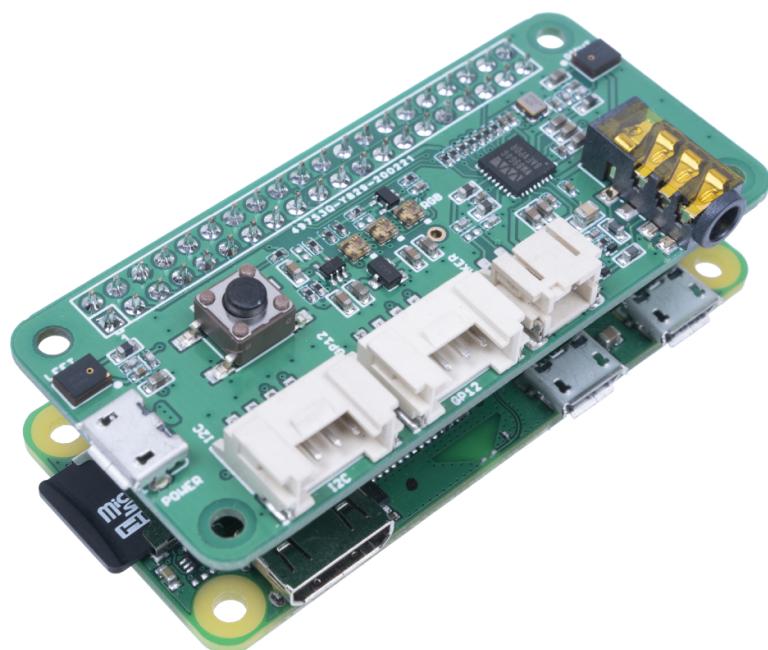


ReSpeaker 2-Mics Pi HAT



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.

Snips Voice Interaction Base Kit Demo



Get One Now A green rectangular button with white text and a white shopping cart icon.

[<https://www.seeedstudio.com/ReSpeaker-2-Mics-Pi-HAT-p-2874.html>]

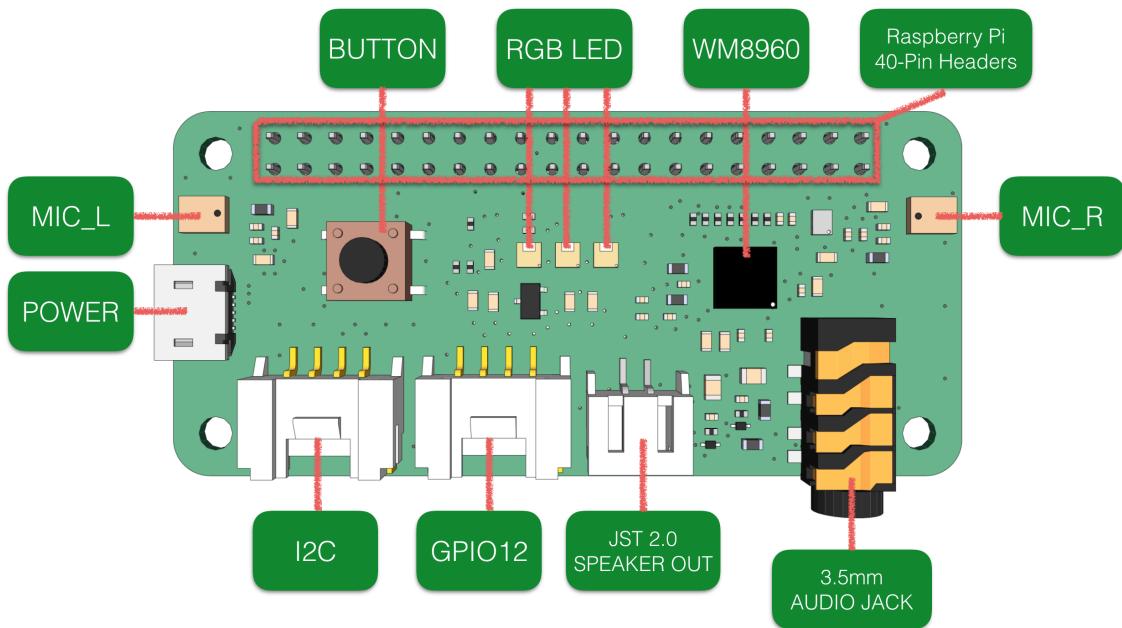
Features

- 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+, Raspberry Pi 3 A+ and Raspberry Pi 4)
- 2 Microphones
- 2 Grove Interfaces
- 1 User Button
- 3.5mm Audio Jack
- JST2.0 Speaker Out
- Max Sample Rate: 48Khz

