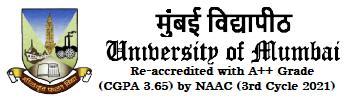
UNIVERSITY OF MUMBAI

**DEPARTMENT OF COMPUTER SCIENCE**



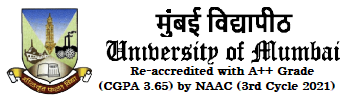
M.Sc. Computer Science – Semester IV

Robotics

JOURNAL

2023-2024

Seat No. 40423



UNIVERSITY OF MUMBAI

**DEPARTMENT OF COMPUTER SCIENCE**

**CERTIFICATE**

This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr..**Koli Kalpesh Ananda** Seat No. **40423** for the course of M.Sc. Computer Science - Semester IV (CBCS) (Revised) during the academic year 2023- 2024 in a satisfactory manner.

**\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_**

**Subject In-charge Head of Department**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**External Examiner**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Practical** | **Page**  **No** | **Date** | **Sign** |
| **1** | **Making a Raspberry Pi headless, and reaching it from the network using WiFi and SSH.** | **1** |  |  |
| **2** | **Using sftp upload files from PC.** | **7** |  |  |
| **3** | **Write Python code to test motors.** | **18** |  |  |
| **4** | **Develop Python code for testing the sensors.** | **26** |  |  |
| **5** | **Add the sensors to the Robot object and develop the line following behavior code** | **28** |  |  |
| **6** | **Add pan and tilt service to the robot object and test it.** | **31** |  |  |
| **7** | **Create an obstacle avoidance behavior for robot and test it.** | **33** |  |  |
| **8** | **Detect faces with Haar cascades.** | **39** |  |  |

**Practical No. 01**

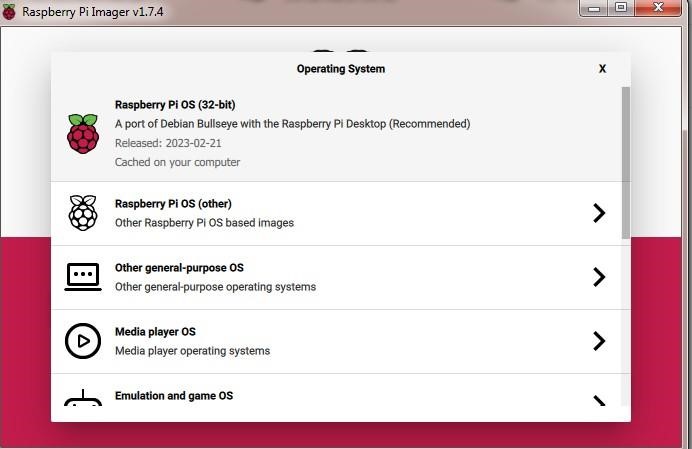
**Aim: Making Raspberry pi headless, and reaching it from the network using wifi and SSH.**

**Implementation:**

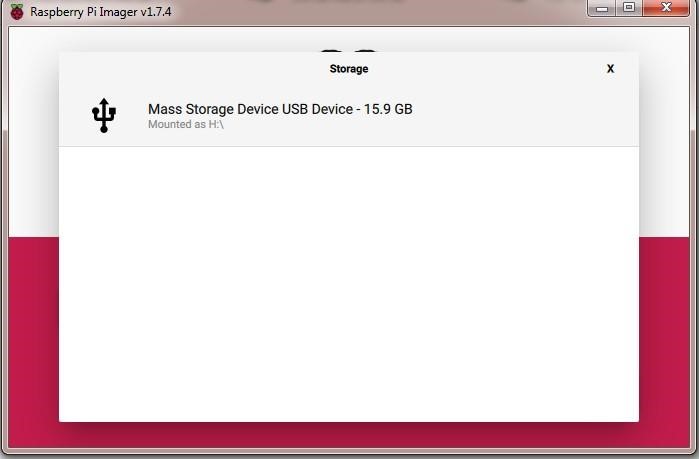
**STEP I :** Download raspberry pi imager for windows. And connect sd card to laptop.

****

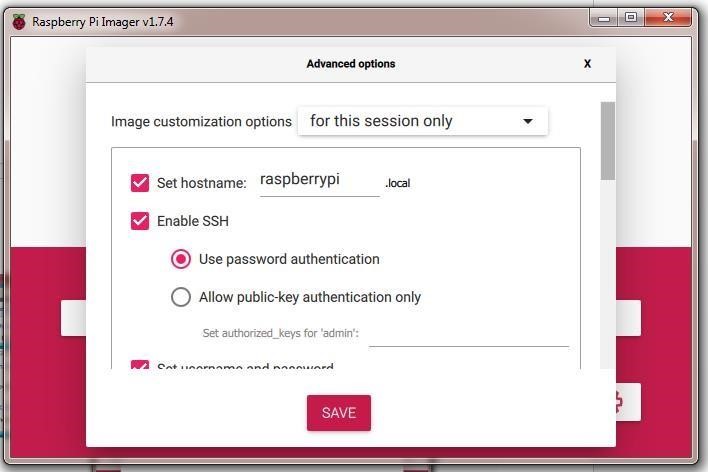
**STEP II : Select operating sysyem (raspberry pi OS 32-bit)**

****

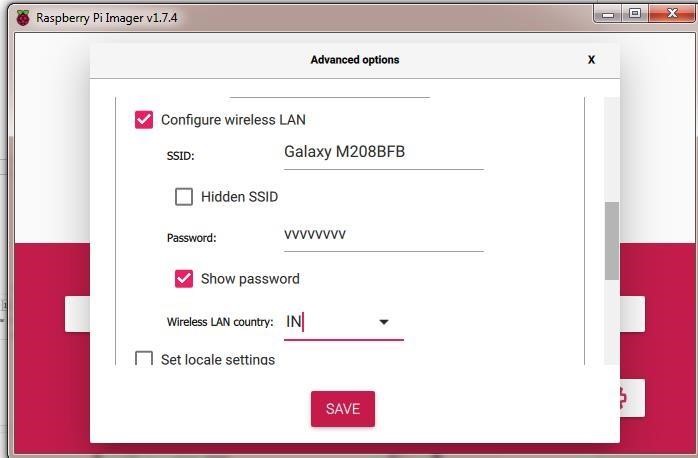
**STEP III : Select Storage.**

****

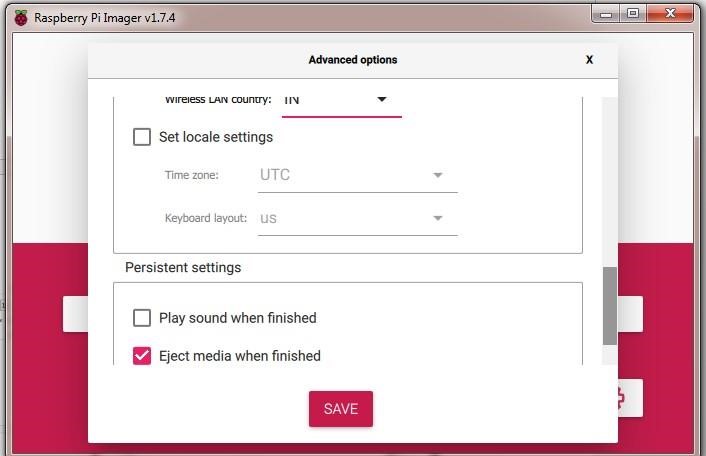
**STEP IV : Click on Setting Icon . and set hostname. Click to enable SSH.**

****

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

****

**STEP VI : Click to save.**

****

**STEP VII : Click on write.**

****

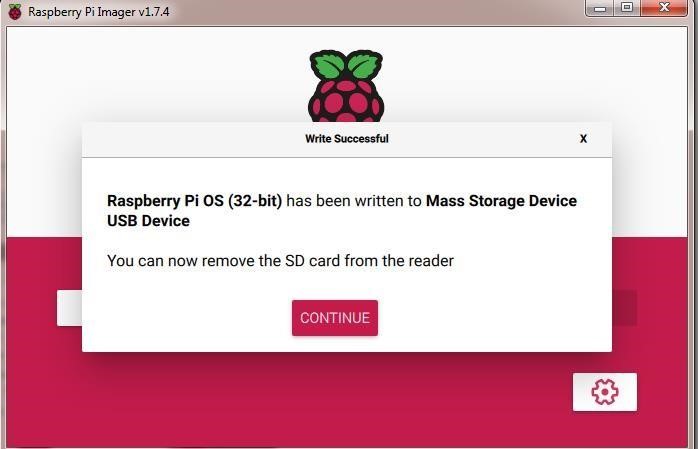
**STEP VIII : Click on yes.**

****

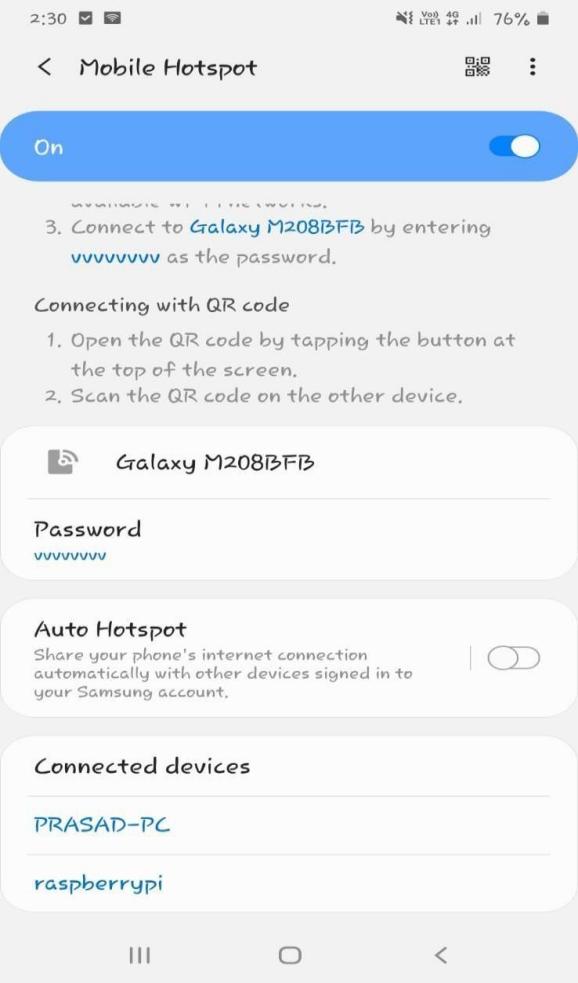
**STEP IX : Verifying raspberry pi .**

****

**STEP X : Click on continue .**

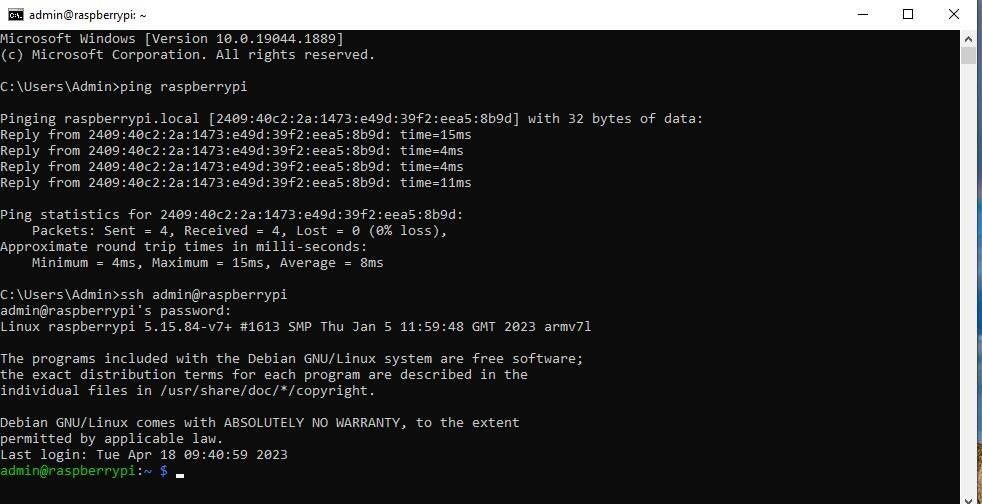
****

**Step XI : check the connection in your mobile hotspot of raspberrypi**

****

**STEP X: On cmd promt execute these command :**

1. **ping raspberrypi**
2. **ssh admin@raspberrypi**

****

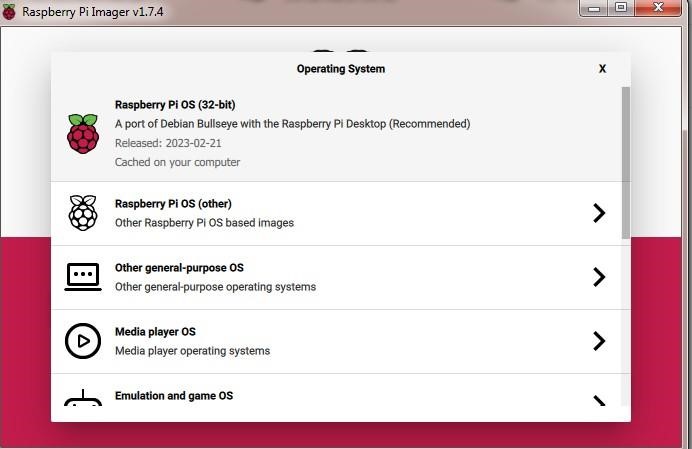
**Practical No 2**

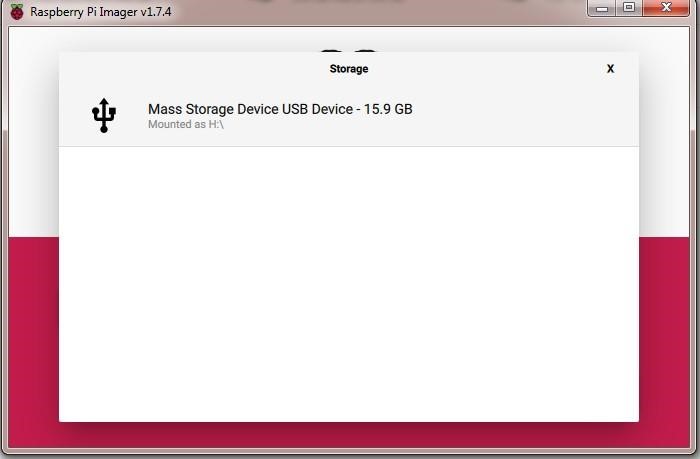
**Aim: Using sftp upload files from PC.**

**Implementation :**

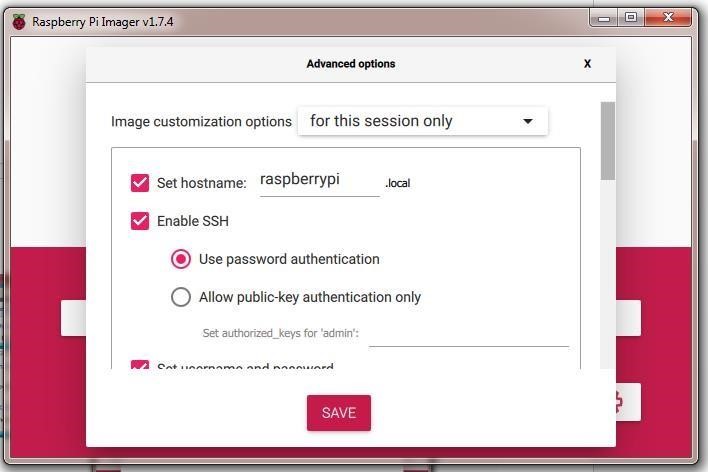
**Step 1:** Intsall the Raspberry Pi Imager

