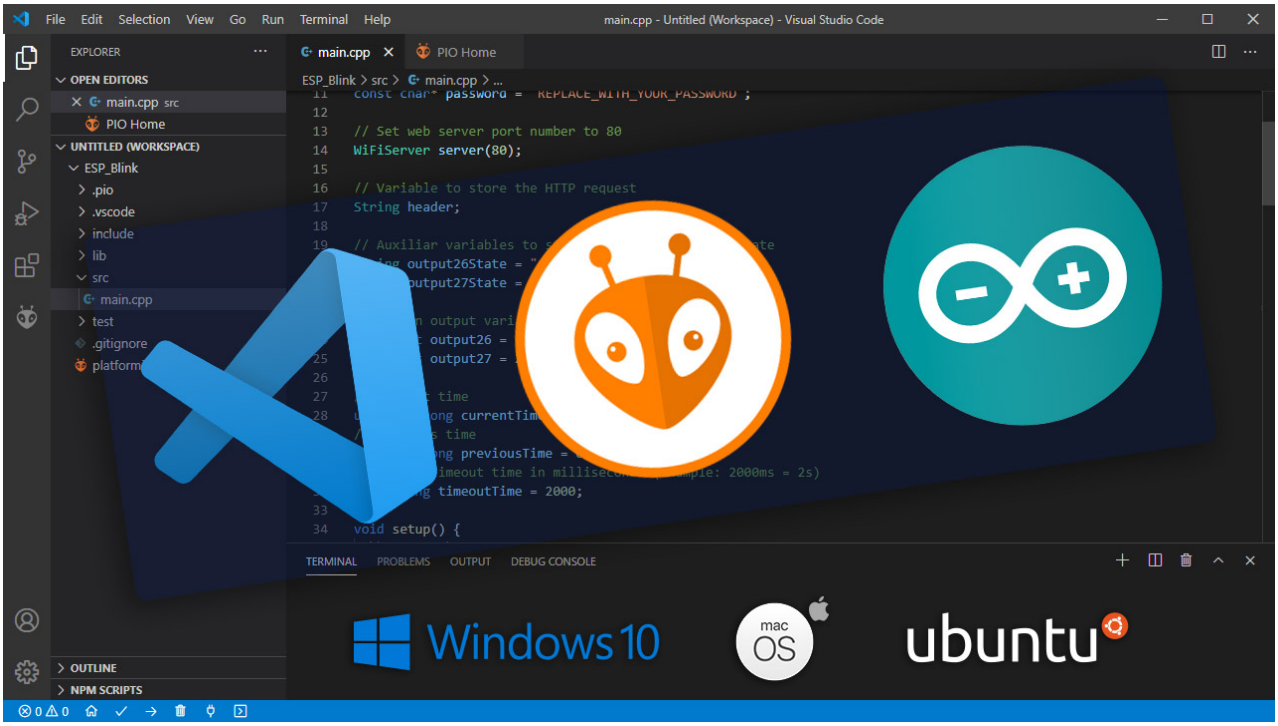
Getting Started with VS Code and PlatformIO IDE for ESP32 and ESP8266 (Windows, Mac OS X, Linux Ubuntu)

randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino

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.

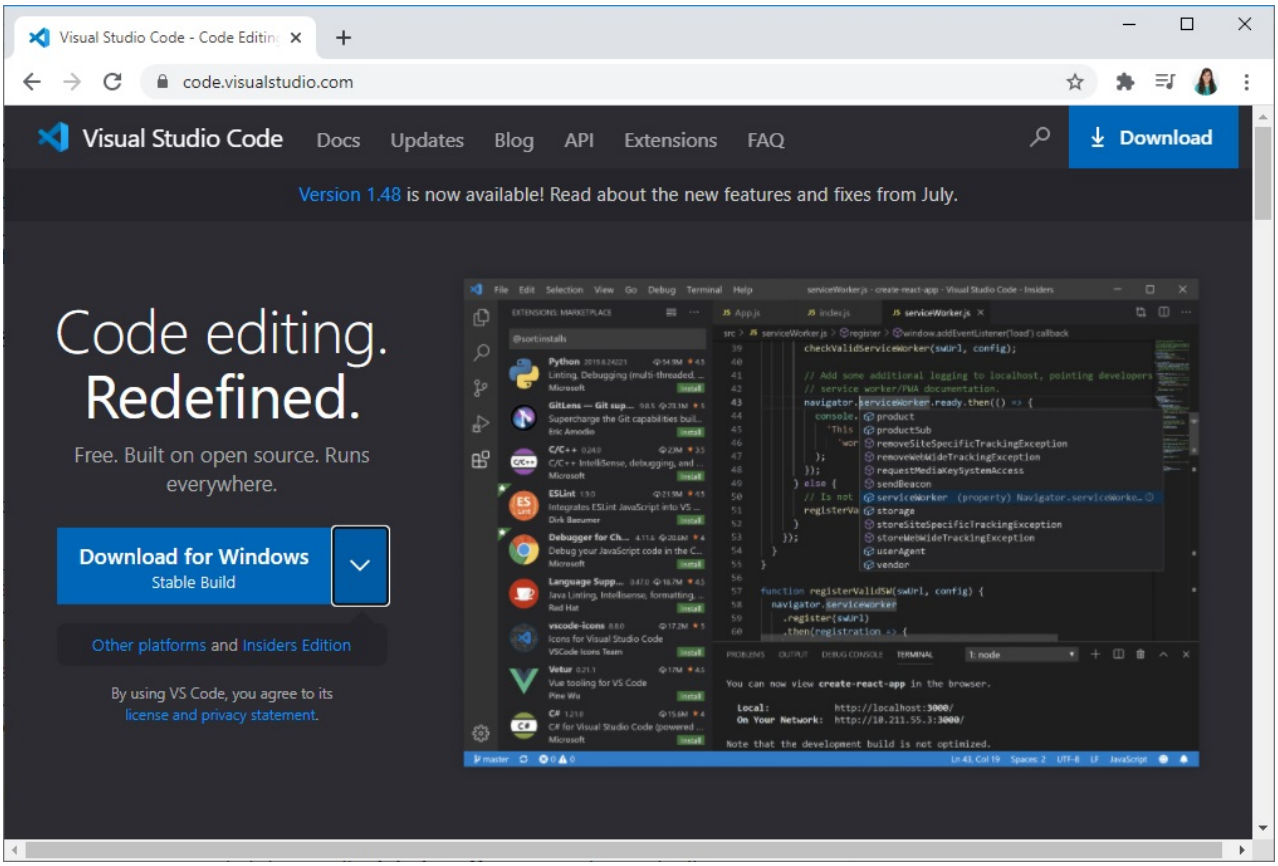


The Arduino IDE works great for small applications. However, for advanced projects with more than 200 lines of code, multiple files, and other advanced features like auto completion and error checking, VS Code with the PlatformIO IDE extension is the best alternative.

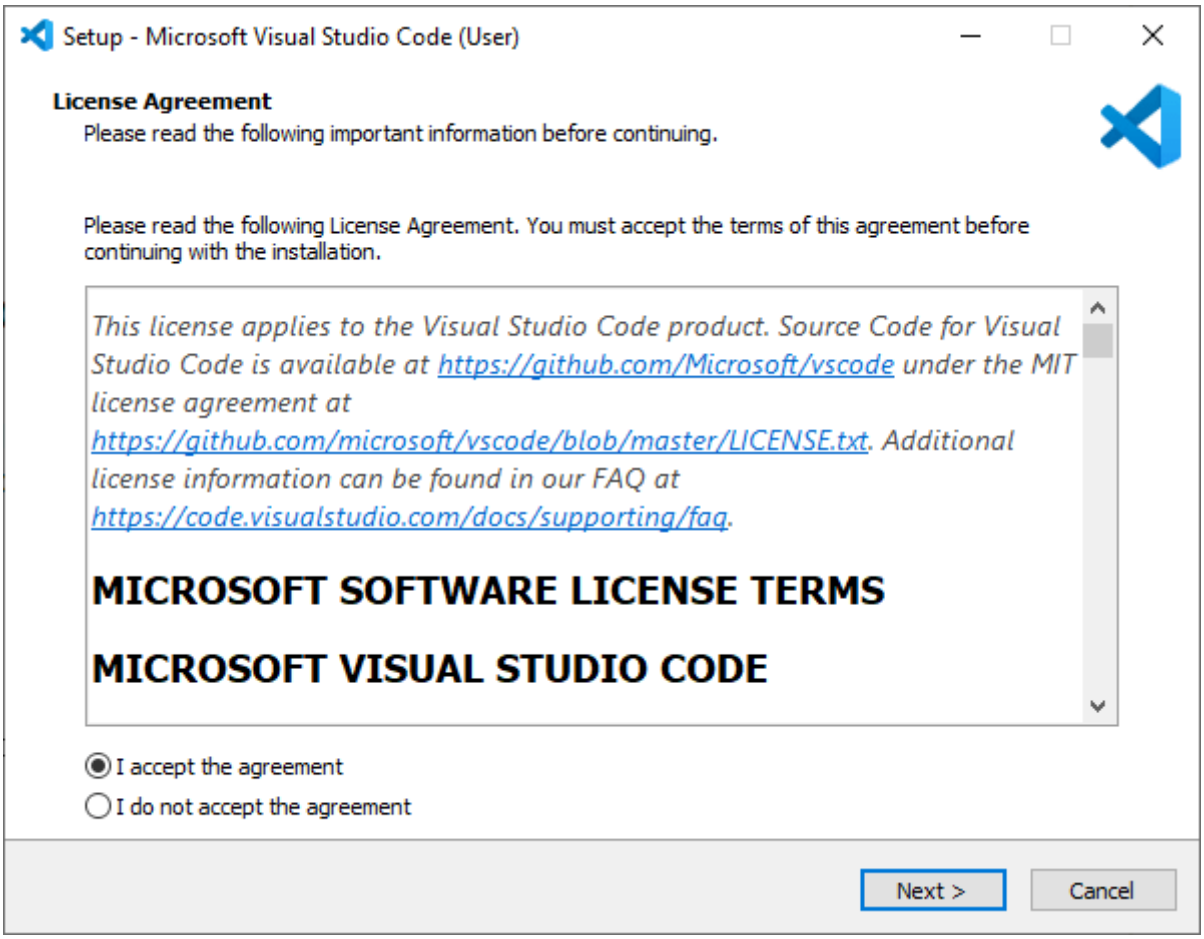
In this tutorial, we'll cover the following topics:

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

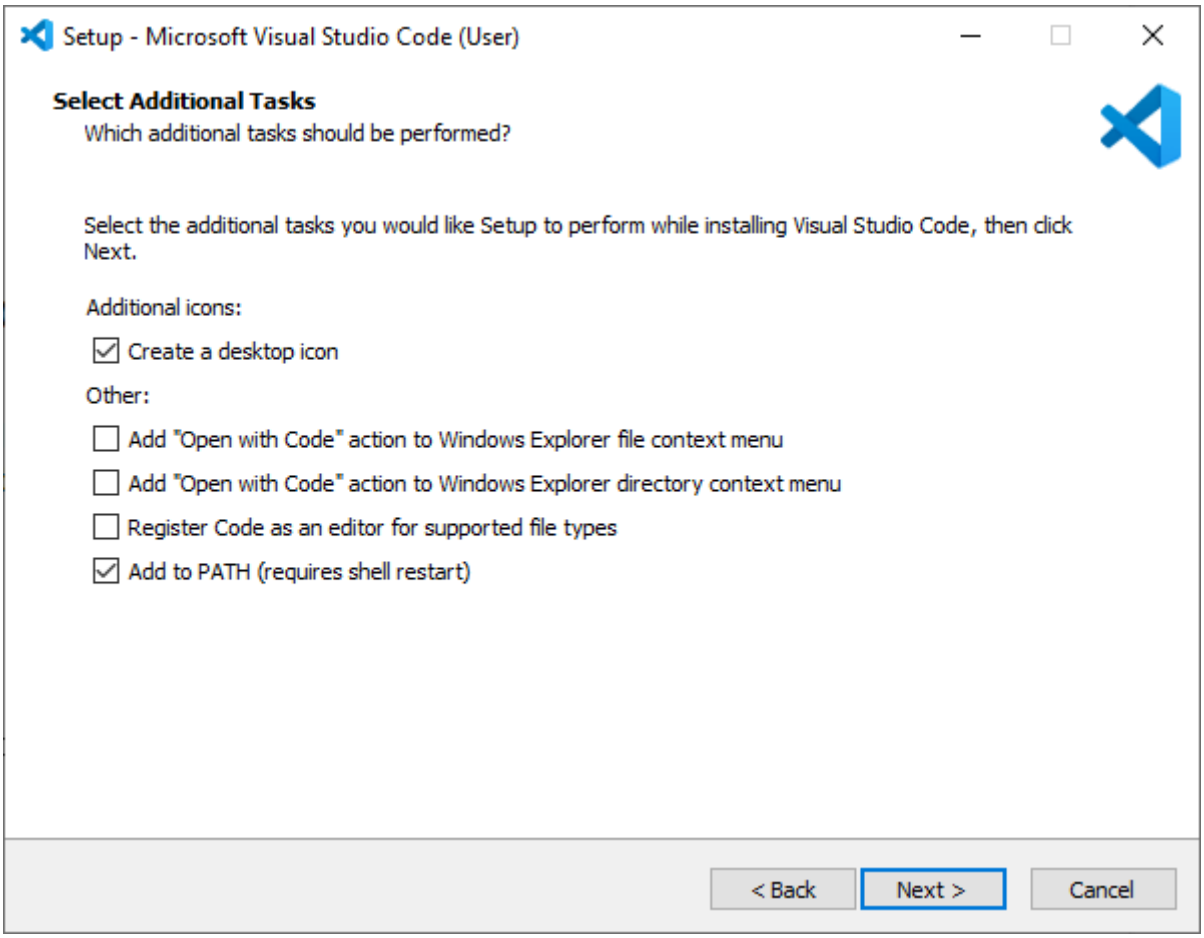
Go to https://code.visualstudio.com/ and download the stable build for your operating system (Windows).



