

Rockchip Linux Audio 开发介绍

发布版本:**0.01**

日期:**2018.06**

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2018 福州瑞芯微电子股份有限公司

超越合理使用范畴，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。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：www.rock-chips.com

客户服务电话：+86-591-83991906

客户服务传真：+86-591-83951833

客户服务邮箱：www.rock-chips.com

概述

本文档主要介绍 Rockchip 在 Linux 平台上 audio 相关开发介绍，以及注意事项。

产品版本

芯片名称	内核版本
RK3308	4.4.x

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2018.06.27	0.1	郑兴	提交 Audio 相关介绍的初始版本

目录

1 Audio on Linux.....1

 1.1 概述.....1

 1.2 准备 Audio 硬件环境.....1

2 Audio Development.....1

 2.1 常用 ALSA 工具.....1

 2.2 常见开发问题.....1

1 Audio on Linux

1.1 概述

目前，在 Linux 平台上，Audio 框架主要使用的是 ALSA。（Advanced Linux Sound Architecture）。ALSA 是 Linux 中提供声音设备驱动的内核组件，用来代替原来的开放声音系统（Open Sound System, OSSv3）。除了声音设备驱动，ALSA 还包含一个用户空间的函数库，开发者可以通过这些高级 API 使用驱动，不必直接与内核驱动进行交互。

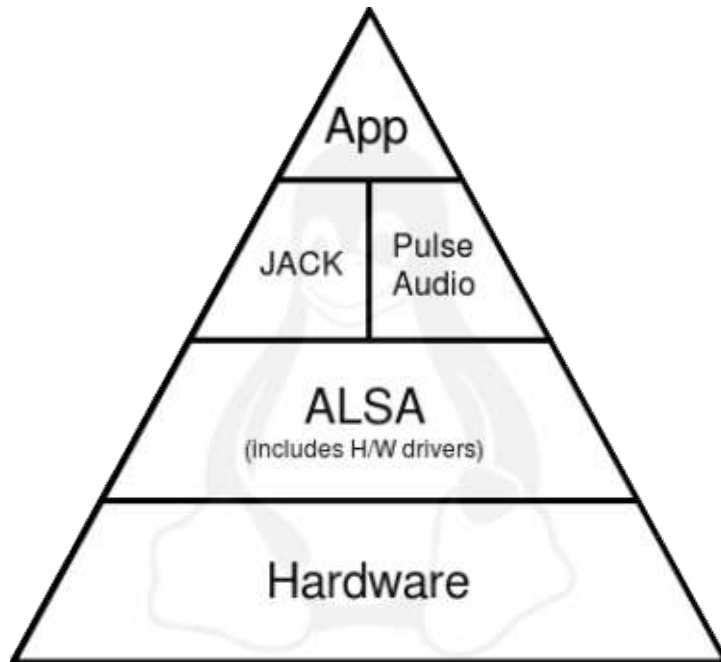


图 1 - The Linux Sound Stack 2017

1.2 准备 Audio 硬件环境

从图 1 可以看到，要实现音频播放，底层（Hardware）的准备工作是必不可少的。这里，我们主要针对 Embedded Linux 环境，简单介绍一下和音频相关的接口及外设：

在 Embedded Linux 平台上，SoC 通常通过 I2S/PCM 接口与外部的 Codec 设备连接，来采集/播放声音。以 I2S 接口为例，SoC 与 Codec 的连接可能是这样：

