



# **MODULE 1**

**Getting Started with ESP32**

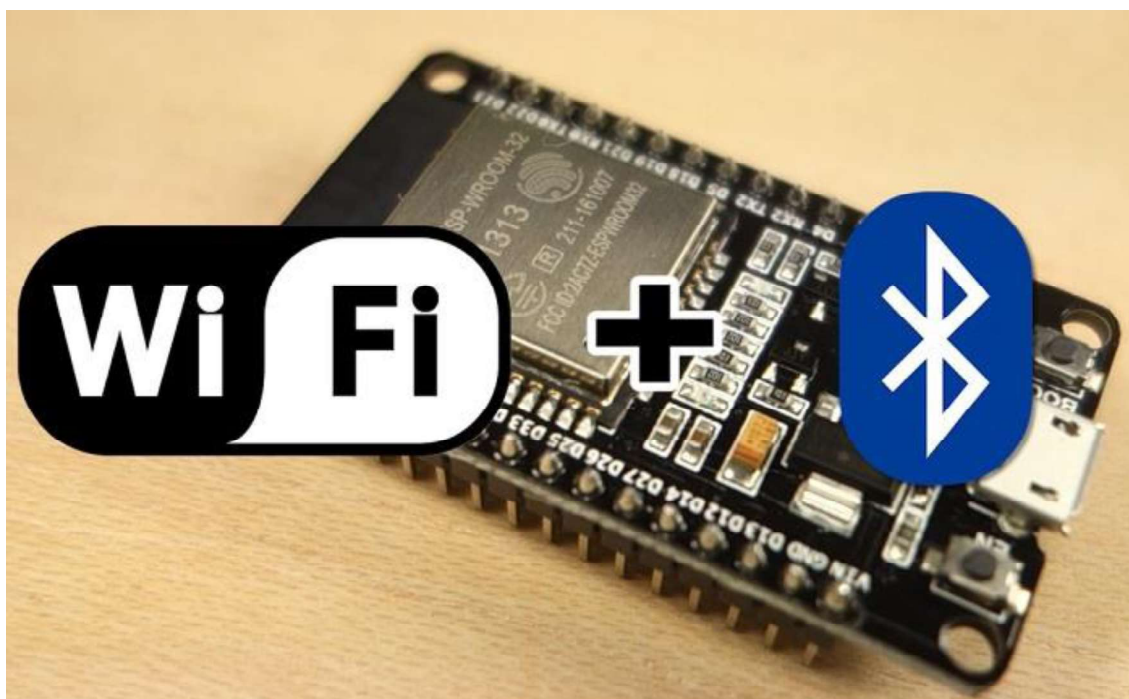
# Unit 1 - Introducing ESP32

---



## Introducing the ESP32 Board

This unit is an introduction to the ESP32 which is the ESP8266 successor. The ESP32 is loaded with lots of new features. It now combines Wi-Fi and Bluetooth wireless capabilities.



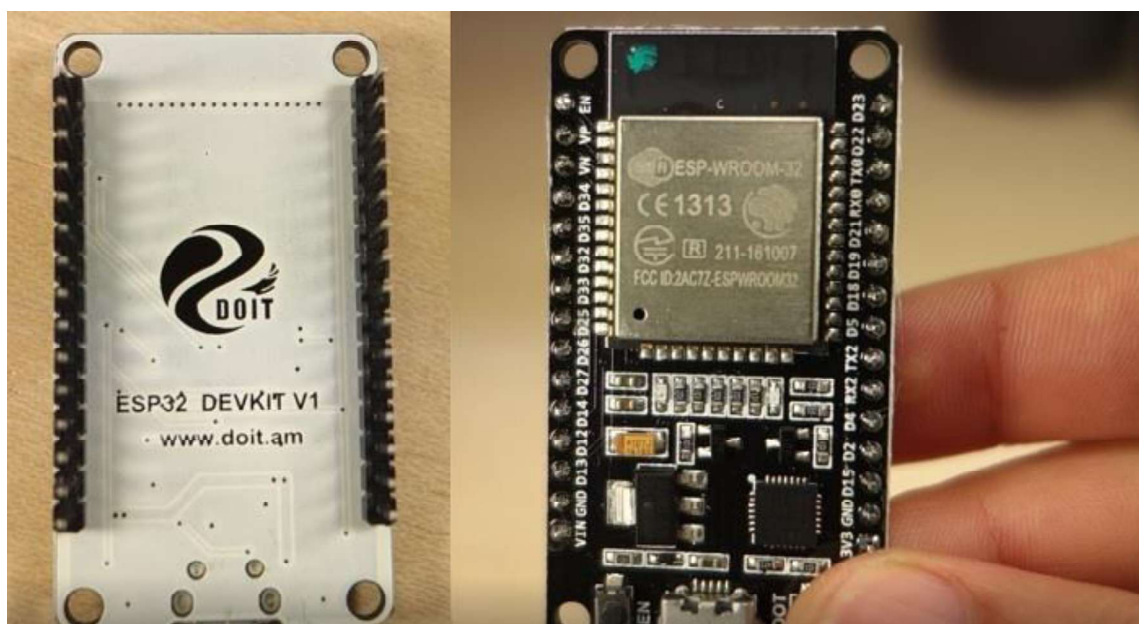
There are a lot of ESP32 development boards. I encourage you to visit the [ESP32.net website](https://www.espressystem.com/esp32) where each ESP32 chip and development board are listed. You can compare their differences and features.





## Introducing the ESP32 DOIT DEVKIT V1 Board

For this course, we'll be using the [ESP32 DEVKIT V1 DOIT board](#), but any other ESP32 with the ESP-WROOM-32 chip will work just fine.



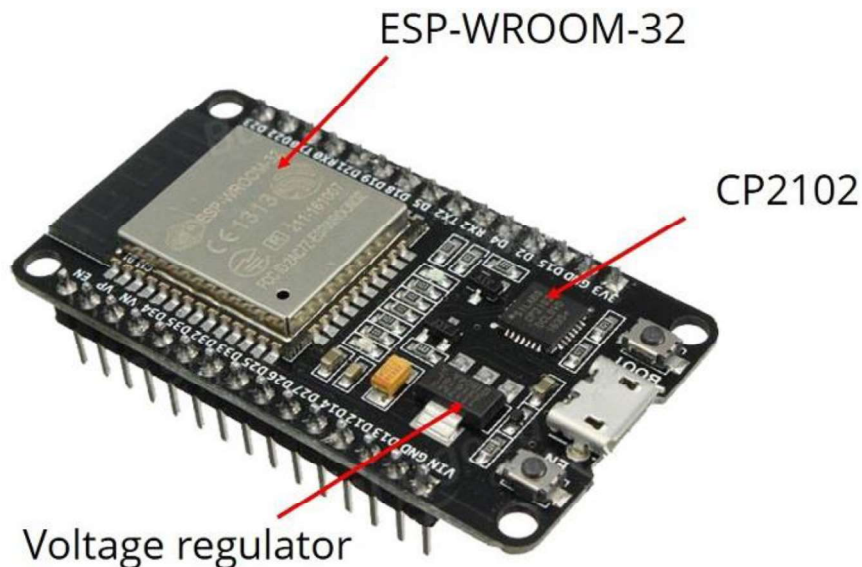
Here's just a few examples of boards that are very similar and compatible with the projects that will be presented throughout this course.



