Contents

基于小智 AI 的语音助手

该项目采用 Freenove ESP32 S3 显示屏实现 AI 语音助手功能，需具备一定的编程功底，并熟悉 ESP-IDF 开发框架及开源大模型技术。

关于本项目

该语音助手项目（https://github.com/Freenove/xiaozhi-esp32）衍生自开源项目（https://github.com/78/xiaozhi-esp32），可在嵌入式设备上调用主流大语言模型（LLM），通过语音活动检测（VAD）、自动语音识别（ASR）、语音转文本（STT）、文本转语音（TTS）、记忆存储及意图识别等多模块服务实现语音对话功能。Freenove 公司已将其适配至 Media Kit 产品，本文将阐述如何在 Media Kit 上运行该项目。

1、**在线模式**：连接至小智（xiaozhi.me）服务器，目前面向个人用户免费开放体验。

2、**离线模式**：需在本地计算机部署全部服务模块（包括 VAD 语音活动检测、ASR 自动语音识别、STT 语音转文本、TTS 文本转语音、记忆存储、意图识别等）。实际体验完全取决于所选模型性能及本地机器配置。其本地服务器项目（https://github.com/Freenove/xiaozhi-esp32-server）衍生自开源项目（https://github.com/xinnan-tech/xiaozhi-esp32-server）。

模式选择建议：

1、**普通用户**：推荐使用在线版本，可获得稳定 AI 助手服务
2、**开发者用户**：可尝试部署离线版本以深入理解 AI 助手的技术实现架构。但需注意，个人电脑可能难以同时运行所有服务（尤其是核心大语言模型 LLM 服务），可能导致体验不佳，因此该模式主要适用于学习研究场景。

注意事项

- **项目版权声明**
  - 语音助手项目：原开发者"Xiage"，Freenove 公司为适配 Media Kit 进行分支修改，基于 MIT 开源协议发布。
  - 本地服务器项目：原开发者"xinnan-tech"，Freenove 公司同样为 Media Kit 集成进行分支改造，采用 MIT 开源协议。
- **支持国家/地区**
  - **在线版本：**
    服务可用性取决于小智(xiaozhi.me)服务器覆盖范围，部分区域可能不可用。当前支持地区详见：https://xiaozhi.me/login?redirect=/console/agents
    用户体验与服务器连接质量直接相关，若至小智服务器的网络状况不佳可能导致性能下降
  - **离线版本:**
    无地域限制，可在全球所有国家/地区部署

- **支持语言：**
  - 在线版本：当前支持普通话、粤语、英语、日语、韩语等。若您不使用这些语言，可能无法与小智 AI 有效交互。
  - 离线版本：取决于所部署的 ASR 模型。默认 FunASR 模型仅支持普通话、粤语、英语、日语及韩语。
- **费用说明:**
  - 在线版本：当前小智(xiaozhi.me)提供免费服务，但我们无法保证该在线服务器将永久免费。
  - 离线版本：前述子服务中部分为付费项目，部分为免费项目——实际成本取决于您的选择。
- **获取帮助:**
  若遵循教程后仍存在问题，请联系：support@freenove.com
  注意：由于在线服务由小智(xiaozhi.me)提供，若其停止服务，我们将同步移除相关文档、教程及代码。

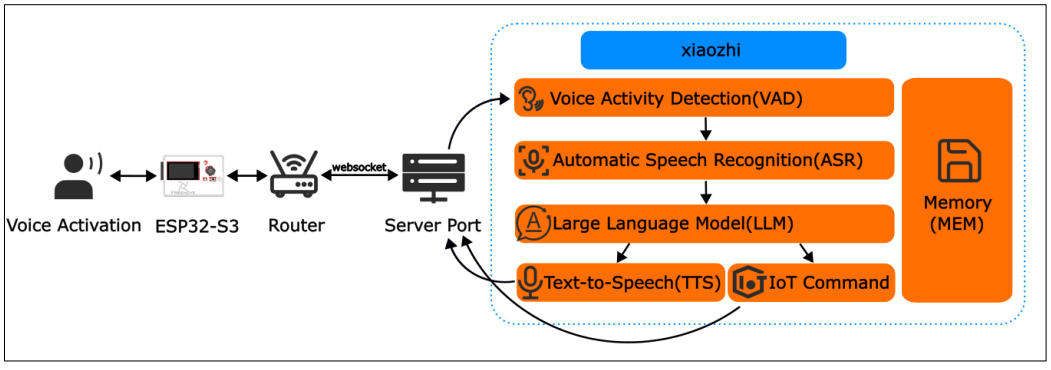
免责声明

本实现基于开源项目（https://github.com/78/xiaozhi-esp32）进行适配开发，仅供第三方学习及 AI 功能测试用途，不提供商业应用支持与推广。本教程仅面向技术爱好者个人学习开发使用。

注意事项：

1. 由于本项目为第三方开源项目，若学习过程中遇到问题，请提交至原项目 issue 跟踪系统：https://github.com/78/xiaozhi-esp32/issues。
2. 当前小智 AI 语音识别仅支持：普通话、粤语、英语、韩语、日语，其他语言暂不支持。
3. 小智服务器界面目前仅支持英文、中文、日文显示，手机号注册仅限以下国家/地区用户（详见下表），其他地区用户暂无法注册

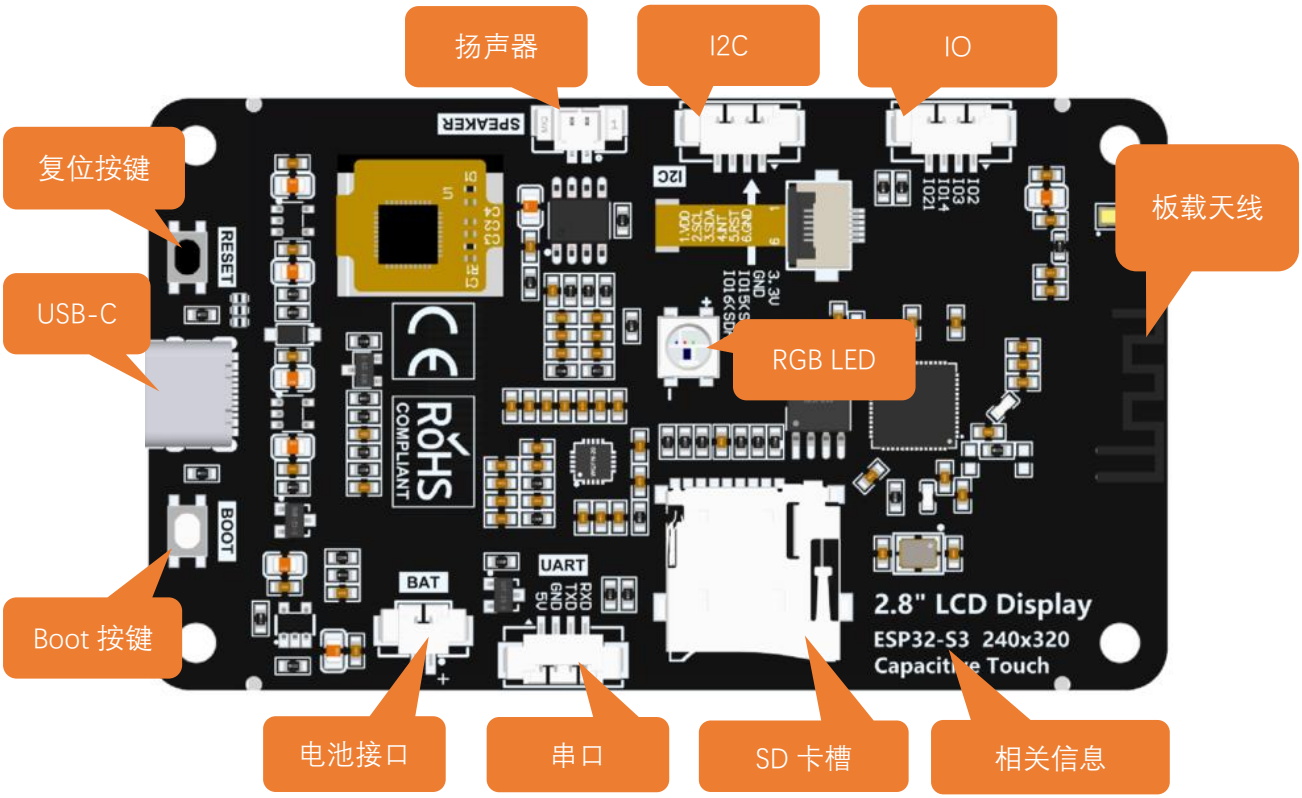
本项目通过 WebSocket 协议实现 ESP32-S3 与小智 AI 服务器的数据通信。



Freenove ESP32 S3 Display

Top	Bottom
	

硬件接口



小智 AI 固件

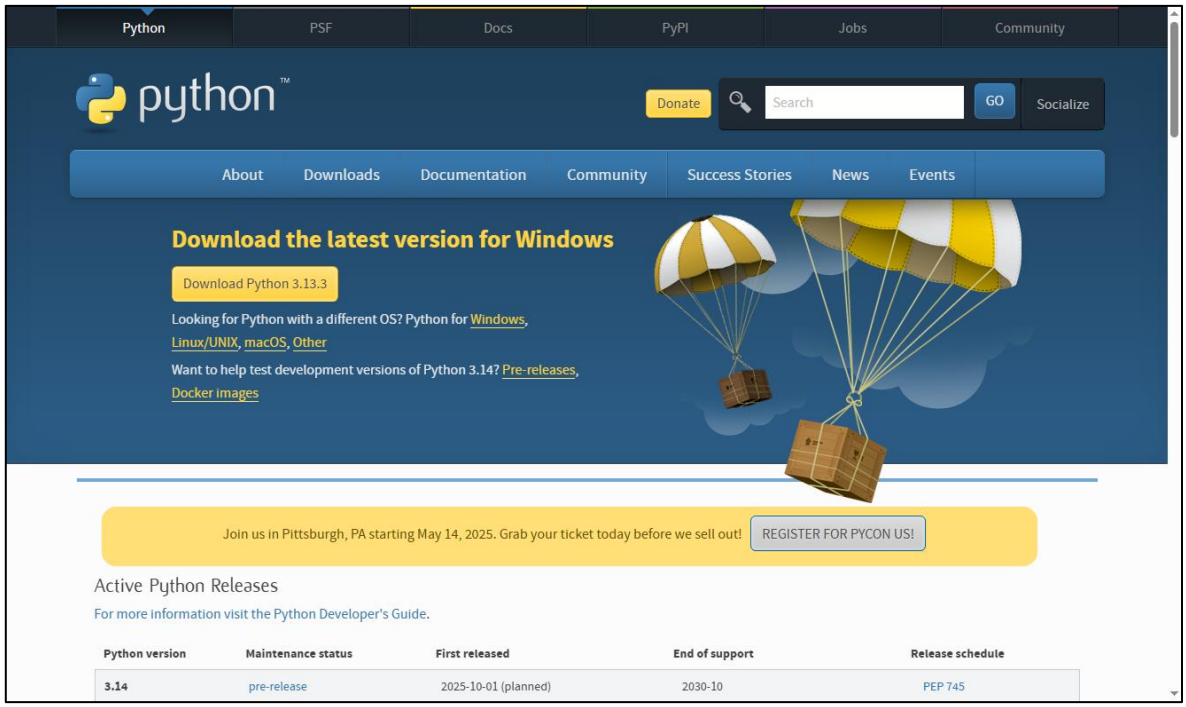
若设备尚未安装小智 AI 固件，请按照后续教程为 ESP32-S3 重新烧录固件。

若设备已预装小智 AI 固件，则可跳过本步骤。

安装 Python（必需）

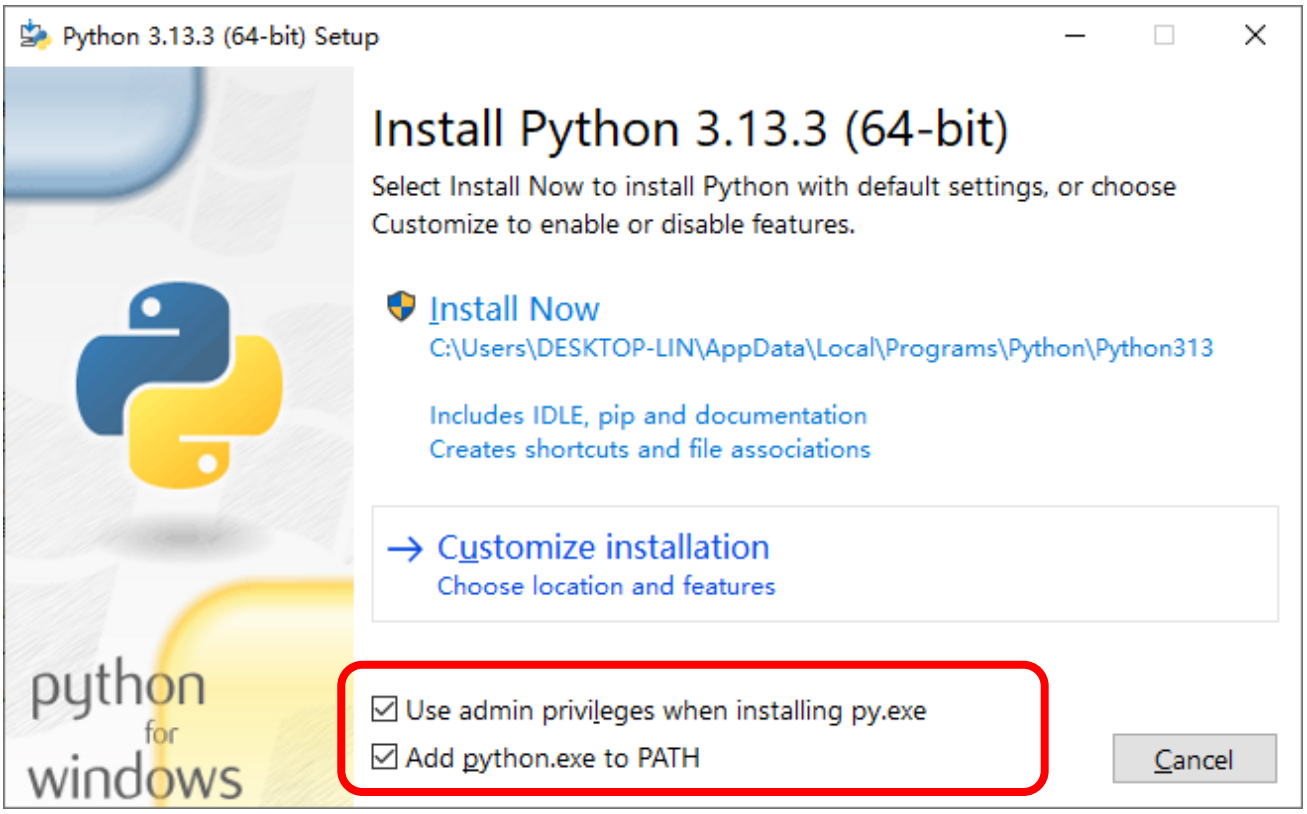
Windows

请下载并安装 Python3 软件包：https://www.python.org/downloads/windows/

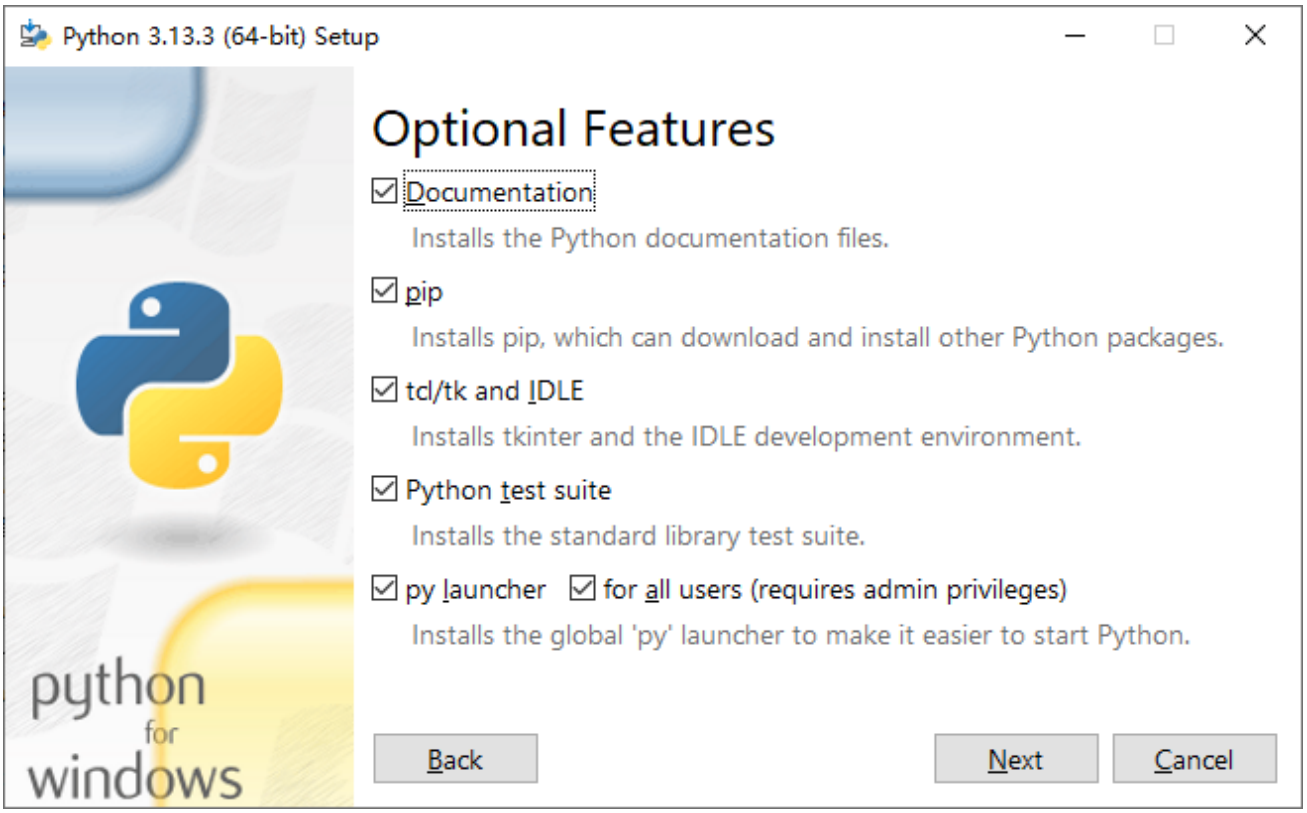


点击下载 Python 3.13.3

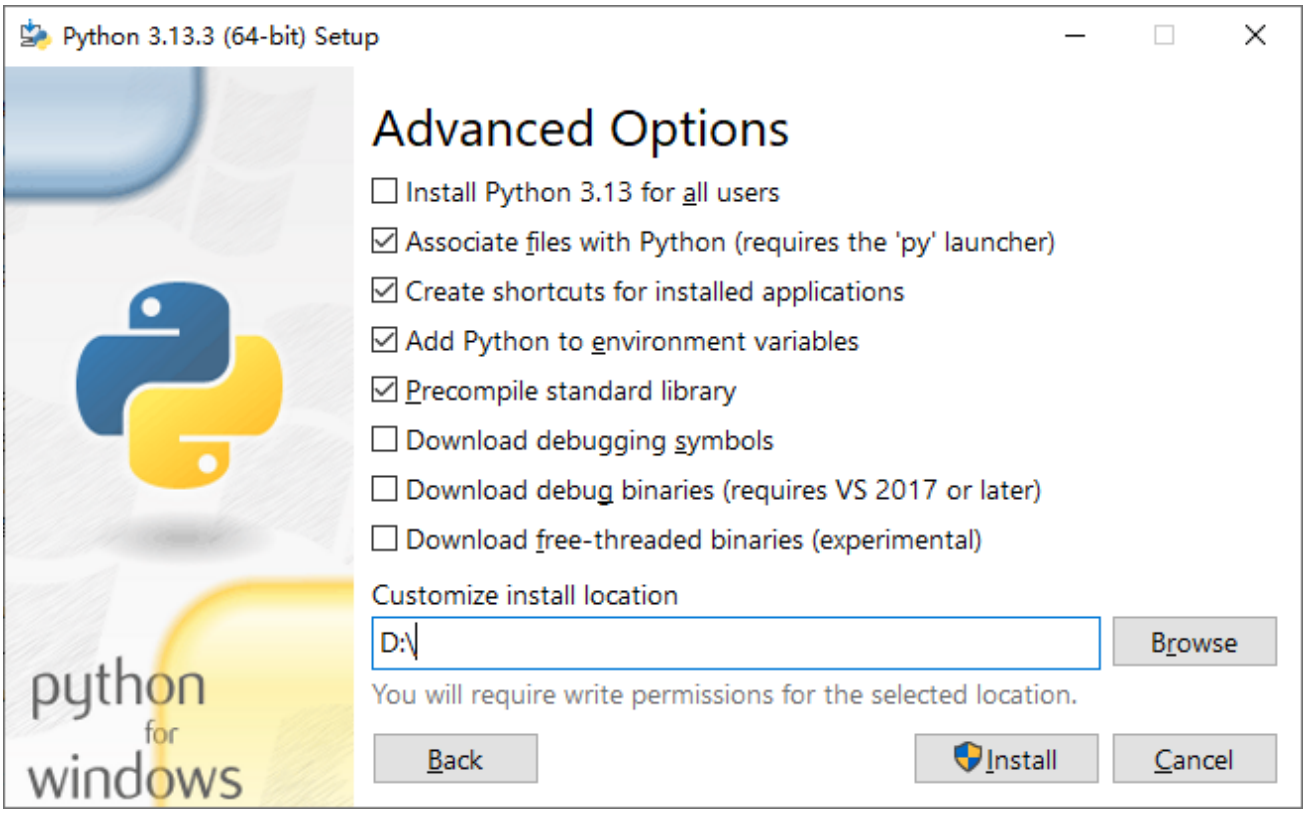
重要提示：必须勾选"将 Python 3.13 添加到 PATH"选项



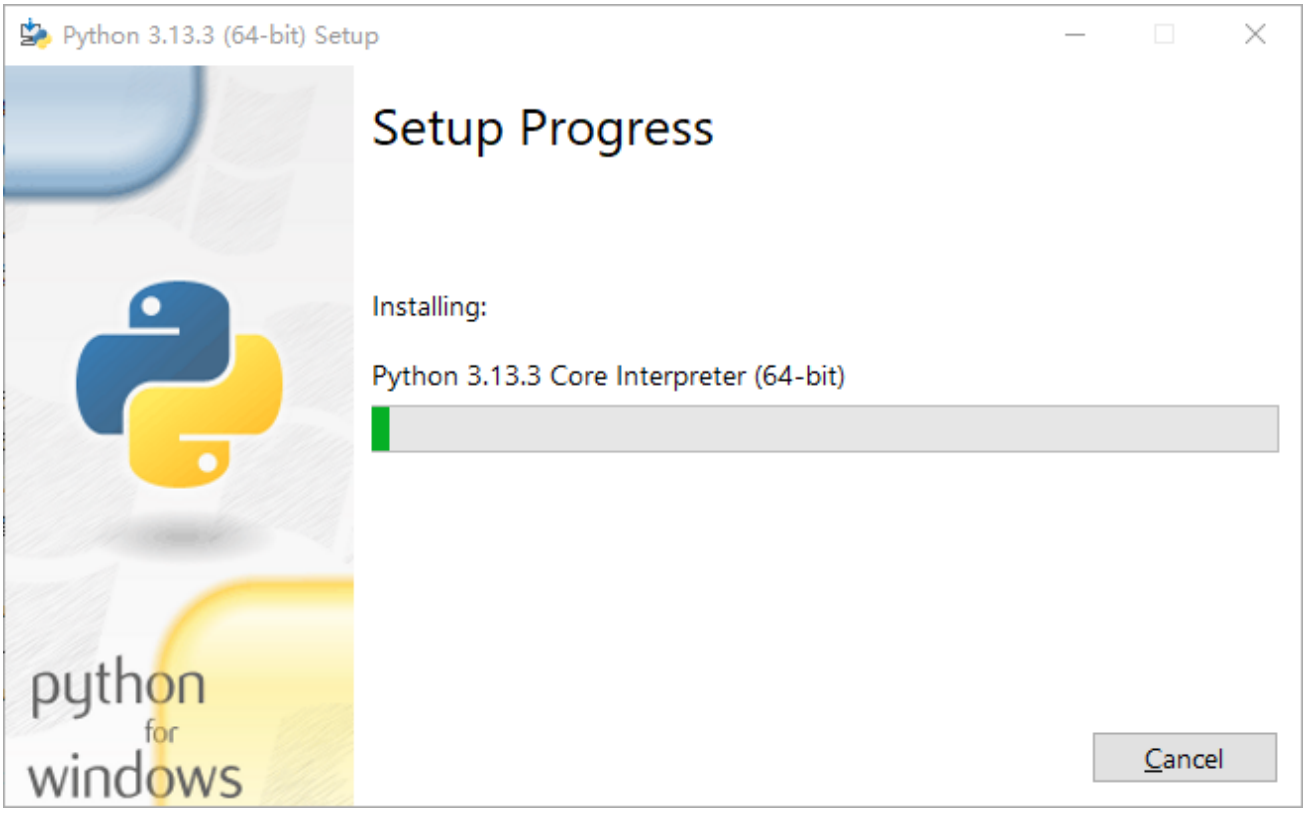
勾选所有选项后点击"Next"



此处可指定 Python 安装路径（建议安装至 D 盘）。新手用户可直接使用默认路径。



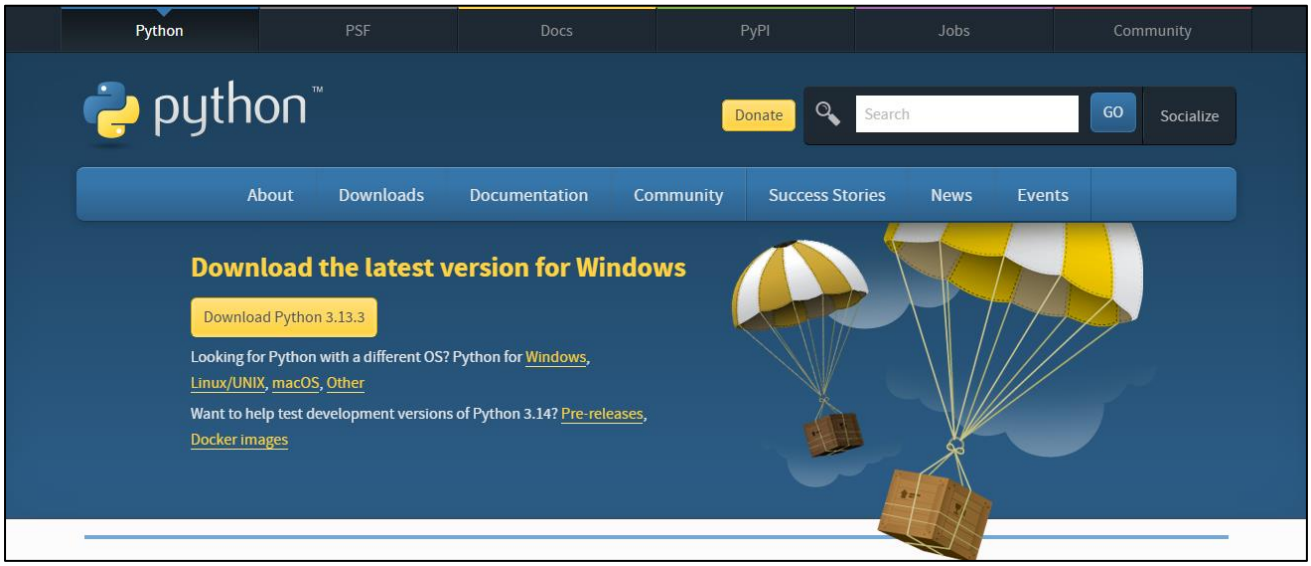
请等待安装完成。



安装已完成。

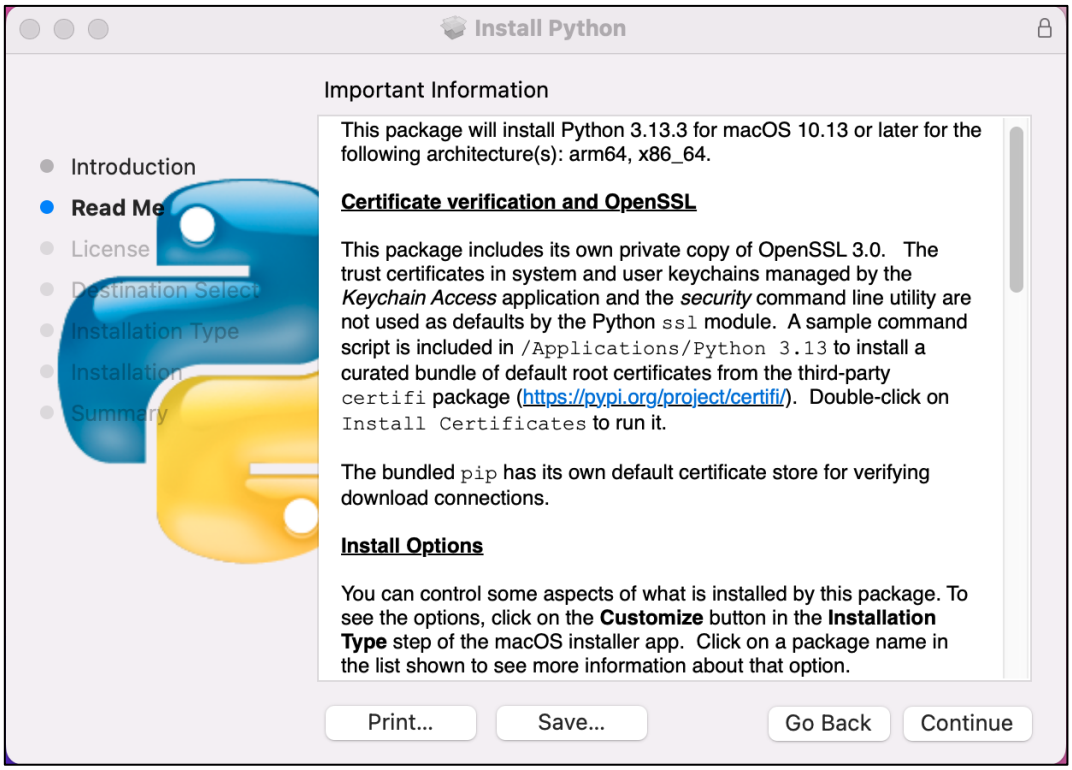
Mac

下载安装包，链接：https://www.python.org/downloads/
点击下载 Python 3.13.3

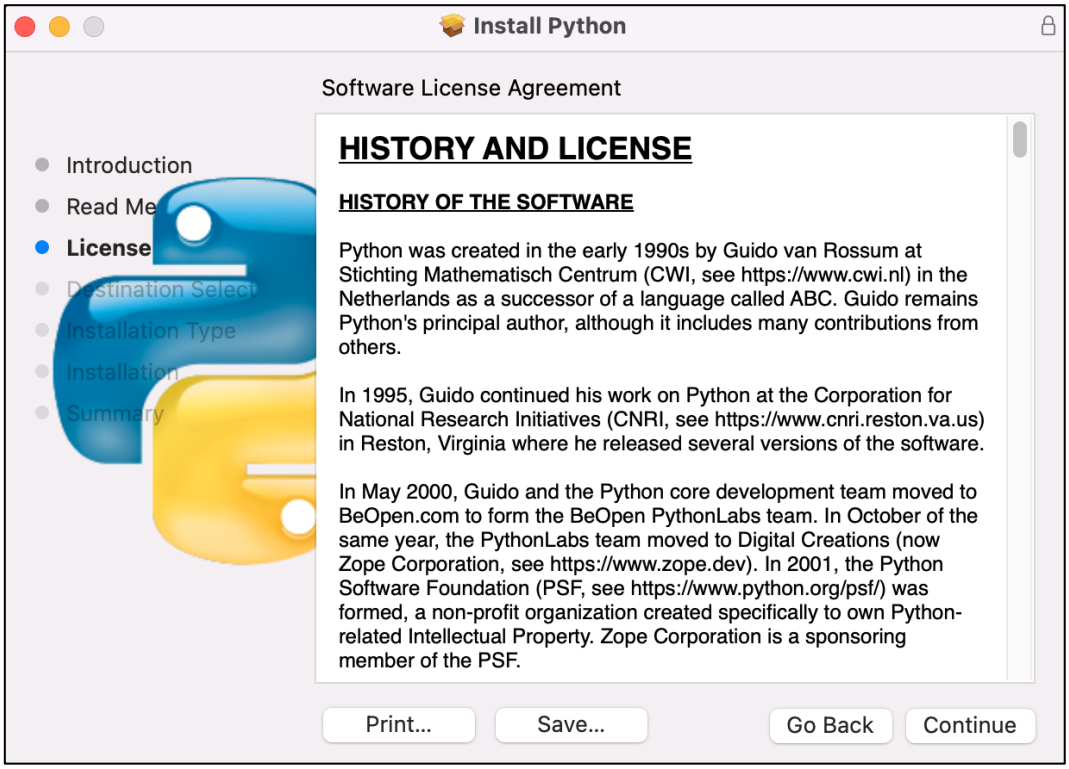


运行下载的安装包，点击"继续"

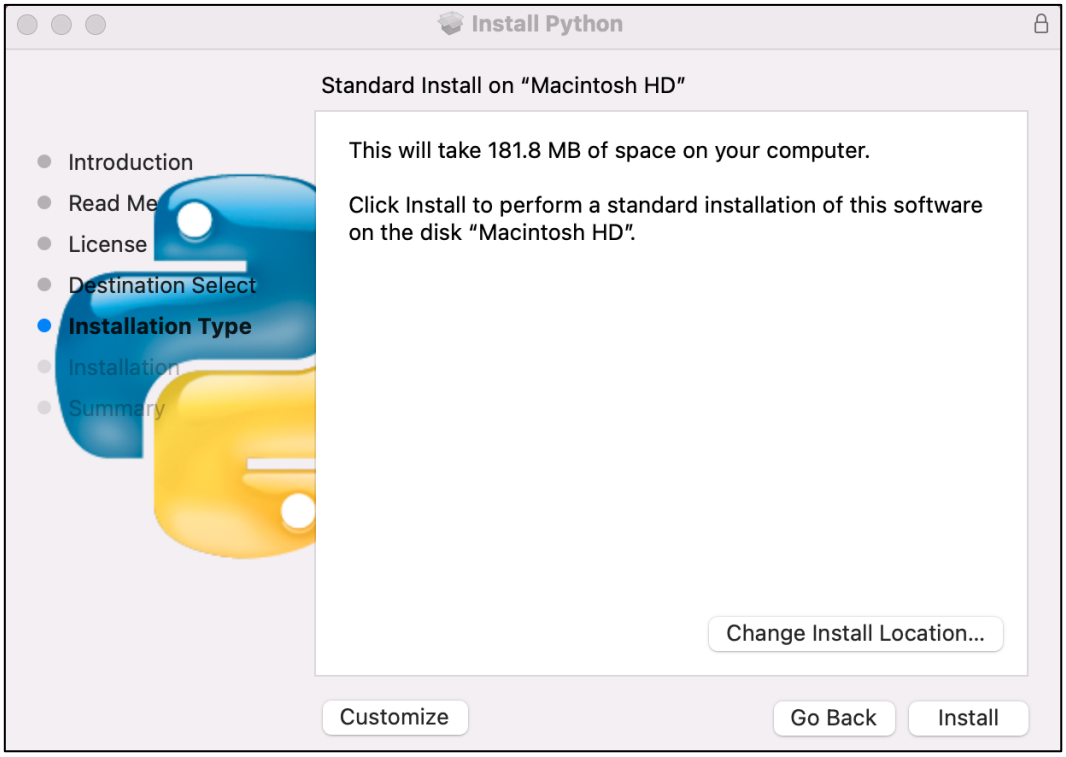
点击 Continue



点击 Continue



点击 Install。若系统提示输入密码，请验证后继续安装。



安装成功。

Linux

检查 Python3 是否已安装

```
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:~$
```

若尚未安装，请执行以下命令进行安装（默认将安装最新版本）

```
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
```

将 python 关联至 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
```

安装 python3.13-venv 虚拟环境

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

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

安装 python3-pip

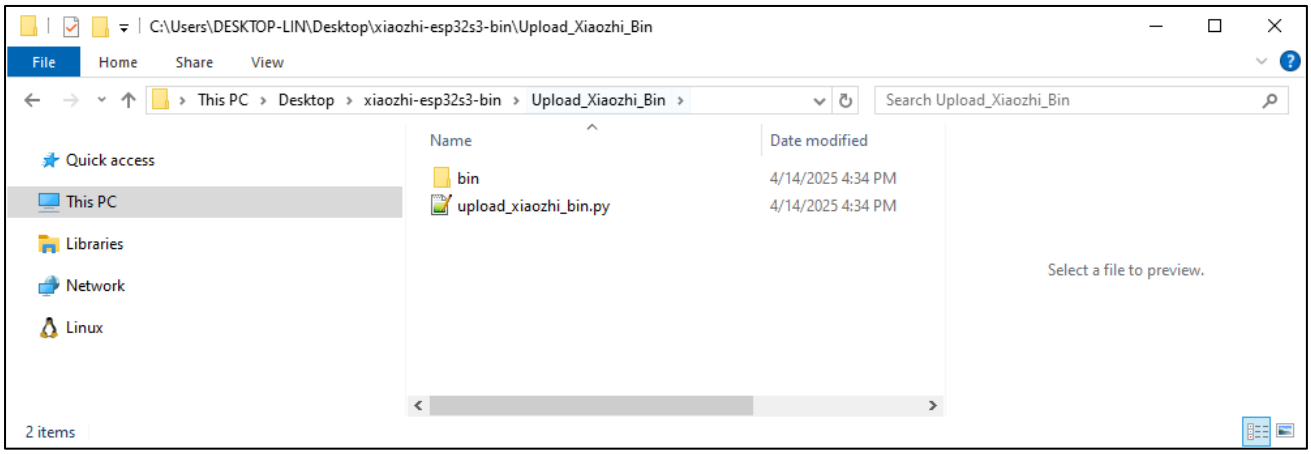
```
sudo apt install python3-pip
```

```
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:~$
```

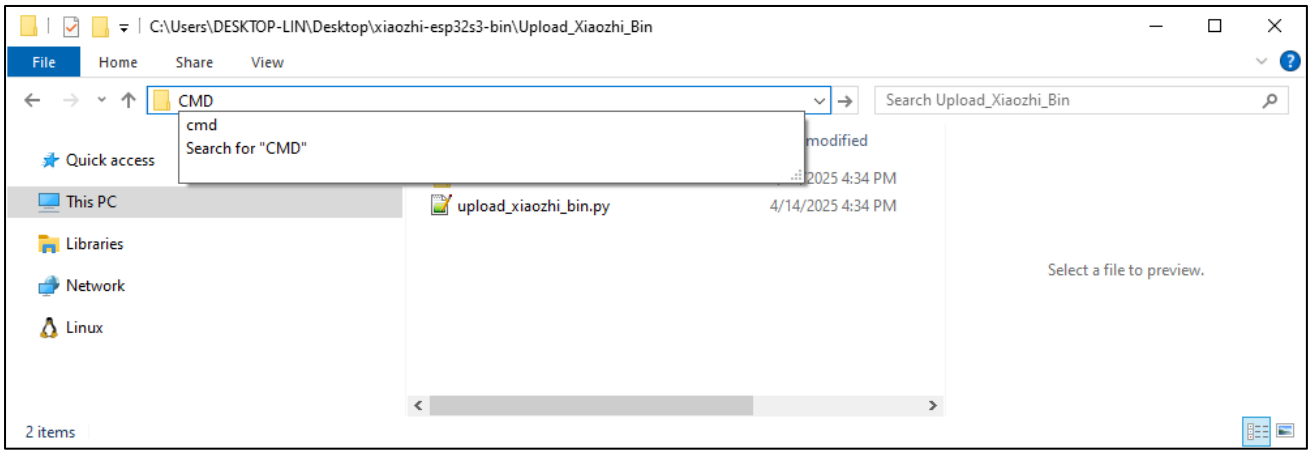
固件烧录

Windows

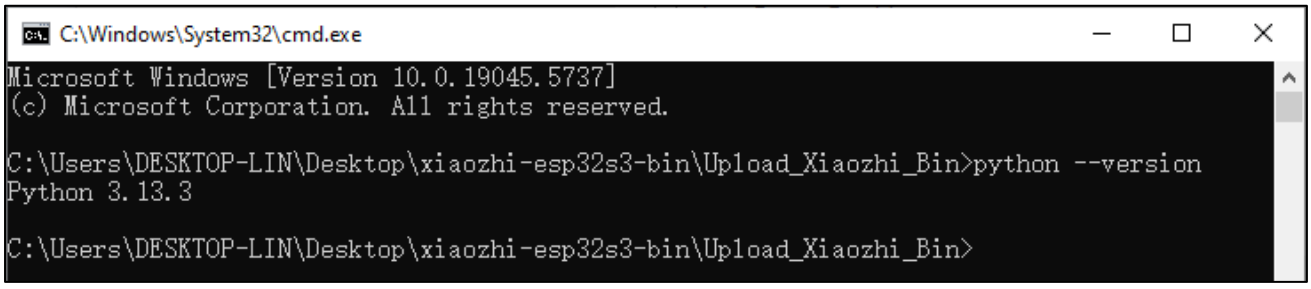
进入 Upload_Xiaozhi_Bin 目录



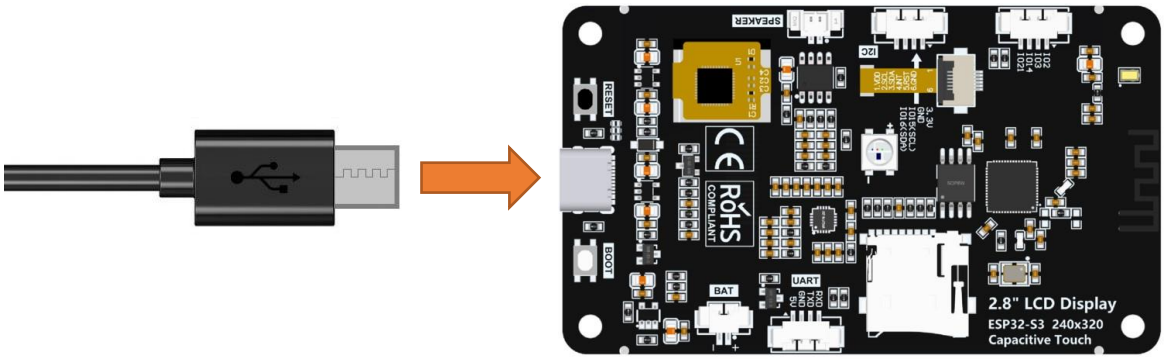
在文件地址栏输入"CMD"并回车



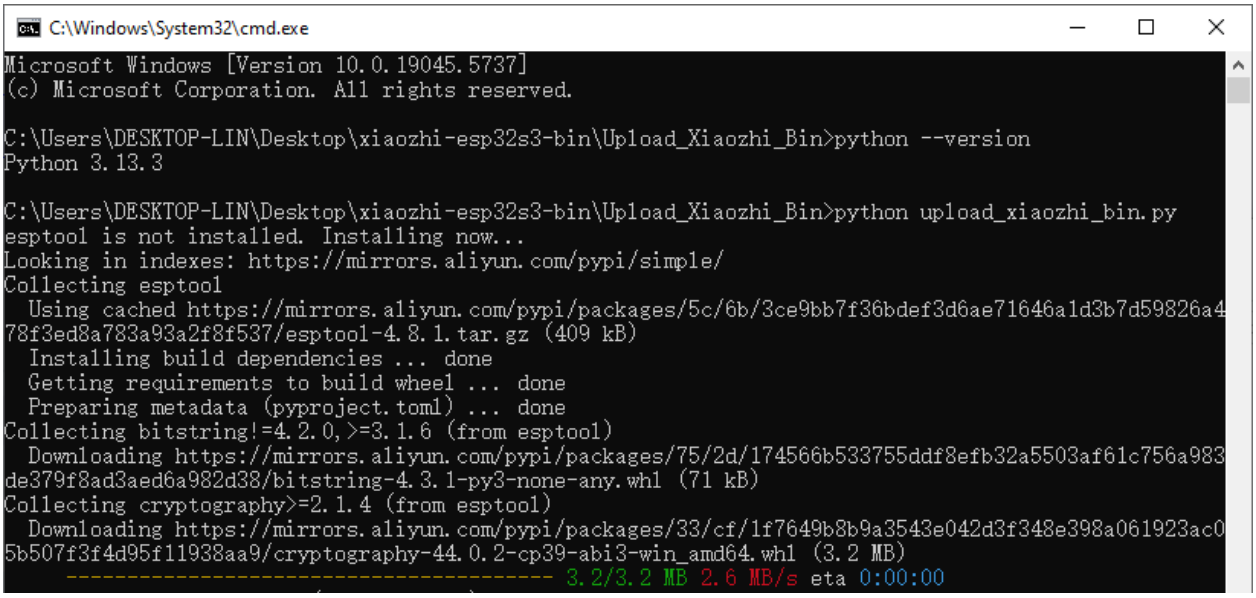
输入"python --version"检查 Python 是否安装。若未显示版本信息，则说明安装异常，请重新安装。



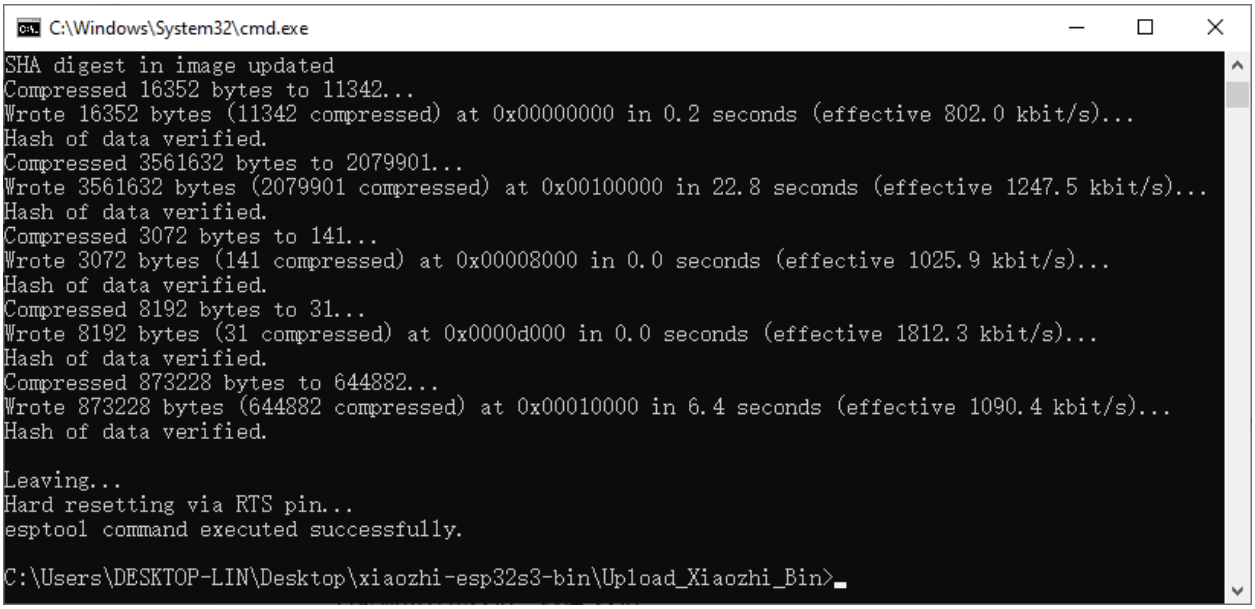
使用 USB 数据线将 ESP32-S3 连接至电脑



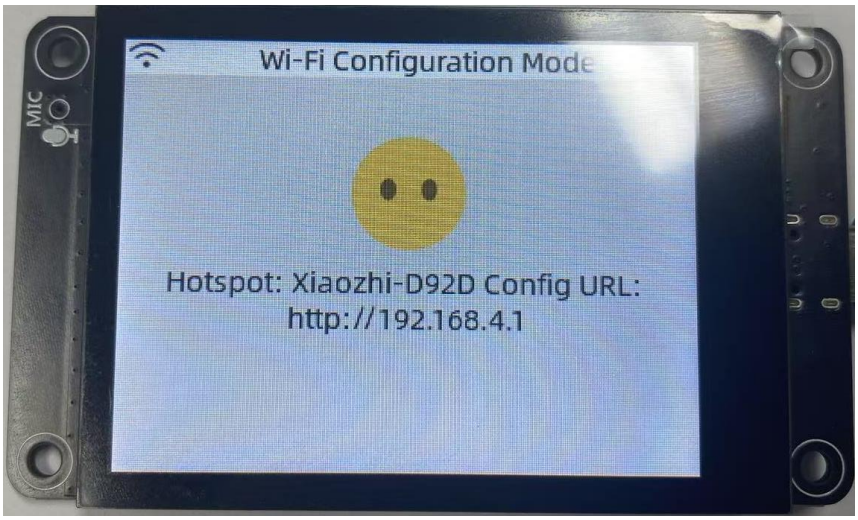
输入“python upload_xiaozhi_bin.py”并按回车键。
如果您的电脑未安装 esptool 或其必需的依赖项，它们将会自动安装。



随后，程序将调用 esptool，将 bin 文件夹中的文件烧录至 ESP32-S3。



您将在 ESP32-S3 开发板上看到以下信息显示。



Mac

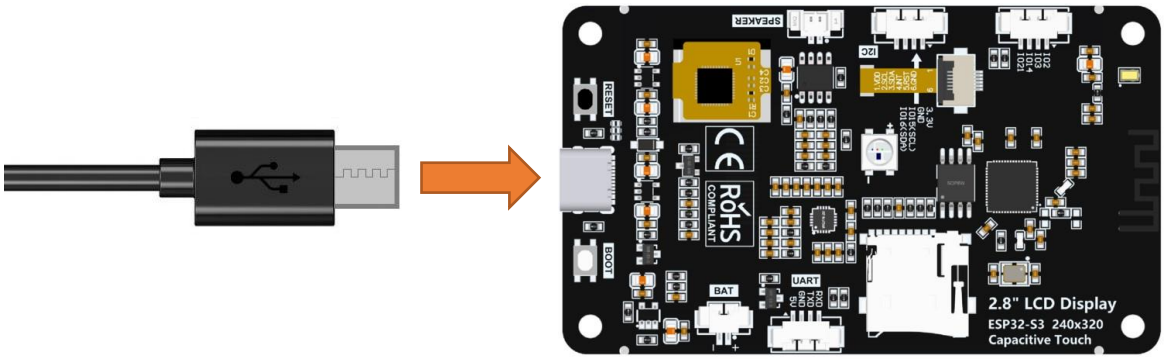
进入 Upload_Xiaozhi_Bin 文件夹。

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

输入 python --version 以检查是否已安装 Python。若未显示版本信息，则说明 Python 未正确安装，请重新安装。

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

使用 USB 数据线将 ESP32-S3 开发板连接至电脑



输入 python upload_xiaozhi_bin.py 并按回车键执行

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

随后系统将自动调用 esptool 工具，将 bin 目录下的固件文件烧录至 ESP32-S3 开发板

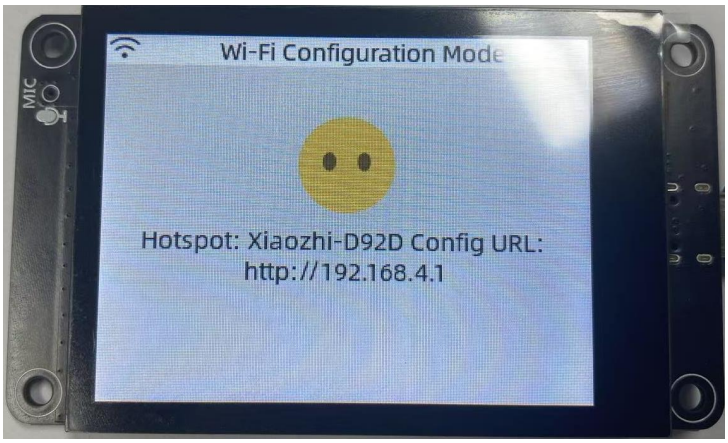
```
Upload_Xiaozhi_Bin — Python ◂ Python upload_xiaozhi_bin.py — 91×24
Serial port /dev/cu.wchusbserial5A4E1051341
Connecting....
Chip is ESP32-S3 (QFN56) (revision v0.2)
Features: WiFi, BLE, Embedded PSRAM 8MB (AP_3v3)
Crystal is 40MHz
MAC: 30:ed:a0:20:bd:9c
Uploading stub...
Running stub...
Stub running...
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...
```

此时 ESP32-S3 开发板将显示以下运行日志:

