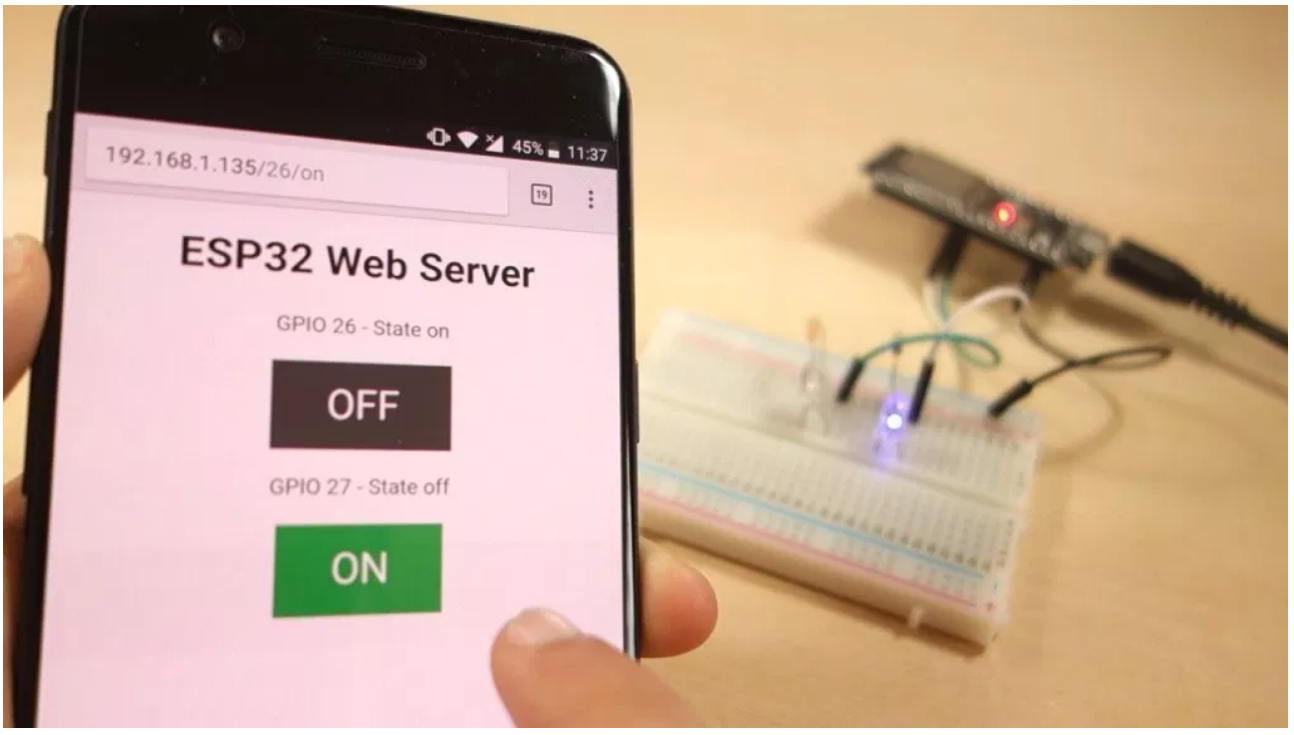
ESP32 Web Server – Arduino IDE

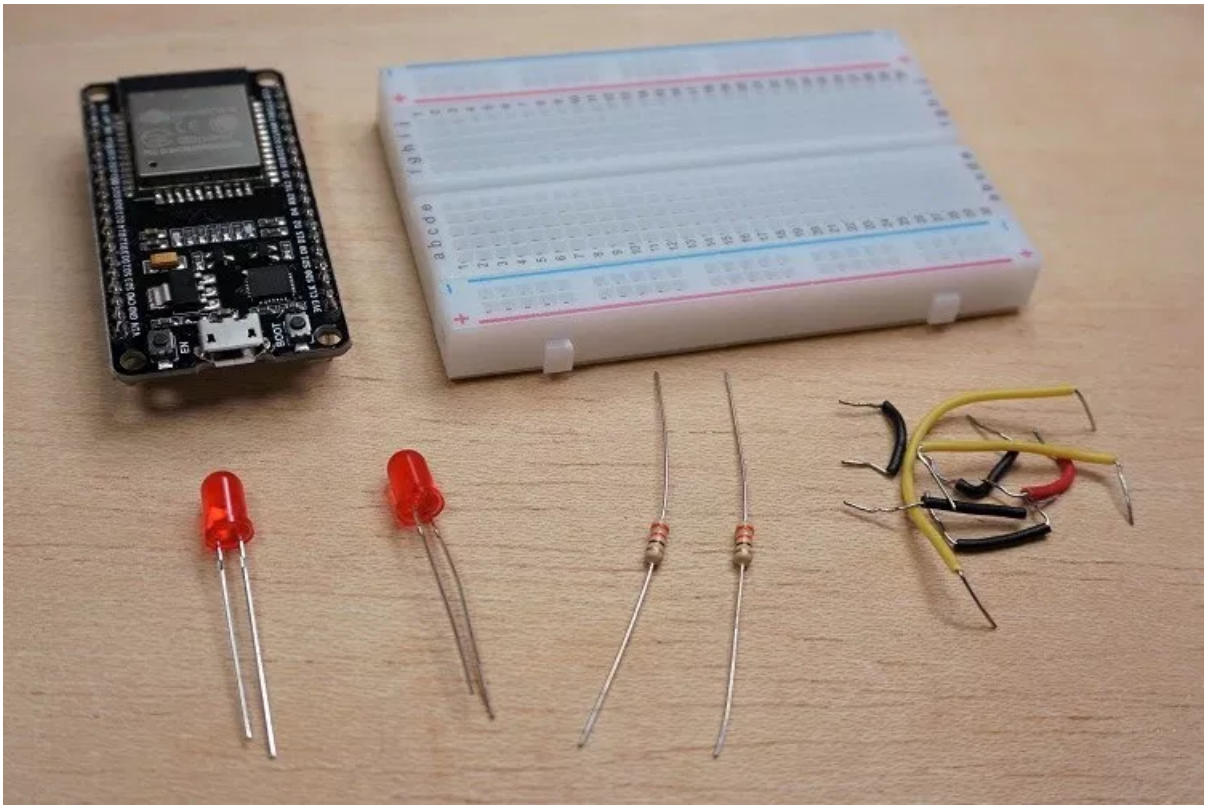
In this project you'll create a standalone web server with an ESP32 that controls outputs (two LEDs) using the Arduino IDE programming environment. The web server is mobile responsive and can be accessed with any device that as a browser on the local network. We'll show you how to create the web server and how the code works step-by-step.



If you want to learn more about the ESP32, read Getting Started Guide with ESP32.

Watch the Video Tutorial

This tutorial is available in video format (watch below) and in written format (continue reading this page).



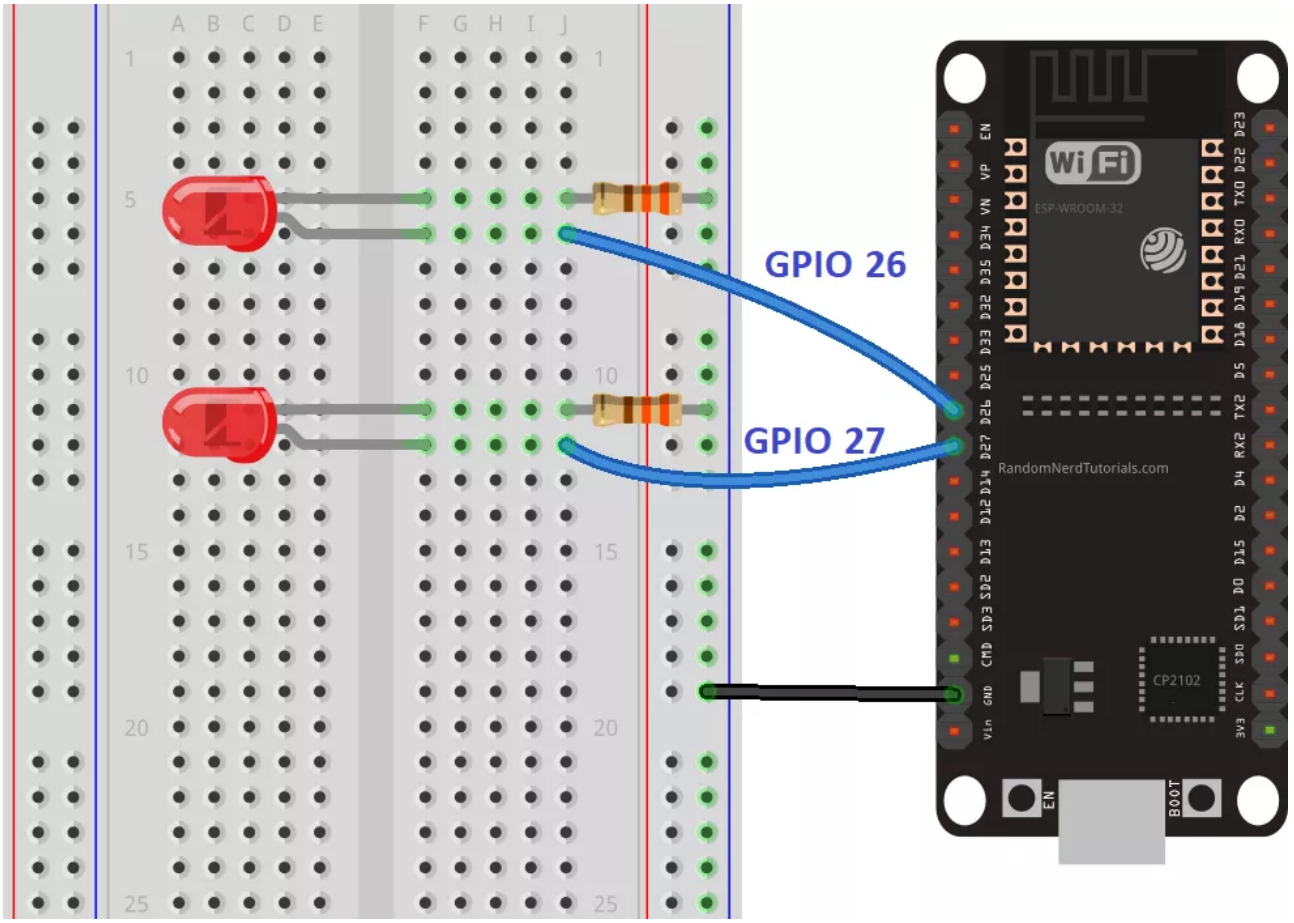
- ESP32 development board – read ESP32 Development Boards Review and Comparison
- 2x 5mm LED
- 2x 330 Ohm resistor
- Breadboard
- Jumper wires

