SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE NERUL, NAVI MUMBAI-400706

DEPARTMENT OF COMPUTER SCIENCE

MSc (CS) PART-II SEMESTER IV

Practical Journal

In

Robotics

Course Code: PSCS401

Submitted by

Aman K. Mulani

SEAT NUMBER -

ROLL NUMBER -

Prof. in charge of Practical: Mrs Pallavi Awate

For the academic year

2023-2024

SIES(NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE
NAAC RE-ACCREDITED 'A' GRADE
SRI CHANDRASEKARENDRA SARASWATHY VIDHYAPURAM
PLOT 1-C SECTOR V
, NERUL NAVI MUMBAI -400706

CERTIFICATE

This is to certify that the Practical Journal of subject entitled, **"Advanced Deep Learning"**, is bona fide work of **Aman K. Mulani** of Masters Of Science in Computer Science (Part 2) submitted in partial fulfillment of the requirements for the award of degree of **MASTER OF SCIENCE** in **COMPUTER SCIENCE** from University of Mumbai. It is also to clarify that this is the original work of the candidate done during the academic year 2023-24

Seat Number: _____ Date : _____

_____ _____

Mrs. Pallavi Awate — External Examiner

(Teacher In-Charge)

_____ _____

Dr.Sheeja Ravi — Dr. Koel Roychoudhury

(Head Of Department) — (Principal)

Index

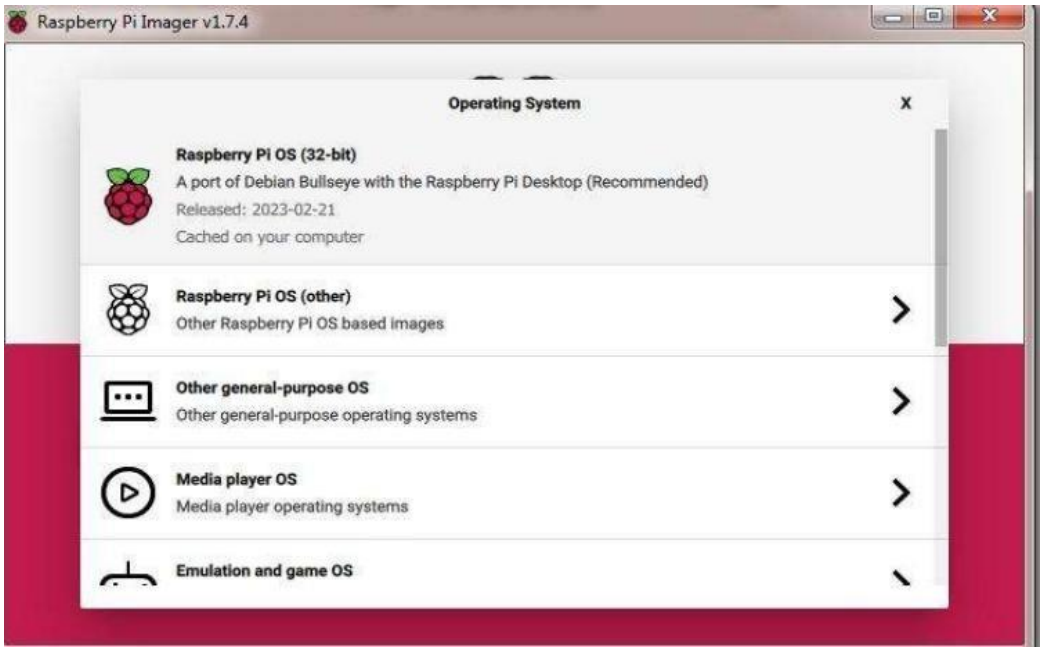
Sr. No.	Practical	Date
1	Making Raspberry pi headless, and reaching it from the network using WIFI and SSH.	
2	Using sftp upload files from PC.	
3	Write Python code to test motors.	
4	Write a script to follow a predetermined path.	
5	Develop Python code for testing the sensors.	
6	Add the sensors to the Robot object and develop the line-following behavior code.	
7	Using the light strip develop and debug the line follower robot.	
8	Create an obstacle avoidance behavior for robot and test it.	
9	Detect faces with Haar cascades.	
10	Use the robot to display its camera as a web app on a phone or desktop, and then use the camera to drive smart color and face-tracking behaviors.	

Practical No:- 1

Aim:- Making Raspberry pi headless, and reaching it from the network using WIFI and SSH.

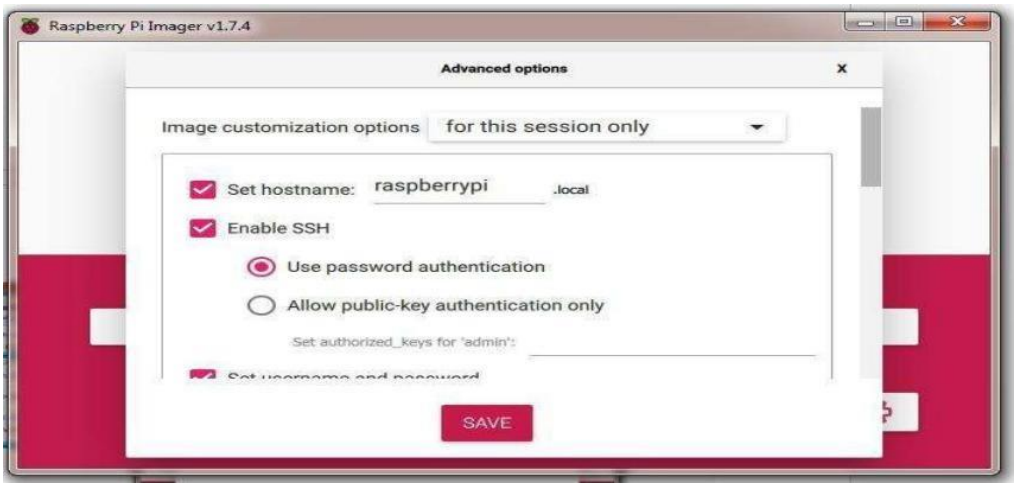
STEP I: Go to www.raspberry.com and click on software tab. Download raspberry pi imager for windows. And connect SD card to laptop.

STEP II : Select operating system (raspberry pi OS 32-bit).

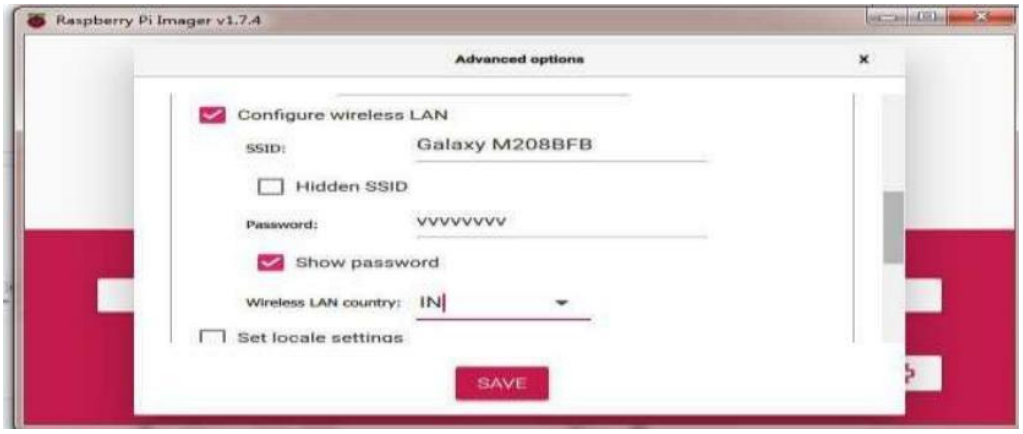


STEP III : Select Storage

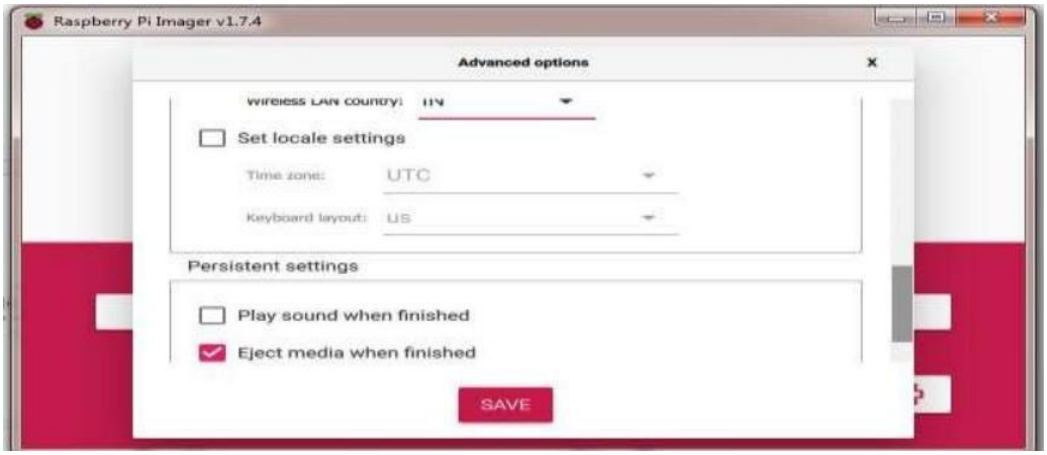
STEP IV : Click on Setting Icon . and set hostname. Click to enable SSH.



STEP V : Click to configure wireless LAN. And select country.



STEP VI : Click to save.



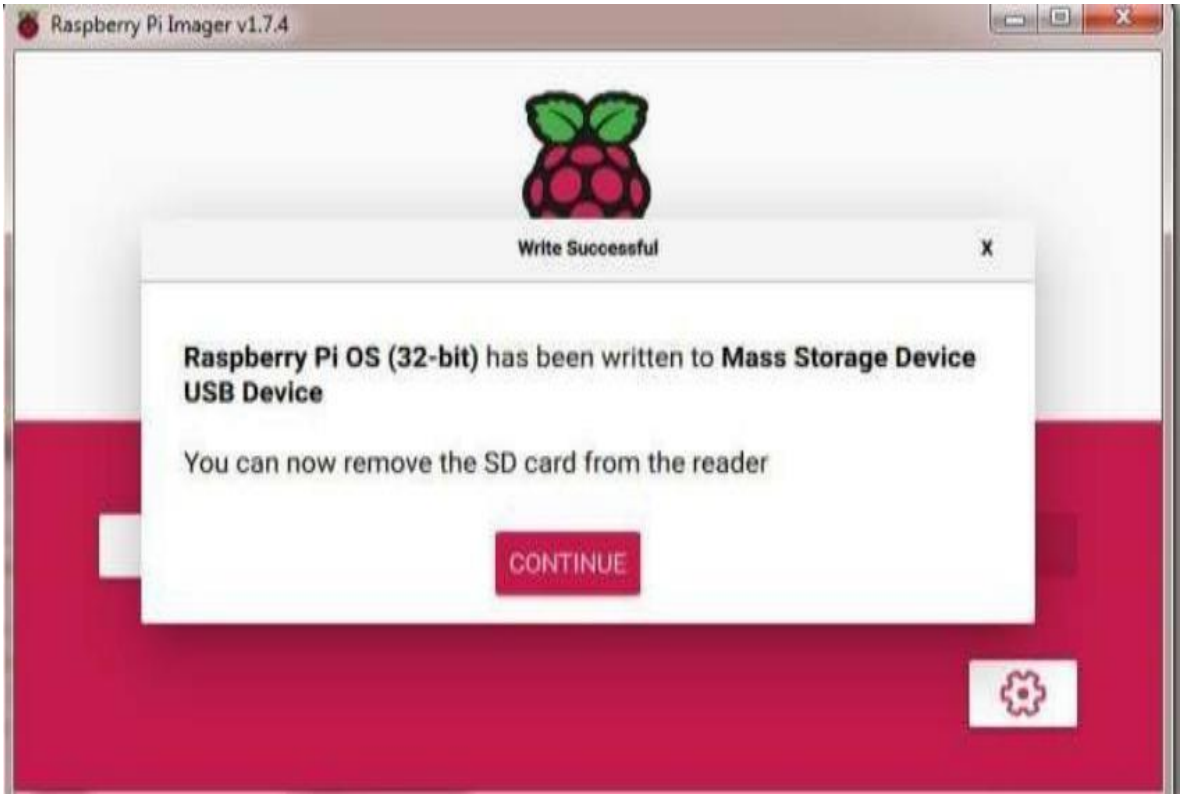
STEP VII : Click on write.

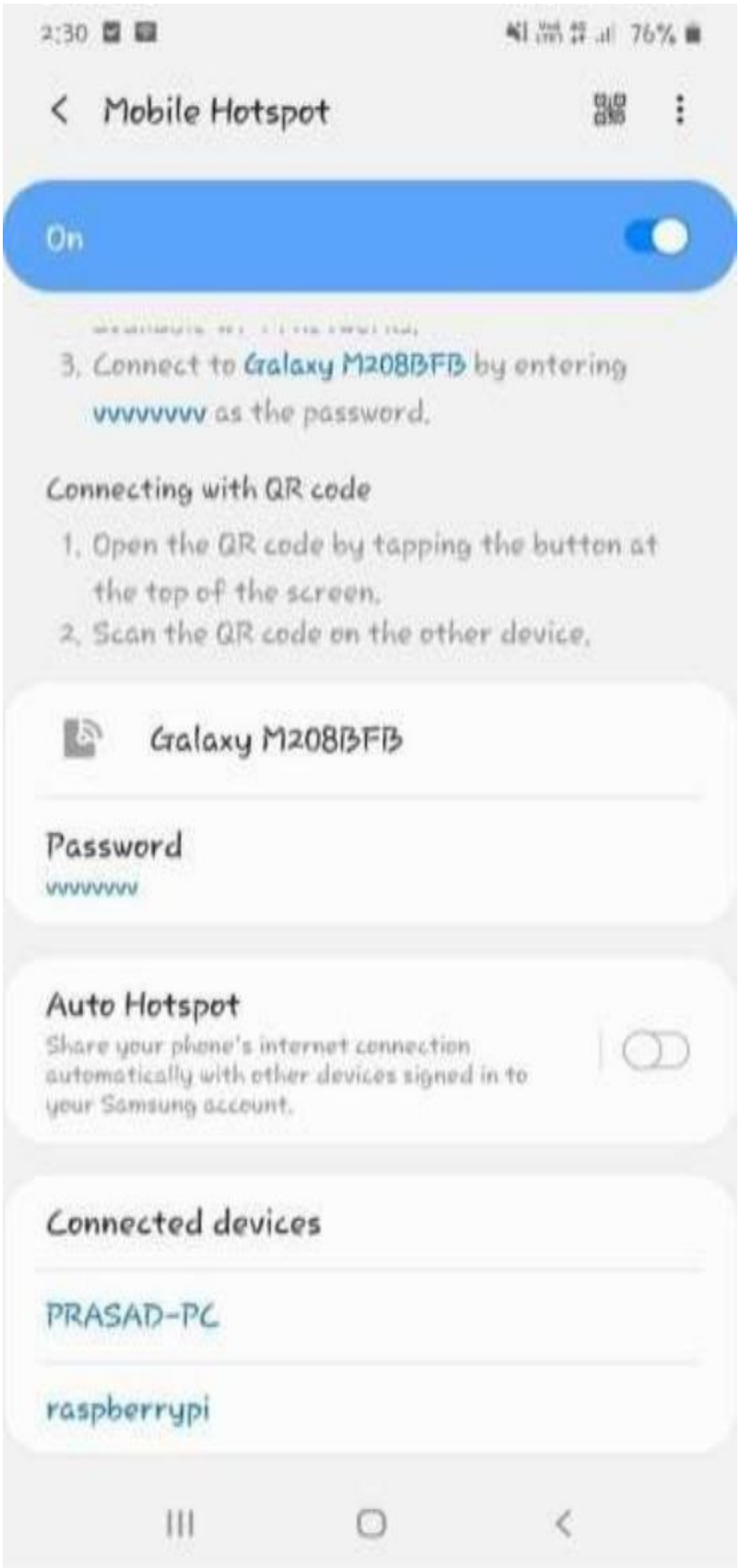
STEP VIII : Click on yes

.

STEP IX : Verifying raspberry pi .

STEP X : Click on continue .

