

基于 OrangePi 的 Microsoft Azure IOT 使用文档

一. 简介

为了让开发者快速简单地在 OrangePi 上使用 Microsoft Azure IOT，我们提供该手册来说明如何快速的搭建开发环境。在搭建开发环境前，我们需准备一个 OrangePi 开发版，OrangePi & Microsoft Azure IOT 固件包或移植包，固件包随套件一起发布，移植包可通过 [github](https://github.com) 获得。

二. Quick Start

通过申请微软发布的套件：Microsoft Iot Start Kit, 我们可以获得 OrangePi 开发版一块，烧录有固件的 SD 卡一张，摄像头及摄像头连接线，以及电源线一根。如下图 1 所示：

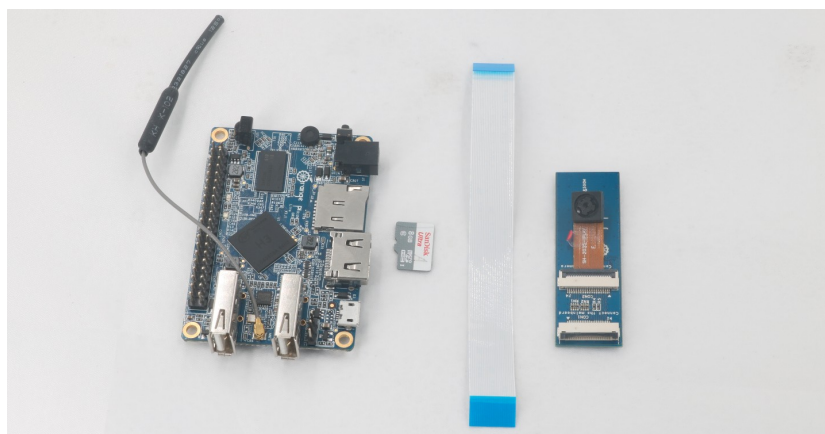


图 1. OrangePi 开发部件

1. 将 SD 卡插入到 OrangePi 开发板的 SD 卡槽内，再将摄像头和摄像头数据线连接到开发板上，连接方式如下图 2.1 和 图 2.2，请注意连接线的连接方式，蓝色面背对接口。

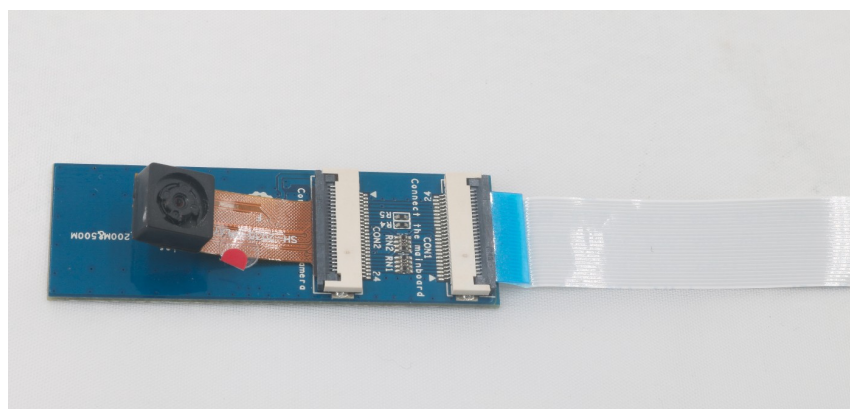


图 2.1 摄像头模块端连接方式



图 2.2 OrangePi 端连接图

2. 准备一个 wifi 热点，将 wifi 热点的名字和密码都设置为 “iotstarterkit”，重启 wifi 并插上 OrangePi 板子的电源上电。
3. 在自己笔记本或电脑上，连接 wifi 热点，并使用相关 IP 地址查找工具，查找 OrangePi 的 IP 地址。
4. 获得 OrangePi 的 IP 地址之后，打开 Host 端的浏览器，输入 OrangePi 的 IP 地址。登陆到主界面，如图 2.3:

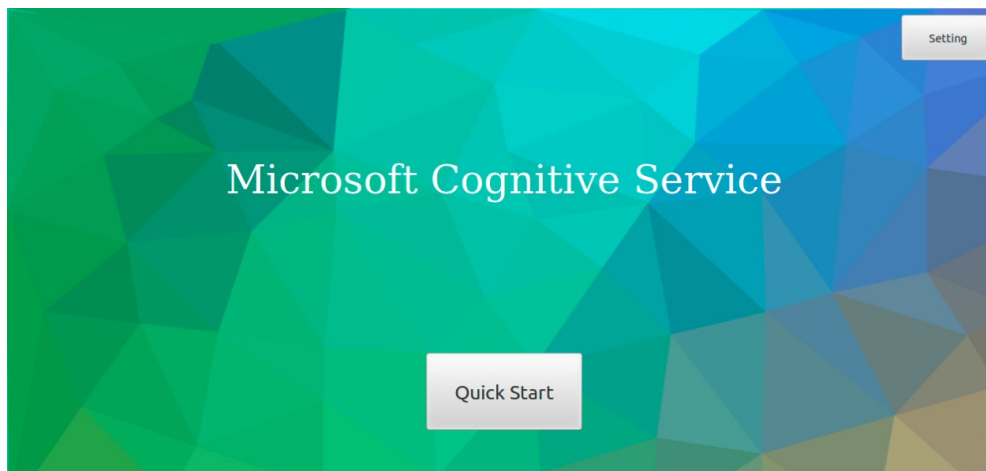


图 2.3 主界面

5. 第一次使用，请点击右上角的 setting 进行初始化设置。如图 2.4 :

