# 1. Study of WOKWI and Tinkercad.

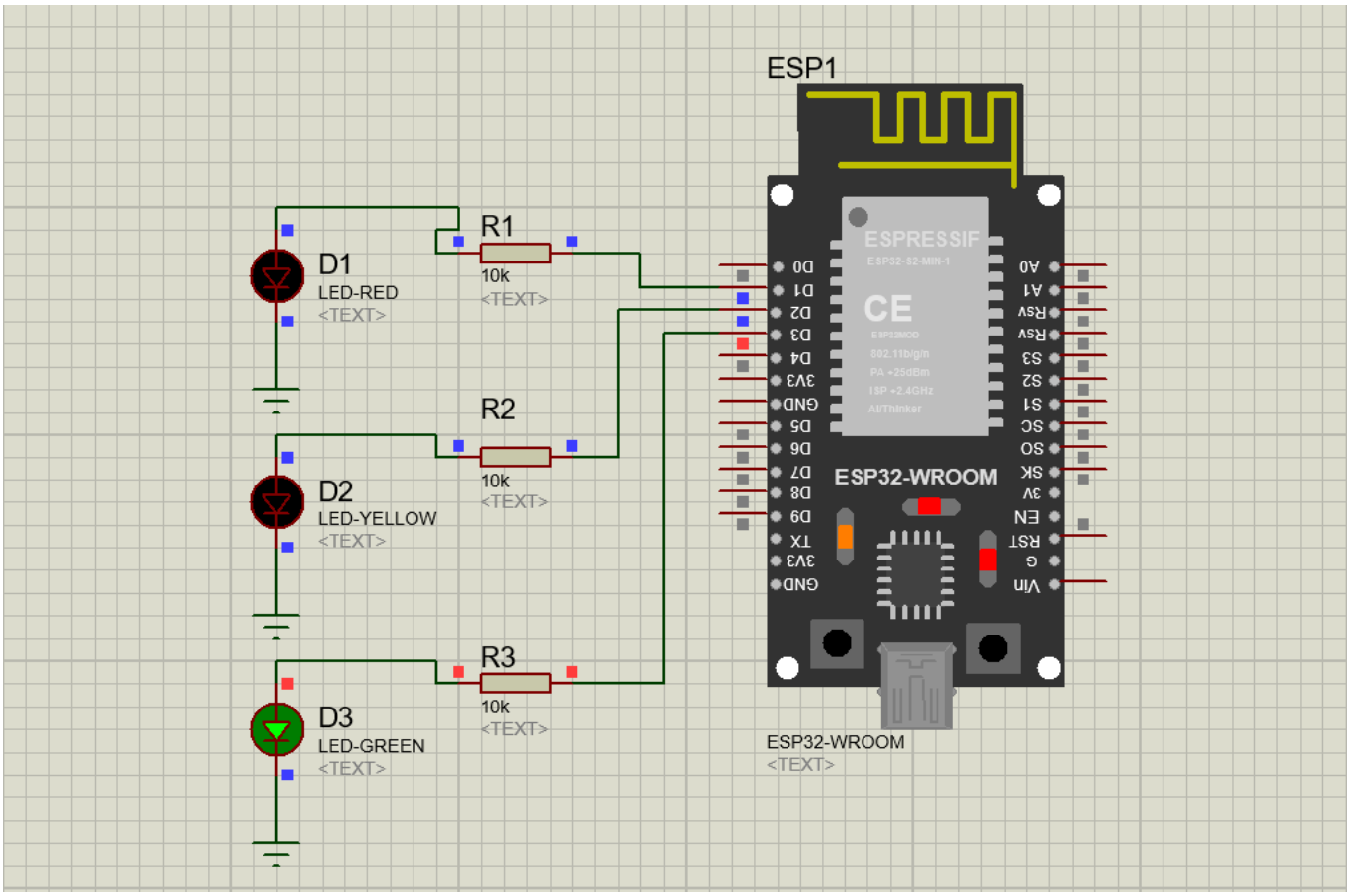
**Traffic Light:**

**Code:**

```
const int rled = 1;
const int oled = 2;
const int gled = 3;

void setup() {
  pinMode(rled, OUTPUT);
  pinMode(gled, OUTPUT);
  pinMode(oled, OUTPUT);
}

void loop() {
  digitalWrite(rled, HIGH);
  digitalWrite(gled, LOW);
  digitalWrite(oled, LOW);
  delay(1000);

  digitalWrite(gled, HIGH);
  digitalWrite(rled, LOW);
  digitalWrite(oled, LOW);
  delay(1000);

  digitalWrite(oled, HIGH);
  digitalWrite(gled, LOW);
```

```
  digitalWrite(rled, LOW);
  delay(1000);
}
```

**Circuit Diagram:**



# 2. Study of RPi Installation, OS Installation.

**Steps:**

1. **Format the SD Card**: First, we'll format the SD card using the SD Card Formatter tool.
2. **Install Raspbian Imager**: Next, we'll download and install Raspbian Imager on our computer.
3. **Select Raspberry Pi 4 – Other – Legacy 32-bit**: In Raspbian Imager, we'll select "Raspberry Pi 4" as the device. Then, under operating systems, we'll choose "Other" and select "Legacy 32-bit" from the options.
4. **Insert the Card into the Raspberry Pi and Plug it On**: After the image is written to the SD card, we'll safely eject it from our computer and insert it into the SD card slot on our Raspberry Pi. Then, we'll connect our Raspberry Pi to a power source to boot it up.
5. **Follow the On-Screen Setup**: Once the Raspberry Pi boots up, we'll follow the on-screen prompts to complete the initial setup, including selecting our country, language, time zone, creating a username/password, and configuring Wi-Fi settings if needed.
6. **Optional**: After setup, we may want to update the Raspberry Pi OS to ensure we have the latest software and security updates. To do this, we'll open a terminal and run the following commands:
    a. sudo apt update
    b. sudo apt upgrade
7. **Optional Configuration**: Depending on our needs, we may want to configure additional settings such as enabling SSH, setting up VNC, or installing specific software packages. We can do this through the Raspberry Pi Configuration tool or via the command line.

# 3. OS Installation, ready it from the network using WIFI and SSH, using SFTP upload files from PC.

**Steps (OS installation):**

1. **Card Reader & SD Card:** First, we'll insert the SD card into the card reader and connect it to our computer.
2. **Formatter:** Then, we'll format the SD card using a disk formatting tool. We can use the built-in formatting tool in our operating system or third-party software like SD Card Formatter.
