

# Raspberry Pi 和 ReSpeaker 麦克风 风 **DIY** 智能家庭助理

Phodal Huang

September 8, 2017

## 目录

|  |   |
|--|---|
| 步骤 1: 材料准备 . . . . .   | 3 |
| 步骤 2: 从 <b>GitHub</b> 下载代码 . . . . .                                     | 3 |
| 步骤 3: 测试阵列麦克风 . . . . .  | 3 |
| 步骤 4: 安装 <b>ReSpeaker Python</b> 库 . . . . .                             | 4 |
| 步骤 5: 设置您的 <b>Wio Link</b> . . . . .                                     | 5 |
| 步骤 6: 将 <b>Bing</b> 密钥和 <b>Wio Token</b> 添加到 <b>Python</b> 代码中 . . . . . | 5 |
| 步骤 7: 运行演示 . . . . .   | 8 |

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/respeaker-raspberry-pi-wio-link-build-home-aut>

我们已经说了这么多年的智能家居，但是我们的家还不够聪明。而这一次，它真的来了，Amazon Echo, Google Home, Apple homekit 等等的智能音箱，带来新的智能。

在这里，ReSpeaker 阵列麦克风作为 Raspberry Pi 的“耳朵”，可以听到你的声控命令并发送给 Raspberry Pi。并根据您的指令，Raspberry Pi 将通过 Wi-Fi 控制 Wio Link 进行所需的操作。

### 步骤 1: 材料准备

- Raspberry Pi (带网络连接) \* 1
- ReSpeaker 阵列麦克风带音频插孔 \* 1
- 带有 Raspberry Pi 图片的 SD 卡 \* 1
- Wio Link 与 WS2812 LED 灯条 \* 1
- USB 线 \* 2
- PC 或 Mac

### 步骤 2: 从 GitHub 下载代码

首先，你需要 SSH 到 Raspberry Pi 上，并在 Raspberry Pi 上从我们的 Github 下载 ReSpeaker 阵列麦克风：

```
1 git clone https://github.com/respeaker/mic_array.git
```

### 步骤 3: 测试阵列麦克风

1. 运行 `pixel_ring.py`，其将通过 USB HID 来控制阵列麦克风上的像素环 (pixel ring)。开始运行后，ReSpeaker 麦克风将开始闪烁。

```
1 sudo pip install pyusb
2 sudo python pixel_ring.py
```

如果您不想使用 `sudo` 访问 USB 设备，请将 `udev.rules` 文件添加到 `/etc/udev/rules.d` 中：

```
1 echo 'SUBSYSTEM=="usb", MODE=="0666"' | sudo tee -a
   /etc/udev/rules.d/60-usb.rules
2 sudo udevadm control -R # then re-plug the usb device
```

2. 从阵列麦克风读取 8 声道音频，并评估声音的 DOA。然后阵列麦克风上的 LED，将显示声音的方向。

```
1 sudo apt-get install python-numpy # or pip install
  numpypythonmic_array.py
```

3. 做 VAD，然后评估 DOA。同样的，阵列麦克风上的 LED 也会显示声音的方向。

```
1 sudo pip install webrtcvad
2 python vad_doa.py
```

4. 做 KWS，然后评估 DOA。接着让 snowboy 工作，并运行 `python kws_doa.py`。然后，您应该对阵列麦克风说“alexa”来唤醒它，如果检测到声音，LED 将显示声音的方向。

```
1 git submodule init
2 git submodule update
3 cd snowboy/swig/Python
4 sudo apt-get install python-dev libatlas-base-dev swig #
  requiremetns to compile snowboy
5 echo 'from snowboydetect import *' > __init__.py # create __init__.py for
  a python module
6 cd ../../.. # chang to the root
  directory of the repository
7 ln -s snowboy/swig/Python snowboydetect
8 python kws_doa.py
```

#### 步骤 4: 安装 **ReSpeaker Python** 库

安装 **ReSpeaker Python Library**，以使用 Bing Speech API

```
1 git clone https://github.com/respeaker/respeaker_python_library.git
2 cd respeaker_python_library
3 python setup.py install
```

这时，你需要一个 **Bing** 密钥，可以阅读这篇文章获取它：[Voice Respeaker](#)。

**步骤 5: 设置您的 Wio Link**

将 **WS2812 LED** 条添加到 **Wio Link** 的数字 **0** 端口, 并更新固件。然后单击“**view api**”并获取访问令牌。

如果您是 **Wio Link** 的新手, 请点击此处查看[维基](#)和指南。

**步骤 6: 将 Bing 密钥和 Wio Token 添加到 Python 代码中**

将下面的代码保存在您的 **Raspberry Pi** 中, 不要忘记填写 **Bing** 密钥和 **Wio Token**:

```
1 from respeaker.bing_speech_api import BingSpeechAPI as Bing
2 import wave
3 from mic_array import MicArray
4 import Queue
5 from pixel_ring import pixel_ring
6 import sys
7 import numpy as np
8 import collections
9 from snowboydetect import SnowboyDetect
10 import time
11 import json
12 from urllib import urlencode
13 from urllib2 import Request, urlopen, URLError, HTTPError
14
15 # write your Wio token here
16
17 WIO_TOKEN = "*****"
18
19 # write your Bing key here
20
21 KEY = "*****"
22 bing = Bing(key=KEY)
23
24 RATE = 16000
25 CHANNELS = 8
26 KWS_FRAMES = 10      # ms
27 DOA_FRAMES = 800    # ms
```

```
28
29 detector = SnowboyDetect('snowboy/resources/common.res',
    'snowboy/resources/snowboy.umdl')detector.SetAudioGain(1)
30 detector.SetSensitivity('0.5')
31
32 # about 5seconds
33 q = Queue.Queue(maxsize=768)
34
35 def gen_queue(q):
36     try:
37         data = q.get(timeout=1)
38         while data:
39             yield data
40             data = q.get(timeout=1)
41     except Queue.Empty:
42         pass
43
44 def controlledLED(onoff=0):
45     try:
46         if onoff == 1:
47             rgb_hex_string = '000080'
48         else:
49             rgb_hex_string = '000000'
50         url =
            'https://cn.wio.seeed.io/v1/node/GroveLedWs2812D0/clear/4/{?access_token={}'.format(
                WIO_TOKEN)
51         request = Request(url, data='')
52         response = urlopen(request)
53         data = response.read()
54         result = json.loads(data)
55         if result['result'] == 'ok':
56             return True
57         else:
58             return False
59     except Exception as err:
60         return False
```

```
61
62 def main():    history = collections.deque(maxlen=int(DOA_FRAMES /
    KWS_FRAMES))
63     global q
64
65     try:
66         with MicArray(RATE, CHANNELS, RATE * KWS_FRAMES / 1000) as mic:
67             for chunk in mic.read_chunks():
68                 history.append(chunk)
69                 # Detect keyword from channel 0
70                 ans = detector.RunDetection(chunk[0:CHANNELS].tostring())
71                 if ans > 0:
72                     print("wake up")
73                     print("start recording")
74                     pixel_ring.arc(12)
75                     q.queue.clear()
76                     for chunk in mic.read_chunks():
77                         q.put(chunk[0:CHANNELS].tostring())
78                         if q.full():
79                             break
80                     print "queue full"
81                     pixel_ring.spin()
82                     text = bing.recognize(gen_queue(q))    # data can be
                        generator
83                     if text:
84                         print('{}'.format(text))
85                         if 'turn on' in text:
86                             controlled(1)
87                         if 'turn off' in text:
88                             controlled(0)
89                     pixel_ring.off()
90
91     except KeyboardInterrupt:
92         pass
93
94     pixel_ring.off()
```

```
95     # except ValueError:
96 #     pass
97 if __name__ == '__main__':
98     main()
```

### 步骤 7: 运行演示

将步骤 6 中的 `python` 代码保存在 `smart_home.py` 中，然后运行它：

```
1 python smart_home.py
```

唤醒它说“snowboy”，然后说“please turn on the light”，或者“please turn off the light”来控制 Wio Link。

Enjoy it!!

原文链接：<http://www.instructables.com/id/Diy-Smart-Home-Assistant-With-Raspberry-Pi-and-Re>

原文链接：<https://www.wandianshenme.com/play/respeaker-raspberry-pi-wio-link-build-home-aut>