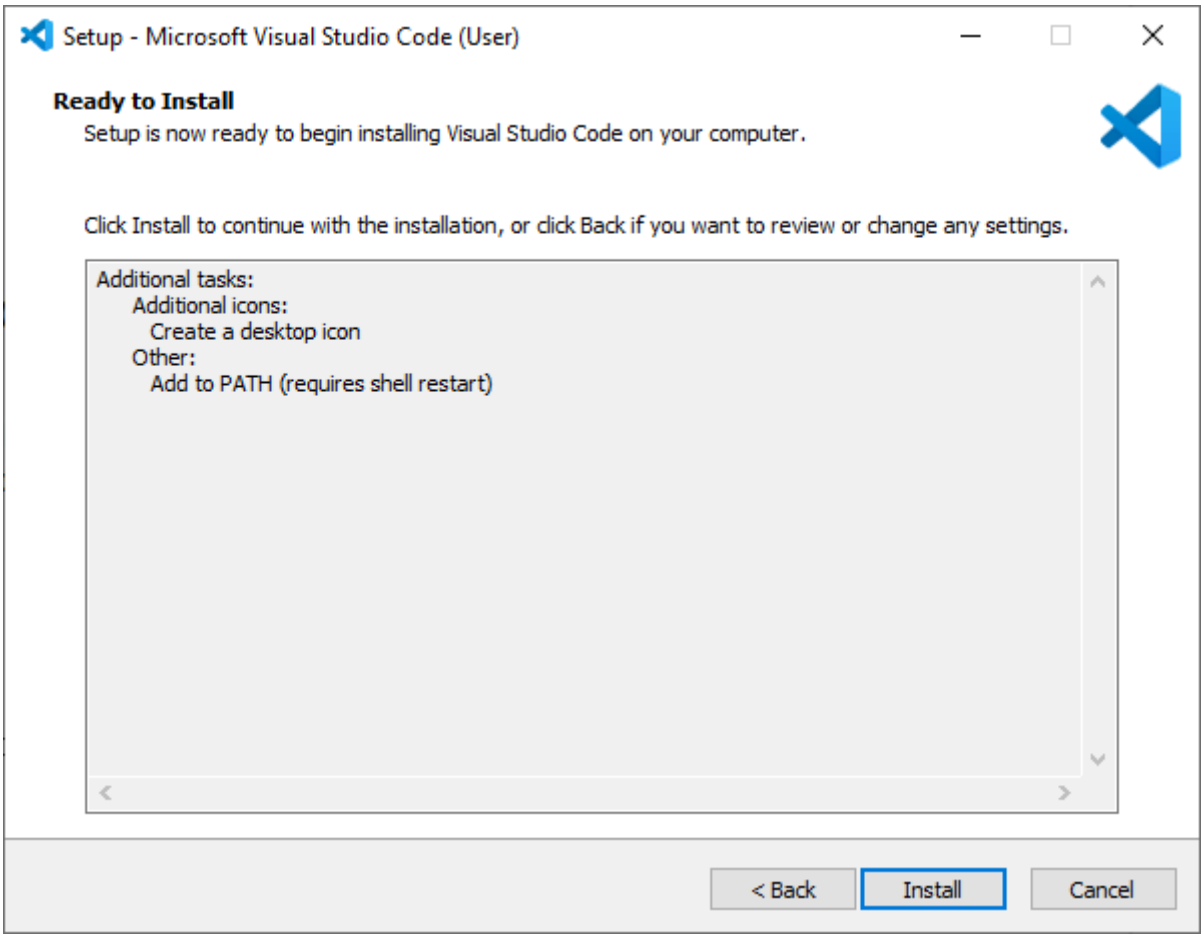
Click on the installation wizard to start the installation and follow all the steps to complete the installation. Accept the agreement and press the **Next** button.



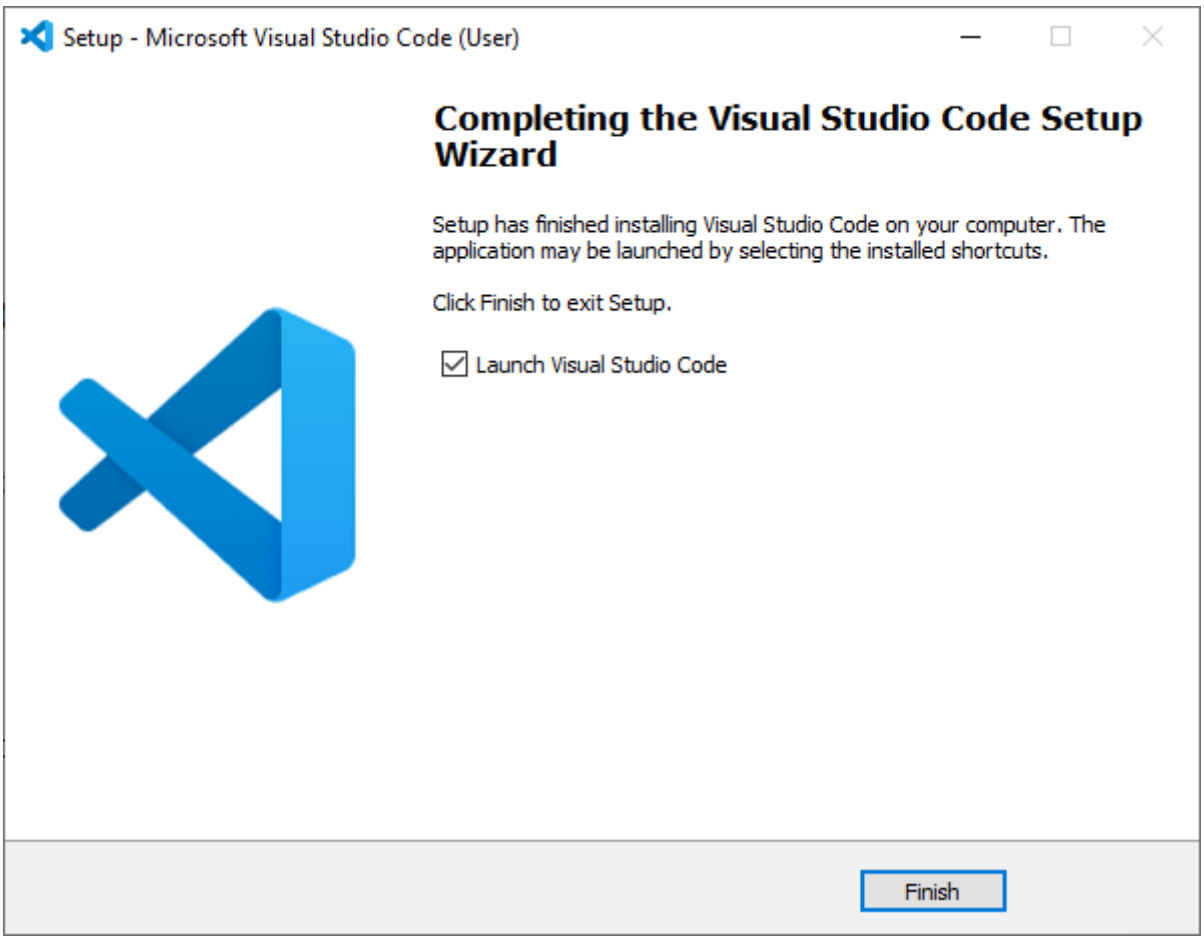
Select the following options and click **Next**.



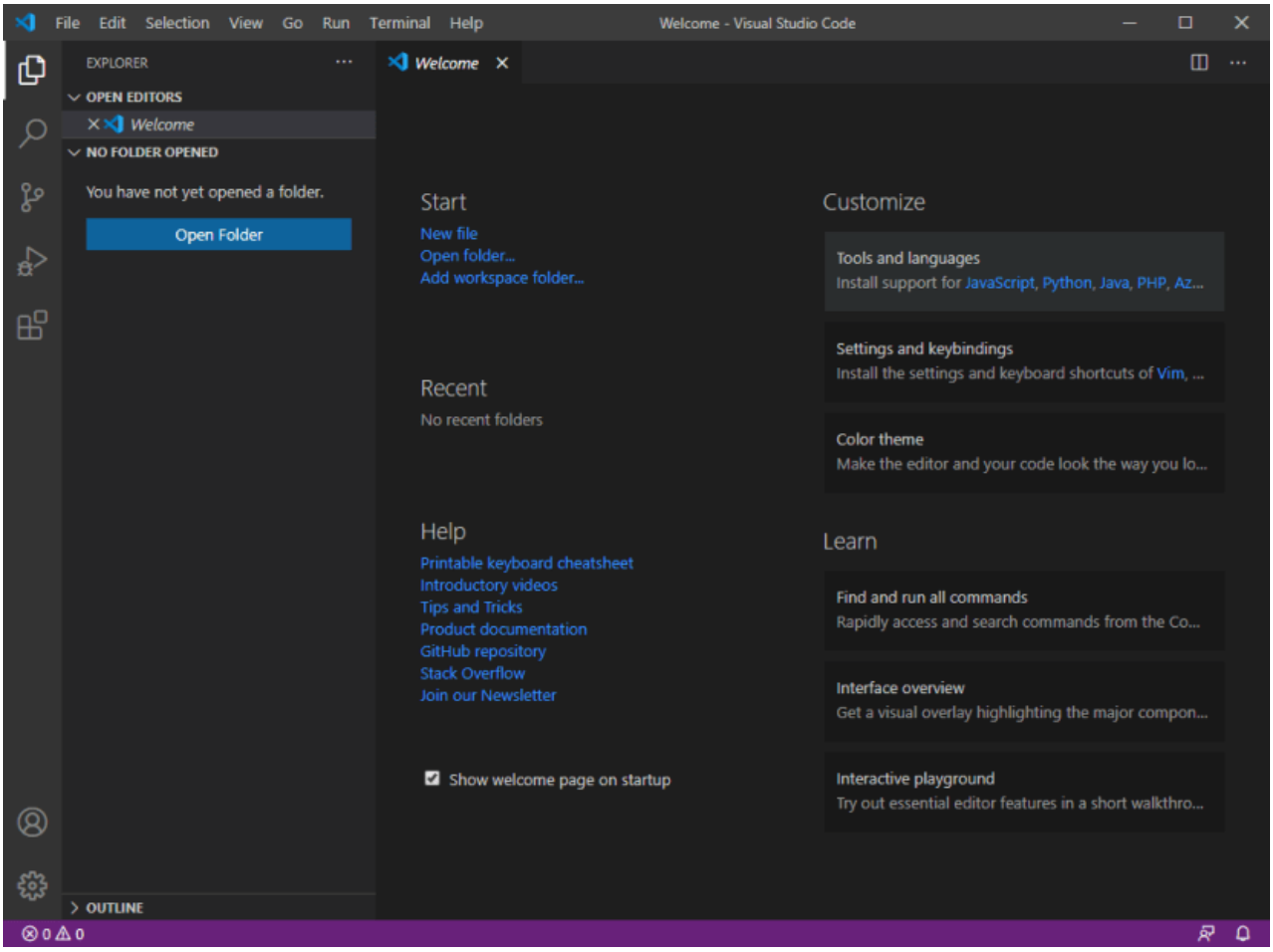
Press the **Install** button.



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 version.



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

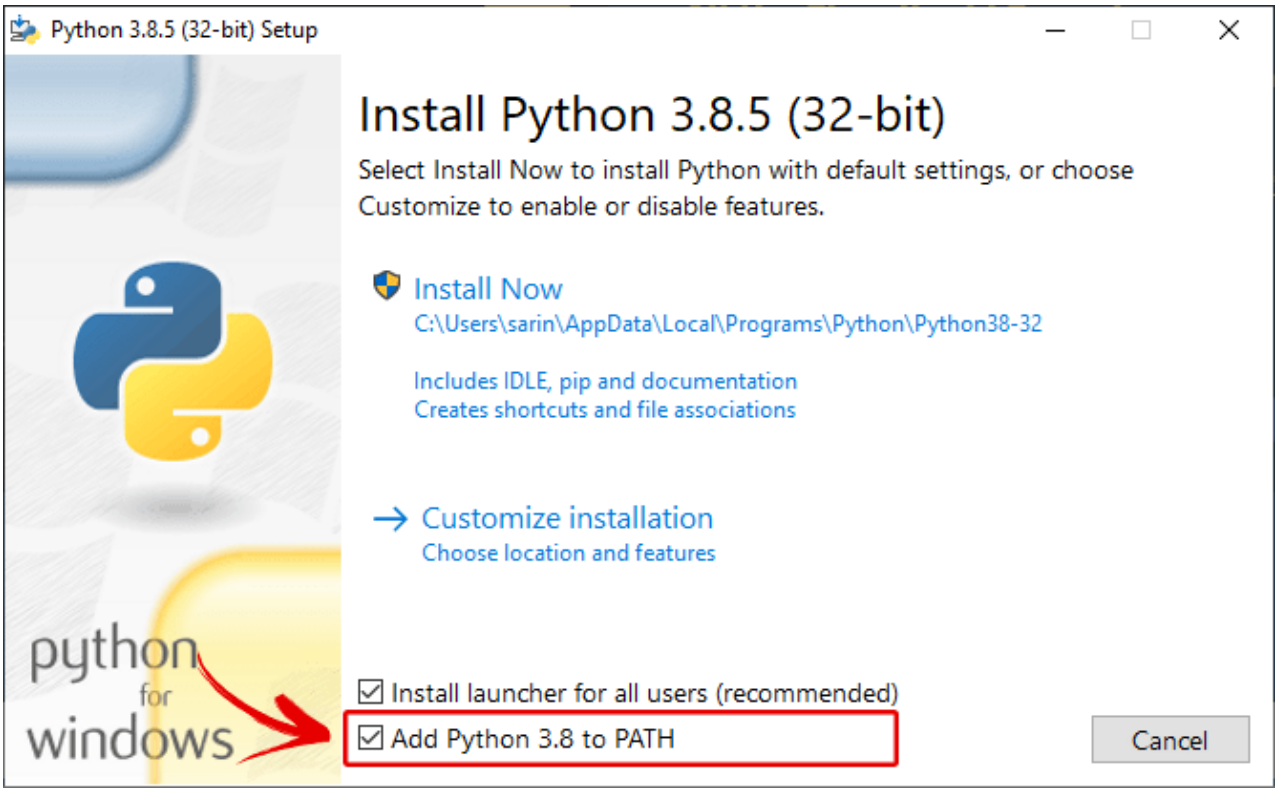
Installing Python on Windows

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.

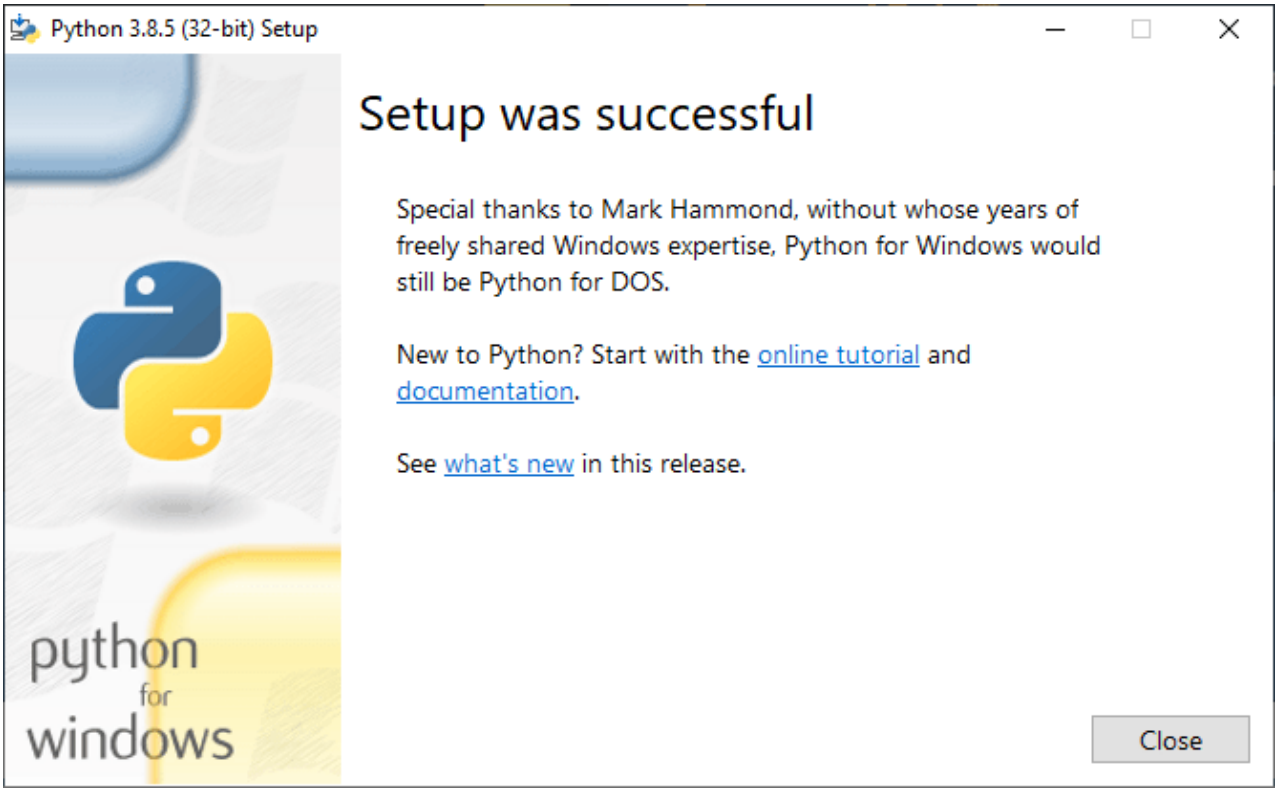
Open the downloaded file to start the Python installation wizard.

