PROJECT REPORT ON

AN ARDUINO BASED BLUETOOTH CONTROLLED CAR

Submitted By

20BCE7314 – K. AMARENDER REDDY

20BCE7628 – P. PAVAN KUMAR

20BCE7306 – SHAIK SAMEER

20BCI7223 – SEELAM VEERA YASWANTH

20BCE7220 – SANGARAJU PRANITH VARMA

20BCE7615 – MUHAMMAD SAHIL

Under the guidance of Prof. Kumar Debasis

VELLORE INSTITUTE OF TECHNOLOGY
ANDHRA PRADESH – 522237.

VIT-AP offers distinctive education through its Undergraduate, Postgraduate and Ph.D Programmes. The teaching-learning process at the institute prepares students for the future. They are involved in an active process of applied learning with the help of experienced faculty who enable the students to channel their talent and intellect to contribute towards the development and sustainability of society. With an emphasis on innovation and applying technology to improve life, the institute offers the following programmes for students at various levels.

ABSTRACT:

It is a robotic car, which works according to the voice of humans through the Bluetooth Module as a command. Because this robotic car has an HC-05 Bluetooth module which connects to any Android device and performs the same functions by converting the voice which is given the command such as right, left, forward, back and stop etc. Here is a robotic car device that works on the voice of human. Many devices have been used in this robotic car, out of them Arduino Uno chip is more important and special, so we have named our project it, “Arduino based voice control car”. As we mentioned, many devices have been used in this, each device has its own specialty. Those complete their work accordingly and this project as well Arduino based voice control car is much useful for those areas where humans can’t reach.

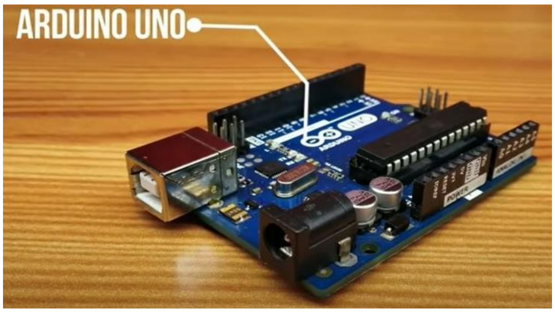
An Arduino based Bluetooth controller car that can be
controlled using three methods.

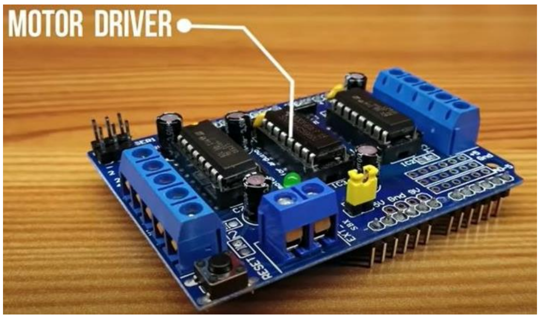
1. Manual
2. Gestures
3. Voice

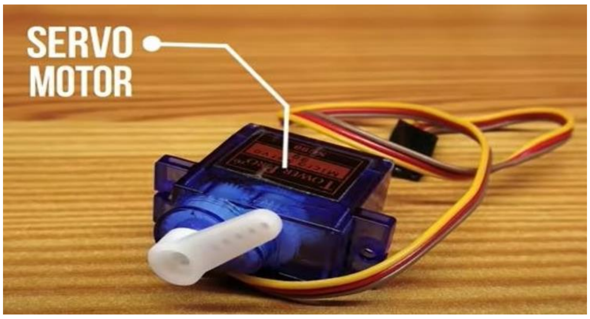
In addition to this, we are including one Bluetooth module (Ileto HC-05 Wireless Bluetooth RF Transceiver Master Slave Integrated Bluetooth Module 6 Pin Wireless Serial Port Communication BT Module for Arduino) that could let developer to control the car via voice over. The application we use for controlling via Bluetooth called “Arduino Bluetooth control” which is available at play store that helps developer connect to the car and the following voice input is received using Google voice over.

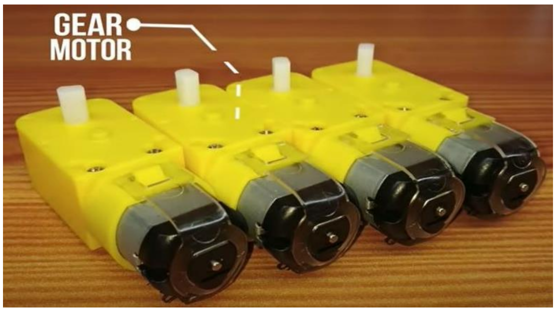
INDEX:

Keywords: Arduino Uno, Bluetooth, Robot, Voice and Android.

