1.1.3 Large Language Model Accessing

Note: To proceed with this section, you will need to register on the appropriate website and obtain the API key for the large model (please refer to the file 1.2 Large Language Model Deployment).

It is important to ensure a stable network connection for the development board. For optimal performance, we also recommend connecting the main controller to a wired network for enhanced stability.

1. Environment Configuration

Note: If you have purchased a robot from our company with built-in large model functionality, the environment is already pre-configured in the robot's image. You can directly proceed to Section 3 of this document to configure the API key.

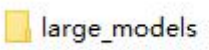
Install Vim and Gedit by running the corresponding commands. Install the necessary software packages and audio libraries required for PyAudio.

```
sudo apt update

sudo apt install vim

sudo apt install gedit

sudo apt install python3 python3-pip python3-all-dev python3-pyaudio portaudio19-dev libsndfile1
```

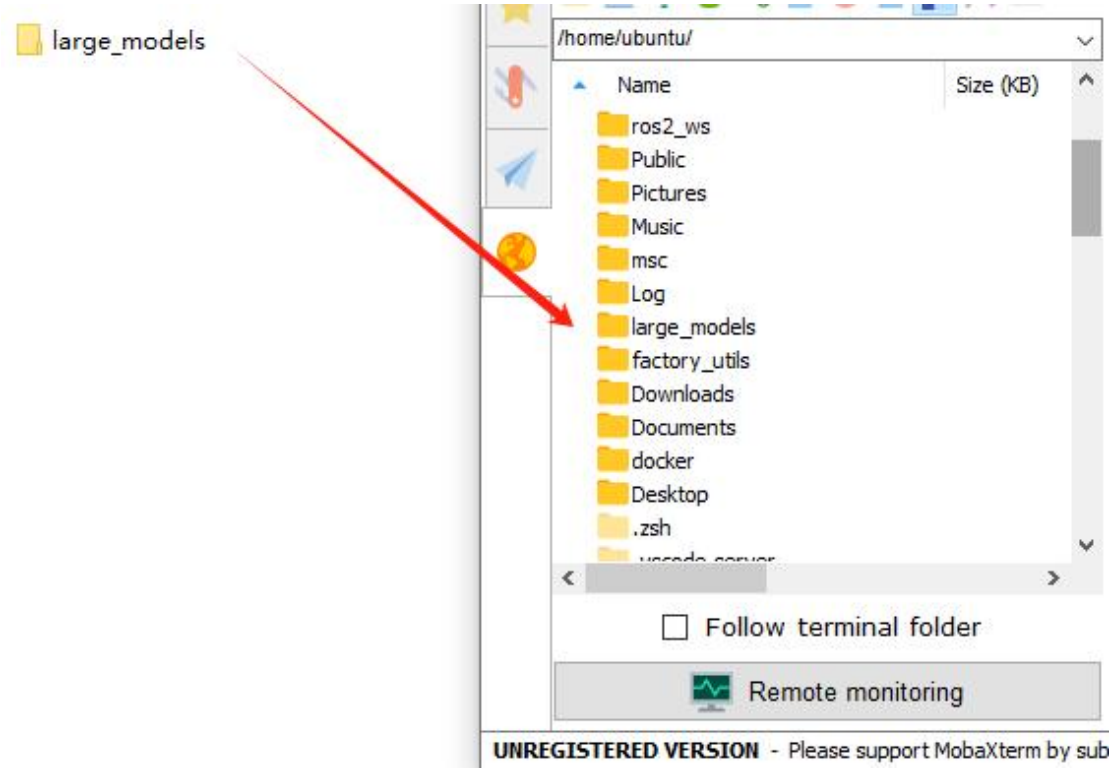
```
> sudo apt update
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Hit:3 https://deb.nodesource.com/node_20.x nodistro InRelease
Hit:4 https://apt.kitware.com/ubuntu jammy InRelease
Get:6 https://download.docker.com/linux/ubuntu jammy/stable arm64 Packages [47.5 kB]
Hit:7 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:9 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [128 kB]
Hit:2 https://repo.download.nvidia.cn/jetson/common r36.3 InRelease
Hit:5 https://repo.download.nvidia.cn/jetson/t234 r36.3 InRelease
Hit:8 https://repo.download.nvidia.cn/jetson/ffmpeg r36.3 InRelease
Get:10 http://packages.ros.org/ros2/ubuntu jammy InRelease [4,682 B]
Get:11 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease [127 kB]
```

2. Importing the Large Model Program Directory

1) In this section, locate the 'Appendix -> 3. Source Code' folder within the same directory as this tutorial document.



