

Microsoft Azure IOT User Manual(Based on Orange PI)

I. Introduction

In order to help developers to use Microsoft Azure IOT on Orange Pis quickly and easily, we provide this manual to illustrate how to build the development environment quickly. Before setting up the development environment, we need to prepare an Orange Pi development board, firmware package or transplant package of OrangePi & Microsoft Azure IOT. The firmware package is released with the Microsoft IoT Starter Kit and the porting package can be obtained through [Github](https://github.com).

II. Quick Start

Through applying for Microsoft IOT Start Kit, which is released by Microsoft, you could get an Orange Pi, SD card with firmware wrote, camera and 24 pin FPC and one power cable. As shown below:

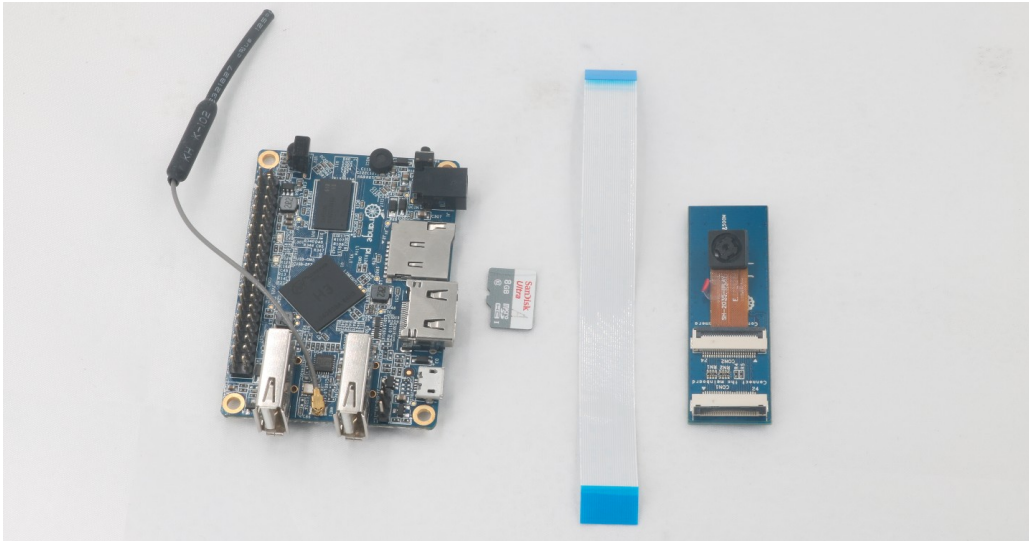


Figure 1. OrangePi development component

1. First, inserting the SD card into SD card slot on Orange PI development board, then connecting the camera and 24pin FPC to the board. Connection mode are as follows, please pay attention to the connection mode of the FPC, blue side is back to the interface.

