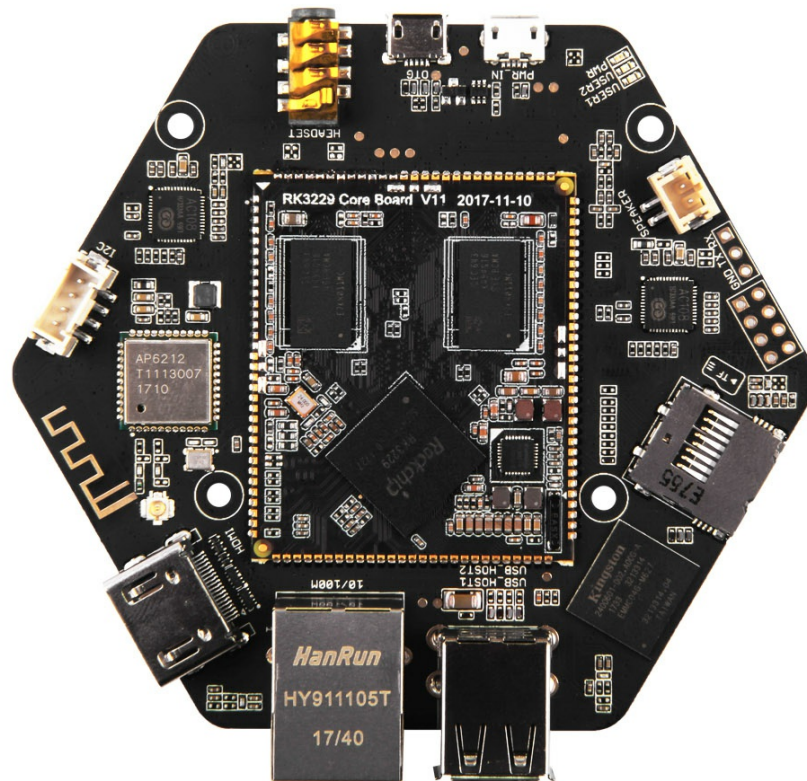


ReSpeaker Core v2



Seeed ReSpeaker Core v2 is designed for voice interactive applications. It is based on quad-core ARM Cortex-A7, up to 1.5Ghz, and 1GB RAM on-board. Besides, it features six microphone array with necessary speech algorithm, like DoA(Direction of arrival), BF(Beam-Forming), AEC(Acoustic echo cancellation) and etc.

ReSpeaker Core v2 runs GNU/Linux operation system. It benefits from powerful and active community, we can use lot of existing software/tools for development, testing and deploy, so that rapid product development become available.

ReSpeaker Core v2 is not only designed for makers/enthusiast, but also a turnkey solution for business company. The hardware consists of two parts, one is the minimized SoC module which is small and easy for manufacturing and ready for final product, the other is a bottom board which can be full customizable.

[Get One Now](#) 

Features

- High performance SoC
- 1GB RAM & 4GB eMMC
- 6 Microphone Array
- USB OTG, USB device
- WiFi b/g/n and BLE 4.0
- Detect range: ~5 meters
- Grove socket for other sensor
- 3.5mm audio jack & JST2.0 connector
- 8 channel ADCs for 6 microphone array and 2 loopback (hardware loopback)

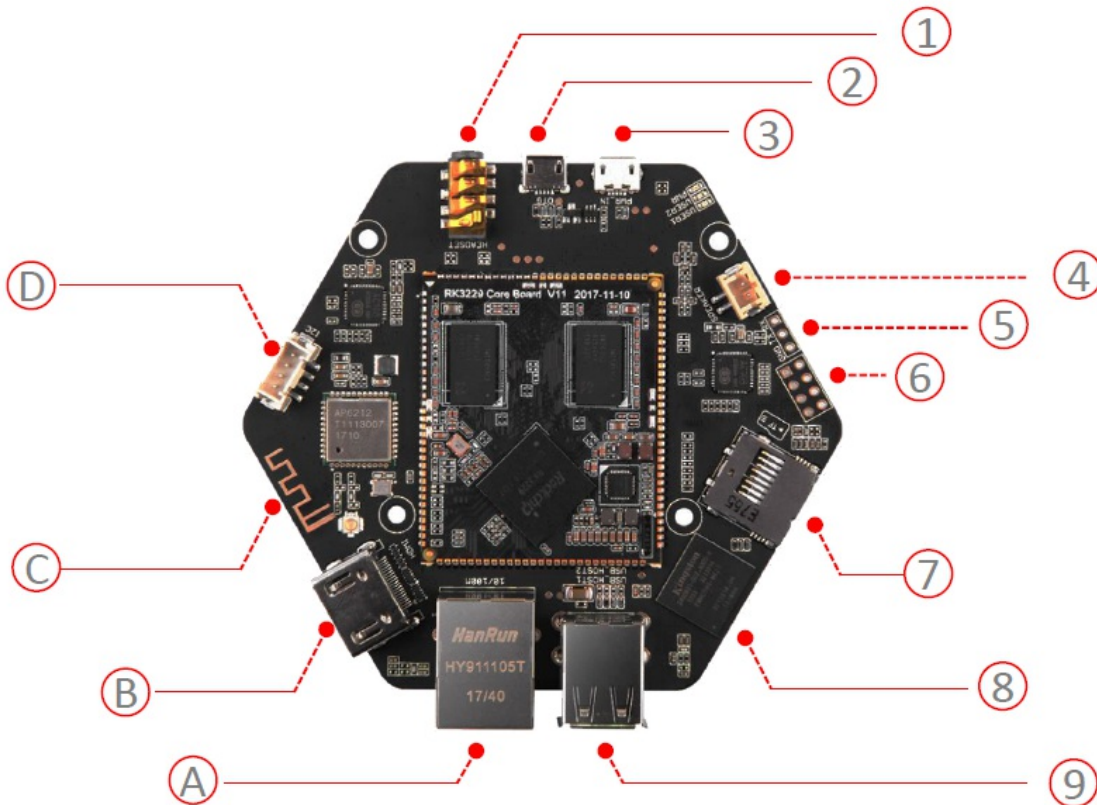
- Debian-based Linux system
- SDK for speech algorithm with Full documents
- C++ SDK and Python wrapper
- Speech algorithms and features
- Keyword wake-up
- BF(Beam-Forming)
- DoA (Direction of arrival)
- NS(Noise suppression)
- AEC (Acoustic echo cancellation) and AGC (Automatic gain control)

