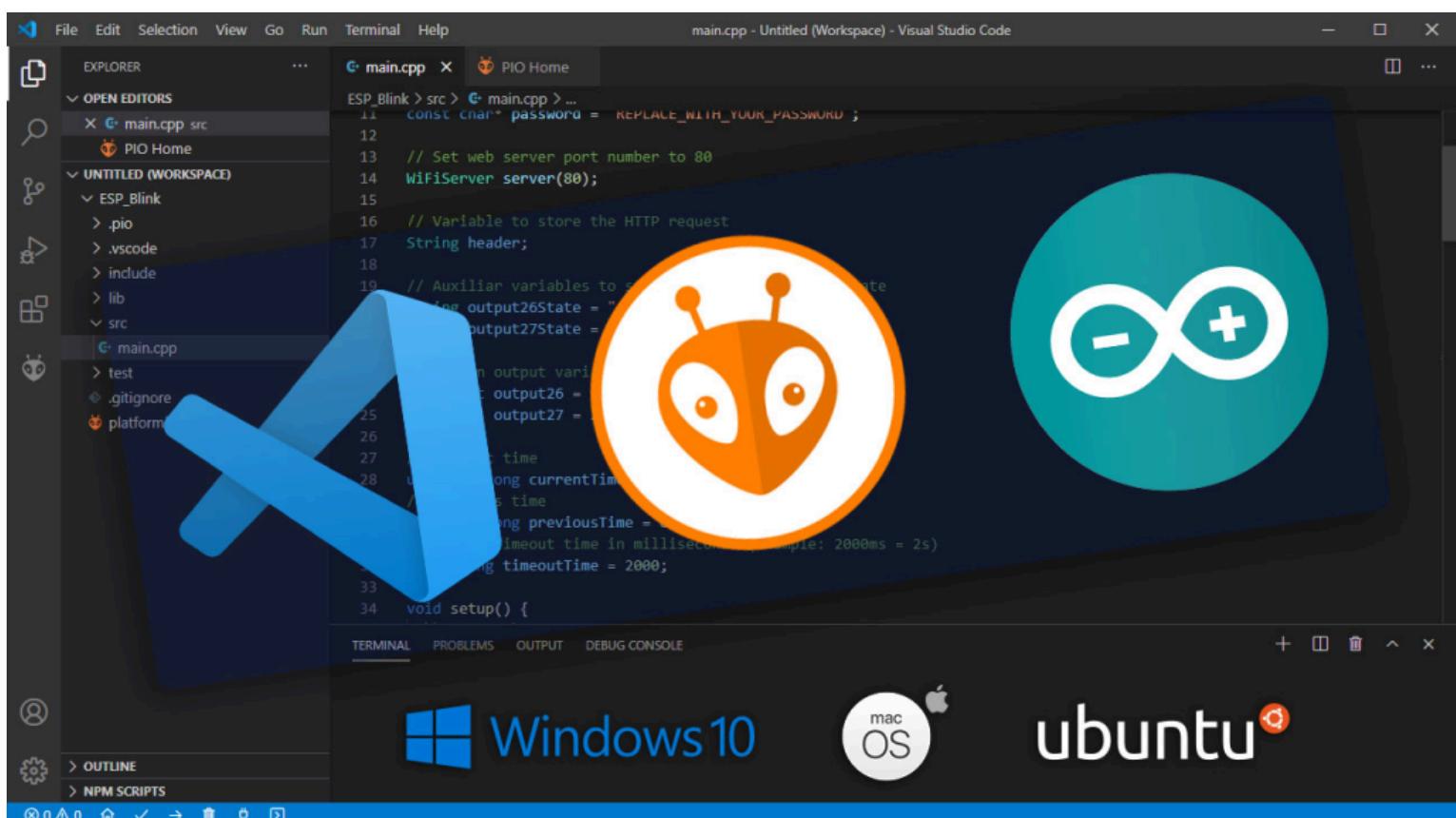


Getting Started with VS Code and PlatformIO IDE for ESP32 and ESP8266 (Windows, Mac OS X, Linux Ubuntu)

Learn how to program the ESP32 and ESP8266 NodeMCU boards using VS Code (Microsoft Visual Studio Code) with PlatformIO IDE extension. We cover how to install the software on Windows, Mac OS X or Ubuntu operating systems.



The Arduino IDE works great for small applications. However, for advanced projects with more than 200 lines of code, multiple files, and other advanced features like auto completion and error checking, VS Code with the PlatformIO IDE extension is the best alternative.

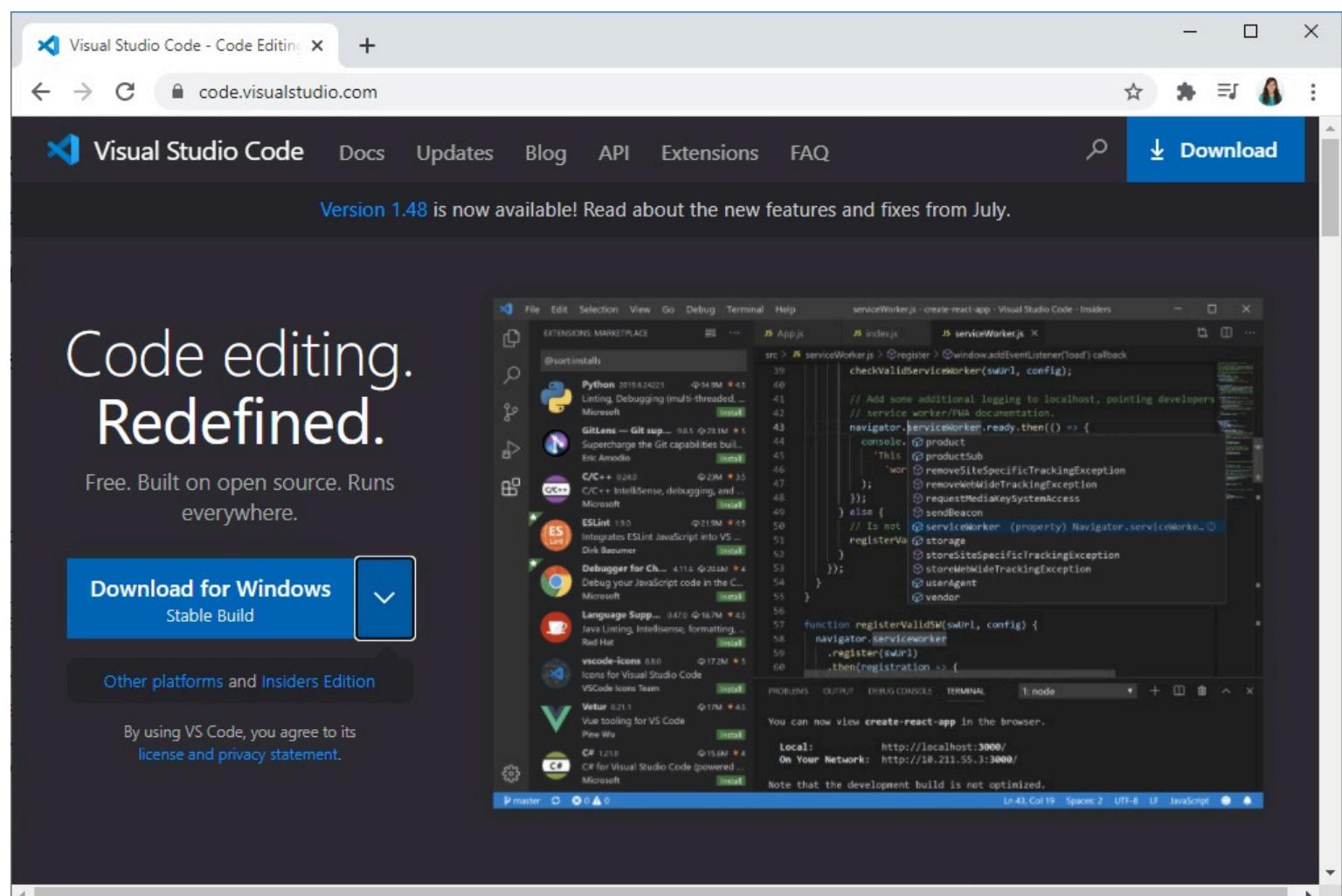
In this tutorial, we'll cover the following topics:

- Installing VS Code (Visual Studio Code):
 - [A\) Windows](#)
 - [B\) Mac OS X](#)
 - [C\) Linux Ubuntu](#)

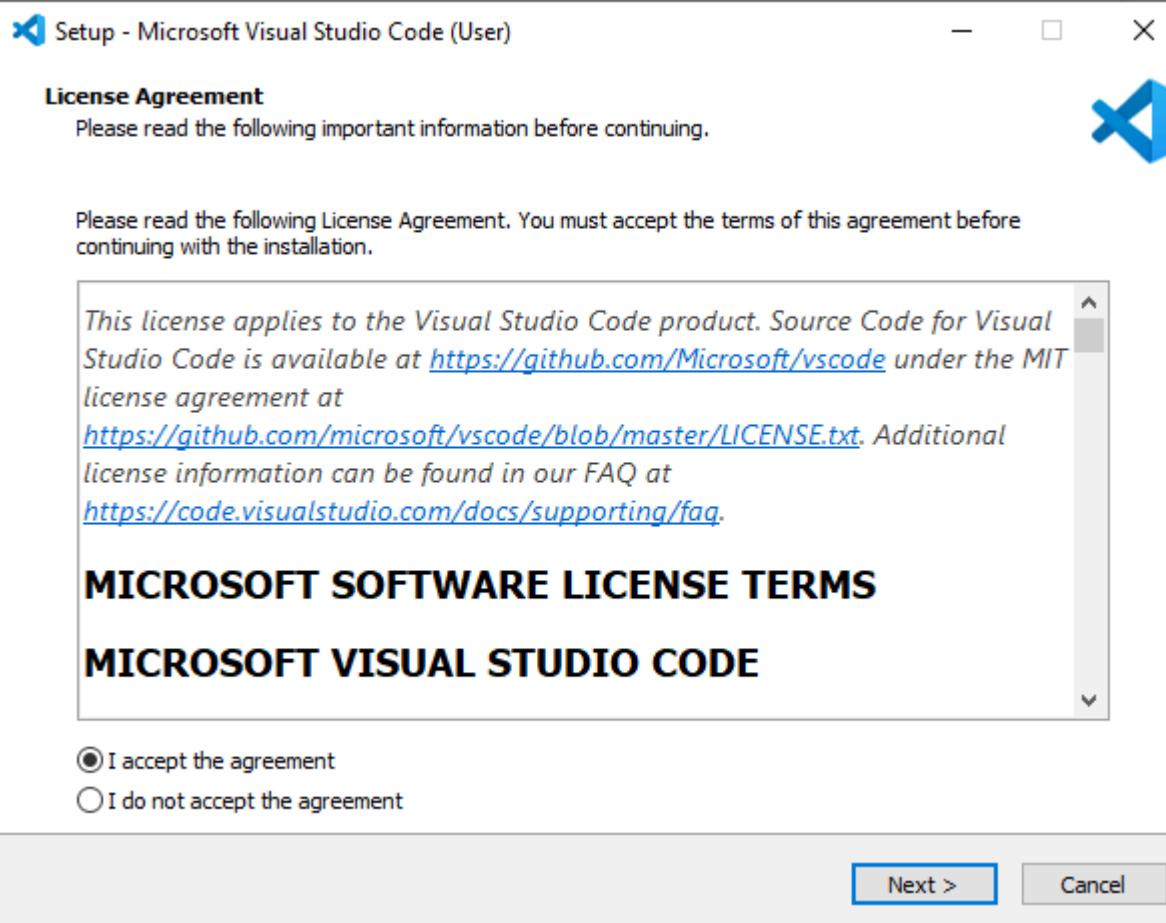
- Visual Studio Quick Interface Overview
- PlatformIO IDE Overview
- Uploading Code using PlatformIO IDE: ESP32/ESP8266
- Changing the Serial Monitor Baud Rate – PlatformIO IDE
- Installing Libraries on PlatformIO IDE

A) Installing VS Code on Windows (Visual Studio Code)

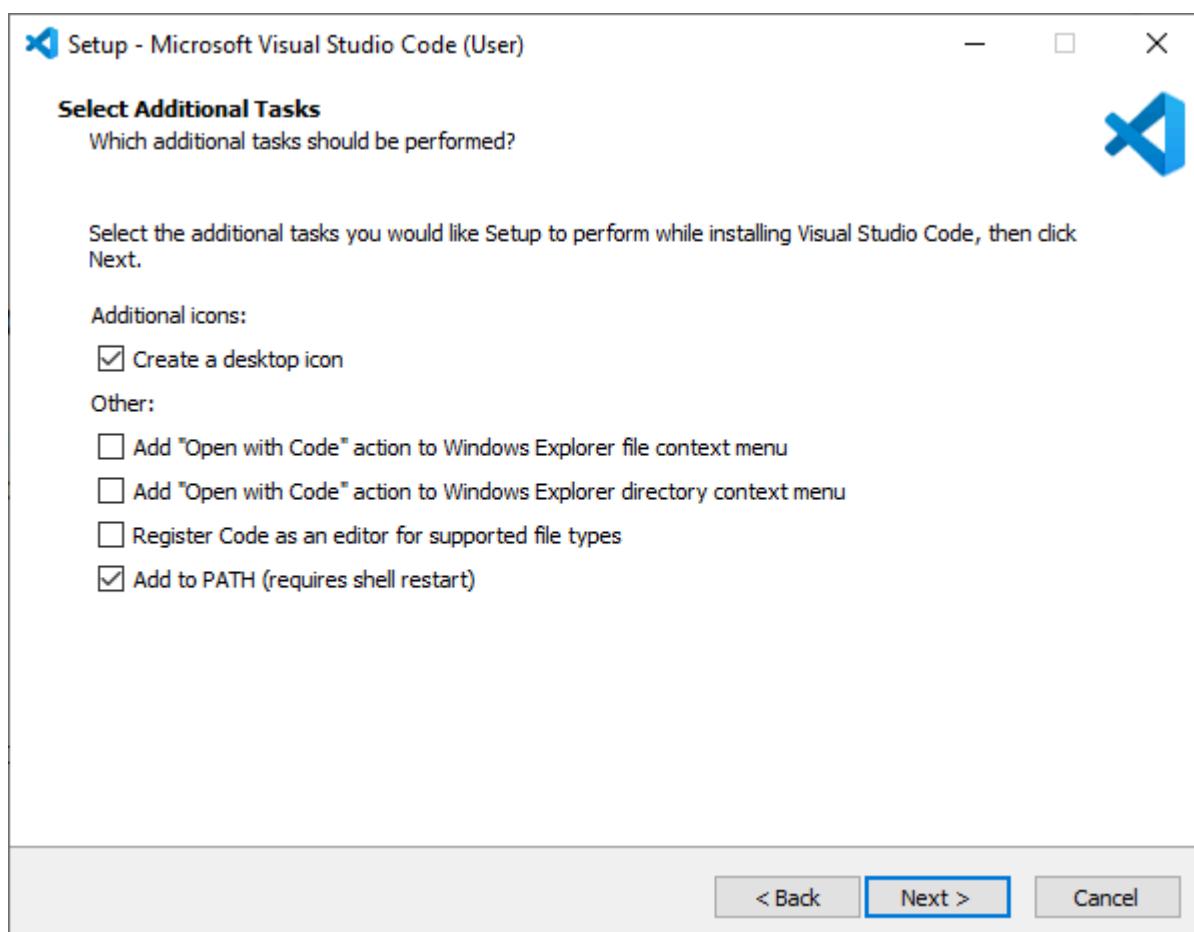
Go to <https://code.visualstudio.com/> and download the stable build for your operating system (Windows).



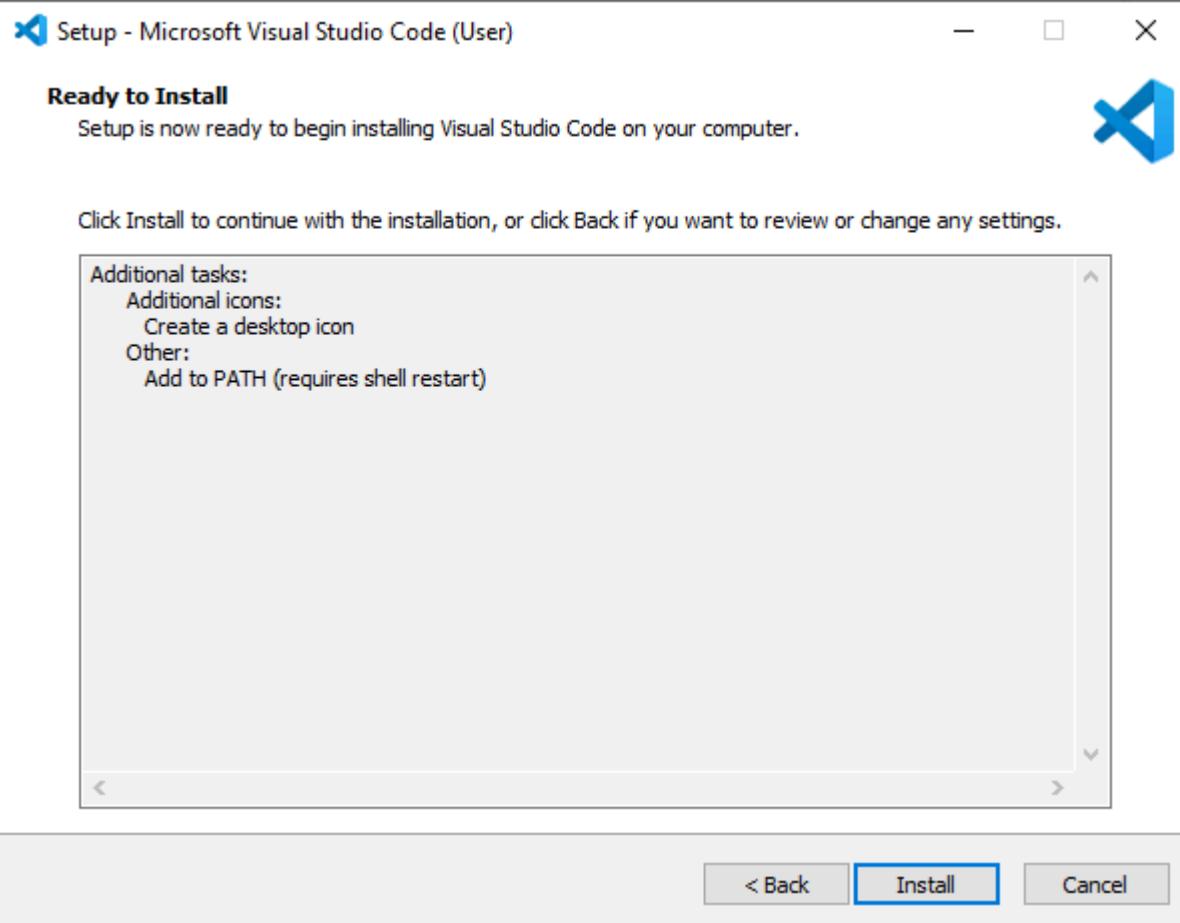
Click on the installation wizard to start the installation and follow all the steps to complete the installation. Accept the agreement and press the **Next** button.



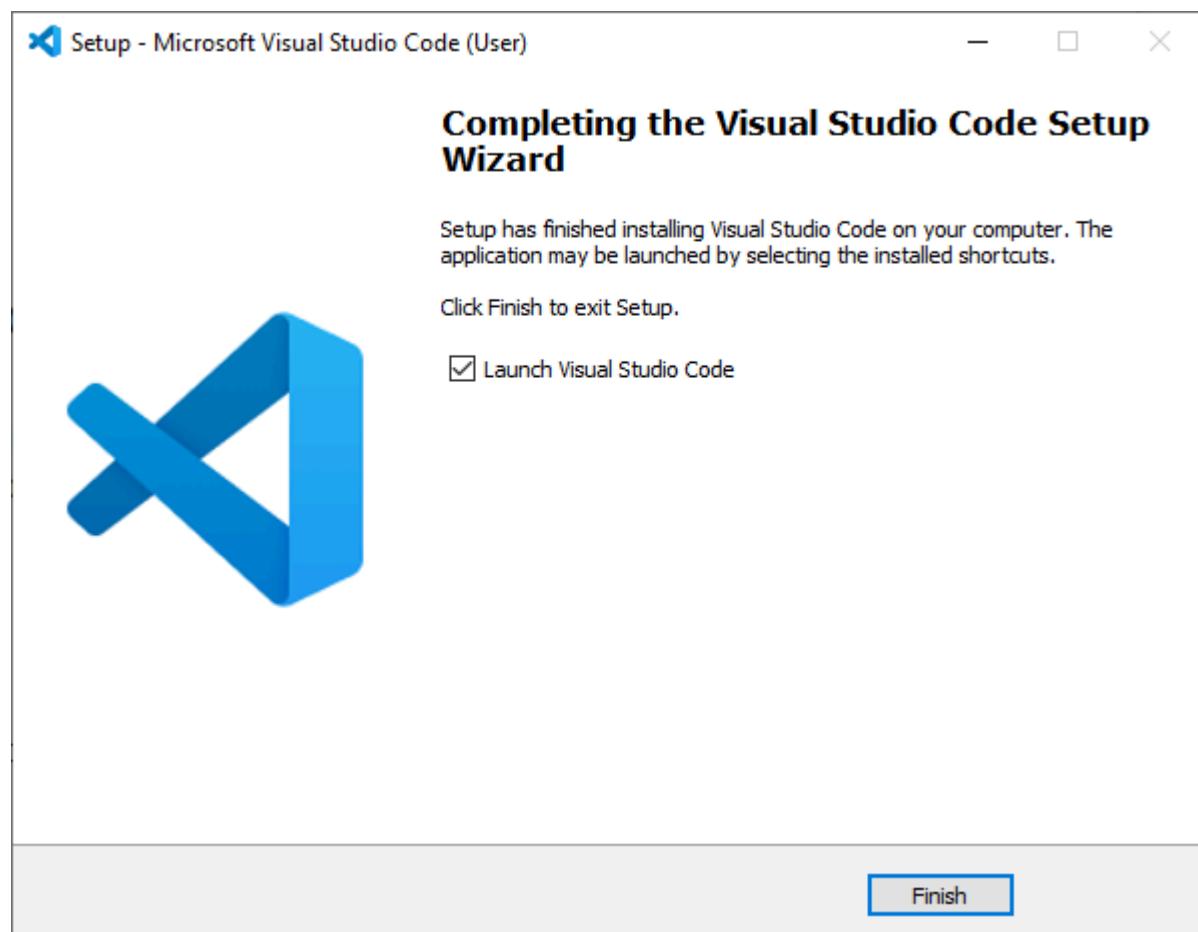
Select the following options and click **Next**.



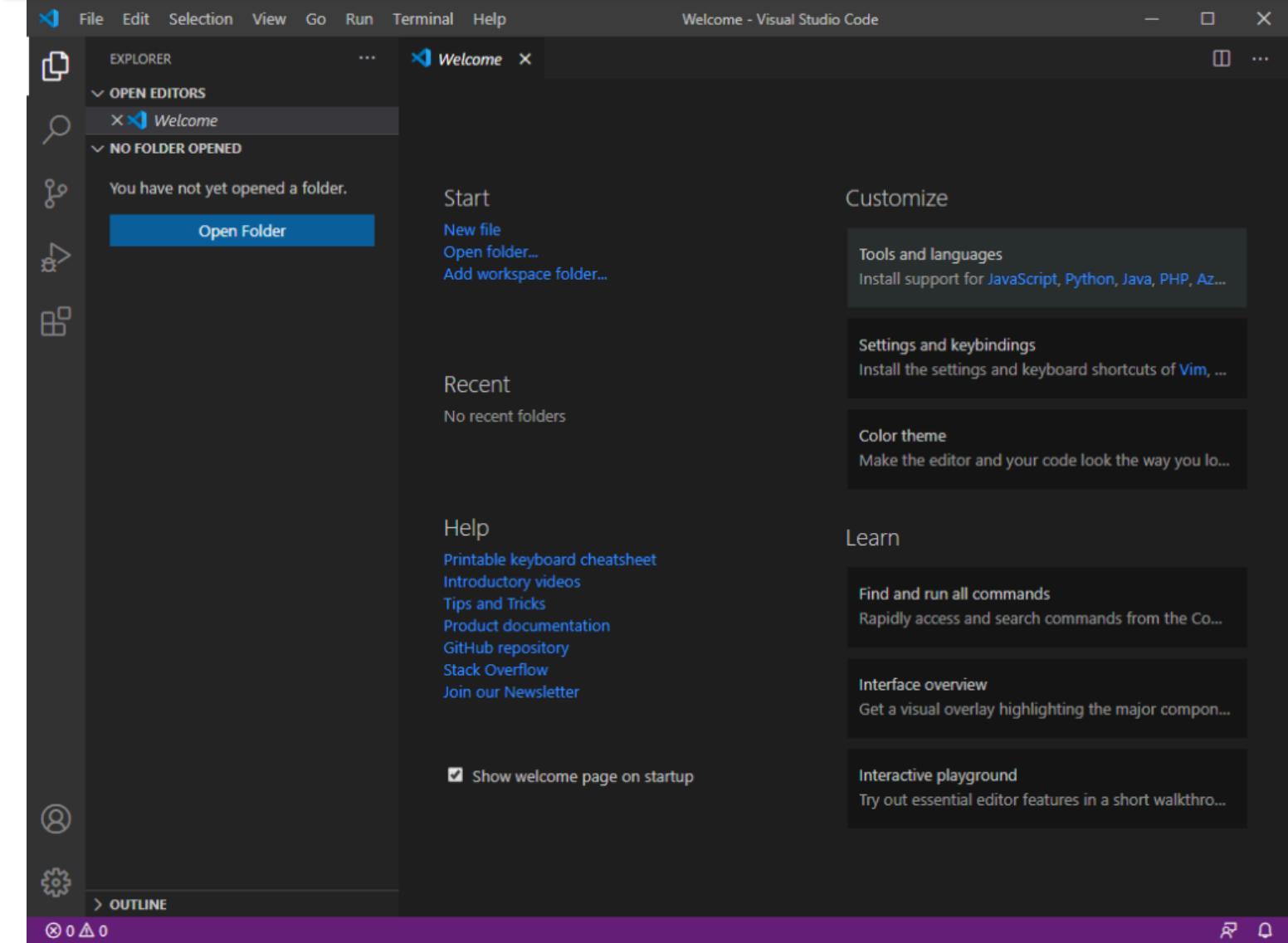
Press the **Install** button.



Finally, click **Finish** to finish the installation.



Open VS Code and you'll be greeted by a Welcome tab with the released notes of the newest



That's it. Visual Studio Code was successfully installed.

Installing Python on Windows

To program the ESP32 and ESP8266 boards with PlatformIO IDE you need Python 3.5 or higher installed in your computer. We're using Python 3.8.5.

Go to python.org/download and download Python 3.8.5 or a newest version.

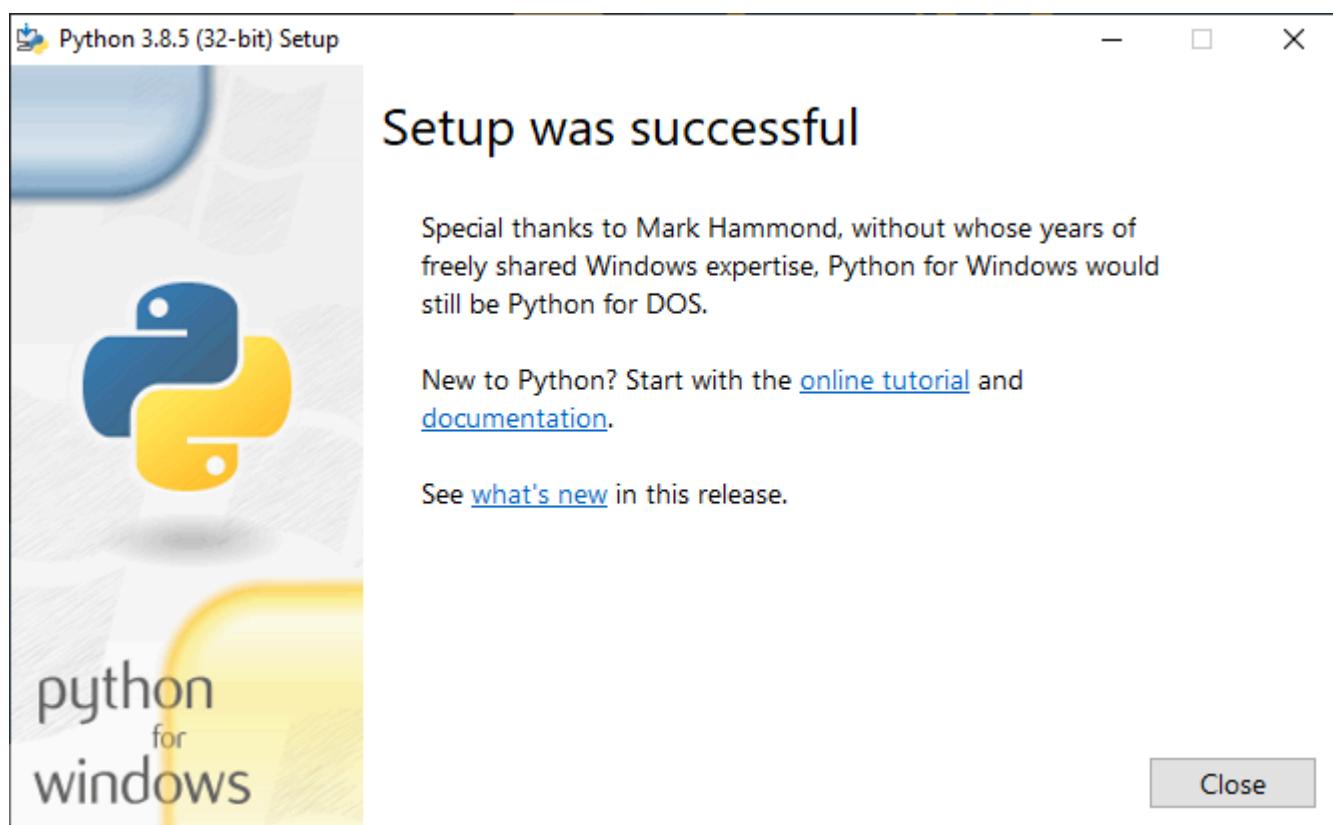
Open the downloaded file to start the Python installation wizard.

The following window shows up.



IMPORTANT: Make sure you check the option Add Python 3.8 to PATH. Then, you can click on the Install Now button.

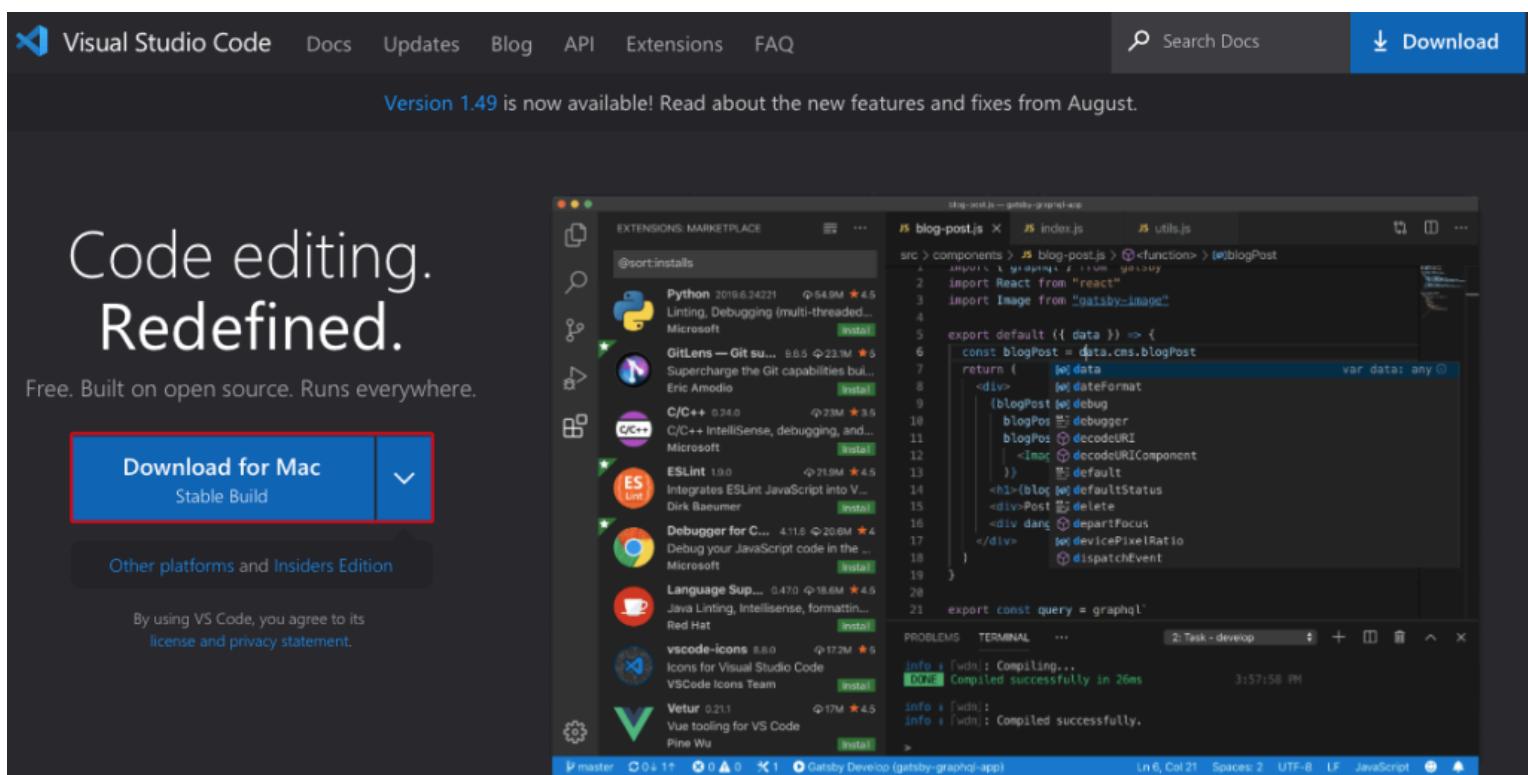
When the installation is successful you'll get the following message.



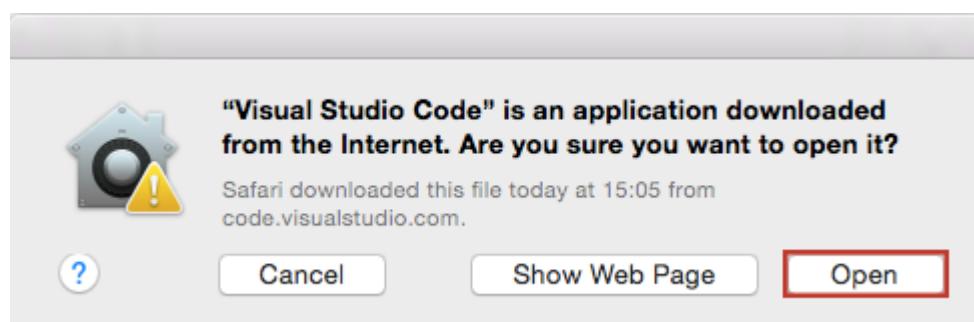
You can click the **Close** button.

B) Installing VS Code on Mac OS X (Visual Studio Code)

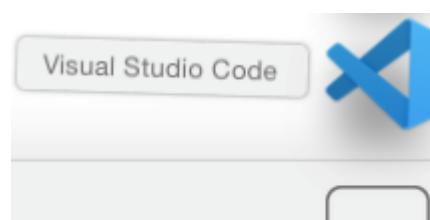
Go to <https://code.visualstudio.com/> and download the stable build for your operating system (Mac OS X).



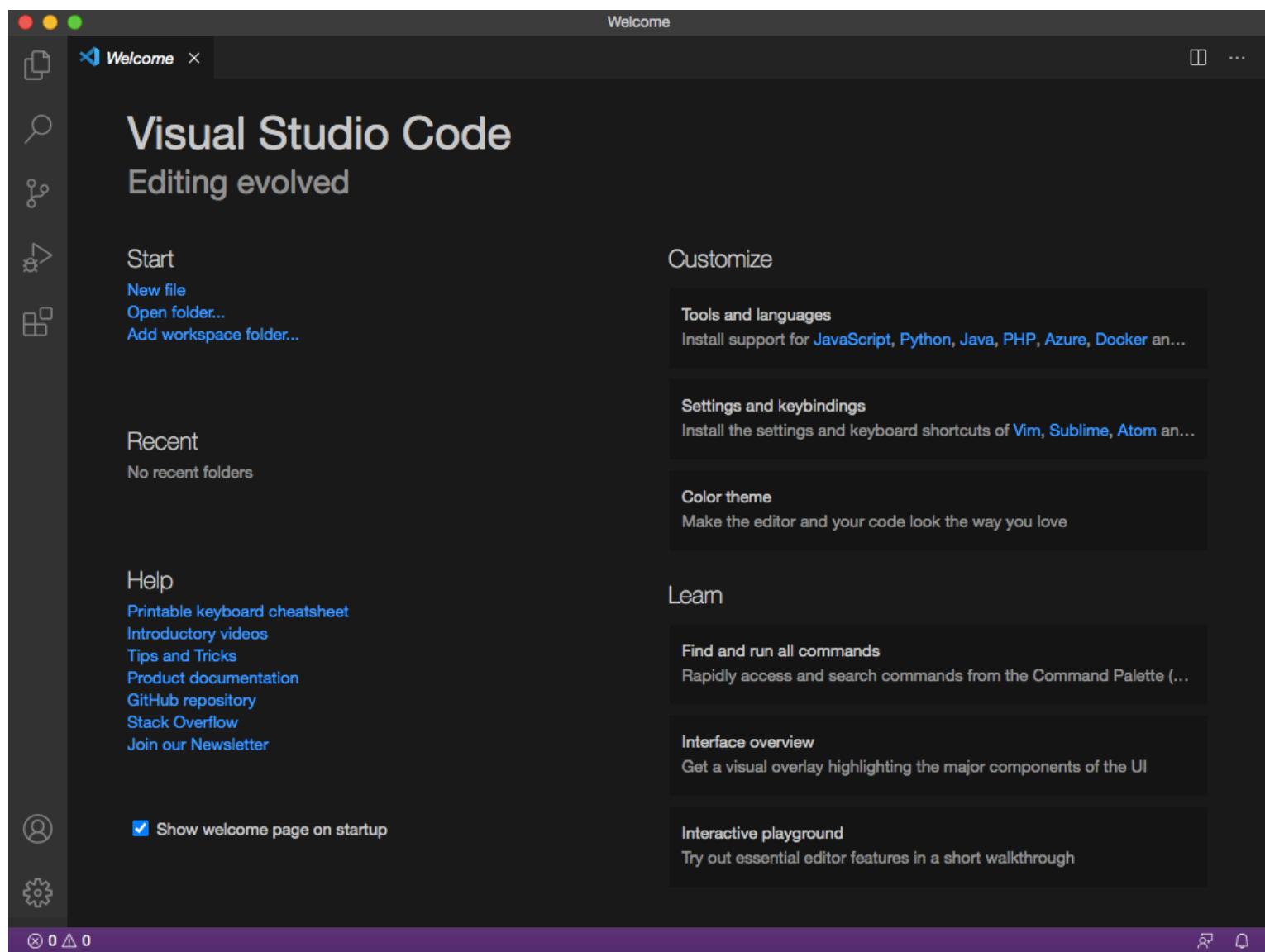
After downloading the Visual Studio Code application file, you'll be prompted with the following message. Press the “Open” button.



Or open your Downloads folder and open Visual Studio Code.



After that, you'll be greeted by a Welcome tab with the released notes of the newest version.



That's it. Visual Studio Code was successfully installed.

Installing Python on Mac OS X

To program the ESP32 and ESP8266 boards with PlatformIO IDE you need Python 3.5 or higher installed in your computer. We're using Python 3.8.5.

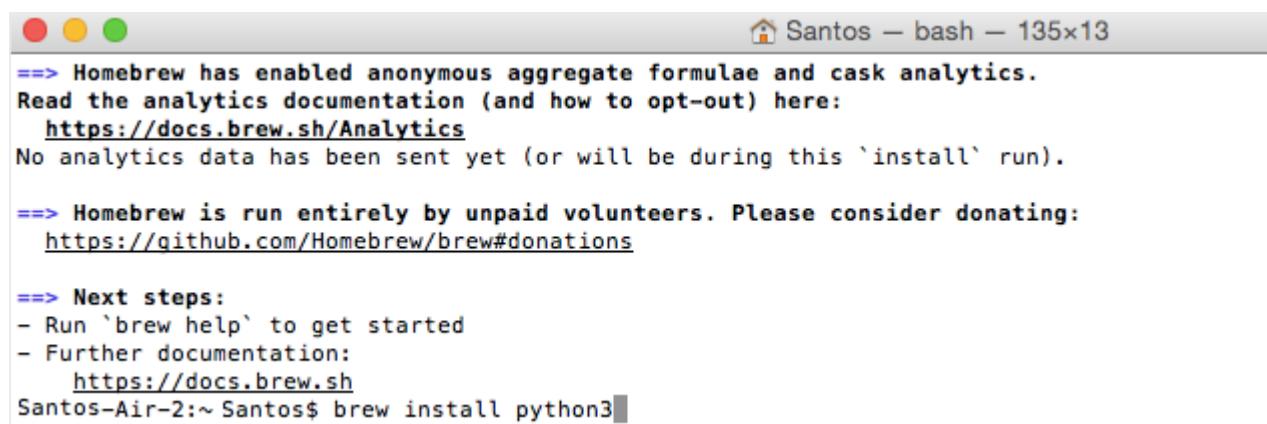
To install Python I'll be using Homebrew. If you don't have the brew command available, type the next command:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```



Then, run the brew command to install Python 3.X:

```
$ brew install python3
```



```
Santos — bash — 135x13
==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (or will be during this 'install' run).

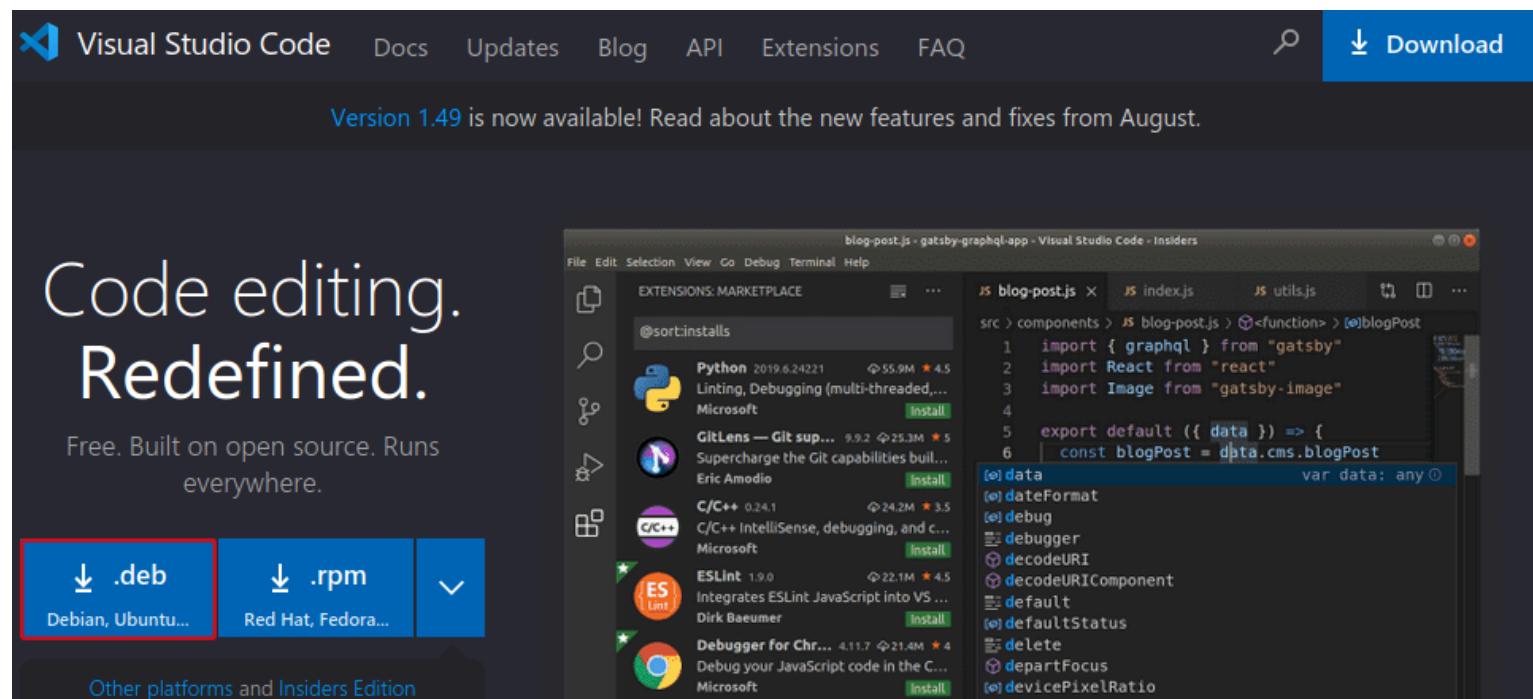
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh
Santos-Air-2:~ Santos$ brew install python3
```

Now, go to this section to install PlatformIO IDE extension.

C) Installing VS Code on Linux Ubuntu (Visual Studio Code)

Go to <https://code.visualstudio.com/> and download the stable build for your operating system (Linux Ubuntu).



The screenshot shows the official Visual Studio Code website. At the top, there's a navigation bar with links for Docs, Updates, Blog, API, Extensions, and FAQ. A search icon and a 'Download' button are also present. Below the navigation, a message says 'Version 1.49 is now available! Read about the new features and fixes from August.' On the left, there's a large promotional image with the text 'Code editing. Redefined.' and 'Free. Built on open source. Runs everywhere.' Below this are download links for '.deb' (Debian, Ubuntu...) and '.rpm' (Red Hat, Fedora...). A dropdown menu is shown, and at the bottom, there's a link for 'Other platforms and Insiders Edition'. On the right side of the screenshot, a portion of the Visual Studio Code interface is visible, showing a code editor with some JavaScript code and a sidebar for extensions like Python, GitLens, C/C++, ESLint, and Debugger for Chrome.

Save the installation file:

You have chosen to open:

code_1.49.1-1600299189_amd64.deb

which is: Debian package (62.2 MB)

from: <https://az764295.vo.msecnd.net>

What should Firefox do with this file?

Open with

Software Install (default)

Save File

Do this automatically for files like this from now on.

Cancel

OK

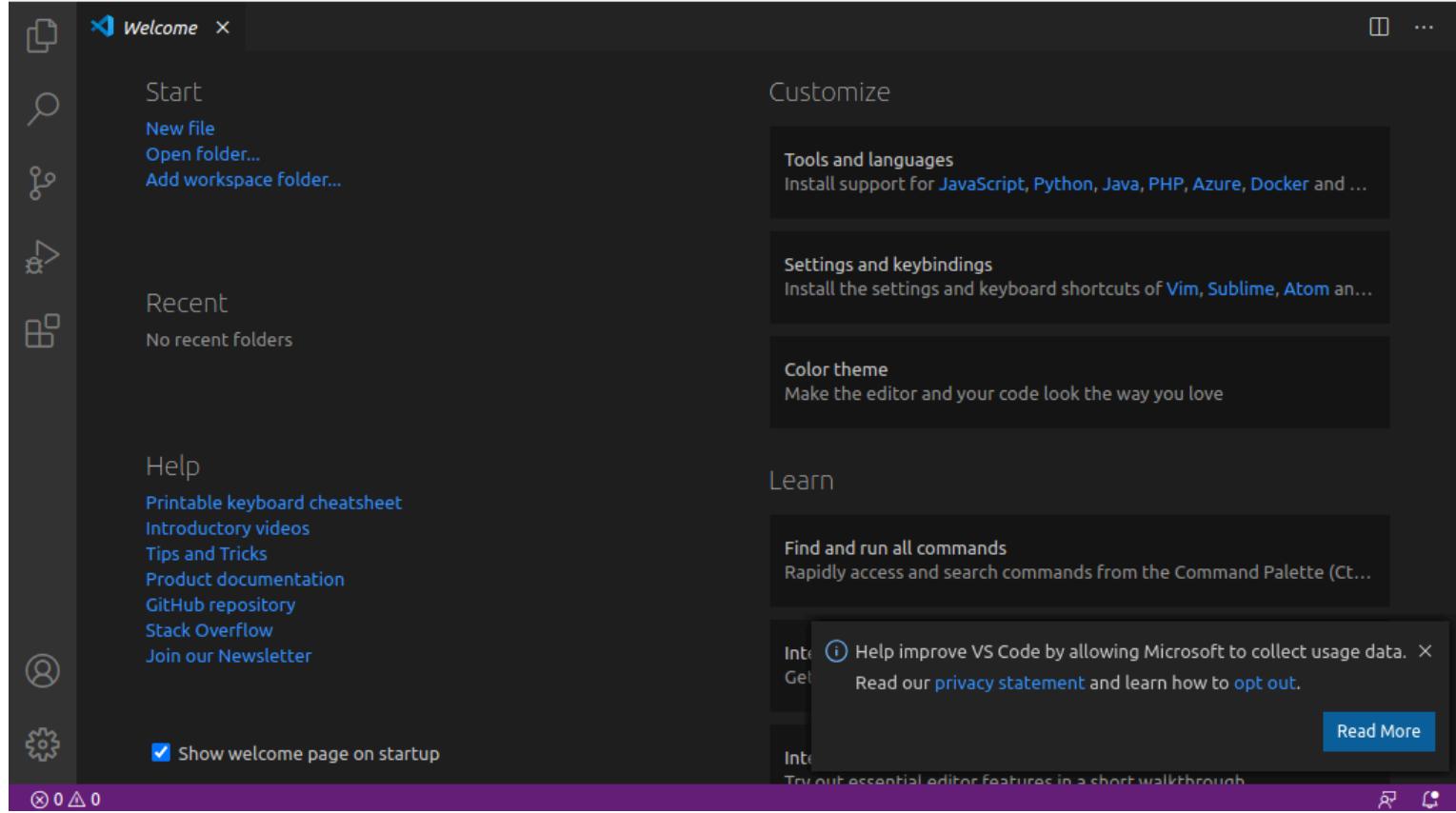
To install it, open a Terminal windows, navigate to your Downloads folder and run the following command to install VS Code.

```
$ cd Downloads  
~/Downloads $ sudo apt install ./code_1.49.1-1600299189_amd64.deb
```

When the installation is finished, VS Code should be available in your applications menu.



Open VS Code and you'll be greeted by a Welcome tab with the released notes of the newest version.



That's it. Visual Studio Code was successfully installed.

Installing Python on Linux Ubuntu

To program the ESP32 and ESP8266 boards with PlatformIO IDE you need Python 3.5 or higher installed in your computer. We're using Python 3.8.

Open the Terminal window and check that you already have Python 3 installed.

```
$ python3 --version
python 3.8.2
```

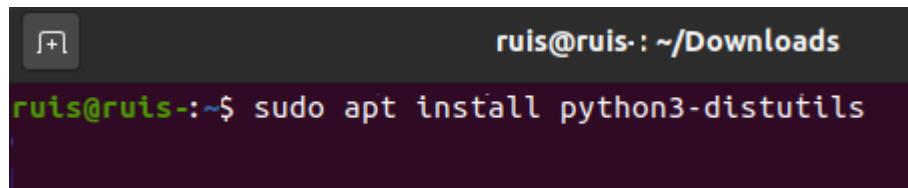
A terminal window with a dark background and light-colored text. It shows the command 'python3 --version' being run and the output 'Python 3.8.2'. The terminal window has a title bar with the user name 'ruis@ruis' and a prompt '\$'.

As you can see in the preceding figure, Python 3.8.2 is already installed.

```
$ sudo apt install python3
```

Whether you already have Python installed or not, you need to run the following command to install Python utilities.

```
$ sudo apt install python3-distutils
```



A screenshot of a terminal window. The title bar says 'ruis@ruis-: ~/Downloads'. The command 'sudo apt install python3-distutils' is typed in the terminal and is being processed.

Now, go to this section to install PlatformIO IDE extension.

Installing PlatformIO IDE Extension on VS Code

It is possible to program the [ESP32](#) and [ESP8266](#) boards using VS Code with the PlatformIO IDE extension. Follow the next steps to install the PlatformIO IDE extension.

Open VS Code:

1. Click on the **Extensions** icon or press **Ctrl+Shift+X** to open the **Extensions** tab
2. Search for “**PlatformIO IDE**”
3. Select the first option
4. Finally, click the **Install** button (Note: the installation may take a few minutes)

File Edit Selection View Go Run Terminal Help Extension: PlatformIO IDE - Visual Studio Code

EXTENSIONS: MARKETP... ⌂ ⌓ ...

Extension: PlatformIO IDE X

2 platformio ide

3 PlatformIO IDE 1.10.0 Development environment for Embed... PlatformIO Install

Bash IDE 1.11.0 A language server for Bash Mads Hartmann Install

4 IntelliJ IDEA Keybindings 0.2.41 Port of IntelliJ IDEA Keybindings, inclu... Keisuke Kato Install

1 COMP2300 IDE 2.0.9 PlatformIO IDE modified for COMP2300 comp2300-anu Install

Java IDE 0.1.1 To make the vscode work as Eclipse or... JavaHub Install

Serverless IDE 0.5.26 Enhanced support for AWS SAM, Clou... ThreadHeap Install

⊗ 0 △ 0

PlatformIO IDE [platformio.platformio-ide]

PlatformIO | ⌂ 1,045,617 | ★★★★★ | Repository

Development environment for Embedded, IoT, Arduino, ARM m...

Details Feature Contributions Changelog Dependencies

PlatformIO IDE for VSCode

A new generation toolset for embedded C/C++ development

PlatformIO is a new generation ecosystem for embedded development.

- Open source, maximum permissive Apache 2.0 license
- Cross-platform IDE and Unified Debugger

After installing, make sure that PlatformIO IDE extension is enabled as shown below.

File Edit Selection View Go Run Terminal Help Extension: PlatformIO IDE - Visual Studio Code

EXTENSIONS ⌂ ⌓ ...

Search Extensions in Marketplace

✓ INSTALLED 2

C/C++ 0.29.0 C/C++ IntelliSense, debugging, and c... Microsoft

PlatformIO IDE 1.10.0 Development environment for Embed... PlatformIO ✓ Enabled

PlatformIO IDE [platformio.platformio-ide]

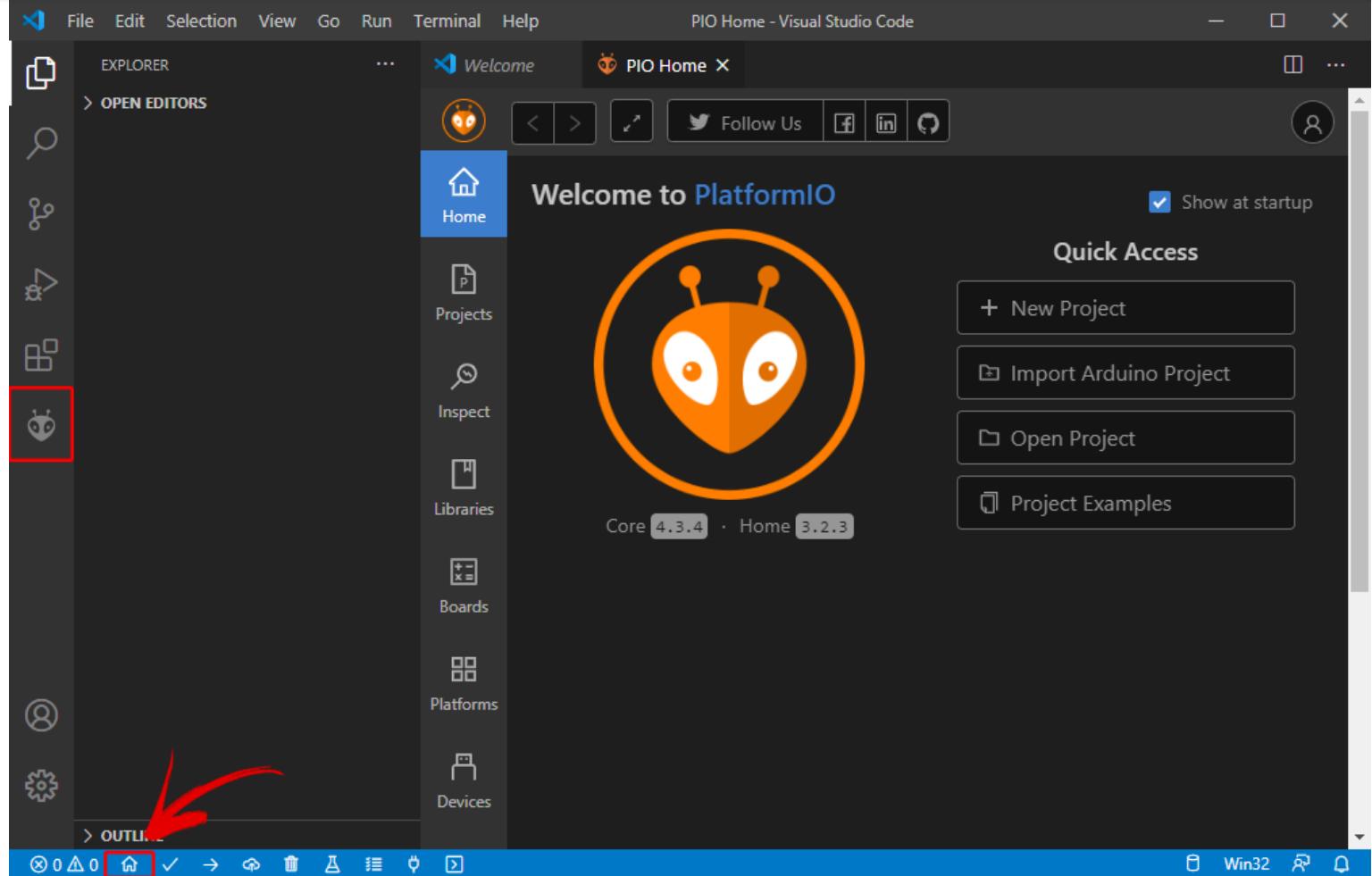
PlatformIO | ⌂ 1,045,744 | ★★★★★ | Repository | Lic

Development environment for Embedded, IoT, Arduino, ARM mbed, Esp...

✓ Enabled Disable Uninstall This extension is enabled globally.

Details Feature Contributions Changelog Dependencies

After that, the PlatformIO icon should show up on the left sidebar as well as an **Home** icon that redirects you to PlatformIO home.



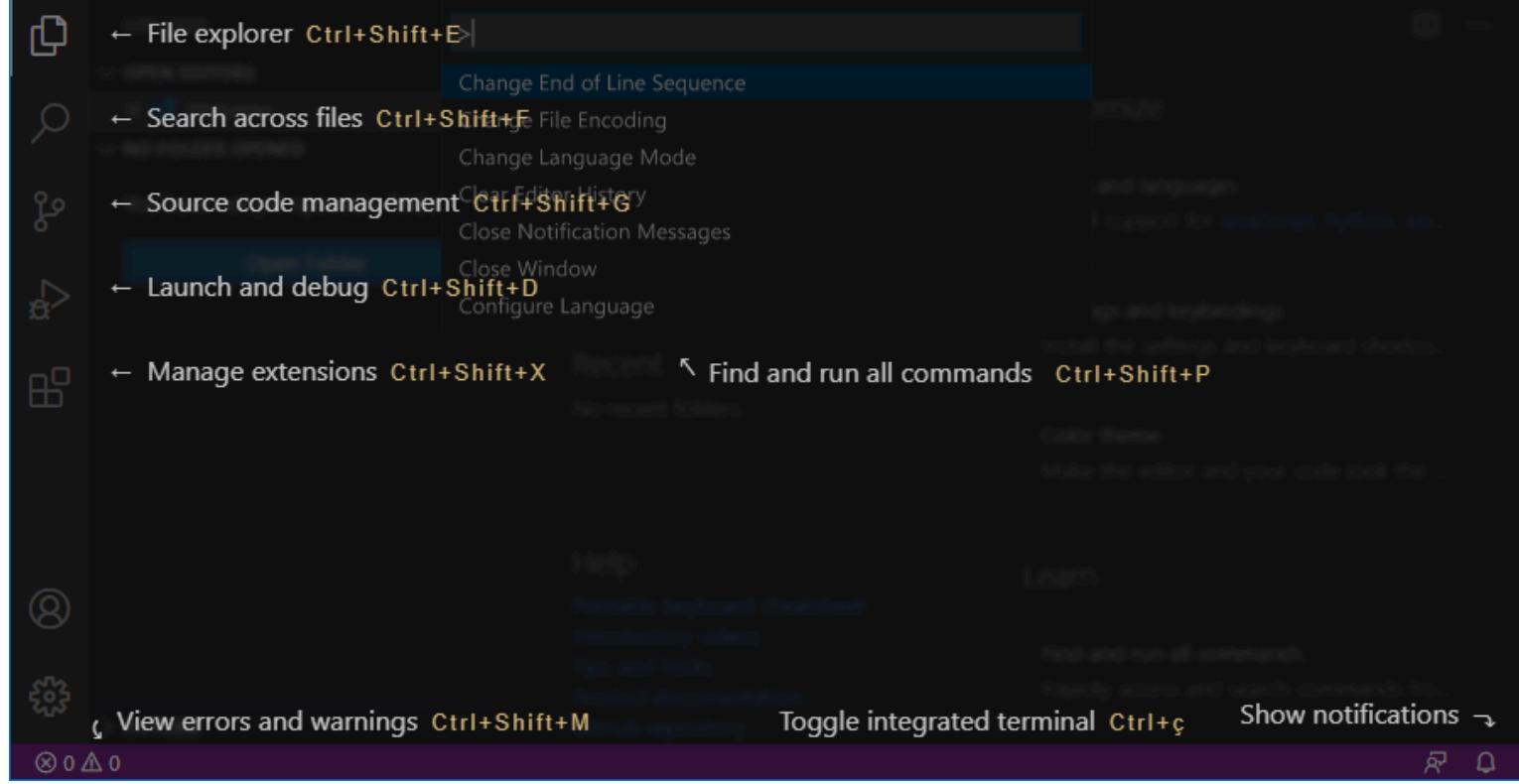
That's it, PlatformIO IDE extension was successfully added to VS Code.

If you don't see the **PIO** icon and the quick tools at the bottom, you may need to restart VS code for the changes to take effect.

Either way, we recommend restarting VS Code before proceeding.

VS Code Quick Interface Overview

Open VS Code. The following print screen shows the meaning of each icon on the left sidebar and its shortcuts:



- File explorer
- Search across files
- Source code management (using gist)
- Launch and debug your code
- Manage extensions

Additionally, you can press **Ctrl+Shift+P** or go to **View > Command Palette...** to show all the available commands. If you're searching for a command and you don't know where it is or its shortcut, you just need to go to the Command Palette and search for it.

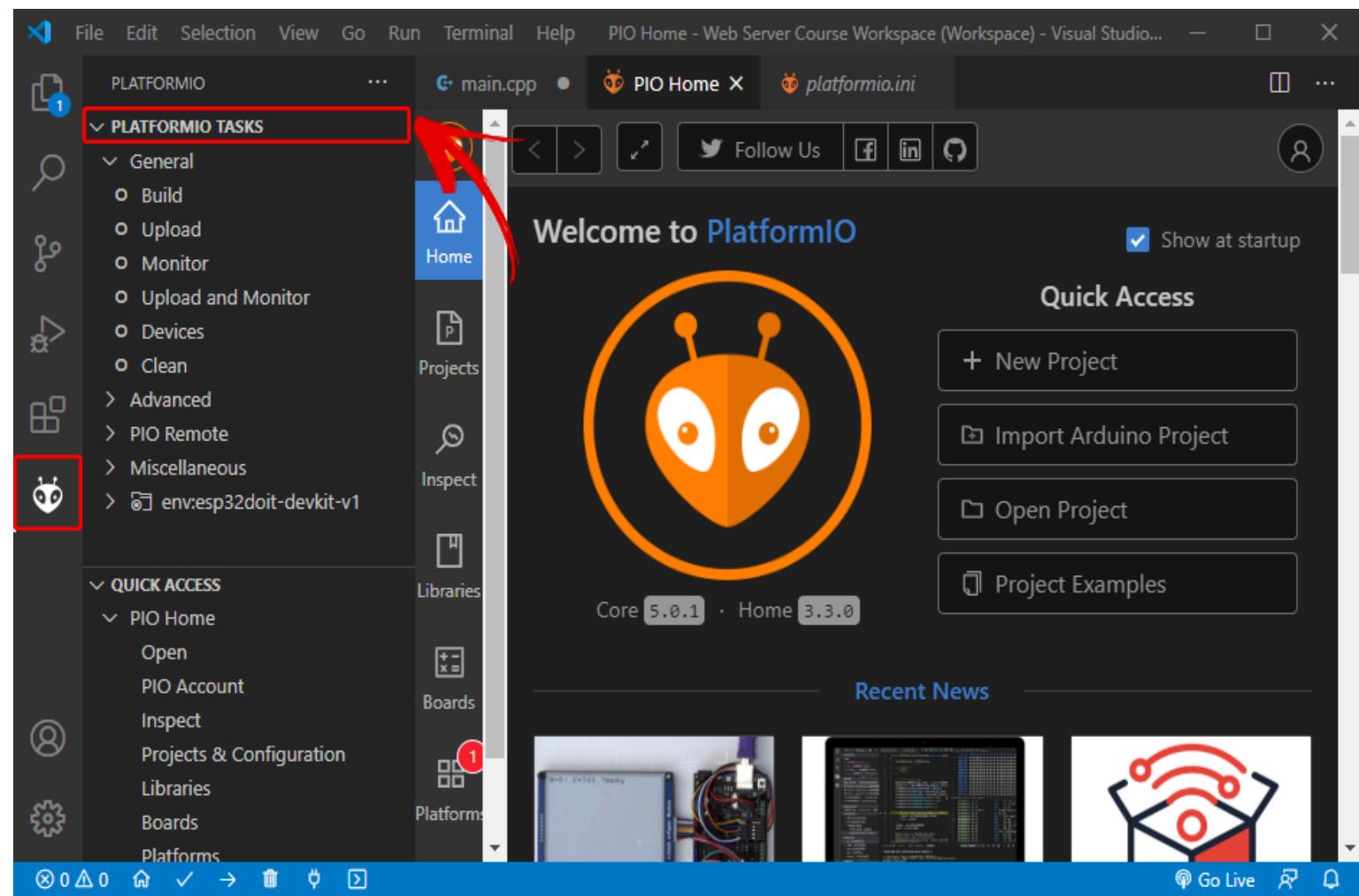
At the bottom, there's a blue bar with PlatformIO commands.



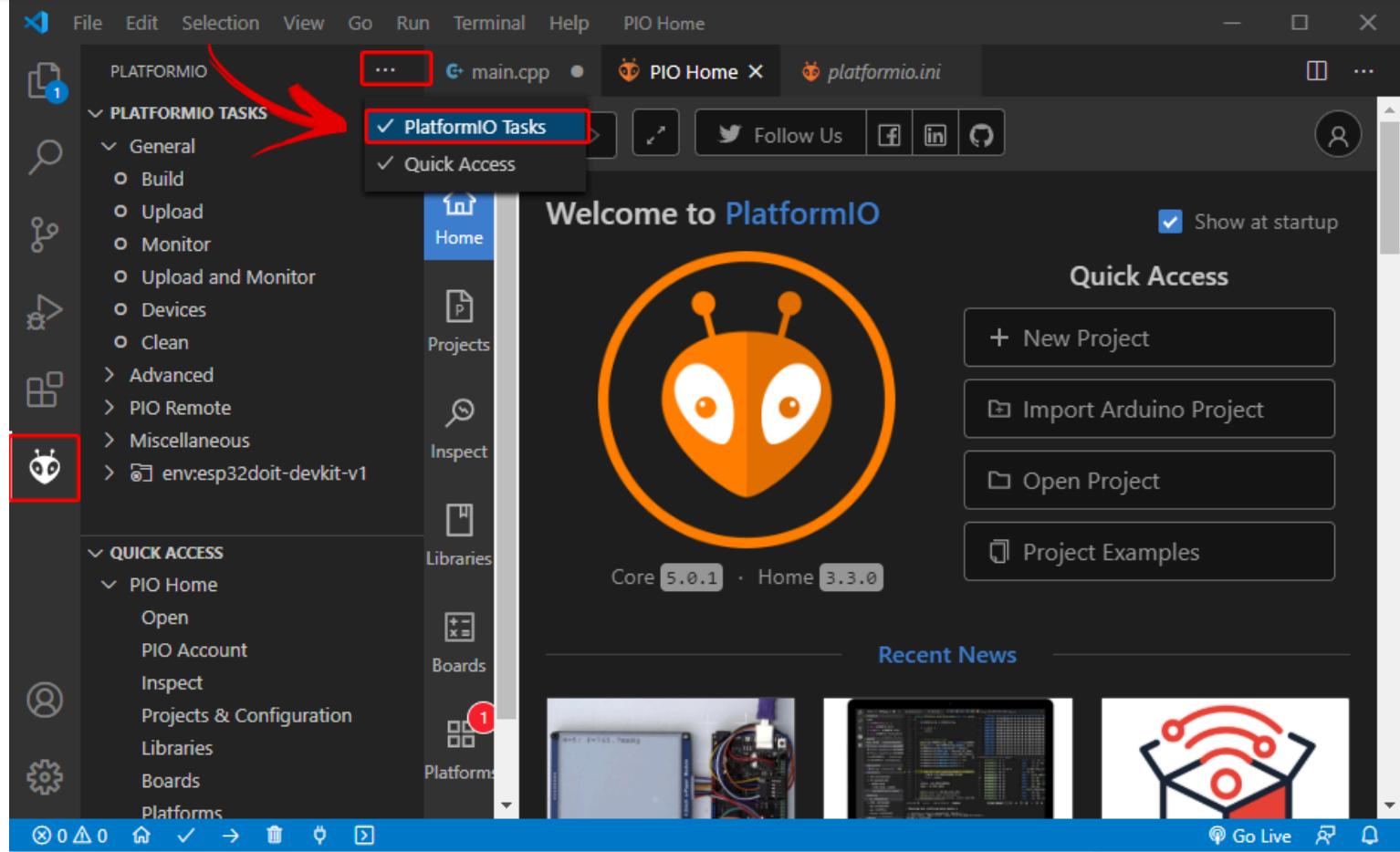
Here's the what icon does from left to right:

- PlatformIO Home
- Build/Compile
- Upload
- Clean
- Serial Monitor
- New Terminal

Alternatively, you can also click on the PIO icon to see all the PlatformIO tasks.



If the tasks don't show up on your IDE when you click the icon, you may need to click on the three dot icon at the top and enable PlatformIO tasks as shown below.

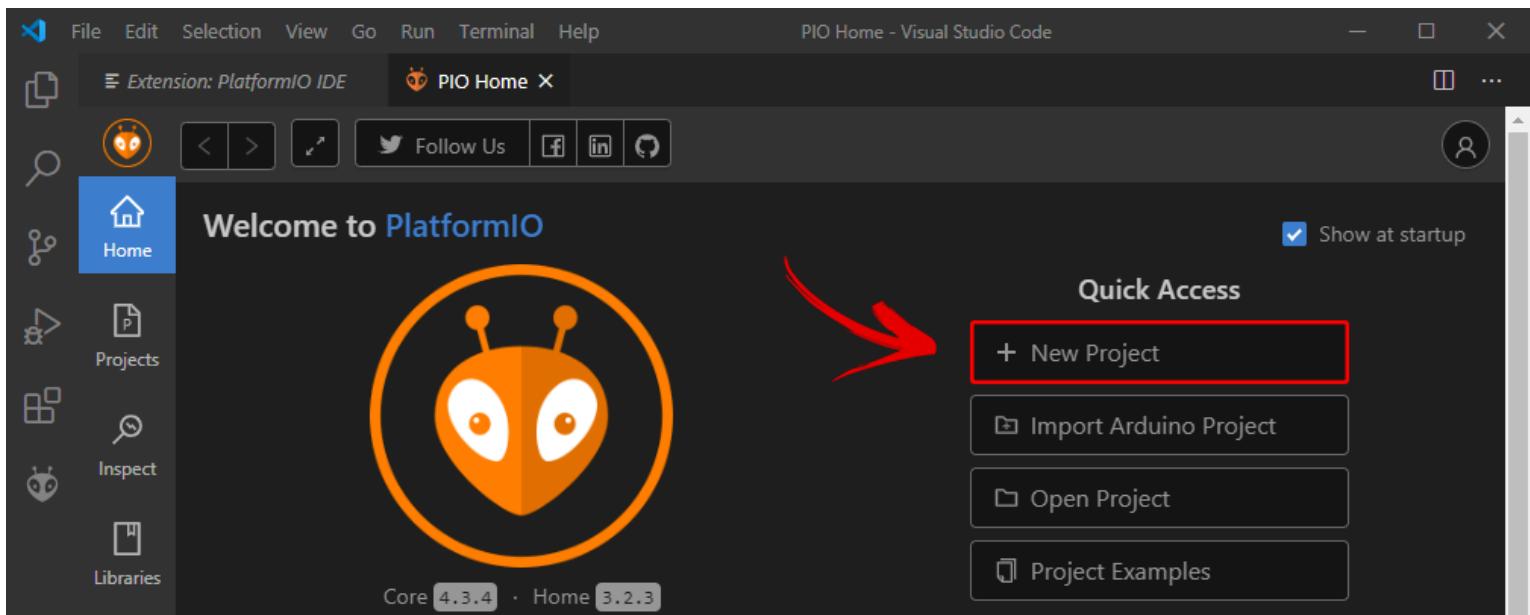


PlatformIO IDE Overview

For you to get an overview on how PlatformIO works on VS code, we'll show you how to create, save and upload a "Blinking LED" sketch to your ESP32 or ESP8266 board.

Create a New Project

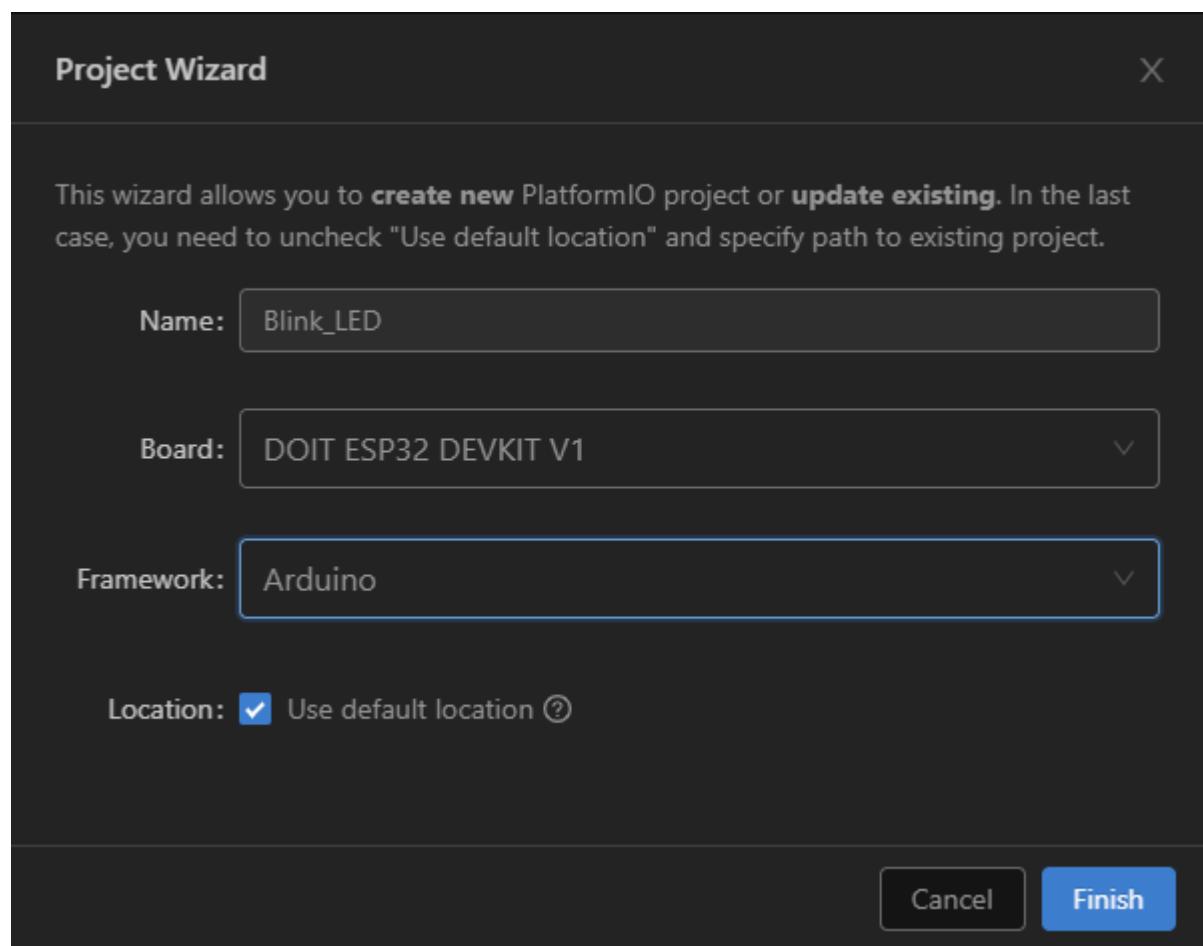
On VS Code, click on the PlartfomIO Home icon. Click on **+ New Project** to start a new project.



Give your project a name (for example *Blink_LED*) and select the board you're using. In our case, we're using the **DOIT ESP32 DEVKIT V1**. The Framework should be “**Arduino**” to use the Arduino core.

You can choose the default location to save your project or a custom location.

The default location is in this path *Documents >PlatformIO >Projects*. For this test, you can use the default location. Finally, click “Finish”.



For this example, we'll be using the **DOIT ESP32 DEVKIT board**. If you are using an **ESP8266 NodeMCU board** the process is very similar, you just need to select your ESP8266 board:

Project Wizard

X

This wizard allows you to **create new** PlatformIO project or **update existing**. In the last case, you need to uncheck "Use default location" and specify path to existing project.

Name: Blink_LED

Board: Espressif ESP8266 ESP-12E

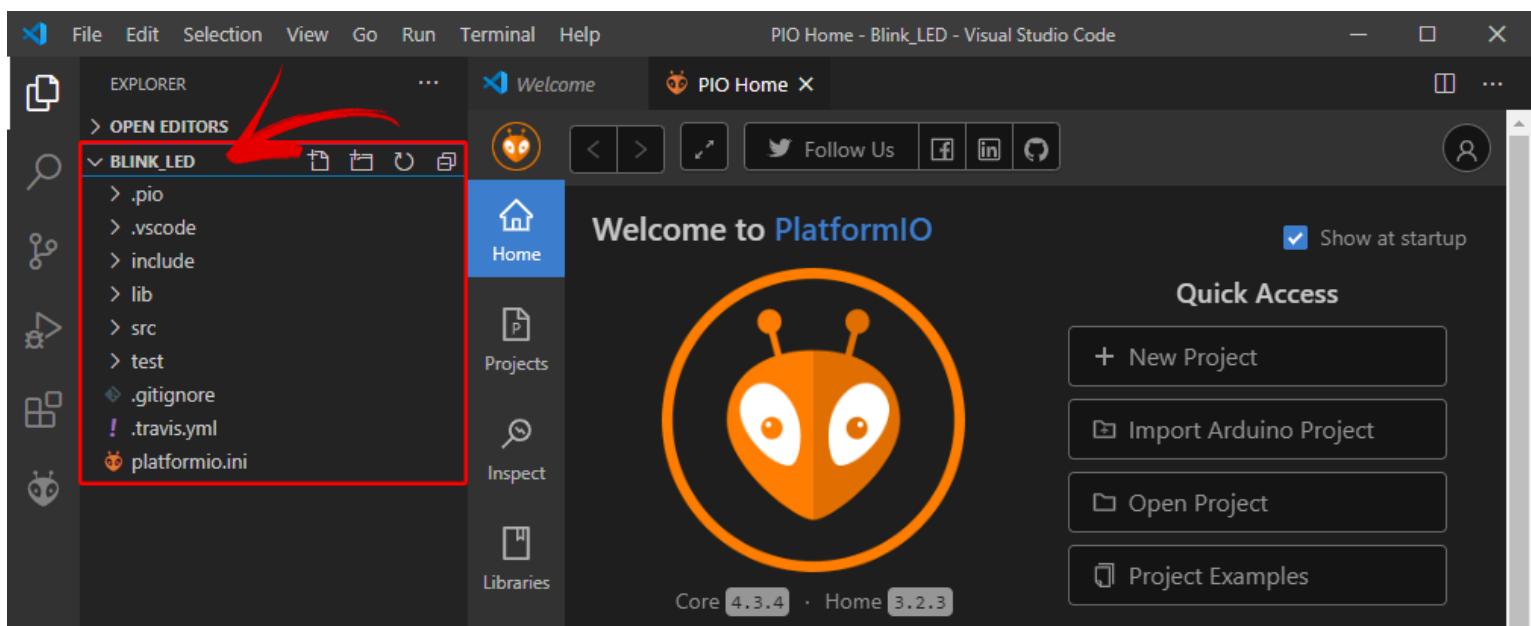
Framework: Arduino

Location: Use default location ⓘ

Cancel

Finish

The Blink_LED project should be accessible from the Explorer tab.

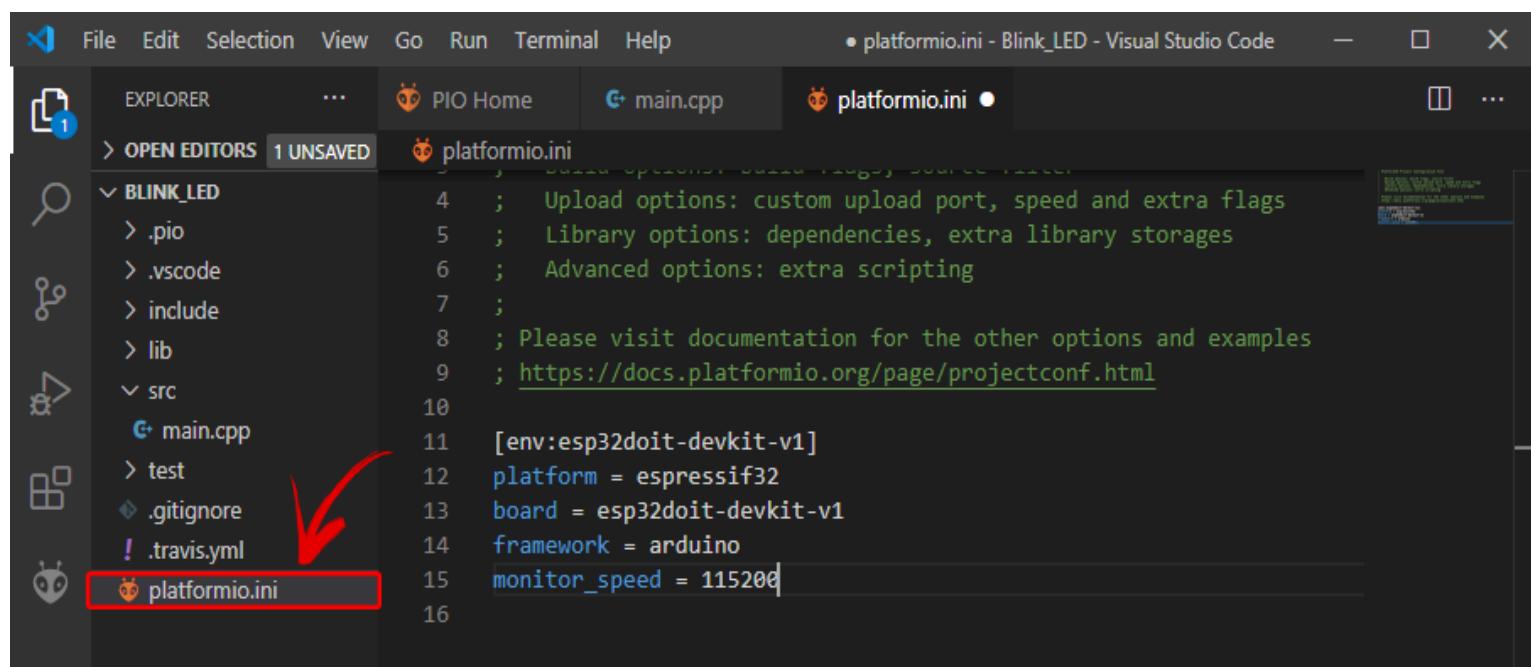


VS Code and PlatformIO have a folder structure that is different from the standard .ino project. If you click on the Explorer tab, you'll see all the files it created under your project folder. It may seem a lot of files to work with. But, don't worry, usually you'll just need to deal with one or two of those files.

Warning: you shouldn't delete, modify or move the folders and the *platformio.ini* file.

platformio.ini file

The *platformio.ini* file is the PlatformIO Configuration File for your project. It shows the platform, board, and framework for your project. You can also add other configurations like libraries to be included, upload options, changing the Serial Monitor baud rate and other configurations.



A screenshot of the Visual Studio Code interface. The title bar says "platformio.ini - Blink_LED - Visual Studio Code". The left sidebar shows a project structure for "BLINK_LED" with files like .pio, .vscode, include, lib, and src/main.cpp. A red arrow points from the text below to the "platformio.ini" file in the list. The main editor area shows the contents of the platformio.ini file:

```
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15 monitor_speed = 115200
16
```

- **platform**: which corresponds to the SoC used by the board.
- **board**: the development board you're using.
- **framework**: the software environment that will run the project code.

With the ESP32 and ESP8266, if you want to use a baud rate of 115200 in your Serial Monitor, you just need to add the following line to your *platformio.ini* file.

```
monitor_speed = 115200
```

After that, make sure you save the changes made to the file by pressing **Ctrl+S**.

In this file, you can also include the identifier of libraries you'll use in your project using the `lib_deps` directive, as we'll see later.

src folder

The `src` folder is your working folder. Under the `src` folder, there's a `main.cpp` file. That's where you write your code. Click on that file. The structure of an Arduino program should open with the `setup()` and `loop()` functions.

The screenshot shows the PlatformIO Home interface within VS Code. The left sidebar displays project files: .pio, .vscode, include, lib, src (which contains main.cpp), test, .gitignore, .travis.yml, and platformio.ini. A red arrow points from the text "Copy the following code to your `main.cpp` file." to the `main.cpp` file in the center editor area. The code in `main.cpp` is:

```
src > main.cpp > ...
1 #include <Arduino.h>
2
3 void setup() {
4     // put your setup code here, to run once:
5 }
6
7 void loop() {
8     // put your main code here, to run repeatedly:
9 }
```

In PlatformIO, all your Arduino sketches should start with the `#include <Arduino.h>`.

Uploading Code using PlatformIO IDE: ESP32/ESP8266

Copy the following code to your `main.cpp` file.

```
*****
Rui Santos
Complete project details at https://RandomNerdTutorials.com/vs-code-platformio-esp32-esp8266/
*****

#include <Arduino.h>

#define LED 2

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(LED, HIGH);
    Serial.println("LED is on");
    delay(1000);
}
```



```
Serial.println("LED is off");
delay(1000);
}
```

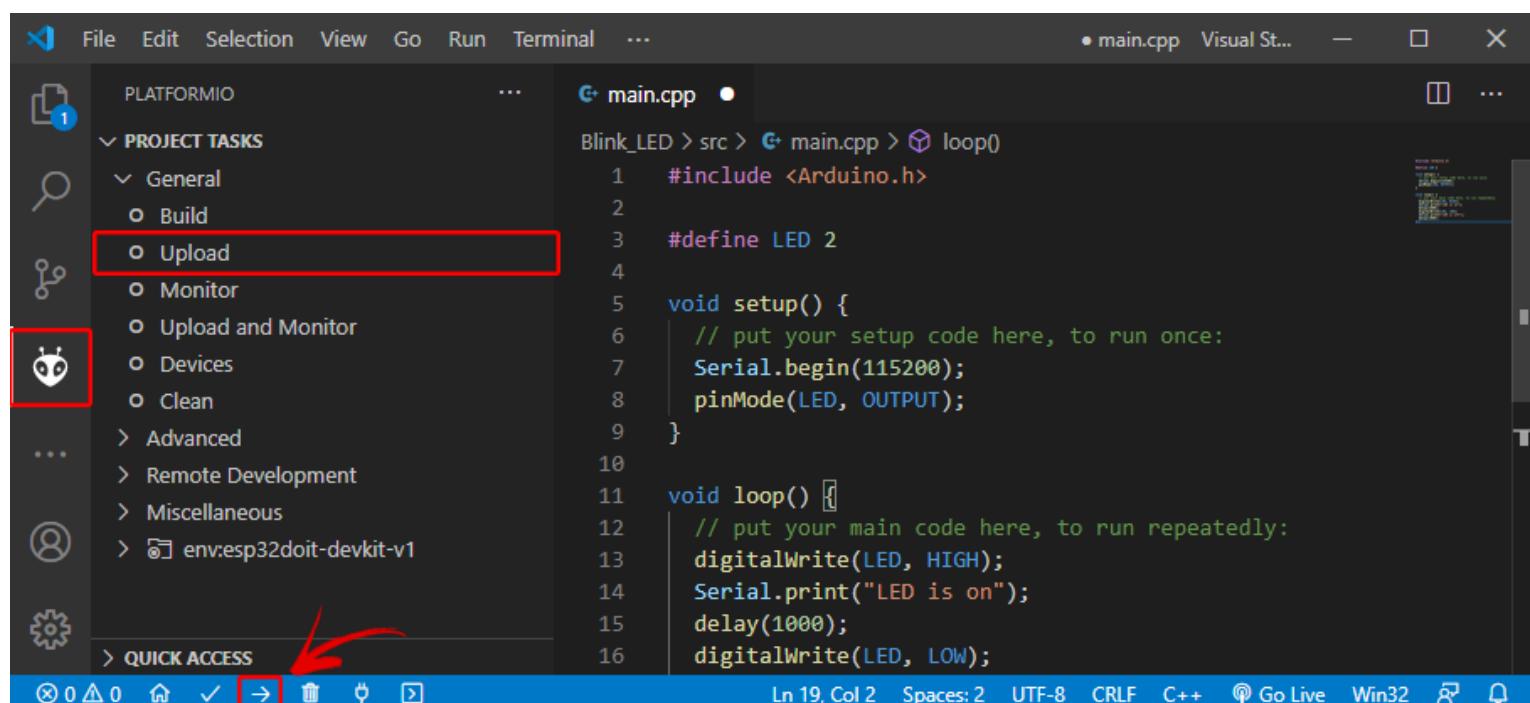
[View raw code](#)

This code blinks the on-board LED every second. It works with the ESP32 and ESP8266 boards (both have the on-board LED connected to GPIO 2).

We recommend that you copy this code manually, so that you see the autocomplete and other interesting features of the IDE in action. Additionally, if you have a syntax error somewhere in your program, it will underline it in red even before compiling.

After that, press **Ctrl+S** or go to **File > Save** to save the file.

Now, you can click on the Upload icon to compile and upload the code. Alternatively, you can go to the PIO Project Tasks menu and select **Upload**.



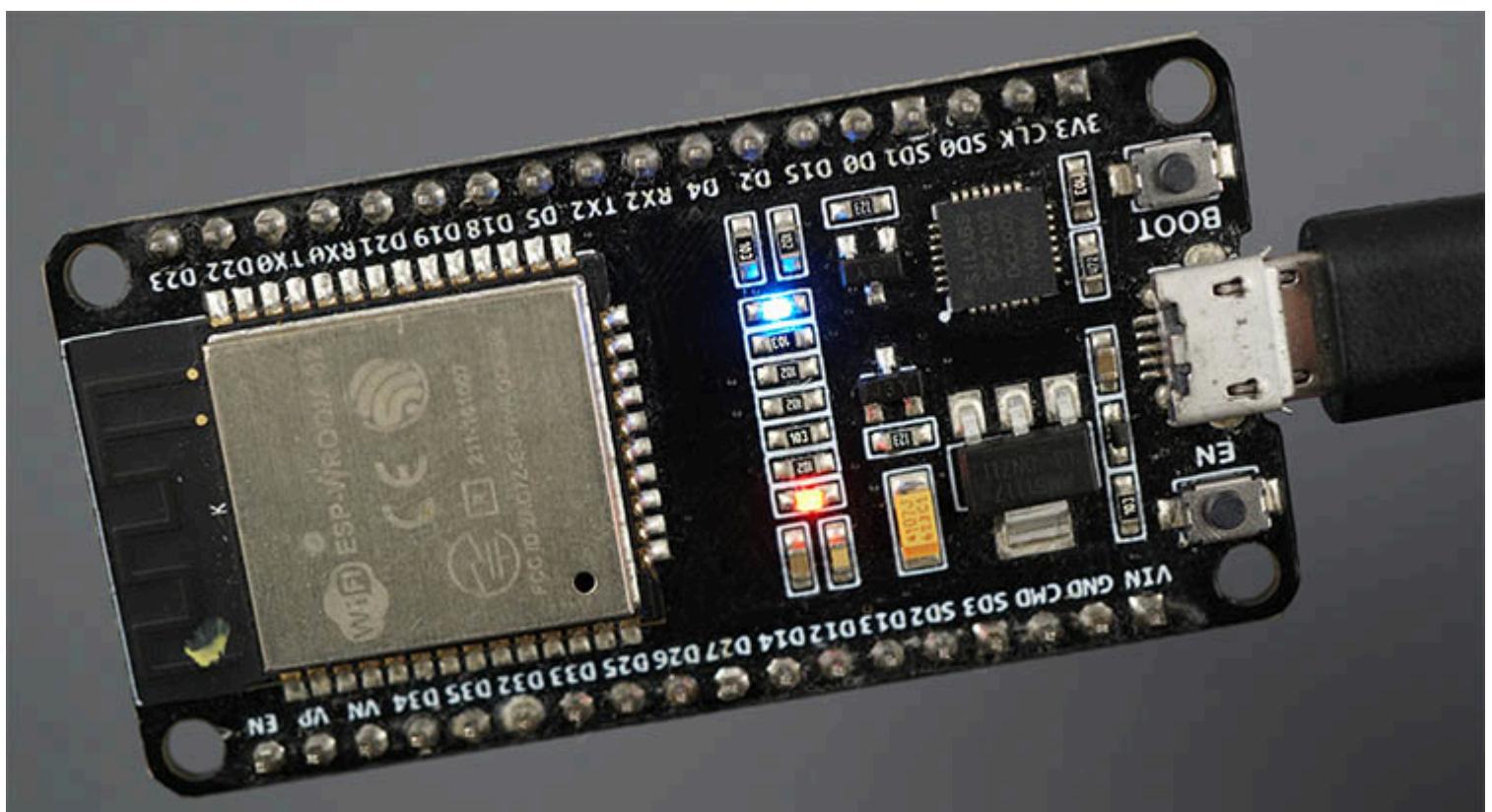
If the code is successfully uploaded, you should get the following message.

The screenshot shows the Visual Studio Code interface with the PlatformIO extension open. The left sidebar displays project tasks under 'PROJECT TASKS' such as General, Build, Upload, Monitor, and Advanced. The main editor area shows a snippet of Arduino code for a Blink LED example. Below the editor is a terminal window with a red border containing the following text:

```
Wrote 216752 bytes (109861 compressed) at 0x00010000 in 2.7 seconds (effective 631.3 kbit/  
Hash of data verified.  
  
Leaving...  
Hard resetting via RTS pin...  
===== [SUCCESS] Took 30.85 seconds =====  
  
Terminal will be reused by tasks, press any key to close it.
```

A large red arrow points from the bottom-left towards the terminal window.

After uploading the code, the ESP32 or ESP8266 should be blinking its on-board LED every second.



Now, click on the Serial Monitor icon and you should see it printing the current LED state.

The screenshot shows the Visual Studio Code interface with the following details:

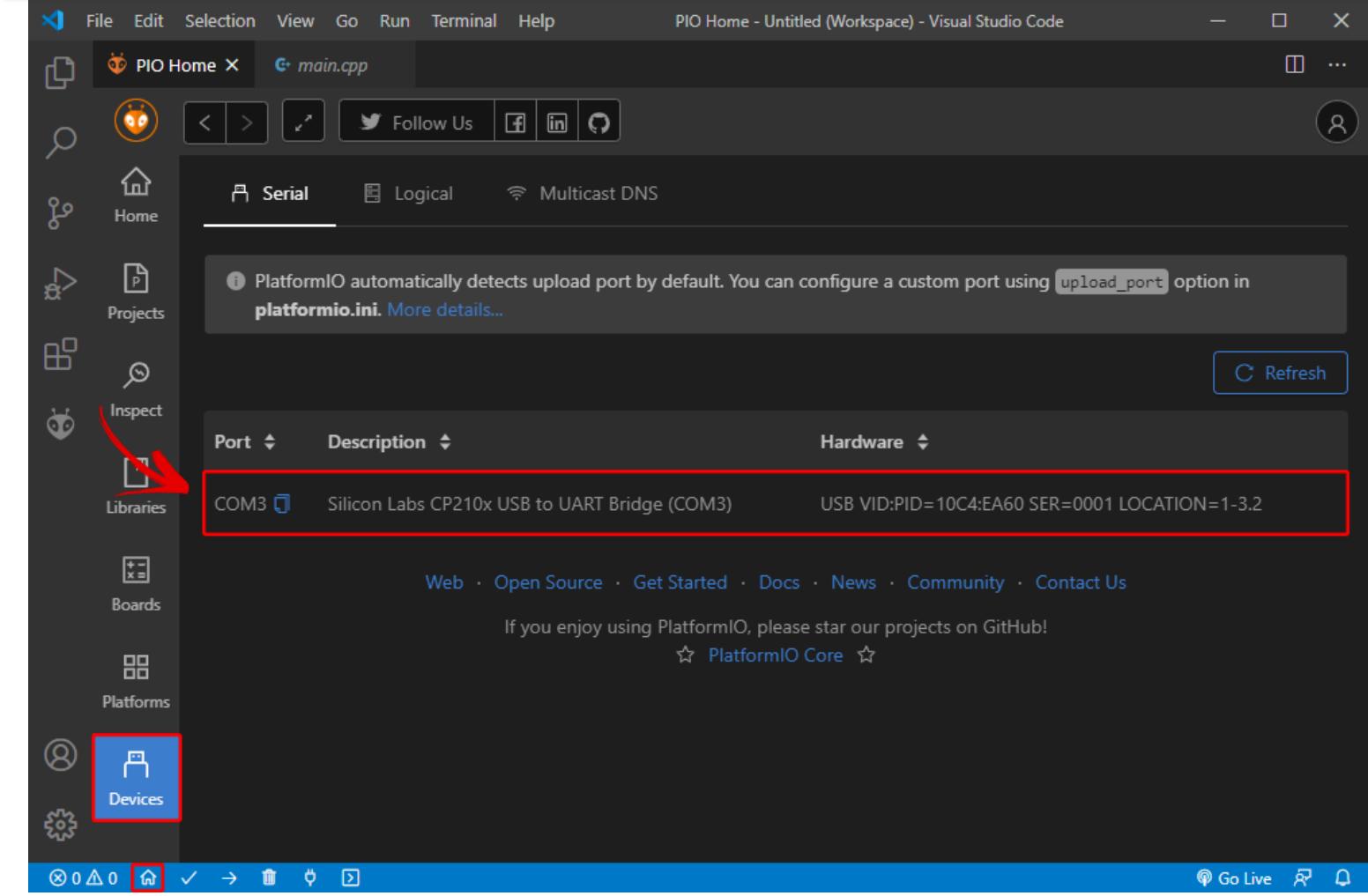
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Project Explorer:** Shows a project named "PLATFORMIO" with a file "main.cpp" and a configuration file "platformio.ini".
- Terminal:** The "TERMINAL" tab is selected, displaying the output of a task named "Task - PlatformIO: Monitor". The output shows the state of an LED:

```
LED is on
LED is off
LED is on
```
- Quick Access:** A red arrow points to the "Serial Monitor" icon in the quick access bar at the bottom left.
- Bottom Status Bar:** ShowsLn 13, Col 27 Spaces: 2 UTF-8 CRLF C++ Go Live Win32

Note: if you don't see the Terminal window, go to the menu Terminal > New Terminal.

Detect COM Port

PlatformIO will automatically detect the port your board is connected to. To check the connected devices you can go to the **PIO Home** and click the **Devices** icon.



Troubleshooting

1) If you try to upload a new sketch to your ESP32 and you get this error message “*A fatal error occurred: Failed to connect to ESP32: Timed out... Connecting...*”. 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 your sketch
- After you see the “*Connecting....*” message in your Arduino IDE, release the finger from the **BOOT** button
- After that, you should see the “*Done uploading*” message

You'll also have to repeat that button sequence every time you want to upload a new sketch. But if you want to solve this issue once for all without the need to press the **BOOT** button, follow the suggestions in the next guide:

- [SOLVED] Failed to connect to ESP32: Timed out waiting for packet header

- [Install USB Drivers – CP210x USB to UART Bridge \(Windows PC\)](#)
- [Install USB Drivers – CP210x USB to UART Bridge \(Mac OS X\)](#)

If you experience any problems or issues with your ESP32, take a look at our in-depth [ESP32 Troubleshooting Guide](#).

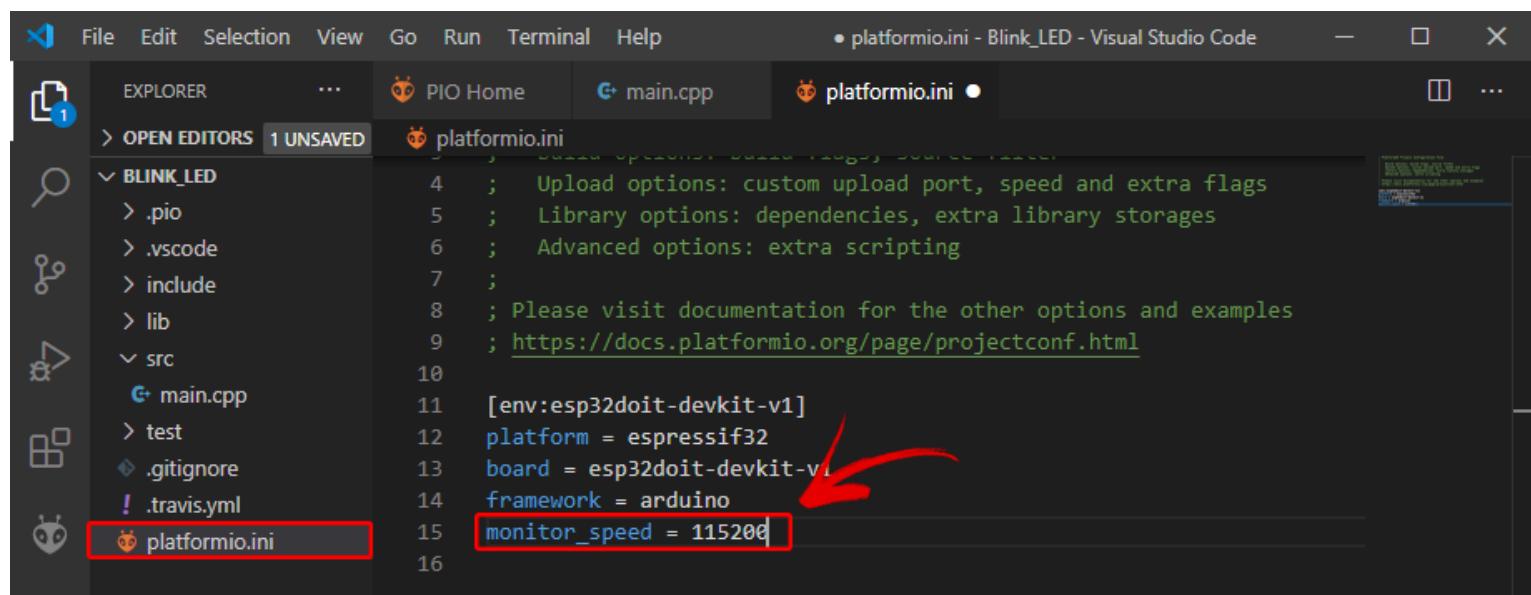
Changing the Serial Monitor Baud Rate – PlatformIO IDE

The default baud rate used by PlatformIO is 9600. However, it is possible to set up a different value as mentioned previously. On the File Explorer, under your project folder, open the *platformio.ini* file and add the following line:

```
monitor_speed = baud_rate
```

For example:

```
monitor_speed = 115200
```



```

File Edit Selection View Go Run Terminal Help
• platformio.ini - Blink_LED - Visual Studio Code
EXPLORER ... PIO Home main.cpp platformio.ini
OPEN EDITORS 1 UNSAVED platformio.ini
> BLINK_LED
> .pio
> .vscode
> include
> lib
> src
> main.cpp
> test
> .gitignore
! .travis.yml
platformio.ini
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
monitor_speed = 115200

```

After that, save that file.

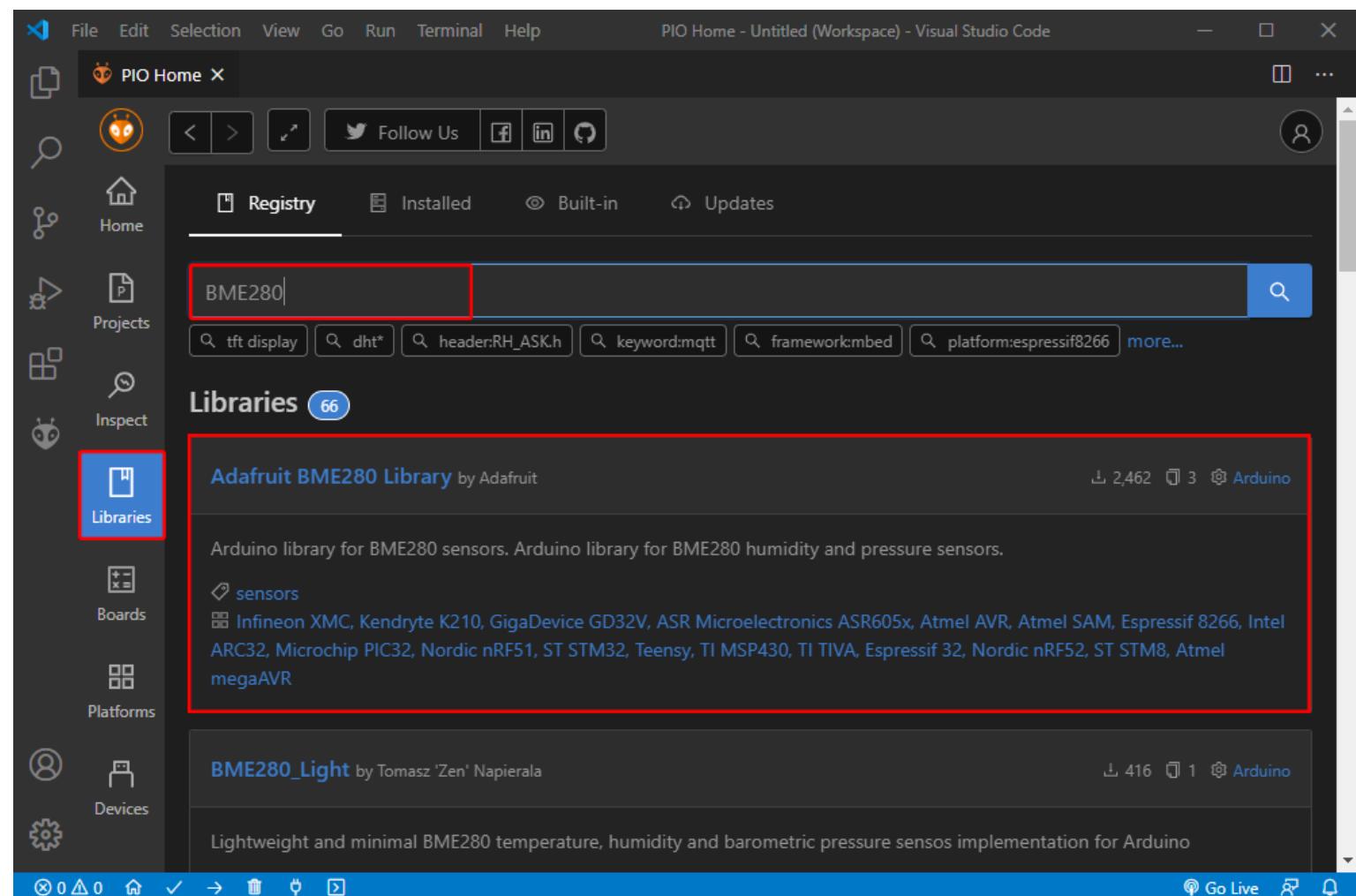
Installing ESP32/ESP8266 Libraries on PlatformIO IDE



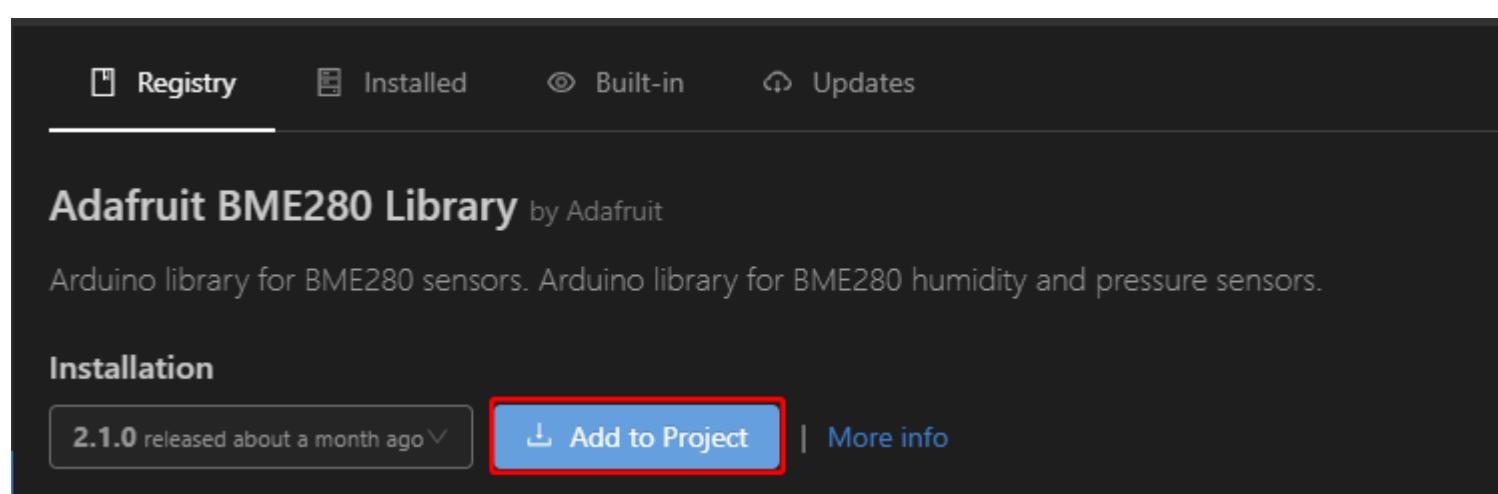
Follow the next procedure if you need to install libraries in PlatformIO IDE.

Click the **Home** icon to go to PlatformIO Home. Click on the **Libraries** icon on the left side bar.

Search for the library you want to install. For example *Adafruit_BME280*.



Click on the library you want to include in your project. Then, click **Add to Project**.



Select the project where you want to use the library.

Add project dependency

X

adafruit/Adafruit_BME280_Library@^2.1.0

Projects\Blink_LED

You can manage your projects in the "Projects" section: create a new or add existing.

Information

- > Registry and Specification
 - > External resources

Cancel

Add

This will add the library identifier using the `lib_deps` directive on the `platformio.ini` file. If you open your project's `platformio.ini` file, it should look as shown in the following image.

```
EXPLORER    ...    main.cpp    PIO Home    platformio.ini X
OPEN EDITORS
WEB ...    Blink_LED
    .pio
    .vscode
    Example
    include
    lib
    src
        main.cpp
        test
        .gitignore
        ! .travis.yml
    platformio.ini

Blink_LED > platformio.ini
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15 monitor_speed = 115200
16 lib_deps = adafruit/Adafruit_BME280_Library@^2.1.0
17
```

Alternatively, on the library window, if you select the **Installation** tab and scroll a bit, you'll see the identifier for the library. You can choose any of those identifiers depending on the options you want to use. The library identifiers are highlighted in red.



Adafruit BME280 Library

by Adafruit
Arduino library for BME280 sensors. Arduino library for BME280 humidity and pressure sensors.

Installation

2.1.0 released about a month ago

[Add to Project](#)

| More info

[Examples](#)

[Installation](#)

[Headers](#)

[Changelog](#)

Library Dependencies

platformio.ini

The PlatformIO Registry is fully compatible with [Semantic Versioning](#) and its "version" scheme `<major>.<minor>.<patch>`. You can declare library dependencies in "platformio.ini" configuration file using `lib_deps` option.

```
; platformio.ini - project configuration file

[env:my_build_env]
platform = infineonxmc
framework = arduino
lib_deps =
    # RECOMMENDED
    # Accept new functionality in a backwards compatible manner and patches
    adafruit/Adafruit_BME280_Library @ ^2.1.0
    # Accept only backwards compatible bug fixes
    # (any version with the same major and minor versions, and an equal or greater patch version)
    adafruit/Adafruit_BME280_Library @ ~2.1.0
    # The exact version
    adafruit/Adafruit_BME280_Library @ 2.1.0
```

Then, go to the `platformio.ini` file of your project and paste the library identifier into that file, like this:

```
lib_deps = adafruit/Adafruit_BME280_Library@^2.1.0
```

If you need multiple libraries, you can separate their name by a coma or put them on different lines. For example:

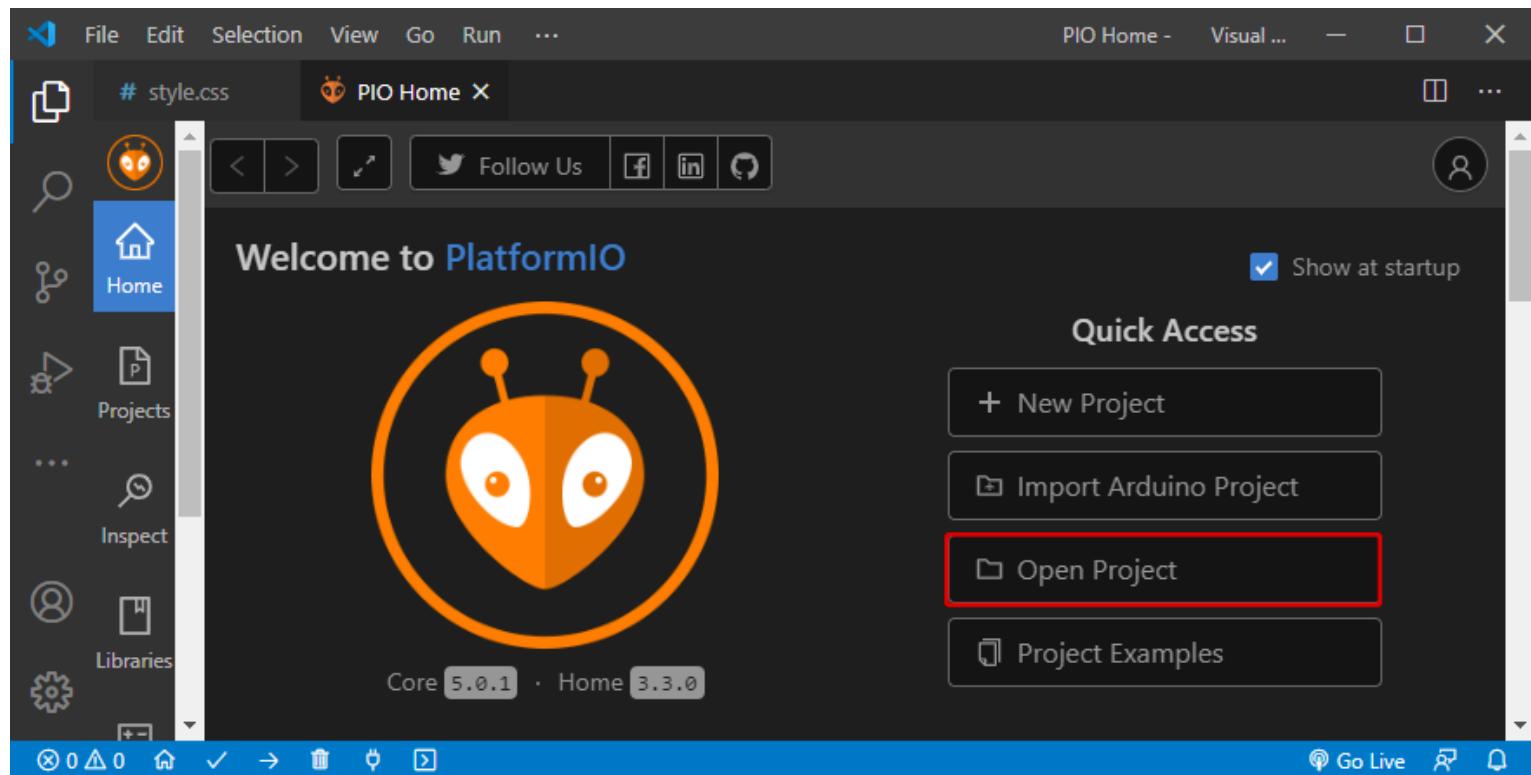
```
lib_deps =
    arduino-libraries/Arduino_JSON @ 0.1.0
    adafruit/Adafruit_BME280_Library @ ^2.1.0
    adafruit/Adafruit Unified Sensor @ ^1.1.4
```

PlatformIO has a built-in powerful Library Manager, that allows you to specify custom dependencies per project in the Project Configuration File `platformio.ini` using `lib_deps`. This



Open a Project Folder

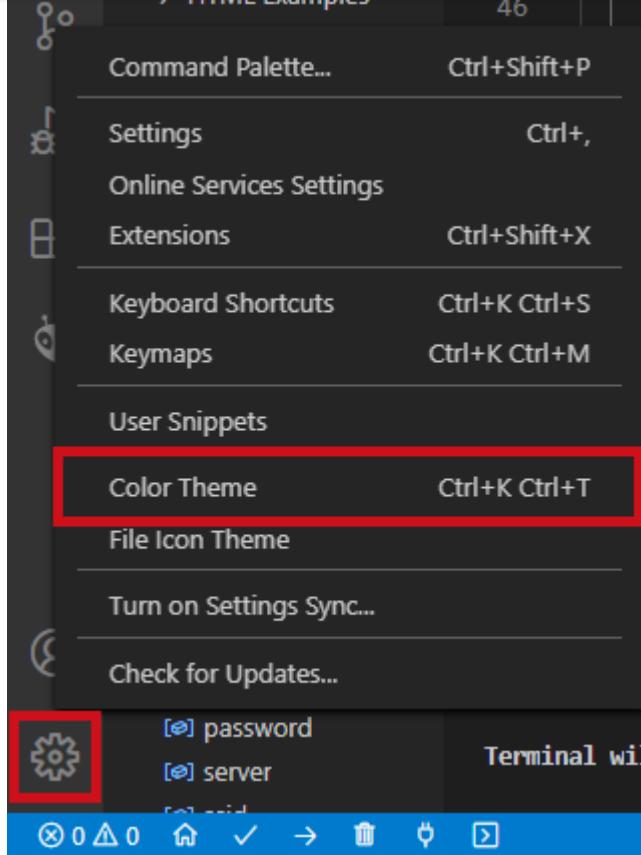
To open an existing project folder on PlatformIO, open VS Code, go to PlatformIO Home and click on **Open Project**. Navigate through the files and select your project folder.



PlatformIO will open all the files within the project folder.

VS Code Color Themes

VS Code lets you choose between different color themes. Go to the **Manage** icon and select **Color Theme**. You can then select from several different light and dark themes.



Shortcuts' List

For a complete list of VS Code shortcuts for Windows, Mac OS X or Linux, check the next link:

- [VS Code Keyboard Shortcuts Reference.](#)

Wrapping Up

In this tutorial you've learned how to install and prepare Visual Studio Code to work with the ESP32 and ESP8266 boards. VS Code with the PlatformIO IDE extension is a great alternative to the classical Arduino IDE, especially when you're working on more advanced sketches for larger applications.

Here's some of the advantages of using VS Code with PlatformIO IDE over Arduino IDE:

- It detects the COM port your board is connected to automatically;
- VS Code IntelliSense: Auto-Complete. IntelliSense code completion tries to guess what you want to write, displaying the different possibilities and provides insight into the parameters that a function may expect;
- Error Highlights: VS Code + PIO underlines errors in your code before compiling;
- Multiple open tabs: you can have several code tabs open at once;

- And much more...

If you're looking for a more advanced IDE to write your applications for the ESP32 and ESP8266 boards, VS Code with the PlatformIO IDE extension is a great option.

We hope you've found this tutorial useful. If you like ESP32 and ESP8266, check the following resources:

- [Learn ESP32 with Arduino IDE \(eBook + course\)](#)
- [Home Automation using ESP8266 \(eBook\)](#)
- [More ESP32 Projects and Tutorials...](#)
- [More ESP8266 Projects and Tutorials...](#)

PCBWay PCB Fabrication & Assembly

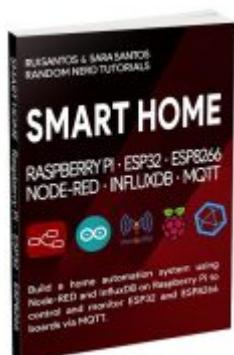
ONLY \$5 for 10 PCBs

✓ 24-hour Build Time ✓ Quality Guaranteed
✓ Most Soldermask Colors:


[Order now](#)



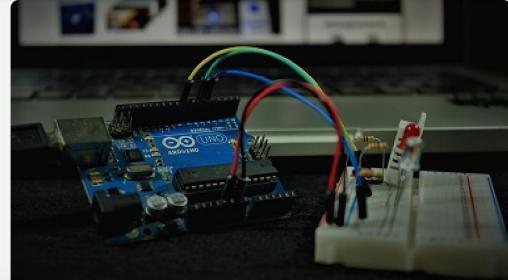
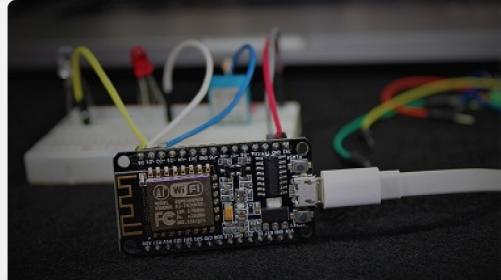
www.pcbway.com



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...