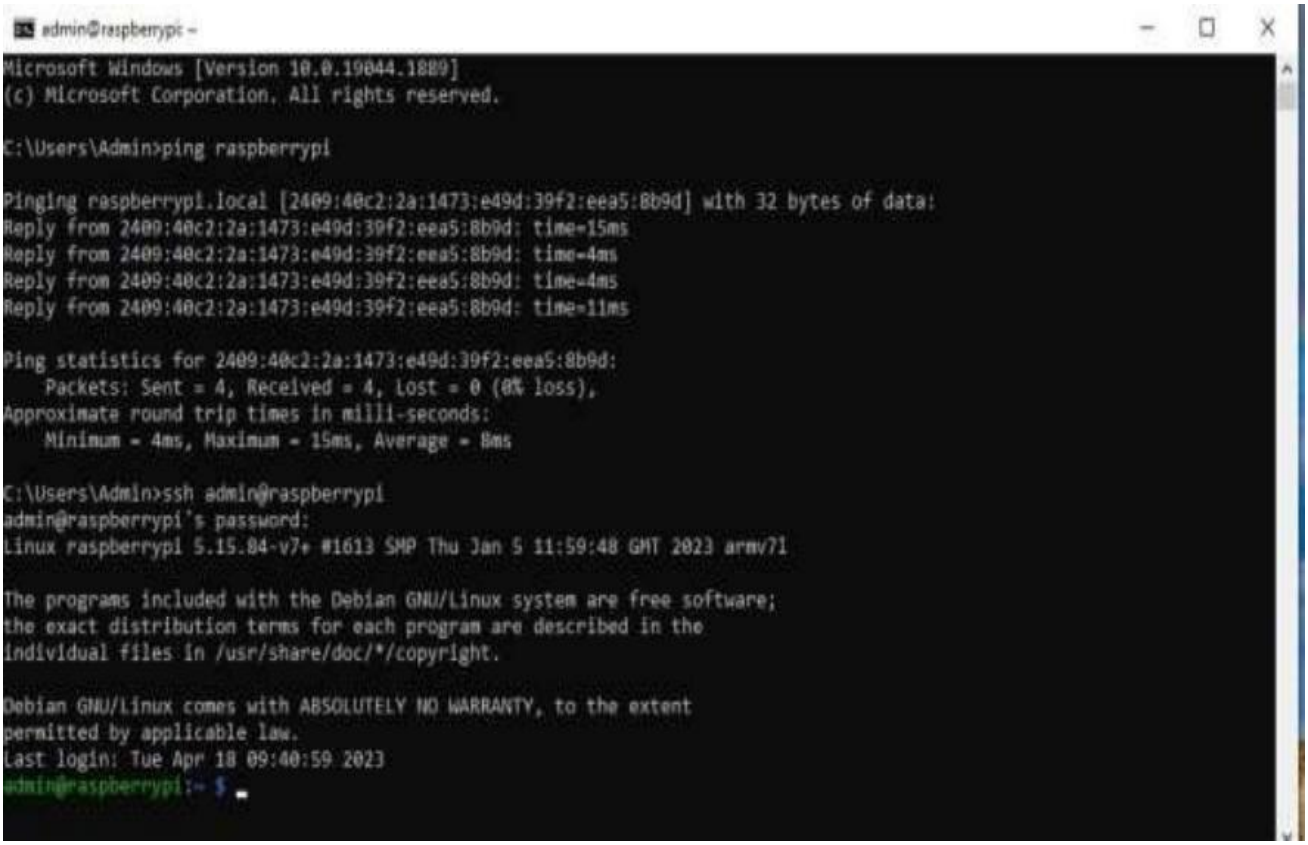



2:30
76%
Mobile Hotspot
On
3. Connect to Galaxy M208BFB by entering vvvvvvvv as the password.
Connecting with QR code
1. Open the QR code by tapping the button at the top of the screen.
2. Scan the QR code on the other device.
Galaxy M208BFB
Password
vvvvvvvv
Auto Hotspot
Share your phone's internet connection automatically with other devices signed in to your Samsung account.
Connected devices
PRASAD-PC
raspberrypi

STEP XI : check the connection in your mobile hotspot of raspberrypi

STEP XII: On cmd prompt execute this command :-

1) ping raspberrypi
2) ssh admin@raspberr

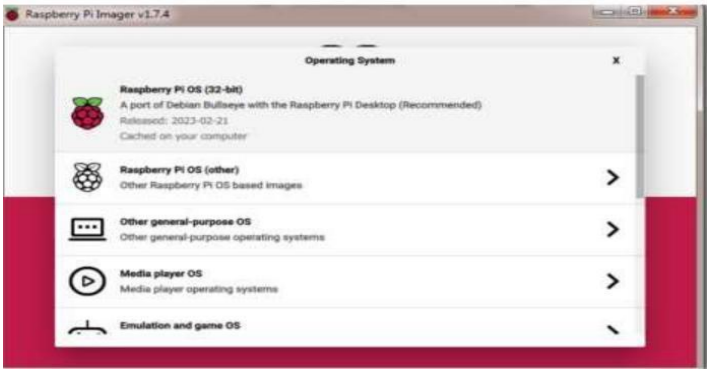


Practical No:- 2

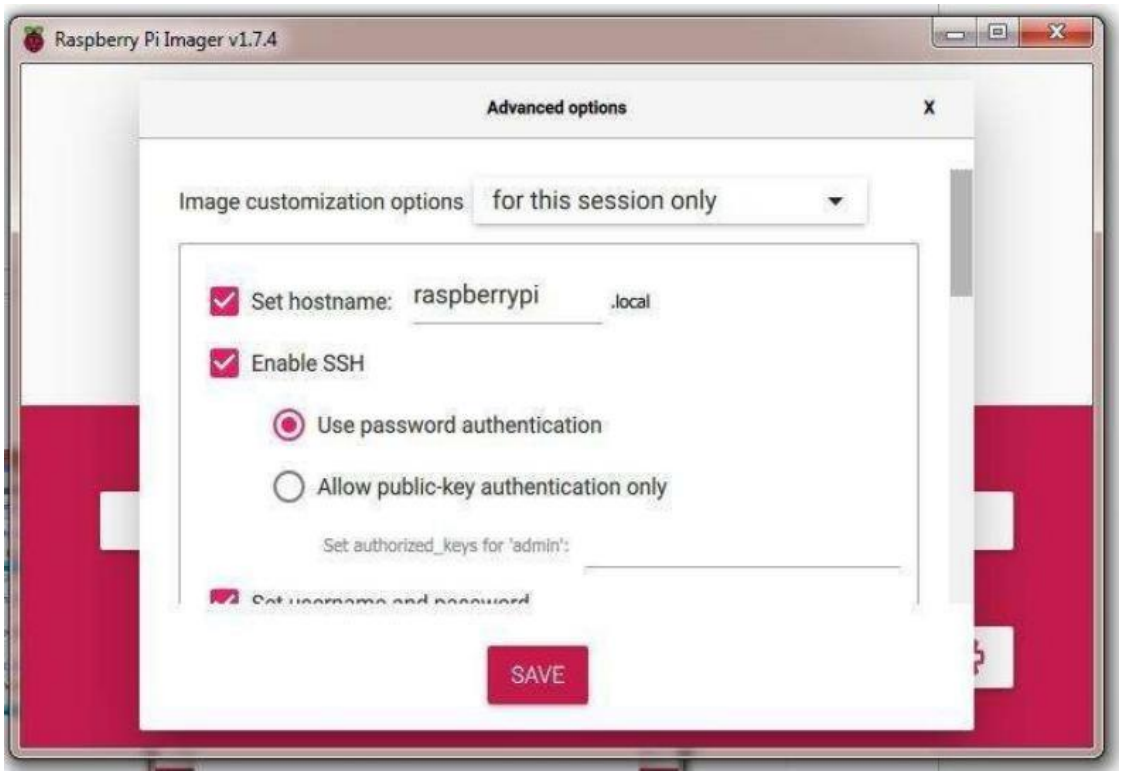
Aim: - Using sftp upload files from PC.

Software required:- Filezilla, GitHub

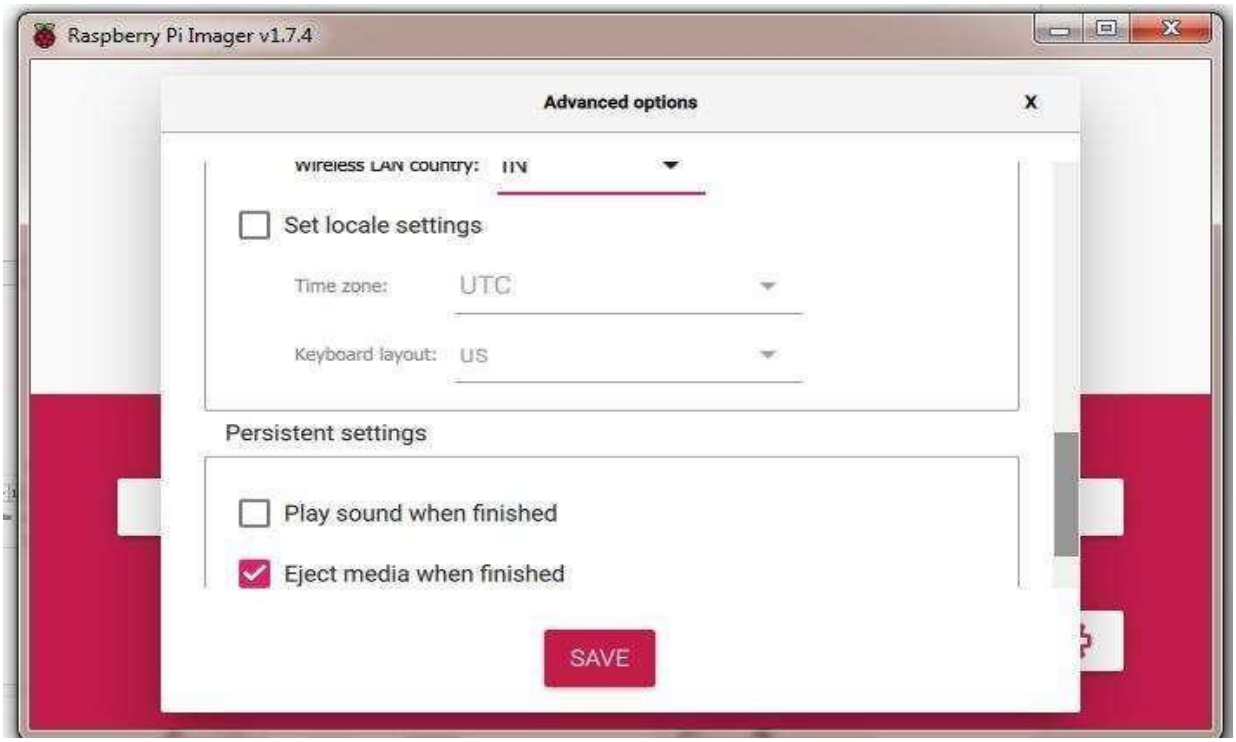
Step 1: Install the Raspberry Pi Imager

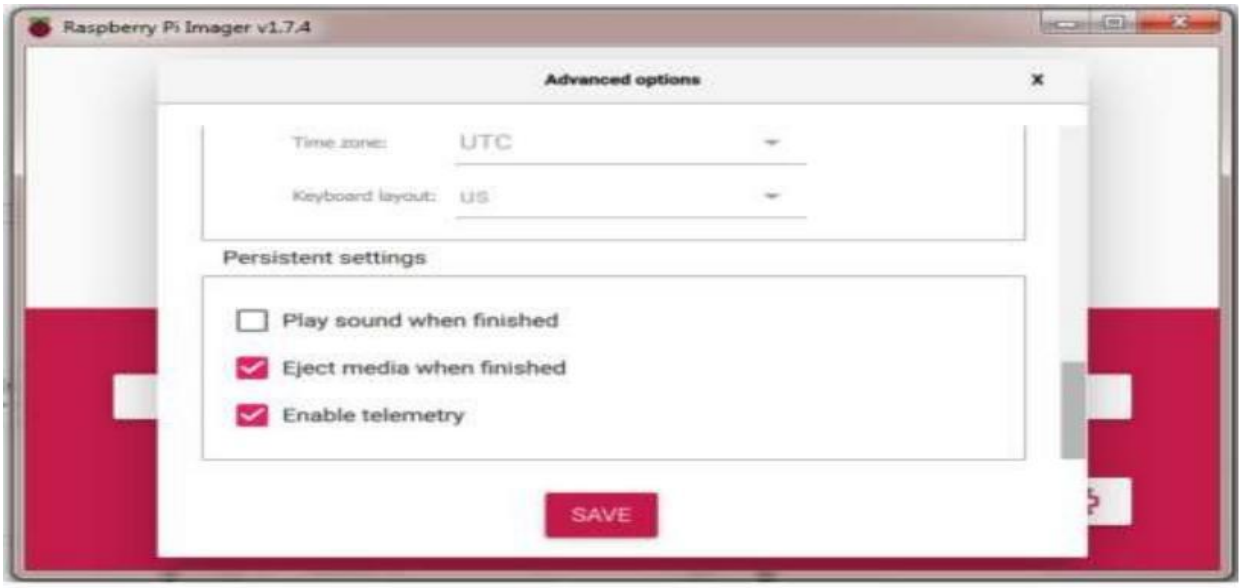


Step 2: Set hostname, enable SSH, Set Username and Password.



Step 3: Set SSID and Password of hotspot which is used.

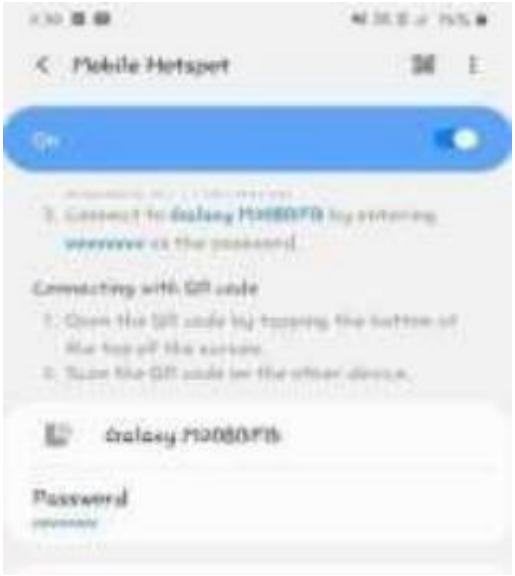

Raspberry Pi Imager v1.7.4
Advanced options
Wireless LAN country: IN
Set locale settings
Time zone: UTC
Keyboard layout: US
Persistent settings
Play sound when finished
Eject media when finished
SAVE


Raspberry Pi Imager v1.7.4
Advanced options
Time zone: UTC
Keyboard layout: US
Persistent settings
Play sound when finished
Eject media when finished
Enable telemetry
SAVE

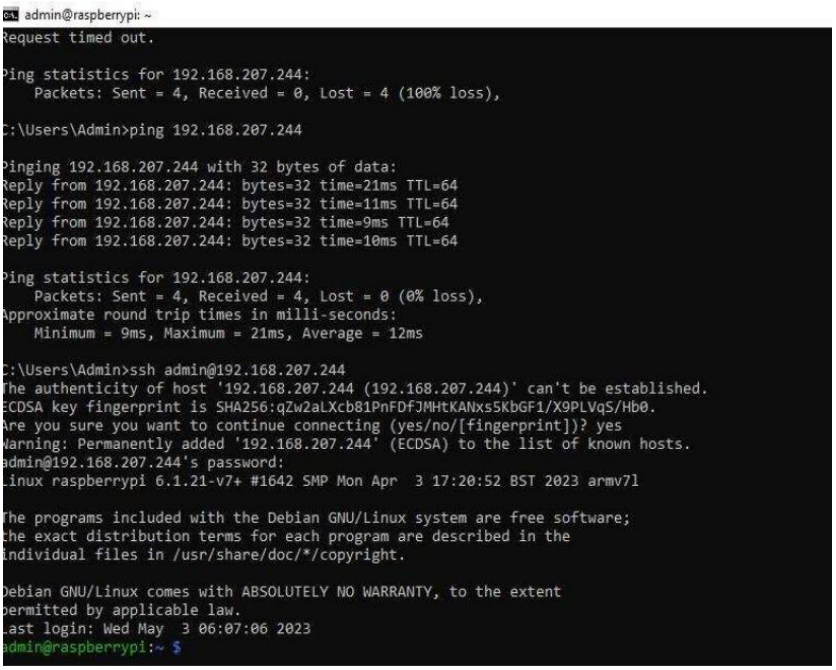

Raspberry Pi Imager v1.7.4
Raspberry Pi

Step 4: Connect Raspberry Pi WIFI and Laptop WIFI to Mobile Device.

