



Audio

模块说明文档

文档版本号: SDK-V1.0

发布日期: 2019-03-30

版权所有©珠海全志科技股份有限公司 2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



、全志和其他全志商标均为珠海全志科技股份有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受全志公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，全志公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。



Audio

文档履历

版本号	日期	制/修订人	内容描述
V1.0	2019-03-30	Allwinner	V316 初始化版本

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	V316 音频系统框架概述	3
2.1	V316 原型机音频硬件框架图	3
2.2	V316 软件框架图	5
3	V316 音频模块介绍	6
3.1	Audiocodec 模块功能	7
3.2	Daudio0/2 模块功能	7
3.3	Daudio2(hdmiaudio) 模块功能	7
3.4	dmic 模块功能	8
4	V316 音频配置	9
4.1	源码结构	9
4.2	内核配置	9
4.2.1	Menuconfig 配置	9
4.2.2	音频驱动配置	11
4.3	系统配置 (sys_config.fex)	12
4.3.1	V316 内置模拟音频通道 (codec) 配置	13

4.3.2	HDMI 音频	14
4.3.3	数字音频总线 (I2S)	15
4.3.4	数字 Mic 总线 (dmic)	18
4.3.5	应用操作 (alsa-lib)	18
4.3.6	调试操作	20
5	Declaration	22





表 目 录



图 目 录

2-1	audio frame	3
2-2	frame	4
2-3	frame	5
4-1		10
4-2		11
4-3		12



1 前言

1.1 编写目的

本文档目的是让开发者了解 V316 音频系统框架，能够在 V316 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 V316 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。
- **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
- 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，分为 overrun 和 underrun



2 V316 音频系统框架概述

2.1 V316 原型机音频硬件框架图

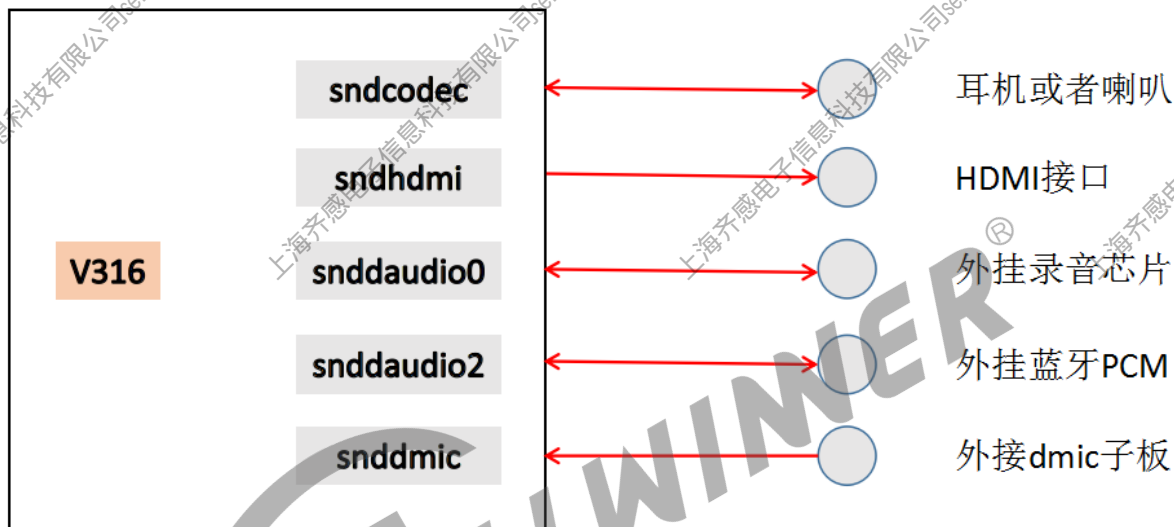


图 2-1: audio frame

各个设备对应的设备节点:

硬件接口设备/	设备节点	/sys/class/sound/cardX/id
Mic	- /dev/snd/controlC0 - /dev/snd/pcmC0D0c	audiocodec
耳机接口喇叭/	- /dev/snd/pcmC0D0p	
接口HDMI	- /dev/snd/controlC1 - /dev/snd/pcmC1D0c - /dev/snd/pcmC1D0p	sndhdm-output record sndhdm
接口audio	- /dev/snd/controlCx - /dev/snd/pcmCxD0c - /dev/snd/pcmCxD0p	snddaudiox

```
| | - /dev/snd/controlCx | |
| DMIC | - /dev/snd/pcmCxD0c | 从中读取获得dmic |
| | - /dev/snd/pcmCxD0p | |
+-----+-----+-----+
```

可以输入以下命令查看系统挂载上的声卡 (为了减少内存, 去掉了声卡的 proc 部分, 可以在 menuconfig 中打开):

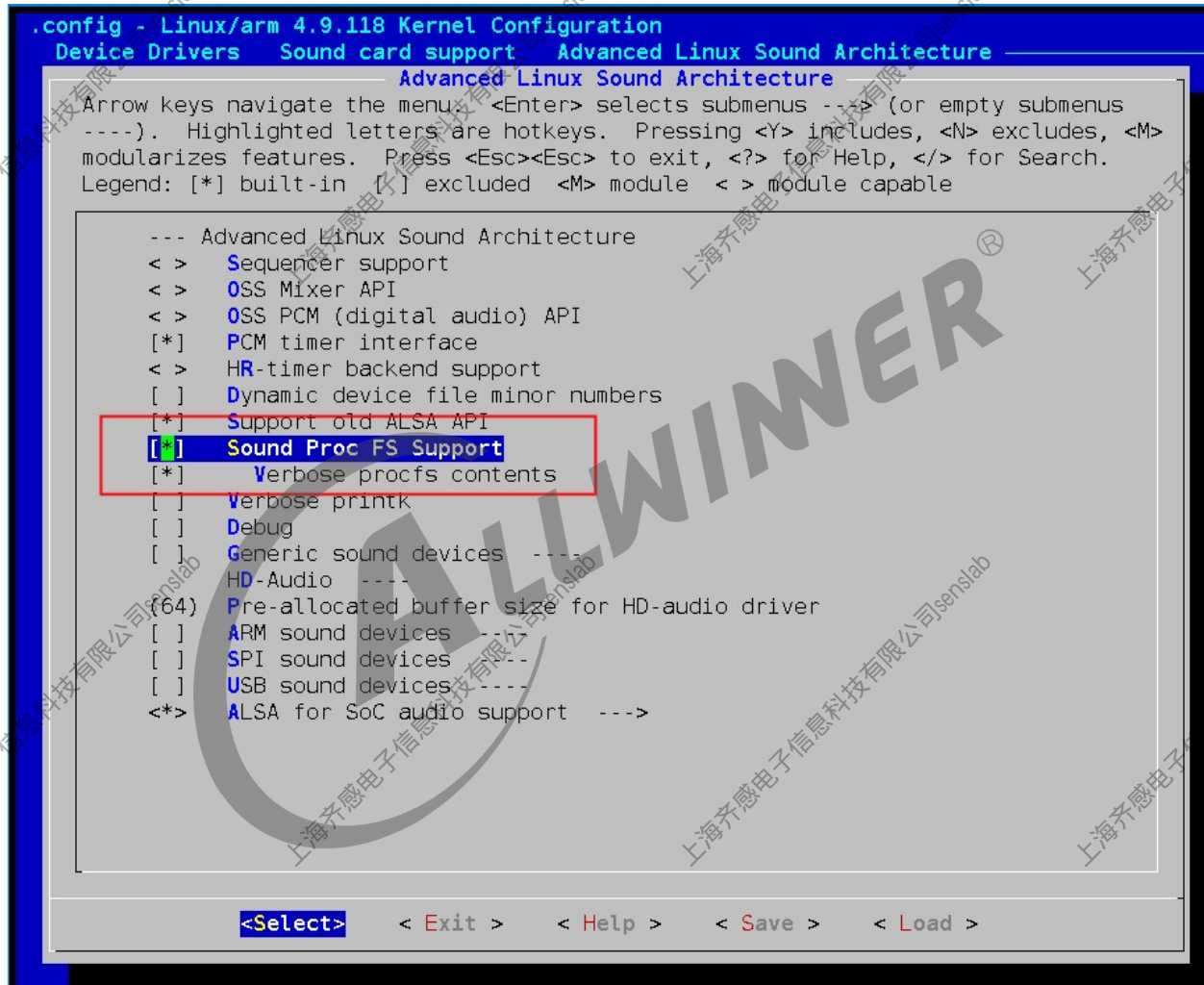


图 2-2: frame

```
cat /proc/asound/cards
0 [audiocodec ]: audiocodec - audiocodec
audiocodec
1 [sndhdmi ]: sndhdmi - sndhdmi
sndhdmi
```

```
2 [snddaudio0 ]: snddaudio0 - snddaudio0
   snddaudio0
3 [snddmic    ]: snddmic - snddmic
   snddmic
```

备注：对于声卡的启用需要配置 `sysconfig` 和 `sun8iw16p1_defconfig`, 具体操作 demo 见下面相应章节描述。

2.2 V316 软件框架图

V316 音频软件框架，大部分沿用原生系统框架。

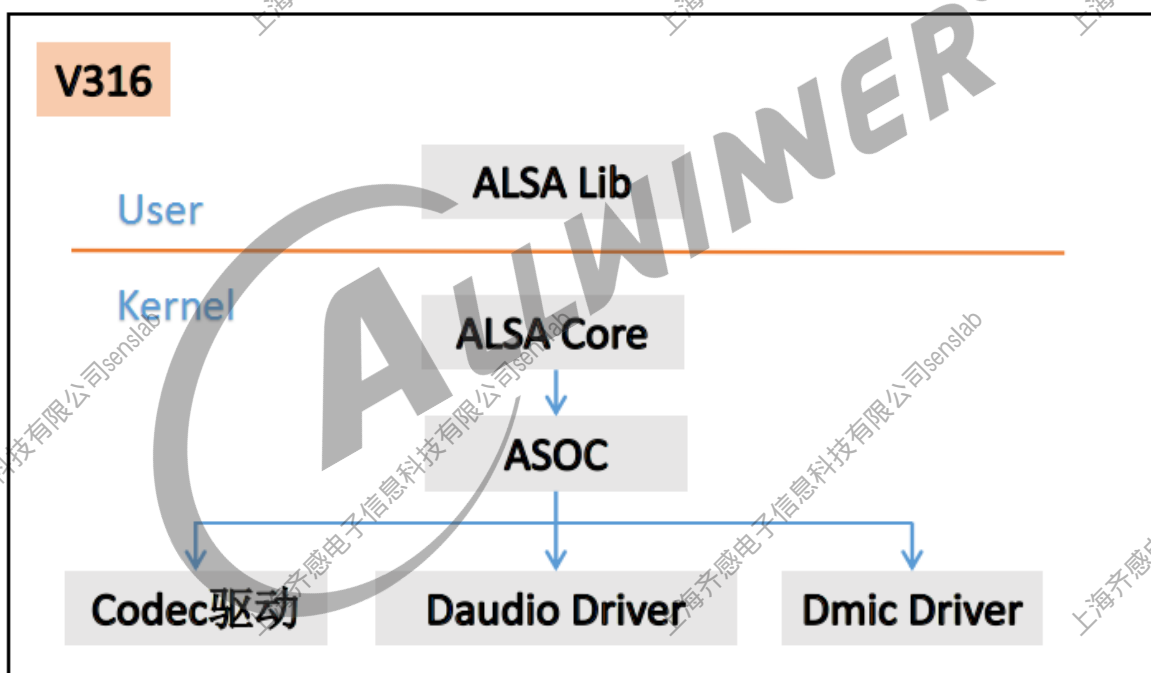


图 2-3: frame

3 V316 音频模块介绍

在 V316 中，最多存在 5 个音频设备。存在 5 个音频设备。分别为 audiocodec, daudio0, HDMI(daudio1), daudio2, dmico。每一个音频设备都采用 asoc 架构实现。

asoc 是建立在标准 alsa 核心层上，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。

● 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 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；
- 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 等等)；一般不包含与板子或 codec 相关的代码。

● 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。

3.1 Audiocodec 模块功能

Audio Codec 驱动所具有的功能:

- 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 mixer 接口;
- 支持 dapm 接口;
- 支持 16bit/24bit 数据精度;
- 支持两路 ADC, 采样率为 8kHz~48kHz;
- 支持两路 DAC, 采样率为 8kHz-192kHz;
- 支持三路输入, 但只支持左右两个声道;

3.2 Daudio0/2 模块功能

驱动所具有的功能:

- 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;
- 支持同时 playback 和 record(全双工模式);
- 支持 i2s、pcm 配置。
- 支持 16bit/24bit/32bit 数据精度

3.3 Daudio2(hdmiaudio) 模块功能

Daudio2(hdmiaudio) 驱动所具有的功能:

- 支持多种采样率格式 (32kHz, 44.1, 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/32bit 数据精度；
- 支持 raw 数据输出；

Hdmiaudio 功能采用 daudio1 接口。

3.4 dmic 模块功能

dmic 驱动所具有的功能：

- 支持多种采样率格式 (8khz, 11.025khz, 12khz, 16khz, 22.05khz, 24khz, 32khz, 44.1khz, 48khz)；
- 只支持 record 模式；
- 支持 16bit/24bit 数据精度；
- 支持 1-8 个声道；

4

V316 音频配置

4.1 源码结构

关键源码目录: `lichee/linux-4.9/sound/soc/sunxi/`

codec 驱动: * `codec: sun8iw16-codec.c` * `cpudai: sunxi-inter-i2s.c` * `machine: sun8iw16-sndcodec.c`

daudio 驱动: * `codec: soc-util.c` * `cpudai: sunxi-daudio.c` * `machine: sunxi-snddaudio.c`

hdmi audio 驱动: * `codec: sunxi-hdmi.c` * `cpudai: sunxi-daudio.c` * `machine: sunxi-sndhdmi.c`

dmic 驱动: * `codec: dmic.c` * `cpudai: sunxi-dmic.c` * `machine: sunxi-snddmic.c`

4.2 内核配置

4.2.1 Menuconfig 配置

在编译服务器上, 目录为 `lichee/linux-4.9` 上, 输入命令: `make ARCH=arm menuconfig`

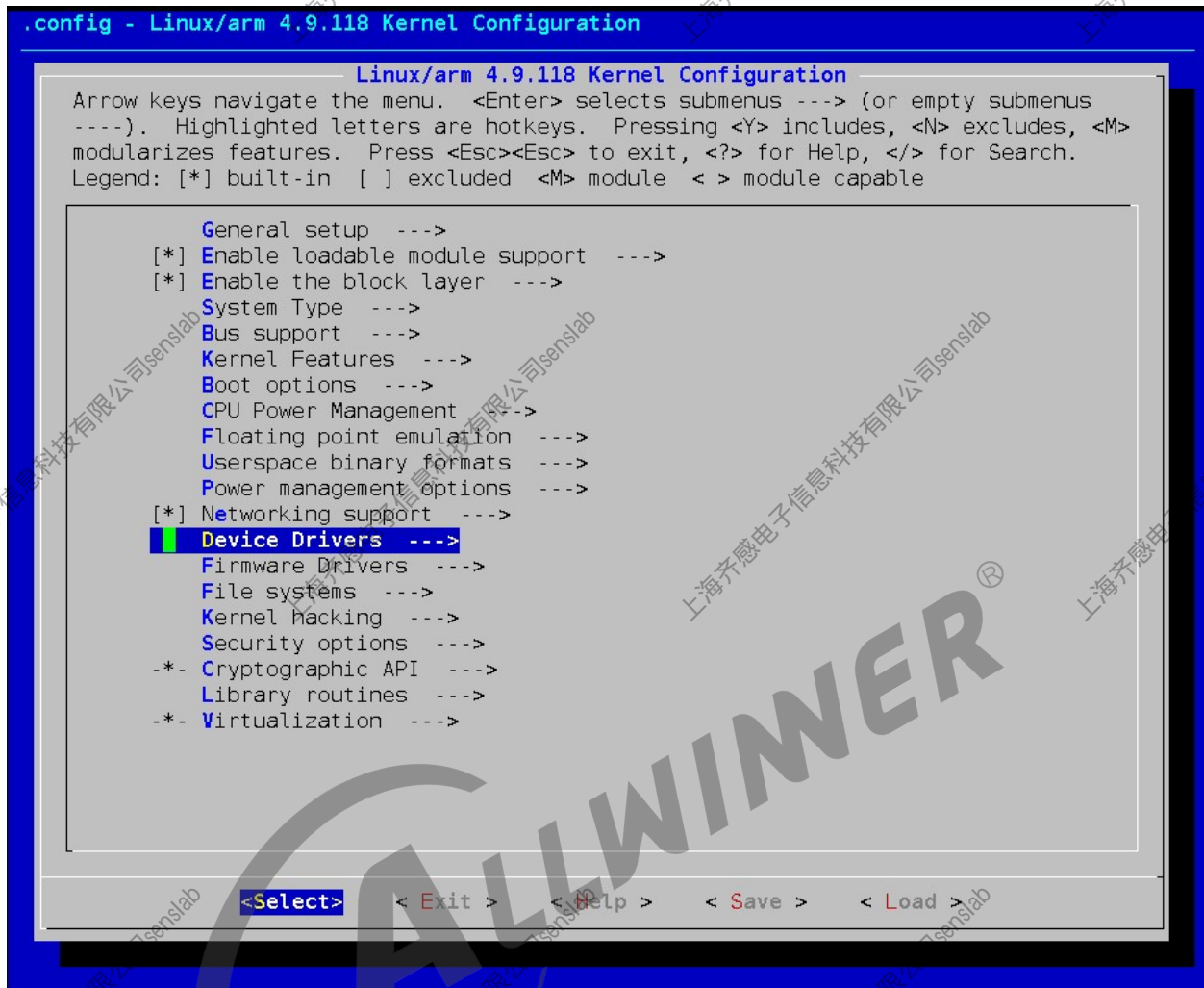


图 4-1

4.2.2 音频驱动配置

```
.config - Linux/arm 4.9.118 Kernel Configuration
Search (SND_SUNXI_SOC)

Symbol: SND_SUNXI_SOC_DAUDIO [=n]
Type : tristate
Defined at sound/soc/sunxi/Kconfig:17
Depends on: SOUND [=m] && !M68K && !UML && SND [=m] && SND_SOC [=m]
Selected by: SND_SUNXI_SOC_SUNXI_HDMIAUDIO [=n] && SOUND [=m] && !M68K && !UML &&

Symbol: SND_SUNXI_SOC_DMIC [=n]
Type : tristate
Defined at sound/soc/sunxi/Kconfig:25
Depends on: SOUND [=m] && !M68K && !UML && SND [=m] && SND_SOC [=m]
Selected by: SND_SUNXI_SOC_SUNXI_DMIC [=n] && SOUND [=m] && !M68K && !UML && SND [=m]

Symbol: SND_SUNXI_SOC_HDMI [=n]
Type : boolean
Defined at sound/soc/sunxi/Kconfig:13
Depends on: SOUND [=m] && !M68K && !UML && SND [=m] && SND_SOC [=m]
Selected by: SND_SUNXI_SOC_SUNXI_HDMIAUDIO [=n] && SOUND [=m] && !M68K && !UML &&

Symbol: SND_SUNXI_SOC_INTER_I2S [=n]
Type : boolean
Defined at sound/soc/sunxi/Kconfig:5
Depends on: SOUND [=m] && !M68K && !UML && SND [=m] && SND_SOC [=m]
Selected by: SND_SUNXI_SOC_SUNXI_I2S [=n] && SOUND [=m] && !M68K && !UML && S

Symbol: SND_SUNXI_SOC_RWFUNC [=n]
Type : boolean

( 39%)
< Exit >
```

图 4-2

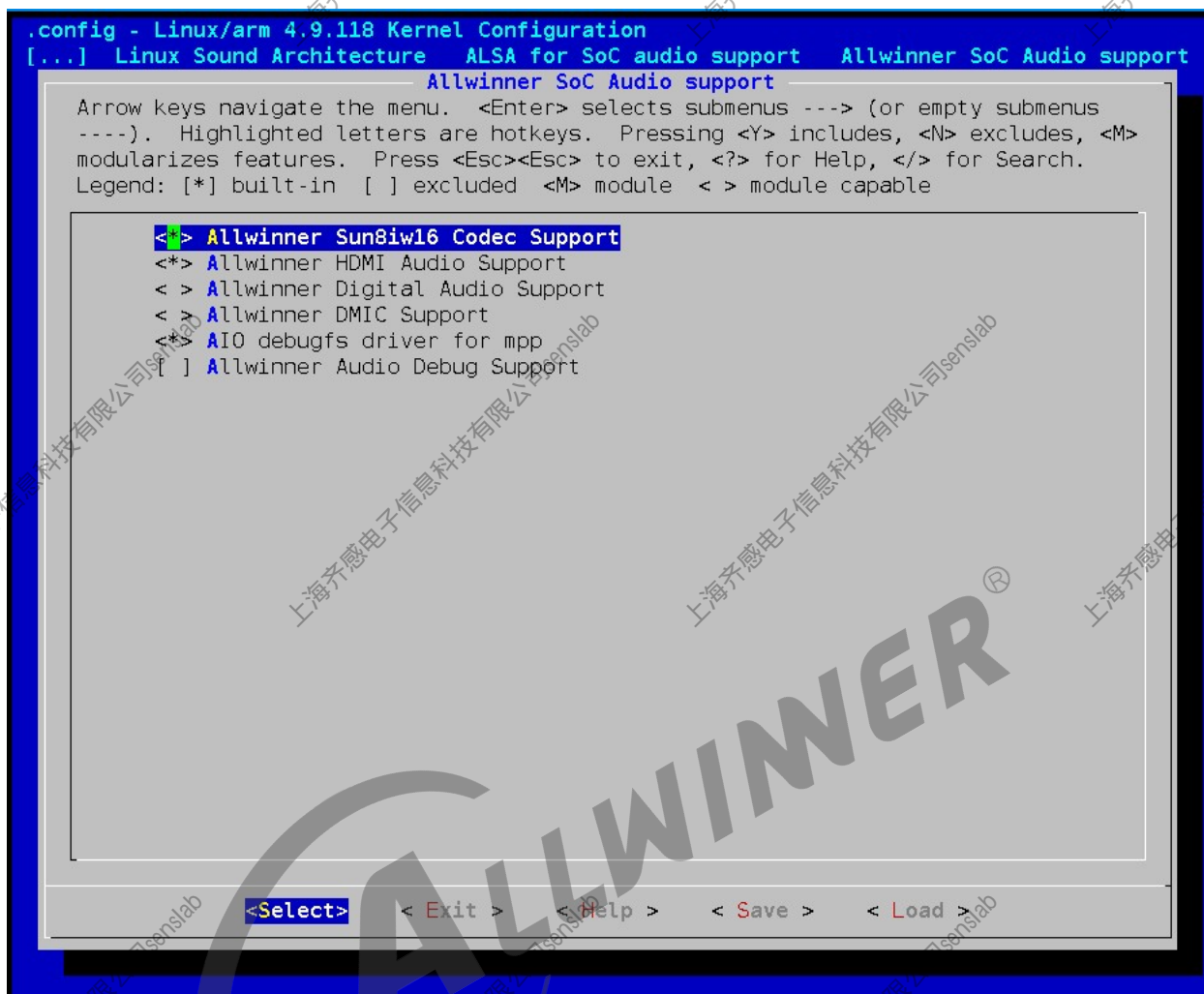


图 4-3

V316 原型机上，CODEC 和 HDMI 都直接编入内核，基于标案低存储低内存，其余端口驱动未编译和启用。

4.3 系统配置（sys_config.fex）

配置文件的位置：lichee/tools/pack/chips/sun8iw16p1/configs/perfl_nor/sys_config.fex 目录下。

4.3.1 V316 内置模拟音频通道 (codec) 配置

● [codec]

配置项配	置项含义
codec_used	是否使用 V316 模拟音频输入输出:0x1: 使用, 0x0: 不使用
lineoutvol	lineout 默认音量设置, 最大值是 0x1f
maingain	Mic1 前端增益, 最大值是 0x7
headsetmicgain	Mic2 前端增益, 最大值是 0x7
adcagc_cfg	Adc 自动增益控制, 0x1: 开启, 0x0: 不开启 (未开放)
adcdrc_cfg	Drc 动态范围控制, 0x1: 开启, 0x0: 不开启
adchpf_cfg	Adc 端高通滤波, 0x1: 开启, 0x0: 不开启
dacdrc_cfg	播放动态音效调节, 0x1: 开启, 0x0: 不开启 (未开放)
dachpf_cfg	播放通路高通滤波开启, 0x1: 开启, 0x0: 不开启 (未开放)
pa_ctl_level	喇叭 pa 需要的使能电平, 0x1: 高电平控制, 0x0: 低电平控制
gpio-spk	外部功放使能脚

配置举例:

```

;-----
;allwinner,lineoutvol : lineout volume:0x0--0x1f 0db-(-43.5db) 1.5db/step
;allwinner,maingain : mainmic gain:0x0--0x7 0x0-0db 0x1:24db 3db/step
;allwinner,headsetmicgain : headphonemic gain:0x0--0x7 0x0-0db 0x1:24db 3db/step
;allwinner,pa_ctl_level : 1:high level control 0:low level control
;allwinner,adcagc_cfg : 1:use adcagc 0:no use
;allwinner,adcdrc_cfg : 1:use adcdrc 0:no use
;allwinner,adchpf_cfg : 1:use adchpf 0:no use
;allwinner,dacdrc_cfg : 1:use dacdrc 0:no use
;allwinner,dachpf_cfg : 1:use dachpf 0:no use
;allwinner,aif2config : 1:use aif2 0:no use
;allwinner,aif3config : 1:use aif3 0:no use
;-----
; NOTE :Make sure sndcodec_used = 0x1,i2s_used = 0x1
;       codec_used = 0x1,if register the sound card audiocodec.
;-----
[sndcodec]
sndcodec_used = 0x1

```

```

[i2s]
i2s_used = 0x1

[codec]
codec_used   = 0x1
lineoutvol   = 0x1b
maingain     = 0x4
headsetmicgain = 0x4
adcagc_cfg   = 0x0
adcdrc_cfg   = 0x0
adchpf_cfg   = 0x0
dacdrc_cfg   = 0x0
dachpf_cfg   = 0x0
pa_ctl_level = 0x0
gpio-spk     = port:PH00<1><default><default><default>

```

4.3.2 HDMI 音频

HDMI 音频设备在 IC 内部直接挂在 Daudio1 上

- [audiohdm]i

配置项	配置项含义
audiohdm_i_used	是否开启 hdmi 音频接口，1：开启，0：不开启

- [sndhdm]i

配置项	配置项含义
sndhdm_i_used	是否注册 hdmi 声卡，1：开启，0：不开启

配置举例：

```

;-----
; NOTE :Make sure audiohdm_i_used = 0x1,sndhdm_i_used = 0x1,
;       if register the sound card hdmi

```

```

;-----
[audiohdm]
audiohdm_iused = 1
[sndhdm]
sndhdm_iused = 1

```

对应 sun8iw16p1.dtsi 配置：配置举例：

```

audiohdm:audio@0x05091000 {
    compatible = "allwinner,sunxi-tdmhdmi";
    reg = <0x0 0x05091000 0x0 0x74>;
    clocks = <&clk_pll_audio>, <&clk_i2s1>;
    status = "okay";
};
sndhdm:sound@2 {
    compatible = "allwinner,sunxi-hdmi-machine";
    sunxi,hdmi-controller = <&audiohdm>;
    status = "okay";
};

```

4.3.3 数字音频总线（I2S）

I2S 接口提供输入和输出功能，一个 mclk，一个 lrcclk，一个 bclk，一个 DOUT，一个 DIN。可根据需求配置。

● [daudio0]

配置项	配置项含义
daudio0_used	是否开启 daudio0 音频接口，1：开启，0：不开启

● [snddaudio0]

配置项	配置项含义
snddaudio0_used	是否注册 daudio0 声卡，1：开启，0：不开启

配置举例:

```

;-----
;audio0_used :0:not use 1:use
;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
;tdm_config:
; 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)
; 2:SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM)
; 3:SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM)
; 4:SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM)
;word_select_size :16bits/20bits/24bits/32bits
;pcm_lrck_period :16/32/64/128/256 表示多少个, 具体关系见bclksunxi-daudio.中关于函数部分cset_clk
;msb_lsb_first :0: msb first; 1: lsb first
;sign_extend :0: zero pending; 1: sign extend
;slot_width_select :8 bit width / 16 bit width / 32 bit width 必须大于或等于使用的采样精度
;frametype :0: short frame = 1 clock width; 1: long frame = 2 clock width
;mclk_div :0: not output(normal setting this) 1/2/4/6/8/12/16/24/32/48/64/96/128/176/192:
; setting mclk as input clock to external codec, freq is pll_audio/mclk_div
;tx_data_mode :0: 16bit linear PCM; (use) 1: reserved;
; :2: 8bit u-law; (no use) 3: 8bit a-law (no use)
;rx_data_mode :0: 16bit linear PCM; (use) 1: reserved;
; :2: 8bit u-law; (no use) 3: 8bit a-law (no use)
;-----
; NOTE :Make sure sndaudio0_used = 0x1, audio0_used = 0x1
; if register the sound card sndaudio0.
;-----

[sndaudio0]
sndaudio0_used = 0

[audio0]
audio0_used = 0
audio_master = 4

```

```
audio_format      = 4
signal_inversion  = 1
slot_width_select = 32
pcm_lrck_period   = 128
msb_lsb_first     = 0
sign_extend       = 0
frametype         = 0
mclk_div          = 1
tdm_config        = 0
tx_data_mode      = 0
rx_data_mode      = 0
```

对应 sun8iw16p1.dtsi 配置，配置举例：

```
daudio0:audio@0x05090000 {
    compatible = "allwinner,sunxi-daudio";
    reg = <0x0 0x05090000 0x0 0x74>;
    clocks = <&clk_pll_audio>, <&clk_i2s0>;
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&daudio0_pins_a>;
    pinctrl-1 = <&daudio0_pins_b>;
    pcm_lrck_period = <0x20>;
    slot_width_select = <0x20>;
    daudio_master = <0x04>;
    audio_format = <0x01>;
    signal_inversion = <0x01>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    tdm_num = <0x0>;
    mclk_div = <0x0>;
    status = "disabled";
};

snddaudio0:sound@1 {
    compatible = "allwinner,sunxi-daudio0-machine";
    sunxi,daudio-controller = <&daudio0>;
    status = "okay";
};
```


4.3.4 数字 Mic 总线 (dmic)

dmic 接口提供输入功能，支持一个 clk，4 个 data pin。可根据需求配置。

- [dmic]

配置项	配置项含义
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

配置举例：

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

```
[dmic]  
dmic_used = 0  
[snddmic]  
snddmic_used = 0
```

4.3.5 应用操作 (alsa-lib)

alsa 是 linux 标准的音频框架，有配套的 alsa-lib 作为应用代码库。tinyalsa: tinyalsa 是 android 的 alsa 音频库，是基于 alsa_lib 的一个精简版。

- [alsa-lib]

```
buildroot/dl/alsa-utiles-1.1.2.tar.bz2  
buildroot/dl/alsa-lib-1.1.2.tar.bz2
```

- [tinyalsa]

buildroot/dl/tinyalsa-v1.0.2.tar.gz 或者

buildroot/dl/tinyalsa-f2a7b6d3d81*.tar.gz

编译 tinyalsa: * 编译目录 IPCLinuxPlatform/system/public/tinyalsa * 编译命令 IPCLinux-Platform/system/public/tinyalsa\$ mk

重新编译打包整个 eyesee 好后, 会自动拷贝 libtinyalsa.so 库和 tinyplay, tinymix, tinycap 工具到相应目录。

由于系统上电启动后会一直开启声卡, 可以关闭 sdvcam 应用, 再使用命令行进行驱动调试。

sndcodec 播放测试:

通路配置

```
tinymix 'AIF1IN0L Mux' AIF1_DA0L
tinymix 'AIF1IN0R Mux' AIF1_DA0R
tinymix 'DACL Mixer AIF1DA0L Switch' 1
tinymix 'DACR Mixer AIF1DA0R Switch' 1
tinymix 'Left Output Mixer DACL Switch' 1
tinymix 'Right Output Mixer DACR Switch' 1
tinymix 'LINEOUTR Mux' ROMIX
tinymix 'LINEOUTL Mux' LOMIX
tinymix 'lineout volume' 18
tinymix 'Lineout Switch' 1 播放命令
```

```
tinypplay xxx.wav -D 0
```

sndcodec 录音测试:

通路配置

```
tinymix 'LADC input Mixer MIC1 boost Switch' 1
tinymix 'RADC input Mixer MIC2 boost Switch' 1
tinymix 'AIF1 AD0R Mixer ADCR Switch' 1
tinymix 'AIF1 AD0L Mixer ADCL Switch' 1
tinymix 'AIF1OUT0R Mux' AIF1_AD0R
tinymix 'AIF1OUT0L Mux' AIF1_AD0L 录音命令
```

```
[]
```

```
tinycap test.wav -D 0 -c 2 -r 16000 需要停止录音时使用  
Ctrl+退出录音。C
```

hdmi 播放测试:

```
tinypayl XXX.wav -D 1
```

4.3.6 调试操作

- 声卡 0 播放设备运行时状态

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

- 声卡 0 录音设备运行时状态

```
cat /proc/asound/card0/pcm0c/sub0/status  
cat /proc/asound/card0/pcm0c/sub0/sw_params  
cat /proc/asound/card0/pcm0c/sub0/hw_params
```

- 应用层设置和内核根据更新的 debugfs 信息

```
cat /sys/kernel/debugfs/mpp/sunxi-ai0
```

- 寄存器信息

打印常用数字域和模拟域寄存器:

audiocodec

```
find -name “” audio_reg_debug
```

```
cat /xxx/audio_reg_debug/audio_reg
```

写数值

0到数字域寄存器地址: x100x00

```
echo 1,1,0x00,0x10 > audio_reg 写数值
```

0到模拟域寄存器地址: x100x00
echo 1,2,0x00,0x10 > audio_reg

读数字域寄存器地址
0的寄存器数值: x00
echo 0,1,0x00 > audio_reg 读模拟域寄存器地址
0的寄存器数值: x00
echo 0,2,0x00 > audio_reg

5

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. tates nor implies warranty of any kind, including fitness for any particular application.