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.

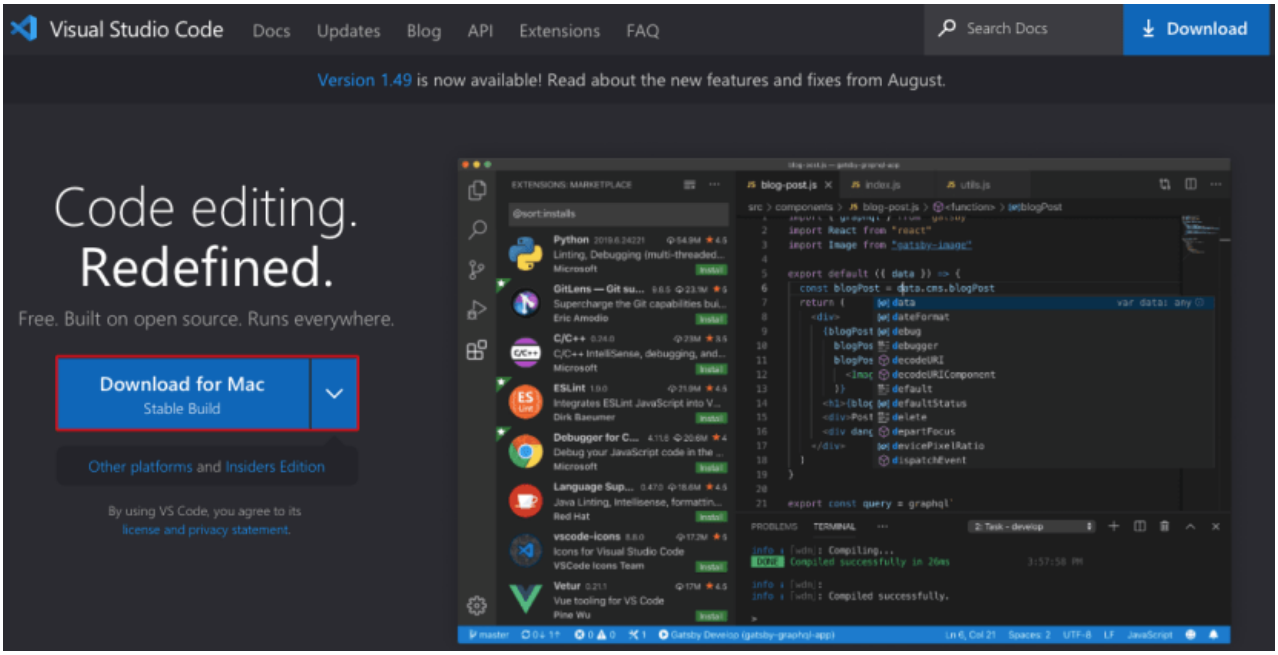


You can click the **Close** button.

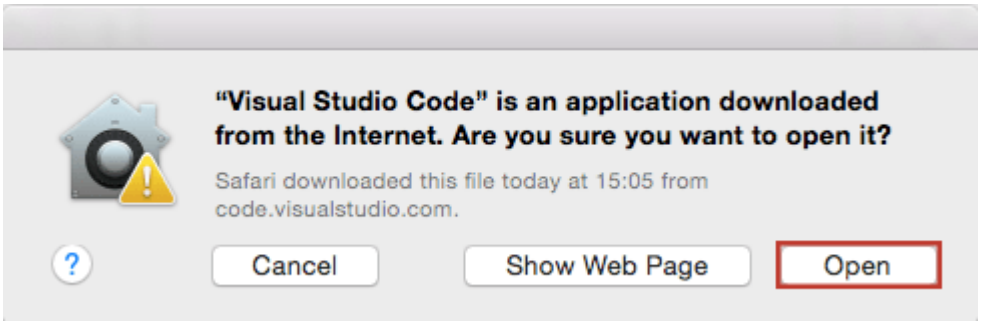
Now, go to this section to install PlatformIO IDE extension.

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).

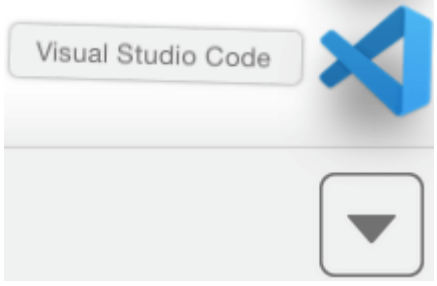


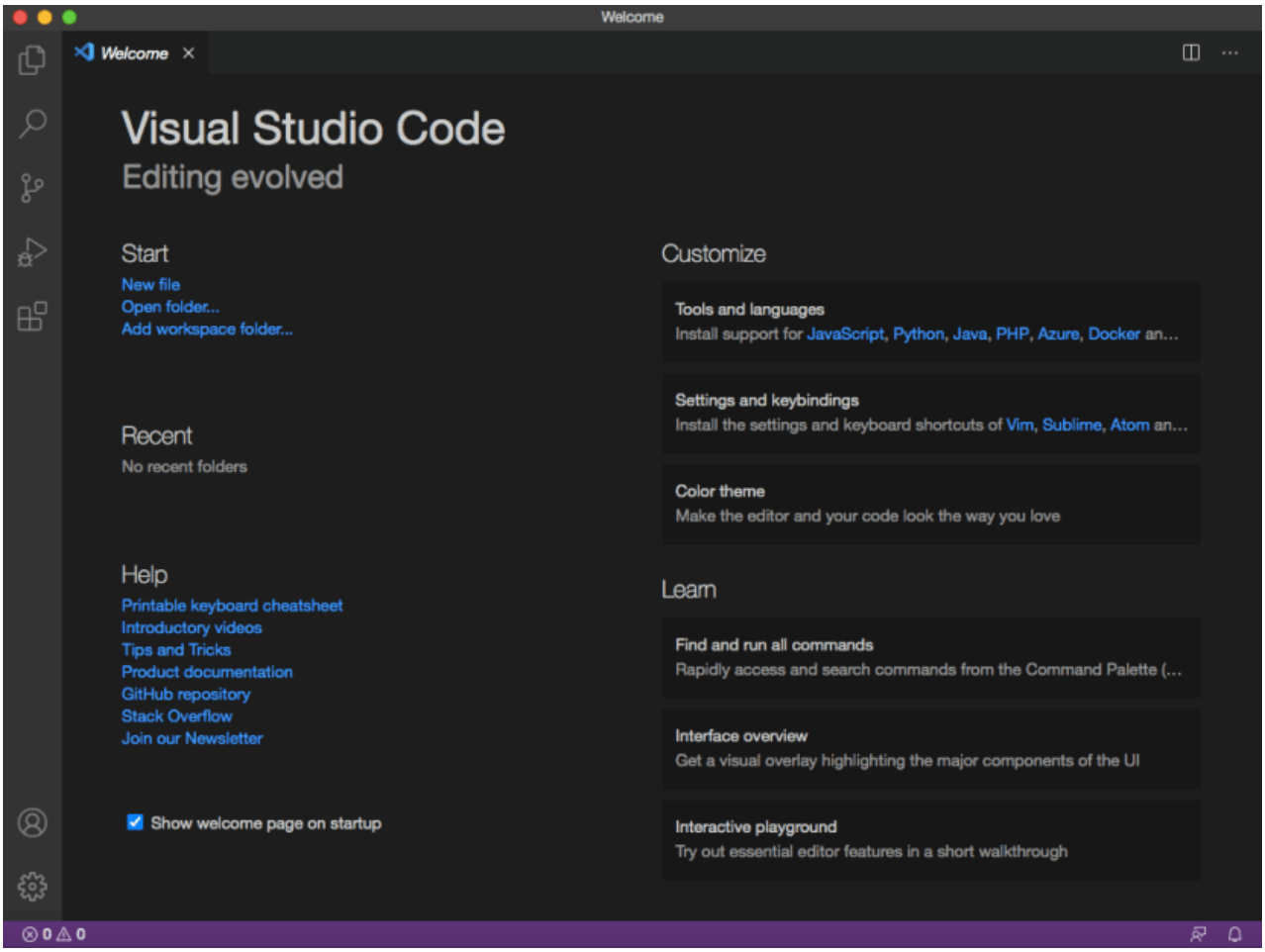
After downloading the Visual Studio Code application file, you'll be prompted with the following message. Press the "**Open**" button.



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.





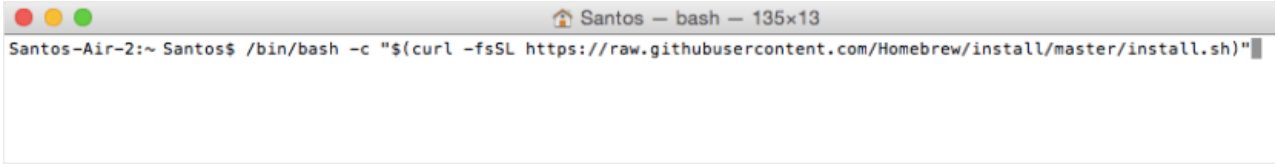
That's it. Visual Studio Code was successfully installed.

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.

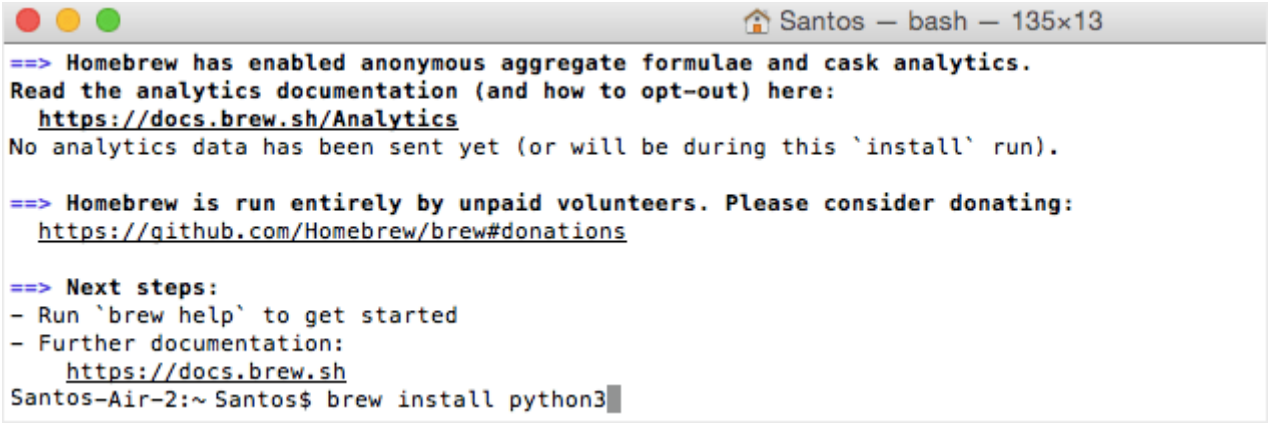
To install Python I'll be using Homebrew. If you don't have the brew command available, type the next command:

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



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

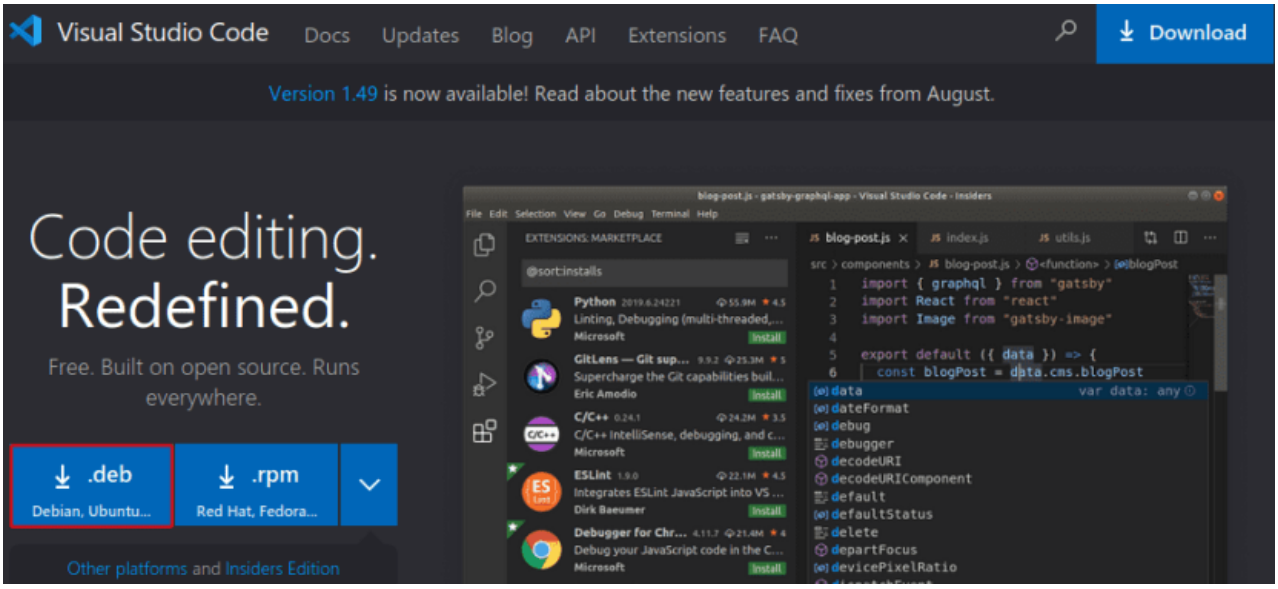
```
$ brew install python3
```



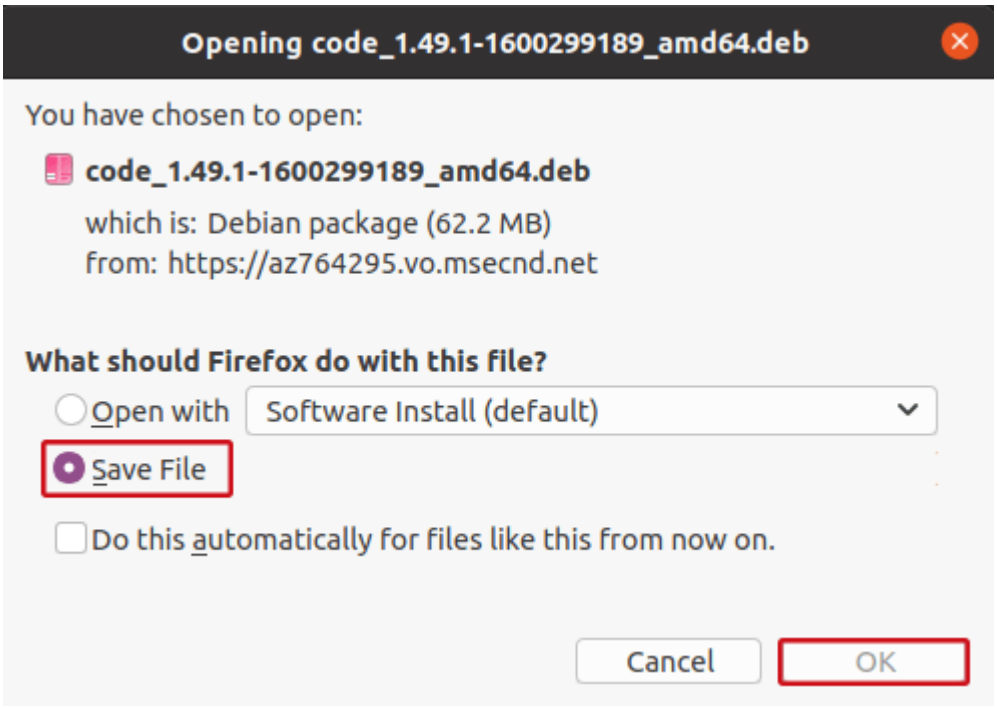
Now, go to this section to install PlatformIO IDE extension.

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).



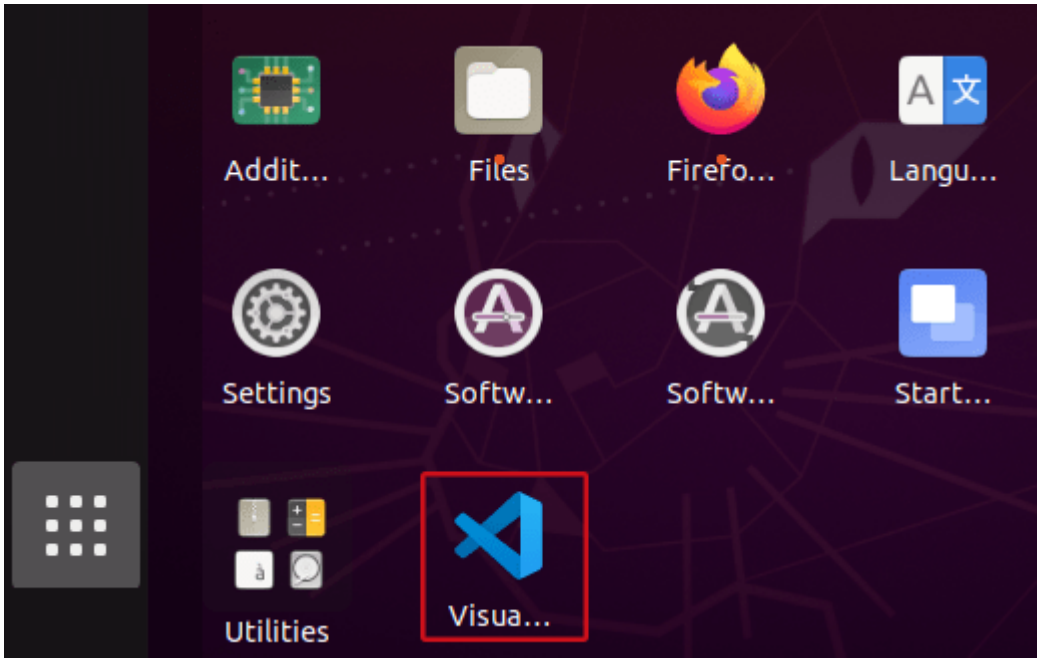
Save the installation file:



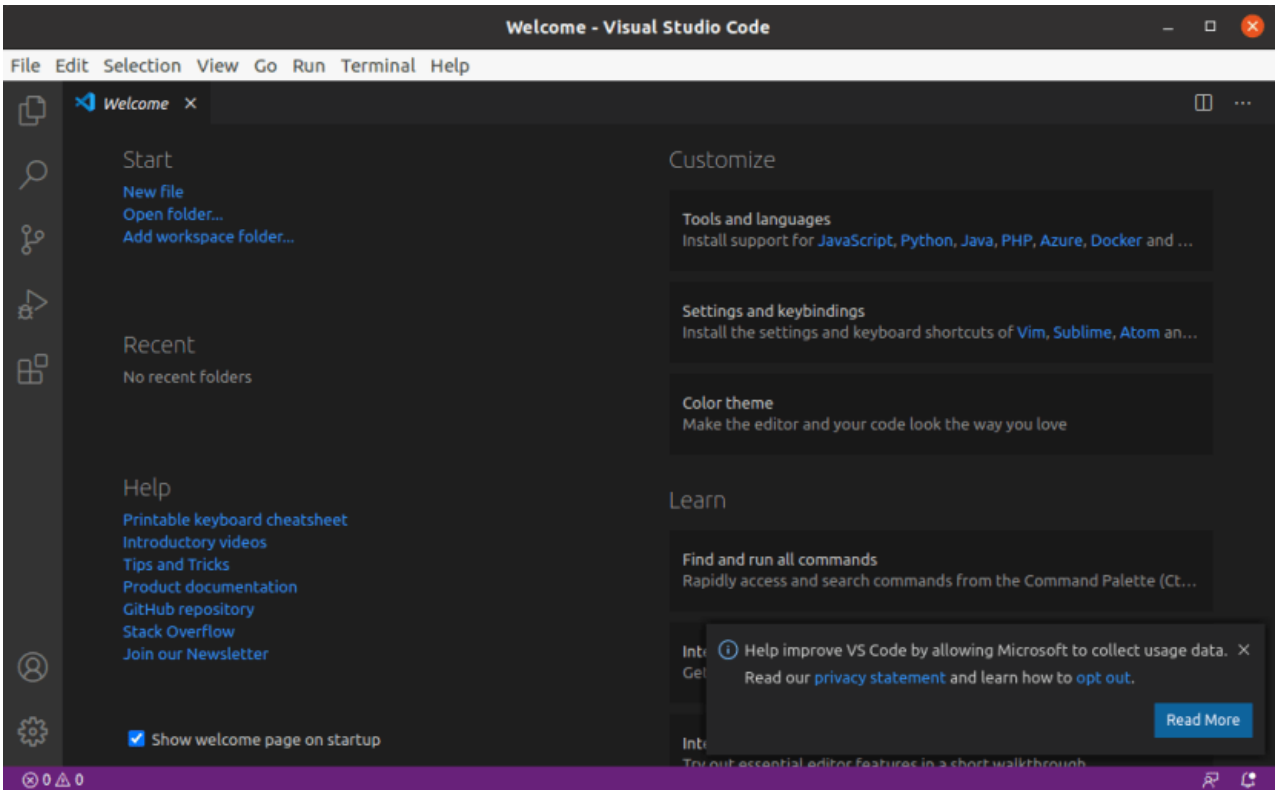
To install it, open a Terminal windows, navigate to your Downloads folder and run the following command to install VS Code.

```
$ 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.



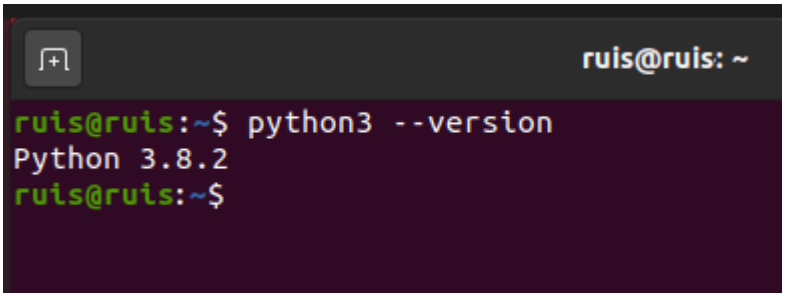
That's it. Visual Studio Code was successfully installed.

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
```



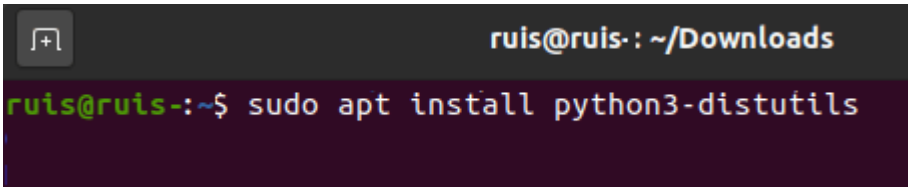
As you can see in the preceding figure, Python 3.8.2 is already installed.

If you don't have Python 3.8.X installed, run the next command to install it:

```
$ 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
```



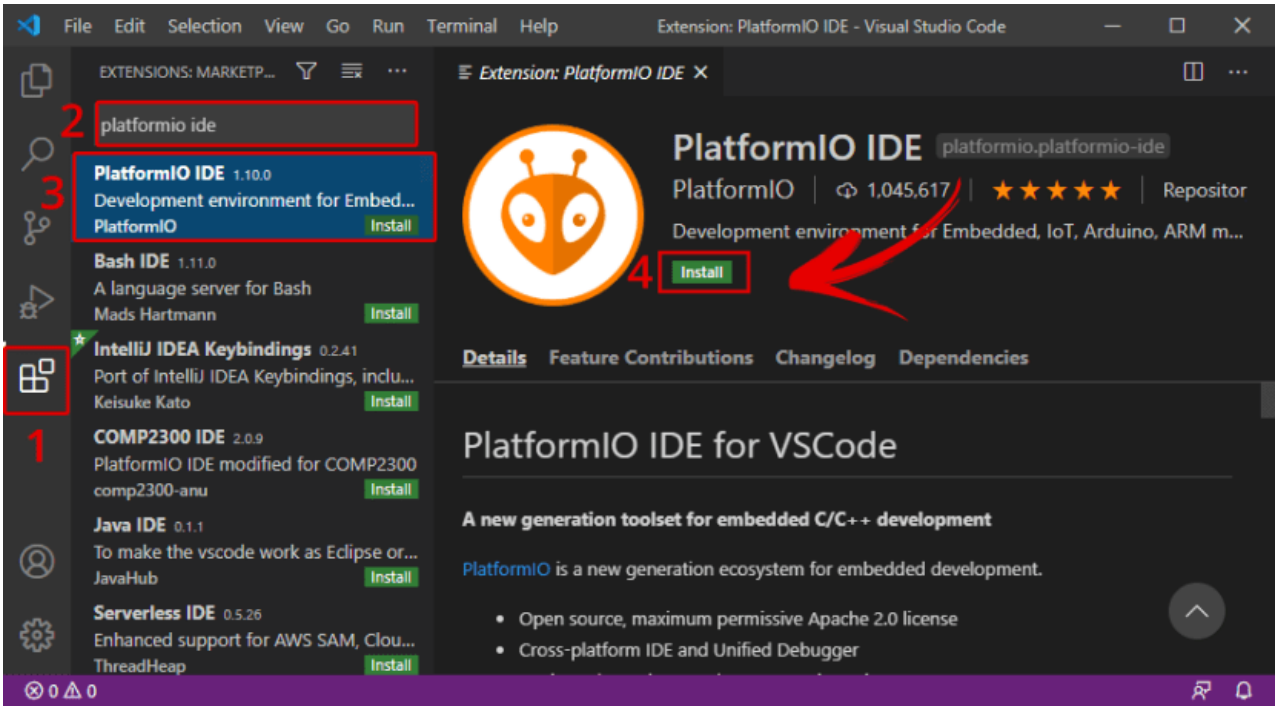
Now, go to this section to install PlatformIO IDE extension.

Installing PlatformIO IDE Extension on VS Code

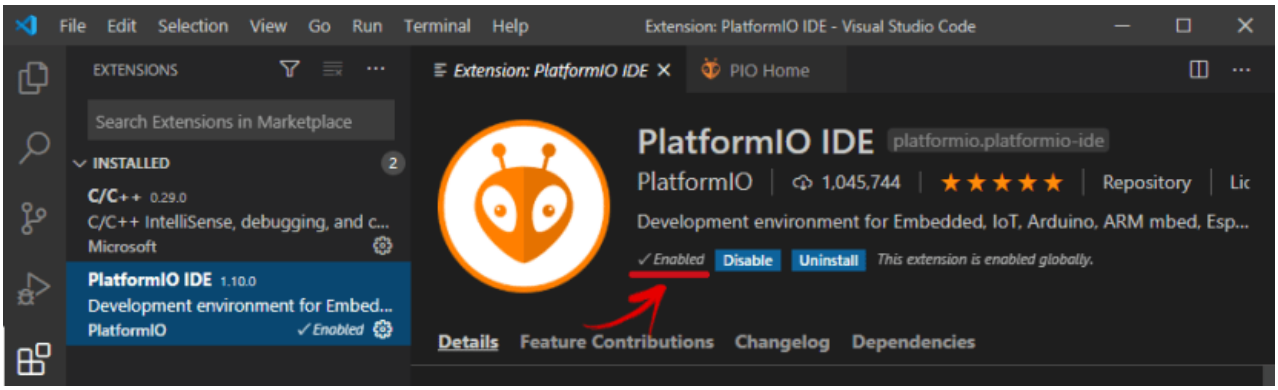
It is possible to program the ESP32 and ESP8266 boards using VS Code with the PlatformIO IDE extension. Follow the next steps to install the PlatformIO IDE extension.

Open VS Code:

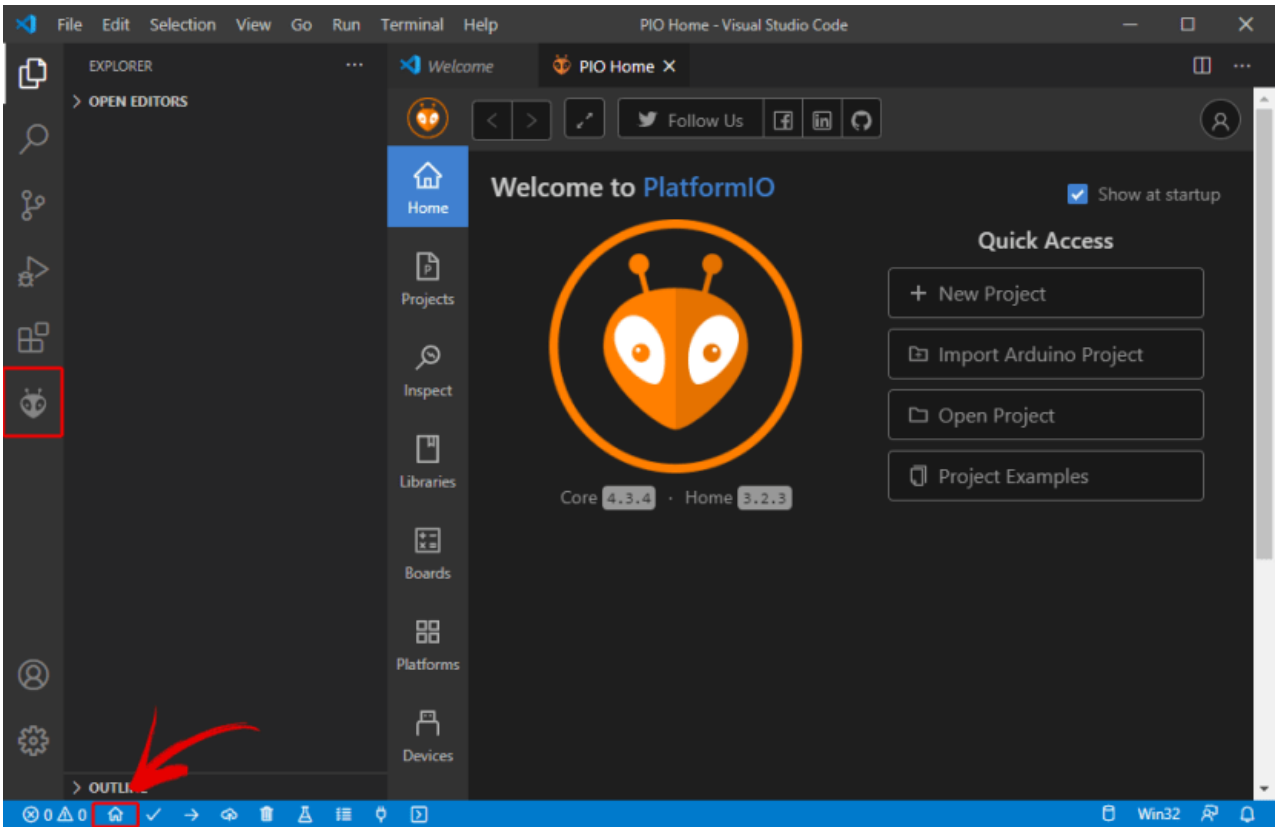
1. Click on the **Extensions** icon or press **Ctrl+Shift+X** to open the **Extensions** tab
2. Search for "**PlatformIO IDE**"
3. Select the first option
4. Finally, click the **Install** button (Note: the installation may take a few minutes)



