Contents

AI Voice Assistant Based on XiaoZhi AI

This project applies the Freenove ESP32 S3 Display to implement an AI voice assistant, which requires a certain level of programming proficiency as well as familiarity with ESP-IDF and open-source large models.

About the Project

This voice assistant project (https://github.com/Freenove/xiaozhi-esp32) is derived from the open-source project (https://github.com/78/xiaozhi-esp32). It enables the invocation of most mainstream large language models (LLMs) on embedded devices and achieves voice conversation functionality through multiple services, including Voice Activity Detection (VAD), Automatic Speech Recognition (ASR), Speech-to-Text (STT), Text-to-Speech (TTS), Memory Storage, and Intent Recognition. Freenove has adapted this project for its Media Kit product. This article will explain how to run the project on the Media Kit.

There are two ways to run this project – online or offline.

1. **Online**: Connected to the xiaozhi.me server, currently available for free trial to individual users.
2. **Offline**: All the aforementioned services (VAD, ASR, STT, TTS, Memory, Intent Recognition, etc.) must be deployed locally on a personal computer. The user experience depends entirely on the selected models and the performance of the local machine. The local server project (https://github.com/Freenove/xiaozhi-esp32-server) is derived from the open-source project (https://github.com/xinnan-tech/xiaozhi-esp32-server).

For users who prefer AI assistants, we recommend using the online version.
For developer-oriented users, you can try deploying the offline version to gain a deeper understanding of the various services required for an AI assistant. However, it's important to note that personal computers may struggle to run all these services simultaneously—especially the core LLM (Large Language Model) service—which could result in a poor AI assistant experience. Therefore, the offline version is primarily valuable for learning and research purposes.

Cautions

- **Project Copyright**:
  - Voice Assistant Project: Originally developed by "Xiage", this project was forked and adapted by Freenove for the Media Kit, released under MIT License.
  - Local Server Project: Originally created by "xinnan-tech", this project was similarly forked and adapted by Freenove for Media Kit integration, licensed under MIT License.
- **Supprted Countries and Regions:**
  - **Online Version**:
    Service availability is determined by xiaozhi.me server coverage, which may exclude certain regions. For current supported areas, please refer to: https://xiaozhi.me/login?redirect=/console/agents;
    User experience is directly affected by server connectivity quality. Poor network conditions to xiaozhi.me servers may degrade performance.

- **Offline Version**:
  Fully location-independent, with deployment possible in all countries and regions without geographical restrictions.

- **Supported Languages:**
  - Online Version: Currently supports Mandarin Chinese, Cantonese, English, Japanese, Korean, and others. If you do not speak these languages, you may not be able to communicate effectively with XiaoZhi AI.
  - Offline Version: Depending on the ASR model you deploy. The default FunASR model only supports Mandarin Chinese, Cantonese Chinese, English, Japanese and Korean.
- **Pricing:**
  - Online Version: Currently, xiaozhi.me provides free services, but we cannot guarantee that the online server will remain free indefinitely.
  - Offline Version: Among the sub-services mentioned, some are paid while others are free—your choice determines the cost.
- **Seeking Help:**
  If you have followed the tutorial and still encounter issues, please contact us at **support@freenove.com**
  Note: Since the online service is provided by xiaozhi.me, if xiaozhi.me discontinues its service, we will also remove related documentation, tutorials, and code.

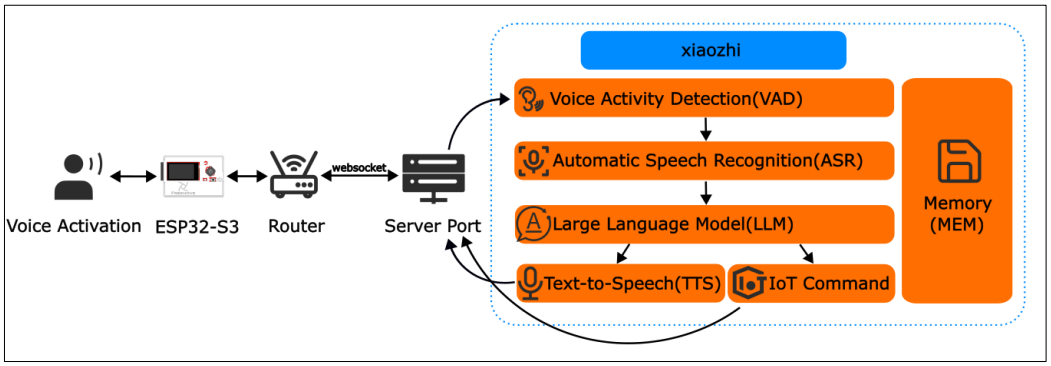
Disclaimer

This implementation is an adaptation of the open-source project available at https://github.com/78/xiaozhi-esp32, 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/78/xiaozhi-esp32/issues
2. Currently, XiaoZhi AI only supports Mandarin Chinese, Cantonese, English, Korean, and Japanese for speech recognition. Other languages are not yet supported.
3. The XiaoZhi server interface currently supports English, Chinese, and Japanese only. Additionally, mobile registration is only available for users in the following countries (see the table below). Users from other countries cannot register yet.

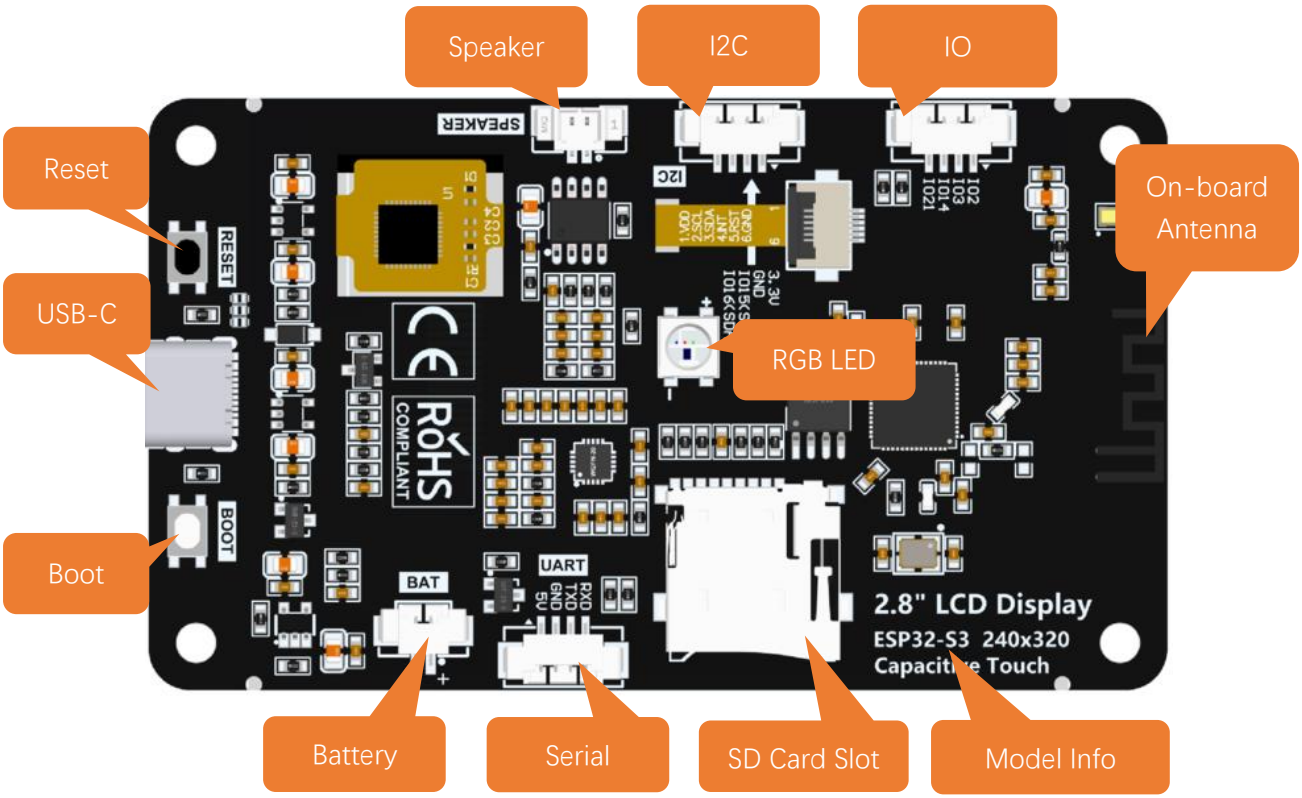
In this project, the ESP32-S3 communicates with XiaoZhi AI server through WebSocket protocol for data exchange.



Freenove ESP32 S3 Display

Top	Bottom
	

Hardware Interfaces



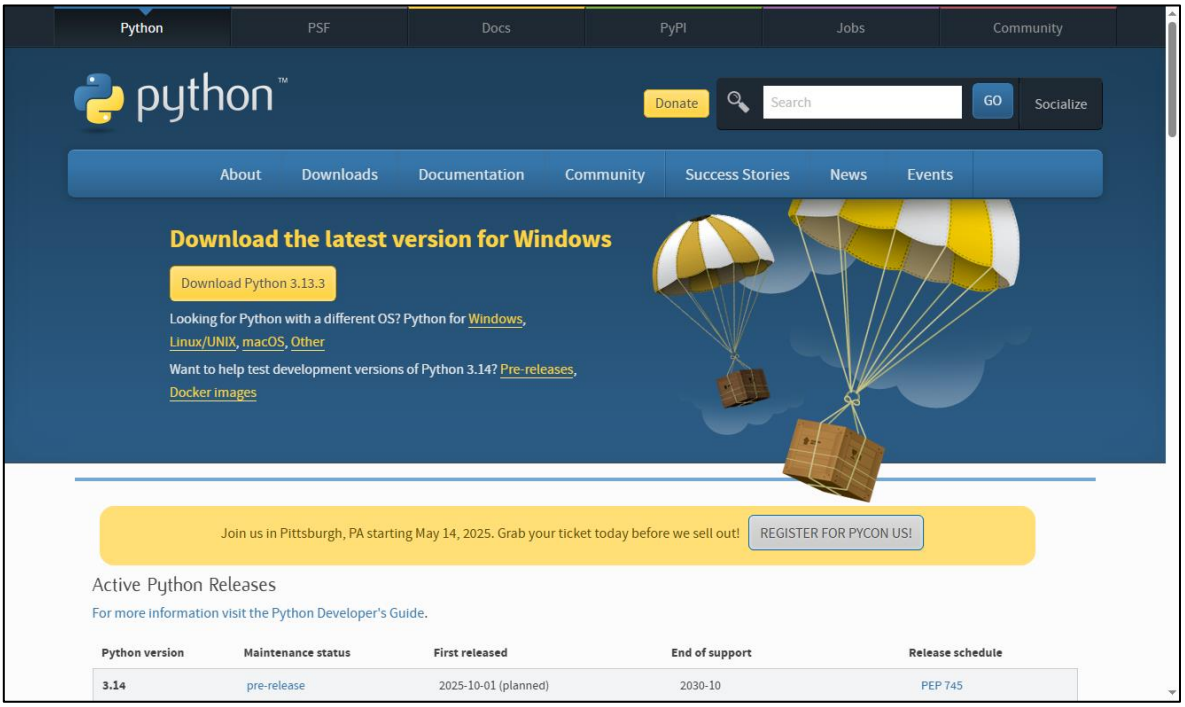
XiaoZhi AI Firmware

If your hardware does not yet have XiaoZhi firmware installed, you can follow the upcoming tutorial to re-flash the firmware onto the ESP32-S3-WROOM.
If your hardware already comes with XiaoZhi firmware pre-installed, you may skip this section.

Installing Python (Required)

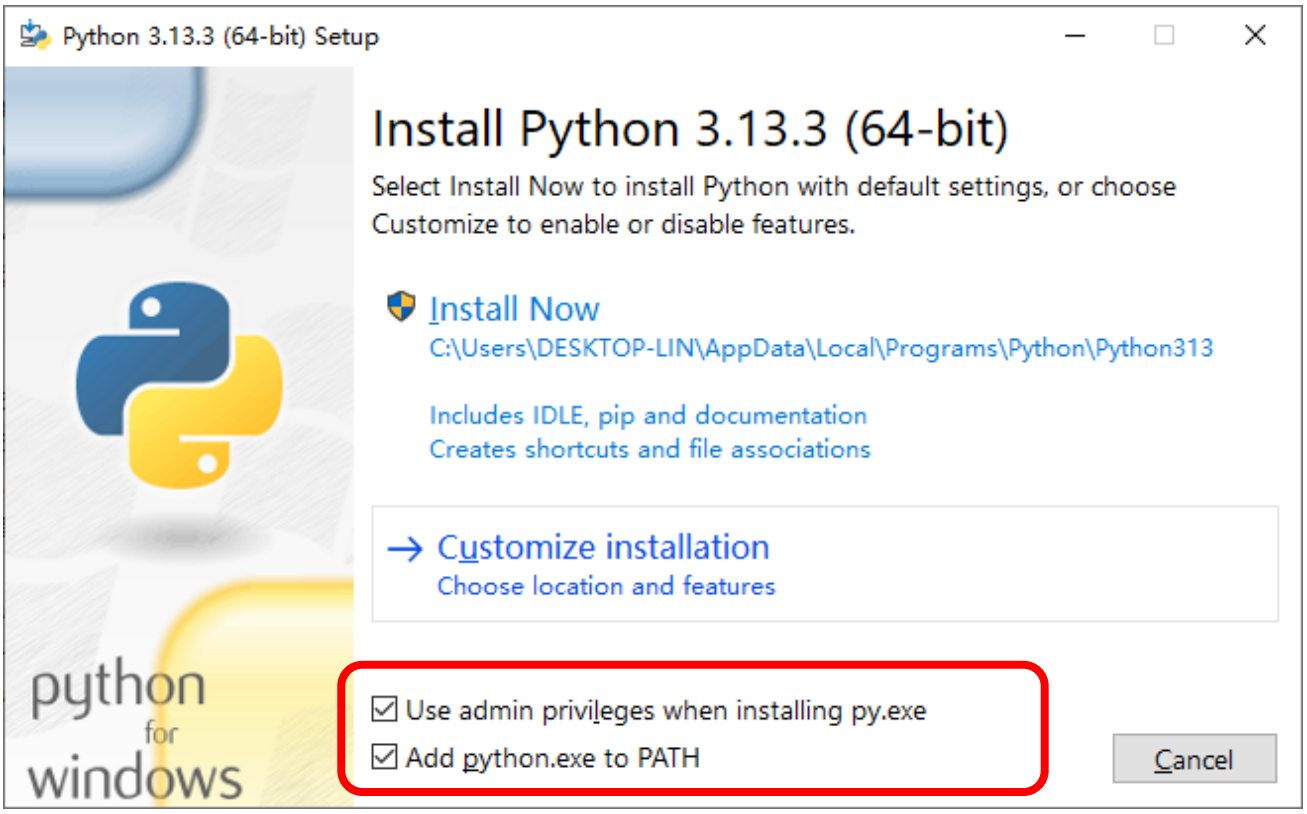
Windows

Download and install Python3 package.
https://www.python.org/downloads/windows/

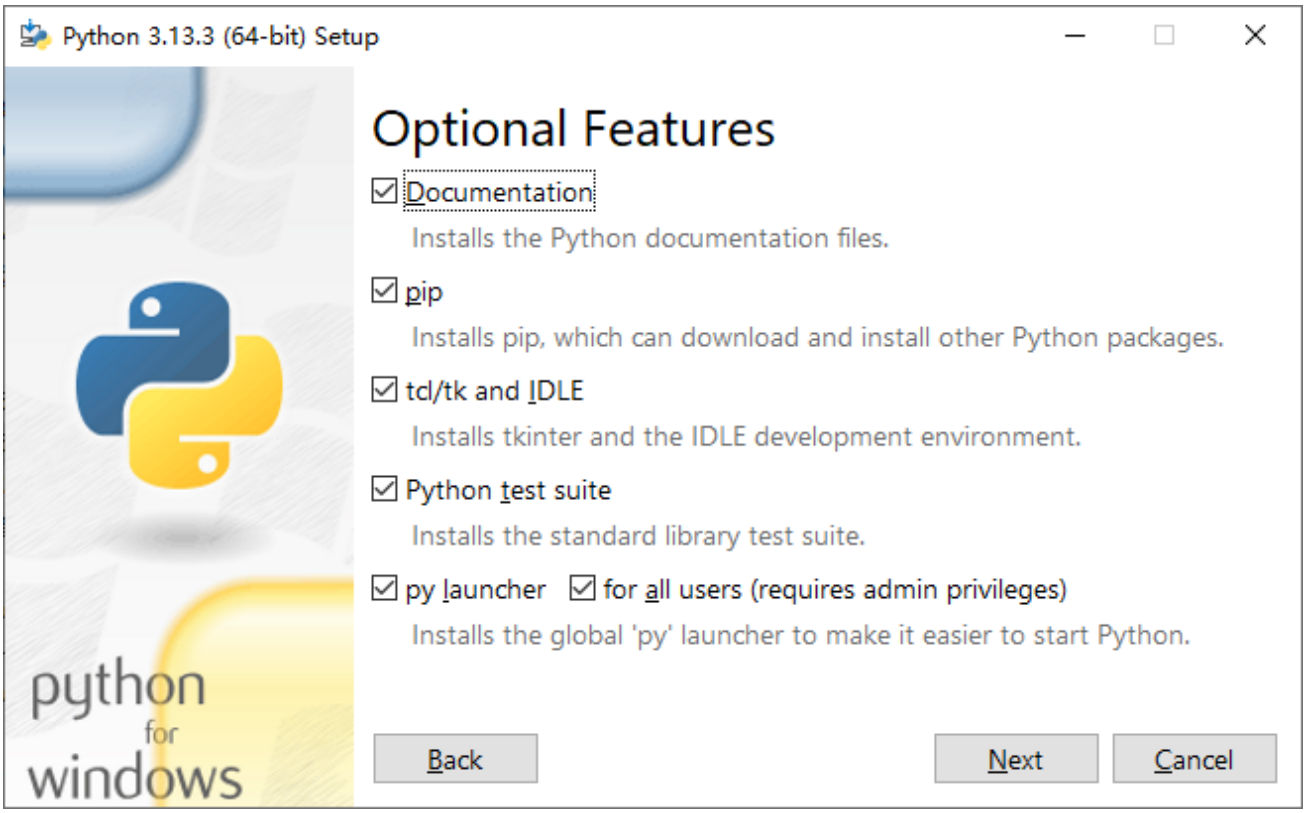


Click **Download Python 3.13.3**

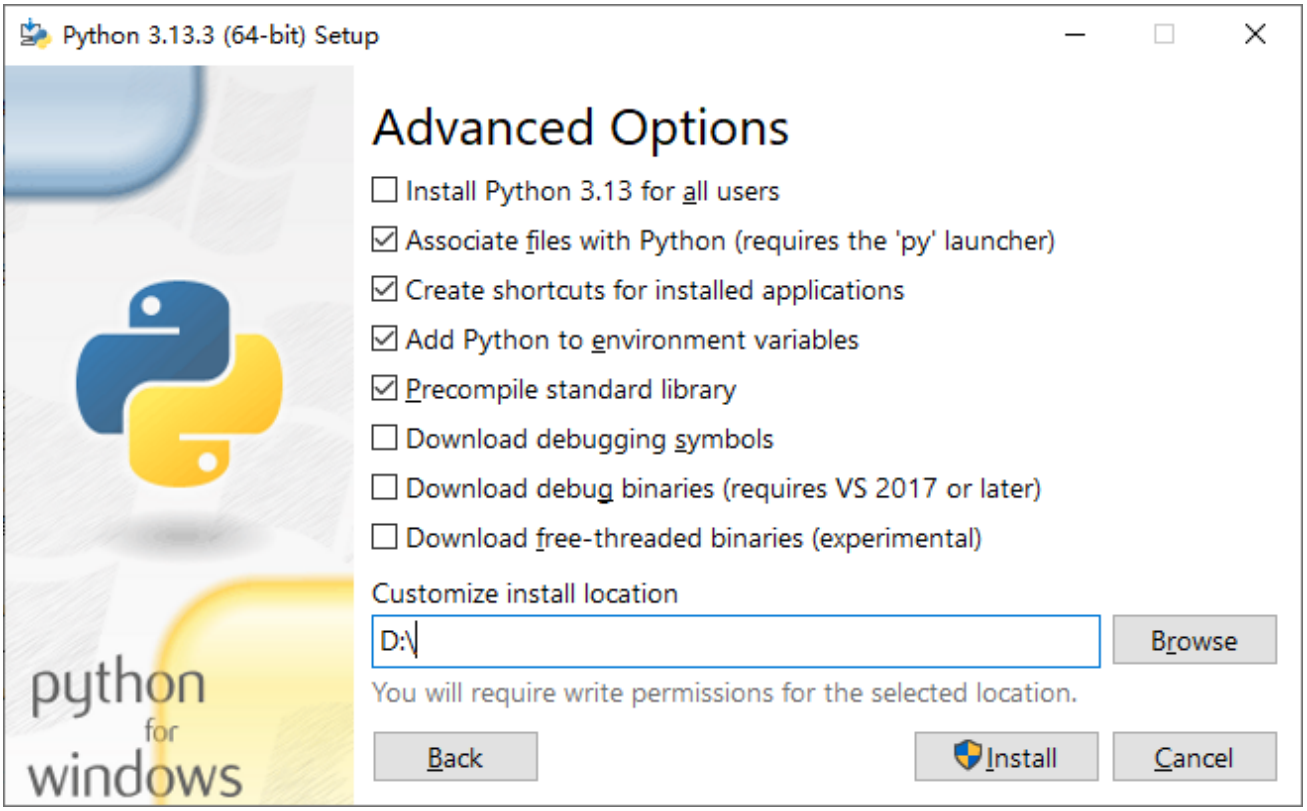
Please note that "Add Python 3.13 to PATH" MUST be check.



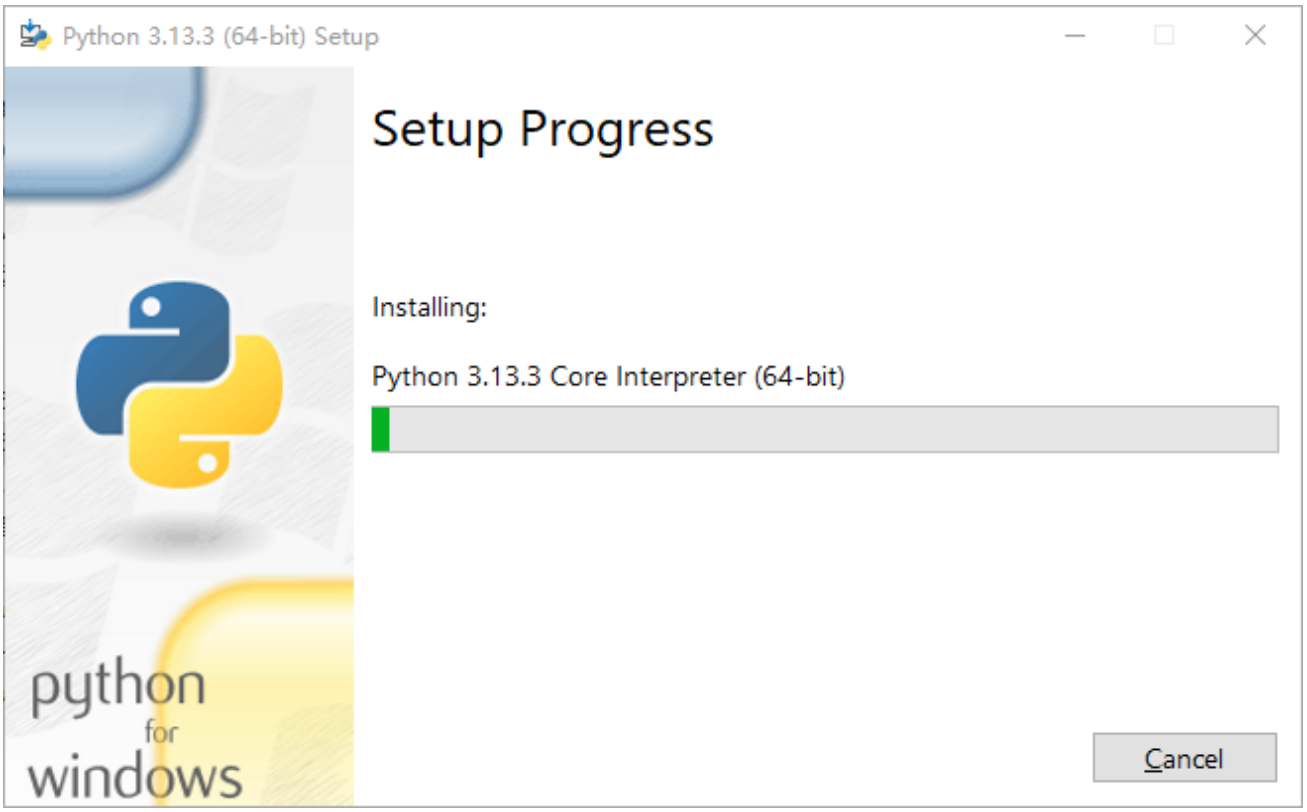
Check all the options and then click "Next".



Here you can select the installation path of Python. We install it at D drive. If you are a novice, you can select the default path.



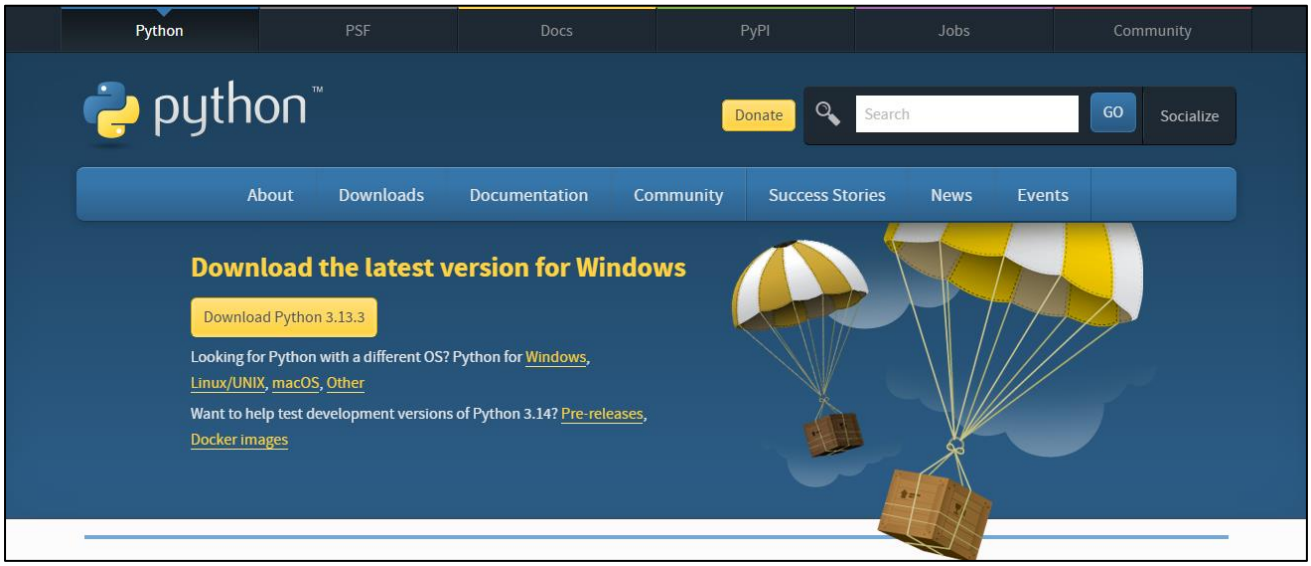
Wait for it to finish installing.



Now the installation is finished.

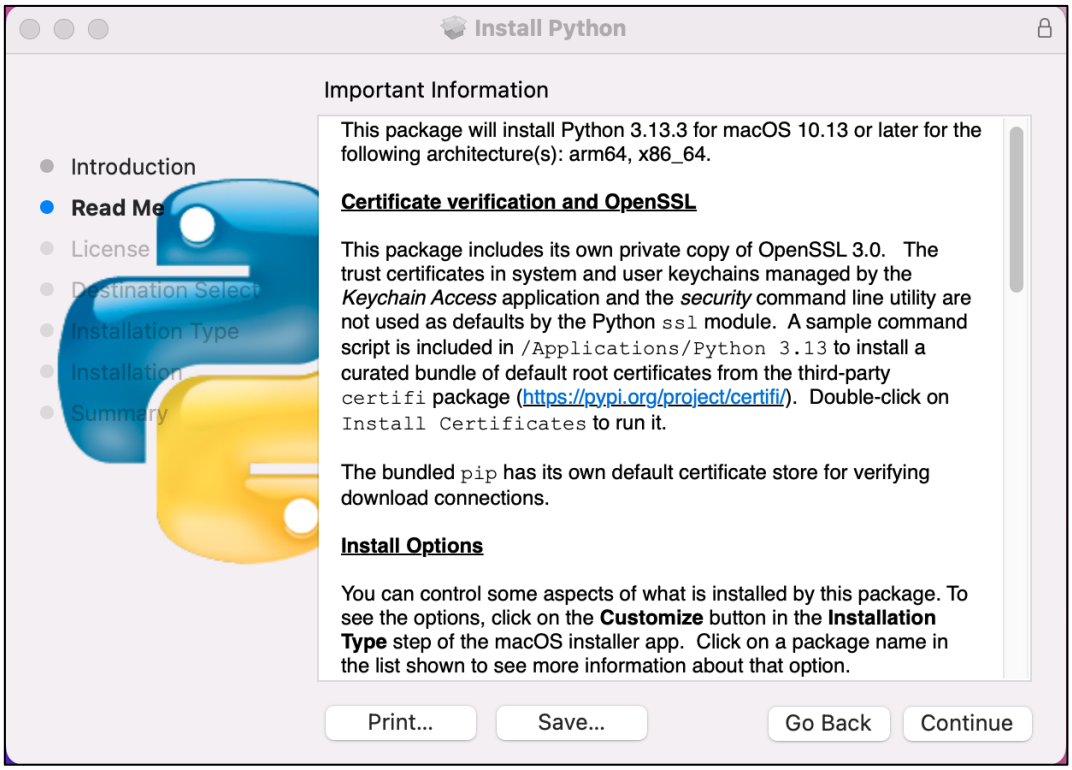
Mac

Download installation package, link: https://www.python.org/downloads/
Click Download Python 3.13.3

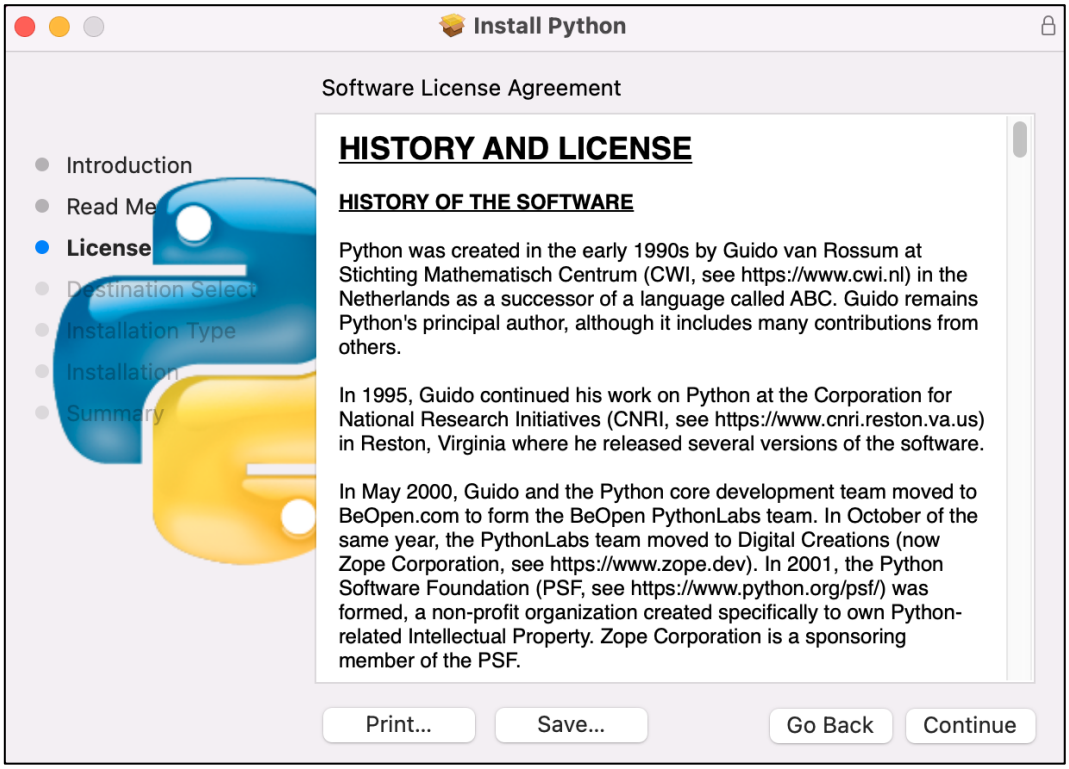


Run the downloaded installation package. Click Continue

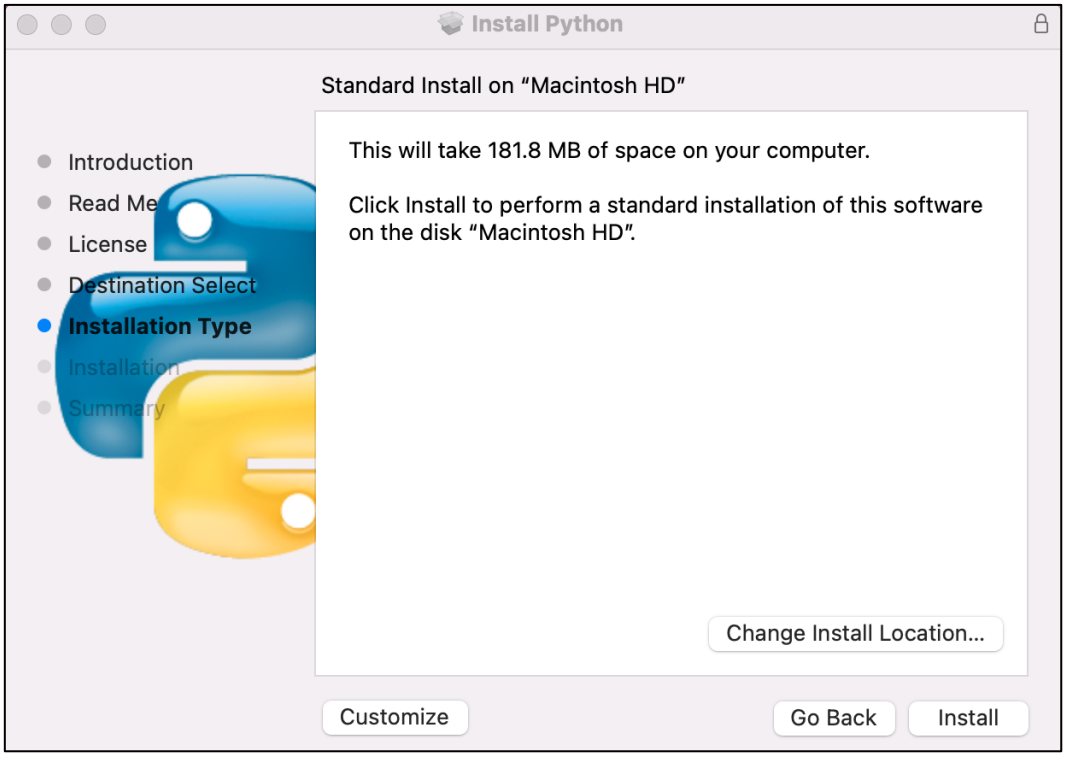
Click Continue



Click Continue



Click Install. If your computer has a password, enter the password and Install Software.



Now the installation succeeds.

Linux

Check whether Python3 has already been installed.

```
python –version
python3 –version
```

```
lin@ubuntu:~$ python --version
python: command not found
lin@ubuntu:~$ python3 --version
bash: /usr/bin/python3: No such file or directory
lin@ubuntu:~$
```

If it is not installed yet, run the following command to install it. This will install the latest version by default.

```
sudo apt install python3
```

```
lin@ubuntu: ~
python: command not found
lin@ubuntu:~$ python3 --version
python3: command not found
lin@ubuntu:~$ sudo apt install python3
Installing:
  python3
```

Link python to Python 3.

```
sudo rm /usr/bin/python
sudo ln -s /usr/bin/python3 /usr/bin/python
```

```
lin@ubuntu: ~
lin@ubuntu:~$ sudo rm /usr/bin/python
lin@ubuntu:~$ sudo ln -s /usr/bin/python3 /usr/bin/python
lin@ubuntu:~$ python --version
Python 3.13.3
```

Install python3.13-venv virtual environment.

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

```
lin@ubuntu: ~/Upload_Xiaozhi_Bin
lin@ubuntu:~/Upload_Xiaozhi_Bin$ sudo apt install python3-venv
```

Install python3-pip.

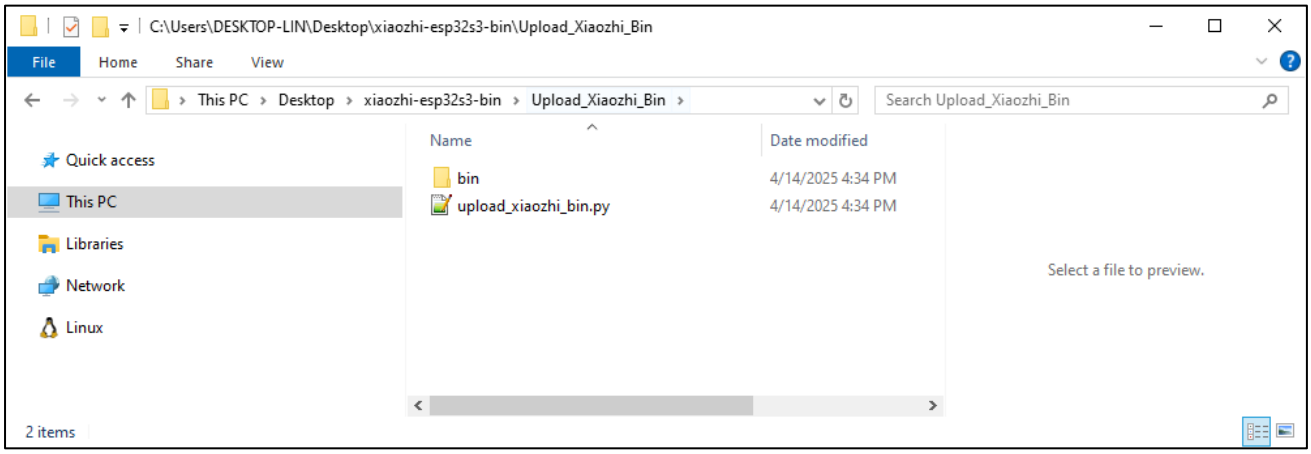
```
sudo apt install python3-pip
```

```
lin@ubuntu: ~
lin@ubuntu:~$ sudo apt install python3-pip
python3-pip is already the newest version (25.0+dfsg-1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
lin@ubuntu:~$ pip --version
pip 25.0 from /usr/lib/python3/dist-packages/pip (python 3.13)
lin@ubuntu:~$
```

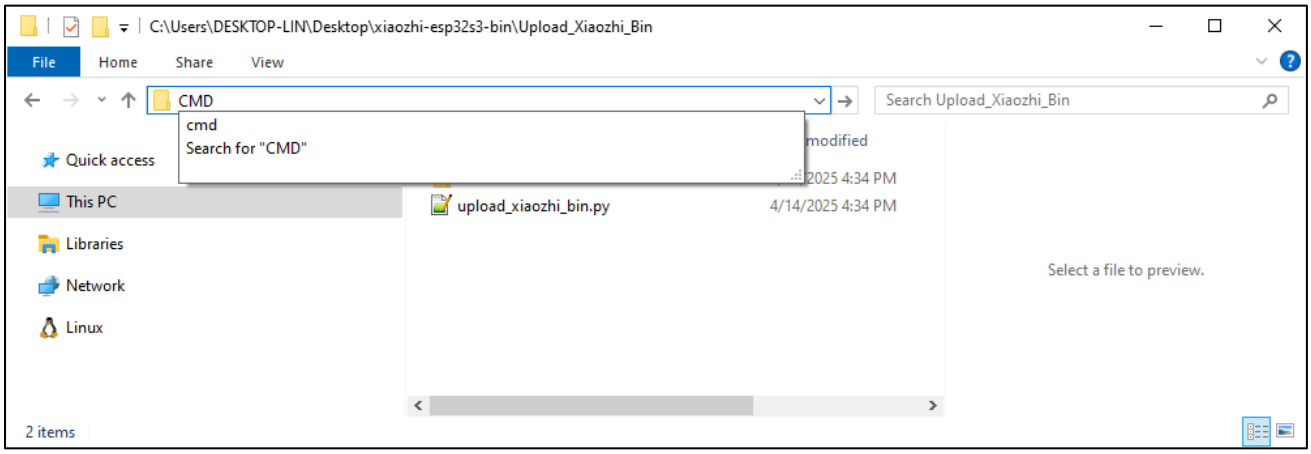
Firmware Uploading

Windows

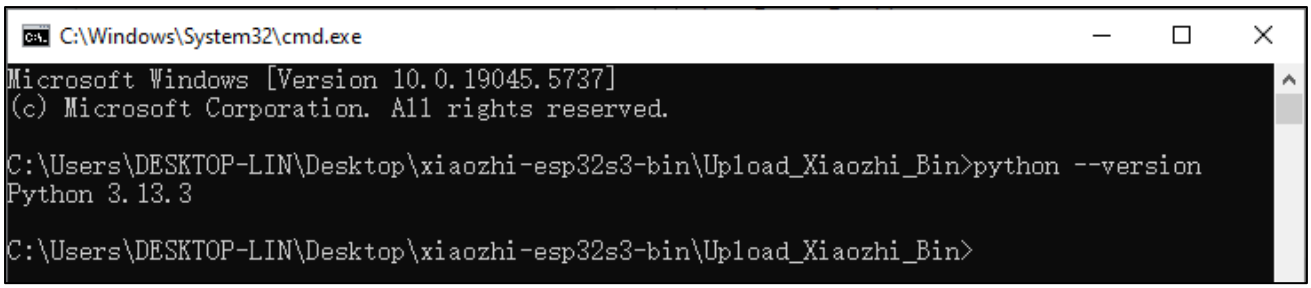
Enter the Upload_Xiaozhi_Bin folder.



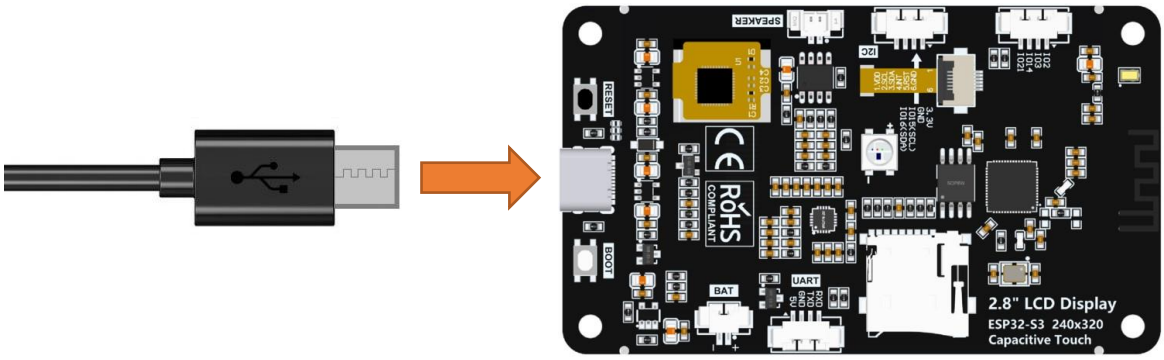
Type "CMD" in the file address bar and press Enter.



Type "**python --version**" to check if Python is installed. If no Python version information is displayed, it means Python is not properly installed—please reinstall it.



Connect the ESP32-S3-WROOM to your computer using a USB cable



Type "python upload_xiaozhi_bin.py" and press Enter.

If your computer does not have esptool or its required dependencies installed, they will be automatically installed.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DESKTOP-LIN\Desktop\xiaozhi-esp32s3-bin\Upload_Xiaozhi_Bin>python --version
Python 3.13.3

C:\Users\DESKTOP-LIN\Desktop\xiaozhi-esp32s3-bin\Upload_Xiaozhi_Bin>python upload_xiaozhi_bin.py
esptool is not installed. Installing now...
Looking in indexes: https://mirrors.aliyun.com/pypi/simple/
Collecting esptool
  Using cached https://mirrors.aliyun.com/pypi/packages/5c/6b/3ce9bb7f36bdef3d6ae71646a1d3b7d59826a4
78f3ed8a783a93a2f8f537/esptool-4.8.1.tar.gz (409 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting bitstring!=4.2.0,>=3.1.6 (from esptool)
  Downloading https://mirrors.aliyun.com/pypi/packages/75/2d/174566b533755ddf8efb32a5503af61c756a983
de379f8ad3aed6a982d38/bitstring-4.3.1-py3-none-any.whl (71 kB)
Collecting cryptography>=2.1.4 (from esptool)
  Downloading https://mirrors.aliyun.com/pypi/packages/33/cf/1f7649b8b9a3543e042d3f348e398a061923ac0
5b507f3f4d95f11938aa9/cryptography-44.0.2-cp39-abi3-win_amd64.whl (3.2 MB)
     ---------------------------------------- 3.2/3.2 MB 2.6 MB/s eta 0:00:00
```

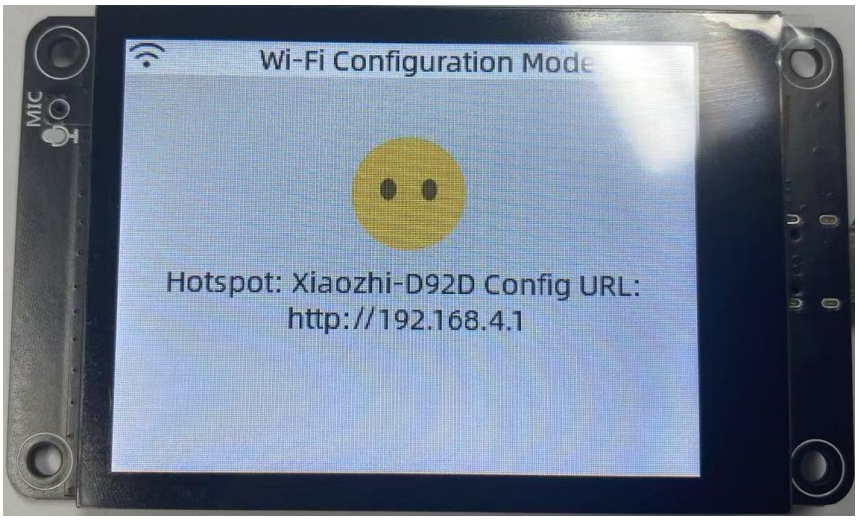
Then, it will invoke esptool to upload the files from the bin folder to the ESP32-S3-WROOM.

```
C:\Windows\System32\cmd.exe
SHA digest in image updated
Compressed 16352 bytes to 11342...
Wrote 16352 bytes (11342 compressed) at 0x00000000 in 0.2 seconds (effective 802.0 kbit/s)...
Hash of data verified.
Compressed 3561632 bytes to 2079901...
Wrote 3561632 bytes (2079901 compressed) at 0x00100000 in 22.8 seconds (effective 1247.5 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 141...
Wrote 3072 bytes (141 compressed) at 0x00008000 in 0.0 seconds (effective 1025.9 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 31...
Wrote 8192 bytes (31 compressed) at 0x0000d000 in 0.0 seconds (effective 1812.3 kbit/s)...
Hash of data verified.
Compressed 873228 bytes to 644882...
Wrote 873228 bytes (644882 compressed) at 0x00010000 in 6.4 seconds (effective 1090.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
esptool command executed successfully.

C:\Users\DESKTOP-LIN\Desktop\xiaozhi-esp32s3-bin\Upload_Xiaozhi_Bin>_
```

You will see the following messages display on ESP32 S3 WROOM board.



Mac

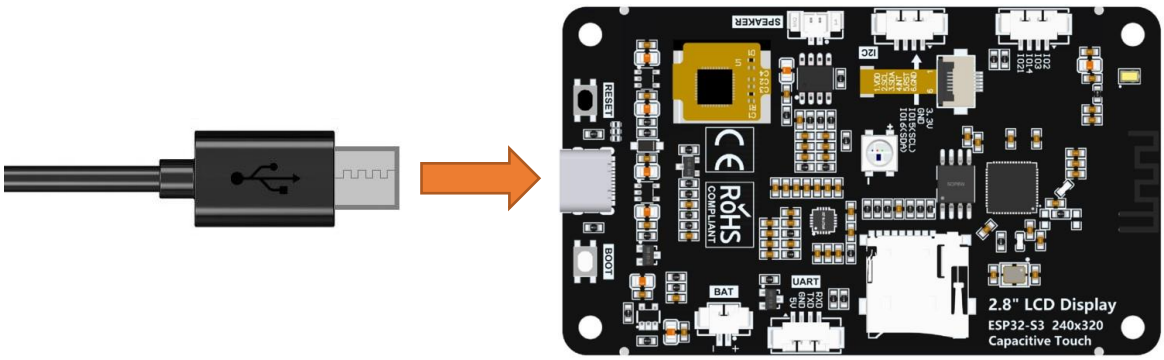
Enter the Upload_Xiaozhi_Bin folder.

```
Upload_Xiaozhi_Bin — -zsh — 91×24
freenove@PandeMacBook-Air ~ % cd Desktop/xiaozhi-esp32s3-bin/Upload_Xiaozhi_Bin
freenove@PandeMacBook-Air Upload_Xiaozhi_Bin %
```

Type "python --version" to check if Python is installed. If no Python version information is displayed, it means Python is not properly installed—please reinstall it.

```
Upload_Xiaozhi_Bin — -zsh — 91×24
freenove@PandeMacBook-Air Upload_Xiaozhi_Bin % python3 --version
Python 3.13.3
freenove@PandeMacBook-Air Upload_Xiaozhi_Bin %
```

Connect the ESP32-S3-WROOM to your computer using a USB cable



Type "python upload_xiaozhi_bin.py" and press Enter.

```
Upload_Xiaozhi_Bin — -zsh — 91×24
freenove@PandeMacBook-Air Upload_Xiaozhi_Bin % python3 upload_xiaozhi_bin.py
```

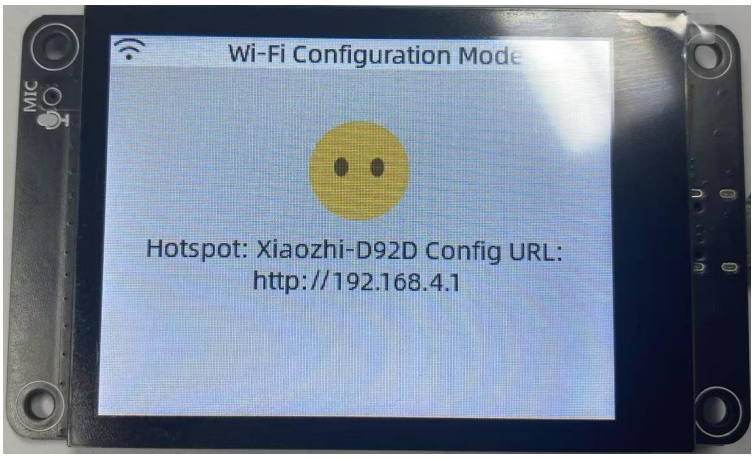
Then, it will invoke esptool to upload the files from the bin folder to the ESP32-S3-WROOM.

