Contents

AI Voice Assistant Based on OPENAI Realtime model

This project applies the Media Kit to develop an AI voice assistant for conversations with OpenAI. It requires basic programming skills and some familiarity with OpenAI.

About the Project

This voice assistant project (https://github.com/Freenove/openai-realtime-embedded) is based on OpenAI's open-source project **openai-realtime-embedded** (https://github.com/openai/openai-realtime-embedded). It enables embedded devices to call OpenAI's Realtime Model APIs, such as GPT-4o Realtime and GPT-4o Mini Realtime, allowing you to build your own AI voice assistant.

Freenove has adapted this project for its Media Kit product. This article will guide you on how to run it on the Media Kit.

Cautions

- **Project Copyright**: The original author of this project is OpenAI. Freenove forked and adapted it for Media Kit, and the project is licensed under the MIT License.
- **Supported Countries/Regions**: This project uses OpenAI's GPT-4o RealTime API, which is not available in all countries/regions. Please check OpenAI's supported countries list: https://platform.openai.com/docs/supported-countries.
  If you cannot access this link, OpenAI's services are likely unavailable in your location.
- **Supported Languages**: OpenAI supports many languages but has no official list. For reference, see: https://platform.openai.com/docs/api-reference/realtime-sessions/create
  https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes)
- **Pricing**: The GPT-4o RealTime API is a paid service, and you must purchase credits from OpenAI to use it.
- **Seeking Help:** If you have followed the tutorial but still encounter issues, contact us at support@freenove.com
  Note: Since both the project and API are provided by OpenAI, if OpenAI discontinues them, we will also remove related documentation, tutorials, and code.

About OpenAI

OpenAI is a leading artificial intelligence research and deployment company committed to ensuring that artificial general intelligence (AGI) benefits all of humanity. Guided by principles of openness, collaboration, and safety, the organization drives the development and practical application of cutting-edge models. These include the renowned GPT series of language models, the DALL·E image generation model, as well as speech recognition and synthesis tools like Whisper.

OpenAI not only provides powerful AI models but also offers API services that enable developers to easily integrate these models into their own products. Recently, OpenAI launched its new Realtime API, which significantly enhances conversational naturalness through **direct streaming of audio input and output**, with automatic interruption handling capabilities.

However, it's important to note that:

1. **OpenAI currently does not offer free services**
2. The Realtime API currently only supports:
   GPT-4o and GPT-4o mini models
   The latest transcription models: GPT-4o Transcribe and GPT-4o mini Transcribe

For more information about Realtime API, please refer to Realtime API - OpenAI API

OpenAI-Realtime-Embedded Disclaimer

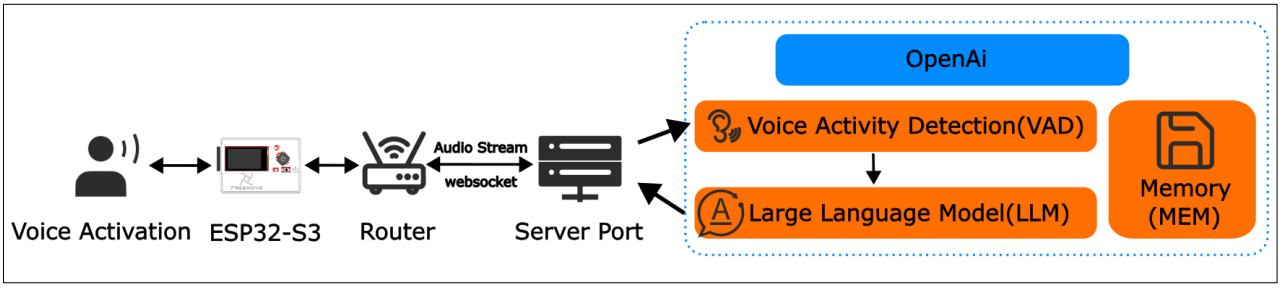
This implementation is an adaptation of the open-source project available at https://github.com/openai/openai-realtime-embedded, provided for third-party learning and AI functionality testing purposes, without any promotion or support for commercial applications. This tutorial is intended solely for personal learning and development by technology enthusiasts.

Notes:

1. As this is a third-party open-source project, if you encounter issues during your learning process, please submit an issue to the original repository https://github.com/openai/openai-realtime-embedded/issues
2. The OpenAI API is a paid service. You must subscribe and enable billing to access any API functionality. Without an active paid account, all OpenAI API features will be unavailable.
3. To use OpenAI services, you must apply for your own API key directly from OpenAI. We do not provide or share any API key information.
   For details about OpenAI API access, please visit:
   https://platform.openai.com/docs/api-reference/introduction

Please see the following flowchart.

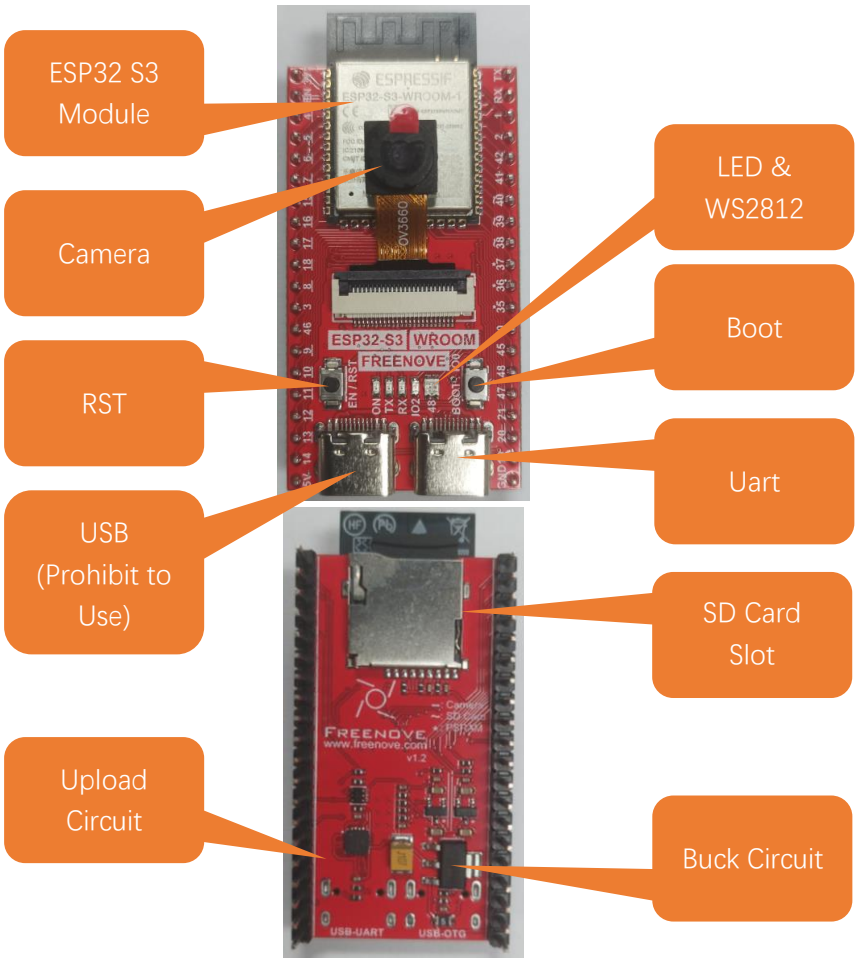
- The ESP32-S3 executes the openai-realtime-embedded source code, establishes a WiFi connection, and streams audio data to OpenAI's servers.
- Upon receiving the audio stream, OpenAI's servers process the data, generate text responses, and synthesize corresponding audio, which is then transmitted back to the ESP32-S3 via WiFi.
- In this project, the ESP32-S3 communicates with OpenAI's servers using the WebSocket protocol.



ESP32 S3 Hardware Specifications

Freenove ESP32-S3 WROOM Board

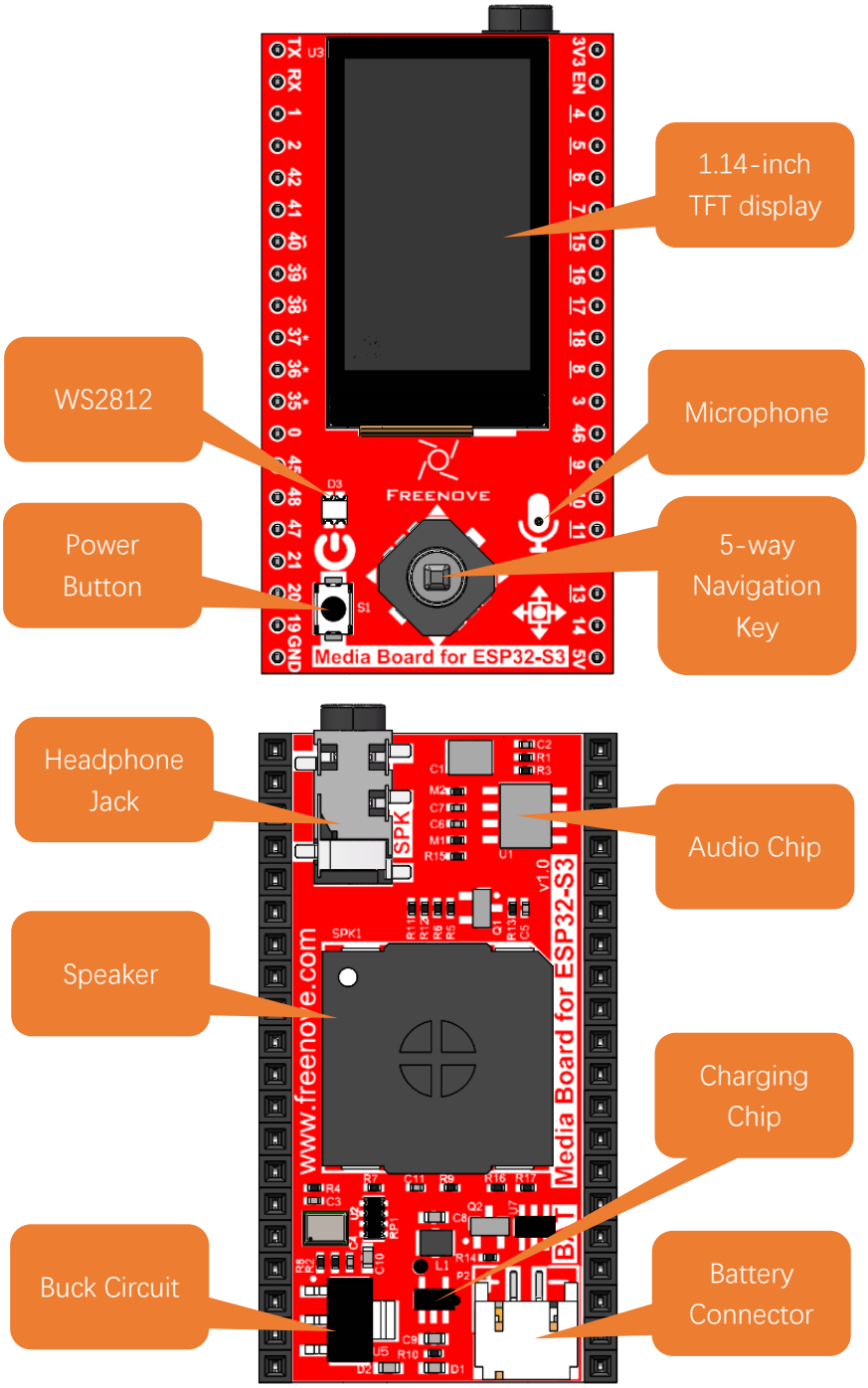
We use the Freenove ESP32-S3 Board as the main control board, which integrates 16Mb Flash and 8Mb PSRAM.



For more information about Freenove ESP32-S3 Board, please refer to
https://github.com/Freenove/Freenove_Ultimate_Starter_Kit_for_ESP32_S3

Audio Circuit

The audio circuit board integrates multiple functional hardware modules, including a 1.14-inch TFT display, a microphone, a battery charging circuit, an audio processing circuit, a headphone jack, and an integrated speaker, as illustrated in the figures below.

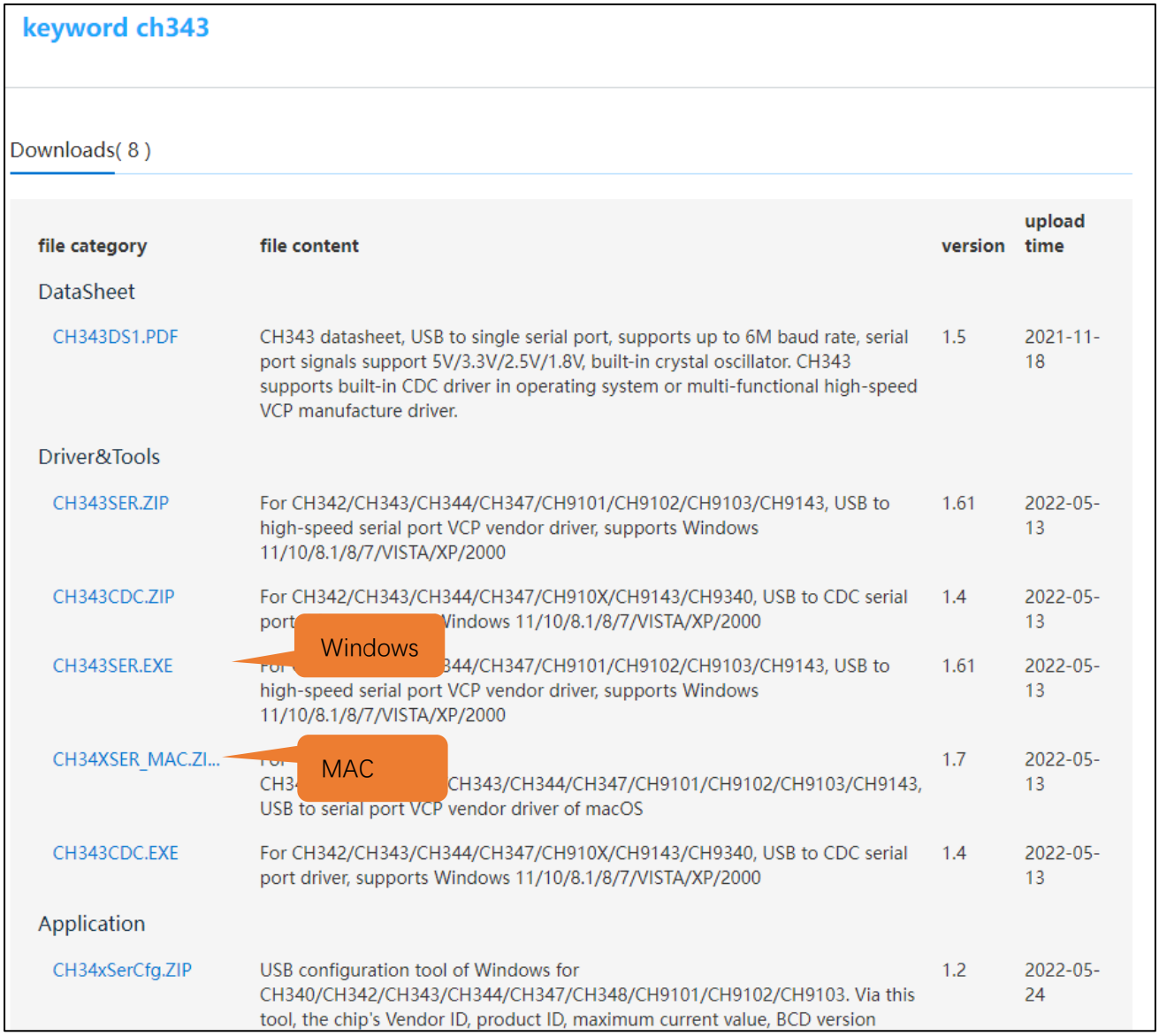


Install CH343 Driver (Required)

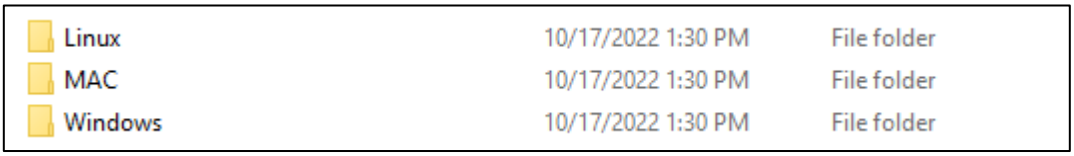
ESP32-S3 WROOM uses CH343 to download codes. So before using it, we need to install CH343 driver in our computers.

Windows

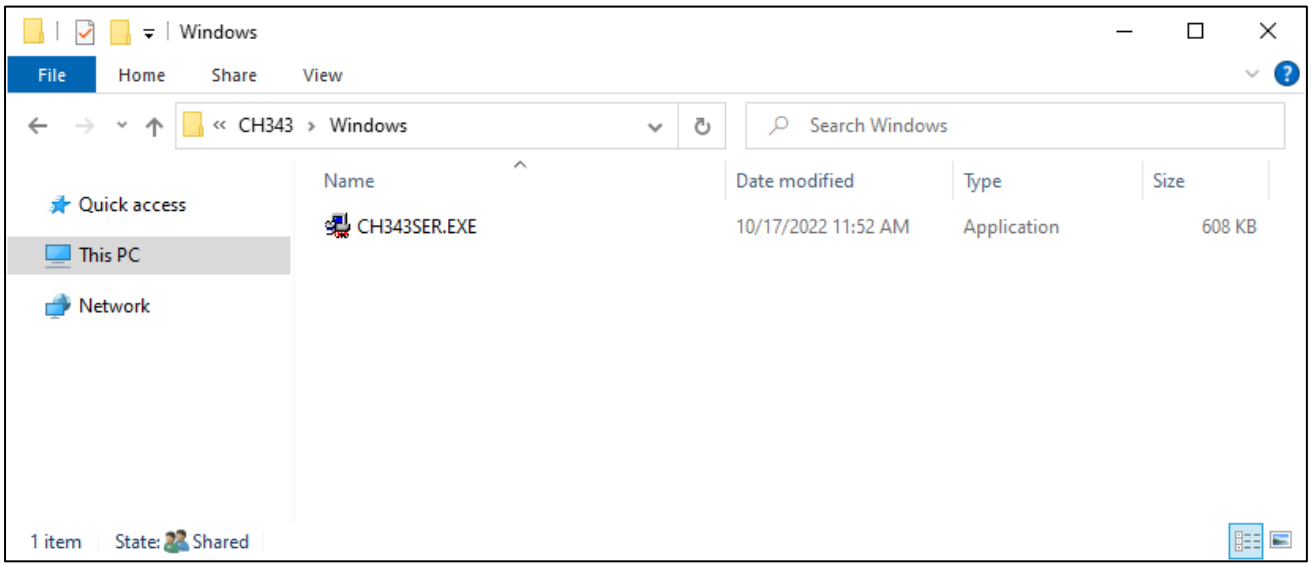
1. First, download CH343 driver, click https://www.wch-ic.com/products/CH343.html? to download the appropriate one based on your operating system.



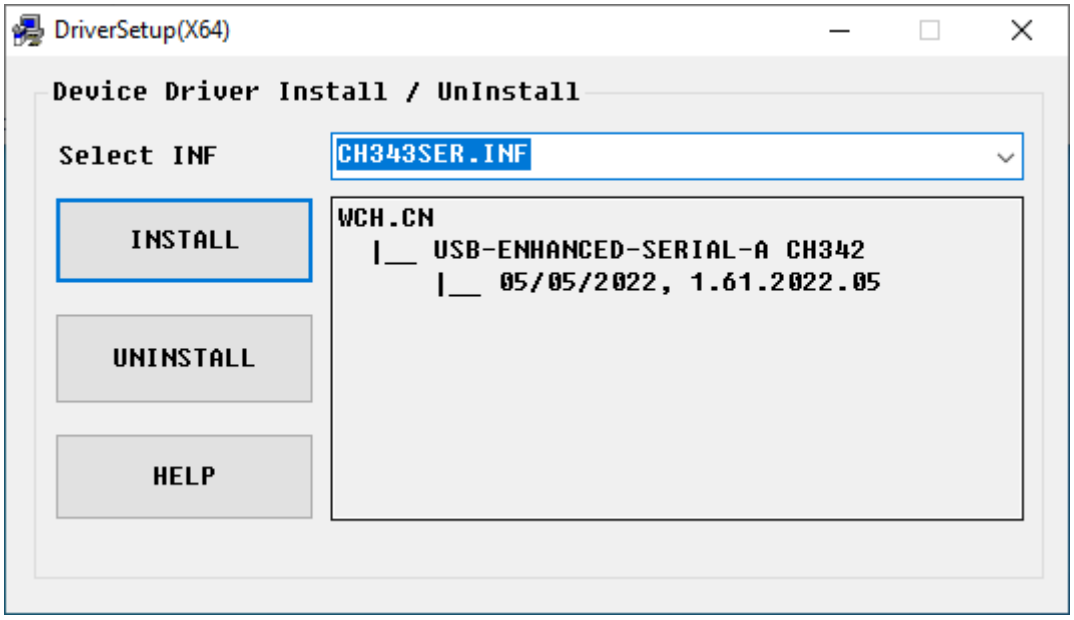
If you would not like to download the installation package, you can open **"Freenove_Media_Kit_for_ESP32-S3/CH343"**, we have prepared the installation package.



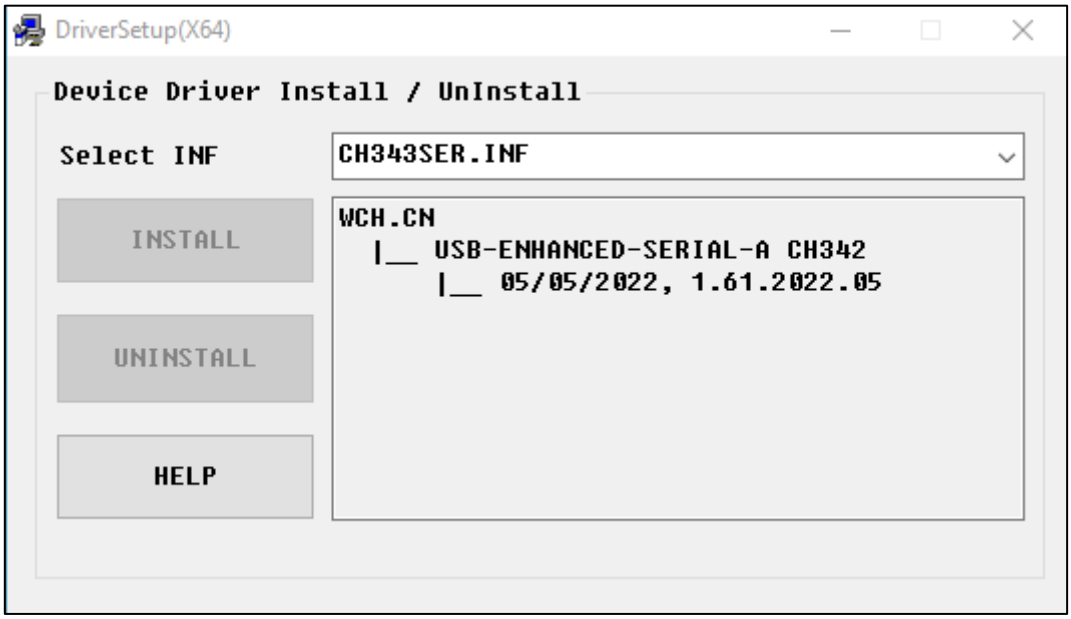
2. Open the folder "**Freenove_Media_Kit_for_ESP32-S3/CH343/Windows/**"



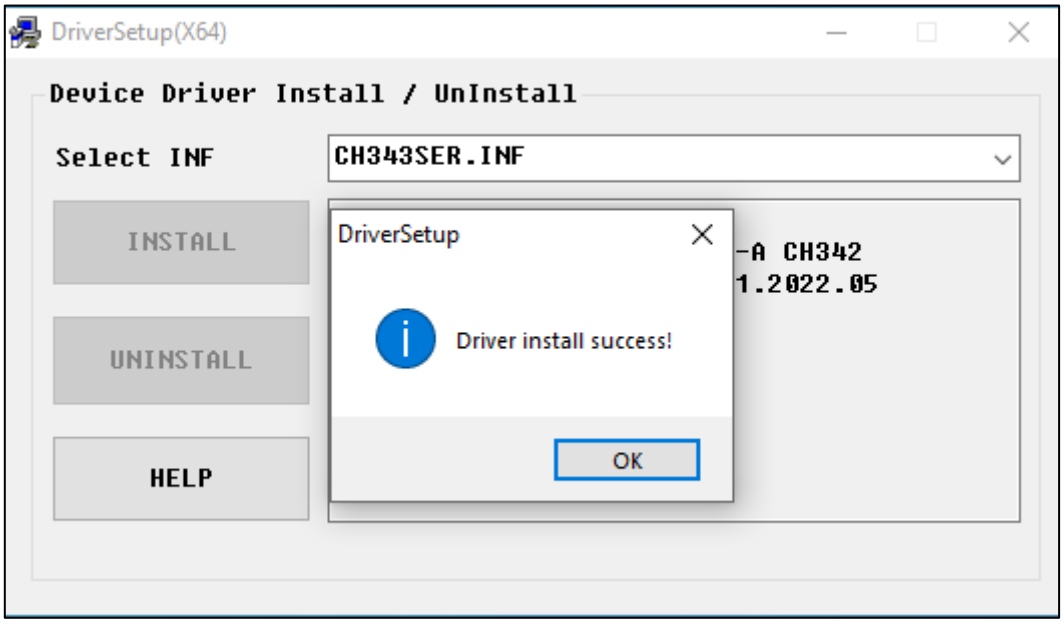
3. Double click "**CH343SER.EXE**".



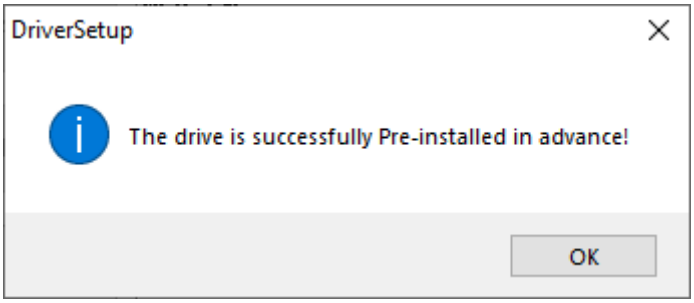
4. Click “INSTALL” and wait for the installation to complete.



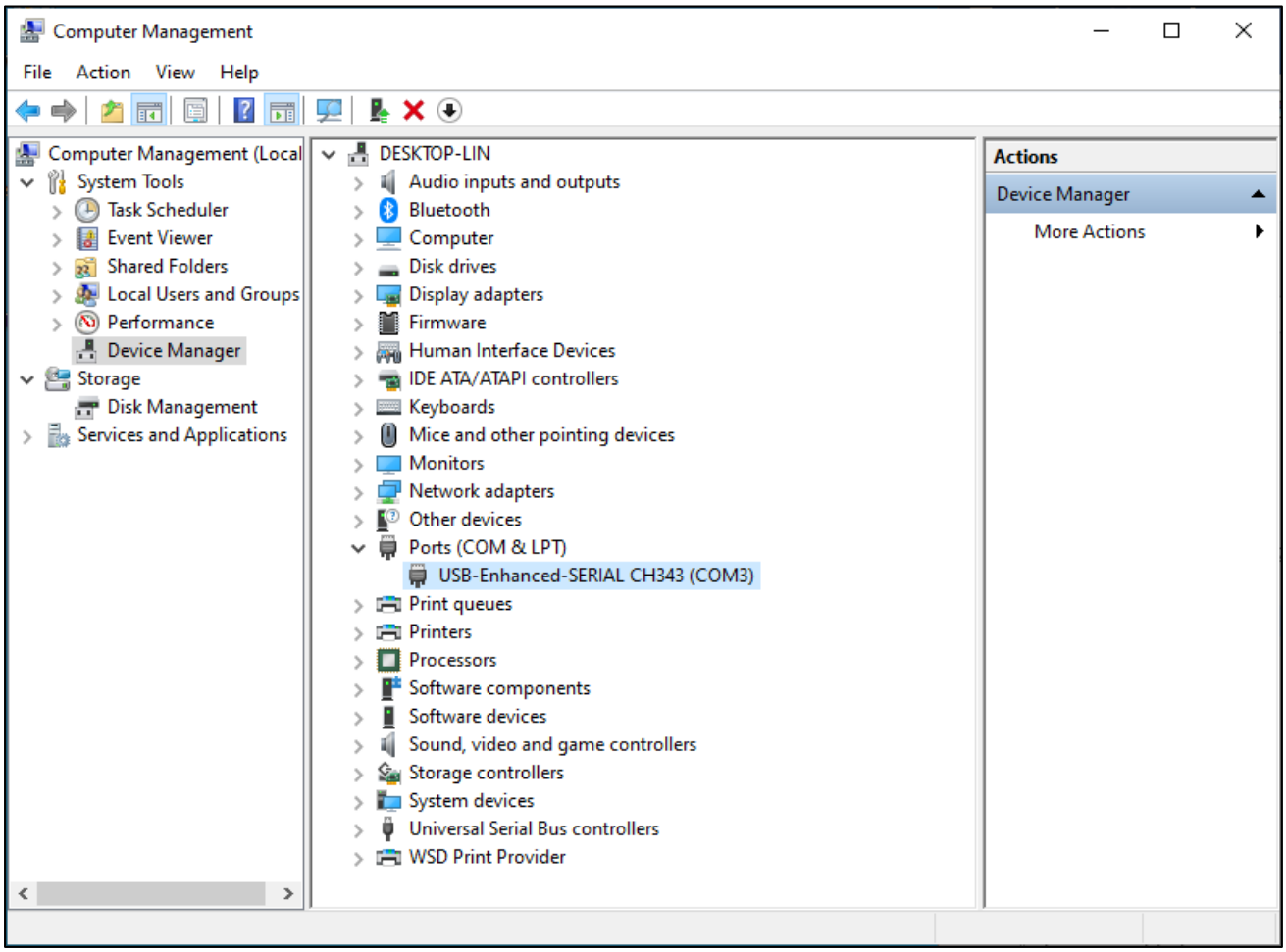
5. If the driver is successfully installed, you should see the following prompt.