3. **Install Raspbian Imager:** Next, we'll download and install the Raspbian Imager tool on our computer from the official Raspberry Pi website.
4. **Select Raspberry Pi 4 – Other – Legacy 32bit:** After installing Raspbian Imager, we'll open it and select the appropriate options for our Raspberry Pi model (Raspberry Pi 4) and architecture (Legacy 32-bit).
5. **Create an Empty SSH File:** Once the Raspbian image is written to the SD card, we'll eject it from our computer and reinsert it. Then, we'll create an empty file named "ssh" (without any file extension) in the root directory of the SD card. This will enable SSH access to the Raspberry Pi.
6. **Configure wpa_supplicant.conf for Wi-Fi:** We'll create a file named "wpa_supplicant.conf" in the boot directory of the SD card. Then, we'll add the necessary Wi-Fi configuration details to the file, such as our network SSID and password.
7. **Insert the SD Card:** With the files configured, we'll insert the SD card into the Raspberry Pi and power it on by plugging it in.
8. **Use Putty to Interact with the Raspberry Pi from Character User Interface:** Now, we'll open PuTTY on our computer and enter the IP address of the Raspberry Pi to establish an SSH connection. We'll use the default login credentials (username: pi, password: raspberry).
9. **Login and Use sudo raspi-config Command:** After logging in, we'll use the sudo raspi-config command to access the Raspberry Pi configuration tool. From here, we can configure various settings, including enabling VNC and SSH.
10. **Close VNC and SSH:** Once we've configured VNC and SSH settings, we'll exit the raspi-config tool and close the SSH connection.
11. **Use Advanced IP Scanner to Get the IP Address of Raspberry Pi:** If we don't already know the IP address of our Raspberry Pi, we can use Advanced IP Scanner or a similar tool to scan our network and identify the IP address assigned to the Raspberry Pi.
12. **Use RealVNC to Get Access to the Raspberry Pi:** We'll open the RealVNC Viewer on our computer and enter the IP address of the Raspberry Pi to establish a VNC connection. We'll use the same login credentials as SSH (username: pi, password: raspberry).
13. **Open Command Prompt and Install Idle3:** Once connected via VNC, we'll open a terminal window and install idle3 using the sudo apt install idle3 command. This will install the Python IDE on our Raspberry Pi.

