Robotics Project Report

ECE 2008

Robotics and Automation

Voice Controlled Robot using Bluetooth Module

PROJECT REPORT

By

Sourodipto Bhowmick (19BEC0803)

Devakinandan Palla(19BEC0812)

TABLE OF CONTENTS

1. Hardware and video demonstration clip

Conclusion

1. References

Abstract

This project was developed in a way that the robot is controlled by voice commands. An android application with a microcontroller is used for required tasks. The connection between the android app and the vehicle is facilitated with Bluetooth technology. The robot is controlled by buttons on the application or by spoken commands of the user. The movement of the robot is facilitated by the two dc servo motors connected with microcontroller at the receiver side. The commands from the application is converted in to digital signals by the Bluetooth RF transmitter for an appropriate range (about 100 meters) to the robot. At the receiver end the data gets decoded by the receiver and is fed to the microcontroller which drives the DC motors for the necessary work. The aim of Voice Controlled Robotic Vehicle is to perform the required task by listening to the commands of the user. A prior preparatory session is needed for the smooth operation the robot by the user.

For the same a code is used for giving instruction to the controller.

After designing the above prototype, a search for adding new feature has been done and a whole new model has been proposed which is centered around ESP-32(with 2 microphones).This model has provided a lot more advantages like decrease in circuit complexity(because esp 32 didn’t need wifi/Bluetooth module),relatively cheaper cost and can work in harsher conditions.

Software used and Why?

The Bluetooth connected on the Arduino board receives text from the Android app as characters and stored them as string to the assigned String. There are words pre-programmed (forward, reverse, right , left and stop) to the arduino , whenever the received text matches with the pre- programmed words ,the arduino executes the command that

assigned to the words. Arduino can connect to Laptop to monitor serial communication and check the working process and the words received by the Bluetooth.

The overall system design consists of following modules:

- ▶️ Hardware Connection.
- ▶️ Application Software
- ▶️ User Access
- ▶️ Voice Recognition
- ▶️ Preprocessing
- ▶️ Prediction of Output

Real life Applications

▶️ The robot is useful in places where humans find difficult to reach but human voice reaches.

E.g. in fire situations, in highly toxic areas.

- ▶️ Command and control of appliances and equipment
- ▶️ Telephone assistance systems
- ▶️ Speech and voice recognition security systems
- ▶️ The robot can be used for surveillance or reconnaissance.

Hardware components and its contribution

- ▶️ L293D motor driver shield-used to drive the 4 gear motors and servo motors at the same time
- ▶️ Ultrasonic sensor-proximity sensor and obstacle detection system
- ▶️ Bluetooth module-for wireless serial connection to the Android app
- ▶️ Servo motor- for rotatory motion
- ▶️ Li-ion battery& its holders

▶️ Robot wheel-to travel back, front , right and left with the instructions and suitable connection

