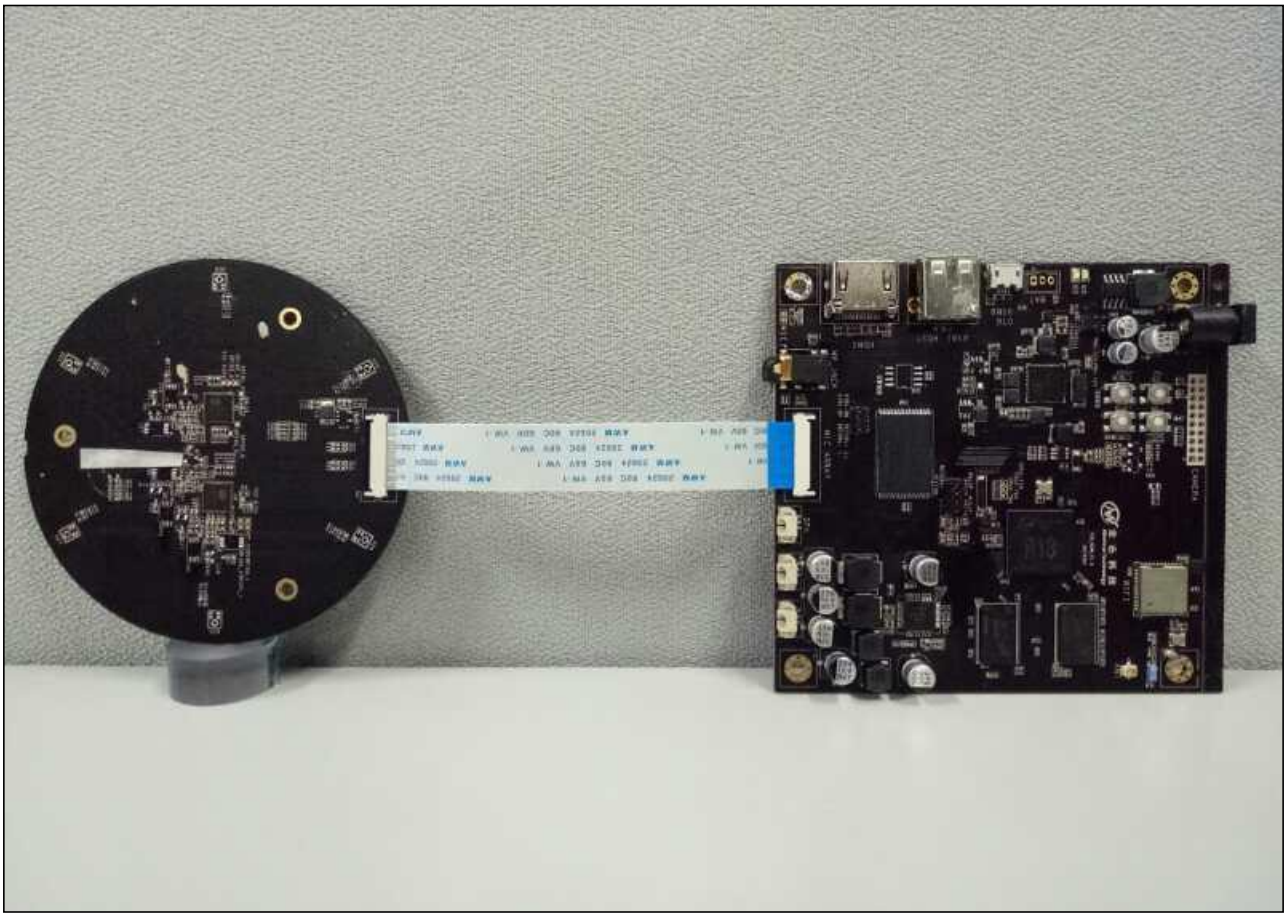
Allwinner SoC-Only 3-Mic Far-Field Dev Kit for Amazon AVS

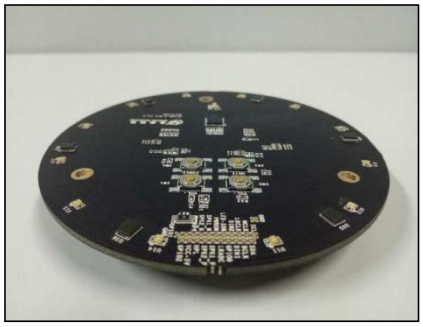
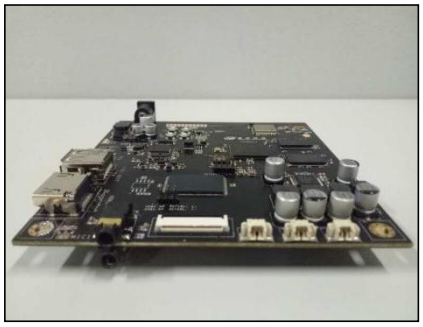
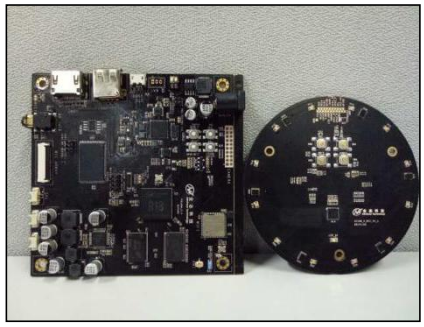
Powered by Allwinner R18 Quad-core Cortex™-A53

A quick start guide 2017/12/28



Overview

The AVS Developer Kit is a qualified device enables Alexa Voice Service (AVS) that allows developer to interact with Alexa Voice Service hands-free. It includes a 6-mic array and one base board that provides various peripheral interfaces. Services to play music, ask questions, set alarms, play iHeartRadio, news, sports updates, weather and more are available on this kit. With Gmems's front end algorithm, the developer kit is equipped with most advanced beamforming, noise reduction and echo cancellation technology so that your commands can be easily picked up even under noisy environment or during music is playing.

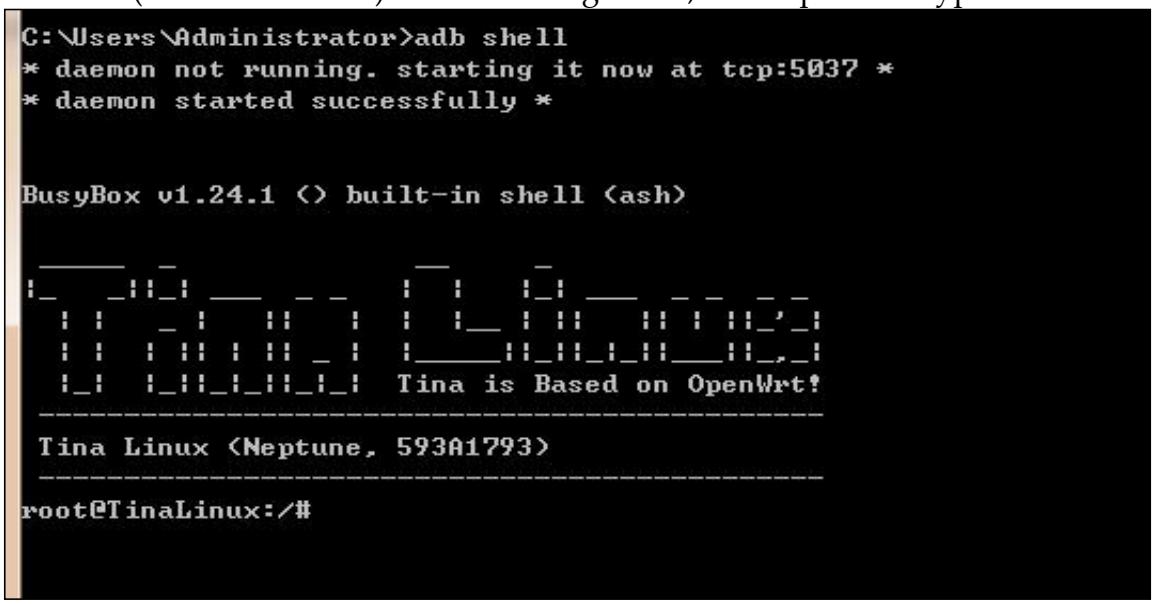




Get start with common tools

There are two common ways to debug an embedded device:

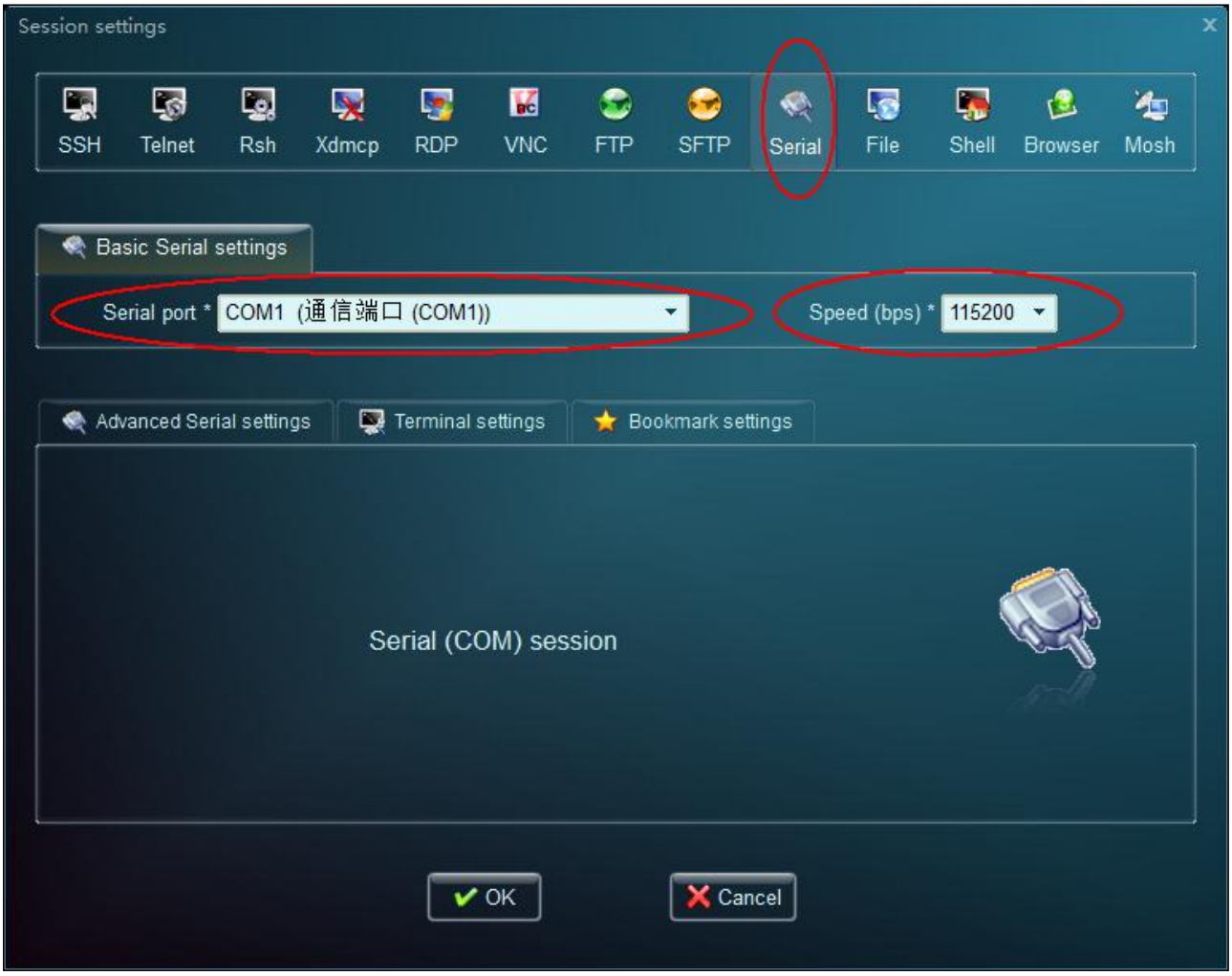
1. Via adb

adb is Android debug bridge. Type **adb shell** in command line(windows) or Terminal(MacOS or Linux) to enter debug mode, developers can type commands here.:



2. Via serial Com

on windows, you need tools like SecureCRT or putty or MobaXterm to use serial port. On Ubuntu, minicom is the most popular application to debug via serial comm. This is the setting on MobaXterm:



3. push and pull files

adb push and **adb pull** commands can push data from PC to developer kit and pull data from developer kit to PC.

```
C:\Users\Administrator>adb push "G:\Amazon Alexa\aw_ama_api.h" /etc/avs
G:\Amazon Alexa\aw_ama_api.h: 1 file pushed. 0.1 MB/s (2607 bytes in 0.019s)

C:\Users\Administrator>adb pull /etc/avs .
/etc/avs/: 10 files pulled. 2.7 MB/s (6757514 bytes in 2.366s)

C:\Users\Administrator>
```

just drag the file to command line tool, it will display the directory automatically.

Connect to Internet

Without internet, developer can't enjoy the Alexa Voice Service. Type the command below to connect to Internet:

wifi_connect_ap_test "ssid" "passwd"

ssid and passwd are your network name and your password respectively.

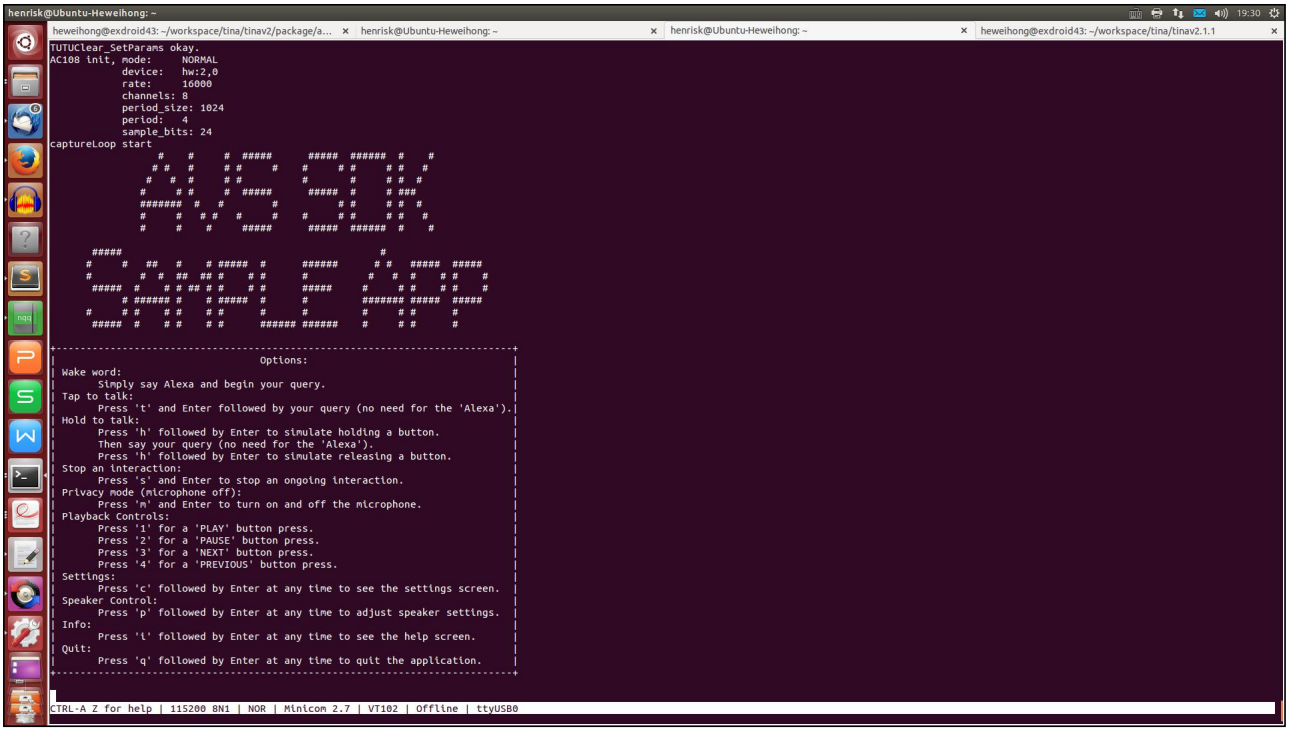
```
root@TinaLinux:/# wifi_connect_ap_test "AW2" "1111111111"

**********************************
***Start wifi connect ap test!***
**********************************
event_label 0x0
WiFi on success!
do cmd LIST_NETWORKS
call event 0xf004
event_label 0x0
It has no wifi auto connect when wifi on!
do cmd SCAN
aw wifi connect state 0xf0
enter get_key_mgmt, ssid AW2
```

Use SampleApp

SampleApp is the application integrated with voice recording, Sensory's wake word engine, Gmems's front end algorithm and Alexa Voice Service. Developer can start hands-free experience with AVS Developer Kit by following commands:

TZ=UTC SampleApp /etc/avs/test.json /etc/avs/ DEBUG9

/etc/avs/test.json is the configuration file includes device registration information and user CA certificates.