Steps (File Transfer):

1. **Download and Install FileZilla:**
    - Go to the FileZilla website (https://filezilla-project.org/) and download the FileZilla Client for Windows.
    - Run the downloaded installer and follow the installation instructions to install FileZilla on your Windows computer.
2. **Open FileZilla:**
    - Once installed, open FileZilla from your Windows Start menu or desktop shortcut.
3. **Enter Raspberry Pi Details:**
    - In FileZilla, enter the following details:
        - Host: Enter the IP address of your Raspberry Pi.
        - Username: Enter the username (typically "pi" by default).
        - Password: Enter the password for the Raspberry Pi.
        - Port: Enter the port number for SSH (typically 22).
    - Click on the "Quickconnect" button.
4. **Successful Connection:**
    - If the login details are correct and the connection is successful, FileZilla will display a message indicating that you are connected to the Raspberry Pi.
5. **Upload File from Windows to Raspberry Pi:**
    - To upload a file from your Windows computer to the Raspberry Pi, navigate to the file on your local machine in the left-hand pane of FileZilla.
    - Navigate to the desired directory on your Raspberry Pi in the right-hand pane.
    - Drag and drop the file from the left-hand pane to the right-hand pane to initiate the upload.
6. **Download File from Raspberry Pi to Windows:**
    - To download a file from the Raspberry Pi to your Windows computer, navigate to the file on your Raspberry Pi in the right-hand pane of FileZilla.
    - Navigate to the desired directory on your local machine in the left-hand pane.
    - Drag and drop the file from the right-hand pane to the left-hand pane to initiate the download.