▶️ Gear motor-adjusting the mechanical power of a system

▶️ Arduino UNO board-to read all the inputs from the given connection

▶️ Jumper wires

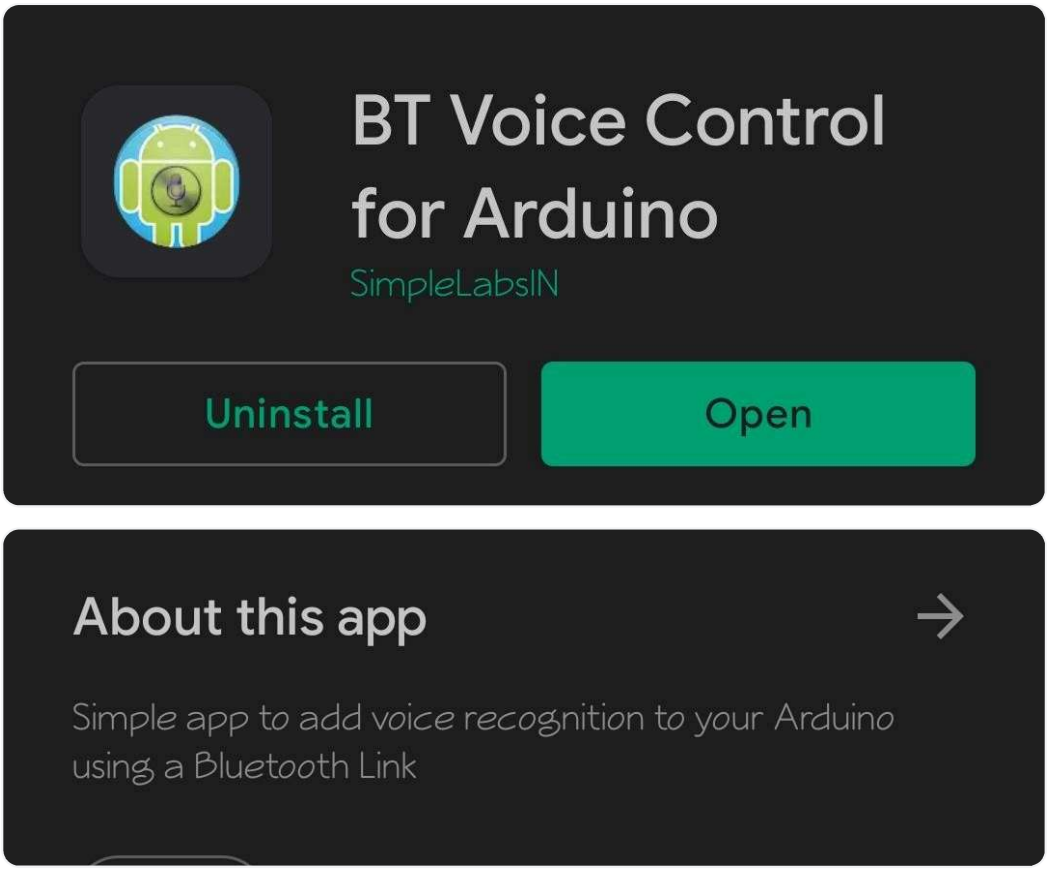
For the updated model the core components were:

i)INMP441 I2S Microphone

ii) ICS-43434 I2S Microphone: ESP32 Dev board

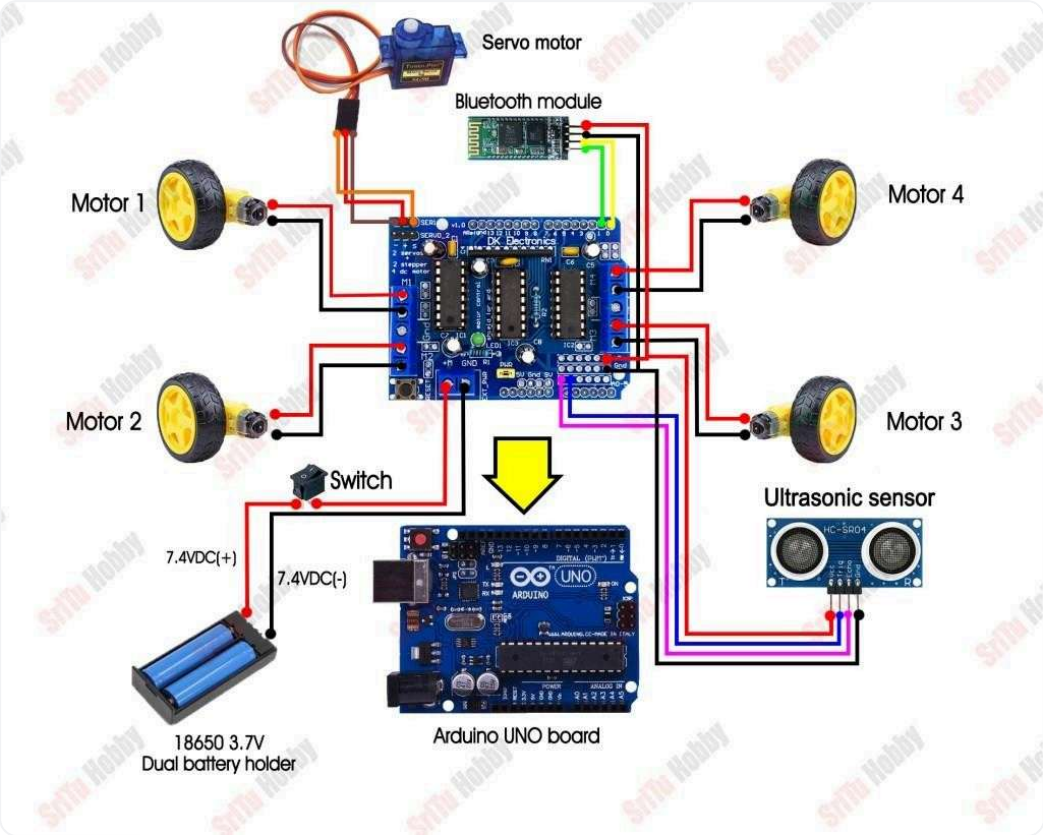
iii) Continuous 360-degree Servo

App installed to interpret our speech command and give commands to the robot:

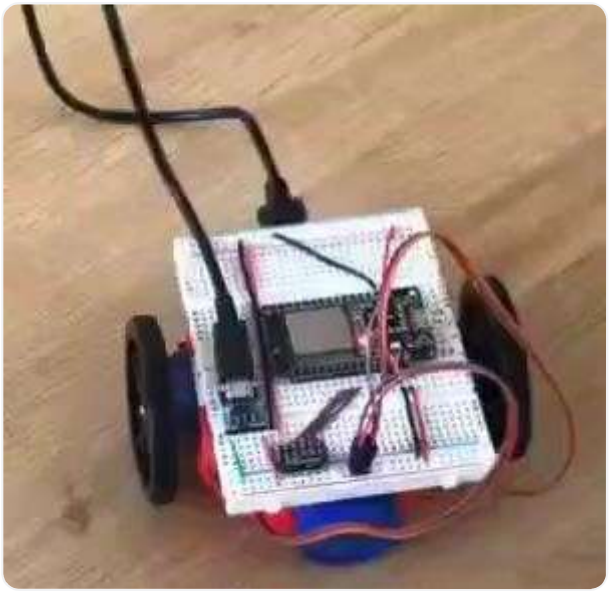


(*No such third party apps were required for the updated ESP32 centric version)

Model



Updated version:



Code on Arduino IDE

sketch_nov13a.ino | ReadMe.adoc

```
String readvoice;
int k=0;
void setup() {
Serial.begin(9600);
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
}

void loop() {
while (Serial.available())
{
delay(3);
char c = Serial.read();
readvoice += c;
}

if(readvoice.length() &gt; 0)
{
Serial.println(readvoice);

if(readvoice == "forward")
{
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
k=1;
}
```

```
if(readvoice == "backward")
{
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
k=2;
}

if(readvoice == "left")
{
if (k==2)
{
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
delay(1000);
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
}
else
{
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
delay(1000);
```

```
{
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
delay(1000);
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
}
}

if(readvoice == "right")
{
if (k==2)
{
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
delay(1000);
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
}
else
{
digitalWrite(2, LOW);
```

```
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
}
else
{
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
delay(1000);
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
}
}

if(readvoice == "stop")
{
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
}
}
readvoice="";
}
```

***A similar code has been inputted to the updated esp 32 centric model**

by installing Esp32 Add on in Arduino IDE

Working

->This is a simplified version of any other voice control robot. No complex coding, easy to understand coding with easy algorithm.

->Download BT Voice Control for Arduino App from Google Playstore.

->The app is developed in such a way that it convert the voice command to text and transfer the text to the connected Bluetooth

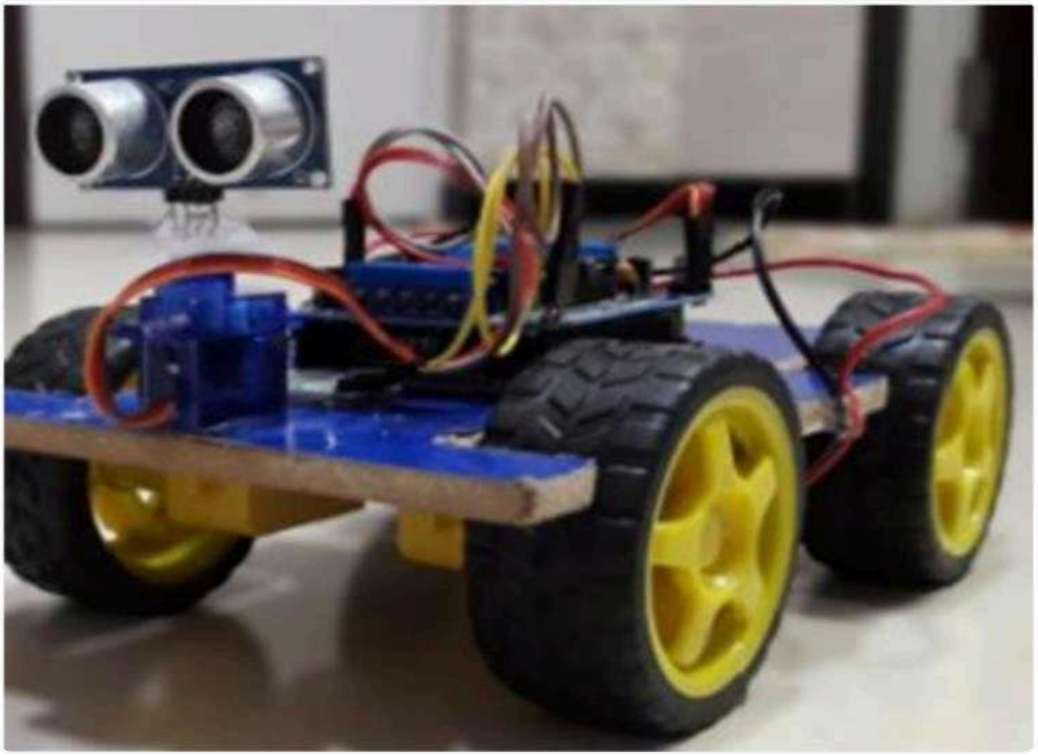
device.

->The Bluetooth connected on the Arduino board receives text from the Android app as characters and stored them as string to the assigned String.

-> There are words pre-programmed (forward, reverse, right , left and stop) to the arduino , whenever the received text matches with the pre-programmed words ,the arduino executes the command that is assigned to the words.

