Q&A and ChatGPT

Feature Introduction

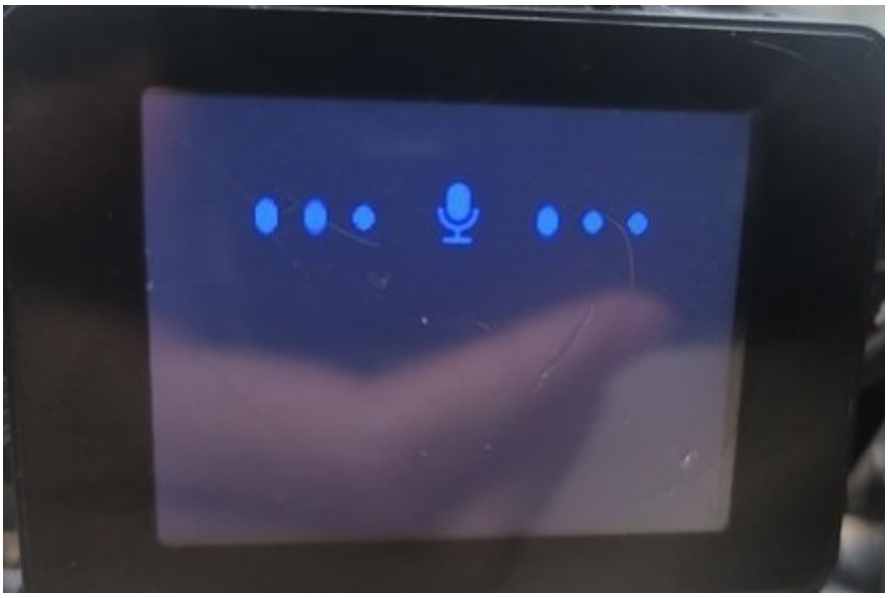
1. This case is a startup program usage that requires configuring the large model's related API-KEY to work properly.
2. This feature enables conversational interaction with users.
3. **This feature requires an internet connection to work properly**

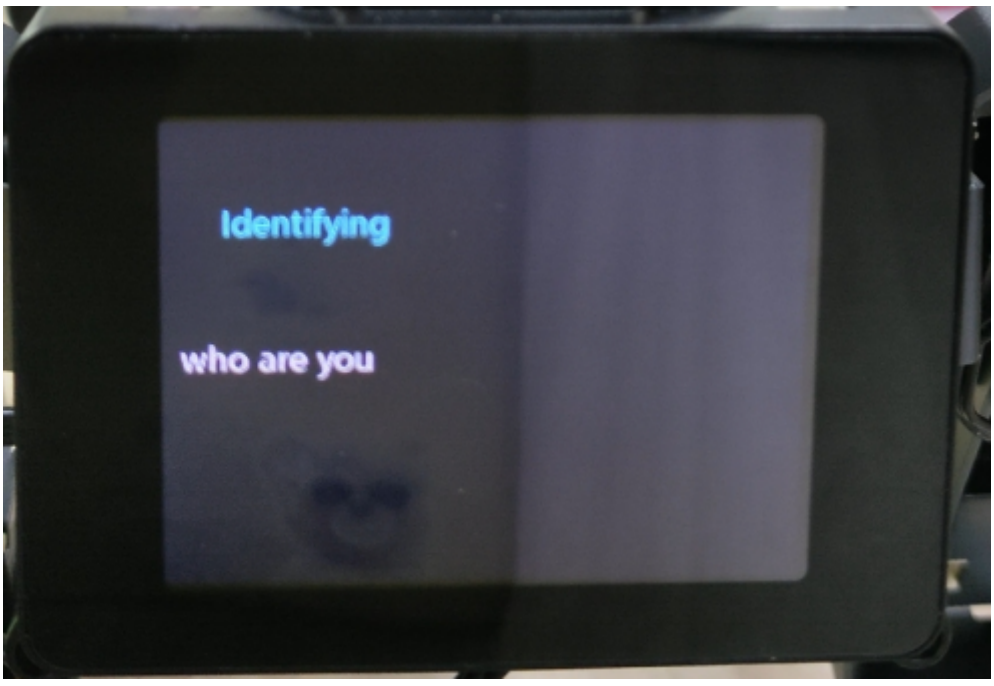
Feature Experience

1. First start up RiderPI, press the button in the upper right corner of the "screen" to enter demo mode, then select the iFlytek Spark feature.

