

Embodied intelligence with LiDAR navigation

Embodied intelligence with LiDAR navigation

Precautions before use:

1. Experimental Purpose
 2. Experimental steps
 3. Analysis of the main source code of the program
 4. Experimental summary
- Notes

Precautions before use:

1. This case can only run with radar hardware, otherwise it cannot be used normally.
2. To use this case, you need to build the map that needs to be navigated in advance. For how to build the map, you can refer to **Radar Mapping and Navigation (S2 Exclusive)** This chapter's section **7. ROS2 Environment Entity Robot Mapping** Instructions.
3. **Before running this case, you need to close the startup program**, please refer to the Raspberry Pi system configuration chapter **9. Close and open the APP/handle control program** This tutorial ends the startup program.
4. You need to fill in the API_KEY of the big model, please refer to the operation method of **AI Big Model Chapter "1. Big Model Prerequisite Use Configuration"**.

1. Experimental Purpose

This chapter learns how to combine the online big language model to realize and learn the process of embodied intelligence with navigation. This case requires audio equipment and cameras, supporting virtual machines, docker systems in Raspberry Pi, and Raspberry Pi systems. **If you don't have the basics of docker and linux, learn to supplement these basics first**

2. Experimental steps

1. Terminal input of the virtual machine (use the virtual machine provided with the tutorial, for how to install and open it, please refer to the use of the virtual machine in the previous tutorial)
Enter the following command in a new terminal

#Open relocation

```
ros2 launch yahboom_dog_cartographer localization_imu_odom.launch.py  
load_state_filename:=/home/yahboom/yahboomcar_ws/maps/mymap.pbstream
```

- Open a new terminal and enter

#Open navigation instructions

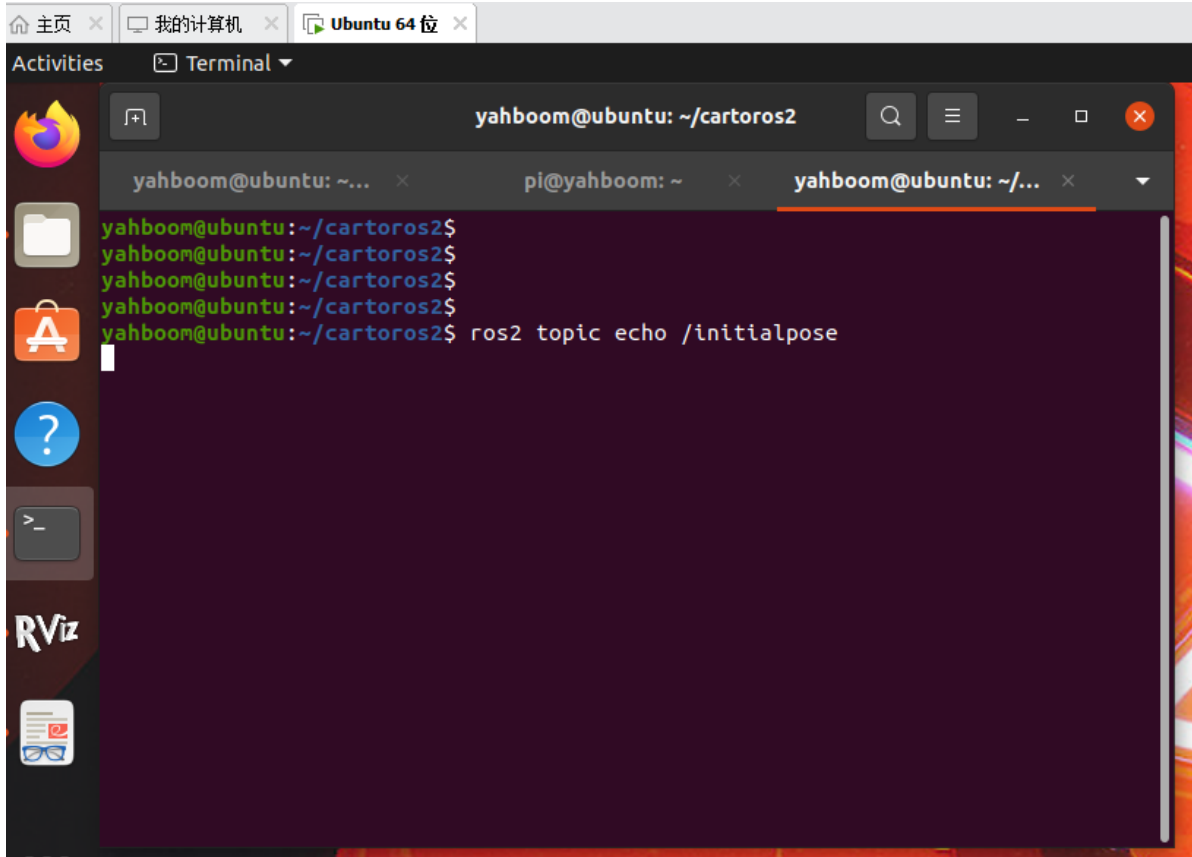
```
ros2 launch yahboom_dog_navigation2 navigation2.launch.py use_sim_time:=False  
map:=/home/yahboom/yahboomcar_ws/maps/mymap.yaml
```