Application Ideas

- Voice Interaction Application
- AI Assistant

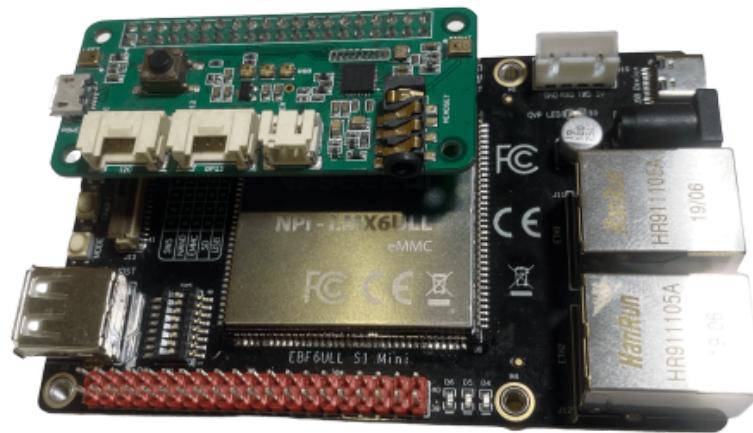
Hardware Overview



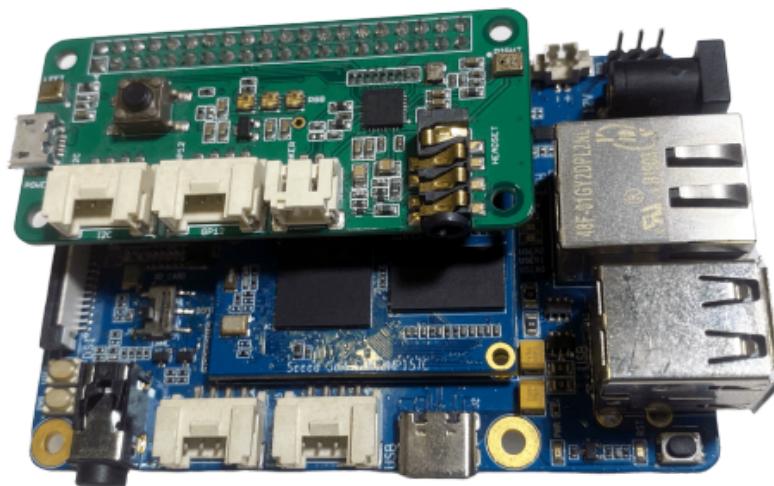
- BUTTON: a User Button, connected to GPIO17
- MIC_L and MIC_R: 2 Microphones on both sides of the board
- RGB LED: 3 APA102 RGB LEDs, connected to SPI interface
- WM8960: a low power stereo codec
- Raspberry Pi 40-Pin Headers: support Raspberry Pi Zero, Raspberry Pi 1 B+, Raspberry Pi 2 B , Raspberry 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

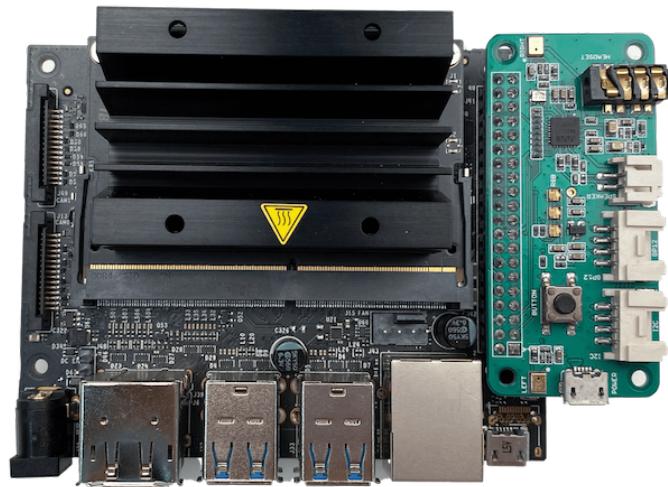
Platform Supported



- **Get started with NPI i.MX6ULL Dev Board Linux SBC**
[<https://wiki.seeedstudio.com/NPi-i.MX6ULL-Dev-Board-Linux-SBC/#iis>]



- **Get started with ODYSSEY – STM32MP157C**
[<https://wiki.seeedstudio.com/ODYSSEY-STM32MP157C/#i2s-on-odyssey-stm32mp157c>]



- **Get started with Nvidia Jetson Nano Series**
[https://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT/#get-started-with-nvidia-jetson-nano-series]

Getting Started

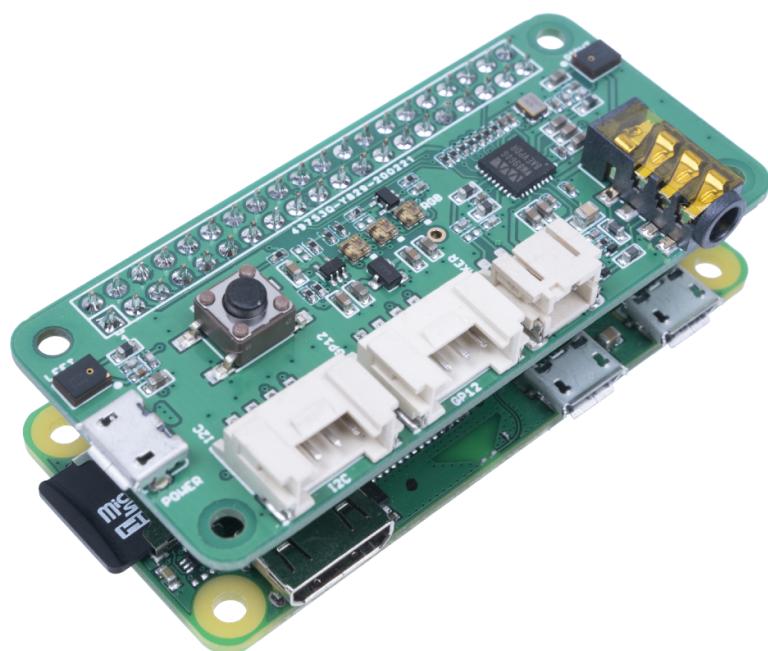
1. Connect ReSpeaker 2-Mics Pi HAT to Raspberry Pi

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)** [<https://www.raspberrypi.org/downloads/raspbian/>] on your Pi. (*updated at 2018.11.13*)

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

```

1 sudo apt-get update
2 sudo apt-get upgrade
3 git clone https://github.com/respeaker/seeed-voicecard.git
4 cd seeed-voicecard
5 sudo ./install.sh
6 reboot