You can use the preceding links or go directly to MakerAdvisor.com/tools to find all the parts for your projects at the best price!

Schematic

Start by building the circuit. Connect two LEDs to the ESP32 as shown in the following schematic diagram – one LED connected to GPIO 26 , and the other to GPIO 27.

Note: We're using the ESP32 DEVKIT DOIT board with 36 pins. Before assembling the circuit, make sure you check the pinout for the board you're using.



ESP32 Web Server Code

Here we provide the code that creates the ESP32 web server. Copy the following code to your Arduino IDE, but don't upload it yet. You need to make some changes to make it work for you.

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

// Load Wi-Fi library
#include <WiFi.h>

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

// Set web server port number to 80
WiFiServer server(80);

// Variable to store the HTTP request
String header;
```

```
// Auxiliar variables to store the current output state
String output26State = "off";
String output27State = "off";

// Assign output variables to GPIO pins
const int output26 = 26;
const int output27 = 27;
```

View raw code

Setting Your Network Credentials

You need to modify the following lines with your network credentials: SSID and password. The code is well commented on where you should make the changes.

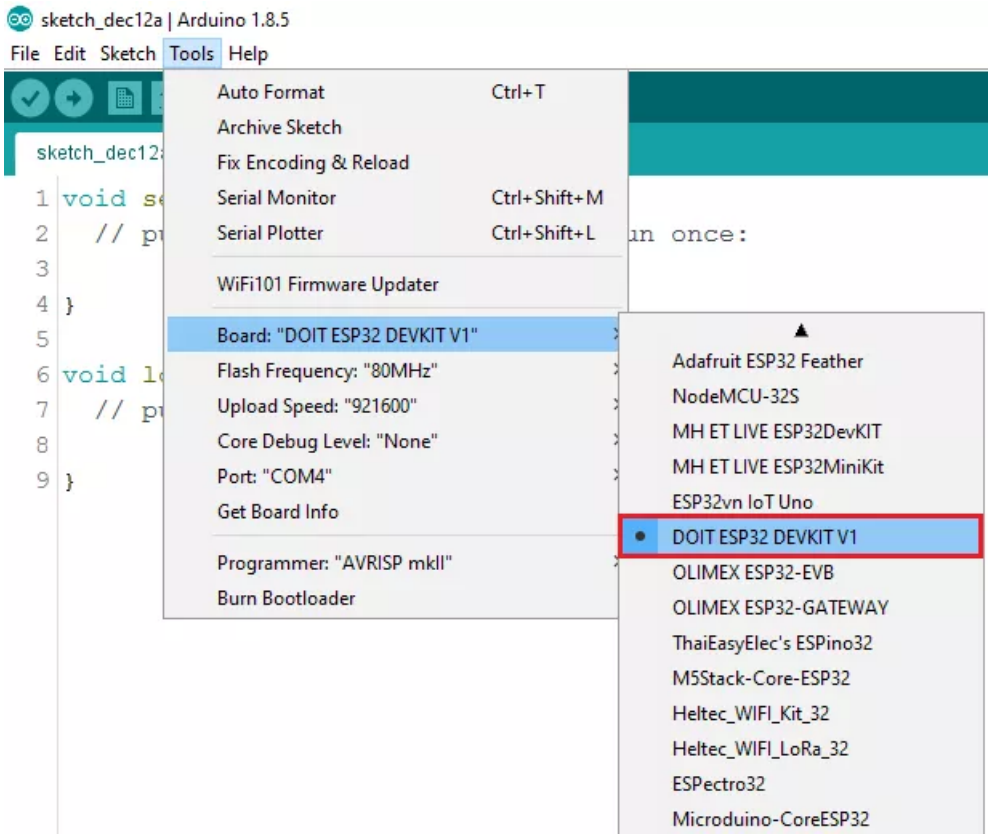
```
// Replace with your network credentials
const char* ssid     = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

Uploading the Code

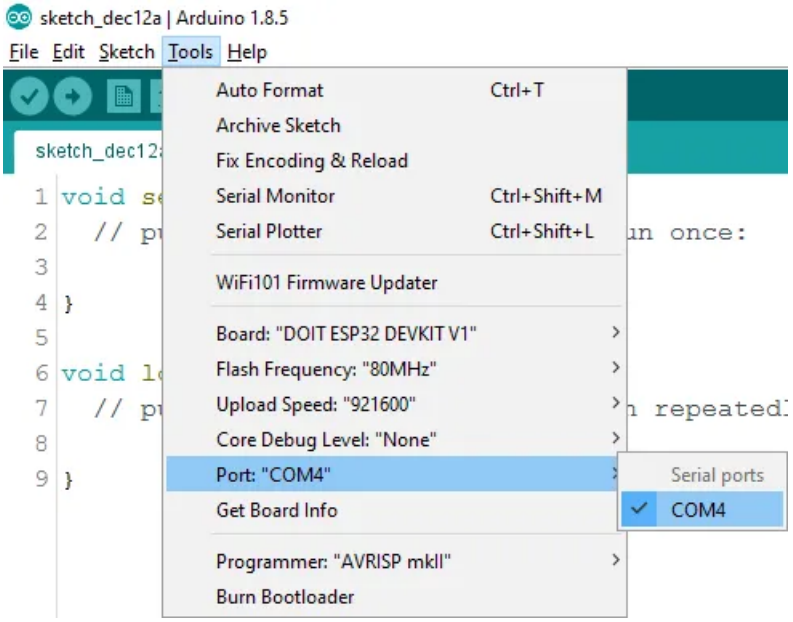
Now, you can upload the code and and the web server will work straight away. Follow the next steps to upload code to the ESP32:

1) Plug your ESP32 board in your computer;

2) In the Arduino IDE select your board in **Tools** > **Board** (in our case we're using the ESP32 DEVKIT DOIT board);

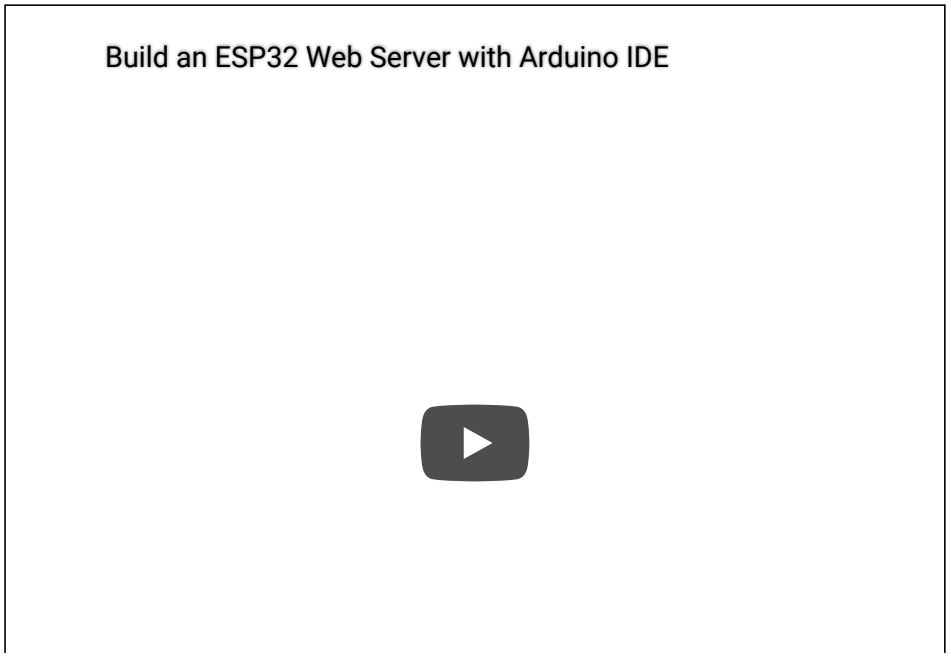


3) Select the COM port in **Tools** > **Port**.



4) Press the **Upload** button in the Arduino IDE and wait a few seconds while the code compiles and uploads to your board.

