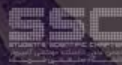


| 9th WSS General Meeting |

IoT WORKSHOP

By:

Mahdi Bahreiny



رند پیشرفت کارگاه

جلسه 2

اتصال ESP به شبکه و
کنترل آن به صورت بی
سیم و لوکال

جلسه 3

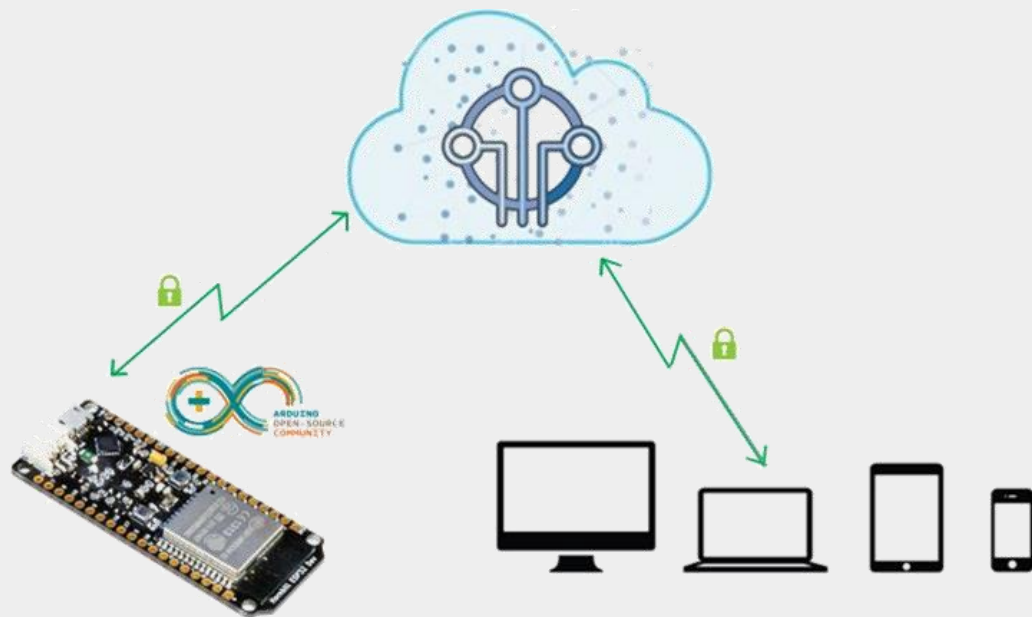
اتصال ESP به اینترنت
و کنترل آن به صورت
بی سیم و از راه دور

جلسه 1

راه اندازی ESP و
کنترل آن باسیم

جلسه 4

راه اندازی پنل کاربری
برای کنترل ESP از
طریق اینترنت



جلسه دوم:

اتصال ESP32 به وایفای و

راه اندازی رله

معرفی رله

- یک کلید الکتریکی است که وظیفه آن قطع و وصل کردن جریان برق پس از فرمان کلید است

الکترومغناطیسی

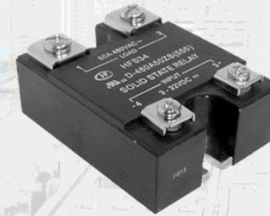
حالت جامد

حرارتی

زماندار



Electromagnetic relay



Solid state relay(SSR)