- Open a new terminal and enter

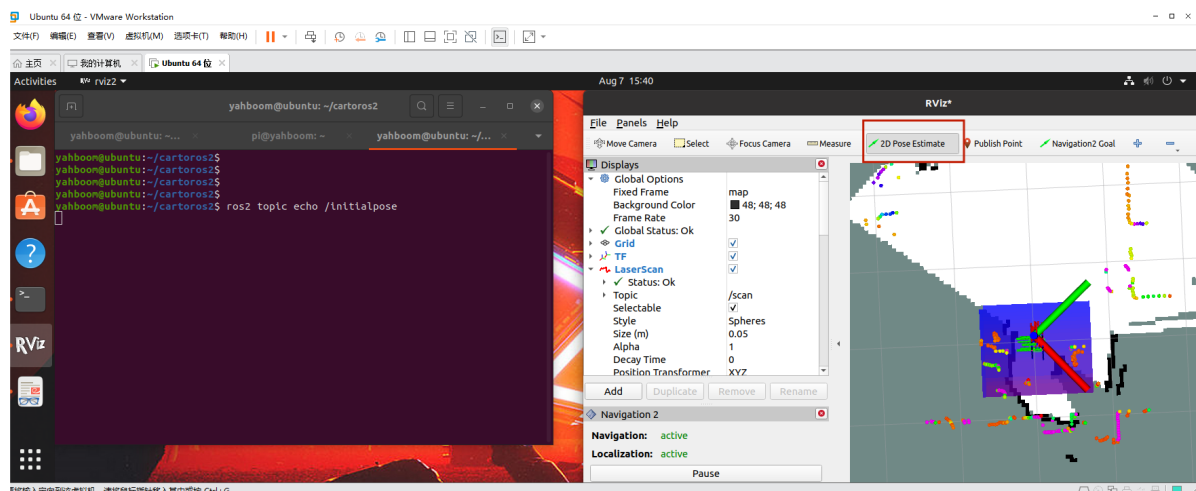
```
ros2 run yahboom_nav2_send_goal yahboom_nav2_send_goal #Open multi-point navigation node
```

- Open a new terminal to input (this terminal is to obtain the location of the navigation point, which is convenient for filling in the subsequent location points)

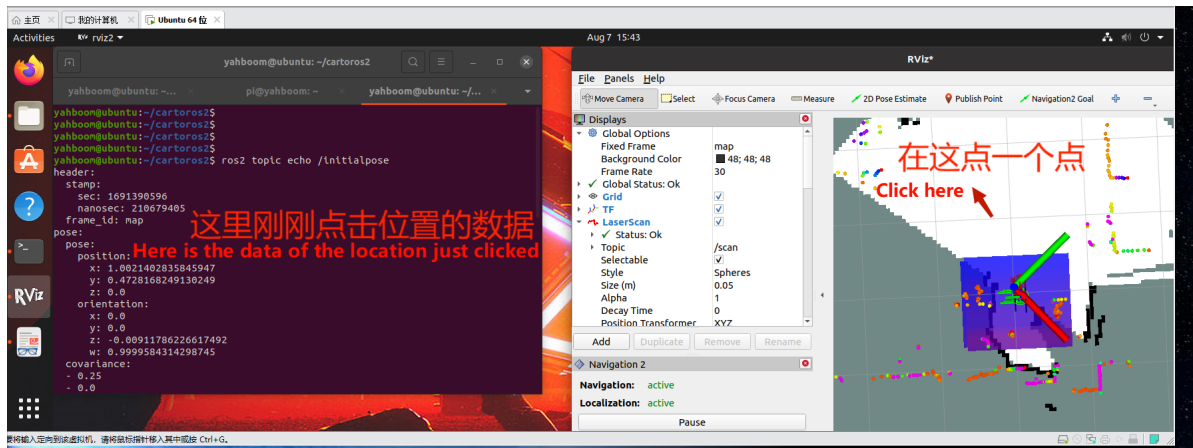
```
ros2 topic echo /initialpose
```



Click the button "2D Pose Estimate" in rviz and click a point on the map where you can walk. At this time, the terminal will print the location of the point you just clicked. (The location of the points filled in later will be used for navigation)