```
Upload_Xiaozhi_Bin — Python ◂ Python upload_xiaozhi_bin.py — 91×24
Serial port /dev/cu.wchusbserial5A4E1051341
Connecting....
Chip is ESP32-S3 (QFN56) (revision v0.2)
Features: WiFi, BLE, Embedded PSRAM 8MB (AP_3v3)
Crystal is 40MHz
MAC: 30:ed:a0:20:bd:9c
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 2000000
Changed.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x00003fff...
Flash will be erased from 0x00100000 to 0x00465fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Flash will be erased from 0x0000d000 to 0x0000efff...
Flash will be erased from 0x00010000 to 0x000e5fff...
```

```
Upload_Xiaozhi_Bin — -zsh — 91×24
SHA digest in image updated
Compressed 16352 bytes to 11342...
Wrote 16352 bytes (11342 compressed) at 0x00000000 in 0.2 seconds (effective 775.3 kbit/s).
..
Hash of data verified.
Compressed 3561632 bytes to 2079901...
Wrote 3561632 bytes (2079901 compressed) at 0x00100000 in 22.8 seconds (effective 1247.2 kb
it/s)...
Hash of data verified.
Compressed 3072 bytes to 141...
Wrote 3072 bytes (141 compressed) at 0x00008000 in 0.0 seconds (effective 889.1 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 31...
Wrote 8192 bytes (31 compressed) at 0x0000d000 in 0.0 seconds (effective 1663.8 kbit/s)...
Hash of data verified.
Compressed 873228 bytes to 644882...
Wrote 873228 bytes (644882 compressed) at 0x00010000 in 6.4 seconds (effective 1089.5 kbit/
s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
esptool command executed successfully.
freenove@PandeMacBook-Air Upload_Xiaozhi_Bin % c
```

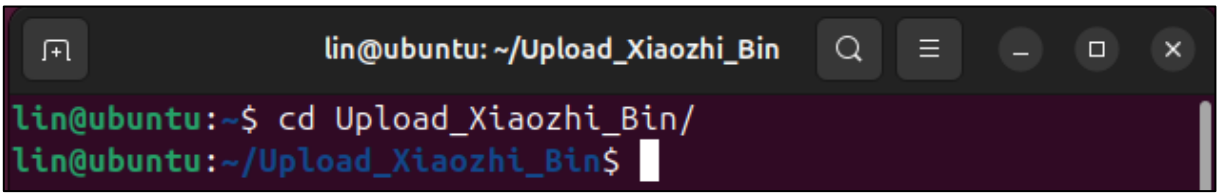
You will see the following messages display on ESP32 S3 WROOM board.



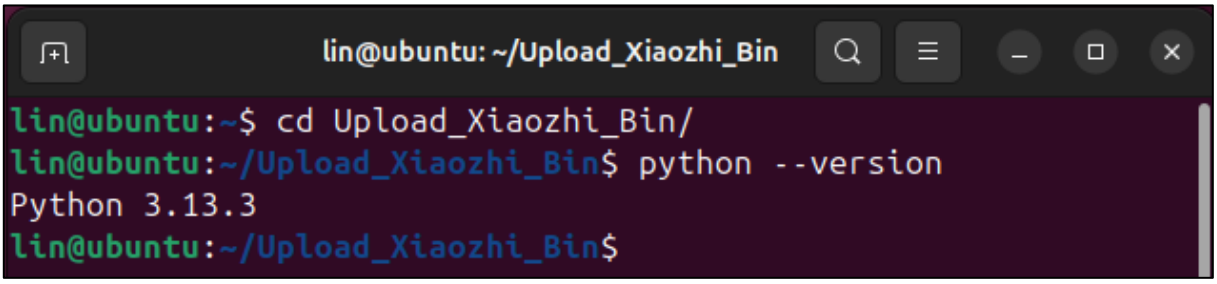
Linux

Enter the Upload_Xiaozhi_Bin folder.

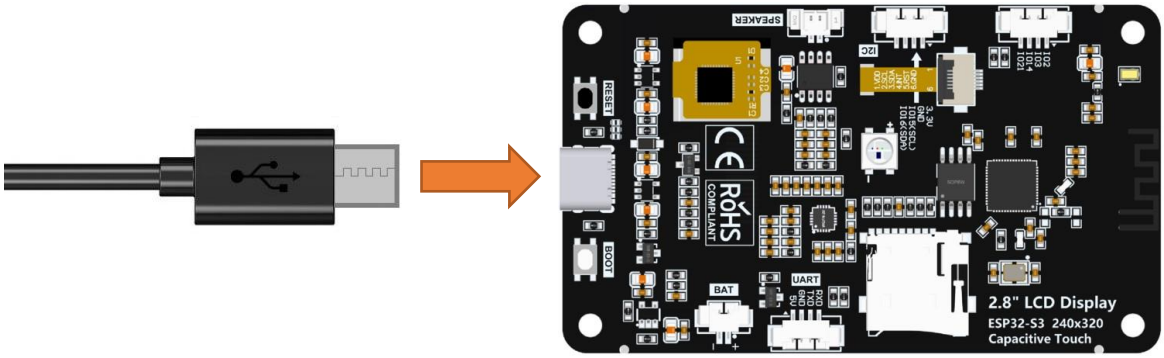
```
cd Upload_Xiaozhi_Bin
```



Enter "**python --version**" to check if the Python environment is installed. If the Python version information is not displayed, it means Python is not properly installed. Please reinstall it.

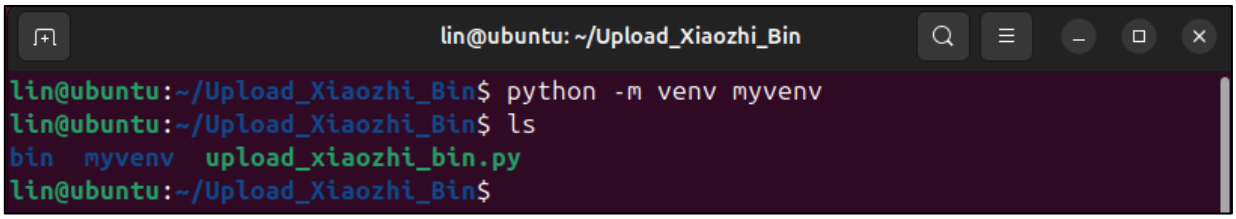


Connect the ESP32-S3-WROOM to your computer using a USB cable, making sure to plug it into the correct Type-C port (do not use the wrong connector).



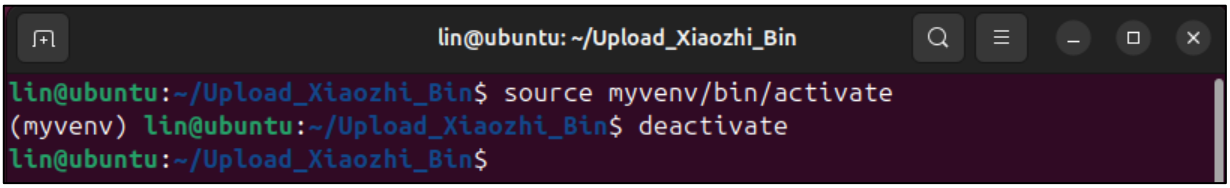
Create a virtual environment and name it as “myvenv”.

```
python -m venv myvenv
```

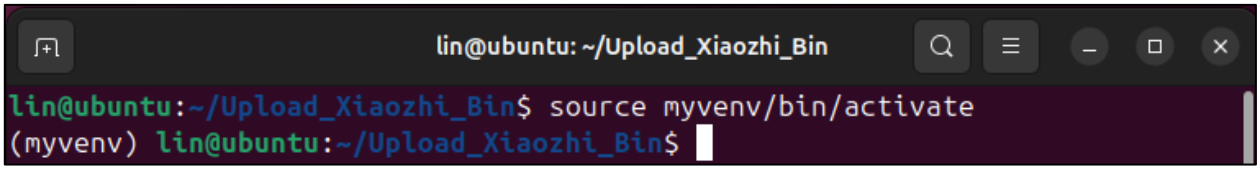


You can run the following command to activate or exit the virtual environment.

```
source myvenv/bin/activate
deactivate
```



Activate the virtual environment.



Run the command to check the port of ESP32S3.

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

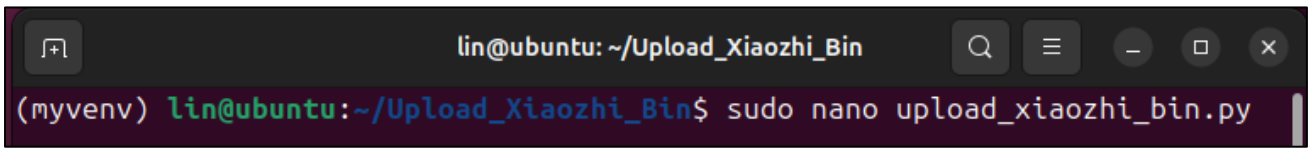
When the ESP32S3 is not connected to the computer, the ports are as shown below.

After connecting the ESP32S3, a new port is generated.

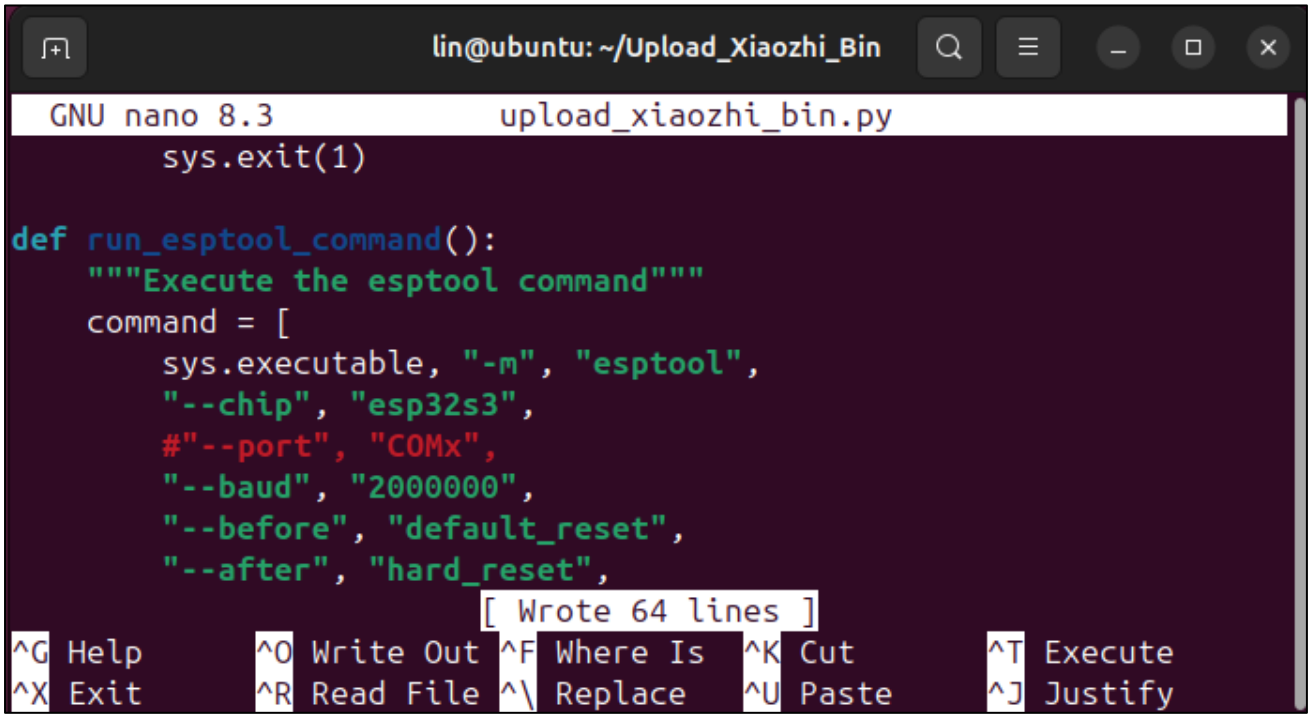
The newly generated one is the port of ESP32S3. Remember it.

Before running the python file, we need to modify the port.
Run the following command to open the python file.

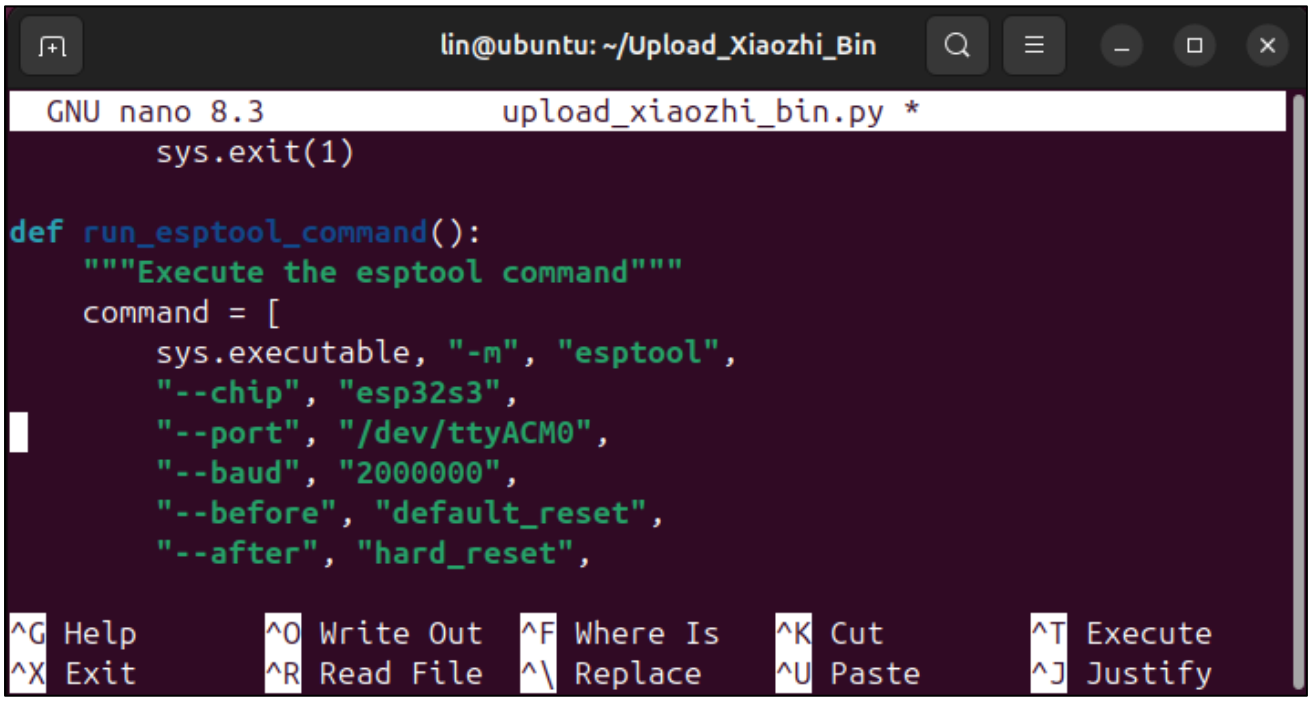
```
sudo nano upload_xiaozhi_bin.py
```



In the text editor, locate the line '--port', 'COMx' and replace 'COMx' with the port number assigned to your ESP32-S3 on Linux computer.



The modification is as shown below.



Press “Ctrl+O” to save the changes and “Ctrl+X” to exit the file.

Run the python file.

```
python upload_xiaozhi_bin.py
```

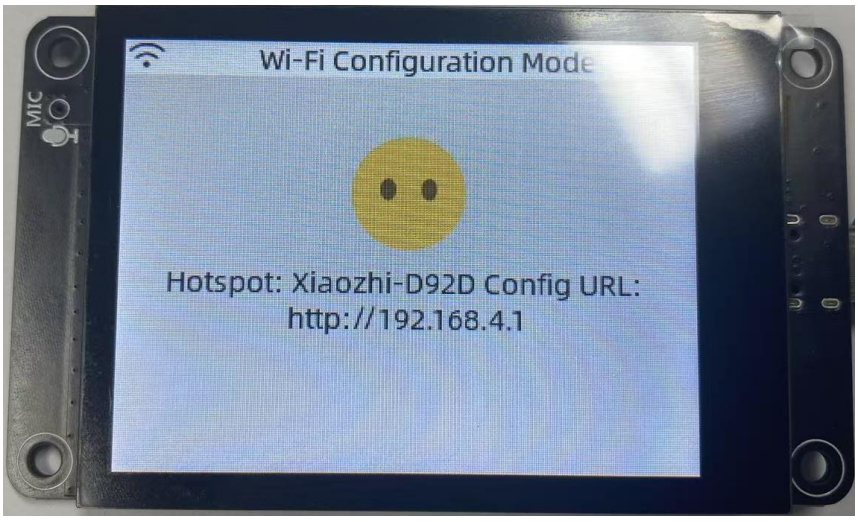
```
lin@ubuntu: ~/Upload_Xiaozhi_Bin
(myvenv) lin@ubuntu:~/Upload_Xiaozhi_Bin$ python upload_xiaozhi_bin.py
esptool is already installed.
Executing esptool command...
esptool.py v4.8.1
Serial port /dev/ttyACM0
Connecting....
Chip is ESP32-S3 (QFN56) (revision v0.2)
Features: WiFi, BLE, Embedded PSRAM 8MB (AP_3v3)
Crystal is 40MHz
MAC: 30:ed:a0:20:bd:9c
Uploading stub...
Running stub...
```

The successful code uploading is as shown below.

