

# Getting Started with the ESP32 Development Board

New to ESP32? Start here! The ESP32 is a series of low-cost and low-power System on a Chip (SoC) microcontrollers developed by Espressif that include Wi-Fi and Bluetooth wireless capabilities and dual-core processor. If you're familiar with the ESP8266, the ESP32 is its successor, loaded with lots of new features.



New to the ESP32? You're in the right place. This guide contains all the information you need to get started with the ESP32. Learn what is an ESP32, how to select an ESP32 board, how to get your first program working, and much more. Here's what we'll cover in this guide:

## Table of Contents

- [Introducing the ESP32](#)



- [ESP32 Development Boards](#)
  - [How to choose an ESP32 development board?](#)
  - [What is the best ESP32 development board for beginners?](#)
- [ESP32 DEVKIT DOIT](#)
- [ESP32 GPIOs Pinout Guide](#)
- [How to program the ESP32?](#)
- [ESP32 with Arduino IDE](#)
- [Upload Code to the ESP32 using Arduino IDE](#)

## Introducing the ESP32

First, to get started, **what is an ESP32?** The ESP32 is a series of chip microcontrollers developed by Espressif.



Why are they so popular? Mainly because of the following features:

- **Low-cost:** you can get an ESP32 starting at \$6, which makes it easily accessible to the general public;
- **Low-power:** the ESP32 consumes very little power compared with other microcontrollers, and it supports low-power mode states like [deep sleep](#) to save power;
- **Wi-Fi capabilities:** the ESP32 can easily connect to a Wi-Fi network to connect to the internet (station mode), or create its own Wi-Fi wireless network ([access point mode](#)) so other devices can connect to it—this is essential for IoT and Home Automation projects—you can have multiple devices communicating with each other using their Wi-Fi capabilities;
- **Bluetooth:** the ESP32 supports [Bluetooth classic](#) and [Bluetooth Low Energy \(BLE\)](#)—which is useful for a wide variety of IoT applications;



- **Dual-core**: most ESP32 are dual-core— they come with 2 Xtensa 32-bit LX6 microprocessors: core 0 and core 1.
- **Rich peripheral input/output interface**—the ESP32 supports a wide variety of input (read data from the outside world) and output (to send commands/signals to the outside world) peripherals like [capacitive touch](#), [ADCs](#), [DACs](#), [UART](#), [SPI](#), [I2C](#), [PWM](#), and much more.
- **Compatible with the Arduino “programming language”**: those that are already familiar with programming the Arduino board, you’ll be happy to know that they can program the ESP32 in the Arduino style.
- **Compatible with MicroPython**: you can program the ESP32 with MicroPython firmware, which is a re-implementation of Python 3 targeted for microcontrollers and embedded systems.

## ESP32 Specifications

If you want to get a bit more technical and specific, you can take a look at the following detailed specifications of the ESP32 (source: <http://esp32.net/>)—for more details, [check the datasheet](#)):



ESP32 module: ESP-WROOM-32

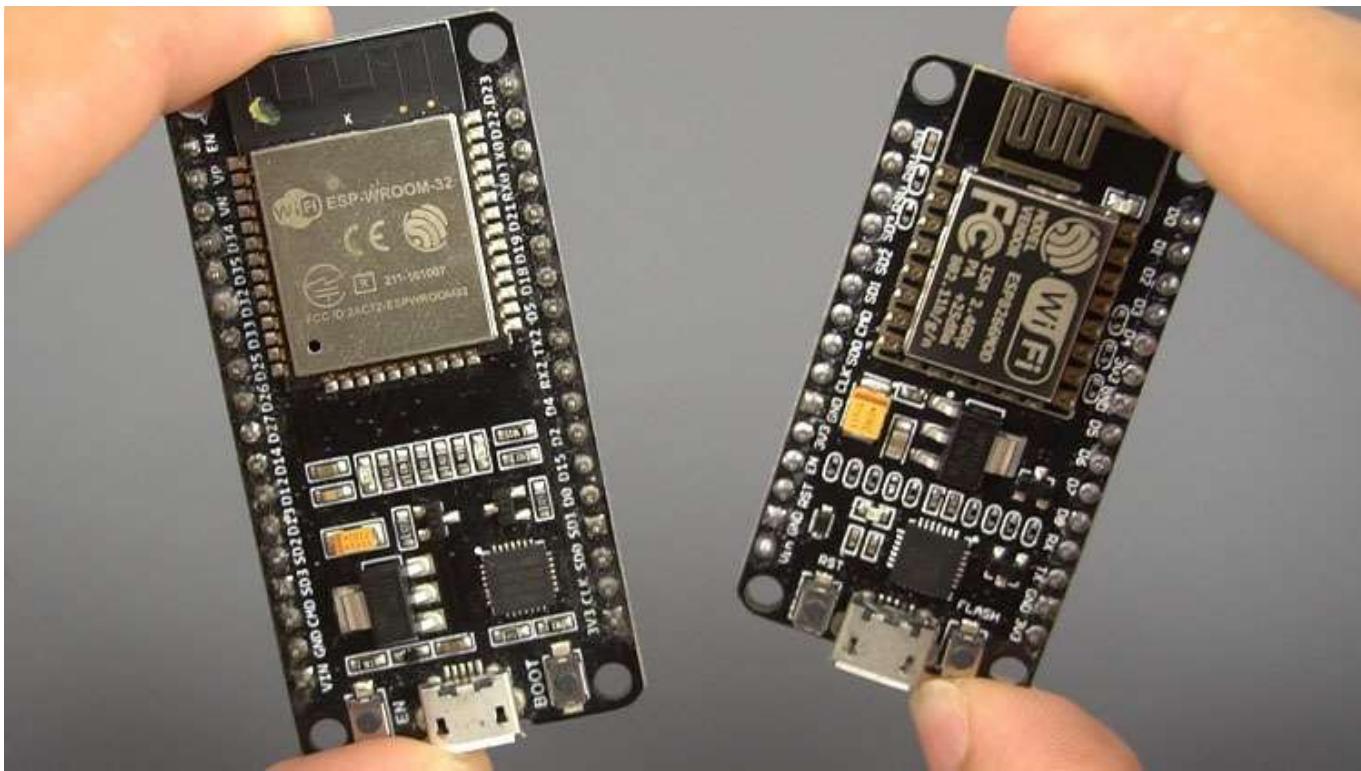
- **Wireless connectivity WiFi**: 150.0 Mbps data rate with HT40
  - **Bluetooth**: [BLE \(Bluetooth Low Energy\)](#) and [Bluetooth Classic](#)



- **Memory:**
  - **ROM:** 448 KB (for booting and core functions)
  - **SRAM:** 520 KB (for data and instructions)
  - **RTC fast SRAM:** 8 KB (for data storage and main CPU during RTC Boot from the deep-sleep mode)
  - **RTC slow SRAM:** 8KB (for co-processor accessing during deep-sleep mode)
  - **eFuse:** 1 Kbit (of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID)
  - **Embedded flash:** flash connected internally via IO16, IO17, SD\_CMD, SD\_CLK, SD\_DATA\_0 and SD\_DATA\_1 on ESP32-D2WD and ESP32-PICO-D4.
    - 0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips)
    - 2 MiB (ESP32-D2WD chip)
    - 4 MiB (ESP32-PICO-D4 SiP module)
- **Low Power:** ensures that you can still use ADC conversions, for example, during [deep sleep](#).
- **Peripheral Input/Output:**
  - peripheral interface with DMA that includes [capacitive touch](#)
  - [ADCs \(Analog-to-Digital Converter\)](#)
  - [DACs \(Digital-to-Analog Converter\)](#)
  - [I<sup>2</sup>C \(Inter-Integrated Circuit\)](#)
  - [UART \(Universal Asynchronous Receiver/Transmitter\)](#)
  - [SPI \(Serial Peripheral Interface\)](#)
  - [I<sup>2</sup>S \(Integrated Interchip Sound\)](#)
  - [RMII \(Reduced Media-Independent Interface\)](#)
  - [PWM \(Pulse-Width Modulation\)](#)
- **Security:** hardware accelerators for AES and SSL/TLS

## Main Differences Between ESP32 and ESP8266





Previously, we mentioned that the ESP32 is the ESP8266 successor. **What are the main differences between ESP32 and ESP8266 boards?**

The ESP32 adds an [extra CPU core](#), faster [Wi-Fi](#), [more GPIOs](#), and supports [Bluetooth 4.2](#) and [Bluetooth low energy](#). Additionally, the ESP32 comes with [touch-sensitive pins](#) that can be used to [wake up the ESP32 from deep sleep](#), and [built-in hall effect sensor](#).

Both boards are cheap, but the ESP32 costs slightly more. While the ESP32 can cost around \$6 to \$12, the ESP8266 can cost \$4 to \$6 (but it really depends on where you get them and what model you're buying).

So, in summary:

- The ESP32 is faster than the ESP8266;
- The ESP32 comes with more GPIOs with multiple functions;
- The ESP32 supports analog measurements on 18 channels (analog-enabled pins) versus just one 10-bit ADC pin on the ESP8266;
- The ESP32 supports Bluetooth while the ESP8266 doesn't;



For a more detailed analysis of the differences between those boards, we recommend reading the following article: [ESP32 vs ESP8266 – Pros and Cons](#).

## ESP32 Development Boards

ESP32 refers to the bare ESP32 chip. However, the “ESP32” term is also used to refer to ESP32 development boards. Using ESP32 bare chips is not easy or practical, especially when learning, testing, and prototyping. Most of the time, you’ll want to use an ESP32 development board.



These development boards come with all the needed circuitry to power and program the chip, connect it to your computer, pins to connect peripherals, built-in power and control LEDs, an antenna for wi-fi signal, and other useful features. Others even come with extra hardware like specific sensors or modules, displays, or a camera in the case of the ESP32-CAM.

## How to Choose an ESP32 Development Board?

Once you start searching for ESP32 boards online, you’ll find there is a wide variety of boards from different vendors. While they all work in a similar way, some boards



may be more suitable for some projects than others. When looking for an ESP32 development board there are several aspects you need to take into account:

- **USB-to-UART interface and voltage regulator circuit.** Most full-featured development boards have these two features. This is important to easily connect the ESP32 to your computer to upload code and apply power.
- **BOOT and RESET/EN buttons** to put the board in flashing mode or reset (restart) the board. Some boards don't have the BOOT button. Usually, these boards go into flashing mode automatically.
- **Pin configuration and the number of pins.** To properly use the ESP32 in your projects, you need to have access to the board pinout (like a map that shows which pin corresponds to which GPIO and its features). So make sure you have access to the pinout of the board you're getting. Otherwise, you may end up using the ESP32 incorrectly.
- **Antenna connector.** Most boards come with an onboard antenna for Wi-Fi signal. Some boards come with an antenna connector to optionally connect an external antenna. Adding an external antenna increases your Wi-Fi range.
- **Battery connector.** If you want to power your ESP32 using batteries, there are development boards that come with connectors for LiPo batteries—this can be handier. You can also power a “regular” ESP32 with batteries through the power pins.
- **Extra hardware features.** There are ESP32 development boards with extra hardware features. For example, some may come with a built-in OLED display, a LoRa module, a SIM800 module (for GSM and GPRS), a battery holder, a camera, or others.

## What is the best ESP32 development board for beginners?

For beginners, we recommend an ESP32 board with a vast selection of available



In most of our ESP32 projects, we use the [ESP32 DEVKIT DOIT board](#), and that's the one we recommend for beginners. There are different versions of this board with a different number of available pins (30, 36, and 38)—all boards work in a similar way.



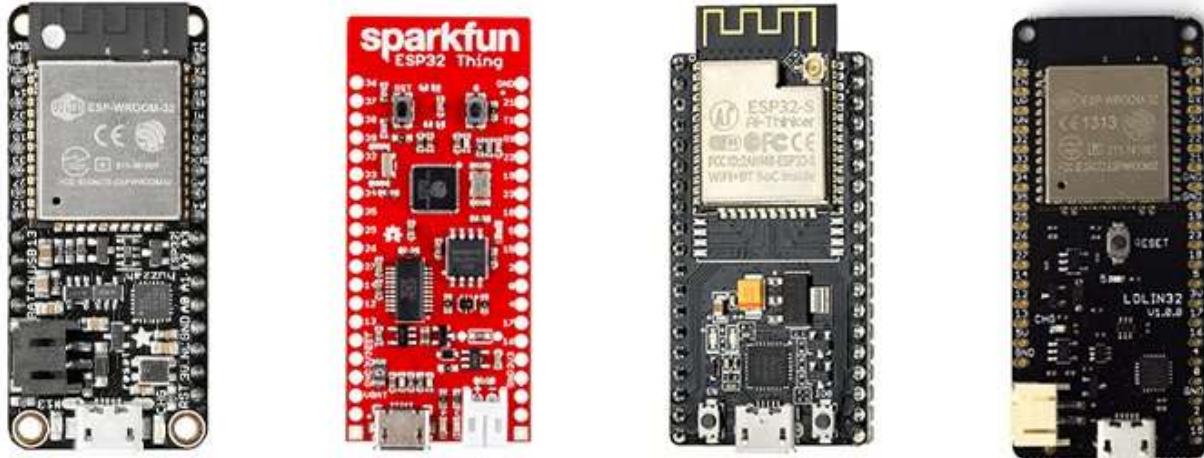
## Where to Buy?