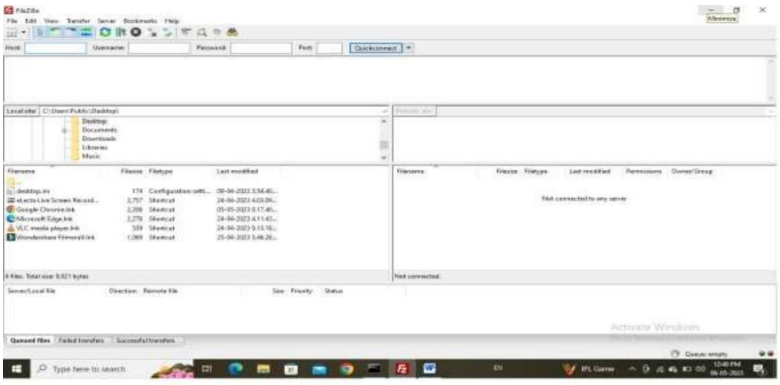
Step 5: Open CMD and type following command.

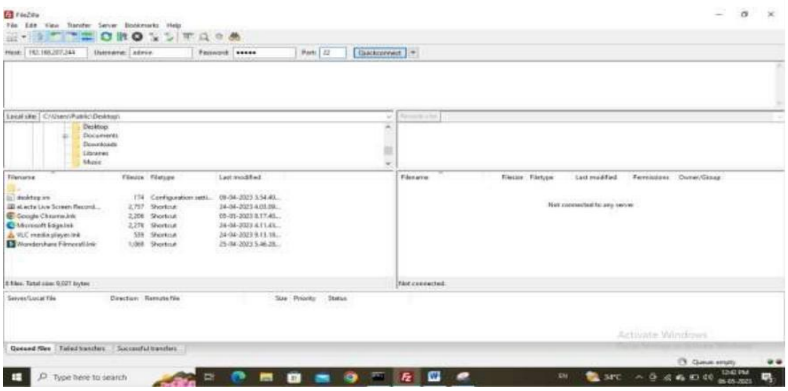


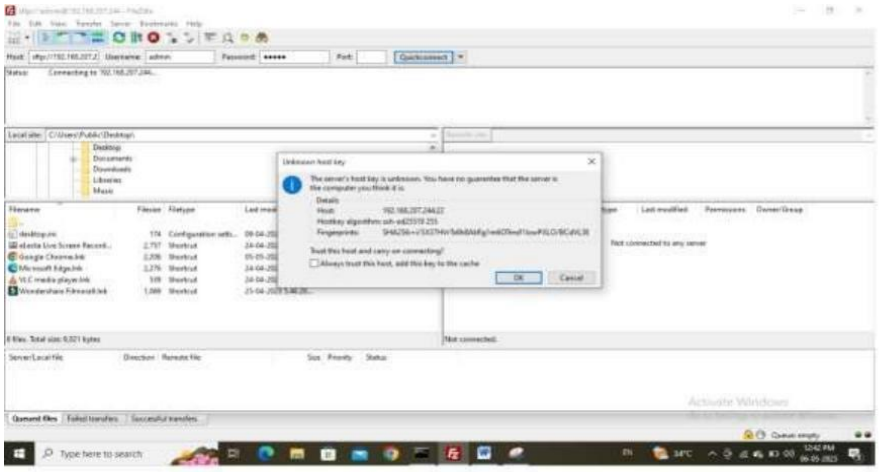
I) ping raspberrypi or ping 162.168.207.244
II) ssh admin@ raspberrypi or ssh admin@ 162.168.207.244 And type password of Admin.

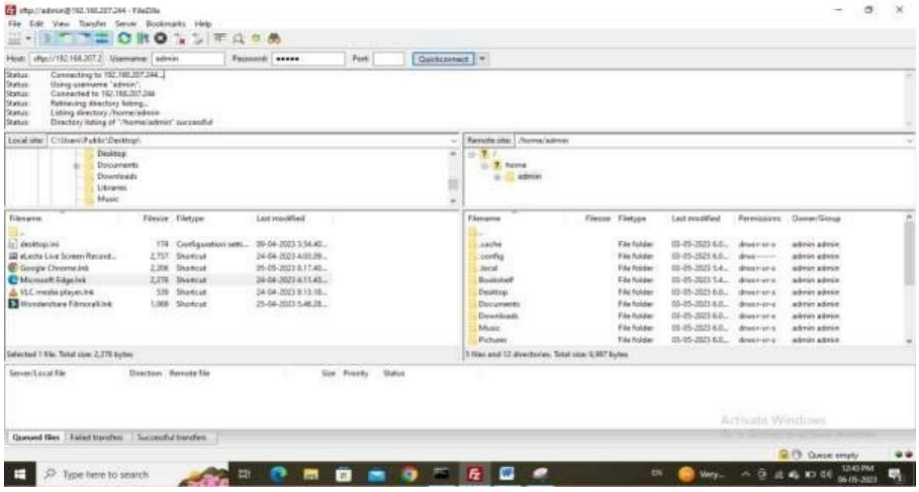


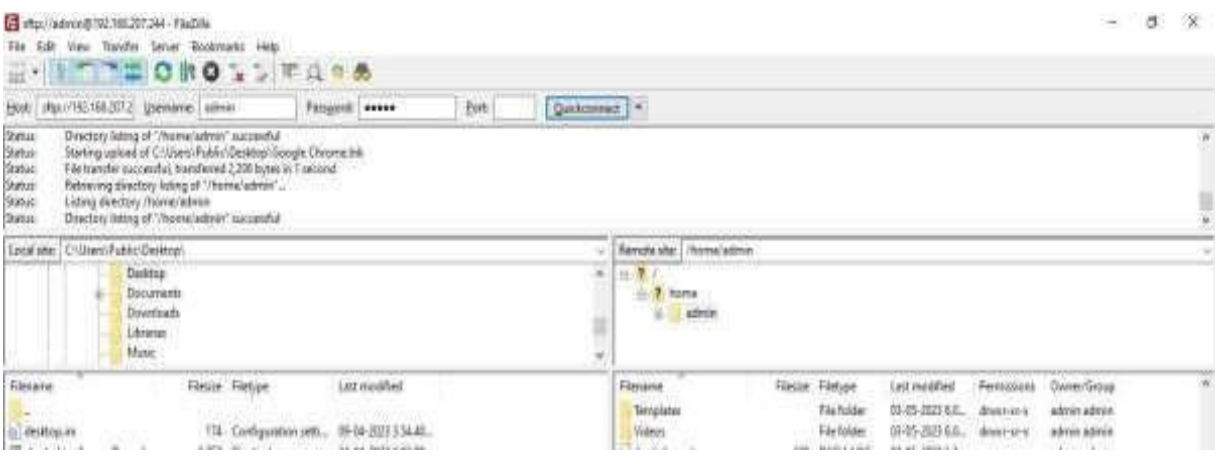
Step 6: Download the FileZilla (Client).











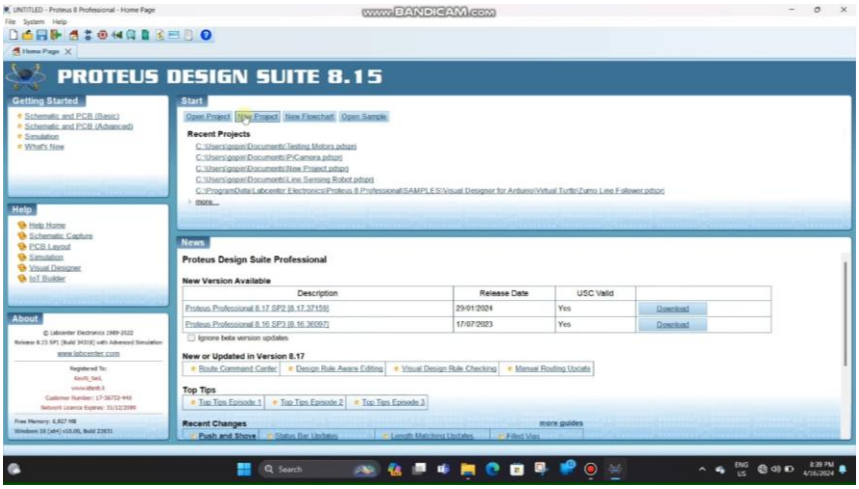
Practical No.3

Aim : Write Python code to test motors.

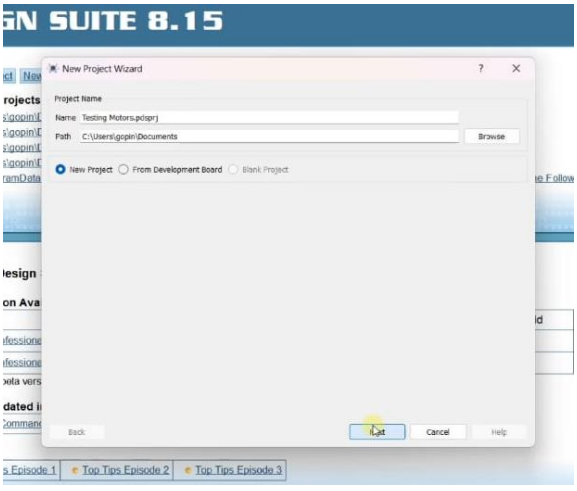
Components:-

- **Raspberry pi Board3**
- **L293D**
- **Simple DC Motor**
- **Button**

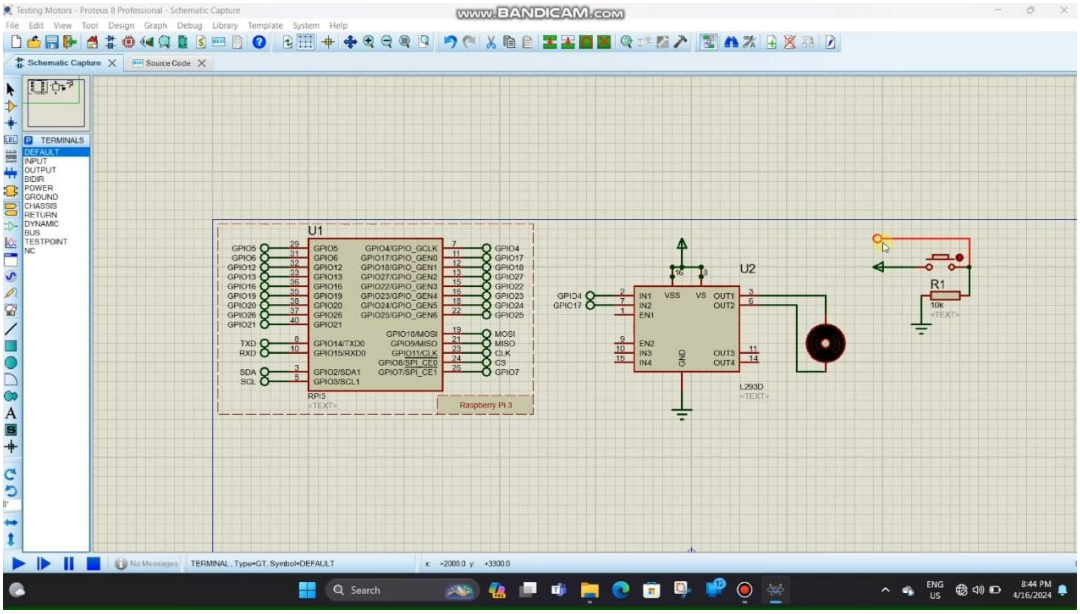
Step 1 : Goto new project and select new project



Step 2 :Change the name of the project and save as Testing Motors.psdprj



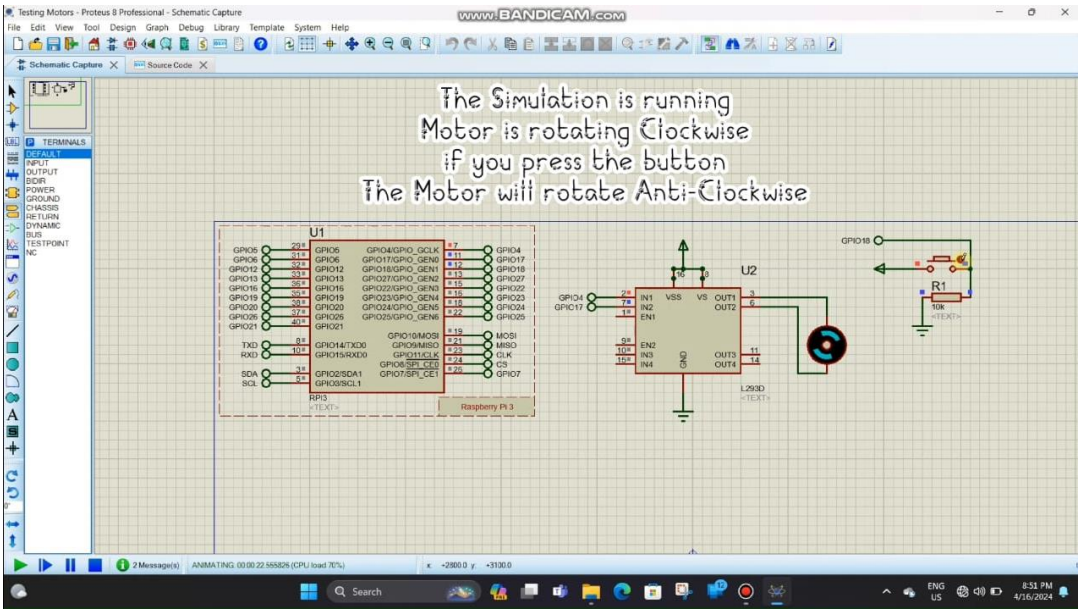
Step 3 : Implementation of the Circuit

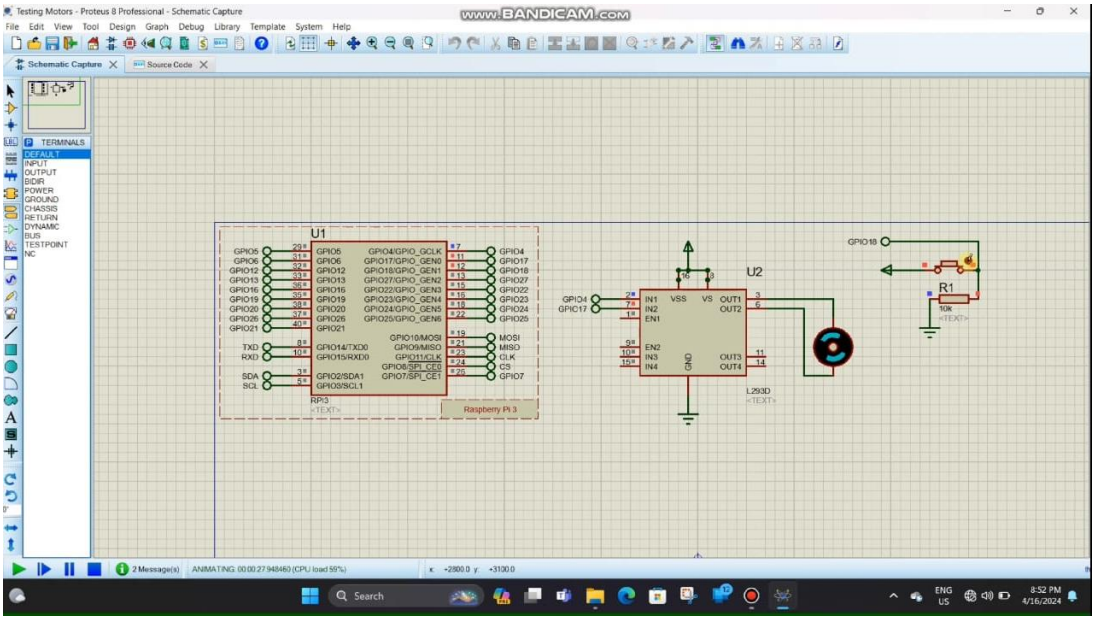


Source Code :

