

Embodied Intelligence

Embodied Intelligence

- [Function Introduction](#)
- [Function Experience](#)
- [Program source code](#)
- [How to add more commands and actions](#)
- [How to replace the large model interface](#)
- [How to run this case in the terminal](#)
- Notes:
 - [How to change the length of the recording time](#)
 - [How to end a running task](#)
- [Functional principle](#)

Function Introduction

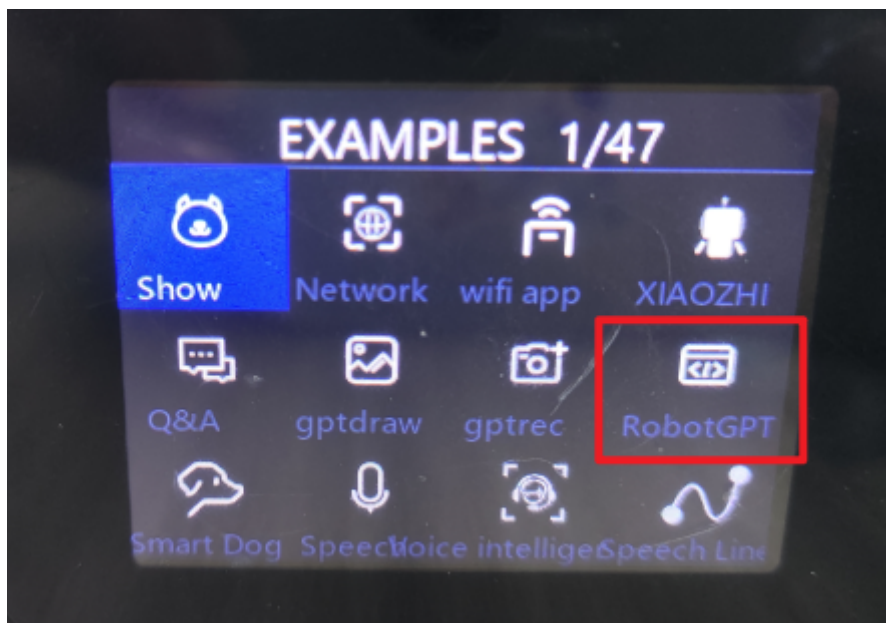
This case is about starting a large program. You need to complete the configuration of the API-KEY related to the large model before you can use it normally.

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.



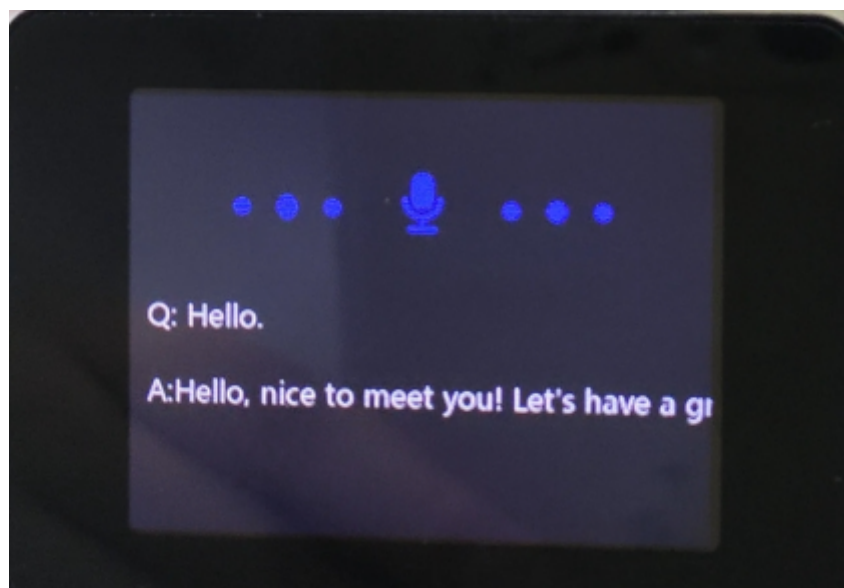
2. After entering the embodied intelligence function, first wake it up by voice, "lulu".



