

# Free Conversation

---

## Free Conversation

[Experiment Objective](#)

[Experimental Steps](#)

[Experimental Results](#)

[Main Source Code Analysis](#)

[Overall Flowchart](#)

## Experiment Objective

---

Understand and master the basic conversational features of the AI model, interact with the smart car through verbal conversation, and enable the smart car to solve the desired problem.

## Experimental Steps

---

1. Observe the IP address of the OLED screen and log in to the remote desktop via VNC.
2. Follow the prerequisite configuration instructions to complete the Tongyi Qianwen key for the Chinese version and the OpenRouter key for the English version.
3. Open a new terminal and run the following command:

```
cd /home/pi/project_demo/09.AI_Big_Model/
```

```
#Startup command for the Chinese version  
python3 Free_QA/chatgpt_main.py
```

```
#Startup command for the English version  
python3 Free_QA/chatgpt_main_en.py
```

4. Wake up the car using the wake-up phrase "Hi, Yahboom" (for international users).
5. After successfully waking up, the car will respond with a honking sound. After waiting for about half a second, you can then state the problem you want the car to help you solve.
6. After the robot recognizes your voice, wait a few seconds for relevant information to be displayed on the terminal interface and speaker. You can also terminate the current conversation by using the wake-up word.
7. This concludes the conversation. To start a new one, repeat steps 4-6.

## Experimental Results

---

1. Waiting for wake-up

```
pi@yahboom:~/project_demo/09.AI_Big_Model $ python3 Free_QA/chatgpt_main.py  
serial /dev/myspeech open  
Waiting for keyword...  
█
```

2. Upon successful wake-up, a green-boxed message will appear, allowing you to ask your question.

```

Keyword detected: 17.Jul 2025 19:04:34
Playing WAVE './didi.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib confmisc.c:1369:(snd_func_refer) Unable to find definition 'cards.2.pcm.hdm.0:
,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:5180:(_snd_config_evaluate) function snd_func_refer returned error: No s
irectory
ALSA lib conf.c:5703:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM hdm.0
ALSA lib confmisc.c:1369:(snd_func_refer) Unable to find definition 'cards.2.pcm.hdm.0:
,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:5180:(_snd_config_evaluate) function snd_func_refer returned error: No s
irectory
ALSA lib conf.c:5703:(snd_config_expand) Evaluate error: No such file or directory
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM hdm.0
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2666:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
Current volume: 31999.0, boot threshold: 3000, End threshold: 1500
start recording
3000 31999.0
Current volume: 36517.0, boot threshold: 3000, End threshold: 1500
3000 36517.0

```

3. Wait for a while, and the terminal will print out the question and answer.

```

Q: How high is mount everest?
A:Mount Everest is the highest mountain in the world above sea level. Its peak e
levation is approximately **8,848.86 meters (29,031.7 feet)** above sea level. T
his measurement was confirmed by a survey conducted by the Chinese and Nepalese
governments in 2020.

```

## Main Source Code Analysis

```

def Speak_Vioce():
    global response
    if TTS_IAT_Tongyi:
        tonyi_tts(response)
    else:
        xinghou_speak_tts(response)

def main():
    while True:
        if detect_keyword():
            os.system("pkill mplayer")
            time.sleep(.2)

            start_recording()
            time.sleep(1)

            if TTS_IAT_Tongyi:
                content = rec_wav_music_Tongyi()
            else:
                content = rec_wav_music()

            if content != "":
                print("Q:"+content)
                #re = ultra_gpt(content) # chinese

```

```

re =tongyi_QA_Model(content)

print("A:"+re)
try:
    response = re
    tts_thread = threading.Thread(target=Speak_Vioce)
    tts_thread.daemon = True
    tts_thread.start()

except:
    pass
if content == 0:
    break
time.sleep(0.1)

```

**detect\_keyword:** Wake-up function handler

**start\_recording:** Recording function handler

Chinese version-specific options:

**rec\_wav\_music\_Tongyi:** Tongyi Qianwen voice recognition

**rec\_wav\_music:** iFlytek Spark voice recognition

You can choose either voice recognition mode. You can enable or disable it in the **API\_KEY.py** file.

When TTS\_IAT\_Tongyi=True, either Tongyi Qianwen voice recognition or iFlytek Spark voice recognition is enabled.

**tonyi\_tts:** Tongyi Qianwen speech synthesis

**Xinghou\_speaktts:** iFlytek Xinghuo speech synthesis

You can choose between two speech synthesis options. You can enable or disable them in the **API\_KEY.py** file. When TTS\_IAT\_Tongyi=True, either Tongyi Qianwen speech synthesis or iFlytek Xinghuo speech synthesis is enabled.

English Version

Speech recognition and synthesis use the iFlytek Spark API by default. You don't need to specify the iFlytek Spark API in API\_KEY.py; simply fill in the **openAI\_KEY** key.

## Modifying the recording duration, start threshold, and end threshold

1. In the terminal, enter:

```

cd /home/pi/project_demo/09.AI_Big_Model/
nano audio.py

```

2. Find the source code shown below.

```

189 quitmark = 0
190 automark = True
191 def start_recording(timel = 3,save_file=SAVE_FILE):
192     global automark,quitmark
193     start_threshold = 3000 #30000
194     end_threshold = 1500 #20000
195     endlast = 15
196     max_record_time = 5

```

- **start\_threshold:** The threshold for starting recording when sound is detected (reduced to 5000 in quiet environments and increased to 150000+ in noisy environments).
- **end\_threshold:** The threshold for stopping recording when sound is detected. A recommended value is 30-50% of start\_threshold.
- **endlast:** The number of times to stop recording. Here, 15 is used. For example, Recording will automatically terminate if 15 consecutive sound values meet the stop threshold.

- max\_record\_time: Recording duration, set to 5 here.  
Note: start\_threshold > end\_threshold. This is a required rule, and its value can be determined based on the environment.

### 3. Directory Structure of the Main Files for this Experiment

```
|— chatgpt_main_en.py #Main Entry Point for the English Version
|— chatgpt_main.py #Main Entry Point for the Chinese Version
|— tongyi_api.py #Tongyi Qianwen Language Model API (If you want to change to
another model, you can modify this file)
|— tongyi_speak_iat.py #Tongyi Qianwen Speech Recognition
|— tongyi_tts.py #Tongyi Qianwen Speech Synthesis
|— xinghou_speak_iat.py #iFlytek Spark Speech Recognition
|— xinghou_tts.py #iFlytek Spark Speech Synthesis
```

## Overall Flowchart