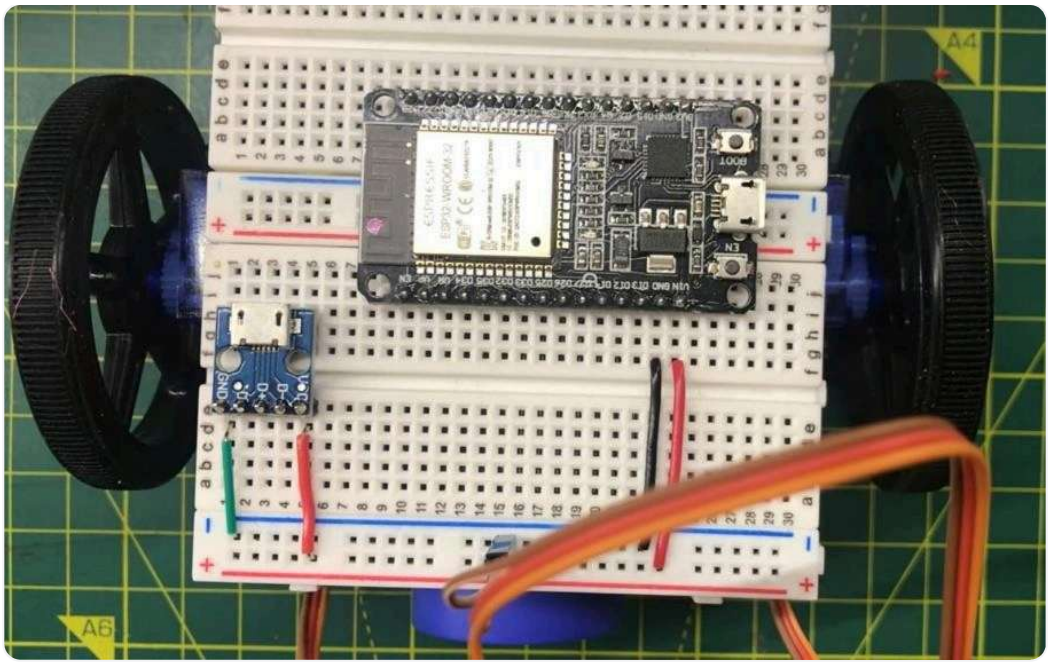
->Arduino can connect to Laptop to monitor serial communication and check the working process and the words received by the Bluetooth.

For the updated version, we have installed the esp 32 add on in the Arduino Ide- while some features we thought of implementing were disabled due to less functionalities, we have managed to run the basic code on to the model and it has run successfully which could be seen in our demonstration in the video clip.



Hardware and video demonstration clip Basic model:

Hardware for new feature incorporation



Steps for Hardware:

First , we bolted the two servos onto the chasis Then attached the wheels

Then placed the breadboard and fitted the esp 32 and the appropriate connections referring to the model(picture provided in above-Pg.8) into it in the top of middle of both ends

Then we ran the code in the Arduino IDE +esp 32 add on

Here are the links of the video clips where we have successfully demonstrated our project:

Basic version: https://drive.google.com/file/d/1cIH16kdHqrrPPRWdTIs5GpOBxGSINzm2/view?usp=drivesdk

Updated version with different hardware components but same working principle and code:

https://drive.google.com/file/d/1aUHIKWdoc2FkciCEj Ovhh-so061D-pF_/view?usp=drivesdk

Conclusion

After coming up with an updated version which has a remodeled hardware design but with using the similar coding and working principle we have inferred these results

▶️ Decreased complexity because esp-32 has an in built wi fi capabilities whereas Arduino employs a separate Bluetooth/wi-fi module. Thus lesser components have been used and the system has been simple through out.

▶️ From what we have done, the esp-32 centric project required significantly lesser cost(around Rs. 350-400 lesser).

▶️ the Esp32 is energy efficient with wireless technology and can withstand hotter temperatures (till 125 degree Celsius) than an Arduino which gives it an upperhand to work in harsher conditions.

Apart from the enabled advantages the updated Esp 32 centric version has provided, the prototype model on its whole has a lot of real life applications and can be used for:

i) employing a robot by commanding through voice in situations unsafe for humans(like fire accidents)

ii) Can be commanded to fix some issues when a platform is set in case of deep water drilling or fixing some small issues which are too minor to employ divers into the sea/ocean.

iii) It can be used for random activities for limbs disabled persons/ older people who can't walk whose job to come and go to keep and get random things would be reduced, thus reducing the potential work they are supposed to do by just commanding the robot through speech command.

iv) By incorporating NCF, it can also be used in noisier places like the cities to get the job done over voice commands.

References

https://how2electronics.com/wireless-bluetooth- controlled-robot-using- arduino/#:~:text=The%20smartphone%20has%20an%20Android,%2D05%20or%20HC%2D06.

https://www.electronicshub.org/bluetooth- controlled-robot-using-arduino/

https://how2electronics.com/wireless-voice-controlled-robot- car-using-arduino/