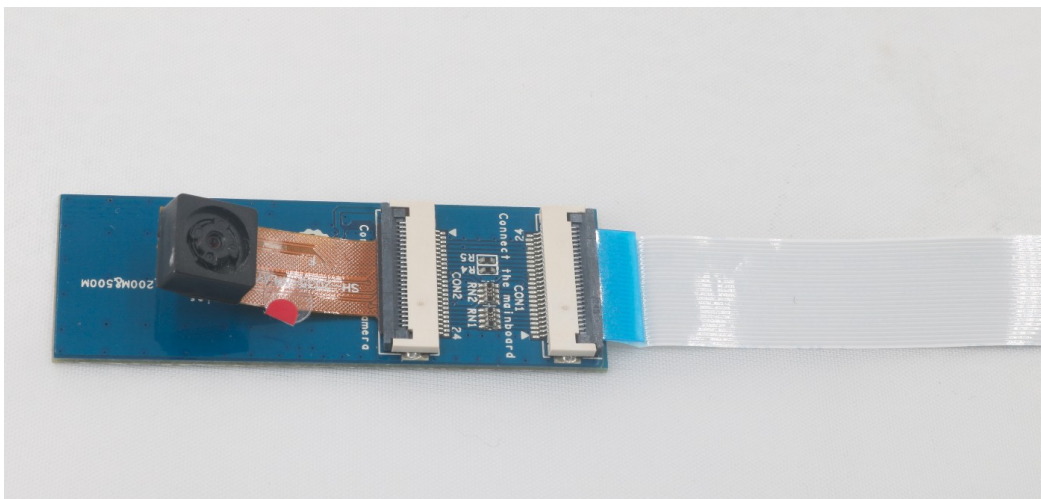


Figure 2.1 Connection mode of camera module

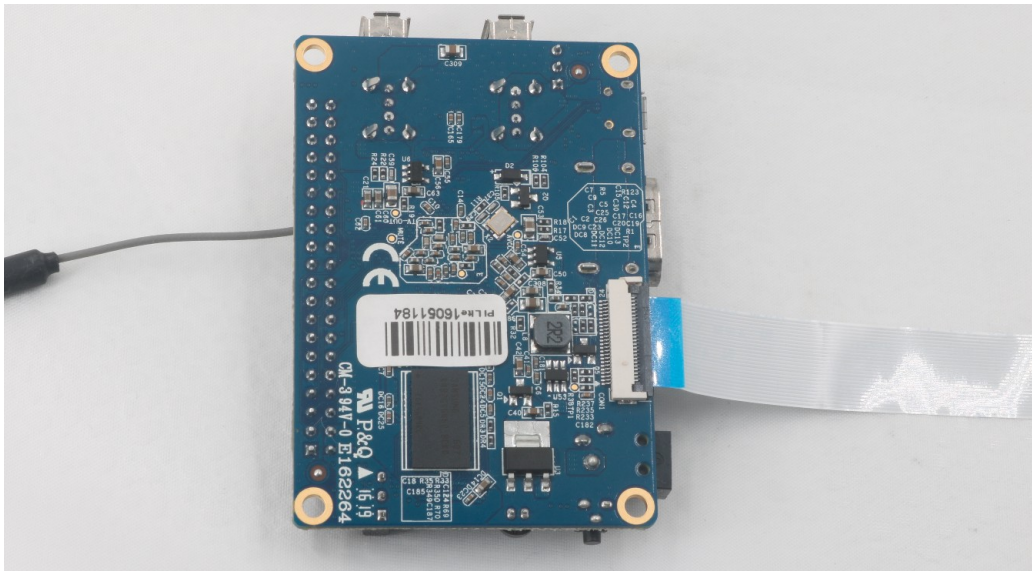


Figure 2.2 Connection mode of OrangePi board

2. Prepare a WIFI AP, set the name and password of WIFI AP into “iotstarterkit” . Restart WiFi AP and then plug OrangePi board into power cable to boot.
3. Connecting to WIFI AP in your laptop or computer and use IP searching tools to search the IP address of Orange pi.
4. After obtaining the IP address of Orange PI, you could open Host browser and enter IP address of Orange PI. Then you will entry to the Home page.

