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 results in 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 stops or try to direct its path away from the obstacles.

Contents

List of Figures

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 deployement 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 are 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 the 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.

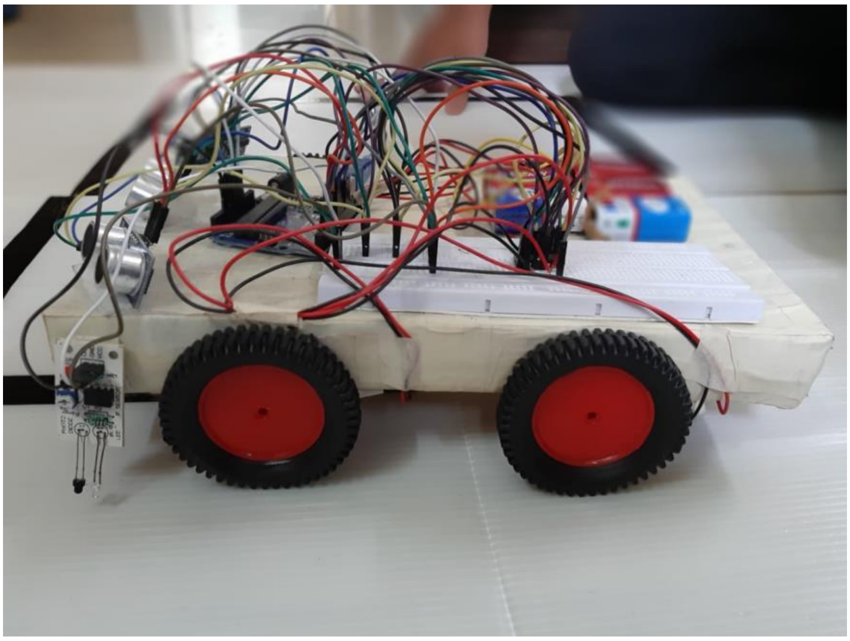


Figure 1 . 1 : S I D E V I E W

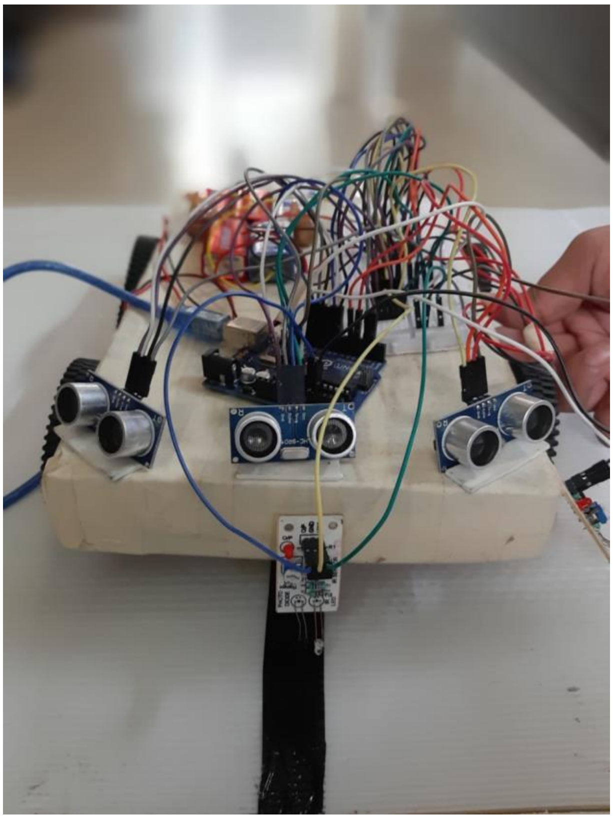


Figure 1 . 2 : FRONT VIEW

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 feet 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 whooping 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.

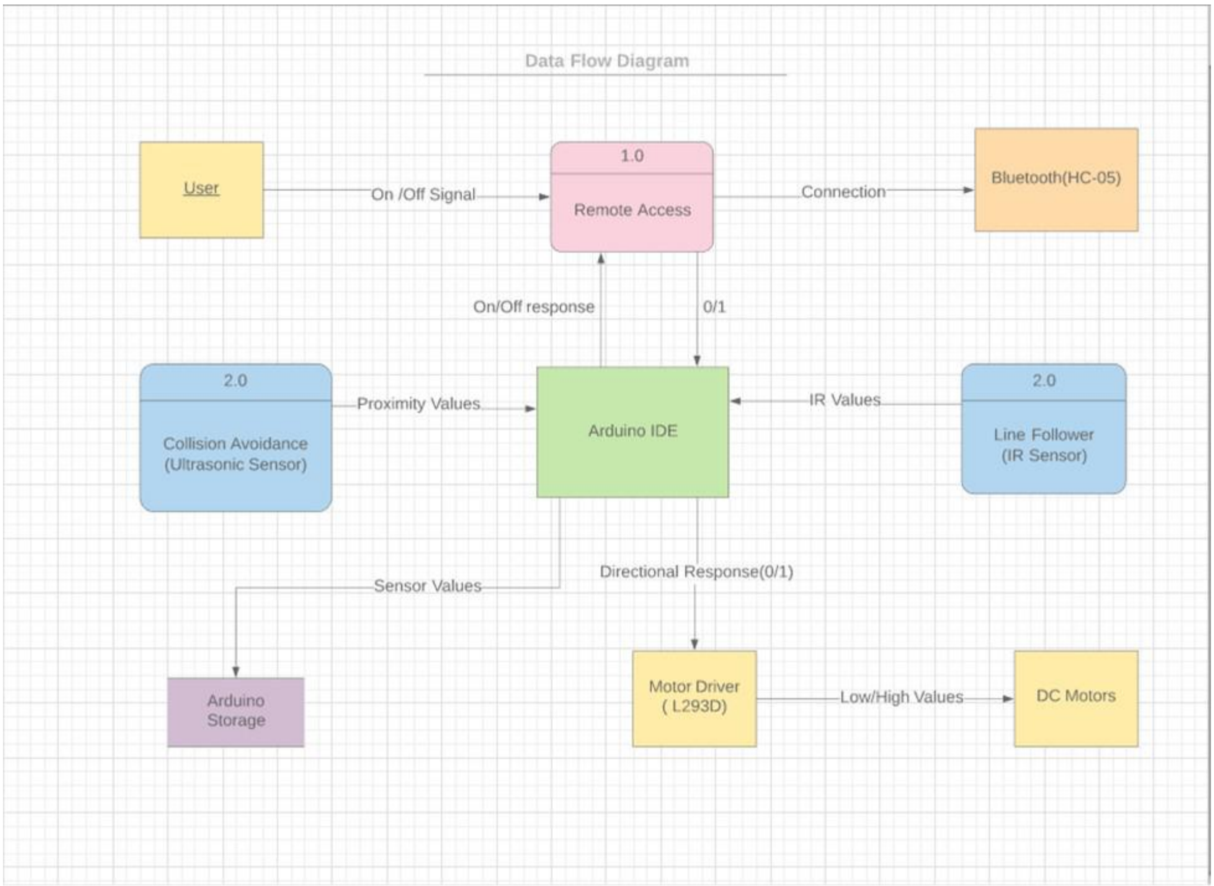
4.4 USER FRIENDLY INTERFACEE

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 .