[ESP32 Web Server – Arduino IDE](#)

[ESP8266 NodeMCU Door Status Monitor with Email Notifications \(IFTTT\)](#)

[ESP32: Send Pushover Notifications \(Arduino IDE\)](#)

Enjoyed this project? Stay updated by subscribing our newsletter!

Your Email Address

 SUBSCRIBE

142 thoughts on “Getting Started with VS Code and PlatformIO IDE for ESP32 and ESP8266 (Windows, Mac OS X, Linux Ubuntu)”



Jean Pierre

September 24, 2020 at 2:35 pm

How about asm?

[Reply](#)



Sara Santos

September 25, 2020 at 8:57 am

Hi.

What are you referring to?

[Reply](#)



KlaRa

September 24, 2020 at 3:47 pm

[Reply](#)



Sara Santos

September 25, 2020 at 8:59 am

Hi.

Yes. Learn more here: <https://docs.platformio.org/en/latest/plus/debugging.html>

Regards,

Sara

[Reply](#)



KlaRa

September 25, 2020 at 9:28 am

Great!

[Reply](#)



Kussoy

November 9, 2020 at 1:27 pm

Troubleshooting, error syntax and useless code ! You have an resolve with Alien code ?

[Reply](#)



Sara Santos

November 10, 2020 at 10:37 pm

[Reply](#)



Car Hage

September 25, 2020 at 9:20 pm

Besides the link Sara mentions here are some other references:

A good video from Andreas Spiess demonstrating hardware debug in platformio:
youtube.com/watch?v=psMqilqlrRQ

Another good reference and video:

hackster.io/brian-lough/use-the-platformio-debugger-on-the-esp32-using-an-esp-prog-f633b6

Most guides show only Windows installation. To get the JTAG interface to work in Linux you need to install the 99-platformio-udev.rules so the USB interface recognizes the device as explained in:

docs.platformio.org/en/latest/plus/debug-tools/esp-prog.html

For MacOS, you apparently need to remove the default FTDI drivers (same reference above).

I recommend the ESP-PROG made by Espressif. Besides the usual Chinese suppliers, it is available from US mainstream distributors for \$12-13 without shipping.

[Reply](#)



Sara Santos

September 27, 2020 at 12:57 pm

Hi.

Thank for sharing this 😊

[Reply](#)



Derek Leong

September 24, 2020 at 5:51 pm

Great tutorial! Do you also have a tutorial on how to use esp idf in Platformio?

[Reply](#)



Sara Santos

September 25, 2020 at 9:16 am

Hi.

Unfortunately, we don't have any tutorial about that subject.

Regards,

Sara

[Reply](#)



BelaC

September 24, 2020 at 6:31 pm

Is there any experience, what is the speed difference on program run/code size VS coded vs. C coded? Is it possible to handle interrupts in VS?

[Reply](#)



Stefan Ludwig

September 24, 2020 at 6:59 pm

[View all posts by Stefan Ludwig](#)

IDE that would be worth the hassle of this setup-process?

[Reply](#)



brian

September 24, 2020 at 8:29 pm

The wrapping up statement gives you the main benefits for your efforts, i will add that your programming skills will be taken to a higher level. I wish i had this when i installed it on my pc. This ia a great tutorial to get you started on what could be a career changing journey for some.

[Reply](#)



A Lee

September 24, 2020 at 10:25 pm

Stefan:

I was wondering the same thing! What is the advantage of using VS Code and PlatformIO! With the Arduino platform, I know, at time, there is issue. I have learned to deal with it. I don't have any project that the Arduino platform doesn't work.

[Reply](#)



Sara Santos

September 25, 2020 at 9:03 am

Hi.

It is very useful if you're setting up a web server with separated HTML, CSS and

can open the files side by side. It also allows you to upload the files to the ESP32 or ESP8266 filesystem easily.

Additionally, if you're working on a quite complex project, it can be useful because it highlights errors, helps with autocompletion, indentation, etc.

Regards,

Sara

[Reply](#)



A Lee

September 25, 2020 at 3:50 pm

Thanks Brian and Sara,

I did see the benefit. I will try VS and PlatformIO. Hopefully, it does not break the Arduino IDE.

[Reply](#)



Daniel

September 24, 2020 at 7:47 pm

Thank you for the recommendation. I would love to get it running immediately

[Reply](#)



Sara Santos

September 25, 2020 at 9:04 am

Great!

Try it on! 😊



Belen Hedderich

September 24, 2020 at 9:50 pm

Excelent Guide to start!

Can not wait for a book with some full projects to fully understand it maybe multi platform and some debugging tutorials?

[Reply](#)



Carl Hage

September 24, 2020 at 10:19 pm

Thanks for a great writeup! It was nice to have instructions for Windows, MacOS, and Linux.

I've always thought the code editing and management of VSCode was so much better than the Arduino. I used the old (not VSCode) Platformio before for ESP and STM, but as Stefan mentions, it's a bunch of overhead. As you mention, for larger projects VSCode is way better.

The best reason to use platformio is to get interactive hardware debugging inside VSCode. That is a game-changer. It would be great to add another tutorial to the series on how to use hardware debug on Windows/Mac/Linux. I have an EDP-Prog (<\$13) but I haven't had time to play with it.

[Reply](#)



Carl Hage

September 24, 2020 at 10:24 pm

I mean't ESP-Prog

[Reply](#)



Sara Santos

September 25, 2020 at 9:11 am

Thanks for your comment.

We are definitely planning to add some more tutorials about this subject.

Regards,

Sara

[Reply](#)



Eduardo Alvim

September 25, 2020 at 1:51 am

Thank you for introducing here this new tool!

It showed very useful here when I used it to create a project using ESP32 FreeRTOS (ESP-IDF).

I already used the VS Code when I was studying FreeRTOS here, but without using the PlatformIO resources. Now it's much better to use it!

[Reply](#)



GIANNI

September 25, 2020 at 9:25 am

much for your patience and precision.

I've a simple question : can I copy all Arduino ESP32 projects now running with Arduino compilation in PlatformIO without any changes (I mean instruction and declaration define etc.) and uploading it with PlatformIO on ESP32 board ? Or it's a long way of conversion and test ?

Regards and thanks again.

[Reply](#)



Sara Santos

September 27, 2020 at 1:12 pm

Hi.

You can easily do that.

In PlatformIO Home, select the Option “Import Arduino Project”.

Tick the option “Use libraries installed by Arduino IDE”.

And that's it.

Regards,

Sara

[Reply](#)



gianni nasazzi

September 27, 2020 at 3:02 pm

Thanks a lot. Very kind of you to answer to my question.

[Reply](#)



Sara Santos

September 28, 2020 at 4:38 pm

You're welcome.

[Reply](#)



Domenico

September 25, 2020 at 10:04 am

Hi Sara, it seems to me very close to ESPHome platform used for Home Assistance. Am I wrong? Thanks again.

[Reply](#)



Zvika

September 25, 2020 at 1:34 pm

Hi,

Is it required to press the boot switch of the esp32 while downloading?

I am asking so as I got:

“*** [upload] Error 1

===== [FAILED] Took 3.74 seconds

=====

The terminal process “platformio ‘run’, ‘–target’, ‘upload’” terminated with exit code: 1.”

Or I am missing something?

Zvika

[Reply](#)



Sara Santos

September 27, 2020 at 1:01 pm

Hi.

Yes, you may need to press the boot button while uploading.

Regards,

Sara

[Reply](#)



Jairo

September 25, 2020 at 1:46 pm

I was really waiting for this tutorial! THANKS! hope to see more about programming with VS Code.

[Reply](#)



Sara Santos

September 27, 2020 at 1:01 pm

Great!

I'm glad you liked it 😊

Regards,

Sara

[Reply](#)



Jordan

September 25, 2020 at 9:05 pm

I am mostly a hobbyist / tinkerer. I've played around with VSCode / Platformio a bit,

My take on it is that VSCode / Platformio is extremely convenient for those who are writing complex code that involves (for example) writing new libraries alongside the actual Arduino or ESP code, or working with files of different kinds all next to each other (as in the article summary). And the syntax error underlining and completion tools are helpful.

But I suspect that VSCode / Platformio is overkill for those who are comfortable with the Arduino IDE, are writing simple code for Arduino or ESP, and are largely following “recipes” online using pre-baked libraries.

VSCode / Platformio has a learning curve in terms of figuring out how folders are organized, where your code is, what the windows do, how to properly update libraries and board configuration files, etc. Hobbyists may find it’s not worth climbing that learning curve.

A short take on it would be that VSCode / Platformio is more “programmer” focused while the Arduino IDE is more “hobbyist” focused.

I myself bounce back and forth between the two — quite often switch back to the Arduino IDE when I find myself frustrated with a basic problem (often board compatibility) that I don’t know how to fix in VSCode / Platformio.

[Reply](#)



Sara Santos

September 27, 2020 at 12:59 pm

Hi Jordan.

That's very true.

It may be overwhelming for beginners. For simpler and smaller programs Arduino IDE works just fine.

Thanks for sharing your experience.

Regards,

Sara

[Reply](#)



jovosm

September 26, 2020 at 5:07 am

Have tried one of Your projects from book ordered from You. Works good in Windows, but

in Linux i have problem not directly pertaining Your lesson but with Arduino IDE has problem with serial. I'm not talking about connecting board to computer's USB, that works fine. It's rather whether You call for "Serial.print" in code or not, when compiling it gives error: No module named 'serial'. I understand that ESP32 needs python to work with IDE

(or that I'm just assuming, correct me if I'm wrong). ESP8266 works fine. I have Python 3 installed ans also installed pyserial(with pip3 comm.) But no matter I tried it gives error again.

[Reply](#)



Sara Santos

September 27, 2020 at 12:54 pm

Hi.

If you're one of our students, I suggest that you place your question on the RNTLAB forum: <https://rntlab.com/forum/>

That way, we can better follow your issue and ask you questions to better understand what's going on.

Regards,
Sara

[Reply](#)



KSanger

April 30, 2021 at 2:44 am

This is probably not a correct answer but you typed “Serial.print” works and “serial” does not. Might the problem be that we need to use a Capital S in Serial? I’ve made that mistake many times in Java and assume Python and Arduino are case sensitive too.

[Reply](#)



Bob

September 26, 2020 at 9:57 am

Thank you for this informative tutorial!

Another HUGE benefit from using PIO is the portability!

When using Arduino IDE, copying a project to another computer involves adding new boards or libraries using the board and library manager (often quite a hassle to satisfy all dependencies).

By using the lib_deps statement in the platformio.ini PIO just downloads missing libraries!

[Reply](#)



Sara Santos

September 27, 2020 at 12:51 pm

That’s true!

That’s another advantage of VS Code with PIO.

Regards,

Sara

[Reply](#)



Hi Sara,

Thanks for providing details and steps for that.

I am ESP8266 and ESP32 developer and I have total 2 to 2.5 years experience for same. Till now we are using ESP32 IDF behind our own SDK and developed almost 10 to 12 products.

I just want to know that VS will support for ESP32 IDF or just for Arduino?

[Reply](#)



Sara Santos

September 27, 2020 at 12:49 pm

Hi.

It also supports IDF.

Regards,

Sara

[Reply](#)



Reeta

June 16, 2021 at 3:56 am

I am newbie for ESP32 and keen to know in detail how to interface it with AM2301 on vscode with platformio to monitor the values both on serial and web, (without arduino)

[Reply](#)



marcel leemans

September 28, 2020 at 2:44 pm

Hi Sara

Thanks for this tutorial !!
after som mistakes it works !!

[Reply](#)



Sara Santos

September 28, 2020 at 4:26 pm

Great!

[Reply](#)



Garry Kraemer

October 1, 2020 at 5:30 pm

I have just started using VS Studio with PlatformIO. How do I “update” external libraries specified in “lib_deps” to the latest version?

In the Arduino IDE, I just click on “Manage Libs” and it updates all my specified external libraries.

[Reply](#)



Sara Santos

October 2, 2020 at 8:22 am

Hi.

If you want to use the latest version of the library, you can search for the library in the Libraries tab, select the installation tab and there should be several options for the lib_deps. Usually there's something indicating the library version. So, you just need to copy the name of the latest version.

Regards,

Sara

[Reply](#)



Ronald Lopez

October 2, 2020 at 9:46 pm

I already followed the example in Windows 10 without any problem. Only have a question: Why is required to load python?

REGards

[Reply](#)



ksanger

April 30, 2021 at 2:47 am

I believe that vscode is a Python program. So to run vscode you must have Python loaded onto the computer being used. Similar to eclipse being a Java program that requires Java to run.

[Reply](#)



JPD

October 2, 2020 at 6:28 pm

Can we do arduino assembly on VScode?

Reply



Donald Garber

October 16, 2020 at 9:47 pm

“Follow the next procedure if you need to install libraries in PlatformIO IDE.”

Do you need to do this if you already have a lot of installed libraries under the Arduino IDE?

Reply



Donald Garber

October 16, 2020 at 10:55 pm

Okay, yes. The libraries I've accumulated are within the Arduino file structure, not the PIO structure.

I don't think I want to import many of my Arduino projects into PIO. Converting .ino files and backtracking its libraries is a little bit of a pain. Going forward, on new projects, I will start using PIO instead of the arduino IDE, and I'll give it a shot. But, the LED_blink example is a big help, and you guys have made implementing PIO pretty easy.

Reply



Donald Garber

October 21 2020 at 10:31 pm



lines of formatted text into a project's platform.ini file) is a better board management practice, as are attaching libraries that may change over time.

You can import older sketches into PIO, you just have to make a few changes, like adding arduino include and changing the .ino file to .cpp.

[Reply](#)



Sara Santos

October 22, 2020 at 10:29 am

Hi Donald.

That's great!

I'm glad you liked using PlatformIO. It's way easier to write code and manage libraries.

Once you get used to it, you won't want to go back 😊

Regards,

Sara

[Reply](#)



Joel

October 29, 2020 at 8:45 pm

I received this error after I installed the BME280 library according to the tutorial, then uploaded:

"In file included from .pio\libdeps\esp32dev\Adafruit BME280
Library\Adafruit_BME280.cpp:31:0:
.pio\libdeps\esp32dev\Adafruit BME280 Library\Adafruit_BME280.h:27:17: fatal error:
SPI.h: No such file or directory"

What is missing?

[Reply](#)



Donald Garber

October 30, 2020 at 2:58 am

Do you have #include <Arduino.h> at the very top of the sketch?

[Reply](#)



Joel

October 30, 2020 at 3:29 pm

Yes, it is the first line of the main.cpp sketch.

[Reply](#)



Carey M Capaldi

November 4, 2020 at 7:12 pm

I have the same issue with that library.

[Reply](#)



Sara Santos

November 5, 2020 at 11:04 am

Hi.

If you're not using the libraries in your sketch, just remove them from the platformio.ini file.

Regards,

Sara





John

December 17, 2020 at 8:59 pm

After purchasing the “building Webservers” book, I installed the platform io extension into VS CODE which I’d had installed for a while. It seemed to work, but I didn’t have time to play with it. Today I opened vscode, received and update, and the PlatformIO home page doesn’t appear nor do the buttons on the status bar. I can’t click on “New Project” because it isn’t there, and no “Home” button on the status bar. I’ve uninstalled PIO, and reinstalled, I uninstalled VScode and reinstalled, nothing seems to work. I’d like to complete this course, but I’m dead in the water right now.

Any suggestions??

[Reply](#)



Eduardo Alvim

December 17, 2020 at 9:15 pm

Hi John, I don’t know exactly what happened there, but did you check whether the PlatformIO plugin is enabled?

Go to the Extensions menu and check if the platformio extension is enabled. Sometimes it is installed, but is disabled.

Well, I don’t have any better clues for this.

[Reply](#)



Carl Hage

December 18, 2020 at 12:05 am

There is an activateOnlyOnPlatformIOProject setting that disables PlatformIO on non PlatformIO projects. I don't know if that is your problem, but I use the feature so all the platformio stuff doesn't get included in my python, etc. projects. It was off when I installed originally, but maybe the default changed.

You can check your settings (gear in lower right), Extensions->Platformio to see if the activateOnlyOnPlatformIOProject is set.

<https://dokk.org/documentation/platformio/v3.6.1/ide/vscode/#platformio-ide-activateonlyonplatformioproject>

I don't remember now how to create a new project if the UI isn't activated, but maybe it needs an empty (or blank line) platformio.ini. If you have a platformio.ini, in your workspace directory, then the PIO extension should activate.

[Reply](#)



John

December 18, 2020 at 12:13 am

Thank you, yes I've checked all the settings, even turn that one on and off to no avail. But thanks for the reply.

[Reply](#)



Dave

December 19, 2020 at 3:02 pm

Hello,

Do you have any keys or suggestions that help with importing Arduino files to Platformio? I have tried and there are some things that are asked by Platformio upon

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

December 19, 2020 at 11:55 pm

Hi.

What things are asked?

Can you be a little more specific?

Regards,

Sara

[Reply](#)**Dave**

December 20, 2020 at 3:14 pm

Hello Sara,

Thank you for your time with an enterprise such as yours I imagine that you are very busy spreading your knowledge of physical computing and networking science.

My issue was with importing Arduino IDE created files into Platformio. I have many many Arduino files and libraries and I was concerned with loosing or changing them to conform to Platformio and not being able to recover them.

Libraries are quite differently handled in Platformio, but since my first question to you I have solved that riddle.

I think that I answered my own questions. For now. I am gratefull to you for getting me started with Platformio.

Thanks

[Reply](#)



Sara Santos

December 21, 2020 at 10:18 am

Great!

I'll try to create more VS Code tutorials, so that it is easier to handle your projects.

Regards,

Sara

[Reply](#)



Carles

December 23, 2020 at 4:43 pm

Hello Sara,

I am manually entering the post “Build an ESP8266 Web Server – Code and Schematics (NodeMCU)” in VS Code and PlatformIO, but I have a problem.

I enter:

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

and at the moment of saving it it becomes:

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

Is there any way to solve it? If not, you will have to do everything with the Arduino IDE

Thank you for your time,

[Reply](#)



Sara Santos

December 24, 2020 at 10:55 am

Hi Carles.

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

Arduino IDE sketches do work on PlatformIO.

Regards,

Sara

[Reply](#)



Carles

December 24, 2020 at 1:46 pm

Hi Sara,

I write “const char“, a space and “ssid”. When I save it it becomes “const char”, a space and “ssid”. The asterisk has passed from the end of the word “char” to the beginning of “ssid”. This is the problem.

Regards,

Carles

[Reply](#)



Carles

December 24, 2020 at 2:00 pm

Sara,

I have taken out the quotes because the asterisks were lost when I answered. In the initial post you can check it.

I write `const char*`, a space and `ssid`. When I save it it becomes `const char`, a space and `*ssid`. The asterisk has passed from the end of the word `char` to the beginning of `ssid`.

Regards,
Carles

[Reply](#)



Carl Hage

December 24, 2020 at 9:46 pm

The VSCode editor has a formatting feature to reformat code to a consistent notation, “prettier”, and seems that it’s enabled. Google “How to disable prettier in VSCode”. Click the gear icon in the lower left corner and you’ll get the settings then look for prett and format. (OR just look at text editor settings.) You can also disable the extension.

There is a setting to format on save, which seems to be what is happening to you.

It could be a different extension than prettier, so you might need to check which extensions are enabled.

[Reply](#)



Carles

December 26, 2020 at 9:49 am

Hi Carl,

I have uninstalled all the extensions except C/C++ and PlatformIO.
Continue changing the asterisk.

Thanks

[Reply](#)



Bart Thompson

December 27, 2020 at 4:29 am

Thanks for the clear write-up.

I noticed many asking why VSCode over arduino....

As an old grey-beard I feel compelled to point out that one of the true great features of VSCode is the git integration. Having built-in source control is truely a blessing. being aware of and using source control is one of the things that separates professional developers from hobby coders.

For those unfamiliar with git or source control, git is the premier means of managing source code as it evolves; and it can save your bacon and it removes the fear of breaking your program when you want to experiment or are just learning a new platform. It truely is magic. I urge everyone to learn it, use it, live it. It will make your development life much easier. The third icon down (the branch one) is the gateway.
Happy Trails..

[Reply](#)



Joschen

January 22, 2021 at 9:12 am

followed your instruction step-by-step (thank you) I got the message concerning
`#include <Arduino.h>`
“The include file not found in browse.path”
any idea what to change, what to add?

