ECE2010- CONTROL SYSTEMS
PROJECT REPORT

OBSTACLE AVOIDANCE, BLUETOOTH CONTROL AND VOICE CONTROL ROBOT USING ARDUINO UNO

Guided by

Dr. Rajesh R

Prepared by

Kapil Sood -20BEC0291

Sathya P -20BEC0642

Kaviya K - 20BEC0643

Vellore Institute of Technology

Vellore

April 2022

Declaration

We would like to express our gratitude to Professor Dr. Rajesh R, who gave us this golden opportunity to do this wonderful project on the topic of obstacle avoidance, bluetooth control and voice control using arduino uno, which also helped us to do a lot of research and we came to know about so many new things.

We are really thankful to you.

List of contents

Contents

Abstract

This project proposes a design variant of obstacle avoidance , bluetooth control and voice control for the robot unit. Further advancement of this technology will help differently-abled people who cannot operate a vehicle by vice of ailments, transportation purposes, hazardous environment places where human interaction might be impossible and so on. Also, the use of sensors would prevent collisions. A Bluetooth module is used to create a communication link between the car and human voice commands via Android Application. The RF transmitter of the module can take human voice commands through the application which will then be converted into encoded digital data up to an adequate range of 12 meters from the robot. The receiver of the module decodes the input data before feeding it to the microcontroller to drive DC motors via motor driver L293D for necessary movements. An Arduino UNO which is the brain of our system is programmed to read voice commands and respond accordingly. Obstacle detection is done by ultrasonic sensor interfaced with the microcontroller. Considering this feature, in the future it might prove a milestone in vehicle automation. Further the project can be developed using Internet of Things, Artificial Intelligence technology where a user can control the car from any corner of the world.

Objectives

The combination of major fields such as mechanical, electrical and electronics provides automation systems which are known as Robots. The growth in these industries are a major reason for the efficiencies in every sector by reducing human effort and interaction. By doing such, this promises us a safer environment on dangerous and insightful grounds. Due to its precision and absolute accuracy it has made a major presence in all the essential fields whether it is education, bio-medicals, engineering and so on.To achieve the obstacle avoidance in the system we are using an Ultrasonic sensor which is connected to the L293D motor driver followed by Arduino UNO.

Introduction

Ultrasonic sensor is used to detect any object at some certain distance using a sensor. This non -contact ultrasonic sensor is used to measure the distance between the object and the sensor. It consists of two transmitters , a control circuit and a receiver for emitting and receiving pulse data respectively. A high ultrasonic sound is emitted by transmitters which will get reflected by any nearby object and the sensor will see toward to get any return echo. The distance for object detection can be subsequently changed by our own means in the coding algorithm. The distance will be calculated from the transmitted signal and receive echo in the control system. The ability of the machine to receive and interpret the human voice or to understand and carry out spoken commands can be concluded as speech recognition. It works on the basis of algorithms codes that match the sound of the detected speech or voice with word sequences and interpret it as a command in ArduinoIDE which is the coding platform for Arduino. With the help of these we can command around our system as per the desired needs. Bluetooth controlled part by using Android mobile phone instead of any other method like buttons, gestures etc. Here we only need to touch a button on an android phone to control the car in forward, backward, left and right directions. So here an android phone is used as a transmitting device and a Bluetooth module placed in the robot is used as a receiver. The Android phone will transmit commands using its in-built Bluetooth to the robot so that it can move in the required direction like moving forward, reverse, turning left, turning right and stop.

Literature Survey:

The essential focus of this research is speech recognition technology by converting speech into text message. Controlling hardware utilizing speech was impractical before. This examination will help us in actualizing this innovation for the debilitated ones who can't drive the vehicle all alone. The utilization of sensors will give more noteworthy wellbeing from abrupt hits because of the auto stopping mechanism and hinder include. A Bluetooth module (HC-05) is utilized to set up a correspondence connection between the vehicle and human voice orders using the Android Application.The RF transmitter of the Bluetooth can take human voice orders which are changed over to encoded advanced information for the benefit of a satisfactory range (up to 100 meters) from the car. The recipient unravels the information before taking care of it to the microcontroller to drive DC engines through engine driver L293D for vital work. An Arduino UNO is modified to peruse voice orders and react appropriately. Ultrasonic sensors interfaced with the Arduino can help in snag identification. Considering this component for the future degree may demonstrate achievement in vehicle robotization.

System Model

Components Required for this project model

- ➔ Arduino Uno board
- ➔ L293D motor driver
- ➔ Ultrasonic sensor
- ➔ Bluetooth module
- ➔ Servo motor
- ➔ Gear motor
- ➔ Robot wheels
- ➔ Li-ion battery holder
- ➔ Li-ion battery