2) Using the MobaXterm remote connection tool (as outlined in the '5.5 Remote Access and File Transfer' tutorial), drag the folder into the root directory of the main controller. The software installation package can be found in the '5 Appendix -> 5.5 Remote Access and File Transfer' directory.



3) Next, execute the command to navigate to the 'speech_pkg' directory."

```
cd ~/large_models/speech_pkg/
```

4) Execute the following commands to install the necessary third-party libraries.

```
pip3 install -r requirements.txt --break-system-packages

pip3 install dashscope --break-system-packages

pip3 install opencv-python --break-system-packages

pip3 install sympy==1.13.1 --break-system-packages

pip3 install torch --break-system-packages
```

```
> pip3 install dashscope --break-system-packages
```

```
> pip3 install opencv-python --break-system-packages
```

```
> pip3 install sympy==1.13.1 --break-system-packages
```

```
> pip3 install torch --break-system-packages
```

5) Then, use the command in the terminal to navigate to the 'speech' directory.

```
cd ~/large_models/speech_pkg/speech
```

```
> cd ~/large_models/speech_pkg/speech
```

6) Run the command to list the files in the 'speech' directory.

```
ls
```

```
> ls
awake.py  __init__.py  jetson_nano  jetson_orin  resources  rpi5  rpi5_docker  speech.so
```

7) Depending on the type of main controller and Python version you're using, switch to the appropriate folder for packaging and distribution. This tutorial uses the Jetson Orin controller as an example.

Type of main controller	Python version
jetson_nano	3.6
jetson_orin	3.10
rpi5	3.11
rpi5_docker	3.8

8) Execute the following command to navigate to the Jetson Orin folder.

```
cd jetson_orin/
```

9) Enter the command to copy the 'speech.so' file to the parent directory.

```
cp -r speech.so ..
```

10) Enter the command to navigate to the parent directory.

```
cd ../..
```

11) Execute the command to package the speech file for the Python environment.

```
pip3 install .
```

12) Enter the command to install and update the OpenAI Python library.

```
pip3 install openai -U
```

3. Key Configuration

1) Open the terminal and enter the following command to navigate to the directory for configuring the large model keys:

```
cd ~/large_models
```

```
> cd ~/large_models
```

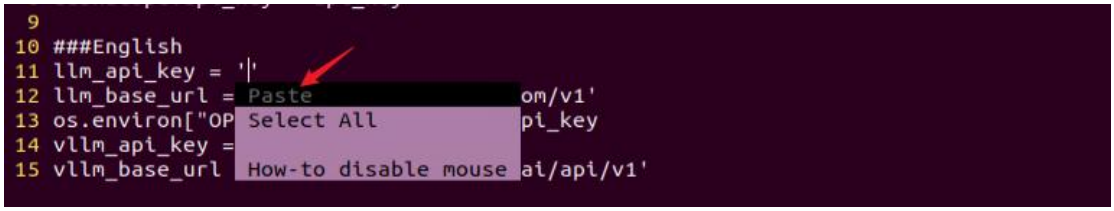
2) Then, open the configuration file by running:

vim config.py

```
> vim config.py
```

3) Once the file is open, configure the OpenAI and OpenRouter keys by filling in the llm_api_key and vllm_api_key parameters, respectively (you can obtain these keys from the '1.2 Large Language Model Deployment' course).

For instance, copy the key created in Section 1.2 of this chapter and paste it into the appropriate field. To paste the key, place the cursor between the quotation marks, hold the "Shift" key, right-click, and select "Paste" (Note: Do not mix keys from different models, as this may cause the functionality to malfunction).



4) After pasting, press the 'Esc' key, then type the following command and press Enter to save the file:

:wq

4. Running the Demo Program

Once the keys are configured, you can run the demo program (openai_llm_demo.py) to experience the text generation capabilities of the large model. For example, the program's prompt might be: 'Write a 50-word article about how technology is changing life.'

```
5 from config import *
6 from speech import speech
7
8 client = speech.OpenAIAPI(llm_api_key, llm_base_url)
9
10 # print(client.llm_origin('Write a 50-word article about changing life with technology', prompt='', model='gpt
   -4o-mini'))
11 print(client.llm('Write a 50-word article about changing life with technology', prompt='', model='gpt-4o-mini'
   )) # https://platform.openai.com/docs/models
```

1) To run the demo, enter the following command in the terminal:

```
python3 openai_llm_demo.py
```

2) After running the program, the output will appear as shown in the image below.