Tina Linux 音频
开发指南

版本号: 2.3
发布日期: 2022.07.14

版本历史

版本号	日期	制/修订人	内容描述
1.0	2019.03.07	AW0985	初始版本, 支持 R328 平台。
1.1	2019.06.12	AW0985	添加对 R6,R11,R16,R18,R30,R311 平台的描述。
1.2	2019.07.15	AW0985	添加对 R7s,R58,R332,R333,MR133,T7 平台的描述。
1.3	2020.05.15	AW0985	添加对 R329 平台的描述。
1.4	2020.05.20	AW0985	R329 平台增加双声卡配置说明。
1.5	2020.06.01	AW0985	添加对 MR813, R818 平台的描述。
1.6	2020.06.15	AW0985	R329 增加部分 dts 配置描述。
1.7	2020.08.04	AW0985	调整文档格式; 修改个别错误描述; 增加 sound procfs 描述。
1.8	2021.04.06	AW0985	添加对 R528 平台的描述。
1.9	2021.05.15	AW0985	完善 R528 部分描述
2.0	2021.10.18	AW0985	R528 平台添加 RX_SYNC 描述
2.1	2021.12.03	AW1417	R329 平台修改 ADC 数目、spdif 配置以及完善部分描述
2.2	2022.03.14	AWA0099	添加 V853 平台的描述
2.3	2022.07.14	KPA0506	添加 F133 平台的描述

版权所有 © 珠海全志科技股份有限公司。保留一切权利

目　录

1 概述 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 模块介绍 3

2.1 驱动框架 3

2.2 音频接口介绍 4

2.3 R6 音频接口 5

2.3.1 硬件资源 5

2.3.2 时钟源 5

2.3.3 代码结构 6

2.3.4 Audiocodec 7

2.3.4.1 内核配置 7

2.3.4.2 sys_config 配置 7

2.3.4.3 codec 数据通路 9

2.3.5 Daudio 10

2.3.5.1 内核配置 11

2.3.5.2 sys_config 配置 11

2.3.6 外挂 codec:AC101 12

2.3.6.1 内核配置 12

2.3.6.2 sys_config&dts 配置 12

2.3.7 标案音频测试方法 13

2.3.7.1 播放 13

2.3.7.2 录音 14

2.4 R7s 音频接口 14

2.4.1 硬件资源 14

2.4.2 时钟源 15

2.4.3 代码结构 15

2.4.4 Audiocodec 16

2.4.4.1 内核配置 16

2.4.4.2 sys_config 配置 16

2.4.4.3 codec 数据通路 17

2.4.5 Daudio 19

2.4.5.1 内核配置 20

2.4.5.2 sys_config 配置 20

2.4.6 标案音频测试方法 21

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.4.6.1 播放 21
2.4.6.2 录音 22
2.5 R11 音频接口 22
2.5.1 硬件资源 22
2.5.2 时钟源 23
2.5.3 代码结构 24
2.5.4 AudioCodec 25
2.5.4.1 内核配置 25
2.5.4.2 sys_config 配置 25
2.5.4.3 codec 数据通路 26
2.5.5 Daudio 28
2.5.5.1 内核配置 28
2.5.5.2 sys_config 配置 28
2.5.6 标案音频测试方法 29
2.5.6.1 播放 30
2.5.6.2 录音 30
2.6 R16 音频接口 30
2.6.1 硬件资源 30
2.6.2 时钟源 31
2.6.3 代码结构 32
2.6.4 AudioCodec 33
2.6.4.1 内核配置 33
2.6.4.2 sys_config 配置 33
2.6.4.3 codec 数据通路 34
2.6.5 Daudio 39
2.6.5.1 内核配置 40
2.6.5.2 sys_config 配置 40
2.6.6 标案音频测试方法 42
2.6.6.1 播放 42
2.6.6.2 录音 42
2.7 R18 音频接口 43
2.7.1 硬件资源 43
2.7.2 时钟源 43
2.7.3 代码结构 44
2.7.4 AudioCodec 44
2.7.4.1 内核配置 45
2.7.4.2 sys_config 配置 45
2.7.4.3 codec 数据通路 47
2.7.5 Daudio 53
2.7.5.1 内核配置 53
2.7.5.2 sys_config 配置 53
2.7.6 SPDIF 55
2.7.6.1 内核配置 55

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.7.6.2 sys_config 配置 55
2.7.7 外挂 codec:AC108 56
2.7.7.1 内核配置 56
2.7.7.2 sys_config&dts 配置 56
2.7.7.3 使用 57
2.7.8 外挂数字功放 TAS5731 57
2.7.8.1 内核配置 58
2.7.8.2 sys_config&dts 配置 58
2.7.8.3 使用 58
2.7.9 HDMI 音频接口 59
2.7.9.1 内核配置 59
2.7.9.2 sys_config 配置 59
2.7.10 标案音频测试方法 59
2.7.10.1 播放 60
2.7.10.2 录音 60
2.8 R30 音频接口 60
2.8.1 硬件资源 60
2.8.2 时钟源 61
2.8.3 代码结构 61
2.8.4 AudioCodec 62
2.8.4.1 内核配置 63
2.8.4.2 sys_config 配置 63
2.8.4.3 codec 数据通路 65
2.8.5 Daudio 71
2.8.5.1 内核配置 71
2.8.5.2 sys_config 配置 71
2.8.6 DMIC 73
2.8.6.1 内核配置 73
2.8.6.2 sys_config 配置 73
2.8.7 标案音频测试方法 74
2.8.7.1 播放 74
2.8.7.2 录音 75
2.9 R58 音频接口 75
2.9.1 硬件资源 75
2.9.2 时钟源 76
2.9.3 代码结构 76
2.9.4 Daudio 76
2.9.4.1 内核配置 76
2.9.4.2 sys_config 配置 77
2.9.4.3 codec 数据通路 79
2.9.5 标案音频测试方法 84
2.9.5.1 播放 84
2.9.5.2 录音 84

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.10 R332 音频接口 85
2.10.1 硬件资源 85
2.10.2 时钟源 85
2.10.3 代码结构 86
2.10.4 AudioCodec 87
2.10.4.1 内核配置 87
2.10.4.2 sys_config 配置 87
2.10.4.3 codec 数据通路 88
2.10.5 Daudio 90
2.10.5.1 内核配置 91
2.10.5.2 sys_config 配置 91
2.10.6 外挂 codec:AC100 92
2.10.6.1 内核配置 92
2.10.6.2 sys_config 配置 92
2.10.7 标案音频测试方法 92
2.10.7.1 播放 93
2.10.7.2 录音 93
2.11 R333 音频接口 93
2.11.1 硬件资源 93
2.11.2 时钟源 94
2.11.3 代码结构 95
2.11.4 AudioCodec 95
2.11.4.1 内核配置 96
2.11.4.2 sys_config 配置 96
2.11.4.3 codec 数据通路 97
2.11.5 Daudio 99
2.11.5.1 内核配置 100
2.11.5.2 sys_config 配置 100
2.11.6 标案音频测试方法 101
2.11.6.1 播放 101
2.11.6.2 录音 102
2.12 R311 音频接口 102
2.12.1 硬件资源 102
2.12.2 时钟源 102
2.12.3 代码结构 103
2.12.4 AudioCodec 103
2.12.4.1 内核配置 104
2.12.4.2 sys_config 配置 104
2.12.4.3 codec 数据通路 106
2.12.5 Daudio 113
2.12.5.1 内核配置 114
2.12.5.2 sys_config 配置 114
2.12.6 Dmic 116

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.12.6.1 内核配置 117
2.12.6.2 sys_config 配置 117
2.12.7 标案音频测试方法 117
2.12.7.1 播放 118
2.13 MR133 音频接口 118
2.13.1 硬件资源 118
2.13.2 时钟源 118
2.13.3 代码结构 119
2.13.4 AudioCodec 119
2.13.4.1 内核配置 120
2.13.4.2 sys_config 配置 120
2.13.4.3 codec 数据通路 122
2.13.5 Daudio 129
2.13.5.1 内核配置 130
2.13.5.2 sys_config 配置 130
2.13.6 Dmic 132
2.13.6.1 内核配置 132
2.13.6.2 sys_config 配置 132
2.13.7 标案音频测试方法 133
2.13.7.1 播放 133
2.14 R328 音频接口 133
2.14.1 硬件资源 133
2.14.2 时钟源 134
2.14.3 代码结构 134
2.14.4 AudioCodec 135
2.14.4.1 内核配置 135
2.14.4.2 sys_config 配置 135
2.14.4.3 codec 数据通路 137
2.14.5 Daudio 140
2.14.5.1 内核配置 141
2.14.5.2 sys_config 配置 141
2.14.6 Dmic 144
2.14.6.1 内核配置 144
2.14.6.2 sys_config 配置 144
2.14.7 SPDIF 145
2.14.7.1 内核配置 145
2.14.7.2 sys_config 配置 145
2.14.8 MAD 146
2.14.8.1 内核配置 146
2.14.8.2 sys_config 配置 146
2.14.8.3 mixer 控件说明 146
2.14.8.4 使用说明 147
2.14.8.5 能量唤醒阈值参数 148

