

A83T 项目

音频驱动使用说明

文档履历

版本号	日期	制/修订人	内容描述
V1.0	2014-10-16		初始版本

Confidential

目 录

A83T 项目.....	1
音频驱动使用说明.....	1
1. 概述.....	iii
1.1. 编写目的	iii
1.2. 适用范围	iii
1.3. 相关人员	iii
2. 模块介绍.....	iv
2.1. 模块功能介绍	iv
2.1.1. Daudio 功能	iv
2.1.2. Spdif 功能	iv
2.1.3. hdmaudio 功能	iv
2.2. 相关术语介绍	iv
2.3. 音频 config 配置	v
2.4. Sysconfig 配置	viii
3. 使用描述.....	xi
3.1. Daudio	xi
3.1.1. playback 应用	xiii
3.1.2. capture 应用	xiv
3.1.3. 播放录音配置 ctl 说明	xiv
3.2. Hdmi	xvi
3.2.1. playback 应用	xvi
3.3. Spdif	xvi
3.3.1. playback 应用	xvii
4. 出错处理.....	xviii
5. 总结.....	xix

1. 概述

1.1. 编写目的

本文档基于 A83T+AC100 平台展开分析，用于指导音频系统的开发。

1.2. 适用范围

适用于 A83T 平台。

1.3. 相关人员

音频相关开发人员。

Confidential

2. 模块介绍

2.1. 模块功能介绍

2.1.1. Daudio 功能

A83T daudio 硬件上通过 TDM 接口与 AC100 进行数据流传输，A83T 通过 RSB 总线对 AC100 进行配置，软件上采用 alsa-asoc 架构实现，具有以下功能。

- (1) 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);
- (2) 支持 mono 和 stereo 模式;
- (3) 支持同时 playback 和 record(全双工模式);
- (4) 支持 3、4 段耳机插拔检测, hook 键检测, 耳机音量加减键定制。

2.1.2. Spdif 功能

spdif 驱动所具有的功能:

- (1) 支持多种采样率格式(22.05khz, 24khz, 32khz, 44.1khz, 48khz, 88.2khz, 96khz, 176.4khz, 192khz);
- (2) 支持 mono 和 stereo 模式;
- (3) 只支持 playback 模式, 不支持 record 模式。
- (4) 支持 rawdata 模式

2.1.3. hdmaudio 功能

hdmaudio 驱动所具有的功能:

- (1) 支持多种采样率格式(8khz, 11.025khz, 12khz, 16khz, 22.05, 24khz, 32khz, 44.1, 48khz, 88.2khz, 96khz, 176.4khz, 192khz);
- (2) 支持 mono 和 stereo 模式;
- (3) 只支持 playback 模式, 不支持 record 模式。
- (4) 支持 rawdata 模式

2.2. 相关术语介绍

Audio Driver: Acronyms	
Acronym	Definition
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.
Asoc	ALSA System on Chip
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 interleaved	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

2.3. 音频 config 配置

(1) make ARCH=arm menuconfig 选中如图蓝色

```

[*] Patch physical to virtual translations at runtime
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
System Type --->
[ ] FIQ Mode Serial Debugger
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
[*] Networking support --->
Device Drivers --->
File systems --->
Kernel hacking --->
Security options --->
--*- Cryptographic API --->
Library routines --->
---
Load an Alternate Configuration File
Save an Alternate Configuration File

```

(2) 选中下图

```

Multifunction device drivers --->
[*] Voltage and Current Regulator Support --->
[*] Pulse-Width Modulation (PWM) Support --->
<*> Multimedia support --->
    Graphics support --->
    <*> Sound card support --->
    [*] HID Devices --->
    [*] USB support --->
    <*> MMC/SD/SDIO card support --->
    < > Sony MemoryStick card support (EXPERIMENTAL) --->
    [ ] LED Support --->
    < > Switch class support --->
    [ ] Accessibility support --->
    [*] Real Time Clock --->
    [*] DMA Engine support --->
    [ ] Auxiliary Display support --->
    < > Userspace I/O drivers --->

```

(3) 选中下图

```

-----
--- Sound card support
<*> Advanced Linux Sound Architecture --->
< > Open Sound System (DEPRECATED) --->

```

(4) 选中下图

```

< > HRTimer backend support
[ ] Dynamic device file minor numbers
[*] Support old ALSA API
[*] Verbose procfs contents
[ ] Verbose printk
[ ] Debug
[*] Generic sound devices --->
[*] ARM sound devices --->
[*] SPI sound devices --->
[*] USB sound devices --->
<*> ALSA for SoC audio support --->