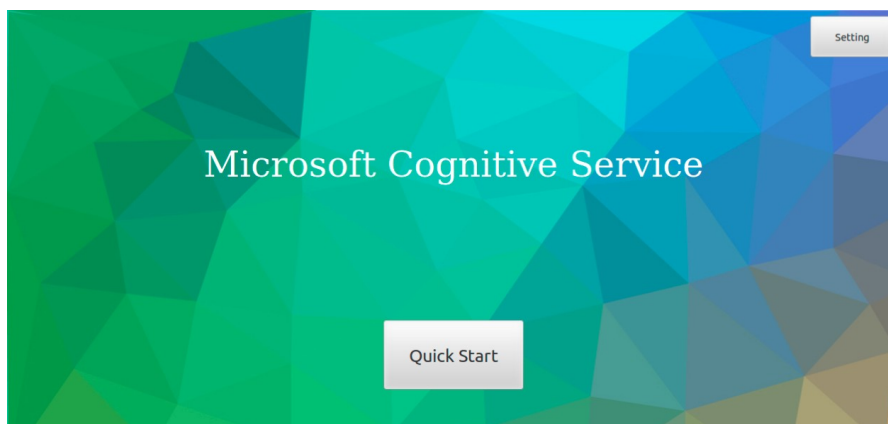


Figure 2.3 Home Page

5. Please click the *Setting* on the upper right corner to initialize the settings while it is your first time to use.

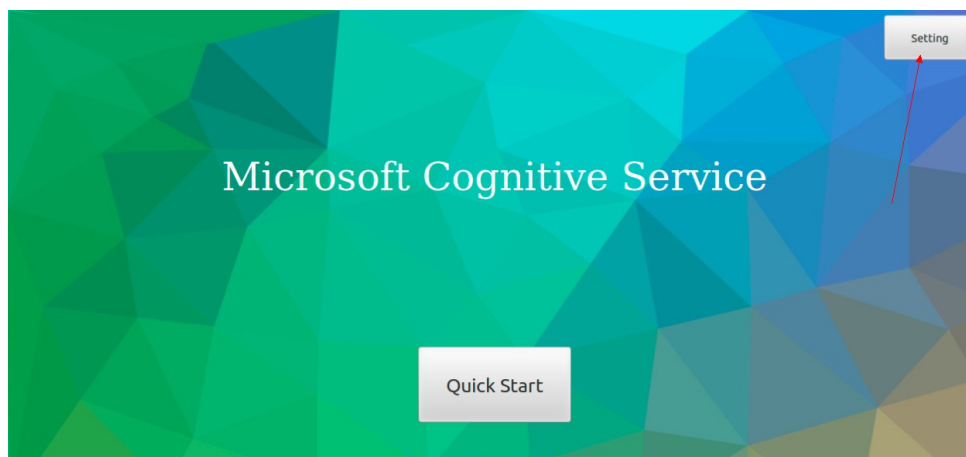


Figure 2.4 Setting

6. According to the input box prompts, enter the corresponding Azure IOT information, Microsoft Cognitive Service information and click “confirm” button.

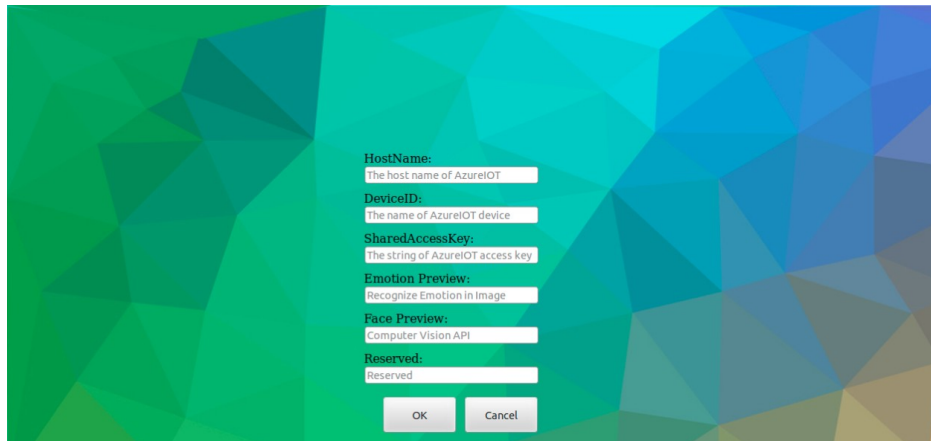
A screenshot of a settings form with a green and blue geometric background. The form contains several input fields with labels and placeholder text: 'HostName:' with 'The host name of AzureIoT', 'DeviceID:' with 'The name of AzureIoT device', 'SharedAccessKey:' with 'The string of AzureIoT access key', 'Emotion Preview:' with 'Recognize Emotion in Image', 'Face Preview:' with 'Computer Vision API', and 'Reserved:' with 'Reserved'. At the bottom are 'OK' and 'Cancel' buttons.

Figure 2.5 Initialization information input

7. Click "Start Quick" button then you will entry to the functional page. You could choose services of voice control and image recognition.

