
intcurRight = 0; intcurDist = 0;
neckControllerServoMotor.write(144);
delay(120);
for(pos = 144; pos>= 36; pos-=18) {
neckControllerServoMotor.write(pos);
delay(90);
checkForward();
curDist = readPing()
if (curDist< COLL_DIST) {
checkCourse();
break;
 }
if (curDist< TURN_DIST) {
changePath();
 }
if (curDist>curDist) {maxAngle = pos;}
if (pos> 90 &&curDist>curLeft) { curLeft = curDist;}
if (pos == 90 &&curDist>curFront) {curFront = curDist;}
if (pos< 90 &&curDist>curRight) {curRight = curDist;}
 }
maxLeft = curLeft; maxRight = curRight;
maxFront = curFront;
}  voidsetCourse() {
if (maxAngle< 90) {turnRight();}
if (maxAngle> 90) {turnLeft();}

maxLeft = 0;
maxRight = 0;
maxFront = 0;
}
```

```
voidcheckCourse() {
moveBackward();
delay(500);
moveStop();
setCourse();
}
voidchangePath() {
if (pos< 90) {lookLeft();}
if (pos> 90) {lookRight();}
} intreadPing() {
delay(70);
unsignedintuS = sonar.ping();
int cm = uS/US_ROUNDTRIP_CM;
return cm;
}
Void checkForward() {
if (motorSet=="FORWARD")  {   leftMotor.run(FORWARD)
rightMotor.run(FORWARD); } }
Void checkBackward() { if(motorSet=="BACKWARD")
{leftMotor.run(BACKWARD);
rightMotor.run(BACKWARD); } }
voidmoveStop() {leftMotor.run(RELEASE); rightMotor.run(RELEASE);}
voidmoveForward() {
motorSet = "FORWARD";
leftMotor.run(FORWARD);
rightMotor.run(FORWARD);
for (speedSet = 0; speedSet< MAX_SPEED; speedSet +=2)
{
leftMotor.setSpeed(speedSet+MOTORS_CALIBRATION_OFFSET);
rightMotor.setSpeed(speedSet);
delay(5);
```

```
}
}
voidmoveBackward() {
motorSet = "BACKWARD";
leftMotor.run(BACKWARD);

rightMotor.run(BACKWARD);
for (speedSet = 0; speedSet< MAX_SPEED; speedSet +=2)
{
leftMotor.setSpeed(speedSet+MOTORS_CALIBRATION_OFFSET);
rightMotor.setSpeed(speedSet);
delay(5);
}
}
voidturnRight()
{
motorSet = "RIGHT";
leftMotor.run(FORWARD);
rightMotor.run(BACKWARD);
delay(400);
motorSet = "FORWARD";
leftMotor.run(FORWARD);
rightMotor.run(FORWARD);
}
 voidturnLeft()
{
motorSet = "LEFT";
leftMotor.run(BACKWARD);
rightMotor.run(FORWARD);
delay(400);
motorSet = "FORWARD";
```

```
leftMotor.run(FORWARD);
rightMotor.run(FORWARD);
}
 voidlookRight() {rightMotor.run(BACKWARD); delay(400);
rightMotor.run(FORWARD);}
voidlookLeft() {leftMotor.run(BACKWARD); delay(400); leftMotor.run(FORWARD);
}
```

4.4 Program to display the distance of obstacle on LCD is as follows:

```
//HC-SR04 Ultrasonic Sensor with LCD dispaly
HC-SR04 Ultrasonic Sensor
VCC to Arduino 5V
GND to Arduino GND
Echo to Arduino pin 12
Trig to Arduino pin 13
LCD Display (I used JHD162A)
VSS to Arduino GND
VCC to Arduino 5V
VEE to Arduino GND
RS to Arduino pin 11
R/W to Arduino pin 10
E to Arduino pin 9
DB4 to Arduino pin 2
DB5 to Arduino pin 3
DB6 to Arduino pin 4
 DB7 to Arduino pin 5
 LED+ to Arduino 5V
 LED- to Arduino GND//
#include <LiquidCrystal.h>              //Load Liquid Crystal Library
```

```
LiquidCrystalLCD(11,10,9,2,3,4,5);  //Create Liquid Crystal Object called LCD
 #define trigPin 13                  //Sensor Echo pin connected to Arduino pin 13
#define echoPin 12                   //Sensor Trip pin connected to Arduino pin 12
void setup()
{
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
LCD.begin(16,2);                     //Tell Arduino to start your 16 column 2 row LCD
LCD.setCursor(0,0);                  //Set LCD cursor to upper left corner, column 0, row
0
LCD.print("Target Distance:");      //Print Message on First Row
}
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;
LCD.setCursor(0,1);                   //Set cursor to first column of second row
LCD.print("    ");                    //Print blanks to clear the row
LCD.setCursor(0,1);                   //Set Cursor again to first column of second row
LCD.print(distance);                  //Print measured distance
LCD.print(" cm");                     //Print your units.
