Contents

[Chapter 1: INTRODUCTION 2](#_Toc519071708)

[1.1 Motivation 2](#_Toc519071709)

[1.2 objective of the project: 2](#_Toc519071710)

[Chapter 2: TOOLS AND TECHNOLOGY USED 3](#_Toc519071711)

[2.1 Ultrasonic Sensor 3](#_Toc519071712)

[2.2 Crash Sensor[10T85 switch] 4](#_Toc519071713)

[2.3 IR sensors 5](#_Toc519071714)

[2.4 Raspberry pi 6](#_Toc519071715)

[2.5 Android app and Firebase 7](#_Toc519071716)

[Chapter 3: COMPLETE INDENTED CODE 8](#_Toc519071717)

[3.1 Code: 8](#_Toc519071718)

[3.1.1 Code for the test course 8](#_Toc519071719)

[3.1.2 Code for android app 14](#_Toc519071720)

[3.2 Block Diagram 15](#_Toc519071721)

[Chapter 4 : SNAPSHOTS 15](#_Toc519071722)

[CHAPTER 5: RESULTS AND DISCUSSIONS 18](#_Toc519071723)

[5.1 Results 18](#_Toc519071724)

[5.2 Discussions 18](#_Toc519071725)

[CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE 18](#_Toc519071726)

**Table of Figures:**

* Figure 1: Ultrasonic Sensor
* figure2:Crash Sensor
* figure3: IR Sensor
* figure4: Raspberry pi with pin diagram
* figure5: Block Diagram of the project
* figure6: The Test course
* figure7:The Ramp test
* figure8:The Curve test
* figure9: The Overtake test
* Figure10: The Parking test

**Abbreviations and Nomenclature:**

1. SDLTS:- Smart Driving License Testing System

2. DL:- Driving License

3. IR Sensor:- Infrared Sensor

4. RTO:- Regional Transport Office

# Chapter 1: INTRODUCTION

Getting a driving license is an important thing in every adult's life. The RTO office issues the license to a trainee provided the trainee passes the prescribed test. These tests should challenge the capability of the trainee in every way possible. The aspirant must be perfect and confident in his/her driving. In the end, the trainee has to earn his/her license.

## 1.1 Motivation

We noticed that there are a few problems with the way these tests are being held. There is no opportunity for the driver to test his skills before the driving test. On the day of the driving test, the middle men from the driving schools may create some problems to the trainee and demand for more money. Bribery may happen and a person who isn't a good driver may end up getting a license which would be disasterous. Too many people will be involved on the day of the test and this may cause distress in the minds of the driver. All these factors serve as motivation for us as we feel that the aspirant must be a perfect driver before getting a license to drive as this would decrease the number of accidents on the road.

## 1.2 objective of the project:

The SDLTS aims at designing a driving license test track which would be efficient and would incorporate the modern technology known to us. We aim to test every single ability that is required while driving on the road. We wish to remove the small problems mentioned above with our automated driving license test.

# Chapter 2: TOOLS AND TECHNOLOGY USED

## 2.1 Ultrasonic Sensor

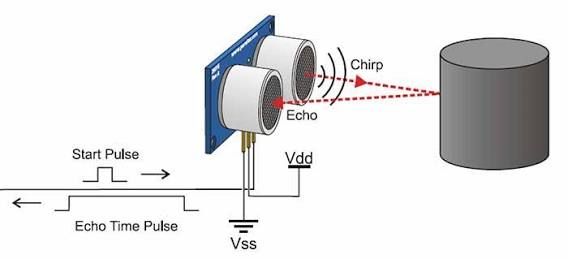
****

figure1: Ultrasonic Sensor

Ultrasonic sensor measures the distance by using Ultrasonic waves. The sensor head emits an Ultrasonic wave and receives the wave reflected back from the target. Ultrasonic sensors measure the distance to the target by measuring the time between emission and reception. An optical sensor has a transmitter and receiver, whereas an Ultrasonic sensor uses a single ultrasonic element for both emission and reception. In Ultrasonic sensor, a single oscillator emits and receives Ultrasonic waves alternately.

