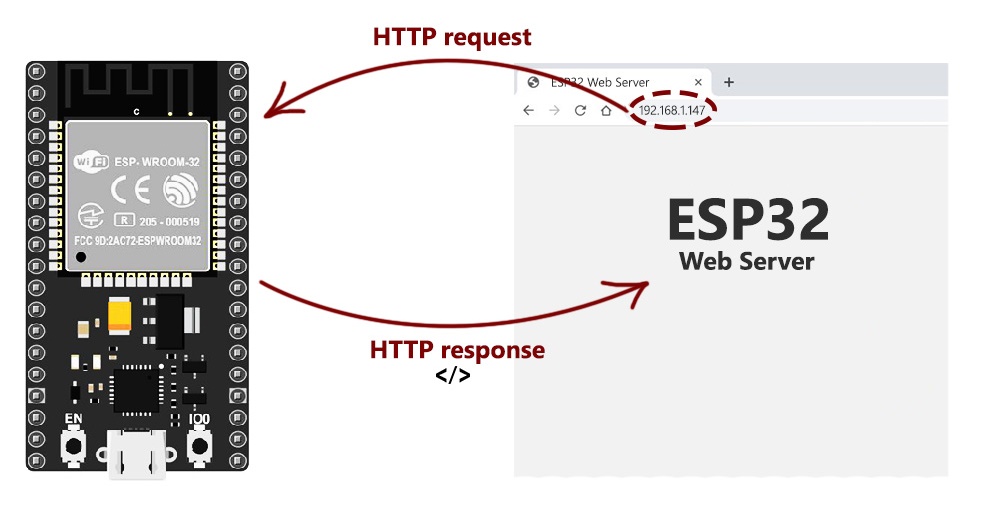
بسم الله الرحمن الرحيم

**ESP32 Web Server**



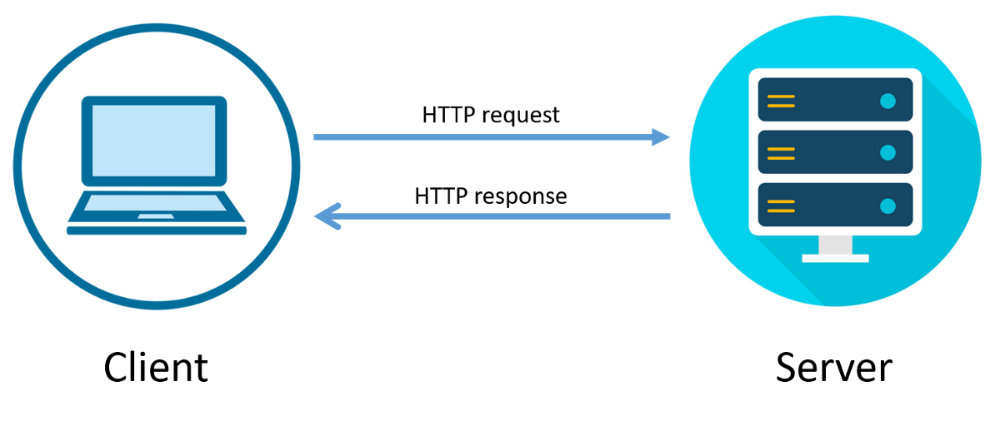
صورة تحتوي على نص, إلكترونيات

تم إنشاء الوصف تلقائياً

The ESP32 is a very cost-effective Wi-Fi module that can be programmed - with little extra effort - to build a standalone web server, in this report, a mechanism for working as a web server will be explained.

**What is a Web server and how it works?**

Web server is a place which stores, processes, and delivers web pages to Web clients. Web client is nothing but a web browser on our laptops and smartphones. The communication between client and server takes place using a special protocol called Hypertext Transfer Protocol (HTTP).



HTTP Web Server Client Illustration

In this protocol, a client initiates communication by making a request for a specific web page using HTTP and the server responds with the content of that web page or an error message if unable to do so (like famous 404 Error).

Pages delivered by a server are mostly HTML documents.

ESP32 Operating Modes

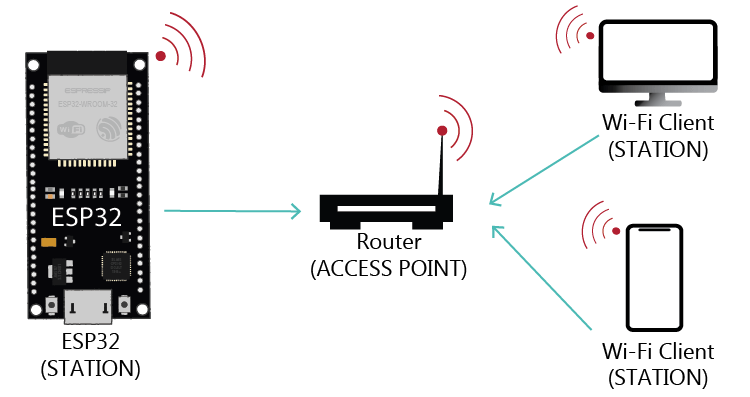
One of the greatest features ESP32 provides is that it cannot only connect to an existing Wi-Fi network and act as a Web Server, but it can also set up a network of its own, allowing other devices to connect directly to it and access web pages. This is possible because ESP32 can operate in three different modes:

* Station mode
* Soft Access
* Point mode
* and both at the same time.

This provides possibility of building mesh networks.

Station (STA) Mode

The ESP32 that connects to an existing Wi-Fi network (one created by your wireless router) is called **Station** (STA).



In STA mode ESP32 gets IP from wireless router to which it is connected.

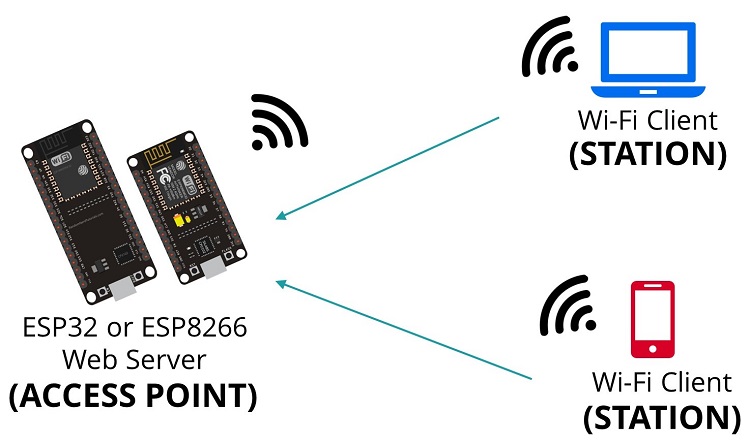
With this IP address, it can set up a web server and **deliver web pages to all connected devices under existing Wi-Fi network**.

Soft Access Point (AP) Mode

The ESP32 that creates its own Wi-Fi network and acts as a hub (Just like Wi-Fi router) for one or more stations is called **Access Point** (AP).

Unlike Wi-Fi router, it does not have interface to a wired network. So, such mode of operation is called **Soft Access Point** (soft-AP).

Also, the maximum number of stations that can connect to it is limited to five.



In AP mode ESP32 creates a new Wi-Fi network and sets SSID (Name of the network) and IP address to it.

With this IP address, it can **deliver web pages to all connected devices under its own network**.

Concept Behind Controlling Things from ESP32 Web Server

So, you might be thinking, “How am I going to control things from a web server that merely processes and delivers web pages?”

Well, then you need to understand what’s going on behind the scenes.

When you type a URL in a web browser and hit ENTER, the browser sends a HTTP request (a.k.a. GET request) to a web server.

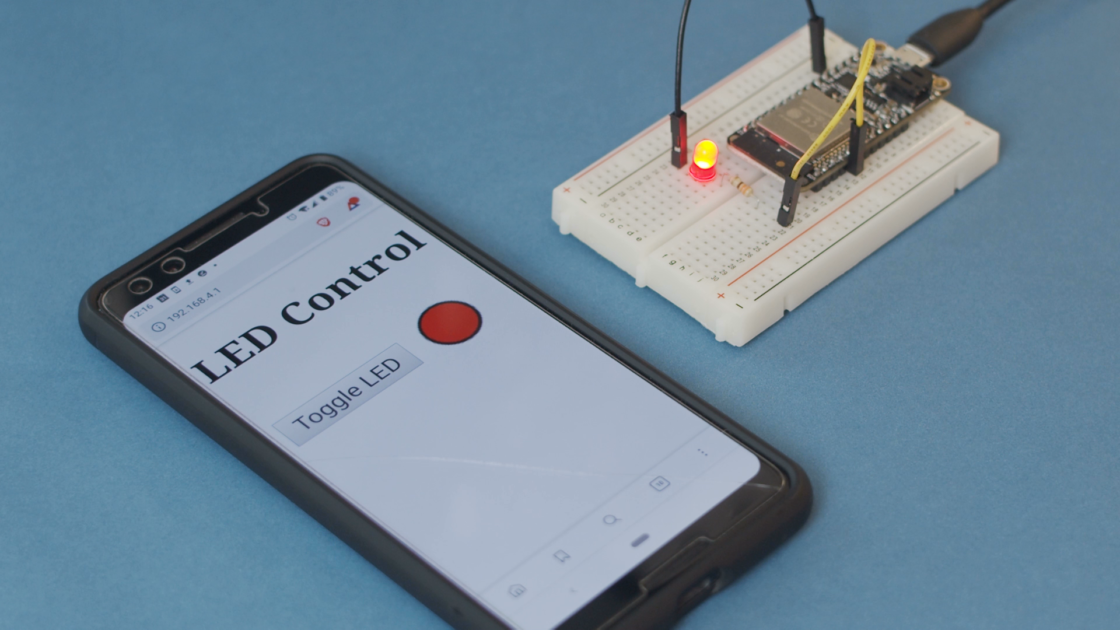
It’s a job of web server to handle this request by doing something.

You might have figured it out by now that we are going to control things by accessing a specific URL.

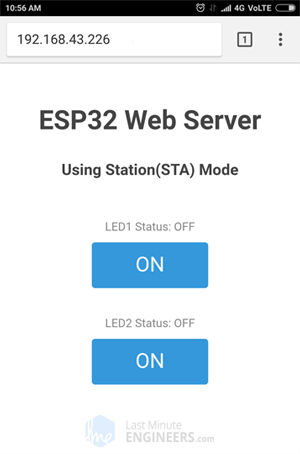
For example, suppose we entered a URL http://192.168.1.1/**ledon** in a browser.

The browser then sends a HTTP request to ESP32 to handle this request. When ESP32 reads this request, it knows that user wants to turn the LED ON.

So, it turns the LED ON and sends a dynamic webpage to a browser showing **LED status: ON**. As easy as Pie!



Accessing the Web Server in STA mode

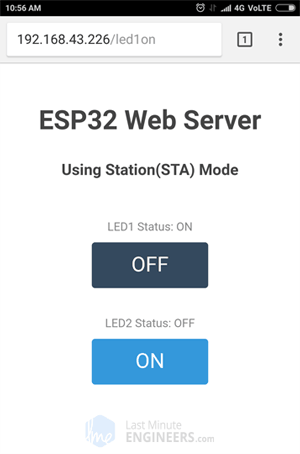


If everything is OK, it will output the dynamic IP address obtained from your router and show **HTTP server started** message.

Next, load up a browser and point it to the IP address shown on the serial monitor.

The ESP32 should serve up a web page showing current status of LEDs and two buttons to control them.

If take a look at the serial monitor at the same time, you can see status of ESP32’s GPIO pins.

Now, click the button to turn LED1 ON while keeping an eye on the URL.

Once you click the button, the ESP32 receives a request for **/led1on** URL.

It then turns the LED1 ON and serves a web page with status of LED updated.

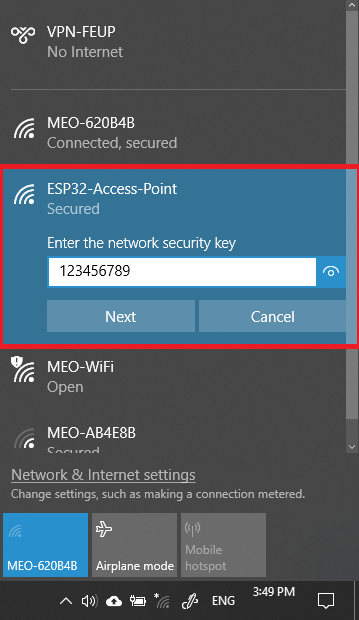
ESP32 as HTTP Server using Wi-Fi Access Point (AP) mode

As the heading suggests, this example demonstrates how to turn the ESP32 into an access point (AP) and serve up web pages to any connected client.

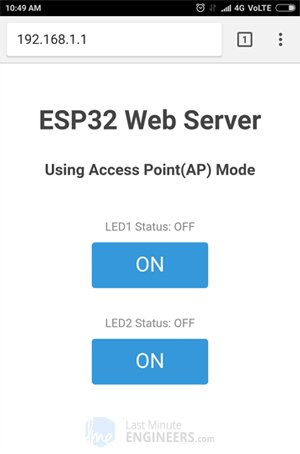
**Accessing the Web Server in AP mode**

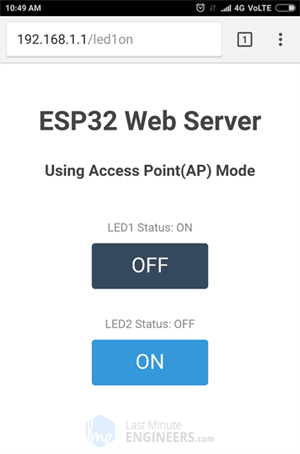
Now should find any device that you can connect to a Wi-Fi network – phone, laptop, etc.

And look for a network called **ESP32-Access-Point**.Join the network with password **123456789**.

After connecting to your ESP32 AP network, load up a browser and point it to 192.168.1.1 The ESP32 should serve up a web page showing current status of LEDs and two buttons to control them.

If take a look at the serial monitor at the same time, you can see status of ESP32’s GPIO pins.





Now, click the button to turn LED1 ON while keeping an eye on the URL.

Once you click the button, the ESP32 receives a request for **/led1on** URL.

It then turns the LED1 ON and serves a web page with status of LED updated.

It also prints the status of GPIO pin on the serial monitor.

The only difference between AP & STA mode is one creates the network and other joins the existing network.

So, rest of the code for handling HTTP requests and serving web page in STA mode is same as that of AP mode explained above. This includes:

* Declaring ESP32’s GPIO pins to which LEDs are connected
* Defining multiple server.on() methods to handle incoming HTTP requests
* Defining server.onNotFound() method to handle HTTP 404 error
* Creating custom functions that are executed when specific URL is hit
* Creating HTML page
* Styling the web page
* Creating buttons and displaying their status.