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.14.8.6 注意事项 149
2.14.9 VAD 149
2.14.9.1 内核配置 149
2.14.9.2 sys_config 配置 150
2.14.9.3 mixer 控件说明 150
2.14.9.4 使用说明 151
2.14.9.5 注意事项 152
2.14.10 标案音频测试方法 152
2.14.10.1 播放 152
2.14.10.2 录音 152
2.15 T7 音频接口 152
2.15.1 硬件资源 153
2.15.2 时钟源 153
2.15.3 代码结构 154
2.15.4 AudioCodec 154
2.15.4.1 内核配置 155
2.15.4.2 sys_config 配置 155
2.15.4.3 codec 数据通路 157
2.15.5 Daudio 160
2.15.5.1 内核配置 161
2.15.5.2 sys_config 配置 161
2.15.6 DMIC 162
2.15.6.1 内核配置 163
2.15.6.2 sys_config 配置 163
2.15.7 SPDIF 163
2.15.7.1 内核配置 164
2.15.7.2 sys_config 配置 164
2.15.8 标案音频测试方法 164
2.15.8.1 播放 164
2.15.8.2 录音 165
2.16 R329 音频接口 165
2.16.1 硬件资源 165
2.16.2 时钟源 166
2.16.3 代码结构 166
2.16.3.1 Linux-Audio 下的代码结构 167
2.16.3.2 Linux-DSP-Audio 下的代码结构 168
2.16.4 AudioCodec 169
2.16.4.1 内核配置 169
2.16.4.2 DTS 配置 169
2.16.4.3 AudioCodec 控件 171
2.16.5 Daudio 172
2.16.5.1 内核配置 172
2.16.5.2 DTS 配置 173

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.16.6 Dmic 176
2.16.6.1 内核配置 176
2.16.6.2 DTS 配置 176
2.16.7 SPDIF 177
2.16.7.1 内核配置 177
2.16.7.2 DTS 配置 177
2.16.8 VAD 178
2.16.8.1 内核配置 178
2.16.8.2 使用说明 178
2.16.8.3 能量唤醒阈值参数 179
2.16.9 标案音频 ALSA 配置 180
2.16.10 DSP 音频相关 180
2.16.10.1 默认支持的命令 181
2.16.11 双声卡配置 182
2.16.11.1 外挂 AC107 182
2.16.11.2 外挂 AC108 186
2.16.11.3 使用 DMIC 188
2.17 MR813 音频接口 189
2.17.1 时钟源 189
2.17.2 代码结构 190
2.17.3 AudioCodec 191
2.17.3.1 内核配置 191
2.17.3.2 DTS 配置 192
2.17.3.3 codec 数据通路 194
2.17.4 Daudio 195
2.17.4.1 内核配置 196
2.17.4.2 DTS 配置 196
2.17.5 DMIC 198
2.17.5.1 内核配置 198
2.17.5.2 DTS 配置 198
2.17.6 SPDIF 199
2.17.6.1 内核配置 199
2.17.6.2 DTS 配置 199
2.17.7 外挂 codec:ac107 200
2.17.7.1 内核配置 200
2.17.7.2 DTS 配置 200
2.17.7.3 使用 201
2.17.8 标案音频测试方法 201
2.17.8.1 播放 201
2.17.8.2 录音 202
2.18 R818 音频接口 202
2.18.1 时钟源 202
2.18.2 代码结构 203

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.3 AudioCodec 204
2.18.3.1 内核配置 204
2.18.3.2 DTS 配置 205
2.18.3.3 codec 数据通路 207
2.18.4 Daudio 208
2.18.4.1 内核配置 209
2.18.4.2 DTS 配置 209
2.18.5 DMIC 211
2.18.5.1 内核配置 211
2.18.5.2 DTS 配置 211
2.18.6 SPDIF 212
2.18.6.1 内核配置 212
2.18.6.2 DTS 配置 212
2.18.7 外挂 codec:ac107 213
2.18.7.1 内核配置 213
2.18.7.2 DTS 配置 213
2.18.7.3 使用 214
2.18.8 标案音频测试方法 214
2.18.8.1 播放 214
2.18.8.2 录音 215
2.19 R528 音频接口 215
2.19.1 时钟源 215
2.19.2 代码结构 216
2.19.3 AudioCodec 217
2.19.3.1 内核配置 217
2.19.3.2 DTS 配置 217
2.19.3.3 codec 数据通路 219
2.19.4 Daudio 221
2.19.4.1 内核配置 222
2.19.4.2 DTS 配置 222
2.19.5 DMIC 223
2.19.5.1 内核配置 224
2.19.5.2 DTS 配置 224
2.19.6 SPDIF 225
2.19.6.1 内核配置 225
2.19.6.2 DTS 配置 225
2.19.7 外挂 codec:ac107 226
2.19.7.1 内核配置 226
2.19.7.2 DTS 配置 226
2.19.7.3 使用 227
2.19.8 RX_SYNC 多声卡同步功能 228
2.19.8.1 内核配置 228
2.19.8.2 dts 配置 228

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.19.9 标案音频测试方法 229
2.19.9.1 播放 229
2.19.9.2 录音 229
2.20 V853 音频接口 229
2.20.1 时钟源 230
2.20.2 代码结构 230
2.20.3 AudioCodec 231
2.20.3.1 硬件特性 231
2.20.3.2 内核配置 231
2.20.3.3 DTS 配置 231
2.20.3.4 codec 数据通路 235
2.20.4 Daudio 237
2.20.4.1 硬件特性 237
2.20.4.2 内核配置 237
2.20.4.3 DTS 配置 237
2.20.5 DMIC 241
2.20.5.1 内核配置 241
2.20.5.2 DTS 配置 241
2.20.6 RX_SYNC 多声卡同步功能 243
2.20.6.1 内核配置 244
2.20.6.2 dts 配置 244
2.20.7 标案音频测试方法 244
2.20.7.1 播放 245
2.20.7.2 录音 245
2.21 F133 音频接口 245
2.21.1 时钟源 245
2.21.2 AudioCodec 246
2.21.2.1 硬件特性 246
2.21.2.2 内核配置 246
2.21.2.3 DTS 配置 247
2.21.3 Daudio 247
2.21.3.1 硬件特性 247
2.21.3.2 内核配置 248
2.21.3.3 DTS 配置 248
2.21.4 DMIC 251
2.21.4.1 硬件特性 251
2.21.4.2 内核配置 251
2.21.4.3 DTS 配置 251
2.21.5 SPDIF 252
2.21.5.1 硬件特性 252
2.21.5.2 内核配置 252
2.21.5.3 DTS 配置 252
2.21.6 标案音频测试方法 253

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.21.6.1 播放 253
2.21.6.2 录音 253

3 3 常用工具及调试方法 255
3.1 3.1 alsa-utils 255
3.1.1 3.1.1 amixer 255
3.1.2 3.1.2 aplay 256
3.1.3 3.1.3 arecord 256
3.1.4 3.1.4 alsaconf 257
3.2 3.2 tinyalsa-utils 260
3.2.1 3.2.1 tinymix 260
3.2.2 3.2.2 tinyplay 260
3.2.3 3.2.3 tinycap 261
3.3 3.3 dump 寄存器 261
3.3.1 3.3.1 dump audiocodec 寄存器 262
3.3.2 3.3.2 dump daudio 寄存器 263
3.3.3 3.3.3 dump dmic 寄存器 264
3.3.4 3.3.4 dump spdif 寄存器 264
3.4 3.4 sound procfs 264

4 4 常用接口说明 266
4.1 4.1 control 接口 266
4.2 4.2 PCM 接口 268

5 开机音乐功能 269
5.1 配置方法 269
5.2 注意事项 271

6 各平台音频模块注意事项 272
6.1 R328 272

ALLWINNER®

插　图

图 2-1 ASoC 框图 4
图 2-2 R6 音频硬件框图 5
图 2-3 R6 时钟源 6
图 2-4 R6 音频通路 9
图 2-5 R7s 音频硬件框图 14
图 2-6 R7s 时钟源 15
图 2-7 R7s 音频通路 17
图 2-8 R11 音频硬件框图 23
图 2-9 R11 时钟源 24
图 2-10 R11 音频通路 26
图 2-11 R16 音频硬件框图 31
图 2-12 R16 时钟源 32
图 2-13 R16 音频通路 34
图 2-14 R18 音频硬件框图 43
图 2-15 R18 时钟源 44
图 2-16 R18 音频通路 47
图 2-17 R30 音频硬件框图 61
图 2-18 R30 时钟源 61
图 2-19 R30 音频通路 65
图 2-20 R58 音频硬件框图 75
图 2-21 R58 ac100 音频通路 79
图 2-22 R332 音频硬件框图 85
图 2-23 R332 时钟源 86
图 2-24 R332 音频通路 88
图 2-25 R333 音频硬件框图 94
图 2-26 R333 时钟源 95
图 2-27 R333 音频通路 97
图 2-28 R311 音频硬件框图 102
图 2-29 R311 时钟源 103
图 2-30 R311 音频通路 106
图 2-31 MR133 音频硬件框图 118
图 2-32 MR133 时钟源 119
图 2-33 MR133 音频通路 122
图 2-34 R328 音频硬件框图 133
图 2-35 R328 时钟源 134
图 2-36 R328 音频通路 137
图 2-37 T7 音频硬件框图 153
图 2-38 T7 时钟源 153
图 2-39 T7 音频通路 157

