Guideline for TASDEVICE driver on Linux

BeagleBone Black/QCOM + tasdevice

Revision history

Ver	Date	Author	Description
1.0	2022/5/30		1. Set i2c clk to 400kHz 2. Add audio RX/TX setting illustration
1.1	2023/6/13		1. Support Kernel 6.01 2. Update Interrupt-free dts setting 3. Add json files
1.2	2023/09/01		Add Sound card register on QCOM
1.3	2023/10/4		Update the interrupt setting in DTS
1.4	2025/1/14		Add tas58xx

Information

tasdevice Driver	pcm9211, tas2110, tas256x, tas27xx, tas58xx
7-bit I^2C Address	0x38 ~ 0x3f, 0x4a~0x4d
Platform	Qualcomm/Mtk/LSI/Hisilicon/BeagleBone Black/AMBA/i.MX
Architecture	ALSA/Tiny-ALSA
Bus type	I2C
Kernel Version	v5.10/v6.6.0-rc2

Release Package I

Driver Package	Source code	.c/.h/makefile/KCONFIG files
	Regbin file	NA, can be generated by json file with the regbin tool
	Jason file stored the detail information of regbin	tas2110-1amp-reg.json tas2562-3amp-reg.json tas257x-1amp-reg.json tas2764-1amp-reg.json tas2770-1amp-reg.json tas2780-1amp-reg.json tas2780-3amp-reg.json
	BBB-compile-tool/	tasdevice-compile-bbb.sh: Compiling script for compiling Linux kernel 5.1.0 for BBB and tasdevice driver
		tasdevice-compile-bbb-ko.sh: Script for compiling Linux kernel 5.10 based on BBB
		Kconfig: Example of Kconfig in codec folder including tasdevice setting
		Makefile: Example of Makefile in codec folder including tasdevice setting
	config_dir/	config-6.6.0-rc2: Example of config file for compiling Linux kernel 6.6.0-rc2 for Ubuntu 23.04
		.tasdevice-bbb-v5.10-config: Example of config file for compiling Linux kernel 5.10 for BBB

Release Package II

Driver Package	ubuntu2304x64-compile-tool/	ubuntu2304-make.sh: Compiling script for compiling Linux kernel 6.6.0-rc2 for ubuntu 23.04 and tasdevice driver
		Kconfig: Example of Kconfig in codec folder including tasdevice setting
		Makefile: Example of Makefile in codec folder including tasdevice setting
	ti,tasdevice.yaml	Example of dts setting

Kernel Driver compiling I

- Extract driver package(mainly the src folder) to the folder “sound/soc/codecs”, and rename it as tasdevice:
- Add the tasdevice compiling info into the Kconfig and Makefile in the sound/soc/codecs. For detail, kindly go to next page and the highlighted item.

File in sound/soc/codecs	Compiling info
Kconfig	source "sound/soc/codecs/tasdevice/Kconfig"
Makefile	obj-$(CONFIG_SND_SOC_TASDEVICE) += tasdevice/

Kernel Driver compiling II

Kconfig in sound/soc/codecs

Kconfig

```
1809        depends on I2C
1810
1811    config SND_SOC_TPA6130A2
1812        tristate "Texas Instruments TPA6130A2 headphone amplifier"
1813        depends on I2C
1814
1815    config SND_SOC_LPASS_WSA_MACRO
1816        depends on COMMON_CLK
1817        tristate "Qualcomm WSA Macro in LPASS(Low Power Audio SubSystem)"
1818
1819    config SND_SOC_LPASS_VA_MACRO
1820        depends on COMMON_CLK
1821        tristate "Qualcomm VA Macro in LPASS(Low Power Audio SubSystem)"
1822
1823    comment "tas2563-mono"
1824    source "sound/soc/codecs/tas2563-android-driver-master/Kconfig"
1825
1826    comment "tas2563-stereo"
1827    source "sound/soc/codecs/tas2563-android-driver-stereo/Kconfig"
1828
1829    comment "tas-device"
1830    source "sound/soc/codecs/tasdevice/Kconfig"
1831    endmenu
1832
```

Makefile in sound/soc/codecs

Makefile

```
611   obj-$(CONFIG_SND_SOC_WM8998)     += snd-soc-wm8998.o
612   obj-$(CONFIG_SND_SOC_WM9081)     += snd-soc-wm9081.o
613   obj-$(CONFIG_SND_SOC_WM9090)     += snd-soc-wm9090.o
614   obj-$(CONFIG_SND_SOC_WM9705)     += snd-soc-wm9705.o
615   obj-$(CONFIG_SND_SOC_WM9712)     += snd-soc-wm9712.o
616   obj-$(CONFIG_SND_SOC_WM9713)     += snd-soc-wm9713.o
617   obj-$(CONFIG_SND_SOC_WM_ADSP)    += snd-soc-wm-adsp.o
618   obj-$(CONFIG_SND_SOC_WM_HUBS)    += snd-soc-wm-hubs.o
619   obj-$(CONFIG_SND_SOC_WSA881X)    += snd-soc-wsa881x.o
620   obj-$(CONFIG_SND_SOC_ZL38060)    += snd-soc-zl38060.o
621   obj-$(CONFIG_SND_SOC_ZX_AUD96P22) += snd-soc-zx-aud96p22.o
622
623   # Amp
624   obj-$(CONFIG_SND_SOC_MAX9877)    += snd-soc-max9877.o
625   obj-$(CONFIG_SND_SOC_MAX98504)   += snd-soc-max98504.o
626   obj-$(CONFIG_SND_SOC_SIMPLE_AMPLIFIER)  += snd-soc-simple-amplifier.o
627   obj-$(CONFIG_SND_SOC_TPA6130A2) += snd-soc-tpa6130a2.o
628   obj-$(CONFIG_SND_SOC_LPASS_WSA_MACRO)    += snd-soc-lpass-wsa-macro.o
629   obj-$(CONFIG_SND_SOC_LPASS_VA_MACRO)     += snd-soc-lpass-va-macro.o
630
631   # Mux
632   obj-$(CONFIG_SND_SOC_SIMPLE_MUX)     += snd-soc-simple-mux.o
633   #obj-$(CONFIG_SND_SOC_TAS2563) += tas2563-android-driver-master/
634   obj-$(CONFIG_SND_SOC_TAS2563) += tas2563-android-driver-stereo/
635   obj-$(CONFIG_SND_SOC_TASDEVICE) += tasdevice/
```

