Free Q&A

Function Introduction

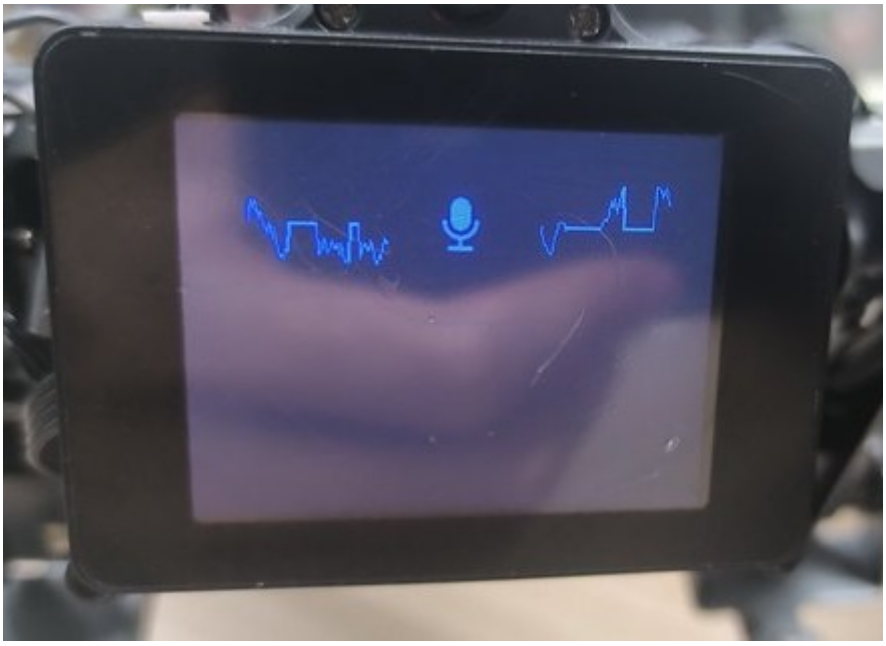
This case is about starting a large program. You need to complete the configuration of the API-KEY related to the large model before you can use it normally. This function is a talkback function for communicating with users.

This function requires an Internet connection to work normally.