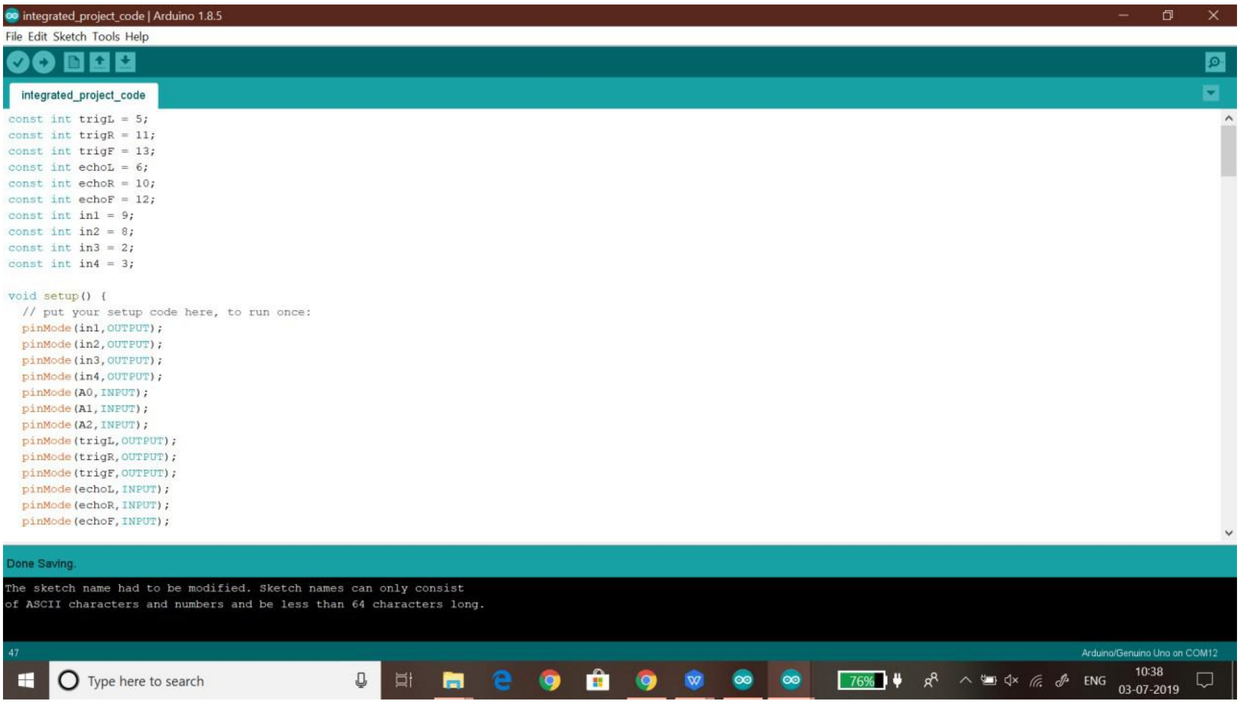
For the purpose of proximity detection and line sensing ultrasonic sensors and IR sensors have been deployed respectively. The project has been tested on a weekly basis for a month by designing new and varied test cases. Further amendments have been made to the back-end code in case of errors.

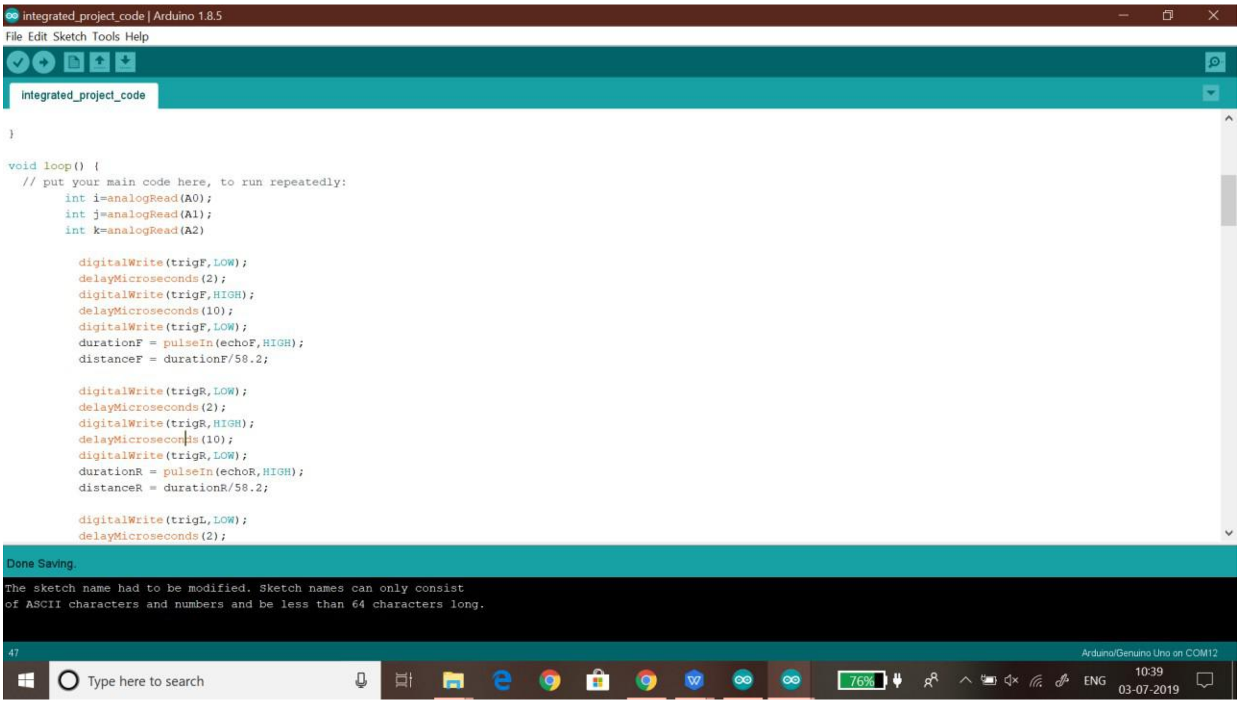
Lastly the project has been iterated so as to meet the mentioned performance requirements.

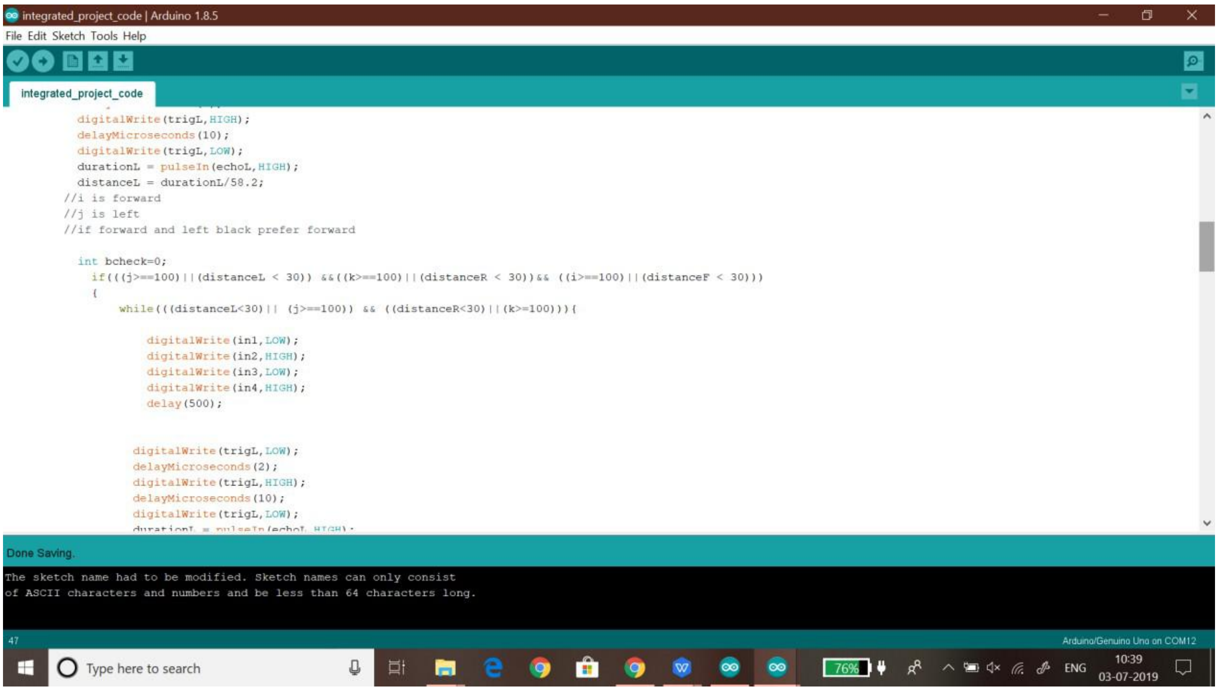
Chapter 8

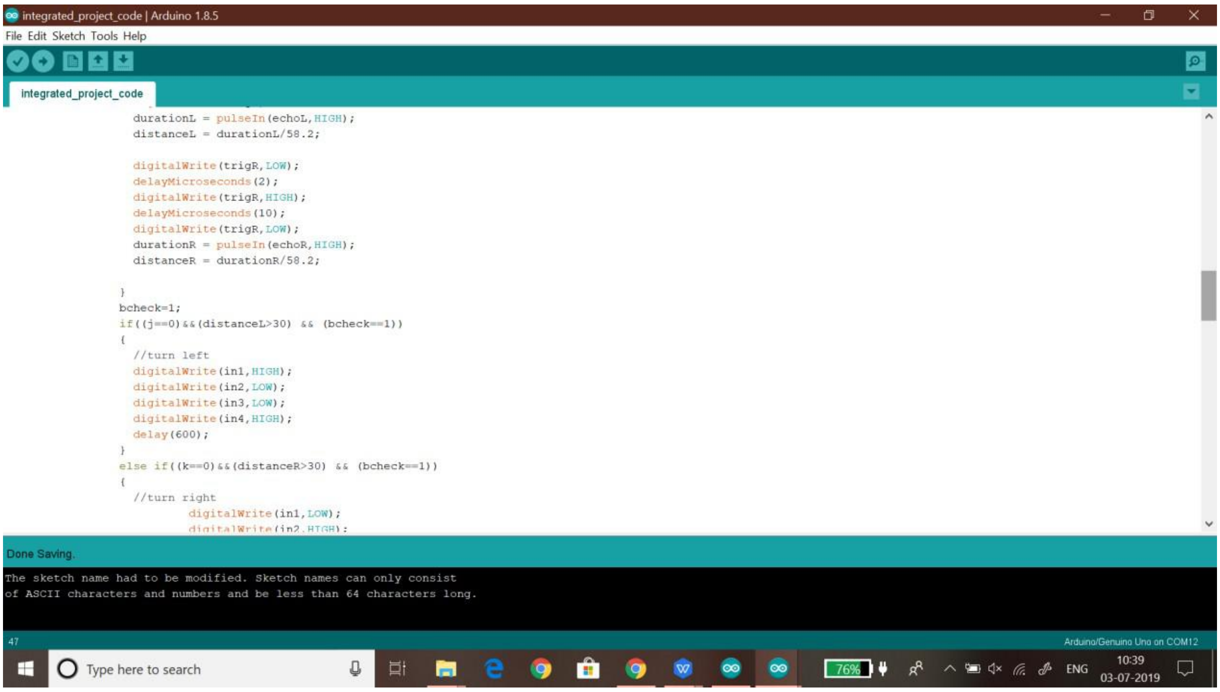
Implementation

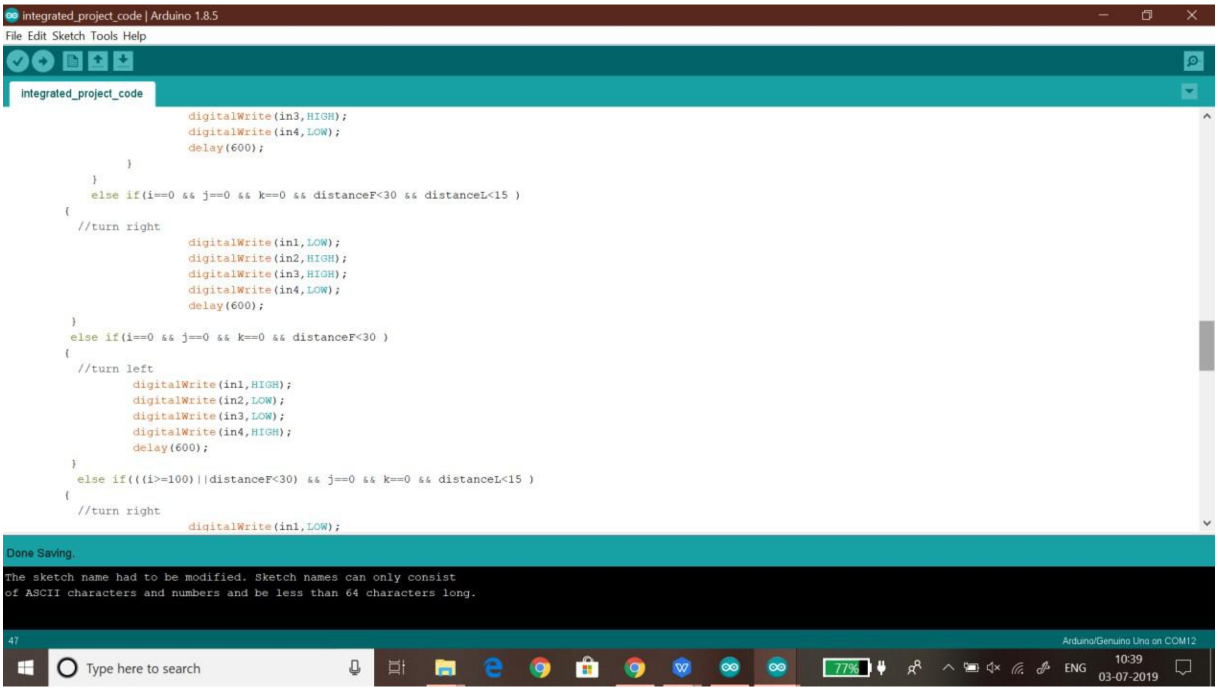
8.1 Arduino

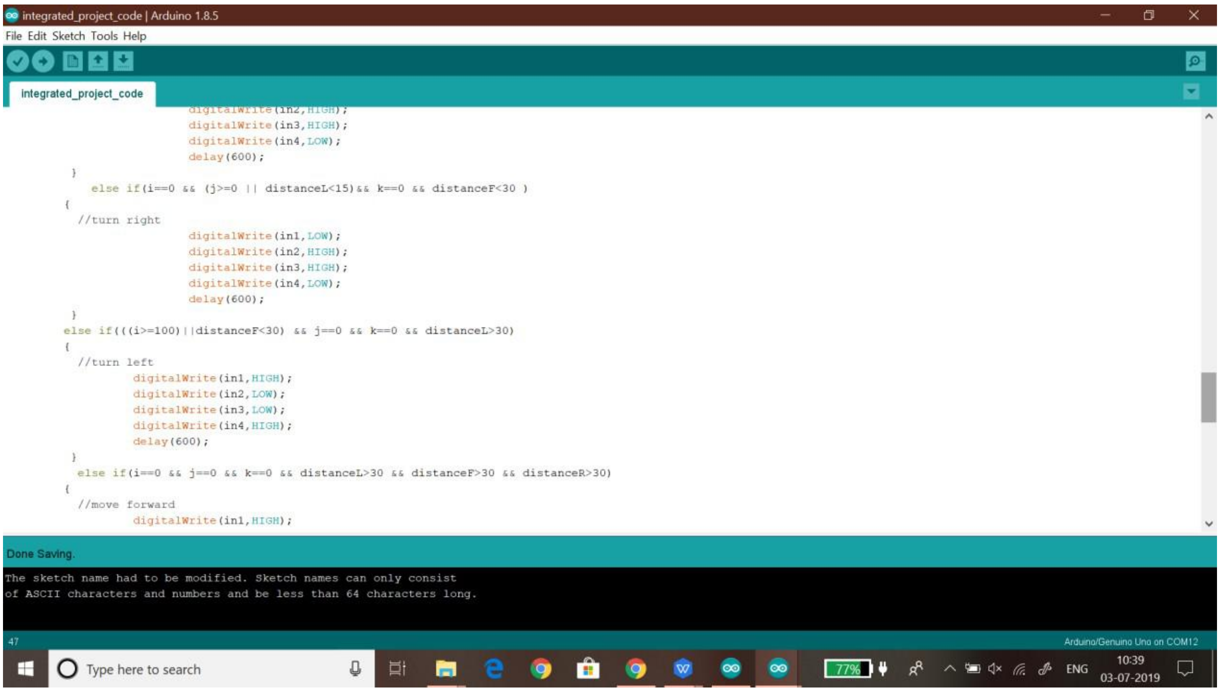


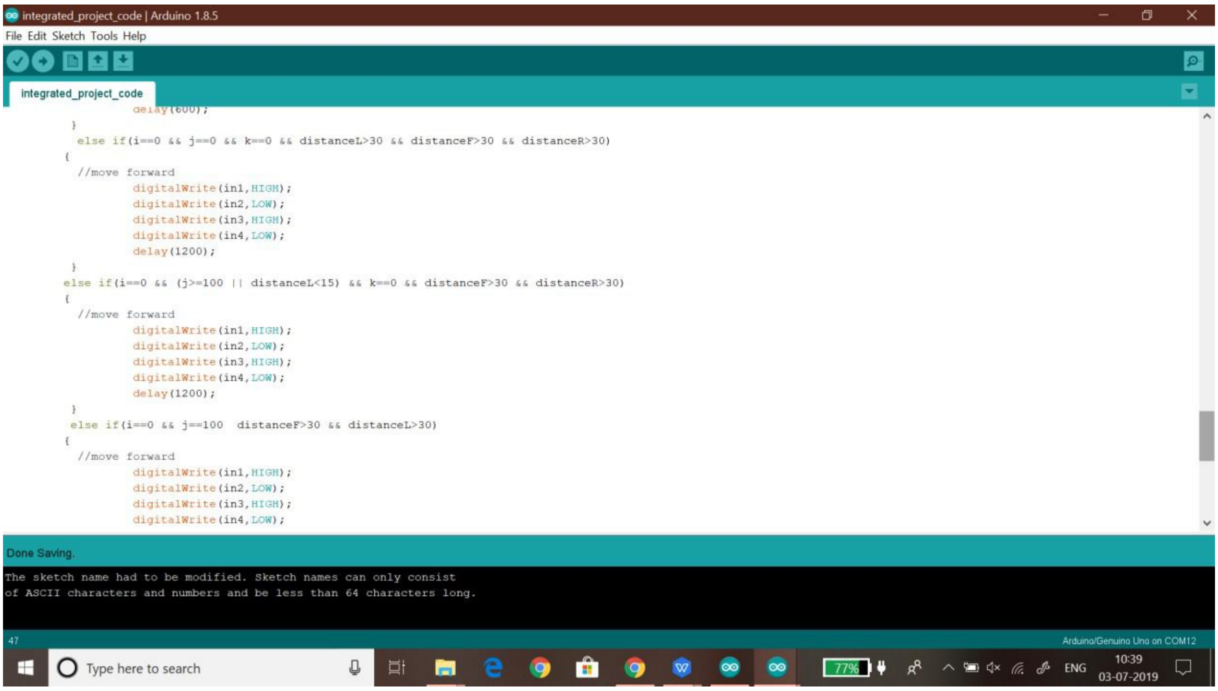


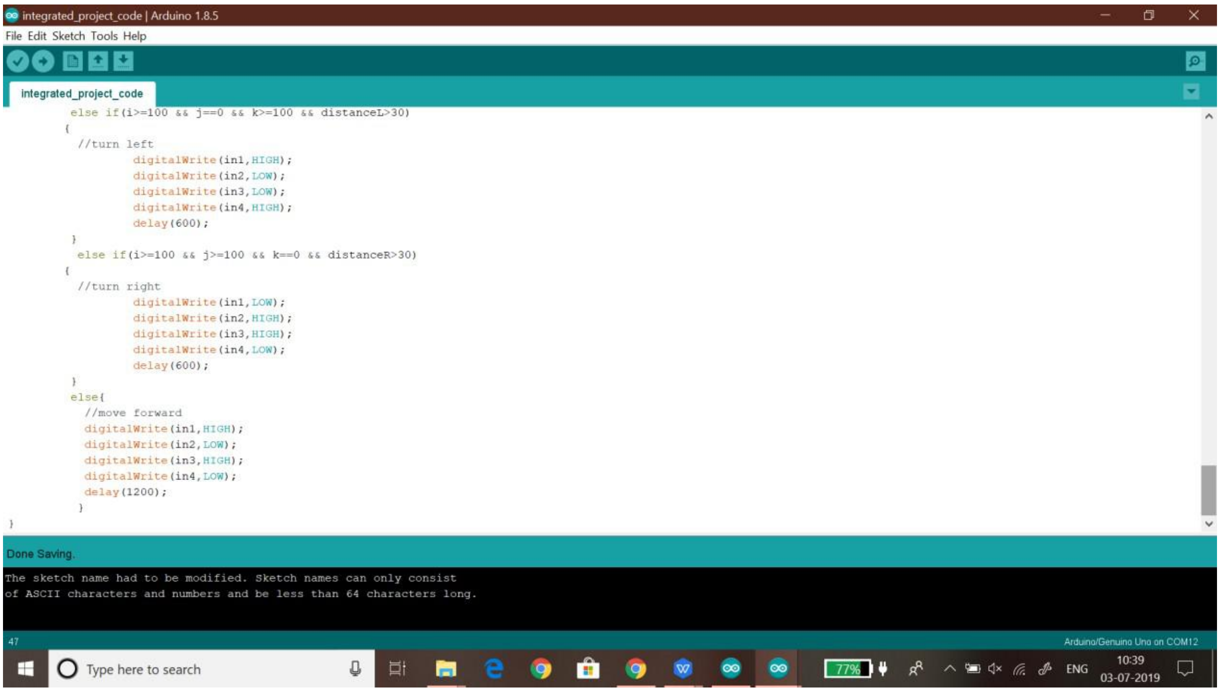

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
```
  digitalWrite(trigL,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigL,LOW);
  durationL = pulseIn(echoL,HIGH);
  distanceL = durationL/58.2;
//i is forward
//j is left
//if forward and left black prefer 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(trigL,HIGH);
          delayMicroseconds(10);
          digitalWrite(trigL,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
Type here to search
76%
ENG
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);
```
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
Type here to search
76%
ENG
10:39
03-07-2019

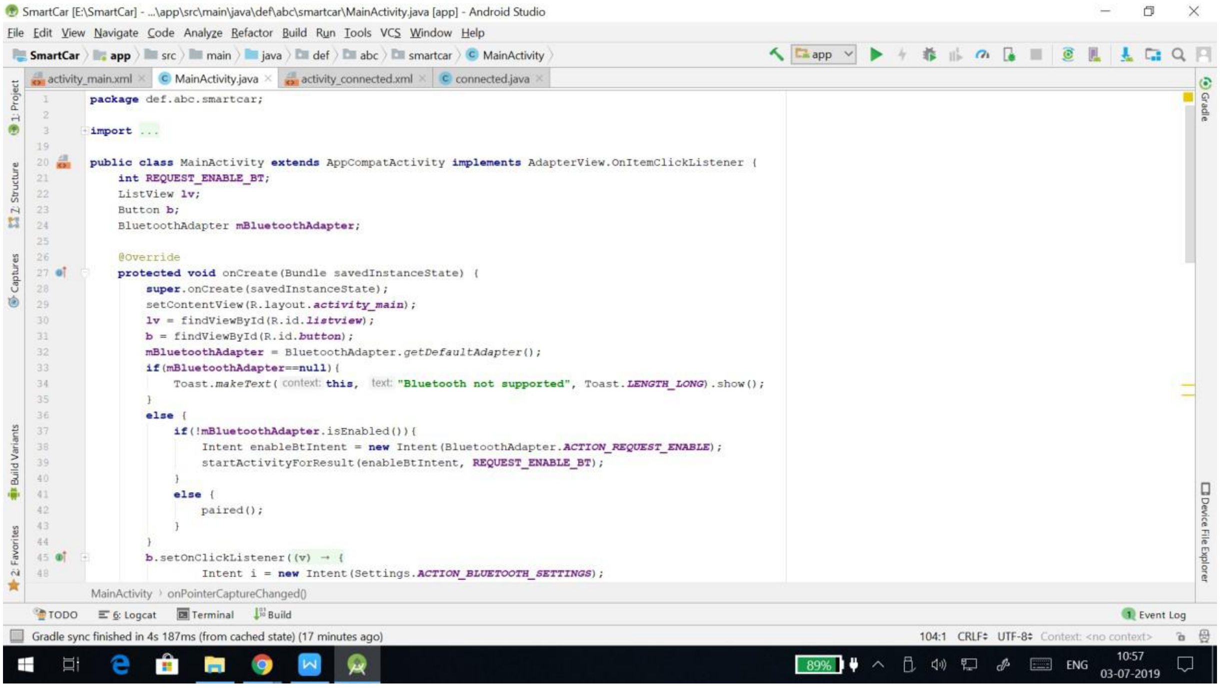

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
delay(600);
}
}
else if(i==0 && j==0 && k==0 && distanceF<30 && distanceL<15)
{
//turn right
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
delay(600);
}
else if(i==0 && j==0 && k==0 && distanceF<30)
{
//turn left
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
delay(600);
}
else if(((i>=100)||distanceF<30) && j==0 && k==0 && distanceL<15)
{
//turn right
digitalWrite(in1,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
Type here to search
77%
ENG
10:39
03-07-2019

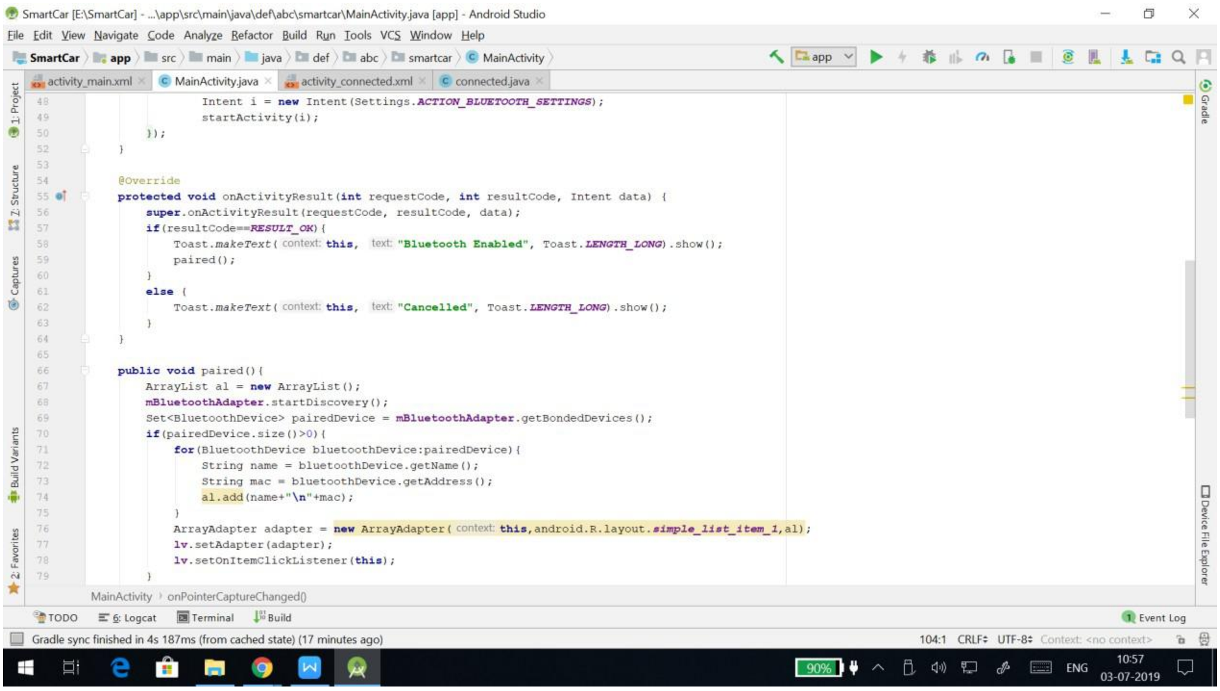

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
delay(600);
}
else if(i==0 && (j>=0 || distanceL<15)&& k==0 && distanceF<30)
{
//turn right
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
delay(600);
}
else if(((i>=100)||distanceF<30) && j==0 && k==0 && distanceL>30)
{
//turn left
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
delay(600);
}
else if(i==0 && j==0 && k==0 && distanceL>30 && distanceF>30 && distanceR>30)
{
//move forward
digitalWrite(in1,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
Type here to search
77%
ENG
10:39
03-07-2019