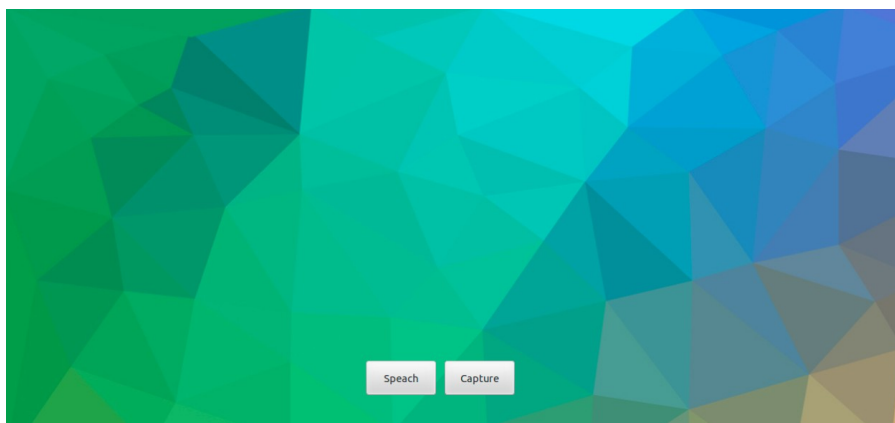


Figure 2.6 Function menu

III. Usage of Image Recognition

After entering image recognition services, you can preview pictures, on the real time video stream captured by the camera, and carry on image recognition services. Specific steps are as follows:

1. Click the "capture" into the real time video browsing page, you will be able to gain the real-time video stream from the OrangePi. Click the "capture" button to captured picture.

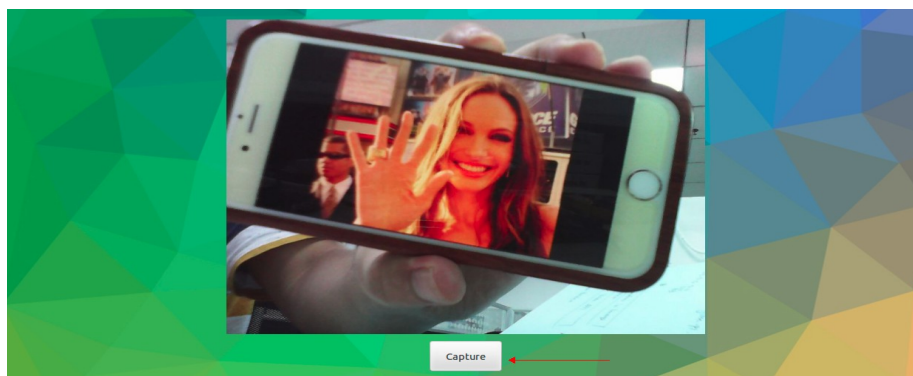


Figure 3.1 mjpg video stream

2. The system could be previewed the captured pictures, and you can re-capture or confirm to

upload according to the captured effect. After clicking the "Upload" button, the captured image will be uploaded to Microsoft Cognitive Service for process.