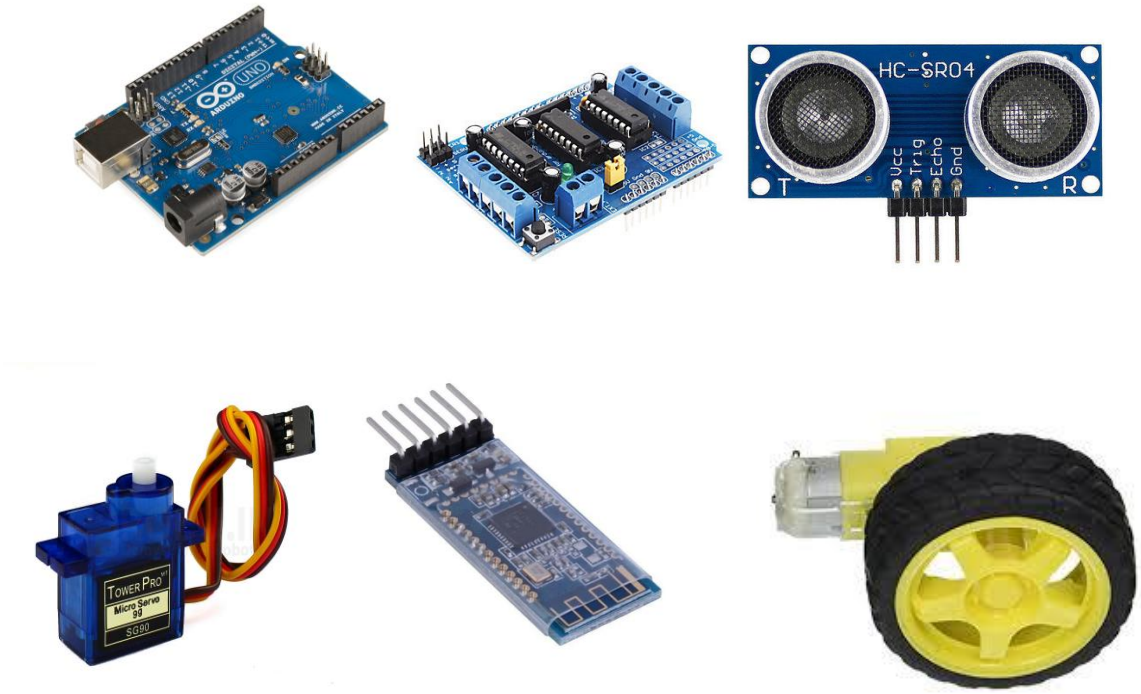


Figure 1(a): Project components

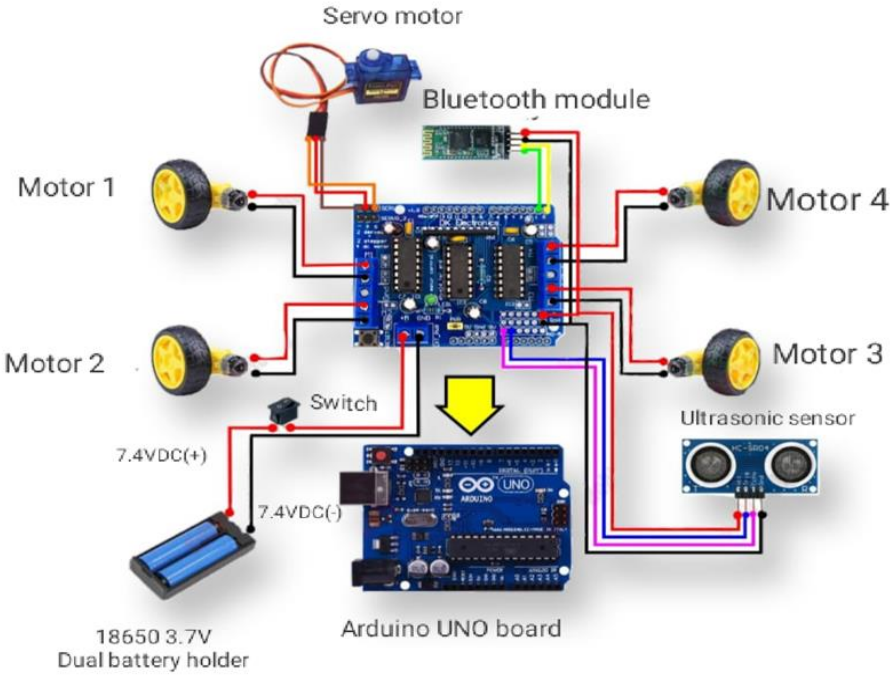


Figure 1(b): Circuit diagram

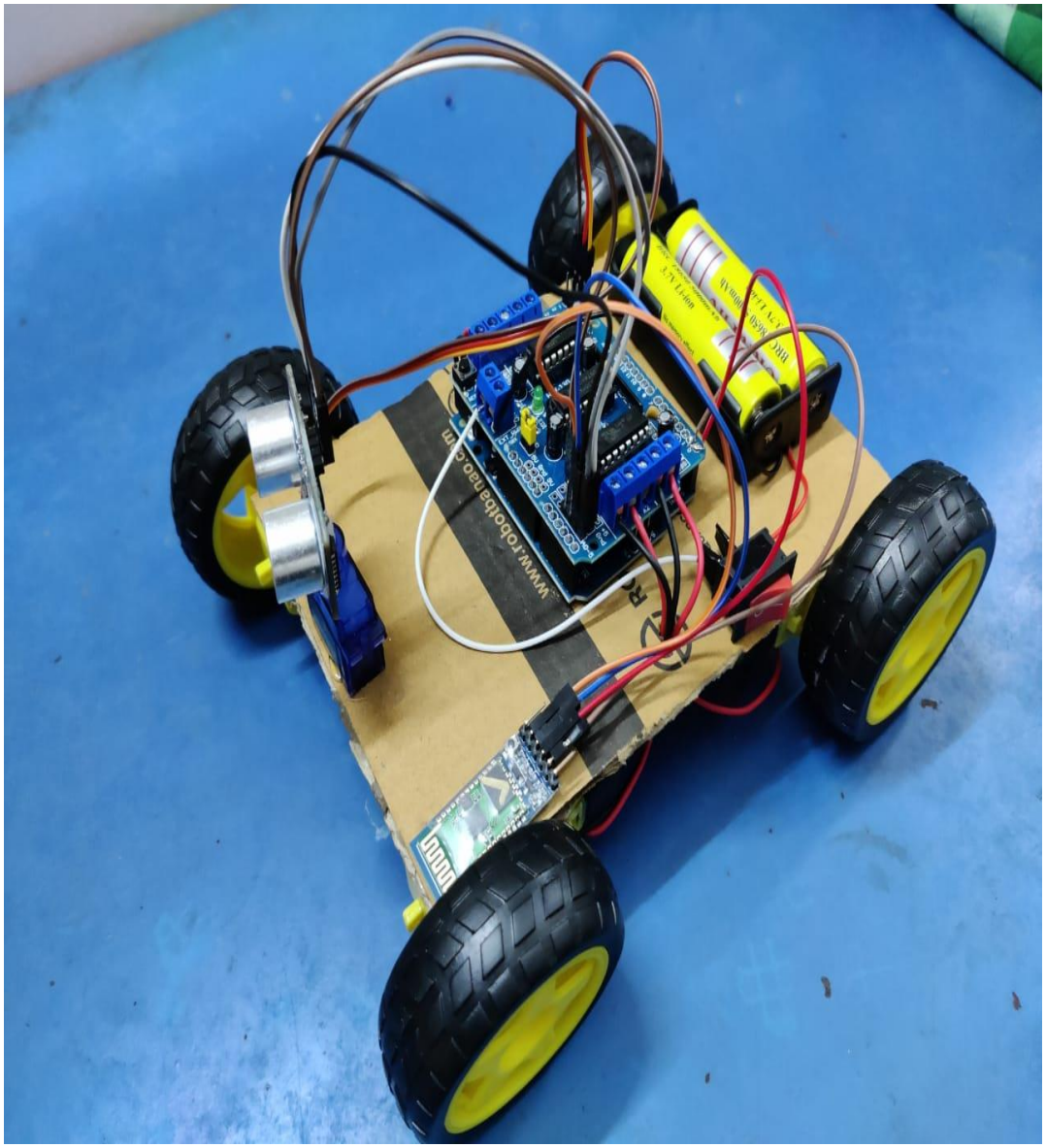


Figure 2(a) : Project model

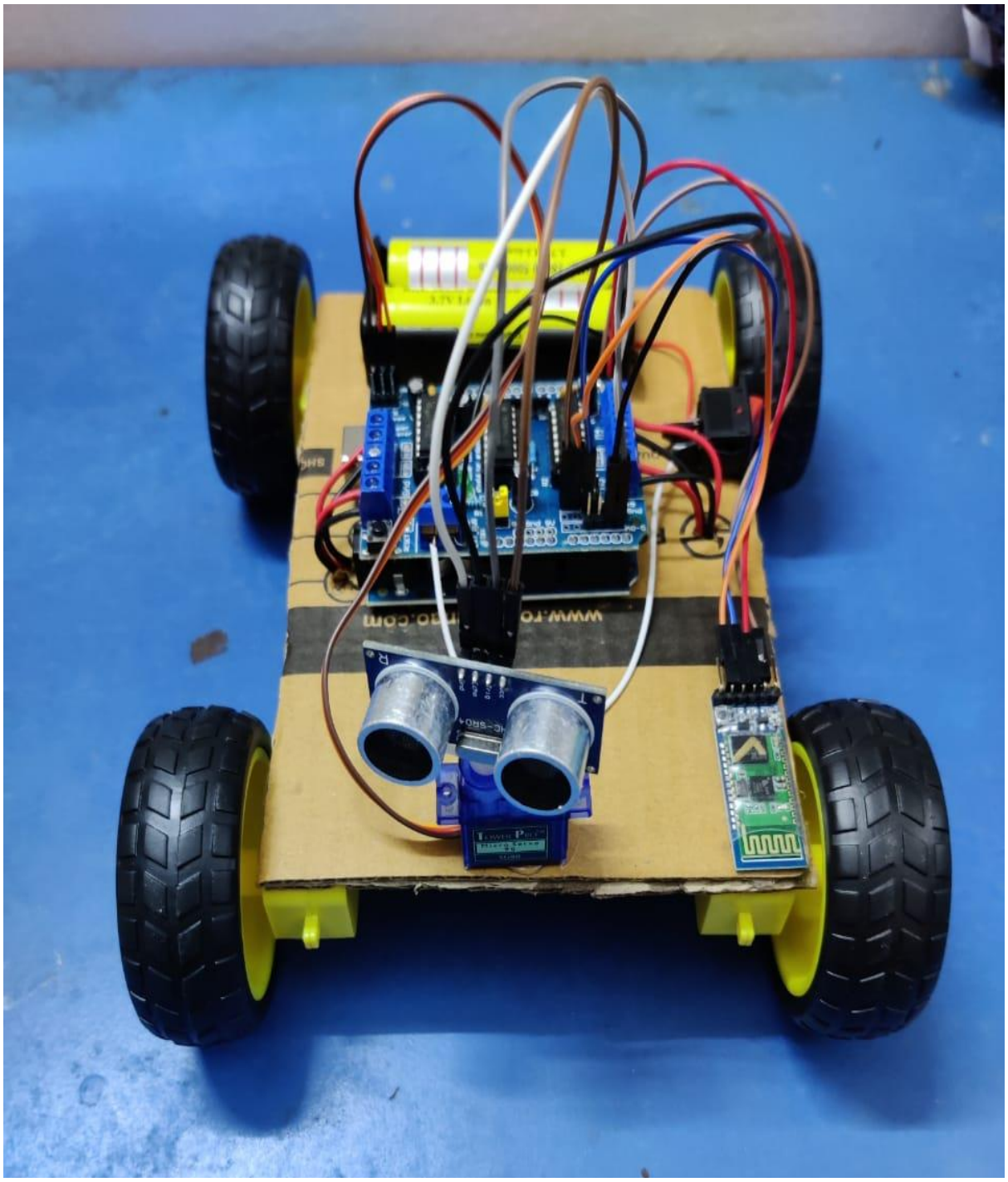


Figure 2(b) : Project model

Methodology

To build this control system we are assembling a 4 wheel chassis kit. The Arduino Uno and L293D Motor Driver are embedded on top of our Chassis, and the L293D motor driver placed top of the arduino uno, both of which are powered by two 3.7 volts Li-ion batteries. Controlling hardware utilizing speech was impractical before. This examination will help us in actualizing this innovation for the debilitated ones who can't drive the vehicle all alone. The utilization of sensors will give more noteworthy wellbeing from abrupt hits because of the auto stopping mechanism and hinder include. A Bluetooth module (HC-05) is utilized to set up a correspondence connection between the vehicle and human voice orders using the Android Application. The RF transmitter of the Bluetooth can take human voice orders which are changed over to encoded advanced information for the benefit of a satisfactory range (up to 12 meters) from the car. The recipient unravels the information before taking care of it to the microcontroller to drive DC engines through engine driver L293D for vital work. An Arduino UNO is coded to peruse voice orders and react appropriately. Ultrasonic sensor interfaced with the Arduino can help in snag identification. Considering this component for the future degree may demonstrate achievement in vehicle robotization. The Android application allows it to connect to the HC-05 module via Bluetooth and sends signals to Arduino so that it can perform the tasks it wants successfully. The application sends 5 signals, numbers from 1 to 5 used to adjust the sandwich agreement accordingly.When the robot car gets power from 7.4v lithium-ion batteries supply, the Arduino Uno gets booted up, and the robot car starts , after this driver gives command to the car via Arduino