

A PROJECT REPORT

on

“AUTO SONIC”

Submitted to
KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

VISHAL PRADYUMN

1605626

UNDER THE GUIDANCE OF
PROF. MEGHANA G RAJ



SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
April 2019

A PROJECT REPORT

on

“AUTO SONIC”

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

VISHAL PRADYUMN

1605626

UNDER THE GUIDANCE OF
PROF. MEGHANA G RAJ



SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2019

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024



CERTIFICATE

This is certify that the project entitled

“AUTO SONIC“

submitted by

VISHAL PRADYUMN

1605626

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2018-2019, under our guidance.

Date:03/07/2019

(Prof. Meghana G Raj)
Project Guide

Acknowledgement

We are profoundly grateful to Prof. MEGHANA G RAJ for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. We would like to express our gratitude towards him for his support in the field of IOT and the various sensors and micro-controllers that can serve the purpose of this project.

VISHAL PRADYUMN

ABSTRACT

The project incorporates the knowledge of IOT, i.e., Internet of Things. The primary focus has been laid on automation. The industrial world is rapidly shifting from manual labour to automated robots. This study focuses on the planning and development of an Automated Path Follower plus Collision Avoider. Looking at the increasing rates of traffic accidents in the today's scenario, there is a need to shift from manually driven vehicles to the ones that can drive themselves. As a result, here, we have laid emphasis on how a system could follow a given path without and not sway away from it. Additionally, there is a need to check for obstacles on the provided path so as to avoid collisions and damage. This would result in a decrement in the rate of accidents due to the functionalities deployed such as quick reaction, instantaneous diversion from obstacles and then reorienting the system to continue onto the predefined path. Further to take the model to a real-world scope, the model has been equipped with a bluetooth sensor. The main feature of the deployed bluetooth include turning the model on/off and linking the system with google maps so as to provide a defined path. The key features of the project have been mentioned below:

Keywords:

- Automation:

1. The system is designed to be fully automatic
2. Automation involves self path-finding and collision avoidance

- Path Follower:

1. The system is designed to follow the path that has been predefined by the user
2. The system shall only sway away from the path upon encountering obstacles
3. The system is designed to reorient itself after avoiding the obstacle.

- Collision Avoidance:

1. The system is designed to avoid collisions and further damage to it.
2. The system shall stop or try to direct its path away from the obstacles.

Contents

1	INTRODUCTION	1
1.1	COLLISION AVOIDANCE	1
1.2	PATH FINDING	2
1.3	BLUETOOTH INCORPORATION	2
1.4	FULLY DEPLOYED MODEL	2
2	LITERATURE SURVEY	4
3	SOFTWARE REQUIREMENTS SPECIFICATION	5
3.1	ARDUINO	5
3.1.1	ARDUINO UNO	5
3.1.2	ARDUINO SOFTWARE IDE	5
3.2	COLLISION AVOIDANCE	5
3.2.1	ULTRASONIC SENSOR	6
3.3	PATH FOLLOWER	6
3.3.1	IR SENSOR	7
3.4	USER INTERFACE	7
3.4.1	HC-05 SENSOR	7
3.5	L293D	8
4	REQUIREMENT ANALYSIS	9
4.1	SOFTWARE	9
4.2	COLLISION AVOIDANCE	9
4.3	LINE FOLLOWING	9
4.4	USER FRIENDLY INTERFACE	9
5	SYSTEM DESIGN	10
5.1	DATA FLOW DIAGRAM	10
6	SYSTEM TESTING	11
6.1	COLLISION AVOIDANCE TABLE	11
6.2	PATH FOLLWER TABLE	12
6.3	INTEGRATED TABLE	12

7	PROJECT PLANNING	15
7.1	CONCEPT AND INTIATION	15
7.2	DEFINITION AND PLANNING	15
7.3	BACK END	15
7.4	FRONT END	15
7.5	MODEL TESTING	16
7.6	PERFORMANCE AND CONTROL	16
8	IMPLEMENTATION	17
8.1	ARDUINO	17
8.2	ANDROID	21
9	SCREENSHOTS OF PROJECT	24
9.1	SENSORS	24
9.2	BOARDS	26
10	CONCLUSION AND FUTURE SCOPE	27
10.1	CONCLUSION	27
10.2	FUTURE SCOPE	27

List of Figures

1.1	SIDE VIEW	3
1.2	FRONT VIEW	3
9.1	IR SENSOR	24
9.2	ULTRASONIC SENSOR	25
9.3	HC-05	25
9.4	ARDUINO UNO BOARD	26
9.5	BREAD BOARD	26

Chapter 1

Introduction

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensor to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an IP address and is able to transfer data over a network.

Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, better understand customers to deliver enhanced customer service, improve decision-making and increase the value of the business.

The deployed model makes use of two basic sensing devices which include an ultrasonic sensor and an IR sensor. The basic function of these sensor are collision avoidance and path following respectively.

We shall further discuss about the deployed sensors in the following sections.