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

button = 12
DC_motor_a = 7
DC_motor_b = 11

GPIO.setup(DC_motor_a, GPIO.OUT)
GPIO.setup(DC_motor_b, GPIO.OUT)
GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_UP)

try:
    while True:
        if GPIO.input(button) == GPIO.LOW:
            GPIO.output(DC_motor_a, GPIO.HIGH)
            GPIO.output(DC_motor_b, GPIO.LOW)
        else:
            GPIO.output(DC_motor_a, GPIO.LOW)
            GPIO.output(DC_motor_b, GPIO.HIGH)
        time.sleep(0.1)
```

Output :


The Simulation is running
Motor is rotating Clockwise
if you press the button
The Motor will rotate Anti-Clockwise



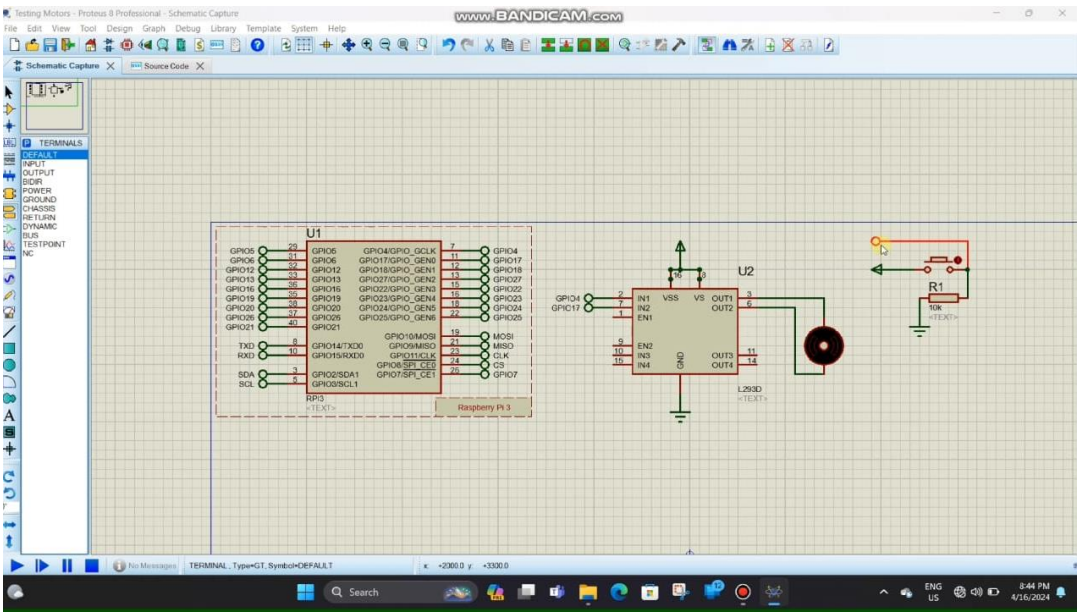
Practical No.4

Aim : Write a script to follow a predetermined path.

Components:-

- **Raspberry pi Board3**
- **L293D**
- **Simple DC Motor**
- **Button**

Implementation of the Circuit

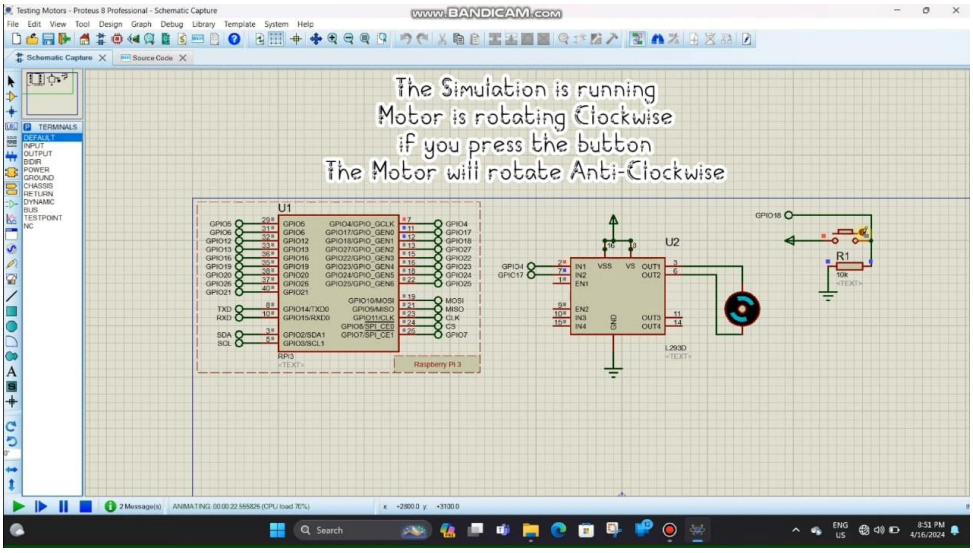


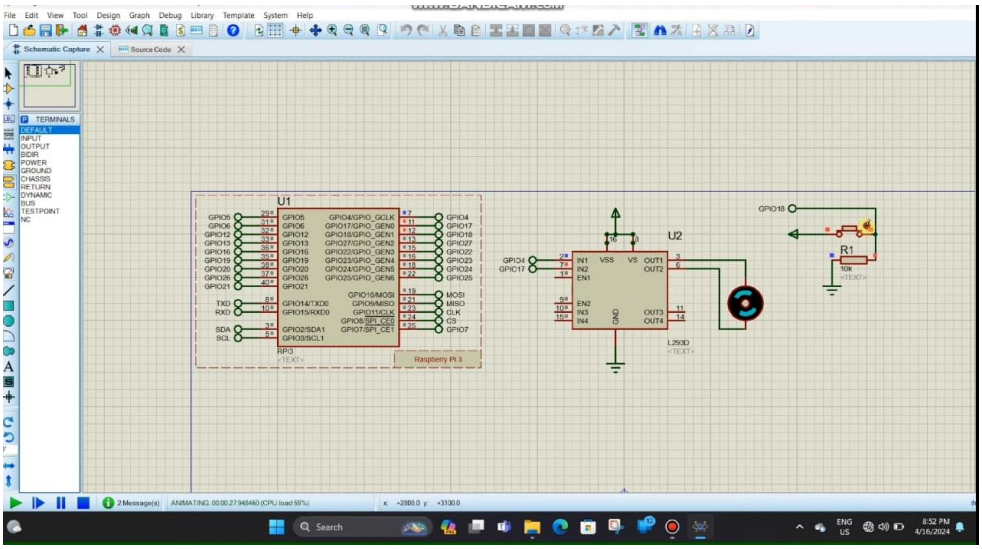
Source Code :

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

button = 12
DC_motor_a = 7
DC_motor_b = 11

GPIO.setup(DC_motor_a, GPIO.OUT)
GPIO.setup(DC_motor_b, GPIO.OUT)
GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

```
try:
    while True:
        if GPIO.input(button) == GPIO.LOW:
            GPIO.output(DC_motor_a, GPIO.HIGH)
            GPIO.output(DC_motor_b, GPIO.LOW)
        else:
            GPIO.output(DC_motor_a, GPIO.LOW)
            GPIO.output(DC_motor_b, GPIO.HIGH)
        time.sleep(0.1)
```

Output :





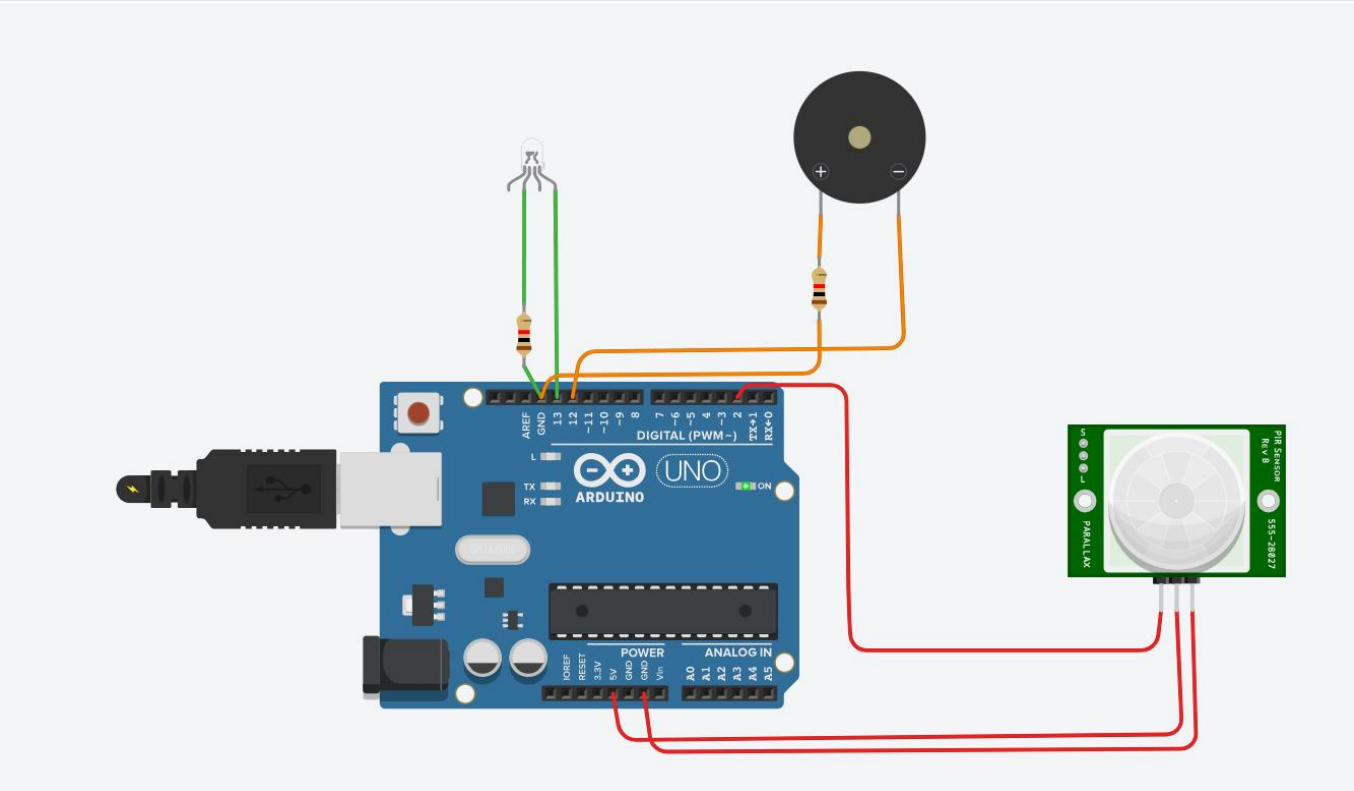
Practical No.5

Aim : Develop Python code for testing the sensors.

Components:-

- **PIR Sensor**
- **Resistor**
- **Piezo**
- **Arduino Uno R3**
- **LED RGB**

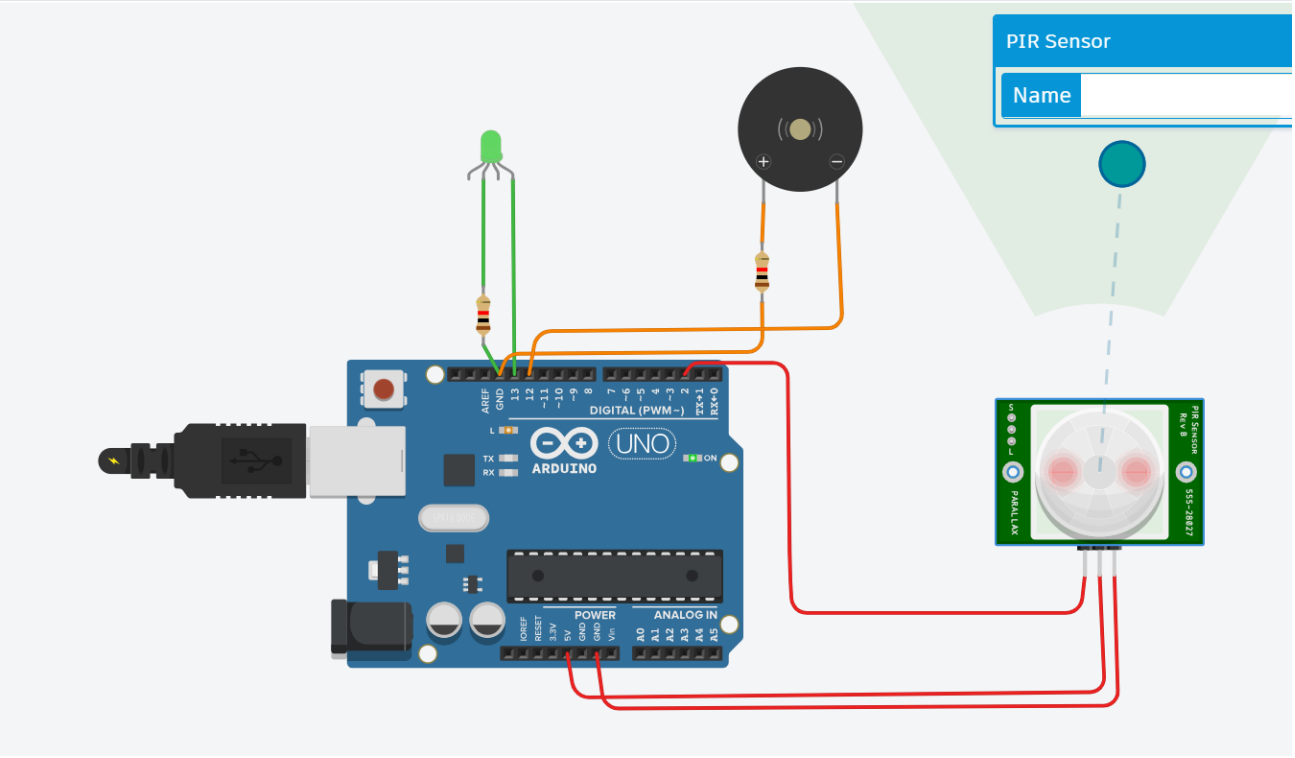
Step 1 : Place the Components in Tinker Cad.



Source Code