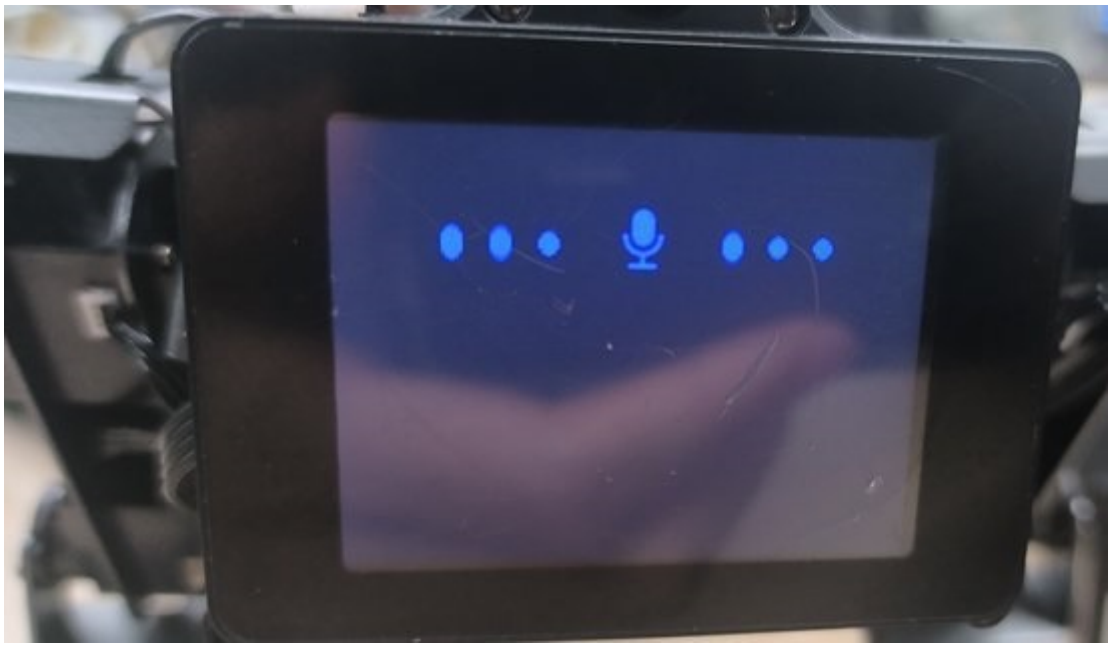
Functional Experience

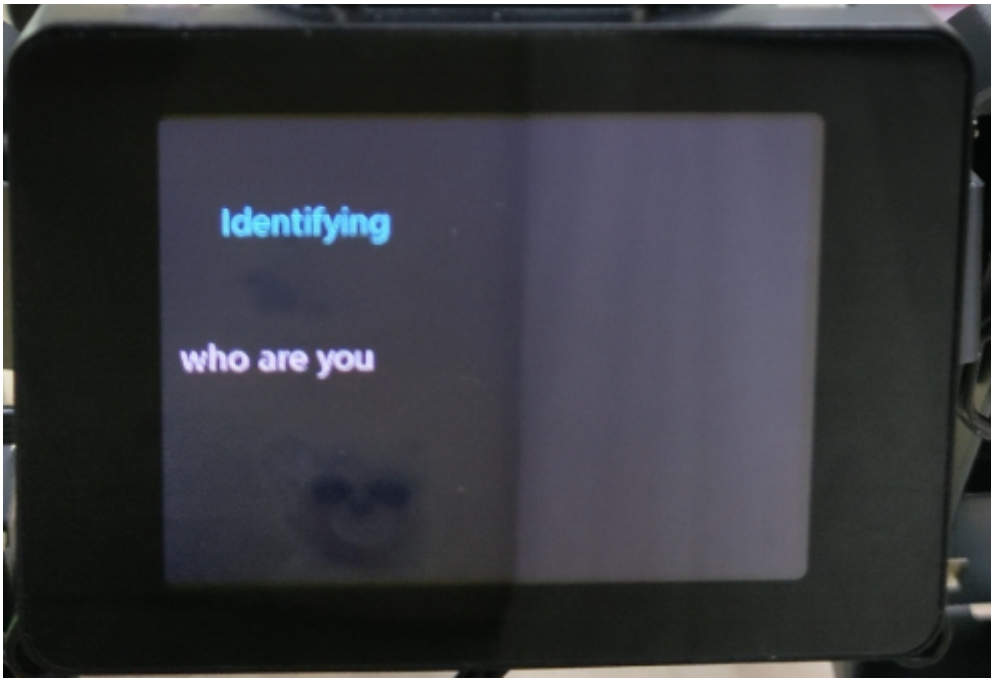
1. Turn on the robot dog first, enter the sample mode by pressing the button on the upper right of the "dog head", and then select the free question and answer function.

2. After entering the free question and answer function, wake it up with voice, "lulu".

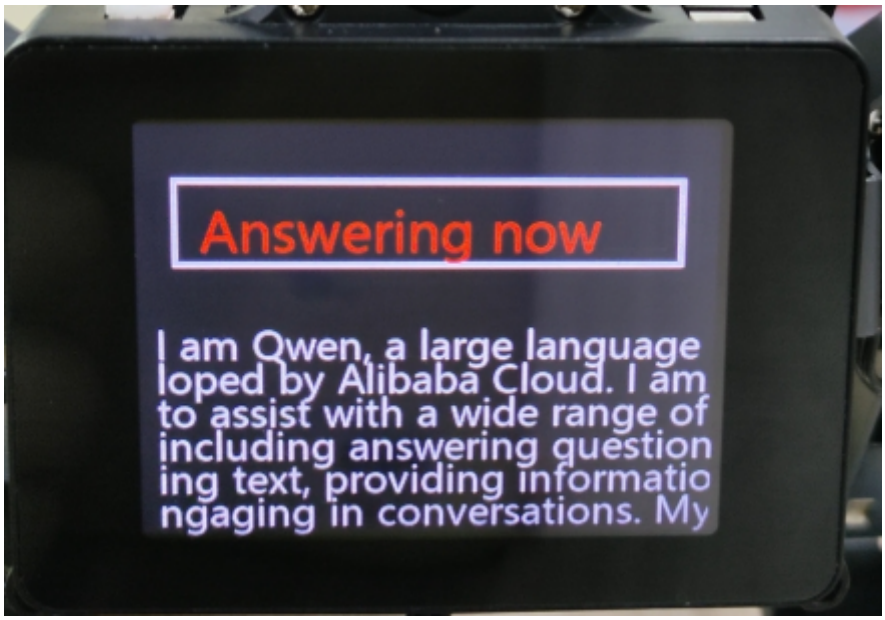


3. When you hear a ding, you can say the question you want to ask.





4. The robot dog will give an answer based on the question asked.



Program source code

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

```
cd /home/pi/RaspberryPi-CM4-main/demos/Free_QA/
tree
```

3. Directory structure description
   ├── answer.mp3 #Synthesized audio
   ├── audio.py #Recording file
   ├── auto_platform.py #Dependencies required for recording
   ├── chatgpt_main.py #Main program
   ├── language_recognize.py #Speech recognition
   ├── libnyumaya.py #Speech wake-up
   ├── xinghou_tts.py #Synthesized audio
   └── xinghou_UltraAPI.py #Large language model interface

How to replace the large model interface

- You can start with the Python version of the platform's corresponding interface program and fill in the necessary information according to the platform's interface and instructions.
- Then encapsulate the executable file into a function. You can refer to the method of "xinghou_UltraAPI.py" and put it in the directory of point 2. For example, the added file name is "mychatgpt.py"
- Open the **chatgpt_main.py** file and replace from xinghou_UltraAPI import * with the newly added from mychatgpt import * at the top.

- Then find this place and replace it with your own encapsulated API function interface.

```
                    scale=font2,
                    mono_space=False,
                )
                display.ShowImage(splash)

                lines = len(display_text.split("\n"))
                tick = 0.3
                if lines > 6:
                    scroll_text_on_lcd(display_text, 10, 111, 6, tick)

                #big model
                sctext = "正在识别" if la == 'cn' else "Identifying"
                lcd_draw_string(draw, 30, 40, sctext, color=(0, 255, 255), scale=font2, mono_space=False)
                display.ShowImage(splash)

                re = Ultra_gpt(content)      You can change the desired large model interface by yourself
                re_e = line_break(re)
                print(re_e)
                re_text = re_e

                lcd_rect(0,40,320,290,splash_theme_color,-1)
                draw.rectangle((20,30,300,80), splash_theme_color, 'white',width=3)

                ananan = "正在回答" if la == 'cn' else "Answering now"
                lcd_draw_string(draw,35,40, ananan, color=(255,0,0), scale=font3, mono_space=False)


                lcd_draw_string(
                draw,
                10,
                111,
                re_text,
                color=(255, 255, 255),
                scale=font2,
                mono_space=False,
                )
                display.ShowImage(splash)

                relines = len(re_text.split("\n"))
                tick = 0.3
                if relines > 6:
                    scroll_text_on_lcd(re_text, 10, 111, 6, tick)
                try:
                    Xinghou_speakTts(re)#播放音频  play
                except:
                    pass
            if content == 0:
                break

        time.sleep(0.1)
else:
```

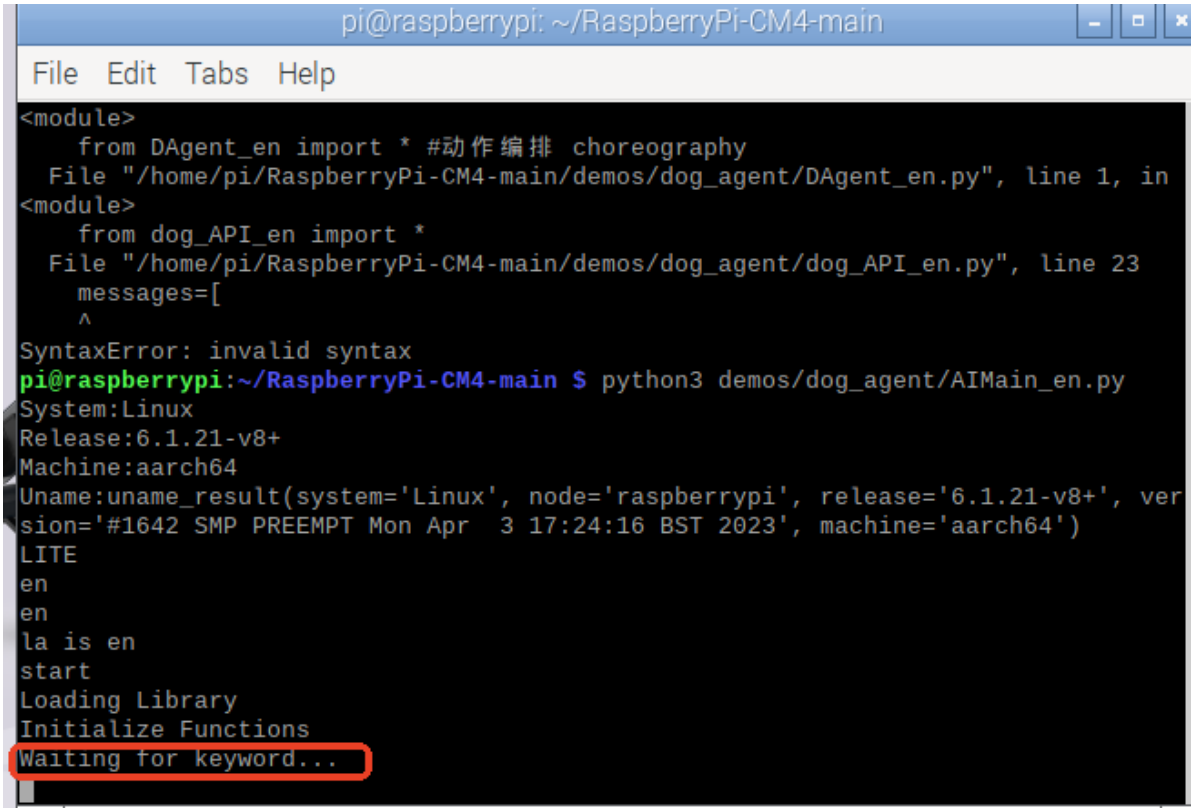
1. Then restart the car and enter this function again, you can run the model platform you replaced. If it cannot run, it means there is an error, and you need to check the syntax and logic of the newly added file yourself.

How to run this example in terminal

1. End the large program first to prevent screen distortion. For details on how to end the large program, please refer to the tutorial on ending the large program in Chapter 1. This will not be described here.
2. Enter the following command in the terminal

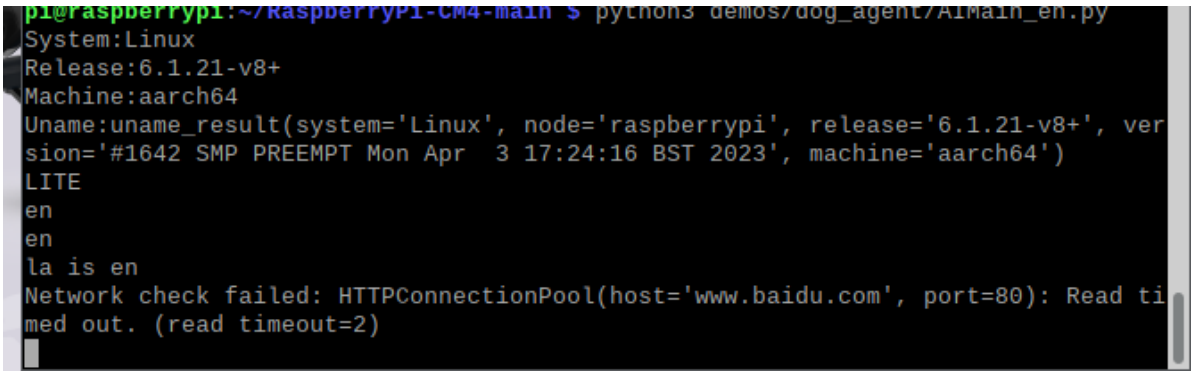
```
cd ~/RaspberryPi-CM4-main
sudo python3 demos/Free_QA/chatgpt_main.py
```

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



Note:

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.



How to change the length of recording time

1. Terminal Input

```
nano ~/RaspberryPi-CM4-main/demos/Free_QA/audio.py
```

2. Find this place and change the place as shown in the figure below, then you can adjust the recording duration according to your own 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.

Functional principle

The specific flow chart is as follows: