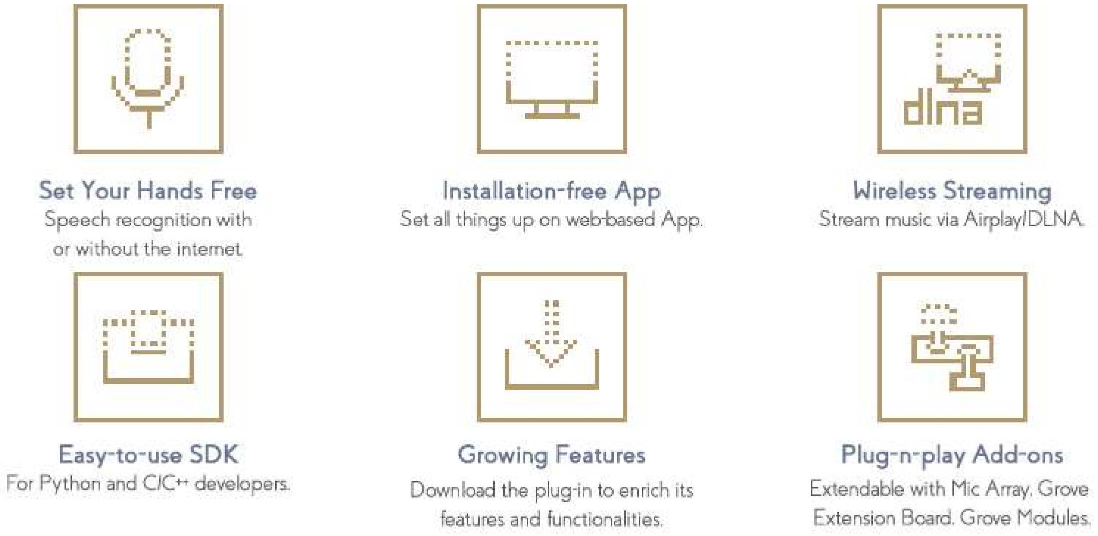
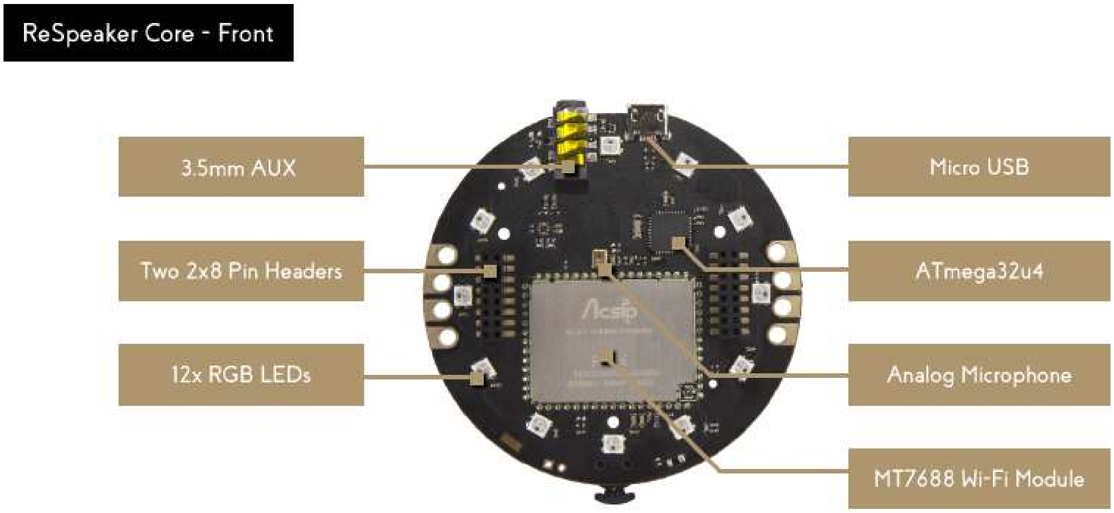
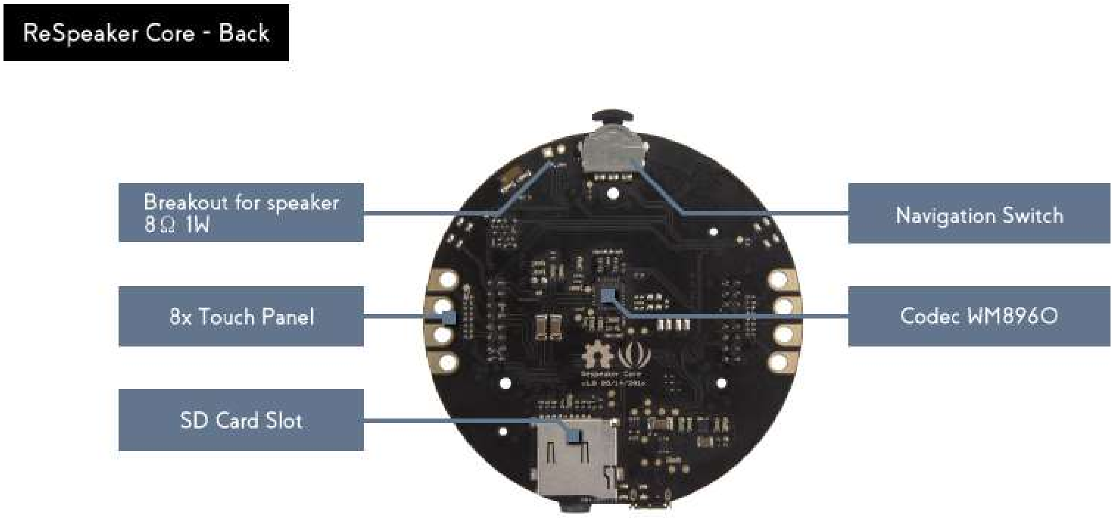
Features


Set Your Hands Free
Speech recognition with or without the internet
Installation-free App
Set all things up on web-based App.
Wireless Streaming
Stream music via Airplay/DLNA.
dlna
Easy-to-use SDK
For Python and C/C++ developers.
Growing Features
Download the plug-in to enrich its features and functionalities.
Plug-n-play Add-ons
Extendable with Mic Array, Grove Extension Board, Grove Modules.

- Set Your Hands Free: Speech recognition with or without the Internet
- Wireless Streaming: stream music via Airplay/DLNA
- Easy-to-use SDK: For Python and C/C++ developers
- Growing Features: Download and plug-in to enrich its features and functionalities.
- Plug-n-play Add-ons: Extendable with Mic Array, Grove Extension Board, Grove Modules.
- Installation-free App: Set all things on webbased App. (Not Available Yet)