```
int pirsensor = 0;
void setup()
{
  pinMode(2, INPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
void loop()
{
  pirsensor = digitalRead(2);
  if(pirsensor == HIGH)
  {
    digitalWrite(13, HIGH);
    tone(12,1000,1000);
  }
  digitalWrite(13, LOW);
}
```

Output :



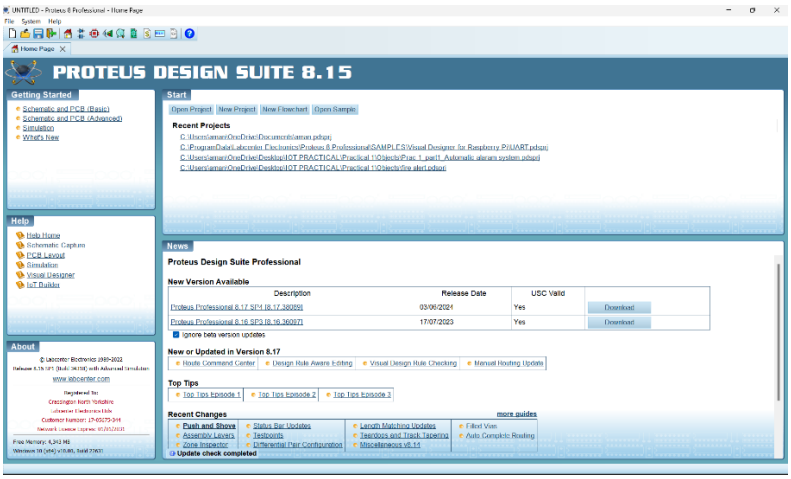
Practical No.6

Aim : Add the sensors to the Robot object and develop the line-following behavior code.

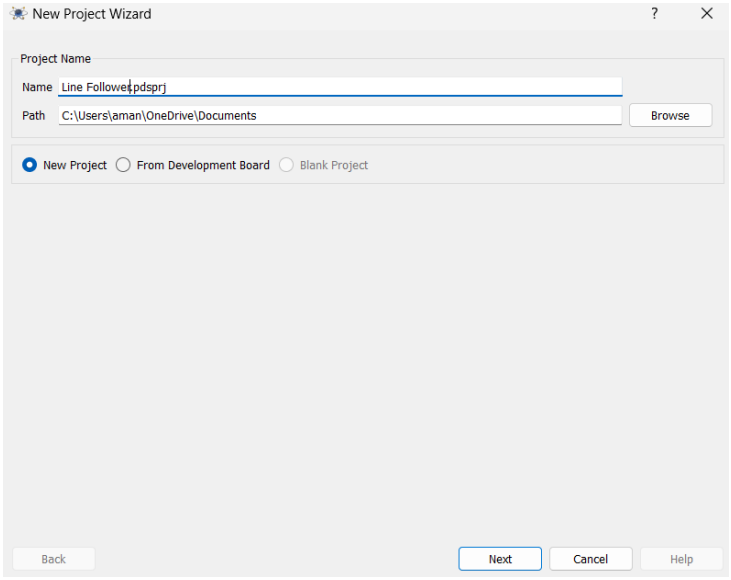
Components:-

- **Arduino**
- **Zumo Robot**
- **Button**
-

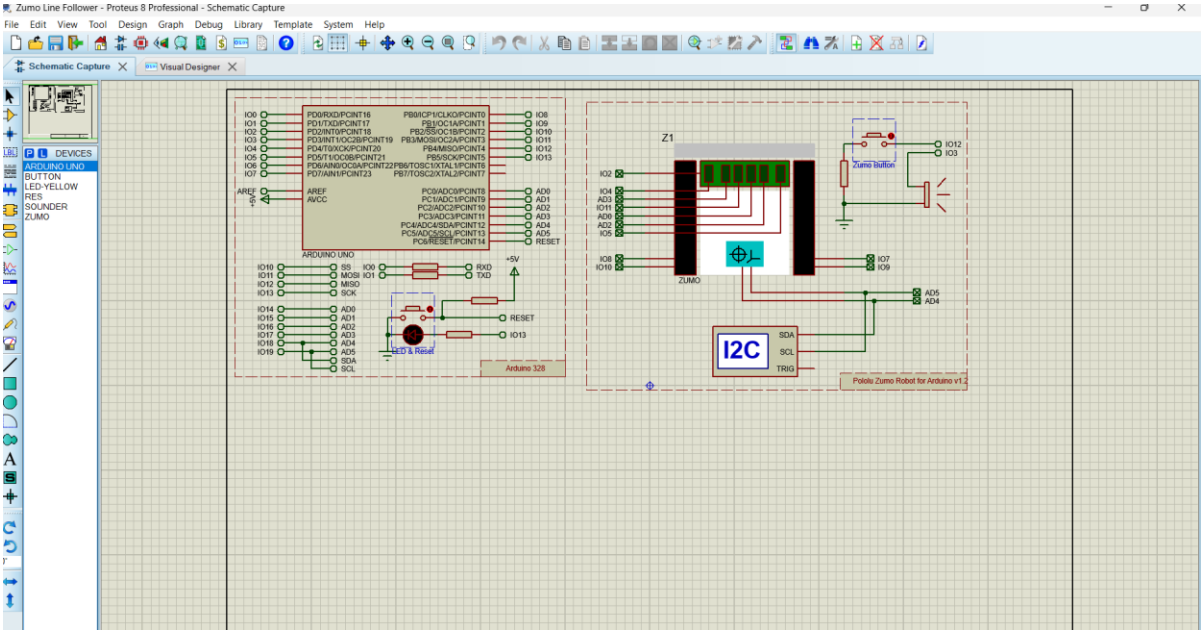
Step 1 : Goto new project and select new project

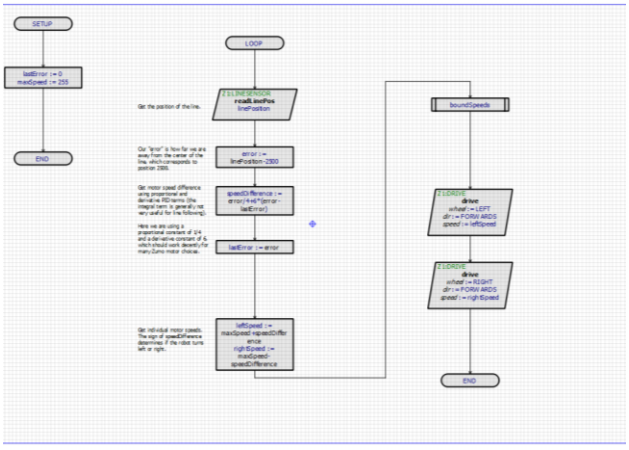


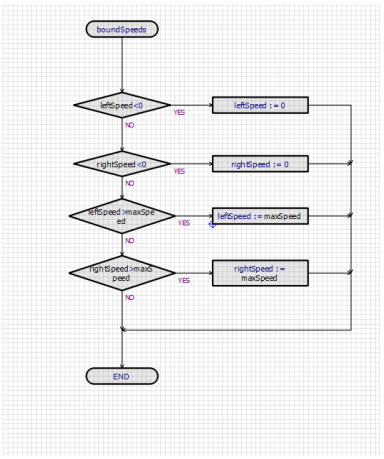
Step 2 :Change the name of the project and save as Line Follower.psdprj

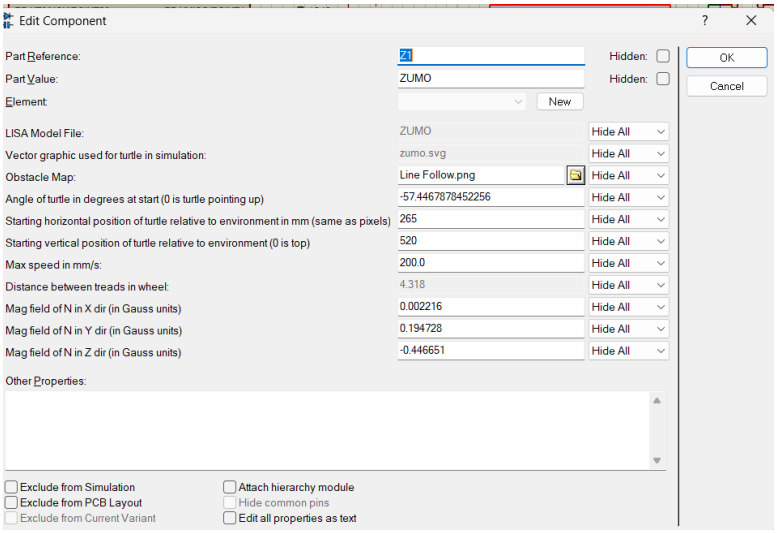


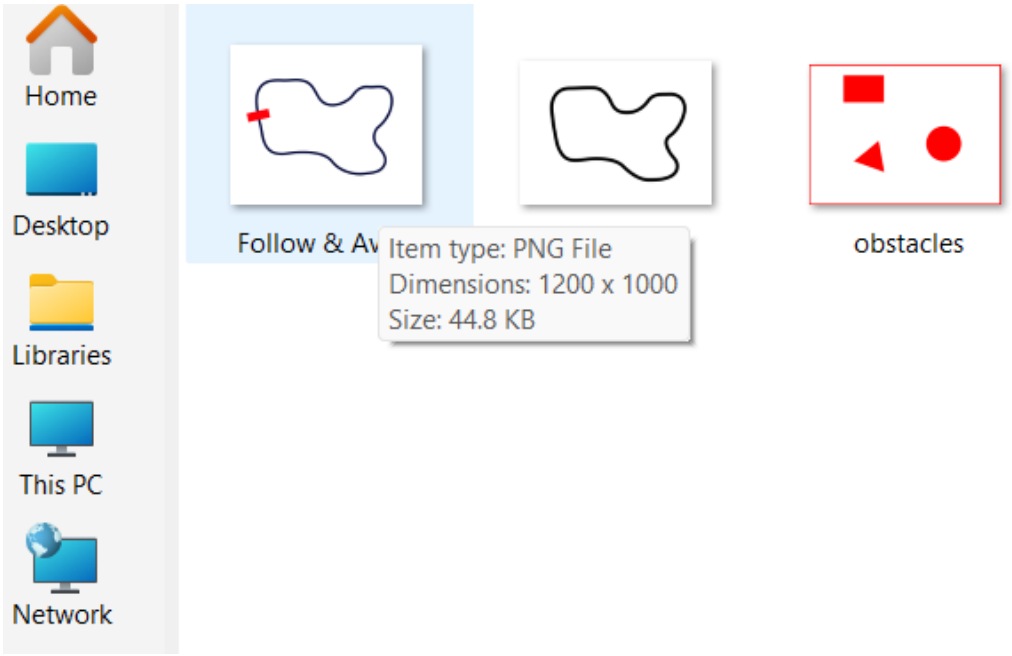
Step 3 : Implementation of the Circuit











Source Code:

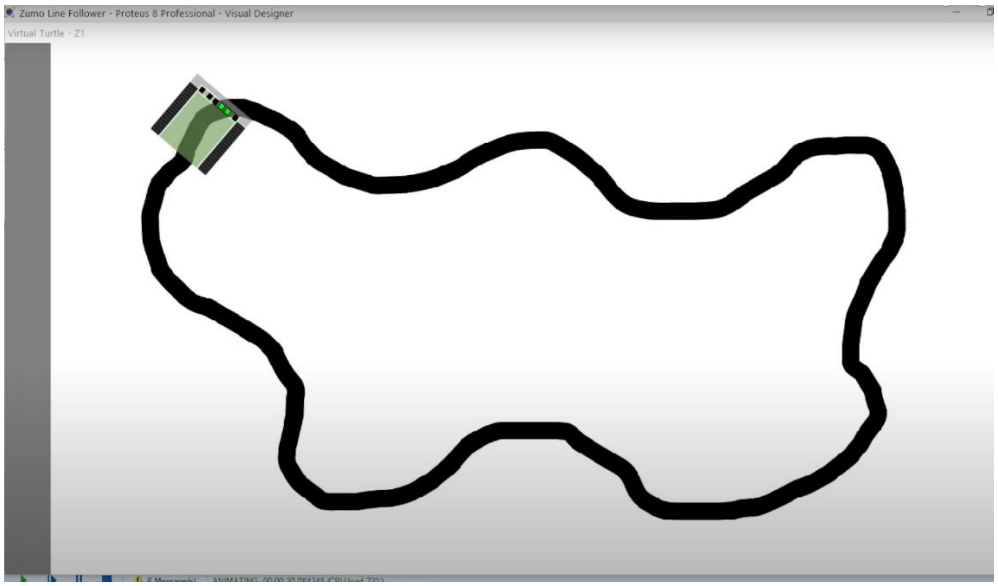
```
#pragma GCC push_options
#pragma GCC optimize ("Os")
#include <core.h>
#include <cpu.h>
#include <TimerOne.h>
#include <L3G.h>
#include <LSM303.h>
#include <Wire.h>
#include <Servo.h>
#include <Zumo.h>
#pragma GCC pop_options


CPU &cpu = Cpu;
TimerOne &timer1 = Timer1;
DRIVE Z1_DRIVE = DRIVE(8, 10, 7, 9);
```

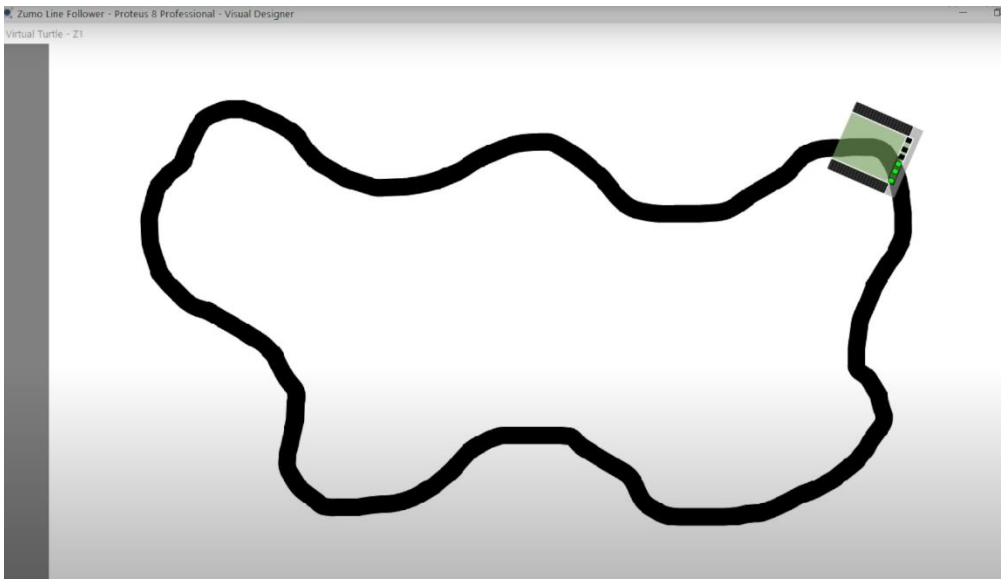
```
LINESENSOR Z1_LINESENSOR = LINESENSOR(4, A3, 11, A0, A2, 5, 2);
COMPASS Z1_COMPASS = COMPASS();
GYRO Z1_GYRO = GYRO();

void peripheral_setup() {
   Z1_DRIVE.begin();
   Z1_LINESENSOR.begin();
   Z1_COMPASS.begin();
   Z1_GYRO.begin();
}

void peripheral_loop() {
}

long var_linePosition;
long var_error;
long var_lastError;
long var_speedDifference;
long var_leftSpeed;
long var_rightSpeed;
long var_maxSpeed;
float var_magX;
float var_magY;
float var_magZ;

void chart_SETUP() {
   var_lastError = 0;
   var_maxSpeed = 255;
}
```

```
void chart_LOOP() {
    var_linePosition = Z1_LINESENSOR.readLinePos();
    var_error = var_linePosition - 2500;
    var_speedDifference = var_error / 4 + 6 * (var_error - var_lastError);
    var_lastError = var_error;
    var_leftSpeed = var_maxSpeed + var_speedDifference;
    var_rightSpeed = var_maxSpeed - var_speedDifference;
    chart_boundSpeeds();
    Z1_DRIVE.drive(1, 1, var_leftSpeed);
    Z1_DRIVE.drive(2, 1, var_rightSpeed);
}

void chart_boundSpeeds() {
    if (var_leftSpeed < 0) {
        var_leftSpeed = 0;
    } else if (var_rightSpeed < 0) {
        var_rightSpeed = 0;
    } else if (var_leftSpeed > var_maxSpeed) {
        var_leftSpeed = var_maxSpeed;
    } else if (var_rightSpeed > var_maxSpeed) {
        var_rightSpeed = var_maxSpeed;
    }
}