Kernel Driver compile III | defconfig Setting

- Confirm whether following three items are set to “y”, if not, set them first, before next step
  - CONFIG_OF=y
  - CONFIG_SOUND=y
  - CONFIG_SND=y
  - CONFIG_SND_SOC=y
- Add following item into defconfig, highlighted part in right picture for detail
  - CONFIG_SND_SOC_TASDEVICE=y

config-wsl

```
2579  # CONFIG_SND_SOC_NAU8822 is not set
2580  # CONFIG_SND_SOC_NAU8824 is not set
2581  # CONFIG_SND_SOC_TPA6130A2 is not set
2582  # CONFIG_SND_SOC_LPASS_WSA_MACRO is not set
2583  # CONFIG_SND_SOC_LPASS_VA_MACRO is not set
2584
2585  #
2586  # tas2563-mono
2587  #
2588  CONFIG_SND_SOC_TAS2563=y
2589  CONFIG_TAS2563_REGMAP=y
2590  CONFIG_TAS2563_CODEC=y
2591  CONFIG_TAS2563_MISC=y
2592
2593  #
2594  # tas2563-stereo
2595  #
2596
2597  #
2598  # tas-device
2599  #
2600  CONFIG_SND_SOC_TASDEVICE=y
2601  # end of CODEC drivers
2602
2603  CONFIG_SND_SIMPLE_CARD_UTILS=y
2604  CONFIG_SND_SIMPLE_CARD=y
2605  CONFIG_SND_X86=y
2606  CONFIG_AC97_BUS=y
2607
2608  #
2609  # HID support
2610  #
2611  # CONFIG_HID is not set
2612
2613  #
2614  # I2C HID support
2615  #
2616  # CONFIG_I2C_HID is not set
2617  # end of I2C HID support
2618
2619  #
2620  # Intel ISH HID support
2621  #
2622  # CONFIG_INTEL_ISH_HID is not set
2623  # end of Intel ISH HID support
2624  # end of HID support
```

Kernel Driver compile IV | Add bin file into zImage I

- Confirm where the firmware file store in the target device
  - The path storing the bin file is defined in kernel/drivers/base/firmware_class.c, normally /system/vendor/firmware, /system/etc/firmware, or /lib/firmware

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

  - Compile the bin file into image.
  - During debug, pushing the bin file into proper folder of the target device is more convenient than compiling into image.

Kernel Driver compile IV | Add bin file into zImage II

– 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="**tas2770-2amp-reg.bin** regulatory.db regulatory.db.p7s“
  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 IV | Compile firmware into Android system

Kernel Driver compile V | DTS config I | 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)
    >;
};
```

```
&i2c2 {
  pinctrl-names = "default";
  #address-cells = <1>;
  #size-cells = <0>;
  status = "okay";
  clock-frequency = <400000>;
  pinctrl-0 = <&i2c2_pins>;
  tasdevice: tasdevice@2d {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas5825";
    reg = <0x2d>;
    reset-gpios = <&gpio1 10 GPIO_ACTIVE_HIGH>;
  };
};
```

Both 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

Kernel Driver compile VI | DTS config II | 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)
    >;
};
```

```
&i2c2 {
  pinctrl-names = "default";
  #address-cells = <1>;
  #size-cells = <0>;
  status = "okay";
  clock-frequency = <400000>;
  pinctrl-0 = <&i2c2_pins>;
  tasdevice: tasdevice@2c {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas5805";
    reg = <0x2c>, <0x2e>, <0x2d>;
    reset-gpios = <&gpio1 10 GPIO_ACTIVE_HIGH>;
  };
};
```

- 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

Kernel Driver compile VII | DTS config III | Interrupt-free driver

- Not setting interrupt information in dts will bypass all the interrupt handling in the code

  PS: 38 in the following pic is the 7-bit i2c address, HEX style.

```
i2c {
  /* example for three devices with interrupt support */
  #address-cells = <1>;
  #size-cells = <0>;
  three: tasdevice@38 {
    compatible = "ti,tas2562";
    reg = <0x38>, /* primary-device */
          <0x3c>, /* secondary-device */
          <0x3e>; /* tertiary-device */
    #sound-dai-cells = <0>;
    reset-gpios = < &gpio1 10 GPIO_ACTIVE_HIGH >;
  };
};
```

Kernel Driver compile IX | DTS config IV | Interruptible driver I

- The setting for interrupt driver is different from the one for interrupt-free driver.

```
100 &i2c2 {
101   #address-cells=<1>;
102   #size-cells=<0>;
103   tas2780: tas2780@38 {
104       compatible = "ti,tas2780";
105       #sound-dai-cells = <0>;
106       reg = <0x38>;
107       reset-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;
108       interrupt-parent = <&gpio1>;
109       interrupts = <15>;
110   };
111 };
```

- Do not forget set the irq-gpio to input mode.

```
42          mcasp0_pins: mcasp0_pins {
43                  pinctrl-single,pins = <
44                          AM33XX_IOPAD(0x9a8, PIN_INPUT_PULLUP | MUX_MODE0) /* mcasp0_ahcklx.mcasp0_axr1*/
45                          AM33XX_IOPAD(0x998, PIN_INPUT_PULLDOWN | MUX_MODE0) /* mcasp0_ahclkr.mcasp0_axr0*/
46                          AM33XX_IOPAD(0x9ac, PIN_INPUT_PULLUP | MUX_MODE0) /* mcasp0_ahcklx.mcasp0_ahclkx */
47                          AM33XX_IOPAD(0x99c, PIN_OUTPUT_PULLDOWN | MUX_MODE2) /* mcasp0_ahclkr.mcasp0_axr2*/
48                          AM33XX_IOPAD(0x994, PIN_OUTPUT_PULLUP | MUX_MODE0) /* mcasp0_fsx.mcasp0_fsx */
49                          AM33XX_IOPAD(0x990, PIN_OUTPUT_PULLDOWN | MUX_MODE0) /* mcasp0_aclkx.mcasp0_aclkx */
50                          AM33XX_IOPAD(0x844, PIN_OUTPUT_PULLDOWN | MUX_MODE7) /* gpio1[17] P9-23*/
51                          AM33XX_IOPAD(0x878, PIN_INPUT_PULLDOWN | MUX_MODE7) /* gpio1[28] P9-12*/
52                  >;
53          };
```

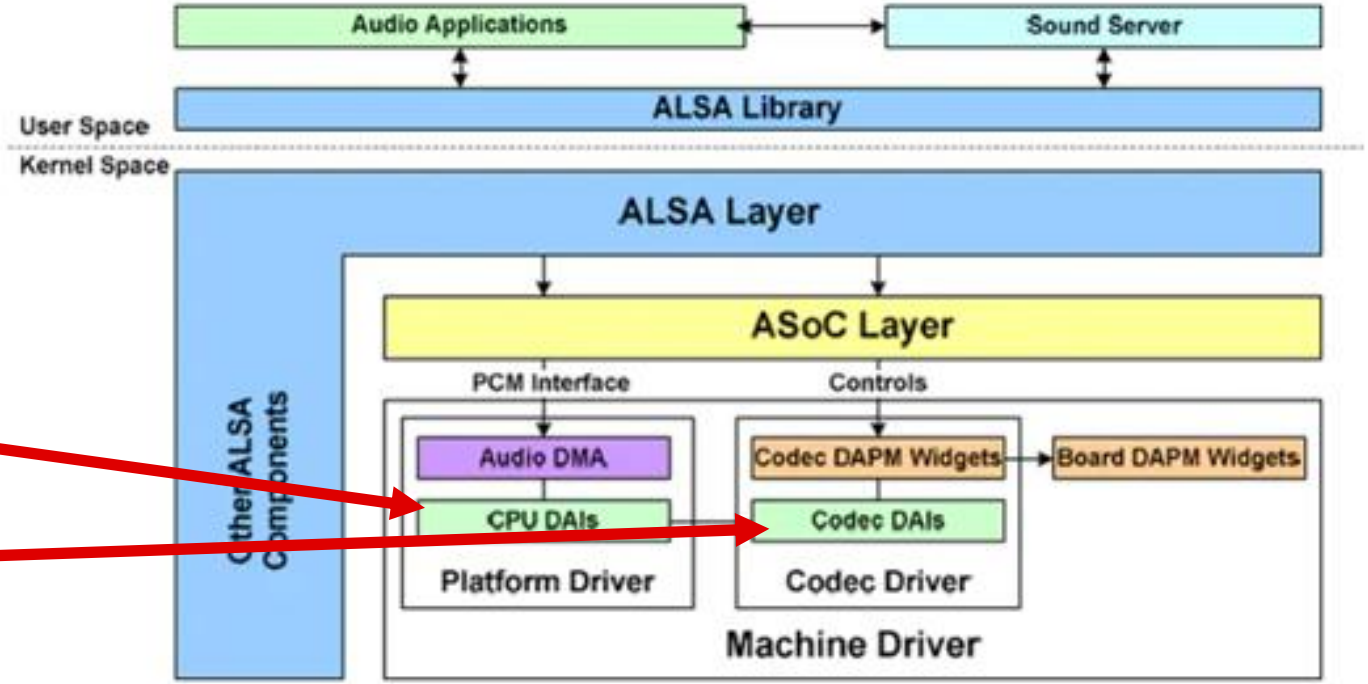
Kernel Driver compile X | DTS config V | Interruptible driver II

- Current code release support interruptible driver for pcm9211, tas2110, tas2560, tas2562, tas2564, tas2770, tas2780, tas58xx, etc

Kernel Driver compile XI | DTS config VI | flexible dsp-a

```
141  sound {
142    compatible = "simple-audio-card";
143    simple-audio-card,name = "TI BeagleBone Black";
144    simple-audio-card,dai-link@0 {
145      format = "dsp_a";
146      bitclock-master = <&sound0_1_master>;
147      frame-master = <&sound0_1_master>;
148
149      sound0_1_master: cpu {
150        sound-dai = <&mcasp0>;
151        clocks = <&clk_mcasp0>;
152      };
153
154      codec {
155        sound-dai = <&tas2780>;
156      };
157    };
158  };
```

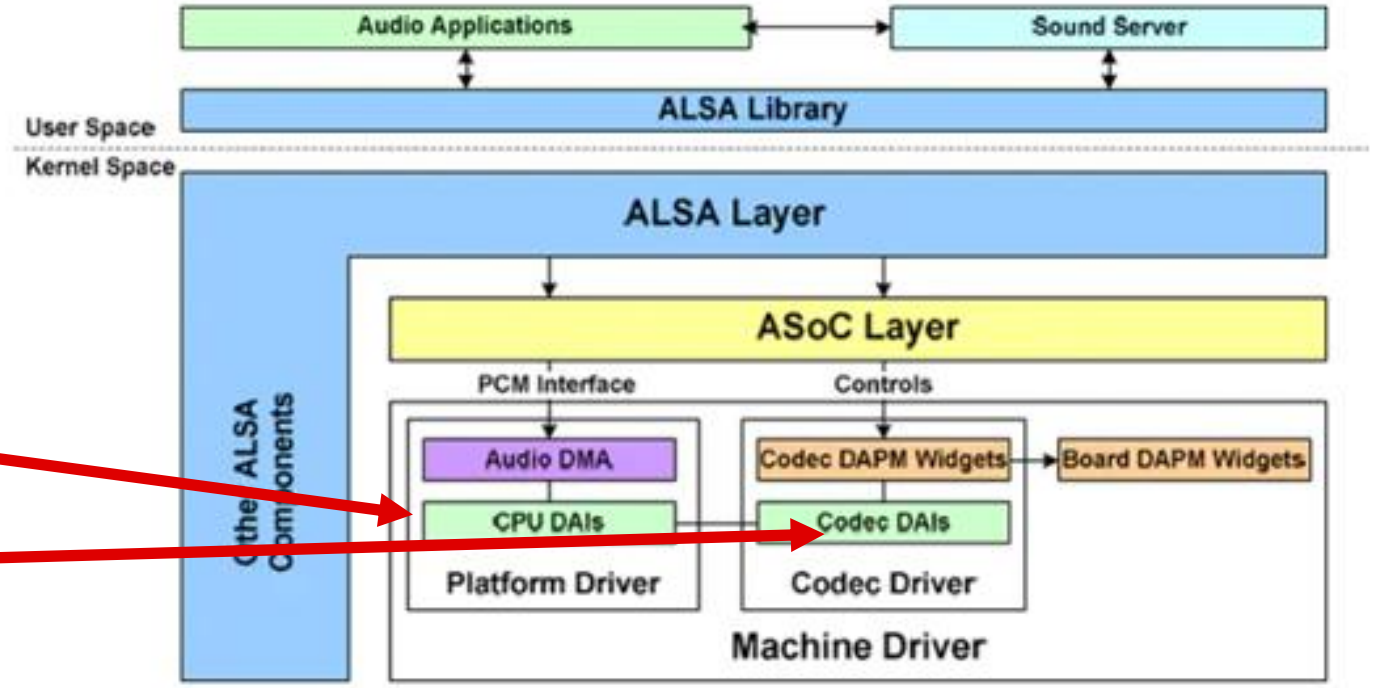
am335x-boneblack-hdmi.dtsi



Kernel Driver compile XII | DTS config VII | i2s

```
141     sound {
142         compatible = "simple-audio-card";
143         simple-audio-card,name = "TI BeagleBone Black";
144         simple-audio-card,dai-link@0 {
145             format = "i2s";
146             bitclock-master = <&sound0_1_master>;
147             frame-master = <&sound0_1_master>;
148
149             sound0_1_master: cpu {
150                 sound-dai = <&mcasp0>;
151                 clocks = <&clk_mcasp0>;
152             };
153
154             codec {
155                 sound-dai = <&tas2780>;
156             };
157         };
158     };
```

am335x-boneblack-hdmi.dtsi



How to Check Sound Card register

- ls /dev/snd
- ls /sys/bus/i2c/devices/2-0038
  - 2: the i2c bus number where chip has been connected
  - 0038: 7-bit i2c addr of the chip, this address is same as the one set in DTS
- cat /proc/asound/pcm
- cat /proc/asound/cards

```
root@am335x-evm:/proc/asound# ls
Black    card0    cards    devices  pcm      timers   version
root@am335x-evm:/proc/asound# cat pcm
00-00: davinci-mcasp.0-tasdevice-codec tasdevice-codec-0 : davinci-mcasp.0-tasdevice-codec tasdevice-codec-0 : playback 1 : capture 1
root@am335x-evm:/proc/asound#
root@am335x-evm:/proc/asound# cat cards
 0 [Black          ]: TI_BeagleBone_B - TI BeagleBone Black
                      TI BeagleBone Black
root@am335x-evm:/proc/asound# [  143.687157] NET: Registered protocol family 15
[  143.997184] Initializing XFRM netlink socket

root@am335x-evm:/proc/asound#
root@am335x-evm:/proc/asound# cd /sys/bus/i2c/devices/2-0038
root@am335x-evm:/sys/bus/i2c/devices/2-0038#
root@am335x-evm:/sys/bus/i2c/devices/2-0038# ls
act_addr         driver           modalias         of_node          reg              regcfg_list      subsystem
devinfo          fwload           name             power            regbininfo_list  regdump          uevent
```

Driver nodes I | Introduction

- In order to debug driver freely, several driver nodes have been defined
  - devinfo
  - reg
  - regdump
  - regbininfo_list
  - regcfg_list
  - fwload
  - i2caddr
- The location of driver nodes
  - /sys/bus/i2c/devices/2-0038 or /sys/class/i2c-adapter/i2c-2/2-0038
    - Last page has already introduced and captured the picture of the driver nodes

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
```

  How to check firmware load successfully

  1. amixer contents

```
root@am335x-evm:/sys/bus/i2c/devices/2-0038# amixer contents
numid=1,iface=MIXER,name='TASDEVICE Profile id'
  ; type=INTEGER,access=rw------,values=1,min=0,max=6,step=0
  : values=-1
```

  2. amixer controls

```
root@am335x-evm:/sys/bus/i2c/devices/2-0038# amixer controls
numid=1,iface=MIXER,name='TASDEVICE Profile id'
```

```
amixer controls
numid=3,iface=MIXER,name='TASDEVICE Profile id'
numid=1,iface=MIXER,name='tas2780-amp-gain-volume'
numid=2,iface=MIXER,name='tas2780-digital-volume'
```

- cat i2caddr

  Illustration: show the active i2c address

```
root@am335x-evm:/sys/bus/i2c/devices/2-0048# cat i2caddr
Active SmartPA-0x48
```

- cat devinfo

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

```
root@am335x-evm:/sys/bus/i2c/devices/2-0038# cat devinfo
No.     DevTyp  Addr
0       tas2780 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 0x2 0xE1 > reg

- cat reg

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

```
root@am335x-evm:/sys/bus/i2c/devices/2-0038# echo 0 0x00 0x00 0x02 0x81 >reg
root@am335x-evm:/sys/bus/i2c/devices/2-0038# [  501.409311] tasdevice-codec 2-0038: reg: count = 22
[  501.409394] tasdevice-codec 2-0038: [tasdevice]reg: chn=0, book=0x00 page=0x00 reg=0x02 val=0x81, cnt=22

root@am335x-evm:/sys/bus/i2c/devices/2-0038# cat reg
i2c-addr: 0x38
Chn0B0x00P0x00R0x02:0x82
```

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
  – eg: echo 1 0x00 0x00 > reg

- cat regdump

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

B0P0-regdump.t
xt

B0P1-regdump.t
xt

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

  PS: If wanted detailed info from specific audio case, kindly use the drive node **regcfg_list**

```
root@am335x-evm:/sys/bus/i2c/devices/2-0038# cat regbininfo_list
Regbin File Version: 0x0105
conf 00: 2-slot dsp-a 16bit+echo ref
conf 01: 2-slot dsp-a 32bit
conf 02: recorder
conf 03: i2s 16bit+IV
conf 04: 2-slot dsp-a + IV
```

Driver nodes V | regcfg_list

```
root@am335x-evm:/sys/bus/i2c/devices/2-0038# echo 00 > regcfg_list
root@am335x-evm:/sys/bus/i2c/devices/2-0038# [  117.928138] tasdevice-codec 2-0038: regcfg: count = 3
[  117.928214] tasdevice-codec 2-0038: [regcfg_list]cfg= 0, cnt=3

root@am335x-evm:/sys/bus/i2c/devices/2-0038#
root@am335x-evm:/sys/bus/i2c/devices/2-0038# cat regcfg_list
Conf 00: 2-slot dsp-a 16bit+echo ref
block type:PRE_POWER_UP  device idx = 0x00
                SINGLE BYTE:
                        PAGE0x00 REG0x5c VALUE = 0xd9
                FIELD:
                        PAGE0x00 REG0x0e MASK0x40 VALUE = 0x40
                FIELD:
                        PAGE0x00 REG0x0f MASK0x40 VALUE = 0x40
                SINGLE BYTE:
                        PAGE0x00 REG0x16 VALUE = 0x40
                        PAGE0x00 REG0x02 VALUE = 0x18
block type:PRE_SHUTDOWN  device idx = 0x00
                FIELD:
                        PAGE0x00 REG0x16 MASK0x40 VALUE = 0x00
                SINGLE BYTE:
                        PAGE0x00 REG0x02 VALUE = 0x02
```

- echo CG > regcfg_list
  - CG is conf NO, it should be 2-digital decimal
  - eg: echo 01 > regcfg_list
- cat regcfg_list
  - Illustration: dump the register setting of the audio case specified by echo command

Test Commands I

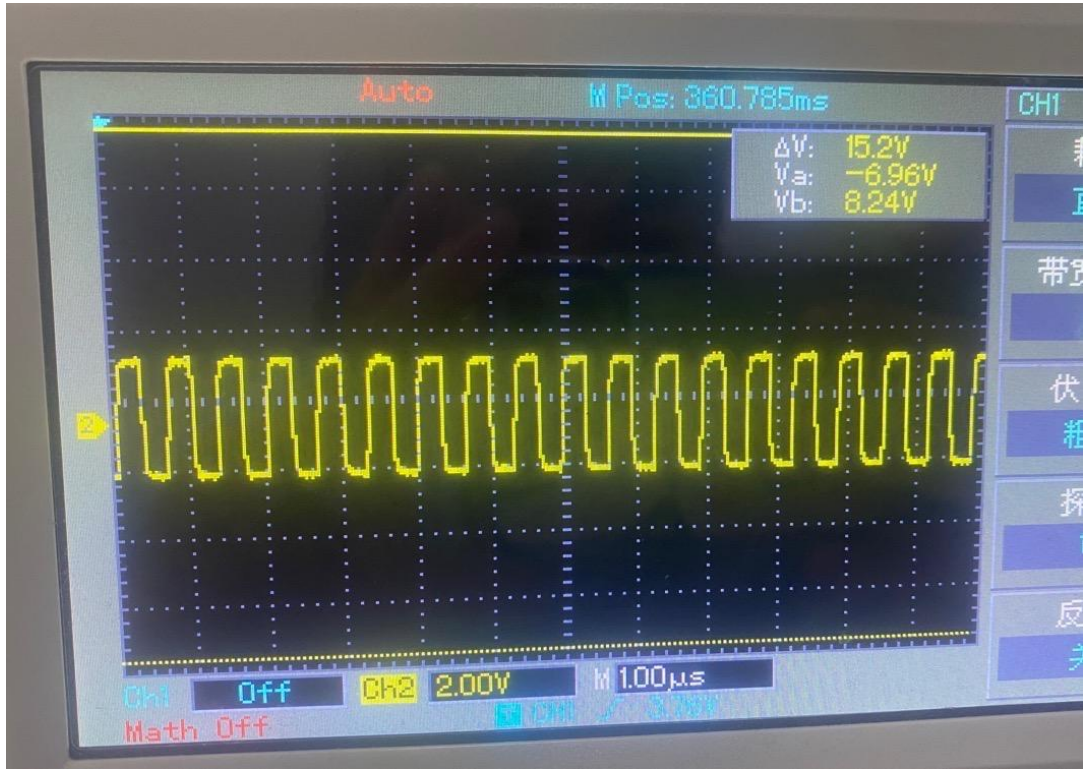
- If tasreg.bin file has not been compiled into zImage, **DO NOT** forget to run:
  - echo > /sys/bus/i2c/devices/2-0038/fwload
    - the 2 is the i2c bus number where tasdevice deployed
    - 0038 is the 7-bit i2c address for tasdevice
    - Above two items depends on the hardware connection, kindly consult your hardware engineer.
- DTS setting Flexible DSP-A
  - amixer cset numid=1,iface=MIXER,name='TASDEVICE Profile id' ***0***
  - aplay ***-c 4 -f S32_LE*** --device="hw:0,0" TDM-4slot-32b-test.wav &
  - Sleep 1
  - amixer cset numid=1,iface=MIXER,name='TASDEVICE Profile id' ***1***
  - arecord ***-c 4 -f S32_LE*** -r 48000 -d 15 --device="hw:0,0" echo_ref.wav

Test Commands II

Rgiht pic is the waveform of WS clk and bit clk for 4-slot TDM

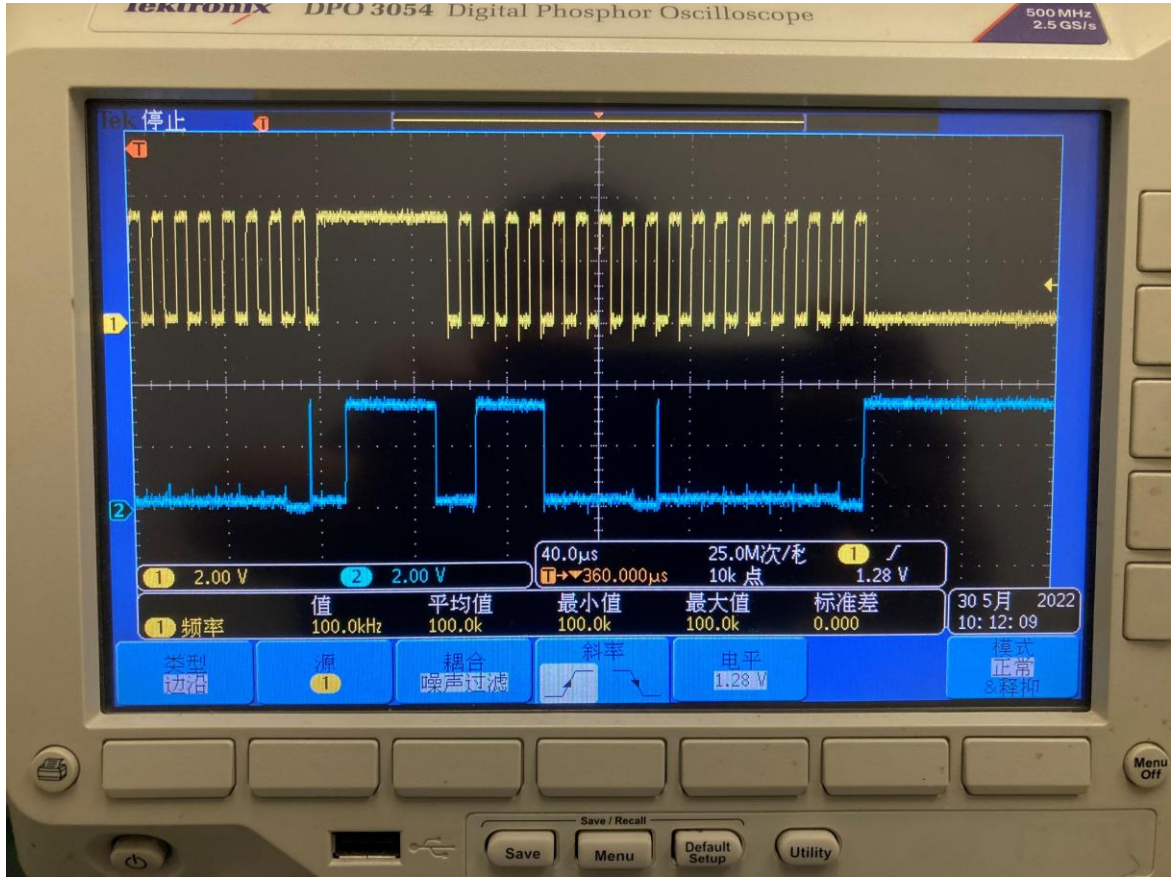
Test Commands III

- 16-bit I2S (Use DTS setting for *I2S*)
  - amixer cset numid=1,iface=MIXER,name='TASDEVICE Profile id' ***3 &***
  - aplay --device="hw:0,0" i2s-16b-test.wav
  - Sleep 1
  - amixer cset numid=1,iface=MIXER,name='TASDEVICE Profile id' ***4***
  - arecord -r 48000 -d 15 --device="hw:0,0" i2s-16b-echoref.wav



Appendix

Appendix I | How to set i2c clk to 400kHz I

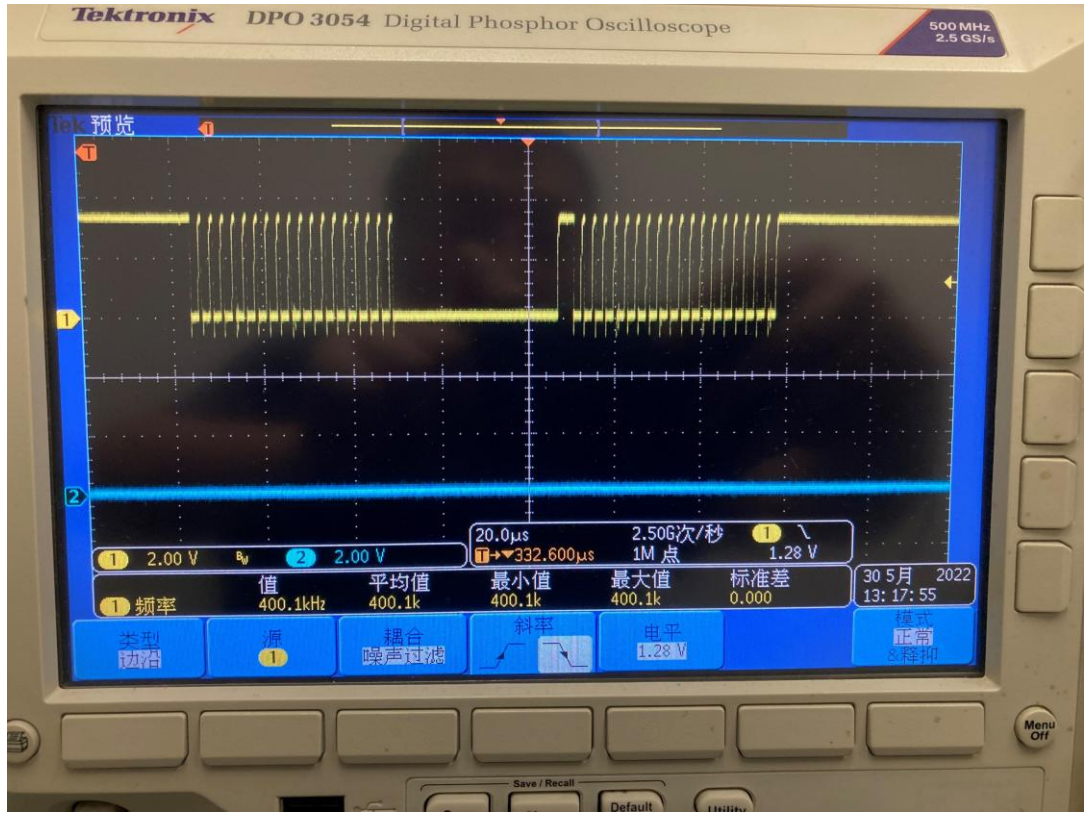


- In default, the i2c is set to 100kHz in BBB. See the pic in the left.

Appendix I | How to set i2c clk to 400kHz II

- Set the clk frequency in device tree. See the highlighted part in the figure on top right.

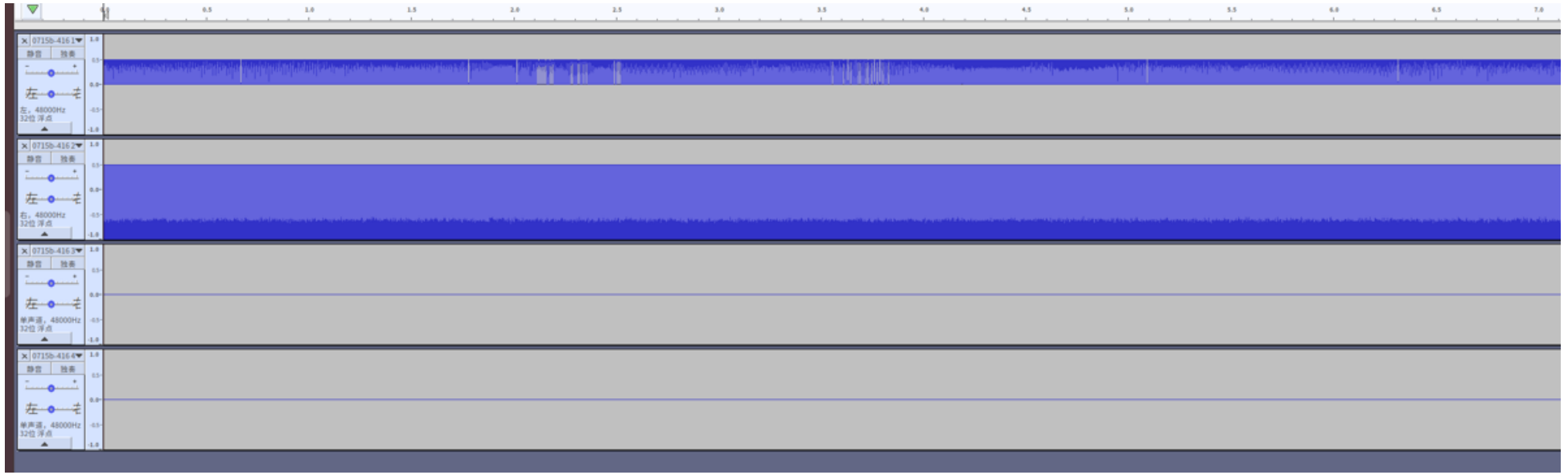
```
99  &i2c2 {
100     #address-cells = <1>;
101     #size-cells = <0>;
102     clock-frequency = <400000>;
103         pcmdevice: pcmdevice@48 {
104          compatible = "ti,pcmdevice";
105          #sound-dai-cells = <0>;
106          reg = <0x48>;
107     };
108 };
```



am335x-bonebl
hdmi-i2c-400kHz

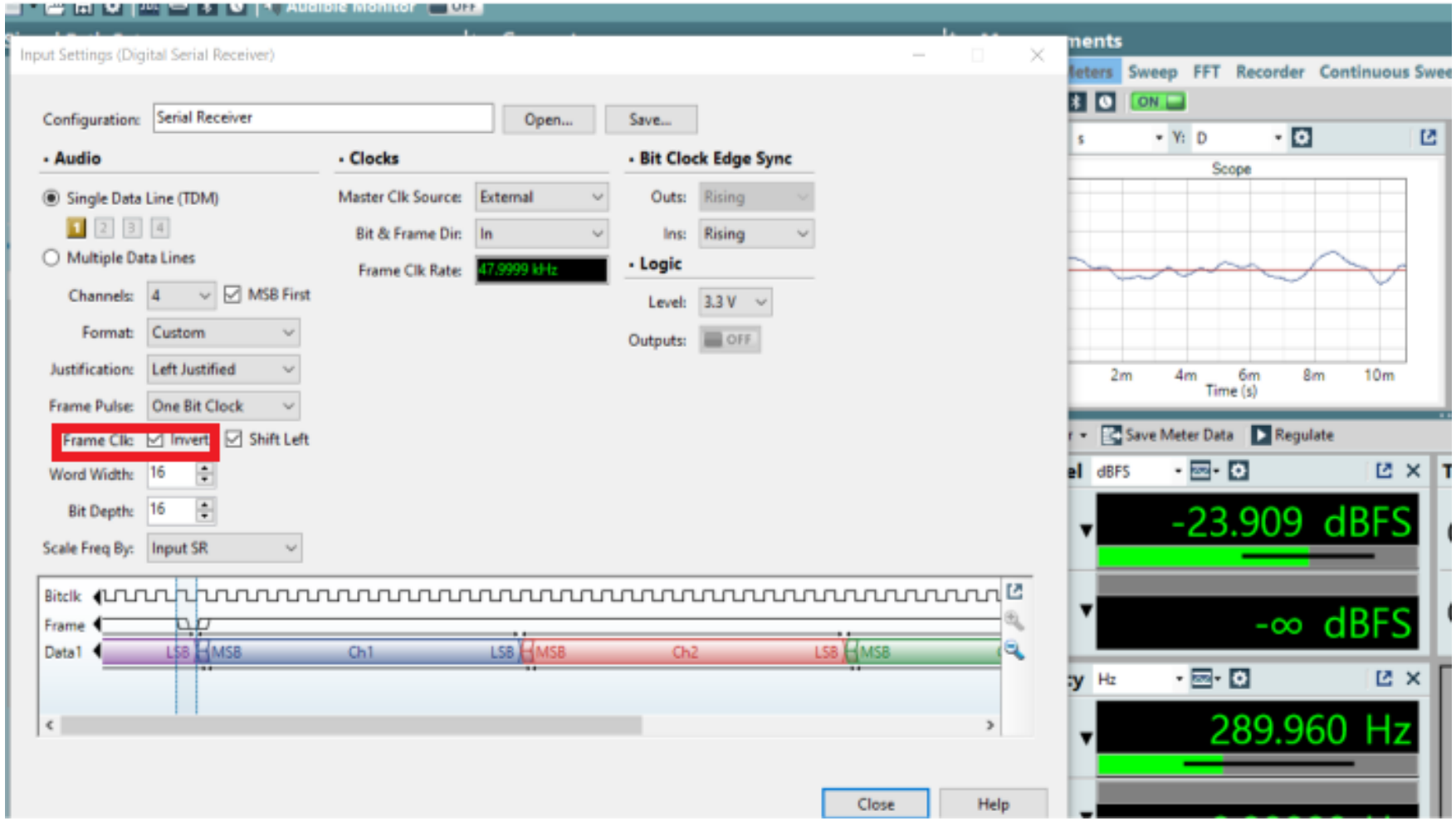
Appendix II | Abnormal echo ref from tas2780-EVM+BBB debug I

- Record the echo ref signal on BBB in 4-slot TDM, the signal is as following



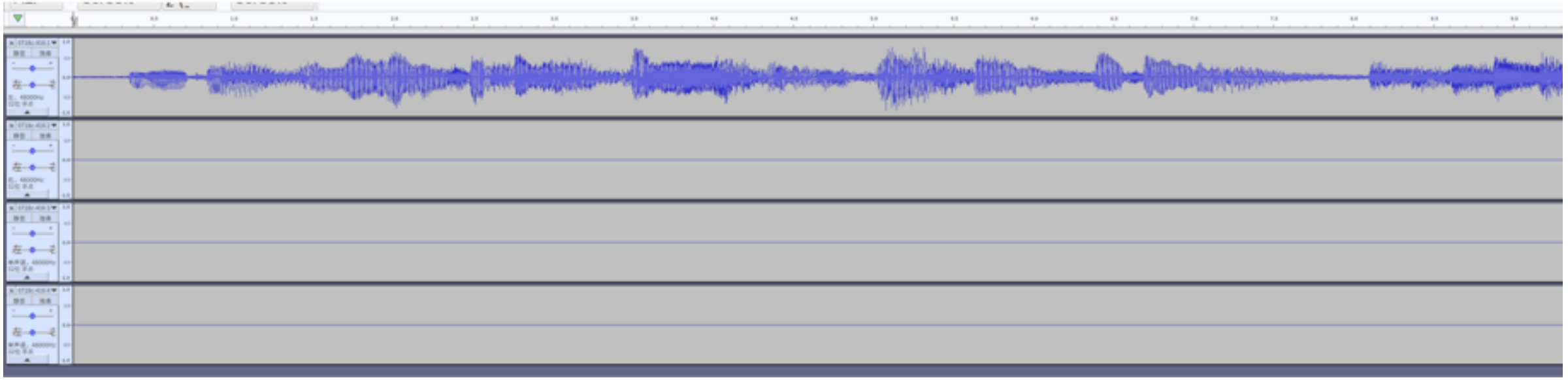
Appendix II | Abnormal echo ref from tas2780-EVM+BBB debug II

- Using AP to capture the DOUT from tas2780-EVM+BBB with following config



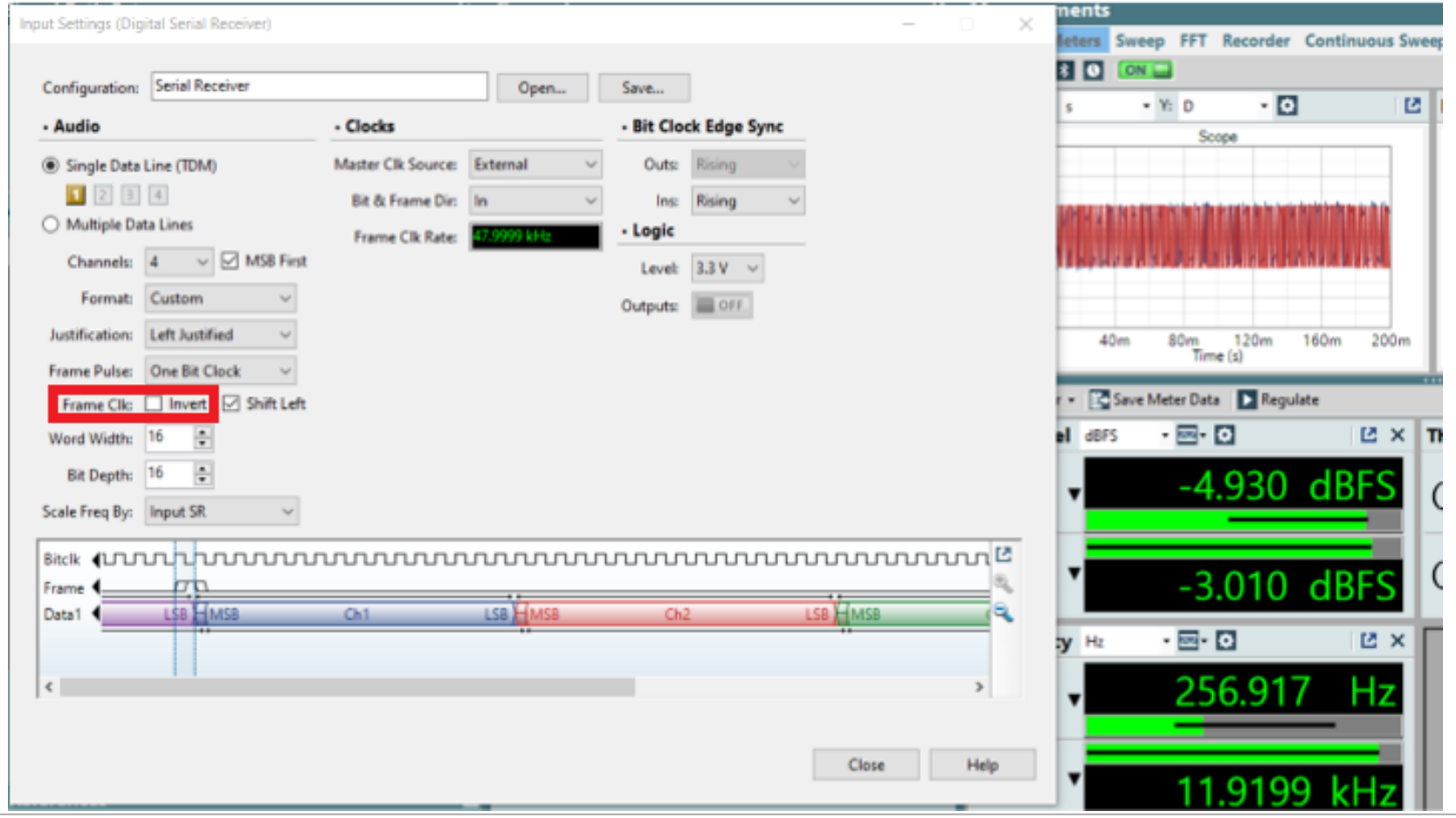
Appendix II | Abnormal echo ref from tas2780-EVM+BBB debug III

- Following is the echo ref data captured by AP



Appendix II | Abnormal echo ref from tas2780-EVM+BBB debug IV

- Using AP to capture the DOUT from tas2780-EVM+BBB with following config without Frame clk Invert selected, the echo ref is abnormal.



Appendix II | Abnormal echo ref from tas2780-EVM+BBB debug V

- According to AP's setting, the Regbin file for TX setting will be corrected(TX_OFFSET=0).

8.9.18 TDM_CFG4 (page=0x00 address=0x0D) [reset=13h]

Bit	Field	Type	Reset	Description
7	TX_KEEPCY	RW	0h	TDM and ICC TX SDOUT LSB data will be driven for full/half cycles when TX_KEEPEN is enabled 0b = Full-cycle 1b = Half-cycle
6	TX_KEEPLN	RW	0h	TDM and ICC TX SDOUT will hold the bus for the following when TX_KEEPEN is enabled 0b = 1 LSB cycle 1b = Always
5	TX_KEEPEN	RW	0h	TDM and ICC TX SDOUT bus keeper enable 0b = Disable bus keeper 1b = Enable bus keeper
4	TX_FILL	RW	1h	TDM and ICC TX SDOUT unused bit field fill 0b = Transmit 0 1b = Transmit Hi-Z
3-1	TX_OFFSET[2:0]	RW	1h	TDM TX start of frame to time slot 0 offset
0	TX_EDGE	RW	1h	TDM TX launch clock polarity 0b = Rising edge of SBCLK 1b = Falling edge of SBCLK

Appendix III | 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 III | DTS in QCOM Platform and Sound card register II

```
                    waipio-qrd-snd-card
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 btfm_bt_sco_a2dp_slim_rx-28 :  : playback 1
00-29: SLIM-DEV1-TX-7 btfm_bt_sco_slim_tx-29 :  : capture 1
gem_row_wifi:/ #
```

Appendix IV | 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 V | How to set DTS for some special cases I

- This guideline mainly introduce the DTS setting for tasdevices, which are same and on the same i2c bus. This page describes how to set DTS for the tasdeives, may be are different, even on the different i2c bus, but with the same tasdevice driver. Here is an extremely example:

  There is one piece of tas5805, and two pieces of tas2562 on i2c-1, two pieces of tas2562 and one piece of tas2780 on i2c-2, two pieces of tas2562 on i2c-3.

  1. Let's define the name-prefix and made the regbin files first.

TASDEVICE	i2c bus	name-prefix	regbin file
tas5805	1	NULL	tas5805-1amp-reg.bin
tas2562	1	rear_left	rear_left-tas2562-2amp-reg.bin
tas2562	2	rear_right	rear_right-tas2562-2amp-reg.bin
tas2780	2	NULL	tas2780-1amp-reg.bin
tas2562	3	NULL	tas2562-2amp-reg.bin

Appendix V | How to set DTS for some special cases II

2. Define the tasdevices in DTS.

```
&i2c1 {
  pinctrl-names = "default";
  #address-cells = <1>;
  #size-cells = <0>;
  status = "okay";
  clock-frequency = <400000>;
  pinctrl-0 = <&i2c1_pins>;
  tas5805: tas5805@2d {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas5805";
    reg = <0x2d>;
    reset-gpios = <&gpio1 10 GPIO_ACTIVE_HIGH>;
  };
  tas2562_1: tas2562@4c {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas2562";
    reg = <0x4c>, <0x4d>;
    sound-name-prefix = "rear_left";
    reset-gpios = <&gpio2 10 GPIO_ACTIVE_HIGH>;
  };
};
```

```
&i2c2 {
  pinctrl-names = "default";
  #address-cells = <1>;
  #size-cells = <0>;
  status = "okay";
  clock-frequency = <400000>;
  pinctrl-0 = <&i2c2_pins>;
  tas2562_2: tas2562@4d {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas2562";
    reg = <0x4d>, <0x4f>;
    sound-name-prefix = "rear_right";
    reset-gpios = <&gpio3 10 GPIO_ACTIVE_HIGH>;
  };
  tas2780: tas2780@3d {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas2780";
    reg = <0x3d>;
    reset-gpios = <&gpio4 10 GPIO_ACTIVE_HIGH>;
  };
};
```

```
&i2c3 {
  pinctrl-names = "default";
  #address-cells = <1>;
  #size-cells = <0>;
  status = "okay";
  clock-frequency = <400000>;
  pinctrl-0 = <&i2c3_pins>;
  tas2562_3: tas2562@4d {
    status = "okay";
    #sound-dai-cells = <0>;
    compatible = "ti,tas2562";
    reg = <0x4d>, <0x4c>;
    reset-gpios = <&gpio5 10 GPIO_ACTIVE_HIGH>;
  };
};
```

Appendix V | How to set DTS for some special cases III

3. Register soundcard in DTS.

```
/ {
  clk_mcasp0_fixed: clk_mcasp0_fixed {
    #clock-cells = <0>;
    compatible = "fixed-clock";
    clock-frequency = <24576000>;
  };

  clk_mcasp0: clk_mcasp0 {
    #clock-cells = <0>;
    compatible = "gpio-gate-clock";
    clocks = <&clk_mcasp0_fixed>;
    enable-gpios = <&gpio1 27 0>; /* BeagleBone Black Clk enable on GPIO1_27 */
  };

  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 = <&tas5805>,
            <&tas2562_1>,
            <&tas2562_2>,
            <&tas2562_3>,
            <&tas2780>;
      };
    };
  };
};
```

am335x-boneblack-hdmi.dtsi

THANKS!