Specifications



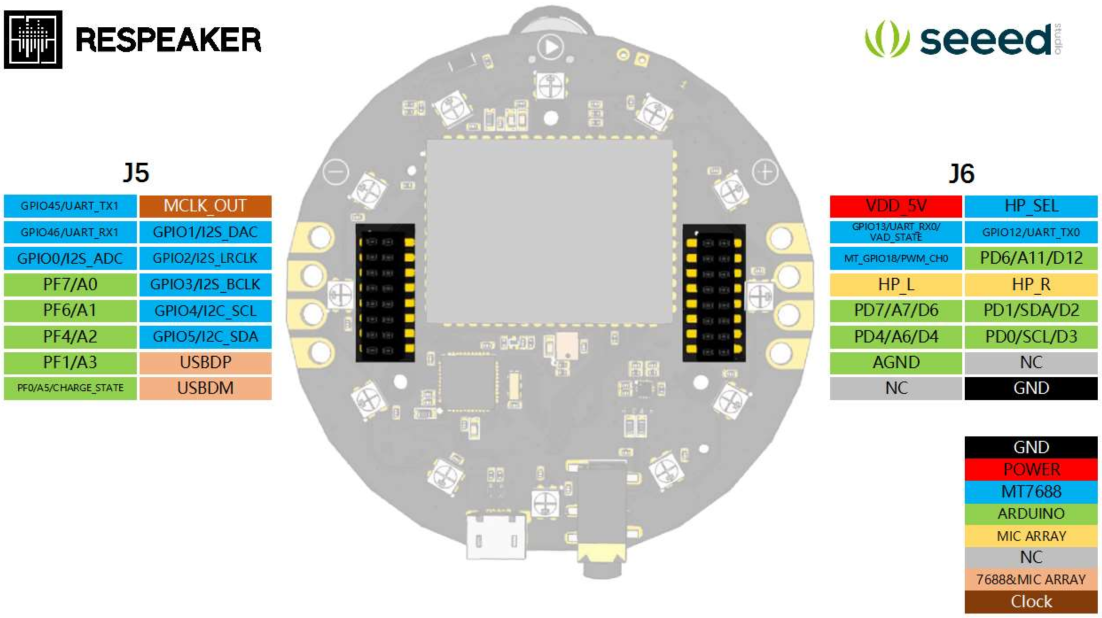


Technology Specs

- AI7688 Wi-Fi Module:
  - Operation system: GNU/Linux based OpenWrt
  - Wi-Fi Network: Support Legacy 802.11b/g and HT 802.11n modes
  - Expansion: Two expansion headers for I2C, GPIO and USB 2.0 host
  - Interfaces: Built-in 3.5mm AUX port, Micro USB and SD card slot
- ATMega32U4 Coprocessor:
  - USB CDC virtual serial port for linux console
  - 12 programmable RGB LED indicators
  - 8 on board touch sensors

- Codec WM8960:
  - DAC SNR 98dB (‘A’ weighted), THD -84dB at 48kHz, 3.3V
  - ADC SNR 94dB (‘A’ weighted), THD -82dB at 48kHz, 3.3V
  - Stereo Class D Speaker Driver with 87% efficiency (1W output)
  - On-chip Headphone Driver
  - 40mW output power into 16Ω at 3.3V
  - THD -75dB at 20mW, SNR 90dB with 16Ω load
  - On-chip PLL provides flexible clocking scheme
  - Sample rates: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48 kHz
- Power Supply: 5V DC
- Dimensions: 70mm diameter
- Weight: 17g

Pin-out Diagram



J5

GPIO45/UART_TX1	MCLK_OUT
GPIO46/UART_RX1	GPIO1/I2S_DAC
GPIO0/I2S_ADC	GPIO2/I2S_LRCLK
PF7/A0	GPIO3/I2S_BCLK
PF6/A1	GPIO4/I2C_SCL
PF4/A2	GPIO5/I2C_SDA
PF1/A3	USBDP
PF0/A5/CHARGE_STATE	USBDM

J6

VDD_5V	HP_SEL
GPIO13/UART_RX0/ VAD_STATE	GPIO12/UART_TX0
MT_GPIO18/PWM_CH0	PD6/A11/D12
HP_L	HP_R
PD7/A7/D6	PD1/SDA/D2
PD4/A6/D4	PD0/SCL/D3
AGND	NC
NC	GND

- GPIO0/I2S_ADC: Drive external encoder/decoder, ADC signal
- GPIO1/I2S_DAC: Drive external encoder/decoder, DAC signal
- GPIO2/I2S_LRCLK: Drive external encoder/decoder, Left/right channel sample clock
- GPIO3/I2S_BCLK: Drive external encoder/decoder, Bit clock

- MCLK_OUT: Master clock for external device
- HP_SEL: Headphone channel select. If use ReSpeaker Mic Array to output audio, set HP_SEL high
- HP_L: Analog audio left channel from ReSpeaker Mic Array
- HP_R: Analog audio right channel from ReSpeaker Mic Array
- AGND: Analog ground for audio

