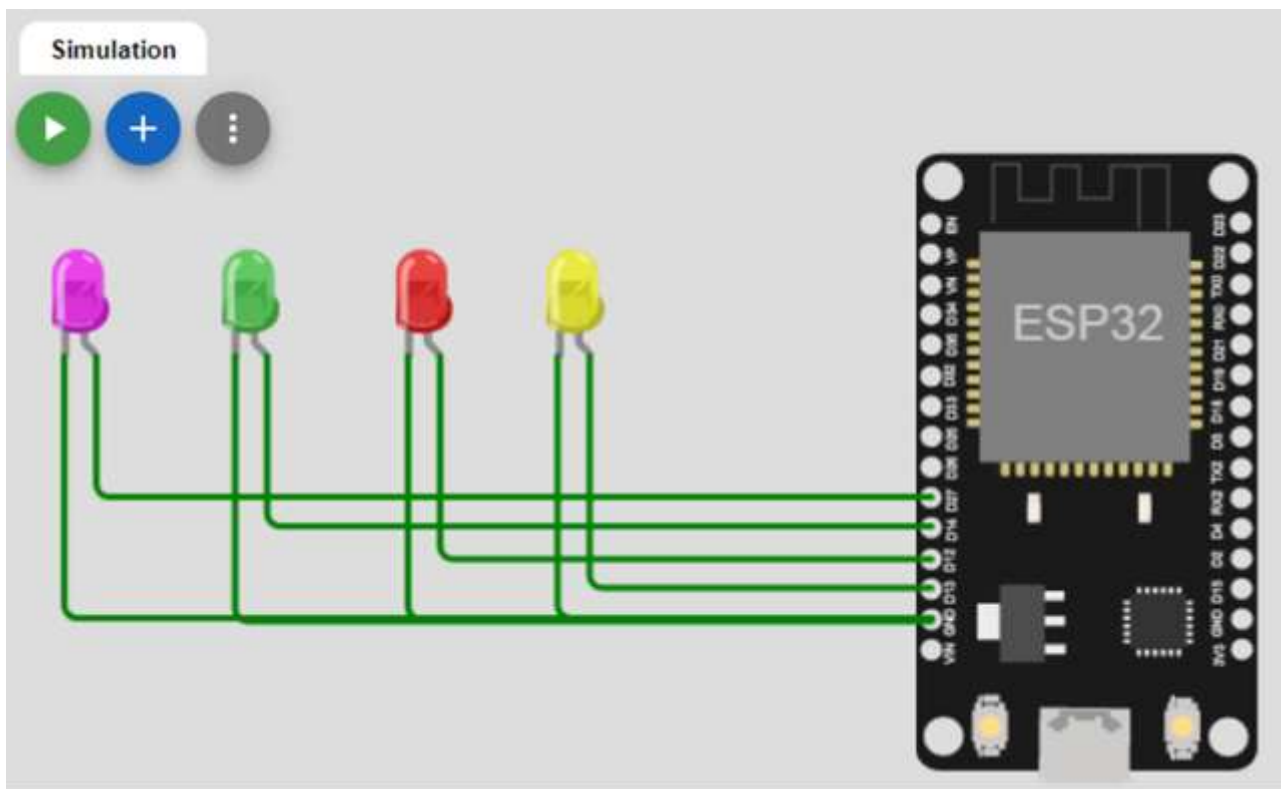


HOME WORKE 1

Design the electronic circuit and write programing code by use ESP32 controller to operate 4 leds indicates the direction of robot movement (violet=right, green=forward, red=stop, yellow=left) by accessing the controller via a web page.

Solution:

The electronic circuit



The programming code

```
#include <WiFi.h>
#include <HTTPClient.h>
const char* ssid = "Wokwi-GUEST";
const char* password = "";
const String url = "https://s-m.com.sa/f.html";
String payload = "";
void setup() {
    Serial.begin(115200);
    Serial.println("Hello, ESP32!");
    Serial.println("traffic light indicates the direction of robot movement
(violet=right, green=forward, red=stop, yellow=left)");
    WiFi.begin(ssid, password);
    pinMode(27, OUTPUT);
    pinMode(14, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
    Serial.print("Connecting to WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.print("OK! IP=");
    Serial.println(WiFi.localIP());

    Serial.print("Fetching " + url + "... ");

}

void loop(){
    HTTPClient http;
    http.begin(url);
    int httpStatusCode = http.GET();
    if (httpStatusCode > 0) {
        Serial.print("HTTP ");
        Serial.println(httpStatusCode);
        payload = http.getString();
        Serial.println();
        Serial.println(payload);
        if( payload == "forward" ){
            digitalWrite(14, HIGH);
        }
        if( payload == "stop" ){
            digitalWrite(12, HIGH);
        }
        if( payload == "left" ){
            digitalWrite(27, HIGH);
        }
        if( payload == "right" ){
```

```

        digitalWrite(13, HIGH);
    }
}
else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
    Serial.println(":-(");
}
}

```

Execution in Simulation

The screenshot displays the VOKWI IDE interface. On the left, the code editor shows an Arduino sketch for an ESP32. The code includes headers for `WiFi.h` and `HTTPClient.h`, defines constants for SSID, password, and a URL, and implements a `setup()` function that initializes serial communication, connects to WiFi, and fetches data from the specified URL. The right side of the IDE features a simulation window. It shows a virtual circuit with an ESP32 board and four LEDs (violet, green, red, yellow) connected to pins 27, 14, 12, and 13 respectively. The serial monitor in the simulation window displays the output of the code, including 'Hello, ESP32!', 'Connecting to WiFi', and 'forward'.

```

1  #include <WiFi.h>
2  #include <HTTPClient.h>
3  const char* ssid = "Wokwi-GUEST";
4  const char* password = "";
5  const String url = "https://s-m.com.sa/f.html";
6  String payload = "";
7  void setup() {
8      Serial.begin(115200);
9      Serial.println("Hello, ESP32!");
10     Serial.println("traffic light indicates the direction of robot movement (violet=right, green=forward, red=stop, yellow=left)");
11     WiFi.begin(ssid, password);
12     pinMode(27, OUTPUT);
13     pinMode(14, OUTPUT);
14     pinMode(12, OUTPUT);
15     pinMode(13, OUTPUT);
16     Serial.print("Connecting to WiFi");
17     while (WiFi.status() != WL_CONNECTED) {
18         delay(500);
19         Serial.print(".");
20     }
21     Serial.print("OK! IP=");
22     Serial.println(WiFi.localIP());
23     Serial.print("Fetchine " + url + "... ");

```

Simulation

00:26.634 14%

ESP32

Hello, ESP32!
 traffic light indicates the direction of robot movement (violet=right, green=forward, red=stop, yellow=left)
 Connecting to WiFi.....OK! IP=10.10.0.2
 Fetching https://s-m.com.sa/f.html... HTTP 200
 forward