1.1 COLLISION AVOIDANCE

A **collision avoidance system**, also known as a **precrash system**, **forward collision warning system**, or **collision mitigating system**, is an automobile safety system designed to prevent or reduce the severity of a collision. It uses a sonar and sometimes laser (LIDAR) and camera (employing image recognition) to detect an imminent crash. GPS sensors can detect fixed dangers such as approaching stop signs through a location database.

Once an impending collision is detected, these systems provide a warning to the driver. When the collision becomes imminent, they take action autonomously without any driver input (by braking or steering or both).

Collision avoidance by braking is appropriate at low vehicle speeds , while collision avoidance by steering may be more appropriate at higher vehicle speeds if lanes are clear. Cars with collision avoidance may also be equipped with adaptive cruise control, using the same forward-looking sensors.

By taking into consideration of small scale deployment regarding our project development, ultrasonic sensor is employed for the detection of potent collisions in the pathway of the system.

1.2 PATH FINDING

Line follower is an autonomous robot which follows either black line in white area or white line in black area. Robot must be able to detect particular line and keep following it. For special situations such as cross overs where robot can have more than one path which can be followed, predefined path must be followed by the robot.

The basic purpose of deploying a line follower is to demonstrate that the model shall follow a specified path (from start to destination)The future scope of a line follower is quite varied. However, looking at the structural size and deployed scope of the model , it has been equipped to follow a specifically designed path and an IR sensor to detect the path.

1.3 BLUETOOTH INCORPORATION

To simply turn the device on and off with physically touching it so as to provide an interface to the user, the model provides support for bluetooth connectivity.

An android application has been further developed to provide with the interface to the user. To incorporate bluetooth with the model, the sensor used is HC05.

1.4 FULLY DEPLOYED MODEL

Ultimately, the model is deployed by combining the functionalities of both a 'path finder' and a 'collision avoidance system'. The basic motto behind the model was to create a system that is able to trace a predefined path ,avoid any collisions present on that path, and then continue onto the same to reach the destination.

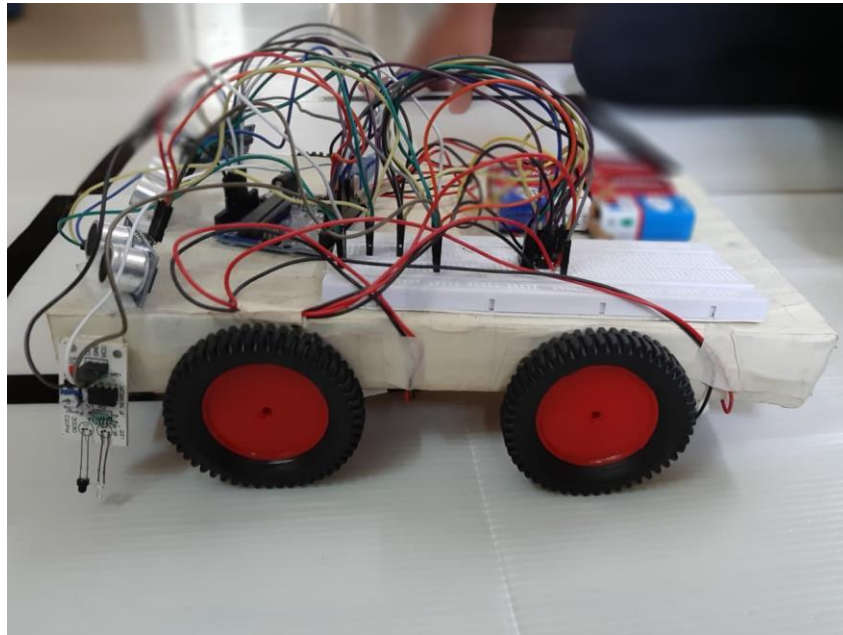


Figure 1 . 1 : SIDE VIEW

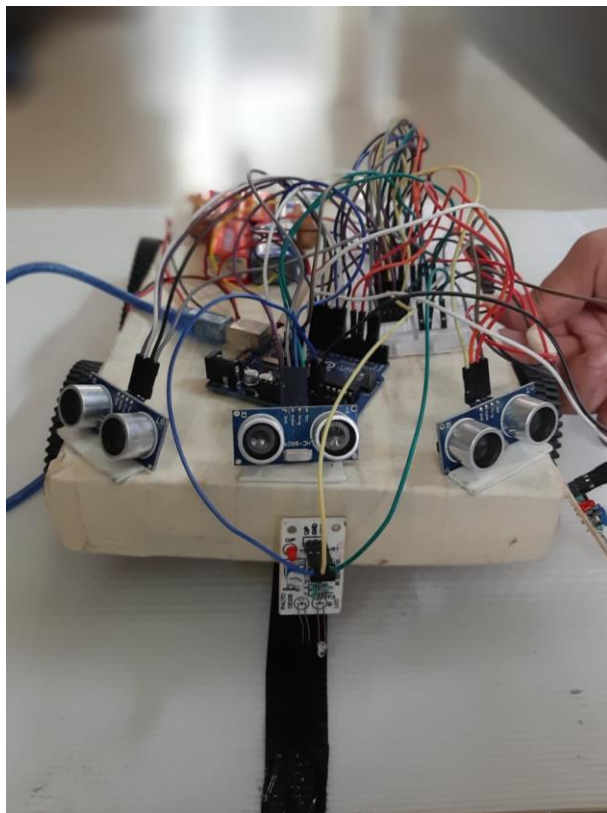


Figure 1 . 2 : FRONT VIEW

Chapter 2

Literature Survey

An **autonomous car**, also known as a **robotic car**, **self-driving car**, or **driver-less car**, is a vehicle that is capable of sensing its environment and moving with little or no human input. Autonomous cars use various kinds of technologies. They can be built with GPS sensing knowledge to help with navigation. They may use sensors and other equipment to avoid collisions.