Note: If you see “The drive is successfully Pre-installed in advance”, it indicates the installation fails. Please make sure you use the USB data cable, not a charging cable.



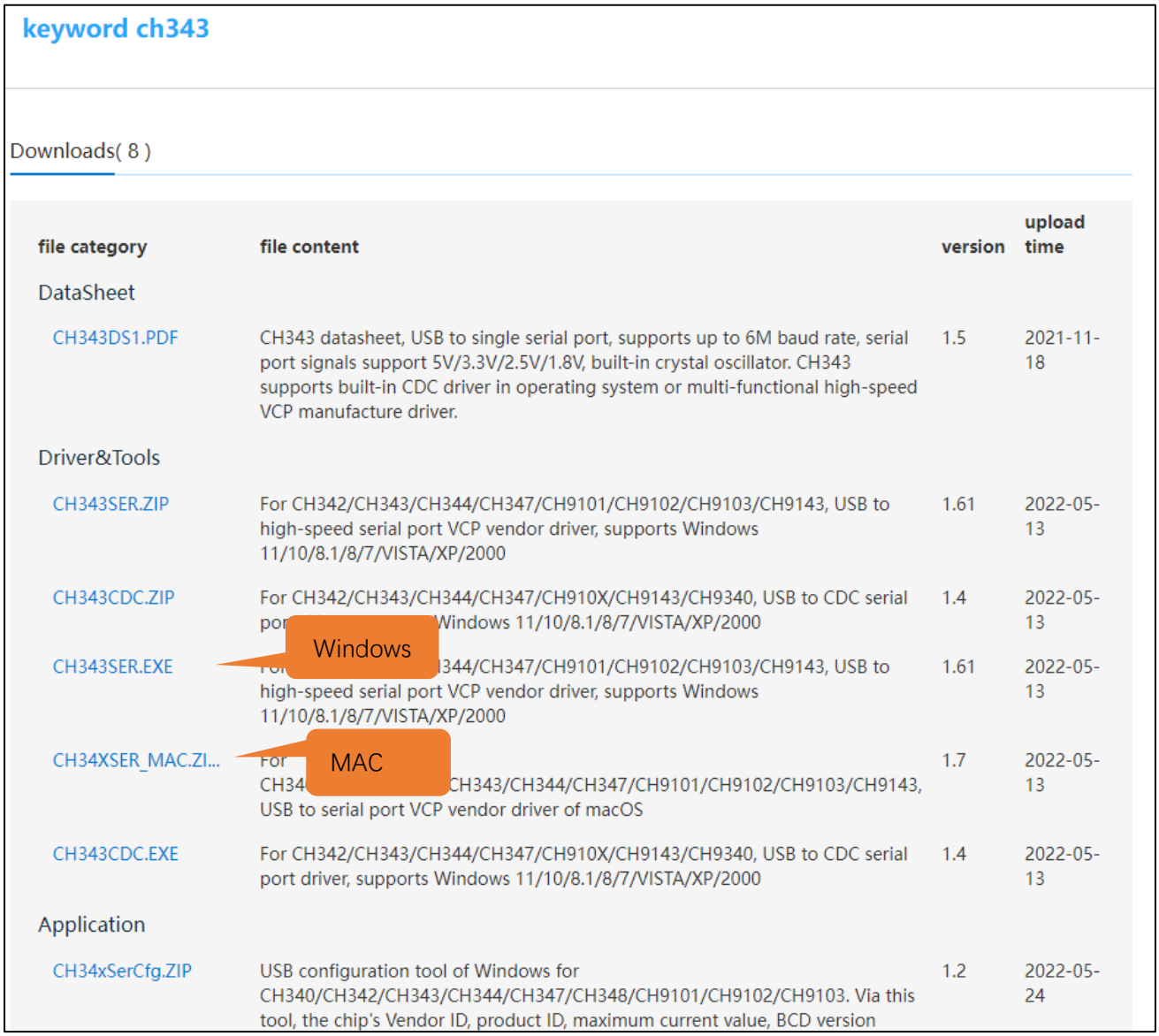
6. When ESP32-S3 WROOM is connected to computer, select “This PC”, right-click to select “Manage” and click “Device Manager” in the newly pop-up dialog box, and you can see the following interface.



7. So far, CH343 has been installed successfully. Close all dialog boxes.

MAC

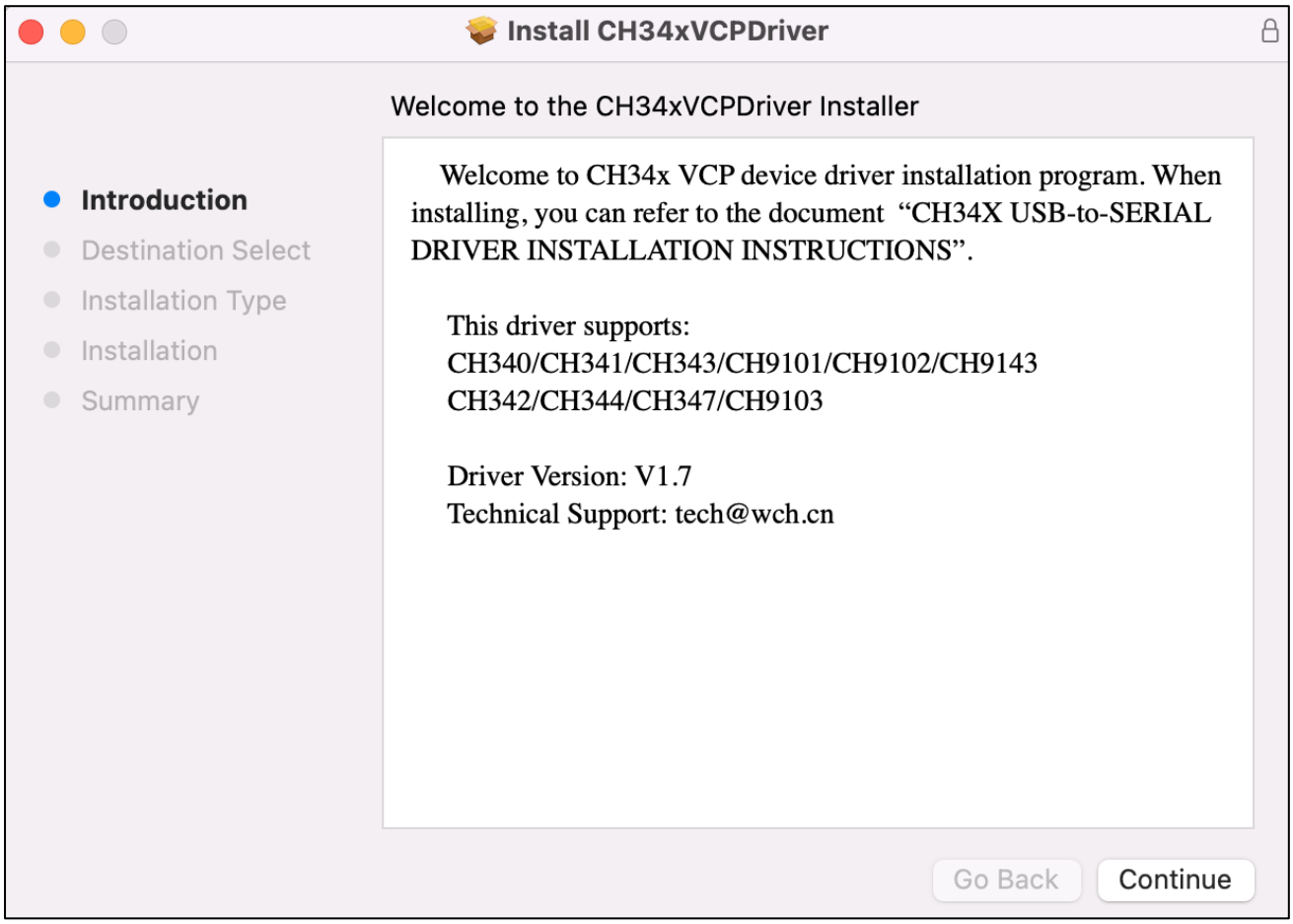
First, download CH343 driver, click http://www.wch-ic.com/search?t=all&q=ch343 to download the appropriate one based on your operating system.



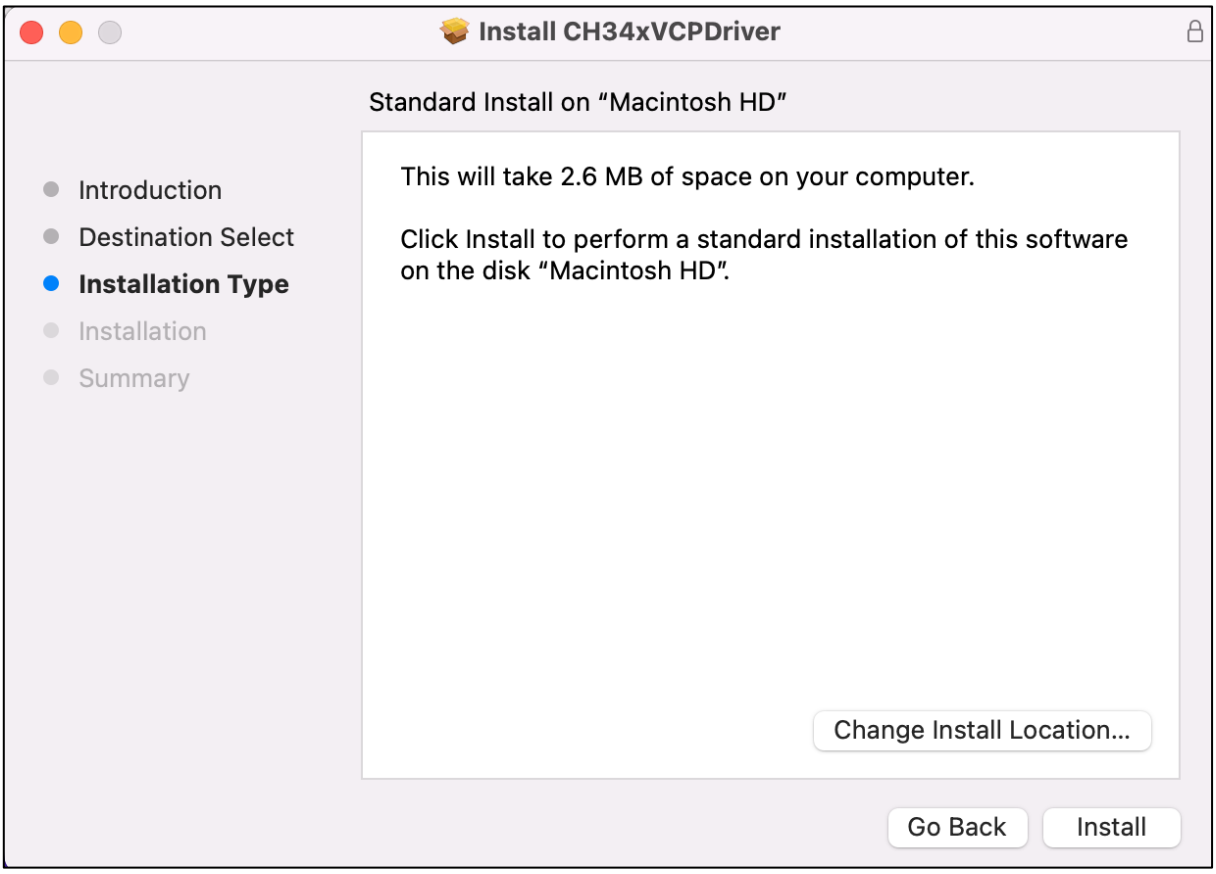
If you would not like to download the installation package, you can open **"Freenove_Media_Kit_for_ESP32-S3/CH343"**, we have prepared the installation package.

Second, open the folder "**Freenove_Media_Kit_for_ESP32-S3/CH343/MAC/**"

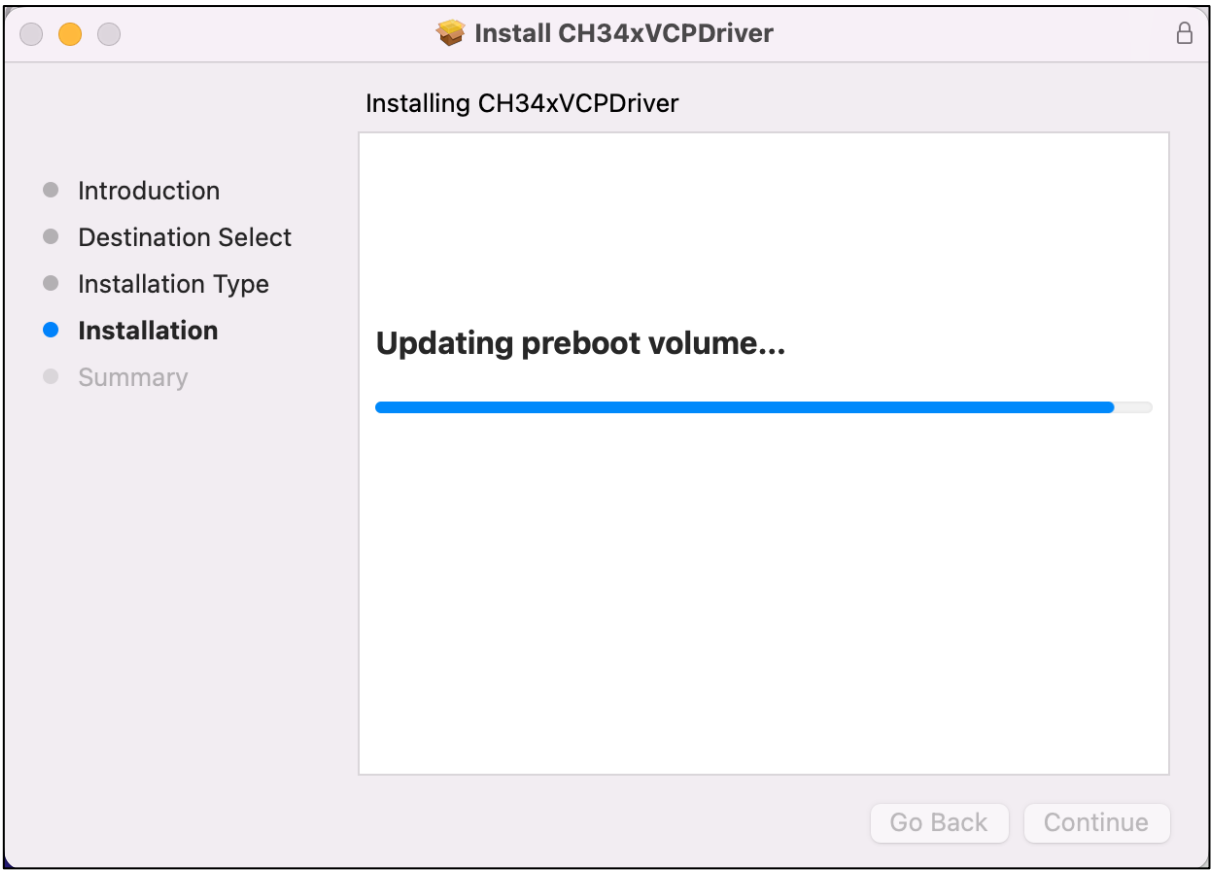
Third, click Continue.



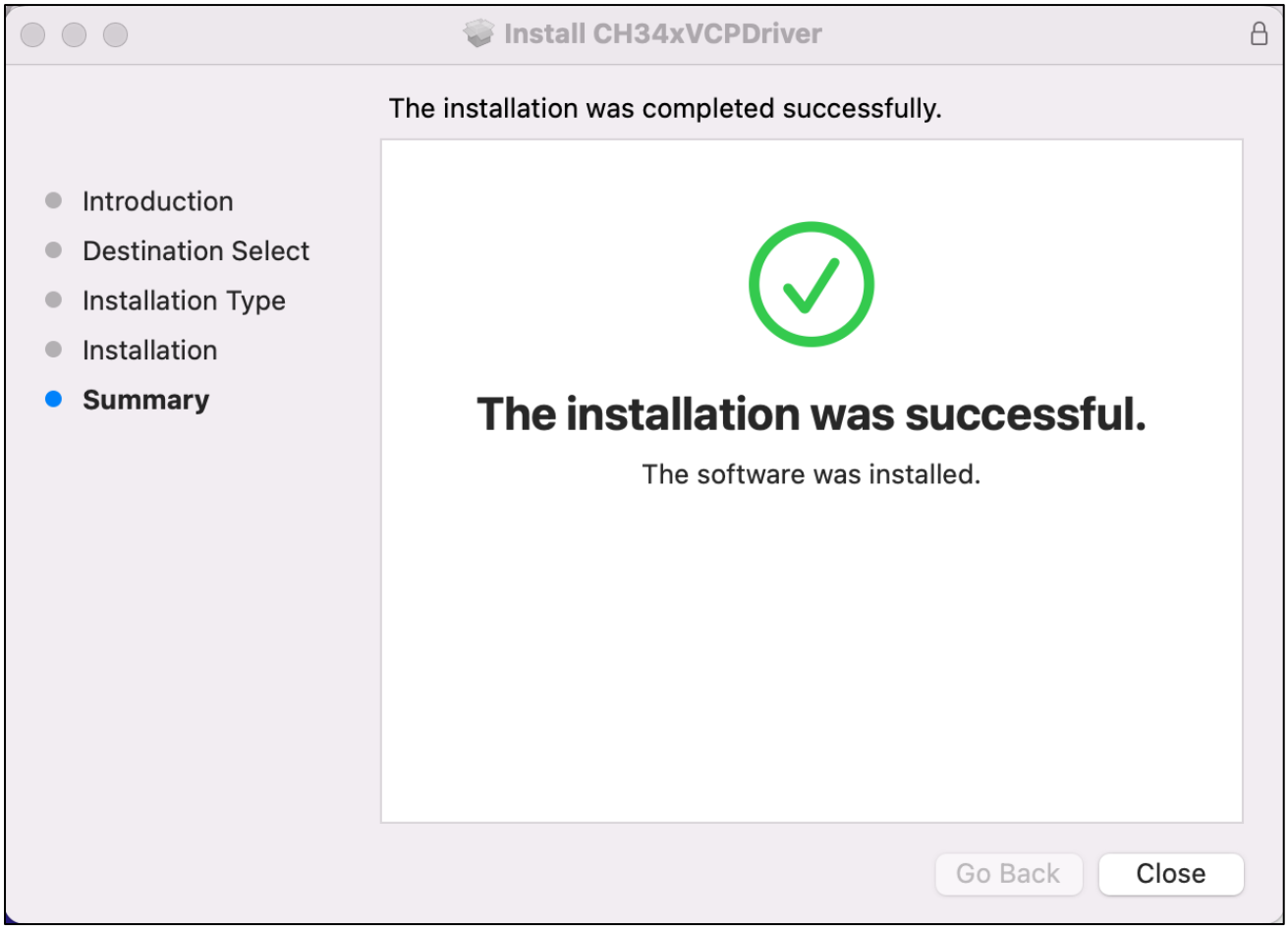
Fourth, click Install.



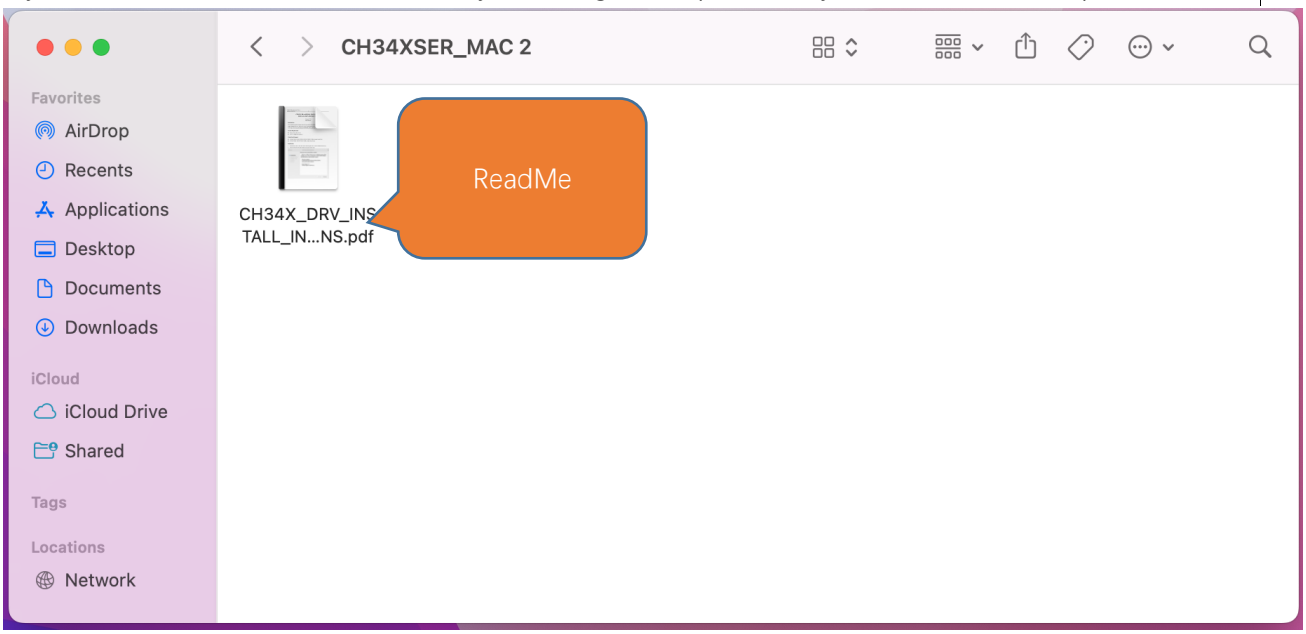
Then, waiting Finish.



Finally, restart your PC.

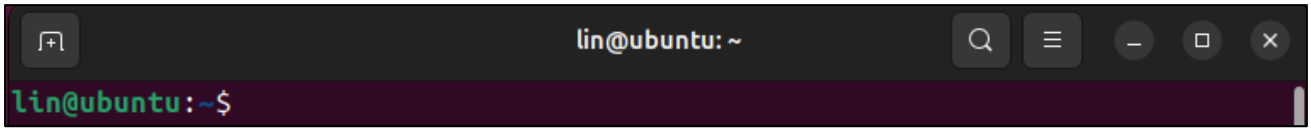


If you still haven't installed the CH340 by following the steps above, you can view readme.pdf to install it.



Linux

Here we take Ubuntu as an example. Open the terminal in Linux system.



Check the port with the command “lsusb”.

```
lsusb
ls /dev/tty*
```

```
lin@lin-virtual-machine:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 004: ID 1a86:55d3 QinHeng Electronics USB Single Serial
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

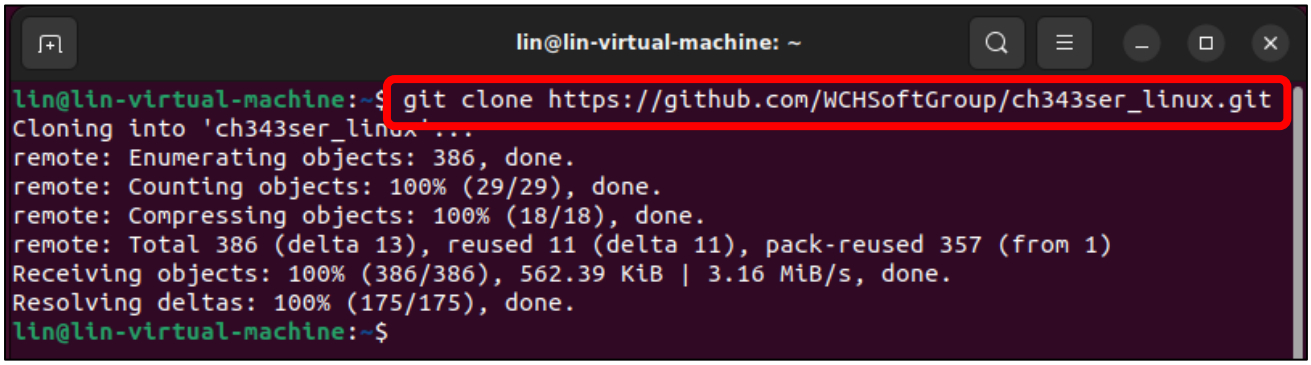
```
lin@lin-virtual-machine:~$ ls /dev/tty*
/dev/tty    /dev/tty23  /dev/tty39  /dev/tty54      /dev/ttyS1   /dev/ttyS25
/dev/tty0   /dev/tty24  /dev/tty4   /dev/tty55      /dev/ttyS10  /dev/ttyS26
/dev/tty1   /dev/tty25  /dev/tty40  /dev/tty56      /dev/ttyS11  /dev/ttyS27
/dev/tty10  /dev/tty26  /dev/tty41  /dev/tty57      /dev/ttyS12  /dev/ttyS28
/dev/tty11  /dev/tty27  /dev/tty42  /dev/tty58      /dev/ttyS13  /dev/ttyS29
/dev/tty12  /dev/tty28  /dev/tty43  /dev/tty59      /dev/ttyS14  /dev/ttyS3
/dev/tty13  /dev/tty29  /dev/tty44  /dev/tty6       /dev/ttyS15  /dev/ttyS30
/dev/tty14  /dev/tty3   /dev/tty45  /dev/tty60      /dev/ttyS16  /dev/ttyS31
/dev/tty15  /dev/tty30  /dev/tty46  /dev/tty61      /dev/ttyS17  /dev/ttyS4
/dev/tty16  /dev/tty31  /dev/tty47  /dev/tty62      /dev/ttyS18  /dev/ttyS5
/dev/tty17  /dev/tty32  /dev/tty48  /dev/tty63      /dev/ttyS19  /dev/ttyS6
/dev/tty18  /dev/tty33  /dev/tty49  /dev/tty7       /dev/ttyS2   /dev/ttyS7
/dev/tty19  /dev/tty34  /dev/tty5   /dev/tty8       /dev/ttyS20  /dev/ttyS8
/dev/tty2   /dev/tty35  /dev/tty50  /dev/tty9       /dev/ttyS21  /dev/ttyS9
/dev/tty20  /dev/tty36  /dev/tty51  /dev/ttyACM0    /dev/ttyS22
/dev/tty21  /dev/tty37  /dev/tty52  /dev/ttyprintk  /dev/ttyS23
/dev/tty22  /dev/tty38  /dev/tty53  /dev/ttyS0      /dev/ttyS24
lin@lin-virtual-machine:~$
```

Generally, the CH34x driver is included in modern Linux kernels, so it should work automatically when the device is connected

If your computer does not have the CH343 driver, you can follow the steps below to install it. If your computer recognizes the CH343 driver, you may skip the following steps.

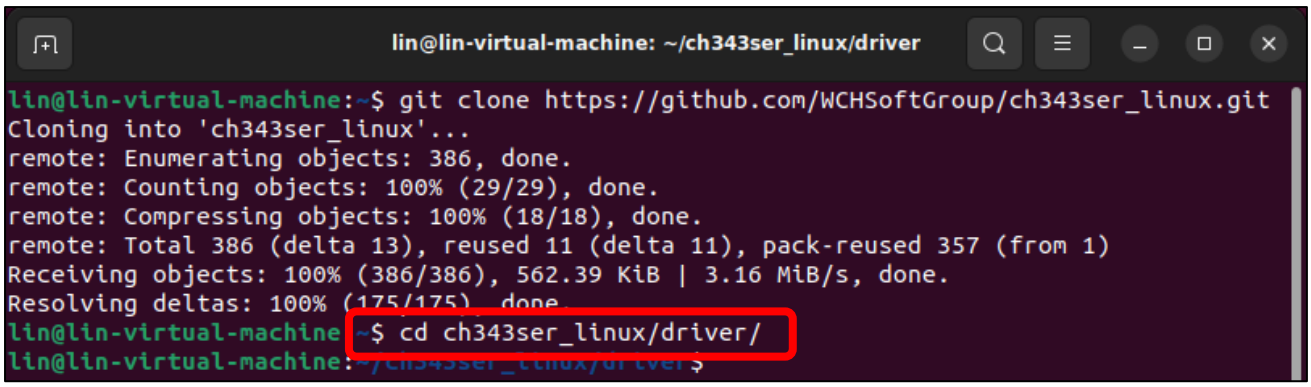
Run the following command to download the driver.