程序输入定向到该虚拟机。请将该程序输入其中或按 Ctrl+G。



2. Input in the terminal of the Raspberry Pi

```
cd /home/pi/DOGZILLA/Samples/4_Big_Modle
python3 ros_dog_agent/Ros_dog_Main_en.py
```

3. Open three new terminals of the Raspberry Pi. All three terminals need to enter the docker (if there is an open docker, there is no need to open a new docker repeatedly. Just enter the same docker with multiple terminals)

- Input in the first terminal

```
cd ~
./run_humble.sh
```

At this time, the first terminal has entered docker

- Enter the second terminal

```
docker ps
```

```
jetson@ubuntu:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9   "/bin/bash"            3 days ago    Up 9 hours                    ecstatic_lewin
jetson@ubuntu:~$
```

Then enter

```
docker exec -it 5b698ea10535 /bin/bash
```

```
jetson@ubuntu:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9   "/bin/bash"            3 days ago    Up 9 hours                    ecstatic_lewin
jetson@ubuntu:~$ docker exec -it 5b698ea10535 /bin/bash
root@ubuntu:/#
```

At this time, the second terminal also successfully entered the docker container, and the third terminal also entered the docker according to the second terminal.

This entry method ensures that these three terminals are in the same docker version container for subsequent operations

4. Terminal input in Raspberry Pi docker
First terminal input:

```
cd ~/ROS_UDP
python3 dog_udp_server.py
```

```
root@raspberrypi:~/ROS_UDP# python3 dog_udp_server.py
[*] UDPServer started, listening 0.0.0.0:5578
```

Second terminal input:

```
ros2 launch bringup Navigation_bringup.launch.py
```

```
root@raspberrypi:/  
File Edit Tabs Help  
at 0x7fff363522f0>  
[yahboomcar_joint_state-3] [13.16, 45.61, 1.34, 10.1, 44.36, -1.09, 10.1, 51.85,  
2.55, 6.53, 51.22, -0.36]  
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&&&&&&&&& 0.10927200317382812  
[yahboomcar_joint_state-3] #####  
[yahboomcar_joint_state-3] [-0.17585449218750002, -0.13996582031250002, -9.72702  
63671875, -1.0365853658536586, -0.426829268292683, -0.6097560975609757, 0.010487  
360583411322, -0.02726797640323639, 5.983139933268229]  
[yahboomcar_joint_state-3] ***** <rclpy.timer.Timer object  
at 0x7fff363522f0>  
[yahboomcar_joint_state-3] [13.16, 45.61, 1.34, 10.1, 44.36, -1.09, 10.1, 51.85,  
2.55, 6.53, 51.22, -0.36]  
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&&&&&&&&& 0.10969948768615723  
[yahboomcar_joint_state-3] #####  
[yahboomcar_joint_state-3] [-0.14475097656250002, -0.131591796875, -9.7401855468  
75, -1.0975609756097562, -0.3658536585365854, -0.6097560975609757, 0.01022947788  
9007993, -0.02749979310565525, 5.983139933268229]  
[yahboomcar_joint_state-3] ***** <rclpy.timer.Timer object  
at 0x7fff363522f0>  
[yahboomcar_joint_state-3] [13.16, 45.61, 1.34, 10.1, 44.36, -1.09, 10.1, 51.85,  
2.55, 6.53, 51.22, -0.36]  
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&&&&&&&&& 0.10920882225036621  
[yahboomcar_joint_state-3] #####
```

Third terminal input:

1. First modify the navigation point, modify it according to the previously printed map navigation point, and add up to 5 navigation points

```
#path
cd ~/yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.10/site-
packages/voice_xgo_ctrl_run
vi ros2_dog_agent.py
```

```
root@raspberrypi:~/ROS_UDP# cd ~/yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.10/site-packages/voice_xgo_ctrl_run
root@raspberrypi:~/yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.10/site-packages/voice_xgo_ctrl_run# vi ros2_dog_agent.py
```