Link SampleApp with your Amazon account

1. adb pull /etc/avs/test.json /home/AVS
   pull the default test.json file to /home/AVS/ directory.
   The test.json layout is as follows:

```
{
```

```
  “authDelegate":{
      "clientSecret":"<Client Secret for your device from the Amazon Developer Portal>",
      "deviceSerialNumber":"<A unique value that you create, similar to a SKU or UPC. E.g.
"123456">",
      "refreshToken":"${SDK_CONFIG_REFRESH_TOKEN}",
      "clientId":"<Client ID for your device from the Amazon Developer Portal>",
      "productId":"<Product ID for your device from the Amazon Developer Portal>"
    },

  "alertsCapabilityAgent":{
      "databaseFilePath":"/<absolute-path-to-db-directory>/<db-file-name>",
      "alarmSoundFilePath":"/<absolute-path-to-alarm-sound>/alarm_normal.mp3",
      "alarmShortSoundFilePath":"/<absolute-path-to-short-alarm-sound>/alarm_short.wav",
      "timerSoundFilePath":"/<absolute-path-to-timer-sound>/timer_normal.mp3",
      "timerShortSoundFilePath":"/<absolute-path-to-short-timer-sound>/timer_short.wav"
  },

  "settings":{
      "databaseFilePath":"/<absolute-path-to-db-directory>/<db-file-name>",
      "defaultAVSClientSettings":{
      "locale":"en-US"
        }
  },
  "certifiedSender":{
      "databaseFilePath":"/<absolute-path-to-db-directory>/<db-file-name>"
  }
}
```

NOTE: The deviceSerialNumber is a unique identifier that you create. It is not provided by Amazon or Allwinner.

The refreshToken is the only parameter related to your own account, please follow below steps to create your own refreshToken.

You can keep other parameters same as the default test.json.

2. install python and pip on your Linux environment, then install Flask and requests:

pip install flask requests

if there is any other missing python packages.please install them in the similar way.

3. Keep the same product ID, client ID and client secret in the default test.json, put the test.json under the same directory of AuthServer.py
4. start AuthServer by running:

   python AuthServer.py

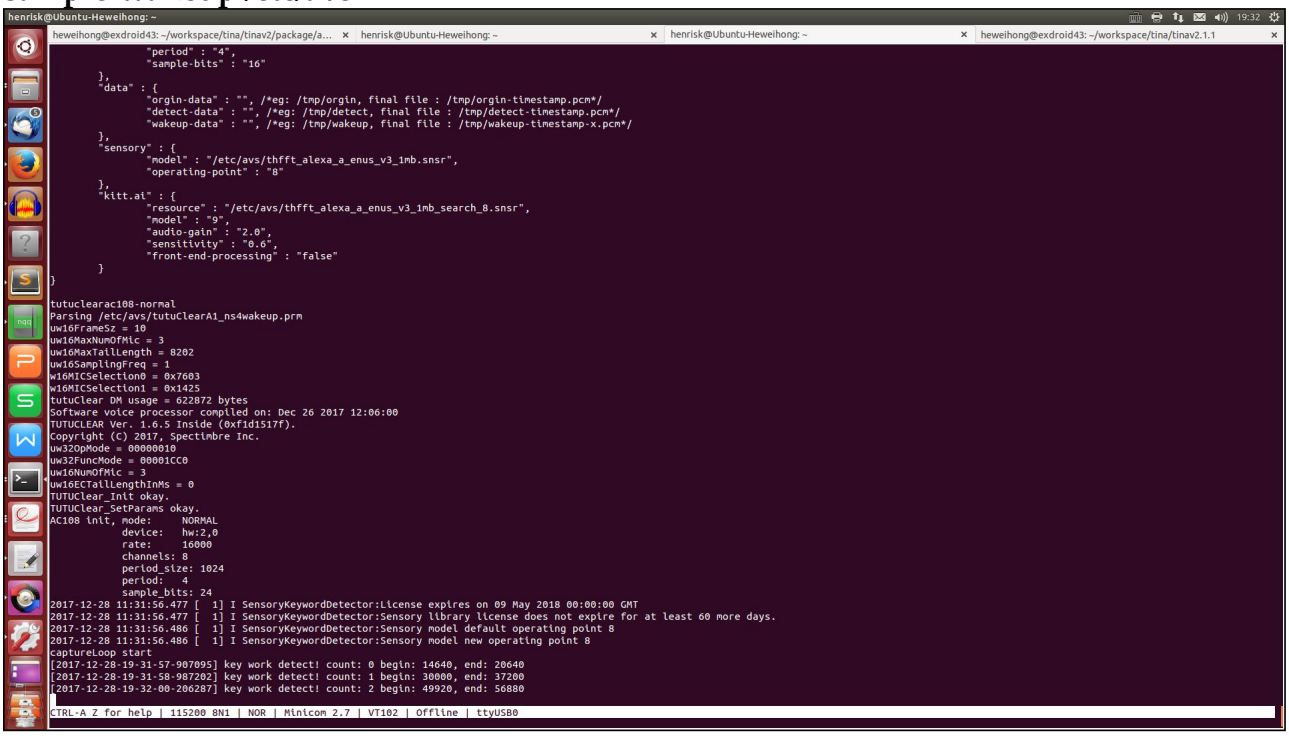
   You should see a message that indicates the server is running.
5. Open your favorite browser and navigate to: http://localhost:3000
6. Follow the on-screen instructions.
7. After you've entered your credentials, the server should terminate itself, and test.json will be populated with your refresh token.
8. **adb push C:\test.json /etc/avs/** push the new test.json to reference kit.
9. run SampleApp again and enjoy your iHeartRadio or Kindle or Amazon music.

Offline quick test tools

sample-wakeup is an offline tool to test wake-up rate for “Alexa” wake word. The purpose of this tool is to avoid network issue for Chinese developers. Developers can quickly test the performance of front end algorithm and Sensory’s Alexa wake word engine without connecting to Internet. Alexa Voice Service is not supported under offline mode.

To use it, simply type the following commands:

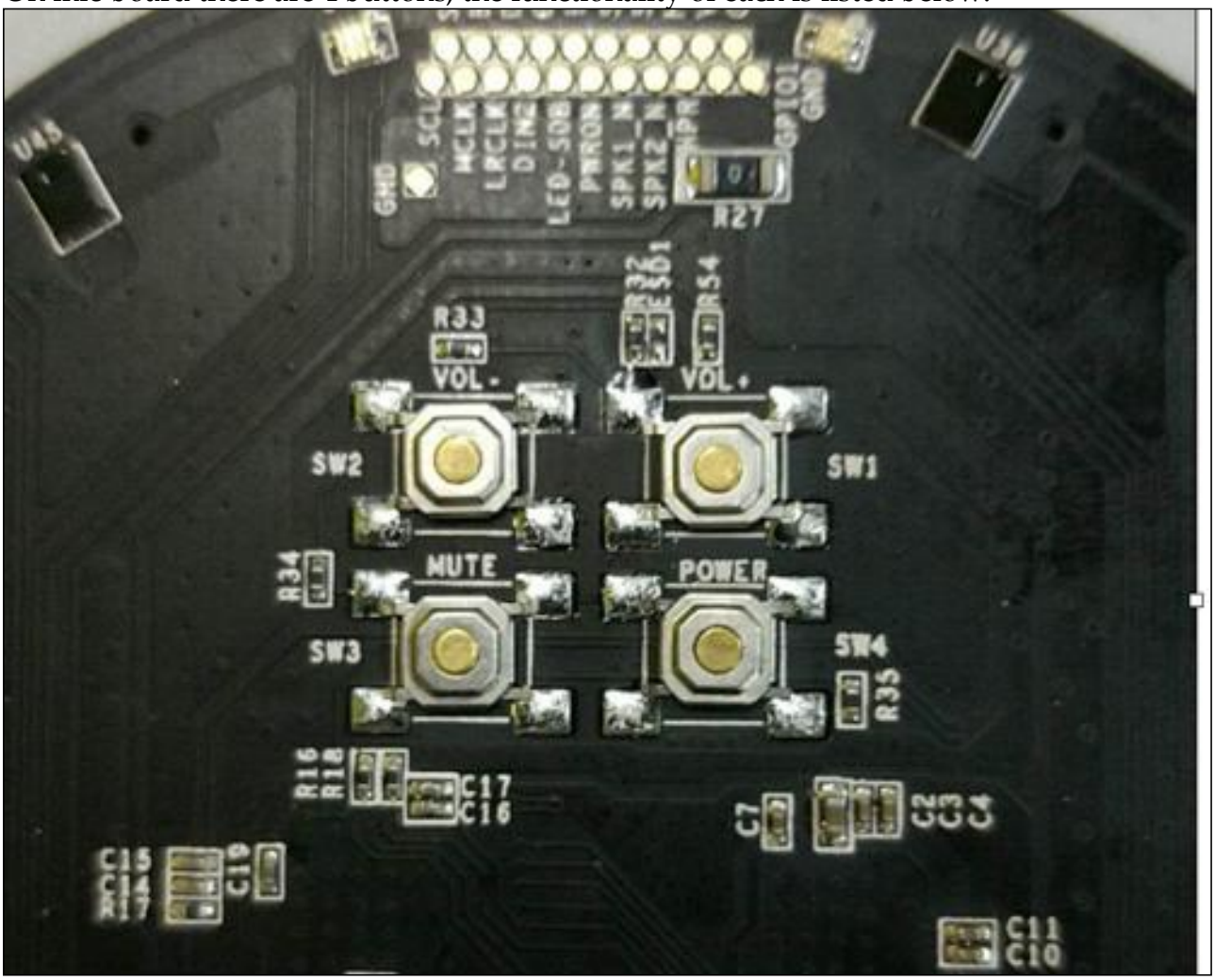
sample-wakeup /etc/avs



Introduction of hardwares

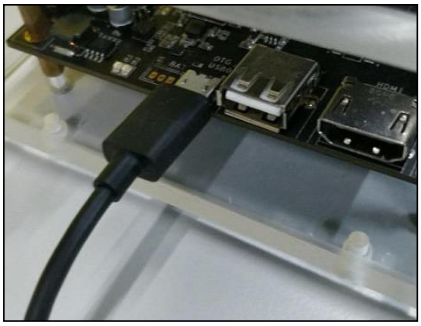
Button description

On mic-board there are 4 buttons, the functionality of each is listed below:

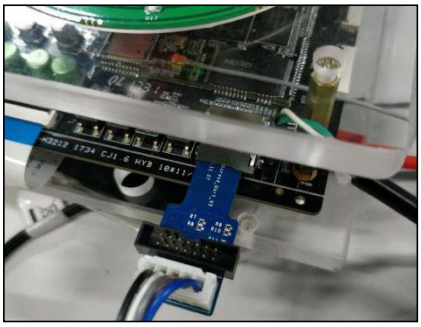


VOL- : voluome down;
VOL+ : volume up, force flash;
MUTE : Mute the microphone to enter private mode
POWER : power on/off

Cable connection



micro-usb for adb and power supply



serial comm with Allwinner TF-card uart debugger

Flash the recovery image

Necessary tools:

1. PhoenixSuit--------Flash tool on windows
2. LiveSuit----------Flsh tool on Ubuntu
3. adb driver
4. mini-USB cable
5. recovery.iso----------firmware image

Steps:

1. install the software, please see Livesuit.doc and PhoenixSuit User Manual V1.0.doc for more information.
2. download the image from our website, select the image in PhoenixSuit and Livesuit.
3. Unplug power cable and mini-USB from the board, press the VOL+ button and hold it until plug in the mini-USB cable to PC, release the button, it should ask if format the device is necessary before start flashing.

Note: documents, configuration file, software, scripts, drivers mentioned in this guide are shipped with the User guide.