You can check the following link to find the ESP32 DEVKIT DOIT board in different stores:

- [ESP32 DEVKIT DOIT board](#)

Other similar boards with the features mentioned previously may also be a good option like the Adafruit ESP32 Feather, Sparkfun ESP32 Thing, NodeMCU-32S, Wemos LoLin32, etc.





## ESP32 DEVKIT DOIT

In this article, we'll be using the ESP32 DEVKIT DOIT board as a reference. If you have a different board, don't worry. The information on this page is also compatible with other ESP32 development boards.

The picture below shows the ESP32 DEVKIT DOIT V1 board, version with 36 GPIO pins.



## Specifications – ESP32 DEVKIT V1 DOIT



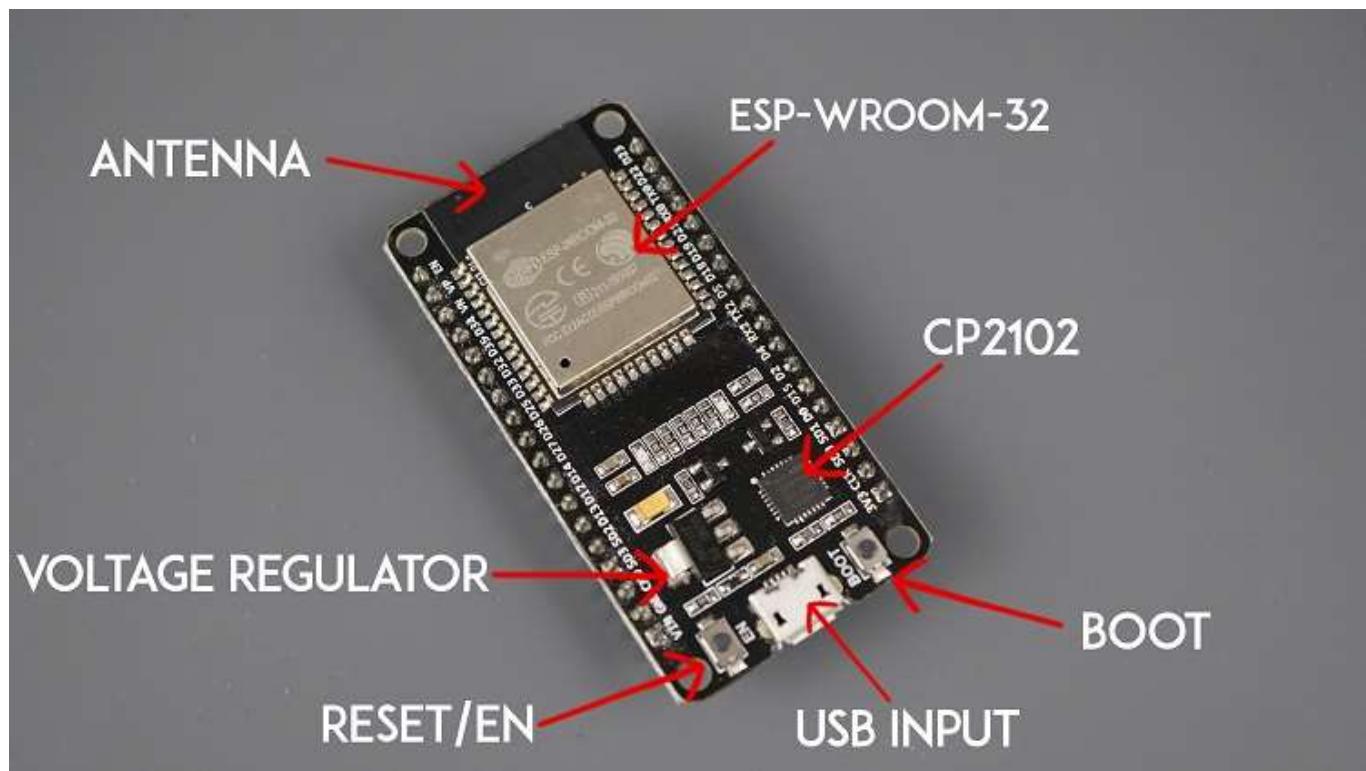
The following table shows a summary of the ESP32 DEVKIT V1 DOIT board features and specifications:

|                           |  |
|---------------------------|--|
| <b>Number of cores</b>    | 2 (dual core)  |
| <b>Wi-Fi</b>              | 2.4 GHz up to 150 Mbits/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, 36, or 38 (depending on the model)   |
| <b>Peripherals</b>        | Capacitive touch, ADC (analog to digital converter), DAC (digital to analog converter), I2C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I2S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more. |
| <b>Built-in buttons</b>   | RESET and BOOT buttons   |
| <b>Built-in LEDs</b>      | built-in blue LED connected to GPIO2; built-in red LED that shows the board is being powered   |
| <b>USB to UART bridge</b> | CP2102   |

This particular ESP32 board comes with 36 pins, 10 on each side. The number of



To learn more about the ESP32 GPIOs, read our GPIO reference guide: [ESP32 Pinout Reference: Which GPIO pins should you use?](#)



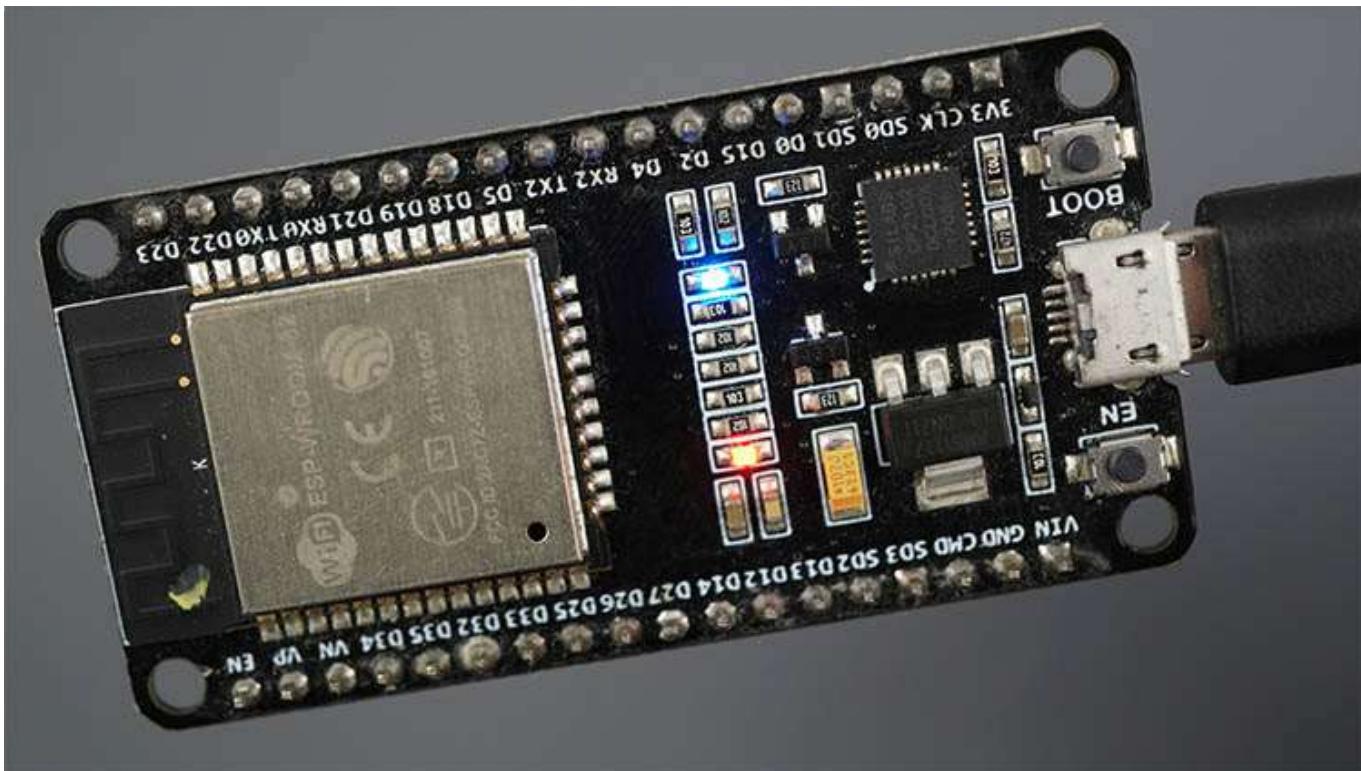
It comes with a microUSB interface that you can use to connect the board to your computer to upload code or apply power.

It uses the CP2102 chip (USB to UART) to communicate with your computer via a COM port using a serial interface. Another popular chip is the CH340. Check what's the USB to UART chip converter on your board because you'll need to install the required drivers so that your computer can communicate with the board (more information about this later in this guide).

This board also comes with a RESET button (may be labeled EN) to restart the board and a BOOT button to put the board in flashing mode (available to receive code). Note that some boards may not have a BOOT button.

It also comes with a built-in blue LED that is internally connected to GPIO 2. This LED is useful for debugging to give some sort of visual physical output. There's also a red LED that lights up when you provide power to the board.





# ESP32 GPIOs Pinout Guide

The ESP32 chip comes with 48 pins with multiple functions. Not all pins are exposed in all ESP32 development boards, and some pins should not be used. The ESP32 DEVKIT V1 DOIT board usually comes with 36 exposed GPIOs that you can use to connect peripherals.

## Power Pins

Usually, all boards come with power pins: 3V3, GND, and VIN. You can use these pins to power the board (if you're not providing power through the USB port), or to get power for other peripherals (if you're powering the board using the USB port).

## General Purpose Input Output Pins (GPIOs)

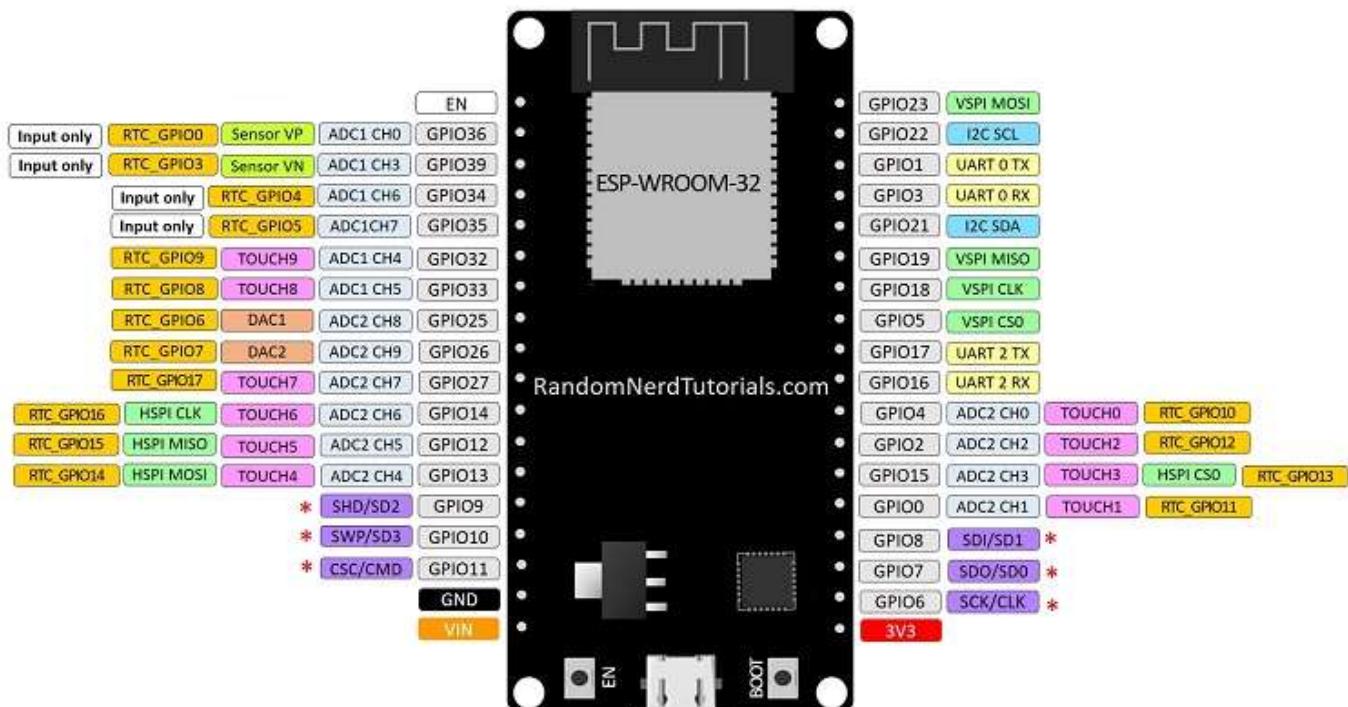
Almost all GPIOs have a number assigned and that's how you should refer to them—by their number.

With the ESP32 you can decide which pins are UART, I<sup>2</sup>C, or SPI – you just need to

If you don't set them on the code, the pins will be configured by default as shown in the figure below (the pin location can change depending on the manufacturer). Additionally, there are pins with specific features that make them suitable or not for a particular project.

## ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



We have a detailed guide dedicated to the ESP32 GPIOs that we recommend you read: [ESP32 Pinout Reference Guide](#). It shows how to use the ESP32 GPIOs and explains what are the best GPIOs to use depending on your project.

The placement of the GPIOs might be different depending on your board model. However, usually, each specific GPIO works in the same way regardless of the development board you're using (with some exceptions). For example, regardless of the board, usually GPIO5 is always the VSPI CS0 pin, GPIO 23 always corresponds to VSPI MOSI for SPI communication, etc.



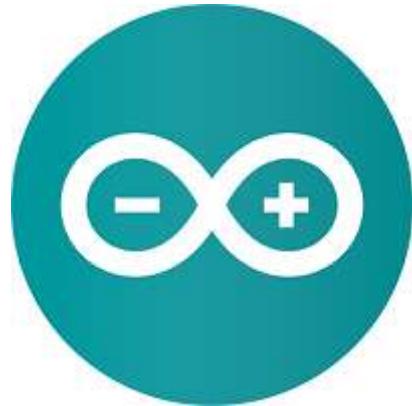
The ESP32 can be programmed using different firmware and programming languages. You can use:

- [Arduino C/C++ using the Arduino core for the ESP32](#)
- [Espressif IDF \(IoT Development Framework\)](#)
- [Micropython](#)
- [JavaScript](#)
- [LUA](#)
- ...

Our preferred method to program the ESP32 is with C/C++ “Arduino programming language”. We also have some guides and tutorials using [MicroPython firmware](#).

Throughout this guide, we’ll cover [programming the ESP32 using the Arduino core for the ESP32 board](#). If you prefer using MicroPython, please refer to this guide: [Getting Started with MicroPython on ESP32](#).

## Programming ESP32 with Arduino IDE



To program your boards, you need an IDE to write your code. For beginners, we recommend using Arduino IDE. While it’s not the best IDE, it works well and is simple and intuitive to use for beginners. After getting familiar with Arduino IDE and you start creating more complex projects, you may find it useful to use [VS Code with the Platformio extension](#) instead.

If you’re just getting started with the ESP32, start with [Arduino IDE](#). At the time of



ESP32. While version 2 works well with Arduino, there are still some bugs and some features that are not supported yet for the ESP32.

## Installing Arduino IDE

To run Arduino IDE, you need JAVA installed on your computer. If you don't, go to the following website to download and install the latest version: <http://java.com/download>.

## Downloading Arduino IDE

To download the Arduino IDE, visit the following URL:

- <https://www.arduino.cc/en/Main/Software>

**Don't install the 2.0 version.** At the time of writing this tutorial, **we recommend using the legacy version (1.8.19)** with the ESP32. While version 2 works well with Arduino, there are still some bugs and some features that are not supported yet for the ESP32.

Scroll down until you find the legacy version section.

### Legacy IDE (1.8.X)



## Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for installation instructions.

**SOURCE CODE**

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

**DOWNLOAD OPTIONS**

**Windows** Win 7 and newer  
**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#) 

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#)  
[Checksums \(sha512\)](#)

# Running Arduino IDE

Grab the folder you've just downloaded and unzip it. Run the executable file called *arduino.exe* (highlighted below).

|                       |                    |                       |           |
|-----------------------|--------------------|-----------------------|-----------|
| revisions.txt         | 10/28/2022 4:10 PM | Text Document         | 97 KB     |
| wrapper-manifest.xml  | 10/28/2022 4:10 PM | Microsoft Edge H...   | 1 KB      |
| arduino.exe           | 10/28/2022 4:10 PM | Application           | 72 KB     |
| arduino.l4j.ini       | 10/28/2022 4:10 PM | Configuration sett... | 1 KB      |
| arduino_debug.exe     | 10/28/2022 4:10 PM | Application           | 69 KB     |
| arduino_debug.l4j.ini | 10/28/2022 4:10 PM | Configuration sett... | 1 KB      |
| arduino-builder.exe   | 10/28/2022 4:10 PM | Application           | 23,156 KB |
| libusb0.dll           | 10/28/2022 4:10 PM | Application exten...  | 43 KB     |
| msvcp100.dll          | 10/28/2022 4:10 PM | Application exten...  | 412 KB    |
| msrvcr100.dll         | 10/28/2022 4:10 PM | Application exten...  | 753 KB    |
| tools-builder         | 10/28/2022 4:12 PM | File folder           |           |
| tools                 | 10/28/2022 4:12 PM | File folder           |           |
| libraries             | 10/28/2022 4:12 PM | File folder           |           |
| lib                   | 10/28/2022 4:11 PM | File folder           |           |
| java                  | 10/28/2022 4:11 PM | File folder           |           |
| hardware              | 10/28/2022 4:11 PM | File folder           |           |
| examples              | 10/28/2022 4:10 PM | File folder           |           |
| drivers               | 10/28/2022 4:10 PM | File folder           |           |

The Arduino IDE window should open.



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_oct28a | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and refresh. The main area displays the following code:

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
```

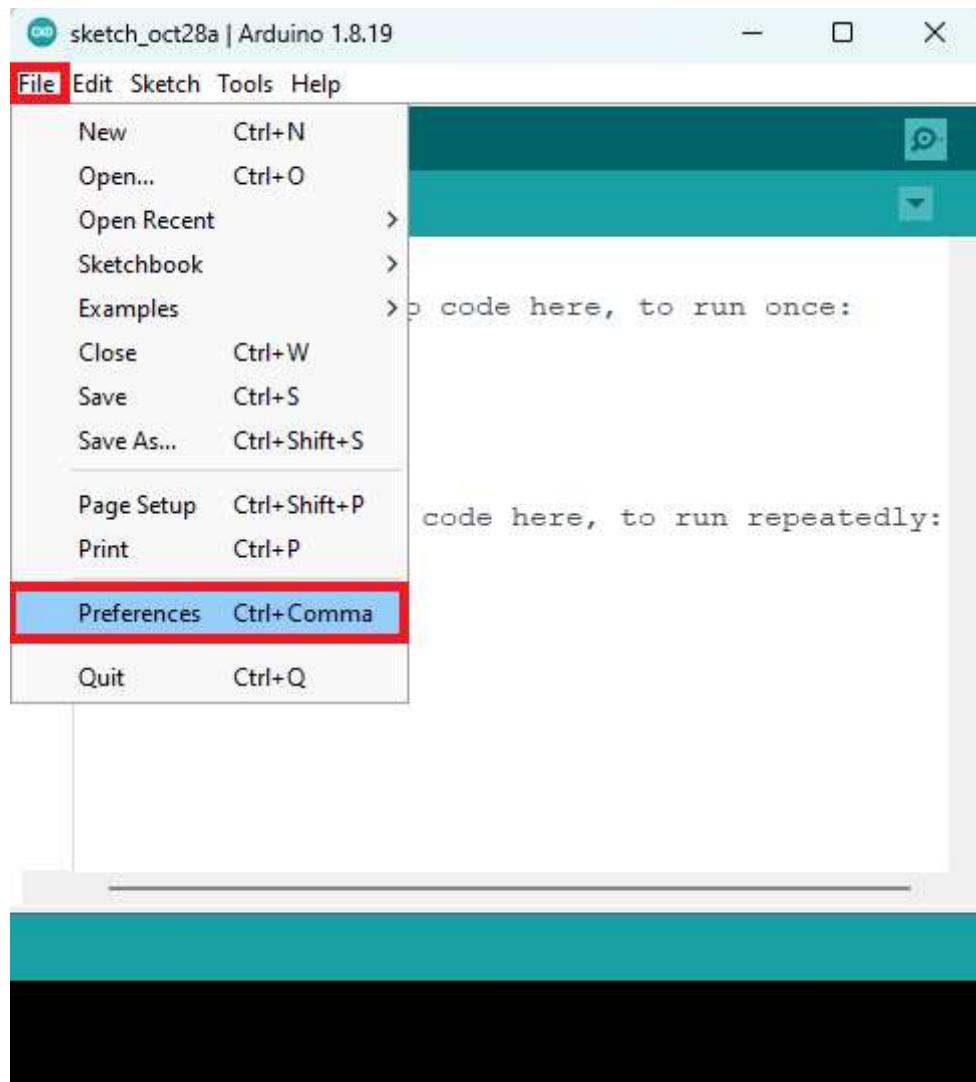
At the bottom of the IDE window, there is a status bar with the text: "MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM6".

## Installing the ESP32 in Arduino IDE

To be able to program the ESP32 using Arduino IDE, you need to add support for the ESP32 boards. Follow the next steps:

1. Go to **File > Preferences**.



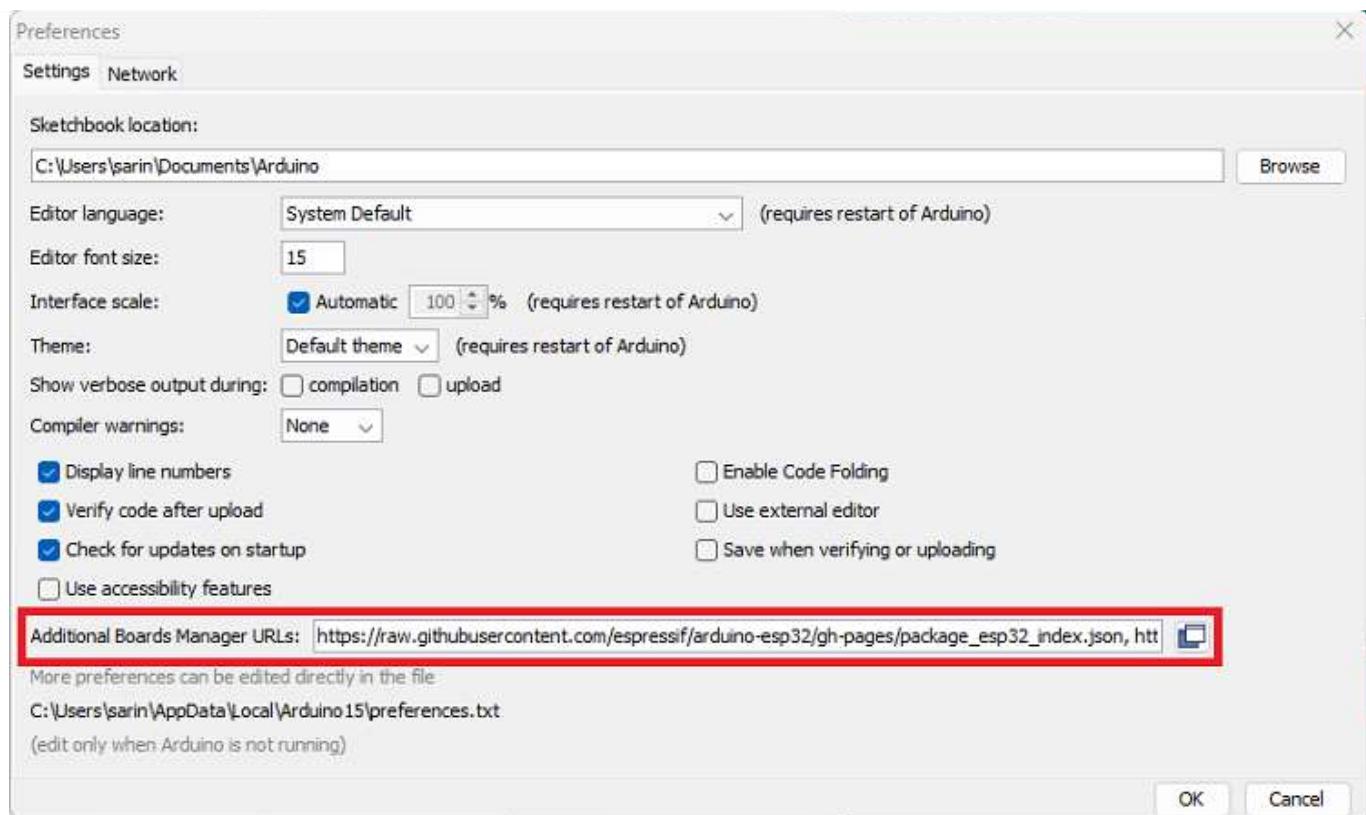


2. Enter the following into the “Additional Board Manager URLs” field. This will add support for ESP32 and ESP8266 boards as well.

[https://raw.githubusercontent.com/espressif/arduino-esp32/master/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/master/package_esp32_index.json),  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

See the figure below. Then, click the “OK” button.

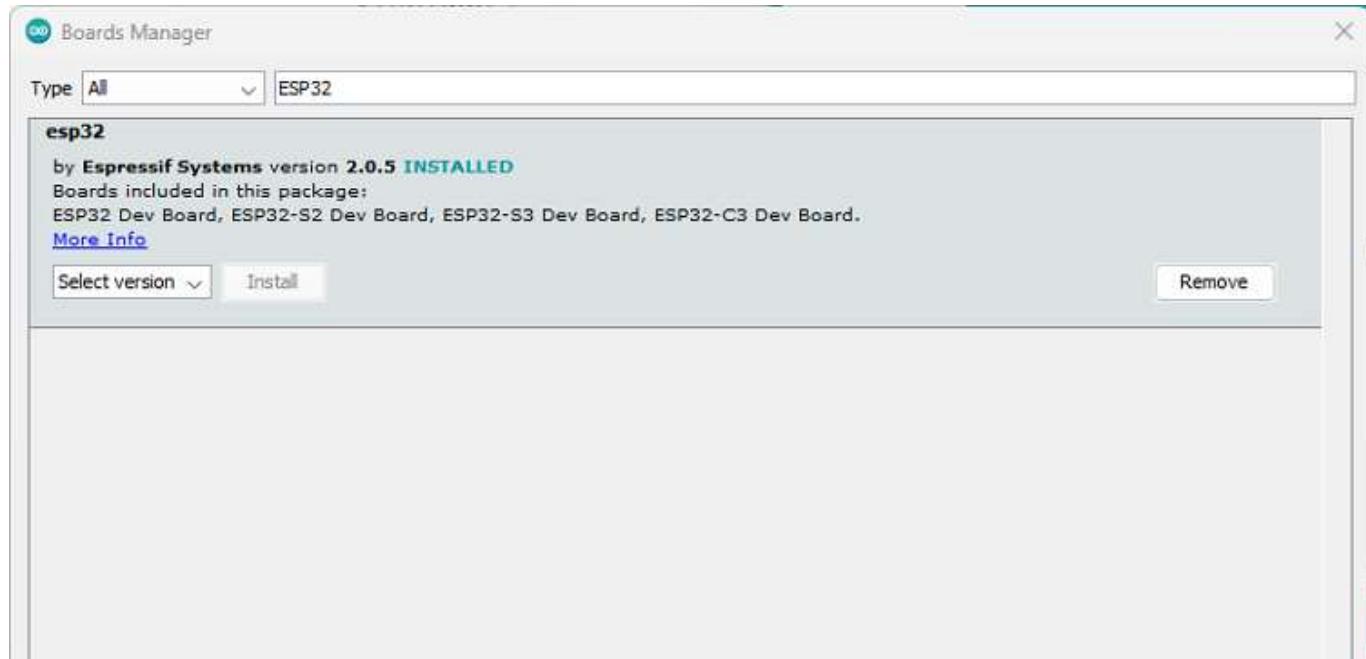




3. Open the **Boards Manager**. Go to **Tools > Board > Boards Manager...**

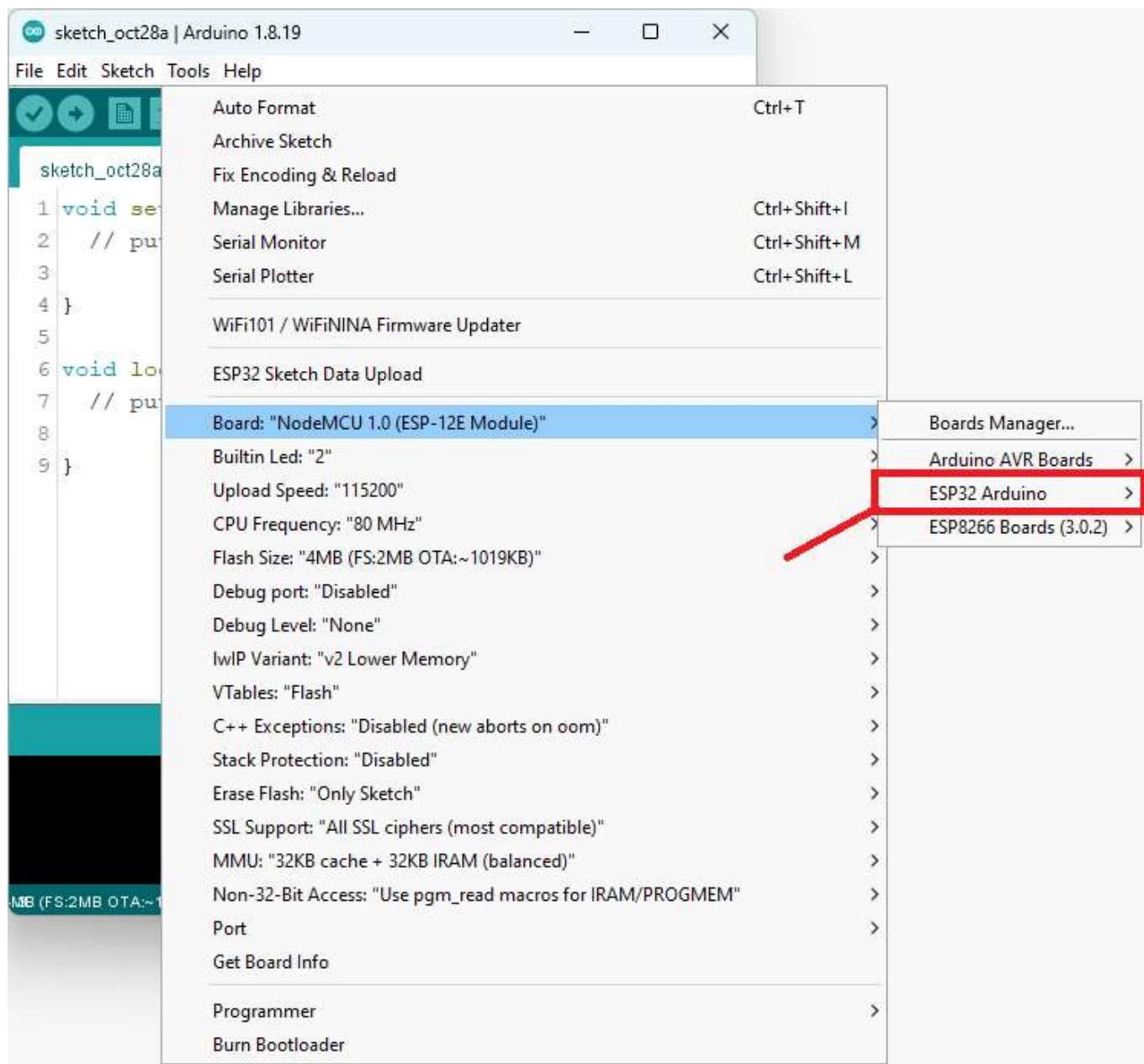
4. Search for **ESP32** and install the “**ESP32 by Espressif Systems**”:

That's it. It will be installed after a few seconds.



After this, restart your Arduino IDE.

Then, go to **Tools > Board** and check that you have ESP32 boards available.