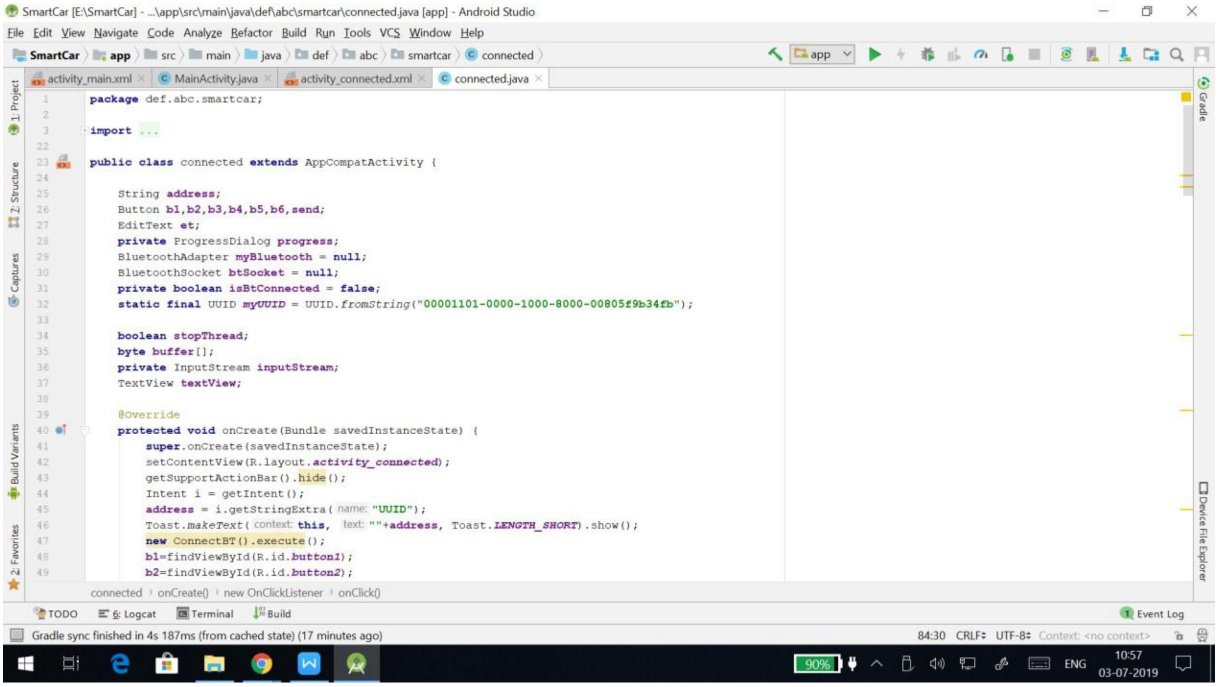

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
}
else if(i==0 && j==0 && k==0 && distanceL>30 && distanceF>30 && distanceR>30)
{
//move forward
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
delay(1200);
}
else 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
Type here to search
77%
ENG
10:39
03-07-2019

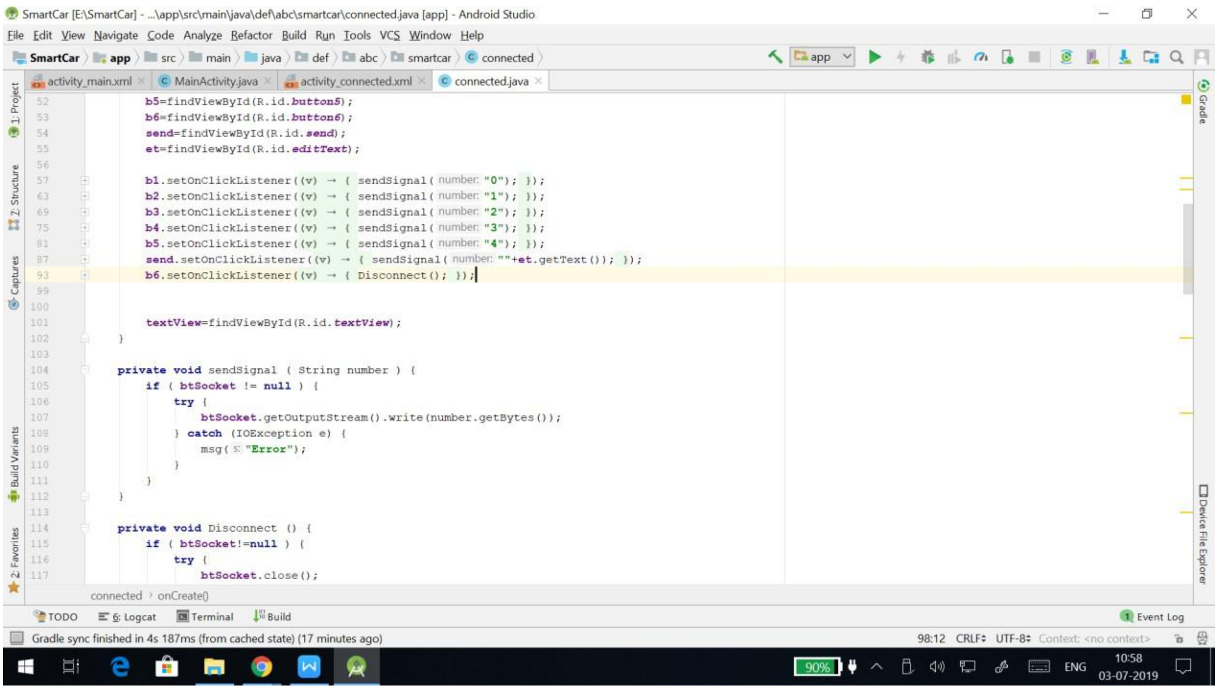

integrated_project_code | Arduino 1.8.5
File Edit Sketch Tools Help
integrated_project_code
else if(i>=100 && j==0 && k>=100 && distanceL>30)
{
//turn left
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
delay(600);
}
else if(i>=100 && j>=100 && k==0 && distanceR>30)
{
//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
Type here to search
77%
ENG
10:39
03-07-2019