```

(5) 选中下图

```
- ALSA for SoC audio support
<*>   SoC daudio0 tdm interface for SUNXI chips
< >   Public Machine for SUNXI chips
<*>   Machine for A83 AC100
<*>   virtual bb interface
< >   HDMI Audio for SUNXI chips
< >   HDMIPCM for the SUN9I SUN8IW6 chips
< >   sunxi On-Chip spdif
< >   SUNXI AudioCodec AC100
<*>   SUNXI AudioCodec AC100 DAPM
```

(6) 上图中只加载了 audiocodec 驱动，包含四部分：

SoC daudio0 tdm interface for SUNXI chips

Machine for A83 AC100

```
virtual bb interface
```

SUNXI AudioCodec AC100 DAPM

如需加载 hdmi 驱动, 需要选择:

HDMI Audio for SUNXI chips

HDMIPCM for the SUN9I SUN8IW6 chips

如需加载 spdif, 需要选择:

sunxi On-Chip spdif

(7) 配置中默认打开了 gpio 检测耳机驱动，公版音频驱动需要将耳机驱动去掉。配置过程如下图

1) 如下图

```

CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
[*] Networking support --->
Device Drivers --->
File systems --->
Kernel hacking --->
Security options --->
-* Cryptographic API --->
Library routines --->
---
Load an Alternate Configuration File
Save an Alternate Configuration File

```

2) 如下图


```

[*] DMA Engine support --->
[ ] Auxiliary Display support --->
< > Userspace I/O drivers --->
    Virtio drivers --->
    Microsoft Hyper-V guest support --->
[*] Staging drivers --->
    Common Clock Framework --->
    Hardware Spinlock drivers --->

```

3) 如下图

```

< > Synaptics i2c rmi4 touchscreen
[ ] Media staging drivers --->
[*] Android --->
< > Telephony support --->
< > USB over WiFi Host Controller

```

4) 去掉 Android GPIO ts3a225 Switch support 驱动配置。

```

< > Android Switch class support --->
< > swith headset support
< > Android GPIO Switch support
[*] Android GPIO ts3a225 Switch support
[*] Android alarm driver

```

2.4. 代码位置

```

lichee\linux-3.4\drivers\mfd\ac100-core.c
lichee\linux-3.4\include\linux\mfd\ac100\core.h
lichee\linux-3.4\sound\soc\codecs\ac100.h
lichee\linux-3.4\sound\soc\codecs\ac100_dapm.c
lichee\linux-3.4\sound\soc\sunxi\daudio0\sunxi-dauido.c
lichee\linux-3.4\sound\soc\sunxi\daudio0\sunxi-dauido0.h
lichee\linux-3.4\sound\soc\sunxi\daudio0\as3_ac100.c
lichee\linux-3.4\sound\soc\sunxi\daudio0\bb_dai.c
lichee\linux-3.4\sound\soc\sunxi\daudio0\snddauido.c

```

```

lichee\linux-3.4\sound\soc\sunxi\audio0\sunxi-daudiodma.c
lichee\linux-3.4\sound\soc\sunxi\audio0\sunxi-daudiodma0.h
lichee\linux-3.4\sound\soc\sunxi\audio0\sunxi_snddaudio0.c

```

2.5. Sysconfig 配置

```

;-----
;audio_master:1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master)      use
;          2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master)  not use
;          3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave) not use
;          4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave)         use
;audio_select:0 is pcm.1 is i2s
;audio_format: 1:SND_SOC_DAIFMT_I2S(standard i2s format).          use
;          2:SND_SOC_DAIFMT_RIGHT_J(right justified format).
;          3:SND_SOC_DAIFMT_LEFT_J(left justified format)
;          4:SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC
rising edge). use
;          5:SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC
rising edge)
;signal_inversion:1:SND_SOC_DAIFMT_NB_NF(normal bit clock + frame)  use
;          2:SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM)
;          3:SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM)  use
;          4:SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM)
;over_sample_rate: support 128fs/192fs/256fs/384fs/512fs/768fs
;sample_resolution:16bits/20bits/24bits
;word_select_size      :16bits/20bits/24bits/32bits
;pcm_sync_period       :16/32/64/128/256
;msb_lsb_first         :0: msb first; 1: lsb first
;sign_extend           :0: zero pending; 1: sign extend
;slot_index            :slot index: 0: the 1st slot - 3: the 4th slot
;slot_width            :8 bit width / 16 bit width
;frame_width           :0: long frame = 2 clock width; 1: short frame
;tx_data_mode          :0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
;rx_data_mode          :0: 16bit linear PCM; 1: 8bit linear PCM; 2: 8bit u-law; 3: 8bit a-law
;-----
[tdm0]
audio_used      = 1
audio_master    = 4
audio_select    = 1
audio_format    = 1
signal_inversion = 1
mclk_fs        = 512
sample_resolution = 16