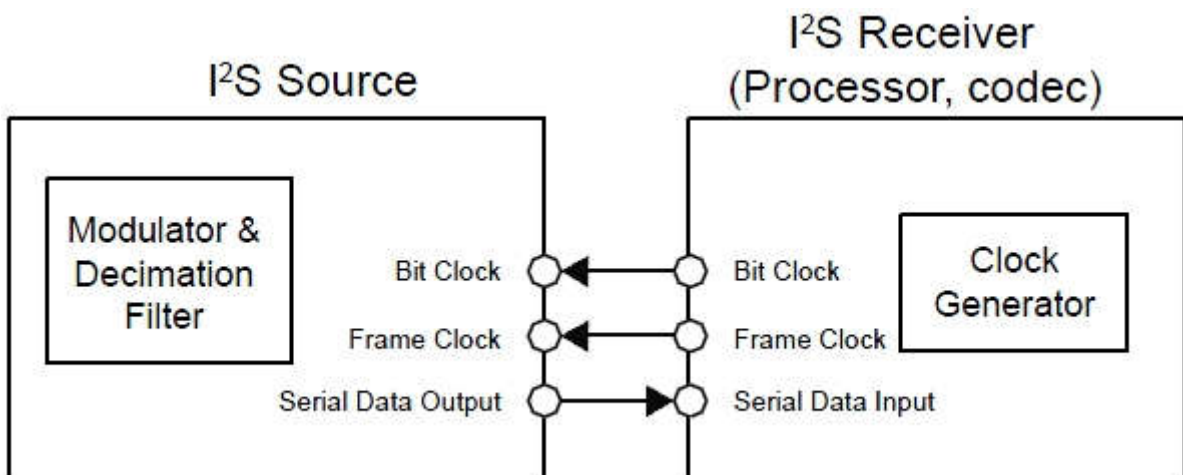
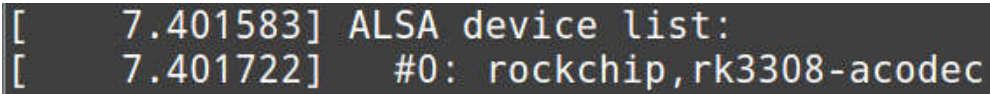


图 2 - I2S Bus 连接图

更多的 I2S 相关协议和资料可以 google:

<https://en.wikipedia.org/wiki/I%C2%B2S>

如果底层的 Audio Hardware 环境已经准备好，在你的平台启动 log 上会看到 codec 成功 binding 的信息:

A terminal window with a dark background showing two lines of text. The first line is "[7.401583] ALSA device list:" and the second line is "[7.401722] #0: rockchip,rk3308-acodec".

```
[ 7.401583] ALSA device list:  
[ 7.401722] #0: rockchip,rk3308-acodec
```

图 3 - ALSA Codec 就绪

2 Audio Development

2.1 常用 ALSA 工具

alsa-utils 提供了 Linux 平台下常用的 ALSA 开发工具：

- aplay

播放音频工具，用户可以用它播放 wav 格式的音频，比如：

```
# aplay sine_440hz_48k_03s.wav
Playing WAVE 'sine_440hz_48k_03s.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

- arecord

录音音频工具，用户可以用它录制麦克风采集的声音，比如录制双声道，48KHz 采样率，16bit 位深的声音文件 record.wav 存放到：/tmp 目录下：

```
# arecord -r 48000 -c 2 -f S16_LE /tmp/record.wav
Recording WAVE '/tmp/record.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

- amixer

音频配置工具，用户可以通过它来配置音频的各项参数。该工具使用的细节比较多，这篇文章写的比较细致，包括前面的 aplay 和 arecord，可以参考：

<https://blog.csdn.net/orz415678659/article/details/8866944>

更多命令细节可以用 help 参数查看。

2.2 常见开发问题

- 指定操作的声卡

可以先用 aplay -L 命令检查声卡是否正常被加载：

```
/ # aplay -L
null
Discard all samples (playback) or generate zero samples (capture)
sysdefault:CARD=rockchiprk3308a
rockchip,rk3308-acodec,
Default Audio Device
sysdefault:CARD=rockchiprk3308p
rockchip,rk3308-pcm,
Default Audio Device
```

比如上图，系统上有两个声卡，我们需要播放声音的声卡是第一个，也就是 card0，所以可以用 -D 指定声卡：

```
# aplay -Dhw:0,0 sine_440hz_48k_03s.wav
Playing WAVE 'sine_440hz_48k_03s.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

其中，hw:0,0 表示的是 card0:device0。如果我们需要把声音推送到第二个声卡上，那就用 hw:1,0 表示：

```
# aplay -Dhw:1,0 sine_440hz_48k_03s.wav
Playing WAVE 'sine_440hz_48k_03s.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

同理，录音的时候，我们也可以用 `arecord -Dhw:0,0` 指定录音的声卡：

```
# arecord -D hw:0,0 -r 48000 -c 2 -f S16_LE /tmp/record.wav
Recording WAVE '/tmp/record.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
```

- 准备 `asound.conf` 文件

`aplay` 和 `arecord` 不使用“-Dhw”参数时，会指定用默认的 `default` 设备打开，具体的描述文件通常在 `/etc/asound.conf`，通过它，用户可以实现更复杂的音频配置，比如描述虚拟音频设备，声道映射，音频通路路由等等。具体可以参考官网的介绍文档：

<https://www.alsa-project.org/main/index.php/Asoundrc>