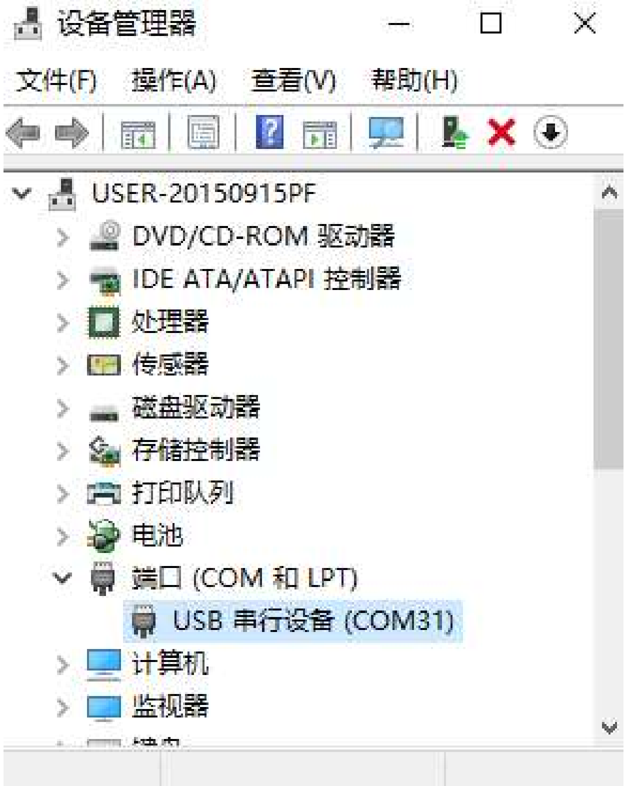
Getting Started

What should I do when I first get a ReSpeaker Core

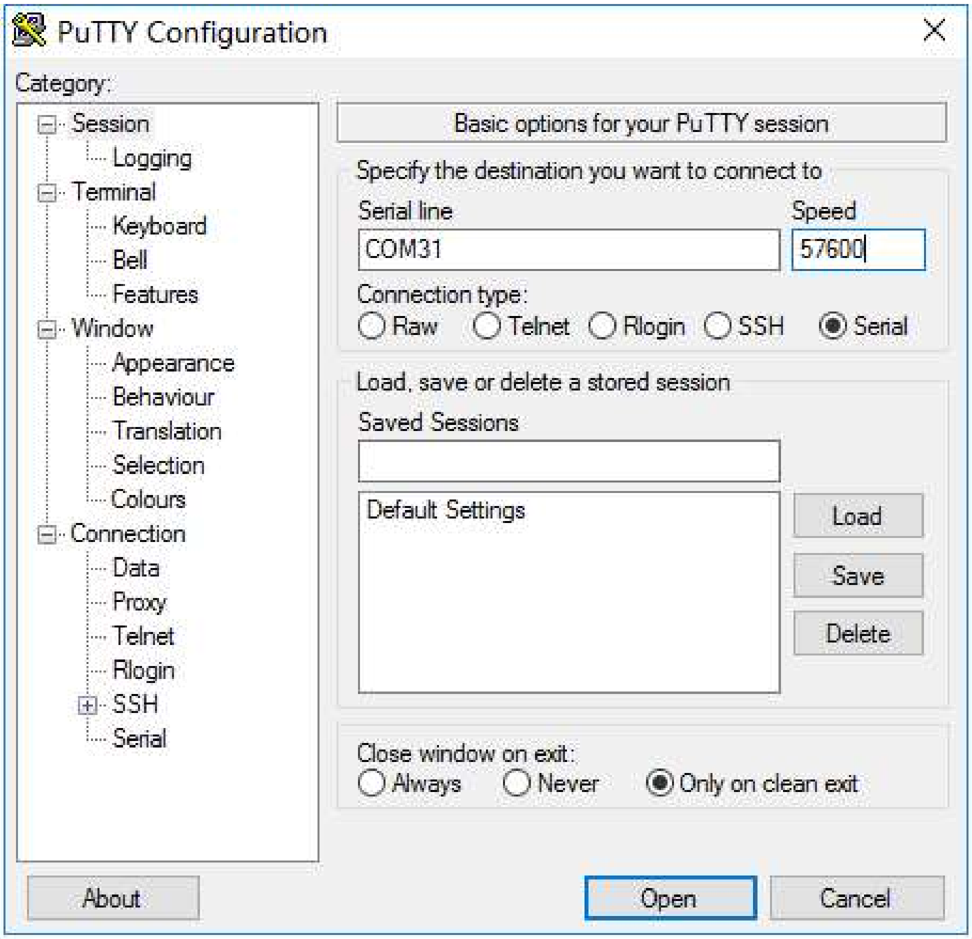
1. Preparations - ReSpeaker Core - PC or Mac - Wi-Fi network - SD card

2. Connect to serial console

- For Windows, putty is recommended.
    - Connect ReSpeaker to your PC via USB, check "Device Manager" for COM port. Here my ReSpeaker COM port is COM31.



- Select Serial under Connection type. In the Serial line field, enter the COM port for your ReSpeaker. In the Speed field, type 57600.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM31

Speed: 57600

Connection type:
○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

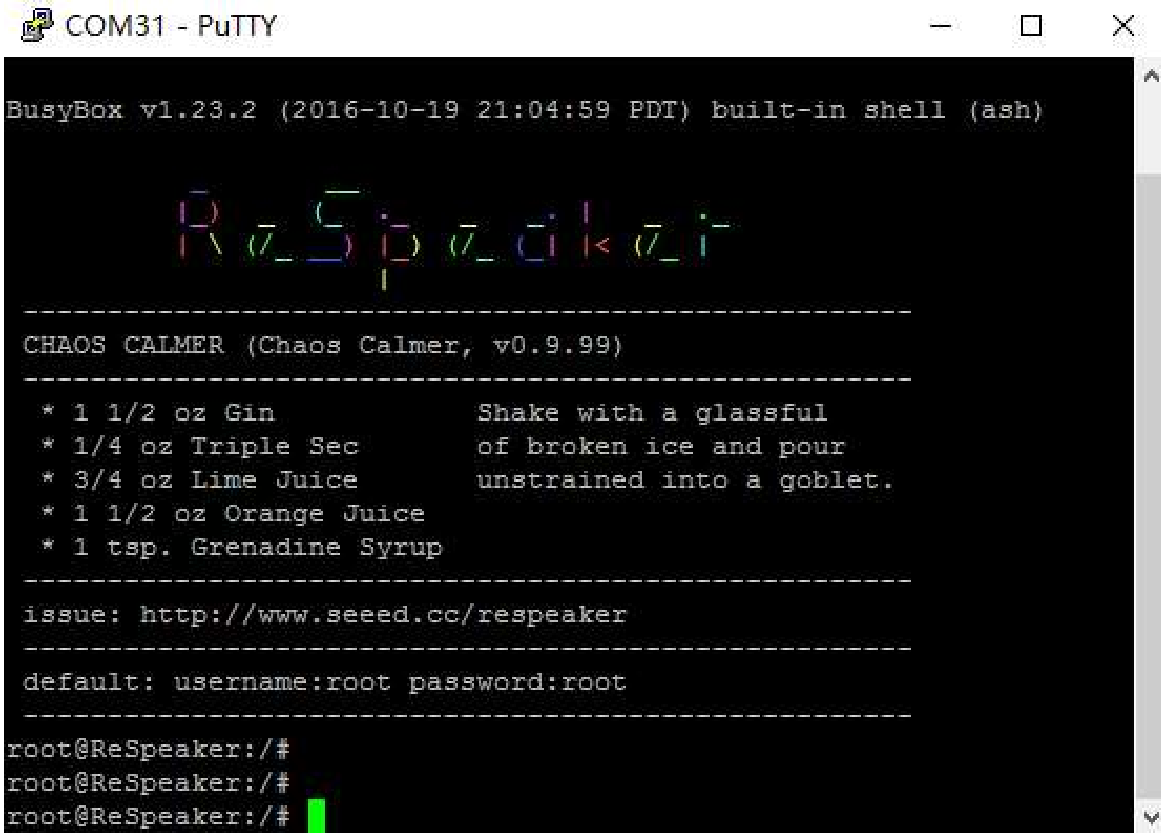
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:
○ Always ○ Never ● Only on clean exit

- Click Open. When you see a blank screen, press the Enter key.



```
BusyBox v1.23.2 (2016-10-19 21:04:59 PDT) built-in shell (ash)

         _      __
        |_) _  (_  ._   _   _. |   _  ._
        | \ (/_ __) |_) (/_ (_| |< (/_ |
                    |
 -----------------------------------------------------
 CHAOS CALMER (Chaos Calmer, v0.9.99)
 -----------------------------------------------------
  * 1 1/2 oz Gin            Shake with a glassful
  * 1/4 oz Triple Sec       of broken ice and pour
  * 3/4 oz Lime Juice       unstrained into a goblet.
  * 1 1/2 oz Orange Juice
  * 1 tsp. Grenadine Syrup
 -----------------------------------------------------
 issue: http://www.seeed.cc/respeaker
 -----------------------------------------------------
 default: username:root password:root
 -----------------------------------------------------
root@ReSpeaker:/#
root@ReSpeaker:/#
root@ReSpeaker:/#
```

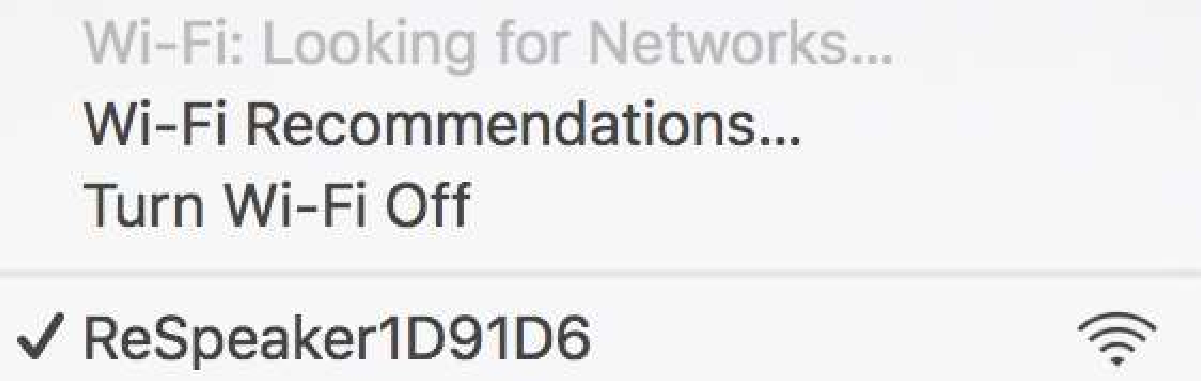
- For Linux/Mac
  - Connect ReSpeaker to your PC via USB
  - Open a Terminal session
  - Type ls /dev/tty.usb* in the Terminal. We should see a list of devices. Look for something like tty.usbmodemXXXXX where XXXXX is usually a random identifier. This is the serial device used to access the system console. Then Use the screen utility to connect to the serial port and set the baudrate to 57600. This is because the baudrate of the system console is 57600 by default.

```
$ ls /dev/tty.usb*
/dev/tty.usbmodem14221
$ screen /dev/tty.usbmodem14221 57600
```

3. Setup Wi-Fi

ReSpeaker is set to Repeater Mode as default, and you have to connect it to an existing wireless network before enjoying the speech recognition with the Internet.