```
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 % ¢
```

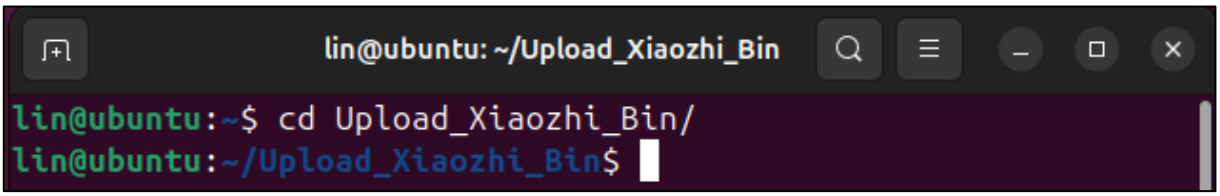
您将在 ESP32-S3 开发板上看到以下信息显示。



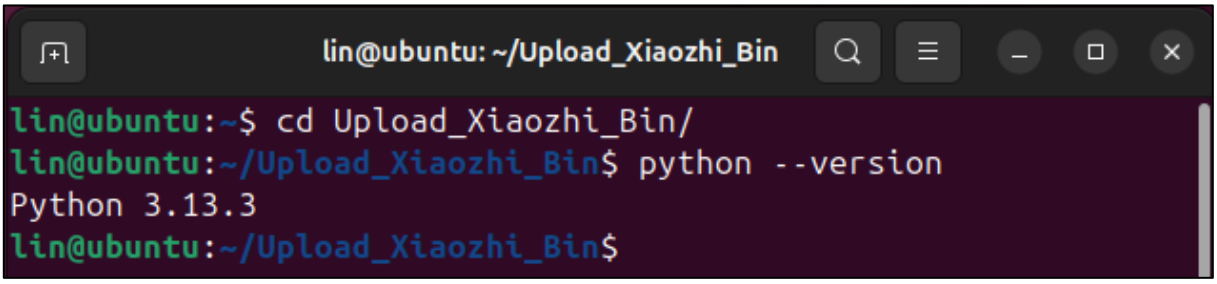
Linux

进入 Upload_Xiaozhi_Bin 目录

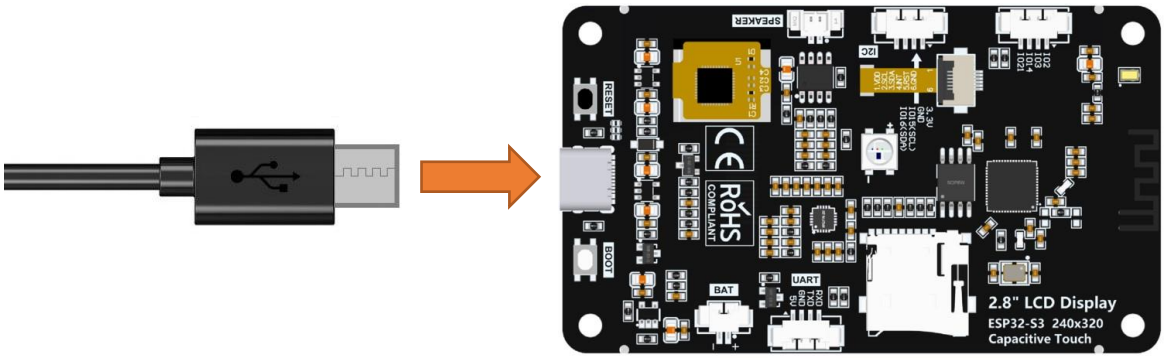
```
cd Upload_Xiaozhi_Bin
```



输入 **python --version** 检查 Python 环境是否已安装。若未显示版本信息，则表明 Python 未正确安装，请重新安装。

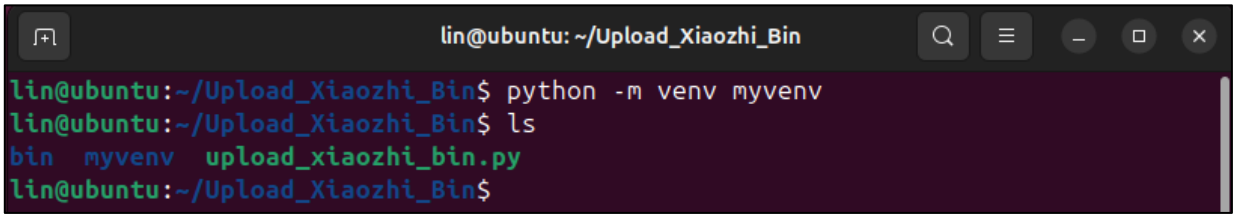


使用 USB 数据线将 ESP32-S3 开发板连接至电脑，请确保插入正确的 Type-C 接口（切勿接错端口）。



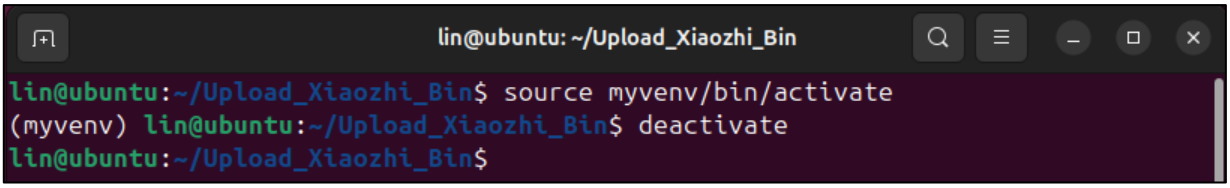
创建名为 myvenv 的虚拟环境

```
python -m venv myvenv
```

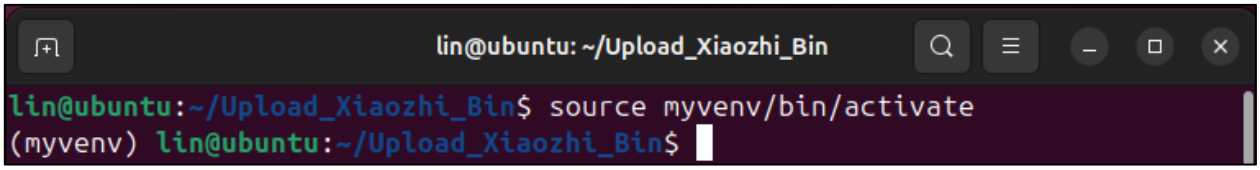


可通过以下命令激活或退出虚拟环境

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



激活虚拟环境



执行以下命令检测 ESP32-S3 端口号

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

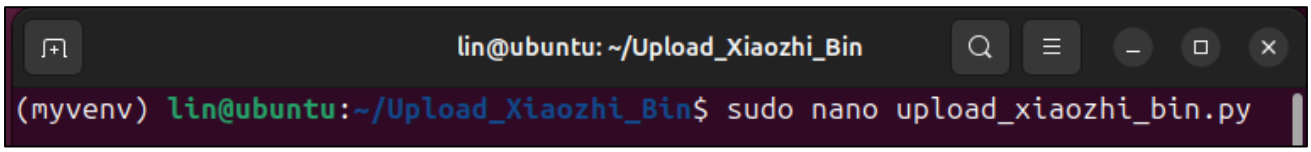
当 ESP32-S3 未连接电脑时，端口显示如下

当连接 ESP32-S3 后，系统将生成新端口

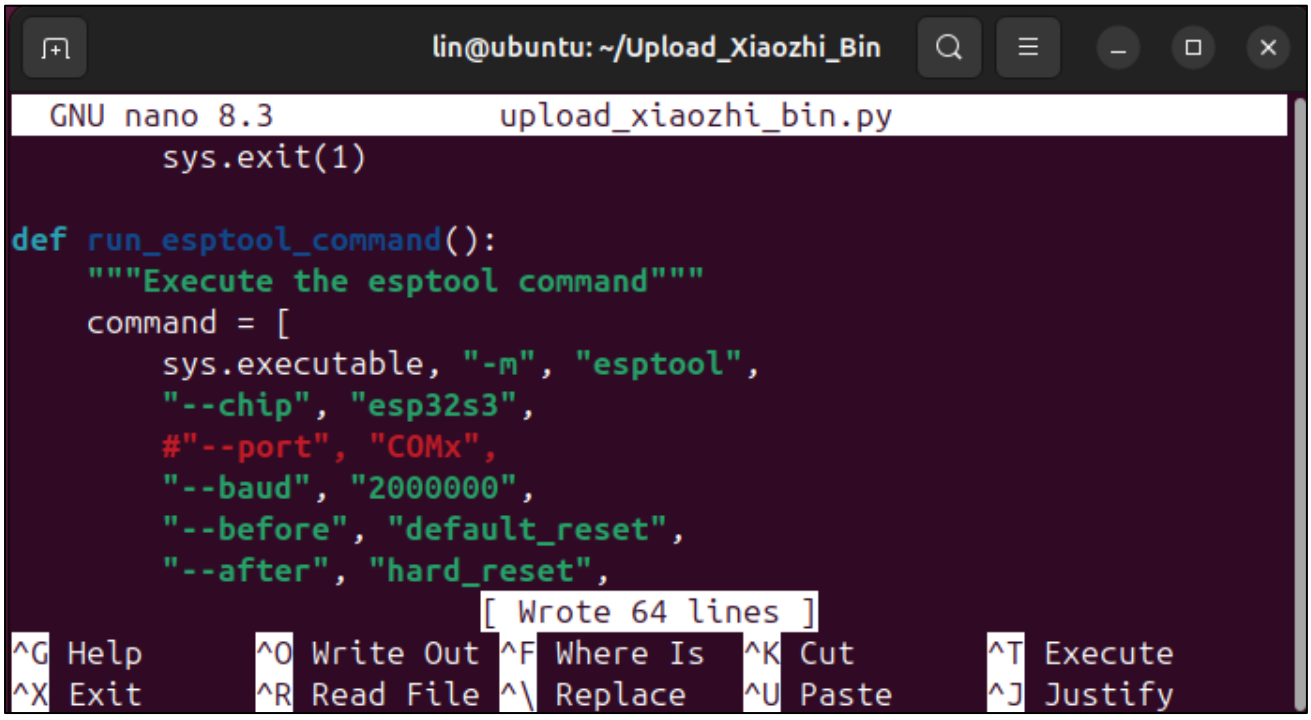
新生成的端口即为 ESP32-S3 所用，请记录该端口号。运行 Python 文件前，需先修改端口配置。

执行以下命令打开 Python 文件：

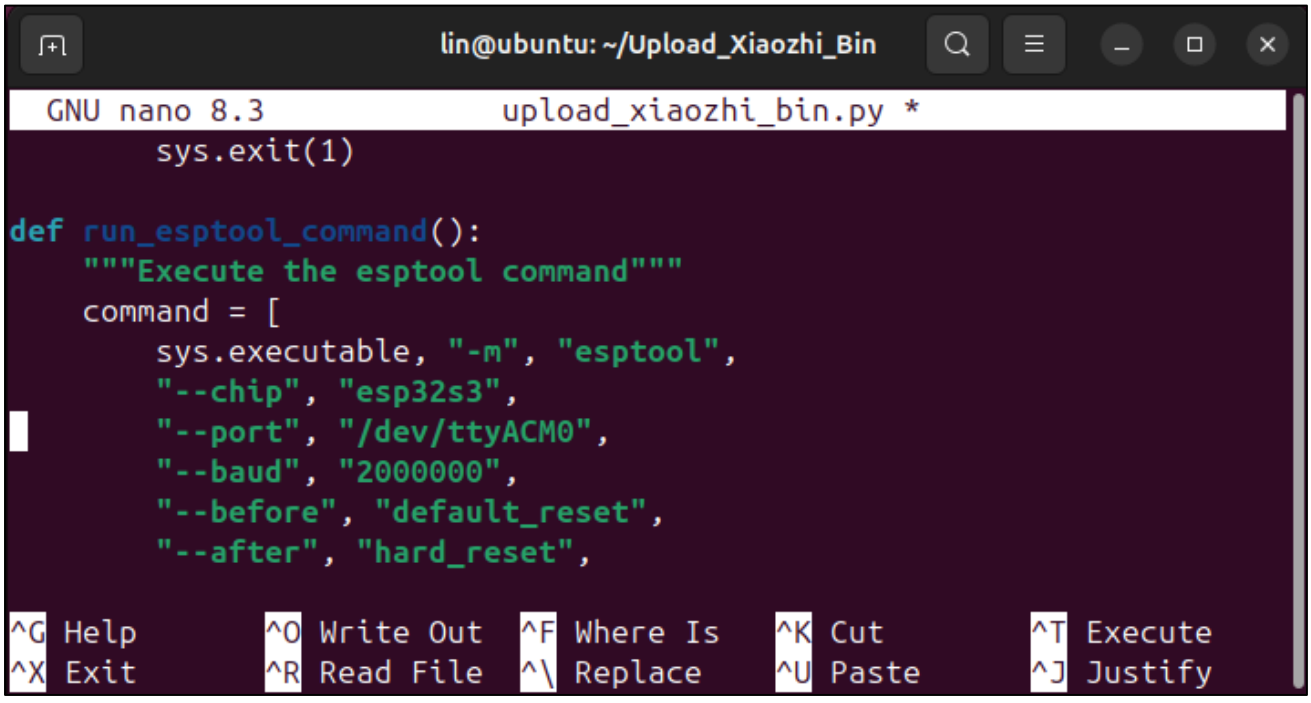
```
sudo nano upload_xiaozhi_bin.py
```



在文本编辑器中定位 '--port' 参数行，将其中的 'COMx' 替换为 Linux 系统中 ESP32-S3 分配的实际端口号



修改后的配置如下所示：



按下 Ctrl+O 保存修改，Ctrl+X 退出文件

执行 Python 文件

```
python upload_xiaozhi_bin.py
```

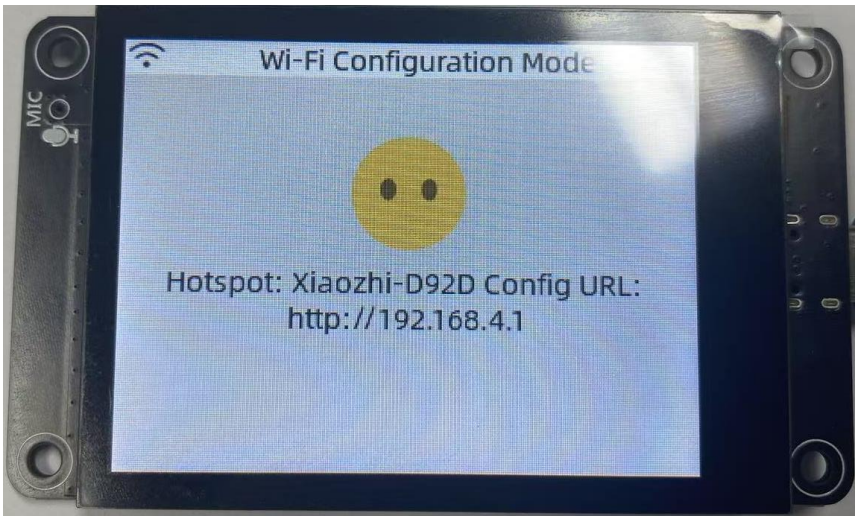
```
lin@ubuntu: ~/Upload_Xiaozhi_Bin
(myvenv) lin@ubuntu:~/Upload_Xiaozhi_Bin$ python upload_xiaozhi_bin.py
esptool is already installed.
Executing esptool command...
esptool.py v4.8.1
Serial port /dev/ttyACM0
Connecting....
Chip is ESP32-S3 (QFN56) (revision v0.2)
Features: WiFi, BLE, Embedded PSRAM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

固件烧录成功的运行结果如下：

```
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$
```

ESP32-S3 开发板显示如下：



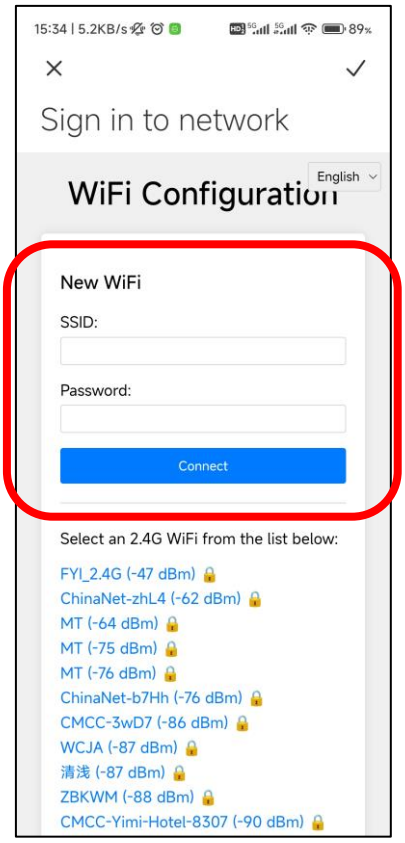
ESP32-S3 网络配置指南

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. 在智能手机上启用 WiFi
2. 寻找名为"Xiaozhi-XXXX"的热点（开放网络，无需密码）
3. 连接该网络以继续

连接 WiFi 后，按屏幕提示点击通知。系统将自动打开手机浏览器并跳转至 http://192.168.4.1



ESP32-S3 的 WiFi 连接设置

输入 WiFi 凭证:

SSID：输入您的 WiFi 网络名称（仅支持 2.4GHz）

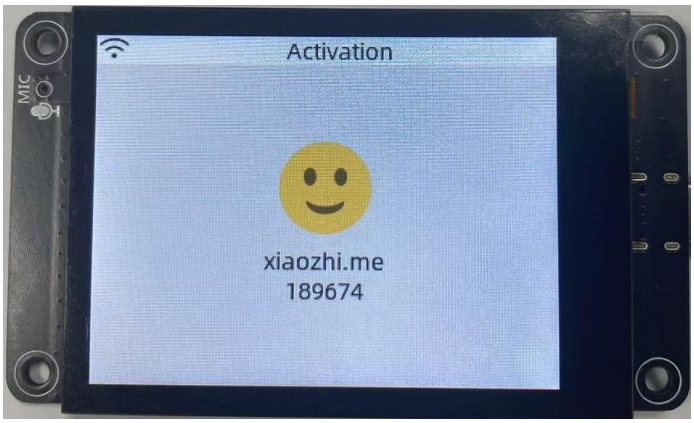
密码：输入 WiFi 密码

点击"连接"继续

重要提示:

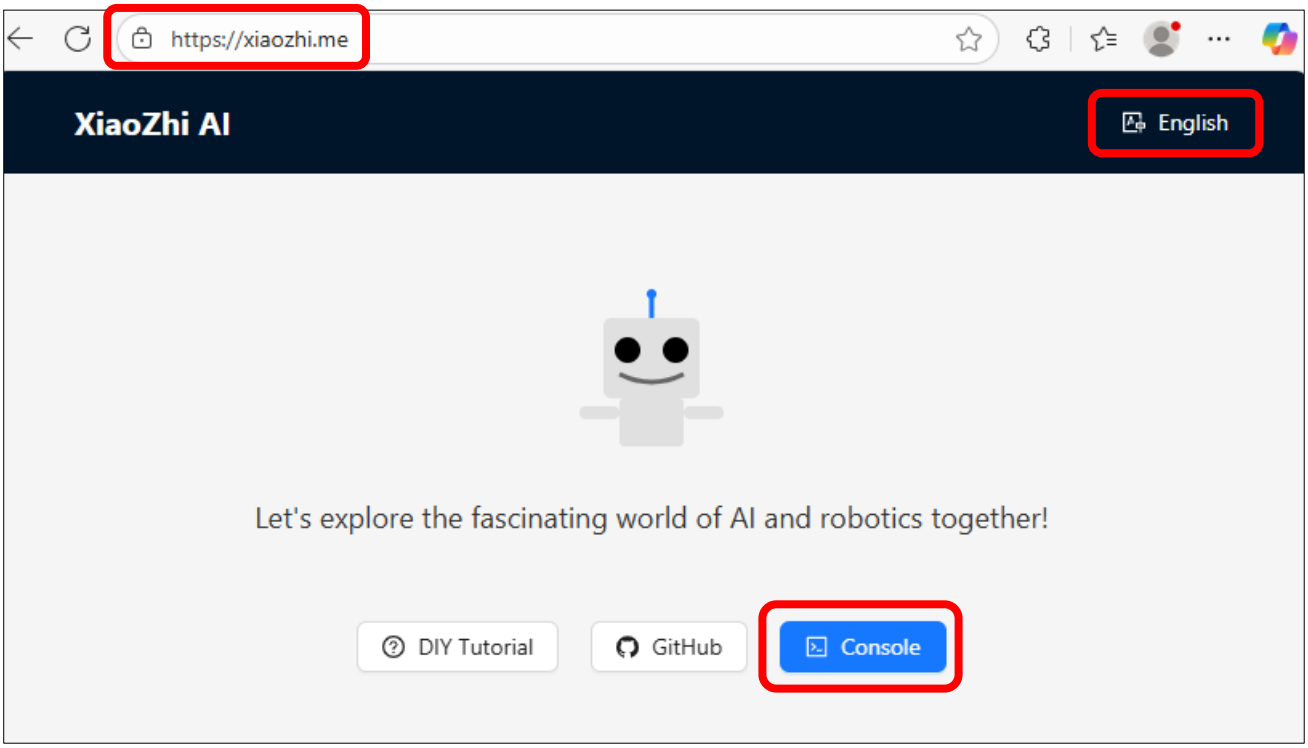
- ESP32-S3 仅支持 2.4GHz WiFi 网络
- 若路由器同时广播 2.4GHz 和 5GHz 信号，请确保设备仅连接 2.4GHz 频段
- 避免使用混合模式（2.4GHz+5GHz 合并）设置，否则可能导致连接失败

当出现以下界面时，表示 ESP32-S3 已成功连接您的 WiFi 网络

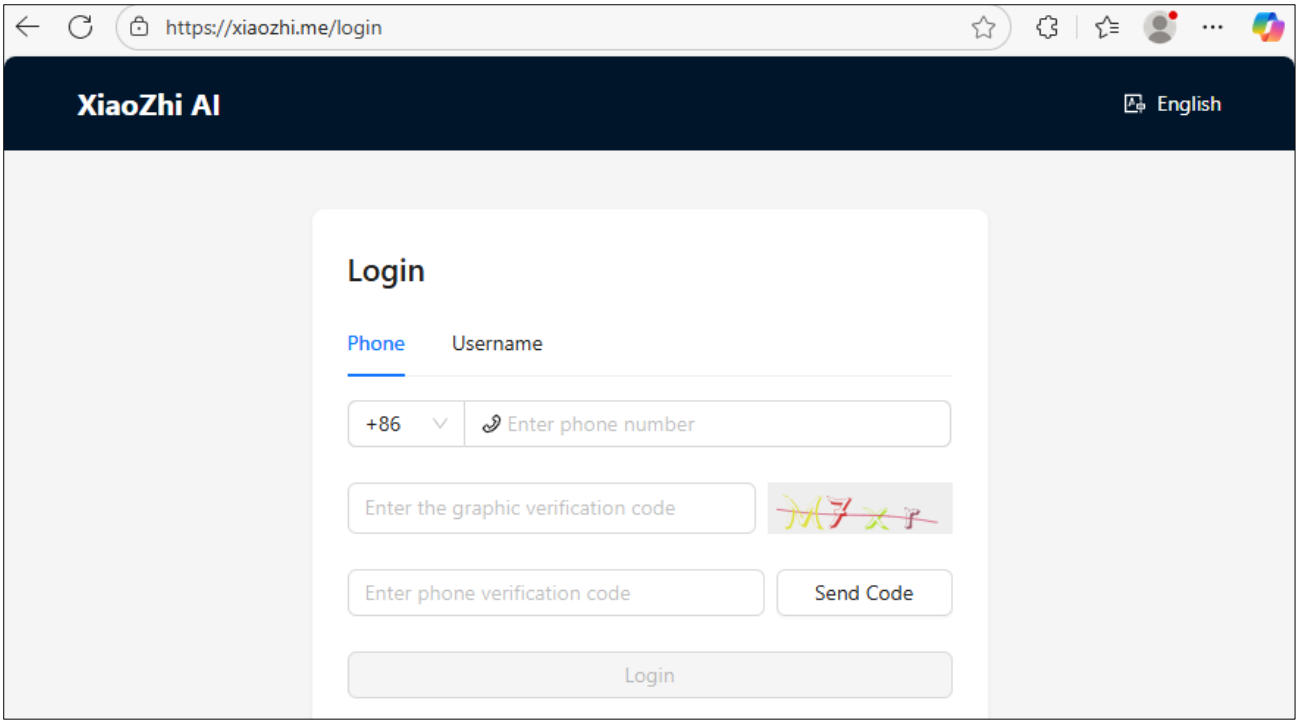


XiaoZhi AI 服务器配置

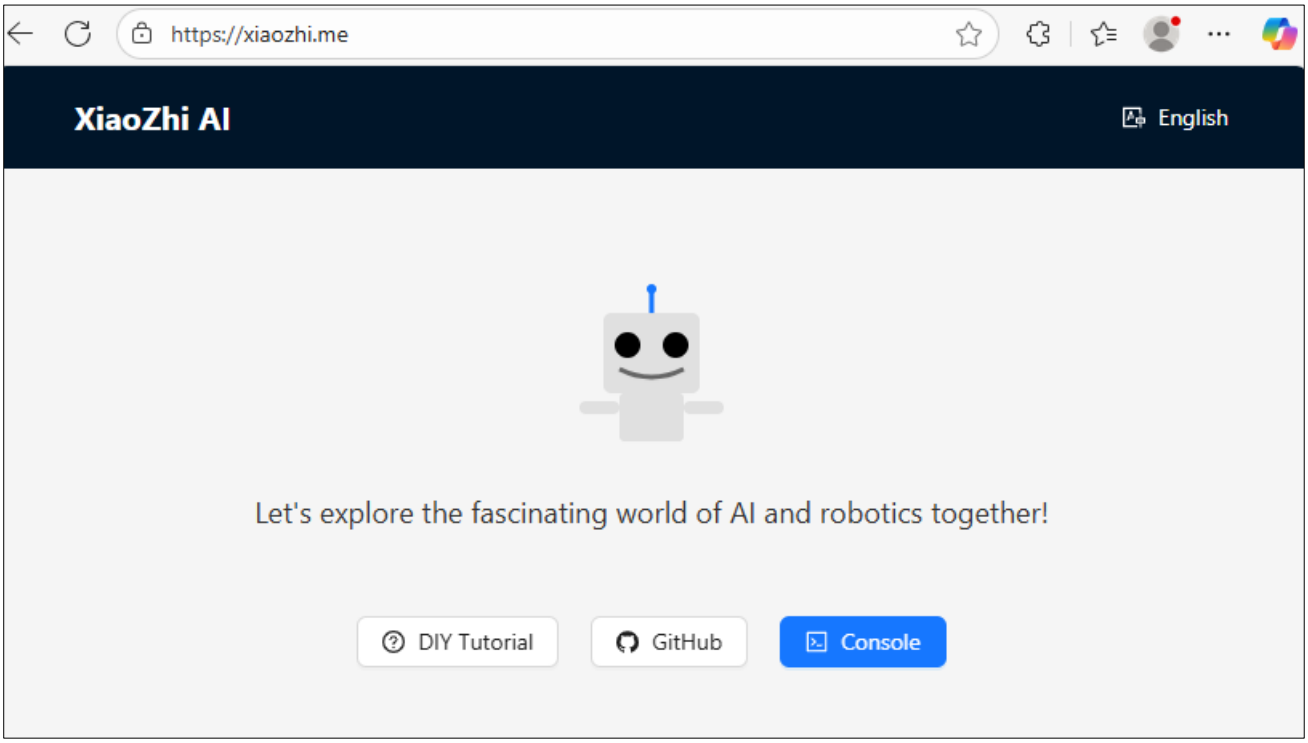
确保您的手机/电脑和 ESP32-S3 连接到同一个路由器 WiFi 网络。
在您的设备上打开浏览器并访问：https://xiaozhi.me/
请注意，由于各国互联网政策不同，部分地区用户在访问网站时可能会遇到困难。具体详情请参考相关国家互联网政策。



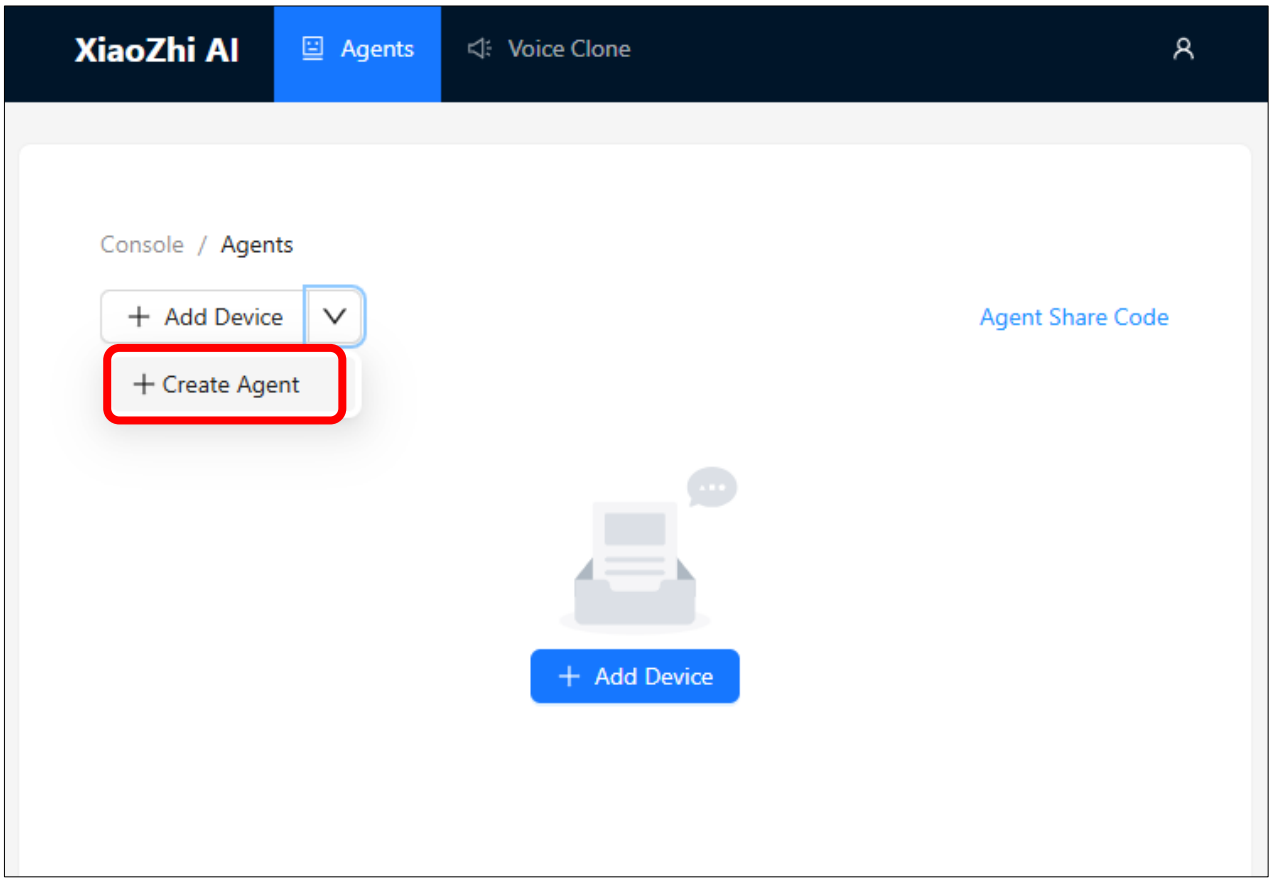
如果您还没有帐户，请注册一个并登录。



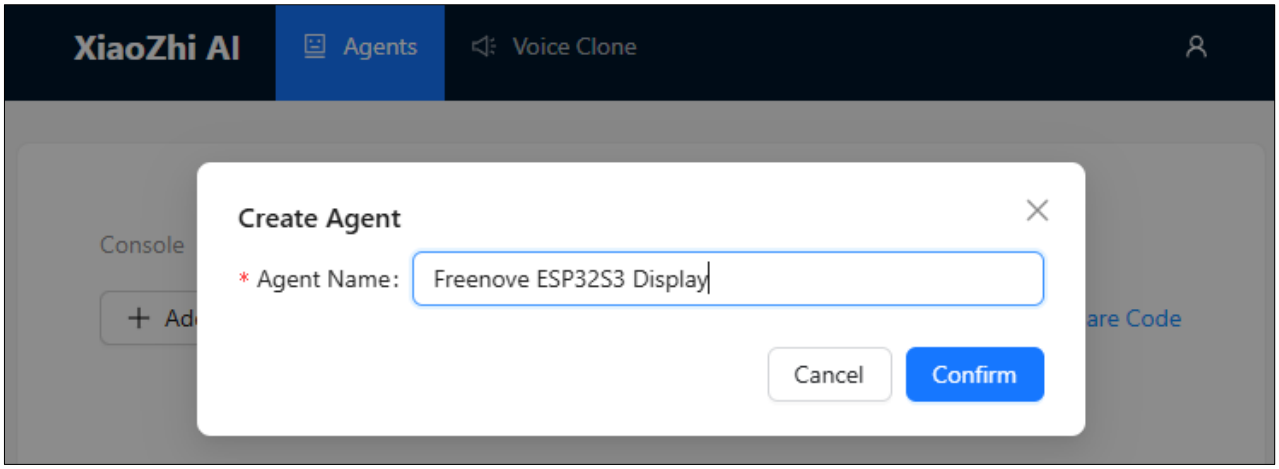
点击“控制台”开始设置您的小智 AI 服务器。



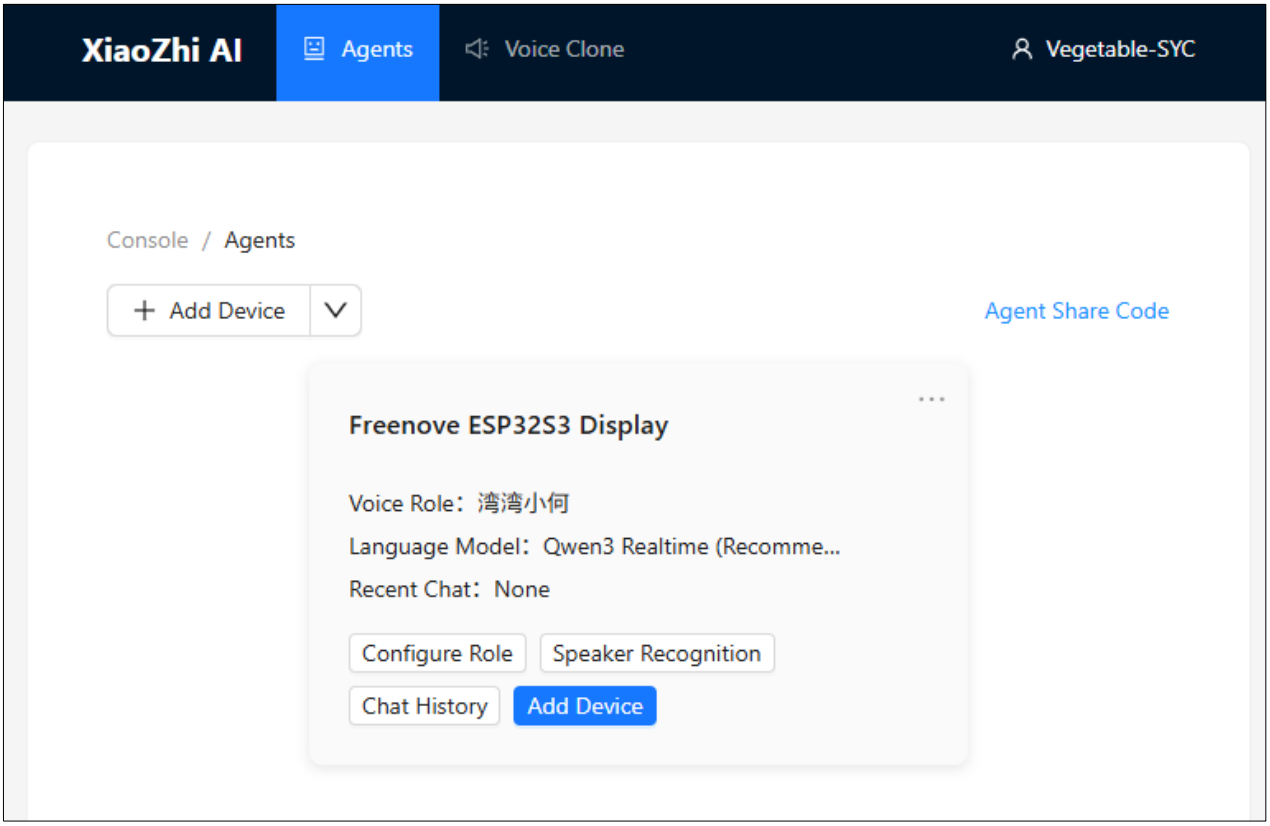
点击“新建智能体”来设置一个新的人工智能助手。



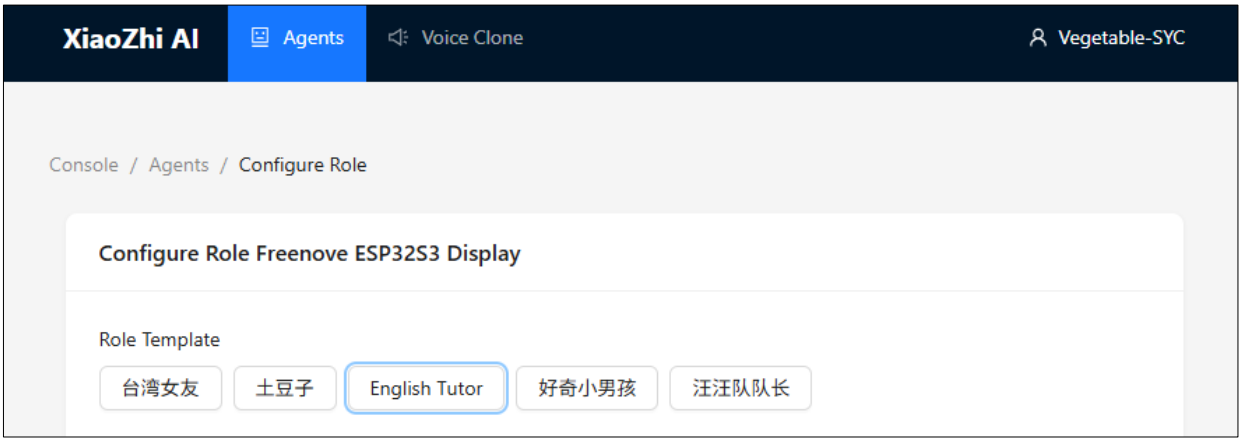
随意命名并点击“确认”。



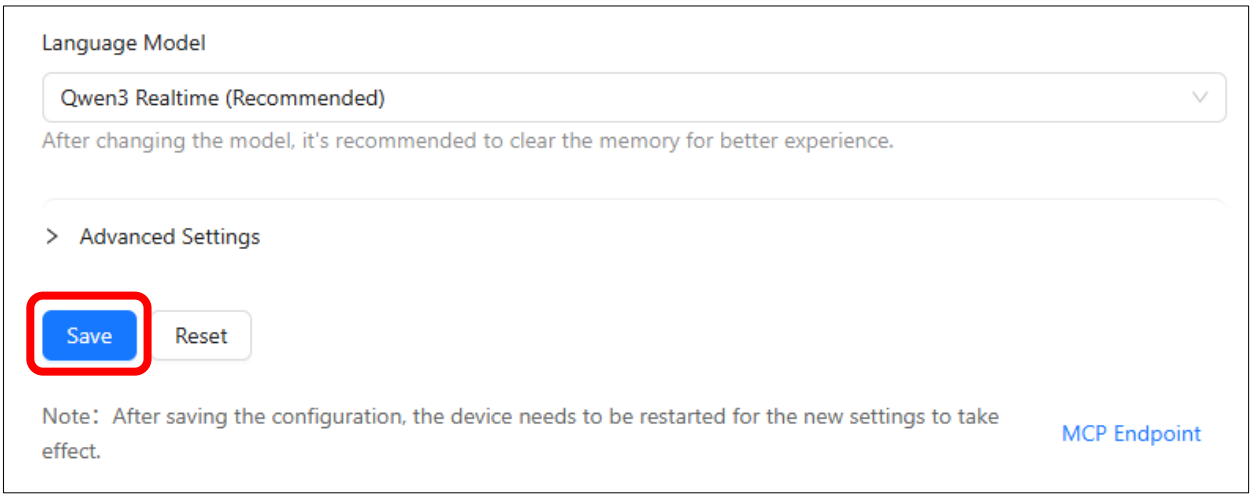
点击“配置角色”来配置您的人工智能助手。



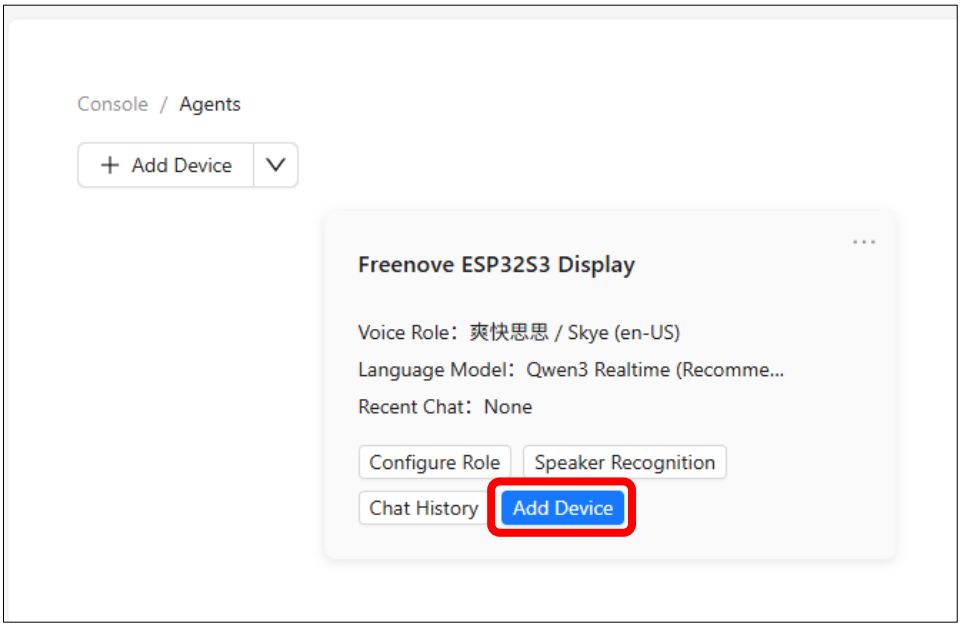
点击“英语家教”（保持所有其他选项不变）。



滚动到页面底部并点击“保存”以确认所有设置。

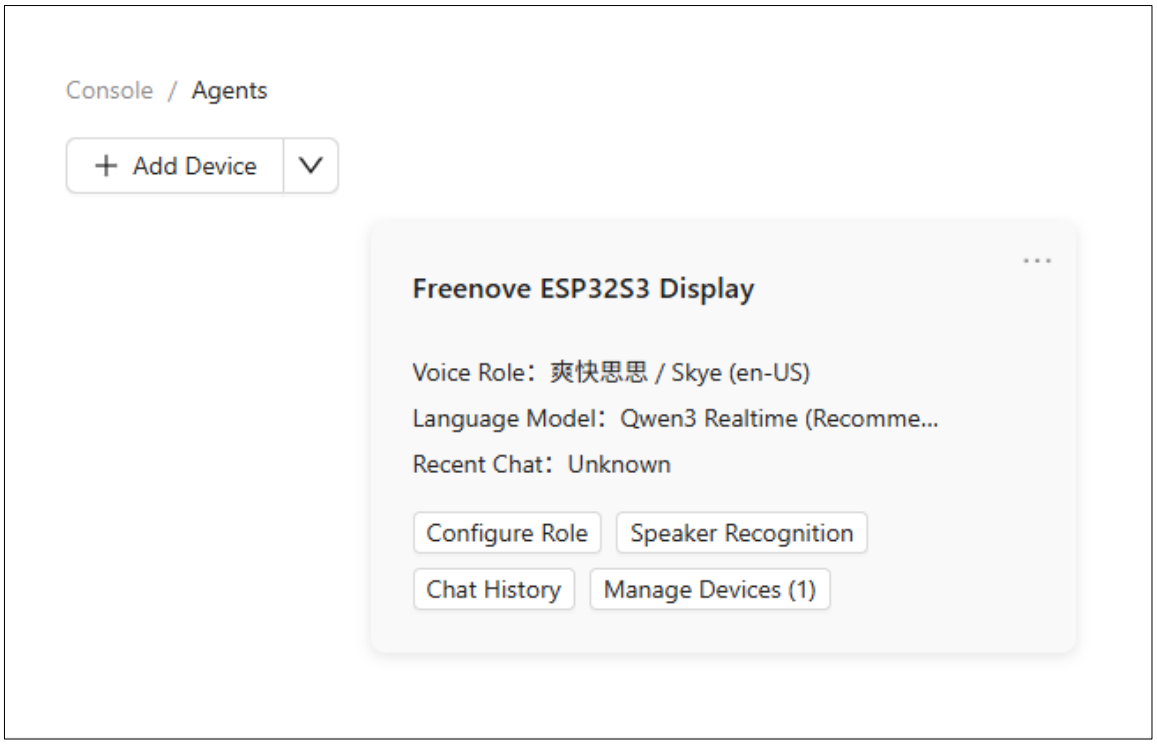


点击“智能体”返回主仪表板并选择“添加设备”以注册新硬件。



在新的弹出窗口中，输入您的 ESP32-S3 上显示的屏幕数字代码。点击“确定”完成配对。

界面现在将如下所示。

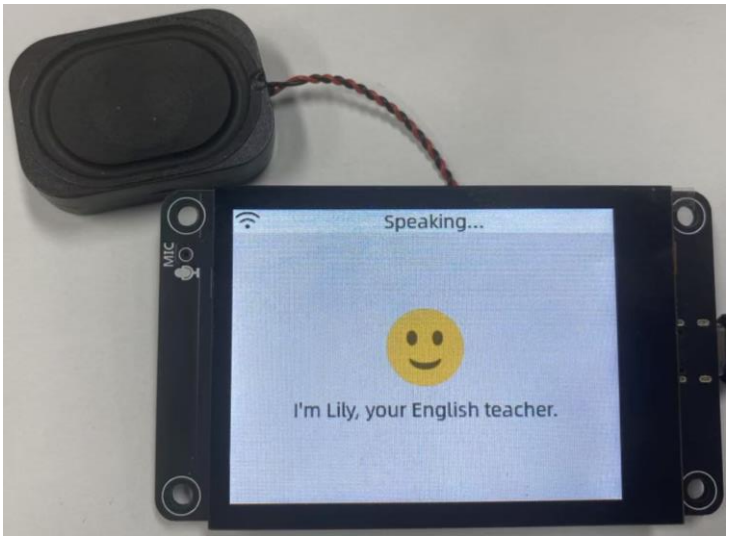


按下 Freenove ESP32 S3 Display 上的 RST 按钮重新启动电路板。

您已成功完成小智 AI 的配置!

连接扬声器

要激活，请对麦克风说“嗨，ESP”；系统现在将响应您的语音命令



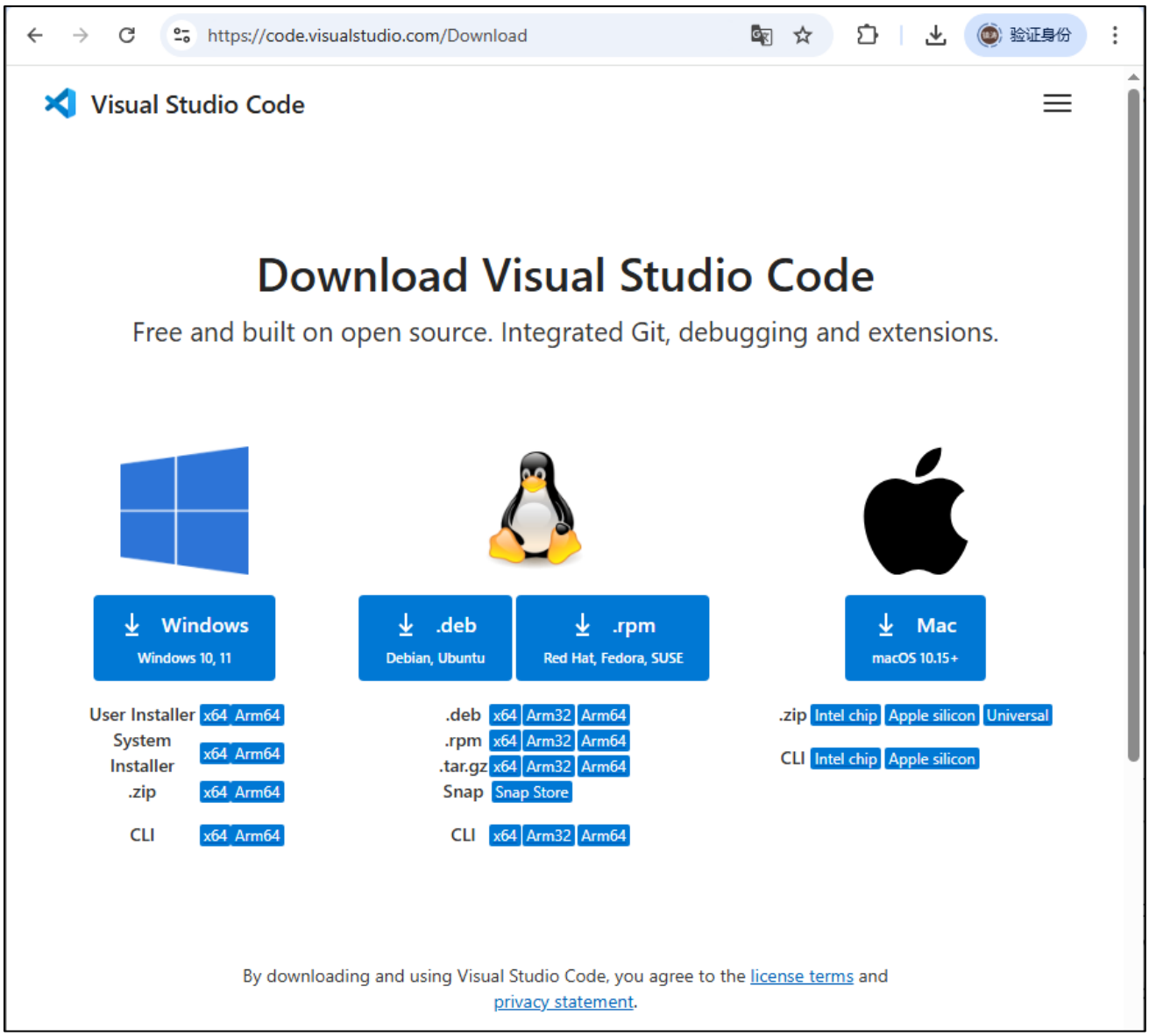
您可以使用中文或英文与它交流。

小智 AI 代码

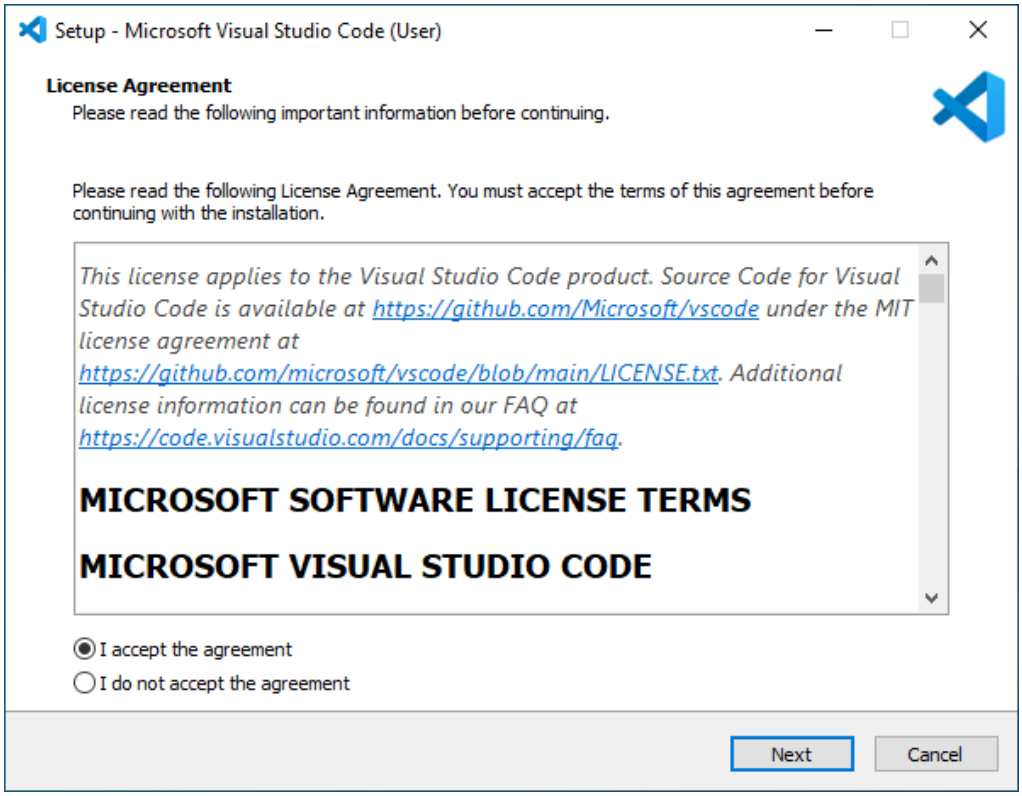
Visual Studio Code

Windows

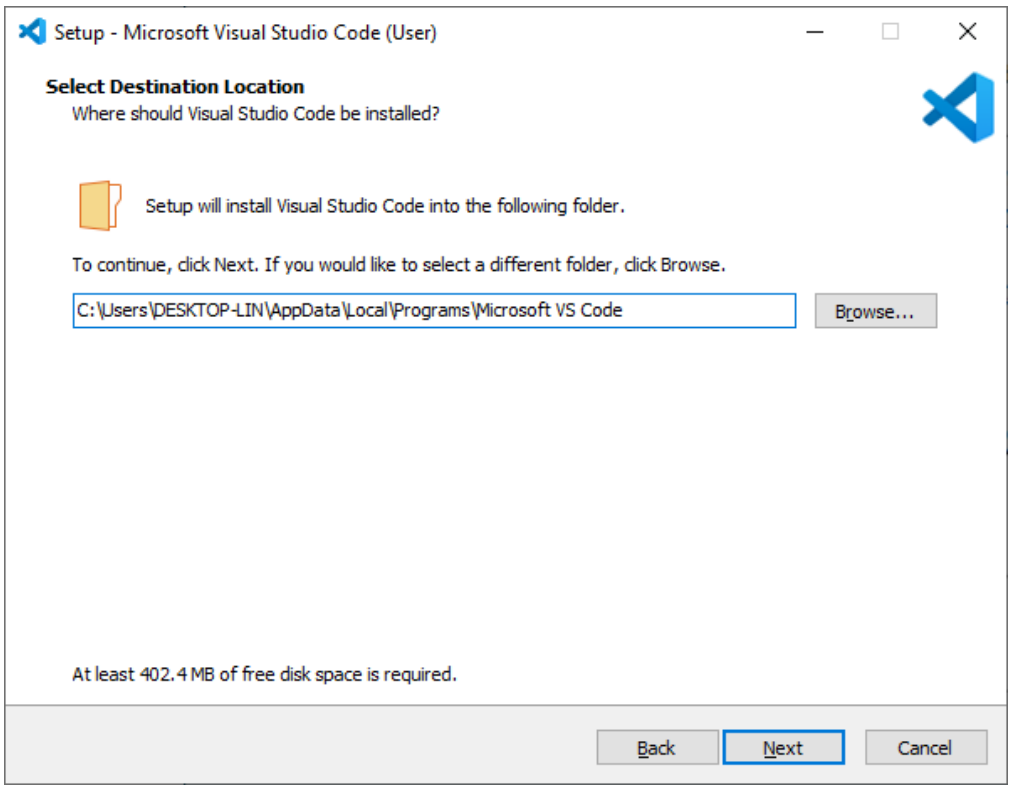
首先，访问 https://code.visualstudio.com/Download 下载 Visual Studio Code。根据您的操作系统选择对应版本，下载后进行安装。