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



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



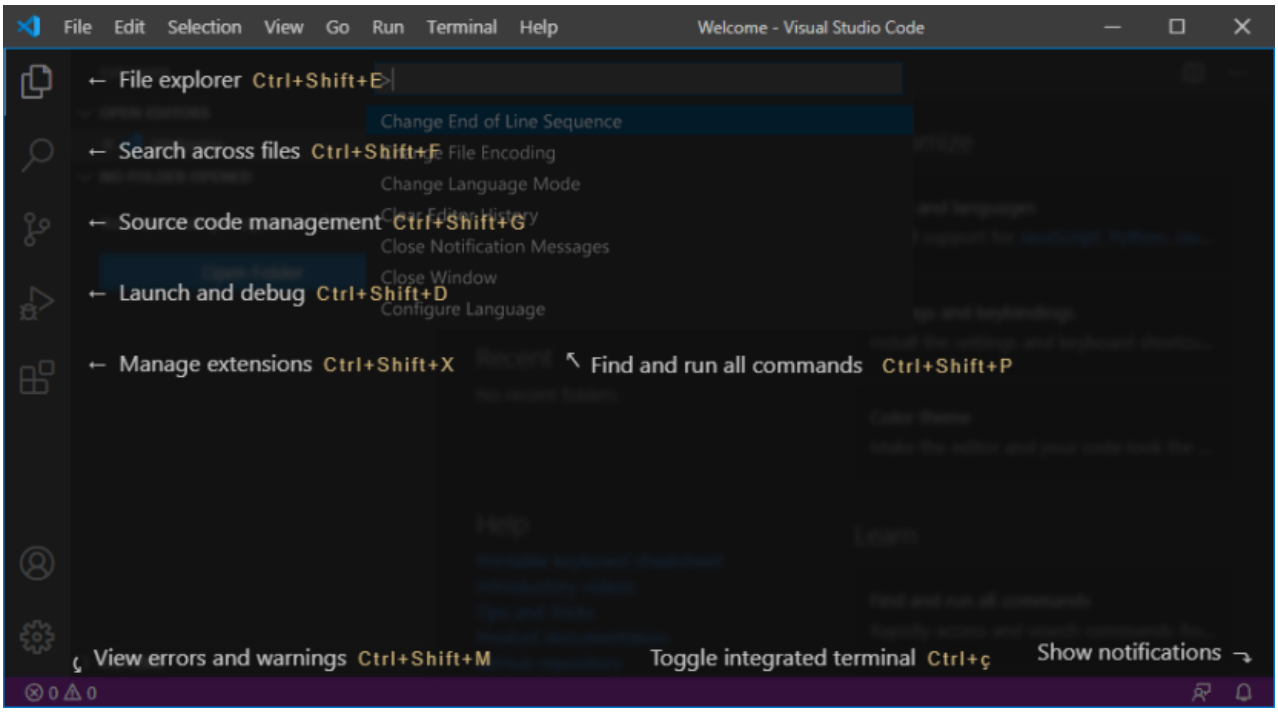
That's it, PlatformIO IDE extension was successfully added to VS Code.

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.

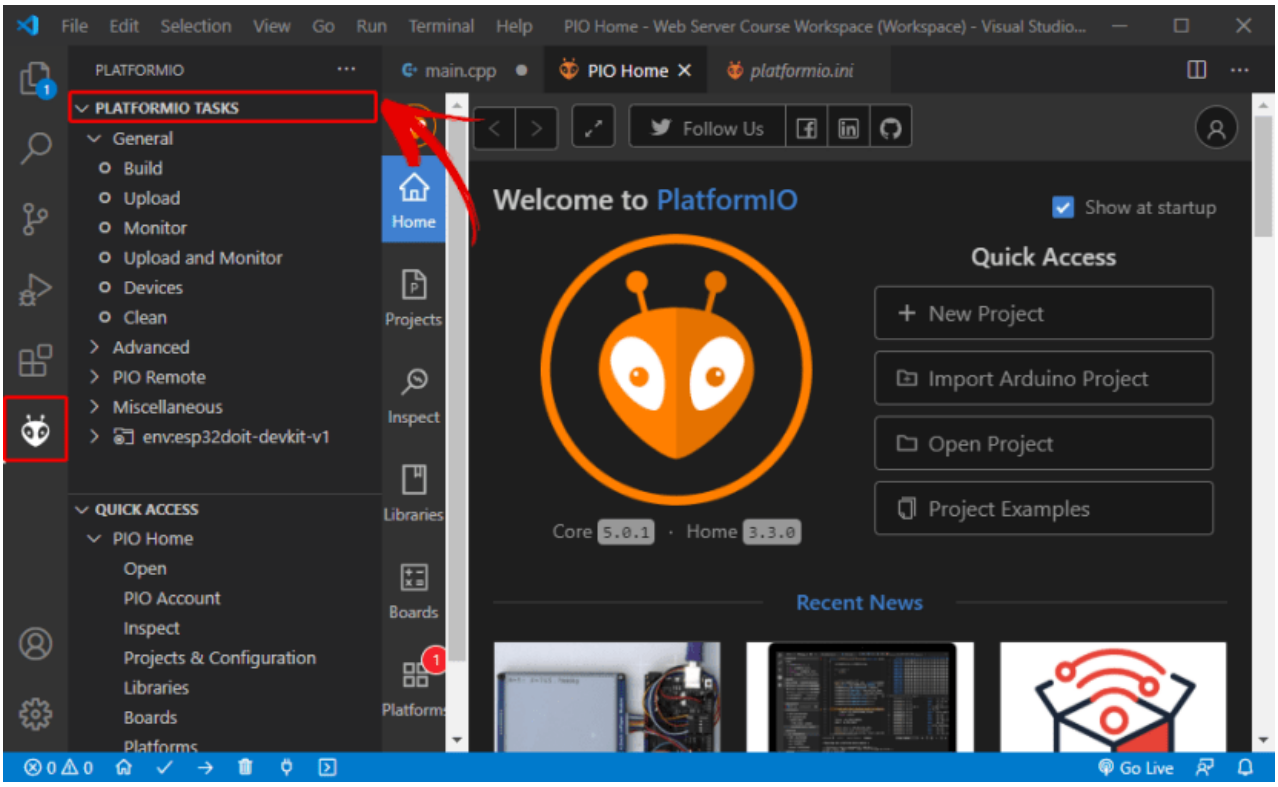
At the bottom, there's a blue bar with PlatformIO commands.

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

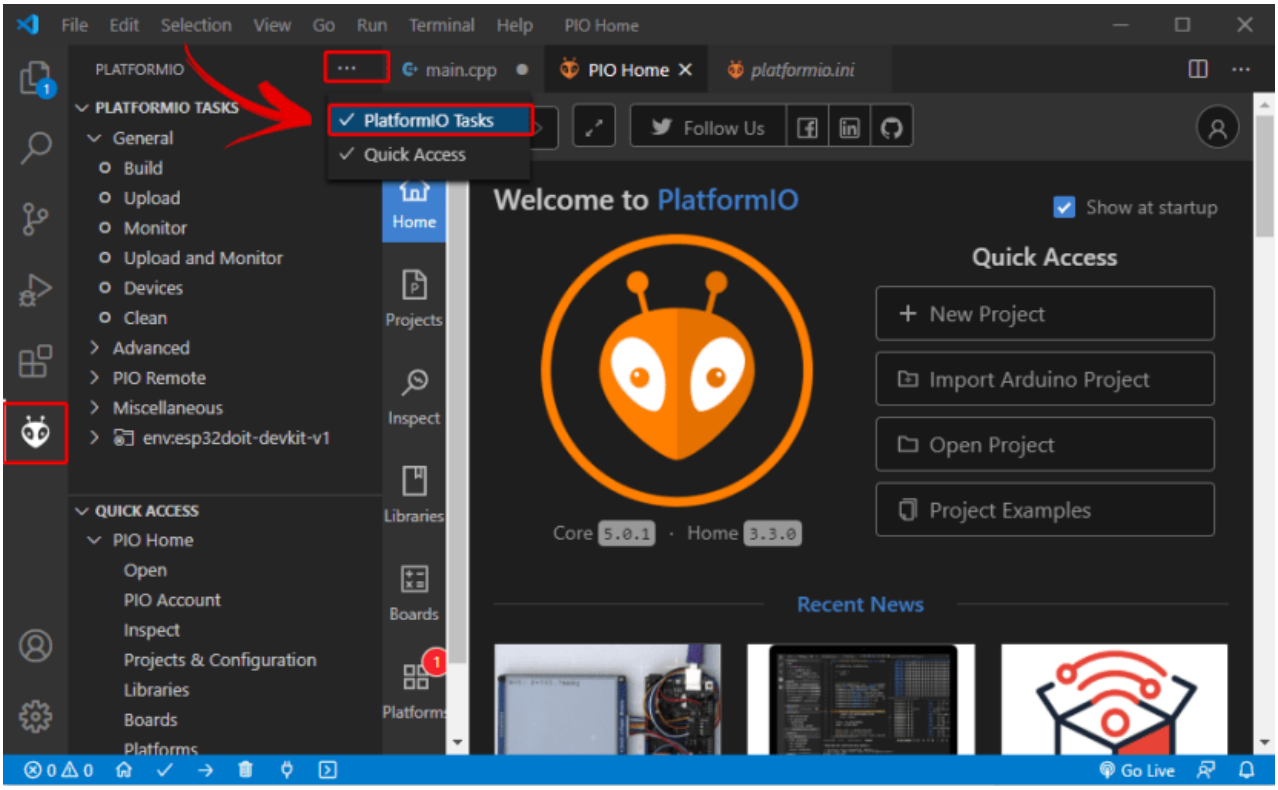
- PlatformIO Home
- Build/Compile
- Upload
- Clean
- Serial Monitor
- New Terminal

If you hover your mouse over the icons, it will show what each icon does.

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.

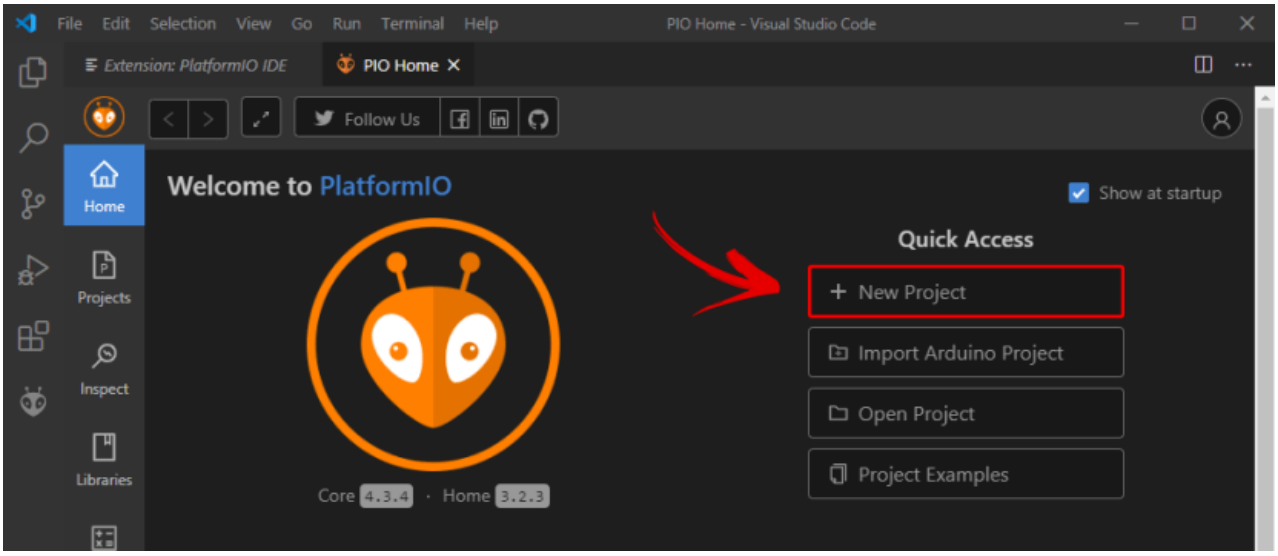


PlatformIO IDE Overview

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

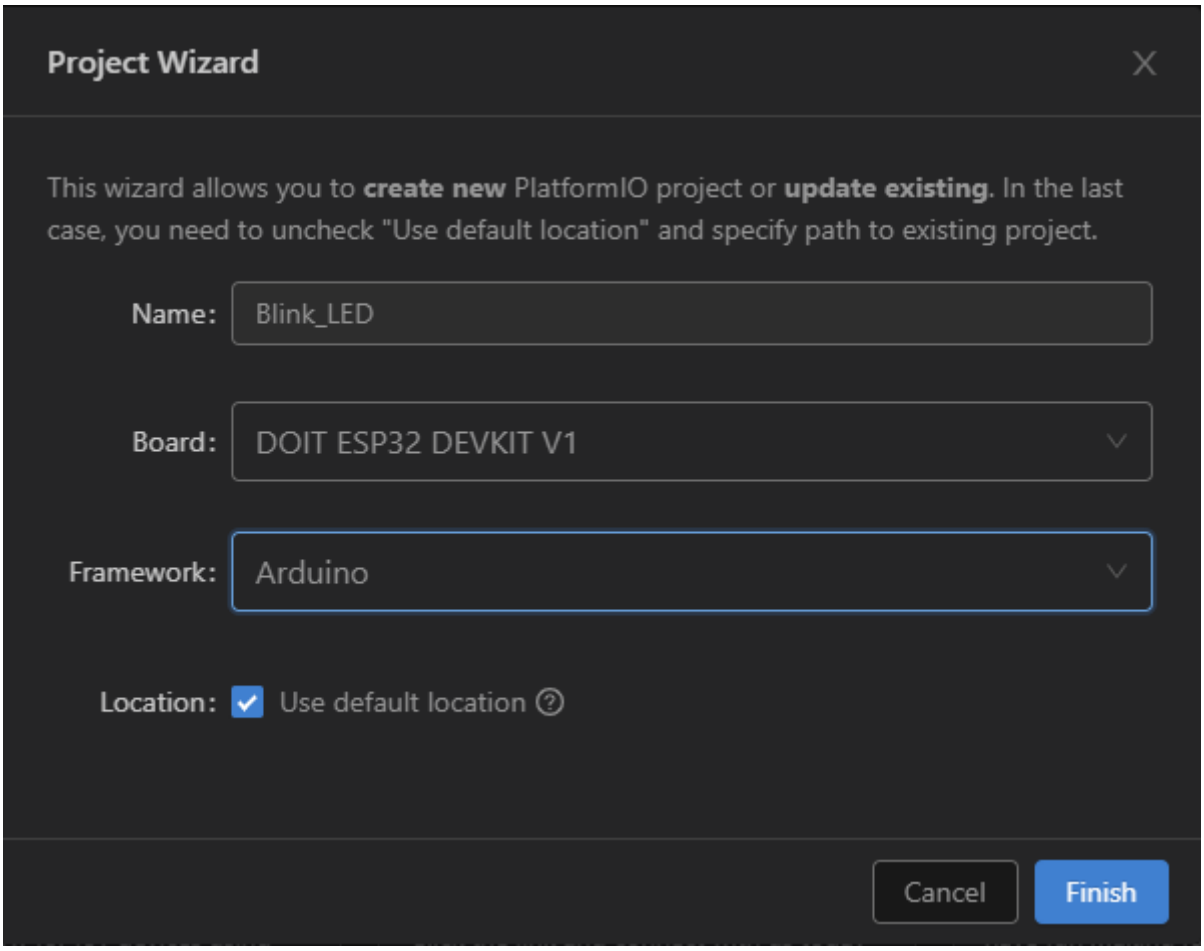
On VS Code, click on the PlartfomIO **Home** icon. Click on **+ New Project** to start a new project.