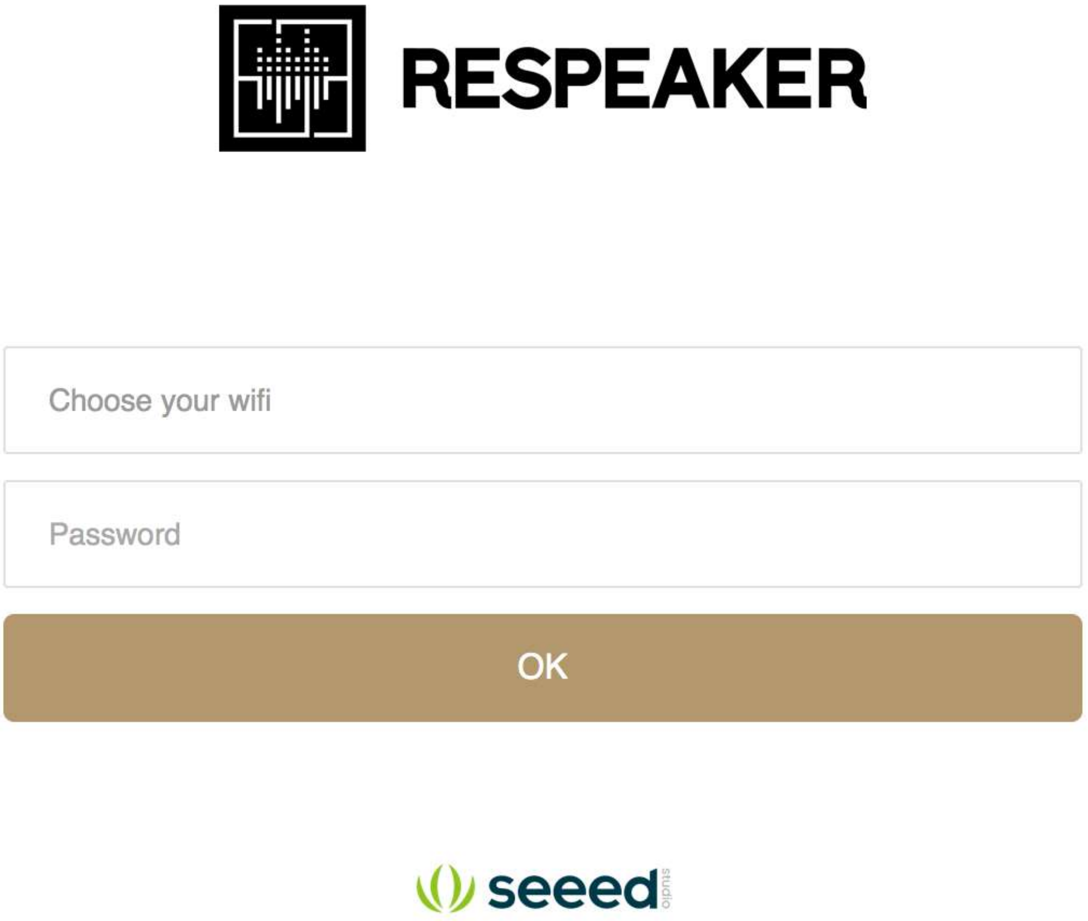
When you first power on ReSpeaker, it will create a Wi-Fi network called "ReSpeakerXXXXXX". Here "XXXXXX" is the last 6 of your ReSpeaker MAC address, which is marked on the board. Connect your computer to this network.



Note

If "ReSpeakerXXXXXX" does not appear, but "LinkIt_Smart_7688_XXXXXX" is found. Please click here.

Once you've obtained an IP address, open a web browser, and enter 192.168.100.1 in the address bar. After a few seconds, a web page will appear asking for ssid and password of an existing Wi-Fi network.



Select the Wi-Fi you wish to connect to and enter the password. When you press the OK button, ReSpeaker will join the specified network.

Now your ReSpeaker is able to visit the Internet.

Also, here is other way to setup Wi-Fi with command line. We can use wictl to scan the wifi and connect it.

```
root@mylinkit:~# wictl
0, SEEED-MKT
1, Xiaomi_Youth
2, ChinaNet-Z5C4
3, STU-EE
4, EMBEST_WIFI
5, SEEED-MKT
6, nobody
7, SEEED-MKT
```

```
8, se.VPN
9, Moovent_HK_2G
10, LinkIt_Smart_7688_1D917A
11, SEEED-MKT
12, SEEED-MKT
13, zhongrixing
Please choose your wifi: 8
Please input the wifi password: 88888888
udhcpc (v1.23.2) started
Sending discover...
Sending discover...
Sending select for 192.168.199.153...
Lease of 192.168.199.153 obtained, lease time 43200
udhcpc: ifconfig apcli0 192.168.199.153 netmask 255.255.255.0 broadcast
udhcpc: setting default routers: 192.168.199.1
success
root@mylinkit:~#
```

Note
If you can't connect Wifi with above methods, please do the factory reset by entering firstboot command.

After the Wifi function being enabled, we can use SSH to connect Respeaker by below command. We can get the Respeaker IP address fromhttp://192.168.100.1/#!/overview WAN IP. The password is root.

```
ssh root@ssh *.*.*.*
```

4. Use SD Card to Extend Storage

More often than not, a limited amount of storage is available on embedded devices(ReSpeaker has only 5M on-board flash storage left for users). More storage for applications and data can expand ReSpeaker's potential, so use SD card to extend storage as an **extroot** is a good choice.

By employing **extroot**, expansion of the storage capacity of your root file system is accomplished by using an added storage device. During the boot process, external storage space is mounted as the root file system, or in an overlay configuration over the original file system.