Taking the scope of the deployed model into consideration we have considered incorporating the vehicle with an IR, an ultrasonic sensor and a bluetooth sensor. To incorporate the above mentioned sensor with the model and to provide intelligence to the system, we have considered deploying the project with IOT implementation, i.e. , by using Arduino IDE and an arduino board for the same.

Further to provide an interface to the user, we have deployed a simple user interface by creating a handheld mobile application using Android Studio.

Chapter 3

Software Requirements Specification

To create any software, multiple resources and varied knowledge is required. In the same fashion, our model poses some requirements for governing different practical problems and deploying corresponding solutions.

3.1 ARDUINO

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control both physically and digitally.

3.1.1 ARDUINO UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It only needs to be connected to a USB cable or a AC-to-DC power adapter or battery to get started.

3.1.2 ARDUINO SOFTWARE IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software is designed to be used with any Arduino board. The IDE has been used to functionalities to all the components connected with the Arduino board.

The IDE Software is required to read the values returning from the sensors and provide corresponding response. It basically acts as an interface between the incoming values and the guided responses.

3.2 COLLISION AVOIDANCE

The primary focus of the model is to detect the presence of obstacles in the pathway and to avoid them in order to prevent any sort of damage.

Once an impending collision is detected, these systems provide a warning to the central intelligence system. When the collision becomes imminent, they take action autonomously without any driver input. For the purpose of collision avoidance, keeping in mind the scope of the model, the model has been equipped with **ultrasonic sensors**

3.2.1 ULTRASONIC SENSOR

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1 foot to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.

This ultrasonic distance sensor provides steady and **accurate distance measurements** within the range of 2cm to a whopping 450cm. It has a focus of less than 15 degrees and an accuracy of about 2mm.

The Ultrasonic sound waves has an extremely high pitch that humans cannot hear and is also free from external noises from passive or active sources. This particular sensor transmits an ultrasonic sound that has a frequency of about 40 kHz.

The purpose of ultrasonic sensor is to detect the presence of obstacles and provide the accurate measure of distance between the vehicle and obstacle to the software so as to deploy immediate responses and actions.

3.3 PATH FOLLOWER

A Path Follower or Line Follower is basically a robot which follows a particular path or trajectory and decides its own course of action which interacts with obstacle. The path can be a black line on the white floor (visible) or a magnetic field (invisible).

The use of line following vehicle is transport the materials from one place to another place in the industries. This robot movement completely depends on the track. For the purpose of line following, the sensor deployed is IR sensor.

3.3.1 IR SENSOR

This is a LED that emits an IR (infrared) Light. The Infrared Emitter is used to transmit infrared signals through an infrared LED, while there is an **Infrared receiver** to get the signals on the other side.

The IR Emitter emits infrared light, means it emits light in the range of Infrared frequency. The wavelength of Infrared (700nm – 1mm) is just beyond the normal visible light. Infrared have the same properties as visible light, like it can be focused, reflected and polarized like visible light.

Making use of reflectance property of IR radiations, the IR Receiver receives the the reflected IR rays and sends the information from an infrared remote control to the arduino uno for decoding signals and performing required actions.

3.4 USER INTERFACE

To provide the user with remote application and help them with operating the device without needing to learn about the internal connections, a simple mobile application has been created to present the UI.

The UI has been created with Android Studio software. To link the user interface with the model, we have exploited the bluetooth functionality for remote connection.

In order to connect the UI to the model via bluetooth, an additional hardware is needed which is the HC-05 sensor.

3.4.1 HC-05 SENSOR

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm.

The HC-05 sensor shall receive the data(such as 1 or 0) sent from the mobile application to perform required actions that include turning the model on and off(respectively) and other data such as GPS data to provide the path that it should follow.

3.5 L293D

The L293D is a quadruple high-current half-H drivers. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. It designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.

The L293D is used to define 'High' and 'Low' voltage values for motors so as to impart rotatory action to the model. This is the core of the model as the basic functions of line tracing and collision avoidance cannot be completed without it since rotation is required in either of them.

Chapter 4

Requirement Analysis

In the course of development of this project, varied types of requirements have been gathered in order to map the varied needs and specifications required.

Further the gathered requirements have been decomposed into several phases for further analysis.

4.1 SOFTWARE

The first and foremost requirement is to research and select a software compatible and economical enough to incorporate the functionalities of the mentioned sensors. The software included for this project is Arduino IDE since it is an open source software and provides appropriate support in the form of a micro-controller with the help of Arduino Uno.