```

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

```

1 pi@raspberrypi:~/seeed-voicecard $ aplay -l
2 **** List of PLAYBACK Hardware Devices ****
3 card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 AI
4     Subdevices: 8/8
5     Subdevice #0: subdevice #0
6     Subdevice #1: subdevice #1
7     Subdevice #2: subdevice #2
8     Subdevice #3: subdevice #3
9     Subdevice #4: subdevice #4
10    Subdevice #5: subdevice #5
11    Subdevice #6: subdevice #6
12    Subdevice #7: subdevice #7
13 card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IFF
14     Subdevices: 1/1
15     Subdevice #0: subdevice #0
16 card 1: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835
17     Subdevices: 1/1
18     Subdevice #0: subdevice #0
19
20 pi@raspberrypi:~/seeed-voicecard $ arecord -l
21 **** List of CAPTURE Hardware Devices ****
22 card 1: seeed2micvoicec [seeed-2mic-voicecard], device 0: bcm2835
23     Subdevices: 1/1
24     Subdevice #0: subdevice #0

```

```
25 pi@raspberrypi:~/seeed-voicecard $
```

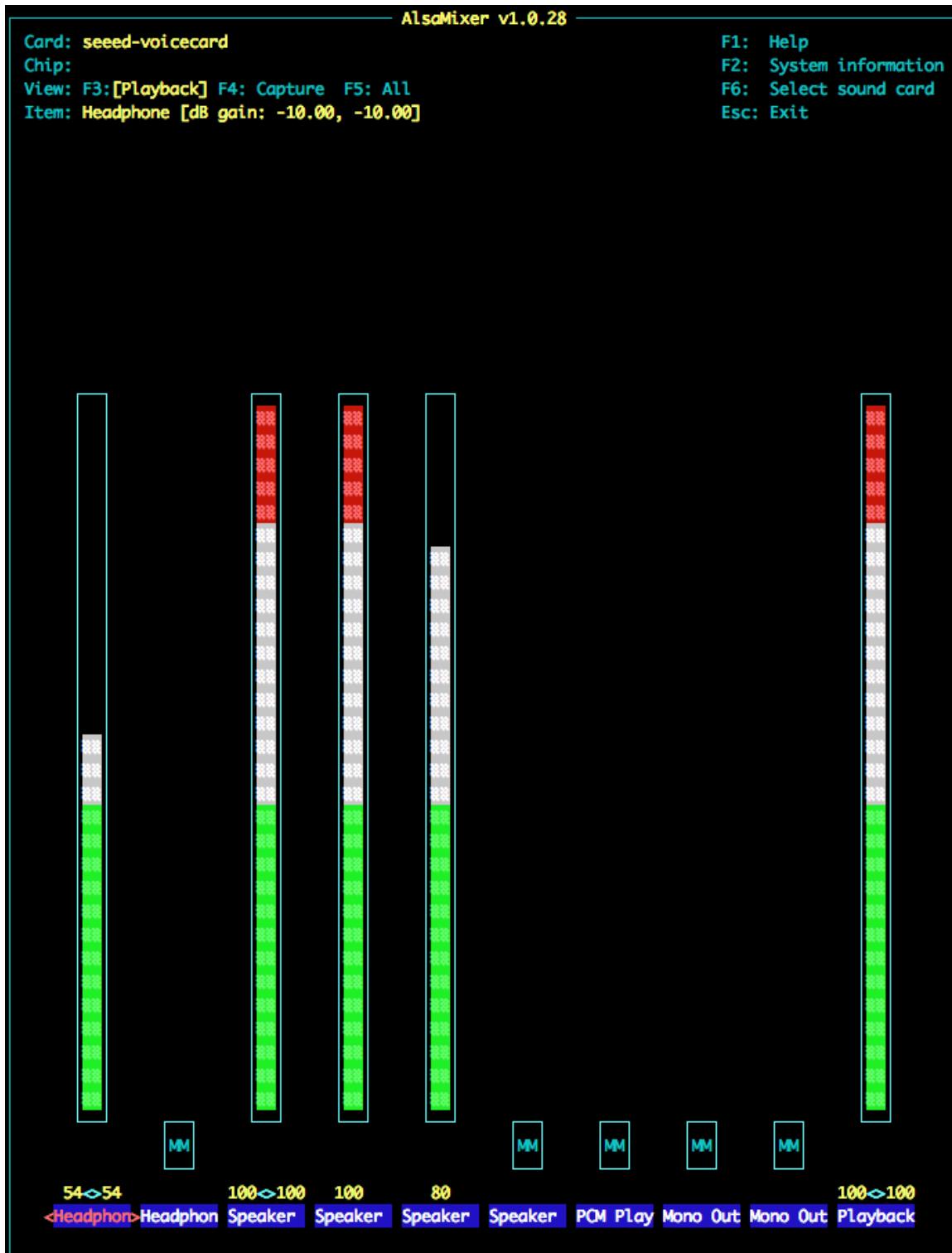
- Step 3. Test, you will hear what you say to the microphones(don't forget to plug in an earphone or a speaker):

```
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](https://en.wikipedia.org/wiki/Alsamixer) [<https://en.wikipedia.org/wiki/Alsamixer>]



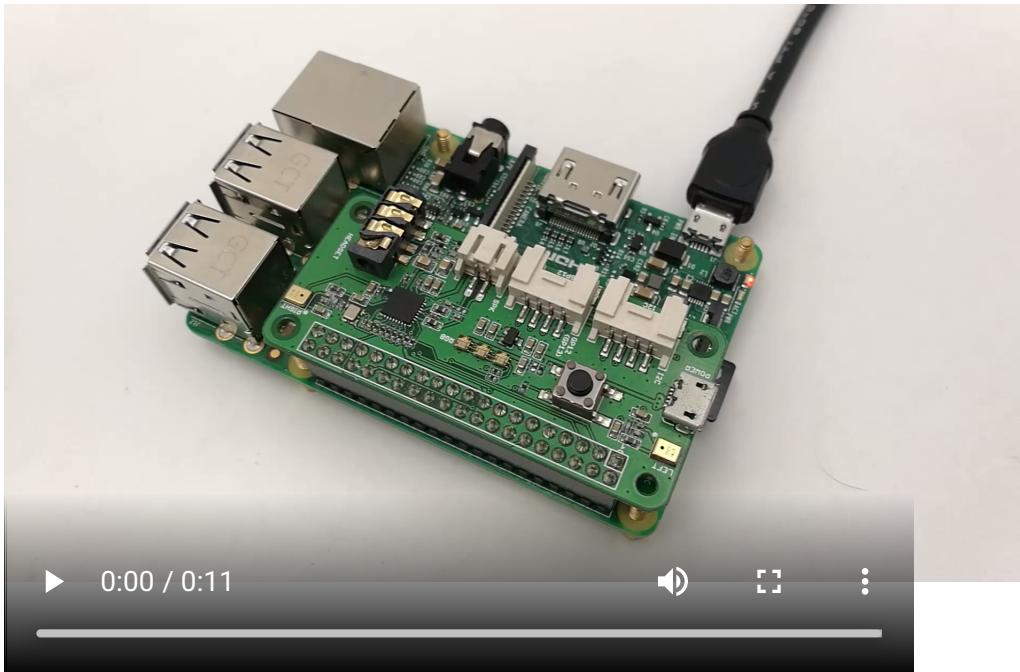
Warning

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.

```
1 sudo pip install spidev
2 cd ~/
3 git clone https://github.com/respeaker/mic_hat.git
4 cd mic_hat
5 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.

```
1 sudo pip install rpi.gpio      // install RPi.GPIO library
2 nano button.py                // copy the following code in button.py
```

```

1 import RPi.GPIO as GPIO
2 import time
3
4 BUTTON = 17
5
6 GPIO.setmode(GPIO.BCM)
7 GPIO.setup(BUTTON, GPIO.IN)
8
9 while True:
10     state = GPIO.input(BUTTON)
11     if state:
12         print("off")
13     else:
14         print("on")
15     time.sleep(1)

```

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

```

1 pi@raspberrypi:~ $ python button.py
2 off
3 off
4 on
5 on
6 off

```

Extract Voice

We use [PyAudio python library](#) [<https://people.csail.mit.edu/hubert/pyaudio/>] to extract voice.

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

```

1 sudo pip install pyaudio
2 cd ~
3 nano get_index.py

```

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

```

1 import pyaudio
2
3 p = pyaudio.PyAudio()
4 info = p.get_host_api_info_by_index(0)
5 numdevices = info.get('deviceCount')
6
7 for i in range(0, numdevices):
8     if (p.get_device_info_by_host_api_device_index(0, i).get
9         print "Input Device id ", i, " - ", p.get_device_inf

```

- 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.

```

1 import pyaudio
2 import wave
3
4 RESPEAKER_RATE = 16000
5 RESPEAKER_CHANNELS = 2
6 RESPEAKER_WIDTH = 2
7 # run getDeviceInfo.py to get index
8 RESPEAKER_INDEX = 2 # refer to input device id
9 CHUNK = 1024
10 RECORD_SECONDS = 5
11 WAVE_OUTPUT_FILENAME = "output.wav"
12
13 p = pyaudio.PyAudio()
14
15 stream = p.open(
16             rate=RESPEAKER_RATE,
17             format=p.get_format_from_width(RESPEAKER_WIDTH),
18             channels=RESPEAKER_CHANNELS,
19             input=True,
20             input_device_index=RESPEAKER_INDEX, )
21
22 print("* recording")
23
24 frames = []
25

```

```

26 for i in range(0, int(RESPEAKER_RATE / CHUNK * RECORD_SECONDS)):
27     data = stream.read(CHUNK)
28     frames.append(data)
29
30 print("* done recording")
31
32 stream.stop_stream()
33 stream.close()
34 p.terminate()
35
36 wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
37 wf.setnchannels(RESPEAKER_CHANNELS)
38 wf.setsampwidth(p.get_sample_size(p.get_format_from_width(RESPEAKER_WIDTH)))
39 wf.setframerate(RESPEAKER_RATE)
40 wf.writeframes(b''.join(frames))
41 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].

```

1 import pyaudio
2 import wave
3 import numpy as np
4
5 RESPEAKER_RATE = 16000
6 RESPEAKER_CHANNELS = 2
7 RESPEAKER_WIDTH = 2
8 # run getDeviceInfo.py to get index
9 RESPEAKER_INDEX = 2 # refer to input device id
10 CHUNK = 1024
11 RECORD_SECONDS = 3
12 WAVE_OUTPUT_FILENAME = "output.wav"
13
14 p = pyaudio.PyAudio()
15
16 stream = p.open(
17         rate=RESPEAKER_RATE,
18         format=p.get_format_from_width(RESPEAKER_WIDTH),
19         channels=RESPEAKER_CHANNELS,
20         input=True,
21         input_device_index=RESPEAKER_INDEX,
22 )
23 print("* recording")
24
25 frames = []
26

```