- Make sure your SD card is plugged into ReSpeaker and /dev/mmcblk0p1 can be detected by df -h or ls /dev.

```
root@ReSpeaker:/# df -h
Filesystem                Size      Used Available Use% Mounted on
rootfs                    1.8M    832.0K    960.0K  46% /
/dev/root                29.0M     29.0M         0 100% /rom
tmpfs                    61.7M    276.0K     61.5M   0% /tmp
/dev/mtdblock6            1.8M    832.0K    960.0K  46% /overlay
overlayfs:/overlay        1.8M    832.0K    960.0K  46% /
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/mmcblk0p1            7.4G      2.5M      7.4G   0% /tmp/run/mountd
```

- Format your SD card into two partitions, one is FAT32, the other is EXT4. EXT4 file system will be as an extroot while FAT32 will be as a normal storage device, which is able to transfer files between ReSpeaker and your PC.

```
umount /dev/mmcblk0p1
fdisk /dev/mmcblk0
 ------------------ fdisk ------------------------
>Command (m for help):o
>Created a new DOS disklabel
>Command (m for help):n
>Partition type
p   primary (0 primary, 0 extended, 4 free)
e   extended (container for logical partitions)
>Select (default p):p
>Partition number (1-4, default 1):1
>First sector (2048-15523839, default 2048):
>Last sector, +sectors or +size{K,M,G,T,P} (2048-15523839, default
>Command (m for help):n
>Partition type
p   primary (1 primary, 0 extended, 3 free)
e   extended (container for logical partitions)
```

```
    >Select (default p):p
    >Partition number (1-4, default 2):2
    >First sector (4196352-15523839, default 4196352):
    >Last sector, +sectors or +size{K,M,G,T,P} (4196352-15523839, defau
    >Command (m for help):w
    >The partition table has been altered.
    >Calling i[  292.010000]  mmcblk0: p1 p2
    >octl() to re-read partition table.
    >Syncing disks.
     ------------------ end ------------------------

    mkfs.fat /dev/mmcblk0p1
    mkfs.ext4 /dev/mmcblk0p2

    # reload mtk_sd kernel module
    rmmod mtk_sd
    insmod mtk_sd
```

- Prepare your external storage root overlay.

```
mount /dev/mmcblk0p2 /mnt ; tar -C /overlay -cvf - . | tar -C /mnt -xf
```

- Create fstab with the following command. This command will create a fstab template enabling all partitions and setting '/mnt/mmcblk0p2' partition as '/overlay' partition.

```
    block detect > /etc/config/fstab;
    sed -i s/option$'\t'enabled$'\t'\'0\'/option$'\t'enabled$'\t'\'1\'/
    sed -i s#/mnt/mmcblk0p2#/overlay# /etc/config/fstab;
    cat /etc/config/fstab;
```

- Check if it is mountable to overlay.

```
    root@mylinkit:/# mount /dev/mmcblk0p2 /overlay/
    root@ReSpeaker:/# df -h
  Filesystem                Size      Used Available Use% Mounted on
  rootfs                    1.8M    832.0K    960.0K  46% /
  /dev/root                29.0M     29.0M         0 100% /rom
  tmpfs                    61.7M    276.0K     61.5M   0% /tmp
  /dev/mtdblock6            5.2G     11.8M      4.9G   0% /overlay
  overlayfs:/overlay        1.8M    832.0K    960.0K  46% /
  tmpfs                   512.0K         0    512.0K   0% /dev
  /dev/mmcblk0p2            5.2G     11.8M      4.9G   0% /tmp/run/moun
  /dev/mmcblk0p1            2.0G      4.0K      2.0G   0% /tmp/run/moun
  /dev/mmcblk0p2            5.2G     11.8M      4.9G   0% /overlay
```

- Reboot ReSpeaker and check again. If SD card is mounted automatically, you are done. More informations about **extroot**, please click here.

5. Install software on ReSpeaker

After extending storage with a SD card, there are enough storage to install software on ReSpeaker.

Install git

```
    opkg update
    opkg install git git-http
```

6. Update Python Library

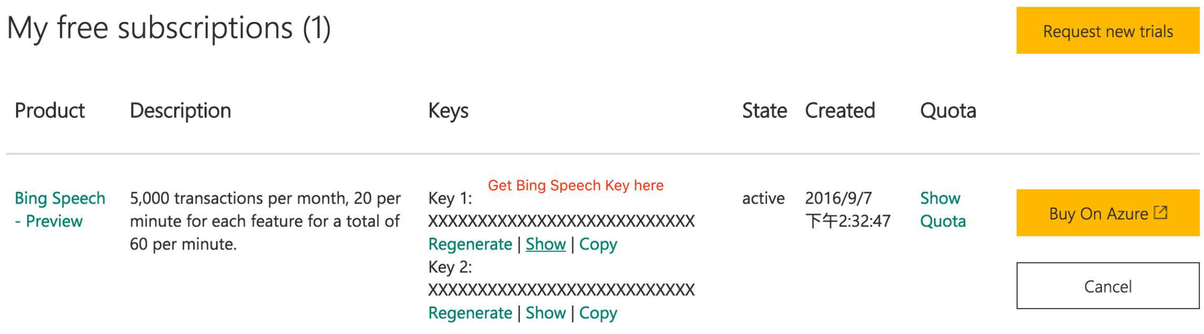
```
git clone https://github.com/respeaker/respeaker_python_library.git
cd respeaker_python_library
python setup.py install
```

First impression with Voice Interaction - ReSpeaker, play music!

With Bing Speech API, ReSpeaker can turn on and recognize audio coming from the microphone in real-time, or recognize audio from a file.

To use Bing Speech API, first you have to get a key of Microsoft Cognitive Services from here, and copy it to BING_KEY = '' , then save the following code in playmusic.py and run it

```
//stop mopidy and alexa to avoid USB device occupation
/etc/init.d/mopidy stop
/etc/init.d/alexa stop
python playmusic.py
```



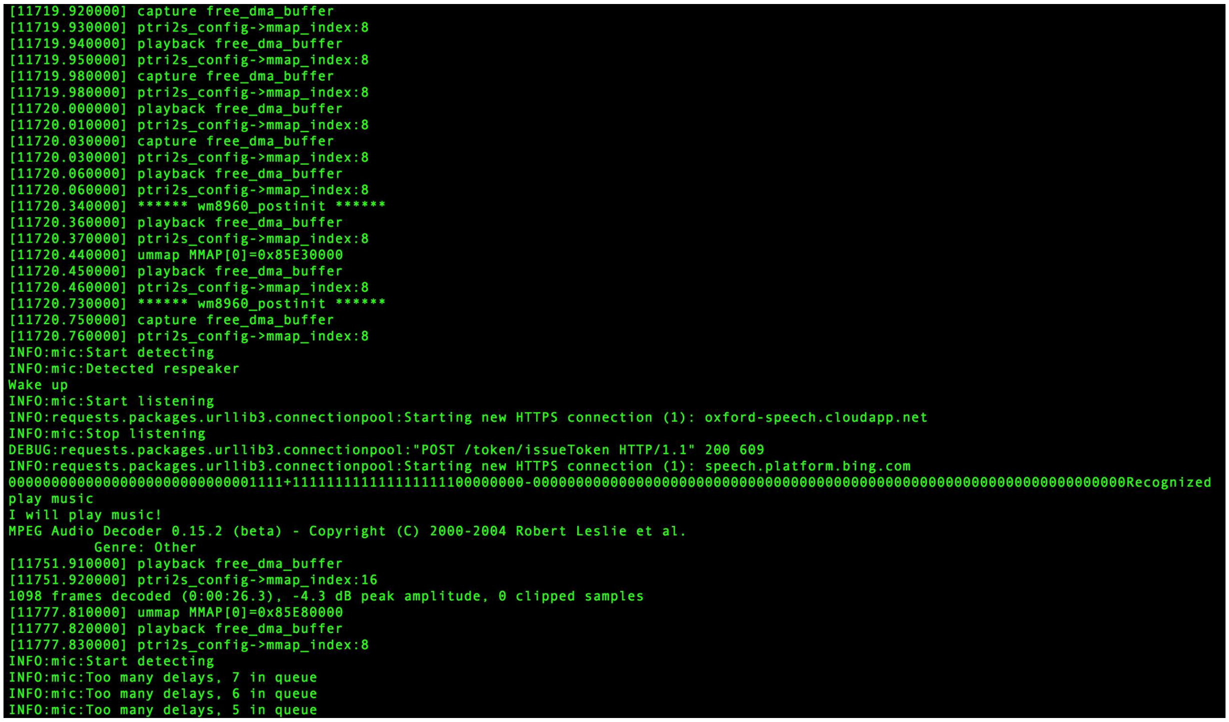
My free subscriptions (1)