```
git clone https://github.com/WCHSoftGroup/ch343ser_linux.git
```



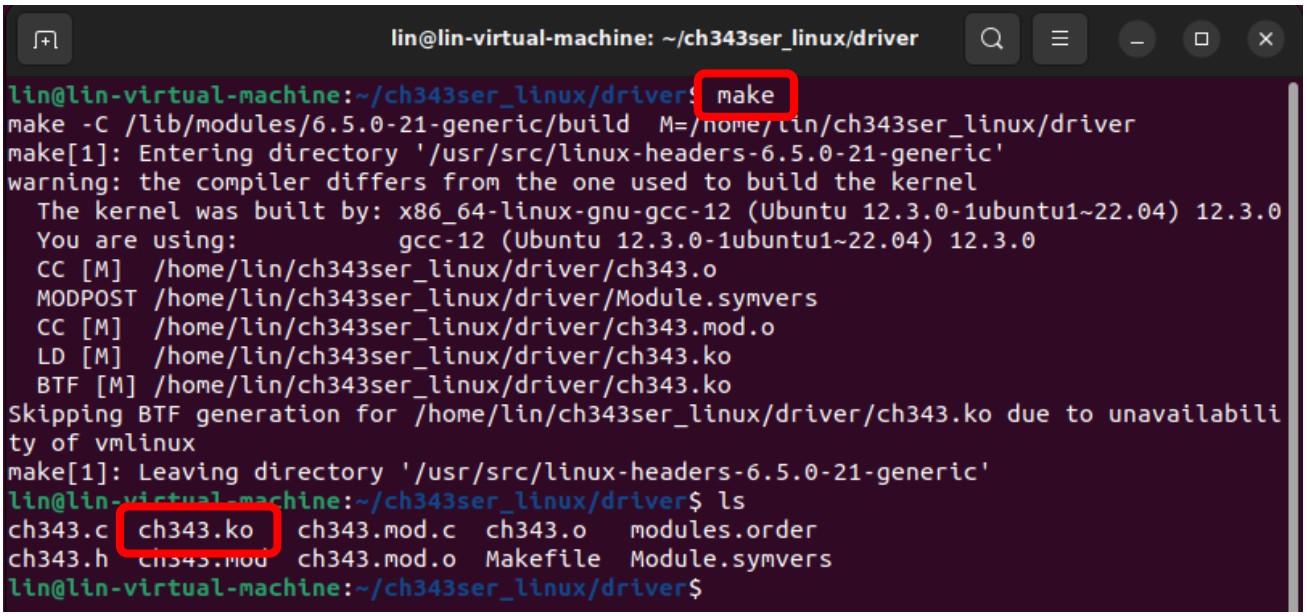
Enter the folder where the driver locates.

```
cd ch343ser_linux/driver/
```



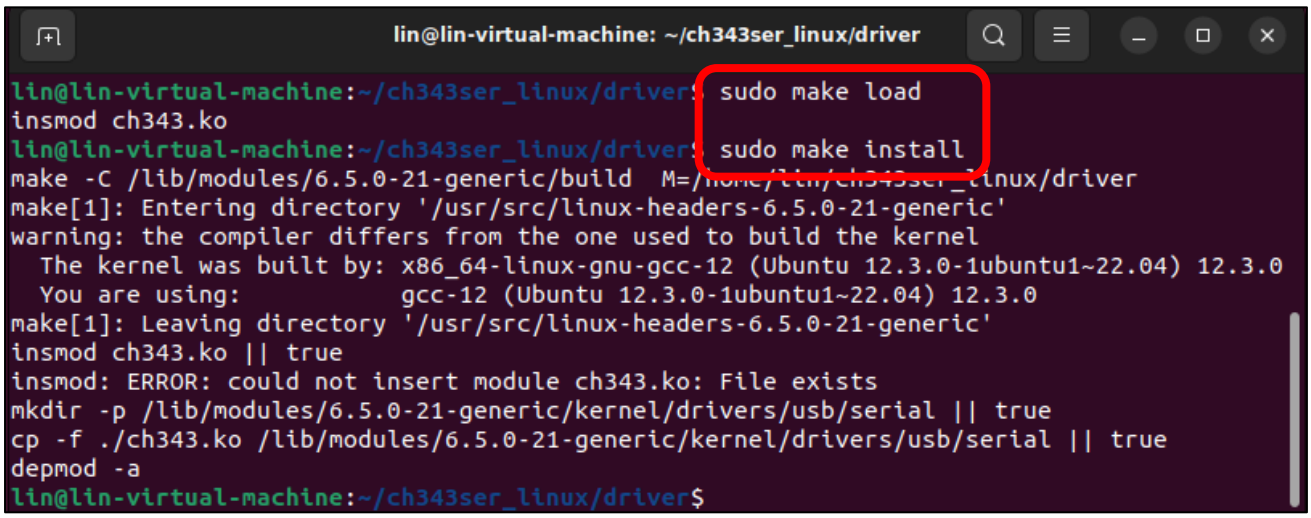
Compile to generate a ch343.ko file.

```
make
```



Install the ch343 chip driver.

```
sudo make load
sudo make install
```

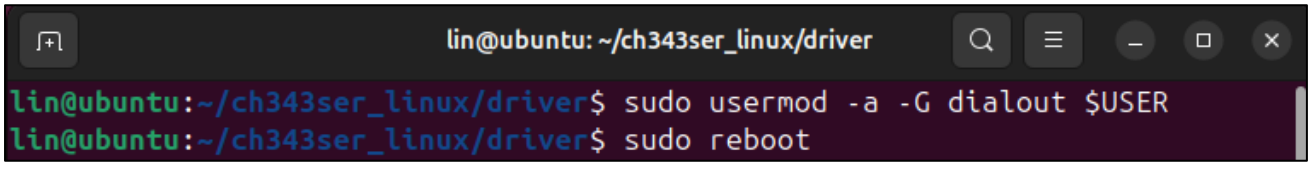


Connect the ESP32S3 to your computer, run the following command and you should see the port.

```
ls /dev/tty*
```

In Linux, higher permissions are required to access "ttyACM0," so privilege escalation commands must be used.