The distance can be calculated using the formula :

Distance L = 1/2 x T x C ,

where: L is the distance between Ultrasonic sensor and target,

T is the time between emission and reception,

C is the sonic speed.

Ultrasonic sensor is being used in two tests in the test course:

1. The Ramp test: The Ultrasonic sensor is used to measure the distance maintained the driver from the end of the track. If he's unable and brings the car backward, points will be deducted accordingly. If he maintains his distance and moves forward in the slope, then no points will be cut. After passing a particular mark, he'll be eligible for the next test.

2. The Overtake test: This test evaluates the overtaking ability of the driver. If he is too close or strays too far from the vehicle he is overtaking, then the points will be deducted. If he overtakes the vehicle in the prescribed lane, then no points will be deducted.

## 2.2 Crash Sensor[10T85 switch]

The switch 10T85 is a snap action switch. It is connected internally in such a way that the pins 1 and 3 are connected. The pin 2 is connected to pin 1 when the switch is closed. The switch is used as a crash sensor.

****

**figure2:Crash Sensor**

External connections :-

1. pin1 is connected to the microcontroller(Raspberry pi 2) GPIO pin.

2. pin 2 is connected to Vcc .

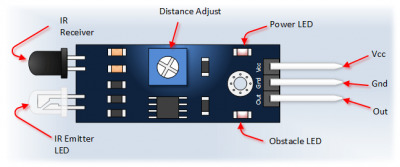
3. pin 3 is connected to ground through a resistor(220 ohms).

The switches are being used in two tests in the course :

1. The Curve test : A total of 10 switches are used, 5 in each side of the curved road. These switches are connected in parallel so that all the switches are connected and if one of them is pressed, it could be detected by the GPIO pin. The switches on the right are connected to GPIO pin 26 and the switches on the left are connected to GPIO pin 19. Since the switches are in parallel, the status of the switches can be read on one GPIO pin. But in this case, we make use of two GPIO pins one for each side of the road.

2. The Parking test: Here, one switch is being used for determine whether the course has been completed or not. After successful completion of the parking test, the driver is to press the switch indicating that he has passed the test under the prescribed time limit. If he is unable to do so, then that particular test will be marked as failed.

## 2.3 IR sensors

****

**figure3: IR Sensor**

Infrared Obstacle Sensor Module has built in IR transmitter and IR receiver that sends out IR energy and looks for reflected IR energy to detect presence of any obstacle in front of the sensor module. The module has on board potentiometer that lets user adjust detection range.

In our test course, IR sensors are being used in two tests-

1. The Curve test : The IR sensor is being used to check if the vehicle has successfully exited the curve test .The driver must pass this IR sensor under the prescribed time limit indicating that he has completed the test successfully. Failure to do this will result in deduction of points allotted for that particular test.

2. The Parking test : Here, two IR sensors are being used to check if the vehicle is straying from the allotted area for parking. If found so, the points will be deducted accordingly.

## 2.4 Raspberry pi

****

**figure4: Raspberry pi with pin diagram**

The Raspberry Pi is a series of small single-board computer. The Raspberry Pi used in our project is Raspberry pi 2. It has 40 pins of which 8 GPIO pins are being used in our project.The power supply provided from the PC through a micro USB cable.LAN cable is used for connection. It has an HDMI port, an SD card slot, an audio jack and four USB slots.

Software used to setup the Raspberry Pi are:

1.win 32 disc imager

2.VNC viewer

3.SD Formatter V4

4.Putty

5.Advanced IP scanner

Raspberry pi has its own operating system called Raspbean operating system which is to be installed into an SD card using the above mentioned software when it is being used for the first time. After setup, create a directory for storing a python file while is to be run.

## 2.5 Android app and Firebase

Android Studio is the official integrated development environment for Google’s Android operating system, built on JetBrains’IntelliJ IDEA software and designed specifically for Android development. Android Studio is a platform where we can build interactive application, it’s a user friendly development app. Different kind of interaction can be built in Android Studio. It is possible be to send the data, retrieve the data from the web with the help of Database commonly used Firebase Database.

Android Application:S SDLTS

SDLTS is an android application used to fetch data about efficiency of a driver and to decide whether he is eligible to get license.

SDLTS is used for registration, for beginning the test and finally to display the result. Result is stored in database which is created using google platform called Firebase.

Firebase platform allows the user to create the database and gives the 'jason' file which can be directly inserted in the application.

Software and technology used for creating the app are:

1. Android studio

2. Google Firebase

# Chapter 3: COMPLETE INDENTED CODE

## 3.1 Code:

### 3.1.1 Code for the test course

*import RPi.GPIO as GPIO*

*import time*

*import Pyrebase*

*config={""}*

*firebase=pyrebase.initialize\_app(config)*

*trigger\_Ramp1=2*

*echo\_Ramp1=3*

*trigger\_Ramp2=17*

*echo\_Ramp2=27*

*ir\_pin1=10*

*ir\_pin2=9*

*ultimatePin=18*

*switch1=26*

*switch2=19*

*test2\_finish=5*

*GPIO.setmode(GPIO.BCM)*

*GPIO.setup(trigger\_Ramp1,GPIO.OUT)*

*GPIO.setup(echo\_Ramp1,GPIO.IN)*

*GPIO.setup(trigger\_Ramp2,GPIO.OUT)*

*GPIO.setup(echo\_Ramp2,GPIO.IN)*

*GPIO.setup(ir\_pin1,GPIO.IN)*

*GPIO.setup(ir\_pin2,GPIO.IN)*

*GPIO.setup(ultimatePin,GPIO.IN)*

*GPIO.setup(switch1,GPIO.IN)*

*GPIO.setup(switch2,GPIO.IN)*

*GPIO.setup(test2\_finish,GPIO.IN)*

*def send\_trigger\_pulse(trigger\_Ramp):*

*GPIO.output(trigger\_Ramp,True)*

*time.sleep(0.0001)*

*GPIO.output(trigger\_Ramp,False)*

*def wait\_for\_echo(value,timeout,echo\_Ramp):*

*count=timeout*

*while GPIO.input(echo\_Ramp) != value and count > 0:*

*count =count-1*

*def get\_distance(trigger\_Ramp,echo\_Ramp):*

*send\_trigger\_pulse(trigger\_Ramp)*

*wait\_for\_echo(True,10000,echo\_Ramp)*

*start=time.time()*

*wait\_for\_echo(False,10000,echo\_Ramp)*

*finish=time.time()*

*pulse\_len=finish-start*

*distance\_cm=pulse\_len/0.000058*

*return (distance\_cm)*

*def Ramp():*

*points=100*

*while True:*

*a=18*

*while a<19:*

*time.sleep(1)*

*a=get\_distance(trigger\_Ramp1,echo\_Ramp1)*

*print(a)*

*if a< 10.8 and a>7.3:*

*points=points-2*

*print(points)*

*time.sleep(2)*

*elif a<= 7.3:*

*points=points-4*

*print(points)*

*else:*

*time.sleep(2)*

*if a<10.8:*

*continue*

*else:*

*if a>19:*

*print(points)*

*return points*

*print(points)*

*return points*

*def curve():*

*crash=0*

*points=100*

*r=0*

*start=time.time()*

*while True:*

*a=GPIO.input(switch1)*

*b=GPIO.input(switch2)*

*if a:*

*while a:*

*a=GPIO.input(switch1)*

*time.sleep(0.2)*

*points=points-10*

*print(points)*

*crash=crash+1*

*print(crash)*

*print(a)*

*continue*

*if b:*

*while b:*

*b=GPIO.input(switch2)*

*time.sleep(0.2)*

*points=points-10*

*print(points)*

*crash=crash+1*

*print(crash)*

*print(b)*

*continue*

*finish=time.time()*

*tTime=int(finish-start)*

*if test2\_finish==1:*

*return points*

*if(tTime>=20):*

*if crash==0:*

*points=0*

*return points,crash*

*def overtake():*

*while True:*

*points=100*

*a=get\_distance(trigger\_Ramp2,echo\_Ramp2)*

*time.sleep(1)*

*print(a)*

*temp=0*

*while a< 25:*

*a=get\_distance(trigger\_Ramp2,echo\_Ramp2)*

*temp+=1*

*if a<7.2 or a>13.5 and a<25:*

*print("too close or too far")*

*points=points-10*

*time.sleep(1)*

*print(points)*

*else:*

*print("good")*

*time.sleep(1)*

*print(points)*

*if temp>0:*

*break*

*return points*

*def parking():*

*points=100*

*c=0*

*start=time.time()*

*while True:*

*finish=time.time()*

*tTime=int(finish-start)*

*print(tTime)*

*if tTime<=20:*

*a=GPIO.input(ir\_pin1)*

*b=GPIO.input(ir\_pin2)*

*if (a==0 or b==0 ):*

*points=0*

*print(points)*

*time.sleep(1)*

*return points*

*u=GPIO.input(ultimatePin)*

*if u==1:*

*return points*

*else:*

*points=0*

*return points*

*while True:*

*data1=0*

*i=0*

*l=[]*

*for i in range(0,20):*

*l.append(data.child("/").child("customer").child(i+1).child("status").get().val())*

*i=i+1*

*while True:*

*for j in l:*

*if(j==1):*

*name=data.child("/").child("customer").child(l.index(j)+1).child("cname").get().val()*

*id=data.child("/").child("customer").child(l.index(j)+1).child("cid").get().val()*

*value1 = Ramp()*

*points,crash = curve()*

*value3 = overtake()*

*value4 = parking()*

*totalScore = value1+points+value3+value4*

*if(totalScore>=350):*

*result='PASS!!!'*

*else:*

*result='FAIL'*

*data.child("/").child("customer").child(l.index(j)+1).child("result").set(result)*

*data.child("/").child("customer").child(l.index(j)+1).child("status").set(data1)*

*break*

*break*

The libraries installed in the code are:

1. Pyrebase

2. time

3.RPi.GPIO

### 3.1.2 Code for android app

[**https://github.com/ajaykami007/applicationandroid**](https://github.com/ajaykami007/applicationandroid)

