

# ESP32 OTA (Over-the-Air) Updates – AsyncElegantOTA using Arduino IDE

In this guide, you'll learn how to do over-the-air (OTA) updates to your ESP32 boards using the AsyncElegantOTA library. This library creates a web server that allows you to upload new firmware (a new sketch) to your board without the need to make a serial connection between the ESP32 and your computer.

Additionally, with this library, you can also upload new files to the ESP32 filesystem (SPIFFS). The library is very easy to use, and it's compatible with the ESPAsyncWebServer library that we often use to build web server projects.



By the end of this tutorial, you'll be able to easily add OTA capabilities to your web server projects with the ESP32 to upload new firmware and files to the filesystem wirelessly in the future.

We have a similar tutorial for the ESP8266 NodeMCU board: [ESP8266 NodeMCU OTA \(Over-the-Air\) Updates – AsyncElegantOTA using Arduino IDE](#)

## Watch the Video Tutorial

This project is available in video format and in written format. You can watch the video below or you can scroll down for the written instructions.

ESP32 OTA (Over-the-Air) Updates using AsyncEl...



## Overview

This tutorial covers:

- [Add the ElegantOTA feature to your web server](#)
- [Upload new firmware via OTA to ESP32 board](#)
- [Upload files to SPIFFS via OTA to ESP32 board](#)

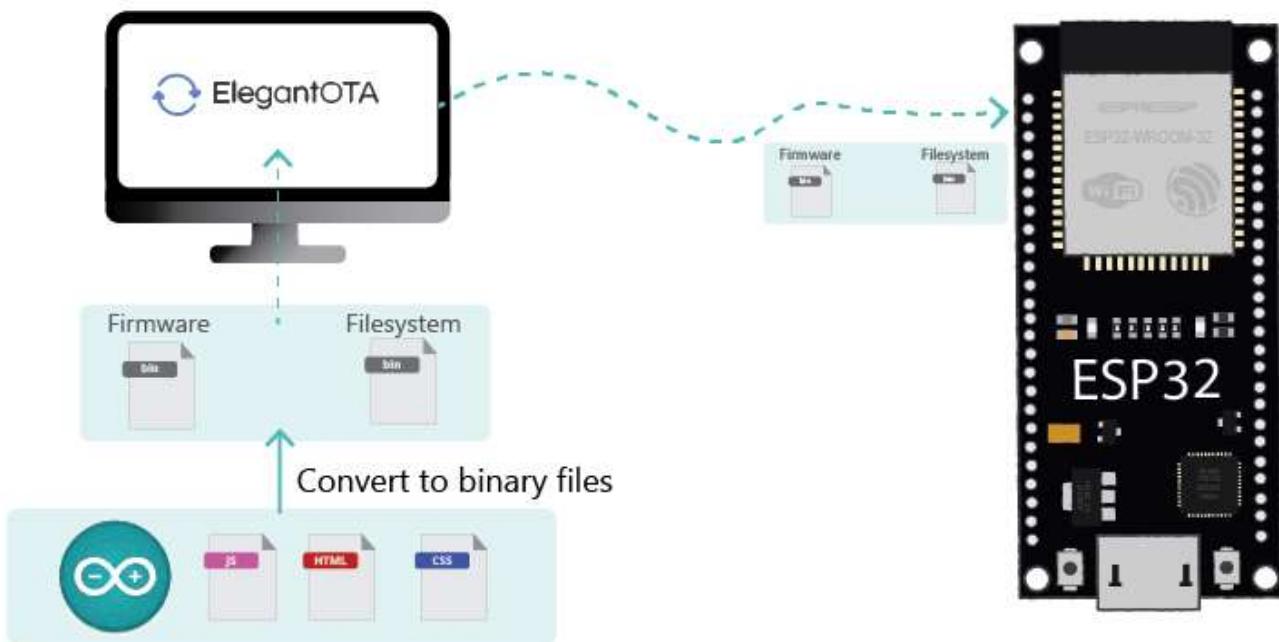
We recommend that you follow all the tutorial steps to understand how ElegantOTA works and how you can use it in your projects. To demonstrate how to do this, we'll

# ESP32 OTA (Over-the-Air) Programming

OTA (Over-the-Air) update is the process of loading new firmware to the ESP32 board using a Wi-Fi connection rather than a serial communication. This functionality is extremely useful in case of no physical access to the ESP32 board.

There are different ways to perform OTA updates. In this tutorial, we'll cover how to do that using the [AsyncElegantOTA library](#). In our opinion, this is one of the best and easiest ways to perform OTA updates.

The AsyncElegantOTA library creates a web server that you can access on your local network to upload new firmware or files to the filesystem (SPIFFS). The files you upload should be in *.bin* format. We'll show you later in the tutorial how to convert your files to *.bin* format.



The only disadvantage of OTA programming is that you need to add the code for OTA in every sketch you upload so that you're able to use OTA in the future. In the case of the AsyncElegantOTA library, it consists of just three lines of code.

## AsyncElegantOTA Library

there is the *BasicOTA* example (that never worked well for us); the [OTA Web Updater](#) (works well, but it isn't easy to integrate with web servers using the `ESPAsyncWebServer` library); and many other examples from different libraries.

Most of our web server projects with the ESP32 use the [ESPAsyncWebServer library](#). So, we wanted a solution that was compatible with that library. The [AsyncElegantOTA library](#) is just perfect for what we want:



- It is compatible with the `ESPAsyncWebServer` library;
- You just need to add three lines of code to add OTA capabilities to your “regular” Async Web Server;
- It allows you to update not only new firmware to the board but also files to the ESP32 filesystem (`SPIFFS`);
- It provides a beautiful and modern web server interface;
- It works extremely well.

If you like this library and you'll use it in your projects, consider supporting the developer's work.

## OTA Updates with AsyncElegantOTA Library – Quick Summary

To add OTA capabilities to your projects using the `AsyncElegantOTA` library, follow these steps:

1. Install [AsyncElegantOTA](#), [AsyncTCP](#), and [ESPAsyncWebServer](#) libraries;
2. Include `AsyncElegantOTA` library at the top of the Arduino sketch:  
`#include <AsyncElegantOTA.h>;`
3. Add this line `AsyncElegantOTA.begin(&server);` before  
`server.begin();`

4. Open your browser and go to `http://<IPAddress>/update`, where `<IPAddress>` is your ESP32 IP address.

Continue reading the tutorial for more detailed steps.

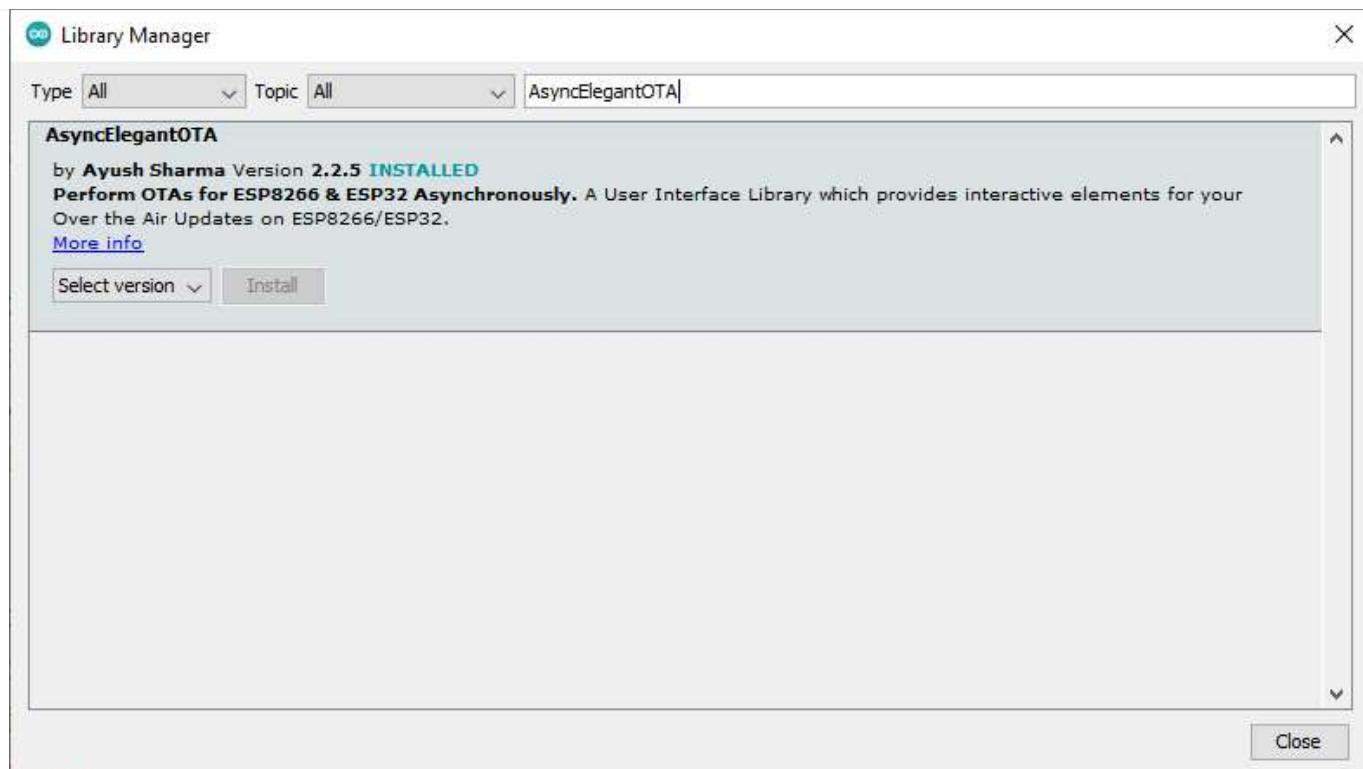
## How does OTA Web Updater Work?

- The first sketch should be uploaded via serial port. This sketch should contain the code to create the OTA Web Updater so that you are able to upload code later using your browser.
- The OTA Web Updater sketch creates a web server you can access to upload a new sketch via web browser.
- Then, you need to implement OTA routines in every sketch you upload so that you're able to do the next updates/uploads over-the-air.
- If you upload a code without an OTA routine, you'll no longer be able to access the web server and upload a new sketch over-the-air.

## Install AsyncElegantOTA Library

In this tutorial, the ESP32 will be programmed using Arduino IDE. If you want to learn how to do the same using VS Code + PlatformIO, follow the next tutorial: [ESP32 OTA \(Over-the-Air\) Updates – AsyncElegantOTA \(VS Code + PlatformIO\)](#)

You can install the AsyncElegantOTA library using the Arduino Library Manager. In your Arduino IDE, go to **Sketch > Include Library > Manage Libraries...** Search for “**AsyncElegantOTA**” and install it.



## Install AsyncTCP and ESPAsyncWebServer Libraries

You also need to install the AsyncTCP and the ESPAsyncWebServer libraries. Click the links below to download the libraries.

- [ESPAsyncWebServer](#)
- [AsyncTCP](#)

These libraries aren't available to install through the Arduino Library Manager, so you need to copy the library files to the Arduino Installation Libraries folder. Alternatively, in your Arduino IDE, you can go to **Sketch > Include Library > Add .zip Library** and select the libraries you've just downloaded.

## AsyncElegantOTA ESP32 Basic Example

Let's start with the basic example provided by the library. This example creates a simple web server with the ESP32. The root URL displays some text, and the /update URL displays the interface to update firmware and filesystem.

Copy the following code to your Arduino IDE

☰ Menu



/\*

Rui Santos

Complete project details

- Arduino IDE: <https://RandomNerdTutorials.com/esp32-ota-over-the-air/>
- VS Code: <https://RandomNerdTutorials.com/esp32-ota-over-the-air-vs-code/>

This sketch shows a Basic example from the AsyncElegantOTA library:  
<https://github.com/ayushsharma82/AsyncElegantOTA>

\*/

```
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <AsyncElegantOTA.h>

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

AsyncWebServer server(80);

void setup(void) {
    Serial.begin(115200);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    Serial.println("");
}
```

[View raw code](#)

Insert your network credentials and the code should work straight away:

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE WITH YOUR PASSWORD":
```

[☰ Menu](#)

# How the Code Works

First, include the necessary libraries:

```
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <AsyncElegantOTA.h>
```

Insert your network credentials in the following variables so that the ESP32 can connect to your local network.

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

Create an `AsyncWebServer` object on port 80:

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

In the `setup()`, initialize the Serial Monitor:

```
Serial.begin(115200);
```

Initialize Wi-Fi:

```
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
Serial.println("");
```

```
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("");  
Serial.print("Connected to ");  
Serial.println(ssid);  
Serial.print("IP address: ");  
Serial.println(WiFi.localIP());
```

Then, handle the client requests. The following lines, send some text

Hi! I am ESP32. when you access the root ( / ) URL:

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {  
    request->send(200, "text/plain", "Hi! I am ESP32.");  
});
```

If your web server needs to handle more requests you can add them (we'll show you in the next example).