```
sudo usermod -a -G dialout $USER
sudo reboot
```



Reboot the system to have the configuration take effect.

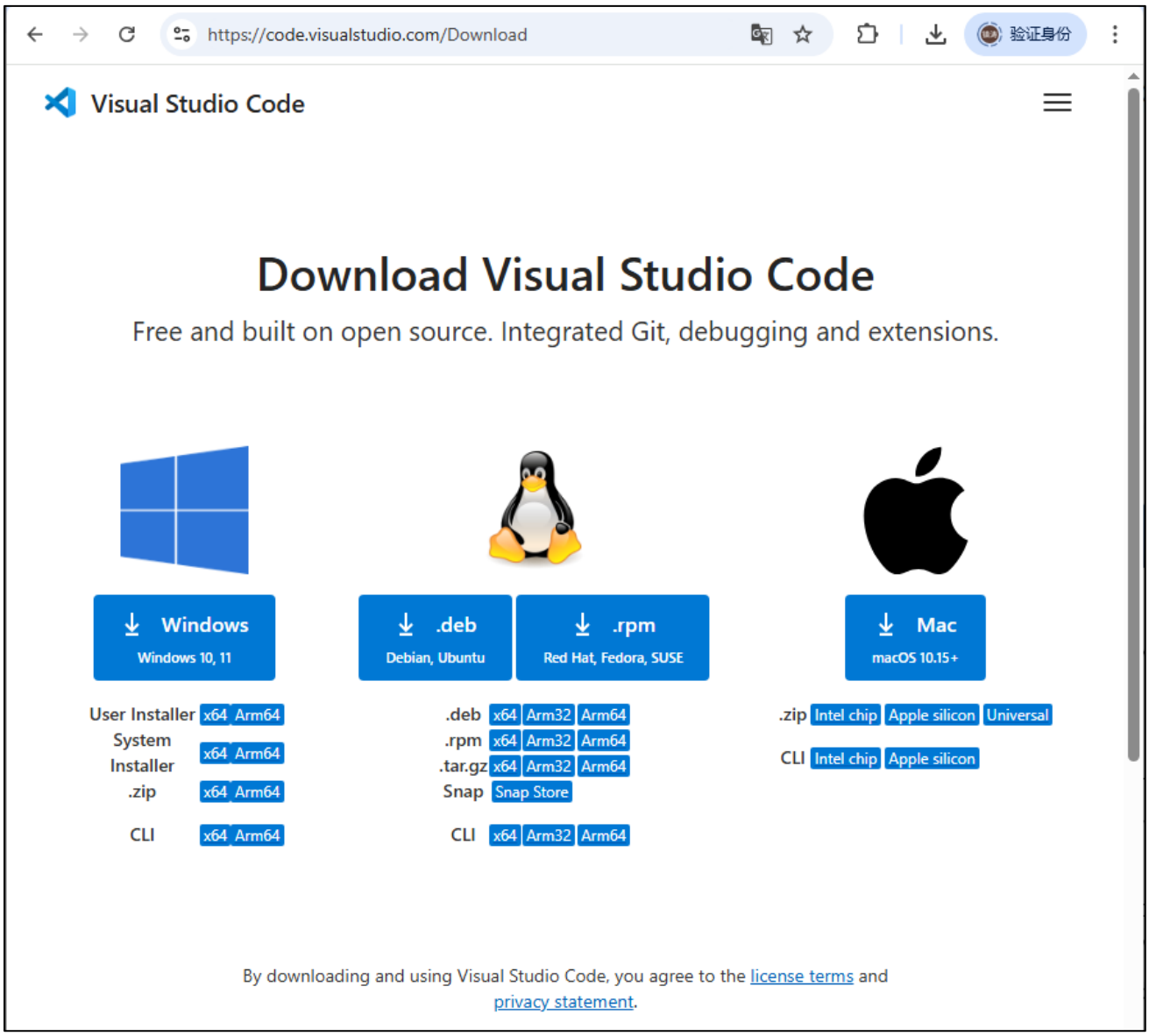
OpenAI Code

Visual Studio Code

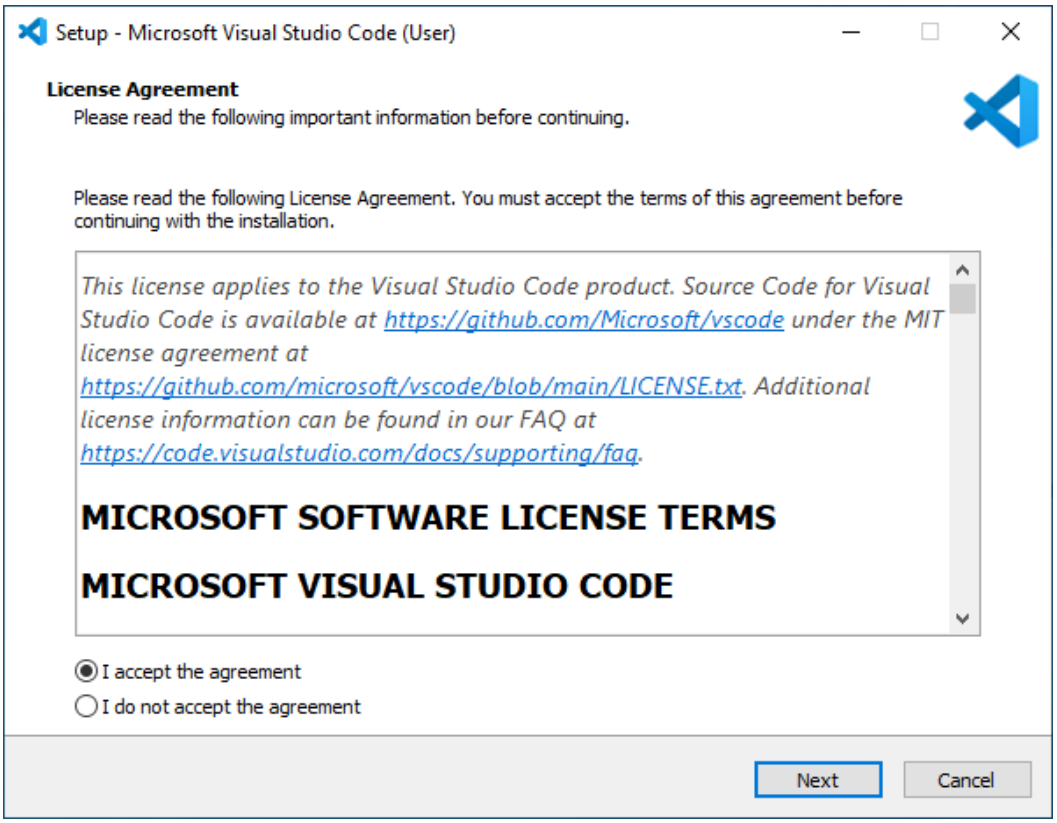
As we use Visual Studio Code to compile and upload code, we need to install the software before proceeding.

Windows

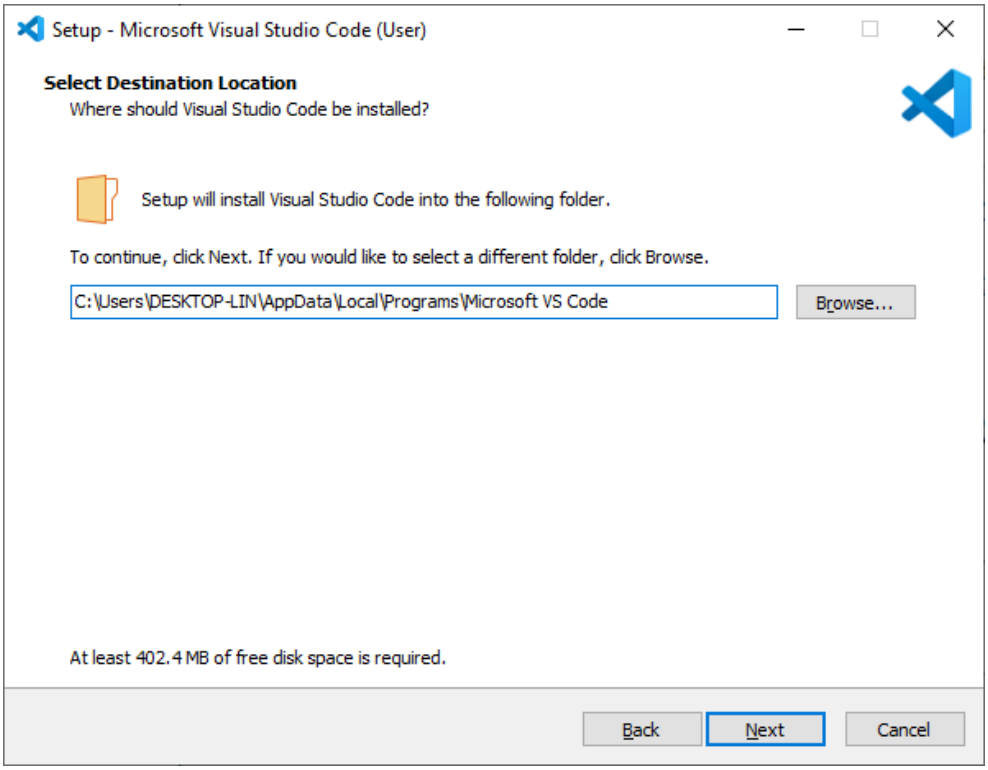
First, download Visual Studio Code by visiting https://code.visualstudio.com/Download. Choose the appropriate version for your operating system, then download and install it



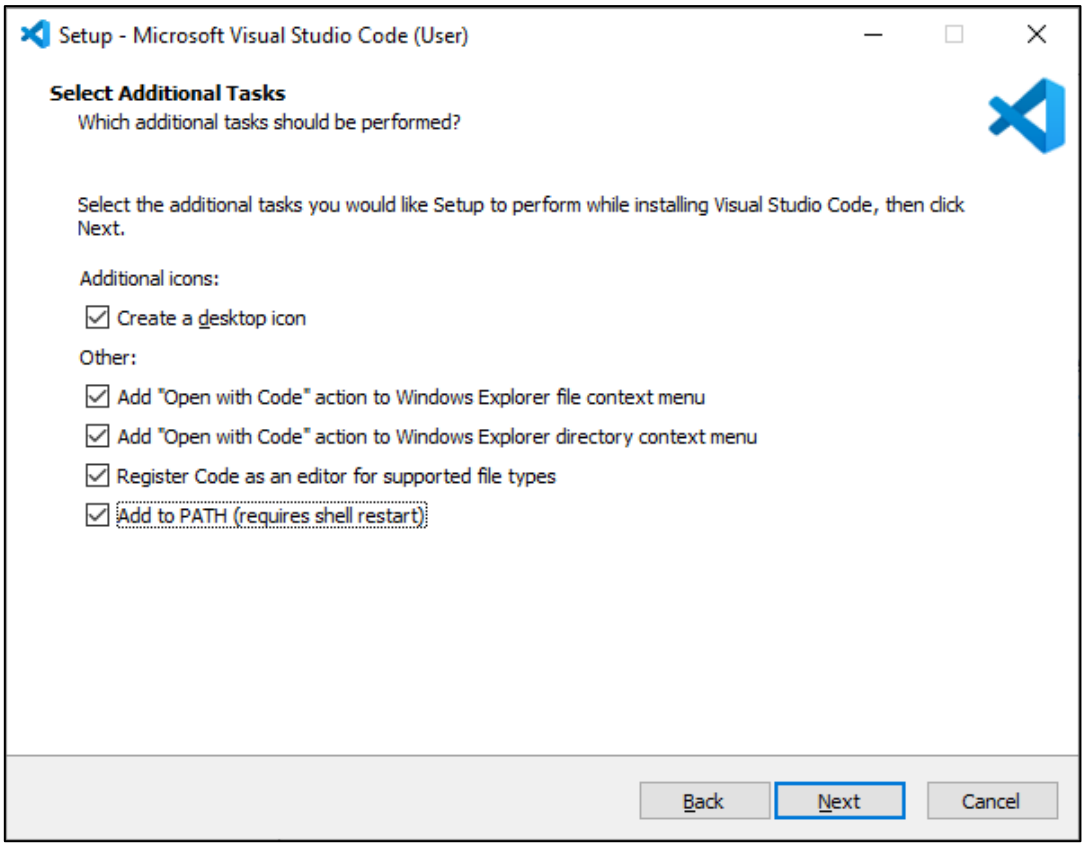
Double-click the downloaded .exe file to run it.
Check the box for "I accept the agreement."
Then click Next.



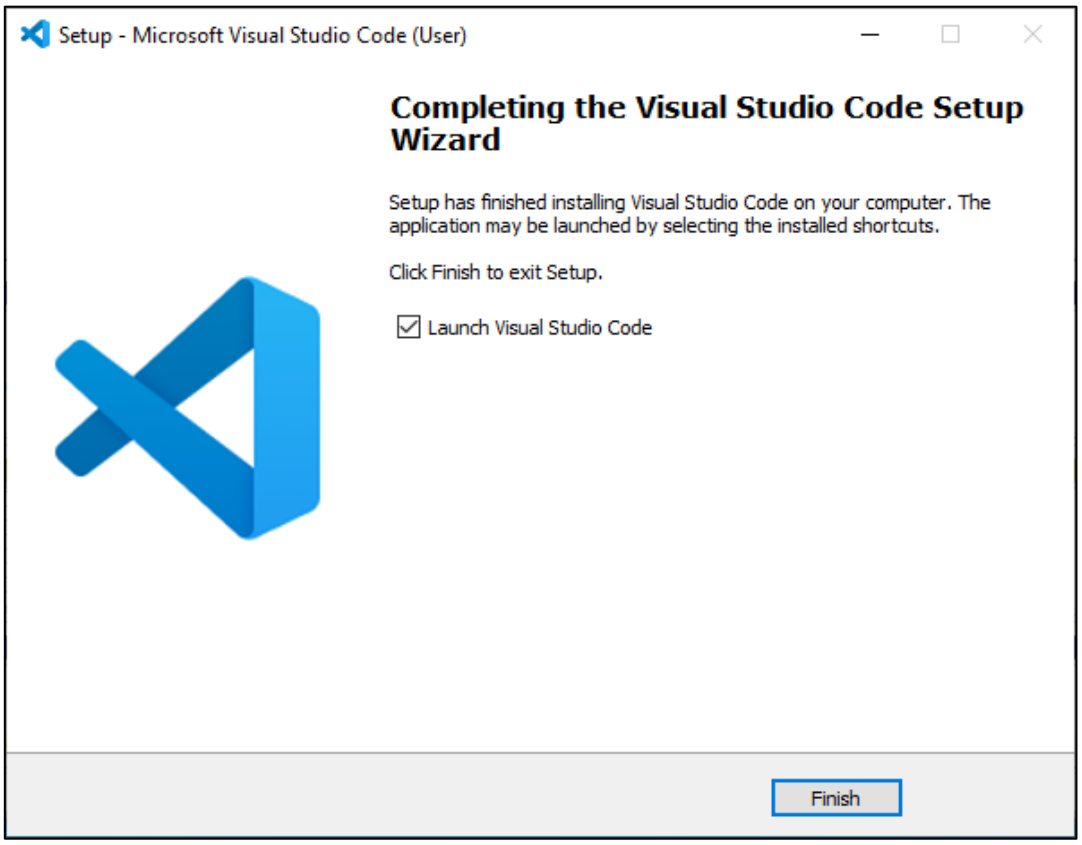
The installation location can be left as the default or changed to a desired path. After that, proceed by clicking Next repeatedly.



On this screen, verify that "Add to PATH" is selected. If unchecked, enable it. Proceed by clicking Next repeatedly to finish the installation.



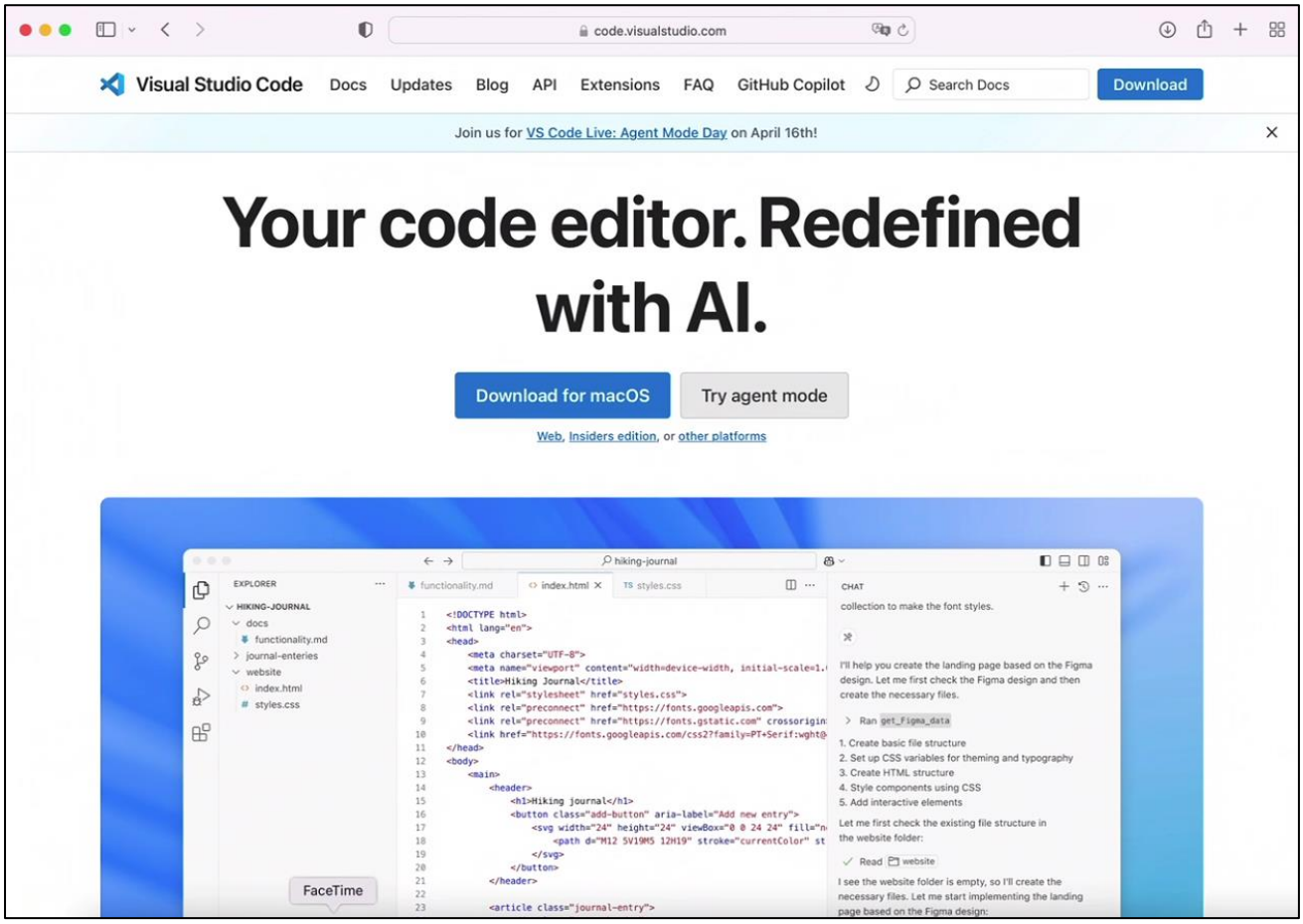
The installation is now complete, as shown in the image below.



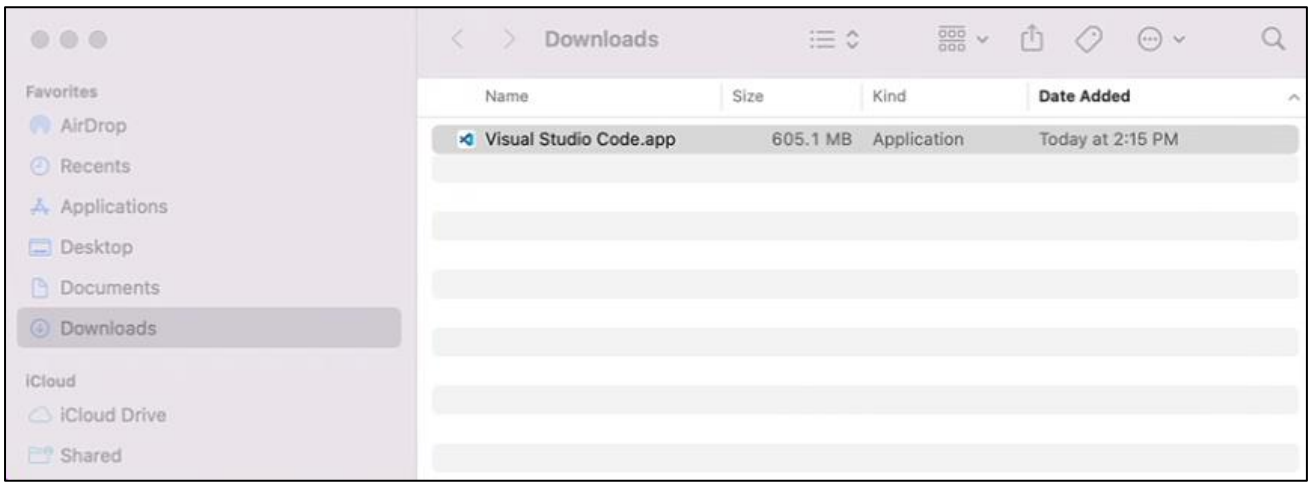
Mac

Typically, MacOS comes with Visual Studio Code pre-installed. If your computer does not have it, please install it first.

Visit https://code.visualstudio.com and click "Download for macOS" .

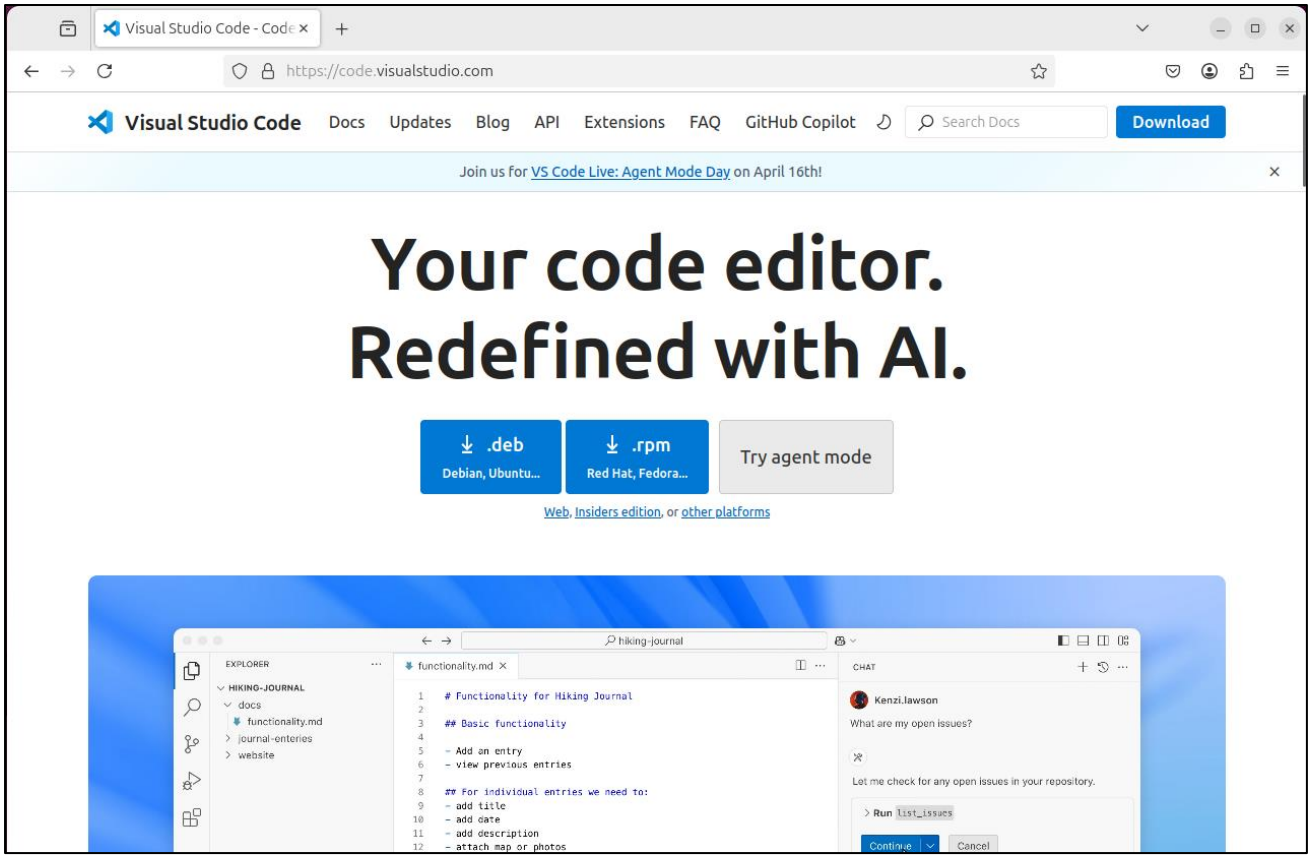


Double click to run the program.

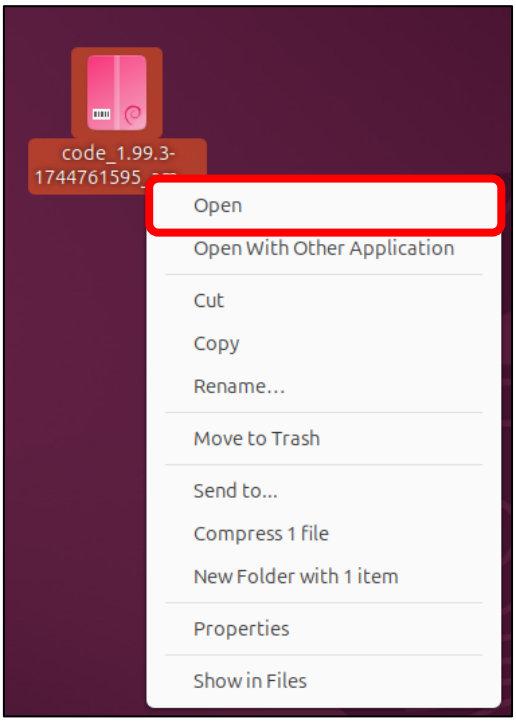


Linux

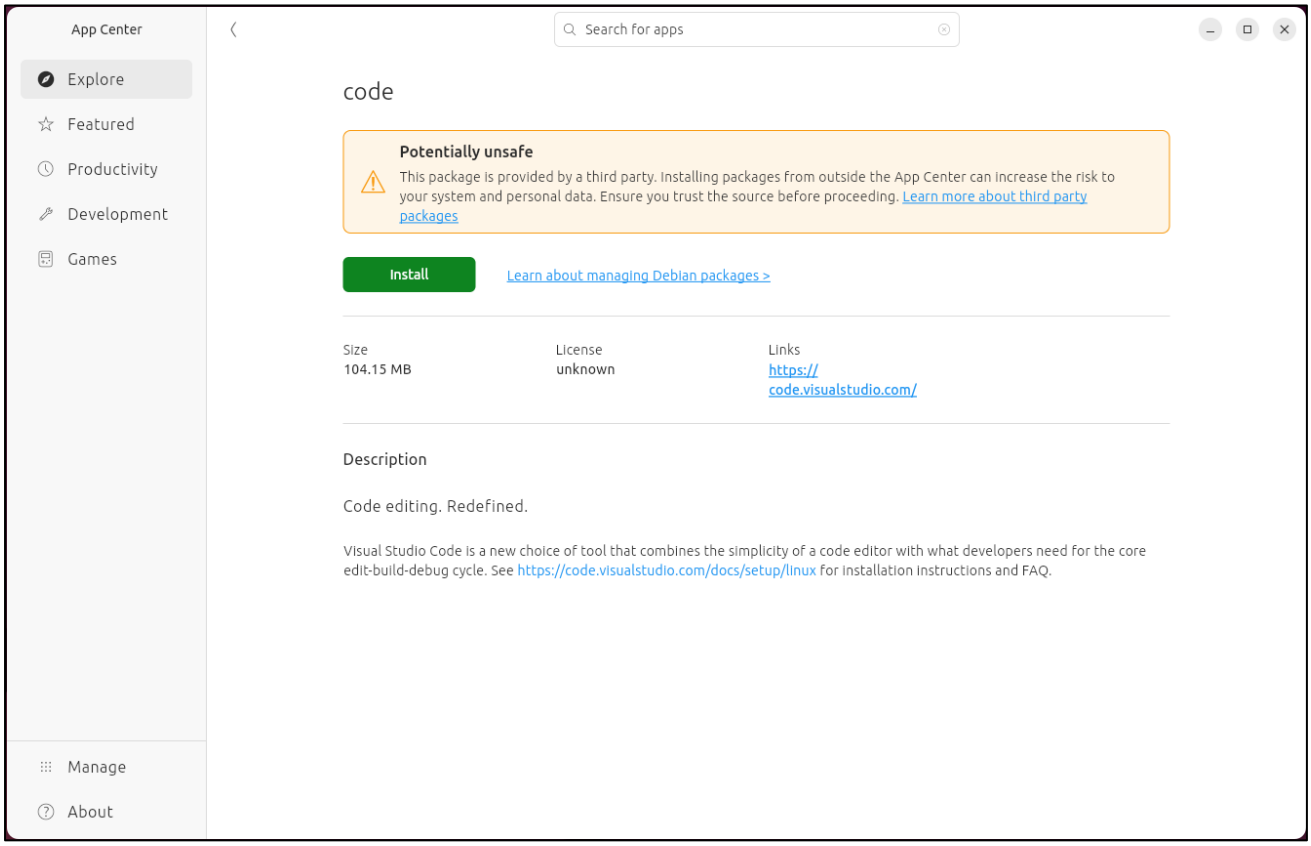
If your computer does not have Visual Studio Code, please install it first.
Visit https://code.visualstudio.com and click ".deb" .



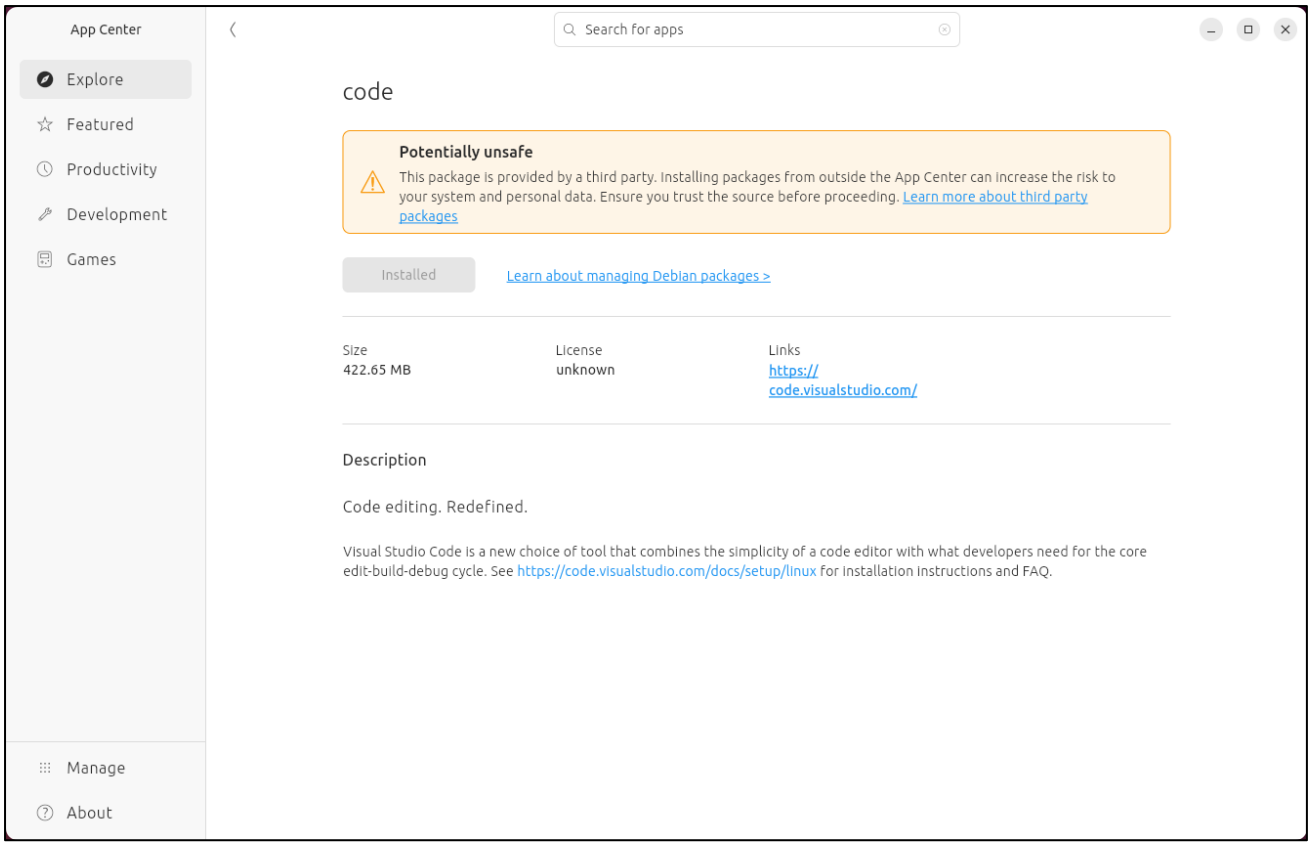
Open the downloaded“code_xxx.deb” file.



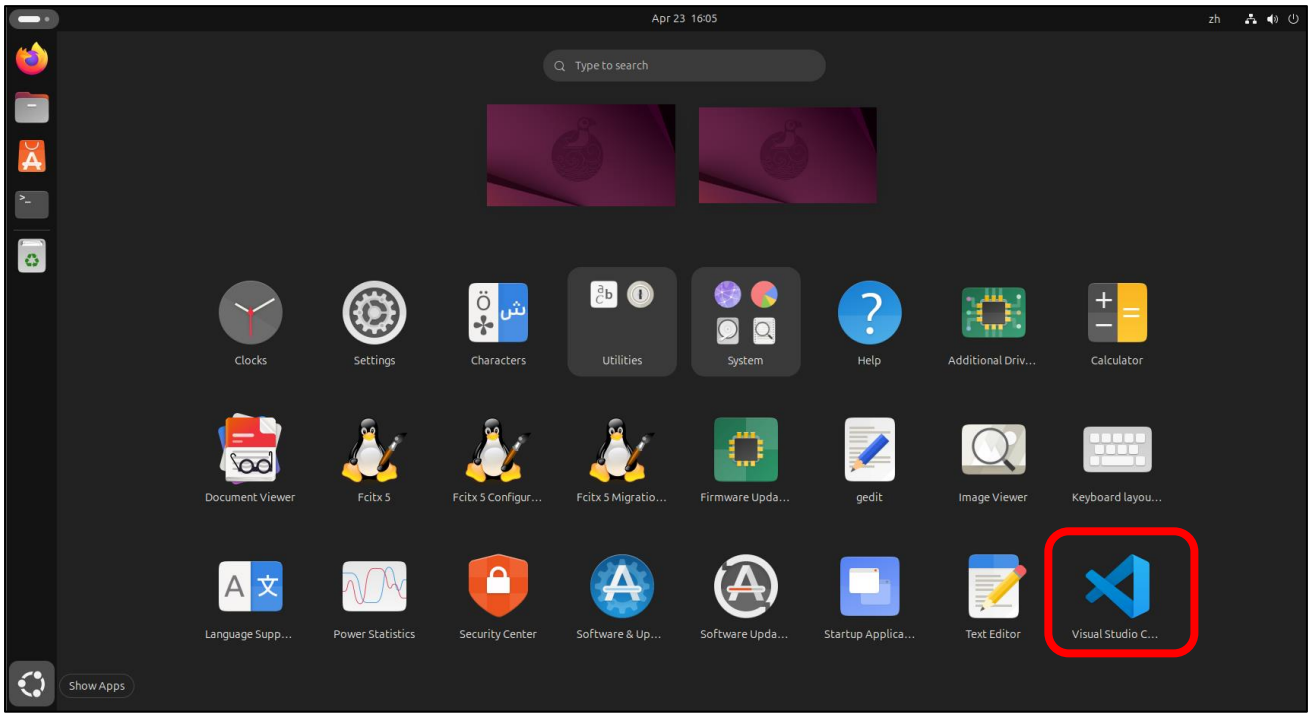
Click "Install" to install Visual Studio Code.



Wait for the installation to complete. Once finished, it should look like the image below.



Click Show Apps and you can see the Visual Studio Code is in the system.

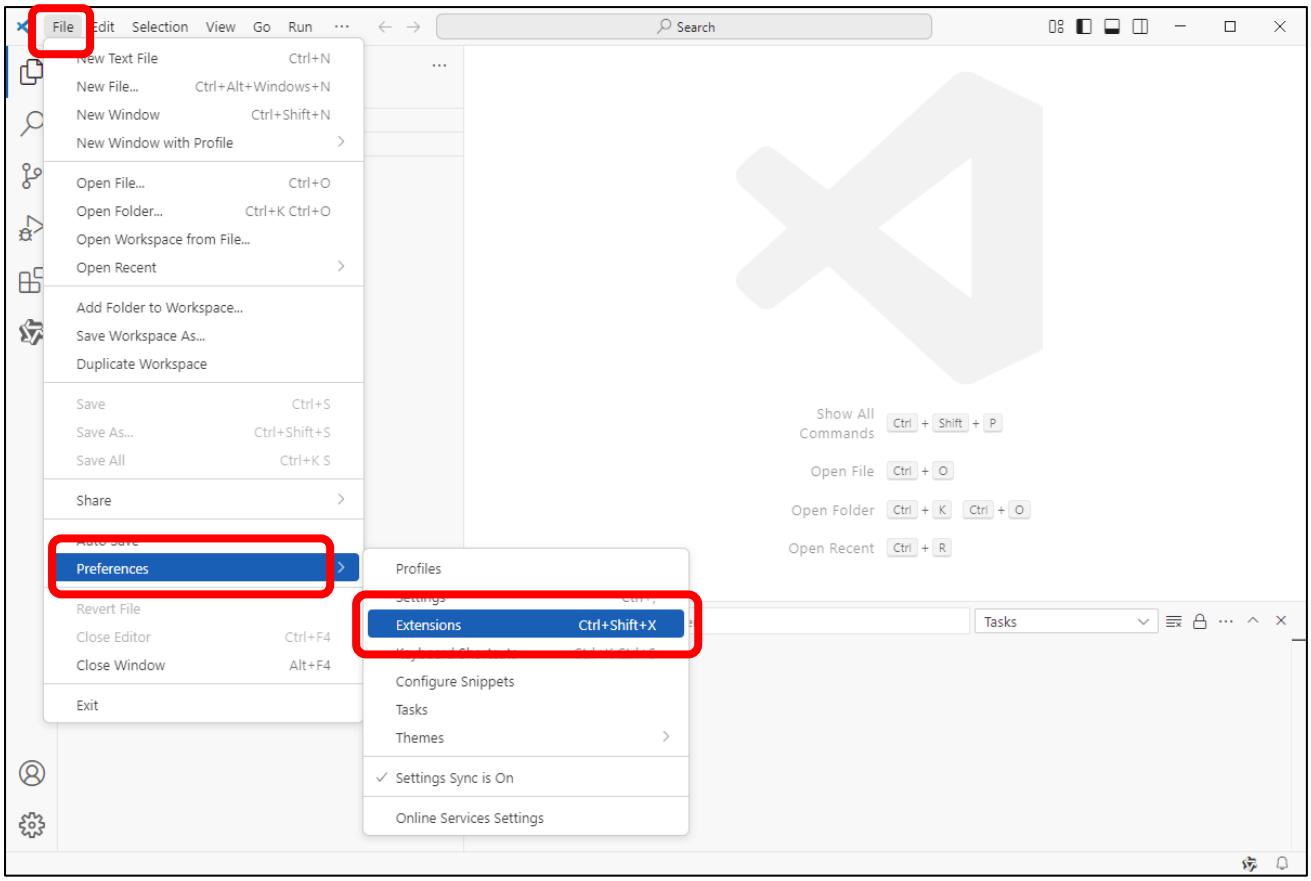


ESP-IDF V5.4.1

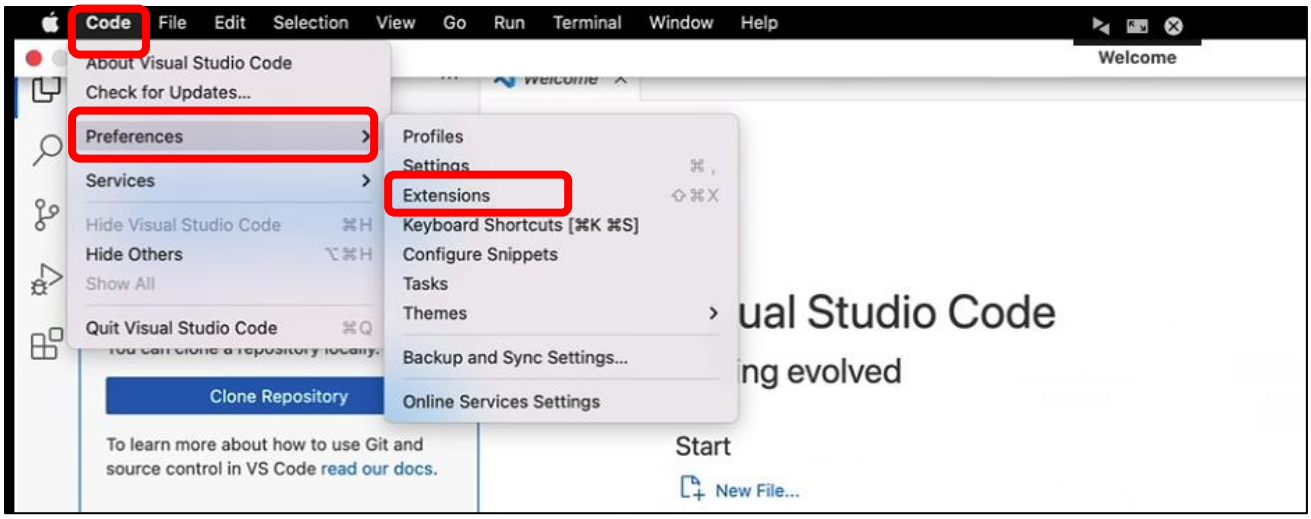
Visual Studio Code is a versatile code editor. To program with the ESP-IDF SDK, we need to install the ESP-IDF extension for it.

Open Visual Studio Code.

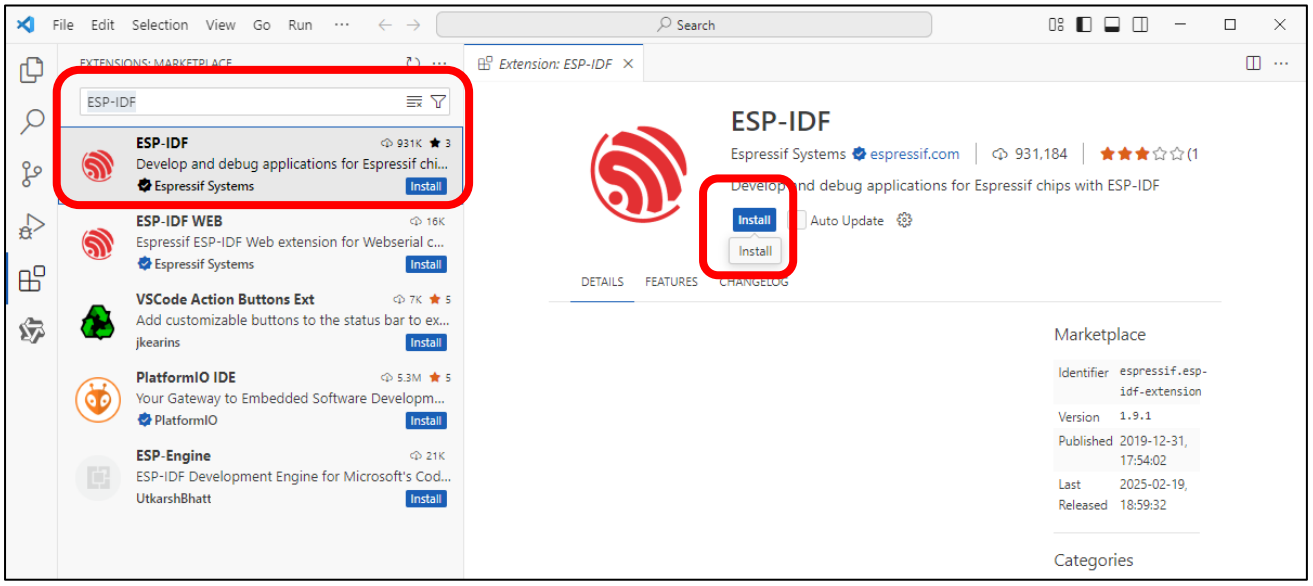
Click on the menu bar: File -> Preferences -> Extensions.



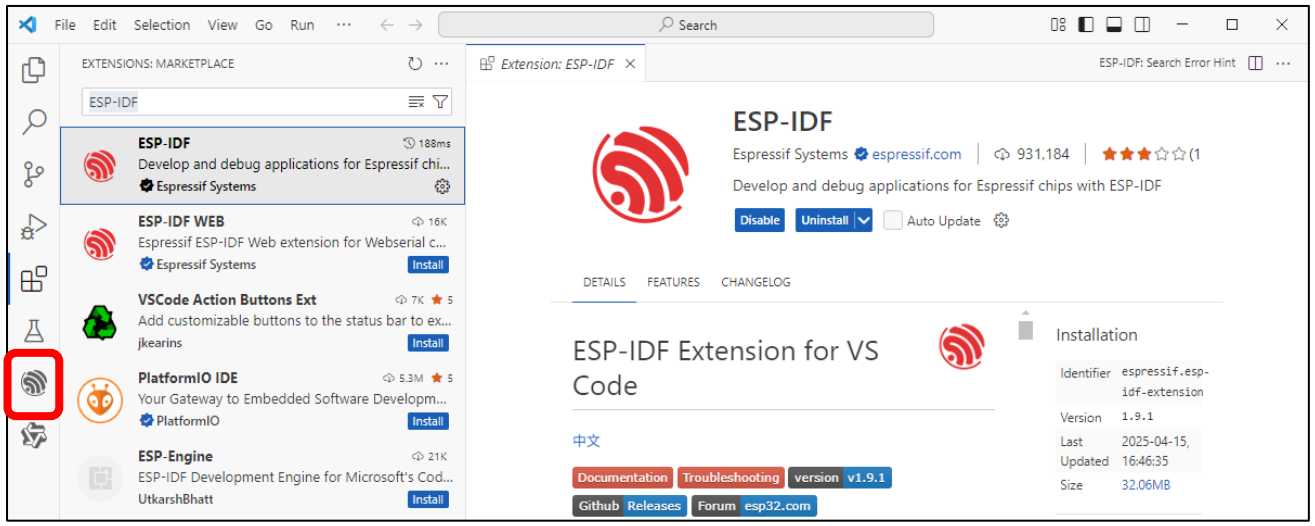
Mac OS: Click "Code"->"Preferences"->"Extensions" on the menu bar.



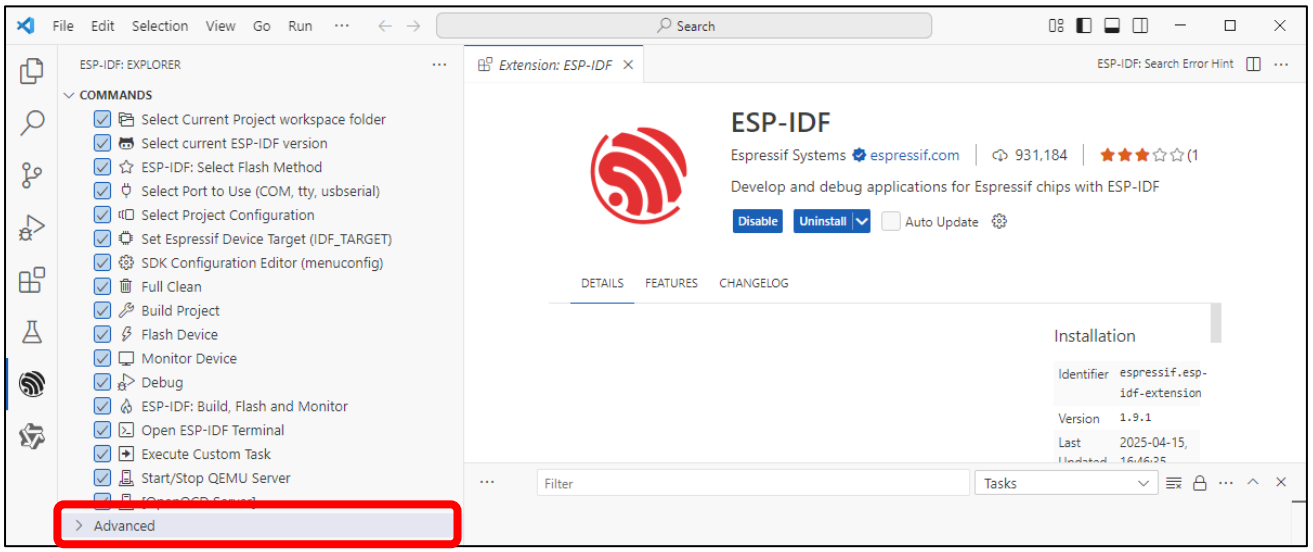
Search for "ESP-IDF" in the extension bar, select the correct result from the list, then click the Install button to proceed.



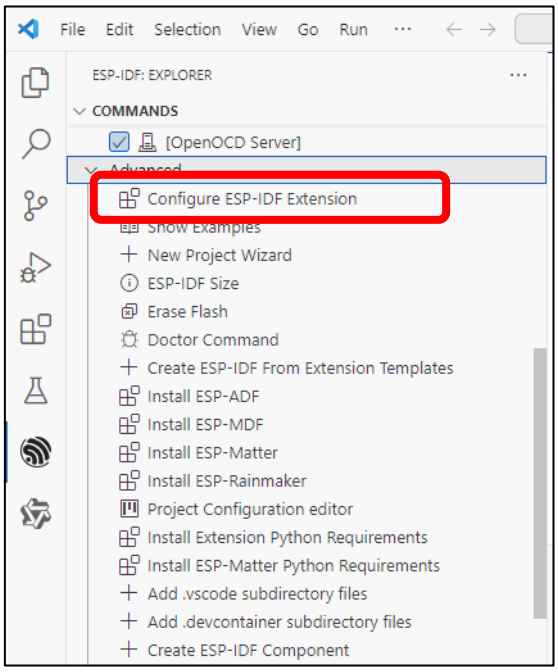
The ESP-IDF extension icon will now appear in the left sidebar - click it to continue.



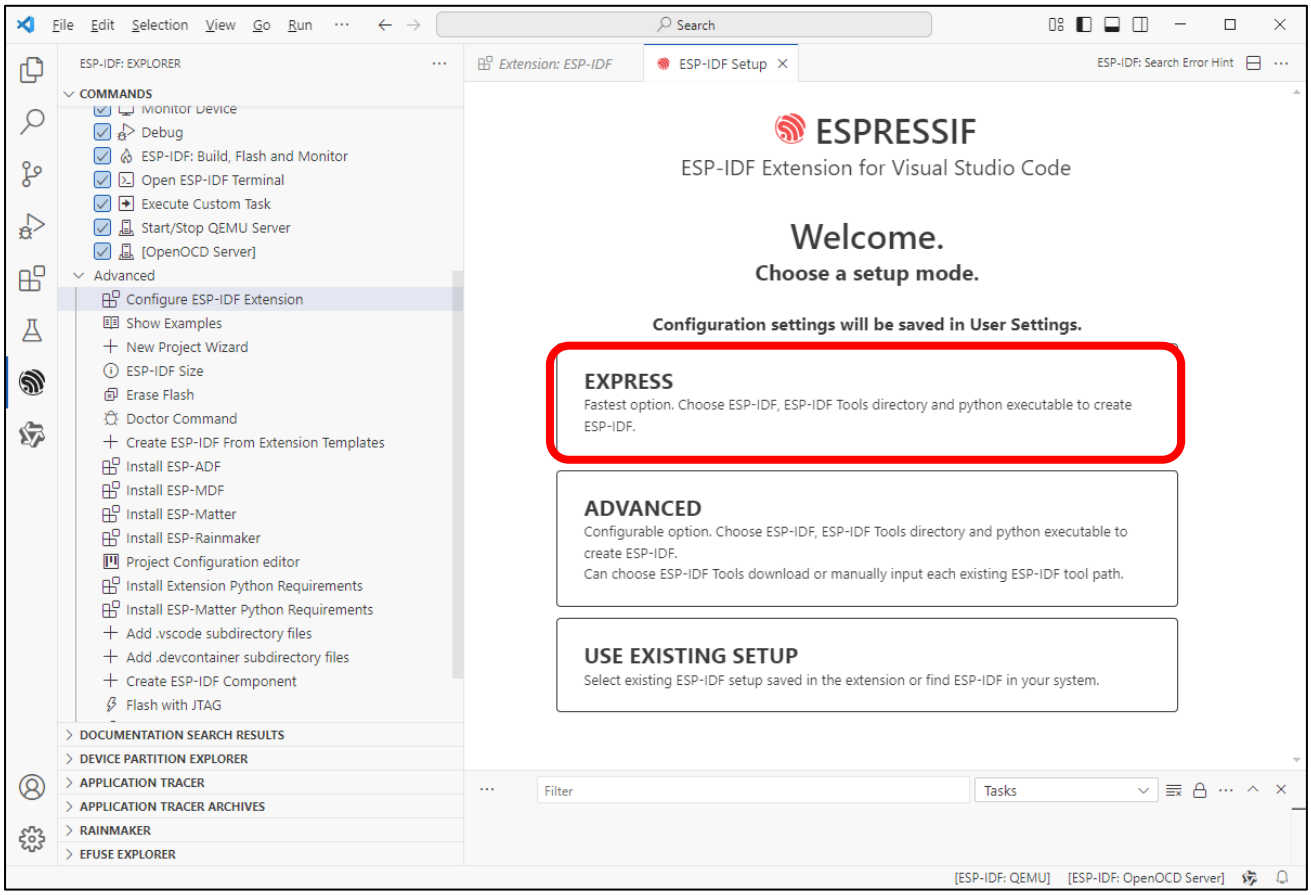
Scroll down with your mouse, locate and click on the "Advanced" option.



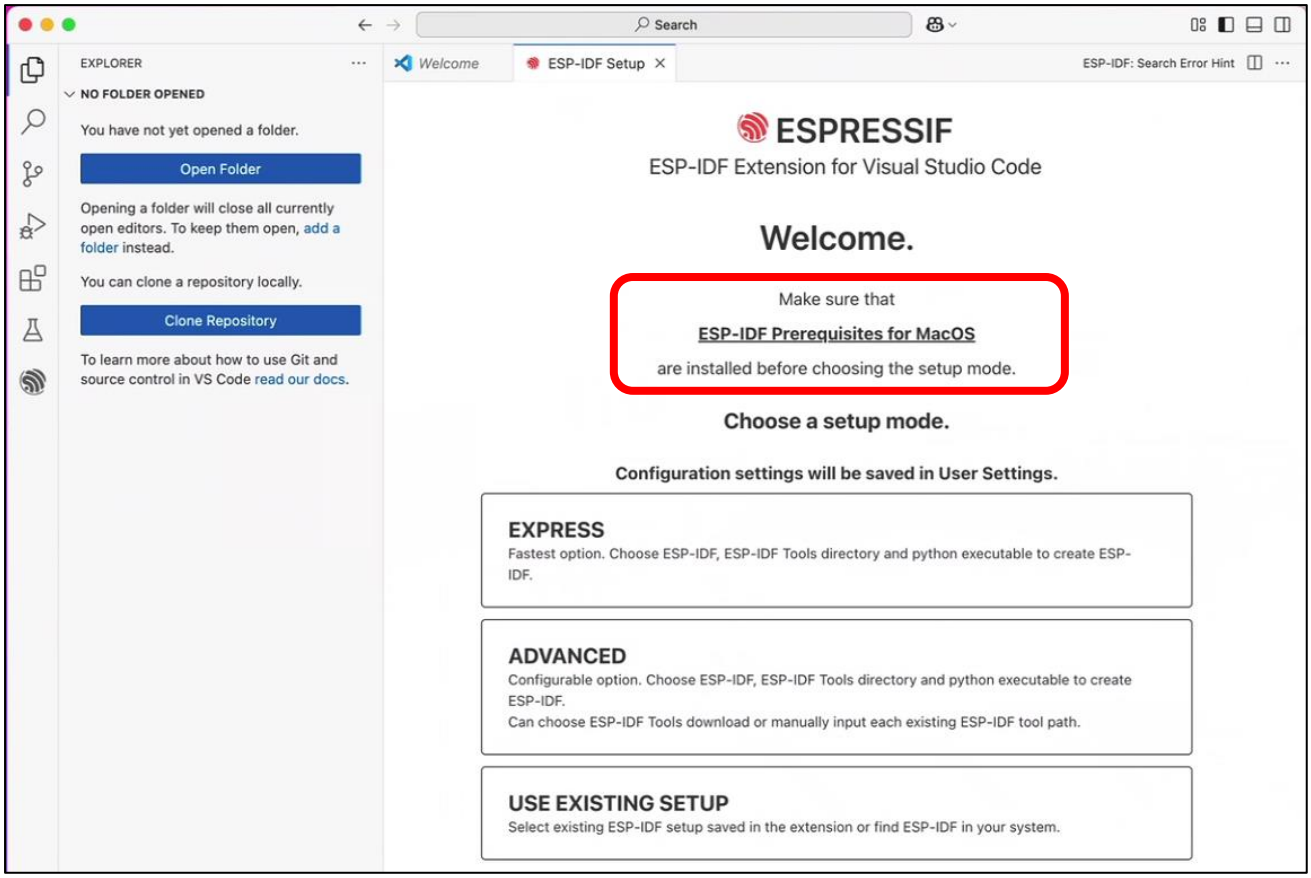
Click the first option: "Configure ESP-IDF Extension".

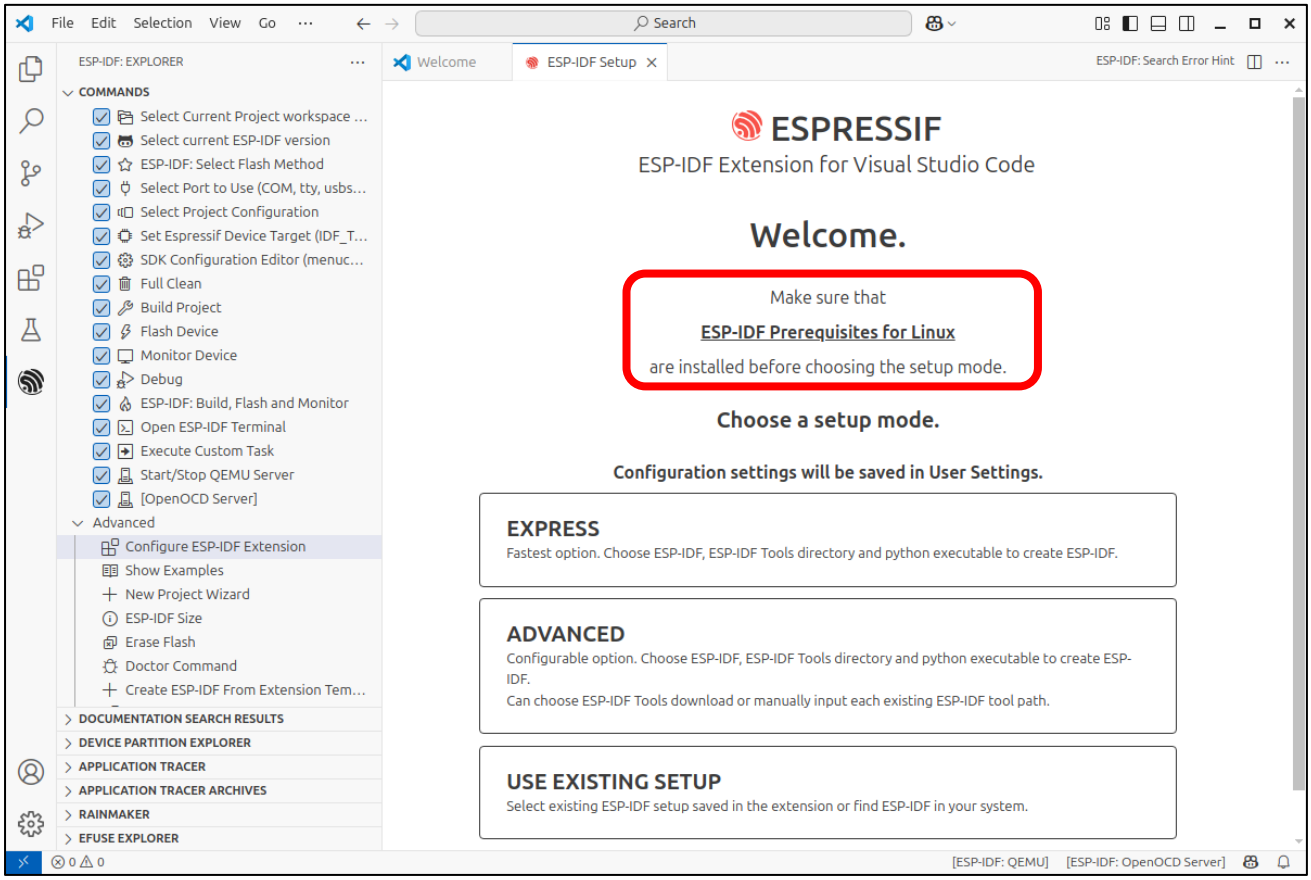


Select "EXPRESS" on the right.



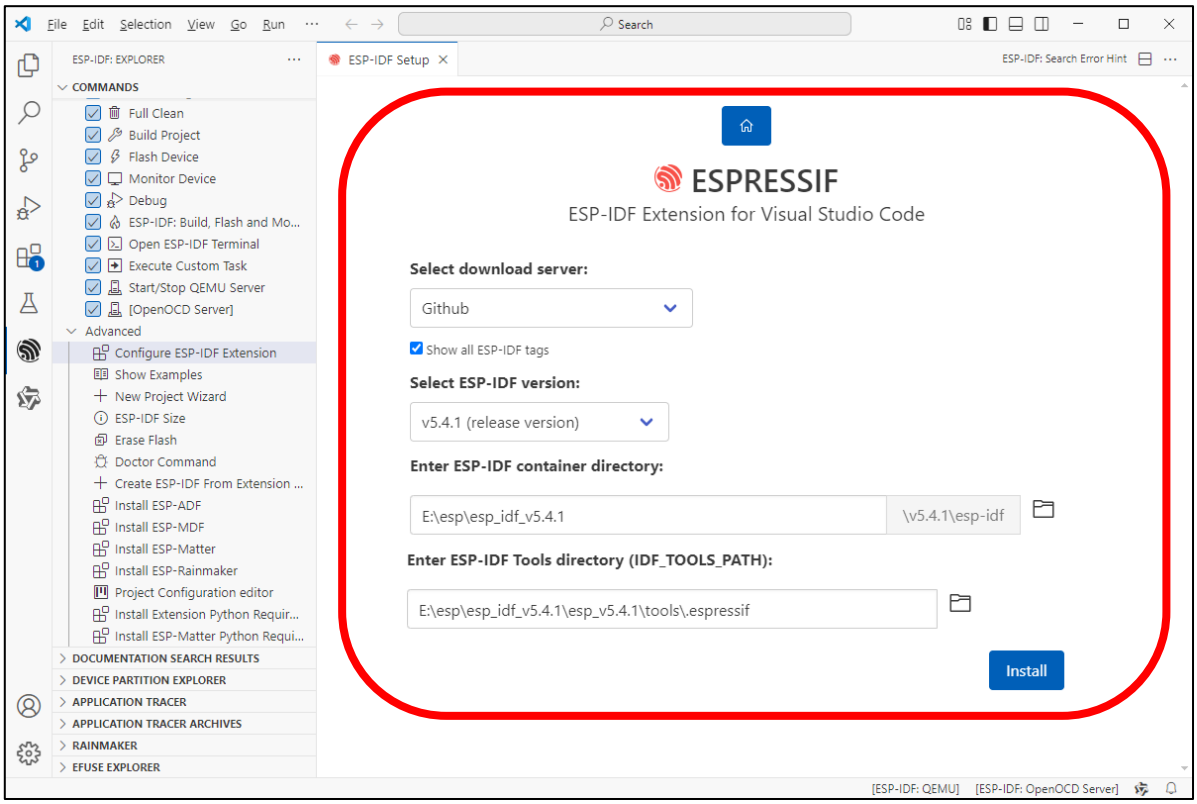
Note: If you're using macOS or Ubuntu, please complete the necessary preparations as prompted before proceeding with installation.





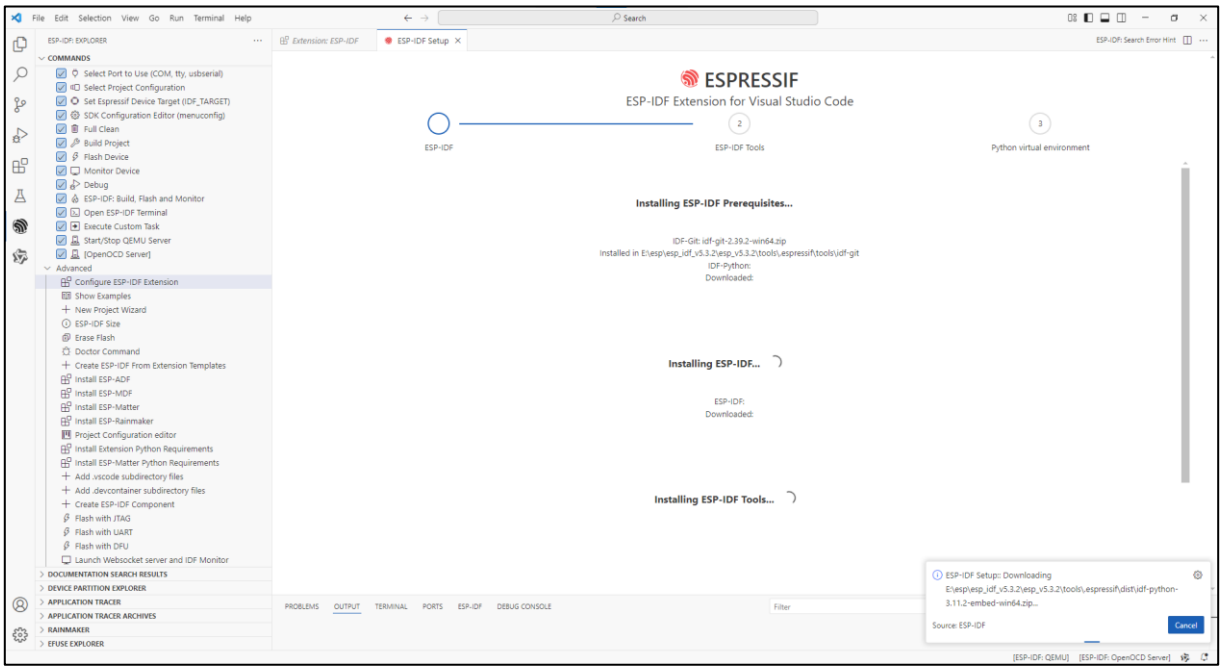
1. Check the box for "Show all ESP-IDF tags"
2. Select "v5.4.1 (release version)" from the dropdown
3. Choose your desired installation path for the ESP-IDF environment
4. Click "Install" to begin the setup

The installation path varies among computer systems, please remember it.



The process will complete automatically.

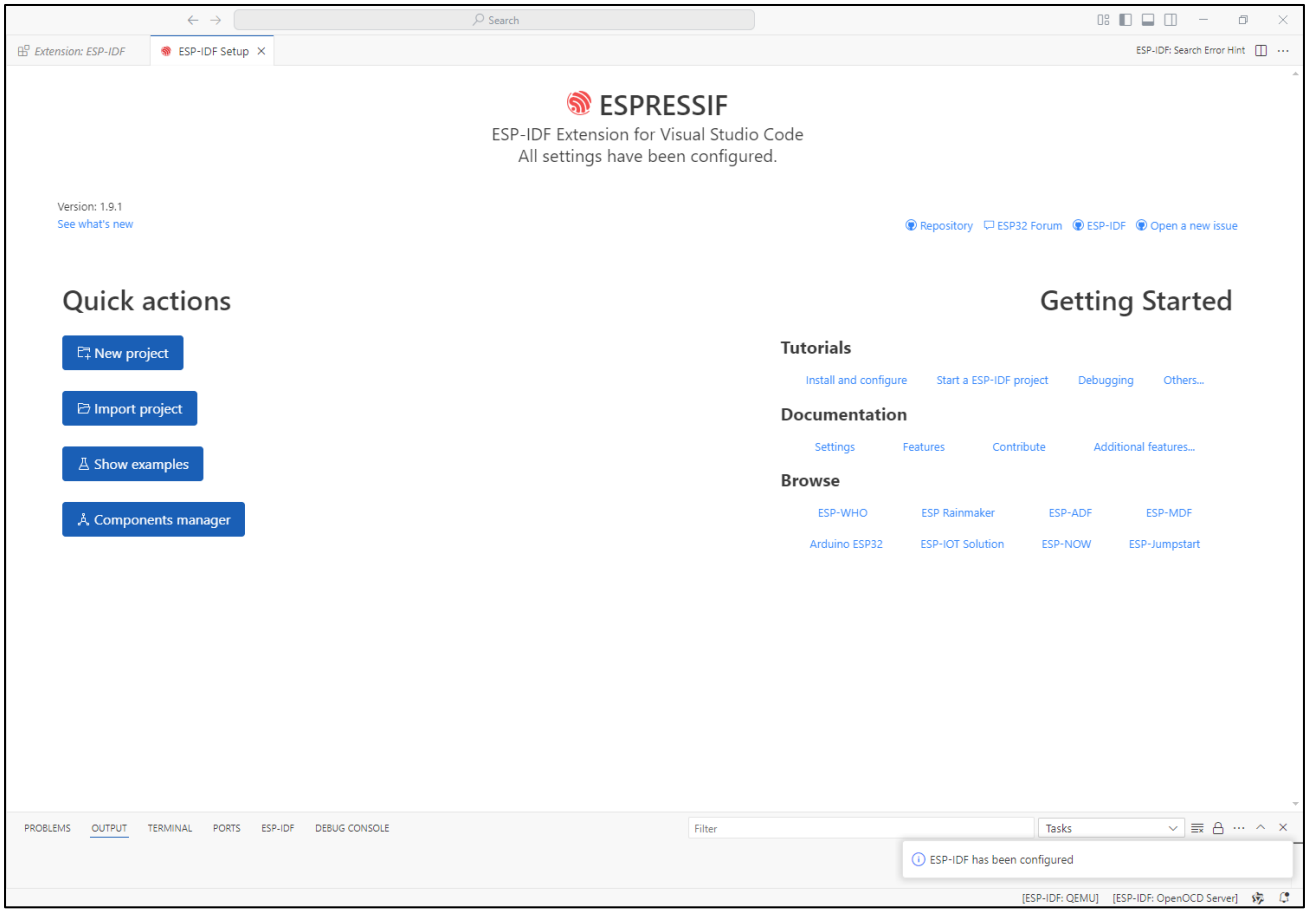
If it failed, locate your chosen ESP-IDF directory, remove the failed installation folder and install it again.



This step may take a while, so please ensure you have a stable and fast internet connection.

If the installation continues to fail, check the relevant link for your operating system below:
Window: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/windows-setup.html
Mac & Linux: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/linux-macos-setup.html

The complete installation is as shown below.



For more about ESP-IDF, please refer to
https://docs.espressif.com/projects/vscode-esp-idf-extension/en/latest/installation.html

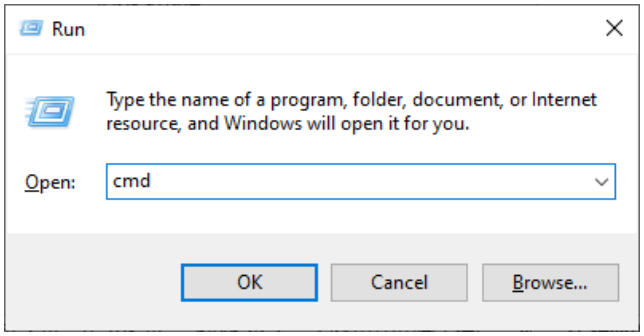
OpenAI Code

This project is derived from the open-source repository: https://github.com/openai/openai-realtime-embedded, licensed under MIT License. We have only adapted it for third-party learning and AI functionality trials, without any commercial promotion or application. This tutorial is intended solely for enthusiasts to supplement their learning.

Code Downloading

Windows

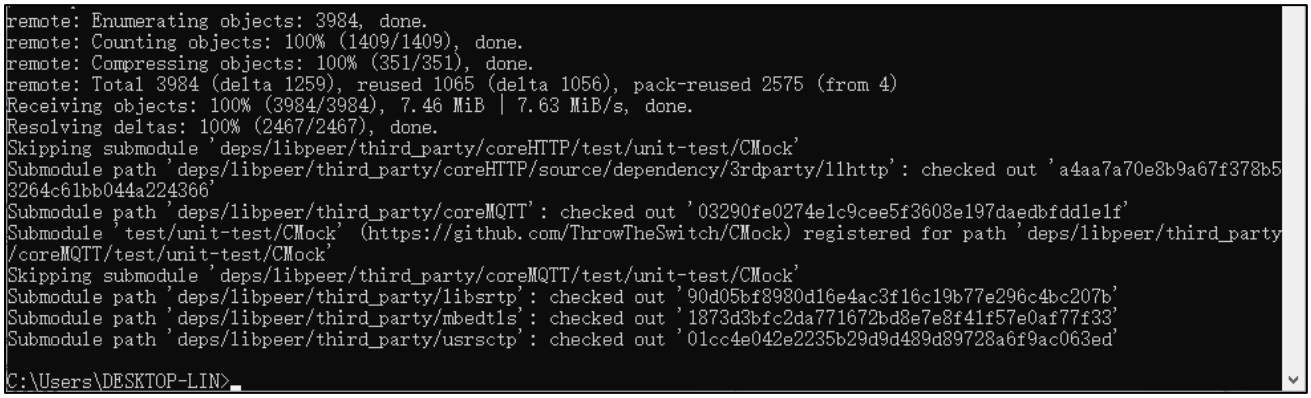
Use the shortcut "Win+R", enter "CMD" in the pop-up window, and open the CMD interface.



In the Terminal, install the code with git command.