Request new trials

Product	Description	Keys	State	Created	Quota	
Bing Speech - Preview	5,000 transactions per month, 20 per minute for each feature for a total of 60 per minute.	Get Bing Speech Key here Key 1: XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Regenerate | Show | Copy Key 2: XXXXXXXXXXXXXXXXXXXXXXXXXXXXX Regenerate | Show | Copy	active	2016/9/7 下午2:32:47	Show Quota	Buy On Azure Cancel

```
import logging
import time
import os
from threading import Thread, Event
from respeaker import Microphone
from respeaker.bing_speech_api import BingSpeechAPI

# use madplay to play mp3 file
os.system('madplay')

# get a key from https://www.microsoft.com/cognitive-services/en-us/spe
BING_KEY = ''


def task(quit_event):
    mic = Microphone(quit_event=quit_event)
    bing = BingSpeechAPI(key=BING_KEY)

    while not quit_event.is_set():
        if mic.wakeup('respeaker'):
            print('Wake up')
            data = mic.listen()
            try:
                text = bing.recognize(data)
                if text:
                    print('Recognized %s' % text)
                    if 'play music' in text:
                        print('I will play music!')
                        os.system('madplay Tchaikovsky_Concerto_No.1p.m
            except Exception as e:
                print(e.message)

def main():
    logging.basicConfig(level=logging.DEBUG)
    quit_event = Event()
    thread = Thread(target=task, args=(quit_event,))
    thread.start()
    while True:
        try:
            time.sleep(1)
        except KeyboardInterrupt:
            print('Quit')
            quit_event.set()
            break
    thread.join()

if __name__ == '__main__':
    main()
```

After "INFO:mic:Start Detecting" coming out, try to say "ReSpeaker" to wake up the program, and say "play music" to let it play music. Then ReSpeaker will play "Tchaikovsky_Concerto_No.1p.mp3" in the current path with **madplay** tool.



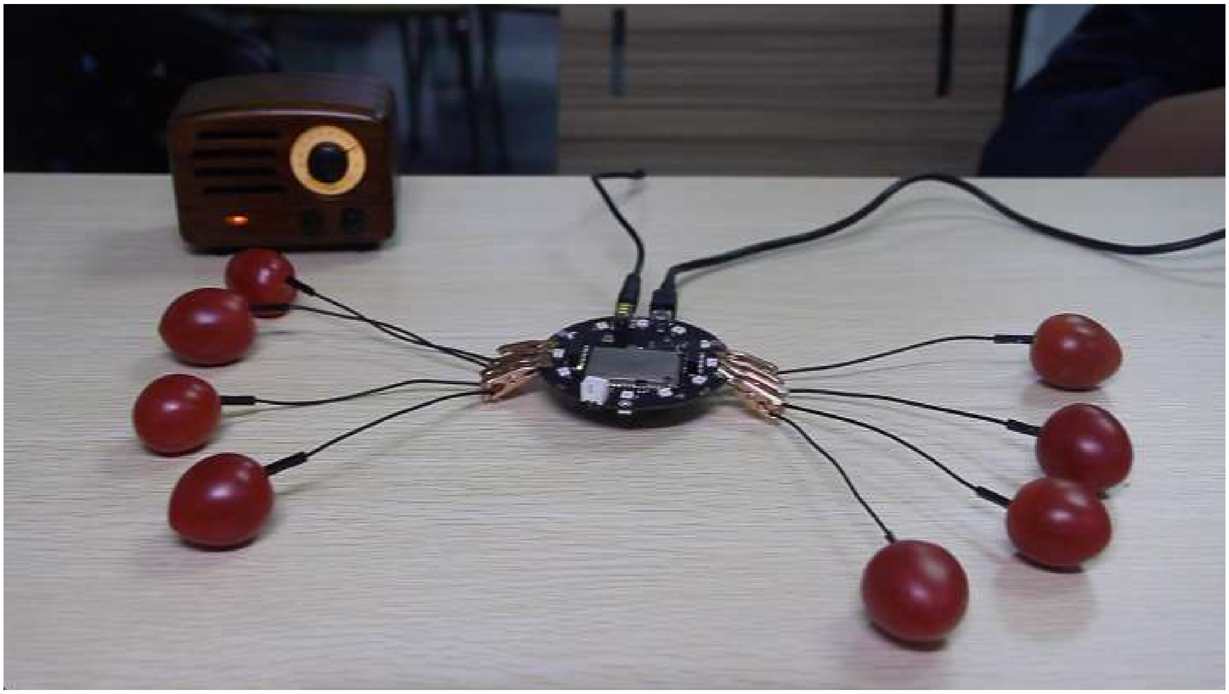
```
[11719.920000] capture free_dma_buffer
[11719.930000] ptri2s_config->mmap_index:8
[11719.940000] playback free_dma_buffer
[11719.950000] ptri2s_config->mmap_index:8
[11719.980000] capture free_dma_buffer
[11719.980000] ptri2s_config->mmap_index:8
[11720.000000] playback free_dma_buffer
[11720.010000] ptri2s_config->mmap_index:8
[11720.030000] capture free_dma_buffer
[11720.030000] ptri2s_config->mmap_index:8
[11720.060000] playback free_dma_buffer
[11720.060000] ptri2s_config->mmap_index:8
[11720.340000] ****** wm8960_postinit ******
[11720.360000] playback free_dma_buffer
[11720.370000] ptri2s_config->mmap_index:8
[11720.440000] ummap MMAP[0]=0x85E30000
[11720.450000] playback free_dma_buffer
[11720.460000] ptri2s_config->mmap_index:8
[11720.730000] ****** wm8960_postinit ******
[11720.750000] capture free_dma_buffer
[11720.760000] ptri2s_config->mmap_index:8
INFO:mic:Start detecting
INFO:mic:Detected respeaker
Wake up
INFO:mic:Start listening
INFO:requests.packages.urllib3.connectionpool:Starting new HTTPS connection (1): oxford-speech.cloudapp.net
INFO:mic:Stop listening
DEBUG:requests.packages.urllib3.connectionpool:"POST /token/issueToken HTTP/1.1" 200 609
INFO:requests.packages.urllib3.connectionpool:Starting new HTTPS connection (1): speech.platform.bing.com
00000000000000000000000000001111+111111111111111111100000000-000000000000000000000000000000000000000000000000000000000000000000000Recognized
play music
I will play music!
MPEG Audio Decoder 0.15.2 (beta) - Copyright (C) 2000-2004 Robert Leslie et al.
          Genre: Other
[11751.910000] playback free_dma_buffer
[11751.920000] ptri2s_config->mmap_index:16
1098 frames decoded (0:00:26.3), -4.3 dB peak amplitude, 0 clipped samples
[11777.810000] ummap MMAP[0]=0x85E80000
[11777.820000] playback free_dma_buffer
[11777.830000] ptri2s_config->mmap_index:8
INFO:mic:Start detecting
INFO:mic:Too many delays, 7 in queue
INFO:mic:Too many delays, 6 in queue
INFO:mic:Too many delays, 5 in queue
```