Then, add the next line to start ElegantOTA:

```
AsyncElegantOTA.begin(&server);
```

Finally, initialize the server:

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

## Access the Web Server

follows (yours may be different):

The screenshot shows a terminal window titled "COM3". The text output is as follows:

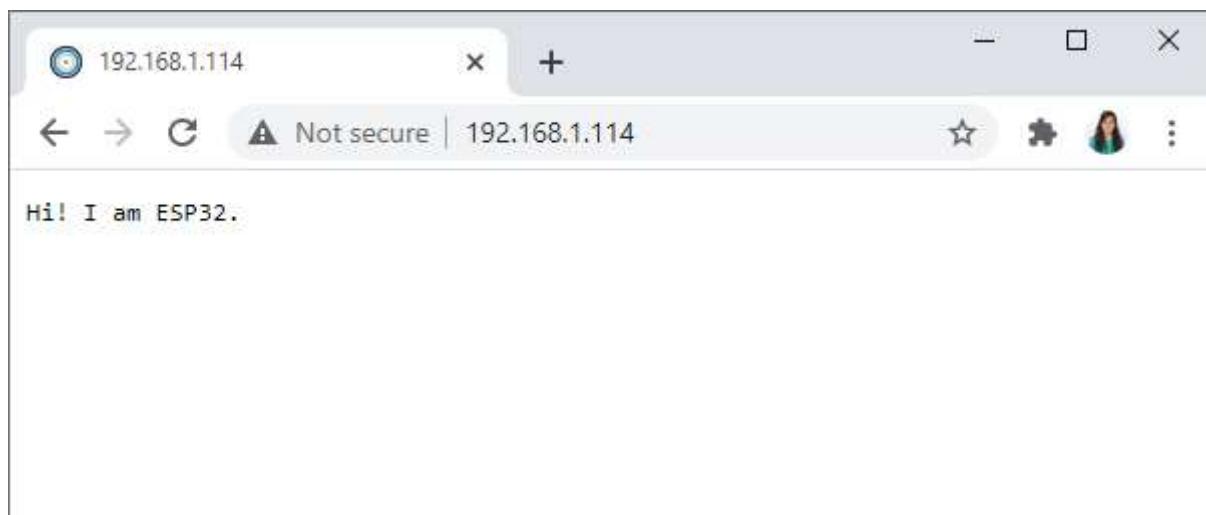
```
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac

...
Connected to 192.168.1.114
IP address: 192.168.1.114
HTTP server started
```

At the bottom of the terminal window, there are several configuration options:

- Autoscroll
- Show timestamp
- Newline dropdown menu
- 115200 baud dropdown menu
- Clear output button

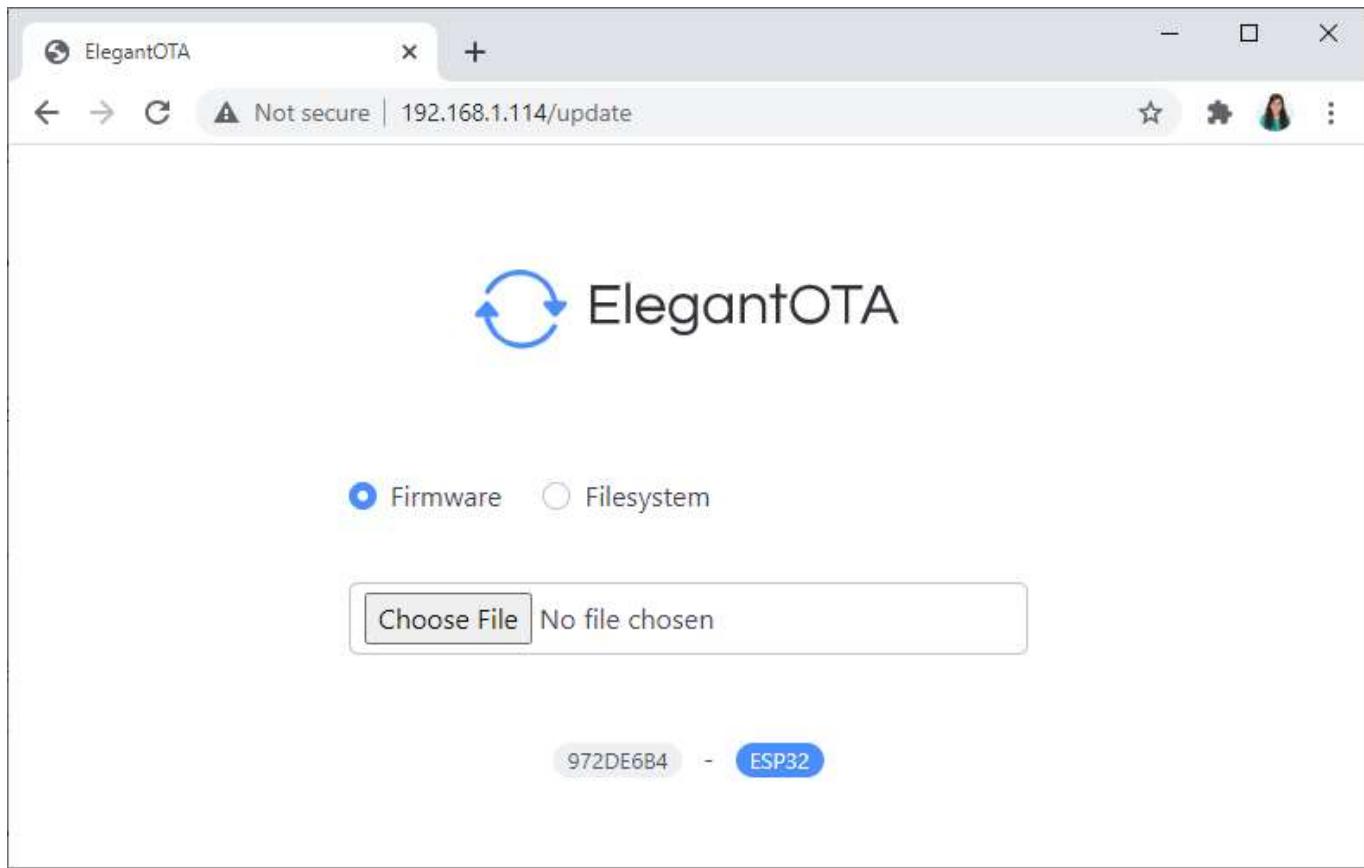
In your local network, open your browser and type the ESP32 IP address. You should get access the root ( / ) web page with some text displayed.



☰ Menu



Now, imagine that you want to modify your web server code. To do that via OTA, go to the ESP IP address followed by `/update`. The following web page should load.



Follow the next sections to learn how to upload new firmware using AsyncElegantOTA.

## Upload New Firmware OTA (Over-the-Air) Updates – ESP32

Every file that you upload via OTA should be in `.bin` format. You can generate a `.bin` file from your sketch using the Arduino IDE.

With your sketch opened, you just need to go to **Sketch > Export Compiled Binary**. A `.bin` file will be generated from your sketch. The generated file will be saved under your project folder.

That's that `.bin` file you should upload using the AsyncElegantOTA web page if you

## Upload a New Web Server Sketch – Example

Let's see a practical example. Imagine that after uploading the previous sketch, you want to upload a new one that allows you to control an LED via a web interface like [this project](#). Here's the steps you need to follow:

1. Copy the following code to your Arduino IDE. Don't forget to insert your network credentials.

```
/*
  Rui Santos
  Complete project details
  - Arduino IDE: https://RandomNerdTutorials.com/esp32-ota-over-
  - VS Code: https://RandomNerdTutorials.com/esp32-ota-over-the-
  Permission is hereby granted, free of charge, to any person obtai
  of this software and associated documentation files.

  The above copyright notice and this permission notice shall be i
  copies or substantial portions of the Software.

*/
// Import required libraries
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <AsyncElegantOTA.h>

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

bool ledState = 0;
```

[View raw code](#)

This is the same code used in [this project](#), but it contains the needed lines of code to handle ElegantOTA:

```
#include <AsyncElegantOTA.h>
```

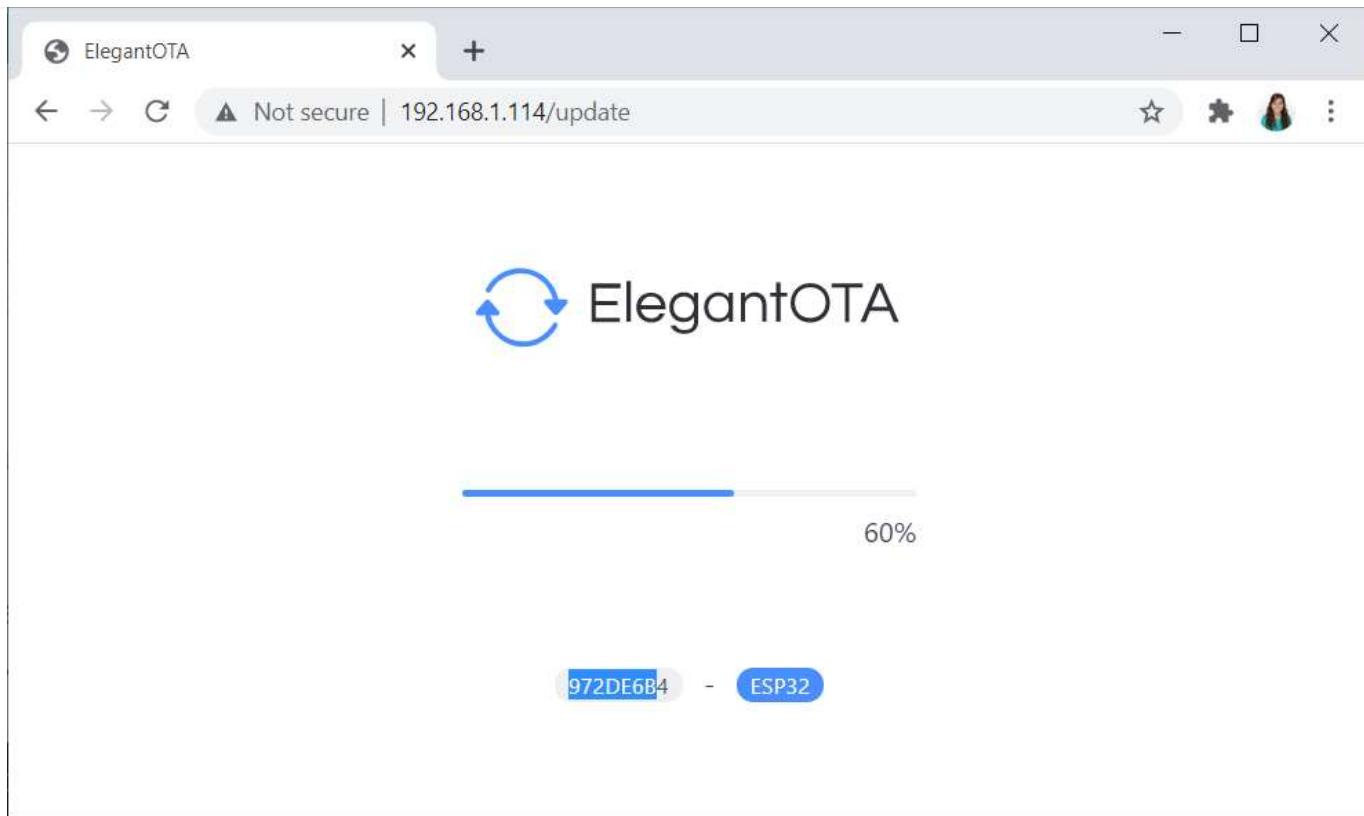
```
AsyncElegantOTA.begin(&server);
```

2. Save your sketch: **File > Save** and give it a name. For example:  
*Web\_Server\_LED\_OTA\_ESP32*.

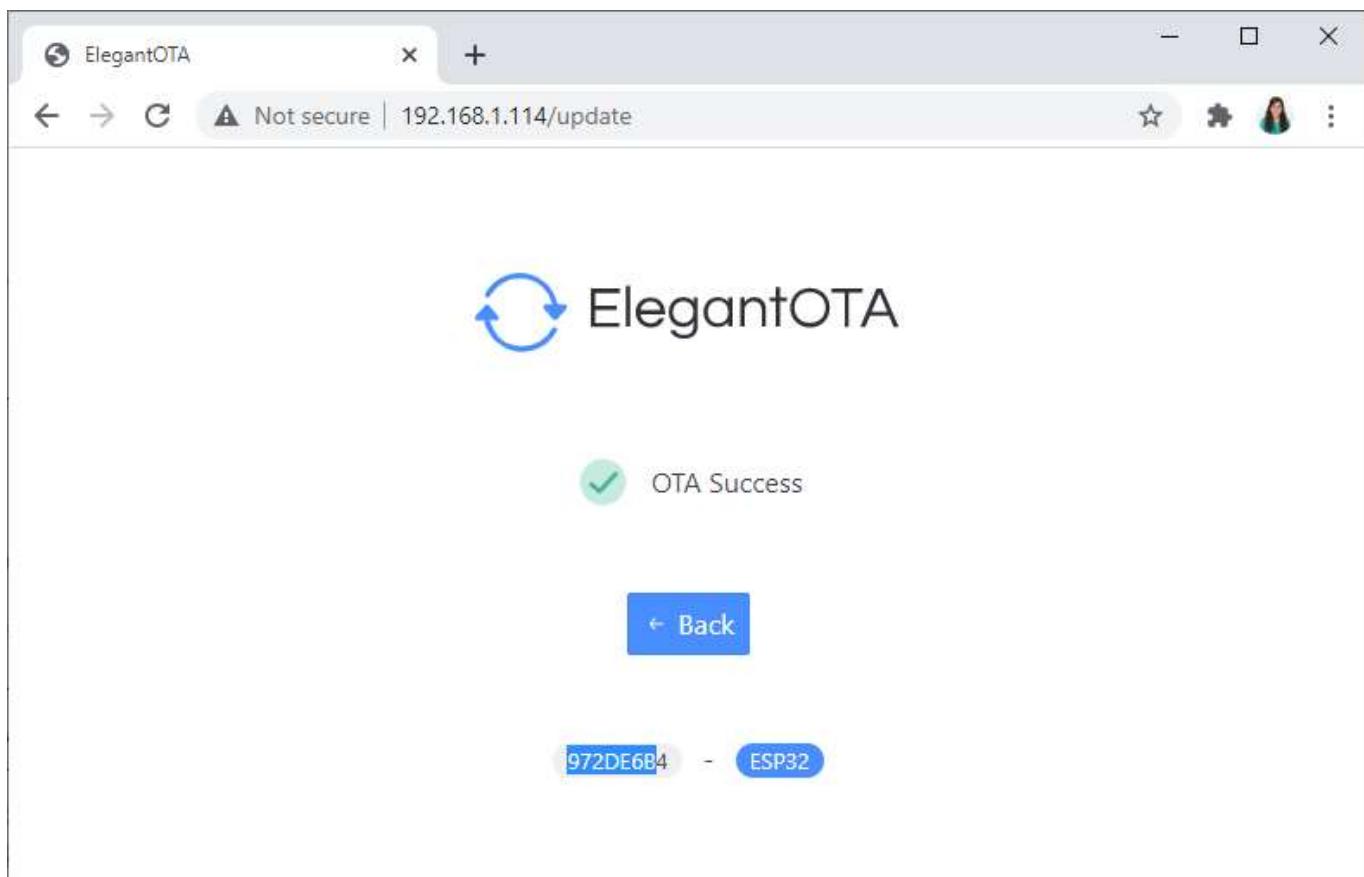
3. Generate a *.bin* file from your sketch. Go to **Sketch > Export Compiled Binary**. A new *.bin* file should be created under the project folder.