```
git clone --recurse-submodules https://github.com/Freenove/openai-realtime-embedded
```

The installation is completed as shown below.



If you do not have the git tool on your computer, please download and install it by visiting https://git-scm.com/downloads

MAC & Linux

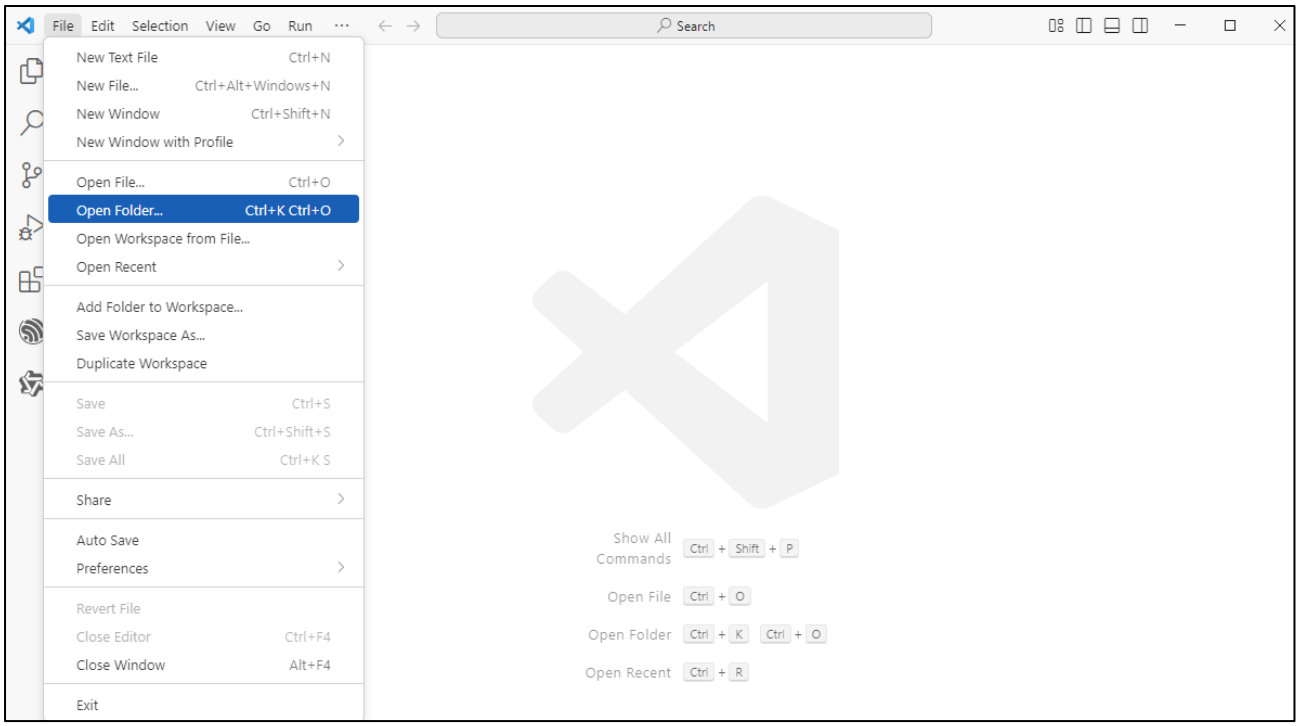
Open the Terminal, run the following command, and wait for the installation to finish.