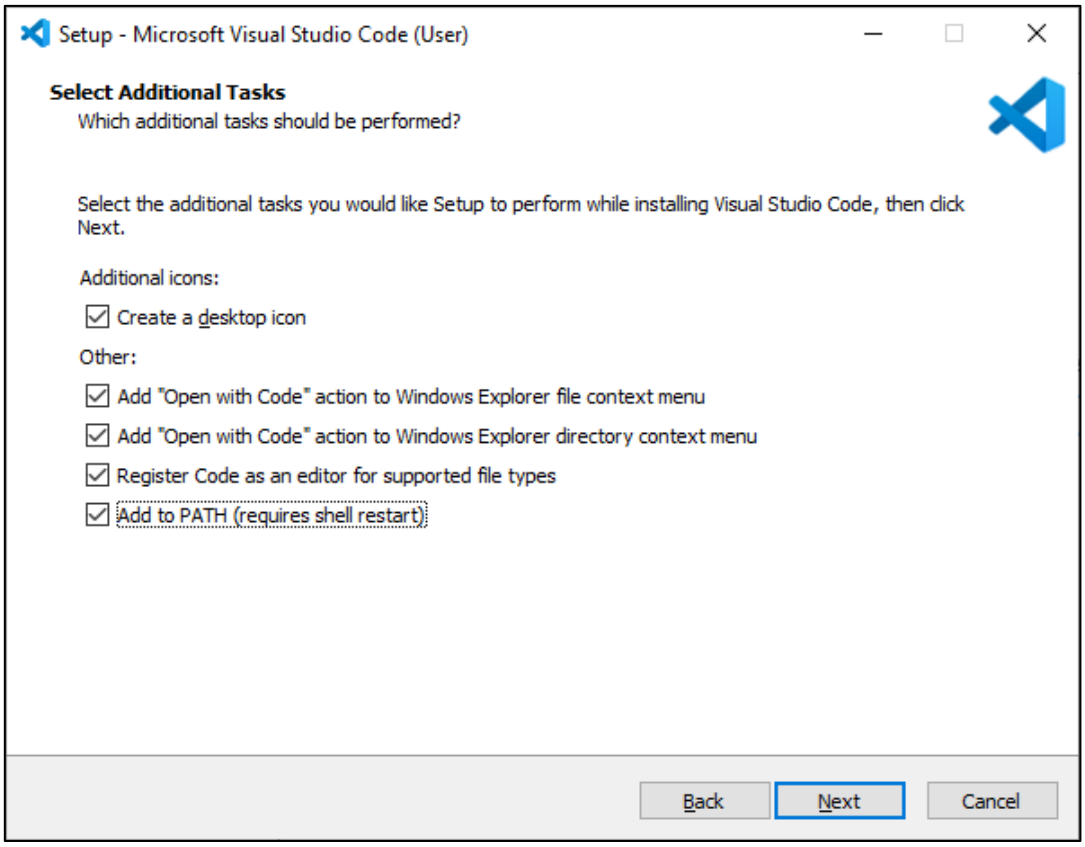
双击下载的 .exe 文件运行安装程序。
勾选“我接受许可协议”选项。
然后点击“下一步”。



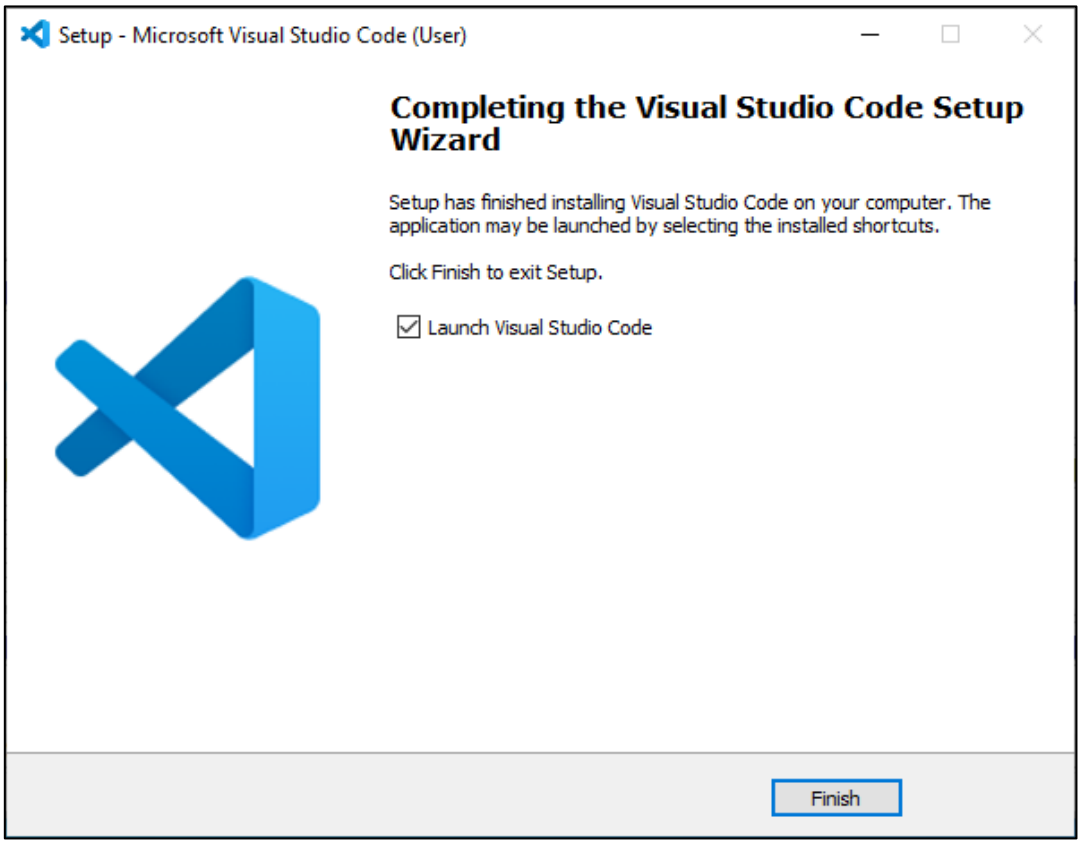
安装路径可保留默认设置，或修改为自定义目录。之后连续点击“下一步”继续。



在此界面，请确认已勾选“添加到 PATH”（如未勾选，请手动启用）。随后连续点击“下一步”完成安装。

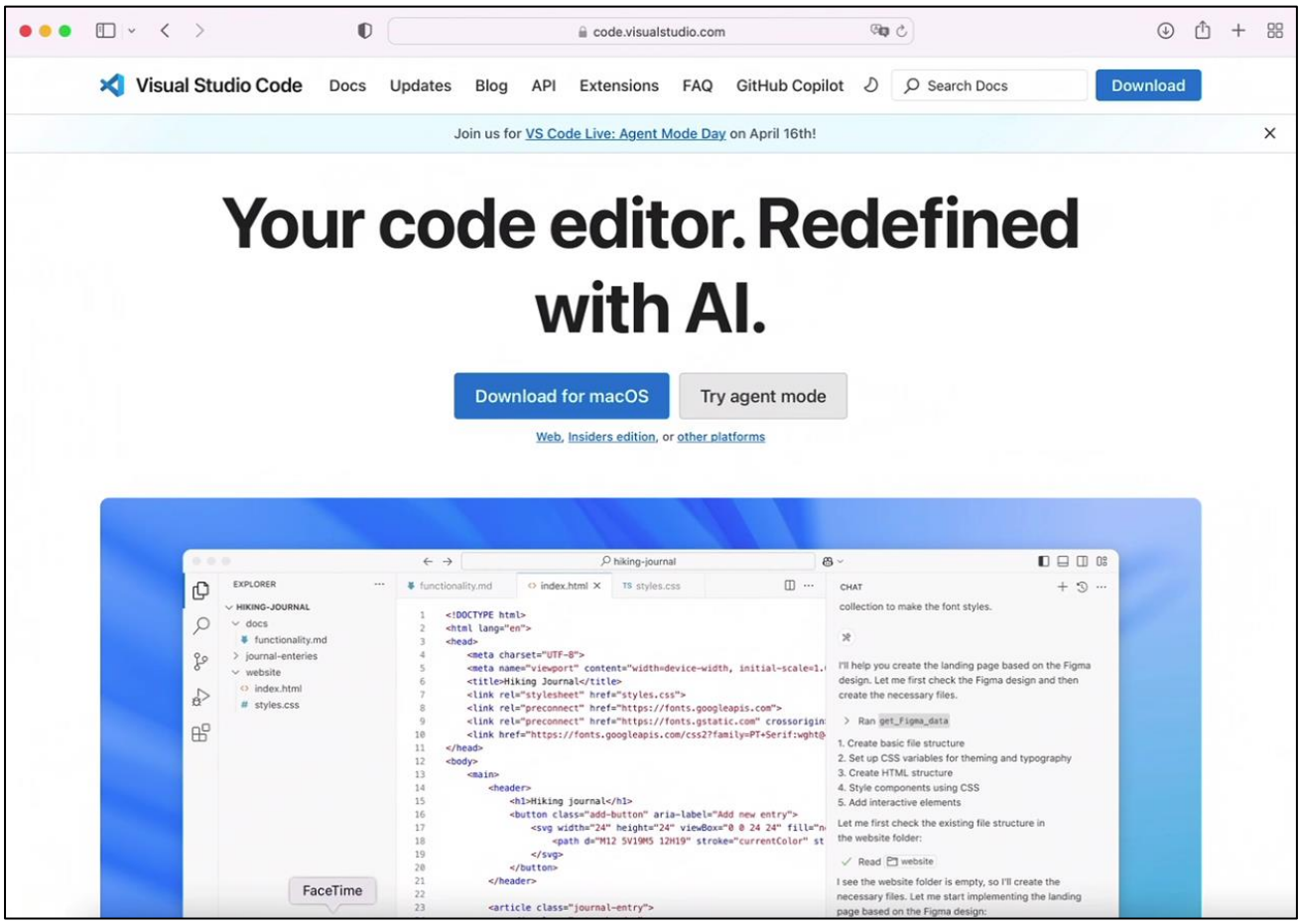

Setup - Microsoft Visual Studio Code (User)
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.
Additional icons:
Create a desktop icon
Other:
Add "Open with Code" action to Windows Explorer file context menu
Add "Open with Code" action to Windows Explorer directory context menu
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
Back
Next
Cancel

安装已完成，如下图所示。

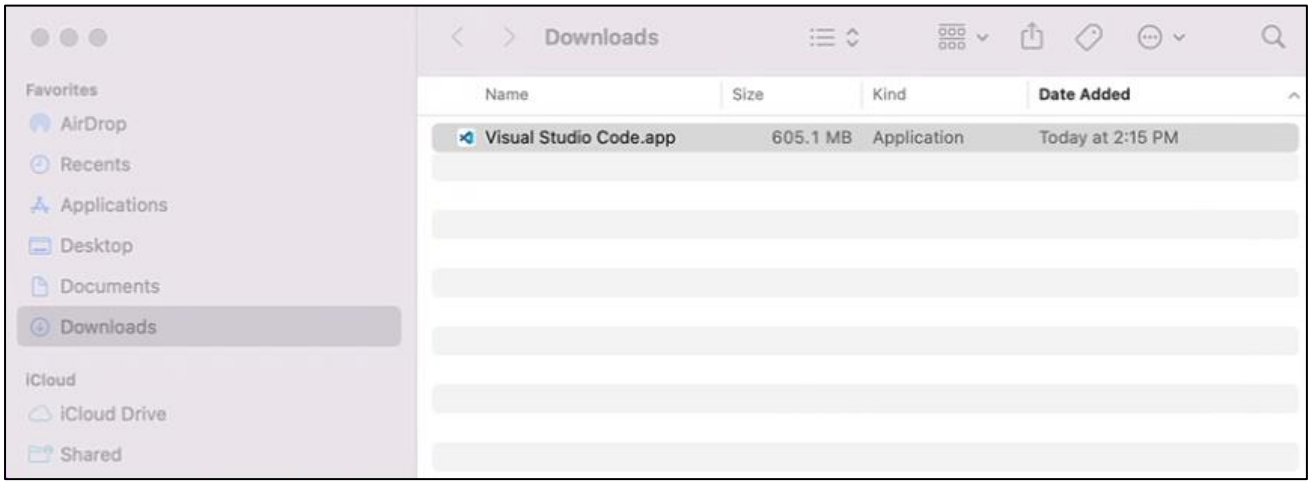

Setup - Microsoft Visual Studio Code (User)
Completing the Visual Studio Code Setup Wizard
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Visual Studio Code
Finish

Mac

通常情况下，macOS 系统已预装 Visual Studio Code。若您的设备未安装，请先完成安装。
访问 https://code.visualstudio.com 官网，点击"Download for macOS"（macOS 版下载）按钮。

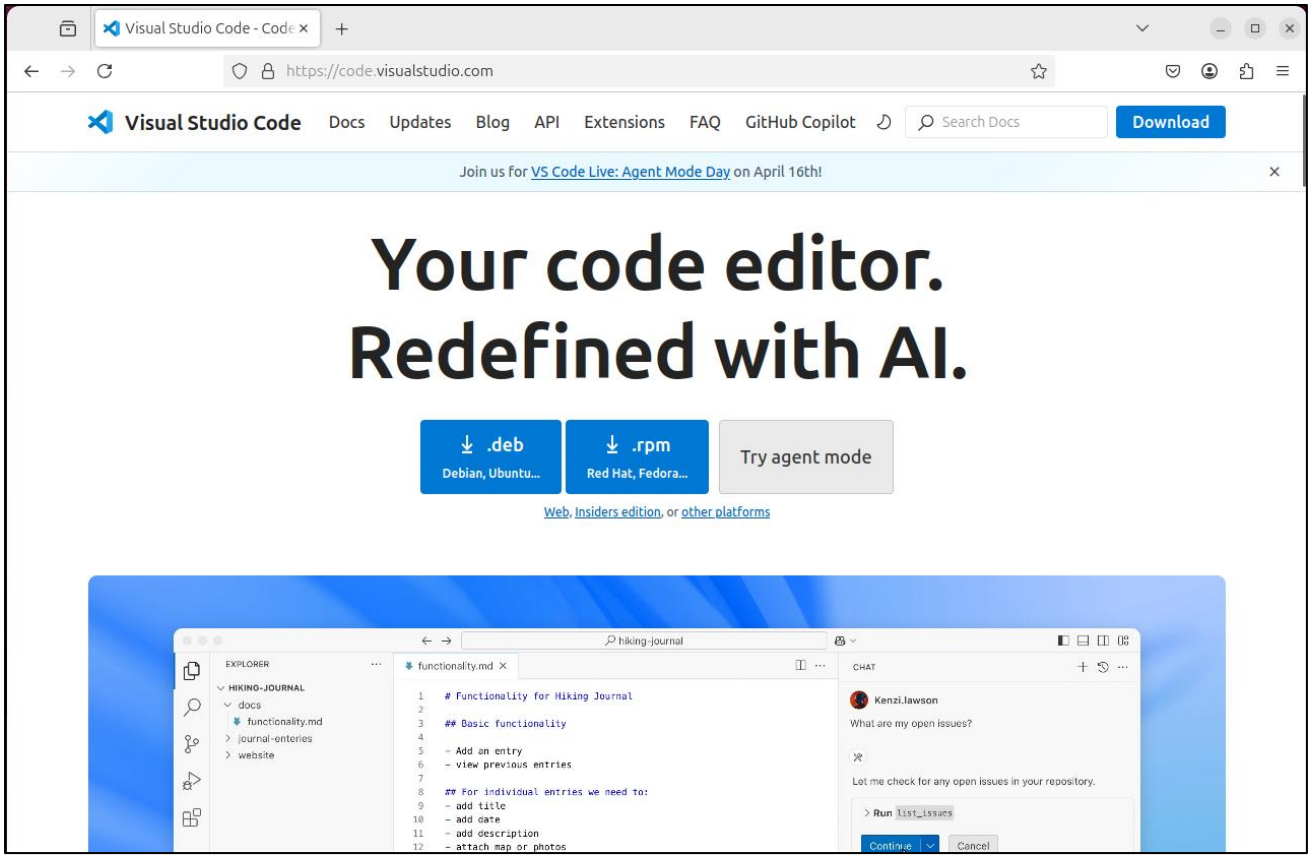


双击运行程序。

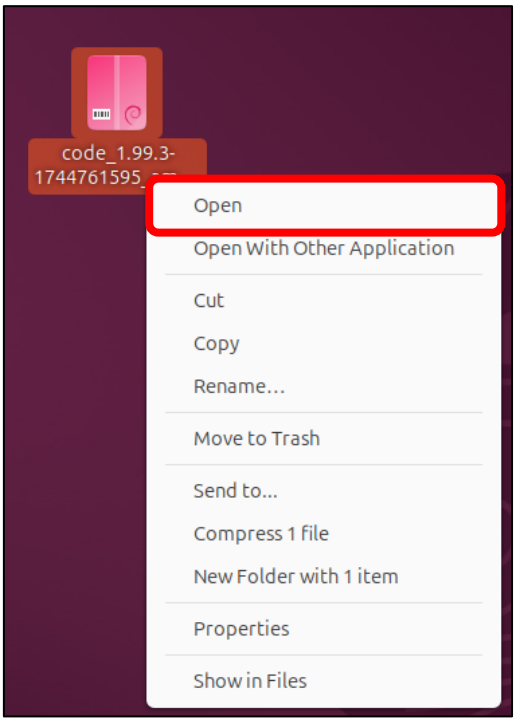


Linux

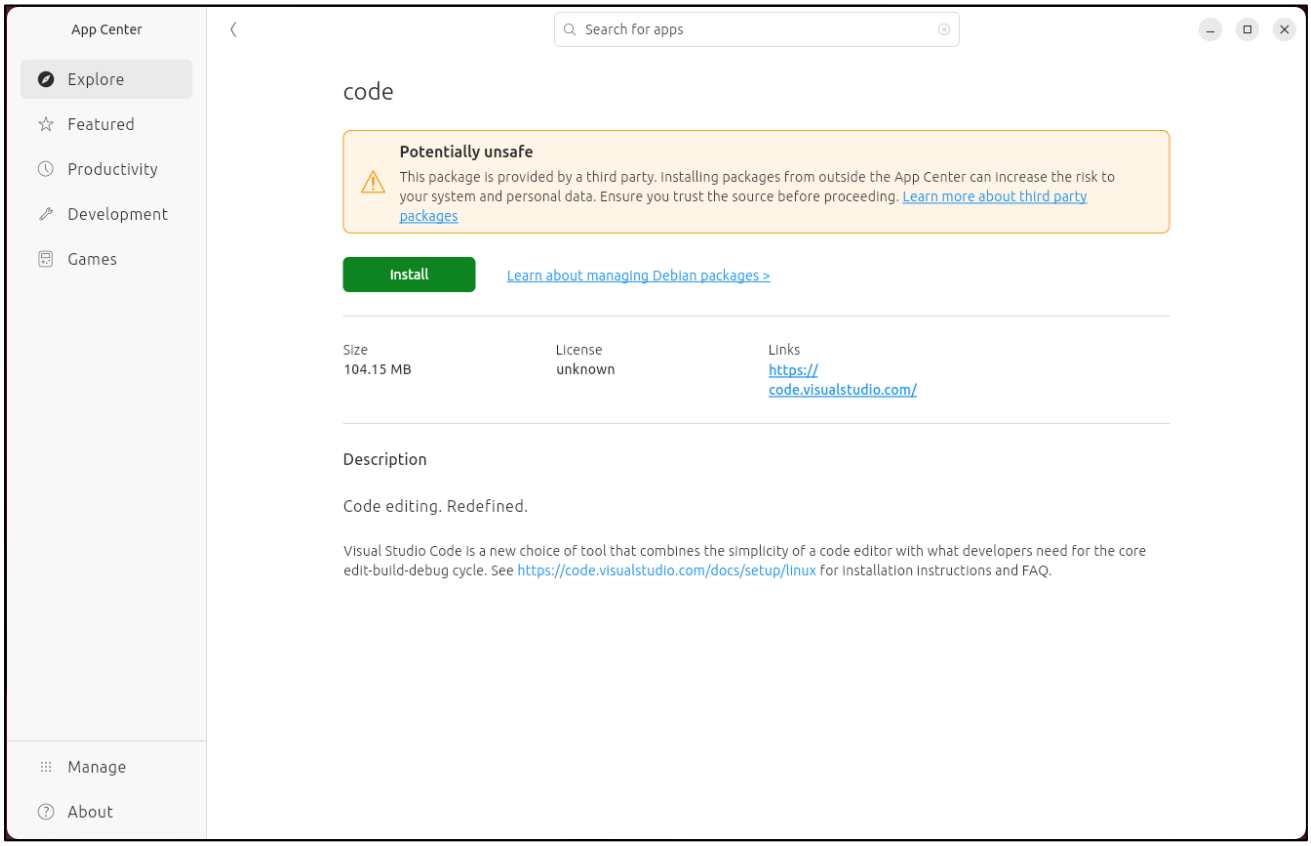
若您的设备未安装 Visual Studio Code，请先完成安装。
访问 https://code.visualstudio.com 官网，点击 .deb 格式安装包。



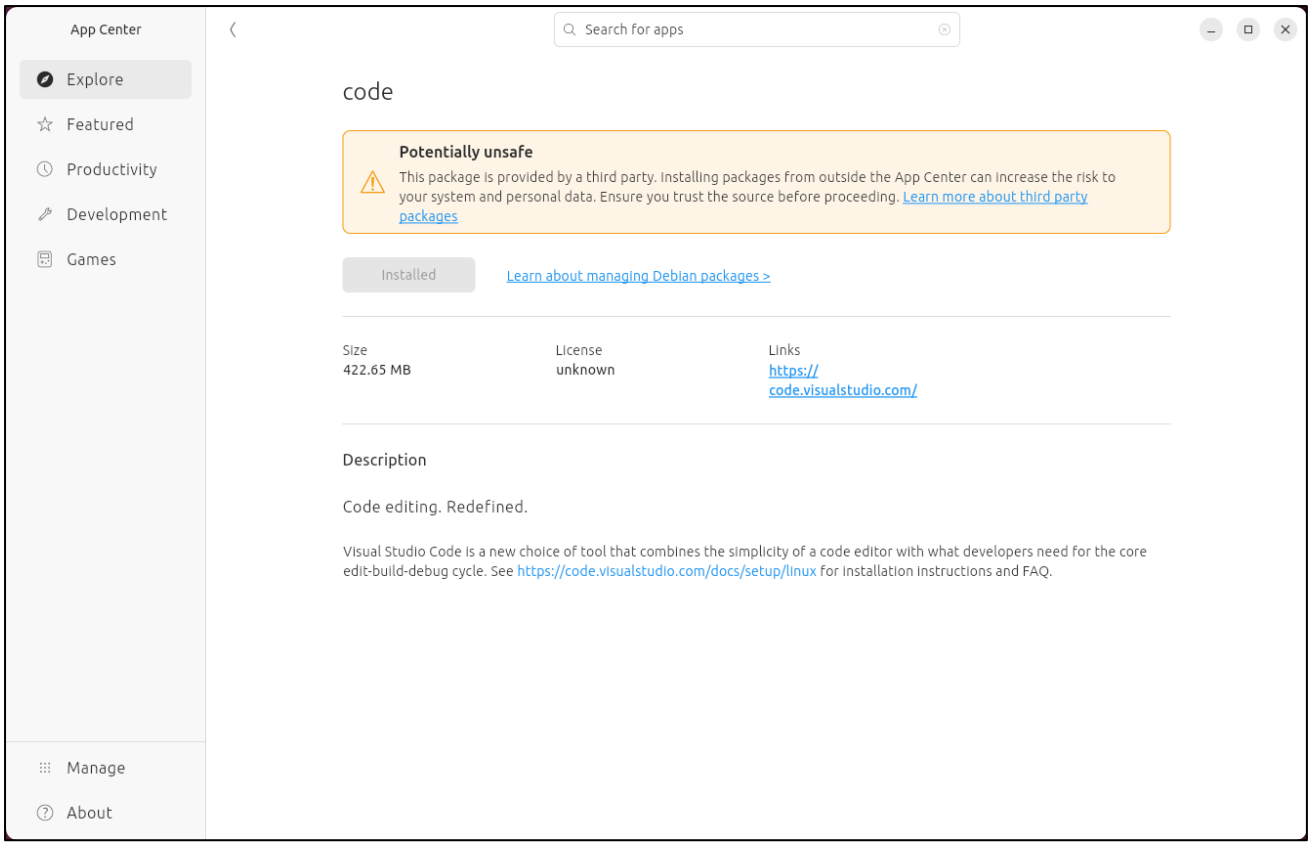
打开下载的 code_xxx.deb 安装包文件。



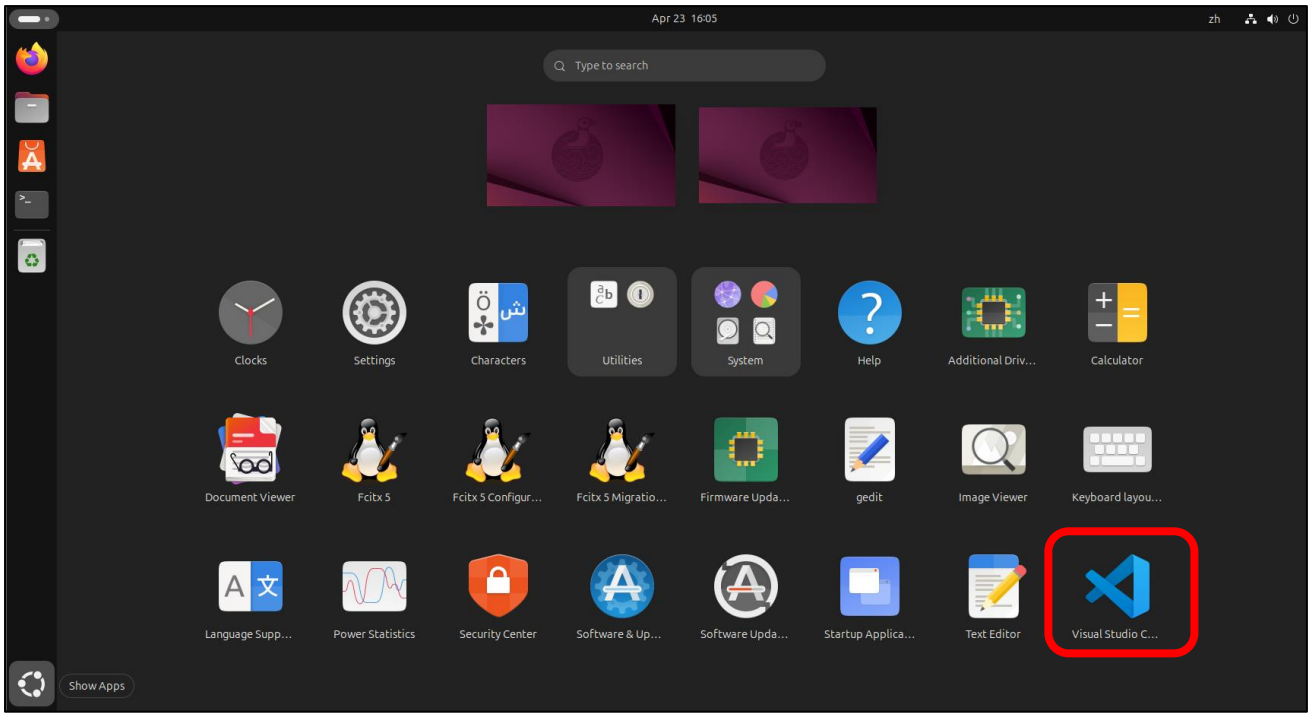
点击 Install（安装） 按钮完成 Visual Studio Code 的安装。



请等待安装完成。安装成功后，界面应如下图所示。



点击 Show Apps（显示应用程序） 即可在系统中看到 Visual Studio Code。

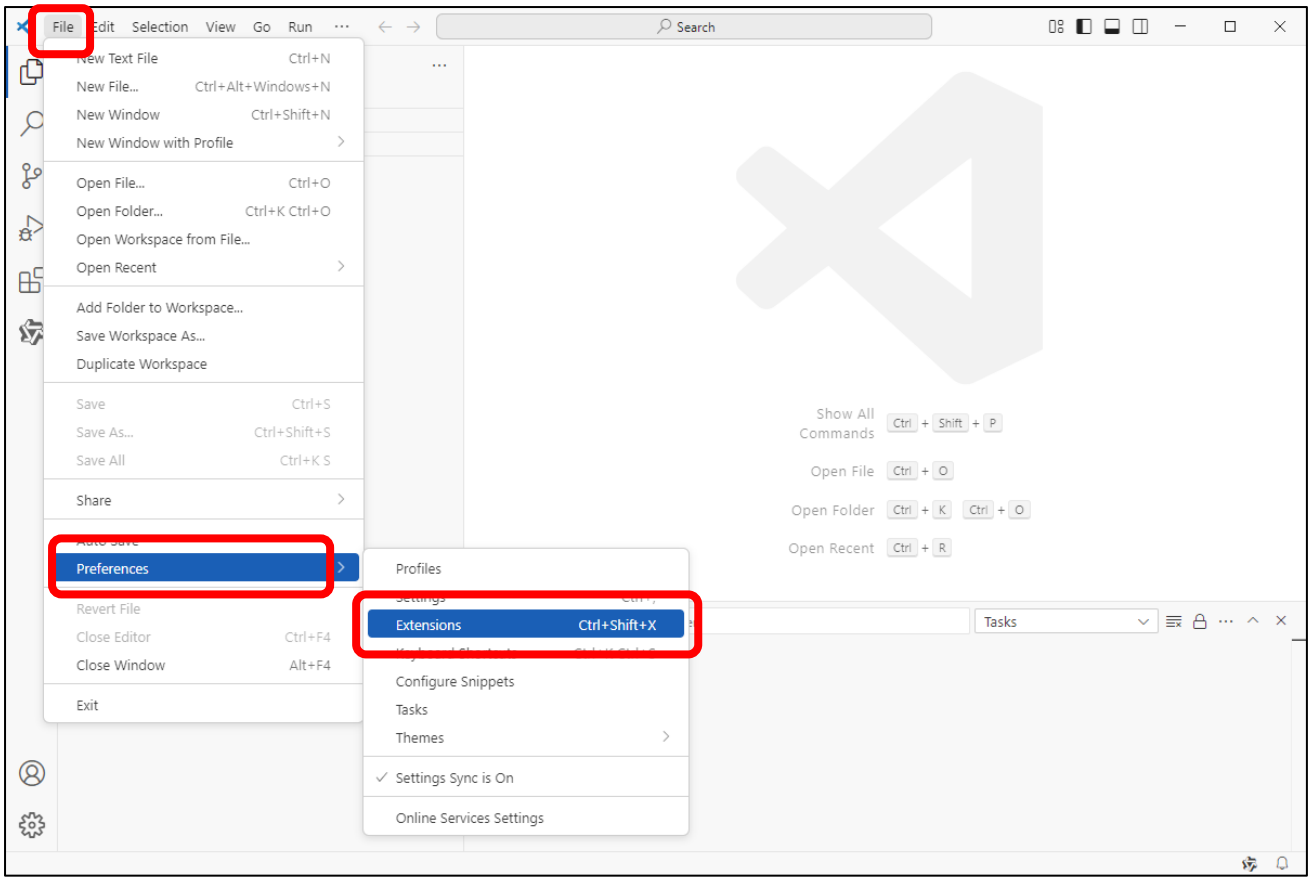


安装 ESP-IDF V5.3.2

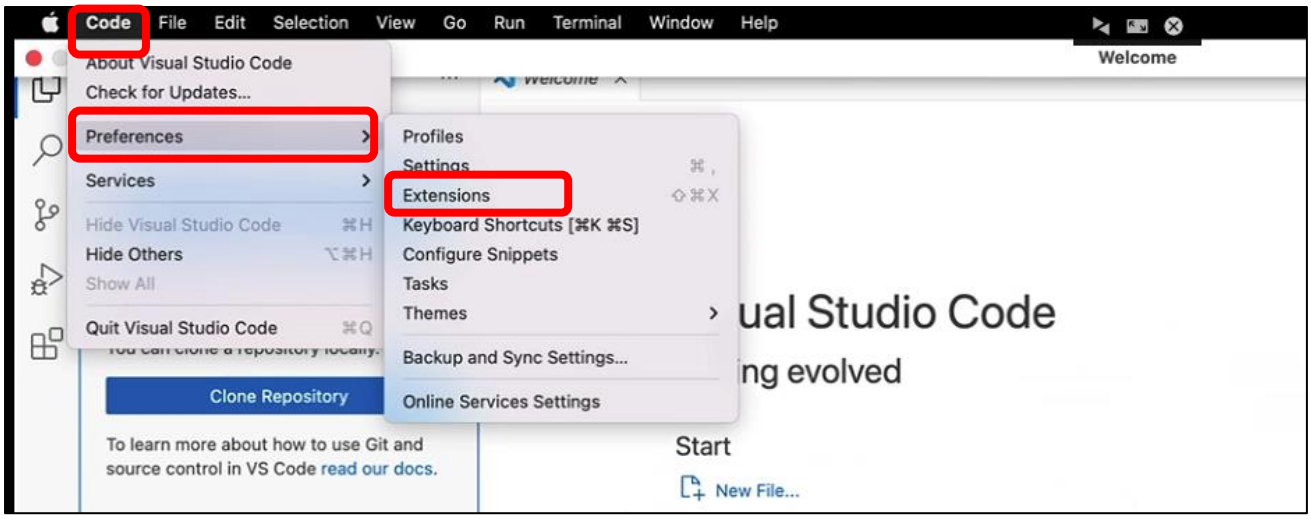
Visual Studio Code 是一款多功能代码编辑器。要使用 ESP-IDF SDK 进行编程，我们需要为其安装 ESP-IDF 扩展。

打开 Visual Studio Code。

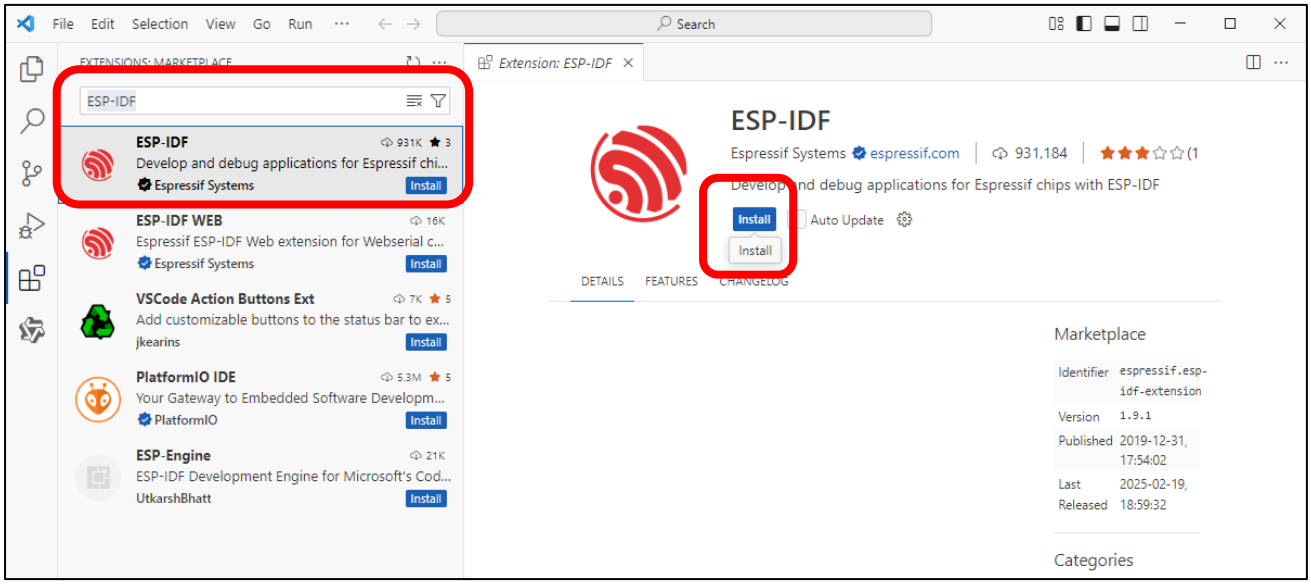
点击菜单栏：文件(File) -> 首选项(Preferences) -> 扩展(Extensions)。



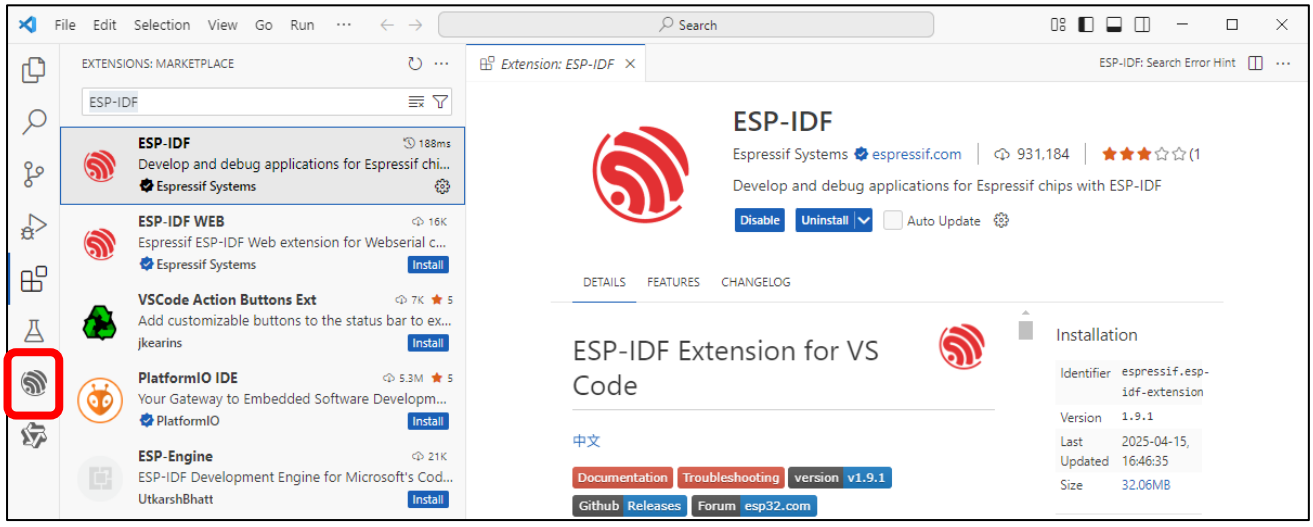
Mac OS 系统：点击菜单栏中的 “Code” -> “Preferences” -> “Extensions”（“扩展”）。



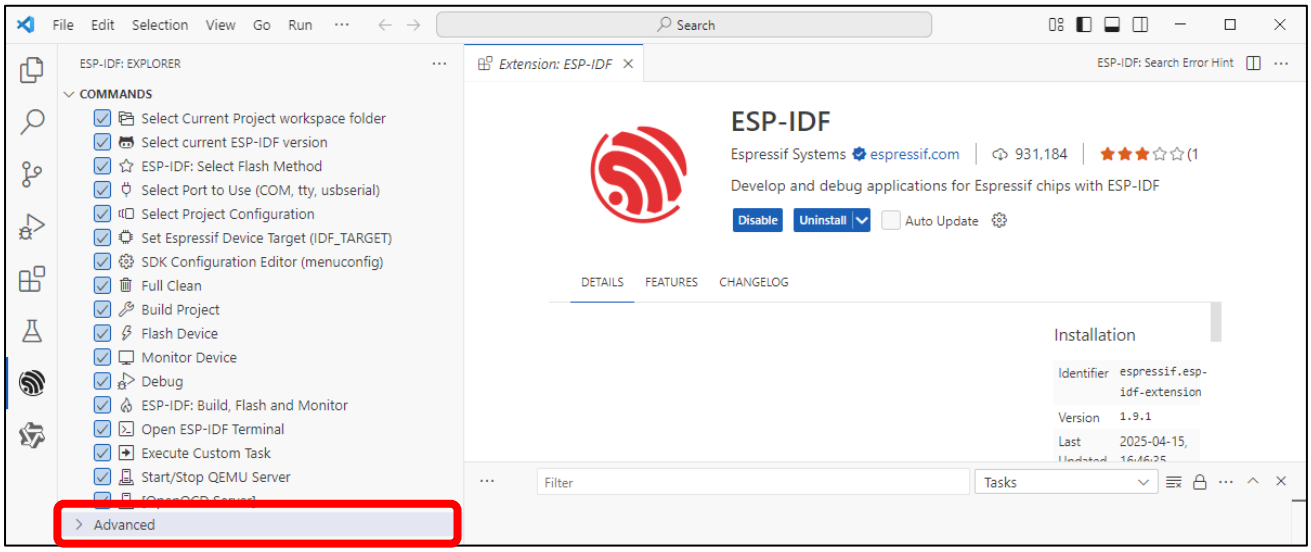
在扩展栏中搜索“ESP-IDF”，从列表中选择正确的结果，然后点击“Install”(安装）按钮继续。



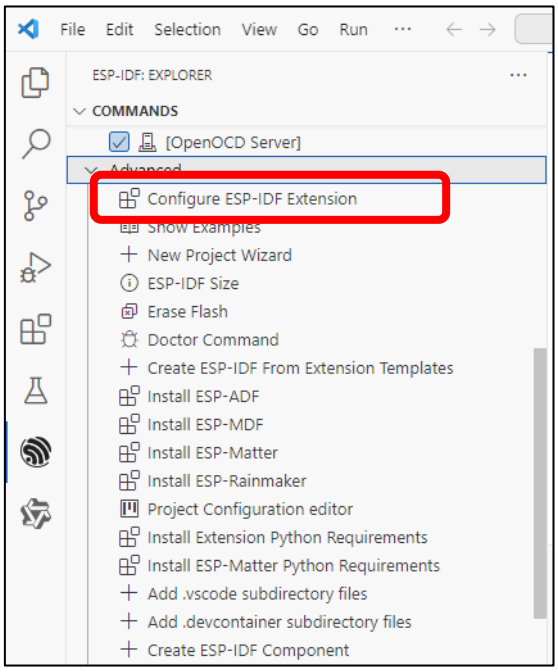
现在，ESP-IDF 扩展的图标会出现在左侧边栏中 —— 点击该图标即可继续操作。



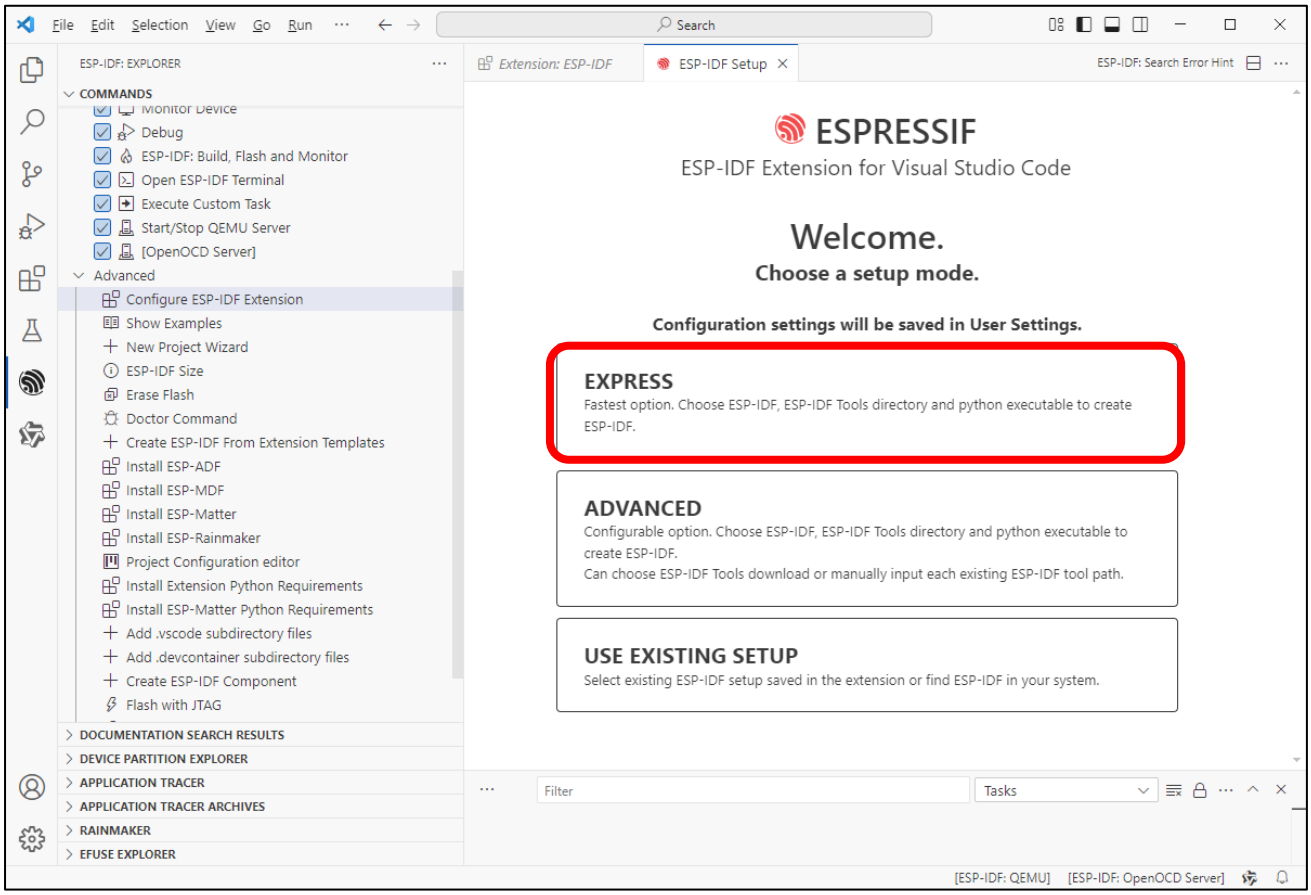
向下滚动鼠标，找到并点击“Advanced”(高级)选项。



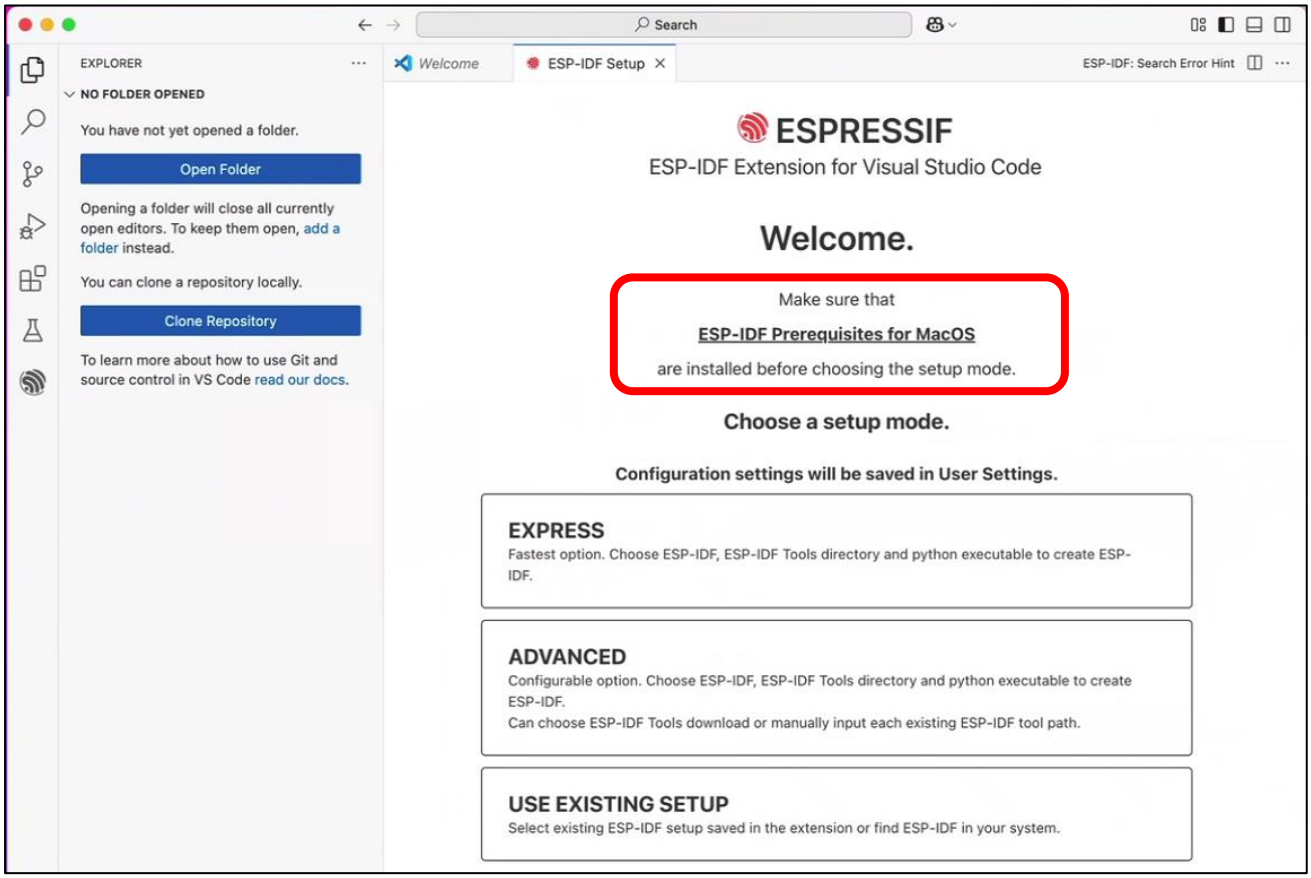
点击第一个选项：“Configure ESP-IDF Extension”(配置 ESP-IDF 扩展)。

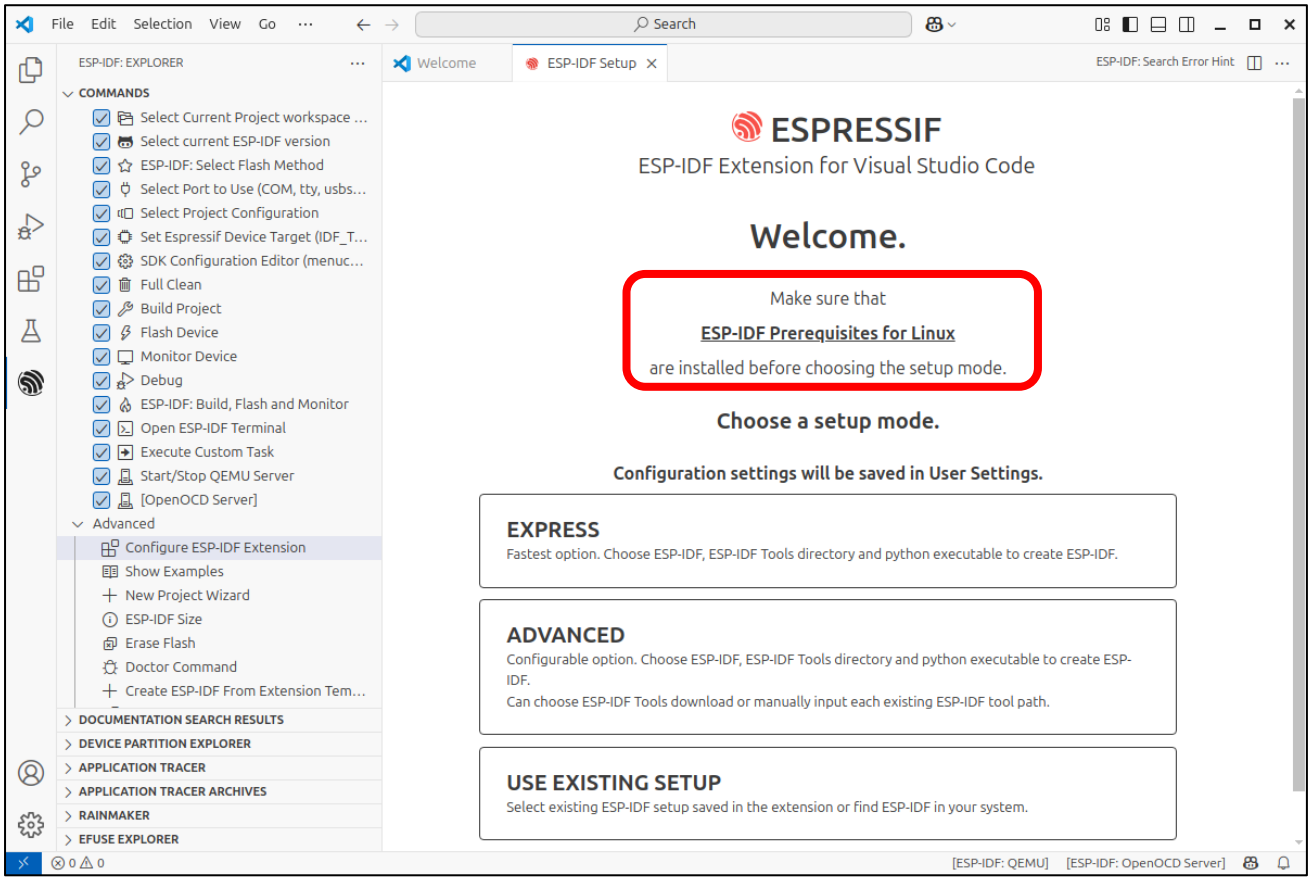


在右侧选择“EXPRESS”（快速安装） 选项。

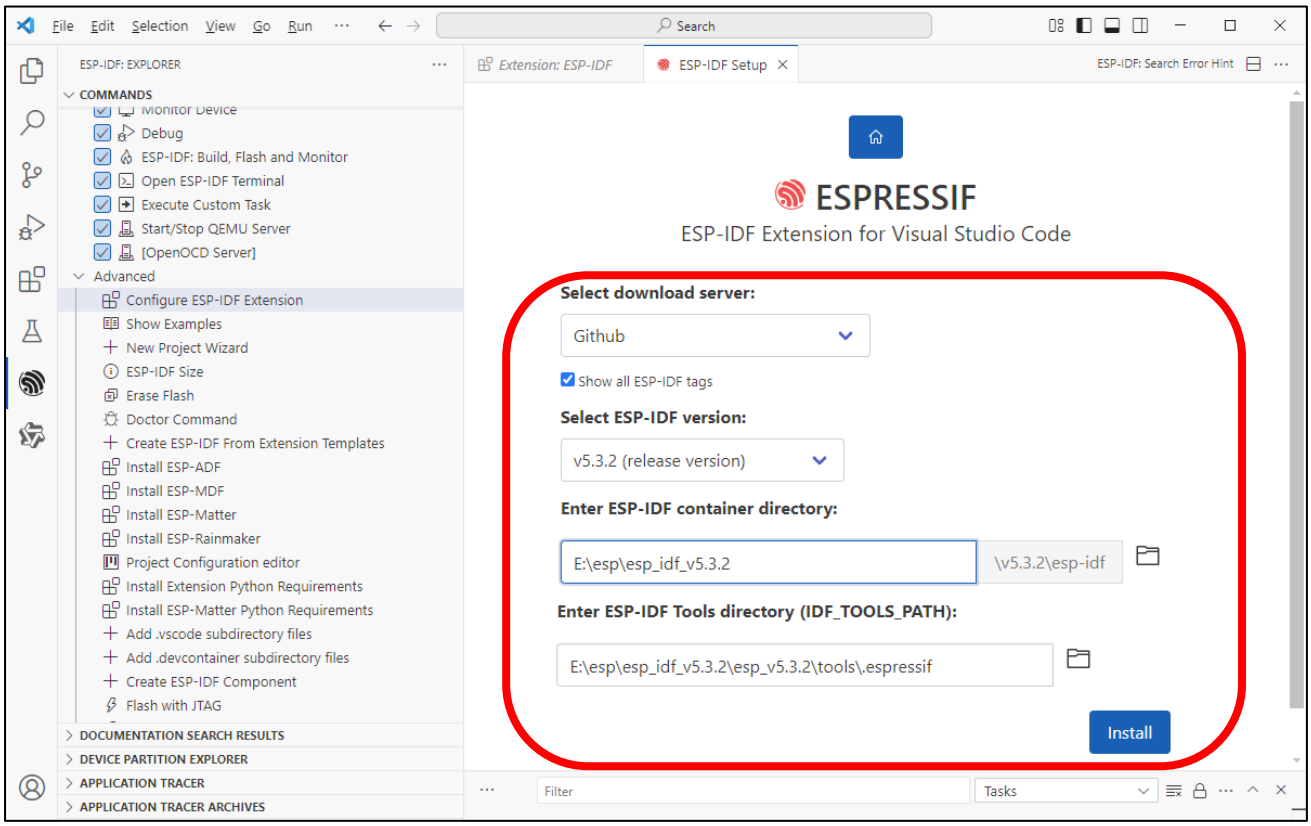


请注意：如果您使用的是 macOS 或 Ubuntu 系统，请根据提示完成必要的准备工作，再继续安装。



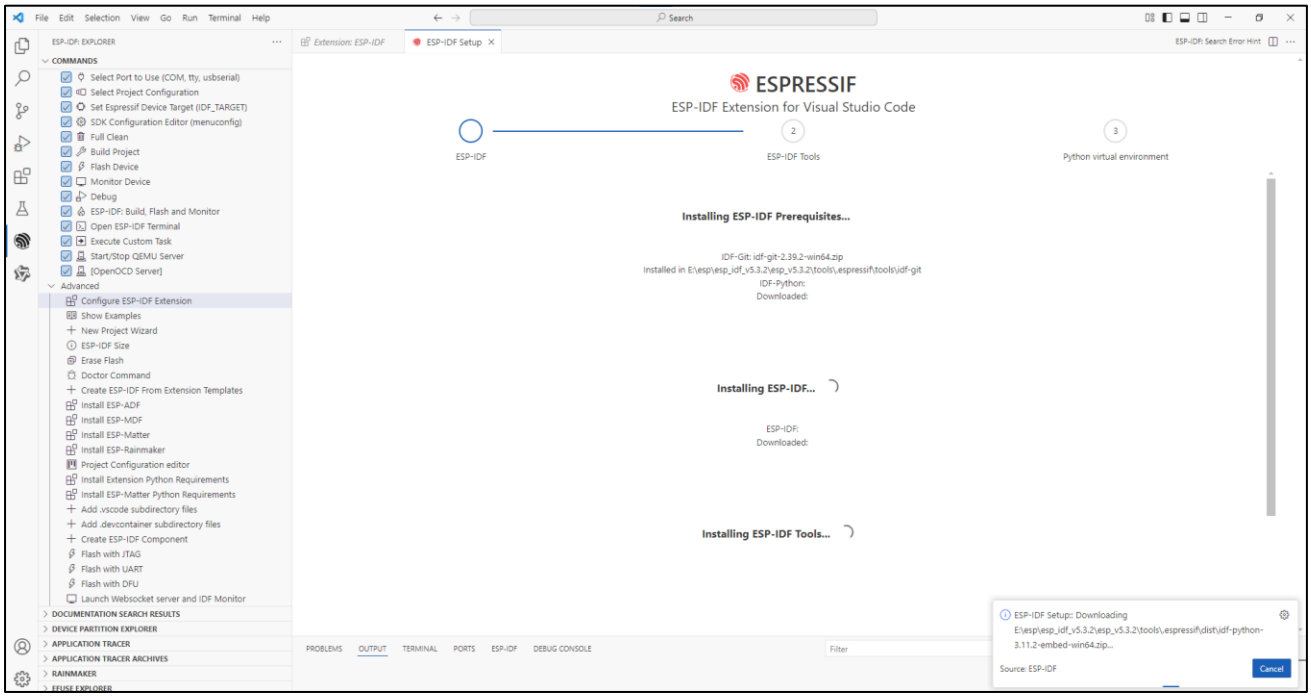


1. 勾选 “Show all ESP-IDF tags”(显示所有 ESP-IDF 版本标签) 的复选框。
2. 从下拉菜单中选择 “v5.3.2 (release version)”(v5.3.2 正式版)。
3. 选择您希望安装 ESP-IDF 环境 的目标路径。
4. 点击 “Install”(安装)按钮开始配置。



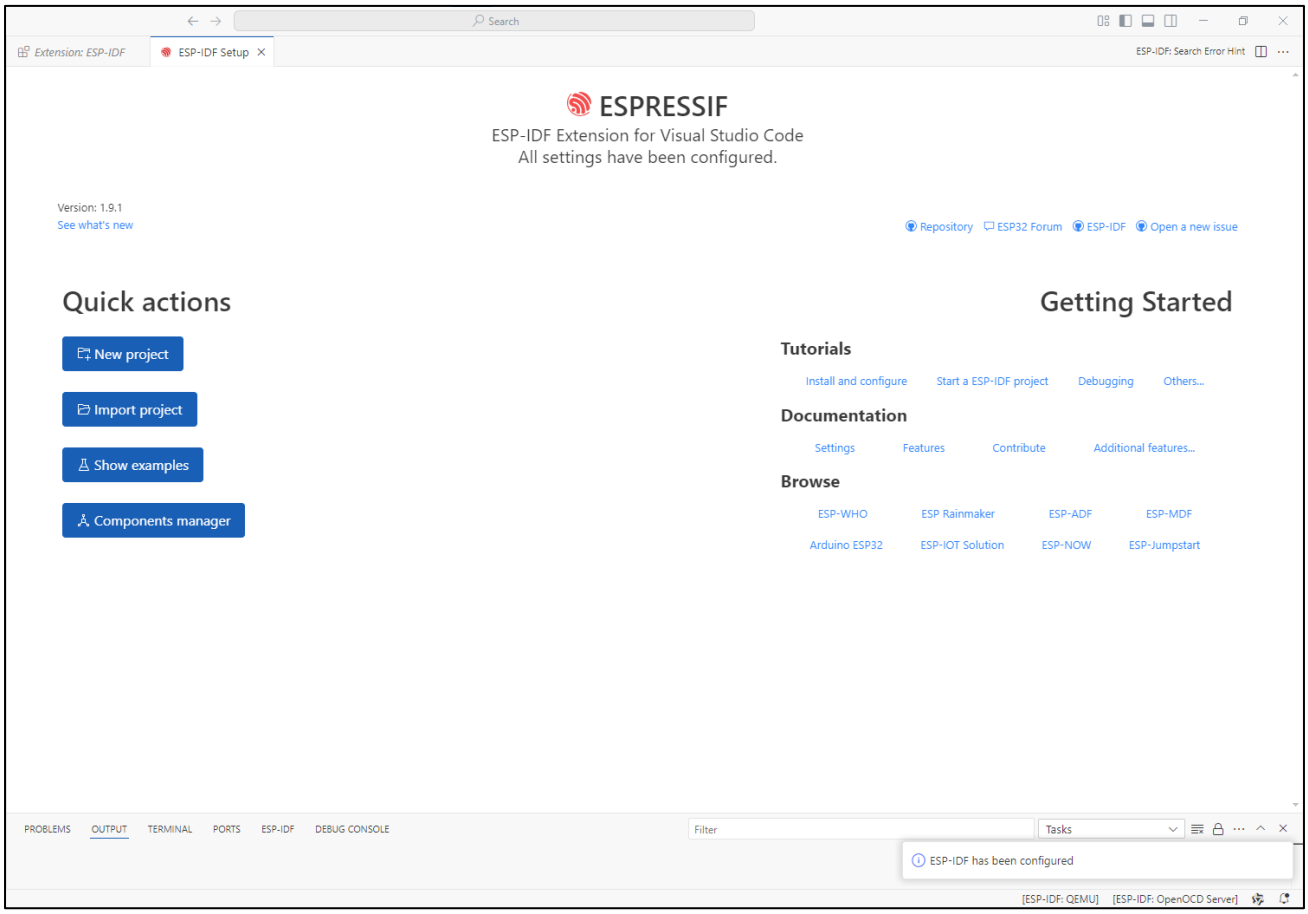
该过程将自动完成。

如果安装失败，请找到您选择的 ESP-IDF 目录，删除其中的安装失败文件夹，然后重新安装。



此步骤可能需要一些时间，请确保您的网络连接稳定且快速。

完整安装过程如下图所示。



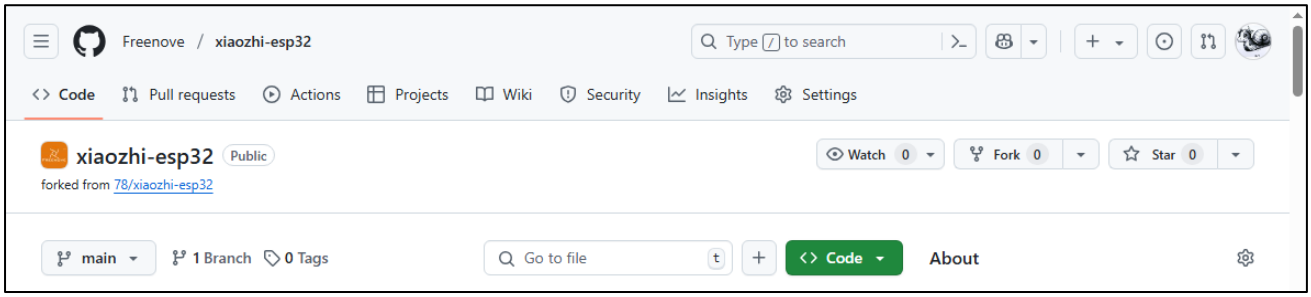
有关 ESP-IDF 的更多信息，请参考

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

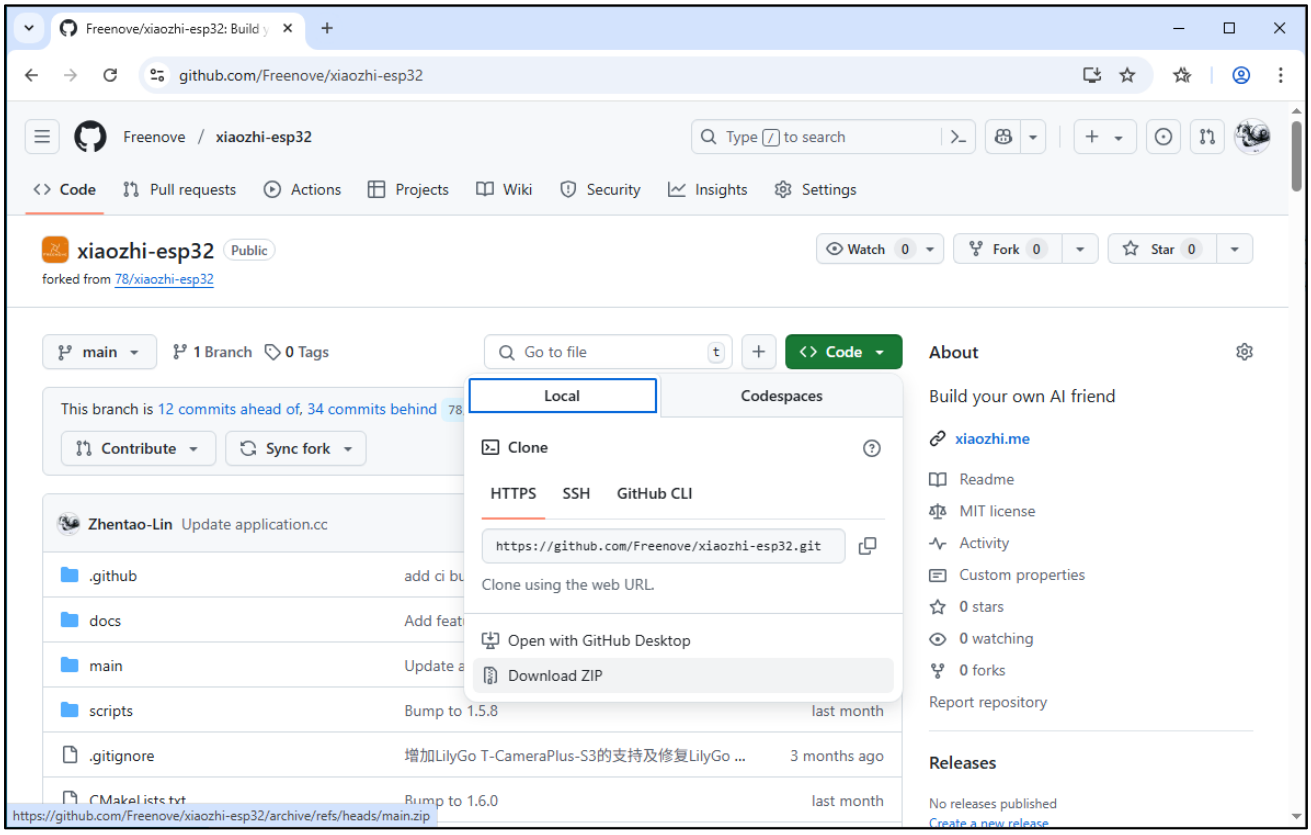
代码下载

Windows

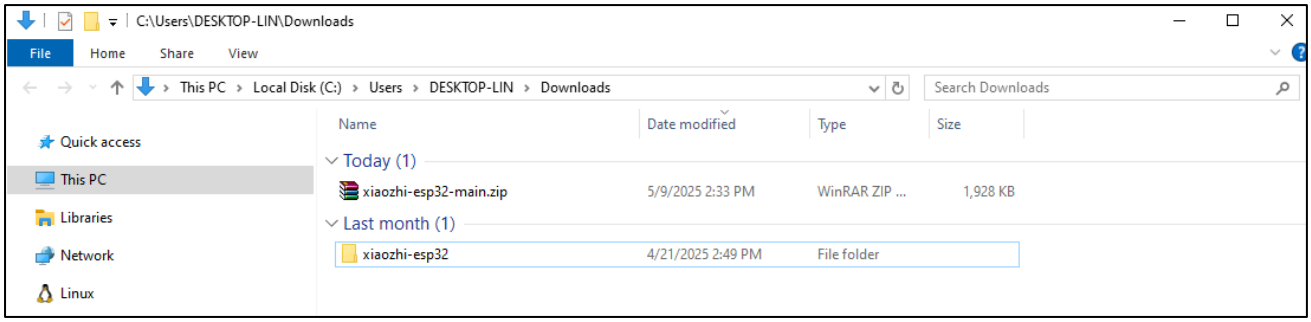
在电脑上打开浏览器，输入“https://github.com/Freenove/xiaozhi-esp32”.



点击“Code”-> “Download ZIP”，将代码下载到你的电脑上。



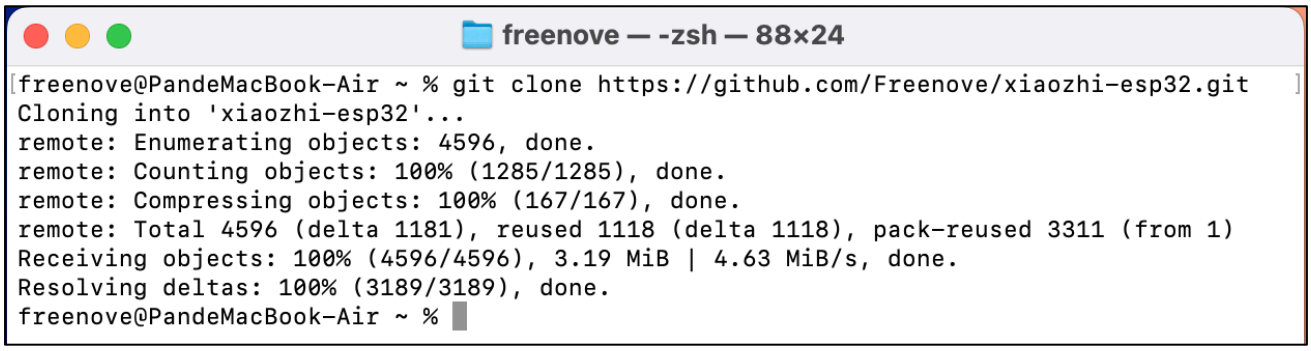
将下载的 zip 文件解压到电脑上，并将解压后的文件夹重命名为“xiaozhi-esp32”。



Mac

打开终端，使用 git 命令下载代码。

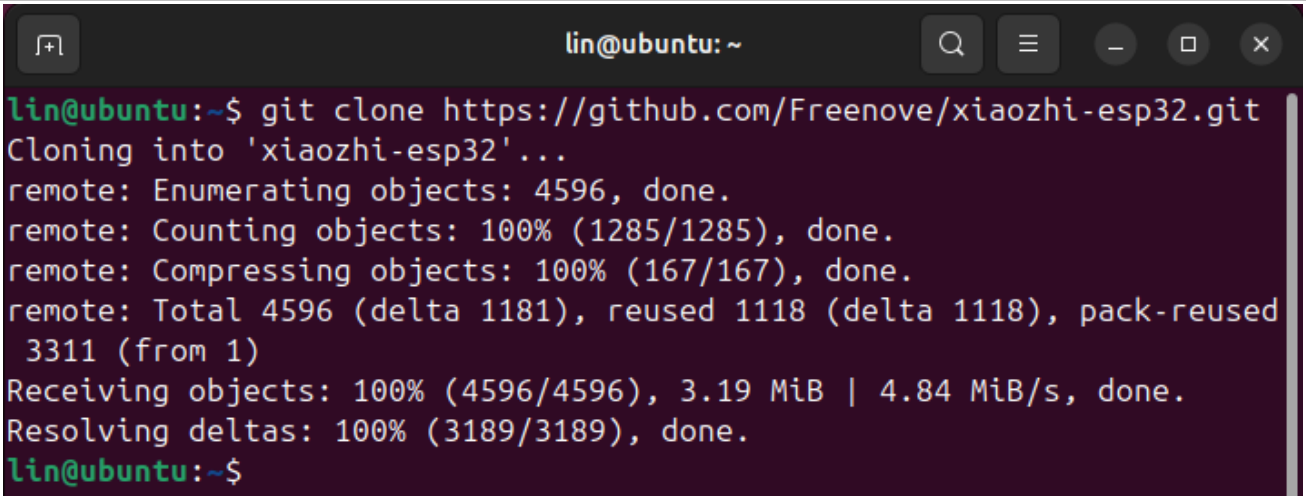
```
git clone https://github.com/Freenove/xiaozhi-esp32.git
```



Linux

打开终端，使用 git 命令下载代码。

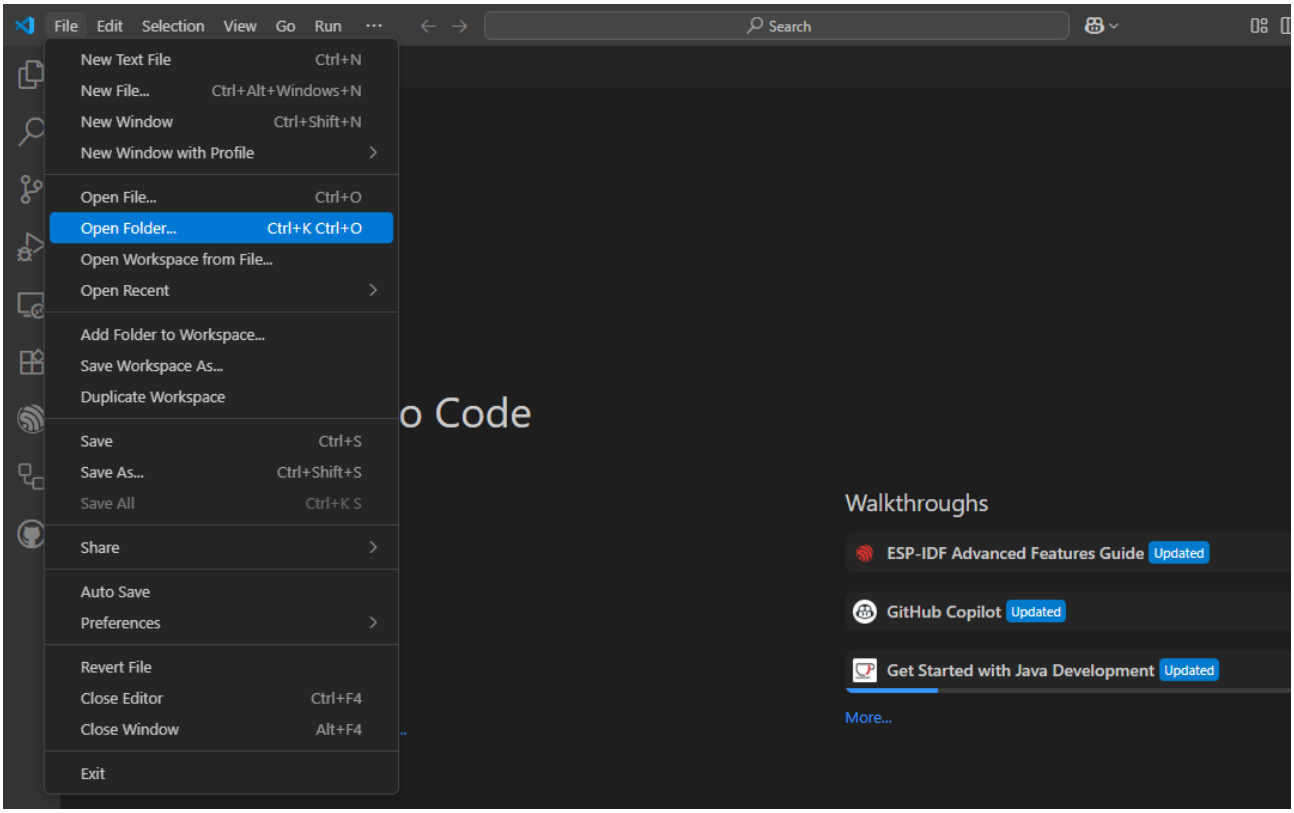
```
git clone https://github.com/Freenove/xiaozhi-esp32.git
```



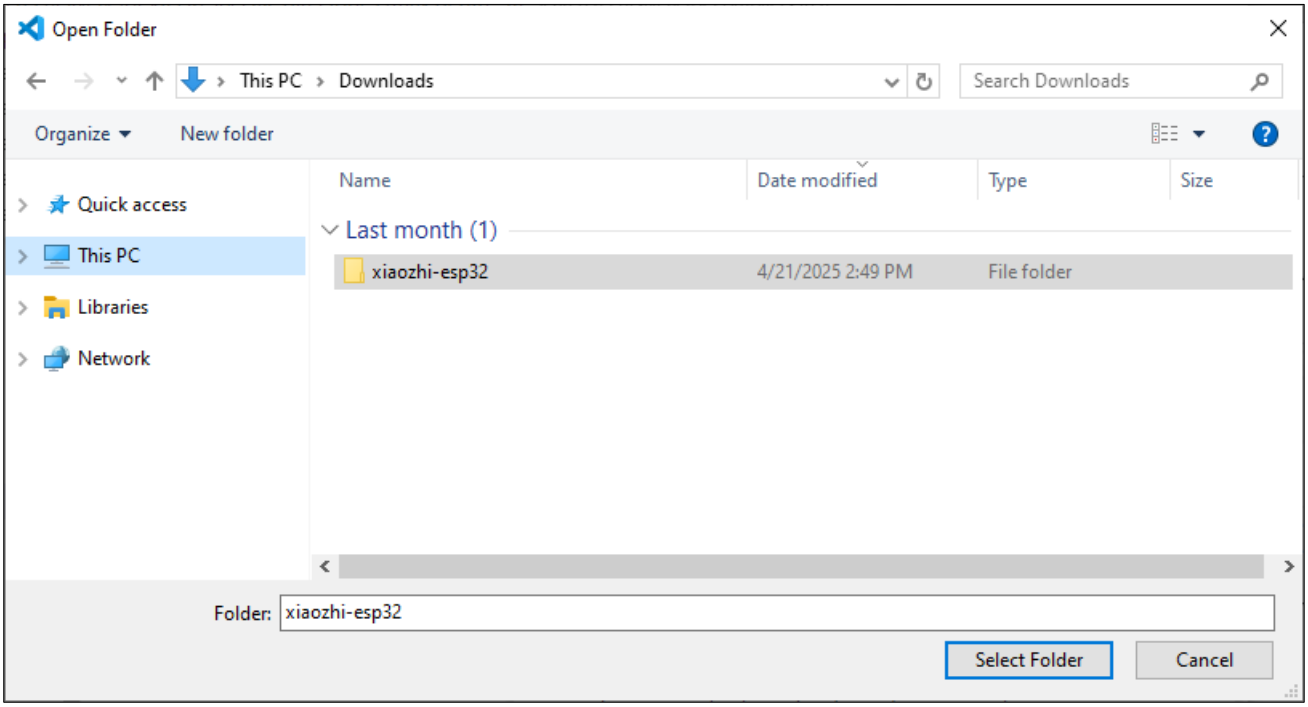
配置代码环境

解压下载的 ZIP 文件。

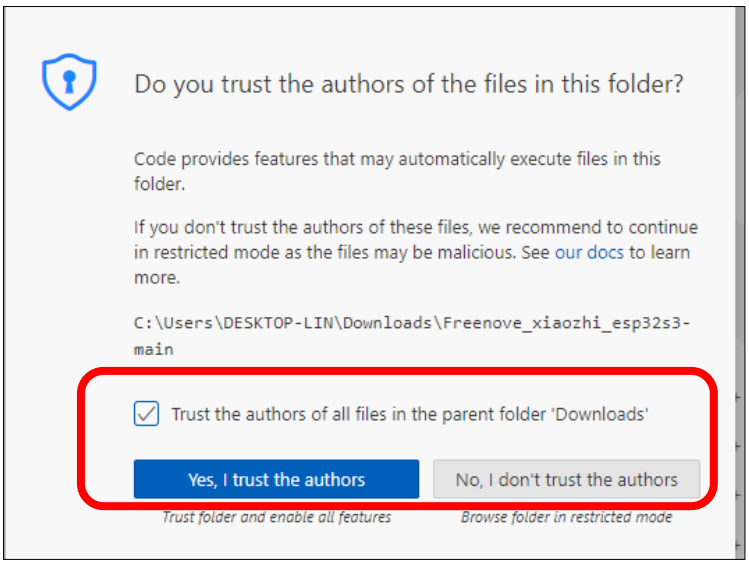
在 Visual Studio Code 中，点击“文件” -> “打开文件夹…”。



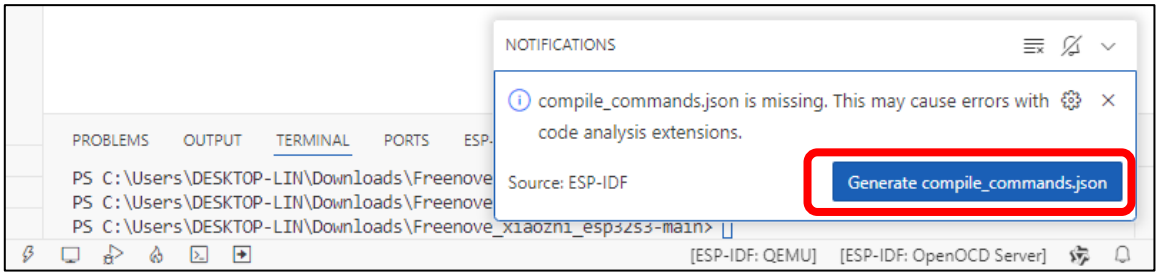
选择 **xiaozhi-esp32** 文件夹。此处以 Windows 系统界面为例，Mac 系统和 Linux 系统的操作类似。



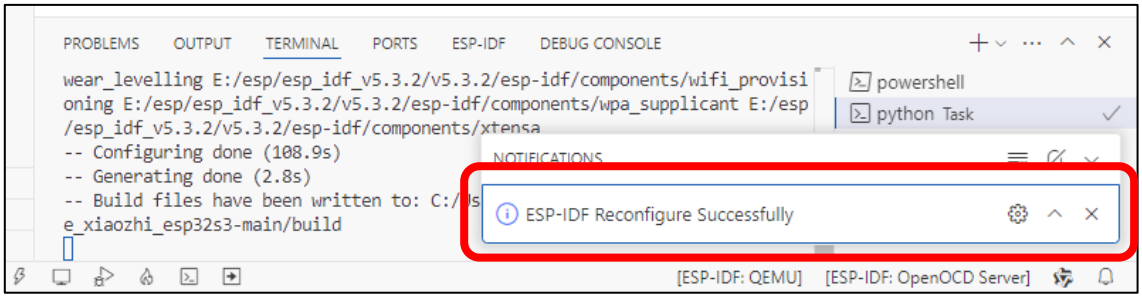
勾选“信任‘Downloads’父文件夹中所有文件的作者”，然后选择“是，我信任这些作者”。



请注意：右下角将弹出提示框，点击“生成 comple_commands.json”，系统将根据该文件下载对应的组件模块代码。

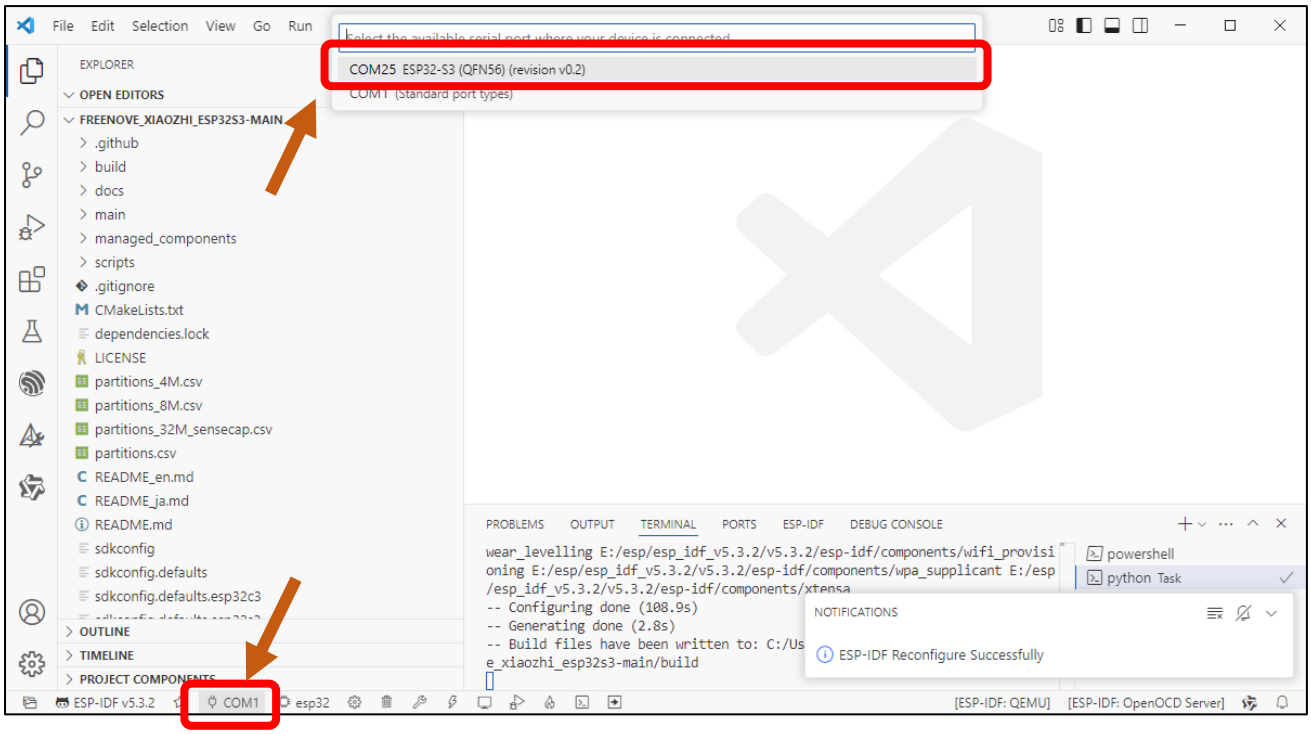


组件安装可能需要一些时间，请耐心等待并避免进行其他操作。安装完成后，右下角会显示完成通知。

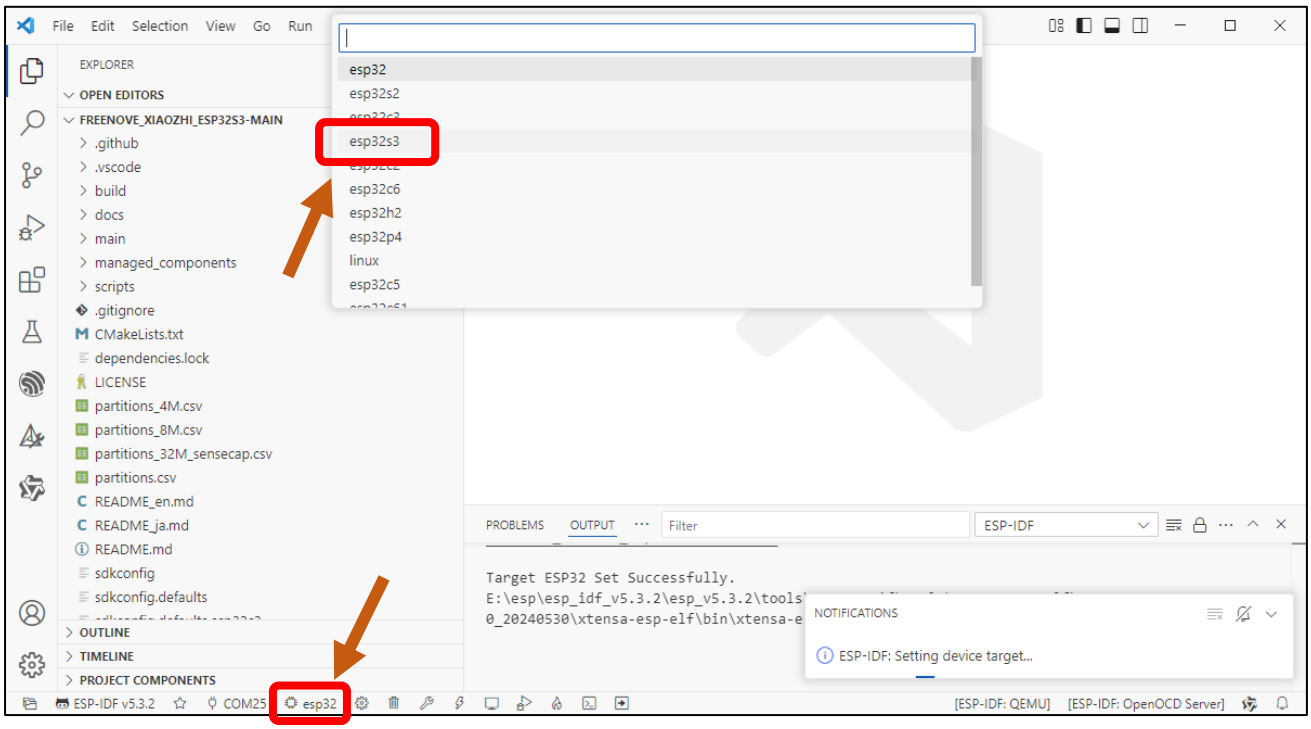


使用 USB 线将 ESP32-S3 连接至电脑，注意必须插入正确的 Type-C 接口（请勿接错插口）

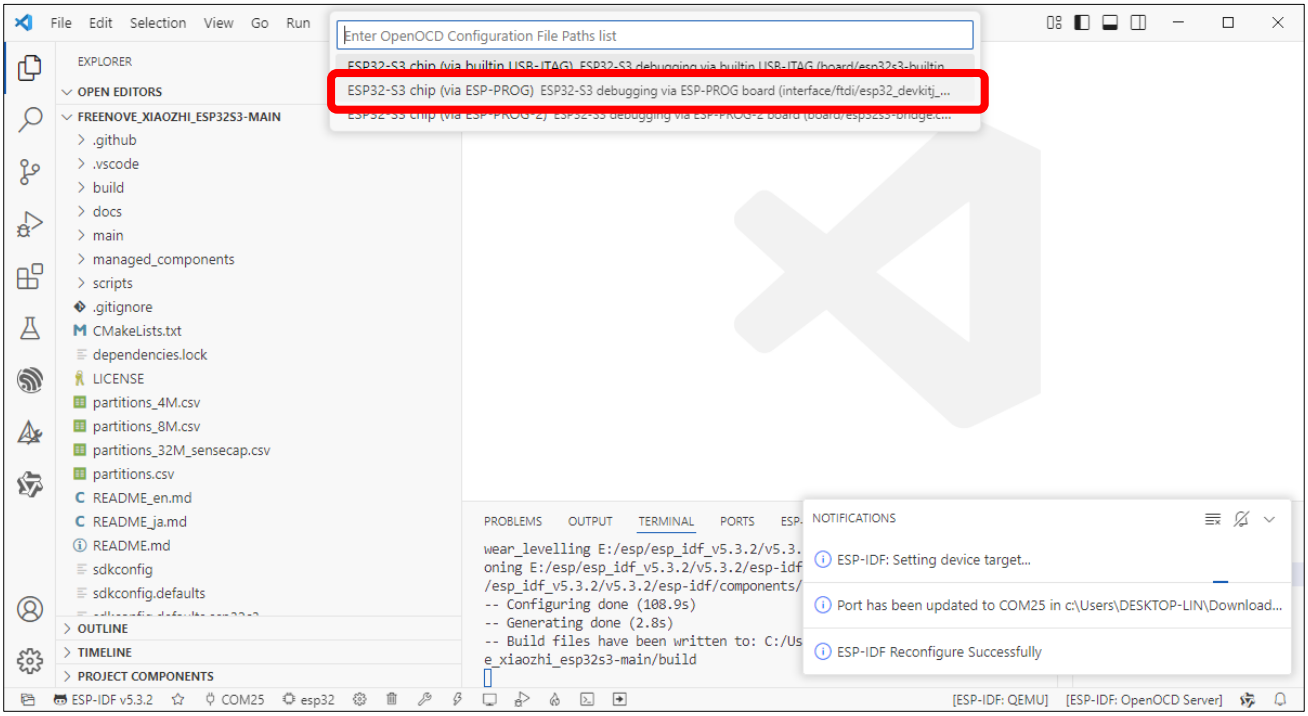
点击左下角的 **'COMx'** 按钮，显示电脑所有可用 COM 端口。找到并选择标有"ESP32-S3"的选项



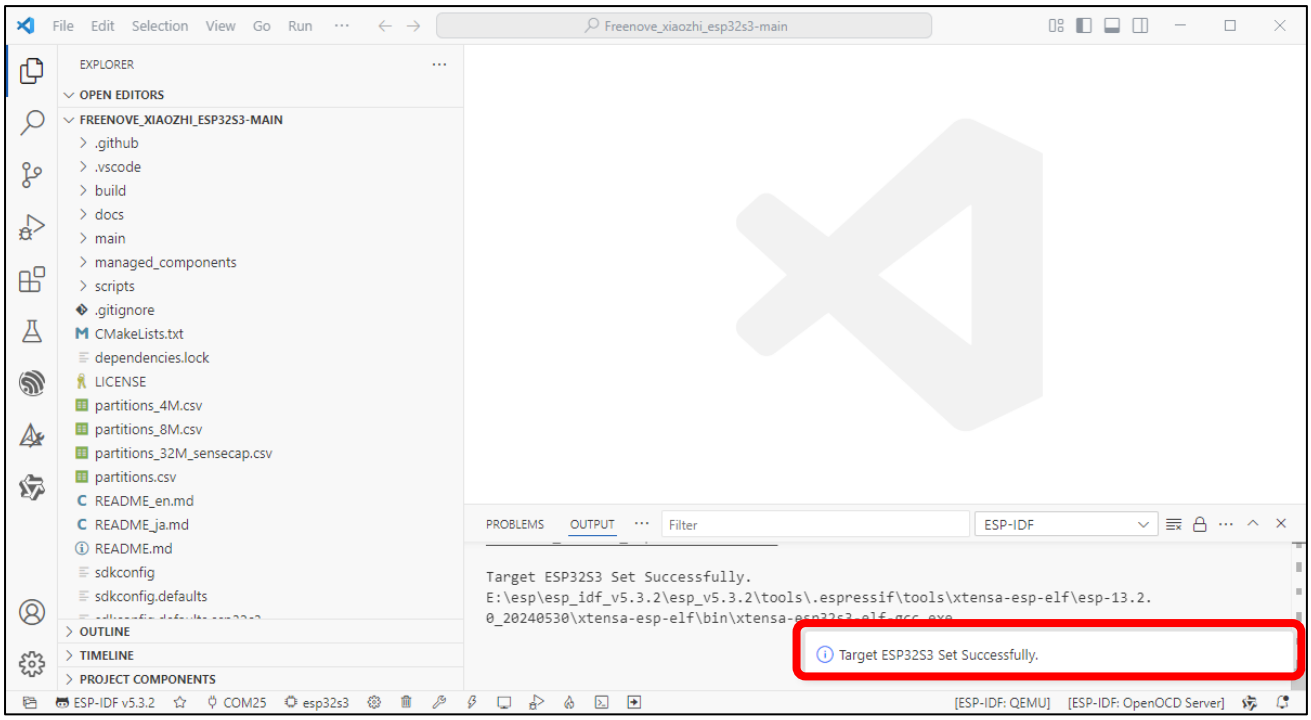
点击左下角的 **'ESP32'** 按钮，显示所有可用 ESP32 型号，然后从列表中选择 'ESP32-S3'。



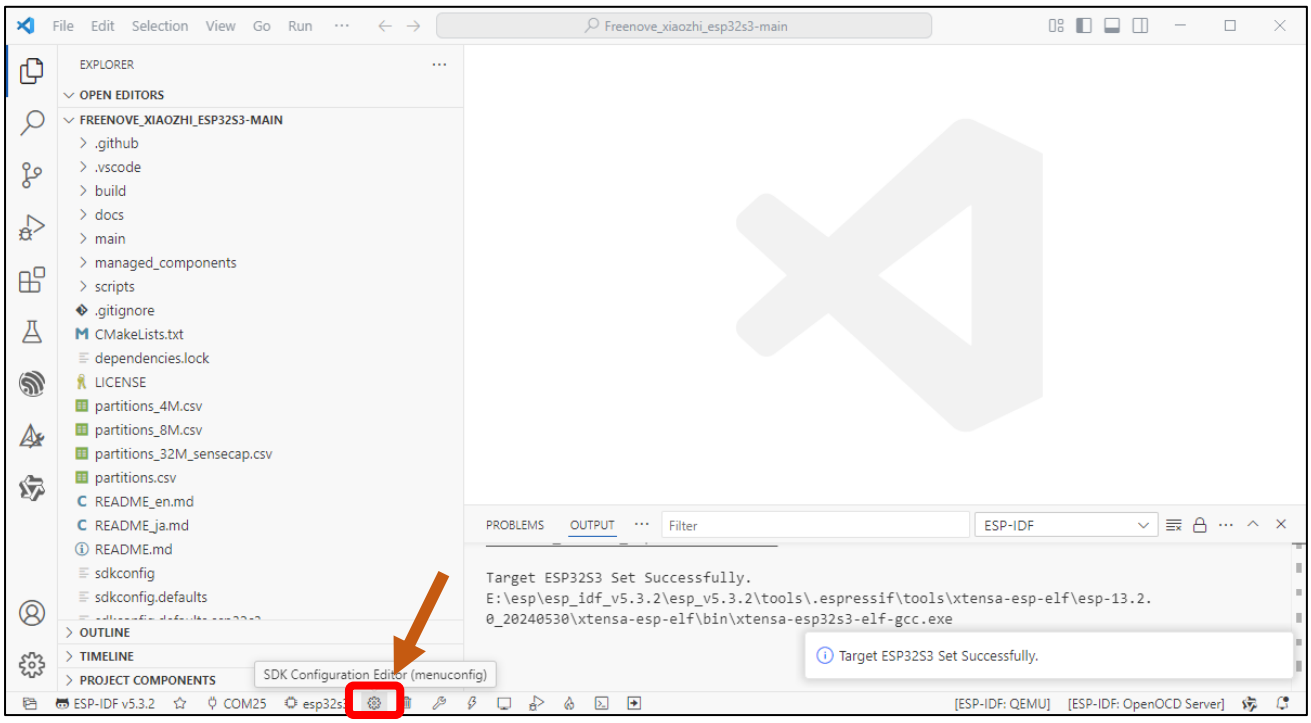
从新出现的选项菜单中，选择 'ESP32-S3 芯片（通过 ESP-PROG） - 使用 ESP-PROG 开发板调试 ESP32-S3...'



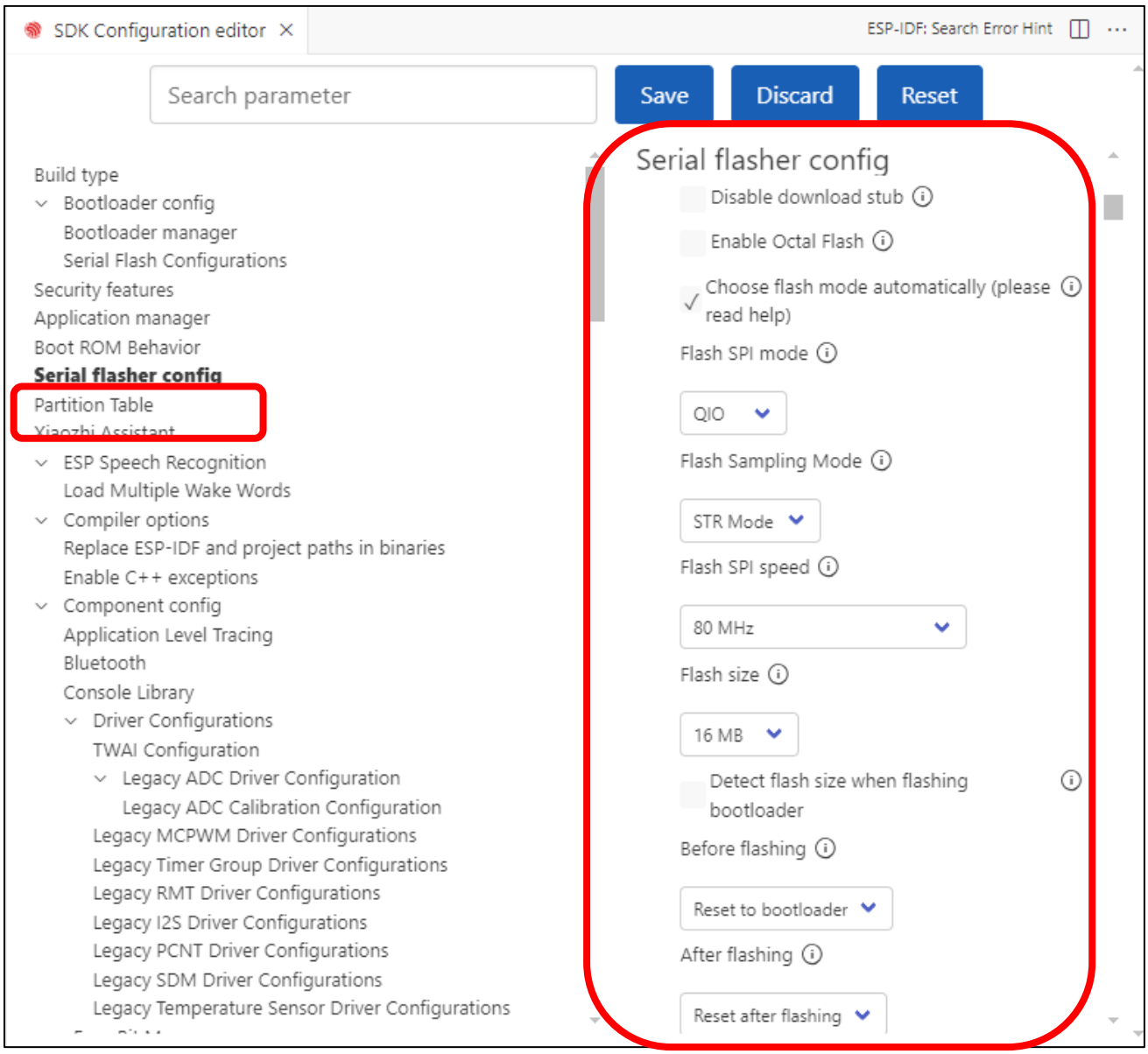
请等待，直到右下角显示**“目标设备 ESP32S3 设置成功”**的提示。



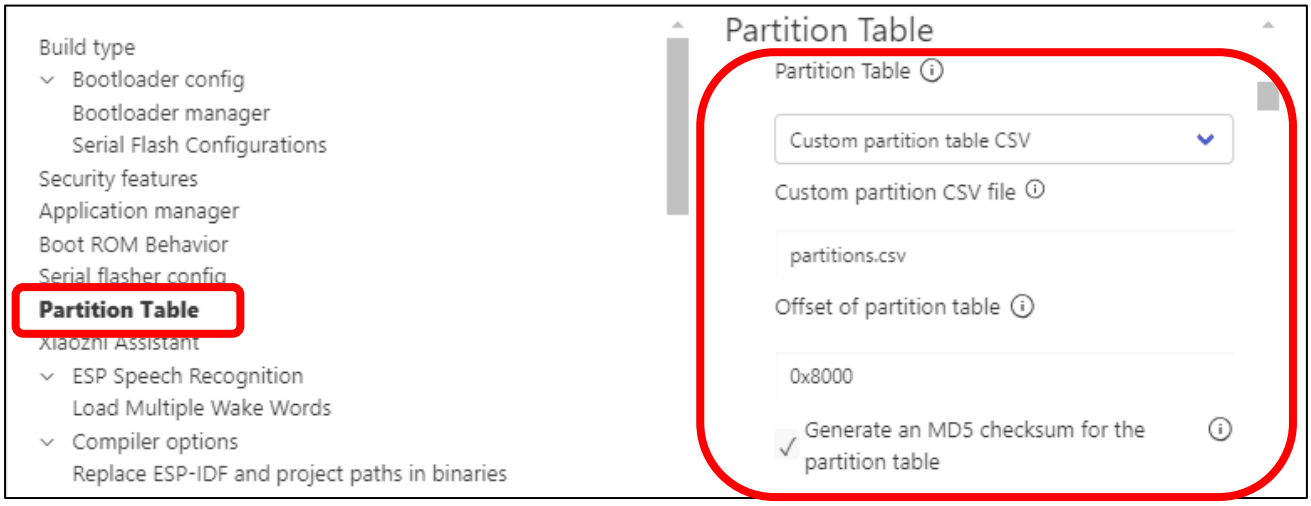
点击底部的“SDK 配置编辑器(menuconfig)”



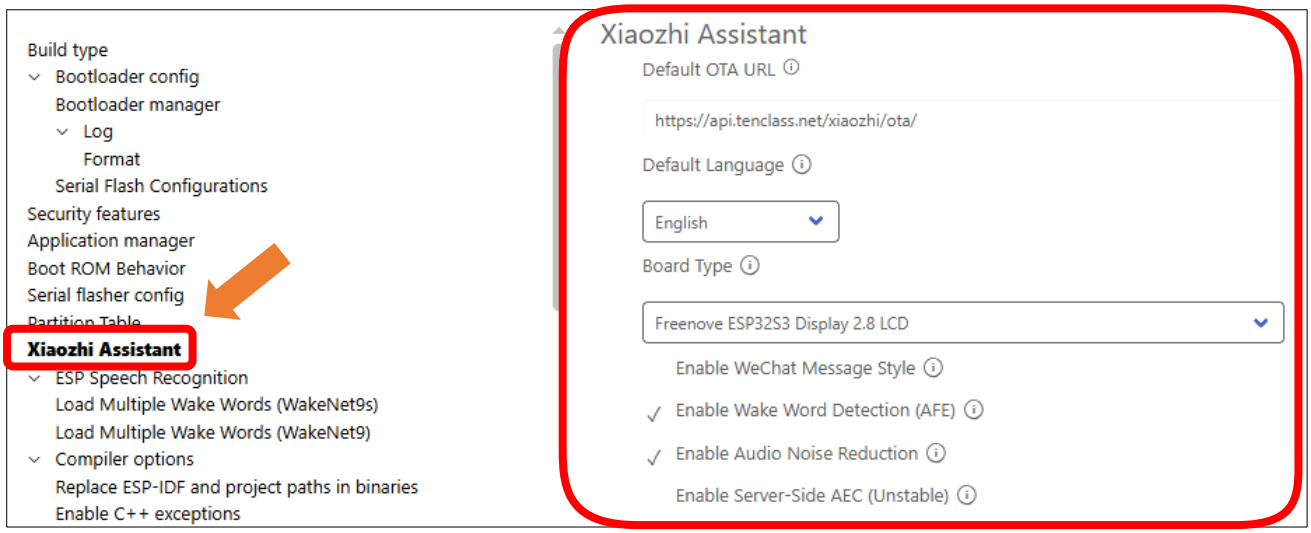
在新界面中，点击 '串行烧录器配置(Serial flasher config)'，并确认各项设置与下图所示配置一致。



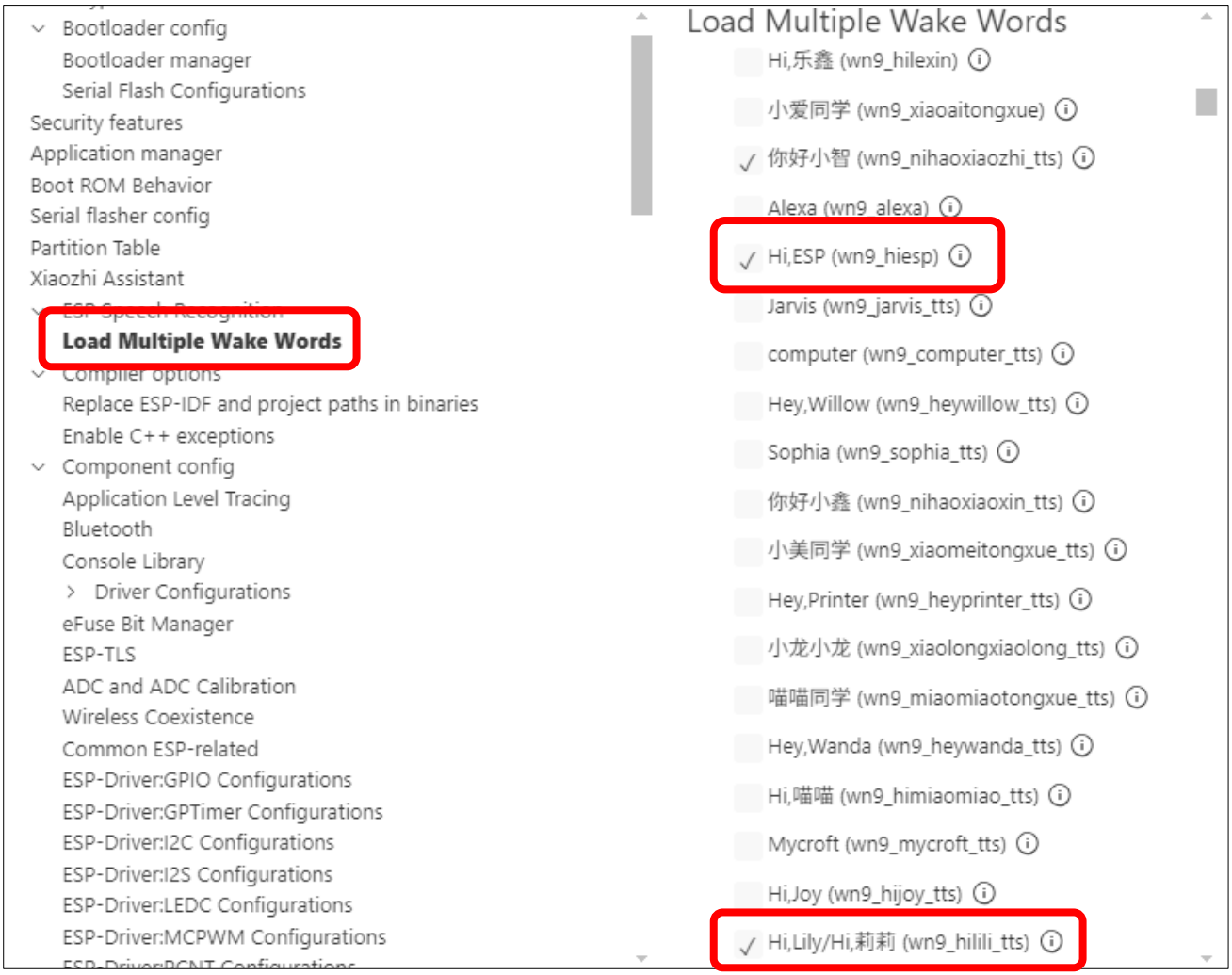
点击 “分区表(Partition Table)”，并确认其设置与下图所示配置一致。



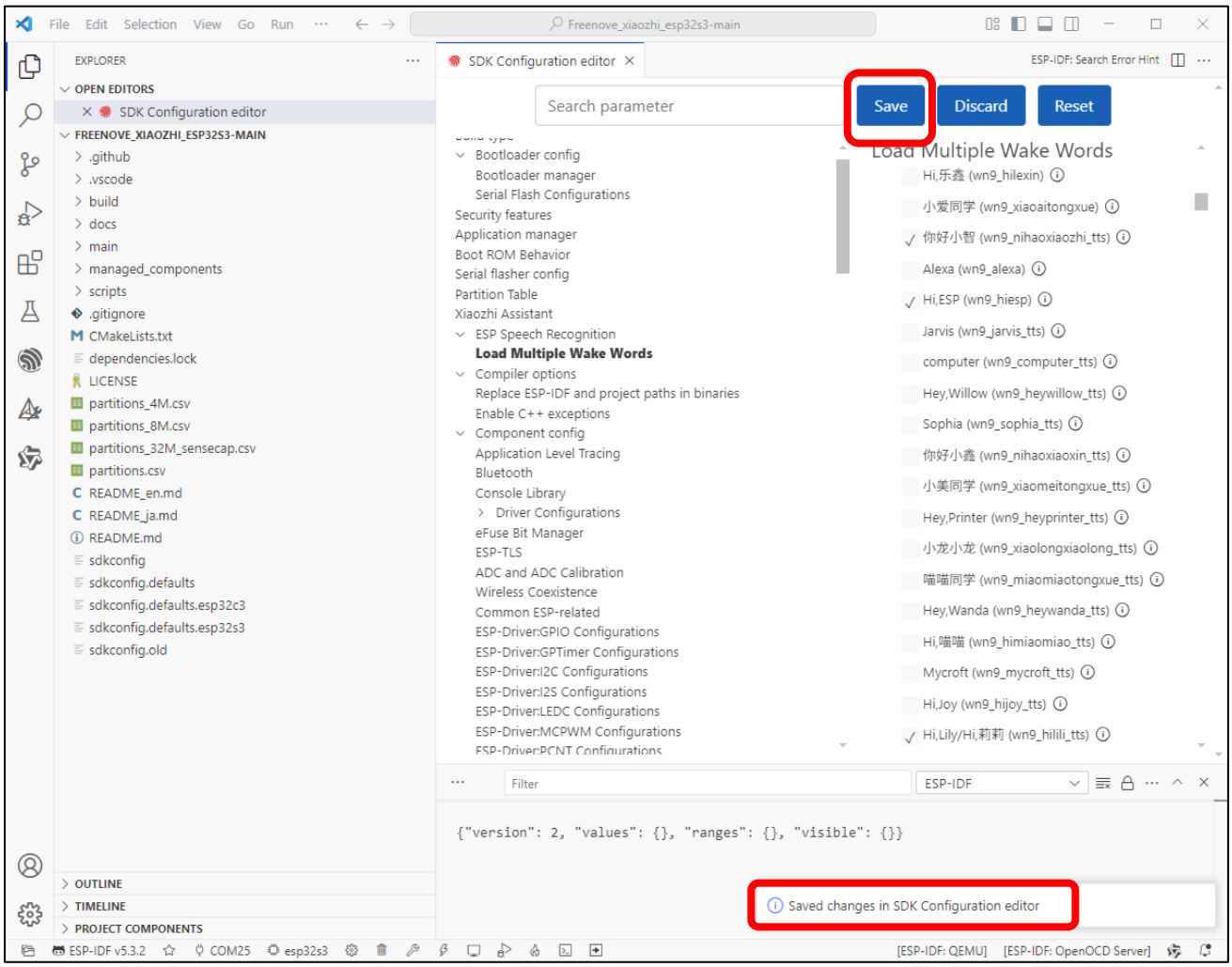
点击 **“Xiao Assistant”**，并确认其设置与下图所示配置一致。



点击 **“加载多个唤醒词(Load Multiple Wake Words)”**，并勾选 'Hi, ESP' 和 'Hi, Lily'（以及其他所需选项）的复选框。



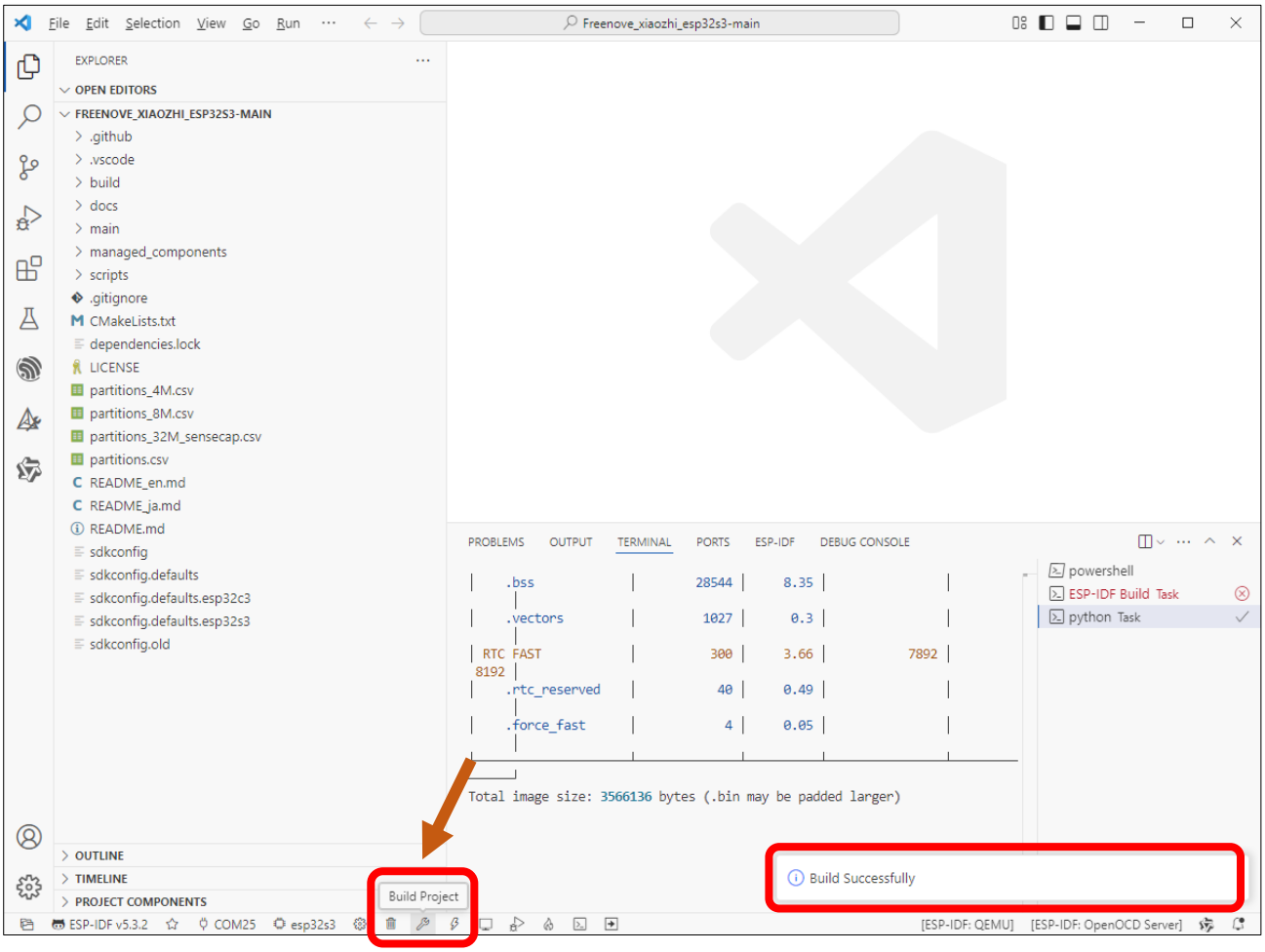
最后点击“保存(Save)”以存储配置，操作成功后底部将显示完成提示。



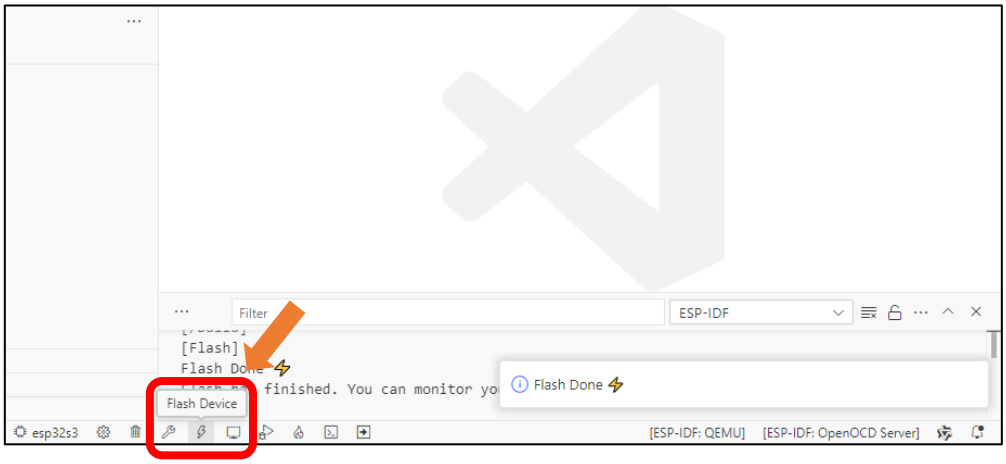
代码编译

开始编译前，请确保前文所述的所有配置均正确无误。点击底部工具栏中的 '全部清理(Full Clean)' 按钮以重置构建缓存。

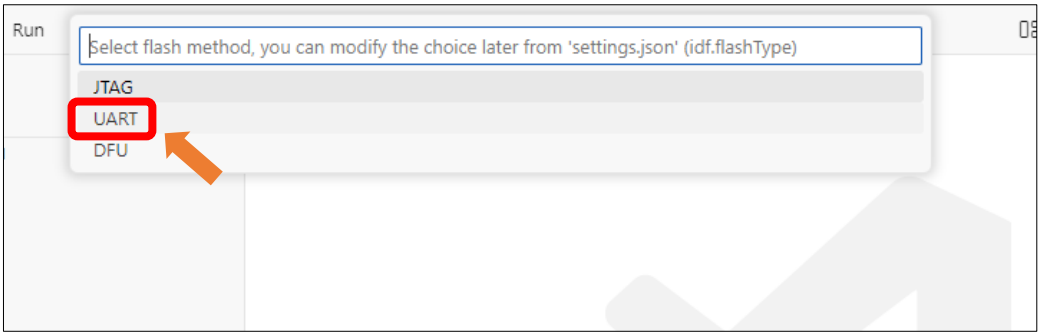
点击底部的 '构建项目(Build Project)' 开始编译整个工程。首次编译耗时可能较长，请耐心等待输出面板显示成功提示。



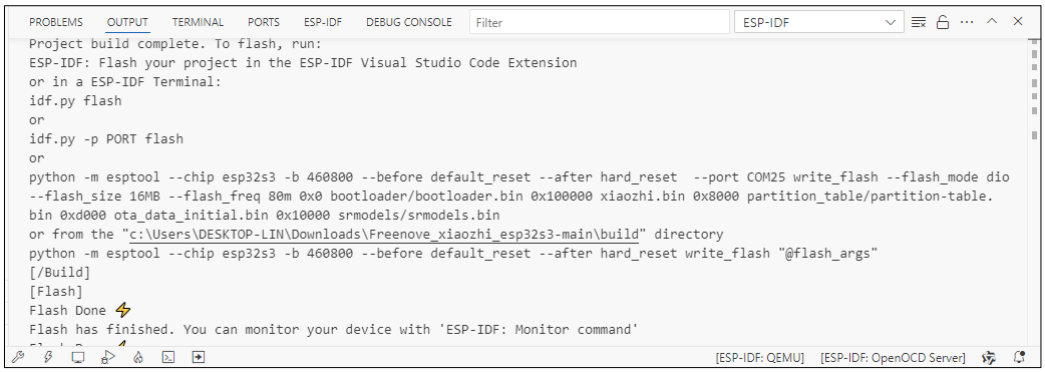
点击底部的 '烧录设备(Flash Device)' 开始将代码上传至 ESP32-S3 模块。



从新出现的选项菜单中，选择 'UART' 并等待代码上传完成。



当看到 '烧录完成。您可以使用"ESP-IDF: Monitor command"监控设备' 的提示时，即表示您已成功将 XiaoZhi AI 固件烧录至 ESP32-S3 模块。



至此，编译已完成，可进行二次开发。

本地服务器

免责声明

本项目基于开源仓库（MIT License）开发：https://github.com/xinnan-tech/xiaozhi-esp32-server。小智 AI 固件运行于虾哥的公司提供的服务器，我们仅对其进行了第三方学习与 AI 功能试用的适配，不涉及任何商业推广或实际应用。本教程仅供爱好者学习补充使用。

在本地服务器部署小智 AI

如果您不想使用小智 AI 官方服务器，也可以在本地计算机上搭建简化版服务端。本节我们将使用开源项目 https://github.com/xinnan-tech/xiaozhi-esp32-server 部署本地服务器，并与 ESP32-S3 建立连接。

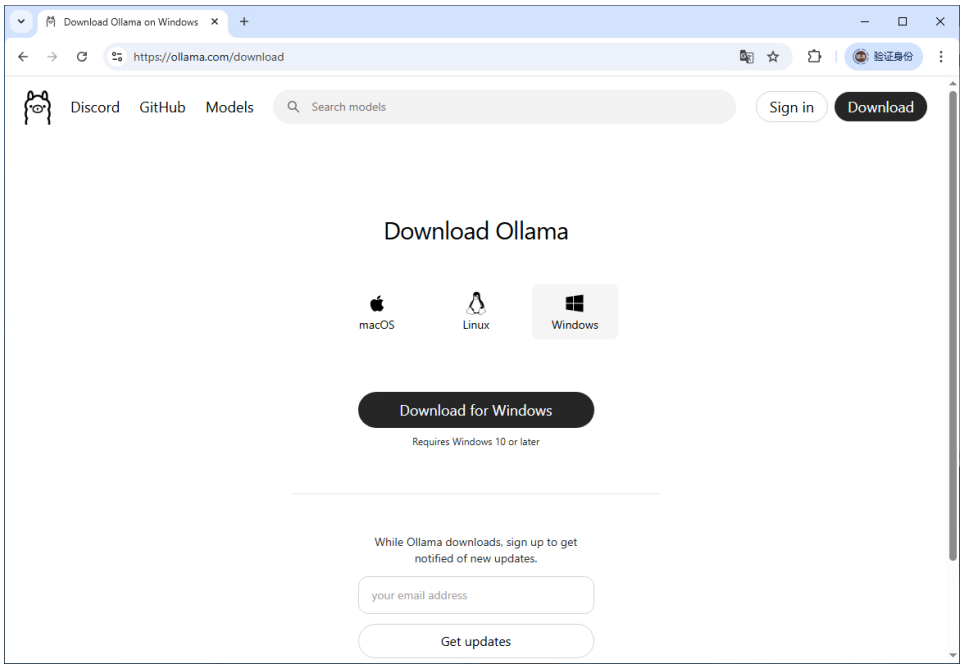
若使用过程中发现代码存在 bug，请前往 https://github.com/xinnan-tech/xiaozhi-esp32-server 提交 issue。请注意，我们对该项目并无深入了解，可能无法提供大量协助。

安装 Ollama

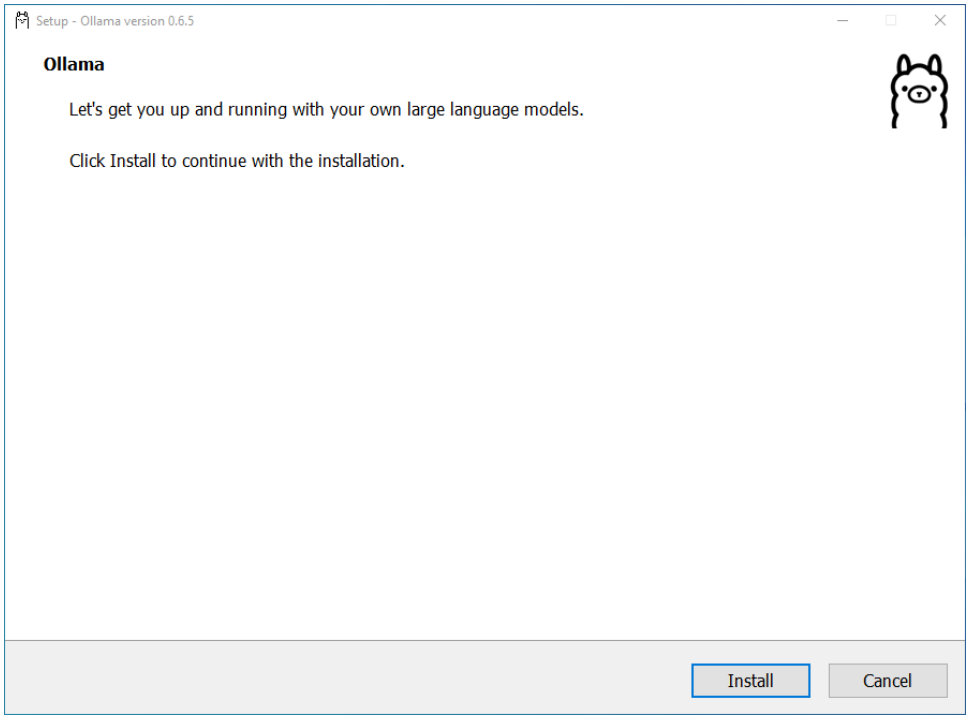
Windows

在开始之前，我们需要在本地安装 Ollama 工具，该工具可让我们在计算机上运行任何开源 AI 模型。

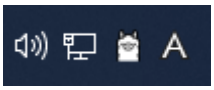
若您尚未安装 Ollama，请访问 https://ollama.com/download 下载并安装。



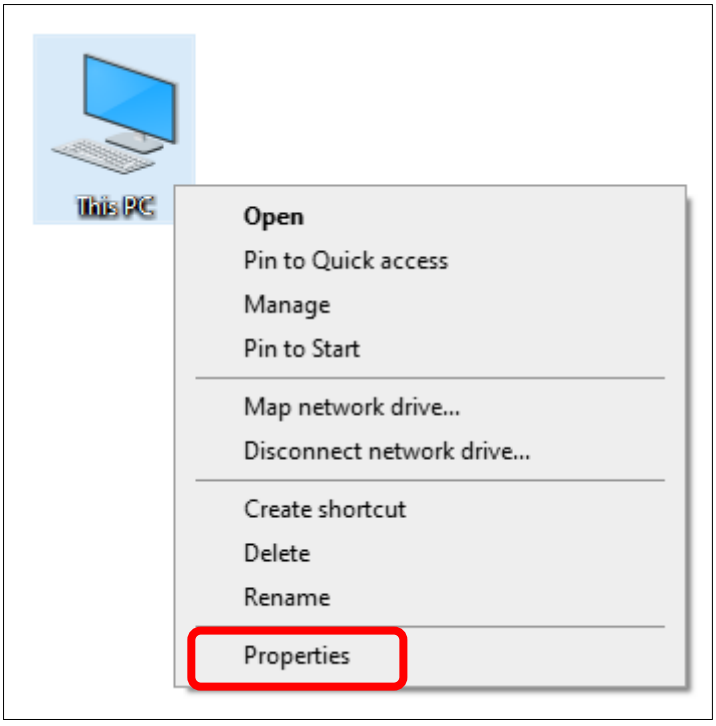
运行 Ollama 安装程序，点击安装。



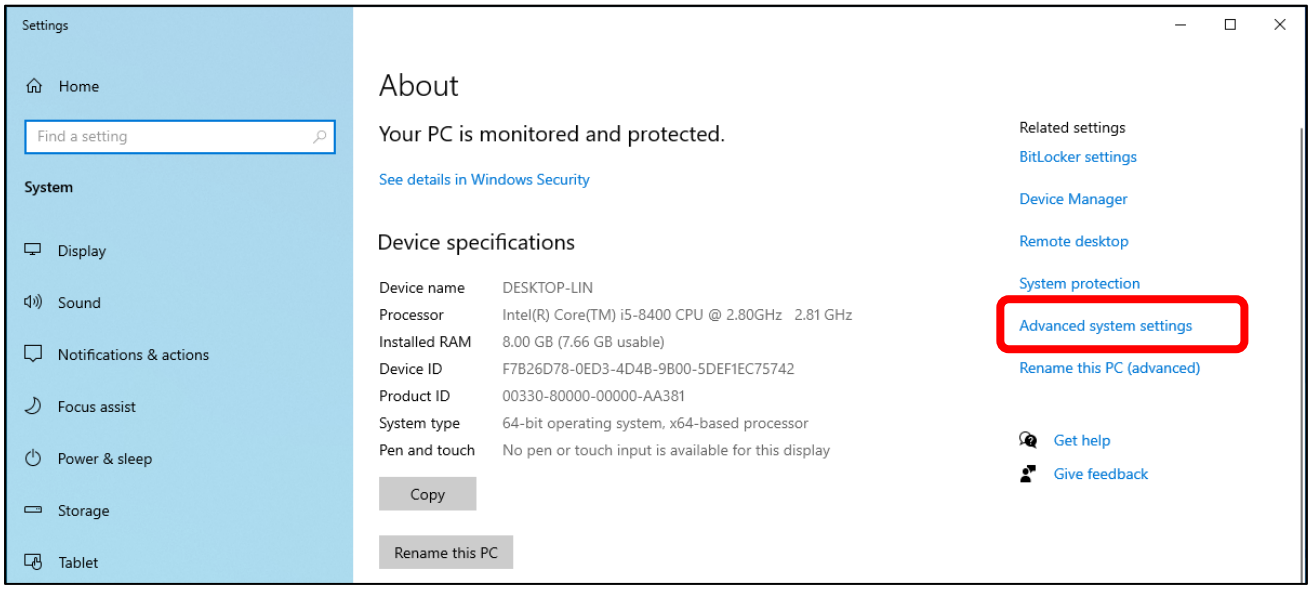
安装完成后，您将在任务栏中看到 Ollama 图标出现。



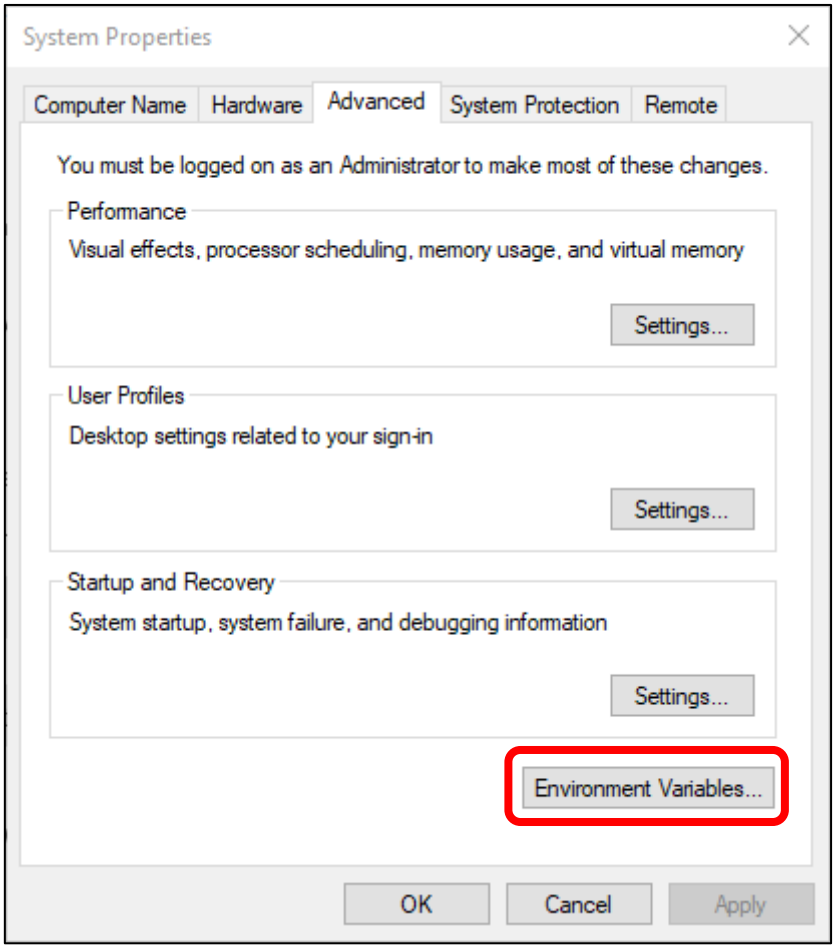
返回桌面，右键点击"此电脑"（或"我的电脑"），从上下文菜单中选择"属性"。



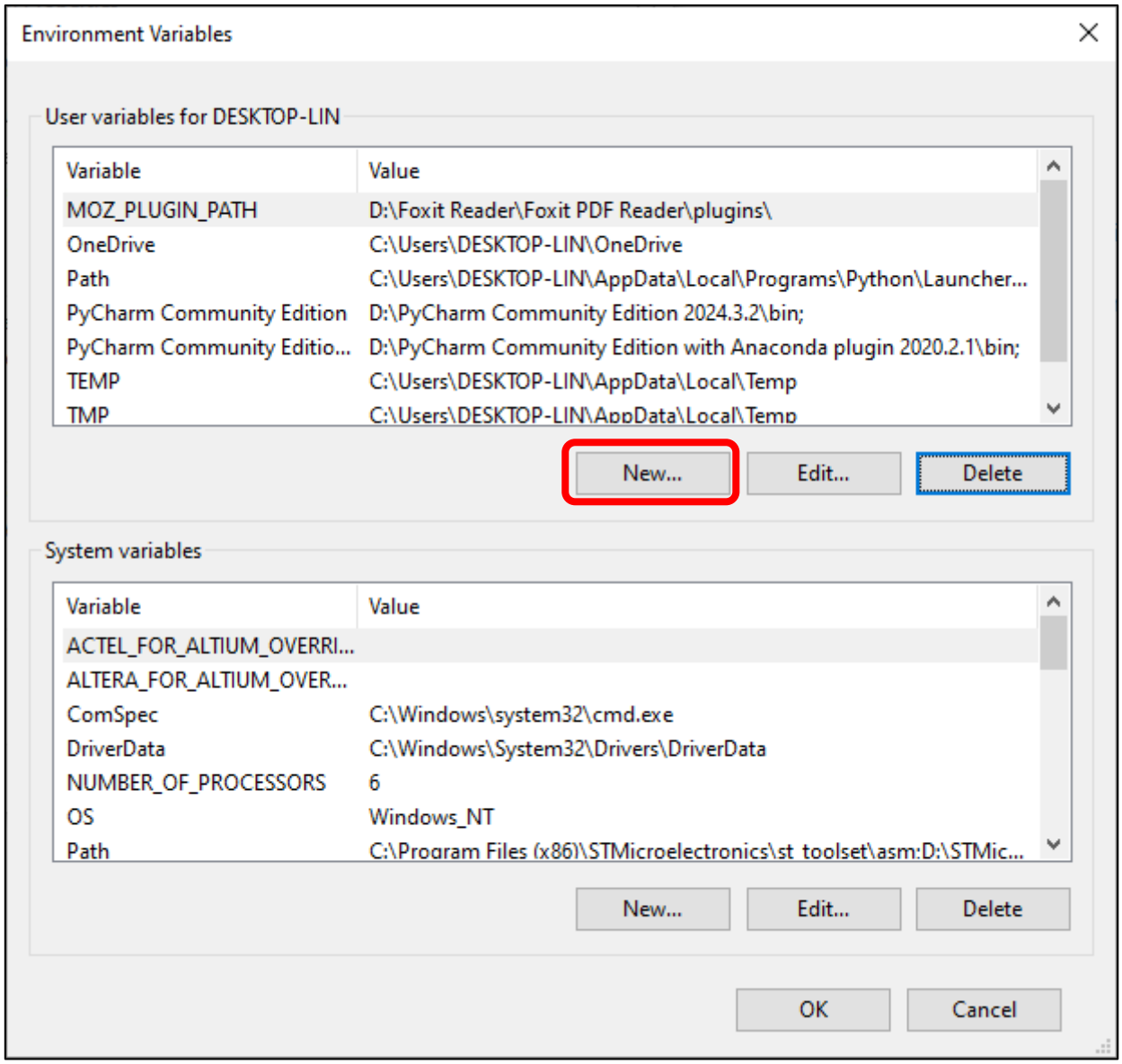
在打开的新窗口中，找到并点击"高级系统设置"。



在打开的新窗口中，点击"环境变量"。



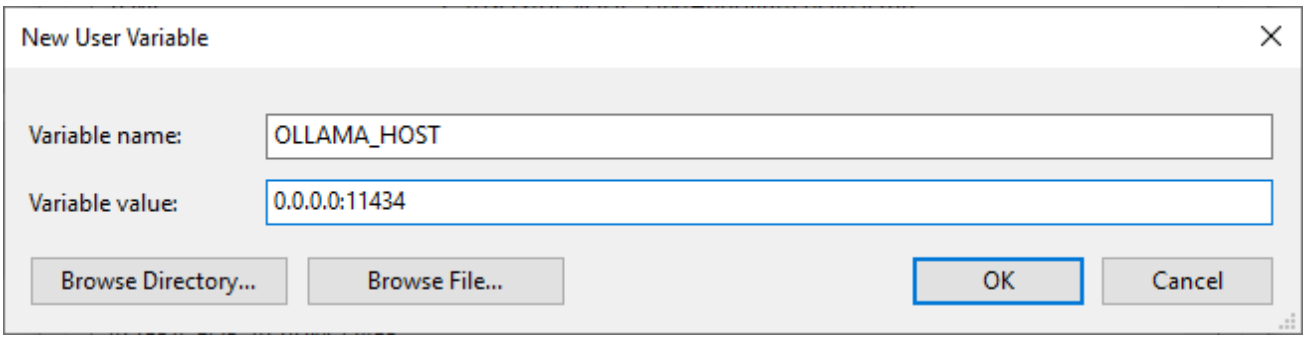
点击“新建”.



在变量名输入框填写：OLLAMA_HOST 在变量值输入框填写：0.0.0.0:11434 点击确定保存

通过此设置，您局域网内的所有设备均可通过本机 IP 地址访问 Ollama 服务。

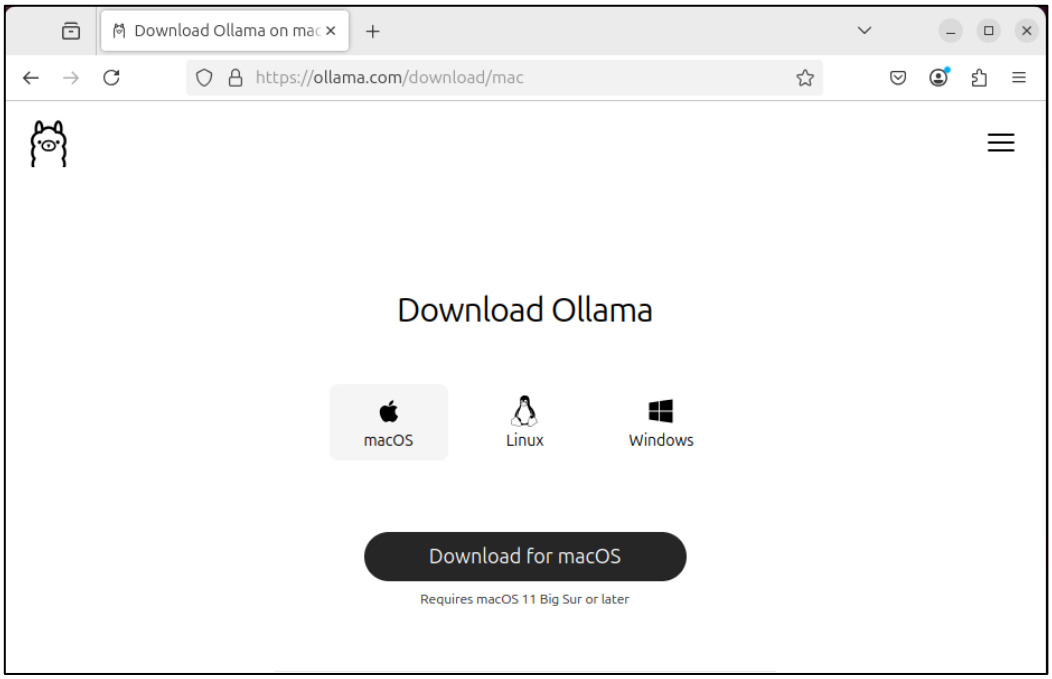
若未进行此配置，Ollama 将仅限本地主机访问（即仅安装该服务的计算机可用）。



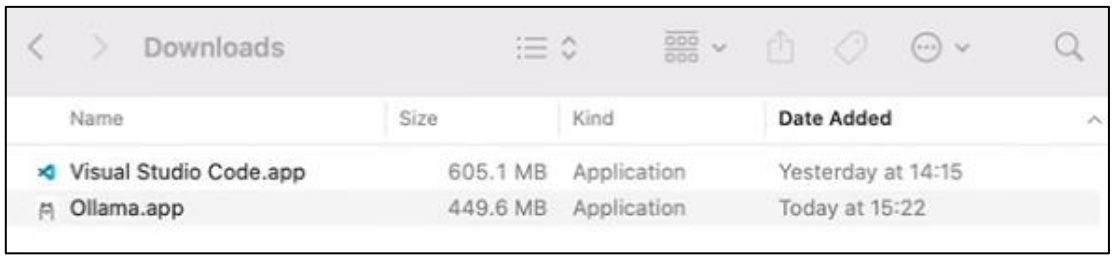
Mac OS

在开始前，我们需要在本地安装 Ollama 工具，这将允许我们在计算机上运行任何开源 AI 模型。

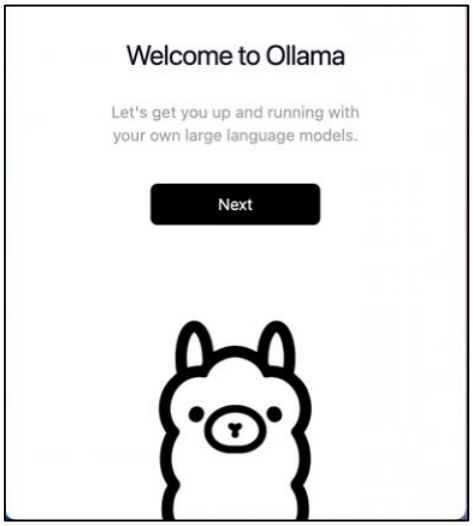
若您尚未安装 Ollama，请访问 https://ollama.com/download 下载并安装。

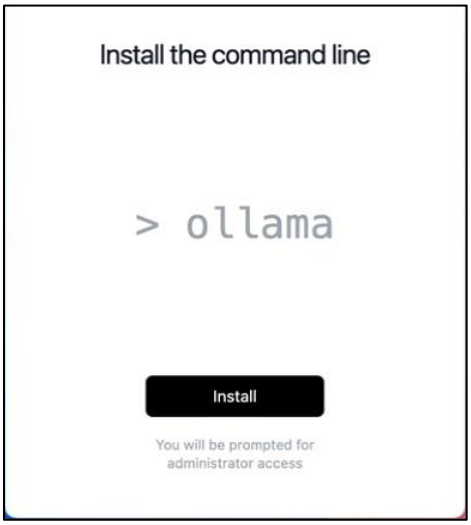


在"下载"文件夹中找到"Ollama.app"，双击打开该应用。



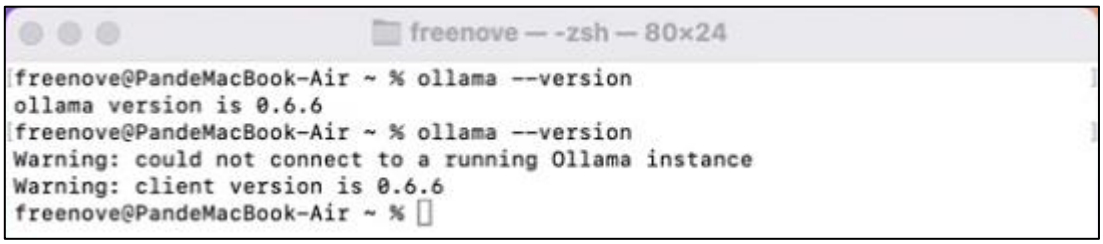
点击"下一步" → "安装" → "完成"。





安装完成后界面将自动关闭。
打开终端，使用指定命令检查 Ollama 是否安装成功：

```
ollama --version
```

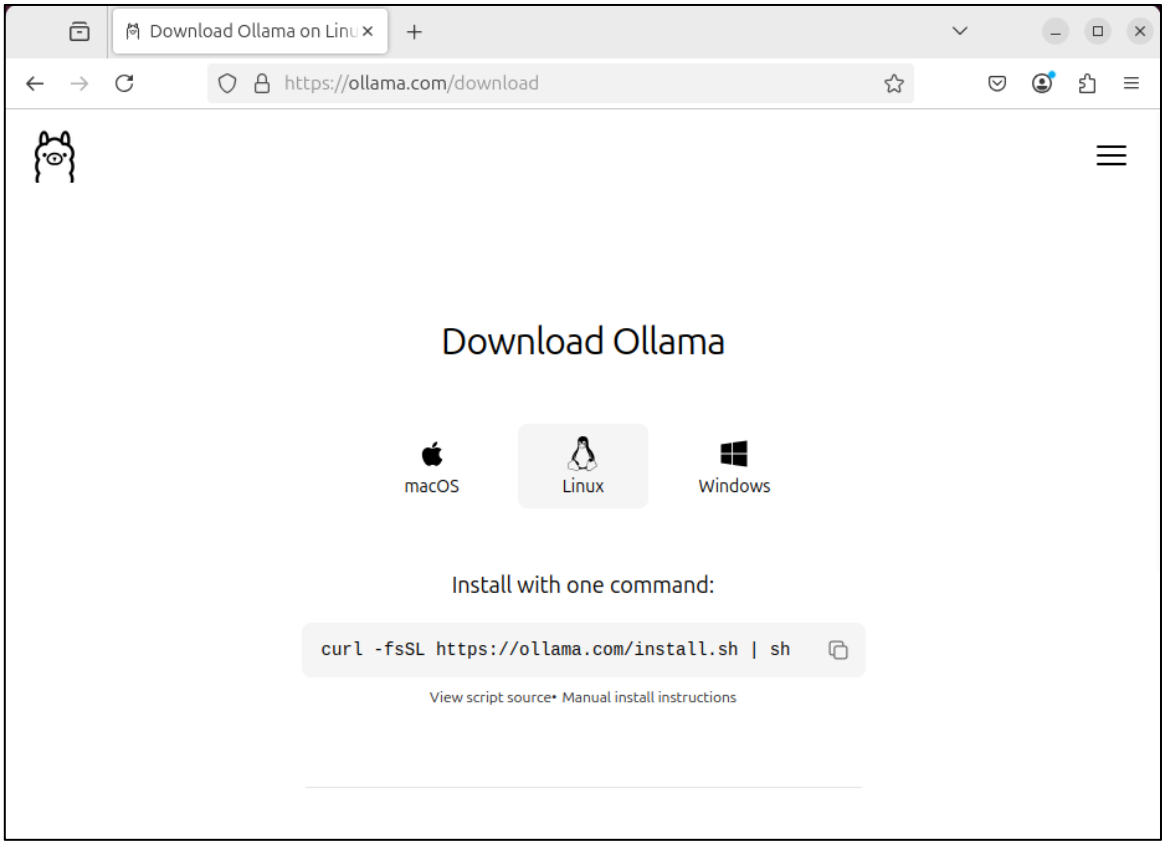


注意：Ollama 位于您的"应用程序"文件夹中。
如图所示：

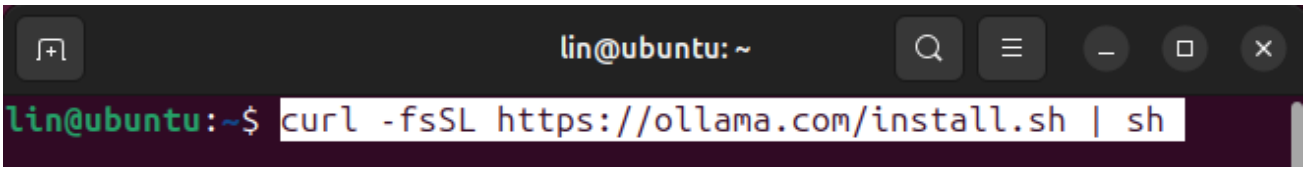
- 若 Ollama 已在运行，执行 ollama --version 将显示版本号
- 若 Ollama 未运行，该命令将返回连接错误（"无法连接到正在运行的 Ollama 实例"）

Linux

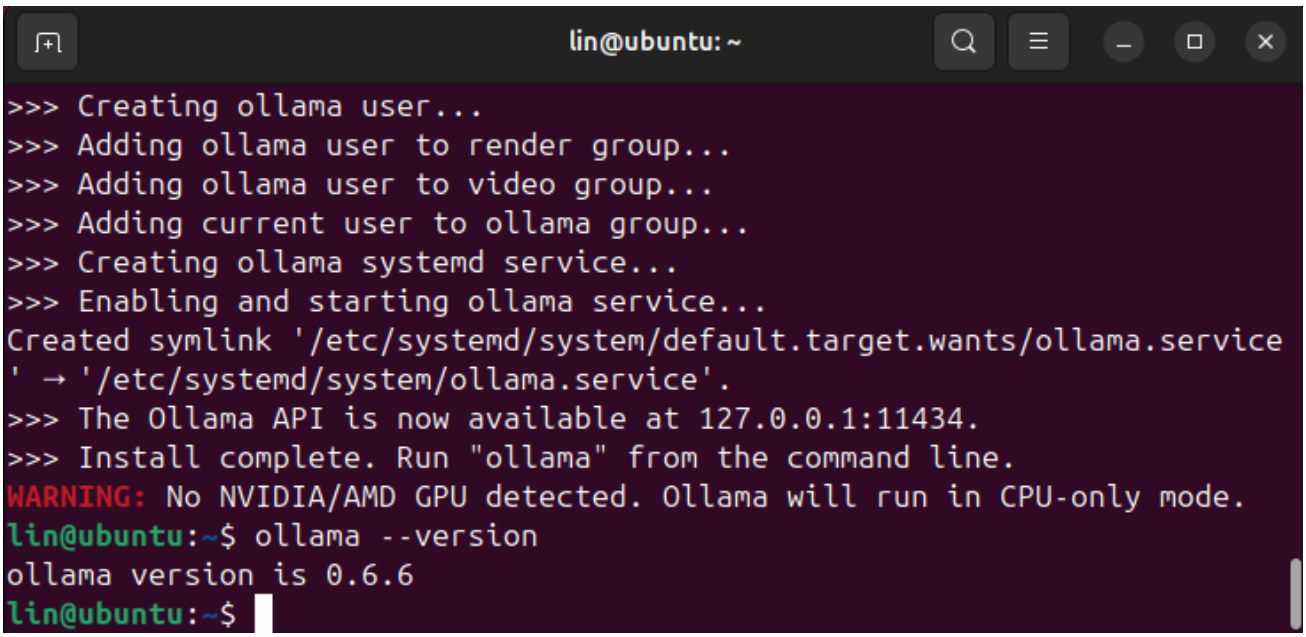
在开始前，我们需要在本地安装 Ollama 工具，这将允许我们在计算机上运行任何开源 AI 模型。
若您尚未安装 Ollama，请访问 https://ollama.com/download 下载并安装。
打开终端，运行以下命令安装 Ollama：



安装过程应如上图所示。您可以通过运行以下命令验证 Ollama 是否安装成功:

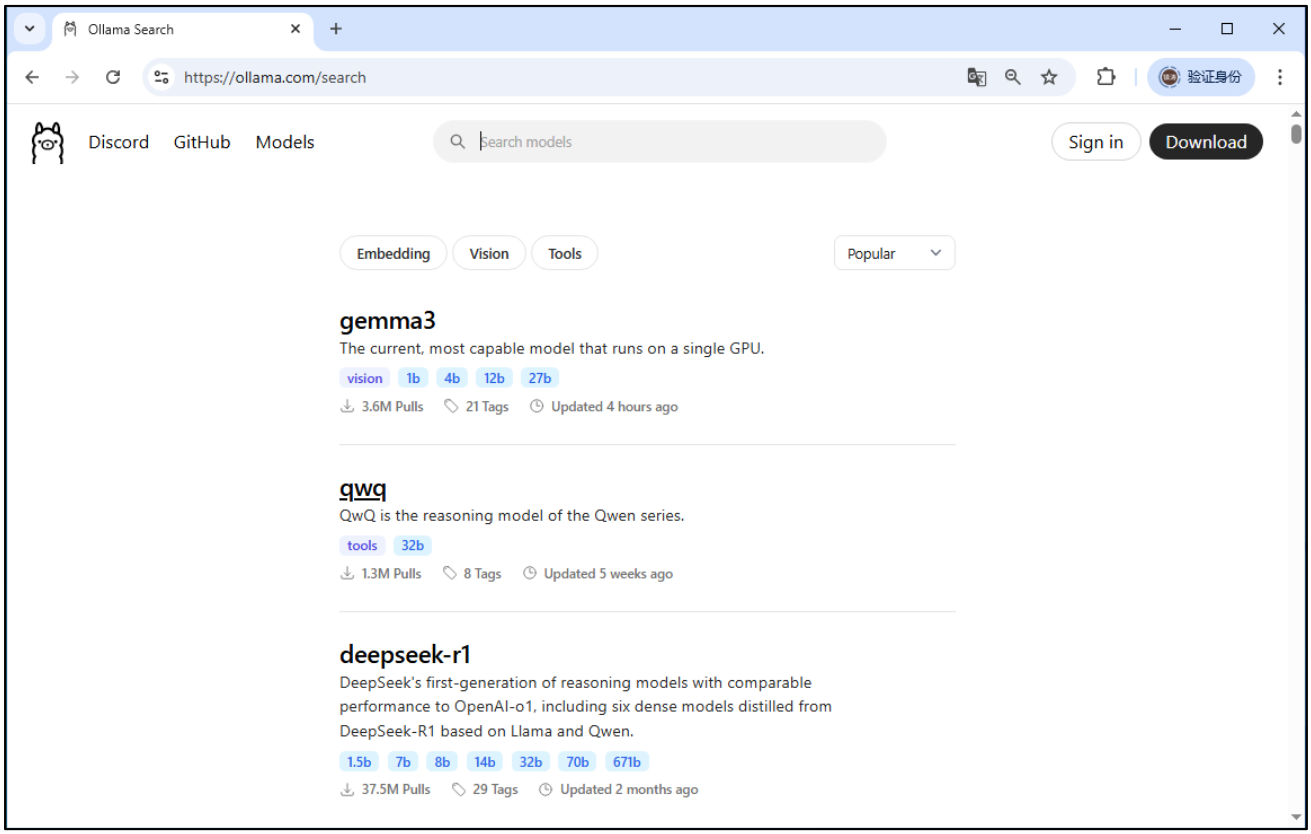


安装完成后应如上图所示。您可以通过运行以下命令验证 Ollama 是否安装成功: **ollama --version**

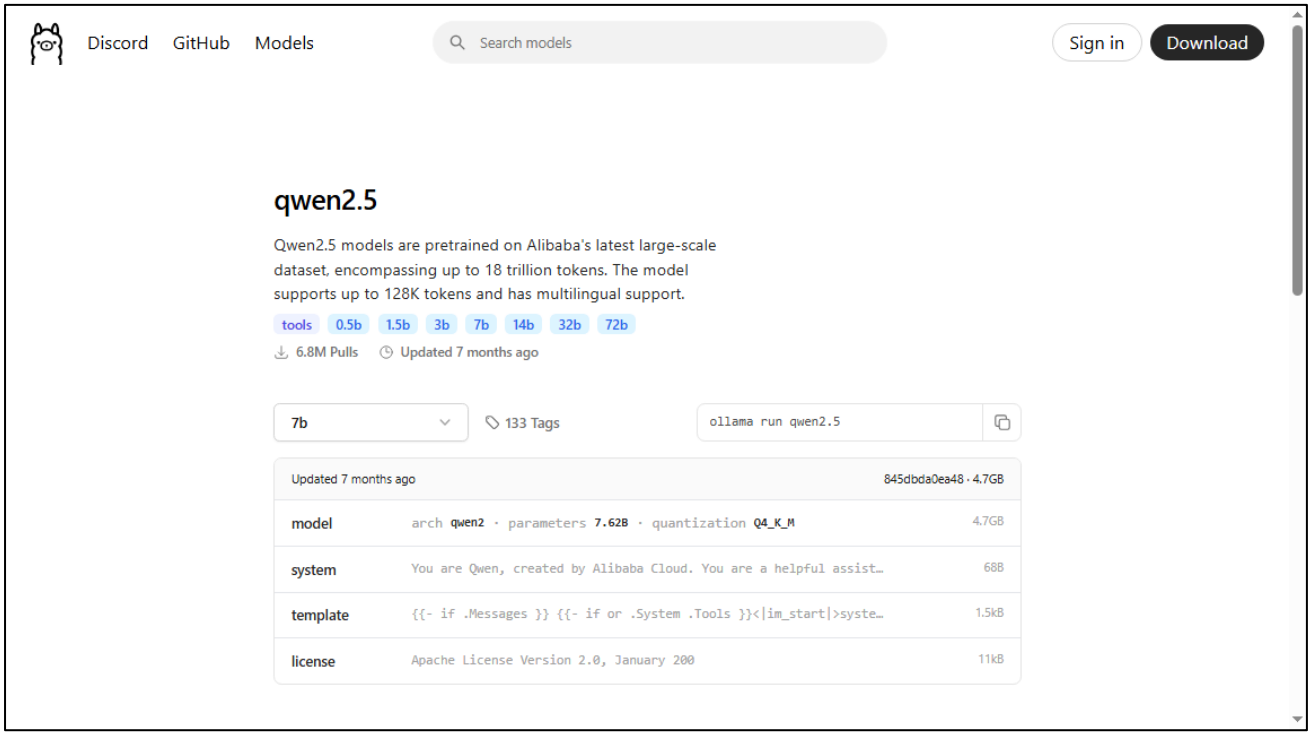


LLM Model

请访问 https://ollama.com/search 并选择适合您计算机配置或您偏好的 LLM 大语言模型。

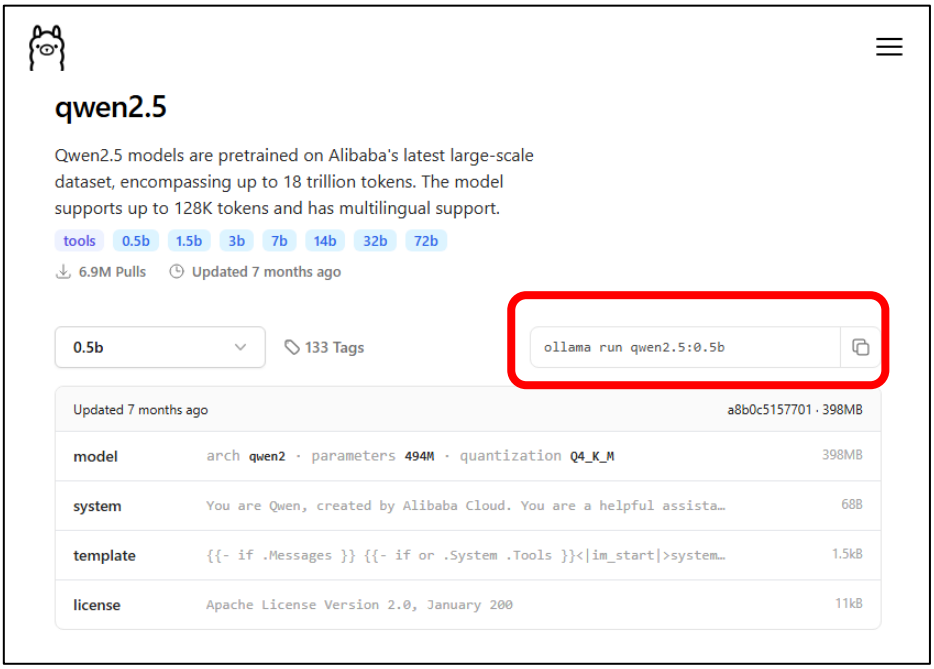


这里我们以 qwen2.5 模型为例，点击"qwen2.5"模型。



请注意，在选择模型时，您需要根据计算机的 GPU 显存或 CPU 内存配置选择合适的模型规格。
1、参数规模更大的模型具备更高智能水平，而较小模型则智能水平相对较低
2、高端配置设备（高性能 GPU/CPU 且内存充足）建议选择大模型以获得最佳表现；低配置设备（GPU/CPU 内存有限）应选用小模型以保证运行流畅
3、若在性能不足的系统上选择过大的模型，可能导致无法加载或推理速度极其缓慢
您可以通过下拉菜单选择适配的模型参数规模。

