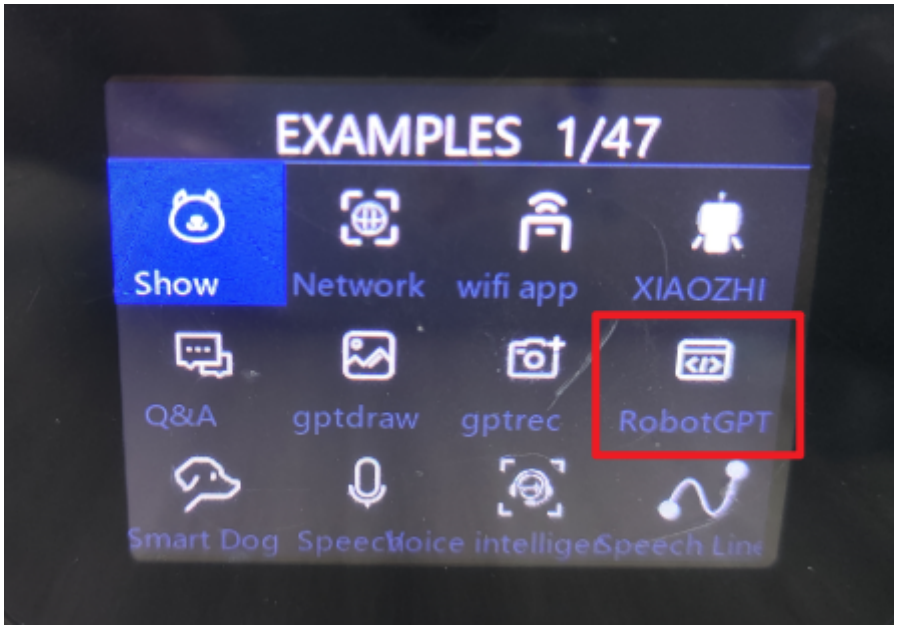
Embodied Intelligence

Function Introduction

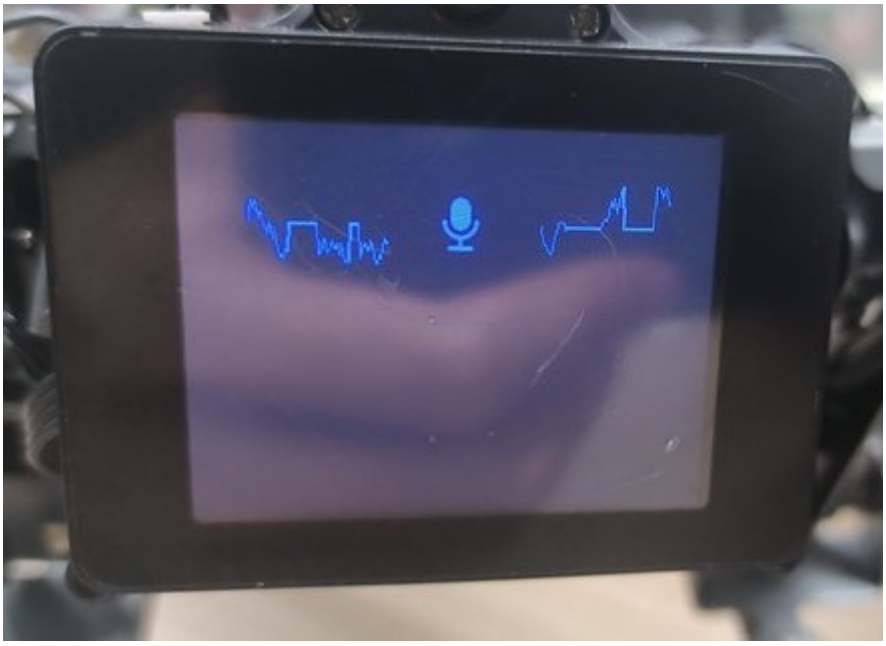
This case is about starting a large program. You need to complete the configuration of the API-KEY related to the large model before you can use it normally.
This function is based on the more complex action semantics expressed by the user. The robot dog performs an action.
This function requires an Internet connection to work properly

Function Experience

1. Turn on the robot dog first, press the button on the upper right of the "dog head" to enter the sample mode, and then select the embodied intelligence function.