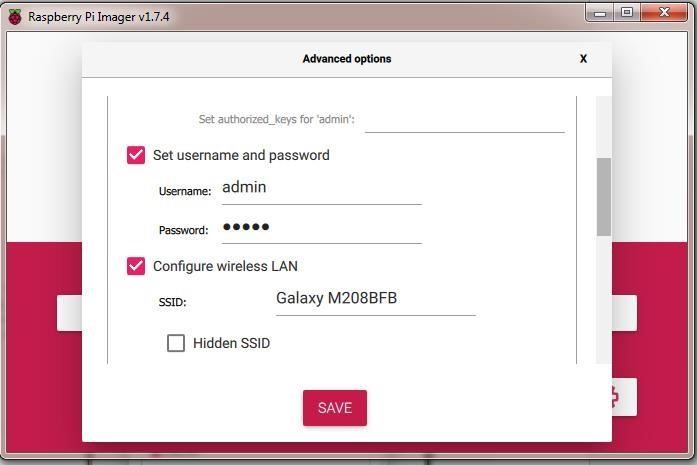
****

****

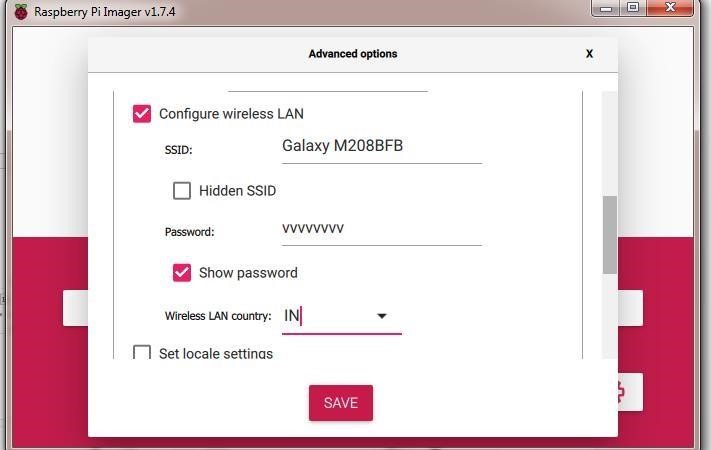
****

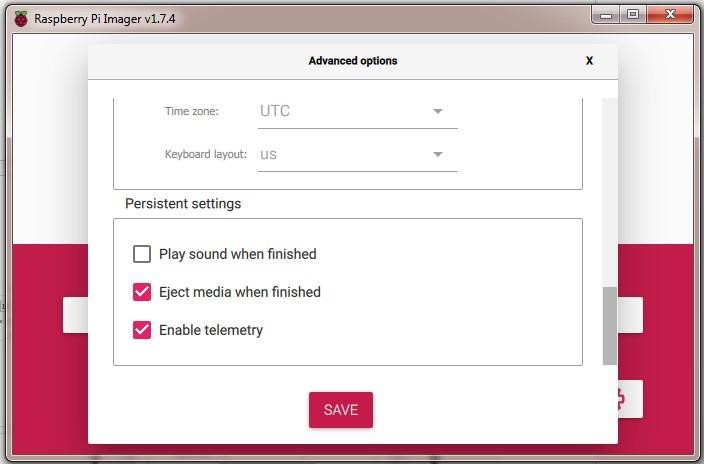
**Step 2: Set Host Name , enable SSH, Set Username and Password.**

****

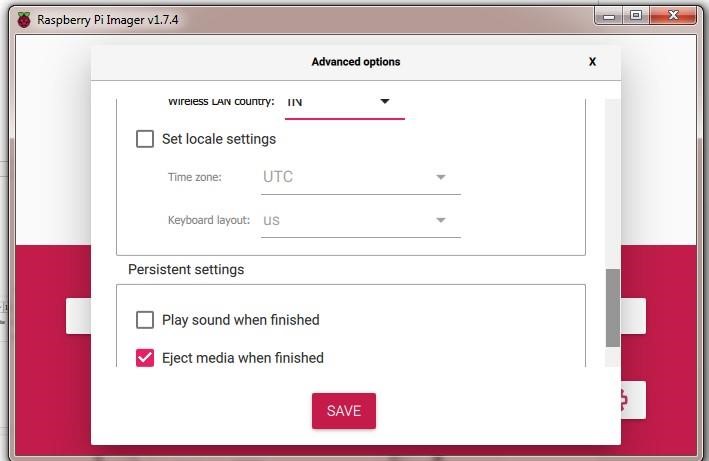
****

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

****

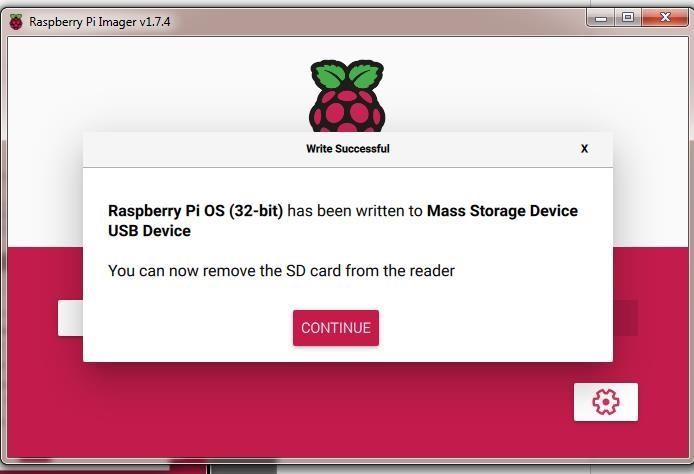
****



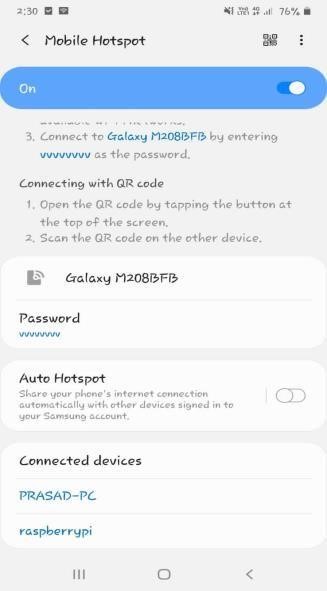
****

****

****

****

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

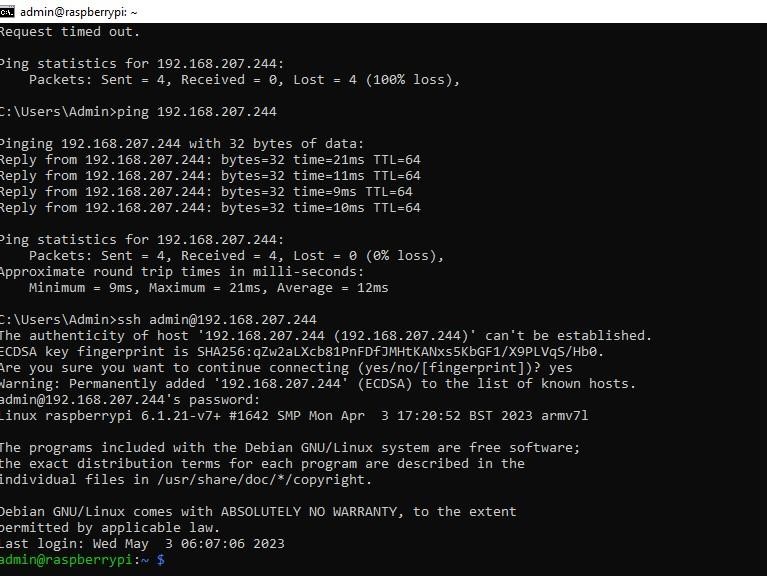
****

**Step 5: Open CMD and tyepe following command**

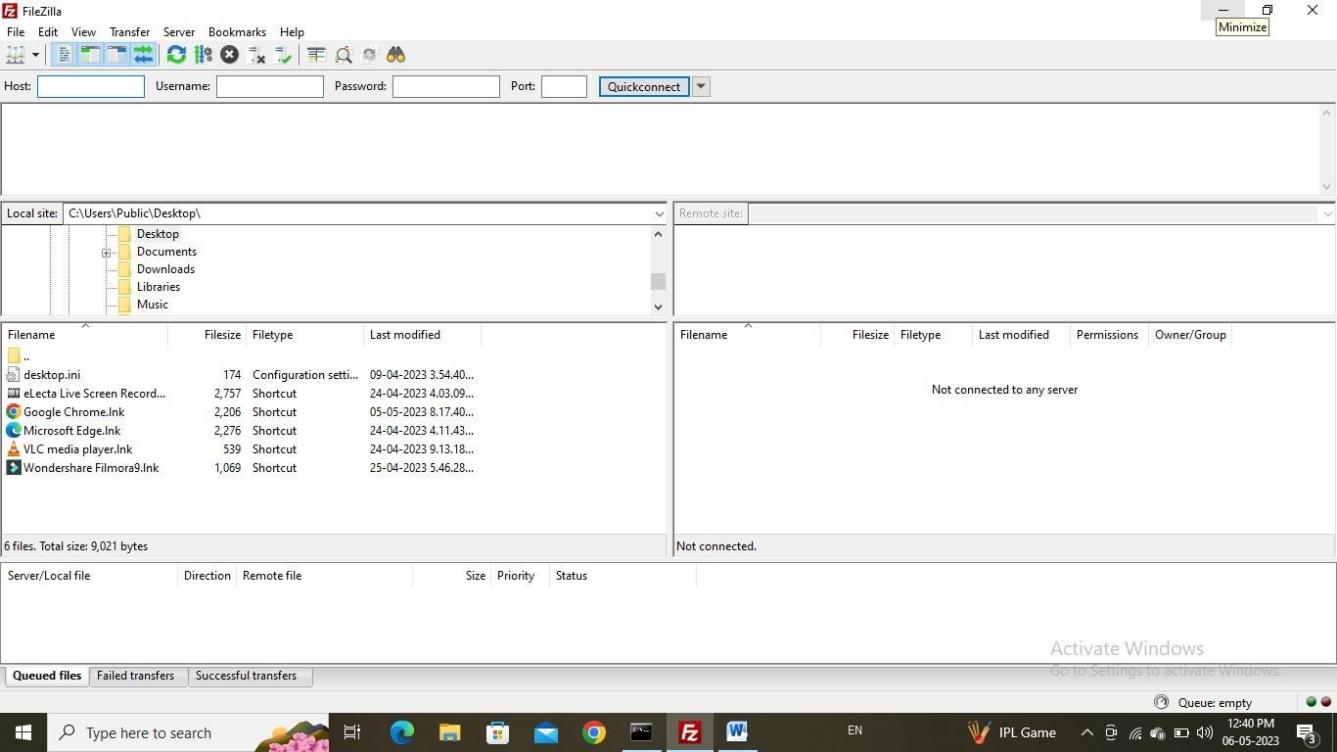
**1 ) ping raspberrypi or ping 162.168.207.244**

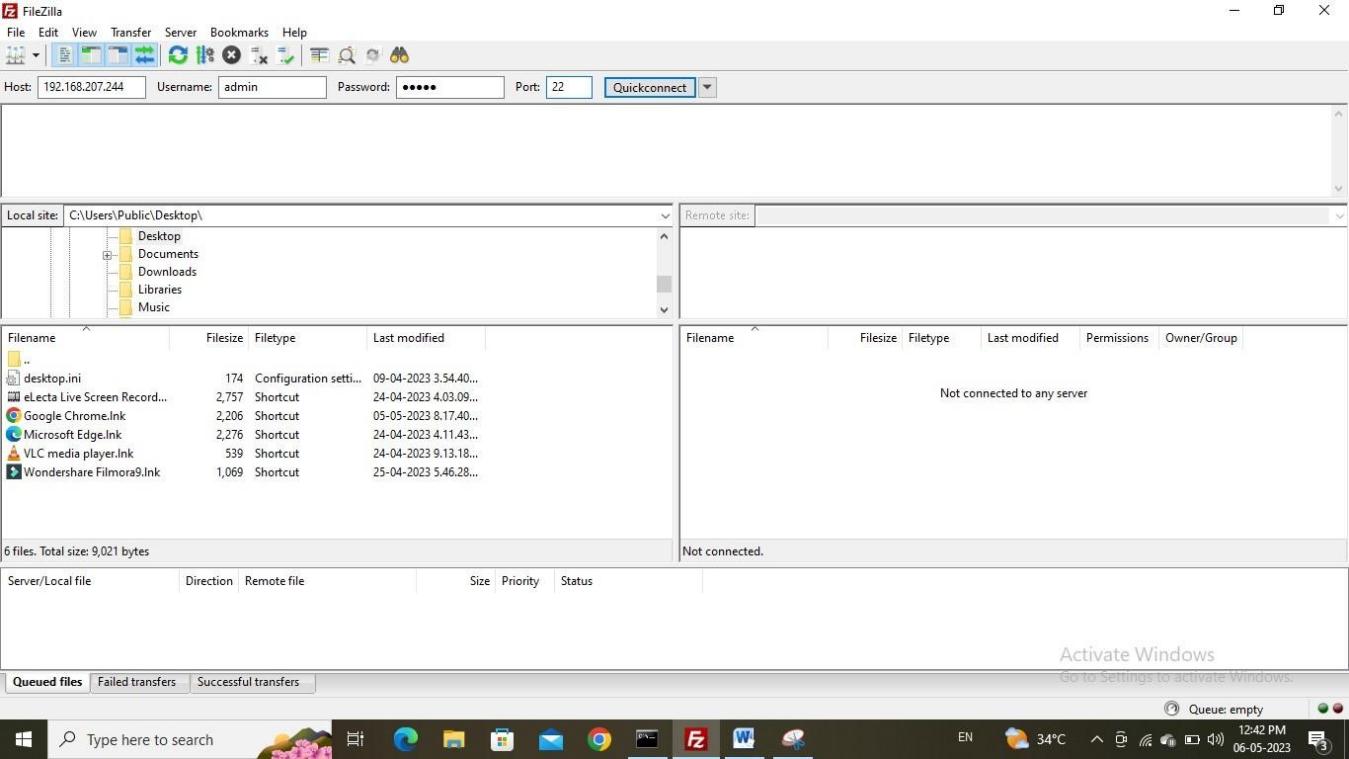
**2) ssh admin@ raspberrypi or ssh admin@**

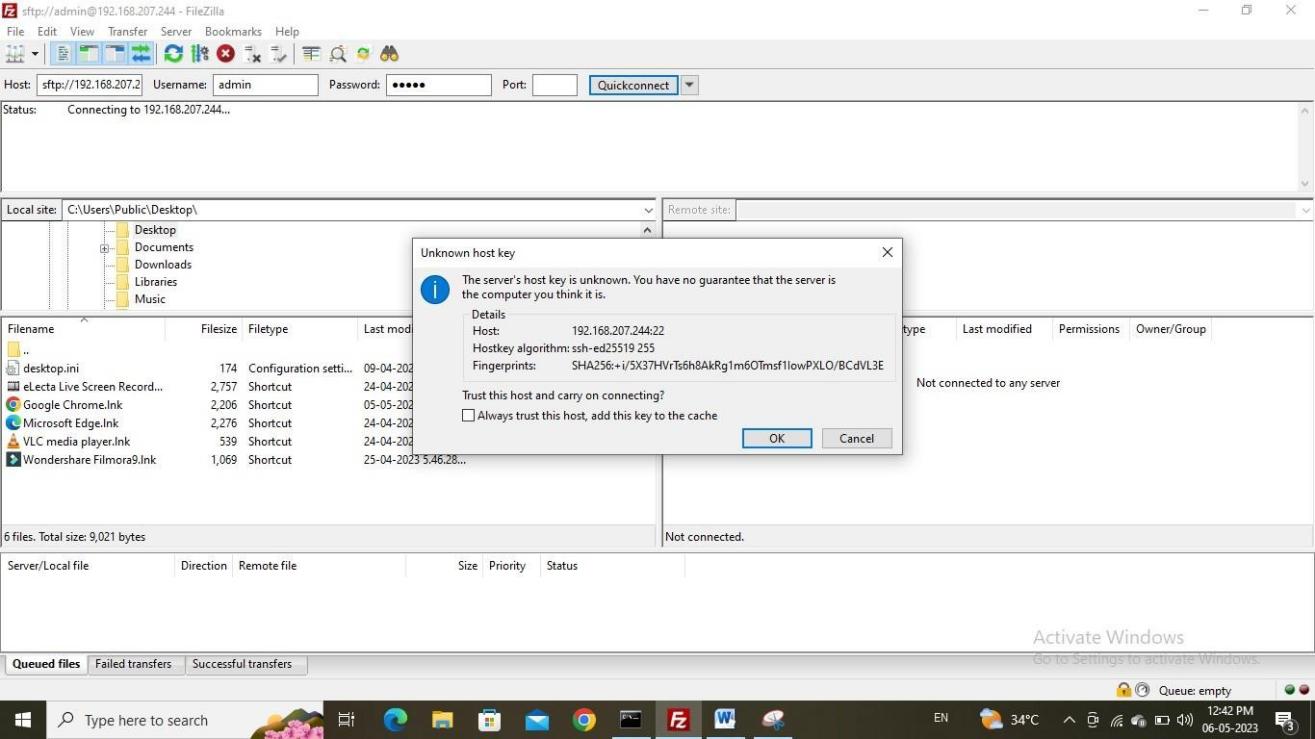
**162.168.207.244 And type password of Admin**

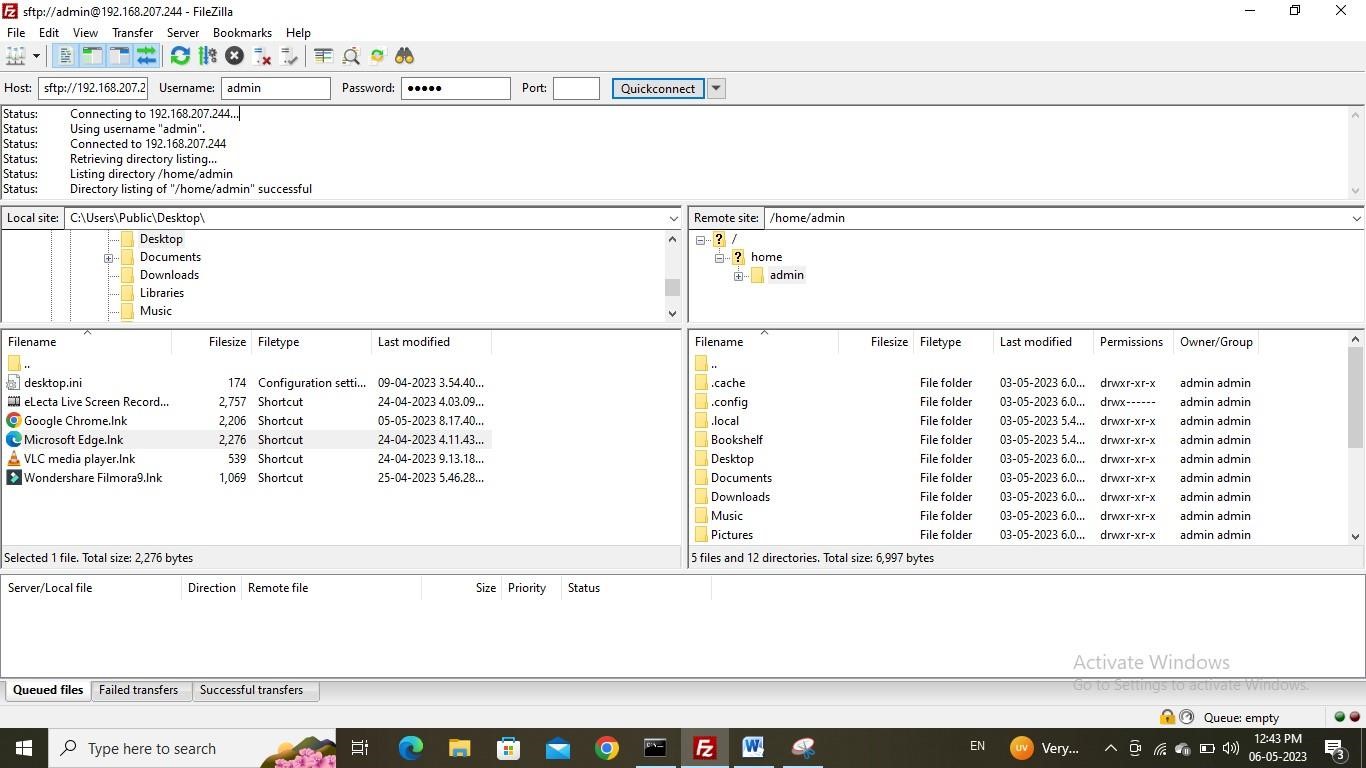
****

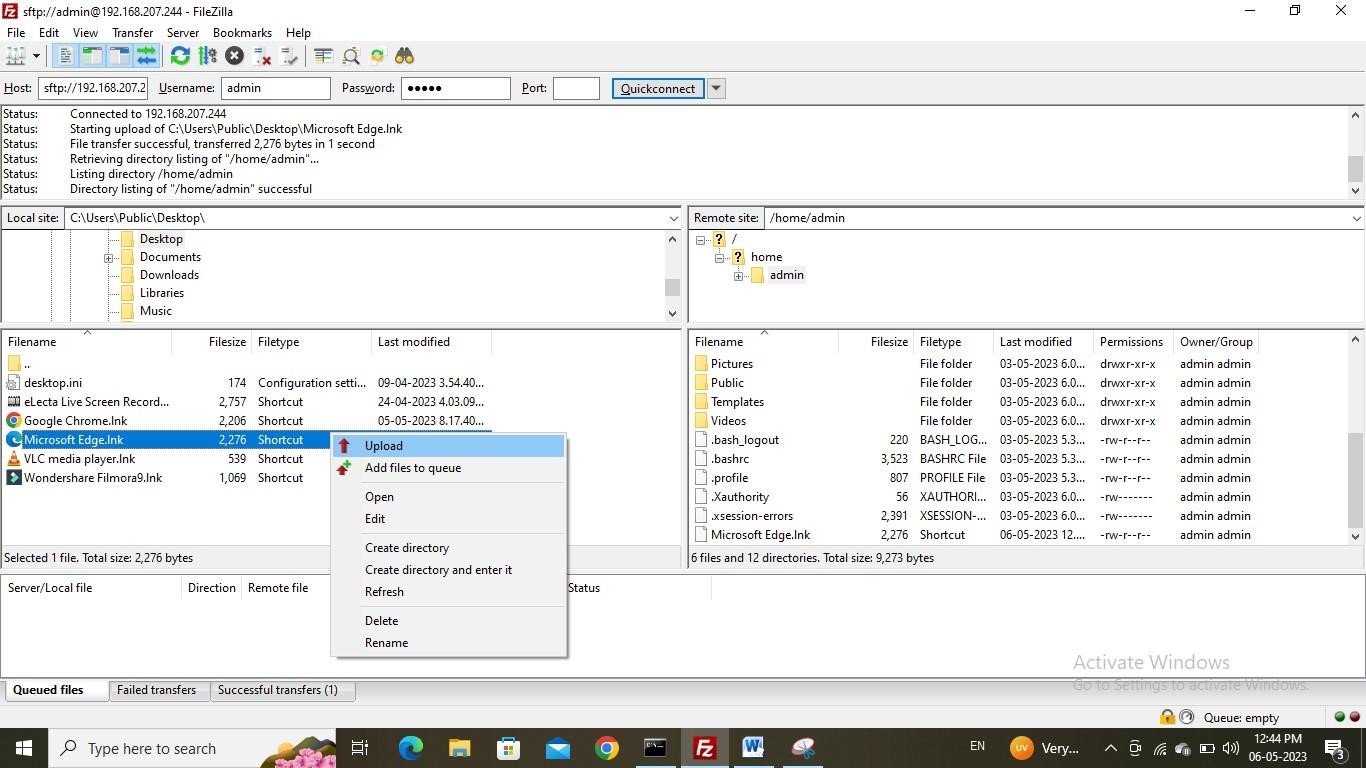
**Step:6 Download the FileZilla (Client)**

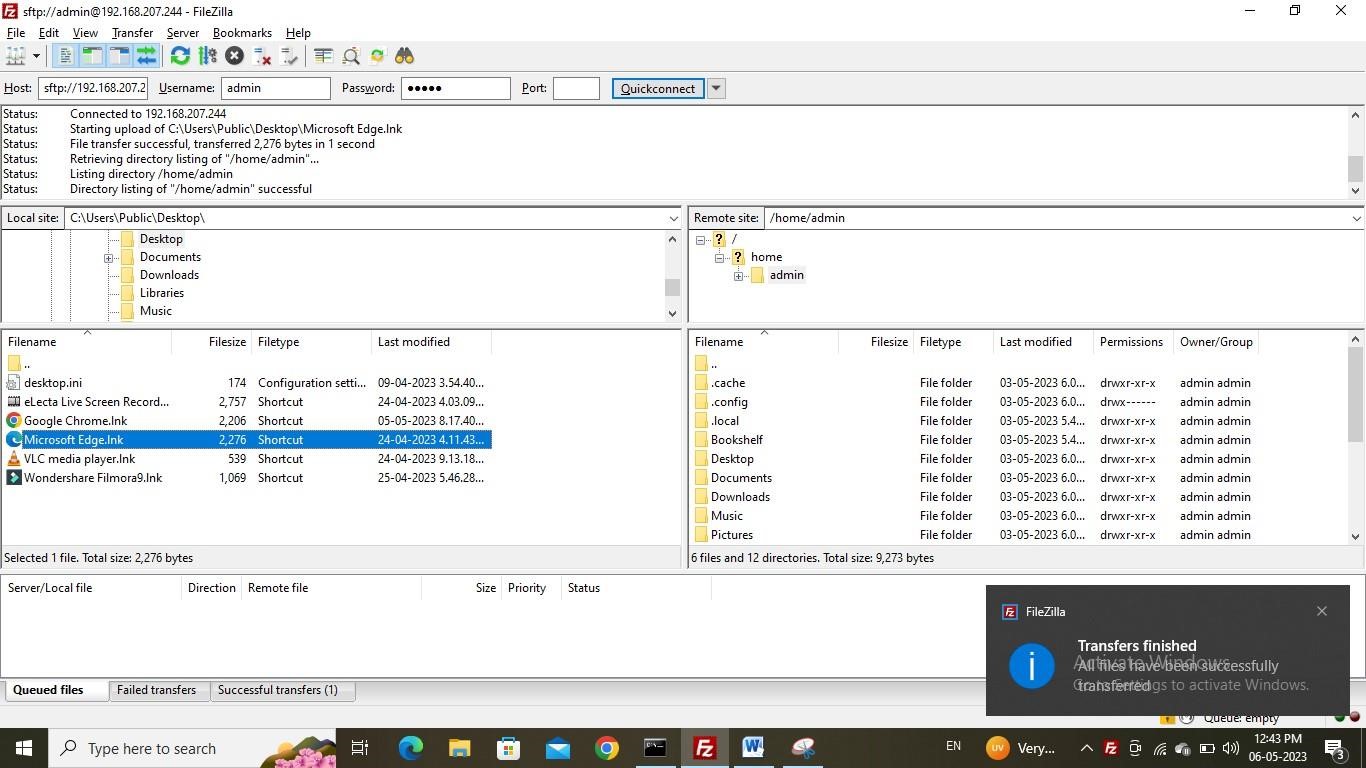
****

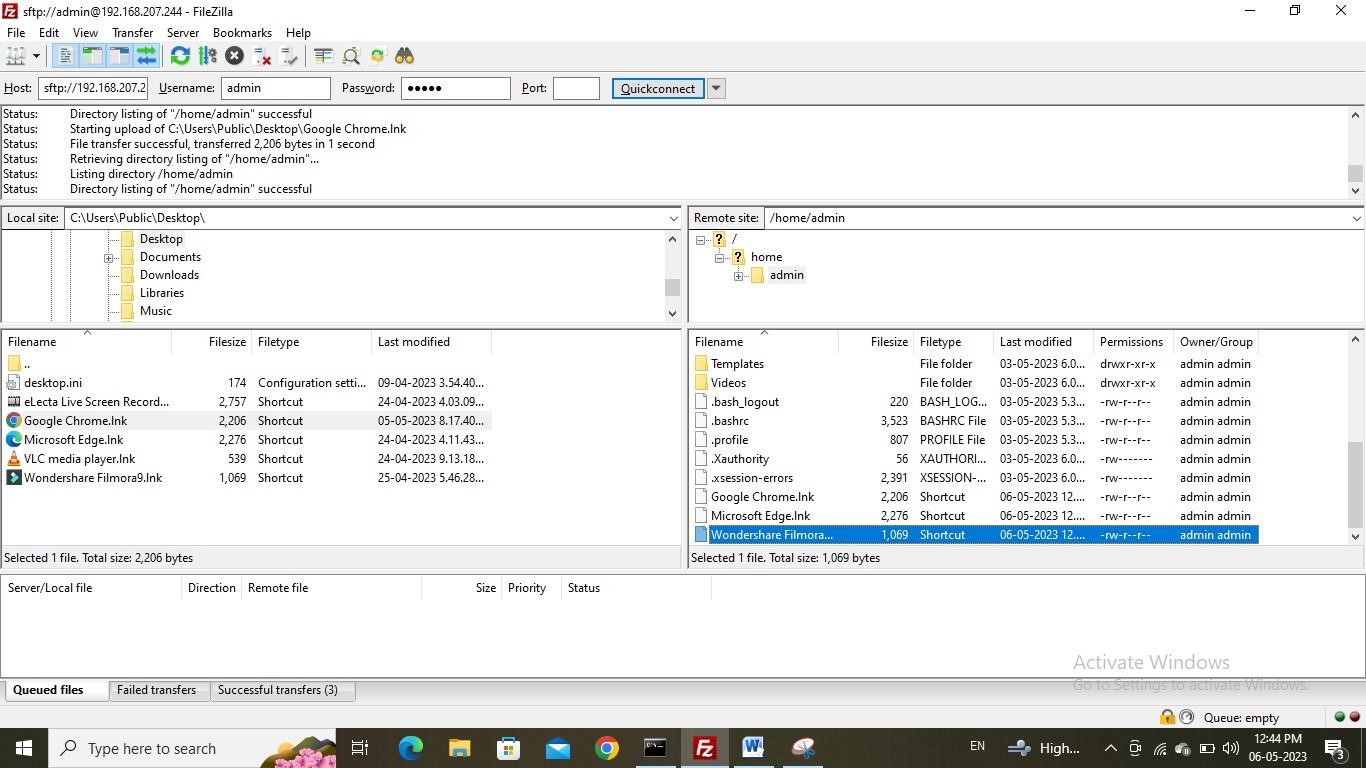
****

****

****

****

****

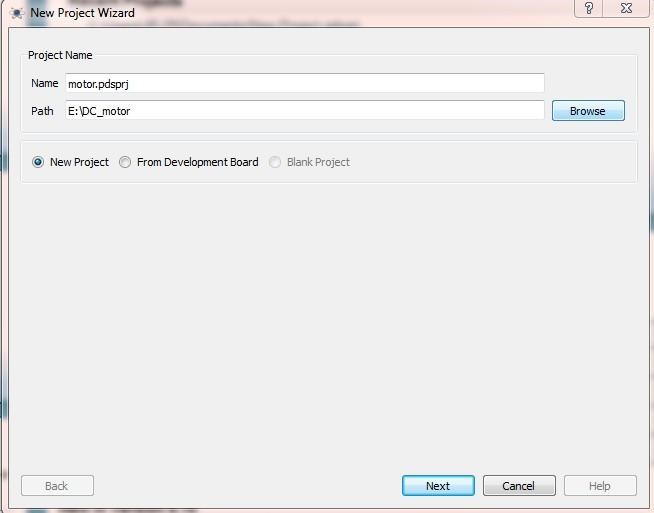
****

**Practical No: 03**

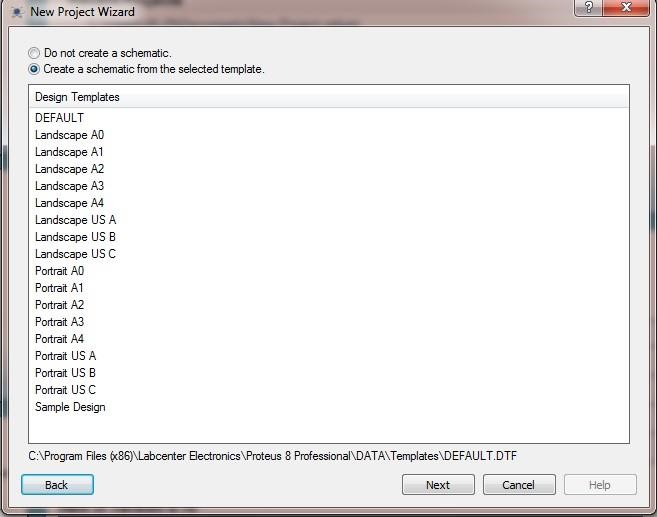
**Aim:** Write python code to test Motors.

