

# **Guideline for Integrated SmartAMP(TAS2563/TAS2781) Linux driver & Factory Tool**

**BeagleBone black/QCOM + Kernel-5.10-rt/Kernel-6.8**

# Revision history

Ver	Date	Author	Description
1.0	2022/9/14		Initial
1.1	2022/9/20		1 <sup>st</sup> round Review & rework
1.2	2022/9/30		Add tinyalsa
1.3	2022/10/9		2 <sup>nd</sup> round Review & rework
1.4	2023/02/05		Add firmware compiling
1.5	2023/08/24		Update DTS setting
1.6	2023/09/02		Add Sound card register on QCOM
1.7	2023/09/04		Add Factory tool section
1.8	2023/10/8		Add direct register setting for screen rotation
1.9	2023/10/26		Add how to set i2c address in dts
1.10	2024/4/15		Add calibrated data load for Android

# Information

tasdevice Driver	TAS2563/TAS2781
7-bit I <sup>2</sup> C Address	TAS2563: 0x4C ~ 0x4F
	TAS2781: 0x38 ~ 0x3F
Platform	Qualcomm/MTK/LSI/BeagleBone Black/AMBA
Architecture	ALSA/Tiny-ALSA
Bus type	I2C
Kernel Version	v5.10

# Release Package II | Driver Package

<b>Kconfig, Makefile</b>	Config for compiling
<b>.c, &amp; .h,</b>	Source code
<b>ti,tas2781.yaml</b>	Sample for dts setting
<b>regbin folder</b>	File to generate a sample register setting file with Regbin tool.
<b>dspfw folder</b>	File to generate a sample dsp firmware and acoustic params file with PPC3 tool.
<b>Verification report</b>	
<b>.ftcfg file</b>	Store speaker basic characterization params for speaker calibration
<b>config_dir/.hybrid-config</b>	Example of config file for compiling Linux kernel 5.10 for BeagleBone Black
<b>hybrid-compile-kon-bbb.sh</b>	Compiling script for compiling Linux kernel 5.10 for BeagleBone Black and integrated-tasdevice driver
<b>hybrid-compile-on-bbb.sh</b>	Compiling script for compiling Linux kernel 5.10 for BeagleBone Black

# Release Package II | Tool Utility

<b>FTC_tool</b>	Source code for calibration
<b>PPC3 tool</b>	Tool for sample dsp firmware and acoustic params and ftcfg file
<b>Non_Integrated_Bin_Tool_v1.3.7</b>	Tool for regbin

# Cross-compiler

- Deploy cross-compiler (shell command)

```
#sudo apt-get install gcc-arm-linux-  
gnueabi hf lzop
```

PS: If you found following installing log,  
pls update apt-get first.



installing.log

```
#apt-get update
```

- Check the compiler version

```
#arm-linux-gnueabi hf-gcc -v
```

gcc version 7.5.0 (Ubuntu/Linaro 7.5.0-  
3ubuntu1~18.04)

OR

gcc version 11.3.0 (Ubuntu 11.3.0-  
1ubuntu1~22.04)

# U-boot I

- Clone U-boot

```
$sudo -i
```

```
#git clone git://git.denx.de/u-boot.git u-boot
```

- Query the branch information

```
#git status
```

```
root@Tintin:/usr/local/BeagleBoneBlack/linux-4.19# cd u-boot/  
root@Tintin:/usr/local/BeagleBoneBlack/linux-4.19/u-boot# git status  
On branch master  
Your branch is up to date with 'origin/master'.  
  
nothing to commit, working tree clean
```

- Query all the tag information

```
#git tag
```



tag.txt

- Switch to specific tag

```
#cd u-boot
```

```
#git checkout v2017.03-rc2 -b tmp
```

```
root@Tintin:/usr/local/BeagleBoneBlack# cd u-boot/  
root@Tintin:/usr/local/BeagleBoneBlack/u-boot# ls  
Kbuild  Licenses  Makefile  api  board  common  configs  doc  dts  
Kconfig MAINTAINERS README  arch  cmd  config.mk  disk  drivers  env  
root@Tintin:/usr/local/BeagleBoneBlack/u-boot# git checkout v2017.03-rc2 -b tmp  
Checking out files: 100% (20757/20757), done.  
Switched to a new branch 'tmp'
```

- Check the difference between current tag and the specific tag

```
#git diff v2016.01
```

# U-boot II

- Download the patch for BBB

```
# wget -c https://rcn-ee.com/repos/git/u-boot-patches/v2017.03-rc2/0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch
```

```
# wget -c https://rcn-ee.com/repos/git/u-boot-patches/v2017.03-rc2/0002-U-Boot-BeagleBone-Cape-Manager.patch
```

```
# patch -p1 < 0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch
```

```
# patch -p1 < 0002-U-Boot-BeagleBone-Cape-Manager.patch
```

Run above commands in the u-boot



0001-am335x\_ev  
Env.txt-bootz-n-fi



0002-U-Boot-Bea  
ne-Cape-Manage



# Linux kernel 5.10-rt Compiling

- Download kernel 5.10-rt([GitHub - beagleboard/linux at 5.10-rt](https://github.com/beagleboard/linux))

```
# git clone https://github.com/beagleboard/linux.git
```

- Uzip the kernel package

```
# sudo unzip linux-5.10-rt.zip
```

```
# cd linux-5.10-rt/
```

- Create compile\_BBB-kernel.sh

```
#!/bin/bash
```

```
apt-get install gcc-arm-linux-gnueabi hf lzop 2>&1 | tee install.log
```

```
make clean 2>&1 | tee clean.log
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- bb.org_defconfig -j $(expr $(nproc) - 1) 2>&1 | tee config.log
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -j $(expr $(nproc) - 1) 2>&1 | tee compile.log
```



- Compile the kernel

compile\_BBB-kernel.sh

```
# chmod 777 compile_BBB-kernel.sh
```

```
# sudo ./compile_BBB-kernel.sh
```

# Obtain root file system

- Download Debian11.4 :

```
# wget -c https://rcn-ee.com/rootfs/eewiki/minfs/debian-11.4-minimal-armhf-2022-07-11.tar.xz
```

```
# wget -c https://rcn-ee.com/rootfs/eewiki/minfs/debian-11.4-minimal-armhf-2022-07-11.tar.xz.sha256sum
```

- Verify sha256sum

```
#sha256sum debian-11.4-minimal-armhf-2022-07-11.tar.xz
```

```
luminlong@luminlong-mybox:~/Linux_new_kernel/debian_package/11.4$ sha256sum debian-11.4-minimal-armhf-2022-07-11.tar.xz
07257d00340fb097a1a9fba34d00b3e2a1224035e3c412a43d55d2356670366e  debian-11.4-minimal-armhf-2022-07-11.tar.xz
```

- Uncomprocessing

```
#tar -xf debian-11.4-minimal-armhf-2022-07-11.tar.xz debian11.4
```

```
# ls ./debian11.4
```

The debian11.4 direction shows following files:

```
an-11.4-minimal-armhf-2022-07-11$ ls
armhf-rootfs-debian-bullseye.tar  user_password.list
image-builder.project
```

- By the way, the user password.list stores the user name and password

```
an-11.4-minimal-armhf-2022-07-11$ cat user_password.list
debian:temppwd
```

# Set up MicroSD card I

- Create partition in SD card
- Use fdisk command
  - #umount /dev/sdb\*
  - #sudo fdisk /dev/sdb
- Input p, check the partitions, like this:

```
Command (m for help): p
Disk /dev/sdb: 7.5 GiB, 7994343424 bytes, 15613952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd8ed8c21

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1   *      2048    411647   409600  200M  c W95 FAT32 (LBA)
/dev/sdb2           411648 15613951 15202304   7.3G 83 Linux

Filesystem/RAID signature on partition 1 will be wiped.
Filesystem/RAID signature on partition 2 will be wiped.
```

- Input d, Delete a Specified partition

```
Command (m for help): d
Partition number (1,2, default 2):

Partition 2 has been deleted.
```

# Set up MicroSD card II

- Input n, create two new partitions
- Input t, then c, change partition 1's type from 'linux' to 'W95 FAT32(LBA)'
- Input a, enable bootable flag on partition1
- Input w, write the partition table to SD card, then exit from fdisk.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

- Format partitions

```
# sudo mkfs.vfat -F 32 -n boot /dev/sdb1
```

```
# sudo mkfs.ext4 -L rootfs /dev/sdb2
```

```
luminlong@luminlong-mybox:~$ sudo mkfs.vfat -F 32 -n boot /dev/sdb1
mkfs.fat 4.1 (2017-01-24)
mkfs.fat: warning - lowercase labels might not work properly with DOS or Windows
luminlong@luminlong-mybox:~$ sudo mkfs.ext4 -L rootfs /dev/sdb2
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 1900288 4k blocks and 475136 inodes
Filesystem UUID: 1e2b67e2-a9ec-46cb-8de0-a1f9902123a5
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

- Mount /dev/sdb2 to /media/rootfs/

```
# sudo mkdir -p /media/rootfs/
```

```
# sudo mount /dev/sdb2 /media/rootfs/
```

# Install & Backup U-Boot

## Install Bootloader

```
# sudo dd if=./u-boot/MLO of=/dev/sdb1 count=2 seek=1  
bs=128k
```

```
# sudo dd if=./u-boot/u-boot-dtb.img of=/dev/sdb1 count=4  
seek=1 bs=384k
```

## Backup Bootloader (optional)

```
# sudo mkdir -p /media/rootfs/opt/backup/u-boot/
```

- /media/rootfs/opt/backup/u-boot/ is the folder mounted with BBB folder mentioned in previous page

```
# sudo cp -v ./u-boot/MLO /media/rootfs/opt/backup/u-boot/
```

```
# sudo cp -v ./u-boot/u-boot-dtb.img  
/media/rootfs/opt/backup/u-boot/
```

# Editing of uEnv.txt

- **Please Edit the uEnv.txt as follows:**

```
loadaddr=0x82000000
fdtaddr=0x88000000
rdaddr=0x88080000
initrd_high=0xffffffff
fdt_high=0xffffffff
#for single partitions:
mmcroot=/dev/mmcblk0p1
```

```
loadximage=load mmc 0:1 ${loadaddr} /boot/vmlinuz-${uname_r}
loadxfdt=load mmc 0:1 ${fdtaddr} /boot/dtbs/${uname_r}/${fdtfile}
loadxrd=load mmc 0:1 ${rdaddr} /boot/initrd.img-${uname_r}; setenv rdsiz ${filesize}
loaduEnvtxt=load mmc 0:1 ${loadaddr} /boot/uEnv.txt ; env import -t ${loadaddr} ${filesize};
loadall=run loaduEnvtxt; run loadximage; run loadxfdt;
mmccargs=setenv bootargs console=tty0 console=${console} ${optargs} ${cape_disable} ${cape_enable} root=${mmcroot}
rootfstype=${mmcrootfstype} ${cmdline}
```

```
uenvcmd=run loadall; run mmccargs; bootz ${loadaddr} - ${fdtaddr};
```

- **Copy uExt.txt to /media/rootfs/boot**

```
# sudo cp ./uExt.txt /media/rootfs/boot
```

# Install Kernel and Root File System

- Add Kernel version into uEnv.txt

```
# export kernel_version=linux5.10.120-rt
# sudo sh -c "echo 'uname_r=${kernel_version}' >>
/media/rootfs/boot/uEnv.txt"
```

- Install root file system

```
# sudo tar xfvp ./denian/*-*-armhf-*/armhf-rootfs-*.tar -
C /media/rootfs/
# sync
# sudo chown root:root /media/rootfs/
# sudo chmod 755 /media/rootfs/
```

- Copy zImage & dtb

```
# sudo cp -v ./linux5.10-rt/arch/arm/boot/zImage
/media/rootfs/boot/vmlinuz-${kernel_version}
# sudo mkdir -p /media/rootfs/boot/dtbs/${kernel_version}/
#sudo cp ./linux5.10-rt/arch/arm/boot/dts/am335x-
boneblack.dtb /media/rootfs/boot/dtbs/${kernel_version}/
```

- Install kernel Modules (optional)

```
# cd ./linux5.10-rt/
# sudo make ARCH=arm CROSS_COMPILE=arm-linux-
gnueabihf- modules_install
INSTALL_MOD_PATH=/media/rootfs -j16
```

- File system Table(/etc/fstab)

```
#sudo sh -c "echo '/dev/mmcblk0p1 / auto errors=remount-ro
0 1' >> /media/rootfs/etc/fstab"
```

# Network configuration (optional)

- Edit: /etc/network/interfaces

```
# sudo vi /media/rootfs/etc/network/interfaces
```

Add below:

```
#/etc/network/interfaces
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet dhcp
```

- Enable eth0

```
# sudo vi /media/rootfs/etc/udev/rules.d/70-  
persistent-net.rules
```

Add below:

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",  
ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",  
NAME="eth0"
```



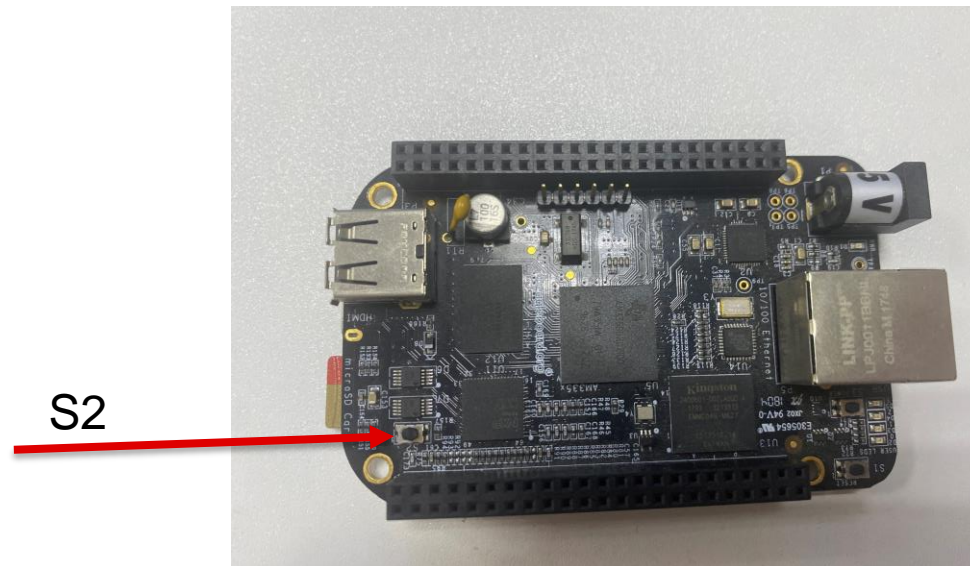
# Remove microSD/SD card

- Remove microSD/SD card

```
# sync
```

```
# sudo umount /media/rootfs
```

- Insert the SD card on BBB
- Press and hold button on S2, and Power on the BBB



# Device Tree | I2C - mono

- Configuration in DTS:

```
i2c2_pins: pinmux_i2c2_pins {
    pinctrl-single,pins = <
        AM33XX_PADCONF(AM335X_PIN_UART1_CTSN, PIN_INPUT_PULLUP, MUX_MODE3)
        AM33XX_PADCONF(AM335X_PIN_UART1_RTSN, PIN_INPUT_PULLUP, MUX_MODE3)
    >;
};
```

```
76 &i2c2 {
77     pinctrl-names = "default";
78     #address-cells = <1>;
79     #size-cells = <0>;
80     status = "okay";
81     clock-frequency = <400000>;
82     pinctrl-0 = <&i2c2_pins>;
83     tasdevice: tasdevice@38 {
84         status = "okay";
85         #sound-dai-cells = <0>;
86         compatible = "ti,tas2781";
87         reg = <0x38>;
88         reset-gpios = <&gpio1 10 GPIO_ACTIVE_HIGH>;
89     };
90 };
```

Two sides must be the same i2c address

- How to check device

If the I2C is successfully registered, Check device through below commands, 2-0038 is the registered device. 0x38 is the i2c address for slave device.

```
# ls /sys/bus/i2c/device/
```

```
debian@beaglebone:~$ ls /sys/bus/i2c/devices/
0-0024 0-0050 2-0038 i2c-0 i2c-2
```

- DTS file

PS: This guideline only offer the interrupt setting for BeagleBone Black. For other platform, kindly consult the platform vendor.



am335x-bonebla  
ck-hdmi.dtsi

# Device Tree | I2C – multiple devices

- Configuration in DTS:

```
i2c2_pins: pinmux_i2c2_pins {
    pinctrl-single,pins = <
        AM33XX_PADCONF(AM335X_PIN_UART1_CTSN, PIN_INPUT_PULLUP, MUX_MODE3)
        AM33XX_PADCONF(AM335X_PIN_UART1_RTSN, PIN_INPUT_PULLUP, MUX_MODE3)
    >;
};
```

```
76 &i2c2 {
77     pinctrl-names = "default";
78     #address-cells = <1>;
79     #size-cells = <0>;
80     status = "okay";
81     clock-frequency = <400000>;
82     pinctrl-0 = <&i2c2_pins>;
83     tasdevice: tasdevice@38 {
84         status = "okay";
85         #sound-dai-cells = <0>;
86         compatible = "ti,tas2781";
87         reg = <0x38>, <0x3f>, <0x3b>;
88         reset-gpios = <&gpio1 10 GPIO_ACTIVE_HIGH>;
89     };
90 };
```

Two sides must be the  
same i2c address

- How to check device

If the I2C is successfully registered, Check device through below commands, 2-0038 is the registered device. 0x38 is the i2c address for slave device.

```
# ls /sys/bus/i2c/device/
```

```
debian@beaglebone:~$ ls /sys/bus/i2c/devices/
0-0024 0-0050 2-0038 i2c-0 i2c-2
```

- DTS file

PS: This guideline only offer the interrupt setting for BeagleBone Black. For other platform, kindly consult the platform vendor.



am335x-boneblack-hdmi.dtsi

# Device Tree | SPI

- Configuration in DTS:

```
bb_spi0_pins: pinmux_bb_spi0_pins {
    pinctrl-single,pins = <
        AM33XX_PADCONF(AM335X_PIN_SPI0_SCLK, PIN_INPUT, MUX_MODE0)
        AM33XX_PADCONF(AM335X_PIN_SPI0_D0, PIN_INPUT, MUX_MODE0)
        AM33XX_PADCONF(AM335X_PIN_SPI0_D1, PIN_INPUT, MUX_MODE0)
        AM33XX_PADCONF(AM335X_PIN_SPI0_CS0, PIN_INPUT, MUX_MODE0)
    >;
};
```

```
76 &spi0 {
77     #address-cells = <1>;
78     #size-cells = <0>;
79     pinctrl-names = "default";
80     pinctrl-0 = <&bb_spi0_pins>;
81     tasdevice:tasdevice@0 {
82         #sound-dai-cells = <0>;
83         compatible = "ti,tas2781";
84         symlink = "bone/spi/0.0";
85         reg = <0>;
86         reset-gpios = <&gpio1 10 GPIO_ACTIVE_HIGH>;
87         status = "okay";
88         spi-max-frequency = <400000>;
89     };
90 };
```

- How to check device.

If the SPI is successfully registered, Check device through below commands, spi0.0 is the registered device.

```
# ls /sys/bus/spi/device/
```

```
debian@beaglebone:~$ ls /sys/bus/spi/devices/
spi0.0
```

- DTS file

PS: This guideline only offer the interrupt setting for BeagleBone Black. For other platform, kindly consult the platform vendor.



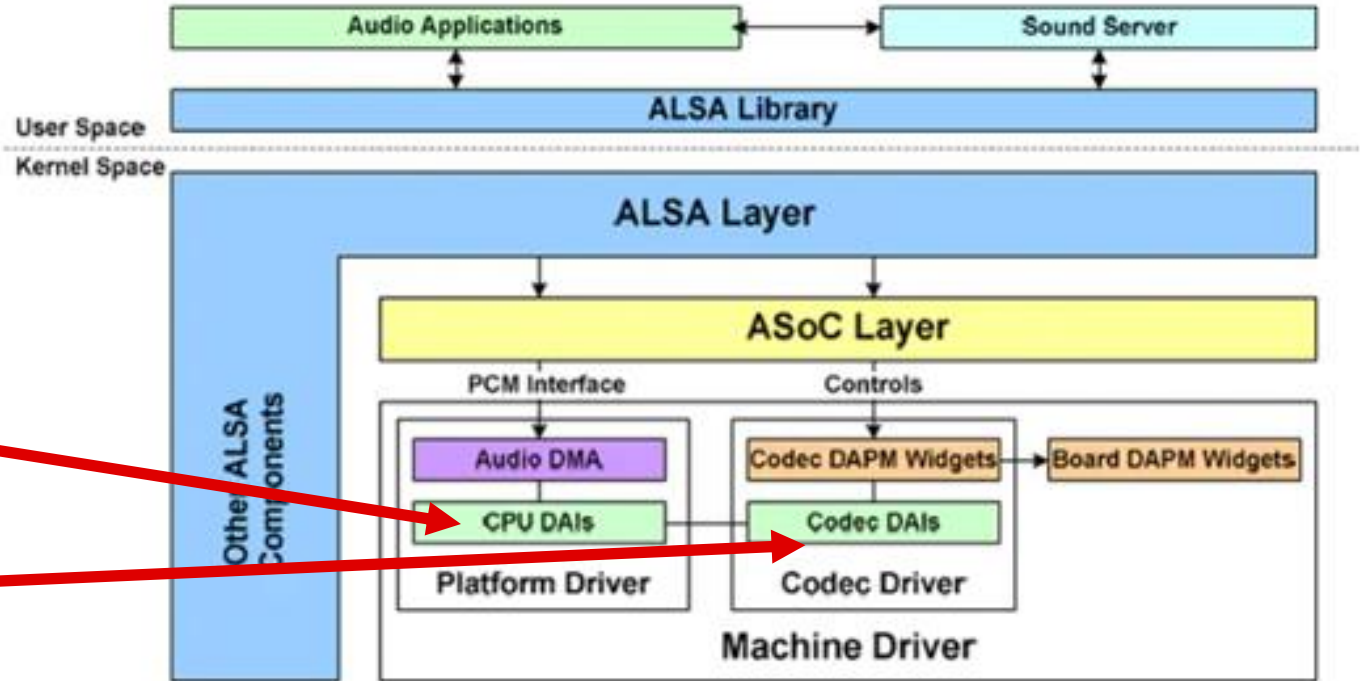
am335x-bonebla  
ck-hdmi.dtsi

# Device Tree | Sound card

```
&mcasp0 {
    #sound-dai-cells = <0>;
    pinctrl-names = "default";
    pinctrl-0 = <&mcasp0_pins>;
    status = "okay";
    op-mode = <0>; /* MCASP_IIS_MODE */
    tdm-slots = <8>;
    serial-dir = < /* 0: INACTIVE, 1: TX, 2: RX */
        2 2 1 0
    >;
    tx-num-evt = <32>;
    rx-num-evt = <32>;
};

sound {
    compatible = "simple-audio-card";
    simple-audio-card,name = "TI BeagleBone Black";
    simple-audio-card,dai-link@0 {
        format = "dsp_a";
        bitclock-master = <&sound0_master>;
        frame-master = <&sound0_master>;
        sound0_master: cpu {
            sound-dai = <&mcasp0>;
            clocks = <&clk_mcasp0>;
        };

        codec {
            sound-dai = <&tasdevice>;
        };
    };
};
```



# Kconfig & Makefile

## Kconfig in sound/soc/codecs

```
99      imply SND_SOC_HDA_HDA
100     imply SND_SOC_ICS43432
101     imply SND_SOC_INNO_RK3036
102     imply SND_SOC_INTEGRATED_TASDEVICE
103     imply SND_SOC_ISABELLE
104     imply SND_SOC_JZ4740_CODEC
105     imply SND_SOC_JZ4725B_CODEC

1788
1789 config SND_SOC_TPA6130A2
1790     tristate "Texas Instruments TPA6130A2 headphone amplifier"
1791     depends on I2C
1792
1793     comment "tas2563"
1794     source "sound/soc/codecs/tas2563/Kconfig"
1795 endmenu
1796
```

## Makefile in sound/soc/codecs

```
611 # Amp
612 obj-$(CONFIG_SND_SOC_MAX9877) += snd-soc-max9877.o
613 obj-$(CONFIG_SND_SOC_MAX98504) += snd-soc-max98504.o
614 obj-$(CONFIG_SND_SOC_SIMPLE_AMPLIFIER) += snd-soc-simple-amplifier.o
615 obj-$(CONFIG_SND_SOC_TPA6130A2) += snd-soc-tpa6130a2.o
616 obj-$(CONFIG_SND_SOC_INTEGRATED_TASDEVICE) += tas2563/
617
```

# Modification in defconfig for I2C interface

- Check whether following items have been enabled in arch/arm/configs/ti\_sdk\_am3x\_release\_defconfig, if not, enable them as following
  - CONFIG\_I2C=y
  - CONFIG\_I2C\_GPIO=y
  - CONFIG\_I2C\_OMAP=y
  - CONFIG\_I2C\_MUX=y

# Modification in defconfig for SPI interface

- Check whether following items have been enabled in arch/arm/configs/ti\_sdk\_am3x\_release\_defconfig, if not, enable them as following
  - CONFIG\_SPI=y
  - CONFIG\_SPI\_GPIO=y
  - CONFIG\_SPI\_SPIDEV=y
  - CONFIG\_SPI\_MASTER=y



# Modification in defconfig for Sound Card

- Add following items in arch/arm/configs/ti\_sdk\_am3x\_release\_defconfig.
  - CONFIG\_SOUND=y
  - CONFIG\_SND=y
  - CONFIG\_SND\_SOC=y
  - CONFIG\_SND\_SIMPLE\_CARD=y
  - CONFIG\_SND\_SOC\_INTEGRATED\_TASDEVICE=y

# Confirm where the firmware file store in the target device

- Below kernel 5.10
  - The path storing the bin file is defined in kernel/drivers/base/firmware\_class.c

```
static const char * const fw_path[] = {
    fw_path_para,
    "/system/vendor/firmware", /*defined
only in Android system*/
    "/system/etc/firmware", /*defined only
in Android system*/
    "/lib/firmware/updates/"
    UTS_RELEASE,
    "/lib/firmware/updates",
    "/lib/firmware/" UTS_RELEASE,
    "/lib/firmware" };
```
- Above and including kernel 5.10
  - The default path storing the bin file is defined in kernel/drivers/base/firmware\_loader/main.c.

```
static const char * const fw_path[] = {
    fw_path_para,
    "/lib/firmware/updates/"
    UTS_RELEASE,
    "/lib/firmware/updates",
    "/lib/firmware/" UTS_RELEASE,
    "/lib/firmware" };
```

## Add bin file into zImage

- During debug, pushing the bin file into proper folder of the target device is more convenient than compiling into image.
- Detailed see [Firmware search paths — The Linux Kernel documentation](#).
- Other firmware paths can be defined in ueventd.rc

83 lines (70 sloc) | 3.09 KB

```
1 import /vendor/etc/ueventd.rc
2 import /odm/etc/ueventd.rc
3
4 firmware_directories /etc/firmware/ /odm/firmware/ /vendor/firmware/ /firmware/image/
5 uevent_socket_rcvbuf_size 16M
6
7 subsystem graphics
8     devname uevent_devpath
9     dirname /dev/graphics
```



ueventd.rc

# Firmware Setting in defconfig

- Compile the bin file into image.
  - Add following settings into defconfig file

```
CONFIG_FW_LOADER=y
CONFIG_FW_LOADER_PAGED_BUF=y
CONFIG_EXTRA_FIRMWARE="tas2781-4amp-reg.bin tas2781-4amp-dsp.bin regulatory.db
regulatory.db.p7s am335x-pm-firmware.elf am335x-bone-scale-data.bin am335x-evm-scale-data.bin am43x-
evm-scale-data.bin"
CONFIG_EXTRA_FIRMWARE_DIR="firmware"
```

    - ❑ CONFIG\_EXTRA\_FIRMWARE\_DIR is path where the bin file is stored into the kernel root. “firmware” is the relative path in the compiling base path.
    - ❑ CONFIG\_EXTRA\_FIRMWARE is the name of firmware name. if multiple bin files are required, input the file name format like **CONFIG\_EXTRA\_FIRMWARE = “a.bin b.bin c.bin”**
      - ✓ Do not forget copy the firmware file into “firmware” folder in the compiling base folder
    - ❑ If this method is enabled, the fw will be loaded during system bootup.
  - During debug, pushing the bin file into proper folder of the target device is more convenient than compiling into image. Once compiling into zImage, every time the bin file is changed, it should be recompiled into zImage instead of copied into the *fw\_path*

*PS: How to compile the firmware into Android, please refer to [Appendix V | Compile firmware into Android system](#)*

# Firmware & ftcfg file naming

file	File name	Remark
regbin	tas2781-1amp-reg.bin tas2563-4amp-reg.bin	Generated by regbin tool
Dsp firmware & params	tas2781-1amp-dsp.bin tas2563-4amp-dsp.bin	Generated by ppc3
ftcfg	tas2781-a.ftcfg tas2781-b.ftcfg tas2563-a.ftcfg tas2563-b.ftcfg tas2563-c.ftcfg tas2563-d.ftcfg ...	
Calibrated file	tas2781-a-cal.bin tas2781-b-cal.bin ...	Generated by <a href="#">ftc tool</a>

# Compile Image with tas2781 driver

- Generate .config
  - #sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- ti\_sdk\_am3x\_release\_defconfig
- Compile the image
  - #sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- zImage dtbs -j16

PS:

1. There's a config file with the name [.hybrid\\_config](#). It can be used directly for compiling without ,config generation.
2. [hybrid-compile-ko-on-bbb.sh](#) will compile the driver as a module, that is ko file.
3. [hybrid-compile-on-bbb.sh](#) will compile the kernel image without the driver.



hybrid.config



hybrid-compile-ko-on-bbb.sh



hybrid-compile-on-bbb.sh

# Audio card registration

- BBB use mcasp0 to output the audio data. If the DTS was configured correctly, the driver will create sound card device and kcontrols, we can check below commands to confirm that.

```
# ls /dev/snd
```

```
# ls /sys/bus/i2c/devices/2-0038
```

```
# cat /proc/asound/pcm
```

```
# cat /proc/asound/cards
```

```
debian@beaglebone:~$ ls /dev/snd/
by-path  controlC0  pcmC0D0c  pcmC0D0p  timer
debian@beaglebone:~$ ls /sys/bus/i2c/devices/2-0038
act_addr  driver      fwload     of_node   regbininfo_list  subsystem
dev_addr  dspfw_config  modalias  power     regcfg_list      uevent
devinfo   dspfwinfo_list  name      reg       regdump
debian@beaglebone:~$ cat /proc/asound/pcm
00-00: davinci-mcasp.0-tasdevice_codec tasdevice-codec.2-0038-0 : davinci-mcasp.0-tasdevice_codec tasdevice-codec.2-0038-0 : playback 1 : capture 1
debian@beaglebone:~$ cat /proc/asound/cards
 0 [Black          ]: TI_BeagleBone_B - TI BeagleBone Black
                          TI BeagleBone Black
```

# Driver nodes I | Introduction

- In order to debug driver freely, several driver nodes have been defined,

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ ls
act_addr  driver      fwload      of_node     regbininfo_list  subsystem
dev_addr  dspfw_config  modalias    power       regcfg_list      uevent
devinfo   dspfwinfo_list  name        reg         regdump
```

- Check below path to access above driver nodes
  - /sys/bus/i2c/devices/2-0038
  - /sys/class/i2c-adapter/i2c-2/2-0038
  - /sys/class/i2c-dev/i2c-2/device/2-0038



# Driver nodes II | fwload/i2caddr/devinfo

- #echo > fwload

Illustration: Use for debug if firmware has not been compiled into rootfs.

```
root@am335x-evm:/sys/bus/i2c/devices/2-0048# echo>fwload
root@am335x-evm:/sys/bus/i2c/devices/2-0048# [ 192.697698] pcmdevice-codec 2-0048: fwload: count = 1
[ 192.719134] pcmdevice-codec 2-0048: pcmdev: regbin_ready start
[ 192.719184] pcmdevice-codec 2-0048: nconfig = 1
[ 192.719214] pcmdevice-codec 2-0048: img_sz = 436 total_config_sz = 144 offset = 292
[ 192.719336] pcmdevice-codec 2-0048: Firmware init complete
```

- #amixer contents

```
debian@beaglebone:/sys/class/i2c-dev/i2c-2/device/2-0038$ amixer contents
numid=3,iface=MIXER,name='Configuration'
; type=INTEGER,access=rw-----,values=1,min=0,max=0,step=0
[ 7044.116010] tasdevice-codec 2-0038: tasdevice_info_dsp: max program num = 1
: values=0
numid=1,iface=MIXER,name='Program'
; type=INTEGER,access=rw-----,values=1,min=0,max=255,step=0
: values=1
numid=2,iface=MIXER,name='TASDEVICE Profile id'
; type=INTEGE[ 7044.116046] tasdevice-codec 2-0038: tasdevice_info_dsp: max configuration num = 2
R,access=rw-----,values=1,min=0,max=6,step=0
: values=0
```

- #cat dev\_addr

Illustration: show the active i2c address

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ cat dev_addr
ndev:1
addr0:38
```

- #cat devinfo

Illustration: Get the basic information of audio device on the board

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ cat devinfo
No.    DevTyp  Addr
0      audev   0x38
```

# Driver nodes II | reg

- `#echo chn 0xBK 0xPG 0xRG 0xXX > reg`

Illustration: Write a value to a certain register

- chn is channel no, must be 1-digital
- BK, PG, RG & XX must be 2-digital HEX
- eg: `0 0x00 0x00 0x05 0x07 > reg`

- `#cat reg`

Illustration: Read back the value from the register which have been echoed before

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ echo 0 0x00 0x00 0x05 0x07 > reg
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ cat reg
i2c - addr: 0x38
Chn0B0x00P0x00R0x05:0x07
```

# Driver nodes III | regdump

- `#echo chn 0xBK 0xPG > regdump`

Illustration: the command dump all the registers of the specific page

- chn is channel no, must be 1-digital
- BK & PG must be 2-digital HEX

- `#cat regdump`

Illustration: run the echo command, show the 7-bit i2c address of the chip and dump the registers



PAGE00-regdum  
p



PAGE01-regdum  
p

# Driver nodes IV | regbininfo\_list &

- #cat regbininfo\_list

Illustration: list the regbin version and dump the name of all the audio cases from regbin file, If wanted detailed info from specific audio case, kindly use the drive node **regcfg\_list**

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ cat regbininfo_list
Regbin File Version: 0x0105
conf 00: I2S-16bit-echoreference-slot0
conf 01: I2S-16bit-echoreference-slot2
conf 02: I2S-16bit-IV-slot0
conf 03: TDM-16bit-echoreference-slot0
conf 04: dsp-16bit-slot0
conf 05: echo-ref
```

- #cat dspfwinfo\_list

Illustration: list the detail of fwdsp bin file and dump the program name and configuration name.

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ cat dspfwinfo_list
tas2781_dsp.bin
Git format
Generated by PPC3V0x14200

ndev: 1
mnPrograms: 1
    ProgramName:    Tuning Mode
mnConfigurations: 2
    nConfig:0
        ConfigName:configuration_audio_Tuning Mode_48 KHz_s1_0  Program:0x00
    nConfig:1
        ConfigName:calibration_Tuning Mode_48 KHz_s1_0  Program:0x00
```

# Driver nodes V | regcfg\_list

```
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ echo 00 >regcfg_list
debian@beaglebone:/sys/bus/i2c/devices/2-0038$ [ 8108.293276] tasdevice-codec 2-0038: regcfg: count = 3
[ 8108.293320] tasdevice-codec 2-0038: [regcfg_list]cfg= 0, cnt=3

debian@beaglebone:/sys/bus/i2c/devices/2-0038$ cat regcfg_list
Conf 00: I2S-16bit-echo-reference-slot0
block type:PRE_POWER_UP device idx = 0x00
  SINGLE BYTE:
    BOOK0x00 PAGE0x00 REG0x0e VALUE = 0xc4
    BOOK0x00 PAGE0x00 REG0x0f VALUE = 0x40
    BOOK0x00 PAGE0x00 REG0x5c VALUE = 0xd9
    BOOK0x00 PAGE0x00 REG0x0d VALUE = 0x01
    BOOK0x00 PAGE0x00 REG0x10 VALUE = 0x04
    BOOK0x00 PAGE0x00 REG0x16 VALUE = 0x40
    BOOK0x00 PAGE0x01 REG0x17 VALUE = 0xc8
  BURST:
    BOOK0x00 PAGE0x04
    REG0x30 = 0x00 REG0x31 = 0x00 REG0x32 = 0x00 REG0x33 = 0x01
  BURST:
    BOOK0x00 PAGE0x08
    REG0x18 = 0x00 REG0x19 = 0x00 REG0x1a = 0x00 REG0x1b = 0x00
  SINGLE BYTE:
    BOOK0x00 PAGE0x08 REG0x1c VALUE = 0x00
  BURST:
    BOOK0x00 PAGE0x08
    REG0x28 = 0x40 REG0x29 = 0x00 REG0x2a = 0x00 REG0x2b = 0x00
  BURST:
    BOOK0x00 PAGE0x0a
    REG0x48 = 0x00 REG0x49 = 0x00 REG0x4a = 0x00 REG0x4b = 0x00
  BURST:
    BOOK0x00 PAGE0x0a
    REG0x58 = 0x40 REG0x59 = 0x00 REG0x5a = 0x00 REG0x5b = 0x00
  SINGLE BYTE:
    BOOK0x00 PAGE0x00 REG0x02 VALUE = 0x00
block type:PRE_SHUTDOWN device idx = 0x00
  SINGLE BYTE:
    BOOK0x00 PAGE0x00 REG0x02 VALUE = 0x02
```

- #echo CG > regcfg\_list
  - CG is conf NO, it should be 2-digital decimal
  - eg: echo 00 > regcfg\_list
- #cat regcfg\_list
  - Illustration: dump the register setting of the audio case specified by echo command

# Driver nodes VI | force\_fw\_load\_chip

- `#echo > force_fw_load_chip`
  - Illustration: Force to redownload the dsp firmware next playback
- `#cat force_fw_load_chip`
  - Illustration: get the dsp firmware downloading information

# Bypass mode

- Write a shell script to test bypass mode. Use **#amixer contents** to check the kcontrol list, then set the parameters we used. “Program” is 1 in bypass mode.
  - PS: For details about regbin configuration, please see the appendix. If set bypass mode, Setting Configuration is unnecessary.

## Shell script with Alsa

```
#!/bin/bash
amixer cset numid=1,iface=MIXER,name="Program" 1
amixer cset numid=2,iface=MIXER,name='TASDEVICE Profile id' 0
aplay --device="hw:0,0" test.wav &
sleep 1
amixer cset numid=2,iface=MIXER,name='TASDEVICE Profile id' 5
arecord -c 2 -f S16_LE -r 48000 -d 15 --device="hw:0,0" 16bit-echoref.wav &
```

1. If dsp bin file is not loaded, “Program” kcontrol can’t work, kindly ignore.
2. 0 will enable dsp, others will disable the dsp.

## Shell script with Tinyalsa

```
#!/bin/bash
tinymix "Program" 1
tinymix "TASDEVICE Profile id" 0
tinyplay test.wav &
sleep 1
tinymix 'TASDEVICE Profile id' 5
tinycap 16bit-echoref.wav &
```

# Tuning mode

- Write a shell script to test tuning mode. “Program” is 0 when using tuning mode, then select the configuration number.
  - PS: For details about fwdsp bin file, please use “**cat dspfwinfo\_list**”.

## Shell script with Alsa

```
#!/bin/bash
amixer cset numid=1,iface=MIXER,name="Program" 0
amixer cset numid=3,iface=MIXER,name="Configuration" 0
amixer cset numid=2,iface=MIXER,name='TASDEVICE Profile id' 4
aplay --device="hw:0,0" test.wav &
sleep 1
amixer cset numid=2,iface=MIXER,name='TASDEVICE Profile id' 5
arecord -c 2 -f S16_LE -r 48000 -d 15 --device="hw:0,0" 16bit-echoref.wav &
```


















































## Shell script with Tinyalsa

```
#!/bin/bash
tinymix "Program" 0
tinymix "Configuration" 0
tinymix "TASDEVICE Profile id" 4
tinyplay test.wav &
sleep 1
tinymix 'TASDEVICE Profile id' 5
tinycap 16bit-echoref.wav &
```



# Example of Regbin file | | tas2563-4amp-reg.json

- In jsn/regbin, there's tas2563-4amp-reg.json.
  - It defines 13 audio cases in Configuration column and 4 pieces of tas2563 in blcoks.
  - The order of device is define in devicetree. See ti,tas2781.yaml for details
  - Sometimes there's difference the audio configs(or audio cases) in between regbin file and dsp file, so audio\_hal will maintain a mapping table to connect both.

Configurations		Blocks	
Music degree 0	  	PRE_SHUTDOWN - Common	 
calibration	  	PRE_POWER_UP - Dev 1	 
voice call	  	PRE_POWER_UP - Dev 3	 
ringtone	  	PRE_POWER_UP - Dev 2	 
attach	  	PRE_POWER_UP - Dev 4	 
spk1 bypass	  		
spk2 bypass	  		
spk3 bypass	  		
spk4 bypass	  		
Music degree 90	  		
Music degree 180	  		
Music degree 270	  		
bypass	  		

# Example of Regbin file I | Mapping Table I

No.	Audio Cases	Regbin No.	Program No. in dsp firmware	Configure No. in dsp firmware
0	Music - default, degree 0° (dev1 & dev2 - left; dev3 & dev4 - right)	0	0	0
1	Calibration	1	0	1
2	Voice call-16bit	2	0	2
3	ringtone	3	0	3
4	Attach	4	0	4
5	SPK1-only (bypass for electrical test)	5	1	NA
6	SPK2-only (bypass for electrical test)	6	1	NA
7	SPK3-only (bypass for electrical test)	7	1	NA
8	SPK4-only (bypass for electrical test)	8	1	NA
9	Music - default, degree 90° (dev2 & dev3 - left; dev4 & dev1 - right)	9	0	0
10	Music - default, degree 180° (dev3 & dev4 - left; dev1 & dev2 - right)	10	0	0
11	Music - default, degree 270° (dev4 & dev1 –left; dev2 & dev3 - right)	11	0	0
12	Bypass (4amps/32bit)	12	1	NA

# Example of Regbin file I | Mapping Table II























- How to directly set for screen rotation.
  - Use kcontrol “TASDEVICE Rotation id” to set one of the four directions.

PS: the value in the columns “the configure No. in dsp firmware” depends on the dsp.bin file used in your current device. Here are the virtual values.

No.	Audio Cases	Regbin No.	Program No. in dsp firmware	Configure No. in dsp firmware
13	Direct rotation degree 0° (dev1 & dev2 - left; dev3 & dev4 - right)	13	NA	NA
14	Direct rotation degree 90° (dev2 & dev3 - left; dev4 & dev1 - right)	14	NA	NA
15	Direct rotation degree 180° (dev3 & dev4 - left; dev1 & dev2 - right)	15	NA	NA
16	Direct rotation degree 270° (dev4 & dev1 –left; dev2 & dev3 - right)	16	NA	NA
17	SPK1-only (dsp for acoustic test)	17	0	0
18	SPK2-only (dsp for acoustic test)	18	0	0
19	SPK3-only (dsp for acoustic test)	19	0	0
20	SPK4-only (dsp for acoustic test)	20	0	0
21	Voice call-32bit	21	0	2
22	Bypass(4amps/16bit)	22	NA	NA

# Example of Regbin file II | tas2563-1amp-reg.json

- In jsn/regbin, there's tas2563-1amp-reg.json.
  - It defines 5 audio cases in Configuration column and 1 pieces of tas2563 in blcoks.
  - Sometimes there's difference the audio configs(or audio cases) in between regbin file and dsp file, so audio\_hal will maintain a mapping table to connect both.

Configurations		Blocks	
00-Music 16bit-16kHz-4slot	  	PRE_SHUTDOWN - Dev 1	 
01-calibration-auto-rate-16bit	  	PRE_POWER_UP - Dev 1	 
02-spk bypass -16k- electrical test	  		
03-bypass-auto-rate-16bit	  		
04-Music-16bit-auto-rate-i2s	  		
05-pdm-rec-i2s-48kHz-32bit-tx-slot-0-1-mic...	  		

# Example of Regbin file II | Mapping Table

PS: the value in the columns “the configure No. in dsp firmware” depends on the dsp.bin file used in your current device. Here are the virtual values.

No.	Audio Cases	Regbin No.	Program No. in dsp firmware	Configure No. in dsp firmware
0	00-Music 16bit-16kHz-4slot	0	0	0
1	01-calibration-auto-rate-16bit	1	0	1
2	02-spk bypass -16k- electrical test	2	1	NA
3	03-bypass-auto-rate-16bit	3	1	NA
4	04-Music-16bit-auto-rate-i2s	4	0	2
5	05-pdm-rec-i2s-48kHz-32bit-tx-slot-0-1-mic-slot-3-ref	5	0	3

# FACTORY TOOL

# FTC tool I

- TAS2563/TAS2781 are smartamp, which need calibrate each speaker impedance at room temperature(about 20~25 degrees centigrade for speaker protection in dsp mode.
  - Every device must have calibration in the factory. If calibration failed, the device must have failure analysis, such as short/open circuit, wrong wire impedance, speaker issue, or deployment issue, etc.
  - Any device changed either smartamp or speaker, must have recalibration.

# FTC tool II

- ftcfg file, is the file to store the spk basic params for speaker calibration
  - File name style:  
[chip name]-[chip number].ftcfg
    - a is first chip, b is second chip, ....
    - all the letter in the file name is lowercase
    - For example, tas2781-a.ftcfg, tas2563-a.ftcfg, tas2781-b.ftcfg
    - the file name is defined in tasdevice.c, see, the right column

```
static const char *ftcfg[] = {  
    "%s-a.ftcfg",  
    "%s-b.ftcfg",  
    "%s-c.ftcfg",  
    "%s-d.ftcfg"  
};
```

- Put ftcfg file (Generated by ppc3) into directory:
  - Linux: /lib/firmware
  - Android: /mnt/vendor/persist/audio/PS: the macro  
TASDEVICE\_CAL\_CFG\_PATH in  
tasdevice-rtc.h ought to match the Operating  
System.



# FTC tool III

- Run FTC tool with root permission
  - # sudo ./tasdevice-fts
- After the calibration process finished, the results save into /home/audio/ti/tasdevice-cal.txt

```
debian@beaglebone:~$ cat /home/audio/ti/tas2781_cal.txt
Ambient temperature = 20.00

Dev[0] Re = 13.41
Dev[0] rms_pow  = 0x00000000
Dev[0] t_limit  = 0x00000000
Dev[0] Result = 0x1
```

# Calibration

- It's necessary to enable IV sense in PPC3.
- Write a shell script to test calibration. Select the **calibration\_tuning\_mode** in “configuration”.
- Playing silence during calibration
- If calibration is successful, tas2563-0x4c-cal.bin/tas2563-0x4d-cal.bin/tas2563-0x4e-cal.bin /tas2781-0x3c-cal.bin/tas2781-0x3f-cal.bin, etc will be saved into
  - Andoird: /system/vendor/firmware/
  - Linux: /lib/firmware/

PS: the macro TASDEVICE\_CAL\_BIN\_PATH in tasdevice-rtc.h ought to match the Operating System.

## Shell script with Alsa

```
#!/bin/bash
amixer cset numid=1,iface=MIXER,name="Program" 0
amixer cset numid=3,iface=MIXER,name="Configuration" 1
amixer cset numid=2,iface=MIXER,name='TASDEVICE Profile id' 1
aplay --device="hw:0,0" silence.wav &
sleep 15
./tasdevice_ftc
```

## Shell script with Tinyalsa

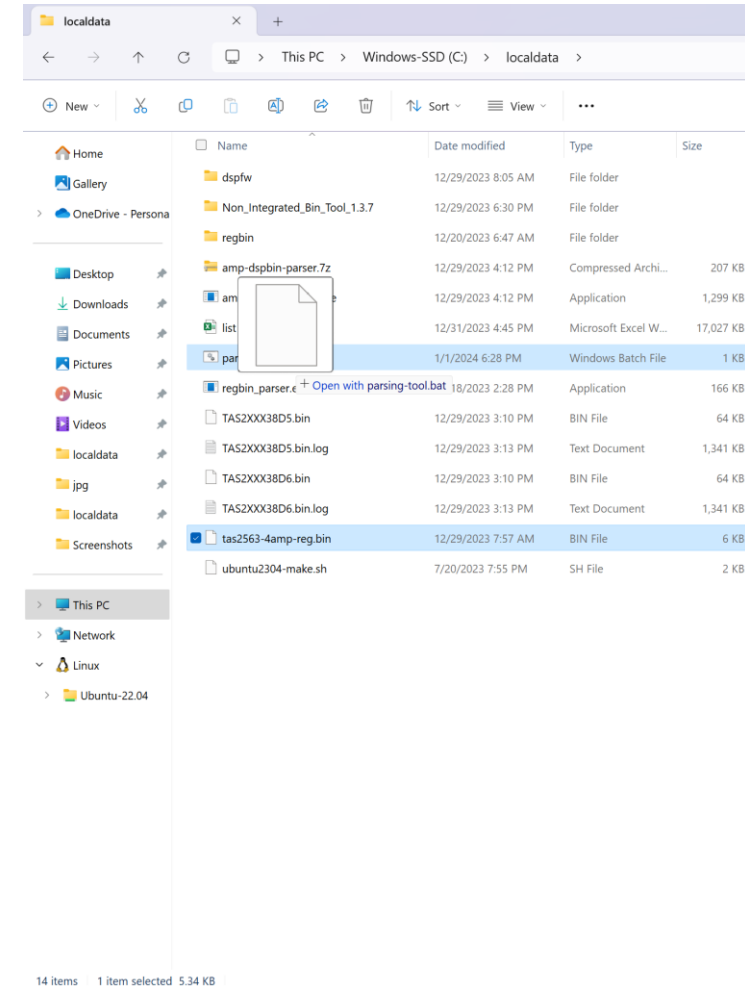
```
#!/bin/bash
tinymix "Program" 0
tinymix "Configuration" 1
tinymix "TASDEVICE Profile id" 1
tinyplay silence.wav &
sleep 15
./tasdevice_ftc
```

# Calibrated Data load in Android system

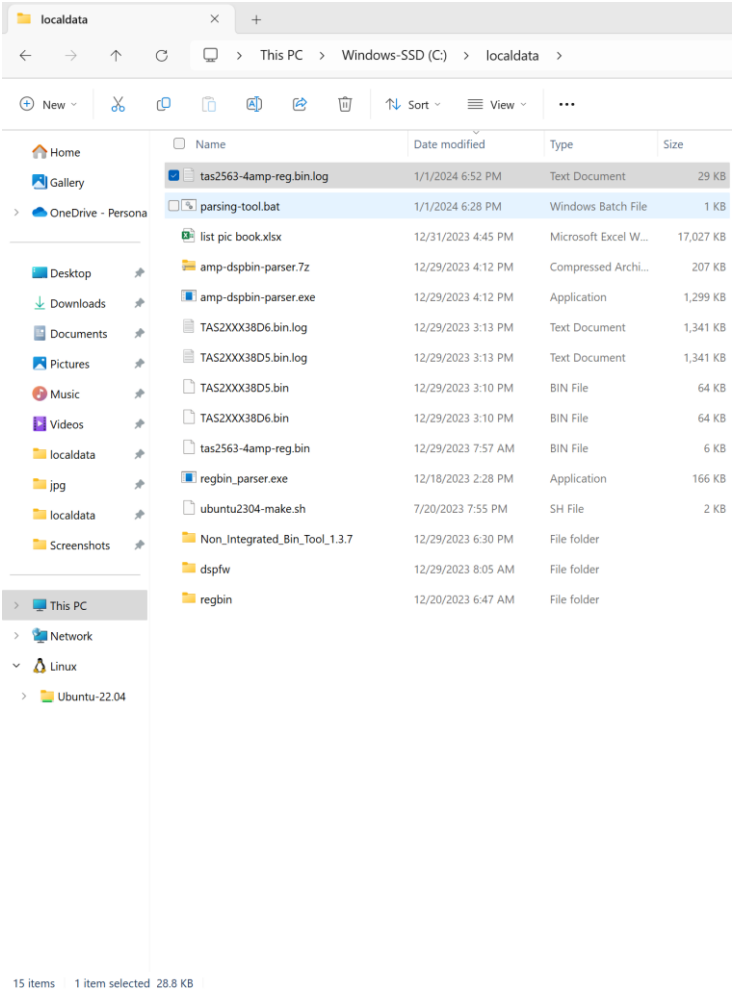
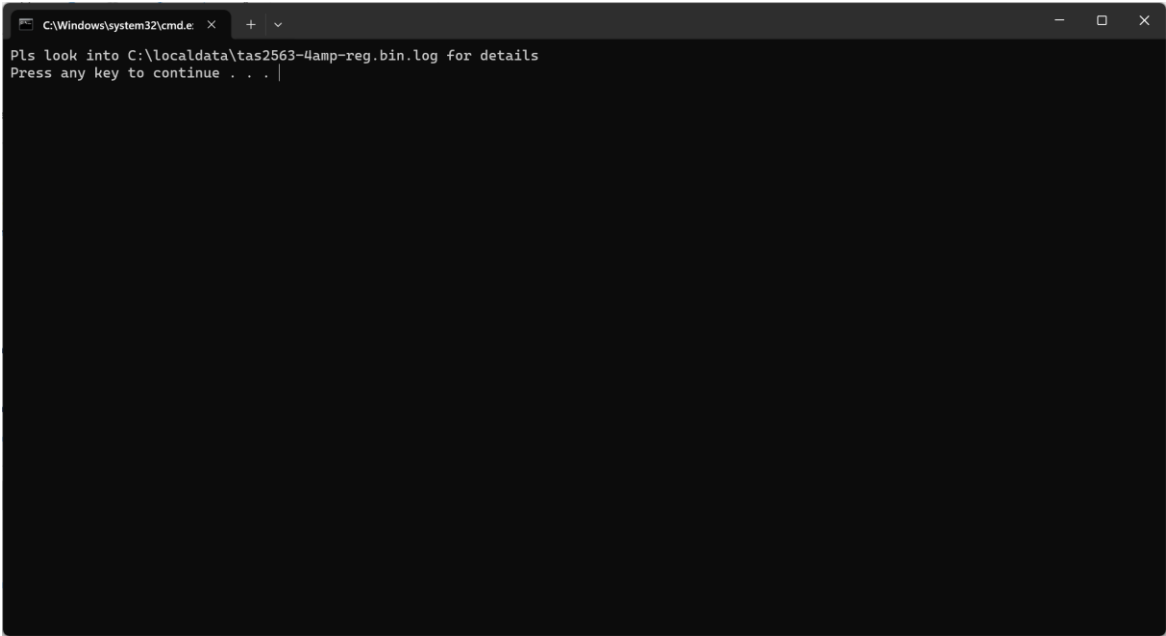
- In Android system, calibrated data was stored into /mnt/vendor/persist/audio/ or other persistent storage, yet some customers feedback that they can not get support on how to access file in that storage via request\_firmware immediately and want us offer a workaround solution for urgent use:
  - During audio\_hal init, pls read the calibrated data, e.g tas2781-0x48-cal.bin or tas2563-0x4e.bin, and write to the driver
    - The sample code for Audio\_hal is in the fct code, see example\_for\_load\_calibrated\_data
    - As to driver code, pls git pull the latest code from [tas2781-linux-drivers/tas2781-linux-driver - Unnamed repository; edit this file 'description' to name the repository.](#)

# amp-dspbin-parser.exe & amp-dspbin-parsing-tool.bat I

- Toolset for dspfw parsing.
- Operations:
  1. Confirm amp-dspbin-parsing-tool.bat & amp-dspbin-parser.exe are in the same folder
  2. Drag the dspfw file onto the amp-dspbin-parsing-tool.bat, then release, see the right picture, tas2563-4amp-reg.bin is dragged onto the amp-dspbin-parsing-tool.bat.
  3. It will popup a command-line screen, pls check the output, the left picture next page shows a successfully parsing
  4. The log of parsing dspfw file will be generated, see the right pic next page, tas2563-4amp-reg.bin.log has already been generated.



# amp-dspbin-parser.exe & amp-dspbin-parsing-tool.bat II



# APPENDIX

# Appendix I | Download Kernel code and SDK for BBB from...

- Code
  - <https://github.com/beagleboard/linux/tree/5.10-rt>
  - <https://openbeagle.org/beagleboard/linux.git>
- SDK
  - [PROCESSOR-SDK-AM335X Software development kit \(SDK\) | TI.com](#)

# Appendix II | Porting Tinyalsa I

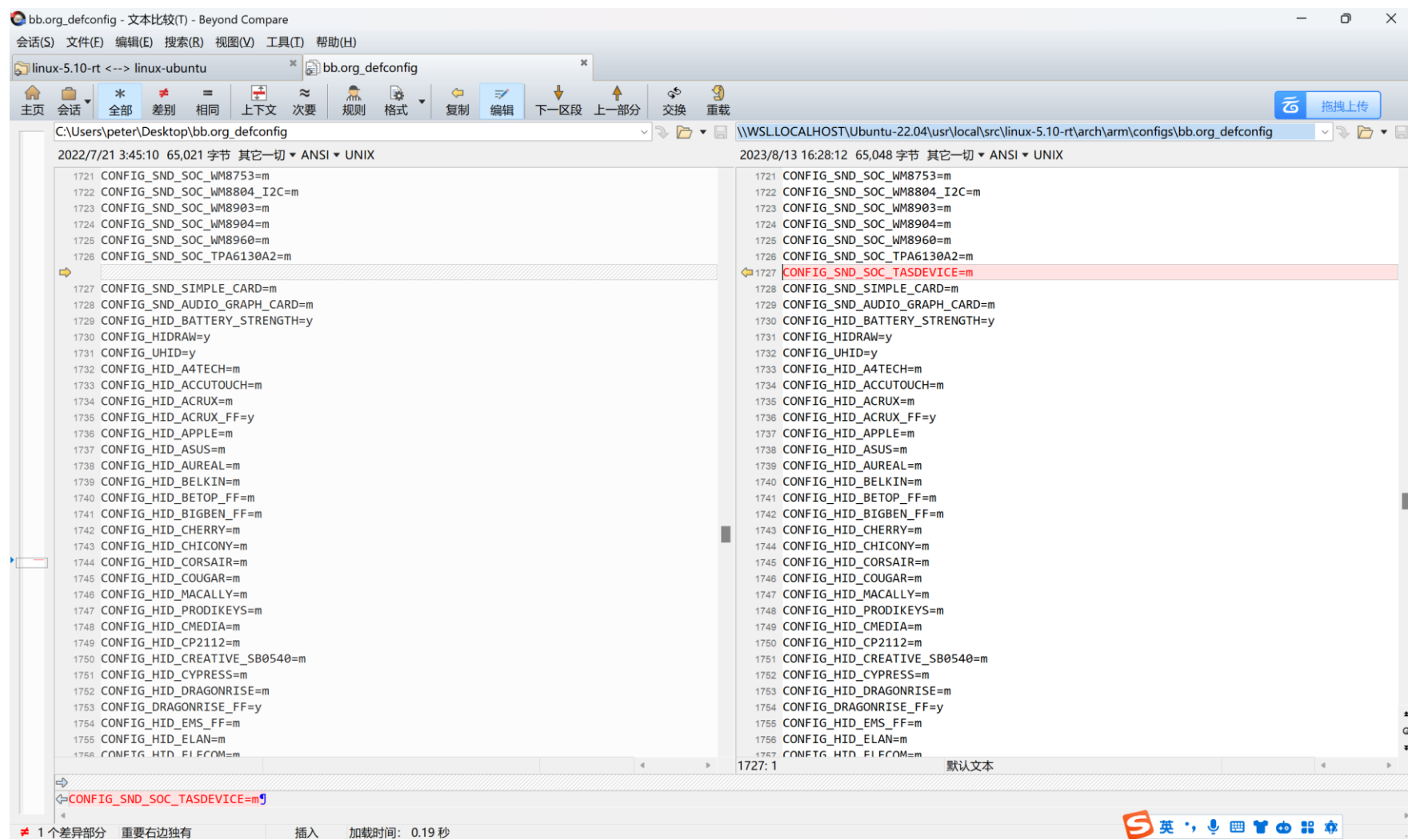
- Reference link:  
[Tinyalsa - Tiny library to interface with ALSA in the Linux kernel - \(tinyalsa\) \(opensourcelibs.com\)](#)  
Download link:  
[GitHub - tinyalsa/tinyalsa: Tiny library to interface with ALSA in the Linux kernel](#)
- Move driver code into BBB's local direction.



# Appendix II | Porting Tinyalsa II | Steps of installation

1. Modify cross compile, add below statements into top of Makefile.  
export ARCH = arm  
export CROSS\_COMPILE = arm-linux-gnueabi-  
export CC = \$(CROSS\_COMPILE)gcc
2. Assign permissions to scripts in tinyalsa package  
#chmod 777 script
3. To build and install with Make, run the commands:  
# sudo make  
# sudo make install  
# sudo ldconfig
4. Once installed, the man pages are available via:  
# man tinyplay  
# man tinycap  
# man tinymix  
# man tinypcm\_info

# Appendix III | Add new module to be compiled



bb.org\_defconfig - 文本比较(T) - Beyond Compare

会话(S) 文件(F) 编辑(E) 搜索(R) 视图(V) 工具(T) 帮助(H)

linux-5.10-rt <--> linux-ubuntu

主页 会话 全部 差别 相同 上下文 次要 规则 格式 复制 编辑 下一区段 上一部分 交换 重载

C:\Users\peter\Desktop\bb.org\_defconfig

2022/7/21 3:45:10 65,021 字节 其它一切 ▾ ANSI ▾ UNIX

```
1721 CONFIG_SND_SOC_WM8753=m
1722 CONFIG_SND_SOC_WM8804_I2C=m
1723 CONFIG_SND_SOC_WM8903=m
1724 CONFIG_SND_SOC_WM8904=m
1725 CONFIG_SND_SOC_WM8960=m
1726 CONFIG_SND_SOC_TPA6130A2=m
1727 CONFIG_SND_SIMPLE_CARD=m
1728 CONFIG_SND_AUDIO_GRAPH_CARD=m
1729 CONFIG_HID_BATTERY_STRENGTH=y
1730 CONFIG_HIDRAW=y
1731 CONFIG_UHID=y
1732 CONFIG_HID_A4TECH=m
1733 CONFIG_HID_ACCUTOUCH=m
1734 CONFIG_HID_ACRUX=m
1735 CONFIG_HID_ACRUX_FF=y
1736 CONFIG_HID_APPLE=m
1737 CONFIG_HID_ASUS=m
1738 CONFIG_HID_AUREAL=m
1739 CONFIG_HID_BELKIN=m
1740 CONFIG_HID_BETOP_FF=m
1741 CONFIG_HID_BIGBEN_FF=m
1742 CONFIG_HID_CHERRY=m
1743 CONFIG_HID_CHICONY=m
1744 CONFIG_HID_CORSAIR=m
1745 CONFIG_HID_COUGAR=m
1746 CONFIG_HID_MACALLY=m
1747 CONFIG_HID_PRODIKEYS=m
1748 CONFIG_HID_CMEDIA=m
1749 CONFIG_HID_CP2112=m
1750 CONFIG_HID_CREATIVE_SB0540=m
1751 CONFIG_HID_CYPRESS=m
1752 CONFIG_HID_DRAGONRISE=m
1753 CONFIG_DRAGONRISE_FF=y
1754 CONFIG_HID_EMS_FF=m
1755 CONFIG_HID_ELAN=m
1756 CONFIG_HID_ELECOM=m
```

\\WSLLLOCALHOST\Ubuntu-22.04\usr\local\src\linux-5.10-rt\arch\arm\configs\bb.org\_defconfig

2023/8/13 16:28:12 65,048 字节 其它一切 ▾ ANSI ▾ UNIX

```
1721 CONFIG_SND_SOC_WM8753=m
1722 CONFIG_SND_SOC_WM8804_I2C=m
1723 CONFIG_SND_SOC_WM8903=m
1724 CONFIG_SND_SOC_WM8904=m
1725 CONFIG_SND_SOC_WM8960=m
1726 CONFIG_SND_SOC_TPA6130A2=m
1727 CONFIG_SND_SOC_TASDEVICE=m
1728 CONFIG_SND_SIMPLE_CARD=m
1729 CONFIG_SND_AUDIO_GRAPH_CARD=m
1730 CONFIG_HID_BATTERY_STRENGTH=y
1731 CONFIG_HIDRAW=y
1732 CONFIG_UHID=y
1733 CONFIG_HID_A4TECH=m
1734 CONFIG_HID_ACCUTOUCH=m
1735 CONFIG_HID_ACRUX=m
1736 CONFIG_HID_ACRUX_FF=y
1737 CONFIG_HID_APPLE=m
1738 CONFIG_HID_ASUS=m
1739 CONFIG_HID_AUREAL=m
1740 CONFIG_HID_BELKIN=m
1741 CONFIG_HID_BETOP_FF=m
1742 CONFIG_HID_BIGBEN_FF=m
1743 CONFIG_HID_CHERRY=m
1744 CONFIG_HID_CHICONY=m
1745 CONFIG_HID_CORSAIR=m
1746 CONFIG_HID_COUGAR=m
1747 CONFIG_HID_MACALLY=m
1748 CONFIG_HID_PRODIKEYS=m
1749 CONFIG_HID_CMEDIA=m
1750 CONFIG_HID_CP2112=m
1751 CONFIG_HID_CREATIVE_SB0540=m
1752 CONFIG_HID_CYPRESS=m
1753 CONFIG_HID_DRAGONRISE=m
1754 CONFIG_DRAGONRISE_FF=y
1755 CONFIG_HID_EMS_FF=m
1756 CONFIG_HID_ELAN=m
1757 CONFIG_HID_ELECOM=m
```

1727: 1 默认文本

1 个差异部分 重要右边独有 插入 加载时间: 0.19 秒

# Appendix IV | DTS in QCOM Platform and Sound card register I

We provide reference DTS for Beagle Bone Black. For i2c setting for pcmdevice, DTS on QCOM platform is same as DTS on BBB. Only difference is the sound card register. Here offer an example on how to register the sound card. You may as well consult Qualcomm on this part for detail.

- `SND_SOC_DAILINK_DEFS(quat_tdm_tx_0,  
DAILINK_COMP_ARRAY(COMP_CPU("msm-dai-q6tdm.36913")),  
DAILINK_COMP_ARRAY(COMP_CODEC("tasdevice-codec.x-00yy", "tasdevice-codec")),  
DAILINK_COMP_ARRAY(COMP_PLATFORM("msm-pcm-routing")));`
  - x is the i2c bus No. where tasdevice connected
  - yy is the 7-bit i2c address in HEX style, the letter ought to be lowercase

e.g:

```
SND_SOC_DAILINK_DEFS(pri_mi2s_rx,  
DAILINK_COMP_ARRAY(COMP_CPU("snd-soc-dummy-dai")),  
DAILINK_COMP_ARRAY(COMP_CODEC("tasdevice-codec.x-00yy", "tasdevice-codec")),  
DAILINK_COMP_ARRAY(COMP_PLATFORM("snd-soc-dummy")));  
  
SND_SOC_DAILINK_DEFS(pri_mi2s_tx,  
DAILINK_COMP_ARRAY(COMP_CPU("snd-soc-dummy-dai")),  
DAILINK_COMP_ARRAY(COMP_CODEC("tasdevice-codec.x-00yy", "tasdevice-codec")),  
DAILINK_COMP_ARRAY(COMP_PLATFORM("snd-soc-dummy")));
```

# Appendix IV | DTS in QCOM Platform and Sound card register II

```
gem_row_wifi:/ # cat proc/asound/pcm
00-00: CODEC_DMA-LPAIF_RXTX-RX-0 multicodec-0 : : playback 1
00-01: CODEC_DMA-LPAIF_RXTX-RX-1 multicodec-1 : : playback 1
00-02: CODEC_DMA-LPAIF_RXTX-RX-2 multicodec-2 : : playback 1
00-03: CODEC_DMA-LPAIF_RXTX-RX-3 multicodec-3 : : playback 1
00-04: CODEC_DMA-LPAIF_RXTX-RX-5 multicodec-4 : : playback 1
00-05: CODEC_DMA-LPAIF_RXTX-RX-6 rx_macro_rx6-5 : : playback 1
00-06: CODEC_DMA-LPAIF_RXTX-TX-3 multicodec-6 : : capture 1
00-07: CODEC_DMA-LPAIF_RXTX-TX-4 multicodec-7 : : capture 1
00-08: CODEC_DMA-LPAIF_VA-TX-0 va_macro_tx1-8 : : capture 1
00-09: CODEC_DMA-LPAIF_VA-TX-1 va_macro_tx2-9 : : capture 1
00-10: CODEC_DMA-LPAIF_VA-TX-2 va_macro_tx3-10 : : capture 1
00-11: PCM_RT_PROXY-TX-1 snd-soc-dummy-dai-11 : : capture 1
00-12: PCM_RT_PROXY-RX-1 snd-soc-dummy-dai-12 : : playback 1
00-13: USB_AUDIO-RX snd-soc-dummy-dai-13 : : playback 1
00-14: USB_AUDIO-TX snd-soc-dummy-dai-14 : : capture 1
00-15: MI2S-LPAIF-RX-PRIMARY tasdevice-codec.3-004c-15 : : playback 1
00-16: MI2S-LPAIF-TX-PRIMARY tasdevice-codec.3-004c-16 : : capture 1
00-17: MI2S-LPAIF_AUD-RX-PRIMARY msm-stub-rx-17 : : playback 1
00-18: MI2S-LPAIF_AUD-TX-PRIMARY msm-stub-tx-18 : : capture 1
00-19: MI2S-LPAIF-RX-TERTIARY msm-stub-rx-19 : : playback 1
00-20: MI2S-LPAIF-TX-TERTIARY msm-stub-tx-20 : : capture 1
00-21: MI2S-LPAIF_RXTX-RX-PRIMARY msm-stub-rx-21 : : playback 1
00-22: MI2S-LPAIF_RXTX-TX-PRIMARY msm-stub-tx-22 : : capture 1
00-23: MI2S-LPAIF_VA-RX-PRIMARY msm-stub-rx-23 : : playback 1
00-24: MI2S-LPAIF_VA-TX-PRIMARY msm-stub-tx-24 : : capture 1
00-25: MI2S-LPAIF_WSA-RX-PRIMARY msm-stub-rx-25 : : playback 1
00-26: MI2S-LPAIF_WSA-TX-PRIMARY msm-stub-tx-26 : : capture 1
00-27: DISPLAY_PORT-RX msm_dp_audio_codec_rx_dai-27 : : playback 1
00-28: SLIM-DEV1-RX-7 btfrm_bt_sco_a2dp_slim_rx-28 : : playback 1
00-29: SLIM-DEV1-TX-7 btfrm_bt_sco_slim_tx-29 : : capture 1
gem_row_wifi:/ #
```

# Appendix V | Compile firmware into Android system

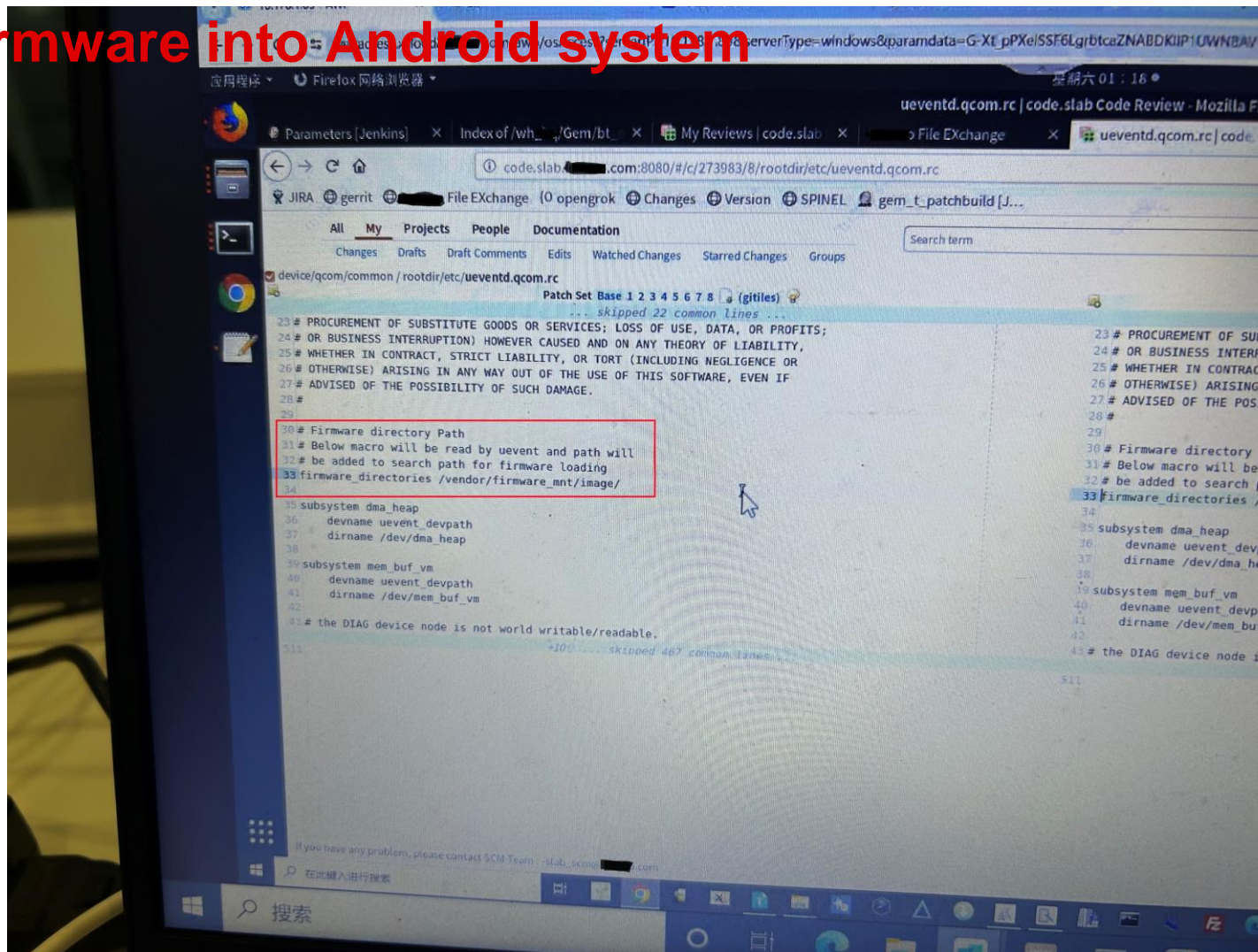
- Add following into mk file

```
#Add backend_conf.xml for agmtool
PRODUCT_COPY_FILES += \
    vendor/qcom/opensource/audio-hal/primary-hal/configs/taro/audio_param/backend_conf.xml:${TARGET_COPY_OUT_VENDOR}/etc/backend_conf.xml \
    vendor/qcom/opensource/audio-hal/primary-hal/configs/taro/audio_param/tas2563-4amp-reg.bin:${TARGET_COPY_OUT_VENDOR}/firmware/tas2563-4amp-reg.bin \
    vendor/qcom/opensource/audio-hal/primary-hal/configs/taro/audio_param/tas2563-4amp-dsp.bin:${TARGET_COPY_OUT_VENDOR}/firmware/tas2563-4amp-dsp.bin
# Audio configuration xml's related to Waipio
```



## Appendix VI | Set path for firmware into Android system

- Kindly see the red rectangular in the right picture and into which setting file to be saved



## Appendix VII | Compile the tas2563 module for Raspberry PI 4

1. Integrated the tas2563 code mentioned in previous pages.
2. Generate the config file for compiling kernel and tas2563 module. Pls put the config file in the compiling base/config\_dir



tas2563-miele.config

3. Compiling the kernel with compiling script. Pls put the script in the compiling base first.



miele-build.sh

## Appendix VII | Compile the tas2563 module for Raspberry PI 4

1. Integrated the tas2563 code mentioned in previous pages.
2. Copy integrated-tasdevice-soundcard-overlay.dts into base/arch/arm64/boot/dts/overlays and add integrated-tasdevice-soundcard-overlay.dts into Makefile in that folder for compiling into dtbo file.



integrated-tasdevice-soundcard-overlay.dts

3. Generate the config file for compiling kernel and tas2563 module. Pls put the config file in the compiling base/config\_dir



tas2563-miele.config

4. Compiling the kernel with compiling script. Pls put the script in the compiling base first.



miele-build.sh



## Appendix VIII | flash Raspberry PI 4 with new kernel image

1. Insert SD card into the host system
2. Execute the update script.



miele-update.sh

## Appendix IX | load tas2563 module

1. After Raspberry PI 4 normally run into the system, then put the .ko file, corresponding dtbo file and scripts in the same folder.
2. Execute the loading-tas2563 script.



insmod-tas2563.sh

## Appendix X | command for share the folder between windows and ubuntu with VirtualBox

```
# mount -t vboxsf sharedir sharedir
```

***THANKS!***