```
Wrote 8192 bytes (31 compressed) at 0x0000d000 in 0.0 seconds (effective 18
43.6 kbit/s)...
Hash of data verified.
Compressed 873228 bytes to 644882...
Wrote 873228 bytes (644882 compressed) at 0x00010000 in 6.3 seconds (effect
ive 1103.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
esptool command executed successfully.
(myvenv) lin@ubuntu:~/Upload_Xiaozhi_Bin$
```

The display on the ESP32 S3 WROOM is as shown below.



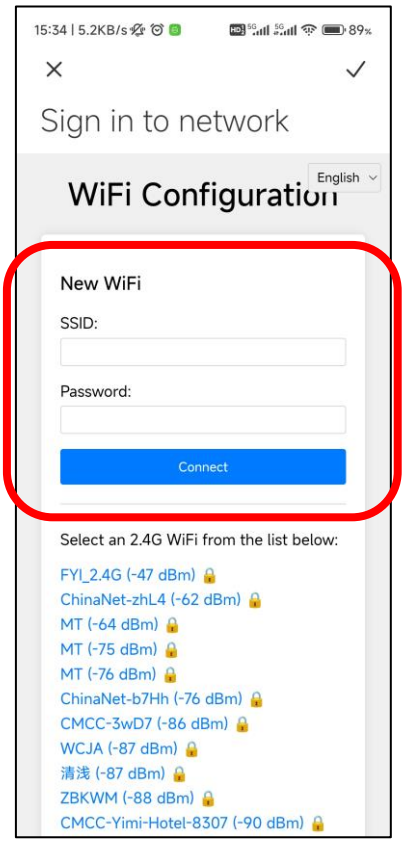
ESP32 S3 WROOM Network Configuration

If your ESP32-S3-WROOM does not yet have the XiaoZhi AI firmware installed, proceed to the
If you want to explore the XiaoZhi AI code, go to the XiaoZhi AI Code section.

If your ESP32-S3-WROOM already has the XiaoZhi AI firmware integrated:

1. On your smart phone, enable WiFi.
2. Look for a hotspot named "Xiaozhi-XXXX" (an open network, no password required).
3. Connect to it to proceed

After connecting to the WiFi, follow the on-screen prompts to tap the notification. This will automatically launch your mobile browser and direct you to http://192.168.4.1.



WiFi Connection Setup for ESP32-S3-WROOM

Enter WiFi Credentials:

SSID: Enter your WiFi network name (2.4GHz only).

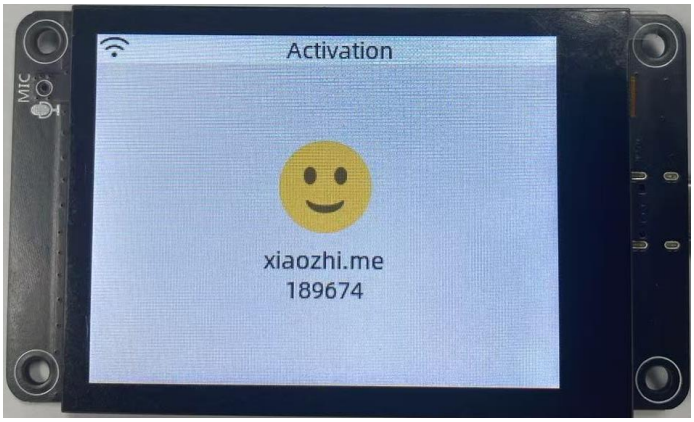
Password: Enter your WiFi password.

Click Connect to proceed.

Important Notes:

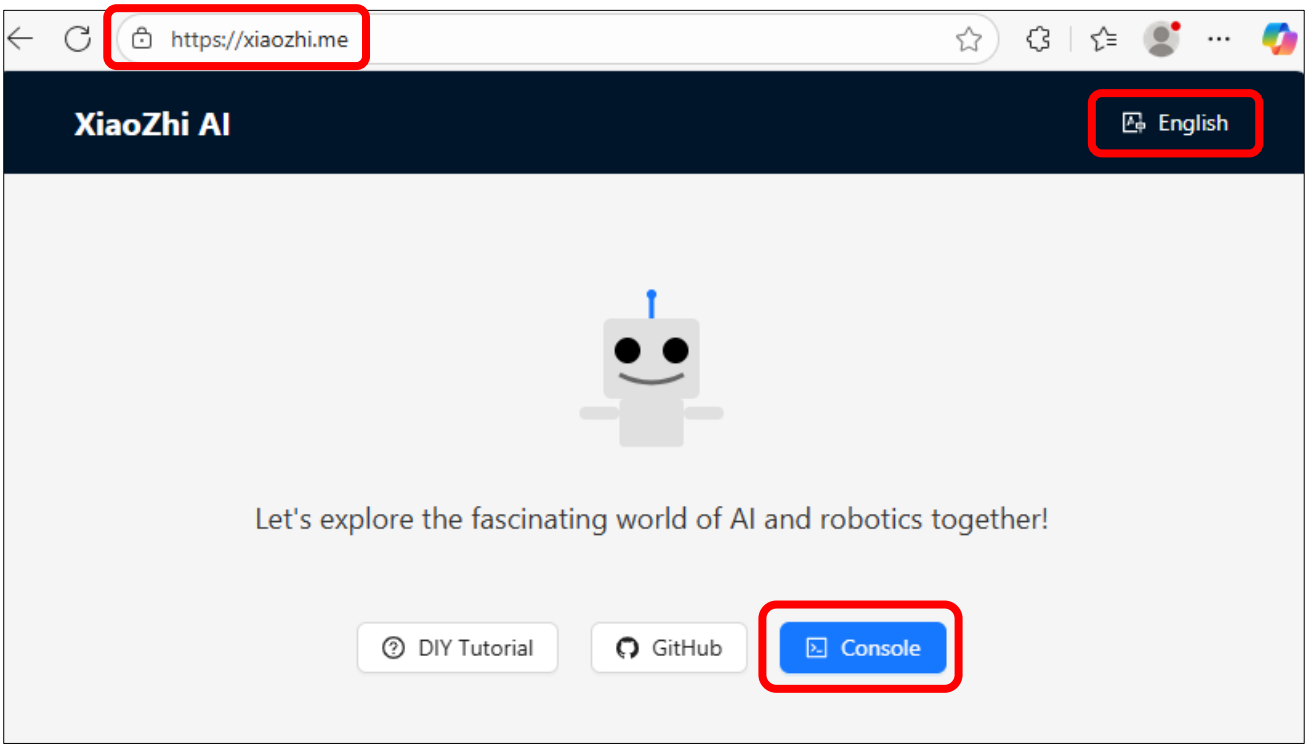
- The ESP32-S3-WROOM only supports 2.4GHz WiFi networks.
- If your router broadcasts both 2.4GHz and 5GHz, ensure the ESP32 connects to the 2.4GHz band only.
- Avoid mixed-mode (2.4GHz + 5GHz combined) settings, as this may prevent successful connection.

When you see the following screen, it means your ESP32-S3 has successfully connected to your WiFi network.



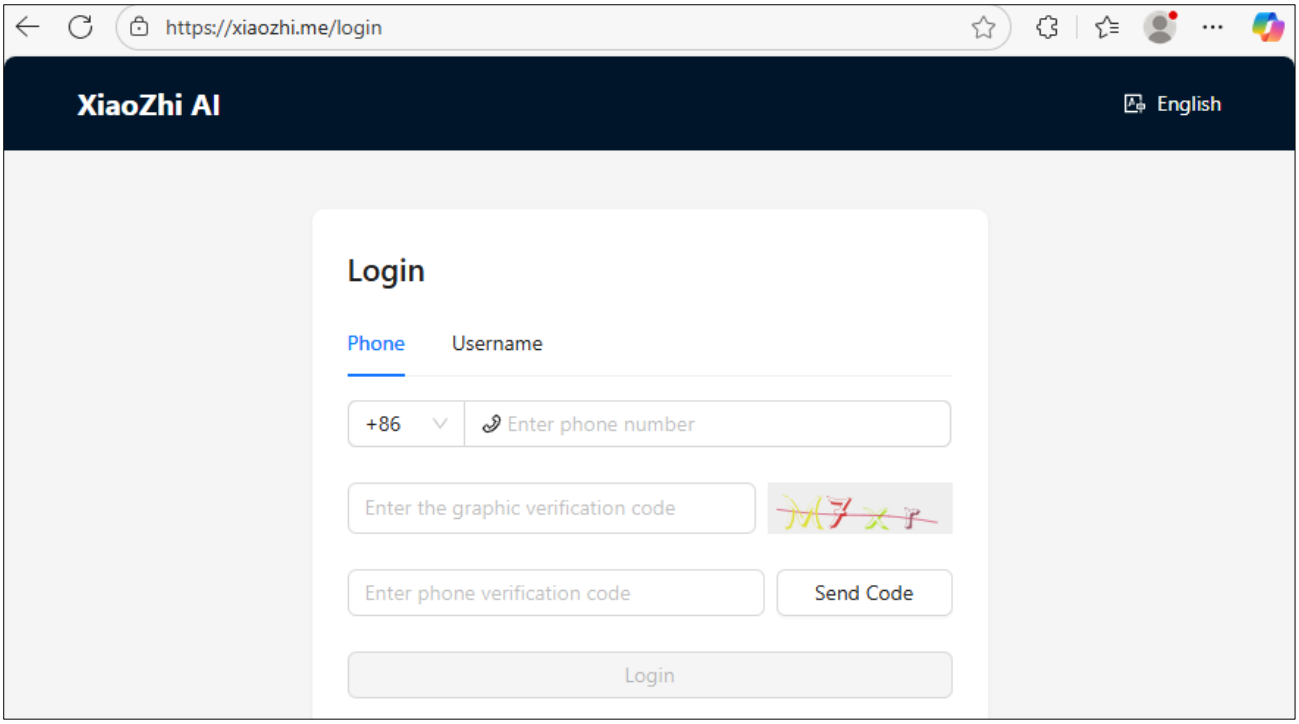
XiaoZhi AI Server Configuration

Ensure your phone/computer and ESP32-S3-WROOM are connected to the same router WiFi network.
Open a browser on your device and visit: https://xiaozhi.me/
Please note that due to varying internet policies in different countries, users from certain regions may experience difficulties accessing the website. For specific details, please refer to the relevant national internet policies.

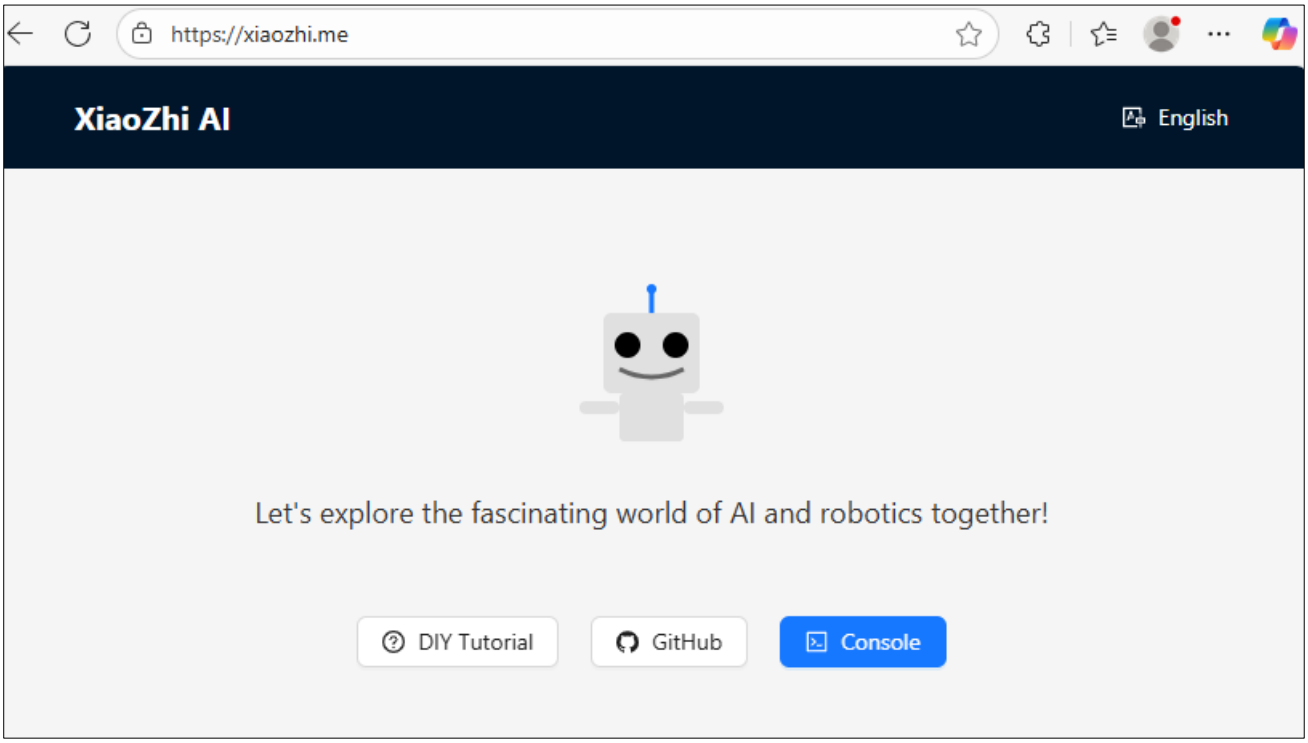


If you don't have an account yet, please click Console and register using your mobile number.
Please note that currently, XiaoZhi AI servers only support mobile number registration from the following countries.

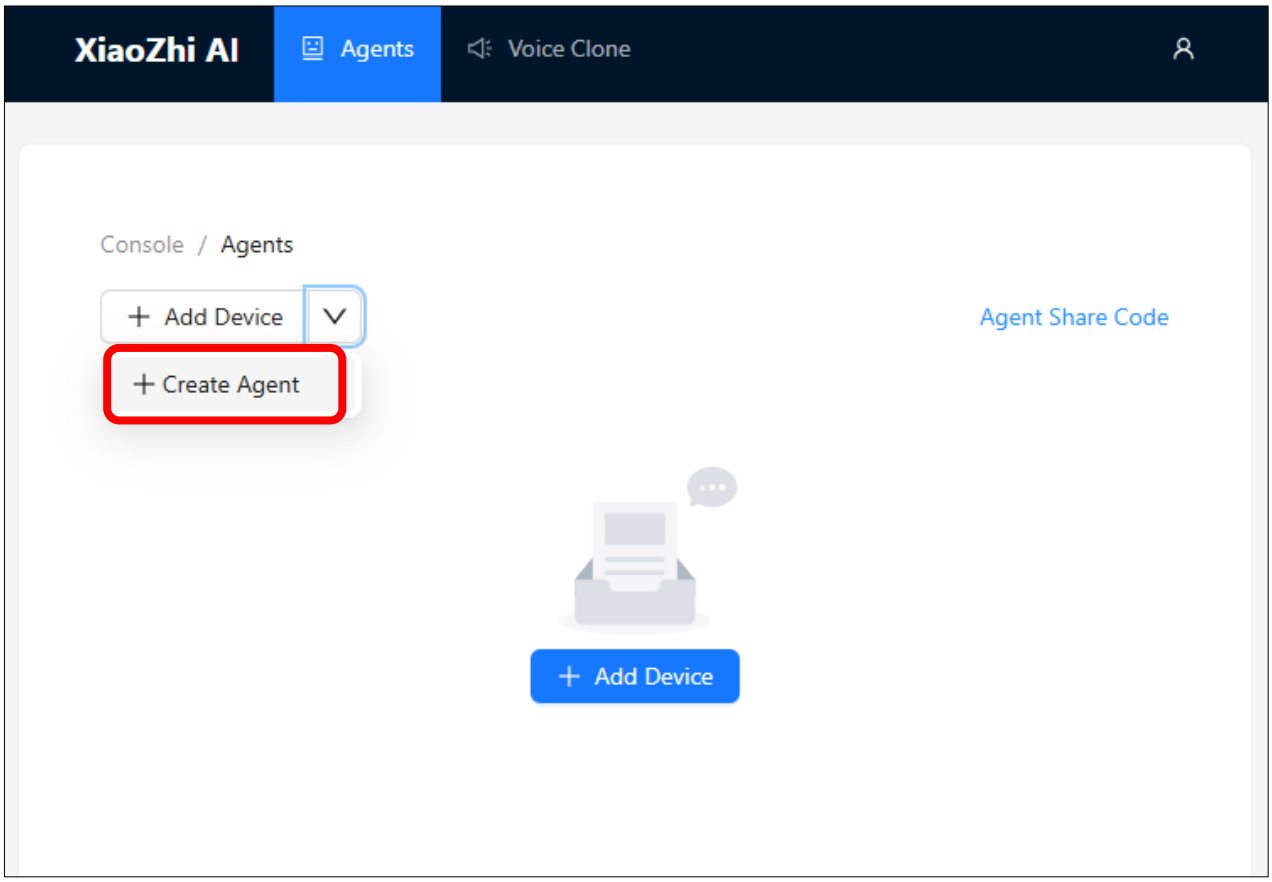
If you do not have an account yet, please register one and login.



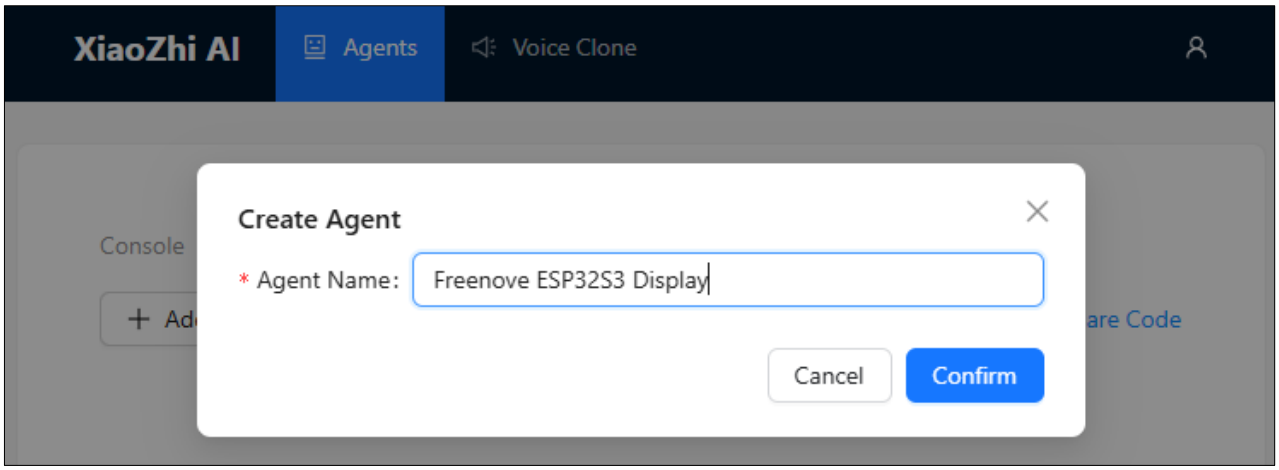
Click "Console" to start setting up your XiaoZhi AI server.



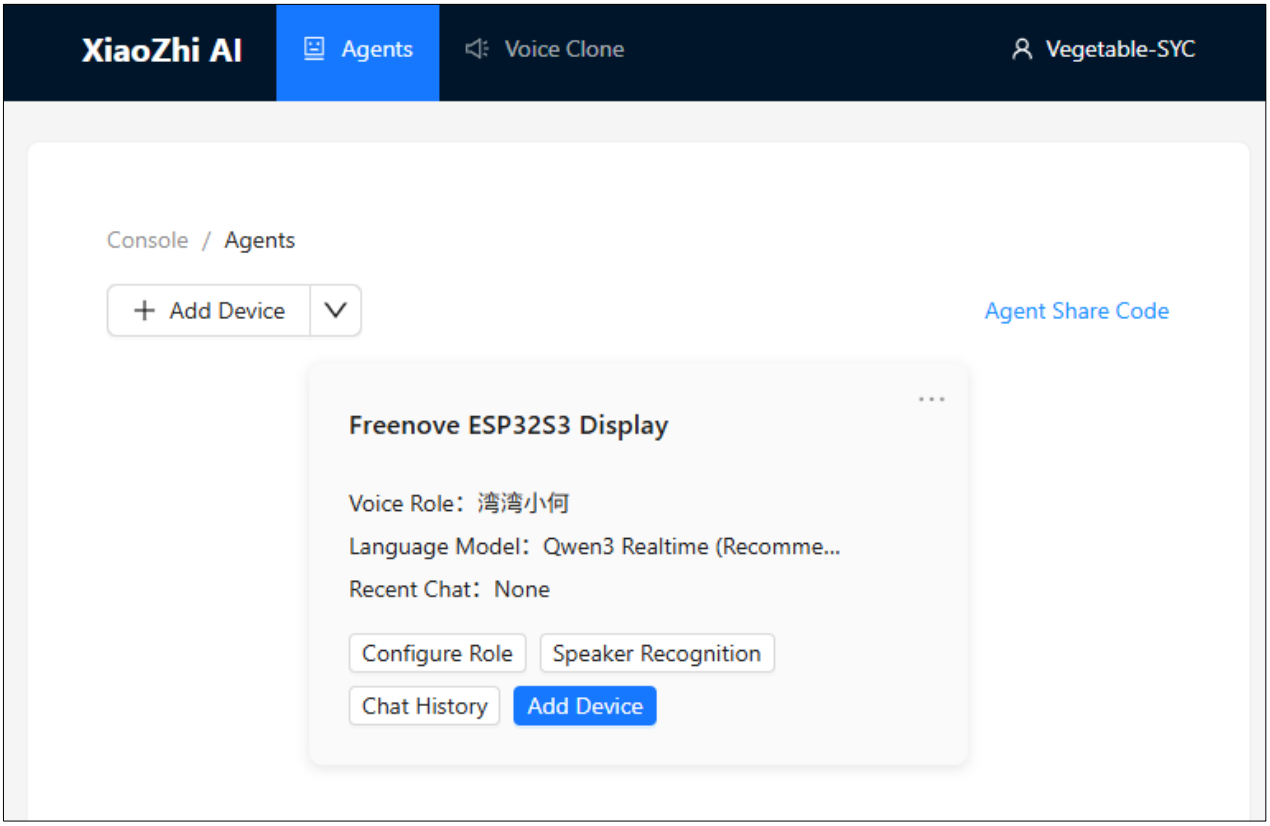
Click "**Create Agent**" to set up a new AI assistant.



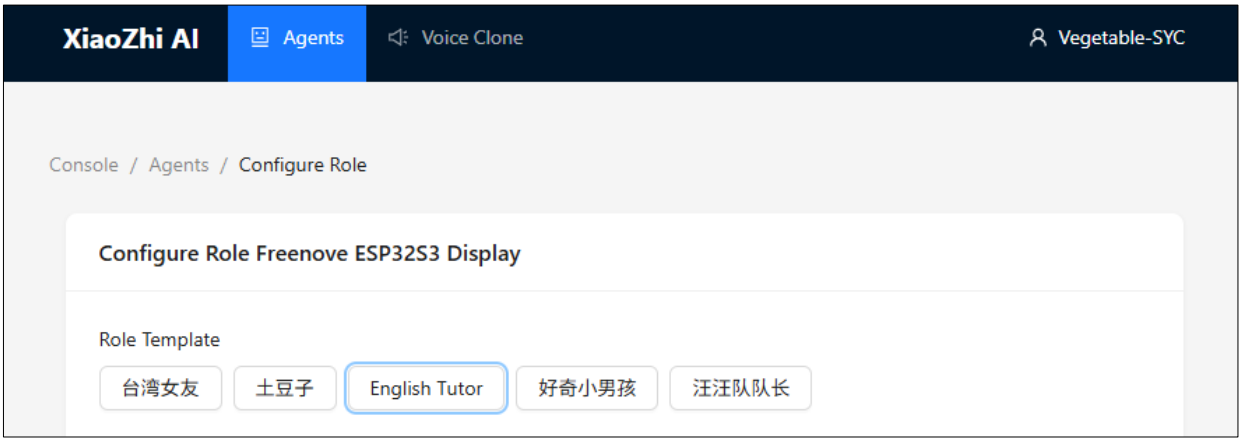
Name it whatever you like and click “Confirm”.



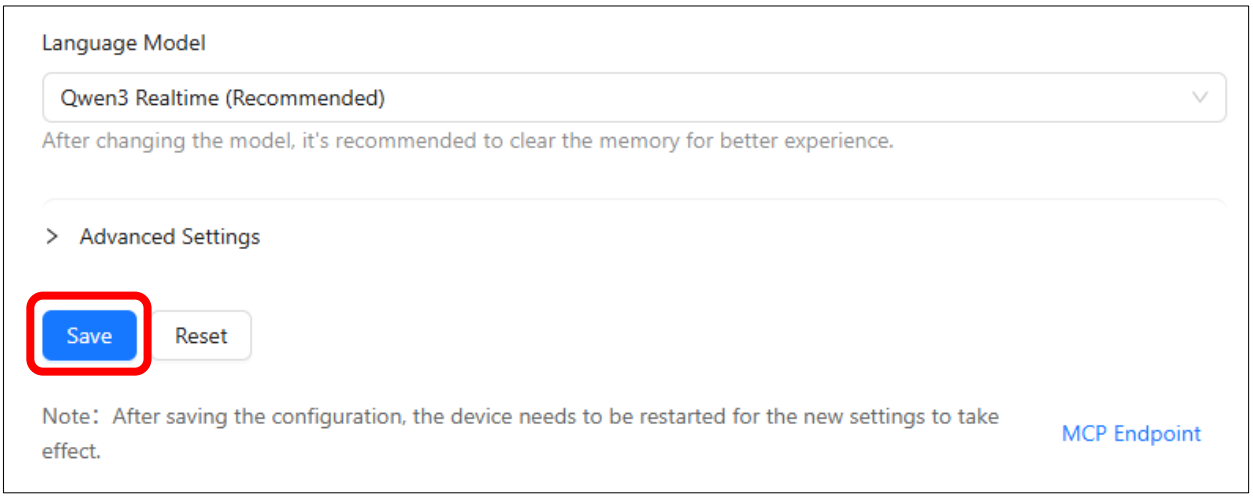
Click "**Configure Role**" to configure your AI assistant.



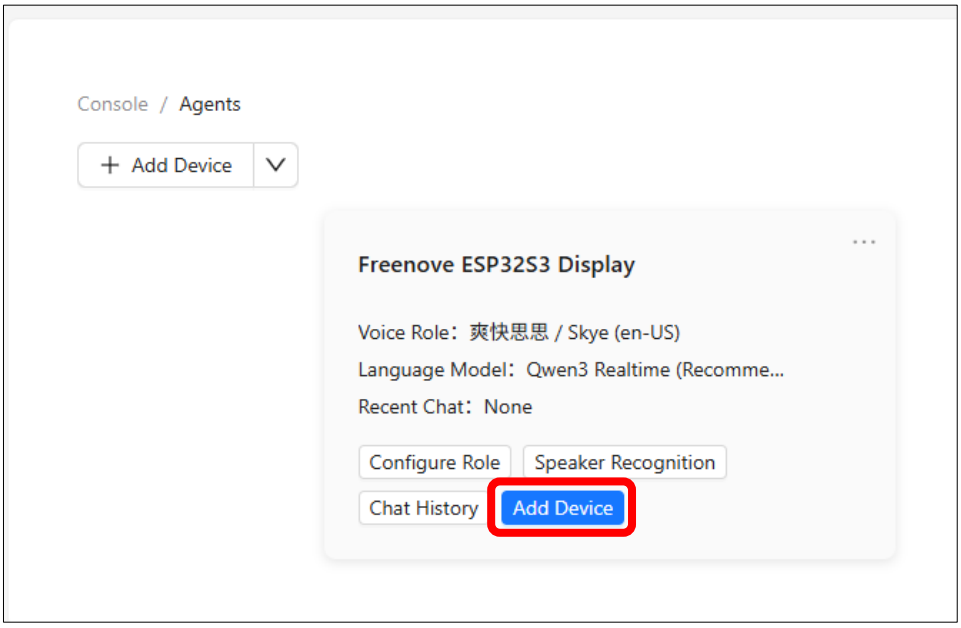
Click "**English Tutor**" (keep all other options unchanged).



Scroll to the bottom of the page and click "Save" to confirm all settings.

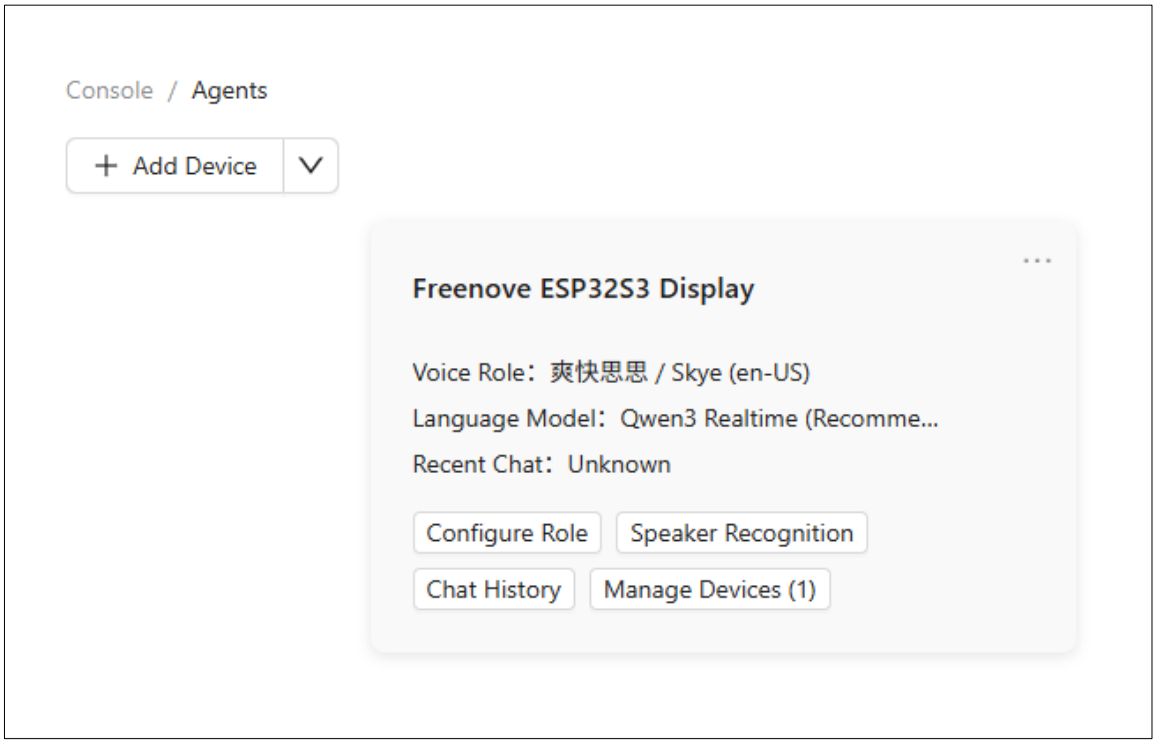


Click "Agents" to return to the main dashboard and select "**Add Device**" to register new hardware.



In the new pop-up window, enter the on-screen numeric code displayed on your ESP32-S3. Click "Confirm" to complete pairing.

The interface will now display as shown below.

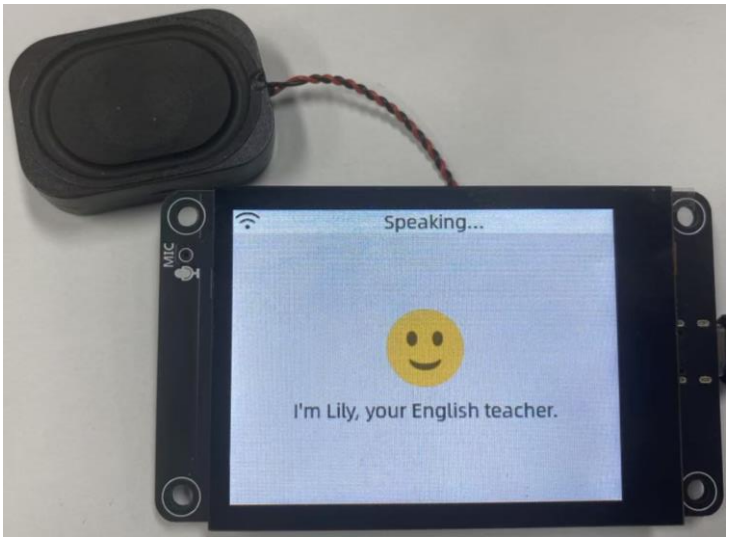


Press the RST button on the Freenove ESP32 S3 Display board to restart the board.

You've successfully finished configuring XiaoZhi AI!

Connect speaker

To activate, say "**Hi, ESP**" to the microphone; the system will now respond to your voice commands



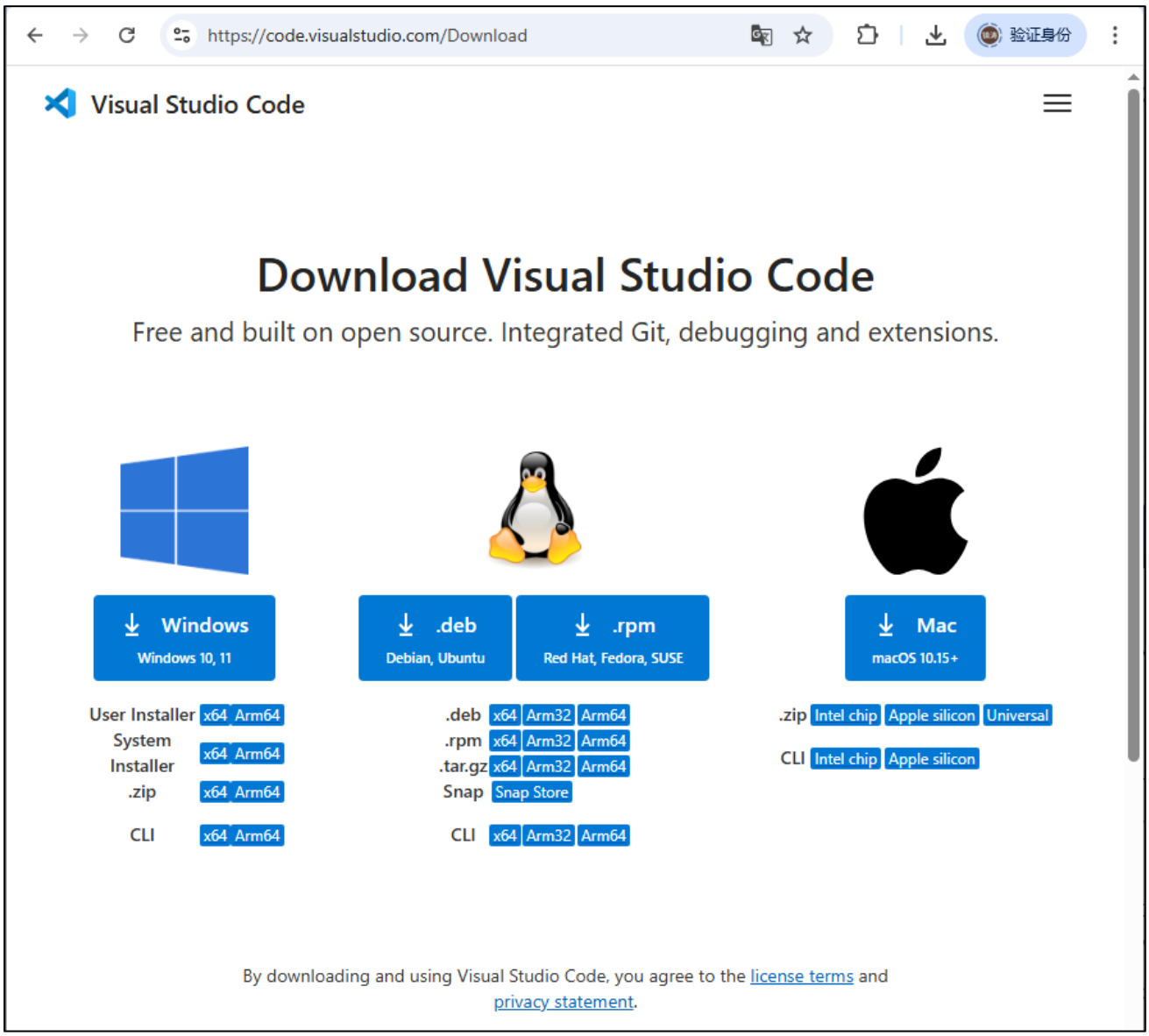
You can communicate with it in either Chinese or English.

XiaoZhi AI Code

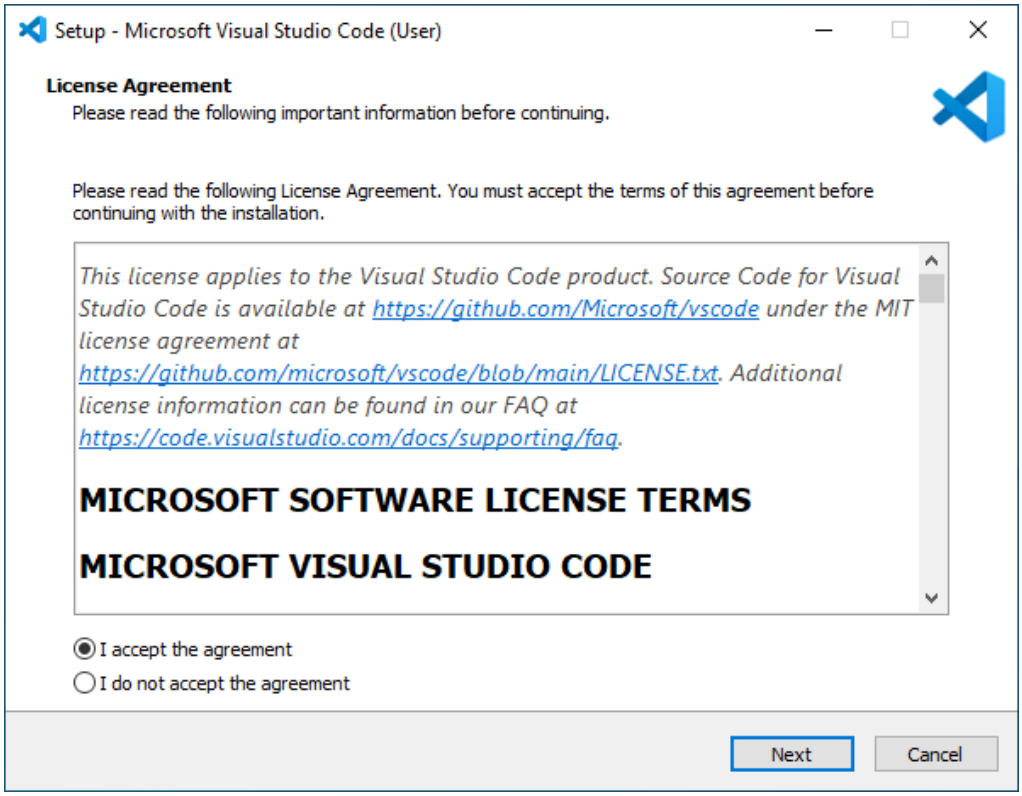
Visual Studio Code

Windows

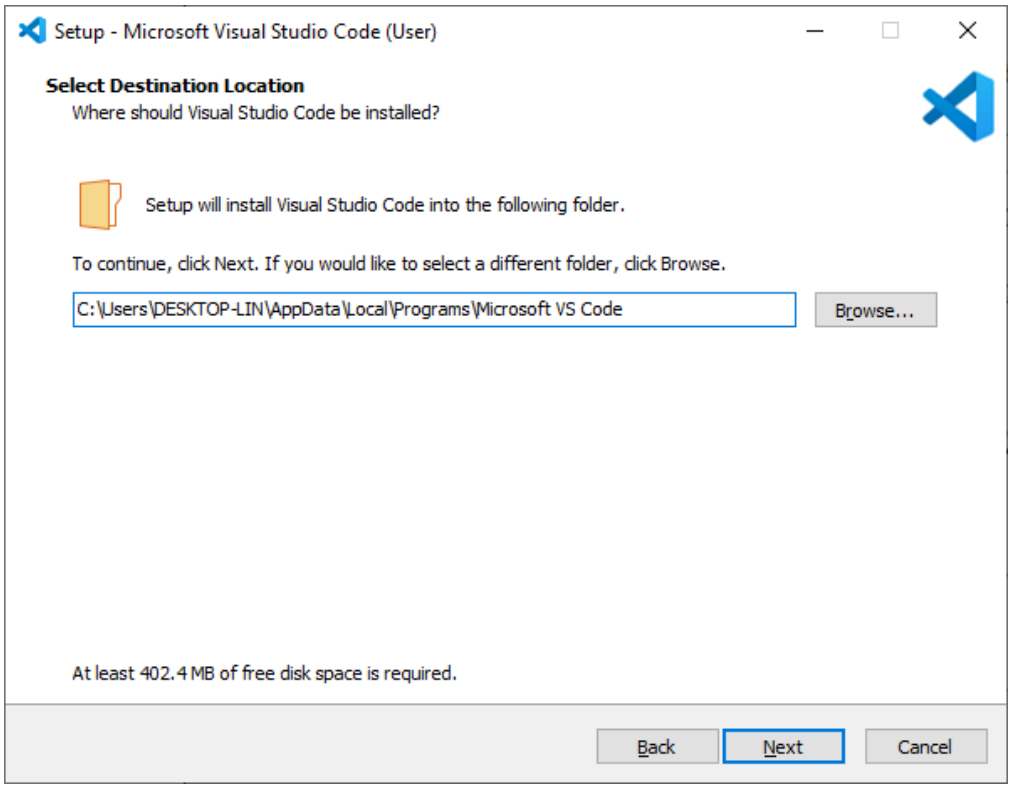
First, download Visual Studio Code by visiting https://code.visualstudio.com/Download. Choose the appropriate version for your operating system, then download and install it.



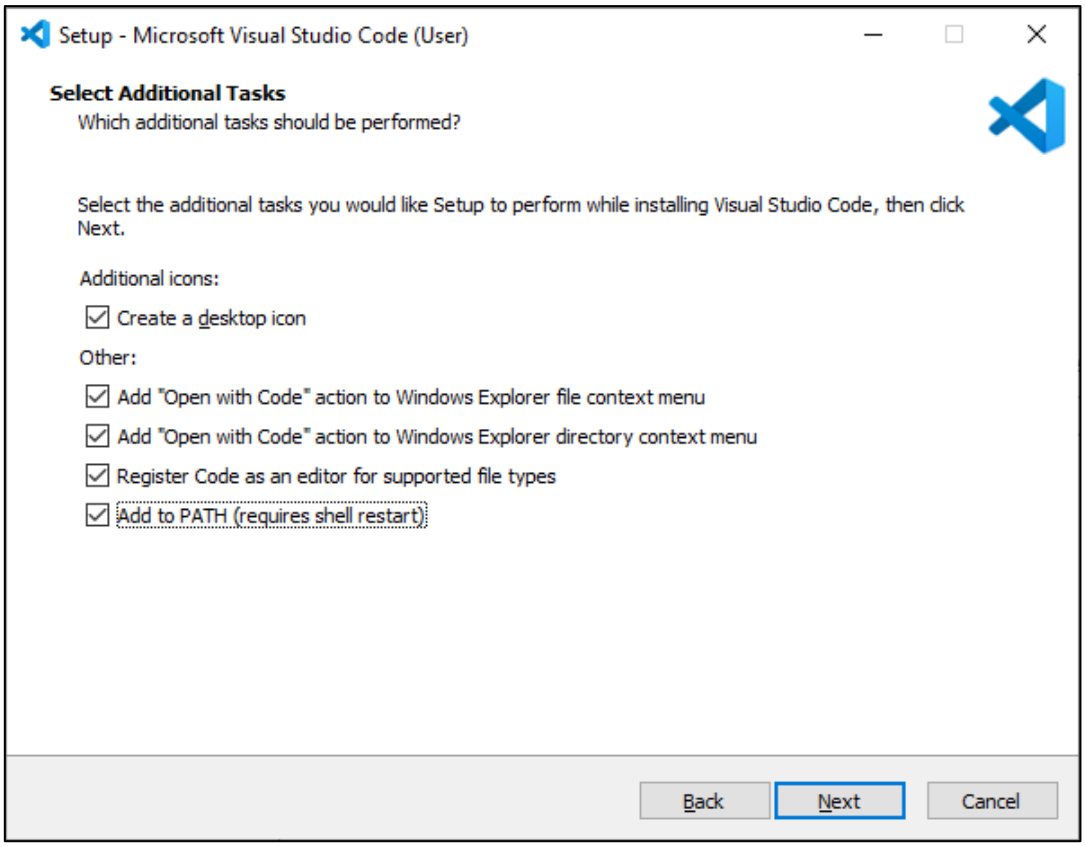
Double-click the downloaded .exe file to run it.
Check the box for "I accept the agreement."
Then click Next.



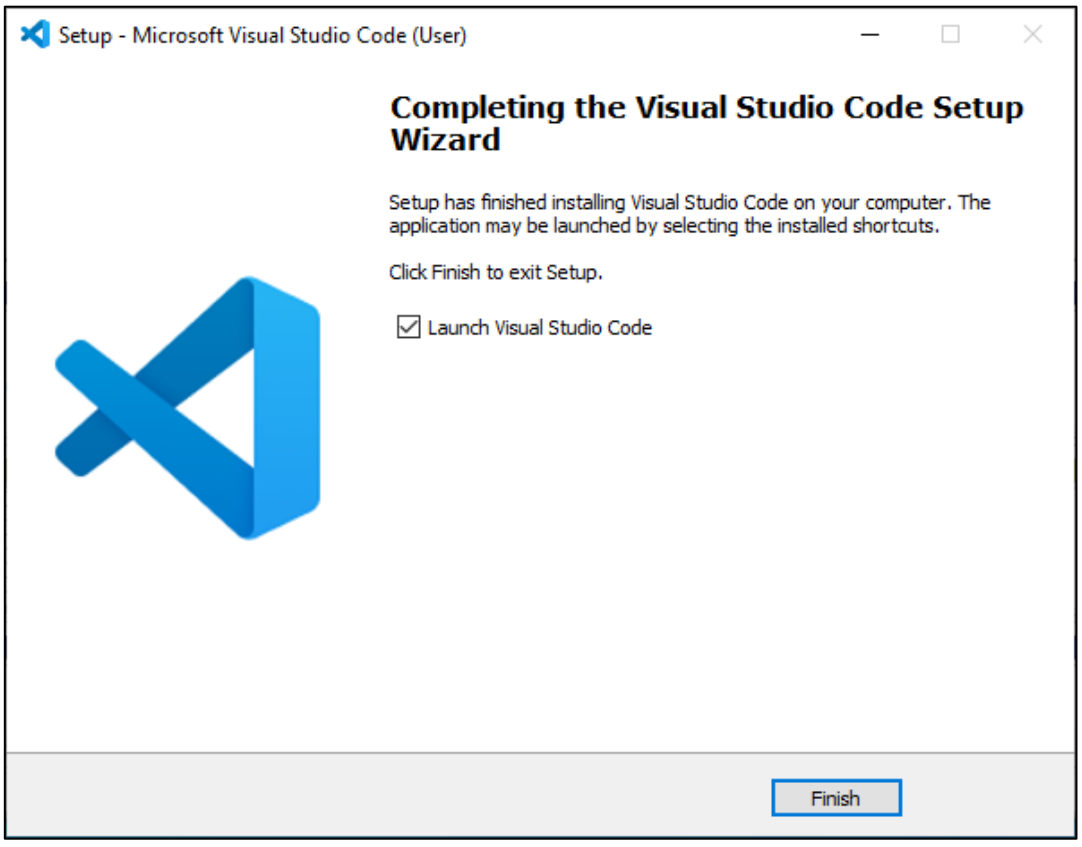
The installation location can be left as the default or changed to a desired path. After that, proceed by clicking Next repeatedly.



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



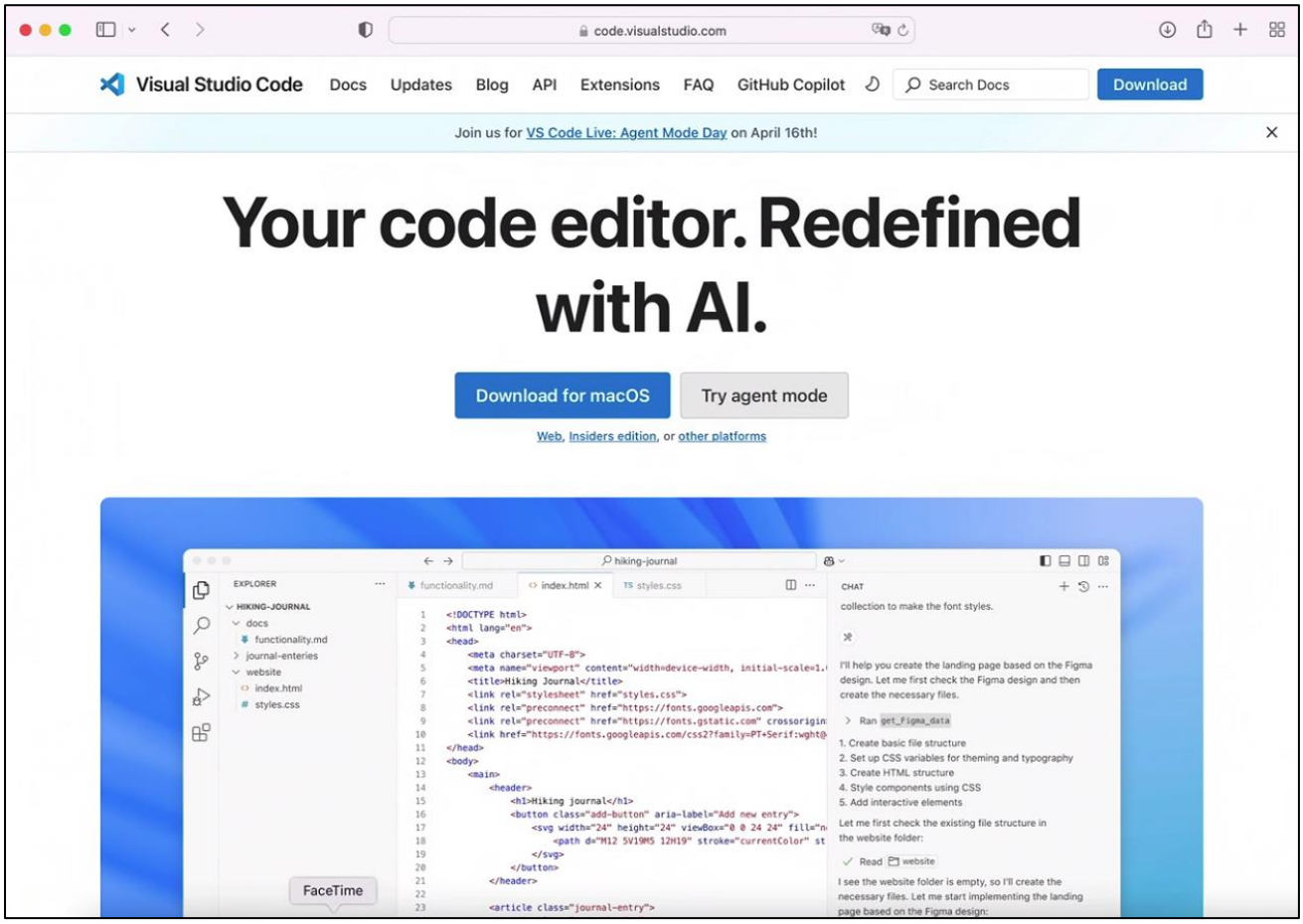
The installation is now complete, as shown in the image below.



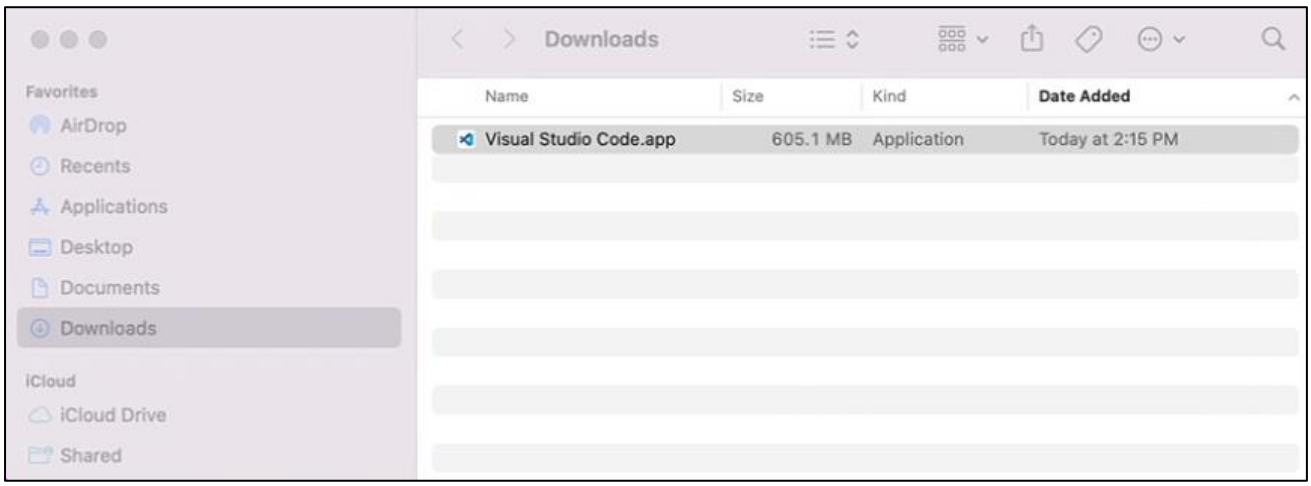
Mac

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".

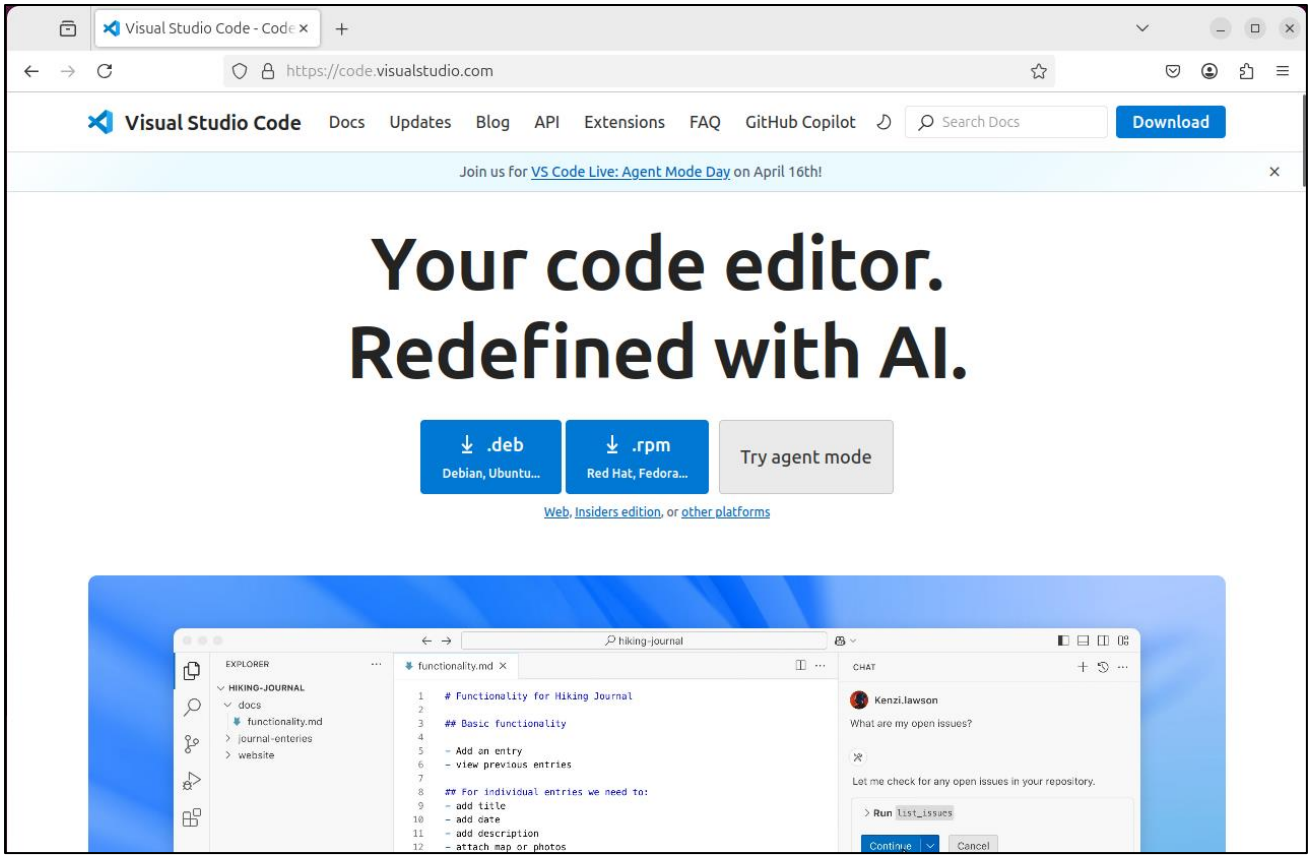


Double click to run the program.

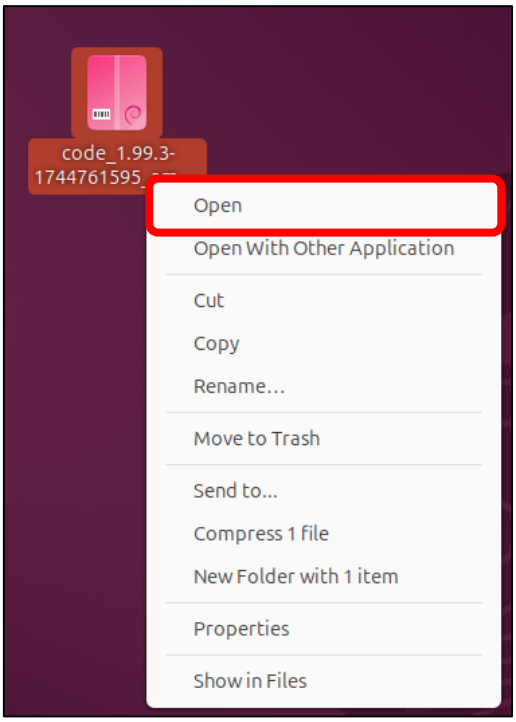


Linux

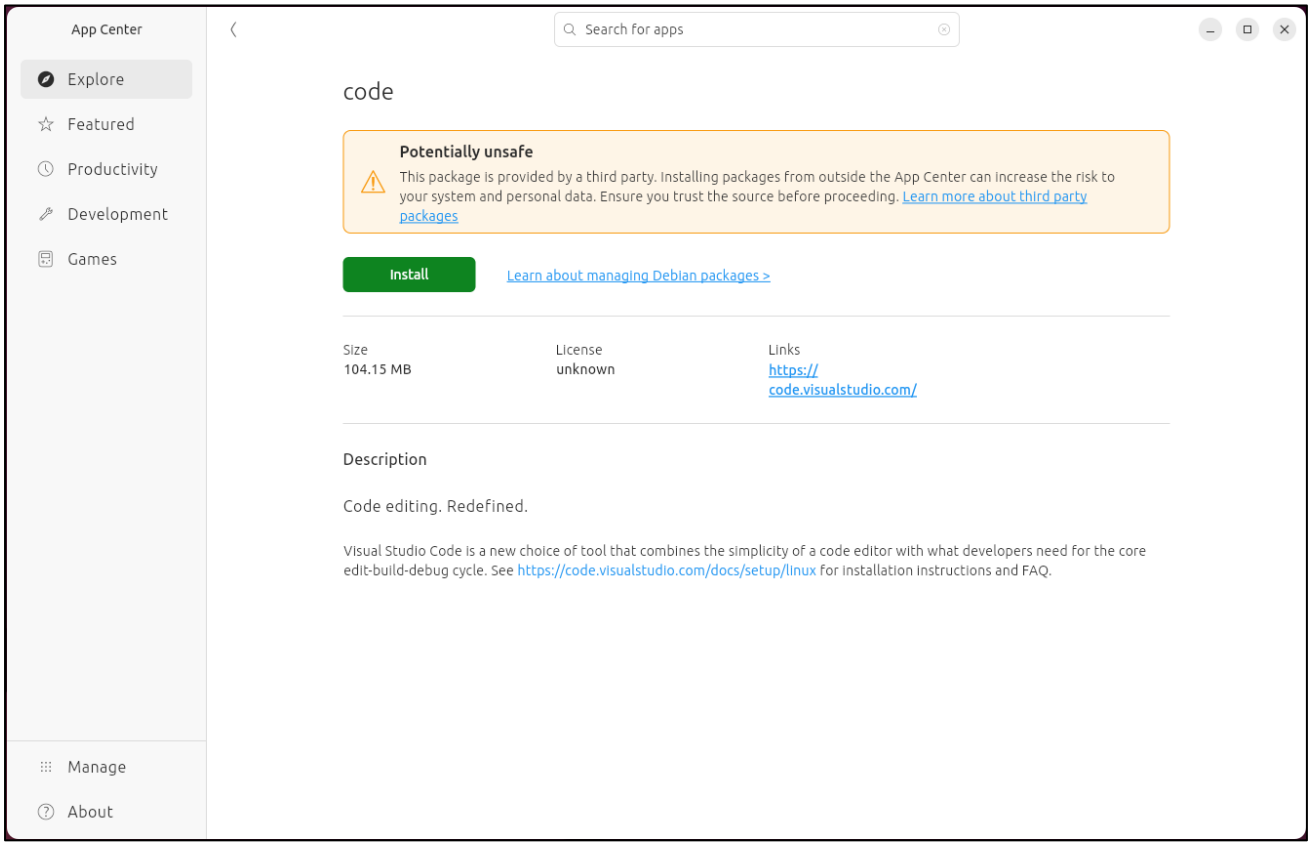
If your computer does not have Visual Studio Code, please install it first.
Visit https://code.visualstudio.com and click ".deb".



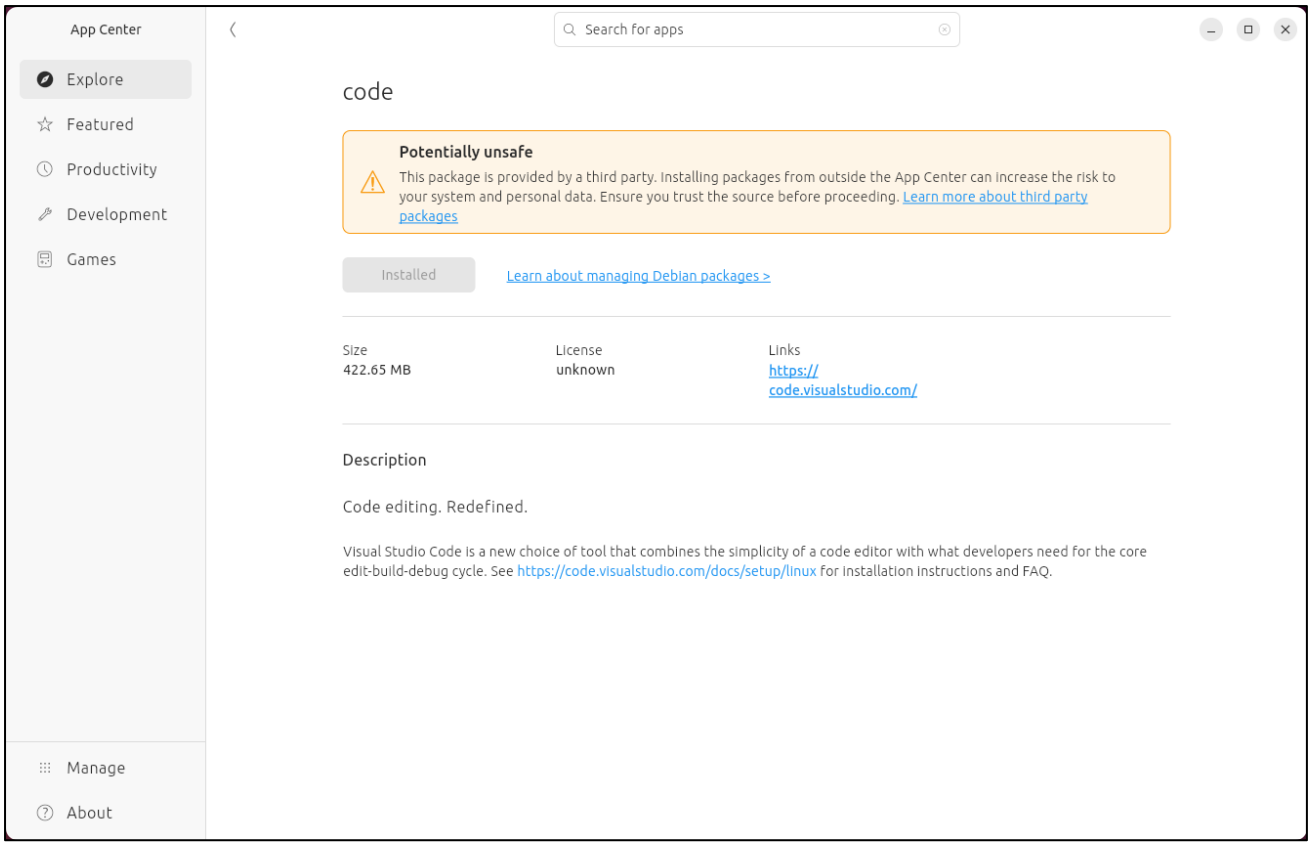
Open the downloaded“code_xxx.deb” file.



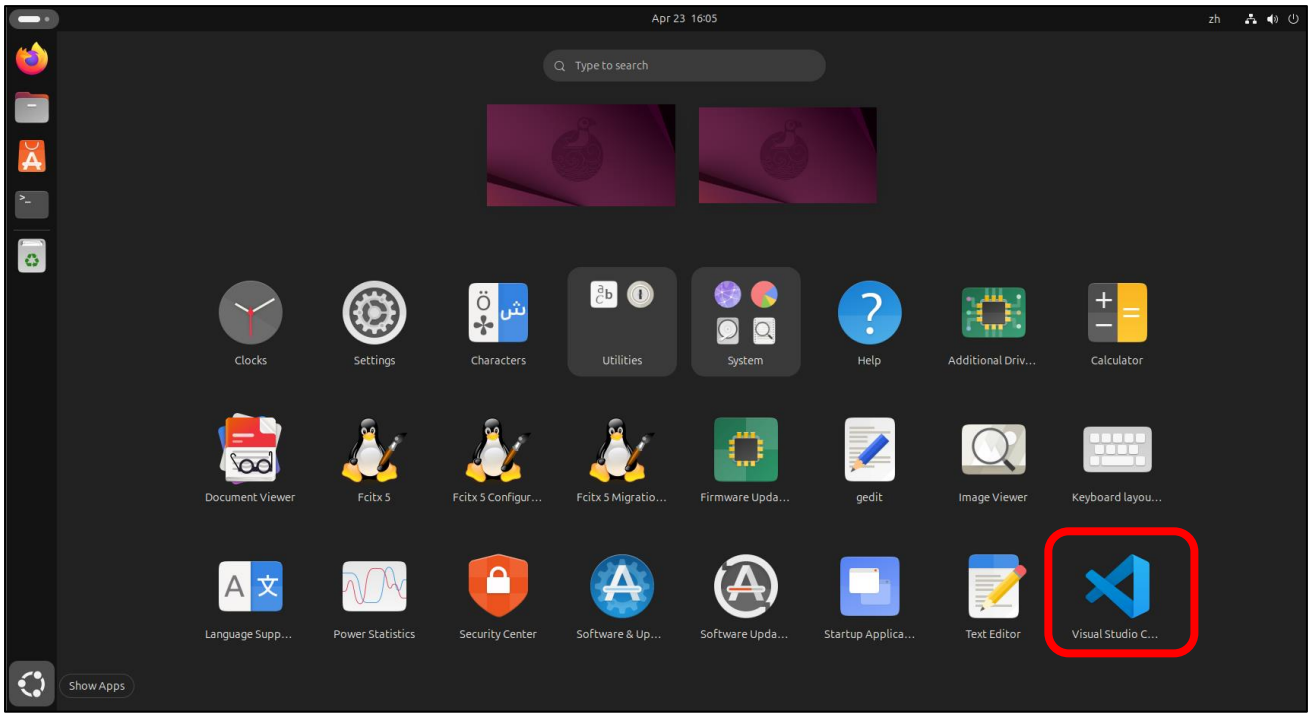
Click "Install" to install Visual Studio Code.



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.

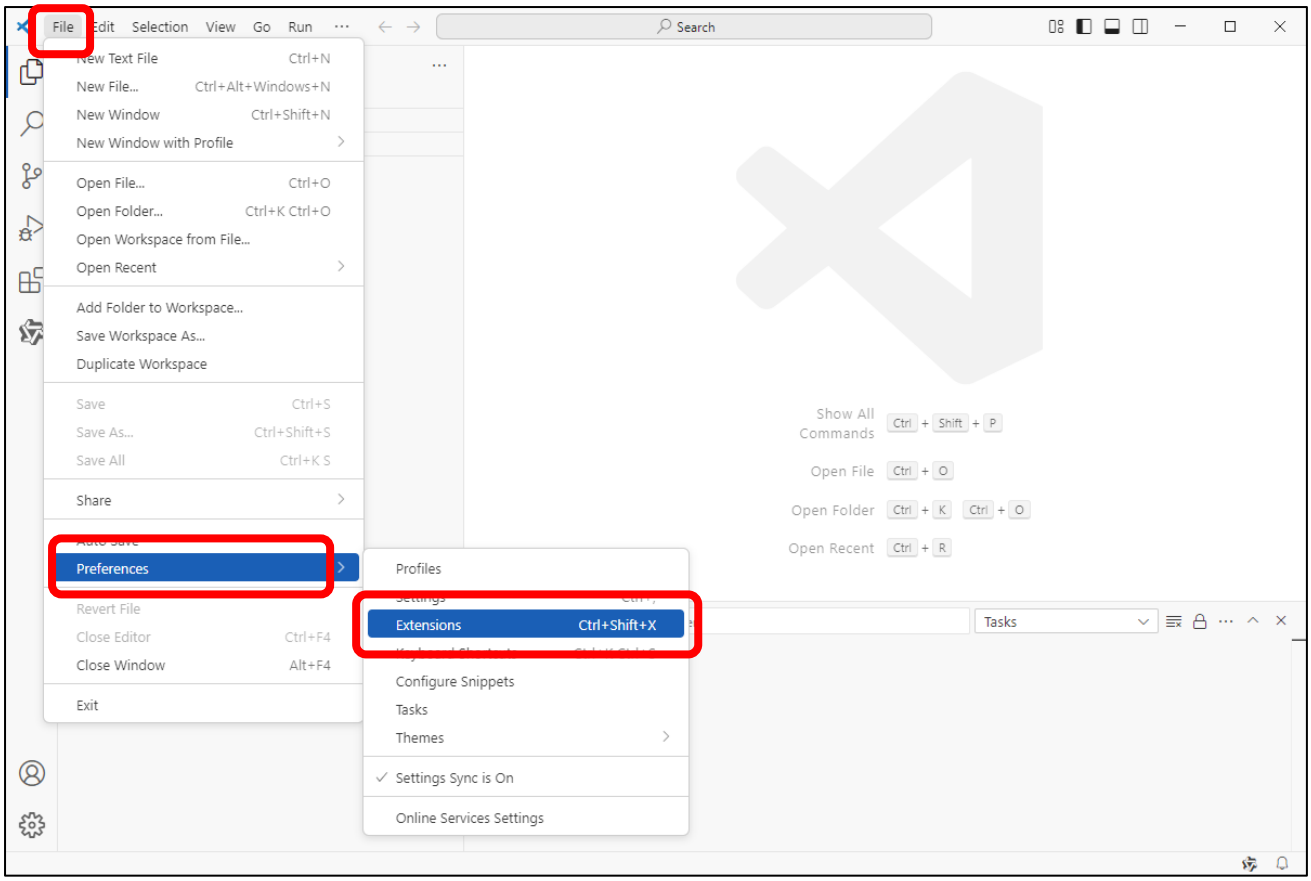


Installing ESP-IDF V5.3.2

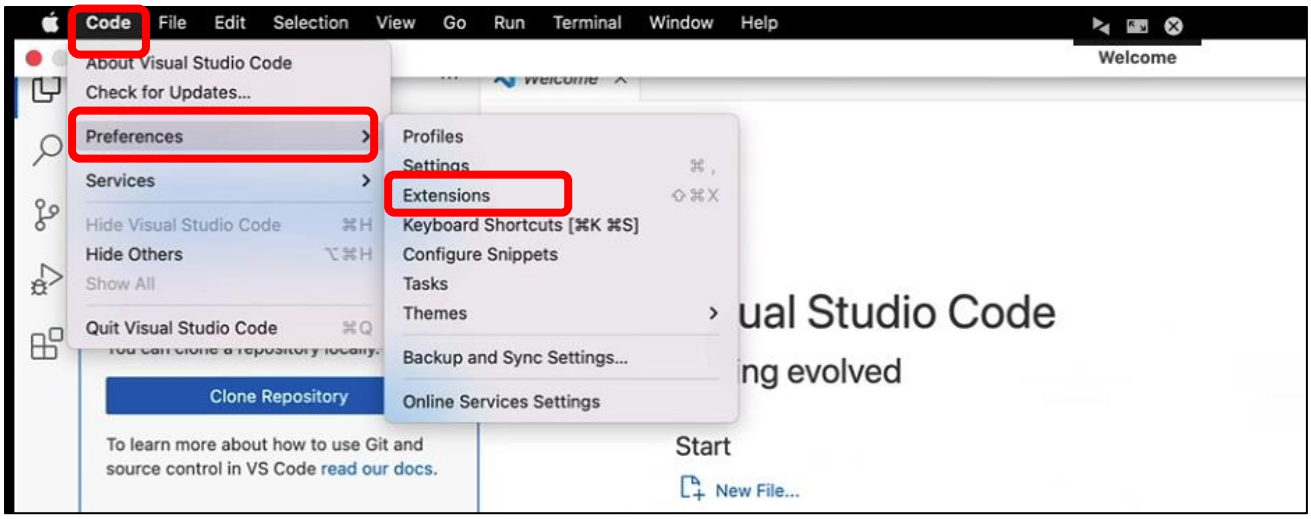
Visual Studio Code is a versatile code editor. To program with the ESP-IDF SDK, we need to install the ESP-IDF extension for it.

Open Visual Studio Code.

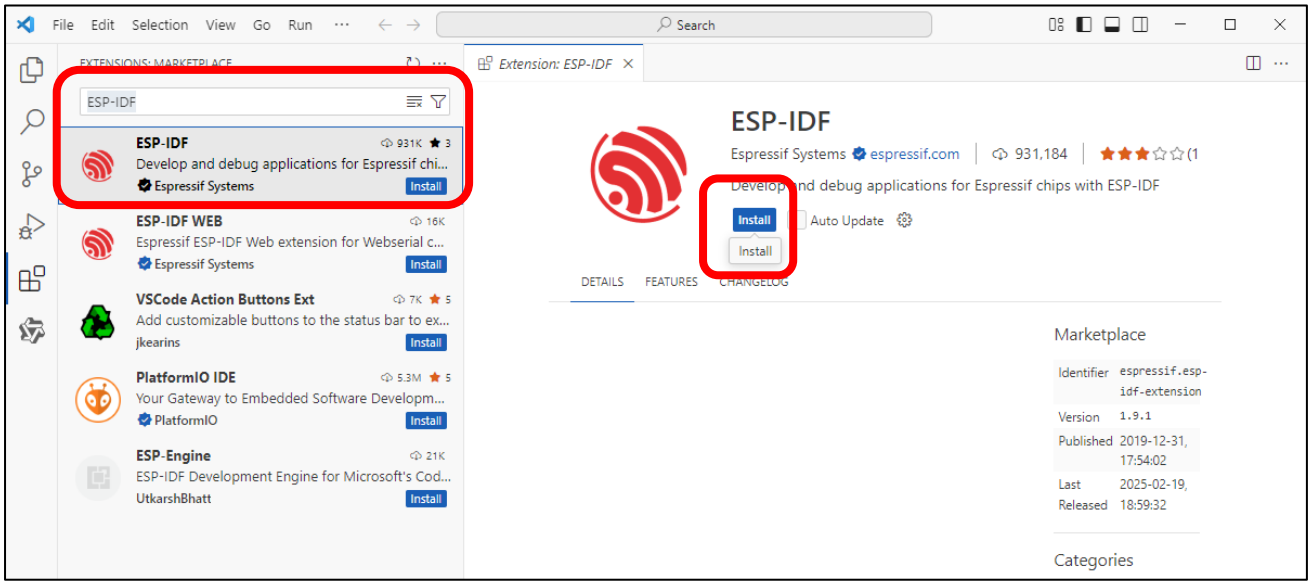
Click on the menu bar: File -> Preferences -> Extensions..



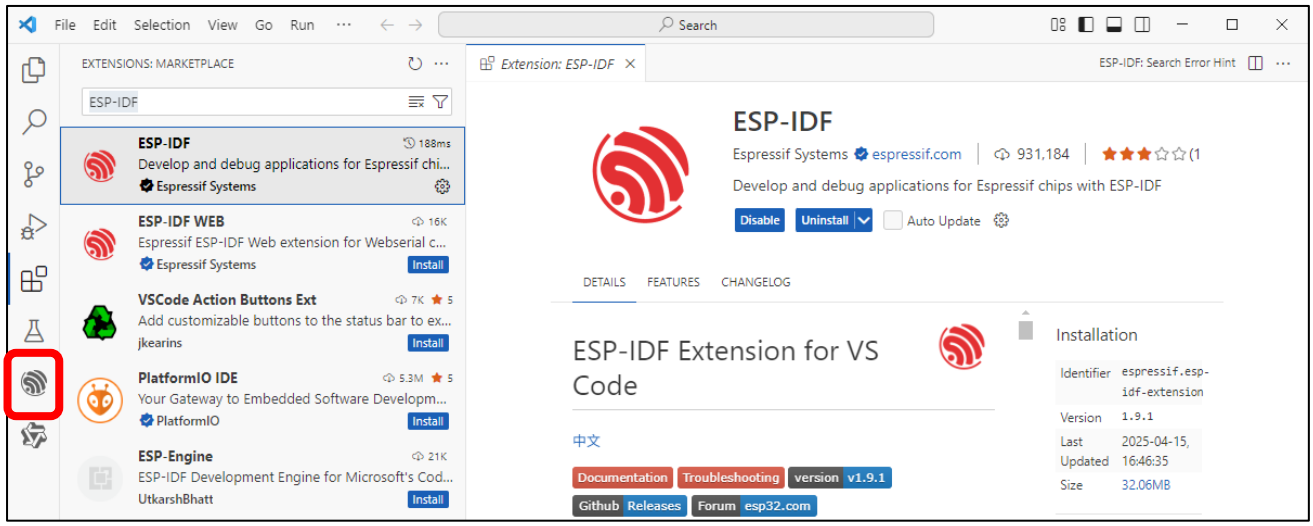
Mac OS: Click “Code” **->** ”Preferences” **->** ”Extensions” on the menu bar.



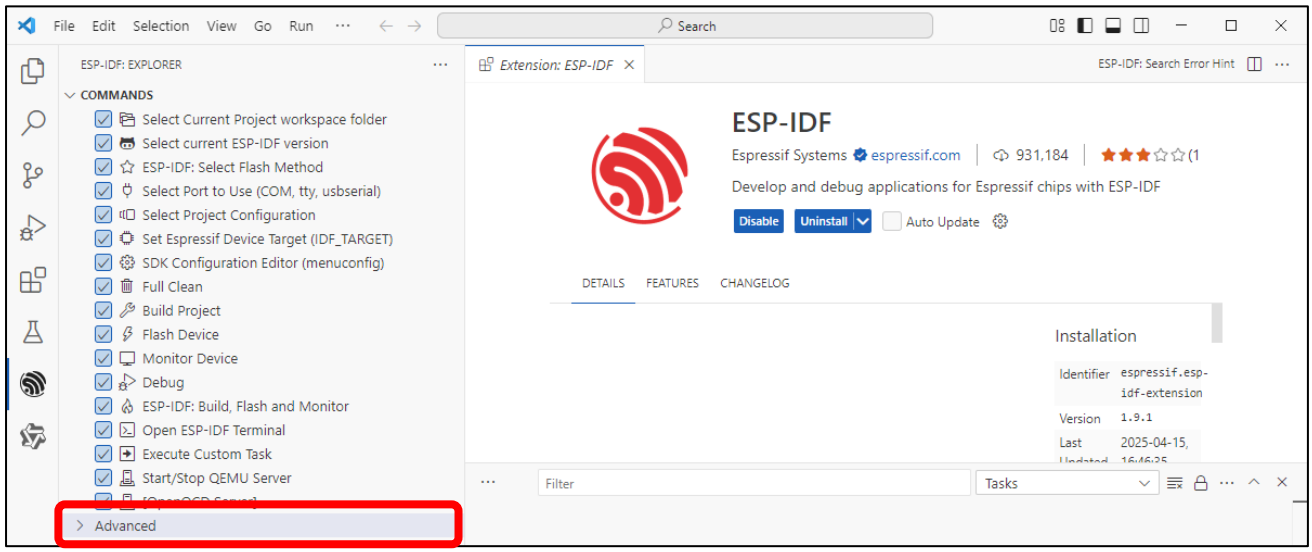
Search for "ESP-IDF" in the extension bar, select the correct result from the list, then click the Install button to proceed.



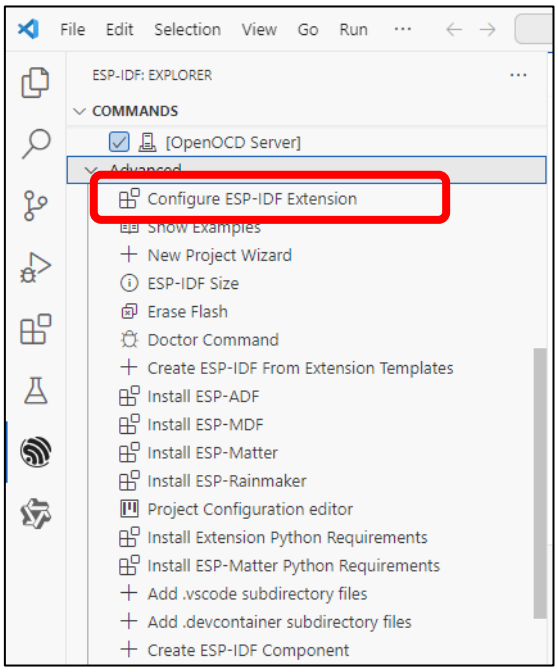
The ESP-IDF extension icon will now appear in the left sidebar - click it to continue.



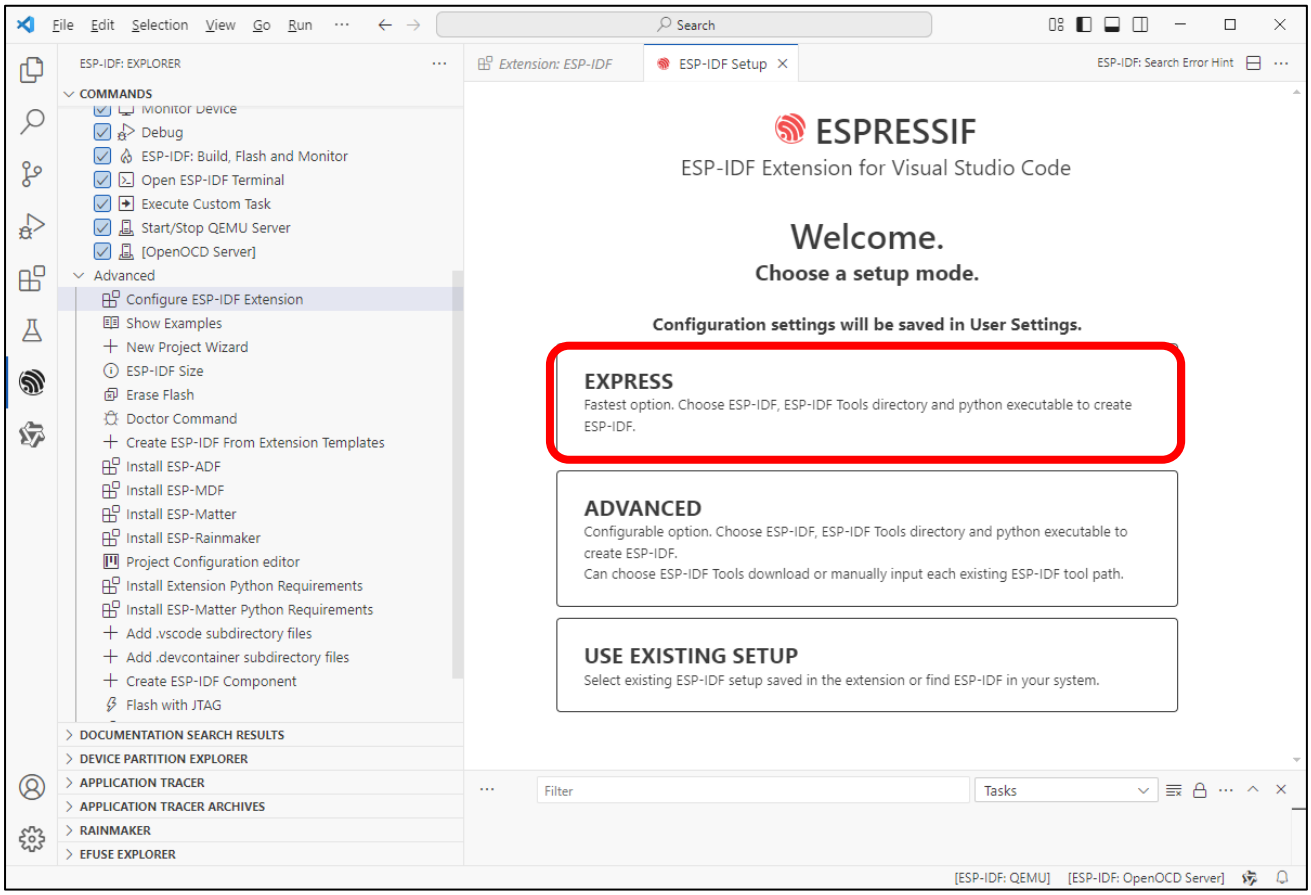
Scroll down with your mouse, locate and click on the "Advanced" option.



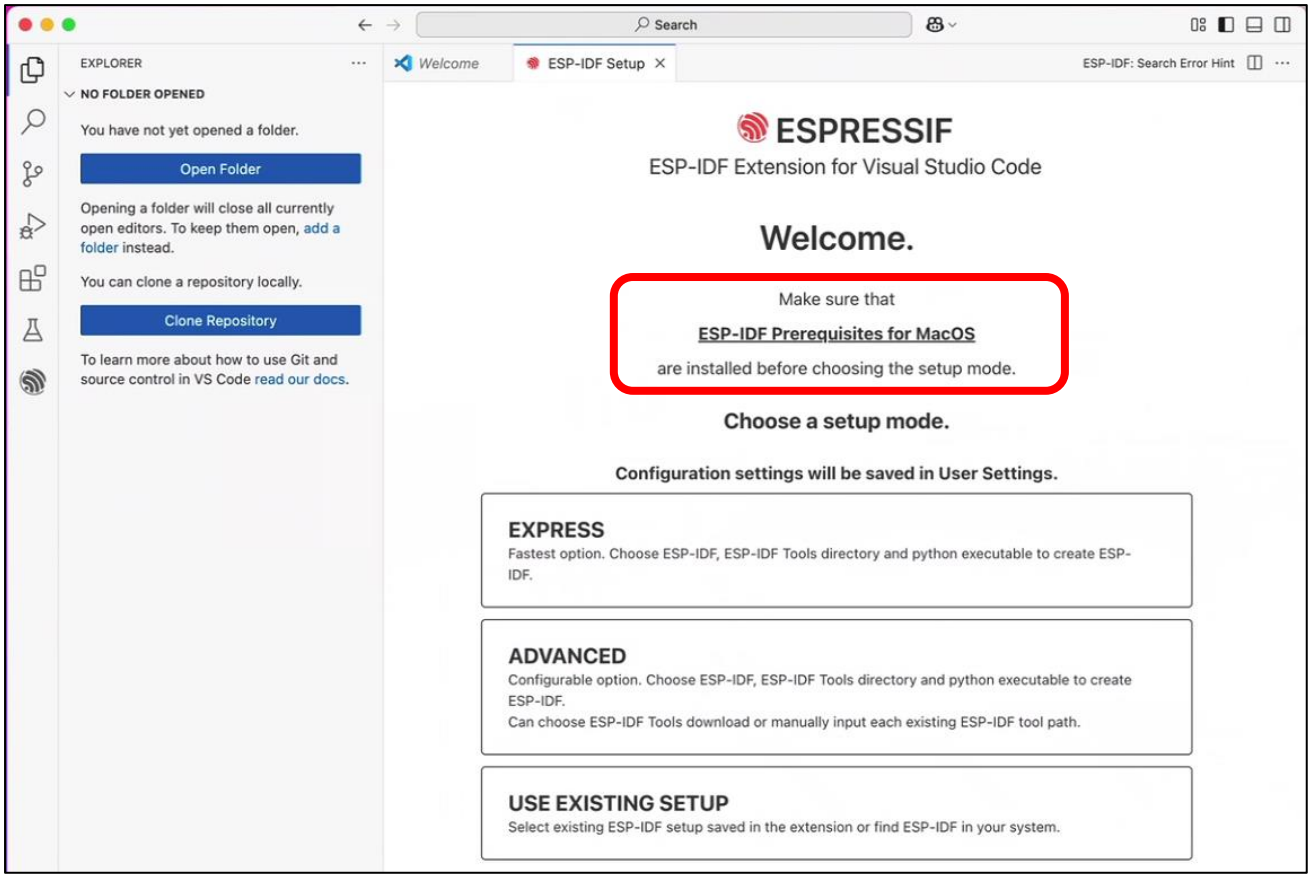
Click the first option: "Configure ESP-IDF Extension".

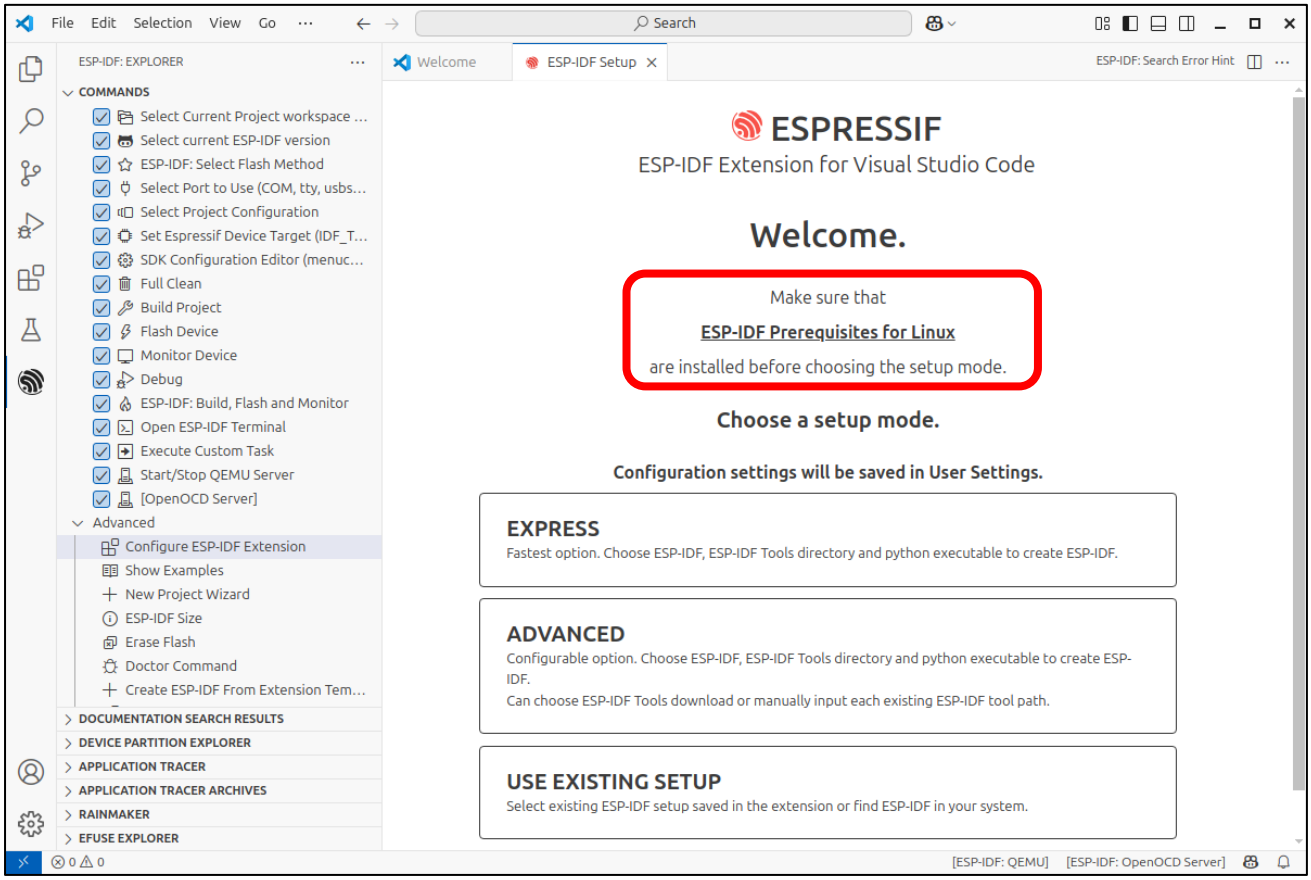


Select "EXPRESS" on the right.

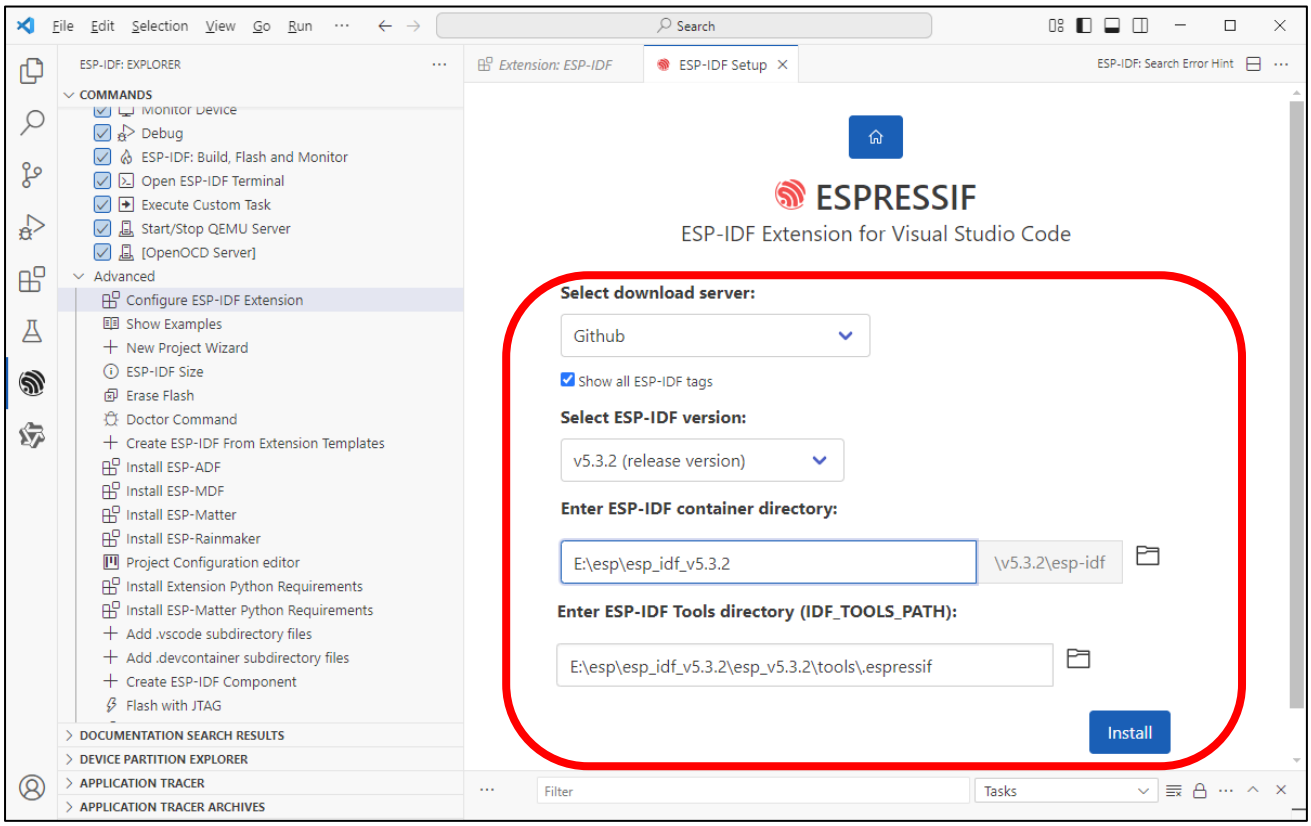


Note: If you're using macOS or Ubuntu, please complete the necessary preparations as prompted before proceeding with installation.



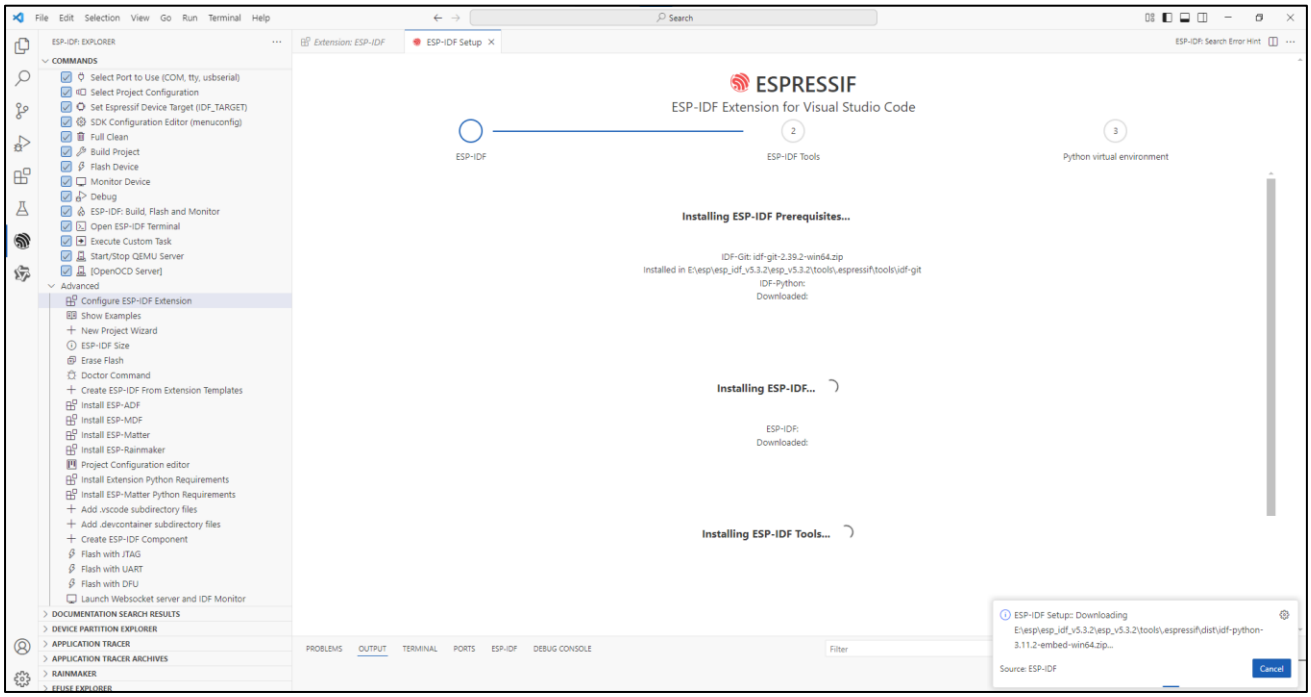


1. Check the box for "Show all ESP-IDF tags"
2. Select "v5.3.2 (release version)" from the dropdown
3. Choose your desired installation path for the ESP-IDF environment
4. Click "Install" to begin the setup



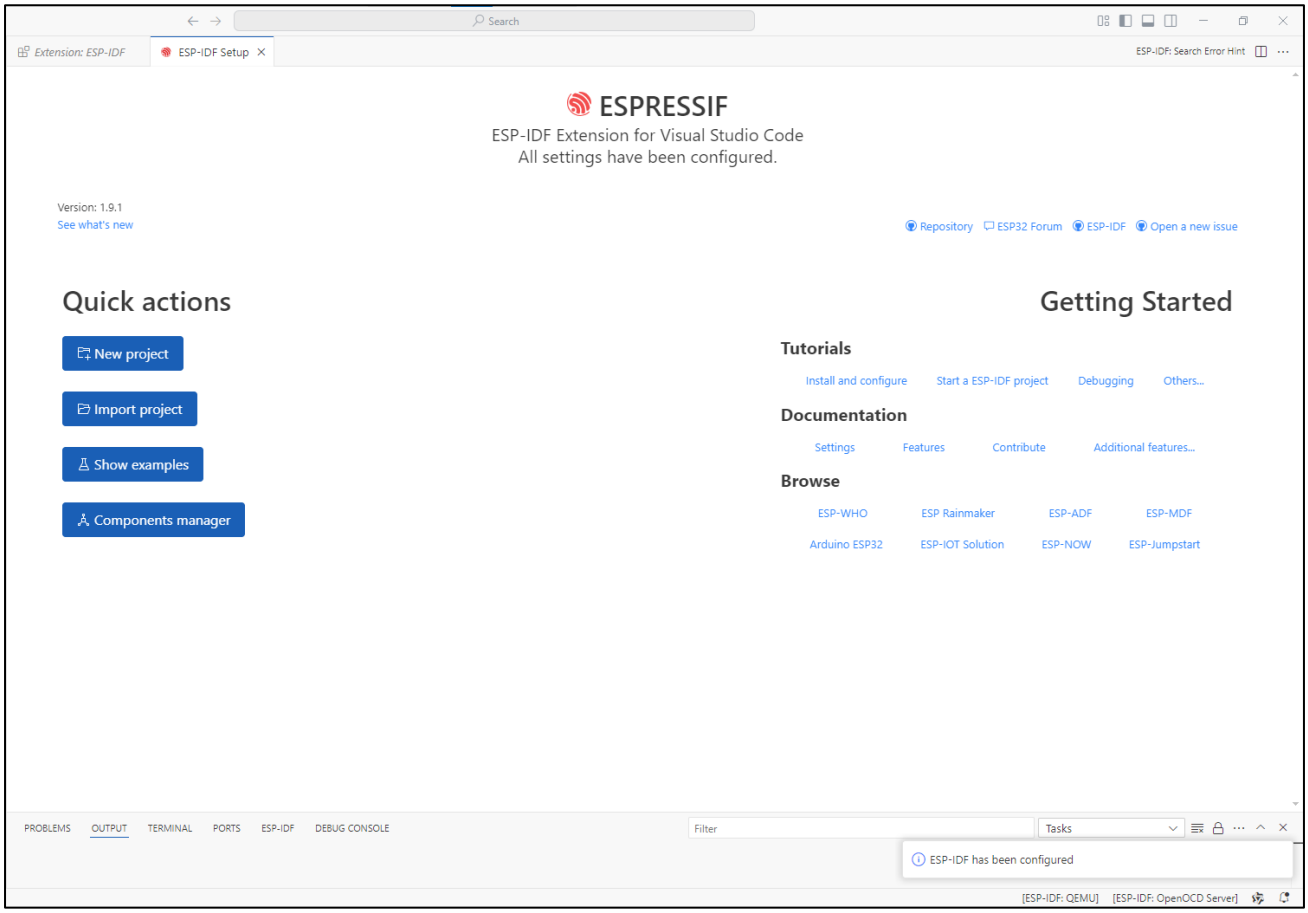
The process will complete automatically.

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.

The complete installation is as shown below.



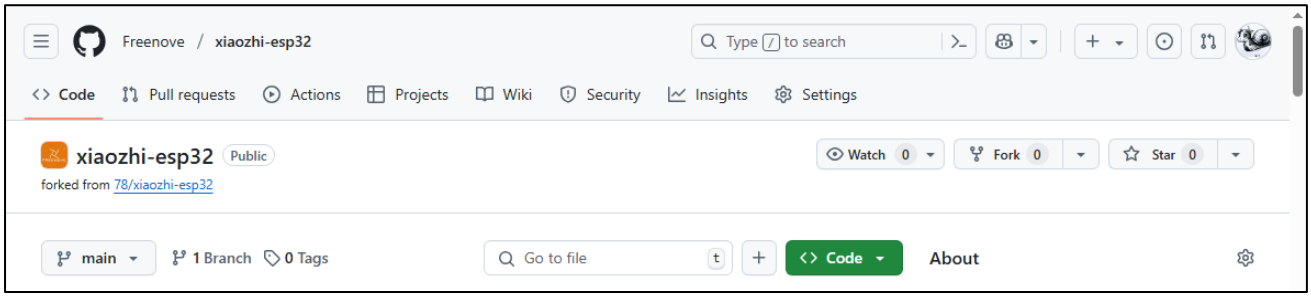
For more about ESP-IDF, please refer to

https://docs.espressif.com/projects/vscode-esp-idf-extension/en/latest/installation.html

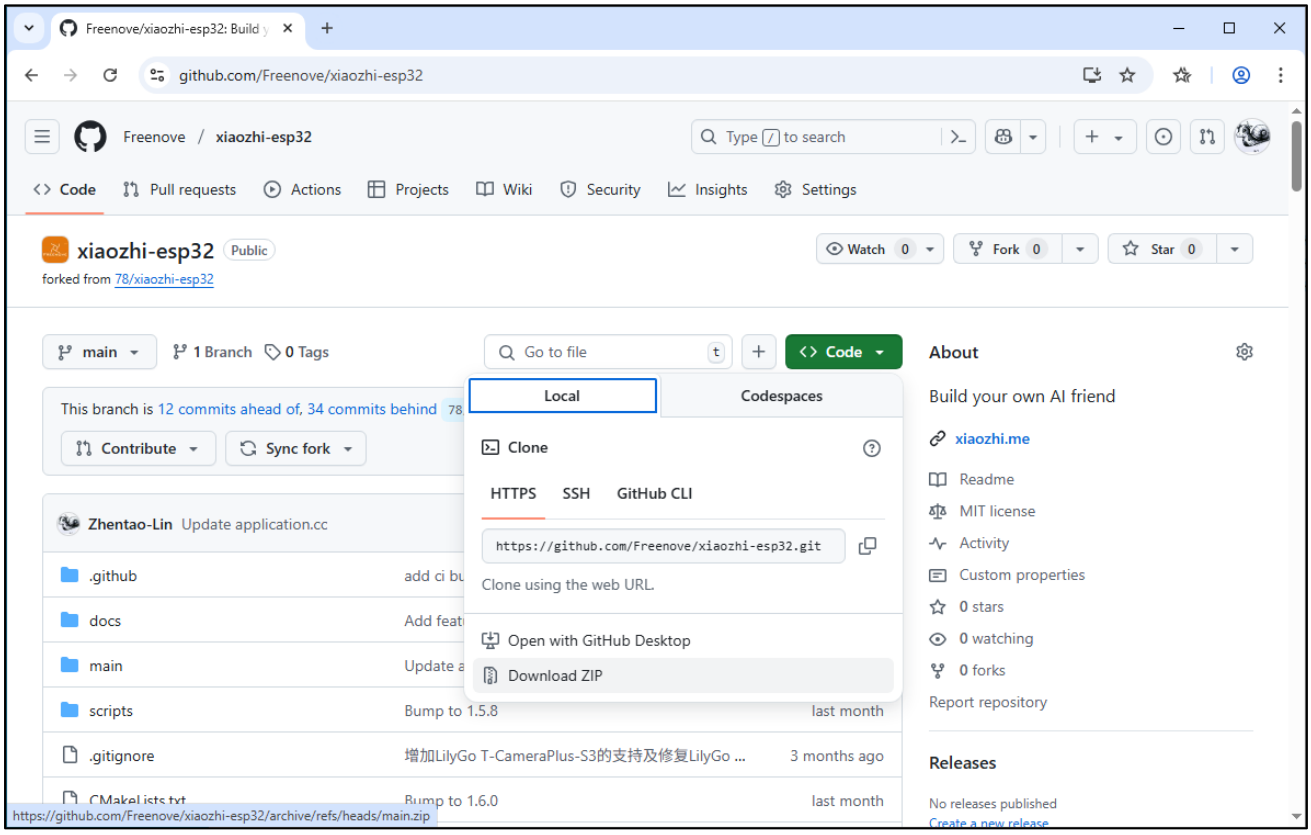
Code Downloading

Windows

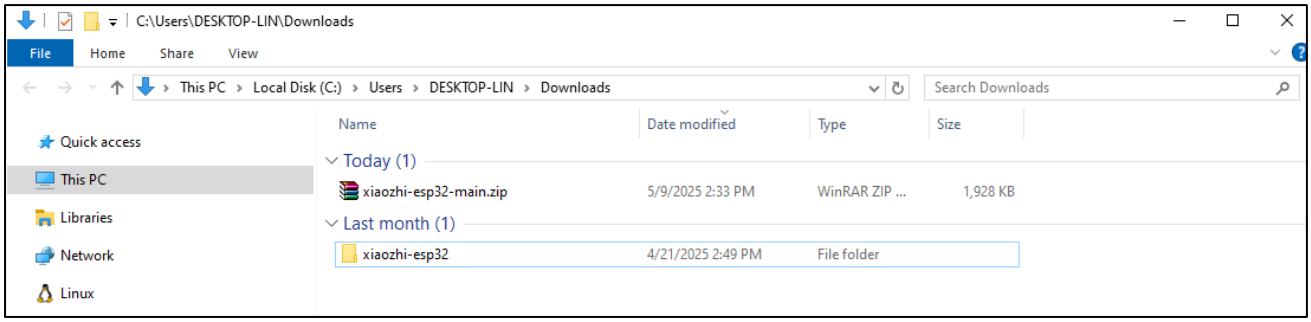
Open a browser on your computer and enter "https://github.com/Freenove/xiaozhi-esp32".



Click "Code" -> "Download ZIP" to download the code to your computer.



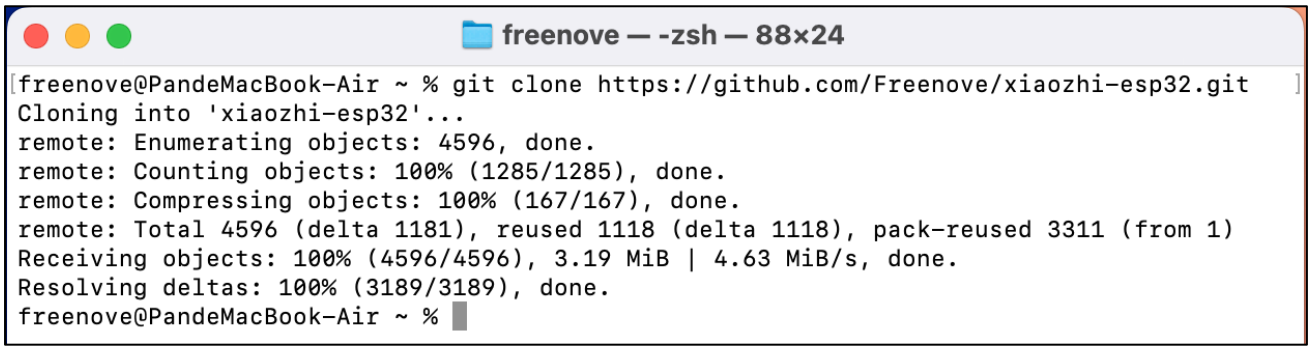
Extract the downloaded zip file to your computer. **Rename the decompressed folder to "xiaozhi-esp32".**



Mac

Open the terminal and download the code with the git command.

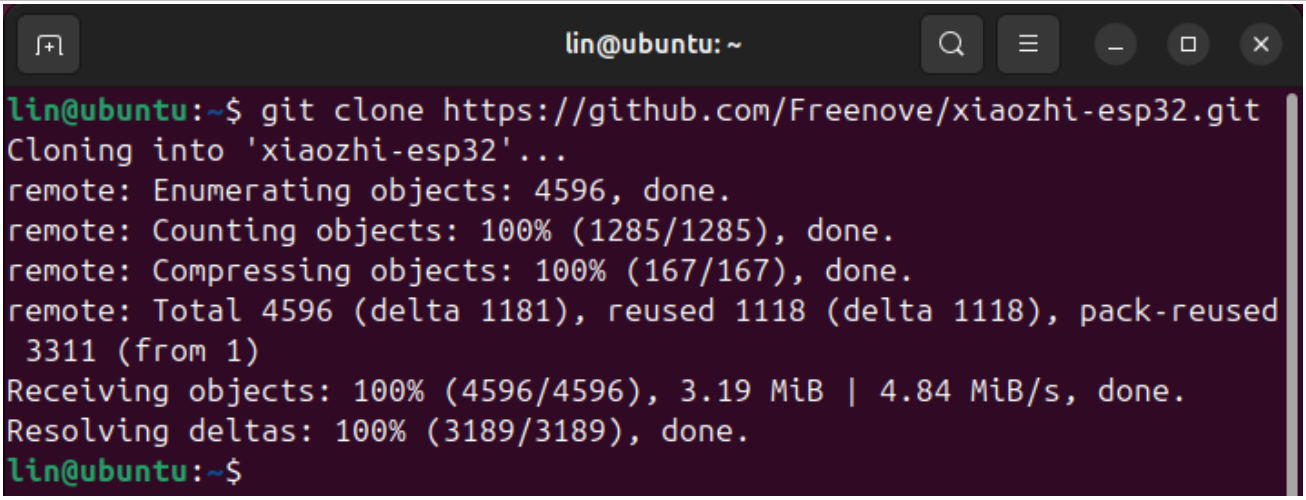
```
git clone https://github.com/Freenove/xiaozhi-esp32.git
```



Linux

Open the terminal and download the code with the git command.

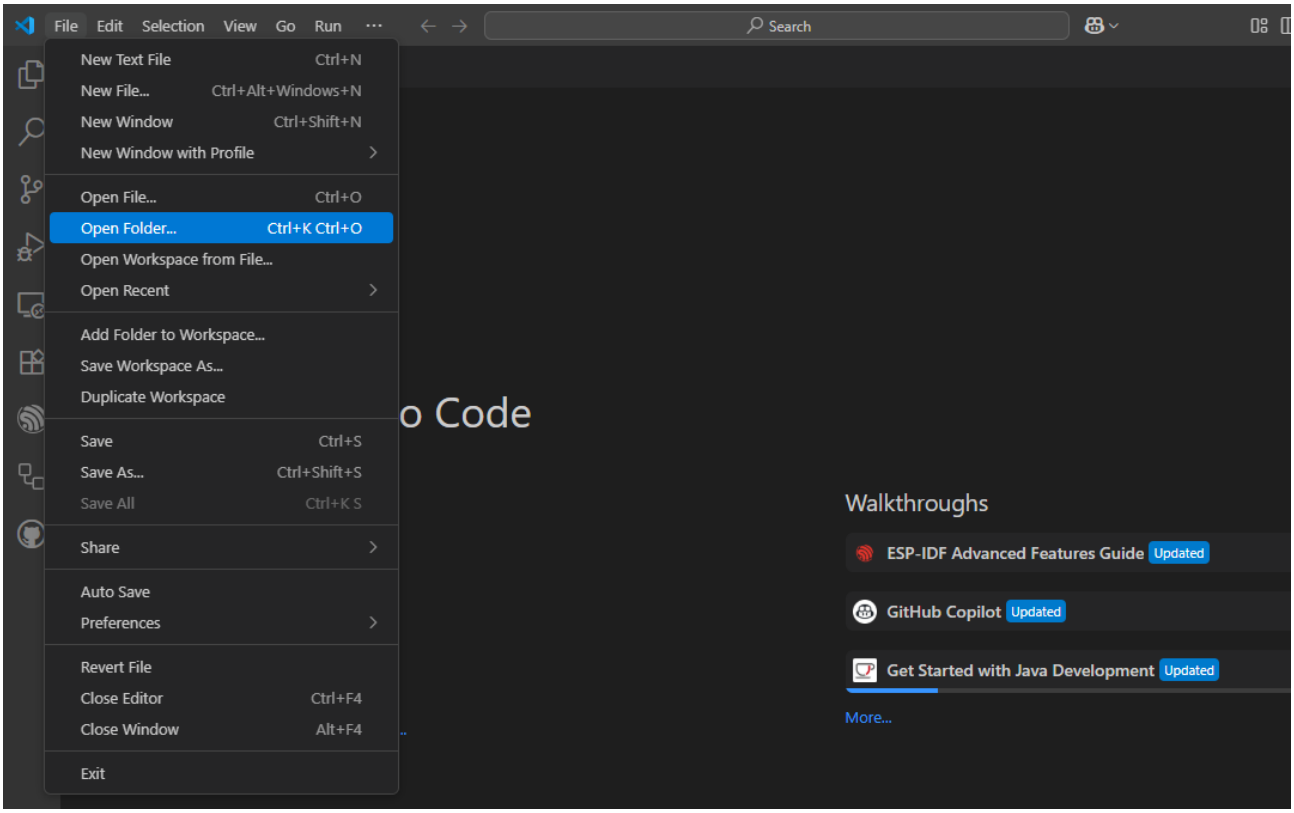
```
git clone https://github.com/Freenove/xiaozhi-esp32.git
```



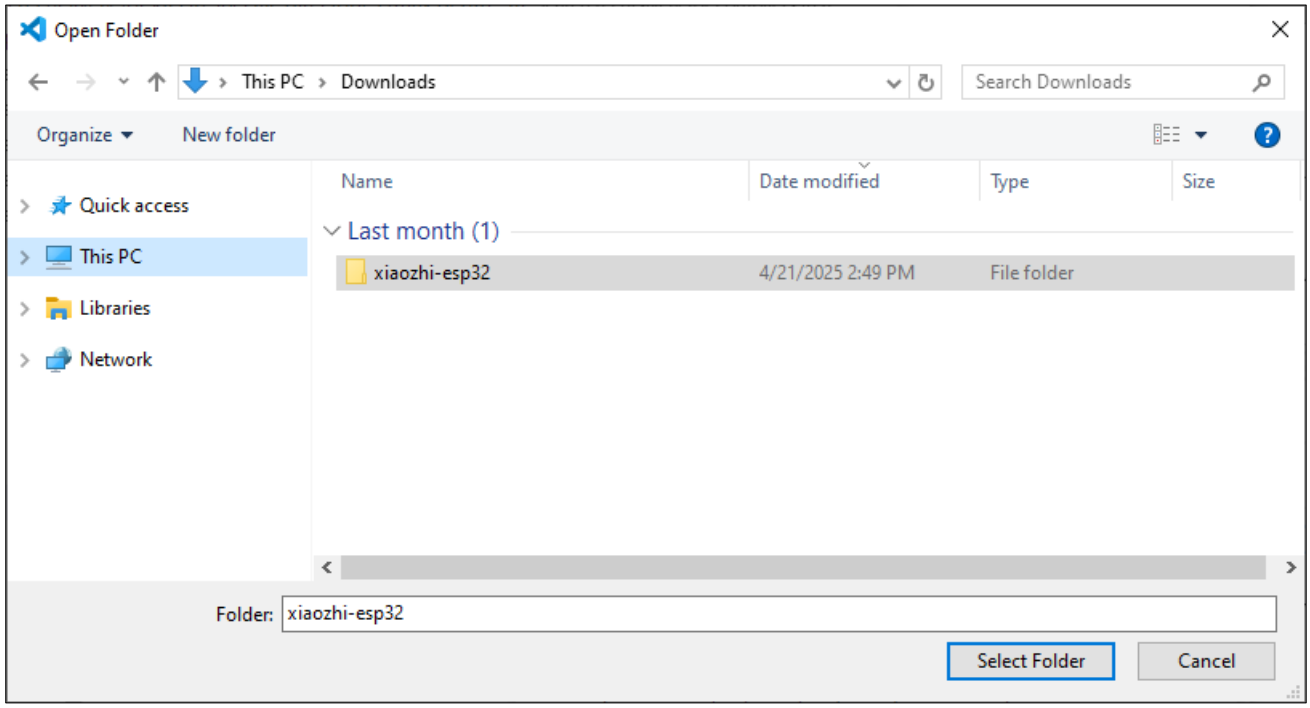
Configure Code Environment

Extract the downloaded ZIP file.

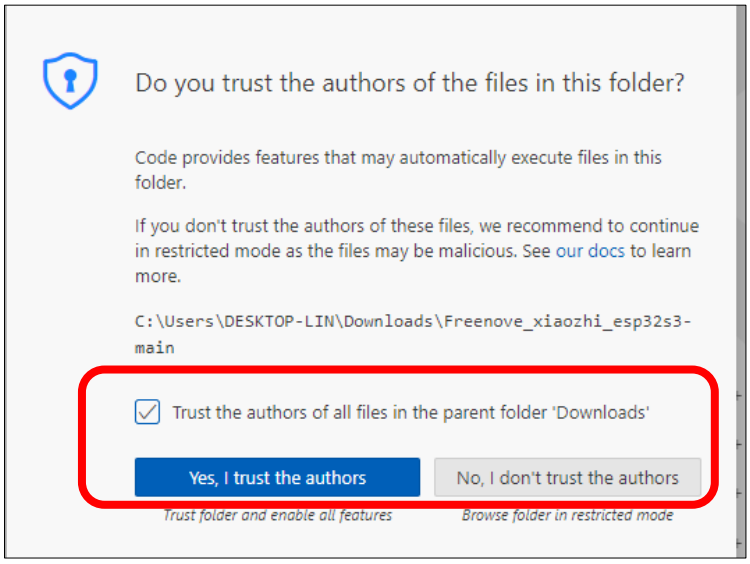
On Visual Studio Code, click “File” -> “Open Folder…”.



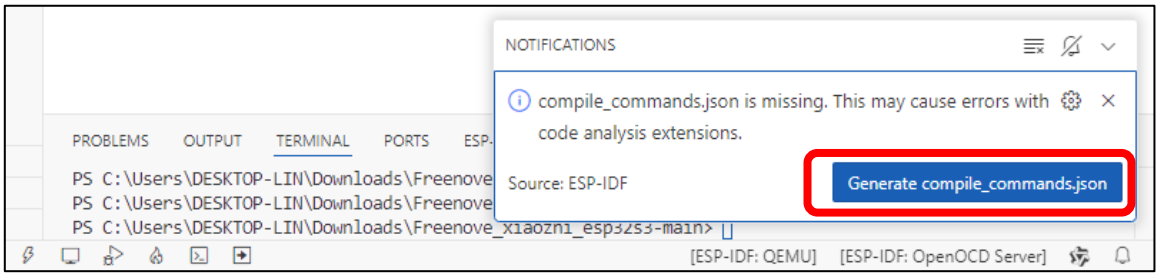
Select the **xiaozhi-esp32** folder. Here, the interface of the Windows system is taken as an example. The operation of the mac system is similar to that of Linux.



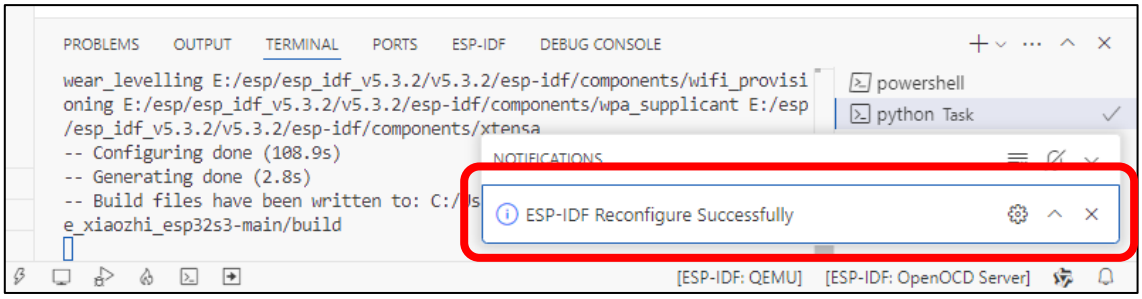
Check the box “Trust the authors of all files in the parent folder ‘Downloads” and select “Yes, I trust the authors”.



Please note: A pop-up notification will appear in the lower-right corner. Click 'Generate comple_commands.json', and it will download the corresponding component module code based on the file."

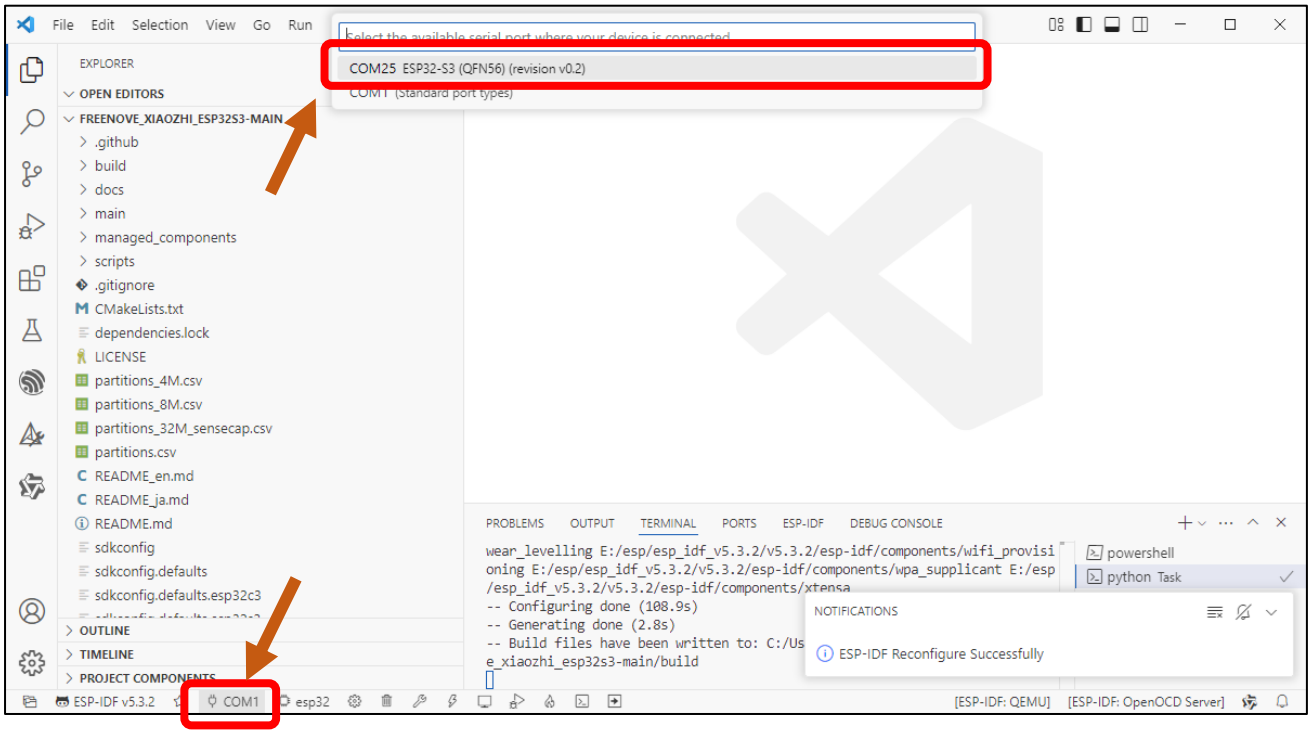


Component installation may take some time. Please wait and avoid other operations. A completion notification will appear in the lower-right corner once finished.

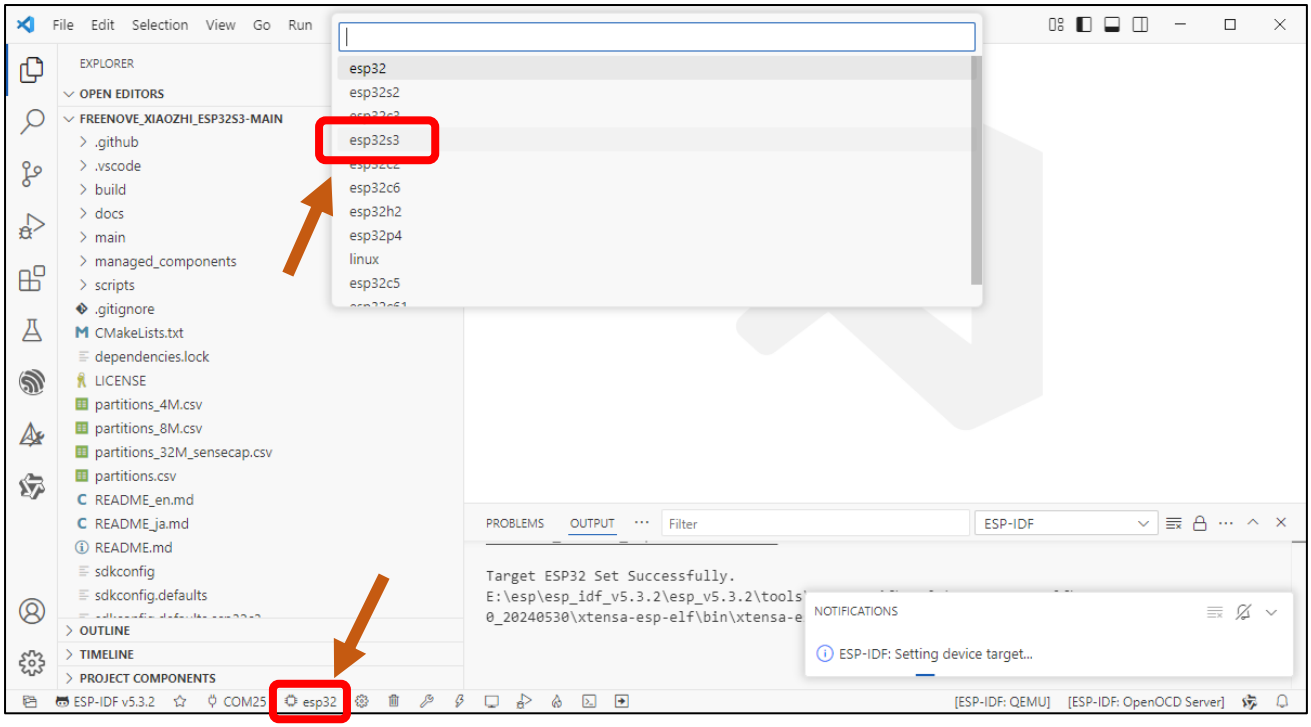


Connect the ESP32-S3-WROOM to your computer using a USB cable, making sure to plug it into the correct Type-C port (do not use the wrong connector)

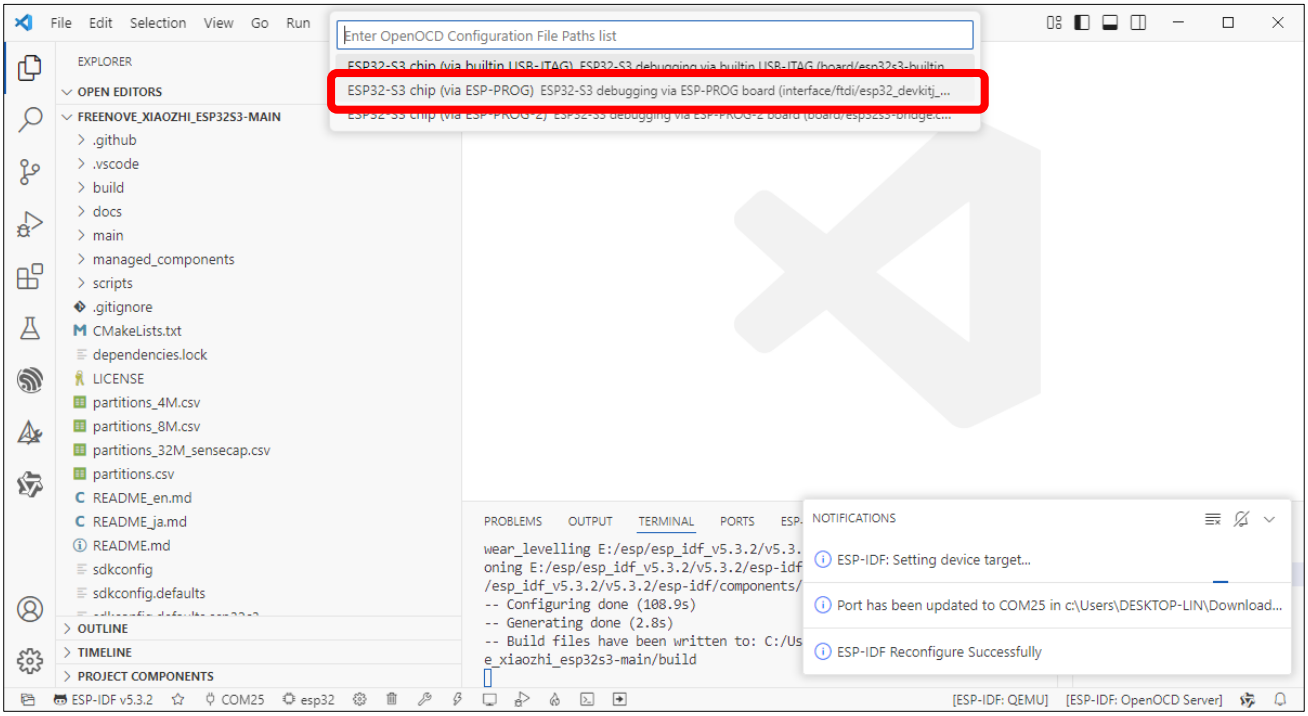
Click on '**COMx**' in the bottom-left corner to display all available COM ports on your computer. Locate and select the entry labeled 'ESP32-S3'.



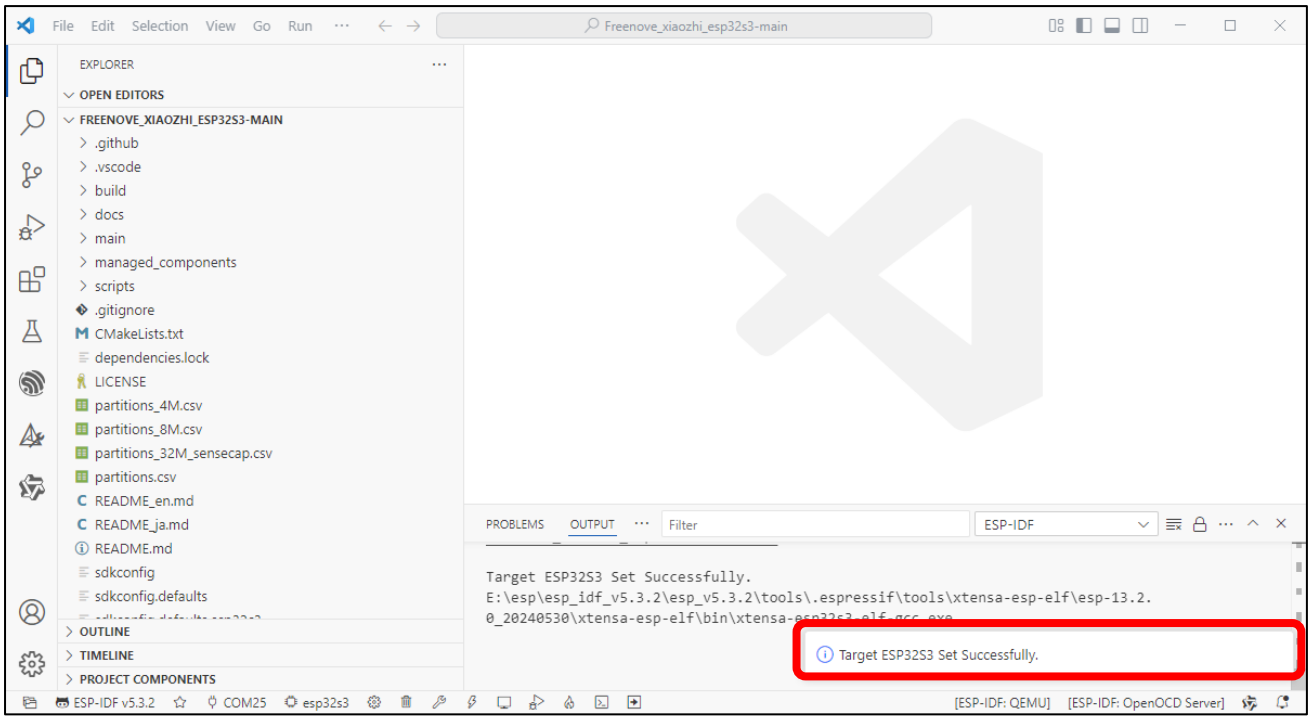
Click the '**ESP32**' button in the bottom-left corner to display all available ESP32 models, then select **'ESP32-S3**' from the list."



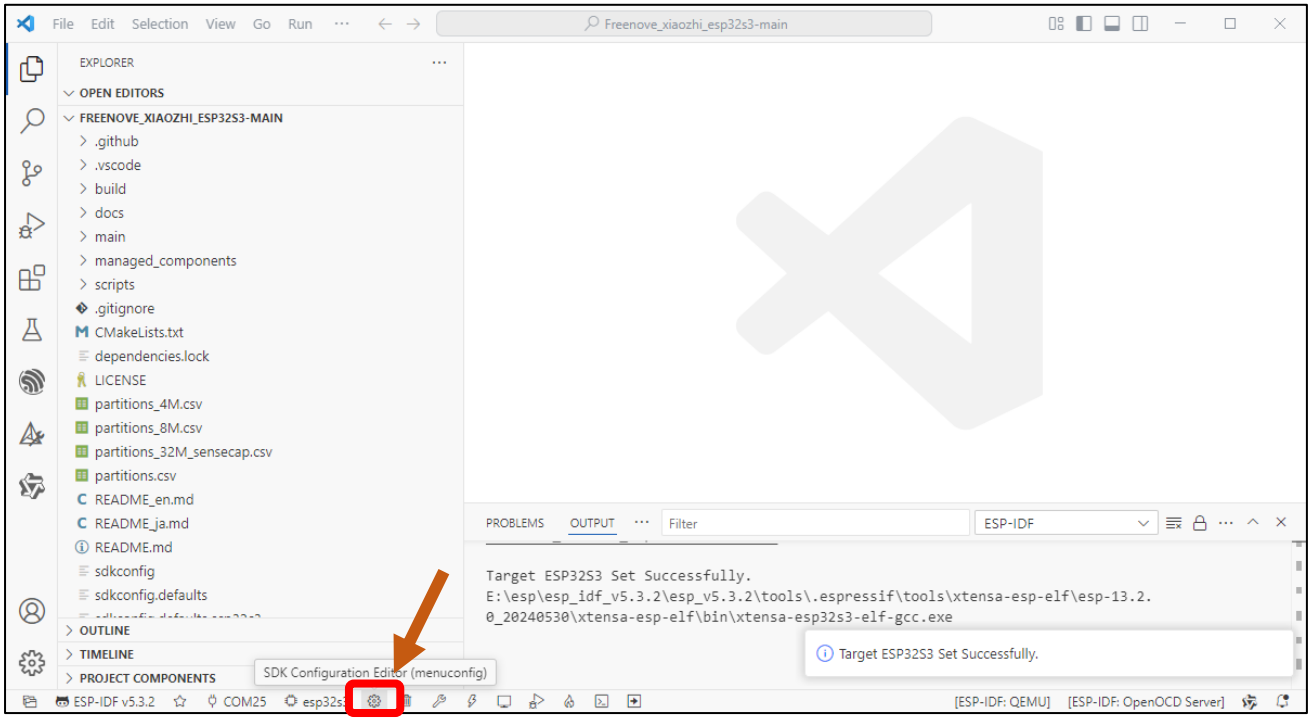
From the new selection menu, choose '**ESP32-S3 Chip (via ESP-PROG) - ESP32-S3 debugging via ESP-PROG Board...**'



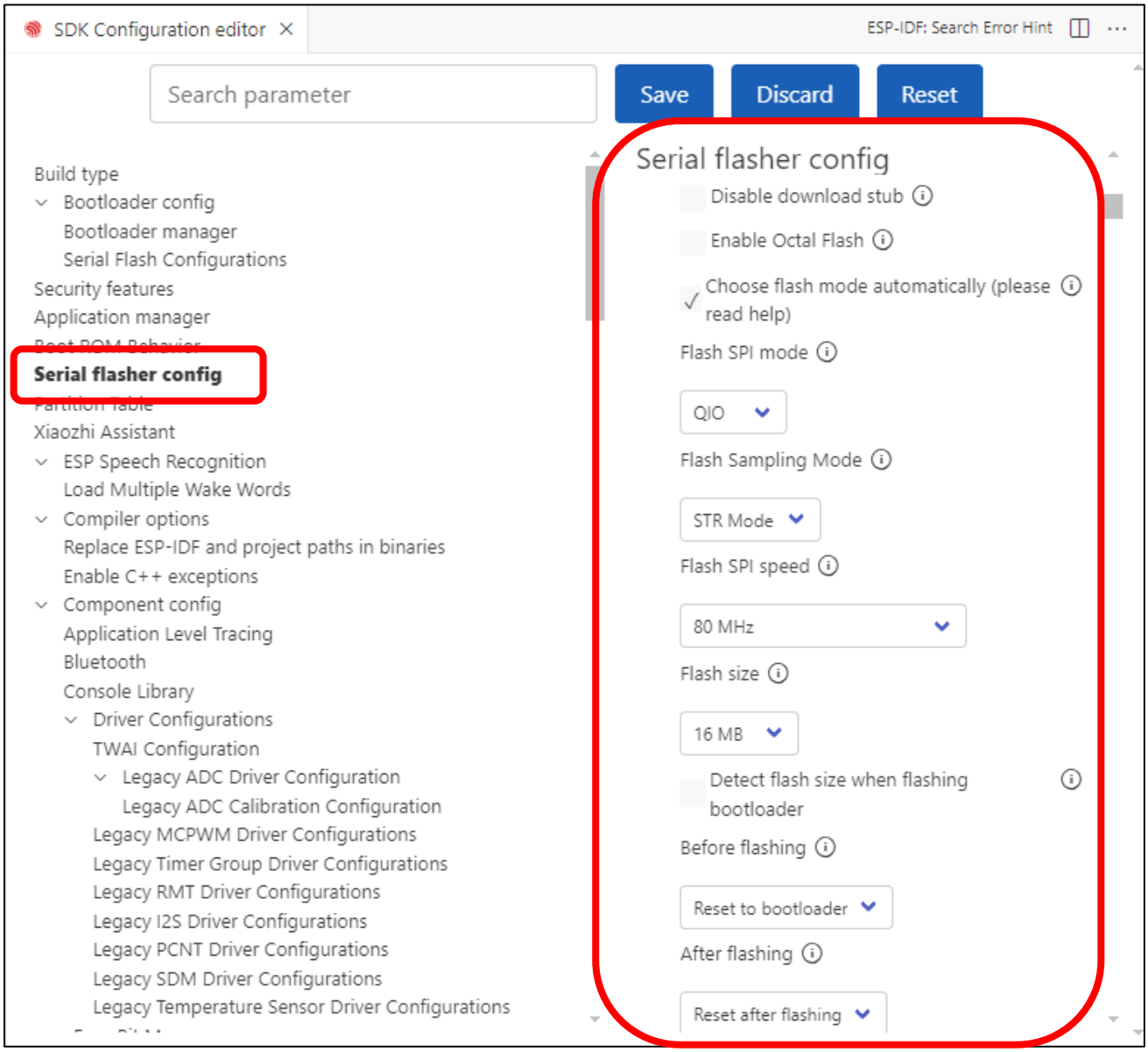
Wait until it shows "**Target ESP32S3 Set Successfully**" at the bottom right.



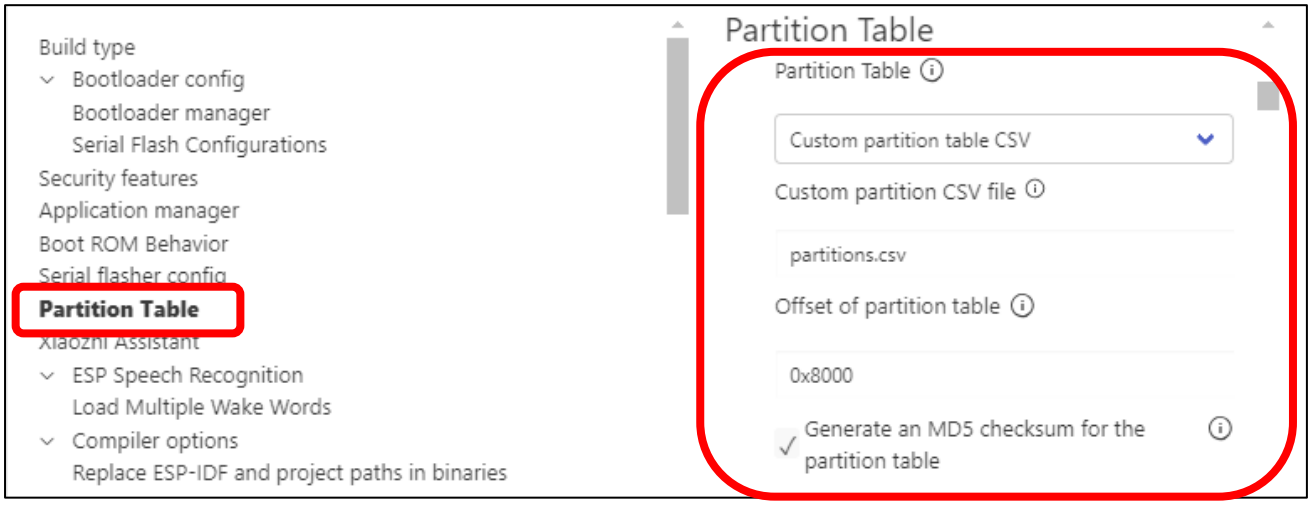
Click "SDK Configuration Editor (menuconfig)" at the bottom.



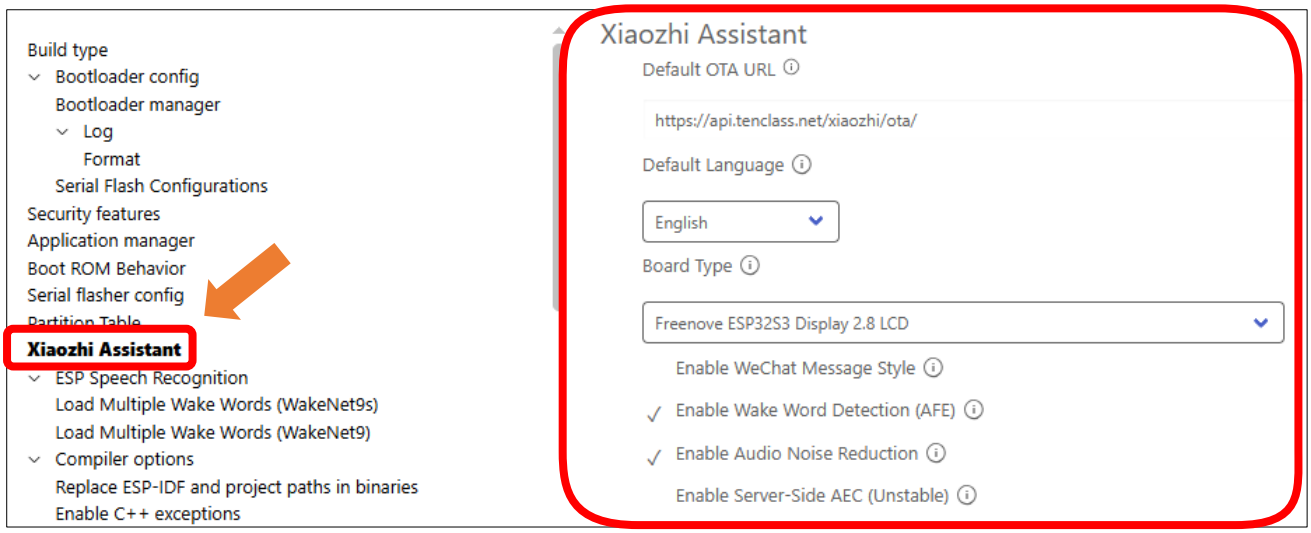
On the new interface, click 'Serial flasher config' and verify that the settings match the configuration shown in the image below.



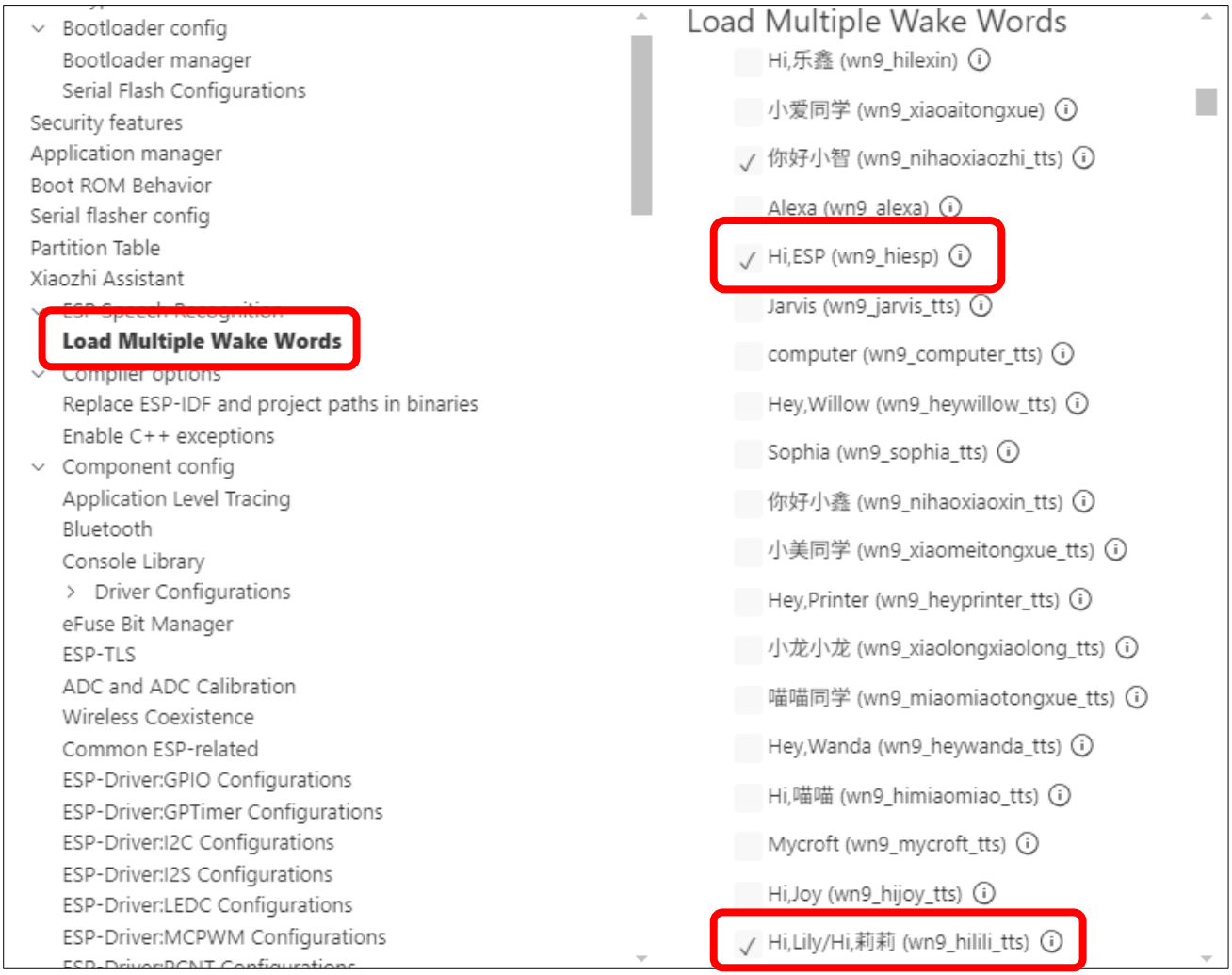
Click "Partition Table" and verify that the settings match the configuration shown in the image below.



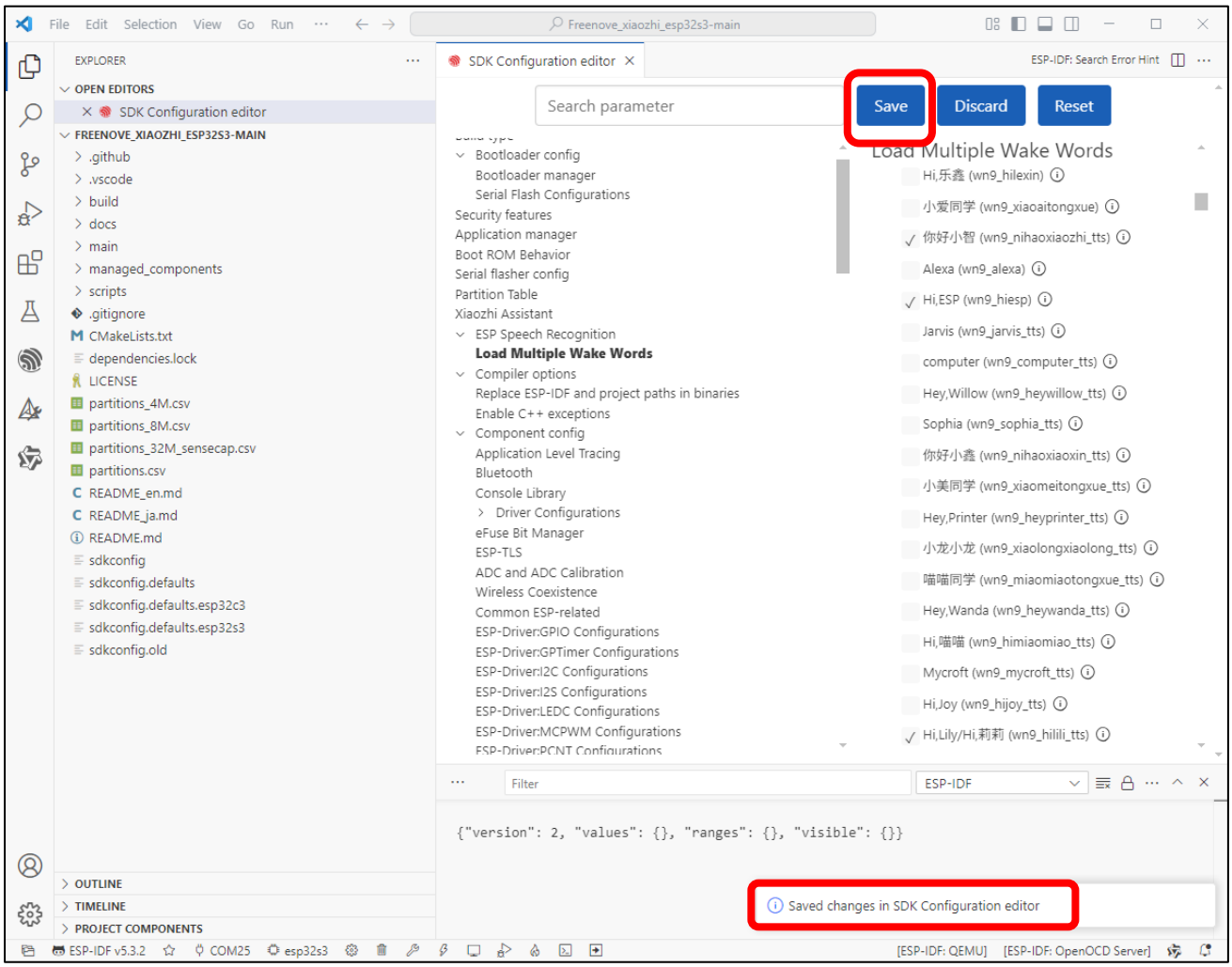
Click **"Xiao Assistant"** and verify that the settings match the configuration shown in the image below.



Click **"Load Multiple Wake Words"** and check the boxes for 'Hi, ESP' and 'Hi, Lily' (and other desired options).



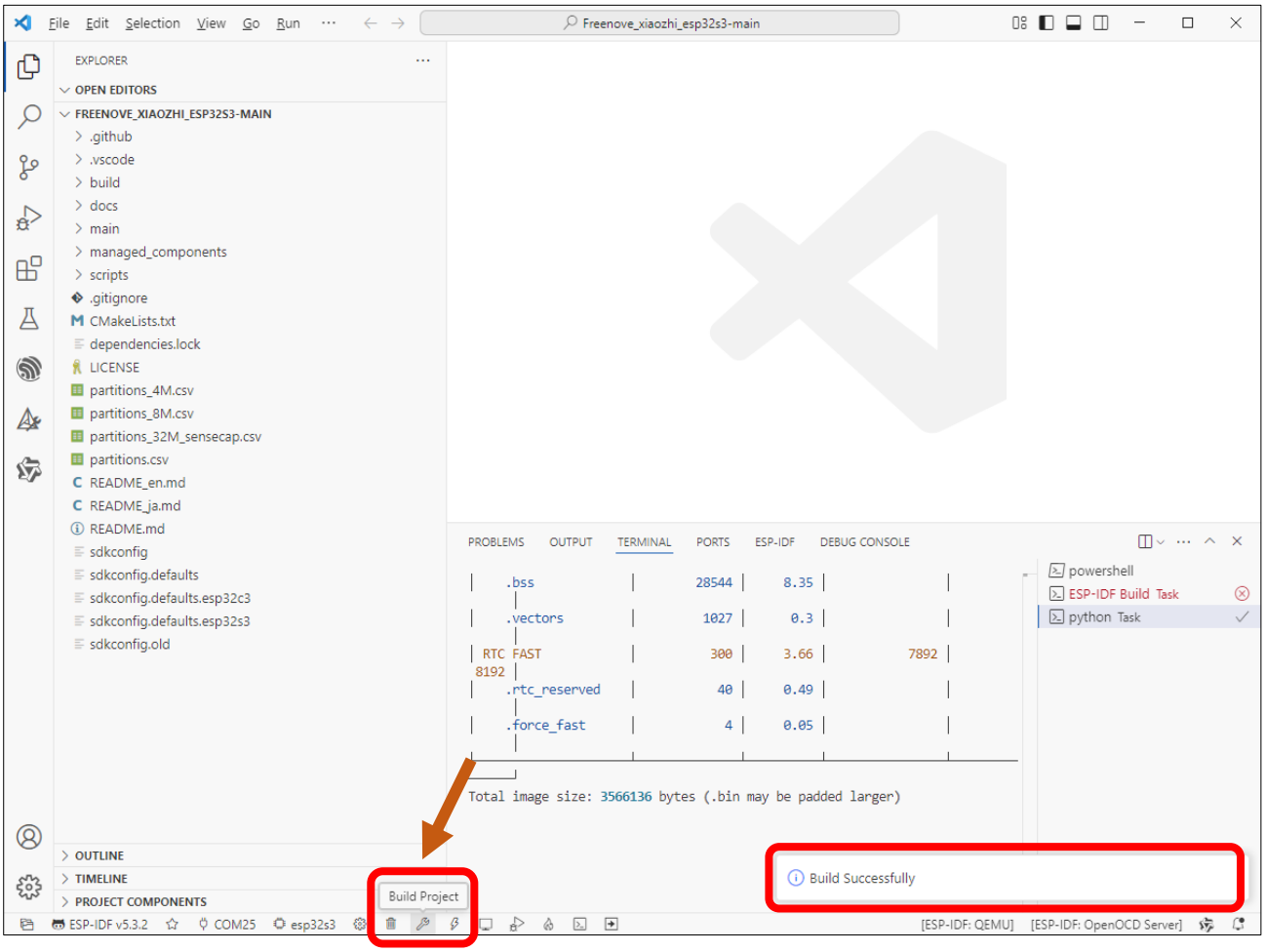
Finally, click **"Save"** to store your configuration. A success message will appear at the bottom upon completion.



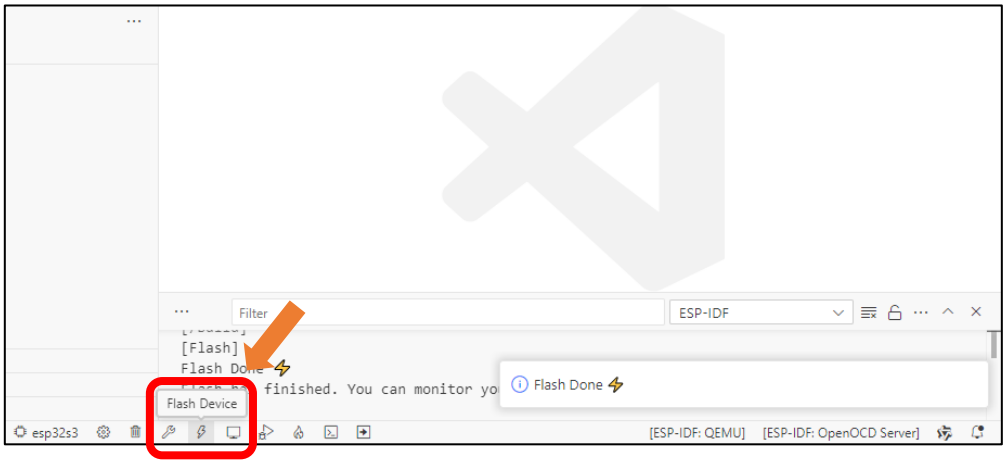
Code Compilation

Before compiling, make sure all aforementioned configurations are correct. Click the 'Full Clean' button (bottom toolbar) to reset build cache.

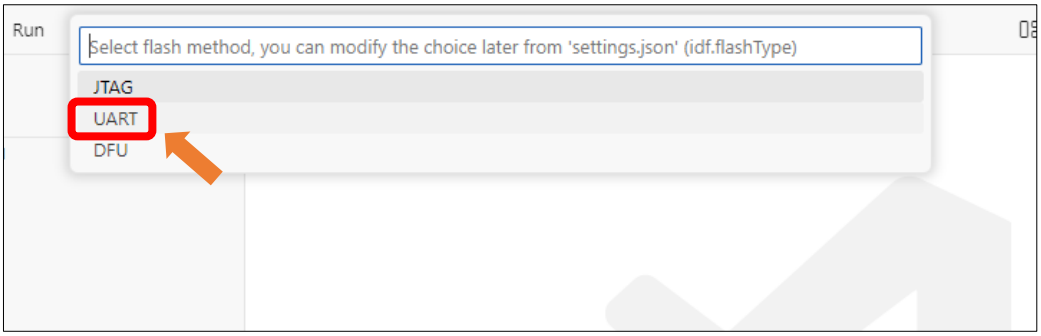
Click 'Build Project' at the bottom to start compiling the entire project. The first compilation may take longer - please wait patiently until the success message appears in the output panel.



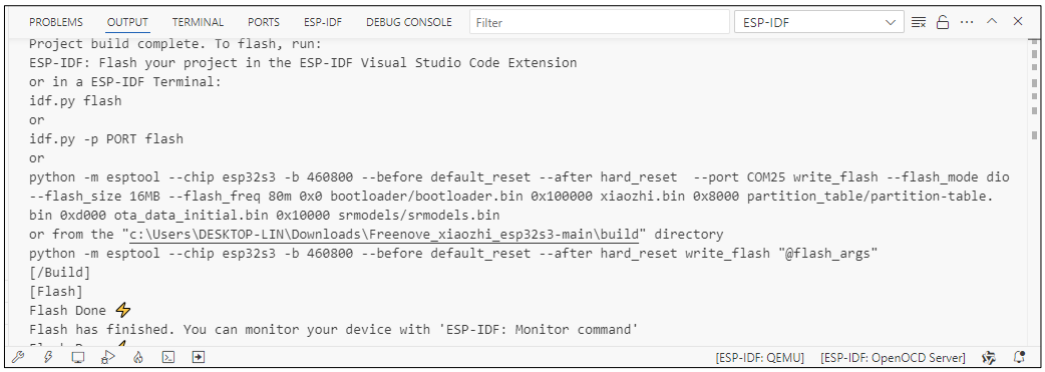
Click 'Flash Device' at the bottom to start uploading the code to your ESP32-S3-WROOM module.



From the new options menu, select '**UART**' and wait for the code upload to complete.