Specification

Features		
Soc(Rockchip RK3229)	CPU	Quad-Core Cortex-A7, up to 1.5GHz
	GPU	Mali400MP, Support OpenGL ES1.1/2.0
	Memory	1GB RAM(Core Module includes RAM and PMU)
	System	Operating Voltage:3.6-5V
		80 pins on-module
		PMU on-module
Peripheral	Networks	WiFi b/g/n; BLE 4.0; Ethernet
	USB	2 x USB Host; 1 x USB OTG; 1 x USB power
	Grove	1 x Grove socket (I2C and Digital)
	Vedio	HDMI 2.0 with HDCP 1.4/2.2, up to 4K/60Hz
	Audio	6 Microphone Array; 3.5mm Audio Jack; JST2.0 Audio output connector
	Storage	4GB eMMC on-board; SD slot
	Others	12 x RGB LEDs; 8 GPIO pins
Power Consumption	Standby Mode	360mA

Hardware Overview

Interface and storage

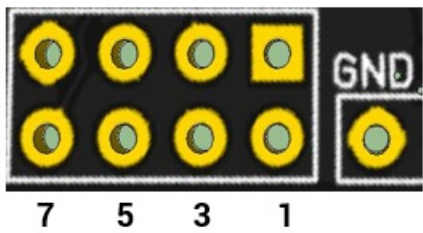
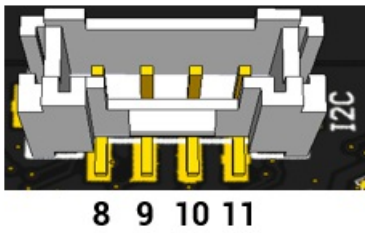


- ① **3.5mm Headphone jack:**
Output audio. You can plug active speakers or Headphones into this port.
- ② **USB OTG:**
This USB Port is used to connect to your computer via serial mode of putty (or other serial tools).
- ③ **USB Power Input:**
This port is used to provide power for Respeaker Core v2.
- ④ **Speaker Jack:**
Output audio for passive speakers. Jst 2.0 Socket.
- ⑤ **UART:**
You also can connect the ReSpeaker Core v2 with your computer via this UART port.
- ⑥ **8 Pins GPIO:**
General Purpose Input Output interface for extended applications.
- ⑦ **SD Card Slot:**
To plug in micro-SD card.
- ⑧ **eMMC:**
Embedded Multi Media Card. You can burn the image into eMMC, so that the ReSpeaker Core v2 can boot from the eMMC.
- ⑨ **USB Host:**
You can plug USB device, such as USB mouse, USB keyboard and USB flash disk into ReSpeaker Core v2 via those two USB hosts.
- Ⓐ **Ethernet:**
Access to the Internet.

- **HDMI:**
Output video.
- **Bluetooth and WIFI Antenna:**
The onboard Antenna is for WIFI and Bluetooth. Also we provide a interface for 2.4G Antenna or PCB Antenna.
- **Grove Socket:**
Grove Socket for digital or I2C.

Pin Out

Pin index definition for headers ::