4.2 COLLISION AVOIDANCE

To serve for the property of avoiding collisions, the model first needs to detect the proximity of the nearby objects so as to take required actions. This feature has been served by deploying ultrasonic sensors. There is a need to detect the proximity on three sides namely the front, left and right side of the vehicle for the purpose of determining the direction the vehicle should turn to. Hence, three ultrasonic sensors have been attached at practically tested angles at the front of the model.

4.3 LINE FOLLOWING

Without a specified path, the model serves no meaning. Hence, the model is designed to trace a predefined path. Here again, three IR sensors are deployed at the front side of the vehicle which shall read the position of the path, hence orienting the vehicle, and determine the direction that the vehicle should follow.

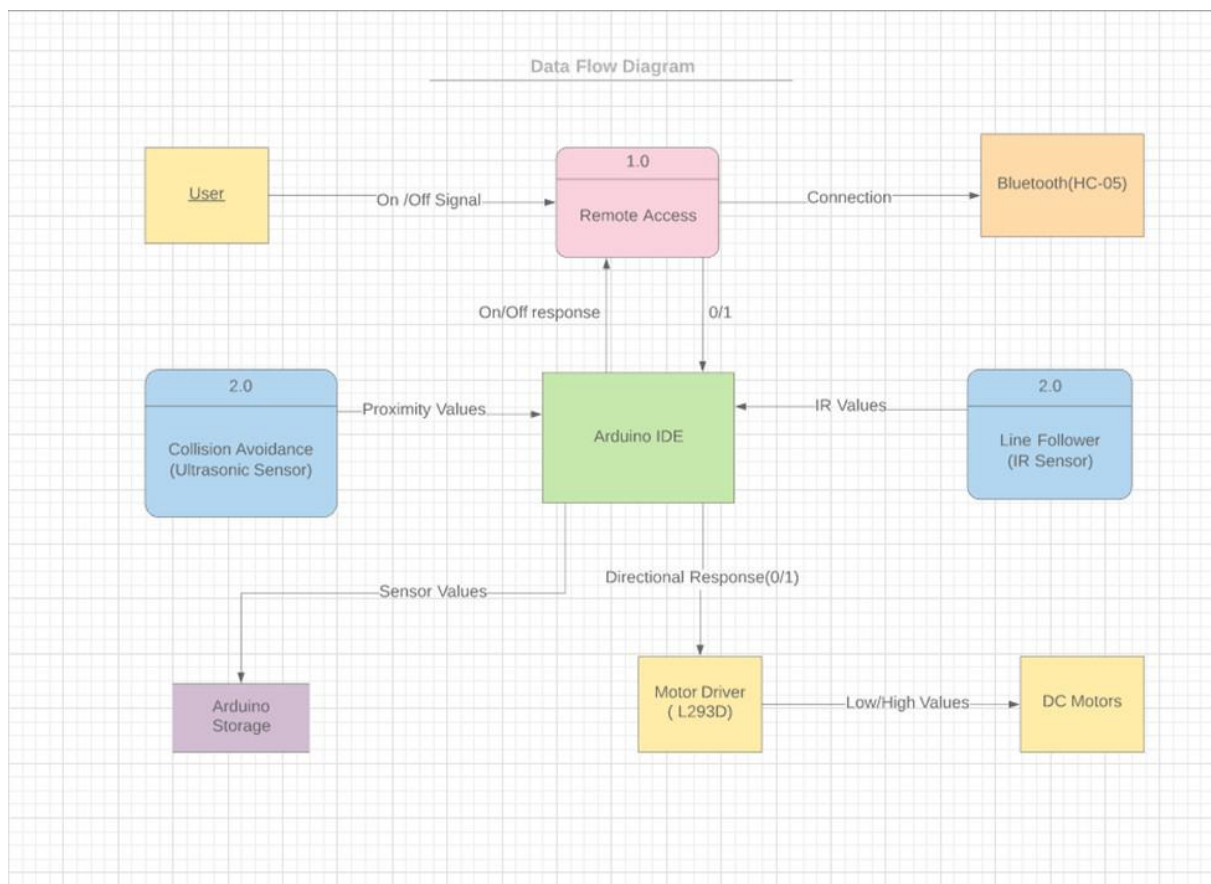
4.4 USER FRIENDLY INTERFACE

To provide the user with a remote control and support the feature of authorization and security, an android based user interface has been developed.

Chapter 5

System Design

5.1 DATA FLOW DIAGRAM



Chapter 6

System Testing

The model testing has been performed in three phases including the integration testing. The details of each phase has been mentioned In the below corresponding tables

6.1 Collision Avoidance Table

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	F-Obstacle	Obstacle in front	Turns left	Turn left
T02	F,L-Obstacle	Obstacle in front and left	Turns right	Turn right
T03	F,R-Obstacle	Obstacle in front and right	Turns left	Turn left
T04	F,L,R-Obstacle	Obstacle in front, left and right	Moves backwards until a there is at least one side free(without obstacle).Turns towards the free side.	Move backwards until a there is at least one side free(without obstacle).Turn towards the free side.

6.2 Path Follower Table

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	F-Path	Forward path	Follows path	Follow path
T02	L-Path	Left Path and no Forward path	Turns left	Turn left
T03	R-Path	Right Path and no Forward path	Turns right	Turn right
T04	F,L-Path	Forward Path and Left Path available	Moves forward	Move Forward
T05	F,R-Path	Forward Path and Right Path available	Moves forward	Move Forward
T06	L,R-Path	Left Path and Right Path available	Turns left	Turn Left
T07	F,L,R-Path	Forward Path, Left Path and Right Path available	Moves forward	Move Forward

6.3 Integrated Table

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	F-Path	No Obstacle in front	Moves Forward	Move Forward
T02	L-Path	No Forward path or obstacle in Forward path and Left path available	Turns left	Turn left
T03	R-Path	No Forward path or obstacle in Forward path and Right path available	Turns right	Turn right
T04	F,L-Path	Forward Path and Left Path available and no obstacle in Forward path	Moves forward	Move Forward

T05	F,R-Path	Forward Path and Right Path available and no obstacle in Forward path	Moves forward	Move Forward
T06	L,R-Path	Left Path and Right Path available and no obstacle in Left path	Turns left	Turn Left
T07	F,L,R-Path	Forward Path, Left Path and Right Path available and no obstacle in Forward path	Moves forward	Move Forward
T08	L,F-Path	Forward Path and Left Path available and obstacle in Forward path	Turns Left	Turn Left
T09	R,F-Path	Forward Path and Right Path available and obstacle in Forward path	Turns Right	Turn Right
T10	R,L-Path	Left Path and Right Path available and obstacle in Left path	Turns Right	Turn Right
T11	L,F,R-Path	Forward Path, Left Path and Right Path available and obstacle in Forward path	Turns Left	Turn Left

T12	R,L,F-Path	Forward Path, Left Path and Right Path available and obstacles in Forward and Left path	Turns Right	Turn Right
T13	No Path	Obstacle in Forward Path and no Alternate Path available/ Obstacle in all Alternate Paths	Moves backwards until a there is at least one path free(without obstacle).Turns towards the free path.	Move backwards until a there is at least one path free(without obstacle).Turns towards the free path.

References: F - Front , L - Left, R- Right

Chapter 7

Project Planning

7.1 CONCEPT AND INITIATION

In the transition to modern world, more and more of the tasks are being handled autonomously by robots so as to increase the productivity, efficiency, economic growth and durability. The basic concept behind the project being the need of automation both in the commercial and the industrial sector. For the primary foundation of the project, we have developed a vehicular model that is able to drive by itself and has minimum human interaction. Further the project can expand to higher levels so as to satisfy the need of the ever growing technical world.

7.2 DEFINITION AND PLANNING

The model deployed needs to serve three basic features which are collision avoidance, line following and providing the user with a control interface. Further there is a need to test the model before final deployment so as to ensure that the model satisfies the requirement specifications.

The development of this project has been achieved by initially dividing the task into smaller sections thus separating them into:

1. Back-end Coding
2. Front-end Coding
3. Model Testing

7.3 BACK END

The back-end involves Arduino based coding. The purpose of back-end is to provide intelligence to the system. The code has been developed and made error-free keeping in mind the varied test cases proposed during model testing.

7.4 FRONT END

The task of creating the user interface has been performed by the Front-end developer. The front-end platform is an android platform and the software used for creating the same is Android Studio.

7.5 MODEL TESTING

The model has been tested over several test case. The weekly analysis included proposing new test cases and verifying whether the model behaved as it was supposed to. The main role of model testing was to look for errors and propose ways to resolve them, thus forcing another iteration of the model development process. Apart from the proposing the test cases, the model was also tested for mechanical structural integrity and was reassembled in necessary situations.

7.6 PERFORMANCE AND CONTROL

During the course of development of the project, numerous mechanical and technical errors were encountered which had to be resolved in a controlled manner. The mechanical errors ranging from selection of an appropriate base to support the weight of all the equipments included, to the various connections required. Further appropriate supply of power has to be decided that shall not harm the sensors and the chips involved in the project.

Moreover a specific motor driver has been included to control the rotation of the motors so as to provide the model with the ability to change it's direction. The motor driver has been selected to be L293D .

For the purpose of proximity detection and line sensing ultrasonic sensors and IR sensors have been deployed respectively. The project has been tested on a weekly basis for a month by designing new and varied test cases. Further amendments have been made to the back-end code in case of errors.

Lastly the project has been iterated so as to meet the mentioned performance requirements.

Chapter 8

Implementation

8.1 Arduino

```

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help

integrated_project_code

const int trigL = 5;
const int trigR = 11;
const int trigF = 13;
const int echoL = 6;
const int echoR = 10;
const int echoF = 12;
const int in1 = 9;
const int in2 = 8;
const int in3 = 2;
const int in4 = 3;

void setup() {
  // put your setup code here, to run once:
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
  pinMode(trigL, OUTPUT);
  pinMode(trigR, OUTPUT);
  pinMode(trigF, OUTPUT);
  pinMode(echoL, INPUT);
  pinMode(echoR, INPUT);
  pinMode(echoF, INPUT);
}
    
```

Done Saving.
The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long.

47 | Arduino/Genuino Uno on COM12 | 10:38 03-07-2019

```

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help

integrated_project_code

}

void loop() {
  // put your main code here, to run repeatedly:
  int i=analogRead(A0);
  int j=analogRead(A1);
  int k=analogRead(A2)

  digitalWrite(trigF, LOW);
  delayMicroseconds(2);
  digitalWrite(trigF, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigF, LOW);
  durationF = pulseIn(echoF, HIGH);
  distanceF = durationF/58.2;

  digitalWrite(trigR, LOW);
  delayMicroseconds(2);
  digitalWrite(trigR, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigR, LOW);
  durationR = pulseIn(echoR, HIGH);
  distanceR = durationR/58.2;

  digitalWrite(trigL, LOW);
  delayMicroseconds(2);
}
    
```

Done Saving.
The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long.

47 | Arduino/Genuino Uno on COM12 | 10:39 03-07-2019

```

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
digitalWrite(trigL,HIGH);
delayMicroseconds(10);
digitalWrite(trigL,LOW);
durationL = pulseIn(echoL,HIGH);
distanceL = durationL/58.2;
//i is forward
//j is left
//if forward and left black prefer forward

int bcheck=0;
if(((j==100)|| (distanceL < 30)) && ((k==100)|| (distanceR < 30)) && ((i==100)|| (distanceF < 30)))
{
    while(((distanceL<30)|| (j==100)) && ((distanceR<30)|| (k==100))){

        digitalWrite(in1,LOW);
        digitalWrite(in2,HIGH);
        digitalWrite(in3,LOW);
        digitalWrite(in4,HIGH);
        delay(500);

        digitalWrite(trigL,LOW);
        delayMicroseconds(2);
        digitalWrite(trigR,HIGH);
        delayMicroseconds(10);
        digitalWrite(trigL,LOW);
        durationL = pulseIn(echoL,HIGH);
    }
}

```

Done Saving.

The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long.

47 Arduino/Genuino Uno on COM12 10:39 03-07-2019

```

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
durationL = pulseIn(echoL,HIGH);
distanceL = durationL/58.2;

digitalWrite(trigR,LOW);
delayMicroseconds(2);
digitalWrite(trigR,HIGH);
delayMicroseconds(10);
digitalWrite(trigR,LOW);
durationR = pulseIn(echoR,HIGH);
distanceR = durationR/58.2;

}
bcheck=1;
if((j==0)&&(distanceL>30) && (bcheck==1))
{
    //turn left
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    delay(600);
}
else if((k==0)&&(distanceR>30) && (bcheck==1))
{
    //turn right
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
}
}

```

Done Saving.

The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers and be less than 64 characters long.

47 Arduino/Genuino Uno on COM12 10:39 03-07-2019


```

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help

integrated_project_code
    delay(600);
  }
  else if(i==0 && j==0 && k==0 && distanceL>30 && distanceF>30 && distanceR>30)
  {
    //move forward
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    delay(1200);
  }
  else if(i==0 && (j>=100 || distanceL<15) && k==0 && distanceF>30 && distanceR>30)
  {
    //move forward
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    delay(1200);
  }
  else if(i==0 && j==100 && distanceF>30 && distanceL>30)
  {
    //move forward
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
  }
}

Done Saving.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.

47
Arduino/Genuino Uno on COM12
10:39
03-07-2019

```

```

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help

integrated_project_code
    else if(i>=100 && j=0 && k>=100 && distanceL>30)
    {
      //turn left
      digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,HIGH);
      delay(600);
    }
    else if(i>=100 && j>=100 && k=0 && distanceR>30)
    {
      //turn right
      digitalWrite(in1,LOW);
      digitalWrite(in2,HIGH);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      delay(600);
    }
    else{
      //move forward
      digitalWrite(in1,HIGH);
      digitalWrite(in2,LOW);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      delay(1200);
    }
  }
}

Done Saving.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers and be less than 64 characters long.

47
Arduino/Genuino Uno on COM12
10:39
03-07-2019