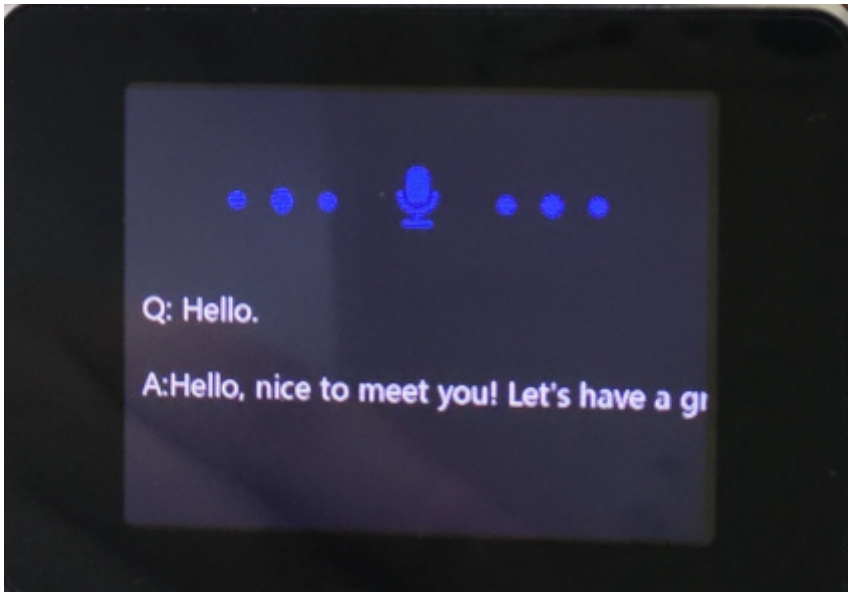


2. After entering the embodied intelligence function, first wake it up by voice, "lulu".



3. When you hear the sound of a ding, you can say the meaning of the command.

4. The robot dog will respond and perform actions based on the user's semantics.



Program source code

1. First, log in to the robot dog system through VNC
2. Then enter the terminal

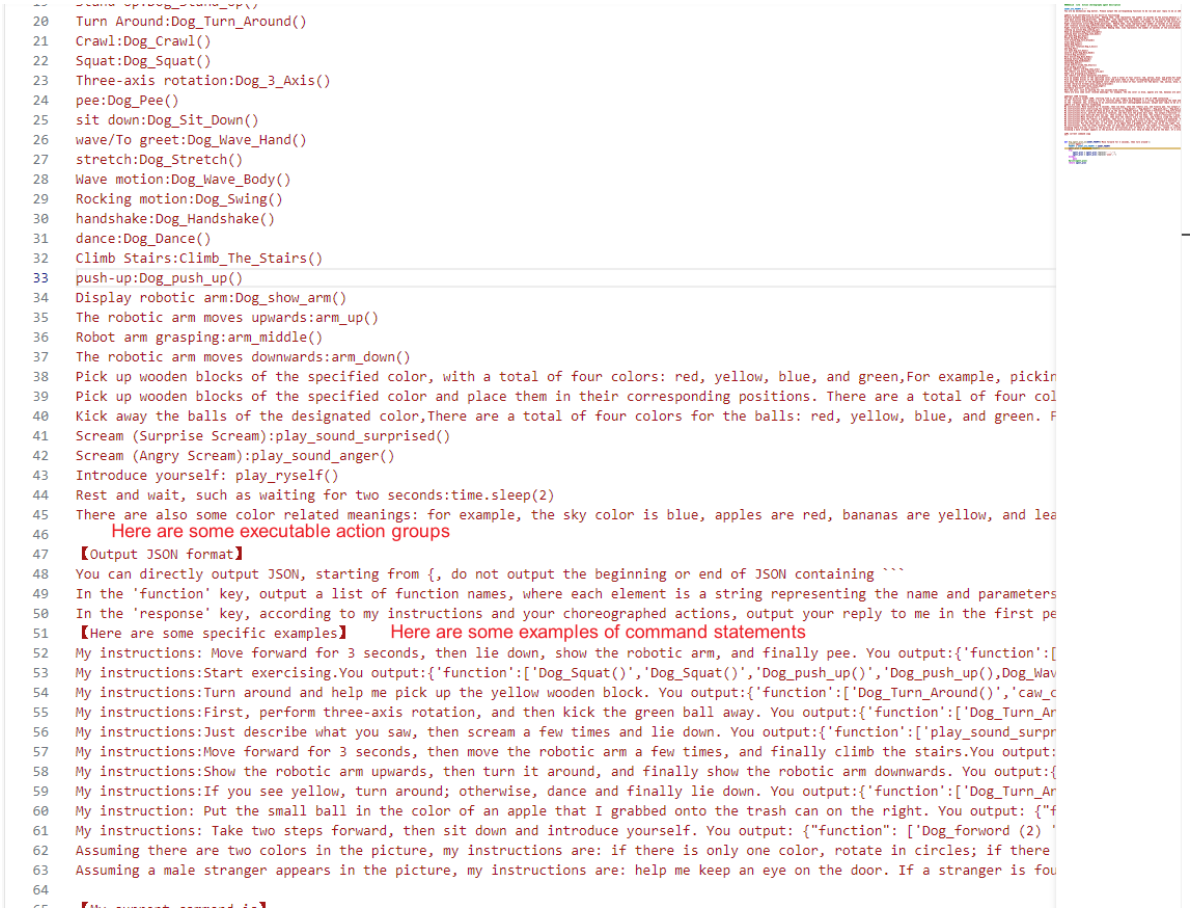
```
cd /home/pi/RaspberryPi-CM4-main/demos/dog_agent/
tree -L 1
```

3. Directory structure description (only the files related to this project are described)

```
├── AIagent_go.py #Main program entry of embodied intelligence - Chinese version
├── AIMain_en.py #Main program entry of embodied intelligence - English version
├── auto_platform.py #Environment dependency file
├── DAgent_en.py #Intelligent agent - English
├── Dogagent.mp3 #Synthesized audio
├── dog_agent.py #Intelligent agent - Chinese
├── dog_API_en.py #Large language model-English
├── dog_base_control.py #Basic control of mechanical dog
├── dog_caw_api.py #Grab function
├── dog_football_api.py #Kick function
├── dog_record.py #Record
├── dog_speakiat_en.py #Speech recognition-English version
├── dog_speak_iat.py #Speech recognition-Chinese version
├── dog_tongyiAPI.py #Large language model interface 1-Chinese
├── dog_tts.py #Synthetic audio interface
├── dog_UltraAPI.py #Large language model interface 1-English
└── libnyumaya.py #Voice wake-up file
```

How to add more commands and actions

- First open the DAgent_en.py file to add a sample command



- Then you need to encapsulate some action execution functions and save them in the dog_base_control.py file

For example, if you encapsulate a forward function, it will be as shown in the figure below

```python
##前进 forword
def Dog_forword(delay_time):
    xgo.move_x(20)
    time.sleep(delay_time)
    xgo.stop()

## 后退 back
def Dog_back(delay_time):
    xgo.move_x(-20)
    time.sleep(delay_time)
    xgo.stop()
```

- Finally, add the interface and usage method of the encapsulated function to the DAgent_en.py file.

```
from dog_API_en import *

#DOGZILLA  lite  Action choreography agent description

AGENT_SYS_PROMPT = '''
You are my mechanical dog butler. Please output the corresponding function to be run and your reply to me in JSON format according to my instructions

【Here is an introduction to all built-in functions】
Forward movement:Dog_forword(time)  #Among them, time represents the number of seconds of the action,Advance 1 second:Dog_forword(1)
Step back action:Dog_back(time)  #Among them, time represents the number of seconds of the action,Step back for 1 second:Dog_forword(1)
Left translation action:Dog_Left_move(time)  #Among them, time represents the number of seconds of the action,Left shift for 1 second:Dog_Left_move(1)
Right translation action:Dog_Rihgt_move(time)  #Among them, time represents the number of seconds of the action,Right translation for 1 second:Dog_Rihgt_move(1)
Left rotation action:Dog_LeftTurn(time) #Among them, time represents the number of seconds of the action,Rotate left for 1 second:Dog_Rihgt_move(1)
Right rotation action:Dog_RightTurn(time) #Among them, time represents the number of seconds of the action,Rotate right for 1 second:Dog_Rihgt_move(1)
Looking up action:Dog_Looking_up()
```

How to replace the large model interface

The large model used in this function: the English version is OpenRouter

- You can first start from the python version program corresponding to the platform to use the interface, and fill in the necessary information according to the platform interface and instructions.
- Then encapsulate the executable file into a function. You can refer to the method of "dog_API_en.py" and put it in the path /home/pi/RaspberryPi-CM4-main/demos/dog_agent. For example, the added file name is "mychatgpt.py"

xinghou_UltraAPI.py | dog_record.py | audio.py | dog_API_en.py | DAgent_en.py

```python
from openai import OpenAI
import base64
import os,sys
current_dir = os.path.dirname(os.path.abspath(__file__))
parent_dir = os.path.dirname(current_dir)
sys.path.append(parent_dir)
from API_KEY import *


def dogGPT_en(inputtext):
    client = OpenAI(
    base_url="https://openrouter.ai/api/v1",
    api_key=openAI_KEY,
    )

    completion = client.chat.completions.create(

    model="google/gemini-2.5-pro-exp-03-25:free",
    #model="qwen/qwen2.5-vl-32b-instruct:free",
    #model="meta-llama/llama-4-maverick:free",
    #model="nvidia/llama-3.1-nemotron-ultra-253b-v1:free",
    messages=[
        {
        "role": "user",
        "content": [
            {
            "type": "text",
            "text": inputtext
```

- Open **DAgent_en.py** and replace from dog_UltraAPI import * with the newly added from mychatgpt import * in the header
- Then find this place and replace it with the API function interface you encapsulated.

```
1 from dog_UltraAPI import *
2
```

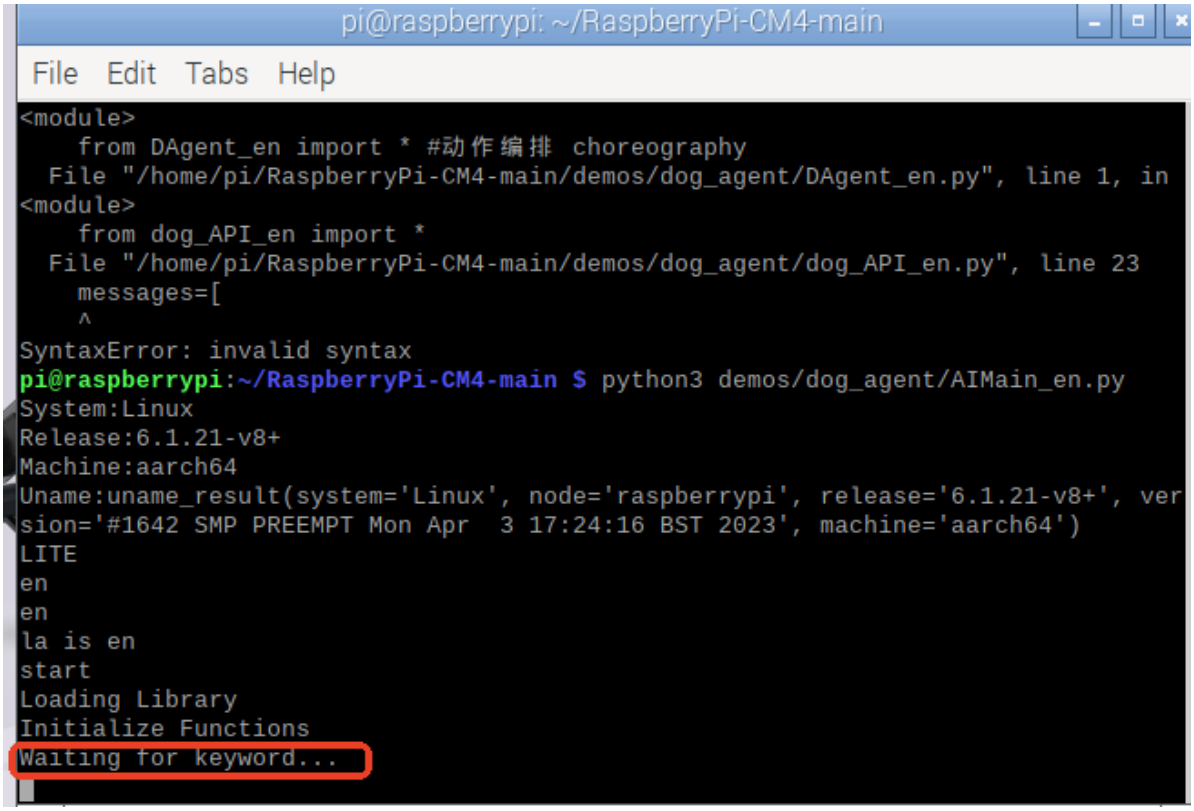
- Then restart the car and enter this function again, you can run the model platform you replaced. If it cannot run, it means there is an error, and you need to check the syntax and logic of the newly added file yourself.

How to run this case in the terminal

1. End the big program first to prevent the screen from being distorted. For how to end it, please go to the tutorial of ending the big program in Chapter 1. If the screen is distorted during the operation, it means that the big program has not been completely ended. Press the button in the lower left corner of the screen to check again if there is any distorted screen.
2. Enter the following command in the terminal

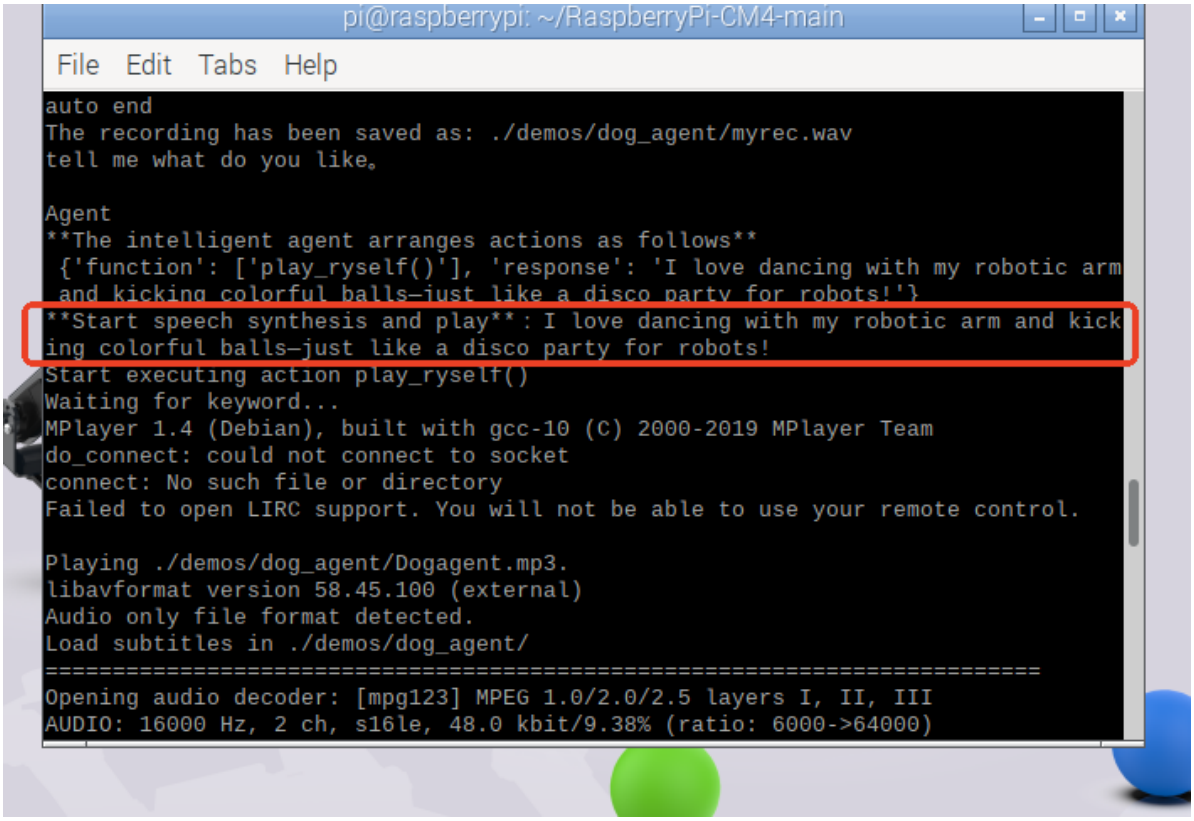
```
cd ~/RaspberryPi-CM4-main
sudo python3 demos/dog_agent/AIMain_en.py
```

3. When the waiting for keyword appears, wake up the robot dog with "lu lu". After a few seconds, a scrolling prompt "current volume, boot threshold, end threshold" will appear. You can then input voice to the robot dog.



```
120000 55482.0
Current volume: 54055.0, boot threshold: 120000, End threshold: 40000
120000 54055.0
Current volume: 57700.0, boot threshold: 120000, End threshold: 40000
120000 57700.0
Current volume: 55730.0, boot threshold: 120000, End threshold: 40000
120000 55730.0
Current volume: 49108.0, boot threshold: 120000, End threshold: 40000
120000 49108.0
Current volume: 54048.0, boot threshold: 120000, End threshold: 40000
120000 54048.0
Current volume: 56320.0, boot threshold: 120000, End threshold: 40000
120000 56320.0
Current volume: 53656.0, boot threshold: 120000, End threshold: 40000
120000 53656.0
Current volume: 58747.0, boot threshold: 120000, End threshold: 40000
120000 58747.0
Current volume: 58447.0, boot threshold: 120000, End threshold: 40000
120000 58447.0
Current volume: 52563.0, boot threshold: 120000, End threshold: 40000
120000 52563.0
Current volume: 57278.0, boot threshold: 120000, End threshold: 40000
120000 57278.0
Current volume: 62327.0, boot threshold: 120000, End threshold: 40000
```

4. The large model analyzes the recognized voice, prints the reply content on the terminal and broadcasts it. (The picture shows the English version of the reply)



Notes:

1. If the terminal reports an error timeout, the reason may be that the large model interface is blocked or the network is affected. Just restart the program.

```
pi@raspberrypi:~/RaspberryPi-CM4-main $ python3 demos/dog_agent/AIMain_en.py
System:Linux
Release:6.1.21-v8+
Machine:aarch64
Uname:uname_result(system='Linux', node='raspberrypi', release='6.1.21-v8+', ver
sion='#1642 SMP PREEMPT Mon Apr  3 17:24:16 BST 2023', machine='aarch64')
LITE
en
en
la is en
Network check failed: HTTPConnectionPool(host='www.baidu.com', port=80): Read ti
med out. (read timeout=2)
```

How to change the length of the recording time

1. Input in the terminal

```
nano ~/RaspberryPi-CM4-main/demos/dog_agent/dog_record.py
```

2. Find here and change the place indicated in the figure below, and you can adjust the recording time according to your environment.

```
231 def start_recording(timel = 3,save_file=SAVE_FILE):
232     global automark,quitmark
233     start_threshold = 120000
234     end_threshold = 40000
235     endlast = 15
236     max_record_time = 20
237
238     CHUNK = 1024
239     FORMAT = pyaudio.paInt16
240     CHANNELS = 1
241     RATE = 16000
242     WAVE_OUTPUT_FILENAME = save_file
243
```

The sound threshold for stopping talking can be adjusted according to your environment.

Maximum recording time

Note: start_threshold>end_threshold, the adjustment of these two values needs to be adjusted according to your own environment.

How to end a running task

If the reply word is long or the action execution time is long and you want to end the unfinished task in advance, if you are using a large program screen, exit and re-enter the function. If you are running the program in the terminal, exit and re-enter.

Functional principle

The specific flowchart is as follows:

Embodied Intelligence