

**SPEECH RECOGNITION ROBOT THROUGH DEVELOPMENT OF ARTIFICIAL
INTELLIGENCE**

BY

**Md. Shakil Mahmud
ID: 152-15-517**

AND

**Md. Rabius Sani Khan
ID: 152-15-532**

This Report Presented in Partial Fulfilment of the Requirements for the Degree
of Bachelor of Science in Computer Science and Engineering.

Supervised By

Ohidujjaman Tuhin

Lecturer

Department of CSE
Daffodil International University

Co-Supervised By

Mr. Md. Reduanul Haque

Lecturer

Department of CSE
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
APRIL 2019**

APPROVAL

This Project titled “Speech recognition robot through development of artificial intelligence”, submitted by Md. Shakil Mahmud, ID No: 152-15-517, Md. Rabius Sani Khan, ID No: 152-15-532 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 6th April, 2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain

Chairman

Professor and Head

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University

Dr. S M Aminul Haque

Internal Examiner

Associate Professor & Associate Head

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University

Saif Mahmud Parvez

Internal Examiner

Lecturer

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University

Dr. Mohammad Shorif Uddin

External Examiner

Professor

Department of Computer Science and Engineering

Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ohidujjaman Tuhin, Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Ohidujjaman Tuhin

Lecturer

Department of CSE

Daffodil International University

Co-Supervised by:

Mr. Md. Reduanul Haque

Lecturer

Department of CSE

Daffodil International University

Submitted by:

Md. Shakil Mahmud

ID: 152-15-517

Department of CSE

Daffodil International University

Md. Rabius Sani Khan

ID: 152-15-532

Department of CSE

Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Supervisor Ohidujaman Tuhin**, Lecturer, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Algorithm and Artificial Intelligence*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain**, Professor and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

In modern day we just don't use machine, we are connected with it. We are not satisfied with our software industries. We need something more than that. We need a combination of software and hardware what can perform some work what we want, what we need and what we lack of. It should be something smart and effective. In this era the smartest thing is our smartphone. We just do not use it, we live with it. What if we make a hardware-based robot that can perform some intelligent work with the help of smartphone? What if we can use it via voice control and wirelessly. This paper contains and analyses the research, procedure and information about a robot with artificial intelligences, voice controller via smartphone.

TABLE OF CONTENTS

CONTENS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
List of figures	viii
CHAPTER	
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objective	1
1.4 Expected Outcome	2
CHAPTER 2: BACKGROUND	3-4
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Studies	3
2.4 Scope of the problem	3
2.5 Challenges	4
CHAPTER 3: REQUIREMENT SPECIFICATION	5-7
3.1 Process Diagram	5

3.2 Circuit Diagram	6
3.3 Block Diagram	6
3.3 Descriptions of Robotic Parts	7
CHAPTER 4: PROGRAM COMPILATION	13-14
4.1 Hardware and Software Requirement Specification	13
4.2 Programming Language	13
4.3 Integrated Development Environment	14
CHAPTER 5: IMPLEMENTAION AND TESTING	15-16
5.1 Implementation of Robotic Vehicle	15
5.2 Implementation of Android Application	16
CHAPTER 6: CONCLUTION AND FUTURE SCOPE	1-3
6.1 Discussion and Conclusion	17
6.2 Future plan	17
REFERENCES	18
APPENDIX	19-37

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Process Diagram	5
Figure 3.2: Design of Circuit Diagram	6
Figure 3.3: Block Diagram	6
Figure 3.4.1: Arduino Uno	7
Figure 3.4.2: L298N Stepper Motor Drive	8
Figure 3.4.3: Gear Motor DC	8
Figure 3.4.4: I2C LCD monitor 16*2	9
Figure 3.4.5: IR obstacle sensor module.	9
Figure 3.4.6: Breadboard.	10
Figure 3.4.7: Wheel	10
Figure 3.4.8: LED light.	11
Figure 3.4.9: 9V rechargeable Battery.	11
Figure 3.4.10: HC-05 BT Module.	12
Figure 5.1: Robotic vehicle	15

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this modern era, we want everything automated. We want physically our work be done. For this type of work, we need some robot that has some Intelligence. Without intelligence it can't work properly. One important thing is we have to communicate with it. To make it simple we should make it remotely and naturally like gesture or by talking. Or we can use a smart handy device like a cross platform between us and robot. A smartphone is really a handy device.

1.2 Motivation

For a hardware-based work there are many challenges we face my problems. First dilemma was to choose a suitable microcontroller. There were many of those. We have chosen Arduino Uno for its low power consumption, durability, cheapness and availability [1]. Then there was another challenge to fix the exact project. Where we will apply our artificial intelligence. How it will understand natural languages.

After some research we have decided following features to keep in our robot:

1. we will make a robotic vehicle that can run or move.
2. It will sense the obstacle.
3. it understand our languages
4. It will make some decision.

1.3 Objectives

Our objective is our robotic vehicle can recognize voice command and give response so that we can easily interact with that vehicle. If we noted our objectives then we can describe it given below:

- a. Robotic vehicle that can run and move.
- b. Speech recognition program through artificial intelligence.
- c. It can make some decision by its own.
- d. Understand Natural Language

1.4 Expected Outcome

- a. Control device by voice command.
- b. Android application can understand Natural Language.
- c. Android application is the medium of vehicle and user.

CHAPTER 2

BACKGROUND

2.1 Introduction

In this modern era, we want everything automated. We want physically our work be done. People want to save their time. Keep their energy and make their work simple. For this type of work, we need some robot that has some Intelligence. Without intelligence it can't work properly. One important thing is we have to communicate with it. To make it simple we should make it remotely and naturally like gesture or by talking.

Or we can use a smart handy device like a cross platform between us and robot. A smartphone is really a handy device. We will give command via smartphone and convert for the Arduino to perform some particular action.

2.2 Related Works

This project is materialized with Arduino and smartphone. Several projects are related with our project. But in some case, they are different like some of them are just controlled by android. or Some of them don't have Artificial intelligence.

1. In 2016, Vineeth Teeda, K.Sujatha, Rakesh Mutukuru spoke to a project where they wanted a system where Arduino will be controlled via voice. Where on board mic will take the input [2].

2. In 2017 Shivanand proposed a project that also based on Arduino and android. It's also controlled by android app. They also used 4wd car chasing [3].

2.3 Comparative Studies

This project focused on how we can implement artificial intelligence in any micro controller and control it by our own natural gesture or words we can see from our related work section previous projects were costly and less user friendly.

2.4 Scope of the problem

We can get the idea from our project. This kind of projects has many applications. Now a days it is very important to use ai in our every particular work and we want ot communicate with

our machine by our natural languages. Then we can get the idea that idea that scope is not limited. Like we can use for education, medical purpose etc.

2.5 Challenges

In this project there are some challenges that we have to face:

- a. Size problem
- b. Weight problem
- c. Power problem
- d. Android app development in many sector
- e. Hardware and software combination.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Process Diagram