delay(250);                          //pause to let things settle
}
```

CHAPTER-5

SOFTWARE APPLICATION

5.1 ANDROID STUDIO:

To communicate with the HC-05 bluetooth module through our android device we need an application which converts our speech format into the text format and transmits it via bluetooth. To design such application we are using android studio software were we can develop an app from scratch. There are multiple ways to approach Android Development but by far the most official and powerful is to use Android Studio. This is the official IDE (Integrated Development Environment) for the Android platform, developed by Google and used to make the majority of the apps that you probably use on a daily basis. Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJIDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and Git Hub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

- As an IDE then, Android Studio's job is to provide the interface for you to create your apps and to handle much of the complicated file-management behind the scenes. The programming language you will be using is either Java or Kotlin. If you choose Java, this will be installed separately on your machine. Android Studio is simply where you will write, edit and save your projects and the files that comprise said projects. At the same

time, Android Studio will give you access to the Android SDK or 'Software Development Kit'. Think of this as an extension to the Java code that allows it to run smoothly on Android devices and take advantage of the native hardware. Java is needed to write the programs, the Android SDK is needed to make those programs run on Android and Android Studio has the job of putting it all together for you. At the same time, Android Studio also enables you to run your code, either through an emulator or through a piece of hardware connected to your machine. You'll then also be able to 'debug' the program as it runs and get feedback explaining crashes etc. so that you can more quickly solve the problem.

5.2 Starting a new project:

Once Android Studio is up and running, you'll want to dive in and create a new project. You can do this by launching Android Studio and then selecting New Project, or you can choose File > New > New Project at any time from the IDE itself.

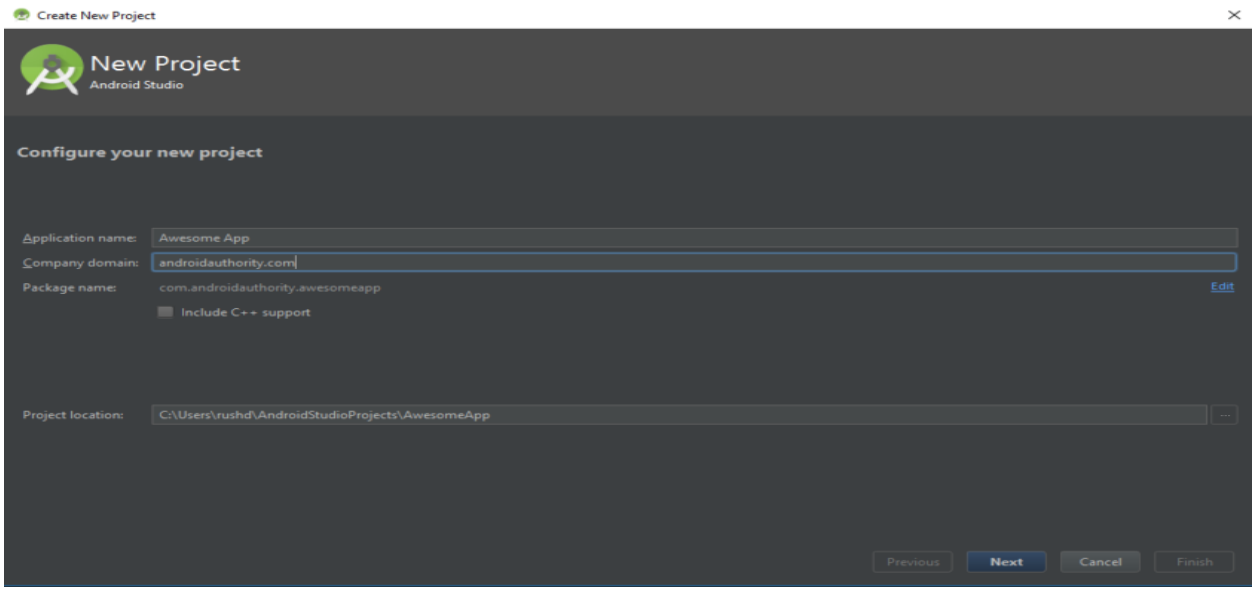


Fig.5.1 Starting a new project of an app

You'll then have the opportunity to choose from a number of different types of activity. Activities are effectively 'screens' in an app. In some cases, this will be the entire app or in others, your app might transition from one screen to the next. You're free to start a new project with no activity (in which case, you would choose 'Add No Activity') but you'll almost always want one, so it's easier to let Android Studio set you up with something resembling a blank app template to begin with.

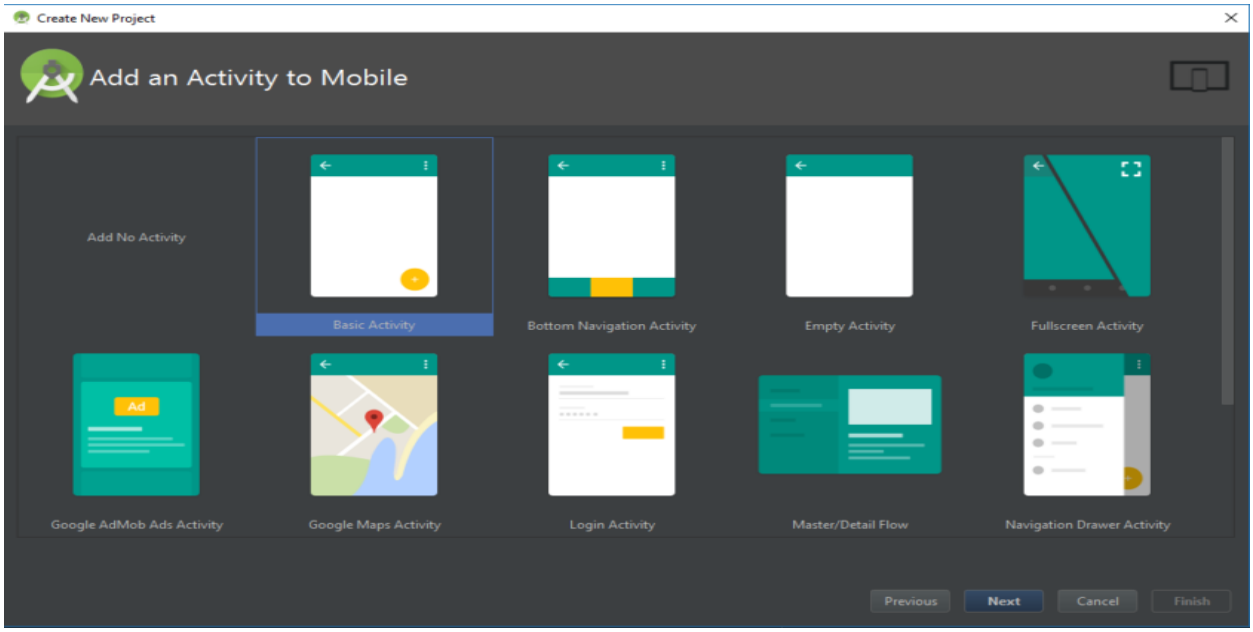


Fig.5.2 Adding activity to Mobile

Often you'll choose a 'Basic Activity', which is the default look and feel for a new Android App. This will include a menu in the top right corner, as well as a FAB button – Floating Action Button – which is a design choice that Google is trying to encourage. An 'Empty Activity' is the same thing but without the added chrome.

Here we are using an empty activity because it is a basic activity area where we can keep all our content in the main activity or we can use sub activities where we need to allocate an specific function for each of that sub activity as we are using in our application

In our application we are arranging a bluetooth icon in the main activity. This works similar to

the main bluetooth

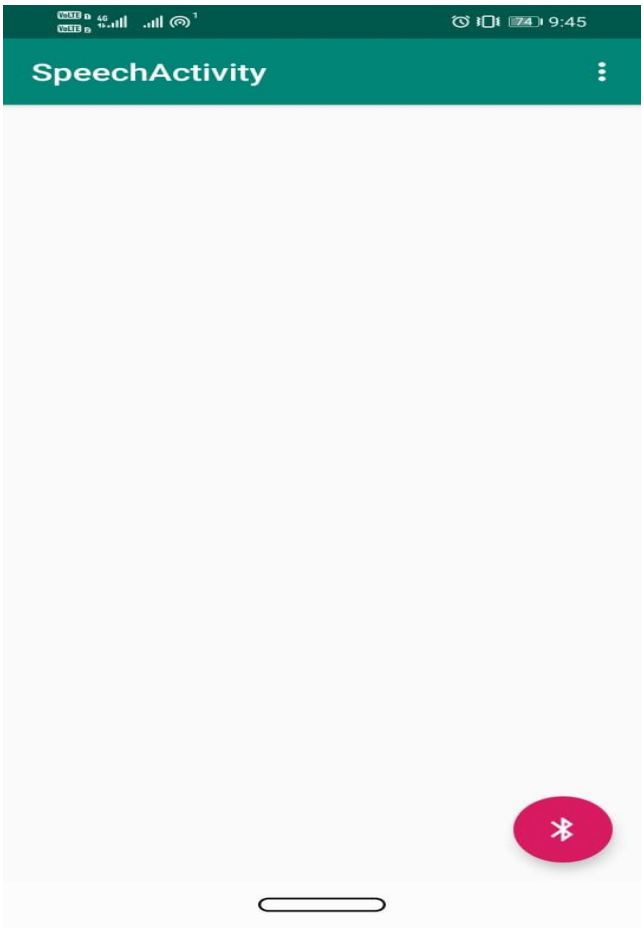


Fig.5.3 Speech Activity

In our application we are arranging a bluetooth icon in the main activity. This works similar to the main bluetooth

When we click on the bluetooth image the applications asks us to enable the bluetooth. When we allow that it starts searching all the nearby available

devices.

Then at the next sub activity area we can see all the available devices present. Now we need to select the required bluetooth device form there that is HC-05.

After clicking on required bluetooth device an new sub activity opens where their is an mic icon kept at center.That mic icon is directly connected to the google voice assistant. When we click on the mic icon it goes to the google voice assistant and then we can give our commands in the form of voice such as Forward,Backward,Left and Right.The voice commands are being converted into text formate and are being displayed on the android device are transmitted to the HC-05 bluetooth module.

Fig.5.4 Inserting mic button to the app

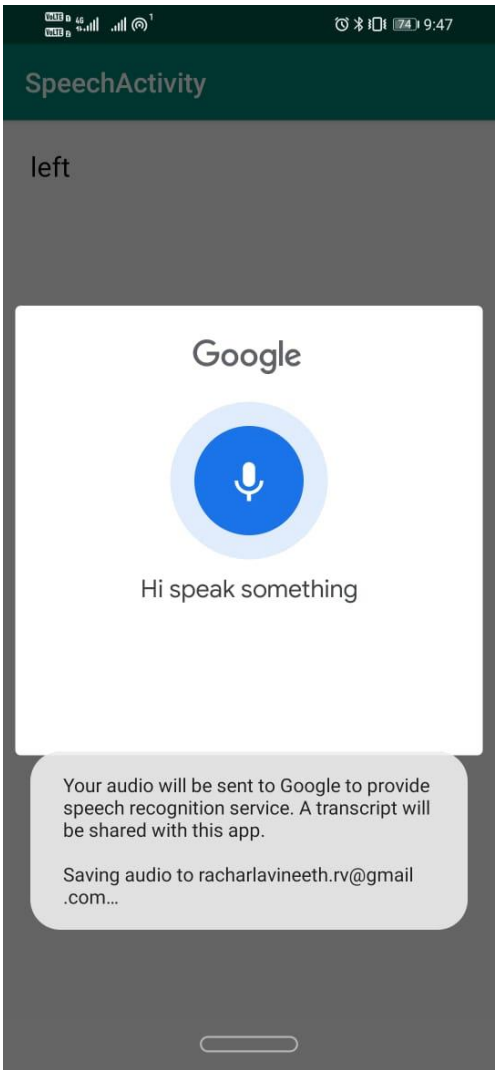


Fig.5.5 Giving voice commands to the app

CHAPTER-6

SANPSHOT OF RESULTS

Hardware design of robotic car includes designing and interfacing all the hardware components and bring them to the structure of car and the connections should be done with extreme care or else it may lead to damage of hardware components. The block diagram of hardware structure is as follows:

6.1 Block diagram

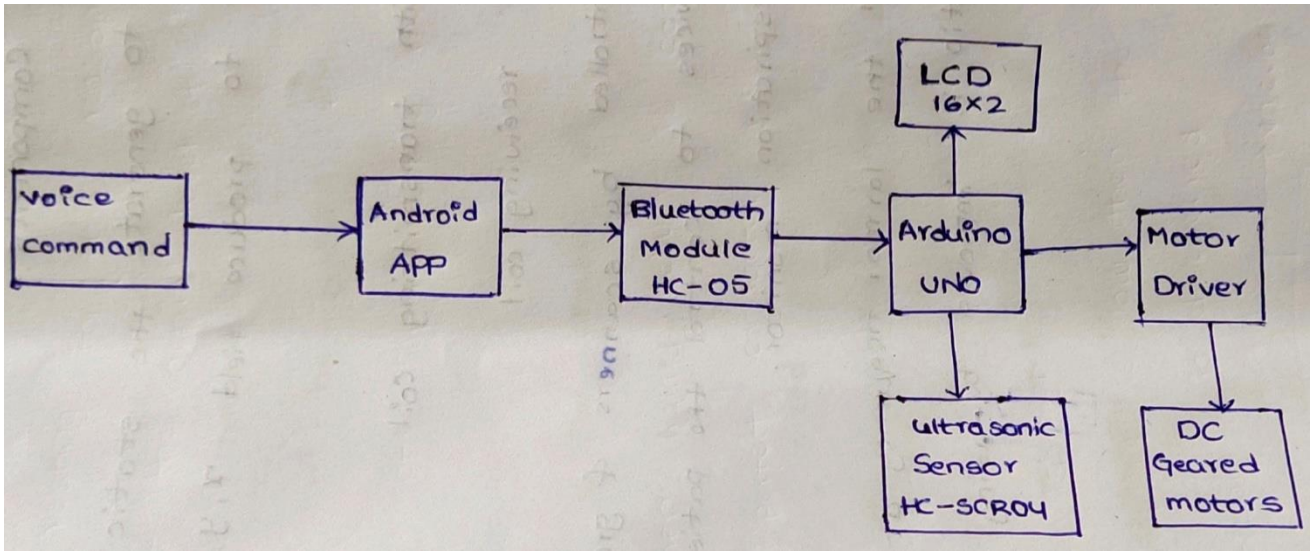


Fig.6.1 Block Diagram

The block diagram describes the connection of components. The main component of robotic car is arduino uno r3. Arduino is connected with motor driver, bluetooth module, LCD, servo motor. The motor driver is connected to motors which are connected to wheels of the car. Servo motor is connected to ultrasonic sensor. The movement of servo motor helps ultrasonic sensor to detect the obstacle in the way of car.

6.2 Working of robotic car

- The main aim of this project is to design an car mainly works through voice commands which automatically detects the obstacle coming in front of it by

measuring the distance between obstacle and car and display it on the LCD board.

- In this project we mainly use voice commands to move the car form source to destination for that we use an software application which is being installed in an android device
- This application receives the voice commands through Google assistant and sends them through the Bluetooth present in the android device to the HC-05 bluetooth module which is being kept on the car
- Initially the car is positioned at a source and it should reach to a destination where we have some obstacles in between the path.
- Now we use command forward through the application and that command is being received by the HC-05 bluetooth module which sends that information to the arduino board.
- Now arduino board process it according to the code and moves the motors using motor shield and continuously checks if any obstacle is present in front of the car using ultrasonic sensor. In this we mainly have two cases.

CASE 1

- If there is any obstacle in front of the car then the ultrasonic sensor receives the echo and that is being received at pin 12 in arduino board and as present in the code the ultrasonic sensor automatically checks both left and right directions using servo motor to find any obstacle and selects the direction where the obstacle is at the maximum position and moves in that direction this process works continuously if an object comes in front of the car. You can change the direction of the car through the voice command also but it takes more time to process the voice command than detecting through ultrasonic sensor. When ever the ultrasonic sensor detects an object the distance between the car and the obstacle is being displayed on the 16*2 LCD display it provides an additional advantage.

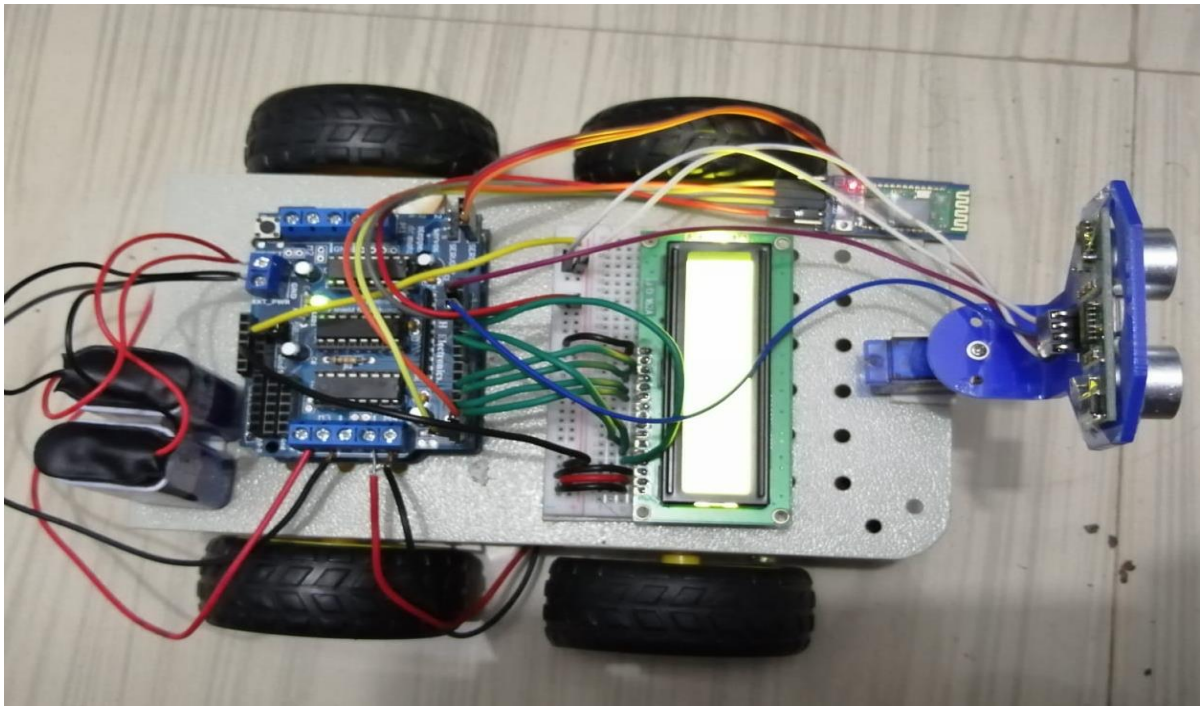
CASE 2

- If there is no obstacle in front of the car then it waits for our commands and moves according to the voice commands which are being received by HC-05 bluetooth module

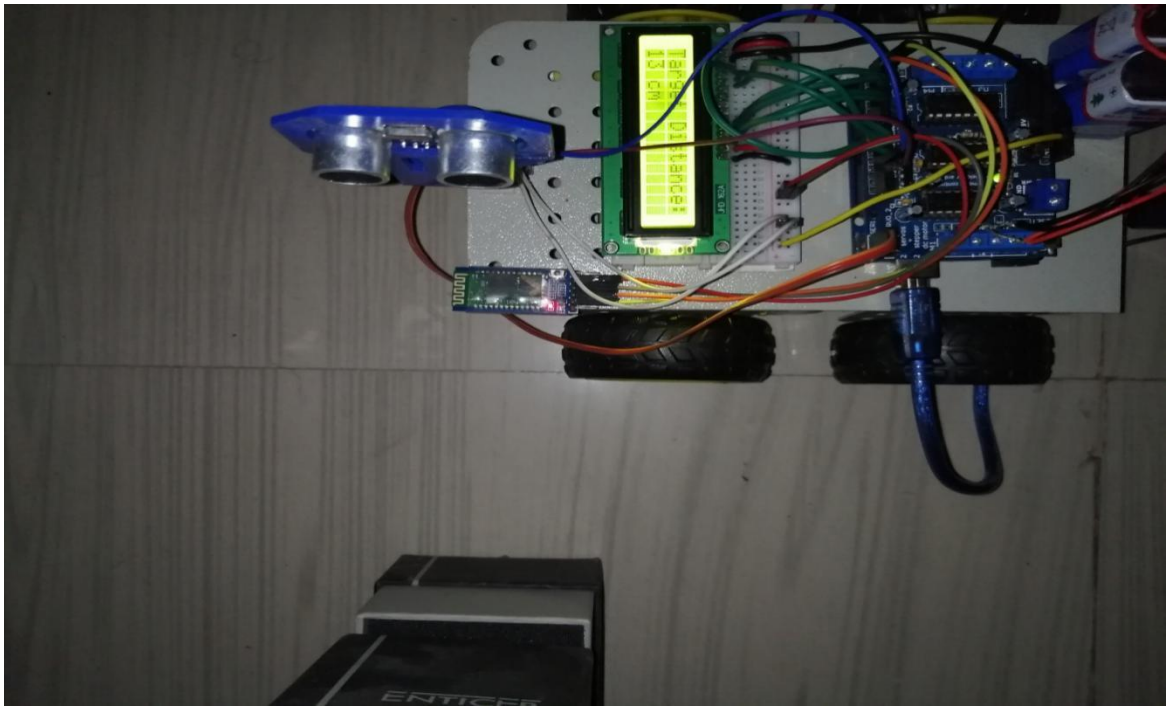
- In this project we are integrating the obstacle detection and voice controlled mechanisms to avoid following problems.

- In obstacle detection car it can only move according to the obstacles present in the path but it cannot move from an source to and destination that's why we added voice controlled mechanism to it.
- Voice controlled car doesn't need obstacle detection mechanism but voice controlled car has a drawback that is it takes more time to process the voice command. Before we find the obstacle and process it the car may collide with obstacle here by using the ultrasonic sensor to find an object can be processed fast and that distance is being calculated and displayed on LCD.