## 3.2 Block Diagram

**P1**

**Ramp()**

**P2,Crash**

**curve()**

**P3**

**overtake()**

**P4**

**parking()**

**TotalScore = P1+P2+P3+P4**

**if**

**TotalScore >=300**

**False**

**True**

**Result = PASS**

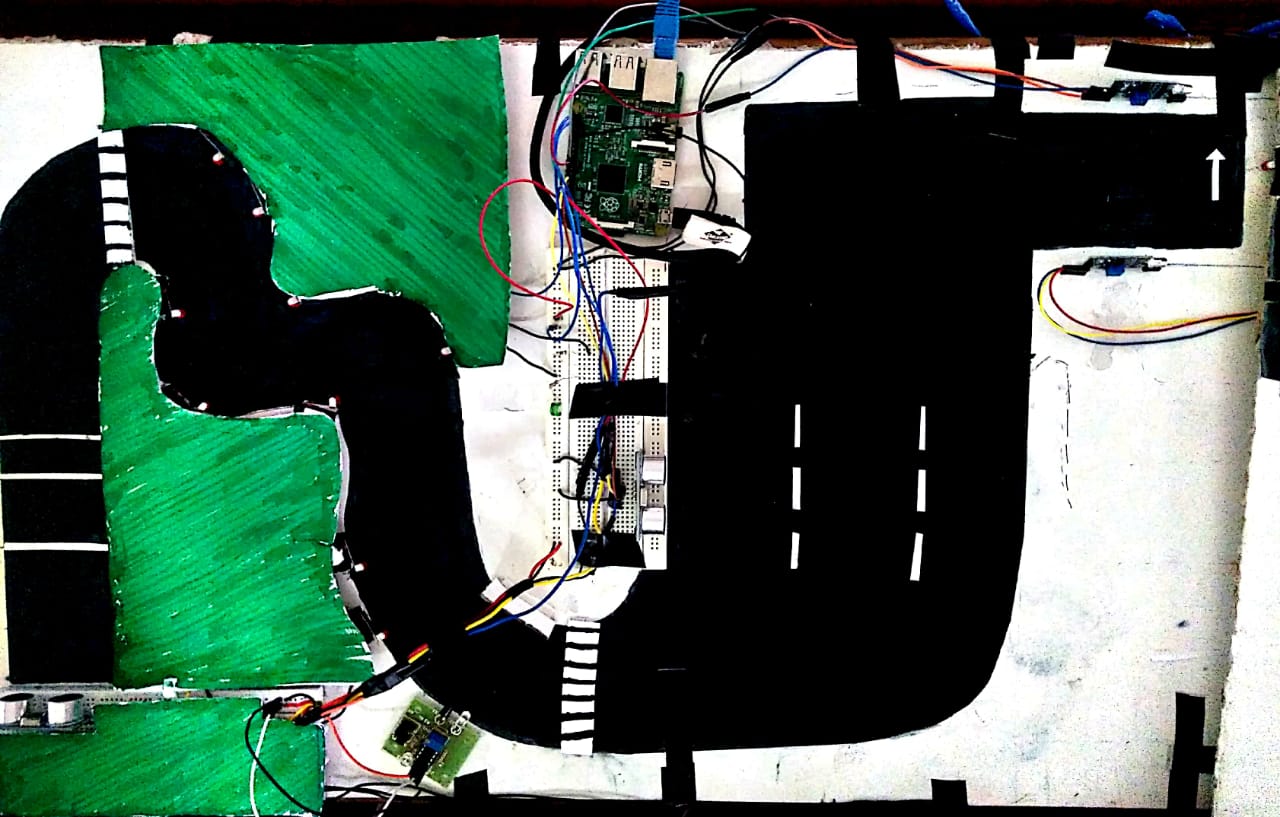
**Result = FAIL**

**Android App**

**figure5: Block Diagram of the project**

# 

# Chapter 4 : SNAPSHOTS

****

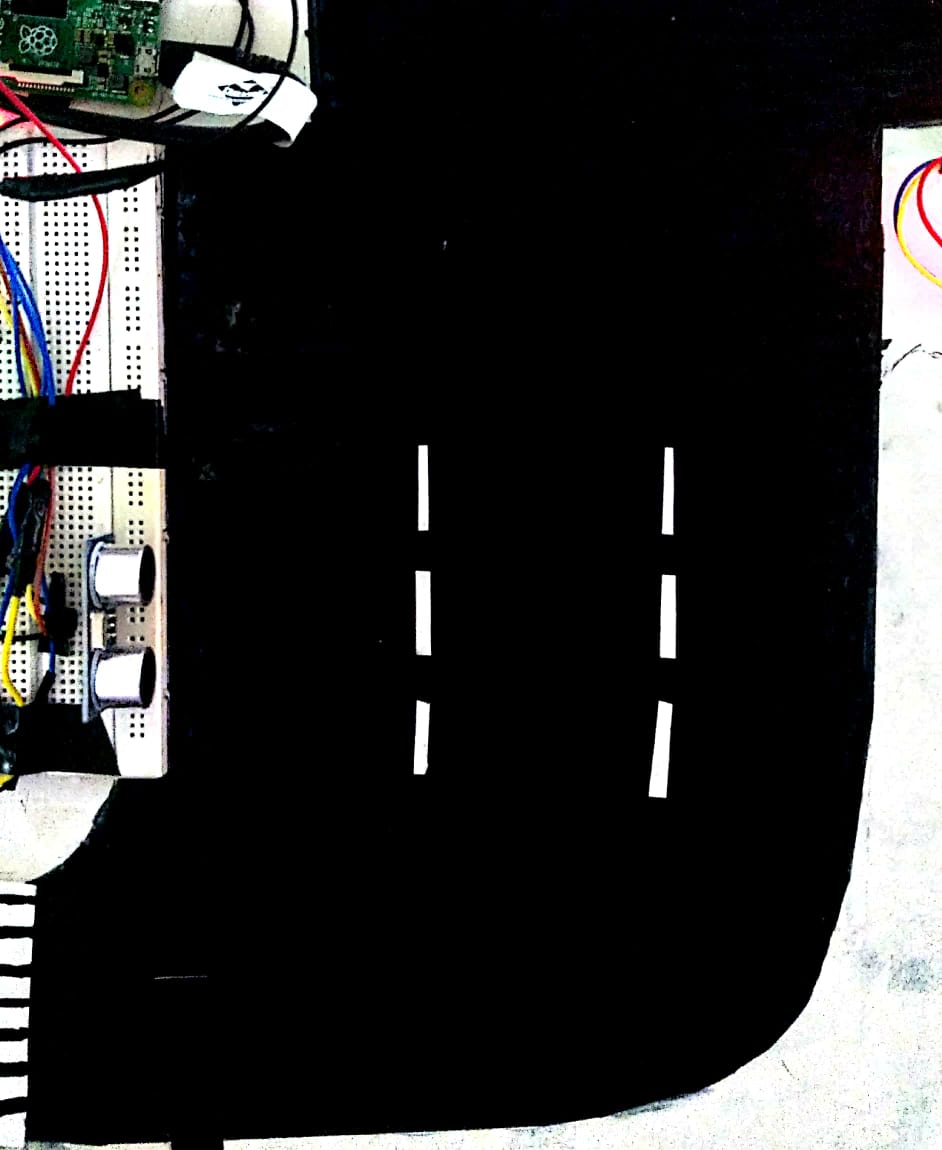
**figure6: The Test course**

****

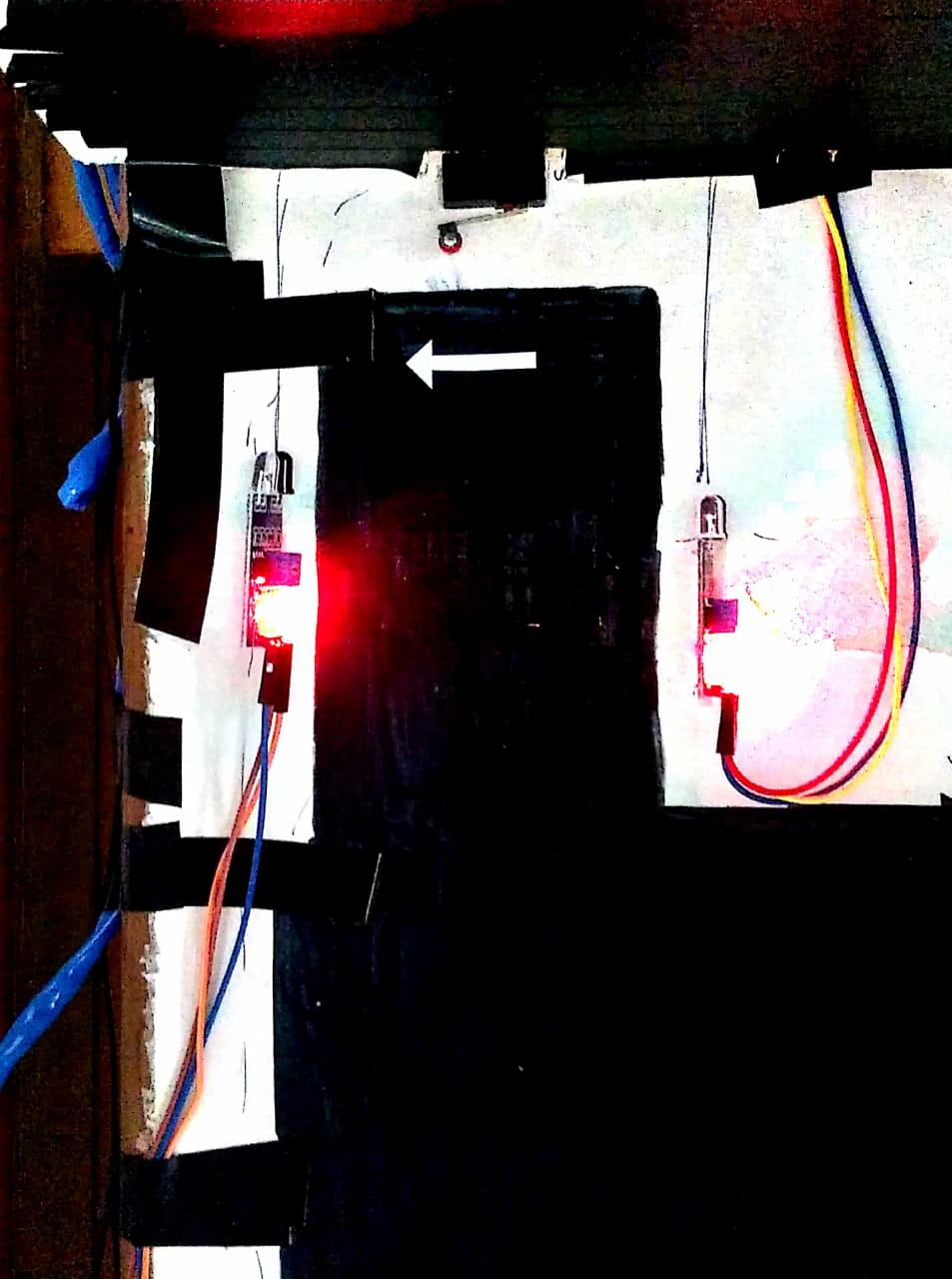
**figure7:The Ramp test**

****

**figure8:The Curve test**

****

**figure9: The Overtake test**

****

**Figure10: The Parking test**

# CHAPTER 5: RESULTS AND DISCUSSIONS

## 5.1 Results

There are four tests in the test course. Out of four tests in the course, each one is for 100 points . So the course evaluates the driver for 400 points. The course begins when the driver presses the 'Begin Test' in the SDLTS app on his android phone. If the driver commits a mistake, then accordingly the points will be deducted. After completion of the course, if the driver has a score equal to or above 300, then he is declared as 'PASSED' and is awarded his driving license. If the score is less than 300 , he/she is declared as 'FAILED' and can retry the test after getting permission from the RTO. The result will be displayed on the SDLTS app on his phone.

## 5.2 Discussions

Some of the discussions are:

1. In the Ramp test, a time limit can be applied in real test scenario .

2. In the Curve test, the some changes can be made like the time limit within which he should finish the test. It can be decreased or increased depending on the complexity and length of the curved track.

3. The overtake test can be made more complex by placing a vehicle in the track making the driver to go around the vehicle.

4. After the test, the driver has to request the RTO office to provide him with the results of each test. The android app can be modified to provide all these information directly after the test by getting the permission from the RTO office.

# CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE

The test course will test every ability of the driver required on the road. If the driver fails, he can always retry the course after getting permission from the RTO office. It is an efficient and automated kind of driving test where no outside interference is possible. The problems of the middle men and bribery are completely abolished here.

The government can easily implement this in every city. It is cost effective rigorous and efficient. It can be implemented on the same track on which the DL test has been conducted till date without having to go through the trouble of rebuilding a new one. It is flexible. The course can be made easier or more difficult according to the wish of the city's RTO office. This is the best way for a driving test to be held. It has already been implemented in a few states. We wish that in a few years it would become more popular and be implemented throughout the nation .