```

```

slot_width_select = 16
pcm_lrck_period    = 32
pcm_lrckr_period   = 1
msb_lsb_first      = 0
sign_extend        = 0
slot_index         = 0
slot_width         = 32
frame_width        = 0
tx_data_mode       = 0
rx_data_mode       = 0
i2s_mclk           = port:PB08<3><1><default><default>
i2s_bclk           = port:PB05<3><1><default><default>
i2s_lrclk          = port:PB04<3><1><default><default>
i2s_dout0          = port:PB06<3><1><default><default>
;i2s_dout1         =
;i2s_dout2         =
;i2s_dout3         =
i2s_din            = port:PB07<3><1><default><default>

[audio0]
audio_int_ctrl      = port:PL12<6><default><default><0>      /*耳机中断 gpio 配置*/
audio_pa_ctrl       = port:PG13<1><default><default><0>      /*喇叭功放 gpio 配置*/
headset_val = 0x3b      /*耳机输出音量：0-----0x3f*/
single_speaker_val = 0x19 /*单喇叭输出音量：0-----0x1f*/
double_speaker_val = 0x1b /*双喇叭输出音量：0-----0x1f*/
speaker_double_used = 1 /*1：双喇叭 0：单喇叭*/
earpiece_val = 0x1e     /*听筒音量：0-----0x1f*/
mainmic_val = 0x4       /*主 mic 录音增益：0-----0x7*/
headsetmic_val = 0x4
ts3a225_gpio_ctrl   = port:PG12<6><default><default><0>      /*ti 芯片方案耳机驱动 gpio 配置*/
dmic_used = 0        /*1：方案配置为 dmic 0：为模拟 mic*/
adc_digital_val = 0xb0b0 /*方案为 d-mic 时，数字增益，保持默认*/
agc_used = 0         /*1：打开 agc 功能*/
drc_used = 1         /*1：打开 drc 功能*/
analog_bb = 0        /*1：数字通话方案*/
digital_bb = 0       /*1：模拟通话方案*/

[spdif0]
spdif_used          = 1
spdif_dout           = port:PE18<3><1><default><default>      /*spdif 输出 gpio 配置*/

```

3. 使用描述

A83T 驱动中用户对音频驱动的操作，完全按照标准 alsa-lib 或者 tinyalsa 操作即可。

3.1. Daudio

驱动中将 Daudio 和 AC100 封装为 card0, 其中 AC100 驱动采用标准的 DAPM 机制实现。例如, 用 android 提供的 tinymix 工具查看 card0 的 ctl。

0	INT	2	AIF1 ADC timeslot 0 volume	160 160
1	INT	2	AIF1 ADC timeslot 1 volume	160 160
2	INT	2	AIF1 DAC timeslot 0 volume	159 159
3	INT	2	AIF1 DAC timeslot 1 volume	160 160
4	INT	2	AIF1 ADC timeslot 0 mixer gain	0 0
5	INT	2	AIF1 ADC timeslot 1 mixer gain	0 0
6	INT	2	AIF2 ADC volume	160 160
7	INT	2	AIF2 DAC volume	160 160
8	INT	2	AIF2 ADC mixer gain	0 0
9	INT	2	ADC volume	160 160
10	INT	2	DAC volume	0 0
11	INT	2	DAC mixer gain	0 0
12	INT	1	digital volume	0
13	INT	2	ADC input gain	3 3
14	INT	1	MIC1 boost amplifier gain	4
15	INT	1	MIC2 boost amplifier gain	4
16	INT	1	LINEINL-LINEINR pre-amplifier gain	4
17	INT	1	AUX1 pre-amplifier gain	4
18	INT	1	AXin to L_R output mixer gain	3
19	INT	1	MIC1 BST stage to L_R outp mixer gain	3
20	INT	1	MIC2 BST stage to L_R outp mixer gain	3
21	INT	1	LINEINL/R to L_R output mixer gain	3
22	INT	1	earpiece volume	30
23	INT	1	speaker volume	27
24	INT	1	line out volume	3
25	INT	1	headphone volume	59
26	ENUM	1	ADCR Mux	ADC
27	ENUM	1	ADCL Mux	ADC
28	BOOL	1	Line Out Mixer MIC1 boost Switch	Off
29	BOOL	1	Line Out Mixer MIC2 boost Switch	Off
30	BOOL	1	Line Out Mixer Rout_Mixer_Switch	Off
31	BOOL	1	Line Out Mixer Lout_Mixer_Switch	Off
32	ENUM	1	MIC2 SRC	MIC2
33	BOOL	1	RIGHT ADC input Mixer MIC1 boost Switch	Off
34	BOOL	1	RIGHT ADC input Mixer MIC2 boost Switch	Off
35	BOOL	1	RIGHT ADC input Mixer LINEINL-R Switch	Off