After opening, you can modify the values of position.x, position.y and orientation.w of the point just obtained from the virtual machine. The value here is an example, and the value of your own environment is the standard. You can modify five points, save and exit after modification.

```

def send_pose(self):
    read_result = self.read_file()
    if read_result != self.readreold and read_result!=None:
        self.readflag = 1
        self.readreold = read_result
        print(read_result)

    if self.readflag == 1:
        self.readflag = 0
        print("MSG:"+read_result)
        if re.search("APoint",read_result):
            #导航去1点
            self.goal_pose.header.frame_id = "map"
            self.goal_pose.pose.position.x = 1.4076685905456543
            self.goal_pose.pose.position.y = 0.038116052746772766
            self.goal_pose.pose.orientation.w = 1.0
            self.publisher.publish(self.goal_pose)
            self.active_goal = True
            self.goal_name = "APoint"
        if re.search("BPoint",read_result):
            #导航去2点
            self.goal_pose.header.frame_id = "map"
            self.goal_pose.pose.position.x = 2.610774040222168
            self.goal_pose.pose.position.y = 0.15808261930942535
            self.goal_pose.pose.orientation.w = 1.0
            self.publisher.publish(self.goal_pose)
            self.active_goal = True
            self.goal_name = "BPoint"

```

vi editor operation: After opening the file, enter i to start editing. After editing, press ESC, then enter :wq to save and exit the file!

Run multi-point navigation node

```

cd yahbomcar_ws/
ros2 run voice_xgo_ctrl_run ros2_dog_agent

```

5. Start the wake-up operation

The wake-up word is **hello yahboom**

```

pi@raspberrypi:~/DOGZILLA/Samples/4_Big_Modle $ python3 dog_agent/AIagent_go_image.py
serial /dev/myspeech open
start
Waiting for keyword...

```

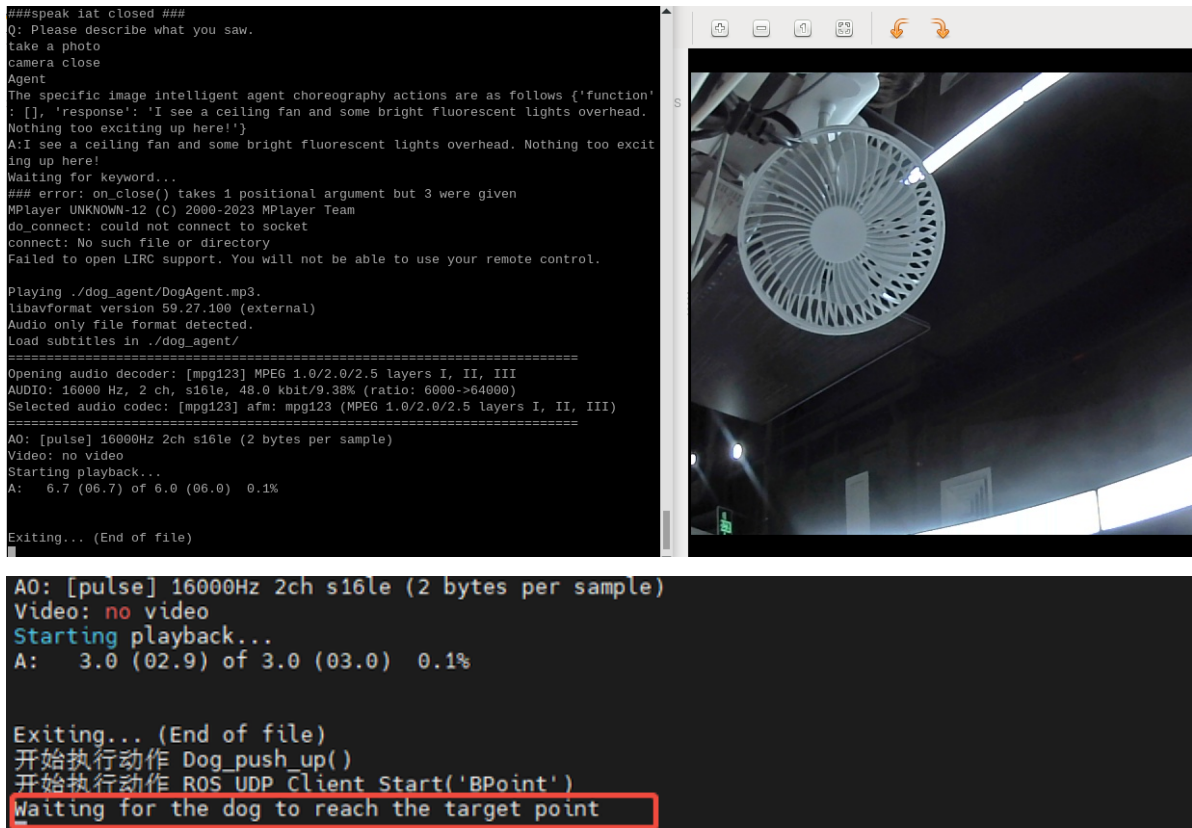
After waking up, you will hear a "ding" sound, and then you can express your thoughts to the robot dog.

```

pi@raspberrypi:~/DOGZILLA/Samples/4_Big_Modle $ python3 dog_agent/AIagent_go_image.py
serial /dev/myspeech open
start
Waiting for keyword...
Keyword detected: 06.Jun 2025 05:22:42
Playing WAVE './ding.wav' : Signed 16 bit Little Endian, Rate 16000 Hz, Mono