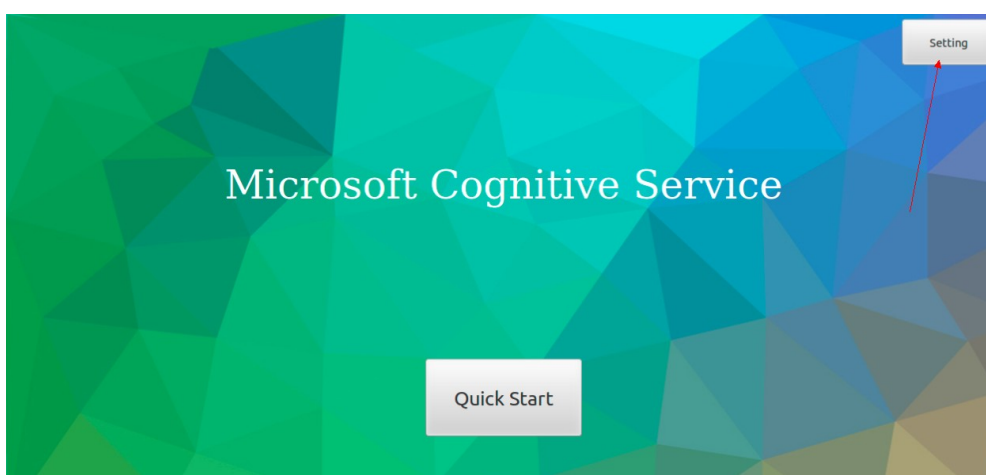


图 2.4 Setting 选择

6. 根据输入框的提示，输入相应的 Azure Iot 信息，以及 Microsoft Cognitive Service 信息并确认。如图 2.5

图 2.5 初始化信息输入

7. 点击 “Quick Start” 按钮之后，进入功能界面。开发者可以选择语音控制和图片认知服务。如图 2.6

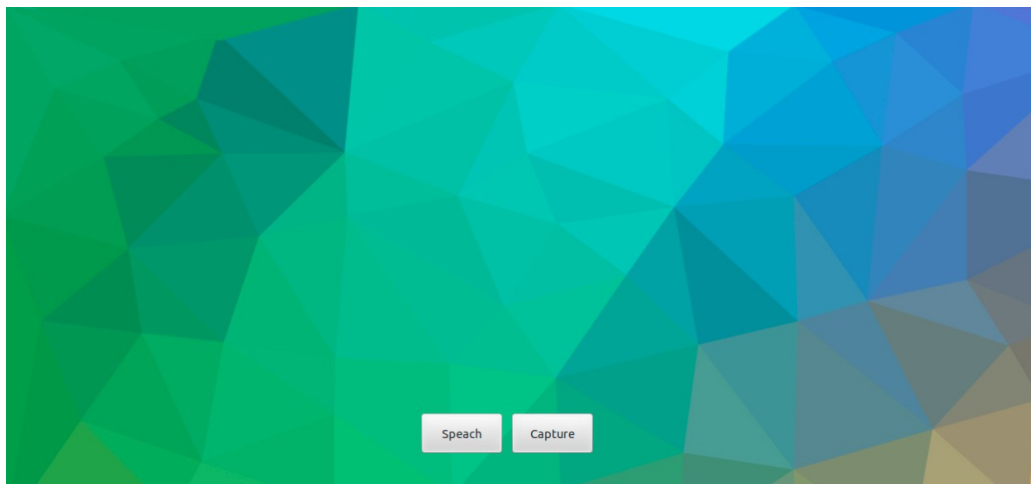


图 2.6 功能界面

三. 图片认知服务的使用

在通过上面的步骤进入图片认知服务之后，可以通过摄像头拍摄的实时视频流预览图片，并进行图片认知服务。具体步骤如下：

1. 点击 “capture” 进入实时视频浏览界面，开发者将可以从 OrangePi 获得实时的视频流，点击 “capture” 按钮进行抓图。如图 3.1

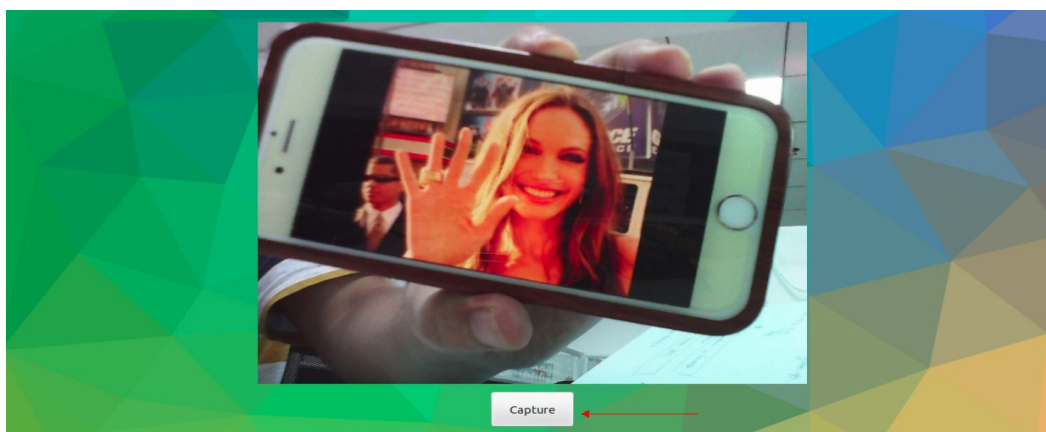


图 3.1 mjpg 视频流

2. 系统会对抓取之后的图进行预览，开发者可以根据抓图的效果进行重抓和确认上传操作。点击 “upload” 按钮之后，抓取的图片将被上传至 Microsoft Cognitive Service 上进行处理。如图 3.2

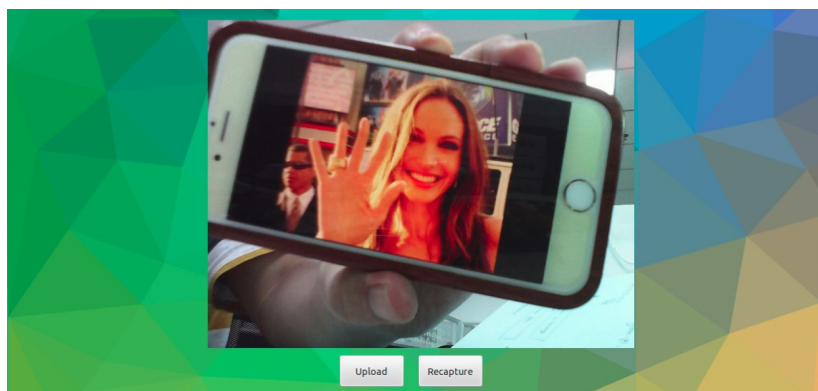


图 3.2 预览截图

3. 处理之后的图片将会和原图一起显示。如图 3.2 和 图 3.3:

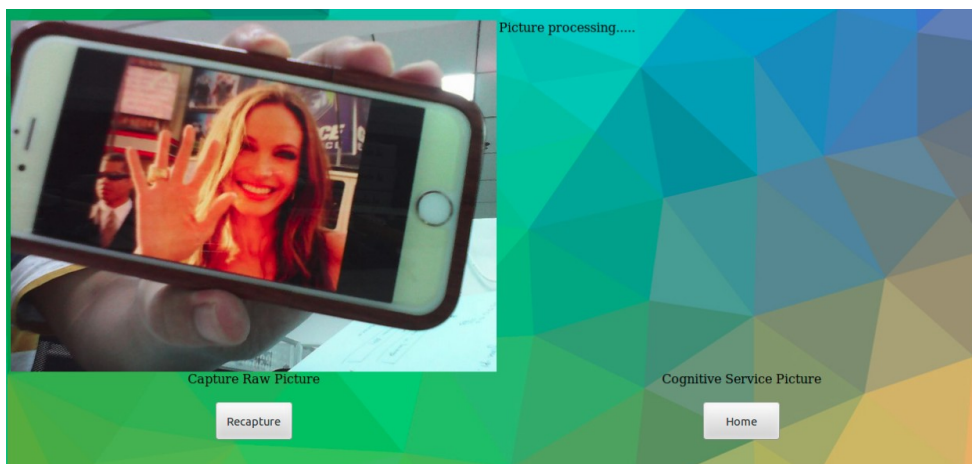
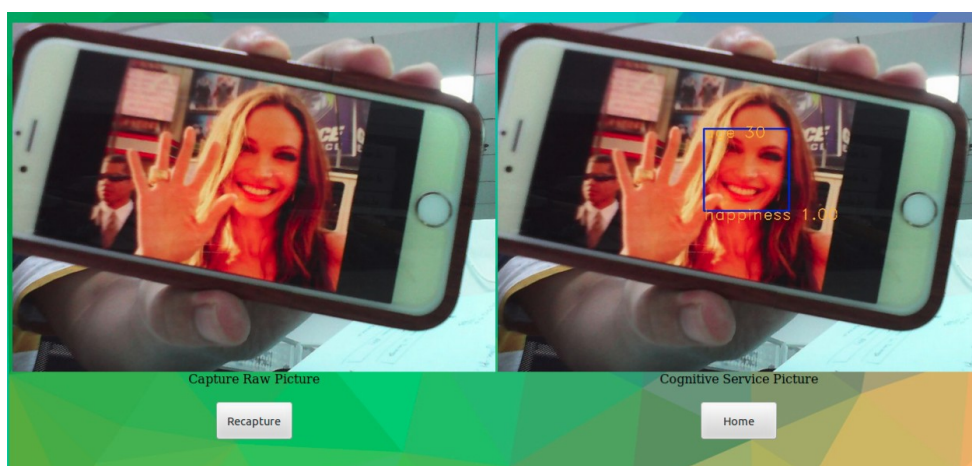


图 3.2 图片上传至 Microsoft Cognitive Service



4. 图 3.2 处理之后的对比图

四. 语音识别模块

语音识别模块给开发者提供了语音控制 IOT 设备的功能。开发者可以通过对 OrangePi 开发板发“open”和“close”命令，以此来控制灯的亮灭。具体步骤如下：

1. 在主界面点击“Quick Start”之后，进入功能界面。点击“speech”按钮，此时请面向 OrangePi 开发板发声，“open”可以打开 LED 灯，“close”可以关闭 LED 灯。如图 4.1

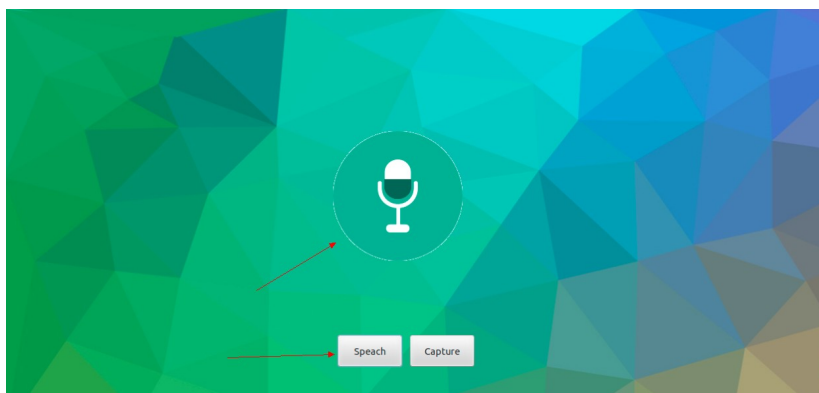


图 4.1 语音识别

2. 但发完口令之后，IOT 端的 LED 就会根据命令亮灭。

五. 源码编译

当我们在一个全新的 OrangePi 上移植 Microsoft Azure IOT，首先我们需要安装其运行所需要的库。请参照下面步骤进行移植：

1. 请插上 OrangePi 的电源线并登陆 OrangePi，在终端输入以下命令：

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

2. 通过 [github](#) 下载源码，源码目录分别如下：

Azure_IOTLib : OrangePi 与 Microsoft Azure IOT 的中间层，主要实现数据上传到 Azure。

Mjpg-streamer : 基于 OrangePi 的 mjpeg 实时视频数据流。

OrangePiSpeech2Text : 基于 Julius 和 Alsa 的语音识别模块。

PortLib : 移植相关的库

SoundLib : Alsa 相关的语音接口。

VideoLib : 基于 V4L2 的视频采集以及图片的 Cognitive Service 接口实现。

Websit : 网站相关的文件。

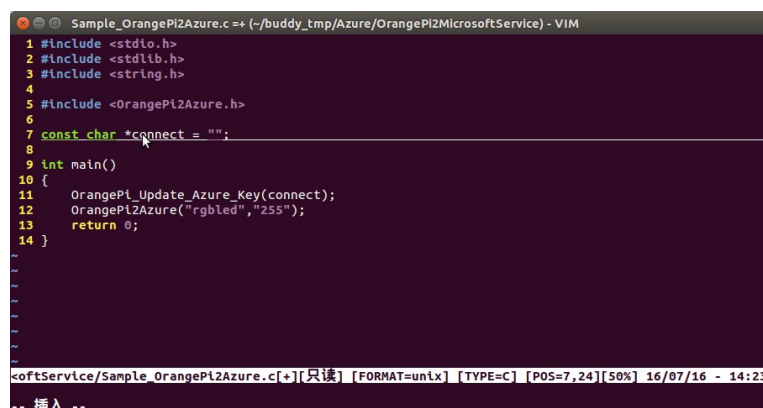
Sample_OrangePi2Azure.c : 基于 OrangePi 的 Azure IOT Demo Code。

Sample_OrangePi2Sound.c : 基于 OrangePi 的语音流采集的 Demo Code。

Sample_OrangePi2Video.c : 基于 OrangePi 的 Cognitive Service Demo Code。

3. 基于 OrangePi 编写程序来使用 Azure IOT 服务。

用户可以参照源码 *Sample_OrangePi2Azure.c* 进行编程，如图 5.1



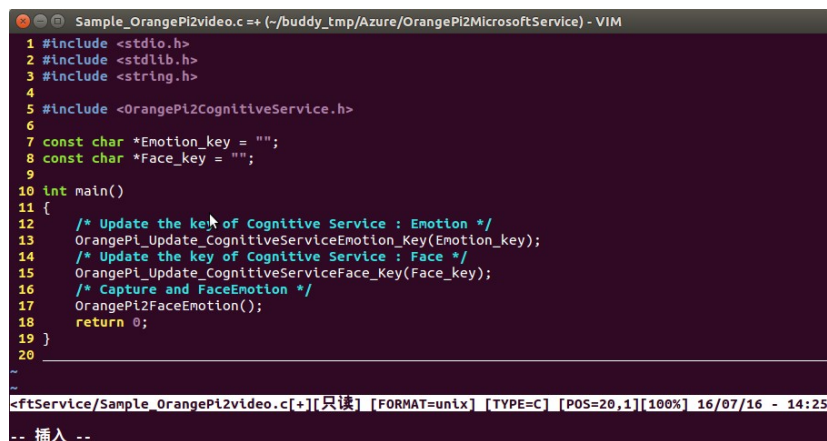
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #include <OrangePi2Azure.h>
6
7 const char *connect = "";
8
9 int main()
10 {
11     OrangePi_Update_Azure_Key(connect);
12     OrangePi2Azure("rgbled", "255");
13     return 0;
14 }
```

图 5.1 OrangePi & Azure IOT Demo Code

源码中通过引用头文件 *OrangePi2Azure.h*，将 Azure IOT 注册字符串添入 *connect* 中。调用 *OrangePi_Update_Key* 函数用来更新本地 Azure 配置信息，调用 *OrangePi2Azure* 函数来上传自定义信息到 Azure。

4. 基于 OrangePi 编写程序来使用 Cognitive Service 服务。

用户可以参照源码 *Sample_OrangePi2video.c* 进行编程，如图 5.2



```
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
```


图 5.2 OrangePi & Cognitive Service Demo Code

源码中通过引用头文件 *OrangePi2CognitiveService.h*，将 Cognitive Service 服务中，Emotion 和 Face 注册字符串分别添加到 *Emotion_key* 和 *Face_key* 中。调用

OrangePi_Update_CognitiveServiceEmotion_Key 函数用来更新本地 Emotion Preview 配置信息，调用 OrangePi_Update_CognitiveServiceFace_Key 函数用来更新本地 Face Preview 配置信息，调用 OrangePi2FaceEmotion 函数来上传图片进行处理，并获得处理后的图片。

5. 基于 OrangePi 编写程序来使用语音识别服务。

用户可以参照源码 OrangePiSpeech2Text/src 子目录下的源码 Audio2Command.c 进行编程，如图 5.3



```
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     fread(buffer, 200, 1, fd);
13
14     /* Analysis the sentence */
15     line = strstr(buffer, "CLOSE");
16     if (line != NULL)
17         system("OrangePi 0");
18     else
19         system("OrangePi 255");
20     fclose(fd);
21
22     return 0;
23 }
24
```

<OrangePiSpeech2Text/src/Audio2Command.c [只读] [FORMAT=unix] [TYPE=C] [POS=1,1][4%] 18/07/16 - 11:25

图 5.3 OrangePi & Speech2Text Demo Code

OrangePi 的语音识别基于开源软件 Julius，用户可以参照上面代码进行编程，并在 OrangePi2MicrosoftService/OrangePiSpeech2Text/base 下 sample.voca 自建语音字典。最后使用脚本 Speech2Text.sh 进行录音并进行语音识别，最后上传信息，控制灯的亮灭。

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