void setup() {
    peripheral_setup();
    chart_SETUP();
}

void loop() {
```

```
    peripheral_loop();
    chart_LOOP();
}
```

Output :





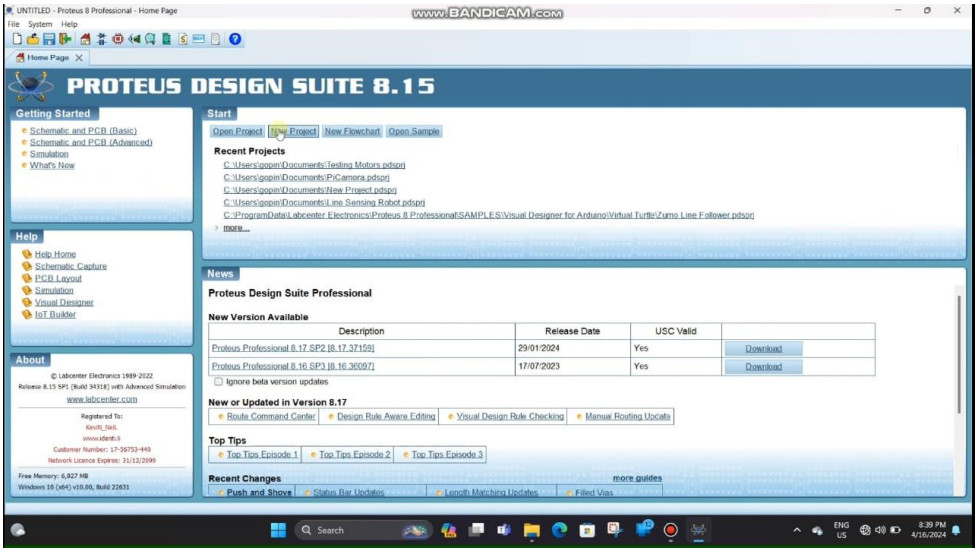
Practical No.7

Aim : Using the light strip develop and debug the line follower robot.

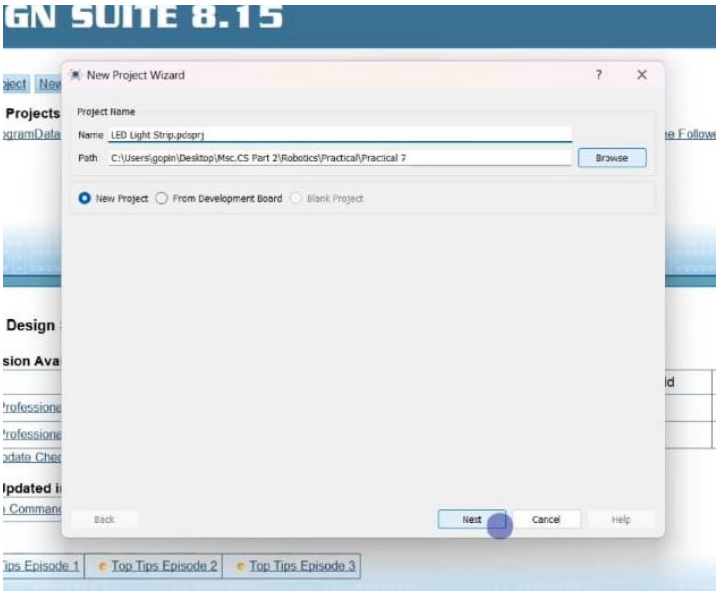
Components:-

- **Raspberry pi Board3**
- **RGB LED Strip**

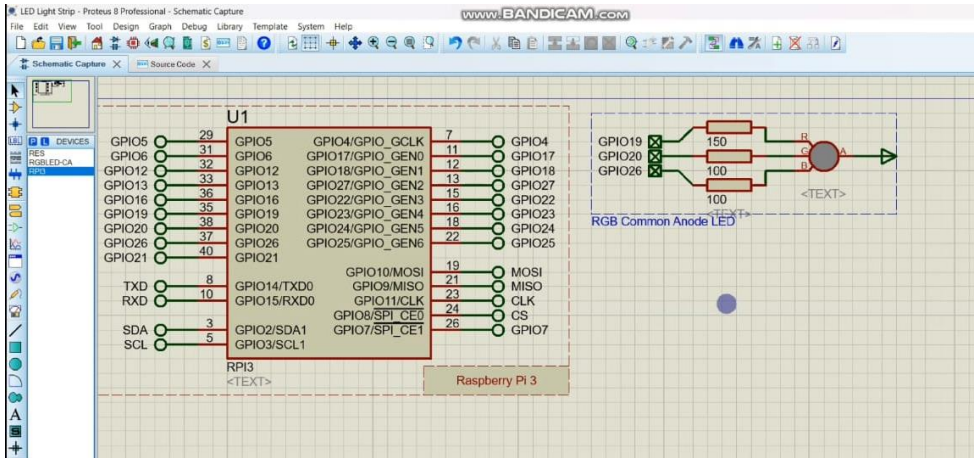
Step 1 : Goto new project and select new project



Step 2 :Change the name of the project and save as LED Light Strip.psdprj



Step 3 : Implementation of the Circuit

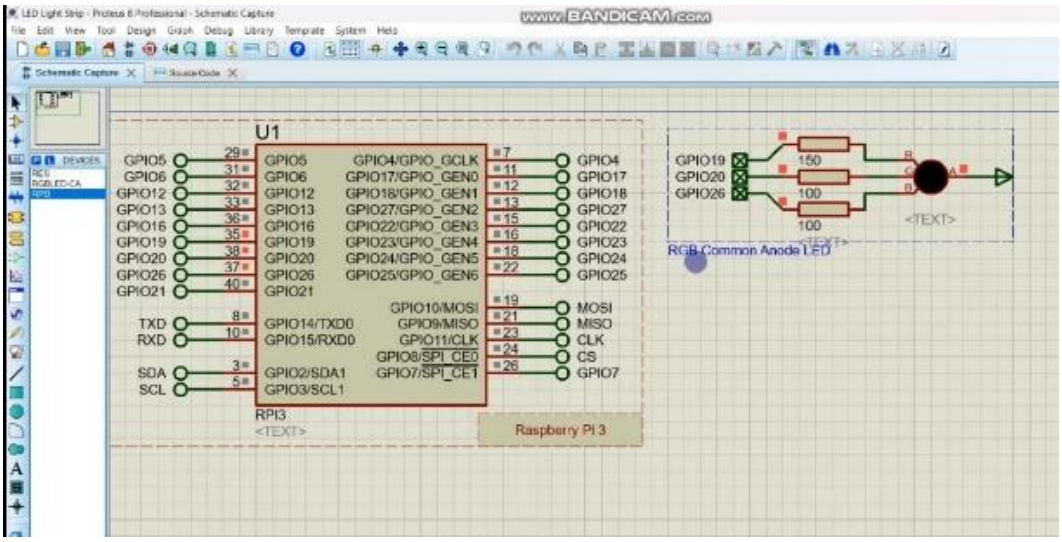


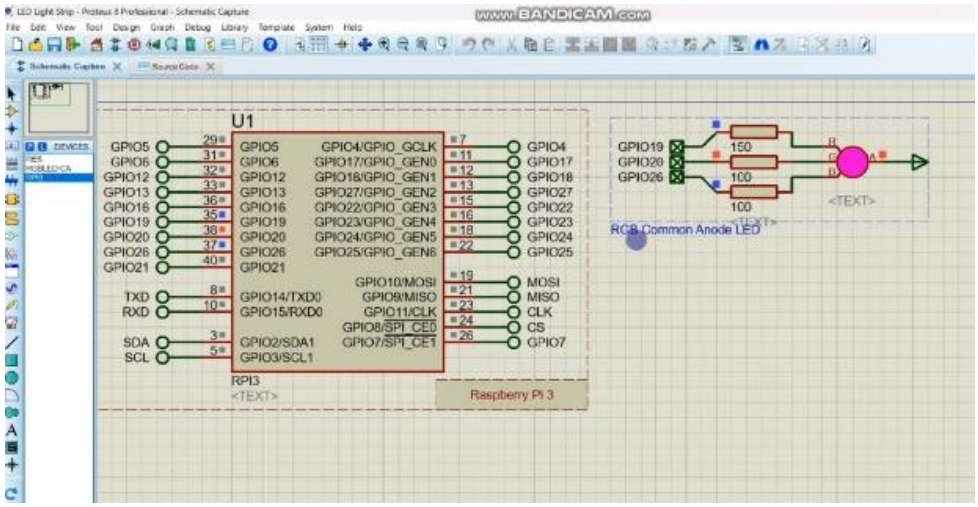
Source Code :

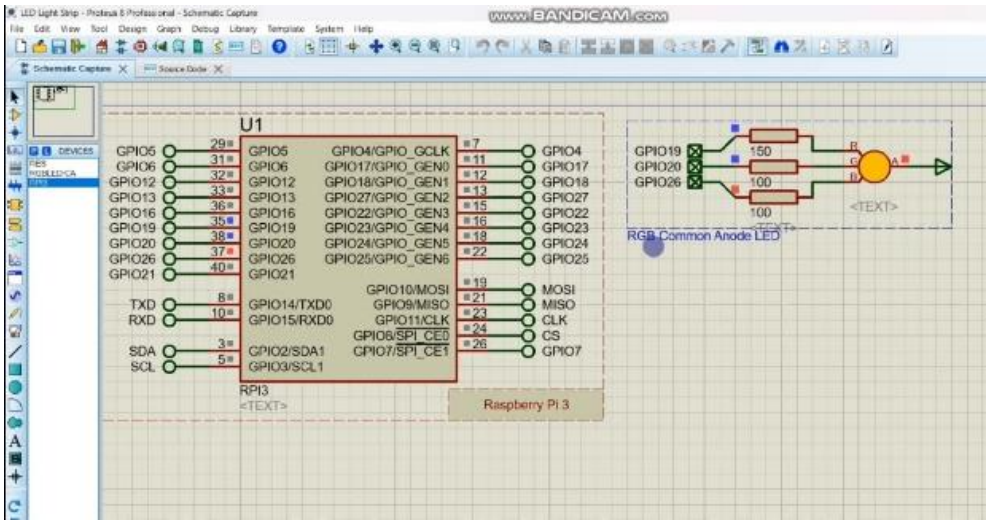
```
from goto import with_goto
from datetime import datetime
import time
import cpu
import FileStore
import timer
import VFP
import Generic

def peripheral_setup():
    pio.cpu = cpu.CPU()
    pio.storage = FileStore.FileStore()
    pio.timer = timer.Timer()
    pio.server = VFP.VfpServer()
    pio.RGBLED1 = Generic.RgbLedCa(pio.GPIO19, pio.GPIO20, pio.GPIO26)
    pio.storage.begin()
    pio.server.begin(0)

def peripheral_loop():
    pio.timer.poll()
    pio.server.poll()

def variables_setup():
    pass

@with_goto
def chart_SETUP():
    return

@with_goto
def chart_LOOP():
    pio.RGBLED1.set(True, True, True)
    time.sleep(0.5)
```

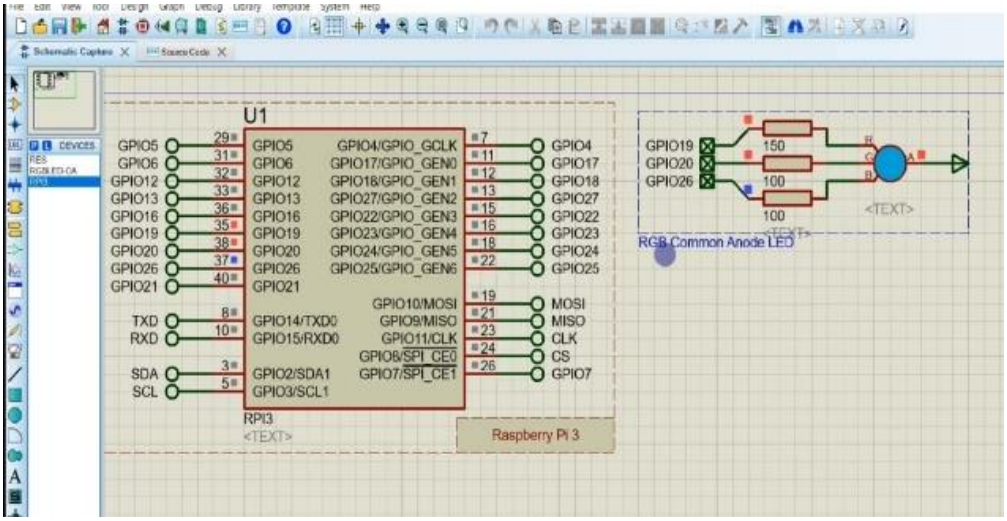
```
    pio.RGBLED1.set(True, False, False)
    time.sleep(0.5)
    pio.RGBLED1.set(True, True, False)
    time.sleep(0.5)
    pio.RGBLED1.set(False, True, False)
    time.sleep(0.5)
    pio.RGBLED1.set(False, True, True)
    time.sleep(0.5)
    pio.RGBLED1.set(False, False, True)
    time.sleep(0.5)
    pio.RGBLED1.set(True, False, True)
    time.sleep(0.5)
    pio.RGBLED1.set(False, False, False)
    time.sleep(0.5)
    return

def main():
    variables_setup()
    peripheral_setup()
    chart_SETUP()
    while True:
        peripheral_loop()
        chart_LOOP()