```

6. The robot dog will first take a picture of the scene, and then process it through the large model to understand the customer's ideas, and then feedback the corresponding text results and audio playback.
For example: navigate to position 1, then dance, and then navigate to position 2 to see what is there.



7. At this point, the interactive process is over. If you need to express your ideas again, just wake up again.

3. Analysis of the main source code of the program

In the "/home/pi/DOGZILLA/Samples/4_Big_Modle/ros_dog_agent/" path, "Ros_dog_Main_en.py" is a main function entry

```
while True:
    if detect_keyword() and huanxin == 0:
        huanxin = 1

    if os.path.exists('./demos/ros_dog_agent/myrec.wav'):
        os.remove('./demos/ros_dog_agent/myrec.wav')
        time.sleep(0.2)

    try:
        start_recording()
        time.sleep(0.2)
        rectext = rec_wav_music_en()
        #rectext = "What's in the picture"
        #print(rectext)
    except:
        print("Cannot hear command,try again")
        huanxin = 0
        continue

    #rectext = 'If you see yellow, dance; otherwise, lie down'
    if rectext != "":
        print("Q:"+rectext)
        take_photo() # Call the camera to take photos
        time.sleep(0.5)
```

```

        read_image()

        #Perform action choreography
        try:
            agent_plan_output = eval(Dog_agent_plan_Image_en(rectext))
            print('The specific image intelligent agent choreography
actions are as follows', agent_plan_output)

            response = agent_plan_output['response'] # Get what the
robot wants to say to me
            #print('start speech synthesis and play:'+response)
            print("A:"+response)
        except:
            print("try again...")
            huanxin = 0
            continue

        #Activate the voice thread so that it can be used while moving
        tts_thread = threading.Thread(target=Speak_Vioce)
        tts_thread.daemon = True
        tts_thread.start()

        for each in agent_plan_output['function']: # Run each function
of intelligent agent planning and orchestration
            try: # There is an abnormality and it continues to run
downwards

                eval(each)
            except:
                continue

            time.sleep(0.5)

        else :
            time.sleep(0.5)
            huanxin = 0

        if rectext == 0:
            break

```

1. Program flow: detect wake-up words->listen to expression semantics->combine with large model to understand->feedback answers
2. In the path of "/home/pi/DOGSZILLA/Samples/4_Big_Modle/ros_dog_agent/", the directory structure description is as follows. Only the files related to this case are listed below

```

├─ AIagent_go_image.py #Main program entry Chinese version
├─ AIMImage_en.py #Main program entry English version
├─ DAgent_Image_en.py #Intelligent agent English version
├─ dog_agent_Image.py #Intelligent agent Chinese version
├─ dog_API_en.py #Online large model interface English version
├─ dog_base_control.py #Basic action interface
├─ dog_football_api.py #Football sports interface
├─ dog_ImageAPI.py #Online large model interface Chinese version 1
├─ dog_qa_api.py #Interface for question classes
├─ dog_lidar_action.py #Interface for radar navigation
├─ dog_record.py #Interface for recording
├─ dog_speak_iat_en.pyc #Interface for English speech recognition
├─ dog_speak_iat.py #Interface for Chinese speech recognition

```



```

└─ dog_tongyiAPI.py #Interface for Tongyi Qianwen model
└─ dog_Track_food_api.py #Interface for tracking objects
└─ dog_tts_en.pyc #Interface for speech synthesis English version
└─ dog_tts.py #Interface for speech synthesis Chinese version

```

3. How to add more commands and actions

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

```

19  def play_sound_surprise():
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_forward (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

