Guideline for TAS2781 Linux driver based on BBB

Kernel-5.10-rt

TEXAS INSTRUMENTS

Revision history

Ver	Date	Author	Description
1.0	2022/9/14		Initial
1.1	2022/9/20		1st round Review & rework
1.2	2022/9/30		Add tinyalsa
1.3	2022/10/9		2nd round Review & rework
1.4	2023/02/05		Add firmware compiling

Information

tasdevice Driver	tas2781
7-bit I^{2}C Address	0x38 ~ 0x3F
Platform	Qualcomm/Mtk/LSI/BeagleBone Black/AMBA
Architecture	ALSA/Tiny-ALSA
Bus type	I2C
Kernel Version	V4.19 ~ V5.10

TEXAS INSTRUMENTS

Release Package

Driver Package	Kconfig, Makefile	Config for compiling
	.c, & .h,	Source code
	dts.readme	Sample for dts setting
	regbin.json	File to generate a sample register setting file with Regbin tool.
	dspfw.json	File to generate a sample dsp firmware and acoustic params file with PPC3 tool.
	Verification report	
	.ftcfg file	Store speaker basic characterization params for speaker calibration
Tool_utility	FTC_tool	Source code for calibration
	PPC3 tool	Tool for sample dsp firmware and acoustic params and ftcfg file
	Non_Integrated_Bin_Tool_v1.3.4	Tool for regbin

Cross-compiler

- Deploy cross-compiler (shell command)

  #sudo apt-get install gcc-arm-linux-gnueabihf lzop

  PS: If you found following installing log,

  pls update apt-get first.

  installing.log

  #apt-get update

- Check the compiler version

  #arm-linux-gnueabihf-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)

TEXAS INSTRUMENTS

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-fix

0002-U-Boot-Bea
one-Cape-Manage

TEXAS INSTRUMENTS

Linux kernel 5.10-rt Compiling

- Download kernel 5.10-rt([GitHub - beagleboard/linux at 5.10-rt](#))

  # 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-gnueabihf lzop 2>&1 | tee install.log

  make clean 2>&1 | tee clean.log

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

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

  compile_BBB-kernel.sh

- Compile the kernel
  #   chmod 777 compile_BBB-kernel.sh
  #   sudo ./compile_BBB-kernel.sh

TEXAS INSTRUMENTS

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

- 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:

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

TEXAS INSTRUMENTS

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:

- Input d, Delete a Specified partition

```
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.
```

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

Partition 2 has been deleted.
```

TEXAS INSTRUMENTS

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/

TEXAS INSTRUMENTS

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/uboot/

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

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

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

TEXAS INSTRUMENTS

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 rdsize ${filesize}
  loaduEnvtxt=load mmc 0:1 ${loadaddr} /boot/uEnv.txt ; env import -t ${loadaddr} ${filesize};
  loadall=run loaduEnvtxt; run loadximage; run loadxfdt;
  mmcargs=setenv bootargs console=tty0 console=${console} ${optargs} ${cape_disable} ${cape_enable} root=${mmcroot}
  rootfstype=${mmcrootfstype} ${cmdline}

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

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

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

TEXAS INSTRUMENTS

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"

TEXAS INSTRUMENTS

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"

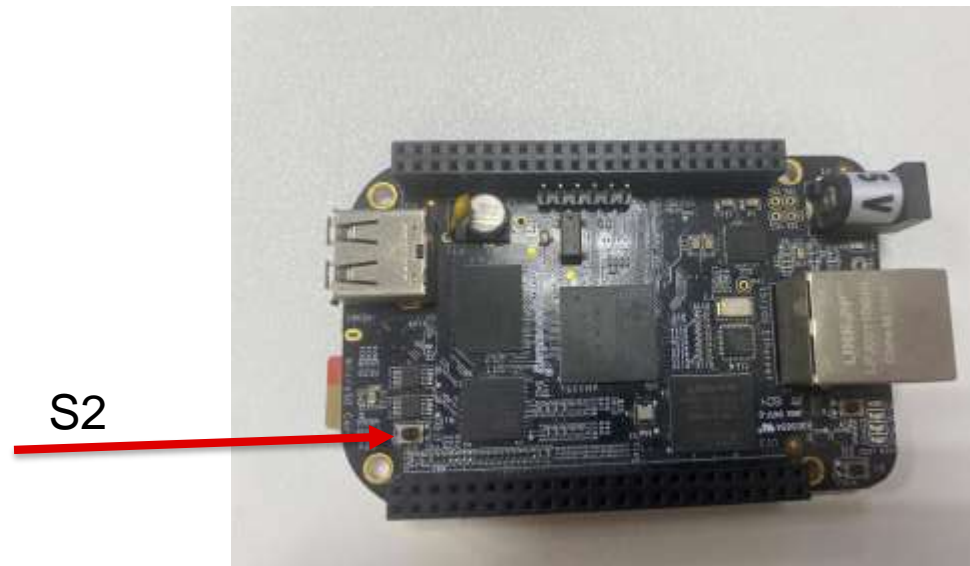TEXAS INSTRUMENTS

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**

S2

TEXAS INSTRUMENTS

Device Tree | I2C

- 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>;
    symlink = "bone/i2c/2";
    pinctrl-0 = <&i2c2_pins>;
    tasdevice: tasdevice@38 {
        status = "okay";
        #sound-dai-cells = <0>;
        compatible = "ti,tas2781";
        reg = <0x38>;
        ti,topleft-channel = <0x38>;
        ti,reset-gpio0 = <&gpio1 10 GPIO_ACTIVE_HIGH>;
        ti,irq-gpio = <&gpio1 15 0>;
    };
};
```

- 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:

am335x-boneblack-hdmi.dtsi

TEXAS INSTRUMENTS

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)
    >;
};
    &i2c2 {
        pinctrl-names = "default";
        #address-cells = <1>;
        #size-cells = <0>;
        status = "okay";
        clock-frequency = <400000>;
        pinctrl-0 = <&i2c2_pins>;
        tasdevice: tasdevice@38 {
            status = "okay";
            #sound-dai-cells = <0>;
            compatible = "ti,tas2781";
            reg = <0x38>;
            ti,topleft-channel = <0x38>;
            ti,reset-gpio0 = <&gpio1 10 GPIO_ACTIVE_HIGH>;
            ti,irq-gpio = <&gpio1 15 0>;
        };
    };
```

- 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:

**am335x-boneblack-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>;
        };
    };
};
```
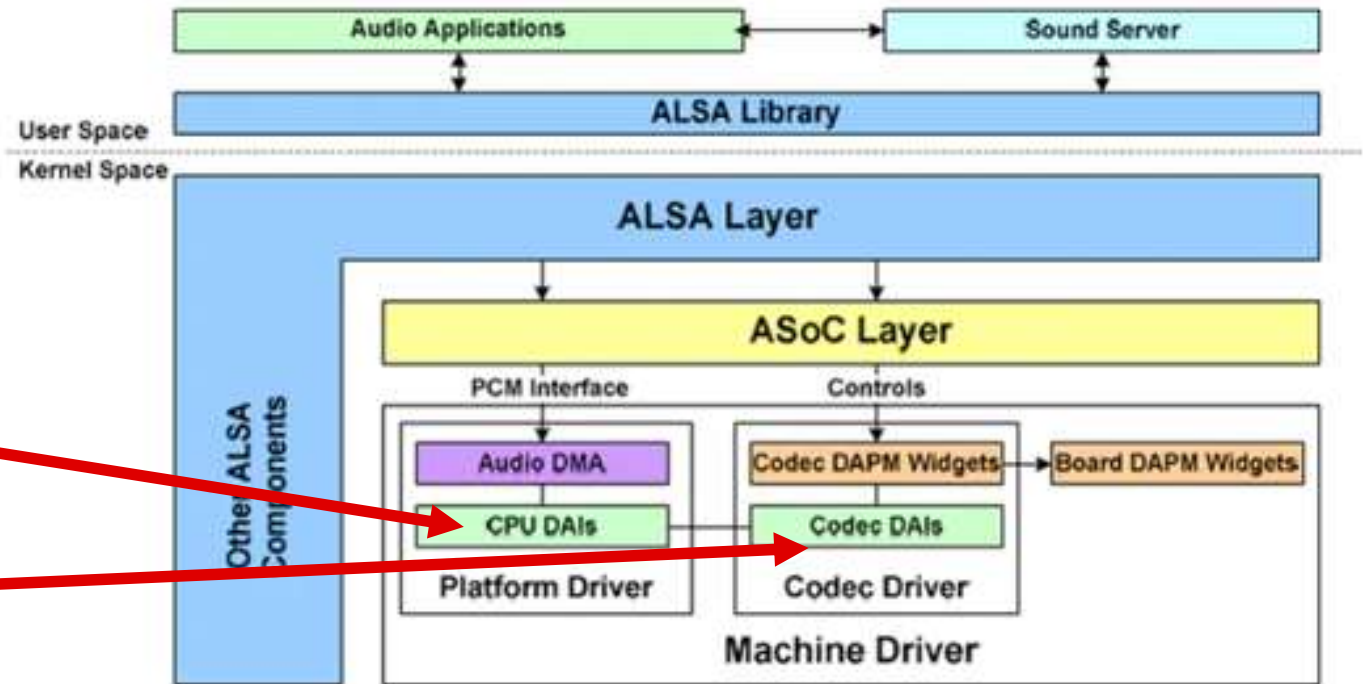
Kconfig & Makefile

Kconfig in sound/soc/codecs

```
 99        imply SND_SOC_HDAC_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
```

TEXAS INSTRUMENTS

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

TEXAS INSTRUMENTS

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

Texas Instruments

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

TEXAS INSTRUMENTS

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 & 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" };

TEXAS INSTRUMENTS

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

TEXAS INSTRUMENTS

Firmware Setting in defconfig

- Compile the bin file into image.

    - Add following settings into defconfig file

    ```
    CONFIG_FW_LOADER=y
    CONFIG_EXTRA_FIRMWARE_DIR="firmware"
    CONFIG_EXTRA_FIRMWARE="tas2781_regbin.bin tas2781_dsp.bin tas2781_cal_38.bin"
    ```

        - ☐ 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"
        - ☐ 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*

TEXAS INSTRUMENTS

Firmware & ftcfg file naming

file	File name	Remark
regbin	tas2781_regbin.bin	Generated by regbin tool
Dsp firmware & params	tas2781_dsp.bin	Generated by ppc3
ftcfg	tas2781_0x38.ftcfg tas2781_0x39.ftcfg …	Generated by ppc3
Calibrated file	tas2781_cal_0x38.bin tas2781_cal_0x39.bin …	Generated by ftc tool

TEXAS INSTRUMENTS

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

TEXAS INSTRUMENTS

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
```

TEXAS INSTRUMENTS

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.

- #amixer contents

- #cat dev_addr

  Illustration: show the active i2c address

- #cat devinfo

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

TEXAS INSTRUMENTS

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

TEXAS INSTRUMENTS

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
```

TEXAS INSTRUMENTS

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-echoreference-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

TEXAS INSTRUMENTS

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 0 when only use tuning mode.
  - PS: For details about regbin configuration, please see the appendix.

Shell script with Alsa

```
#! /bin/bash
amixer cset numid=1,iface=MIXER,name="Program" 1
amixer cset numid=3,iface=MIXER,name="Configuration" 0
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 &
```

Shell script with Tinyalsa

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

TEXAS INSTRUMENTS

Tuning mode

- Write a shell script to test tuning mode. "Program" is 1 when use 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 set "Program" 0
tinymix set "Configuration" 0
tinymix set "TASDEVICE Profile id" 4
tinyplay test.wav &
sleep 1
tinymix set 'TASDEVICE Profile id' 5
tinycap 16bit-echoref.wav &
```

TEXAS INSTRUMENTS

Calibration test

- It's necessary to enable IV sense in PPC3.

- Write a shell script to test calibration. Select the calibration_tuning_mode in "configuration".

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' 4
aplay --device="hw:0,0" silence.wav &
sleep 15
./tas2781_ftc
```

Shell script with Tinyalsa

```
#! /bin/bash
tinymix set "Program" 0
tinymix set "Configuration" 1
tinymix set "TASDEVICE Profile id" 4
tinyplay silence.wav &
sleep 15
./tas2781_ftc
```

TEXAS INSTRUMENTS

FTC tool

- Run FTC tool with root permission
  - # sudo ./tas2781_ftc

- Put TAS2781-A.ftcfg file (Generated by ppc3) into direction: /mnt/vendor/persist/audio/