if __name__ == '__main__':
    main()
```

Output :




www.BANDICAM.com
U1
GPIO5
GPIO6
GPIO12
GPIO13
GPIO16
GPIO19
GPIO20
GPIO26
GPIO21
TXD
RXD
SDA
SCL
GPIO4/GPIO_GCLK
GPIO17/GPIO_GEN0
GPIO18/GPIO_GEN1
GPIO27/GPIO_GEN2
GPIO22/GPIO_GEN3
GPIO23/GPIO_GEN4
GPIO24/GPIO_GEN5
GPIO25/GPIO_GEN6
GPIO10/MOSI
GPIO9/MISO
GPIO11/CLK
GPIO8/SPI_CE0
GPIO7/SPI_CE1
GPIO14/TXD0
GPIO15/RXD0
GPIO2/SDA1
GPIO3/SCL1
GPIO4
GPIO17
GPIO18
GPIO27
GPIO22
GPIO23
GPIO24
GPIO25
MOSI
MISO
CLK
CS
GPIO7
RPI3
Raspberry Pi 3
150
100
RGB Common Anode LED


www.BANDICAM.com
U1
RPI3
Raspberry Pi 3
RGB Common Anode LED


U1
RPI3
Raspberry Pi 3
RGB Common Anode LED

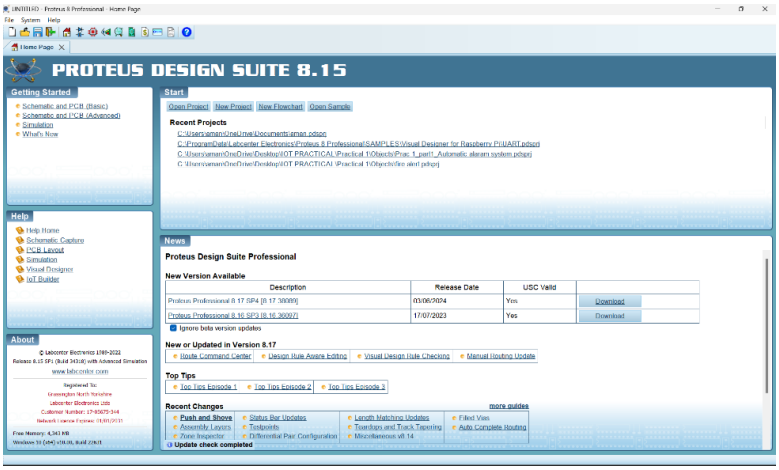
Practical No.8

Aim : Create an obstacle avoidance behavior for robot and test it.

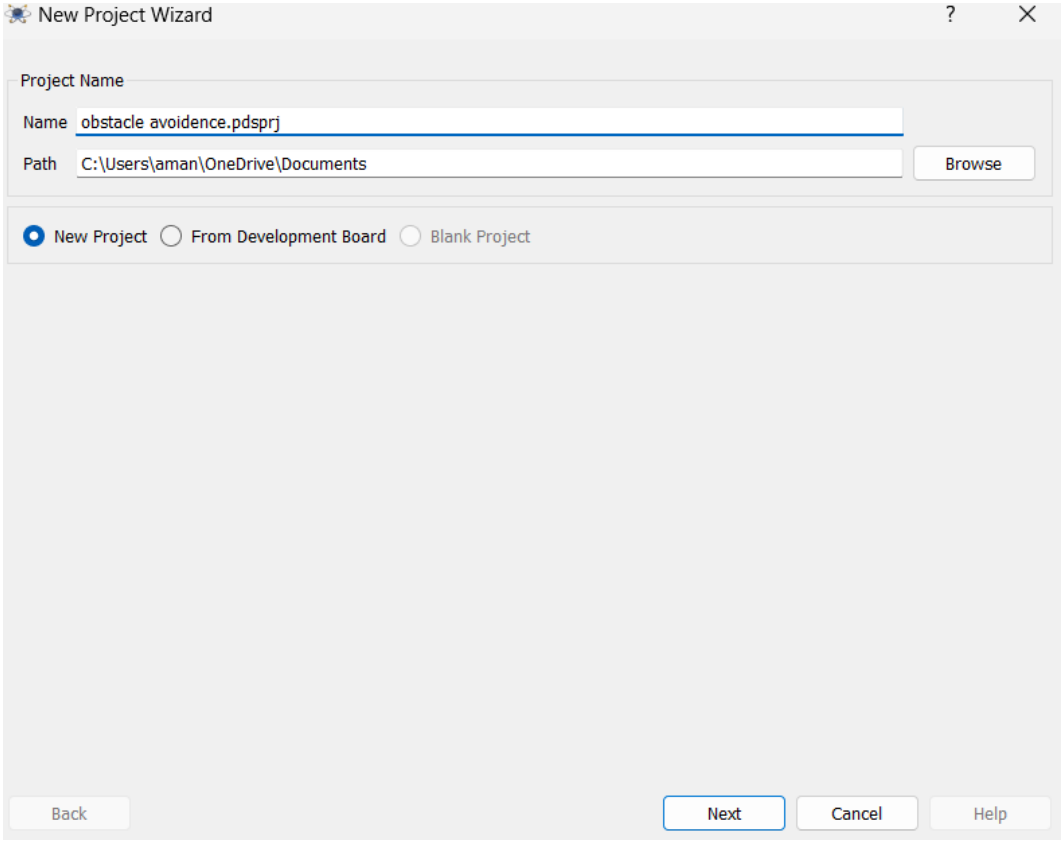
Components:-

- **Arduino Uno**
- **Zumo Robot**

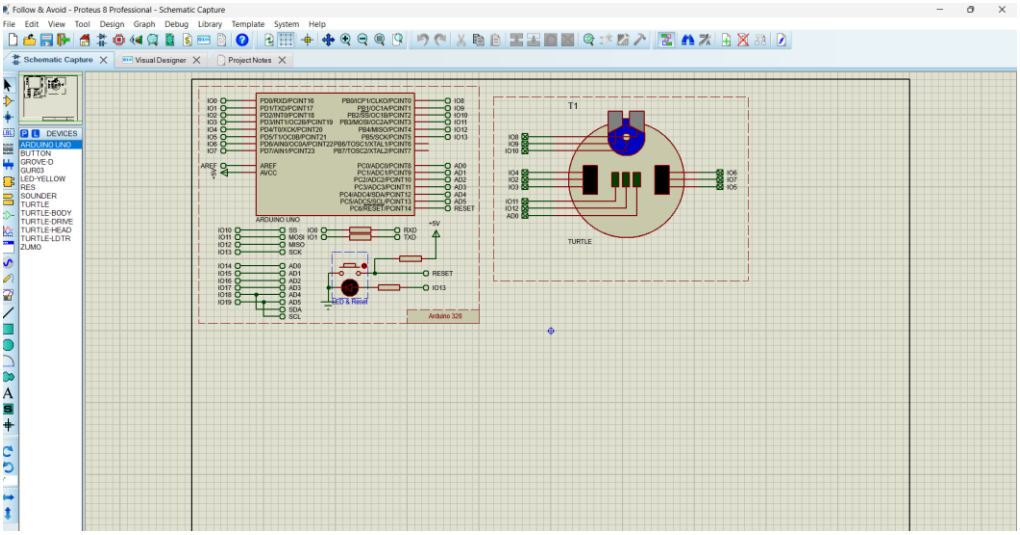
Step 1 : Goto new project and select new project

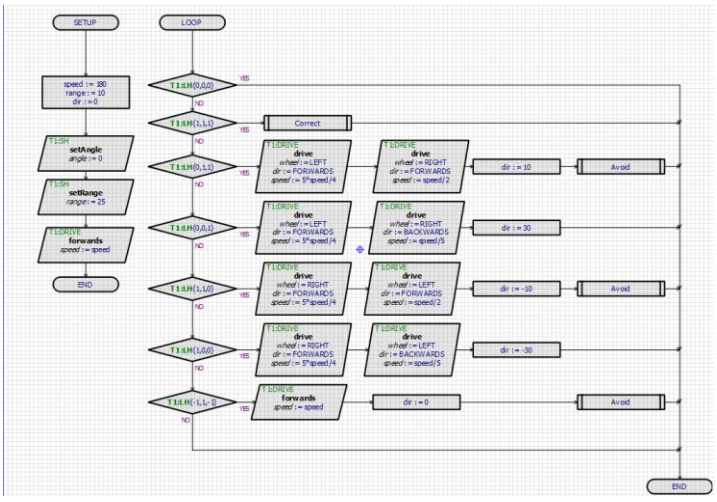


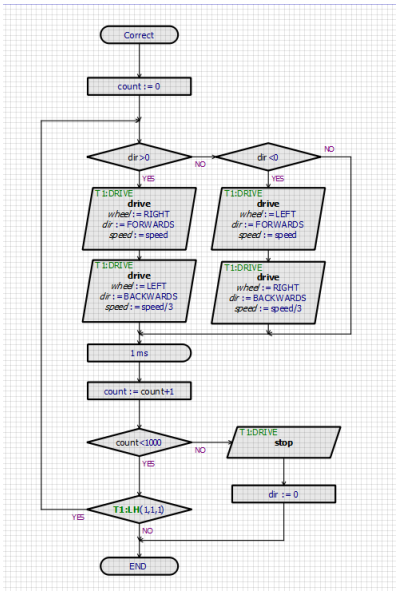
Step 2 :Change the name of the project and save as Follow & Avoid.psdprj

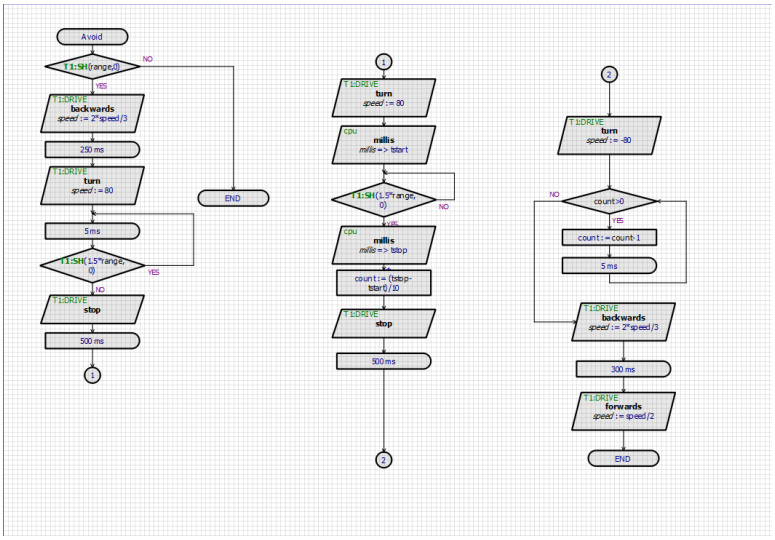


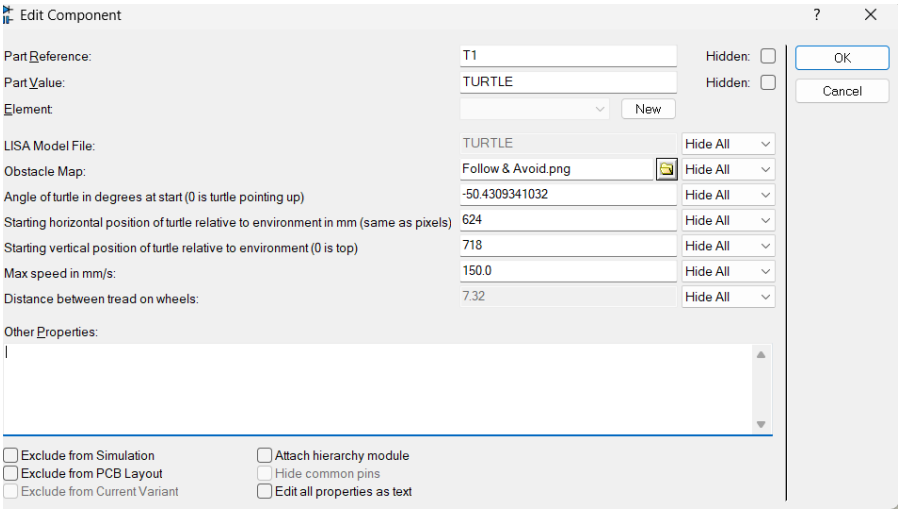
Step 3 : Implementation of the Circuit

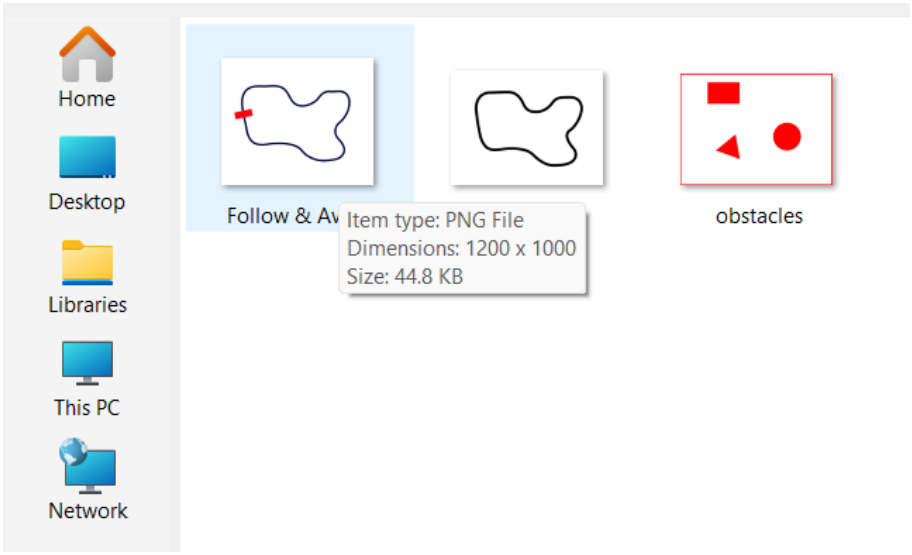












Source Code :

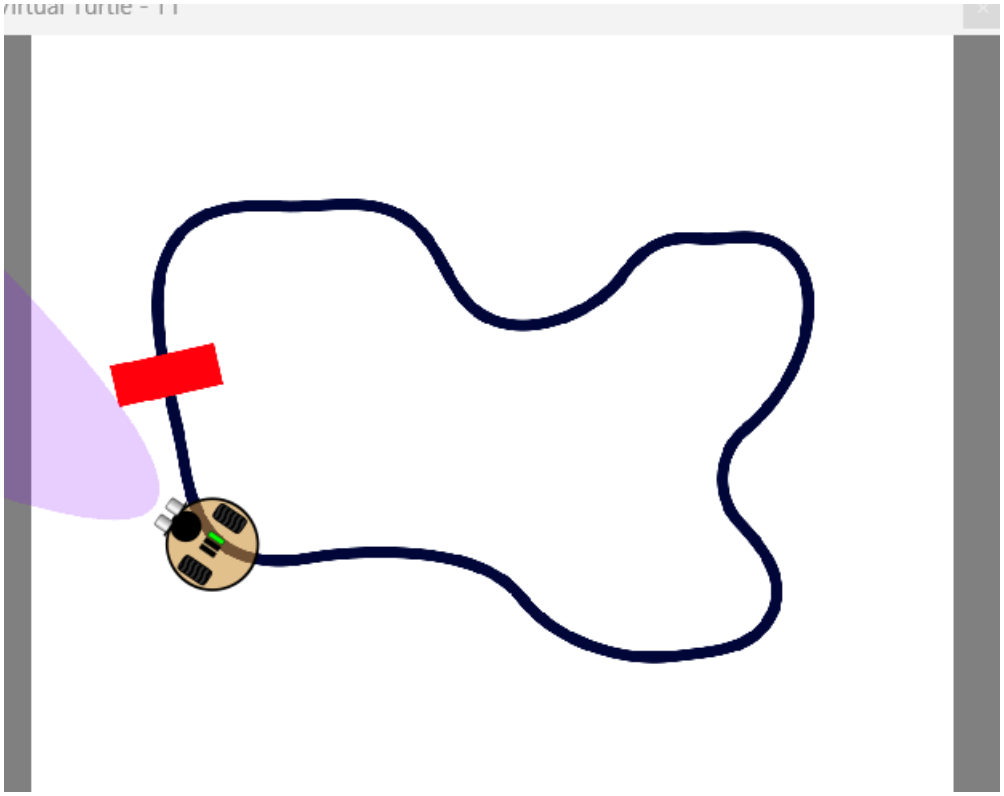
```
#pragma GCC push_options
#pragma GCC optimize ("Os")
#include <core.h>
#include <cpu.h>
#include <TimerOne.h>
#include <Servo.h>
#include <Turtle.h>
#pragma GCC pop_options

CPU &cpu = Cpu;
TimerOne &timer1 = Timer1;
TurtleDrive T1_DRIVE = TurtleDrive(2, 4, 3, 6, 7, 5);
TurtleSonarHead T1_SH = TurtleSonarHead(8, 9, 10);
TurtleLineHunter T1_LH = TurtleLineHunter(11, 12, A0);
```

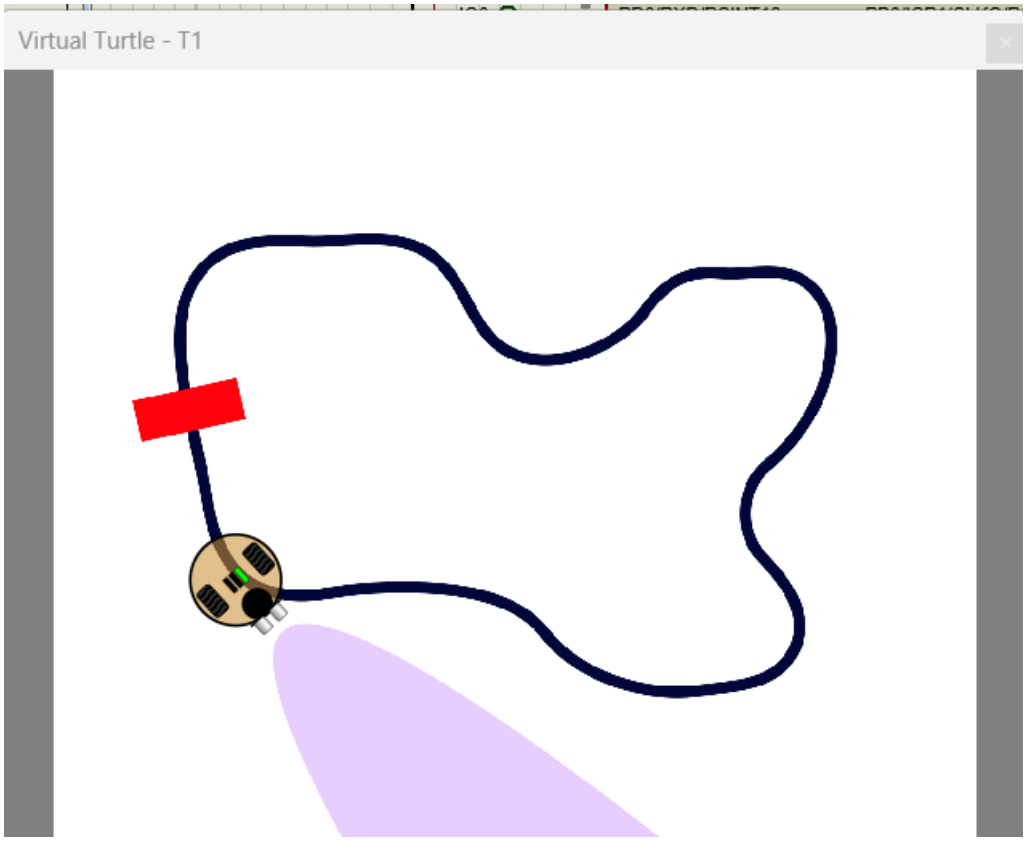
```
void peripheral_setup() {
   T1_DRIVE.begin();
   T1_SH.begin();
   T1_LH.begin();
}

void peripheral_loop() {
}

long var_speed;
long var_dir;
long var_count;
long var_range;
long var_fast;
long var_slow;
long var_tstart;
long var_tstop;

void chart_SETUP() {
   var_speed = 180;
   var_range = 10;
   var_dir = 0;
   T1_SH.setAngle(0);
   T1_SH.setRange(25);
   T1_DRIVE.forwards(var_speed);
}

void chart_LOOP() {
   if (!(T1_LH(0, 0, 0))) {
      if (T1_LH(1, 1, 1)) {
         chart_Correct();
      } else {
         if (T1_LH(0, 1, 1)) {
            T1_DRIVE.drive(1, 1, 5 * var_speed / 4);
            T1_DRIVE.drive(2, 1, var_speed / 2);
            var_dir = 10;
            chart_Avoid();
         } else if (T1_LH(0, 0, 1)) {
            T1_DRIVE.drive(1, 1, 5 * var_speed / 4);
            T1_DRIVE.drive(2, 0, var_speed / 5);
            var_dir = 30;
         } else if (T1_LH(1, 1, 0)) {
            T1_DRIVE.drive(2, 1, 5 * var_speed / 4);
            T1_DRIVE.drive(1, 1, var_speed / 2);
            var_dir = -10;
            chart_Avoid();
         } else if (T1_LH(1, 0, 0)) {
            T1_DRIVE.drive(2, 1, 5 * var_speed / 4);
            T1_DRIVE.drive(1, 0, var_speed / 5);
            var_dir = -30;
```