Upon seeing the message 'Flash has finished. You can monitor your device with "ESP-IDF: Monitor command"', this indicates you have successfully uploaded XiaoZhi AI's firmware to the ESP32-S3-WROOM module.



At this point, the compilation is complete and you're ready for secondary development.

Local Server

Disclaimer

This project is derived from the open-source repository: https://github.com/xinnan-tech/xiaozhi-esp32-server, licensed under MIT License. The XiaoZhi AI firmware operates on servers provided by Xiage's company. 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.

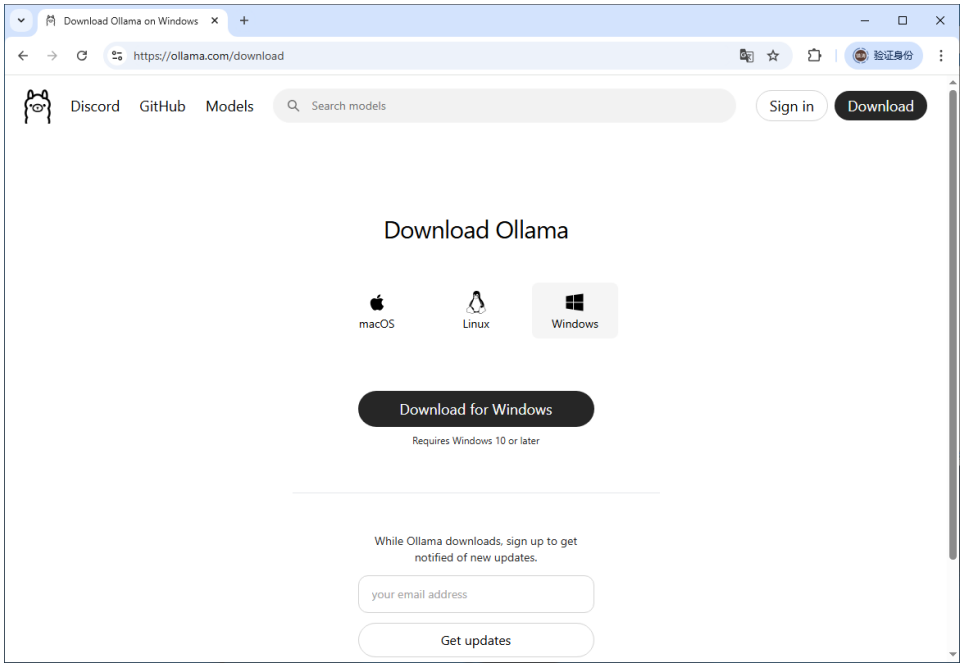
Deploying XiaoZhi AI on a Local Server

If you prefer not to use the XiaoZhi AI server, you can also set up a simplified version of the server on your own computer. In this section, we will use the open-source project at https://github.com/xinnan-tech/xiaozhi-esp32-server to deploy a local server and establish a connection with the ESP32 S3 WROOM.
If you encounter any bugs in the code during use, please submit an issue at https://github.com/xinnan-tech/xiaozhi-esp32-server. Please note that we do not have an in-depth understanding of this project and may not be able to provide extensive assistance.

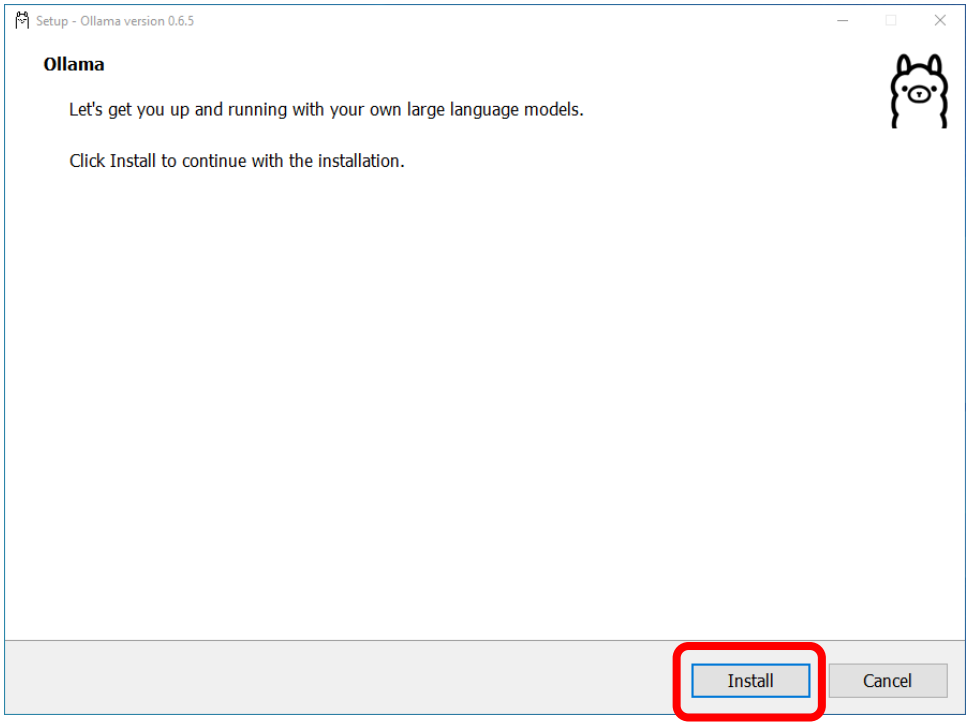
Installing Ollama

Windows

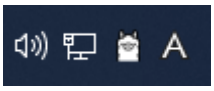
Before getting started, we need to install the Ollama tool locally, which allows us to run any open-source AI model on our computer.
If you haven't installed Ollama yet, please visit https://ollama.com/download to download and install it.



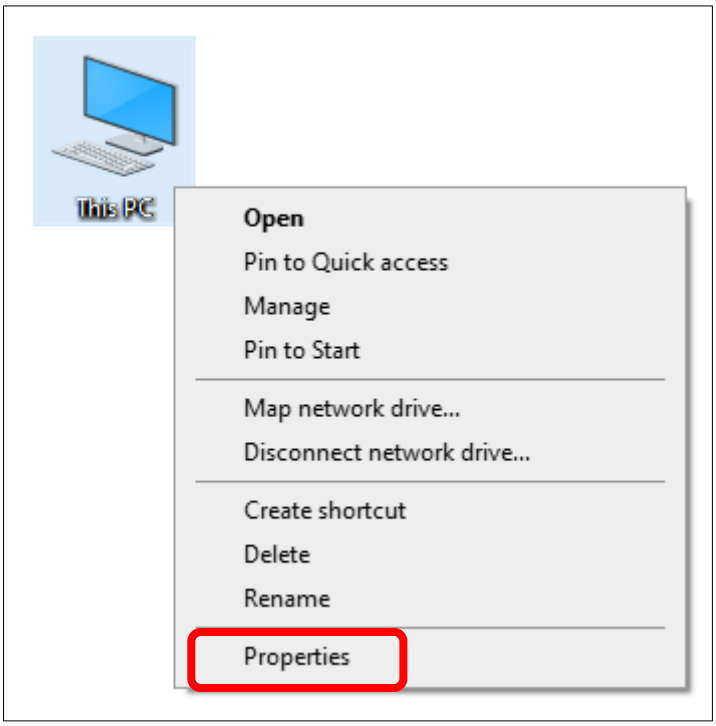
Run the Ollama Installing Package, click Install.



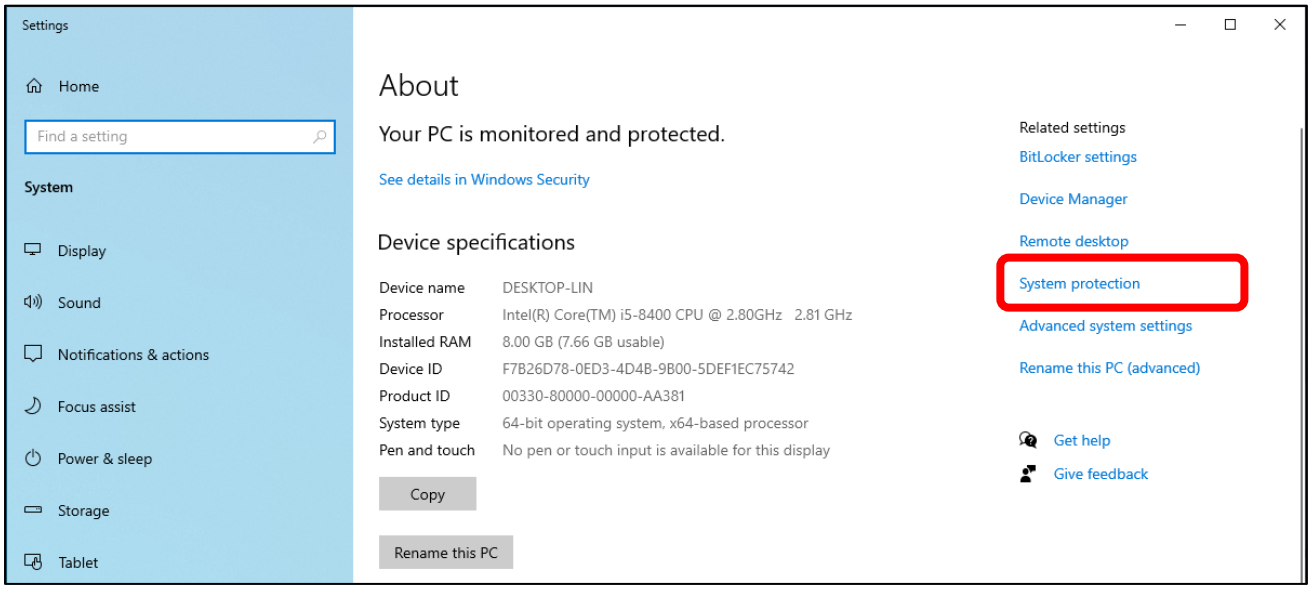
After installation, you'll see the Ollama icon appear in your taskbar.



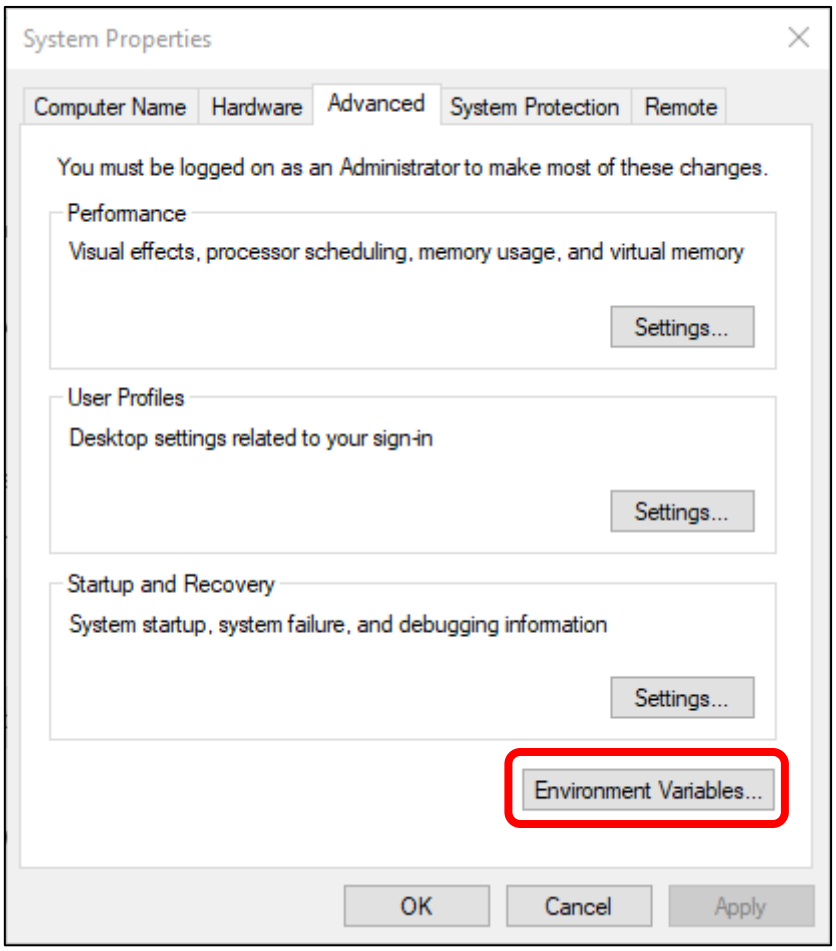
Return to your desktop, right-click on "This PC" (or "My Computer"), and select "Properties" from the context menu.



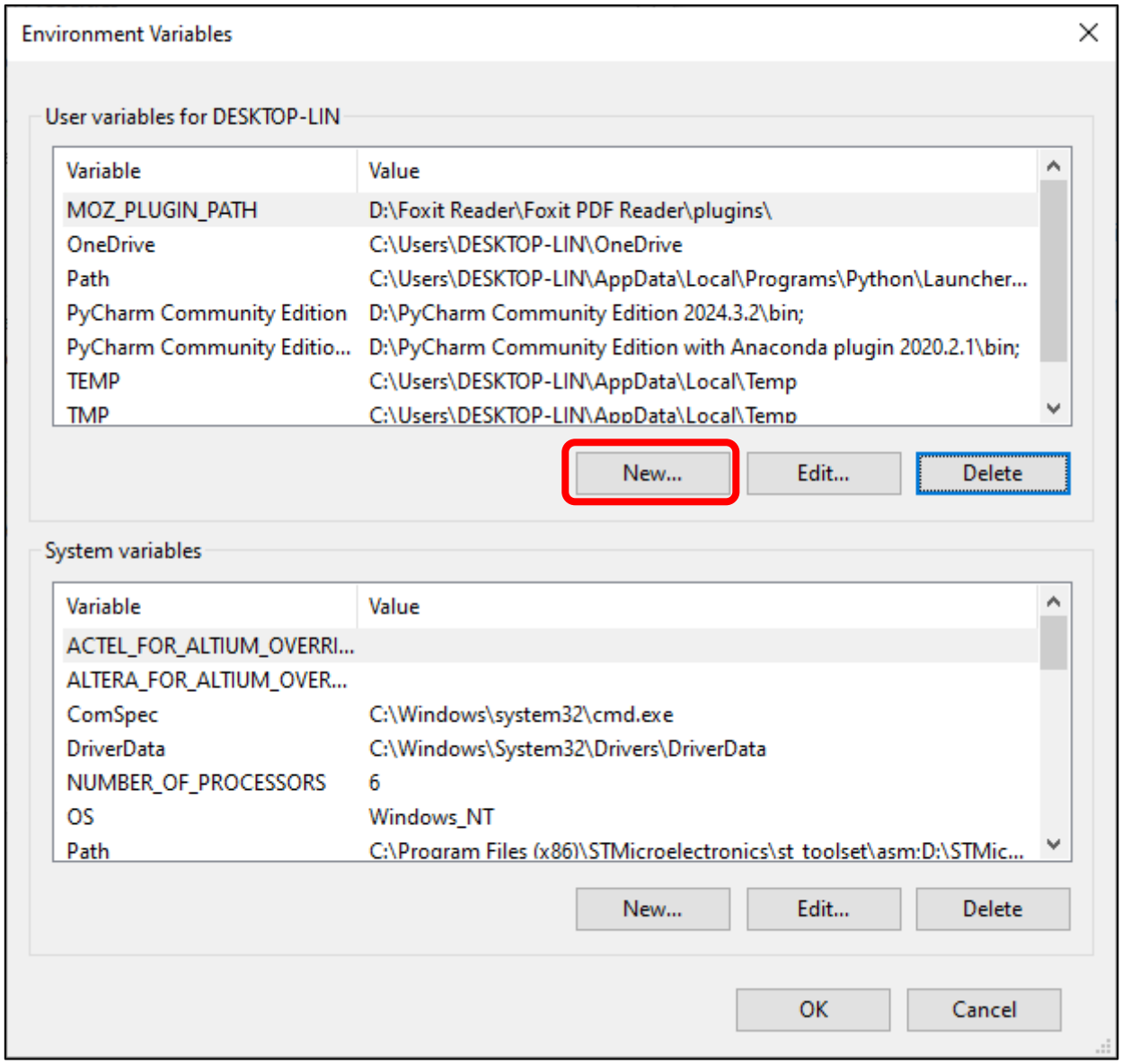
In the newly opened window, locate and click on "Advanced system settings".



In the newly opened window, click "Environment Variables".



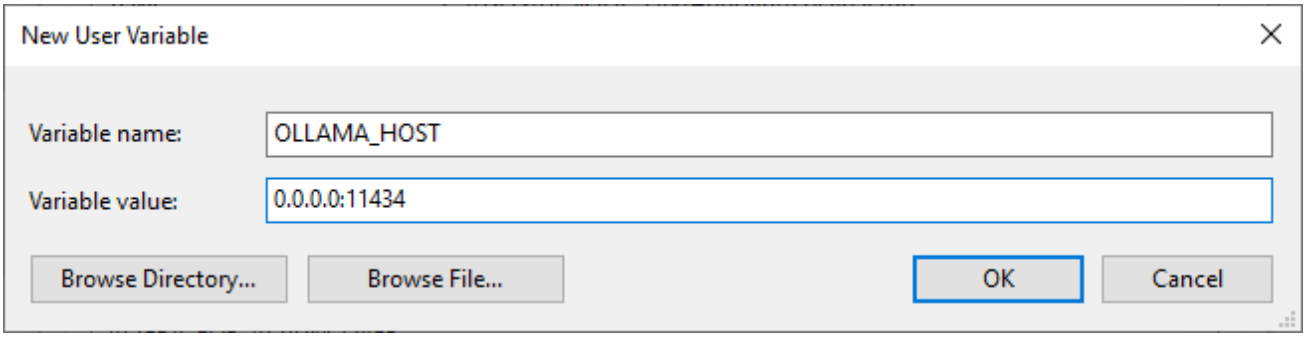
Click “New”.



In the Variable name field, enter: OLLAMA_HOST; In the Variable value field, enter: 0.0.0.0:11434; Click OK to save.

By doing so, all devices on your local network to access Ollama via your computer's IP address.

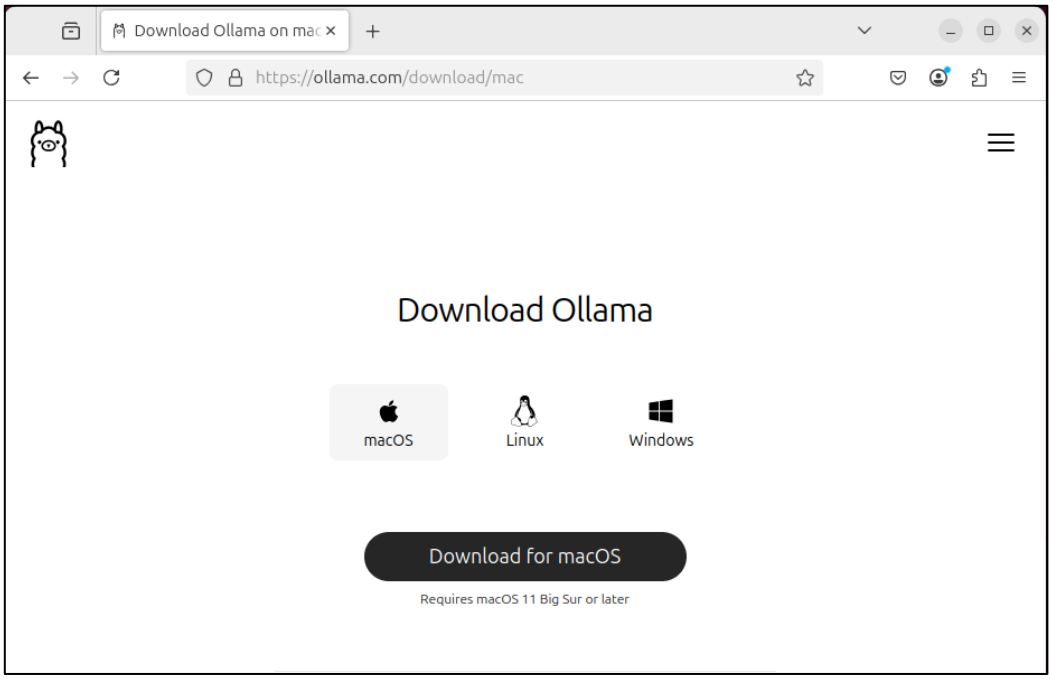
Without this setting, Ollama would only be accessible locally (from the host computer itself).



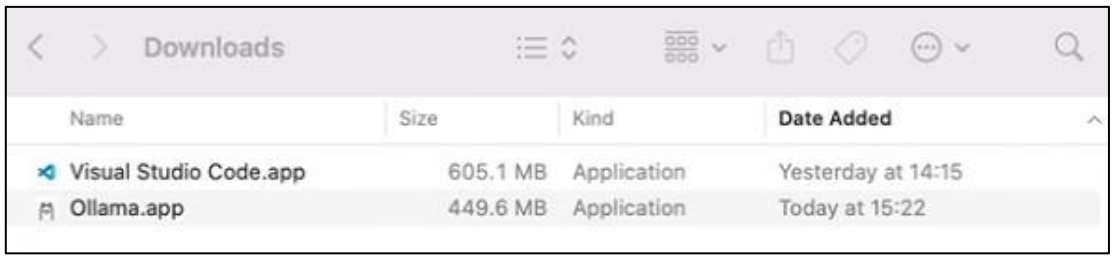
Mac OS

Before getting started, we need to install the Ollama tool locally, which allows us to run any open-source AI model on our computer.

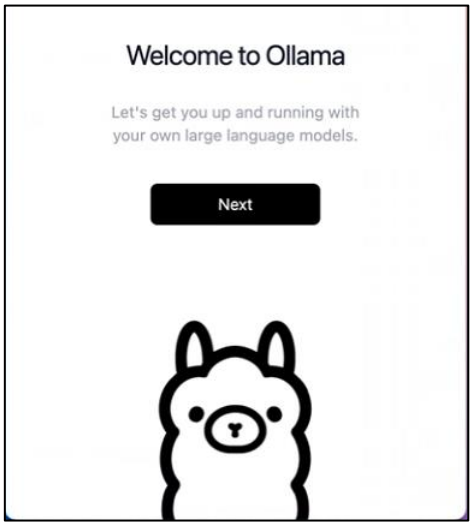
If you haven't installed Ollama yet, please visit https://ollama.com/download to download and install it.

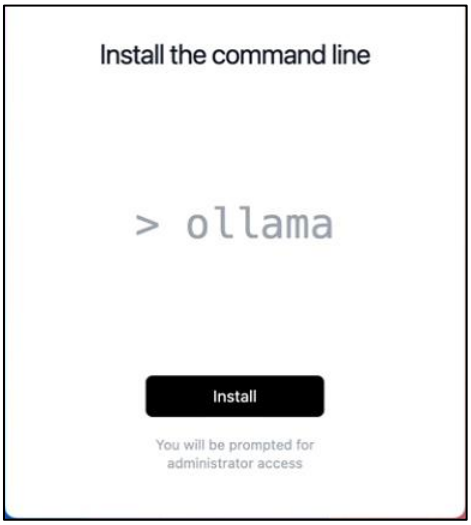


Find "Ollama.app" under "Downloads" and double click to open it.



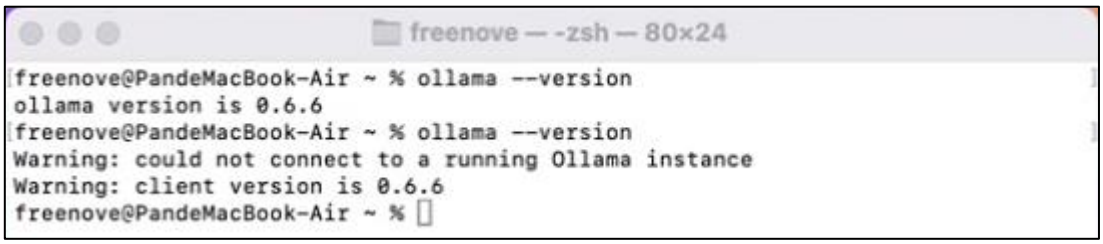
Click "Next" -> "Install" -> "Finish".





The interface will close automatically upon completion.
Open your terminal and check if Ollama is properly installed using the specified command.

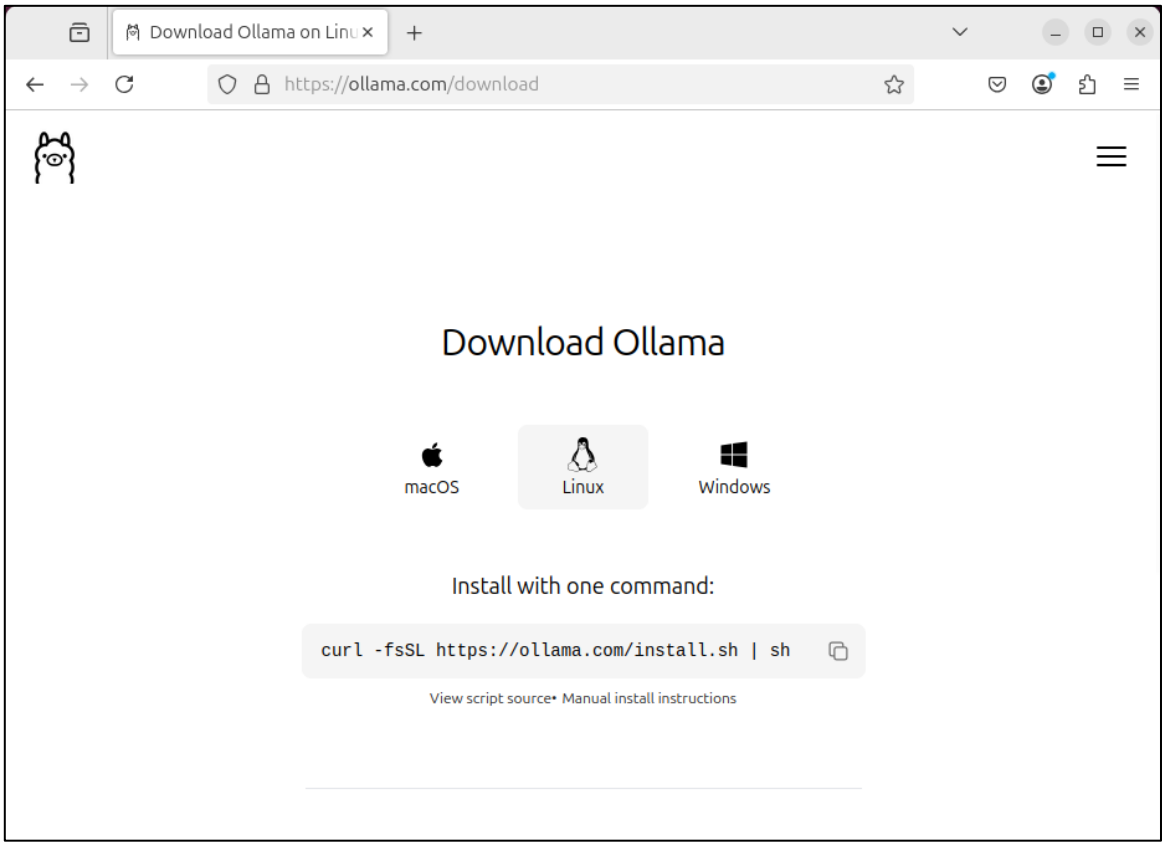
```
ollama --version
```



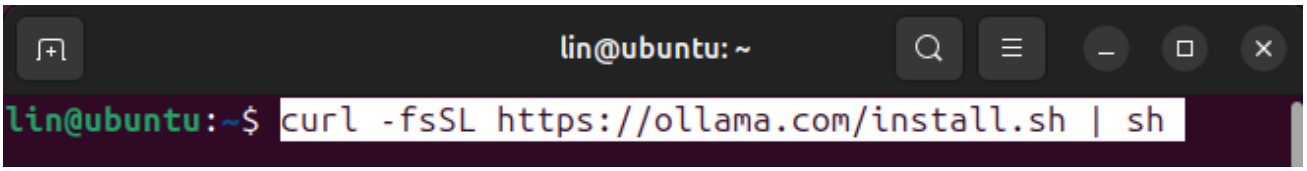
Note: Ollama is located in your Applications folder.
As shown in the image above:
If Ollama is already running, executing ollama --version will display its version number.
If Ollama isn't running, the same command will return a connection error ("could not to connect to a running Ollama instance")

Linux

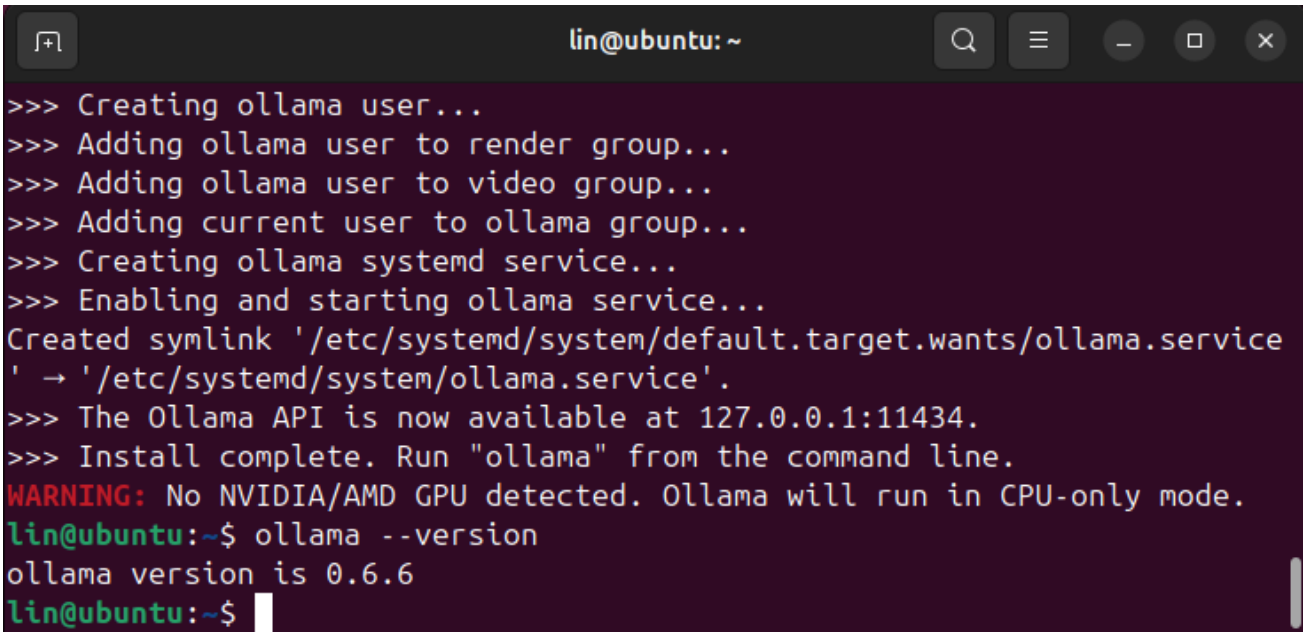
Before getting started, we need to install the Ollama tool locally, which allows us to run any open-source AI model on our computer.
If you haven't installed Ollama yet, please visit https://ollama.com/download to download and install it.



Open the Terminal and run the following command to install Ollama.

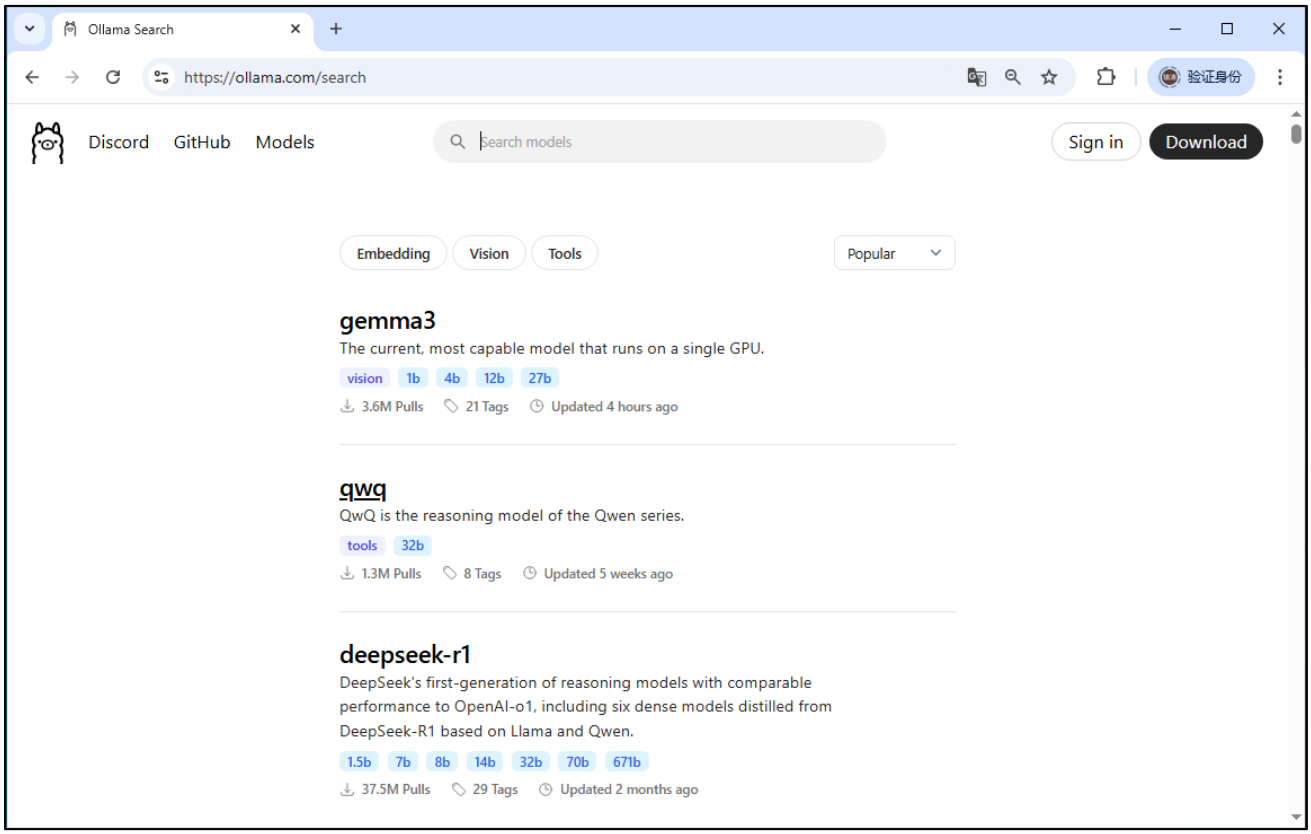


The installation should appear as shown in the image above. You can verify Ollama's installation by running the command: **ollama --version**

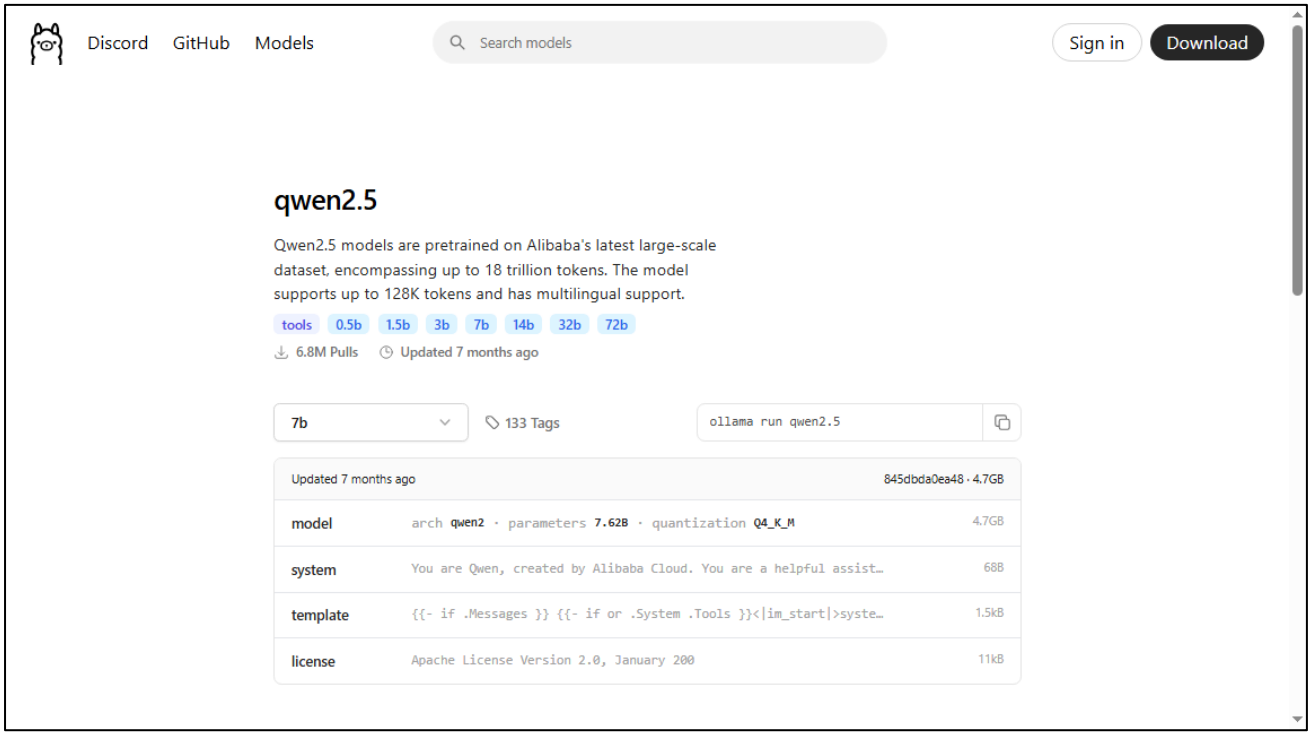


LLM Model

Please visit https://ollama.com/search and select the LLM model that suits your computer or your favorite.



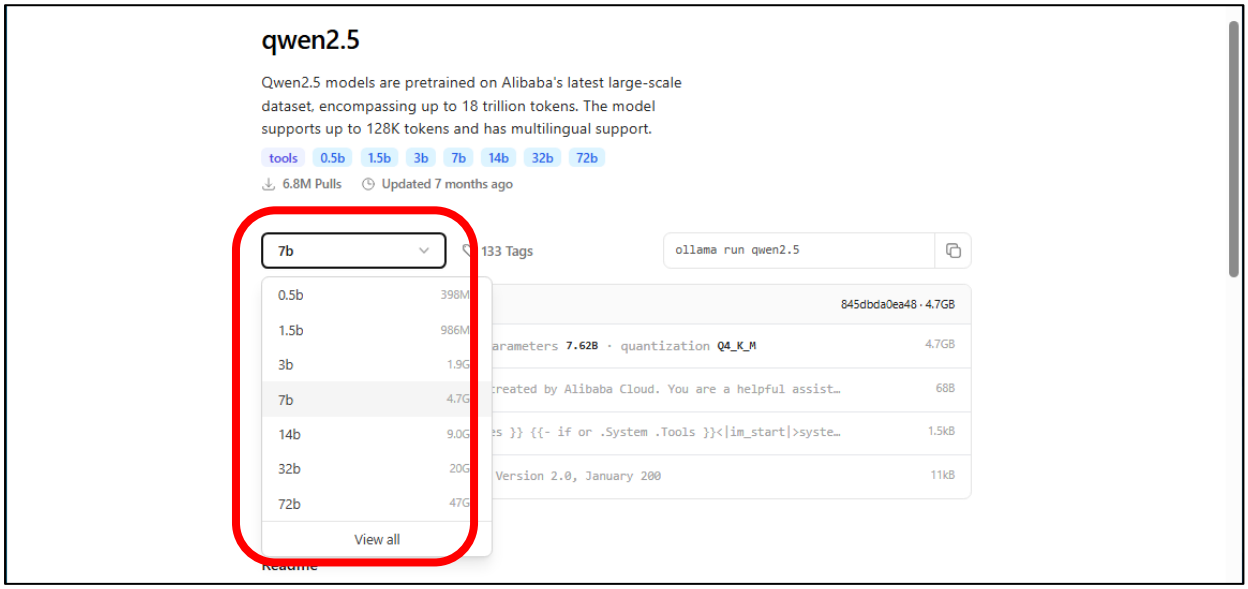
Here we take qwen2.5 as an example. Click “qwen2.5” model.



Please note that when selecting a model, you need to choose the appropriate model based on your computer's GPU memory or CPU RAM configuration.

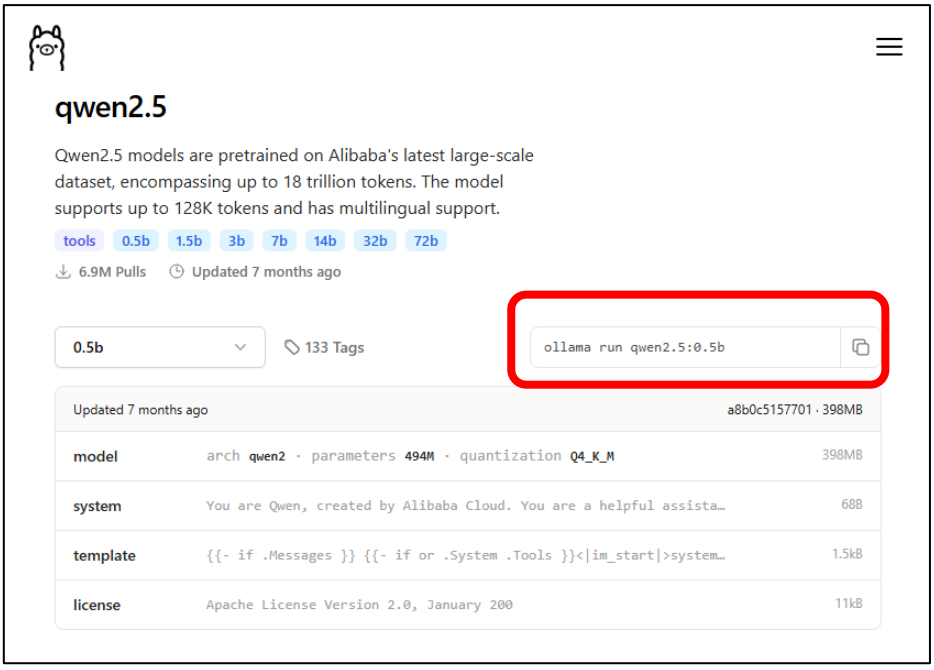
1. Larger models offer higher intelligence, while smaller models provide lower intelligence.
2. For high-end systems (strong GPU/CPU with ample memory), choose larger models for optimal performance; for low-end systems (limited GPU/CPU memory), opt for smaller models to ensure smooth operation.
3. Selecting an oversized model on a weak system may cause failure to load or extremely slow inference speeds.

You can select the appropriate model parameters via the dropdown menu.



Smaller models are less capable but faster. For this demonstration, we'll use qwen2.5:0.5b as an example. Copy the command from the webpage:

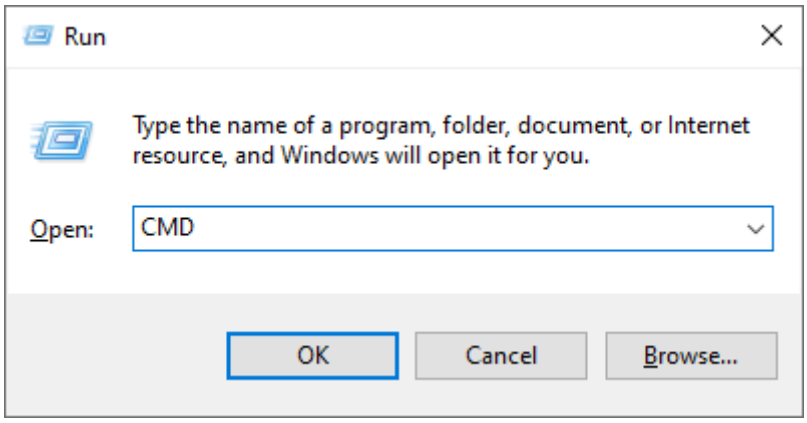
ollama run qwen2.5:0.5b



Next, install your preferred LLM model by selecting the version compatible with your operating system.

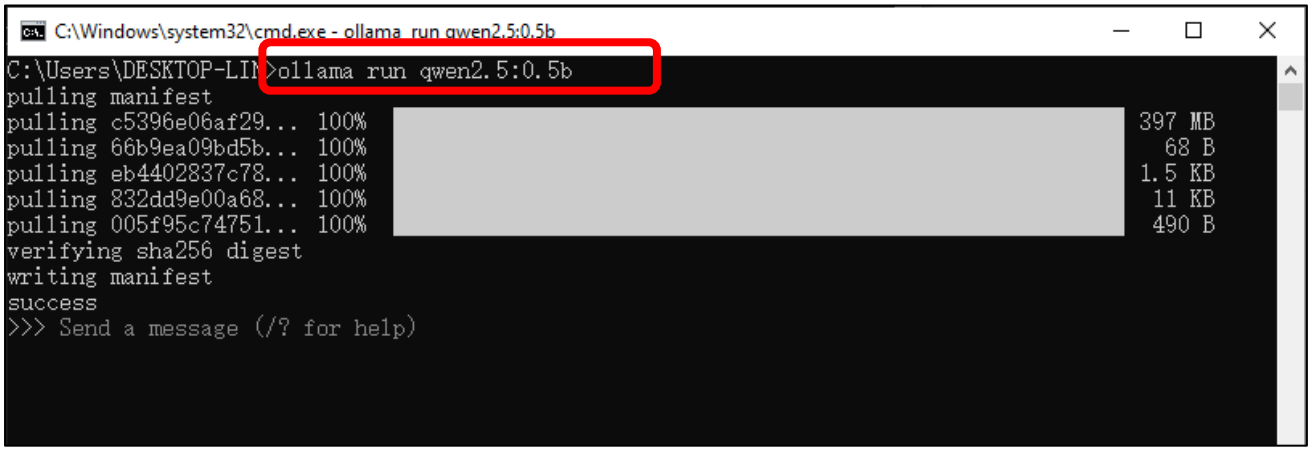
Windows

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

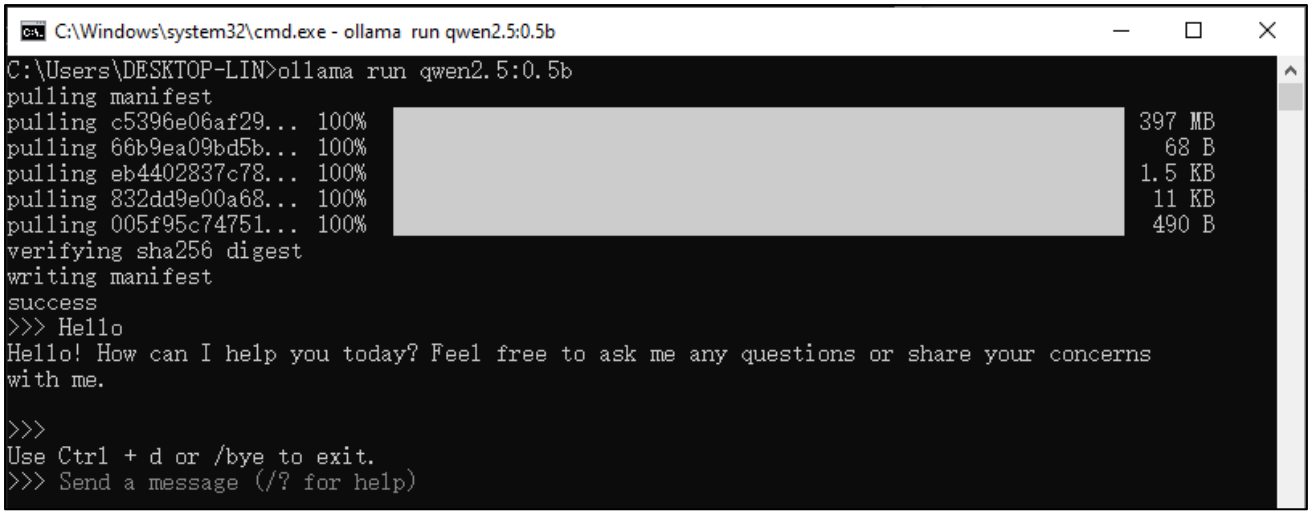


Run the command "**ollama --version**" to see if ollama has been installed.

Enter"**ollama run qwen2.5:0.5b**" to download the model to the local.machine.

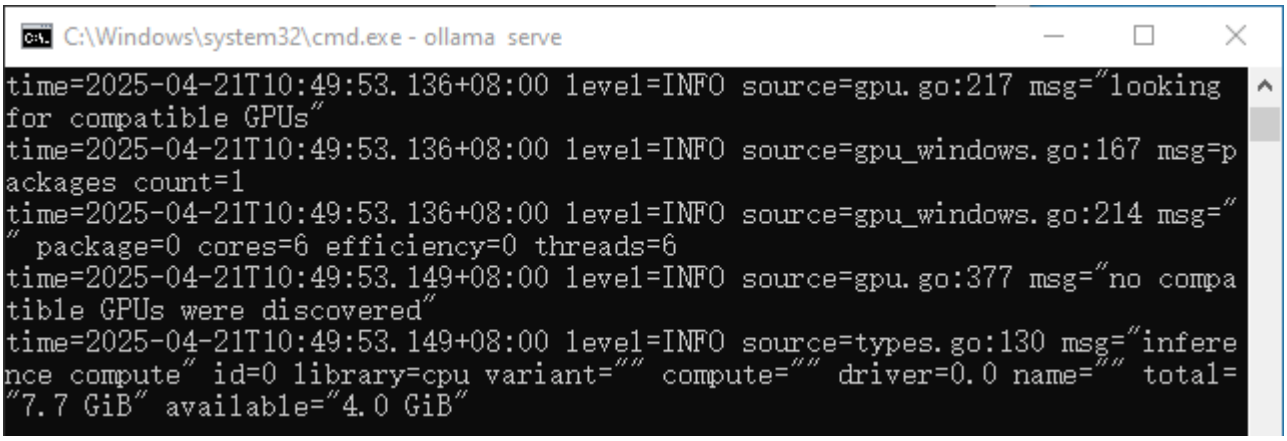


After installation is complete, you can directly chat with Qwen2.5-0.5B in the CMD interface.

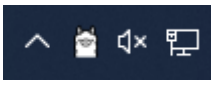


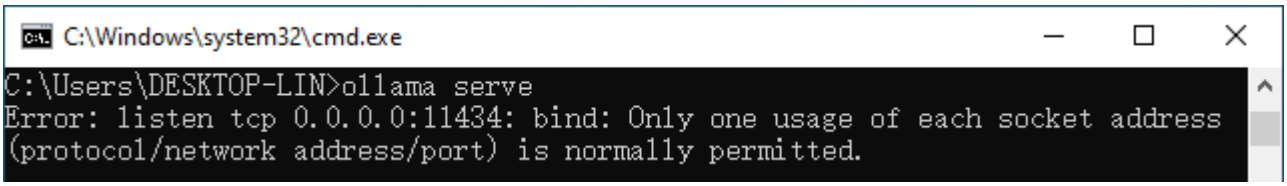
You can press Ctrl+D to exit chat mode.

You can start the Ollama server by running the command "**ollama serve**"



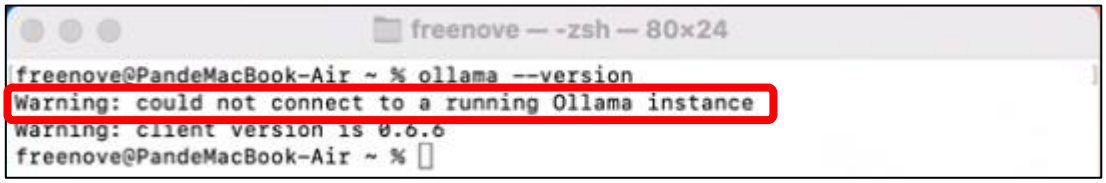
Note: If Ollama is already running (indicated by its icon in the system taskbar), executing ollama serve will cause an error. Both methods launch the same service.





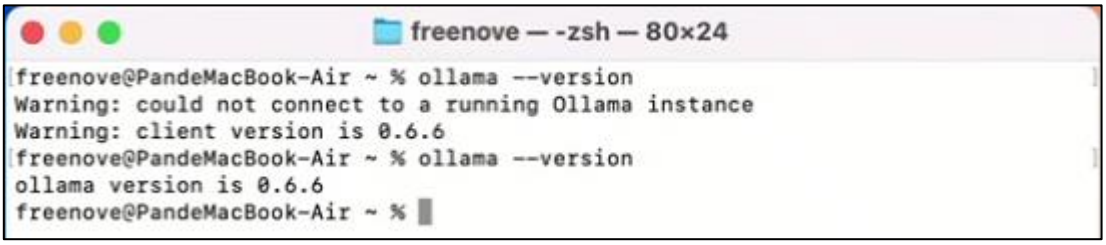
Mac OS

Run the command "**ollama --version**" on the Terminal to check whether Ollama has been installed.

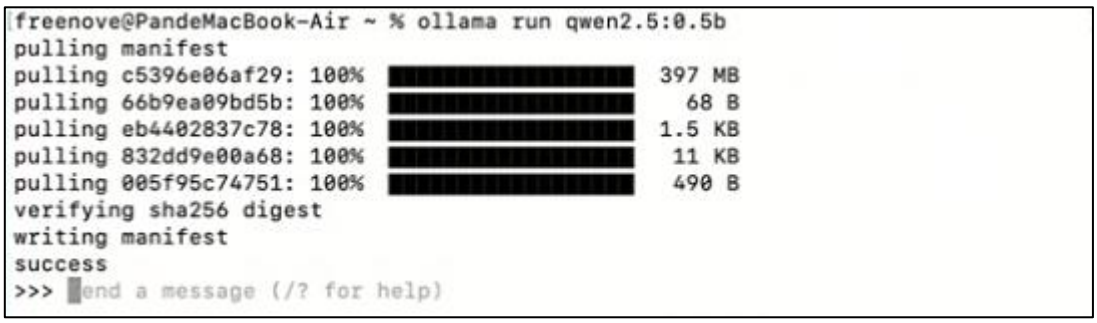


If you see the prompt "Warning: could not connect to a running Ollama instance", it indicates that Ollama has not been run. Go to Applications to run it.

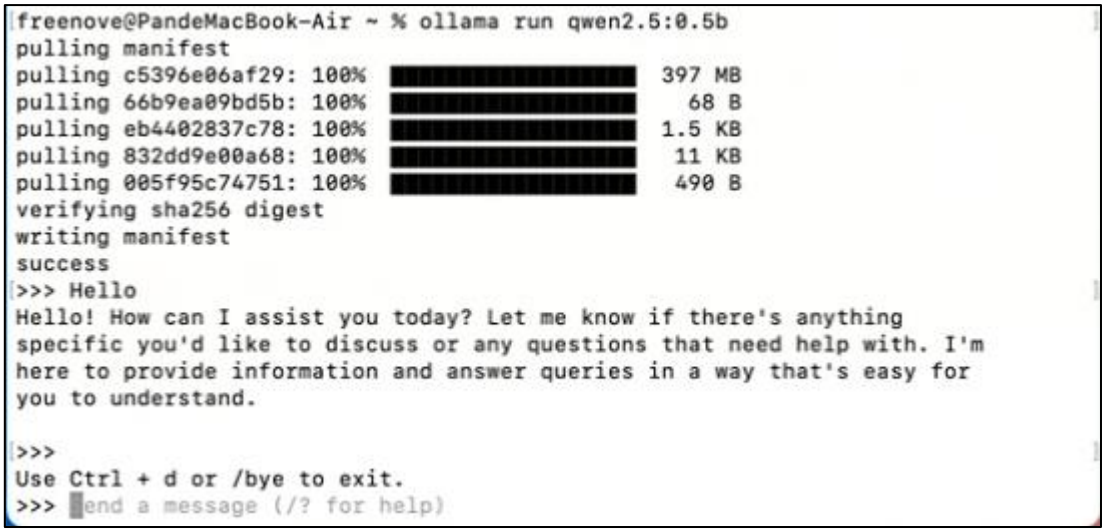
Check whether it is running again on the Terminal.



On the Terminal, run "**ollama run qwen2.5:0.5b**" to install the model to your computer.

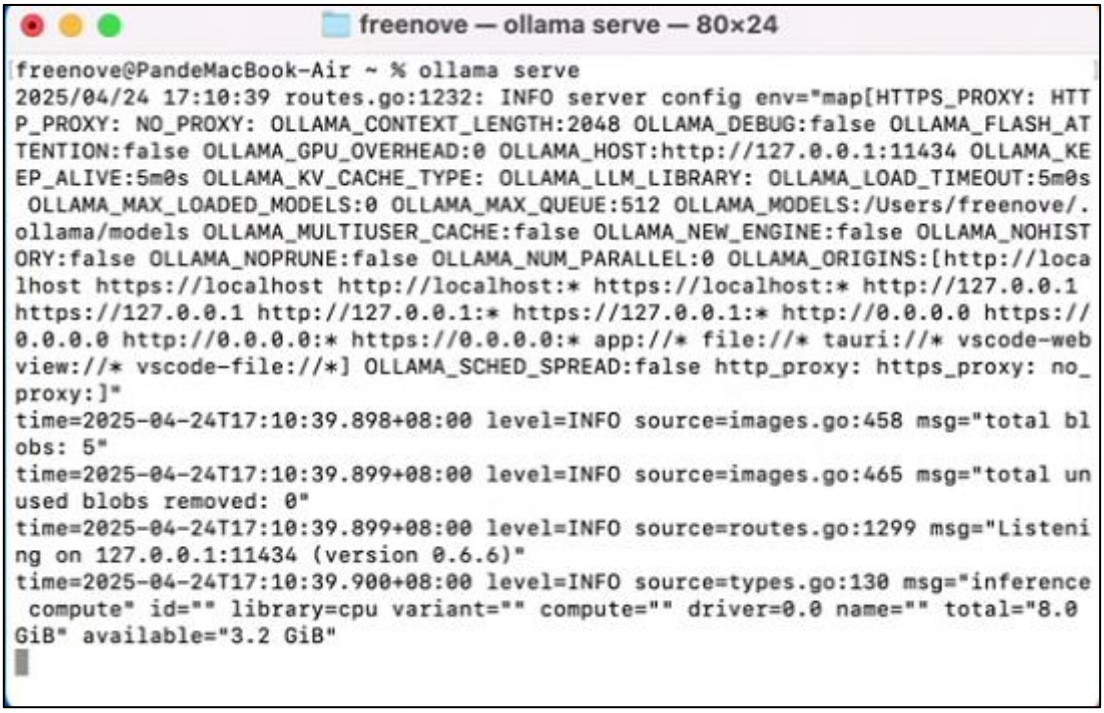


After installation is complete, you can directly chat with Qwen2.5-0.5B in the Terminal interface.



You may exit by pressing “Ctrl+D”.

You can start the Ollama Server by running the command “**ollama serve**”.

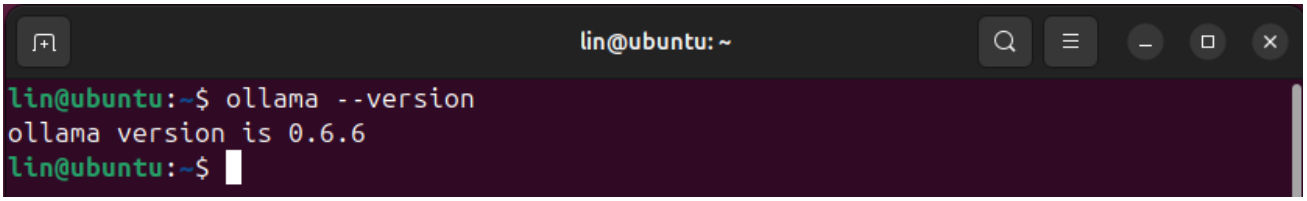


If Ollama has been running, you will see the following message.

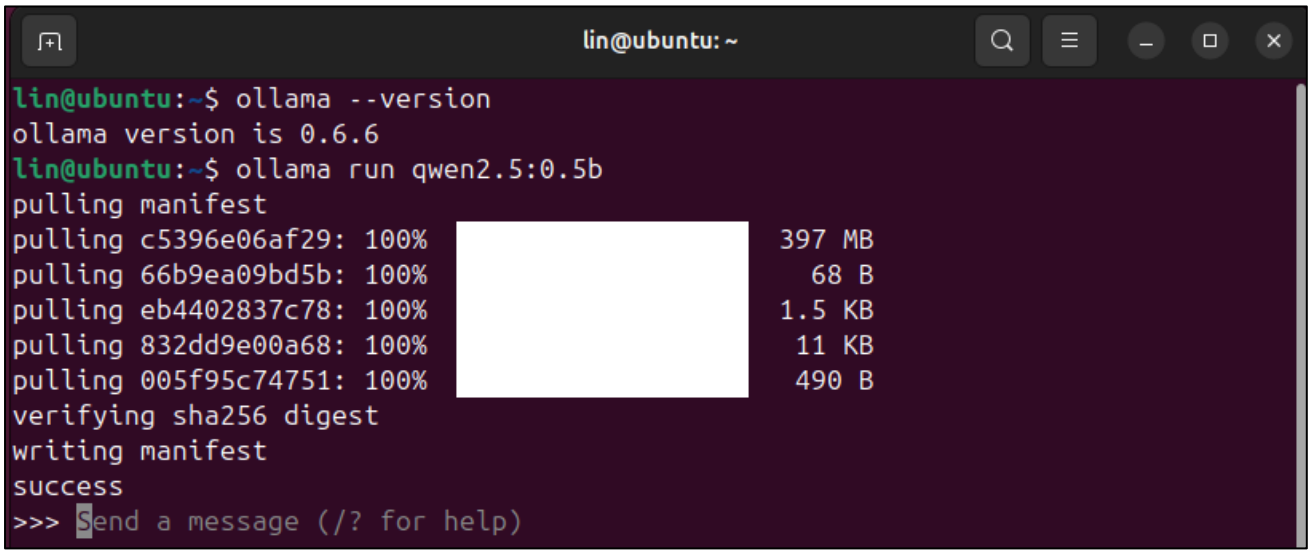
To access Ollama's user guide, run command Ollama.

Linux

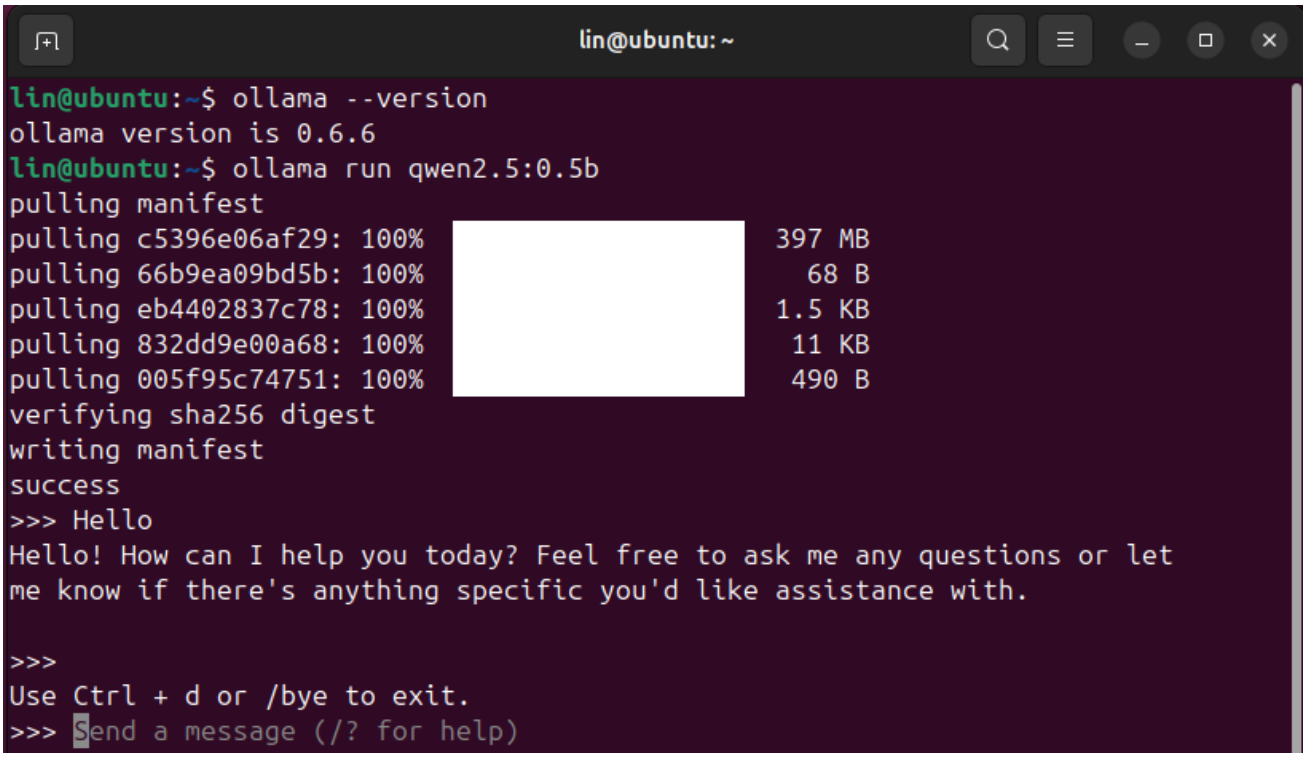
Run the command "**ollama --version**" on the Terminal to check whether Ollama has been installed.



Run the command"**ollama run qwen2.5:0.5b**" to download the model to your computer.

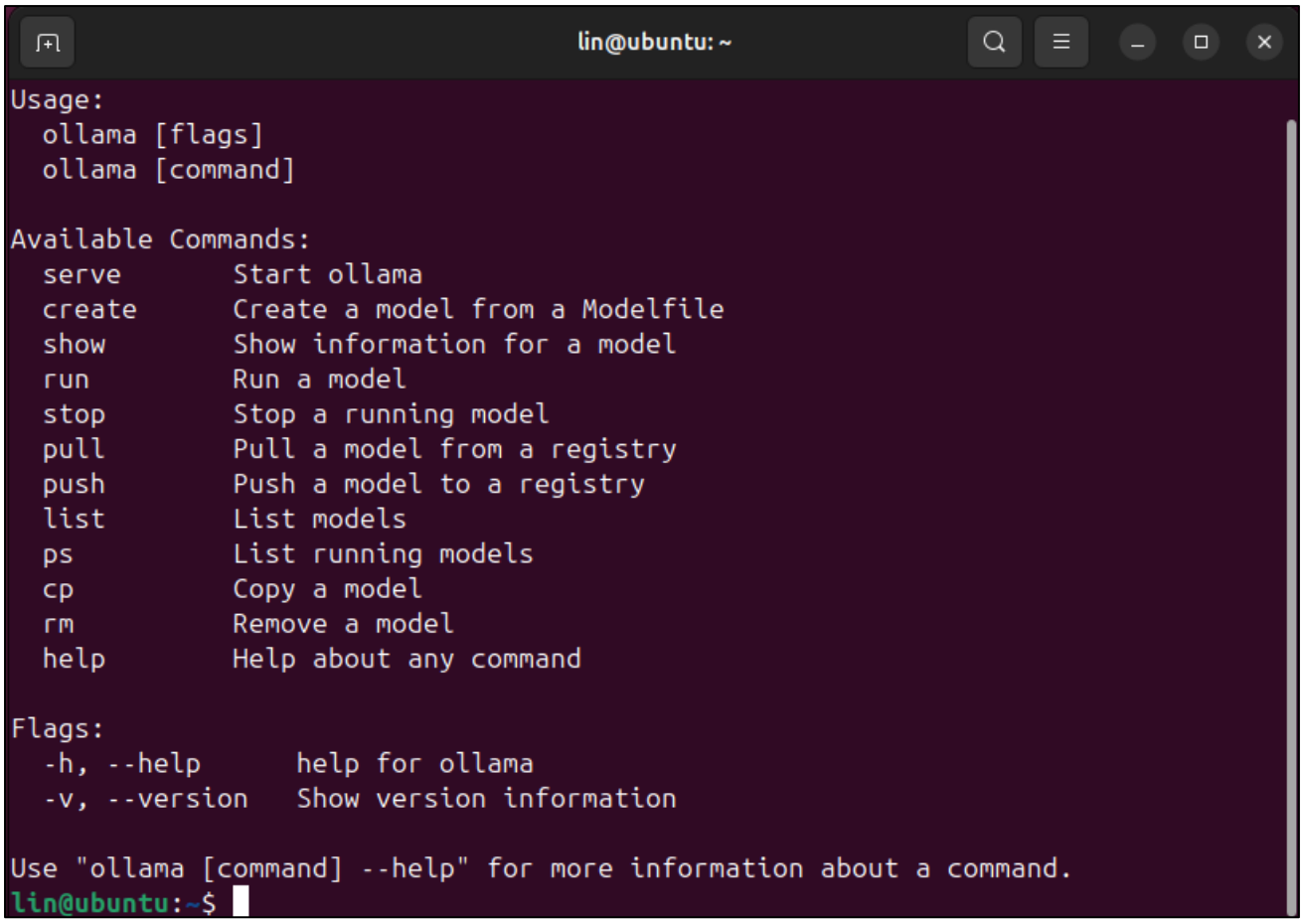


After installation is complete, you can directly chat with Qwen2.5-0.5B in the Terminal interface



To exit it, press "Ctrl+D".

To access Ollama's user guide, run command Ollama.



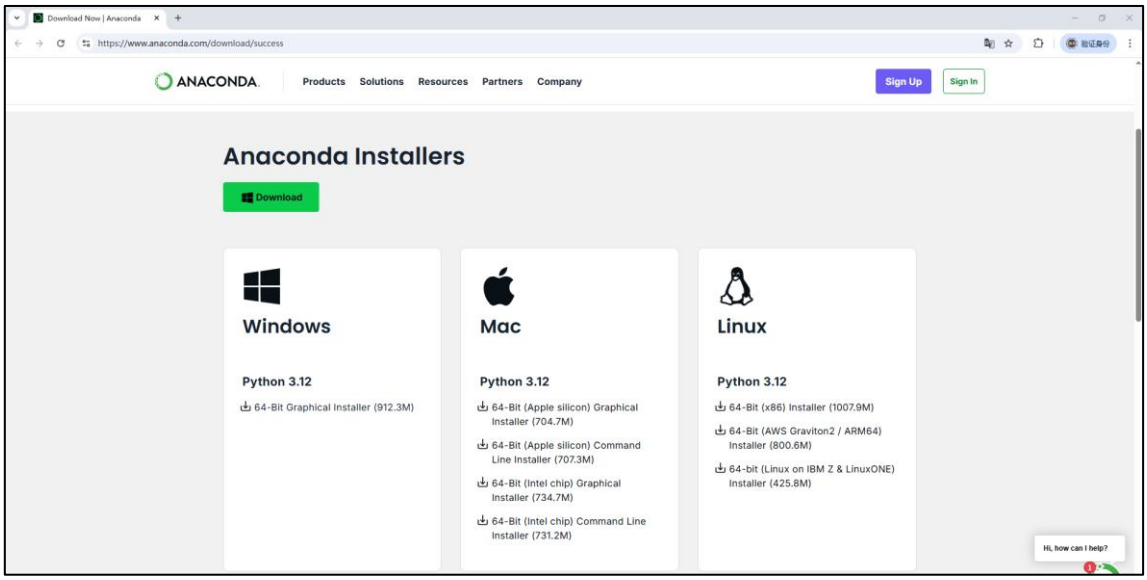
Installing Conda

The xiaozhi-esp32-server open-source project offers four installation methods. In this tutorial, we'll demonstrate the simplest configuration example. For other usage methods, please refer to the project's website for further exploration.

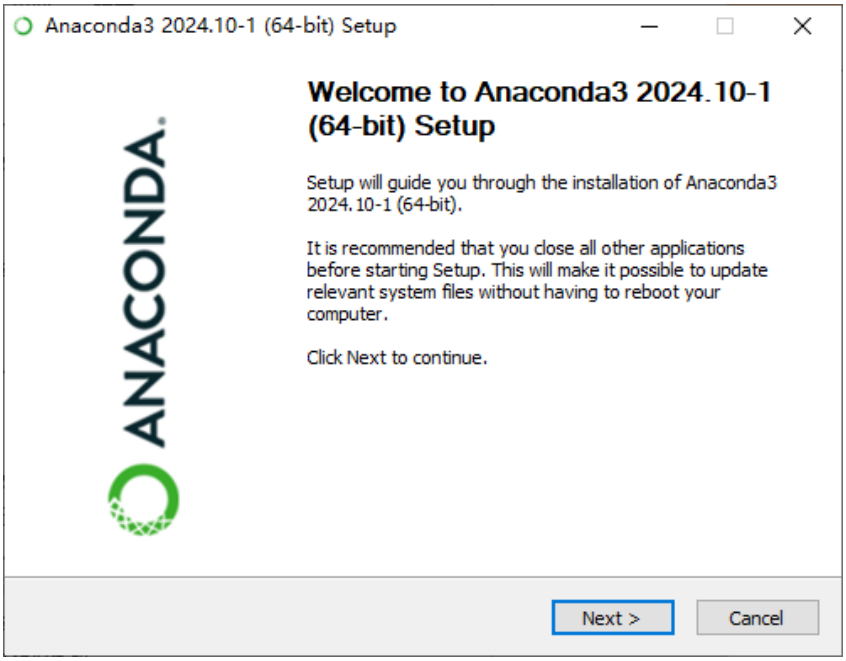
Windows

This example uses Conda for dependency management. Therefore, you'll need to have Conda installed on your system beforehand. If you haven't installed Conda yet, you can download and install it from: https://www.anaconda.com/download/success
Select the appropriate installer for your operating system.

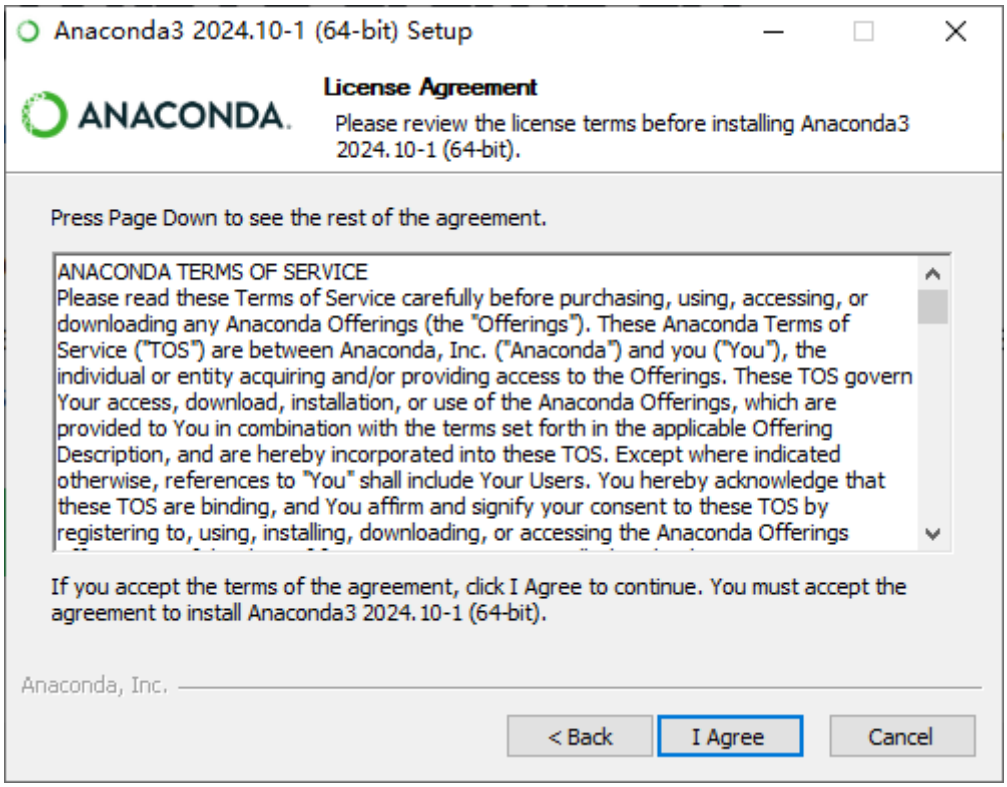
Miniconda is an installer by Anaconda that comes preconfigured for use with the Anaconda Repository.



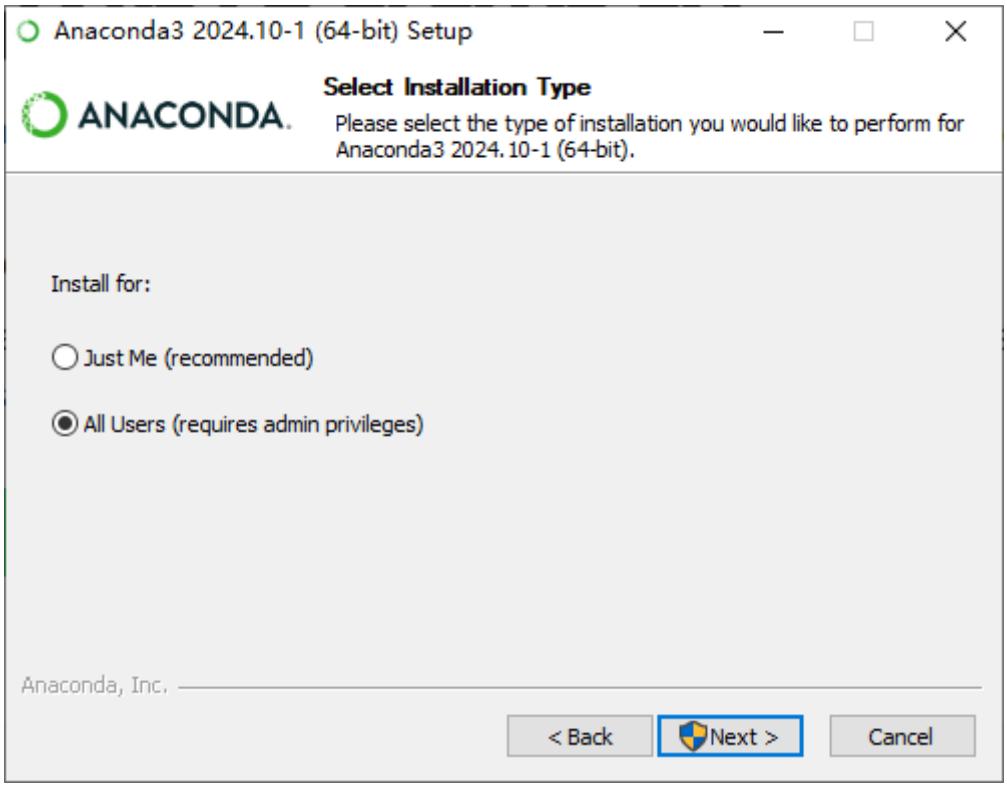
Here, we'll use Windows as an example. Double-click the downloaded Conda installer and click Next.



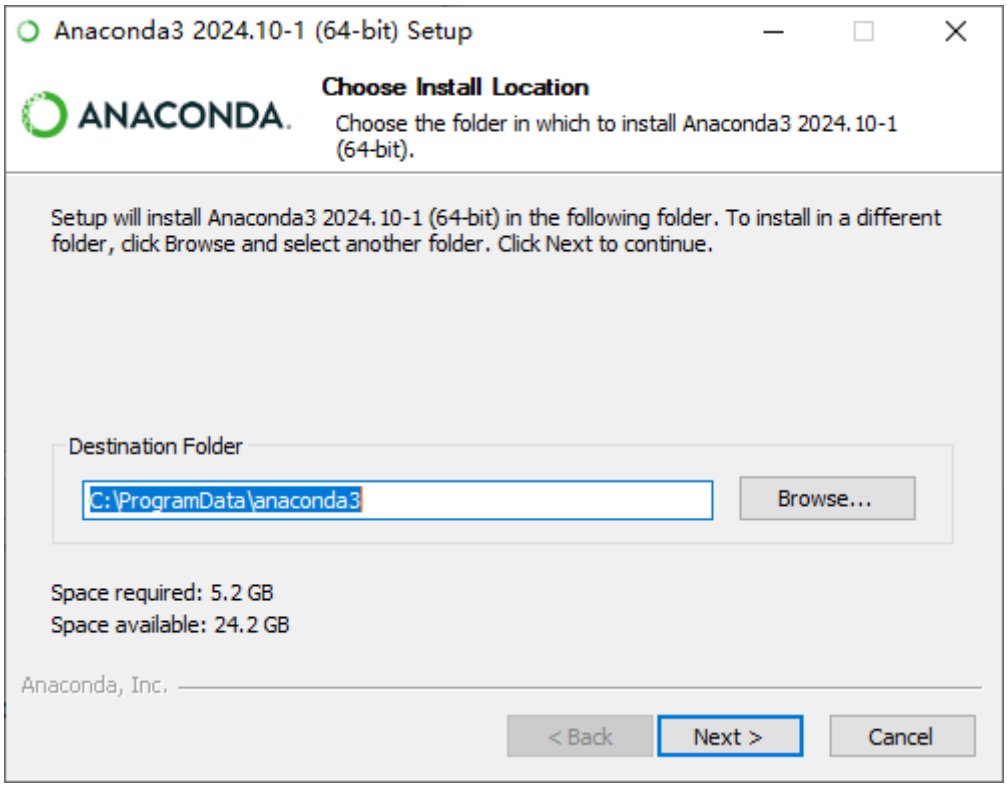
Click “I Agree”.



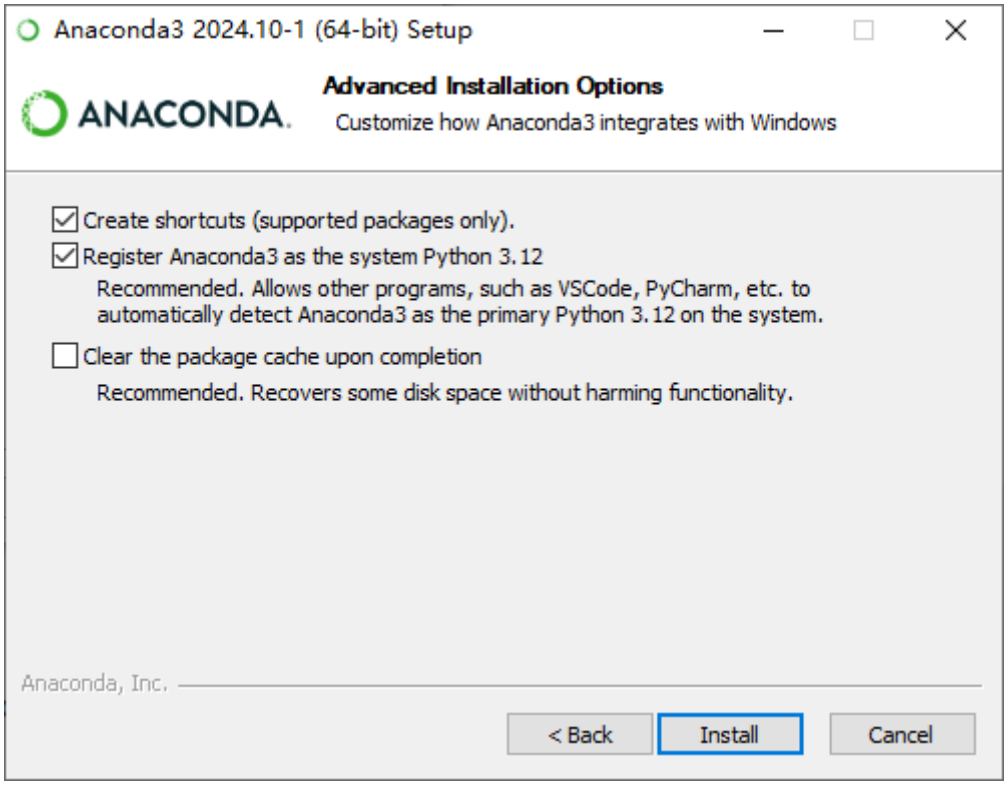
Select the installation type based on your needs. Typically, choose "All Users" for system-wide installation.



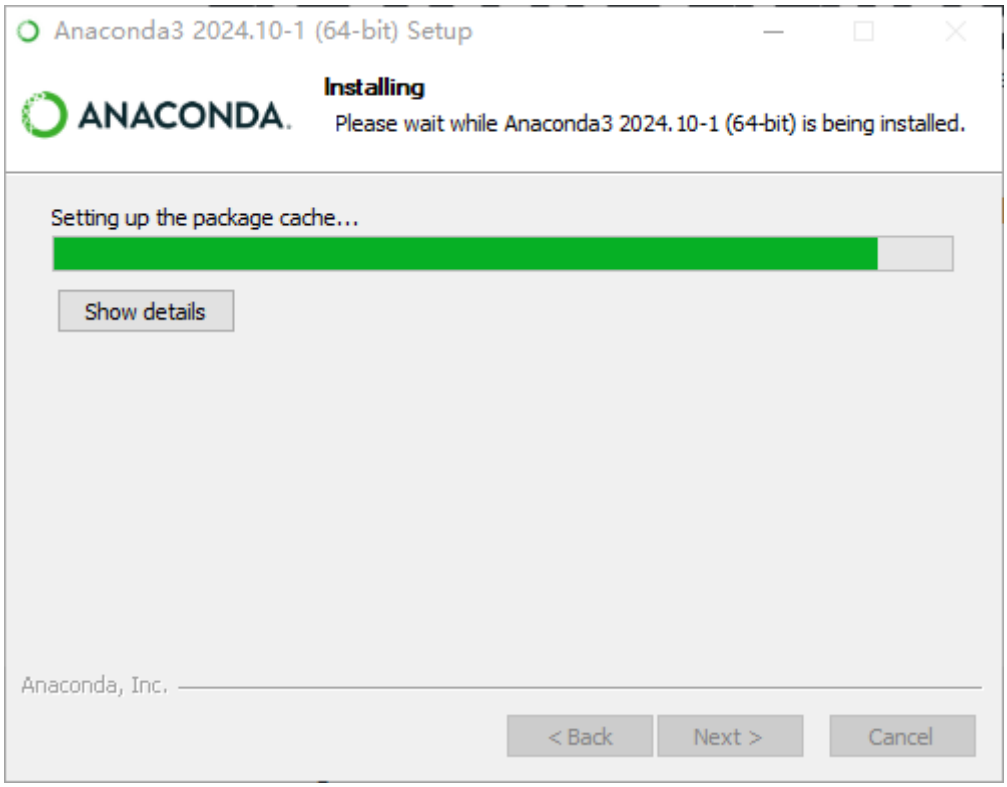
Specify the installation location for the software,



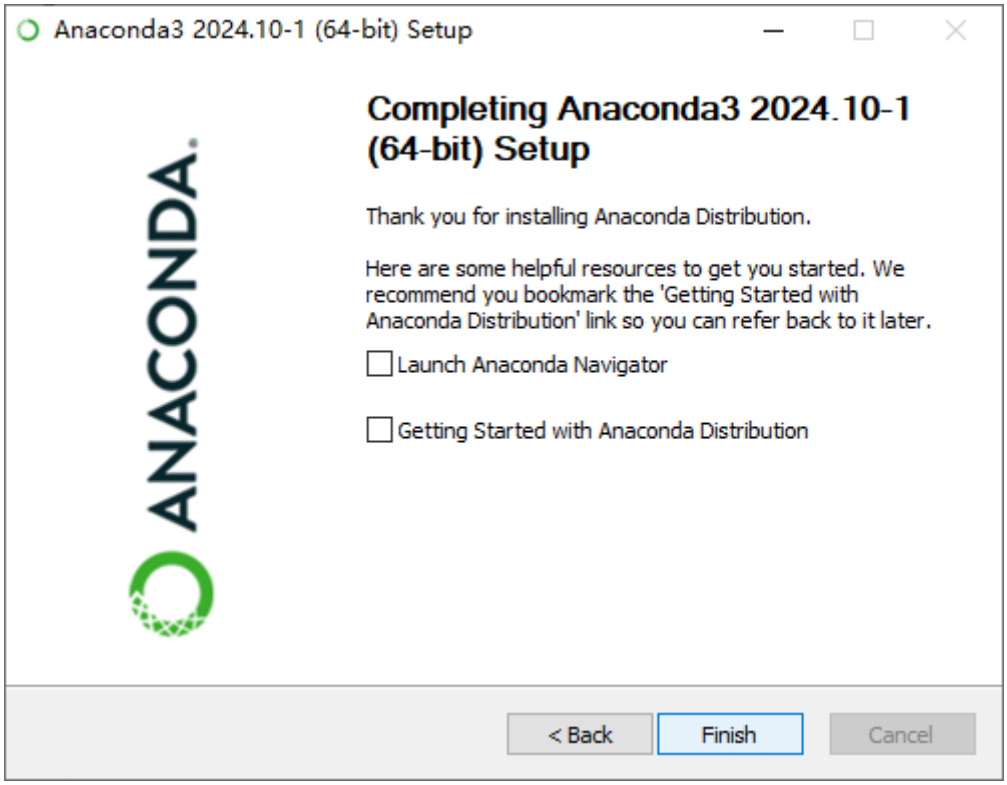
Keep the following configuration as default and click Install.



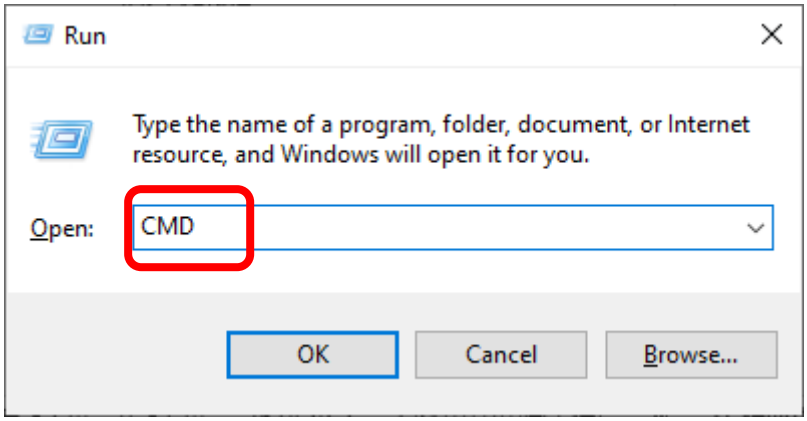
Wait for the installation to finish, which may take a while.



So far, the software has been installed.



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



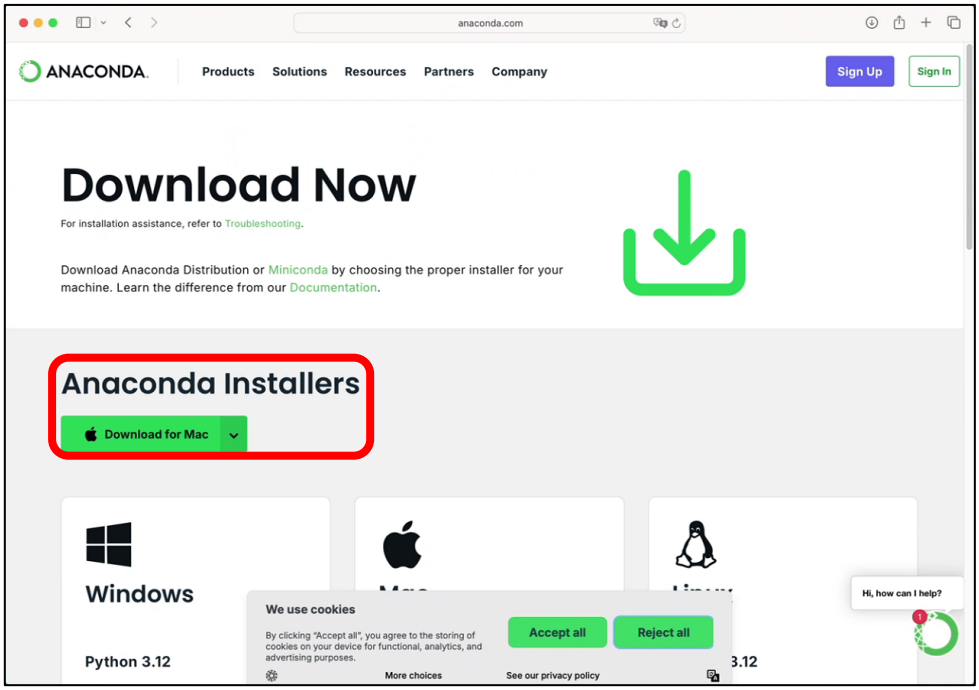
Enter **conda --version** and press Enter. If Anaconda3 is installed correctly, you should see version information like this:

Mac OS

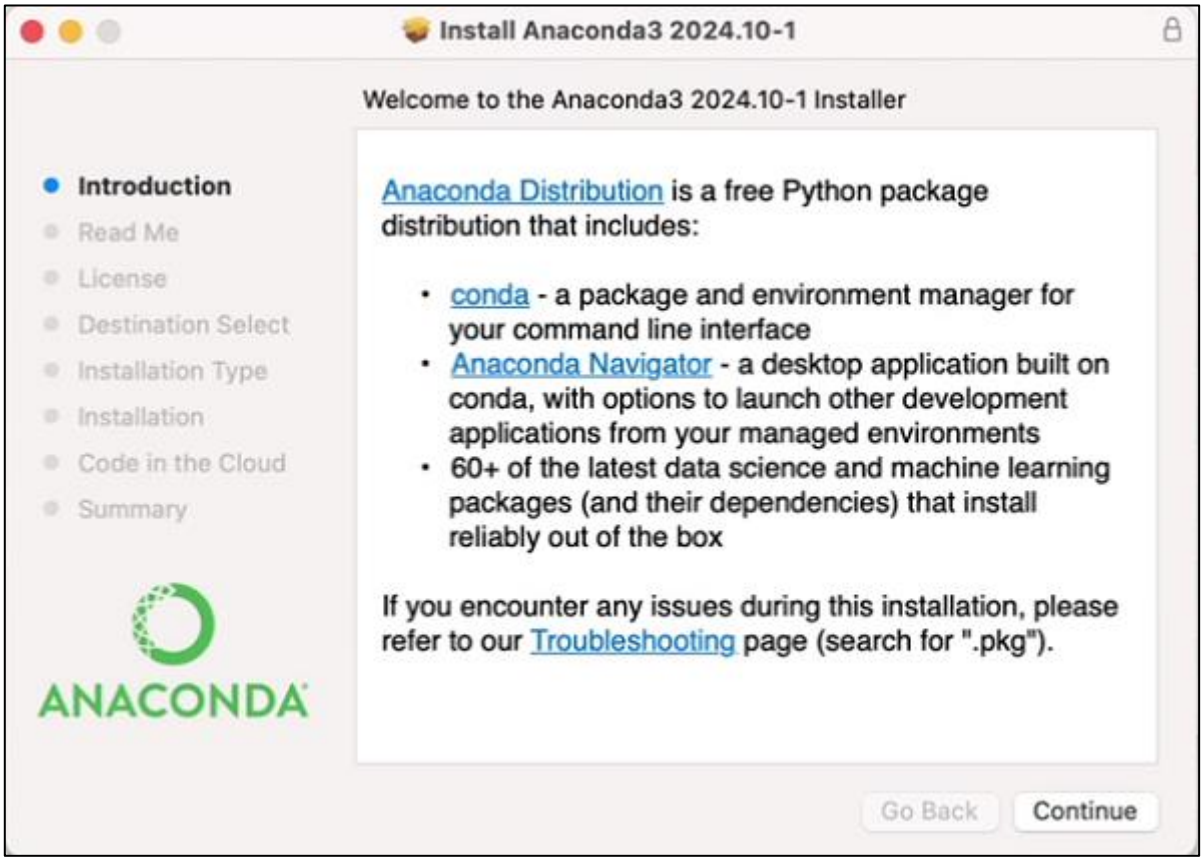
This example uses Conda for dependency management. Therefore, you'll need to have Conda installed on your system beforehand. If you haven't installed Conda yet, you can download and install it from: https://www.anaconda.com/download/success

Select the appropriate installer for your operating system.

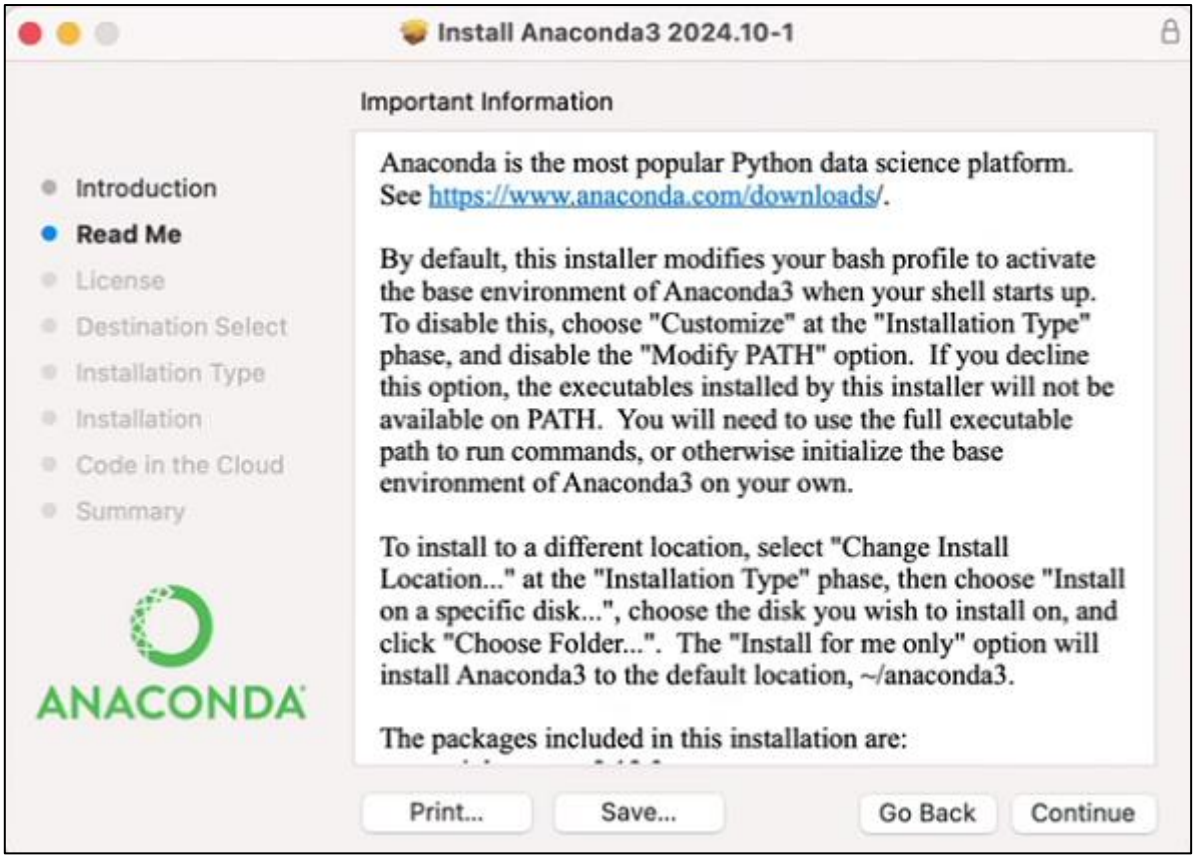
Miniconda is an installer by Anaconda that comes preconfigured for use with the Anaconda Repository.



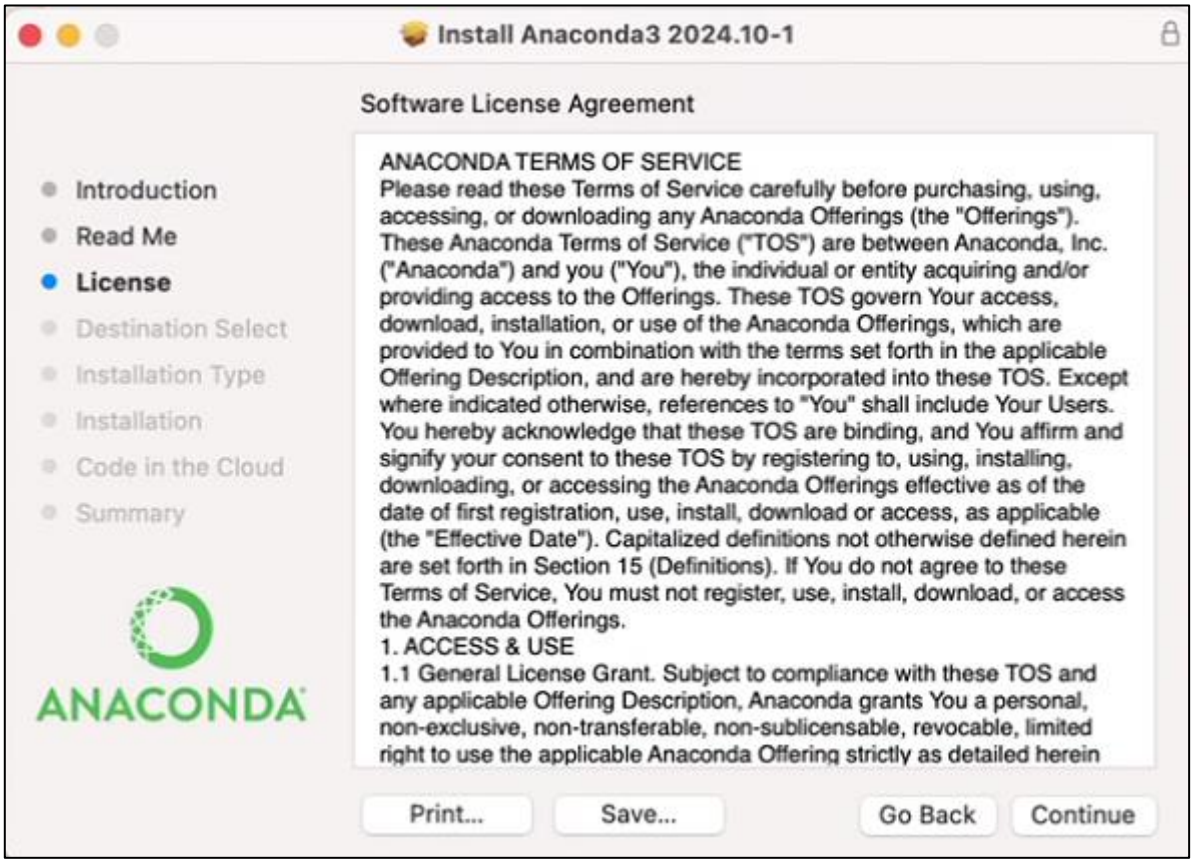
Double click to open the Conda application and click Continue.



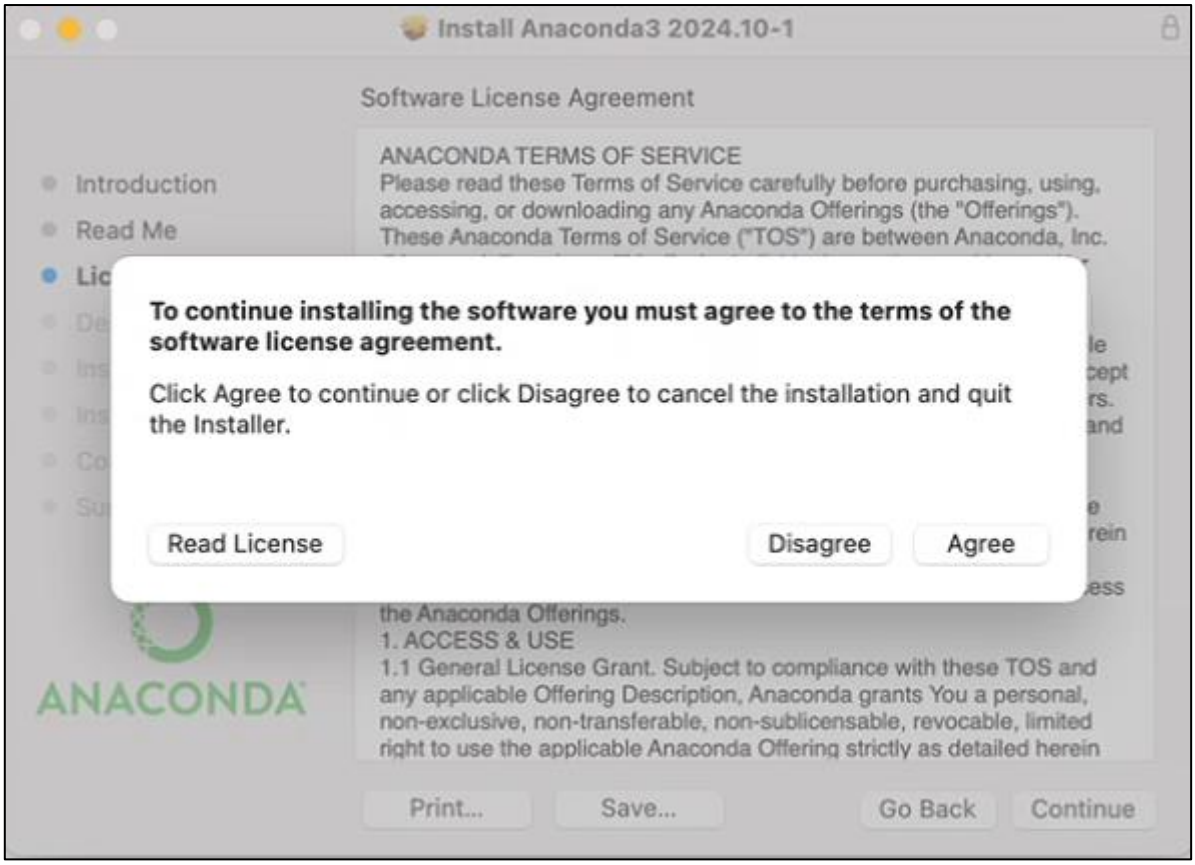
Click Continue.



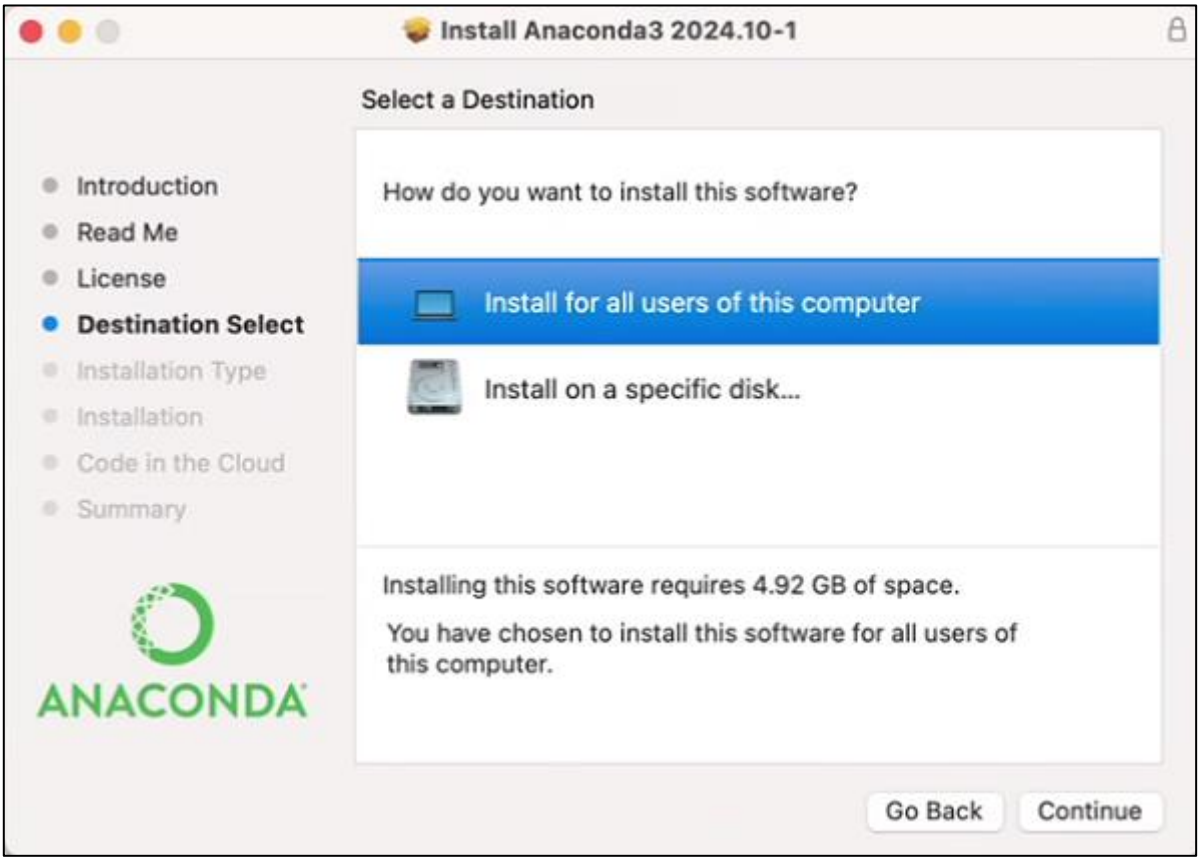
Click Continue.



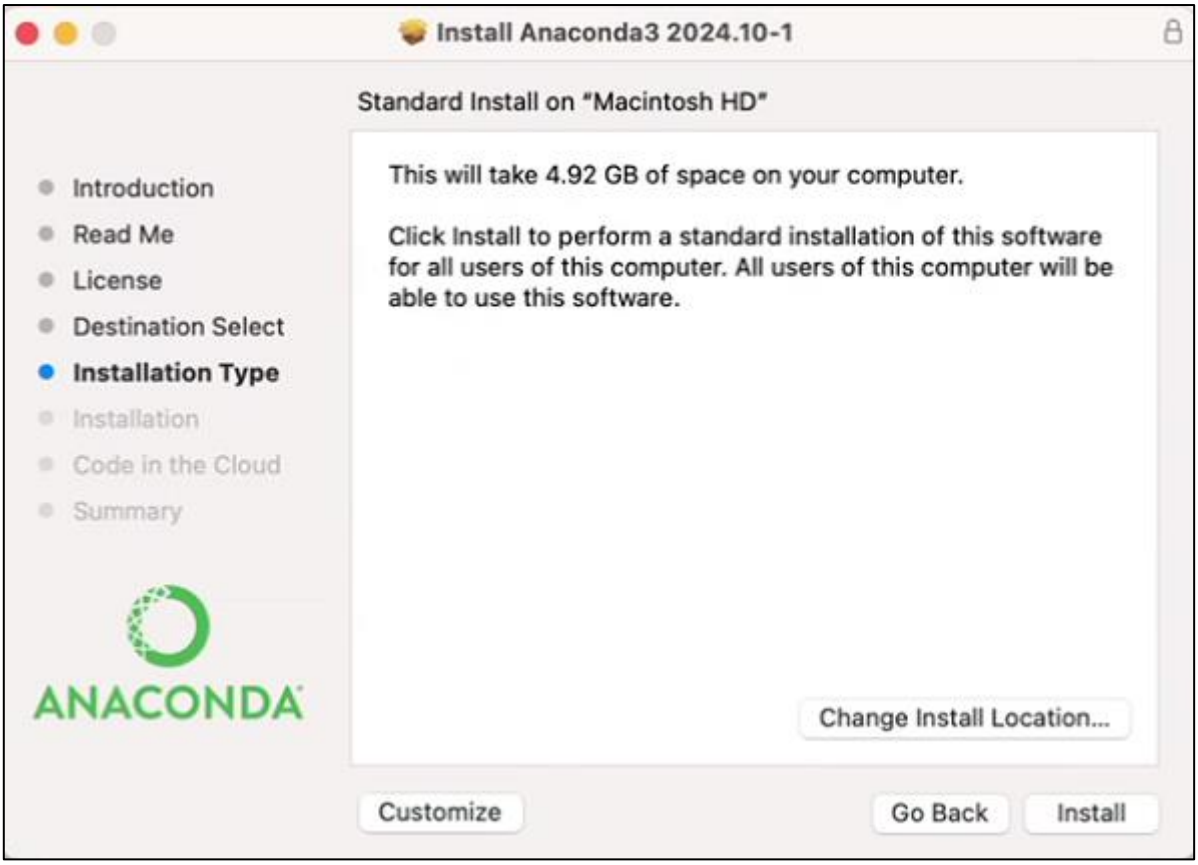
Click Agree.



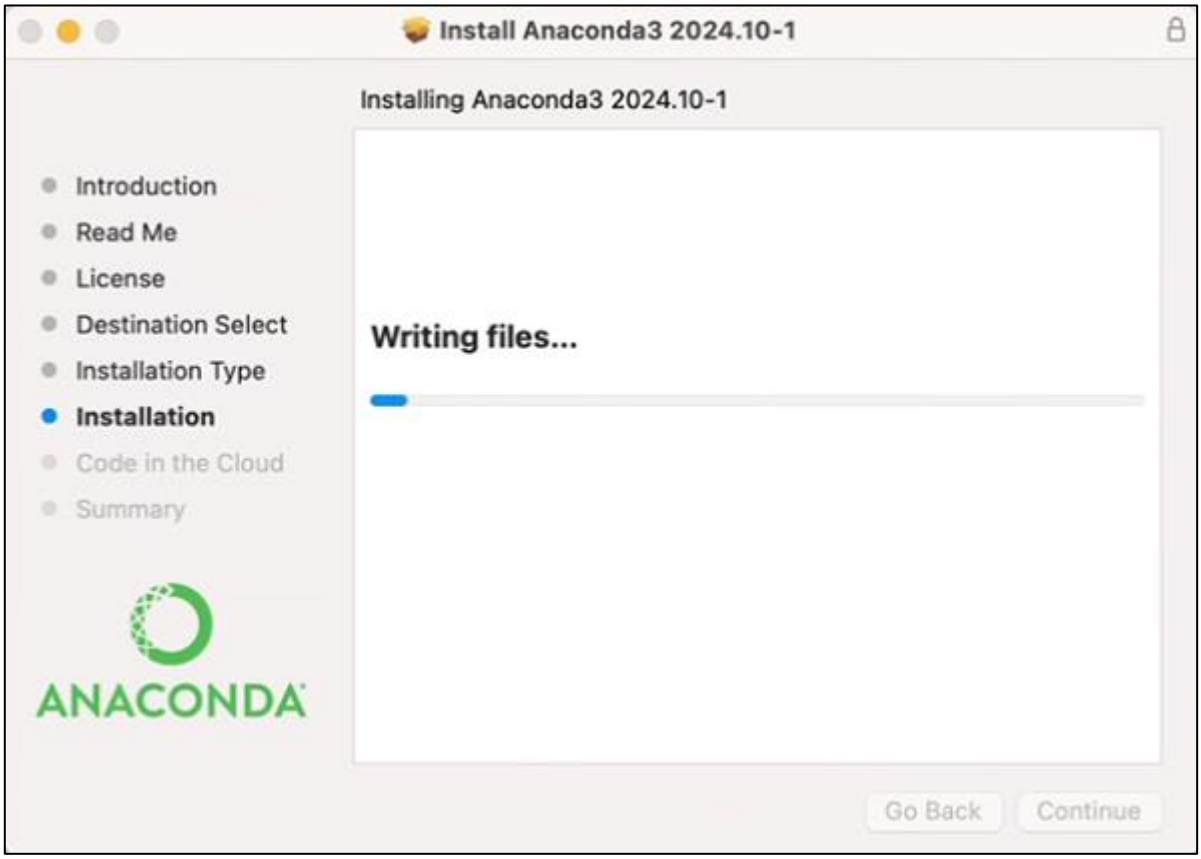
Click "Continue" to proceed with default settings.



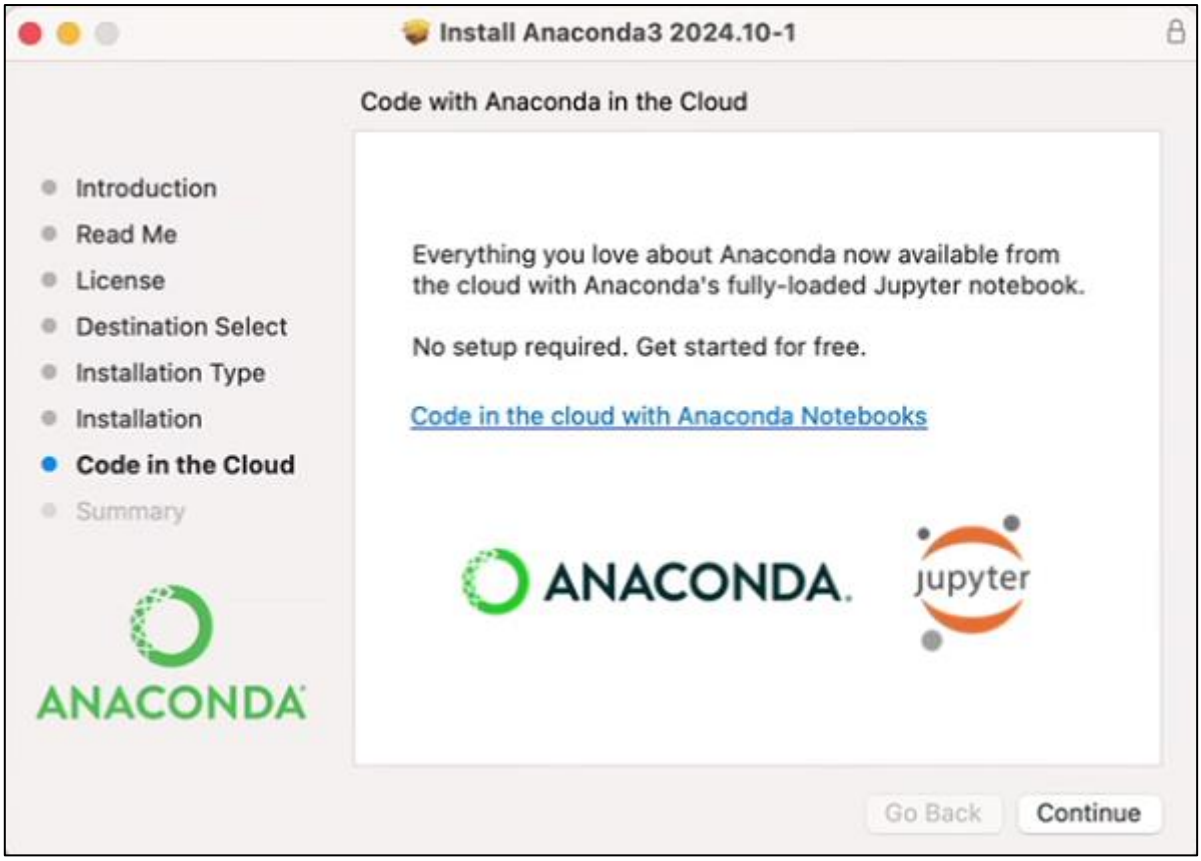
Click Install.



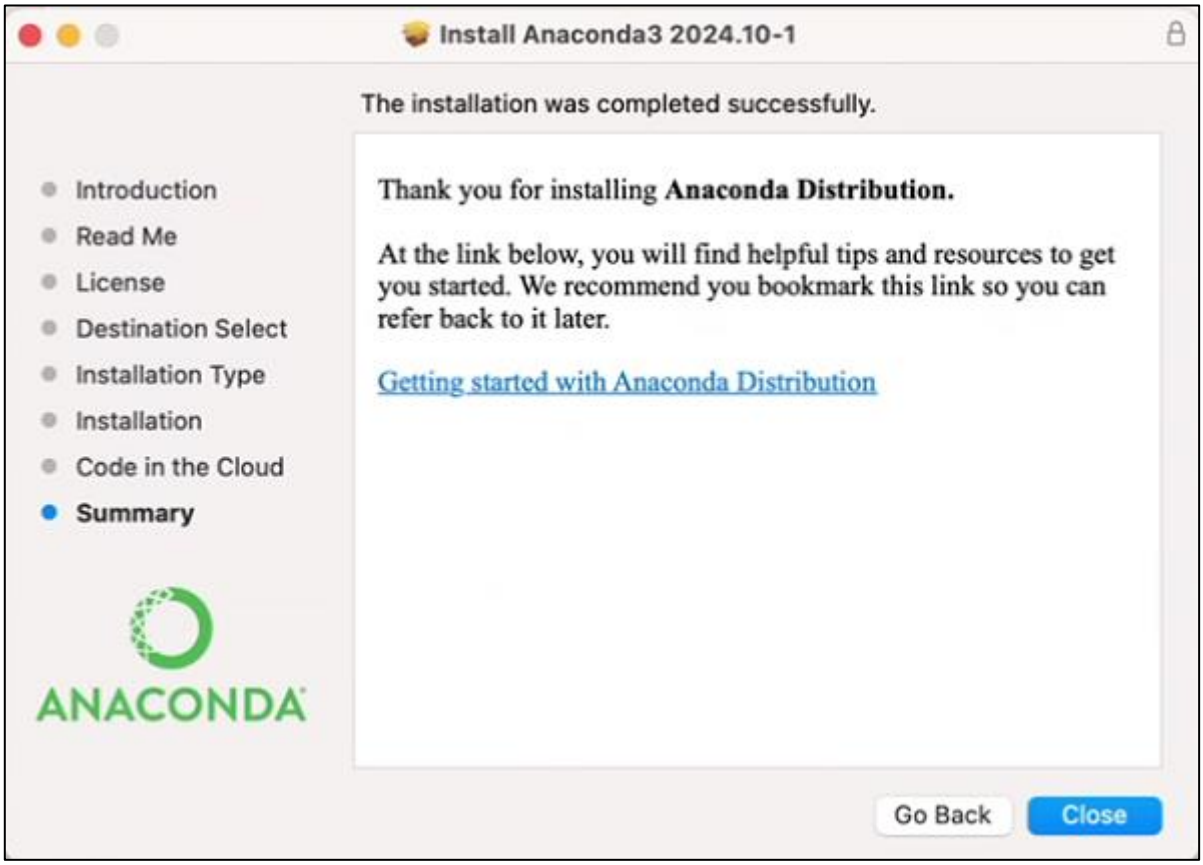
Wait for minutes for the installation to complete.



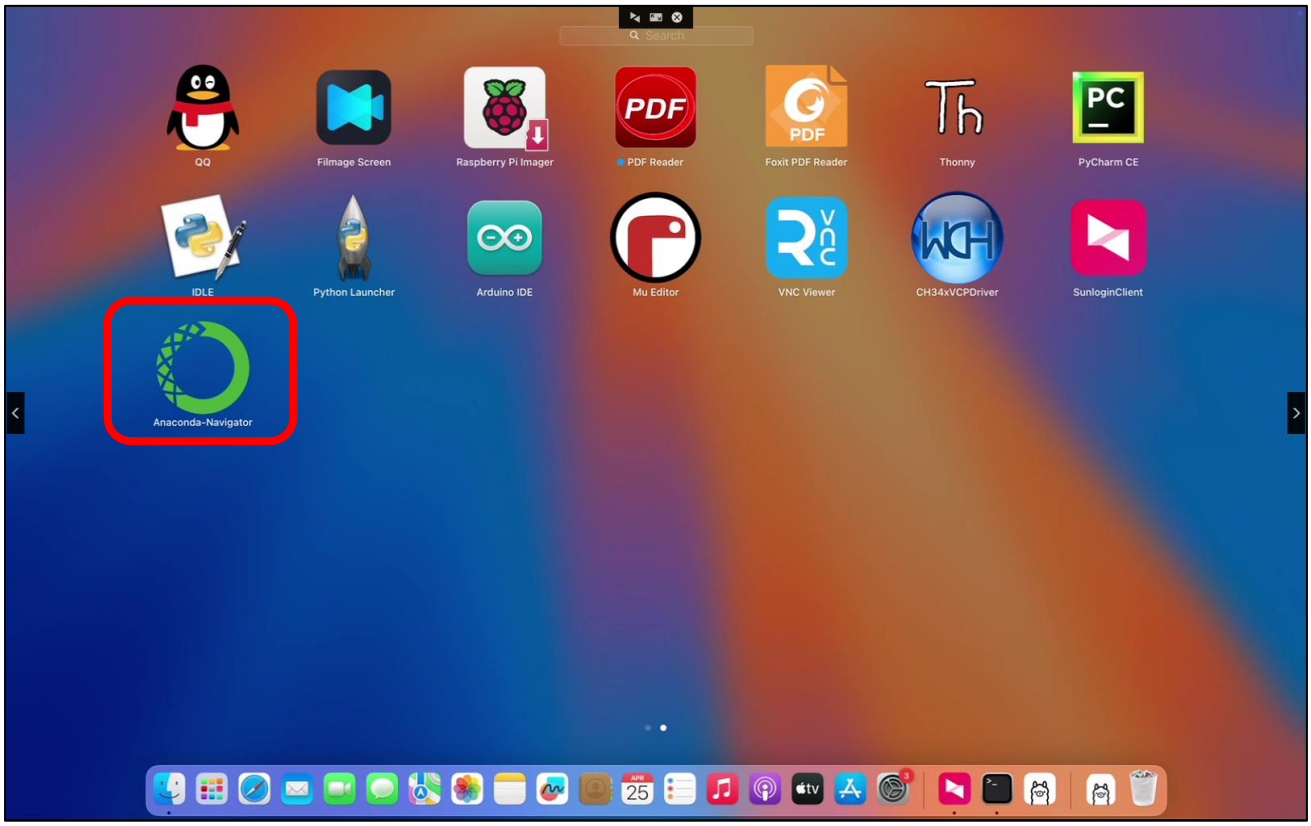
Click Continue.



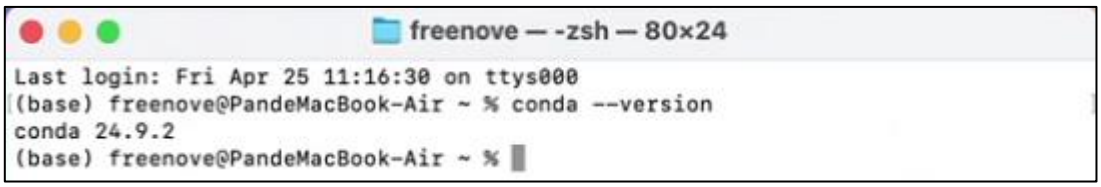
Click Close.



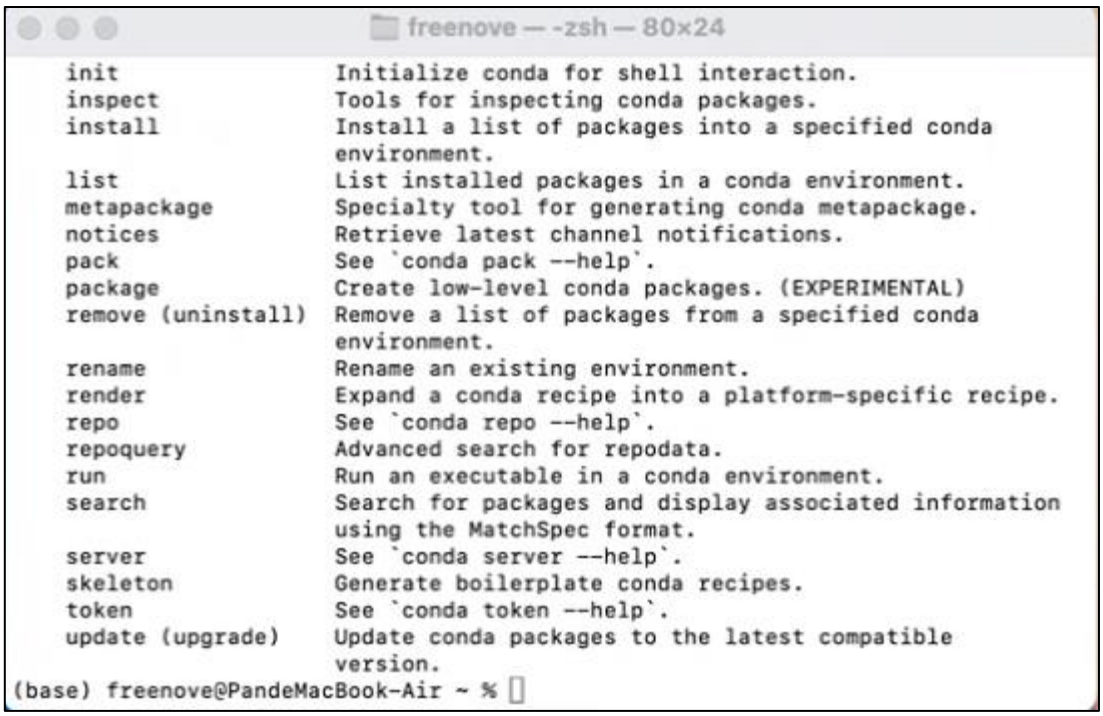
You have now successfully installed Conda. The application will be available in your programs/applications lists.



Double-click to run it. This step will not produce any visible response.
Then reopen the terminal. You will notice the "(base)" prompt appearing.
You can also check the conda version by running the command **conda --version**.



You can use conda -h to view more usage instructions.

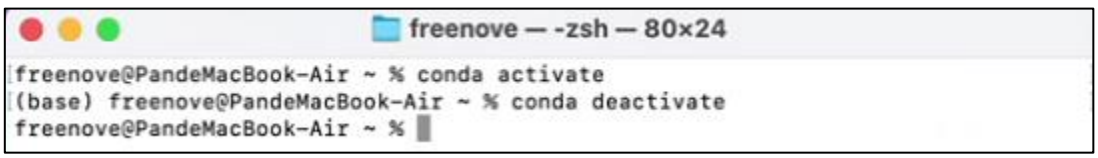


If you are using conda for the first time, you need to run the command **conda init** to initialize and activate the installed conda environment.

