



Android10

音频模块使用说明书

1.0
2020.02.26

文档履历

版本号	日期	制/修订人	内容描述
1.0	2020.02.26		

目录

1. 前言	1
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
1.4 相关术语	1
2. 音频系统框架概述	3
2.1 原型机音频硬件框架	3
2.2 软件框架图	3
3. A133 音频模块介绍	5
3.1 ASOC 音频系统	6
3.1.1 Codec 驱动	6
3.1.2 Platform 驱动	7
3.1.3 Machine 驱动	7
3.2 audiocodec 驱动功能	7
3.3 Daudio 模块功能	8
3.4 DMIC 模块功能	8
3.5 SPDIF 模块功能	8
4. A133 音频配置	10
4.1 源码结构	10
4.2 内核配置	10

4.2.1 menuconfig 配置	10
4.3 board.dts 配置	13
4.4 audiocodec 通路配置说明	16
4.4.1 audiocodec 声卡音频控件说明	16
4.4.2 audiocodec 音量调节说明	17
4.4.2.1 音频输出音量、增益控制:	17
4.4.2.2 音频输入音量、增益控制:	19
4.4.3 系统音频场景	21
4.4.3.1 喇叭播放	21
4.4.3.2 板载 MIC 录音	21
4.4.3.3 耳机播放	22
4.4.3.4 耳机 MIC 录音	22
5. FQA	22
5.1 查看播放或录音参数	22
5.2 tinyalsa 工具的使用	24
5.3 音频播放/录制 HAL 层获取数据方法	25
6. Declaration	26

1. 前言

1.1 编写目的

本文档目的是让开发者了解 A133 音频系统框架，能够在 A133 平台上开发新的音频方案。

1.2 适用范围

本模块说明适用于 A133/A100 Android 10 + Linux 4.9 平台。

1.3 相关人员

音频系统开发人员。

1.4 相关术语

- **ALSA: Advanced Linux Sound Architecture**
- **DMA:** 即直接内存存取,指数据不经 **cpu**,直接在设备和内存,内存和内存,设备和设备之间传输。
- **OSS: Open Sound System**
- **样本长度 (sample):** 样本是记录音频数据最基本的单位,常见的有 8 位和 16 位
- **通道数 (channel):** 该参数为 1 表示单声道,2 则是立体声。
- **帧 (frame):** 帧记录了一个声音单元,其长度为样本长度与通道数的乘积。
- **采样率 (rate):** 每秒钟采样次数,该次数是针对帧而言。
- **周期 (period):** 音频设备一次处理所需要的帧数,对于音频设备的数据访问以及音频数据的存储,都是以此为单位。
- **交错模式 (interleave):** 是一种音频数据的记录模式,在交错模式下,数据以连续帧的形式存放,即首先记录完帧 1 的左声道样本和右声道样本(假设为立体声格式),再开始帧 2 的记录,而在非交错模式下,首先记录的是一个周期内所有帧的左声道样本,再记录右声道样本,数据是以连续通道的方式存储。不过多数情况下,我们只需要使用交错模式就可以了。

- **audiocodec**: 芯片内置音频接口
- **daudio**: 数字音频接口，可配置成 i2s/pcm 格式标准音频接口



2. 音频系统框架概述

2.1 原型机音频硬件框架

A133 原型机音频硬件只有 audiocodec，audiocodec 连接 mic 作为输入，speaker 作为输出，headset 即作为输入又作为输出。输入以下命令查看系统当前音频设备节点：

```
cat proc/asound/cards  
0 [sun50iw10codec ]: sun50iw10-codec - sun50iw10-codec  
sun50iw10-codec
```

2.2 软件框架图

音频软件框架如图所示。

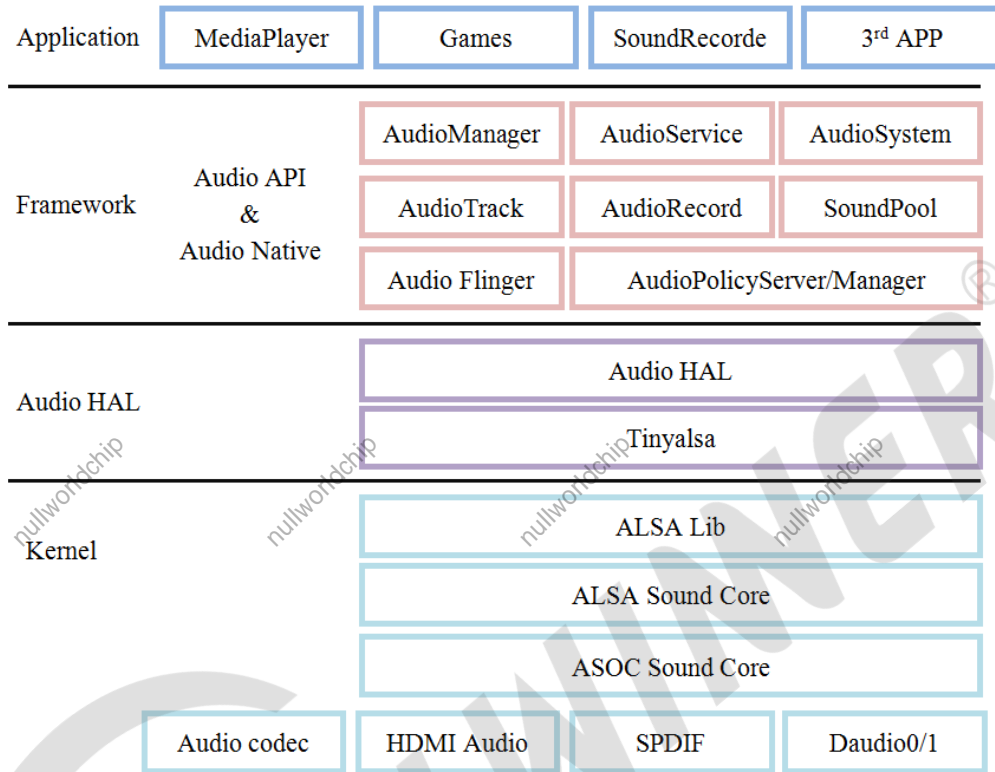


图 1: A133 音频软件框架图

3. A133 音频模块介绍

在 A133 中，存在 7 个音频设备，分别是：

- daudio0
- daudio1
- daudio2
- daudio3
- audiocodec
- DMIC
- SPDIF

硬件框图如图所示：

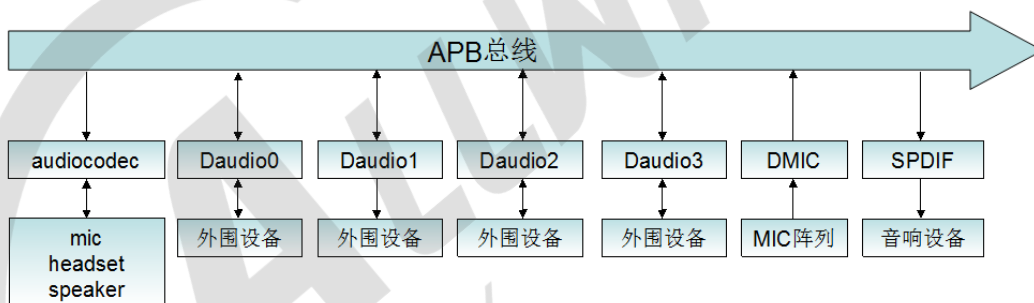


图 2: 音频硬件框图

每一个音频设备都采用 asoc 架构实现，asoc 是建立在标准 alsa 驱动层上，为了更好地支持嵌入式处理器和移动设备中的音频 codec 的一套软件体系，asoc 将音频系统分为 3 部分：Machine，Platform 和 Codec。软件框架图如图所示：

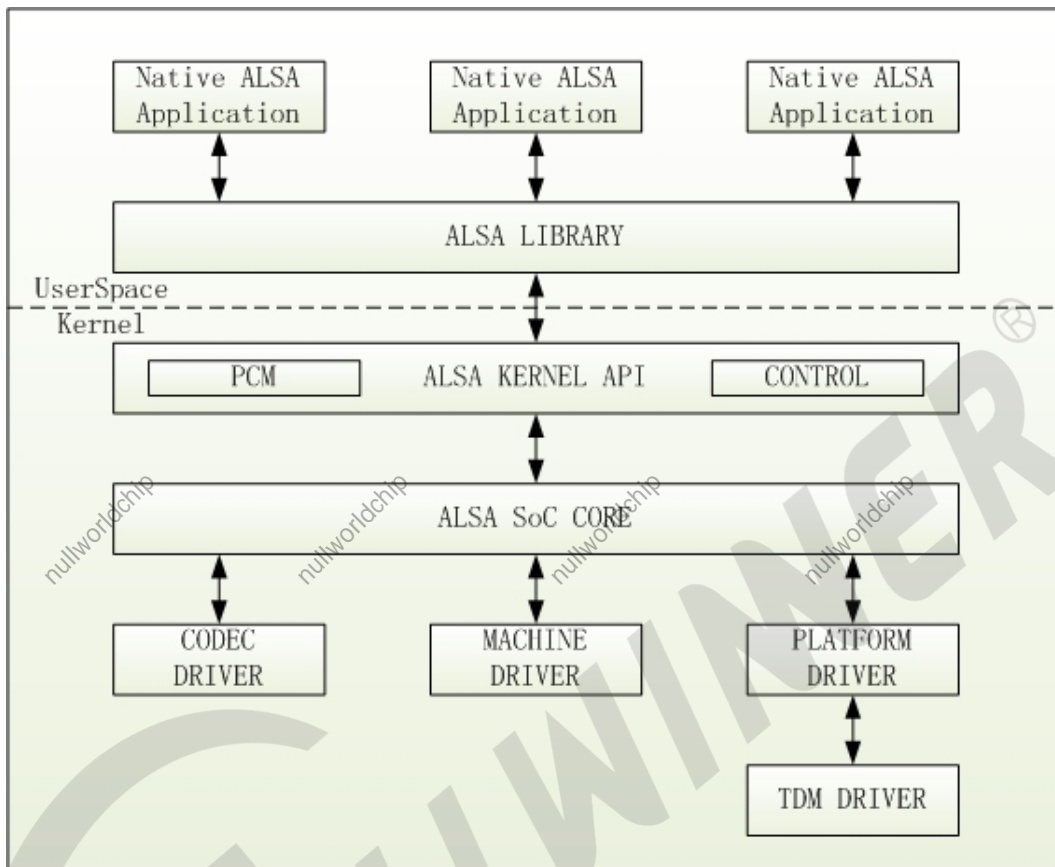


图 3: A133 ALSA 软件框架图

3.1 ASOC 音频系统

3.1.1 Codec 驱动

ASoC 中的一个重要设计原则就是要求 Codec 驱动是平台无关的，它包含了一些音频的控件 (Controls)，音频接口，DAMP (动态音频电源管理) 的定义和某些 Codec IO 功能。为了保证硬件无关性，任何特定于平台和机器的代码都要移到 Platform 和 Machine 驱动中。所有的 Codec 驱动都要提供以下特性：

- Codec DAI (Digital Audio Interface) 和 PCM 的配置信息；
- Codec 的 IO 控制方式 (I2C, SPI 等)；
- Mixer 和其他的音频控件；

- Codec 的 ALSA 音频操作接口；

必要时，也可以提供以下功能：

- DAPM 描述信息；
- DAPM 事件处理程序；
- DAC 数字静音控制；

3.1.2 Platform 驱动

它包含了该 SoC 平台的音频 DMA 和音频接口的配置和控制（I2S，PCM，AC97 等等），一般不包含与板子或 codec 相关的代码。

3.1.3 Machine 驱动

单独的 Platform 和 Codec 驱动是不能工作的，它必须由 Machine 驱动把它们结合在一起才能完成整个设备的音频处理工作。

3.2 audiocodec 驱动功能

audiocodec 是具有数模转换功能的内置模块，可将音频数字信号转换成模拟信号发送出去，通常接耳机、扬声器等。

audiocodec 驱动支持以下功能：

- 支持多种采样率格式 (8KHz, 11.025 KHz, 12 KHz, 16 KHz, 22.05 KHz, 24 KHz, 32 KHz, 44.1 KHz, 48 KHz, 96KHz, 192KHz)；
- 支持 mono 和 stereo 模式；
- 支持同时 playback 和 record(全双工模式)；
- 支持 3 节、4 节耳机插拔检测；

sun50iw10-codec.c : 目录位于 longan/kernel/linux-4.9/sound/soc/sunxi 中，负责 audiocodec 音频 codec 的部分，注册为 codec,codec_dai 模型。

3.3 Daudio 模块功能

Daudio 驱动具有以下功能：

- 支持多种采样率格式（8kHz, 11.025kHz, 16kHz, 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz）
- TDM 模式最多支持 16 个通道
- 支持全双工模式
- 支持 i2s、pcm 配置
- 支持 16-bit、20-bit、24-bit、32-bit 数据精度

`sunxi_daudio.c`：该文件处理 `daudio` 部分，在 `asoc` 中框架中设计为 `cpu_dai` 模型，其中 `platform` 也在此注册。`sunxi-snddaudio.c`：该文件处理 `daudio` 部分，在 `asoc` 中框架中设计为 `machine` 模型。A133 未将 Daudio 配置，若使用需要单独配置。

3.4 DMIC 模块功能

DMIC 驱动具有以下功能：

- 支持多种采样率格式（8kHz, 16kHz, 24kHz, 32kHz, 44.1kHz, 48kHz）
- 支持 16-bit、24-bit 数据精度
- 支持最高 8 通道
- 多个 DMIC 通道同时使用时，采样率必须一致，使能必须同开同关
- 支持过采样率 64OSR 和 128OSR

`sunxi_dmic.c`：该文件处理 `dmic` 部分，在 `asoc` 中框架中设计为 `cpu_dai` 模型，其中 `platform` 也在此注册。`sunxi-snddmic.c`：该文件处理 `sunxi-snddmic` 部分，在 `asoc` 中框架中设计为 `machine` 模型。A133 未将 DMIC 配置，若使用需要单独配置。

3.5 SPDIF 模块功能

SPDIF 驱动具有以下功能：

- 支持多种采样率格式（22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz）
- 支持 mono 和 stereo 模式
- 支持 16-bit、20-bit、24-bit 数据精度
- 支持 raw 数据输出

sunxi-sndspdif.c：该文件处理 spdif 部分，在 asoc 中框架中设计为 machine 模型。sunxi_spdif.c：该文件处理 spdif 部分，在 asoc 中框架中设计为 cpu_dai 模型，其中 platform 也在此注册。A133 未将 SPDIF 配置，若使用需要单独配置。

4. A133 音频配置

4.1 源码结构

代码结构如下所示：

```
\longan\kernel\linux-4.9\sound\soc\sunxi$ tree
├── Kconfig
├── Makefile
├── spdif-utils.c
├── sunxi-cpudai.c
├── sunxi-daudio.c
├── sunxi-daudio.h
├── sun50iw10-codec.c
├── sun50iw10-codec.h
├── sun50iw10-sndcodec.c
├── sunxi-dmic.c
├── sunxi-dmic.h
├── sunxi-snddaudio.c
├── sunxi-snddmic.c
├── sunxi-sndspdif.c
├── sunxi-spdif.c
└── sunxi-spdif.h
```

4.2 内核配置

4.2.1 menuconfig 配置

在编译服务器上，目录为 \longan\kernel\linux-4.9 上，输入命令

```
make ARCH=arm64 menuconfig
```

执行结果如图所示：

```

Linux/arm64 4.9.170 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus
submenus ----). Highlighted letters are hotkeys. Press
<N> excludes, <M> modularizes features. Press <Esc><E
Help, </> for Search. Legend: [*] built-in [ ] exclu

General setup ---->
[*] Enable loadable module support ---->
[*] Enable the block layer ---->
Platform selection ---->
Bus support ---->
Kernel Features ---->
Boot options ---->
Userspace binary formats ---->
Power management options ---->
CPU Power Management ---->
[*] Networking support ---->
Device Drivers ---->
Firmware Drivers ---->
File systems ---->
[ ] Virtualization ---->
Kernel hacking ---->
Security options ---->
-*- Cryptographic API ---->
Library routines ---->
    
```

图 4: menuconfig 配置

音频驱动配置:

- Device Drivers -->
- <*> Sound card support -->
- <*> Advanced Linux Sound Architecture -->

```

Advanced Linux Sound Architecture
Arrow keys navigate the menu. <Enter> selects submenus
submenus ----). Highlighted letters are hotkeys. Press
<N> excludes, <M> modularizes features. Press <Esc><Es
Help, </> for Search. Legend: [*] built-in [ ] exclu

--- Advanced Linux Sound Architecture
< > Sequencer support
< > OSS Mixer API
< > OSS PCM (digital audio) API
[*] PCM timer interface
< > HR-timer backend support
[ ] Dynamic device file minor numbers
[*] Support old ALSA API
[*] Sound Proc FS Support
[*]   Verbose procfs contents
[ ] Verbose printk
[ ] Debug
[*] Generic sound devices --->
    HD-Audio ----
(64) Pre-allocated buffer size for HD-audio driver
[*] SPI sound devices ----
[*] USB sound devices --->
<*> ALSA for SoC audio support --->
    
```

图 5: menuconfig 音频驱动配置

- <*> ALSA for SoC audio support -->
- <*> Allwinner SoC Audio support -->

```
Allwinner SoC Audio support
Arrow keys navigate the menu. <Enter> selects submenu
submenus ----). Highlighted letters are hotkeys. Pre
<N> excludes, <M> modularizes features. Press <Esc><E
Help, </> for Search. Legend: [*] built-in [ ] exclu

[*] Allwinner Sun50iw10 Codec Support
[*] Allwinner sun50iw10 jack support
< > Allwinner Digital Audio Support
< > Allwinner SPDIF Support
< > Allwinner DMIC Support
[ ] Allwinner Audio Debug Support
```

图 6: menuconfig Allwinner 音频驱动配置

原型机只配置 audiocodec，用户可根据需求重新配置。

4.3 board.dts 配置

配置文件目录：longan/device/config/chips/A133/configs/c3/board.dts（c3 根据具体版型选择）

以 audiocodec 为例，配置项及说明如下：

```

/* Audio Driver Modules */
codec:codec@0x05096000 {
    /* MIC and headphone gain setting */
    mic1gain = <0x1f>;
    mic2gain = <0x1f>;
    /* ADC/DAC DRC/HPF func enabled */
    /* 0x1:DAP_HP_EN; 0x2:DAP_SPK_EN; 0x3:DAP_HPSPK_EN */
    adcdrc_cfg = <0x0>;
    adchpf_cfg = <0x0>;
    dacdrc_cfg = <0x0>;
    dachpf_cfg = <0x0>;
    /* Volume about */
    digital_vol = <0x00>;
    lineout_vol = <0x1a>;
    headphonegain = <0x00>;
    /* Pa enabled about */
    pa_level = <0x01>;
    pa_msleep_time = <0x78>;
    gpio-spk = <&pio PH 6 1 1 1 1>;
    /* regulator about */
    avcc-supply = <&reg_aldol>;
    cpvin-supply = <&reg_eldol>;
    status = "okay";
};

sndcodec:sound@0 {
    status = "okay";
};
    
```

配置项	配置项
mic1gain	mic1录音增益配置初始值，当前默认最大值0x1f (31)
mic2gain	mic2录音增益配置初始值，当前默认最大值0x1f (31)
adcdrc_cfg	输入ADC/DRC功能开关，0关闭，其他开
adchpf_cfg	输入ADCHPF功能开关，0关闭，其他开
dacdrc_cfg	输出DAC/DRC功能开关，0关闭，其他开
dachpf_cfg	输出DACHPF功能开关，0关闭，其他开
digital_vol	输出数字音量配置初始值，当前默认最大值寄存器写值0x00
lineout_vol	LINEOUT输出音量配置初始值，当前默认值0x1a (26)
headphonegain	HPOUT耳机、喇叭输出增益配置初始值，当前默认最大值寄存器写值0x00
pa_level	功能使能设置值，默认值0x1
pa_msleep_time	功放使能引脚使能延时时间
gpio-spk	对应功能芯片使能引脚
*-supply	音频电路相关电源管理
status	音频驱动模块使能，“okay”打开，“disabled”关闭

图 7: board.dtsaudiocodec 配置说明

以 daudio1 为例，配置说明如下：

```

daudio1:daudio@0x05091000 {
    mclk_div = <0x01>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x80>;
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    slot_width_select = <0x20>;
    status = "disabled";
};

snddaudio1:sound@4 {
    status = "disabled";
};
    
```

配置项	配置项含义
status	是否开启daudio1/snddaudio1, okay: 开启, disabled: 不开启
mclk_div	时钟分频, 默认 0x00
frametype	长帧或短帧 0: long frame = 2 clock width; 1: short frame
tdm_config	i2s或pcm选择 0:pcm 1:i2s
sign_extend	未使用
tx_data_mode	未使用
rx_data_mode	未使用
msb_lsb_first	未使用
pcm_lrck_period	表示LRCK时钟占有的BCLK周期数 (BCLK/LRCK), pcm格式: 32/64/128/256, i2s格式: 16/32/64/128
audio_format	1 SND_SOC_DAIFMT_I2S(standard i2s format). use 表示标准 i2s 格式; 2 SND_SOC_DAIFMT_RIGHT_J(right justified format). 表示右对齐格式; 3 SND_SOC_DAIFMT_LEFT_J(left justified format) 表示左对齐格式; 4 SND_SOC_DAIFMT_DSP_A 短帧模式 并设置 frame_width 为 0.短帧; 5 SND_SOC_DAIFMT_DSP_B 长帧模式 并设置 frame_width 为 1.长帧;
daudio_master	Master/slave模式: 1:daudio1 slave; 4:daudio1 master
signal_inversion	信号的翻转, 比如标准的 i2s 模式, 如果 LRCK 翻转是模式, 那么用示波器测量, 左右声道是跟标准 i2s 模式相反的。如果 BCLK 是翻转模式, 那么用示波器测量, BCLK 信号是翻转的。 1 SND_SOC_DAIFMT_NB_NF(normal bit clock + frame) use 表示 BCLK 采用正常模式, LRCK 也正常模式 2 SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM) 表示 BCLK 采用正常模式, LRCK 采用翻转模式 3 SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM) use 表示 BCLK 采用翻转模式, LRCK 采用正常模式 4 SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM) 表示 BCLK 采用翻转模式, LRCK 采用翻转模式
slot_width_select	数据 word 的宽度, 对 i2s 模式, pcm 模式都有效。16bits/20bits/24bits/32bits

图 8: board.dtsaudio 配置说明

4.4 audiocodec 通路配置说明

A133 audiocodec 路由通路如下所示。

```

Mixer name: 'sun50iw10-codec'
Number of controls: 15
ctl      type      num      name                                     value
0        ENUM      1        codec hub mode                          hub_disable
1        ENUM      1        ADC Swap                                Off
2        INT       1        digital volume                           63
3        INT       1        MIC1 gain volume                         31
4        INT       1        MIC2 gain volume                         31
5        INT       1        LINEOUT volume                           26
6        INT       2        DAC volume                               160 160
7        INT       2        ADC volume                               160 160
8        INT       1        headphone gain                           7
9        ENUM      1        LINEOUT Output Select                    DAC_SINGLE
10       BOOL      1        ADCL Input MIC1 Boost Switch             Off
11       BOOL      1        ADCR Input MIC2 Boost Switch             Off
12       BOOL      1        Headphone Switch                         On
13       BOOL      1        HpSpeaker Switch                         On
14       BOOL      1        LINEOUT Switch                           Off
    
```

图 9: audiocodec 路由通路

4.4.1 audiocodec 声卡音频控件说明

audiocodec 声卡音频控件说明如下：

序号	名称	选值范围	说明
0	codec hub mode	0 / 1	模块同源输出使能，默认值0关闭，（0-disable / 1-enable）
1	ADC Swap	0 / 1	ADC录音左右通道交换，默认值0关闭，（0-Off / 1-On）
2	digital volume	0 ~ 63	数字音量大小控制，默认最大值63，可用于调节输出音量大小
3	MIC1 gain volume	0 ~ 31	MIC1录音增益大小控制，默认最大值31，原型机对应板载MIC，可用于调节MIC1录音音量大小
4	MIC2 gain volume	0 ~ 31	MIC1录音增益大小控制，默认最大值31，原型机对应耳机MIC，可用于调节MIC2录音音量大小
5	LINEOUT volume	0 ~ 31	喇叭输出音量大小控制，默认值26，注意仅适用于硬件外围电路LINEOUT信号后接喇叭输出情况下控制喇叭输出音量大小，其他情况无效
6	DAC volume	0 ~ 255	DAC通道音量控制，默认值160，可用于调节输出音量大小
7	ADC volume	0 ~ 255	ADC通道音量控制，默认值160，可用于调节输入音量大小
8	headphone gain	0 ~ 7	耳机、喇叭输出增益大小控制，默认最大值7，注意仅适用于硬件外围电路HPROUT信号后接耳机、喇叭输出情况下控制输出音量大小，其他情况无效
9	LINEOUT Output Select	0 / 1	喇叭差分输出选择，（0-非差分输出 / 1-差分输出）注意仅适用于硬件外围电路LINEOUT信号后接喇叭输出情况下
10	ADCL Input MIC1 Boost Switch	0 / 1	MIC1（板载MIC）录音开关控制，（0-Off / 1-On）
11	ADCR Input MIC2 Boost Switch	0 / 1	MIC2（耳机MIC）录音开关控制，（0-Off / 1-On）
12	Headphone Switch	0 / 1	耳机输出开关控制，控制耳机音频输出，（0-Off / 1-On）
13	HpSpeaker Switch	0 / 1	喇叭输出开关控制，（0-Off / 1-On），注意仅适用于硬件电路HPROUT信号后接喇叭输出控制，且前置条件需打开耳机输出开关，其他情况无效
14	LINEOUT Switch	0 / 1	喇叭输出开关控制，（0-Off / 1-On），注意仅适用于硬件电路LINEOUT信号后接喇叭输出控制，其他情况无效

图 10: audiocodec 音频控件说明

4.4.2 audiocodec 音量调节说明

注意：A133 版型与过往的 A50 版型在音频通路设计等方面上都存在着差异，故驱动层音频相关设计上也会有所差异与调整，具体仍需以当前 A133 版型配置为准。

4.4.2.1 音频输出音量、增益控制：

1、输出数字音量控制：可调整相应 audiocodec 模块的音频输出数字音量，可调节范围：0~63，驱动默认最大值 63。

- 相应音频控件：digital volume
- 相应寄存器读写：寄存器地址 0x05096000, 寄存器位：bit:17:12 (注意：该寄存器位写值越小实际音量越大，寄存器位写值越大实际音量越小)。

寄存器读操作：

```
console:/ #echo 0x05096000 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作：

```
console:/ #echo 0x05096000 0x00(value) > /sys/class/sunxi_dump/write
```

2、DAC 左右通道音量控制：可调整相应 audiocodec 模块的 DAC 的左右通道输出音量，可调节范围：0~255，驱动默认配置值 160。

- 相应音频控件：DAC volume
- 相应寄存器读写：寄存器地址 0x05096004, 寄存器位：bit:15:8(L), bit: 7:0(R)。

寄存器读操作：

```
console:/ #echo 0x05096004 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作:

```
console:/ #echo 0x05096004 0x00(new value) > /sys/class/sunxi_dump/write
```

3、HPOUT 耳机、喇叭输出增益控制: 可调整相应 audiocodec 模块的 HPOUT 输出增益控制, 可调节范围: 0~7, 驱动默认配置值 7。

- 相应音频控件: **headphone gain**
- 相应寄存器读写: 寄存器地址 0x05096310, 寄存器位: bit:30:28 (注意: 该寄存器位写值越小实际音量越大, 寄存器位写值越大实际音量越小)。

寄存器读操作:

```
console:/ #echo 0x05096310 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作:

```
console:/ #echo 0x05096310 0x00(new value) > /sys/class/sunxi_dump/write
```

4、LINEOUT 输出音量控制: 可调整相应 audiocodec 模块的 LINEOUT 输出的音量控制, 可调节范围: 0~31, 驱动默认配置值 26。

- 相应音频控件: **LINEOUT volume**
- 相应寄存器读写: 寄存器地址 0x05096310, 寄存器位: bit:4:0。

寄存器读操作:

```
console:/ #echo 0x05096310 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作:

```
console:/ #echo 0x05096310 0x00(new value) > /sys/class/sunxi_dump/write
```

注意: 该音量控制仅用于 IC LINEOUT 信号后接的输出音量控制, 其他情况无效。

4.4.2.2 音频输入音量、增益控制:

1、ADC 左右通道音量控制: 可调整相应 audiocodec 模块的 ADC 的左右通道输入音量, 可调节范围: 0~255, 驱动默认配置值 160。

- 相应音频控件: ADC volume
- 相应寄存器读写: 寄存器地址 0x05096034, 寄存器位: bit:15:8(L), bit: 7:0(R)。

寄存器读操作:

```
console:/ #echo 0x05096034 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作:

```
console:/ #echo 0x05096034 0x00(new value) > /sys/class/sunxi_dump/write
```

2、MIC1 输入增益控制: 可调整相应 audiocodec 模块的 MIC1 的输入增益, 可调节范围: 0~31, 驱动默认配置值 31。

- 相应音频控件: MIC1 gain volume
- 相应寄存器读写: 寄存器地址 0x05096300, 寄存器位: bit:12:8。

寄存器读操作:

```
console:/ #echo 0x05096300 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作:

```
console:/ #echo 0x05096300 0x00(new value) > /sys/class/sunxi_dump/write
```

3、MIC2 输入增益控制: 可调整相应 audiocodec 模块的 MIC2 的输入增益, 可调节范围: 0~31, 驱动默认配置值 31。

- 相应音频控件: MIC2 gain volume
- 相应寄存器读写: 寄存器地址 0x05096304, 寄存器位: bit:12:8。

寄存器读操作:

```
console:/ #echo 0x05096304 > /sys/class/sunxi_dump/dump && cat /sys/class/sunxi_dump/dump
```

寄存器写操作:

```
console:/ #echo 0x05096304 0x00(new value) > /sys/class/sunxi_dump/write
```

注意: 若是需要进行音频输入、输出功率测试则需要事先确认相应的驱动层 audiocodec 模块的输入输出 DRC 功能配置已关闭。

具体修改操作:

- 配置文件路径: /longan/device/config/chips/A133/configs/(具体板型)/board.dts
- 配置修改操作如下图所示:

```

/* Audio Driver Modules */
codec:codec@0x05096000 {
    /* MIC and headphone gain setting */
    mic1gain      = <0x1F>;
    mic2gain      = <0x1F>;
    /* ADC/DAC DRC/HPF func enabled */
    /* 0x1:DAP_HP_EN; 0x2:DAP_SPK_EN; 0x3:DAP_HPSPK_EN */
    adcdrc_cfg    = <0x0>;
    adchpf_cfg    = <0x0>;
    dacdrc_cfg    = <0x0>;
    dachpf_cfg    = <0x0>;
    /* volume about */
    digital_vol   = <0x00>;
    lineout_vol   = <0x1a>;
    headphonegain = <0x00>;
    /* Pa enabled about */
    pa_level      = <0x01>;
    pa_msleep_time = <0x78>;
    gpio-spk     = <&pio PH 6 1 1 1 1>;
    /* regulator about */
    avcc-supply  = <&reg_ald01>;
    cpvin-supply = <&reg_eld01>;
    status = "okay";
};
    
```