```
            } else if (T1_LH(-1, 1, -1)) {
                T1_DRIVE.forwards(var_speed);
                var_dir = 0;
                chart_Avoid();
            }
        }
    }
}

void chart_Correct() {
    var_count = 0;
l3:
    if (var_dir > 0) {
        T1_DRIVE.drive(2, 1, var_speed);
        T1_DRIVE.drive(1, 0, var_speed / 3);
    } else if (var_dir < 0) {
        T1_DRIVE.drive(1, 1, var_speed);
        T1_DRIVE.drive(2, 0, var_speed / 3);
    }
    delay(1);
    var_count = var_count + 1;
    if (var_count < 1000) {
        if (T1_LH(1, 1, 1)) goto l3;
    } else {
        T1_DRIVE.stop();
        var_dir = 0;
    }
}

void chart_Avoid() {
    if (T1_SH(var_range, 0)) {
        T1_DRIVE.backwards(2 * var_speed / 3);
        delay(250);
        T1_DRIVE.turn(80);
        do {
            delay(5);
        } while ((!(T1_SH(1.5 * var_range, 0))) == false);
        T1_DRIVE.stop();
        delay(500);
        T1_DRIVE.turn(80);
        var_tstart = cpu.millis();
        while (!(T1_SH(1.5 * var_range, 0))) {
        }
        var_tstop = cpu.millis();
        var_count = (var_tstop - var_tstart) / 10;
        T1_DRIVE.stop();
        delay(500);
        T1_DRIVE.turn(-80);
        while (var_count > 0) {
            var_count = var_count - 1;
```

```
            delay(5);
        }
        T1_DRIVE.backwards(2 * var_speed / 3);
        delay(300);
        T1_DRIVE.forwards(var_speed / 2);
    }
}

void setup() {
    peripheral_setup();
    chart_SETUP();
}

void loop() {
    peripheral_loop();
    chart_LOOP();
}
```

Output :




Virtual Turtle - T1

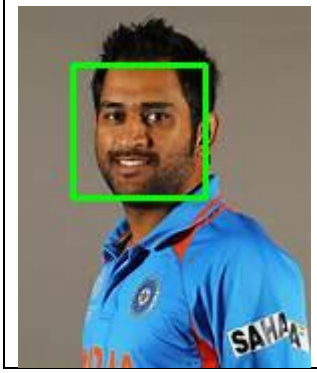
Practical No.9

Aim: Detect faces with Haar cascades.

Source Code:

```
import cv2
from google.colab.patches import cv2_imshow

img = cv2.imread("face.jpg")

gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

haar_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
"haarcascade_frontalface_default.xml")

faces_rect = haar_cascade.detectMultiScale(gray_img, scaleFactor=1.3, minNeighbors=5)

print(f"Number of faces detected: {len(faces_rect)}")

for (x, y, w, h) in faces_rect:
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2_imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Number of faces detected: 1



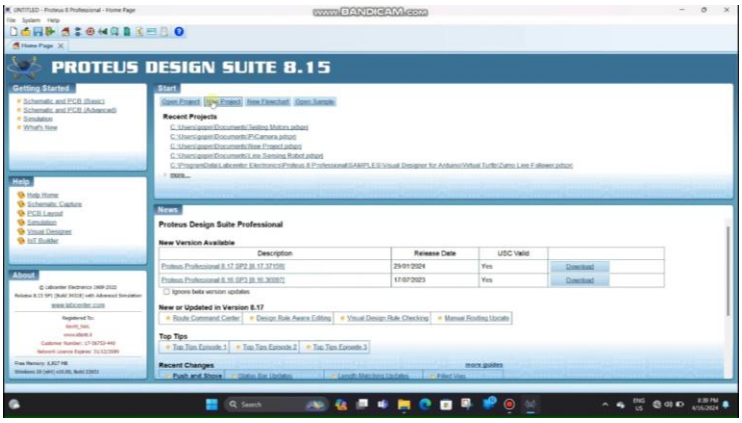
Practical No.10

Aim : Use the robot to display its camera as a web app on a phone or desktop, and then use the camera to drive smart color and face-tracking behaviors.

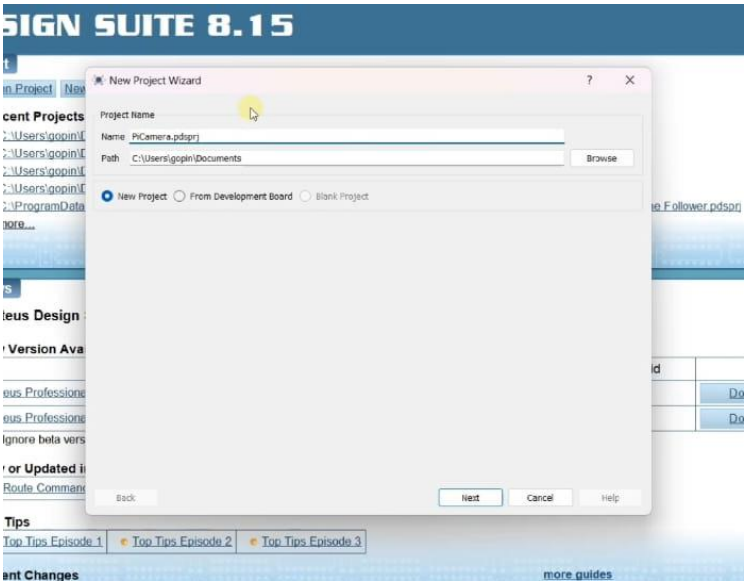
Components:-

- **LCD TFT**
- **Raspberry PI Camera**
- **Button**

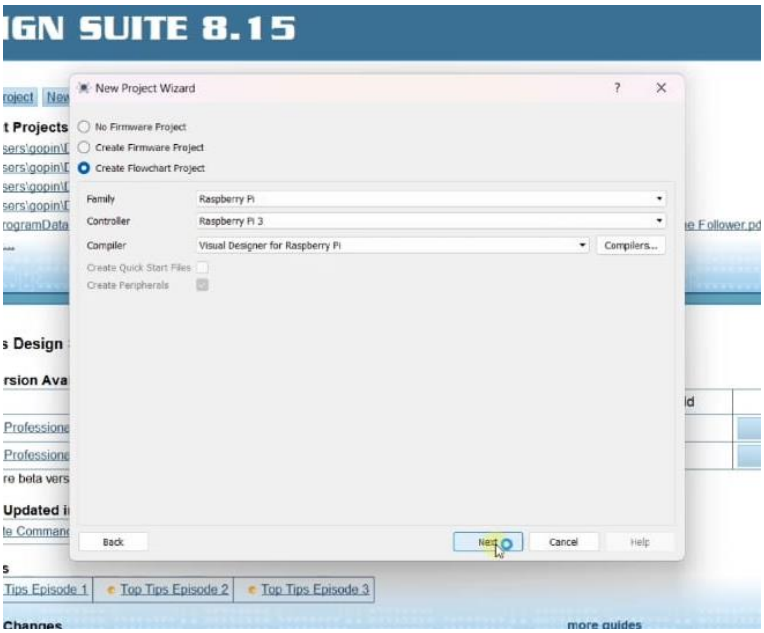
Step 1 : Goto new project and select new project



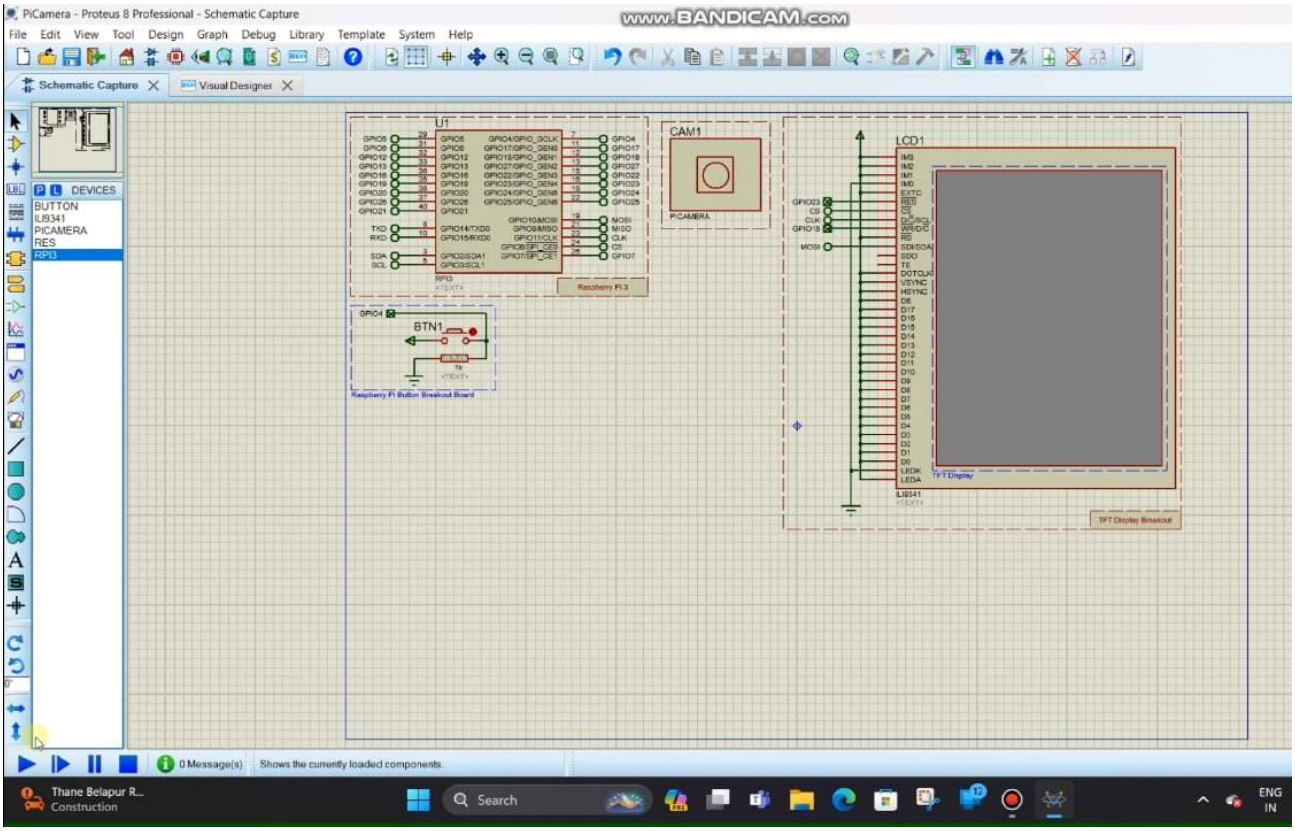
Step 2 :Change the name of the project and save as PiCamera.psdprj



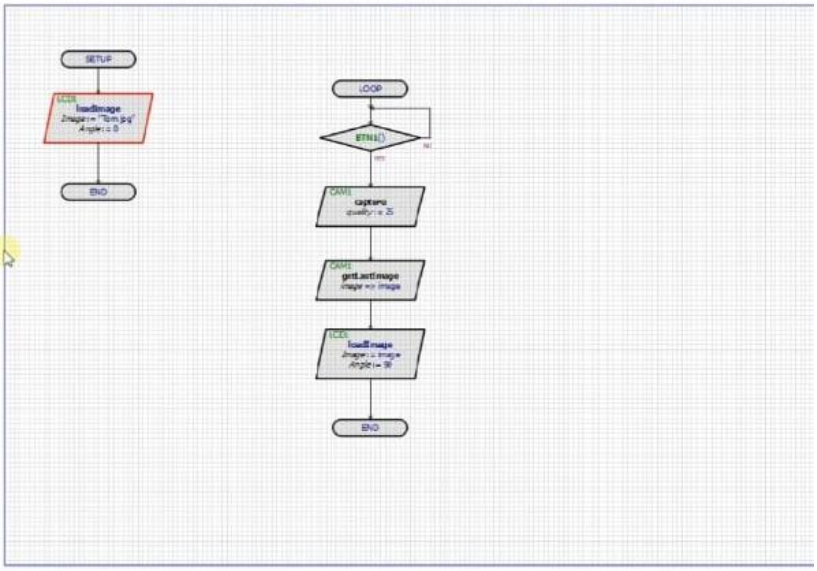
Step 3 : Create a Flowchart Project



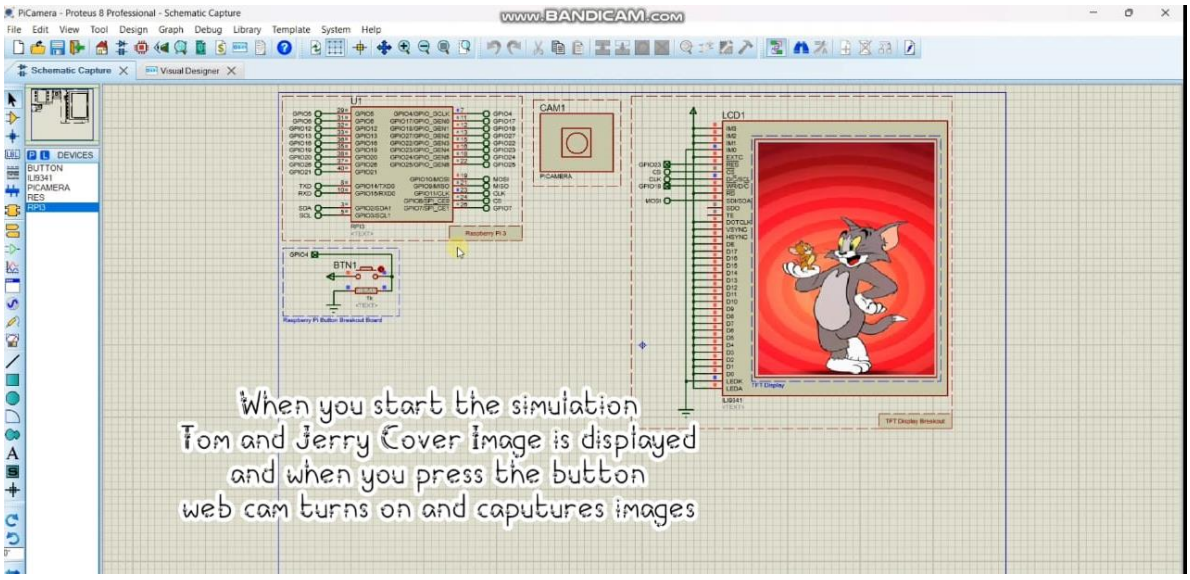
Step 4 : Implementation of the Circuit

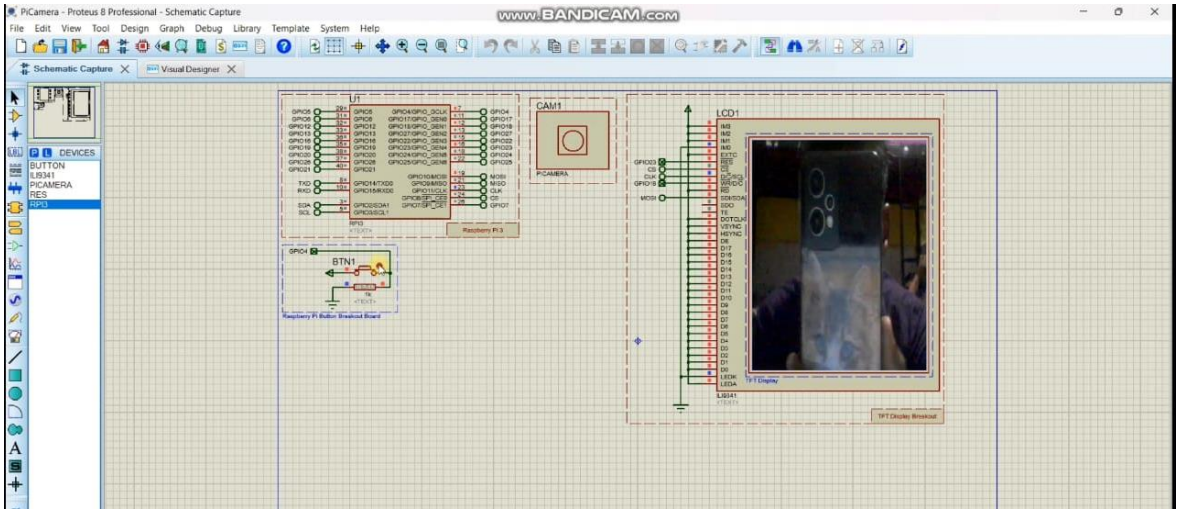


Flow Chart :



Output :




PiCamera - Proteus 8 Professional - Schematic Capture
www.BANDICAM.com
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Visual Designer
DEVICES
BUTTON
ILI9341
PICAMERA
RES
RPI3
U1
CAM1
PICAMERA
LCD1
BTN1
GPIO4
MOSI
TFT Display
TFT Display Breakout