# 4. Develop python code for testing sensors and motors (RPi).

**For sensors:**

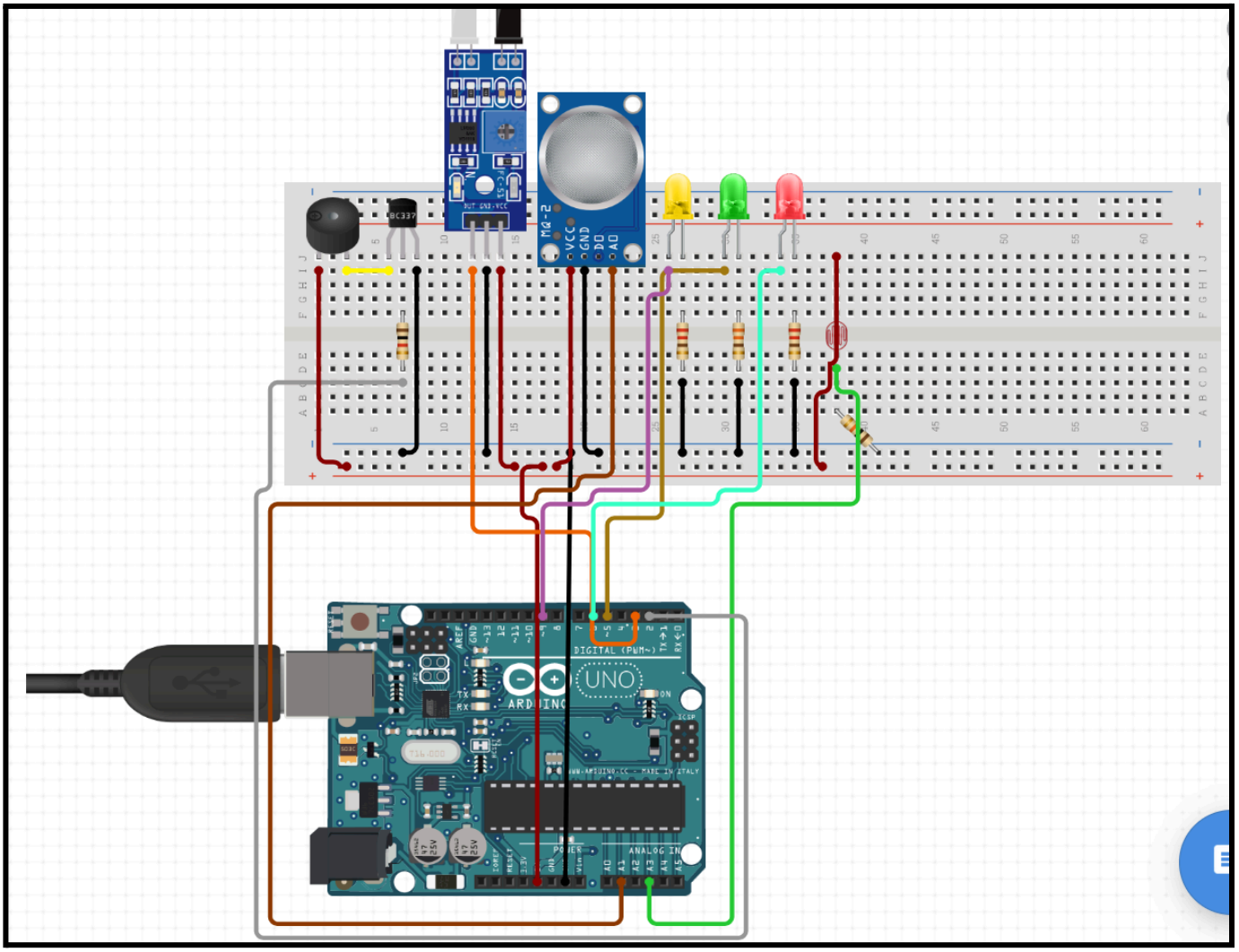
**Code:**

```
import RPi.GPIO as GPIO
import time


rled = 6
oled = 9
gled = 5
buzzer = 2
light =
obstacle = 8
gas = 16


GPIO.setmode(GPIO.BCM)
GPIO.setup(rled , GPIO.OUT)
GPIO.setup(oled , GPIO.OUT)
GPIO.setup(gled , GPIO.OUT)
GPIO.setup(buzzer , GPIO.OUT)
GPIO.setup(light, GPIO.IN)
GPIO.setup(obstacle, GPIO.IN)
GPIO.setup(gas, GPIO.IN)
GPIO.output(buzzer,false)


try:
    while True:
```

```
lightsense = GPIO.input(light)
obstaclesense = GPIO.input(obstacle)
gassense = GPIO.input(gas)
if lightsense==False:
    GPIO.output(rled,True)
    GPIO.output(gled,True)
    GPIO.output(oled,True)
    print("Light sensed")
if lightsense==True:
    GPIO.output(rled,False)
    GPIO.output(gled,False)
    GPIO.output(oled,False)
    print("Light not sensed")
if obstaclesense==True:
    GPIO.output(buzzer,True)
    print("Obstacle sensed")
    time.sleep(0.25)
    GPIO.output(buzzer,False)
    print("Obstacle sensed")
    time.sleep(0.25)
if obstaclesense==False:
    GPIO.output(buzzer,False)
    print("Obstacle not sensed")
if gassense==True:
    GPIO.output(buzzer,True)
    print("Gass Sensed")
    time.sleep(0.5);
```

```
        GPIO.output(buzzer,False)
        print("Gas Sensed")
        time.sleep(0.5);
    if gassense==False:
        GPIO.output(buzzer,False)
        print("Gass not sensed")


except KeyboardInterrupt:
    GPIO.cleanup()
```