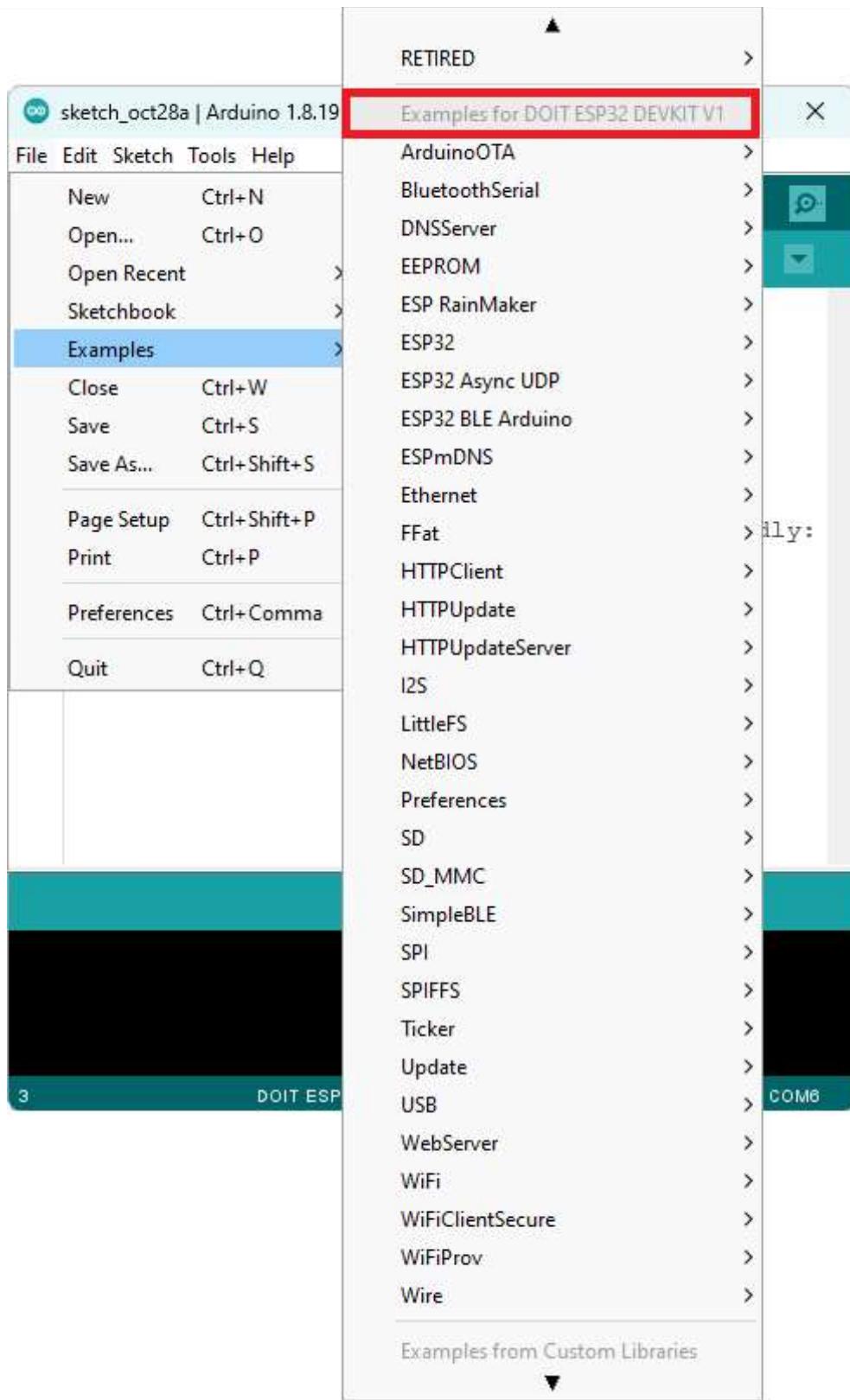


Now, you're ready to start programming your ESP32 using Arduino IDE.

## ESP32 Examples

In your Arduino IDE, you can find multiple examples for the ESP32. First, make sure you have an ESP32 board selected in **Tools > Board**. Then, simply go to **File >**





## Update the ESP32 Core in Arduino IDE

Once in a while, it's a good idea to check if you have the latest version of the ESP32 boards add-on installed.



You just need to go to **Tools > Board > Boards Manager**, search for **ESP32**, and check the version that you have installed. If there is a more recent version available, select that version to install it.

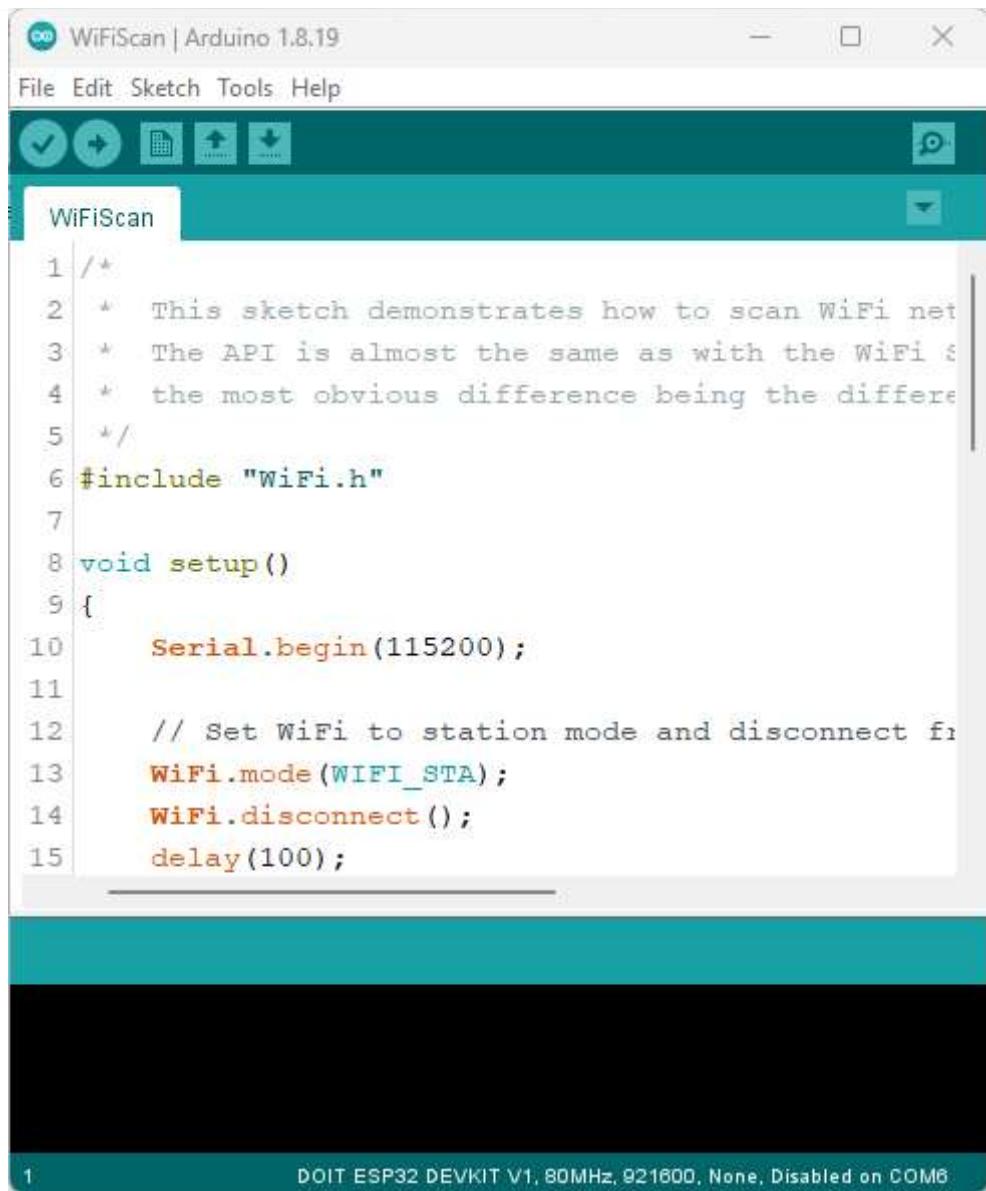
## Upload Code to the ESP32 using Arduino IDE

To show you how to upload code to your ESP32 board, we'll try a simple example available in the Arduino IDE examples for the ESP32.

First, make sure you have an ESP32 selected in **Tools > Board**. Then, go to **File > Examples > WiFi > WiFiScan**.

This will load a sketch that scans Wi-Fi networks within the range of your ESP32 board.





The screenshot shows the Arduino IDE interface with the title bar "WiFiScan | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Upload. The main window displays the "WiFiScan" sketch code. The code is as follows:

```
1 /*
2 * This sketch demonstrates how to scan WiFi net
3 * The API is almost the same as with the WiFi library
4 * the most obvious difference being the difference in
5 */
6 #include "WiFi.h"
7
8 void setup()
9 {
10     Serial.begin(115200);
11
12     // Set WiFi to station mode and disconnect from
13     // any access point
14     WiFi.mode(WIFI_STA);
15     WiFi.disconnect();
16     delay(100);
```

The status bar at the bottom indicates "DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on COM8".

Connect your ESP32 development board to your computer using a USB cable. If you have an ESP32 DEVKIT DOIT board, the built-in red LED will turn on. This indicates the board is receiving power.

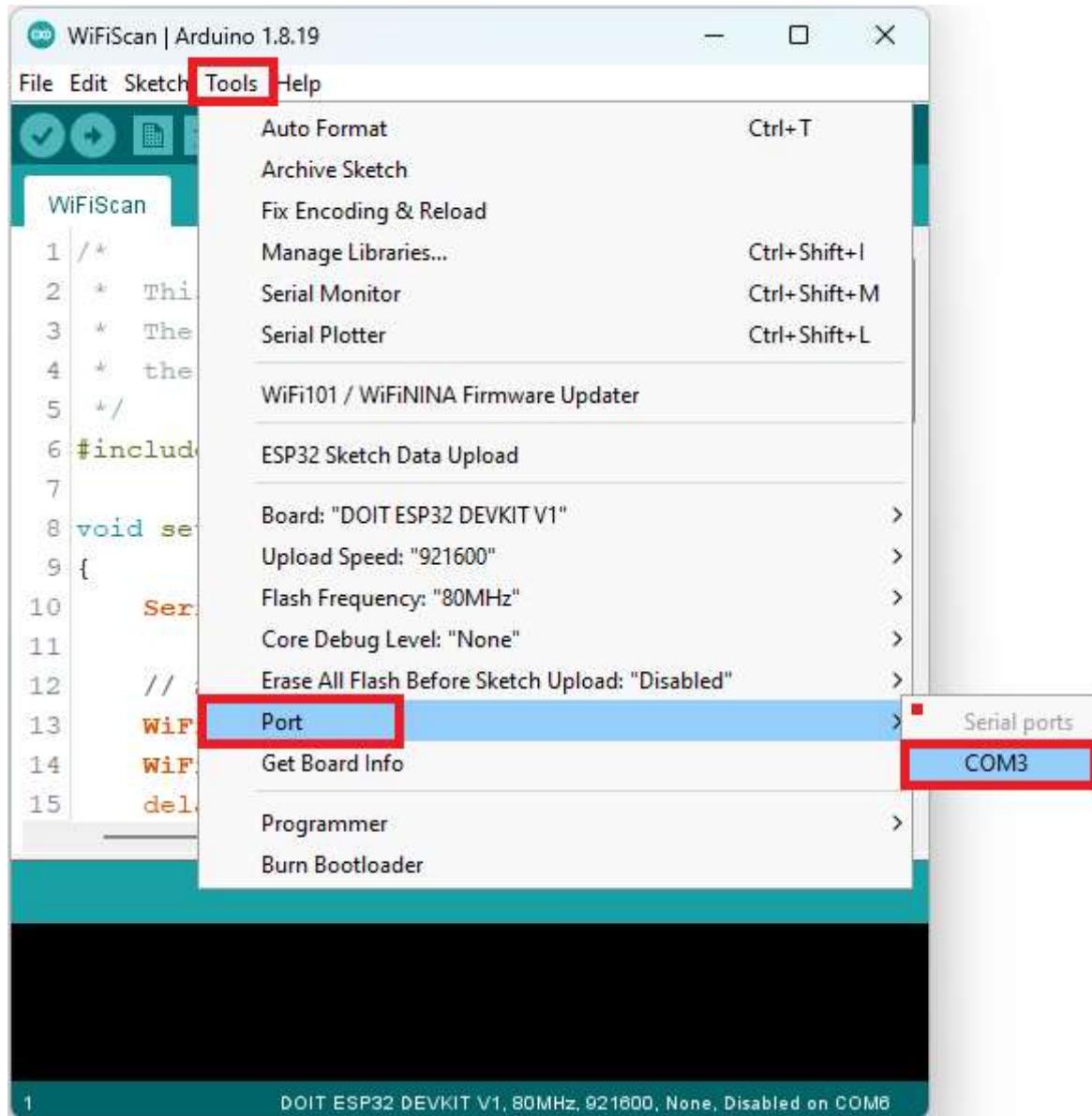
**Important:** you must use a USB cable with data wires. Some USB cables from chargers or power banks are power only and they don't transfer data—these won't work.

Now, follow the next steps to upload the code.

- 1) Go to **Tools > Board**, scroll down to the ESP32 section and select the name of



2) Go to **Tools** > **Port** and select a COM port available. If the COM port is grayed out, this means you don't have the required USB drivers. Check the section [Installing USB Drivers](#) before proceeding.



3) Press the upload button.



Some boards will automatically go into flashing mode and the code will be successfully uploaded straight away.

Failed to connect to ESP32: Timed out... Connecting...

Or something like:

A fatal error occurred: Failed to connect to ESP32: Wrong boot mode detected (0x13)! The chip needs to be in download mode.

This means the ESP32 was not in flashing mode when you tried to upload the code. In this situation, you should long press the board **BOOT** button, when you start seeing the “**Connecting....**” message on the debugging window.

**Note:** in some boards, a simple trick can make the ESP32 go into flashing mode automatically. Check it out on the following tutorial: [\[SOLVED\] Failed to connect to ESP32: Timed out waiting for packet header](#).

Now, the code should be successfully uploaded to the board. You should get a “**Done uploading**” message”.

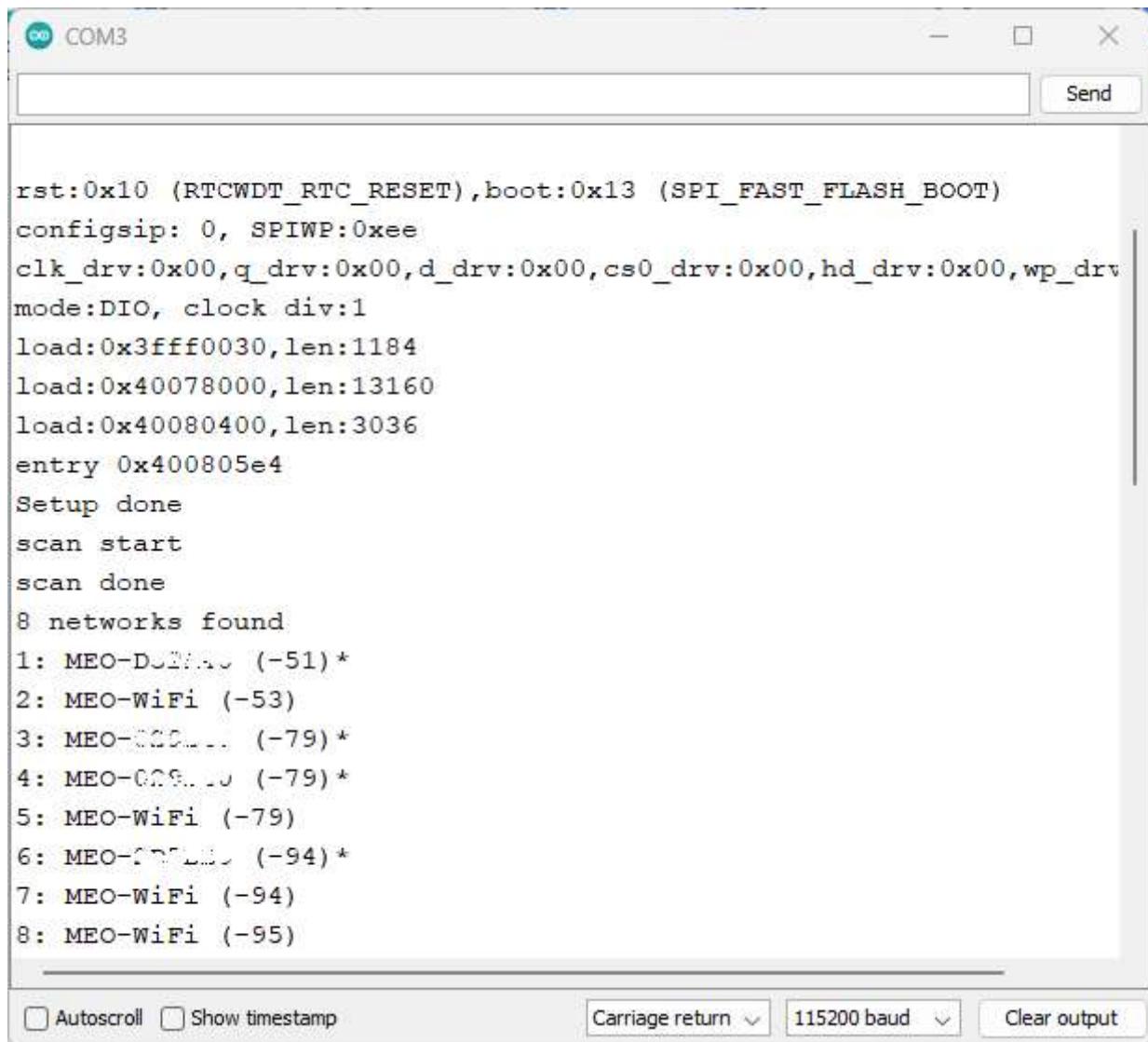
```
Done uploading.  
Wrote 681936 bytes (444520 compressed) at 0x00010000 in 7.0 s  
Hash of data verified.  
  
Leaving...  
Hard resetting via RTS pin...
```

# Demonstration

To see if the code is working as expected, open the Serial Monitor at a baud rate of 115200.



You should get a list of nearby wi-fi networks.