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)
 - ├── Alagent_go.py #Main program entry of embodied intelligence - Chinese version
 - ├── AMain_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

```
20 Turn_Around:Dog_Turn_Around()
21 Crawl:Dog_Crawl()
22 Squat:Dog_Squat()
23 Three-axis rotation:Dog_3_Axis()
24 pee:Dog_Pee()
25 sit down:Dog_Sit_Down()
26 wave/To greet:Dog_Wave_Hand()
27 stretch:Dog_Stretch()
28 Wave motion:Dog_Wave_Body()
29 Rocking motion:Dog_Swing()
30 handshake:Dog_Handshake()
31 dance:Dog_Dance()
32 Climb Stairs:Climb_The_Stairs()
33 push-up:Dog_push_up()
34 Display robotic arm:Dog_show_arm()
35 The robotic arm moves upwards:arm_up()
36 Robot arm grasping:arm_middle()
37 The robotic arm moves downwards:arm_down()
38 Pick up wooden blocks of the specified color, with a total of four colors: red, yellow, blue, and green,For example, pickin
39 Pick up wooden blocks of the specified color and place them in their corresponding positions. There are a total of four col
40 Kick away the balls of the designated color,There are a total of four colors for the balls: red, yellow, blue, and green. F
41 Scream (Surprise Scream):play_sound_surprised()
42 Scream (Angry Scream):play_sound_anger()
43 Introduce yourself: play_rysel()
44 Rest and wait, such as waiting for two seconds:time.sleep(2)
45 There are also some color related meanings: for example, the sky color is blue, apples are red, bananas are yellow, and lea
46 Here are some executable action groups
47 【Output JSON format】
48 You can directly output JSON, starting from {, do not output the beginning or end of JSON containing ```
49 In the 'function' key, output a list of function names, where each element is a string representing the name and parameters
50 In the 'response' key, according to my instructions and your choreographed actions, output your reply to me in the first pe
51 【Here are some specific examples】 Here are some examples of command statements
52 My instructions: Move forward for 3 seconds, then lie down, show the robotic arm, and finally pee. You output: {'function':[
53 My instructions:Start exercising.You output: {'function': ['Dog_Squat()', 'Dog_Squat()', 'Dog_push_up()', 'Dog_push_up()', 'Dog_Wav
54 My instructions:Turn around and help me pick up the yellow wooden block. You output: {'function': ['Dog_Turn_Around()', 'caw_c
55 My instructions:First, perform three-axis rotation, and then kick the green ball away. You output: {'function': ['Dog_Turn_Ar
56 My instructions:Just describe what you saw, then scream a few times and lie down. You output: {'function': ['play_sound_surpr
57 My instructions:Move forward for 3 seconds, then move the robotic arm a few times, and finally climb the stairs.You output:
58 My instructions:Show the robotic arm upwards, then turn it around, and finally show the robotic arm downwards. You output: {
59 My instructions:If you see yellow, turn around; otherwise, dance and finally lie down. You output: {'function': ['Dog_Turn_Ar
60 My instruction: Put the small ball in the color of an apple that I grabbed onto the trash can on the right. You output: {"f
61 My instructions: Take two steps forward, then sit down and introduce yourself. You output: {"function": ['Dog_forword (2) '
62 Assuming there are two colors in the picture, my instructions are: if there is only one color, rotate in circles; if there
63 Assuming a male stranger appears in the picture, my instructions are: help me keep an eye on the door. If a stranger is fou
64
65
```



- 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

```

15  ##前进 forward
16  def Dog_forward(delay_time):
17      xgo.move_x(20)
18      time.sleep(delay_time)
19      xgo.stop()
20
21  ## 后退 back
22  def Dog_back(delay_time):
23      xgo.move_x(-20)
24      time.sleep(delay_time)
25      xgo.stop()
--

```

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

```


1  from dog_API_en import *
2
3  #DOGZILLA lite Action choreography agent description
4
5  AGENT_SYS_PROMPT = '''
6  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
7
8  [Here is an introduction to all built-in functions]
9  Forward movement:Dog_forward(time) #Among them, time represents the number of seconds of the action,Advance 1 second:Dog_forward(1)
10 Step back action:Dog_back(time) #Among them, time represents the number of seconds of the action,Step back for 1 second:Dog_forward(1)
11 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)
12 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)
13 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)
14 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)
15 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 x dog_record.py x audio.py x dog_API_en.py x DAgent_en.py x
1  from openai import OpenAI
2  import base64
3  import os,sys
4  current_dir = os.path.dirname(os.path.abspath(__file__))
5  parent_dir = os.path.dirname(current_dir)
6  sys.path.append(parent_dir)
7  from API_KEY import *
8
9
10 def dogGPT_en(inputtext):
11     client = OpenAI(
12         base_url="https://openrouter.ai/api/v1",
13         api_key=openAI_KEY,
14     )
15
16     completion = client.chat.completions.create(
17
18         model="google/gemini-2.5-pro-exp-03-25:free",
19         #model="qwen/qwen2.5-vl-32b-instruct:free",
20         #model="meta-llama/llama-4-maverick:free",
21         #model="nvidia/llama-3.1-nemotron-ultra-253b-v1:free",
22         messages=[
23             {
24                 "role": "user",
25                 "content": [
26                     {
27                         "type": "text",
28                         "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.

```
pi@raspberrypi: ~/RaspberryPi-CM4-main
File Edit Tabs Help
<module>
  from DAgent_en import * #动作编排 choreography
  File "/home/pi/RaspberryPi-CM4-main/demos/dog_agent/DAgent_en.py", line 1, in
<module>
  from dog_API_en import *
  File "/home/pi/RaspberryPi-CM4-main/demos/dog_agent/dog_API_en.py", line 23
    messages=[
    ^
SyntaxError: invalid syntax
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
start
Loading Library
Initialize Functions
Waiting for keyword...
```

```

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)

```

pi@raspberrypi: ~/RaspberryPi-CM4-main
File Edit Tabs Help
auto end
The recording has been saved as: ./demos/dog_agent/myrec.wav
tell me what do you like.

Agent
**The intelligent agent arranges actions as follows**
{'function': ['play_ryself()'], 'response': 'I love dancing with my robotic arm
and kicking colorful balls—just like a disco party for robots!'}
**Start speech synthesis and play**: I love dancing with my robotic arm and kick
ing colorful balls—just like a disco party for robots!
Start executing action play_ryself()
Waiting for keyword...
MPlayer 1.4 (Debian), built with gcc-10 (C) 2000-2019 MPlayer Team
do_connect: could not connect to socket
connect: No such file or directory
Failed to open LIRC support. You will not be able to use your remote control.

Playing ./demos/dog_agent/Dogagent.mp3.
libavformat version 58.45.100 (external)
Audio only file format detected.
Load subtitles in ./demos/dog_agent/
=====
Opening audio decoder: [mpg123] MPEG 1.0/2.0/2.5 layers I, II, III
AUDIO: 16000 Hz, 2 ch, s16le, 48.0 kbit/9.38% (ratio: 6000->64000)

```

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+', version='#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 timed 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     Maximum recording time
238     CHUNK = 1024
239     FORMAT = pyaudio.paInt16
240     CHANNELS = 1
241     RATE = 16000
242     WAVE_OUTPUT_FILENAME = save_file
243
```

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