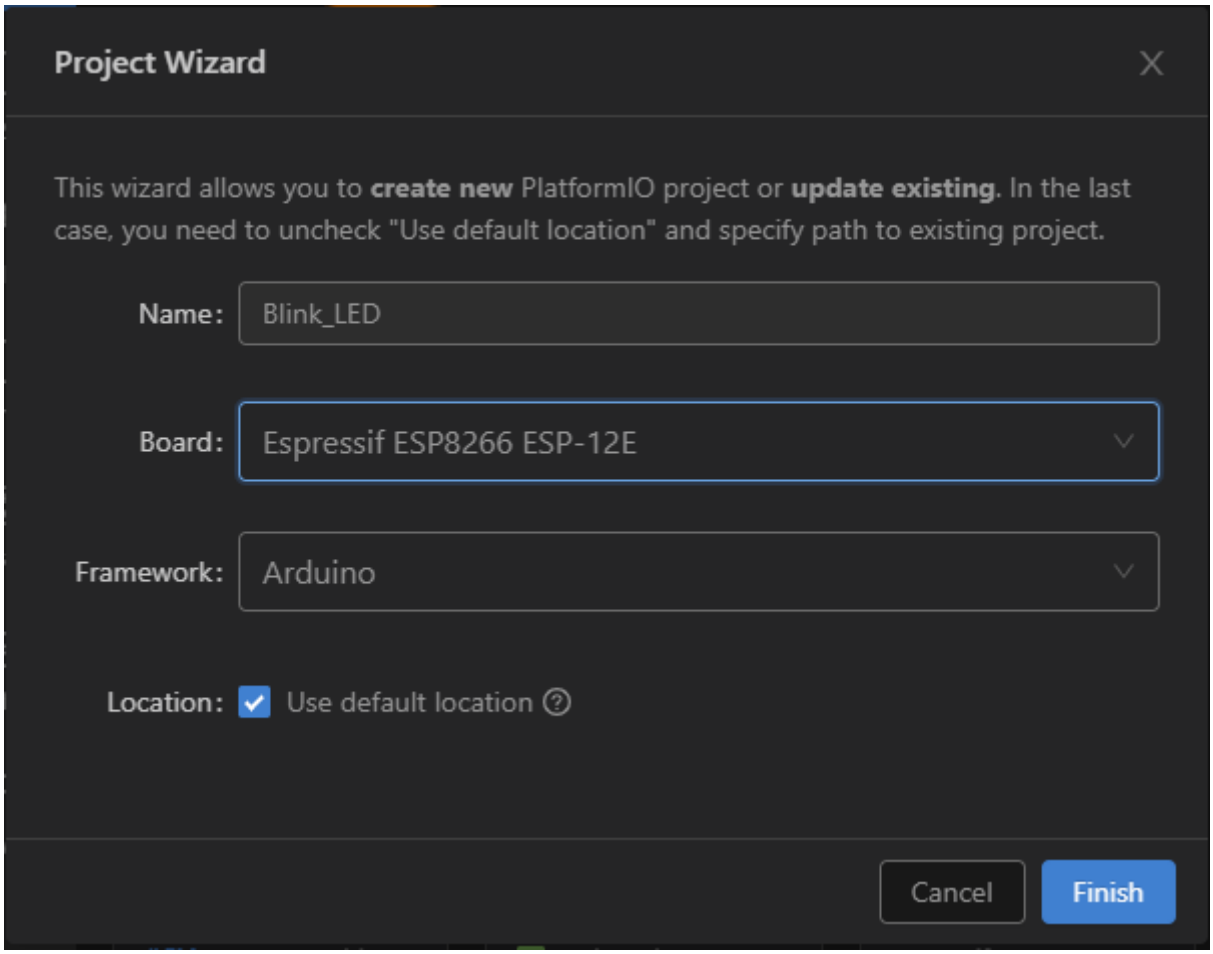
Give your project a name (for example *Blink_LED*) and select the board you're using. In our case, we're using the **DOIT ESP32 DEVKIT V1**. The Framework should be "**Arduino**" to use the Arduino core.

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:



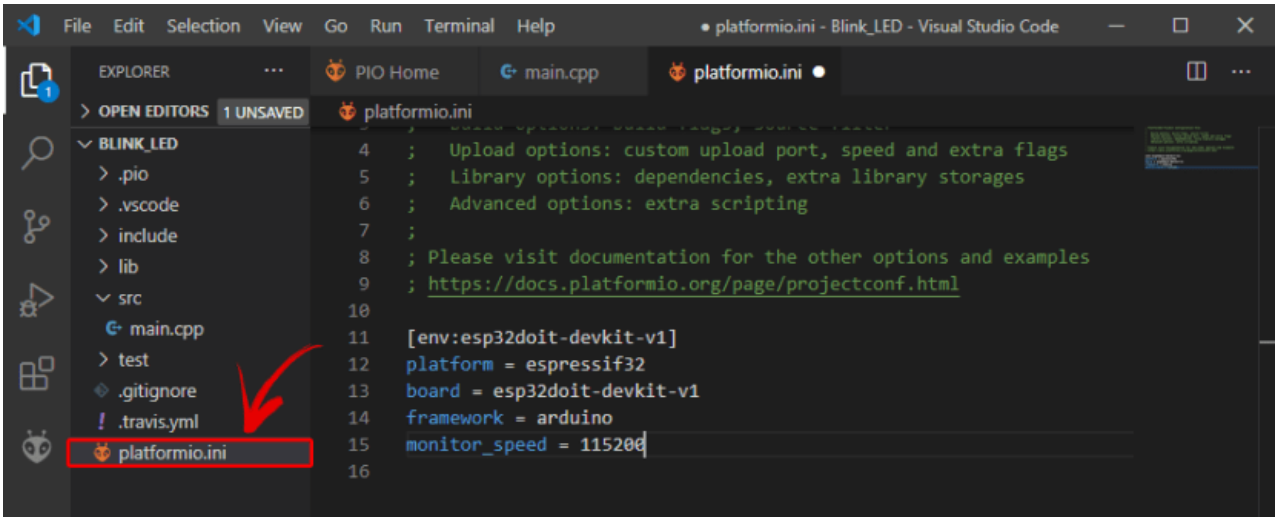
The Blink_LED project should be accessible from the Explorer tab.

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. Otherwise, you will no longer be able to compile your project using PlatformIO.

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.



- **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.

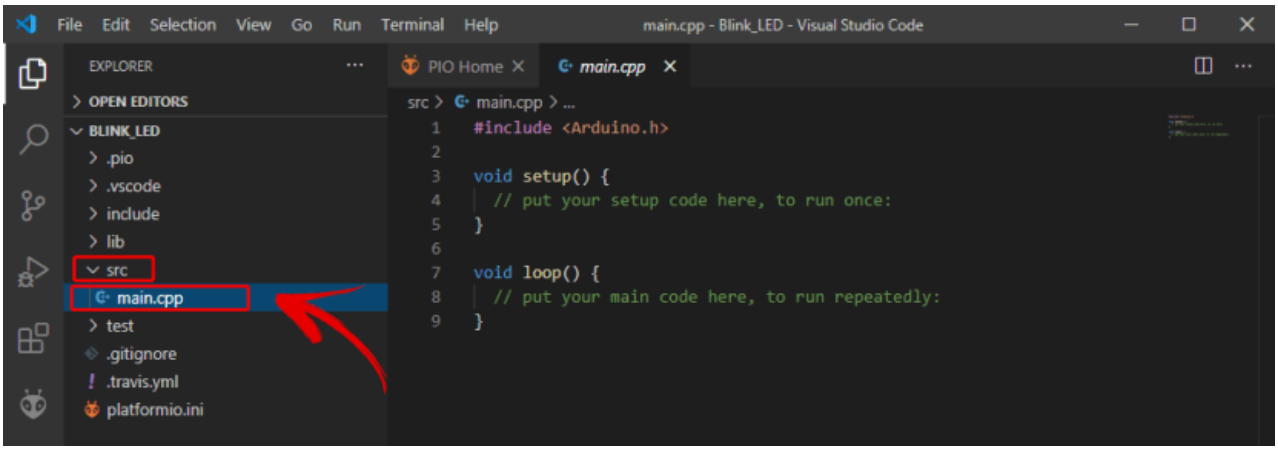
```
monitor_speed = 115200
```

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.



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-ide-esp32-esp8266-arduino/
*********/

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

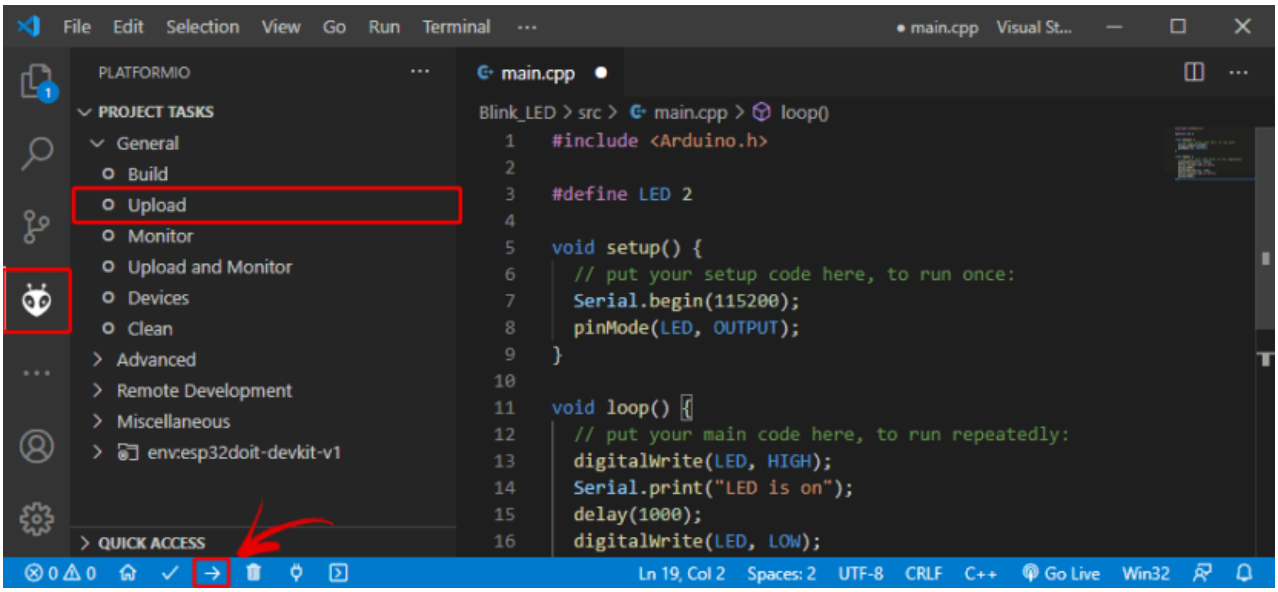
View raw code

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).

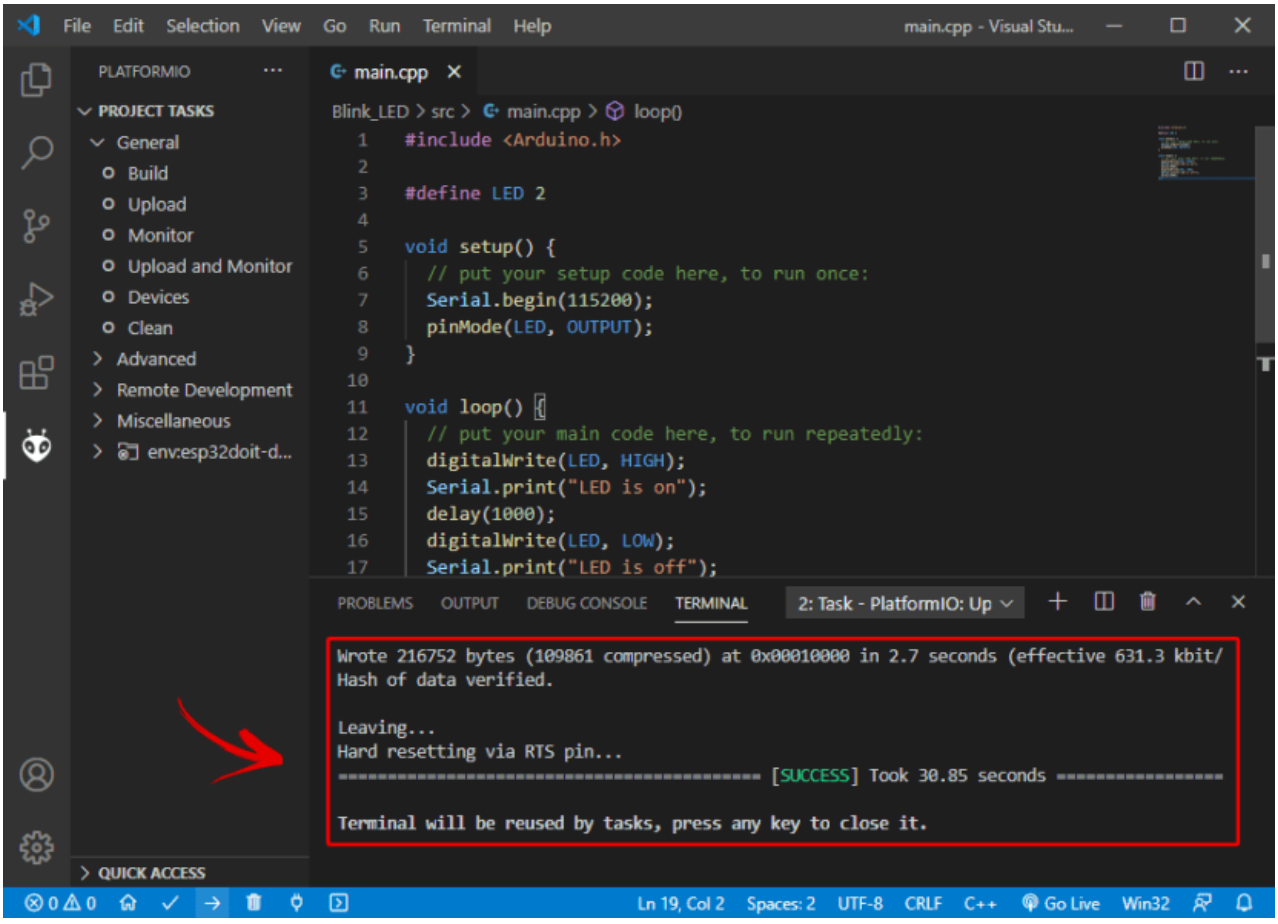
We recommend that you copy this code manually, so that you see the autocompletion and other interesting features of the IDE in action. Additionally, if you have a syntax error somewhere in your program, it will underline it in red even before compiling.

