

DEVELOPMENT GUIDE

AMOS-820

VAB-820

Android BSP 3.1.0

Copyright

Copyright © 2015 VIA Technologies Incorporated. All rights reserved.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of VIA Technologies, Incorporated.

Trademarks

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

Disclaimer

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right to make changes to the products described in this manual at any time without prior notice.

Revision History

Version	Date	Remarks
1.00	2015/2/25	Initial version
1.01	2015/3/3	Corrected wrong FSL patch version
1.02	2015/3/11	TIC reviewed
1.03	2015/3/26	Corrected “boot selection jumper” picture
1.04	2015/3/30	Corrected light LVDS panel argument
1.05	2015/4/15	1. Rename to AMOS-820 VAB-820 Android BSP V3.1.0 Development Guide 2. Corrected “1. Introduction” description 3. Corrected “2.2.2. Decompress Freescale BSP” description

Table of Contents

1. Introduction.....	1
1.1. Overview.....	1
1.2. Package Content	1
1.2.1. BSP Folder Contents	2
1.2.2. EVK Folder Contents	2
1.2.3. Document Folder Contents	3
2. Setup Building Environment.....	4
2.1. Configure Ubuntu host computer	4
2.1.1. Install the Java6 JDK	4
2.1.2. Install the Packages	4
2.2. Download the Source Code	5
2.2.1. Download Freescale i.MX6 Android BSP	5
2.2.2. Decompress Freescale BSP	6
3. Building BSP	7
3.1. Preparing	7
3.1.1. Get Google Android Source Code	7
3.1.2. Add VAB-820 Support.....	8
3.2. Building VAB-820 BSP	8
3.2.1. Compiling	8
4. Making Android System Booting Media.....	10
4.1. Requirements	10
4.2. Procedure	10
4.3. Setting the display device.....	15

5. Source Code Modification	19
5.1. ADV7180 support	19
5.1.1. Kernel support	19
5.1.2. Android support	21
5.1.2.1. MMAP	21
5.1.2.2. Video quality support	23
5.1.2.3. S-video switch	24
5.1.2.4. Snapshot crash	25
5.2. LVDS support	25
5.2.1. Add LVDS mode	25
5.2.2. Touch Panel Calibrator	26
5.3. Auto-Mount External Storage Device	28
5.4. I2C-Tool	29
5.5. Bootloader	30
5.5.1. Add bootloader argument	30
5.5.2. Uboot print message	31
5.5.3. Uboot support microSD and eMMC	31
5.5.4. Uboot set gpio	33
5.5.5. U-boot set 2G RAM	33
5.5.6. U-boot set factory data reset	36
5.6. COM1 support flow control	36
5.7. Add support sgtl5000	37
5.7.1. Use tinymix to change audio parameter	38
5.7.2. support sgtl5000 sound card	38
5.7.3. choose sound card from uboot	42
5.8. Add usb and usb otg keyboard/mouse resume	43
5.9. Add support of VNT9271	43

5.10.	Ethernet Static IP	50
5.10.1.	Gateway	50
5.10.2.	Netmask	51
5.10.3.	Settings record	51
5.10.4.	Enable service when booting	53
5.10.5.	Check Input format	54
5.10.6.	Send message to inform Ethernet.apk	55
5.10.7.	Set the default value	56
5.11.	Set WiFi as the preferred network	58
5.12.	Fix USB device detection	60
5.13.	Add SPI ROM	60
5.14.	Add support of CAN bus	64
5.15.	Add support of 3G	67
5.16.	Add support of GPS	71
5.17.	Add eMMC 5.0 running on eMMC 4.4 mode	73
5.18.	Fix IPU error under thermal testing	73
5.19.	Fix 3G incorrect signal strength info	73
5.20.	Fix failure in opening the camera apk	74
5.21.	Support AUO 10.4" touch panel	74
5.22.	Correct navigation bar in low profile	76
5.23.	Correct kernel menuconfig that is not working	76
5.24.	Fix camera recording issue	76
5.25.	Add omxplayer package	77
5.26.	Fix volume limit issue	77
5.27.	Fix hotspot issue	78
5.28.	Delete si_LK language	80
5.29.	Set default sleep time to 30 min	80

5.30. Add DualLite support 80

List of Figures

Figure 1 VAB-820 package content..... 1

Figure 2 BSP folder contents.....2

Figure 3 Source Code download link.....5

Figure 4 Boot select jumper 12

Figure 5 Boot select jumper 14

Figure 6 u-boot operation environment 14

Figure 7 LVDS power selection jumper 16

Figure 8 LVDS power selection jumper 17

List of Tables

Table 1 LVDS power selection jumper 16

Table 2 LVDS power selection jumper 16

1. Introduction

The purpose of this document is to provide a practical introduction on developing software for the AMOS-820 (Bare board: VAB-820 Freescale i.MX6 Quad) on a Linux development host only.

1.1. Overview

The VIA VAB-820 platform is an embedded system powered by ARM processor with Android 4.4.2 operating system by default. VAB-820's Android BSP(Board Support Package) offers a development package. Users can develop it under an Ubuntu environment.

There are four major boot components for Android, the **"u-boot.bin"**, **"boot.img"**, **"system.img"** and **"recovery.img"**. The system will not boot successfully into an Android environment if one of these files does not exist in the boot media.

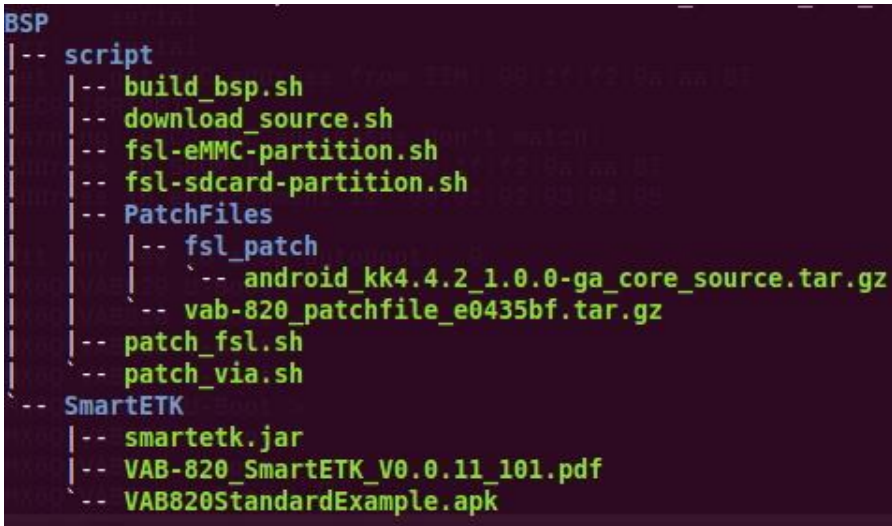
1.2. Package Content

There are three folders in VAB-820 package.



Figure 1 VAB-820 package content

1.2.1. BSP Folder Contents



```
BSP
-- script
| -- build_bsp.sh
| -- download_source.sh
| -- fsl-eMMC-partition.sh
| -- fsl-sdcard-partition.sh
| -- PatchFiles
| | -- fsl_patch
| | | -- android_kk4.4.2_1.0.0-ga_core_source.tar.gz
| | | -- vab-820_patchfile_e0435bf.tar.gz
| -- patch_fsl.sh
| -- patch_via.sh
-- SmartETK
| -- smartetk.jar
| -- VAB-820_SmartETK_V0.0.11_101.pdf
| -- VAB820StandardExample.apk
```

Figure 2 BSP folder contents

- **script**: Scripts for downloading the source code, patching, building, and installing the image. There are several necessary patch files for VAB-820 in *script/PatchFiles* folder.

1.2.2. EVK Folder Contents

- **app**: android apk files for evaluation only.
- **Tool**: the drivers when user would like to use android-sdk through USB-OTG.
- **Images_autoinstall_sd**:

- **scripts:** `mk_install_sd.sh`. User can execute this script to make a VAB-820 linux bootable Micro SD storage card, in order to install Android evaluation image to VAB-820 on board eMMC storage.
- **out:**
 - Image files “**u-boot.bin**”, “**ulmage**”, “**rootfs.tar.bz2**” to make a Linux bootable Micro SD card.
 - **img_android:** Android image files, “**u-boot.bin**”, “**boot.img**”, “**system.img**”, “**recovery.img**” and `fsl-eMMC-partition.sh`.

1.2.3. Document Folder Contents

- Development Guide
- User’s Manual



Note:

If a user needs the supporting files for all the software mentioned in VAB-820 Android BSP document, please contact our regional sales representative for assistance.

2. Setup Building Environment

This chapter will guide you through setting up the developing environment. All the instructions in this guide are for Ubuntu 12.04 (64Bit). Please install the Ubuntu to your PC/NB in advance.

2.1. Configure Ubuntu host computer

In order to develop Android under Linux, user has to install the related tools. For more information, user can refer to [Google Android website](http://source.android.com/source/initializing.html) (<http://source.android.com/source/initializing.html>).

2.1.1. Install the Java6 JDK

Default source list in Ubuntu 12.04 does not provide Java 6 JDK. User has to add a new server manually.

```
user@ubuntu:~$ sudo add-apt-repository ppa:webupd8team/java
user@ubuntu:~$ sudo apt-get update
user@ubuntu:~$ sudo apt-get install oracle-java6-installer
```

2.1.2. Install the Packages

```
user@ubuntu:~$ sudo apt-get install git gnupg flex bison gperf build-essential zip curl zlib1g-dev libc6-dev libncurses5-dev x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386 uuid-dev liblzo2-dev libz-dev ia32-libs libc6-dev-i386 lib32z1
```

2.2. Download the Source Code

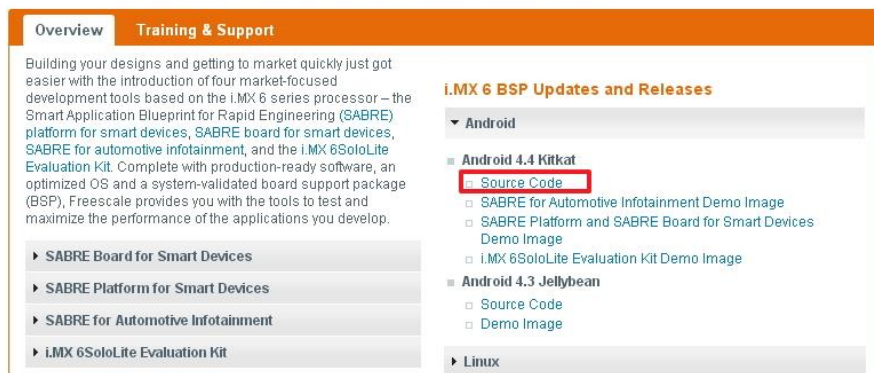
The VAB-820 Android BSP is developed based on the BSP released by Freescale: “**android_kk4.4.2_1.0.0-ga**” of *i.MX6 SABRE board for smart devices*. Users can get it from Freescale official web site.

2.2.1. Download Freescale i.MX6 Android BSP

The i.MX6 Android source bundle can be found and downloaded from here:

https://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX6_SW

IMX6_SW: i.MX 6 Series Software and Development Tool Resources ☆



Overview **Training & Support**

Building your designs and getting to market quickly just got easier with the introduction of four market-focused development tools based on the i.MX 6 series processor – the Smart Application Blueprint for Rapid Engineering (SABRE) platform for smart devices, SABRE board for smart devices, SABRE for automotive infotainment, and the i.MX 6SoloLite Evaluation Kit. Complete with production-ready software, an optimized OS and a system-validated board support package (BSP), Freescale provides you with the tools to test and maximize the performance of the applications you develop.

- ▶ SABRE Board for Smart Devices
- ▶ SABRE Platform for Smart Devices
- ▶ SABRE for Automotive Infotainment
- ▶ i.MX 6SoloLite Evaluation Kit

i.MX 6 BSP Updates and Releases

- ▼ Android
 - Android 4.4 Kitkat
 - ▢ **Source Code**
 - ▢ SABRE for Automotive Infotainment Demo Image
 - ▢ SABRE Platform and SABRE Board for Smart Devices Demo Image
 - ▢ i.MX 6SoloLite Evaluation Kit Demo Image
 - Android 4.3 Jellybean
 - ▢ Source Code
 - ▢ Demo Image
- ▶ Linux

Figure 3 Source Code download link

After downloading the “**Source Code**”, user can get an archived file called “**android_kk4.4.2_1.0.0-ga_core_source.gz**”.

2.2.2. Decompress Freescale BSP

User can decompress “**android_kk4.4.2_1.0.0-ga_core_source.gz**” by using the following command.

```
user@ubuntu:~$ gunzip android_kk4.4.2_1.0.0-ga_core_source.gz
user@ubuntu:~$ mv android_kk4.4.2_1.0.0-ga_core_source
  android_kk4.4.2_1.0.0-ga_core_source.tar
user@ubuntu:~$ tar xvf android_kk4.4.2_1.0.0-ga_core_source.tar
```

User will get a folder: **android_kk4.4.2_1.0.0-ga_core_source**

The file “**kk4.4.2_1.0.0-ga.tar.gz**” which is located at “**android_kk4.4.2_1.0.0-ga_core_source/code/**” includes iMX6 patch files for Android 4.4.2. User has to put this file to **path/script/PatchFiles/fsl_patch** for BSP compiling.

3. Building BSP

Once the building environment has been configured as described in Chapter 2, the development environment is ready to compile. This chapter will guide you through building the BSP by script which was obtained from VAB-820 BSP.

3.1. Preparing

Make sure that you have the necessary files as shown in Figure 2 under **BSP/script**.

There are three steps to build VAB-820 BSP:

1. Download Android 4.4.2 source code.
2. Patch source code for supporting VAB-820.
3. Compile

3.1.1. Get Google Android Source Code

Execute download script

```
user@ubuntu:~$ cd BSP/script
user@ubuntu:~/BSP/script$ ./download_source.sh
```

It will start to download source code to **BSP/src/ (created by download_source.sh)**. This step will take several hours.



Note:

The original download link of repo in Freescale document is

<https://dl-ssl.google.com/dl/googlesource/git-repo/repo>

But this link is broken since Oct. 2013. User has to use the other download link in
download_source.sh.

3.1.2. Add VAB-820 Support

Add Freescale and VAB-820 support:

```
user@ubuntu:~/BSP/script$ ./patch_fsl.sh
user@ubuntu:~/BSP/script$ ./patch_via.sh
```

These scripts will apply Freescale patches and vab-820 patches.

3.2. Building VAB-820 BSP

The source should be ready when user has completed the steps in Ch 3.1, now user can compile the BSP.

3.2.1. Compiling

User can compile the VAB-820 BSP.

```
user@ubuntu:~$ cd BSP/script
user@ubuntu:~/BSP/script$ ./build_bsp.sh
```

Then it will start compiling process, this process will take several hours.

Finally, the Android image files will be generated in

BSP/src/out/target/product/sabresd_6dq/ and we had copied the necessary files to **BSP/out/**:

boot.img recovery.img system.img u-boot.bin

Compiling parameter of build_esp.sh:

--build-choice:

uboot

kernel

android

all

--lunch-type:

eng: development configuration with additional debugging tools

(default)

user: limited access; suitable for production

--cpus:

Default value is 4

Example:

```
./build_esp.sh --build-choice=android --cpus=1 --lunch-type=user
```

4. Making Android System Booting Media

VAB-820 Android BSP supports booting from on board eMMC by default. This section will guide you through making the Android system boot media for VAB-820.

4.1. Requirements

- Linux development computer
- Micro SD storage card. Recommended size is 4GB, class 4 at least.
- EVK folder

4.2. Procedure

Step 1: Insert the Micro SD card to Linux development computer.

Step 2: Open “Terminal” utility.

Step 3: The Micro SD storage card can be identified and auto mounted once inserted to the computer. You can check Micro SD card code name by `$ df -h`

```
user@ubuntu:~$ df -h
...
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1       7.2G  531M   6.3G   8%  /media/usb
```

However, there are some instances that the Micro SD storage card cannot be identified or auto mounted after inserting to the computer. In that case, you can try the other way to identify the Micro SD card.

```
user@ubuntu:~$ dmesg | grep -i removable  
[105.502517] sd 9:0:0:0 [sdb] Attached SCSI removable disk
```

The Micro SD storage card code name for this example is identified as **sdb**.

Step 4: The EVK path for this example is /home/user/EVK, user can enter the folder “Images_autoinstall_sd/scripts” which is under EVK.

```
user@ubuntu:~$ sudo cd /home/user/EVK/Images_autoinstall_sd/scripts  
user@ubuntu:~/EVK/Images_autoinstall_sd/scripts$ ls  
mk_install_sd.sh
```

Step 5: User can run the script “**mk_install_sd.sh**” to make a bootable Micro SD card. Thus, user can install the Android image into VAB-820 eMMC through the Micro SD card. The target Micro SD storage card code name for this example is identified as **sdb**.

```
user@ubuntu:~/EVK/Images_autoinstall_sd/scripts$ sudo ./mk_install_sd.sh  
/dev/sdb
```

User can unplug the Micro SD card from Linux development PC after the message “Done!” appears.

Step 6: Insert the Micro SD card into VAB-820. Make sure to set the jumper setting to make it bootable from Micro SD. For more details about the boot select jumper, please refer to the VAB-820 User Manual.

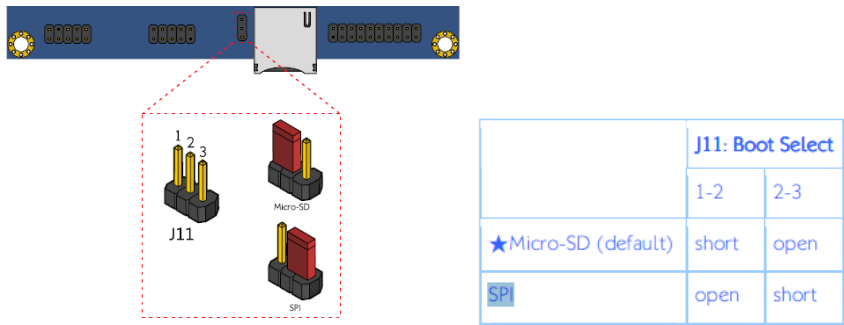


Figure 4 Boot select jumper

**Note:**

- 1) It is recommended to connect COM2 debug cable to make sure the u-boot parameter is correct.
- 2) Make sure the u-boot parameter is clean. User can type the command below in u-boot:
u-boot> destroyenv
u-boot> reset

Here, the VAB-820 should boot into linux.

Step 7: Type this command to format/partition/flash Android image into SPI/eMMC.

```
$/fsl-eMMC-partition.sh -f /dev/mmcblk0
```

The u-boot will flash into SPI ROM and the others will flash into eMMC.

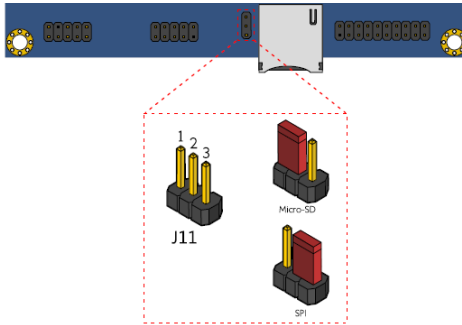
Shutdown linux by using this command:

```
$poweroff
```

Step 8: Set jumper J11 (2-3 short) to SPI booting selection. Then power on the VAB-820 and set uboot parameter through COM2 debug port.

Make sure the u-boot parameter is clean. User can type below command in u-boot:

```
u-boot> destroyenv  
u-boot> reset
```



	J11: Boot Select	
	1-2	2-3
★Micro-SD (default)	short	open
SPI	open	short

Figure 5 Boot select jumper

Step 9: Connect the VAB-820 and host PC through J5 (COM 2) of VAB-820. Run “putty” on host PC to receive the booting message. Power on the VAB-820 and press any key to stop the booting process as shown in Figure4.

```
COM4 - PuTTY
U-Boot 2009.08 (Oct 18 2013 - 09:36:30)

CPU: Freescale i.MX6 family T01.2 at 792 MHz
Thermal sensor with ratio = 180
Temperature: 23 C, calibration data 0x5734ed69
mx6q pll1: 792MHz
mx6q pll2: 528MHz
mx6q pll3: 480MHz
mx6q pll8: 50MHz
ipg clock : 660000000Hz
ipg per clock : 660000000Hz
uart clock : 800000000Hz
cpbl clock : 600000000Hz
ahb clock : 1320000000Hz
axi clock : 2640000000Hz
emi_slow clock: 1320000000Hz
ddr clock : 5280000000Hz
usdhc1 clock : 1980000000Hz
usdhc2 clock : 1980000000Hz
usdhc3 clock : 1980000000Hz
usdhc4 clock : 1980000000Hz
nfc clock : 240000000Hz
Board: MX6Q-VAB820:[ POR]
Boot Device: I2C
I2C: ready
DRAM: 1 GB
MMC: FSL_USDHC: 0,FSL_USDHC: 1,FSL_USDHC: 2,FSL_USDHC: 3
JEDEC ID: 0xbff:0x25:0x4a
Reading SPI NOR flash 0xc0000 [0x2000 bytes] -> ram 0x27600c48
SUCCESS

*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: got MAC address from IIM: 00:00:00:00:00:00
FE00 [PRIME]
Hit any key to stop autoboot: 0
MX6Q VAB820 U-Boot >
```

Figure 6 u-boot operation environment

4.3. Setting the display device

[HDMI]

To set HDMI as boot display:

```
setenv display run hdmi
```

To set HDMI resolution:

Default resolution is 1920x1080, change the red part with the desired resolution mode which is defined in variable hdmi.

For example: To set 1280 x 720, change 1920 x 1080 to 1280 x 720.

```
setenv hdmi 'setenv bootargs ${bootargs}  
video=mxcfb0:dev=hdmi,1280x720M@60,bpp=32 video=mxcfb1:off  
video=mxcfb2:off video=mxcfb3:off fbmem=28M vmalloc=400M  
androidboot.hardware=freescale audio=hdmi'
```

[LVDS]

There are three LVDS panels supported by VAB-820. Here is the list:

- AUO 22" G220SVN01.0
- AUO 10.4" G104XVN01.0
- AUO 7" G070VW01 V0

The LVDS type for this example is AUO 22" G220SVN01.0.

Check the LVDS power selection setting:

J3 : LVDS power select

IVDD		PVDD	
*1-3	+12V	2-4	+3.3V
3-5	+5V	*4-6	+5V

Table 1 LVDS power selection jumper

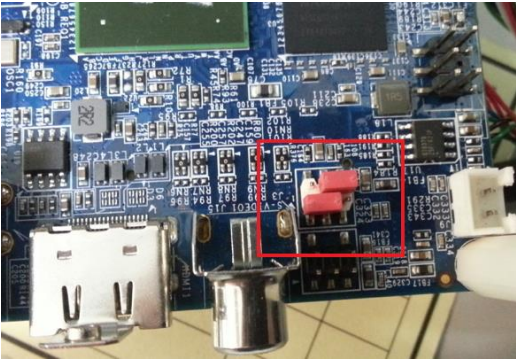


Figure 7 LVDS power selection jumper

To set LVDS AUO 22" G220SVN01.0 as display device:

```
setenv display run lvds auo g220
```

The LVDS type for this example is AUO 10.4" G104XVN01.0.

Check the LVDS power selection setting:

J3 : LVDS power select			
IVDD		PVDD	
*1-3	+12V	*2-4	+3.3V
3-5	+5V	4-6	+5V

Table 2 LVDS power selection jumper

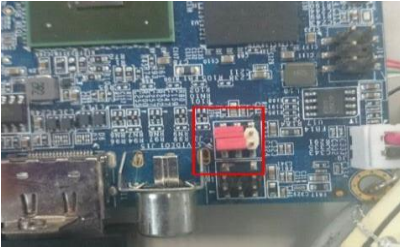


Figure 8 LVDS power selection jumper

To set LVDS AUO 10.4" G104XVN01.0 as display device:

```
setenv display run lvds auo g140
```

To set the other LVDS panels, user can add a customized lvds parameter when finishing porting lvds driver(ldb.c). Here is an example; the “red” part is changeable depending on the LVDS SPEC.

```
setenv lvds custmize 'setenv bootargs ${bootargs}
video=mxcfb0:dev=ldb,LDB-WSXGA+,if=RGB24,bpp=32 ldb=sp10
video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off fbmem=28M
androidboot.hardware=freescale audio=sgtl5000'
```

```
setenv display run lvds custmize
```

[DUAL DISPLAY]

To set Dual display (auo 10.4" + hdmi 720p):

```
setenv display run dual_display
```

To set other configurations of dual display, here is an example: the “red” part is changeable depending on the LVDS SPEC and HDMI mode that user wants to set.

```
setenv dual_display 'setenv bootargs ${bootargs}
video=mxcfb0:dev=ldb,LDB-XGA,if=RGB24,bpp=32 ldb=sin0
video=mxcfb1:dev=hdm1,1280x720M@60,bpp=32 video=mxcfb2:off
video=mxcfb3:off fbmem=28M vmalloc=400M androidboot.hardware=freescale
audio=sgtl5000'
```

```
setenv display run dual display
```

5. Source Code Modification

This section will describe source codes that have been modified.

5.1. ADV7180 support

Original kernel has adv7180 driver which is unused. We must set its pin definition and enable the related data path.

5.1.1. Kernel support

We setup ADV7180 pin definition in this file.

- `src/kernel_imx/arch/arm/mach-mx6/board-mx6q_sabrelite.c`

```
static iomux_v3_cfg_t mx6q_sabrelite_pads[] = {
...(skip)

    /* I2C1 - ADV7180 */
    MX6Q_PAD_CSI0_DAT8__I2C1_SDA,
    MX6Q_PAD_CSI0_DAT9__I2C1_SCL,

...(skip)

    /* CSI - ADV7180 */
    // (Sylvia) added
    MX6Q_PAD_CSI0_DAT12__IPU1_CSI0_D_12,
    MX6Q_PAD_CSI0_DAT13__IPU1_CSI0_D_13,
    MX6Q_PAD_CSI0_DAT14__IPU1_CSI0_D_14,
    MX6Q_PAD_CSI0_DAT15__IPU1_CSI0_D_15,
    MX6Q_PAD_CSI0_DAT16__IPU1_CSI0_D_16,
    MX6Q_PAD_CSI0_DAT17__IPU1_CSI0_D_17,
    MX6Q_PAD_CSI0_DAT18__IPU1_CSI0_D_18,
    MX6Q_PAD_CSI0_DAT19__IPU1_CSI0_D_19,
    MX6Q_PAD_CSI0_VSYNC__IPU1_CSI0_VSYNC,
    MX6Q_PAD_CSI0_MCLK__IPU1_CSI0_HSYNC,
    MX6Q_PAD_CSI0_PIXCLK__IPU1_CSI0_PIXCLK,

    MX6Q_PAD_SD1_DAT0__GPIO_1_16, // CSI0_PWN
    MX6Q_PAD_SD1_DAT1__GPIO_1_17, // CSI0_RST_B
}
```

Setup ADV7180 when booting.

```
// (Sylvia) copy from board-mx6q_sabreauto.c
static void adv7180_pwn(int powdn)
```

```

{
    // (Sylvia) copied from mx6q_csi0_cam_powerdown()
    if (powdn)
        gpio_set_value(MX6Q_SABRELITE_CSI0_PWN, 1);
    else
        gpio_set_value(MX6Q_SABRELITE_CSI0_PWN, 0);

    msleep(2);
}

// (Sylvia) copy from board-mx6q_sabreauto.c
static struct fs1_mxc_tvin_platform_data adv7180_data = {
    .dvddio_reg    = NULL,
    .dvdd_reg      = NULL,
    .avdd_reg      = NULL,
    .pvdd_reg      = NULL,
    .pwn           = adv7180_pwn,
    .reset         = NULL,
    .cvbs          = true,
    .io_init       = mx6q_csi0_io_init,
};

...(skip)

static struct i2c_board_info mxc_i2c0_board_info[] __initdata = {

    {
        // (Sylvia) copy from board-mx6q_sabreauto.c
        I2C_BOARD_INFO("adv7180", 0x21),
        .platform_data = (void *)&adv7180_data,
    },

};

```

We added two input sources, CVBS and S-video, instead of “CSI MEM”

- `src/kernel_imx/drivers/media/video/mxc/capture/adv7180.c`

```

static int ioctl_s_parm(struct v4l2_int_device *s, struct
v4l2_streamparm *a)
{
...(skip)
    //Select input from CVBS or S-Video
    switch (a->parm.raw_data[0]) {
    case 1:
        /* CVBS input on ANT3 */
        //insel = true;
        //adv7180_hard_reset(insel);
        adv7180_write_reg(ADV7180_INPUT_CTL, 0x04);
        pr_err ("ADV7180: Set input to CVBS\n");
        break;
    case 2:
        /* S-Video input, Y on ANT1, C on ANT2 */
        //insel = false;
        //adv7180_hard_reset(insel);

```

```

        adv7180_write_reg(ADV7180_INPUT_CTL, 0x06);
        pr_err("ADV7180: Set input to S-Video\n");
        break;
    default:
        pr_err("ADV7180: Unkown input select, set nothing\n");
        break;
    }
    ... (skip)
}

```

- `src/kernel_imx/drivers/media/video/mxc/capture/mxc_v4l2_capture.c`

```

static struct v4l2_input mxc_capture_inputs[MXC_V4L2_CAPTURE_NUM_INPUTS]
= {
    ... (skip)
    {
        .index = 1,
        .name = "CVBS", //changed by Dylan, default "CSI MEM"
        .type = V4L2_INPUT_TYPE_CAMERA,
        .audioset = 0,
        .tuner = 0,
        .std = V4L2_STD_UNKNOWN,
        .status = V4L2_IN_ST_NO_POWER,
    },
    { //Added by Dylan, add S-Video input
        .index = 2,
        .name = "S-VIDEO",
        .type = V4L2_INPUT_TYPE_CAMERA,
        .audioset = 0,
        .tuner = 0,
        .std = V4L2_STD_UNKNOWN,
        .status = V4L2_IN_ST_NO_POWER,
    },
};

```

You can search the keywords “CVBS” and “S-video” to find the settings of CVBS and S-video.

5.1.2. Android support

5.1.2.1. MMAP

After patch kernel, we still can’t see the image as HAL needs mmap.

- `src/hardware/imx/mx6/libcamera2/DeviceAdapter.h`

```
private:
    int         deviceThread();
    int         autoFocusThread();
    CameraFrame mcaptureBuffers[MAX_PREVIEW_BUFFER]; // Dean
```

● src/hardware/imx/mx6/libcamera2/DeviceAdapter.c

```
status_t DeviceAdapter::registerCameraBuffers(CameraFrame *pBuffer,
                                              int          & num)
{
    ... (skip)

    memset(mcaptureBuffers, 0, sizeof(mcaptureBuffers)); //Dean
    for (int i = 0; i < num; i++) {
        CameraFrame *buffer = pBuffer + i;
        memset(&buf, 0, sizeof (buf));
        buf.index      = i;
        buf.type       = V4L2_BUF_TYPE_VIDEO_CAPTURE;
        buf.memory     = V4L2_MEMORY_MMAP; // Dean
        buf.m.offset   = buffer->mPhyAddr;
        buf.length     = buffer->mSize;

        ret = ioctl(mCameraHandle, VIDIOC_QUERYBUF, &buf);
        if (ret < 0) {
            FLOGE("Unable to query buffer (%s)", strerror(errno));
            return ret;
        }
        // Associate each Camera buffer
        buffer->setObserver(this);
        mDeviceBufs[i] = buffer;

        /* ---Add by Dean--- */
        mcaptureBuffers[i].mSize = buf.length;
        mcaptureBuffers[i].mPhyAddr = (size_t)buf.m.offset;
        mcaptureBuffers[i].mVirtAddr = (void *)mmap(NULL,
            mcaptureBuffers[i].mSize,
            PROT_READ | PROT_WRITE, MAP_SHARED, mCameraHandle,
            mcaptureBuffers[i].mPhyAddr);
        mcaptureBuffers[i].mFormat = HAL_PIXEL_FORMAT_YCbCr_420_SP;
        mcaptureBuffers[i].mWidth = buffer->mWidth;
        mcaptureBuffers[i].mHeight = buffer->mHeight;
        memset(mcaptureBuffers[i].mVirtAddr, 0xFF,
            mcaptureBuffers[i].mSize);
        /* ---End--- */
    }
    ... (skip)
}

... (skip)

CameraFrame * DeviceAdapter::acquireCameraFrame()
{
    ... (skip)
```

```

int index = cfilledbuffer.index;
fAssert(index >= 0 && index < mBufferCount);
mDeviceBufs[index]->mTimeStamp = systemTime();

memcpy(mDeviceBufs[index]->mVirtAddr,
mcaptureBuffers[index].mVirtAddr, mDeviceBufs[index]->mSize); //Dean

return mDeviceBufs[index];
}

```

5.1.2.2. Video quality support

When we switch from photo mode to video mode in camera, camera will crash. The reason is that adv7180 uses NTSC(720x480) or PAL(720x576) but media_profile doesn't support these sizes.

- src/device/fsl-proprietary/media_profile/media_profiles_1080p.xml

We define quality_high for PAL and quality_low for NTSC.

```

-->
<!-- Dean: add for adv7180 camera. high for PAL and low for NTSC
-->
<EncoderProfile quality="high" fileFormat="mp4" duration="30">
  <Video codec="h264"
    bitRate="3000000"
    width="720"
    height="576"
    frameRate="30" />

  <Audio codec="aac"
    bitRate="96000"
    sampleRate="44100"
    channels="1" />

  <Camera previewFrameRate="30" />
  <EncoderOutputFileFormat name="mp4" />
</EncoderProfile>

<EncoderProfile quality="low" fileFormat="mp4" duration="30">
  <Video codec="h264"
    bitRate="3000000"
    width="720"
    height="480"
    frameRate="30" />

  <Audio codec="aac"
    bitRate="96000"

```

```

        sampleRate="44100"
        channels="1" />

        <Camera previewFrameRate="30" />
            <EncoderOutputFileFormat name="mp4" />
        </EncoderProfile>
    <!-- End -->

```

- src/device/fsl/sabresd_6dq/init.rc

We set default quality as low for NTSC.

```

# set ADV7180 camera format. 1:NTSC, 2:PAL
setprop adv7180 fmt 1

```

- src/packages/apps/Camera2/src/com/android/camera/VideoModule.java

We load this setting in camera.

```

        quality = CamcorderProfile.QUALITY_LOW;
    }
}

/* Fixme:
 * Camera will hang when switch to video mode with uvc camera.
 */
quality = 1; // temp solution

// Set video duration limit. The limit is read from the preference,
// unless it is specified in the intent.

```

5.1.2.3. S-video switch

Freescall adds a patch file that can be auto-detected input source in android 4.4.2.

5.1.2.4. Snapshot crash

Camera will crash when we snapshot because the formula of alignment is wrong.

- src/hardware/imx/mx6/libcamera2/PhyMemAdapter.cpp

The original value is 16.

```
switch (format) {  
    case HAL_PIXEL_FORMAT_YCbCr_420_SP:  
        // Dean: correct wrong align formula.  
        size = width * ((height + 15) & (~15)) * 3 / 2;  
        break;
```

5.2. LVDS support

5.2.1. Add LVDS mode

Add 3 modes 800x480, 1024x768, and 1680x1050 for LVDS.

- src/kernel_imx/drivers/video/mxc/ldb.c

```
static struct fb_videomode ldb_modedb[] = {  
    {  
        "480C60", 60, 800, 480, 29850,  
        89, 164,  
        23, 10,  
        10, 10,  
        0,  
        FB_VMODE_NONINTERLACED,  
        FB_MODE_IS_DETAILED,  
    },  
    ...  
    {  
        "LDB-XGA", 60, 1024, 768, 15385,  
        220, 40,  
        21, 7,  
        60, 10,  
        0,  
        FB_VMODE_NONINTERLACED,  
        FB_MODE_IS_DETAILED,  
    },  
    ...  
}
```

```

        "LDB-WSXGA+", 60, 1680, 1050, KHZ2PICOS(146250),
        280, 104,
        30, 3,
        176, 6,
        0,
        FB_VMODE_NONINTERLACED,
        FB_MODE_IS_DETAILED,
    },
};

```

5.2.2. Touch Panel Calibrator

We would like the touch panel to run calibrator when the system starts for the first time.

- src\device\fsl\imx6\imx6.mk

Add

```

PRODUCT_COPY_FILES += \
device/fsl/common/input/Vendor_0eef_Product_0001.idc:system/usr/idc/eGalax
Touch_VirtualDevice.idc \
device/fsl/imx6/etc/eGTouchA.ini:data/eGTouchA.ini \
device/fsl/imx6/etc/eGTouchD:system/bin/eGTouchD \

```

Our touch panel uses USB cable, when we plug in the cable, sensor service will detect it as an input device, but the device name is too long and it will cause an error.

- src\hardware\imx\libsensor\SensorBase.cpp

```

int SensorBase::sensorBaseGetSysfsPath(const char* inputName)
{
    FILE *fd = NULL;
    const char *dirname = "/sys/class/input/";
    char sysfs_name[PATH_MAX], *endptr;
    char *filename = NULL, buf[80];
    (skip...)
    if (fd) {
        memset(buf, 0, 80);
        n = fread(buf, 1, 80, fd);
        (skip...)
    }
}

```

Modify kernel code by eGalax document

- `src\kernel_imx\imx\drivers\input\evdev.c`

```
static bool evdev_match(struct input_handle *handle, struct input_dev
*dev)
{
    #define VID_EETI 0x0EEF
    if((BUS_USB == dev->id.bustype) && (VID_EETI==dev->id.vendor))
        return false;
    return true;
}
static struct input_handler evdev_handler = {
    .event          = evdev_event,
    .match          = evdev_match,
    .connect        = evdev_connect,
    .disconnect     = evdev_disconnect,
    .fops           = &evdev_fops,
    .minor          = EVDEV_MINOR_BASE,
    .name           = "evdev",
    .id_table       = evdev_ids,
};
```

- `src\kernel_imx\imx\drivers\input\joydev.c`

```
static bool joydev_match(struct input_handler *handler, struct input_dev
*dev)
{
    /* Avoid touchpads and touchscreens */
    if (test_bit(EV_KEY, dev->evbit) && test_bit(BTN_TOUCH, dev->keybit))
        return false;

    /* Avoid tablets, digitisers and similar devices */
    if (test_bit(EV_KEY, dev->evbit) && test_bit(BTN_DIGI, dev->keybit))
        return false;

    #define VID_EETI 0x0EEF
    if((BUS_VIRTUAL == dev->id.bustype) && (VID_EETI==dev->id.vendor))
        return false;
    return true;
}
```

- `src\kernel_imx\imx\drivers\input\mousedev.c`

```
static bool mousedev_match(struct input_handle *handle, struct input_dev
*dev)
{
    #define VID_EETI 0x0EEF
    if((BUS_USB == dev->id.bustype) && (VID_EETI==dev->id.vendor))
        return false;
    if((BUS_VIRTUAL == dev->id.bustype) && (VID_EETI==dev-
>id.vendor))
```

```

        return false;
    }
    return true;
}

static struct input_handler mousedev_handler = {
    .event =          mousedev_event,
    .match =          mousedev_match,
    .connect =        mousedev_connect,
    .disconnect =      mousedev_disconnect,
    .fops =            &mousedev_fops,
    .minor =           MOUSEDEV_MINOR_BASE,
    .name =             "mousedev",
    .id_table =         mousedev_ids,
};

```

5.3. Auto-Mount External Storage Device

USB storage is mounted to /sdcard/udisk

Micro SD card is mounted to /sdcard/extsd

Media partition of eMMC is mounted to /sdcard

Auto-Mount of SATA HD does not work now.

- src/device/fs/sabresd_6dq/vold.fstab

```

#mount SDHC4 SD card /mnt/sdcard as primary storage forMX6Q SABER_LITE
RevC
dev_mount extsd /mnt/extsd auto /devices/platform/sdhci-esdhc-
imx.1/mmc host/mmc1

```

In the original setting, external SD doesn't write the permission, so we need to add the permission.

- src/frameworks/base/data/etc/platform.xml

```

<permission name="android.permission.READ_EXTERNAL_STORAGE" >
    <group gid="sdcard_r" />
    <group gid="media_rw" />
</permission>

<permission name="android.permission.WRITE_EXTERNAL_STORAGE" >

```

```
<group gid="sdcard_rw" />
<group gid="media_rw" />
</permission>
```

5.4. I2C-Tool

Download i2c-tool-3.1.0 from <http://www.lm-sensors.org/wiki/I2CTools>

Unpack the folder to src/external/i2c-tools-3.1.1

Add a new file Android.mk in src/external/i2c-tools-3.1.1

```
LOCAL_PATH:= $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE_TAGS := eng
LOCAL_C_INCLUDES += $(LOCAL_PATH) $(LOCAL_PATH)/$(KERNEL_DIR)/include
LOCAL_SRC_FILES := tools/i2cbusses.c tools/util.c
LOCAL_MODULE := i2c-tools
include $(BUILD_STATIC_LIBRARY)

include $(CLEAR_VARS)

LOCAL_MODULE_TAGS := eng
LOCAL_SRC_FILES:=tools/i2cdetect.c
LOCAL_MODULE:=i2cdetect
LOCAL_CPPFLAGS += -DANDROID
LOCAL_SHARED_LIBRARIES:=libc
LOCAL_STATIC_LIBRARIES := i2c-tools
LOCAL_C_INCLUDES += $(LOCAL_PATH) $(LOCAL_PATH)/$(KERNEL_DIR)/include
include $(BUILD_EXECUTABLE)

include $(CLEAR_VARS)

LOCAL_MODULE_TAGS := eng
LOCAL_SRC_FILES:=tools/i2cget.c
LOCAL_MODULE:=i2cget
LOCAL_CPPFLAGS += -DANDROID
LOCAL_SHARED_LIBRARIES:=libc
LOCAL_STATIC_LIBRARIES := i2c-tools
LOCAL_C_INCLUDES += $(LOCAL_PATH) $(LOCAL_PATH)/$(KERNEL_DIR)/include
include $(BUILD_EXECUTABLE)

include $(CLEAR_VARS)

LOCAL_MODULE_TAGS := eng
LOCAL_SRC_FILES:=tools/i2cset.c
LOCAL_MODULE:=i2cset
LOCAL_CPPFLAGS += -DANDROID
LOCAL_SHARED_LIBRARIES:=libc
```

```
LOCAL_STATIC_LIBRARIES := i2c-tools
LOCAL_C_INCLUDES += $(LOCAL_PATH) $(LOCAL_PATH)/$(KERNEL_DIR)/include
include $(BUILD_EXECUTABLE)

include $(CLEAR_VARS)

LOCAL_MODULE_TAGS := eng
LOCAL_SRC_FILES:=tools/i2cdump.c
LOCAL_MODULE:=i2cdump
LOCAL_CPPFLAGS += -DANDROID
LOCAL_SHARED_LIBRARIES:=libc
LOCAL_STATIC_LIBRARIES := i2c-tools
LOCAL_C_INCLUDES += $(LOCAL_PATH) $(LOCAL_PATH)/$(KERNEL_DIR)/include
include $(BUILD_EXECUTABLE)
```

5.5. Bootloader

5.5.1. Add bootloader argument

- src/bootable/bootloader/uboot-
imx/include/configs/mx6q_sabrelite_android.h

```
#define CONFIG_EXTRA_ENV_SETTINGS
"netdev=eth0\0" \
"ethprime=FEC0\0" \
"bootargs_base=setenv bootargs console=ttymxcl,115200\0" \
"bootargs_android=setenv bootargs ${bootargs} init=init \
androidboot.console=ttymxcl \0" \
"hdmI=setenv bootargs ${bootargs} \
video=mxcfb0:dev=hdmI,1920x1080M@60,bpp=32 video=mxcfb1:off \
video=mxcfb2:off video=mxcfb3:off fbmem=28M vmalloc=400M \
androidboot.hardware=freescale audio=hdmI\0" \
"lvds_auo_g140=setenv bootargs ${bootargs} video=mxcfb0:dev=ldb,LDB- \
XGA,if=RGB24,bpp=32 ldb=sin0 video=mxcfb1:off video=mxcfb2:off \
video=mxcfb3:off fbmem=28M vmalloc=400M androidboot.hardware=freescale \
audio=sgtl5000\0" \
"lvds_auo_g220=setenv bootargs ${bootargs} video=mxcfb0:dev=ldb,LDB- \
WSXGA+,if=RGB24,bpp=32 ldb=spl0 video=mxcfb1:off video=mxcfb2:off \
video=mxcfb3:off fbmem=28M vmalloc=400M androidboot.hardware=freescale \
audio=sgtl5000\0" \
"lvds_auo_g007=setenv bootargs ${bootargs} \
video=mxcfb0:dev=ldb,480C60,if=RGB24,bpp=32 ldb=sin0 video=mxcfb1:off \
video=mxcfb2:off video=mxcfb3:off fbmem=28M vmalloc=400M \
androidboot.hardware=freescale audio=sgtl5000\0" \
"display=run hdmI\0" \
"save_para=saveenv\0" \
"boot_media=booti mmc3\0" \
"bootcmd=run bootargs_base bootargs_android display save_para \
boot media\0"
```

5.5.2. Uboot print message

- src\bootable\bootloader\uboot-
imx\include\configs\mx6q_sabrelite_android.h

```
#define CONFIG_SYS_LONGHELP          /* undef to save memory */
#define CONFIG_SYS_PROMPT            "MX6Q VAB820 U-Boot > "
#define CONFIG_AUTO_COMPLETE
```

- src\bootable\bootloader\uboot-
imx\board\freescaler\mx6q_sabrelite\mx6q_sabrelite.c

```
int checkboard(void)
{
    printf("Board: MX6Q-VAB820:[ ");

    switch (__REG(SRC_BASE_ADDR + 0x8)) {
        (skip...)
    }
}
```

5.5.3. Uboot support microSD and eMMC

MicroSD uses uSDHC2 while eMMC uses uSDHC4 in uboot, so we have 4 uSDHC.

- src\bootable\bootloader\uboot-
imx\include\configs\mx6q_sabrelite_android.h

```
#ifdef CONFIG_CMD_MMC
#define CONFIG_MMC
#define CONFIG_GENERIC_MMC
#define CONFIG_IMX_MMC
#define CONFIG_SYS_FSL_USDHC_NUM    4
#define CONFIG_SYS_FSL_ESDHC_ADDR    0
#define CONFIG_SYS_MMC_ENV_DEV    2
#define CONFIG_DOS_PARTITION    1
#define CONFIG_CMD_FAT    1
#define CONFIG_CMD_EXT2    1
```

- src\bootable\bootloader\uboot-
imx\board\freescaler\mx6q_sabrelite\mx6q_sabrelite.c

```

struct fsl_esdhc_cfg usdhc_cfg[4] = {
    {USDHC1_BASE_ADDR, 1, 1, 1, 0},
    {USDHC2_BASE_ADDR, 1, 1, 1, 0},
    {USDHC3_BASE_ADDR, 1, 1, 1, 0},
    {USDHC4_BASE_ADDR, 1, 1, 1, 0},
};

(skip...)

iomux_v3_cfg_t mx6q_usdhc4_pads[] = {
    MX6Q_PAD_SD4_CLK__USDHC4_CLK,
    MX6Q_PAD_SD4_CMD__USDHC4_CMD,
    MX6Q_PAD_SD4_DAT0__USDHC4_DAT0,
    MX6Q_PAD_SD4_DAT1__USDHC4_DAT1,
    MX6Q_PAD_SD4_DAT2__USDHC4_DAT2,
    MX6Q_PAD_SD4_DAT3__USDHC4_DAT3,

    MX6Q_PAD_SD4_DAT4__USDHC4_DAT4,
    MX6Q_PAD_SD4_DAT5__USDHC4_DAT5,
    MX6Q_PAD_SD4_DAT6__USDHC4_DAT6,
    MX6Q_PAD_SD4_DAT7__USDHC4_DAT7,
};

int usdhc_gpio_init(bd_t *bis)
{
    s32 status = 0;
    u32 index = 0;

    for (index = 0; index < CONFIG_SYS_FSL_USDHC_NUM; index++) {
        switch (index) {

            case 0:

                mxc_iomux_v3_setup_multiple_pads(mx6q_usdhc1_pads,
                                                  sizeof(mx6q_usdhc1_pads) /
                                                  sizeof(mx6q_usdhc1_pads[0]));
                break;

            case 1:

                mxc_iomux_v3_setup_multiple_pads(mx6q_usdhc2_pads,
                                                  sizeof(mx6q_usdhc2_pads) /
                                                  sizeof(mx6q_usdhc2_pads[0]));
                break;

            case 2:

                mxc_iomux_v3_setup_multiple_pads(mx6q_usdhc3_pads,
                                                  sizeof(mx6q_usdhc3_pads) /
                                                  sizeof(mx6q_usdhc3_pads[0]));
                break;

            case 3:

                mxc_iomux_v3_setup_multiple_pads(mx6q_usdhc4_pads,

```

```

        sizeof(mx6q_usdhc4_pads) /
        sizeof(mx6q_usdhc4_pads[0]));
    break;

    default:
        printf("Warning: you configured more USDHC
controllers"
              "(%d) then supported by the board
(%d)\n",
              index + 1, CONFIG_SYS_FSL_USDHC_NUM);
        return status;
    }
    status |= fsl_esdhc_initialize(bis, &usdhc_cfg[index]);
}
return status;
}

```

5.5.4. Uboot set gpio

- src/bootable\bootloader\uboot-
imx\board\freescaler\mx6q_sabrelite\mx6q_sabrelite.c

```

void enet_board_init(void)
{
    iomux_v3_cfg_t enet_reset =
        (MX6Q_PAD_ENET_CRS_DV_GPIO_1_25 &
         ~MUX_PAD_CTRL_MASK) | MUX_PAD_CTRL(0x88);

    /* phy reset: gpio3-23 */
    set_gpio_output_val(GPIO1_BASE_ADDR, (1 << 25), 0);
    set_gpio_output_val(GPIO6_BASE_ADDR, (1 << 30),
                        (CONFIG_FEC0_PHY_ADDR >> 2));
    set_gpio_output_val(GPIO6_BASE_ADDR, (1 << 25), 1);
    set_gpio_output_val(GPIO6_BASE_ADDR, (1 << 27), 1);
    set_gpio_output_val(GPIO6_BASE_ADDR, (1 << 28), 1);
    set_gpio_output_val(GPIO6_BASE_ADDR, (1 << 29), 1);
    mxc_iomux_v3_setup_multiple_pads(enet_pads,
    ARRAY_SIZE(enet_pads));
    mxc_iomux_v3_setup_pad(enet_reset);
    udelay(500);
    set_gpio_output_val(GPIO1_BASE_ADDR, (1 << 25), 1);
    mxc_iomux_v3_setup_multiple_pads(enet_pads_final,
    ARRAY_SIZE(enet_pads_final));
}

```

5.5.5. U-boot set 2G RAM

VAB-820 has 1G or 2G RAM SKU. We added 2G RAM support.

- `src\bootable\bootloader\uboot-imx\`
`board\freescall\mx6q_sabrelite\mx6q_sabrelite.c`

```
int dram_init(void)
{
    // Steven
    ulong nRamSize = (1u * 1024 * 1024 * 1024);
    u32 reg = readl(GPIO2_BASE_ADDR + GPIO_PSR);

    if (!(reg & 0x01)) { nRamSize = (2u * 1024 * 1024 * 1024); }
    gd->bd->bi_dram[0].start = PHYS_SDRAM_1;
    gd->bd->bi_dram[0].size = nRamSize;
    return 0;
}
```

- `src\bootable\bootloader\uboot-`
`imx\board\freescall\mx6q_sabrelite\flash_header.S`

```
// steven: use flash plug_in
#define CONFIG_FLASH_PLUG_IN
...
#ifndef CONFIG_FLASH_PLUG_IN
    #ifdef VAB820_2G
        // 2G RAM clock settings

    #else
        // 1G RAM clock settings

    #endif
#else
    // auto detect RAM size and settings
#endif
```

- `src/kernel_imx/arch/arm/mach-mx6/board-mx6q_sabrelite.c`

```
// steven
static u32 s_vab820_ram_size = SZ_1G;
static void __init fixup_mxc_board(...)
{
    char *str;
    struct tag *t;
    int i = 0;
    struct ipuv3_fb_platform_data *pdata_fb = sabrelite_fb_data;

    for_each_tag(t, tags) {
        if (t->hdr.tag == ATAG_MEM) {
            s_vab820_ram_size = t->u.mem.size;
        }
        (skip)
    }
}
```

```
(skip)
static void __init mx6q_sabrelite_reserve(void)
{
    /*
     * Reserve primary framebuffer memory if its base address
     * is set by kernel command line.
     */
    fb_array_size = ARRAY_SIZE(sabrelite_fb_data);
    if (fb_array_size > 0 && sabrelite_fb_data[0].res_base[0] &&
        sabrelite_fb_data[0].res_size[0]) {
        if (sabrelite_fb_data[0].res_base[0] >
s_vab820_ram_size)
            printk(KERN_INFO"UI Performance downgrade with
FB phys address %x!\n",
                    sabrelite_fb_data[0].res_base[0]);
        memblock_reserve(sabrelite_fb_data[0].res_base[0],
                        sabrelite_fb_data[0].res_size[0]);
        memblock_remove(sabrelite_fb_data[0].res_base[0],
                        sabrelite_fb_data[0].res_size[0]);
        sabrelite_fb_data[0].late_init = true;
        ipu_data[ldb_data.ipu_id].bypass_reset = true;
        fb0_reserved = 1;
    }
    for (i = fb0_reserved; i < fb_array_size; i++)
        if (sabrelite_fb_data[i].res_size[0]) {
            /* Reserve for other background buffer. */
            phys =
memblock_alloc_base(sabrelite_fb_data[i].res_size[0],
                    SZ_4K,
s_vab820_ram_size);
            memblock_remove(phys,
sabrelite_fb_data[i].res_size[0]);
            sabrelite_fb_data[i].res_base[0] = phys;
        }

#if defined(CONFIG_MXC_GPU_VIV) || defined(CONFIG_MXC_GPU_VIV_MODULE)
    if (imx6q_gpu_pdata.reserved_mem_size) {
        phys =
memblock_alloc_base(imx6q_gpu_pdata.reserved_mem_size,
                    SZ_4K, s_vab820_ram_size);
        memblock_remove(phys,
imx6q_gpu_pdata.reserved_mem_size);
        imx6q_gpu_pdata.reserved_mem_base = phys;
    }
#endif

#if defined(CONFIG_ION)
    if (imx_ion_data.heaps[0].size) {
        phys = memblock_alloc(imx_ion_data.heaps[0].size,
SZ_4K);
        memblock_free(phys, imx_ion_data.heaps[0].size);
        memblock_remove(phys, imx_ion_data.heaps[0].size);
        imx_ion_data.heaps[0].base = phys;
    }
#endif
}
```

5.5.6. U-boot set factory data reset

The default data reset uses mmc1(microSD) to boot kernel, but we use mmc0.

- src\bootable\bootloader\uboot-imx\include\configs\mx6q_sabrelite_android.h

```
#define CONFIG_ANDROID_RECOVERY_BOOTARGS MMC NULL
#define CONFIG_ANDROID_RECOVERY_BOOTCMD_MMC \
    "booti mmc3 recovery"
#define CONFIG_ANDROID_RECOVERY_CMD_FILE "/recovery/command"
```

5.6. COM1 support flow control

Code modified by Peter.

- src\kernel_imx\arch\arm\mach-mx6\board-mx6q_sabrelite.c

```
static iomux_v3_cfg_t mx6q_sabrelite_pads[] = {
    (skip...)
    MX6Q_PAD_SD3_DAT0_UART1_RTS,
    MX6Q_PAD_EIM_D24_GPIO_3_24,
    MX6Q_PAD_EIM_D25_UART1_DSR,
    (skip...)
}
(skip...)
static const struct imxuart_platform_data mx6_vab820_uart1_data
__initconst = {
    .flags      = IMXUART_HAVE_RTSCTS | IMXUART_USE_DCEDETE |
IMXUART_SDMA,
    .dma_req_rx = MX6Q_DMA_REQ_UART1_RX,
    .dma_req_tx = MX6Q_DMA_REQ_UART1_TX,
};

static inline void mx6q_sabrelite_init_uart(void)
{
    imx6q_add_imx_uart(0, &mx6_vab820_uart1_data);
    imx6q_add_imx_uart(1, NULL);
}
(skip...)
static void __init mx6_sabrelite_board_init(void)
{
    (skip...)
```

```

gpio_set_value(MX6Q_SABRELITE_USB_HUB_RST_B, 1);

gpio_set_value(IMX_GPIO_NR(3, 24), 0);

imx6q_add_mxc_pwm(1);

(skip...)
}

```

- `src\kernel_imx\arch\arm\plat-mxc\include\mach\iomux-mx6q.h`

```

(skip...)
#define MX6Q_UART_RTSDTR_PAD_CTRL      (PAD_CTL_PKE | PAD_CTL_PUE |
\
    PAD_CTL_PUS_100K_DOWN | PAD_CTL_SPEED_MED | \
    PAD_CTL_DSE_40ohm | PAD_CTL_SRE_FAST | PAD_CTL_HYS)

(skip...)
#define MX6Q_PAD_EIM_D24_UART1_DTR      \
    (MX6Q_PAD_EIM_D24_UART1_DTR |
MUX_PAD_CTRL(MX6Q_UART_RTSDTR_PAD_CTRL))
(skip...)
#define MX6Q_PAD_SD3_DAT0_UART1_RTS      \
    (MX6Q_PAD_SD3_DAT0_UART1_RTS |
MUX_PAD_CTRL(MX6Q_UART_RTSDTR_PAD_CTRL))

```

- `src\kernel_imx\drivers\tty\serial\imx.c`

```

static void
imx_set_termios(struct uart_port *port, struct ktermios *termios,
                struct ktermios *old)
{
    (skip...)
    old_ucr1 = readl(sport->port.membase + UCR1);
    writel(old_ucr1 & ~(UCR1_TXMPTYEN | UCR1_RRDYEN | UCR1_RTSDEN),
           sport->port.membase + UCR1);
    old_ucr3 = readl(sport->port.membase + UCR3);
    writel(old_ucr3 & ~(UCR3_RI | UCR3_DCD), sport->port.membase +
           UCR3);

    while ( !(readl(sport->port.membase + USR2) & USR2_TXDC))
        barrier();

    (skip...)
}

```

5.7. Add support sgtl5000

5.7.1. Use tinymix to change audio parameter

- src\external\tinyalsa\Android.mk

```
include $(CLEAR_VARS)
LOCAL_C_INCLUDES:= external/tinyalsa/include
LOCAL_SRC_FILES:= tinyplay.c
LOCAL_MODULE := tinyplay
LOCAL_SHARED_LIBRARIES:= libcutils libutils libtinyalsa
#scott modify
LOCAL_MODULE_TAGS := eng

include $(BUILD_EXECUTABLE)

include $(CLEAR_VARS)
LOCAL_C_INCLUDES:= external/tinyalsa/include
LOCAL_SRC_FILES:= tinycap.c
LOCAL_MODULE := tinycap
LOCAL_SHARED_LIBRARIES:= libcutils libutils libtinyalsa
#scott modify
LOCAL_MODULE_TAGS := eng

include $(BUILD_EXECUTABLE)

include $(CLEAR_VARS)
LOCAL_C_INCLUDES:= external/tinyalsa/include
LOCAL_SRC_FILES:= tinymix.c
LOCAL_MODULE := tinymix
LOCAL_SHARED_LIBRARIES:= libcutils libutils libtinyalsa
#scott modify
LOCAL_MODULE_TAGS := eng

include $(BUILD_EXECUTABLE)
```

5.7.2. support sgtl5000 sound card

- src\hardware\imx\alsa\tinyalsa_hal.c

```
#include "audio_hardware.h"
#include "config_sgtl5000.h"
#include "config_wm8962.h"
#include "config_wm8958.h"
(skip...)
#define PRODUCT_DEVICE_PROPERTY "ro.product.device"
#define PRODUCT_NAME_PROPERTY "ro.product.name"
#define PRODUCT_DEVICE_IMX "imx"
#define PRODUCT_DEVICE_AUTO "sabreauto"
#define SUPPORT_CARD_NUM 8
(skip...)
struct audio_card *audio_card_list[SUPPORT_CARD_NUM] = {
    &sgtl5000_card,
    &wm8958_card,
```

```

        &wm8962_card,
        (skip...)
    }

static void select_output_device(struct imx_audio_device *adev)
{
    (skip...)

    if (adev->card_list[0]->supported_out_devices &
        AUDIO_DEVICE_OUT_AUX_DIGITAL)
        adev->out_device = AUDIO_DEVICE_OUT_AUX_DIGITAL;
    else
        adev->out_device = AUDIO_DEVICE_OUT_WIRED_HEADPHONE;

    /*if mode = AUDIO_MODE_IN_CALL*/
    ALOGV("headphone %d ,headset %d ,speaker %d, earpiece %d, \n",
        headphone_on, headset_on, speaker_on, earpiece_on);
    (skip...)
}

static int out_set_parameters(struct audio_stream *stream, const char
*kvpairs)
{
    (skip...)

    if (adev->card_list[0]->supported_out_devices &
        AUDIO_DEVICE_OUT_AUX_DIGITAL)
    {
        adev->out_device = AUDIO_DEVICE_OUT_AUX_DIGITAL;
        out->device = AUDIO_DEVICE_OUT_AUX_DIGITAL;
    }
    else
    {
        adev->out_device = AUDIO_DEVICE_OUT_WIRED_HEADPHONE;
        out->device = AUDIO_DEVICE_OUT_WIRED_HEADPHONE;
    }
    select_output_device(adev);

    pthread_mutex_unlock(&out->lock);
    (skip...)
}

static int adev_open_output_stream(struct audio_hw_device *dev,
                                audio_io_handle_t handle,
                                audio_devices_t devices,
                                audio_output_flags_t flags,
                                struct audio_config *config,
                                struct audio_stream_out **stream_out)
{
    (skip...)

    if (ladev->card_list[0]->supported_out_devices &
        AUDIO_DEVICE_OUT_AUX_DIGITAL)
        devices = AUDIO_DEVICE_OUT_AUX_DIGITAL;
    else
        devices = AUDIO_DEVICE_OUT_WIRED_HEADPHONE;

    out = (struct imx_stream_out *)calloc(1, sizeof(struct

```

```

imx_stream_out));
    if (!out)
        return -ENOMEM;
(skip...)
}

```

- src\hardware\imx\alsa\config_sgtl5000.h

```

#ifndef ANDROID_INCLUDE_IMX_CONFIG_SGTL5000_H
#define ANDROID_INCLUDE_IMX_CONFIG_SGTL5000_H

#include "audio_hardware.h"

#define MIXER_SGTL5000_SPEAKER_VOLUME "Speaker Volume"

/*These value are never change*/
static struct route_setting defaults_sgtl5000[]={
    /*general*/
    {
        .ctl_name = MIXER_SGTL5000_SPEAKER_VOLUME,
        .intval = 96,
    },
    {
        .ctl_name = NULL,
    },
};

/* ALSA cards for IMX, these must be defined according different board /
kernel config*/
static struct audio_card sgtl5000_card = {
    .name = "sgtl5000-audio",
    .driver_name = "sgtl5000-audio",
    .supported_out_devices = (AUDIO_DEVICE_OUT_SPEAKER |
        AUDIO_DEVICE_OUT_WIRED_HEADPHONE |
        AUDIO_DEVICE_OUT_DEFAULT),
    .supported_in_devices = AUDIO_DEVICE_IN_WIRED_HEADSET,
    .defaults = defaults_sgtl5000,
    .bt_output = NULL,
    .speaker_output = NULL,
    .hs_output = NULL,
    .earpiece_output = NULL,
    .vx_hs_mic_input = NULL,
    .mm_main_mic_input = NULL,
    .vx_main_mic_input = NULL,
    .mm_hs_mic_input = NULL,
    .vx_bt_mic_input = NULL,
    .mm_bt_mic_input = NULL,
    .card = 0,
    .out_rate = 0,
    .out_channels = 0,
    .out_format = 0,
    .in_rate = 0,
    .in_channels = 0,
    .in_format = 0,
}

```

```
};
#endif /* ANDROID_INCLUDE_IMX_CONFIG_SGTL5000_H */
```

- src\hardware\libhardware_legacy\audio\AudioPolicyManagerBase.cpp

Since HDMI volume is always set to full, but sgtl5000 doesn't. The output device needs to be set as headphone so that we can adjust the volume. Input device needs to be set as wire headset, otherwise input will have no voice.

```
audio_devices_t
AudioPolicyManagerBase::getDeviceForStrategy(routing_strategy strategy,
                                              bool
fromCache)
{
    (skip...)
        device = AUDIO_DEVICE_OUT_WIRED_HEADPHONE;

    ALOGVV("getDeviceForStrategy() strategy %d, device %x", strategy,
device);
    return device;
}

audio_devices_t AudioPolicyManagerBase::getDeviceForInputSource(int
inputSource)
{
    device = AUDIO_DEVICE_IN_WIRED_HEADSET;

    ALOGV("getDeviceForInputSource() input source %d, device %08x",
inputSource, device);
    return device;
}
```

- src\kernel\imx\sound\soc\imx\imx-sgtl5000.c

```
static struct snd_soc_dai_link imx_sgtl5000_dai[] = {
    {
        .name          = "HiFi",
        .stream_name   = "HiFi",
        .codec_dai_name = "sgtl5000",
        .codec_name     = "sgtl5000.2-000a",
        .cpu_dai_name   = "imx-ssi.1",
        .platform_name  = "imx-pcm-audio.1",
        .init           = imx_3stack_sgtl5000_init,
        .ops            = &imx_sgtl5000_hifi_ops,
    },
}
```

```
};
static int __init imx_sgtl5000_init(void)
{
    (skip...)
    if (machine_is_mx35_3ds() || machine_is_mx6q_sabrelite())
        imx_sgtl5000_dai[0].codec_name = "sgtl5000.2-000a";
    else
        imx_sgtl5000_dai[0].codec_name = "sgtl5000.1-000a";
    (skip...)
}
```

5.7.3. choose sound card from uboot

- src/kernel_imx/sound/soc/soc-core.c

```
(skip...)
static int soc_new_pcm(struct snd_soc_pcm_runtime *rtd, int num);
int g_audio_device = 0;
(skip...)

static int soc_bind_dai_link(struct snd_soc_card *card, int num)
{
    (skip...)
    char *device_codec_name;
    if(g_audio_device == 1) device_codec_name = "mxo_hdmi_soc.0";
    else
        device_codec_name = "sgtl5000.2-000a";

    if (rtd->complete)
        return 1;
    dev_dbg(card->dev, "binding %s at idx %d\n", dai_link->name,
num);
    (skip...)
    list_for_each_entry(codec, &codec_list, list) {
        if (!strcmp(codec->name, dai_link->codec_name)
&& !strcmp(device_codec_name, dai_link->codec_name)) {
            rtd->codec = codec;
        }
    }
}

static int __init set_audio_output(char *options)
{
    if (!strcmp(options, "hdmi"))
        g_audio_device = 1;
    else
        g_audio_device = 0;

    return 1;
}

setup("audio=", set_audio_output);
```

5.8. Add usb and usb otg keyboard/mouse resume

- `src\device\fsl\imx6\etc\init.rc`

```
# Enable USB host1 remote wakeup event wakeup android system
write /sys/bus/platform/drivers/fsl-ehci/fsl-ehci.1/power/wakeup
enabled
```

Modify code based on Freescale document

- `src\kernel_imx\arch\arm\plat-mxc\usb_common.c`

```
static int usb_register_remote_wakeup(struct platform_device *pdev)
{
    (skip...)
    device_init_wakeup(&pdev->dev, true);
    enable_irq_wake(irq);

    return 0;
}
```

- `src\kernel_imx\drivers\usb\core\hub.c`

```
void usb_set_device_state(struct usb_device *udev,
                          enum usb_device_state new_state)
{
    (skip...)
    if (wakeup >= 0)
        device_init_wakeup(&udev->dev, wakeup);
}
```

5.9. Add support of VNT9271

Freescale android bsp originally supports ath6kl driver, but VNT9271 uses ath9k_htc driver, so user has to modify makefile.

- `src\device\fsl\sabresd_6dq\BoardConfig.mk`

```
# Wifi
BOARD_WLAN_VENDOR := Atheros
# for atheros vendor
```

```

ifeq ($(BOARD_WLAN_VENDOR),ATHEROS)
BOARD_WLAN_DEVICE                := ar9485
BOARD_HAS_ATH_WLAN                := true
WPA_SUPPLICANT_VERSION           := VER_0_8_ATHEROS
BOARD_WPA_SUPPLICANT_DRIVER       := NL80211
WIFI_DRIVER_MODULE_PATH           :=
"/system/lib/modules/ath9k_htc.ko"
WIFI_DRIVER_MODULE_NAME           := "ath9k_htc"
WIFI_SDIO_IF_DRIVER_MODULE_PATH   :=
"/system/lib/modules/cfg80211.ko"
WIFI_SDIO_IF_DRIVER_MODULE_NAME   := "cfg80211"
WIFI_SDIO_IF_DRIVER_MODULE_ARG    := ""
WIFI_COMPAT_MODULE_PATH           :=
"/system/lib/modules/compat.ko"
WIFI_COMPAT_MODULE_NAME           := "compat"
WIFI_COMPAT_MODULE_ARG            := ""

```

- src\hardware\imx\wlan\atheros\compat-wireless\Android.mk

```

$(mod_cleanup) :
$(MAKE) -C $(ATH_ANDROID_SRC_BASE) ARCH=arm
CROSS_COMPILE=$(ATH_CROSS_COMPILE) KLIB=$(ATH_LINUXPATH)
KLIB_BUILD=$(ATH_LINUXPATH)
COMPAT_CURDIR=$(ATH_ANDROID_ROOT)/$(ATH_ANDROID_SRC_BASE) clean
mkdir -p $(TARGET_OUT)/etc/firmware/ath9k/AR9485/hw2.1.1/
mkdir -p $(TARGET_OUT)/lib/modules/

ath9k_htc_module_file :=drivers/net/wireless/ath/ath9k/ath9k_htc.ko
#$(ATH_ANDROID_SRC_BASE)/$(ath6kl_module_file):$(mod_cleanup)
$(TARGET_PREBUILT_KERNEL) $(ACP)
$(ATH_ANDROID_SRC_BASE)/$(ath9k_htc_module_file):$(mod_cleanup)
$(TARGET_PREBUILT_KERNEL) $(ACP)
$(MAKE) -C $(ATH_ANDROID_SRC_BASE) O=$(ATH_LINUXPATH) ARCH=arm
CROSS_COMPILE=$(ATH_CROSS_COMPILE) KLIB=$(ATH_LINUXPATH)
KLIB_BUILD=$(ATH_LINUXPATH)
COMPAT_CURDIR=$(ATH_ANDROID_ROOT)/$(ATH_ANDROID_SRC_BASE)
$(ACP) -fpt $(ATH_ANDROID_SRC_BASE)/compat/compat.ko
$(TARGET_OUT)/lib/modules/
$(ACP) -fpt $(ATH_ANDROID_SRC_BASE)/net/wireless/cfg80211.ko
$(TARGET_OUT)/lib/modules/
$(ACP) -fpt $(ATH_ANDROID_SRC_BASE)/net/mac80211/mac80211.ko
$(TARGET_OUT)/lib/modules/
$(ACP) -fpt
$(ATH_ANDROID_SRC_BASE)/drivers/net/wireless/ath/ath.ko
$(TARGET_OUT)/lib/modules/
$(ACP) -fpt
$(ATH_ANDROID_SRC_BASE)/drivers/net/wireless/ath/ath9k/ath9k_hw.ko
$(TARGET_OUT)/lib/modules/
$(ACP) -fpt
$(ATH_ANDROID_SRC_BASE)/drivers/net/wireless/ath/ath9k/ath9k_common.ko
$(TARGET_OUT)/lib/modules/
$(ACP) -fpt
$(ATH_ANDROID_SRC_BASE)/drivers/net/wireless/ath/ath9k/ath9k_htc.ko
$(TARGET_OUT)/lib/modules/

```

```
include $(CLEAR_VARS)
LOCAL_MODULE := ath9k_htc.ko
LOCAL_MODULE_TAGS := optional
LOCAL_MODULE_CLASS := ETC
LOCAL_MODULE_PATH := $(TARGET_OUT)/lib/modules
LOCAL_SRC_FILES := $(ath9k_htc_module_file)
include $(BUILD_PREBUILT)
```

- src\hardware\imx\wlan\atheros\compat-wireless\drivers\net\wireless\ath\Makefile

```
obj-$(CONFIG_ATH9K) += ath9k/
obj-$(CONFIG_ATH_COMMON) += ath.o
```

Ath9k_htc driver depends on other drivers, so we need to modify source code.

- src\hardware\libhardware_legacy\wifi_ath.c

```
#ifndef WIFI_DRIVER_MODULE_ARG
#define WIFI_DRIVER_MODULE_ARG ""
#endif
#ifndef WIFI_FIRMWARE_LOADER
#define WIFI_FIRMWARE_LOADER ""
#endif
#define WIFI_TEST_INTERFACE "sta"

#ifndef WIFI_DRIVER_MODULE_PATH
#define WIFI_DRIVER_MODULE_PATH "/system/lib/modules/ath9k.ko"
#endif
#ifndef WIFI_DRIVER_MODULE_NAME
#define WIFI_DRIVER_MODULE_NAME "ath9k"
#endif

#ifndef WIFI_DRIVER_HW_MODULE_PATH
#define WIFI_DRIVER_HW_MODULE_PATH "/system/lib/modules/ath9k_hw.ko"
#endif
#ifndef WIFI_DRIVER_HW_MODULE_NAME
#define WIFI_DRIVER_HW_MODULE_NAME "ath9k_hw"
#endif

#ifndef WIFI_DRIVER_COMMON_MODULE_PATH
#define WIFI_DRIVER_COMMON_MODULE_PATH "/system/lib/modules/ath9k_common.ko"
#endif
#ifndef WIFI_DRIVER_COMMON_MODULE_NAME
#define WIFI_DRIVER_COMMON_MODULE_NAME "ath9k_common"
```

```
#endif

#ifndef WIFI_DRIVER_ATH_MODULE_PATH
#define WIFI_DRIVER_ATH_MODULE_PATH      "/system/lib/modules/ath.ko"
#endif
#ifndef WIFI_DRIVER_ATH_MODULE_NAME
#define WIFI_DRIVER_ATH_MODULE_NAME      "ath"
#endif

#ifndef WIFI_DRIVER_MAC_MODULE_PATH
#define WIFI_DRIVER_MAC_MODULE_PATH      "/system/lib/modules/mac80211.ko"
#endif
#ifndef WIFI_DRIVER_MAC_MODULE_NAME
#define WIFI_DRIVER_MAC_MODULE_NAME      "mac80211"
#endif

#ifndef WIFI_DRIVER_HW_MODULE_ARG
#define WIFI_DRIVER_HW_MODULE_ARG        ""
#endif

#ifndef WIFI_DRIVER_COMMON_MODULE_ARG
#define WIFI_DRIVER_COMMON_MODULE_ARG    ""
#endif

#ifndef WIFI_DRIVER_ATH_MODULE_ARG
#define WIFI_DRIVER_ATH_MODULE_ARG        ""
#endif

#ifndef WIFI_DRIVER_MAC_MODULE_ARG
#define WIFI_DRIVER_MAC_MODULE_ARG        ""
#endif

(skip...)

#ifdef WIFI_DRIVER_MODULE_PATH
static const char DRIVER_MODULE_NAME[] = WIFI_DRIVER_MODULE_NAME;
static const char DRIVER_MODULE_TAG[] = WIFI_DRIVER_MODULE_NAME " ";
static const char DRIVER_MODULE_PATH[] = WIFI_DRIVER_MODULE_PATH;
static const char DRIVER_MODULE_ARG[] = WIFI_DRIVER_MODULE_ARG;

static const char DRIVER_HW_MODULE_NAME[] =
WIFI_DRIVER_HW_MODULE_NAME;
static const char DRIVER_HW_MODULE_TAG[] =
WIFI_DRIVER_HW_MODULE_NAME " ";
static const char DRIVER_HW_MODULE_PATH[] =
WIFI_DRIVER_HW_MODULE_PATH;
static const char DRIVER_HW_MODULE_ARG[] =
WIFI_DRIVER_HW_MODULE_ARG;

static const char DRIVER_COMMON_MODULE_NAME[] =
WIFI_DRIVER_COMMON_MODULE_NAME;
static const char DRIVER_COMMON_MODULE_TAG[] =
WIFI_DRIVER_COMMON_MODULE_NAME " ";
static const char DRIVER_COMMON_MODULE_PATH[] =
WIFI_DRIVER_COMMON_MODULE_PATH;
```

```
static const char DRIVER_COMMON_MODULE_ARG[] =
WIFI_DRIVER_COMMON_MODULE_ARG;

static const char DRIVER_ATH_MODULE_NAME[] =
WIFI_DRIVER_ATH_MODULE_NAME;
static const char DRIVER_ATH_MODULE_TAG[] =
WIFI_DRIVER_ATH_MODULE_NAME " ";
static const char DRIVER_ATH_MODULE_PATH[] =
WIFI_DRIVER_ATH_MODULE_PATH;
static const char DRIVER_ATH_MODULE_ARG[] =
WIFI_DRIVER_ATH_MODULE_ARG;

static const char DRIVER_MAC_MODULE_NAME[] =
WIFI_DRIVER_MAC_MODULE_NAME;
static const char DRIVER_MAC_MODULE_TAG[] =
WIFI_DRIVER_MAC_MODULE_NAME " ";
static const char DRIVER_MAC_MODULE_PATH[] =
WIFI_DRIVER_MAC_MODULE_PATH;
static const char DRIVER_MAC_MODULE_ARG[] =
WIFI_DRIVER_MAC_MODULE_ARG;
#endif

(skip...)
int wifi_load_driver()
{
(skip...)

    if (is_wifi_driver_loaded()) {
        return 0;
    }

    if (insmod(DRIVER_COMPAT_MODULE_PATH, DRIVER_COMPAT_MODULE_ARG) < 0)
        return -1;

    if (insmod(DRIVER_SDIO_IF_MODULE_PATH, DRIVER_SDIO_IF_MODULE_ARG) <
0)
        return -1;

    if (insmod(DRIVER_MAC_MODULE_PATH, DRIVER_MAC_MODULE_ARG) < 0)
        return -1;

    if (insmod(DRIVER_ATH_MODULE_PATH, DRIVER_ATH_MODULE_ARG) < 0)
        return -1;

    if (insmod(DRIVER_HW_MODULE_PATH, DRIVER_HW_MODULE_ARG) < 0)
        return -1;

    if (insmod(DRIVER_COMMON_MODULE_PATH, DRIVER_COMMON_MODULE_ARG) < 0)
        return -1;

    if (insmod(DRIVER_MODULE_PATH, DRIVER_MODULE_ARG) < 0)
        return -1;
(skip...)
}

int wifi_load_p2p_driver()
{

```

```

(skip...)
if (insmod(DRIVER_COMPAT_MODULE_PATH, DRIVER_COMPAT_MODULE_ARG) < 0)
    return -1;

if (insmod(DRIVER_SDIO_IF_MODULE_PATH, DRIVER_SDIO_IF_MODULE_ARG) <
0)
    return -1;

if (insmod(DRIVER_MAC_MODULE_PATH, DRIVER_MAC_MODULE_ARG) < 0)
    return -1;

if (insmod(DRIVER_ATH_MODULE_PATH, DRIVER_ATH_MODULE_ARG) < 0)
    return -1;

if (insmod(DRIVER_HW_MODULE_PATH, DRIVER_HW_MODULE_ARG) < 0)
    return -1;

if (insmod(DRIVER_COMMON_MODULE_PATH, DRIVER_COMMON_MODULE_ARG) < 0)
    return -1;

if (insmod(DRIVER_MODULE_PATH, DRIVER_MODULE_ARG) < 0)
    return -1;
(skip...)
}

static int _wifi_unload_driver()
{
(skip...)

if (count) {

    if (rmmod(DRIVER_COMMON_MODULE_NAME) != 0) {
        return -1;
    }

    /* unload compat kernel module */
    if (rmmod(DRIVER_HW_MODULE_NAME) != 0) {
        return -1;
    }

    if (rmmod(DRIVER_ATH_MODULE_NAME) != 0) {
        return -1;
    }

    /* unload compat kernel module */
    if (rmmod(DRIVER_MAC_MODULE_NAME) != 0) {
        return -1;
    }

    if (rmmod(DRIVER_SDIO_IF_MODULE_NAME) != 0) {
        return -1;
    }

    /* unload compat kernel module */
    if (rmmod(DRIVER_COMPAT_MODULE_NAME) != 0) {
        return -1;
    }
}

```

```

    }
    (skip...)
}

```

When we have compiled the right driver, next thing to do is to start wifi service, we also need to add wpa_supplicant.conf and htc_9271.fw to support wifi service.

- src\device\fs\limx6\etc\init.rc

```

# give system access to wpa_supplicant.conf for backup and restore
chmod 0775 /system/etc/dhcpd/dhcpd-run-hooks
mkdir /data/local 0751 root root

#wifi
mkdir /system/etc/wifi 0770 wifi wifi
chmod 0770 /system/etc/wifi
chmod 0660 /system/etc/wifi/wpa_supplicant.conf
chown wifi wifi /system/etc/wifi/wpa_supplicant.conf

#wpa_supplicant
mkdir /data/misc/wifi 0770 wifi wifi
mkdir /data/misc/wifi/sockets 0770 wifi wifi
chmod 0770 /data/misc/wifi
chmod 0660 /data/misc/wifi/wpa_supplicant.conf
chown wifi wifi /data/misc/wifi
chown wifi wifi /data/misc/wifi/wpa_supplicant.conf

```

- src\device\fs\limx6\etc\wpa_supplicant

```

##### wpa_supplicant configuration file template #####
update_config=1
ctrl_interface=wlan0
eapol_version=1
ap_scan=1
fast_reauth=1

```

- src\device\fs\limx6\limx6.mk

```

PRODUCT_COPY_FILES += \
    (skip...)
device/fsl/imx6/etc/wpa_supplicant.conf:system/etc/wifi/wpa_supplicant.conf \
    device/fsl/imx6/etc/htc_9271.fw:system/lib/modules/htc_9271.fw \
    device/fsl/imx6/etc/htc_9271.fw:system/etc/firmware/htc_9271.fw \
    (skip...)

```

- src\device\fs\sabresd_6dq\init.rc

```
(skip...)
service wpa_supplicant /system/bin/wpa_supplicant -dd -Dnl80211 -iwlan0
-c /data/misc/wifi/wpa_supplicant.conf
    socket wpa wlan0 dgram 660 wifi wifi
    group system wifi net
    disabled
    oneshot
(skip...)
```

- src\external\wpa_supplicant_ath\src\common\wpa_ctrl.c

```
#ifndef CONFIG_CTRL_IFACE_CLIENT_DIR
#define CONFIG_CTRL_IFACE_CLIENT_DIR "/data/misc/wifi/sockets"
#endif /* CONFIG_CTRL_IFACE_CLIENT_DIR */
```

5.10.Ethernet Static IP

Ethernet static IP is sated by ethernet.apk which is obtained from freescale android_jb4.3_1.0.0-ga patch. The functions of this AP are not complete so we added some functions.

5.10.1.Gateway

We control route by NetworkManagementService class.

- Ethernet\com\fs\ethernet\EthernetManager.java

```
// clear default route
RouteInfo[] routelist = mNMService.getRoutes(ifname);
for(int i = 0; i < routelist.length; i++) {
    if (routelist[i].isDefaultRoute()) {
        mNMService.removeRoute(ifname, routelist[i]);
    }
}
// add default route
InetAddress routeAddr =
NetworkUtils.numericToInetAddress(info.getRouteAddr());
RouteInfo mRoute = new RouteInfo(routeAddr);
mNMService.addRoute(ifname, mRoute);
```

```
// add host route
RouteInfo mHostRoute = new RouteInfo(new LinkAddress(routeAddr, 32),
null);
mNMService.addRoute(iframe, mHostRoute);
```

5.10.2.Netmask

- Ethernet\com\fs\ethernet\EthernetManager.java

```
ifcg.setLinkAddress(new LinkAddress(
    ipAddr,
    Integer.parseInt(info.getPrefixLength())
));
```

5.10.3.Settings record

We read / write data by PreferenceManager class. We write data when pressing 'confirm' in ethernet.apk and read data when enabling the service or opening the dialog.

- Ethernet\com\fs\ethernet\EthernetConfigDialog.java

```
/* save settings to .xml */
public void saveData() {
    String isConfig = SystemProperties.get("net.eth0.config", "0");

    if (isConfig.equals("1")) {
        String mode = SystemProperties.get("net.eth0.mode", "");
        String ip = SystemProperties.get("net.eth0.ip", "");
        String dns = SystemProperties.get("net.eth0.dns1", "");
        String netmask = SystemProperties.get("net.eth0.prefixlength",
        "");
        String gw = SystemProperties.get("net.eth0.route", "");

        mSettingData =
        PreferenceManager.getDefaultSharedPreferences(mContext);
        SharedPreferences.Editor PE = mSettingData.edit();

        PE.putString("MODE", mode);
        PE.putString("IP", ip);
        PE.putString("DNS", dns);
        PE.putString("NETMASK", netmask);
        PE.putString("GATEWAY", gw);

        PE.commit();
        Log.d(TAG, "    Store settings succeeded!");
    }
```

```

    } else {
        Log.d(TAG, "    Nothing change.");
    }
}

```

- Ethernet\com\fsl\ethernet\EthernetService.java

```

package com.fsl.ethernet;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.os.IBinder;
import android.util.Log;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import com.fsl.ethernet.EthernetManager;
import android.os.SystemProperties;

public class EthernetService extends Service {
    public static final String TAG = "EthernetService";
    private Context ctx;
    private SharedPreferences mSettingData;
    private EthernetManager mEthManager;

    @Override
    public void onCreate() {
        ctx = getApplicationContext();
        mSettingData =
PreferenceManager.getDefaultSharedPreferences(ctx);
        mEthManager = new EthernetManager(ctx);
    }

    public void onStart(Intent intent, int startId) {
        String mode = mSettingData.getString("MODE", "");

        Log.d(TAG, "Mode is " + mode);
        /* Default is dhcp, so we only load data when manual mode */
        if (mode.equals("manual")) {
            Log.d(TAG, "Setup ethernet static IP.");
            String ip = mSettingData.getString("IP", "");
            String dns = mSettingData.getString("DNS", "");
            String netmask = mSettingData.getString("NETMASK", "");
            String gw = mSettingData.getString("GATEWAY", "");

            SystemProperties.set("net.eth0.config", "1");

            SystemProperties.set("net.eth0.mode", mode);
            SystemProperties.set("net.eth0.ip", ip);
            SystemProperties.set("net.eth0.dns1", dns);
            SystemProperties.set("net.eth0.prefixlength", netmask);
            SystemProperties.set("net.eth0.route", gw);

            mEthManager.resetInterface();

```

```

    }
    // We just need do this service once when booting
    stopSelf();
}

@Override
public IBinder onBind(final Intent intent) {
    return null;
}
}

```

- Ethernet\com\fsl\ethernet\MainActivity.java

```

public void readData() {
    mSettingData = PreferenceManager.getDefaultSharedPreferences(this);

    String mode = mSettingData.getString("MODE", "");

    /* Default is dhcp, so we only load data when manual mode */
    if (mode.equals("manual")) {
        String ip = mSettingData.getString("IP", "");
        String dns = mSettingData.getString("DNS", "");
        String netmask = mSettingData.getString("NETMASK", "");
        String gw = mSettingData.getString("GATEWAY", "");

        SystemProperties.set("net.eth0.config", "1");

        SystemProperties.set("net.eth0.mode", mode);
        SystemProperties.set("net.eth0.ip", ip);
        SystemProperties.set("net.eth0.dns1", dns);
        SystemProperties.set("net.eth0.prefixlength", netmask);
        SystemProperties.set("net.eth0.route", gw);
    }
}
}

```

5.10.4.Enable service when booting

Accept the bootup broadcast and enable the Ethernet service.

- Ethernet\com\fsl\ethernet\BootUpReceiver.java

```

package com.fsl.ethernet;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;

public class BootUpReceiver extends BroadcastReceiver{

```

```
@Override
public void onReceive(Context c, Intent intent) {

    if(intent.getAction().equals(Intent.ACTION_BOOT_COMPLETED)){
        Intent sIntent = new Intent(c, EthernetService.class);
        c.startService(sIntent);
    }

}
}
```

- Ethernet\AndroidManifest.xml

```
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>

(skip...)
<!-- For bootUpReceiver-->
<receiver android:name=".BootUpReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
<service android:name=".EthernetService"/>
```

5.10.5.Check Input format

- Ethernet\com\fs\ethernet\EthernetConfigDialog.java

```
public void checkSettings() {
    String mStr = null;
    Matcher m;
    boolean isUsingDefault = false;

    // check IP format
    mStr = mIpaddr.getText().toString();
    m = Patterns.IP_ADDRESS.matcher(mStr);
    if (!m.matches()) {
        mIpaddr.setText("127.0.0.1", TextView.BufferType.EDITABLE);
        isUsingDefault = true;
    }

    // check Prefix length
    int networkPrefixLength = -1;
    try {
        networkPrefixLength =
Integer.parseInt(mNetMask.getText().toString());
        if (networkPrefixLength < 0 || networkPrefixLength > 32) {
            mNetMask.setText("24", TextView.BufferType.EDITABLE);
            isUsingDefault = true;
        }
    }
```

```

    } catch (NumberFormatException e) {
        mNetMask.setText("24", TextView.BufferType.EDITABLE);
        isUsingDefault = true;
    }

    // check Gateway format
    mStr = mGw.getText().toString();
    m = Patterns.IP_ADDRESS.matcher(mStr);
    if (!m.matches()) {
        if (!mStr.equals("0.0.0.0")) {
            mGw.setText("0.0.0.0", TextView.BufferType.EDITABLE);
            isUsingDefault = true;
        }
    }

    if (isUsingDefault) {
        Toast.makeText(mContext, "Error input format. Using default
settings.", Toast.LENGTH_SHORT).show();
    }
}

```

5.10.6. Send message to inform Ethernet.apk

- src/frameworks/base/core/java/android/net/EthernetDataTracker.java

```

private void runDhcp() {
    Thread dhcpThread = new Thread(new Runnable() {
        public void run() {
            DhcpInfoInternal dhcpInfoInternal = new DhcpInfoInternal();
            System.out.println("Ethernet setting...");

            if (!NetworkUtils.runDhcp(mIface, dhcpInfoInternal)) {
                Log.e(TAG, "DHCP request error:" +
NetworkUtils.getDhcpError());

                // Send message to inform the ConnectivityService to
                // setup static IP. David
                mNetworkInfo.setDetailedState(DetailedState.CONNECTED,
                null, mHwAddr);

                Message msg =
                mCsHandler.obtainMessage(EVENT_STATE_CHANGED, mNetworkInfo);
                msg.sendToTarget();

                return;
            }

            mLinkProperties = dhcpInfoInternal.makeLinkProperties();
            mLinkProperties.setInterfaceName(mIface);

            mNetworkInfo.setDetailedState(DetailedState.CONNECTED, null,
            mHwAddr);

            Message msg = mCsHandler.obtainMessage(EVENT_STATE_CHANGED,
            mNetworkInfo);
            msg.sendToTarget();
        }
    });
    dhcpThread.start();
}

```

- `src/frameworks/base/services/java/com/android/server/ConnectivityService.java`

```
// must be stateless - things change under us.
private class NetworkStateTrackerHandler extends Handler {
    public NetworkStateTrackerHandler(Looper looper) {
        super(looper);
    }

    @Override
    public void handleMessage(Message msg) {
        NetworkInfo info;

        switch (msg.what) {
            (skip...)
            case NetworkStateTracker.EVENT_NETWORK_SUBTYPE_CHANGED:
                info = (NetworkInfo) msg.obj;
                type = info.getType();
                updateNetworkSettings(mNetTrackers[type]);
                break;

                // Send message to Ethernet.apk to setup the eth0. David
            case NetworkStateTracker.EVENT_RESTORE_DEFAULT_NETWORK:
                info = (NetworkInfo) msg.obj;
                sendConnectedBroadcastDelayed(info,
getConnectivityChangeDelay());
                log("Ethernet setup completely.");
                break;
        }
    }
}
```

5.10.7. Set the default value

Set the default values of DNS and gateway.

- `Ethernet\com\fs\ethernet\EthernetConfigDialog.java`

```
public void checkSettings() {
    String mStr = null;
    Matcher m;
    boolean isUsingDefault = false;

    // check IP format
    mStr = mIpaddr.getText().toString();
    m = Patterns.IP_ADDRESS.matcher(mStr);
    if (!m.matches()) {
        mIpaddr.setText("127.0.0.1", TextView.BufferType.EDITABLE);
        isUsingDefault = true;
    }
}
```

```

        // check Prefix length
        int networkPrefixLength = -1;
        try {
            networkPrefixLength =
Integer.parseInt(mNetMask.getText().toString());
            if (networkPrefixLength < 0 || networkPrefixLength > 32) {
                mNetMask.setText("24", TextView.BufferType.EDITABLE);
                isUsingDefault = true;
            }
        } catch (NumberFormatException e) {
            mNetMask.setText("24", TextView.BufferType.EDITABLE);
            isUsingDefault = true;
        }

        // check Gateway format
        mStr = mGw.getText().toString();
        m = Patterns.IP_ADDRESS.matcher(mStr);
        if (!m.matches()) {
            if (!mStr.equals("0.0.0.0")) {
                mGw.setText("0.0.0.0", TextView.BufferType.EDITABLE);
                isUsingDefault = true;
            }
        }

        // check DNS format
        mStr = mDns.getText().toString();
        m = Patterns.IP_ADDRESS.matcher(mStr);
        if (!m.matches()) {
            mDns.setText("8.8.8.8", TextView.BufferType.EDITABLE);
            isUsingDefault = true;
        }

        if (isUsingDefault) {
            Toast.makeText(mContext, "Error input format. Using default
settings.", Toast.LENGTH_SHORT).show();
        }
    }
}

```

- Ethernet\com\fs\ethernet\EthernetManager.java

```

void configureInterface(EthernetDevInfo info) {
    String ifname = info.getIfName();

    if
(info.getConnectMode().equals(EthernetDevInfo.ETHERNET_CONN_MODE_DHCP)) {
        Log.e(TAG, "Setup DHCP");
        try {
            mNMService.setInterfaceDown(ifname);
            mNMService.setInterfaceUp(ifname);
        } catch (RemoteException re){
            Log.e(TAG, "DHCP configuration failed: " + re);
        } catch (IllegalStateException e) {
            Log.e(TAG, "DHCP configuration fialed: " + e);
        }
    }
}

```

```

    } else {
        InterfaceConfiguration ifcg = null;
        try {
            InetAddress ipAddr =
NetworkUtils.numericToInetAddress(info.getIpAddress());
            // set IP and netmask
            ifcg = mNMService.getInterfaceConfig(ifname);
            ifcg.setLinkAddress(new LinkAddress(
                ipAddr,
                Integer.parseInt(info.getPrefixLength())
            ));
            ifcg.setInterfaceUp();

            mNMService.setInterfaceConfig(ifname, ifcg);

            if (!info.getRouteAddr().equals("0.0.0.0")) {
                // clear default route
                RouteInfo[] routelist = mNMService.getRoutes(ifname);
                for(int i = 0; i < routelist.length; i++) {
                    if (routelist[i].isDefaultRoute()) {
                        mNMService.removeRoute(ifname, routelist[i]);
                    }
                }
                // add default route
                InetAddress routeAddr =
NetworkUtils.numericToInetAddress(info.getRouteAddr());
                RouteInfo mRoute = new RouteInfo(routeAddr);
                mNMService.addRoute(ifname, mRoute);
                // add host route
                RouteInfo mHostRoute = new RouteInfo(new
LinkAddress(routeAddr, 32), null);
                mNMService.addRoute(ifname, mHostRoute);
            } else {
                Log.d(TAG, "no gateway setting");
            }
            Log.d(TAG, "Static IP configuration succeeded");
        } catch (RemoteException re) {
            Log.e(TAG, "Static IP configuration failed: " + re);
        } catch (IllegalStateException e) {
            Log.e(TAG, "Static IP configuration fialed: " + e);
        }
        Log.d(TAG, "set ip manually " + info.toString());
        SystemProperties.set("net.dns1", info.getDnsAddr());
        SystemProperties.set("net." + ifname +
".dns1", info.getDnsAddr());
        SystemProperties.set("net." + ifname + ".dns2", "0.0.0.0");
    }
}

```

5.11. Set WiFi as the preferred network

- src/frameworks\base\packages\SettingsProvider\res\values\defaults.xml

```
<integer name="def_network_preference">1</integer>
```

- src\frameworks\base\packages\SettingsProvider\src\com\android\providers\settings\DatabaseHelper.java

```
loadBooleanSetting(stmt, Settings.Global.AIRPLANE_MODE_ON,
    R.bool.def_airplane_mode_on);

// Add Global.NETWORK_PREFERENCE to set preferred network connection.
loadIntegerSetting(stmt, Settings.Global.NETWORK_PREFERENCE,
    R.integer.def_network_preference);

loadStringSetting(stmt, Settings.Global.AIRPLANE_MODE_RADIOS,
    R.string.def_airplane_mode_radios);
```

- src\frameworks\base\servicesjava\com\android\ConnectivityService.java

```
private int getPersistedNetworkPreference() {
    final ContentResolver cr = mContext.getContentResolver();

    final int networkPrefSetting = Settings.Global.getInt(cr,
Settings.Global.NETWORK_PREFERENCE, -1);
    if (networkPrefSetting != -1) {
        return networkPrefSetting;
    }
    return ConnectivityManager.DEFAULT_NETWORK_PREFERENCE;
}

(skip)

private void handleConnect(NetworkInfo info) {
    final int newNetType = info.getType();
    setupDataActivityTracking(newNetType);
    // snapshot isFailover, because sendConnectedBroadcast() resets it
    boolean isFailover = info.isFailover();
    final NetworkStateTracker thisNet = mNetTrackers[newNetType];
    final String thisIface =
        thisNet.getLinkProperties().getInterfaceName();

    // if this is a default net and other default is running
    // kill the one not preferred
    if (mNetConfigs[newNetType].isDefault()) {
        if (mActiveDefaultNetwork != -1 &&
            mActiveDefaultNetwork != newNetType) {
            if (isNewNetTypePreferredOverCurrentNetType(newNetType)) {
                // tear down the other
                NetworkStateTracker otherNet =
                    mNetTrackers[mActiveDefaultNetwork];
                if (DBG) {
```

```

        log("Policy requires " +
            otherNet.getNetworkInfo().getTypeName() +
            " teardown");
    }

    } else {
        // Ethernet can co-exist with other network,
        // so we do not need to teardown other
        if (newNetType != TYPE_ETHERNET) {
            // don't accept this one
            if (DBG) {
                log("Not broadcasting CONNECT_ACTION " +
                    "to torn down network " +
                    info.getTypeName());
            }
            teardown(thisNet);
            return;
        }
    }
}

```

5.12. Fix USB device detection

USB device can't detect some devices because it doesn't reset when system boots.

- src\kernel_imx\arch\arm\mach-mx6\board-mx6q_sabrelite.c

```

static void __init mx6_sabrelite_board_init(void)
{
    (skip...)
    /* release USB Hub reset */
    // (Sylvia) modified
    //gpio_set_value(MX6Q_SABRELITE_USB_HUB_RESET, 1);
    //gpio_set_value(MX6Q_SABRELITE_USB_HUB_RST_B, 1);
    gpio_request(MX6Q_SABRELITE_USB_HUB_RST_B, "usbhub-reset");
    gpio_direction_output(MX6Q_SABRELITE_USB_HUB_RST_B, 1);
    gpio_set_value(MX6Q_SABRELITE_USB_HUB_RST_B, 0);
    msleep(1);
    gpio_set_value(MX6Q_SABRELITE_USB_HUB_RST_B, 1);
    (skip...)
}

```

5.13. Add SPI ROM

Create chip initial in kernel.

- src\kernel_imx\arch\arm\mach-mx6\board-mx6q_sabrelite.c

```
static struct fec platform_data fec_data __initdata = {
    .init = mx6q_sabrelite_fec_phy_init,
    .phy = PHY_INTERFACE_MODE_RGMII,
#ifdef CONFIG_MX6_ENET_IRQ_TO_GPIO
    .gpio_irq = MX6_ENET_IRQ,
#endif
};

static int mx6q_sabrelite_spi_cs1[] = {
    MX6Q_SABRELITE_ECSPI1_CS1,
};

static const struct spi_imx_master mx6q_sabrelite_spi_data1 __initconst
= {
    .chipselect      = mx6q_sabrelite_spi_cs1,
    .num_chipselect  = ARRAY_SIZE(mx6q_sabrelite_spi_cs1),
};

static int mx6q_sabrelite_spi_cs3[] = {
    MX6Q_SABRELITE_ECSPI3_CS0,
    MX6Q_SABRELITE_ECSPI3_CS1,
};

static const struct spi_imx_master mx6q_sabrelite_spi_data3 __initconst
= {
    .chipselect      = mx6q_sabrelite_spi_cs3,
    .num_chipselect  = ARRAY_SIZE(mx6q_sabrelite_spi_cs3),
};
#if defined(CONFIG_MTD_M25P80) || defined(CONFIG_MTD_M25P80_MODULE)
/*
static struct mtd_partition imx6_sabrelite_spi_nor_partitions[] = {
    {
        .name = "bootloader",
        .offset = 0,
        .size = 0x00100000,
    },

    {
        .name = "kernel",
        .offset = MTDPART_OFS_APPEND,
        .size = MTDPART_SIZ_FULL,
    },
};
*/

static struct flash_platform_data imx6_sabrelite__spi_flash_data = {
    .name = "m25p80",
    //.parts = imx6_sabrelite_spi_nor_partitions,
    //.nr_parts = ARRAY_SIZE(imx6_sabrelite_spi_nor_partitions),
    .type = "sst25vf032b", //"sst25vf016b",
};
#endif

static struct spi_board_info imx6_sabrelite_spi_nor_device[] __initdata
```

```
= {
#if defined(CONFIG_MTD_M25P80)
{
        .modalias = "m25p80",
        .max_speed_hz = 20000000, /* max spi clock (SCK) speed

in HZ */
        .bus_num = 0,
        .chip_select = 0,
        .platform_data = &imx6_sabrelite__spi_flash_data,
    },
    {
        .modalias = "m25p80",
        .max_speed_hz = 20000000, /* max spi clock (SCK) speed

in HZ */
        .bus_num = 2,    // ECSPI3=2
        .chip_select = 0,
        .platform_data = &imx6_sabrelite__spi_flash_data,
    },
    {
        .modalias = "m25p80",
        .max_speed_hz = 20000000, /* max spi clock (SCK) speed

in HZ */
        .bus_num = 2,    // ECSPI3=2
        .chip_select = 1,
        .platform_data = &imx6_sabrelite__spi_flash_data,
    },
#endif
};
(skip...)
static void __init mx6_sabrelite_board_init(void)
{
    (skip...)
    i2c_register_board_info(2, mxc_i2c2_board_info,
        ARRAY_SIZE(mxc_i2c2_board_info));

    /* SPI */
    imx6q_add_ecspi(0, &mx6q_sabrelite_spi_data1);
    imx6q_add_ecspi(2, &mx6q_sabrelite_spi_data3);
    spi_device_init();

    imx6q_add_mxc_hdmi(&hdmi_data);
    (skip...)
}
```

Add new SPI rom chip MXIC and Winbone

- [src/bootable/bootloader/uboot-imx/drivers/mtd/spi/imx_spi_nor_sst.c](#)

```
#define WRITE_DISABLE(a)          spi_nor_cmd_1byte(a, WRDI)
#define ENABLE_WRITE_STATUS(a)    spi_nor_cmd_1byte(a, EWSR)

/* steven: add idcode2 to check device id */
/* steven: add page_size for non-sst spi rom */
struct imx_spi_flash_params {
```

```

    u8          idcode1;
+   u8          idcode2;
    u32         block_size;
    u32         block_count;
    u32         device_size;
+   u32         page_size;
    const char  *name;
};

@@ -60,13 +64,54 @@ to imx_spi_flash(struct spi_flash *flash)
static const struct imx_spi_flash_params imx_spi_flash_table[] = {
{
    .idcode1      = 0x25,
+   .idcode2      = 0x41,
    .block_size   = SZ_64K,
    .block_count  = 32,
    .device_size  = SZ_64K * 32,
+   .page_size    = 256,
    .name         = "SST25VF016B - 2MB",
},
{ /* vendor id = 0xBF */
    .idcode1      = 0x25,
+   .idcode2      = 0x4A,
    .block_size   = SZ_64K,
    .block_count  = 64,
    .device_size  = SZ_64K * 64,
+   .page_size    = 256,
    .name         = "SST25VF032B - 4MB",
},
{ /* vendor id = 0xEF */
    .idcode1      = 0x40,
+   .idcode2      = 0x16,
    .block_size   = SZ_64K,
    .block_count  = 64,
    .device_size  = SZ_64K * 64,
+   .page_size    = 256,
    .name         = "W25Q32FV - 4MB",
},
{ /* vendor id = 0xC2 */
    .idcode1      = 0x20,
+   .idcode2      = 0x16,
    .block_size   = SZ_64K,
    .block_count  = 64,
    .device_size  = SZ_64K * 64,
+   .page_size    = 256,
    .name         = "WX25L3206E - 4MB",
},
{ /* vendor id = 0x20 */
    .idcode1      = 0xBA,
+   .idcode2      = 0x16,
    .block_size   = SZ_64K,
    .block_count  = 64,
    .device_size  = SZ_64K * 64,
+   .page_size    = 256,
    .name         = "N25Q032A - 4MB",
},
};

```

```

/* steven: support mwl */
#include "spi_mwl.c"
+
static s32 spi_nor_flash_query(struct spi_flash *flash, void* data)
{
    u8 au8Tmp[4] = { 0 };
@@ -505,7 +550,8
@@ struct spi_flash *spi_flash_probe(unsigned int bus, unsigned int cs, un
signed in

    for (i = 0; i < ARRAY_SIZE(imx_spi_flash_table); ++i) {
        params = &imx_spi_flash_table[i];
        if (params->idcode1 == idcode[1])
-
+ /* steven: check device id */
+     if ( (params->idcode1 == idcode[1]) && (params-
>idcode2 == idcode[2]) ) break;
    }

@@ -521,9 +567,19
@@ struct spi_flash *spi_flash_probe(unsigned int bus, unsigned int cs, un
signed in
    imx_sf->flash.name = params->name;
    imx_sf->flash.size = params->device_size;

-    imx_sf->flash.read = spi_nor_flash_read;
-    imx_sf->flash.write = spi_nor_flash_write;
-    imx_sf->flash.erase = spi_nor_flash_erase;
+ /* steven: support mwl */
+    switch(idcode[0]) {
+    case 0xBF: /* sst spi rom */
+        imx_sf->flash.read = spi_nor_flash_read;
+        imx_sf->flash.write = spi_nor_flash_write;
+        imx_sf->flash.erase = spi_nor_flash_erase;
+        break;
+    default: /* m25p32, w25q32, WX25L3206E spi rom */
+        imx_sf->flash.read = mwl_spi_nor_flash_read;
+        imx_sf->flash.write = mwl_spi_nor_flash_write;
+        imx_sf->flash.erase = mwl_spi_nor_flash_erase;
+        break;
+    }

```

- src/bootable/bootloader/uboot-imx/drivers/mtd/spi/spi_mw1.c

5.14.Add support of CAN bus

Add pin definition and chip initial in kernel.

- src/kernel_imx\arch\arm\mach-mx6\board-mx6q_sabrelite.c

```
static iomux_v3_cfg_t mx6q_sabrelite_pads[] = {
(skip...)
    /* CAN1 */
    MX6Q_PAD_SD3_CMD__CAN1_TXCAN,
    MX6Q_PAD_SD3_CLK__CAN1_RXCAN,
    /* CAN2 */
    MX6Q_PAD_KEY_COL4__CAN2_TXCAN,
    MX6Q_PAD_KEY_ROW4__CAN2_RXCAN,
}
(skip...)
static struct gpio mx6q_sabrelite_flexcan_gpios[] = {
    //{ MX6Q_SABRELITE_CAN1_EN, GPIOF_OUT_INIT_LOW, "flexcan1-en" },
    //{ MX6Q_SABRELITE_CAN1_STBY, GPIOF_OUT_INIT_LOW, "flexcan1-
stby" },
};

static void mx6q_sabrelite_flexcan0_switch(int enable)
{
    if (enable) {
        //gpio_set_value(MX6Q_SABRELITE_CAN1_EN, 1);
        //gpio_set_value(MX6Q_SABRELITE_CAN1_STBY, 1);
    } else {
        //gpio_set_value(MX6Q_SABRELITE_CAN1_EN, 0);
        //gpio_set_value(MX6Q_SABRELITE_CAN1_STBY, 0);
    }
}

static void mx6q_sabrelite_flexcan1_switch(int enable)
{
    if (enable) {
        //gpio_set_value(MX6Q_SABRELITE_CAN1_EN, 1);
        //gpio_set_value(MX6Q_SABRELITE_CAN1_STBY, 1);
    } else {
        //gpio_set_value(MX6Q_SABRELITE_CAN1_EN, 0);
        //gpio_set_value(MX6Q_SABRELITE_CAN1_STBY, 0);
    }
}

static const struct flexcan_platform_data
mx6q_sabrelite_flexcan0_pdata __initconst = {
    .transceiver_switch = mx6q_sabrelite_flexcan0_switch,
};

static const struct flexcan_platform_data
mx6q_sabrelite_flexcan1_pdata __initconst = {
    .transceiver_switch = mx6q_sabrelite_flexcan1_switch,
};
(skip...)
static void __init mx6_sabrelite_board_init(void)
{
(skip...)
    int ret;
(skip...)
    imx6q_add_hdmi_soc();
    imx6q_add_hdmi_soc_dai();

    ret = gpio_request_array(mx6q_sabrelite_flexcan_gpios,
```

```

        ARRAY_SIZE(mx6q_sabrelite_flexcan_gpios));
    if (ret)
        pr_err("failed to request flexcan1-gpios: %d\n", ret);
    else
    {
        imx6q_add_flexcan0(&mx6q_sabrelite_flexcan0_pdata);
        imx6q_add_flexcan1(&mx6q_sabrelite_flexcan1_pdata);
    }

    clko2 = clk_get(NULL, "clko2_clk");
(skip...)
}

```

Modify flexcan driver

- src/kernel_imx/drivers/net/can/flexcan.c

```

static int flexcan_start_xmit(struct sk_buff *skb, struct net_device
*dev)
{
    (skip...)
    struct net_device_stats *stats = &dev->stats;
(skip...)
    //can_put_echo_skb(skb, dev, 0);
    writel(can_id, &regs->cantxfg[FLEXCAN_TX_BUF_ID].can_id);
    writel(ctrl1, &regs->cantxfg[FLEXCAN_TX_BUF_ID].can_ctrl1);

    (skip...)
    stats->tx_bytes += cf->can_dlc;
    stats->tx_packets++;

    return NETDEV_TX_OK;
}
(skip...)
static irqreturn_t flexcan_irq(int irq, void *dev_id)
{
    (skip...)
    //struct net_device_stats *stats = &dev->stats;
(skip...)
    if (reg_iflag1 & (1 << FLEXCAN_TX_BUF_ID)) {
        //stats->tx_bytes += can_get_echo_skb(dev, 0);
        //stats->tx_packets++;
        writel((1 << FLEXCAN_TX_BUF_ID), &regs->iflag1);
        netif_wake_queue(dev);
    }

    return IRQ_HANDLED;
}

```

5.15.Add support of 3G

Use driver provided by u-blox, copy “build “external” “hardware”

“system/core/init” “system/core/liblog” folders in RIL_sc_02.01 folder to our “src” folder. One file needs to manually patch.

- src\build\target\product\core.mk

```
PRODUCT_COPY_FILES += \
    system/core/rootdir/init.usb.rc:root/init.usb.rc \
    system/core/rootdir/init.trace.rc:root/init.trace.rc \
    $(LOCAL_PATH)/rootdir/etc/init.gprs-pppd:system/bin/init.gprs-pppd
\
    $(LOCAL_PATH)/rootdir/etc/ppp/chap-secrets:system/etc/ppp/chap-
secrets \
    $(LOCAL_PATH)/rootdir/etc/ppp/ip-down:system/bin/ip-down \
    $(LOCAL_PATH)/rootdir/etc/ppp/ip-up:system/bin/ip-up \
    $(LOCAL_PATH)/rootdir/etc/ppp/chap-secrets:system/etc/ppp/pap-
secrets \
    $(LOCAL_PATH)/rootdir/etc/ppp/stop_pppd:system/bin/stop_pppd \
    $(LOCAL_PATH)/rootdir/etc/r ril/repository.txt:system/etc/r ril/repo
sitatory.txt \
    $(LOCAL_PATH)/rootdir/etc/r ril-repo.sh:system/bin/r ril-repo.sh

ifeq ($(strip $(RIL_COM_INTERFACE)),spi)
PRODUCT_COPY_FILES += \
    $(LOCAL_PATH)/rootdir/etc/stop_muxd:system/bin/stop_muxd \
    $(LOCAL_PATH)/rootdir/etc/ppp/chat-isp1:/system/etc/ppp/chat-isp1
\
    $(LOCAL_PATH)/rootdir/etc/ppp/chat-isp2:/system/etc/ppp/chat-isp2
\
    $(LOCAL_PATH)/rootdir/etc/ppp/peers/spi-
gprs1:system/etc/ppp/peers/gprs1 \
    $(LOCAL_PATH)/rootdir/etc/ppp/peers/spi-
gprs2:system/etc/ppp/peers/gprs2
else ifeq ($(strip $(RIL_COM_INTERFACE)),uart)
PRODUCT_COPY_FILES += \
    $(LOCAL_PATH)/rootdir/etc/stop_muxd:system/bin/stop_muxd \
    $(LOCAL_PATH)/rootdir/etc/ppp/chat-isp1:/system/etc/ppp/chat-isp1
\
    $(LOCAL_PATH)/rootdir/etc/ppp/chat-isp2:/system/etc/ppp/chat-isp2
\
    $(LOCAL_PATH)/rootdir/etc/ppp/peers/uart-
gprs1:system/etc/ppp/peers/gprs1 \
    $(LOCAL_PATH)/rootdir/etc/ppp/peers/uart-
gprs2:system/etc/ppp/peers/gprs2
else ifeq ($(strip $(RIL_COM_INTERFACE)),usb)
PRODUCT_COPY_FILES += \
    $(LOCAL_PATH)/rootdir/etc/ppp/chat-isp1:/system/etc/ppp/chat-isp1
\
    $(LOCAL_PATH)/rootdir/etc/ppp/chat-isp2:/system/etc/ppp/chat-isp2
\

```

```

$(LOCAL_PATH)/rootdir/etc/ppp/peers/usb-
gprs1:system/etc/ppp/peers/gprs1 \
$(LOCAL_PATH)/rootdir/etc/ppp/peers/usb-
gprs2:system/etc/ppp/peers/gprs2
Endif

```

According to the document provided by u-blox

“Android_RIL_Souce_code_application.pdf,” we need to start 3G service in initial.

- src\device\fs\lmx6\AndroidProducts.mk

```

# Copied from build/target/product/AndroidProducts.mk

RIL_COM_INTERFACE := usb

```

- src\device\fs\lmx6\etc\init.rc

```

(skip...)
service hostapd /system/bin/hostapd /data/misc/wifi/hostapd.conf
    socket hostapd_wlan0 dgram 660 root wifi
    user root
    group wifi
    oneshot
    disabled

#service ril-daemon /system/bin/rild
#    class main
#    socket rild stream 660 root radio
#    socket rild-debug stream 660 radio system
#    socket rild-ppp stream 660 radio system
#    user root
#    group radio cache inet misc audio sdcard_rw log

#service pppd_gprs /etc/ppp/init.gprs-pppd
#    class main
#    user root
#    group radio cache inet misc
#    disabled
#    oneshot

service wifi_mac /system/bin/sh /system/etc/check_wifi_mac.sh
    class late_start
    oneshot
(skip...)

```

- src\device\fs\lsabresd_6dq\init.rc

```
(skip...)
# No bluetooth hardware present
setprop wlan.interface wlan0

chmod 0660 /dev/ttyACM1
chown radio radio /dev/ttyACM1
chmod 0660 /dev/ttyACM2
chown radio radio /dev/ttyACM2
chmod 0660 /dev/ttyACM3
chown radio radio /dev/ttyACM3

chown radio radio /system/etc/r ril
chmod 0770 /system/etc/r ril
chown radio radio /system/etc/r ril/repository.txt
chmod 0660 /system/etc/r ril/repository.txt

setprop net.r ril.repository notready

# mount the debugfs
mount debugfs none /sys/kernel/debug/
(skip...)
service wpa_supplicant /system/bin/wpa_supplicant -dd -Dnl80211 -iwlan0
-c /data/misc/wifi/wpa_supplicant.conf
socket wpa wlan0 dgram 660 wifi wifi
group system wifi net
disabled
oneshot

service ril-repo /system/bin/r ril-repo.sh
class main
user root
group audio
oneshot

service ril-daemon /system/bin/rild -l /system/lib/librapid-ril-core.so
-- -a /dev/ttyACM1 -n /dev/ttyACM2
class main
socket rild stream 660 root radio
socket rild-debug stream 660 radio system
user root
group radio cache inet misc audio

service pppd_data0 /system/bin/init.gprs-pppd
user root radio
group radio cache inet mise
disabled
oneshot

service pppd_data1 /system/bin/init.gprs-pppd
user root radio
group radio cache inet mise
disabled
oneshot
```

```

service pppd_term /system/bin/stop_pppd 15
    class main
    disabled
    oneshot

service pppd_kill /system/bin/stop_pppd 9
    class main
    disabled
    oneshot

# Orion InG
service ing /system/bin/ingsvcd -c /system/etc/gps/Orion.ini
    socket athrkv9988      stream 666 system system
    socket athrshmsocet   stream 666 system system
    socket athrsupl59991  stream 666 system system
    class late_start
    user root
    group gps
    oneshot
(skip...)

```

In the original rild driver, we need to modify it to allow loading the correct library.

- src\hardwareril\rild\rild.c

```

(skip...)
int main(int argc, char **argv)
{
    (skip...)
    start_uevent_monitor();
    /*
        switch (modem_type){
            case ZTE_MODEM:
                rilLibPath = REFERENCE_RIL_ZTE_PATH;
                break;

            case HUAWEI_MODEM:
            case AMAZON_MODEM:
            default:
                rilLibPath = REFERENCE_RIL_DEF_PATH;
                break;
        }
    */
    if (rilLibPath == NULL) {
        (skip...)
    }
    (skip...)
}

```

5.16.Add support of GPS

Use the driver v2.0 provided by u-blox, copy to “/src/hardware/gps”.

Modify BoardConfig.mk to use u-blox driver.

- src\device\fsl\sabresd_6dq\BoardConfig.mk

```
BOARD_HAVE_HARDWARE_GPS := true
USE_ATHR_GPS_HARDWARE := false
USE_QEMU_GPS_HARDWARE := false
```

Modify imx6.mk to copy the files that driver needs.

- src\device\fsl\imx6\imx6.mk

```
PRODUCT_COPY_FILES += \
(skip...)
    device/fsl/imx6/etc/gps.conf:system/etc/gps.conf \
    device/fsl/imx6/etc/u-blox.conf:system/etc/u-blox.conf \
```

Modify u-blox.conf to choose the correct gps port.

- src\device\fsl\imx6\etc\ublox.conf

```
UDP_SERVER_PORT      46434
SERIAL_DEVICE         /dev/ttyACM3
BAUDRATE              115200
BAUDRATE_DEF          9600
```

To use GPS, we need to type AT command. According to “Android_RIL_Souce_code_application.pdf,” we can add AT command.

- src\packages\app\Settings\AndroidManifest.xml

```
<activity android:name=".wifi.WifiStatusTest"
android:theme="@android:style/Theme.Holo.DialogWhenLarge">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
```

```

        <category
android:name="android.intent.category.DEVELOPMENT_PREFERENCE" />
        <category android:name="android.intent.category.DEFAULT"
/>

    </intent-filter>
</activity>

    <activity android:name="RilOemHookTest"
        android:label="@string/testing_RIL_OEMHook"
        android:process="com.android.phone">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category
android:name="android.intent.category.DEVELOPMENT_PREFERENCE" />
            </intent-filter>
        </activity>

```

- src\packages\app\Settings\res\values\string.xml

```

<!-- User picture title [CHAR LIMIT=30] -->
<string name="user_picture_title">Photo ID</string>

    <string name="testing_RIL_OEMHook">RIL OEM Hook Test</string>
    <string name="ril_oem_choose">Choose a RIL OEM Hook API to
test:</string>
    <string name="radio_api1">API 1 (datalen=0)</string>
    <string name="radio_api2">API 2 (datalen=1)</string>
    <string name="radio_api3">API 3 (datalen=6)</string>
    <string name="radio_api4">API 4 Type command:</string>
    <string name="radio_run">Run</string>
    <string name="ril_oem_response">Module response:</string>

<!-- Cell Broadcast settings title [CHAR LIMIT=50] -->

```

- src\packages\app\Settings\res\xml\wireless_settings.xml

```

<PreferenceScreen
    android:key="mobile_network_settings"
    android:title="@string/network_settings_title"
    android:dependency="toggle_airplane">
    <intent
        android:action="android.intent.action.MAIN"
        android:targetPackage="com.android.phone"

android:targetClass="com.android.phone.MobileNetworkSettings" />
    </PreferenceScreen>

    <PreferenceScreen
        android:title="@string/testing_RIL_OEMHook" >
        <intent
            android:action="android.intent.action.MAIN"

```

```

        android:targetPackage="com.android.settings"

        android:targetClass="com.android.settings.RilOemHookTest" />
    </PreferenceScreen>

```

5.17.Add eMMC 5.0 running on eMMC 4.4 mode

To add a workaround that can support eMMC 5.0 running on eMMC 4.4 mode:

- src\kernel_imx\drivers\mmc\core\mmc.c

```

static int mmc_read_ext_csd(struct mmc_card *card, u8 *extcsd)
{
    (skip...)
    card->ext_csd.rev = ext_csd[EXT_CSD_REV];
    if (card->ext_csd.rev > 7) {
        printk(KERN_ERR "%s: unrecognized EXT_CSD revision %d\n",
    (skip...)
    }
}

```

5.18.Fix IPU error under thermal testing

To modify **gpu3DMinClock** value in order to fix this issue based on Freescale community discussion.

- src\kernel_imx\drivers\mxl\gpu-
vivi\hal\os\linux\kernel\gc_hal_kernel_driver.c

```

module_param(showArgs, int, 0644);

int gpu3DMinClock = 3;
module_param(gpu3DMinClock, int, 0644);

```

5.19.Fix 3G incorrect signal strength info

To use GSM signal strength by default instead of using LTE signal:

- `src\frameworks\base\telephony\java\android\telephony\SignalStrength.java`

```
public int getLteLevel() {
    (skip...)
    //if(snrIconLevel != -1) return snrIconLevel;
    //if(rsrpIconLevel != -1) return rsrpIconLevel;

    If(mLteSignalStrength > 63) rssiIconLevel =
    SIGNAL_STRENGTH_NONE_OR_UNKNOWN;
}
```

5.20. Fix failure in opening the camera apk

Camera apk will fail to open if there is no input signal. Set NTSC as the default signal for v4l2 auto detection with no input signal to avoid failure in opening the camera apk.

- `src\kernel_imx\drivers\media\video\mxm\capture\adv7180.c`

```
static void adv7180_get_std(v4l2_std_id *std)
{
    (skip...)
} else {
    // give default standard to NTSC to avoid camera apk open fail
    *std = V4L2_STD_NTSC;
    idx = ADV7180_NTSC;

    /*std = V4L2_STD_ALL;
    idx = ADV7180_NOT_LOCKED;
    dev_dbg(&adv7180_data.sen.i2c_client->dev,
    // "Got invalid video standard!\n");
    }
}
```

5.21. Support AUO 10.4" touch panel

Add vendor ID and device ID for touch panel, the setting value is suggested by panel's vendor FAE and might not exactly match with the device.

- src\kernel_imx\drivers\hid\hid-ids.h

```
#define USB_VENDOR_ID_CYGNAL          0x10c4
#define USB_DEVICE_ID_CYGNAL_RADIO_SI470X      0x818a

// kenny add for AUO 10" IPCT touch panel
#define USB_DEVICE_ID_CYGNAL_TOUCH_SCREEN      0x88cd
```

- src\kernel_imx\drivers\hid\hid-core.c

```
// kenny add for AUO 10" IPCT touch panel
{ HID_USB_DEVICE(USB_VENDOR_ID_CYGNAL,
USB_DEVICE_ID_CYGNAL_TOUCH_SCREEN) },
```

- src\kernel_imx\drivers\hid\hid-multitouch.c

```
// kenny add
/* AUO 10" touch panel*/
{ .driver_data = MT_CLS_DEFAULT,
HID_USB_DEVICE(USB_VENDOR_ID_CYGNAL,
USB_DEVICE_ID_CYGNAL_TOUCH_SCREEN) },
```

Copy the touch driver and idc file to system

- src\device\fs\imx6\imx6.mk

```
PRODUCT_COPY_FILES += \
.
.
.
device/fs/common/input/Vendor_10C4_Product_88CD.idc:system/usr/idc/Vendor_10c4_Product_88cd.idc
.
.
.
kernel imx/drivers/hid/hid-multitouch.ko:system/lib/modules/hid-multitouch.ko
```

Load touch driver when system boots up

- src\device\fs\imx6\etc\init.rc

```
# kenny add for AUO 10" touch panel
insmod /system/lib/modules/hid-multitouch.ko
```

5.22. Correct navigation bar in low profile

In low resolution, ex 800x600, navigation bar will be missing.

Add a layout file for low-profile.

- `src/frameworks/base/packages/SystemUI/res/layout-land/navigation_bar.xml`

5.23. Correct kernel menuconfig that is not working

System will copy `android_defconfig` to replace our `menuconfig` when building the android. This step has been deleted.

- `src/build/core/Makefile`

```
$(TARGET_KERNEL_CONFIGURE): kernel_imx/arch/arm/configs/$(TARGET_KERNEL_DEFCONF) $(KERNEL_CONFIGURE)
- $(MAKE) -C kernel_imx $(TARGET_KERNEL_DEFCONF) $(KERNEL_ENV)
  install -D $(KERNEL_CONFIGURE) $(TARGET_KERNEL_CONFIGURE)

$(TARGET_PREBUILT_KERNEL): $(TARGET_KERNEL_CONFIGURE)
```

5.24. Fix camera recording issue

When audio output is HDMI and camera recording is running, camera will lag and recording may fail.

The root cause is that HDMI does not have audio input and it will close audio input. The flow of audio closing has some problem.

- src/external/tinyalsa/pcm.c

```
int pcm_close(struct pcm *pcm)
{
    if (pcm == &bad_pcm)
        return 0;

+   if (pcm->fd < 0) {
+       free(pcm);
+       return 0;
+   }
+ }
```

5.25.Add omxplayer package

Add freescale external codec 'omxplayer package' and modify build_bsp.sh

Add file:

- src/device/fsl-codec/*
- src/external/fsl_imx_omx/*
- script/build_bsp.sh

```
cd ${SRC_ROOT}
+ # setup external codec
+ source ${SRC_ROOT}/build/envsetup.sh && \
+ lunch ${ANDROID_CFG}-${LUNCH_TYPE} && \
+ ./switch_build_to.sh full && \
+ ./clean_obj_before_building.sh && \
+ make PRODUCT-${ANDROID_CFG}-
${LUNCH_TYPE} 2>&1 | tee build_${ANDROID_CFG}_android.log || exit 1
```

5.26.Fix volume limit issue

Fix the issue that volume cannot exceed 10

- src/frameworks/base/core/res/res/values/config.xml

```
<!--
Safe headphone volume index. When music stream volume is below this index
the SPL on headphone output is compliant to EN 60950 requirements for
portable music
players. -->
<integer name="config_safe_media_volume_index">15</integer>

<!--
Configure mobile network MTU. The standard default is set here but each c
arrier
may have a specific value set in an overlay config.xml file. -->
```

5.27. Fix hotspot issue

When enabling either wifi or hotspot, the other one cannot be used or hang. We added 3 seconds of delay time between disable and enable to solve this problem.

- src/packages/apps/Settings/src/com/android/settings/TetherSettings.java

```
if (mWifiManager.getWifiApState() == WifiManager.WIFI_AP_STATE_ENABLED) {
    mWifiManager.setWifiApEnabled(null, false);
    mWifiManager.setWifiApEnabled(mWifiConfig, true);
    mEnableWifiAp.setEnabled(false);

    new CountDownTimer(3000, 500) {
        @Override
        public void onFinish() {
            mWifiManager.setWifiApEnabled(mWifiConfig, true);
        }
        public void onTick(long millisUntilFinished) {
            mEnableWifiAp.setEnabled(false);
            mEnableWifiAp.setSummary("Reset hotspot. Please wait...");
        }
    }.start();
} else {
```

- src/packages/apps/Settings/src/com/android/settings/wifi/WifiApEnabl
er.java

```

        Settings.Global.putInt(cr, Settings.Global.WIFI_SAVED_STATE,
1);

        new CountDownTimer(3000,500){
            @Override
            public void onFinish() {
                if (mWifiManager.setWifiApEnabled(null, true)) {
                    /* Disable here, enabled on receiving success broa
dcast */
                    mCheckBox.setEnabled(false);
                } else {
                    mCheckBox.setSummary(R.string.wifi_error);
                }
            }
            public void onTick(long millisUntilFinished) {
            }
        }.start();

        return;
    }

```

- src/packages/apps/Settings/src/com/android/settings/wifi/WifiEnabler.j
ava

```

===== mSwitch.setEnabled(false);
===== if (!mWifiManager.setWifiEnabled(isChecked)) {
===== // Error
===== mSwitch.setEnabled(true);
===== Toast.makeText(mContext, R.string.wifi_error, Toast.LENGTH_SHO
RT).show();
// wait 3 sec then open wifi
new CountDownTimer(3000,500){
    @Override
    public void onFinish() {
        mSwitch.setEnabled(false);
        if (!mWifiManager.setWifiEnabled(true)) {
            // Error
            mSwitch.setEnabled(true);
            Toast.makeText(mContext, R.string.wifi_error, Toas
t.LENGTH_SHORT).show();
        }
    }
    public void onTick(long millisUntilFinished) {
    }
}.start();
} else {

```

```
// open wifi
mSwitch.setEnabled(false);
if (!mWifiManager.setWifiEnabled(isChecked)) {
    // Error
    mSwitch.setEnabled(true);
    Toast.makeText(mContext, R.string.wifi_error, Toast.LENGTH
_SHORT).show();
}
```

5.28.Delete si_LK language

In settings-> Language & input, there is a blank option. It is Sinhala(si_LK).

- src/build/target/product/languages_full.mk

```
# These are all the locales that have translations and are displayable
# by TextView in this branch.
PRODUCT_LOCALES := en_US en_IN fr_FR it_IT es_ES et_EE de_DE nl_NL cs_CZ
pl_PL ja_JP zh_CN zh_HK ru_RU ko_KR nb_NO es_US da_DK el_GR tr_TR
pt_PT pt_BR rm_CH sv_SE bg_BG ca_ES en_GB fi_FI hi_IN hr_HR hu_HU in_ID
iw_IL lt_LT lv_LV ro_RO sk_SK sl_SI sr_RS uk_UA vi_VN tl_PH ar_EG fa_IR
th_TH sw_TZ ms_MY af_ZA zu_ZA am_ET hi_IN en_XA ar_XB fr_CA km_KH lo_LA
ne_NP si_LK mn_MN hy_AM az_AZ ka_GE
```

5.29.Set default sleep time to 30 min

src/frameworks/base/packages/SettingsProvider/res/values/defaults.xml

```
<bool name="def_dim_screen">true</bool>
<integer name="def_screen_off_timeout">1800000</integer>
```

5.30.Add DualLite support

There are several files should be modified. Here is the list.

In uboot:

- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/flash_header.`
- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/mx6q_sabrelite.c`
- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/vab820_mx6dl_1g.cfg`
- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/vab820_mx6dl_2g.cfg`
- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/vab820_mx6q_1g.cfg`
- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/vab820_mx6q_2g.cfg`
- `src/bootable/bootloader/uboot-imx/board/freescale/mx6q_sabrelite/vab820_mx6q_4g.cfg`
- `src/bootable/bootloader/uboot-imx/cpu/arm_cortexa8/mx6/generic.c`
- `src/bootable/bootloader/uboot-imx/include/configs/mx6q_sabrelite_android.h`

In kernel:

- `src/kernel_imx/arch/arm/mach-mx6/board-mx6q_sabrelite.c`
- `src/kernel_imx/arch/arm/plat-mxc/include/mach/iomux-mx6dl.h`
- `src/kernel_imx/drivers/mtd/devices/m25p80.c`
- `src/kernel_imx/arch/arm/mach-mx6/cpu.c`

- `src/kernel_imx/arch/arm/plat-mxc/include/mach/mxc.h`
- `src/kernel_imx/drivers/media/video/mxc/capture/adv7180.c`

In android:

- `src/external/fsl_imx_omx/OpenMAXIL/src/component/fsl_parser/FslParser.cpp`