RK3308

Asound. conf 配置说明

发布版本:**1.20**

日期:**2018.09**

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

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

前言

概述

本文主要针对 RK3308 alsa 框架的 asound.conf 配置说明

产品版本

芯片名称	内核版本
RK3308	4.4

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
2018.7.14	1.00	sch	初始版本
2018.8.23	1.10	sch	增加 EQ_DRC
2018.9.4	1.20	sch	调整一下插件顺序

目录

1 asound.conf 的作用

asound.conf 根据应用需求对 kernel 中的声卡进行配置，常见的配置有放音增益设置和录音声道的转换，Alsa 通过 PCM 插件来实现所需配置，本文重点讲解插件如下：

Asym 插件
Hooks 插件
Soft volume 插件
dmix 插件
Ladspa 插件
Multi 插件
Plug 插件
ctl_elems 插件
Dsnoop 插件
Dshare 插件

其他请参考 alsa-lib-1.1.5\src\pcm 下代码

Asound.conf 的位置在 buildroot\board\rockchip\rk3308\fs-overlay\etc 下

2 Asym 插件配置 arecord/ aplay 默认播放设备

```
pcm.!default
{
    type asym
    playback.pcm {
        type plug
        slave.pcm "softvol"
    }
    capture.pcm {
        type plug
        slave {
            pcm "hw:0,0"
        }
    }
}
```

Playback.pcm 是播放 plug 插件，依赖设备 softvol（第一节中所描述插件）， capture.pcm 是录音 plug 插件， 依赖设备是 hw:0,0 该设备可以通过 arecord -l 命令去查看

3 Softvol 插件设置 PCM 增益的范围和分辨率

```
    pcm.softvol {
        type softvol
        slave.pcm "playback"
        control {
            name "Master Playback Volume"
```

```
        card 0
    }
    min_dB -40.0
    max_dB 0.0
    resolution 100
}
```

Softvol 插件依赖 playback 设备，其作用用来设置增益的范围和分辨率
min_dB -40.0　最小增益
max_dB 0.0　最大增益 ，可以大于 0DB
resolution 100 分辨率 100 等份

备注：该增益调节的仅仅是 PCM 的数字增益。

4 Dmix 插件用来配置播放参数【虚拟声卡】

```
pcm.playback {
    type dmix
    ipc_key 5978293 # must be unique for all dmix plugins!!!!
    ipc_key_add_uid yes
    slave {
        pcm "hw:7,0,0"
        channels 2
        period_size 1024
        buffer_size 4096
    }
    bindings {
        0 0
        1 1
    }
}
```

Playback 设备依赖 hw:7,0,0(该设备是内核虚拟的声卡设备，和实际的硬件未发生关联)，可以通过 aplay -l 去查看

```
    channels 2 ---通道数
    period_size 1024   ----单次传输 1024 个字节
    buffer_size 4096   -----总 BUFFER 大小

 bindings {
        0 0    --- playback 0 通道映射 hw:7,0,0   0 通道
        1 1    --- playback 1 通道映射 hw:7,0,0   1 通道
    }
```

5 Plug 插件提供一个录音接口

```
pcm.fake_record {
    type plug
    slave.pcm "hw:7,1,0"
}
```

后台启动一个进程，从 fake_record 录制声音，该插件的依赖设备是虚拟声卡。

6 Plug 插件提供一个放音接口

```
pcm.fake_play {
    type plug
    slave.pcm "ladspa"
}
```

第四节提到的后台进程，从 fake_record 录制的声音送到 fake_play

7 Ladspa 插件实现自定义算法

```
   pcm.ladspa {
    type ladspa
    slave.pcm "plug:real_playback"
    channels 2
    path "/usr/lib"
    playback_plugins [{
        label eq_drc_stereo
            input {
                controls [4]
            }
    }]
}
```

如果想拦截送往声卡 PCM 经过自定义算法，可以通过这个插件来配置

plug:real_playback---- 依赖放音插件 real_playback

channels 2 ----算法的通道数

path "/usr/lib" ---- 算法库所在的路径

label eq_drc_stereo --- 算法库名

controls [4] --- 参数。

8 Dshare 插件配置播放参数【物理声卡】

```
pcm.real_playback {
    type dshare
    ipc_key 5978293 # must be unique for all dmix plugins!!!!
    ipc_key_add_uid yes
    slave {
        pcm "hw:0,0"
        channels 2
    rate 48000
        period_size 1024
        buffer_size 4096
    }
    bindings {
        0 0
        1 1
    }
}
```

real_playback 设备依赖 hw:0,0 可以通过 aplay -l 去查看
channels 2 ---通道数
period_size 1024 ----单次传输 1024 个字节
buffer_size 4096 -----总 BUFFER 大小

```
bindings {
        0 0    --- real_playback 0 通道映射 hw:0,0   0 通道
        1 1    --- real_playback 1 通道映射 hw:0,0   1 通道
    }
```

9 multi 插件来实现各种录音的通道的组合

上层算法使用的顶层录音节点是：
2mic_loopback
4mic_loopback
6mic_loopback

目前已经有的 multi 插件设备如下：

multi_2_2（2MIC + 2 LOOPBACK）
multi_2_1 (2MIC + 1 LOOPBACK)
multi_4_1(4MIC + 1 LOOPBACK)
multi_8 (6MIC + 2 LOOPBACK) ---- 目前 KERNEL DTS 将多通道的虚拟成一个声卡，因此该插件仅作参考。

接下来以 multi_2_1 和 multi_8 为做详细讲解。

```
pcm.multi_2_1 {
 type multi                    ------ 插件类型
 slaves.a.pcm "hw:0,0"         ------ a 依赖设备
 slaves.a.channels 4           ------ a 设备的通道，该通道必须是偶数，且小于依赖设备的通道
 bindings.0.slave a
 bindings.0.channel 0          ---- multi_2_1 的 0 通道 对应 a 设备的 0 通道
 bindings.1.slave a
 bindings.1.channel 1          ---- multi_2_1 的 1 通道 对应 a 设备的 1 通道
 bindings.2.slave a
 bindings.2.channel 2          ---- multi_2_1 的 2 通道 对应 a 设备的 2 通道
}
```

```
pcm.multi_8 {
    type multi                    ------ 插件类型
    slaves.a.pcm "hw:1,0"         ------ a 依赖设备
    slaves.b.pcm "hw:0,0"         ------ b 依赖设备
    slaves.a.channels 8           ------ a 设备的通道，该通道必须是偶数，且小于依赖设备的通道
    slaves.b.channels 2           ------ b 设备的通道，该通道必须是偶数，且小于依赖设备的通道
    bindings.0.slave a
    bindings.0.channel 0      ---- multi_8 的 0 通道 对应 a 设备的 0 通道
    bindings.1.slave a
    bindings.1.channel 1      ---- multi_8 的 1 通道 对应 a 设备的 1 通道
    bindings.2.slave a
    bindings.2.channel 2      ---- multi_8 的 2 通道 对应 a 设备的 2 通道
    bindings.3.slave a
    bindings.3.channel 3     ---- multi_8 的 3 通道 对应 a 设备的 3 通道
    bindings.4.slave a
    bindings.4.channel 4      ---- multi_8 的 4 通道 对应 a 设备的 4 通道
    bindings.5.slave a
    bindings.5.channel 5        ---- multi_8 的 5 通道 对应 a 设备的 5 通道
    bindings.6.slave b
    bindings.6.channel 0        ---- multi_8 的 6 通道 对应 b 设备的 0 通道
    bindings.7.slave b
    bindings.7.channel 1        ---- multi_8 的 7 通道 对应 b 设备的 1 通道
}
```

10 使用 dsnoop 插件来配置录音参数

有些应用往往不设置录音参数，使用默认参数会导致底层 BUFFER 开的太大，DMA 出错。所以可以使用 dsnoop 插件来设置录音参数。

```
pcm.dsnooped_6_2 {
        ipc_key 1027
         type dsnoop
         slave {
                  pcm "hw:0,0"
                  channels 8
                  rate 16000
                  period_size 1024
                  buffer_size 4096
         }
 }
```

Ipc_key 1027 ---- dsnoop 插件多进程共享，需要一个共享内存，该值代表共享内存 ID，每个 dsnoop 插件必须分配一个唯一的 ipc_key

type dsnoop ---- 指定该插件的类型为 dsnoop
pcm "hw:0,0" ---- 从设备为 hw:0,0
channels 8 --- 通道为 8
rate 16000 --- 采样率 16K
period_size 1024 --- DMA 中断一次，传输 1024 个采样点
buffer_size 4096 --- BUFFER SIZE，访问 ALSA 接口使用的 BUFFER 大小，单位字节。

物理 MIC 对应的录音声卡通道，请参考 DTS 相关配置说明。

11 Rk3308 asound.conf 各种插件关系图

其中蓝色 hw:0,0 是录音声卡，通过 arecord -l 查看

黑色 hw:0,0, hw:7,0,0 是放音声卡，通过 aplay -l 查看