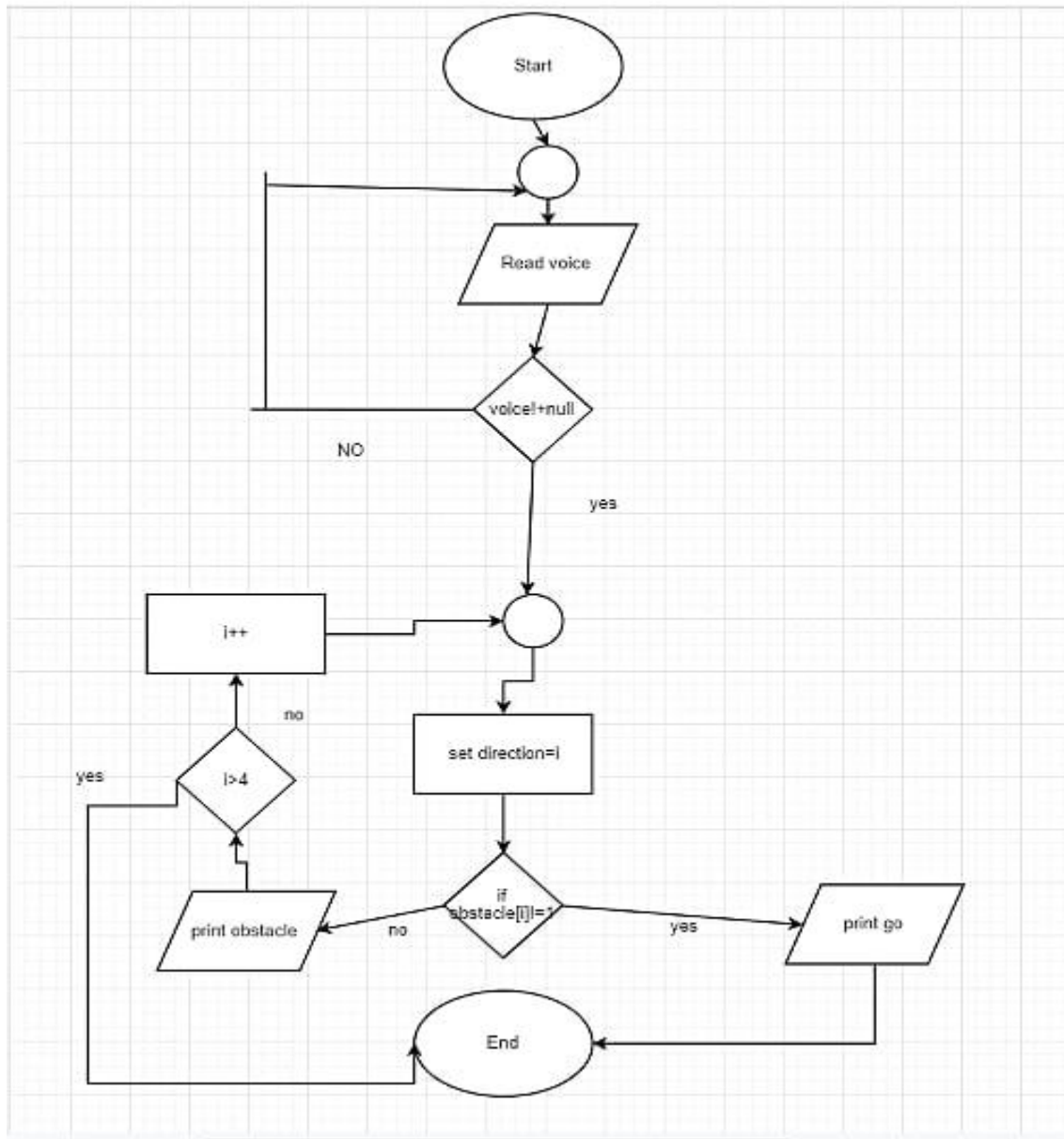


Figure 3.1: Process Diagram

3.2 Design of Circuit Diagram

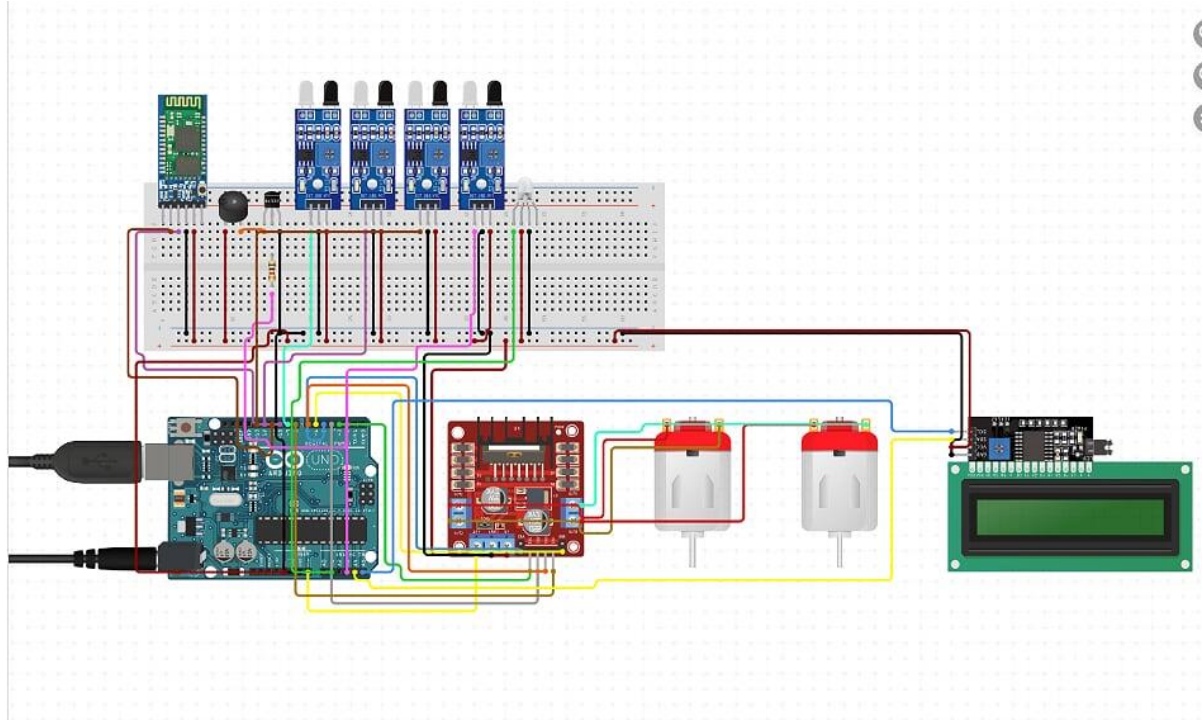


Figure 3.2: Design of Circuit diagram.

3.3 Block Diagram

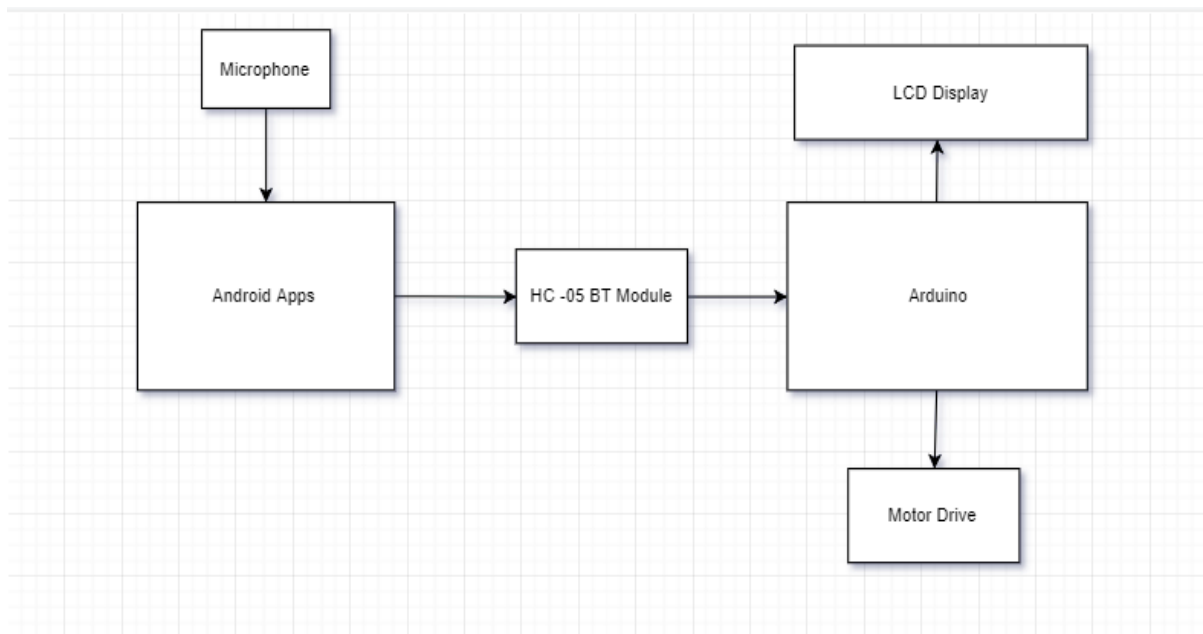


Figure 3.3: Block Diagram

3.4 Description of Robotic parts

3.4.1 Arduino

Arduino is a single board microcontroller. It is the brain of our projects. It can be used to control any hardware. We have connected all of our hardware with it. It's an open source hardware licensed under LGPL or GPL [4]. Arduino doesn't have any default operating system(os). Storage type is Flash and EEPROM and memory is SRAM [5].



Figure 3.4.1: Arduino Uno

3.4.2. L298N Stepper Motor Drive:

Stepper motor drive normally used for control any motor. It can provide 12-35 dc voltage to our targeted hardware. We connect this drive to our Arduino and we connect 4 dc gear motor with it.

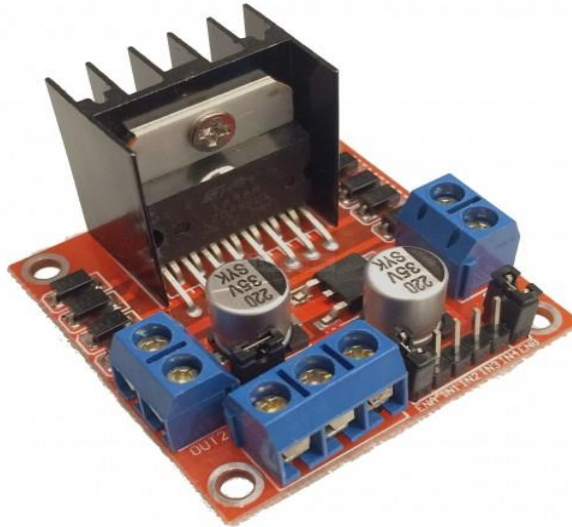


Figure 3.4.2: L298N Stepper Motor Drive.

3.4.3. Gear Motor DC:

To run any wheel by electricity we need motor. To make this robot we used 4 gear dc motor. It is more suitable for robot than any other motor. Every motor need 3V voltage to run [6].



Figure 3.4.3: Gear Motor DC

3.4.4. LCD Display 16*2:

We used LCD display to confirm the signal and texts which is sent to Arduino via HC-05 BT module. This 16*2 LCD display can print 2 lines. Each line can contain 16 character.

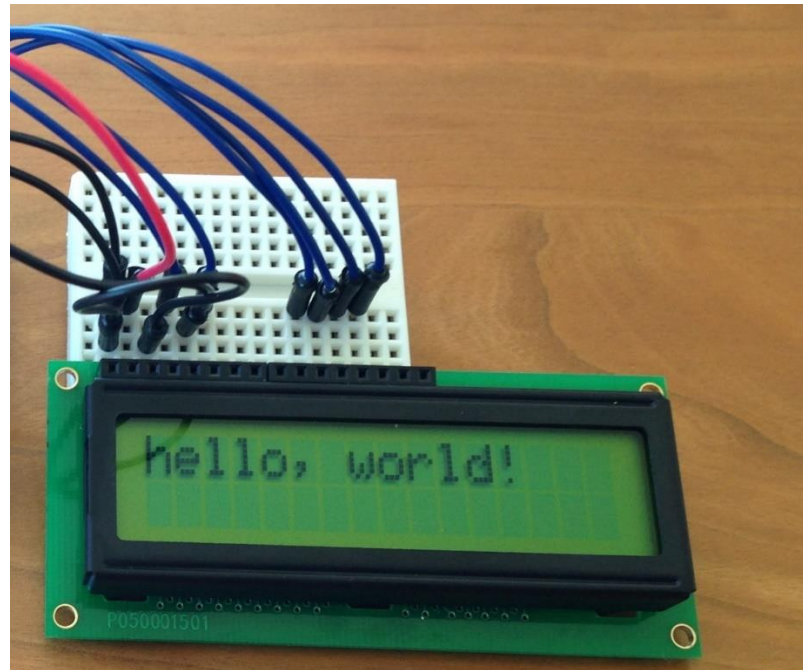


Figure 3.4.4: I2C LCD monitor 16*2.

3.4.5. IR obstacle sensor module

IR obstacle sensor module can detect obstacle by Infrared ray. we used four of this module to detect obstacle and take decision.

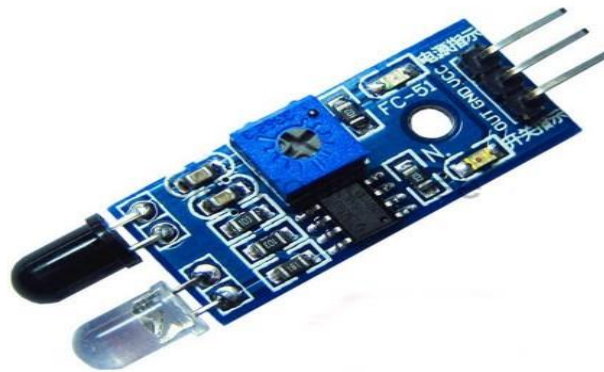


Figure 3.4.5: IR obstacle sensor module.

3.4.6. Breadboard

Breadboard is normally used for making circuit. To create parallel and series connection we used breadboard. All of the module is connected in this breadboard.