If you see below error code, the mopidy is running in background and is using the USB device. So try to run /etc/init.d/mopidy stop mopidy and run your command again.

```
root@ReSpeaker:~# python playmusic.py
Usage: madplay [OPTIONS] FILE [...]
Try `madplay --help' for more information.
Exception in thread Thread-2:
Traceback (most recent call last):
File "/usr/lib/python2.7/threading.py", line 810, in __bootstrap_inner
```

Application

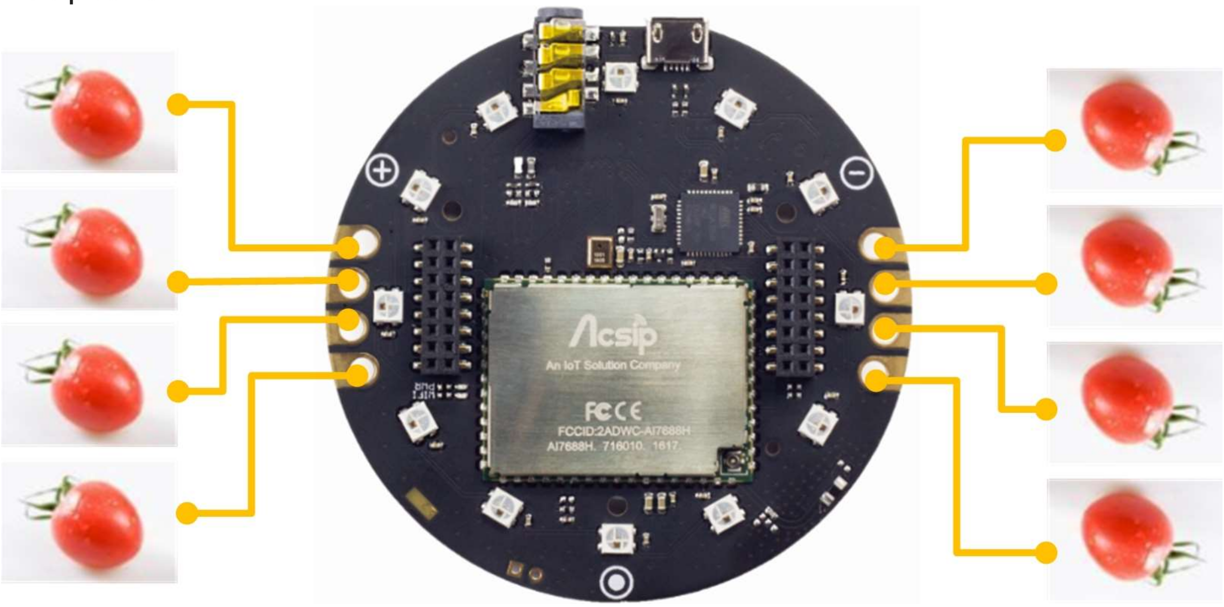
Fruit piano



Rather than the on board MT7688 Wi-FI module which runs the Linux based OpenWrt, ReSpeaker is also powered by the ATmega32u4 chip and it’s absolutely Arduino compatible, which means, we can use ReSpeaker as a powerful Arduino board and do many ‘Arduino’ things. It’s for learning, it’s for practicing, and it’s for fun.

For example, you can program it with Arduino IDE to have a special DIY piano that is built on 8 cherry tomatoes connecting to the 8 touch sensors of

ReSpeaker.



1. git clone https://github.com/respeaker/piano.gitOn ReSpeaker, download the repository
2. Download ReSpeaker Arduino Library in your computer
3. Upload piano.ino to ReSpeaker's Arduino Leonardo (ATmega32U4)
4. Run python piano.py on ReSpeaker's serial console

Weather Cloud

Weather Cloud is an awesome project for ReSpeaker. This cool build turns a ReSpeaker into a Weather Cloud, which is able to show you the whether with vivid light and sounds.

In this project, Openwrt is in charge of getting realtime weather information from the Internet, making voice interaction and audio output, while Arduino is responsible for controlling the colorful RGB LEDs.

1. git clone https://github.com/jerryyip/WeatherCloud.git on ReSpeaker, download the repository
2. Download ReSpeaker Arduino Library in your computer
3. Upload pixels_pattern.ino in ReSpeaker Arduino Library to ReSpeaker's Arduino
4. Get OpenWeatherMap appid from here and copy it to appID = "" in main.py, don't forget to add your city in city=""
5. Stop mopidy service on OpenWrt before using SPI bridge /etc/init.d/mopidy stop
6. Run python main.py and say "ReSpeaker, what is the weather like?" to ReSpeaker.
7. For more details about how to make a Weather Could, please click here.

ReSpeaker Mic Array

ReSpeaker Mic Array

The ReSpeaker Mic Array can be stacked (connected) right onto the top of ReSpeaker Core to significantly improve the voice interaction experience. It is developed based on the XVSM-2000 Smart Microphone from XMOS. The board integrates 7 PDM microphones to help enhance ReSpeaker's acoustic DSP performance to a much higher level.

About our software

ReSpeaker Arduino Library

ReSpeaker Arudino Library provides the following features:

- Supported capacitive touch sensing
- Implemented WS2812 RGB LED driver
- Built USB to Serial bridge and SPI bridge between - Arduino (ATmega32U4) and linux based OpenWrt (MT7688)

ReSpeaker Python Library

ReSpeaker is an open project to create voice enabled objects. ReSpeaker python library is an open source python library to provide basic functions of voice interaction.

It uses PocketSphinx for keyword spotting and uses webrtcvad for voice activity detecting.

More informations please go to our GitHub Repositories

FAQ

Q1: How to do Factory Reset?

- Open the serial console or a ssh session and run firstboot. More detail.

Q2: How to Rescue from a failed upgrade?

- When the respeaker failed to boot into its openwrt system, we can't access the system through web terminal, ssh or serial console. We can follow Rescue instruction to recover it.

Q3: ReSpeaker fail to find my Wi-Fi

- Try factory reset first.
- And the Wi-Fi Channel 12 is not supported by ReSpeaker. Make sure your router is not using that channel.

Q4: Wifi-Config

- We advise you to configure Wi-Fi via WEB-UI and if it can't be used, try command line tool wictl at the serial console.

Q5: How to change BING speech api recognize language

- If you don't need to change the wake up words, just change text = bing.recognize(data) into text = bing.recognize(data,language="zh-CN") is fine. More details.

Q6: Got SD card warning message "Volume was not properly unmounted. Some data may be corrupt. Please run fsck"

- If the files on the SD card are fine, ignore it. Otherwise, try to format it with sd card formatter.

Q7: Bad flash from Arduino

- Re-flash the bootloader on openwrt.

```
/etc/init.d/mopidy stop  # stop mopidy if it's running, mopidy-hallo pl
/etc/init.d/alexa stop      # stop alexa if it's running
mt7688_pinmux set ephy gpio
cd /etc/arduino
avrdude -c linuxgpio -p m32u4 -e -U lfuse:w:0xFF:m -U hfuse:w:0xD8:m -U
```

Q8: Forgot the password of WebUI

- Reset the juci password

```
orangectl passwd root 12345678  //replace 12345678 with the password yo
```

Q9: How to support google speech or other Speach TO Text(STT) Engine?