Bluetooth App. The commands of the driver are, “Move Forward”, “Move backwards ”, “turn left” and “turn right.” After that, the car senses the obstacle in front/back via sensors (ultrasonic sensor) it stops . Then the car waits for the next given by the driver depending upon the command Car will move accordingly.

Performance Analysis:

When the car gets power from a 7.4v lithium-ion battery supply, the Arduino Uno gets powered, and the car becomes functional. Wheels attached to the four motors make the car move and turn. Mechanically, for the car to move forward or backward the four motors start to produce torque for the wheels in the same forward or backward direction. Now for it to turn right, the two motors on the right hand side of chassis start rotating backwards while the other two motors on the left hand side start moving forward. This makes the car turn rightwards on the spot. Similarly is the case with left turning where the motors move just vice versa.

Bluetooth control part

First we initialize the four DC motors making use of <AFMotor.h> library followed by setting the speed of the four motors (in built function of the library). The bluetooth module and the phone communicate via serial communication at a bode rate of 9600. When the bluetooth module receives a signal 'F' from the mobile, the arduino makes the car go FORWARD by moving all the dc motors in forward direction. Similarly is the case with 'B', 'L', 'R', 'S'
We remove the Rx and Tx pins from the bluetooth module and we upload the code. Then we reconnect the Rx and TX pins to the bluetooth module, if not done the code won't get uploaded and an error will be thrown. Now we connect the module to our mobile phones, it will show up as 'HC-05'. Once connected, we can make operate the car by means of our mobile phone by pressing the left, right, up and down arrow in the app.

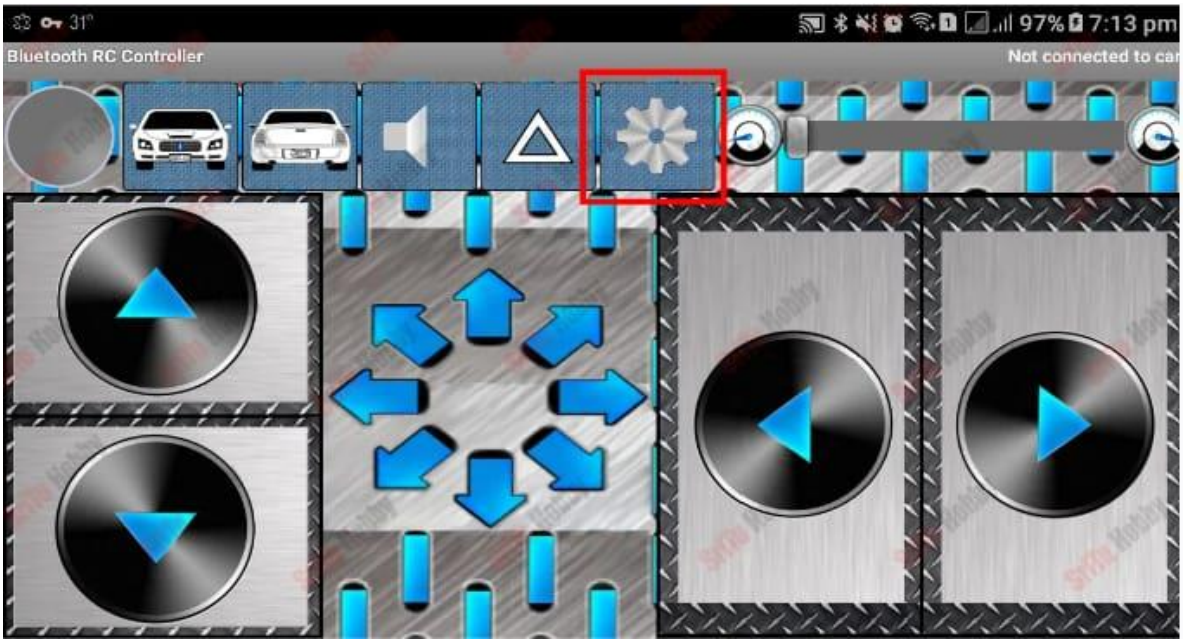
Code for bluetooth control part

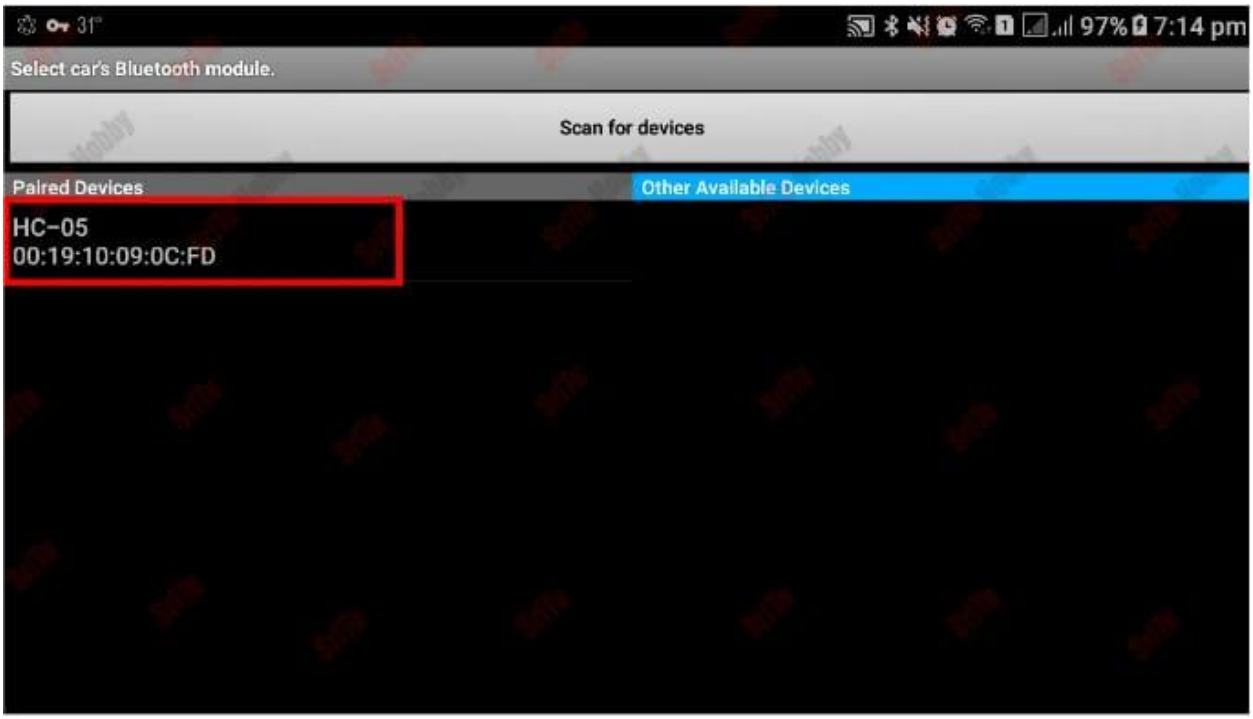
```
#include <AFMotor.h>

#define Speed 170

char value;

AF_DCMotor M1(1);
AF_DCMotor M2(2);
```

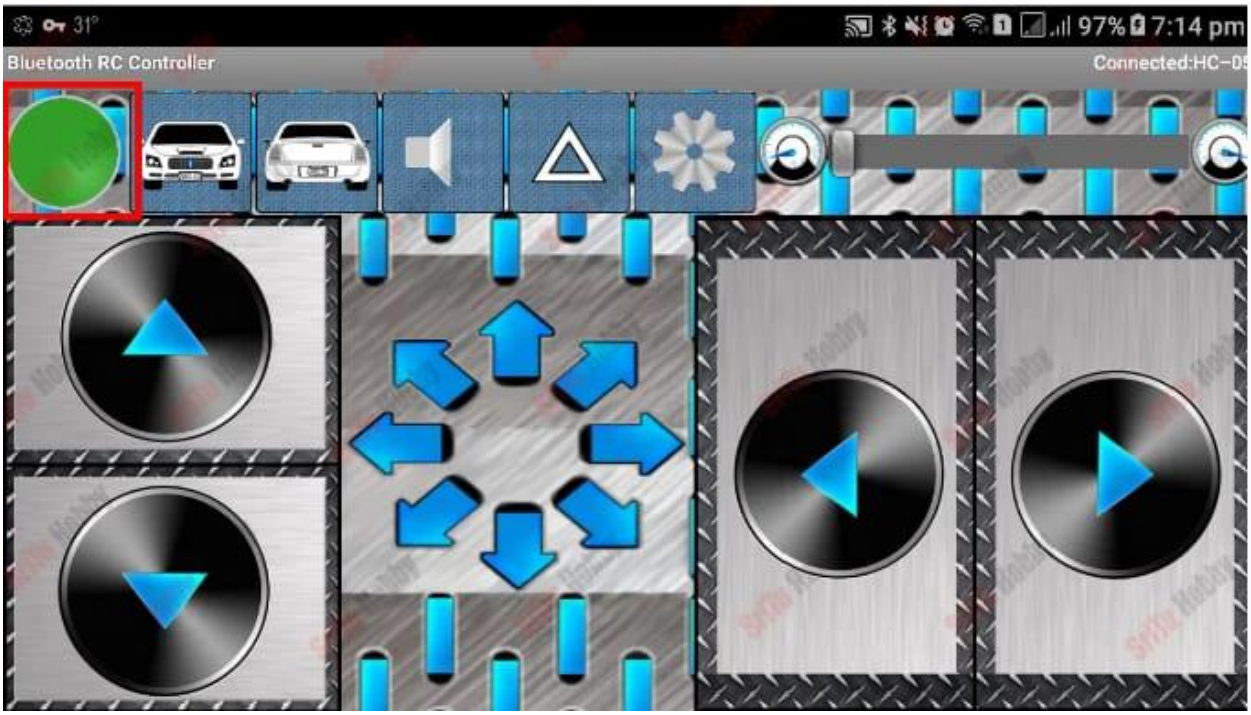
```
AF_DCMotor M3(3);
AF_DCMotor M4(4);
void setup() {
  Serial.begin(9600);
  M1.setSpeed(Speed);
  M2.setSpeed(Speed);
  M3.setSpeed(Speed);
  M4.setSpeed(Speed);
}
void loop() {
  Bluetoothcontrol();
}
void Bluetoothcontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
  }
  if (value == 'F') {
    forward();
  } else if (value == 'B') {
    backward();
  } else if (value == 'L') {
    left();
  } else if (value == 'R') {
    right();
  } else if (value == 'S') {
    Stop();
  }
}

void forward() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
```

```
}
void backward() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void right() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
}
void left() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void Stop() {
  M1.run(RELEASE);
  M2.run(RELEASE);
  M3.run(RELEASE);
  M4.run(RELEASE);
}
```

Figure 3 : Bluetooth RC controller android app


31°
97% 7:13 pm
Bluetooth RC Controller
Not connected to car


31°
97% 7:14 pm
Select car's Bluetooth module.
Scan for devices
Paired Devices
Other Available Devices
HC-05
00:19:10:09:0C:FD



Obstacle avoidance part

For this part, an ultrasonic sensor is mounted atop of servo motor to make it mobile. When an object is placed in front of the car (<12cm), the sensor detects it and stops the car. The servo motor arm then rotates the sensor first rightward then leftwards, in both the cases the distance of nearest objects in both directions is noted. The car moves in the direction in which this distance is lesser.
We write the code for an ultrasonic sensor. The ultrasonic sensor triggers an ultrasonic wave which gets reflected from the surface of the object placed in front of it. This reflected wave is then detected back from the echo pin of the sensor. The sensor perceives this object in terms of time taken for the wave to come back. We write a code converting this time to distance by a simple formula involving time and speed of sound.

Code for Obstacle avoidance part

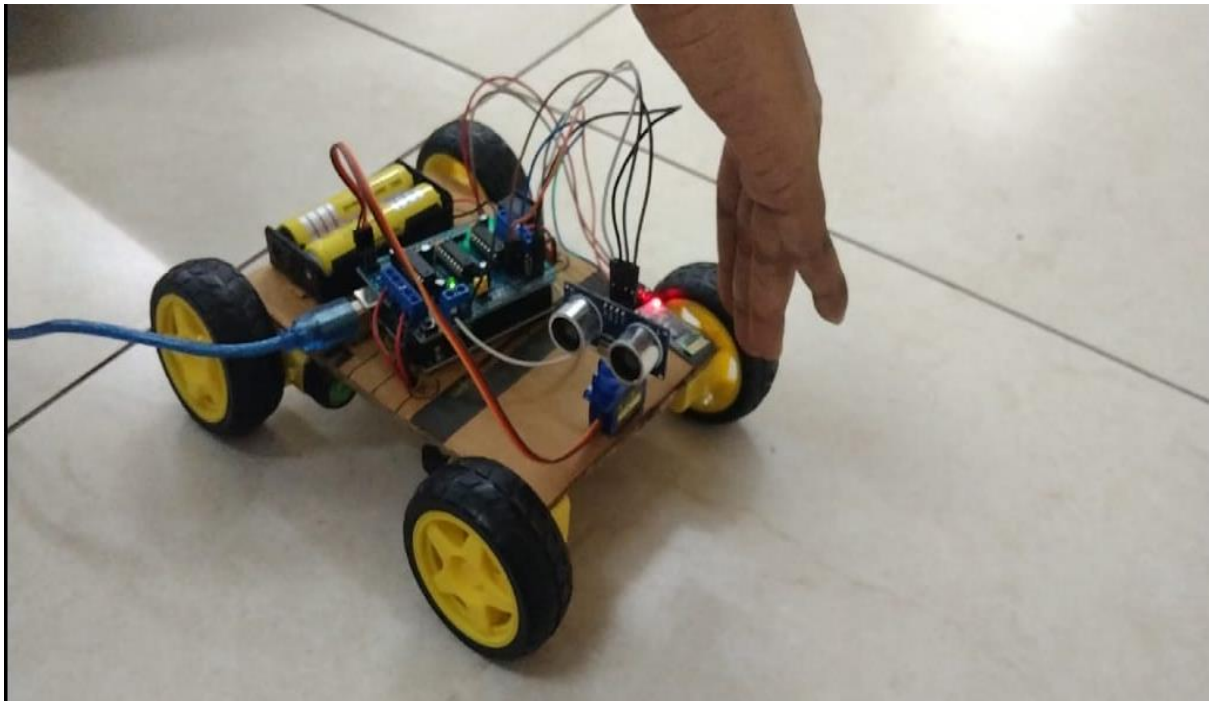
```
#include <Servo.h>
#include <AFMotor.h>
#define Echo A4
#define Trig A5
#define motor 10
#define Speed 170
#define spoint 103
char value;
int distance;
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;
Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);
void setup() {
  Serial.begin(9600);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  servo.attach(motor);
  M1.setSpeed(Speed);
  M2.setSpeed(Speed);
  M3.setSpeed(Speed);
  M4.setSpeed(Speed);
}
void loop() {
```

```
  Obstacle();
}
void Obstacle() {
  distance = ultrasonic();
  if (distance <= 12) {
    Stop();
    backward();
    delay(100);
    Stop();
    L = leftsee();
    servo.write(spoint);
    delay(800);
    R = rightsee();
    servo.write(spoint);
    if (L < R) {
      right();
      delay(500);
      Stop();
      delay(200);
    } else if (L > R) {
      left();
      delay(500);
      Stop();
      delay(200);
    }
  } else {
    forward();
  }
}
// Ultrasonic sensor distance reading function
int ultrasonic() {
  digitalWrite(Trig, LOW);
  delayMicroseconds(4);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
```

```
  digitalWrite(Trig, LOW);
  long t = pulseIn(Echo, HIGH);
  long cm = t / 29 / 2; //time convert distance
  return cm;
}
void forward() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
}
void backward() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void right() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
}
void left() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void Stop() {
  M1.run(RELEASE);
  M2.run(RELEASE);
  M3.run(RELEASE);
  M4.run(RELEASE);
}
```

```
int rightsee() {
  servo.write(20);
  delay(800);
  Left = ultrasonic();
  return Left;
}
int leftsee() {
  servo.write(180);
  delay(800);
  Right = ultrasonic();
  return Right;}
```

Figure 4: Obstacle avoidance



Voice control part

We make use of an app which uses google voice to speech recognition. The app interprets our speech words like ‘Go’, ‘Stop’, ‘Right’, ‘Left’ and ‘Back’ as simple special Chars like ^,-,<,>,* and sends it to the bluetooth module by serial communication. The module sends this to the microcontroller which we have

coded to interpret these symbols as functions of forward();, backward(); etc. So we can control the car only by our voice and a simple mobile app.

Code for Voice control part

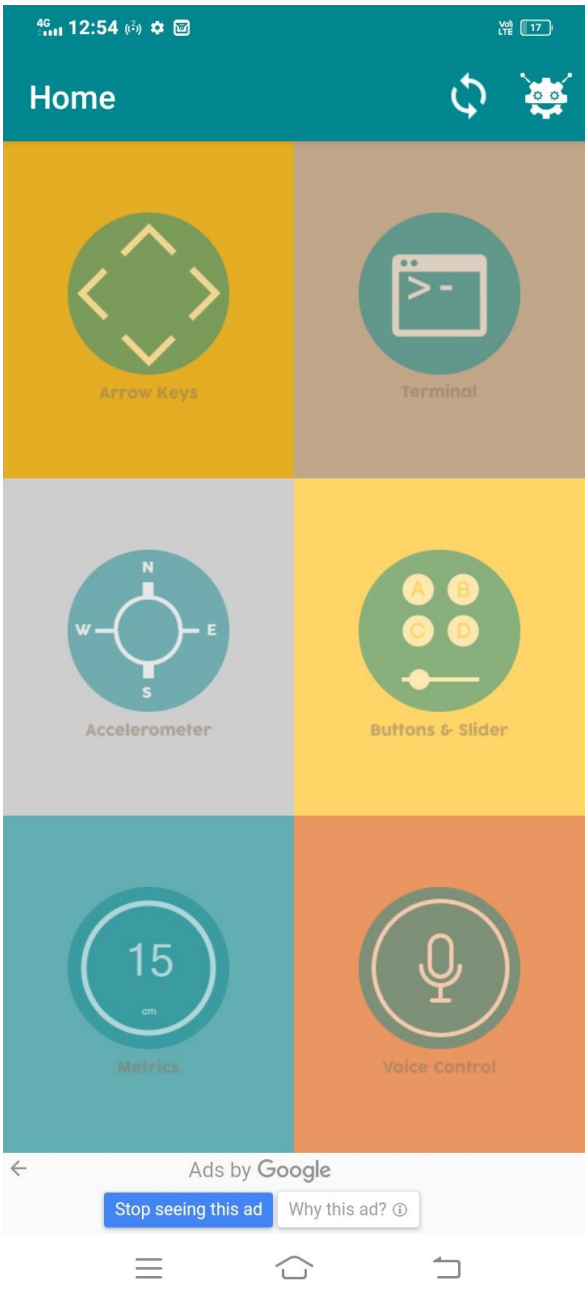
```
#include <Servo.h>
#include <AFMotor.h>
#define Echo A4
#define Trig A5
#define motor 10
#define Speed 170
#define spoint 103
char value;
int distance;
int Left;
int Right;
int L = 0;
int R = 0;
Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);
void setup() {
  Serial.begin(9600);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  servo.attach(motor);
  M1.setSpeed(Speed);
  M2.setSpeed(Speed);
  M3.setSpeed(Speed);
  M4.setSpeed(Speed);
}
void loop() {
  voicecontrol();
}
```

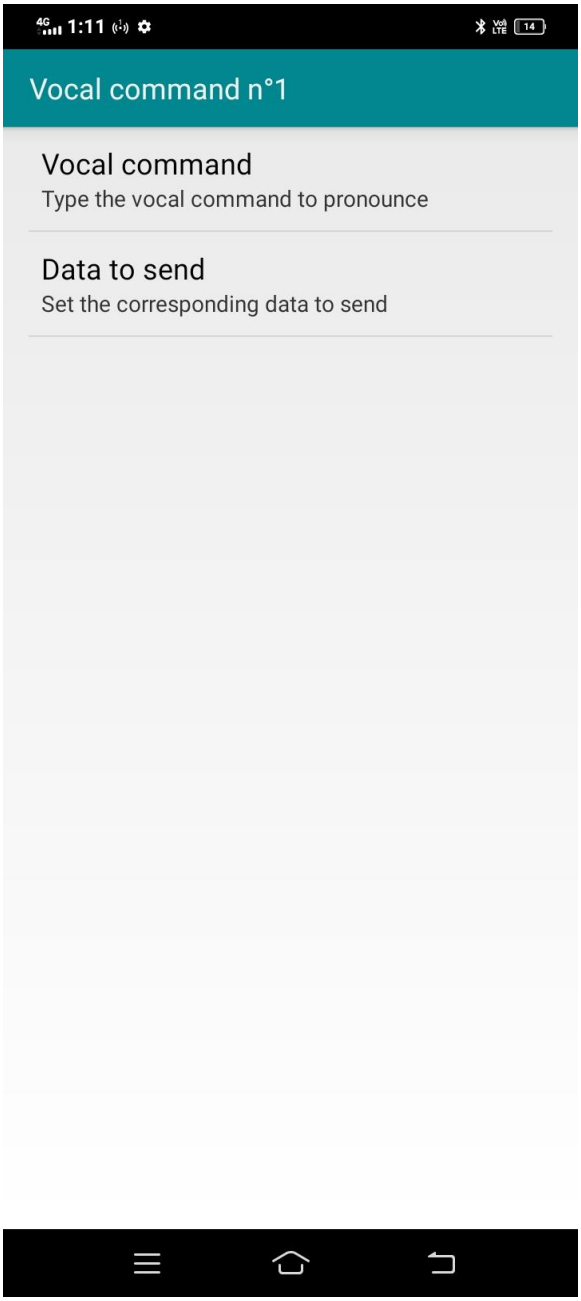
```
void voicecontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
    if (value == '^') {
      forward();
    } else if (value == '-') {
      backward();
    } else if (value == '<') {
      L = leftsee();
      servo.write(spoint);
      if (L >= 10 ) {
        left();
        delay(1000);
        Stop();
      } else if (L < 10) {
        Stop();
      }
    } else if (value == '>') {
      R = rightsee();
      servo.write(spoint);
      if (R >= 10 ) {
        right();
        delay(1000);
        Stop();
      } else if (R < 10) {
        Stop();
      }
    } else if (value == '*') {
      Stop();
    }
  }
}

int ultrasonic() {
```

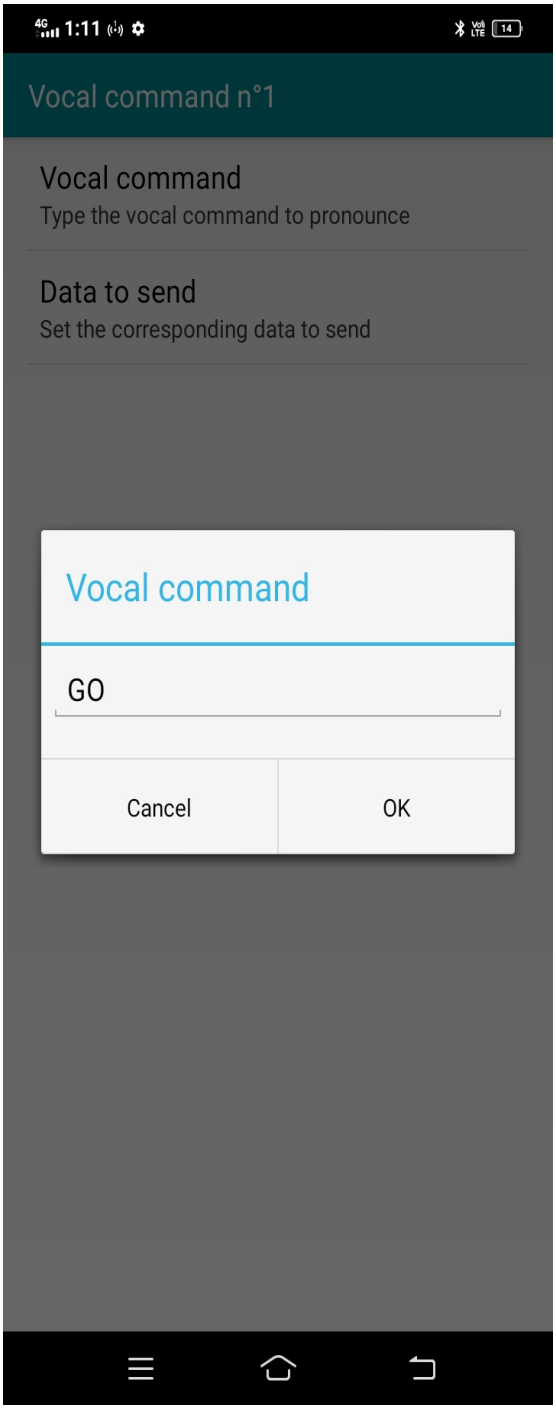
```
  digitalWrite(Trig, LOW);
  delayMicroseconds(4);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  long t = pulseIn(Echo, HIGH);
  long cm = t / 29 / 2; //time convert distance
  return cm;
}
void forward() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
}
void backward() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void right() {
  M1.run(BACKWARD);
  M2.run(BACKWARD);
  M3.run(BACKWARD);
  M4.run(BACKWARD);
}
void left() {
  M1.run(FORWARD);
  M2.run(FORWARD);
  M3.run(FORWARD);
  M4.run(FORWARD);
}
void Stop() {
  M1.run(RELEASE);
```

```
  M2.run(RELEASE);
  M3.run(RELEASE);
  M4.run(RELEASE);
}
int rightsee() {
  servo.write(20);
  delay(800);
  Left = ultrasonic();
  return Left;
}
int leftsee() {
  servo.write(180);
  delay(800);
  Right = ultrasonic();
  return Right;}
```

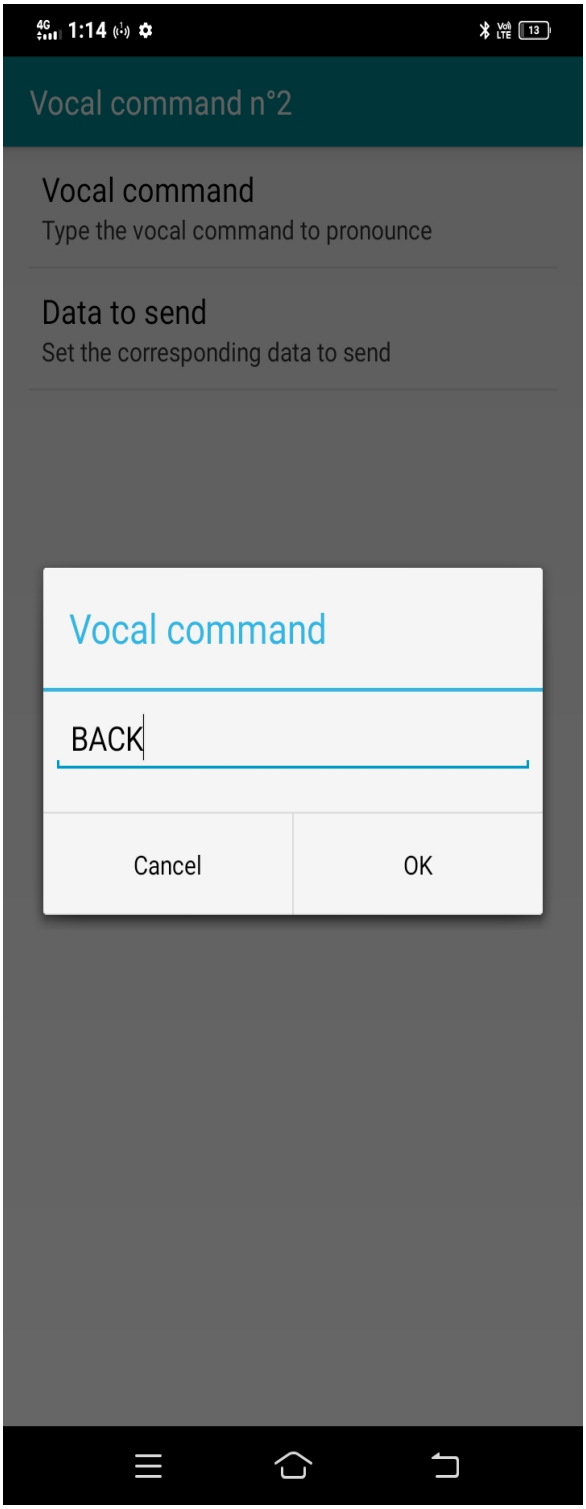
Figure 5 : Arduino bluetooth control android app


12:54
Home
Arrow Keys
Terminal
N
W
E
S
Accelerometer
A
B
C
D
Buttons & Slider
15
cm
Metrics
Voice Control
Ads by Google
Stop seeing this ad
Why this ad?

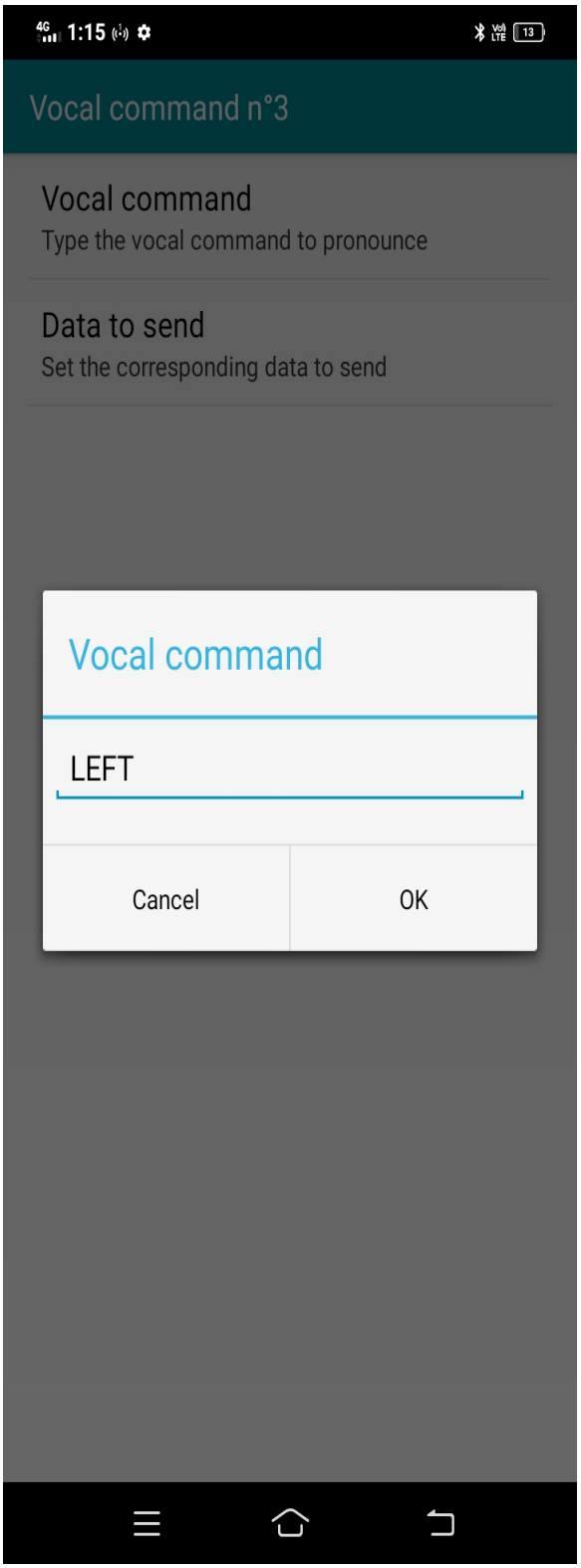

1:11
Vocal command n°1
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send

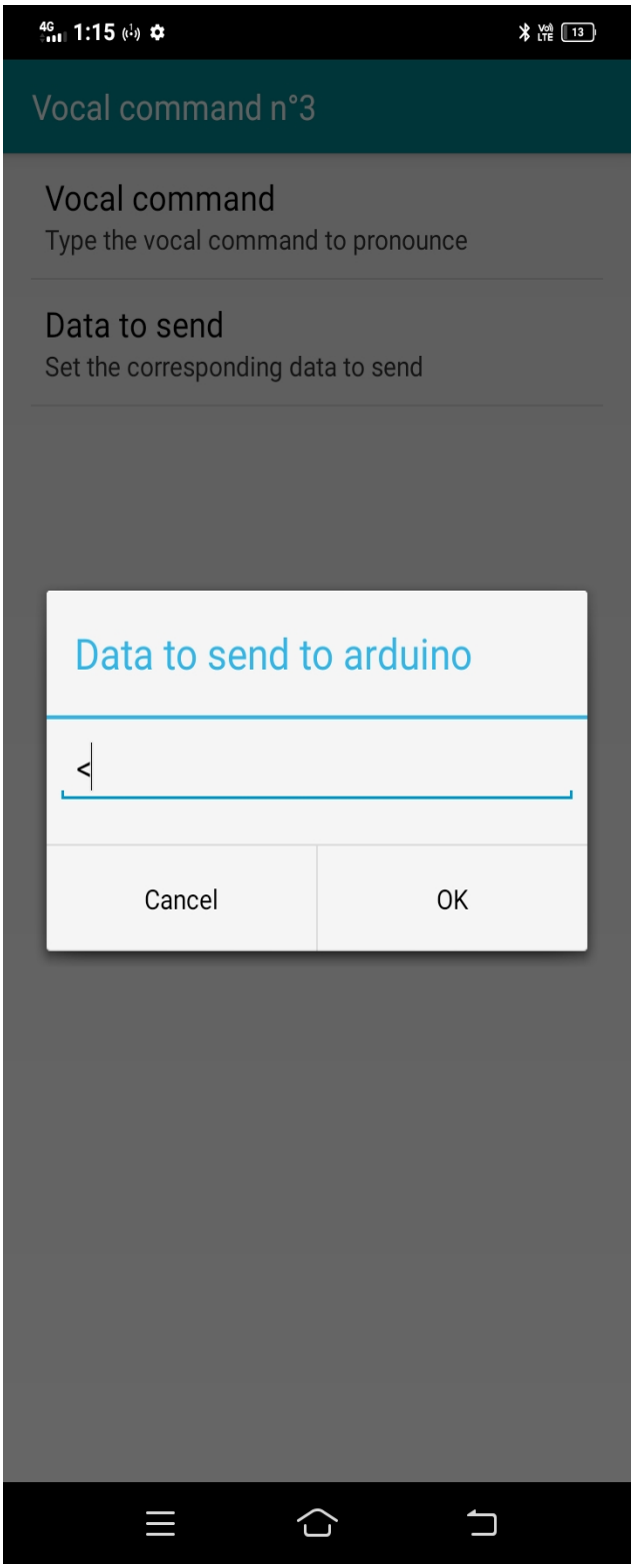

1:11
Vocal command n°1
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Vocal command
GO
Cancel
OK

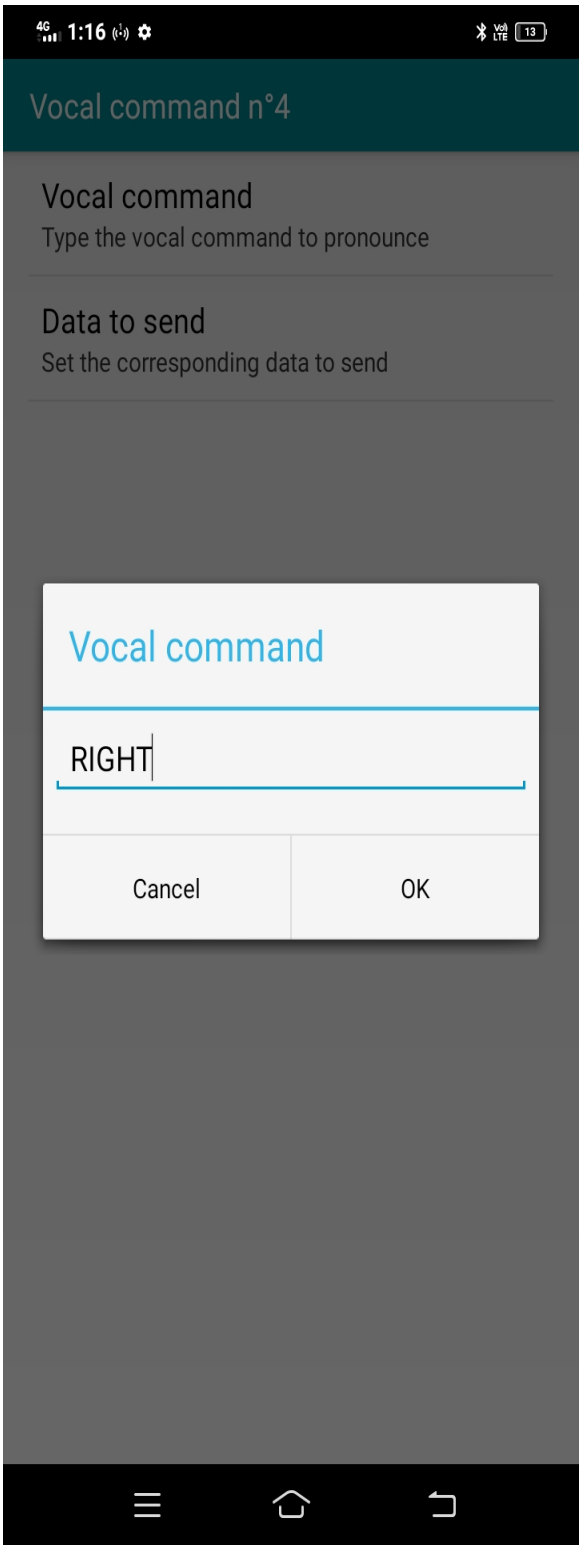

1:11
Vocal command n°1
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Data to send to arduino
^
Cancel
OK

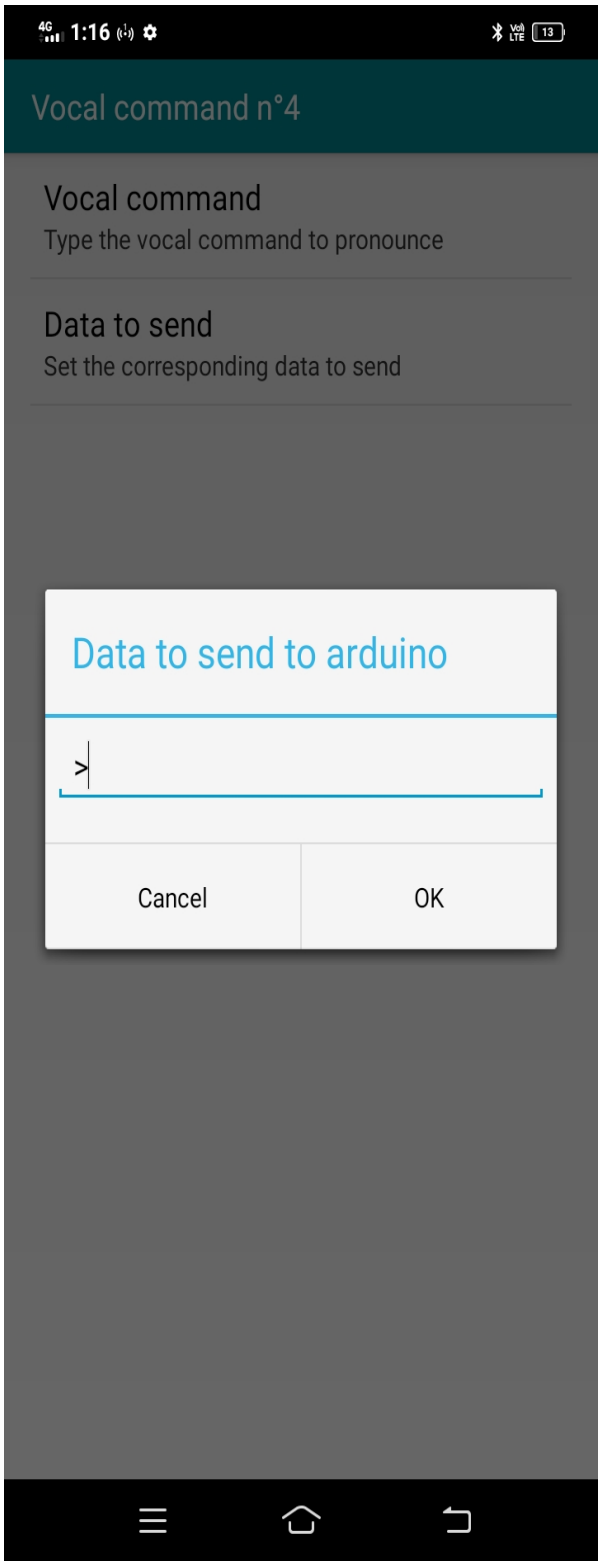

1:14
Vocal command n°2
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Vocal command
BACK
Cancel
OK