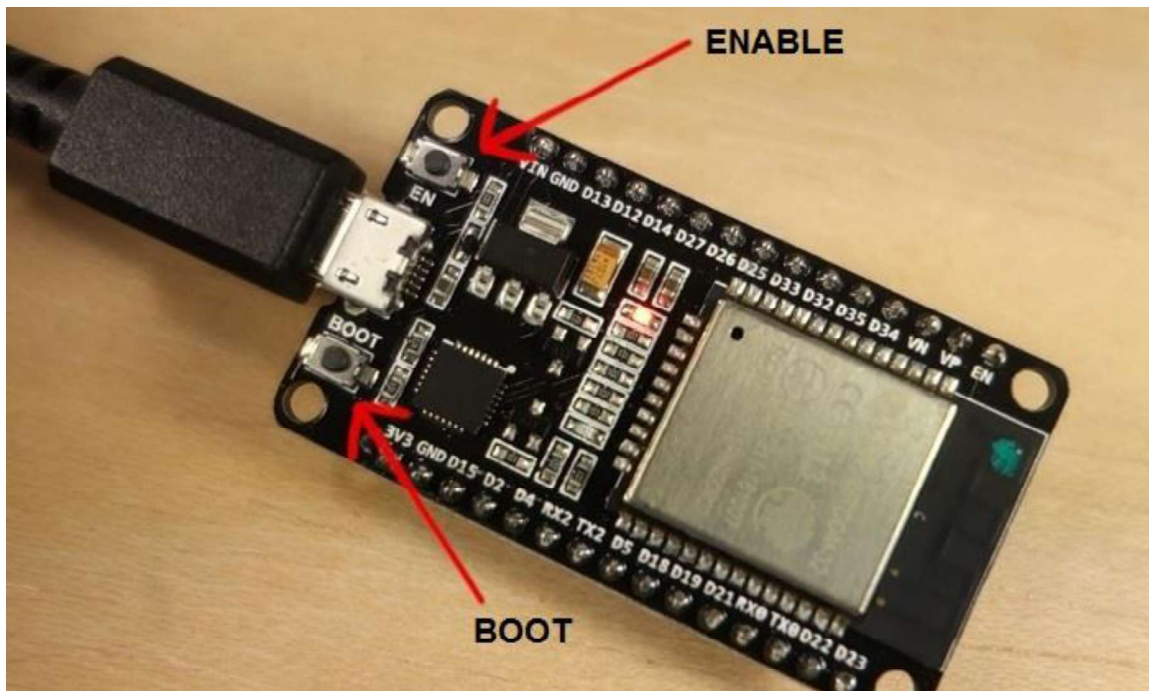
We'll provide a more in depth guide that compares the differences between the main ESP32 development boards and how to identify them. In summary, any ESP32 board should work with this course with small changes in the wiring.

## Features

Let's take a closer look at the board. It comes with the ESP-WROOM-32 chip. It has a 3.3V voltage regulator that drops the input voltage to power the ESP32 chip. And it also comes with a CP2102 chip that allows you to plug the ESP32 to your computer to program it without the need for an FTDI programmer.



The board has two on-board buttons: the ENABLE and the BOOT button.



If you press the ENABLE button, it reboots your ESP32. If you hold down the BOOT button and then press the enable, the ESP32 reboots in programming mode. You don't need to worry about these details at the moment, because we'll explore them in the next sections.



If you don't know where to get the ESP32, you can check this page on [Maker Advisor](#).

## Specifications

When it comes to the ESP32 chip specifications, you'll find that:

- The ESP32 is dual core, this means it has 2 processors.
- It has Wi-Fi and bluetooth built-in.
- It runs 32 bit programs.
- The clock frequency can go up to 240MHz and it has a 512 kB RAM.
- This particular board has 30 pins, 15 in each row. (There's a new version of this board with 36 pins.)
- It also has wide variety of peripherals available, like: capacitive touch, ADCs, DACs, UART, SPI, I2C and much more. We'll explore these functionalities later in the course.

Specifications - ESP32 DEVKIT V1 DOIT	
<b>Number of cores</b>	2 (Dual core)
<b>Wi-Fi</b>	2.4 GHz up to 150 Mbit/s
<b>Bluetooth</b>	BLE (Bluetooth Low Energy) and legacy Bluetooth
<b>Architecture</b>	32 bits
<b>Clock frequency</b>	Up to 240 MHz
<b>RAM</b>	512 KB
<b>Pins</b>	30
<b>Peripherals</b>	Capacitive touch, ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I <sup>2</sup> C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I <sup>2</sup> S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.

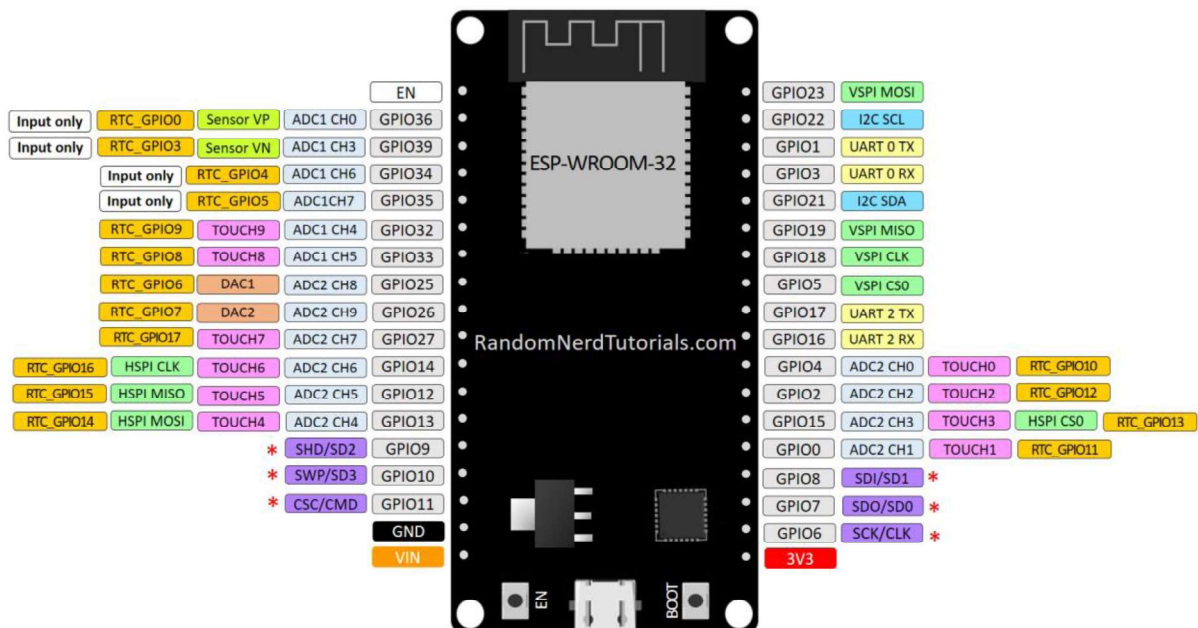
## ESP32 Pinout

The following figure clearly describes the board GPIOs and their functionalities. We recommend printing this pinout for a future reference. You can download the pinout in *.pdf* or *.png* files:

- [Printable version](#)
- [Image version 30 pins](#)
- [Image version 36 pins](#)

## ESP32 DEVKIT V1 – DOIT

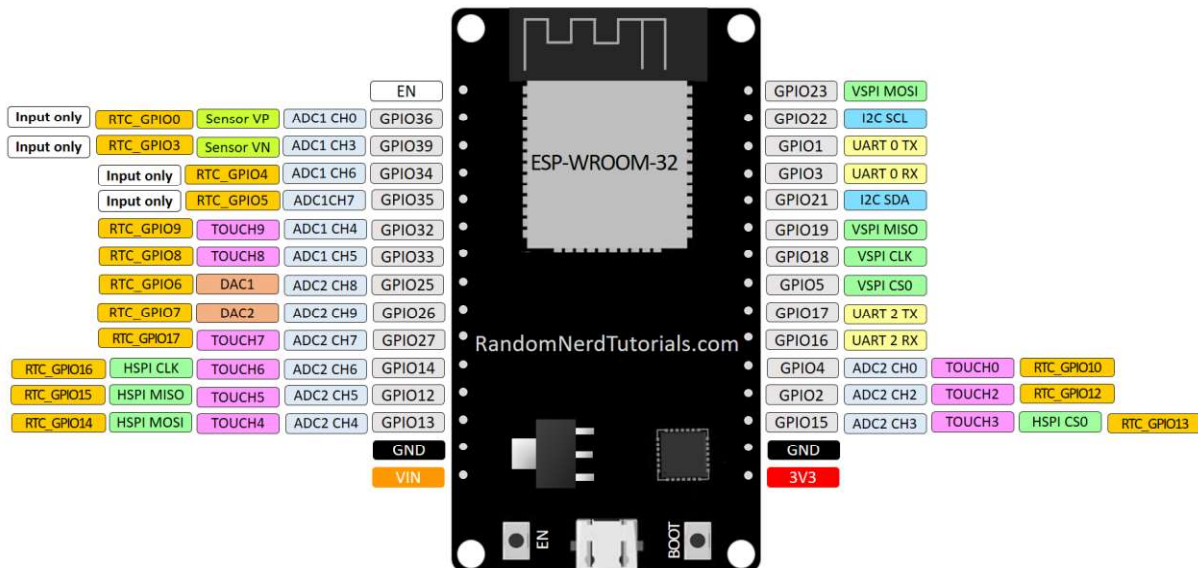
version with 36 GPIOs



\* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and CSC/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

## ESP32 DEVKIT V1 – DOIT

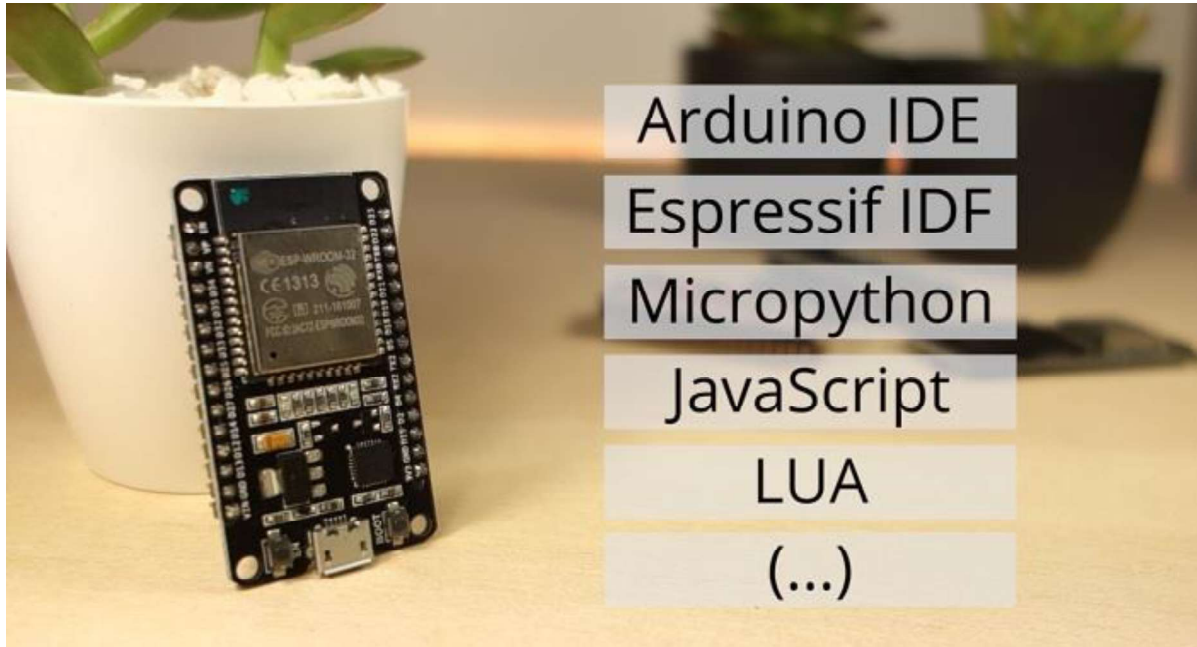
version with 30 GPIOs



**Note:** To learn more about the ESP32 GPIOs, we recommend taking a look at the [Guide for ESP32 GPIOs](#) section at the end of the eBook. You can check what are the appropriate pins for your projects and what pins you should avoid using.

## Programming Environments

The ESP32 can be programmed in different programming environments. You can use the Arduino IDE, Espressif IDF (IoT Development Framework), Micropython, JavaScript, LUA, etc. Throughout this course will be focusing mainly on programming the ESP32 with the Arduino IDE.



## Next

Go to the next section to learn how to setup the ESP32 on the Arduino IDE.

# Unit 2 - Installing ESP32 in Arduino IDE (Windows, Mac OS X, and Linux)

---

**Important:** before starting this installation procedure, make sure you have the latest version of the Arduino IDE installed in your computer. If you don't, uninstall it and install it again. Otherwise, it may not work.

The ESP32 is currently being integrated with the Arduino IDE just like it was done for the ESP8266. This add-on for the Arduino IDE allows you to program the ESP32 using the Arduino IDE and its programming language. You can find the latest Windows instructions at the official [GitHub repository](#).

## IMPORTANT NOTE:

1. If this is your first time installing the ESP32 on the Arduino IDE, simply follow the installation procedure described below.
2. If you've already installed the ESP32 add-on using the old method, you should remove the *espressif* folder first. Go to the end of this Unit to learn how to remove the espressif folder.

## 1. Installing the ESP32 Board

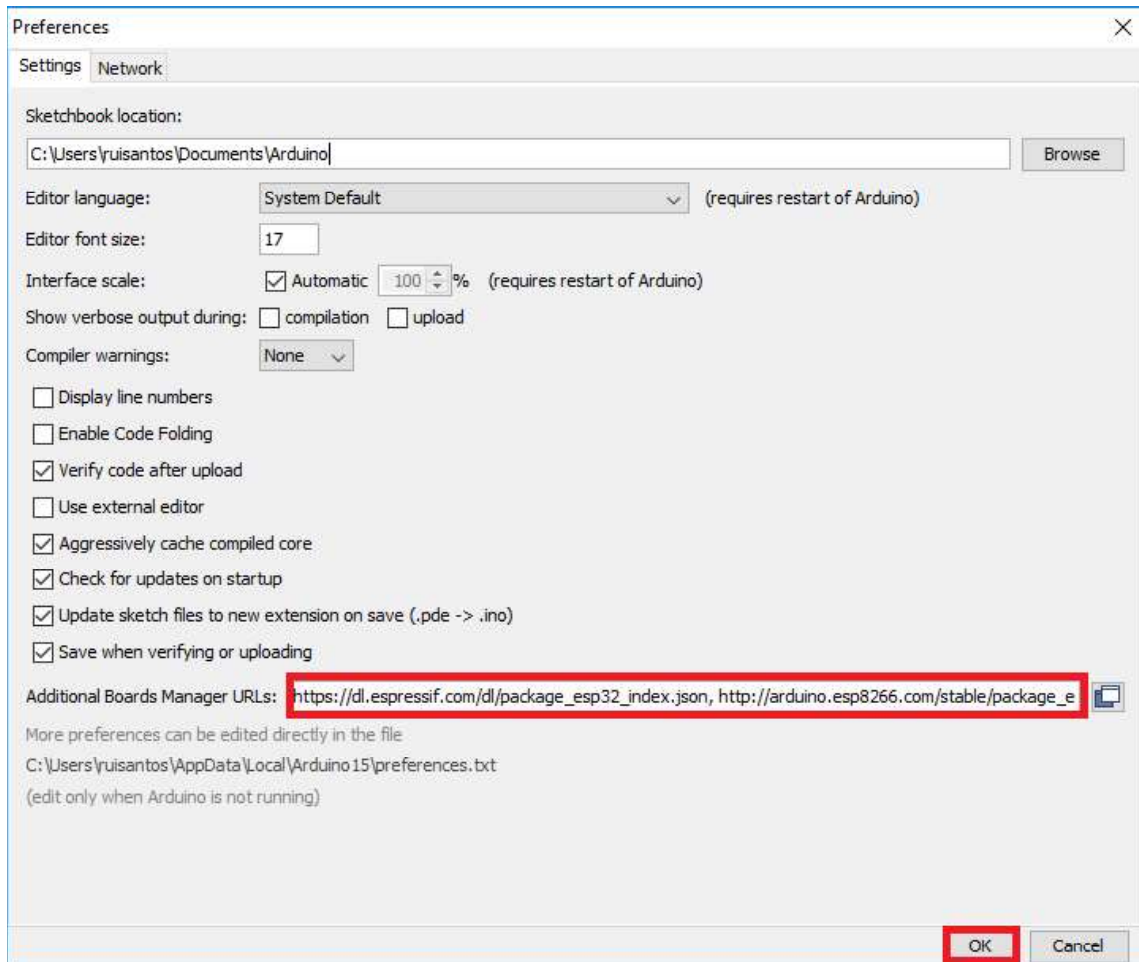
To install the ESP32 board in your Arduino IDE, follow these next instructions:

- 1) Open the preferences window from the Arduino IDE. Go to **File ▶ Preferences**
- 2) Enter **`https://dl.espressif.com/dl/package_esp32_index.json`** into the **"Additional Board Manager URLs"** field as shown in the figure below. Then, click the **"OK"** button.