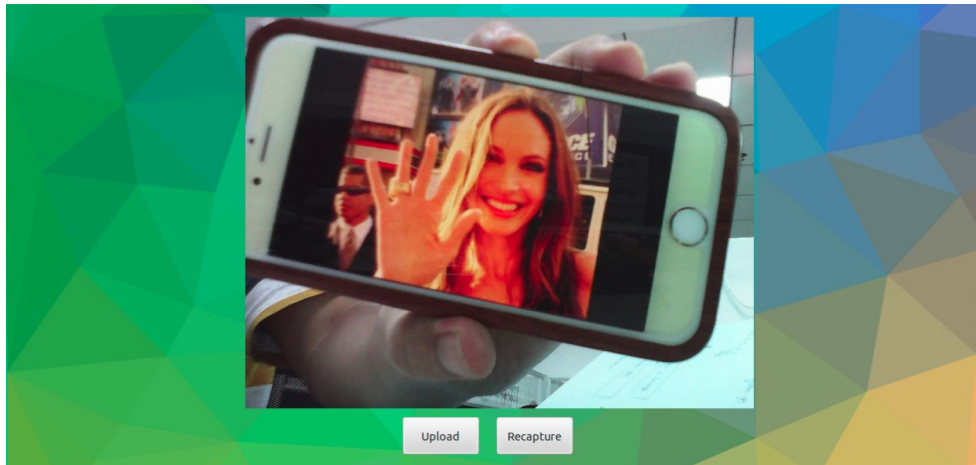


Figure 3.2 Capture preview image

3. The after-processed image will be displayed with original image together.

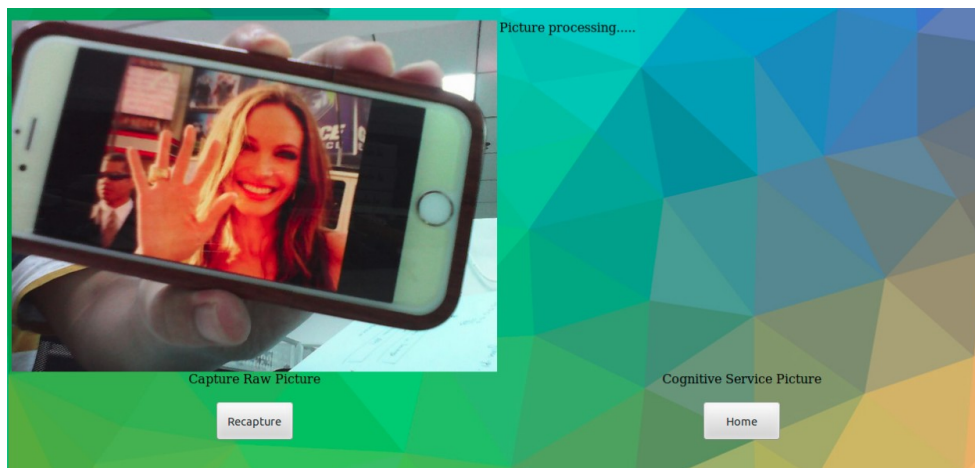


Figure 3.3 Comparison between original image and after-process image

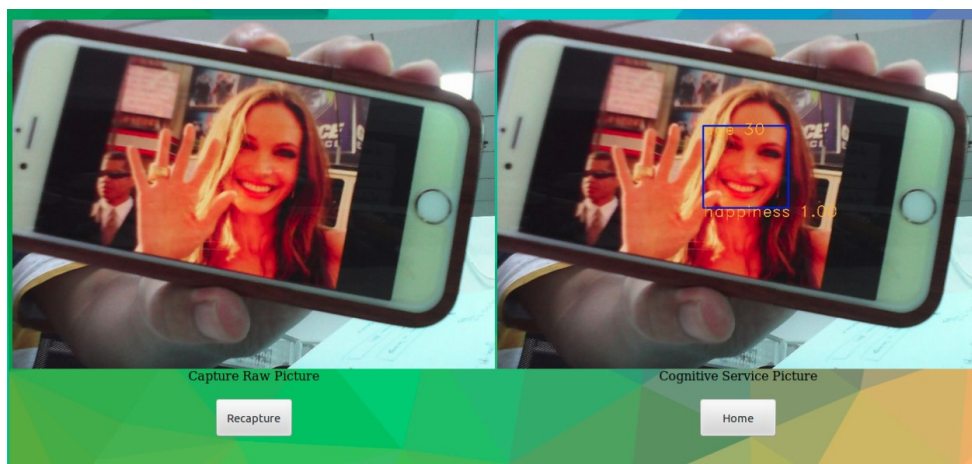


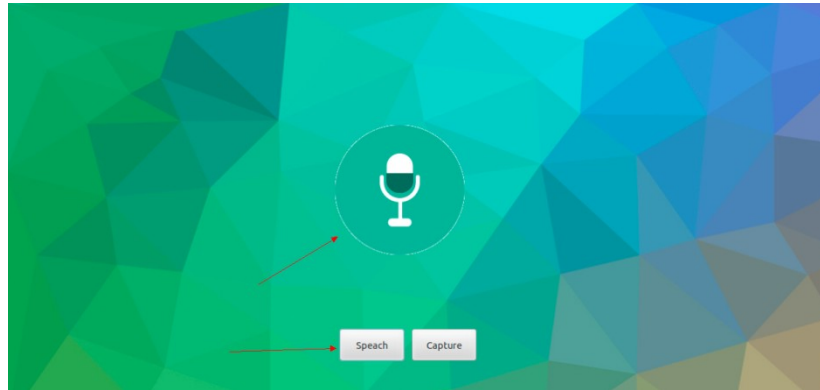
Figure 3.4 Upload image to Microsoft Cognitive Service

IV. Speech Recognition Module

The speech recognition module provides the function of voice control IOT device. You can command OrangePi Board "open" and "close" to control the turn-on or turn-off the LED. Specific steps are as follows:

1. After clicking "Quick Start" button on the Home page into functional page, click "speech" button and make a sound to the Orange PI. "open" could turn the LED on, "close" could turn

n off the LED.