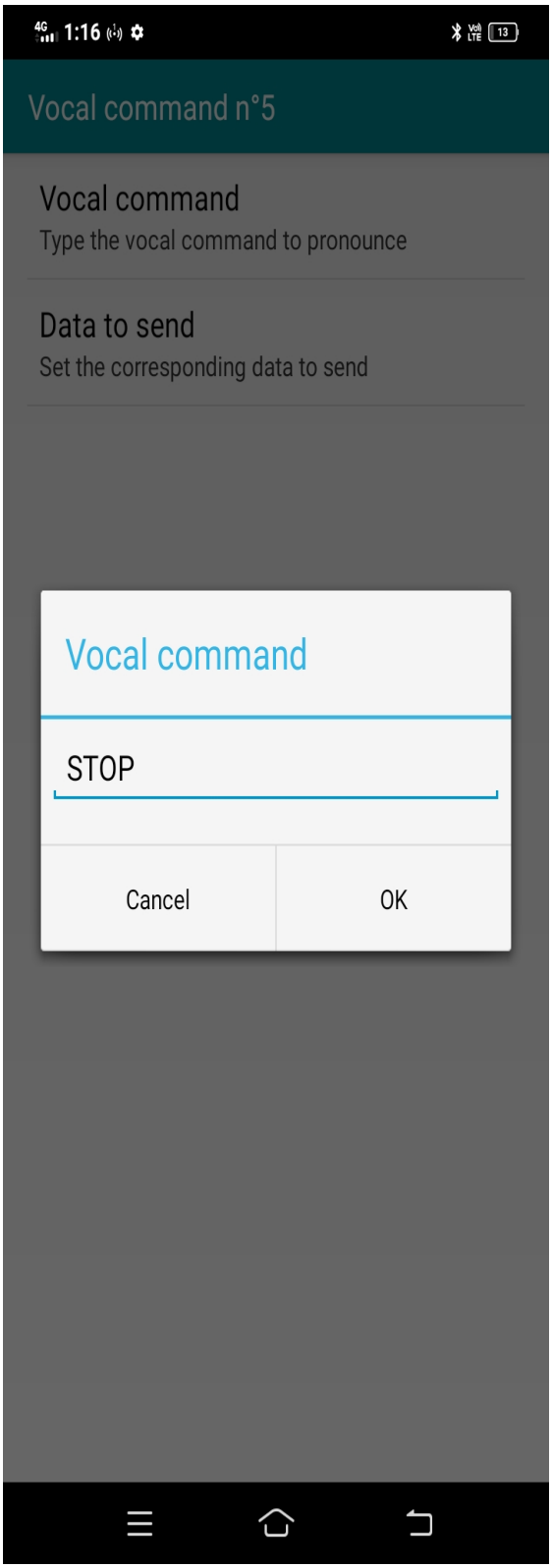

12:48
Vocal command n°2
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Data to send to arduino
-
Cancel
OK

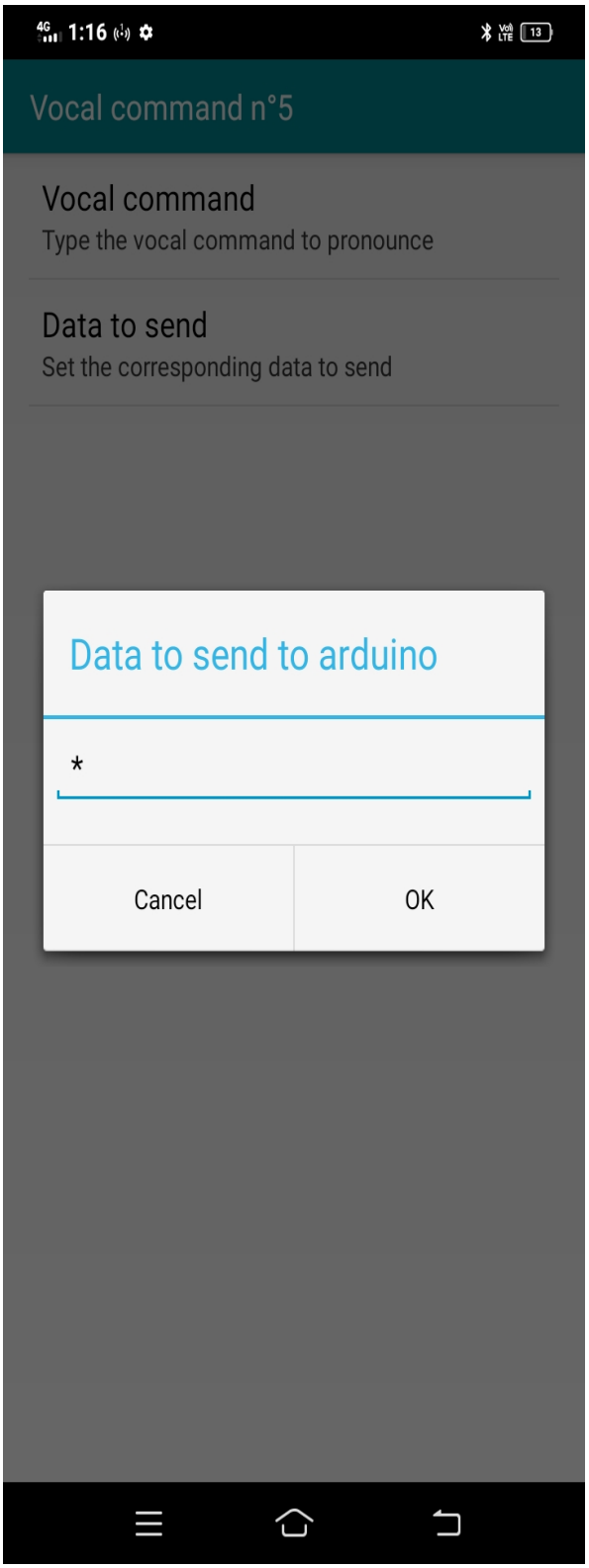

1:15
Vocal command n°3
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Vocal command
LEFT
Cancel
OK


1:15
Vocal command n°3
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Data to send to arduino
<
Cancel
OK


1:16
Vocal command n°4
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Vocal command
RIGHT
Cancel
OK


1:16
Vocal command n°4
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Data to send to arduino
>
Cancel
OK


1:16
Vocal command n°5
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Vocal command
STOP
Cancel
OK


1:16
Vocal command n°5
Vocal command
Type the vocal command to pronounce
Data to send
Set the corresponding data to send
Data to send to arduino
*
Cancel
OK

Results & Discussions

Therefore, our voice-controlled obstacle-avoiding and bluetooth controlling robot car prevents the car from collisions and is aware of its surroundings . It also finds obstacles and makes movements according to the user voice command. Our robot car is accurate because the Bluetooth module is used for communication and works best between short ranges as the robot can travel in two seconds when input is provided. We have used an ultrasonic sensor for its advantage.The ultrasonic sensor helps to detect the obstacle on the rear side hence, helping the car to avoid collisions. Voice command is used for detection of advanced communication with the car using the android app. Therefore, a good performance is available for this project.

Conclusion & Future work

1) The present automobiles are not suitable for handicapped and old age people.
2) This technique of our project will help them to drive their vehicle on their own.
3) This use of sensors will provide greater safety from sudden hits due to auto braking and slow down feature.
4) Prevention from hazardous and fatal situations.
5) Automatic braking and Bluetooth module controller will make the process handy and easy to detect and to provide judgments for the vehicle.

References

- https://www.researchgate.net/publication/351992623_Obstacle_Avoidance_and_Voice_Control_Unit_for_Autonomous_Car
- https://create.arduino.cc/projecthub/adityahazarika08/voice-controlled-obstacle-avoidance-robotic-car-using-mobile-624620
- https://create.arduino.cc/projecthub/ashshaks/bluetooth-control-robotic-car-0d9444