8 pins header	Grove Socket
	

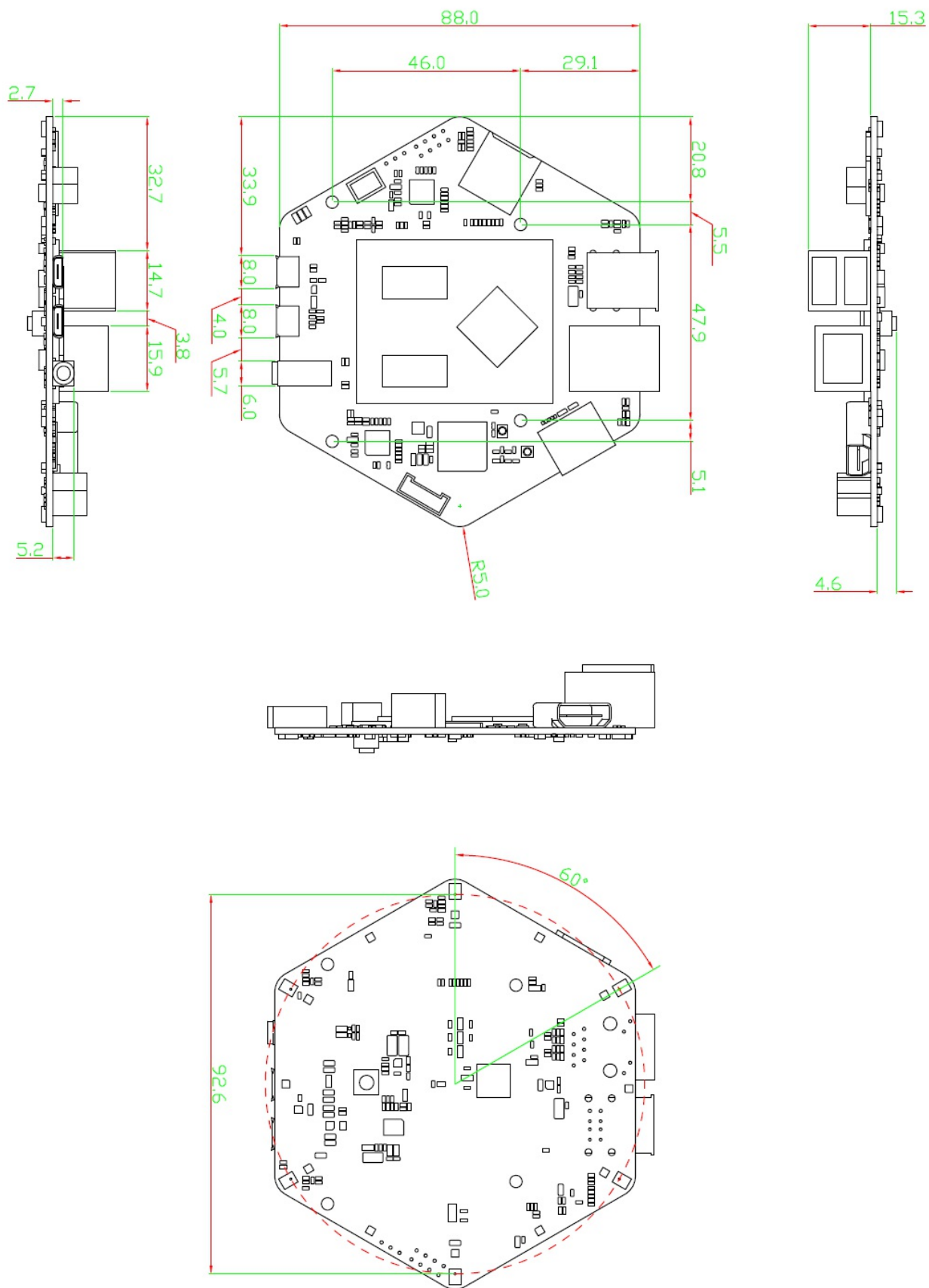
GPIO Pins

MRAA	HEADER PIN INDEX	SYSFS PIN	RK3229 PIN
0	0	1091	GPIO2_D3
1	1	--	VCC
2	2	1043	GPIO1_B3
3	3	1127	GPIO3_D7
4	4	1017	GPIO0_C1
5	5	1067	GPIO2_A3
6	6	--	GND
7	7	1013	GPIO0_B5
8	8	1085	GPIO2_C5
9	9	1084	GPIO2_C4
10	10	--	VCC
11	11	--	GND

I2C Pins

MRAA	HEADER PIN INDEX	SYSFS PIN	RK3229 PIN
0	8	--	I2C2_SCL
0	9	--	I2C2_SDA

Dimensions



Applications

- Smart speaker
- Intelligent voice assistant systems

- Voice recorders
- Voice conferencing system
- Meeting communicating equipment
- Voice interacting robot
- Car voice assistant
- Other scenarios need voice command

Getting Started

Prerequisites

- ReSpeaker Core V2
- Wi-Fi Network
- 4GB (or more) SD card and SD card reader
- PC or Mac
- [USB To Uart Adapter](#) (Optional)
- 5V 1A Micro USB adapter for power (Optional)
- two Micro-USB cables

Caution

Please plug the USB cable gently, otherwise you may damage the interface. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click [here](#) to buy

Image Installation

Similar to the Raspberry Pi, you need to install the ReSpeaker Core v2 image from an SD card to get up and running. We offer two ways to boot the Respeaker core v2. You can either boot from the SD card or [boot from the eMMC](/boot from the eMMC).

A. Boot from the SD card



- **Step 1.** Click the OneDrive icon above to download our latest image zip files: `respeaker-debian-9-lxqt-sd-*****-4gb.img.xz` OR `respeaker-debian-9-iot-sd-*****-4gb.img.xz`.

Section	Description
iot / lxqt	The lxqt version comes with a desktop GUI while the iot version does not. If you are new to ReSpeaker Core V2, lxqt version is recommended.
flasher / sd	The flasher version is used to flash the onboard eMMC, after flashing you can remove the SD card. The sd version will require the SD card to stay inserted all the time.

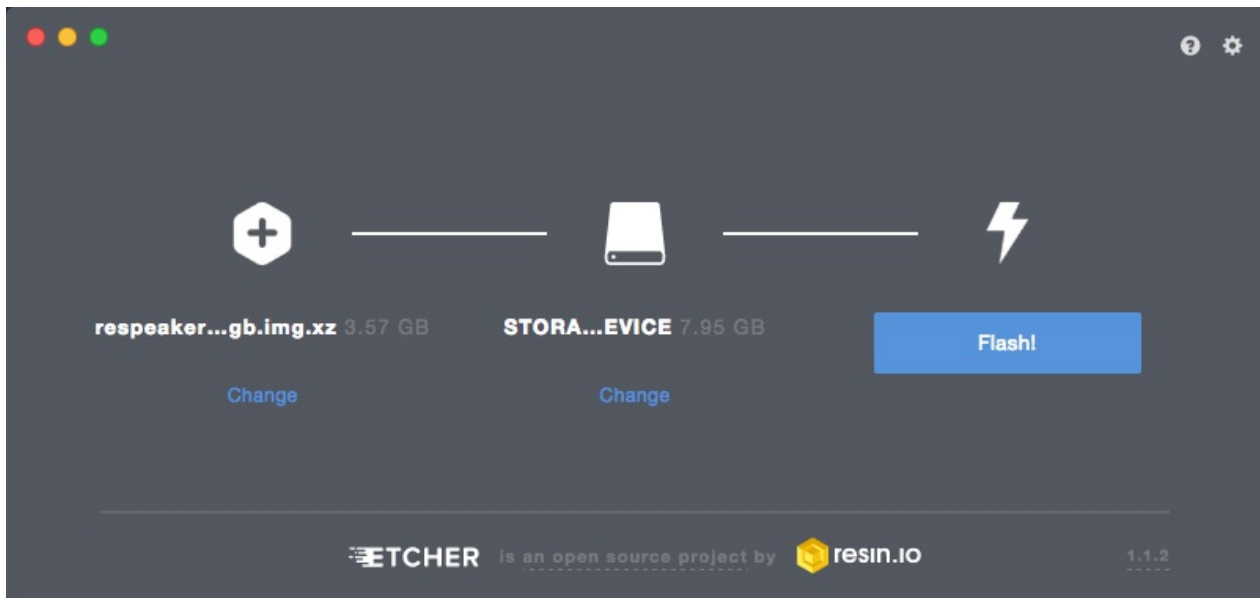
For development, we recommend the **lxqt + sd** version. So please download the **respeaker-debian-9-lxqt-sd-[date]-4gb.img.xz** file.

Note

This wiki is based on the `**respeaker-debian-9-lxqt-sd-20180118-4gb.img.xz**` version. This wiki have dropped all the verbiages of operations on old system versions, please make sure you have download the latest image.