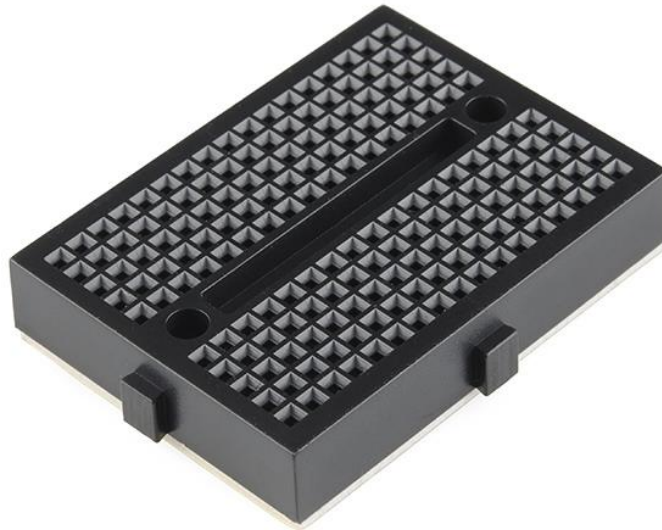


Figure 3.4.6: Breadboard.

3.4.7. Wheel

To run any car wheel is needed. We used four wheels to run this car which is connected with four gear motor.



Figure 3.4.7: Wheel.

3.4.8.LED light

We used some led light to check either some of our module is working or not.



Figure 3.4.8: LED light.

3.4.9 Battery

To run any electronics, we need power. To run our Arduino board, we needed 9v and for motor drive we used 2 9v battery. Totally we used 3 9v battery.



Figure 3.4.9: 9V rechargeable Battery.

3.4.10 HC-05

HC-05 BT Module is a Bluetooth module usually used for contact wirelessly between a short distance. We used this module in our project to connect our android with Arduino Uno.



Figure 3.4.10: HC-05 BT Module.

CHAPTER 4

PROGRAM COMPILATION

4.1 Hardware and Software Requirement Specification

4.1.1 Hardware

To create this project, we used several modules then we connect them with Arduino via different pin. 0, 1 pin is used respectively tx and rx of hc-05 bt module. 3, 4, 5, 6 is respectively used for motor driver's n4, n3, n2, n1. Four IR obstacle detector is connected with 7, 8, 9 and 10.

Pin 11 is connected with Buzzer Module and pin 12 is connected with an LED.

Here is a diagram of our Arduino and integration of all of the tools.

4.1.2 Software

Arduino is an open source software for implementing code in Arduino. It is IDE. We write our Arduino's code is here.

4.2 Programming Language

4.2.1 C

It is a medium level language is very important which have both feature of high-level and low-level programming language. we can write script for drivers and kernels. It also offers dynamic memory allocation. We use c for writing Arduino's code.

4.2.2 Java

We use programming language Java for create our android application. So, I think it very important for create Android Application. We all know google introduce a platform Android Studio to android developer which is very user friendly. There have two combine language Java and XML. So, java is the comfortable language for create Android application.

4.2.3 XML

Extensible Mark-up Language is a text-based Mark-up language. XML tags identify the data and are used to store and organize the data which are used to display the data. XML code is use to create user interface of our Android Application. In the IDE Android Studio, we develop our Android Application. In Android Studio, XML is the language to crate interface.

4.3 Integrated Development Environment

4.3.1 Starter

As we said our robot is controlled by android based smartphone app. So, we developed an android app with android studio. Which is interfaced with Arduino.

4.3.2 Android Studio

Android Studio is the official IDE for developing Google's android operating system's apps. For many reasons' android is very popular for many types of applications. Especially for wireless programming, socket programming. In this project we need a device to convert the natural languages to signal. Android app will do the trick.

First, we developed GUI for our android app. These are the pictures from android studio of our app's interfaces.

Then we developed the apps code according to our functionality.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Robotic Vehicle