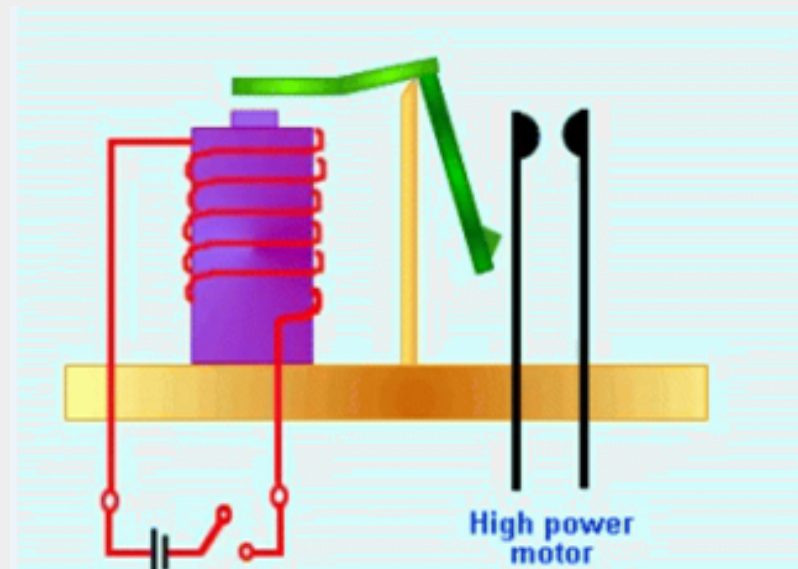
Thermal relay



Time relay

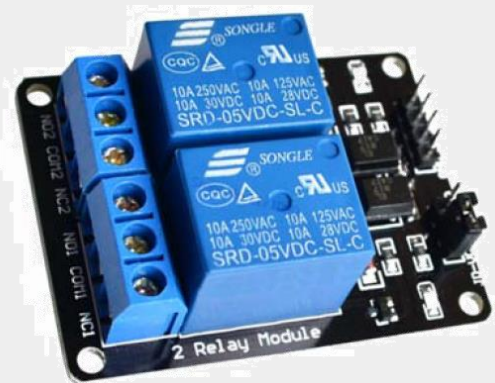
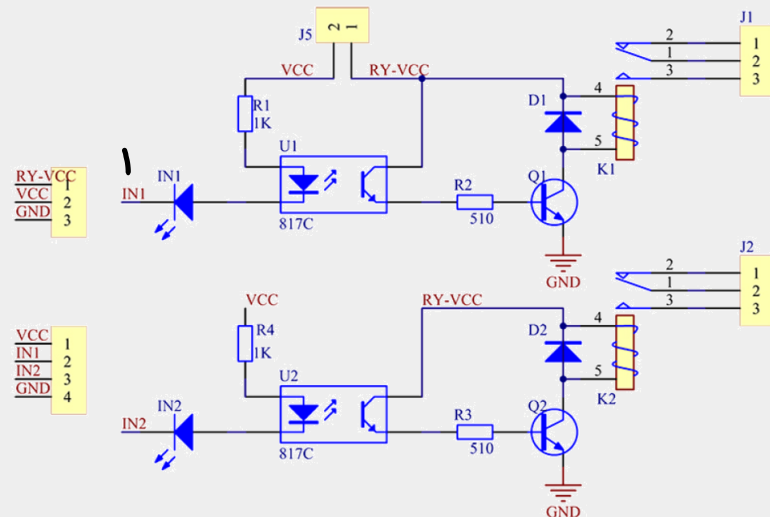
رله الکترومغناطیسی

- کنترل تا جریان 10 آمپر با 100 میلی آمپر
- اتصال به صورت فیزیکی
- قابل استفاده در جریان AC/DC