5) Wait for the "**Done uploading**" message.



Project Overview

Before going straight to the project, it is important to outline what our web server will do, so that it is easier to follow the steps later on.

- The web server you'll build controls two LEDs connected to the ESP32 GPIO 26 and GPIO 27 ;
- You can access the ESP32 web server by typing the ESP32 IP address on a browser in the local network;
- By clicking the buttons on your web server you can instantly change the state of each LED.

This is just a simple example to illustrate how to build a web server that controls outputs, the idea is to replace those LEDs with a relay, or any other electronic components you want.

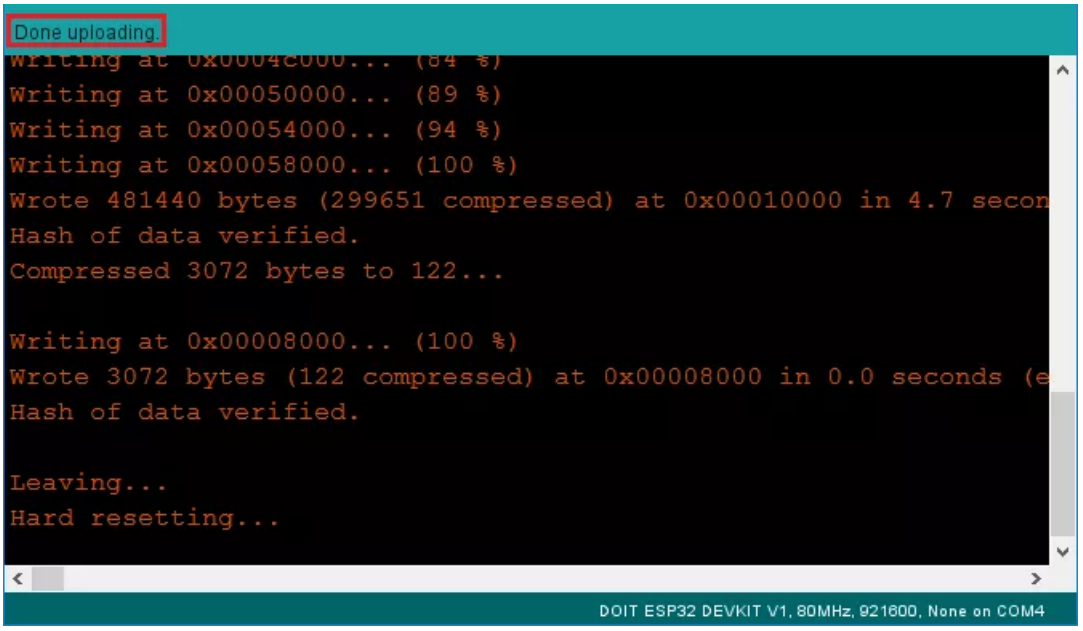
Installing the ESP32 board in Arduino IDE

There's an add-on for the Arduino IDE that allows you to program the ESP32 using the Arduino IDE and its programming language. Follow one of the following tutorials to prepare your Arduino IDE:

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

Parts Required

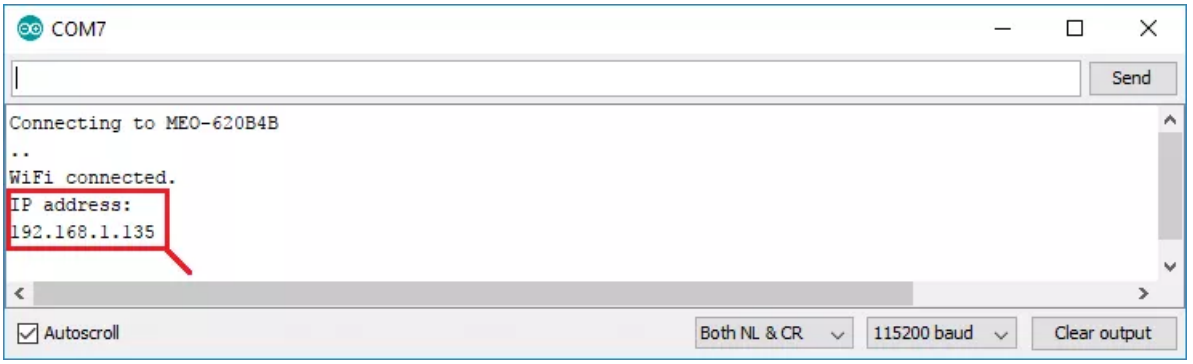
For this tutorial you'll need the following parts:



Finding the ESP IP Address

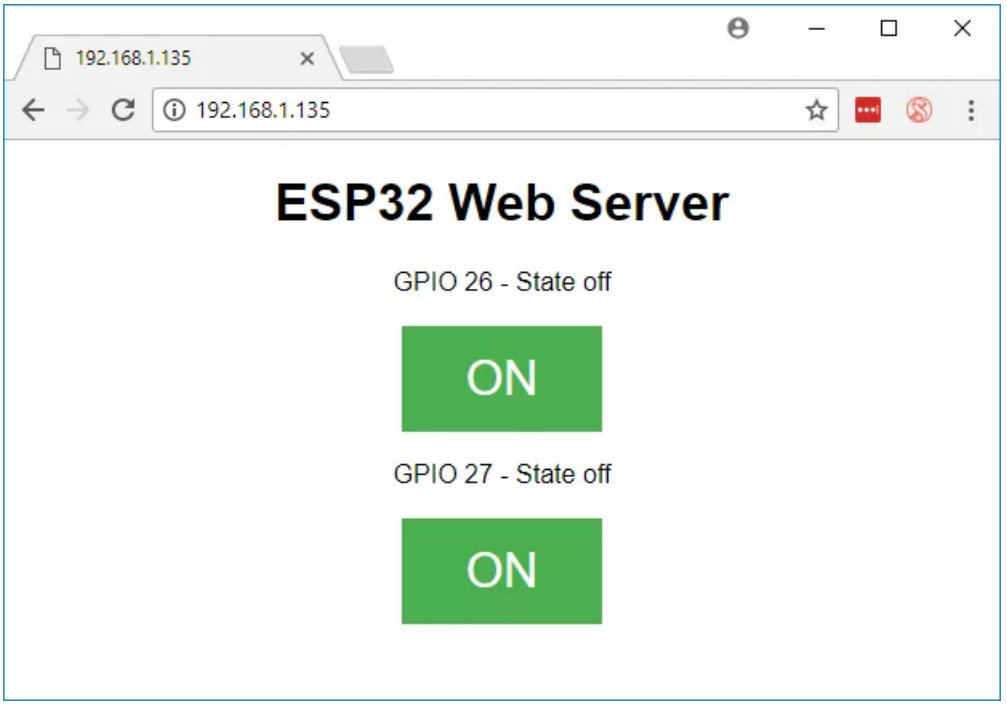
After uploading the code, open the Serial Monitor at a baud rate of 115200.

Press the ESP32 EN button (reset). The ESP32 connects to Wi-Fi, and outputs the ESP IP address on the Serial Monitor. Copy that IP address, because you need it to access the ESP32 web server.

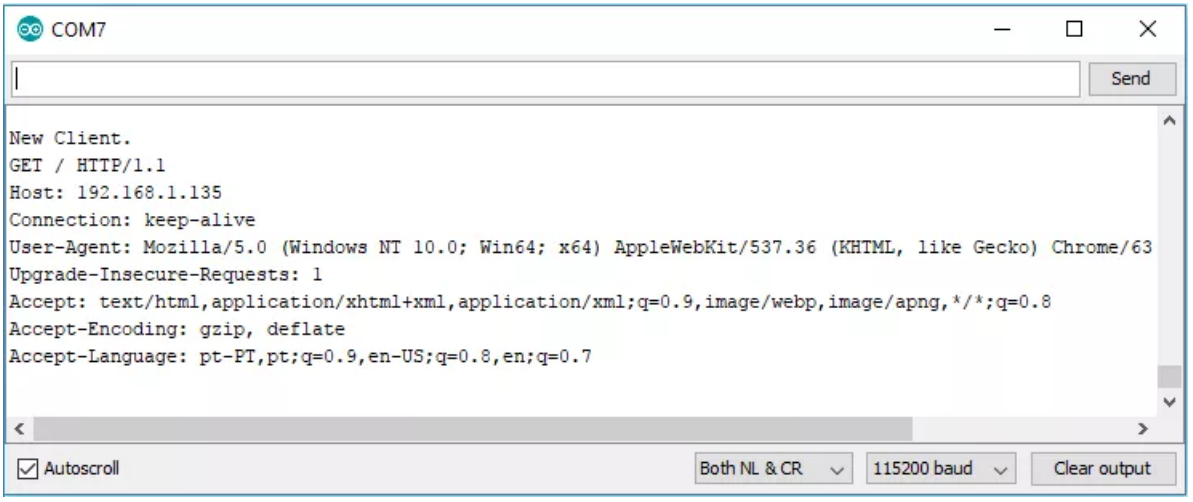


Accessing the Web Server

To access the web server, open your browser, paste the ESP32 IP address, and you'll see the following page. In our case it is **192.168.1.135**.



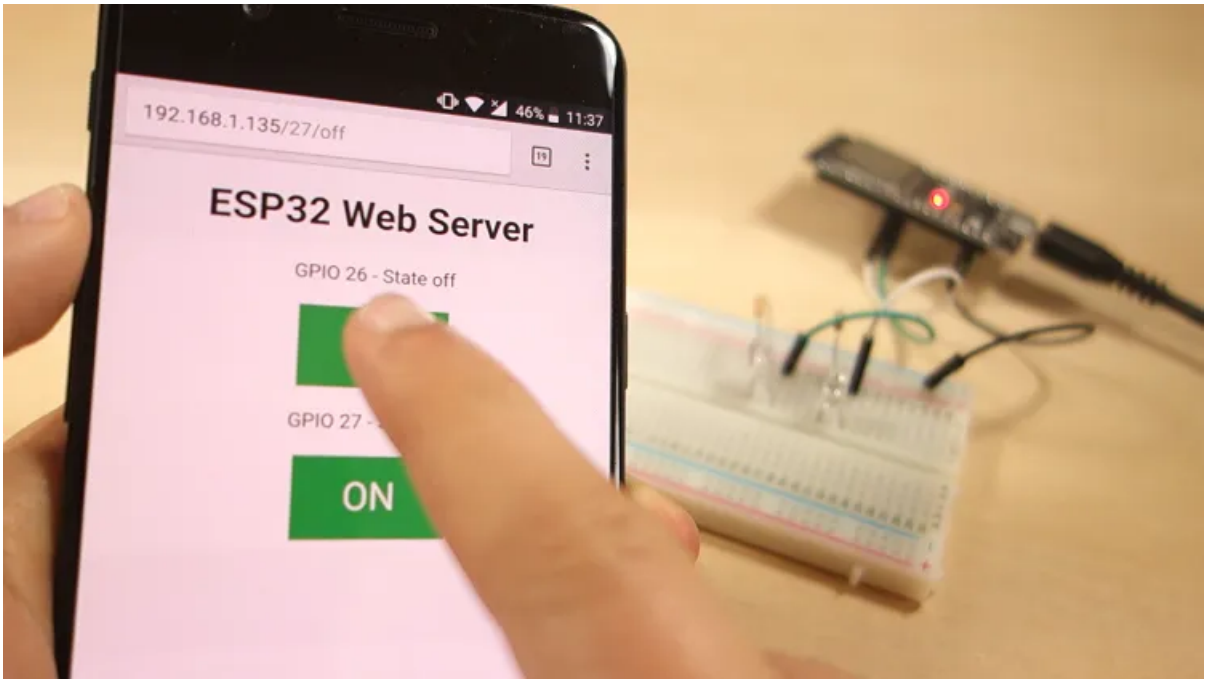
If you take a look at the Serial Monitor, you can see what's happening on the background. The ESP receives an HTTP request from a new client (in this case, your browser).



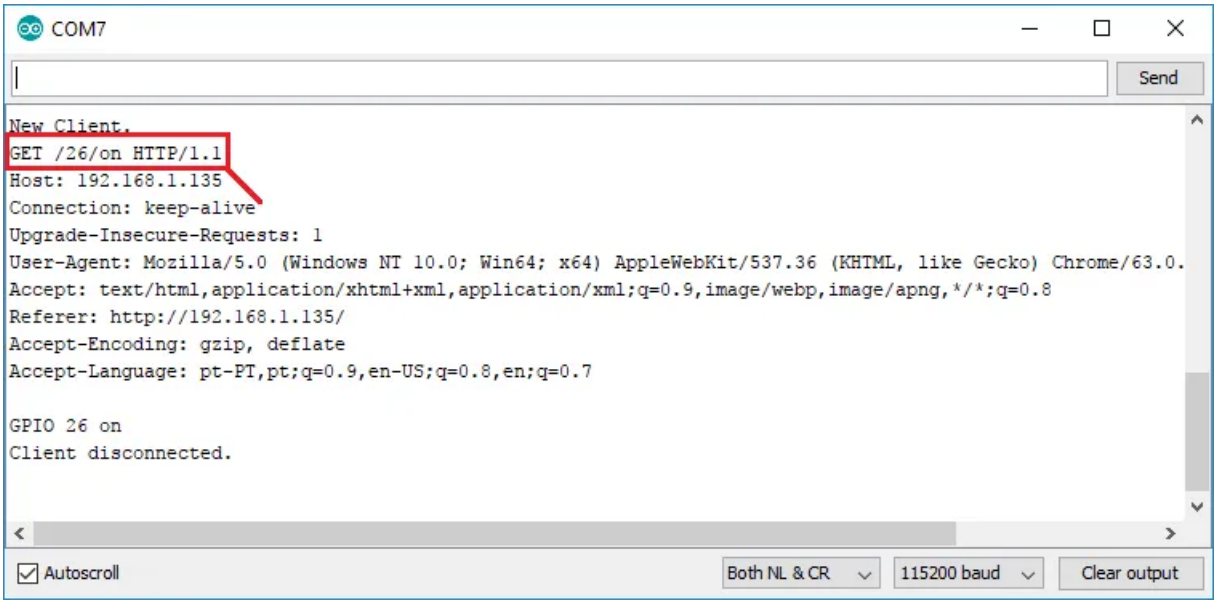
You can also see other information about the HTTP request.

Testing the Web Server

Now you can test if your web server is working properly. Click the buttons to control the LEDs.



At the same time, you can take a look at the Serial Monitor to see what's going on in the background. For example, when you click the button to turn GPIO 26 ON, ESP32 receives a request on the **/26/on** URL.



When the ESP32 receives that request, it turns the LED attached to GPIO 26 ON and updates its state on the web page.

The button for GPIO 27 works in a similar way. Test that it is working properly.

How the Code Works

In this section will take a closer look at the code to see how it works.

The first thing you need to do is to include the WiFi library.

```
#include <WiFi.h>
```

As mentioned previously, you need to insert your ssid and password in the following lines inside the double quotes.

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

Then, you set your web server to port 80.

```
WiFiServer server(80);
```

The following line creates a variable to store the header of the HTTP request:

```
String header;
```

Next, you create auxiliar variables to store the current state of your outputs. If you want to add more outputs and save its state, you need to create more variables.

```
String output26State = "off";
String output27State = "off";
```

You also need to assign a GPIO to each of your outputs. Here we are using GPIO 26 and GPIO 27 . You can use any other suitable GPIOs.

```
const int output26 = 26;
const int output27 = 27;
```

setup()

Now, let's go into the `setup()` . First, we start a serial communication at a baud rate of 115200 for debugging purposes.

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

You also define your GPIOs as OUTPUTs and set them to LOW.

```
// Initialize the output variables as outputs
pinMode(output26, OUTPUT);
pinMode(output27, OUTPUT);

// Set outputs to LOW
digitalWrite(output26, LOW);
digitalWrite(output27, LOW);
```

The following lines begin the Wi-Fi connection with `WiFi.begin(ssid, password)` , wait for a successful connection and print the ESP IP address in the Serial Monitor.

```
// Connect to Wi-Fi network with SSID and password
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
// Print local IP address and start web server
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
server.begin();
```

loop()

In the `loop()` we program what happens when a new client establishes a connection with the web server.

The ESP32 is always listening for incoming clients with the following line:

```
WiFiClient client = server.available(); // Listen for incoming clients
```

When a request is received from a client, we'll save the incoming data. The while loop that follows will be running as long as the client stays connected. We don't recommend changing the following part of the code unless you know exactly what you are doing.

```
if (client) { // If a new client connects,
  Serial.println("New Client."); // print a message out in the serial port
  String currentLine = ""; // make a String to hold incoming data from the client
  while (client.connected()) { // loop while the client's connected
    if (client.available()) { // if there's bytes to read from the client,
      char c = client.read(); // read a byte, then
      Serial.write(c); // print it out the serial monitor
      header += c;
      if (c == '\n') { // if the byte is a newline character
      // if the current line is blank, you got two newline characters in a row.
      / that's the end of the client HTTP request, so send a response:
        if (currentLine.length() == 0) {
        // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
        // and a content-type so the client knows what's coming, then a blank line:
          client.println("HTTP/1.1 200 OK");
          client.println("Content-type:text/html");
          client.println("Connection: close");
          client.println();
```

The next section of if and else statements checks which button was pressed in your web page, and controls the outputs accordingly. As we've seen previously, we make a request on different URLs depending on the button pressed.

```
// turns the GPIOs on and off
if (header.indexOf("GET /26/on") >= 0) {
  Serial.println("GPIO 26 on");
  output26State = "on";
  digitalWrite(output26, HIGH);
} else if (header.indexOf("GET /26/off") >= 0) {
  Serial.println("GPIO 26 off");
```

```
    output26State = "off";
    digitalWrite(output26, LOW);
  } else if (header.indexOf("GET /27/on") >= 0) {
    Serial.println("GPIO 27 on");
    output27State = "on";
    digitalWrite(output27, HIGH);
  } else if (header.indexOf("GET /27/off") >= 0) {
    Serial.println("GPIO 27 off");
    output27State = "off";
    digitalWrite(output27, LOW);
  }
```

For example, if you've press the GPIO 26 ON button, the ESP32 receives a request on the **/26/ON URL** (we can see that that information on the HTTP header on the Serial Monitor). So, we can check if the header contains the expression **GET /26/on**. If it contains, we change the `output26state` variable to ON, and the ESP32 turns the LED on.

This works similarly for the other buttons. So, if you want to add more outputs, you should modify this part of the code to include them.

Displaying the HTML web page

The next thing you need to do, is creating the web page. The ESP32 will be sending a response to your browser with some HTML code to build the web page.

The web page is sent to the client using this expressing `client.println()` . You should enter what you want to send to the client as an argument.

The first thing we should send is always the following line, that indicates that we are sending HTML.

```
<!DOCTYPE HTML><html>
```

Then, the following line makes the web page responsive in any web browser.

```
client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
```

And the following is used to prevent requests on the favicon. – You don't need to worry about this line.

```
client.println("<link rel=\"icon\" href=\"data:,\">");
```

Styling the Web Page

Next, we have some CSS text to style the buttons and the web page appearance. We choose the Helvetica font, define the content to be displayed as a block and aligned at the center.

```
client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-ali
```

We style our buttons with the #4CAF50 color, without border, text in white color, and with this padding: 16px 40px. We also set the text-decoration to none, define the font size, the margin, and the cursor to a pointer.

```
client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");
client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");
```

We also define the style for a second button, with all the properties of the button we've defined earlier, but with a different color. This will be the style for the off button.

```
client.println(".button2 {background-color: #555555;}</style></head>");
```

Setting the Web Page First Heading

In the next line you can set the first heading of your web page. Here we have "**ESP32 Web Server**", but you can change this text to whatever you like.

```
// Web Page Heading
client.println("<h1>ESP32 Web Server</h1>");
```

Displaying the Buttons and Corresponding State

Then, you write a paragraph to display the GPIO 26 current state. As you can see we use the `output26State` variable, so that the state updates instantly when this variable changes.

```
client.println("<p>GPIO 26 - State " + output26State + "</p>");
```

Then, we display the on or the off button, depending on the current state of the GPIO. If the current state of the GPIO is off, we show the ON button, if not, we display the OFF button.

```
if (output26State=="off") {
  client.println("<p><a href=\"/26/on\"><button class=\"button\">ON</button></a></p>");
} else {
  client.println("<p><a href=\"/26/off\"><button class=\"button button2\">OFF</button></a></p>");
}
```

We use the same procedure for GPIO 27 .

Closing the Connection

Finally, when the response ends, we clear the `header` variable, and stop the connection with the client with `client.stop()`.

```
// Clear the header variable
header = "";
// Close the connection
client.stop();
```

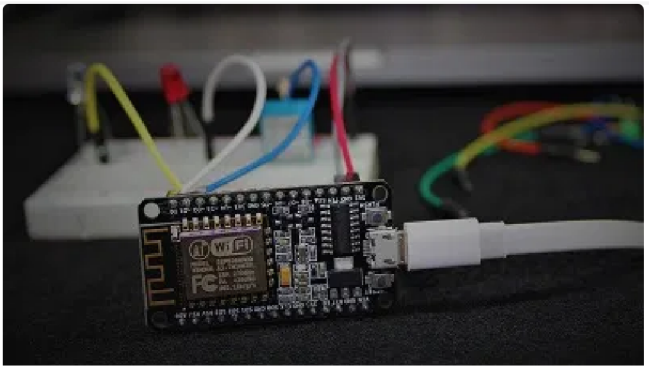
Wrapping Up

In this tutorial we've shown you how to build a web server with the ESP32. We've shown you a simple example that controls two LEDs, but the idea is to replace those LEDs with a relay, or any other output you want to control. For more projects with ESP32, check the following tutorials:

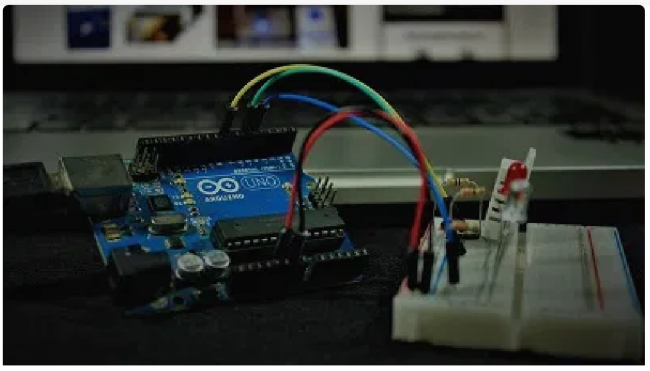
- Build an All-in-One ESP32 Weather Station Shield
- ESP32 Servo Motor Web Server
- Getting Started with ESP32 Bluetooth Low Energy (BLE)
- More ESP32 tutorials

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.

Recommended Resources



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...

Control Sonoff Basic Switch with ESP Easy Firmware and Node-RED

ESP32/ESP8266 PWM with MicroPython – Dim LED

ESP8266 Web Server using SPIFFS (SPI Flash File System) – NodeMCU

ESP32 Web Server using SPIFFS (SPI Flash File System)

ESP32/ESP8266 Digital Inputs and Digital Outputs with MicroPython

Installing ESP8266 Board in Arduino IDE (Windows, Mac OS X, Linux)

108 thoughts on “ESP32 Web Server – Arduino IDE”

Duncan Amos
April 13, 2017 at 7:46 pm

It seems a waste of the capabilities within the ESP32 to just switch two LEDs when you can do that with an ESP-01 so easily and cheaply.

Nice to see the ESP32 making an appearance though…

Reply

Gerd
April 14, 2017 at 10:24 am

right

Reply

Rui Santos
April 16, 2017 at 6:00 pm

This is a basic project that readers can build upon, the idea is to share a basic web server example that you can customise to control any appliance easily

Reply

Sunil
April 13, 2017 at 10:00 pm

Very clear – must try this.

Reply

Rui Santos
April 16, 2017 at 6:00 pm

Thank you Sunil!

Reply

Ton
April 14, 2017 at 10:51 pm

Thanks for the sketch, works fine.

Reply

Rui Santos
April 16, 2017 at 5:59 pm

Thanks for letting me know!
Regards,
Rui

Reply

sunil vijaya
May 23, 2017 at 1:11 am

hi rui,

tried your code(meant for ESP32) here with ESP8266 wi fi shield connected to arduino uno 3,

i am getting errors. of course – i am doing something wrong initially.
the board i select is arduino uno. cant see any ESP stuff on drop down on Tools.
(what do i do to see the ESP8266 on drop down menu?)
programmer – arduinoisp.

i tried an update on formware but programmer not responding message.

here's the error i get at compilation:
C:\Users\i5\Documents\Arduino\libraries\arduino_585573\src/WiFi101.h:37:2: note: previous declaration 'wl_status_t WL_NO_SHIELD'

WL_NO_SHIELD = 255,

^

In file included from C:\Program Files (x86)\Arduino\libraries\WiFi\src/WiFi.h:26:0,

from C:\Users\i5\Documents\Arduino\ruiWiFi\ruiWiFi.ino:8:

C:\Program Files (x86)\Arduino\libraries\WiFi\src/utility/wl_definitions.h:52:26: error: redeclaration of 'WL_IDLE_STATUS'

WL_IDLE_STATUS = 0,

^
and so on where WL_ is encountered.

I think i mentioned that i got the ethernet shield on my first attempt.
but unfortunately i have struggled with this wi fi shield.

with other code i tried i get error – wifi shield NOT present

Pl. help.

sunil

Reply

Oliver
April 14, 2019 at 5:31 pm

Great tutorial!
I like your work!

Thanks for sharing your knowledge!

Reply

Niels
September 26, 2019 at 8:46 am

I'm very late on this but if anyone else ever has this issue; you'll have to add the ESP boards to your additional board ulrs in the arduino IDE. found here:
https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/

Reply

Patrick McGrath
January 18, 2018 at 12:20 am

Thank you for your efforts.

Reply

Joel E Farnham
February 22, 2018 at 11:17 pm

How would I implement a login page on this web server? I don't want unauthorized users switching equipment that is hooked up to the ESP32. Great tutorial. Awesome website too!

Reply

Rui Santos
March 14, 2018 at 6:16 pm

Hi,
I know how to do it, but unfortunately I don't have any tutorials on that subject…
Thanks for your feedback. Regards,
Rui

Reply

dilmac
March 9, 2018 at 7:12 pm

Thanks a lot for code.. Have nice day !

Reply

Sara Santos
March 10, 2018 at 10:04 am

You're welcome! ☺

Reply

Jean-Luc Berrier
March 10, 2018 at 10:18 am

Hi,

This sketch is giving me error, on arduino IDE and on PlatormIO

The IP adress is connected.
I can access it from outside. The lamps are going on/off. No problem
But an error message is shown at the serial monitor.
The 2 server sketch is showing the same behavior but it could NOT be connected from "outside".

WiFi connected.
IP address:

192.168.0.116
New Client.
GET /26/on HTTP/1.1
Host: 192.168.0.116
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 11_2_6 like Mac OS X) AppleWebKit/604.5.6 (KHTML, like Gecko) Version/11.0 MException in thread rx:
Traceback (most recent call last):
File “C:\Python27\Lib\threading.py”, line 801, in __bootstrap_inner
self.run()
File “C:\Python27\Lib\threading.py”, line 754, in run
self.__target(*self.__args, **self.__kwargs)

File “c:\users\jean-luc\.platformio\penv\lib\site-packages\serial\tools\miniterm.py”, line 448, in reader
self.console.write_bytes(data)
File “c:\users\jean-luc\.platformio\penv\lib\site-packages\serial\tools\miniterm.py”, line 63, in write_bytes
self.byte_output.write(byte_string)
IOError: [Errno 0] Error

Reply

Rui Santos
March 14, 2018 at 5:53 pm

Hi Jean,
As I said in a previous comment I don’t use PlatformIO.
That section describes the write behavior of the web server:

WiFi connected.
IP address:
192.168.0.116
New Client.
GET /26/on HTTP/1.1
Host: 192.168.0.116
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 11_2_6 like Mac OS X) AppleWebKit/604.5.6 (KHTML, like Gecko) Version/11.0 MException in

That’s exactly what it should be doing.

But where's that error being printed?

Traceback (most recent call last):
File "C:\Python27\Lib\threading.py", line 801, in __bootstrap_inner
self.run()
File "C:\Python27\Lib\threading.py", line 754, in run
self.__target(*self.__args, **self.__kwargs)

File "c:\users\jean-luc\.platformio\penv\lib\site-packages\serial\tools\miniterm.py", line 448, in reader
self.console.write_bytes(data)
File "c:\users\jean-luc\.platformio\penv\lib\site-packages\serial\tools\miniterm.py", line 63, in write_bytes
self.byte_output.write(byte_string)
IOError: [Errno 0] Error

Reply

jean-luc berrier
March 15, 2018 at 4:01 pm

Works normally on a 64bits machine...

Reply

Jean-Luc Berrier
March 10, 2018 at 10:25 am

The two lines:
const char* ssid = ".....";
const char* password = "....";
are giving me the following error in ARDUINO IDE:
"sketch_mar10a:33: error: invalid conversion from 'const char*' to 'char*' [-fpermissive]

WiFi.begin(ssid, password);"

I have to use:
char* ssid = ".....";
char* password = ".....";

On PlatformIO, this fault is not observed.

At the other hand, on PlatformIO, if I use “char*…” only, I’ve an error message.

Any idea?

Reply

Rui Santos
March 14, 2018 at 5:50 pm

Hi Jean,
I don’t use PlatformIO. Do you have the latest version of the ESP32 add-on for the Arduino IDE installed?

Reply

cristian
May 23, 2018 at 7:04 pm

How could this be done with a pic microcontroller, which part of this code would go in the pic microcontroller?

Reply

Rui Santos
June 11, 2018 at 4:55 pm

Hello Cristian, unfortunately I don’t have any tutorials with the PIC microcontroller…
Thanks for asking,
Rui

Reply

Bob M.
June 6, 2018 at 3:35 am

I'm trying this in a hotel room and I get the following:

Connecting to Hilton Honors
.....
WiFi connected
IP address:
172.20.0.58

And then when I try to access IP using http://172.20.0.58 the web page basically times out. So I added 172.20.0.58 as a firewall connection security exclusion, and still no luck as the web browser still times out. I'm thinking that this has something to do with the Hotel's network preventing me from "hacking" another computer. Any ideas? I'll try this again when I get home....oh and BTW neat stuff you are doing here with your site, thanks for sharing.

Reply

Rui Santos
June 11, 2018 at 4:20 pm

Are you using Google Chrome? Can you try different web browser? Some readers experienced problems with this project on Safari. Your code is correct and you are accessing the right IP address, so it should load the web server. Trying on different devices or using a different browser might help you find what's happening. (it shouldn't be necessary to change any security or firewall configurations... unless you have a very strict antivirus that blocks many features).

Reply

Ravian
August 1, 2018 at 6:55 pm

Great Tutorial man! Cheers!

Reply

Rui Santos
August 4, 2018 at 4:15 pm

Thanks for reading!

Reply

James Comben
August 10, 2018 at 9:55 am

cannot declare variable ‘server’ to be of abstract type ‘WiFiServer’
I am stuck, can you help?

Reply

Simone
August 18, 2018 at 7:42 pm

Hi everbody. First of all thank you for the great tutorial.

Unfortunately i think i found a bug, this is what I read in my serial monitor when I use chrome:

New Client.
GET / HTTP/1.1
Host: 192.168.1.110
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: it-IT,it;q=0.9,en-GB;q=0.8,en;q=0.7,en-US;q=0.6

Client disconnected.

New Client.

As you can see there is a string that should not be there (“New Client.”).
I found this error on chrome, both from a PC and a smartphone, while the incognito mode seems to work properly. I’ve tried edge too and it works well.

Any suggestions?

Reply

Sara Santos
August 20, 2018 at 3:24 pm

Hi Simone.
That situation can be solved using the trick we show here: https://rntlab.com/question/solved-esp32-web-server-drops-connection-crashes/
I hope this helps,
Regards, Sara.

Reply

Fernando
August 19, 2018 at 8:25 pm

It works fine with Arduino IDE thanks

Reply

tracy
September 11, 2018 at 1:02 am

I've been searching the internet trying to find out how to add some code to this to make my ip static with no luck, is there a simple way to do this that maybe escaping my search? thanks

Reply

Rui Santos
September 12, 2018 at 10:34 am

Hello Tracy,
Please take a look at this thread: https://rntlab.com/question/ip-address-of-server-keeps-changing/

Are you familiar with MAC Addresses? Basically a MAC address is a unique identifier assigned to your ESP32 (or any network device). You can use this function to retrieve your ESP32 MAC address while using a WiFi or web server example:

```
String mac = WiFi.macAddress();
Serial.println(mac);
```

Then, you login into your router dashboard (the router ip and login/pass are usually printed in the router label). You can assign a fixed IP address to your ESP MAC address, so it always has the same IP address.
The steps to assign a fixed IP to a MAC address is very similar to all the routers.

Reply

Peter
October 5, 2018 at 7:38 pm

Hi,

I have some kit led + resistors, but somehow I have only 220 ohm and not 330 ohm resistors in my kit, is this will work?

Kind regards

Reply

Sara Santos
October 6, 2018 at 9:38 am

Hi Peter!
Yes, you can use 220Ohm resistor without any problem.
Regards,
Sara ☺

Reply

joe
October 16, 2018 at 9:25 pm

great!! nice tutorial…

Reply

Sara Santos
October 17, 2018 at 8:37 am

Thanks ☺

Reply

Anwar Bashir
October 27, 2018 at 11:59 am

This is a great tutorial. I have the code working. Just one issue if I use ‘GET’ on WiFi credentials the are cached on the browser url, so can I use ‘POST’ instead. I tried but when I send the from with POST, the header.indexOf(“PUT…. does not see the data. Any idea what I am doing wrong?

Reply

Sara Santos
October 29, 2018 at 10:21 am

Hi Anwar.
I personally, never used that method. So, I have no idea what may be wrong.
I’m sorry I can’t help.
Regards,
Sara ☺

Reply

Dave
November 6, 2018 at 3:32 am

Hi Folks,

Great tutorials – I'm learning a lot. One startup problem though – it compiles and uploads nicely, then com7 starts laying down ".....". If I push the Reset button on the ESP32, I get:

.....ets Jun 8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:6084
load:0x40080000,len:6696
entry 0x400802e4
Connecting to Chicken_Run
....., and more dots. It never does connect though. It worked yesterday on the simpler issue, but I must have changed something.

Any ideas?

Thanks
Dave

Reply

Sara Santos
November 8, 2018 at 10:03 am

Hi Dave.
You have to hold down the BOOT button, until the dots disappear.
Take a look at bullet 4) on the ESP32 troubleshooting guide:
https://randomnerdtutorials.com/esp32-troubleshooting-guide/
Let me know if it helped.
Regards,
Sara ☺

Reply

Dave
November 8, 2018 at 1:33 pm

Hi Sara,

That worked perfectly – thank you.

It had connected nicely a couple of times, but I couldn't get it to repeat. Your comment explains why – on those times it worked, I must have accidentally held the BOOT button down long enough for it to go through. Obviously (DOH!), I hadn't put the length of the button press together with successful / unsuccessful connects, but your suggestion was – of course! – dead on.

Thank you. Your support is appreciated very much.

Dave, Canada

Reply

Sara Santos
November 8, 2018 at 1:54 pm

I'm glad it works!
Thank you! ☺

Reply

Vidarr
November 22, 2018 at 6:35 pm

Thanks for this. I am having a problem with pin 27. I have two ESP32 boards and the pin 27 doesn't work either one with this code. I check the code and don't see problem. What could this be?

Thank you!

Reply

Sara Santos
November 23, 2018 at 12:37 pm

Hi Vidar.
What is the problem with GPIO 27 exactly?
Which ESP32 board are you using?

Reply

luis moreno
December 15, 2018 at 10:29 pm

Hello i have a question how i can add more buttons to controle more leds ?

Reply

Rui Santos
December 17, 2018 at 3:50 pm

You need to duplicate the web page section that generates the HTML buttons, but I don't have any example on that exact subject

Reply

Chiefer
December 22, 2018 at 7:21 pm

How can I make some changes in html code here, I mean make it more elegant (css)? 😉

Reply

Sara Santos
December 26, 2018 at 4:33 pm

Hi Chiefer.
You can add the CSS directly on the HTML code and send it directly on the client.println() lines.
Or you can create a separated CSS file that you mention in the HTML text.
I recommend taking a look at the following tutorial that uses separated CSS and HTML files. It is a much easier way to change the appearance of the page.
https://randomnerdtutorials.com/esp32-web-server-spiffs-spi-flash-file-system/
I hope this helps.
Regards,
Sara

Reply

Gia Duy
January 17, 2019 at 6:20 am

Nice tutorial. Thanks for sharing Rui Santos. I have a problem want to ask. Could I implement this but using Ethernet cable to connect to the Internet instead of WiFi. And if yes how can I do that? Thanks a lot. Please answer me ASAP !!! ☺

Reply

Sara Santos
January 17, 2019 at 4:37 pm

Hi Gia.
I've never used ESP32 with Ethernet. I just use it with Wi-Fi.
We have a tutorial with Arduino and Ethernet (we don't have anything about ESP32 and Ethernet)
https://randomnerdtutorials.com/arduino-ethernet-web-server-with-relay/
Regards,
Sara

Reply

Maria Müller
January 23, 2019 at 12:34 am

Diese Seite ist ja für die Ausländer, wo ist den die DEUTSCHE Seite???????????????