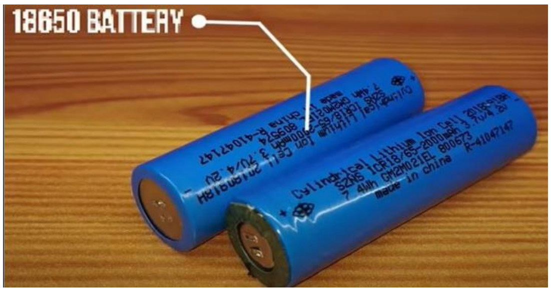








BLUETOOTH MODULE



1. INTRODUCTION

In this project, the robot control car basically works on human speech command. We can say wireless Bluetooth robot, the android application is installed in our Smartphone which works as a transmitter. The commands are given by this android application. The robotics car can be controlled wireless voice commands directly from the user. The robot can move forward, backward, left, right, stop and rotation can also be stopped. The Arduino based voice control car is interfaced with a Bluetooth module HC05 which is connected to the Arduino. The car can be moved easily from one location to another location. The car will be moving according to the voice commands given by the user. This can be moved in forward - move the car in forward direction, backward – move the car in Backward direction, left – move the car in left direction, and right – move the car in right direction, stop – stop the car, rotation – rotate the car according to the different commands given by the user. As we know that Arduino is programmable, so we have to do the programming using C or JAVA language. When the programming is of Arduino is done, we connect all the connection as required for the robot. Here is a robotic car device that works on the voice of human. We connect android application and Bluetooth module using Bluetooth link. The command is given by the AMR voice and Arduino Bluetooth controller by the user. The command by the user is converted into digital form. The range of this robot is up to 50 meters. Push button is use to change the rotation direction. If we want to make this for a certain purpose the range can be increase. These commands are received by Bluetooth module and Arduino perform the operation according to the given commands the given commands by the user are converted into digital form. Here the circuit design is used to L298N motor driver, a Bluetooth module, a 300 RPM gear motor, a power supply battery (6 volt and 9 volt), jumper wires an ultrasonic sensor, and an Arduino Uno board etc.

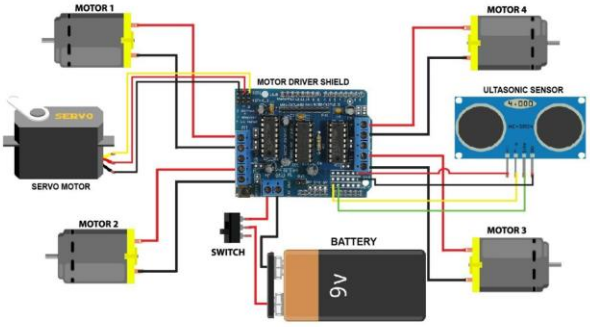
2. BACKGROUND

This project of a simple voice control robotics car it connects to the Arduino UNO Bluetooth via Bluetooth module App. Motor Driver connected to the Arduino UNO via model driver pin name in 1 connect to the Arduino pin No.4, model driver pin name in 2 connect to the Arduino pin No.5, model driver pin name in 3 connect to the Arduino pin No.6, model driver pin name in 4 connect to the Arduino pin No.8. Robotics car controlled by the Arduino UNO and Bluetooth and Arduino and Arduino UNO connected to the Bluetooth module connect via Arduino pin name TXD connect to Bluetooth pin name RXD, Arduino pin name GND connect to Bluetooth pin name GND, Arduino pin name RXD connect to Bluetooth pin name TXD Arduino pin name VCC connect to Bluetooth pin name 5V. User given the command in Bluetooth application and Bluetooth connect to Arduino UNO so Robotics car receive the command by Arduino UNO. If user say move left Arduino accept command and move left. The robotics car can move command left, right & around.

3. PROBLEM DEFINITION

This is an Arduino based Bluetooth controlled car using Arduino UNO, Motor Driver, Servo Motor, Ultrasonic sensor, wheels, Gear Motor, Bluetooth Module and a battery.

- The Preliminary idea is to Arduino based car which works through Bluetooth controls, voice commands and by controller Application.
- A self-controlled Robotic Car using Arduino. This robotic car uses Ultra Sonic Sensor to detect Obstacles which are in front of it, whenever it detects obstacles then its Ultra Sonic Sensor moves in all directions to calculates the best possible distance to move freely.



4. OBJECTIVES

- This project's main motive is to expand the knowledge for smart micro-controllers like Arduino, which is widely used in the latest IoT technologies.
- Making the use of Connectivity Modules like Bluetooth HC-05, to understand the different modes of operating and communications with the micro-controller board.

- To get familiar with Codes and programs used for the controlling of the Arduino Uno board.
- Introductory project for any Student, to get the working of the Arduino board and its software.

5. METHODOLOGY/PROCEDURE

- For this project we will use a toy car toy. Here we have selected an toy car with a left-hand drive feature. We have changed its RF circuit and our Arduino circuit. This car will have two motor motors front and rear. A front-wheel drive vehicle will be used to provide vehicle direction means a left turn or a right turn. The rear-wheel drive vehicle will also be used to drive the car forward and forward. The Bluetooth module will be used to receive commands from the Android phone and the Arduino UNO will be used to control the entire system.

- For this project we will use a toy car toy. Here we have selected an toy car with a left-hand drive feature. We have changed its RF circuit and our Arduino circuit. This car will have two motor motors front and rear. A front-wheel drive vehicle will be used to provide vehicle direction means a left turn or a right turn. The rear-wheel drive vehicle will also be used to drive the car forward and forward. The Bluetooth module will be used to receive commands from the Android phone and the Arduino UNO will be used to control the entire system.

- The Motor driver of the car is connected to Arduino to run the car. The driver's input pins are connected to the Arduino digital pins. Here we have used two DC motors to driver car in which one motor is connected at output pin of motor driver and another motor is connected with rest pins. Battery is used to power a motor driver to drive motors. Bluetooth module's rx and tx pins are directly connected to tx and rx of the Arduino. And the vcc and ground pin of Bluetooth module is connected to +5 volt and Arduino ground. And a 9-volt battery is used to power the circuit in Arduino's Vin pin.

- The Bluetooth-enabled car will move with the touch button on the Android Bluetooth mobile app. To run this project first we will need to download the Bluetooth app from Google Play Store. We may use any Bluetooth device that will support or send the data. Here are some application names that might work well. - Bluetooth Spp pro / Bluetooth controller for Arduino.

- When we will touch forward button in Bluetooth controller app then car will start moving in forward direction and it continues running forward until the finger will be moved . It also works through Voice command by saying “move forward”.

- When we will touch backward button in Bluetooth controller app then car will start moving in reverse direction and it continues reverse until the finger will be moved. It also works through Voice command by saying “move backward”.

- When we will touch left button in Bluetooth controller app then car will start moving in left direction and it continues left until the finger will be moved. In this condition front side motor will turn front side wheels in left direction and rear motor will run in forward direction. It also works through Voice command by saying “take left”.

- When we will touch right button in Bluetooth controller app then car will start moving in right direction and it continues right until the finger will be moved. In this condition front side motor will turn front side wheels in right direction and rear motor will run in forward direction. It also works through Voice command by saying “take right”.

- It also have an option making U-turn through voice control by saying “take U-turn”.

Key Functionalities and Innovations:

- Our project controls the car not through sensors or transmitter but using Bluetooth which is the easiest way to communicate today. Remote control for this project is an Android tool with a built-in Bluetooth module.
- Bluetooth is a serial communication method that can connect two devices. Here we have included a Bluetooth module that connects to the phone's Bluetooth, which allows us to communicate and allow us to control it.
- The Bluetooth module does not work alone in controlling the car. The main part of the car control is played by Arduino UNO which comes with the micro-controller . Arduino played a major role in the robotic category and made it easy to convert digital and analog signals into physical movements. This project is Bluetooth-based because it gives us a wide range of control and performance.

- It also allows us to change the remote at any time, which means we can use any Android device including phones, tablets, computers. Physical barriers such as walls, doors, etc. They do not work in car control.
- Even this car will have a rechargeable battery instead of simple ones which will help in future use as it will help us in avoiding to buy new batteries after a time.

6. RESULTS AND DISCUSSION

The result of the project “Arduino based voice control car” is moving all the direction right, left, forward, back, 360° and stop. Here is a robotic car device that works on the voice of human commands via Google. speech recognize. The Arduino Bluetooth controller app. send a voice command as data Bluetooth module understand him to Arduino Uno chip and motor diver are run the wheels. Its low power consumption and user friendly. Many devices have been used in this robotic car, out of them Arduino uno chip is more important and special, so we have named our project it, "Arduino based voice control car".

7. CONCLUSION AND FUTURE SCOPE

It is a robotic device will be controlling the human voice commanding through the Bluetooth module. This project works on human voice command with android application. It is easy to use for simple voice command forward, back, left, right, stop and rotation. The Arduino based voice control car is easily control to human voice controlling commands are successfully transmitted the signals. Voice control robot is much useful for those area where human cannot reach. This robotic car is small in size so we can use this project for spying or special, implement in this project so we can use this robotic car in police, agriculture purpose, military application, playing for kids, industrial purpose and also for surveillance devices.

FUTURE SCOPE

It can also be modernized in future Because it has many such devices which have their own characteristics. Which can also do more work, so that it can bring more modernity in the future as well As if you have seen its processing. This is a voice-controlled car. If we see it in a larger format then it can be added to any car. And with this help, any physically handicapped person can control the car with his voice and run on the road. Because "Arduino based voice control car" gives it all the control. As if he can move the car with his voice, he also has many controls without any physical touch. They can also turn the car forward, back, right, left, stop and 360° angle or as much as they want. But some controls are also near the car which is connected to it due to Artificial Intelligence. For example, if something in the way while the car is running, using an ultrasonic sensor, the car automatic reduces its speed and stops the car. Its security system is also very much because it is always monitored by the camera installed in it. At present, the range of "Arduino based voice control car" is only 50 meters. But it can be further connected to Wi-Fi to extend its range to at least 1 kilometre. So that can be used to secure the border of our country, we can monitor and spy on the border. We can also use it in agriculture, such as to avoid the chaos of the chaotic creatures in the fields, and it can be used to protect the farm crop from any kind of damage. And in future it can be made even more excellent and useful

8. REFERENCES

https://www.arduino.cc/

9. CODES

```
//voice based sensor controlled car


#include <AFMotor.h>
#include <Servo.h>
#include <NewPing.h>


#define TRIG_PIN A0
#define ECHO_PIN A1
#define MAX_DISTANCE 300




NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
```

```
String command;

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

Servo myservo;
int speed = 200;

void setup() {

Serial.begin(9600);
myservo.attach(10);
myservo.write(45);

}

void loop() {
 int distance = sonar.ping_cm();
 if(Serial.available()>0) {
  command="";
  delay(2);
  command = Serial.readString();
  delay(2);
  Serial.println(command);
  if(command == "*switch to speed mode#"){
     speed = 255;
  }
  else if(command == "*switch to normal mode#")
  {
      speed = 190;
  }
  if(command == "*go back#"){
    backward();
  }else if(command == "*turn left#" || command == "*take left#" ){
    left();
  }else if(command == "*turn right#" || command == "*take right#" ){
    right();
  }else if(command == "*stop#") {
    Stop();
 }
  else if(command == "*take u turn#" || command == "*take you turn#" || command == "*take
U turn#") {
    uturn();
  }


}
while(command == "*go#") {
  forward();
}
}


void forward() {


  int distance = sonar.ping_cm();
  if(distance < 10)
  {
    Stop();
    command = "";
    }
  else
  {
  motor1.setSpeed(speed);
  motor1.run(FORWARD);
  motor2.setSpeed(speed);
  motor2.run(FORWARD);
  motor3.setSpeed(speed);
  motor3.run(FORWARD);
  motor4.setSpeed(speed);
  motor4.run(FORWARD);
  }

}

void backward () {
  motor1.setSpeed(speed);
  motor1.run (BACKWARD);
  motor2.setSpeed(speed);
  motor2.run (BACKWARD);
  motor3.setSpeed(speed);
  motor3.run (BACKWARD);
  motor4.setSpeed(speed);
  motor4.run (BACKWARD);
  delay (2000);
  motor1.run (RELEASE);
  motor2.run (RELEASE);
  motor3.run (RELEASE);
  motor4.run (RELEASE);
}
```

```
void left () {
 myservo. write(90);
 delay (500);
 myservo. write(45);
 delay(500);
 motor1.setSpeed(255);
 motor1.run (BACKWARD);
 motor2.setSpeed(255);
 motor2.run (BACKWARD);
 motor3.setSpeed(255);
 motor3.run (FORWARD);
 motor4.setSpeed(255);
 motor4.run (FORWARD);
 delay (400);
 motor1.run (RELEASE);
 motor2.run (RELEASE);
 motor3.run (RELEASE);
 motor4.run (RELEASE);
}

void right () {
 myservo. write(-90);
 delay (500);
 myservo. write(45);
 delay (400);
 motor1.setSpeed(255);
 motor1.run (FORWARD);
 motor2.setSpeed(255);
 motor2.run(FORWARD);
```

```
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
  delay(400);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
void uturn() {
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
  delay(800);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}



void Stop() {
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}

#include <AFMotor.h>

//initial motors pin
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

char command;

void setup()
{
  Serial.begin(9600);  //Set the baud rate to your Bluetooth module.
}

void loop(){
  if(Serial.available() > 0){
    command = Serial.read();
    Stop(); //initialize with motors stoped
    //Change pin mode only if new command is different from previous.
    //Serial.println(command);
    switch(command){
    case 'F':
      forward();
      break;
    case 'B':
       back();
      break;
    case 'L':
      left();
      break;
    case 'R':
      right();
      break;
    }
  }
}

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

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

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

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