- Install speech_recognition library following the guide

Q10: Failed to run Alexa with error "IOError: [Errno -9998] Invalid number of channels"

- There is another application or alexa instance using the audio input device. Run /etc/init.d/alexa stop and /etc/init.d/mopidy stop to stop them. To disable mopidy to startup, run /etc/init.d/mopidy disable.

Q11: Failed to run python playmusic.py

- It should be that mopidy is running in background and is using the USB device. try to run /etc/init.d/mopidy stop to stop mopidy and run your command again.

Q12: Don't have a RPC connection

- You need to reflash the firmware, following the guide

Q13: SFTP & FTP - We don't have a FTP on respeaker, just SFTP.

Q14: Serial Console locked up

- Try to update the arduino code.

Q15: How to disable 'ap' mode - We could set the 'ssid' flag of the 'ap' interface to '' at vi /etc/config/wireless. Then the ap will be hidden.

Q16: I2C Sound card issue - We need check codec driver compatible name and codec i2c address. Then rebuild the image firmware.

Q17: Respeaker is often wakeup even no voice.

- We can reduce the false positive rate by increase the threshold in the keyword.txt, but it also reduce the sensibility.
- Another way is to adapt your voice with current acoustic module, more detail via http://cmusphinx.sourceforge.net/wiki/tutorialadapt.
- It will effectively improve individual keyword recognition, but may make the recognizer less general.

Q18: How to control GPIO pins from respeaker?

- We can use https://github.com/respeaker/respeaker_python_library/blob/master/respeaker/gpio.py
- The example is https://github.com/respeaker/respeaker_python_library/blob/master/respeaker/spi.py, which uses GPIOs to simulate SPI

Q19: How to change wake up word?

- keywords.txt contains keywords and their threshold. For example, keywords.txt from here is

```
respeaker /1e-30/
alexa /1e-30/
play music /1e-40/
```

respeaker is a keyword, 1e-30 is its threshold. To improve sensitive, we can decrease the threshold, for example, 1e-50. We should know decreasing the threshold will increase False Acceptance Rate.

If you want to add new keyword, you should firstly add the keyword to dictionary.txt. The dictionary.txt is like:

```
respeaker   R IY S P IY K ER
alexa   AH L EH K S AH
play    P L EY
music   M Y UW Z IH K
```

The first part is a name (respeaker, alexa or music), the second part is its phonemes. You can find words in a large dictionary at here.

- then change the code:

```
if mic.wakeup('respeaker'):
```

- The directoy of the keyword is /root/respeaker_python_library/respeaker/pocketsphinx-data

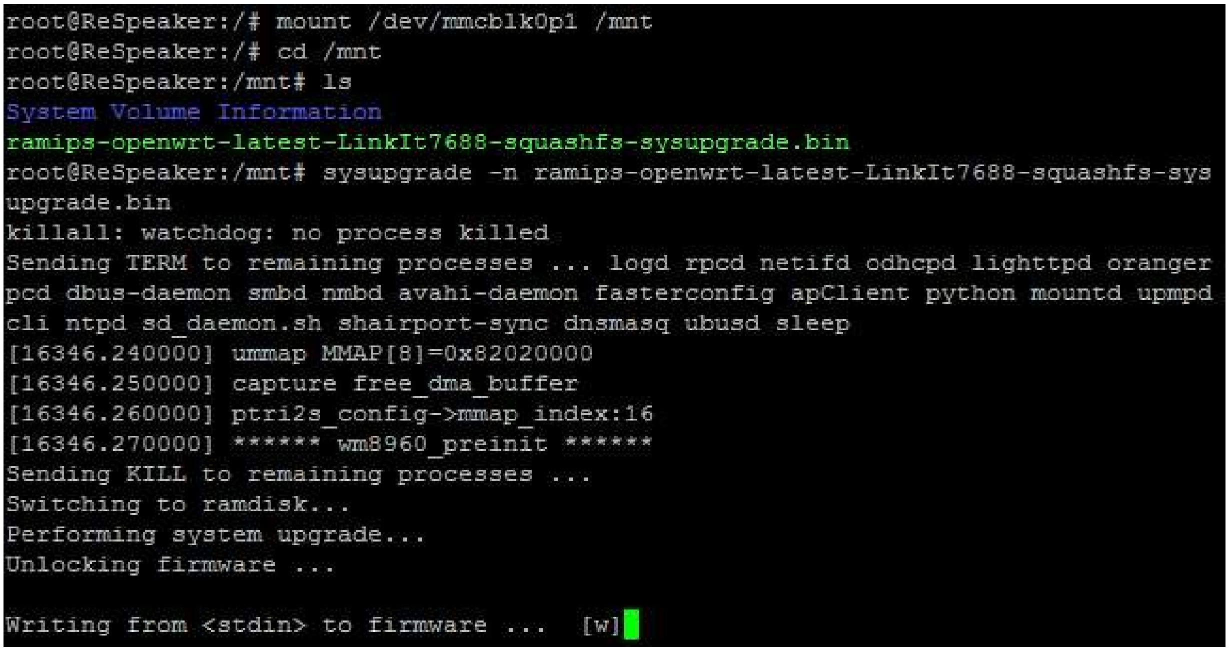
Q20: System recovery by factory image.

Note: If you can not update your ReSpeaker via Web or can not visit http://192.168.100.1/home.html, please click here. to download the lastest firmware on your computer, copy it to a SD card and plug the SD card into ReSpeaker.

Connect to the serial console. of ReSpeaker, type the following command lines to update the firmware:

```
mount /dev/mmcblk0p1 /mnt
cd /mnt
sysupgrade -n -F ramips-openwrt-latest-LinkIt7688-squashfs-sysupgrade.b
```

It will cost about 3 minutes for ReSpeaker to install the firmware and reboot, please **don't turn off**ReSpeaker when updating.



```
root@ReSpeaker:/# mount /dev/mmcblk0p1 /mnt
root@ReSpeaker:/# cd /mnt
root@ReSpeaker:/mnt# ls
System Volume Information
ramips-openwrt-latest-LinkIt7688-squashfs-sysupgrade.bin
root@ReSpeaker:/mnt# sysupgrade -n ramips-openwrt-latest-LinkIt7688-squashfs-sys
upgrade.bin
killall: watchdog: no process killed
Sending TERM to remaining processes ... logd rpcd netifd odhcpd lighttpd oranger
pcd dbus-daemon smbd nmbd avahi-daemon fasterconfig apClient python mountd upmpd
cli ntpd sd_daemon.sh shairport-sync dnsmasq ubusd sleep
[16346.240000] ummap MMAP[8]=0x82020000
[16346.250000] capture free_dma_buffer
[16346.260000] ptri2s_config->mmap_index:16
[16346.270000] ****** wm8960_preinit ******
Sending KILL to remaining processes ...
Switching to ramdisk...
Performing system upgrade...
Unlocking firmware ...

Writing from <stdin> to firmware ...  [w]
```

Resources

- **[Eagle]**ReSpeaker Core v1.0 SCH
- **[Eagle]**ReSpeaker Core v1.0 BRD
- **[PDF]**ReSpeaker Core v1.0 Schematic(pdf)
- **[PDF]**ReSpeaker Core v1.0 PCB bottom(pdf)
- **[PDF]**ReSpeaker Core v1.0 PCB top(pdf)
- **[Document]**ReSpeaker_Detail