全部选择配置为0
关闭

图 11: audiocodecDRC 功能关闭

4.4.3 系统音频场景

4.4.3.1 喇叭播放

number	ctl_name	value
1	Headphone Switch	1
2	HpSpeaker Switch	1

4.4.3.2 板载 MIC 录音

number	ctl_name	value
1	ADCL Input MIC1 Boost Switch	1
2	ADCR Input MIC2 Boost Switch	0

4.4.3.3 耳机播放

number	ctl_name	value
1	Headphone Switch	1
2	HpSpeaker Switch	0

4.4.3.4 耳机 MIC 录音

number	ctl_name	value
1	ADCL Input MIC1 Boost Switch	0
2	ADCR Input MIC2 Boost Switch	1
3	ADC Swap	1

5. FQA

5.1 查看播放或录音参数

原型机只使用 `codec` 播放，因此只有 `audiocodec` 一个设备节点，输入以下命令查看系统当前音频设备节点：

```
cat proc/asound/cards
0 [sun50iw10codec ]: sun50iw10-codec - sun50iw10-codec
sun50iw10-codec
```

执行以下命令可获取当前播放硬件参数，如下所示：

```
cat proc/asound/card0/pcm0p/sub0/hw_params

access: RW_INTERLEAVED
format: S16_LE
subformat: STD
channels: 2
rate: 48000 (48000/1)
period_size: 2720
buffer_size: 5440
```

若当前正在播放，执行以下命令可获取当前播放状态，如下所示：

```
cat /proc/asound/card0/pcm0p/sub0/status
state: RUNNING
owner_pid : 4820
trigger_time: 1578540242.793497380
tstamp   : 1578540410.366986967
delay    : 3040
avail    : 2400
avail_max : 2656
-----
hw_ptr   : 7390016
appl_ptr : 7393056
```

执行以下命令可获取当前录音硬件参数，如下所示：

```
.cat proc/asound/card0/pcm0p/sub0/hw_params
access: RW_INTERLEAVED
format: S16_LE
subformat: STD
channels: 2
rate: 48000 (48000/1)
period_size: 1024
buffer_size: 2048
```

若当前正在录音，执行以下命令可获取当前录音状态，如下所示：

```
cat proc/asound/card0/pcm0p/sub0/status
state: RUNNING
owner_pid : 4187
```

```
trigger_time: 4291.710215239
tstamp      : 4314.001112062
delay       : 1216
avail       : 832
avail_max   : 1184
-----
hw_ptr      : 1069984
appl_ptr    : 1071200
```

在没有打开任何节点，既不播放也不录音，或获取的设备节点并未播放或录音时，状态为 `closed`，如下所示：

```
cat /proc/asound/card0/pcm0c/sub0/hw_params
closed
```

5.2 tinypalsa 工具的使用

在 `android/external/tinypalsa` 目录下使用 `mm` 编译，会生成 `tinycap` `tinypaly` `tinymix` `tinypcminfo` `tinypcmhostless` 这五个调试工具，`tinypcminfo` `tinypcmhostless` 在 A133 平台中未使用。

编译生成的调试工具均在 `/android/out/target/product/版本号/system/bin` 下，以 `tinymix` 举例，安装命令方法：

1. `adb root`
2. `adb remount`
3. `adb push tinymix /system/bin/`
4. `adb shell`
5. `cd system/bin/`
6. `chmod 777 tinymix`

调试工具的用途与用法：

1. **tinycap** 录音测试工具。用于操作 `audiocedec`，DMIC 的音频录音设备节点。

```
Usage: tinycap file.wav [-D card] [-d device] [-c channels] [-r rate] [-b bits] [-p period_size] [-n n_periods] [-T capture time]
```

2. **tinypplay** 播放测试工具。用于操作 **audiocedec** 的音频播放设备节点。

```
Usage: tinypplay file.wav [-D card] [-d device] [-p period_size] [-n n_periods]
```

3. **tinymix** 查看音频通路相关的各项配置参数，并通过命令修改参数配置。

5.3 音频播放/录制 HAL 层获取数据方法

获取步骤，源码请参考 `audio_data_dump.c`:

1. 更改 **selinux** 权限，小机中执行命令 `setenforce 0`
2. 设置抓取输出或者输入数据属性
 - `setprop persist.vendor.audio.dump_data.out true`
 - `setprop persist.vendor.audio.dump_data.in true`
3. 默认文件保存路径：`/data/vendor/audio_d/out.pcm` / `/data/vendor/audio_d/in.pcm`

6. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This document neither states nor implies warranty of any kind, including fitness for any particular application.