版权所有 © 珠海全志科技股份有限公司。保留一切权利

图 2-40R329 音频硬件框图 165
图 2-41R329 时钟源 166
图 2-42R329 软件驱动框架 167
图 2-43MR813 时钟源 190
图 2-44MR813 音频通路 194
图 2-45R818 时钟源 203
图 2-46R818 音频通路 207
图 2-47R528 时钟源 216
图 2-48V853 时钟源 230

版权所有 © 珠海全志科技股份有限公司。保留一切权利

表　格

表 2-128 AudioCodec codec 节点配置项 (linux4.9) 232
表 2-129 AudioCodec codec_plat 节点配置项 (linux4.9) 233
表 2-130 AudioCodec codec_mach 节点配置项 (linux4.9) 233
表 2-131 AudioCodec 模块板级配置项 234
表 2-132 amixer 控件表 235
表 2-133 I2S/PCM daudio(n)_plat 节点配置项 (linux4.9) 238
表 2-134 I2S/PCM daudio(n)_mach 节点配置项 (linux4.9) 239
表 2-135 I2S/PCM 模块板级配置项 240
表 2-136 DMIC dmic_plat 节点配置项 (linux4.9) 242
表 2-137 DMIC dmic_mach 节点配置项 (linux4.9) 242
表 2-138 DMIC 模块板级配置项 243

版权所有 © 珠海全志科技股份有限公司。保留一切权利

1 概述

1.1 编写目的

介绍 Tina 平台音频模块的使用方法。

1.2 适用范围

Allwinner 软件平台 Tina。

Allwinner 硬件平台 R6, R7s, R11, R16, R18, R30, R58, R328, R332, R333, R311, MR133, T7, R329, MR813, R818, R818B, R528, H133, V853, F133。

1.3 相关人员

Tina 平台下进行音频模块开发的工程师。

1.4 相关术语

术语	解释说明
ALSA	Advanced Linux Sound Architecture
DMA	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，常使用 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

版权所有 © 珠海全志科技股份有限公司。保留一切权利

术语	解释说明
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 标准音频接口
Dmic	数字麦接口
MAD	Mic Activity Detector, 语音能量检测模块
AGC	Automatic Gain Control
DRC	Dynamic Range Control

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2 模块介绍

Linux 中的音频子系统采用 ALSA 架构实现。ALSA 目前已经成为了 Linux 的主流音频体系结构。在内核设备驱动层，ALSA 提供了 alsa-driver，同时在应用层，ALSA 为我们提供了 alsa-lib，应用程序只要调用 alsa-lib 提供的 API，即可以完成对底层音频硬件的控制。

2.1 驱动框架

Tina SDK 对各个平台的音频设备驱动均采用 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：Codec，Platform 和 Machine。

1. 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,SPI 等);
   - Mixer 和其他的音频控件;
   - Codec 和 ALSA 音频操作接口;

2. 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，AC97 等等）；一般不包含与板子或 codec 相关的代码。

3. 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。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

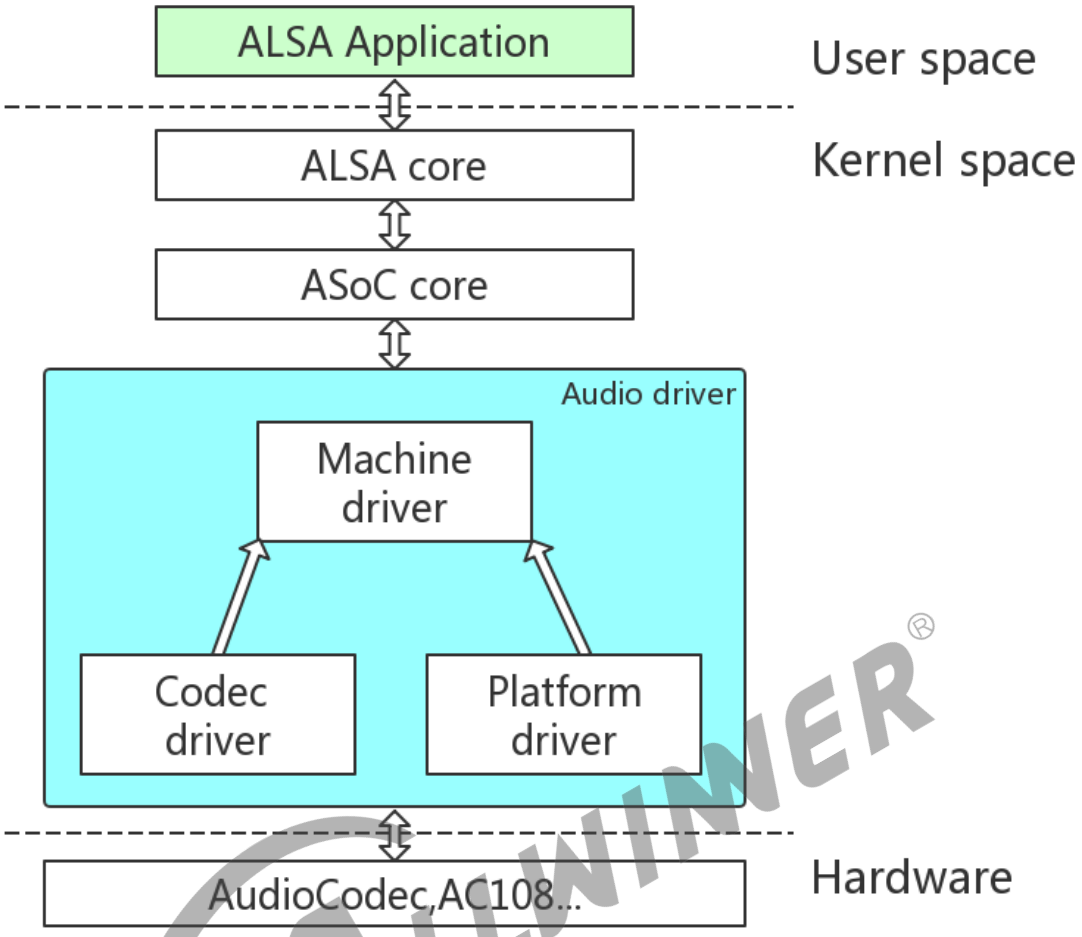


图 2-1：ASoC 框图

2.2 音频接口介绍

我们提供的音频接口有：

- AudioCodec
- Daudio(I2S)
- Dmic
- Spdif
- MAD

不同芯片平台的音频接口资源会有差异；不同版本的内核，对应的 ALSA 驱动也有所不同；下面会对各个芯片作详细介绍。

2.3 R6 音频接口

2.3.1 硬件资源

R6 包含 2 个音频模块，分别是内置 audiocodec 以及 daudio0。

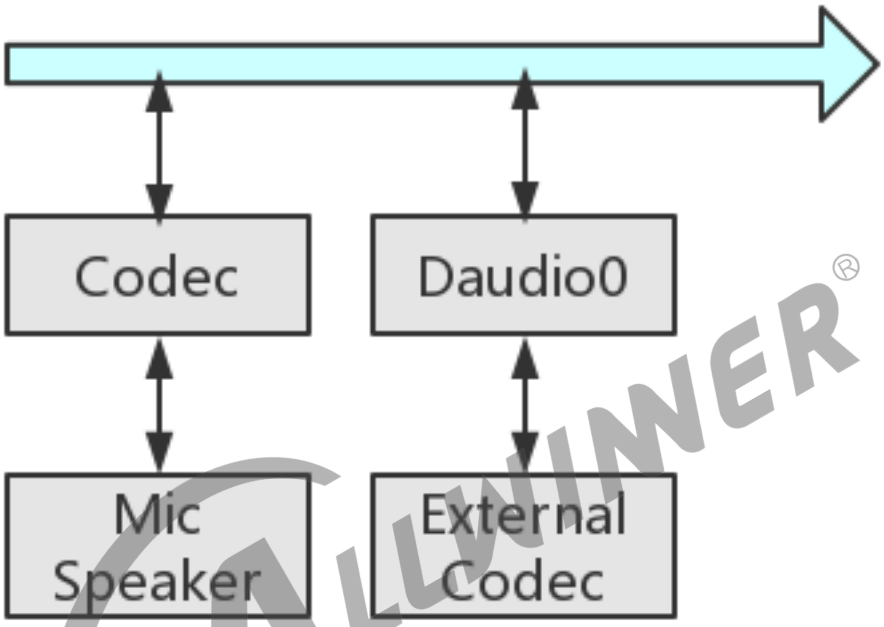


图 2-2：R6 音频硬件框图

2.3.2 时钟源

R6 中，2 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

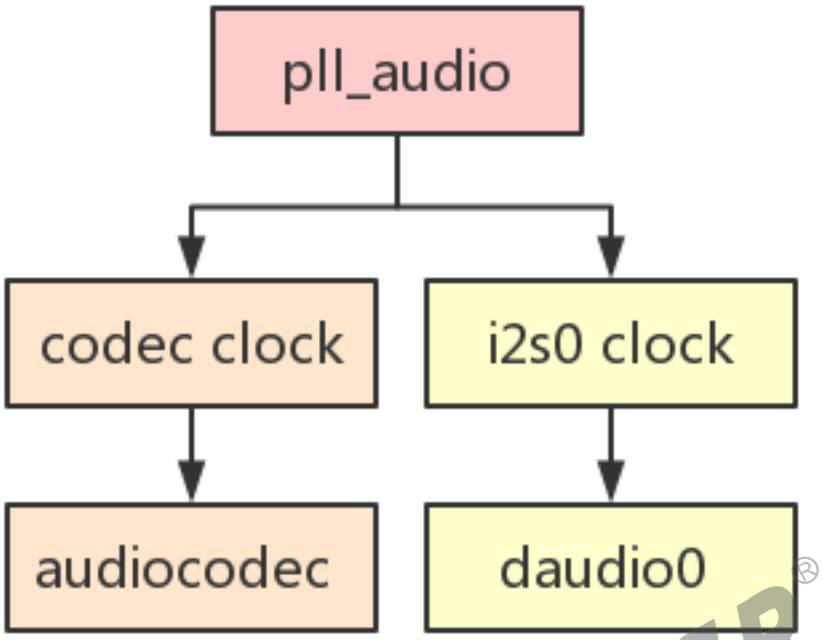


图 2-3: R6 时钟源

2.3.3 代码结构

```
linux-3.10/sound/soc/sunxi/
├── sun3iw1_ac101.c                // daudio+ac101的machine驱动
├── sun3iw1_codec.c                // codec 驱动
├── sun3iw1_codec.h
├── sun3iw1_daudio.c               // daudio的platform驱动
├── sun3iw1_daudio.h
├── sun3iw1_sndcodec.c             // codec machine驱动
├── sunxi_cpudai.c                 // codec platform驱动
├── sunxi_cpudai.h
├── sunxi_dma.c                    // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi_dma.h
├── sunxi_rw_func.c                // 通用文件，读写模拟/数字寄存器的接口
└── sunxi_rw_func.h

linux-3.10/sound/soc/codecs/
├── ac101.c                         // daudio+ac101的codec驱动
└── ac101.h
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.3.4 Audiocodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 一路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 一路模拟输出：一路立体声 headphone 输出 (HPL, HPR)
- 四路模拟输入：MIC,FMINL,FMINR,LINEIN
- 支持同时 playback 和 record(全双工模式)

2.3.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*> ASoC support for SUNXI  --->
            <*> ASoC support for sun3iw1 audiocodec
            <*> ASoC support for internal-codec cpudai
            <*> ASoC support for sun3iw1 audiocodec machine
```

2.3.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used    = 0x1
;-----------------------------------------------------------------------------
[cpudai]
cpudai_used      = 0x1
;-----------------------------------------------------------------------------
[codec]
codec_used       = 0x1
headphonevol     = 0x3b
maingain         = 0
pa_sleep_time    = 30
gpio-spk         = port:PD03<1><1><default><default>
gpio_shdn        = 1
```

sndcodec 配置，即 machine 驱动的相关配置。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用

cpudai 配置，即 platform 驱动的相关配置。

cpudai 配置	cpudai 配置说明
cpudai_used	是否使用 cpudai 驱动。0：不使用；1：使用

codec 配置，即内置 audiocodec 驱动的相关配置。

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
headphonevol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
micgain	mic 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB. 如果作为 aec 回路，则需要设置为 0dB
pa_sleep_time	操作 PA 之后的延时时间 (用来避免 pop 音), 单位 ms
gpio-spk	PA 使能引脚
gpio_shdn	PA 引脚使能方式。0: 低电平有效；1：高电平有效

说明

- ***如果想要正常加载 audiocodec 声卡，需要把 codec,platform,machine 驱动都选上，即 codec_used,cpudai_used,sndcodec_used 都置为 1；***
- ***headphonevol 等值会在驱动初始化的时候设置，进入系统后还可以通过 amixer 工具对应控件进行再次修改；***
- ***注意 gpio-spk 是否配置正确，是否有其他模块复用了该 gpio;***
- ***除了 gpio-spk 指定 pa 使能引脚外，驱动中也会检测 gpio_num 字段，所以可以直接将 gpio 号赋值 gpio_num;***
- ***注意 gpio_shdn，实际功放的 PA 引脚是高电平有效，还是低电平有效***

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.3.4.3 codec 数据通路

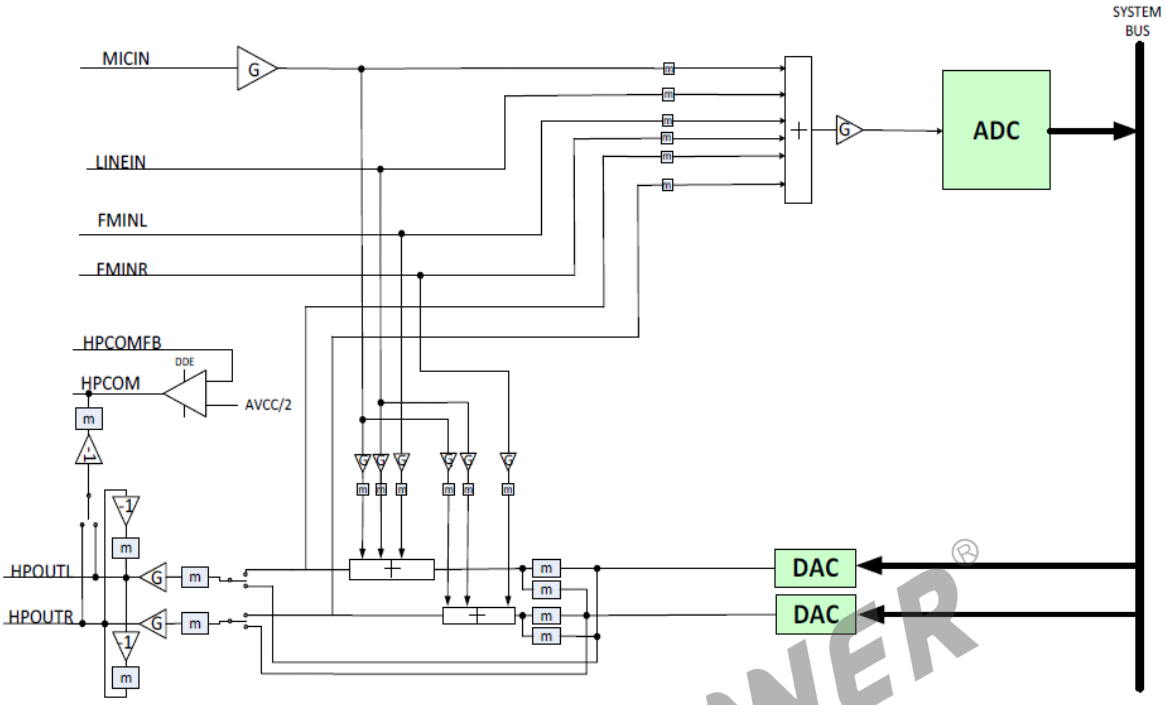


图 2-4: R6 音频通路

R6 平台的 audiocodec 驱动会在播歌的时候自动设置相关通路，默认 audio map:

```
播歌
DACL --> HP_L Mux --> HPOUTL
DACR --> HP_R Mux --> HPOUTR
```

录音功能则根据需要操作对应空间使能通路:

```
录制单MIC数据
MICIN --> ADC Mixer -> ADC

录制内部AEC数据(不需要外围回采电路)
Left Output Mixer --> ADC Mixer -> ADC
Right Output Mixer --> ADC Mixer -> ADC
```

R6 相关控件如下表:

控件名称	功能	数值
ADC INPUT GAIN control	ADC 增益	0–7, 表示-4.5–6dB
ADC MIC Boost AMP en	MIC Boost AMP 使能	0: 关闭; 1: 开启
ADC MIC Boost AMP gain control	MIC 增益	0–7, 0:0dB, 1~7:15–33dB

控件名称	功能	数值
ADC PA speed select	PA 速度选择	0:normal; 1:fast
ADC mixer mute for FML	ADC Mixer 设置，使能 FML 通路	0: 关闭; 1: 开启
ADC mixer mute for FMR	ADC Mixer 设置，使能 FMR 通路	0: 关闭; 1: 开启
ADC mixer mute for left output	ADC Mixer 设置，使能 left output Mixer 通路	0: 关闭; 1: 开启
ADC mixer mute for linein	ADC Mixer 设置，使能 linein 通路	0: 关闭; 1: 开启
ADC mixer mute for mic	ADC Mixer 设置，使能 mic 通路	0: 关闭; 1: 开启
ADC mixer mute for right output	ADC Mixer 设置，使能 right output Mixer 通路	0: 关闭; 1: 开启
LINEIN GAIN control	linein 到 output mixer 的增益	0–7, 0~7:0–14dB, 2dB/step
MICIN GAIN control	MIC 到 outpu mixer 的增益	0–7, 表示-4.5–6dB
dac digital volume	DAC 数字音量	0~63, 表示 0~-73.08dB, -1.16dB/step
head phone volume	headphone 音量	0 表示 mute, 1~63 表示-62dB~0dB, -1dB/step

录音通路设置举例：

1. 录音单声道数据

```
通过MICIN录音:
amixer -D hw:audiocodec cset name='ADC MIC Boost AMP en' 1
amixer -D hw:audiocodec cset name='ADC mixer mute for mic' 1
amixer -D hw:audiocodec cset name='ADC MIC Boost AMP gain control' 4
```

2. 内部 AEC(可省去外部 AEC 电路)

```
amixer -D hw:audiocodec cset name='ADC mixer mute for left ouput' 1
```

2.3.5 Daudio

硬件特性

- 一路 I2S/PCM;
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode

- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 2 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.3.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*> ASoC support for SUNXI  --->
            <*> ASoC support for daudio platform
            <*> ASoC support for sun3iw1 & ac101 daudio machine
```

2.3.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used = 1
over_sample_rate        = 128

[daudio0]
daudio0_used            = 1
word_select_size        = 32
pcm_sync_period         = 32
pcm_lsb_first           = 0
over_sample_rate        = 128
slot_width_select       = 16
pcm_sync_type           = 0
pcm_start_slot          = 0
tdm_config              = 1
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用
over_sample_rate	支持 128fs/192fs/256fs/384fs/512fs/768fs

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用
word_select_size	支持 16bits/20bits/24bits/32bits
pcm_sync_period	16/32/64/128/256

版权所有 © 珠海全志科技股份有限公司。保留一切权利

daudio 配置	daudio 配置说明
pcm_lsb_first	0: msb first; 1: lsb first
over_sample_rate	支持 128fs/192fs/256fs/384fs/512fs/768fs
slot_width_select	16bits/20bits/24bits/32bits
pcm_sync_type	0: long frame sync; 1: short frame sync
pcm_start_slot	0: 1st slot; 1: 2nd slot; 2: 3th slot; 3:4th slot
tdm_config	0:pcm 1:i2s

2.3.6 外挂 codec:AC101

R6 标案使用的 AC101 作双声道录音，audiocodec 则录制回路作 AEC 下面对 R6 如何配置使用 AC101 作简单介绍

2.3.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*> AC101 Codec
         <*> ASoC support for SUNXI  --->
            <*> ASoC support for daudio platform
            <*> ASoC support for sun3iw1 & ac101 daudio machine
```

2.3.6.2 sys_config&dts 配置

R6 通过 TWI1 控制 AC101, 而 I2S0 用于音频数据的传输

TWI 部分配置, 可通过 dts 进行配置:

```
linux-3.10/arch/arm/boot/dts/sun3iw1p1-sitar-mic2.dts
twi1: twi@0x01c27400{
    ac101@1a {
        compatible = "x-powers,ac101";
        reg = <0x1a>;
        audio_int_ctrl = <&pio PL 12 6 1 1 0>;
        audio_pa_ctrl = <&pio PG 13 1 1 1 0>;
        speaker_val = <0x1b>;
        headset_val = <0x3b>;
        single_speaker_val = <0x19>;
        double_speaker_val = <0x1b>;
        speaker_double_used = <1>;
        earpiece_val = <0x1e>;
        mainmic_val = <0x4>;
        headsetmic_val = <0x4>;
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
        dmic_used = <0>;
        adc_digital_val = <0xb0b0>;
        agc_used = <0>;
        drc_used = <1>;
        linein_to_spk_used = <0>;
        linein_to_hp_used = <0>;
        linein_to_aif2_used = <0>;
    };
}
```

I2S 部分配置可以通过 dts 配置，也可以通过 sys_config 覆盖 dts 的配置

```
[snddaudio0]
snddaudio0_used = 1
over_sample_rate        = 128
sunxi,snddaudio-codec = "ac101.1-001a"
sunxi,snddaudio-codec-dai = "ac101"

[daudio0]
daudio0_used            = 1
word_select_size        = 32
pcm_sync_period         = 32
pcm_lsb_first           = 0
over_sample_rate        = 128
slot_width_select       = 16
pcm_sync_type           = 0
pcm_start_slot          = 0
tdm_config              = 1
```

i2s 相关格式需要根据 AC101 spec 进行配置

而 snddaudio0 中，注意 codec 的名称，需要与实际 AC101 的 dev name 相匹配, 而 codec-dai 名称则与 AC101 驱动中设置的 dai name 相匹配

2.3.7 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

2.3.7.1 播放

如《R6 AudioCodec 数据通路》章节所说，驱动代码中已固定配置了播放通路

进入系统后直接通过 aplay 工具进行播放即可，如:

```
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='headphone volume' 50
```

2.3.7.2 录音

标案使用 AC101 进行双声道录音录音前需要配置 AC101 的音频通路，SDK 默认在启动时会进行设置，相关配置脚本在:

```
/etc/init.d/rc.final
```

可以直接通过 arecord 命令进行录音:

```
arecord -Dhw:sndac1011001a -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.4 R7s 音频接口

2.4.1 硬件资源

R7s 包含 2 个音频模块，分别是内置 AudioCodec 以及 Daudio0。

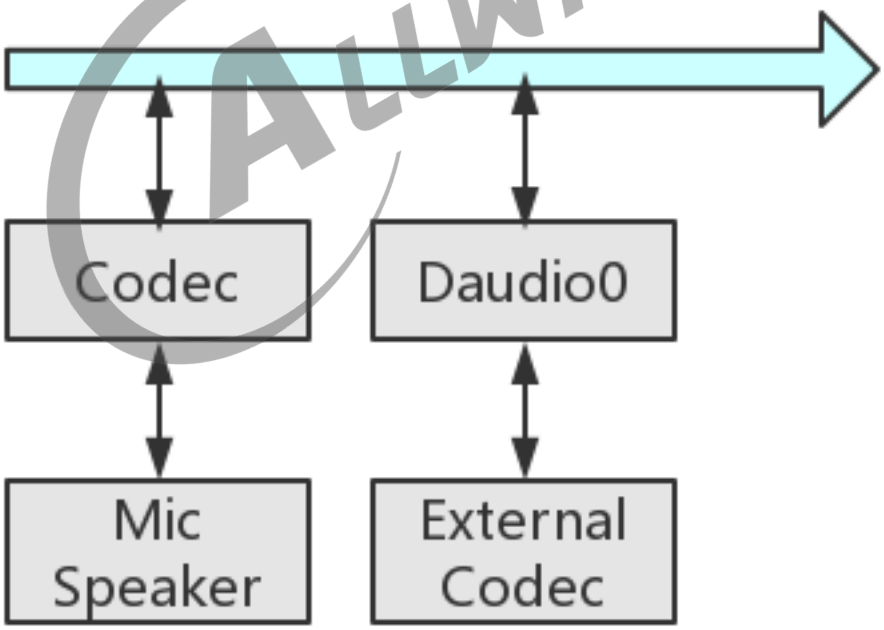


图 2-5: R7s 音频硬件框图

2.4.2 时钟源

R7s 中，2 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

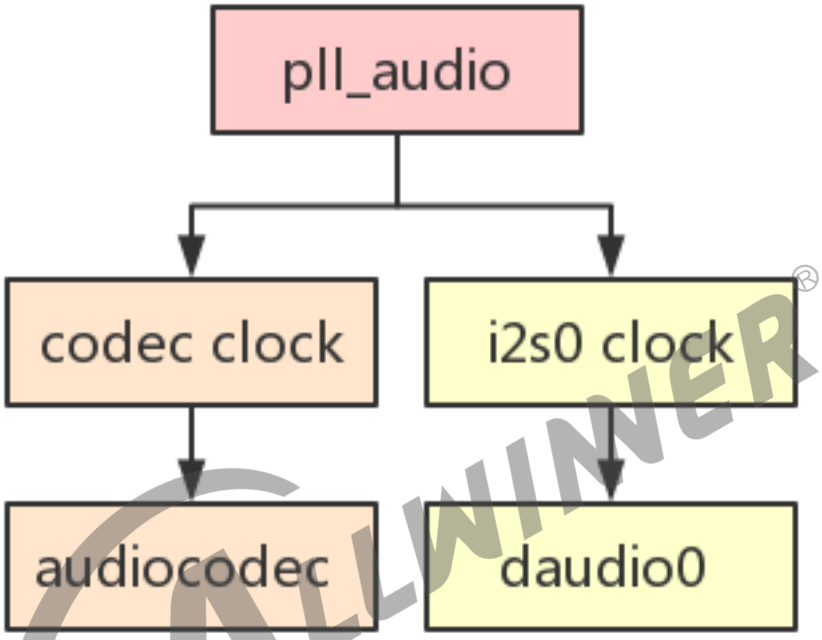


图 2-6: R7s 时钟源

2.4.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sunxi-pcm.c                    // 提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sun8iw8
│   ├── sunxi_codec.c              // cpudai 驱动
│   ├── sunxi_codecdma.c           // codec platform驱动
│   ├── sun8iw8_sndcodec_new.c     // codec 驱动
│   └── sunxi_sndcodec.c           // codec machine驱动
├── sunxi-daudio.c                 // daudio platform驱动
└── sunxi-snddaudio.c              // daudio machine驱动

linux-4.9/sound/soc/soc-utils.c    // daudio codec驱动
```

2.4.4 Audiocodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出:
  - 一路立体声 LINEOUT 输出 (LINEOUTP, LINEOUTN)
  - 一路立体声 headphone 输出 (HPOUTL, HPOUTR)
- 两路模拟输入：MIC1,MIC2
- 支持同时 playback 和 record(全双工模式)
- 支持 ADC 的 AGC,DRC 功能
- 支持 DAC 的 DRC 功能

2.4.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Audiocodec for the SUNXI chips
```

2.4.4.2 sys_config 配置

```
[codec]
headphone_vol           = 0x3b
lineout_vol             = 0x1a
audio_pa_ctrl           = port:PB05<1><default><default><0>
adcagc_used             = 0
adcdrc_used             = 0
dacdrc_used             = 0
adchpf_used             = 0
dachpf_used             = 0
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
headphone_vol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
audio_pa_ctrl	PA 使能引脚
adcagc_used	1:use adcagc 0:no use
adcdrc_used	1:use adcdrc 0:no use
dacdrc_used	1:use dacdrc 0:no use
adchpf_used	1:use adchpf 0:no use
dachpf_used	1:use dachpf 0:no use

2.4.4.3 codec 数据通路

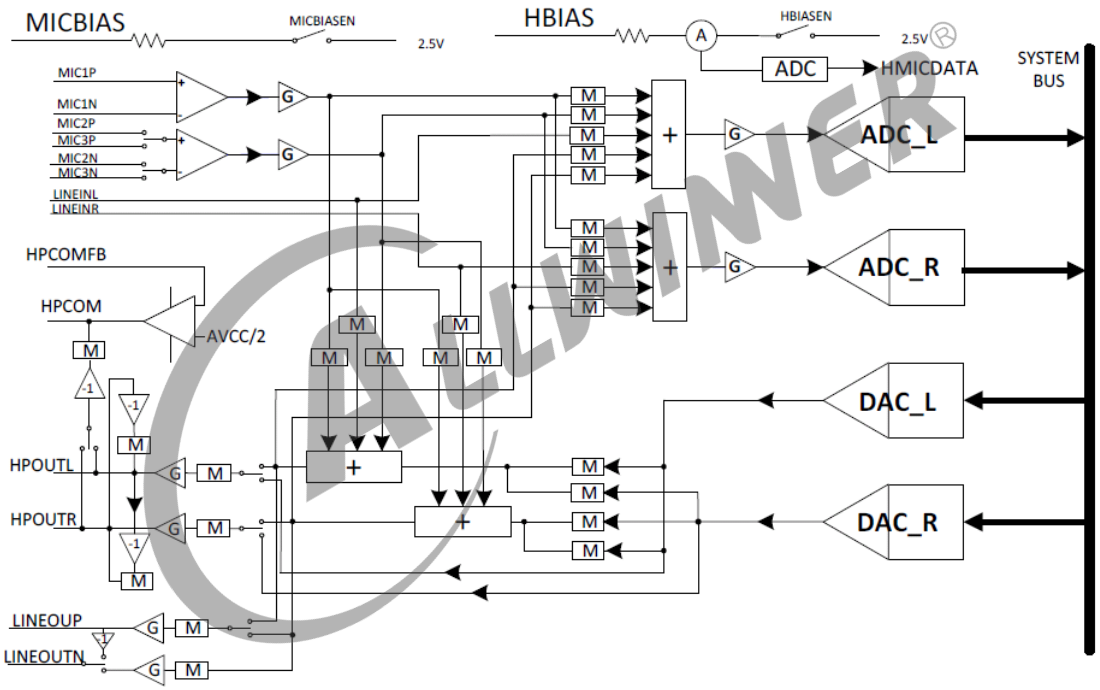


图 2-7: R7s 音频通路

```
播歌
DACL --> Left Output Mixer --> LINEOUTL
DACR --> Right Output Mixer --> LINEOUTR

录音
MIC1P --> LADC input Mixer --> ADCL
MIC2P --> RADC input Mixer --> ADCR
```

控件名称	功能	数值
Lineout volume	lineout 音量设置	0–31, 表示-43.5–0dB
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB
HP_L Mux	HP_L Mux 设置	0:DACL HPL Switch; 1:MIXER_L Switch
HP_R Mux	HP_R Mux 设置	0:DACR HPR Switch; 1:MIXER_R Switch
LADC input Mixer MIC1 boost Switch	LADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LADC input Mixer MIC2 boost Switch	LADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LADC input Mixer l_output mixer Switch	LADC input Mixer 设置，使能 l_output 通路	0: 关闭; 1: 开启
LADC input Mixer r_output mixer Switch	LADC input Mixer 设置，使能 r_output 通路	0: 关闭; 1: 开启
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
MIC1 boost AMP gain control	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC1_G boost stage output mixer control	MIC1 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC2_G boost stage output mixer control	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
RADC input Mixer MIC1 boost Switch	RADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 l_output 通路	0: 关闭; 1: 开启
RADC input Mixer r_output mixer Switch	RADC input Mixer 设置，使能 r_output 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
SPK_L Mux	SPK_L Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
SPK_R Mux	SPK_R Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

通路设置举例：

1. 播放通路

```
通过lineout播放:
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 6
```

2. 录音通路

```
通过MIC1,MIC2录音:
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC2 SRC' 0
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -D hw:audiocodec cset name='MIC2 boost AMP gain control' 4
```

2.4.5 Daudio

硬件特性

- 一路 I2S/PCM；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.4.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Allwinner Digital Audio Support
```

2.4.5.2 sys_config 配置

```
[tdm0]
daudio_used          = 0
daudio_master            = 4
daudio_select            = 1
audio_format             = 1
signal_inversion         = 1
sample_resolution    = 16
slot_width_select        = 16
pcm_lrck_period          = 32
pcm_lrckr_period         = 1
msb_lsb_first        = 0
sign_extend              = 0
tx_data_mode             = 0
rx_data_mode             = 0
;i2s_mclk             = port:PB08<2><1><default><default>
i2s_bclk             = port:PG11<2><1><default><default>
i2s_lrclk            = port:PG10<2><1><default><default>
i2s_dout0            = port:PG12<2><1><default><default>
i2s_dout1            =
i2s_dout2            =
i2s_dout3            =
i2s_din              = port:PG13<2><1><default><default>
```

tdm0 配置	tdm0 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
daudio_select	0: pcm mode; 1: i2s mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

tdm0 配置	tdm0 配置说明
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
sample_resolution	采样精度,16bit, 24bit,32bit
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	可配置 16/32/64/128/256 个 bclk
pcm_lrckr_period	可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
i2s_bclk	i2s_bclk 引脚
i2s_lrclk	i2s_lrclk 引脚
i2s_dout0	i2s_dout 引脚
i2s_din	i2s_din 引脚

2.4.6 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

2.4.6.1 播放

```
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Lineout volume' 24
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 0

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='Lineout volume' 50
```

2.4.6.2 录音

表示下使用 audiocodec 进行 MIC1,MIC2 录音。

```
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -D hw:audiocodec cset name='MIC2 SRC' 0
amixer -D hw:audiocodec cset name='MIC2 boost AMP gain control' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.5 R11 音频接口

2.5.1 硬件资源

R11 包含 2 个音频模块，分别是内置 AudioCodec 以及 Daudio0。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

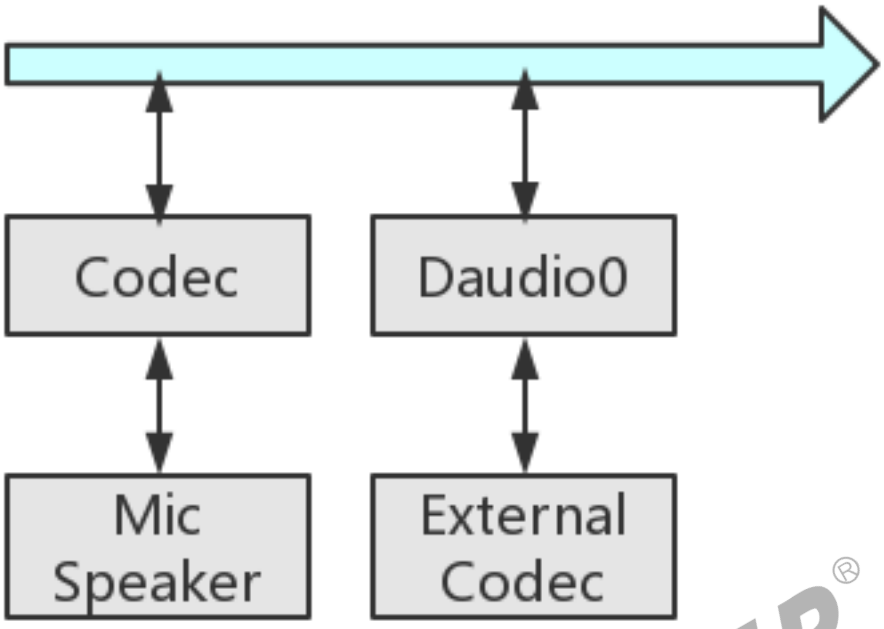


图 2-8: R11 音频硬件框图

2.5.2 时钟源

R11 中，2 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

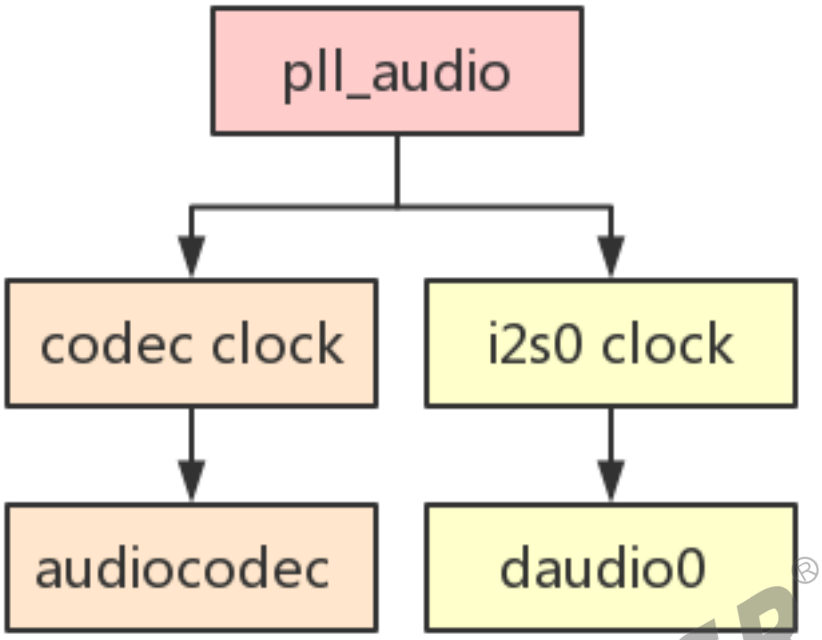


图 2-9: R11 时钟源

2.5.3 代码结构

```
linux-3.4/sound/soc/sunxi/
├── audiocodec
│   ├── sun8iw8_sndcodec_new.c      // codec 驱动
│   ├── sun8iw8_sndcodec.h
│   ├── sunxi_codec.c               // cpu dai 驱动
│   ├── sunxi_sndcodec.c            // codec machine 驱动
│   ├── sunxi_codecdma.c            // codec platform 驱动
│   └── sunxi_codecdma.h
└── daudio0
    ├── snddaudio0.c                // daudio codec 驱动
    ├── sunxi-daudio0.c             // daudio cpu dai 驱动
    ├── sunxi-daudio0.h
    ├── sunxi-daudiodma0.c          // daudio platform 驱动
    ├── sunxi-daudiodma0.h
    └── sunxi-snddaudio0.c          // daudio machine 驱动
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.5.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 一路模拟输出: 一路立体声 LINEOUT 输出 (LINEOUTP, LINEOUTN)
- 一路路模拟输入：MIC1
- 支持同时 playback 和 record(全双工模式)
- 支持 ADC 的 AGC,DRC 功能
- 支持 DAC 的 DRC 功能

2.5.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Audiocodec for the SUNXI chips
         <*>   Audiocodec Machine for codec chips
         <*>   Audiocodec for the SUN8IW8 chips
```

2.5.4.2 sys_config 配置

```
[audio0]
headphone_vol           = 0x3b
lineout_vol             = 0x1a
audio_pa_ctrl           = port:PB05<1><default><default><0>
audio_pa_active_level   = 1
adcagc_used             = 0
adcdrc_used             = 0
dacdrc_used             = 0
adchpf_used             = 0
dachpf_used             = 0
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

audio0 配置	audio0 配置说明
headphone_vol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
lineout_vol	lineout volume，可设定范围 0~0x1f, 0 或者 1 表示 mute, 2~31 表示-43.5dB~0dB, 1.5dB/step
audio_pa_ctrl	PA 使能引脚
audio_pa_active_level	1:high level active; 0:low level active
adcagc_used	1:use adcagc 0:no use
adcdrc_used	1:use adcdrc 0:no use
dacdrc_used	1:use dacdrc 0:no use
adchpf_used	1:use adchpf 0:no use
dachpf_used	1:use dachpf 0:no use

2.5.4.3 codec 数据通路

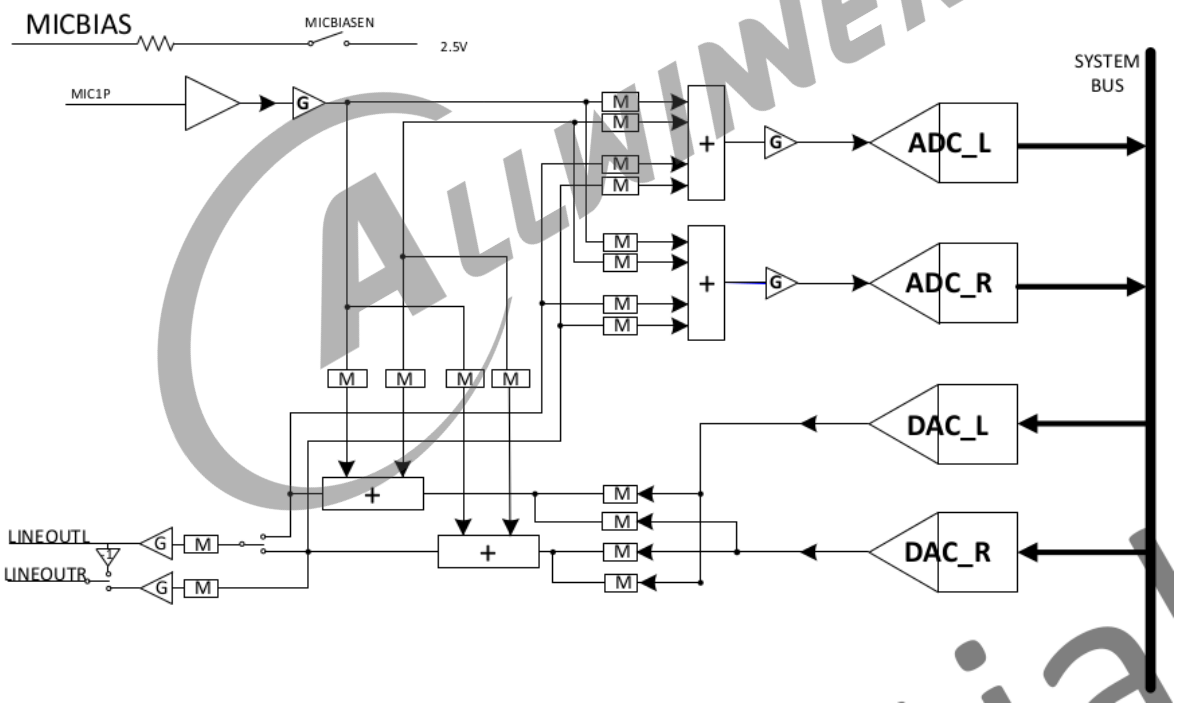


图 2-10: R11 音频通路

```
播歌
DACL --> Left Output Mixer --> LINEOUTL
DACR --> Right Output Mixer --> LINEOUTR

录音
MIC1P --> LADC input Mixer --> ADCL
```