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

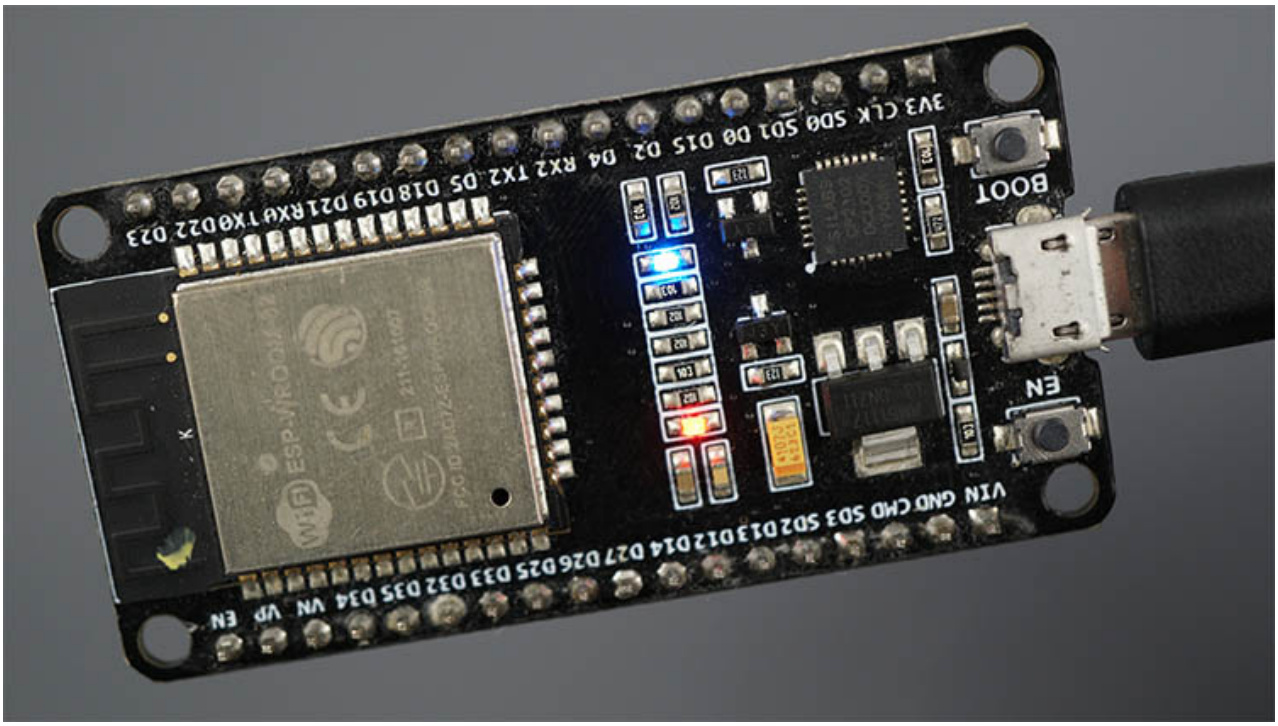
Now, you can click on the Upload icon to compile and upload the code. Alternatively, you can go to the PIO Project Tasks menu and select **Upload**.



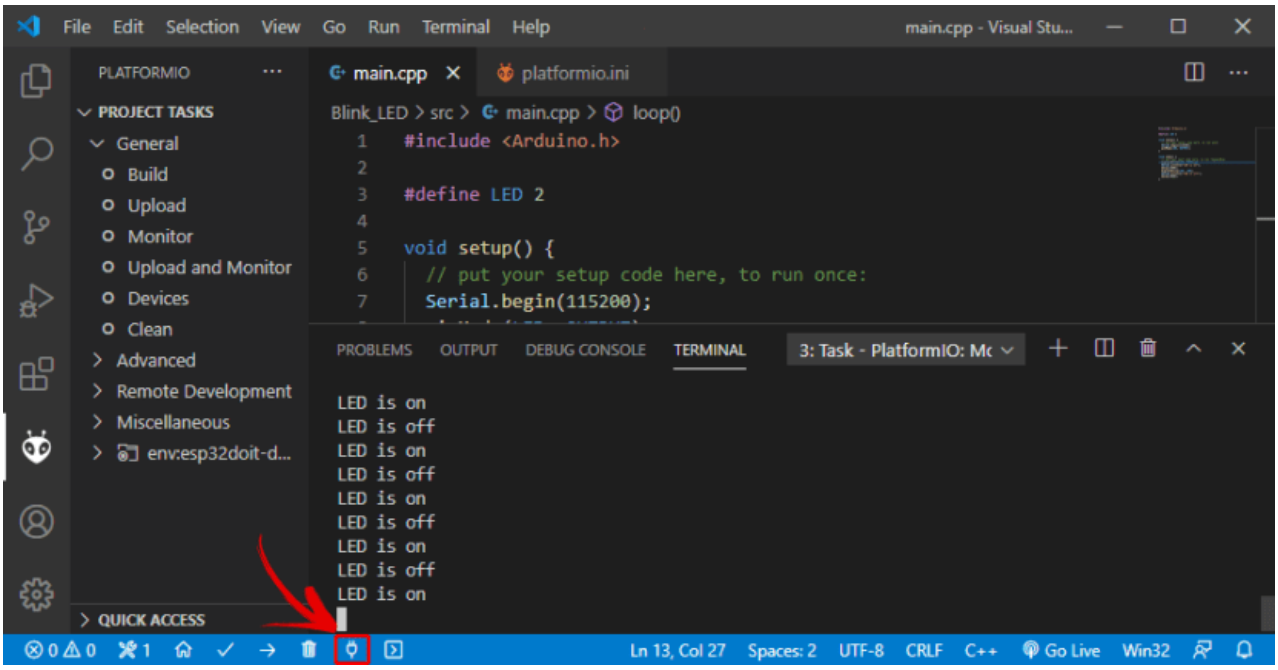
If the code is successfully uploaded, you should get the following message.



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



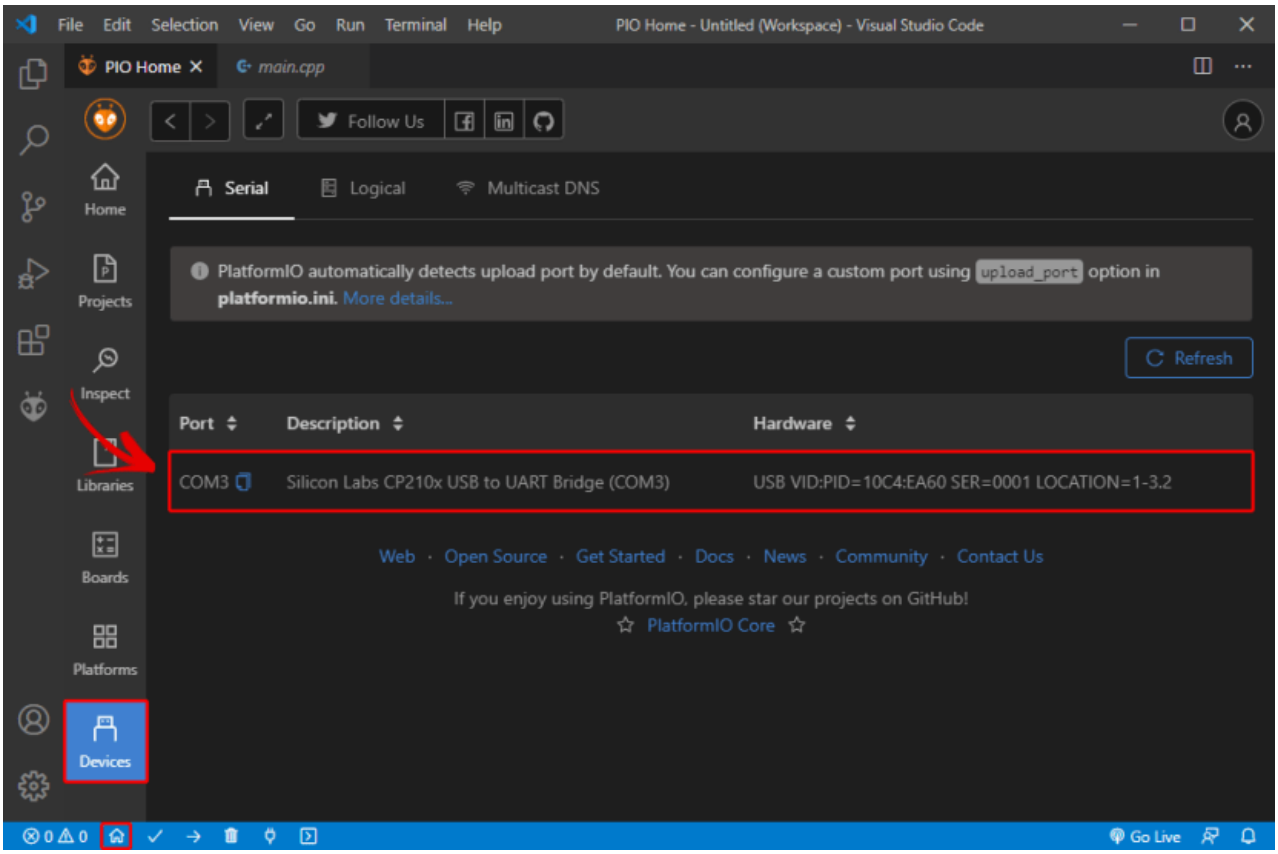
Now, click on the Serial Monitor icon and you should see it printing the current LED state.



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 por 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

2) If you get the error "COM Port not found/not available", you might need to install the CP210x Drivers:

- 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.

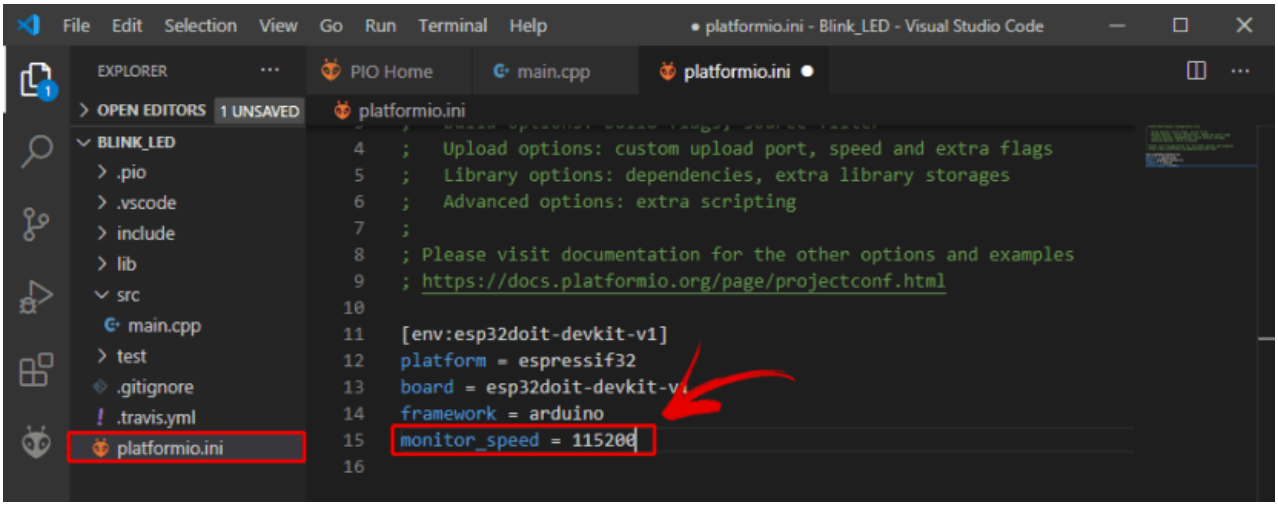
Changing the Serial Monitor Baud Rate – PlatformIO IDE