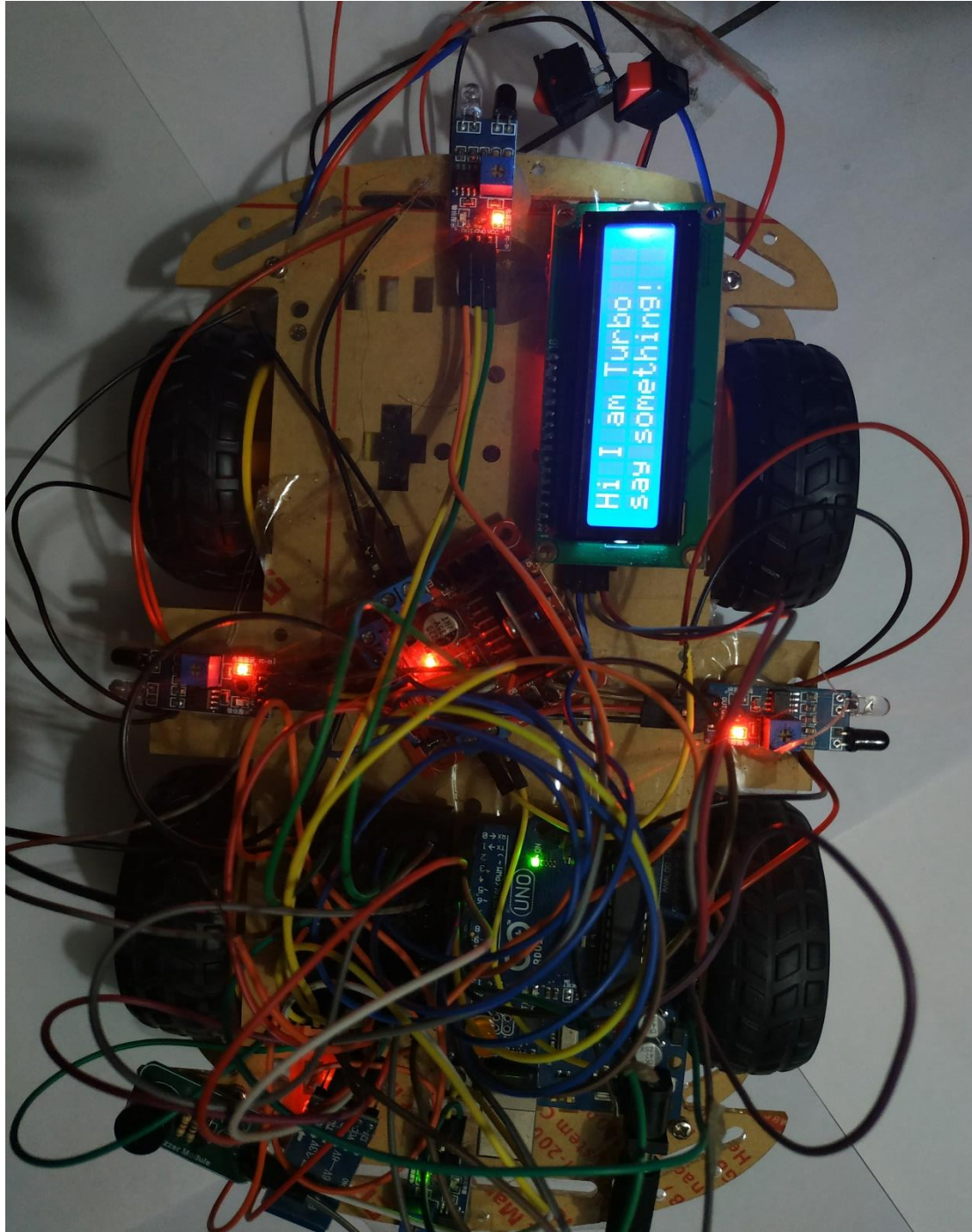


Figure 5.1: Robotic vehicle

We developed a program in our Arduino software that will run our robot. Arduino will act as the brain of the robot. On the other hand, our smartphone app will convert any natural signal to Arduino compatible signal.

In this case, the Android app will use Bluetooth module HC-05 and send the converted signal via exact serial no which is particularly set in the Arduino program.

In Android app, we used btsocket and some of activity override to take the voice command and send them to the Arduino.

Since Arduino is the brain of our robot it will decide either our robot will move or not. It will also determine the direction of the movement of the robot.

Our robot can move in four directions respectively left, right, front and back. Since it has sensors it can detect obstacles. When it detects obstacles then it won't run in that direction.

It can understand several commands like go forward, left, right, come, return, halt, dance, observe, etc.

For of its some command, it will choose the most suitable direction by itself. Like when we will say observe it will find the suitable direction.

As we said it has 4 wheels and 3 9v battery every battery is enough powerful to run its task. 9 v is for our Arduino board and others are for the motor drive which runs the dc motor, as a result, our wheel moves.

We can control the speed of the robot. We developed a different function with different delay value which controls the velocity of the car.

There is a reset command which will reset the values for the global variables.

5.2 Implementation of Android Application

As we said, we developed an android app which can perform many important tasks. It can convert and languages to text then it can send it to the Arduino to perform its particular action.

First, we have to connect a paired Bluetooth device. Then we will get our main interface. Here we can get a speak button. Then there will be a pop-up window that will take the audio and convert it to the text via google and internet.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

Our proposed system gives us the concept of using embedded system in a purpose where our daily life will be improved, easier, effective. Where we can use smartphone, Bluetooth, embedded system in a single purpose integrated.

Of Course, we could use more modern wireless technology like Wi-Fi in this type of project and on-board controller system.

6.2 Future Plan

After completing this project, we can say that it was really a difficult task. There were many challenges. There are some features we didn't add yet. Eventually we will add them. Following feature will be added in the future:

1. Grabbing hand
2. GPS
3. Use Raspberry Pie
4. Online live control
5. Wi-Fi control

References:

- [1] <https://en.wikipedia.org/wiki/Arduino>
- [2] <https://www.ijeat.org/wp-content/uploads/papers/v5i6/F4684085616.pdf>
- [3] <https://microcontrollerslab.com/voice-controlled-robot-arduino/>
- [4] "Getting Started: FOUNDATION > Introduction". *Arduino.cc*
- [5] <https://en.wikipedia.org/wiki/Arduino>
- [6] <http://www.me.umn.edu/courses/me2011/arduino/technotes/dcmotors/motor-tutorial/>