36	BOOL	1	RIGHT ADC input Mixer LINEINR Switch	Off
37	BOOL	1	RIGHT ADC input Mixer AUXINR Switch	Off
38	BOOL	1	RIGHT ADC input Mixer Rout_Mixer_Switch	Off
39	BOOL	1	RIGHT ADC input Mixer Lout_Mixer_Switch	Off
40	BOOL	1	LEFT ADC input Mixer MIC1 boost Switch	Off
41	BOOL	1	LEFT ADC input Mixer MIC2 boost Switch	Off
42	BOOL	1	LEFT ADC input Mixer LININL-R Switch	Off
43	BOOL	1	LEFT ADC input Mixer LINEINL Switch	Off
44	BOOL	1	LEFT ADC input Mixer AUXINL Switch	Off
45	BOOL	1	LEFT ADC input Mixer Lout_Mixer_Switch	Off
46	BOOL	1	LEFT ADC input Mixer Rout_Mixer_Switch	Off
47	ENUM	1	AIF2 DAC SRC Mux	Left_s right_s AIF2
48	ENUM	1	AIF3OUT Mux	
49	BOOL	1	AIF2 ADR Mixer AIF1 DA0R Switch	Off
50	BOOL	1	AIF2 ADR Mixer AIF1 DA1R Switch	Off
51	BOOL	1	AIF2 ADR Mixer AIF2 DACL Switch	Off
52	BOOL	1	AIF2 ADR Mixer ADCR Switch	Off
53	BOOL	1	AIF2 ADL Mixer AIF1 DA0L Switch	Off
54	BOOL	1	AIF2 ADL Mixer AIF1 DA1L Switch	Off
55	BOOL	1	AIF2 ADL Mixer AIF2 DACR Switch	Off
56	BOOL	1	AIF2 ADL Mixer ADCL Switch	Off
57	ENUM	1	AIF2INR Mux	AIF2_DACR
58	ENUM	1	AIF2INL Mux	AIF2_DACL
59	ENUM	1	AIF2OUTR Mux	AIF2_ADCR
60	ENUM	1	AIF2OUTL Mux	AIF2_ADCL
61	ENUM	1	EAR Mux	DACR
62	ENUM	1	SPK_L Mux	MIXEL Switch
63	ENUM	1	SPK_R Mux	MIXER Switch
64	ENUM	1	HP_L Mux	DACL HPL Switch
65	ENUM	1	HP_R Mux	DACR HPR Switch
66	BOOL	1	Right Output Mixer DACL Switch	Off
67	BOOL	1	Right Output Mixer DACR Switch	Off
68	BOOL	1	Right Output Mixer AUXINR Switch	Off
69	BOOL	1	Right Output Mixer LINEINR Switch	Off
70	BOOL	1	Right Output Mixer LINEINL-LINEINR Switc	Off
71	BOOL	1	Right Output Mixer MIC2Booststage Switch	Off
72	BOOL	1	Right Output Mixer MIC1Booststage Switch	Off
73	BOOL	1	Left Output Mixer DACR Switch	Off
74	BOOL	1	Left Output Mixer DACL Switch	Off
75	BOOL	1	Left Output Mixer AUXINL Switch	Off
76	BOOL	1	Left Output Mixer LINEINL Switch	Off
77	BOOL	1	Left Output Mixer LINEINL-LINEINR Switch	Off
78	BOOL	1	Left Output Mixer MIC2Booststage Switch	Off
79	BOOL	1	Left Output Mixer MIC1Booststage Switch	Off

