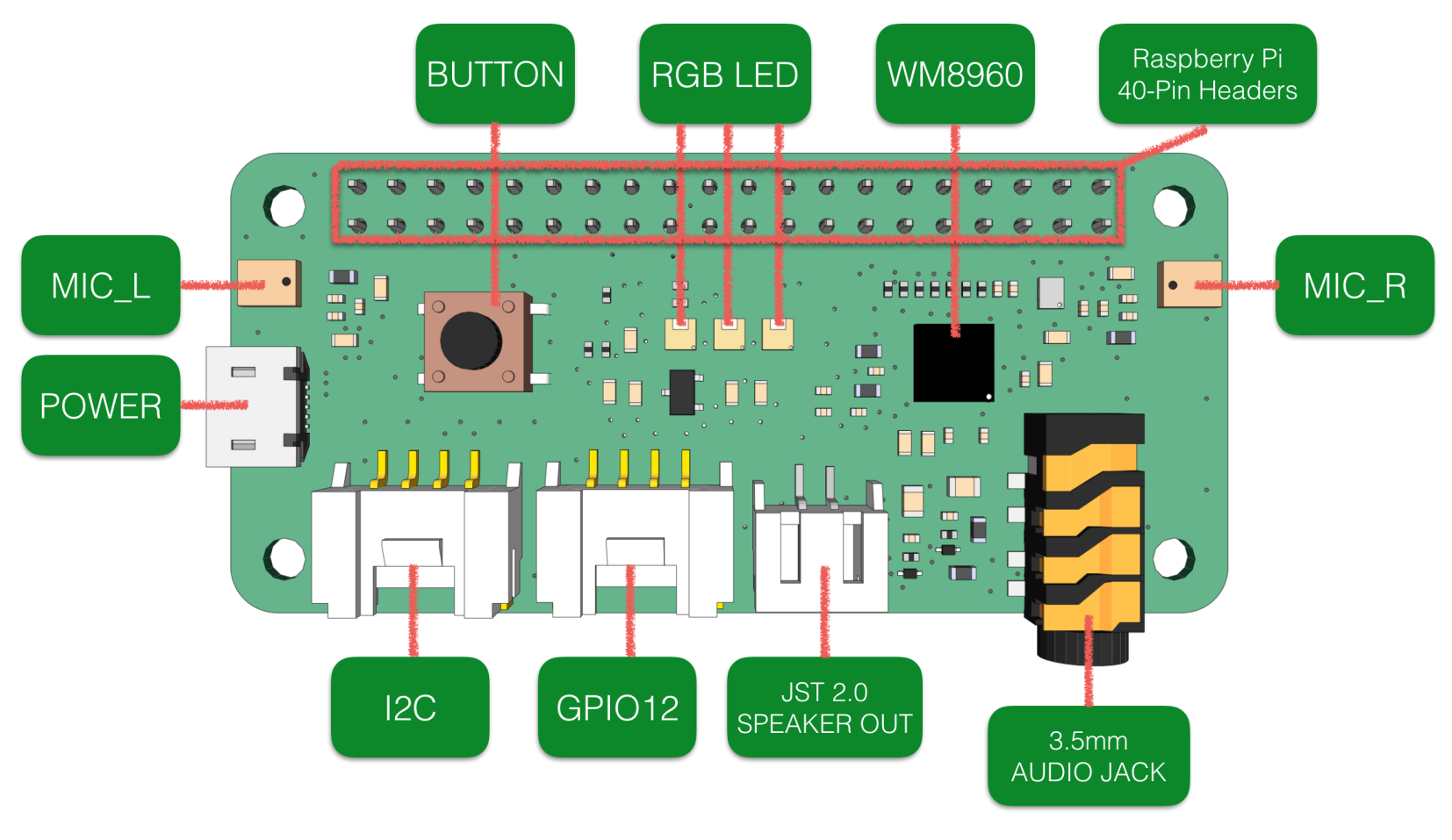
产品特征[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "_1" \o "Permanent link" \t "_blank)

* Raspberry Pi兼容（支持Raspberry Pi Zero和Zero W，Raspberry Pi B +，Raspberry Pi 2 B和Raspberry Pi 3 B）
* 2个麦克风
* 2个Grove接口
* 1个自定义按钮
* 3.5mm音频接口
* JST2.0音频输出接口

创意应用[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#_2)

* 声音交互应用
* AI助手

硬件概述[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#_3)



* 按钮：连接到GPIO17的用户自定义按钮
* MIC\_L＆MIC\_R：左边右边各有一个麦克风
* RGB LED：3个APA102 RGB LED，连接到树莓派的SPI接口
* WM8960：低功耗立体声编解码器
* Raspberry Pi 40针头：支持Raspberry Pi Zero，Raspberry Pi 1 B +，Raspberry Pi 2 B和Raspberry Pi 3 B
* POWER：用于为ReSpeaker 2-Mics Pi HAT供电的Micro USB端口，请在使用扬声器时为电路板供电，以提供足够的电流。
* I2C：Grove I2C端口，连接到I2C-1
* GPIO12：Grove数字端口，连接到GPIO12和GPIO13
* JST 2.0 SPEAKER OUT：用于连接扬声器，JST 2.0连接器
* 3.5mm音频插孔：用于连接带3.5mm音频插头的耳机或扬声器

入门指导[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#_4)

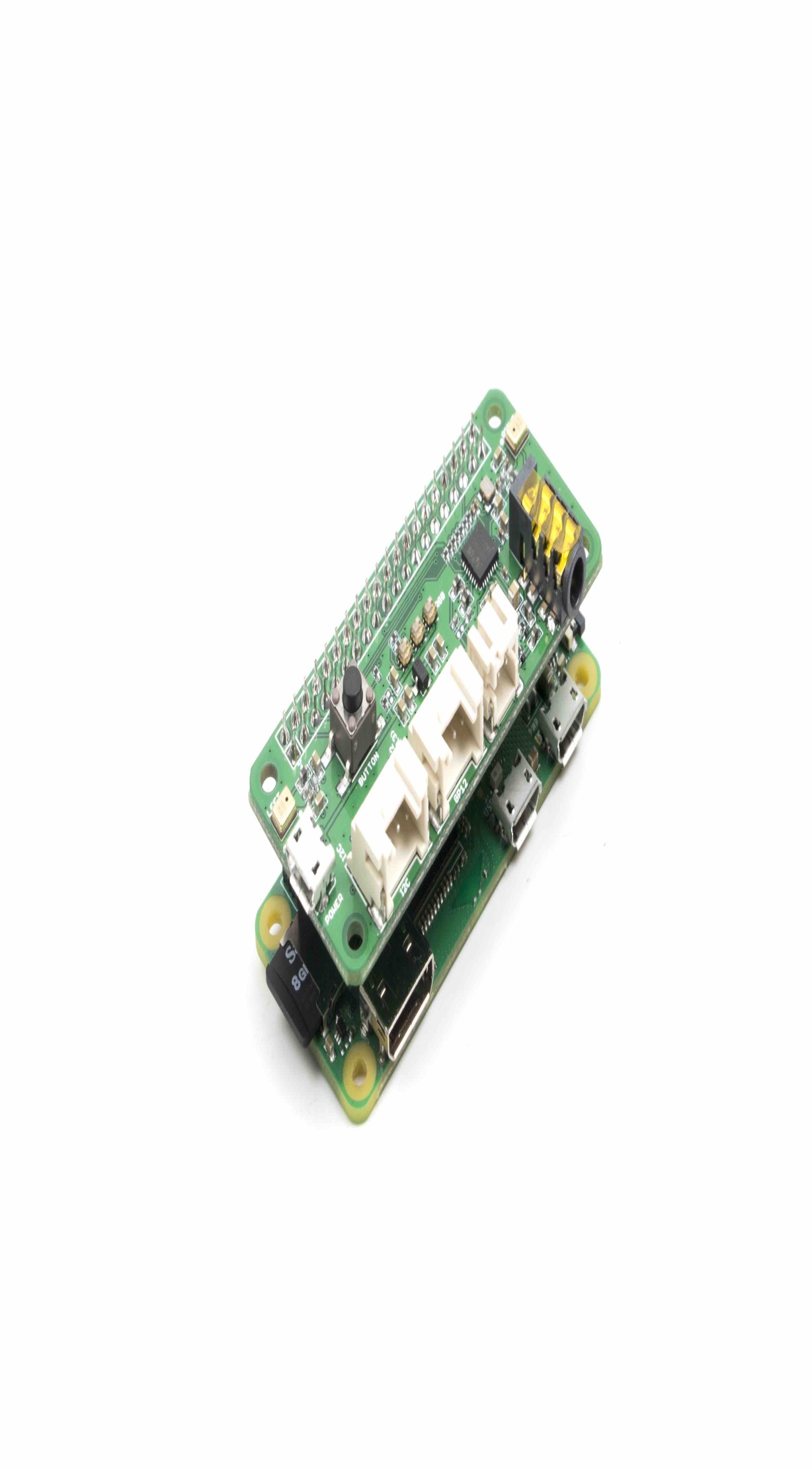
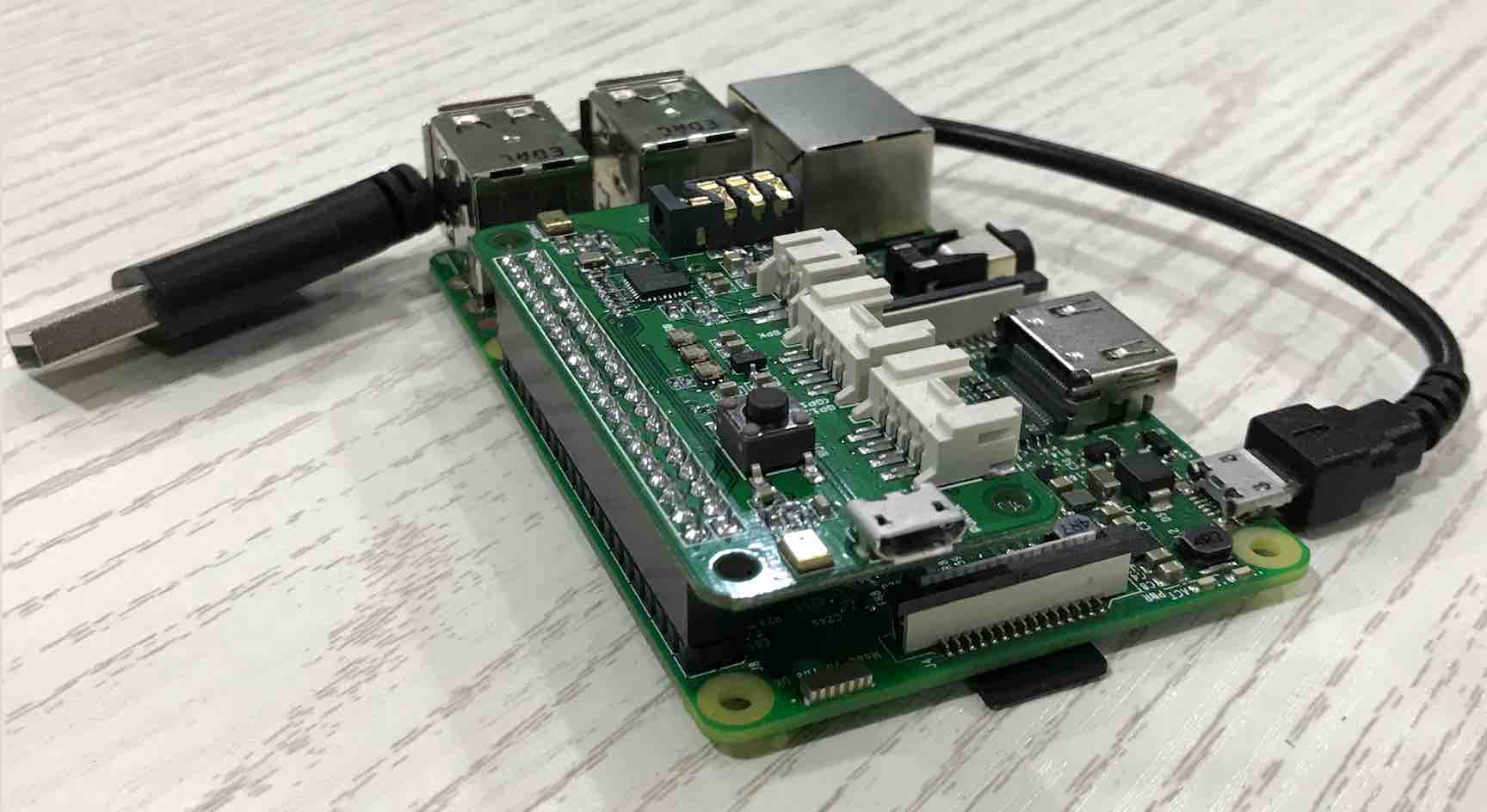
1. 系统配置与驱动安装[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "1" \o "Permanent link" \t "_blank)

**step 1. 把ReSpeaker 2-Mics Pi HAT插入到Raspberry Pi**

把 ReSpeaker 2-Mics Pi HAT 插入到 Raspberry Pi, 确保插入Raspberry Pi的时候针脚对齐。

**Note**

不要在上电的时候，热插拔ReSpeaker 2-Mics Pi HAT.



**step 2. 烧录系统，登陆，换源**

因为当前的Pi内核目前不支持wm8960编解码器，所以我们需要手动构建。

1. 确保您正在您的Pi上运行[最新的Raspbian操作系统（debian 9）](https://www.raspberrypi.org/downloads/raspbian/" \t "_blank)。 *（更新于2018.06.27）*，您可以用etcher进行系统烧录
2. 您可以用 [VNC](https://www.raspberrypi.org/documentation/remote-access/vnc/)或者PUTTY连接树莓派，但之前请配置好wifi
3. 在安装驱动之前，请根据以下流程切换源到清华。

pi@raspberrypi ~ $ sudo nano /etc/apt/sources.list

用#注释掉原文件内容，用以下内容取代：

deb http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ stretch main non-free contrib

deb-src http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ stretch main non-free contrib

**step 3. 驱动下载并安装** 运行下面命令

sudo apt-get update

sudo apt-get upgrade

git clone https://github.com/respeaker/seeed-voicecard.git

cd seeed-voicecard #下载声卡驱动

sudo ./install.sh #安装声卡驱动

reboot #重启

**step 4. 检查声卡名称是否与源代码seeed-voicecard相匹配.**

pi@raspberrypi:~/seeed-voicecard $ aplay -l

\*\*\*\* List of PLAYBACK Hardware Devices \*\*\*\*

card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]

Subdevices: 8/8

Subdevice #0: subdevice #0

Subdevice #1: subdevice #1

Subdevice #2: subdevice #2

Subdevice #3: subdevice #3

Subdevice #4: subdevice #4

Subdevice #5: subdevice #5

Subdevice #6: subdevice #6

Subdevice #7: subdevice #7

card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]

Subdevices: 1/1

Subdevice #0: subdevice #0

card 1: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835-i2s-wm8960-hifi wm8960-hifi-0 []

Subdevices: 1/1

Subdevice #0: subdevice #0

pi@raspberrypi:~/seeed-voicecard $ arecord -l

\*\*\*\* List of CAPTURE Hardware Devices \*\*\*\*

card 1: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835-i2s-wm8960-hifi wm8960-hifi-0 []

Subdevices: 1/1

Subdevice #0: subdevice #0

pi@raspberrypi:~/seeed-voicecard $

2. 录音播放测试[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "2" \o "Permanent link" \t "_blank)

**step 1. 录播测试** 可以用arecord录制，然后用aplay播放：(不要忘记插耳机或者喇叭):

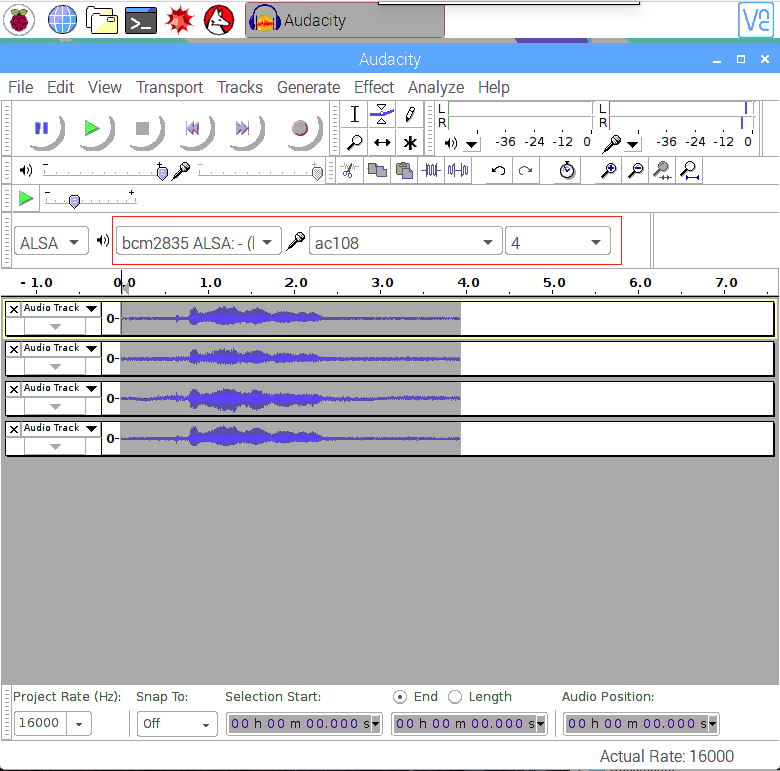
arecord -f cd -Dhw:1 | aplay -Dhw:1

也可以通过audacity软件测试。打开Audacity后，选择 **AC108和2通道** 作为输入，**bcm2835 alsa: - (hw：0，0)** 作为输出来测试：

$ sudo apt update

$ sudo apt install audacity

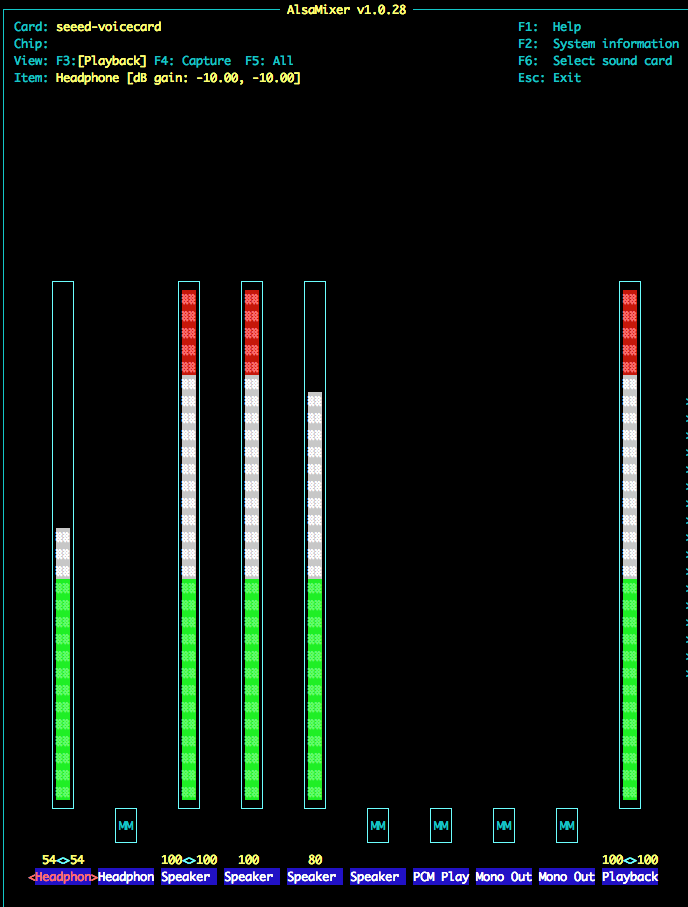
$ audacity // 运行 audacity



**step 2. 调节音量（可跳过）**

**alsamixer** 是用于配置声音设置和调整音量，高级Linux声音体系结构（ALSA）的图形混音器程序。

pi@raspberrypi:~ $ alsamixer



**Note**

首先请用F6选择seeed-2mic的声卡设备。

左和右箭头键用于选择通道或设备，“向上和向下箭头”控制当前所选设备的音量。 退出程序使用ALT + Q或按Esc键。 [More information](https://en.wikipedia.org/wiki/Alsamixer)

3. 安装python和虚拟环境[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "3-python" \o "Permanent link" \t "_blank)

这样是是为了隔离SDK与系统Python包关系。

pi@raspberrypi:~ $ cd /home/pi

pi@raspberrypi:~ $ git clone https://github.com/respeaker/4mics\_hat.git

pi@raspberrypi:~ $ cd /home/pi/4mics\_hat

pi@raspberrypi:~/4mics\_hat $ sudo apt install python-virtualenv # 安装 python2 虚拟环境工具

pi@raspberrypi:~/4mics\_hat $ virtualenv --system-site-packages ~/env # 建立虚拟环境，命名位env,放在~目录下

pi@raspberrypi:~/4mics\_hat $ source ~/env/bin/activate # 激活虚拟环境

(env) pi@raspberrypi:~/4mics\_hat $ pip install spidev gpiozero # 安装需要的工具包

Google Assistant SDK[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#google-assistant-sdk)

**Warning**

因为我们在中国，无法直接使用Google的服务。必须搭建可以访问google的路由器，然后连接到路由。

1. 配置流程[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "1_1" \o "Permanent link" \t "_blank)

在开始使用[Google Assistant](https://assistant.google.com/" \t "_blank)之前，首先您应该将Google Assistant Library整合到您的raspberry pi系统中。 以下是[Google官方指导](https://developers.google.com/assistant/sdk/prototype/getting-started-pi-python/run-sample)的链接。

以下指南还将向您介绍如何开始使用Google助手。

**step 1. 配置开发人员项目，并获取JSON文件**

请根据[指南](https://developers.google.com/assistant/sdk/prototype/getting-started-pi-python/config-dev-project-and-account#config-dev-project) 第一步到第四步在Google Cloud Platform上配置项目，并创建一个OAuth Client ID JSON文件。 不要忘记将JSON文件复制到您的Raspberry Pi。

**step 2. 安装google-assistant-library**

Google Assistant SDK软件包，包含在设备上运行Google Assistant所需的所有代码，包括库和示例代码。 使用pip在虚拟环境中安装最新版本的Python包：

source env/bin/activate #打开虚拟环境

(env) $ python -m pip install --upgrade google-assistant-library

**step 3. 授权Google Assistant SDK**

授权Google Assistant SDK，使Google Assistant对给定的Google帐户进行查询。 把步骤1中的JSON文件复制到树莓派/home/pi下。

pi@raspberrypi:~ $ google-oauthlib-tool --client-secrets /home/pi/client\_secret\_client-id.json --scope https://www.googleapis.com/auth/assistant-sdk-prototype --save --headless

**/home/pi/client\_secret\_client-id.json** 是你的JSON文件的路径，确保Json文件的名字匹配。 运行命令后，应该显示如下所示。 复制URL并将其粘贴到浏览器中（这可以在您的树莓派或任何其他电脑上完成）。 同意后，您的浏览器将显示代码，例如“4 / XXXX”。 复制并将此代码粘贴到终端中。

Please go to this URL: https://...

Enter the authorization code:

这个时候应该显示: OAuth credentials initialized. 如果显示: InvalidGrantError then an invalid code was entered. 请重试, 确保拷贝整个code.

**step 4. 安装 pulseaudio 并且让他在后台运行**

pi@raspberrypi:~ $ sudo apt install pulseaudio

pi@raspberrypi:~ $ pulseaudio &

[1] 1244

pi@raspberrypi:~ $ W: [pulseaudio] server-lookup.c: Unable to contact D-Bus: org.freedesktop.DBus.Error.NotSupported: Unable to autolaunch a dbus-daemon without a $DISPLAY for X11

W: [pulseaudio] main.c: Unable to contact D-Bus: org.freedesktop.DBus.Error.NotSupported: Unable to autolaunch a dbus-daemon without a $DISPLAY for X11

E: [pulseaudio] bluez4-util.c: org.bluez.Manager.GetProperties() failed: org.freedesktop.DBus.Error.UnknownMethod: Method "GetProperties" with signature "" on interface "org.bluez.Manager" doesn't exist

**Note**

请忽略pulseaudio错误信息。

**step 5. 开始使用Google Assistant示例**

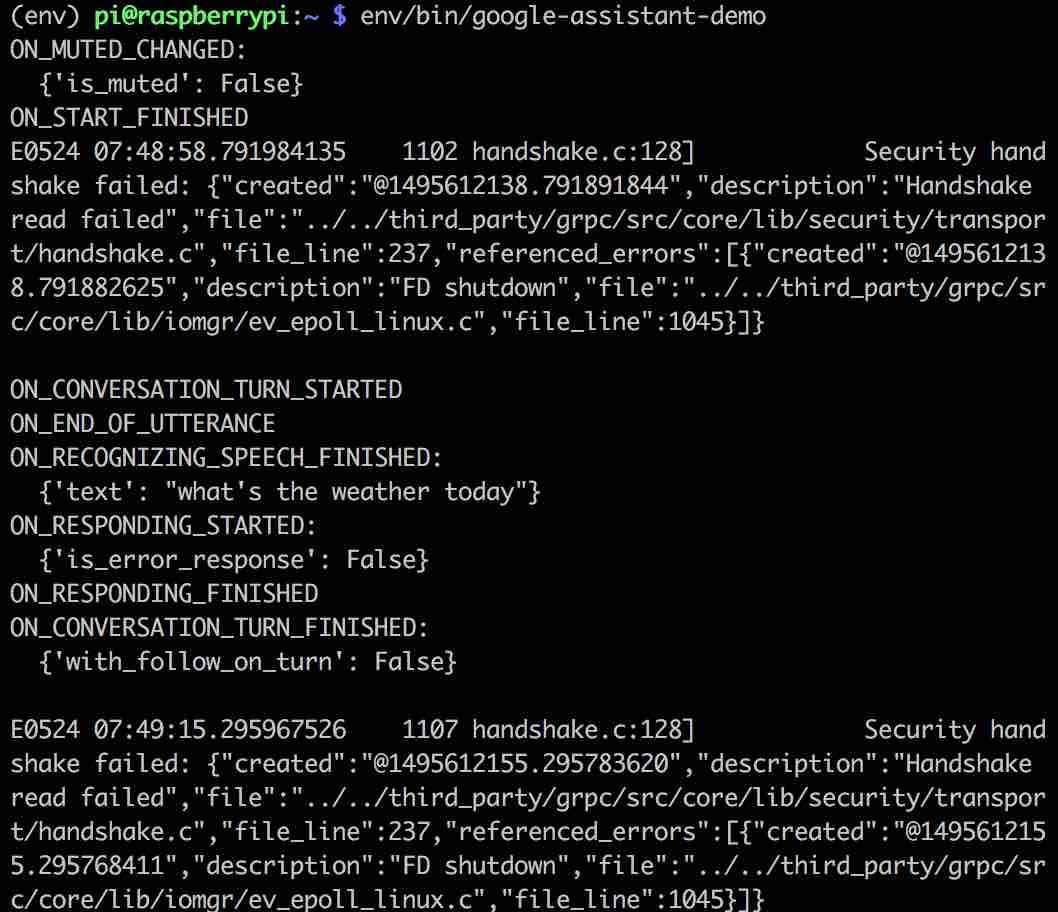
pi@raspberrypi:~ $ alsamixer // To adjust the volume

pi@raspberrypi:~ $ source env/bin/activate

(env) pi@raspberrypi:~ $ env/bin/google-assistant-demo

**step 6. 唤醒Google Assistant**

先说 *Ok Google* 或者 *Hey Google*, 然后说您的询问. 语音助手就会响应您的问题。如果语音助手没有响应， 请按照 [疑难解答说明](https://developers.google.com/assistant/sdk/prototype/getting-started-pi-python/troubleshooting#hotword).

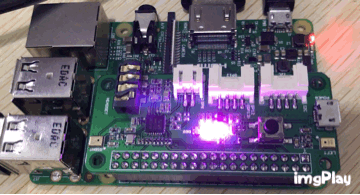


2.常见问题解决方法[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "2_1" \o "Permanent link" \t "_blank)

如果您遇到问题，请参考 [常见疑难解答说明](https://developers.google.com/assistant/sdk/prototype/getting-started-pi-python/troubleshooting) 。

3. 控制APA102 LED的示例[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "3-apa102-led" \o "Permanent link" \t "_blank)

每个板载APA102 LED都有一个额外的驱动芯片，驱动芯片设置LED的颜色，然后保持该颜色，直到接收到新的命令。



请在执行之前打开SPI，具体步骤如下:

- 输入： `sudo raspi-config`;

- 选择 "Interfacing Options";

- 选择 "SPI";

- 选择 “Yes”

- 选择 “OK”

- 选择 “Finish”

配置完后，可以执行下列命令行来运行led示例

cd ~/

git clone https://github.com/respeaker/mic\_hat.git

sudo pip install spidev #安装spi的驱动

cd mic\_hat

python pixels.py

4.如何使用用户自定义按钮[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "4" \o "Permanent link" \t "_blank)

板子上面有个用户自定义按钮，连接到GPIO17. 我们可以调用python和RPi.GPIO来读取状态。

sudo pip install rpi.gpio // install RPi.GPIO library

nano button.py // copy the following code in button.py

import RPi.GPIO as GPIO

import time

BUTTON = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(BUTTON, GPIO.IN)

while True:

state = GPIO.input(BUTTON)

if state:

print("off")

else:

print("on")

time.sleep(1)

Save the code as button.py, then run it. It should display "on" when you press the button:

pi@raspberrypi:~ $ python button.py

off

off

on

on

off

5. 用按钮来触发Google Assisant[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "5-google-assisant" \o "Permanent link" \t "_blank)

您可以用按键来代替"ok google"来激活Google Assisant.

* 更新 pushtotalk.py

cd /usr/local/lib/python2.7/dist-packages/googlesamples/assistant/grpc

sudo nano pushtotalk.py

请到文件第301行, 然后根据下面的code来更新。

with SampleAssistant(conversation\_stream,

grpc\_channel, grpc\_deadline) as assistant:

# If file arguments are supplied:

# exit after the first turn of the conversation.

if input\_audio\_file or output\_audio\_file:

assistant.converse()

return

# If no file arguments supplied:

# keep recording voice requests using the microphone

# and playing back assistant response using the speaker.

# When the once flag is set, don't wait for a trigger. Otherwise, wait.

wait\_for\_user\_trigger = not once

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(17,GPIO.IN)

while True:

if wait\_for\_user\_trigger:

state = GPIO.input(17)

logging.info('Press the button to send a new request...')

if state:

continue

else:

pass

# click.pause(info='Press Enter to send a new request...')

continue\_conversation = assistant.converse()

# wait for user trigger if there is no follow-up turn in

# the conversation.

wait\_for\_user\_trigger = not continue\_conversation

# If we only want one conversation, break.

if once and (not continue\_conversation):

break

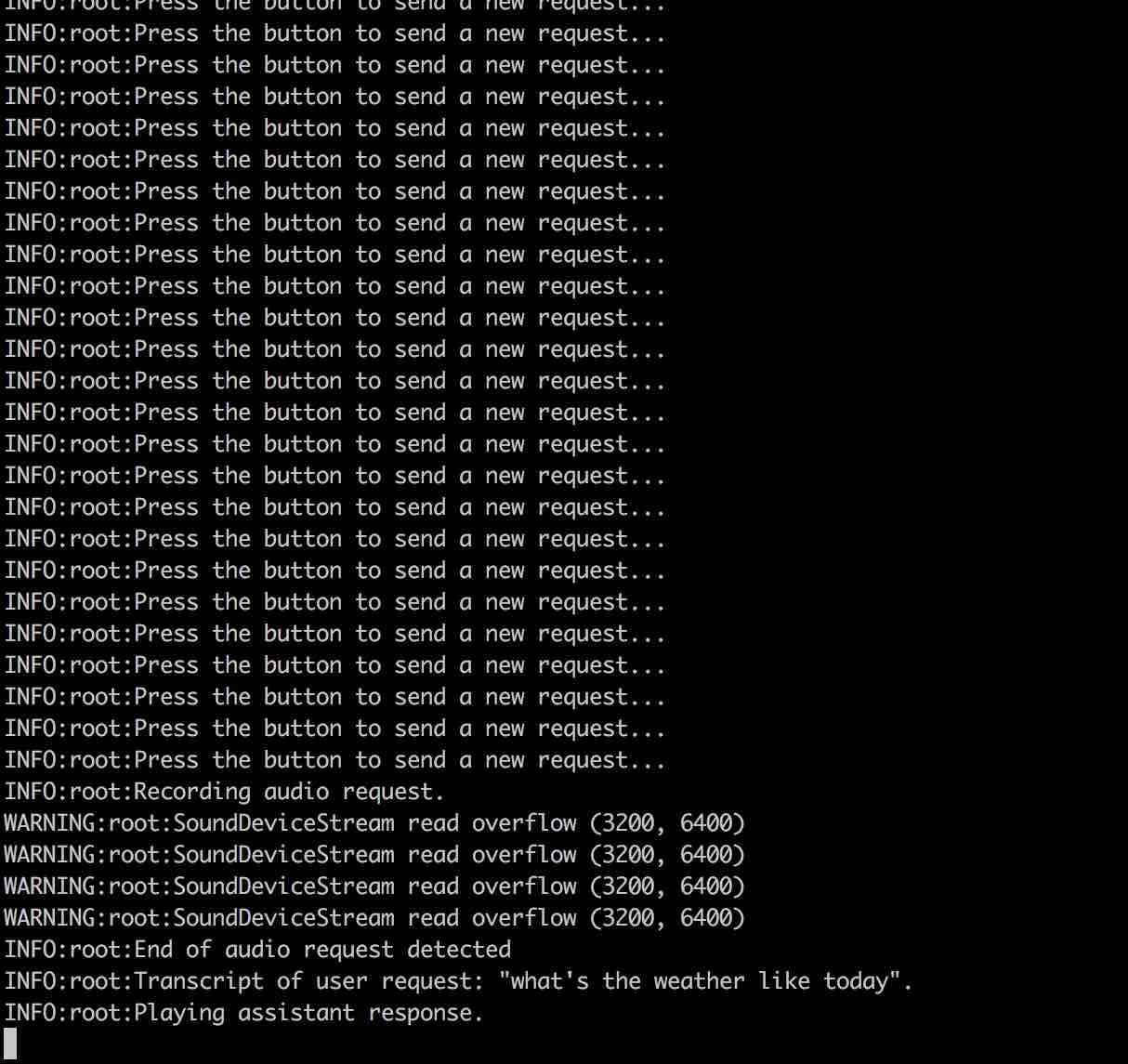
if \_\_name\_\_ == '\_\_main\_\_':

main()

* 运行下面程序来进行测试:

$ googlesamples-assistant-pushtotalk

* 程序运行的结果如下图所示:



Alexa SDK 和 DuerOs SDK[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#alexa-sdk-dueros-sdk)

由于国内登录不上 Google Assisant ，所以使用在国内能连接的 Alexa 和 百度 DuerOs 作为语音引擎，开发出能让大多数人使用的语音互动系统。

1. 配置和DOA测试[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "1-doa" \o "Permanent link" \t "_blank)

**step 1. 配置 Voice engine**

pi@raspberrypi:~ $ source ~/env/bin/activate # 激活Python虚拟环境, 如果已经激活，调到下一步。

(env) pi@raspberrypi:~ $ cd ~/4mics\_hat

(env) pi@raspberrypi:~/4mics\_hat $ sudo apt install libatlas-base-dev # 安装 snowboy dependencies

(env) pi@raspberrypi:~/4mics\_hat $ sudo apt install python-pyaudio #安装pyaudio音频处理包

(env) pi@raspberrypi:~/4mics\_hat $ pip install ./snowboy\*.whl # 安装 snowboy for KWS

(env) pi@raspberrypi:~/4mics\_hat $ pip install ./webrtc\*.whl # 安装 webrtc for DoA

(env) pi@raspberrypi:~ $ cd ~/

(env) pi@raspberrypi:~ $ git clone https://github.com/voice-engine/voice-engine #write by seeed

(env) pi@raspberrypi:~ $ cd voice-engine/

(env) pi@raspberrypi:~ $ python setup.py install

(env) pi@raspberrypi:~ $ cd examples

(env) pi@raspberrypi:~ $ nano kws\_doa.py

**step 2. 修改**kws\_doa.py**的第14-21行，以适应 2-Mics：**

from voice\_engine.doa\_respeaker\_4mic\_array import DOA

def main():

src = Source(rate=16000, channels=2)

ch1 = ChannelPicker(channels=2, pick=1)

kws = KWS()

doa = DOA(rate=16000)

然后保存退出。

**step 3. 运行**

在虚拟环境下运行 python kws\_doa.py。请用 snowboy 来唤醒，我们就可以看到方位的信息。

2. 百度中文语音互动或者alexa英文语音互动[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/" \l "2-alexa" \o "Permanent link" \t "_blank)

**step 1. 配置和安装相关依赖**

pi@raspberrypi:~ $ source ~/env/bin/activate # activate the virtual, if we have already activated, skip this step

(env) pi@raspberrypi:~ $ cd ~/

(env) pi@raspberrypi:~ $ git clone https://github.com/respeaker/avs

(env) pi@raspberrypi:~ $ cd avs # install Requirements

(env) pi@raspberrypi:~ $ python setup.py install

(env) pi@raspberrypi:~/avs $ sudo apt install gstreamer1.0

(env) pi@raspberrypi:~/avs $ sudo apt install gstreamer1.0-plugins-good

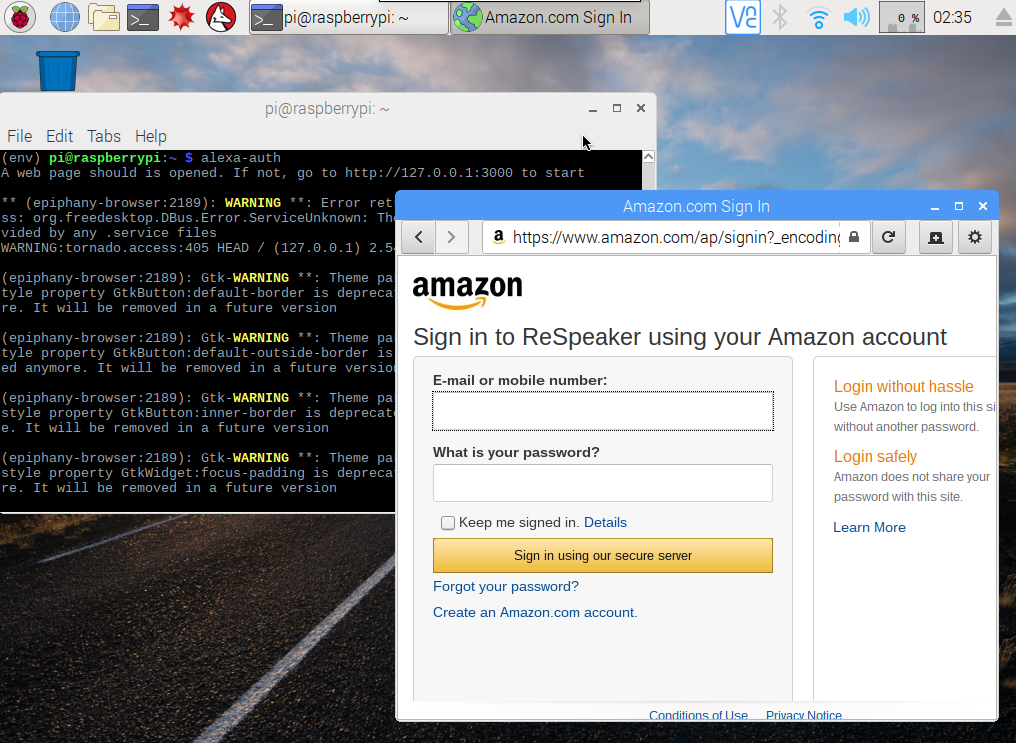
(env) pi@raspberrypi:~/avs $ sudo apt install gstreamer1.0-plugins-ugly

(env) pi@raspberrypi:~/avs $ sudo apt install python-gi gir1.2-gstreamer-1.0

(env) pi@raspberrypi:~/avs $ pip install tornado

**step 2. 取得授权**

在终端运行 alexa-auth ，然后登陆获取alexa的授权， 或者运行 dueros-auth 获取百度的授权。 授权的文件保存在/home/pi/.avs.json。



**Note**

如果我们在 alexa-auth 和 dueros-auth之间切换, 请先删除 /home/pi/.avs.json 。 这个是隐藏文件，请用 ls -la 显示文件。

**step 2. 配置**

(env) pi@raspberrypi:~ $ cd /home/pi

(env) pi@raspberrypi:~ $ git clone https://github.com/respeaker/respeaker\_v2\_eval.git

(env) pi@raspberrypi:~ $ cd respeaker\_v2\_eval/alexa

(env) pi@raspberrypi:~/respeaker\_v2\_eval/alexa $ cp ~/4mics\_hat/pixels.py ./pixels.py

(env) pi@raspberrypi:~/respeaker\_v2\_eval/alexa $ nano ns\_kws\_doa\_alexa.py

按照下面的信息更新第15-50行的设置:

from voice\_engine.kws import KWS

#from voice\_engine.ns import NS

#from voice\_engine.doa\_respeaker\_4mic\_array import DOA

from avs.alexa import Alexa

from pixels import pixels

def main():

logging.basicConfig(level=logging.DEBUG)

src = Source(rate=16000, channels=2, frames\_size=800)

ch1 = ChannelPicker(channels=2, pick=1)

#ns = NS(rate=16000, channels=1)

kws = KWS(model='snowboy')

#doa = DOA(rate=16000)

alexa = Alexa()

alexa.state\_listener.on\_listening = pixels.listen

alexa.state\_listener.on\_thinking = pixels.think

alexa.state\_listener.on\_speaking = pixels.speak

alexa.state\_listener.on\_finished = pixels.off

src.link(ch1)

ch1.link(kws)

#ch1.link(ns)

#ns.link(kws)

kws.link(alexa)

#src.link(doa)

def on\_detected(keyword):

#logging.info('detected {} at direction {}'.format(keyword, doa.get\_direction()))

logging.info('detected {}'.format(keyword))

alexa.listen()

kws.set\_callback(on\_detected)

**step 3. 让我们High起来!**

现在请在虚拟环境下运行 python ns\_kws\_doa\_alexa.py , 我们会在终端看到很多 debug 的消息. 当我们看到 **status code: 204** 的时候, 请说 snowboy 来唤醒 respeaker。接下来 respeaker 上的 led 灯亮起来, 我们可以跟他对话, 比如问，"谁是最帅的?" 或者 "播放刘德华的男人哭吧哭吧不是罪"。小伙伴，尽情的 High 起来吧。

STT (语音转文字)[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#stt)

本部分将介绍百度STT（语音到文本）功能以及GPIO控件。 这是GPIO配置。 如果您没有风扇，可以在GPIO12 / GPIO13上连接2个LED进行演示。

| GPIO | Turn On | Faster | Slower | Turn Off |
| --- | --- | --- | --- | --- |
| GPIO12 | 1 | 0 | 1 | 0 |
| GPIO13 | 0 | 1 | 0 | 0 |

**Step 1. 安装依赖**

sudo apt install mpg123

pip install baidu-aip monotonic pyaudio

**Step 2. 从百度获取key**[**Here**](https://console.bce.baidu.com/ai/?fromai=1#/ai/speech/overview/index)**.**

**Step 3. 下载源码并执行**[**Smart\_Fan.py**](https://github.com/SeeedDocument/MIC_HATv1.0_for_raspberrypi/raw/master/src/baidu_STT/Smart_fan.py)

输入下列命令运行代码

cd ~

wget https://github.com/SeeedDocument/MIC\_HATv1.0\_for\_raspberrypi/raw/master/src/baidu\_STT.zip

unzip baidu\_STT.zip

cd baidu\_STT

python Smart\_Fan.py

**Warning**

请在运行 Smart\_Fan.py之前添加百度密钥 @ line 36,37,38。 您还可以通过运行synthesis\_wav.py来生成所有者的声音。 请在第6,7,8行添加百度密钥，并将字符串修改为您要生成的内容。

**Step 4. 说 '开风扇'.**

**Step 5. 你会看到风扇开启.**

**Step 6. 可以试试 '快一点', '慢一点' 或 '关风扇'.**

FAQ(疑问解答)[¶](http://wiki.seeedstudio.com/cn/ReSpeaker_2_Mics_Pi_HAT/#faq)

**Q1:严格按照本 wiki 操作，驱动还是安装失败，怎么办？**

A1:如果按照上述方法安装驱动均失败，请点击下面固件安装

[我是固件](https://v2.fangcloud.com/share/7395fd138a1cab496fd4792fe5?folder_id=188000207913)

需要以下几点需要注意，第一，lite版本是没有图形界面的精简版,建议您安装有图形界面的。第二，烧了固件后，记得换源。第三， 如果要使用交互功能之前请命令行输入alexa-auth或dueros-auth申请授权，授权成功后会在/home/pi目录下生成.avs.json文件，这时才能使用交互功能。第四，/home/pi目录下会有 respeaker的例程文件夹,可以根据用的mic不同而使用相应的例程。但是请烧录系统后在respeaker目录下更新下例程，可以在respeaker目录下执行git pull origin master命令来更新。

**Q2: #include "portaudio.h" Error when run "sudo pip install pyaudio".**

A2: 命令行输入如下命令

sudo apt-get install portaudio19-dev

**Q3 关于安装snowboy时出现不适合该平台的警告提醒**

A3: 目前snowboy只能兼容python2，所以通过在安装python的虚拟环境时，请确保是python2

**Q4 有时候 sudo python file.py 时候会出现依赖问题**

A4:测试时发现sudo执行时候默认从系统环境执行，而wiki中用到的依赖都是装在~/env 下的，可以通过 sudo ~/env/bin/python file.py来解决

**Q5 可以通过3.5毫米音频插孔的播放来听到声音，但是在运行ns\_kws\_doa\_alexa\_with\_light.py时听不到声音**

A5： 我们有3个播放器（mpv，mpg123和gstreamer）可以使用。 mpg123更适合语音识别和唤醒更，它更具响应性； 而AudioPlayer 更适用gstreamer> mpv> mpg123。 Gstreamer支持更多音频格式，并且在raspberry pi上运行良好。 我们还可以使用环境变量PLAYER指定AudioPlayer的播放器。 所以请尝试以下命令启用语音。

sudo apt install mpg123

PLAYER=mpg123 python ns\_kws\_doa\_alexa\_with\_light.py

**Q6 在运行语音交互时候喊 snowboy 没反应**

A6:请运行audacity以确保4个频道良好。 如果有一个没有数据的频道，当我们说snowboy时就没有回复。