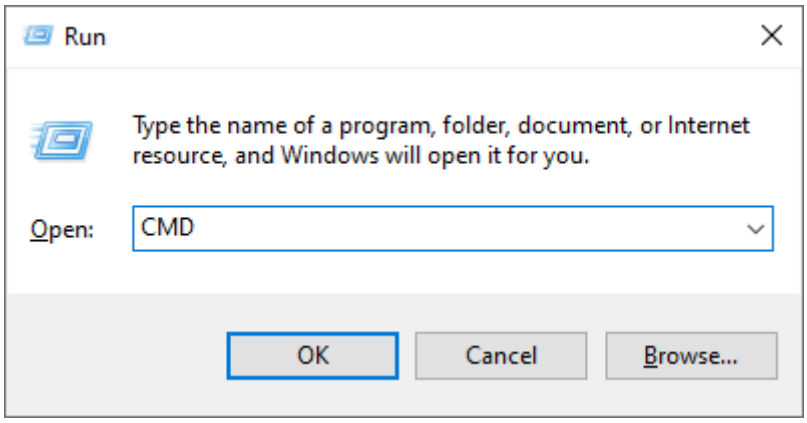
较小规模的模型能力稍弱但运行更快。本次演示我们将以 qwen2.5:0.5b 模型为例。
请从网页复制以下命令：
ollama run qwen2.5:0.5b



请选择与您操作系统兼容的版本，安装您首选的大语言模型(LLM)。

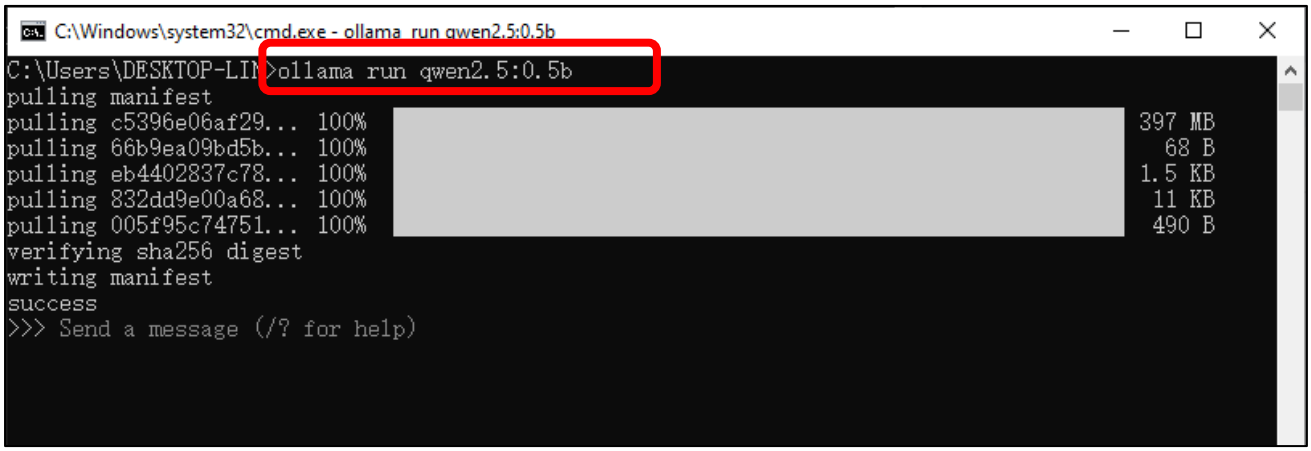
Windows

请使用快捷键"Win+R"，在弹出的窗口中输入"CMD"，即可打开命令提示符界面。

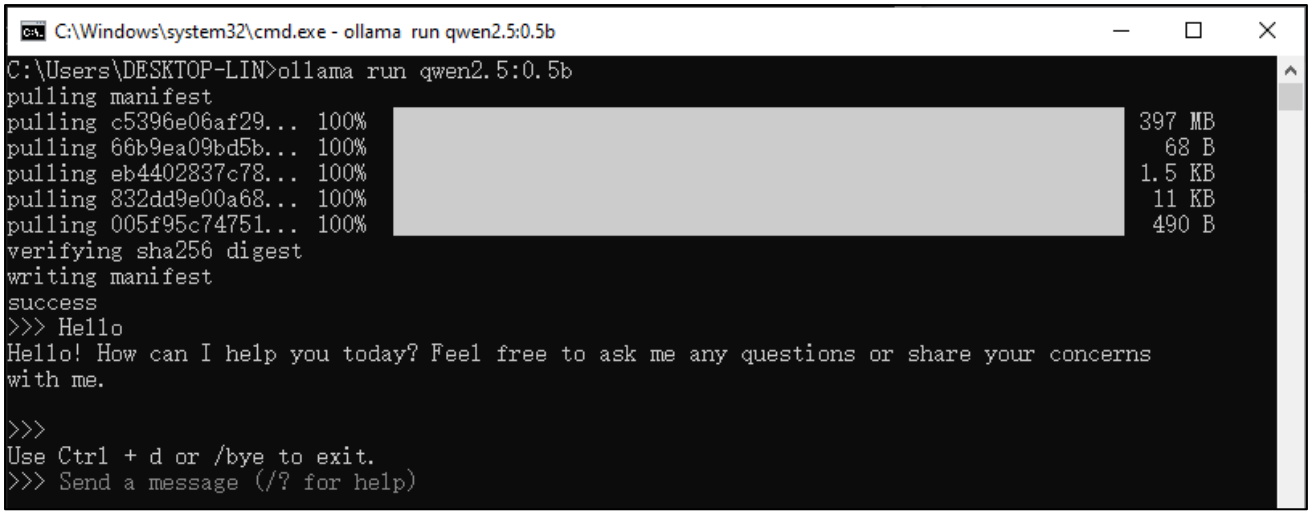


运行命令 **“ollama --version”** 以验证是否已成功安装 ollama。

输入命令 **“ollama run qwen2.5:0.5b”** 将模型下载至本地计算机。



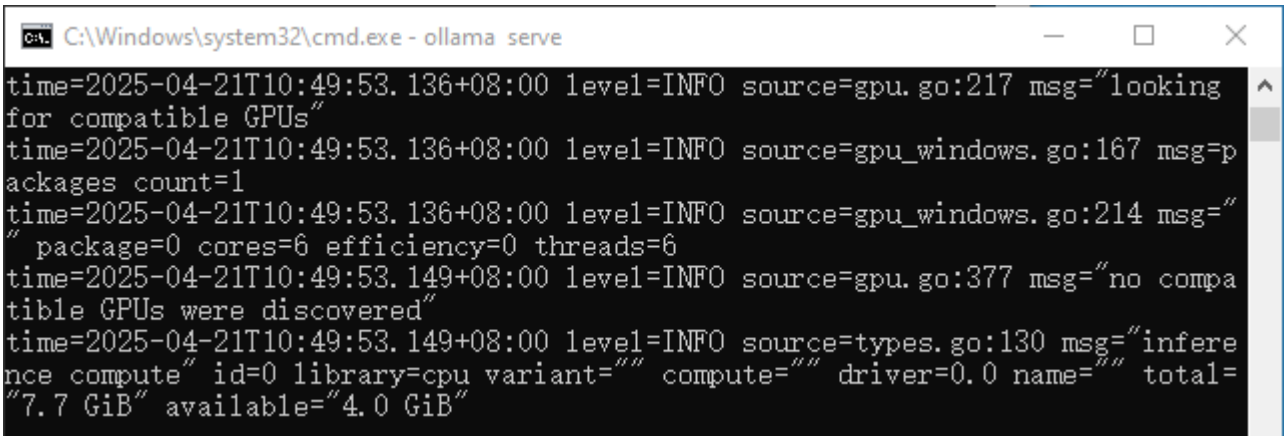
安装完成后，您可直接在命令提示符（CMD）界面与 Qwen2.5-0.5B 进行对话。



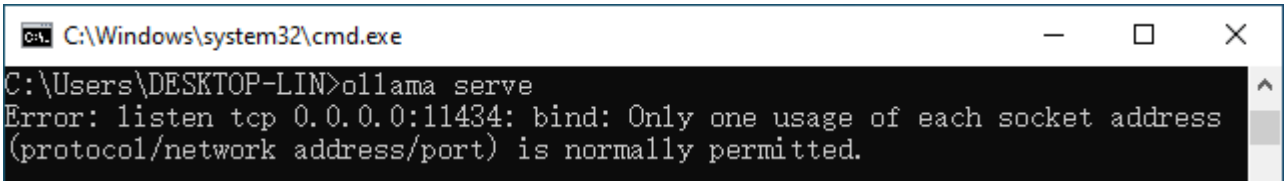
您可通过以下方式退出或启动服务:

1、按下 Ctrl+D 组合键退出对话模式

2、运行命令 **“ollama serve”** 启动服务端

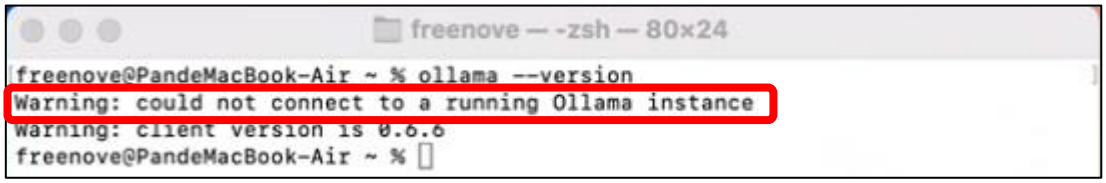


若 Ollama 已在运行（系统任务栏显示其图标），再次执行 ollama serve 将引发错误。这两种方式实际启动的是同一服务。



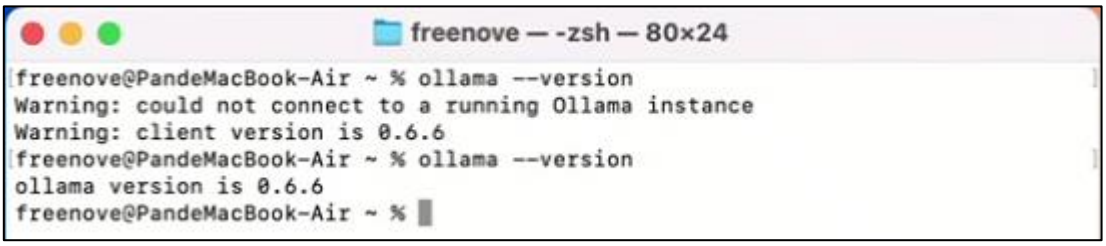
Mac OS

在终端中运行命令 **“ollama --version”** 以检测 Ollama 是否已安装成功。

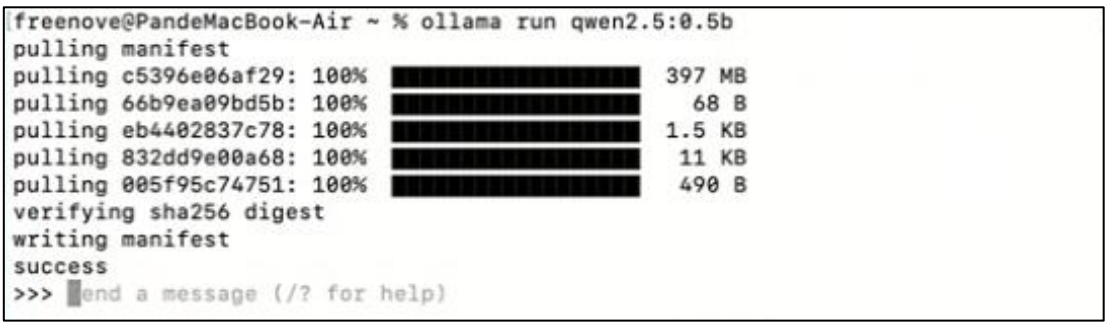


若出现提示 “Warning: could not connect to a running Ollama instance”，表明 Ollama 服务未启动。请前往应用程序目录手动运行。

请在终端重新检测服务运行状态。



在终端中运行 **“ollama run qwen2.5:0.5b”** 以将模型安装至本地计算机。



安装完成后，您可以直接在终端界面与 Qwen2.5-0.5B 进行对话。

```
freenove@PandeMacBook-Air ~ % ollama run qwen2.5:0.5b
pulling manifest
pulling c5396e06af29: 100% ▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕ 397 MB
pulling 66b9ea09bd5b: 100% ▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕  68 B
pulling eb4402837c78: 100% ▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕ 1.5 KB
pulling 832dd9e00a68: 100% ▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕  11 KB
pulling 005f95c74751: 100% ▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕▕ 490 B
verifying sha256 digest
writing manifest
success
>>> Hello
Hello! How can I assist you today? Let me know if there's anything
specific you'd like to discuss or any questions that need help with. I'm
here to provide information and answer queries in a way that's easy for
you to understand.

>>>
Use Ctrl + d or /bye to exit.
>>> Send a message (/? for help)
```

您可以通过按下“Ctrl+D”退出。

您可以通过运行命令“ollama serve”来启动 Ollama 服务器。

```
freenove — ollama serve — 80×24
freenove@PandeMacBook-Air ~ % ollama serve
2025/04/24 17:10:39 routes.go:1232: INFO server config env="map[HTTPS_PROXY: HTT
P_PROXY: NO_PROXY: OLLAMA_CONTEXT_LENGTH:2048 OLLAMA_DEBUG:false OLLAMA_FLASH_AT
TENTION:false OLLAMA_GPU_OVERHEAD:0 OLLAMA_HOST:http://127.0.0.1:11434 OLLAMA_KE
EP_ALIVE:5m0s OLLAMA_KV_CACHE_TYPE: OLLAMA_LLM_LIBRARY: OLLAMA_LOAD_TIMEOUT:5m0s
 OLLAMA_MAX_LOADED_MODELS:0 OLLAMA_MAX_QUEUE:512 OLLAMA_MODELS:/Users/freenove/.
ollama/models OLLAMA_MULTIUSER_CACHE:false OLLAMA_NEW_ENGINE:false OLLAMA_NOHIST
ORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL:0 OLLAMA_ORIGINS:[http://loca
lhost https://localhost http://localhost:* https://localhost:* http://127.0.0.1
https://127.0.0.1 http://127.0.0.1:* https://127.0.0.1:* http://0.0.0.0 https://
0.0.0.0 http://0.0.0.0:* https://0.0.0.0:* app://* file://* tauri://* vscode-web
view://* vscode-file://*] OLLAMA_SCHED_SPREAD:false http_proxy: https_proxy: no_
proxy:]"
time=2025-04-24T17:10:39.898+08:00 level=INFO source=images.go:458 msg="total bl
obs: 5"
time=2025-04-24T17:10:39.899+08:00 level=INFO source=images.go:465 msg="total un
used blobs removed: 0"
time=2025-04-24T17:10:39.899+08:00 level=INFO source=routes.go:1299 msg="Listeni
ng on 127.0.0.1:11434 (version 0.6.6)"
time=2025-04-24T17:10:39.900+08:00 level=INFO source=types.go:130 msg="inference
 compute" id="" library=cpu variant="" compute="" driver=0.0 name="" total="8.0
GiB" available="3.2 GiB"
```

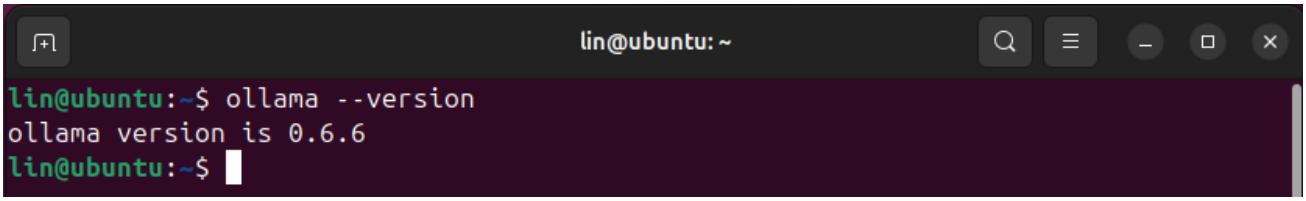
如果 Ollama 已在运行，您将看到以下信息。

```
freenove — -zsh — 80×24
freenove@PandeMacBook-Air ~ % ollama serve
Error: listen tcp 127.0.0.1:11434: bind: address already in use
freenove@PandeMacBook-Air ~ %
```

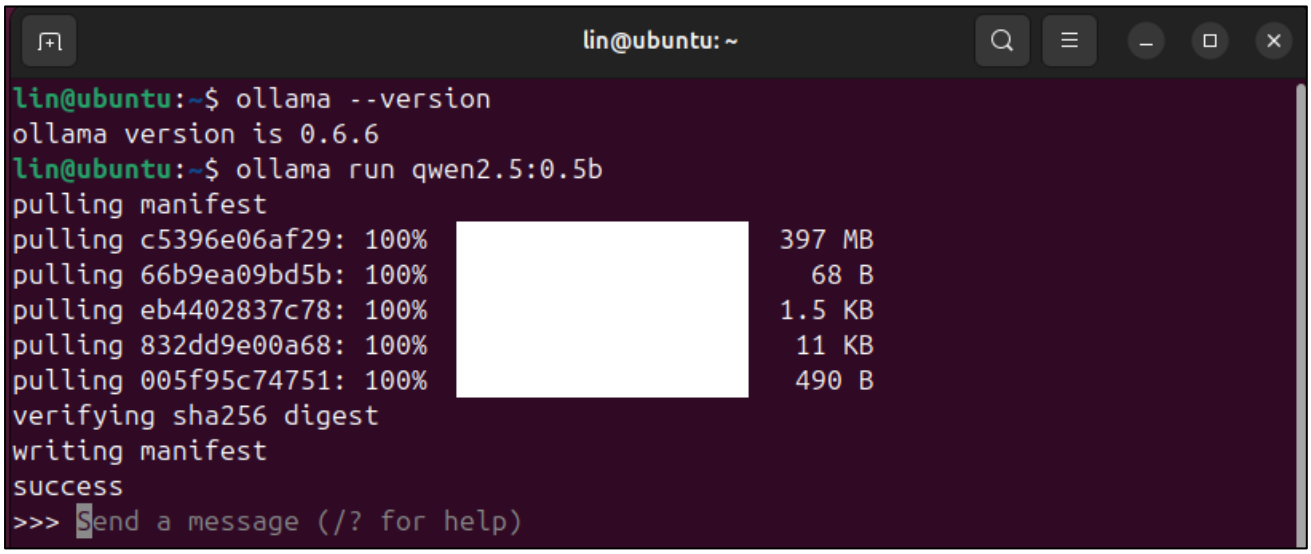
要访问 Ollama 的用户指南，请运行命令 Ollama。

Linux

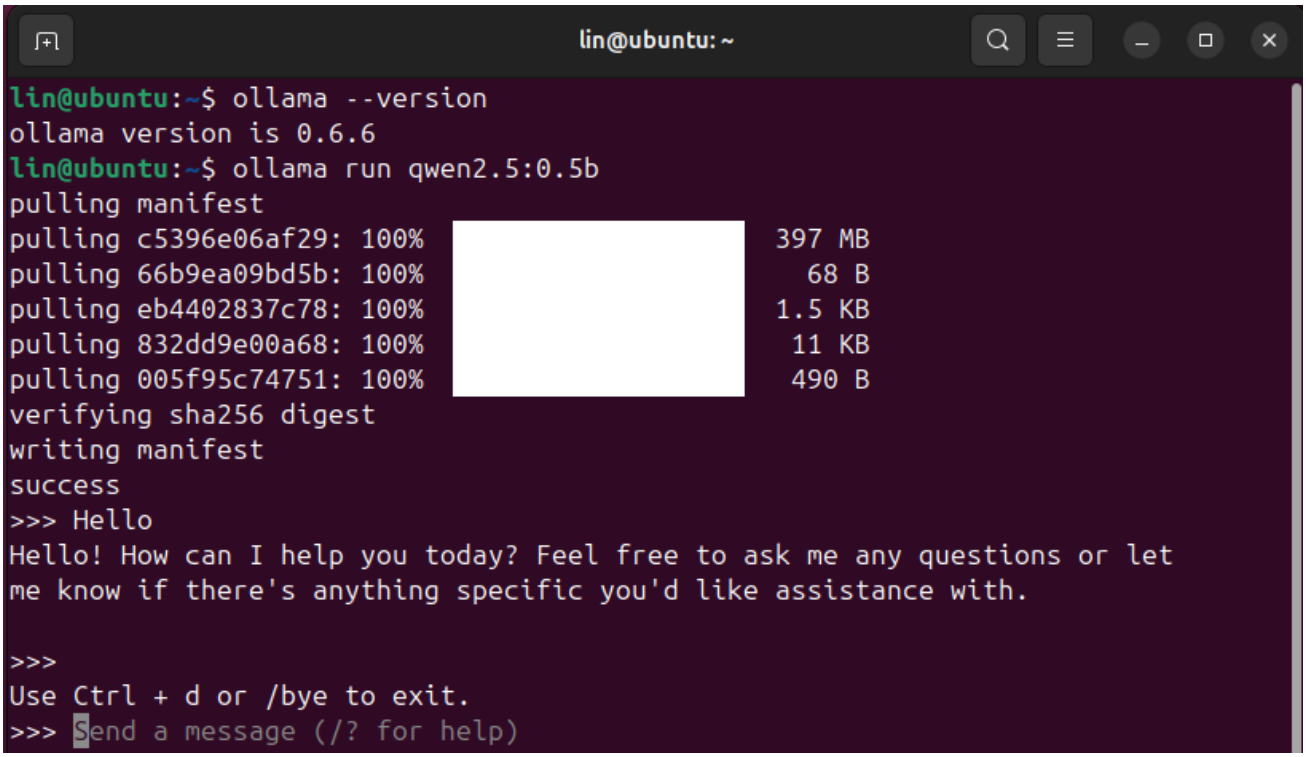
在终端运行命令 **“ollama --version”** 以检查 Ollama 是否已安装。



运行命令 **“ollama run qwen2.5:0.5b”** 将模型下载到您的计算机。

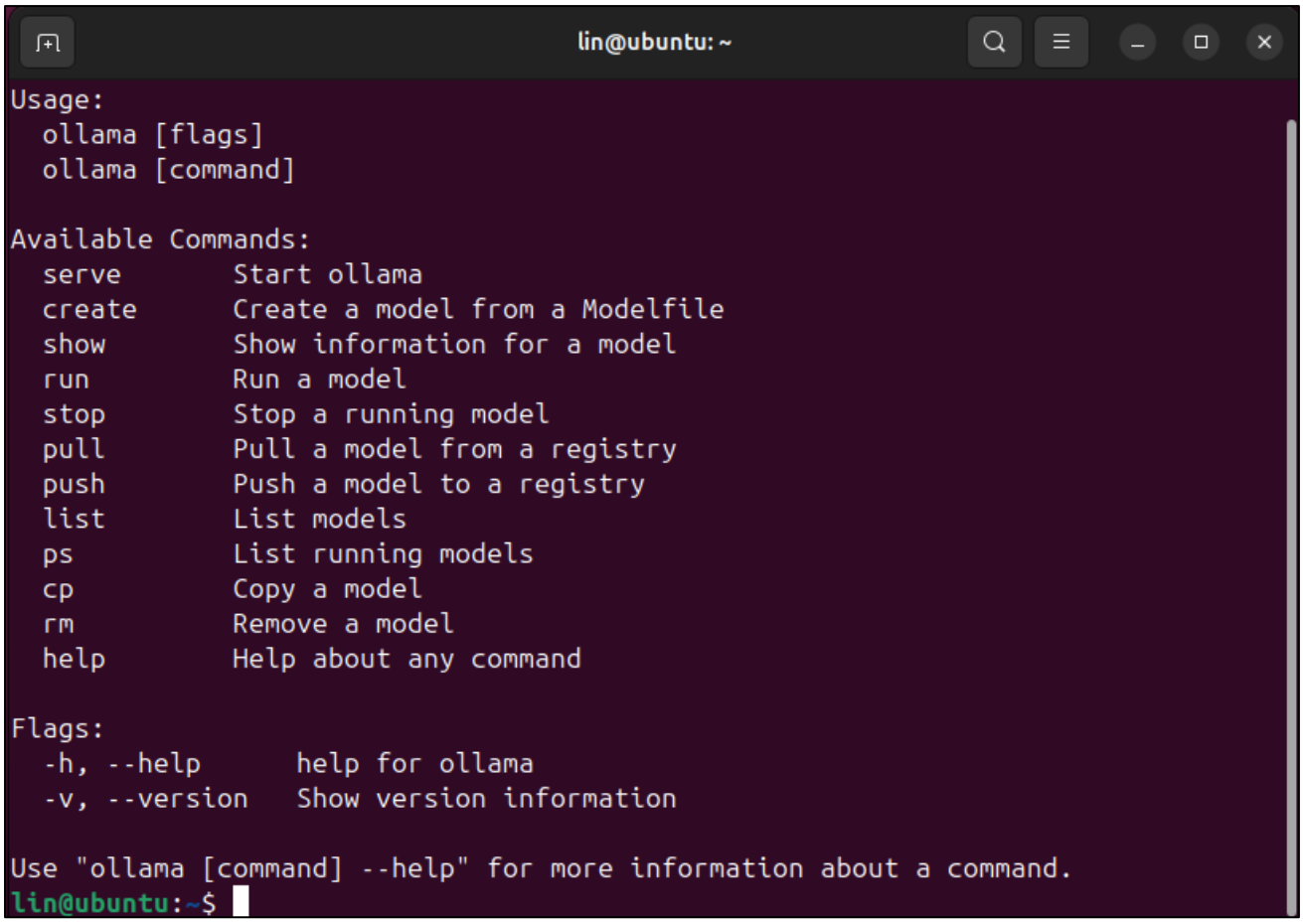


安装完成后，您可以直接在终端界面与 Qwen2.5-0.5B 进行对话。



按 Ctrl+D 即可退出。

要查看 Ollama 用户指南，请运行命令 ollama。

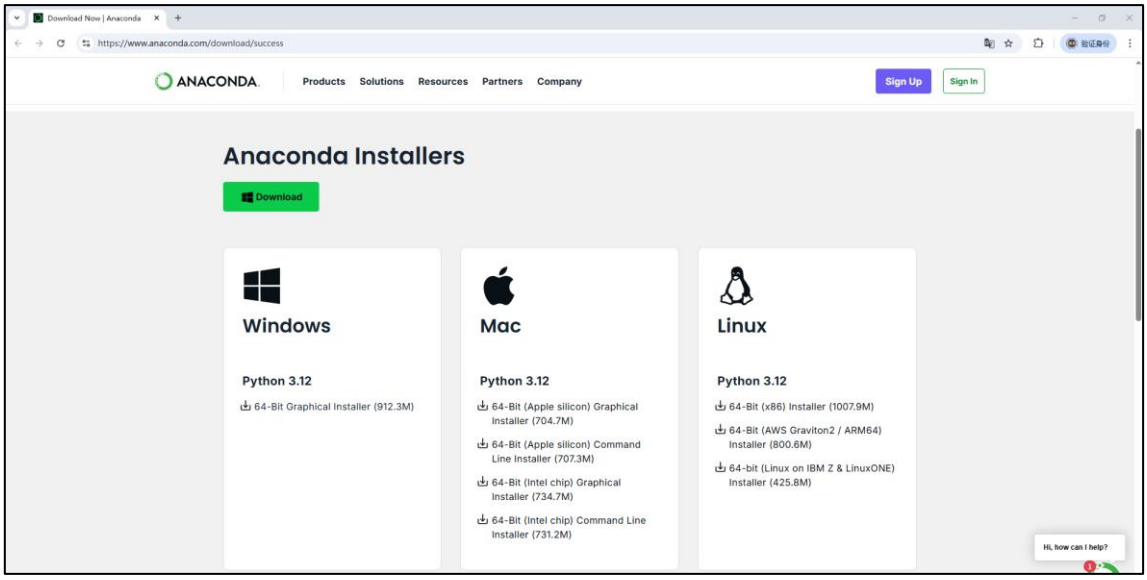


安装 Conda

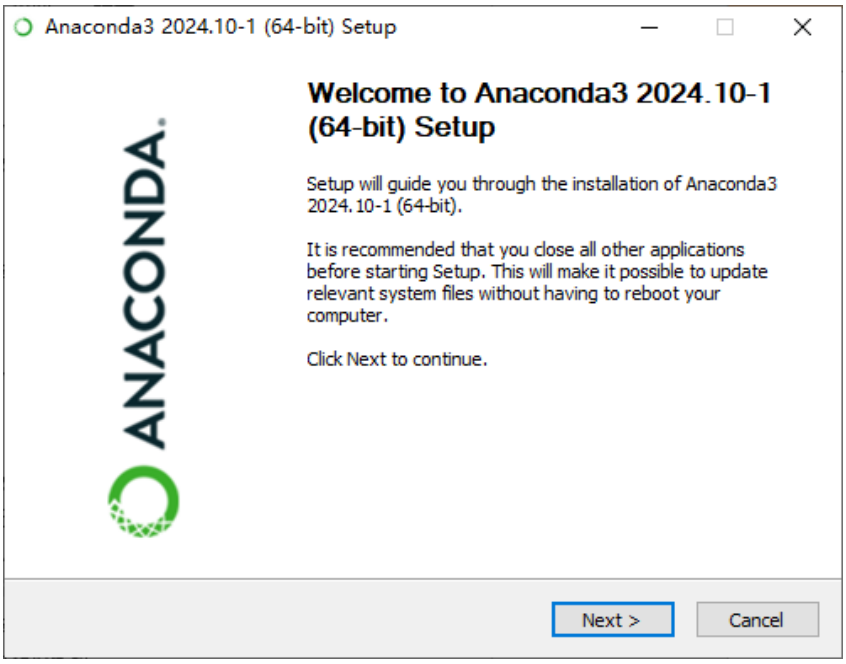
小智-ESP32 服务器开源项目提供四种安装方式。本教程将演示最简单的配置示例，其他使用方法请参考项目官网进一步探索。

Windows

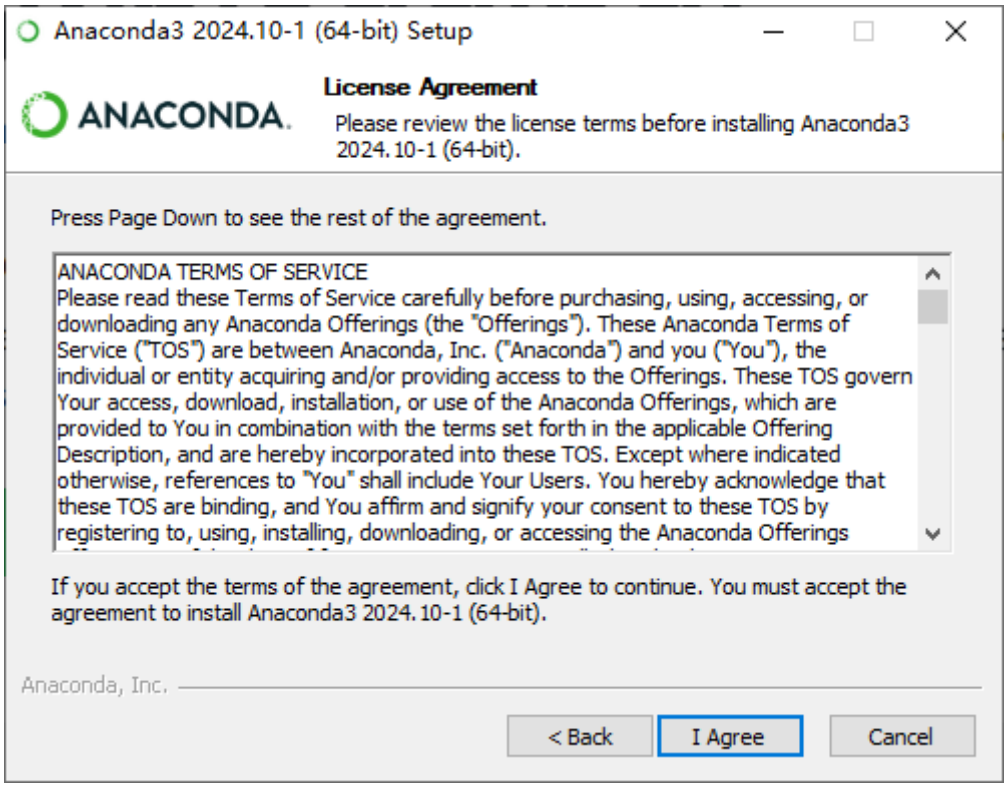
本示例使用 Conda 进行依赖项管理，因此您需要提前在系统中安装 Conda。若尚未安装，可从以下地址下载安装：https://www.anaconda.com/download/success
请选择适合您操作系统的安装程序。
Miniconda 是 Anaconda 提供的轻量级安装工具，已预配置支持 Anaconda 资源库。



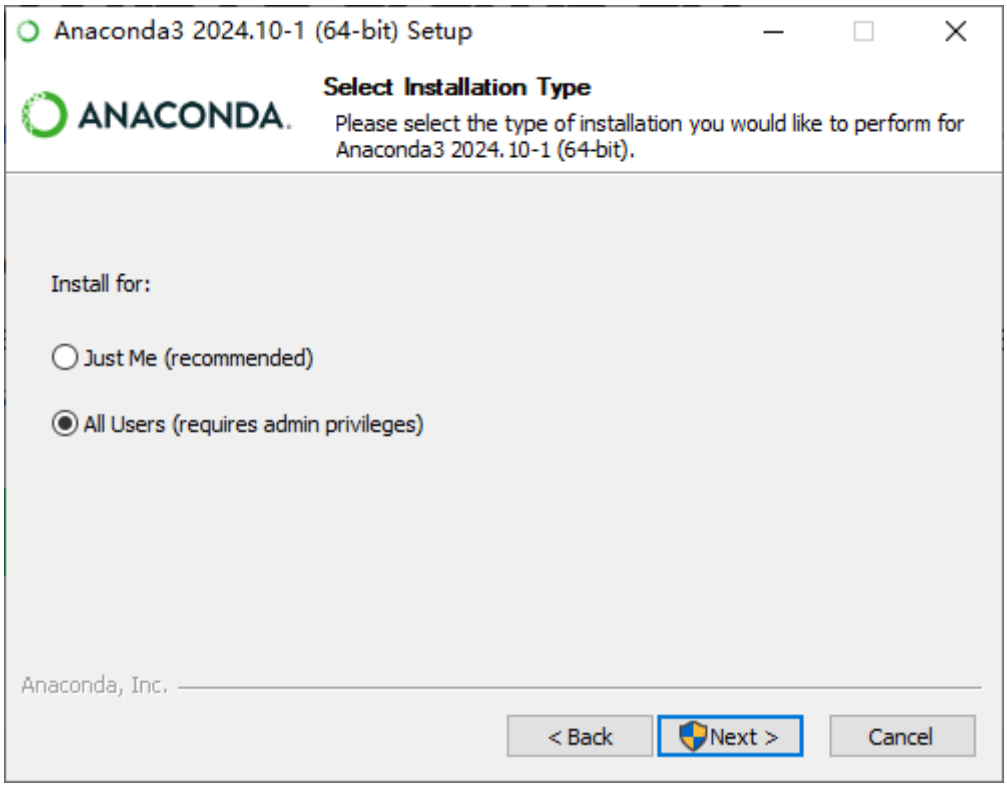
此处我们以 Windows 系统为例。双击下载的 Conda 安装程序，点击 下一步 继续。



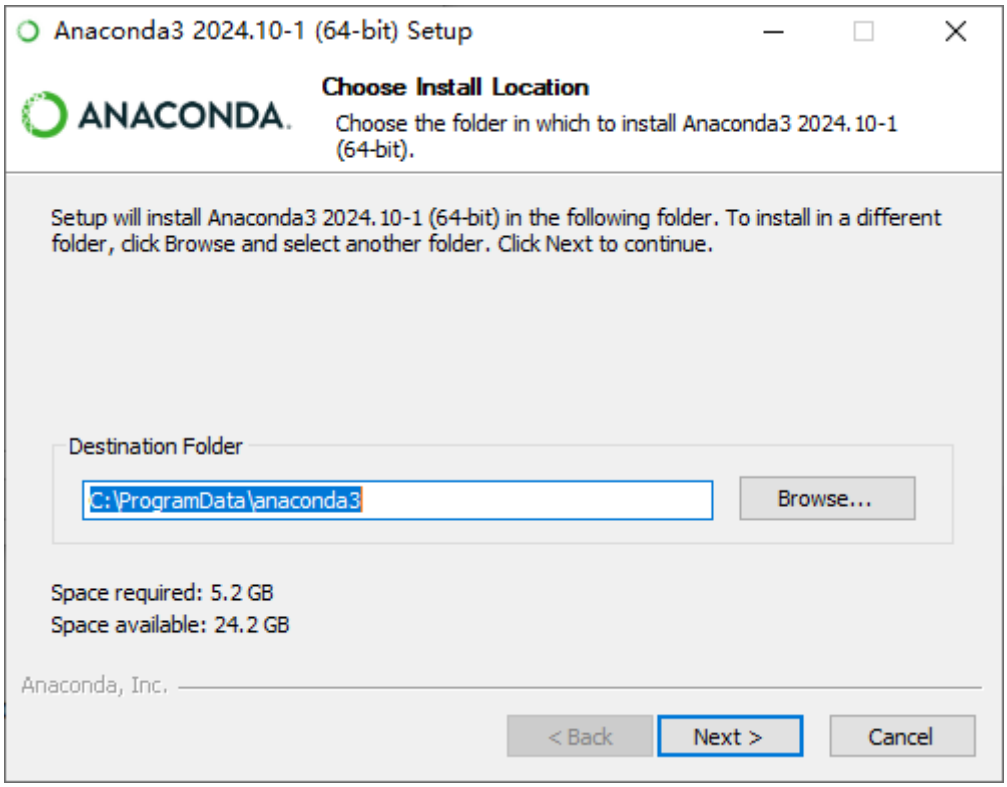
点击“我同意”.



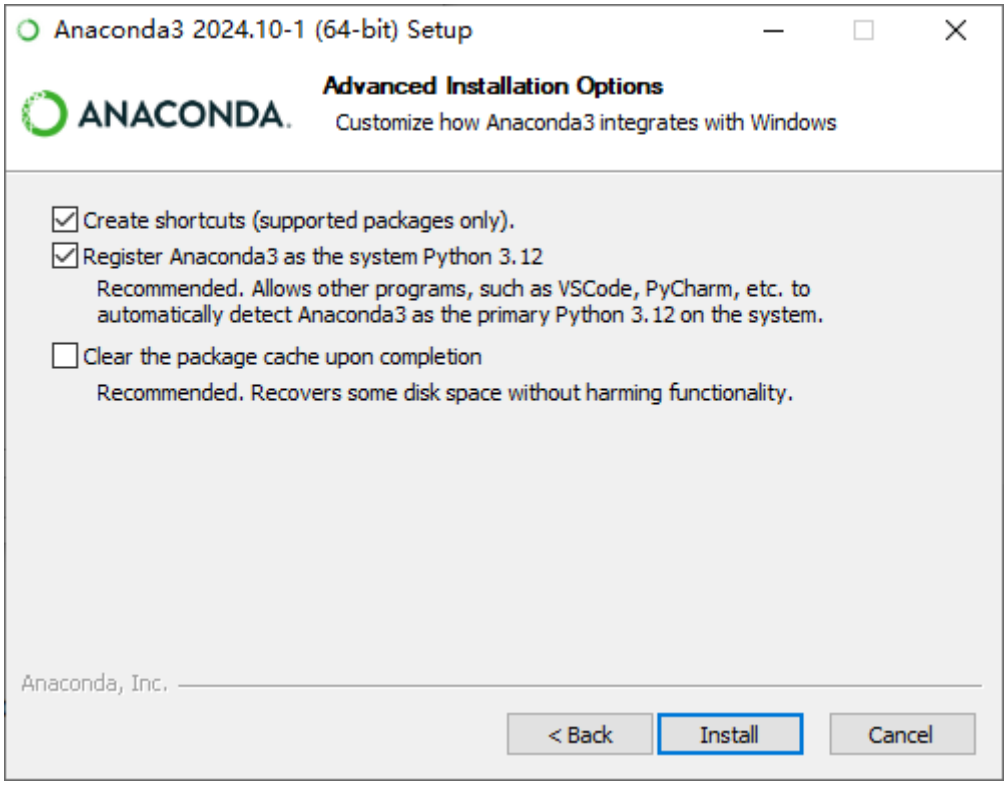
根据需求选择安装类型，通常建议选择“为所有用户安装”以实现系统级部署。



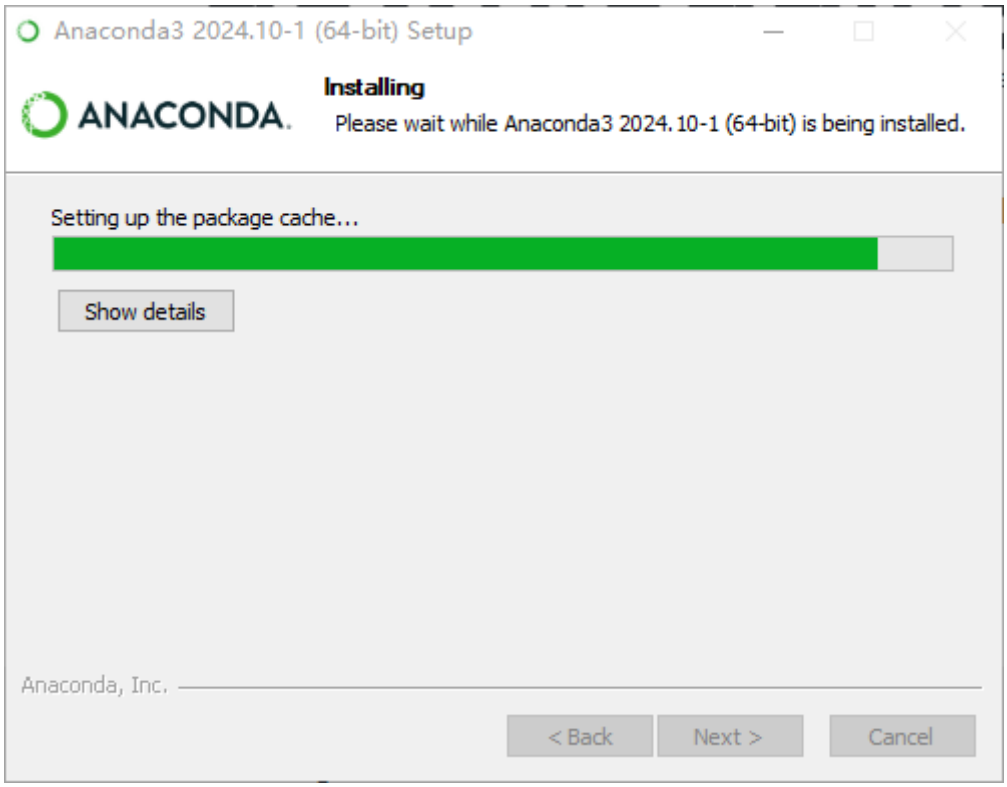
指定软件安装位置



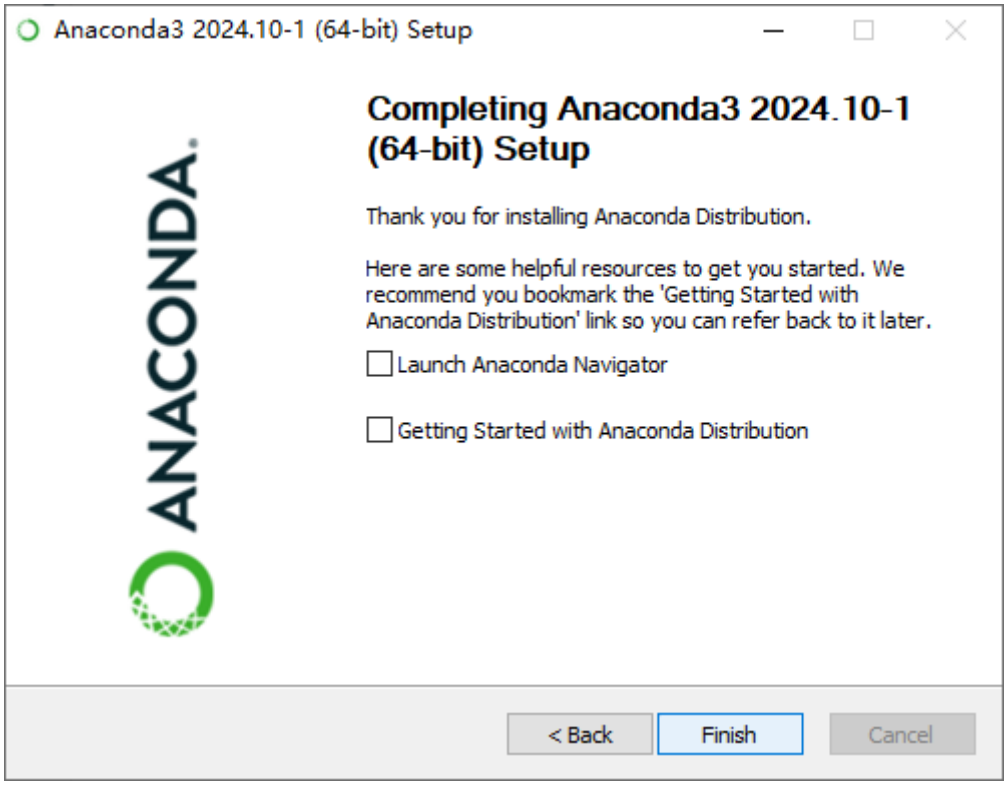
保持以下配置为默认值，点击 安装 继续。



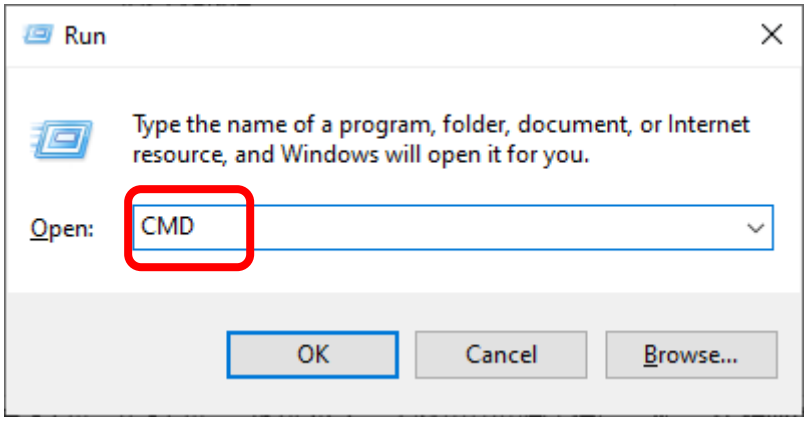
请等待安装完成，此过程可能需要一些时间。



至此，软件已安装完成。



使用快捷键 Win+R，在弹出的窗口中输入 cmd 即可打开命令提示符界面。



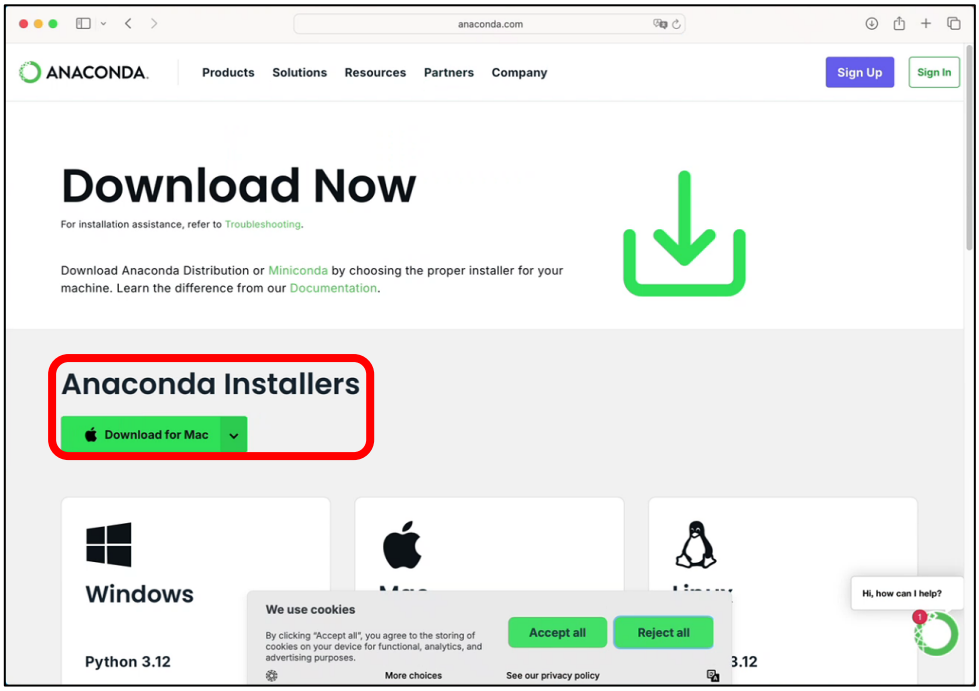
输入 **conda --version** 并按回车键。若 Anaconda3 安装正确，您将看到如下版本信息：

Mac OS

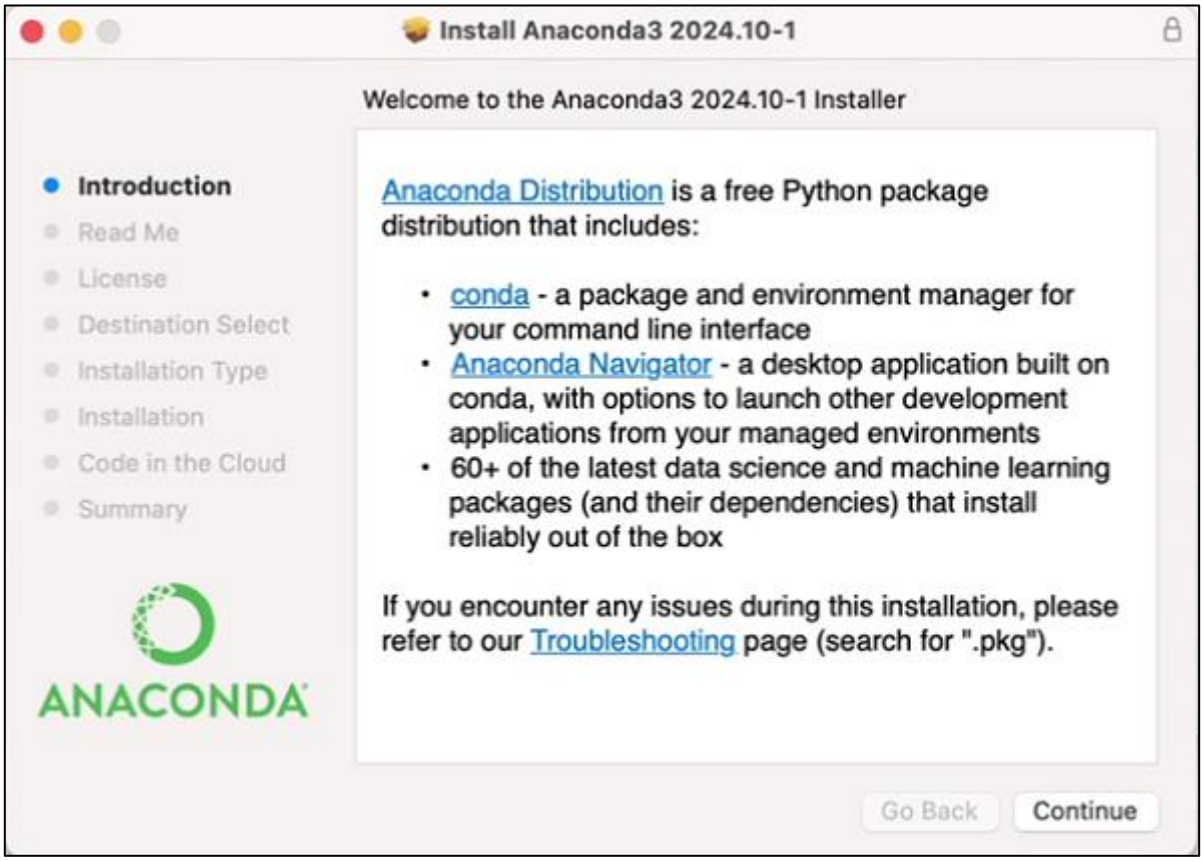
本示例使用 Conda 进行依赖管理，因此需提前在系统中安装 Conda。若尚未安装，请从以下地址下载安装： https://www.anaconda.com/download/success

请根据操作系统选择对应的安装程序。

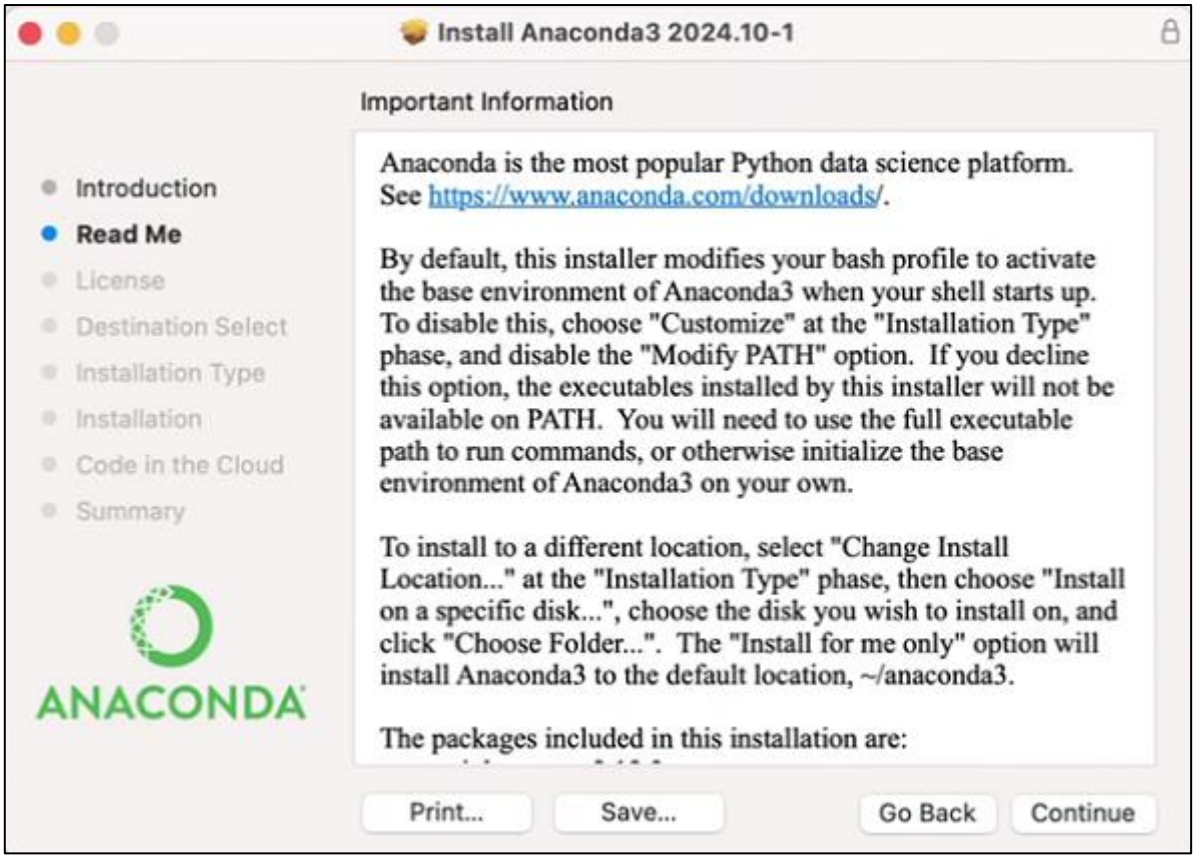
Miniconda 是 Anaconda 官方提供的轻量级安装工具，已预配置可无缝对接 Anaconda 资源库。



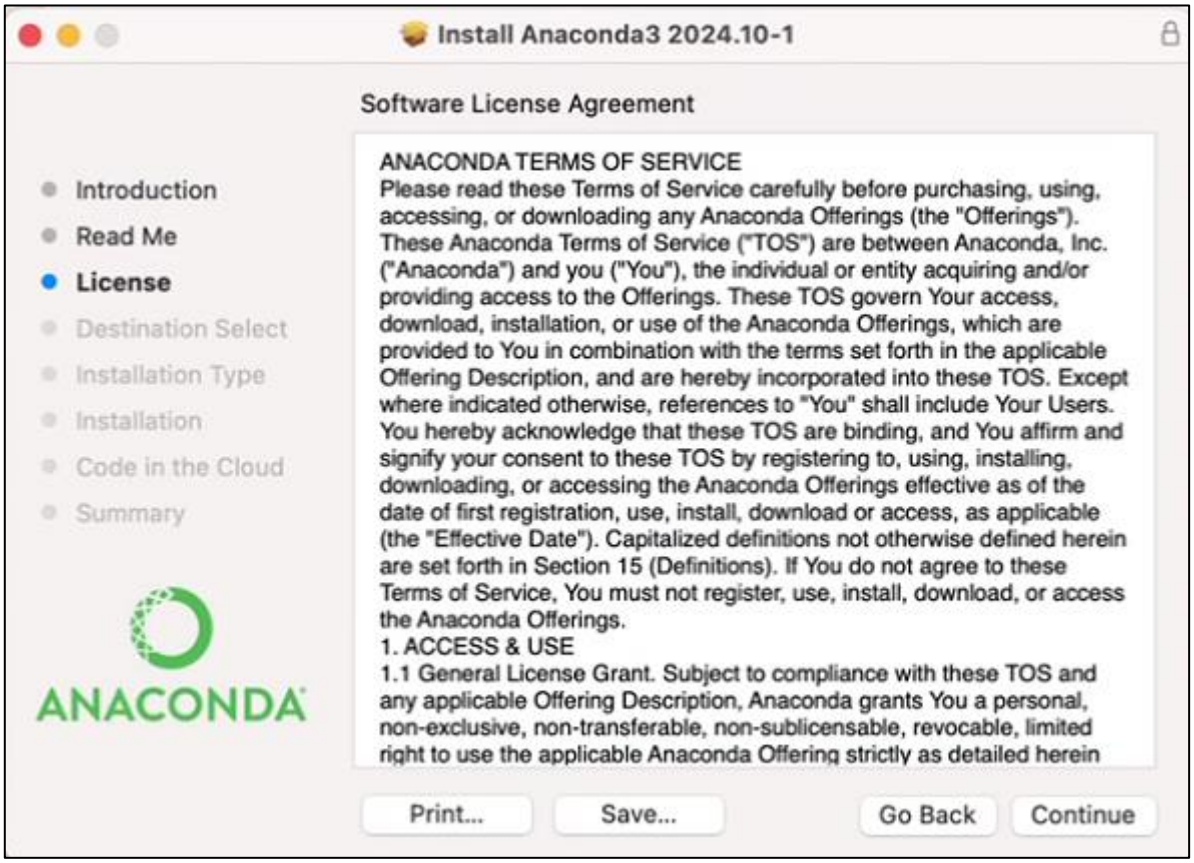
双击打开 Conda 安装程序，点击 继续 按钮。



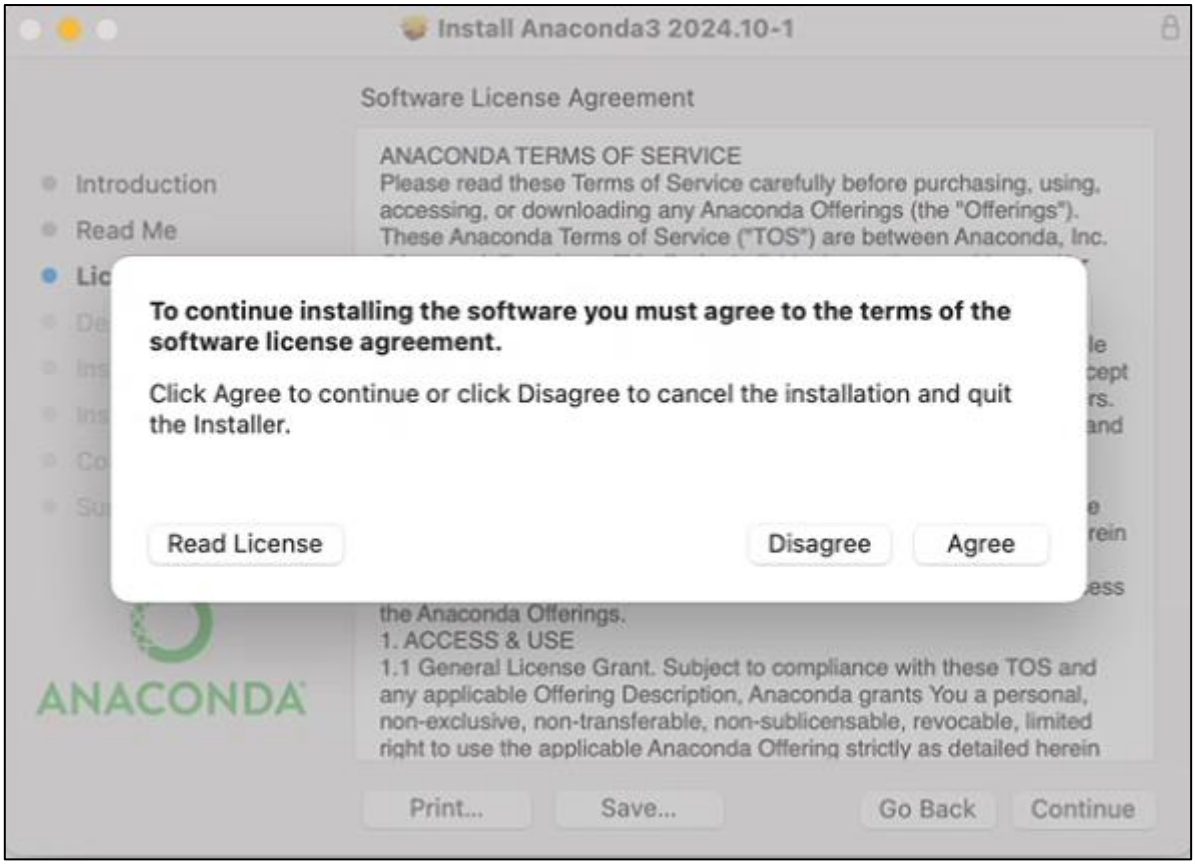
点击继续



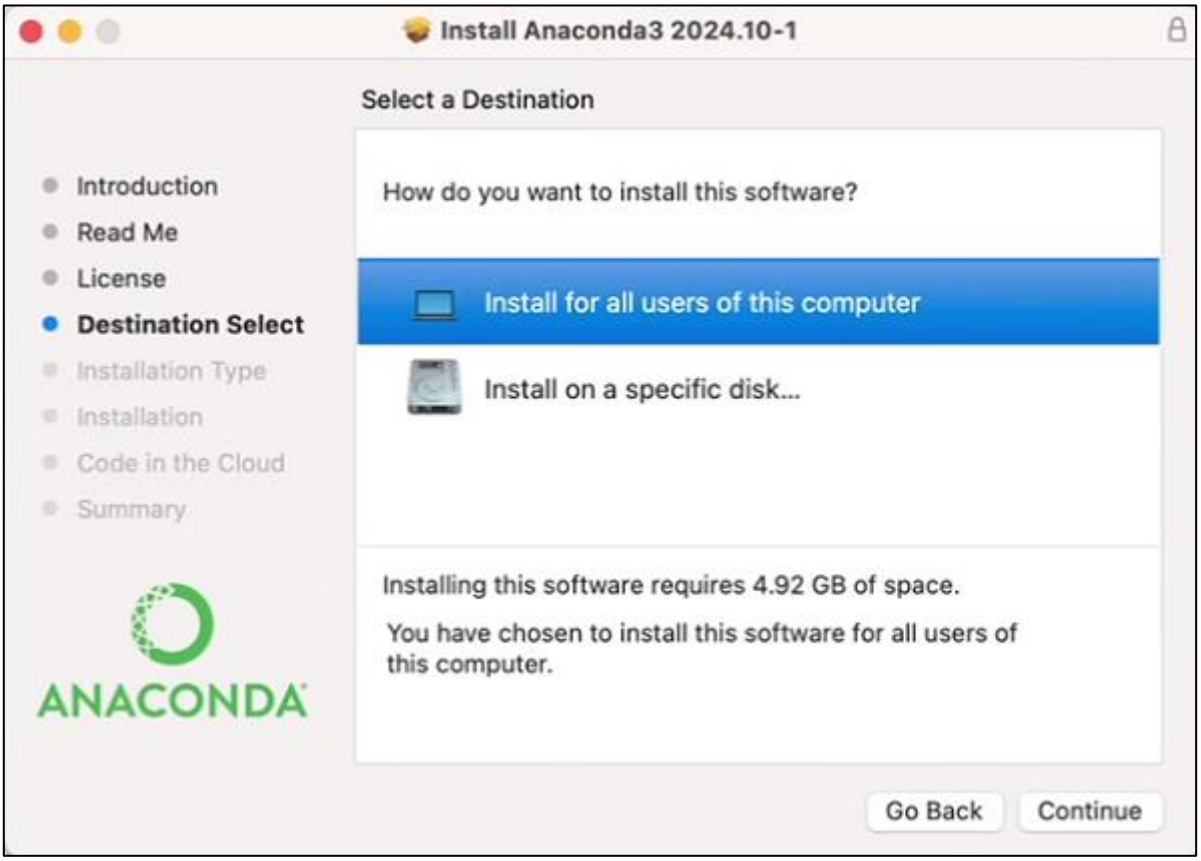
点击继续


Install Anaconda3 2024.10-1
Software License Agreement
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Code in the Cloud
Summary
ANACONDA
ANACONDA TERMS OF SERVICE
Please read these Terms of Service carefully before purchasing, using, accessing, or downloading any Anaconda Offerings (the "Offerings"). These Anaconda Terms of Service ("TOS") are between Anaconda, Inc. ("Anaconda") and you ("You"), the individual or entity acquiring and/or providing access to the Offerings. These TOS govern Your access, download, installation, or use of the Anaconda Offerings, which are provided to You in combination with the terms set forth in the applicable Offering Description, and are hereby incorporated into these TOS. Except where indicated otherwise, references to "You" shall include Your Users. You hereby acknowledge that these TOS are binding, and You affirm and signify your consent to these TOS by registering to, using, installing, downloading, or accessing the Anaconda Offerings effective as of the date of first registration, use, install, download or access, as applicable (the "Effective Date"). Capitalized definitions not otherwise defined herein are set forth in Section 15 (Definitions). If You do not agree to these Terms of Service, You must not register, use, install, download, or access the Anaconda Offerings.
1. ACCESS & USE
1.1 General License Grant. Subject to compliance with these TOS and any applicable Offering Description, Anaconda grants You a personal, non-exclusive, non-transferable, non-sublicensable, revocable, limited right to use the applicable Anaconda Offering strictly as detailed herein
Print...
Save...
Go Back
Continue

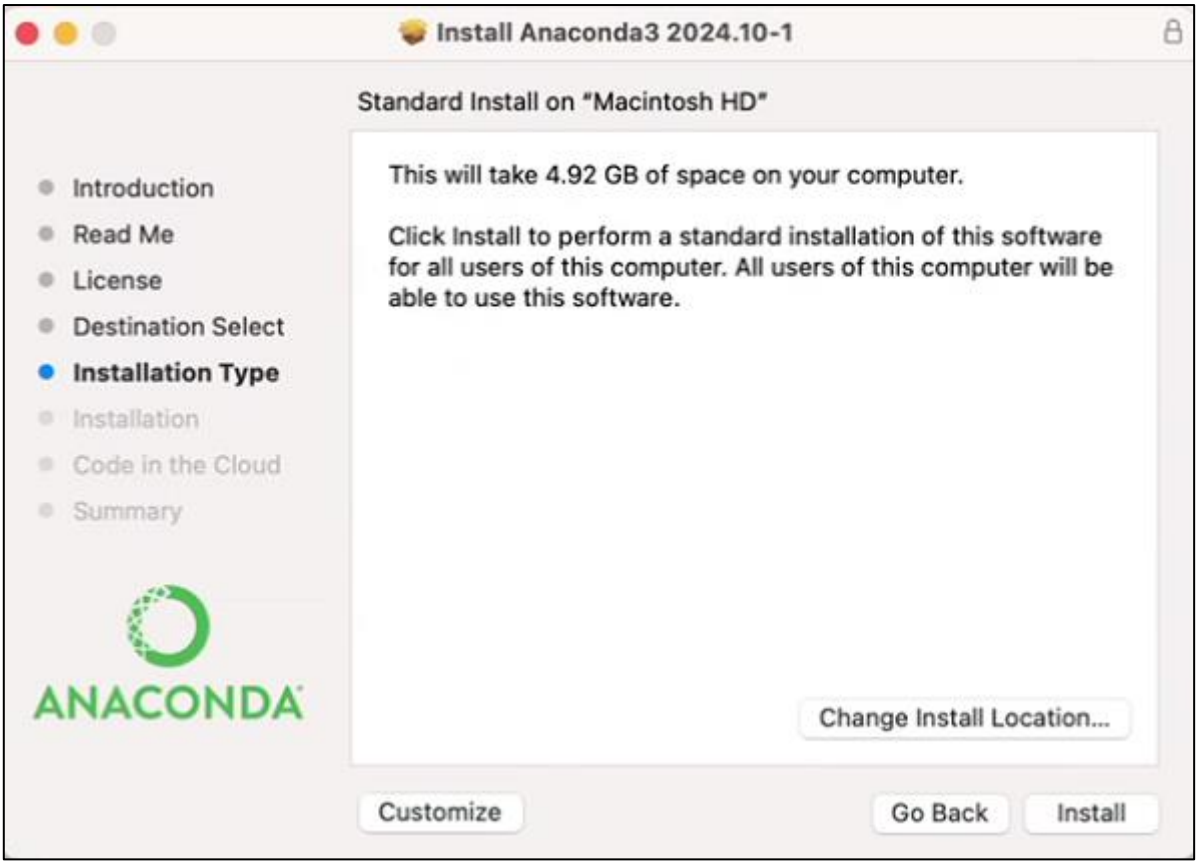
点击同意


Install Anaconda3 2024.10-1
Software License Agreement
Introduction
Read Me
To continue installing the software you must agree to the terms of the software license agreement.
Click Agree to continue or click Disagree to cancel the installation and quit the Installer.
Read License
Disagree
Agree
ANACONDA
the Anaconda Offerings.
1. ACCESS & USE
1.1 General License Grant. Subject to compliance with these TOS and any applicable Offering Description, Anaconda grants You a personal, non-exclusive, non-transferable, non-sublicensable, revocable, limited right to use the applicable Anaconda Offering strictly as detailed herein
Print...
Save...
Go Back
Continue

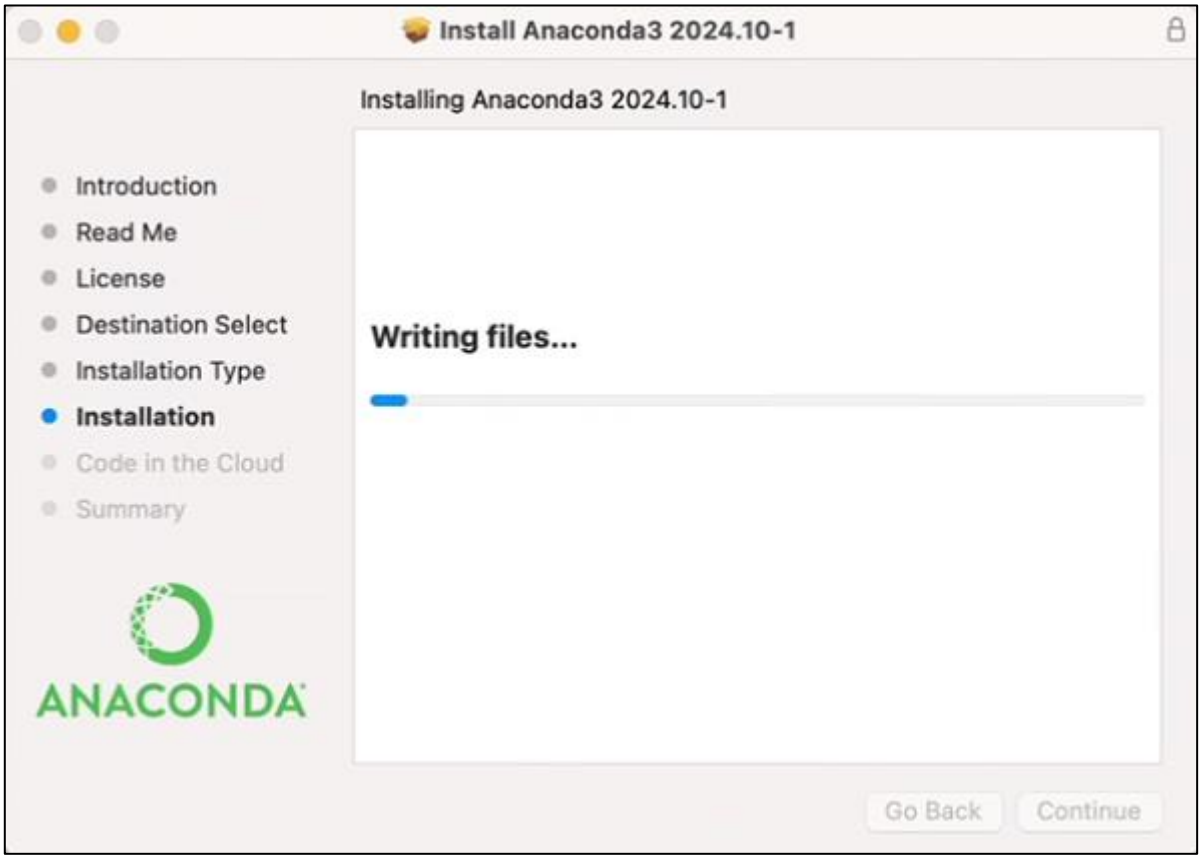
点击“继续”使用默认设置进行安装。



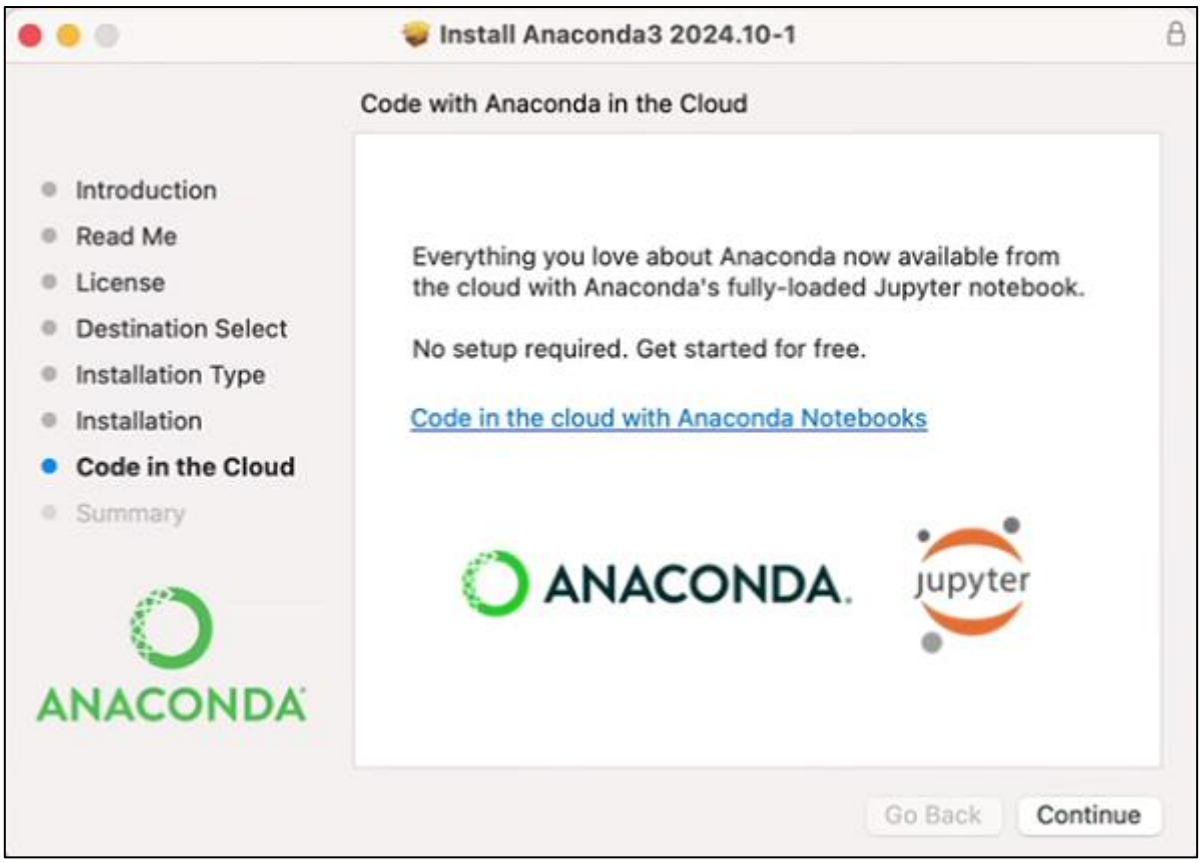
点击 安装。



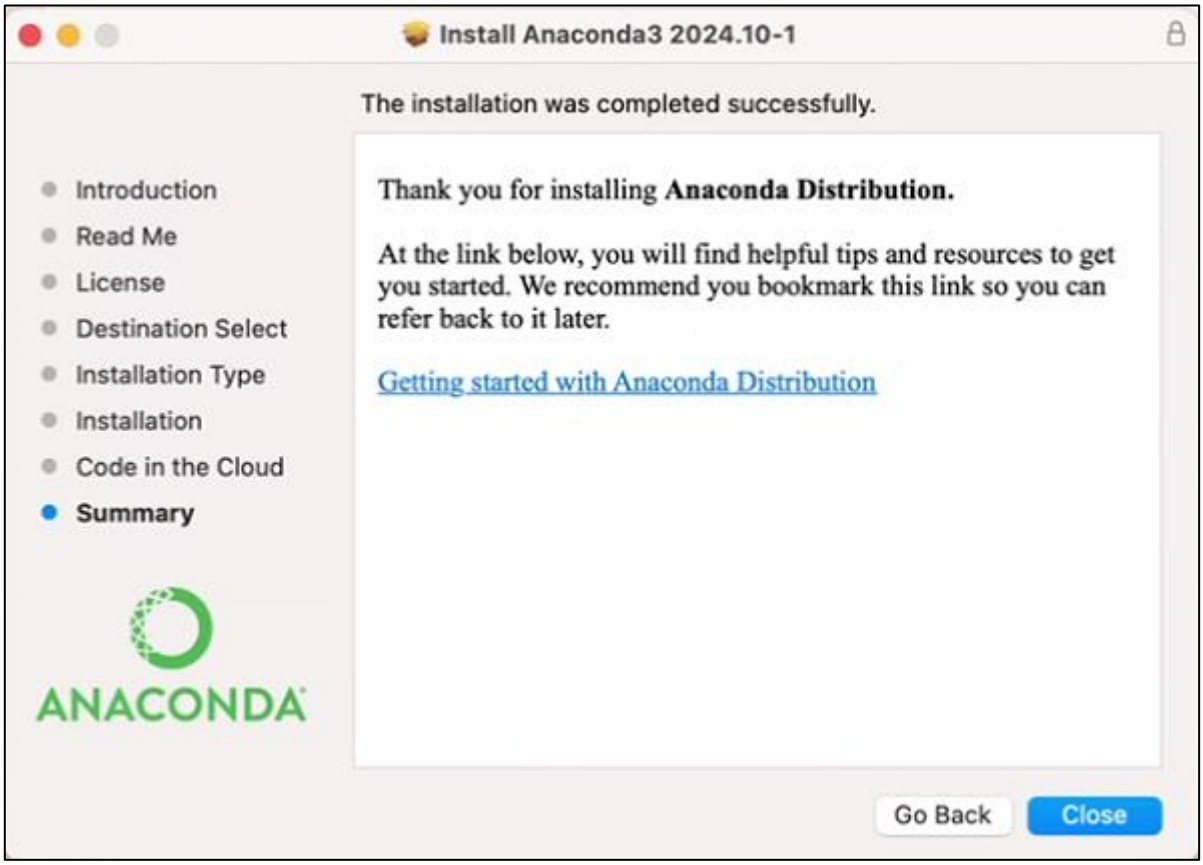
请等待数分钟以完成安装。



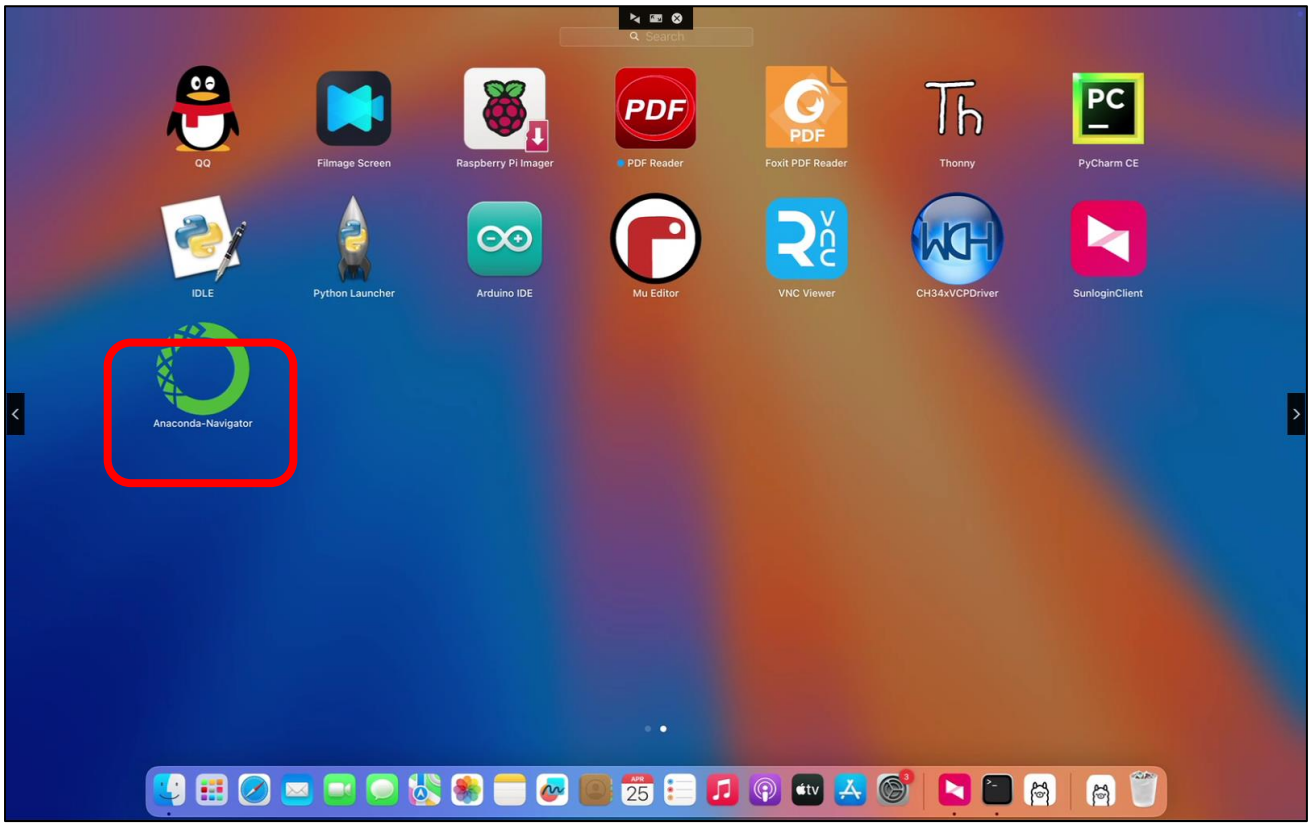
点击继续



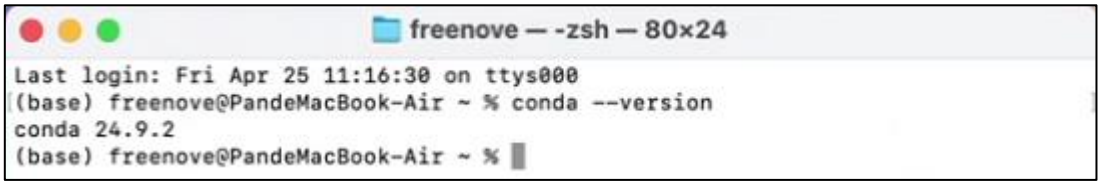
点击关闭



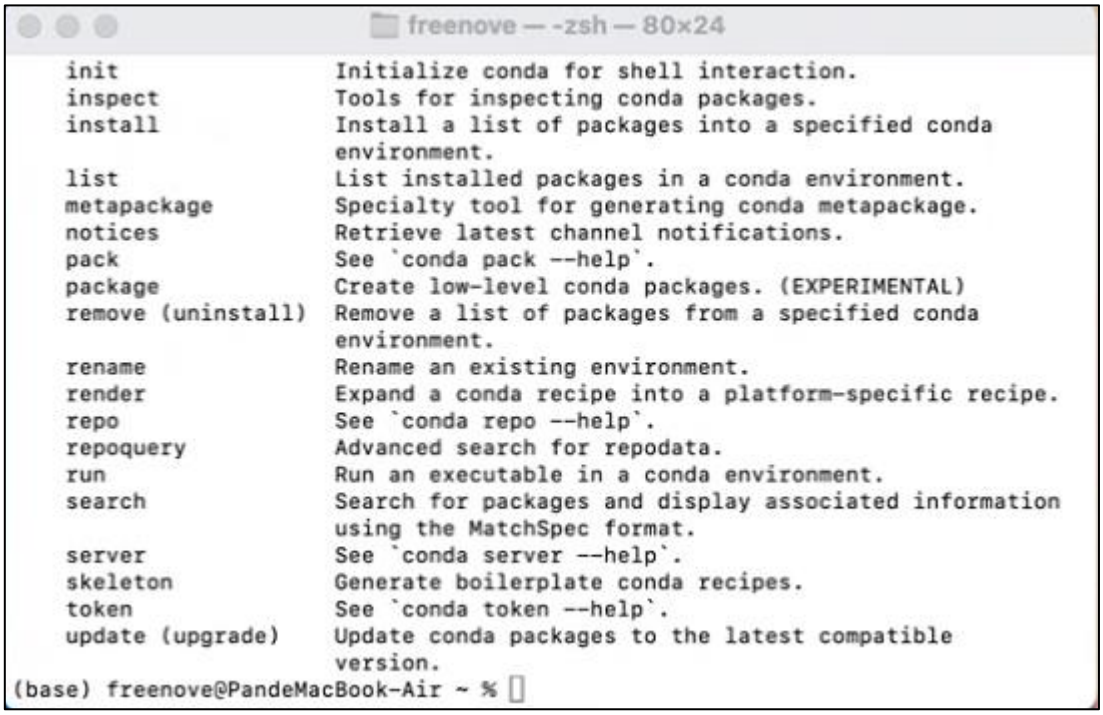
Conda 已成功安装，该应用程序将出现在您的程序/应用列表中。



双击运行（此步骤不会显示任何可见响应），然后重新打开终端，此时应出现"(base)"前缀提示符。您也可通过运行命令 conda --version 验证版本信息。



您可通过运行 conda -h 查看更多使用说明。

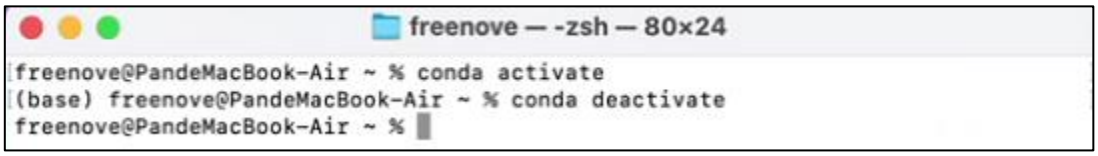


首次使用 conda 时，需运行 conda init 命令初始化并激活已安装的 conda 环境。

```
conda init
```

您可通过 conda activate 启用虚拟环境，或使用 conda deactivate 退出当前环境。

```
conda activate
conda deactivate
```



要实现终端启动时自动激活 conda 环境，请执行： conda config --set auto_activate_base true
若要禁用此自动激活功能，请运行： conda config --set auto_activate_base false

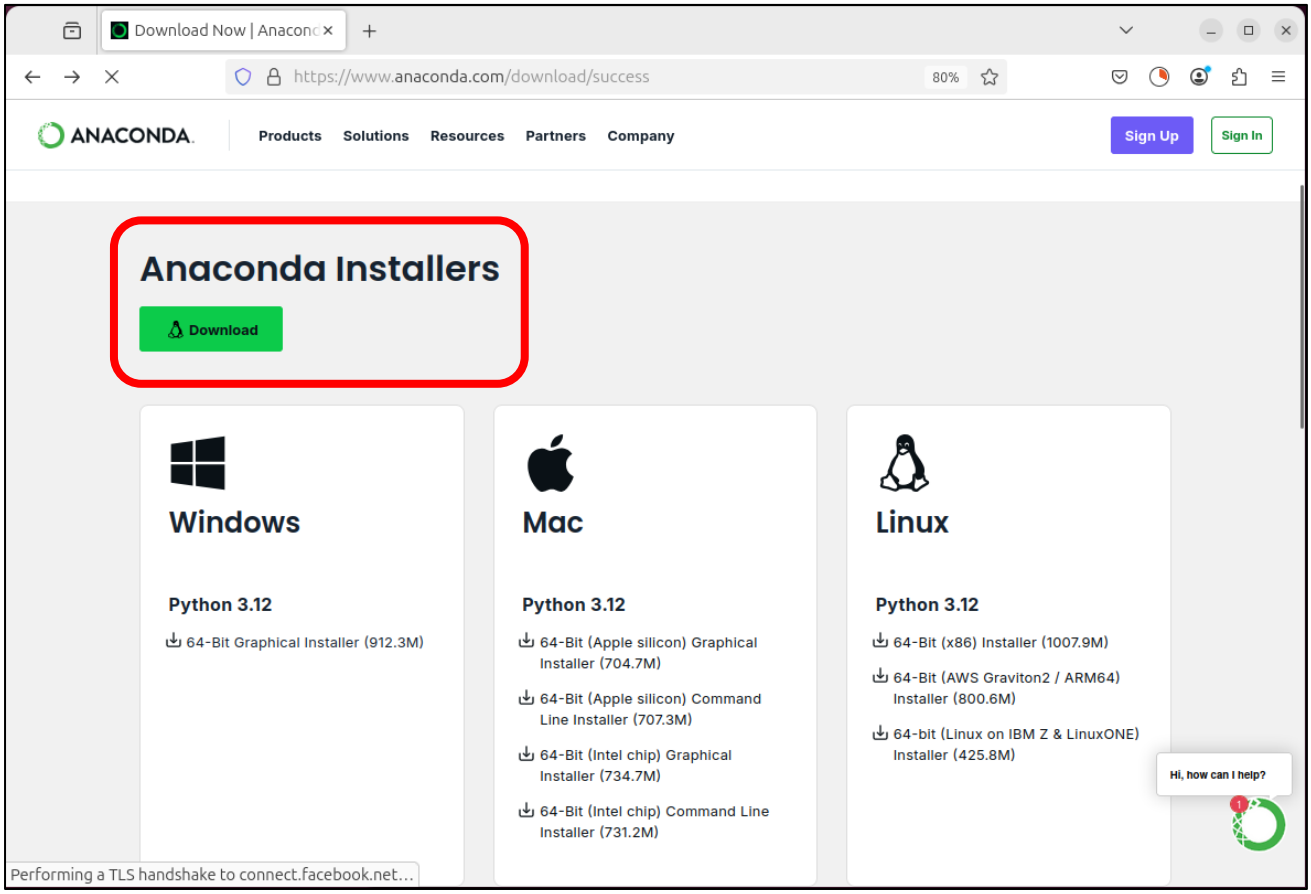
```
conda config --set auto_activate_base false
conda config --set auto_activate_base true
```

Linux

本示例使用 Conda 进行依赖管理，因此需提前在系统中安装 Conda。若尚未安装，请从以下地址下载：
https://www.anaconda.com/download/success

根据操作系统选择对应的安装程序。

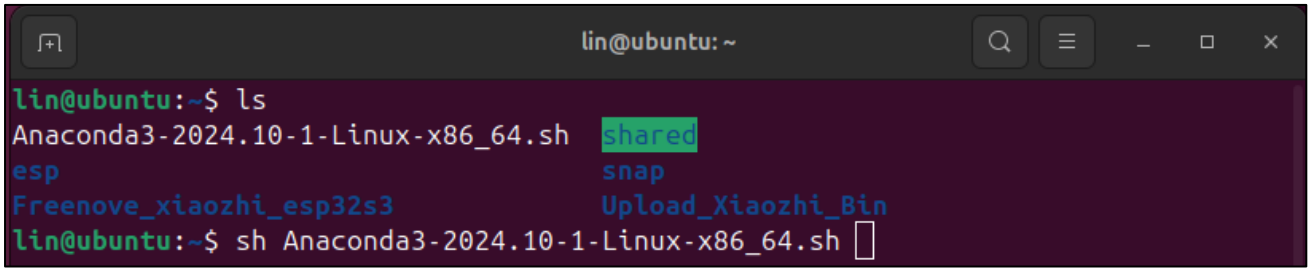
Miniconda 是 Anaconda 官方提供的轻量级安装工具，已预配置可无缝对接 Anaconda 资源库。



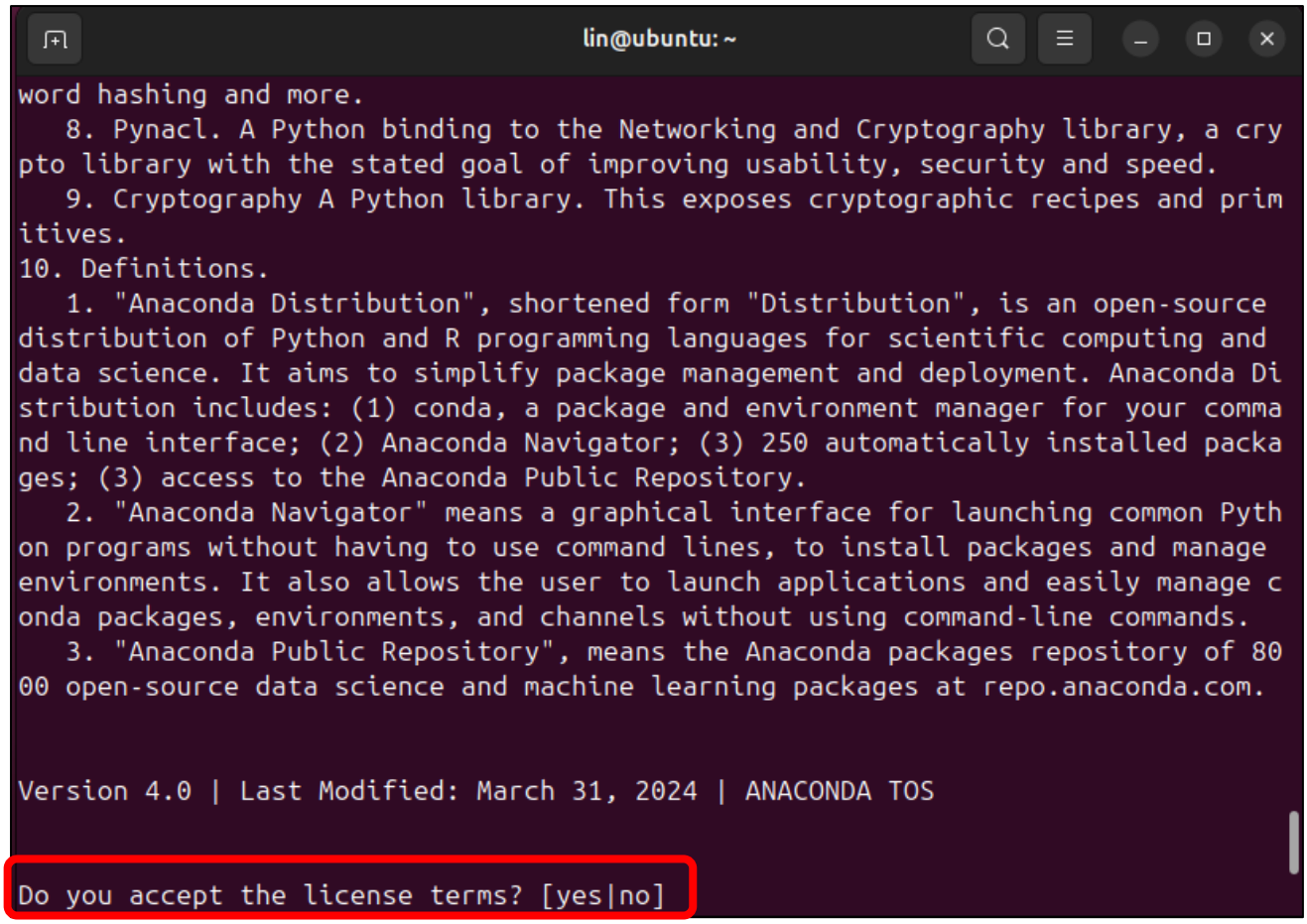
此处下载的文件名为 "Anaconda3-2024.10-1-Linux-x86_64.sh"（注意：不同计算机上的文件名可能有所差异）。

要安装 Anaconda，请打开终端并执行以下命令：

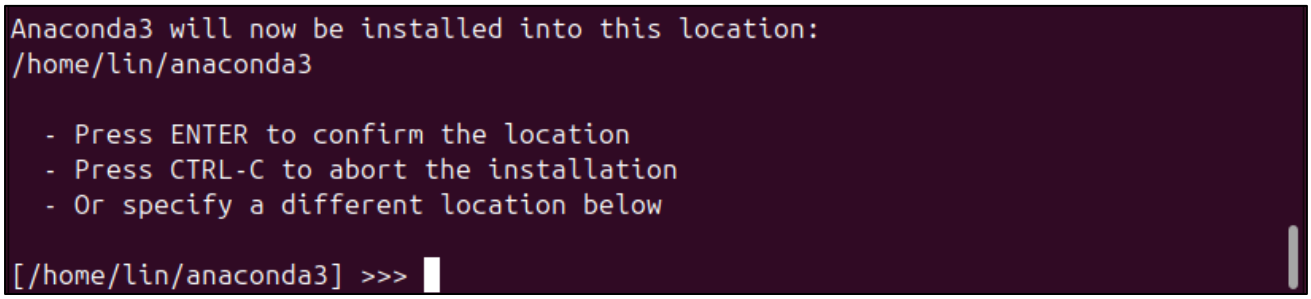
```
sh Anaconda3-2024.10-1-Linux-x86_64.sh
```



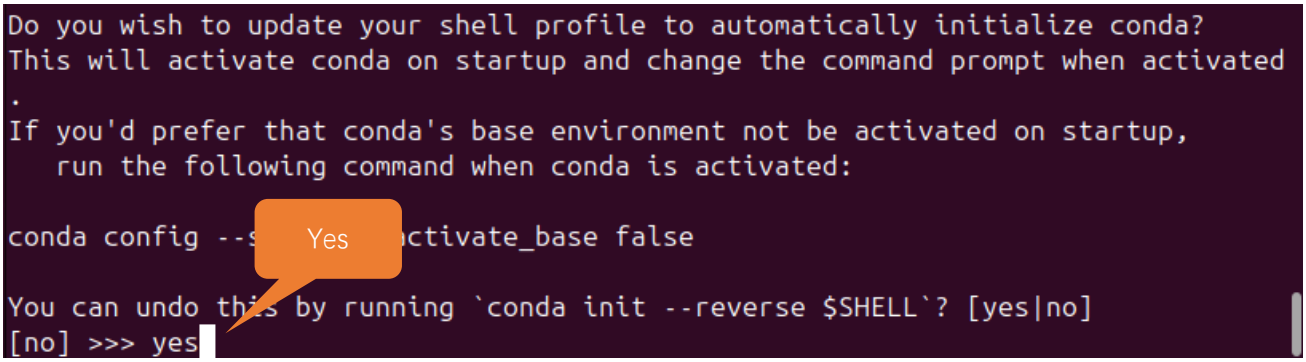
持续按住回车键，直到出现如下提示后输入“yes”。



选择安装路径后按回车键，可直接使用默认位置。



安装过程需要联网，请确保网络连接稳定并耐心等待数分钟，直至出现以下提示信息。
需输入 Yes 确认继续。



出现以下提示即表示 conda 已成功安装。

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

要在终端启动时自动激活 conda 环境，请执行： **conda config --set auto_activate_base true**
若要禁用此自动激活功能，请运行： **conda config --set auto_activate_base false**

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

我们建议禁用自动激活功能，请执行： conda config --set auto_activate_base false

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

重新打开终端后，执行 conda --version 命令即可验证 conda 版本。

```
conda --version
```

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

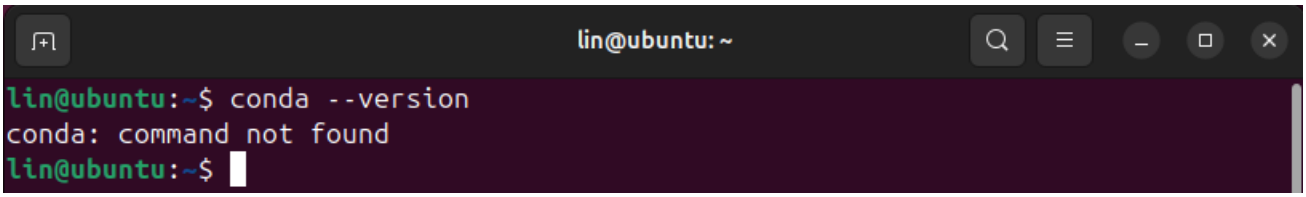
以下两条命令可用于激活或退出 conda 虚拟环境。

```
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:~$
```

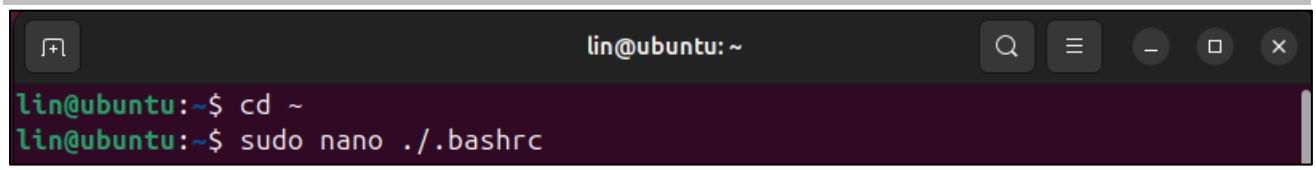
如果在检查 conda 版本时出现以下错误，

```
conda –version
```



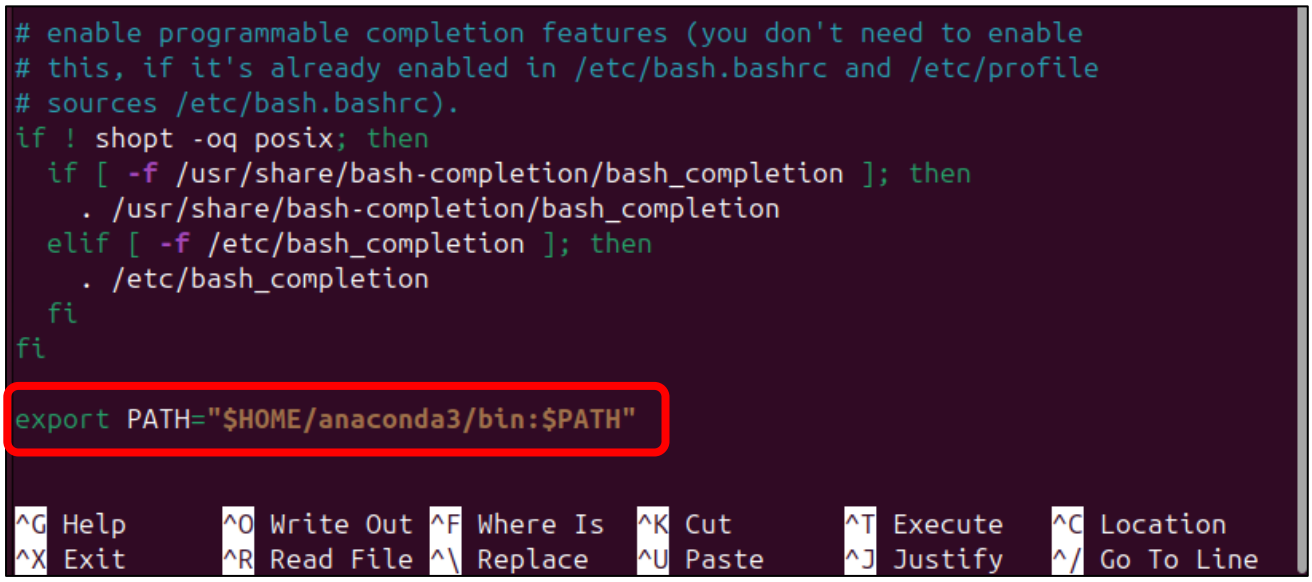
这表明虽然已安装 Conda，但尚未添加到您的 PATH 环境变量中。
请按以下步骤将 Conda 添加到 PATH：
使用 nano 编辑 ".bashrc" 文件：

