Hichips-Parrot Board Software

User Manual

Shenzhen Hichips Technologies Co., Ltd.

Address: Room 321, 3rd floor, building B1, yongqi business building, yintian industrial zone, xixiang street, bao 'an district, Shenzhen 518100

People's Republic of China

Website: http://www.hichips.cn

Email: support@hichips.cn

Revision History

Revision Number	Author	Date	Description
V0.1	Zhang Lingjun	2018.11.11	Initial Draft

Contents

1. Overview

Hichips-Parrot Board(Allwinner SoC-Only 3-Mic Far-Field Dev Kit) for Amazon AVS with allwnner R18 chip design.

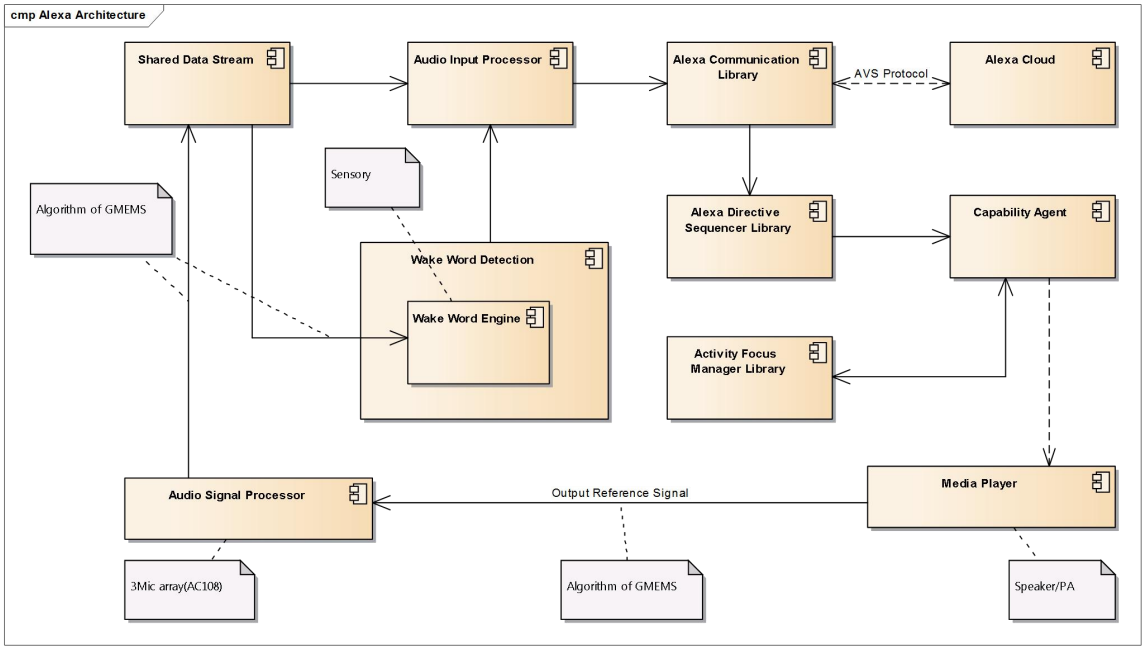
The Allwinner SoC-3-Mic Far-Field Devlopment Kit is for Amazon AVS offers far-field performance with a 3 analog mic array, and the quad-core ARM Cotex-A53-base design does not require an external DSP, the combination of which makes the development kit a cost-effective solution for integrating Alexa into finished products, Its rich features and cost-effectiveness make audio intelligence further close to our daily life. Features includes:

- Audio front end for beamforming, noise reduction, and echo cancellation
- Sensory wake word engine tuned to “Alexa”
- Client application built with the AVS Device SDK
- High performance with low power consumption
- Reserves application headroom for non-AVS applications
- Built-in battery-operated designs
- Supports 3 MICs both-side placement

