ESP32 Web Server using SPIFFS (SPI Flash File System)

In this tutorial we'll show you how to build a web server that serves HTML and CSS files stored on the ESP32 filesystem. Instead of having to write the HTML and CSS text into the Arduino sketch, we'll create separated HTML and CSS files.

For demonstration purposes, the web server we'll build controls an ESP32 output, but it can be easily adapted for other purposes like displaying sensor readings.

Recommended reading: ESP8266 Web Server using SPIFFS

ESP32 Filesystem Uploader Plugin

To follow this tutorial you should have the ESP32 Filesystem Uploader plugin installed in your Arduino IDE. If you haven't, follow the next tutorial to install it first:

- Install ESP32 Filesystem Uploader on Arduino IDE

Note: make sure you have the latest Arduino IDE installed, as well as the ESP32 add-on for the Arduino IDE. If you don't, follow one of the next tutorials to install it:

- **Windows** instructions – Installing the ESP32 Board in Arduino IDE
- **Mac and Linux** instructions – Installing the ESP32 Board in Arduino IDE

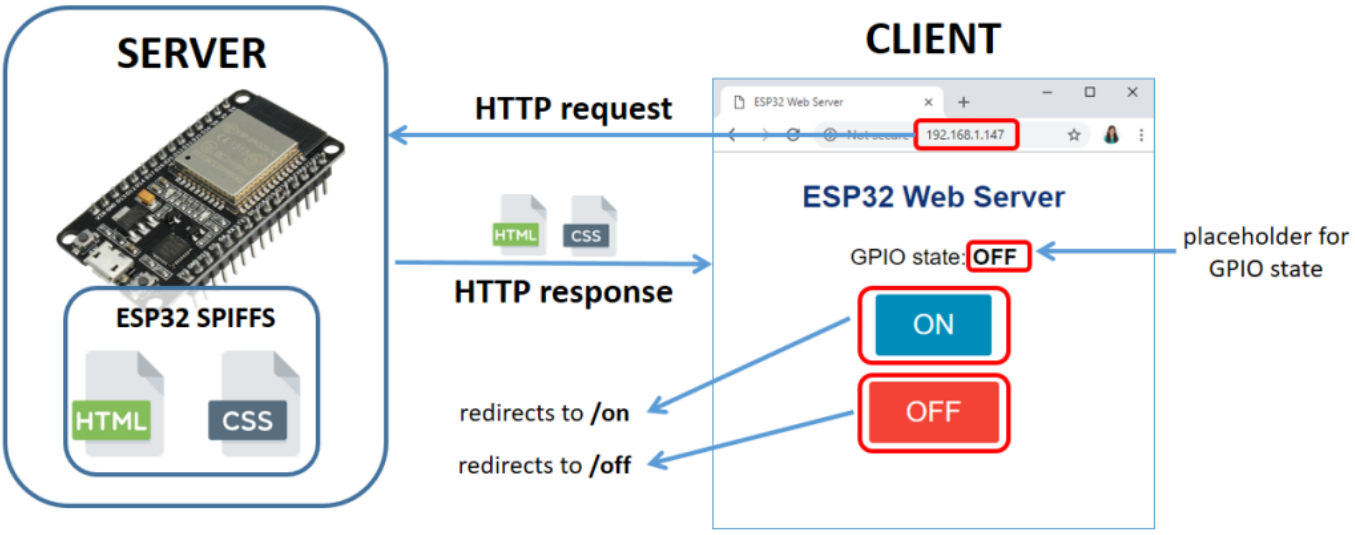
Project Overview

Before going straight to the project, it's important to outline what our web server will do, so that it is easier to understand.

- The web server you'll build controls an LED connected to the ESP32 GPIO 2. This is the ESP32 on-board LED. You can control any other GPIO;
- The web server page shows two buttons: ON and OFF – to turn GPIO 2 on and off;

- The web server page also shows the current GPIO state.

The following figure shows a simplified diagram to demonstrate how everything works.



- The ESP32 runs a web server code based on the ESPAsyncWebServer library;
- The HTML and CSS files are stored on the ESP32 SPIFFS (Serial Peripheral Interface Flash File System);
- When you make a request on a specific URL using your browser, the ESP32 responds with the requested files;
- When you click the ON button, you are redirected to the root URL followed by **/on** and the LED is turned on;
- When you click the OFF button, you are redirected to the root URL followed by **/off** and the LED is turned off;
- On the web page, there is a placeholder for the GPIO state. The placeholder for the GPIO state is written directly in the HTML file between % signs, for example **%STATE%**.

Installing Libraries

In most of our projects we've created the HTML and CSS files for the web server as a String directly on the Arduino sketch. With SPIFFS, you can write the HTML and CSS in separated files and save them on the ESP32 filesystem.

One of the easiest ways to build a web server using files from the filesystem is by using the ESPAsyncWebServer library. The ESPAsyncWebServer library is well documented on its GitHub page. For more information about that library, check the following link:

- https://github.com/me-no-dev/ESPAsyncWebServer

Installing the ESPAsyncWebServer library

Follow the next steps to install the ESPAsyncWebServer library:

1. Click here to download the ESPAsyncWebServer library. You should have a .zip folder in your Downloads folder
2. Unzip the .zip folder and you should get ESPAsyncWebServer-master folder
3. Rename your folder from ~~ESPAsyncWebServer-master~~ to ESPAsyncWebServer
4. Move the ESPAsyncWebServer folder to your Arduino IDE installation libraries folder

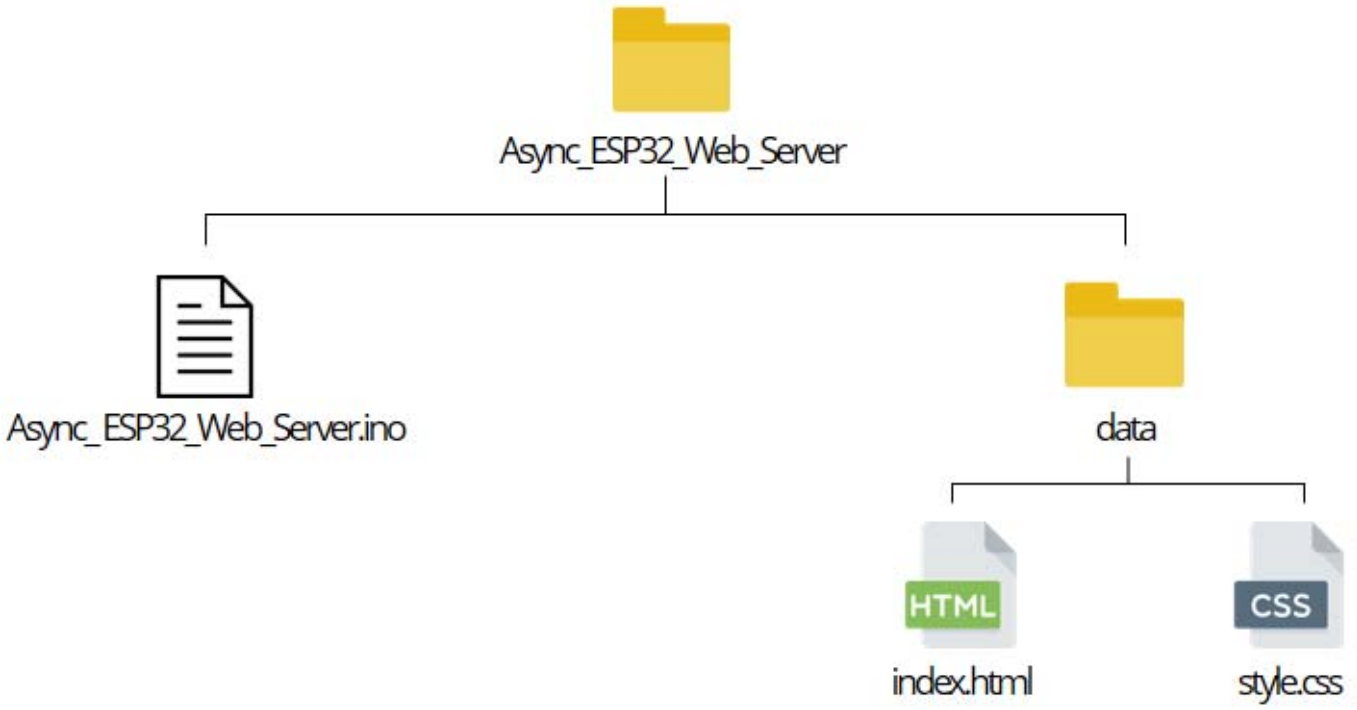
Installing the Async TCP Library for ESP32

The ESPAsyncWebServer library requires the AsyncTCP library to work. Follow the next steps to install that library:

1. Click here to download the AsyncTCP library. You should have a .zip folder in your Downloads folder
2. Unzip the .zip folder and you should get AsyncTCP-master folder
3. Rename your folder from ~~AsyncTCP-master~~ to AsyncTCP
4. Move the AsyncTCPfolder to your Arduino IDE installation libraries folder
5. Finally, re-open your Arduino IDE

Organizing your Files

To build the web server you need three different files. The Arduino sketch, the HTML file and the CSS file. The HTML and CSS files should be saved inside a folder called **data** inside the Arduino sketch folder, as shown below:



Creating the HTML File

The HTML for this project is very simple. We just need to create a heading for the web page, a paragraph to display the GPIO state and two buttons.

