A PROJECT/DISSERTATION ETE REVIEW REPORT
ON

Voice Controlled Car using Arduino and Bluetooth Module

Submitted in partial fulfillment of the requirement

for the award ofthe degree of

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

PROJECT REPORT

Submitted to:

Dr. E. Rajesh

Submitted by

1. PANKAJ KUMAR (19SCSE1010115)

2.SHAHRUKH IFTEKHAR (19SCSE1010148)

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING GALGOTIAS UNIVERSITY, GREATER
NOIDA**

TABLE OF CONTENTS

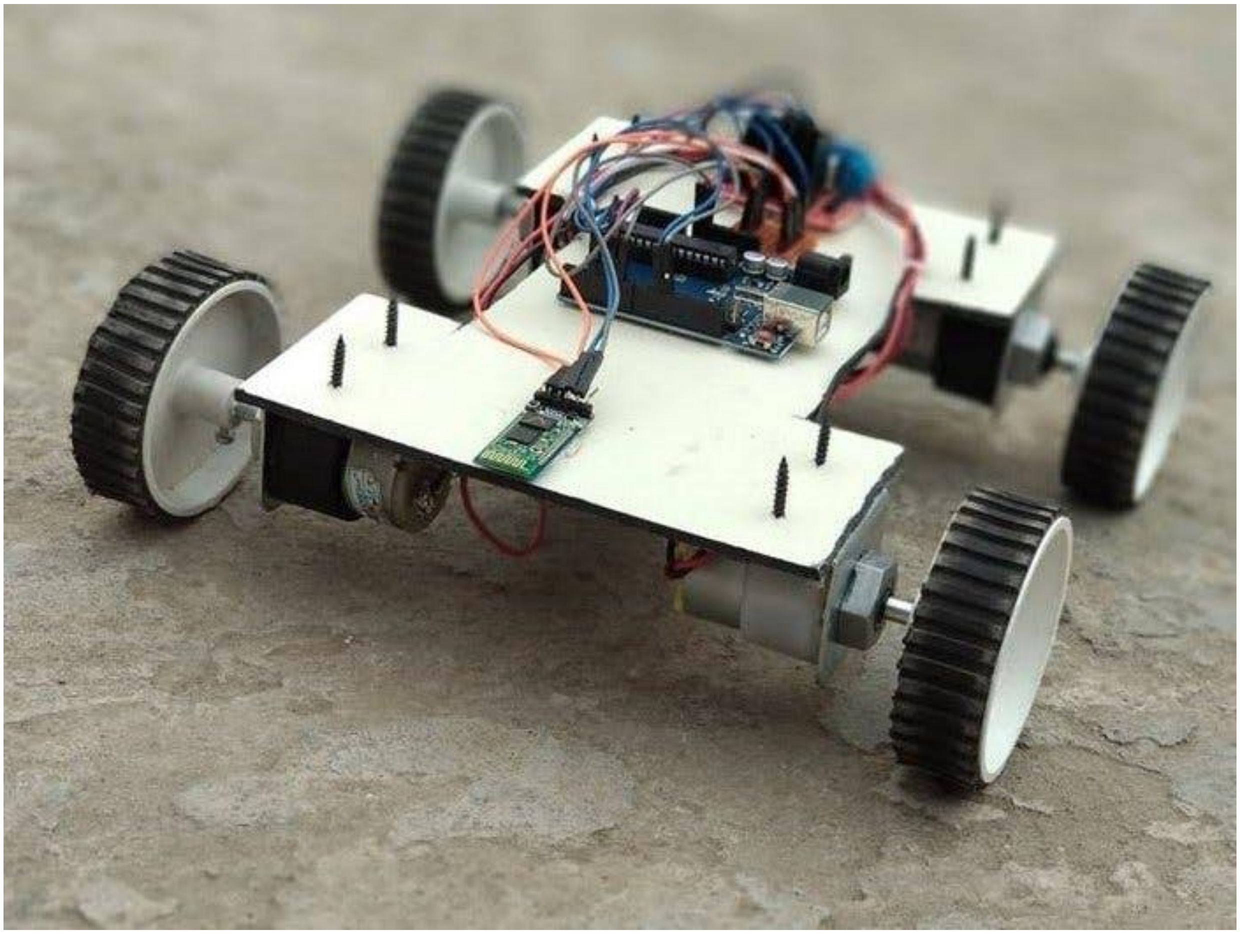
ABSTRACT:

This project builds a voice-controlled car that can be controlled by voice commands which reacts in accordance to the corresponding voice command. However, noise and distance handling require future development. Simple voice commands like left, right, forward, back, stop is used to run the car. These commands are given to Bluetooth module via an android application. The Bluetooth module and control unit are combined to store and test the voice commands. When an instruction for the automobile (car) is identified, a command message is sent to Arduino UNO, the Microcontroller of the car by the Bluetooth device. This command is analyzed by the microcontroller and followed up. In the vehicle, Image processing can be utilized to become aware of the shade and the obstacles. This work has been limited to the ZigBee system in the short-range (100mts range), and is linked to the car over long distance via long-range modules.

Keywords: Arduino UNO, Bluetooth module, Image processing, Zigbee system, speech recognition.

INTRODUCTION :

This project is related to a voice-controlled car, which makes this project speech an important part of identifying commands and delivering them to the car via a wireless system. Identification of speech is also called as "automatic speech recognition (ASR)”. This undertaking has numerous elements like, interaction between laptop and microprocessor, input and understanding of speech, Wireless numerical data distribution microprocessor to microprocessor, the control of several motors using microprocessors.



A. Voice and Speech

Voice is a sound which is produced by living beings. Voice uses airflow that comes from lungs. Air makes pressure over vocal folds which vibrate. Normally speech produces a whisper in our throat by using neck, chest, and abdomen this whisper becomes our speech. Our speech is unique for every person and also it helps other people to understand each other’s personality, mood and most importantly it helps people to communicate. Sounds propagate using mechanical waves for traveling around gases, liquids and solids. Mechanical waves transfer their energy from one medium to another medium while using vibration. Microphone is a hardware device which can convert analog input to a digital output. Digital data can be understood, modify and store by computers. Computers can recognize speech by using some complex algorithms and good dictionaries for these algorithms. Some systems use the Hidden Markov Model (HMM) and the Mel frequency cepstral coefficients (MFCC) techniques as well as the techniques of frequency spectral decomposition to use these two algorithms.

B. Arduino Communication

To communicate with Arduino we first need to install its free software from the internet and install. The software is very easy to use and installing it creates just one .uno files on the microprocessors these files confuse the user because there are many different files generating. After installing Arduino it is ready to usage including dictionaries using dictionaries is very easy on the Arduino and Arduino does not requires any configuration setting when programming. User can use USB cable to connect Arduino and after that user can dump his code to Arduino far more easily and quickly than micro-controller.

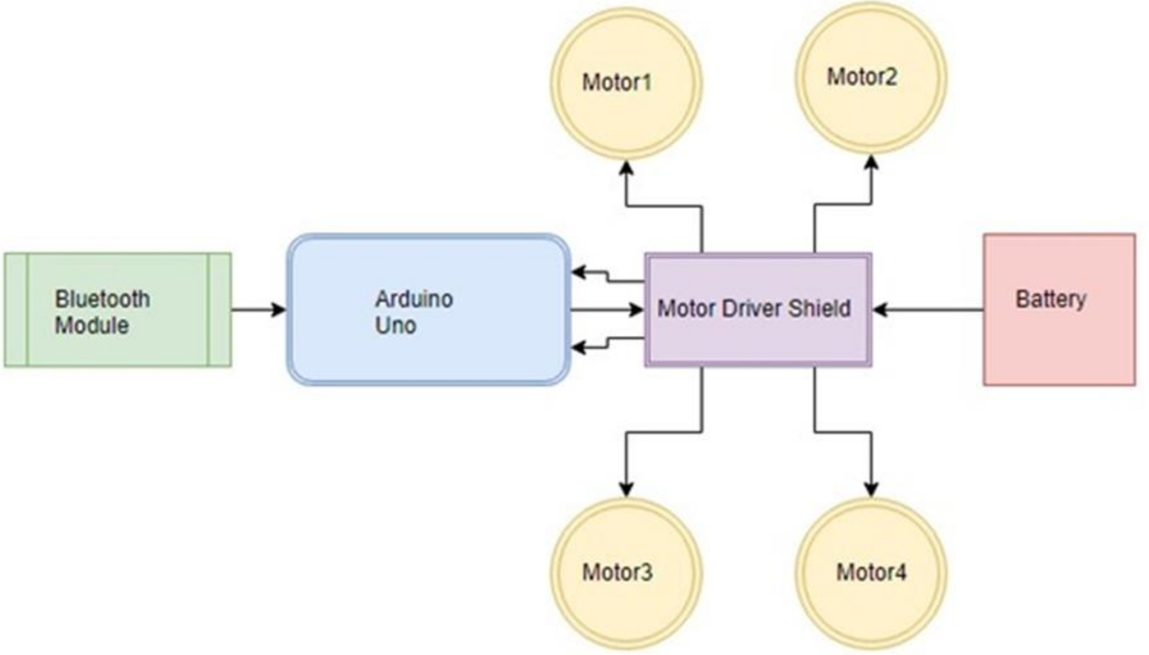
C. Bluetooth Communication

Bluetooth is one of the popular devices to communicate in short range it is used on computers, cell phones, head phones and many other devices. Bluetooth devices use 2.4 to 2.5 GHz frequency to communicate with each other. Bluetooth standardized as IEEE 802.15.1 but then it changed that 802.15.1 Bluetooth's range is 2400–2483.5 MHz approximately. Bluetooth devices generally use frequency-hopping spread spectrum communication technique to communicate with each other.

LITERATURE SURVEY:

Worldwide investment in industrial robots up 19% in 2003. In first half of 2004, orders robots were up another 18% to the highest level ever recorded. Worldwide growth in the period 2004-2007 forecast at an average annual rate of about 7%. Over 600,000 household robots in use – several millions in the next few years. UNECE issues its 2004 World Robotics survey We refer the navigation system from ARDUINO BASED VOICE CONTROLLED ROBOT and author of K.Kannan and Dr. J Selvakumar. They are defining the modes of speaking Robot. There are generally three modes of speaking, including: • Isolated word (or phrase) mode: In which the user speaks individual words (or phrases) drawn from a specified vocabulary. • Connected word mode: In which the user speaks fluent speech consisting entirely of words from a specified vocabulary (e.g., telephone numbers). • Continuous speech mode: In which the user can speak fluently from a large (often unlimited) vocabulary. We refer the Voice Recognition system from SPEECH RECOGNITION FOR ROBOTIC CONTROL and author of Prof. Bhuvaneshwari Jolad and Mohnish Arora. They define how to recognise voice of Robot. • Small vocabulary systems which provide recognition capability for up to 100 words. • Medium vocabulary systems which provide recognition capability for from 100 to 1000 words. • Large vocabulary systems which provide recognition capability for over 1000 words.

DATA FLOW DIAGRAM:



EXISTING SYSTEM:

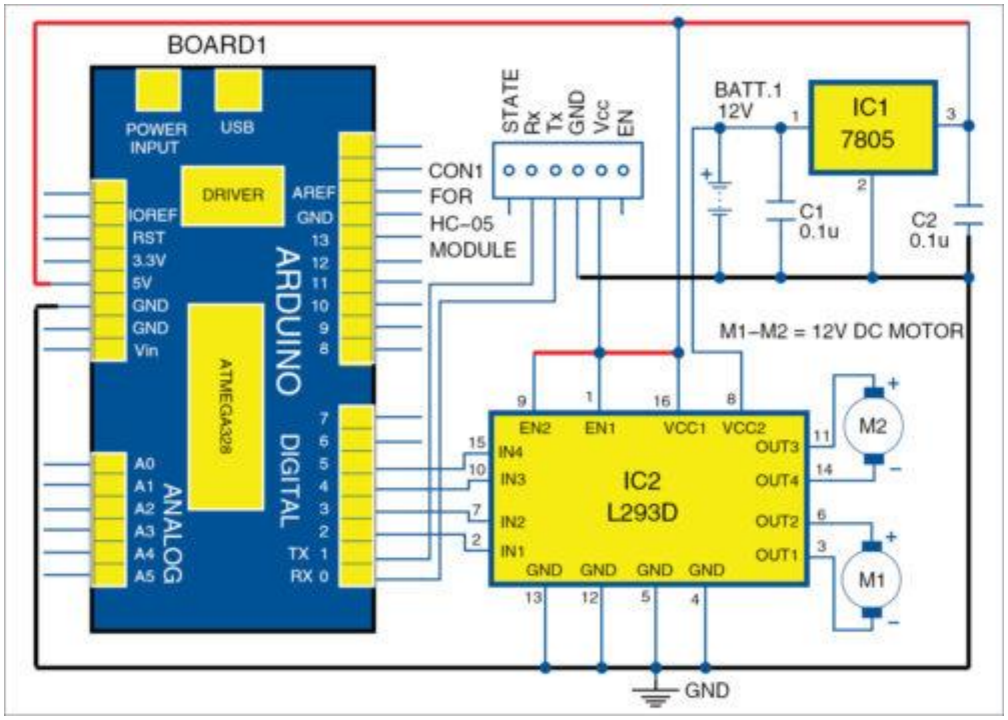
The current systems are robots like line follower robot, edge averting robot, DTMF robot, gesture-controlled robot. These types of robots are not efficient since they require more power to run, cost is also very high. In the existing system they don’t use voice commands, making it not possible for physically handicapped people to drive. The voice commands are interpreted via an offline server in real time. The commands are at once transmitted to the server directly by the means of a wired network. The car is built primarily on a platform based on a microcontroller. Some of the fields that can likewise be equally enhanced are the effect of the mouth-microphone range on the robotic, the overall performance (scope) of the robot and the effect of noise on the translation of speech to textual content. In the existing system BitVoicer Server is used, it's a database for speech processing and automation synthesis. It was designed to make voice operation possible with simple gadgets having low processing power. Microcontrollers usually do not have enough storage and computing ability to perform sophisticated speech treatment and synthesis. By doing the tough work BitVoicer Server removes the consequences of these limitations so that the microcontroller can assign its key functionality to most of its origin sources.

WORKING OF THE PROJECT:

As mentioned above Voice Commands are processed by phone, and speech-to-text conversion is done within the app using Google’s speech-recognition technology. Text is then sent to the receiver side via Bluetooth. Text received via Bluetooth is forwarded to Arduino Uno board using UART serial communication protocol. Arduino code checks the text received. Whenever the text is a matching string, Arduino controls the movements of the robot accordingly in forward, backward, Turning Right, Turning Left & Stop.

Signal logic levels at the different stages of the circuits for proper controlling of the robotic car are given below.

Users Command	Arduino O/P Digital Pins (2,3,4,5)	L293D Input Pins (2, 7, 10, 15)	L293D Output Pins (3, 6, 11, 14)
Forward	H L H L	H L H L	H L H L
Backward	L H L H	L H L H	L H L H
Left	- H L L L for 1s, then H L H L if previous Command was Forward - H L L L for 1s. then L H L H if previous Command was Backward		
Right	- L L H L for 1s, then H L H L if previous Command was Forward - L L H L for 1s, then L H L H if previous Command was Backward		
Stop	L L L L	L L L L	L L L L



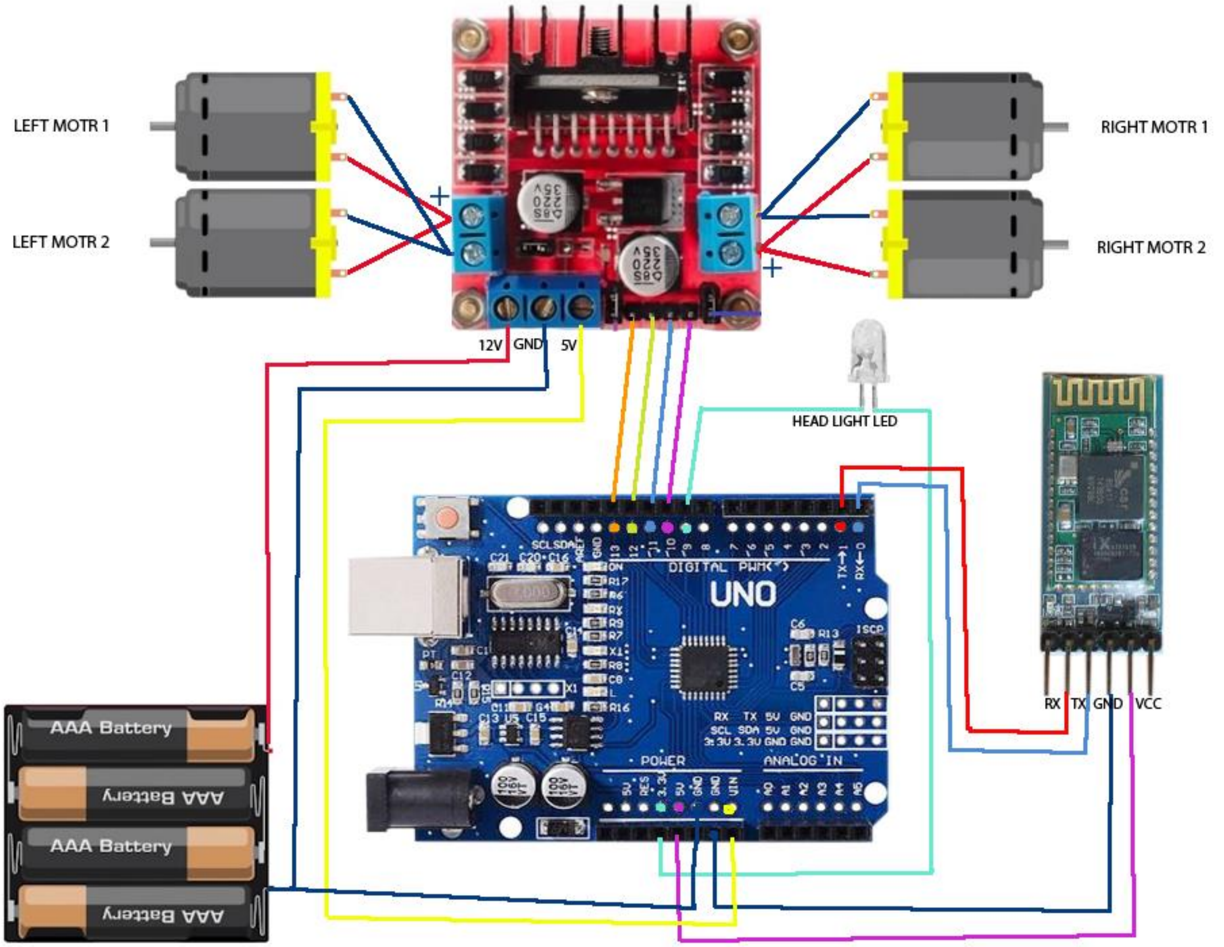
CODE:

```
#include <SoftwareSerial.h>

SoftwareSerial BT(0, 1);
String readvoice;

void setup() {
 BT.begin(9600);
 Serial.begin(9600);
  pinMode(4, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

}
void loop() {
  while (BT.available()){  //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = BT.read(); //Conduct a serial read
  readvoice += c; //build the string- "forward", "reverse", "left" and "right"
  }
  if (readvoice.length() > 0) {
    Serial.println(readvoice);

  if(readvoice == "*forward#")
  {
    digitalWrite(3, HIGH);
    digitalWrite (4, HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    delay(100);
  }

  else if(readvoice == "*back#")
  {
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(6,HIGH);
    delay(100);
  }

  else if (readvoice == "*left#")
 {
   digitalWrite (3,HIGH);
   digitalWrite (4,LOW);
   digitalWrite (5,LOW);
   digitalWrite (6,LOW);
  delay (800);
     digitalWrite(3, HIGH);
   digitalWrite (4, HIGH);
   digitalWrite(5,LOW);
   digitalWrite(6,LOW);
   delay(100);

 }

else if ( readvoice == "*right#")
{
  digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (800);
     digitalWrite(3, HIGH);
   digitalWrite (4, HIGH);
   digitalWrite(5,LOW);
   digitalWrite(6,LOW);
   delay(100);
}

else if (readvoice == "*stop#")
{
  digitalWrite (3, LOW);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (100);
}
else if (readvoice == "*keep watch in all direction#")
{
  digitalWrite (3, HIGH);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (100);
}
 else if (readvoice == "*show me Garba#")
{
 digitalWrite (3, LOW);
   digitalWrite (4, HIGH);
   digitalWrite (5, LOW);
   digitalWrite (6, LOW);
   delay (400);
      digitalWrite(3, HIGH);
    digitalWrite (4, HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    delay(600);
    digitalWrite (3, LOW);
   digitalWrite (4, HIGH);
   digitalWrite (5, HIGH);
   digitalWrite (6, LOW);
   delay (500);
   digitalWrite (3, HIGH);
   digitalWrite (4, LOW);
   digitalWrite (5, LOW);
   digitalWrite (6, HIGH);
   delay (500);

digitalWrite (3, LOW);
   digitalWrite (4, HIGH);
   digitalWrite (5, LOW);
   digitalWrite (6, LOW);
   delay (400);
      digitalWrite(3, HIGH);
    digitalWrite (4, HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    delay(600);
    digitalWrite (3, LOW);
   digitalWrite (4, HIGH);
   digitalWrite (5, HIGH);
   digitalWrite (6, LOW);
   delay (500);
   digitalWrite (3, HIGH);
   digitalWrite (4, LOW);
   digitalWrite (5, LOW);
   digitalWrite (6, HIGH);
   delay (500);digitalWrite (3, LOW);
   digitalWrite (4, HIGH);
   digitalWrite (5, LOW);
   digitalWrite (6, LOW);
   delay (400);
      digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
   digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
   digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
```

```
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
   digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
   digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
```

```
  digitalWrite (6, LOW);
  delay (400);
     digitalWrite(3, HIGH);
   digitalWrite (4, HIGH);
   digitalWrite(5,LOW);
   digitalWrite(6,LOW);
   delay(600);
   digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, HIGH);
  digitalWrite (6, LOW);
  delay (500);
  digitalWrite (3, HIGH);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, HIGH);
  delay (500);
}


readvoice="";}} //Reset the variable
```