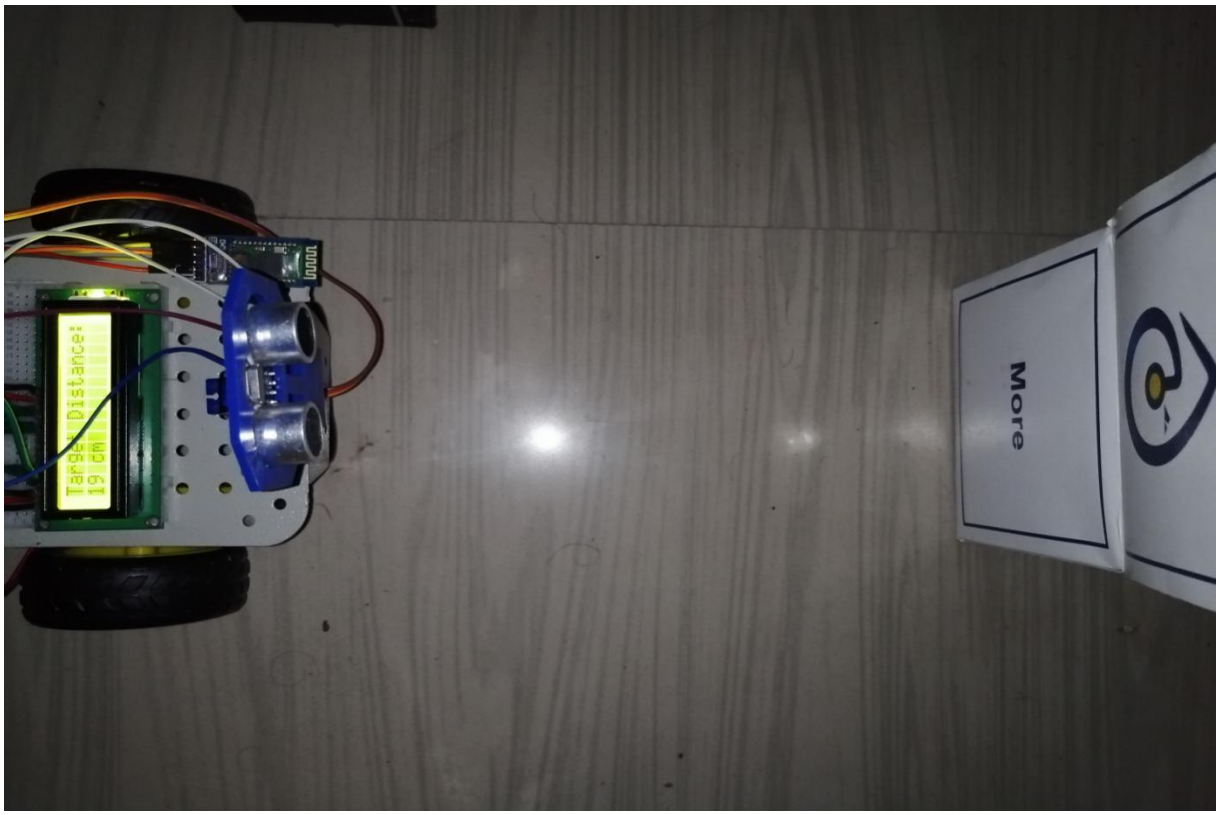
6.3 OUTPUTS:



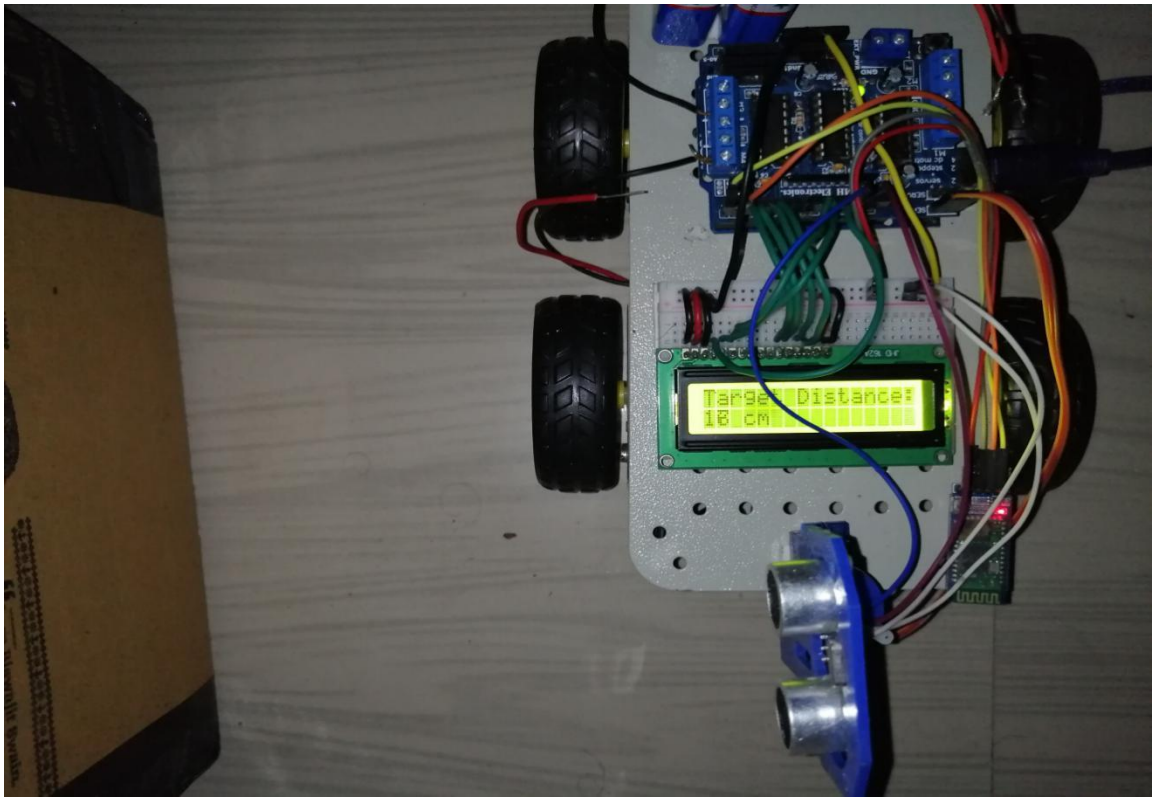
All the connections are being made as per the circuit diagram and the code is being uploaded into the arduino board using arduino USB B cable.