```
conda init
```

You can use **conda activate** to enable a virtual environment, or **conda deactivate** to exit it.

```
conda activate
conda deactivate
```

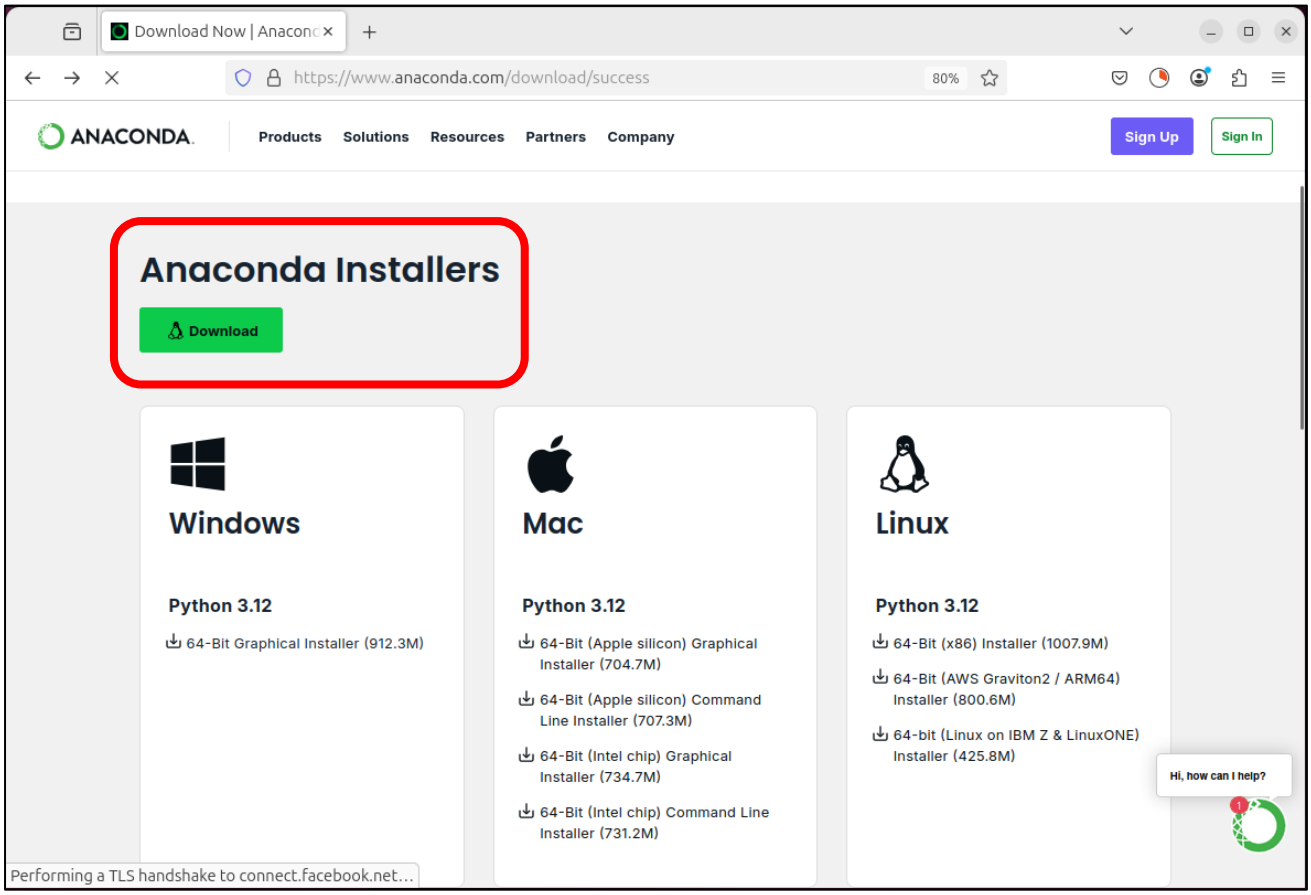


To automatically activate the conda environment upon terminal launch, use: **conda config --set auto_activate_base true**
To disable this auto-activation, use:**conda config --set auto_activate_base false**

```
conda config --set auto_activate_base false
conda config --set auto_activate_base true
```

Linux

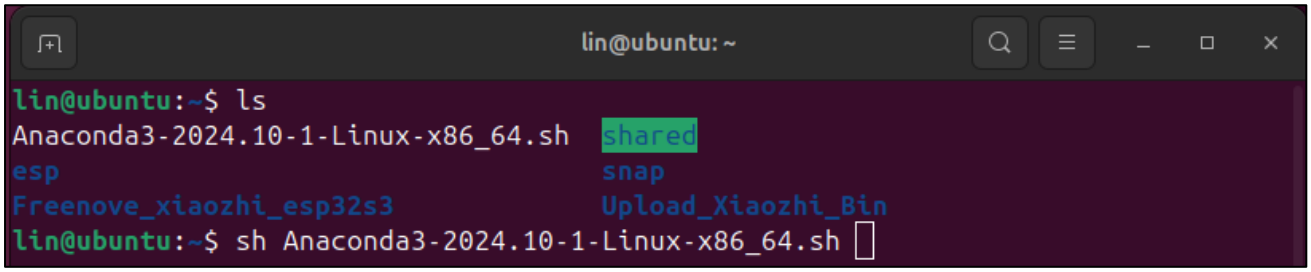
This example uses Conda for dependency management. Therefore, you'll need to have Conda installed on your system beforehand. If you haven't installed Conda yet, you can download and install it from: https://www.anaconda.com/download/success
Select the appropriate installer for your operating system.
Miniconda is an installer by Anaconda that comes preconfigured for use with the Anaconda Repository.



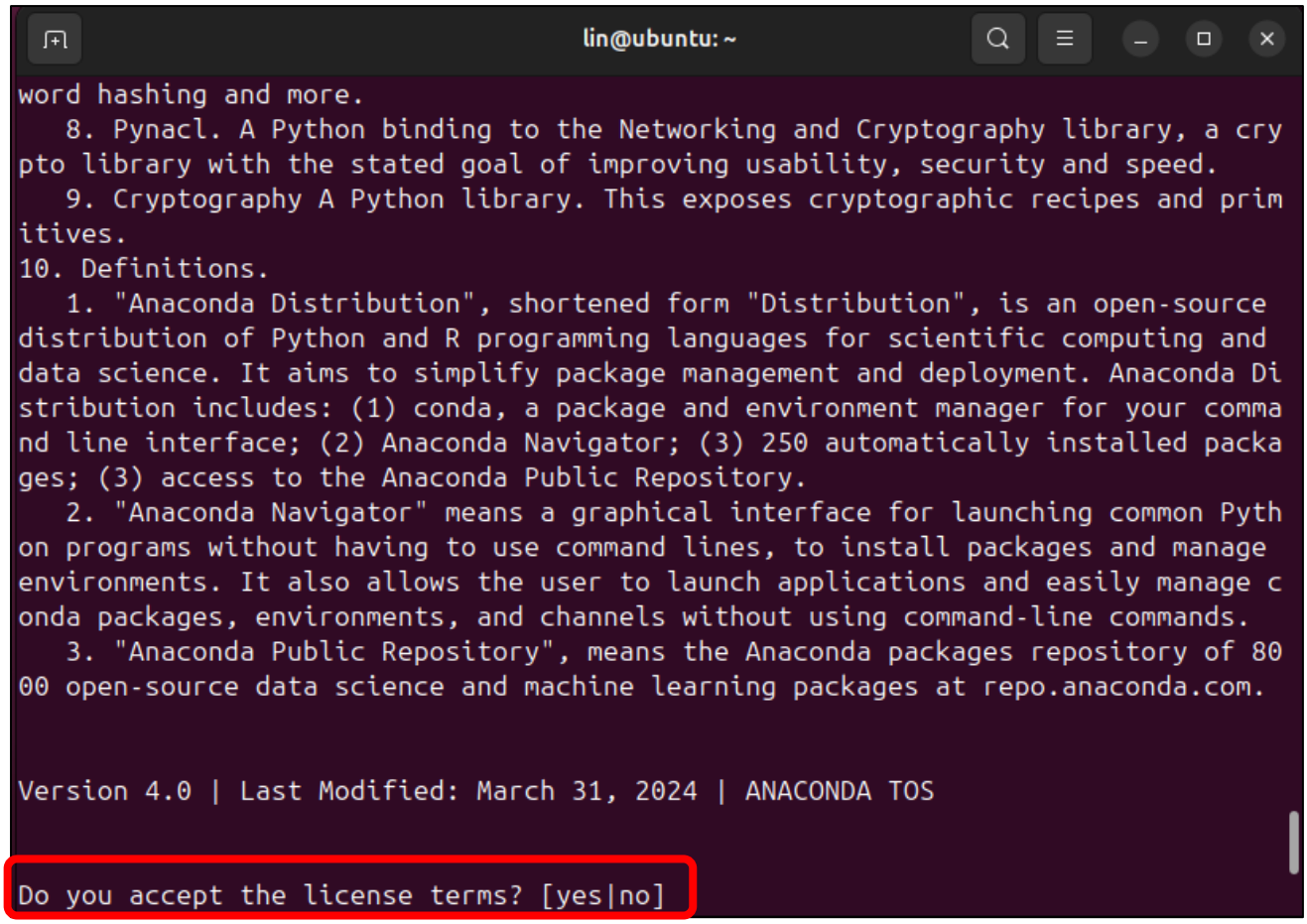
The downloaded file here is named "Anaconda3-2024.10-1-Linux-x86_64.sh". Note that the filename may vary across different computers.

To install Anaconda, open a terminal and execute the following command:

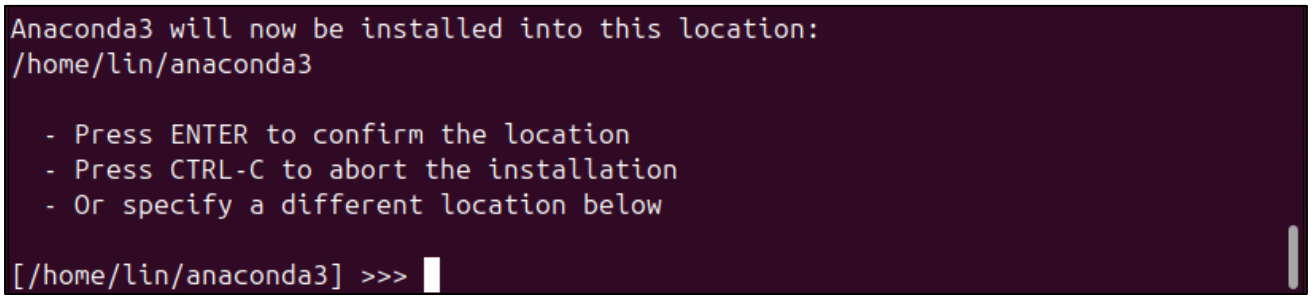
```
sh Anaconda3-2024.10-1-Linux-x86_64.sh
```



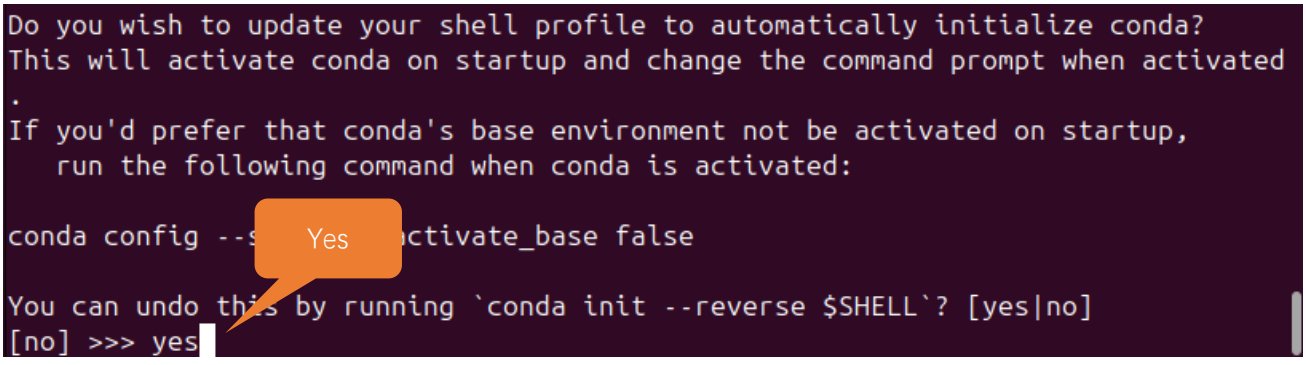
Keep pressing the Enter key and release it until you see the prompt as shown below. Type “Yes”.



Select where to install the application and press Enter. You may use the default location.



The installation requires an internet connection. Please ensure you have a stable network connection and wait patiently for a few minutes until the following prompt appears on your screen.
Note: You will need to type Yes to proceed.



The appearance of the following prompt indicates that conda has been successfully installed.

```
You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[no] >>> yes
no change     /home/lin/anaconda3/condabin/conda
no change     /home/lin/anaconda3/bin/conda
no change     /home/lin/anaconda3/bin/conda-env
no change     /home/lin/anaconda3/bin/activate
no change     /home/lin/anaconda3/bin/deactivate
no change     /home/lin/anaconda3/etc/profile.d/conda.sh
no change     /home/lin/anaconda3/etc/fish/conf.d/conda.fish
no change     /home/lin/anaconda3/shell/condabin/Conda.psm1
no change     /home/lin/anaconda3/shell/condabin/conda-hook.ps1
no change     /home/lin/anaconda3/lib/python3.12/site-packages/xontrib/conda.xsh
no change     /home/lin/anaconda3/etc/profile.d/conda.csh
no change     /home/lin/.bashrc
No action taken.
Thank you for installing Anaconda3!
lin@ubuntu:~$
```

To automatically activate the conda environment upon terminal launch, use: **conda config --set auto_activate_base true**

To disable this auto-activation, use: **conda config --set auto_activate_base false**

```
conda config --set auto_activate_base false
conda config --set auto_activate_base true
```

We do not recommend auto-activation. Therefore, run "conda config --set auto_activate_base false"

```
lin@ubuntu:~$ conda config --set auto_activate_base false
lin@ubuntu:~$
```

Reopen the Terminal, run the command conda --version to check the conda version.

```
conda --version
```

```
lin@ubuntu: ~
lin@ubuntu:~$ conda --version
conda 24.9.2
lin@ubuntu:~$
```

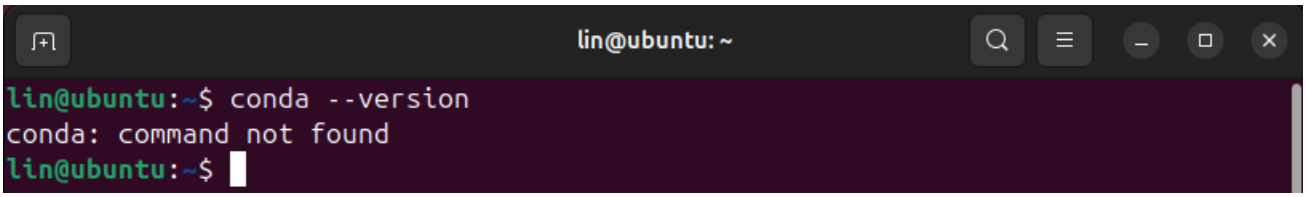
The following two commands allow you to activate or exit the conda virtual environment.

```
conda activate
conda deactivate
```

```
lin@ubuntu: ~
lin@ubuntu:~$ conda --version
conda 24.9.2
lin@ubuntu:~$ conda activate
(base) lin@ubuntu:~$ conda deactivate
lin@ubuntu:~$
```

If you see the following error when checking the conda version,

```
conda –version
```

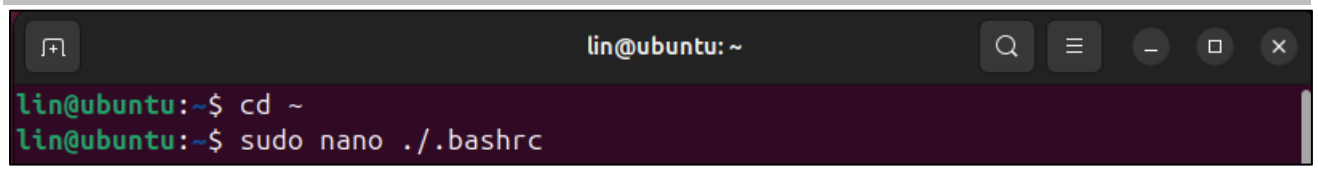


it indicates that while Conda is installed, it hasn't been added to your PATH environment variable.

Please follow these steps to add Conda to your PATH:

Edit the "**.bashrc**" file using nano:

```
cd ~
sudo nano ./.bashrc
```



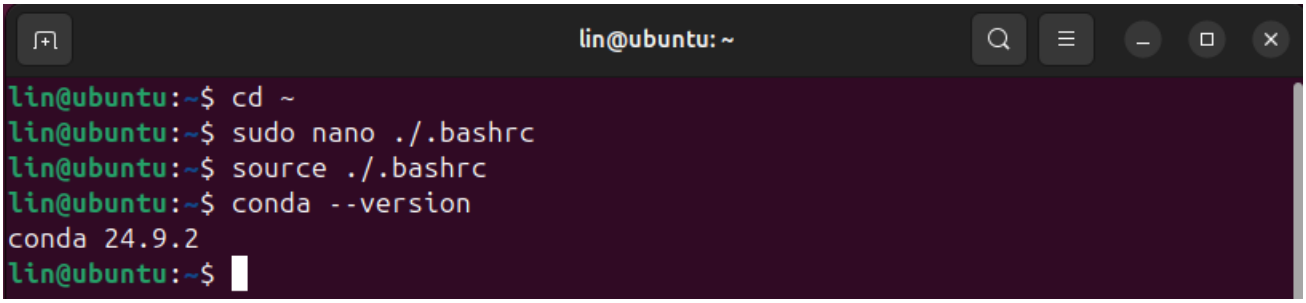
Add the following contents to the end of the file.

```
export PATH="$HOME/anaconda3/bin:$PATH"
```

Press “Ctrl+O” to save the file and “Ctrl+X” to exit editing.

Run the source command to have it take effect, and check the conda version again.

```
source ./.bashrc
conda --version
```

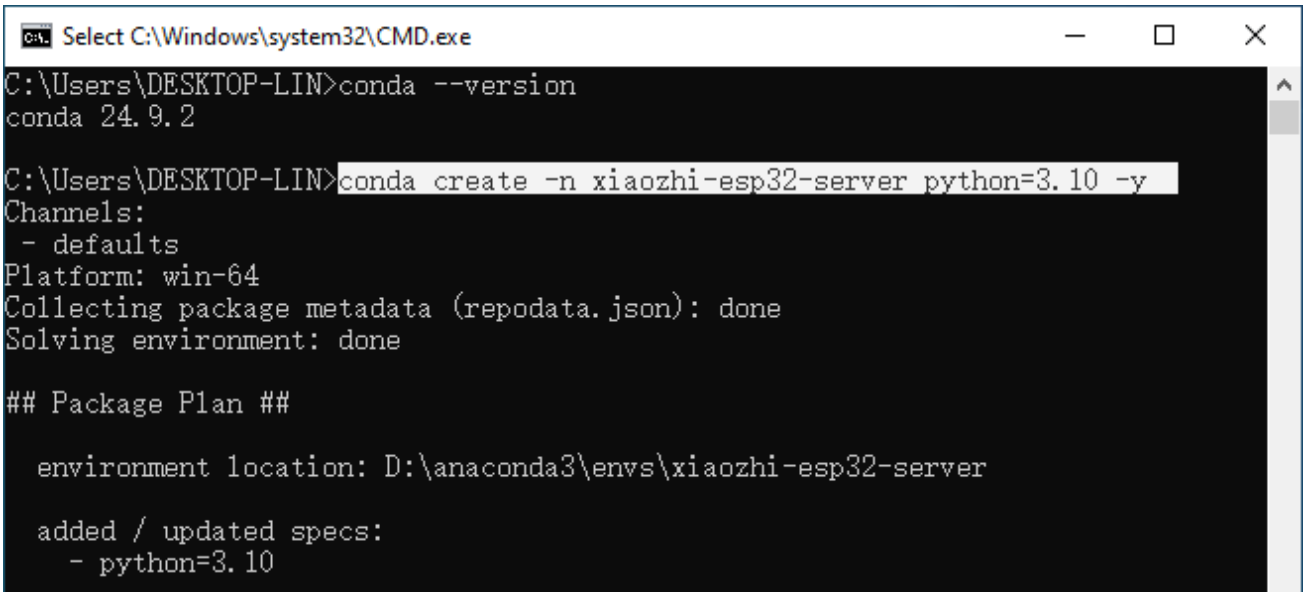


Deploying Virtual Environment

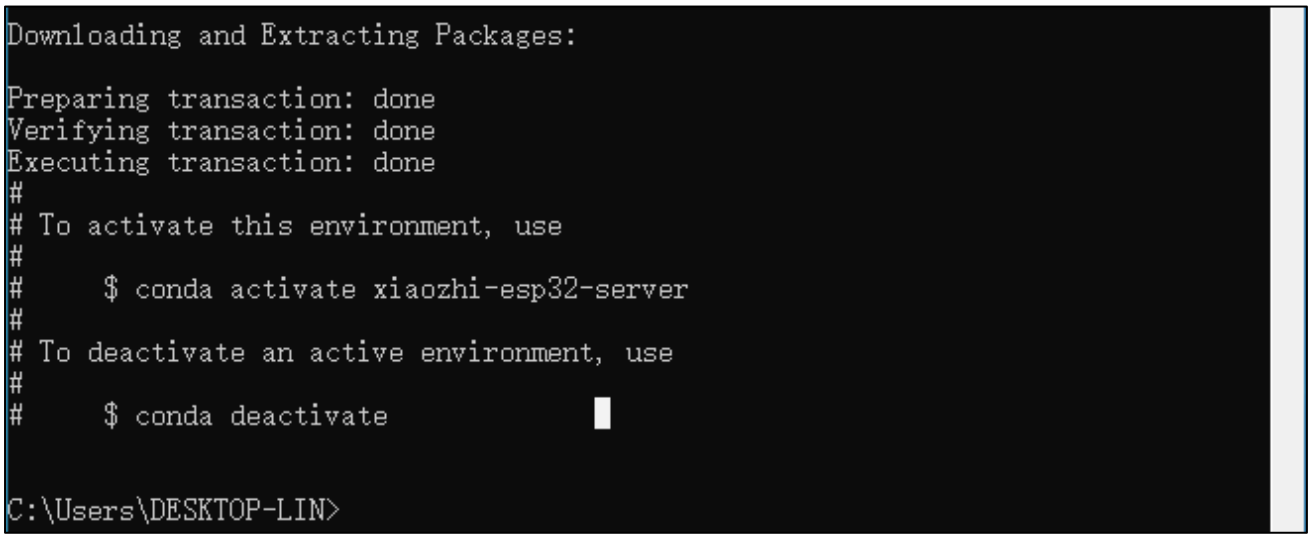
Please note that the commands for deploying virtual environments are universal across Windows, Mac, and Ubuntu systems. The examples shown here use Windows, but the same operations apply to other platforms.

Open the CMD/Terminal interface, run the following command to create a virtual environment named "xiaozhi-esp32-server" with Python 3.10 pre-installed.

```
conda create -n xiaozhi-esp32-server python=3.10 -y
```

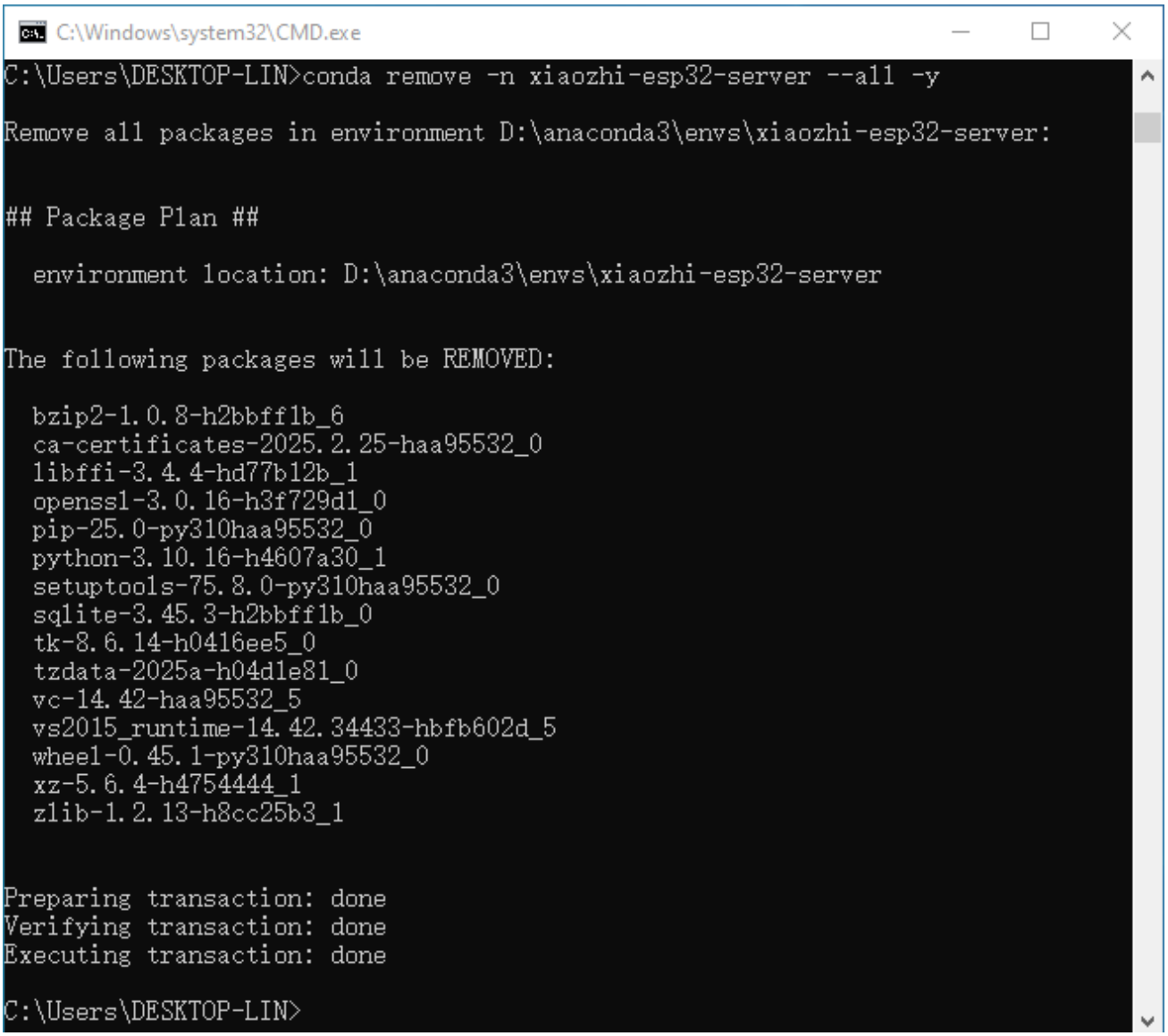


When you see the following messages, it indicates that the virtual environment has been created.



To delete the virtual environment, run the following command:

```
conda remove -n xiaozhi-esp32-server --all -y
```



You can also use the following two commands to activate or exit the virtual environment.

```
conda activate xiaozhi-esp32-server
conda deactivate
```

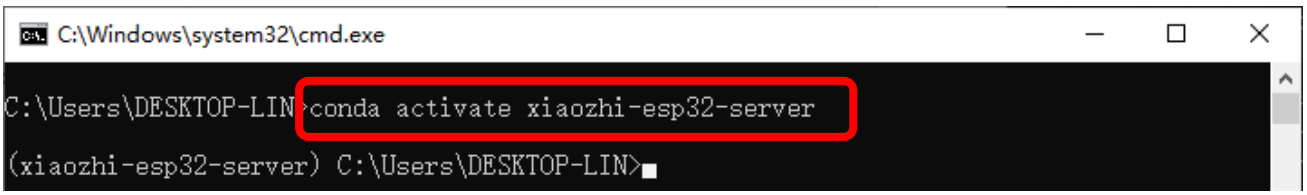
Important Note:

If you receive a prompt suggesting to run conda init when activating your environment, execute “conda init” and restart your terminal for changes to take effect

Deploying xiaozhi-esp32-server

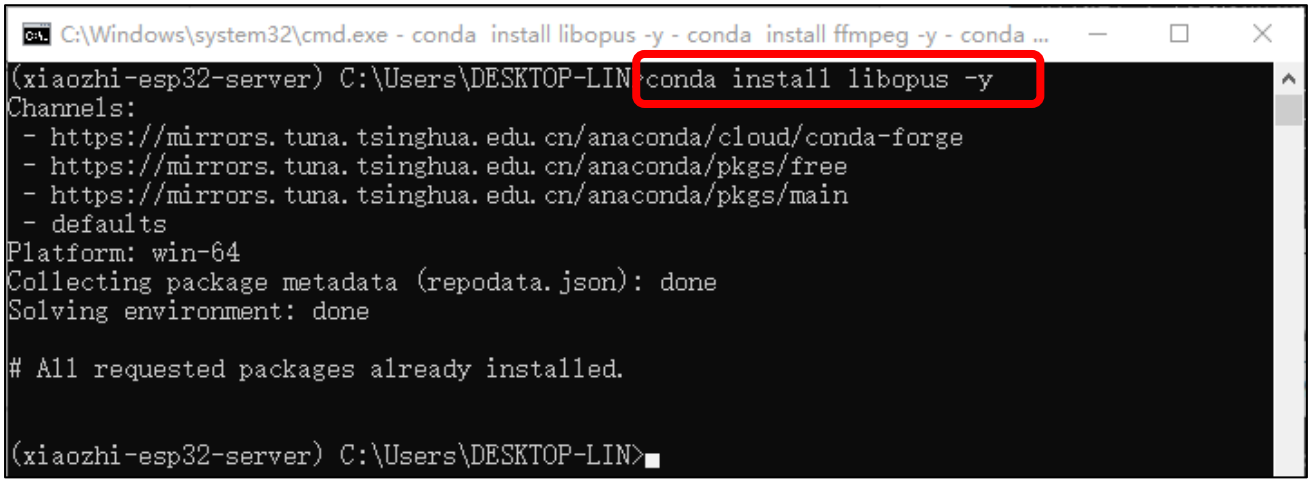
If you're a Windows user, open the Command Prompt (CMD).
For macOS or Ubuntu users, launch the Terminal instead.
The tutorial primarily uses Windows screenshots for demonstration. Where differences exist, we'll provide corresponding examples from other operating systems.
Activate the virtual environment.

```
conda activate xiaozhi-esp32-server
```



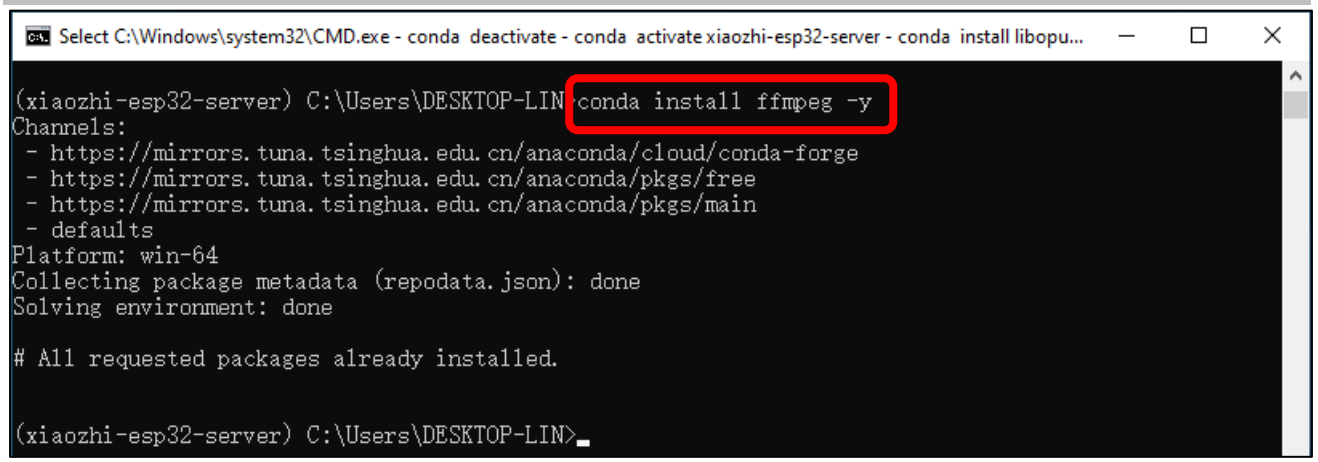
Install libopus under the virtual environment.

```
conda install libopus -y
```



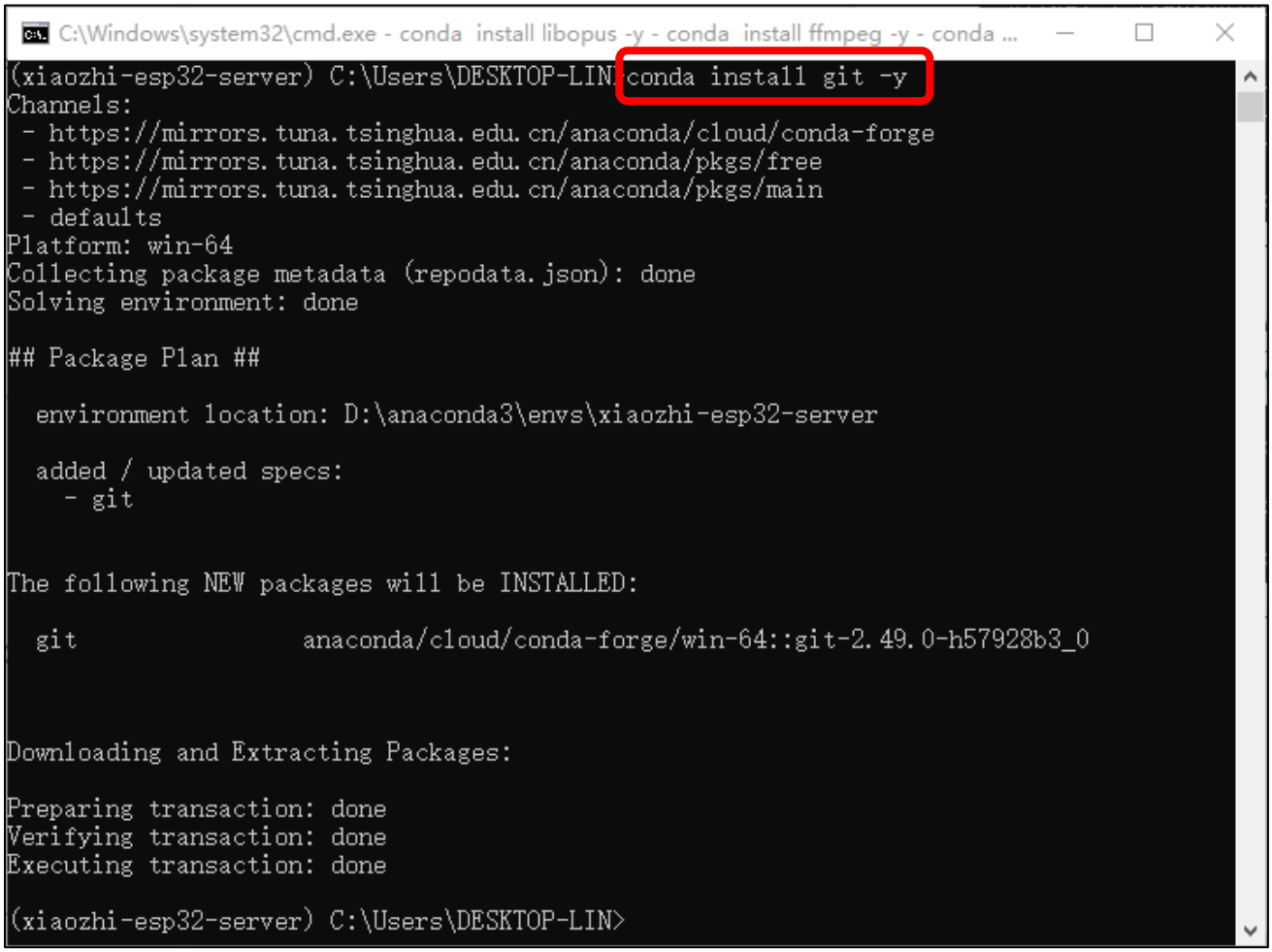
Install ffmpeg under the virtual environment.

```
conda install ffmpeg -y
```



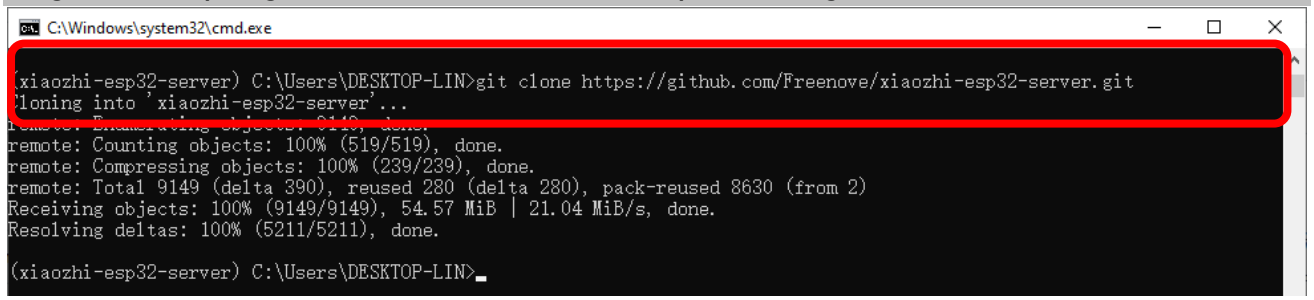
Install git under the virtual environment.

```
conda install git -y
```



Use the git clone command to download the source code of the server.

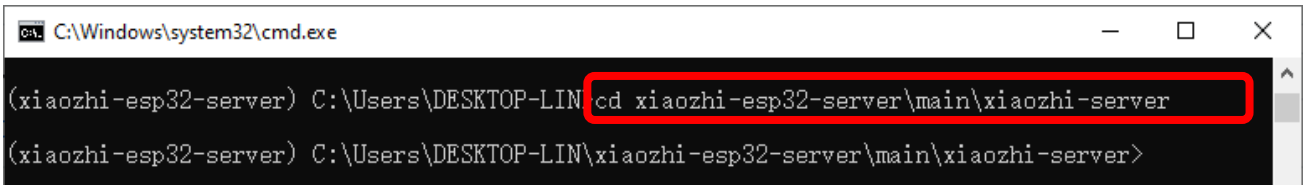
```
git clone https://github.com/Freenove/xiaozhi-esp32-server.git
```



Navigate to the server's source code directory.

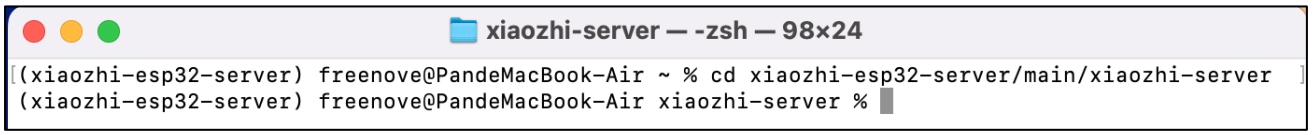
Windows users: Use backslashes (\) in paths

```
cd xiaozhi-esp32-server\main\xiaozhi-server
```



Mac or Linux users: Use forward slashes (/) in paths

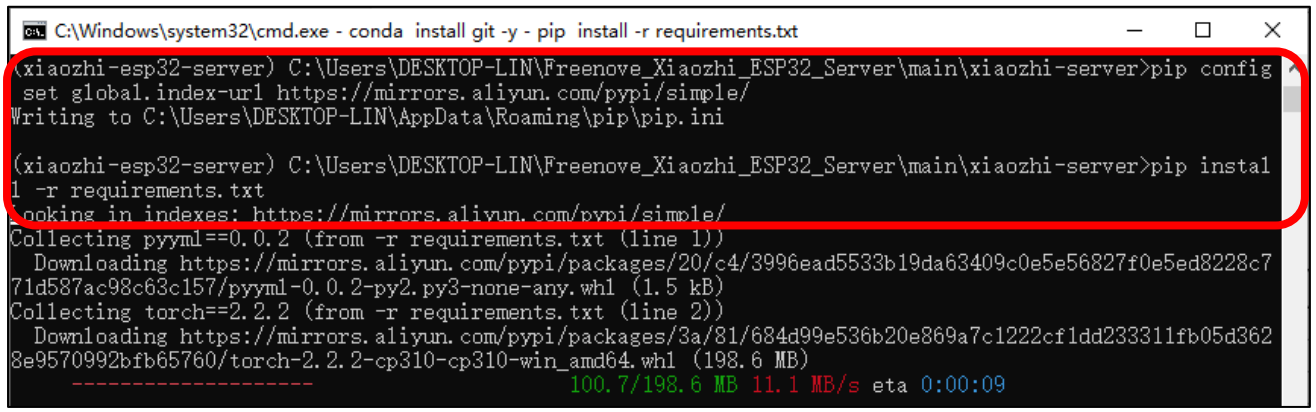
```
cd xiaozhi-esp32-server/main/xiaozhi-server
```



Install the required libraries for the server source code.

This process may take some time — ensure you have a stable internet connection and do not interrupt the installation.

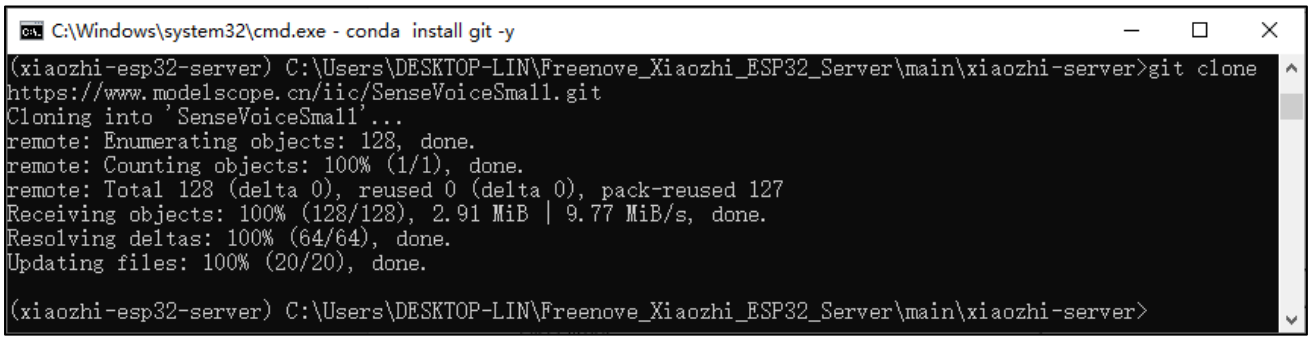
```
pip config set global.index-url https://mirrors.aliyun.com/pypi/simple/
pip install -r requirements.txt
```



The installation is complete when the output matches the following screenshot.

Install the voice model.

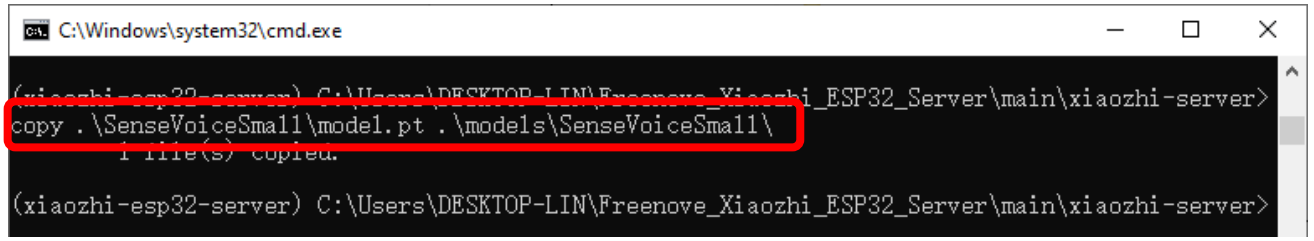
```
git clone https://www.modelscope.cn/iic/SenseVoiceSmall.git
```



Use the copy command to copy the model.pt file from SenseVoiceSmall to the models/SenseVoiceSmall folder.

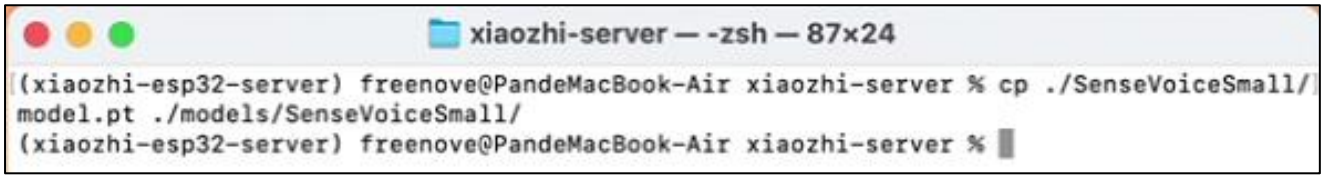
If you are a Windows user, use the copy command.

```
copy .\SenseVoiceSmall\model.pt .\models\SenseVoiceSmall\
```



If you are a Mac or Linux user, use the cp command.

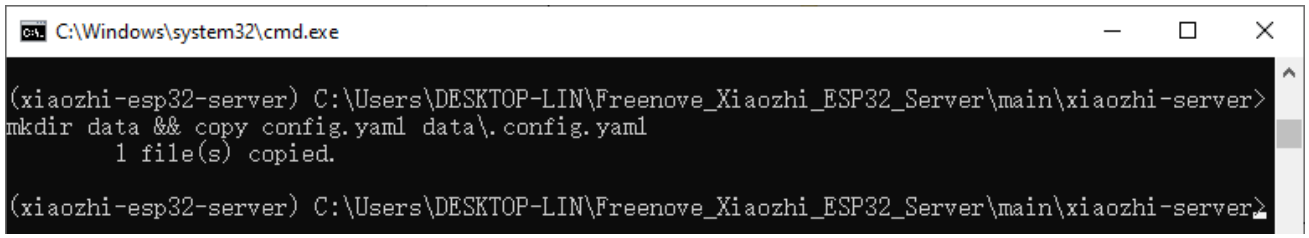
```
cp ./SenseVoiceSmall/model.pt ./models/SenseVoiceSmall/
```



Entering the command "mkdir data && copy config.yaml data.config.yaml" in the CMD interface, it will create a folder named "data" in the xiaozhi-server and copy the "config.yaml" file from the current directory into the "data" folder, renaming it as ".config.yaml".

If you are a Windows user, please execute:

```
mkdir data && copy config.yaml data\.config.yaml
```



If you are a MAC/Linux user, run the following one:

```
mkdir data && cp config.yaml data/.config.yaml
```

Open and modify the config.yaml.

On Windows, run:

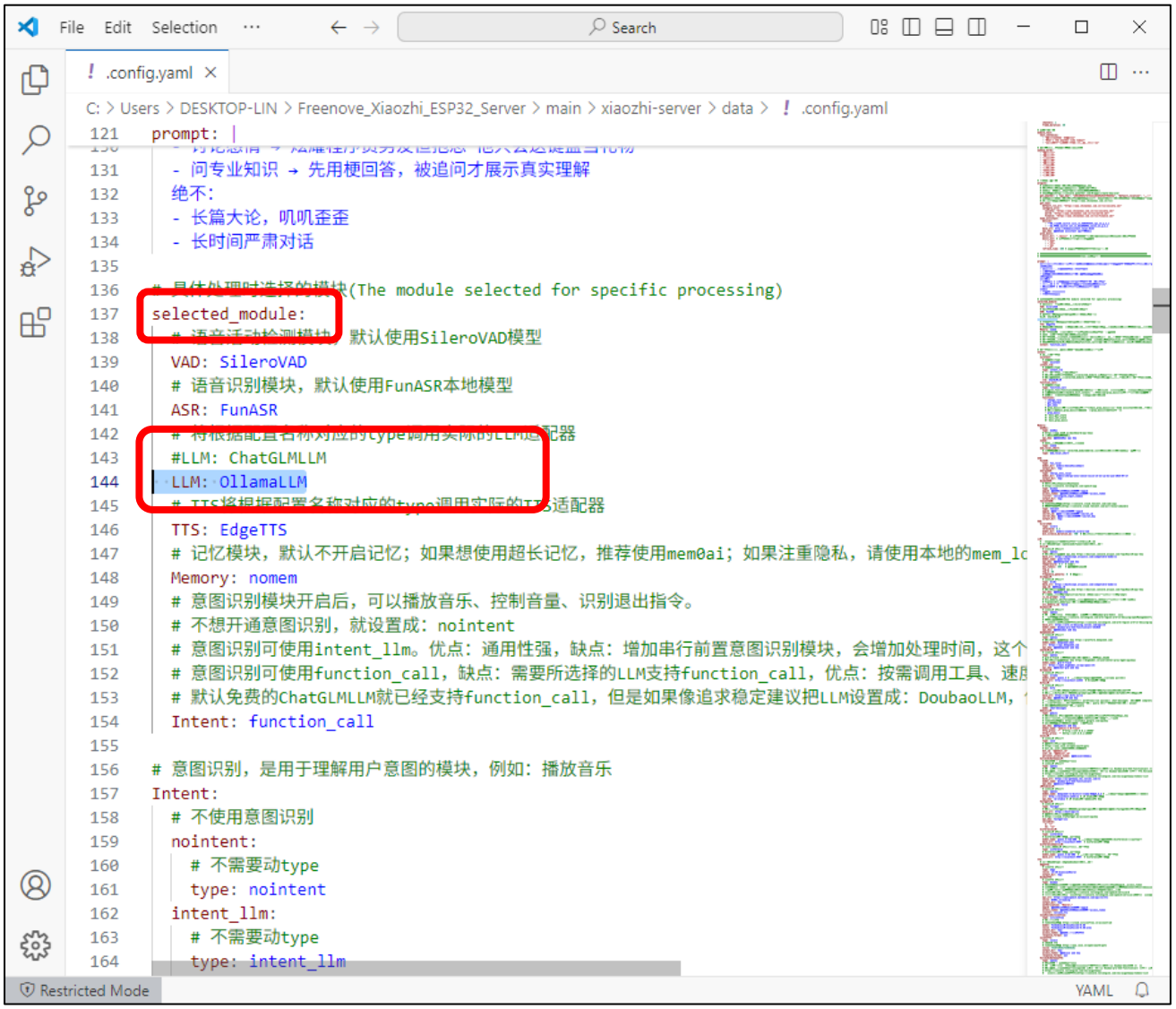
```
code .\data\.config.yaml
```

On Mac/Linunx. Run:

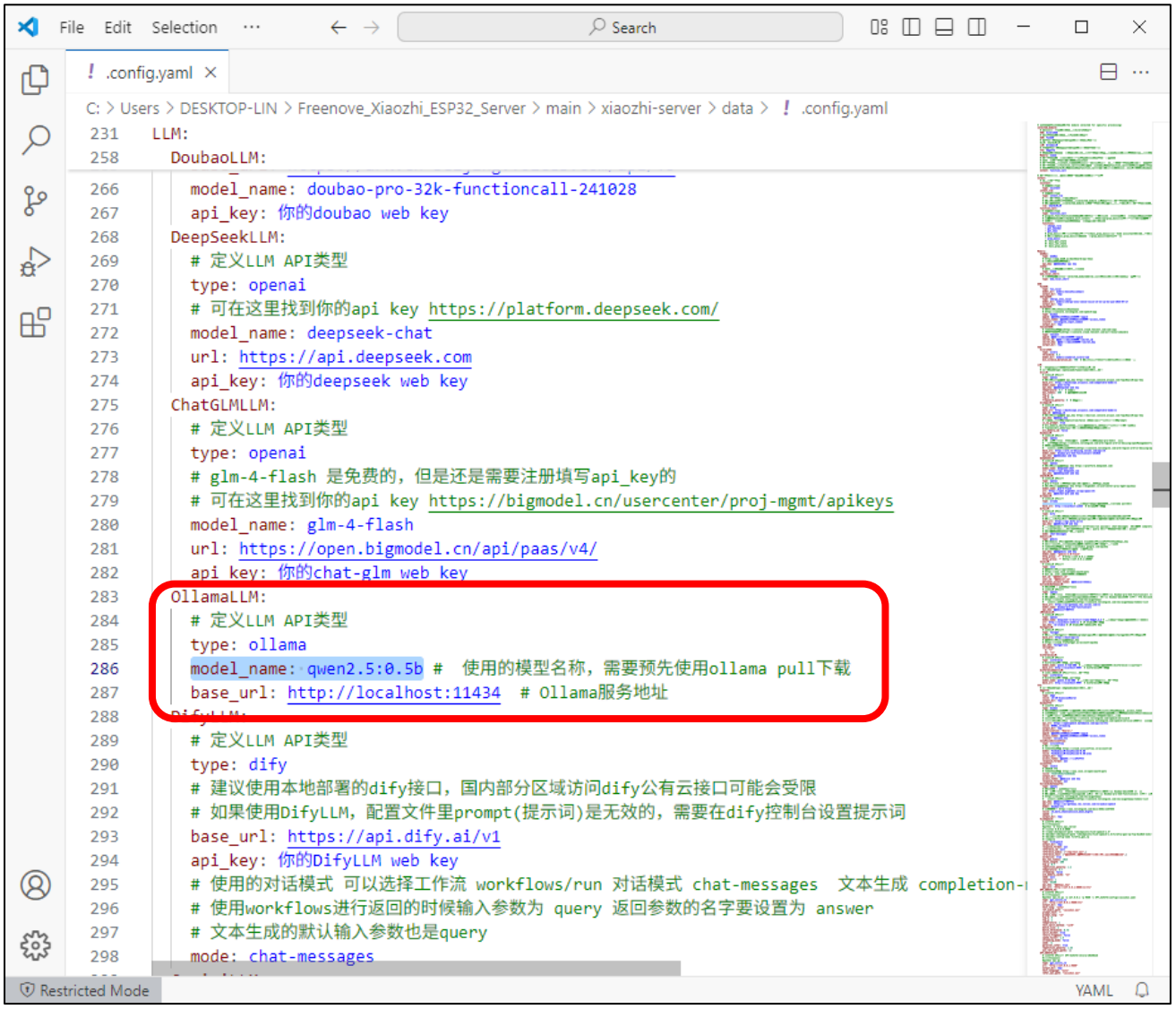
```
code ./data/.config.yaml
```

Note: If your VSCode is not properly installed, running the command may result in an error. You can also manually open this file using VSCode.

Find"**selected_module:** , change "**LLM: ChatGLMLLM**" to "**LLM: OllamaLLM**"



Find "**OllamaLLM:**" under "**LLM:**", change "**model_name: qwen2.5**" to "**model_name: qwen2.5:0.5b**".



Important Note: To ensure smooth usage of XiaoZhi AI's visual recognition feature, please follow the steps below to configure the Visual Recognition Large Model (VLLM). If you do not require this feature at the moment, you may skip this step and proceed with the next steps.

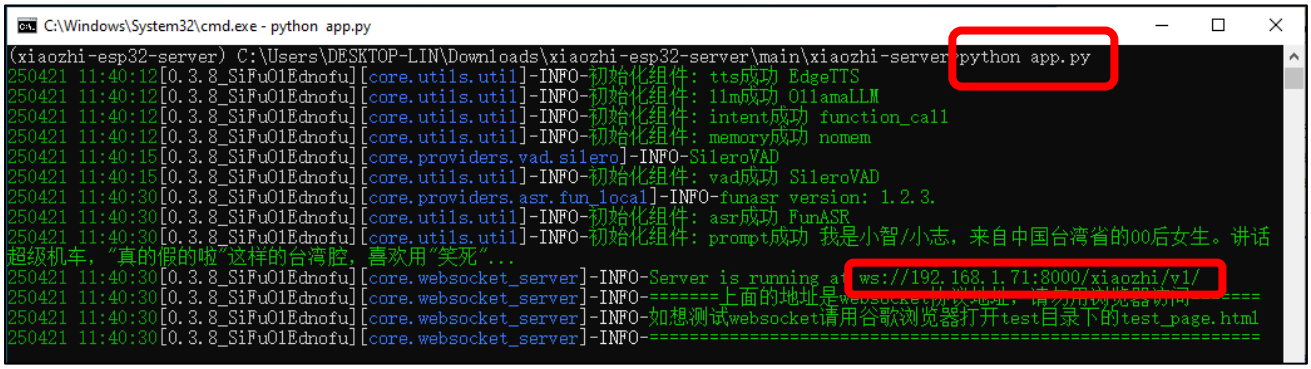
Continue editing the config.yaml file: First, follow the prompted steps to register for the corresponding API key.Then, insert the generated API key into the code.

Save and exit the file.

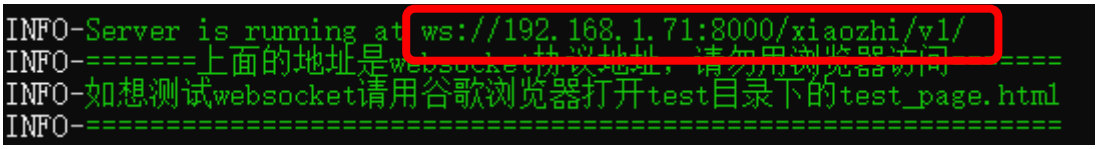
You can also choose other models, such as the default ChatGLM-LLM. Please note that configuring different LLM models requires you to explore and set them up manually.

Run the xiaozhi-esp32-server code.

```
python app.py
```

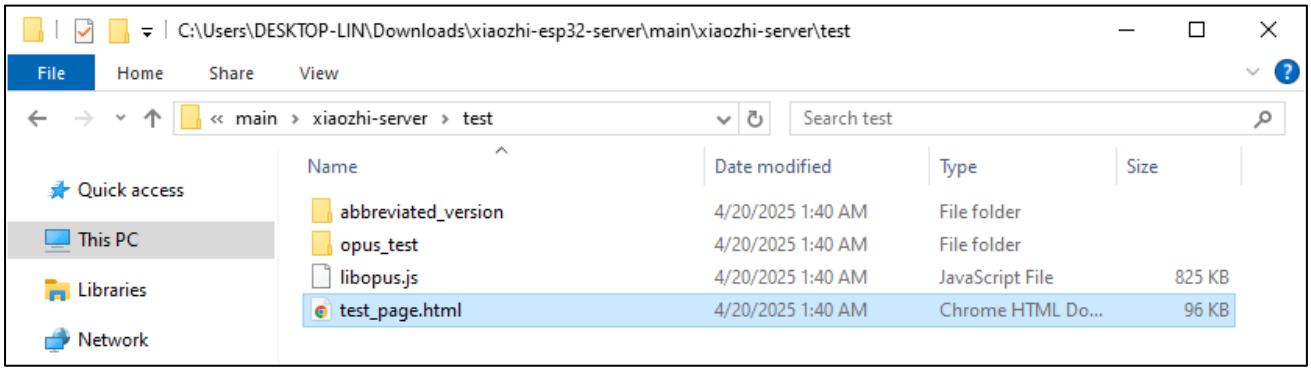


Note: The server will now show an access port—remember it, as you'll need it later in the tutorial.



At this point, you can use a browser to open the HTML file located in xiaozhi-esp32-server\main\xiaozhi-server\test.

The testing steps are as follows.



Click “Connect”.

Test xiaozhi-esp32-server by typing any message and clicking "Send".

If the server is running properly, you can start chatting with it.

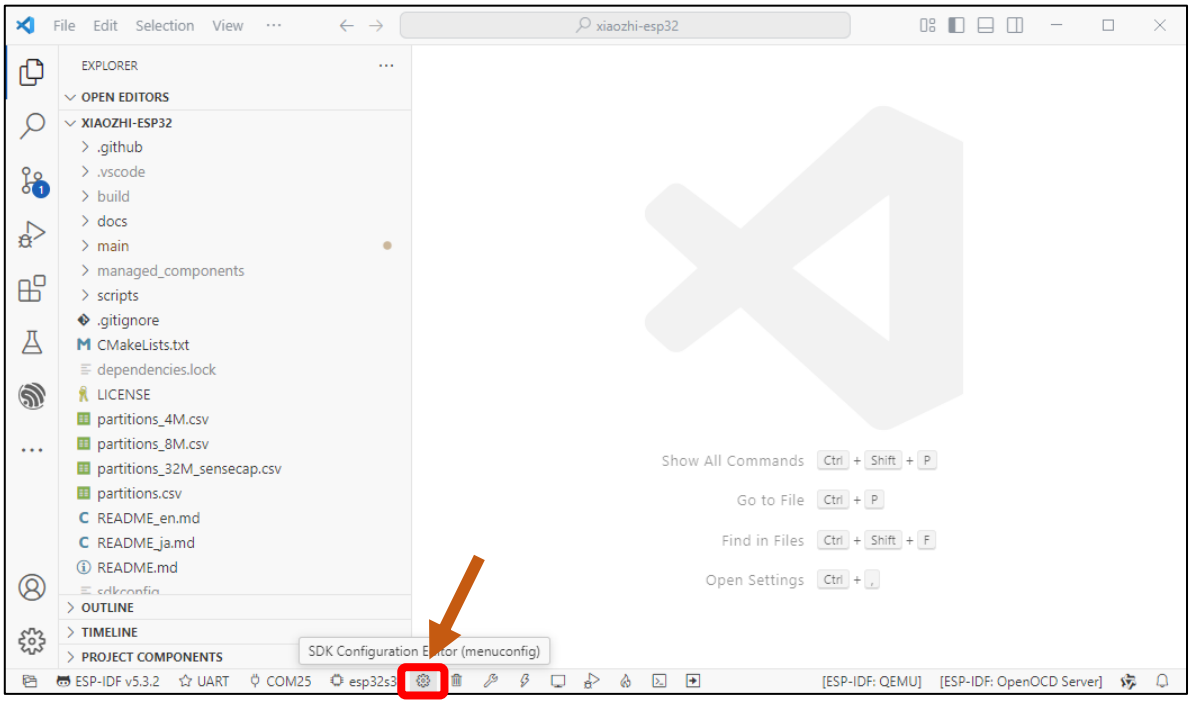
Important: Both xiaozhi-esp32-server and Ollama must be running simultaneously. If Ollama is not active, you'll see an error message like the example below.

You can refer to LLM Model to run Ollama.

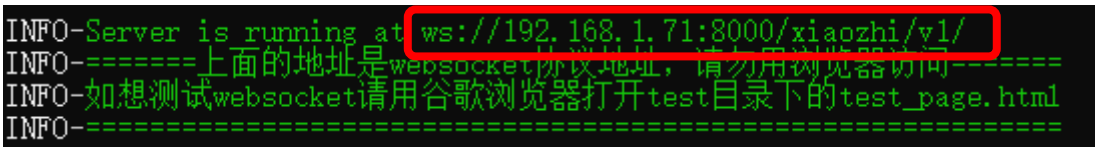
Visiting xiaozhi-esp-server via ESP32S3

Please note that in the previous code, we explained the configuration of the XiaoZhi AI code. In this chapter, we need to modify the project configuration to enable the ESP32S3 to access the local server of xiaozhi-esp32-server.

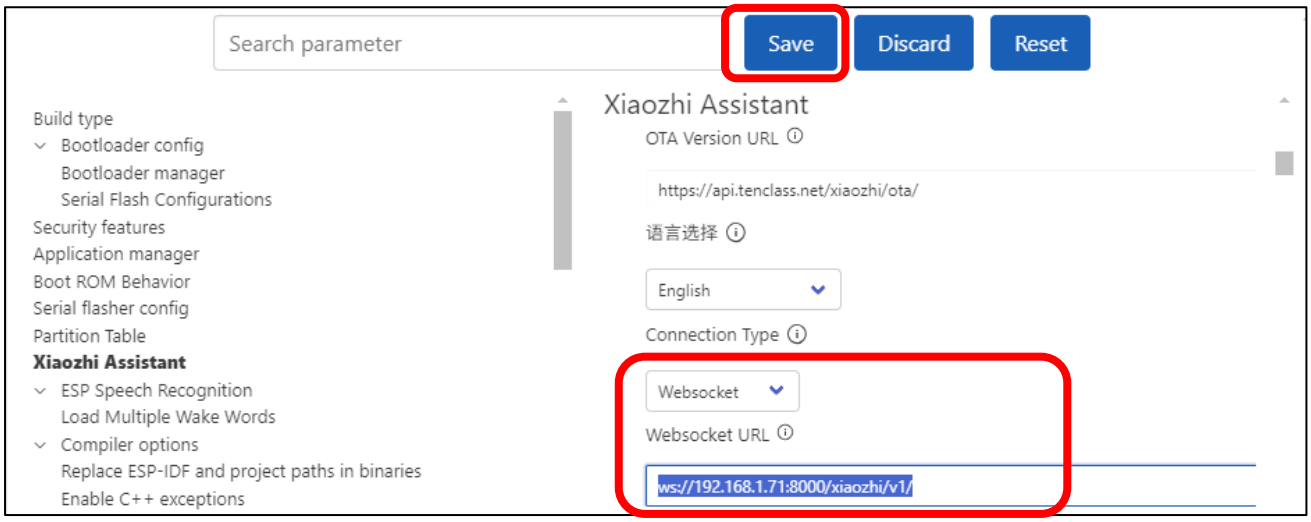
Open Visual Studio Code and select the previous xiaozhi-esp32 project. Click on the SDK Configuration Editor (menuconfig).



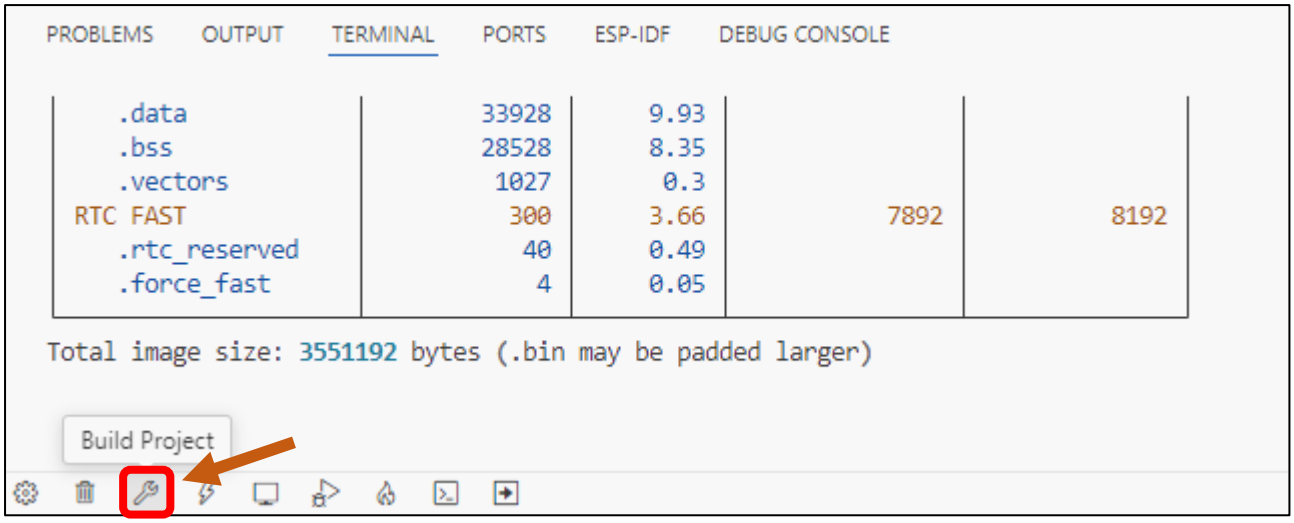
Set the Connection Type to “Websocket”, and type in the access port that the xiaozhi-esp32-server previously printed to connect.



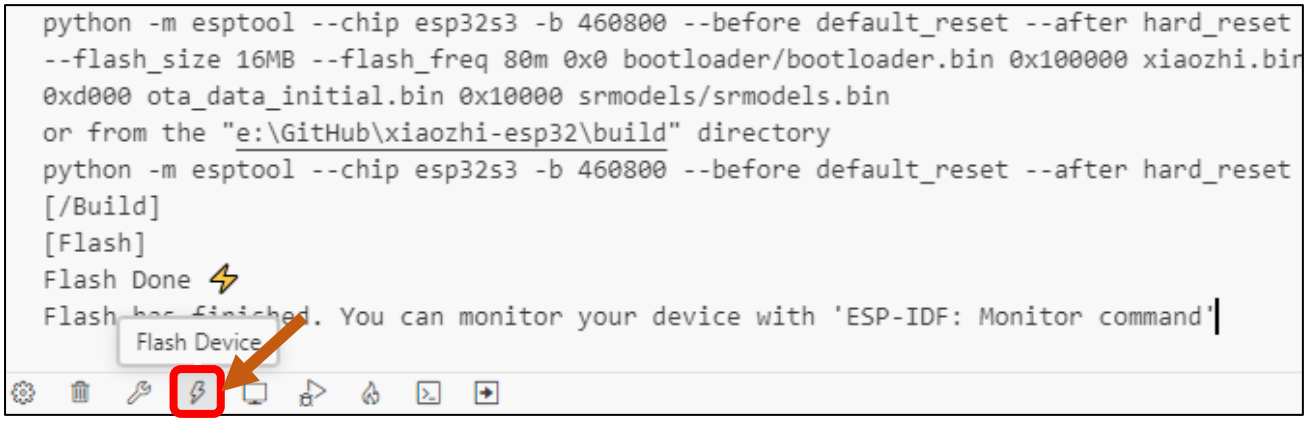
Click save and compile the code again, as shown below.



Click “Build Project” at the bottom of the interface to compile to code.



Click “Flash Device” at the bottom to upload the code to the ESP32S3.



Congratulations! You have now completed the setup for XiaoZhi AI. Simply say "Hi, ESP" into the microphone to start chatting with your local server.

Note:
The local server requires high-performance hardware. If your PC isn't very powerful, try using LLM APIs from big tech firms, as they are less demanding on your system.