- **Step 2.** Plug the SD card into your PC or MAC with an SD card reader. You need an SD card with a capacity of more than 4G.
- **Step 3.** Click to download [Etcher](#) here, and burn the `*.img.xz` file directly to your SD card with Etcher. Or unzip the `*.img.xz` file to a `*.img` file, then burn it to SD card with other image writing tools.

Click the Plus icon to add the image you just download, the software will automatically select the SD card you plug. Then click Flash! to start burning. It will take about 10 minutes to finish.



- **Step 4.** After writing the image to the SD card, insert the SD card in your ReSpeaker Core V2. Power the board using the PWR_IN micro usb port and DO NOT remove the SD card after powering on. ReSpeaker Core V2 will boot from the SD card, and you can see USER1 and USER2 LEDs light up. USER1 is typically configured at boot to blink in a heartbeat pattern and USER2 is typically configured at boot to light during SD card accesses. Now, you should go to the next part: Serial Console.

B. Boot from the eMMC

You also can flash the ReSpeaker image files to the ReSpeaker's eMMC (onboard flash memory) directly by using your PC or Mac. Then the ReSpeaker will boot from its eMMC (onboard flash memory) and not from the SD card.

- **Step 1.** Download our latest image zip file `respeaker-debian-9-iot-flasher-*****-4gb.img.xz` OR `respeaker-debian-9-lxqt-flasher-*****-4gb.img.xz` at OneDrive. The lxqt version comes with Debian desktop and the iot version does not. And the flasher version is for flashing eMMC, and the sd version is for booting from SD card.
- **Step 2.** Burn the `*.img.xz` file directly to SD card with Etcher, or unzip the `*.img.xz` file to a `*.img` file, then burn it to SD card with other image writing tools.
- **Step 3.** After burning SD card, insert the SD card in the ReSpeaker Core V2. Power the board using the PWR_IN micro usb port and do not remove the SD card while it's flashing.

During the flashing process, you'll see the USER1 and USER2 LEDs blink alternately. It will take about 10 minutes to complete. When the LEDs turn off, you can power off the board, pull out the SD card and power again. If the LEDs light up, that means the image was flashed to the eMMC correctly.

You can also check the image version with this command: `cat /etc/issue.net`.

Serial Console

Now your ReSpeaker Core V2 can boot, you might want to get access to the Linux system via a console, to setup the WiFi, etc. You have two ways to get the console:

- A. The OTG USB port - This requires a running Linux system on the board
- B. The UART port - This is the hard way to access the console, it can be used for debugging low level issues

A. Connection via OTG

- **Step 1.** Find a micro USB cable, and please make sure it's a data cable (not just a power cable), plug the micro USB end to the ReSpeaker's OTG micro USB port (There're two micro USB ports on the ReSpeaker board, which are labeled with different silk-screen, one is PWR_IN and another is OTG), then plug another end of this cable into your computer.



- **Step 2.** Check at your computer if the serial port has risen:
 - Windows: check the device manager, there should be new serial devices named COMx where x is an increasing number. If you use Windows XP/7/8, maybe you need install [Windows CDC drivers](#).
 - Linux: ls /dev/ttyACM*, you should get /dev/ttyACMx where x will vary depending on which USB port you used.
 - Mac: ls /dev/cu.usb*, you should get /dev/cu.usbmodem14xx where xx will vary depending on which USB port you used.
- **Step 3.** Use your favorite serial debugging tool to connect the serial port, the serial has: 115200 baud rate, 8Bits, Parity None, Stop Bits 1, Flow Control None. For examples:
 - Windows: use [PUTTY](#), select Serial protocol, fill in the correct COM port of ReSpeaker Core V2, 115200 baud, 8Bits, Parity None, Stop Bits 1, Flow Control None.
 - Linux: Depend on your USB To TTL Adapter, it could be screen /dev/ttyACM0(,1, and so on) 115200 or screen /dev/ttyUSB0(,1, and so on) 115200
 - Mac: Depend on your USB To TTL Adapter, it could be screen /dev/cu.usbserial1412(,1422, and so on) 115200 or screen /dev/cu.usbmodem1412(,1422, and so on) 115200
- **Step 4.** The default user name is respeaker, and password is respeaker too.

B. Connection via The UART port

In this section we will guide you how to establish a connection from your computer to your ReSpeaker using your USB to TTL adapter which will be connected to the ReSpeaker's Uart port (Uart port located just to the left of the ReSpeaker speaker plug).

- **Step 1.** Connect Uart port and your PC/Mac with an USB To TTL Adapter. Note that the voltage of RX/TX are 3.3V. If you don't have an USB To TTL Adapter, you can click [here](#) to get one.
- **Step 2.** Use the following Serial debugging tools with 115200 baud:
 - Windows: use [PUTTY](#), select Serial protocol, fill in the correct COM port of ReSpeaker Core V2, 115200 baud, 8Bits, Parity None, Stop Bits 1, Flow Control None.
 - Linux: Depend on your USB To TTL Adapter, it could be screen /dev/ttyACM0(,1, and so on) 115200 or screen /dev/ttyUSB0(,1, and so on) 115200.
 - Mac: Depend on your USB To TTL Adapter, it could be screen /dev/cu.usbserial1412(,1422, and so on) 115200 or screen /dev/cu.usbmodem1412(,1422, and so on) 115200.
- **Step 3.** The login user name is respeaker, and password is respeaker too.
- **Step 4.** If you do not have a USB to TTL Adapter, you may also use an Arduino. If using an Arduino, connect one end of a jumper wire to the RESET pin on the Arduino and the other end to the GND pin on the Arduino. This will bypass your Arduino's ATMEGA MCU and turn your Arduino into a USB to TTL adapter, see video tutorial [here](#). Now connect the GND pin on the Arduino to the GND pin on the Uart port of the Respeaker. Connect the Rx pin on the Arduino to the Rx pin

on the Uart port of the Respeaker. Connect the Tx pin on the Arduino to the Tx pin on the Uart port of the Respeaker. And lastly, connect the Arduino to your PC/Mac via the Arduino's USB cable. Now check that your Mac or Linux PC finds your Arduino by typing this command:

```
ls /dev/cu.usb* (Mac)
ls /dev/ttyACM* (Linux)
```

You should get back something like:

```
/dev/cu.usbmodem14XX where XX will vary depending on which USB port you used (on Mac)
/dev/ttyACMX where X will vary depending on which USB port you used (on Linux)
```

Now follow step 2 above to connect to your Respeaker over this serial connection. And note this is a one time procedure as you'll next setup your Respeaker for Wi-Fi connectivity and then connect via ssh or VNC going forward.

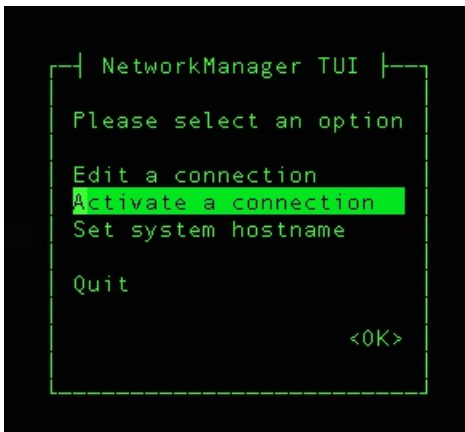
Network Setting Up

A. Wi-Fi Setting Up

Configure your ReSpeaker's network with the Network Manager tool, nmtui. nmtui will already be installed on the ReSpeaker image.

```
respeaker@v2:~$ sudo nmtui # respeaker user needs sudo
```

Then you will see a config page like this, select `Activate a connection` and press `Enter` key.



Select your Wi-Fi for ReSpeaker V2, press `Enter` key and type your Wi-Fi password and press `Enter` key again. When you see a `*` mark, it means that your ReSpeaker has successfully connected to your Wi-Fi network. Tap `Esc` key twice to leave the network manager config tool.



Now find the IP address of your ReSpeaker by using the command below.

```
ip address
```

In the example below, we can see that this ReSpeaker's IP address is `192.168.7.108`

```
root@v2:/home/respeaker# ip address

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1
    link/sit 0.0.0.0 brd 0.0.0.0
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether e0:76:d0:37:38:6d brd ff:ff:ff:ff:ff:ff
    inet **192.168.7.108**/24 brd 192.168.7.255 scope global dynamic wlan0
        valid_lft 604332sec preferred_lft 604332sec
    inet6 2601:647:4680:ebf0:ec0a:5965:e710:f329/64 scope global noprefixroute dynamic
        valid_lft 345598sec preferred_lft 345598sec
    inet6 fe80::64de:cac8:65ef:aac8/64 scope link
        valid_lft forever preferred_lft forever
```

In addition to the Netw or Manager GUI interface, Netw ork Manager also has a command line tool. If you are connecting to a hidden Wi-Fi network, you'll need to use this command line tool:

```
nmcli c add type wifi con-name mywifi ifname wlan0 ssid your_wifi_ssid
nmcli con modify mywifi wifi-sec.key-mgmt wpa-psk
nmcli con modify mywifi wifi-sec.psk your_wifi_password
nmcli con up mywifi
```

B. Ethernet Connectivity

You can connect to a network using an Ethernet cable. Just plug the Ethernet cable which has connected to the Internet will be OK.

Connect to SSH & VNC

A. SSH

SSH server starts automatically in ReSpeaker V2. For Windows Users, third-party SSH clients are available. For Linux/Mac Users, SSH client is built in.

- Windows: Use PUTTY, select SSH protocol, fill in correct IP address and click open. Login as respeaker user and password is respeaker too.
- Linux/Mac:

```
ssh respeaker@192.168.***.***
// password: respeaker
```

Note

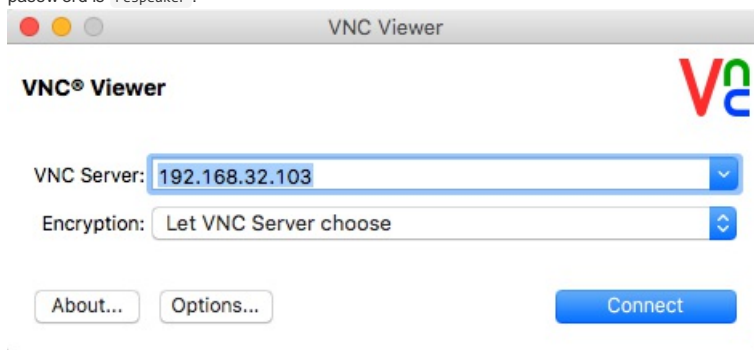
Note that if experience slow performance using SSH, please switch to a less crowded WiFi network.

B. VNC

In order to acquire the authorization from Alexa, you need to use VNC Viewer. The system has VNC server built-in. The VNC server will launch the **lxqt** desktop GUI which is a lightweight Qt desktop environment.

The VNC service also starts automatically. Use [VNC Viewer](#) or [VNC Viewer for Google Chrome](#) to connect to the desktop of ReSpeaker Core v2.

To use VNC, connect your PC/Mac and ReSpeaker v2 to the same Wi-Fi network. Then open VNC Viewer, type `192.168.xxx.xxx` at the address bar. `192.168.xxx.xxx` is IP address of the board, you can use the command **ifconfig** to check. If you meet **Unencrypted connection**, click Continue to go on. The password is `respeaker`.



Note!!!

- Please note that the VNC connection relies on good quality of the network, please have a mental preparation that you will probably get very low refresh rate of the VNC display.

Connect to Speaker or Headset

The board uses the built-in codec of the SOC to render playback. Both the JST speaker port and the headset port are driven by their own amplifier, and both amplifiers are connected to the same codec of the SOC. The sound card driver that **SEED** implemented drives both the capture device and the playback device. So there's no discrete capture or playback sound card in ALSA device list. They're all named `seed-8mic-voicecard`.

The simplest way to heard sound from the board is to plugin a headset. If you prefer loud speaker, the board can output up to 8W of drive capability.

Record and Play

1. Test via ALSA

As this is a technical documentation of development phase, the index of the sound device may change along versions. So check out the correct device index first with the following commands:

```
respeaker@v2:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: seed8micvoicec [seed-8mic-voicecard], device 0: 100b0000.i2s1-ac108-pcm0 ac108-pcm0-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: bluetoothvoice [bluetooth-voice], device 0: 100e0000.i2s2-bt-sco-pcm bt-sco-pcm-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
respeaker@v2:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: seed8micvoicec [seed-8mic-voicecard], device 1: 100b0000.i2s1-rk3228-hifi rk3228-hifi-1 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: bluetoothvoice [bluetooth-voice], device 0: 100e0000.i2s2-bt-sco-pcm bt-sco-pcm-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Find the sound card whose name has **seed** prefix. For the example above, the capture device is **hw:0,0**, the playback device is **hw:0,1**. Then test recording and playing sound with the following commands:

```
# record & playback 2 channels audio
```

```
arecord -Dhw:0,0 -f S16_LE -r 16000 -c 2 hello.wav
aplay -Dhw:0,1 -r 16000 -c 2 hello.wav

# record 8 channels audio
# there are 6 microphones on board, and ac108 compose the 2 remaining channels.
arecord -Dhw:0,0 -f S16_LE -r 16000 -c 8 hello_8ch.wav
```

2. Test via PulseAudio

First check whether the PulseAudio is running:

```
respeaker@v2:~$ ps aux|grep pulse|grep -v grep
respeak+ 1109  0.0  0.7 363272 7932 ?        S< 1  01:01   0:00 /usr/bin/pulseaudio --start --log-target=syslog
```

If it's not, please refer to PulseAudio's documentation to enable the auto-spawn of PulseAudio. Then test via:

```
parecord --channels=8 --rate=16000 --format=s16le hello2.wav
paplay hello2.wav
```

Furthermore, the default ALSA device now hooks to PulseAudio, so using the following commands also plays/records sound via PulseAudio:

```
arecord -v -f cd hello3.wav
aplay hello3.wav
```

So far we learned the basic operations of the ReSpeaker Core v2 board, let's move forward. We can use ReSpeaker Core v2 to build our own AVS(Alexa Voice Service) device or Dueros(Voice assistance of Baidu) device.

Play with AVS (Alexa Voice Service)

This guide will show you how to build an AVS device based on the ReSpeaker Core V2.

Step 1. Install AVS library (Python)

```
respeaker@v2:~$ sudo apt update
respeaker@v2:~$ pip install avs
```

This will also install the following executables into `~/local/bin`: alexa-audio-check, alexa-auth, dueros-auth, alexa-tap and alexa.

Tap the command below to Check the audio configuration:

```
respeaker@v2:~$ ~/.local/bin/alexa-audio-check
```

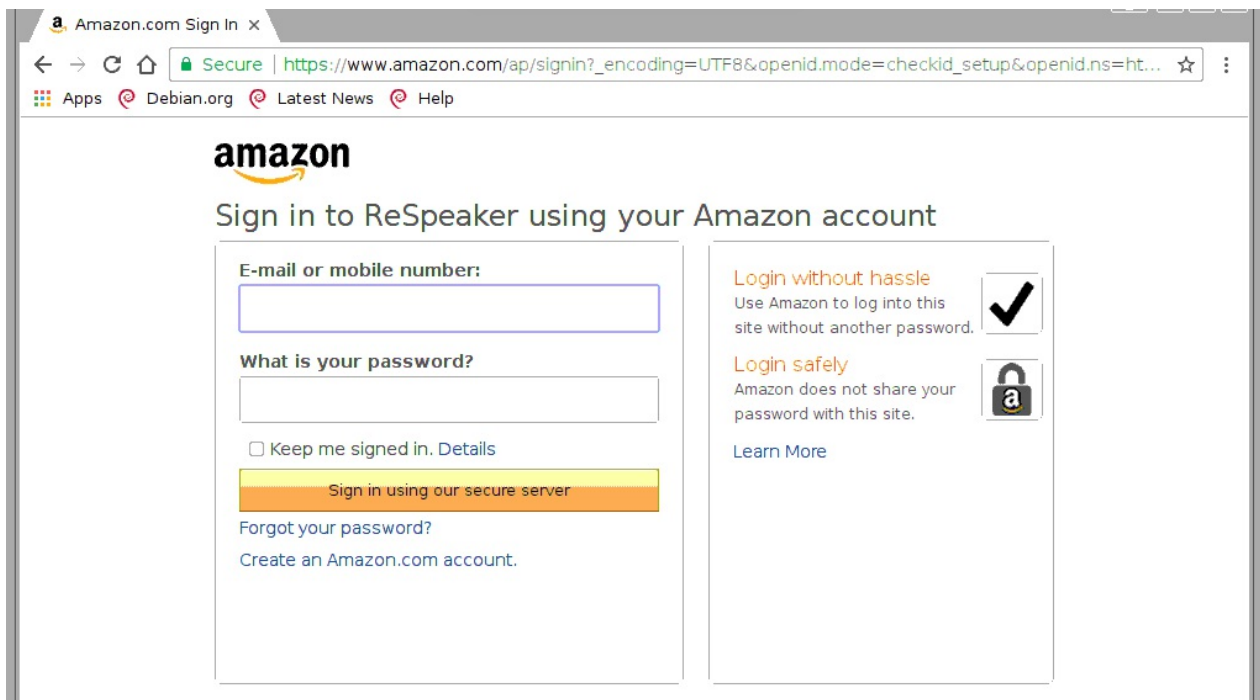
This script calculates the RMS of the sound recorded by the microphones.

Step 2. Authorize Alexa

Connect to the board via [VNC](#). In the VNC desktop, open terminal and execute:

```
respeaker@v2:~$ ~/.local/bin/alexa-auth
```

This script will open the web browser automatically, the web browser will display a login page. Sign in with your Amazon account:



After succeed you will see:



Now you can close the VNC client. The following commands can be executed in the SSH (If you prefer the VNC desktop, the terminal in VNC desktop also works).

Step 3. Have fun with Alexa Applications

We provide three python files based on Alexa, you can choose them freely.

- `Alexa-tap.py` : Use the **Enter** key to wakeup Alexa, we call it Alexa Tap to Play.
- `ns_kw_s_alexa.py` : Use the key word **Alexa** to wakeup Alexa, we call it Alexa Hands-Free.
- `ns_kw_s_alexa_with_light.py` : The same as `ns_kw_s_alexa.py`, add the LED effect, We call it Alexa with light.

Alexa Tap to Play

Tap the command below in the terminal Of Putty(SSH is recommended).

```
respeaker@v2:~$ ~/.local/bin/alexa-tap
```

Wait until you see **on_ready** in the log printing. Press **Enter** key of your computer and talk to Alexa(Only support English now).

Alexa Hands-Free via snowboy

```
sudo apt install libatlas-base-dev # required by snowboy
git clone https://github.com/respeaker/respeaker_v2_eval.git
cd respeaker_v2_eval
pip install --no-deps snowboy*.whl # install pre-build snowboy
pip install webrtc_audio_processing*.whl
pip install voice-engine
python ns_kws_alexa.py
```

Wait until you see **on_ready** in the log printing, say **Alexa** to trigger the conversation with Alexa.

Alexa With light effect:

```
pip install pixel-ring
python ns_kws_alexas_with_light.py
```

The same as last one, say **Alexa** to trigger the conversation with Alexa. You will see the LED shining while this program is running.

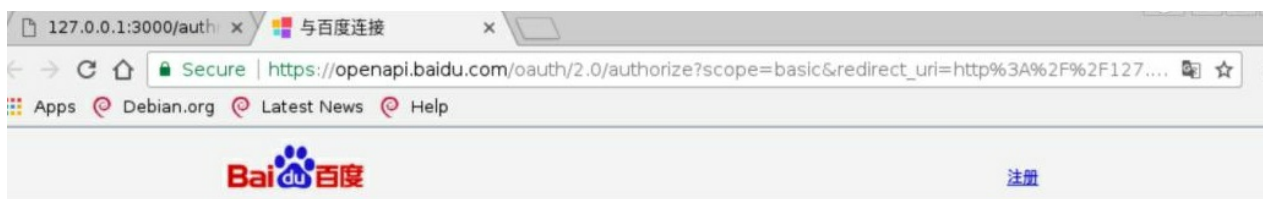
Play with Dueros (Voice assistance of Baidu)

The same as AVS, the only difference is that you need to delete one profile file. Before [get the authorization](#), you should run the command below to delete the **avs.json**.

```
rm -f ~/.avs.json
```

Then you can get the authorization from Baidu by running the following command:

```
respeaker@v2:~$ ~/.local/bin/dueros-auth
```



After login, the following step is the same as AVS, Please refer to [AVS Demos](#).

When you run the python program, you can say **Alexa** to wake up the Baidu voice assistance.

GPIO

This part will introduce how to use **MRAA** and **UPM** to control GPIO and Grove Socket on Respeaker Core v2.

Step 1. Update MRAA and UPM libraries to latest version

First, we need to check the kernel version of the system we're running, if you're not sure that you flashed the system image of version 20171128 and later.

```
uname -a
```

If you're using system image prior to version 4.4.95-respeaker-r2, please update the kernel first with

```
sudo apt update
sudo apt install linux-image-4.4.95-respeaker-r2
```

Then we install the latest MRAA and UPM packages.

```
sudo apt install python-mraa python-upm libmraa1 libupm1 mraa-tools
```

Step 2. Check your platform information

```
#only have bus 0 and id=03(/dev/i2c-3), 0 is the i2c number for mraa and upm
respeaker@v2:~$ mraa-i2c list
Bus 0: id=03 type=linux

#mraa gpio numbers and system gpio numbers and it's pinmux
respeaker@v2:~$ mraa-gpio list
00      GPIO091: GPIO
01      VCC:
02      GPIO43: GPIO
03      GPIO127: GPIO
04      GPIO17: GPIO
05      GPIO67: GPIO
06      GND:
07      GPIO13: GPIO
08      I2C2_SCL: I2C
09      I2C2_SDA: I2C
10      VCC:
11      GND:
12      GPIO66: GPIO
```

The description of the PIN defines for the ReSpeaker Core V2 board please refer to [Pin Out()]

Step 3. Demos with MRAA or UPM

A. Use MRAA Library

Control GPIO Directly

Materials

ReSpeaker Core v2	Grove - Buzzer
	
Get ONE Now	Get ONE Now

Connect the Grove PIR sensor's **SIG** pin to the ReSpeaker Core v2's header pin **0** with jumper. Don't forget to wire the VCC and GND at the same time. Then tap the code below into your console

```
respeaker@v2:~$ python
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170118] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import mraa
>>> x = mraa.Gpio(0)
>>> x.dir(mraa.DIR_OUT)
0
>>> x.write(0)
0
>>> x.write(1)
0
>>>
```

When you tap **x.write(1)**, you will hear a scream from the buzzer.

PIR Motion Sensor example

Materials

ReSpeaker Core v2	Grove - PIR Motion Sensor

	
Get ONE Now	Get ONE Now

In this example, we're gonna to listen on the trigger of the Grove PIR sensor, in Python code. Connect the Grove PIR sensor's **D1** pin to the ReSpeaker Core v2's header pin **0** with jumper. Don't forget to wire the VCC and GND at the same time. Then copy the code below into a new file and save it as a python file, name as **mraa_pir.py**. Copy this file into your ReSpeaker Core v2.