```

8.2 Android

```

package def.abc.smartcar;

import ...

public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {
    int REQUEST_ENABLE_BT;
    ListView lv;
    Button b;
    BluetoothAdapter mBluetoothAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lv = findViewById(R.id.listView);
        b = findViewById(R.id.button);
        mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        if (mBluetoothAdapter == null) {
            Toast.makeText(context, this, text: "Bluetooth not supported", Toast.LENGTH_LONG).show();
        }
        else {
            if (!mBluetoothAdapter.isEnabled()) {
                Intent enableBluetooth = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBluetooth, REQUEST_ENABLE_BT);
            }
            else {
                paired();
            }
        }
        b.setOnClickListener((v) -> {
            Intent i = new Intent(Settings.ACTION_BLUETOOTH_SETTINGS);
    
```

```

Intent i = new Intent(Settings.ACTION_BLUETOOTH_SETTINGS);
startActivity(i);
});
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK) {
        Toast.makeText(context, this, text: "Bluetooth Enabled", Toast.LENGTH_LONG).show();
        paired();
    }
    else {
        Toast.makeText(context, this, text: "Cancelled", Toast.LENGTH_LONG).show();
    }
}

public void paired() {
    ArrayList al = new ArrayList();
    mBluetoothAdapter.startDiscovery();
    Set<BluetoothDevice> pairedDevice = mBluetoothAdapter.getBondedDevices();
    if (pairedDevice.size() > 0) {
        for (BluetoothDevice bluetoothDevice : pairedDevice) {
            String name = bluetoothDevice.getName();
            String mac = bluetoothDevice.getAddress();
            al.add(name + "\n" + mac);
        }
        ArrayAdapter adapter = new ArrayAdapter(context, this, android.R.layout.simple_list_item_1, al);
        lv.setAdapter(adapter);
        lv.setOnItemClickListener(this);
    }
}
    
```

```

1 package def.abc.smartcar;
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 public class connected extends AppCompatActivity {
24
25     String address;
26     Button b1,b2,b3,b4,b5,b6,send;
27     EditText et;
28     private ProgressDialog progress;
29     BluetoothAdapter myBluetooth = null;
30     BluetoothSocket btSocket = null;
31     private boolean isBTConnected = false;
32     static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
33
34
35     boolean stopThread;
36     byte buffer[];
37     private InputStream inputStream;
38     TextView textView;
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_connected);
44         getSupportActionBar().hide();
45         Intent i = getIntent();
46         address = i.getStringExtra( name: "UUID");
47         Toast.makeText( context: this, text: ""+address, Toast.LENGTH_SHORT).show();
48         new ConnectBT().execute();
49         b1=findViewById(R.id.button1);
50         b2=findViewById(R.id.button2);
51     }
52 }

```

```

52 b5=findViewById(R.id.button5);
53 b6=findViewById(R.id.button6);
54 send=findViewById(R.id.send);
55 et=findViewById(R.id.editText);
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104 private void sendSignal ( String number ) {
105     if ( btSocket != null ) {
106         try {
107             btSocket.getOutputStream().write(number.getBytes());
108         } catch (IOException e) {
109             msg("Error");
110         }
111     }
112 }
113
114 private void Disconnect () {
115     if ( btSocket!=null ) {
116         try {
117             btSocket.close();
118         }
119     }
120 }
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
```

```

180
181     final Handler handler = new Handler();
182     stopThread = false;
183     buffer = new byte[1024];
184     Thread thread = new Thread((Runnable) () -> {
185         while(!Thread.currentThread().isInterrupted() && !stopThread)
186         {
187             try
188             {
189                 int byteCount = inputStream.available();
190                 if(byteCount > 0)
191                 {
192                     byte[] rawBytes = new byte[byteCount];
193                     inputStream.read(rawBytes);
194                     final String string=new String(rawBytes, charsetName: "UTF-8");
195                     Log.i("A", string);
196                     handler.post(() -> {
197                         textView.append(string);
198                     });
199                 }
200             }
201             catch (IOException ex)
202             {
203                 stopThread = true;
204             }
205         }
206     });
207     thread.start();
208 }
209
210
211
212
213
214
215
216
217
    
```

connected onCreate()

Gradle sync finished in 4s 187ms (from cached state) (17 minutes ago) 98:12 CRLF UTF-8 Context: <no context>