Create an *index.html* file with the following content or download all the project files here:

```
<!DOCTYPE html>
<html>
<head>
  <title>ESP32 Web Server</title>
  <meta name="viewport" content="width=device-width, initial-scale=
  <link rel="icon" href="data:,">
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>ESP32 Web Server</h1>
  <p>GPIO state: <strong> %STATE%</strong></p>
  <p><a href="/on"><button class="button">ON</button></a></p>
  <p><a href="/off"><button class="button button2">OFF</button></a>
```

```
</body>
</html>
```

View raw code

Because we're using CSS and HTML in different files, we need to reference the CSS file on the HTML text. The following line should be added between the <head> </head> tags:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

The <link> tag tells the HTML file that you're using an external style sheet to format how the page looks. The **rel** attribute specifies the nature of the external file, in this case that it is a **stylesheet**—the CSS file—that will be used to alter the appearance of the page.

The **type** attribute is set to **"text/css"** to indicate that you're using a CSS file for the styles. The **href** attribute indicates the file location; since both the CSS and HTML files will be in the same folder, you just need to reference the filename: **style.css**.

In the following line, we write the first heading of our web page. In this case we have "ESP32 Web Server". You can change the heading to any text you want:

```
<h1>ESP32 Web Server</h1>
```

Then, we add a paragraph with the text "GPIO state: " followed by the GPIO state. Because the GPIO state changes accordingly to the state of the GPIO, we can add a placeholder that will then be replaced for whatever value we set on the Arduino sketch.

To add placeholder we use **%** signs. To create a placeholder for the state, we can use `%STATE%`, for example.

```
  <p>GPIO state: <strong>%STATE%</strong></p>
```

Attributing a value to the `STATE` placeholder is done in the Arduino sketch.

Then, we create an ON and an OFF buttons. When you click the on button, we redirect the web page to to root followed by `/on` url. When you click the off button you are redirected to the `/off` url.

```
  <p><a href="/on"><button class="button">ON</button></a></p>
  <p><a href="/off"><button class="button button2">OFF</button></a></
```

Creating the CSS file

Create the *style.css file* with the following content or download all the project files here:

```
  html {
    font-family: Helvetica;
    display: inline-block;
    margin: 0px auto;
    text-align: center;
  }
  h1{
    color: #0F3376;
    padding: 2vh;
  }
  p{
    font-size: 1.5rem;
  }
  .button {
    display: inline-block;
    background-color: #008CBA;
```

```
    border: none;
    border-radius: 4px;
    color: white;
    padding: 16px 40px;
    text-decoration: none;
    font-size: 30px;
    margin: 2px;
    cursor: pointer;
}
.button2 {
    background-color: #f44336;
```

View raw code

This is just a basic CSS file to set the font size, style and color of the buttons and align the page. We won't explain how CSS works. A good place to learn about CSS is the W3Schools website.

Arduino Sketch

Copy the following code to the Arduino IDE or download all the project files here. Then, you need to type your network credentials (SSID and password) to make it work.

```
/*********
  Rui Santos
  Complete project details at https://randomnerdtutorials.com
*********/

// Import required libraries
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include "SPIFFS.h"

// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
```

```
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

// Set LED GPIO
const int ledPin = 2;
// Stores LED state
String ledState;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

// Replaces placeholder with LED state value
String processor(const String& var){
  Serial.println(var);
  if(var == "STATE"){
```

View raw code

How the Code Works

First, include the necessary libraries:

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include "SPIFFS.h"
```

You need to type your network credentials in the following variables:

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

Next, create a variable that refers to GPIO 2 called `ledPin`, and a String variable to hold the led state: `ledState`.

```
const int ledPin = 2;
String ledState;
```

Create an `AsynWebServer` object called `server` that is listening on port 80.

```
AsyncWebServer server(80);
```

processor()

The `processor()` function is what will attribute a value to the placeholder we've created on the HTML file. It accepts as argument the placeholder and should return a String that will replace the placeholder. The `processor()` function should have the following structure:

```
String processor(const String& var){
  Serial.println(var);
  if(var == "STATE"){
    if(digitalRead(ledPin)){
      ledState = "ON";
    }
    else{
      ledState = "OFF";
    }
    Serial.print(ledState);
    return ledState;
  }
  return String();
}
```

This function first checks if the placeholder is the `STATE` we've created on the HTML file.

```
if(var == "STATE"){
```

If it is, then, accordingly to the LED state, we set the `ledState` variable to either ON or OFF.

```
if(digitalRead(ledPin)){
  ledState = "ON";
}
else{
  ledState = "OFF";
}
```

Finally, we return the `ledState` variable. This replaces the placeholder with the `ledState` string value.

```
return ledState;
```

setup()

In the `setup()`, start by initializing the Serial Monitor and setting the GPIO as an output.

```
Serial.begin(115200);
pinMode(ledPin, OUTPUT);
```

Initialize SPIFFS:

```
if(!SPIFFS.begin(true)){
  Serial.println("An Error has occurred while mounting SPIFFS");
```

```
    return;
  }
```

Wi-Fi connection

Connect to Wi-Fi and print the ESP32 IP address:

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Connecting to WiFi..");
}
Serial.println(WiFi.localIP());
```

Async Web Server

The ESPAsyncWebServer library allows us to configure the routes where the server will be listening for incoming HTTP requests and execute functions when a request is received on that route. For that, use the `on()` method on the server object as follows:

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send(SPIFFS, "/index.html", String(), false, processor);
});
```

When the server receives a request on the root "*/*" URL, it will send the *index.html* file to the client. The last argument of the `send()` function is the processor, so that we are able to replace the placeholder for the value we want – in this case the `ledState`.

Because we've referenced the CSS file on the HTML file, the client will make a request for the CSS file. When that happens, the CSS file is sent to the client:

```
server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest *request
  request->send(SPIFFS, "/style.css","text/css");
});
```

Finally, you need to define what happens on the `/on` and `/off` routes. When a request is made on those routes, the LED is either turned on or off, and the ESP32 serves the HTML file.

```
server.on("/on", HTTP_GET, [](AsyncWebServerRequest *request){
  digitalWrite(ledPin, HIGH);
  request->send(SPIFFS, "/index.html", String(),false, processor);
});
server.on("/off", HTTP_GET, [](AsyncWebServerRequest *request){
  digitalWrite(ledPin, LOW);
  request->send(SPIFFS, "/index.html", String(),false, processor);
});
```

In the end, we use the `begin()` method on the `server` object, so that the server starts listening for incoming clients.