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



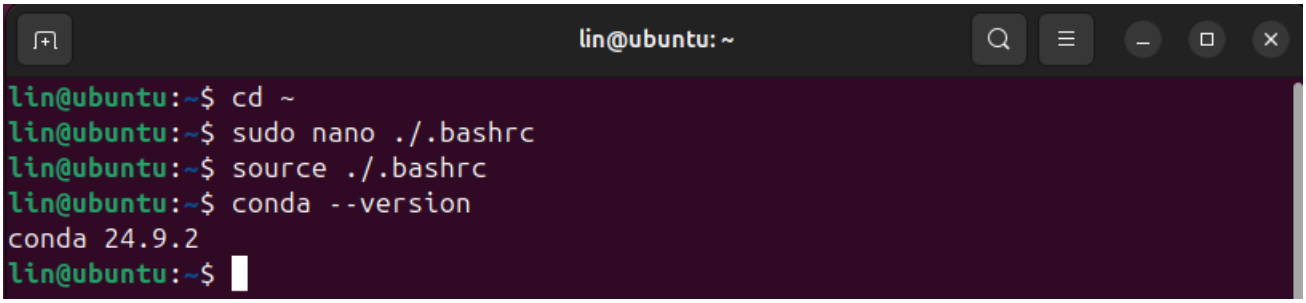
请将以下内容添加到文件末尾：

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



按 Ctrl+O 保存文件，按 Ctrl+X 退出编辑。
运行以下命令使配置生效，然后重新检查 conda 版本：

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

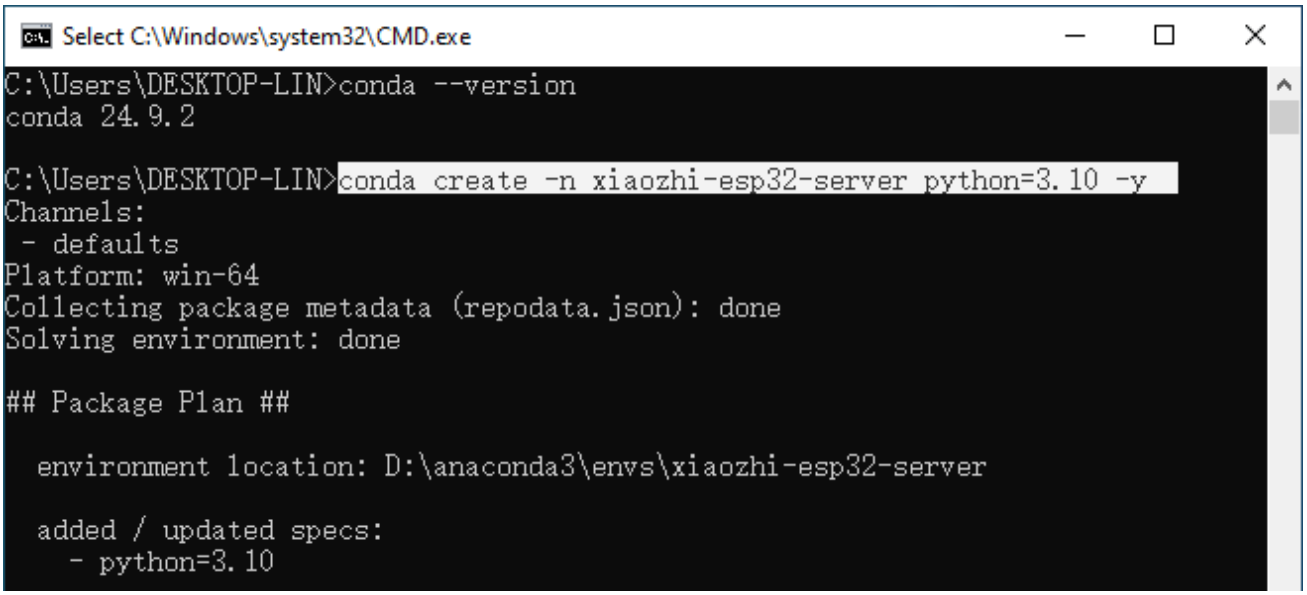


部署虚拟环境

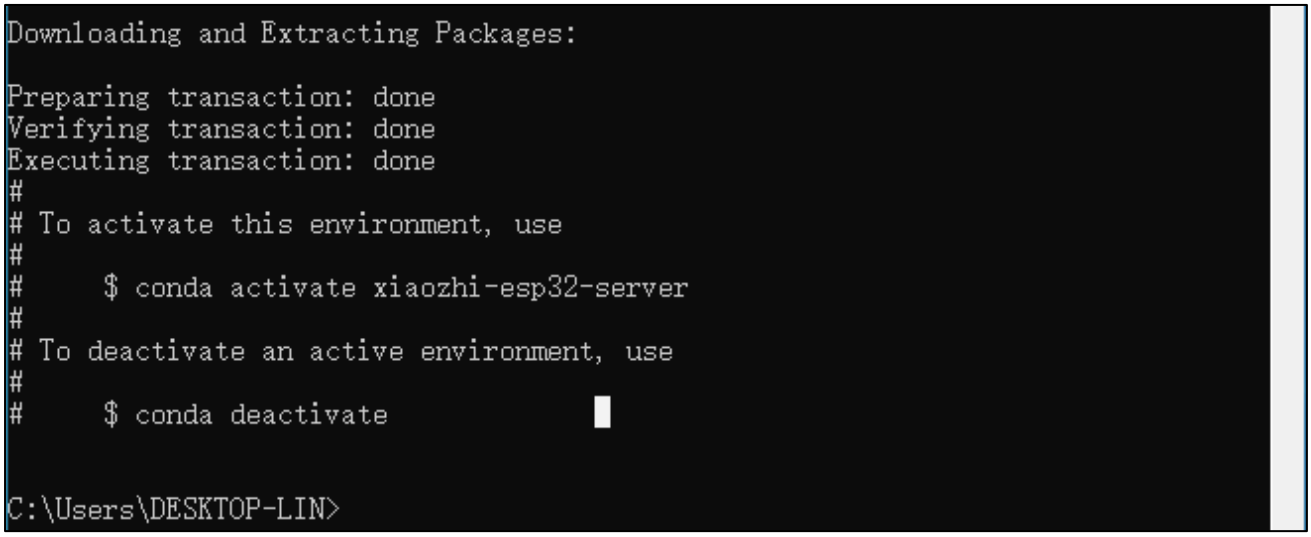
请注意，部署虚拟环境的命令在 Windows、Mac 和 Ubuntu 系统中是通用的。本文示例使用 Windows 系统，但相同操作适用于其他平台。

打开 CMD/Terminal 终端界面，运行以下命令创建一个名为"xiaozhi-esp32-server"、预装 Python 3.10 的虚拟环境。

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

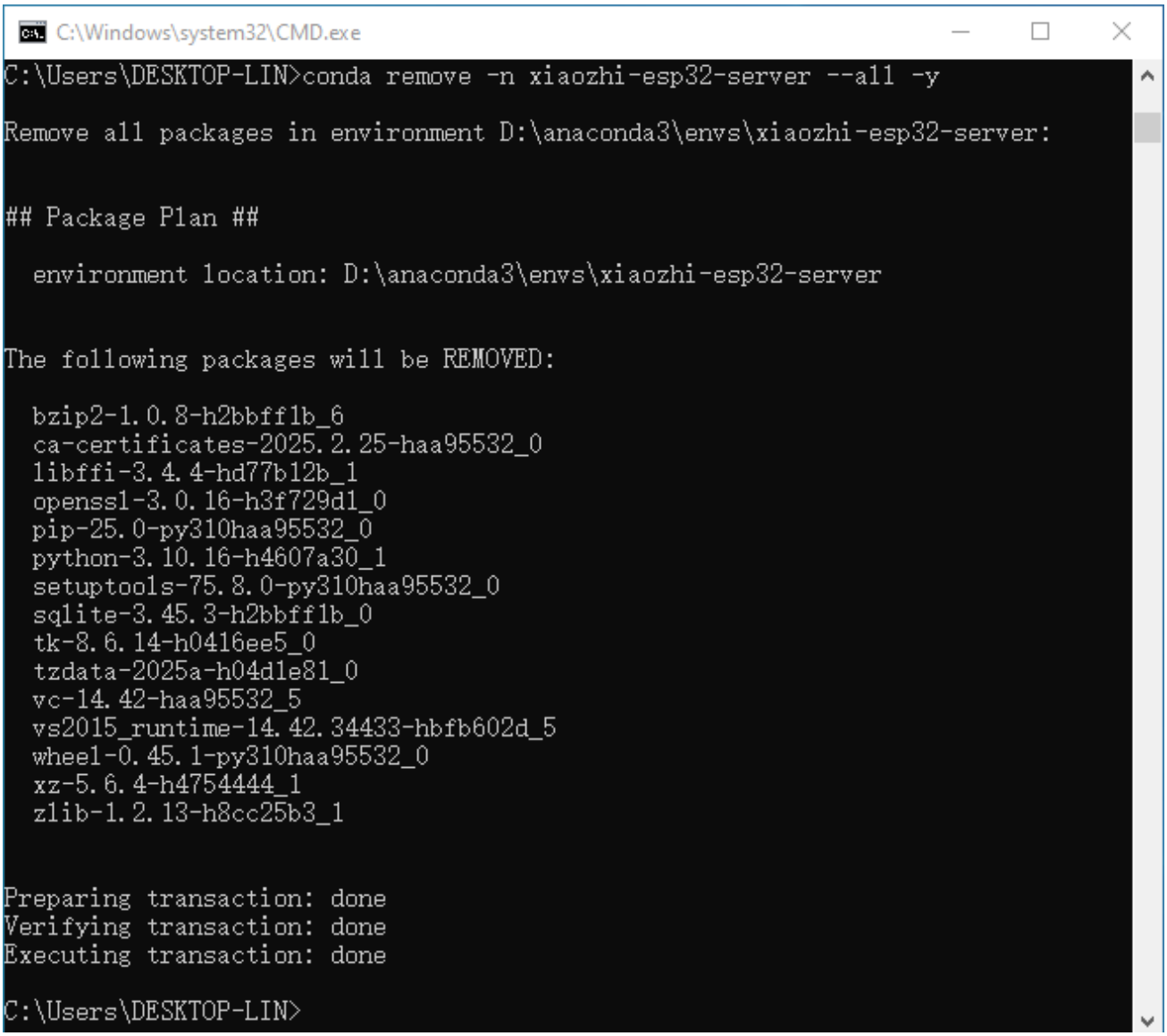


当出现以下提示信息时，表示虚拟环境已创建成功。



要删除虚拟环境，请运行以下命令：

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



您也可以使用以下两条命令来激活或退出虚拟环境。

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

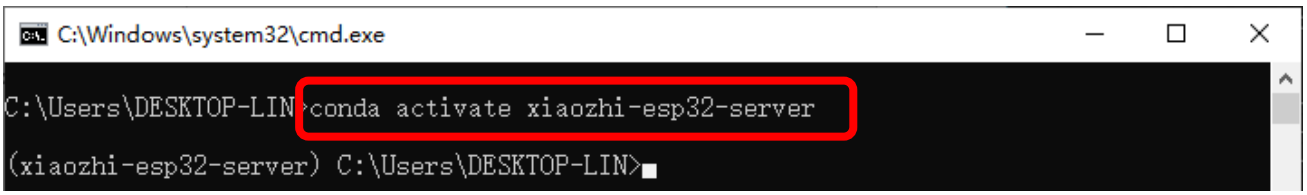
重要提示:

如果您在激活环境时收到提示建议运行 conda init，请执行“conda init”命令并重新启动终端以使更改生效。

部署 xiaozhi-esp32-server 环境

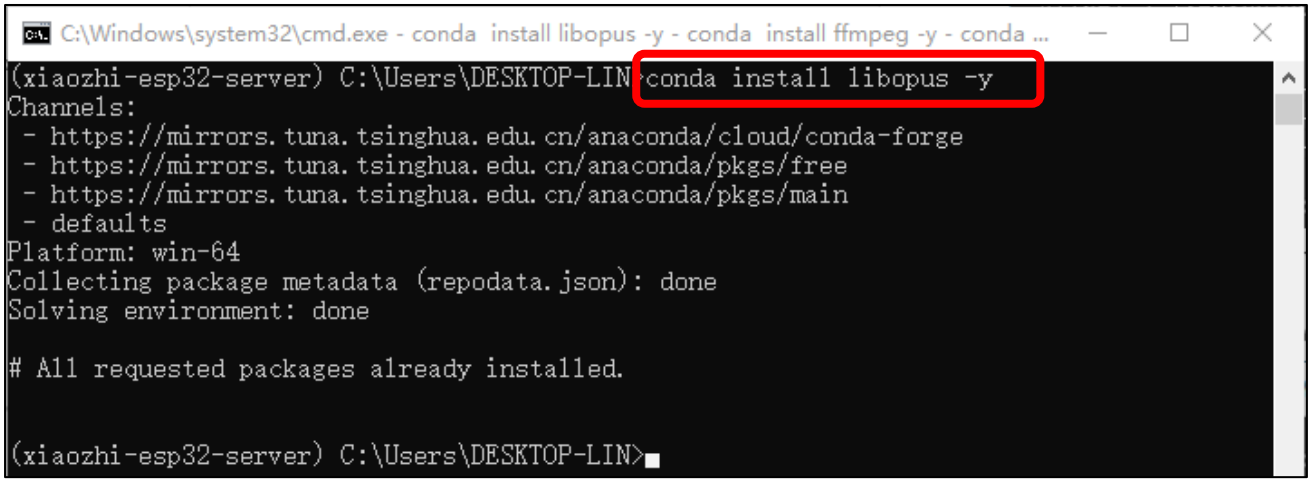
Windows 用户请打开命令提示符（CMD） macOS 或 Ubuntu 用户请使用终端
本教程主要以 Windows 截图进行演示。若存在操作差异，我们将提供其他系统的对应示例。
激活虚拟环境

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



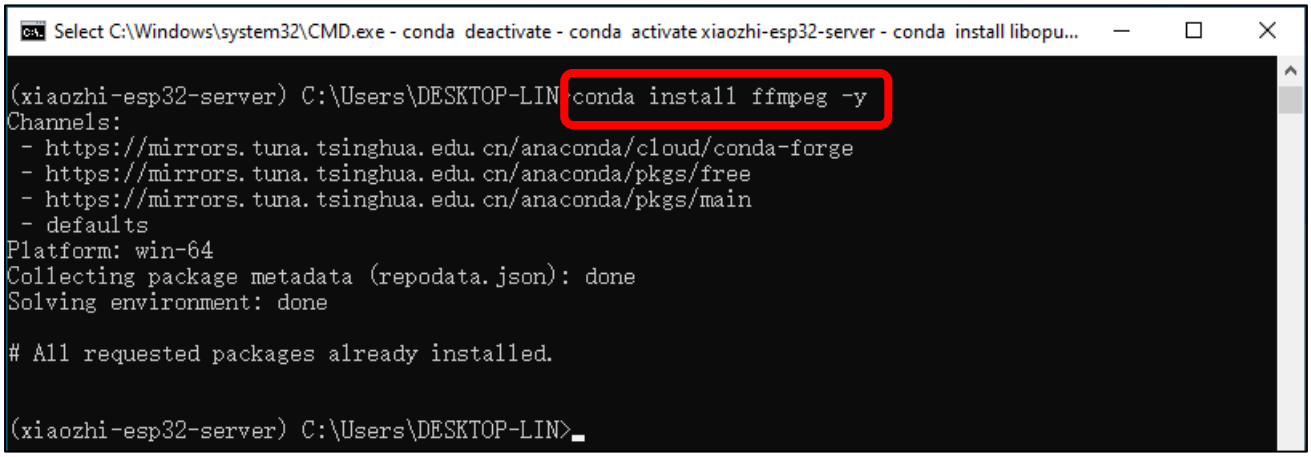
在虚拟环境中安装 libopus

```
conda install libopus -y
```



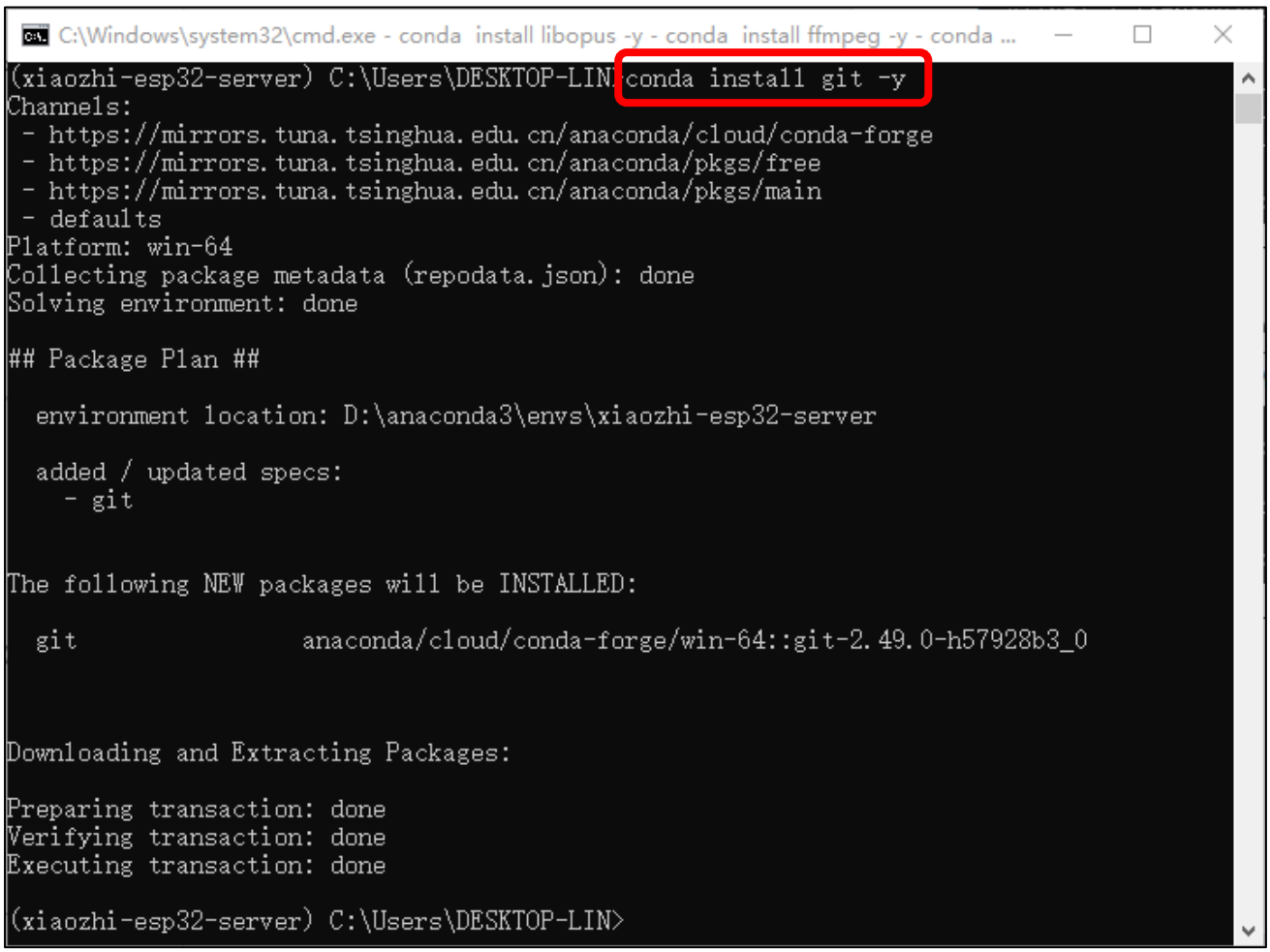
在虚拟环境中安装 FFmpeg

```
conda install ffmpeg -y
```



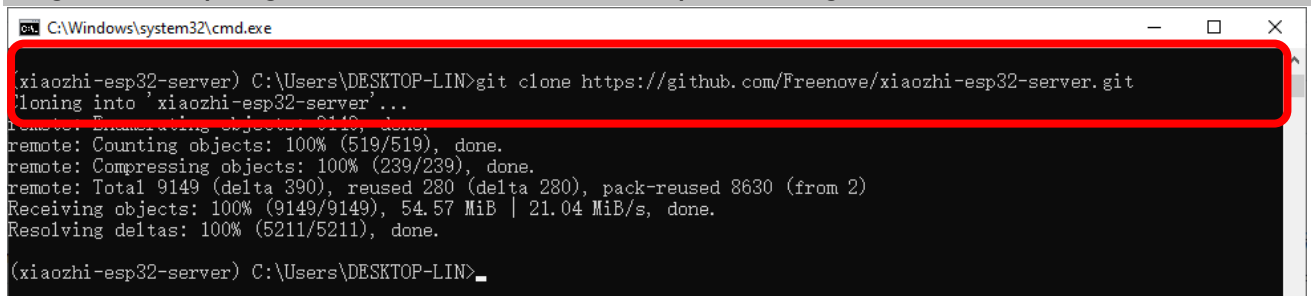
在虚拟环境中安装 Git

```
conda install git -y
```



使用 git clone 命令下载服务器源代码

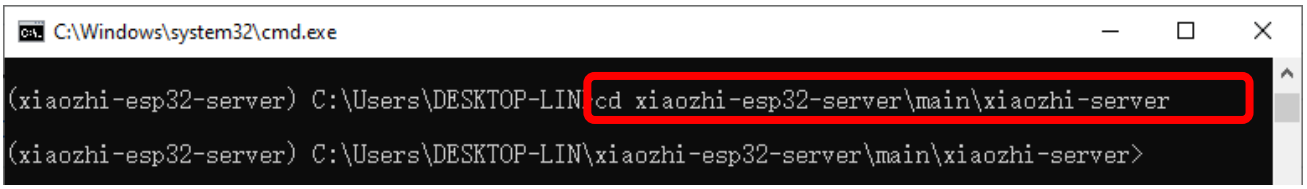
```
git clone https://github.com/Freenove/xiaozhi-esp32-server.git
```



进入服务器源代码目录

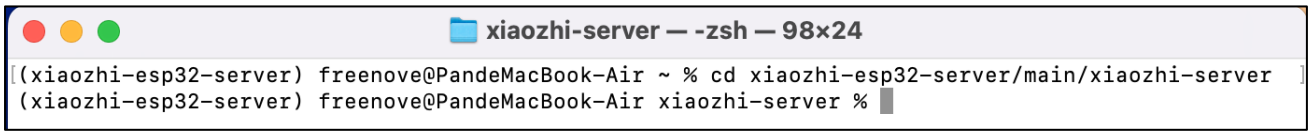
Windows 用户请注意：路径中使用反斜杠(\\)

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



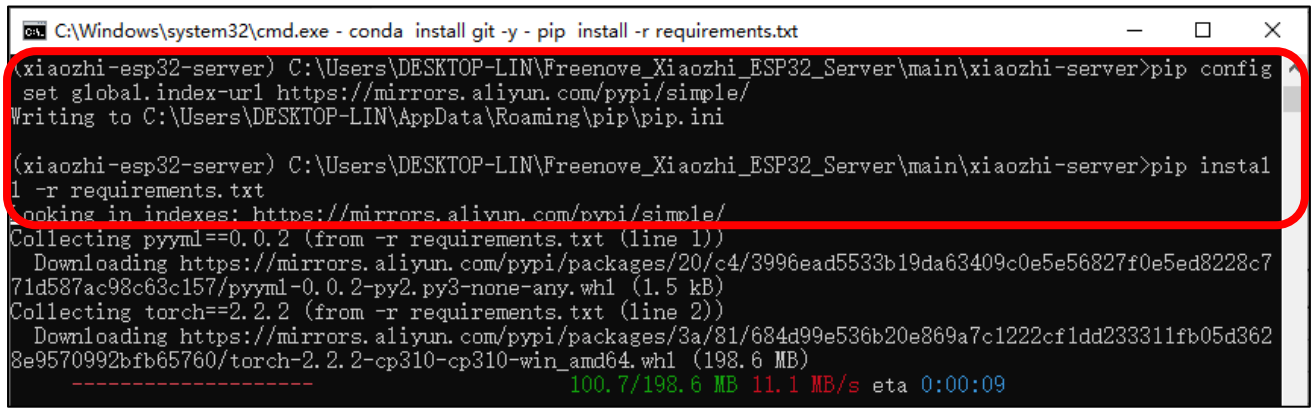
Mac/Linux 用户请注意：路径中使用正斜杠(/)

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



安装服务器源代码所需的库文件。 此过程可能需要一些时间，请确保网络连接稳定，并勿中断安装过程。

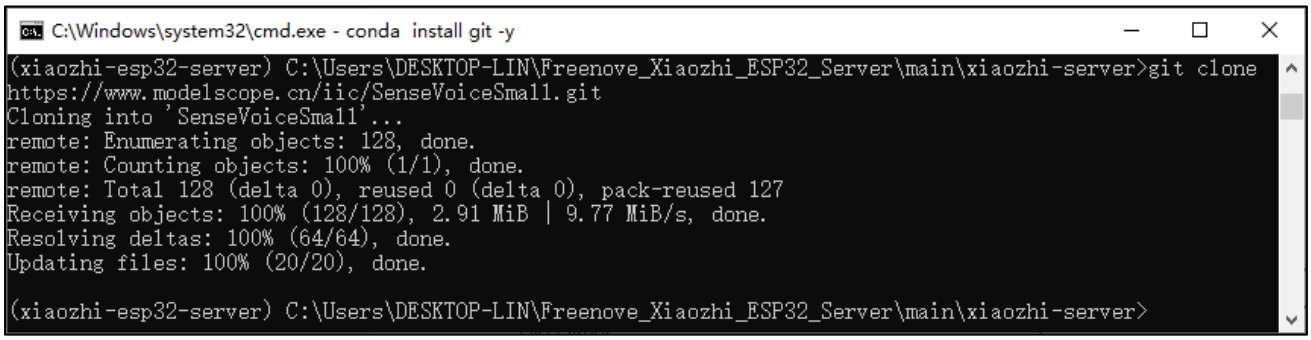
```
pip config set global.index-url https://mirrors.aliyun.com/pypi/simple/
pip install -r requirements.txt
```



当输出内容与以下截图匹配时，表示安装已完成。

安装语音模型

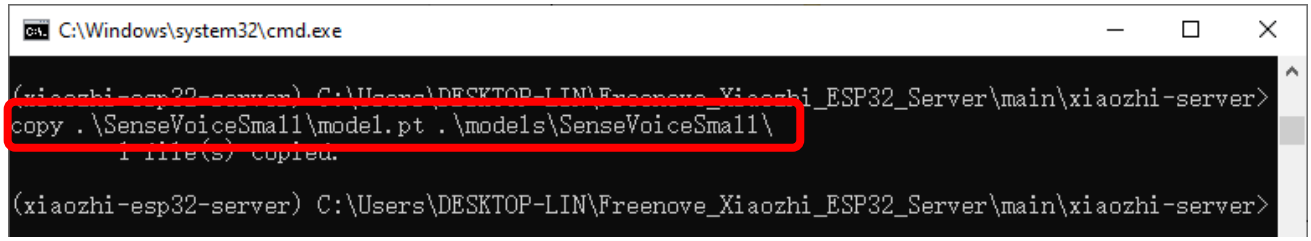
```
git clone https://www.modelscope.cn/iic/SenseVoiceSmall.git
```



使用 copy 命令将 model.pt 文件从 SenseVoiceSmall 复制到 models/SenseVoiceSmall 文件夹

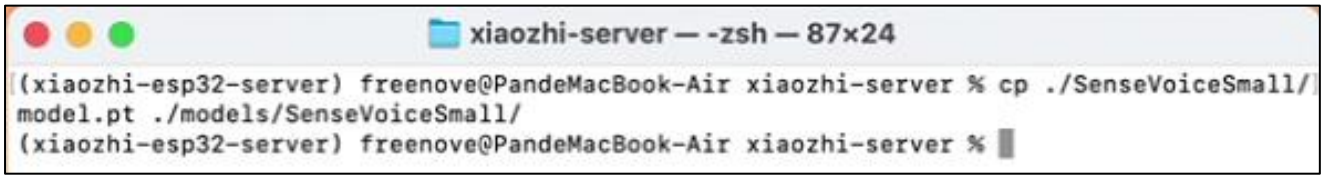
Windows 用户请使用 copy 命令

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



macOS/Linux 用户请使用 cp 命令

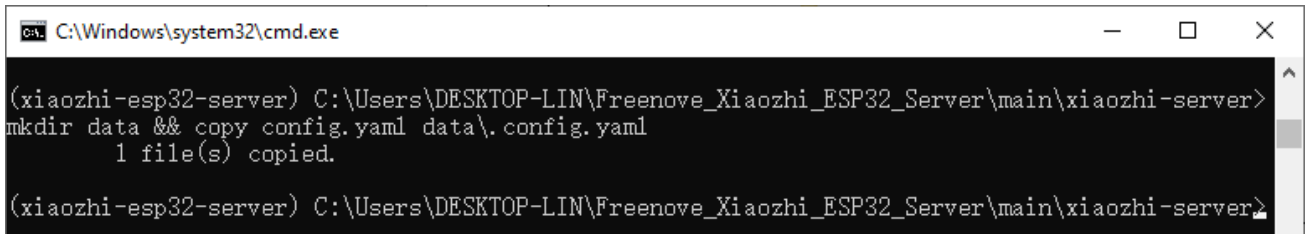
```
cp ./SenseVoiceSmall/model.pt ./models/SenseVoiceSmall/
```



在 CMD 界面中输入命令“mkdir data && copy config.yaml data.config.yaml”，它将在 xiaozhi-server 中创建一个名为“data”的文件夹，并将当前目录下的“config.yaml”文件复制到“data”文件夹中，重命名为“.config.yaml”。

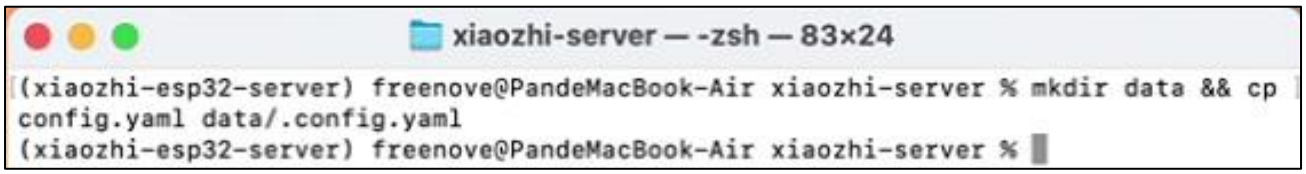
如果您是 Windows 用户，请执行：

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



如果您是 Mac/Linux 用户，请运行以下命令：

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



打开并修改 config.yaml 文件。

在 Windows 上运行:

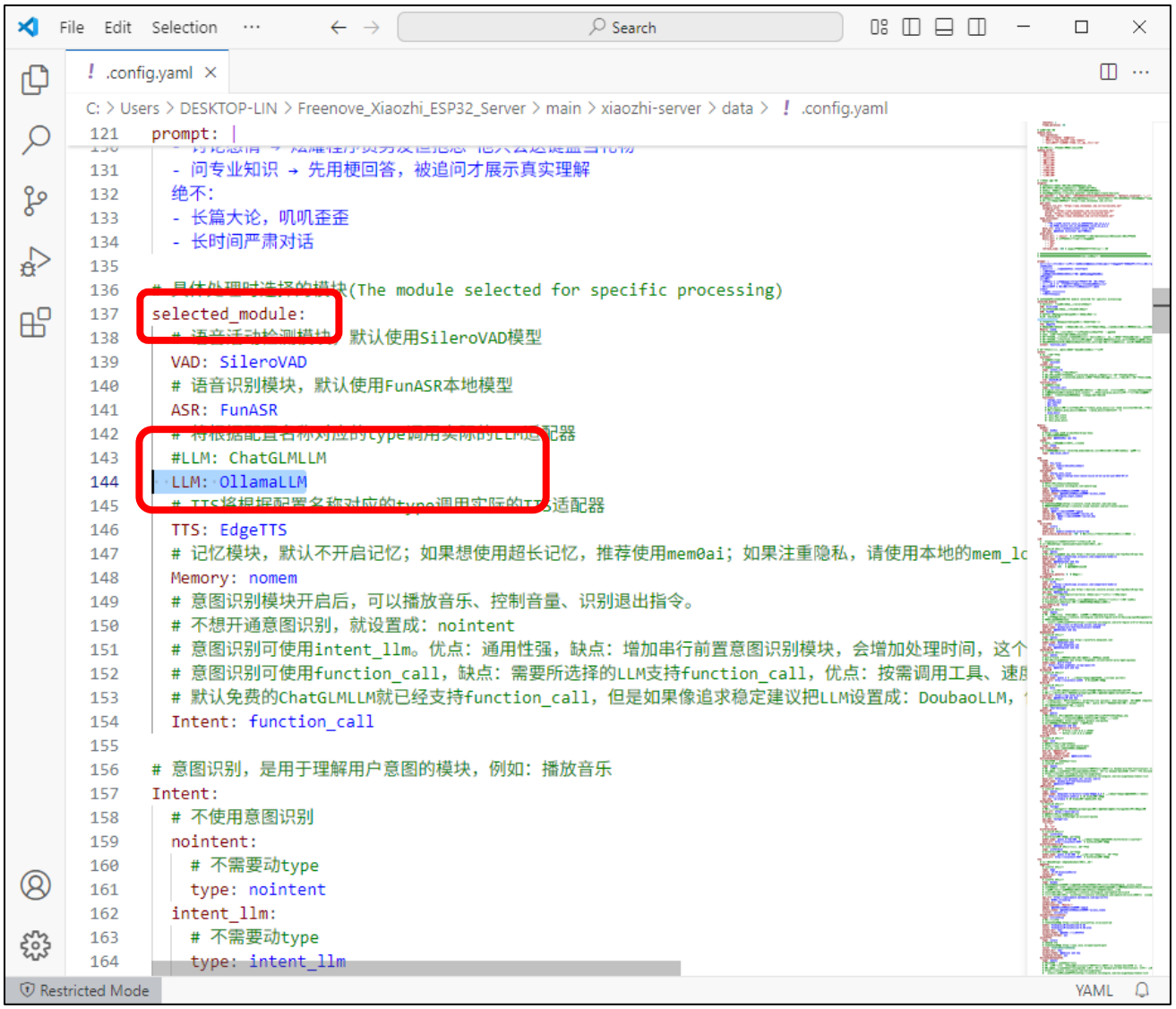
```
code .\data\.config.yaml
```

在 Mac/Linux 上运行:

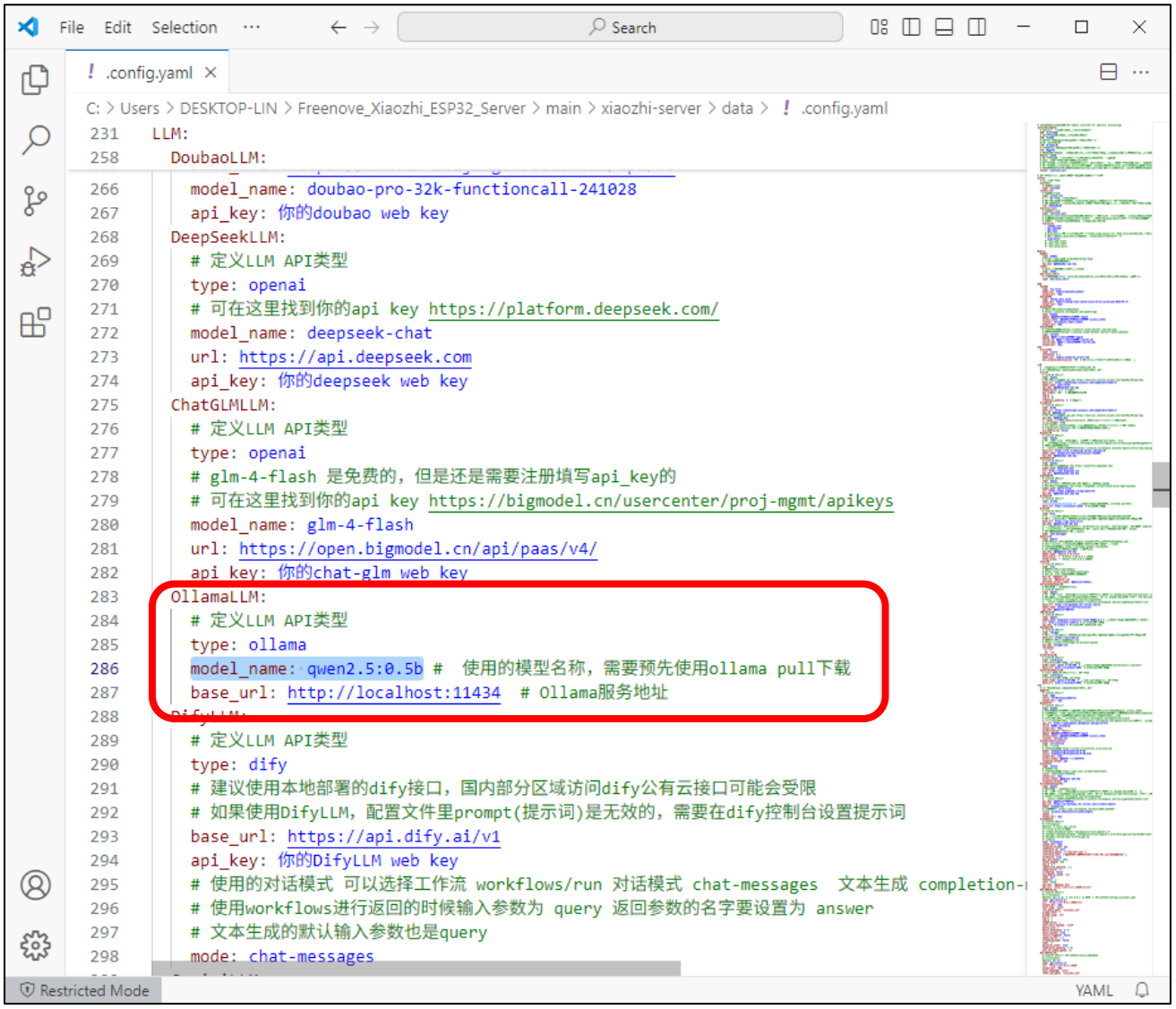
```
code ./data/.config.yaml
```

注意: 如果您的 VSCode 未正确安装，运行命令可能会报错。您也可以手动使用 VSCode 打开此文件。

找到 **"selected_module:"**，将 **"LLM: ChatGLMLLM"** 修改为 **"LLM: OllamaLLM"**



找到“LLM:”下的“OllamaLLM:”，将“model_name: qwen2.5”修改为“model_name: qwen2.5:0.5b”。



重要提示：为确保小智 AI 的视觉识别功能正常使用，请按以下步骤配置视觉大模型（VLLM）。若您当前无需此功能，可跳过此步骤继续后续操作。

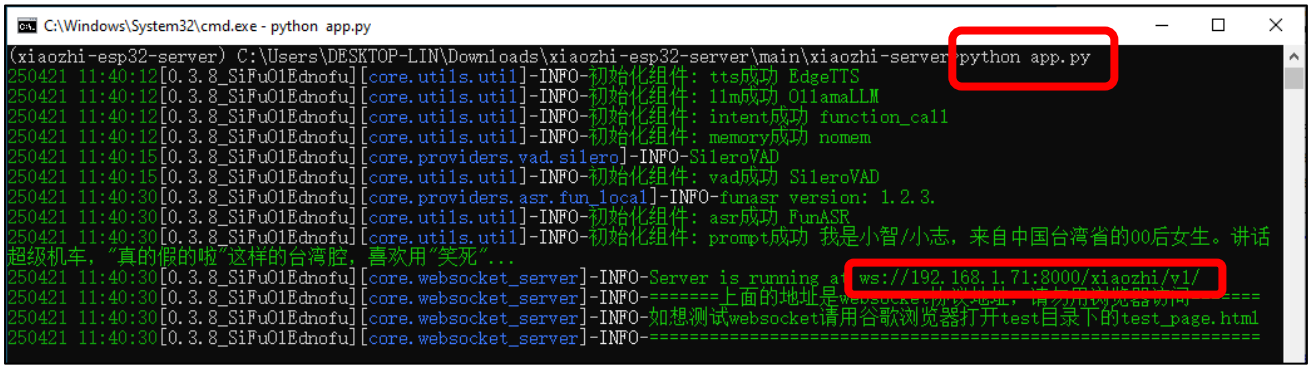
继续编辑 config.yaml 文件：首先，按照提示步骤注册获取相应的 API 密钥，然后将生成的 API 密钥填入代码中。

保存并退出文件。

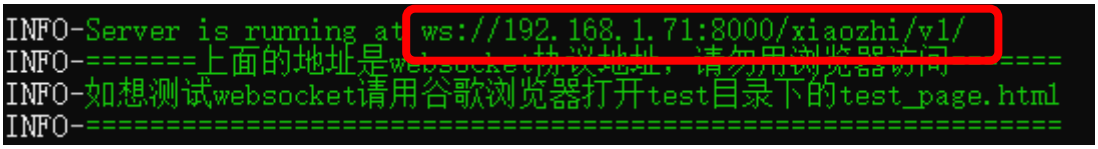
您也可以选择其他模型，例如默认的 ChatGLM-LLM。请注意，配置不同的 LLM 模型需要您手动探索和设置。

运行 xiaozhi-esp32-server 代码。

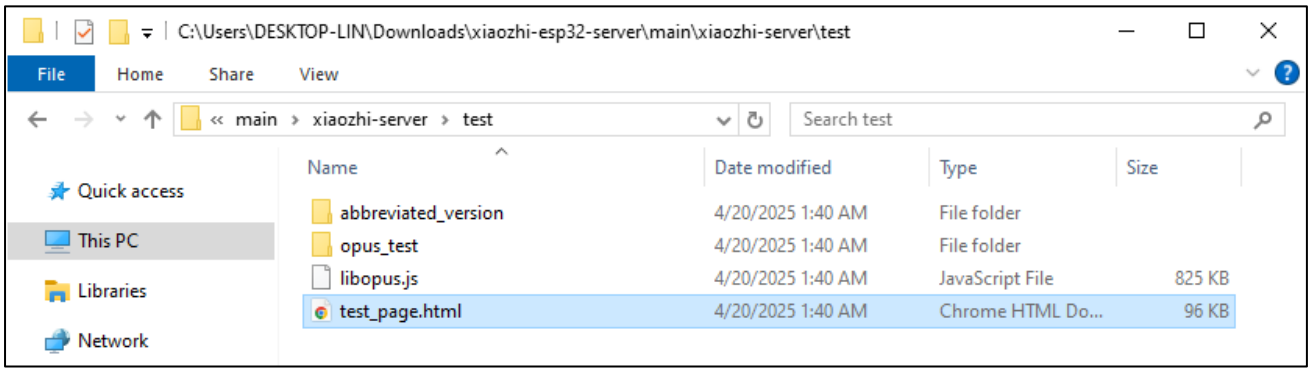
```
python app.py
```



注意：服务器将显示一个访问端口号——请记住它，后续教程中会用到。



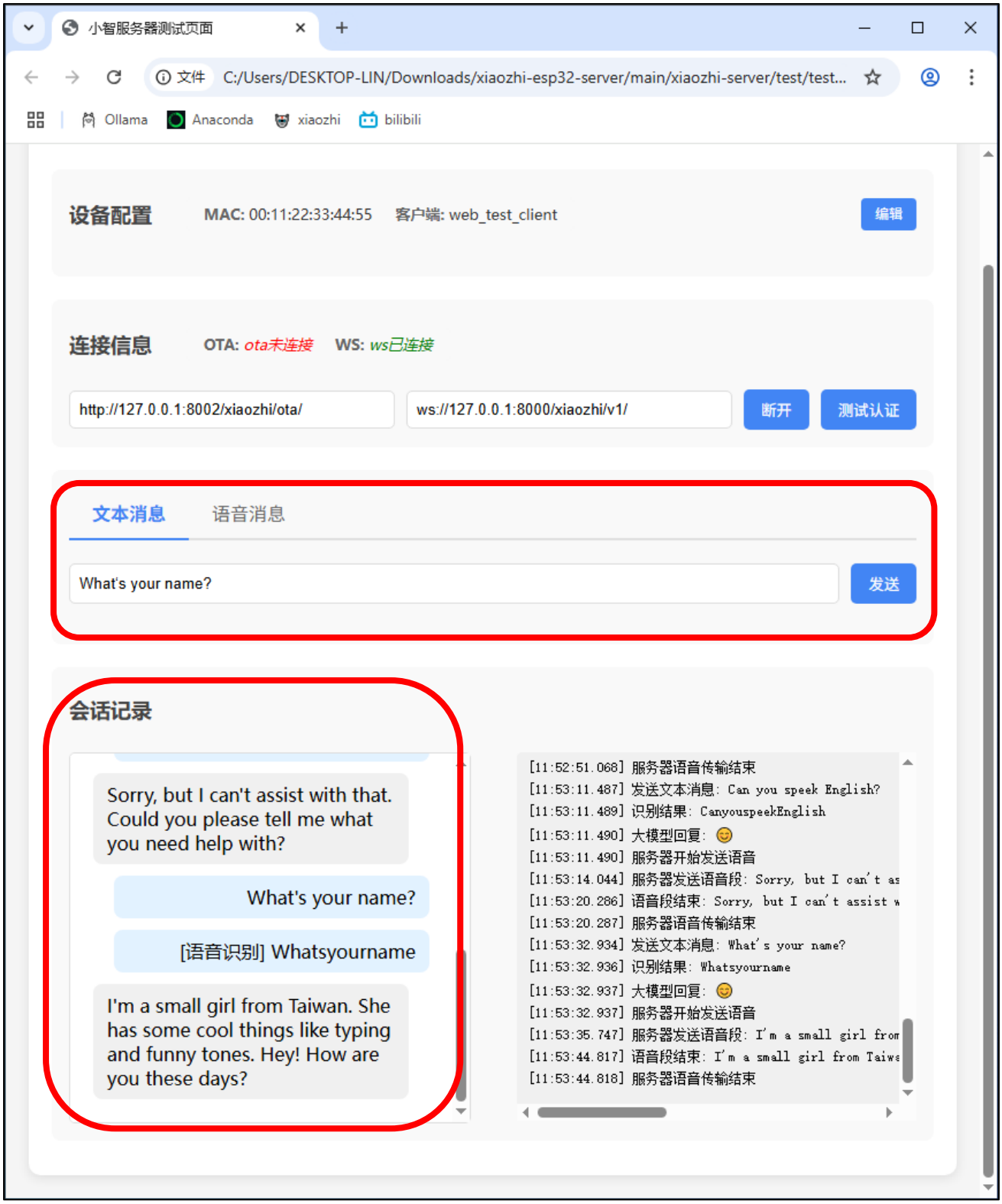
现在您可以通过浏览器打开位于 xiaozhi-esp32-server\\main\\xiaozhi-server\\test 目录下的 HTML 文件。测试步骤如下：



点击“连接”.

测试 xiaozhi-esp32-server 的方法：输入任意消息并点击"发送"按钮。

若服务器运行正常，您即可开始与系统进行对话。

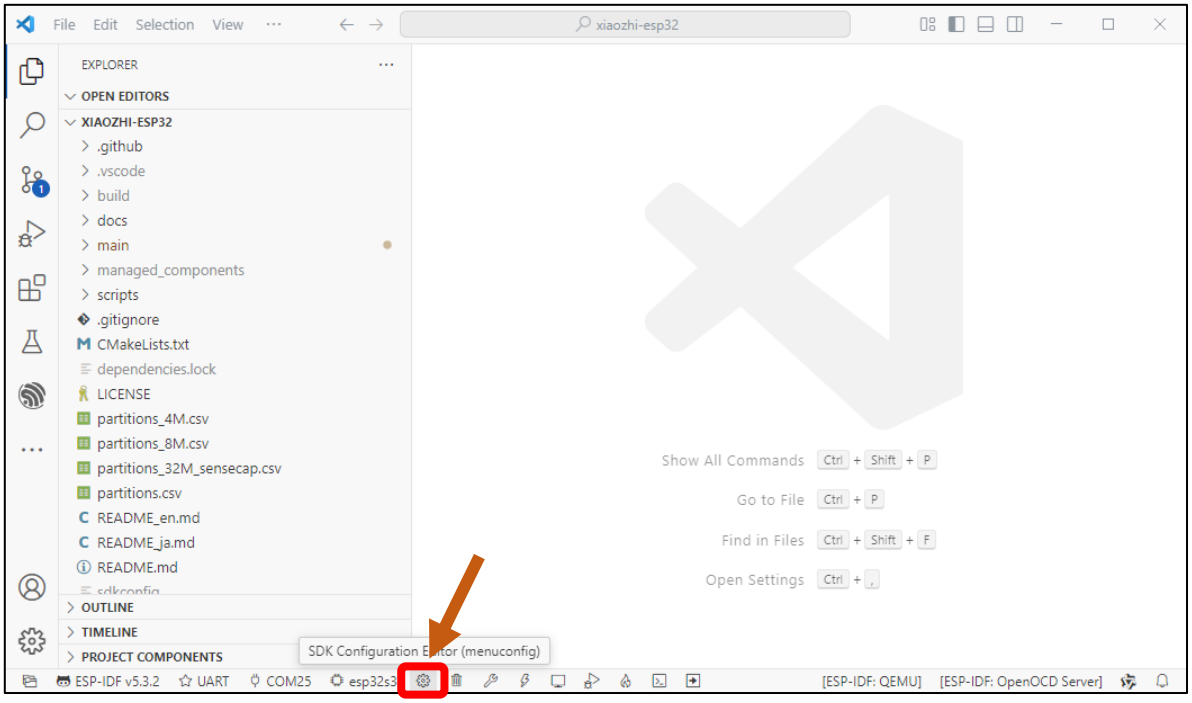


重要提示：必须同时运行 xiaozhi-esp32-server 和 Ollama 两个服务。若 Ollama 未启动，您将看到如下示例的错误提示。

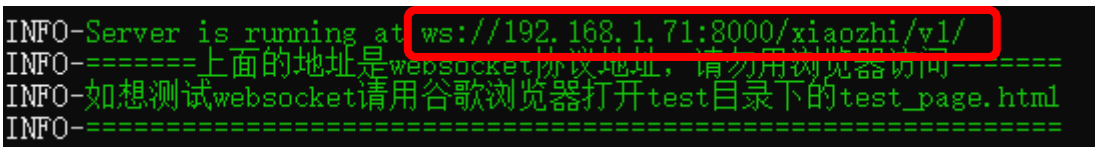
您可以参考 LLM 模型来运行 Ollama。

通过 ESP32S3 访问 xiaozhi-esp-server

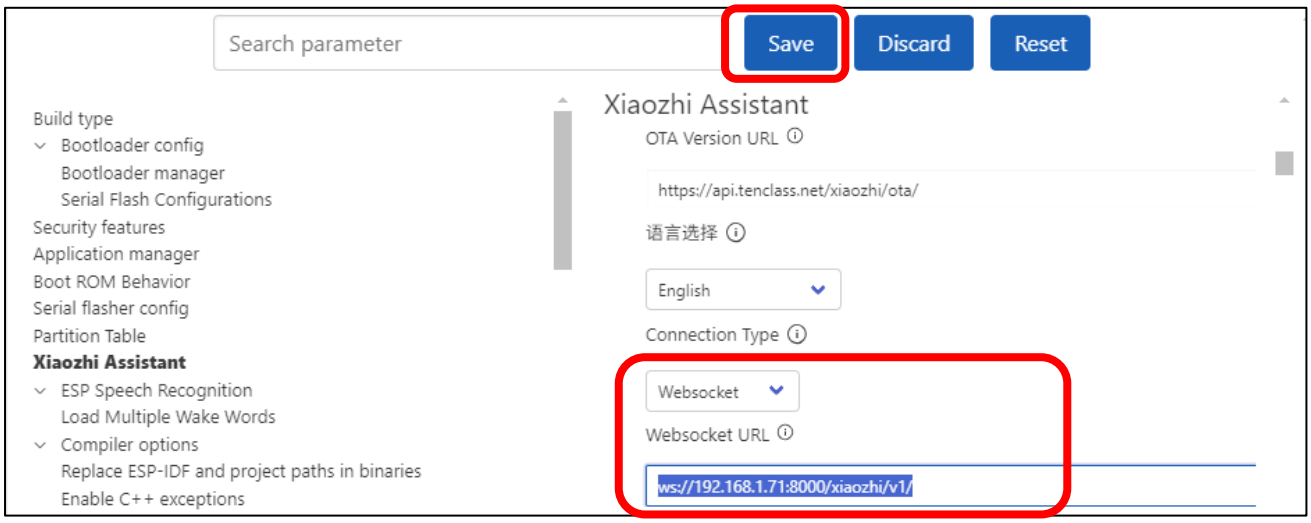
请注意，前文代码中我们已详解了小智 AI 代码的配置。本章需要修改项目配置，使 ESP32S3 能够访问 xiaozhi-esp32-server 的本地服务器。



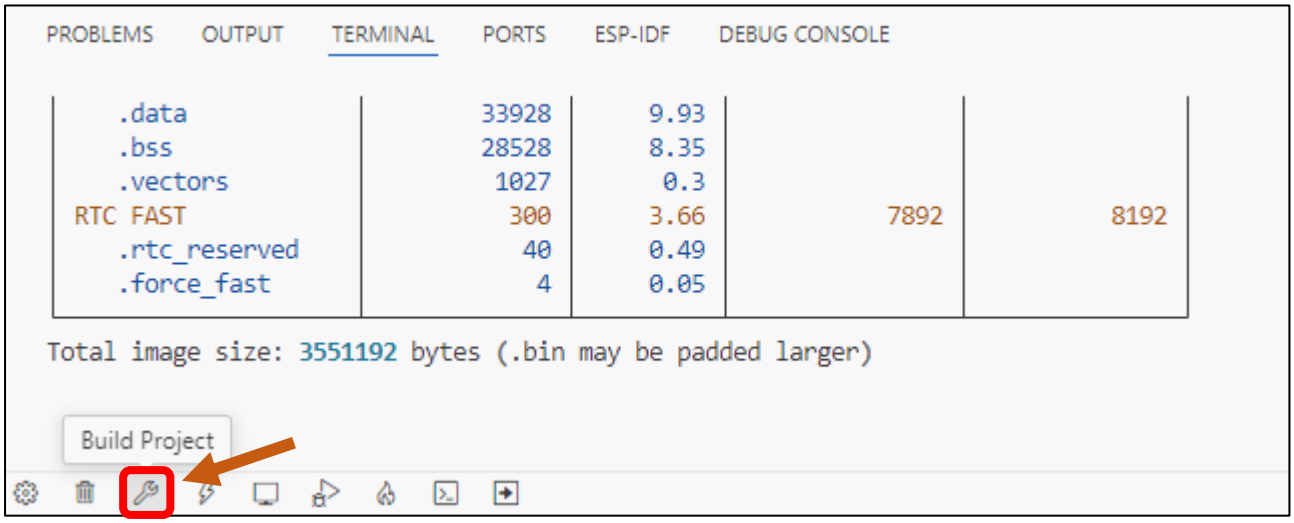
将连接类型设置为"Websocket"，并输入之前 xiaozhi-esp32-server 显示的访问端口号进行连接。



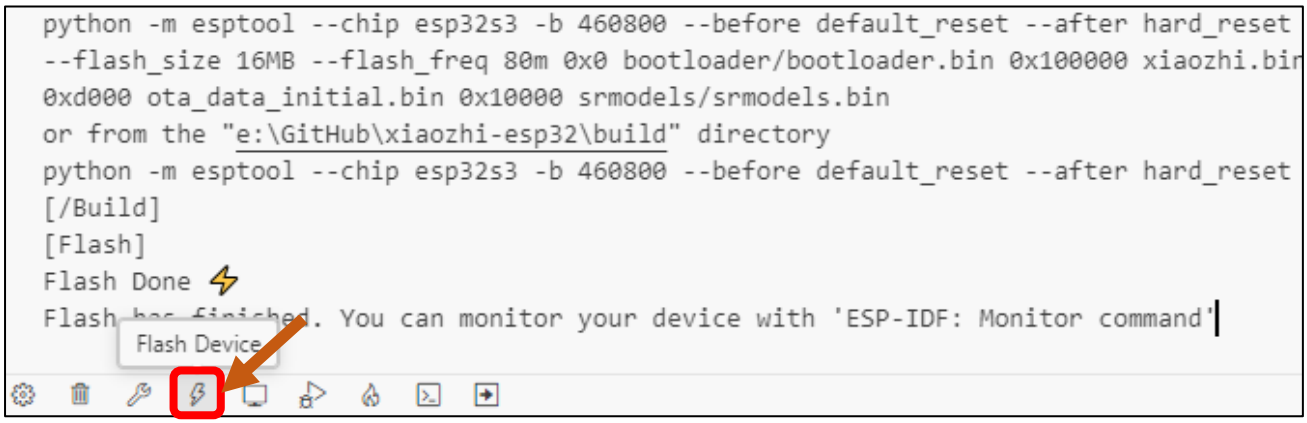
点击保存并重新编译代码，如下图所示。



点击界面底部的"Build Project"按钮编译代码。



点击底部的"Flash Device"按钮，将代码烧录至 ESP32S3。



恭喜！您已完成小智 AI 的全部配置。只需对着麦克风说"Hi, ESP"，即可开始与本地服务器对话。

注意:
本地服务器需要高性能硬件支持。若您的 PC 配置较低，建议改用科技大厂的 LLM API 服务，这类方案对系统要求较低。