```

27  for i in range(0, int(RESPEAKER_RATE / CHUNK * RECORD_SECONDS)):
28      data = stream.read(CHUNK)
29      # extract channel 0 data from 2 channels, if you want to extract
30      # channel 1, change [0::2] to [1::2]
31      a = np.fromstring(data,dtype=np.int16)[0::2]
32      frames.append(a.tostring())
33
34  print("* done recording")
35
36  stream.stop_stream()
37  stream.close()
38  p.terminate()
39
40  wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
41  wf.setnchannels(1)
42  wf.setsampwidth(p.get_sample_size(p.get_format_from_width(RESPI
43  wf.writeframes(b''.join(frames))
44  wf.close()

```

Get Started with Nvidia Jetson Nano Series

The wm8960/Respeaker-2-Mic-Hat driver now works on the Jetson Nano platform, here follows the testing steps:

Install WM8960 from seeed-linux-dtoverlays



Note

This is tested for Jetson source R32.4.2 or JetPack Image 4.4.

STEP 1. Clone the repo

```

1 cd ~
2 git clone https://github.com/Seeed-Studio/seeed-linux-dtoverlays
3 cd ~/seeed-linux-dtoverlays

```

STEP 2. Build dtbo & driver

```

1 export CUSTOM_MOD_FILTER_OUT="lis3lv02d mcp25xxfd gt9xx seeed-v
2 KBUILD=/usr/src/linux-headers-4.9.140-tegra-ubuntu18.04_aarch64/

```

STEP 3. Install the Driver

```
sudo -E KBUILD=/usr/src/linux-headers-4.9.140-tegra-ubuntu18.04_aal
```

STEP 4. Install dtbo

```
1 sudo cp overlays/jetsonnano/jetson-seeed-2mic-wm8960.dtbo /boot  
2 sudo /opt/nvidia/jetson-io/config-by-hardware.py -n "Seeed Voice"
```

STEP 5. Reboot

```
sudo reboot
```

STEP 6. Restore Alsa widgets settings



Note

Must wait a moment the time sound card busy after login.

```
1 cd ~/seeed-linux-dtoverlays  
2 alsactl -f extras/wm8960_asound.state-jetson-nano restore 1
```

STEP 7. Capture & Playback

```
arecord -D hw:1,0 -f S32_LE -r 48000 -c 2 | aplay -D hw:1,0 -f S32_L
```

Enabling Voice Recognition at the Edge with
Picovoice



Picovoice [<https://picovoice.ai/>] **enables enterprises to innovate and differentiate rapidly with private voice AI.** Build a unified AI strategy around your brand and products with our speech recognition and **Natural-language understanding (NLU) technologies** [<https://searchenterpriseai.techtarget.com/definition/natural-language-understanding-NLU>].

Seeed has partnered with Picovoice to bring Speech Recognition solution at the edge using ReSpeaker 2-Mic Pi HAT [<https://www.seeedstudio.com/ReSpeaker-2-Mics-Pi-HAT-p-2874.html>] **for developers.**

Picovoice is an end-to-end platform for building voice products on your terms. It enables creating voice experiences similar to Alexa and Google. But it entirely runs 100% on-device. There are advantages of Picovoice:

- **Private:** Everything is processed offline. Intrinsically HIPAA and GDPR compliant.
- **Reliable:** Runs without needing constant connectivity.
- **Zero Latency:** Edge-first architecture eliminates unpredictable network delay.
- **Accurate:** Resilient to noise and reverberation. It outperforms cloud-based alternatives by wide margins.
- **Cross-Platform:** Design once, deploy anywhere. Build using familiar languages and frameworks.

Picovoice with ReSpeaker 2-Mic Pi HAT and Raspberry Pi Getting Started

Step 1. Follow the **above step-by-step tutorial of ReSpeaker 2-Mic Pi HAT with Raspberry Pi** before the following.

Note: Please make sure that the `APA102` LEDs are working properly on the ReSpeaker 2-Mic Pi HAT with Raspberry Pi.

Step 2. Type the following command on the terminal to **install the Picovoice demo for ReSpeaker 2-Mic Pi HAT**.

```
$ sudo pip3 install pvrespeakedemo
```

Note: Please make sure you have `pip3` installed on your Raspberry Pi

Demo Usage

The demo utilises the ReSpeaker 2-Mic Pi HAT on a Raspberry Pi with Picovoice technology to control the LEDs. **This demo is triggered by the wake word "Picovoice" and will be ready to take follow-on actions, such as turning LEDs on and off, and changing LED colors.**

After the installation is finished, type this command to run the demo on the terminal:

```
$ picovoice_respeaker_demo
```

Voice Commands

Here are voice commands for this demo:

- **Picovoice**

The demo outputs:

```
wake word
```

- **Turn on the lights**

You should see the lights turned on and the following message on the terminal:

```
1 {
```

```

2     is_understood : 'true',
3     intent : 'turnLights',
4     slots : {
5         'state' : 'on',
6     }
7 }
```

The list of commands are shown on the terminal:

```

1 context:
2   expressions:
3     turnLights:
4       - "[switch, turn] $state:state (all) (the) [light, lights]"
5       - "[switch, turn] (all) (the) [light, lights] $state:stat
6     changeColor:
7       - "[change, set, switch] (all) (the) (light, lights) (co]
8   slots:
9     state:
10    - "off"
11    - "on"
12     color:
13    - "blue"
14    - "green"
15    - "orange"
16    - "pink"
17    - "purple"
18    - "red"
19    - "white"
20    - "yellow"
```

also, you can try this command to change the colour by:

- **Picovoice, set the lights to orange**

Turn off the lights by:

- **Picovoice, turn off all lights**

Demo Video Demonstration

Offline Voice Assistant on Raspberry Pi



Demo Source Code

The demo is built with the **Picovoice SDK**

[<https://github.com/Picovoice/picovoice>]. The demo source code is available on GitHub at <https://github.com/Picovoice/picovoice/tree/master/demo/respeaker> [<https://github.com/Picovoice/picovoice/tree/master/demo/respeaker>].

Different Wake Words

The **Picovoice SDK** [<https://github.com/Picovoice/picovoice>] includes free sample wake words licensed under Apache 2.0, including major voice assistants (e.g. "Hey Google", "Alexa") and fun ones like "Computer" and "Jarvis".

Custom Voice Commands

The lighting commands are defined by a Picovoice *Speech-to-Intent context*. You can design and train contexts by typing in the allowed grammar using Picovoice Console. You can test your changes in-browser as you edit with the microphone button. Go to Picovoice Console (<https://picovoice.ai/console/> [https://picovoice.ai/console/]) and sign up for an account. Use the **Rhino Speech-to-Intent editor** to make contexts, then train them for Raspberry Pi.



ReSpeaker



Multiple Wake Word Examples

Offline Wake Word & Voice Commands on Raspberry Pi



To demonstrate the Picovoice's capability we have also prepared a multi wake word examples using ReSpeaker 2-Mic Pi HAT with Raspberry Pi! Different wake word can set to execute certain tasks.

This package contains a commandline demo for controlling ReSpeaker 2-Mic Pi HAT LEDs using Porcupine.

Porcupine

Porcupine is a highly-accurate and lightweight wake word engine. It enables building always-listening voice-enabled applications. It is

- using deep neural networks trained in real-world environments.
 - compact and computationally-efficient. It is perfect for IoT.
 - cross-platform. Raspberry Pi, BeagleBone, Android, iOS, Linux (x86_64), macOS (x86_64), Windows (x86_64), and web browsers are supported.
- Additionally, enterprise customers have access to the ARM Cortex-M SDK.

- scalable. It can detect multiple always-listening voice commands with no added runtime footprint.
- self-service. Developers can train custom wake word models using [Picovoice Console](https://picovoice.ai/console/) [<https://picovoice.ai/console/>].

Multi Wake Word Getting Started

Running the following command in terminal to install demo driver:

```
$ sudo pip3 install ppnrespeakerdemo
```

Multi Wake Word Usage

Run the following in terminal after the driver installation:

```
$ porcupine_respeaker_demo
```

Wait for the demo to initialize and print [Listening] in the terminal. Say:

Picovoice

The demo outputs:

```
detected 'Picovoice'
```

The lights are now set to green. Say:

Alexa

The lights are set to yellow now. Say:

Terminator

to turn off the lights.

Wake Word to Colors

Below are the colors associated with supported wake words for this demo:

-  Alexa

- Bumblebee
- Computer
- Hey Google
- Hey Siri
- Jarvis
- Picovoice
- Porcupine
- Terminator

Multiple Wake Word Example Source Code

Please see the complete source code of this example here:

<https://github.com/Picovoice/porcupine/tree/master/demo/respeaker>
[<https://github.com/Picovoice/porcupine/tree/master/demo/respeaker>].

Picovoice with ReSpeaker 2-Mic Pi HAT and Raspberry Pi Zero Getting Started

Step 1. Follow the **above step-by-step tutorial of ReSpeaker 2-Mic Pi HAT with Raspberry Pi Zero** before the following.

Note: Please make sure that the `APA102` LEDs are working properly on the ReSpeaker 2-Mic Pi HAT with Raspberry Pi Zero.

Step 2. Install the `wiringpi` library by typing the following on the terminal.

```
$ sudo apt-get install wiringpi
```

Step 3. From the root of the repository, type the following command on the terminal to **install the Picovoice demo for ReSpeaker 2-Mic Pi HAT**.

```
1 $ gcc -std=c99 -O3 -o demo/respeaker-rpi0/picovoice_demo_mic \
2 -I sdk/c/include/ demo/respeaker-rpi0/picovoice_demo_mic.c \
3 -ldl -lasound -lwiringPi
```

Demo Usage

The demo utilises the ReSpeaker 2-Mic Pi HAT on a Raspberry Pi Zero with Picovoice technology to control the LEDs. **This demo is triggered by the wake word "Picovoice" and will be ready to take follow-on actions, such as turning LEDs on and off, and changing LED colors.**

After the installation is finished, type this command from the root of the repository, to run the demo on the terminal:

```
1 $ ./demo/respeaker-rpi0/picovoice_demo_mic \
2 sdk/c/lib/raspberry-pi/arm11/libpicovoice.so \
3 resources/porcupine/lib/common/porcupine_params.pv \
4 resources/porcupine/resources/keyword_files/raspberry-pi/picovoj
5 0.65 \
6 resources/rhino/lib/common/rhino_params.pv \
7 demo/respeaker/pvrespeakerdemo/respeaker_raspberry-pi.rhn \
8 0.5 \
9 plughw:CARD=seeed2micvoicec,DEV=0
```

Voice Commands

Here are voice commands for this demo:

- **Picovoice**

The demo outputs:

```
wake word
```

- **Turn on the lights**

You should see the lights turned on and the following message on the terminal:

```
1 {
2     is_understood : 'true',
3     intent : 'turnLights',
4     slots : {
5         'state' : 'on',
6     }
```

```
7 }
```

The list of commands are shown on the terminal:

```
1 context:
2   expressions:
3     turnLights:
4       - "[switch, turn] $state:state (all) (the) [light, lights]"
5       - "[switch, turn] (all) (the) [light, lights] $state:stat"
6     changeColor:
7       - "[change, set, switch] (all) (the) (light, lights) (co]"
8   slots:
9     state:
10      - "off"
11      - "on"
12     color:
13      - "blue"
14      - "green"
15      - "orange"
16      - "pink"
17      - "purple"
18      - "red"
19      - "white"
20      - "yellow"
```

also, you can try this command to change the colour by:

- **Picovoice, set the lights to orange**

Turn off the lights by:

- **Picovoice, turn off all lights**

Demo Video Demonstration

End-to-End Voice Recognition on Raspber...



Demo Source Code

The demo is built with the **Picovoice SDK**

[<https://github.com/Picovoice/picovoice>]. The demo source code is available on GitHub at <https://github.com/Picovoice/picovoice/tree/master/demo/respeaker-rpi0> [<https://github.com/Picovoice/picovoice/tree/master/demo/respeaker-rpi0>].

Different Wake Words

The **Picovoice SDK** [<https://github.com/Picovoice/picovoice>] includes free sample wake words licensed under Apache 2.0, including major voice assistants (e.g. "Hey Google", "Alexa") and fun ones like "Computer" and "Jarvis".

Custom Voice Commands

The lighting commands are defined by a Picovoice *Speech-to-Intent context*. You can design and train contexts by typing in the allowed grammar using Picovoice Console. You can test your changes in-browser as you edit with the microphone button. Go to Picovoice Console (<https://picovoice.ai/console/> [https://picovoice.ai/console/]) and sign up for an account. Use the **Rhino Speech-to-Intent editor** to make contexts, then train them for Raspberry Pi Zero.



ReSpeaker



Multiple Wake Word Examples

Offline Voice Commands on Raspberry Pi ...



To demonstrate the Picovoice's capability we have also prepared a multi wake word examples using ReSpeaker 2-Mic Pi HAT with Raspberry Pi Zero! Different wake word can set to execute certain tasks.

This package contains a commandline demo for controlling ReSpeaker 2-Mic Pi HAT LEDs using Porcupine.

Porcupine

Porcupine is a highly-accurate and lightweight wake word engine. It enables building always-listening voice-enabled applications. It is

- using deep neural networks trained in real-world environments.
 - compact and computationally-efficient. It is perfect for IoT.
 - cross-platform. Raspberry Pi, BeagleBone, Android, iOS, Linux (x86_64), macOS (x86_64), Windows (x86_64), and web browsers are supported.
- Additionally, enterprise customers have access to the ARM Cortex-M SDK.

- scalable. It can detect multiple always-listening voice commands with no added runtime footprint.
- self-service. Developers can train custom wake word models using [Picovoice Console](https://picovoice.ai/console/) [<https://picovoice.ai/console/>].

Multi Wake Word Getting Started

Step 1 Install the `wiringpi` library (if you haven't installed already) by typing the following on the terminal.

```
$ sudo apt-get install wiringpi
```

Step 2. From the root of the repository, type the following command on the terminal to **install the Porcupine demo for ReSpeaker 2-Mic Pi HAT**.

```
1 $ gcc -std=c99 -O3 -o demo/respeaker-rpi0/porcupine_demo_mic \
2 -I include/ demo/respeaker-rpi0/porcupine_demo_mic.c \
3 -ldl -lasound -lwiringPi
```

Multi Wake Word Usage

Run the following in terminal from the root of the repository:

```
1 $ ./demo/respeaker-rpi0/porcupine_demo_mic \
2 lib/raspberry-pi/arm11/libpv_porcupine.so \
3 lib/common/porcupine_params.pv \
4 0.65 \
5 plughw:CARD=seeed2micvoicec,DEV=0 \
6 resources/keyword_files/raspberry-pi/alexa_raspberry-pi.ppn \
7 resources/keyword_files/raspberry-pi/computer_raspberry-pi.ppn \
8 resources/keyword_files/raspberry-pi/hey\ google_raspberry-pi.ppn \
9 resources/keyword_files/raspberry-pi/hey\ siri_raspberry-pi.ppn \
10 resources/keyword_files/raspberry-pi/jarvis_raspberry-pi.ppn \
11 resources/keyword_files/raspberry-pi/picovoice_raspberry-pi.ppn \
12 resources/keyword_files/raspberry-pi/porcupine_raspberry-pi.ppn \
13 resources/keyword_files/raspberry-pi/bumblebee_raspberry-pi.ppn \
14 resources/keyword_files/raspberry-pi/terminator_raspberry-pi.ppn
```

Wait for the demo to initialize and print `[Listening]` in the terminal. Say:

Picovoice

The demo outputs:

```
detected 'Picovoice'
```



The lights are now set to `green`. Say:

Alexa

The lights are set to `yellow` now. Say:

Terminator

to turn off the lights.

Wake Word to Colors

Below are the colors associated with supported wake words for this demo:

- `Alexa`
- `Bumblebee`
- `Computer`
- `Hey Google`
- `Hey Siri`
- `Jarvis`
- `Picovoice`
- `Porcupine`
- `Terminator`

Multiple Wake Word Example Source Code

Please see the complete source code of this example here:

<https://github.com/Picovoice/porcupine/tree/master/demo/respeaker-rpi0>
[<https://github.com/Picovoice/porcupine/tree/master/demo/respeaker-rpi0>].

Picovoice Tech Support

If you encounter technical problems using Picovoice, please visit **Picovoice** [<https://github.com/Picovoice>] for discussions.

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** [<http://www.raspbian.org/RaspbianMirrors>] 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.