```
server.begin();
```

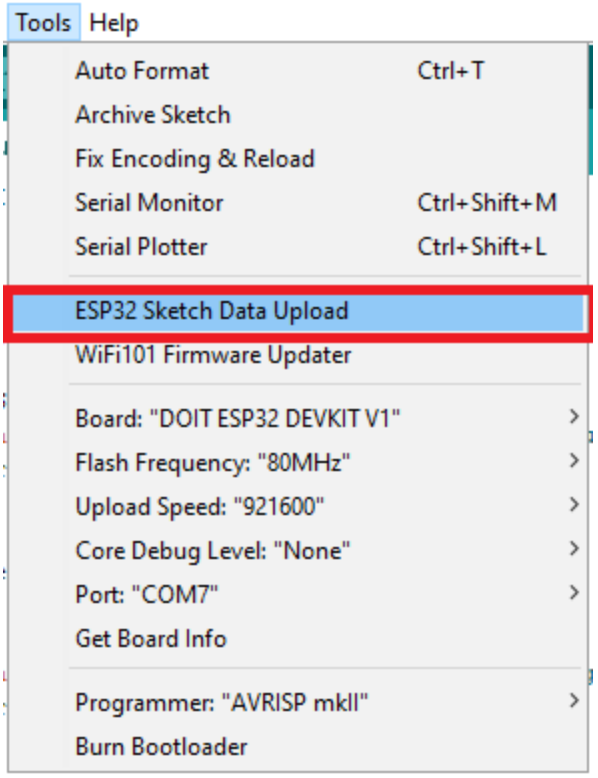
Because this is an asynchronous web server, you can define all the requests in the `setup()`. Then, you can add other code to the `loop()` while the server is listening for incoming clients.

Uploading Code and Files

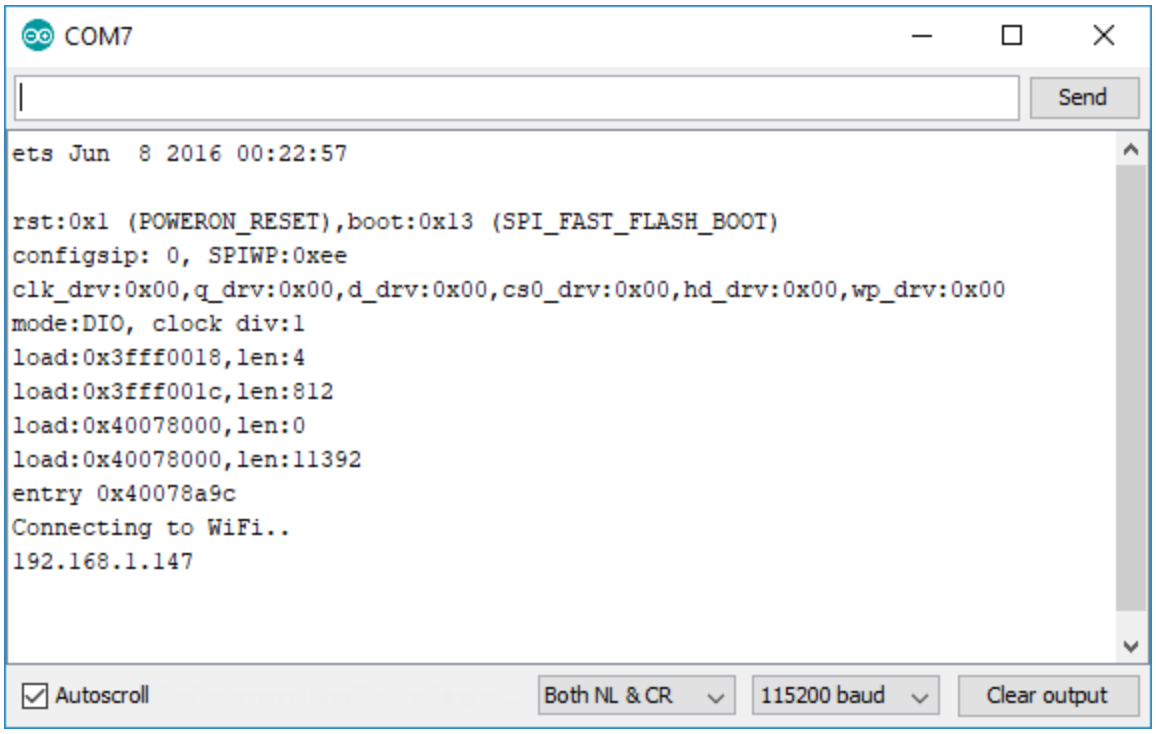
Save the code as *Async_ESP32_Web_Server* or download all the project files here. Go to **Sketch** > **Show Sketch Folder**, and create a folder called **data**. Inside that folder you should save the HTML and CSS files.

Then, upload the code to your ESP32 board. Make sure you have the right board and COM port selected. Also, make sure you've added your networks credentials to the code.

After uploading the code, you need to upload the files. Go to **Tools** > **ESP32 Data Sketch Upload** and wait for the files to be uploaded.



When everything is successfully uploaded, open the Serial Monitor at a baud rate of 115200. Press the ESP32 "**ENABLE**" button, and it should print the ESP32 IP address.



Demonstration

Open your browser and type the ESP32 IP address. Press the ON and OFF buttons to control the ESP32 on-board LED. Also, check that the GPIO state is being updated correctly.

Wrapping Up

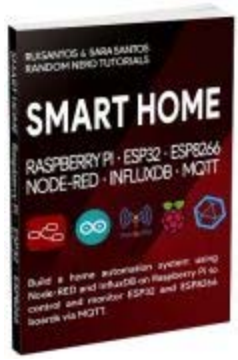
Using SPI Flash File System (SPIFFS) is specially useful to store HTML and CSS files to serve to a client – instead of having to write all the code inside the Arduino sketch.

The ESPAsyncWebServer library allows you to build a web server by running a specific function in response to a specific request. You can also add placeholders to the HTML file that can be replaced with variables – like sensor readings, or GPIO states, for example.

If you liked this project, you may also like:

- ESP32 Web Server with BME280 – Mini Weather Station
- ESP32 Web Server – Arduino IDE
- Getting Started with MicroPython on ESP32 and ESP8266

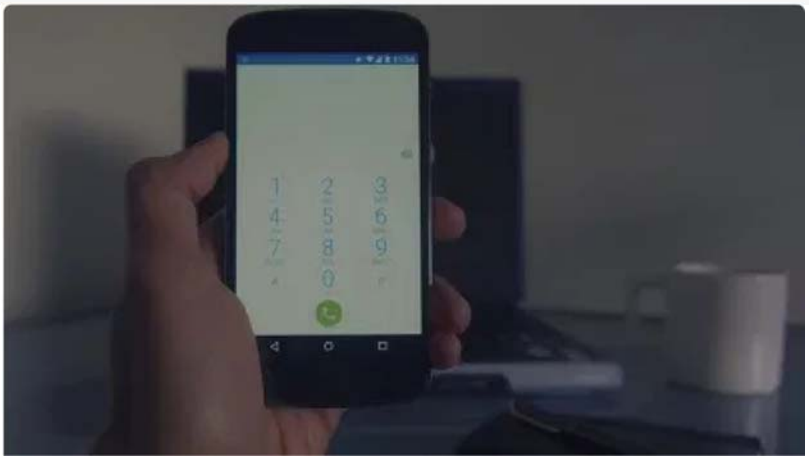
*This is an excerpt from our course: **Learn ESP32 with Arduino IDE**. If you like ESP32 and you want to learn more, we recommend enrolling in Learn ESP32 with Arduino IDE course.*



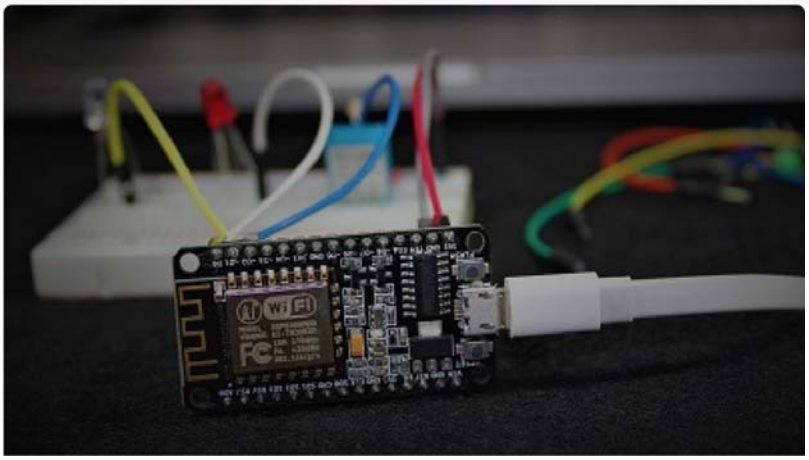
SMART HOME with Raspberry Pi, ESP32, ESP8266 [eBook]

Learn how to build a home automation system and we'll cover the following main subjects: Node-RED, Node-RED Dashboard, Raspberry Pi, ESP32, ESP8266, MQTT, and InfluxDB database **DOWNLOAD »**

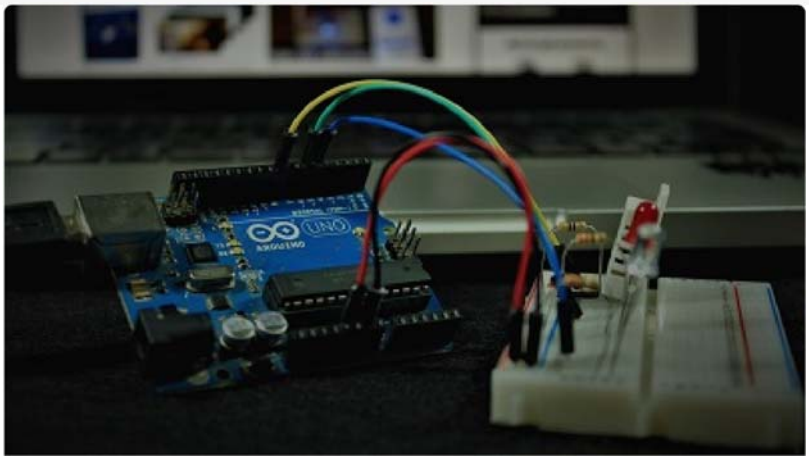
Recommended Resources



Build a Home Automation System from Scratch » With Raspberry Pi, ESP8266, Arduino, and Node-RED.



Home Automation using ESP8266 eBook and video course » Build IoT and home automation projects.



Arduino Step-by-Step Projects » Build 25 Arduino projects with our course, even with no prior experience!

What to Read Next…

Visualize Your Sensor Readings from Anywhere in the World (ESP32/ESP8266 + MySQL + PHP)

ESP32: Getting Started with Deta Base (Unlimited and Free Database for Developers)

Get Epoch/Unix Time with the ESP32 (Arduino)

SUBSCRIBE

144 thoughts on “ESP32 Web Server using SPIFFS (SPI Flash File System)”

Evan

October 19, 2018 at 3:47 pm

Can you configure the device to be in Station mode where it creates its own hotspot so there is no internet / router needed?

Reply

Sara Santos

October 19, 2018 at 4:08 pm

Hi Evan.
I think you mean setting the ESP32 as an access point, where there is no need for router.
Yes, you can set the ESP32 as an acess point. Take a look at the following tutorial:
– How to Set an ESP32 Access Point (AP) for Web Server
I hope this helps.
Regards,
Sara 🙂

Reply

Swarup

December 25, 2021 at 5:10 pm

could you demonstrate setting it as Soft AP and access local files to display this same website as in the tutorial.

Reply

Sara Santos

December 27, 2021 at 2:56 pm

Hi.
You just have to add the lines to set the access point:
https://randomnerdtutorials.com/esp32-access-point-ap-web-server/
Regards,
Sara

Reply

Jagadish M

May 4, 2022 at 10:56 am

how can we update bin file that has stored in ufs we need to update in esp32. if u know please add that coding lines.

Reply

Sastry Dasigi

October 19, 2018 at 4:05 pm

Thanks for an amazing tutorial, Rui!

Reply

Imre

October 19, 2018 at 6:01 pm

Can HTML file contain JavaScript code?

Reply

Sara Santos

October 21, 2018 at 1:29 pm

Yes.

Reply

micky

May 27, 2020 at 6:11 am

Can I upload Javascript file?

Reply

Sara Santos

May 27, 2020 at 3:30 pm

Yes.

Reply

Paul

October 20, 2018 at 3:51 am

Hi, incredible good… thanks!
Can you store the CSS and HTML files into an SD card, so that you can actualize it every time you want without requiring to update your sketch?

Reply

Adam

October 20, 2018 at 9:56 am

You don’t need to re-compile whenever you want to upload the HTML files. They are stored in the ESP32 file system: SPIFFS. Totally separate memory from code – that was the point of this article and not to store the HTML in

the code, but in the dedicated local filing system. Saves you needing any SD, its built into the ESP32 own memory map.

Reply

Sara Santos

October 21, 2018 at 3:32 pm

Hi Adam.
That's right! Thank you for answering the question.
Regards,
Sara 🙂

Reply

Sara Santos

October 21, 2018 at 3:30 pm

Hi Paul.
Yes, alternatively you can store your files in an SD card.
You need to use the SD.h library to manipulate the files on the SD card.
Regards,
Sara 🙂

Reply

ENRIQUE GARCIZA

March 27, 2023 at 8:39 pm

Hi, very nice. Do you have some example in how to do it with SD card.
I try using this example only changing:
request->send(SPIFF,"/index.html", "text/html", processor);
by
request->send(SD,"/index.html", "text/html", processor);
Then I try
request->send(SPIFFS, "/index.html", String(), false, processor);
by
request->send(SD, "/index.html", String(), false, processor);

With no sucess.
Please
Advice

Reply

Rho

October 20, 2018 at 1:39 pm

Hey Team, this looks awesome! Will have a go when I get back home.
I'll try the LED tutorial first, then implement some crazier stuff 😉
(Probably a LED strip/matrix)

Thanks!

Reply

Sara Santos

October 21, 2018 at 3:49 pm

Hi Rho.
That's great!
Then, tell us how it went 🙂
Regards,
Sara 🙂

Reply

Leo
October 20, 2018 at 10:14 pm

Excelente como todos tus tutoriales

Reply

Sara Santos
October 21, 2018 at 3:49 pm

Thanks Leo.
Regards,
Sara 🙂

Reply

Denis Brion
October 23, 2018 at 5:32 pm

Well, I suppose SD cards can hold up to 32 G data, SPIFFS maybe 1 M
Then SPIFFS is convenient with unfrequent changes (suppose html page countains data: if they are updated once a second -temperature; ca 10 chars, with a time stamp in a huamn friendly format -, it needs ca one day to fill SPIFFS …. and some decades to fill SD card).
I hope this calculation is not absurd (and I am aware I missed number of repeated writes on SD cards, and maybe on ESP flash).

Reply

Sara Santos
October 24, 2018 at 11:13 am

Hi Denis, you are right.
As stated in this tutorial, using SPIFFS is useful to
– Create configuration files with settings;
– Save data permanently;
– Create files to save small amounts of data instead of using a microSD card;
– Save HTML and CSS files to build a web server;
It shouldn't be used to create datalogging projects that required large amounts of data or multiple updates for second.
Thank you for your comment,
Regards,
Sara 🙂

Reply

João Lopes
November 9, 2018 at 3:56 pm

Excellent! Very good tutorial. thanks

Reply

mohan

November 14, 2018 at 10:08 am

how to send text file in sd card to a web server in esp wroom32

Reply

Sara Santos

November 14, 2018 at 10:40 am

Hi.
We don’t have a specific tutorial about that subject.
But we have a project in which we load the html file from an SD card to a web server using ESP32. You can take a look at the code in the following project:
https://randomnerdtutorials.com/build-an-all-in-one-esp32-weather-station-shield/
I hope it helps.
Regards,
Sara 🙂

Reply

Alan Cross

November 16, 2018 at 8:11 pm

Hi guys, I have the webserver running with SPIFFS. I have tried adding a slider to get a value back into ESP without much success. Without using Ajax or Java, is there an easy way of getting a slider value fro the webpage into ESP32?

Reply

Sara Santos

November 19, 2018 at 12:35 pm

Hi Alan.
I don’t have a specific example using just a slider.
We have an example that uses a slider to control a servo motor.
You can take a look at the code and try to figure out how it works:
https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/
Regards,
Sara

Reply

Shehu Bello

December 4, 2018 at 11:14 pm

Good job ! Is it possible to add image in the data folder for HTML file to reference it? Is it also possible to use bootstrap with the help of SPIFFS? Thanks a lot

Reply

Sara Santos

December 5, 2018 at 5:59 pm

Hi.
Yes, you can use bootstrap.
I think you can also add images in the data folder, but you need to be careful with the space it occupies (I haven't tried images yet, but I think it might work).

Reply

Shehu Bello

December 4, 2018 at 11:48 pm

Hi Sara!
I am having trouble uploading file in the Data folder with the following error message:

SPIFFS not supported on esp32

Reply

Sara Santos

December 5, 2018 at 6:06 pm

Hi again.
Please update your ESP32 repository.
Try taking a look at the following issue and see if that solves your problem:
github.com/me-no-dev/arduino-esp32fs-plugin/issues/5
Have you installed the ESP32 filesystem uploader?
https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/
Regards,
Sara 🙂

Reply

Rashad

December 9, 2018 at 7:22 am

Thanks for an amazing tutorial, how to make led state update in all devices? and how many device can connect to this server ?i connected 4 device and all worked very good.only led state doesnt update in all devices. thanks

Reply

Sara Santos

December 14, 2018 at 5:20 pm

Hi Rashad.
To update the LED state you need to refresh the device's browser.
If you want it to update automatically, you can use ajax in your webserver code, for example.
You can connect as many clients as you want, as long as the connection closes after the response is sent.
Regards,
Sara 🙂

Reply

Viktor Lovas
April 8, 2021 at 2:57 pm

dear Sara!
Thank you for this tutorial. It is very helpful such a newby like me.
Do you have any suggestion/tutorial/description (or what ever) about automatically updating browsers on different devices? You mentioned ajax, but I could not find anything yet. Thank you in advance.
Regards,
Viktor

Reply

Sara Santos
April 8, 2021 at 4:41 pm

Hi Viktor.
For automatic updates, take a look a server-sent events and/or websockets:

https://randomnerdtutorials.com/esp32-web-server-sent-events-sse/
https://randomnerdtutorials.com/esp32-websocket-server-arduino/
Regards,
Sara

Reply

Lovas Viktor
April 26, 2021 at 10:10 am

Thank you, Sara!! I will have a look at it. Thnx. Regards, Viktor.

rick paul
December 29, 2018 at 10:31 pm

I tried for several hours to install the ESP32 file loader as described. It does not show up under my tools drop down. Any suggestions? I am using Arduino 1.8.8 and win10

Reply

Sara Santos
December 31, 2018 at 3:13 pm

Hi Rick.
Are you using just one version of Arduino IDE? Are you sure you're placing the .jar file on the right directory?

Have you tried our exact instructions:
https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/
You can also try to follow Github's instructions:
https://github.com/me-no-dev/arduino-esp32fs-plugin
If you could provide more details, it may be easier to find out what's wrong.
Regards,
Sara 🙂

Reply

Viktor Lovas
April 8, 2021 at 2:21 pm

Hi Rick!
I had the same problem. It took a bit amount of time to solve it:
Initially after unzipping the file from github, I got following dir structure:
/Arduino-/tools/ESP32FS-1.0/ESP32FS/tool/esp32fs.jar
after changing the directory structure it should look like following:
/Arduino-/tools/ESP32FS/tool/esp32fs.jar
(that means, there was an extra directory: ESP32FS-1.0, which I have deleted, after its content was moved into the tools directory)
After doing this change, everything works fine 😉

Reply

Daniel
January 13, 2019 at 1:05 pm

There's no option to upload ESP32 sketch data on Arduino 1.8.8. Am I missing something here I'm on Linux Solus Mate

Reply

Sara Santos

January 14, 2019 at 10:29 am

Hi Daniel.
You need to install the ESP32 Filesystem Uploader on Arduino IDE.
Follow the next tutorial to do that:
https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/
Regards,
Sara 🙂

Reply

Daniel

January 17, 2019 at 7:13 am

Thanks Sara 🙂 … I just did now the issue is I can't upload the ESP32 Sketch data. I get the following error message:

[SPIFFS] data : /home/object-undefined/Arduino/ESP32_Async_Web_Server/data
[SPIFFS] start : 2691072
[SPIFFS] size : 1468
[SPIFFS] page : 256
[SPIFFS] block : 4096

/style.css
/index.html
[SPIFFS] upload :
/tmp/arduino_build_905232/ESP32_Async_Web_Server.spiffs.bin
[SPIFFS] address: 2691072
[SPIFFS] port : /dev/ttyUSB0
[SPIFFS] speed : 921600
[SPIFFS] mode : dio
[SPIFFS] freq : 80m

Traceback (most recent call last):
File “/home/object-undefined/Arduino/hardware/espressif/esp32/tools/esptool.py”, line 35, in
import serial.tools.lists_ports as list_ports
ImportError: No module named lists_ports
SPIFFS Upload failed!

Reply

Nash

March 20, 2019 at 4:23 am

I have done everything on you guide. but Serial monitor show
Backtrace: 0x400874f8:0x3ffc64c0 0x400875f7:0x3ffc64e0 0x400d4793:0x3ffc6500 0x400eaf04:0x3ffc6530 0x400eb272:0x3ffc6550 0x400eb5a1:0x3ffc65a0 0x400ead18:0x3ffc6600 0x400eab5a:0x3ffc6650 0x400eabf3:0x3ffc6670 0x400eac3e:0x3ffc6690 0x400e9f44:0x3ffc66b0 0x400e9ef3:0x3ffc66d0 0x400d5b12:0x3ffc6700

Rebooting…
assertion “false && “item should have been present in cache”” failed: file “/Users/ficeto/Desktop/ESP32/ESP32/esp-idf-

public/components/nvs_flash/src/nvs_item_hash_list.cpp", line 85, function:
void nvs::HashList::erase(size_t)
abort() was called at PC 0x400d4793 on core 0

How can I fix it?

Reply

Sara Santos
March 21, 2019 at 11:04 am

Hi Nash.
I've searched for a while and I found some people with the same problem but using other sketches.
Unfortunately, I haven't found a clear solution for that problem:
github.com/espressif/arduino-esp32/issues/884
github.com/espressif/arduino-esp32/issues/2159
github.com/espressif/arduino-esp32/issues/1967

In this discussion, they suggest erasing the flash memory before uploading the new sketch: bbs.esp32.com/viewtopic.php?t=8888

I hope this helps in some way.

Regards,
Sara

Reply

Tracy
April 1, 2019 at 12:53 pm

Hi,
Do you have code for wifi controlled robot using ESP32 where we can control the direction of robot via web server?
Thanks . Looking forward for your reply

Reply

Musi Mumu

June 16, 2019 at 8:02 am

Hello Sara
Tolling about ESP32 in AP mode From factory is programmed to accept maximum 4 clients connections Is known that can be extended to 8 or more Can you help me to increase the maximum connections to 8 at list

Regards

Reply

Reinhart

July 12, 2019 at 1:28 am

Hello Sara,

i able to upload the sketch in my arduino ide, in the serial monitor showing the text just like above except the connecting line and ip address. it’s not showing at all. why is that?

Reply

Sara Santos

July 12, 2019 at 10:55 pm

Hi.
That usually happens when people forget to insert their network credentials.
Please make sure that you've entered your network credentials in the code and that they are correct.
Also, make sure that your ESP32 is relatively close to your router.
REgards,
Sara

Reply

Reinhart

July 13, 2019 at 4:34 am

Hello Sara,

i think found out the problem, it seems when i upload the sketch with tools->esp32 sketch data upload. it uploaded successfully, but when i run the serial monitor it kind of not work. it only read the previous uploaded sketch using the normal upload mode (ctrl-u)

Reply

dayorbit

March 19, 2020 at 8:48 am

Nice tutorial! My contribution is this link(https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/) to find tools->esp32 sketch data upload. It is not automatic. Please, update tutorial to use such link to add the menu to Arduino.

Reply

Sara Santos

March 20, 2020 at 11:07 am

Hi.
In that project, we refer at the beginning of the post, that you need to install the plugin: https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/
Regards,
Sara

Reply

Fred

August 16, 2019 at 2:22 am

Hi Sara!
I've already implemented a project using the WebServer.h library instead of

the ESPAsyncWebServer library on a ESP32.

Is there a way to handle SPIFFS with WebServer.h or I need to convert my project using the ESPAsyncWebServer library ?

Reply

Sara Santos

August 24, 2019 at 10:46 am

Hi Fred.
I think it would be easier to convert your project to the ESPASyncWebServer library.
It is not as straightforward using the WebServer library (in my opinion):
https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/examples/FSBrowser/FSBrowser.ino
Regards,
Sara

Reply

Akhilesh

August 31, 2019 at 8:12 am

Hello ,
The tutorial is very informative and explained with details. I'm curious to know whether we can upload the HTML file to SPIFFS using platform IO. is there any way to upload files in platform IO?

Reply

win

February 5, 2020 at 3:06 pm

Hi,

You have to create a folder name data/ same level as src/
Then in terminal write “pio run -t uploadfs”

Reply

Sara Santos

February 5, 2020 at 3:10 pm

Thanks for sharing that 😀

Reply

Ved

September 11, 2019 at 12:15 pm

Rebooting…
assertion “false && “item should have been present in cache”” failed: file “/Users/ficeto/Desktop/ESP32/ESP32/esp-idf-public/components/nvs_flash/src/nvs_item_hash_list.cpp”, line 85, function:

void nvs::HashList::erase(size_t)
abort() was called at PC 0x400d4793 on core 0

This error is caused when the Arduino IDE “Tools” options for ESP32 flashing MISBEHAVE. Not kidding! And it is possibly a result of a prior code that tweaked the nvs flash and SPIFFS partitions of ESP32.

Solution:
STEP 1:
Restart Arduino IDE and check if you are getting the standard options on the “Tools” tab when flashing to ESP32, i.e. Flash Freq, Size, Mode, Partition Scheme etc.

STEP 2:
If these are present, go ahead and flash your code to ESP32. Make sure all options have been selected properly. SUCCESS! EXIT!

STEP 3:
If these (Flash Freq, Size, Mode, Partition Scheme etc.) options do not appear under “Tools” tab, GO TO STEP 1, until STEP 2 becomes true.

Reply

Sara Santos
September 11, 2019 at 4:48 pm

Hi Ved.
Thanks for sharing.
I had a few readers reporting a similar issue and I didn’t know how to solve that.
I hope this helps other people.
Thank you 😀

Reply

Manoj Dixit
October 6, 2019 at 11:03 am

Hi,
Thanks for the detailed tutorial.

I want to try SPIFF with SOFTAP. Is there anything that you can help me with?

Reply

Sara Santos
October 6, 2019 at 4:46 pm

Hi.
You can take a look at this tutorial that shows how to set an ESP32 soft access point: https://randomnerdtutorials.com/esp32-access-point-ap-web-server/
I hope this helps.
Regards,
Sara

Reply

Peter Laneville
November 16, 2019 at 10:29 pm

Could someone enlighten me on what would be involved in porting this sketch/files to a secure server (HTTPS).

I can create the HTTPS server on my ESP32, connect to it but am at a loss in sending the html response from SPIFFS or in replacing the status as there doesn't seem to be an equivalent processor function.

Reply

Mani

December 31, 2019 at 1:50 am

I missed a step. It was esp32 uploader. It works perfectly now. Thanks

Reply

Alexandru Oltean

January 16, 2020 at 7:55 pm

Hi, incredible good tutorial, thanks!
i have tried to make a stepper motor run when i Route to set GPIO to HIGH or LOW. it is somehow working but every-time i hit one of the buttons my esp32 restarts
……

/ Route to set GPIO to HIGH
server.on("/on", HTTP_GET, [](AsyncWebServerRequest *request){
digitalWrite(ledPin, HIGH);

```
StepsRequired = STEPS_PER_OUT_REV * 2 ;
steppermotor.setSpeed(1000);
steppermotor.step(StepsRequired);
request->send(SPIFFS, “/index.html”, String(), false, processor);
});

// Route to set GPIO to LOW
server.on(“/off”, HTTP_GET, [](AsyncWebServerRequest *request){
digitalWrite(ledPin, LOW);
StepsRequired = – STEPS_PER_OUT_REV * 2 ;
steppermotor.setSpeed(1000);
steppermotor.step(StepsRequired);
request->send(SPIFFS, “/index.html”, String(), false, processor);
});
```

any suggestions ? Thank you .

Reply

Sara Santos
January 17, 2020 at 12:26 pm

Hi Alexandru.
What pins do you have connected to your motors?
You’re probably using pins that you shouldn’t.
See the pinout for the ESP32 here: https://randomnerdtutorials.com/esp32-pinout-reference-gpios/
Regards,
Sara

Reply

Alexandru Oltean

January 17, 2020 at 7:17 pm

Thanks a lot for the quick response !!

……

```
//Include the Arduino Stepper Library
#include
// Define Constants
// Number of steps per internal motor revolution
const float STEPS_PER_REV = 32;

// Amount of Gear Reduction
const float GEAR_RED = 64;

// Number of steps per geared output rotation
const float STEPS_PER_OUT_REV = STEPS_PER_REV * GEAR_RED;

// Define Variables

// Number of Steps Required
int StepsRequired;

// Create Instance of Stepper Class
// Connected to ULN2003 Motor Driver In1, In2, In3, In4
// Pins entered in sequence 1-3-2-4 for proper step sequencing

Stepper steppermotor(STEPS_PER_REV, 26, 33, 25, 32);

void setup()
{
}

void loop()
{
```

```
// Rotate CW 1/2 turn slowly
StepsRequired = STEPS_PER_OUT_REV * 2 ;
steppermotor.setSpeed(1000);
steppermotor.step(StepsRequired);
delay(1000);