```



If you want to map the navigation point to a place name, such as the tea room is at point A, then write it as shown in the figure below

```

14  Scream (Surprise Scream):play_sound_surprised()
15  Scream (Angry Scream):play_sound_anger()
16  Introduce yourself:play_myself_en()
17  Interface for inquiring about weather and dates: QA.Contect_en(str) #Among them, str is the question asked, such as: What's the weather like today: QA.f
18  Kick away balls/blocks of specified colors, including red, yellow, blue, and green. For example, kick away balls/blocks of sky color:play_football_color
19  Interface for navigating to a certain location: ROS_UDP_Client_Start(str) #Among them, str is the point of position, with a total of 26 selectable point
20  Description interface for navigation related images:Image_Conctect_en(str)#Among them, str is the relevant on-site question asked,For example, describe
21  Rest and wait, such as waiting for two seconds:time.sleep(2)
22  There are also some color related meanings: for example, the sky color is blue, apples are red, bananas are yellow, and leaves are green
23  There are also some navigation related points: for example, the pantry is point A, the restroom is point B, and the conference room is point C.
24  It should be noted that only words related to self introduction and self introduction should be used to call the play_rysel() function. Do not use it
25  It should be noted that when I ask something similar to "what do you see", do not call the interface functions of face tracking, gesture recognition, a
26  It should be noted that when I ask "who are you" with similar meanings, your answer only needs to be: let me think for a moment
27  【Output JSON format】

```

- 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, then as shown in the figure below


```

15  ##前进 forward
16  def Dog_forword(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 of the encapsulated function to the dog_agent_Image.py file (the English version is DAgent_Image_en.py).

```

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_forword(time) #Among them, time represents the number of seconds of the action,Advance 1 second:Dog_forword(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_forword(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()

```

4. How to replace the large model interface

This involves a lot of DIY operations, and it is not recommended for novices to replace the model.

The large model used in this function is **OpenRouter**

- You can first use the Python version program of the platform corresponding to 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" or "dog_tongyiAPI.py" and put it in the directory of "/home/pi/DOGZILLA/Samples/4_Big_Modle/ros_dog_agent/". For example, the added file name is "mychatgpt.py"

```

dog_tongyiAPI.py
1 from openai import OpenAI
2 import os
3 import base64
4 import sys
5
6
7 current_dir = os.path.dirname(os.path.abspath(__file__))
8 parent_dir = os.path.dirname(current_dir)
9 sys.path.append(parent_dir)
10
11 from API_KEY import *
12
13
14
15 # base 64 编码格式
16 def encode_image(image_path):
17     with open(image_path, "rb") as image_file:
18         return base64.b64encode(image_file.read()).decode("utf-8")
19
20 def QwenVL_api_picture(PROMPT='机械狗管家智能体'):
21     base64_image = encode_image("./dog_agent/rec.jpg")
22     client = OpenAI(
23         # 若没有配置环境变量, 请用百炼API Key将下行替换为: api_key="sk-xxx"
24         api_key= TONYI_key,
25         base_url="https://dashscope.aliyuncs.com/compatible-mode/v1",
26     )
27     completion = client.chat.completions.create(
28         model="qwen-vl-max-latest", #qwen-vl-max-latest
29         messages=[
30             {
31                 "role": "user",
32                 "content": [
33                     {
34                         "type": "image_url",
35                         "image_url": {"url": f"data:image/jpg;base64,{base64_image}"},
36                     },
37                     {
38                         "type": "text",
39                         "text": PROMPT
40                     },
41                 ],
42             },
43         ],
44     )
45     #print(completion.model_dump_json())
46     #print('大模型调用成功!')
47     result = eval(completion.choices[0].message.content)
48     #print(result)
49
50     return result

```

```

dog_ImageAPI.py
145
146 def image_main(appid, api_key, api_secret, imageunderstanding_url, question):
147
148     wsParam = Ws_Param(appid, api_key, api_secret, imageunderstanding_url)
149     websocket.enableTrace(False)
150     wsUrl = wsParam.create_url()
151     ws = websocket.WebSocketApp(wsUrl, on_message=on_message, on_error=on_error, on_close=on_close, on_open=on_open)
152     ws.appid = appid
153     #ws.imagedata = imagedata
154     ws.question = question
155     ws.run_forever(sslopt={"cert_reqs": ssl.CERT_NONE})
156
157
158 def getText(role, content):
159     jsoncon = {}
160     jsoncon["role"] = role
161     jsoncon["content"] = content
162     text.append(jsoncon)
163     return text
164
165
166 def getlength(text):
167     length = 0
168     for content in text:
169         temp = content["content"]
170         leng = len(temp)
171         length += leng
172     return length
173
174
175 def checklen(text):
176     #print("text-content-tokens:", getlength(text[1:]))
177     while (getlength(text[1:]) > 8000):
178         del text[1]
179     return text
180
181
182
183
184
185 def Dog_Image(imagetext):
186     global answer, text
187     imagedata = open("./dog_agent/rec.jpg", 'rb').read()
188     imageunderstanding_url = "wss://spark-api.cn-huabei-1.xf-yun.com/v2.1/image"#云端环境的服务地址
189     text = [{"role": "user", "content": str(base64.b64encode(imagedata), 'utf-8'), "content_type": "image"}]
190
191     question = checklen(getText("user", imagetext))
192     image_main(appid, api_key, api_secret, imageunderstanding_url, question)
193     getText("assistant", answer)
194     #print(answer)
195     return answer
196

```