APPENDIX

APPENDIX A: ARDUINO CODE

Arduino Code Demonstration:

```
String voice;
int red = 12;
int GREEN = 3;
int BLUE = 4;

#define in1 6
#define in2 5
#define in4 4
#define in3 3
int LED = 13; // Use the onboard Uno LED
int buzzPin = 11;
int isObstaclePinFront = 7; // This is our input pin

int isObstaclePinBack = 8;
int isObstaclePinLeft = 9;
int isObstaclePinRight = 10;

int isObstacleFront = HIGH; // HIGH MEANS NO OBSTACLE
int isObstacleBack = HIGH;
int isObstacleLeft = HIGH;
int isObstacleRight = HIGH;

void go() {
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  delay(1);
```

```

digitalWrite(6, LOW);
digitalWrite(4, LOW);
digitalWrite(3, LOW);
digitalWrite(5, LOW);
}

void goSense(int m)
{
  for(int i=0;i<m;i++)
  {
    if (isObstacleFront == LOW)
    {
      Serial.println("OBSTACLE Detected!");
      digitalWrite(LED, HIGH);
      digitalWrite(buzzPin, HIGH);
      off();
    }
    else
    {
      go();
    }
  }
}

void back() {
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  delay(1);

  digitalWrite(6, LOW);
  digitalWrite(4, LOW);
  digitalWrite(3, LOW);
  digitalWrite(5, LOW);
}

```

```

/
void backSense(int m)
{
  for(int i=0;i<m;i++)
  {
    if (isObstacleBack == LOW)
    {
      Serial.println("OBSTACLE Detected!");
      digitalWrite(LED, HIGH);
      digitalWrite(buzzPin, HIGH);
      off();
    }
    else
    {
      back();
    }
  }
}

void right()
{
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(1);
  digitalWrite(6, LOW);
  digitalWrite(4, LOW);
  digitalWrite(3, LOW);
  digitalWrite(5, LOW);
}

void rightSense(int m)
{
  for(int i=0;i<m;i++)

```

```

    }
    if (isObstacleRight == LOW)
    {
        Serial.println("OBSTACLE Detected!");
        digitalWrite(LED, HIGH);
        digitalWrite(buzzPin, HIGH);
        off();
    }
    else
    {
        right();
    }
}
}
}
void left() {
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, HIGH);
    delay(1);
    digitalWrite(6, LOW);
    digitalWrite(4, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
}
void leftSense(int m)
{
    for(int i=0;i<m;i++)
    {
        if (isObstacleLeft == LOW)
        {
            Serial.println("OBSTACLE Detected!");
            digitalWrite(LED, HIGH);
            digitalWrite(buzzPin, HIGH);

```

```

        off();
    }
    else
    {
        right();
    }
}
}

void off() {
    digitalWrite(6, LOW);
    digitalWrite(4, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(LED, LOW);
    digitalWrite(red, LOW);
    digitalWrite(buzzPin, LOW);
}

void setup() {
    Serial.begin(9600);
    pinMode(buzzPin, OUTPUT);
    pinMode(LED, OUTPUT);
    //pinMode(12, OUTPUT);
    pinMode(isObstaclePinFront, INPUT);
    pinMode(isObstaclePinBack, INPUT);
    pinMode(isObstaclePinLeft, INPUT);
    pinMode(isObstaclePinRight, INPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(red, OUTPUT);
}
}

```

```

void loop() {
  isObstacleFront = digitalRead(isObstaclePinFront);
  isObstacleLeft = digitalRead(isObstaclePinLeft);
  isObstacleRight = digitalRead(isObstaclePinRight);
  isObstacleBack = digitalRead(isObstaclePinBack);
  if (isObstacleRight == LOW||isObstacleFront
==LOW||isObstacleLeft==LOW||isObstacleBack==LOW)
  {
    Serial.println("OBSTACLE Detected!");
    digitalWrite(LED, LOW);
    digitalWrite(red, HIGH);
    digitalWrite(buzzPin, HIGH);
    //off();
  }
  else
  {
    Serial.println("Freeeee!");
    digitalWrite(LED, HIGH);
    digitalWrite(red, LOW);
    digitalWrite(buzzPin, LOW);
  }

  while (Serial.available()) {
    delay(10);
    char c = Serial.read();
    if (c == '#')
    {
      break;
    }
    voice += c;
  }

  if (voice.length() > 0) {
    Serial.println(voice);
  }
}

```



```

Serial.println("clear");
digitalWrite(LED, LOW);
digitalWrite(12, LOW);

if (voice == "on" || voice == "all" || voice == "go" || voice == "forward")
{
  goSense(250);
}
else if (voice == "back" || voice == "come" || voice == "previous position")
{
  backSense(250);
}
else if (voice == "off" || voice == "stop" || voice == "halt")
{
  off();
}
else if (voice == "red" || voice=="led"|| voice=="light")
{
  //digitalWrite(5, LOW);
  digitalWrite(red, HIGH);
}
else if (voice == "red off" )
{
  digitalWrite(red, LOW);
  //digitalWrite(6, HIGH);
}
else if (voice == "buz"||voice=="make sound"|| voice=="sound" )
{
  //digitalWrite(5, LOW);
  digitalWrite(buzzPin, HIGH);
}
else if (voice == "buz off"||voice=="mute"|| voice=="quite" )

```

```

{
  //digitalWrite(5, LOW);
  digitalWrite(buzzPin, LOW);
}
else if (voice == "left" || voice == "turn left") {
  leftSense(1000);
}
else if (voice == "right" || voice == "turn right") {
  rightSense(360);
}
else if (voice == "dance" || voice == "turn around") {
  rightSense(360);
  delay(250);
  goSense(250);
  delay(250);

  leftSense(1200);
  delay(250);
  goSense(250);
  delay(250);
  rightSense(360);
  delay(250);
  backSense(250);
  delay(1000);
  leftSense(1200);
  delay(250);
  backSense(250);
}

delay(200);
voice = "";
}
}