10:58 03-07-2019

Chapter 9

Screenshots of Project

9.1 SENSORS



Figure 9. 1: IR SENSOR

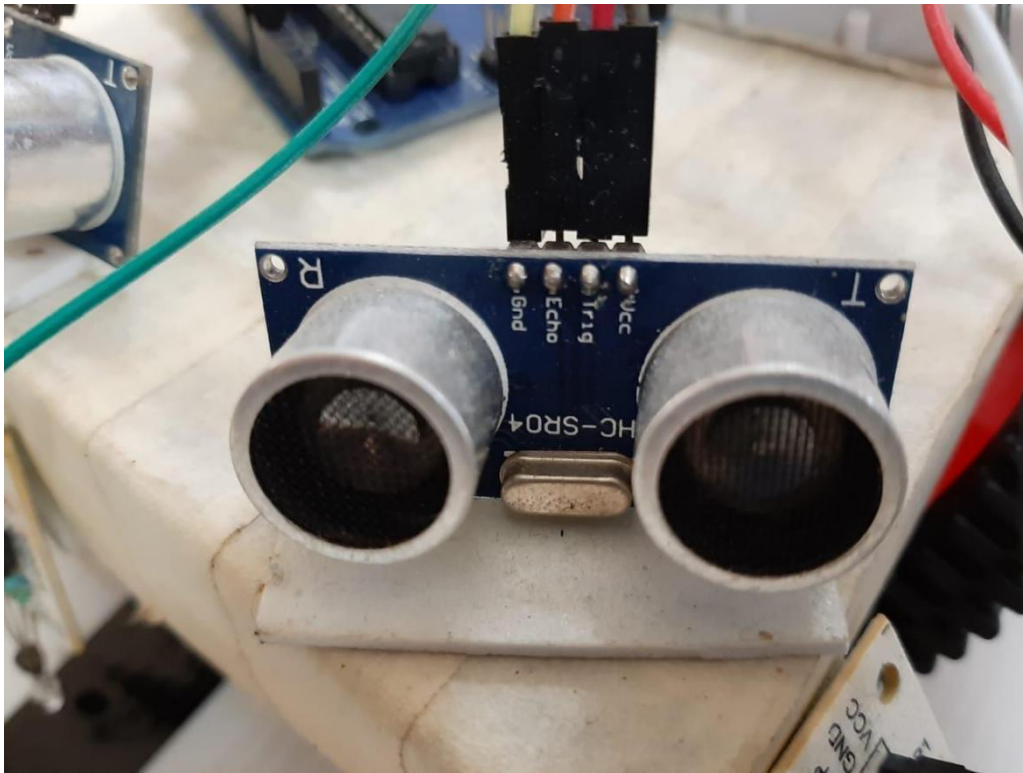


Figure 9 . 2 : ULTRASONIC SENSOR



Figure 9 . 3 : HC-05

9.2 BOARDS



Figure 9 . 4 : ARDUINO UNO BOARD

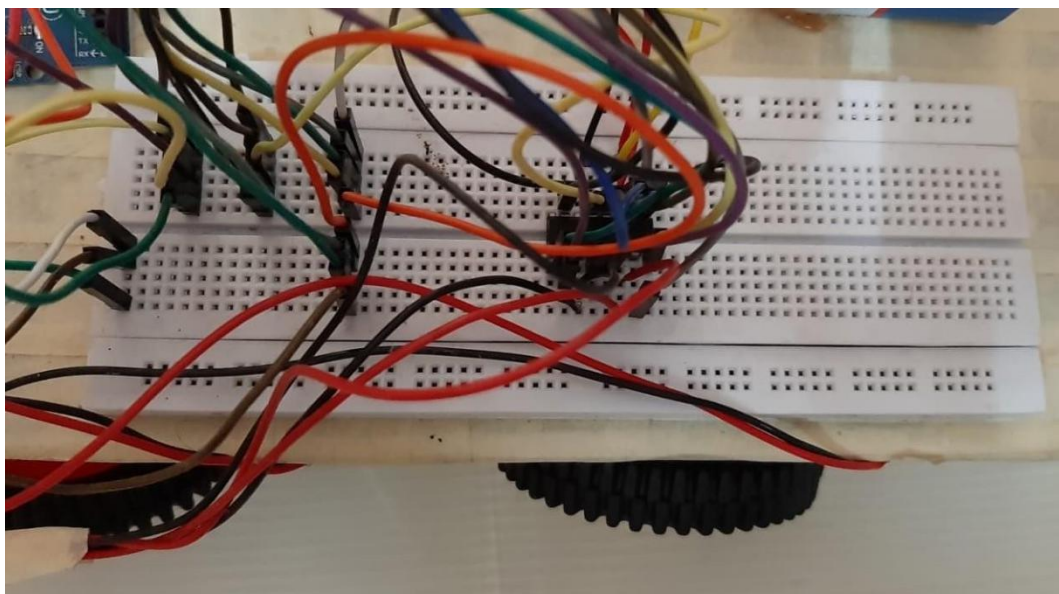


Figure 9 . 5 : BREAD BOARD

Chapter 10

Conclusion and Future Scope

10.1 Conclusion

In the due course of development of a system that can autonomously trace a line/path and avoid any obstacles , multiple sensors have been deployed and the working of model has been tested for various detailed cases so as to satisfy the mentioned requirements. Looking at the scope and economic prospect, the model has been equipped with IR sensors and ultrasonic sensors. Further to provide a simple interface to the user,an android application has been developed which can be linked to the system via bluetooth using the HC-05 sensor.

10.2 Future Scope

The future prospect of the developed system covers both large-scale and small-scale applications as mentioned below:

10.2.1 Incorporation inside on-road vehicles

10.2.2 Incorporation inside small autonomous house-cleaning robots

10.2.3 Autonomous food serving/document distribution in restaurants/offices

10.2.4 Self-driving wheelchairs and baby carts

10.2.5 Autonomous tourism guides

10.2.6 Self rearranging chairs in offices

References

- 1) www.google.com
- 2) www.wikipedia.com
- 3) www.developer.android.com
- 4) www.arduino.cc

Appendix-I

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF STUDENT VISHAL PRADYUMN

ROLL NO 1605626

PROJECT TITLE AUTO SONIC

ABSTRACT OF THE PROJECT Before the coding part could begin, there was a need to define the design of the project. This has been provided with the help of a simple yet descriptive data flow diagram. Additionally, the different tasks have been divided to each individual member. Post coding, there was a need to locate the missing functionalities and potential errors. This has been resolved by testing the model on various test cases and verifying whether the model's behaviour matches the expected behaviour.

CONTRIBUTION

CONTRIBUTION TO THE PROJECT REPORT System Design, System testing, Project Planning

CONTRIBUTION DURING IMPLEMENTATION Designing of test cases

CONTRIBUTION FOR THE PROJECT DEMONSTRATION / PRESENTATION Introduction, requirements and conclusion

SIGNATURE OF GUIDE

SIGNATURE OF STUDENT