2.
3. Figure 4.1

recognition

Speech

After your command, LED of IOT will turn-on or turn-off accordingly.

V. Source Code Compilation

Before transplanting Microsoft Azure IOT on a new OrangePi, you need to install the libraries for Orange PI runing. Please refer to the following steps for transplantation:

1. Inserting power cable to the OrangePi and logging in OrangePi, entering the following command in the terminal:

```
sudo apt-get install gcc make julius julius-voxforge
```

2. Download source code through [github](#) , source directory are as follows:

Azure_IOTLib: The middle layer between OrangePi and Microsoft Azure IOT, the main realization is uploaded the data to Azure.

Mjpeg-streamer: Realized real time video data stream of mjpeg which based on OrangePi.

OrangePiSpeech2Text: Speech recognition module which based on Julius and Alsa.

PortLib: Transplant related libraries.

SoundLib: Related speech connector of Alsa.

VideoLib: Interface implementation of video acquisition and image cognitive service which based on V4L2.

Websit: Related files of website.

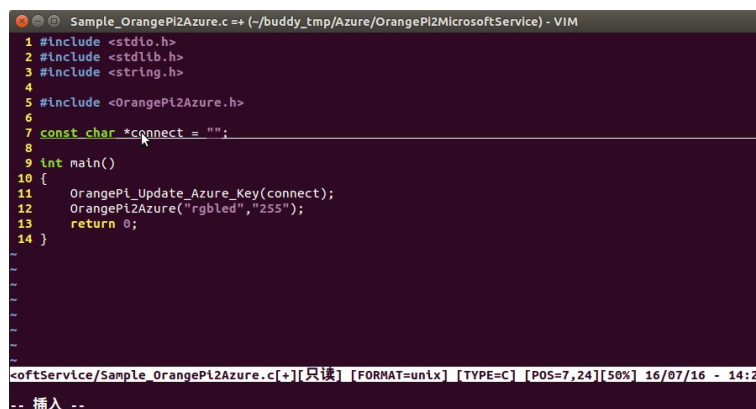
Sample_OrangePi2Azure.c: Azure IOT Demo Code based on OrangePi.

Sample_OrangePi2Sound.c: Voice stream acquisition Demo Code based on OrangePi.

Sample_OrangePi2Video.c: Cognitive Service Demo Code based on OrangePi.

3. Write programs by using Azure IOT which based on OrangePi.

You can compile by referenced to `Sample_OrangePi2Azure.c`, shown as *Figure 5.1 OrangePi & Azure IOT Demo Code* shows.

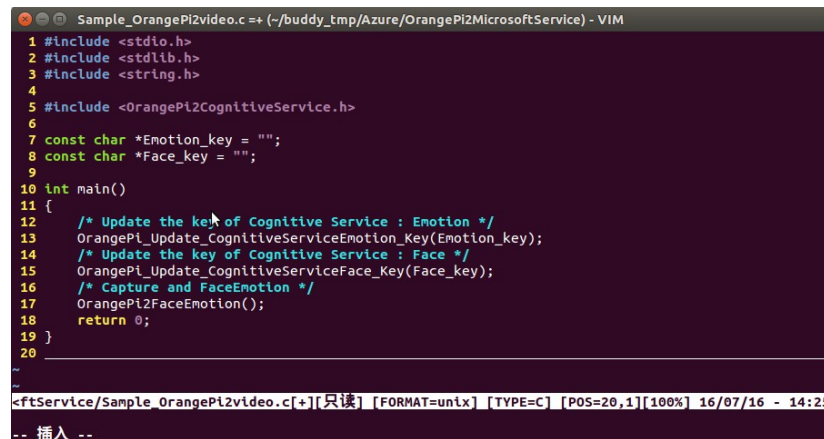


Through the referenced of header file of OrangePi2Azure.h on source code, you could fill Azure IOT

register string to *connect*. Updating local Azure configuration information by calling *OrangePi_Update_Key* function, and unloading custom information to Azure by calling *OrangePi2Azure* function.