**OUTPUT:**

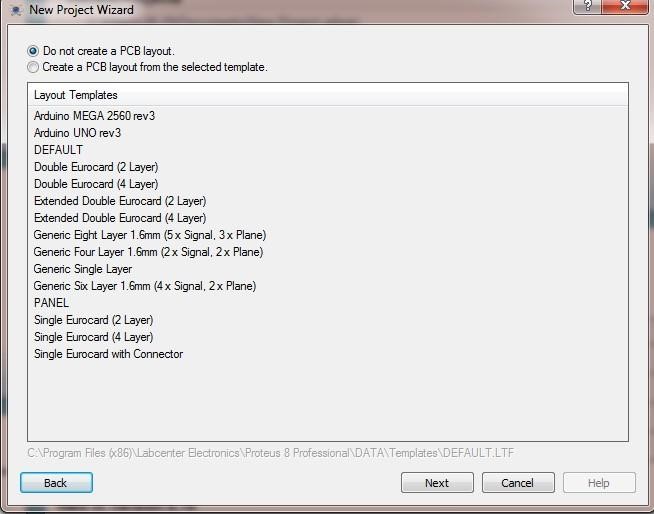
**Give project name.**

****

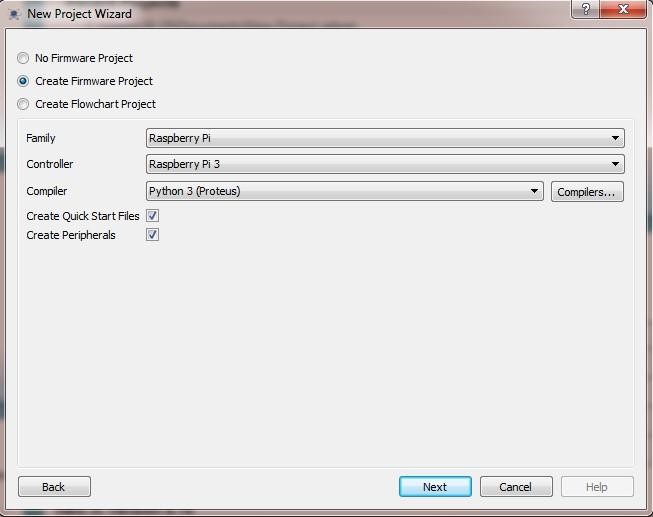
**Select “create a schematic from the selected template”.**

****

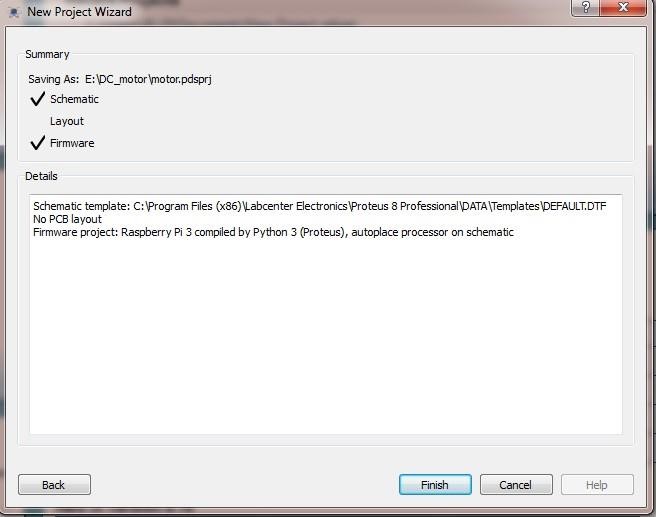
**Select “Do not create a PCB layout”**

****

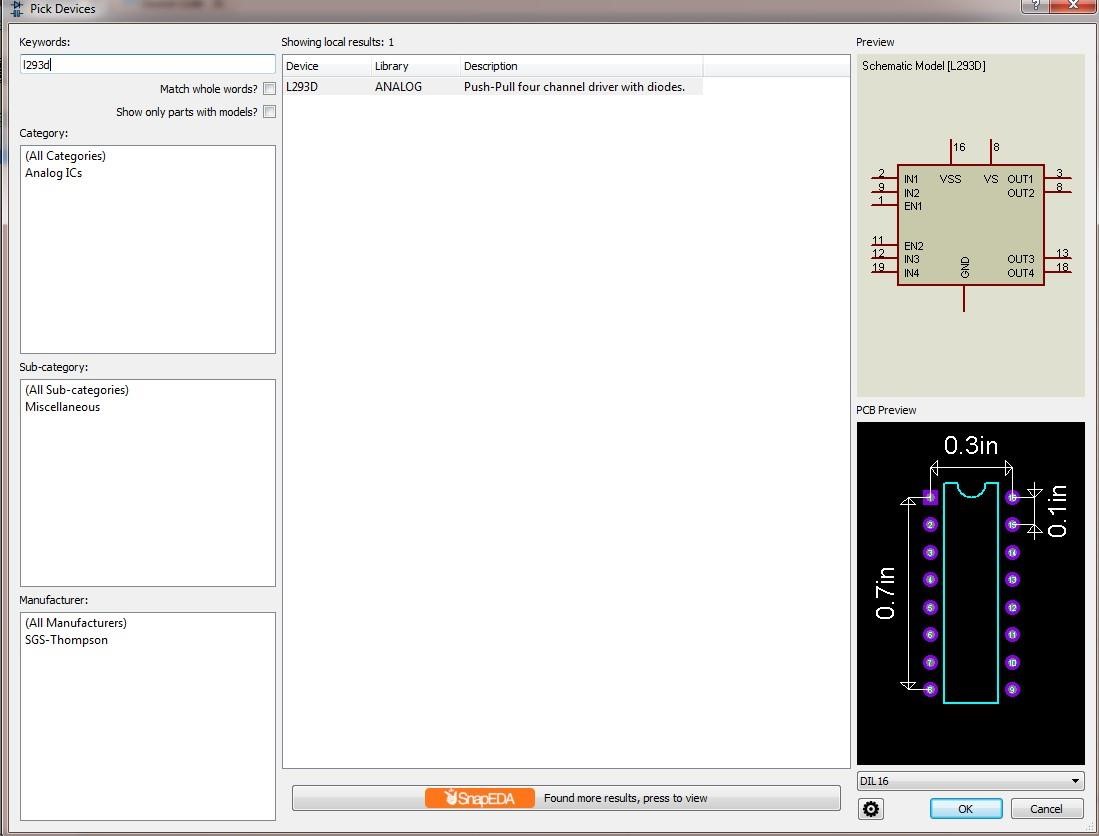
**Select “Create firmware project”. and create firmware project in proteus and select Raspberry pi Family.**

****

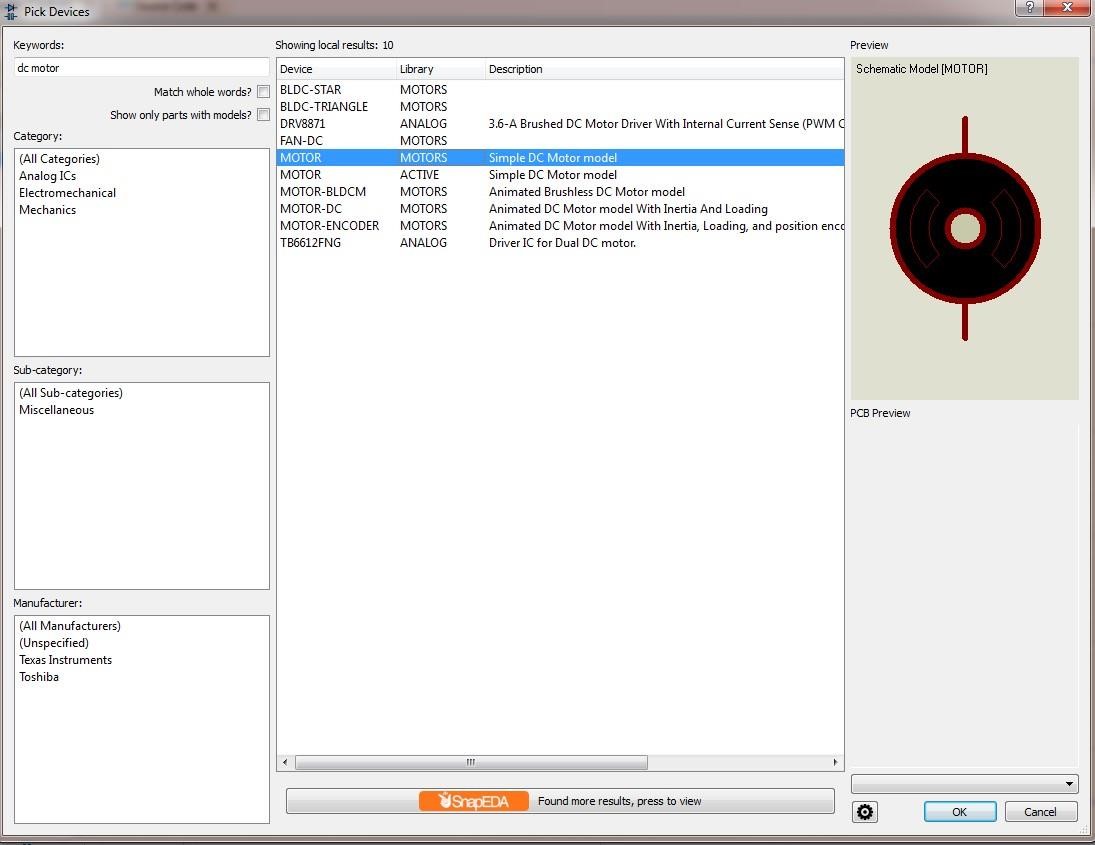
**Click on Finish.**

****

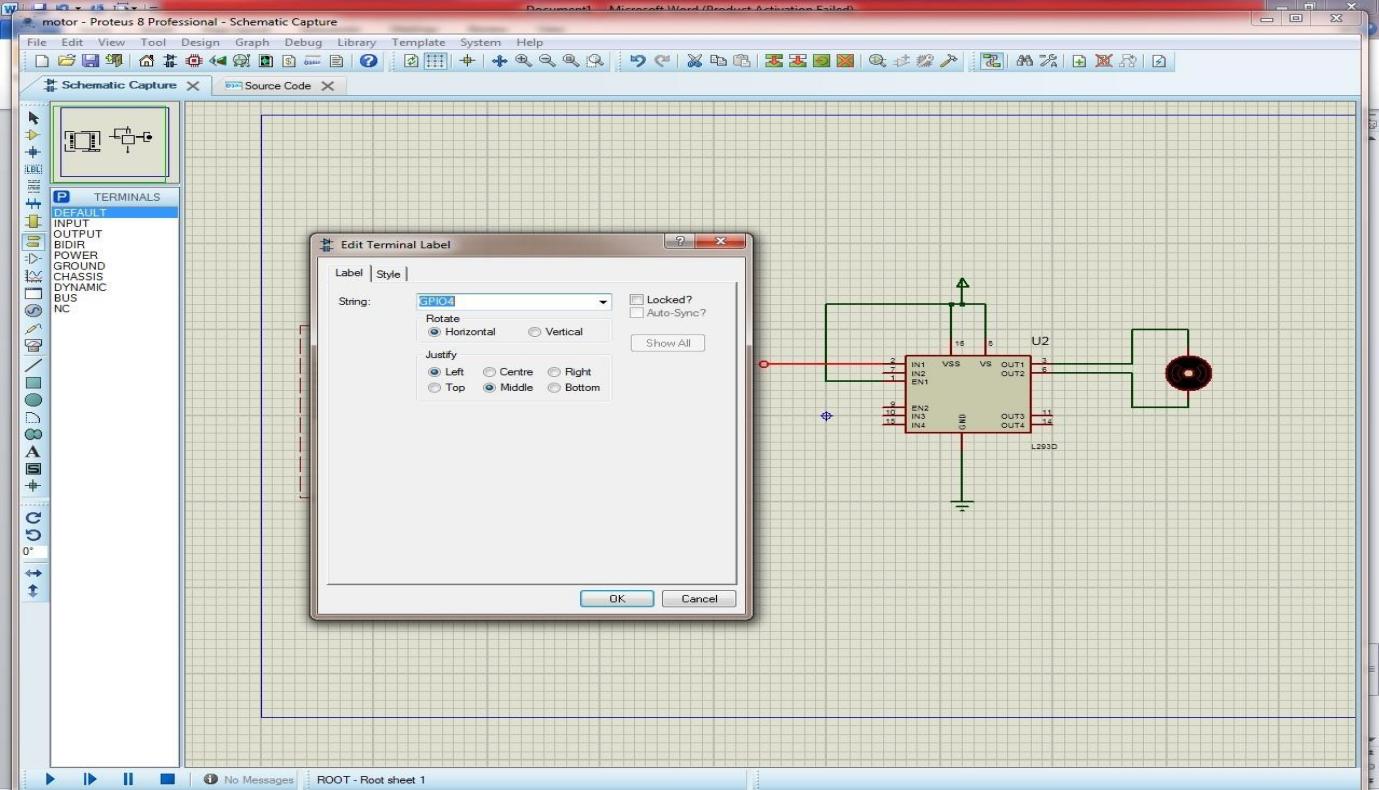
**Go to the liabrary>pick devices>L293D**

****

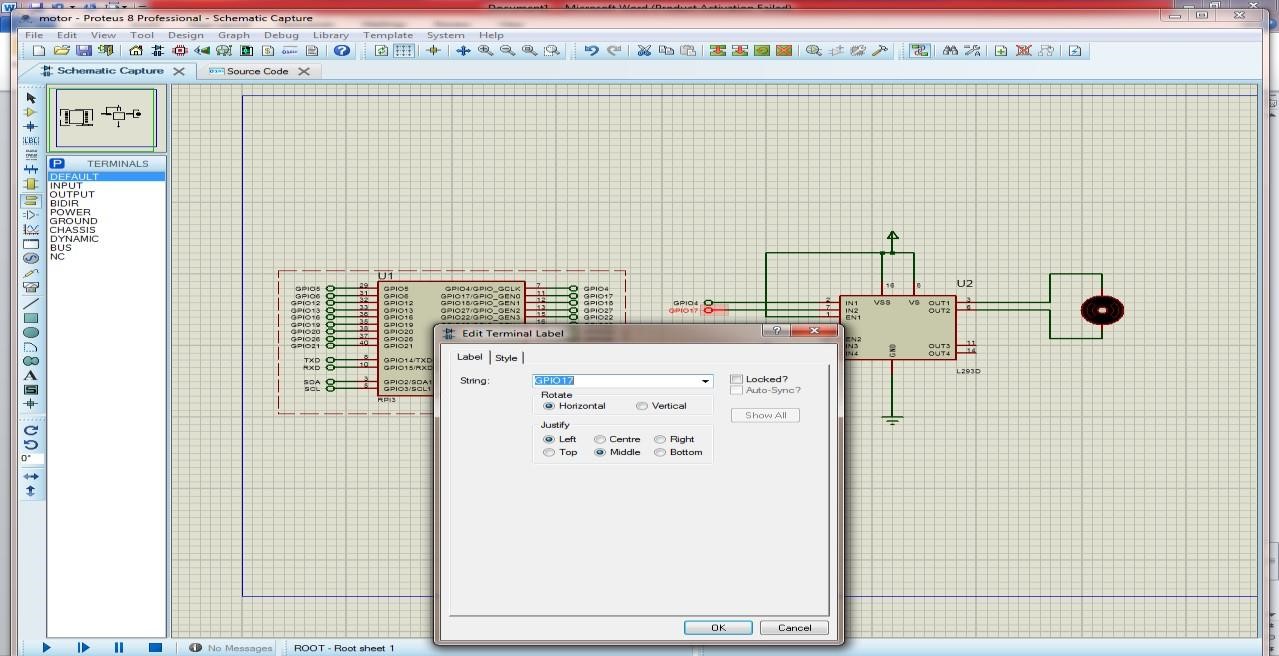
**Select MOTOR**

****

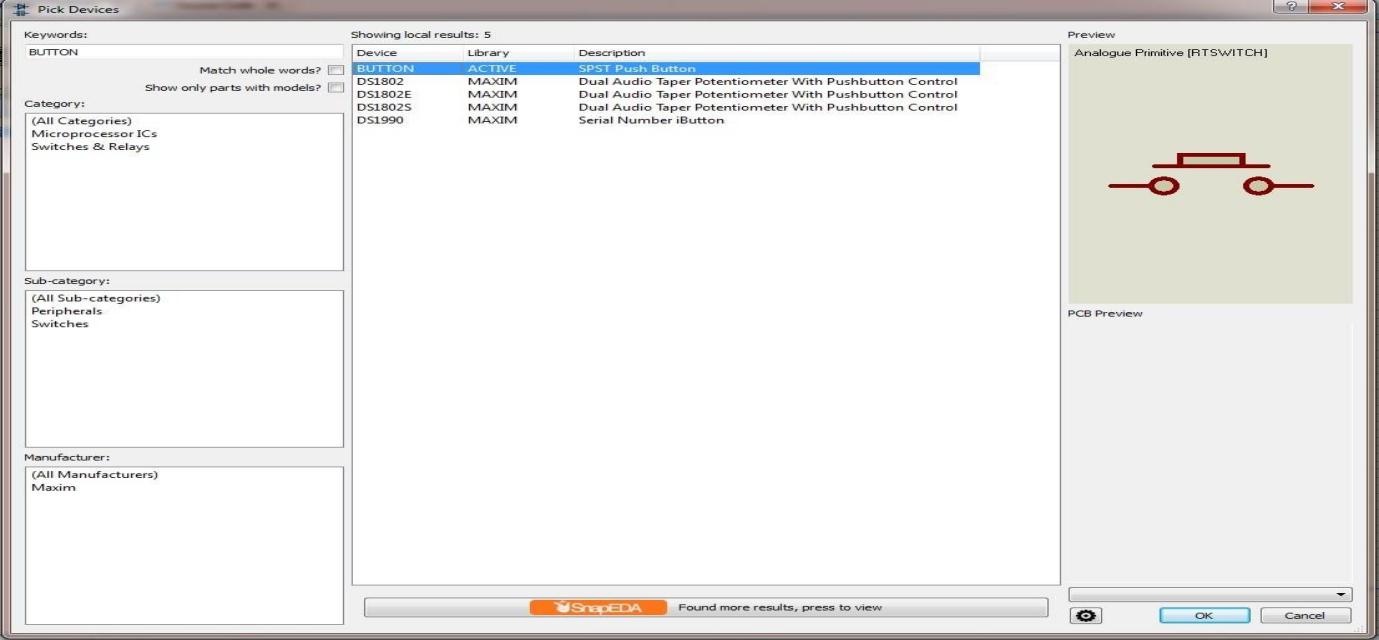
**Edit terminal label as GPIO4**

****

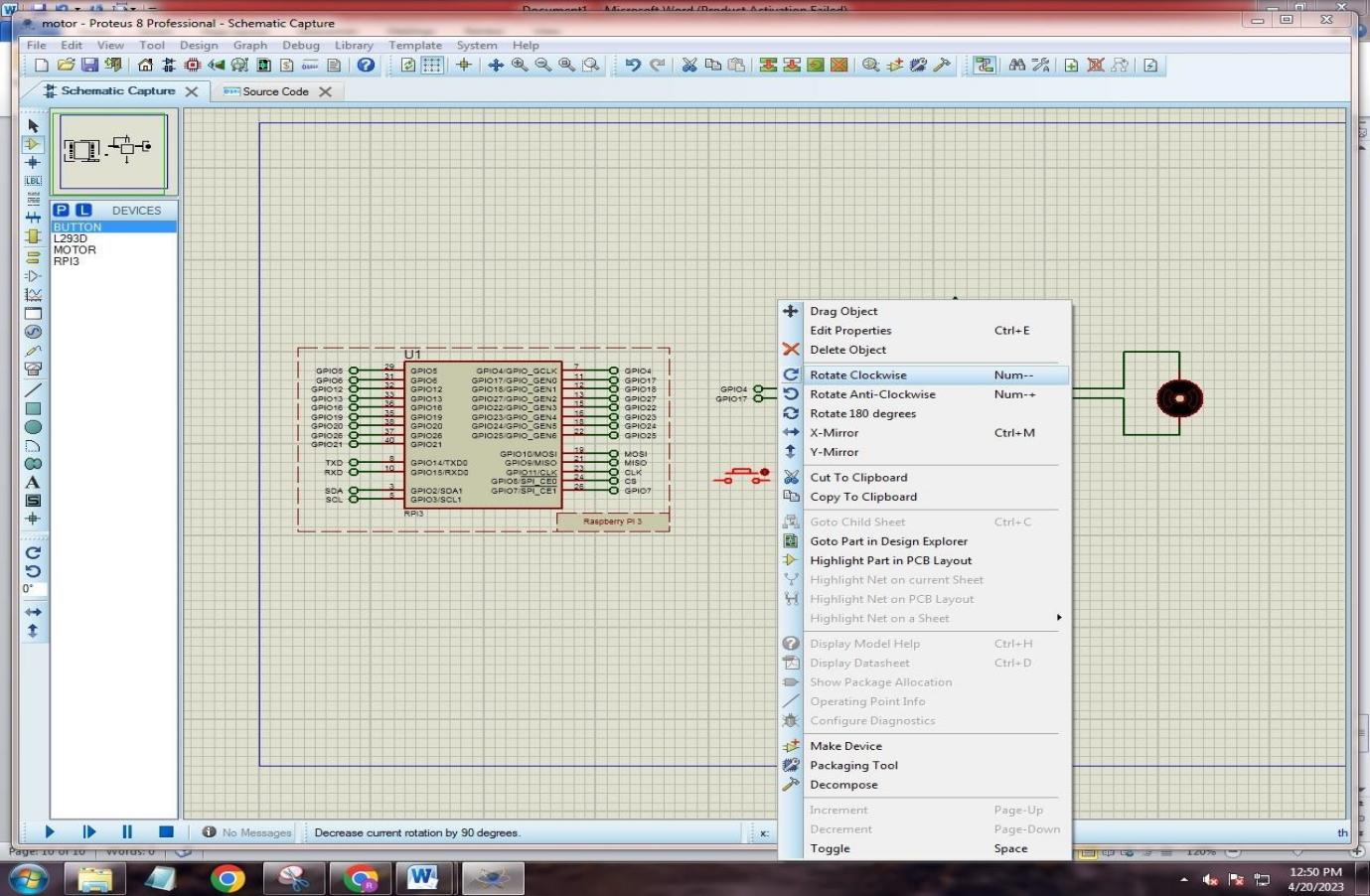
**Edit terminal label as GPIO17**

****

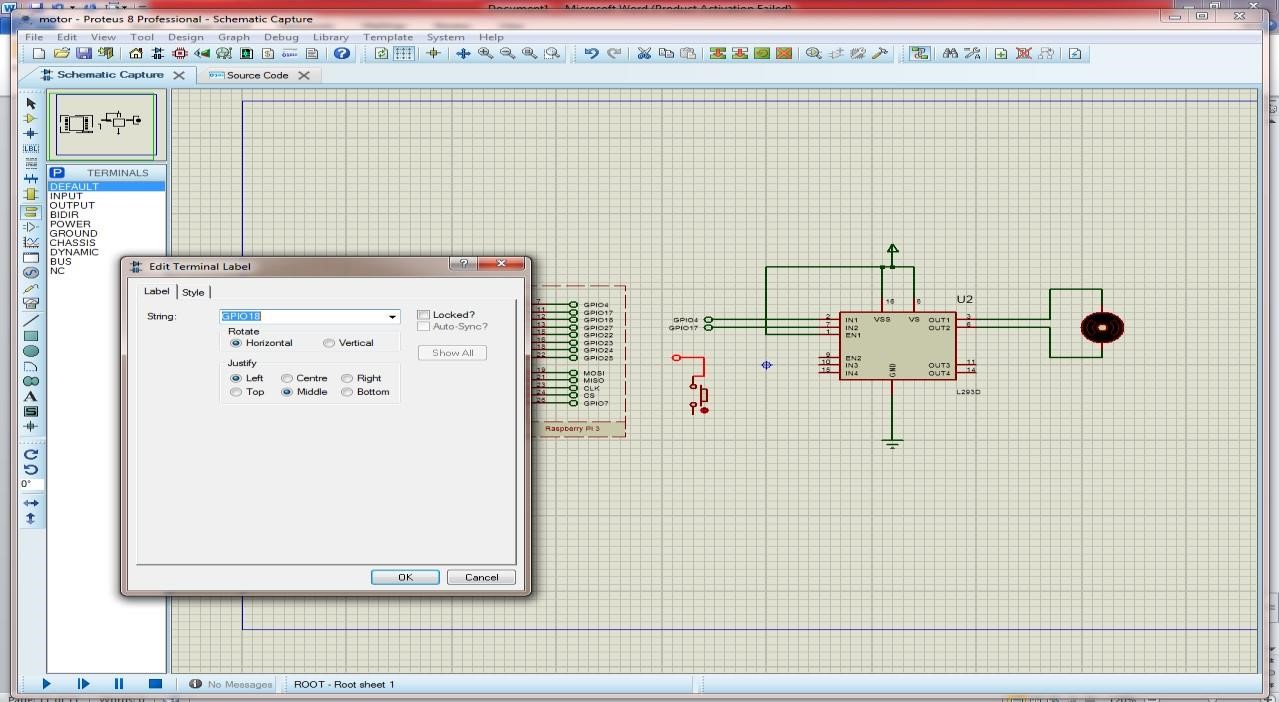
**Select BUTTON**

****

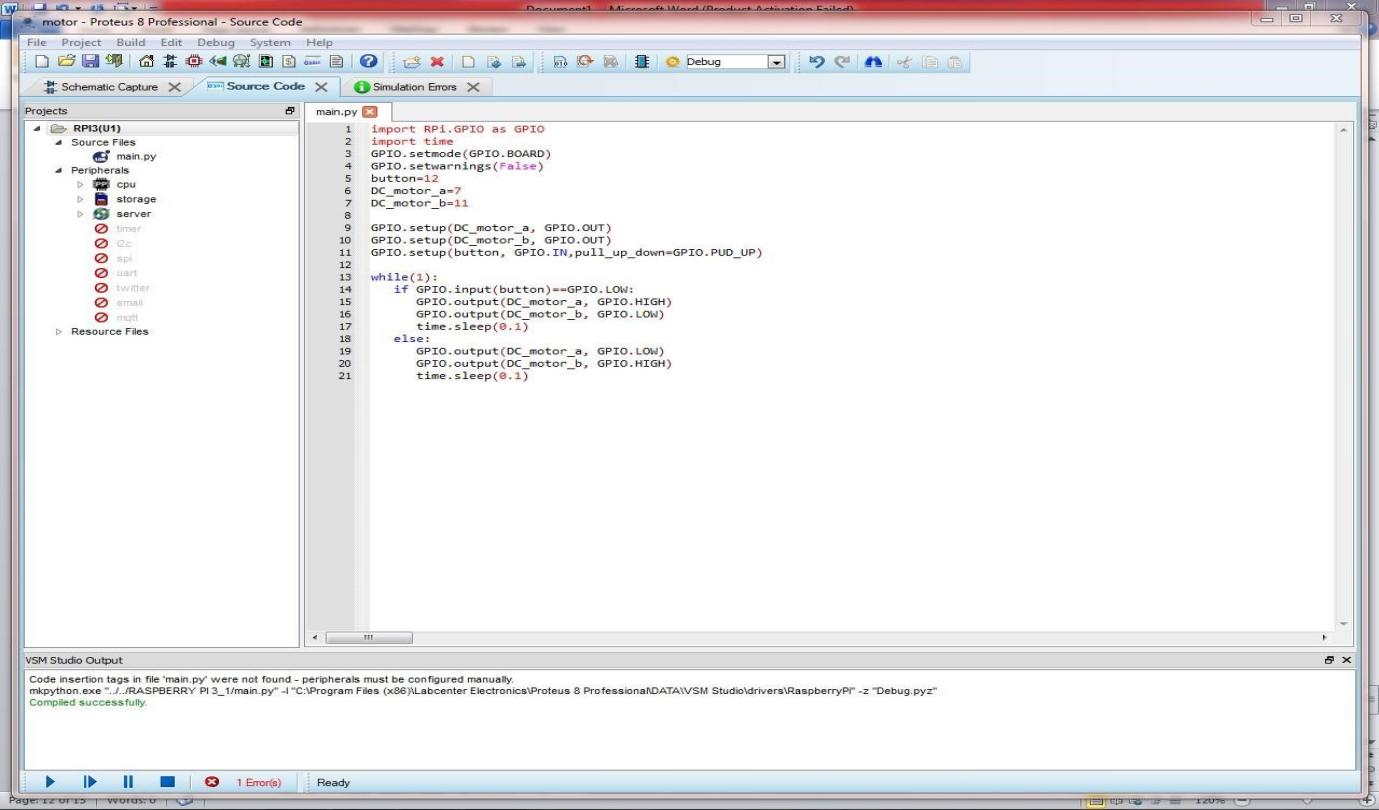
**Rotate button in clockwise**

****

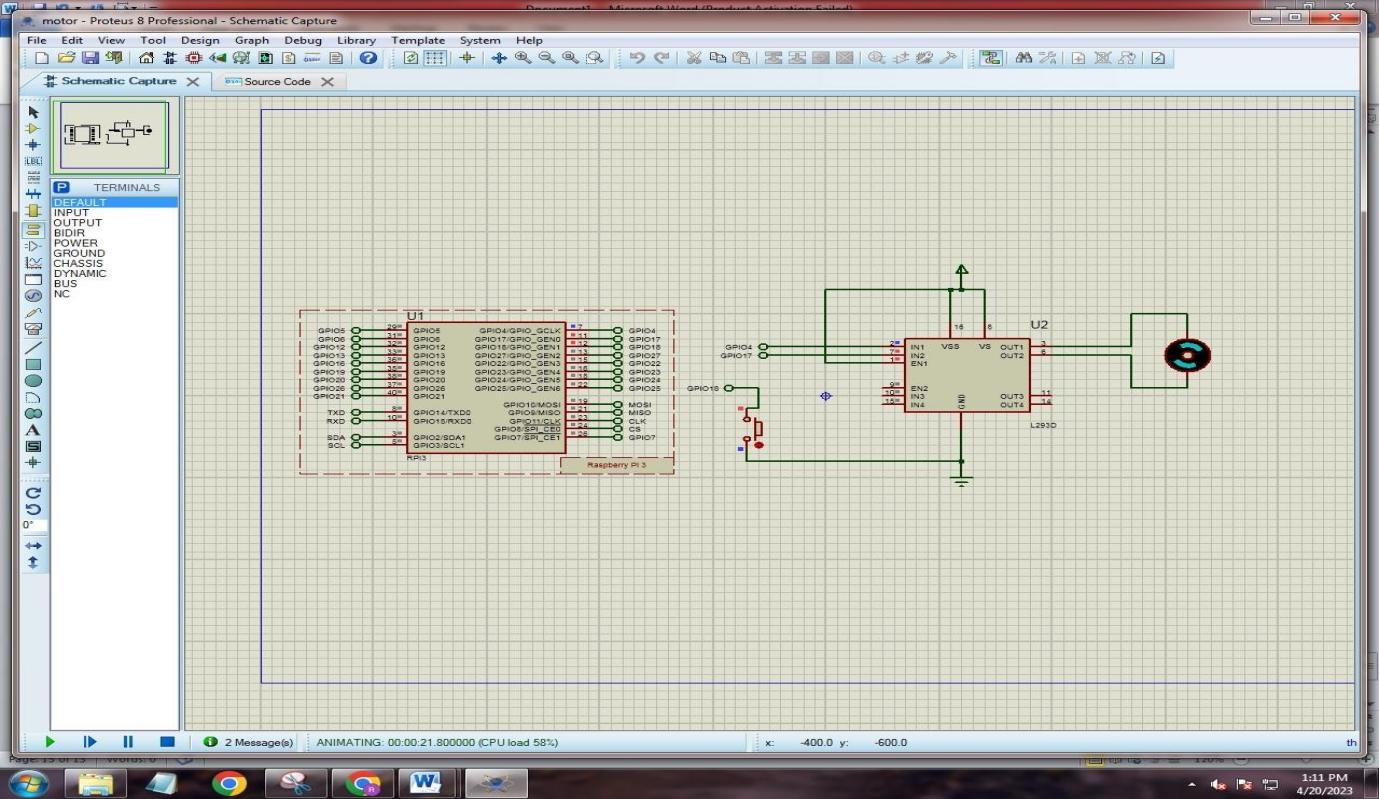
**Edit terminal label as GPIO18**

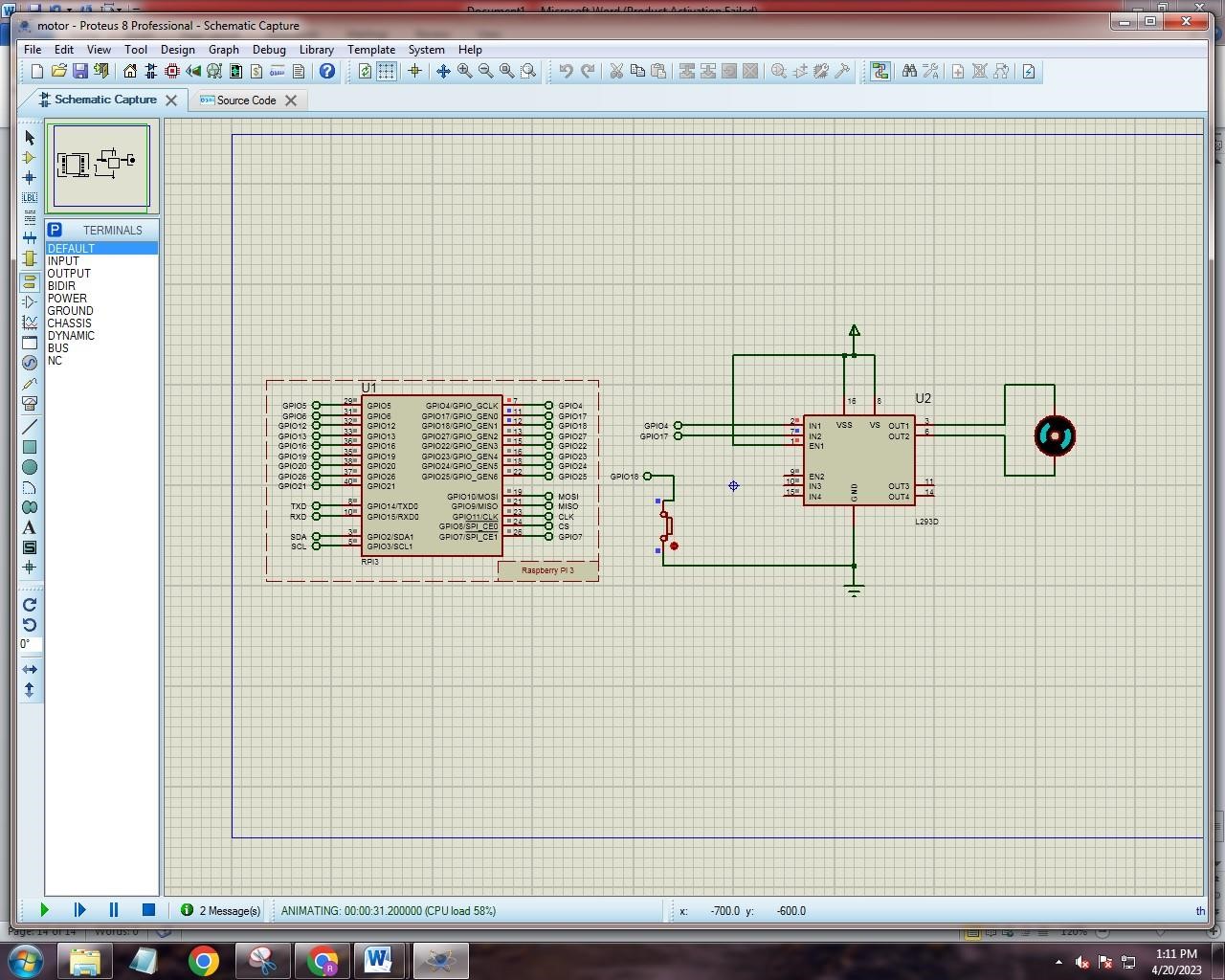
****

**Type code in proteus source code page**

****

**Start Simulation.**

****

****

**Practical No.4**

**Aim: Develop Python code for testing the sensors.**

**Step 1: Place the following component in TinkerCard.**

**Compontes:**

1. **PIR Sensor**
2. **Resistor**
3. **Piezo**
4. **Arduino Uno R3**
5. **LED RGB**

**Step 2:Type the following code int pirsensor = 0; void setup()**

{

pinMode(2, INPUT); pinMode(12,

OUTPUT); pinMode(13, OUTPUT);

} void loop() { pirsensor = digitalRead(2); if (pirsensor == HIGH)

{

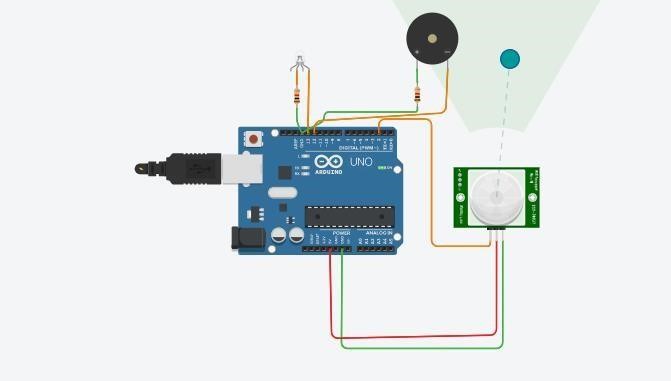
digitalWrite(13,HIGH); tone(12,500,500);

}

digitalWrite(13,LOW)

}

**Ouput:**

****

**Practical No: 5**

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

**Step 1:** Download the following libraries

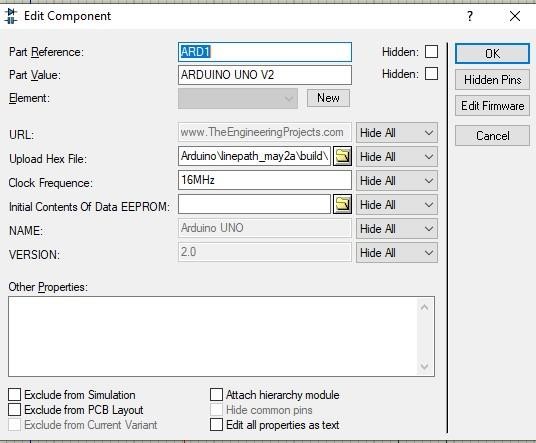