```
rst:0x10 (RTCWDT_RTC_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
mode:DIO, clock div:1
load:0x3fff0030,len:1184
load:0x40078000,len:13160
load:0x40080400,len:3036
entry 0x400805e4
Setup done
scan start
scan done
8 networks found
1: MEO-DU... (-51) *
2: MEO-WiFi (-53)
3: MEO-CC... (-79) *
4: MEO-C... (-79) *
5: MEO-WiFi (-79)
6: MEO-... (-94) *
7: MEO-WiFi (-94)
8: MEO-WiFi (-95)
```

Autoscroll  Show timestamp Carriage return 115200 baud Clear output

This means everything went as expected.

## Installing ESP32 USB Drivers

After connecting the ESP32 board to your computer, if the COM port in Arduino IDE is grayed out, it means you don't have the necessary USB drivers installed on your computer.

Most ESP32 boards either use the CP2101 or CH340 drivers. Check the USB to UART converter on your board, and install the corresponding drivers.



You'll easily find instructions with a quick google search. For example "install CP2101 drivers Windows".

## Wrapping Up

We hope you've found this getting started guide useful. I think we've included all the required information for you to get started. You learned what is an ESP32, how to choose an ESP32 development board, and how to upload new code to the ESP32 using Arduino IDE.

Want to learn more? We recommend the following tutorials to get started:

- [ESP32 Digital Inputs and Digital Outputs \(Arduino IDE\)](#)
- [ESP32 Web Server Tutorial](#)

Also, don't forget to take a look at the ESP32 pinout to learn how to use its GPIOs:

- [ESP32 Pinout Reference: Which GPIO pins should you use?](#)

If you're serious about learning about the ESP32, we recommend taking a look at our best-selling eBook:

- [Learn ESP32 with Arduino IDE eBook](#)

You can also check all our free ESP32 tutorials and guides on the following link:

- [More ESP32 Projects](#)

If you like ESP32 make sure you [subscribe to our blog](#), so you don't miss upcoming projects.

**Do you have any questions? Leave a comment down below!**

Thanks for reading.



# ONLY \$5 for 10 PCBs

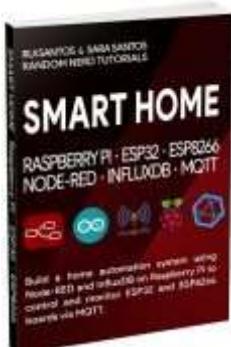
✓ 24-hour Build Time ✓ Quality Guaranteed

✓ Most Soldermask Colors:



[Order now](#)

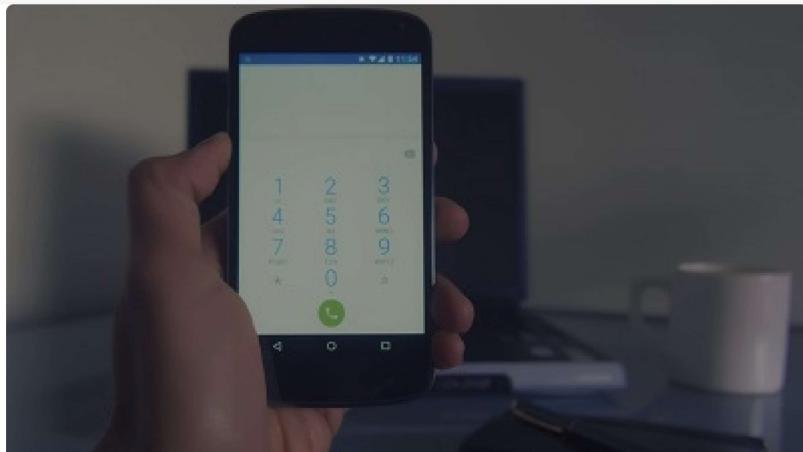
[www.pcbway.com](http://www.pcbway.com)



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

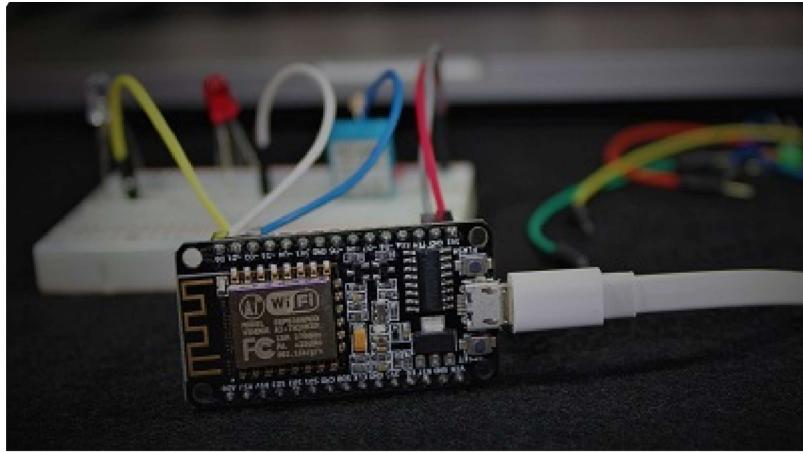
Learn how to build a home automation system and we'll cover the following main subjects: Node-RED, Node-RED Dashboard, Raspberry Pi, ESP32, ESP8266, MQTT, and InfluxDB database [DOWNLOAD »](#)

## Recommended Resources

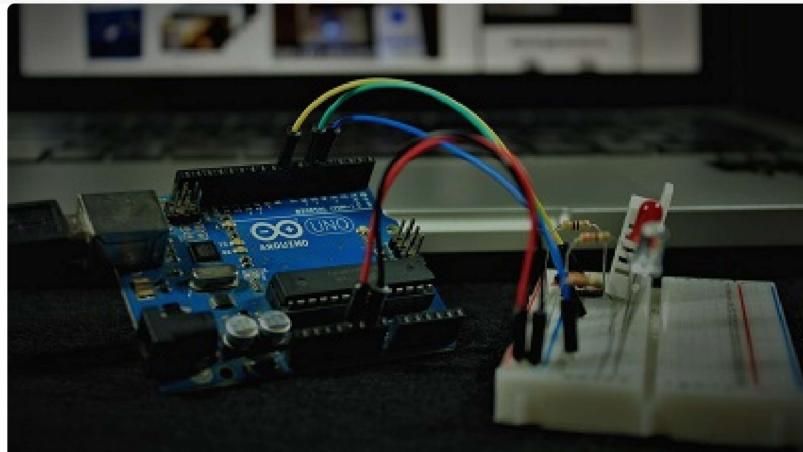


[Build a Home Automation System from Scratch »](#) With Raspberry Pi, ESP8266, Arduino and Node-RED





[\*\*Home Automation using ESP8266 eBook and video course »\*\*](#) Build IoT and home automation projects.



[\*\*Arduino Step-by-Step Projects »\*\*](#) Build 25 Arduino projects with our course, even with no prior experience!

## What to Read Next...



[ESP8266 NodeMCU Web Server with Slider: Control LED Brightness \(PWM\)](#)

---

[ESP32 Door Status Monitor with Email Notifications \(IFTTT\)](#)



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

Your Email Address

 SUBSCRIBE

## 92 thoughts on “Getting Started with the ESP32 Development Board”



Jithendra

December 29, 2016 at 2:02 pm

Esp8266/esp32+usb camera stream to webpage is possible?

[Reply](#)



Rui Santos

December 29, 2016 at 7:06 pm

It won't be able to stream, but with the ESP you can create a web server that has a camera stream embedded (for example, as you would embed a youtube video in an HTML page).

[Reply](#)





**Don French**

December 29, 2016 at 6:34 pm

How about a little more about the dev board? What are the user-accessible components – buttons, ports, pins,. A look at the back of the board? What are the dimensions? When you said press the Upload button it sounds like there is a button on the board called the Upload button but I think you meant select Upload from the Sketch menu in the Arduino IDE. Is that correct? You might want to make that clearer.

[Reply](#)



**Rui Santos**

December 29, 2016 at 7:01 pm

Hi Don,

Thanks for the comment! The back of the board has no components, it looks like the ESP-12E NodeMCU Kit module.

There's only two buttons in the ESP32 Dev Module: EN and BOOT. I've updated the image in the blog post to highlight those two buttons and make it easy to see them.

The Upload button I've mentioned in the post was for the Arduino IDE – I've also fixed that sentence.

The image in the “Pin assignment” section describes exactly the function of each pin.

I hope this helps,

Thank you!

Rui





**Jeff Young**

December 30, 2016 at 5:44 pm

Hi RUI I wanted to know about the esp32 so now I can start playing thsnks so much love all your getting started projects jeff

[Reply](#)



**idan Ben-Moshe**

January 17, 2017 at 8:56 pm

Does esp32 already has BLE functionalities on arduinoIDE today?

[Reply](#)



**Rui Santos**

January 24, 2017 at 7:13 pm

I don't think so, but it should have in the future. I'll write more on that subject when it's available

[Reply](#)





**Jeff Young**

January 26, 2017 at 7:46 pm

Hi Rui very good will get a esp32 soon

[Reply](#)



**Enrique Gongora**

January 27, 2017 at 2:59 pm

Rui I am a bit confused about your course Build a Home Automation System for \$ 100 because I see the need to make a wiring for each element that wants to automate starting from a single ESP8266 this prevents me from using it in an existing construction wasting the wireless qualities of the ESP8266 module by Please tell me what I understood? Thank you

[Reply](#)



**Michael**

January 16, 2018 at 5:30 pm

Thanks for posting this. So many people think that going with ESP8266 and ESP32 you have to learn a whole new system and can't run under the Arduino ide. So many are stuck with limited I/O, processing power, etc. So



[Reply](#)



**Ken S**

September 8, 2018 at 9:48 am

Great tutorials, you have got me into both the ESP 32 and ESP8266.

Worth pointing out that the Arduino IDE now has lots of examples of using these devices. I used these with your tutorial and got going by just plugging the device into the PC USB port – no extra circuitry necessary.

[Reply](#)



**Sara Santos**

September 12, 2018 at 9:27 am

Hi Ken!

That's true there are lots of examples on the Arduino IDE that help you get started! That's great for beginners!

Thank you for sharing!

Regards,

Sara 😊

[Reply](#)



**Ken S**

September 12 2018 at 3:35 pm



I also use Visual Micros IDE for Arduino which uses the Arduino IDE but is an add on for Microsoft Visual Studio. That is really nice because it makes available the power of Visual Studio with for example Intellisense, online debugging and GIT software backup and revision control. The community edition of VS is free for small scale users, as is Visual Studio Team Services – their GIT Server. Your tutorials for the ESP32 and ESP8266 work in this environment too.

Best Regards

Ken

[Reply](#)



**Franklin**

September 13, 2018 at 12:20 pm

So you have to press the button on the esp32 to upload. Won't that little button wear out soon . for me because I make too many mistakes and have to reload. Is there a work around?

[Reply](#)



**Sara Santos**

September 13, 2018 at 1:56 pm

Hi Franklin.



I know it is a bit frustrating to press the boot button every time you need to upload new code, specially when there are a lot of mistakes and we want to test things right away.

Unfortunately I don't have any work around to share. You really need to press that button.

However, if anyone knows any work around, please share.

Regards,

Sara 😊

[Reply](#)



**Franklin**

September 13, 2018 at 12:27 pm

Thank you Rui and Sara for sharing. It has helped me tremendously as I am sure many others. The projects are well put together and very helpful and easy to understand. You are kind in your answers.

[Reply](#)



**Sara Santos**

September 13, 2018 at 1:57 pm

Thank you so much for your kind words!

We're really happy to hear that.

Regards,

Sara 😊





**Kevin**

October 18, 2018 at 6:34 pm

Thanks for the article. Does the ESP32 have analog pins? If so, which ones are they?

Kevin

[Reply](#)



**Sara Santos**

October 19, 2018 at 1:39 pm

Hi Kevin.

Yes, the ESP32 has analog pins: digital to analog converter pins(DAC) and analog to digital converter (ADC) pins.

If you want to read analog sensors, just use the ADC pins. These are the ADC pins: GPIOs: 0, 2, 4, 12, 13, 14, 15, 25, 26, 27 32, 33, 34, 35, 36, and 39.

If you want to produce analog signals, use the DAC pins: GPIO25 and 26.

Regards,

Sara 😊

[Reply](#)



**RobinBlood**



But be careful, the ADCs are separated into two banks.  
The 2nd bank is deactivated when you enable WiFi.

[Reply](#)



**Ion Gheorghe**

May 12, 2020 at 3:52 pm

Thank you Robin for this great info.  
I usually does not read manuals, so you save me a lot of troubles.  
Have a great day.

[Reply](#)



**Mariano Hortal**

October 22, 2018 at 2:50 pm

Trying to compile the blinking LED project I get a compilation error in  
esptool.py line 34:  
Importers of: No module named serial  
Can you help, please?

[Reply](#)



**Sara Santos**

October 22, 2018 at 2:54 pm



Hi Mariano.

It seems you're having the same problem as described here:

<https://github.com/espressif/arduino-esp32/issues/13>

Take a look at that article and see if you can solve it.

Other tutorials that may be helpful:

Intallation guide: <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

Troubleshooting guide: <https://randomnerdtutorials.com/esp32-troubleshooting-guide/>

I hope this helps.

Tell me how it went.

Regards,

Sara 😊

[Reply](#)



**Sara Santos**

October 23, 2018 at 8:55 am

Please read Alexandre Viriato answer.

It should help. 😊

[Reply](#)



**Mariano Hortal**

October 23, 2018 at 4:43 pm



Thanks Sara.

[Reply](#)



**Mariano Hortal**

October 23, 2018 at 4:45 pm

Although I am not running Windows but Ubuntu 18, the problem was the same: missing module “serial” in python. In my case the solution was to run sudo apt-get install python-serial

Now it works OK!!!

[Reply](#)



**Sara Santos**

October 24, 2018 at 9:21 am

Great! 😊

[Reply](#)



**Saber**

November 6, 2018 at 8:44 am



Are you planing to create similar ESP32 course using esp-idf instead of arduino?  
or can you please refer me to a website that teaches how to use esp-idf?  
Thank you very much

[Reply](#)



**Sara Santos**

November 8, 2018 at 10:07 am

Hi Saber.  
At the moment, we don't have anything planned about esp-idf.  
We're working on a new course using MicroPython that will be available soon.  
I don't know any good website that teaches how to use esp-idf.  
A good starting point is the official documentation:  
[docs.espressif.com/projects/esp-idf/en/latest/index.html](https://docs.espressif.com/projects/esp-idf/en/latest/index.html)  
I hope this helps.  
Regards,  
Sara 😊

[Reply](#)



**Omemanti**

November 22, 2018 at 10:53 am

So far great work :)..



question, I used this to upload a sketch, it worked the first time. Now I want to do it again but it fails..

any suggestions?

the error i get from the arduino IDE;

//////////

esptool.py v2.3.1

Connecting.....

Chip is ESP32D0WDQ6 (revision 1)

A fatal error occurred: Timed out waiting for packet header

A fatal error occurred: Timed out waiting for packet header

[Reply](#)



Sara Santos

November 22, 2018 at 11:38 am

Hi.

I think that error means your ESP32 is not in flashing mode when uploading the code.

You need to press the BOOT button while uploading code until you see the “Connecting...” message in the Arduino IDE serial monitor.

Please take a look at bullet 4 in our ESP32 troubleshooting guide:

<https://randomnerdtutorials.com/esp32-troubleshooting-guide/>

Let me know if that solved your problem.

Regards,

Sara 😊

[Reply](#)





**Omemanti**

November 22, 2018 at 11:46 am

Thnx,

well, I guess I did it wrong a couple of times, again pressing the boot button until connecting.... and now it worked again 😊

thnx for the quick response.

[Reply](#)



**Safalya**

February 26, 2019 at 2:46 pm

The above didn't work for me. LED's weren't blinking.

To resolve this, I created a /hardware folder and pasted

<https://github.com/espressif/arduino-esp32>.

Used esp32 Dev module board. Then it worked.

[Reply](#)



**Tejash kumar**

March 25, 2019 at 1:14 pm



Hi

i am working on esp32 dev kit,for developing Home automation,  
suddenly i got struck with spi flash boot error, i am trying to resolve it, but we  
cannot, please can any one help us solve the problem.

Thanks,

[Reply](#)



**Sara Santos**

March 26, 2019 at 10:19 am

Hi.

What is the error that you are getting?

You can check our troubleshooting guide and see if you can resolve your issue: <https://randomnerdtutorials.com/esp32-troubleshooting-guide/>

Your error is probably this one: <https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/>

Regards,

Sara

[Reply](#)



**Dave Tuttle**

April 29, 2019 at 3:40 pm

Nice example but you don't mention that if you use GPIO 2 instead of GPIO23. there is already a blue LED connected to the dev board.



[Reply](#)



**Rui Santos**

May 1, 2019 at 10:56 am

Hello Dave, in my example I'm controlling GPIO 23, but you are right there's an on-board built-in LED on GPIO 2.

[Reply](#)



**Mike McKinney**

October 24, 2019 at 6:38 am

Ok I am completely a nooby to this but can you tell me what kind of computer you are hooking the ESP32 up to in order to write the code?

[Reply](#)



**Sara Santos**

October 24, 2019 at 2:01 pm

Hi Mike.

It's a regular laptop with windows.

Regards,

Sara





**Tommy Hollifield**

November 23, 2019 at 2:25 am

I'm running a Wemos ESP32 OLED board. GREAT tutorial! I would never have figured out to add the "Wire.begin(5,4);," line without you. I'm running the Adafruit SSD1306\_128x64\_i2c sample and don't understand the display.startscrolldiagright(start, end) function. It doesn't seem to do anything. Seems 4 parameters would be needed for a diagonal scroll? I've looked at the Adafruit\_SSD1306.cpp library code without understanding what it does. Is more info on the library available?

[Reply](#)



**Sara Santos**

November 24, 2019 at 3:22 pm

Hi Tommy.

You can read this tutorial for more details:

<https://randomnerdtutorials.com/esp8266-0-96-inch-oled-display-with-arduino-ide/>

Regards,

Sara

[Reply](#)





**kumar jha**

February 13, 2020 at 4:29 pm

Hi can any one tell me how to measure elbow angle using MCP6050 GY521 and ESP32?

[Reply](#)



**Sara Santos**

February 14, 2020 at 11:32 am

Hi.

At the moment, we don't have any tutorial about that subject.

Regards,

Sara

[Reply](#)



**SHIMUL**

March 7, 2020 at 7:16 am

THANKS

[Reply](#)





**Ion Gheorghe**

April 17, 2020 at 11:39 am

Hello Sara

I am using ESP32 Doit devkit v1 as you suggested, from Banggood, but after compiling, it cannot connect to upload the code.

The port is correct because when I plug it first time it did show me the port installed successfully. Here is the error message. What I can do ?

Thank you

Arduino: 1.8.12 (Windows 7), Board: "DOIT ESP32 DEVKIT V1, 80MHz, 115200, None"

Sketch uses 622174 bytes (47%) of program storage space. Maximum is 1310720 bytes.

Global variables use 38736 bytes (11%) of dynamic memory, leaving 288944 bytes for local variables. Maximum is 327680 bytes.