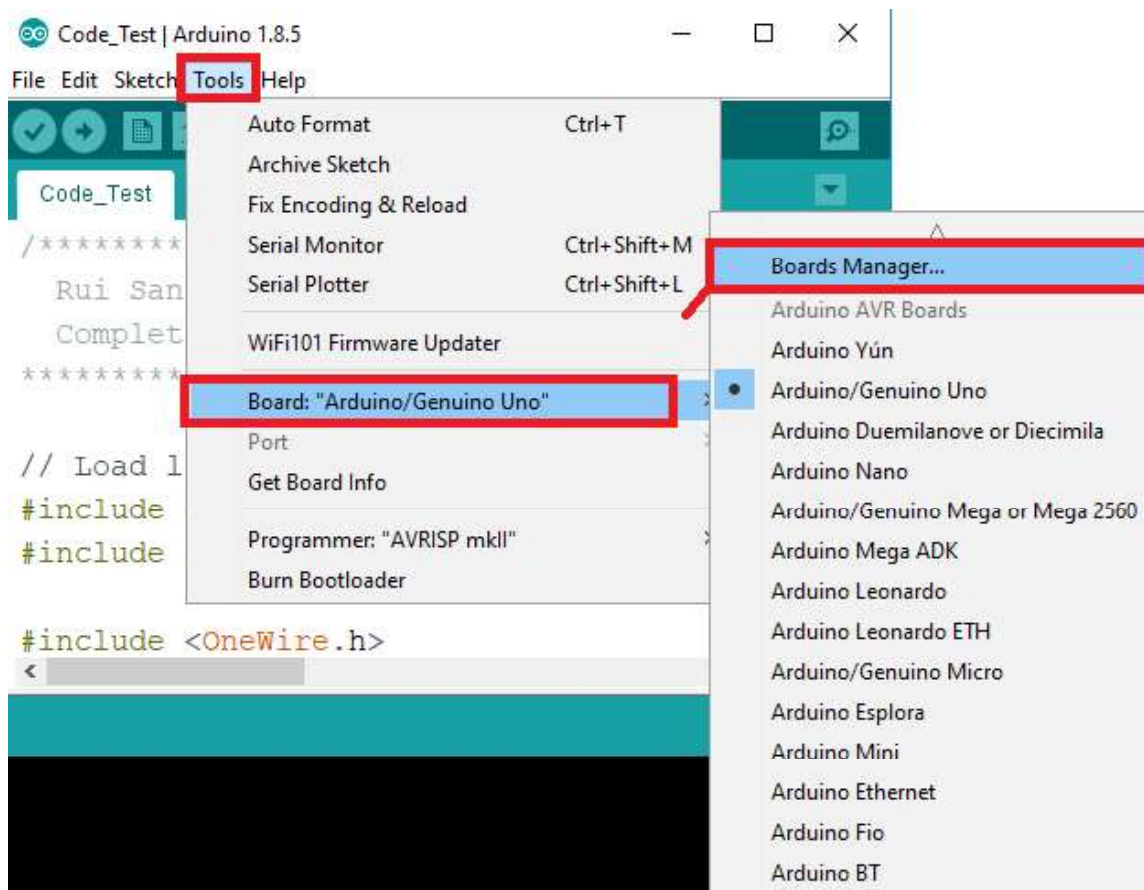
**Note:** if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

```
https://dl.espressif.com/dl/package_esp32_index.json,  
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

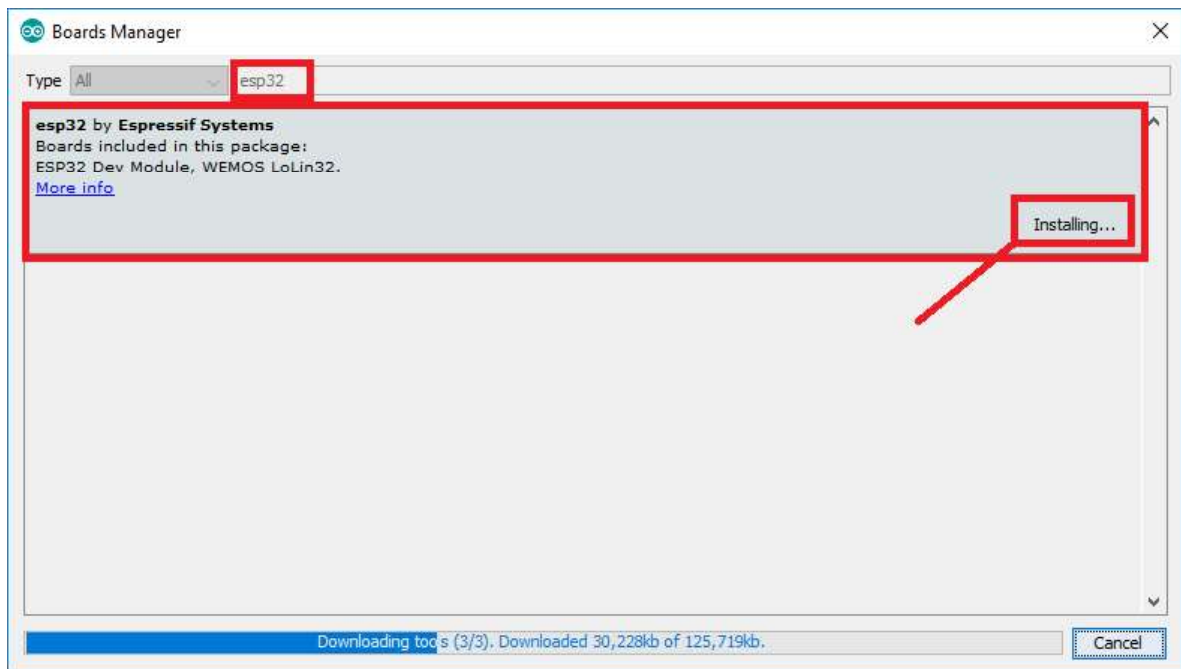




3) Open boards manager. Go to **Tools** ▶ **Board** ▶ **Boards Manager...**



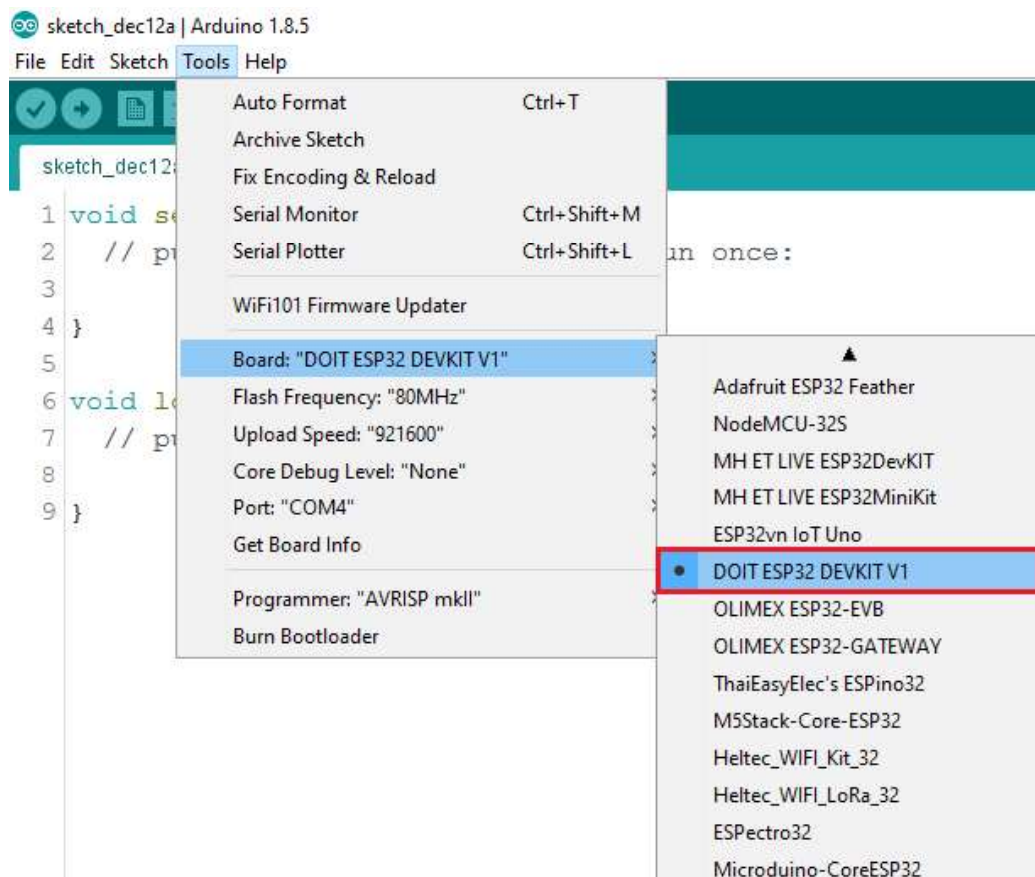
4) Search for ESP32 and press install button for the “ESP32 by Espressif Systems”:



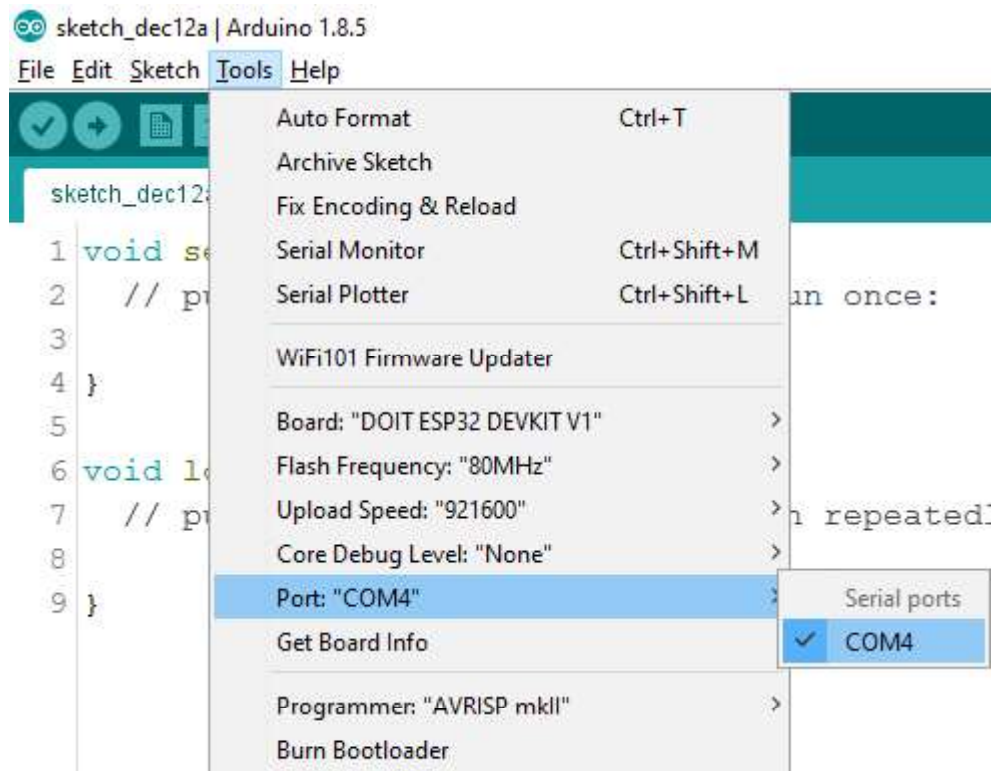
## Testing the Installation

Plug your [ESP32 DOIT DEVKIT V1 Board](#) to your computer. Then, follow these steps:

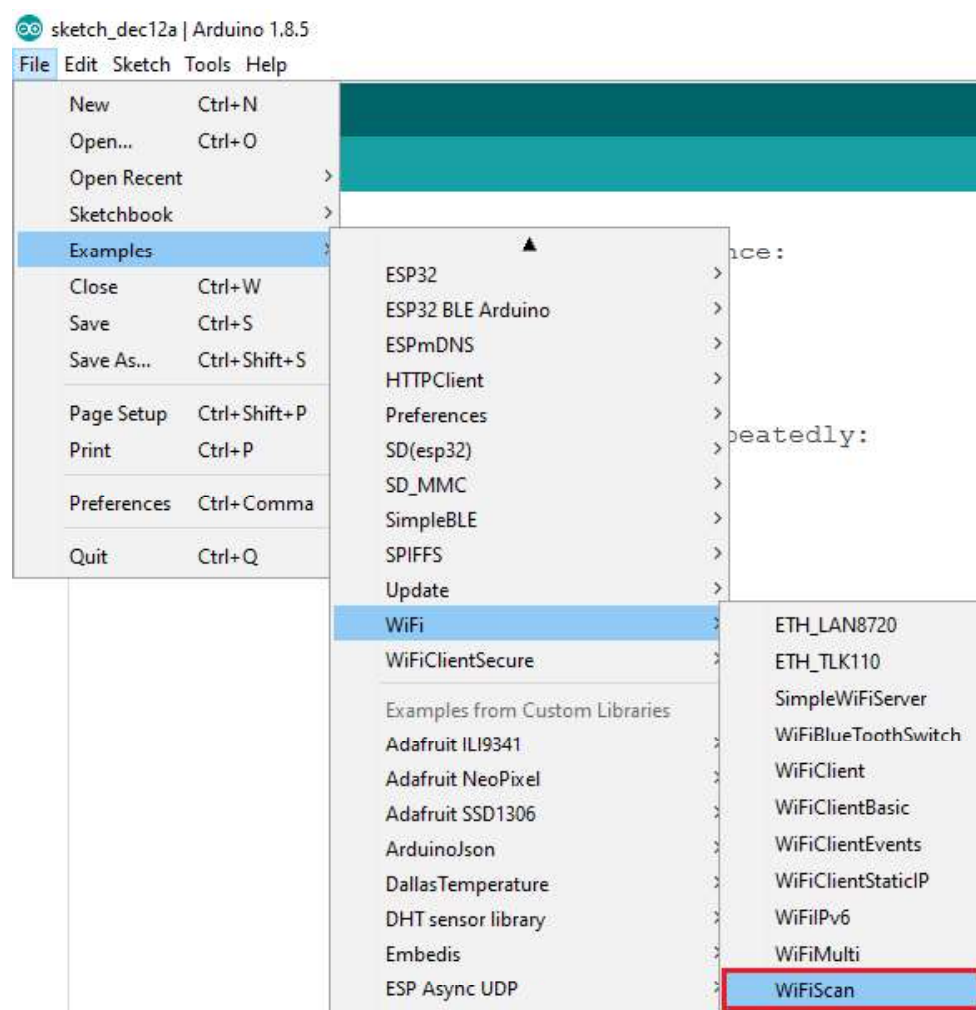
- 1) Open Arduino IDE
- 2) Select your **Board** in **Tools** ► **Board** menu (in our case it's the **DOIT ESP32 DEVKIT V1**)



3) Select the Port (if you don't see the COM Port in your Arduino IDE, you need to install the [ESP32 CP210x USB to UART Bridge VCP Drivers](#)):

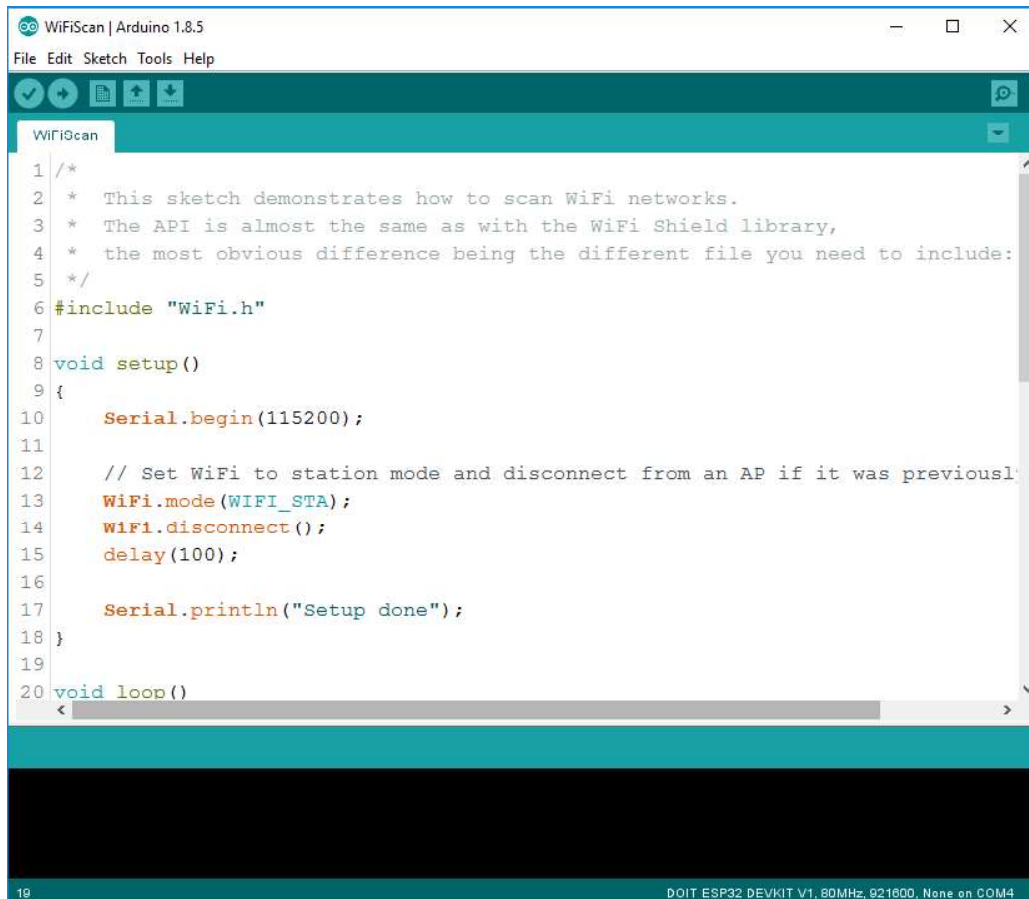


4) Open the following example under **File** ▶ **Examples** ▶ **WiFi (ESP32)** ▶ **WiFi Scan**





5) A new sketch opens:



The screenshot shows the Arduino IDE interface with a new sketch named 'WiFiScan' open. The code in the sketch is as follows:

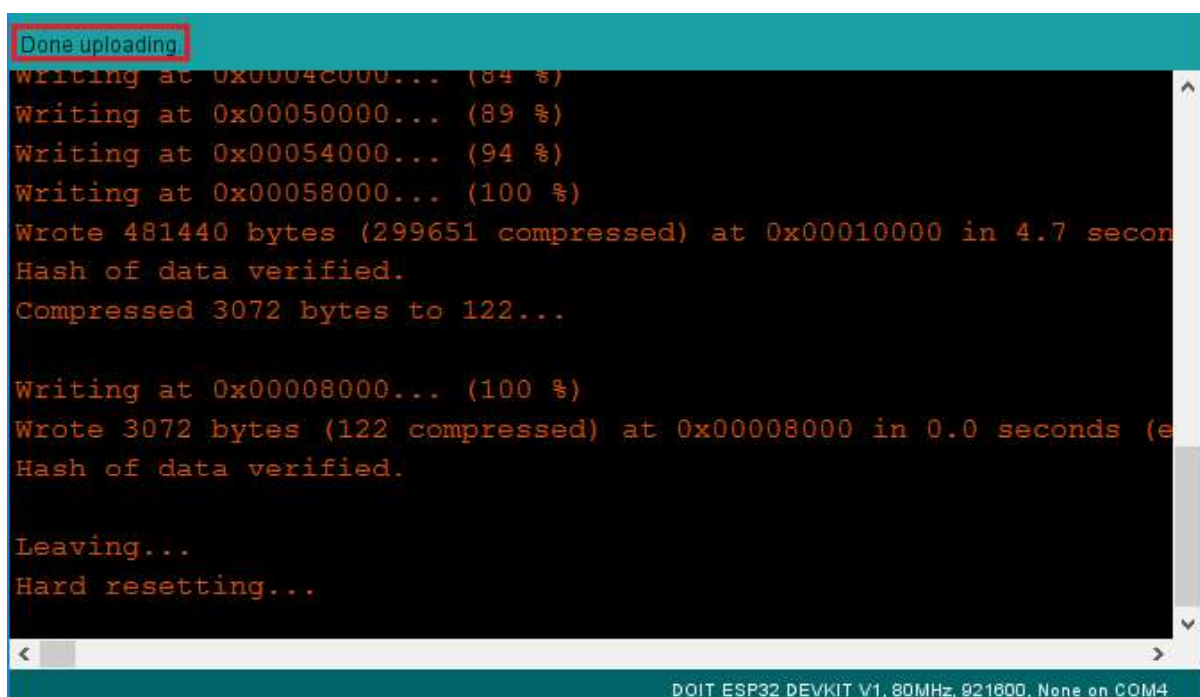
```
1 /*
2  * This sketch demonstrates how to scan WiFi networks.
3  * The API is almost the same as with the WiFi Shield library,
4  * the most obvious difference being the different file you need to include:
5  */
6 #include "WiFi.h"
7
8 void setup()
9 {
10     Serial.begin(115200);
11
12     // Set WiFi to station mode and disconnect from an AP if it was previously
13     WiFi.mode(WIFI_STA);
14     WiFi.disconnect();
15     delay(100);
16
17     Serial.println("Setup done");
18 }
19
20 void loop()
```

The status bar at the bottom indicates the board is 'DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4'.

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



7) If everything went as expected, you should see a **"Done uploading."** message.



The screenshot shows the serial monitor output of the upload process. The text is as follows:

```
Done uploading
writing at 0x0004c000... (84 %)
Writing at 0x00050000... (89 %)
Writing at 0x00054000... (94 %)
Writing at 0x00058000... (100 %)
Wrote 481440 bytes (299651 compressed) at 0x00010000 in 4.7 seconds
Hash of data verified.
Compressed 3072 bytes to 122...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (e
Hash of data verified.

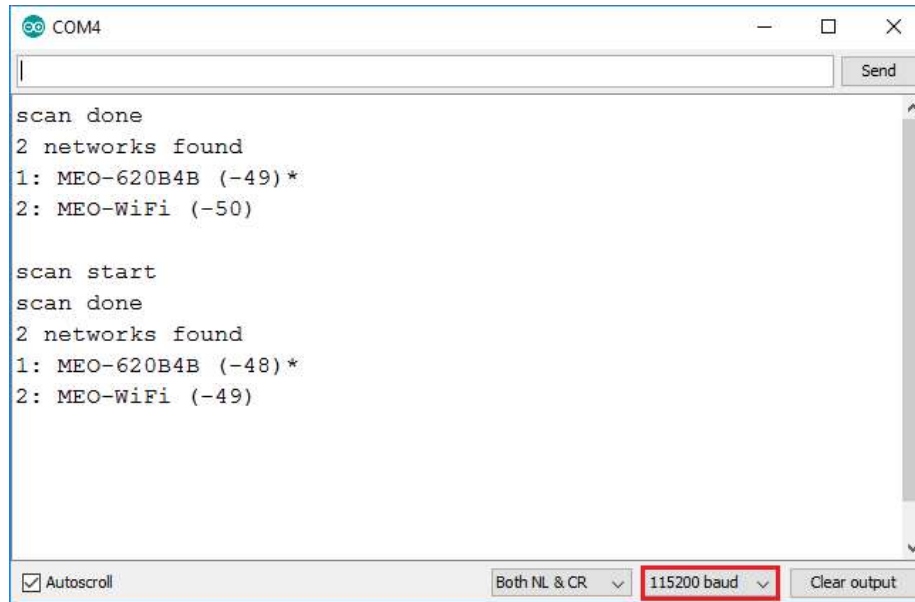
Leaving...
Hard resetting...
```

The status bar at the bottom indicates the board is 'DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4'.

8) Open the Arduino IDE Serial Monitor at a baud rate of 115200:



9) Press the ESP32 on-board Enable button and you should see the networks available near your ESP32:



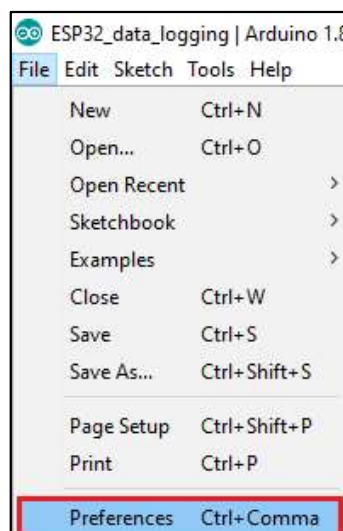
This is a very basic tutorial that illustrates how to prepare your Arduino IDE for the ESP32 on your computer.

## 2. Deleting the espressif folder

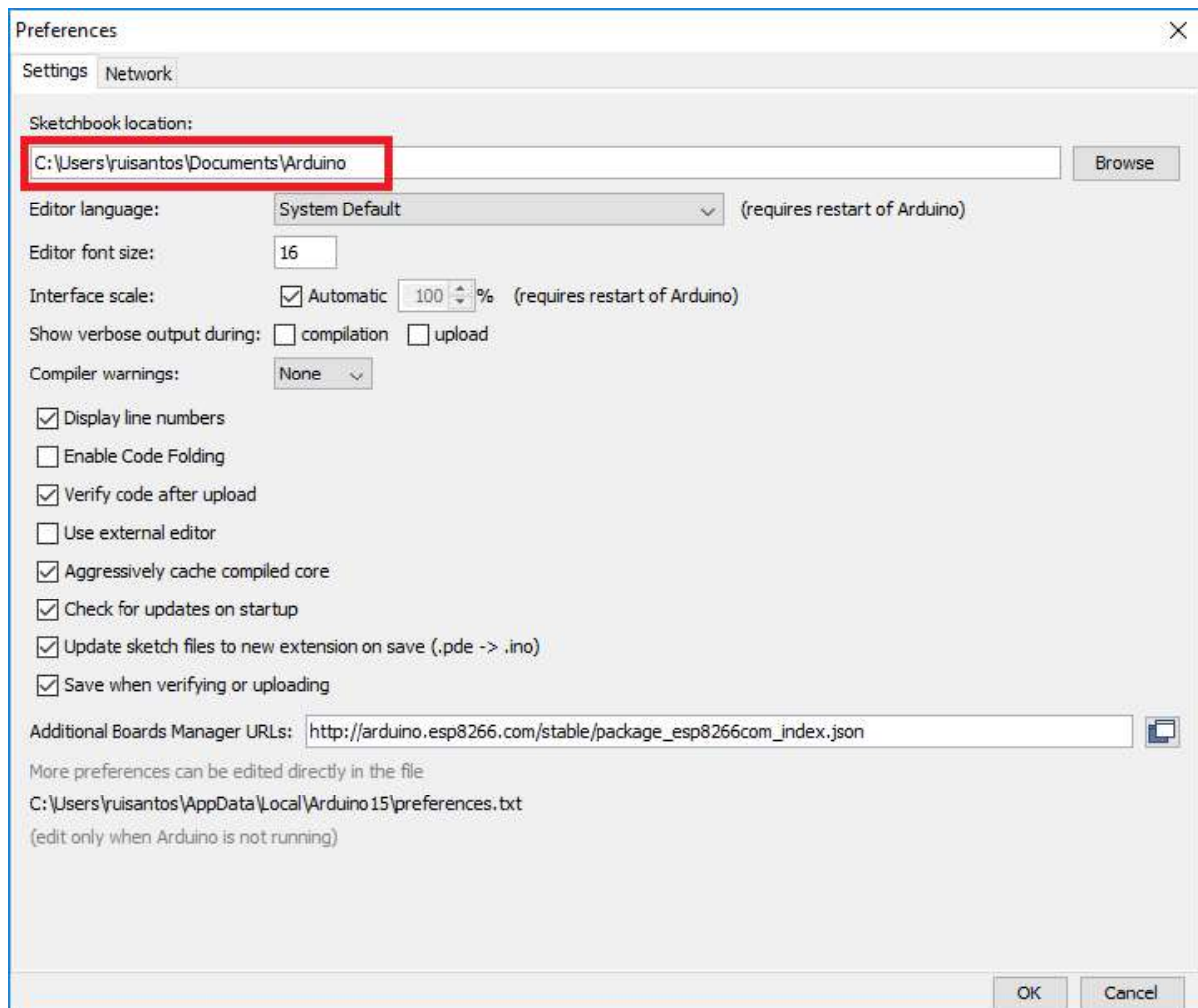
**If this is your first time installing the ESP32 on Arduino IDE you can ignore this section.**

If you've followed the older installation procedure and you've manually installed the ESP32 add-on with Git GUI, you need to remove the espressif folder from your Arduino IDE.

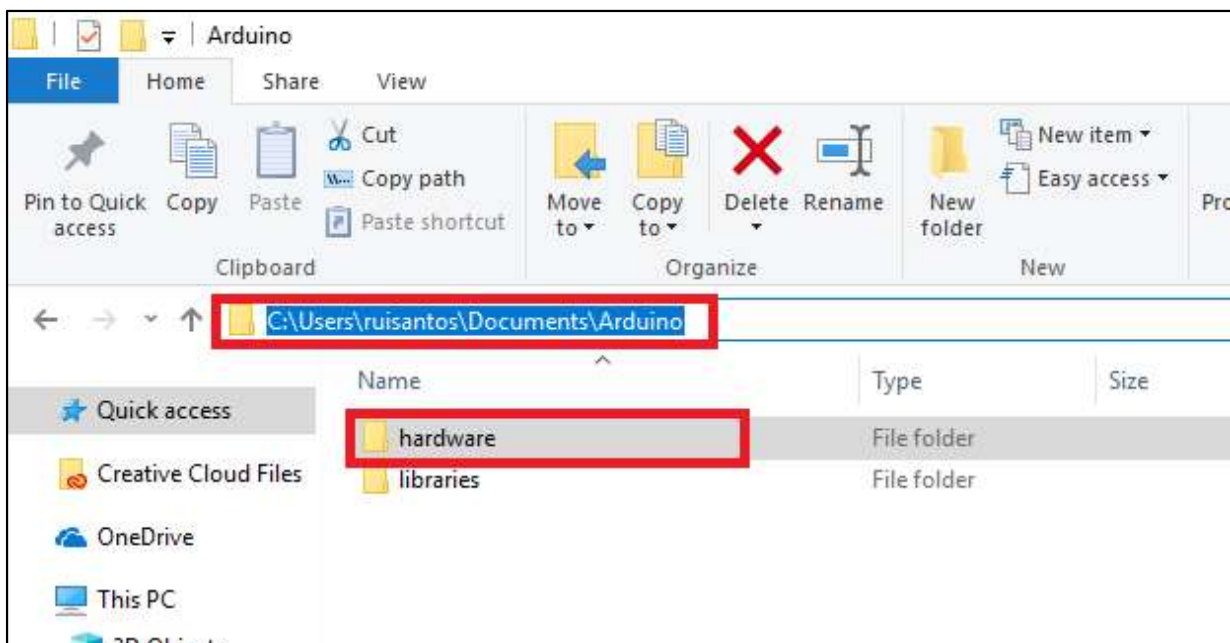
To find your espressif folder and Arduino IDE location (installation path), open your Arduino IDE and go to **File ► Preferences**:



Copy the location from the “**Sketchbook location**” field:

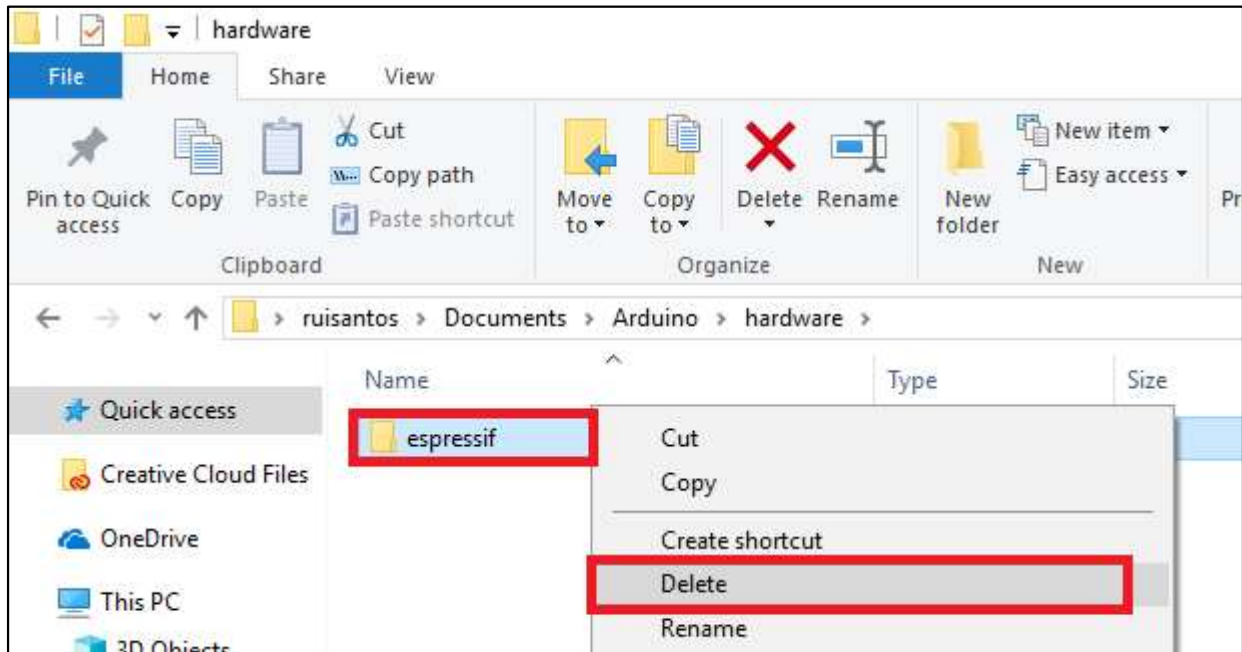


Go to your Arduino IDE location directory: `C:\Users\ruiasantos\Documents\Arduino` and open the **hardware** folder:





Then, delete the **espressif** folder:



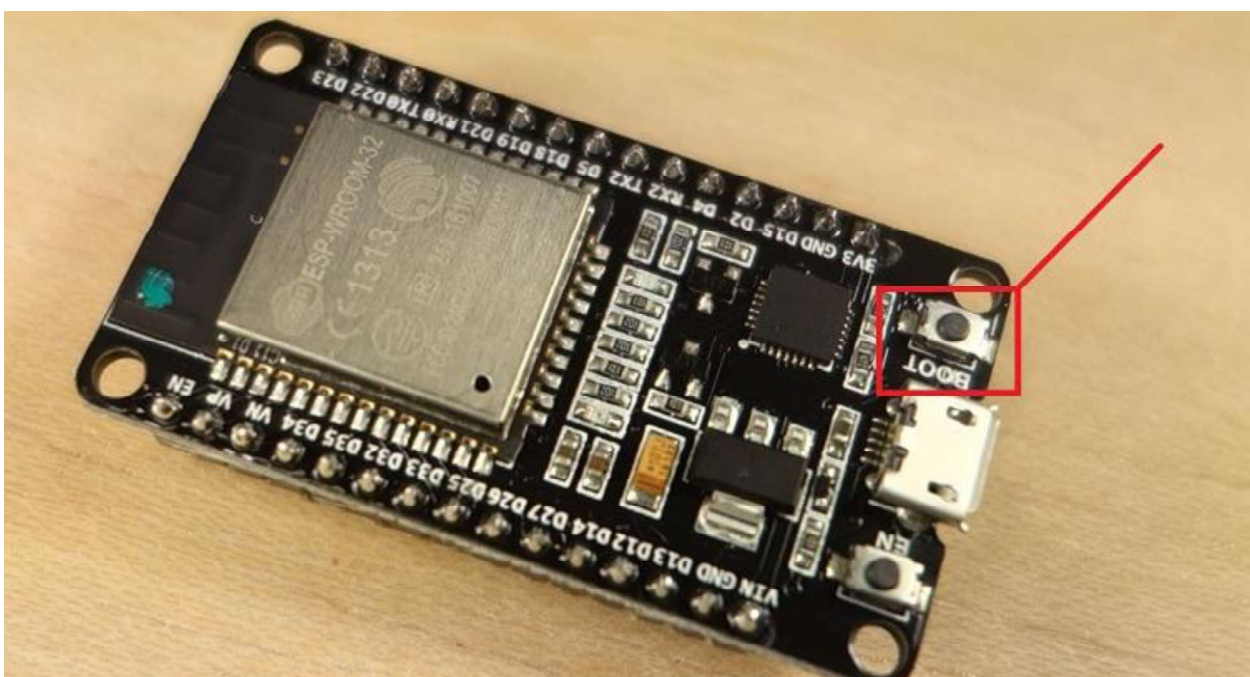
We took those screenshots using Windows 10, but a very similar procedure is done for Mac OS X and Linux.

## Troubleshooting Tip #1:

### “Failed to connect to ESP32: Timed out... Connecting...”

When you try to upload a new sketch to your ESP32 and it fails to connect to your board, it means that your ESP32 is not in flashing/uploading mode. Having the right board name and COM port selected, follow these steps:

- Hold-down the “**BOOT**” button in your ESP32 board.



- Press the “**Upload**” button in the Arduino IDE to upload a new sketch:



After you see the “**Connecting....**” message in your Arduino IDE, release the finger from the “**BOOT**” button:

```
Uploading...
Archiving built core (caching) in: C:\Users\RUIISAN~1\AppData\Local\Temp\arduino_cache_959683\c
Sketch uses 501366 bytes (38%) of program storage space. Maximum is 1310720 bytes.
Global variables use 37320 bytes (12%) of dynamic memory, leaving 257592 bytes for local varia
esptool.py v2.1
Connecting.....
Chip is ESP32D0WDQ6 (revision (unknown 0xa))
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...
```

After that, you should see the “**Done uploading**” message.

## Troubleshooting Tip #2:

### COM Port not found/not available

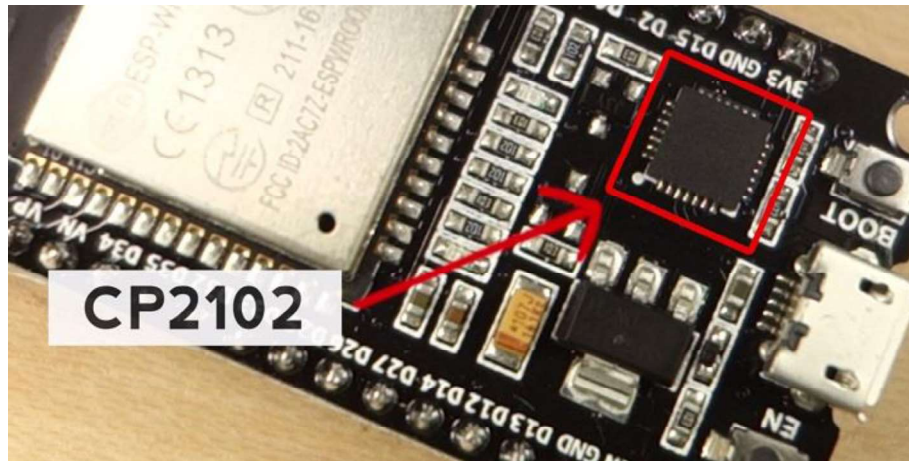
If you plug your ESP32 board to your computer, but you can’t find the ESP32 Port available in your Arduino IDE (it’s grayed out):



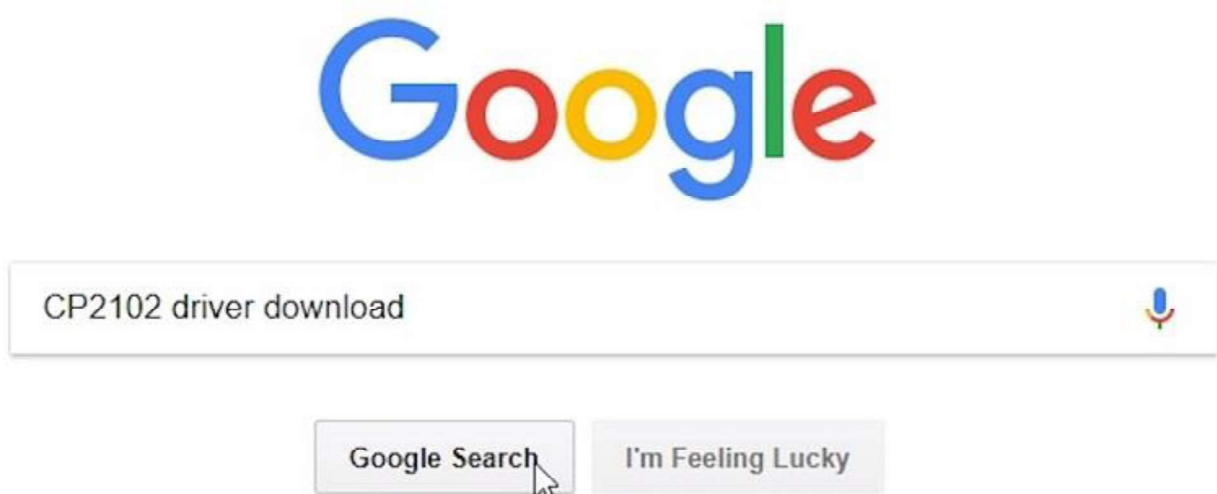
It might be one of these two problems: **1. USB drivers missing** or **2. USB cable without data wires**.

**1.** If you don't see your ESP's COM port available, this often means you don't have the USB drivers installed. Take a closer look at the chip next to the voltage regulator on board and check its name.

The [ESP32 DEVKIT V1 DOIT](#) board uses the CP2102 chip.



Go to Google and search for your particular chip to find the drivers and install them in your operating system.



You can download the CP2102 drivers on the [Silicon Labs](#) website.



## CP210x USB to UART Bridge VCP Drivers

The CP210x USB to UART Bridge Virtual COM Port (VCP) drivers are required for device operation as a Virtual COM Port to facilitate host communication with CP210x products. These devices can also interface to a host using the [direct access driver](#). These drivers are static examples detailed in application note 197: The Serial Communications Guide for the CP210x, download an example below:

[AN197: The Serial Communications Guide for the CP210x](#)

### Download Software

The CP210x Manufacturing DLL and Runtime DLL have been updated and must be used with v6.0 and later of the CP210x Windows VCP Driver. Application Note Software downloads affected are AN144SW.zip, AN205SW.zip and AN223SW.zip. If you are using a 5.x driver and need support you can download archived [Application Note Software](#).

[Legacy OS software and driver package download links and support information >](#)

### Download for Windows 10 Universal (v10.1.1)

Platform	Software	Release Notes
 Windows 10 Universal	<a href="#">Download VCP (2.3 MB)</a>	<a href="#">Download VCP Revision History</a>

After they are installed, restart the Arduino IDE and you should see the COM port in the Tools menu.

**2.** If you have the drivers installed, but you can't see your device, double-check that you're using a USB cable with data wires. USB cables from powerbanks often don't have data wires (they are charge only). So, your computer will never establish a serial communication with your ESP32. Using a proper USB cable should solve your problem.