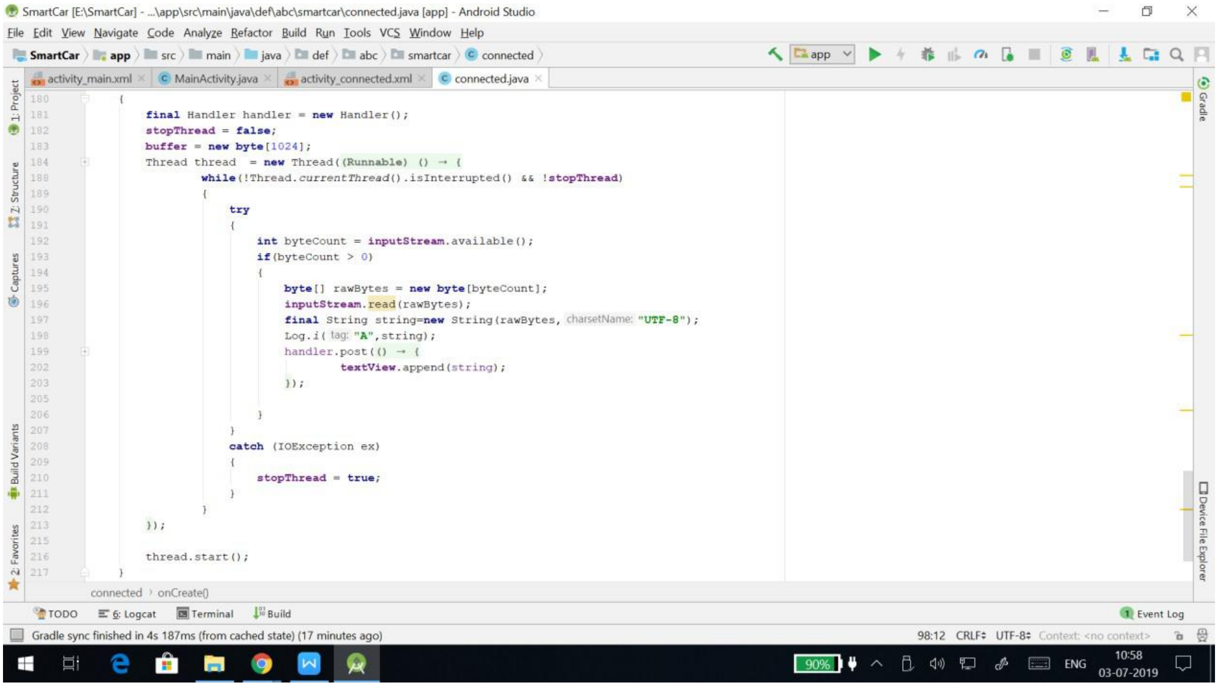
8.2 Android


SmartCar [E:\SmartCar] - ...\app\src\main\java\def\abc\smartcar\MainActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SmartCar app src main java def abc smartcar MainActivity
activity_main.xml MainActivity.java activity_connected.xml connected.java
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 enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
}
else {
paired();
}
}
b.setOnClickListener((v) → {
Intent i = new Intent(Settings.ACTION_BLUETOOTH_SETTINGS);
MainActivity › onPointerCaptureChanged()
TODO 6: Logcat Terminal Build Event Log
Gradle sync finished in 4s 187ms (from cached state) (17 minutes ago)
104:1 CRLF UTF-8 Context: <no context>
89% ENG 10:57 03-07-2019


SmartCar [E:\SmartCar] - ...\app\src\main\java\def\abc\smartcar\MainActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SmartCar app src main java def abc smartcar MainActivity
activity_main.xml MainActivity.java activity_connected.xml connected.java
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);
}
MainActivity › onPointerCaptureChanged()
TODO 6: Logcat Terminal Build Event Log
Gradle sync finished in 4s 187ms (from cached state) (17 minutes ago)
104:1 CRLF UTF-8 Context: <no context>
90% ENG 10:57 03-07-2019


SmartCar [E:\SmartCar] - ...\app\src\main\java\def\abc\smartcar\connected.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SmartCar app src main java def abc smartcar connected
activity_main.xml MainActivity.java activity_connected.xml connected.java
package def.abc.smartcar;
import ...
public class connected extends AppCompatActivity {
String address;
Button b1,b2,b3,b4,b5,b6,send;
EditText et;
private ProgressDialog progress;
BluetoothAdapter myBluetooth = null;
BluetoothSocket btSocket = null;
private boolean isBtConnected = false;
static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
boolean stopThread;
byte buffer[];
private InputStream inputStream;
TextView textView;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_connected);
getSupportActionBar().hide();
Intent i = getIntent();
address = i.getStringExtra(name: "UUID");
Toast.makeText(context: this, text: ""+address, Toast.LENGTH_SHORT).show();
new ConnectBT().execute();
b1=findViewById(R.id.button1);
b2=findViewById(R.id.button2);
connected onCreate() new OnClickListener onClick()
TODO 6: Logcat Terminal Build Event Log
Gradle sync finished in 4s 187ms (from cached state) (17 minutes ago)
84:30 CRLF UTF-8 Context: <no context>
90% ENG 10:57 03-07-2019


SmartCar [E:\SmartCar] - ...\app\src\main\java\def\abc\smartcar\connected.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SmartCar app src main java def abc smartcar connected
activity_main.xml MainActivity.java activity_connected.xml connected.java
b5=findViewById(R.id.button5);
b6=findViewById(R.id.button6);
send=findViewById(R.id.send);
et=findViewById(R.id.editText);
b1.setOnClickListener((v) → { sendSignal(number: "0"); });
b2.setOnClickListener((v) → { sendSignal(number: "1"); });
b3.setOnClickListener((v) → { sendSignal(number: "2"); });
b4.setOnClickListener((v) → { sendSignal(number: "3"); });
b5.setOnClickListener((v) → { sendSignal(number: "4"); });
send.setOnClickListener((v) → { sendSignal(number: ""+et.getText()); });
b6.setOnClickListener((v) → { Disconnect(); });
textView=findViewById(R.id.textView);
}
private void sendSignal (String number) {
if (btSocket != null) {
try {
btSocket.getOutputStream().write(number.getBytes());
} catch (IOException e) {
msg(s: "Error");
}
}
}
private void Disconnect () {
if (btSocket!=null) {
try {
btSocket.close();
connected onCreate()
TODO 6: Logcat Terminal Build Event Log
Gradle sync finished in 4s 187ms (from cached state) (17 minutes ago)
98:12 CRLF UTF-8 Context: <no context>
90% ENG 10:58 03-07-2019


SmartCar [E:\SmartCar] - ...\app\src\main\java\def\abc\smartcar\connected.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
MainActivity.java
activity_connected.xml
connected.java
final Handler handler = new Handler();
stopThread = false;
buffer = new byte[1024];
Thread thread = new Thread((Runnable) () → {
while(!Thread.currentThread().isInterrupted() && !stopThread)
try
int byteCount = inputStream.available();
if(byteCount > 0)
byte[] rawBytes = new byte[byteCount];
inputStream.read(rawBytes);
final String string=new String(rawBytes, charsetName: "UTF-8");
Log.i(tag: "A",string);
handler.post(() → {
textView.append(string);
});
catch (IOException ex)
stopThread = true;
});
thread.start();
Gradle sync finished in 4s 187ms (from cached state) (17 minutes ago)
10:58
03-07-2019