2. AVS SDK introduction

2.1. Alexa Architecture

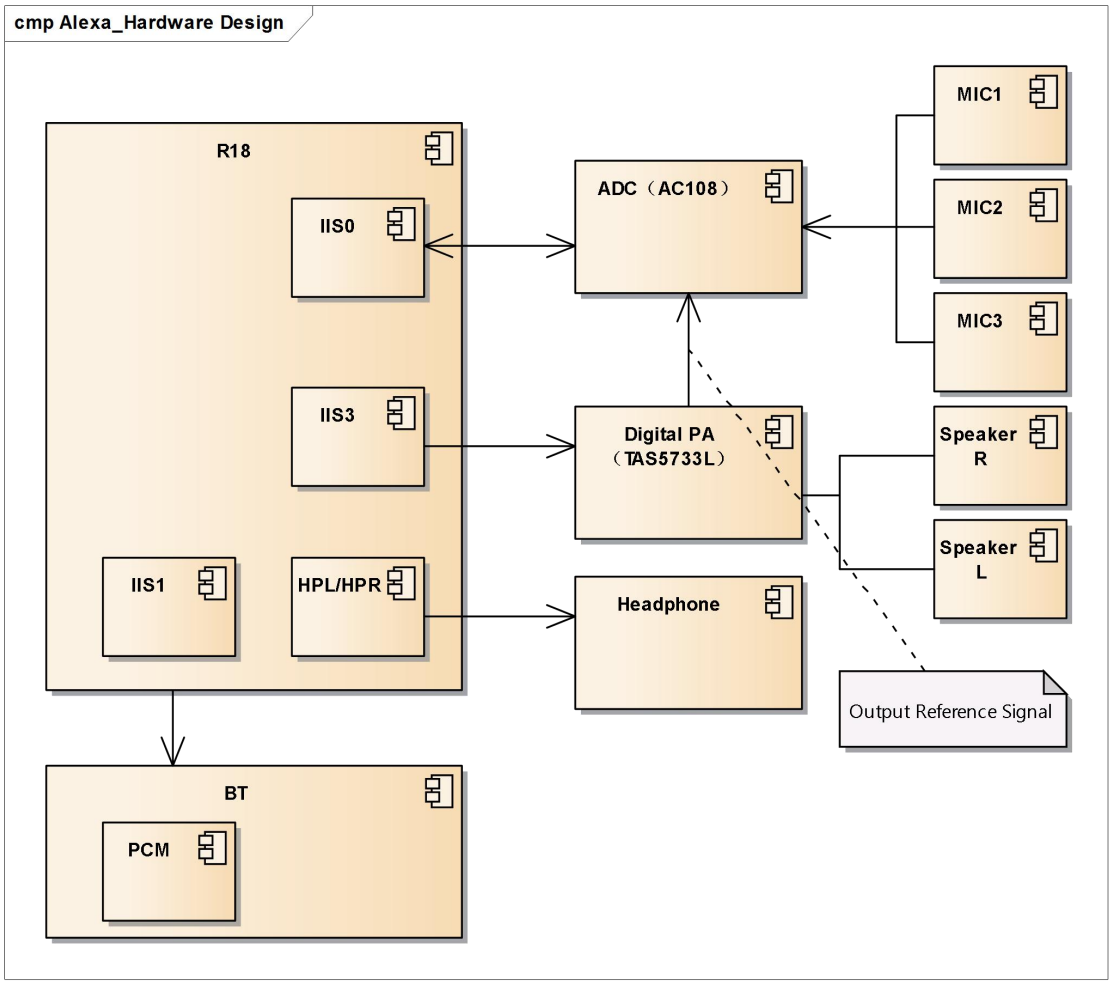
Alexa is build base on AVS SDK. The architecture is shown in this diagram below:



Alexa Architecture

2.2. Hardware design

The architecture of hardware design is shown in the diagram below:



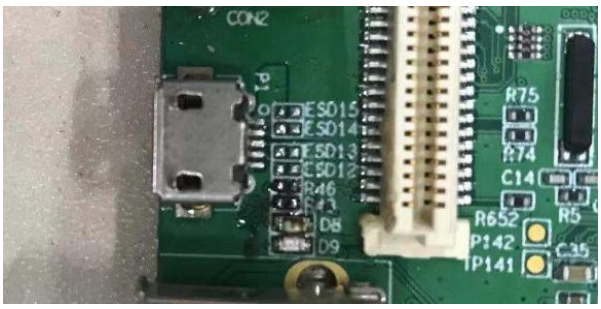
2.3. Code Path and Menuconfig
AVS SDK and SampleApp`s source code is in the tina/package/avs/.
If you want to use SampleApp, please input make menuconfig before complie and seleted all the option

3. Amazon AVS Quick Start Guide

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 3-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

3.1. Via adb

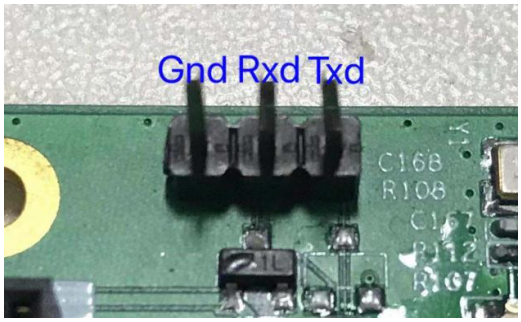


Mini-USB cable

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:

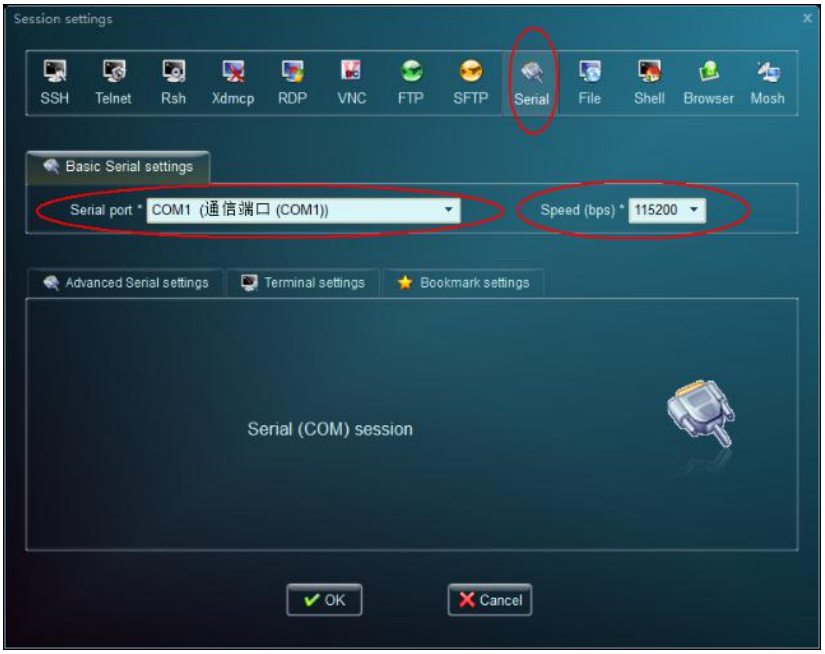
```
root@zlinj:~# adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *


BusyBox v1.27.2 () built-in shell (ash)

 _____  _              __     _
|_   _||_| ___  _ _   |  |   |_| ___  _ _  _ _
  | |   _ |   ||   |  |  |__ | ||   || | ||_'_|
  | |  | || | || _ |  |_____||_||_|_||___||_,_|
  |_|  |_|||_|_||_|_|  Tina is Based on OpenWrt!
 ------------------------------------------------
 Tina Linux (Neptune, 5B1E1071)
 ------------------------------------------------
root@TinaLinux:/#
```

3.2. Via serial port



Uart serial port

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 com. This is the setting on MobaXterm:



3.3. 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.

```
root@zlinj:~# adb push R18/config.json /etc/avs
143 KB/s (9951 bytes in 0.067s)
root@zlinj:~# adb pull /etc/avs/config.json R18/
123 KB/s (9951 bytes in 0.078s)
root@zlinj:~#
```

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 iPhone 12345678

*********************************
***Start wifi connect ap test
***ssid    :iPhone
***password:12345678
*********************************
WiFi on success!
WiFi connected ap!
update scan results enter
 1900 root      1140 S    /sbin/udhcpc -i wlan0 -h When_you_like_AW -S -T 10
 1945 root      1140 S    sh -c /etc/wifi/udhcpc_wlan0 restart
 1946 root      1304 S    {udhcpc_wlan0} /bin/sh /etc/rc.common /etc/wifi/udhc
 1945 root      1140 S    sh -c /etc/wifi/udhcpc_wlan0 restart
 1946 root      1304 S    {udhcpc_wlan0} /bin/sh /etc/rc.common /etc/wifi/udhc
wifi state busing,waiting
WiFi connected ap!

*****************************
Wifi connect ap : Success!
*****************************
root@TinaLinux:/#
```

3.4. 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 steps:

1) Developer should connect wifi first. Command: wifi_connect_ap_test "ssid" "password"
2) Before start-up AVS, you need to fill in the equipment serial number, Developer can use command below:

Fetch-device-sn /etc/avs/AlexaClientSDKConfig.json

```
root@TinaLinux:/# fetch-device-sn /etc/avs/AlexaClientSDKConfig.json
update deviceSerialNumber: 5c1004ce503442040000741000000000 to /etc/avs/AlexaClientSDKConfig.json successful!
root@TinaLinux:/#
```

3) First start-up SampleApp, need to verify equipment, start-up SampleApp use command below:

SampleApp /etc/avs/AlexaClientSDKConfig.json /etc/avs/ DEBUG9

A verification prompt will appear after the connection AVS server is successful:

```
###################################
#          NOT YET AUTHORIZED          #
###################################

#########################################################################
############################
#          To authorize, browse to: 'https://amazon.com/us/code' and enter the
code:DFVELG          #
```

```
##############################################################################
############################

##################################################
#           Checking for authorization (1)...           #
##################################################
```

Browser access : https://amazon.com/us/code, enter the Amazon account, then input the verification code.

Subsequent startup will automatically skip this step unless the configuration files in the device are deleted.

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

```
captureLoop start
2010-01-01 00:00:58.830 [  1] 5 CBLAuthDelegate:addAuthObserver:observer=0x3276ad00
2010-01-01 00:00:58.831 [  1] 5 CapabilitiesDelegate:addCapabilitiesObserver:observer=0x3276ad08
2010-01-01 00:00:58.831 [  1] 5 CapabilitiesDelegate:addCapabilitiesObserver:observer=0x326acba0
privacyMute: 0, size: 1
GpioHandler 38 disactive(0)
                #    #     #  #####      #####  ######  #    #
               # #   #     # #     #    #     # #     # #   #
              #   #  #     # #          #       #     # #  #
             #     # #     #  #####      #####  #     # ###
             #######  #   #        #          # #     # #  #
             #     #   # #   #     #    #     # #     # #   #
             #     #    #     #####      #####  ######  #    #

      #####                                      #
     #     #   ##   #    # #####  #      ######      # #   #####  #####
     #        #  #  ##  ## #    # #      #           #   #  #    # #    #
      #####  #    # # ## # #    # #      #####      #     # #    # #    #
           # ###### #    # #####  #      #          ####### #####  #####
     #     # #    # #    # #      #      #          #     # #      #
      #####  #    # #    # #      ###### ######     #     # #      #

      SDK Version 1.7.1

+-----------------------------------------------------------------------+
|                              Options:                                 |
| Wake word:                                                            |
|       Simply say Alexa and begin your query.                          |
| Tap to talk:                                                          |
|       Press 't' and Enter followed by your query (no need for the 'Alexa').|
| Hold to talk:                                                         |
|       Press 'h' followed by Enter to simulate holding a button.       |
|       Then say your query (no need for the 'Alexa').                  |
|       Press 'h' followed by Enter to simulate releasing a button.     |
| Stop an interaction:                                                  |
|       Press 's' and Enter to stop an ongoing interaction.             |
| Privacy mode (microphone off):                                        |
|       Press 'm' and Enter to turn on and off the microphone.          |
```

4) Speak “Alexa” and talk with KIT. If wake-up success, the LED will light.

3.5. Link SampleApp with your Amazon account

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.mp
3",
"alarmShortSoundFilePath":"/<absolute-path-to-short-alarm-sound>/alar
m_short.wav",
"timerSoundFilePath":"/<absolute-path-to-timer-sound>/timer_normal.mp
3",
"timerShortSoundFilePath":"/<absolute-path-to-short-timer-sound>/time
r_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.

1. 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: [1](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 toke
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.

3.6. 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/config.json

```
root@TinaLinux:/# sample-wakeup /etc/avs/config.json
2010-01-01 00:00:23.391 [  1] I sdkVersion: 1.7.1
fopen config file:/etc/avs/config.jsonpos=0flen=9951flen1=9951read len=9951new_obj=0x23a4f700
m_config_json=0x23a4f760, file_size=9951
fopen config file:/etc/openwrt_releaseauto_platform: tulip-noma
platform config:
{
  "voice-loopback":0,
  "detector":"sensory",
  "provider":"ac108",
  "filter":"tutuclear",
  "tutuclear":{
    "base":"tutuclear",
    "signal-channel-0":"ac108-0",
    "signal-channel-1":"ac108-1",
    "signal-channel-2":"ac108-3",
    "reference-channel-0":"ac108-2",
    "reference-channel-1":"",
    "type":"filter",
    "prm-file":"\/etc\/avs\/tutuClearA1_ns4wakeup_stereo.prm",
    "filter-block":160,
    "input-sample-rate":16000,
    "input-sample-bits":32,
    "input-channels":5,
    "output-sample-rate":16000,
    "output-sample-bits":32,
    "output-channels":3,
    "output-data-file":""
  },
  "mute":{
    "mute-enable":"active",
    "mute-disable":"disactive",
```

Next, speak “Alexa” to the mics, If wake up success, it will be this log:

```
captureLoop start
[TTSUNXI_GenRnd] 0x16890001
[TTSUNXI_GenCode] ioctl(ptTTSUNXIState->w32FsInfo, 1)
[TTSUNXI_GenRnd] 0x356e7573
[TTSUNXI_GenCode] grep    'sun50iw1' /sys/cla[illegible]/sunxi_info/sys_info |wc -l
[2010-01-01-08-01-47-464271] key work detect count: 0 begin: 1249920, end: 1256640
[TTSUNXI_GenRnd] 0x16890001
[TTSUNXI_GenCode] ioctl(ptTTSUNXIState->w32FsInfo, 1)
[2010-01-01-08-01-51-232801] key work detect count: 1 begin: 1309920, end: 1316160
[2010-01-01-08-01-52-833906] key work detect count: 2 begin: 1335360, end: 1341840
[2010-01-01-08-01-54-693557] key work detect count: 3 begin: 1366080, end: 1372080
[2010-01-01-08-01-57-121962] key work detect count: 4 begin: 1404000, end: 1410480
[TTSUNXI_GenRnd] 0x356e7573
[TTSUNXI_GenCode] grep    'sun50iw1' /sys/class/sunxi_info/sys_info |wc -l
[TTSUNXI_GenRnd] 0x16890001
[TTSUNXI_GenCode] ioctl(ptTTSUNXIState->w32FsInfo, 1)
[TTSUNXI_GenRnd] 0x356e7573
[TTSUNXI_GenCode] grep    'sun50iw1' /sys/class/sunxi_info/sys_info |wc -l
[TTSUNXI_GenRnd] 0x16890001
[TTSUNXI_GenCode] ioctl(ptTTSUNXIState->w32FsInfo, 1)
```

4. Audio play test

4.1. Play test（Speaker）

Developer can start playing test by following commands:

aplay -Dhw:2,0 *.wav

```
root@TinaLinux:/# aplay -Dhw:2,0 /mnt/example.wav
Playing WAVE '/mnt/example.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@TinaLinux:/#
```

4.2. Play test（Headset）

Developer need to set audio channel before playing, set audio channel commands:

amixer cset name='AIF1IN0R Mux' 'AIF1_DA0R'
amixer cset name='AIF1IN0L Mux' 'AIF1_DA0L'
amixer cset name='DACR Mixer AIF1DA0R Switch' 1
amixer cset name='DACL Mixer AIF1DA0L Switch' 1
amixer cset name='DAC volume' 160
amixer cset name='HP_R Mux' 'DACR HPR Switch'
amixer cset name='HP_L Mux' 'DACL HPL Switch'
amixer cset name='Headphone Switch' 1
amixer cset name='External Speaker Switch' 0

Developer can start playing test by following commands:

aplay -Dhw:0,0 *.wav

```
root@TinaLinux:/# aplay -Dhw:0,0 /mnt/example.wav
Playing WAVE '/mnt/example.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@TinaLinux:/#
```

5. Development environment

5.1. Hardware resource

1) A R18 development board
2) Power adapter, USB to serial port, micro-usb line
3) PC and compiler server

5.2. Software resource

1) The compiler server must only use the system of Linux 64bit. It is recommended to install ubuntu12.04 64bit.
2) The packages that need to be installed as follows:

Gcc, binutils, bzip2, flex, python, perl, make, ia32-libs, find, grep, diff, unzip, gawk, getopt, subversion,Libz-dev, libc headers.

Ubuntu can directly install the following commands:

Sudo apt-get install build-essential subversion git-core libncurses 5-dev zlib1g-dev gawk flex quilt libssl-dev

3) firmware burning tools: LiveSuit (Linux) or PhonixSuit (window).

5.3. Code compilation and packaging

$ source build/envsetup.sh

=> Setting environment variables

$ lunch

=> Select scheme. See below

$ make [-jN]

=> Compile, -jN: Parameter select the number of parallel compilation process.

$ pack [-d]

=> pack, -d: The serial port information of the firmware package is transferred to the TF card output.

Option Description:

R18 development plan => tulip_*

After the compilation is completed. The system image will be packaged in the out/<board>/ directory.

6. Flash image

6.1. Necessary tools

1) PhoenixSuit--------Flash tool on windows
2) LiveSuit----------Flash tool on Ubuntu
3) adb driver
4) mini-USB cable
5) tina_tulip-noma_uart0.img----------sdk image

6.2. Flash Steps

1) install the software, please see Livesuit.doc and R18_PhoenixSuitUserManual_V1.1.pdf for more information.

2) 2. download the image from our website, select the image in PhoenixSuit and Livesuit.
3) 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.