ARCHITECTURE :



PROPOSED SYSTEM:

In this proposed device we perform a variety of research on control style variants for robots. It shows that it’s feasible to study to successfully manipulate actual world objects with solely voice (human voice) as a control mechanism. The reason of this lookup is to provide simple robotic hardware architecture so that this shape can focal point on Bluetooth connection infrastructure. It is also beneficial for academic robotics due to the fact human beings can construct their personal robots with low cost. When the app is operating in the system, a microphone on the mobile is used to identify user voice commands. Commands are interpreted and the program utilizes Google's speech-recognition software to translate voice to text within the app. The text will then be sent with the aid of Bluetooth to the receiver part. The microcontroller Arduino UNO R3 has 32kB of ISP flash memory, 2kB of RAM and 1kB of EEPROM. The panel incorporates serial communication connectivity with UART, SPI and I2C. The MCU will operate at 16MHz clock speed. The digital Arduino I / O pins 3, 4, 5 and 6 are programmed as output pins in this design. For serial communication with the Bluetooth unit, pins 0 and 1 of Arduino are used. Text obtained with the aid of Bluetooth is forwarded to Arduino UNO microcontroller panel by the usage of UART serial conversation protocol. Table 3.1 displays the voice commands used to monitor the robots and their functions

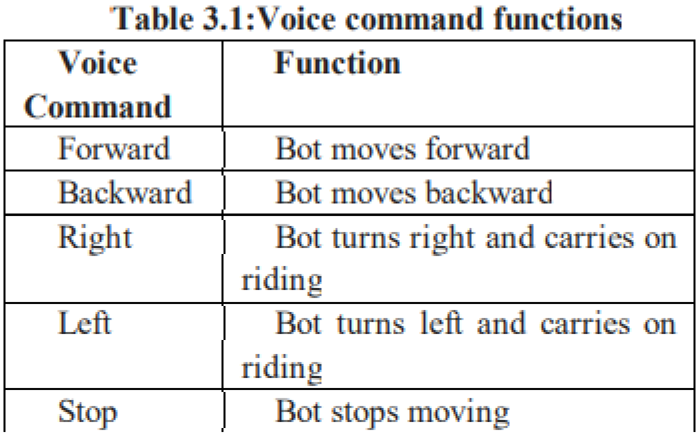


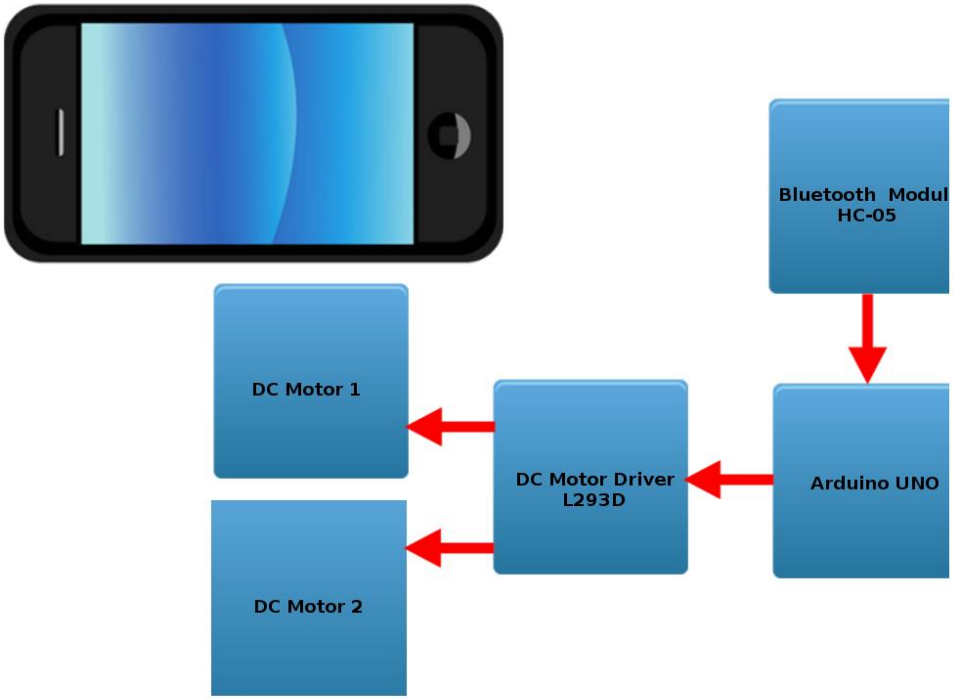
Table 3.1:Voice command functions

Voice Command	Function
Forward	Bot moves forward
Backward	Bot moves backward
Right	Bot turns right and carries on riding
Left	Bot turns left and carries on riding
Stop	Bot stops moving

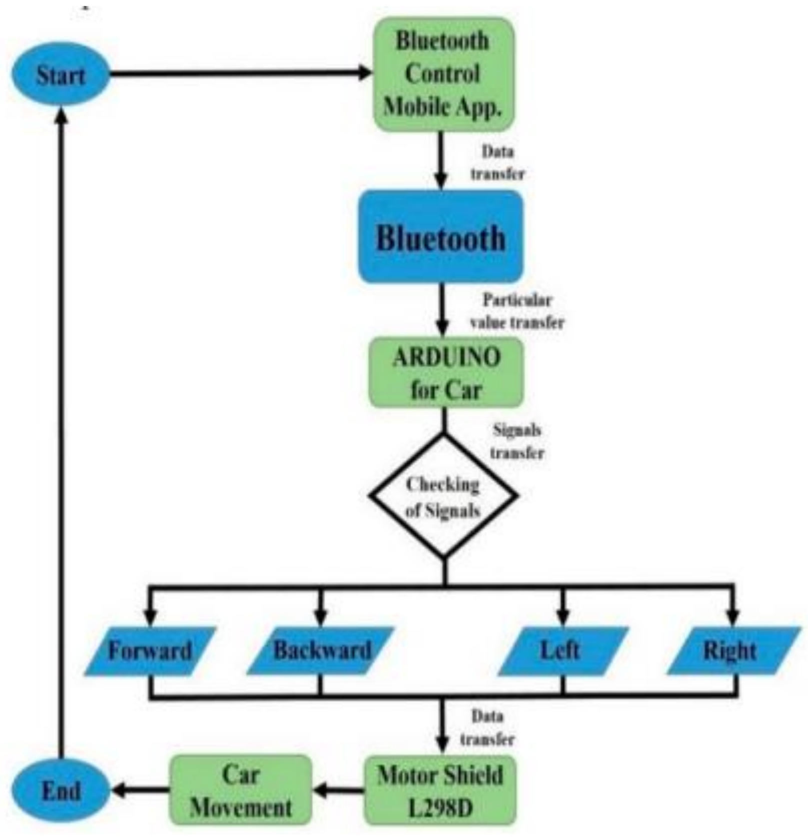
The voice commands to the robotic device are dispatched via Bluetooth with the aid of an Android device. These commands are received on the robotic device by using Bluetooth module set up on it. The motor driver circuit is used to manipulate the velocity of the car. The complete circuitry is powered by the usage of a 12V rechargeable battery hooked up on the system.

Testing

Upon successfully pairing the device, open the app on the smart phone and press on the Bluetooth textual and emblematic pushbutton. The number of associated gadgets will now be shown. Select HC-05 from the listing to join the smart phone with HC-05 Bluetooth module on the receiver side. After successful connection, ‘connected’ will be displayed on the primary screen of Voice control app. Press the pushbutton with microphone image and a prompt will show up asking for voice commands. When it appears, voice instructions are detected via the app, which converts them into textual content and sends it to the receiver aspect wirelessly by using Bluetooth. On the receiving side, Arduino tests the text. If it is a matching string, it controls the moves of the robot in accordance to the description


Bluetooth Modul
HC-05
Arduino UNO
DC Motor Driver
L293D
DC Motor 1
DC Motor 2

Flowchart of Voice controlled car:



COMPONENTS REQUIRED:

¬ *Arduino Uno Board*

¬ *Arduino Cable*

¬ *Arduino Cable*

¬ *Motors –2*

¬ *Motors -2*

¬ *Robot Chassis*

¬ *Caster Wheel*

¬ *Wheels – 2*

¬ *Bluetooth module (HC-05)*

¬ *Connecting wires* ¬ *9V battery and connector*

¬ *Android App - BT Voice Control for Arduino (Google Play store)*

DESCRIPTION:

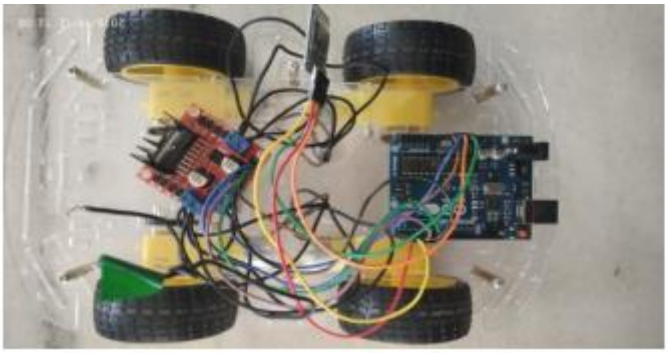
The principle this toy concept was based on was voice recognition technology. The car operates by having a voice receiver box on it where it receives your voice commands. The car has a 50 foot radius from where it is initially place on the ground. The child interacts with the car by shouting out commands like "faster", "slower", "right", "left", and "stop". The child will be able to feel like they are in full control of the car by doing this, thus giving them the sense of being important. The child programs their own voice into the receiver so that only they can operate the car. The car has an emergency shut down when it reaches the boundary of the 50 foot radius. The child must also be within that 50 foot radius when shouting commands in order for the car to initiate the desired command.

The car is relatively cheap to produce because it's exterior is made of plastic. The voice recognition technology also is very cheap these days, and since the receiver only has to have a few words programmed into it, the cost is also reduced. As long as the car is not dropped from a 3 story window, the child would have no problem with its durability. It can withstand crashes against curbs and other such similar obstacles. There is no choking hazard risk in regards with this toy. The on/off switch is located in a safe area where the child's hands won't be near any moving parts. Also, there won't be any sharp edges on the car that can inadvertently cut the child.

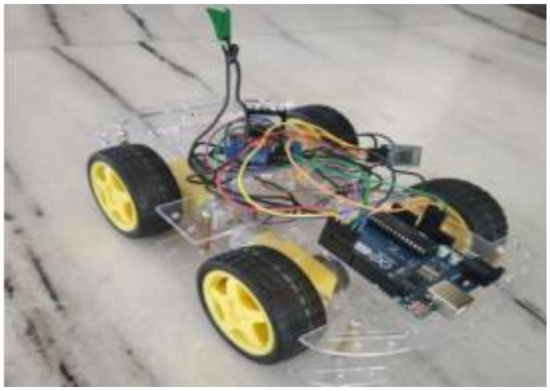
IV. RESULTS

- Voice controlled car is working and all the functions are followed by the bot.
- Unlike DTMF robot, the car is connected to the mobile phone wirelessly making it comfortable to the user to control the car.
- The line follower robot moves only in a particular path, in case there is an obstacle in its path it won't move unless the obstacle is removed.
- Various languages can be given as a voice input using Google's speech recognition technology rather than using an offline BitVoicer server. The result will be shown in following figure.

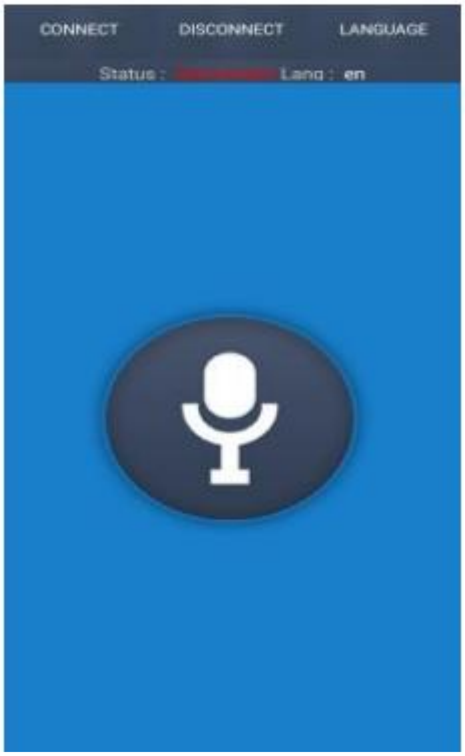
The result will be shown in following figures:-



Voice controlled car (Top view)

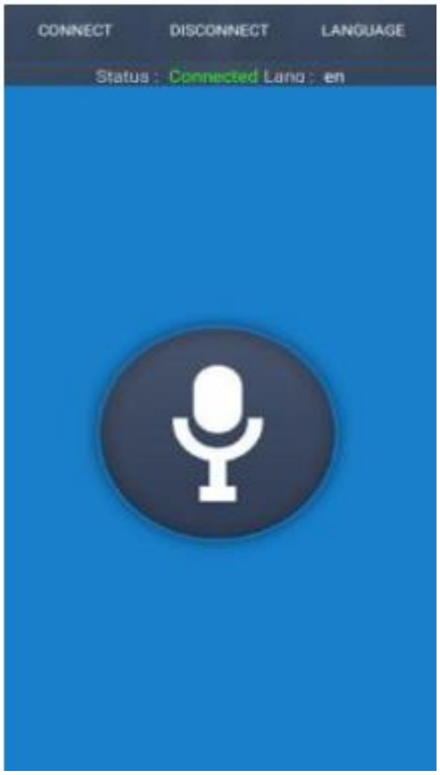


The following figures shows the Bluetooth connectivity to the Bluetooth module



Status : [illegible] Lang : en

(Status Disconnected)



Status : Connected Lang : en

(Status Connected

<u>CONCLUSION:</u>

In this project the voice control was designed for a hoe assistant robot. The order of speech signals is automatically transmitted via a wired network to the server. The car is built primarily on a platform based on a microcontroller. Evaluation of the output of the original tests is carried out with promising implications. Possible developments to feasible technologies in households, schools, vehicle networks and businesses are also addressed. Several areas that may additionally be discussed are the impact of noise on speech to textual content translation. The accent of the speaker no longer affects the robotic activity because speech commands are interpreted using a cloud-based server that works independently of the speaker accent. The use of renewable energy sources for robotic operation would not only increase the value of robotic energy, but would also be environmentally friendly. Solar cells can be a suitable power source to use. The design of the robotic assistant is ideal for applications ranging from chemical manufacturing to comfortable home circumstances. Accuracy of detecting a voice command correctly is found to be 75%.

REFERENCES:

Android smartphone with app. Android speech-recognition app (voicecontrol.apk) used here was developed using MIT App Inventor. When the app is running in the smartphone, user's voice commands are detected by the microphone present in the phone.

Commands are processed, and speech-to-text conversion is done within the app using Google's speech-recognition technology. Text is then sent to the receiver side (that is, robotic car) via Bluetooth.

Arduino Uno R3. Arduino Uno is an AVR ATmega328P microcontroller (MCU)-based development board with six analogue input pins and 14 digital I/O pins. The MCU has 32kB ISP flash memory, 2kB RAM and 1kB EEPROM. The board provides the capability of serial communication via UART, SPI and I2C. The MCU can operate at a clock frequency of 16MHz. In this project, digital I/O pins 2, 3, 4 and 5 of Arduino are configured as output pins. Pins 0 and 1 of Arduino are used for serial communication with HC-05 Bluetooth module.

Text received via Bluetooth is forwarded to Arduino Uno board using UART serial communication protocol. Arduino program voice_ctrl.ino checks the text received and, if it is a matching string, Arduino controls the movements of the robot accordingly. Voice commands used for controlling the robot and their functions