```
import mraa

def on_trigger(gpio):
    print("pin " + repr(gpio.getPin(True)) + " = " + repr(gpio.read()))

pin = 0

try:
    x = mraa.Gpio(pin)
    print("Starting ISR for pin " + repr(pin))
    x.dir(mraa.DIR_IN)
    # respeaker v2 only support EDGE_BOTH
    x.isr(mraa.EDGE_BOTH, on_trigger, x)
    var = raw_input("Press ENTER to stop")
    x.isrExit()
except ValueError as e:
    print(e)
```

Then run the code with the command below. (make sure you have located in the folder which contains the **mraa_pir.py** you've just saved)

```
sudo python mraa_pir.py
```

The result will be like



```
$ sudo python mraa_pir.py
Starting ISR for pin 0
Press ENTER to stoppin 1091 = 0
pin 1091 = 0
pin 1091 = 1
...
```

B. Use UPM Library

The UPM project implements sensors' driver based on the MRAA library, so we no longer need to care about the GPIO programming or what the I2C address of a sensor, all the default informations and logics for a particular sensor has been wrapped into a UPM library. UPM has supported bunch of sensors. <https://iotdk.intel.com/docs/master/upm/modules.html>. But please note that we didn't confirm every sensor works on the ReSpeaker Core V2.

Example for Grove Digital Light Sensor

Materials

ReSpeaker Core v2	Grove - PIR Motion Sensor
	
Get ONE Now	Get ONE Now

This is an example for the Grove Digital Light Sensor, which is copied from the UPM github repo.

Please plug the PIR Motion Sensor into your Respeaker Core v2 via the Grove socket.

Then copy the code below into a new file and save it as a python file, name as **tsl2561.py**. Copy this file into your ReSpeaker Core v2.

```
#!/usr/bin/env python
# Author: Zion Orent <zorent@ics.com>
# Copyright (c) 2015 Intel Corporation.
#
# Permission is hereby granted, free of charge, to any person obtaining
# a copy of this software and associated documentation files (the
# "Software"), to deal in the Software without restriction, including
# without limitation the rights to use, copy, modify, merge, publish,
# distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so, subject to
# the following conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
# LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
# WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

from __future__ import print_function
import time, sys, signal, atexit
from upm import pyupm_tsl2561 as upmTsl2561

def main():
    # Instantiate a digital light sensor TSL2561 on I2C
    myDigitalLightSensor = upmTsl2561.TSL2561()

    ## Exit handlers ##
    # This function stops python from printing a stacktrace when you hit control-C
    def SIGINTHandler(signum, frame):
        raise SystemExit

    # This function lets you run code on exit, including functions from myDigitalLightSensor
    def exitHandler():
        print("Exiting")
        sys.exit(0)

    # Register exit handlers
    atexit.register(exitHandler)
    signal.signal(signal.SIGINT, SIGINTHandler)

    while(1):
        print("Light value is " + str(myDigitalLightSensor.getLux()))
        time.sleep(1)
if __name__ == '__main__':
    main()
```

The result should be something like:

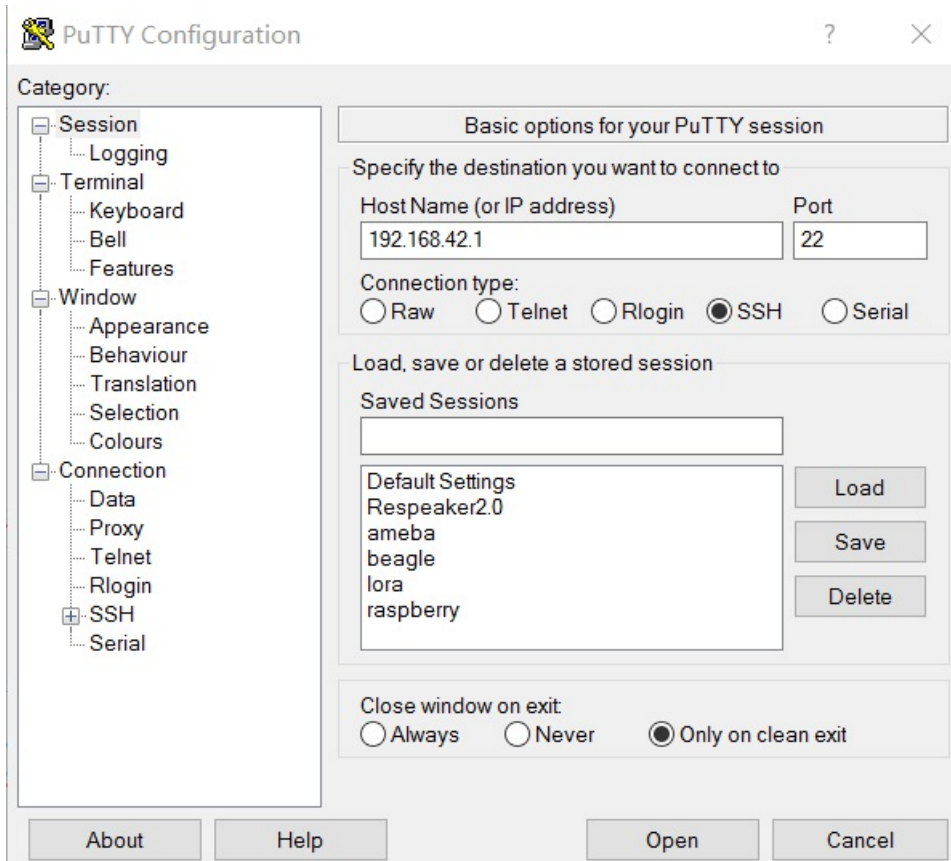
```
respeaker@v2:~$ python tsl2561.py
Light value is 0
Light value is 38
Light value is 20
Light value is 54
Light value is 13
Light value is 44
Light value is 31
```

FAQs

Q1: How to record and play with Audacity?

A1: The **lxqt** version has pre-installed Audacity, Please click the **Bird button** at the lower left corner, and you will find it at the **Sound & Video -> Audacity**.

When you opened the Audacity, please click the little black arrow to choose the record and play device and set as the picture below.



- **Step 3.** When you get into the Serial Console, you can [setup the WiFi](#)

Q3: How to adjust the volume?

A3: You can use Alsamixer to adjust the playback volume and capture sensitivity.

- **Step 1.** Tap the following code to open Alsamixer:

```
Alsamixer
```

- **Step 2.** Press **F6** on your keyboard to choose **Seeed-8mic-voicex** card.
- **Step 3.** You will find the interface as the picture below. You can choose the playback voice or record channel by pressing **Right** or **Left** key. And you can adjust the value by pressing the **Up** or **Down** key.