ماژول رله دو کاناله

Z/N	Relay
1	0
0	1



کنترل رله با سریال

relaySerial.ino

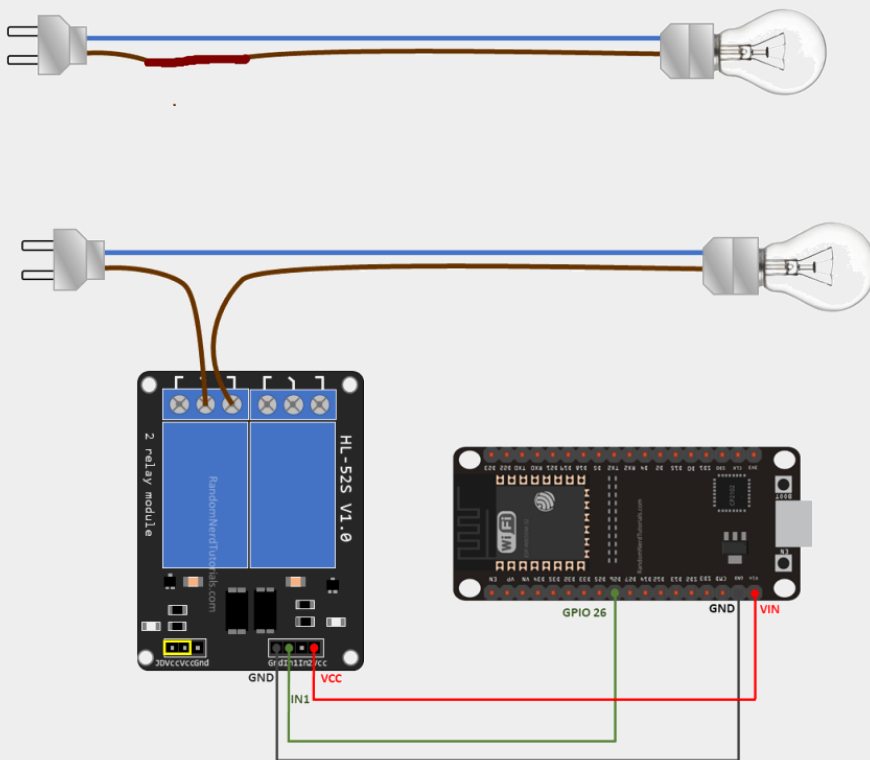
```
1  #define RELAY_PIN 27
2  bool relay = 0;
3  void setup() {
4    Serial.begin(115200);
5    pinMode(RELAY_PIN, OUTPUT);
6    digitalWrite(RELAY_PIN, !relay);
7  }
8  void loop() {
9    if (Serial.available()) {
10     String command = Serial.readStringUntil('\n');
11     processCommand(command);
12   }
13 }
14 void processCommand(String command) {
15   int eqIdx, andIdx;
16   eqIdx = command.indexOf("=");
17   andIdx = command.indexOf("&");
18   String key = command.substring(0, eqIdx);
19   String value = command.substring(eqIdx + 1, andIdx);
20   if(key == "relay"){
21     relay = value.toInt();
22     digitalWrite(RELAY_PIN, !relay);
23   }
24 }
25
```

Serial Monitor

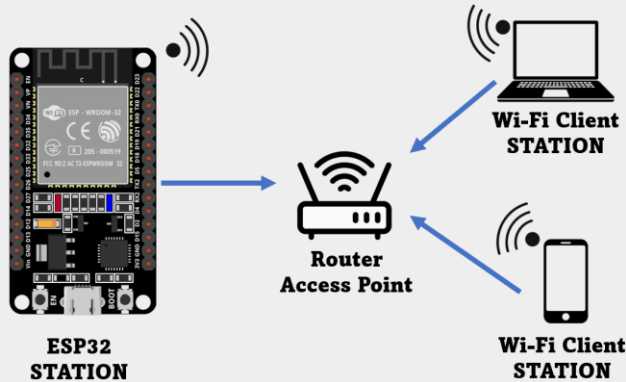
|relay=1

New Line

115200 baud



اتصال وایفای در ESP32



WiFi.mode(WIFI_STA)

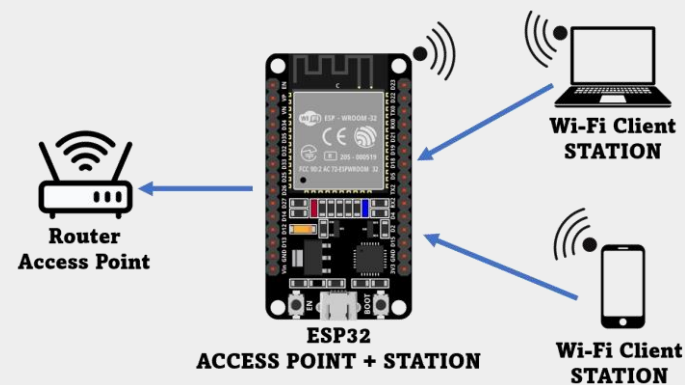
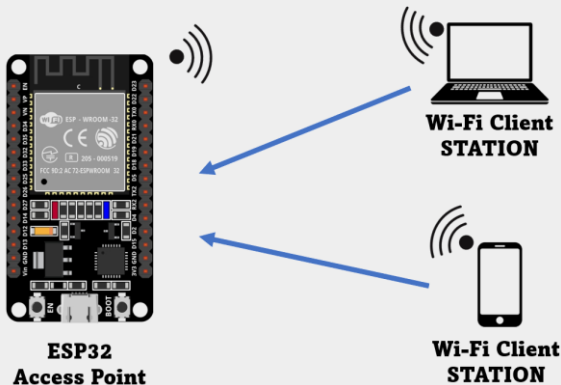
• حالت Station

WiFi.mode(WIFI_AP)

• حالت Access Point

WiFi.mode(WIFI_AP_STA)

• حالت STA+AP





رخداده‌ها در اتصال وایفای

- اتصال به شبکه

- دریافت IP

- قطع اتصال از شبکه

```
6 void wifiConnectedHandler(WiFiEvent_t event, WiFiEventInfo_t info){
7   Serial.println("connected to WiFi access point!");
8 }
9 void wifiIpHandler(WiFiEvent_t event, WiFiEventInfo_t info){
10  Serial.println("Wifi got Ip. IP address: ");
11  Serial.println(WiFi.localIP());
12 }
13 void wifiDisconnectedHandler(WiFiEvent_t event, WiFiEventInfo_t info){
14  Serial.println("Disconnected from WiFi access point");
15 }
16 void setup(){
17   Serial.begin(115200);
18   WiFi.onEvent(wifiConnectedHandler, WiFiEvent_t::ARDUINO_EVENT_WIFI_STA_CONNECTED);
19   WiFi.onEvent(wifiIpHandler, WiFiEvent_t::ARDUINO_EVENT_WIFI_STA_GOT_IP);
20   WiFi.onEvent(wifiDisconnectedHandler, WiFiEvent_t::ARDUINO_EVENT_WIFI_STA_DISCONNECTED);
21   WiFi.begin(ssid, password);
22   Serial.println("Wait for WiFi... ");
23 }
```

عملیات وایفای با سریال

- اسکن شبکه های موجود با دستور scan=true در سریال

```
37 void scanNetworks() {
38     int n = WiFi.scanNetworks();
39     if (n == 0) {
40         Serial.println("no networks found");
41     } else {
42         String result = String(n) + " networks
43         found\n";
44         for (int i = 0; i < n; ++i)
45             result += String(i + 1) + " | " + WiFi.
46             SSID(i) + " | " + String(WiFi.RSSI(i)) +
47             "\n";
48         Serial.println(result);
49     }
50     WiFi.scanDelete();
51 }
```

- اتصال به یک شبکه با دستور
ssid=galax&password=123456789 در سریال

```
51 void connectWifi(String ssid, String password)
52 {
53     staSsid = ssid;
54     staPassword = password;
55     WiFi.disconnect();
56     WiFi.begin(staSsid, staPassword);
57     String resultText = "ssid: " + ssid + " |
58     pass=" + password;
59     Serial.println(resultText);
60     while (WiFi.status() != WL_CONNECTED) {
61         delay(500);
62         Serial.print(".");
63     }
64     resultText = "\nconnected to " + ssid + "\n
65     IP address is: " + WiFi.localIP().toString();
66     Serial.println(resultText);
67 }
```

قابلیت OTA

```
1 #include <WiFi.h>
2 #include <ESPmDNS.h>
3 #include <WiFiUdp.h>
4 #include <ArduinoOTA.h>
5 const char* ssid = "YOUR-SSID";
6 const char* password = "YOUR-PASSWORD";
7 void setup() {
8     WiFi.mode(WIFI_STA);
9     WiFi.begin(ssid, password);
10    ArduinoOTA.begin();
11 }
12 void loop() {
13     ArduinoOTA.handle();
14 }
```

- امکان پروگرام کردن به صورت بی سیم
- مناسب برای ESP32 در حال فعالیت
- تنها با قرار دادن تکه کدهای زیر در برنامه اصلی



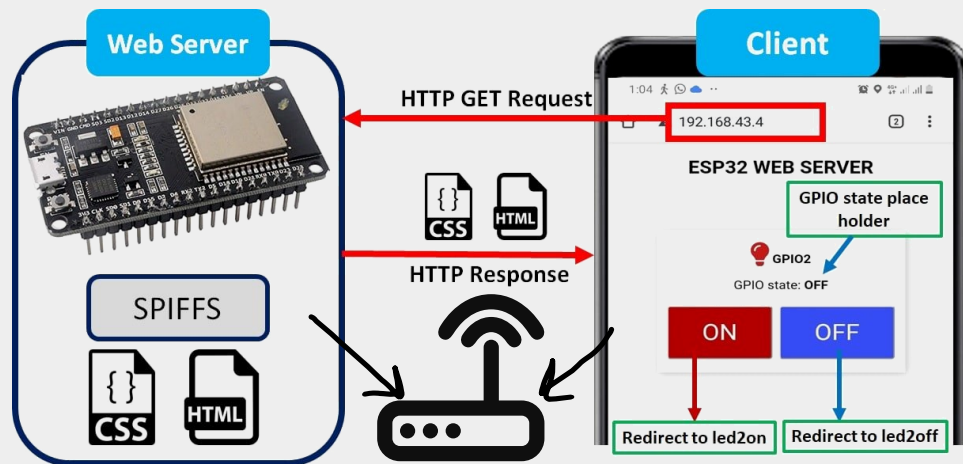
راه اندازی وب سرور در ESP

- ESP32 می‌تواند به عنوان یک سرور عمل کند که به requestهای پروتکل HTTP (که برای مثال از طرف مرورگرها می‌آید) پاسخ دهد

• هدف:

1. نمایش مقادیر سنسورها در مرورگر (دما، رطوبت، روشنایی)

2. کنترل ESP از طریق مرورگر (رله و اتصال وایفای)



صفحه HTML

- زبان نشانه گذاری است و ساختار اصلی هر صفحه وب با استفاده از آن طراحی و برنامه نویسی می شود.
- دکمه ها، تصاویر، لینک ها و حتی جداولی که در وبسایت های مختلف وجود دارد با استفاده از آن پیاده سازی و طراحی می شوند.

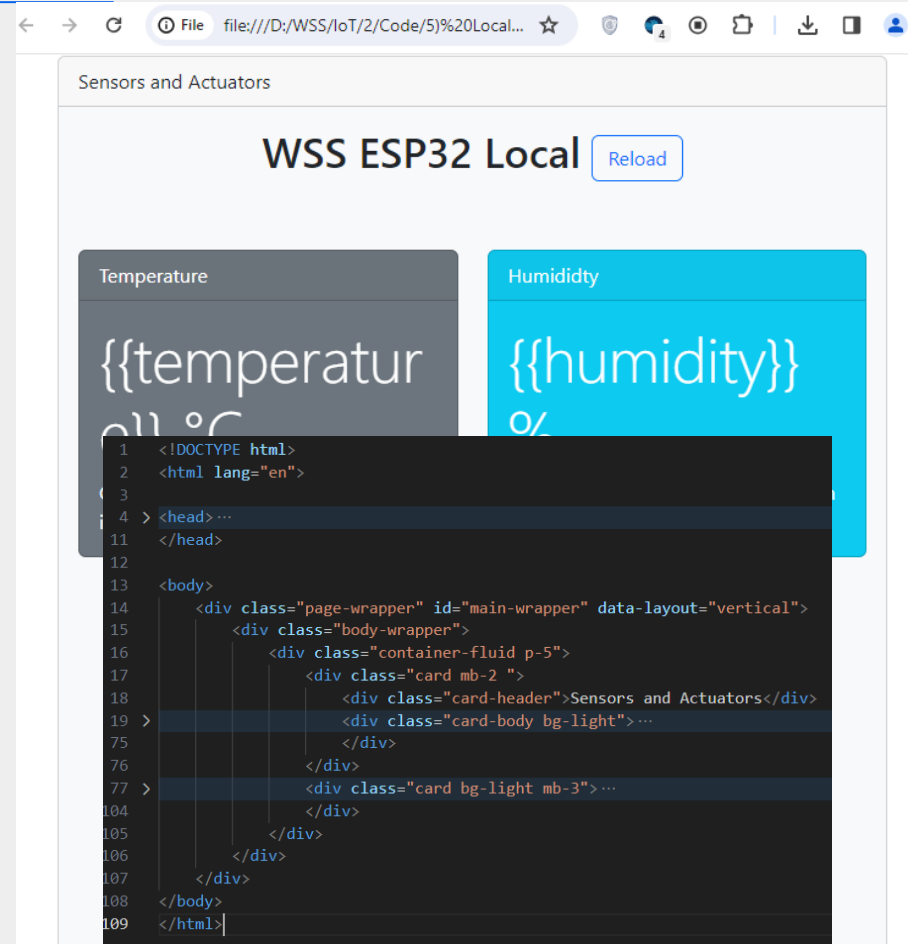
HTML Page Structure

`<!DOCTYPE html>` ← Tells version of HTML
`<html>` ← HTML Root Element

`<head>` ← Used to contain page HTML metadata
 `<title>Page Title</title>` ← Title of HTML page
`</head>`

`<body>` ← Hold content of HTML
 `<h2>Heading Content</h2>` ← HTML heading tag
 `<p>Paragraph Content</p>` ← HTML paragraph tag
`</body>`

`</html>`



نحوه کار سرور ESP

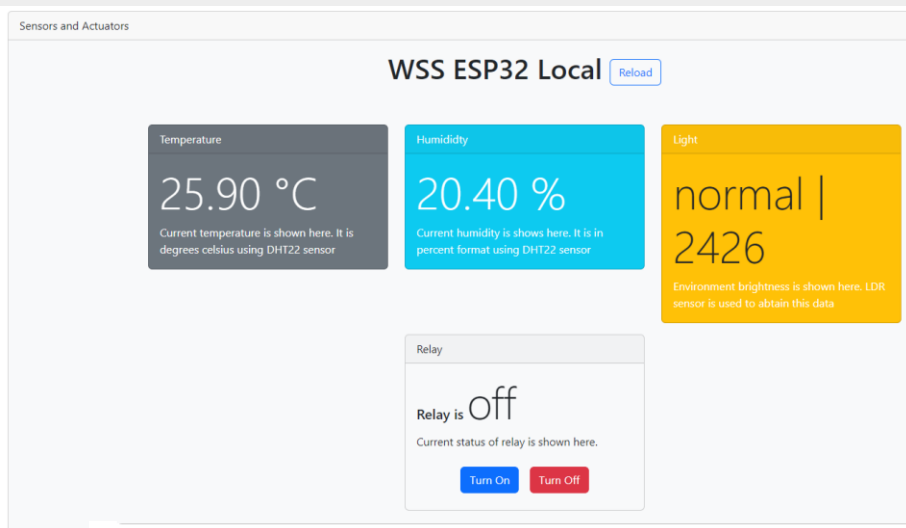
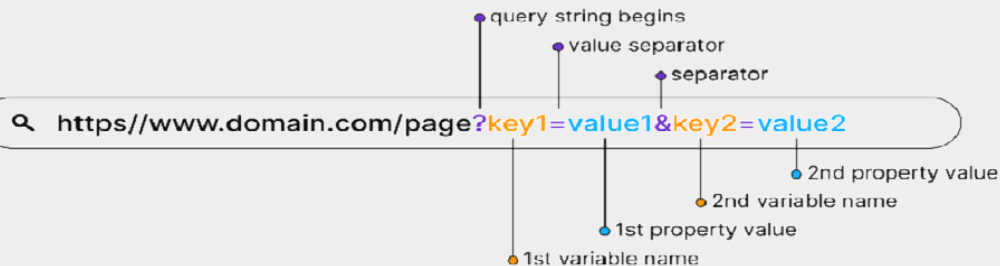
1. درخواست HTTP GET از طریق مرورگر

2. اندازه گیری سنسورها توسط ESP

3. قرار دادن مقادیر اندازه گیری شده در یک رشته HTML

4. فرستادن رشته HTML به عنوان HTTP Response به مرورگر

+ امکان کنترل ESP از طریق پارامترها در درخواست HTTP



WiFi Connection

ESP32 is connected to Sharif-WiFi

WiFi SSID

Enter SSID of network

Password

Enter password of network

Connect

Scan

نمایش صفحه HTML

- آماده‌سازی داده‌ها جهت نمایش به کاربر
- جایگذاری داده‌های آماده شده در قالب HTML ذخیره شده در برنامه
- ارسال صفحه‌ی HTML آماده شده به کاربر

```
50 //Perparing data required by app
51 temperature = dht22.getTemperature();
52 humidity = dht22.getHumidity();
53 light = analogRead(LDR_PIN);
54 String lightStat;
55 if (light < 1000)
56 |   lightStat = "dark";
57 else if (light < 3000)
58 |   lightStat = "normal";
59 else
60 |   lightStat = "bright";
61 lightStat += " | " + String(light);
62 String relayStat = relay ? "on" : "off";
63 String wifiStat = (WiFi.status() != WL_CONNECTED) ? "disconnected" : "connected to " + ssid;
64
65 //Placing appropriate data in HTML template in html.h
66 String html = HTML_CONTENT;
67 html.replace("{}temperature{}", String(temperature));
68 html.replace("{}humidity{}", String(humidity));
69 html.replace("{}light{}", lightStat);
70 html.replace("{}relay{}", relayStat);
71 html.replace("{}wifiStat{}", wifiStat);
72 html.replace("{}wifiNetworks{}", wifiNetworks);
```

```
74 //Sending rendred html file to user
75 server.send(200, "text/html", html);
```

مدیریت پارامترها در درخواست HTTP

- اگر کاربر همراه با درخواست HTTP پارامتر هم ارسال کرد به آن واکنش مناسب داده می شود.

- URL?relay=false

روشن/خاموش کردن رله

- URL?scan=true

اسکن شبکه های وایفای اطراف و نمایش در صفحه

- URL?ssid=xx&password=yy

اتصال به یک شبکه ی وایفای

```
23 void handleRoot() {
24     String wifiNetworks;
25     //Check whether any parameter is available in HTTP request
26     if (server.args() > 0) {
27         //switching relay
28         if (server.argName(0) == "relay") {
29             relay = server.arg(0).toInt();
30             digitalWrite(RELAY_PIN, !relay);
31         }
32         //scanning available wifi networks
33         else if (server.argName(0) == "scan") {
34             int n = WiFi.scanNetworks();
35             wifiNetworks = "<b>List of available networks</b>";
36             for (int i = 0; i < n; ++i)
37                 wifiNetworks += "<li class='list-group-item'>" + WiFi.SSID(i) + "</li>";
38             WiFi.scanDelete();
39         }
40         //connecting to a wifi network
41         else if (server.argName(0) == "ssid") {
42             ssid = server.arg(0);
43             password = server.arg(1);
44             server.send(200, "text/html", "<h1>connecting to " + ssid + " ... </h1>");
45             connectWifi();
46             return;
47         }
48     }
49 }
```


ممنون از
توجه شما