```
1 deb http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ stret
2 deb-src http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ s
```



Schematic Online Viewer

Resources

- **[Eagle] Respeaker_2_Mics_Pi_HAT_SCH**
[https://files.seeedstudio.com/wiki/MIC_HATv1.0_for_raspberrypi/src/ReSpeaker%202-Mics%20Pi%20HAT_SCH.zip]
- **[Eagle] Respeaker_2_Mics_Pi_HAT_PCB**
[https://files.seeedstudio.com/wiki/MIC_HATv1.0_for_raspberrypi/src/ReSpeaker%202-Mics%20Pi%20HAT_PCB.zip]
- **[PDF] Respeaker_2_Mics_Pi_HAT_SCH**
[https://files.seeedstudio.com/wiki/MIC_HATv1.0_for_raspberrypi/src/ReSpeaker%202-Mics%20Pi%20HAT_SCH.pdf]
- **[PDF] Respeaker_2_Mics_Pi_HAT_PCB**
[https://files.seeedstudio.com/wiki/MIC_HATv1.0_for_raspberrypi/src/ReSpeaker%202-Mics%20Pi%20HAT_PCB.pdf]

- **[3D] ReSpeaker 2 Mics Pi HAT 3D** [https://files.seeedstudio.com/wiki/MIC_HATv1.0_for_raspberrypi/src/ReSpeaker%202-Mics%20Pi%20HAT.zip]
- **[Driver] Seeed-Voice Driver** [<https://github.com/respeaker/seeed-voicecard>]
- **[Algorithms] Algorithms includes DOA, VAD, NS** [https://github.com/respeaker/mic_array]
- **[Voice Engine] Voice Engine project, provides building blocks to create voice enabled objects** [<https://github.com/voice-engine/voice-engine>]
- **[Algorithms] AEC** [<https://github.com/voice-engine/ec>]

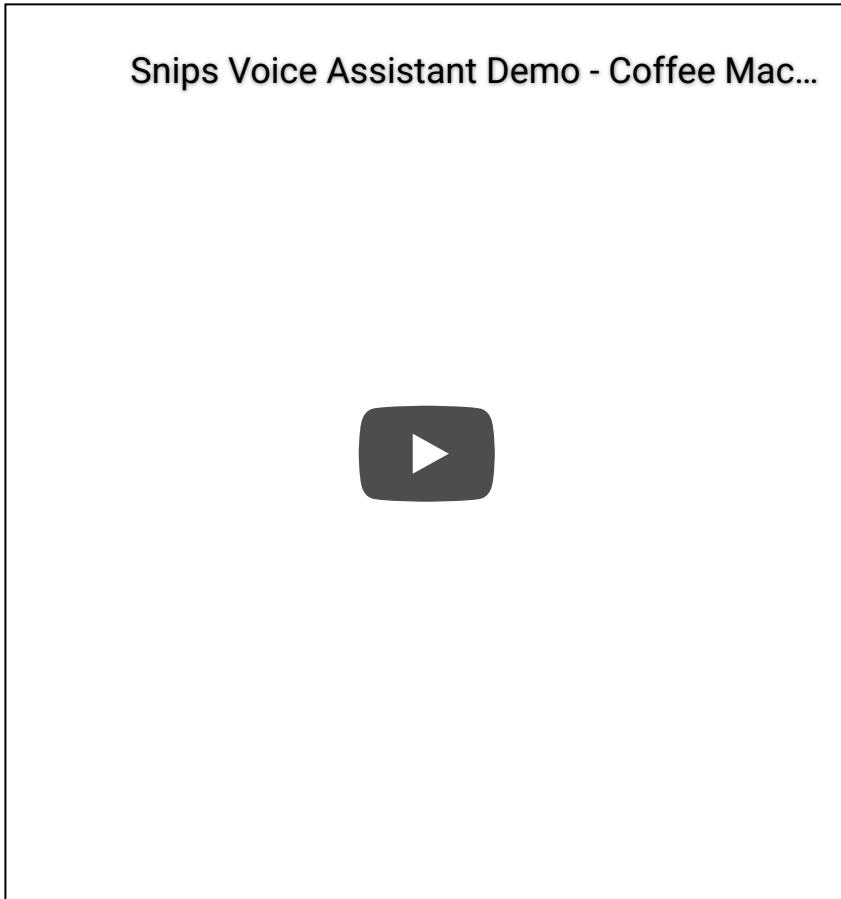
Projects

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.



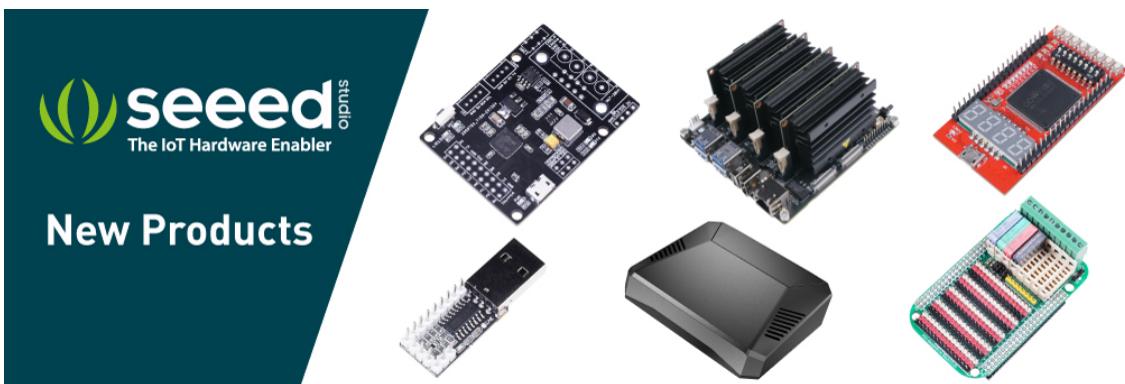
(<https://www.hackster.io/idreams/build-your-own-amazon-echo-using-a-rpi-and-respeaker-hat-7f44a0>)

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 caffeineation schedule private.



Tech Support

Please submit any technical issue into our [forum](#) [<https://forum.seeedstudio.com/>].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]