4. Now, you need to upload that file using the ElegantOTA page. Go to your ESP IP address followed by `/update`. Make sure you have the **firmware** option selected. Click on **Choose File** and select the *.bin* file you've just generated.



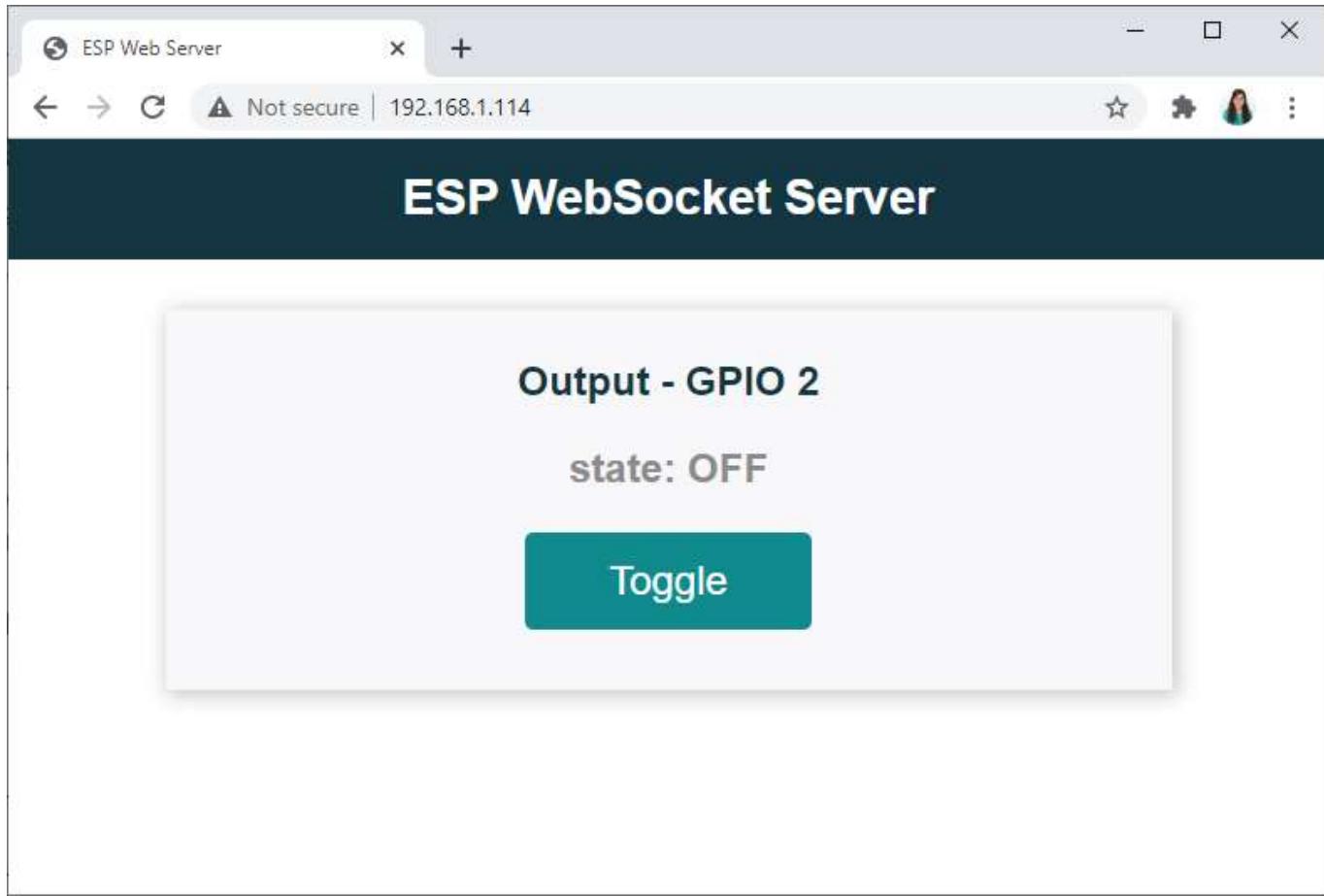
5. When it's finished, click on the **Back** button.



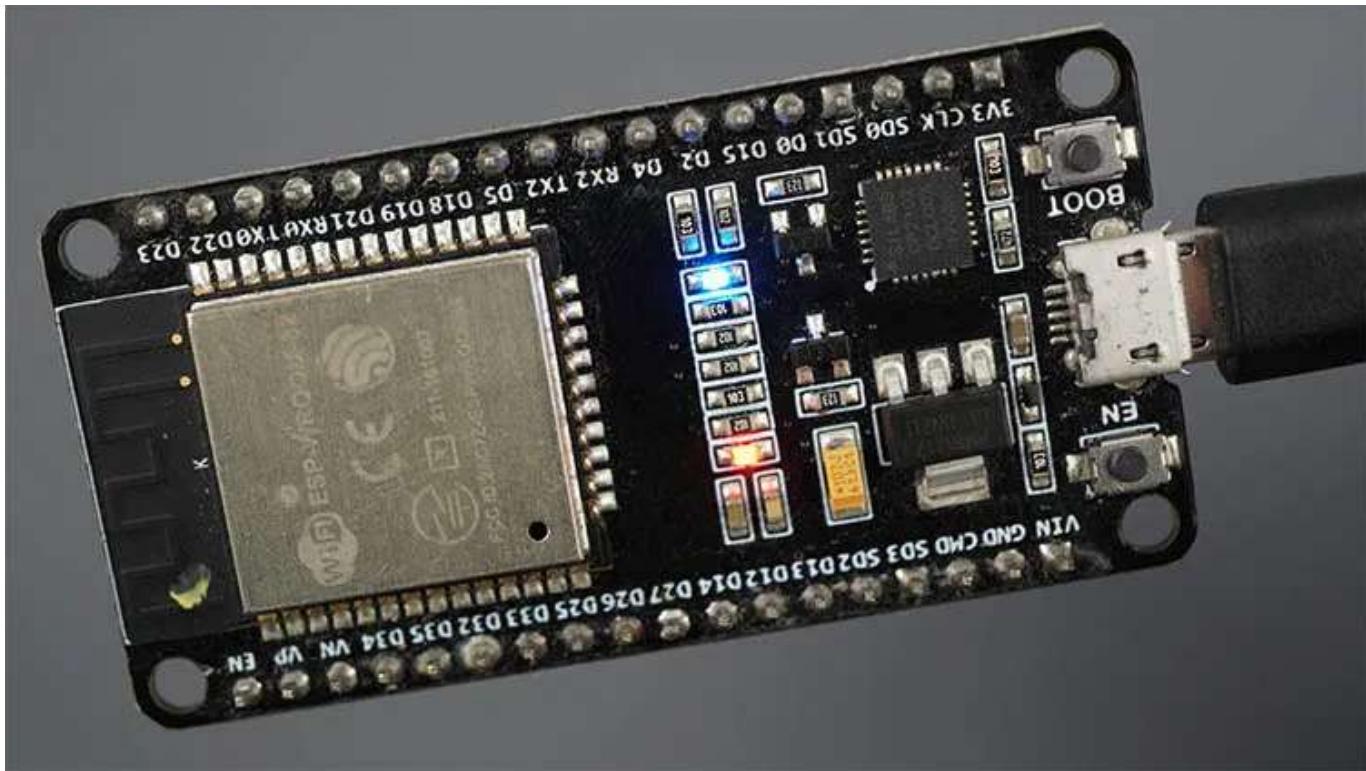
☰ Menu



6. Then, you can go to the root (/) URL to access the new web server. This is the page you should see when you access the ESP IP address on the root ( / ) URL.



You can click on the button to turn the ESP32 on-board LED on and off.



Because we've also added OTA capabilities to this new web server, we can upload a new sketch in the future if needed. You just need to go to the ESP32 IP address followed by `/update`.

Congratulations, you've uploaded new code to your ESP32 via Wi-Fi using AsyncElegantOTA.

Continue reading if you want to learn how to upload files to the ESP32 filesystem (SPIFFS) using AsyncElegantOTA.

# Upload Files to Filesystem OTA (Over-the-Air) Updates – ESP32

In this section you'll learn to upload files to the ESP32 filesystem (SPIFFS) using AsyncElegantOTA.

# ESP32 Filesystem Upload Plugin

Before proceeding, you need to have the ESP32 Uploader Plugin installed in your Arduino IDE. Follow the next tutorial before proceeding:

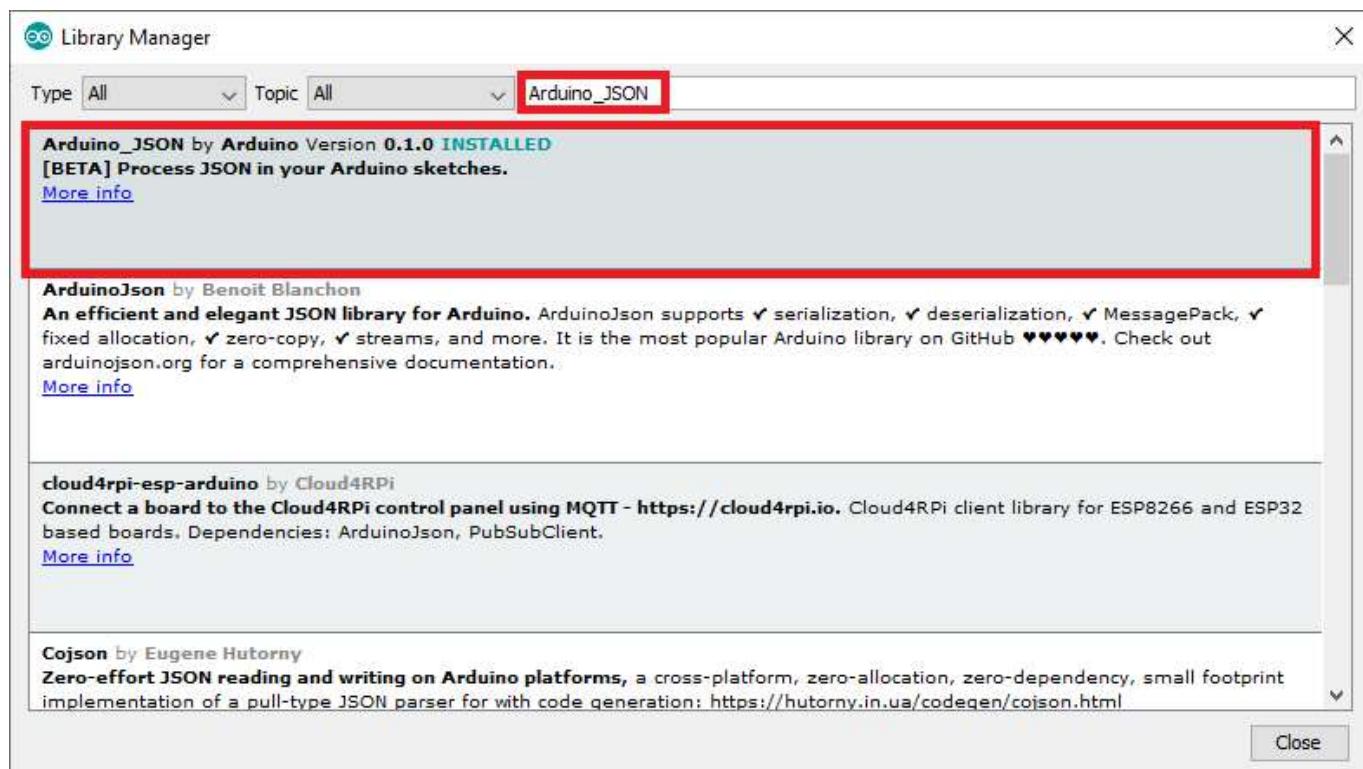
- [Install ESP32 Filesystem Uploader in Arduino IDE](#)

## Web Server with Files from SPIFFS

Imagine the scenario that you need to upload files to the ESP32 filesystem, for example: configuration files; HTML, CSS and JavaScript files to update the web server page; or any other file that you may want to save in SPIFFS via OTA.

To show you how to do this, we'll create a new web server that serves files from SPIFFS: HTML, CSS and JavaScript files to build a web page that controls the ESP32 GPIOs remotely.

Before proceeding make sure you have the [Arduino\\_JSON library by Arduino version 0.1.0](#) installed. You can install this library in the Arduino IDE Library Manager. Just go to **Sketch > Include Library > Manage Libraries** and search for the library name as follows: Arduino\_JSON.



Copy the following code to your Arduino IDE.

## Complete project details

- Arduino IDE: <https://RandomNerdTutorials.com/esp32-ota-over-the-air-arduino/>
- VS Code: <https://RandomNerdTutorials.com/esp32-ota-over-the-air-arduino/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

\*/

```
// Import required libraries
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include "SPIFFS.h"
#include <Arduino_JSON.h>
#include <AsyncElegantOTA.h>

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

[View raw code](#)

Insert your network credentials in the following variables and save the code.

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

Create a `.bin` file from this sketch as shown previously (this sketch includes the needed lines of code to provide OTA capabilities).

Go to the ESP32 IP address followed by `/update` and upload the new firmware.

Next, we'll see how to upload the files.

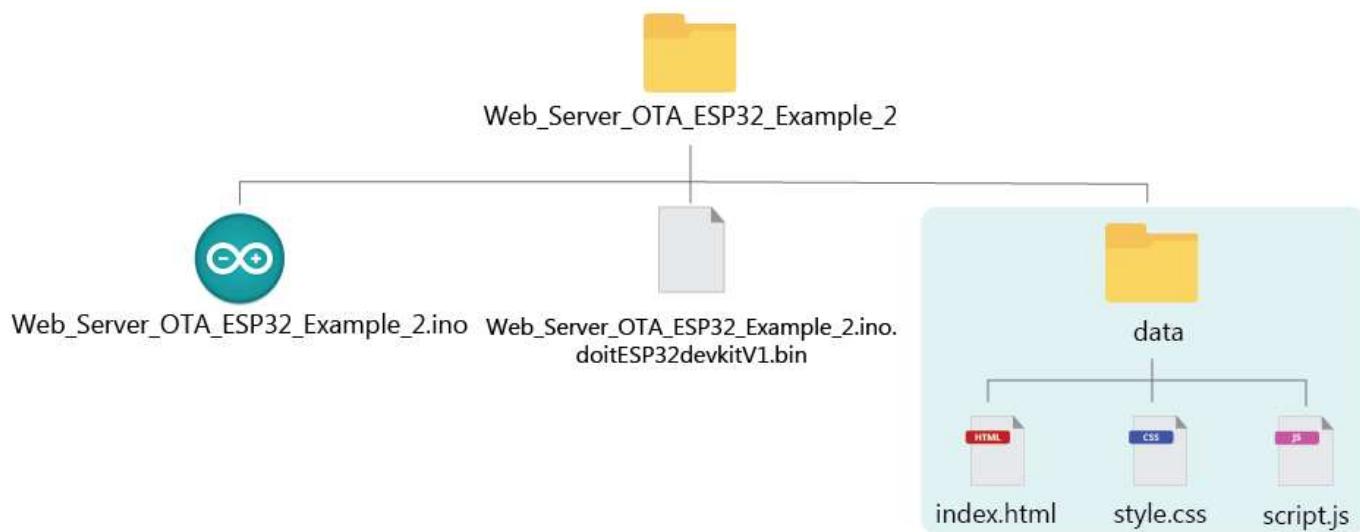
## Update Filesystem

Under the project folder, create a folder called `data` and paste the following HTML, CSS, and JavaScript files (click on the links to download the files):

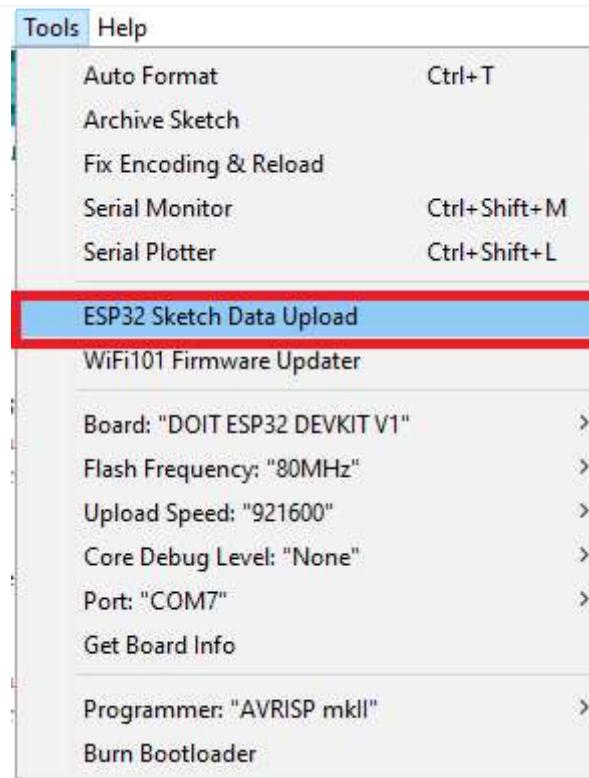
- [HTML file: `index.html`](#)
- [CSS file: `style.css`](#)
- [JavaScript file: `script.js`](#)
- [Download all files](#)

To find your project folder, you can simply go to **Sketch > Show Sketch Folder**.

This is where your data folder should be located and how it looks:



After this, with the ESP32 disconnected from your computer (that's the whole purpose of OTA), click on **ESP32 Data Sketch Upload**.



You'll get an error because there isn't any ESP32 board connected to your computer – don't worry.

Scroll up on the debugging window until you find the `.spiffs.bin` file location. That's that file that you should upload (in our case the file is called `Web_Server_OTA_ESP32_Example_2.spiffs.bin`.

```

// Import required libraries
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include "SPIFFS.h"
#include <Arduino_JSON.h>
#include <AsyncElegantOTA.h>

// Replace with your network credentials

```

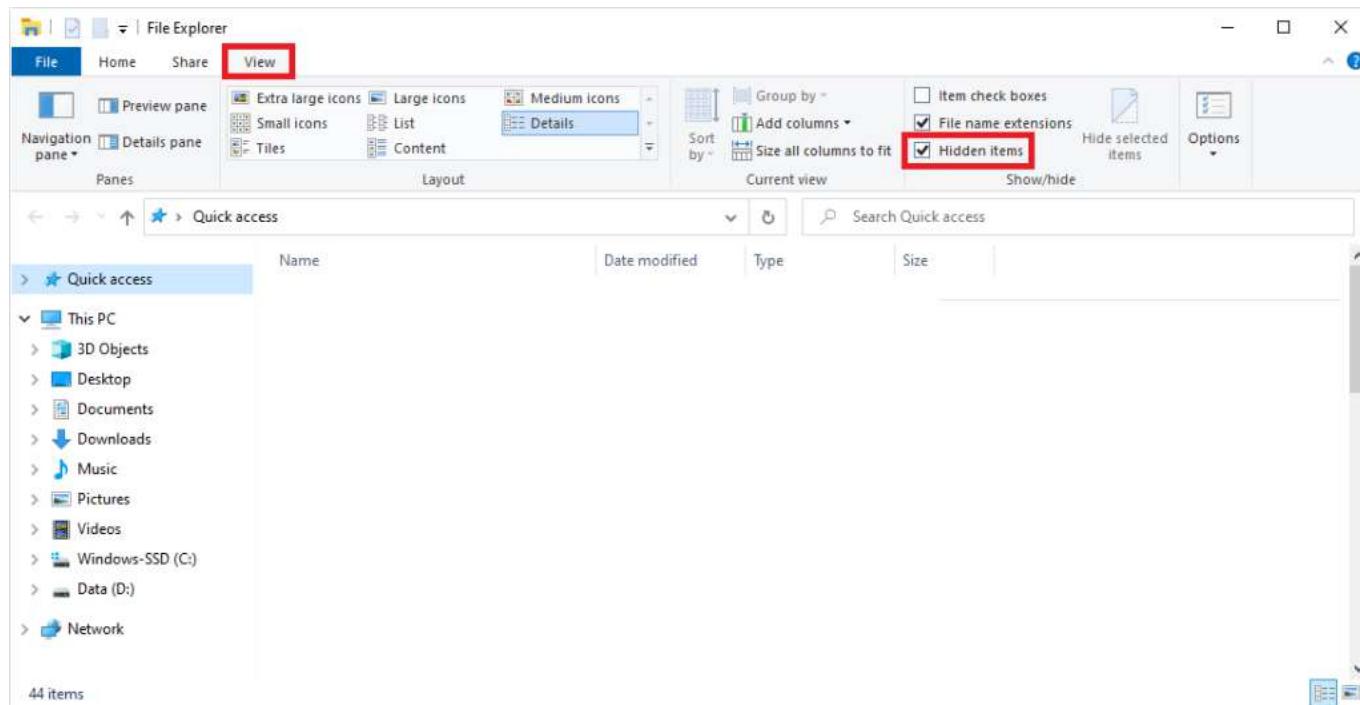
SPIFFS Upload failed!

[SPIFFS] upload : C:\Users\sarin\AppData\Local\Temp\arduino\_build\_675367\Web\_Server\_OTA\_ESP32\_Example\_2.spiffs.bin  
[SPIFFS] address: 2686976  
[SPIFFS] port : COM3  
[SPIFFS] speed : 921600  
[SPIFFS] mode : dio  
[SPIFFS] freq : 80m

And this is the path where our file is located:

C:\Users\sarin\AppData\Local\Temp\arduino\_build\_675367\Web\_server\_OTA\_ESP32\_E  
xample\_2.spiffs.bin

To access that file on my computer, I need to make hidden files visible (the *AppData* folder was not visible). Check if that's also your case.



Once you reach the folder path, you want to get the file with *.spiffs.bin* extension.

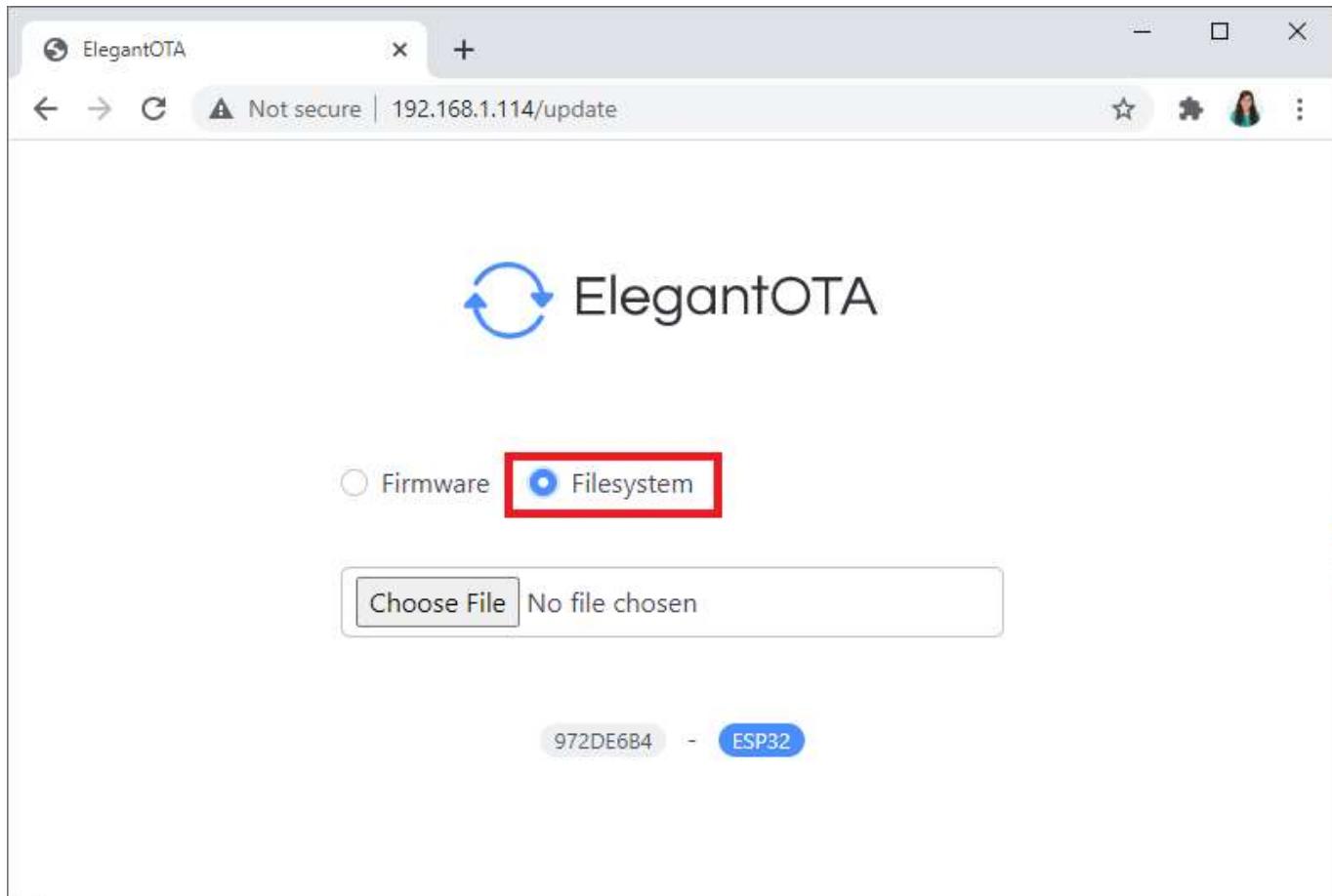
core	1/27/2021 6:26 PM	File folder
libraries	1/27/2021 6:26 PM	File folder
preproc	1/27/2021 6:26 PM	File folder
sketch	1/27/2021 6:26 PM	File folder
build.options.json	1/27/2021 6:25 PM	JSON File
includes.cache	1/27/2021 6:26 PM	CACHE File
Web_Server_OTA_ESP32_Example_2.ino.bin	1/27/2021 6:26 PM	BIN File
Web_Server_OTA_ESP32_Example_2.ino.elf	1/27/2021 6:26 PM	ELF File
Web_Server_OTA_ESP32_Example_2.ino.partitions.bin	1/27/2021 6:26 PM	BIN File
Web_Server_OTA_ESP32_Example_2.spiffs.bin	1/27/2021 6:26 PM	BIN File

To make things easier you can copy that file to your project folder

☰ Menu

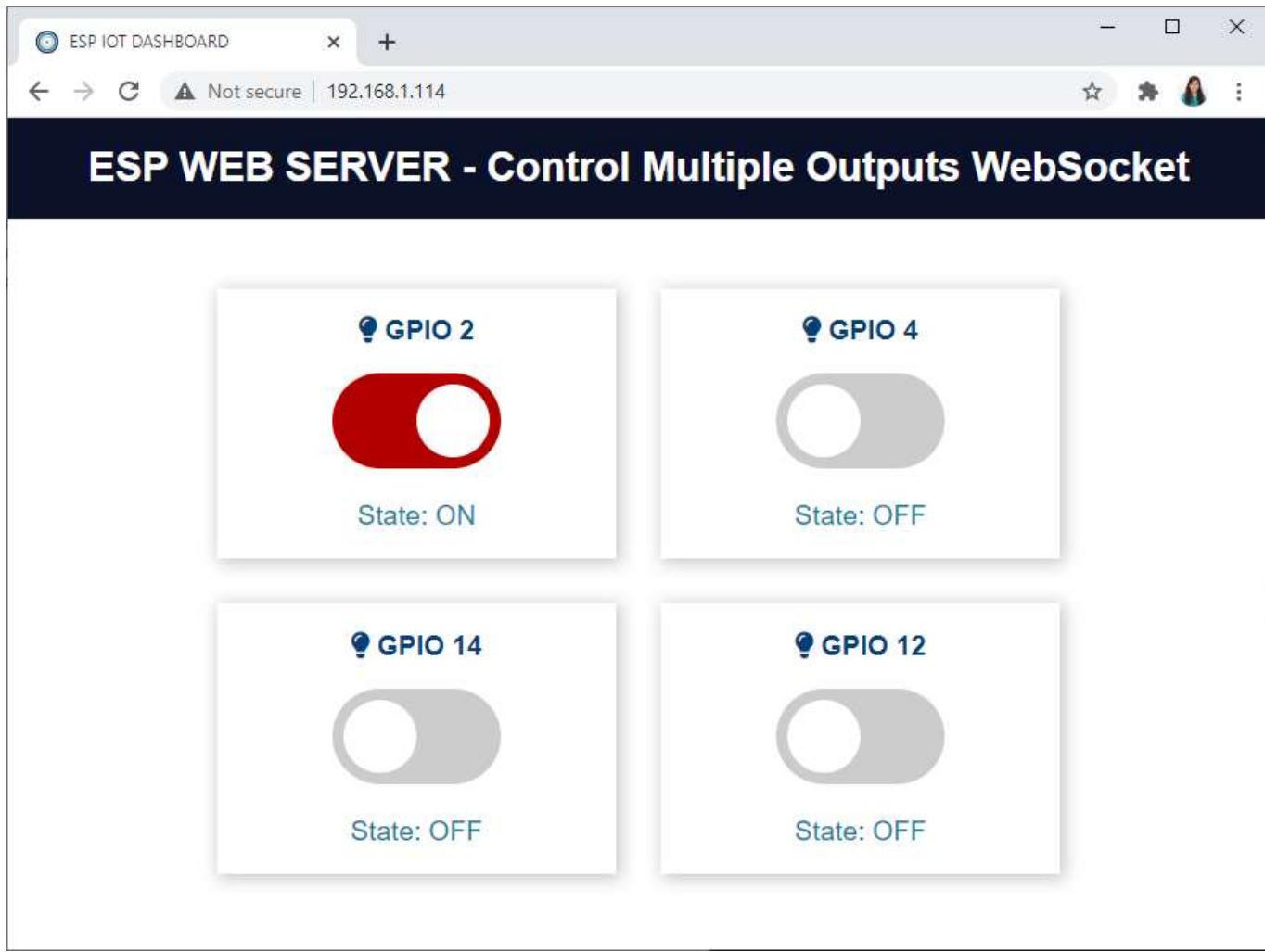


Now that we have a *.bin* file from the *data* folder, we can upload that file. Go to your ESP32 IP address followed by `/update`. Make sure you have the **Filesystem** option selected.



Then, select the file with the *.spiffs.bin* extension.

After successfully uploading, click the **Back** button. And go to the root ( / ) URL again. You should get access to the following web page that controls the ESP32 outputs using [Web Socket protocol](#).



To see the web server working, you can connect 4 LEDs to your ESP32 on GPIOs: 2, 4, 12, and 14. You should be able to control those outputs from the web server.

If you need to update something on your project, you just need to go to your ESP32 IP address followed by /update .

Congratulations! You've successfully uploaded files to the ESP32 filesystem using AsyncElegantOTA.

## Wrapping Up

In this tutorial you've learned how to add OTA capabilities to your Async Web Servers using the [AsyncElegantOTA library](#). This library is super simple to use and allows you to upload new firmware or files to SPIFFS effortlessly using a web page. In our opinion, the AsyncElegantOTA library is one of the best options to handle OTA web

We hope you've found this tutorial useful.

Learn more about the ESP32 with our resources:

- [Build ESP32 Web Servers with Arduino IDE \(eBook\)](#)
- [Learn ESP32 with Arduino IDE](#)
- [More ESP32 Projects and Tutorials...](#)

Thanks for reading.



The advertisement for PCBWay features a green and yellow design. At the top left is the PCBWay logo with the tagline "PCB Fabrication & Assembly". Below it, a large yellow banner proclaims "ONLY \$5 for 10 PCBs". To the right of the banner is a photograph of several printed circuit boards (PCBs) being processed in a manufacturing facility. A URL "wwwpcbway.com" is visible in the background of the photo. On the left side of the banner, there are two bulleted lists: "✓ 24-hour Build Time" and "✓ Quality Guaranteed", followed by "✓ Most Soldermask Colors:" and a color swatch showing ten different colors. A yellow button at the bottom left says "Order now".

### [eBook] Build Web Servers with ESP32 and ESP8266 (2nd Edition)

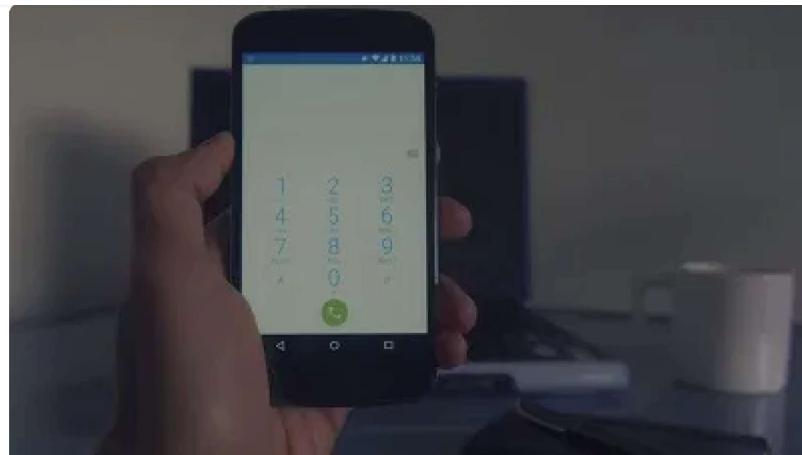


Build Web Server projects with the ESP32 and ESP8266 boards to control outputs and monitor sensors remotely. Learn HTML, CSS, JavaScript and client-server communication protocols [DOWNLOAD »](#)

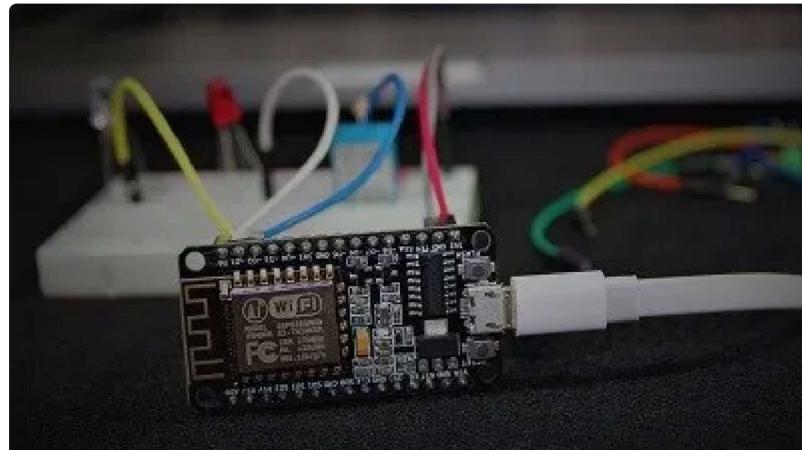
## Recommended Resources

≡ Menu

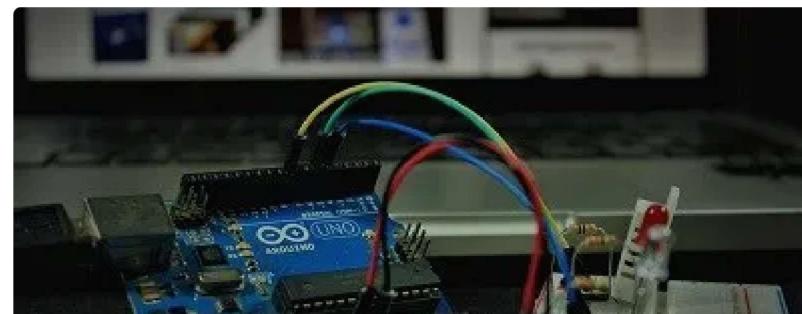




[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino, and Node-RED.



[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...

[ESP32 Data Logging Temperature to MicroSD Card](#)

---

## ESP32 Web Server Hosting Files from MicroSD Card

**Enjoyed this project? Stay updated by subscribing our newsletter!**

SUBSCRIBE

## 100 thoughts on “ESP32 OTA (Over-the-Air) Updates – AsyncElegantOTA using Arduino IDE”



**Scott jacobs**

March 9, 2021 at 12:58 pm

SCORE! Well done.....

**Bruce Calder**

March 9, 2021 at 4:40 pm

This is great. I have been (trying) to use OTA for years and very rarely has it worked. This seems to work every time (sort of) so that's a plus. The only thing odd is that after loading the "Web Server Sketch – Example" it will not connect to the Wifi. I see "Connecting to WiFi.." repeating but it never connects unless I reboot/reset the ESP32, then it connects every other time. In other words, after uploading the code it doesn't work until I reboot the ESP32. Then it connects. If I reboot it again, it doesn't connect. If I reboot it again, it works. Very odd behavior. Any thoughts.

[Reply](#)**Sara Santos**

March 9, 2021 at 5:10 pm

Hi Bruce.

That's indeed a very odd behavior.

We've experimented with the library with different examples, and it never failed.

We also tried it with the ESP8266, and everything went fine.

Do you have another board to experiment with?

Regards,

Sara

[Reply](#)

Menu



**Ed**

March 11, 2021 at 3:31 am

experimenting too. currently I have a program (on an ESP8266 using the proper ESP8266 code) with both STA and AP,Async webserver that connects with static IP and the elegantOTA does not seem to want to start. Have not figured out why yet. disabling the AP makes no difference. Will try to find it (the example runs fine)

[Reply](#)**Ed**

March 11, 2021 at 5:06 pm

OK I found the problem. As you can see, most of the examples, make their WiFi connection, then issue a 'server.on("/.....' request, and then do their elegant ota and server.begin() requests, placing it at the end of setup().

That is what I did.....but I had a couple more server requests,,,one of them called 'update'.

need i say more 😊

[Reply](#)**Juerg**

March 9, 2021 at 6:08 pm

[☰ Menu](#)

Hi Sara

An excellent description – as usual!

I'm wondering if this works with LittleFS too as the SPIFF file system seems to be deprecated for the ESP32. Have you tried it and is it possible to make it work in a similar way?

Regards, Juerg

[Reply](#)



**Sara Santos**

March 9, 2021 at 10:18 pm

Hi.

SPIFFS is only deprecated for the ESP8266.

This tutorial also works with littlefs (at least with the ESP8266).

We'll publish a similar tutorial for the ESP8266 by the end of this week (with littlefs).

Regards,

Sara

[Reply](#)



**Juerg**

March 14, 2021 at 12:30 pm

Hi Sara

I have just tested the OTA library with an ESP32 with LITTLEFS: Indeed everything works fine and smooth, even with a username and password for the OTA website (IoT security). The ESP32 just had to be rebooted

Menu



itself.

Best Regards, Juerg

[Reply](#)



**Michel**

March 9, 2021 at 8:28 pm

Wow, Very good job

[Reply](#)



**Ed**

March 10, 2021 at 12:16 pm

Great,I have been trying this with the ESP8266 and works very well.  
Never had real trouble with the regular OTA, including webOTA, but had the occasional odd behaviour. Main advantage here is that it can be used with Async webpages and it looks a lot better.

Very well explained

[Reply](#)

Great example. I will definitely use this.

I am a little concerned that just about anyone can corrupt your esp32 server by uploading a new (possibly a virus) program to it. There seems to be no security and a very generic “update” catch phrase to access it.

[Reply](#)



**Stefan**

March 10, 2021 at 8:38 pm

I clicked on view raw of the second example

link [https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/raw/master/Projects/ESP32/AsyncElegantOTA/ESP32\\_Web\\_Server\\_LED\\_OTA/ESP32\\_Web\\_Server\\_LED\\_OTA.ino](https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/raw/master/Projects/ESP32/AsyncElegantOTA/ESP32_Web_Server_LED_OTA/ESP32_Web_Server_LED_OTA.ino)

and did a <ctrl-a, ctrl-c ctrl-v

then I tried to compile the example but I get a lot of error-messages

Do you RE-test EVERYTHING by following your tutorial?

I guess not otherwise you would have recognized the bugs yourself

exit status 1

control reaches end of non-void function [-Werror=return-type]

[Reply](#)



**Sara Santos**

☰ Menu



Hi.

Can you provide more details about the error?

I've just compiled the code again and it was fine.

Regards,

Sara

[Reply](#)



**Stefan Ludwig**

March 10, 2021 at 9:58 pm

Hi Sara,

thank you for answering so quick. Did you do a Copy & paste from the website and then compile it?

The I have activated verbose output so the complete output is more than the forum-software allows

I guess this part is important

```
C:\Users\Stefan\Documents\Arduino\Web_Server_LED_OTA_ESP32\Web_Server_LED_OTA_ESP32.ino: In function 'String processor(const String&):'
```

```
Web_Server_LED_OTA_ESP32:202:1: error: control reaches end of non-void  
function [-Werror=return-type]
```

```
}
```

```
cc1plus.exe: some warnings being treated as errors
```

```
Multiple libraries were found for "WiFi.h"
```

Used: P:\Portable-

```
Apps\arduino1.8.13\portable\packages\esp32\hardware\esp32\1.0.4\libraries\  
WiFi
```

Using library WiFi at version 1.0 in folder: P:\Portable-  
Apps\arduino1.8.13\portable\packages\esp32\hardware\esp32\1.0.4\libraries\  
WiFi

Using library AsyncTCP at version 1.1.1 in folder:  
C:\Users\Stefan\Documents\Arduino\libraries\AsyncTCP

Using library ESPAsyncWebServer at version 1.2.3 in folder:  
C:\Users\Stefan\Documents\Arduino\libraries\ESPAsyncWebServer

Using library FS at version 1.0 in folder: P:\Portable-  
Apps\arduino1.8.13\portable\packages\esp32\hardware\esp32\1.0.4\libraries\  
FS

Using library AsyncElegantOTA at version 2.2.5 in folder:  
C:\Users\Stefan\Documents\Arduino\libraries\AsyncElegantOTA

Using library Update at version 1.0 in folder: P:\Portable-  
Apps\arduino1.8.13\portable\packages\esp32\hardware\esp32\1.0.4\libraries\  
Update

exit status 1

control reaches end of non-void function [-Werror=return-type]

best regards Stefan

[Reply](#)



Sara Santos

March 10, 2021 at 10:10 pm

Yes, I did that and it compiles fine for me.

What's the Arduino IDE and esp32 boards version that you have?

Modify your processor() function to be like this:

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

Let me know if this solves your issue.

Regards,

Sara

[Reply](#)



**Stefan Ludwig**

March 11, 2021 at 7:24 am

I did modify the first code and tested the OTA with this modified first code-version.

The one that just says Hi! I am ESP32.

This worked

When I inserted your version of String processor  
there was no formatting. no indentions so I pressed ctrl-t for autoformatting.  
but nothing changed.

then I moved the function String processor above the html-code and voila  
there autoformatting worked.

So my conclusion is that something is written “non-international” inside the  
rawliteral.

Something inside the raw-literal has a syntactical error. No idea what this  
could be as I have never worked with html-code.

Now idea if – when I post the html-code section here that all the character-  
translations done by the forum-software keeps the error

anyway I post it here

```
// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");

const char index_html[] PROGMEM = R"rawliteral(
```

ESP Web Server

```
html {
  font-family: Arial, Helvetica, sans-serif;
  text-align: center;
}
h1 {
  font-size: 1.8rem;
  color: white;
}
h2{
  font-size: 1.5rem;
```

```
color: #143642;  
}  
.topnav {  
overflow: hidden;  
background-color: #143642;  
}  
body {  
margin: 0;  
}  
.content {  
padding: 30px;  
max-width: 600px;  
margin: 0 auto;  
}  
.card {  
background-color: #F8F7F9;;  
box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5);  
padding-top:10px;  
padding-bottom:20px;  
}  
.button {  
padding: 15px 50px;  
font-size: 24px;  
text-align: center;  
outline: none;  
color: #fff;  
background-color: #0f8b8d;  
border: none;  
border-radius: 5px;  
-webkit-touch-callout: none;  
-webkit-user-select: none;  
-khtml-user-select: none;  
-moz-user-select: none;  
-ms-user-select: none;
```

```
}
```

.button:hover {background-color: #0f8b8d;}

.button:active {

background-color: #0f8b8d;

box-shadow: 2 2px #CDCDCD;

transform: translateY(2px);

}

.state {

font-size: 1.5rem;

color:#8c8c8c;

font-weight: bold;

}

## ESP Web Server

### ESP WebSocket Server

Output – GPIO 2

state: %STATE%

Toggle

```
var gateway = `ws://${window.location.hostname}/ws`;
var websocket;
window.addEventListener('load', onLoad);
function initWebSocket() {
  console.log('Trying to open a WebSocket connection...');
  websocket = new WebSocket(gateway);
  websocket.onopen = onOpen;
  websocket.onclose = onClose;
  websocket.onmessage = onMessage; // <- add this line
}
function onOpen(event) {
  console.log('Connection opened');
}
```

```
setTimeout(initWebSocket, 2000);
}

function onMessage(event) {
var state;
if (event.data == "1"){
state = "ON";
}
else{
state = "OFF";
}
document.getElementById('state').innerHTML = state;
}

function onLoad(event) {
initWebSocket();
initButton();
}
function initButton() {
document.getElementById('button').addEventListener('click', toggle);
}
function toggle(){
websocket.send('toggle');
}

)rawliteral”;

void notifyClients() {
ws.textAll(String(ledState));
}
best regards Stefan
```

[Reply](#)

**Stefan Ludwig**

March 11, 2021 at 8:17 am

Posting html-code in this commenting-software SUCKS,  
The comment-software eats half of the code

to be honest: Sara and Rui you should consider using a completely  
DIFFERENT commenting-software. All this is about programming. Which  
means SOURCECODE is a very important part.

So the Commenting-part should be able to show Sourcecode as Sourcecode  
with a FIXED-DISTANCE-font and as CODE-SECTIONS like ANY other  
programming-user-forum does.

I did some reading about PROGMEN and rawliterals and found the  
syntactical error.

Then I did do the following:

I took my mouse holding down the left mousebutton to start marking the  
sourcecode of the seconds sourcecode (the one that is called  
`Web_Server_LED_OTA_ESP32`)

I finished the marking of the source-code by keeping the shift-key-pressed  
using cursor down to mark all characters that belong to the sourcecode.

Then I pasted this into the Arduino-IDE. As a cross-checking I pasted it into  
UltraEdit, Notepad++, and standard-notepad.

Always the same result:

At the end of the html-code there is

`)rawliteral";`

the closing bracket of the command rawliteral is one line below the closing “edgy” bracket “>” of the html-tag

it is the same inside the “RAW-Code”-page

as soon as I removed the CR so the source-code looks like this  
)  
rawliteral”;

the closing-tag and closing bracket and rawliteral all in the SAME line

the code compiled.

This is why I highly doubt that you did do an EXACT copy and paste into Arduino-IDE without any additional hand-formatting

How would it be possible that your Arduino-IDE could remove a CR (a carriage-return) while my version and any other texteditor does not?

I developed the habit of testing / repeating ALL steps (if I say ALL steps I mean ALL steps!!)

before uploading code into a user-forum. Even after changing a single character.

Which means for example I upload a code-example to the Arduino-forum as a code-section. I use the select all link to copy the code into the clipboard I paste this clipboard-content into an absolutely empty new opened sketch and do a REAL upload into the microcontroller and testing the code and watch the serial output does the code behave like expected?

So this means I do ALL steps another user will do if he tries my code.

Maybe you have tested it with PlatForm-IO or did remove the CR because you thought it was a type by you

There is nothing more frustrating for newbees as if a tutorial pretends to

best regards Stefan

[Reply](#)



**Rui Santos**

March 11, 2021 at 11:36 am

I've also tested the exact code on my computer right now and it works fine for me. I honestly don't know what's missing either.

I've copied the exact code from our website again to both Arduino IDE and VS Code and both compiled just fine.

[Reply](#)



**Stefan Ludwig**

March 11, 2021 at 8:20 am

again this forumsoftware eat up the html-code so I add it modified by inserting underlines between each character of the html-code-part  
your codeversion looks like this

```
</_h_t_m_l>_  
rawliteral";
```

it should be in ONE line

```
</_h_t_m_l>_)rawliteral";
```

best regards Stefan

☰ Menu



**Sara Santos**

March 11, 2021 at 10:56 am

Hi Stefan.

I'm sorry for the issue.

But I did test the code, as I told you.

I copied the raw code, paste it into Arduino IDE, and recompiled it. It went fine, as you can see here: <https://imgur.com/ylc6n11>.

This is all very weird, and I can't find any explanation about this. What's the version of Arduino IDE that you are using? I'm using 1.8.13.

All I want is that the tutorials work for everyone straight away.

I'll try to investigate a bit more about this issue.

Regards,

Sara

[Reply](#)

**Stefan Ludwig**

March 11, 2021 at 8:23 pm

I tried it with Arduino Ide 1.8.12 and 1.8.13 both showed the same problem.  
Then I tried it with Arduino-IDE 1.8.13 on another computer same problem.

the combination of

– changing the code of function “processor” to return String()

with

– putting the html-tag and the keyword rawliteral in the same line of code

Menu



Beside this weird syntax-pickiness this OTA-functionality is great.

Thank you very much for providing this.

What really astonished me was the fact that – if I call the ESP32's website with the LED-toggle-button from multiples computers that on each computer the written state of the LED gets updated the same tenth second I click on the button.

No reloading of the website requiered. That is really great!

Do you happen to know a WYSIWYG-website designer that makes it easier to create the HTML-code?

Or do you have a tutorial that shows how the HTML-elements like buttons, sliders etc. are programmed?

I mean giving an example and explaining the details through variations:

positioning the button

button-size

button color

button-text

how to evaluate the button-click

how to change the button text/color on run-time

etc.

best regards Stefan

[Reply](#)



**Sara Santos**

March 12, 2021 at 12:18 am

Hi.

≡ Menu



That's very strange behavior. Rui also tested the code, and it went fine. What operating system do you use? We are on Windows.

Yes, the synchronization of all clients is a great feature, thanks to the WebSocket protocol.

I'm not familiar with software like that. We design our own webpages. In our latest course, "Build Web Servers with ESP32 and ESP8266," we go into more detail about writing the HTML and CSS for your web pages, handling the HTML elements, and how to use JavaScript.

Whenever I have a doubt about HTML elements and how to style them, I usually refer to the w3schools website.

Regards,  
Sara

[Reply](#)



**Sara Santos**

March 12, 2021 at 5:14 pm

We updated the code with your suggestions.

Regards,  
Sara

[Reply](#)



**Stefan Ludwig**

March 12, 2021 at 5:42 pm

Hi Sara,

thank you very much.

Now a simple copy & paste compiles.

From both “sources” the Website-section (with the colored code) and the raw code (just text)

OK I’m gonna take a look into your Build Web Servers with ESP32-course

best regards Stefan

[Reply](#)



**Eric Carter**

March 17, 2021 at 8:24 pm

Really great tutorial – excellent level of clarity.

Where I cam unstuck was my Arduino IDE was an Ubuntu SNAP package and for the life of me I could not find the ...SPIFFS.bin file. Solved by installing the IDE from the web site but then had to solve all the python2 – python3 problems. Anyway working really well so now I need to think of a use for it!

[Reply](#)



**Robert Fujita**

March 18, 2021 at 8:24 pm

Menu



H Sara,

I get the following error message when attempting to run the Arduino IDE:

Arduino: 1.8.13 (Mac OS X), Board: “Adafruit ESP32 Feather, 80MHz, 921600, None, Default”

Traceback (most recent call last):

File “esptool.py”, line 57, in

File “/Library/Python/2.7/site-

packages/PyInstaller/loader/pyimod03\_importers.py”, line 389, in

load\_module

File “serial/tools/list\_ports.py”, line 29, in

File “/Library/Python/2.7/site-

packages/PyInstaller/loader/pyimod03\_importers.py”, line 389, in

load\_module

File “serial/tools/list\_ports\_posix.py”, line 31, in

File “/Library/Python/2.7/site-

packages/PyInstaller/loader/pyimod03\_importers.py”, line 389, in

load\_module

File “serial/tools/list\_ports\_osx.py”, line 32, in

ValueError: dlsym(RTLD\_DEFAULT, kIOMasterPortDefault): symbol not

found

Failed to execute script esptool

Multiple libraries were found for “WiFi.h”

Used:

/Users/TinkersHome/Library/Arduino15/packages/esp32/hardware/esp32/1.0.4/libraries/WiFi

Not used:

/private/var/folders/5y/zlr8vhg579vbb3glg\_x9dfsr0000gn/T/AppTranslocation/1E1F157B-C1FB-4E6D-B760-

D8182C3AD58B/d/Arduino.app/Contents/Java/libraries/WiFi

exit status 255

/private/var/folders/5y/zlr8vhg579vbb3glg\_x9dfsr0000gn/T/AppTranslocation/

D8182C3AD58B/d/Arduino.app/Contents/Java/arduino-builder returned 255  
Error compiling for board Adafruit ESP32 Feather.

Would you please tell me what my problem is? Thanks

[Reply](#)



**Steve Mueller**

March 18, 2021 at 10:14 pm

I downloaded esp32 and esp 8266 but it didn't go into Arduino IDE library. It went somewhere .when I re download

It said its already been installed.I have windows 10. What can be done?

[Reply](#)



**Stefan Ludwig**

March 19, 2021 at 5:56 pm

"I downloaded ESP32 and ESP8266" is a pretty vague description of what you have

might done. For programming ESP32/8266 you need to add two additional board-definition url into the preferences. So the minimum is that you describe in detail what you have really have done. And I guess you tried to compile he code. You have to activate verbose output and then analyse the output for the errors that were found during compilation and post the errors here.

Menu



[Reply](#)**Andre**

March 18, 2021 at 10:46 pm

When I tried adding this to an existing project and go to <http://ip/update>, all that shows is the word “OK” i.e. no upload box or any other text.

[Reply](#)**Andre Basel**

March 19, 2021 at 12:52 am

Please ignore the above... I had not put some files in the correct place

[Reply](#)**Andre Basel**

March 19, 2021 at 1:05 am

Will this work with an ESP-01

[Reply](#)

Menu



**Andre**

March 19, 2021 at 1:15 am

I put the index and cc files in the correct place but still no joy. The serial monitor reports, “14:14:06.077 -> No message sent” and “ok” appears in the browser.

[Reply](#)**Andre**

March 22, 2021 at 11:06 pm

I worked it out. I was already using /update for something else.

[Reply](#)**Bob**

March 20, 2021 at 11:32 am

Another excellent tutorial – many thanks

Can I suggest changing:

```
request->send(200, "text/plain", "Hi! I am ESP32.");
```

to:

```
request->send(200, "text/plain", FILE "\nCompiled: " DATE " " TIME);
```

This will return the path and INFO name currently loaded IDE and when it was

[☰ Menu](#)

Also... recommend using fixed IP address otherwise you need to connect to USB to find IP address which defeats the use of OTA

[Reply](#)



**Sara Santos**

March 23, 2021 at 11:23 am

Thanks for the suggestions.

Regards,

Sara

[Reply](#)



**Martin Keller**

March 20, 2021 at 2:46 pm

Perfect! It works with ESP32 as AP too, no external scripts needed.

Thanx for the tip

[Reply](#)



**Andre**

March 22, 2021 at 11:07 pm

How does one add a logon to the upload page so that not anyone can upload firmware.

[Reply](#)



**Stefan Ludwig**

March 23, 2021 at 5:45 am

If you encounter a problem or if you want a new feature do a 5 minute-research.

5 minutes is almost nothing and in a part of all cases you have success in just 5 minutes.

I developed this habit. For googling and for coding.

So I took a look into the file AsyncElegantOTA.h  
and VOILÀ:

you can find there two lines for setting up a username and a password

I haven't tested this myself yet.

best regards Stefan

[Reply](#)



**Andre**

March 23, 2021 at 7:57 am

I did indeed google but had no joy. Looking at the Code I thought that the credentials referred to the Wifi Password... my coding is a tad rusty 😊

I will do a bit more digging in the \*.h file.

Thanks again.

[Reply](#)



**Andre**

March 23, 2021 at 8:21 am

Ok so this works i.e. adding credentials to this line, “void begin(AsyncWebServer server, *const char* username = “username”, *const char\** password = “password”)

I tried setting them where they are declared in at the bottom of the file but that did not work.

I just need to now work out how to log off as once you have logged on it appears to keep you logged in.

[Reply](#)



**Thalis Mazzarino**

March 29, 2021 at 10:59 am

Hello, very good article. A question would be possible, say on any device with

☰ Menu



network and password on esp32 through the use of the web server page, when so desired?

[Reply](#)



**Rodrigo Brasil**

March 30, 2021 at 5:33 pm

Hello Sara and Rui.

I'm trying to upload my .bin file into ESP32CAM through OTA, but is occurring a problem...The web page shows up well and I can choose the .bin file, however after show up 100%, doesn't appear the message OTA Succes and the button BACK. The web page stay on 100% message always.  
After that the ESP32 CAM reboot and run the old firmware.  
Can you help me with this issue?

[Reply](#)



**Fritz**

April 3, 2021 at 8:38 am

Very neat solution! A little typo: You explain that the update interface is started by:

Now, you need to upload that file using the ElegantOTA page. Go to your ESP IP address followed by /updated.

≡ Menu



Thanks and regards

[Reply](#)



**Sara Santos**

April 3, 2021 at 8:57 pm

Hi.

Yes, you are right.

Thanks for noticing.

It is fixed now.

Regards,

Sara

[Reply](#)



**Jayant Panhalkar**

April 21, 2021 at 9:49 am

Hi Sara,

Thank you for sharing the awesome technology. Excellent code base.

I just have 2 queries

Can we update firmware and files from intermate, Like if I kept both .bin file on my google drive, every power on Node MCU will check if files are updated and update itself with new .bin files of firmware and other files(html, java, css, image)

I need to work on some data in csv file, how I can get the csv file from my

≡ Menu



[Reply](#)**Sara Santos**

April 21, 2021 at 10:14 am

Hi.

Thanks for reaching out.

I think that should be possible, but I don't have any tutorials about that subject.

Regards,

Sara

[Reply](#)**Dave Z**

April 29, 2021 at 11:27 pm

I am trying to use AsyncElegantOTA with an existing sketch, but am getting a bunch of "Multiple definition" errors.

I have the sketch set up to allow me to switch Access points, so, I am also including <WebServer.h>. No matter what I tried, I could not get away from multiple errors, so, I loaded your sketch in a new project, It compiles fine.

But, as soon as I add

```
#include <WebServer.h>
```

to you sketch, I get a host of errors, leading me to believe WebServer.h and AsyncElegantOTA.h are causing problems for each other.

Is there a way to include both?

Is

[☰ Menu](#)

```
sketch\menu.cpp.o:(.bss.AsyncElegantOTA+0x0): multiple definition of
AsyncElegantOTA'
sketch\AsyncIOTWebServer.ino.cpp.o:(.bss.AsyncElegantOTA+0x0): first
defined here
sketch\pin_manager.cpp.o:(.bss.AsyncElegantOTA+0x0): multiple definition
ofAsyncElegantOTA'
sketch\AsyncIOTWebServer.ino.cpp.o:(.bss.AsyncElegantOTA+0x0): first
defined here
sketch\system.cpp.o:(.bss.AsyncElegantOTA+0x0): multiple definition of
AsyncElegantOTA'
sketch\AsyncIOTWebServer.ino.cpp.o:(.bss.AsyncElegantOTA+0x0): first
defined here
sketch\temperatures.cpp.o:(.bss.AsyncElegantOTA+0x0): multiple definition
ofAsyncElegantOTA'
sketch\AsyncIOTWebServer.ino.cpp.o:(.bss.AsyncElegantOTA+0x0): first
defined here
sketch\wifi_code.cpp.o:(.bss.AsyncElegantOTA+0x0): multiple definition of
`AsyncElegantOTA'
sketch\AsyncIOTWebServer.ino.cpp.o:(.bss.AsyncElegantOTA+0x0): first
defined here
collect2.exe: error: ld returned 1 exit status
exit status 1
Error compiling for board ESP32 Dev Module.
```

[Reply](#)



Sara Santos

May 1, 2021 at 11:11 am

Hi.

If you're using the WebServer.h library, it is better to follow this tutorial  
instead: <https://randomnerdtutorials.com/esp32-over-the-air-ota/>

≡ Menu



Regards,  
Sara

[Reply](#)



**Dave**

May 3, 2021 at 4:28 pm

Thanks for the answer Sara, will this also be compatible with  
<ESPAsyncWebServer.h>?

It may seem as if I cannot make up my mind what I want to use, but, I am only using “WebServer.h” to add new WiFi credentials if I am not within range of a currently known AP, but the project runs on <ESPAsyncWebServer.h> once an AP has been established.

[Reply](#)



**Sara Santos**

May 3, 2021 at 4:57 pm

Hi.

I'm not sure if using both libraries at the same time conflict.  
This OTA procedure in this tutorial is compatible with the  
ESPAsyncWebServer. I haven't tested if it is compatible with WebServer.h.  
You have to try it and see what happens.  
We have this tutorial about Wi-Fi functions with the ESP32 that might be  
useful for your projects: <https://randomnerdtutorials.com/esp32-useful-wi-fi/>

≡ Menu



Regards,  
Sara

[Reply](#)



**Dave Z**

May 3, 2021 at 5:09 pm

OK, Thanks, I will try it, I have already been round and round with so many different ways of trying to accomplish this, one more attempt it is. You & Rui have been an excellent source of quality information!

Thanks!

[Reply](#)



**Christo**

May 10, 2021 at 7:24 am

Hi, nice tutorial and just what i need. I consider myself a novice where esp32 is concerned. I have just 1 question, i need to upload temperature logs to a server hosted outside of my network. So, i will do an http get request with temp values. Can i use the OTA library as well as do the above mentioned tasks. Temp logs get uploaded every 5 minutes

[Reply](#)

**Stefan**

May 11, 2021 at 8:42 am

OTA means a compiled binary file that contains a PROGRAM will be transferred over WiFi into a reserved part of the ESP32-flash-memory. After storing the new compiled program into the OTA-area successfully, this compiled program is transferred into the regular program-flash-memory.

compiled binary PROGRAM——>——any Computer——>WiFi——>ESP32——Flash-PROGRAM-memory

That is a completely different thing than  
ESP32—temperature-Data-in-RAM-Memory——>——WiFi——>——external Server

different direction: Computer——>—ESP32 versus ESP32——>—Server

different data location: flash / versus RAM

different data nature: PROGRAM versus TemperatureDATA

there could not much more be different between those two tasks

[Reply](#)**Jan**

May 27, 2021 at 4:41 pm

Hi Sara,

Menu



Reloading the web page gives me the update page again with Browse again.  
Do I need to change something for ESP32-CAM?

[Reply](#)



**KAUSHAL BHUVA**

July 18, 2021 at 5:29 am

Same problem for me. I think, ESP32-CAM has a special sequence of putting to boot mode by pulling GPIO0 to GND. I too am stuck with same issue. @Sara and Rui, can you suggest us, what to be done?

[Reply](#)



**Sara Santos**

July 18, 2021 at 2:08 pm

Hi.

I think it is better to post an issue in the library gihutb page:

<https://github.com/ayushsharma82/ElegantOTA/issues>

Regards,

Sara

[Reply](#)



**Tuan Anh**

June 2 2021 at 3:40 am

≡ Menu



I had problems when update firmware “abort() was called at pc 0x40143f45 on core 1”. ESP32 immediately reset. What do I need to do to make it work properly?

[Reply](#)



**Stefan**

December 5, 2021 at 6:36 pm

one possible reason for aborted updates is a wrong partition scheme  
I'm using 4MB 1MB FS OTA ~1019KB  
This means your ESP has to have a 4MB flash-chip  
best regards Stefan

[Reply](#)



**randytsuch**

August 6, 2021 at 3:56 pm

Hi

Thanks a lot for your tutorials, I have ElegantOTA working on my ESP32 project now.

Wondering if there is a way to write the filesystem to SD instead of SPIFFS?

I have a webserver running with files on a micro SD card (working thanks to your other tutorials), and would like to make OTA updates to the html files.

≡ Menu



[Reply](#)**randytsuch**

September 6, 2021 at 2:18 pm

FWIW, I gave up trying to save files to a micro sd card using elegant OTA.

I changed my code to use store and read the html files to/from SPIFFS. This is working very well, its now really easy to make changes to the webpage, and then load them with elegantOTA.

Still using the uSD card to store files, no problem using SPIFFS and uSD at the same time.

[Reply](#)**Aage Rasmussen**

September 5, 2021 at 8:53 pm

Hi Sara

I get error for ws.cleanupClient();

AsyscWebSocket has no member named cleanupClients

I'm using a Mac an Arduini 1.8.15

[Reply](#)

Menu



**Sara Santos**

September 6, 2021 at 9:13 am

Hi.

Did you install the EspAsyncWebServer library as mentioned in the tutorial?

Regards,

Sara

[Reply](#)**Sara Santos**

September 7, 2021 at 10:18 am

Answer here: <https://rntlab.com/question/asyncota/>

[Reply](#)**Phil G**

September 21, 2021 at 2:23 pm

I have followed along with the tutorial and updating firmware is working very well except that the progress bar always reads 100%. Then I tried the update file SPIFFS example and I'm stuck. After disconnecting the ESP32 and clicking Tools/ESP32 Sketch Data Upload it fails as expected but the Debug window is blank. I have searched for .spiffs.bin but haven't found any files. I

Menu



[Reply](#)**Stefan**

December 5, 2021 at 6:33 pm

you should open an issue on Github about this

<https://github.com/ayushsharma82/ElegantOTA/issues>

best regards Stefan

[Reply](#)**thirteen\_hua**

October 15, 2021 at 10:23 am

Hi, after I upload files with the file system, The uploaded file name cannot be found when use “spiffs opendir (folder\_name)”. i don't know why.

[Reply](#)**Ian**

November 13, 2021 at 8:34 pm

Hi,

[☰ Menu](#)

[Reply](#)**Sara Santos**

November 14, 2021 at 11:46 am

Hi.

I don't think so.

Regards,  
Sara

[Reply](#)**Stefan**

December 5, 2021 at 6:31 pm

for changing the design you would have to dive deeply into the vue-stuff (java / javascript based framework) that is working behind the curtain.

best regards Stefan

[Reply](#)**gianni**

December 5, 2021 at 4:17 pm

Menu



Hello,

always the best compliments for the resources and the work done. I use a machine linux mint 20.xx and, I often have difficulties in the paths of the files, in fact, obviously, different from windows. A description for linux would also be useful for us. That said, I stuck on the point "After this, with the ESP32 disconnected from your computer (that's the whole purpose of OTA), click on ESP32 Data Sketch Upload" in fact it describes the path for windows (C:\Users\sarin\AppData\Local\Temp\arduino\_build\_675367\Web\_server\_OTA\_ESP32\_Example\_2.spiiffs.bin) but not for linux. Where should I find the affected.bin file? I did some research but they didn't show, exactly the file you described.  
any help would be appreciated, thanks.

[Reply](#)



**Stefan**

December 5, 2021 at 6:29 pm

The filename is the same as your \*.ino-file name just with the extension \*.bin

On a windows machine you would type a part of the filename and the search-function would show you all files containing the characters you have searched for.

Does Linux not have something similar?

Me personal I use the a tool named "everything.exe" which is listening to the filesystem all the time and recognises any change in realtime.

So a simple search for \*.bin

over ALL folders of my 2TB hardisk with 1.3 million ! files  
~~and sorting the files newest first~~

Menu



best regards Stefan

[Reply](#)

gianni

December 5, 2021 at 7:45 pm

680 / 5000

Risultati della traduzione

Thanks for the reply. I apologize for any errors and misunderstandings but I use google translator, I'm Italian. also for this reason, something is lost in the translation of the examples. I seemed to understand that, in windows the "constant" folder is "\ AppData \ Local \ " the rest is variable. In this case is there an equivalent in linux? I am not an expert, otherwise I would not follow the tutorials. in addition are you sure that the name of the generated file corresponds to the name of the project.bin (in linux)? I ask because the research shows eg. projectName.ino.doitESP32devkitV1.bin, and again projectName.ino.bin and many other places in different places. It becomes a stressful quest, in my case.

[Reply](#)

Stefan

December 6, 2021 at 7:11 am

Hi Gianni,

you can test this by choosing a very specific name for your \*.ino-file example:

≡ Menu



then the filename will be

I-want-to-find-you-xyz.ino.esp32.bin

and this \*.bin-file is stored in the eat same folder as the \*.ino-file

This is how it works in windows.

If this is different on linux. Just another thing that makes Linux beginner-UN-friendly

If you sort the files by date and the last thing you did was creating this \*.bin-file

it should be on top of the \*.bin-files or at least in the top 10

best regards Stefan

[Reply](#)



gianni

December 6, 2021 at 2:43 pm

Thanks for the replies.

after a day of research I came to some conclusions / solutions for me which place, could be useful to others:

As said I am using a machine with OS linux mint 20.

Regarding the project, at the point of creation “sketch data upload, upload failed” look for the description of the binary file, as described in the tutorial. take note only of the partial name (my full name in arduino IDE is: [SPIFFS] upload:

/tmp/arduino\_build\_462571/SPIFFS\_AsyncElegantOTA\_051221.spiffs.bin) copy the partial: arduino\_build\_462571.

I open the file manager “NEMO” I go to the / proc folder and look for: arduino\_build\_462571. in a few seconds it displays the folder containing the file.bin (SPIFFS\_AsyncElegantOTA\_051221.spiffs.bin in my case)

/ proc is a virtual folder with many other folders inside that will be deleted and recreated, with different names at each restart. So / proc you will always find it but the inside will always be different. As I said, look for the arduino\_build\_xxxxxx reference “always” in / proc. Use “nemo” and in a few seconds you will find the file. do not use “caja, nautilus etc” as it would take hours to find it among the dozens and dozens of folders. Do not use other managers eg. fsearch, they will not read virtual folders, I have tried 7 of the most famous but have not found anything. That’s it, I hope it helps someone by avoiding going crazy like me.

**Sara Santos**

December 6, 2021 at 11:49 am

Hi.

It should be displayed in your Arduino IDE as shown in the picture:

<https://i1.wp.com/randomnerdtutorials.com/wp-content/uploads/2021/01/Get-SPIFFS-Bin-File-Path.png>

Regards,

Sara

[Reply](#)**Bryan**

December 14, 2021 at 10:18 pm

Hi Rui/Sara

Menu



Thanks for the tutorial. I was using arduion 2 beta, but it didn't support elegantOTA file uploads using the ESP32\_Sketch\_Data\_Upload tool, so I dropped back to arduino ide 1.8.16, which does support the tool (linux desktop). However, I can export the firmware binary and upload it via ElegantOTA webserver on the ESP32, but when I run the ESP32\_Sketch\_Data\_Upload tool to update spiffs files, it immediately says "SPIFFS Error: serial port not defined!" and stops compilation/uploading. So, it never creates the spiffs.bin file that I need to upload via OTA to the ESP32. Any suggestions other than possibly going backwards further with the IDE?? I'd really like to be able to update the firmware and files remotely or I'll have to continually disconnect my esp32 to reprogram it for updates via USB. Thanks in advance!

[Reply](#)**Sara Santos**

December 15, 2021 at 10:24 am

Hi

Have you tried to scroll up on the debugging window and check if it has created the path?

Alternatively, you can also try to use VS Code instead:

<https://randomnerdtutorials.com/esp32-ota-over-the-air-vs-code/>

Regards,

Sara

[Reply](#)**ilker**

Menu



Thanks for code it works with wifi but I wanna do this ESP32 SIM800L(TTGO TCALL)'s gprs(Sim card).How can I do that?

[Reply](#)



**bradshaw lupton**

December 26, 2021 at 4:24 am

Thank you for your excellent work and assistance over the past 5 years!

I have an ESP32 AI-thinker, with a wifi http client script. that works super standalone.

The script runs fine with either OTAWebUpdater, or ElegantOTA.

With either OTA approach, on the AI-THINKER esp32 (my required platform). both `_abort()` on core 1, apparently identically. I tried putting some debug statements in `Update.cpp` on the OTAWebUpdater `_abort()` calls, but didnt get anything useful. I moved ot ElegantOTA and had the same problem.

I haven't exercised the camera.

Thank you for your help. the magic esp32 cam AI-thinker is required, with its cool sd card too!

Bradshaw in Buzzards Bay MA

23:14:47.477 ->

23:14:47.477 -> `abort()` was called at PC 0x40082066 on core 1

23:14:47.477 ->

≡ Menu



```
0x40091ce1:0x3ffccc40 0x40082066:0x3ffcccc0 0x400ecf47:0x3ffccd10  
0x400df3a9:0x3ffccd30 0x400dca46:0x3ffccd50 0x400dcc09:0x3ffccd70  
0x400d3192:0x3ffccda0 0x400d32a9:0x3ffcce00 0x400d8941:0x3ffcce60  
0x40162366:0x3ffcccec0 0x400d4fbe:0x3ffcccef0 0x400d68b2:0x3ffccf70  
0x400d6a55:0x3ffccfc0 0x400dbc66:0x3ffccfe0 0x400dbce5:0x3ffcd010  
0x400dc362:0x3ffcd030  
23:14:47.980 ->  
23:14:47.980 ->  
23:14:47.980 ->  
23:14:47.980 ->  
23:14:47.980 -> ELF file SHA256: 0000000000000000  
23:14:47.980 ->  
23:14:47.980 -> Rebooting...  
23:14:47.980 -> ets Jun 8 2016 00:22:57
```

[Reply](#)



**Bradshaw Lupton**

December 26, 2021 at 4:27 am

abort() was called at PC 0x40082066 on core 1

is there a way to convert the PC back to a source line, or library containing the code? That would facilitate my debugging.

Bradshaw

[Reply](#)

**bradshaw lupton**

December 26, 2021 at 2:38 pm

elegantOTA is working on an esp32 dev, see for OTA and minimal spiffs.

I dont seem to have the same set the memory model with the AI-thinker esp32 or the esp32s-cam boards.

Seems I need to learn more about the memory model specification. I bet that is in boards.txt.

Thanks Bradshaw at Buzzards Bay.

[Reply](#)**Andrew Clark**

January 3, 2022 at 7:09 pm

Hi. Excellent tutorial. I have followed the steps to perform the basic firmware update and everything goes well until I perform the actual update. The process starts and it jumps from 0 to “25%” quickly and then just stalls. Not sure what’s happening but I have repeated several times with no luck. Using Arduino IDE 1.8.10 on Mac with an ESP32 by Heltec. Any thoughts? Thanks!

[Reply](#)**Sara Santos**

Menu



Hi.

It is better to post this issue on the library issues page:

<https://github.com/ayushsharma82/AsyncElegantOTA/issues>

Regards,

Sara

[Reply](#)



**Stefan**

January 5, 2022 at 3:53 pm

Did you check if the WiFi-signal was strong enough? In my experience if you have a very weak signal RSSI below -75dbi it becomes difficult to do OTA-updates.

Second thing to check: what is the partition-scheme that you are using?  
it should be a partition scheme that has 1MB for OTA  
best regards Stefan

[Reply](#)



**Bernie**

March 29, 2022 at 4:57 am

I am having the same problem as reported by Andrew Clark. I don't have a signalling issue.

not enough sram or flash to fit this file.

If this is a partition issue, how do I set the partition scheme to 1 MB? And how does this solve the lack of memory problem?

Thanks

Bernie

[Reply](#)



Bernie

March 29, 2022 at 5:01 am

Note – I am using Arduino IDE 2.0 – if that makes a difference.

[Reply](#)



Bernie

April 1, 2022 at 4:30 am

Problem solved (with the help of Ayush).

The issue I was having is that my application (vs1053 chip) produces 1,250 external interrupts per second. When you have such high rates of external or timer interrupts, the bootloader refuses to complete without a warning.

So I had the vs1053 stop producing interrupts before the upload (this can be

[Reply](#)**Laszlo Zsiros**

January 16, 2022 at 4:47 pm

I loaded up the both codes for ESP32 from your tutorial (basic example and websocket server with LED).

Updated them with each other to test the OTA function. Update function worked well, but after update new sketch does not loaded to run without a reset button push. But in this case I should go to the controller to reset microcontroller the OTA function is less useful.

What should I do to restart automatically after update and avoid manual restart by reset button?

[Reply](#)**Matthew Allan**

January 19, 2022 at 10:48 pm

Hi Sara,

Is there any way that you know of that I can upload both FW.bin and SPIFFS.bin files OTA via an HTTPS webpage? I have successfully done it with FW.bin files using the generic HttpsOTAUpdate.h library but have not been able to get spiffs files to work! I would like to have multiple ESP32 devices update themselves from the same github repository when the reference FW version.text file is changed. Again this I have achieved but only

[☰ Menu](#)

[Reply](#)**Cornflake**

February 4, 2022 at 11:57 pm

Hi

I have to warn you.

The zipped elegant OTA Webpage Code in the file elegantWebpage.h in the Variable ELEGANT\_HTML[] PROGMEM is obfuscated. For this you can't know what it really do. My noscript Addon in the Browser informed me, that the Code connect to an CDN for buymeaCoffee Webpage. If this Page will be one time be hacked, then all IoT Devices who use the AsyncElegantOTA Library are in the danger of infection by a Virus or can be hacked.

Please check the Source Code for your own to decide if this is a problem for you.

Greeting Cornflake

[Reply](#)**Ayush**

February 5, 2022 at 8:28 pm

Hi Cornflake,

This is Ayush, creator & maintainer of ElegantOTA, I came to know about your comment when a dear follower of mine sent me a email stating your comment. I would just like to issue a bit of guidance for you:

[☰ Menu](#)

webpage source is on the github repository itself for transparency.

Please don't spread fake news before doing complete research on the topic. You could have compiled the webpage yourself and see if the hash of file was different. Open source is built on trust and I have no intention of disrupting it.

On a side note, the noscript tag refers to "Buy me a coffee" SDK which uses HTTPS and their own domain. For someone to spoof "Buy me a coffee" domain, they have to be on your LAN network which in itself makes it a useless target because the attacker is already inside your network.

[Reply](#)



**Sara Santos**

February 8, 2022 at 11:00 am

Hi.

Thanks for clarifying this.

I love your libraries. Keep up the great work.

I have bought you several coffees some time ago:)

Regards,

Sara

[Reply](#)



**Bernie**

April 1, 2022 at 4:36 am

Problem of Elegant OTA uploading failing and freezing in the middle of the upload is solved (with the help of Ayush). See January 1, 2022 issue by Andrew Clark.

This problem happens only for very large uploads, such as 1 MB binary files. (Note AsyncWebServer is at least 800 KB).

The issue I had is that my application (vs1053 chip) produces 1,250 external interrupts per second. When you have such high rates of external or timer interrupts, the bootloader refuses to complete without a warning and freezes.

So I had the vs1053 stop producing interrupts before the upload (this can be done automatically by ElegantOA as well), and the uploads all completed.

[Reply](#)



**JoergSH**

April 2, 2022 at 3:43 pm

I ask You  
on Twitter to run as Stand alone Server. You say is won't work but that's wrong.

The Solution is:

```
/**********SSID and Pass for AP*****/  
static const char *ap_ssid = "Esp_32";  
static const char *ap_pass = "temp_pass";
```

AsvncWebServer server(80):

```
void setup(void) {
    Serial.begin(115200);

    WiFi.softAP(ap_ssid, ap_pass);
    Serial.print("Access point running. IP address: ");
    Serial.print(WiFi.softAPIP());
    Serial.println("");

    // server.begin();
    server.on("/", HTTP_GET, [](AsyncWebRequest *request) {
        request->send(200, "text/plain", "Hi! I am ESP32.");
    });

    AsyncElegantOTA.begin(&server); // Start ElegantOTA
    server.begin();
    Serial.println("HTTP server started");
}
```

[Reply](#)



**frakor**

May 9, 2022 at 9:23 am

Hello. I have a question.

Is it possible to loosen the LOOP () loop even if the microcontroller does not connect to the wifi network? It seems that the cycle in the setup () section waits for registration in the wifi network and only then initializes loop (). I need the main program to run independently of the connection. I am new to the field and I welcome any advice. I apologize for the imperfect English I'm Czech and I took the help of a google translator.

≡ Menu



**Sara Santos**

May 10, 2022 at 8:55 pm

Hi.

You can use Wi-Fi events instead.

Check this tutorial: <https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/#10>

Regards,

Sara

[Reply](#)

**Joerg**

May 10, 2022 at 9:15 pm

I think You could put that part in the loop like :

```
void loop() {  
  
    // Wait for connection  
    If (WiFi.status() = WL_CONNECTED) {  
  
        Serial.println("");  
        Serial.print("Connected to ");  
        Serial.println(ssid);  
        Serial.print("IP address: ");  
        Serial.println(WiFi.localIP());
```

```
});
```

```
AsyncElegantOTA.begin(&server); // Start ElegantOTA  
server.begin();  
Serial.println("HTTP server started");  
}
```

```
// put Your code here
```

```
}
```

[Reply](#)



**Eric Carter**

May 12, 2022 at 6:29 pm

So had this all working wonderfully on a load of devices then upgraded to Ubuntu 22.04! Now in case anyone else hits this the OTA update will not work on the version of Firefox bundled with Ubuntu because it is a snap package. You need to remove it and install a stand alone version as per <https://www.omgubuntu.co.uk/2022/04/how-to-install-firefox-deb-apt-ubuntu-22-04>

There is also another little problem that crops up with ESPAsyncWebserver due to recent changes in ESP32 framework so if you end up with compiler errors involving mbedtls when using version 2.0.3 try this solution

<https://github.com/philbowles/ESPAsyncWebServer/issues/3>

[Reply](#)

**Anh**

May 26, 2022 at 3:25 am

I can't login to the ip address even though the serial monitor port prints out:  
HTTP server started. Can someone tell me how to fix it?

[Reply](#)**Sara Santos**

May 30, 2022 at 8:42 am

Hi.

Can you provide more details about the issue?

Regards,

Sara

[Reply](#)

## Leave a Comment

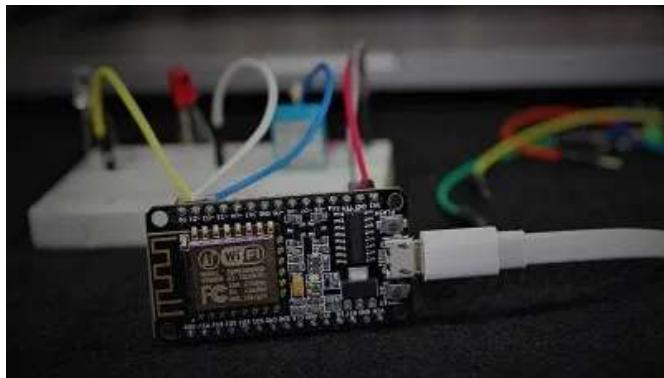
[☰ Menu](#)

Name \* Email \* Website Notify me of follow-up comments by email. Notify me of new posts by email.

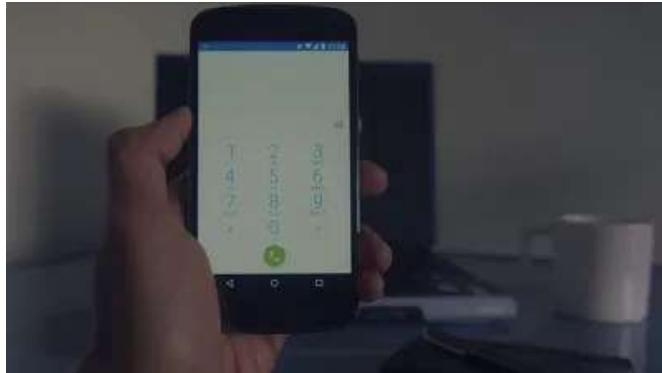
Post Comment



[Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »](#)



[Home Automation using ESP8266 eBook and video course » Build IoT and home](#)



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





































































































































































[About](#)   [Support](#)   [Terms and Conditions](#)   [Privacy Policy](#)   [Refunds](#)   [Complaints' Book](#)

[MakerAdvisor.com](#)   [Join the Lab](#)

Copyright © 2013-2022 · RandomNerdTutorials.com · All Rights Reserved

Exclusive Member of Mediavine Home