1. L298 Motor Driver Library for Proteus
2. Arduino UNO Library for Proteus V2.0
3. Infrared Sensor Library for Proteus

**Step 2:** Extract this files and pest it in Proteus libraries folder and restart the Proteus .

**Step 3:** Download the Proteus IDE and type the following code after that save it and Export Complied

**Library**

**Step 4:** Upload this HEX file in Arduino

****

Code: void setup()

{

pinMode(2,INPUT); pinMode(3,INPUT); pinMode(10,OUTPUT); pinMode(11,OUTPUT);

pinMode(12,OUTPUT); pinMode(13,OUTPUT);

}

void loop(){

int v=digitalRead(2); int

s=digitalRead(3);

if(v==1 and s==1){

digitalWrite(13,1); digitalWrite(12,0); digitalWrite(11,1); digitalWrite(10,0);

}

if(v==1 and s==0){

digitalWrite(13,0); digitalWrite(12,1);

digitalWrite(11,0); digitalWrite(10,1); }

if(v==0 and s==1){

digitalWrite(13,1); digitalWrite(12,0); digitalWrite(11,0); digitalWrite(10,1);

}

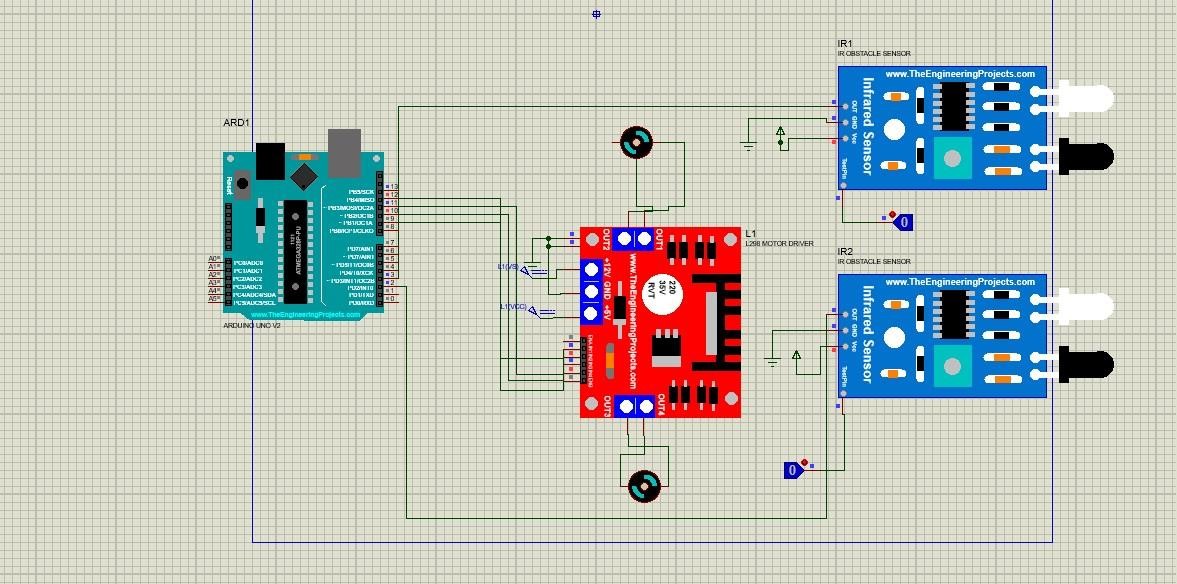
if(v==0 and s==0){

digitalWrite(13,0); digitalWrite(12,1);

digitalWrite(11,0); digitalWrite(10,1);

}

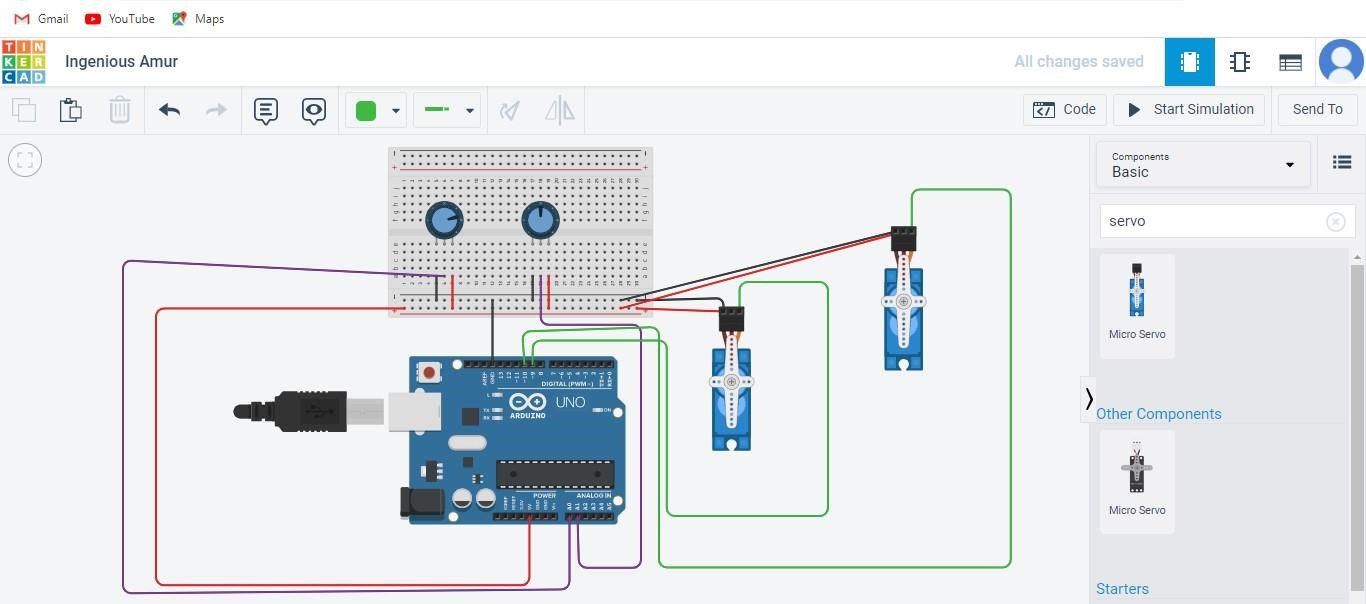
}

**Output: **

**Practical No. 6**

**Aim:** Add pan and tilt service to the robot object and test it.

**Take Arduino Uno R3 , Breadboard, two Servo motor and make connections: Take two potentiometer and change resistance to 10: Join wires:**