}
```

……..

It is an NODEMCU-32S model.
i have tested only this script with my esp32 and works fine, but when i try to integrate it with // Route to set GPIO to LOW or HIGH it restarts my esp32. any suggestions? Thank you.

Reply

Amin

March 12, 2020 at 10:24 am

Hi, can you you please explain in lay man terms what is the difference between this ESP32 webserver and the other similar tutorial you have? https://randomnerdtutorials.com/esp32-web-server-arduino-ide/

Would be good to know the pros/cons of each. Thanks again!

Reply

Rui Santos

April 2, 2020 at 5:22 pm

Hello Amin, It's these projects are very similar. A web server means that your ESP hosts a web page and serves it to a client.

Reply

Amin

April 2, 2020 at 8:35 pm

Thanks for your reply. So the only difference between these two methods is that one stores the web page on the SPIFFS memory and the other in the regular memory? (SRAM?)
Is there a reason why you would have both methods? Is one better than the other in some case?

Reply

Lorenzo Garcia

April 23, 2020 at 4:55 am

Hello, I really liked your tutorial, it's fabulous. I have a doubt and I try to understand the operation of the code in detail. Why do the arguments "String ()" and "false" pass in the third and fourth arguments of the Send method? Can't they receive a null value and if this is why?
Thank you.

Reply

Sara Santos

April 23, 2020 at 10:42 am

Hi.
Because that's the way it is used to send an HTML saved on SPIFFS with processors.
You can learn more here: github.com/me-no-dev/ESPAsyncWebServer#respond-with-content-coming-from-a-file-containing-templates
Regards,
Sara

Reply

Lorenzo Garcia

April 30, 2020 at 12:48 am

Hola! Primero gracias por el compartir sus trabajo. ¿Existe alguna forma de acceder al server del ESP32 de formal on line, es decir desde fuera de la red local?
Muchas gracias!

Reply

Audrius

May 6, 2020 at 10:52 am

Your site is awesome 🙂 especially who wants to make web server. Big plus for explanation on the picture like in this post (placeholders, redirects/on and off), for me its really helped better understand the code. Best wishes 🙂

Reply

Sara Santos

May 6, 2020 at 2:28 pm

Thanks 😀

Reply

Lorenzo Garcia

June 3, 2020 at 12:45 am

Hi, is there a way to write the SPIFFS file through OTA?

Reply

Dave Walmsley

July 9, 2020 at 7:22 am

just a small thing. The Serial Monitor shows
STATE

OFFSTATE
ONSTATE
OFFSTATE
ONSTATE
OFFSTATE
ONSTATE
OFF

but I can't find where to add the cr/lf after 'OFF'/'ON'.
Changing the Serial.print(ledState); to a println() seems to lockup.

Dave

Reply

John Anderson
January 19, 2021 at 9:37 pm

This same question was bugging me Dave.

Seeing OFFSTATE ONSTATE …. repeatedly scrolling on the serial monitor looked a bit meaningless so I replaced Serial.println(var); (at the start of the processor function) with:
Serial.println(" ");
Serial.print(var);
Serial.print(" is ");
– which makes the display a bit more readable.

Reply

Anthony

September 23, 2020 at 2:44 am

Hi,
I am new to this.
Where can I download the library (ESPAsyncWebServer.h and SPIFFS.h) and how to add to the IDE?

Reply

Sara Santos

September 23, 2020 at 4:43 pm

Hi.
You don’t need to download the SPIIFS.h libary. It is included in the ESP32 Arduino code by default.
The ESPAsyncWebServer library can be downloaded using this link: https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip
Regards,
Sara

Reply

Rinke

October 14, 2020 at 7:43 am

Dear Sara,

Thank you for your superb site!

When using SPIFFS together with SofwareSerial to monitor the CO2 concentration with a MHZ19 sensor, I observed a reboot upon SPIFFS access with the message: Guru Meditation Error: Core 1 panic'ed (Cache disabled but cached memory region accessed). This problem has also been reported elsewhere when using SPIFFS together with SoftwareSerial.

The solution is to make sure that during SPIFFS operations no interrupts occur. That is: all serial input must be completely disabled. To do this, one can put the SPIFFS operation(s) in a seperate function as indicated in the example below.

///

SoftwareSerial CO2_Serial(CO2_RX, CO2_TX);

void handleSPIFFSactions() {
CO2_Serial.end(); //Before SPIFFS Access, stop the Software Serial connection
gpio_intr_disable((gpio_num_t)CO2_RX); //Also disable the ESP32 pin used for RX
delay(500); //Optional safety delay
writeFile(SPIFFS, "/ABCon.txt", ABCon); //Carry out the SPIFFS action(s)
delay(500); //Optional safety delay
gpio_intr_enable((gpio_num_t)CO2_RX); //Re-enable the ESP32 pin used for RX
CO2_Serial.begin(9600); //Restart the serial communication
}

Reply

brandden

October 26, 2020 at 2:36 pm

hi may i know what i need to change for the arduino sketch if i am using an esp8266 instead? especially for the “AsyncWebServerRequest *request” portion. thank you

Reply

Sara Santos

October 26, 2020 at 11:08 pm

Hi.
We have a similar tutorial for the ESP8266. Here it is:
https://randomnerdtutorials.com/esp8266-web-server-spiffs-nodemcu/
Regards,
Sara

Reply

Martin

October 31, 2020 at 7:19 pm

Hello first of all great tutorial and it is working as explained.
But I have a bigger ESP32 project and therefore I want to move all the AsyncWebServer stuff in a own cpp/h file in a class (eg MyWebServer) and also all the processor() and server.on(…) methods since my .ino file is already big enough.
But unfortunatly I have big problems to get it compile :-/

Can you give me some hint how that can be achived?
Ideally I want to create just the MyWebServer object in the .ino file and all the

other stuff is coded in another cpp/h file

thanks & best regards
Martin

Reply

mark
November 7, 2020 at 7:00 am

Another great tutorial! You guys are doing an awesome job.

Can you do a micropython tut on how to web serve an html file from SPIFFs on an ESP32? I have my ESP AP and web server set up, via your tutorials, but would like to use already created code and images.

Thanks

Reply

Sara Santos
November 7, 2020 at 10:41 am

Hi Mark.
At the moment, we don't have any tutorials covering that subject.
Maybe the following resources might help you:
techtutorialsx.com/2017/06/04/esp32-esp8266-micropython-uploading-files-to-the-file-system/
http://docs.micropython.org/en/v1.9.3/esp8266/esp8266/tutorial/filesystem.html

Regards,
Sara

Reply

barry
November 20, 2020 at 8:27 pm

Hi Sara,
Very well done tutorials. I've read several and am trying to put together something a bit more complex using an ESP32.

I'd like to have an array of buttons in a table 2 rows x 3 columns. Each cell in the table will be a button. There is a image in each cell (50x50px). Clicking the button will change the image and there will be a associated function on the ESP32(drive a pin or something).

It appears I will need to create a unique button and a unique

server.on("/on", HTTP_GET, [](AsyncWebServerRequest *request){
digitalWrite(ledPin, HIGH);
request->send(SPIFFS, "/index.html", String(), false, processor);
});

for each button. The contents of the server.on("/on <— will have to indicate which particular cell was clicked. Ex: R2C3 (row, col)

Am I heading in the proper direction?

Thanks much!
Barry

Reply

Sara Santos

November 21, 2020 at 11:47 am

Hi Barry.
I recommend taking a look at this tutorial that shows how to control multiple GPIOs: https://randomnerdtutorials.com/esp32-async-web-server-espasyncwebserver-library/
You can then use SPIFFS to save the HTML and CSS files.
Regards,
Sara

Reply

Barry

November 22, 2020 at 6:53 pm

Thanks much for the link.
Is there any reason I CAN'T use array style references? Ex: button[n][state]
It may be a pain to parse the passed string or possibly the HTML won't accept an array.
I'm a bit outside my areas of experience and plowing through docs is really time consuming.

Thanks again!

Reply

Jó Sales

December 30, 2020 at 6:25 pm

Hello. Its required to use the “ESP32 Filesystem Uploader Plugin” to works? I tried to load the program using the default function in Arduino IDE but I cannot to connect to IP neither see the html page.

Thank you.

Reply

Sara Santos

December 30, 2020 at 7:27 pm

Hi.
Yes, to follow this tutorial, you need to have the plugin installed in your Arduino IDE.
Regards,
Sara

Reply

Jó Sales

December 31, 2020 at 12:02 pm

Hello, thanks for the fast reply. Could you explain or point where I can undestand why this only works when use the plugin?

Thank you for all help.

Reply

Sara Santos

January 3, 2021 at 12:02 pm

Hi.
You need the plugin to upload the HTML and CSS files to the board.
Regards,
Sara

Reply

John Anderson

January 19, 2021 at 9:38 pm

This same question was bugging me Dave (9 July 2020).

Seeing OFFSTATE ONSTATE …. repeatedly scrolling on the serial monitor looked a bit meaningless so I replaced Serial.println(var); (at the start of the processor function) with:
Serial.println(" ");
Serial.print(var);
Serial.print(" is ");
– which makes the display a bit more readable.

Reply

Matt

February 1, 2021 at 5:12 pm

Hello, great site, I managed to get it working and i have been trying unsuccessfully to load a image from my index.html.
I have teh images file loaded in spiffs, and my html code is correct but it doesnt find the image…
Do i have to do something with my routes?
Seems it is not as simple as just adding to the html code and uploading the image

Reply

Sara Santos

February 1, 2021 at 10:03 pm

Hi.
You also need to add some lines to handle the request for the image. For example, if your image is called example.png, you should add something line this:

server.on(“/example”, HTTP_GET, [](AsyncWebServerRequest *request){
request->send(SPIFFS, “/example.png”, “image/png”);
});
I think it would be useful taking a look at this tutorial: https://randomnerdtutorials.com/display-images-esp32-esp8266-web-server/ /see Option 2)
Regards,
Sara

Reply

Bruce

May 29, 2021 at 12:50 pm

Thank you this is very useful is there a way to add ina219 reading to this webpage ?

Reply

ALBERTO CARVALHO BRANQUINHO

June 9, 2021 at 6:15 pm

Hi,

Fantastic site, thanks a lot.

Is the #include “SPIFFS.h” correct? I’m receving the message “SPIFFS.h: No such file or directory”

Trying #include “FS.h” I receive the messagem bellow.

C:\Users\alber\Documents\Arduino\ESP32_Async_Web_Server\ESP32_Async_Web_Server.ino: In function ‘void setup()’:

ESP32_Async_Web_Server:45:24: error: no matching function for call to ‘fs::FS::begin(bool)’

45 | if(!SPIFFS.begin(true)){

Thanks.

Reply

Sara Santos

June 10, 2021 at 1:53 pm

Hi.
Make sure you have an ESP32 board selected in Tools > Boards.
Regards,
Sara

Reply

dante

June 15, 2021 at 3:05 pm

hola no me muestra la pagina me dise pagina no encontrda

Reply

Sara Santos

June 15, 2021 at 3:32 pm

That’s because you’ve probably didn’t upload the files to your board.
Regards,
Sara

Reply

dante

June 16, 2021 at 7:39 pm

SE SUPOINE QUE LOS ESTOY CARGANDO CON VISUAL CODE
ME PODRIAS APOYAR NOSE QUE PASA

Reply

Sara Santos

June 17, 2021 at 1:08 pm

Hola.
Puedes leer este tutorial sobre VS Code y SPIFFS:
https://randomnerdtutorials.com/esp32-vs-code-platformio-spiffs/
Saludos
Sara Santos

Reply

Bernard PIERRE

August 5, 2021 at 2:42 pm

Bonjour Sara,

Ce code fonctionne parfaitement!
J’ai inséré image dans la page dans SPIFFS que j’intègre dans ma page HTMLpar ce code
que j’ai ajouté:

server.on(“/image_2”, HTTP_GET, [](AsyncWebServerRequest *request){
request->send(SPIFFS, “/image_2.png”, “image/png”);
});

Mon PB c’est que la page se recharge complétement quand je clique sur le bouton (qui fonctionne)?
est ce normal?
Comment je peux faire pour éviter cela?

Merci pour vos tutos qui sont géniaux!!

Reply

Sara Santos
August 6, 2021 at 11:22 am

Hi.
Take a look at this tutorial that might help: https://randomnerdtutorials.com/display-images-esp32-esp8266-web-server/
Regards,
Sara

Reply

uecken

August 9, 2021 at 8:40 am

Hi. I want to play mp3 file in SPIFFS by accessing ESP32 Web server.
First, I must set mp3 source file from SPIFFS to “src of source tag”.
Below is the code.

However, I confused how should set the mp3 data to src.
Can you help me?

Reply

Sara Santos

August 9, 2021 at 9:31 am

Hi.
This might help: https://www.w3schools.com/html/html5_audio.asp
Regards,
Sara

Reply

uecken

August 9, 2021 at 2:04 pm

Sorry, my code was not written.
Yes, like you posted, this audio src is “horse.mp3”

Your browser does not support the audio element.

However, I don't know how this '<source src=' can read mp3 file from SPIFFS.

Reply

Tito

September 12, 2021 at 4:01 am

Hi Sara/Rui:
Can we use this project "ESP32 Web Server using SPIFFS (SPI Flash File System)" with Access Point? Means, not using router, can we run webpage from access point where index.html will be inside SPIFFS? Please guide me

Reply

Sara Santos

September 14, 2021 at 9:57 am

Hi.
Yes, you can do that.
There's no problem.
To learn how to set an access point, you can check this tutorial:
https://randomnerdtutorials.com/esp32-access-point-ap-web-server/
Regards,
Sara

Reply

Jesio

September 18, 2021 at 11:22 pm

ola, pode me ajudar?
Tenho 2 esp32. Cada um com seu ip com webserver. Num deles que é o ip 192.168.0.7 tem um html e nele um link que quando clicar vá para o outro esp32 no ip 192.168.0.5, sendo que não consigo, ele sempre coloca o ip atual na frente e nao encontra o ip que eu quero(do outro esp).

Esse é o javascript:

```
function ligaRele1(){
var ligaRele1 = new XMLHttpRequest(); ligaRele1.open('GET', '/ligaRele1', false);
ligaRele1.send();
if (ligaRele1.readyState === 4 && ligaRele1.status === 200) {
window.location.replace("192.168.0.5");
}
}
```

na barra de endereço ele manda dessa forma:
http://192.168.0.7/192.168.0.5

nao sei porque ele esta inserindo o ip 192.168.0.7 na frente, impodindo o redirecionamento…

pode me ajudar?
att

Reply

Denis

October 2, 2021 at 9:09 am

Hello Sara, very good tutorial, I used your sketch directly for testing, following the instructions step by step, but in my browser nothing is displayed, I am using an ESP32-C3-DevKitM-1 card with 4Mb with SPIFFS . What's going on ?
Thank you in advance for your return.
Regards

Reply

Sara Santos

October 2, 2021 at 10:49 am

Hi.
Did you successfully upload the files to the filesystem?
Regards,
Sara

Reply

francis

October 12, 2021 at 1:47 pm

@Denis,
1/ Did you succeed to compile and download the sketch with this ESP32-C3 ?
I also have this dev board and cannot compile.
2/ The sketch data uploader didn't work also for me giving target error while it work for ESP32 wroom

Reply

francis
October 15, 2021 at 4:51 am

Problem solved :
1/ I find out that I have old versing for Asyncwebserver andAsyncTCP that was used instead of the latest one and also in the additional board manager I have linked the original ESP32 and later le esp32_dev that sound conflicting.
2/ Orginal data uploader didn't work for ESP32-C3 you have to install a forked one for that you can find here : https://github.com/lorol/arduino-esp32fs-plugin/releases

Reply

Ricardo Hortigüela
November 11, 2021 at 12:25 pm

I'm trying to setup my esp32, but I can't manage to compile the poject. The problems seems to be that there are problems with the WiFi.h library, as there

are duplicates:
Multiple libraries were found for “WiFi.h”
Used:
C:\Users\rhort\Documents\ArduinoData\packages\esp32\hardware\esp32\1.0.6\libraries\WiFi
Not used: C:\Program Files\WindowsApps\ArduinoLLC.ArduinoIDE_1.8.51.0_x86__mdqgnx93n4wtt\libraries\WiFi
exit status 1
Error compiling for board DOIT ESP32 DEVKIT V1.

Any idea on how to fix this issue?

Reply

Sara Santos
November 11, 2021 at 4:07 pm

Hi.
The problem is not with Wi-Fi.
It is something else. What line is highlighted when you get the error?
Regards,
Sara

Reply

Joan
January 1, 2022 at 7:15 pm

Hi Sara
Thankyou for your post! It’s very useful.

I am wondering how can deploy if we need more than one string to return to the client. For example, %STATE% and %VERSION%.

I added a new “if (var==VERSION)” on the processor code and a new line on HTML file ”

Version: **%VERSION%**

“, but it only returns the %STATE%.

Could you help me please?

thanks and have a Happy New Year!!!

Reply

Sara Santos
January 3, 2022 at 2:57 pm

Hi.
Try the following:

```
else if (var==VERSION) {
(your code here)
}
```

Regards,
Sara

Reply

Neil Wrightson

January 20, 2022 at 10:45 pm

Hi,
In my application, I'm logging information to a file (sensor.log) in spiffs.
I'm using PlatformIO/VSC/Win10, how do I download this file from the ESP32 spiffs area back to the PC?
Everything PlatformIO/ElegantOTA seams to be upload only with no means of downloading files.

Reply

Henry

March 1, 2022 at 5:52 pm

Hey, i wonder how to create html, css, and the js code before implement them to the arduino file so we could upload it to the spiffs system. I mean it should be any mistake when we develop the code to display what we really want in the browser, how to develop and like compile it? Do you use another software to build the like vs code or atom?

How to do that? Do you have any references for me?

Reply

Sara Santos

March 1, 2022 at 7:15 pm

Hi.
You can create those files on VS Code for example.
There is a plugin for VS Code that allows you to create a live local server with those files to check is everything is working as expected.
The plugin is called “live server”.
Regards,
Sara

Reply

agung

March 28, 2022 at 5:32 pm

Hello team, when i run my ip then the page show “page not found”. Why should i do?

Reply

Sara Santos

March 28, 2022 at 10:34 pm

Hi.
I’m sorry, I didn’t understand your question.
Can you reformulate it?

Regards,
sara

Reply

Sara Santos
March 30, 2022 at 10:00 am

Hi.
You probably didn't upload the files to the filesystem.
Regards,
Sara

Reply

Ari
April 20, 2022 at 7:41 am

Hi sara,
thanks for this great tutorial, i have problem…
have you face error like this?

E (154187) task_wdt: Task watchdog got triggered. The following tasks did not reset the watchdog in time:
E (154187) task_wdt: – async_tcp (CPU 0/1)
E (154187) task_wdt: Tasks currently running:
E (154187) task_wdt: CPU 0: IDLE0
E (154187) task_wdt: CPU 1: IDLE1

E (154187) task_wdt: Aborting.
abort() was called at PC 0x4013f00c on core 0

ELF file SHA256: 0000000000000000

Backtrace: 0x400886d8:0x3ffbf860 0x40088955:0x3ffbf880 0x4013f00c:0x3ffbf8a0 0x40086ea1:0x3ffbf8c0 0x40151007:0x3ffbc220 0x40140907:0x3ffbc240 0x4008b115:0x3ffbc260 0x40089966:0x3ffbc280

Rebooting…

I got that when I input string to webserver?

Reply

Jacque Wilson
July 6, 2022 at 3:08 pm

Hi… love this site! Can you tell/show how to push data files, like ones in this tutorial, using VSCODE/Platformio?

Reply

Sara Santos
July 6, 2022 at 3:59 pm

Hi.
Follow this tutorial: https://randomnerdtutorials.com/esp32-vs-code-platformio-spiffs/

Regards,
Sara

Reply

Jacque Wilson
July 6, 2022 at 5:09 pm

Sorry I missed that. Thank you for quick response! I am learning so much. Without a doubt, one of the best microprocesser tutorial sites on the internet!

Reply

Richard Pitre
July 10, 2022 at 11:30 pm

It works great using Plaform IO

Reply

Mark
August 11, 2022 at 11:27 am

Hi Sara,
Thanks for the great tutorial. I have had an issue when trying to upload the

receiver code, would you have any idea how I can overcome it…

esptool.py v3.1
Serial port COM6
Traceback (most recent call last):
File “esptool.py”, line 4582, in
File “esptool.py”, line 4575, in _main
File “esptool.py”, line 4074, in main
File “esptool.py”, line 120, in get_default_connected_device
File “esptool.py”, line 313, in **init**
File “serial__init__.py”, line 90, in serial_for_url
File “serial\serialwin32.py”, line 64, in open
serial.serialutil.SerialException: could not open port ‘COM6’:
FileNotFoundError(2, ‘The
system cannot find the file specified.’, None, 2)
[31616] Failed to execute script esptool
SPIFFS Upload failed!

Thanks

Reply

Sara Santos
August 12, 2022 at 10:25 am

Hi.
Make sure you close all serial connections with the board (Serial Monitor) before trying to upload the files to SPIFFS.
Regards,
Sara

Reply

Laurent

September 14, 2022 at 4:36 pm

If I may, as I had same issue, close the debug window. With the trace window open, it didn t work at all

Laurent

Reply

Dave W.

September 9, 2022 at 10:53 pm

Hi Sara,

I've been trying to figure out how to add an AsyncWebServer for a few days now, and I've searched through all the examples I can find. All of them assume you're using the Arduino IDE, which I'm sure is easier to explain. What if you're using esp-idf with the Arduino libs as a component, and building with ninja? (I'm using ESP32). Now it becomes much harder! Which is where I am. If the build system used gmake with standard makefiles, I'd have it working in short order. But CMake seems like a rats nest to me. I need to figure out how to get CMake to add include paths to these new libs, and then build them. Perhaps this is beyond the scope of this tutorial, but a clear step-by-step on how to do this is sorely needed!

Reply

Sara Santos

September 10, 2022 at 4:10 pm

Hi.
I'm sorry but I'm not familiar with those subjects.
Regards,
Sara

Reply

Muthukumar

October 17, 2022 at 1:38 pm

Hi Sara,

I have doubt here,

ESP32 Web Server

GPIO state: **%STATE%**

ON

OFF

Is it possible to make a single button to map a gpio pin(26) to ON and OFF. Second button to map to 33 with another button.

I am expecting, 26 gpio pin default state as OFF, this will switch to ON when I press this button, same page I want to switch back to OFF state. I want to load only my custom css, js, html files.

Similarly, gpio 33 for another button operation.

Unlike code:(some other website I saw this approach)
server.on("/servo", HTTP_GET, [](AsyncWebServerRequest *request){
String pageHTML = "0 degree90 degree180 degree";

```
// Add the following 'if' statement to the callback function
if (request->hasParam("angle")) {
String angle = request->getParam("angle")->value();
if (angle == "0"){
myservo.write(0);
} else if (angle == "90"){
myservo.write(90);
} else if (angle == "180"){
myservo.write(180);
}
}
// Make sure the code is inside the last curly closing bracket!
});
```

Thanks,
Muthukumar

Reply

Franklin

November 29, 2022 at 6:53 pm

Hi Sara,

I have a question regards the placeholders.
I'm writing a code for my project and I'm using this tutorial.

The problem I have is that I want to use the modulo (remainder) which is the same symbol (%) as for the placeholders.

Because I use multiple remainders (so multiple % symbols, but not for placeholders!) the code wants to fill in the space between the two remainder symbols instead of calculating the remainder left over when one operand is divided by a second operand

This leads to my code not running properly since the code between the two remainder symbols has vanished and the code is then incorrect.

Is there any way to use the modulo symbol for my remainder function instead for the placeholder?

I'm stuck and don't know what to do!
Kind regards,
Franklin

Reply

Sara Santos
November 29, 2022 at 11:43 pm

Hi.
You need to use the corresponding HTML entity instead of the % sign:
https://unicode-table.com/en/0025/
Regards,
Sara

Reply

Baaskar

March 15, 2023 at 5:48 pm

Hi,
Thanks for the detailed post.
I am new to ESP32 and Wifi Projects.
I learned new things very easily from your web page.

How can i control the same from outside my network.
kindly looking for your guideline

Reply

Sara Santos

March 16, 2023 at 10:36 am

Hi.
The easiest way is to create a secure tunnel to your network using ngrok or cloudflare zero trust, for example.
Regards,
Sara

Reply

Arnaud

May 16, 2023 at 2:13 pm

Hi Sara !

Thank you for the tutorial.
One question though, can we add html to store in the build'in memory to a bin file in order to update it with OTA mechanism ?

Best regards,
Arnaud

Reply

Sara Santos

May 17, 2023 at 10:00 pm

Hi.
Yes. Check this tutorial: https://randomnerdtutorials.com/esp32-ota-over-the-air-arduino/
Check the section called "Upload Files to Filesystem OTA (Over-the-Air) Updates – ESP32".
Regards,
Sara

Reply

Christoph Brune

June 4, 2023 at 8:04 pm

Hallo Sara,

herzlichen Dank für das tolle Tutorial. Nach einigen flüchtigen Fehlern läuft es jetzt toll.
Was mir allerdings aufgefallen ist, das wenn ich zeitgleich mit 2 Endgeräten, z.B. Laptop und Handy gleichzeitig auf die Seite zugreife, und auf einem Gerät schalte, wird die Gipo-Anzeige auf dem anderen nicht umgeschaltet, auch wenn man die Seite auf dem anderen Gerät aktualisiert. Der tatsächliche Schaltzustand wird nicht synchronisiert. Will sagen: Ich schalte am Laptop etwas ein, möchte auf dem Handy z.B. wieder ausschalten, sehe aber nicht den tatsächlichen Schaltzustand und kann somit nicht sicher erkennen, ob das Gerät an oder aus ist. Das ist für eine praktische Nutzung etwas hinderlich, oder ?
Vieleicht gibt es ja auch eine Lösung für das Problem ?

Danke und liebe Grüße
Christoph

Reply

MICHEL

August 7, 2023 at 8:20 am

Hi Sara, thanks for all these quality tutorials you've posted.
Only a question: is it possible to use HTML and CSS files with the WebServer library? I did that but only with a HTML file and I don't know how to send the CSS file after.

Reply

Sara Santos

August 7, 2023 at 10:03 am

Hi.
What specific library are you referring to?
Are you referring to the library we use in this tutorial?
Regards,
Sara

Reply

MICHEL

September 11, 2023 at 2:57 pm

I'm referring to the simple WebServer.h library.
Since I asked the question, I've found how to send CSS and JS files.
I don't use ESPAsyncWebServer.h library because i need only one client a a time.
And I haven't really understood the syntaxe:

```
server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest *request){
request->send(SPIFFS, "/style.css","text/css");
});
```

Reply

MICHEL

September 11, 2023 at 3:28 pm

with ESPAsyncWebServer: how can I call an API for exemple with WebServer:
if(serveur.argName(0) == “Cmd” && serveur.arg(0) == “O” && serveur.argName(1) == “volet”){
equipement = serveur.arg(1);
action = “Ouverture”;
taux = serveur.arg(2);
reponse = “Ok Volet ” + equipement + ” en ” + action + ” a ” + taux + “%”;
}

Reply

Leave a Comment

Name *

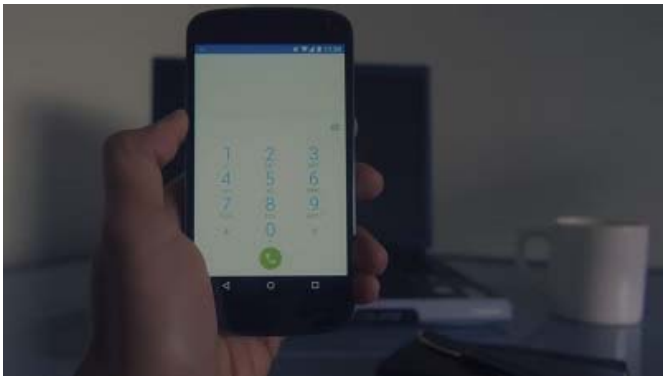
Email *

Website

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

Post Comment



Build Web Servers with ESP32 and ESP8266 » boards to control outputs and monitor sensors remotely.