```

APPENDIX B: ANDROID CODE

Android Code: Turbo control

```
package com.led.led;

import android.content.ActivityNotFoundException;

import android.speech.RecognizerIntent;

import android.support.v7.app.ActionBarActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.bluetooth.BluetoothSocket;

import android.content.Intent;

import android.view.View;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.SeekBar;

import android.widget.TextView;

import android.widget.Toast;

import android.app.ProgressDialog;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.os.AsyncTask;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Locale;
```

```

public class ledControl extends ActionBarActivity {

    private static final int REQ_CODE_SPEECH_INPUT = 100;

    private TextView mVoiceInputTv;

    private Button mSpeakBtn;

    Button btnOn, btnOff, btnDis;

    SeekBar brightness;

    TextView lumn;

    String address = null;

    private ProgressDialog progress;

    BluetoothAdapter myBluetooth = null;

    BluetoothSocket btSocket = null;

    private boolean isBtConnected = false;

    //SPP UUID. Look for it

    static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB");

    @Override

    protected void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        Intent newint = getIntent();

        address = newint.getStringExtra(DeviceList.EXTRA_ADDRESS);

        setContentView(R.layout.activity_led_control);

        btnOn = (Button)findViewById(R.id.button2);

        btnOff = (Button)findViewById(R.id.button3);

        btnDis = (Button)findViewById(R.id.button4);

```

```

brightness = (SeekBar)findViewById(R.id.seekBar);

lumn = (TextView)findViewById(R.id.lumn);

new ConnectBT().execute();

mVoiceInputTv = (TextView) findViewById(R.id.voiceInput);

mSpeakBtn = (Button) findViewById(R.id.btnSpeak);

mSpeakBtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        startVoiceInput();

    }

});

btnOn.setOnClickListener(new View.OnClickListener()

{

    @Override

    public void onClick(View v)

    {

        turnOnLed(); //method to turn on

    }

});

btnOff.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v)

    {

        turnOffLed(); //method to turn off

```

```

    }
});

btnDis.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Disconnect(); //close connection
    }
});

brightness.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
        if (fromUser==true)
        {
            lumn.setText(String.valueOf(progress));
            try
            {
                btSocket.getOutputStream().write(String.valueOf(progress).getBytes());
            }
            catch (IOException e)
            {
            }
        }
    }
});

```

```

    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
    }

});

}

private void Disconnect()
{
    if (btSocket!=null) //If the btSocket is busy
    {
        try
        {
            btSocket.close(); //close connection
        }
        catch (IOException e)
        { msg("Error");}
    }

    finish(); //return to the first layout
}

private void turnOffLed()
{

```

```

if (btSocket!=null)
{
    try
    {
        btSocket.getOutputStream().write("TF".toString().getBytes());
    }
    catch (IOException e)
    {
        msg("Error");
    }
}
}

private void turnOnLed()
{
    if (btSocket!=null)
    {
        try
        {
            btSocket.getOutputStream().write("TO".toString().getBytes());
        }
        catch (IOException e)
        {
            msg("Error");
        }
    }
}

```



```

    }
}
// fast way to call Toast
private void msg(String s)
{
    Toast.makeText(getApplicationContext(),s,Toast.LENGTH_LONG).show();
}
private void startVoiceInput() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Hello, How can I help you?");
    try {
        startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
    } catch (ActivityNotFoundException a) {
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case REQ_CODE_SPEECH_INPUT: {
            if (resultCode == RESULT_OK && null != data) {

```

```

        ArrayList<String> result =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

        mVoiceInputTv.setText(result.get(0));
        if (btSocket!=null)
        {
            try
            {
                btSocket.getOutputStream().write(result.toString().getBytes());
            }
            catch (IOException e)
            {
                msg("Error");
            }
        }
        break;
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_led_control, menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

private class ConnectBT extends AsyncTask<Void, Void, Void> // UI thread
{
    private boolean ConnectSuccess = true; //if it's here, it's almost connected

    @Override
    protected void onPreExecute()
    {
        //show a progress dialog
        progress = ProgressDialog.show(lcdControl.this, "Connecting...", "Please wait!!!");
    }

    @Override
    protected Void doInBackground(Void... devices) //while the progress dialog is shown,
    the connection is done in background
    {
        try

```

```

    {
        if (btSocket == null || !isBtConnected)
        {
            // get the mobile bluetooth device
            myBluetooth = BluetoothAdapter.getDefaultAdapter();
            //connects to the device's address and checks if it's available
            BluetoothDevice dispositivo = myBluetooth.getRemoteDevice(address);
            // create a RFCOMM (SPP) connection
            btSocket = dispositivo.createInsecureRfcommSocketToServiceRecord(myUUID);
            BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
            btSocket.connect();//start connection
        }
    }
    catch (IOException e)
    {
        ConnectSuccess = false;//if the try failed, you can check the exception here
    }
    return null;
}

@Override
protected void onPostExecute(Void result) //after the doInBackground, it checks if
everything went fine
{
    super.onPostExecute(result);
}

```

```
if (!ConnectSuccess)
{
    msg("Connection Failed. Is it a SPP Bluetooth? Try again.");
    finish();
}
else
{
    msg("Connected.");
    isBtConnected = true;
}
progress.dismiss();
}
}
}
```