- After the calibration process finished, the results save into /home/audio/ti/tas2781_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
```

TEXAS INSTRUMENTS

Appendix

TEXAS INSTRUMENTS

Download Kernel 5.10-rt for BBB from…

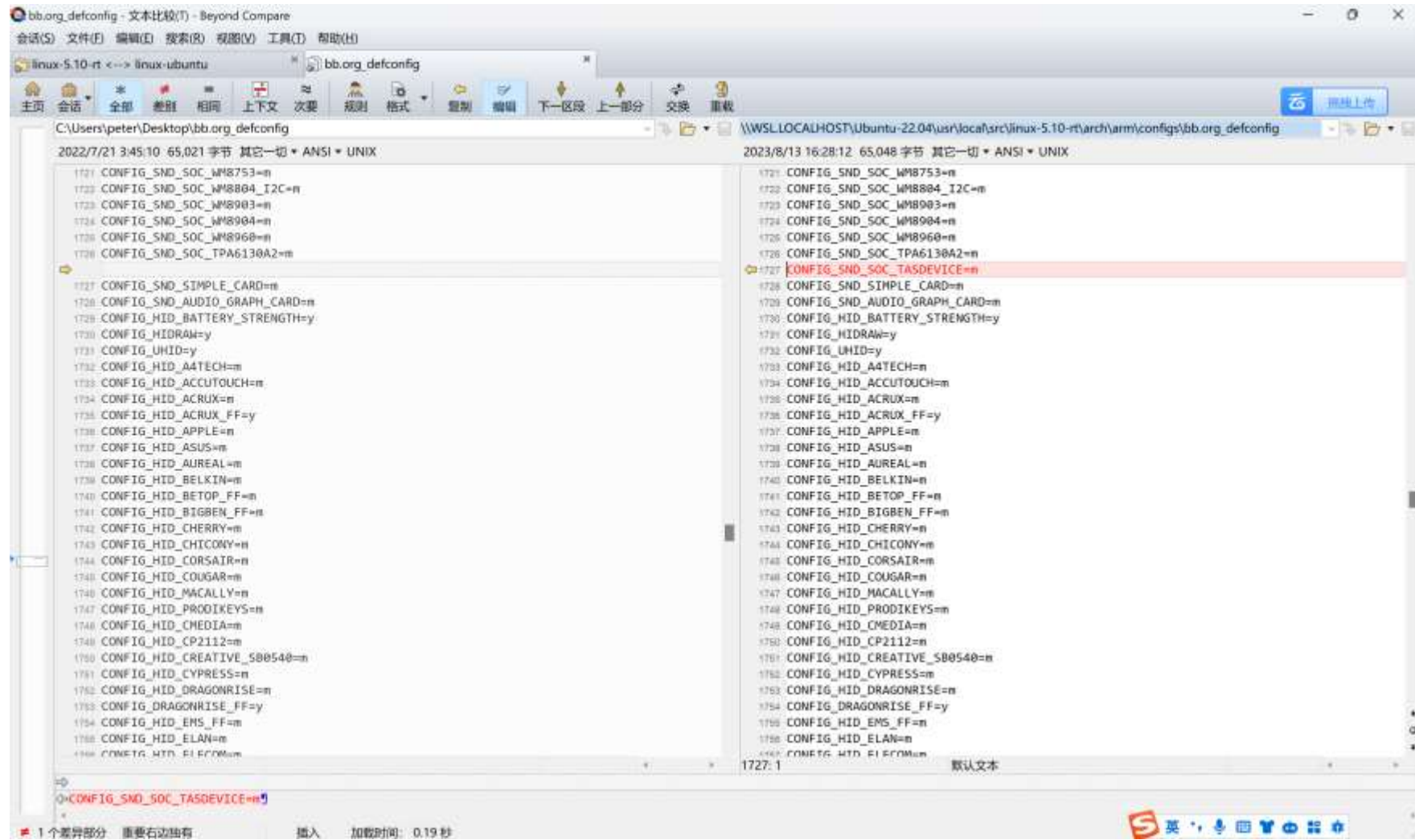- https://github.com/beagleboard/linux/tree/5.10-rt

TEXAS INSTRUMENTS

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.

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-gnueabihf-
   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 tinypcminfo

TEXAS INSTRUMENTS

Add new module to be compiled

TEXAS INSTRUMENTS

THANKS!

TEXAS INSTRUMENTS