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
```



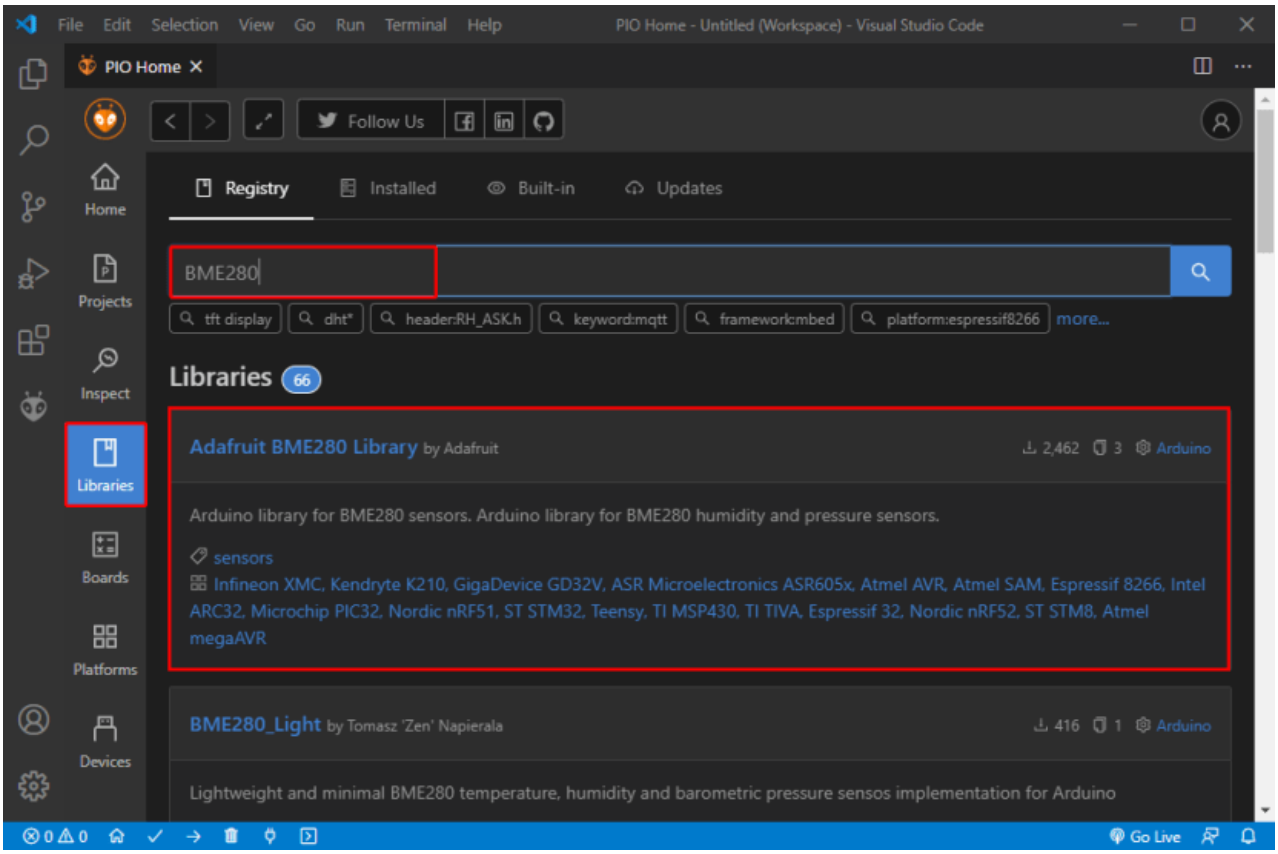
After that, save that file.

Installing ESP32/ESP8266 Libraries on PlatformIO IDE

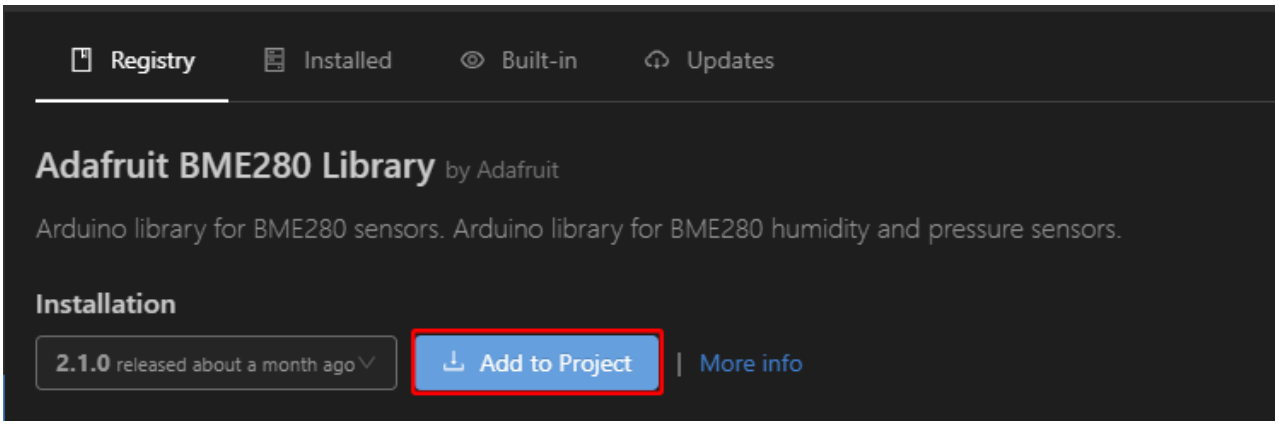
Follow the next procedure if you need to install libraries in PlatformIO IDE.

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

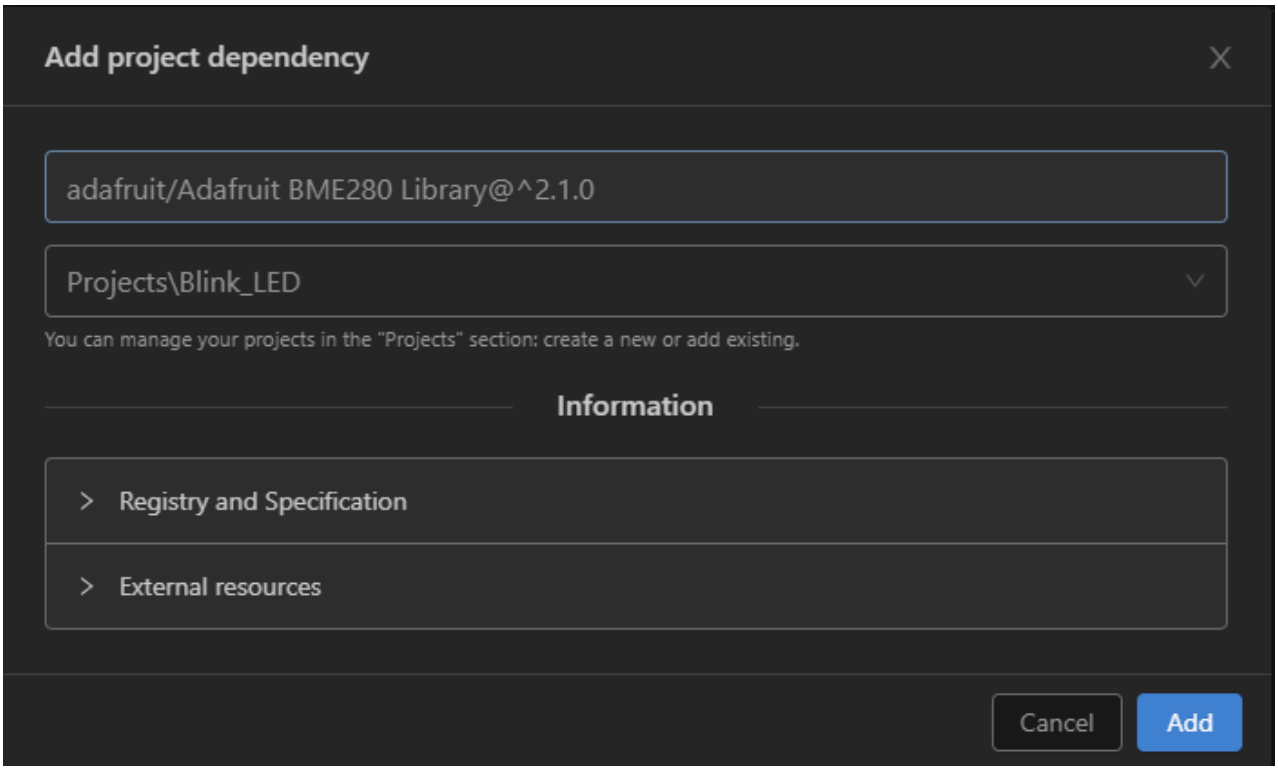
Search for the library you want to install. For example *Adafruit_BME280*.



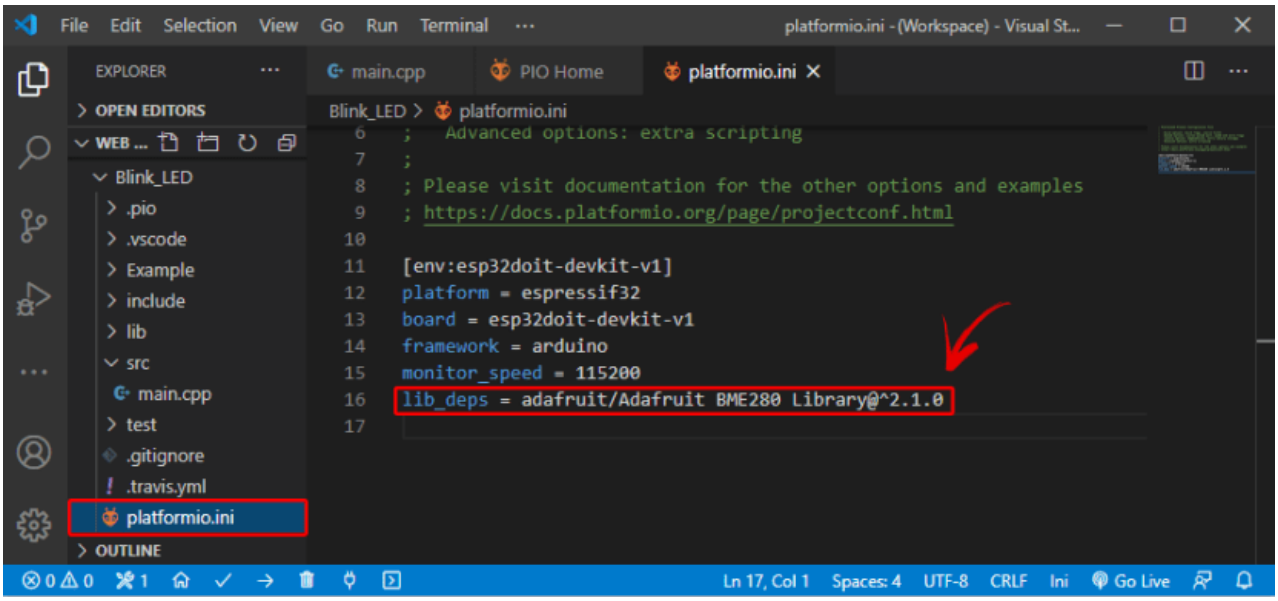
Click on the library you want to include in your project. Then, click **Add to Project**.



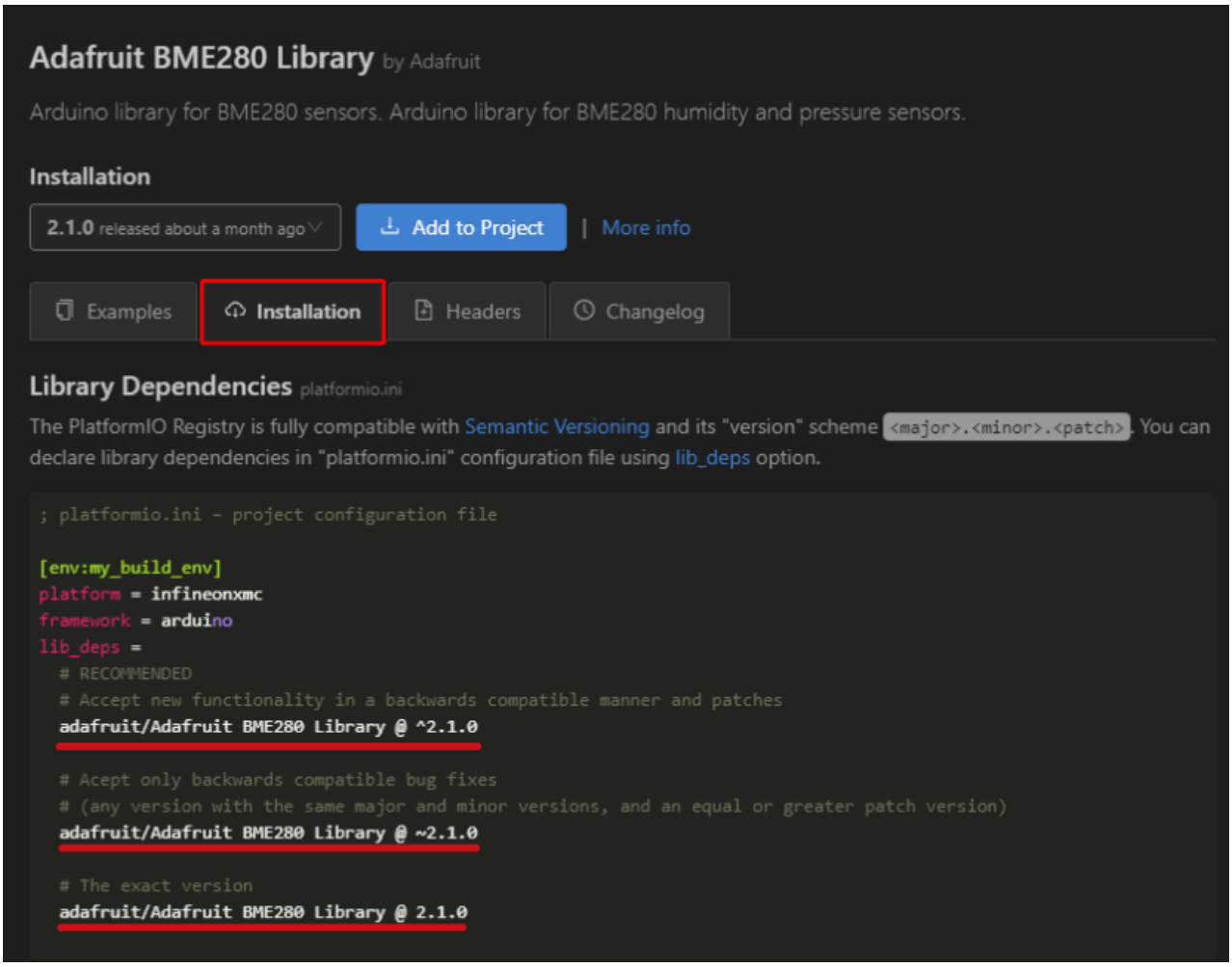
Select the project were you want to use the library.



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.



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.



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
```

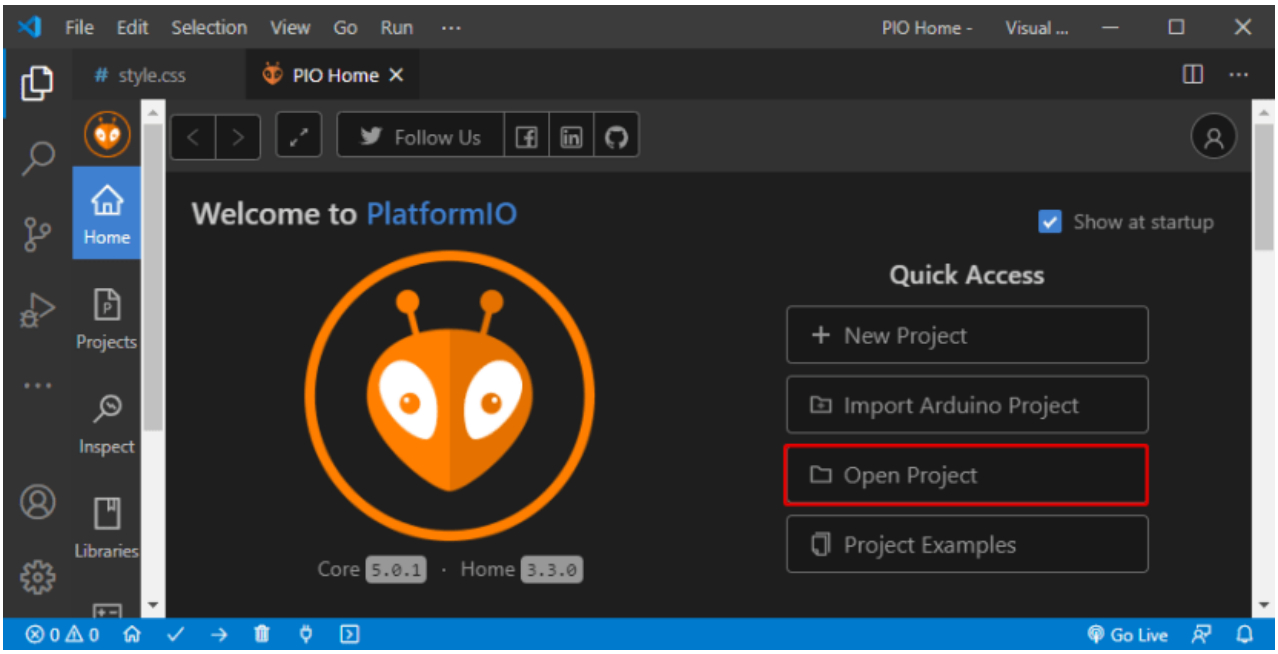
If you need multiple libraries, you can separate their name by a coma or put them on different lines. For example:

```
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 will tell PlatformIO to automatically download the library and all its dependencies when you save the configuration file or when you compile your project.

Open a Project Folder

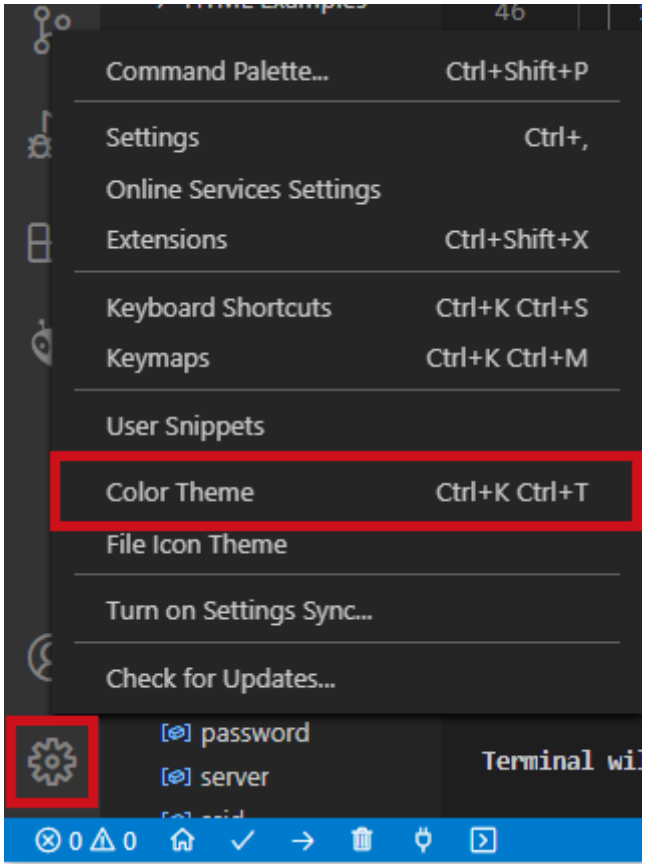
To open an existing project folder on PlatformIO, open VS Code, go to PlatformIO Home and click on **Open Project**. Navigate through the files and select your project folder.



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.