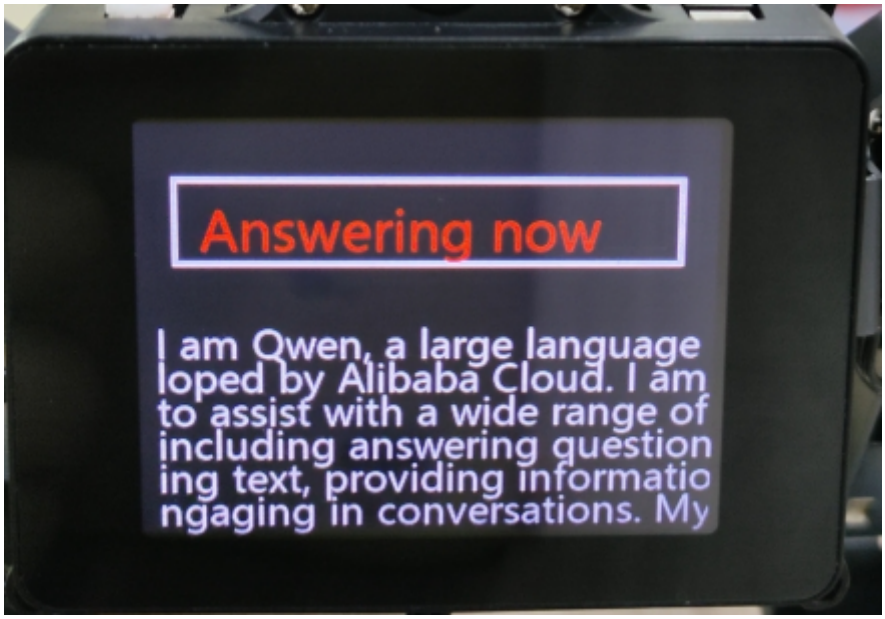
2. After entering the iFlytek Spark feature, first use voice to wake up with "lulu".

3. When you hear a "ding" sound, you can speak your question.





4. RiderPI will respond based on your question.



Program Source Code

1. First, log in to the RiderPI system via VNC
2. Then enter the following in the terminal:

```
# For CM4
cd /home/pi/RaspberryPi-CM4-main/demos/Free_QA/

# For CM5
cd /home/pi/RaspberryPi-CM5/demos/Free_QA/

tree
```

3. Directory Structure Description
   ├── answer.mp3 # Synthesized audio
   ├── audio.py # Recording file
   ├── auto_platform.py # Dependencies needed for recording
   ├── chatgpt_main.py # Main program
   ├── language_recognize.py # Speech recognition
   ├── libnyumaya.py # Voice wake-up
   ├── xinghou_tts.py # Audio synthesis
   └── xinghou_UltraAPI.py # Large language model interface
4. If you want to replace with your own model interface

- First get the Python version program for the corresponding platform's interface, fill in necessary information according to the platform's interface and instructions.
- Then wrap the runnable file into a function, refer to the "xinghou_UltraAPI.py" method and place it in the directory from step 2, for example, adding a file named "mychatgpt.py"
- Open the **chatgpt_main.py** file, replace `from xinghou_UltraAPI import *` at the top with `from mychatgpt import *`

- Then find this location and replace it with your own wrapped 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)
                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
                except:
                    pass
            if content == 0:
                break

        time.sleep(0.1)
else:
```

可以自行更换想要的大模型接口

You can change the interface of the larger model you want.

5. Then restart the device and enter this feature again to run your replaced model platform. If it doesn't work, there's an error and you need to check the syntax and logic of the newly added file yourself.

How to Run This Case from Terminal

1. First end the main program to prevent screen flickering. For instructions on how to end it, please refer to the tutorial in Chapter 1 about ending the main program. This won't be described here.
2. Enter the following command in the terminal:

```
# For CM4
cd /home/pi/RaspberryPi-CM4-main
python3 demos/Free_QA/chatgpt_main.py

# For CM5
cd /home/pi/RaspberryPi-CM5
python3 demos/Free_QA/chatgpt_main.py
```

How to Change Recording Duration

1. Enter in terminal:

```
# For CM4
nano ~/RaspberryPi-CM4-main/demos/Free_QA/audio.py
# For CM5
nano ~/RaspberryPi-CM5/demos/Free_QA/audio.py
```

2. Find this location and change the area indicated in the image below to adjust recording duration according to your environment.

```
229 def start_recording(timel = 3,save_file=SAVE_FILE):
230     global automark,quitmark
231     start_threshold = 120000
232     end_threshold = 100000 #说话停止的声音阈值，根据自己的环境调整
233     endlast = 15
234     max_record_time = 5 #录音最大时间，此处为5s
```

Adjust the sound threshold for stopping speaking according to your environment.

Maximum recording time: 5 seconds.

Note: start_threshold > end_threshold, these two values need to be adjusted according to your own environment.

Feature Principle

The specific flowchart is as follows: