ReSpeaker 2-Mics Pi HAT SKU：107100001

ReSpeaker 2-Mics Pi HAT is a dual-microphone expansion board for Raspberry Pi designed for AI and voice applications. This means that you can build a more powerful and flexible voice product that integrates Amazon Alexa Voice Service, Google Assistant, and so on.

The board is developed based on WM8960, a low power stereo codec. There are 2 microphones on both sides of the board for collecting sounds and it also provides 3 APA102 RGB LEDs, 1 User Button and 2 on-board Grove interfaces for expanding your applications. What is more, 3.5mm Audio Jack or JST 2.0 Speaker Out are both available for audio output.

https://www.youtube.com/embed/MwLEawbP0ZU

Features

- Raspberry Pi compatible(Support Raspberry Pi Zero and Zero W, Raspberry Pi B+, Raspberry Pi 2 B, Raspberry Pi 3 B, Raspberry Pi 3 B+ and Raspberry Pi 3 A+)
- 2 Microphones
- 2 Grove Interfaces
- 1 User Button
- 3.5mm Audio Jack
- JST2.0 Speaker Out
- Max Sample Rate: 48Khz

Specification

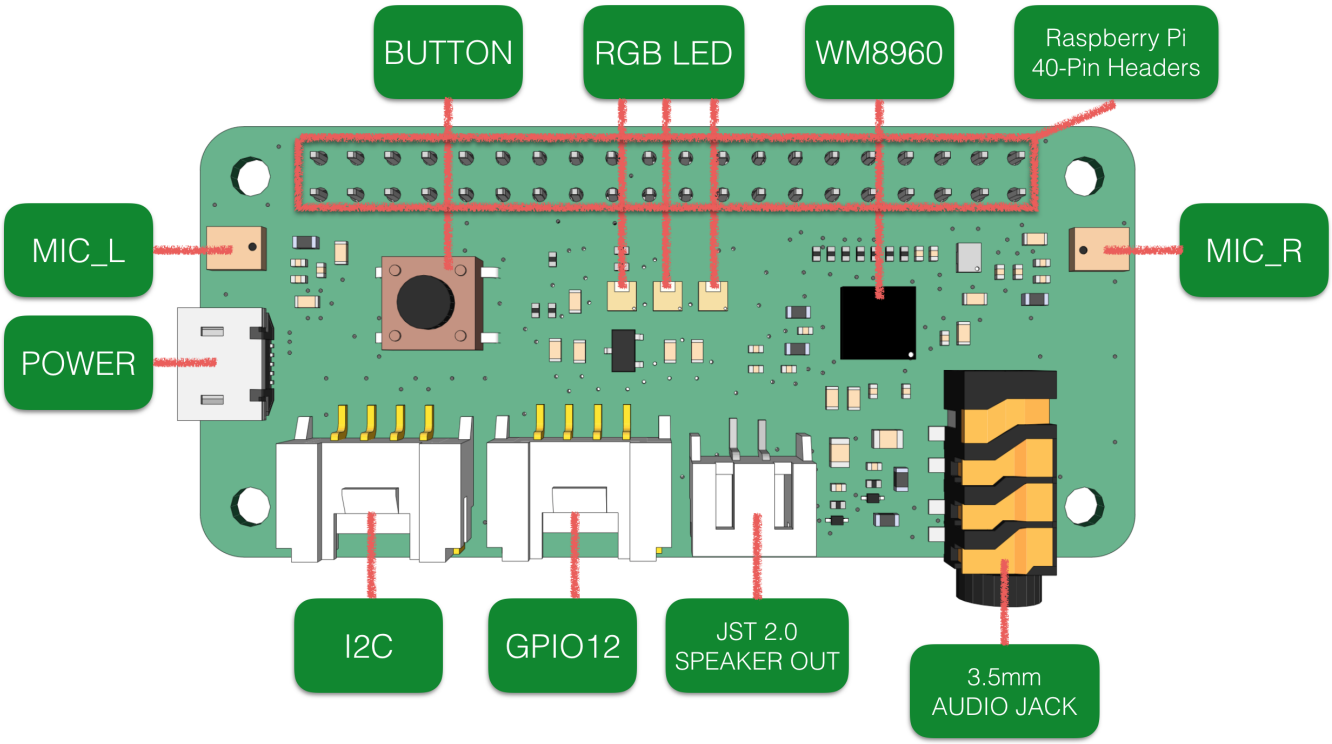
Parameter	Value/Range
Package size	L: 144mm W: 77mm H: 27mm

Parameter	Value/Range
Gross Weight	33g

Application Ideas

- Voice Interaction Application
- AI Assistant

Hardware Overview



- BUTTON: a User Button, connected to GPIO17
- MIC_Land MIC_R: 2 Microphones on both sides of the board
- RGB LED: 3 APA102 RGB LEDs, connected to SPI interface
- WM8960: a low power stereo codec
- Raspberry Pi 40-Pin Headers: support Raspberry Pi Zero, Raspberry Pi 1 B+, Raspberry Pi 2 B , Raspberry Pi 3 B and Raspberry Pi 3 B+
- POWER: Micro USB port for powering the ReSpeaker 2-Mics Pi HAT, please power the board for providing enough current when using the speaker.
- I2C: Grove I2C port, connected to I2C-1
- GPIO12: Grove digital port, connected to GPIO12 & GPIO13
- JST 2.0 SPEAKER OUT: for connecting speaker with JST 2.0 connector
- 3.5mm AUDIO JACK: for connecting headphone or speaker with 3.5mm Audio Plug
- Size: L:64mm W:29mm H:15mm
- Weight: 11.6g

Getting Started

1. Connect ReSpeaker 2-Mics Pi HAT to Raspberry Pi

Mount ReSpeaker 2-Mics Pi HAT on your Raspberry Pi, make sure that the pins are properly aligned when stacking the ReSpeaker 2-Mics Pi HAT.

Raspberry Pi Connection

Raspberry Pi zero Connection

2. Setup the driver on Raspberry Pi

While the upstream wm8960 codec is not currently supported by current Pi kernel builds, upstream wm8960 has some bugs, we had fixed it. We must build it manually.

Make sure that you are running the lastest Raspbian Operating System(debian 9) on your Pi. *(updated at 2018.11.13)*

- Step 1. Get the seeed voice card source code, install and reboot.

```
sudo apt-get update
sudo apt-get upgrade
git clone https://github.com/respeaker/seeed-voicecard.git
cd seeed-voicecard
sudo ./install.sh
reboot
```

- Step 2. Check that the sound card name matches the source code seeed-voicecard by command aplay -l and arecord -l.

```
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 $
```

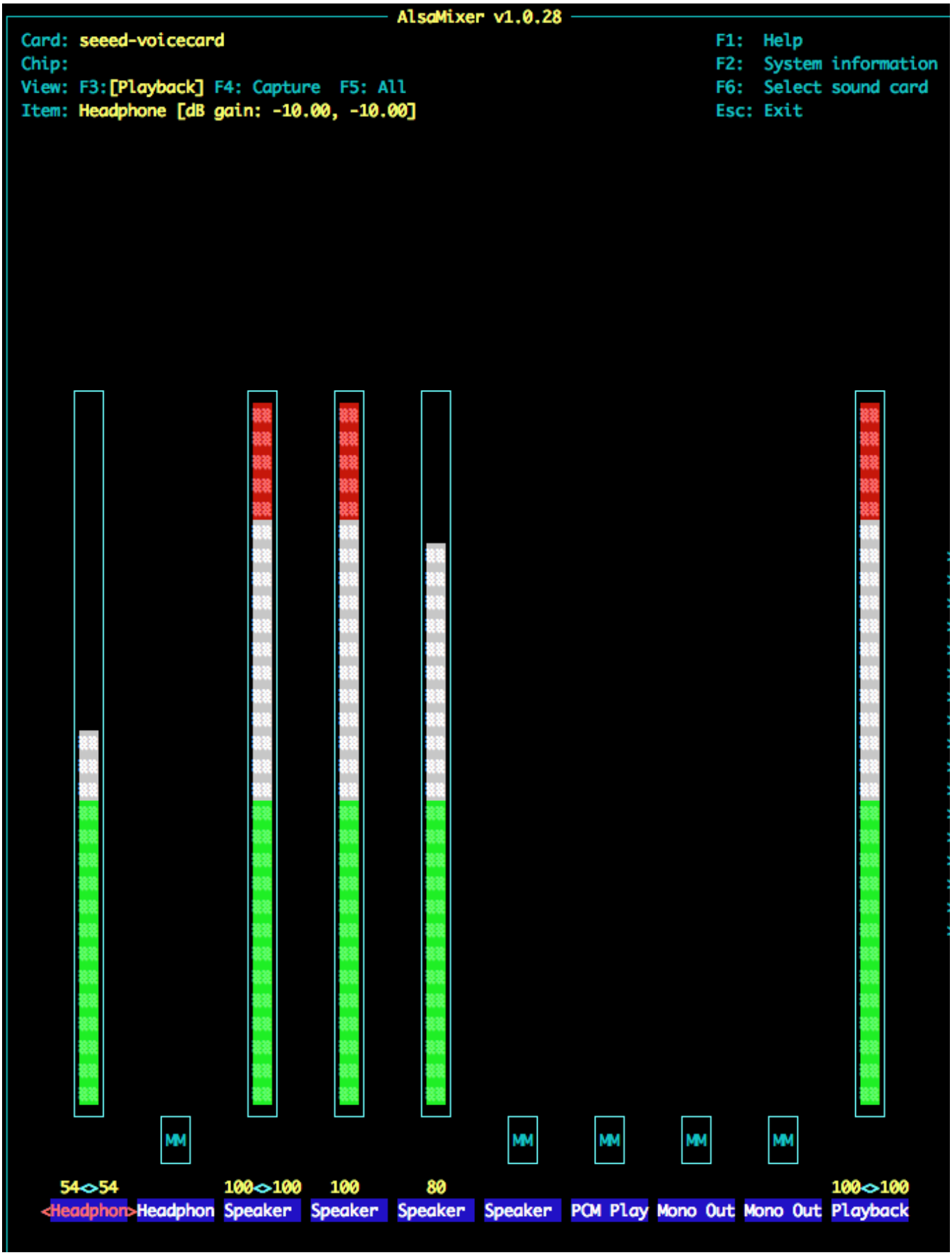
- Step 3. Test, you will hear what you say to the microphones(don't forget to plug in an earphone or a speaker):

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

3. Configure sound settings and adjust the volume with alsamixer

alsamixer is a graphical mixer program for the Advanced Linux Sound Architecture (ALSA) that is used to configure sound settings and adjust the volume.

```
pi@raspberrypi:~ $ alsamixer
```



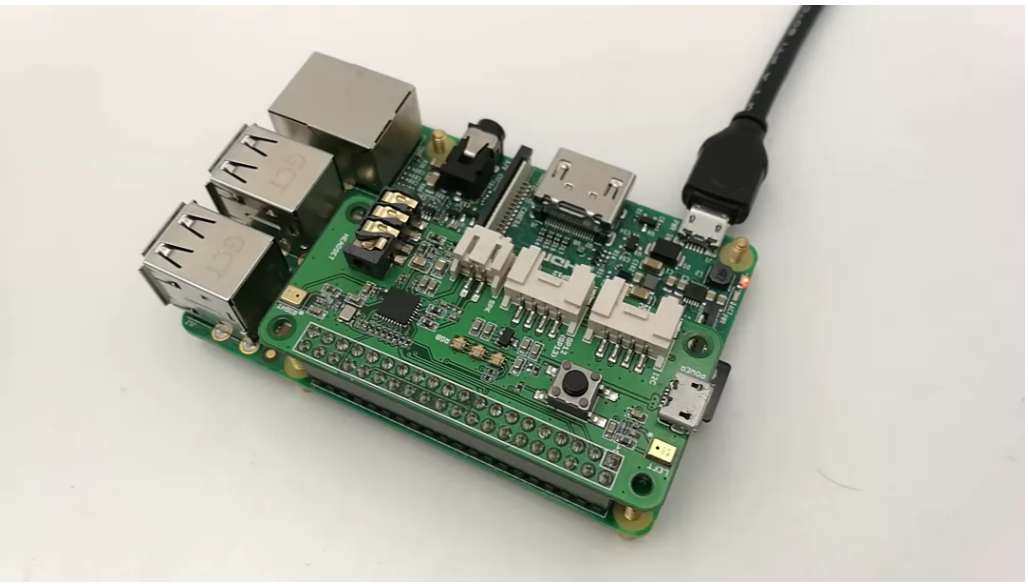
The Left and right arrow keys are used to select the channel or device and the Up and Down Arrows control the volume for the currently selected device. Quit the program with ALT+Q, or by hitting the Esc key. More information

!!!Warning Please use the F6 to select seeed-2mic-voicecard device first.

4. Use the on-board APA102 LEDs

Each on-board APA102 LED has an additional driver chip. The driver chip takes care of receiving the desired color via its input lines, and then holding this color until a new command is received.

```
sudo pip install spidev
cd ~/
git clone https://github.com/respeaker/mic_hat.git
cd mic_hat
python pixels.py
```



5. How to use User Button

There is an on-board User Button, which is connected to GPIO17. Now we will try to detect it with python and RPi.GPIO.

```
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
```

Google Assistant SDK

To get started with Google Assistant(what is Google Assistant), the first is that you should integrate the Google Assistant Library into your raspberry pi system. Here is the link to Google official guidance.

And the following guide will also show you how to get started with Google Assistant.

1. Configure a Developer Project and get JSON file

Follow step 1. 2. 3. 4. in the guide to configure a project on Google Cloud Platform and create an OAuth Client ID JSON file. Don't forget to copy the JSON file to your Raspberry Pi.

2. Use a Python virtual environment to isolate the SDK and its dependencies from the system Python packages.

```
sudo apt-get update
sudo apt-get install python3-dev python3-venv # Use python3.4-venv if the package
cannot be found.
python3 -m venv env
env/bin/python -m pip install --upgrade pip setuptools
source env/bin/activate
```

3. Install google-assistant-library

The Google Assistant SDK package contains all the code required to get the Google Assistant running on the device, including the library and sample code. Use pip to install the latest version of the Python package in the virtual environment:

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

4. Authorize the Google Assistant SDK

Authorize the Google Assistant SDK sample to make Google Assistant queries for the given Google Account. Reference the JSON file you copied over to the device in Step 1.

```
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` should be the path of your JSON file, your should modify the commmand above
- After running the command, it should display as shown below. Copy the URL and paste it into a browser (this can be done on your development machine, or any other machine). After you approve, a code will appear in your browser, such as "4/XXXX". Copy this and paste this code into the terminal.

```
Please go to this URL: https://...
Enter the authorization code:
```

- It should then display: OAuth credentials initialized.
- If instead it displays: InvalidGrantError then an invalid code was entered. Try again, taking care to copy and paste the entire code.

5. Start the Google Assistant demo

```
pi@raspberrypi:~ $ alsamixer    // To adjust the volume
pi@raspberrypi:~ $ source env/bin/activate
(env) pi@raspberrypi:~ $ env/bin/google-assistant-demo
```

6. Play with Google Assistant

Say *Ok Google* or *Hey Google*, followed by your query. The Assistant should respond. If the Assistant does not respond, follow the troubleshooting instructions.

```
(env) pi@raspberrypi:~ $ env/bin/google-assistant-demo
ON_MUTED_CHANGED:
  {'is_muted': False}
ON_START_FINISHED
E0524 07:48:58.791984135    1102 handshake.c:128]           Security hand
shake failed: {"created":"@1495612138.791891844","description":"Handshake
read failed","file":"../../third_party/grpc/src/core/lib/security/transpor
t/handshake.c","file_line":237,"referenced_errors":[{"created":"@149561213
8.791882625","description":"FD shutdown","file":"../../third_party/grpc/sr
c/core/lib/iomgr/ev_epoll_linux.c","file_line":1045}]}

ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
  {'text': "what's the weather today"}
ON_RESPONDING_STARTED:
  {'is_error_response': False}
ON_RESPONDING_FINISHED
ON_CONVERSATION_TURN_FINISHED:
  {'with_follow_on_turn': False}

E0524 07:49:15.295967526    1107 handshake.c:128]           Security hand
shake failed: {"created":"@1495612155.295783620","description":"Handshake
read failed","file":"../../third_party/grpc/src/core/lib/security/transpor
t/handshake.c","file_line":237,"referenced_errors":[{"created":"@149561215
5.295768411","description":"FD shutdown","file":"../../third_party/grpc/sr
c/core/lib/iomgr/ev_epoll_linux.c","file_line":1045}]}
```

7. Troubleshooting

See the Troubleshooting page if you run into issues.

8. Raspbian image

As Raspbian Operating System is updated to Debian 9, we won't provide our Raspbian image anymore. Click here to get the lastest Raspbian Operating System.

- How to install the image

STT(Speech to Text) SDK

This part will introduce Baidu STT(Speech to Text) functions together with GPIO control. Here is the GPIOs configuration. If you do not have a fan, You can connect 2 LEDs on GPIO12/GPIO13 to demonstrate.

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

- **Step 1. Install dependiencies**

```
sudo apt install mpg123
pip install baidu-aip monotonic pyaudio
```

- **Step 2. Get Baidu key from Here.**
- **Step 3. Download the Smart_Fan.py**

```
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 Please add baidu key @ line 36,37,38 before running python Smart_Fan.py. You also can generate your owner voice by running the synthesis_wav.py. Please do add baidu keys at line 6,7,8 and modify string to what you want to generate.

- **Step 4. Let's say '开风扇'.**
- **Step 5. You will see the fan moving.**
- **Step 6. Let's try '快一点', '慢一点' and '关风扇'.**

Extract Voice

We use PyAudio python library to extract voice.

- Step 1, We need to run the following script to get the device index number of 2 Mic pi hat:

```
sudo pip install pyaudio
cd ~
nano get_index.py
```

- Step 2, copy below code and paste on get_index.py.

```
import pyaudio

p = pyaudio.PyAudio()
info = p.get_host_api_info_by_index(0)
numdevices = info.get('deviceCount')

for i in range(0, numdevices):
```

```
        if (p.get_device_info_by_host_api_device_index(0,
i).get('maxInputChannels')) > 0:
            print "Input Device id ", i, " - ",
p.get_device_info_by_host_api_device_index(0, i).get('name')
```

- Step 3, press Ctrl + X to exit and press Y to save.
- Step 4, run 'sudo python get_index.py' and we will see the device ID as below.

```
Input Device id  2  -  seeed-2mic-voicecard: - (hw:1,0)
```

- Step 5, change `RESPEAKER_INDEX = 2` to index number. Run python script record.py to record a speech.

```
import pyaudio
import wave

RESPEAKER_RATE = 16000
RESPEAKER_CHANNELS = 2
RESPEAKER_WIDTH = 2
# run getDeviceInfo.py to get index
RESPEAKER_INDEX = 2  # refer to input device id
CHUNK = 1024
RECORD_SECONDS = 5
WAVE_OUTPUT_FILENAME = "output.wav"

p = pyaudio.PyAudio()

stream = p.open(
            rate=RESPEAKER_RATE,
            format=p.get_format_from_width(RESPEAKER_WIDTH),
            channels=RESPEAKER_CHANNELS,
            input=True,
            input_device_index=RESPEAKER_INDEX,)

print("* recording")

frames = []

for i in range(0, int(RESPEAKER_RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    frames.append(data)

print("* done recording")

stream.stop_stream()
stream.close()
p.terminate()
```

```
wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
wf.setnchannels(RESPEAKER_CHANNELS)
wf.setsampwidth(p.get_sample_size(p.get_format_from_width(RESPEAKER_WIDTH)))
wf.setframerate(RESPEAKER_RATE)
wf.writeframes(b''.join(frames))
wf.close()
```

- Step 6. If you want to extract channel 0 data from 2 channels, please follow below code. For other channel X, please change [0::2] to [X::2].

```
import pyaudio
import wave
import numpy as np

RESPEAKER_RATE = 16000
RESPEAKER_CHANNELS = 2
RESPEAKER_WIDTH = 2
# run getDeviceInfo.py to get index
RESPEAKER_INDEX = 2  # refer to input device id
CHUNK = 1024
RECORD_SECONDS = 3
WAVE_OUTPUT_FILENAME = "output.wav"

p = pyaudio.PyAudio()

stream = p.open(
            rate=RESPEAKER_RATE,
            format=p.get_format_from_width(RESPEAKER_WIDTH),
            channels=RESPEAKER_CHANNELS,
            input=True,
            input_device_index=RESPEAKER_INDEX,)

print("* recording")

frames = []

for i in range(0, int(RESPEAKER_RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    # extract channel 0 data from 2 channels, if you want to extract channel 1,
please change to [1::2]
    a = np.fromstring(data,dtype=np.int16)[0::2]
    frames.append(a.tostring())

print("* done recording")

stream.stop_stream()
stream.close()
p.terminate()

wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
wf.setnchannels(1)
```

```
    wf.setsampwidth(p.get_sample_size(p.get_format_from_width(RESPEAKER_WIDTH)))
    wf.setframerate(RESPEAKER_RATE)
    wf.writeframes(b''.join(frames))
    wf.close()
```

FAQ

Q1: #include "portaudio.h" Error when run "sudo pip install pyaudio".

A1: Please run below command to solve the issue.

```
sudo apt-get install portaudio19-dev
```

Q2: How to change the Raspbian Mirrors source?

A2: Please refer to Raspbian Mirrors and follow below instructions to modify the source at begining.

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

For example, we suggest use the tsinghua source for China users. So please modify the sources.list as below.

```
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
```

Resources

- **[Eagle]** Respeaker_2_Mics_Pi_HAT_SCH
- **[Eagle]** Respeaker_2_Mics_Pi_HAT_PCB
- **[PDF]** Respeaker_2_Mics_Pi_HAT_SCH
- **[PDF]** Respeaker_2_Mics_Pi_HAT_PCB
- **[3D]** ReSpeaker 2 Mics Pi HAT 3D
- **[Driver]** Seeed-Voice Driver
- **[Algorithms]** Algorithms includes DOA, VAD, NS
- **[Voice Engine** Voice Engine project, provides building blocks to create voice enabled objects
- **[Algorithms]** AEC

Projects

Build Your Own Amazon Echo Using a RPI and ReSpeaker HAT: How to build your own Amazon Echo using a Raspberry Pi and ReSpeaker 2-Mics HAT.

https://www.hackster.io/idreams/build-your-own-amazon-echo-using-a-rpi-and-respeaker-hat-7f44a0

Your personal home barista comes to life with this voice-enabled coffee machine: An open-source, private-by-design coffee machine that keeps your favorite coffee and caffeination schedule private.

https://www.youtube.com/embed/4gN1bvl24ZM

Tech Support

Please submit any technical issue into our forum or drop mail to techsupport@seeed.cc.