Thank you!

[Reply](#)



WILLPOWER

February 18, 2021 at 7:24 pm

hi, my esp32 board was working fine. I was working in arduino. I followed your instructions for PlatformIO and when I uploaded my board got a fatal error.

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header
*** [upload] Error 2

What can I do to fix this error and upload to my board?

I'm on Ubuntu 20.04.2 LTS.

Thank you

[Reply](#)



Sara Santos

February 19, 2021 at 10:59 am

Hi.

Press the ESP32 on-board BOOT button when you start seeing a lot of dots on the debugging window.

Regards,

Sara

[Reply](#)



WILLPOWER

February 19, 2021 at 7:49 pm

Hi Sara,

I also used the capacitor technique I saw on the site's instructions, but it did not work either...

When I did a search about this error, I found this: MD5 of file does not match data in flash!

What do you think is happening and how do you think it can be fixed ?

Thank you !

Will

[Reply](#)



Sara Santos

February 20, 2021 at 4:36 pm

Hi.

I'm not sure what is happening.

Do you have any peripherals connected to your board?

If you have, you should disconnect them before uploading the code.

Regards,

Sara

[Reply](#)



Sergio Trujillo

February 18, 2021 at 8:41 pm

I am working with VSCode although I miss how to configure the partitions like the Arduino. I am using an ESP32 and would like to choose Minimal SPIFFS (1.9MB APP with OTA / 190KB SPIFFS) for OTA updates. Thanks, great job greetings.

[Reply](#)



Sara Santos

Hi.

I think this is what you must be looking for:

<https://docs.platformio.org/en/latest/platforms/espressif32.html#partition-tables>

Regards,

Sara

[Reply](#)



Sergio Trujillo

February 19, 2021 at 4:30 pm

That's it!! well thank you very much. All the best.

[Reply](#)



Chris

March 6, 2021 at 5:45 pm

Hi! Thanks for the tutorial. I was able to follow this and get the LED flashing, but I'm a bit confused as to why Python was needed, seeing as the code is written in C++. I found this article because I was hoping to program an ESP32 with MicroPython via VSCode. Is there a way to do this?

[Reply](#)



Sara Santos

March 7, 2021 at 12:00 pm

Hi.

Yes, there is an extension to write MicroPython code in VS Code.

However, there isn't an "interface" with buttons to upload the code and so on.

Regards,
Sara

[Reply](#)



Chris Astles

March 7, 2021 at 12:04 pm

Thanks for your reply 😊 So why is Python necessary for this tutorial?

By the way, I found RT-Thread which is a MicroPython extension for VSCode, works really well and has full UI

[Reply](#)



Sara Santos

March 7, 2021 at 4:27 pm

Hi.

Here's the reason: <https://docs.platformio.org/en/latest/faq.html?highlight=python#install-python-interpreter>

I didn't know about RT-Thread. I have to take a look.

Regards,
Sara

[Reply](#)



abdellatifrahali

April 19, 2021 at 8:41 pm

Hi Sara:

can you tell me about this error:

'Accès refusé.', None, 5)_____

Thak you

[Reply](#)



Sara Santos
April 19, 2021 at 9:12 pm

Hi.

That means that VS Code can't establish a serial connection with your board.

That may be because you have the board connected on another program like Arduino IDE. Close Arduino IDE.

It may also be the case that VS Code can't find the right COM port.

Check that it is connecting to the right port. If it isn't, you can manually set a COM port on the platformio.ini file. Add the following to the platformio.ini file:
`upload_port = COMX`

Replace X with the number of the COM port.

I hope this helps.

Regards,

Sara

[Reply](#)



Tabosan
May 2, 2021 at 4:04 pm

Ciao

come posso modificare la libreria dei display Nextion per usarla con PIO?
vedi tutorial rnd per usare nextion e esp32

[Reply](#)



Jesper Schumacher

July 15, 2021 at 5:52 am

Hi

Is ota programming supported in PlatformIO?

BR Jesper

[Reply](#)



Sara Santos

July 16, 2021 at 9:41 am

Yes.

[Reply](#)



michael andersen

July 30, 2021 at 9:25 am

Hi

I use a TTGO T-Journal board, but I can not find this board in the list of PIO boards.

Do you know what board I should use?

[Reply](#)



Sara Santos

July 30, 2021 at 9:58 am

Hi



We have a guide for that board: <https://randomnerdtutorials.com/ttgo-t-journal-esp32-camera-getting-started/>

Regards,

Sara

[Reply](#)



michael

July 30, 2021 at 10:53 am

Hi Sara

Yes I know I can select this module with the Arduino IDE, but in VS code with platformio plugin I can not select this board.
It is not in the boards list.

Any other suggestions?

[Reply](#)



Sara Santos

July 30, 2021 at 11:22 am

Hi again.

Try the Espressif ESP32 Dev Module.
Tell me if it works.

Regards,

Sara

[Reply](#)



Jorge Amodio

October 14, 2021 at 8:01 am



Hi There,

How to build a project for a board using the Arduino framework that has the ESP32-S2 like the Saola-1 from Espressif?

TIA

Jorge

[Reply](#)



Sara Santos

October 14, 2021 at 5:01 pm

Hi.

Take a look at this discussion: community.platformio.org/t/esp32-s2-esp32s2-saola-support/13467/8

Regards,

Sara

[Reply](#)



Eric Carter

October 24, 2021 at 2:05 pm

Hi

Interesting exercise this VScode, which I was introduced to in your Build Web Servers book, but rather a steep learning curve. One thing that puzzles me is that running an esp8266 program on the Arduino IDE I get the following results on memory usage:
Sketch uses 441937 bytes (46%) of program storage space. Maximum is 958448 bytes.
Global variables use 31384 bytes (38%) of dynamic memory, leaving 50536 bytes for local variables. Maximum is 81920 bytes.

Yet when I run it on VScode I get:

RAM: [=====] 52.4% (used 42956 bytes from 81920 bytes)

I not too sure if the two measurements are referring to the same thing although Global Variables seems to correspond to RAM at 81920 bytes

[Reply](#)



Sara Santos

October 24, 2021 at 5:11 pm

Hi.

To be honest, I'm not sure why that happens.

Maybe they are using different versions of the ESP8266 boards' installation??

Regards,

Sara

[Reply](#)



Eric Carter

October 24, 2021 at 6:16 pm

Could be Sara.

Further test are that the compiled bin file is 446kB for the Arduino IDE and 649kB for the VS code. The source file is identical in each case and will OTA update from the Arduino IDE version but either fails to complete or completes with HTTP error in the VScode case.

I will try a few board variations but in each test it was for a sonoff S20
Will let you know if I nail it – otherwise I am sticking with the Arduino IDE

[Reply](#)



Bernard

November 8, 2021 at 6:07 pm

I really wanted to go Linux but no matter which distro I tried, I don't seem to get it right with VS CODE. Every time there is an error concerning the pins_arduino.h . I have even given up on using Linux though I could quickly install if someone showed me how to properly install VS CODE. I have followed this tutorial to the letter but still nothing doing.

[Reply](#)



William Burris

November 27, 2021 at 4:19 am

I've tried everything I can think of, but PlatformIO is not able to get an input com port.

When it failed to discover automatically I tried –

upload_port = /dev/ttyUSB*

upload_port = Com[1]

upload_port = Com[3]

All failed, I am using the “esp32doit-devkit-v1”, Amazon description is “DOIT DEVIT V1 ESP32-WROOM-32 Development Board ESP32 WiFi+Bluetooth Dev Module CP2102 for Arduino”

I'm using a Windows 10 operation system.

When I plug the board in the LED lights up solid. I do not see anything in the drive letters, or devices that indicate an ESP32 is connected. I don't know if I should, this is all new to me.

Any help would be greatly appreciated.

Thanks.

[Reply](#)



Sara Santos

Hi.

Were you able to program your ESP board with Arduino IDE before?

Maybe you don't have the ESP32 USB drivers installed?

Most ESP32 use the CP2102 USB drivers. See bullet number 6 of our troubleshooting guide: <https://randomnerdtutorials.com/esp32-troubleshooting-guide/>

Regards,

Sara

[Reply](#)



William Burris

November 27, 2021 at 3:50 pm

Thanks Sara, I'm not sure if it was missing drivers or a bad cable. I manually installed the drivers first, but still was having problems but wiggled cable and the ESP32 powered up. I uploaded Blink_LED again, SUCESS!! Throwing away cable now!!

[Reply](#)



Sara Santos

November 27, 2021 at 5:19 pm

Great!

I'm glad you found the issue.

Regards,

Sara

[Reply](#)



Victor Das Neves

December 27, 2021 at 3:07 pm

Hi,

Having purchased your Build Web Servers with ESP32 & ESP8266, and had great fun and learnt a great deal from the tutorials.

I have started a new project to use the ESP8266 based on the knowledge gained from the tutorials.

I would like to add a function that gets called in the main.cpp but I do not want the function to be within the main.cpp as the program is growing fast and I would like to have this called function as a separate file.

The problem I have is how to include a function so that the main.cpp calls this function, and how to be able to declare variables in this function that it will be declared in the main.cpp (so the function will use the data in the variables and then save the modified data into another variable name but to be used by the main.cpp or later by another function), and in which directory do I save the function or functions too?

Thankiny you in advance

Victor

[Reply](#)



Jens

January 10, 2022 at 1:27 pm

Hi!

When I remove all serial/print statements the LED is not blinking at all on my ESP-WROOM-32.

Have you ever tried if the onboard LED is really pinned to GPIO2? From the schematics I do not see an onboard LED either, maybe I got an old /cheap one from Aliexpress even it is black?

[Reply](#)



Thorsten

January 25, 2022 at 11:49 am

How to deal with variables(globals too) and function calls over different .h or .cpp files? I want to use my code splitted up like the tabs functionality allows it in the Arduino IDE. Could you show me how to do it?

Thx & Best, thorsten

[Reply](#)



Sara Santos

January 25, 2022 at 3:37 pm

Hi.

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

Regards,

Sara

[Reply](#)



Eduardo Alvim

January 26, 2022 at 7:37 pm

Hi Thorsten, usually, when you declare a global variable in an .h/.hpp file it will be available in your main script. You just need to create your .h/.hpp file and save it in the same folder where your main.cpp file is located. Then, you need to include this .h/.hpp file in your main script with `#include "your_file.h"`. It's the same thing that you do with Arduino.

Is that it your question? Correct me if it's wrong.





Jose Luis

February 4, 2022 at 6:57 pm

Hi, I have a problem to build any program in Visual Studio (I have the same error in Arduino IDE): I only use the Build option (it's supposed it doesn't matter if you have the board connected; anyway, I've tested it with and without several ESP32 connected) and I get several warnings: "No se esperaba & en este momento". Finally I get the error "El proceso del terminal "C:\Users\josel.platformio\penv\Scripts\platformio.exe 'run'" finalizó con el código de salida 1".

Do you have any idea to solve this?

[Reply](#)



StefanL38

February 6, 2022 at 10:40 pm

If I try to finish the project-setup I get this error

Could not initialize project

PIO Core Call Error: "The current working directory

F:\myData\PlatformIO\Projects\MyFirstProjekt will be used for the project.\r\n\r\nThe next files/directories have been created in

F:\myData\PlatformIO\Projects\MyFirstProjekt\r\ninclude – Put project header files here\r\nlib – Put here project specific (private) libraries\r\nsrc – Put project source files here\r\nplatformio.ini – Project Configuration File\r\nPlatform Manager: Installing espressif32\r\n\r\nError: HTTPSConnectionPool(host='api.registry.ns1.platformio.org', port=443): Max retries exceeded with url: /v3/search?

query=type%3A%22platform%22+name%3A%22espressif32%22 (Caused by ProtocolError('Connection aborted.', ConnectionResetError(10054, 'Eine vorhandene Verbindung wurde vom Remotehost geschlossen', None, 10054, None)))"

So what am I doing wrong???





Patrick Fitzgerald

February 20, 2022 at 12:15 pm

I followed this guide using a Rock 3A SBC and a Wemos D1 ESPduino board. Esptool flashed successfully when the ESPduino was on a USB2.0 port finishing with Hard resetting via RTS pin.

When the ESPduino was on a USB3.0 port esptool still autodetected the board but the output messages:

Connecting.....-----.....-----.....-----

failed to connect to ESP32: Timed out waiting for packet header

[Reply](#)



Sara Santos

February 21, 2022 at 12:08 pm

Hi.

press the on-board BOOT button when the dots start showing up on the debugging window.

Regards,

Sara

[Reply](#)



Patrick Fitzgerald

February 21, 2022 at 8:04 pm

Yes. Holding down the boot button did work.

I just had to hold it down longer. Then the upload completes successfully.

[Reply](#)



Sara Santos

February 22, 2022 at 4:04 pm

Great!

[Reply](#)



McKwen2

March 26, 2022 at 1:26 pm

Hi Sara. Thanks for this very informative post. Been referring to it while installing VS Code with PIO extension. Any tip on how to point the compiler to the proper location of libraries so I do not get “fatal error: ESPAsyncWebServer.h: No such file or directory” upload statements when compiling? I am 100% sure the libraries are in the ../include folder together with the header files. My .cpp file did not have any problems anymore meaning the includePath is in order. thanks.

[Reply](#)



Sara Santos

March 26, 2022 at 4:58 pm

Hi.

How are you installing the libraries on VS Code?

Regards,

Sara

[Reply](#)



mike w

March 29, 2022 at 2:34 pm

I am having an issue using pinMode in one of my projects. It seems that pinMode can't be found by platform IO. The Arduino.H file is included but no joy when trying to build. The code builds just fine without the calls to pinMode. The blink code example builds and runs with no issues. I suspect a path problem but just can't nail it down.

[Reply](#)



mike w

March 29, 2022 at 5:25 pm

Why would Platform IO be unable to find pinMode in one project yet find it in the blink example? In both cases Arduino.h is included. I suspect a path problem. Any clues you can give would be great.

[Reply](#)



Sara Santos

March 30, 2022 at 9:55 am

Hi Mike.

What is exactly the issue you're getting?

Regards,

Sara

[Reply](#)



mike w

March 30 2022 at 11:00 am

[Free eBooks](#)

[About](#)

[Contact](#)

[Courses Login](#)

[Get Courses](#) ▾



Thanks for getting back to me. I solved the issue, cockpit error.

[Reply](#)



Rubens

April 17, 2022 at 3:37 am

Hola

yo intente de programar un board esp32 hicé la tarrea como enseñe en la guia pero en la carga me dice falla

[Reply](#)



John

April 20, 2022 at 9:57 am

Hi Sara,

I am using Arduino IDE to develop software for ESP8266/ESP32 boards. I intend to try PIO: if I upload to these boards with PIO, can I reuse these boards should I revert to Arduino IDE ?

[Reply](#)



Rodrigo

May 19, 2022 at 4:11 pm

Hi Sara! How do I use the WiFiManager library in Visual Code with PlatformIO IDE?
Thank you very much



Sara Santos
May 20, 2022 at 8:51 pm

Hi.

Search for the library in the Libraries tab in the PIO Home menu. Then, there should be a button to add the library to your project.

Regards,
Sara

[Reply](#)



Ceesa
July 7, 2022 at 9:21 am

Thank you. This got me started with VS Code and the ESP32. The only instructions missing were the for the ESP USB driver. I think some mention needs to be made before the running of VS Code because people that are new to the ESP or doing new installs of VS Code or on new computer are likely to run into the problem.

Of course the other place to mention drivers could be under Troubleshooting. My particular ESP32 required me to hold down Boot to load firmware.

Thanks again.

[Reply](#)



Sara Santos
July 11, 2022 at 8:44 am



REgards,
Sara

[Reply](#)



Imran Mumtaz

December 20, 2022 at 5:40 pm

Hi Sara,

I have started using platformio and am facing the issue that when I create new project my project wizard is too slow and is taking too much time and stuck in the project initialization. Kindly guide what can I do to fix this?

Thanks,

Imran

[Reply](#)



Sarah Justus

December 24, 2022 at 2:51 am

Not her but I had this same issue and I fixed it by running Visual Studio Code as admin.

[Reply](#)



Imran Mumtaz

December 25, 2022 at 8:45 am

Thanks I am going to try this solution and let you know if it works which I hope it will.

[Reply](#)



Mezac

February 5, 2023 at 3:50 pm

I installed VS CODE, PlatformIO IDE, I'm recording on DOIT ESP32 DEVKIT V1, everything works perfectly.

But I'm looking for simple codes like the use of if, else, etc... and I can't find examples. Please where can I find sample codes to use in VS CODE\PlatformIO ?

[Reply](#)



Sara Santos

February 5, 2023 at 7:14 pm

You can use the same code you use in Arduino IDE on VS Code.

Regards,

Sara

[Reply](#)



Mezac

February 13, 2023 at 1:52 pm

Thank you for turning the lights on!

[Reply](#)



Philipp

Hello,

every time i tryed to upload the code i get a Error at the

" Linking .pio\build\lolin32_lite\firmware.elf "

Im not using the esp32 lolin32_lite and im not that good with that compiler stuff

pls help 😊

[Reply](#)



Claus

April 30, 2023 at 1:46 pm

Followed the installation guide – but when I create a project

```
[env:d1_wroom_02]
platform = espressif8266
board = d1_wroom_02
framework = esp8266-nanos-sdk
monitor_speed = 115200
```

no src/main.cpp is auto injected

when I create one

```
#include <Arduino.h>
```

```
#define LED 2
```

```
void setup() {
// put your setup code here, to run once:
Serial.begin(115200);
pinMode(LED, OUTPUT);
}
```

```
void loop() {
// put your main code here, to run repeatedly...
```

```
Serial.println("LED is on");
delay(1000);
digitalWrite(LED, LOW);
Serial.println("LED is off");
delay(1000);
}
```

it cannot find <Arduino.h>

Is there some prerequisites not mentioned ?

[Reply](#)



Sara Santos

May 2, 2023 at 8:30 am

Hi.

When creating the project, the framework should be arduino.

You selected a different one.

Regards,

Sara

[Reply](#)



Claus

May 4, 2023 at 8:45 pm

Hi Sara

Arduino Platform – also when the target is an ESP32 D1 Live Wroom

[Reply](#)

Yes – it works – thanks for the article and hint.

[Reply](#)



Rocky

October 9, 2023 at 5:29 pm

Great tutorial! It was really helpful!

I have Ubuntu and python 3.10.12, and there was just one more step to get platformIO going, for anyone getting an interpreter error

```
sudo apt install python3-venv
```

Found it on

<https://community.platformio.org/t/platformio-ide-cant-find-python-interpreter-on-linux/24262/3>

[Reply](#)



Andrii

October 11, 2023 at 8:15 pm

Perfect tutorial! Thanks a lot.

The only thing I've modified in the code is instead of "LED" I used "LED_BUILTIN" because It did not blink otherwise.

[Reply](#)



Daddy McGuiness

Free eBooks

About

Contact

Courses Login

Get Courses ▾



Really great, worked first time. Thanks

[Reply](#)

Leave a Comment

Name *

Email *

Website

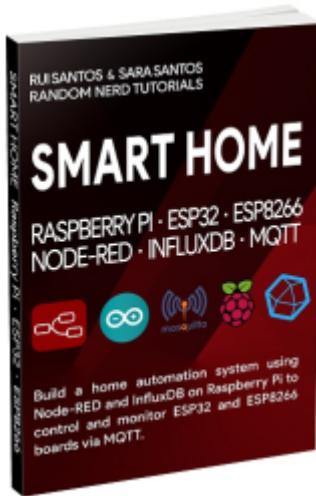
Notify me of follow-up comments by email.

Notify me of new posts by email.

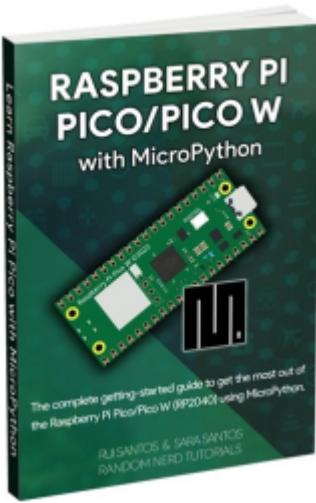
[Post Comment](#)



[Edition\) Course](#) » Complete guide to program the ESP32 with Arduino IDE!



[SMART HOME with Raspberry Pi, ESP32, and ESP8266](#) » learn how to build a complete home automation system.



🔥 [Learn Raspberry Pi Pico/Pico W with MicroPython](#) » The complete getting started guide to get the most out of the the Raspberry Pi Pico/Pico W (RP2040) microcontroller board using MicroPython programming language.