****

**Code:**

#include <Servo.h> int sensorValue = 0; int outputValue = 0; int sensorValue1 = 0; int outputValue1 = 0; Servo servo\_9; Servo

servo\_10; void setup()

{

pinMode(A0, INPUT); servo\_9.attach(9, 500,

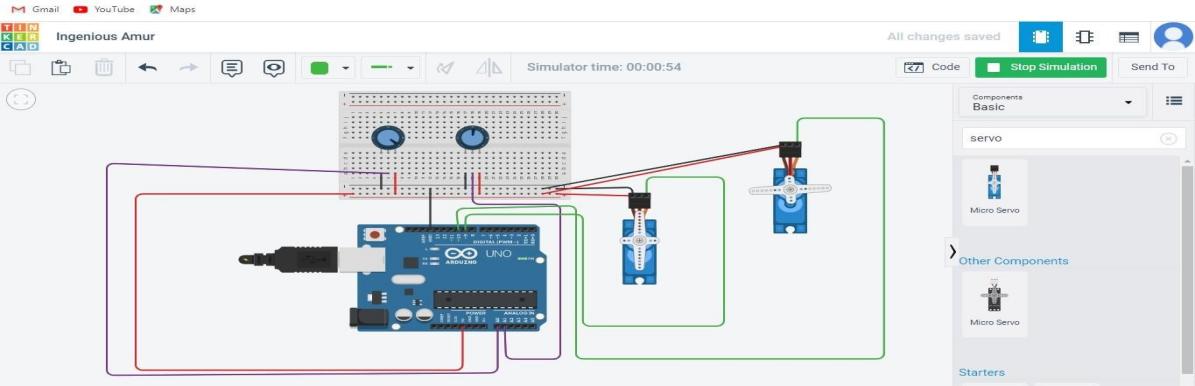
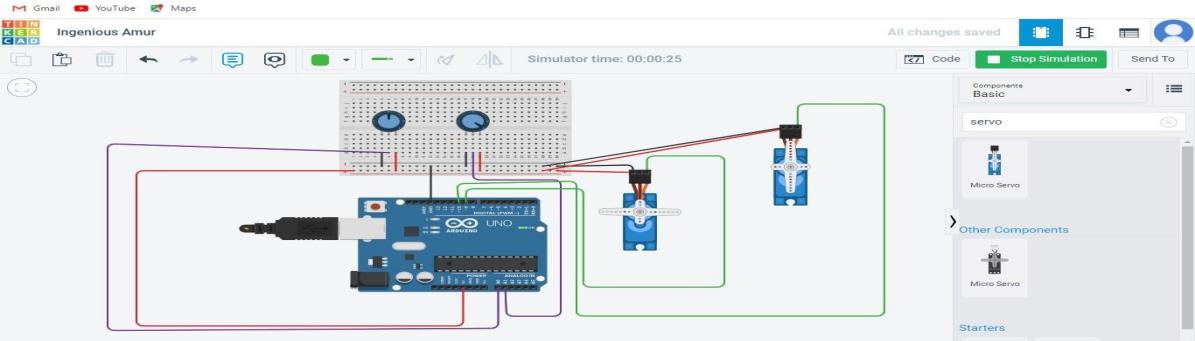
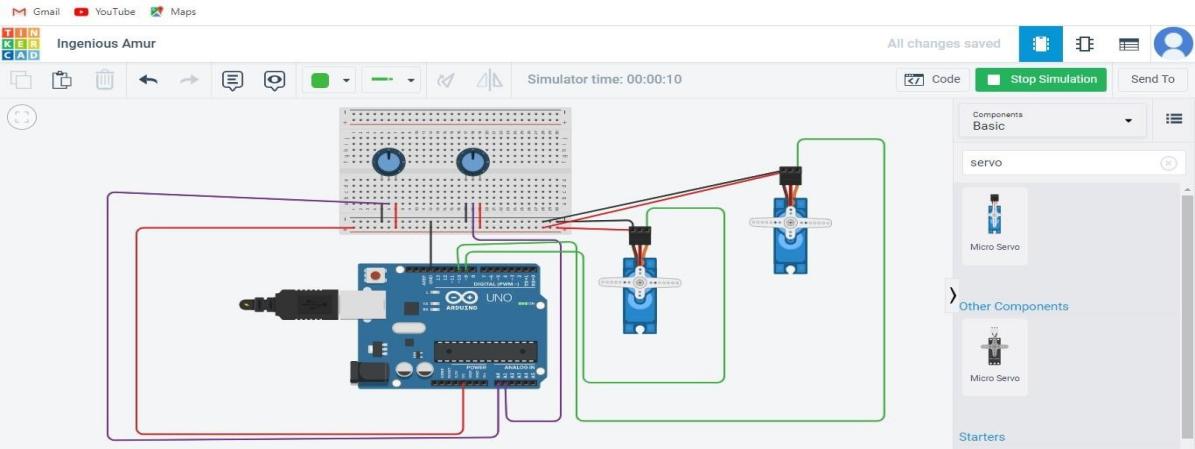
2500); pinMode(A1, INPUT); servo\_10.attach(10, 500, 2500); } void loop()

{

sensorValue = analogRead(A0); outputValue = map(sensorValue, 0, 1023, 0, 180); servo\_9.write(outputValue); delay(10); // Delay a little bit to improve simulation performance sensorValue1 = analogRead(A1); outputValue1 = map(sensorValue1, 0, 1023, 0, 180); servo\_10.write(outputValue1); delay(10); // Delay a little bit to improve simulation performance

}

**Output:**

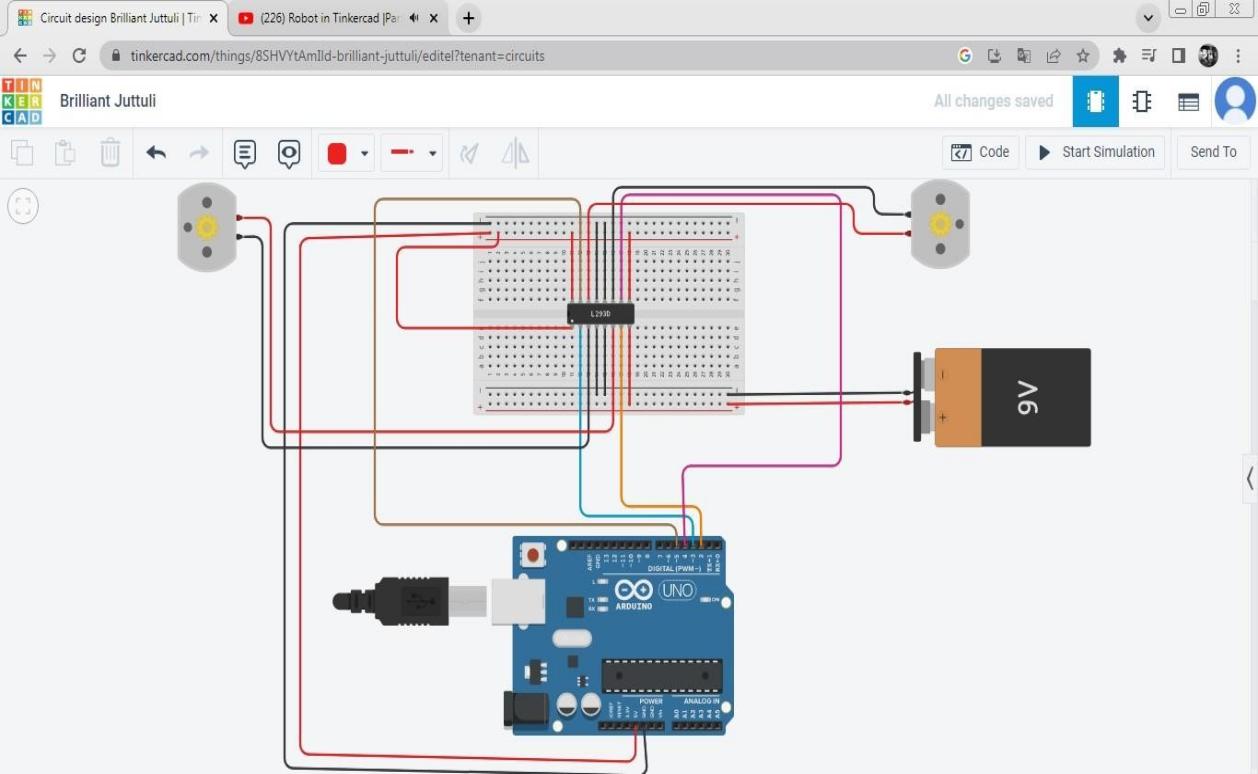


**Practical No. 7**

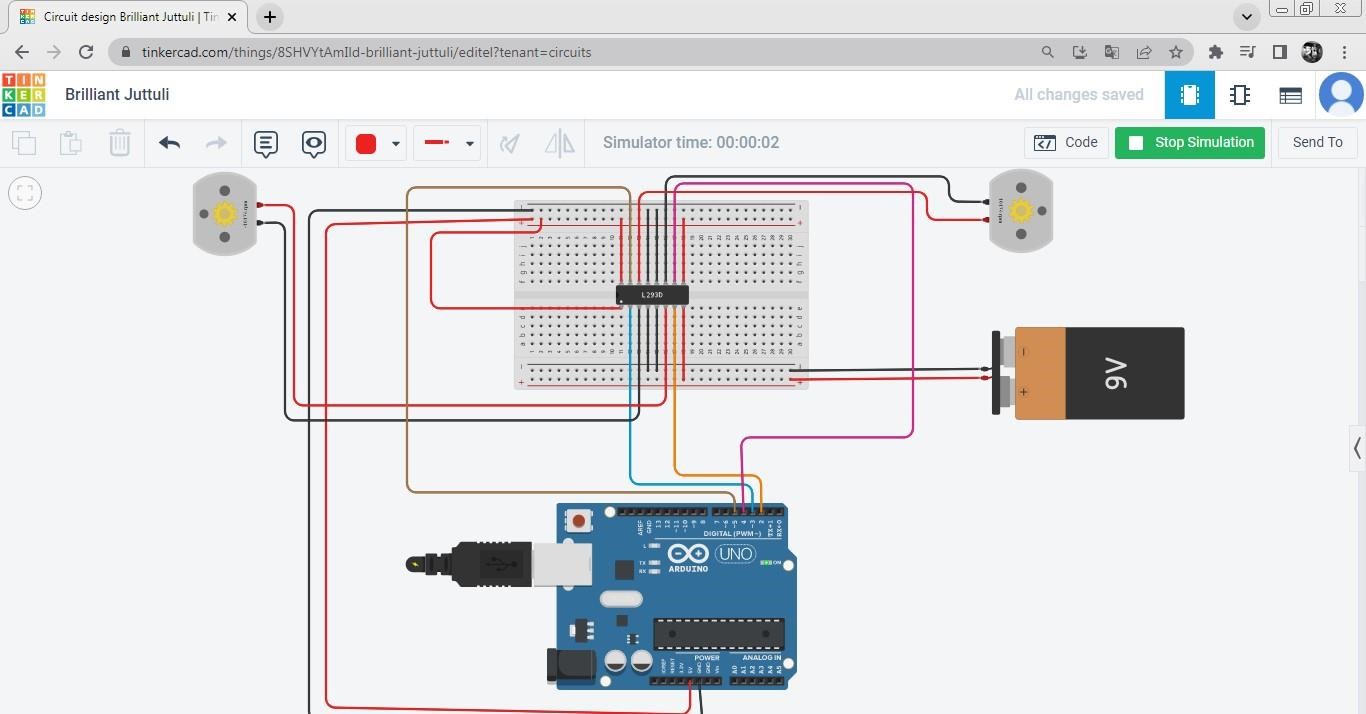
**Aim:** Create an obstacle avoidance behavior for robot and test it.

**Part A]**

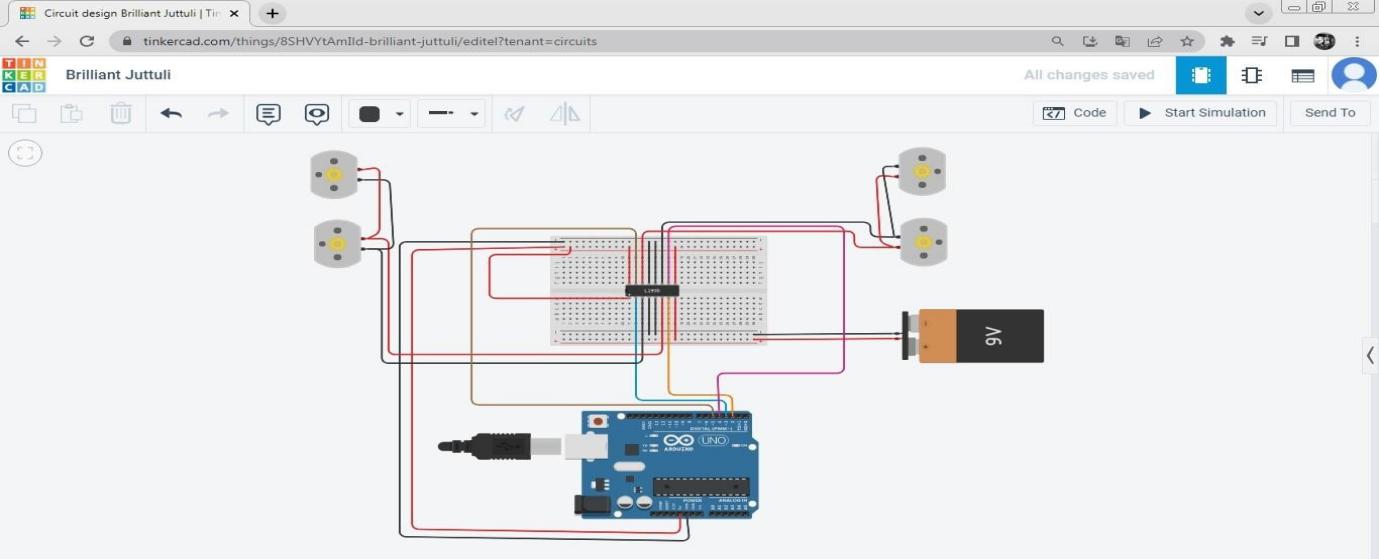
**Take DC Motor and 9 Volt Battery, 2 Dc motors, Ardino UNO R3, 9V Battery and Breadboard small: Connect wires:**

****

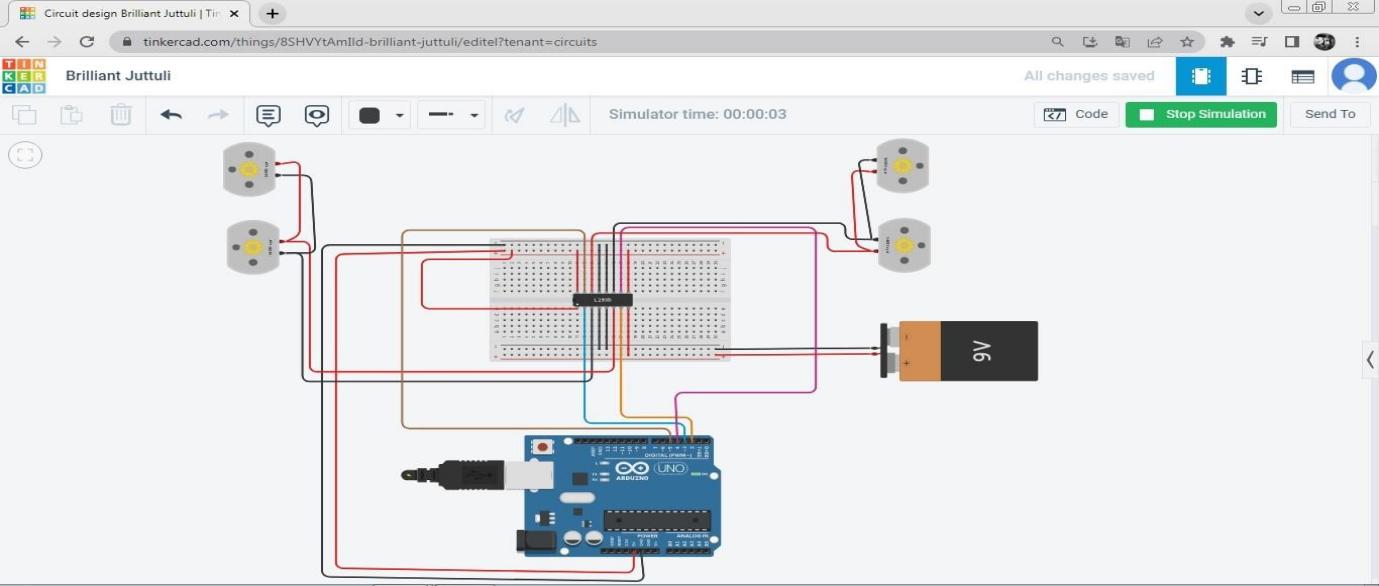
**Output:**

****

**Take 2 more motors & connect wires:**

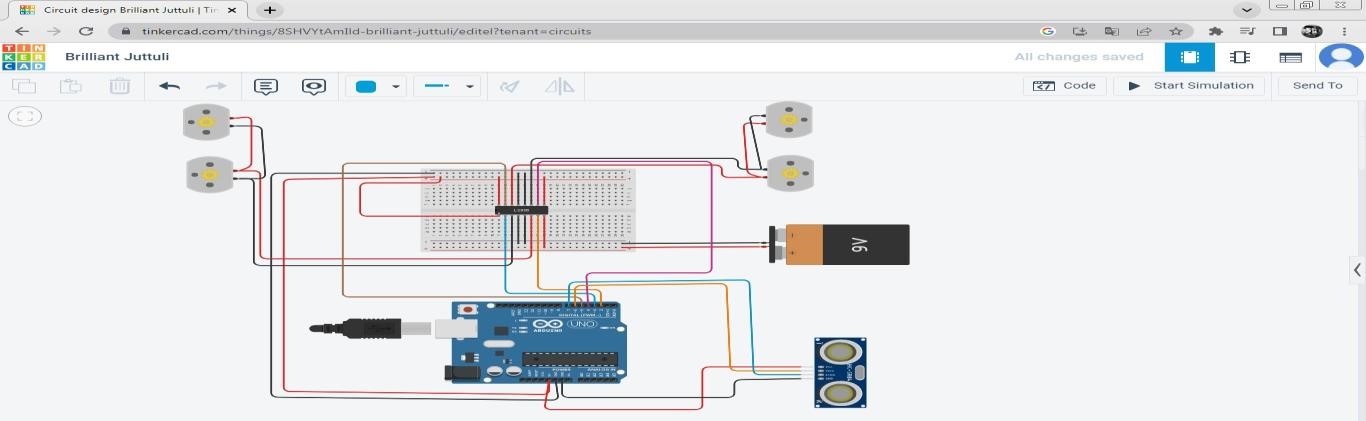
****

**Output:**

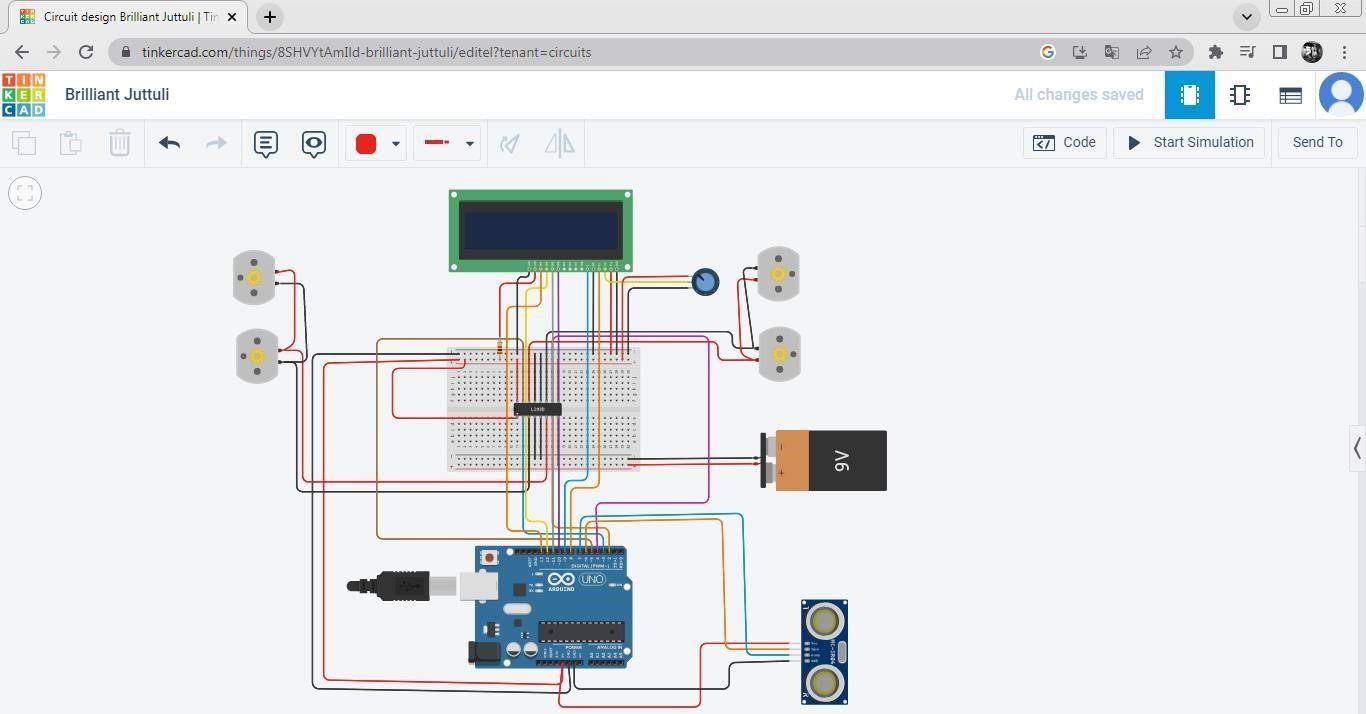
****

**Part B]**

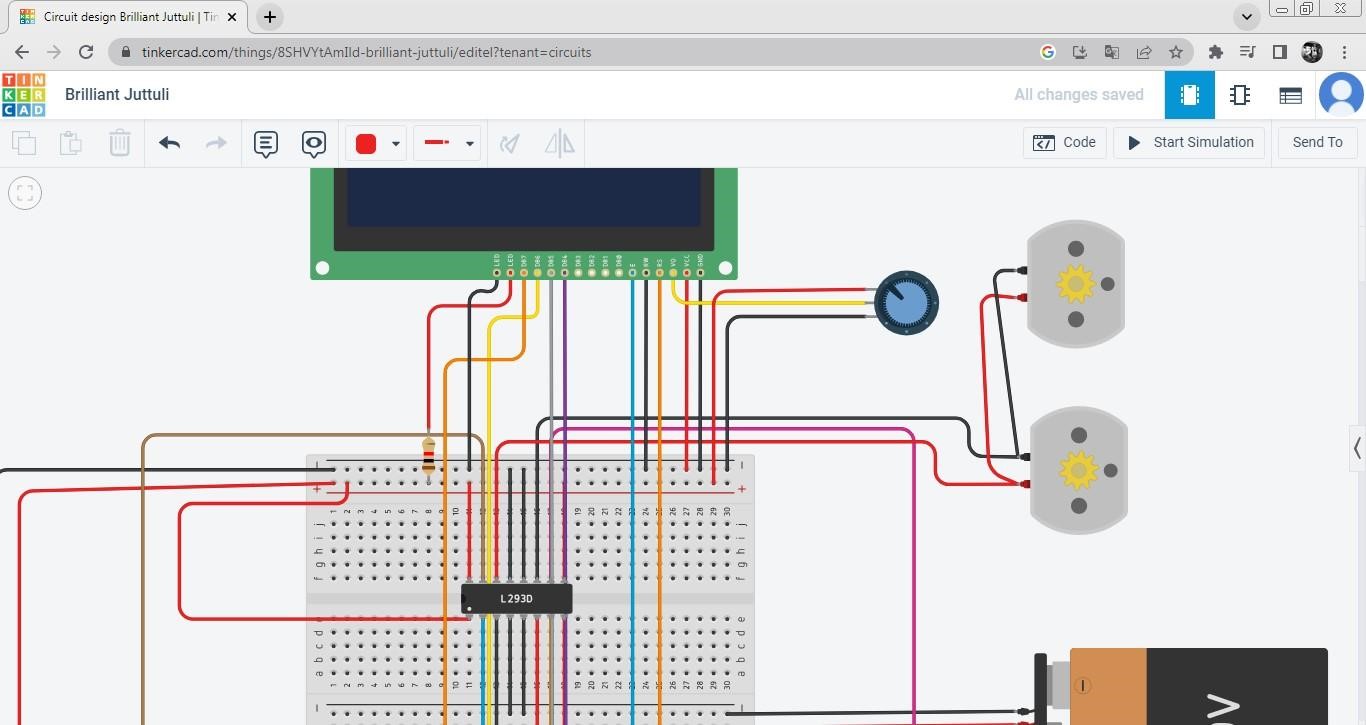
**Take ultrasonic distance and connect :**

****

**Take LCD, Potentiometer and connect devices:**

****

**Take resistor and connect:**

****

Write code:

//code for obstacle avoiding robot #include <LiquidCrystal.h> LiquidCrystal lcd(8,9,10,11,12,13); long cm, duration; const int echoPin = 7; const int trigPin = 6; const int lm1 = 2; const int lm2 = 3; const int rm3 = 4; const int rm4 = 5; void setup()

{

pinMode(lm1, OUTPUT); pinMode(lm2, OUTPUT); pinMode(rm1, OUTPUT); pinMode(rm2, OUTPUT); pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); Serial.begin(9600); lcd.begin(16,2);

} void loop()

{ digitalWrite(trigPin, LOW); delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(5); digitalWrite(trigPin, LOW); duration = pulseIn(echoPin,HIGH); //converting time into distance in centimetre cm = duration\*0.034/2; if(cm < 20)

{ stop\_bot(); delay(2000); go\_back(); delay(2000); stop\_again(); delay(1000); go\_left(); delay(1000);

} else { go\_straight(); delay(1000);

}

Serial.print("Distance:CM");

Serial.println(cm);

}

void go\_straight()

{

lcd.setCursor(0,0); lcd.print("NOTHING AHEAD"); lcd.setCursor(0,1); lcd.print("MOVING FORWARD"); digitalWrite(lm1,HIGH); digitalWrite(lm2,LOW); digitalWrite(rm1,HIGH);

digitalWrite(rm2,LOW);

} void go\_back()

{ lcd.clear(); lcd.setCursor(0,0);

lcd.print("TAKING REVERSE"); lcd.setCursor(0,1); lcd.print(cm); digitalWrite(lm2,HIGH); digitalWrite(lm1,LOW); digitalWrite(rm2,HIGH);

digitalWrite(rm1,LOW);

}

void stop\_bot() { lcd.clear(); lcd.setCursor(0,0);

lcd.print("SOMETHING AHEAD");

lcd.setCursor(0,1); lcd.print("STOP!"); digitalWrite(lm1,LOW); digitalWrite(lm2,LOW); digitalWrite(rm1,LOW);

digitalWrite(rm2,LOW);

} void stop\_again() { lcd.clear(); lcd.setCursor(0,0); lcd.print("BREAK FOR TURN"); digitalWrite(lm1,LOW); digitalWrite(lm2,LOW); digitalWrite(rm1,LOW);

digitalWrite(rm2,LOW);

} void go\_left()

{ lcd.clear(); lcd.setCursor(0,0); lcd.print("TURNING LEFT"); lcd.setCursor(0,1); lcd.print(cm); digitalWrite(lm1,LOW); digitalWrite(lm2,LOW); digitalWrite(rm1,HIGH);

digitalWrite(rm2,LOW);

} void go\_right()

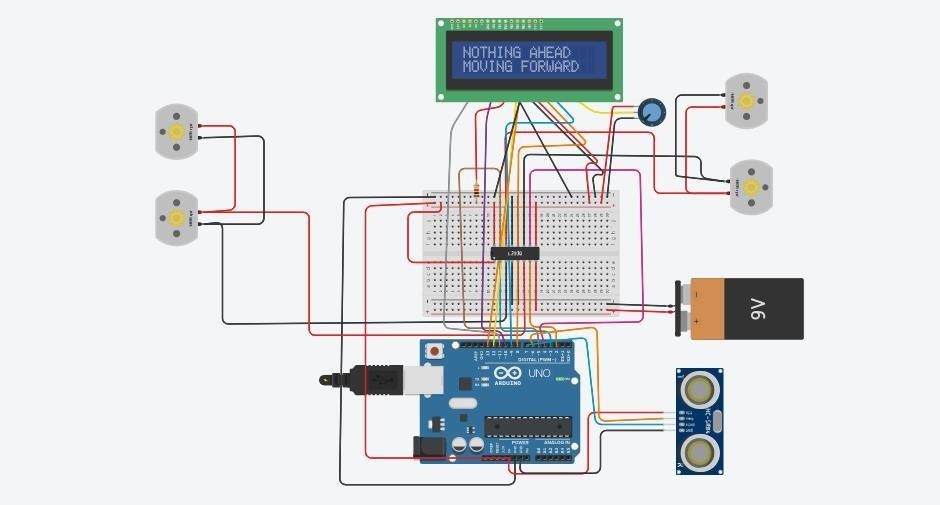
{ lcd.clear(); lcd.setCursor(0,0); lcd.print("TURNING RIGHT");

lcd.setCursor(0,1); lcd.print(cm); digitalWrite(lm1,HIGH); digitalWrite(lm2,LOW); digitalWrite(rm1,LOW);

digitalWrite(rm2,LOW);

}

**OUTPUT:**

****

**Practical No: 8**

**Aim: Detect faces with haar cascades.**

**CODE:**

# Importing OpenCV package import cv2

# Reading the image img = cv2.imread('test.jpg')

# Converting image to grayscale gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# Loading the required haar-cascade xml classifier file haar\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades

+ "haarcascade\_frontalface\_default.xml") eye\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_eye.xml')

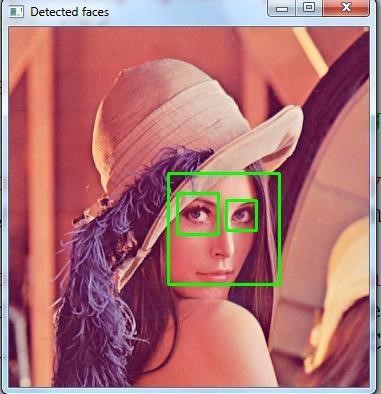
# Applying the face detection method on the grayscale image faces\_rect = haar\_cascade.detectMultiScale(gray\_img, 1.3, 5) eyes = eye\_cascade.detectMultiScale(gray\_img)

# Iterating through rectangles of detected faces for (x, y, w, h) in faces\_rect: cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2) for (ex,ey,ew,eh) in eyes: cv2.rectangle(img,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('Detected faces', img)

cv2.waitKey(0)

**OUTPUT:**

****