```

31     }
32 }
33 }
34
35 result = completion.choices[0].message.content
36 #print(result)
37 return result
38
39
40
41 def encode_image(image_path):
42     with open(image_path, "rb") as image_file:
43         return base64.b64encode(image_file.read()).decode("utf-8")
44
45
46
47 def dogGPT_Image_en(inputtext):
48     client = OpenAI(
49         base_url="https://openrouter.ai/api/v1",
50         api_key=openAI_KEY,
51     )
52
53     image_path = "./dog_agent/rec.jpg"
54     base64_image = encode_image(image_path)
55
56     completion = client.chat.completions.create(
57
58         model="qwen/qwen2.5-vl-32b-instruct:free",
59         #model="meta-llama/llama-3.2-11b-vision-instruct:free",
60         messages=[
61             {
62                 "role": "user",
63                 "content": [
64                     {
65                         "type": "text",
66                         "text": inputtext
67                     },
68                     {
69                         "type": "image_url",
70                         "image_url": {"url": f"data:image/jpg;base64,{base64_image}"},
71                     }
72                 ]
73             }
74         ]
75     )
76
77     result = completion.choices[0].message.content
78     #print(result)
79     return result
80
81
82
83

```

- Open **Ros_dog_Main_en.py**

```

dog_agent_Image.py
1 from dog_ImageAPI import *
2 from dog_tongyiAPI import *
3

```

- Take this tutorial as an example: Then add from dog_UltraAPI import * from mychatgpt import * in the header

```

1 from dog_ImageAPI import *
2 from dog_tongyiAPI import *
3 from mychatgpt import *
4

```

- Then find this place and replace it with your own encapsulated API function interface.
- 5. If you want to change the threshold for recording start and the duration of recording, you can change this file. Enter in the terminal

```
nano /home/pi/DOGZILLA/Samples/4_Big_Modle/ros_dog_agent/dog_record.py
```

Change the recording part of this file as shown in the figure below

```

quitmark = 0
automark = True
def start_recording(timel = 3,save_file=SAVE_FILE):
    global automark,quitmark
    start_threshold = 3000 #30000 Start threshold
    end_threshold = 1500 #2000 Stop threshold
    endlast = 15 # The total number of times detected is lower than the stop threshold, and the recording is automatically stopped
    max_record_time = 10 #The maximum recording duration

    CHUNK = 1024
    FORMAT = pyaudio.paInt16
    CHANNELS = 1
    RATE = 16000
    WAVE_OUTPUT_FILENAME = save_file

```

Parameter meaning:

- start_threshold = 3000 #Start recording when a sound higher than this value is detected. This value changes according to the environment

- `end_threshold = 1500` #Sound lower than this value is detected. This value changes according to the environment
- `endlast = 15` #Stop recording when the number of sounds lower than `end_threshold` is detected. Here it is 15 times
- `max_record_time = 5` #Record audio duration, here is 5

Note: start_threshold must be greater than end_threshold (start_threshold > end_threshold)

Generally, the ideal value of `end_threshold` is half of `start_threshold`, which can be adjusted according to your own environment.

6. If you feel that the recorded audio cannot be recognized by the online large model because the sound is too small, you can adjust the value here to amplify the recorded audio.
Terminal input