**Circuit diagram:**



**For motors:**

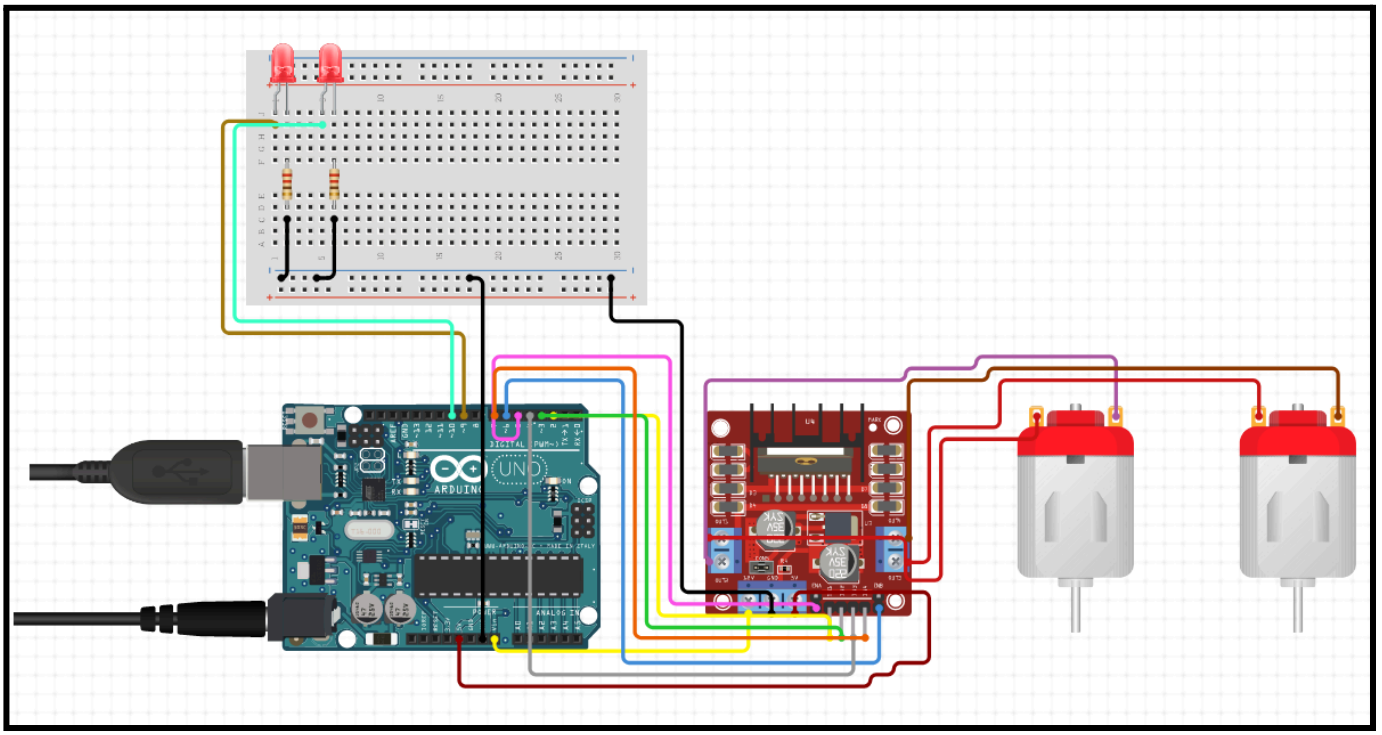
**Code:**

```
import RPi.GPIO as GPIO
import time

m11 = 2
m12 = 3
m21 = 4
m22 = 6
led1 = 10
led2 = 9

GPIO.setmode(GPIO.BOARD)
GPIO.setup(m11, GPIO.OUT)
GPIO.setup(m12, GPIO.OUT)
GPIO.setup(m21, GPIO.OUT)
GPIO.setup(m22, GPIO.OUT)
GPIO.setup(led1, GPIO.OUT)
GPIO.setup(led2, GPIO.OUT)

try:
    while True:
        GPIO.output(m11, GPIO.HIGH)
        GPIO.output(m12, GPIO.LOW)
        GPIO.output(m21, GPIO.HIGH)
        GPIO.output(m22, GPIO.LOW)
```

```
GPIO.output(led1, GPIO.HIGH)
GPIO.output(led2, GPIO.HIGH)
time.sleep(5)

GPIO.output(m11, GPIO.LOW)
GPIO.output(m12, GPIO.HIGH)
GPIO.output(m21, GPIO.LOW)
GPIO.output(m22, GPIO.HIGH)
GPIO.output(led1, GPIO.LOW)
GPIO.output(led2, GPIO.LOW)
time.sleep(5)

GPIO.output(m11, GPIO.HIGH)
GPIO.output(m12, GPIO.LOW)
GPIO.output(m21, GPIO.LOW)
GPIO.output(m22, GPIO.HIGH)
GPIO.output(led1, GPIO.LOW)
GPIO.output(led2, GPIO.HIGH)
time.sleep(5)

GPIO.output(m11, GPIO.LOW)
GPIO.output(m12, GPIO.HIGH)
GPIO.output(m21, GPIO.HIGH)
GPIO.output(m22, GPIO.LOW)
GPIO.output(led1, GPIO.HIGH)
GPIO.output(led2, GPIO.LOW)
time.sleep(5)
```

```
        GPIO.output(m11, GPIO.HIGH)
        GPIO.output(m12, GPIO.HIGH)
        GPIO.output(m21, GPIO.HIGH)
        GPIO.output(m22, GPIO.HIGH)
        GPIO.output(led1, GPIO.LOW)
        GPIO.output(led2, GPIO.LOW)
        time.sleep(5)

except KeyboardInterrupt:
    GPIO.cleanup()
```

**Circuit diagram:**



# 5. Write a script to follow a predetermined path (ESP32).

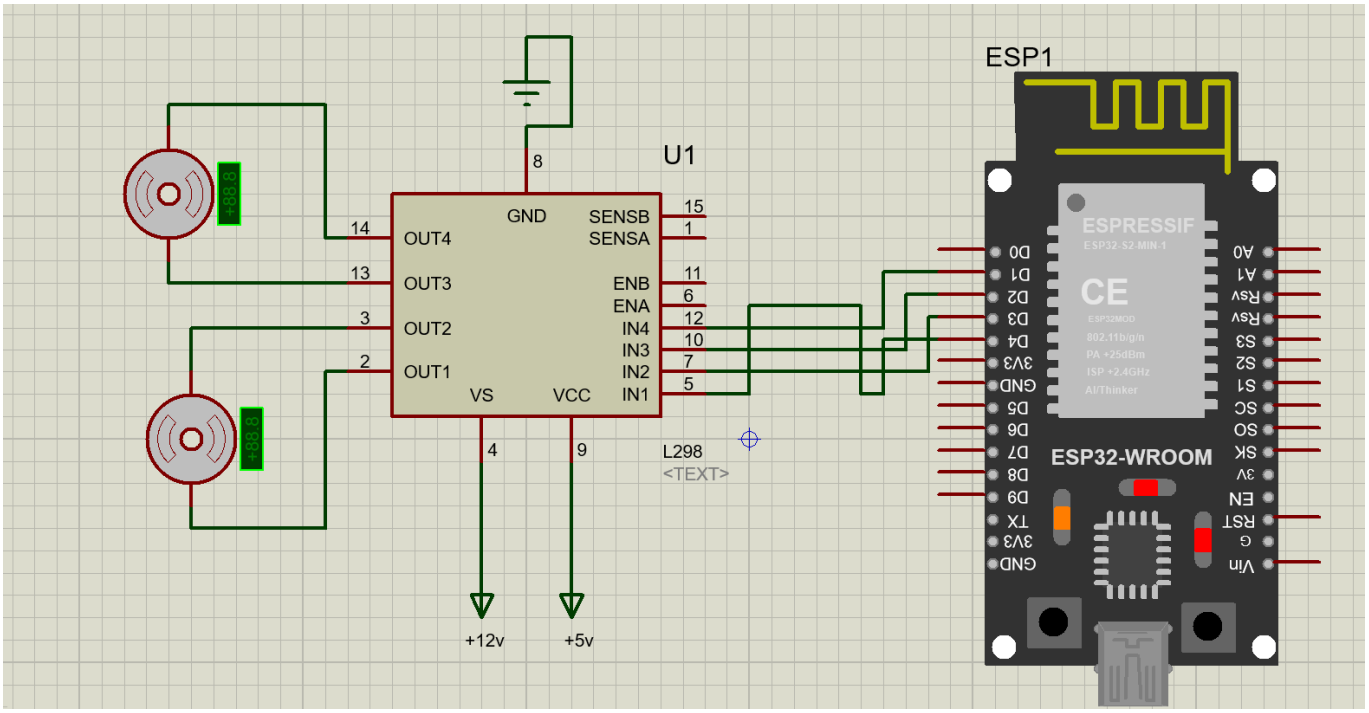
**Code:**

```
const int m11 = 11;
const int m12 = 10;
const int m21 = 7;
const int m22 = 6;

void moveForward() {
    digitalWrite(m11, HIGH);
    digitalWrite(m12, LOW);
    digitalWrite(m21, HIGH);
    digitalWrite(m22, LOW);
}

void turnRight() {
    digitalWrite(m11, LOW);
    digitalWrite(m12, HIGH);
    digitalWrite(m21, HIGH);
    digitalWrite(m22, LOW);
}

void turnLeft() {
    digitalWrite(m11, HIGH);
    digitalWrite(m12, LOW);
    digitalWrite(m21, LOW);
    digitalWrite(m22, HIGH);
```

```
}
void stop() {
    digitalWrite(m11, LOW);
    digitalWrite(m12, LOW);
    digitalWrite(m21, LOW);
    digitalWrite(m22, LOW);
}

void setup() {
      pinMode(m11, OUTPUT);
    pinMode(m12, OUTPUT);
    pinMode(m21, OUTPUT);
    pinMode(m22, OUTPUT);
}

void loop() {
    // Move forward for 5 seconds
    moveForward();
    delay(5000);

    // Turn right for 2 seconds
    turnRight();
    delay(2000);

    // Move forward for 5 seconds
    moveForward();
    delay(5000);
```

```
    // Turn right for 2 seconds
    turnRight();
    delay(2000);

    // Move forward for 5 seconds
    moveForward();
    delay(5000);

    // Turn right for 2 seconds
    turnRight();
    delay(2000);

    // Move forward for 5 seconds
    moveForward();
    delay(5000);

    // Turn right for 2 seconds to return to the starting position
    turnRight();
    delay(2000);

    // Stop briefly
    stop();
    delay(1000);
}
```

## Circuit Diagram:



# 6. Create obstacle avoidance behavior and test it (ESP32).

**Code:**

```
int m11 = 13;

int m12 =12;

int m21 =14;

int m22 = 27;


int ir_1 = 26;

int ir_2 = 25;


int obsense1;

int obsense2;


void setup()

{

  pinMode(m11,OUTPUT);

  pinMode(m12,OUTPUT);

  pinMode(m21, OUTPUT);

  pinMode(m22, OUTPUT);


  pinMode(ir_1,INPUT);

  pinMode(ir_2,INPUT);

}


void loop()

{
```

```
  obsense1 = digitalRead(ir_1);
  obsense2 = digitalRead(ir_2);

  if(obsense1 == 1 && obsense2 == 1){
    //Forward
    digitalWrite(m11,HIGH);
    digitalWrite(m12,LOW);
    digitalWrite(m21,HIGH);
    digitalWrite(m22,LOW);
//    delay(1000);
  }

  if(obsense1 == 0 && obsense2 == 0){
    //Forward
    digitalWrite(m11,LOW);
    digitalWrite(m12,HIGH);
    digitalWrite(m21,LOW);
    digitalWrite(m22,HIGH);
//    delay(1000);
  }

  if(obsense1 == 0 && obsense2 == 1){
    //Right
    digitalWrite(m11,HIGH);
    digitalWrite(m12,LOW);
    digitalWrite(m21,LOW);
```

```
    digitalWrite(m22,HIGH);
//     delay(1000);
  }

  if(obsense1 == 1 && obsense2 == 0){
    //Left
    digitalWrite(m11,LOW);
    digitalWrite(m12, HIGH);
    digitalWrite(m21,HIGH);
    digitalWrite(m22, LOW);
//     delay(1000);
  }



  }
```

**Circuit Dlagram:**

# 7. Create a bluetooth, voice and gesture control robot and test it (ESP32).

**Code:**

```
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) ||
!defined(CONFIG_BLUEDROID_ENABLED)

#error Bluetooth is not enabled! Please run make menuconfig to and enable it

#endif


int m11 = 13;

int m12 =12;

int m21 =14;

int m22 = 27;

BluetoothSerial SerialBT;


void forward(){

  digitalWrite(m11,LOW);

  digitalWrite(m12,HIGH);

  digitalWrite(m21,LOW);

  digitalWrite(m22,HIGH);

  //delay(3000);

}

void stops(){

  //stop

  digitalWrite(m11,LOW);

  digitalWrite(m12,LOW);

  digitalWrite(m21,LOW);
```

```
  digitalWrite(m22,LOW);
//  delay(1300);
  }
void left(){
  //LEFT
  digitalWrite(m11,LOW);
  digitalWrite(m12,LOW);
  digitalWrite(m21,LOW);
  digitalWrite(m22,HIGH);
  delay(200);
  }
void right(){
  //rIGHT
  digitalWrite(m11,LOW);
  digitalWrite(m12,HIGH);
  digitalWrite(m21,LOW);
  digitalWrite(m22,LOW);
  delay(200);
  }
void reverse(){
  digitalWrite(m11,HIGH);
  digitalWrite(m12,LOW);
  digitalWrite(m21,HIGH);
  digitalWrite(m22,LOW);
  delay(300);
  }
void setup() {
```

```
  pinMode(m11,OUTPUT);
  pinMode(m12,OUTPUT);
  pinMode(m21, OUTPUT);
  pinMode(m22, OUTPUT);
  Serial.begin(115200);
  SerialBT.begin("ESP32test"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");
//  pinMode(LED_BUILTIN, OUTPUT);
}

char keyvalue;
void loop() {
  if (SerialBT.available()) {
    keyvalue = SerialBT.read();
  //  SerialBT.write(keyvalue.toString);
  Serial.println(keyvalue);

  if (keyvalue=='F' || keyvalue=='f' || keyvalue=='U')
  {

    Serial.println("Foward Movement.....");
    digitalWrite(m11,LOW);
    digitalWrite(m12,HIGH);
    digitalWrite(m21,LOW);
    digitalWrite(m22,HIGH);
     }
    if(keyvalue=='L' || keyvalue=='l')
```

```
 {
  Serial.println("Left Movement.....");
  left();
  }
  if(keyvalue=='R' || keyvalue=='r')
 {
  Serial.println("right Movement.....");
  right();
  }
  if(keyvalue=='B' || keyvalue=='b' || keyvalue=='D')
 {
  Serial.println("Backward Movement.....");
  reverse();
  }
 if(keyvalue=='S' || keyvalue=='s')
 {
  Serial.println("Stop Movement.....");
  stops();
  }
}
//delay(20);
}
```

**Circuit Dlagram:**

## 8. Create a line following robot and test it (ESP32).

**Code:**

**Circuit diagram:**

## 9. Create an edge following and test it (ESP32).

**Code:**

**Circuit diagram:**