```
git clone --recurse-submodules https://github.com/Freenove/openai-realtime-embedded
```

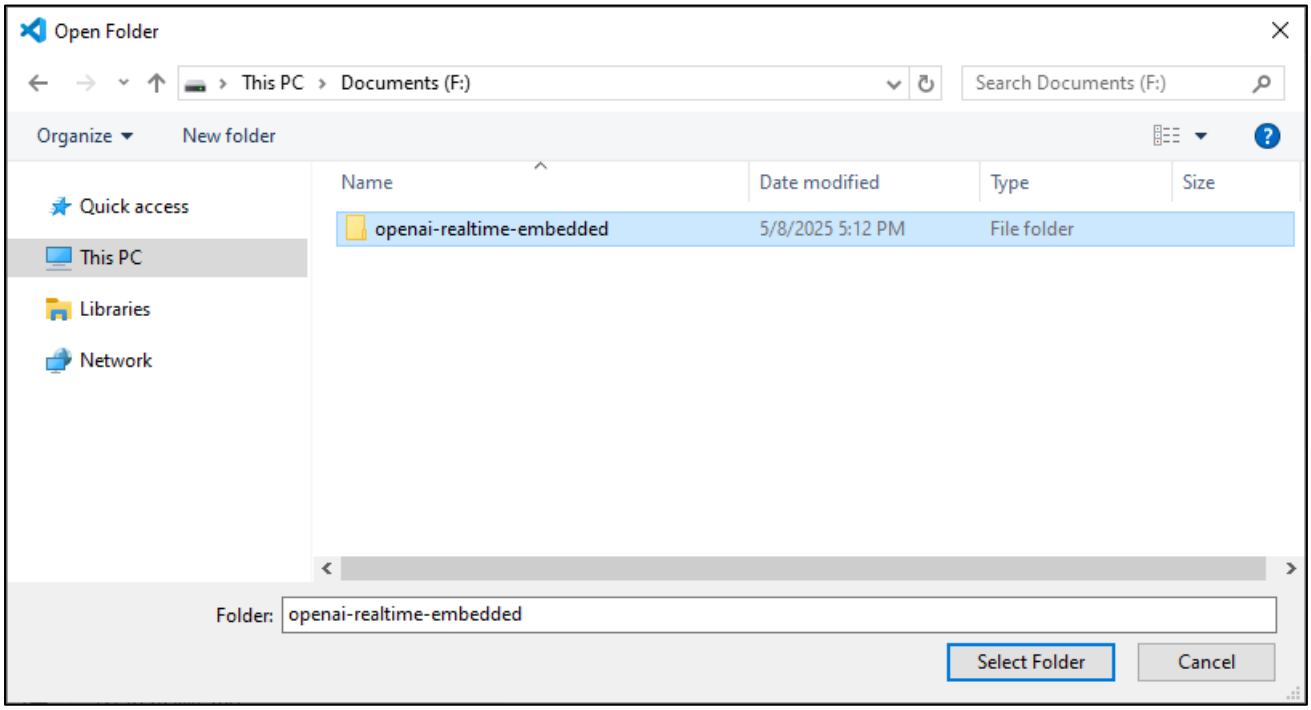
The installation completes as shown below.

Configuring Code Environment

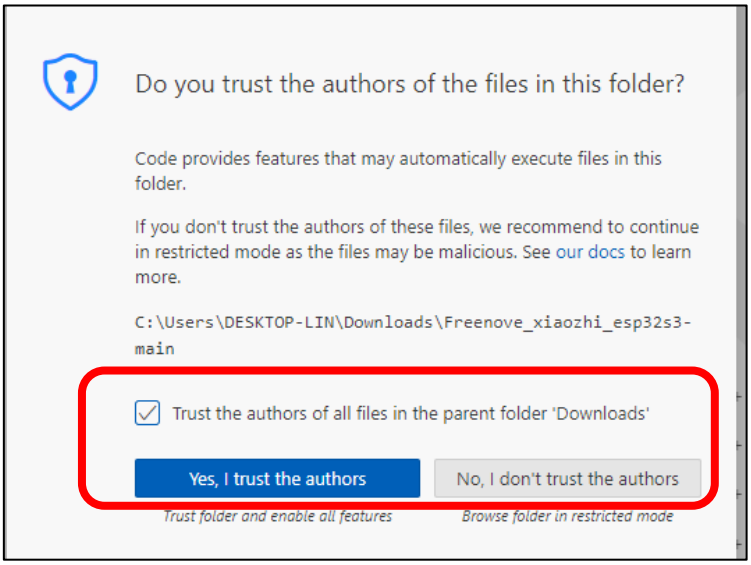
On Visual Studio Code, click “File” -> “Open Folder…”



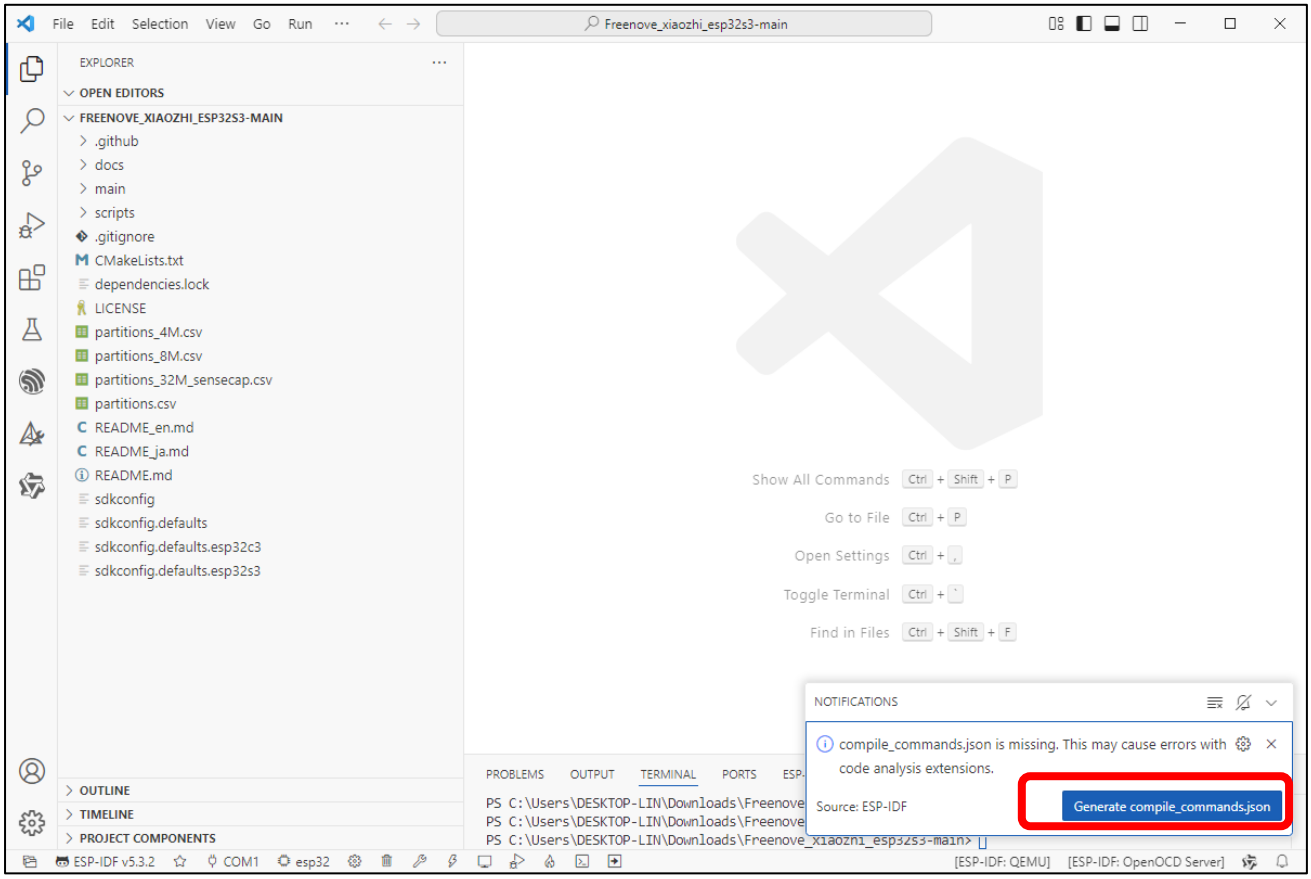
Select the openai-realtime-embedded older. Here, the interface of the Windows system is taken as an example. The operation on the mac and Linux system is similar.



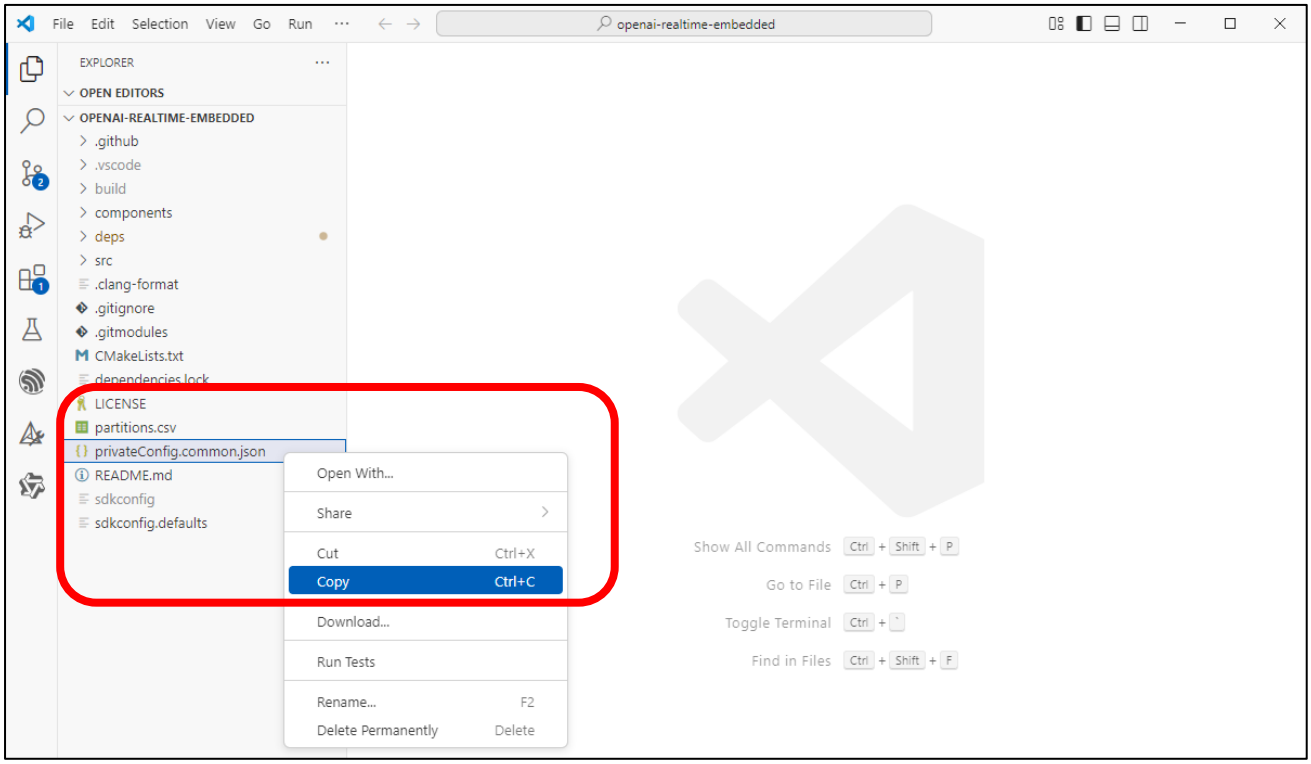
Check the box “Trust the authors of all files in the parent folder ‘Downloads” and select “Yes, I trust the authors”.



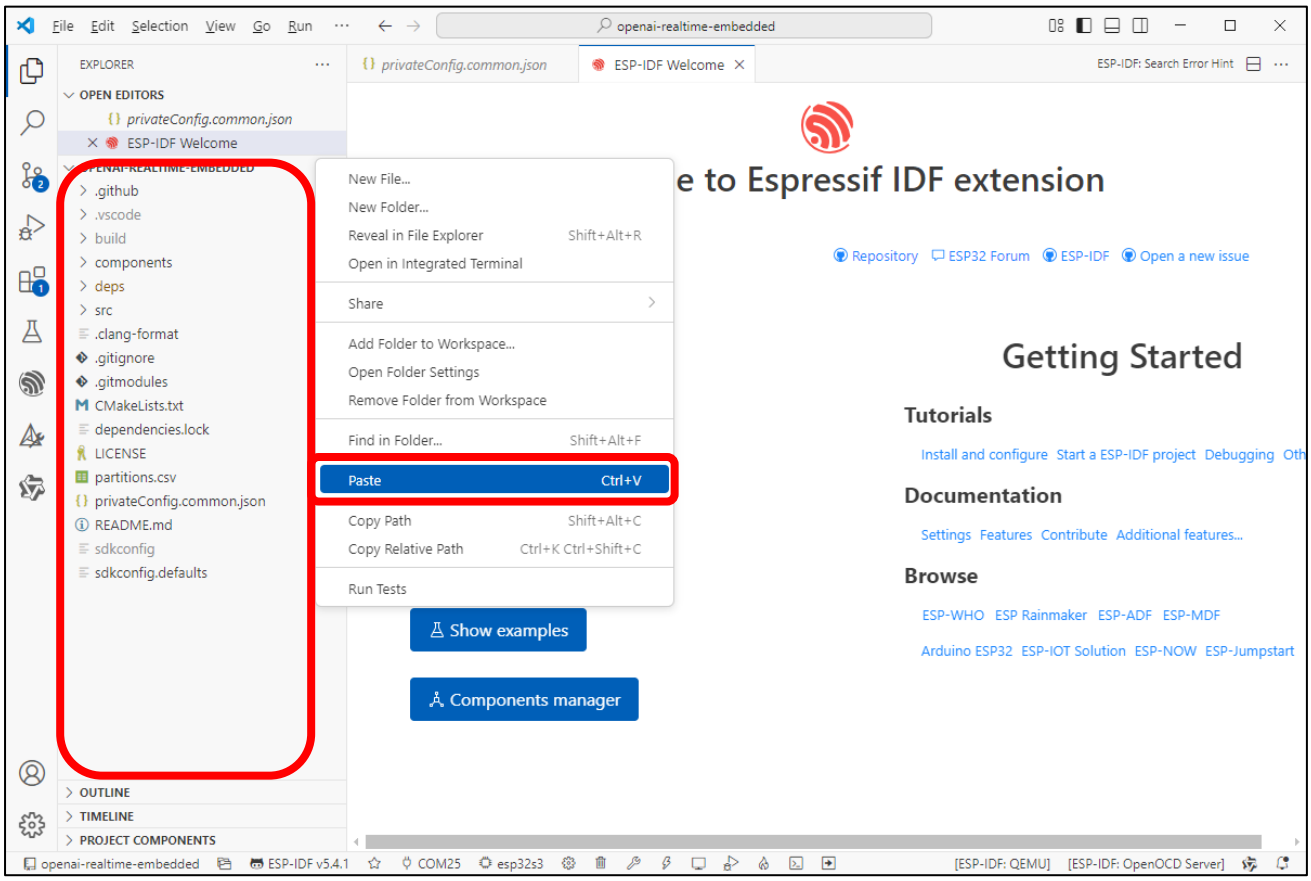
Please note: A pop-up notification 'Generate comple_commands.json' will appear in the lower-right corner. **Please disregard it. DO NOT click it.**



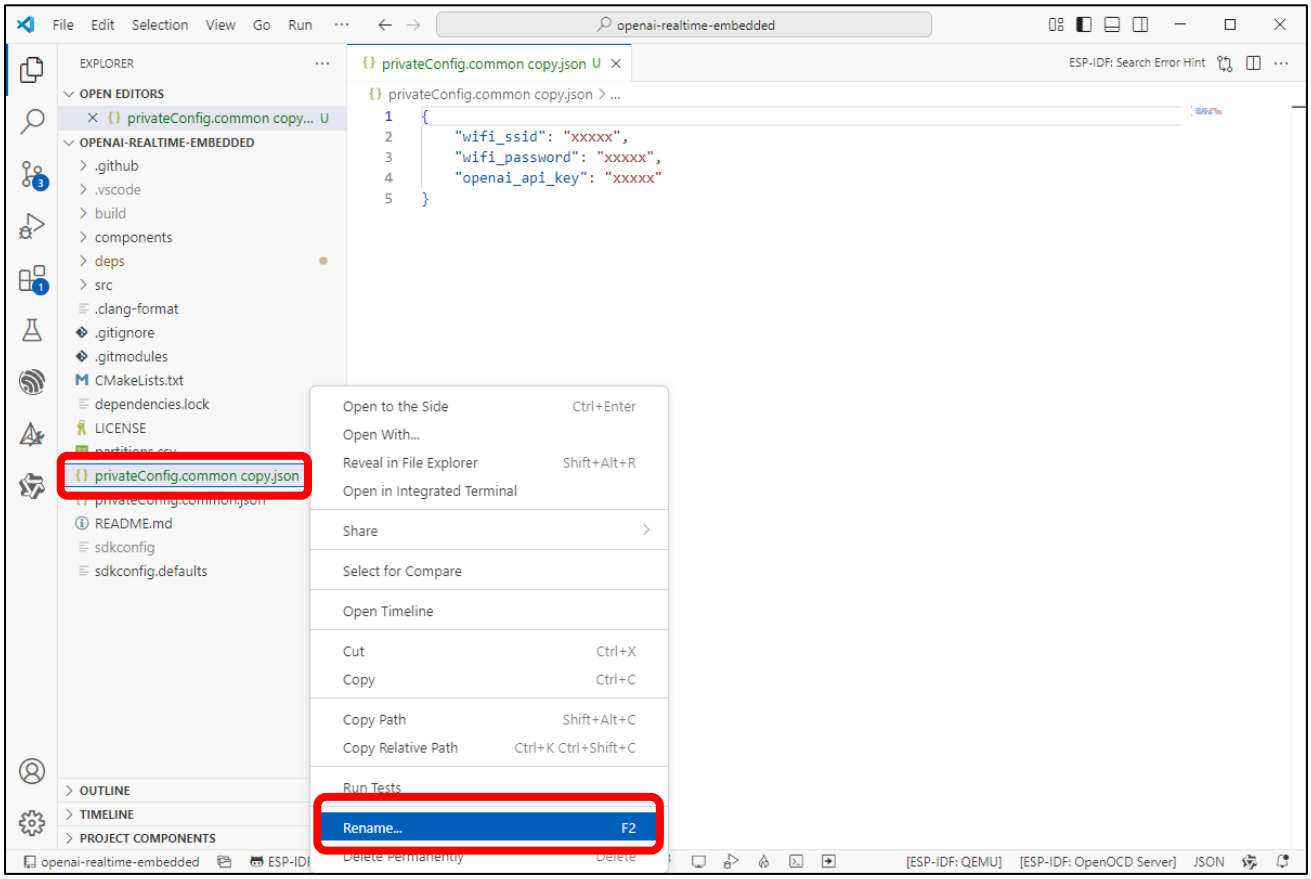
Find "**privateConfig.common.json**" on the left, right click it and click copy.



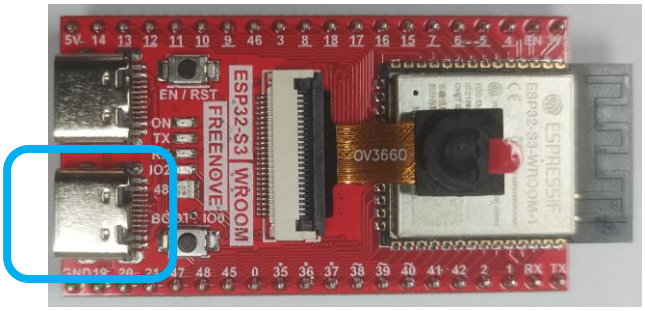
Click on a blank area in the left project panel, then right-click and click Paste.



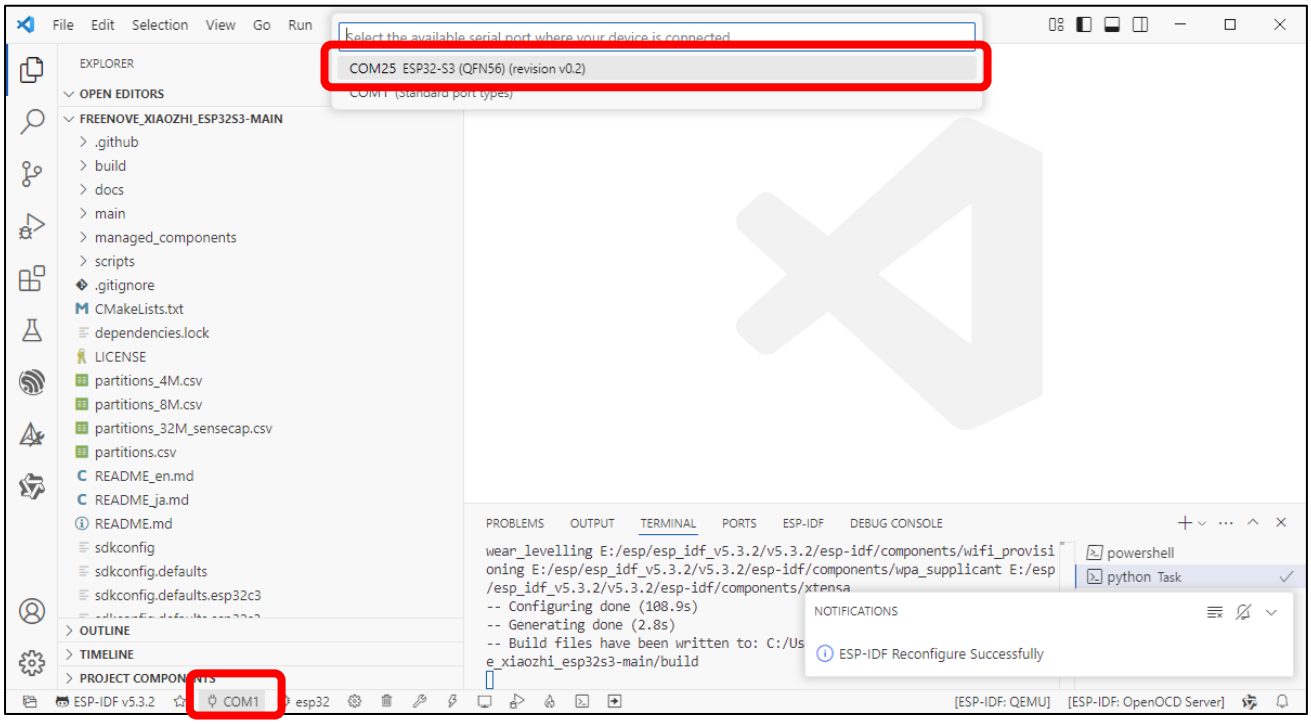
Rename it as "**privateConfig.json**" to protect your personal information.



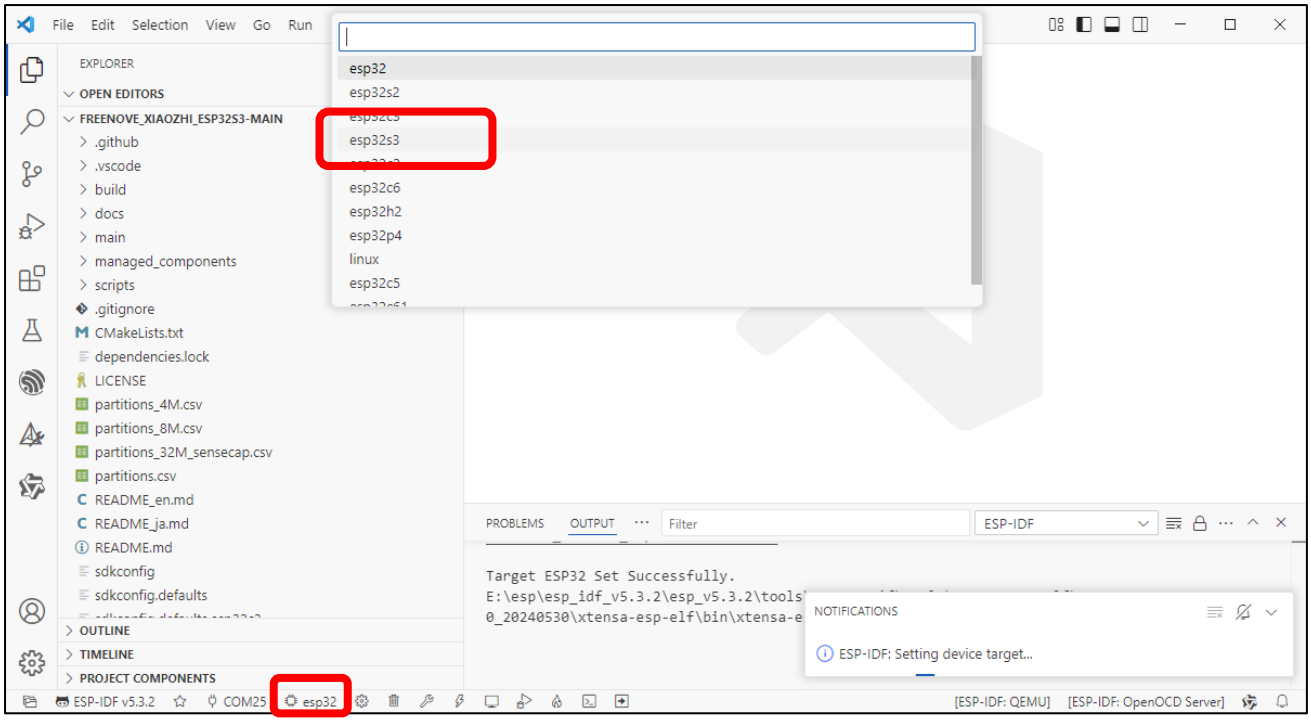
Connect the ESP32S3 WROOM board to your computer with the USB cable (do not use the wrong connector).



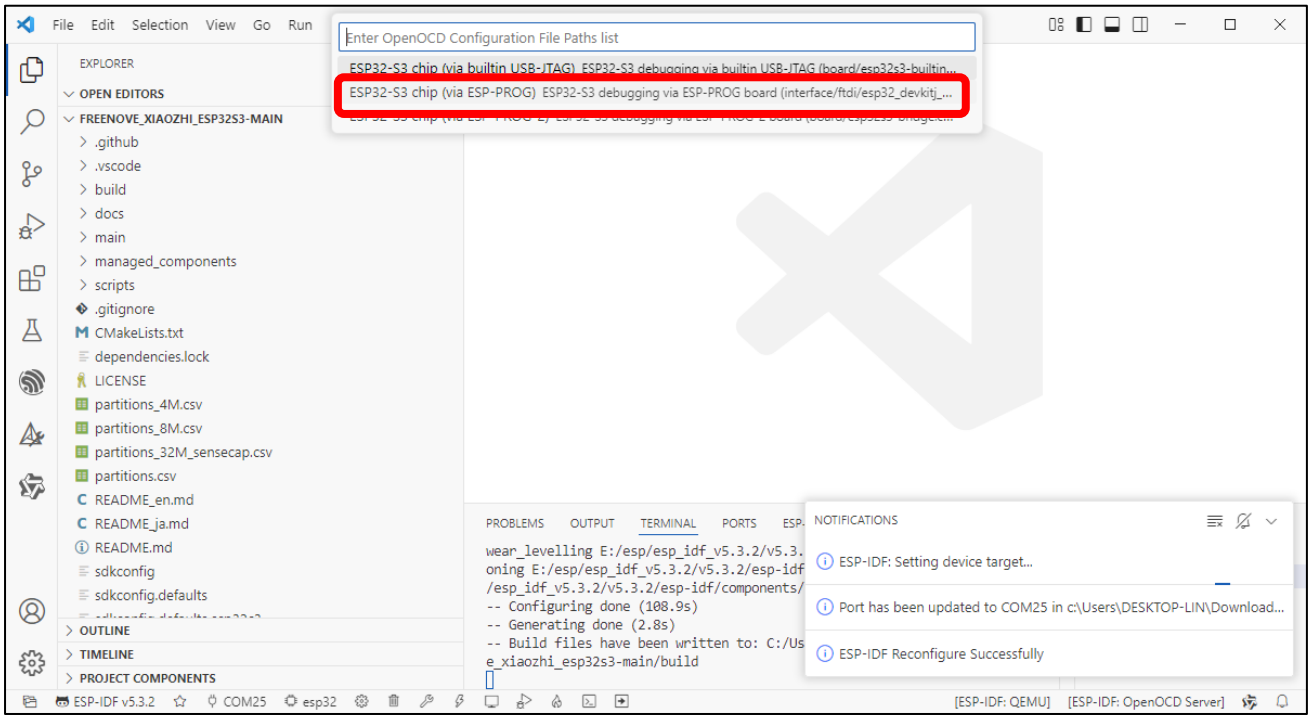
Click on 'COMx' in the bottom-left corner to display all available COM ports on your computer. Locate and select the entry labeled 'ESP32-S3'.



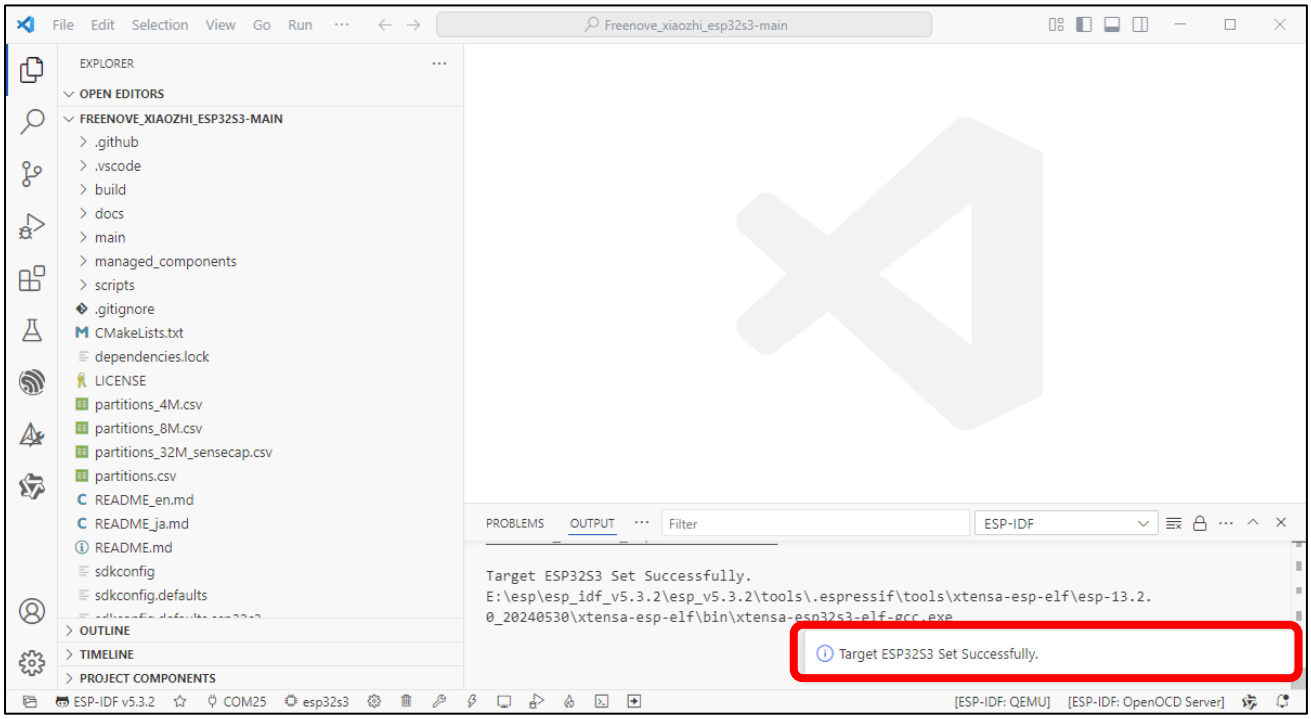
Click the 'ESP32' button in the bottom-left corner to display all available ESP32 models, then select 'ESP32S3' from the list."



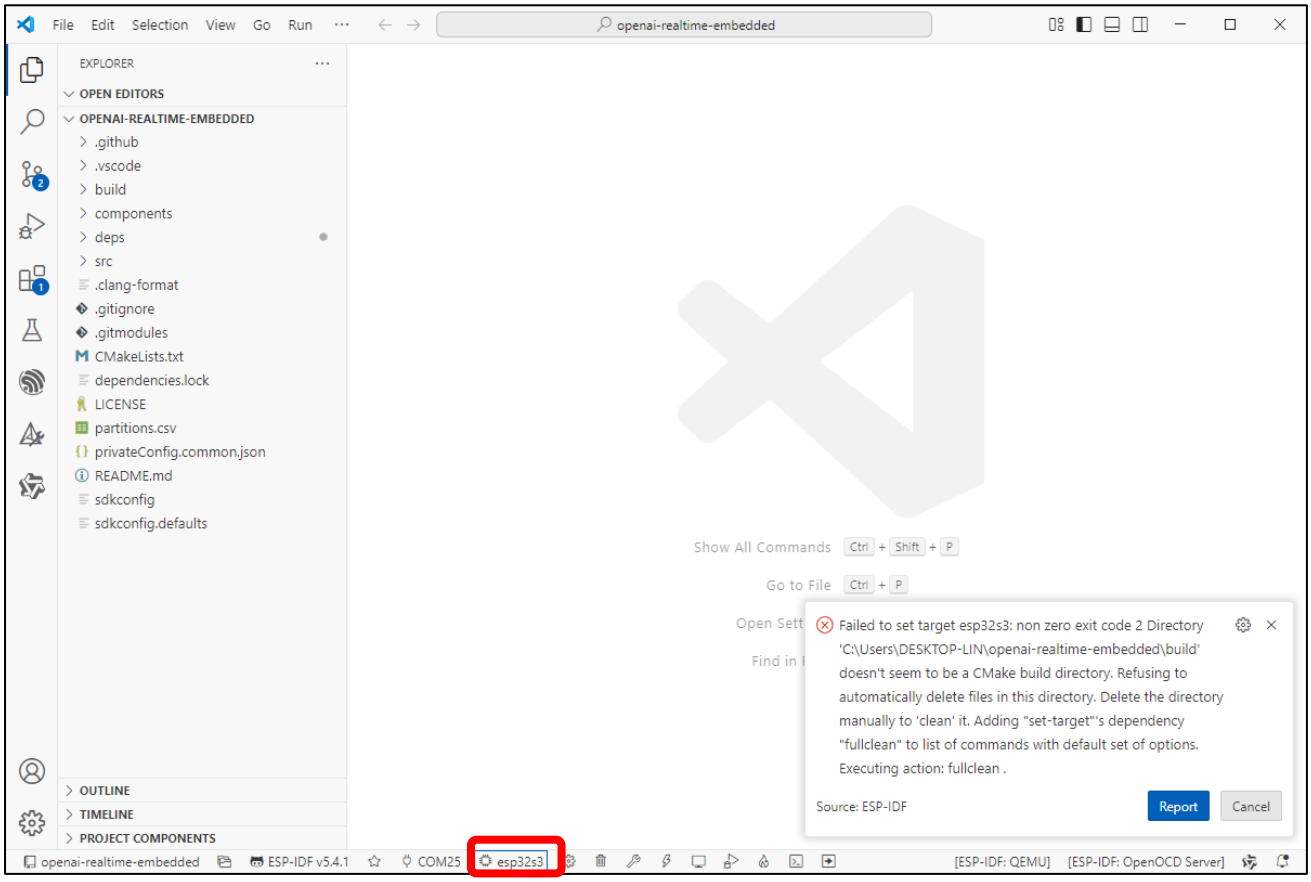
From the new selection menu, choose 'ESP32-S3 Chip (via ESP-PROG) - ESP32-S3 debugging via ESP-PROG Board...'



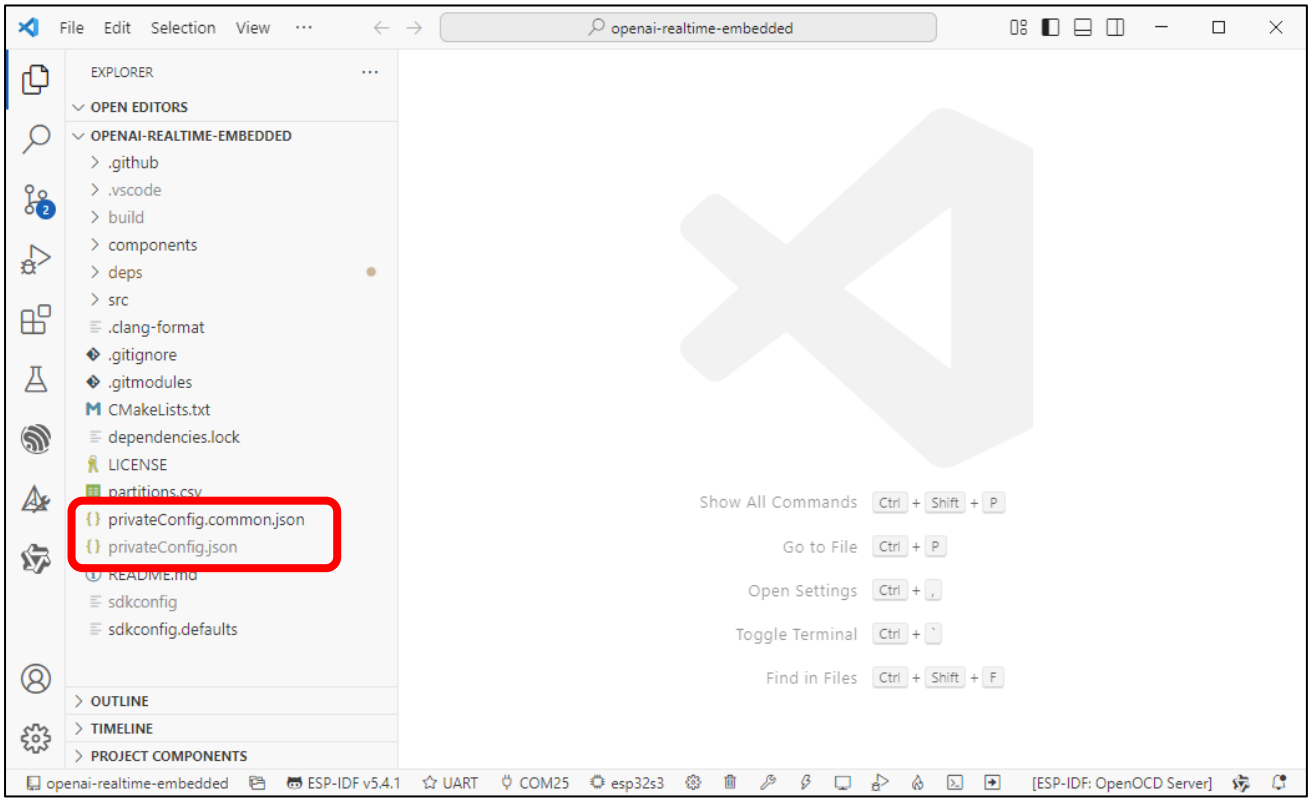
Wait until it shows “Target ESP32S3 Set Successfully“” at the bottom right.



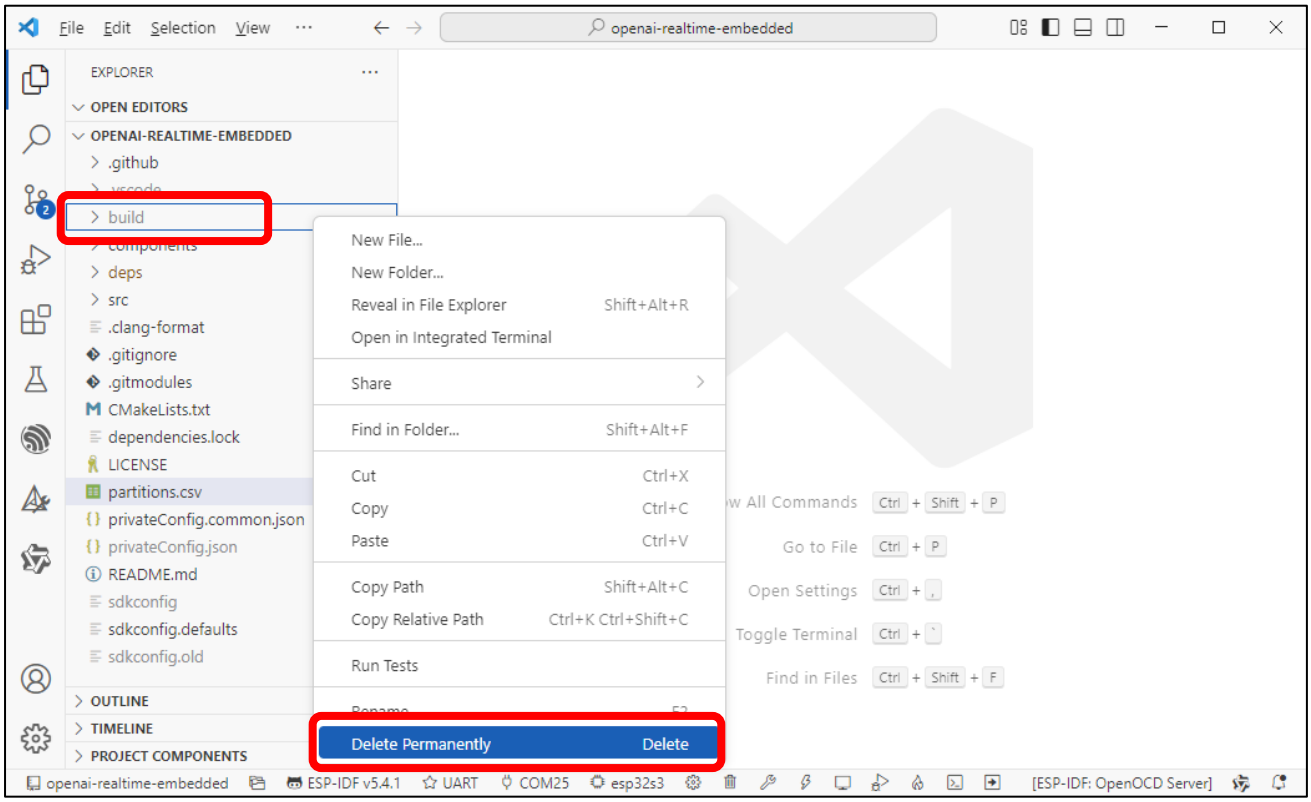
It if fails as shown in the figure below,



the most possible cause is that you do not copy **privateConfig.common.json** and rename it as **privateConfig.json**. Please operate this step again.



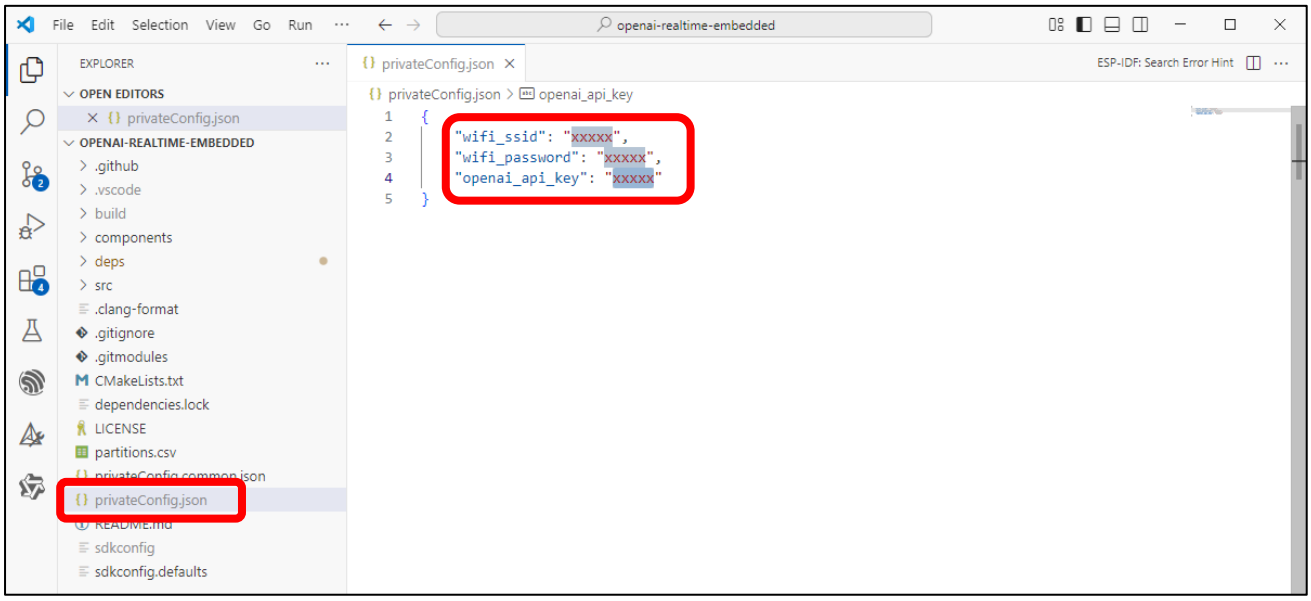
If it still fails, please delete the build folder, restart the software, and Set Espressif DeviceTarget again.



If it continues to fail, it indicates that the ESP-IDF is not successfully installed. Please install it again [ESP-IDF V5.4.1]

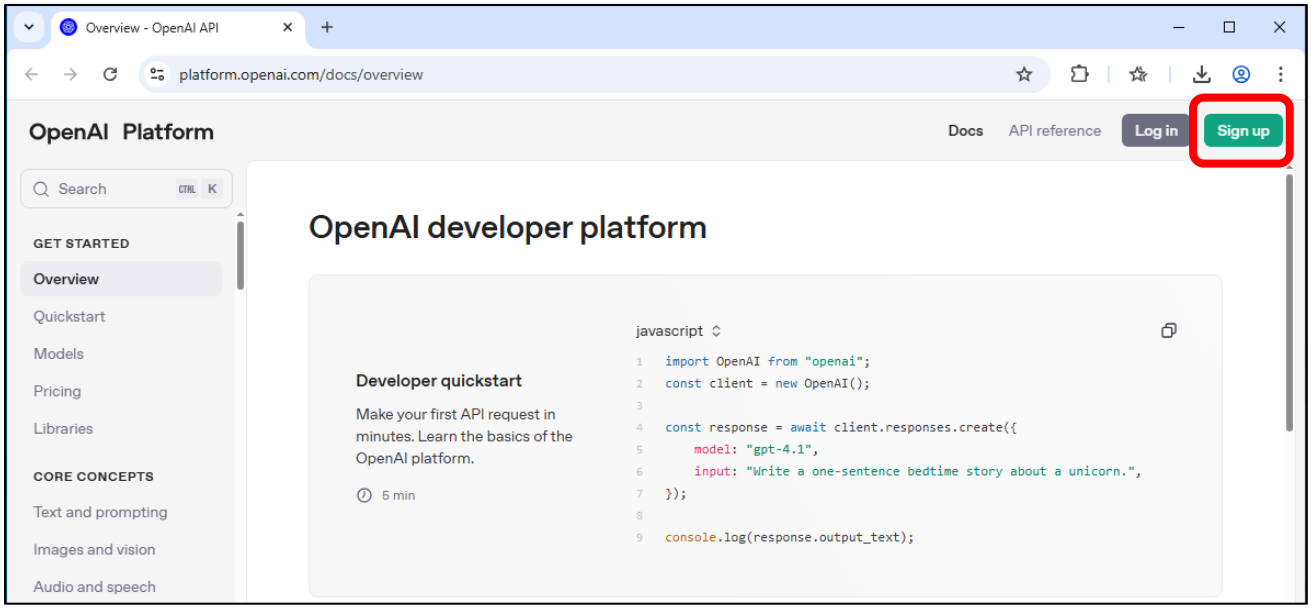
Registering Open API Keys

Click "**privateConfig.json**". On the right side, enter your router name, password, and OpenAI API key. If you don't have an OpenAI API key, you'll need to register and purchase one. Currently, OpenAI does not offer free services.

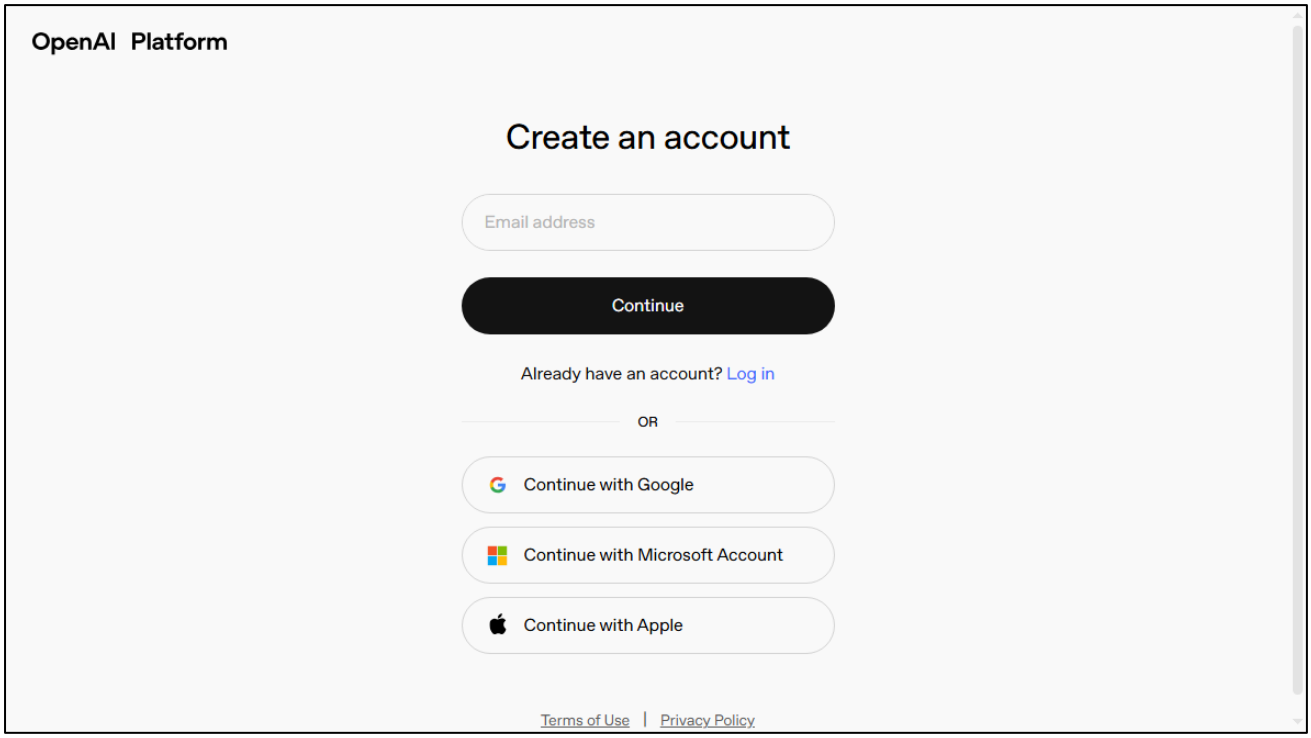


If you don't have an OpenAI API key yet, please visit https://platform.openai.com/docs/overview to sign up and purchase one.

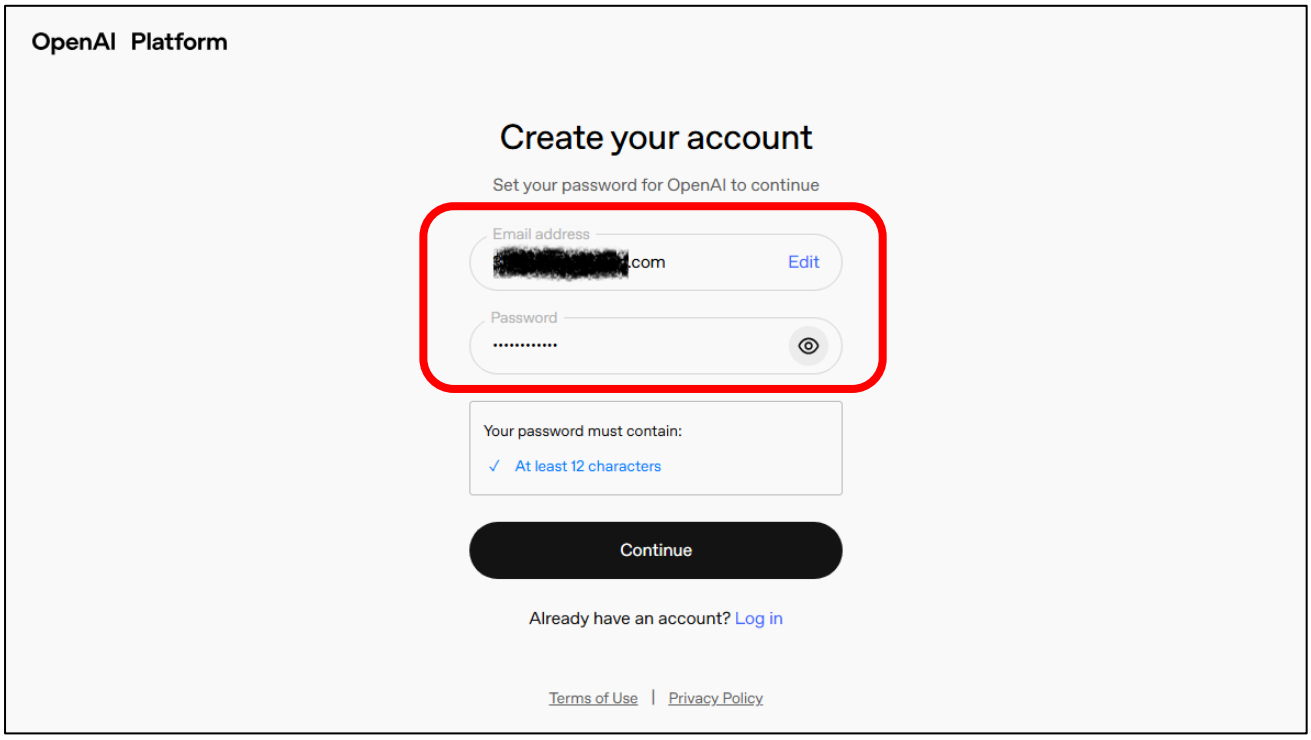
You'll need to create an account. Click "Sign up" in the top-right corner.



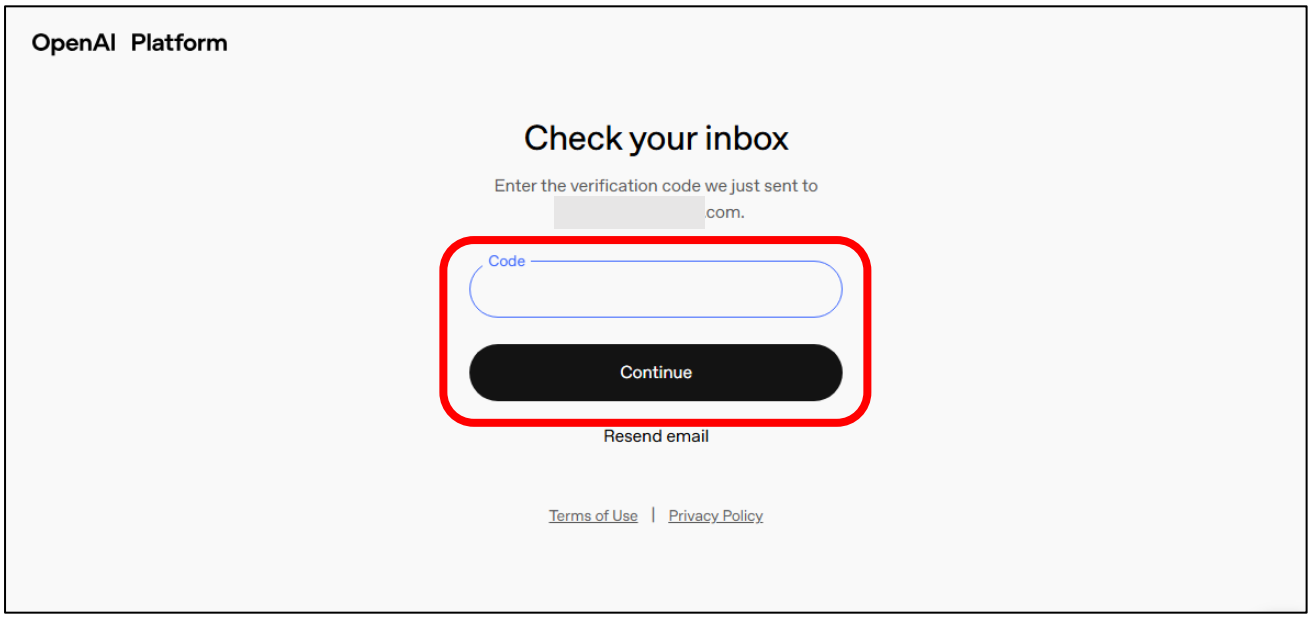
Create the account in any way you prefer.



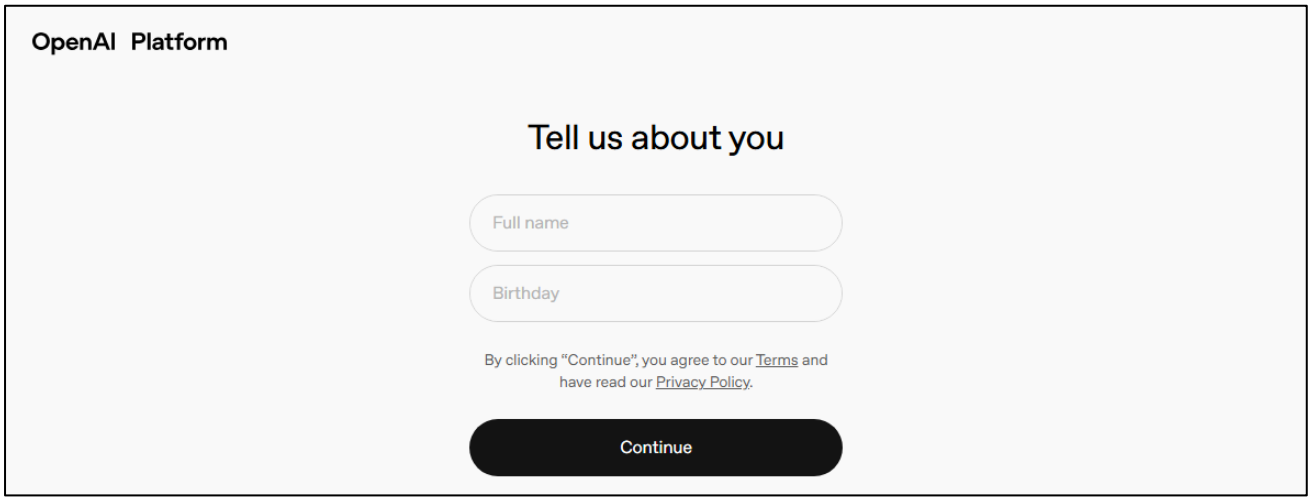
Here we take Email address as an example. Input the email address and password, and click Continue.



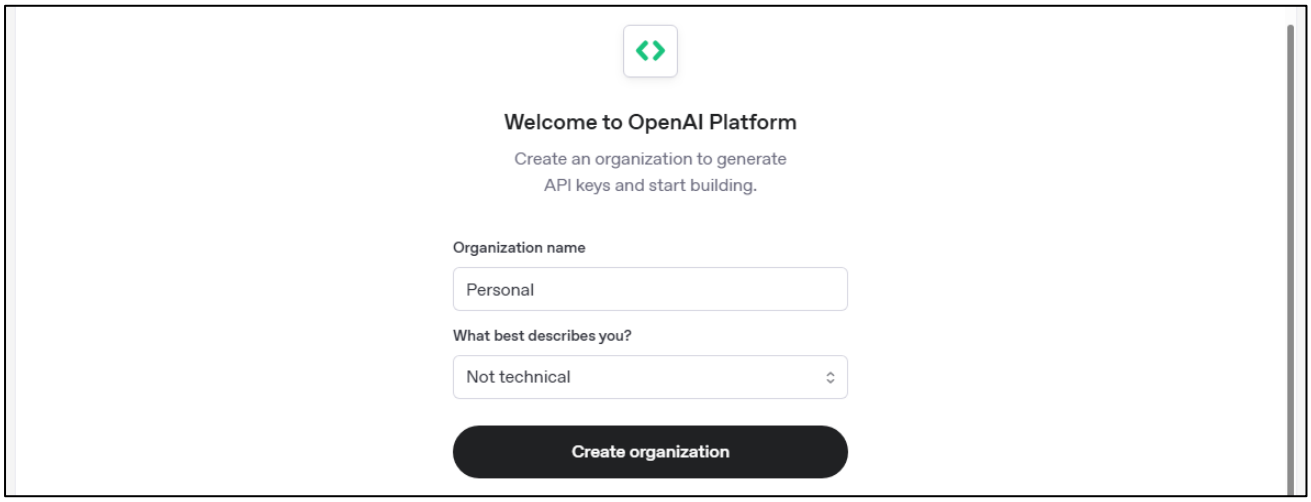
OpenAI Platform will send a verification code to your email address. Input the code to the bar, click Continue.



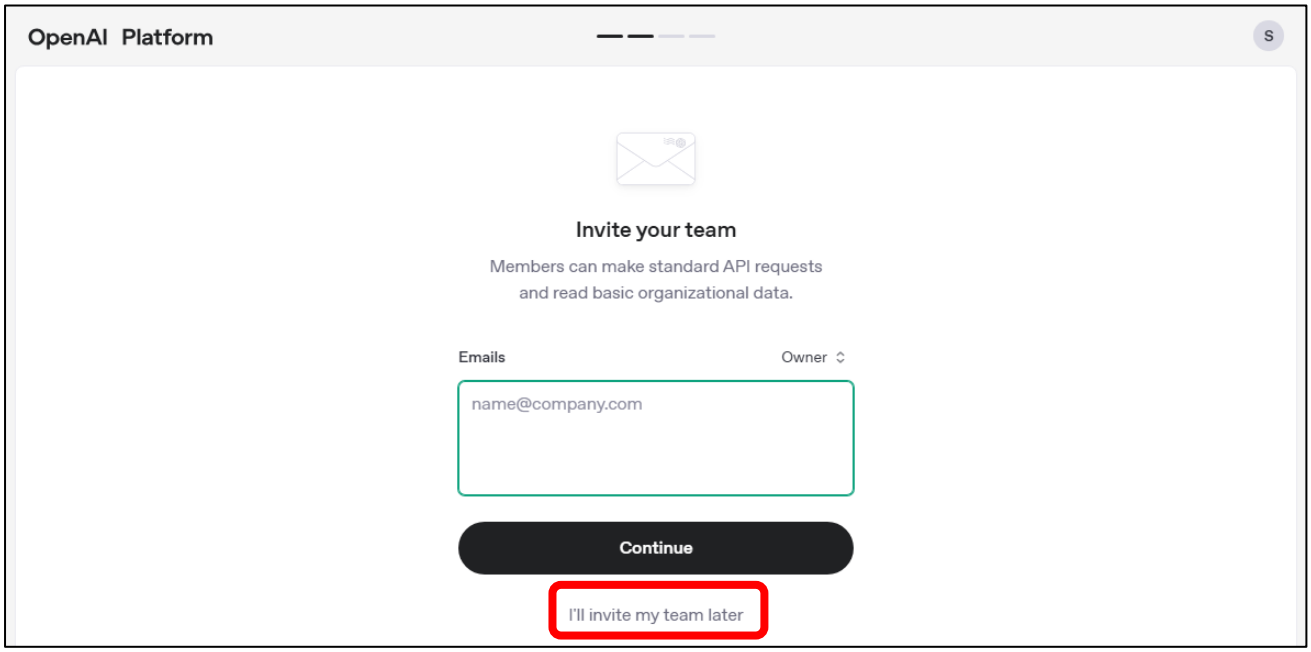
Complete your personal information.



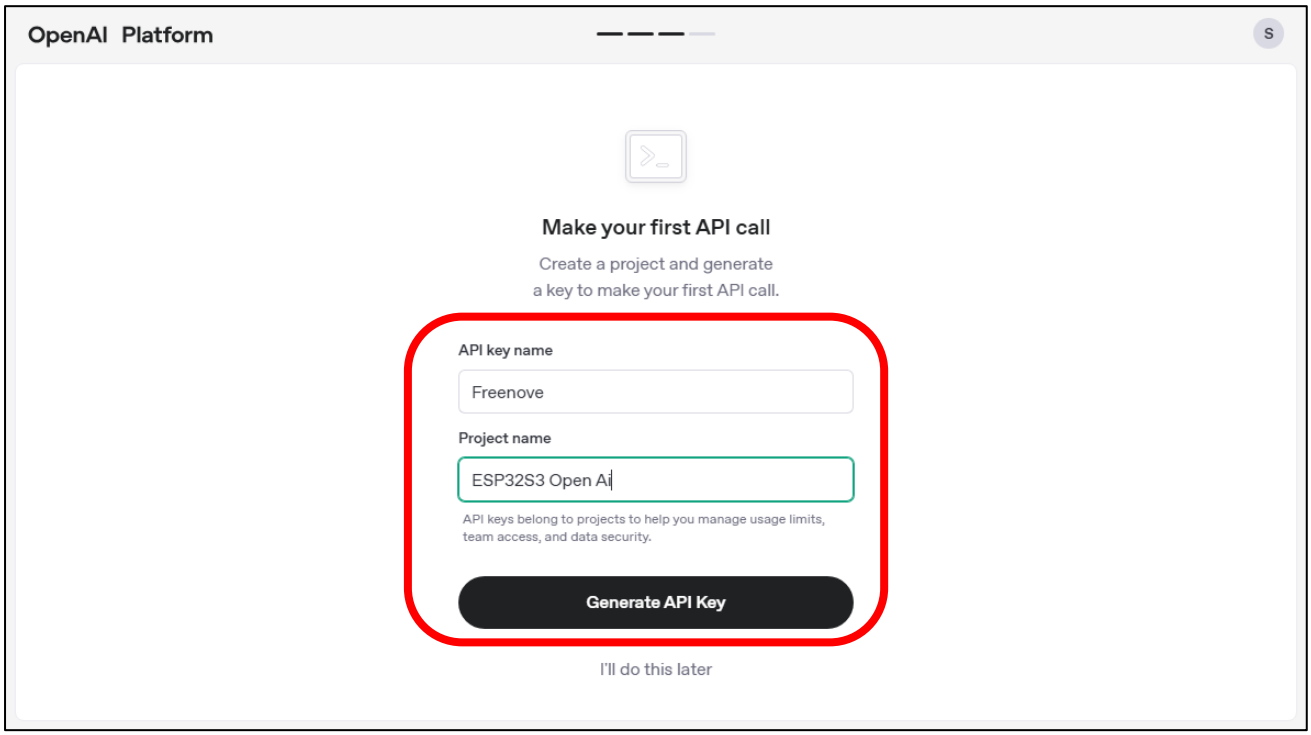
You can either fill the Organization Name or not, then click Create organization.



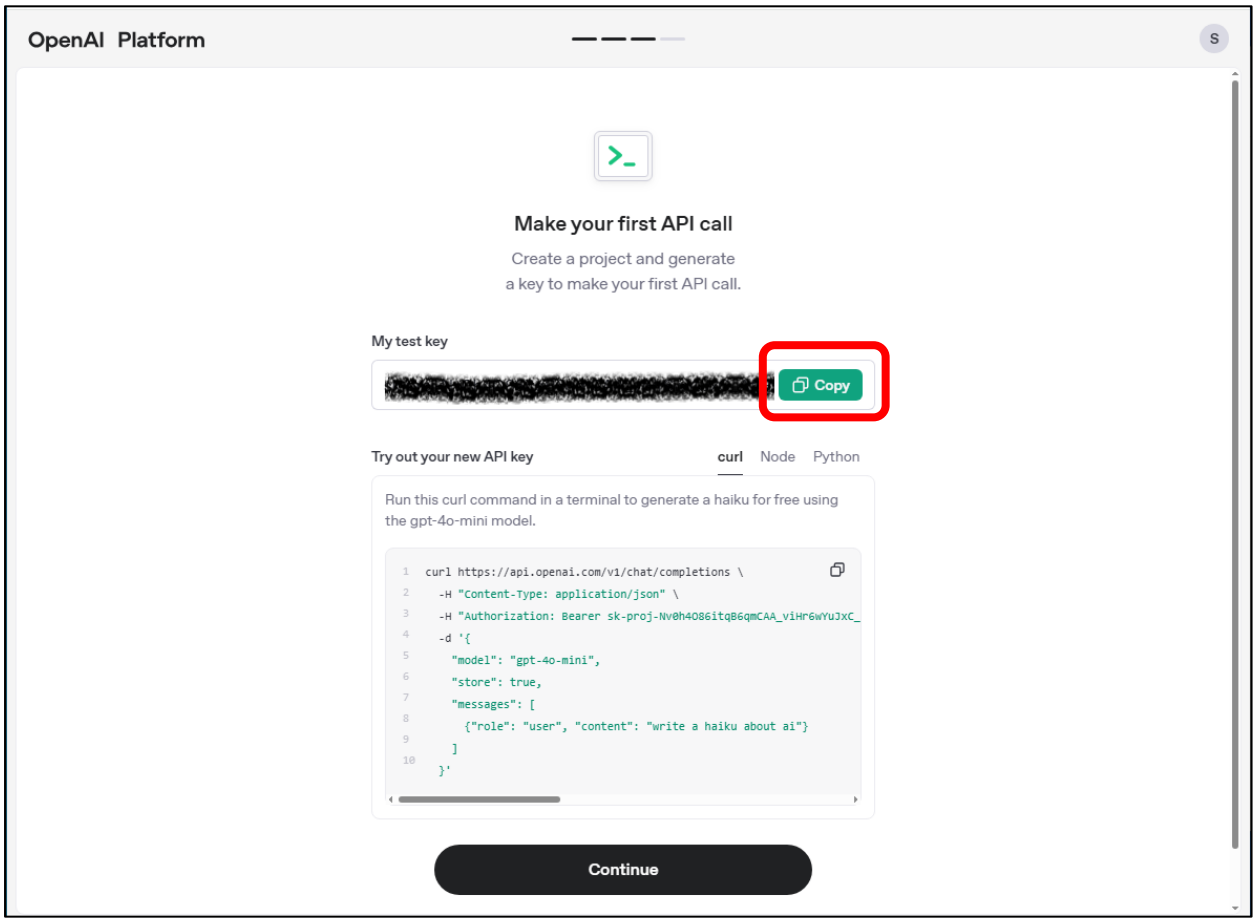
You can click "I'll invite my team later".



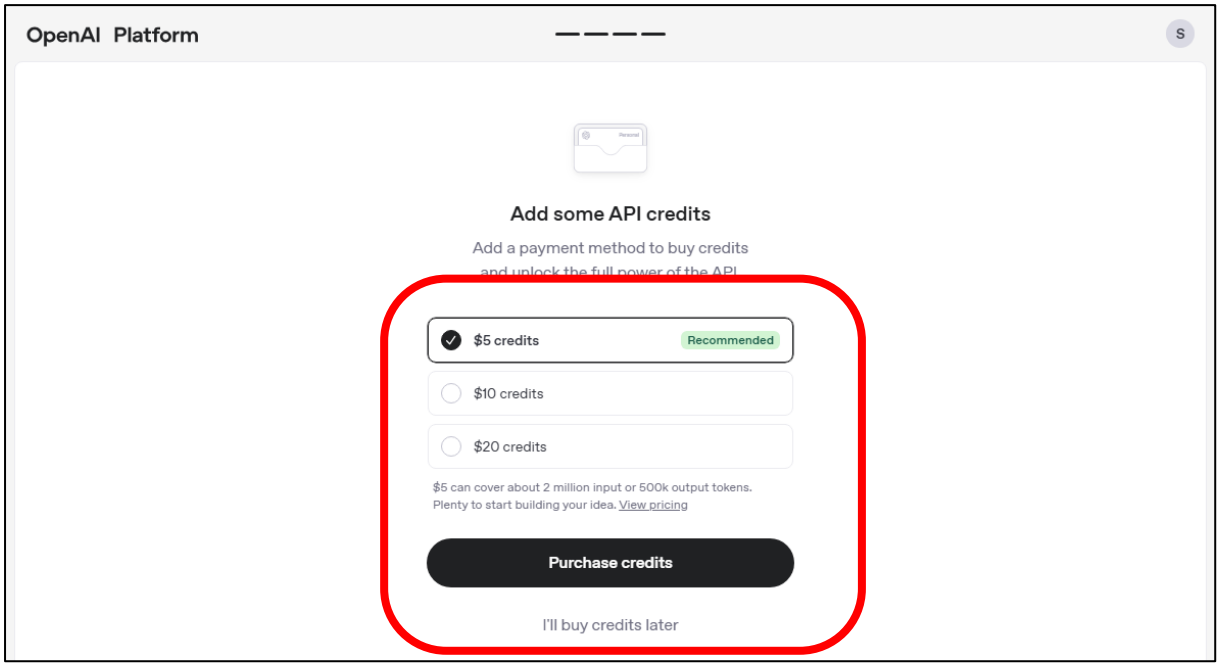
Name it whatever you like, and then click "Generate API Key".



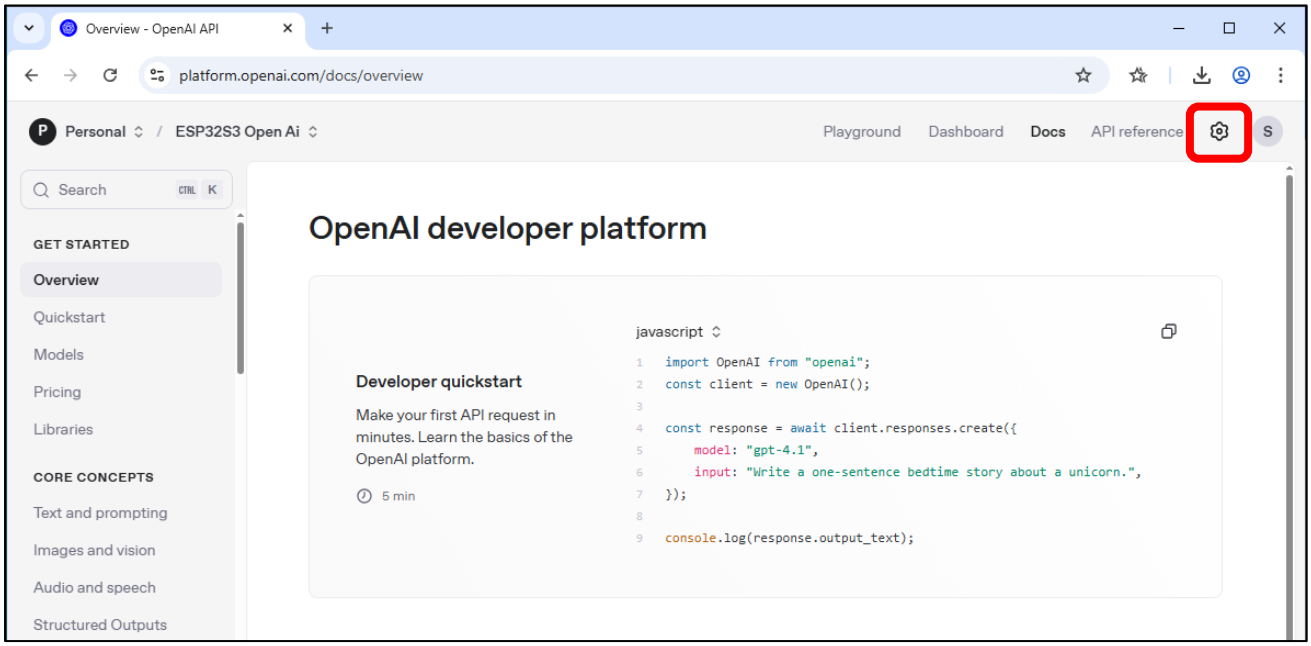
Click Copy and paste your "My test key" somewhere to save it. If you click "Continue", you will no longer be able to view the key again!
We recommend creating a separate document to store your key securely and prevent loss.



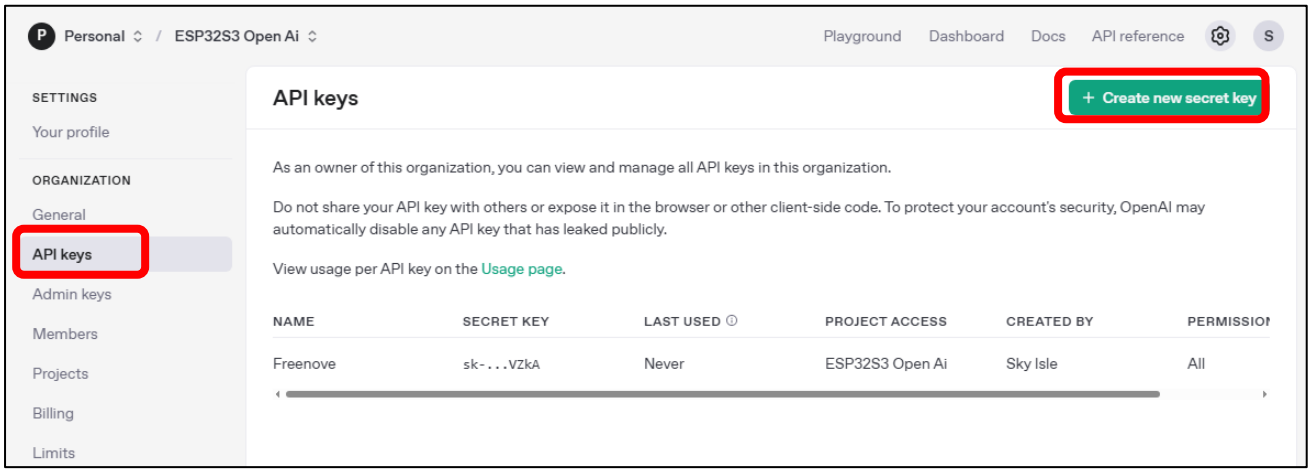
OpenAI operates on a paid service model. You need to purchase for credits to ensure uninterrupted access to its features.



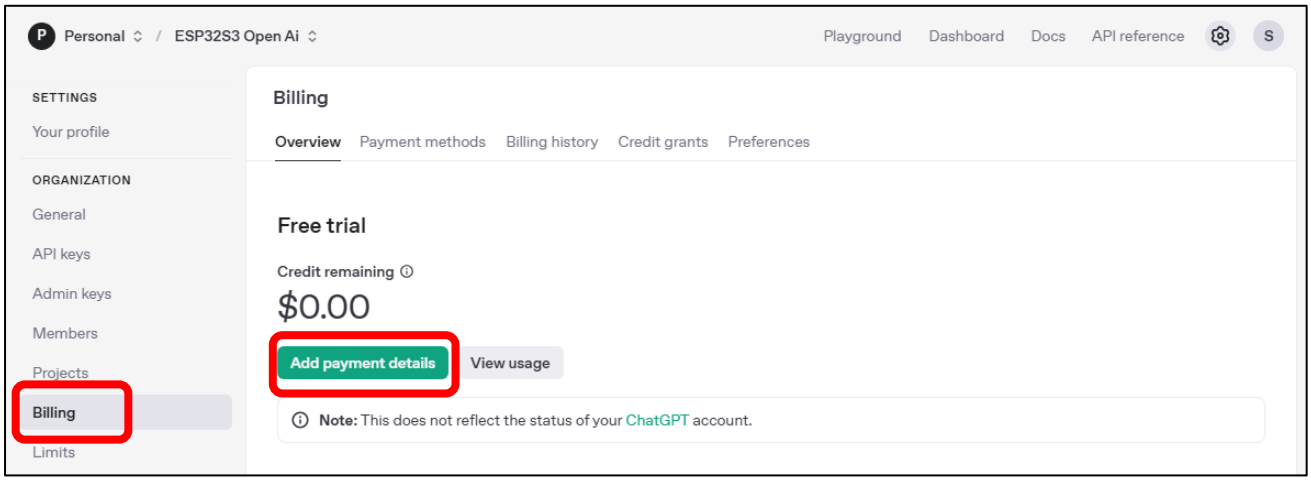
After completion, the interface will appear as shown below. Click the settings icon in the upper right corner.



You can click "API Keys" to manage your OpenAI keys.

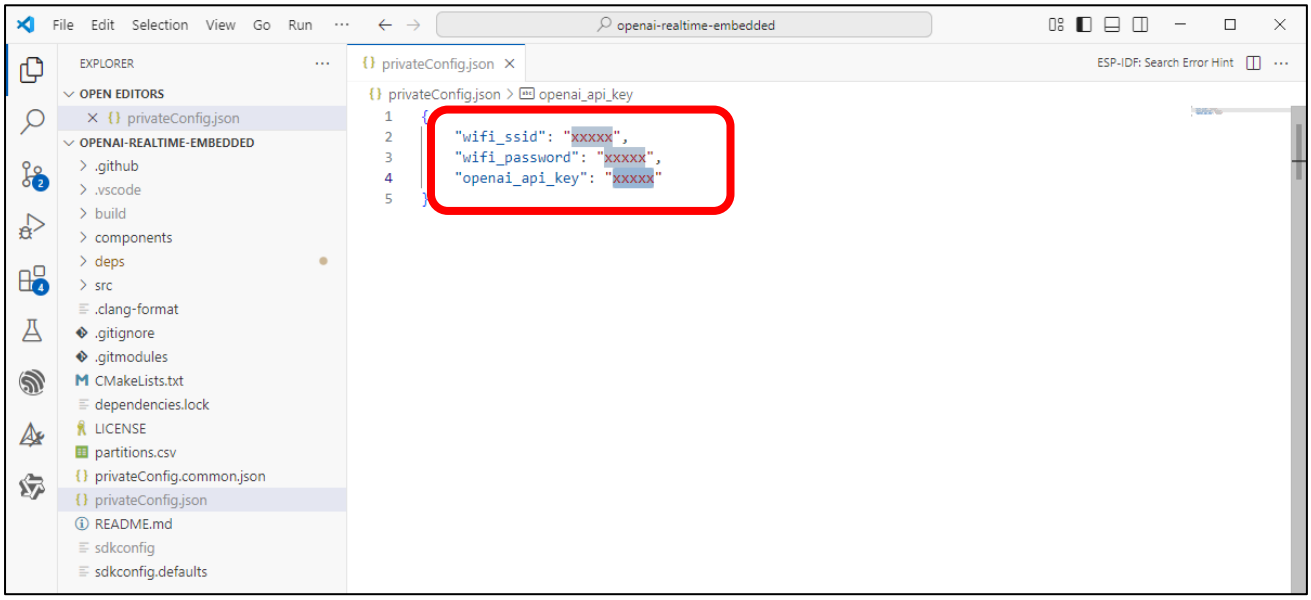


Click "Billing" to check your available balance, or click "Add Payment Details" to top up funds.



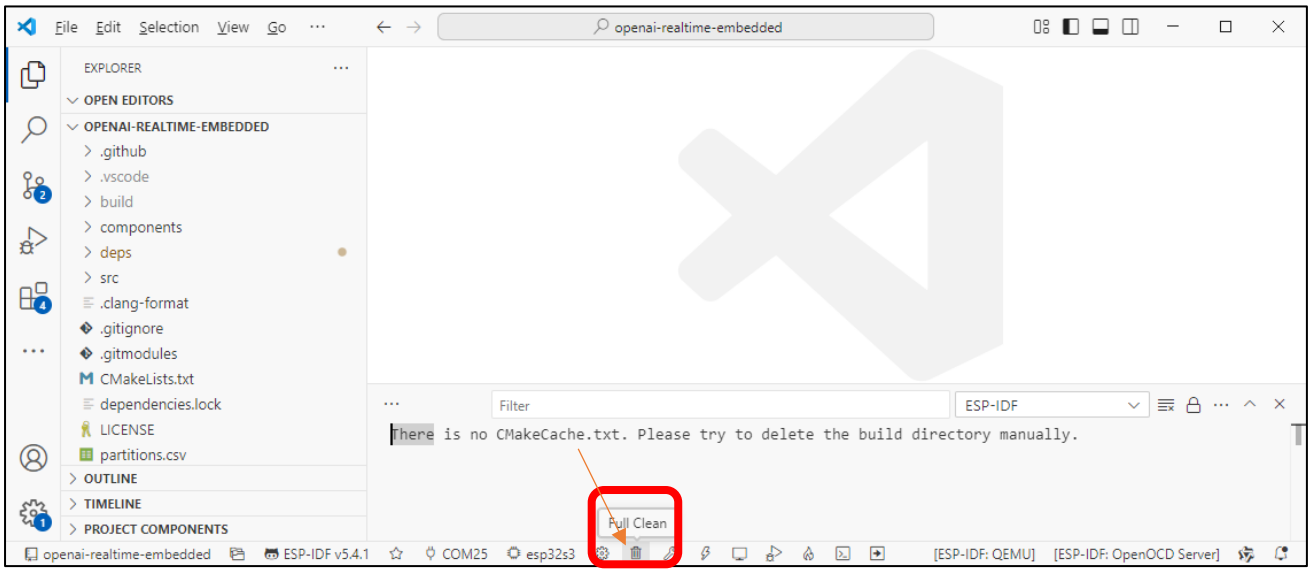
Now you have your own working OpenAI API keys. Feel free to explore more features on the website.

Next, let's return to the code. Enter your router name, password, and OpenAI API key in their respective fields, then save the file.

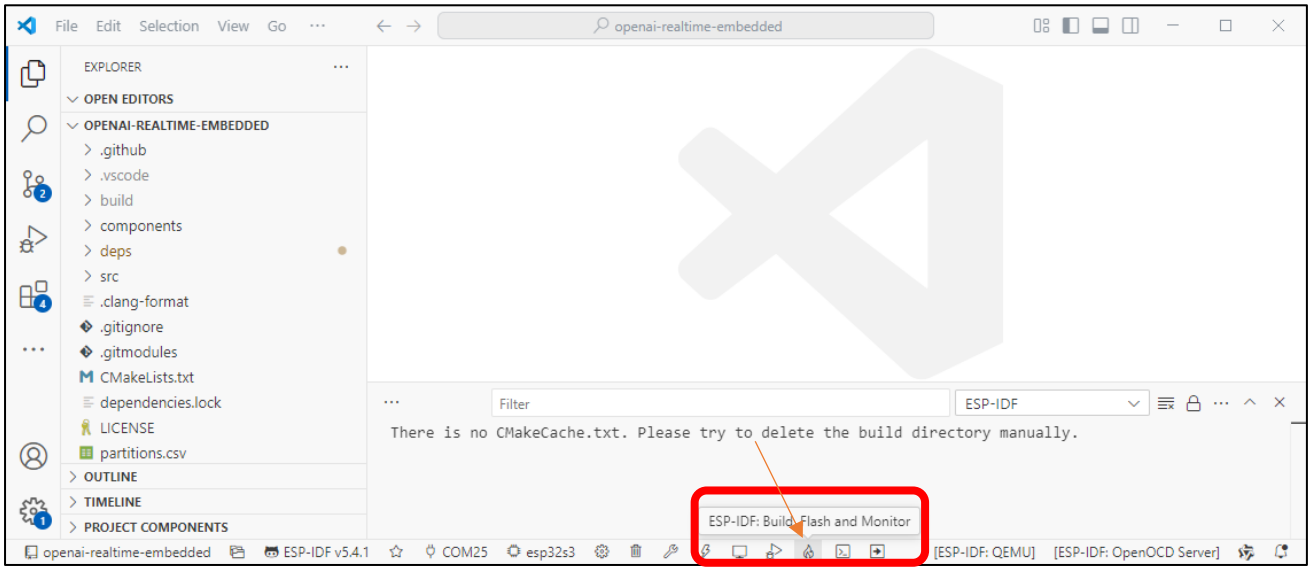


Code Compilation and Uploading

After ensuring everything is properly configured, let's start compiling the code.
Click "Full Clean" below to clear all previous compilation data.

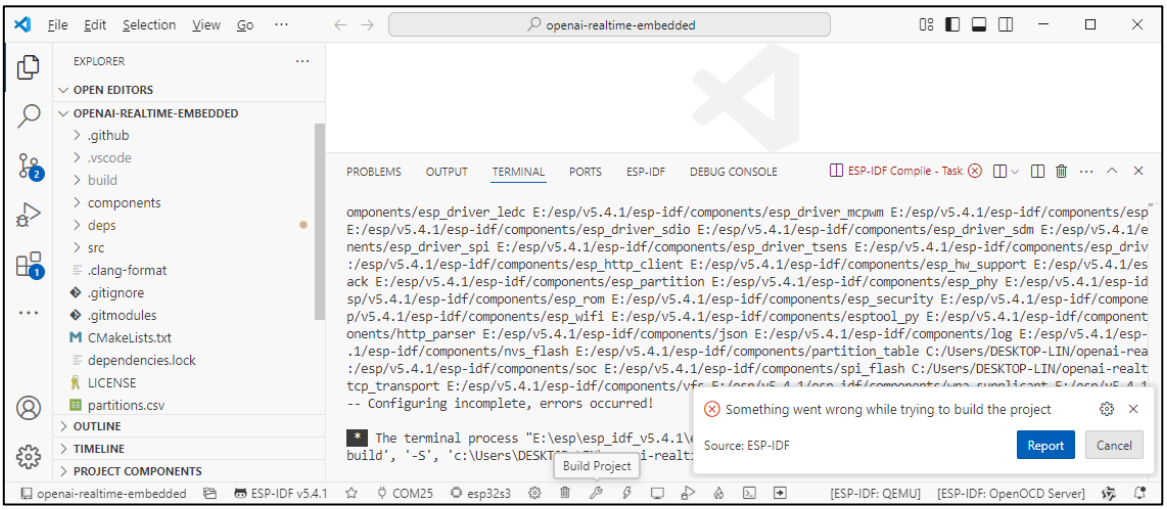


Click "ESP-IDF: Build, Flash and Monitor" below. It will compile the code, upload it to the ESP32-S3, and open the serial monitor.

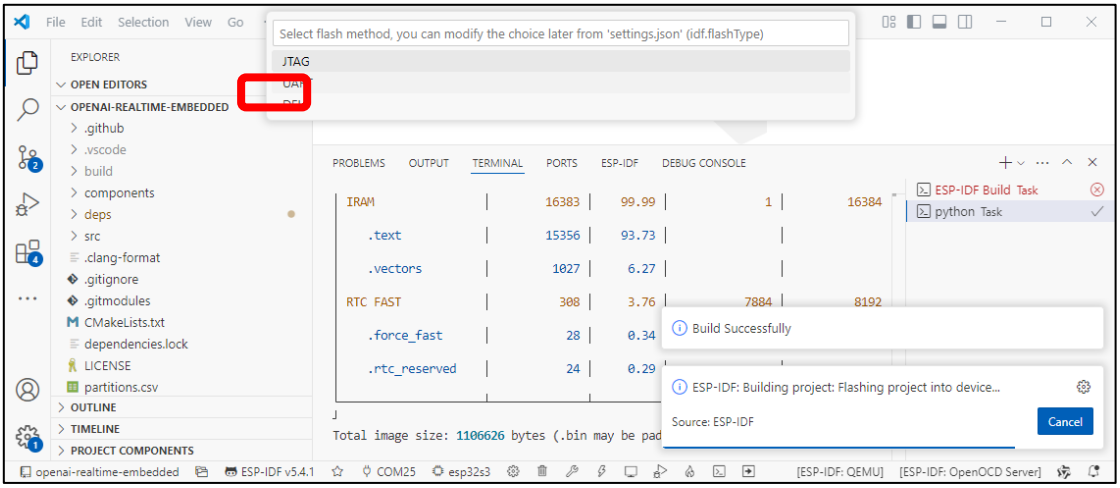


The first compilation may take longer—please ensure a stable internet connection and wait patiently.

If you encounter the following error during compilation, it's typically caused by issues with the ESP-IDF toolchain. This often occurs due to network problems that prevented some toolchain components from installing properly. Please reinstall ESP-IDF. You can refer to ESP-IDF V5.4.1.



When the code compilation is complete and you upload it for the first time, you will see the prompt below. Please select "UART".



After the upload completes, the serial monitor will automatically open and establish an internet connection to access OpenAI. You can now start chatting via the ESP32-S3.