

H6 音频模块说明书

0.8

2017.03.27

文档履历

版本号	日期	制/修订人	内容描述
0.8	2017.03.27		

confidential

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. H6 音频系统框架概述	3
2.1 H6 原型机音频硬件框架图	3
2.2 H6 软件框架图	4
3. H6 音频模块介绍	5
3.1 公共部分	6
3.2 Audio_hub 驱动功能	6
3.3 AC200 驱动功能	7
3.4 Daudio 模块功能	8
3.5 HDMI 模块功能	8
3.6 DMIC 模块功能	9
3.7 SPDIF 模块功能	9
4. H6 音频配置	10
4.1 源码结构	10
4.2 内核配置	11
4.2.1 menuconfig 配置	11

4.3 sys_config 配置	14
4.3.1 DMIC 驱动加载	14
4.3.2 SPDIF 驱动加载	15
4.3.3 HDMI 驱动加载	15
4.3.4 CODEC 驱动加载	16
4.3.5 蓝牙 SCO 驱动挂载	18
4.4 AC200 通路配置说明	18
4.4.1 系统音频播放场景	19
4.4.1.1 系统 LINEOUT 输出	19
4.4.1.2 Linein 输入到系统	20
4.4.1.3 Mic1 输入到系统	20
4.4.1.4 Mic2 输入到系统	20
4.4.1.5 Mic1 + Mic2 立体声输入到系统	21
4.4.1.6 Mic1 直接输出到 LINEOUT	21
4.5 Audio_hub 配置说明	22
4.5.1 Audio_hub 音频路径配置	22
4.6 Audio_hub 操作流程	24
5. 音频输入输出切换策略	28
5.1 FVD 音频输出策略	28
5.2 音频输入策略	29
6. TV_BOX 音频特性	31
6.1 单/多通路音频输出的使用	31

6.2 音频设备热插拔	33
6.3 音频透传的使用	35
6.4 硬件混音的使用	37
6.5 tinyalsa 命令的使用	38
7. FAQ	42
8. Declaration	44

confidential

1. 前言

1.1 编写目的

本文档目的是让开发者了解 H6 音频系统框架，能够在 H6 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 H6 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。

- HDMIaudio: 内置 hdmi 音频接口
- SPDIF: 外置音响音频设备接口，一般使用同轴电缆或光纤接口
- I2S: 外置音频通道接口
- AGC: 录音自动增益控制
- DRC: 音频输出动态范围控制
- daudio: 数字音频接口，可配置成 i2s/pcm 格式标准音频接口
- aif: 数字音频接口
- xrun: 音频流异常状态

confidential

2. H6 音频系统框架概述

2.1 H6 原型机音频硬件框架图

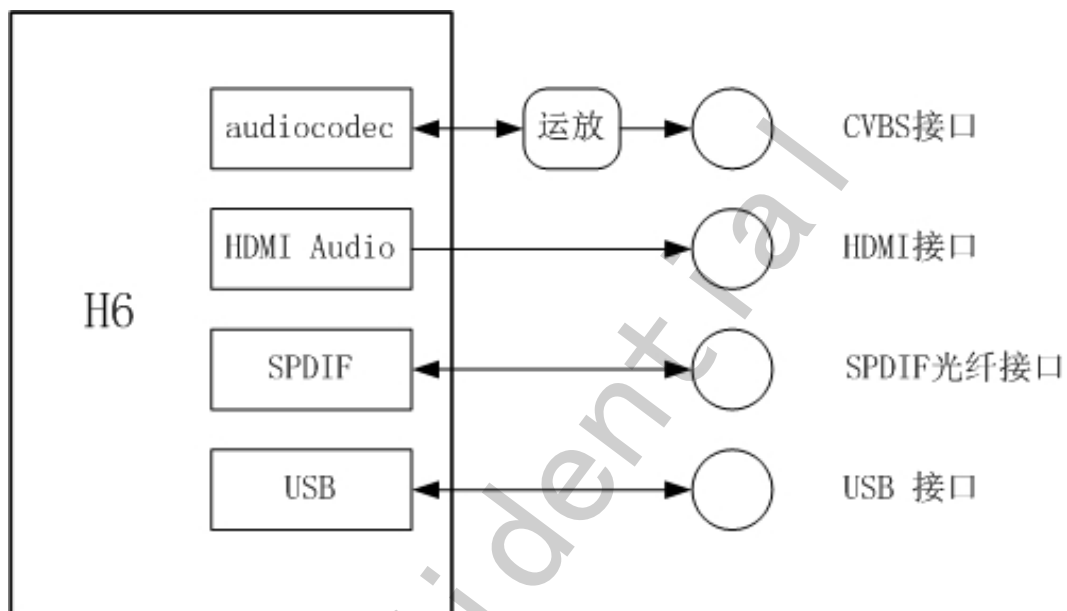


图 1

输入以下命令查看系统当前音频设备节点：

```
cat proc/asound/cards
0 [sndacx00codec ]: sndacx00-codec - sndacx00-codec
    sndacx00-codec
1 [sndhdmi      ]: sndhdmi - sndhdmi
    sndhdmi
2 [sndspdif     ]: sndspdif - sndspdif
    sndspdif
```


2.2 H6 软件框架图

H6 音频软件框架如图所示，H6 盒子有一套独立的音频输入输出策略，同时具备原生系统所不具备的功能，例如支持 HDMI，USB，CVBS 等接口的热插拔，支持单/多路输出，支持音频透传等。

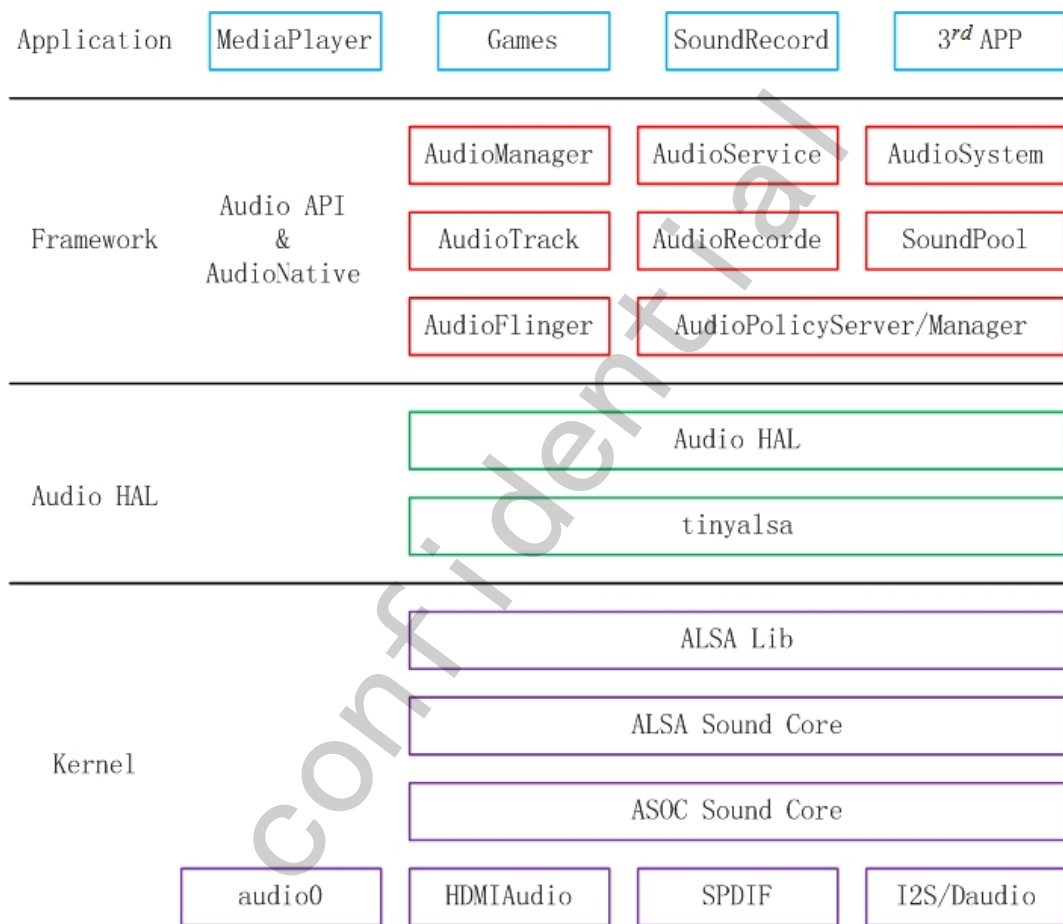


图 2

3. H6 音频模块介绍

在 H6 中，存在 6 个音频设备，分别是：

- daudio0
- daudio1（接 HDMI）
- daudio2
- daudio3（AC200 声卡）
- DMIC
- SPDIF

硬件框图如图所示：

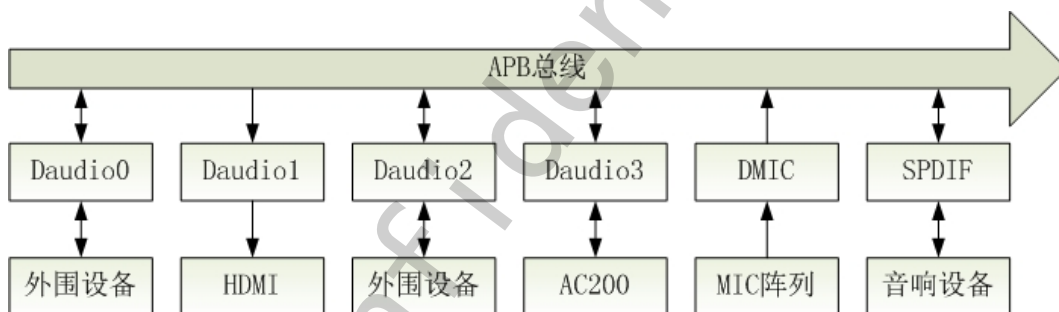


图 3

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：

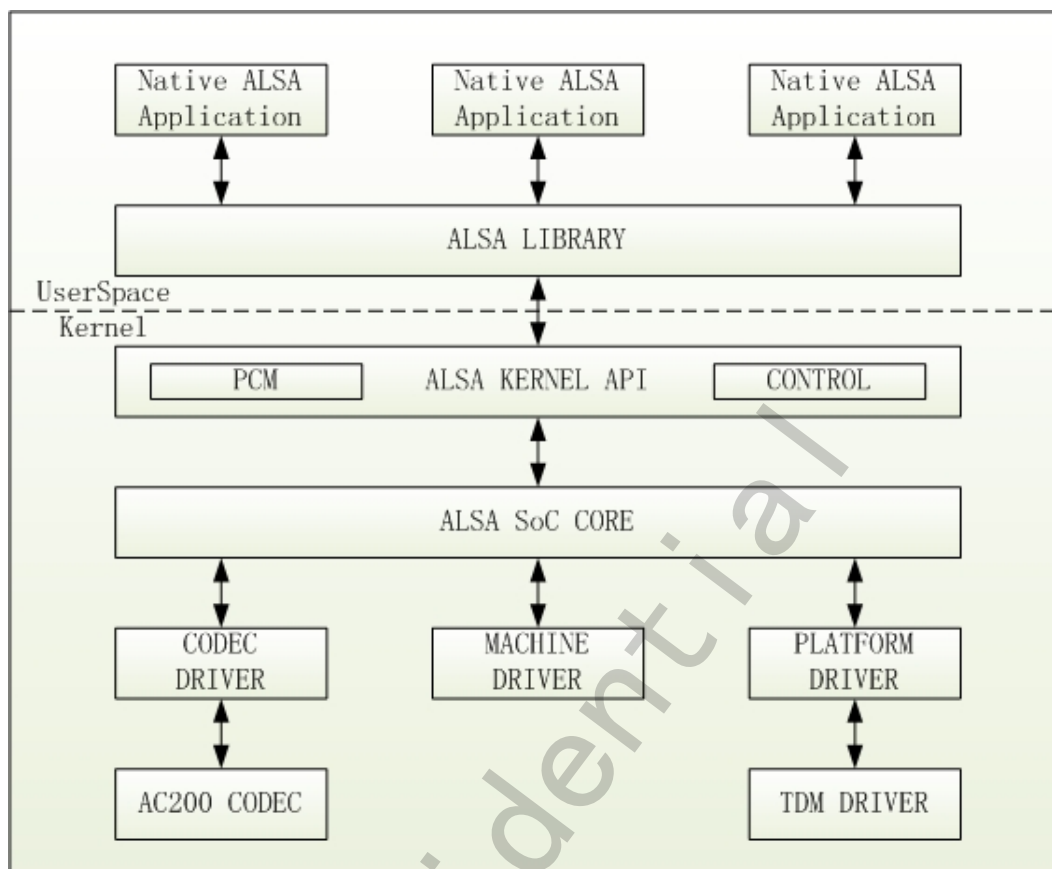


图 4

3.1 公共部分

Platform (dma 注册):

sunxi_dma.c : 该文件内处理 dma 部分，主要负责提供注册 platform 设备的公共函数。

3.2 Audio_hub 驱动功能

H6 有两组 I2S，一组沿用以前平台，另一组工 Audio_hub 使用，两组不可同时使用。Audio_hub 是 H6 特有模块，集成了音频的基本输入输出功能，还有硬件混音特殊功能，常应用在卡拉 OK 场景。具有三个输入端，四路 I2S，即可完成 3 路数据的混音，混音通路如图所示：

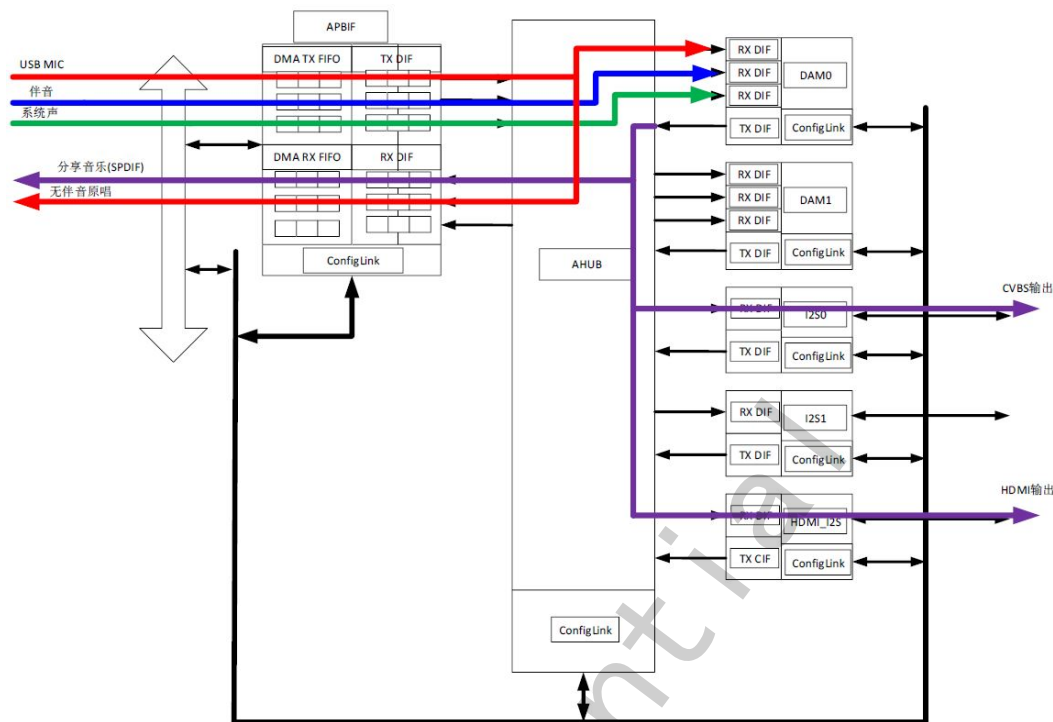


图 5

3.3 AC200 驱动功能

AC200 是具有数模转换和模数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 CVBS，也可以接 mic 录音，将模拟音频数据转换成数字信号供上层应用使用。

AC200 驱动支持以下功能：

- 播放录音支持多种采样格式 (8khz, 11.025khz, 16khz, 22.05khz, 24khz, 32khz, 44.1khz, 48khz, 96khz, 192khz)
- 支持 mono 和 stereo 模式
- 支持全双工模式
- 支持 Audio hub 功能

acx00.c: 目录位于 sound/soc/codec 中，负责 ac200 音频 codec 的部分，注册为 codec,codec_dai 模型

3.4 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）
- 支持 mono 和 stereo 模式
- 支持全双工模式
- 支持 i2s、pcm 配置
- 支持 16bit 24bit 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

3.5 HDMI 模块功能

HDMI 驱动具有以下功能：

- 支持多种采样率格式（32khz, 44.1khz, 48khz, 96khz, 192khz）
- 支持 mono 和 stereo 模式
- 只支持 playback 模式，不支持 record 模式
- 支持 16bit 24bit 数据精度
- 支持 raw 数据输出

`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

`sndhdm.c`：该文件处理 HDMI 解码库接口设置部分，在 asoc 中框架中设计为 `codec` 模型

`sunxi-sndhdm.c`：该文件处理 daudio1 部分，在 asoc 中框架中设计为 `machine` 模型

3.6 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 16bit 24bit 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`：该文件处理 `sunxi-snddmic` 部分，在 `asoc` 中框架中设计为 `machine` 模型

3.7 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
- 支持全双工模式
- 支持 16bit 24bit 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

4. H6 音频配置

4.1 源码结构

代码结构如下所示：

```
lichee/linux-3.10/sound/soc/sunxi$ tree |——Kconfig
```

```
|——Makefile
```

```
|——snddaudio.c
```

```
|——sndhdmi.c
```

```
|——spdif-utils.c
```

```
|——sunxi_ahub.c
```

```
|——sunxi_ahub_cpudai.c
```

```
|——sunxi_ahub_daudio.c
```

```
|——sunxi_ahub.h
```

```
|——sunxi_cpudai.c
```

```
|——sunxi_cpudai.h
```

```
|——sunxi_daudio.c
```

```
|——sunxi_daudio.h
```

```
|——sunxi_dma.c
```

```
|——sunxi_dma.h
```

```
|——sunxi_dmic.c
```

```
|——sunxi_dmic.h
```

```
|——sunxi_rw_func.c
```

- |——sunxi_rw_func.h
- |——sunxi-sndahub.c
- |——sunxi_sndcodec.c
- |——sunxi-snddaudio.c
- |——sunxi-snddmic.c
- |——sunxi-snddsd.c
- |——sunxi-sndhdmi.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 \lichee\linux-3.10 上，输入命令

```
make ARCH=arm64 menuconfig
```

执行结果如图所示：

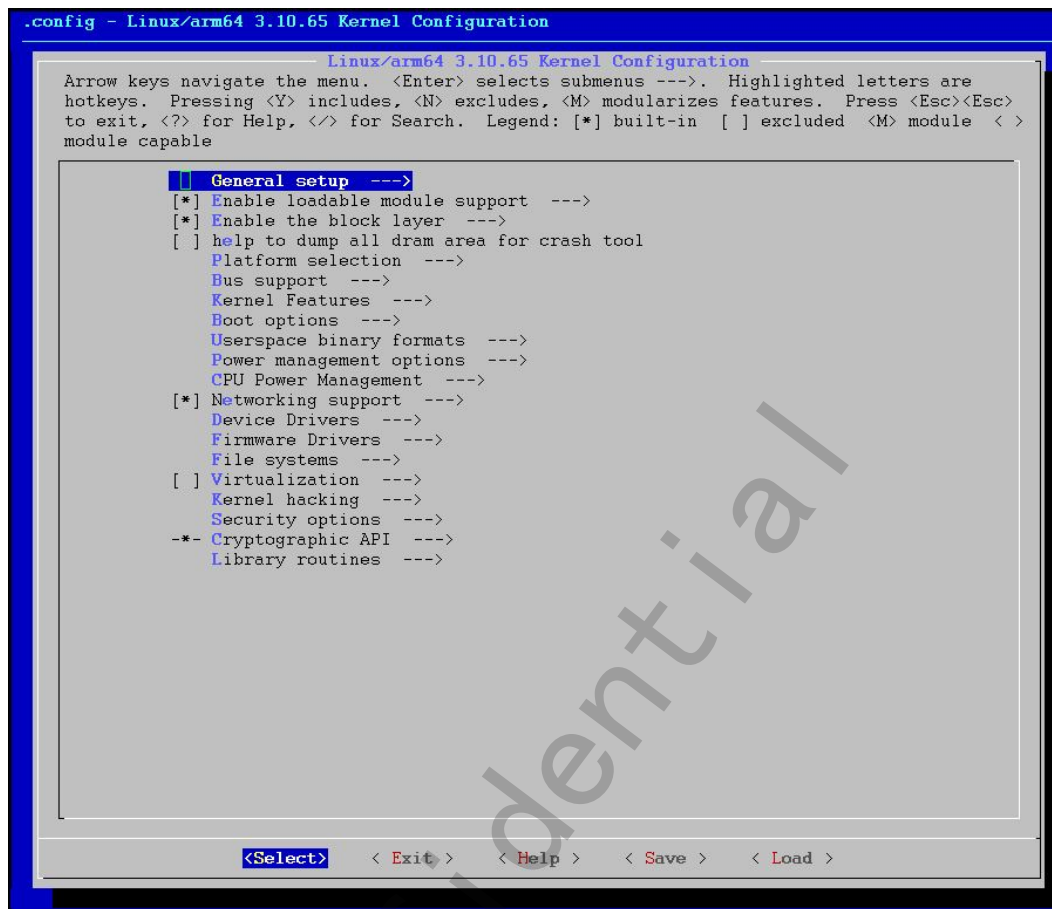


图 6

音频驱动配置:

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

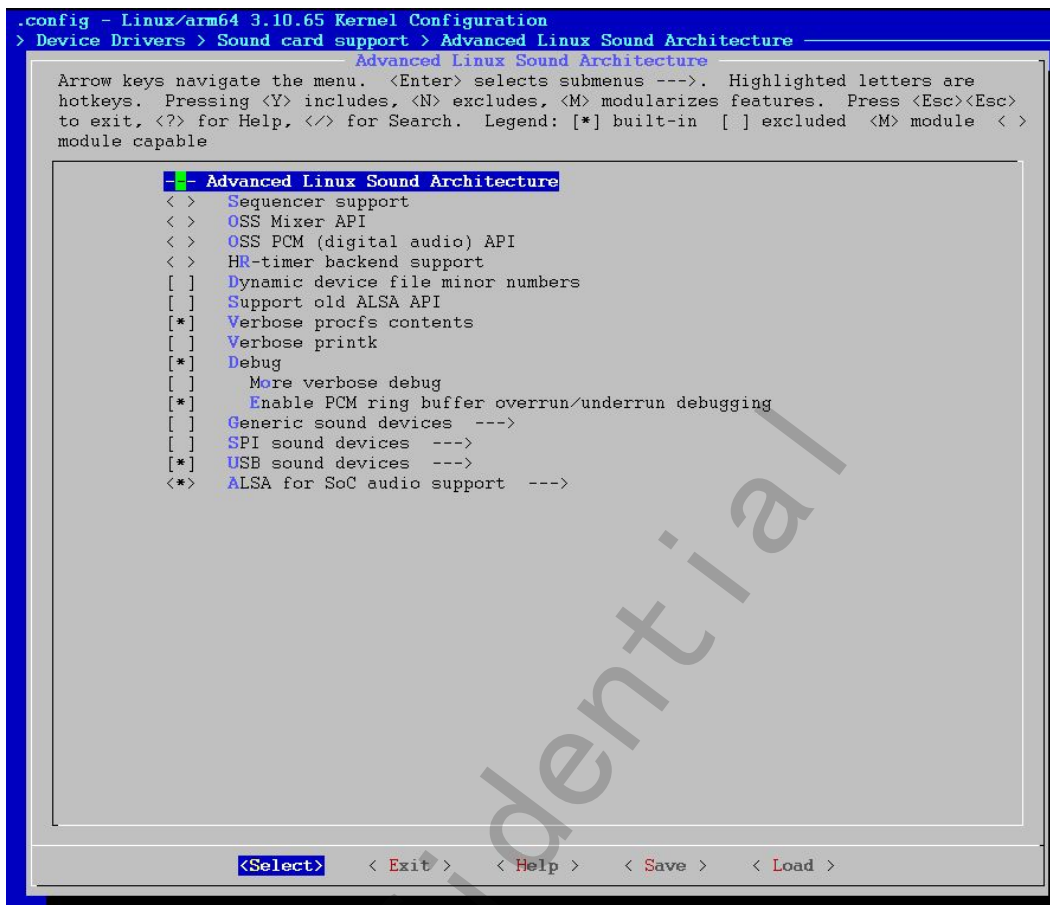


图 7

- <*> ALSA for SoC audio support -->
- <*> ASoC support for SUNXI -->

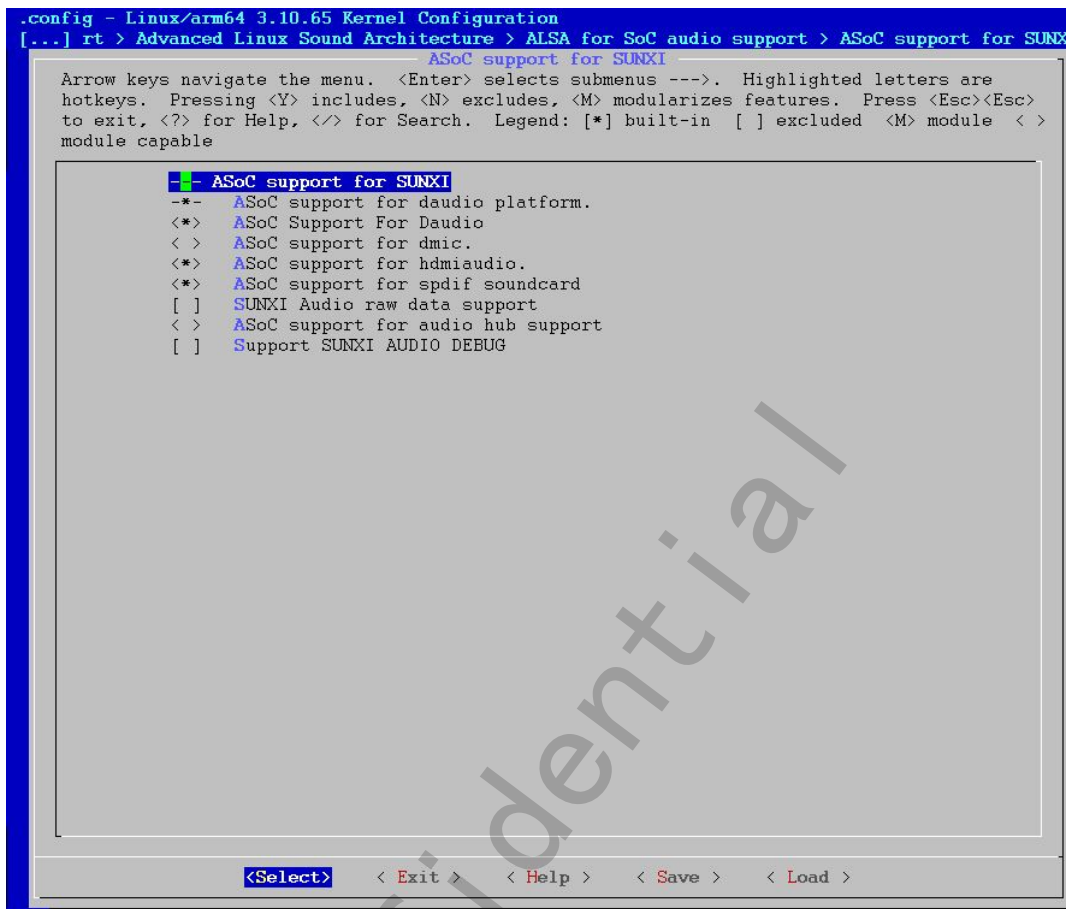


图 8

H6 所有音频模块都直接编入内核。

4.3 sys_config 配置

配置文件的位置：lichee/tools/pack/chips/sun50iw6p1/configs/petrel-p1/sys_config.fex

4.3.1 DMIC 驱动加载

DMIC 配置项如下所示：

- dmic_used: 是否开启 DMIC 硬件，1: 开启，0: 不开启

- **snddmic_used**: 是否注册 DMIC 声卡, 1: 开启, 0: 不开启

```
-----  
; NOTE :Make sure dmic_used = 0x1,  
; snddmic_used = 0x1, if register the sound card dmic.  
-----  
[dmic]  
dmic_used = 0  
[snddmic]  
snddmic_used = 0
```

在 menuconfig 中 DMIC 配置成 Y 后, **dmic_used** 和 **snddmic_used** 配置成 1 便可使用。

4.3.2 SPDIF 驱动加载

SPDIF 配置项如下所示:

- **spdif_used**: 是否开启 SPDIF 硬件, 1: 开启, 0: 不开启
- **sndspdif_used**: 是否注册 SPDIF 声卡, 1: 开启, 0: 不开启

```
-----  
; NOTE :Make sure spdif_used = 0x1,  
; spdifmach_used = 0x1, if register the sound card spdif.  
-----  
[spdif]  
spdif_used = 1  
[sndspdif]  
sndspdif_used = 1
```

4.3.3 HDMI 驱动加载

HDMI 配置项如下所示:

- **audiohdm_i_used**: 是否开启 HDMI 硬件, 1: 开启, 0: 不开启

- sndhdm_i_used: 是否注册 HDMI 声卡, 1: 开启, 0: 不开启
;-----
; NOTE :Make sure audiohdm_i_used = 0x1,
; sndhdm_i_used = 0x1, if register the sound card hdm_i. Daudio1 used for HDMI
;-----
[audiohdm_i]
audiohdm_i_used = 1
[sndhdm_i]
sndhdm_i_used = 1

4.3.4 CODEC 驱动加载

CODEC 挂载在 daudio3 上, daudio 配置项如下所示:

```
;-----  
; NOTE :Make sure snddaudio3_used = 0x1,  
; daudio3_used = 0x1, if register the Daudio3 used for ACX00 codec card  
;-----  
[snddaudio3]  
snddaudio3_used = 1  
;-----  
[daudio3]  
pcm_lrck_period = 0x20  
slot_width_select = 0x20  
daudio_master = 0x04  
audio_format = 0x01  
signal_inversion = 0x01  
frametype = 0x00  
tdm_config = 0x01  
mclk_div = 0x01  
daudio3_used = 1
```

pcm_lrck_period	it is used to program the number of BCLKs per channel of sample frame. This value is interpreted as follow
	<p>1. PCM mode: Number of BCLKs within(Left + Right)channel width 注意在pcm模式下, pcm_lrck_period代表左和右声道相加, 2个声道的大小;</p> <p>2. I2S/Left-Justified/Right-Justified mode: Number of BCLKs within each individual channel width(Left or Right), pcm_lrck_period代表左或者右声道, 一个声道的大小;</p> <p>3. 在i2s模式下, 一个lrck的宽度: 2×32。假如fs=48k, 那么需要的bclk是$3.072M = 2 \times 32 \times 48k$; $bclk_div = 24.576M / 3.072M = 8$;</p> <p>4. 在pcm模式下, 一个lrck的宽度就是32。假如fs=8k, 那么需要的bclk是: $32 \times 8k = 256k$</p>
	并需要确保: i2s模式下: $channel \times slot_width_select \leq 2 \times pcm_lrck_period$; pcm模式下: $channel \times slot_width_select \leq pcm_lrck_period$;
slot_width_select	16bits/20bits/24bits/32bits 数据word的宽度。slot_width_select必须大于等于采样精度, 要不然会出现数据丢失。相当于slot_width_select装载sample_resolution传输, 所以slot_width_select > sample_resolution;这里slot_width_select对i2s模式, pcm模式都有效。
daudio_master	<p>1 SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master) 表示codec做master, i2s做slave</p> <p>2 SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master)</p> <p>3 SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave)</p> <p>4 SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave) use 表示codec做slave, i2s做master</p>
audio_format	<p>1 SND_SOC_DAIFMT_I2S(standard i2s format). use 表示标准i2s格式</p> <p>2 SND_SOC_DAIFMT_RIGHT_J(right justified format). 表示右对齐格式</p> <p>3 SND_SOC_DAIFMT_LEFT_J(left justified format) 表示左对齐格式</p> <p>4 SND_SOC_DAIFMT_DSP_A 短帧模式 并设置 frametype 为0. 短帧</p> <p>5 SND_SOC_DAIFMT_DSP_B 长帧模式 并设置frametype 为1. 长帧</p>
signal_inversion	<p>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</p> <p>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</p> <p>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</p> <p>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</p> <p>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。参考上面的标准i2s时序图。</p>
frametype	0: short frame = 1 clock width; 1: long frame = 2clock width 长帧还是短帧。2个bclk代表长帧 (SND_SOC_DAIFMT_DSP_B模式), 1个bclk代表短帧 (SND_SOC_DAIFMT_DSP_A模式)。用于pcm模式的配置中, i2s无效
tdm_config	0 is pcm. 1 is i2s 0配置成pcm模式; 1配置成i2s模式
daudio_used	daudio_used==1使用daudio; daudio_used ==0不使用daudio;

图 9

4.3.5 蓝牙 SCO 驱动挂载

蓝牙 SCO 设备挂在在 daudio2, 配置如下:

```
;-----  
; NOTE :Make sure snddaudio2_used = 0x1,  
; daudio2_used = 0x1, if register the sound card DAUDIO2.  
;-----  
[snddaudio2]  
snddaudio2_used = 1  
;-----  
[daudio2]  
pcm_lrck_period = 0x40  
slot_width_select = 0x20  
daudio_master = 0x04  
audio_format = 0x04  
signal_inversion = 0x03  
frametype = 0x00  
tdm_config = 0x01  
mclk_div = 0x01  
daudio2_used = 1
```

4.4 AC200 通路配置说明

H6AC200 codec 驱动中所有通路的切换配置都需要通过操作控件来完成。下面我们以 linux 平台举例说明每一种场景下通路配置。下面列出 linux 环境下通过 tinymix (tinyalsa 提供) 列出的驱动加载的控件:


```
Mixer name: 'sndacx00-codec'
Number of controls: 49
ctl      type    num    name                                     value
0        INT     2      I2S Mixer ADC Volume                   0 0
1        INT     1      I2S Mixer DAC Volume                   0
2        INT     2      DAC Mixer ADC Volume                   0 0
3        INT     2      DAC Mixer DAC Volume                   0 0
4        INT     1      Line Out Mixer Volume                   3
5        INT     2      MIC Out Mixer Volume                   3 3
6        INT     1      ADC Input Volume                       3
7        INT     1      LINEOUT Volume                         25
8        INT     1      MIC1 Boost Volume                      4
9        INT     1      MIC2 Boost Volume                      4
10       ENUM     1      sunxi daudio audio hub mode            hub_disable
11       ENUM     1      Right LINEOUT Mux                      Right OMixer
12       ENUM     1      Left LINEOUT Mux                       Left OMixer
13       BOOL     1      Right I2S Mixer I2SDACR Switch          off
14       BOOL     1      Right I2S Mixer ADCR Switch             off
15       BOOL     1      Left I2S Mixer I2SDACL Switch           off
16       BOOL     1      Left I2S Mixer ADCL Switch              off
17       BOOL     1      Right DAC Mixer I2SDACR Switch          on
18       BOOL     1      Right DAC Mixer ADCR Switch            off
19       BOOL     1      Left DAC Mixer I2SDACL Switch           on
20       BOOL     1      Left DAC Mixer ADCL Switch              off
21       BOOL     1      Right Input Mixer MIC1 Switch           off
22       BOOL     1      Right Input Mixer MIC2 Switch           off
23       BOOL     1      Right Input Mixer PhonePN Switch        off
24       BOOL     1      Right Input Mixer PhoneP Switch         off
25       BOOL     1      Right Input Mixer LINEINR Switch        off
26       BOOL     1      Right Input Mixer OMixerR Switch        off
27       BOOL     1      Right Input Mixer OMixerL Switch        off
28       BOOL     1      Left Input Mixer MIC1 Switch            off
29       BOOL     1      Left Input Mixer MIC2 Switch            off
30       BOOL     1      Left Input Mixer PhonePN Switch         off
31       BOOL     1      Left Input Mixer PhoneN Switch          off
32       BOOL     1      Left Input Mixer LINEINL Switch         off
33       BOOL     1      Left Input Mixer OMixerL Switch         off
34       BOOL     1      Left Input Mixer OMixerR Switch         off
35       BOOL     1      Right Output Mixer MIC1 Switch          off
36       BOOL     1      Right Output Mixer MIC2 Switch          off
37       BOOL     1      Right Output Mixer PhonePN Switch       off
38       BOOL     1      Right Output Mixer PhoneP Switch        off
39       BOOL     1      Right Output Mixer LINEINR Switch       off
40       BOOL     1      Right Output Mixer DACR Switch          on
41       BOOL     1      Right Output Mixer DACL Switch          off
42       BOOL     1      Left Output Mixer MIC1 Switch           off
43       BOOL     1      Left Output Mixer MIC2 Switch           off
44       BOOL     1      Left Output Mixer PhonePN Switch        off
45       BOOL     1      Left Output Mixer PhoneN Switch         off
46       BOOL     1      Left Output Mixer LINEINL Switch        off
47       BOOL     1      Left Output Mixer DACL Switch           on
48       BOOL     1      Left Output Mixer DACR Switch           off
```

图 10

4.4.1 系统音频播放场景

4.4.1.1 系统 LINEOUT 输出

ctrl 配置：

下列配置默认以左对左，右对右的方式进行配置，左右相关可以自行调节。

number	ctl_name	value
1	LINEOUT Volume	0-31(0 为 mute)
2	Left DAC Mixer I2SDACL Switch	1
3	Right DAC Mixer I2SDACR Switch	1
4	Left Output Mixer DACL Switch	1
5	Right Output Mixer DACR Switch	1
6	Left LINEOUT Mux	Left OMixer
7	Right LINEOUT Mux	Right OMixer

4.4.1.2 Linein 输入到系统

number	ctl_name	value
1	Left Input Mixer LINEINL Switch	1
2	Right Input Mixer LINEINR Switch	1
3	Left I2S Mixer ADCL Switch	1
4	Right I2S Mixer ADCR Switch	1

4.4.1.3 Mic1 输入到系统

number	ctl_name	value
1	Left Input Mixer MIC1 Switch	1
2	Right Input Mixer MIC1 Switch	1
3	Left I2S Mixer ADCL Switch	1
4	Right I2S Mixer ADCR Switch	1

4.4.1.4 Mic2 输入到系统

number	ctl_name	value
1	Left Input Mixer MIC2 Switch	1

number	ctl_name	value
2	Right Input Mixer MIC2 Switch	1
3	Left I2S Mixer ADCL Switch	1
4	Right I2S Mixer ADCR Switch	1

4.4.1.5 Mic1 + Mic2 立体声输入到系统

- Mic1 左声道, Mic2 右声道

number	ctl_name	value
1	Left Input Mixer MIC1 Switch	1
2	Right Input Mixer MIC2 Switch	1
3	Left I2S Mixer ADCL Switch	1
4	Right I2S Mixer ADCR Switch	1

- Mic2 左声道, Mic1 右声道

number	ctl_name	value
1	Left Input Mixer MIC2 Switch	1
2	Right Input Mixer MIC1 Switch	1
3	Left I2S Mixer ADCL Switch	1
4	Right I2S Mixer ADCR Switch	1

4.4.1.6 Mic1 直接输出到 LINEOUT

number	ctl_name	value
1	LINEOUT Volume	0-31(0 为 mute)
2	Left Output Mixer MIC1 Switch	1
3	Right Output Mixer MIC1 Switch	1

number	ctl_name	value
4	Left LINEOUT Mux	Left OMixer
5	Right LINEOUT Mux	Right OMixer

Mic2、LINEIN 直接输出到 LINEOUT 类推。

4.5 Audio_hub 配置说明

4.5.1 Audio_hub 音频路径配置

H6Audiohub 也是通过 tinymix 工具配置控件从而影响内部各音频接口的连接情况，下图是 linux 环境下通过 tinymix 工具打印出来的 Audiohub 控件节点：

```

Mixer name: 'sndahub'
Number of controls: 21
ctl      type      num      name                                     value
0         ENUM       1        DAM1chan2 Src Select                   NONE
1         ENUM       1        DAM1chan1 Src Select                   NONE
2         ENUM       1        DAM1chan0 Src Select                   NONE
3         ENUM       1        DAM0chan2 Src Select                   NONE
4         ENUM       1        DAM0chan1 Src Select                   NONE
5         ENUM       1        DAM0chan0 Src Select                   NONE
6         ENUM       1        I2S3 Src Select                        NONE
7         ENUM       1        I2S2 Src Select                        NONE
8         ENUM       1        I2S1 Src Select                        NONE
9         ENUM       1        I2S0 Src Select                        NONE
10        ENUM       1        APBIF2 Src Select                      NONE
11        ENUM       1        APBIF1 Src Select                      NONE
12        ENUM       1        APBIF0 Src Select                      NONE
13        BOOL       1        I2S0IN Switch                          off
14        BOOL       1        I2S0OUT Switch                         off
15        BOOL       1        I2S1IN Switch                          off
16        BOOL       1        I2S1OUT Switch                         off
17        BOOL       1        I2S2IN Switch                          off
18        BOOL       1        I2S2OUT Switch                         off
19        BOOL       1        I2S3IN Switch                          off
20        BOOL       1        I2S3OUT Switch                         off

```

图 11

上图控件可以分成两部分：

- 控件 0-12 用于表示对应的 rxif 所连接的 txif，可以配置的值如下所示

- NONE
- APBIF_TXDIF0
- APBIF_TXDIF1
- APBIF_TXDIF2
- I2S0_TXDIF
- I2S1_TXDIF
- I2S2_TXDIF
- I2S3_TXDIF
- DAM0_TXDIF
- DAM1_TXDIF

- 控件 13-18 为是为了 DAPM 机制所需要而是使用的虚拟控件，是用时需要打开所需要使用 PIN 的 IN/OUT switch

以下通过几个实例说明 Audio_hub 的路径配置

- APB0->I2S0 播放

number	ctl_name	value
9	I2S0 Src Select	APBIF_TXDIF0
14	I2S0OUT Switch	On

- APB0 ->DAM1 Chan 1-> I2S2 播放

number	ctl_name	value
1	DAM1Chan1 Src Select	APBIF_TXDIF0
7	I2S2 Src Select	DAM0_TXDIF
18	I2S2OUT Switch	On

- I2S3->DAM0 Chan1->APB2 & APB2-->DAM0 Chan2->APB2 混音录制

number	ctl_name	value
1	DAM1Chan1 Src Select	I2S3_TXDIF
7	DAM1Chan2 Src Select	APBIF_TXDIF2
18	APBIF2 Src Select	DAM1_TXDIF
19	I2S3IN Switch	On

4.6 Audio_hub 操作流程

Audio_hub 驱动设计框图如下，共设计成五个声卡设备，其中音频路径配置通过 Sndahub 声卡配置，APB0、APB1、APB2 分别设计成 Sndahub 声卡下三个设备，Sndaudio0 和 Sndaudio1 为 Daudio 声卡设备，Sndhdm1 与内部 HDMI 相连，Sndacx00codec 由 I2S 接口与 AC200 组成。

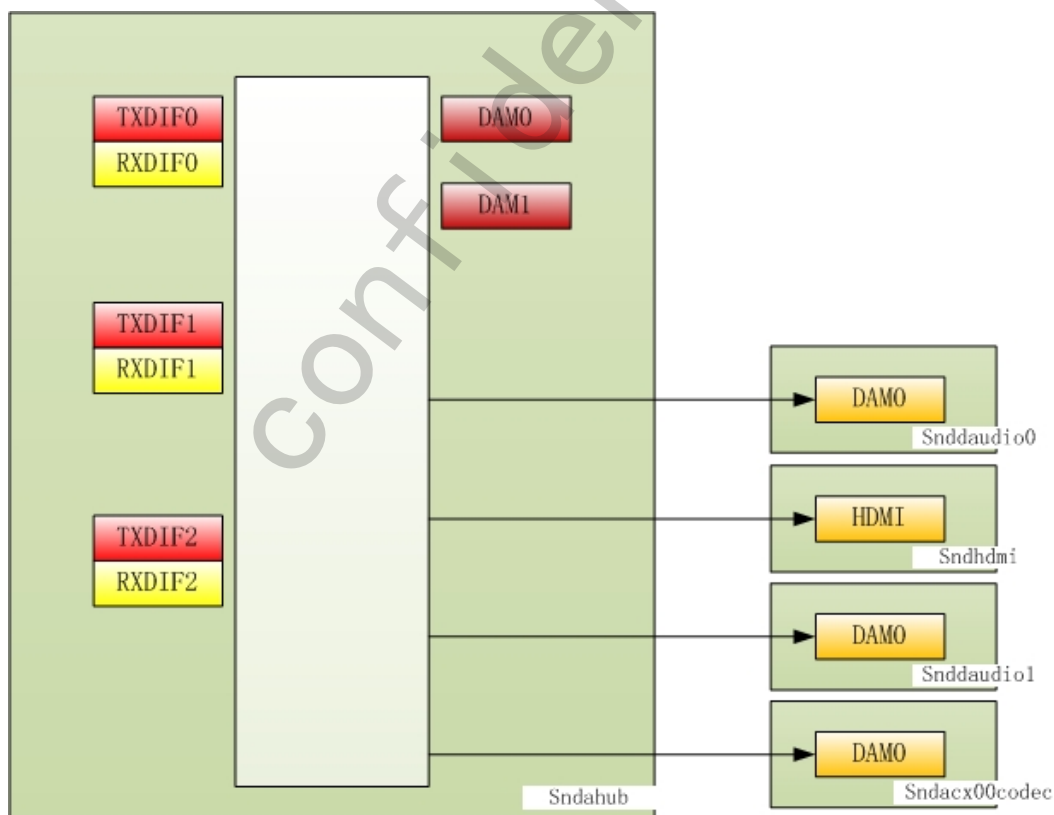


图 12

HDMI 播放操作流程如图所示：

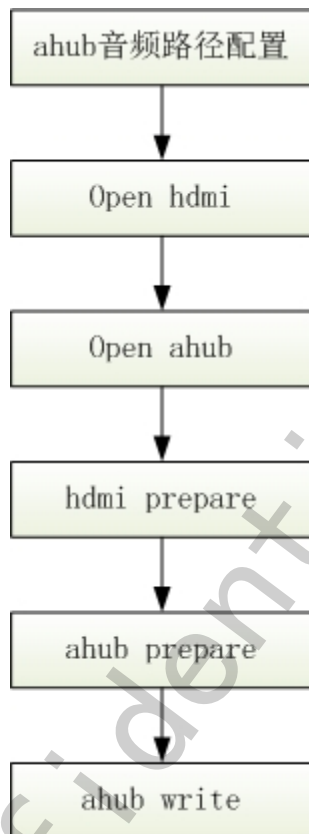


图 13

AC200 播放操作流程如图所示：

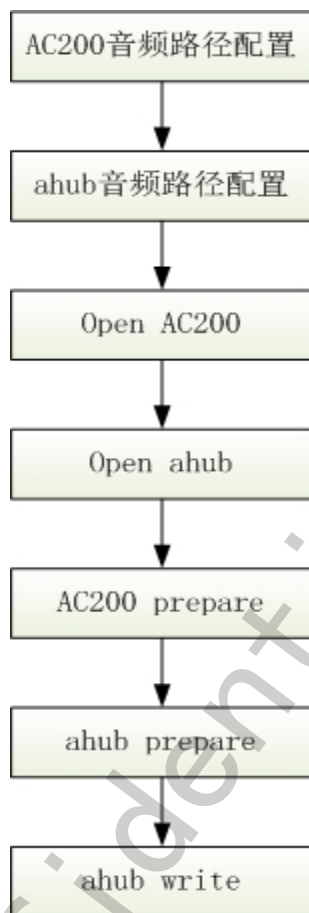


图 14

Daudio 录制流程如图所示:

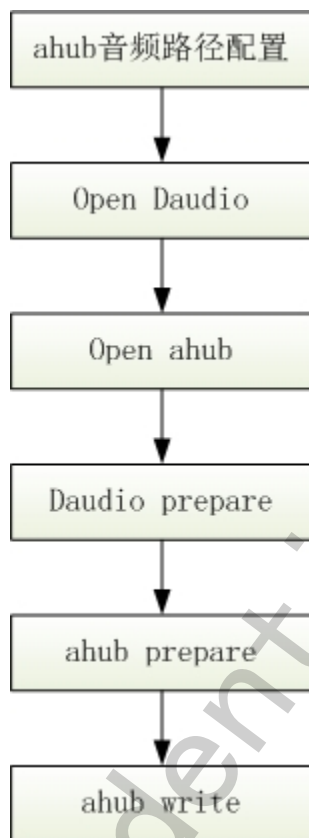


图 15

5. 音频输入输出切换策略

5.1 FVD 音频输出策略

H6 Android4.4 音频输出策略如图所示。

当前状态	音频输出	具体操作	操作后显示状态	音频输出
只接HDMI	HDMI输出 或用户手 动更改	拔出HDMI线	输出为NONE	切换到CODEC
		接入CVBS线	输出为CVBS	CODEC
		接入USB音频输出设备	输出为CVBS	切换到USB
		待机唤醒	输出为CVBS	USB
		关机开机	输出为CVBS	USB
只接CVBS	CODEC输 出或用户 手动更改	拔出CVBS线	输出为NONE	输出为CODEC
		接入HDMI线	输出为HDMI	添加HDMI
		接入USB音频输出设备	输出为HDMI	切换到USB
		待机唤醒	输出为HDMI	USB
		关机开机	输出为HDMI	USB
同时接入 HDMI和 CVBS	默认HDMI 和CVBS或 用户手动 更改	拔出CVBS线	输出为HDMI	HDMI和CODEC
		拔出HDMI线	输出为NONE	CODEC
		接入USB音频输出设备	输出为NONE	切换到USB
		待机唤醒	输出为NONE	USB
		关机开机	输出为NONE	USB
无接入任 何设备	默认 CODEC输 出	接入HDMI线	输出为HDMI	添加HDMI
		接入CVBS线	添加CVBS	CODEC
		接入USB音频输出设备	输出CVBS、HDMI	切换到USB
		待机唤醒	输出CVBS、HDMI	USB
		关机开机	输出CVBS、HDMI	USB
USB音频 输出	仅USB音 频输出	插拔HDMI、CVBS，待 机唤醒，关机开机	相应输出	输出为USB
		拔出USB音频设备	不变	切换到当前显 示对应音频输 出
		手动选择其他音频	不变	切换对应输出
开启透传 模式	SPDIF透 传	插拔HDMI、CVBS，待 机唤醒，关机开机	相应输出	输出为SPDIF
	HDMI透传	插拔HDMI、CVBS，待 机唤醒，关机开机	不变	输出为HDMI
		接入USB音频输出设备	不变	切换USB

图 16

H6 AndroidN 音频输出策略如图所示。

当前状态	音频输出	具体操作	操作后显示状态	音频输出
只接HDMI	HDMI输出 或用户手 动更改	拔出HDMI线	输出为NONE	切换到CODEC
		接入CVBS线	输出为CVBS	CODEC
		接入USB音频输出设备	输出为CVBS	选择后切换到USB
		待机唤醒	输出为CVBS	USB
		关机开机	输出为CVBS	USB
只接CVBS	CODEC输出 或用户手 动更改	拔出CVBS线	输出为NONE	输出为CODEC
		接入HDMI线	输出为HDMI	添加HDMI
		接入USB音频输出设备	输出为HDMI	选择后切换到USB
		待机唤醒	输出为HDMI	USB
		关机开机	输出为HDMI	USB
同时接入 HDMI和 CVBS	默认HDMI 和CVBS或 用户手动 更改	拔出CVBS线	输出为HDMI	HDMI和CODEC
		拔出HDMI线	输出为NONE	CODEC
		接入USB音频输出设备	输出为NONE	选择后切换到USB
		待机唤醒	输出为NONE	USB
		关机开机	输出为NONE	USB
无接入任 何设备	默认 CODEC输出	接入HDMI线	输出为HDMI	添加HDMI
		接入CVBS线	添加CVBS	CODEC
		接入USB音频输出设备	输出CVBS、HDMI	选择后切换到USB
		待机唤醒	输出CVBS、HDMI	USB
		关机开机	输出CVBS、HDMI	USB
USB音频 输出	仅USB音 频输出	插拔HDMI、CVBS，待 机唤醒，关机开机	相应输出	输出为USB
		拔出USB音频设备	不变	切换到当前显示对 应音频输出
		手动选择其他音频	不变	切换对应输出
开启透传 模式	SPDIF透 传	插拔HDMI、CVBS，待 机唤醒，关机开机	相应输出	输出为SPDIF
	HDMI透传	插拔HDMI、CVBS，待 机唤醒，关机开机	不变	输出为HDMI
		接入USB音频输出设备	不变	选择后切换到USB

图 17

5.2 音频输入策略

H6 音频输入策略如图所示：

当前设备	音频输入	具体操作	音频输入
CODEC	内置MIC或LINE IN	接入USB音频输入设备	切换到USB音频输入
USB音频设备	USB音频输入	拔出USB音频设备	切换到CODEC音频输入

图 18

confidential

6. TV_BOX 音频特性

TV_BOX 的音频系统具有以下特性：

- 支持单/多路音频输出
- 支持 HDMI USB 音频设备的热插拔
- 支持音频透传
- AndroidN 支持主音量

6.1 单/多通路音频输出的使用

打开设置 -> 声音 -> 音频输出模式，显示如图所示：

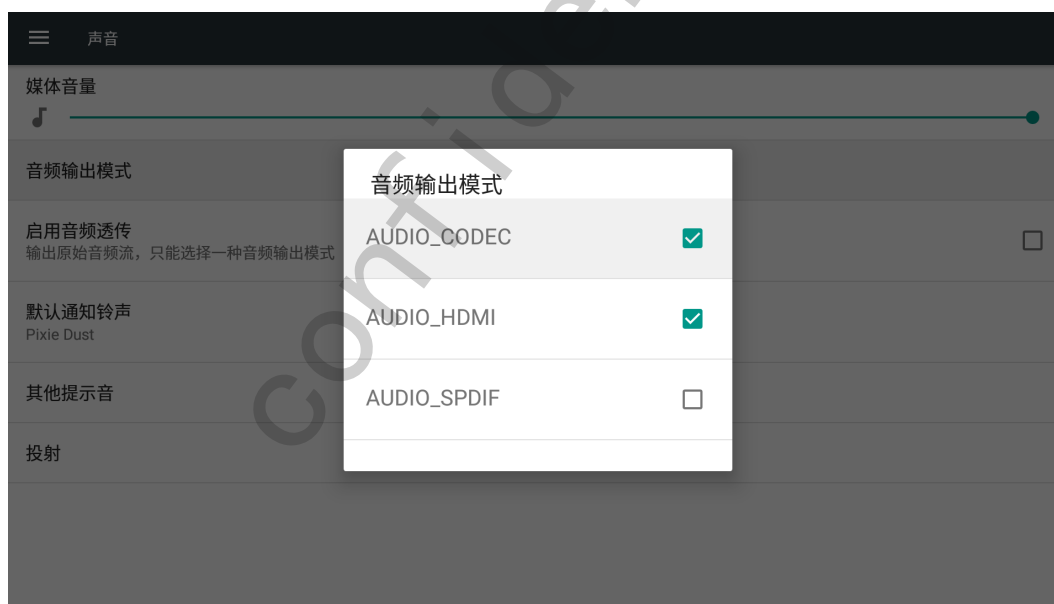


图 19

- AUDIO_CODEC 对应 CVBS，电视信号 AV
- AUDIO_HDMI 对应 HDMI，电视信号 HDMI
- AUDIO_SPDIF 对应 SPDIF，光纤接口，外接音箱

若让声音单路输出，只勾选一个选项，例如控制声音只从 HDMI 输出，只勾选 AUDIO_HDMI。将盒子和电视机用 HDMI 线连接后，电视机信号切换到 HDMI，播放音乐或视频就可听到声音。

若让声音多路输出，可勾选多个选项，例如控制声音从 HDMI 和 CVBS 同时输出，则勾选 AUDIO_CODEC 和 AUDIO_HDMI 选项。将盒子和电视机用 HDMI 和 CVBS 线连接后，播放音乐或视频，分别切换电视机信号到 AV 或者 HDMI，都可以听到声音。

在盒子第一次启动时，默认输出是 AUDIO_CODEC 和 AUDIO_HDMI。用户也可根据需求更改默认输出，例如修改为 AUDIO_CODEC,AUDIO_HDMI,AUDIO_SPDIF。

代码目录在 android/device/softwinner/common/common.mk
PRODUCT_PROPERTY_OVERRIDES +=
audio.output.active=AUDIO_CODEC,AUDIO_HDMI
audio.input.active=AUDIO_CODEC

涉及到的代码目录：

Android4.4

1. Audio 系统设置：

android/vendor/tvd/packages/tvdsettings/src/com/android/settings/AudioChannelsSelect.java

2. Audio 输入输出扩展接口：

android/frameworks/base/media/java/android/media/AudioManager.java

3. HAL 层响应接口：

android/hardware/aw/audio/audio_hw.c

AndroidN

1. Audio 系统设置：

android/packages/apps/settings/src/com/android/settings/notification/AudioChannelsSelect.java

2. Audio 输入输出扩展接口：

android/device/softwinner/common/addons/frameworks/audio/java/AudioManagerEx

3. HAL 层响应接口：

android/hardware/aw/audio/audio_hw.c

代码说明：

- AudioChannelsSelect.java 用来获取音频设备, AudioManager/AudioManagerEx::getAudioDevices, 做出勾选选项后的响应, AudioManager/AudioManager::setAudioDeviceActive
- AudioManager/AudioManagerEx 用来向 HAL 层设置当前被选中的 active 音频输出通路, 并保存音频输出通路到系统数据库中
- audio_hw.c 真正设置音频输出通路

6.2 音频设备热插拔

1. HDMI 线热插拔, 将 HDMI 线和 CVBS 线与电视机连接

- 拔出 HDMI 线, 设置 -> 声音 -> 音频输出模式 AUDIO_HDMI 未选中, 音频数据不可从 HDMI 输出
- 连接 HDMI 线, 设置 -> 声音 -> 音频输出模式 AUDIO_HDMI 选中, 音频数据可从 HDMI 输出

2. USB 音频输入设备热插拔, 将 HDMI 线和 CVBS 线与电视机连接

- 连接 USB 音频输入设备 (一般为 USB 带 MIC 摄像头), 音频输入设备切换到此设备
- 拔出 USB 音频输入设备, 音频输入设备切回到 CODEC, 默认的音频输入设备是 CODEC

3. USB 音频输出设备热插拔, 将 HDMI 线和 CVBS 线与电视机连接

- Android 4.4
 - 连接 USB 音频输出设备 (一般为 USB 音箱), 音频输出直接切换到 USB 音箱
 - 拔出 USB 音箱, 音频输出设备恢复之前音频输出设备
- Android N

- 连接 USB 音频输出设备（一般为 USB 音箱），弹出对话框如图所示，显示是否选中 USB 作为输出设备
- 若选中，音频输出模式仅为 USB 音频输出设备一路，音频数据只可从 USB 音频设备输出
- 若未选中，音频输出模式不变
- 断开 USB 音频输出设备，音频输出模式切换到 HDMI

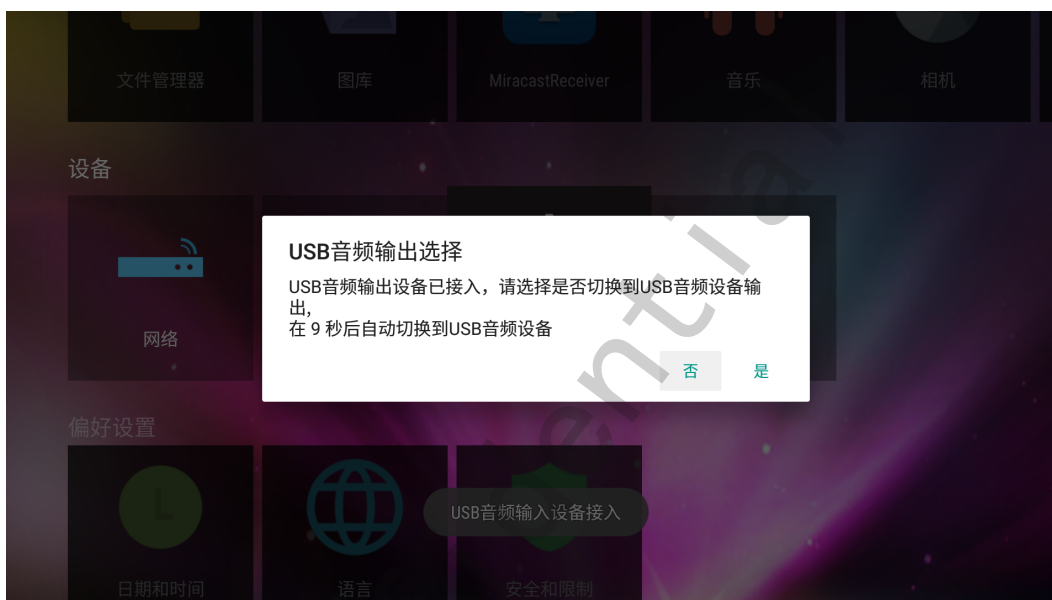


图 20

涉及到的代码目录：

Android4.4

1. Audio 设备插拔事件广播管理

android/frameworks/base/services/java/com/android/server/AudioDeviceManagerObserver.java

2. Audio 切换策略

android/frameworks/base/services/java/com/android/server/AudioManagerPolicy.java

3. Audio 输入输出扩展接口：

android/frameworks/base/media/java/android/media/AudioManager.java

4. HAL 层响应接口：

android/hardware/aw/audio/audio_hw.c

AndroidN

1. Audio 设备插拔事件广播管理

android/device/softwinner/common/addons/framework/audio/java/AudioDeviceManagerObserver.java

2. Audio 切换策略

android/device/softwinner/common/addons/framework/audio/java/AudioManagerPolicy.java

3. Audio 输入输出扩展接口：

android/device/softwinner/common/addons/frameworks/audio/java/AudioManagerEx

4. HAL 层响应接口：

android/hardware/aw/audio/audio_hw.c

代码说明：

- **AudioDeviceManagerObserver.java** 用于监听音频设备的热插拔，有音频设备热插拔时驱动会发送广播，**AudioDeviceManagerObserver.java** 监听这个广播
- **AudioManagerPolicy.java** 用于制定策略，切换音频输入输出通道
- **AudioManager/AudioManagerEx** 用来向 HAL 层设置 active 音频输入输出通路，并保存音频输出通路到系统数据库中
- **audio_hw.c** 真正设置音频输入输出通路

6.3 音频透传的使用

打开设置 -> 声音 -> 启用音频透传，勾选音频输出模式，如图所示。

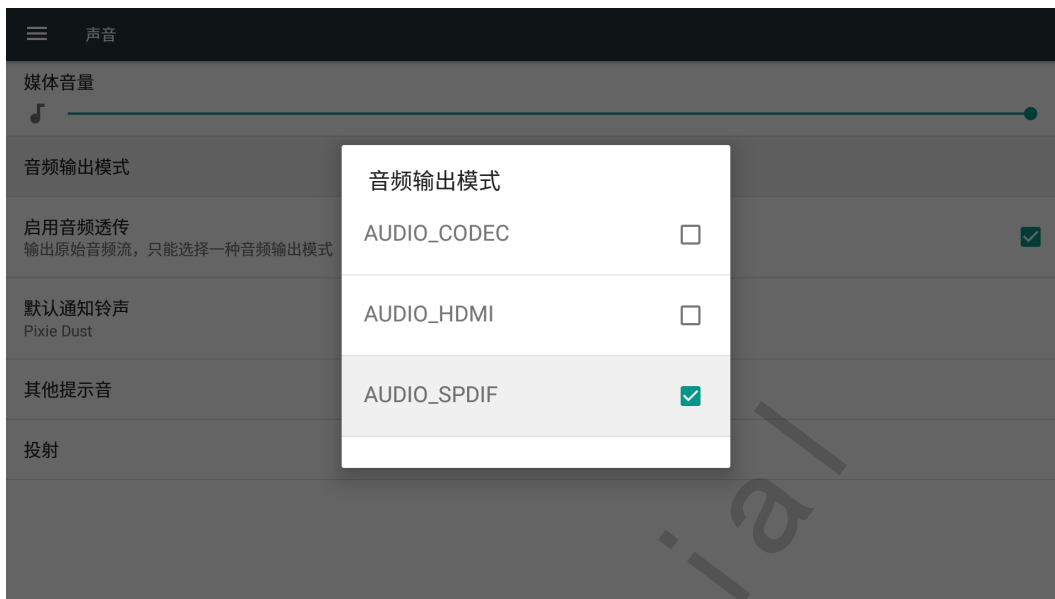


图 21

说明：

- 透传模式下，音频输出模式只能勾选一种，即单路输出
- 透传模式下，只有 HDMI 和 SPDIF 才可实现透传
- 透传模式下，选择 CODEC 或 USB，无透传效果，但不影响正常音频输出
- 透传模式下，无法使用遥控器控制音量和静音

涉及到的代码目录：

Android4.4

1. Audio 系统设置：

android/vendor/tvd/packages/tvdsettings/src/com/android/settings/SoundSettings.java

android/vendor/tvd/packages/tvdsettings/src/com/android/settings/AudioChannelsSelect.java

2. Audio 输入输出扩展接口：

android/frameworks/base/media/java/android/media/AudioManager.java

3. HAL 层响应接口：

android/hardware/aw/audio/audio_hw.c

AndroidN

1. Audio 系统设置：

android/packages/apps/settings/src/com/android/settings/notification/SoundSettings.java

android/packages/apps/settings/src/com/android/settings/notification/AudioChannelsSelect.java

2. Audio 输入输出扩展接口：

android/device/softwinner/common/addons/frameworks/audio/java/AudioManagerEx

3. HAL 层响应接口：

android/hardware/aw/audio/audio_hw.c

代码说明：

- SoundSettings.java 用来设置数据库中的音频透传全局变量
- AudioChannelsSelect.java 做出勾选音频输出模式选项后的响应，AudioManager/AudioManager::setAudioDeviceActive
- AudioManager/AudioManagerEx::setAudioDeviceActive 获取数据库透传全局变量并设置透传属性，MediaPlayer.setRawDataMode/SystemProperties.set，多媒体根据透传属性选择是否解码音频数据
- audio_hw.c 设置音频通路，直接将未解码的数据按照 PCM 双声道格式存储输出

6.4 硬件混音的使用

H6 支持硬件混音，在 HAL 层集成了混音流程，混音示意图如图所示，应用场景一般是卡拉 OK。

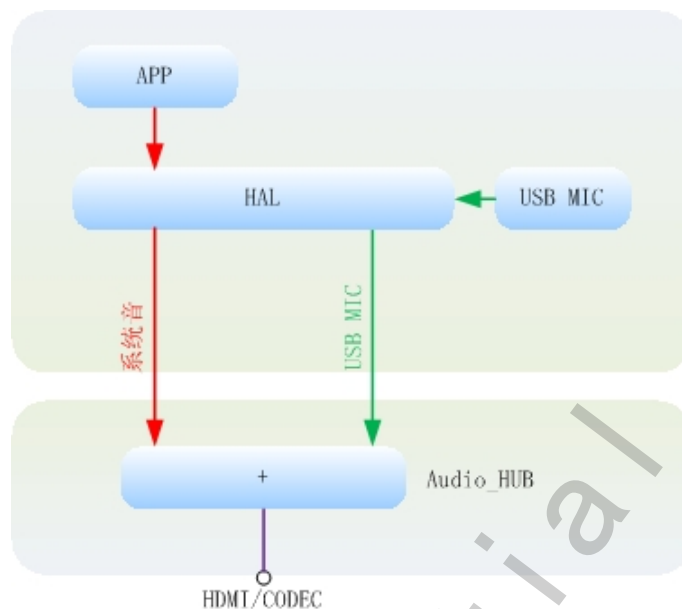


图 22

APP 将要播放的系统数据送入 HAL 层，HAL 层将 MIC 数据和系统数据送入 Audio_HUB。Audio_HUB 是 H6 的混音模块，Audio_HUB 将混音后的数据送入 HDMI 或 CVBS。

硬件混音不是公共功能，需要 HAL 层与应用对接。通常上层应用使用 `AudioManager::setParameters` 传递信息，HAL 层接收后打开混音流程。硬件混音的优点是延时少、音质好。

6.5 tinypalsa 命令的使用

android/external/tinypalsa 目录下编译，生成以下四个命令，调试音频通路常常使用这四个命令：

- `tinypcminfo` 用于查看声卡设备参数范围
- `tinypplay` 用于直接调用 `tinypalsa` 接口，不走 HAL 层，播放音频
- `tinypcap` 用于直接调用 `tinypalsa` 接口，不走 HAL 层，录音
- `tinypmix` 用于查看声卡设备节点的通路

生成的四个命令均在 `/android/out/target/product/版本号/system/bin` 下，以 `tinymix` 举例，安装命令方法：

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

使用方法：

`tinypcminfo`:

- `tinypcminfo -D card -d device`

例如查看 CODEC 参数范围，执行 `tinypcminfo -D 2 -d 0`，执行结果如图所示：

```
PCM out:
  Rate:   min=8000Hz   max=192000Hz
 Channels: min=1       max=2
Sample bits: min=16    max=32
Period size: min=32    max=131072
Period count: min=1    max=8

PCM in:
  Rate:   min=8000Hz   max=192000Hz
 Channels: min=1       max=2
Sample bits: min=16    max=32
Period size: min=32    max=131072
Period count: min=1    max=8
root@petrel-p1:/ #
```

图 23

`tinypplay`:

- `tinypplay file.wav -D card -d device`

例如用 HDMI 播放 `/sdcard/music.wav`，执行 `tinypplay /sdcard/music.wav -D 0 -d 0`

`tinycap`:

- `tinycap file.wav -D card -d device -c channels -r rate -b bits`

例如用 CODEC 录一段双声道、48K 采样率、16 位的 wav 音频文件，可执行 `tinycap /sdcard/music.wav -D 0 -d 0 -c 2 -r 48000 -b 16`

`tinymix`:

- `tinymix -D card`
- `tinymix -D card ctl value`

例如查看 CODEC 通路，执行 `tinymix -D 0` 或者 `tinymix`，`tinymix` 默认是 0，执行结果如图所示：

具体 card 节点号，可执行命令 `cat proc/asound/cards` 查看。若修改通路值，执行 `tinymix`，执行结果如图所示：

ctl	type	num	name	value
0	INT	1	digital volume	0
1	INT	1	Lineout volume	31
2	INT	1	MIC1_G boost stage output mixer control	3
3	INT	1	MIC2_G boost stage output mixer control	3
4	INT	1	MIC1 boost AMP gain control	4
5	INT	1	MIC2 boost AMP gain control	4
6	INT	1	LINEINL/R to L_R output mixer gain	3
7	INT	1	ADC input gain control	3
8	ENUM	1	codec hub mode	hub_disable
9	BOOL	1	RADC input Mixer r_output mixer Switch	Off
10	BOOL	1	RADC input Mixer l_output mixer Switch	Off
11	BOOL	1	RADC input Mixer LINEINR Switch	Off
12	BOOL	1	RADC input Mixer MIC1 boost Switch	On
13	BOOL	1	RADC input Mixer MIC2 boost Switch	Off
14	BOOL	1	LADC input Mixer r_output mixer Switch	Off
15	BOOL	1	LADC input Mixer l_output mixer Switch	Off
16	BOOL	1	LADC input Mixer LINEINL Switch	Off
17	BOOL	1	LADC input Mixer MIC1 boost Switch	On
18	BOOL	1	LADC input Mixer MIC2 boost Switch	Off
19	ENUM	1	Lineout_L Mux	MIXER_L Switch
20	ENUM	1	Lineout_R Mux	MIXER_R Switch
21	BOOL	1	Right Output Mixer DACL Switch	Off
22	BOOL	1	Right Output Mixer DACR Switch	Off
23	BOOL	1	Right Output Mixer LINEINR Switch	Off
24	BOOL	1	Right Output Mixer MIC1Booststage Switch	Off
25	BOOL	1	Right Output Mixer MIC2Booststage Switch	Off
26	BOOL	1	Left Output Mixer DACR Switch	Off
27	BOOL	1	Left Output Mixer DACL Switch	Off
28	BOOL	1	Left Output Mixer LINEINL Switch	Off
29	BOOL	1	Left Output Mixer MIC1Booststage Switch	Off
30	BOOL	1	Left Output Mixer MIC2Booststage Switch	Off
31	BOOL	1	External Speaker Switch	On

图 24

```

root@petrel-p1:/proc/asound # tinymix -D 2 4 0
tinymix -D 2 4 0
root@petrel-p1:/proc/asound # tinymix -D 2
tinymix -D 2
Mixer name: 'sndacx00-codec'
Number of controls: 49

```

ctl	type	num	name	value
0	INT	2	I2S Mixer ADC Volume	0 0
1	INT	1	I2S Mixer DAC Volume	0
2	INT	2	DAC Mixer ADC Volume	0 0
3	INT	2	DAC Mixer DAC Volume	0 0
4	INT	1	Line Out Mixer Volume	0
5	INT	2	MIC Out Mixer Volume	3 3
6	INT	1	ADC Input Volume	3
7	INT	1	LINEOUT Volume	25
8	INT	1	MIC1 Boost Volume	4
9	INT	1	MIC2 Boost Volume	4
10	ENUM	1	sunxi daudio audio hub mode	hub_disable
11	ENUM	1	Right LINEOUT Mux	Right OMixer
12	ENUM	1	Left LINEOUT Mux	Left OMixer
13	BOOL	1	Right I2S Mixer I2SDACR Switch	Off
14	BOOL	1	Right I2S Mixer ADCR Switch	Off
15	BOOL	1	Left I2S Mixer I2SDACL Switch	Off
16	BOOL	1	Left I2S Mixer ADCL Switch	Off
17	BOOL	1	Right DAC Mixer I2SDACR Switch	On
18	BOOL	1	Right DAC Mixer ADCR Switch	Off
19	BOOL	1	Left DAC Mixer I2SDACL Switch	On
20	BOOL	1	Left DAC Mixer ADCL Switch	Off
21	BOOL	1	Right Input Mixer MIC1 Switch	Off
22	BOOL	1	Right Input Mixer MIC2 Switch	Off
23	BOOL	1	Right Input Mixer PhonePN Switch	Off
24	BOOL	1	Right Input Mixer PhoneP Switch	Off
25	BOOL	1	Right Input Mixer LINEINR Switch	Off
26	BOOL	1	Right Input Mixer OMixerR Switch	Off
27	BOOL	1	Right Input Mixer OMixerL Switch	Off
28	BOOL	1	Left Input Mixer MIC1 Switch	Off
29	BOOL	1	Left Input Mixer MIC2 Switch	Off
30	BOOL	1	Left Input Mixer PhonePN Switch	Off
31	BOOL	1	Left Input Mixer PhoneN Switch	Off
32	BOOL	1	Left Input Mixer LINEINL Switch	Off
33	BOOL	1	Left Input Mixer OMixerL Switch	Off
34	BOOL	1	Left Input Mixer OMixerR Switch	Off
35	BOOL	1	Right Output Mixer MIC1 Switch	Off
36	BOOL	1	Right Output Mixer MIC2 Switch	Off
37	BOOL	1	Right Output Mixer PhonePN Switch	Off
38	BOOL	1	Right Output Mixer PhoneP Switch	Off
39	BOOL	1	Right Output Mixer LINEINR Switch	Off
40	BOOL	1	Right Output Mixer DACR Switch	On
41	BOOL	1	Right Output Mixer DACL Switch	Off
42	BOOL	1	Left Output Mixer MIC1 Switch	Off
43	BOOL	1	Left Output Mixer MIC2 Switch	Off
44	BOOL	1	Left Output Mixer PhonePN Switch	Off
45	BOOL	1	Left Output Mixer PhoneN Switch	Off
46	BOOL	1	Left Output Mixer LINEINL Switch	Off
47	BOOL	1	Left Output Mixer DACL Switch	On
48	BOOL	1	Left Output Mixer DACR Switch	Off

图 25

7. FAQ

1. 透传播放 DTS 格式的音频文件，无声音输出

- 排查问题

1. 查看设置 -> 声音 -> 启动透传有没有被勾选
2. 查看设置 -> 声音 -> 音频输出模式有没有勾选对应的音频输出
3. 如仍有问题请与 FAE 联系

2. 播放音视频，无声音输出

- 排查问题

1. 查看设置 -> 声音 -> 音频输出模式有没有勾选对应的音频输出
2. 查看设备设备节点是否存在，执行命令 `cat /proc/asound/cards`，执行结果如下所示

```
cat proc/asound/cards
0 [sndacx00codec ]: sndacx00-codec - sndacx00-codec
  sndacx00-codec
1 [sndhdmi      ]: sndhdmi - sndhdmi
  sndhdmi
2 [sndspdif     ]: sndspdif - sndspdif
  sndspdif
```

3. 使用 CODEC 录音，回放有电流声

- 排查问题

1. 不接入麦克风，使用 `tinycap` 录音，执行命令 `tinycap /sdcard/music.wav -D 0 -d 0`
2. 使用音频软件 Audition 查看波形，若波形有小毛刺，说明是底噪

4. 用户 apk 播放音频，有抖音

- 排查问题

1. 检查 apk 是否没有按照获取到的 buffer 大小向下送音频数据
5. 蓝牙语音通话无声音或声音错误
 - 排查问题
 1. 查看 daudio0 设备节点是否存在
 2. 查看 menuconfig 配置，查看 daudio 是否已经配置成 y，查看方法详见章节 4.2.1
 3. 查看 sys_config.fex 配置，查看 daudio2 是否配置正确，查看方法详见章节 4.3.5

confidential

8. 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 datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

confidential