```
H:\Users\Nelu\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\2.6.1/esptool.exe --chip esp32 --port COM13 --baud 115200 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect 0xe000
```

```
H:\Users\Nelu\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4/tools/partitions/boot_app0.bin 0x1000
```

```
H:\Users\Nelu\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4/tools/sdk/bin/bootloader_dio_80m.bin 0x10000
```

```
H:\Users\Nelu\AppData\Local\Temp\arduino_build_604947\Get_Mac.ino.bin 0x8000
```

```
H:\Users\Nelu\AppData\Local\Temp\arduino_build_604947\Get_Mac.ino.partitions.bin
```

esptool.py v2.6

Serial port COM13

Connecting.....\_\_\_\_\_

An error occurred while uploading the sketch



A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

This report would have more information with “Show verbose output during compilation” option enabled in File -> Preferences.

[Reply](#)



**Ion Gheorghe**

April 17, 2020 at 11:41 am

I forgot to mention that port 13 works perfectly for esp8266.  
I wanted to get the MAC from some esp8266, and i did not have any problem with the port.

[Reply](#)



**Ion Gheorghe**

April 17, 2020 at 11:56 am

What i can read when i press TOOLS  
Board-Doit ESP32 Devkit V1  
Upload speed 115200  
Flash frequency 80 MHz  
Core debug level none  
When i press GET BOARD INFO I GET THIS:



PID EA60

SN none

Programer AVRISP mkII

I bought a batch of 10 pieces.

I tried 3 and all they do the same.

Please help.

Thank you

[Reply](#)



**Ion Gheorghe**

April 17, 2020 at 12:02 pm

Sorry,

It is solved by pressing the BOOT button.

I works now

[Reply](#)



**Teddy Chen**

April 17, 2020 at 9:05 pm

As Sara mentioned it in previously, putting a 10 uF capacitor between the EN and GND will let the IDE upload the code automatically without the need to press the EN button.

[Reply](#)





**Ion Gheorghe**

April 17, 2020 at 9:12 pm

Thank you for suggestion Teddy,  
But i had a batch of 10 ESP received from Chine and they were not  
mounted anywhere.  
I just had the USB cable attached.  
I have the habit to get the MAC for all, when i get them.  
Next time i will follow your advice, put it in a bread board and use the  
capacitor.

[Reply](#)



**Bob**

October 30, 2020 at 2:33 pm

I just got an esp32 that is different looking than those above. ESP-32 Dev kit C V4.

I didn't notice when I bought it but it doesn't seem to have an internal antenna but does the the jack for an external one. Very happy about that!

Got it here so you can have a look:

[amazon.com/gp/product/B081BCDFK3/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o01\\_s01?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B081BCDFK3/ref=ppx_yo_dt_b_asin_title_o01_s01?ie=UTF8&psc=1)

Might want to add that one to the pics above. 😊



[Reply](#)



**Ion Gheorghe**

December 30, 2020 at 12:51 pm

I developed a device with ESP32 DOIT.

I want to change and move it to ESP32 HUZZAH32.

The reason is the battery charger on board for Adafruit module

I read the whole data sheet but i am confused about the pins i can use without any conflict.

On the DOIT i had, first a membrane keypad.

4 Outputs for horizontal lines with the following GPIO:

KH1-26

KH2-19

KH3-18

KH4-23

3 Inputs for vertical line. Each line is pulled up by 1K resistor:

KV1-35

KV2-33

KV3-34

2 Leds Outputs:

Red-21

Green-22

1 Output = Enable control line for your Adafruit MiniBoost 5V @ 1A –

TPS61023:

Pwr\_On-5

1 Input from and RFID Reader (TX-line of the reader)

RFID\_In\_25



I need help please. Just to select some pins corresponding to the one on DOIT and without any conflict.

Adafruit support did not help. They said to post in forum, there nobody read it or answer, so i am stuck.

Please help.

Thank you very much.

[Reply](#)



Sara Santos

December 30, 2020 at 7:22 pm

Hi Ion.

I have not experience with that board.

But, taking a look at the pinout, I think you can use these pins:

4 Outputs for horizontal lines with the following GPIO:

KH1-26 → **26 (Huzzah)** You can use the same

KH2-19 → **19 (Huzzah)** You can use the same

KH3-18 → **18 (Huzzah)** You can use the same

KH4-23 → **23 (Huzzah)** You can use the same

3 Inputs for vertical line. Each line is pulled up by 1K resistor:

KV1-35 → **32 (Huzzah)** There is no GPIO 35 on the Huzzah, so you can use any other pin that can act as an input

KV2-33 → **33 (Huzzah)** You can use the same

KV3-34 → **34 (Huzzah)** You can use the same

2 Leds Outputs:

Red-21 → **21 (Huzzah)** You can use the same

Green-22 → **22 (Huzzah)** You can use the same

1 Output = Enable control line for your Adafruit MiniBoost 5V @ 1A –



Pwr\_On-5 → **5 (Huzzah)** You can use the same

1 Input from and RFID Reader (TX-line of the reader)

RFID\_In-25 → **25 (Huzzah)** You can use the same

I think these pins will work taking a look at the pinout, but I haven't experimented it.

Regards,

Sara

[Reply](#)



**Ion Gheorghe**

December 30, 2020 at 7:27 pm

Thank you very much Sara.

You know how to read and interpret better a pin and its functions.

I will get today the Adafruit modules , and test them in a breadboard.

I will send you the result when i am done.

Once again, Thank you very much.

[Reply](#)



**Sara Santos**

December 30, 2020 at 7:32 pm

Great!

Then, let me know.

Regards,



[Reply](#)



**Terry Harrison**

January 5, 2021 at 3:50 pm

Can you tell me if you have any articles using Platform IO for developing with the ESP 32?

[Reply](#)



**Sara Santos**

January 6, 2021 at 2:09 pm

Hi.

We have this article: <https://randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino/>

Regards,

Sara

[Reply](#)



**Mitchell Anderson**

January 15, 2021 at 4:44 pm



getting the following.

Arduino: 1.8.13 (Windows 10), Board: "DOIT ESP32 DEVKIT V1, 80MHz, 921600, None"

Sketch uses 856262 bytes (65%) of program storage space. Maximum is 1310720 bytes.

Global variables use 39696 bytes (12%) of dynamic memory, leaving 287984 bytes for local variables. Maximum is 327680 bytes.

esptool.py v2.6

Serial port COM1

Connecting.....

---

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

This report would have more information with  
“Show verbose output during compilation”  
option enabled in File -> Preferences.

[Reply](#)



**Sara Santos**

January 16, 2021 at 11:23 am



Hi.

Read this: <https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/>

I hope this helps.

Regards,

Sara

[Reply](#)



**Ion**

September 6, 2021 at 3:49 pm

Hello,

I have a question at which i cannot find the answer, and Google gave up.

An ESP32 development kit has an 5V power pin, an internal regulator down to 3.3 volts and the processor.

At which voltage works the internal processor ?

I mean, if i supply a 3.3V or 3.0V , and not 5V at power pin, meaning the input of the regulator, what will be the output at processor level, and if the whole development kit can function ?

What is the lowest voltage i can supply to the 5V pin , and have the system working ?

[Reply](#)



**yodrack**

June 30, 2022 at 7:32 am



Hi,  
I just bought two ESP32 wroom  
when I connect the USB to my computer (Win10), the board is not recognized  
by the device manager. I changed the cable and I tried with the second ESP.  
The problem is always the same.  
I think there is some config problem with my PC, can you advise me ?  
thanks

[Reply](#)



yodrack

June 30, 2022 at 9:02 am

in addition : I tried to connect the board to another PC (Win10) and it works !

the driver is the same, I copy the driver from the Win10PC to the Win11PC  
but no result

[Reply](#)



Sara Santos

June 30, 2022 at 2:29 pm

Hi.

Check the ESP32 troubleshooting guide point number 6:

<https://randomnerdtutorials.com/esp32-troubleshooting-guide/>

Regards,



[Reply](#)



**yodrack**

June 30, 2022 at 3:45 pm

Hi Sara,

thank you for your help.

I checked the informations :

- the cable is an USB cable including data wires. (it works with Arduino and other boards)
- the USB drivers is not missing. For being sure I perform a new downloaded and install

the cable and the board are OK and work when plugged to another computer running Win10, I compared the 2 computers (the Win10 and the Win11) the drivers are the same. It's seems the problem come from Win11.

what is your advise

regards

[Reply](#)



**yodrack**

July 6, 2022 at 9:03 am

Hi Sara

I tried to manually install the driver with :



RUNDLL32.EXE SETUPAPI.DLL,InstallHinfSection DefaultInstall 132  
C:\Users\yodra\Downloads\CP210x\_Universal\_Windows\_Driver\silabser.inf

nothing happens and the file silabser.sys is not copied to the driver directory

it seems something block or prevent the driver installation

please advise , I am lost ....  
regards

[Reply](#)



**Sara Santos**

July 6, 2022 at 4:07 pm

Hi.

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

Download the driver from here: <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

Then, unzip the file and check the installation instructions on the txt file.

Regards,

Sara

[Reply](#)



**yodrack**

July 7, 2022 at 8:47 am

Hi,



I downloaded the driver from silabs.com then I right click on the inf file then Install as said in the instruction text.

and nothing, no instruction, nothing happen

that's why I tried to install manually with the command RUNDLL32.EXE  
but no result

[Reply](#)



yodrack

July 9, 2022 at 4:50 pm

Hi Sara,

Thank for your answer.

there is no popup when I right click on the inf file. Nothing happen.

I tried all ways to install manually, nothing works

I received another ESP32 bought at another brand (at Bangood store),  
this board is equipped with an cp2102 and ..... it works on Win11. I have  
not been obliged to install the driver, it uses the standard Microsoft driver  
(I checked with device manager)

So maybe the first ESP has a hardware problem, or it uses a fake  
Cp2102,

anyway, I can use it, I will desolder the cp2102 chips and replace with a  
CH340



regards

[Reply](#)



**Sara Santos**

July 11, 2022 at 8:51 am

Hi.

Are you sure the older board was using the CP2102?

Maybe it had a CH340?

Also, did you check your USB cable? Some USB cables are power only and don't transfer data.

Regards,

Sara



**yodrack**

July 11, 2022 at 4:48 pm

HI

Yes the older board use CP2102

The USB cable in a data xfer cable

[Reply](#)



**NIKHIL V P**

July 11, 2022 at 4:48 pm



This board is it have Hall effect and temparature sensor

[Reply](#)



Ivo

November 3, 2022 at 3:19 pm

Typo:

current: The ESP32 is a bit more expensive than the ESP32.

fix: The ESP32 is a bit more expensive than the ESP8266.

[Reply](#)



Sara Santos

November 3, 2022 at 3:27 pm

Right!

Thanks. It's fixed now.

Regards,

Sara

[Reply](#)



David D



Hello,

I have tried and tried many different things and am unable to get an ESP32 board to send a message via WiFi to a Raspberry Pi. My goal is to sense a switch position and send a message indicating if the switch is closed or open. I have code that works well for sensing position but no matter what I try I can't get it to successfully send a message to the Raspberry Pi. I have successfully published and subscribed via local host on the Raspberry Pi. I think I have successfully sent a message to the WiFi but can't get the Raspberry Pi to see it???

I put together a Word document that shows one of the many things I have tried and would like to send it to you to see if you have any ideas on things to try to get it working.

Thanks in advance for any help you could offer.

David

[Reply](#)



**Mohd Shahid**

November 9, 2022 at 3:57 am

can we have the same steps for esp32s module?

[Reply](#)



Hi Santos's

I love your work and have purchased items of interest. But it took me a long time to start to use my ESP32's that I bought when they first hit the market. I have always used Sublime as my editor of choice when working with all the Arduino's but hated the Arduino IDE. Over the last bunch of years I have tried lots of other editors with bad success. I went back to SublimeText editor and found they changed there add on for Arduino to include Platformio as its base but keeps the simple Sketch directory structure and hides the Platformio layout. I have never been happier. Have you tried the SublimeText editor with the Deviot Arduino extension? Keep up the good work. FYI-my latest project is using 7 ESP32 using ESP-NOW

[Reply](#)



**Sara Santos**

November 11, 2022 at 5:41 pm

Hi.

Never tried PlatformIO with Sublime text. I have to take a look at it. Many of our readers like Sublime text too.

My favorite method is using VS Code with the PlatformIO extension.

Regards,

Sara

[Reply](#)



**Expired Expressif**

November 30, 2022 at 8:50 pm



the address for installing ESP boards are the obsolete ones – even thou the image below shows the correct address :))

these are no more maintained, you'll get the latest version 1..6 instead of 2.0.5:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json),

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

[Reply](#)



**Sara Santos**

November 30, 2022 at 10:00 pm

Right.

Thanks for noticing.

I already fixed that.

Regards,

Sara

[Reply](#)



**Bernie**

December 14, 2022 at 9:02 am

Hi Sara and Rui –



What do the RTC pins do? And what is EMAC?

Thanks

[Reply](#)



**Sara Santos**

December 14, 2022 at 10:28 pm

Hi.

Read the following tutorials:

- <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>
- <https://randomnerdtutorials.com/esp32-spi-communication-arduino/#esp32-spi-peripherals>

Regards,

Sara

[Reply](#)



**Richard Nault**

December 27, 2022 at 4:08 am

PROBLEM: I have not been able to send AT commands to my ESP32 (KeeYees, CP2102)

CONTEXT: Arduino IDE v1.8.19 under Mac OS Catalina

I am able to upload sketches, print on Serial Monitor have used many different sketches to control leds, temperature sensors, etc.

ABOUT AT COMMANDS: I am able to read characters from the text box of



EXTRAS: I tried Terminal + Screen and the Serial app from decisivetactics and in both cases, I do connect with the ESP32, I do receive strings printed by a basic sketch but no feedback at all when I hit the send button with any AT command.

What am I doing wrong?

Thank you for your excellent work!

[Reply](#)



Alexandre

January 6, 2023 at 6:08 pm

Hi.

First I want to say thank you to Rui and Sara for their quality tutorials. I initially didn't pay attention to detail in the console when uploading a sketch to my esp32 board, and that was until I got to the step where I should learn how to store webpage files in the board's flash memory and then access it using the SPIFFS tool. When I uploaded my html, CSS and JS files into the board, then the server program sketch, I observed a very strange behavior from the board.

Trying to dig a little, I discovered that my board uses the ESP32 D0WD V3 core (so doesn't have onboard flash memory). I don't know if this behavior is said to lack flash memory; that's why I seek advice of an Expert in the field. To clarify, when I just upload the files of my webpage to the board and then a simple sketch that checks if these files are present, it responds successfully. But when I add WiFi functionality to this sketch, the board can't even connect to the network; whereas when I use the same WiFi configuration without including the SPIFFS library (just for a test), the board doesn't even take more than 3 seconds to connect to the network



program, it is indicated that the sketch takes up 11% of RAM and 62% of ROM. I use platformIO.

Your reactions will greatly relieve me. Cordially,  
Alexandre

[Reply](#)



**Sara Santos**

January 6, 2023 at 6:53 pm

Hi.

I'm sorry, but I'm not familiar with that issue.

Have you tried using a different board?

It might be a hardware problem probably... Can you share a link to your board?

Regards,

Sara

[Reply](#)



**Aavesh bagwan**

March 1, 2023 at 3:07 pm

Hii,

I was trying to add esp32 boards to arduino ide via board manager  
but i keep getting this errors

Error downloading [https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)



didn't opened.

i don't know why but this link in not opening in my pc and laptop and that's why i am not able to add esp32 boards in arduino ide please help.

[Reply](#)



**chicoviski**

March 11, 2023 at 1:58 am

Rui, Sara, Greetings!

I really liked your tutorials, and the ability to explain that leaves few doubts.  
When will the ebooks be released in Portuguese.

Abraços do Brasil!!

[Reply](#)



**Sara Santos**

March 11, 2023 at 11:58 am

Hi.

Thank you for your nice words.

Unfortunately, our Portuguese-speaking public is very reduced.

So, I don't think we'll have a Portuguese version.

Regards,

sara





**James Cotter**

March 31, 2023 at 6:35 pm

is there a trick to powering the esp-wroom-32 via the vin port? I used 5v and would not work, I only am able to power the board via a the micro usb port...?

[Reply](#)



**Sara Santos**

April 1, 2023 at 10:13 am

Hi.

Did you also connect the GND?

Regards,

Sara

[Reply](#)



**Mark**

May 16, 2023 at 5:54 pm

Along the same path, our board powers up and runs just fine using the Micro-USB or the 5V input, but just sets and stares back when supplied with 3.3V.



GND. Is there a pin that needs to be strapped high or low to let the ESP know where it's receiving its power from? Three different and new ESPs have been tried with identical results. The 5V input and on-board LDO can't be used due to the current demands on the 3.3V distribution (large LCD and surrounding component set).

Thanks

[Reply](#)



**tamba**

June 15, 2023 at 7:07 pm

Hi

thanks for this great tutorials!

I used info for my first esp32 and helped me a lot!  
but here is not info for newer esp32 like esp32-s3

[Reply](#)



**Beavis**

August 10, 2023 at 6:17 pm

From the tutorial at <https://randomnerdtutorials.com/getting-started-with-esp32/>

“At the time of writing this tutorial, we recommend using the legacy version

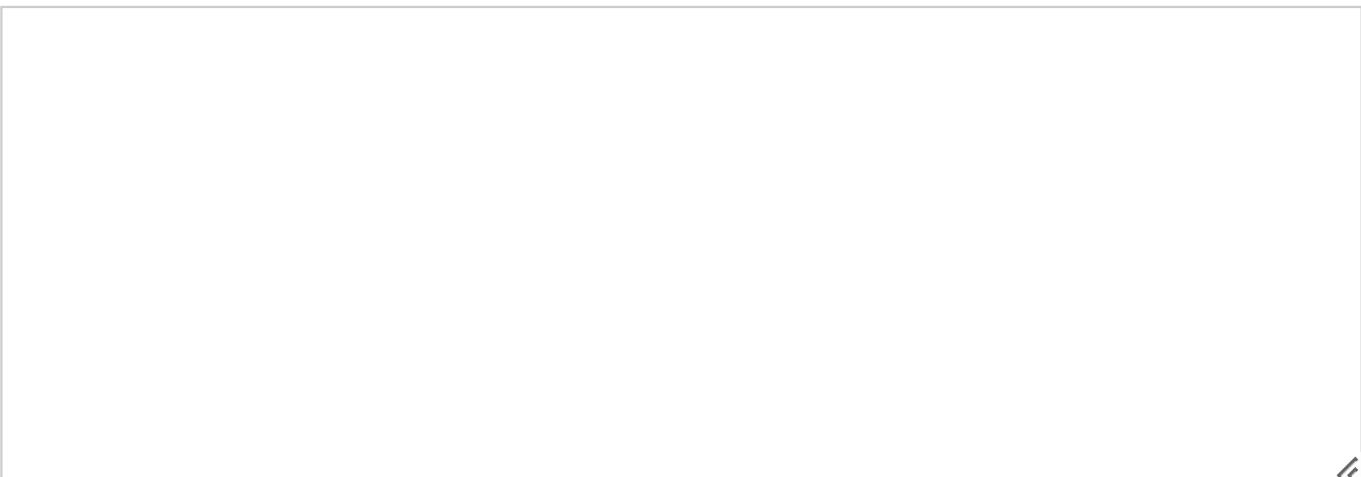


I could not find the date this tutorial was posted. Without any context, the above quoted excerpt is not helpful. Please include the date tutorials were posted on at the top of each page, I think it would be helpful to many.

Thanks!

[Reply](#)

## Leave a Comment

A large, empty rectangular area with a thin black border, likely a placeholder for a comment or message.

Name \*

Email \*

Website

Notify me of follow-up comments by email.

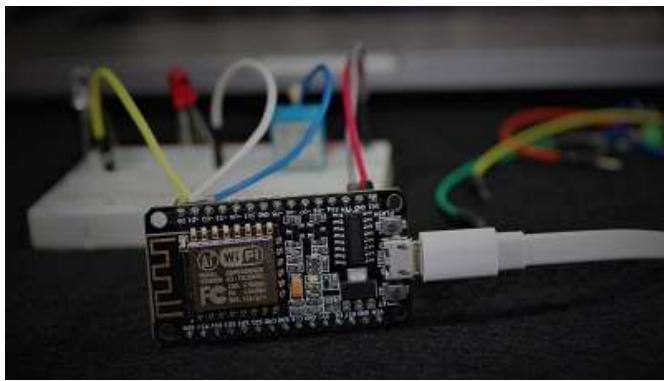
Notify me of new posts by email.

[Post Comment](#)

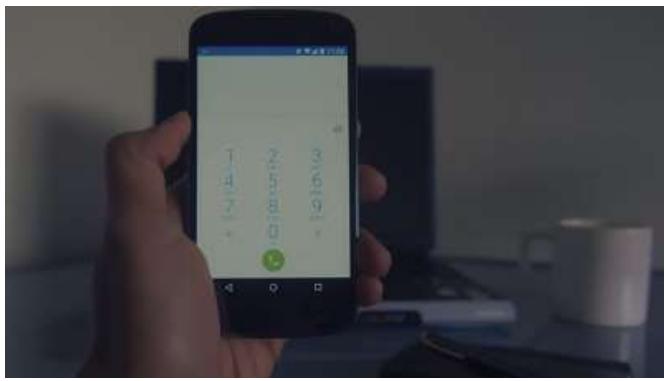




[Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »](#)



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



[Build Web Servers with ESP32 and ESP8266 » boards to control outputs and monitor sensors remotely.](#)









































































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

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

Copyright © 2013-2023 · RandomNerdTutorials.com · All Rights Reserved