80	BOOL	1	DACR Mixer ADCR Switch	Off
81	BOOL	1	DACR Mixer AIF2DACR Switch	Off
82	BOOL	1	DACR Mixer AIF1DA1R Switch	Off
83	BOOL	1	DACR Mixer AIF1DA0R Switch	Off
84	BOOL	1	DACL Mixer ADCL Switch	Off
85	BOOL	1	DACL Mixer AIF2DACL Switch	Off
86	BOOL	1	DACL Mixer AIF1DA1L Switch	Off
87	BOOL	1	DACL Mixer AIF1DA0L Switch	Off
88	BOOL	1	AIF1 AD1R Mixer AIF2 DACR Switch	Off
89	BOOL	1	AIF1 AD1R Mixer ADCR Switch	Off
90	BOOL	1	AIF1 AD1L Mixer AIF2 DACL Switch	Off
91	BOOL	1	AIF1 AD1L Mixer ADCL Switch	Off
92	BOOL	1	AIF1 AD0R Mixer AIF1 DAOR Switch	Off
93	BOOL	1	AIF1 AD0R Mixer AIF2 DACR Switch	Off
94	BOOL	1	AIF1 AD0R Mixer ADCR Switch	Off
95	BOOL	1	AIF1 AD0R Mixer AIF2 DACL Switch	Off
96	BOOL	1	AIF1 AD0L Mixer AIF1 DAOL Switch	Off
97	BOOL	1	AIF1 AD0L Mixer AIF2 DACL Switch	Off
98	BOOL	1	AIF1 AD0L Mixer ADCL Switch	Off
99	BOOL	1	AIF1 AD0L Mixer AIF2 DACR Switch	Off
100	ENUM	1	AIF1IN1R Mux	AIF1_DA1R
101	ENUM	1	AIF1IN1L Mux	AIF1_DA1L
102	ENUM	1	AIF1IN0R Mux	AIF1_DA0R
103	ENUM	1	AIF1IN0L Mux	AIF1_DA0L
104	ENUM	1	AIF1OUT1R Mux	AIF1_AD1R
105	ENUM	1	AIF1OUT1L Mux	AIF1_AD1L
106	ENUM	1	AIF1OUT0R Mux	AIF1_AD0R
107	ENUM	1	AIF1OUT0L Mux	AIF1_AD0L
108	BOOL	1	AIF2INR Mux VIR switch aif2inr aif3Switc	Off
109	BOOL	1	AIF2INL Mux VIR switch aif2inl aif3Switc	Off
110	BOOL	1	AIF2INR Mux switch aif2inr aif2Switch	Off
111	BOOL	1	AIF2INL Mux switch aif2inl aif2Switch	Off
112	BOOL	1	External Speaker Switch	On
113	BOOL	1	Headphone Switch	On
114	BOOL	1	Earpiece Switch	On

3.1.1. playback 应用

(1) 打开相关设备的 mixer 接口:

```
struct mixer *mixer_open(unsigned int card)
```

(2) 配置音频输出通路:

```
tinymix_set_value(mixer, 57, 0)
```

该示例中 mixer 参数为打开的 mixer 接口, 57 代表 57 号 ctl

```
static void tinymix_set_value(struct mixer *mixer, unsigned int id,
```

```

                                unsigned int value)
{
    struct mixer_ctl *ctl;
    enum mixer_ctl_type type;
    unsigned int num_values;
    unsigned int i;

    ctl = mixer_get_ctl(mixer, id);
    type = mixer_ctl_get_type(ctl);
    num_values = mixer_ctl_get_num_values(ctl);

    for (i = 0; i < num_values; i++) {
        if (mixer_ctl_set_value(ctl, i, value)) {
            fprintf(stderr, "Error: invalid value\n");
            return;
        }
    }
}

```

(3) struct pcm *pcm_open(unsigned int card, unsigned int device,
unsigned int flags, struct pcm_config *config)

：打开相应的音频流接口

(4) int pcm_write(struct pcm *pcm, void *data, unsigned int count)

：向打开的音频流中送入音频数据

应用程序一直通过 pcm_write 向 buffer 中添加数据，直至音频播放结束。

(5) int pcm_close(struct pcm *pcm)：关闭打开的音频接口

(6) void mixer_close(struct mixer *mixer)：关闭打开的 mixer

3.1.2. capture 应用

Capture 的配置与播放类似，详细方式可以参考 tinyalsa。

3.1.3. 播放录音配置 ctl 说明

3.1.3.1. Headphone 输出

播放通路设置：

number	ctl_name	value
1	Headphone volume	0-----31
2	AIF1INOR Mux	AIF1_DA0R
3	AIF1INOL Mux	AIF1_DA0L
4	DACR Mixer AIF1DA0R Switch	1
5	DACL Mixer AIF1DA0L Switch	1
6	HP_R Mux	DACR HPR Switch

7	HP_L Mux	DACL HPL Switch
8	Headphone Switch	1

3.1.3.2. Speaker 输出

播放通路设置:

number	ctl_name	value
1	speaker volume	0-----31
2	AIF1IN0R Mux	AIF1_DA0R
3	AIF1IN0L Mux	AIF1_DA0L
4	DACR Mixer AIF1DA0R Switch	1
5	DACL Mixer AIF1DA0L Switch	1
6	Right Output Mixer DACR Switch	1
7	Left Output Mixer DACL Switch	1
8	SPK_L Mux	MIXEL Switch
9	SPK_R Mux	MIXER Switch
10	External Speaker Switch	1

3.1.3.3. 主 mic 录音

录音通路设置:

number	ctl_name	value
1	AIF1OUT0L Mux	AIF1_AD0L
2	AIF1OUT0R Mux	AIF1_AD0R
3	AIF1 AD0L Mixer ADCL Switch	1
4	AIF1 AD0R Mixer ADCR Switch	1
5	ADCR Mux	ADC
6	ADCL Mux	ADC
7	LEFT ADC input Mixer MIC1 boost Switch	1
8	RIGHT ADC input Mixer MIC1 boost Switch	1
9	MIC1 boost amplifier gain	0----7

3.1.3.4. 耳 mic 录音

录音通路设置:

number	ctl_name	value
1	AIF1OUT0L Mux	AIF1_AD0L

2	AIF1OUT0R Mux	AIF1_AD0R
3	AIF1 AD0L Mixer ADCL Switch	1
4	AIF1 AD0R Mixer ADCR Switch	1
5	ADCR Mux	ADC
6	ADCL Mux	ADC
7	RIGHT ADC input Mixer MIC2 boost Switch	1
8	LEFT ADC input Mixer MIC2 boost Switch	1
9	MIC2 SRC	MIC2
10	MIC2 boost amplifier gain	0-----7

3.1.3.5. D-MIC 录音

录音通路设置：

number	ctl_name	value
1	AIF1OUT0L Mux	AIF1_AD0L
2	AIF1OUT0R Mux	AIF1_AD0R
3	AIF1 AD0L Mixer ADCL Switch	1
4	AIF1 AD0R Mixer ADCR Switch	1
5	ADCR Mux	DMIC
6	ADCL Mux	DMIC

其他详细的通路设计操作请参考 hal 层设计，以及 audio-path 文件。

3.2. Hdmi

Hdmi 声卡注册为 card1。

3.2.1. playback 应用

- (1) struct pcm *pcm_open(unsigned int card, unsigned int device, unsigned int flags, struct pcm_config *config)
：打开相应的音频流接口
- (2) int pcm_write(struct pcm *pcm, void *data, unsigned int count)
：向打开的音频流中送入音频数据
应用程序一直通过 pcm_write 向 buffer 中添加数据，直至音频播放结束。
- (3) int pcm_close(struct pcm *pcm)：关闭打开的音频接口

3.3. Spdif

Spdif 声卡注册为 card2。

3.3.1. playback 应用

- (1) `struct pcm *pcm_open(unsigned int card, unsigned int device, unsigned int flags, struct pcm_config *config)`
: 打开相应的音频流接口
- (2) `int pcm_write(struct pcm *pcm, void *data, unsigned int count)`
: 向打开的音频流中送入音频数据
应用程序一直通过 `pcm_write` 向 buffer 中添加数据, 直至音频播放结束。
- (3) `int pcm_close(struct pcm *pcm)`: 关闭打开的音频接口

Confidential

4. 出错处理

模块设计中是否考虑了各种场景下的容错处理能力？如果有，请按照流程设计或者接口设计的模式给出出错处理的过程。

Confidential

5. 总结

系统总结一下此文档设计。给出可能存在的不足等。

Confidential