Chapter 9

Screenshots of Project

9.1 SENSORS

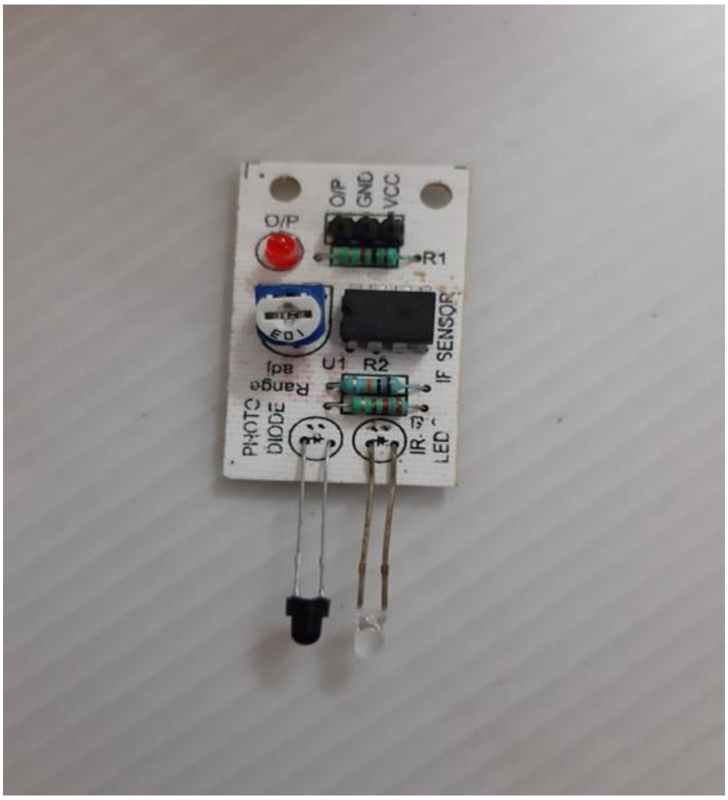


Figure 9. 1: IR SENSOR

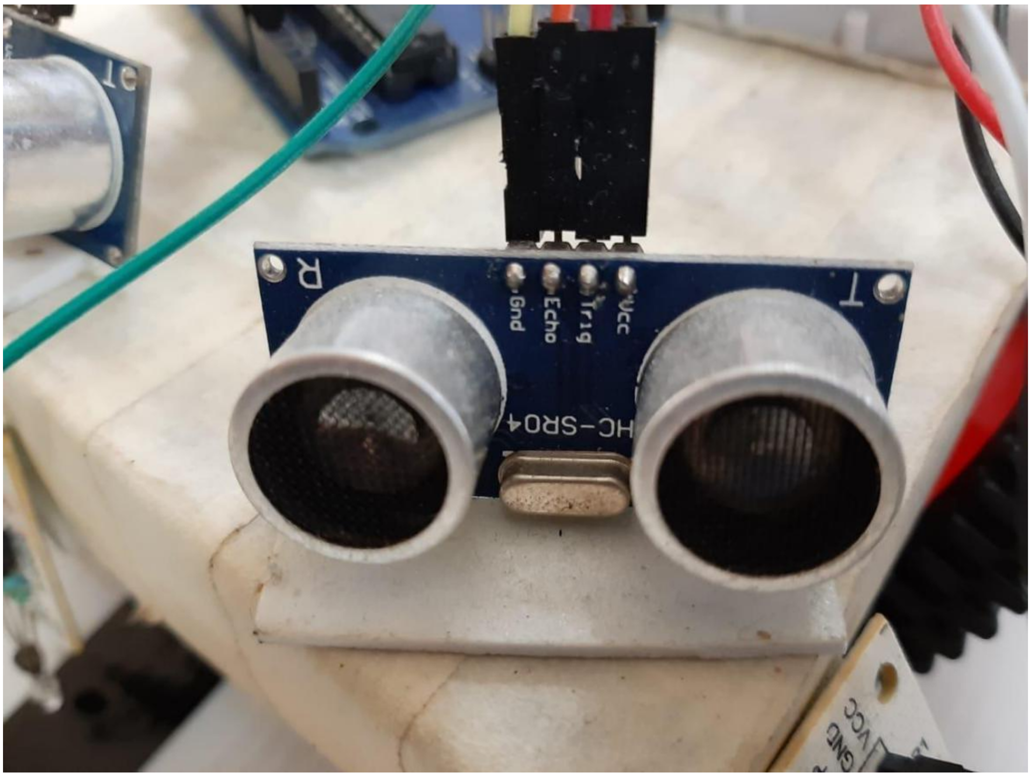


Figure 9 . 2 : ULTRASONIC SENSOR

Figure 9 . 3 : HC-05

9.2 BOARDS

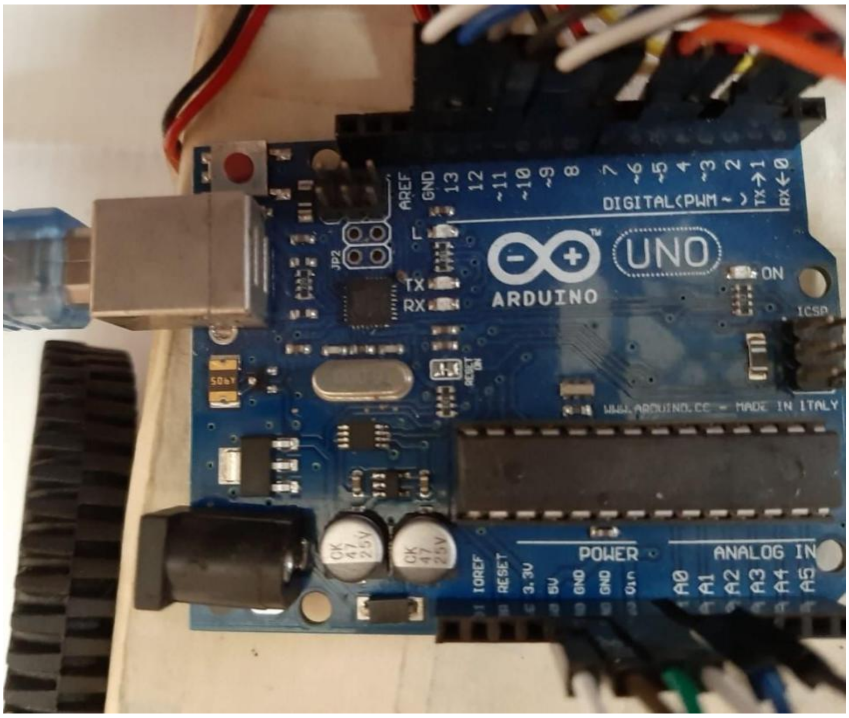


Figure 9 . 4 : A R D U I N O U N O B O A R D

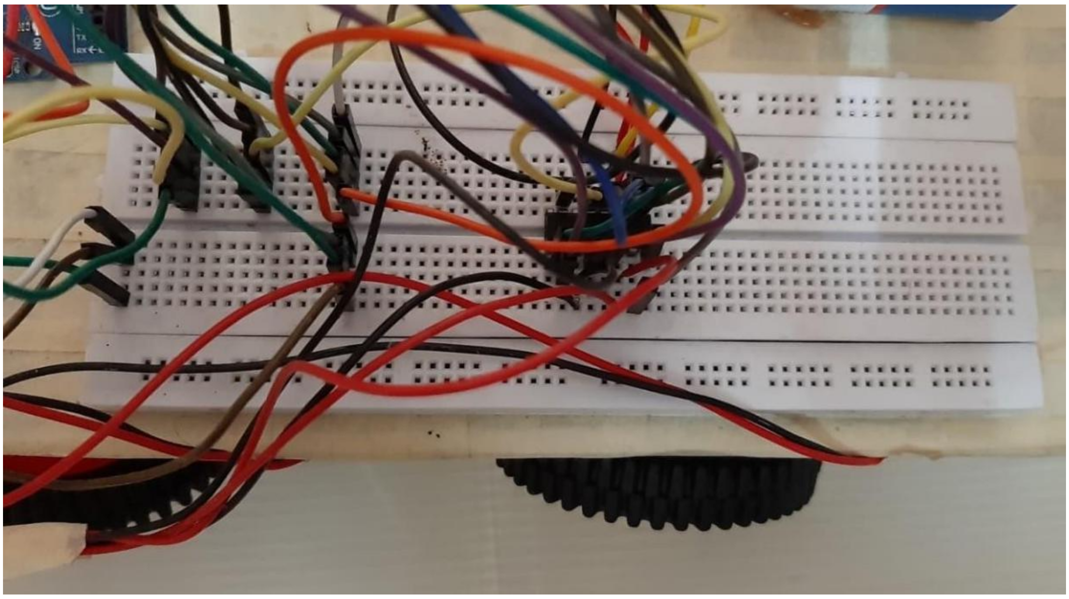


Figure 9 . 5 : B R E A D B O A R D

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