```
nano /home/pi/DOGZILLA/Samples/4_Big_Modle/ros_dog_agent/dog_record.py
```

```
269 wf.close()
270 print(f"The recording has been saved as: {WAVE_OUTPUT_FILENAME}")
271
272 amplify_audio_librosa("recorded_audio.wav", "recorded_audio.wav" gain_factor=5.0) #放大它 Enlarge it
273
```

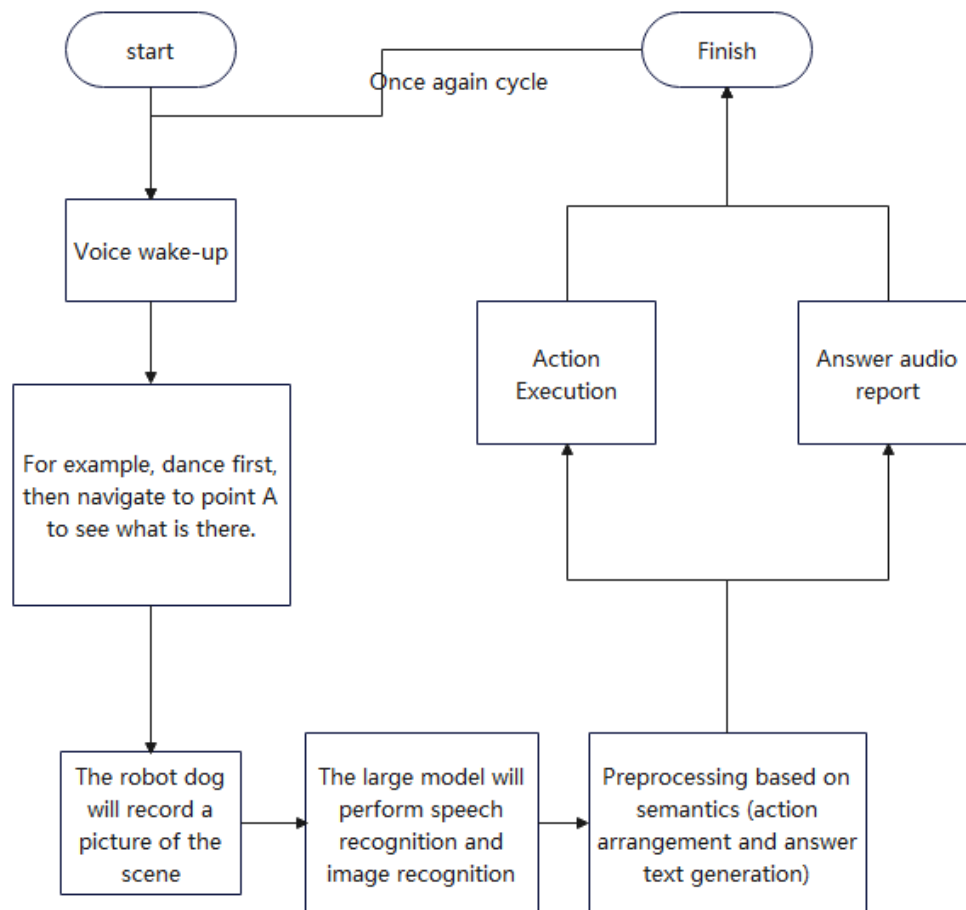
The tutorial here is amplified 5 times, and here you can make an adjustment according to the distance of the sound source.

Note: If the distance is too far to record audio at all, adjusting the parameters here will be meaningless.

It is recommended that the distance from the sound source of the audio recording should not exceed 1.5m.

4. Experimental summary

Through the above description, the flowchart of this case is as follows:



If you don't know what this case means, here are some reference examples

Examples:

1. A series of action instructions, such as: dance, turn in a circle, push-ups, move forward for 3s, wave your hand and start.
 2. Some sentences of action instructions, such as: first show your body, then turn in a circle and lie down, and finally say hello to me.
 3. Action instructions + some questions, such as: dance, help me kick the sky-colored ball on the desktop, and finally tell me today's date.
 4. Navigate to the corresponding point + perform actions, such as: help me go to point 1 to see what there is, then go to point 2 to dance, and finally go to point 3 to exercise.
 5. You can say some speculative sentences, such as: I don't want to see the yellow ball on the desktop. / I am very sad now.
 6. You can consult some related to the live film, such as: If you see the red ball, dance, otherwise lie down and rest for a while.
- You can use your imagination and interact boldly.

Notes

1. If this error occurs when the program is started, you can press "ctrl+C" to end the program and then restart it.

```

peter@peter:~/mystudy/Free_QX$ python3 chatgpt_mdapi.py
serial /dev/myspeech open
Network check failed: HTTPConnectionPool(host='www.baidu.com', port=80): Max retries exceeded with url: / (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fff84058610>: Failed to establish a new connection: [Errno -3] Temporary failure in name resolution'))
检测网络没连上，请重启网络
  
```