4. Writing program by using cognitive service which based on OrangePi.

You can compile by referenced to *Sample_OrangePi2video.c*, shown as *Figure 5.2 OrangePi & Cognitive Service Demo Code*.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #include <OrangePi2CognitiveService.h>
6
7 const char *Emotion_key = "";
8 const char *Face_key = "";
9
10 int main()
11 {
12     /* Update the key of Cognitive Service : Emotion */
13     OrangePi_Update_CognitiveServiceEmotion_Key(Emotion_key);
14     /* Update the key of Cognitive Service : Face */
15     OrangePi_Update_CognitiveServiceFace_Key(Face_key);
16     /* Capture and FaceEmotion */
17     OrangePi2FaceEmotion();
18     return 0;
19 }
20
-- 插入 --
```

Figure 5.2 OrangePi & Cognitive Service Demo Code

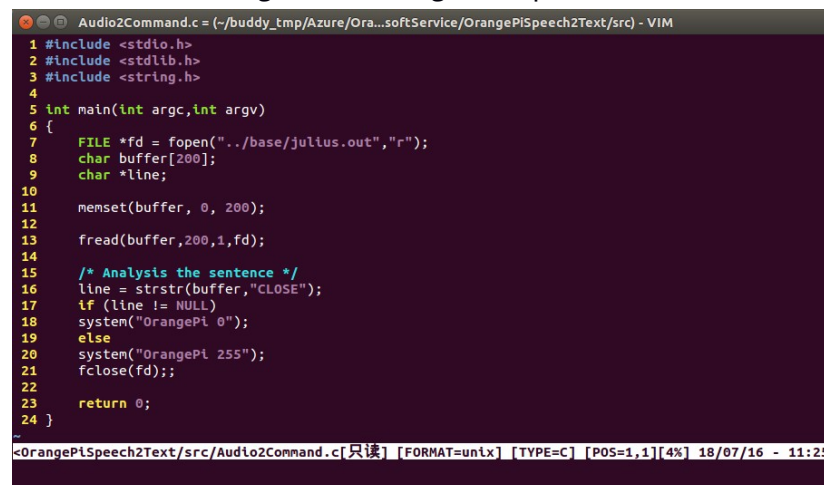
5.

Through the referenced of header file of *OrangePi2 Cognitive Service.h*, you could add Emotion and Face register string to *Emotion_key* and *Face_key* at Cognitive Service. Updating local Emotion Preview configuration information by calling *OrangePi_Update_CognitiveServiceEmotion_Key* function.

Updating Face Preview configuration information by calling *OrangePi_Update_CognitiveServiceFace_Key* function. Uploading image for processing and get the after-process image by calling *OrangePi2Face Emotion* function.

6. Using speech recognition service which based on OrangePi compilaion.

You could compile by referenced to source code, *Audio2Command.c*, on the sub directory of *OrangePiSpeech2Text/src*, shown as *Figure 5.3 OrangePi & Speech2Text Demo Code*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, int argv)
6 {
7     FILE *fd = fopen("../base/julius.out", "r");
8     char buffer[200];
9     char *line;
10
11     memset(buffer, 0, 200);
12
13     fread(buffer, 200, 1, fd);
14
15     /* Analysis the sentence */
16     line = strstr(buffer, "CLOSE");
17     if (line != NULL)
18         system("OrangePi 0");
19     else
20         system("OrangePi 255");
21     fclose(fd);
22
23     return 0;
24 }
```

Figure 5.3 OrangePi & Speech2Text Demo Code

Speech recognition of OrangePi is based on open software Juliu. You could programme by referenced to the *Figure 5.3 OrangePi & Speech2Text Demo Code* first, and then built up voice dictionary on *sample.voca* of *OrangePi2MicrosoftService/OrangePiSpeech2Text/base*. Finally you could use the script *Speech2Text.sh* to record and voice recognition, and upload information to control the lights on or off.

Github: <https://github.com/zhaoyifan/OrangePi2MicrosoftAzure.git>