Reply

Sara Santos
January 23, 2019 at 2:27 pm

Der Inhalt unserer Website ist nur in englischer Sprache

Reply

KingC
February 20, 2019 at 3:57 pm

Hmm.. I get an IP in Seriel Monitor, but I can not Connect to it. Can not find that IP in my router or in Fing eighter! What am I doing wrong.

Reply

Sara Santos
February 20, 2019 at 5:17 pm

Hi.
Without any further details, it is very difficult to understand what might be wrong.
Can you provide more details?
Regards,
Sara

Reply

KingC
February 20, 2019 at 7:22 pm

I can try ☺

I start Programming the ESP32 With Arduino IDE!

Programming OK.
Serial Monitor tells me the IP adress
Try conecting to IP via browser but only get's Time out!

CMD in Windows and try to ping the board

Pinging 10.99.10.226 with 32 bytes of data:
Request timed out.
Reply from 10.99.10.2: Destination host unreachable.
Reply from 10.99.10.2: Destination host unreachable.
Reply from 10.99.10.2: Destination host unreachable.

Times out or unreachable

Also tried With static IP

Reply

AndyB
July 30, 2019 at 2:05 pm

KingC – I have the same issue. Did you resolve your problem… If so, how?

Reply

Constantin
February 21, 2019 at 9:49 pm

Hey, nice tutorial, but a very important part would be: at every button push, to redirect the browser to the root, so in case of refresh, the switch will not be re-activated by mistake. Is there a way to do this?
Thanks

Reply

Sara Santos
February 22, 2019 at 6:21 pm

Hi Constantin.
To prevent the switch to be reactivated, you can add an if statement on your code checking the current state of the switch, and if it is the same, don't do anything.
When you click the button, you are redirected to the /26/on url to make the request. So, I don't know how to redirect the browser to the root to prevent making unnecessary requests. It should be possible, but I'm not familiar with that subject.
Regards,
Sara

Reply

KingC
February 23, 2019 at 11:24 am

Is there a way to show an image from an URL in this Project?

I would like to show a image from a surveillance camera beneath the buttons!
The URL would be something like: http://someurl/cgi-bin/snapshot.cgi

Reply

Jaime
March 7, 2019 at 9:33 pm

Hi, first of all I would like to thank you for this awesome tutorial.
Im just starting to work with the ESP32 itselft along with HTML and CSS . I understand that in order to style and build the web page, "in line style" was used but i don't quite understand the syntax. I assume that it as to do with the Wifi library right? do you have any recommendations to where should i go in order to get a better understaning of how i should build my custom webpage using the same library?

Best regards from Portugal!
Jaime

Reply

Sara Santos
March 9, 2019 at 12:12 pm

Hi Jaime.
I agree with you that it is a bit tricky to integrate the CSS and HTML with the library we're using.
Basically, what you should do is:

– compact your HTML (that should include the CSS) text in some lines, while being readable at the same time

– add a backslash (\) before every double quote (the HTML text has double quotes. To send those double quotes to the client, without interfering with the double quotes of the client.println(""), you need to add an escape character called backslash (\) before the double quotes)

– use the client.println() function to send the HTML text to the client.

Alternatively, you can compact all your HTML+CSS in a String variable and sent it all at once to the client. However, it will be more difficult to handle variables that change like the state of the output.

There's also the alternative to load all your HTML and CSS from SPIFFS and use an asynchronous web server: https://randomnerdtutorials.com/esp32-web-server-spiffs-spi-flash-file-system/

I hope this helps and thank you for your interest in our work.

Regards,
Sara

Reply

Jefferson Guedes Dias
March 8, 2019 at 11:50 pm

Hello Rui.

I'm having trouble when I connect with the IP address I receive in my IDE.

When I put the IP in the browser (Chrome, Firefox), the following message appears:

html {font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;} .button {background-color: # 4CAF50; border: none; color: white; padding: 16px 40px; text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;} .button2 {background-color: # 555555;} ESP32 Web Server GPIO 26 – State off ON GPIO 27 – State off ON

What could be wrong?

Thank you very much.

Reply

Sara Santos
March 9, 2019 at 11:58 am

Hi Jefferson.
I'm sorry you're having issues with the project.
Are you using our exact code, or did you make any modifications?

Reply

Sara Santos
March 9, 2019 at 12:25 pm

I found that we had an issue displaying the code on the website.
Everything should be fixed now.
Can you copy the code again? https://raw.githubusercontent.com/RuiSantosdotme/ESP32-Course/master/code/WiFi_Web_Server_Outputs/WiFi_Web_Server_Outputs.ino
Regards,

Reply

Celso Ferrarini
May 16, 2019 at 12:34 am

Excelente!!!
Muito útil.

Reply

victorfdez
May 21, 2019 at 4:40 pm

Hi, thx por your post. I tried your code and works very well. If I understood correctly, we connect ESP to our router, which provides an internal IP, then we connect another device to ESP throught our router. It is possible to do this without router? I mean, asigning an ip statically to ESP and then connect other device to ESP directly, without router.
Thx in advance

Reply

Sara Santos
May 22, 2019 at 11:16 am

Hi Victor.
Yes, you can do that. You can set the ESP32 as an access point.
Read this tutorial: https://randomnerdtutorials.com/esp32-access-point-ap-web-server/
Regards,
Sara

Reply

victorfdez
May 22, 2019 at 3:40 pm

Thx! This is exactly what I need.

Reply

victorfdez
May 22, 2019 at 3:42 pm

I'm pretty new on ESP32 world. I have another question, It is possible to use ESP32 as proxy between a station and the router? I mean, every time the station (mobile, pc...) wants to reach internet, every package it wants to send to the router, it must past through the ESP32 first. Something like: station->ESP32->router->internet->router->ESP32->station. Routing tables whould be necessary?
Thx again for your time and patience.

Reply

Keith Guillotte
May 25, 2019 at 9:56 pm

Hello! I followed your instructions precisely... everything I did produced the exact outcome you documented. As the last step, I enabled the sketch but fail to get a connection. I get as far as "Connecting to mySSID....." and then I just get an endless number of dots. I'm currently using a Mac OS with the Arduino IDE. I specifically took your advice and purchased an ESP32 DEVKIT DOIT board. I'm at a complete loss as to how to proceed. Up until the point where I would have viewed my board's IP address on the Serial Monitor, everything had gone perfectly. Any thoughts?

Reply

Sara Santos
May 27, 2019 at 5:53 pm

Hi Keith.
That message with infinite dots, usually happens when one forgets to insert the network credentials on the code, or the network credentials are incorrect.
Please make sure that you have your network credentials inserted and without typos.
Regards,
Sara

Reply

eze
June 3, 2019 at 3:05 pm

Is it necessary to use the LEDs or can it be done without them?, And now it is being used and is looking for a server to communicate with an android application to make a gps

Reply

Sara Santos
June 4, 2019 at 3:43 pm

I'm sorry, but I didn't understand your question.

Reply

Ian Paterson
June 12, 2019 at 2:09 pm

Have been experimenting with the ESP 32 TTGO. Uploaded this example and you have given me lots to think about regarding talking to oled display and further capabilities.

Excellent code, concise instruction, loaded and running first attempt!

Thank You ☺

Reply

Sara Santos
June 13, 2019 at 11:14 am

Thank you for following our work.
Regards,
Sara

Reply

Eric Mills
July 26, 2019 at 10:25 pm

Rui thank you for a very complete tutorial. I used your project to remote control an old police lightbar that I converted to all LED. I used a pmos circuit to switch the higher amperage circuits and it all works great! I have 6 different function switches. I have video if you want.

Reply

Sara Santos
July 27, 2019 at 11:28 am

Hi Eric.
That would be great.
Just send us an email saying you want to share your video and we'll get back to you.
Use our contact page: https://randomnerdtutorials.com/support/
Thank you.
Regards,
Sara

Reply

Gaver
August 8, 2019 at 3:46 pm

Great tutorial…
I'm curious – can the 32 be a standalone webserver – meaning – no router / local network required?
I'd like to be able to connect to one using my cell phone without having to authenticate thru a wifi network.
Hopefully there's a way where if you know the IP address – you could connect without requiring a router…

Reply

Sara Santos
August 8, 2019 at 4:09 pm

Hi Gaver.
Yes, you can do that.
You just need to set the ESP32 as an access point.
See our tutorial about setting the ESP32 as an access point:
https://randomnerdtutorials.com/esp32-access-point-ap-web-server/
Regards,
Sara

Reply

gaver
August 8, 2019 at 5:20 pm

Thanks Sara for your quick and informative reply!
Very helpful...

Reply

twinclouds
August 8, 2019 at 5:35 pm

It works fine. I am just wondering it it is possible to add password protection of the server as I am intend to use it over the internet by forwarding its localnet port. If it can be done, it will be much more useful.
Thanks for your good work.

Reply

Shashi Kiran
August 16, 2019 at 6:04 am

Can the ESP32 have multiple clients ? (not the one with browser, and a smartphone on another)

The ESP32 should be able to receive requests from a browser or SmartPhone on one end, and also other ESPxx “STA”tions at the other end.

I want to be able to connect multiple ESP8266 configured as STAtions and sending temperature/humidity/water flow etc to your article's ESP32 configured as SoftAP and in turn this ESP32 will render the charts on a web server.

Essentially, ESP32 it will act as a AP to the end point STAs, and in turn act as an AP to the browser.

The next question is : Can this ESP32 be configured to talk to a Home Wireless router so that it can send data to the cloud after it gets data from multiple STAs.
Thanks

Reply

Tomas
September 8, 2019 at 9:00 am

Hello!
I have problem with my serial monitor-only response from my esp32 is something like this: ؟ @$'l}؟؟؟ì؟f2؟=9؟/؟؟yS؟
On another sites i read that this is because i have a wrong board configuration.
My board is esp32 devkit v1 and all settings I have same like you.
Maybe its because i have a 30 pin module and not 36?
Thanks

PS sorry for my english

Reply

Sara Santos
September 8, 2019 at 4:23 pm

Hi Tomas.

That is probably due to wrong baud rate on the Serial Monitor.
In the Serial Monitor, on the bottom right corner, select the 115200 baud rate.

That has nothing to do with the number of pins.

Regards,
Sara

Reply

Bernard Lheureux
October 4, 2019 at 12:12 pm

Hello Sara and Rui,
I'm trying to combine lora receiver and web server to publish LoRAData.
I succeed to get LoRAData OR web server BUT never both at the same time ;-).
If I receive lora packets the web server doesn't answer and if I change the code the web server is answering but no more LoRA packets received. As I'm not a C/C++ guru I ask your help. Below is the code when http server is OK but LoRa is NOK.
Thanks a lot in advance.
PS : Wordfence is blocking my code. How can I sent it to you ?

Bernard

Reply

Sara Santos
October 4, 2019 at 6:25 pm

Hi.
I see from your email that you have access to our "Learn ESP32 with Arduino IDE" course.
We have something similar to what you want to build on Module 11.
The code for receiving packets and building a web server simultaneously is on Unit 3.
Access your course here: https://rntlab.com
If you have any doubts, you can use the forum for the RNTLAB members.
Regards,
Sara

Reply

Bernard Lheureux
October 5, 2019 at 4:45 pm

Hi Sara,
thanks for that, I will make a try on that.

Bernard

Reply

Marcos
October 5, 2019 at 5:14 pm

hello, could you please tell me where you got the files for the schematic diagram? thanks

Reply

Sara Santos
October 6, 2019 at 4:30 pm

Hi Marcos.
What do you mean?
We use Fritzing to design our diagrams.
Regards,
Sara

Reply

Ivan Victor
November 4, 2019 at 7:18 pm

Boa Tarde Rui,
Usei esse exemplo de projeto e funcionou muito bem. Mas estou tendo um problema com o outros aparelhos conectados ao wifi, eles estão sendo desconectados quando o ESP está conectado. Por exemplo, o meu celular fica tentando se conectar e não consegue, até que eu desligue o ESP, ai ele volta a conseguir conectar.
Se puder me ajudar agradeço.

Reply

Jean Pierre Daviau
November 21, 2019 at 1:17 am

Hi,

How can we have the time with the wifi server?

JPD

Reply

Sara Santos
November 21, 2019 at 10:01 am

Hi Jean Pierre.
You can follow this tutorial: https://randomnerdtutorials.com/esp32-ntp-client-date-time-arduino-ide/
Regards,
Sara

Reply

Robert Ranger
November 23, 2019 at 11:33 am

Hi, please don't waste any time on my question, it was my error in inputting my SSID, once corrected the program worked fine.
Many thanks for such a well explained tutorial, essential for a beginner such as myself, I look forward to following your other tutorials.
Thanks again
Bob

Reply

Sara Santos
November 24, 2019 at 12:05 pm

Hi Robert.
I'm glad you've found the issue.
Thank you for following our projects.
Regards,
Sara

Reply

JAFE RIBEIRO DE FIGUEIREDO FILHO
December 19, 2019 at 6:16 pm

Is 5 V the output voltage on the GPIOs? Thanks.

Reply

Sara Santos
December 19, 2019 at 8:07 pm

The GPIO output is 3.3V.
Regards,
Sara

Reply

Dieter Trouwborst
December 28, 2019 at 2:00 pm

Thanks for all

Very clearfull

Reply

Jean Dilissen
December 31, 2019 at 1:42 pm

works perfect, thanks for the good explanation. Just what I was looking for.

Reply

Twinclouds
December 31, 2019 at 5:41 pm

I would like to use it on internet using my DDNS and opening an port on my router. Is there any way to add password protection?

Reply

Jhonathan
March 11, 2020 at 1:18 pm

your tutorial helped me a lot in a part of my project. Thank you.

Reply

Amin
March 11, 2020 at 11:45 pm

Great tutorial! But how do I do when I want to integrate OTA updates into this code? You have a seperate tutorial* on OTA and I have tried myself to combine it with what is said here, but it was too complex for me. I guess it's because two different "systems" are used; one being WiFi.h/WiFiServer and the other WebServer.h?

Can you guide us how to make a ESP32 webserver which has OTA capability?

*https://randomnerdtutorials.com/esp32-over-the-air-ota-programming/

Reply

Samir El-Arishy
March 22, 2020 at 10:44 am

It took me less than half an hour to mod the code to accept four relays instead of the coded two. This shows how good your code is as a template for any project involves multiple pins.

I came across a funny one ….
My dev board esp32 has the pins in inverse mode. So, your code display the opposite of what the pin status on. So, after uploading the web page indicates that the pins are ON and they actually off. !!!!

I checked the comments and no one had this situation….

Here are the details of my esp32 dev board.
https://www.aliexpress.com/item/32799809384.html

I know it entails mod the code , but I wanted to check with you first

Reply

Sara Santos
March 23, 2020 at 11:21 am

Hi.
I never experimented with that board, so I'm not sure if it behaves differently than my ESP32.

Having the pins in inverse mode can be "normal" depending on the configuration of your relays.

For example:

Normally Closed configuration (NC):
HIGH signal – current is flowing
LOW signal – current is not flowing

Normally Open configuration (NO):
HIGH signal – current is not flowing
LOW signal – current in flowing

We have another tutorial specifically for relays that you may want to take a look at: https://randomnerdtutorials.com/esp32-relay-module-ac-web-server/

Thank you for your comment.

Regards,
Sara

Reply

Zypoklos
June 12, 2020 at 7:25 pm

Hello,
thanks for the very good instructions. Especially the description of the code is very helpful

Kind regards from Germany

Reply

Sara Santos
June 16, 2020 at 8:44 pm

Thanks 😀

Reply

Ron Brinkman
June 15, 2020 at 5:29 pm

The project works as advertised – Thanks for your good work!

I suggest you continue to link to your web page entitled [SOLVED] Failed to connect to ESP32: Timed out waiting for packet header at https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/ because I experienced this problem when loading this project.

In addition, if I leave the project running overnight it quits working. I either have to press the EN button or reload the sketch to get it working again. Since I will ultimately want to deploy my end application remotely I will want an implementation that does not require physical access. Is there another fix for the need to push the EN button?

Reply

Sara Santos
June 16, 2020 at 10:37 am

Hi Ron.
Follow the suggestion in this discussion to check for a wi-fi connection and restart if it drops:
https://rntlab.com/question/esp32-wifi-not-connecting-after-2-hours/
Regards,
Sara

Reply

Ron Brinkman
June 18, 2020 at 6:26 pm

Hi Sara,
Thanks, that did the trick. In some cases the ESP32 had to keep trying to reconnect (every 2 seconds) for two minutes before it got reconnected to WiFi. Don't know what the problem is with the router, but this bandaids it.

I suggest both of these fixes be included in each project description. I only happened to know about the first fix because I saw it in another RNT project description. I had not run on to the second fix yet.

Reply

Leave a Comment

Name *

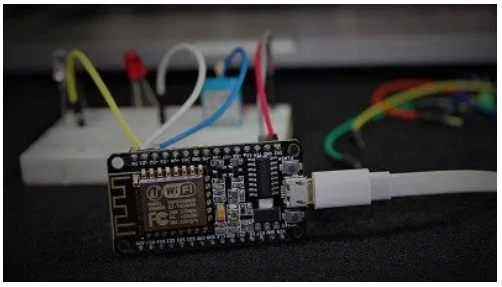
Email *

Website

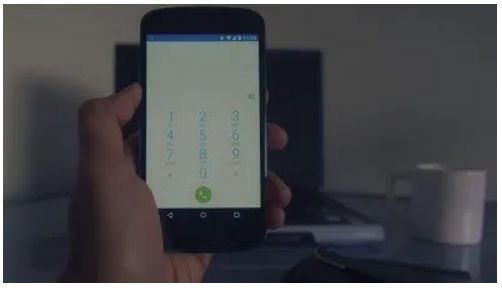
Notify me of follow-up comments by email.

Notify me of new posts by email.

Post Comment



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



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