Here the ultrasonic sensor is detecting an object present at 13cm left side which is being displayed o the LCD display so now it checks at the right side to move.



Initially the car has detected an object in front of it at 19 cm which is being displayed on the LCD display and now it checks at both side to move.



At right side the target obstacle is being detected at 18 cm away from the car now it checks at which side the object is at maximum distance and moves towards that direction

CHAPTER-7
CONCLUSION

Conclusion:

The proposed system shows how the car is controlled with Bluetooth module using Android Smart Phone. It also shows that the obstacles in its path, deviates its direction from the obstacles and the distance between the obstacle and the car is displayed using LCD. The proposed system also shows that how the car automatically reaches the destination when user gives the specific commands. We need to add some specifications like fire detection, metal detection to the proposed system so that it could be used in army but due to lack of components we could not add those specifications.

Future Enhancement:

We need to integrate some extra components to increase the applications of the vehicle i.e, by integrating a camera up front of the vehicle we can use it as a complete surveillance vehicle which can monitor and detect the object. Features like metal detectors and fire detectors can be included in this project and used as a vehicle to bring back the soldiers from the battlefield safely. Further this project can be developed as a fully automated humanoid robot which is used to detect the enemies in the war field and shot them.

References:

1. A.K.Shri Vastava, A. varma, S.P.Singh, Distance Measurement of an Object or Obstacle by Ultrasound Sensors, International Journal of Computer Therapy and Engineering, Vol.2, No.1, February 2010.
2. Christofer N. Yalung, Cid Mathew S. Adolfo, Analysis of Obstacle using Ultrasonic Sensor, IRJET, Vol.4, issue.01, Jan 2017.
3. Emmanuel Gbenga Dada, Abdul kadir Hammidu Alkali, Dauda Elijah Mshelia and Abdul-Azeez Adam Zarma, Bluetooth Activated Robotic Car, International Journal of Computer Engineering and Information Technology, Vol.10, No.8, August 2018
4. HAN Duong-bo, ZENG Bing, GE Liang-quan, XIAO Peng and WANG Fu-gang, "Design of voice control intelligent mini vehicle based on SPCE061A, Computer Measurement and Control, vol. 09, September 2007.
5. Hai-Wu Lee, Zi-Chen Fang, Jin-Yu Chen, "A Study of WIFI Control Wheeled Robot System with Ultrasonic Obstacle Avoidance", Consumer Electronics-Taiwan (ICCE-TW) 2018 IEEE International Conference on, pp. 1-2, 2018.
6. Harish Kumar. N, Deepak. G, Nagaraja. J, An IOT based Obstacle Detection and Alerting System in Vehicles using Ultrasonic Sensor, Vol. 05, issue.20, 2017.
7. Mr.Vedant Chikhale, Mr.Raviraj Gharat, Ms.Shamika Gogate, Mr.Roshan Amireddy, Voice Controlled Robotic System using Arduino Microcontroller, International Journal of Technology and Research, Vol.3, issue.4, April 2017.
8. R.Pahuja and N.Kumar , “Android Mobile phone Controlled Bluetooth Robot”, using 8051 Microcontroller proceedings, vol. 2, no. 7.
9. Souvik Paul, Saumedhik Biswas, Atreyo Sengupta, Banhishikha Basu, Sreya Basu, Arduino Bases Bluetooth Controlled RC Car, International Journal of Computer Science and Communication, Vol.11, No.1, September 2019 – March 2020.
10. N. Anju Latha, B. Rama Murthy, K. Bharat Kumar, Distance Sensing with Ultrasonic Sensor and Arduino, IRJAIIT, Vol.2, issue 5, 2016

11. Yasir Ali Memon, Imaaduddin Motan, Muhamad Ali Akbar, Sarmad Hameed, Moez UL Hasan, Speech Recognition System for Voice Controlled with Real Time Obstacle Detection and Avoidance, International Journal of Electrical, Electronics and Communication Engineering, Vol.4 , issue.9, September 2016.