控件名称	功能	数值
Lineout volume	lineout 音量设置	0–31, 表示-43.5–0dB
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB
LADC input Mixer MIC1 boost Switch	LADC input Mixer 设置，使能 MIC 通路	0: 关闭; 1: 开启
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
MIC1 boost AMP gain control	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC1_G boost stage output mixer control	MIC1 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
SPK_L Mux	SPK_L Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
SPK_R Mux	SPK_R Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
digital volume	数字音量设置	0–63, 表示-73.08–0dB

通路设置举例：

1. 播放通路

```
通过lineout播放:
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 0
```

2. 录音通路

```
通过MIC1录音:
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.5.5 Daudio

硬件特性

- 一路 I2S/PCM；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 2 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.5.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   SoC daudio0 tdm interface for SUNXI chips
         <*>   Daudio0 Public Machine for SUNXI chips
```

2.5.5.2 sys_config 配置

```
[tdm0]
daudio_used          = 1
daudio_master            = 4
daudio_select            = 1
audio_format             = 1
signal_inversion         = 1
sample_resolution    = 16
slot_width_select        = 16
pcm_lrck_period          = 32
pcm_lrckr_period         = 1
msb_lsb_first        = 0
sign_extend              = 0
tx_data_mode             = 0
rx_data_mode             = 0
;i2s_mclk             = port:PB08<2><1><default><default>
i2s_bclk             = port:PG11<2><1><default><default>
i2s_lrclk            = port:PG10<2><1><default><default>
i2s_dout0            = port:PG12<2><1><default><default>
i2s_dout1            =
i2s_dout2            =
i2s_dout3            =
i2s_din              = port:PG13<2><1><default><default>
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

tdm0 配置	tdm0 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
daudio_select	0: pcm mode; 1: i2s mode
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
sample_resolution	采样精度,16bit, 24bit,32bit
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	可配置 16/32/64/128/256 个 bclk
pcm_lrckr_period	可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
i2s_bclk	i2s_bclk 引脚
i2s_lrclk	i2s_lrclk 引脚
i2s_dout0	i2s_dout 引脚
i2s_din	i2s_din 引脚

2.5.6 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.5.6.1 播放

```
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Lineout volume' 24
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 0

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='Lineout volume' 50
```

2.5.6.2 录音

表示下使用 AudioCodec 进行单声道录音

```
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 1 /tmp/test.wav
```

2.6 R16 音频接口

2.6.1 硬件资源

R16 包含 3 个音频模块，分别是内置 AudioCodec,I2S0 以及 I2S1。

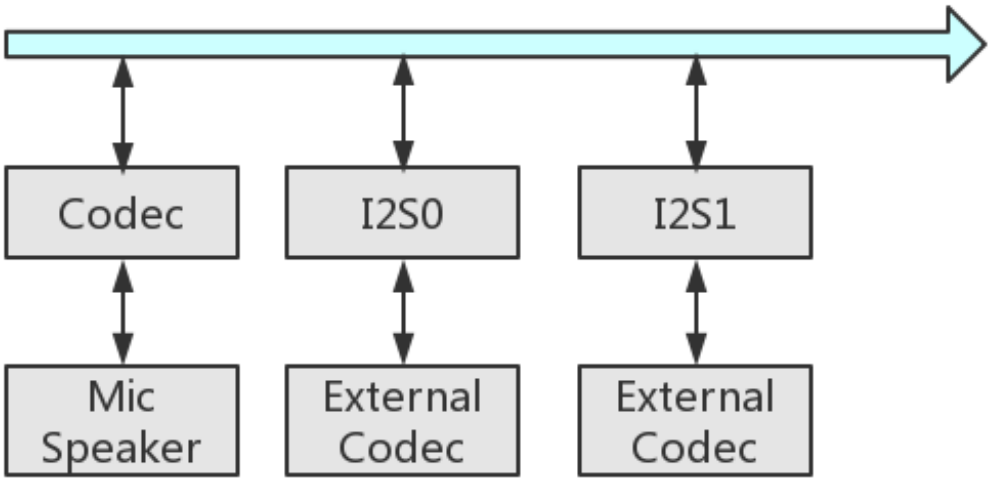


图 2-11：R16 音频硬件框图

2.6.2 时钟源

R16 中，3 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

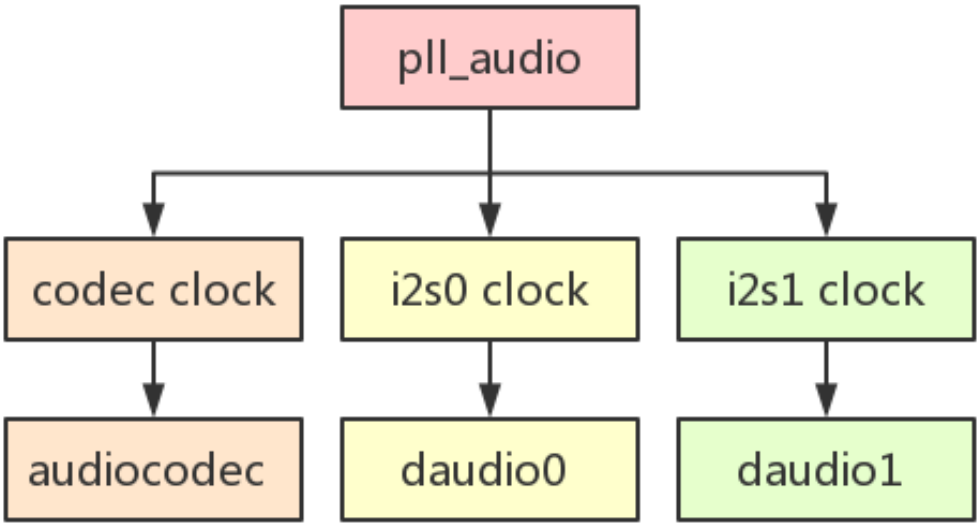


图 2-12: R16 时钟源

2.6.3 代码结构

```
linux-3.4/sound/soc/sunxi/
├── audiocodec
│   ├── sun8iw5_machine.c              // codec machine驱动
│   ├── sun8iw5_sndcodec.c             // codec 驱动
│   ├── sun8iw5_sndcodec.h
│   ├── sunxi_codecdma.c               // codec platform驱动
│   ├── sunxi_codecdma.h
│   └── sunxi_codec.c                  // cpu dai驱动
├── i2s0
│   ├── sndi2s0.c                      // i2s codec 驱动
│   ├── sunxi-i2s0dma.c                // i2s platform驱动
│   ├── sunxi-i2s0dma.h
│   ├── sunxi-i2s0.c                   // i2s cpu dai驱动
│   ├── sunxi-i2s0.h
│   └── sunxi-sndi2s0.c                // i2s machine驱动
└── i2s1
    ├── sndi2s1.c                      // i2s codec 驱动
    ├── sunxi-i2s1dma.c                // i2s platform驱动
    ├── sunxi-i2s1dma.h
    ├── sunxi-i2s1.c                   // i2s cpu dai驱动
    ├── sunxi-i2s1.h
    └── sunxi-sndi2s1.c                // i2s machine驱动
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.6.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出:
  - 一路立体声 headphone 输出 (HPOUTL,HPOUTR)
  - 一路立体声 phoneout 输出 (PHONEOUTP,PHONEOUTN)
- 四路路模拟输入：MIC1,MIC2,linein,phonein
- 支持 headphone 驱动
- 支持同时 playback 和 record(全双工模式)

2.6.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Audiocodec for the SUNXI chips
         <*>   Audiocodec Machine for sun8iw5 chips
         <*>   Audiocodec for the SUN8IW5 chips
```

2.6.4.2 sys_config 配置

```
[audio0]
audio_used              = 1
headphone_vol           = 0x3b
pa_double_used          = 1
headphone_direct_used   = 1
headset_mic_vol         = 3
main_mic_vol            = 1
;audio_linein_detect    = port:PB07<0><default><default><0>
audio_pa_ctrl           = port:PD11<1><default><default><0>
pa_gpio_reverse         = 0
aif2_used               = 0
aif3_used               = 0
headphone_mute_used     = 0
aif1_lrlk_div           = 0x40
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

audio0 配置	audio0 配置说明
audio0	是否使用 audiocodec 驱动。0：不使用；1：使用
headphone_vol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
pa_double_used	是否同时使用两个 DAC，0：不使用；1：使用
headphone_direct_used	是否使用 headphone 输出，0：不使用；1：使用
main_mic_vol	MIC1 默认增益, 0–7, 0:0dB, 1~7:24–42dB,3dB/step
headset_mic_vol	MIC2 默认增益, 0–7, 0:0dB, 1~7:24–42dB,3dB/step
audio_pa_ctrl	PA 使能引脚
pa_gpio_reverse	PA 使能引脚是否颠倒, 0: 正常, 即 high level active; 1: 颠倒, 即 low level active
aif1_lrlk_div	aif1 的 lrck 分频系数

2.6.4.3 codec 数据通路

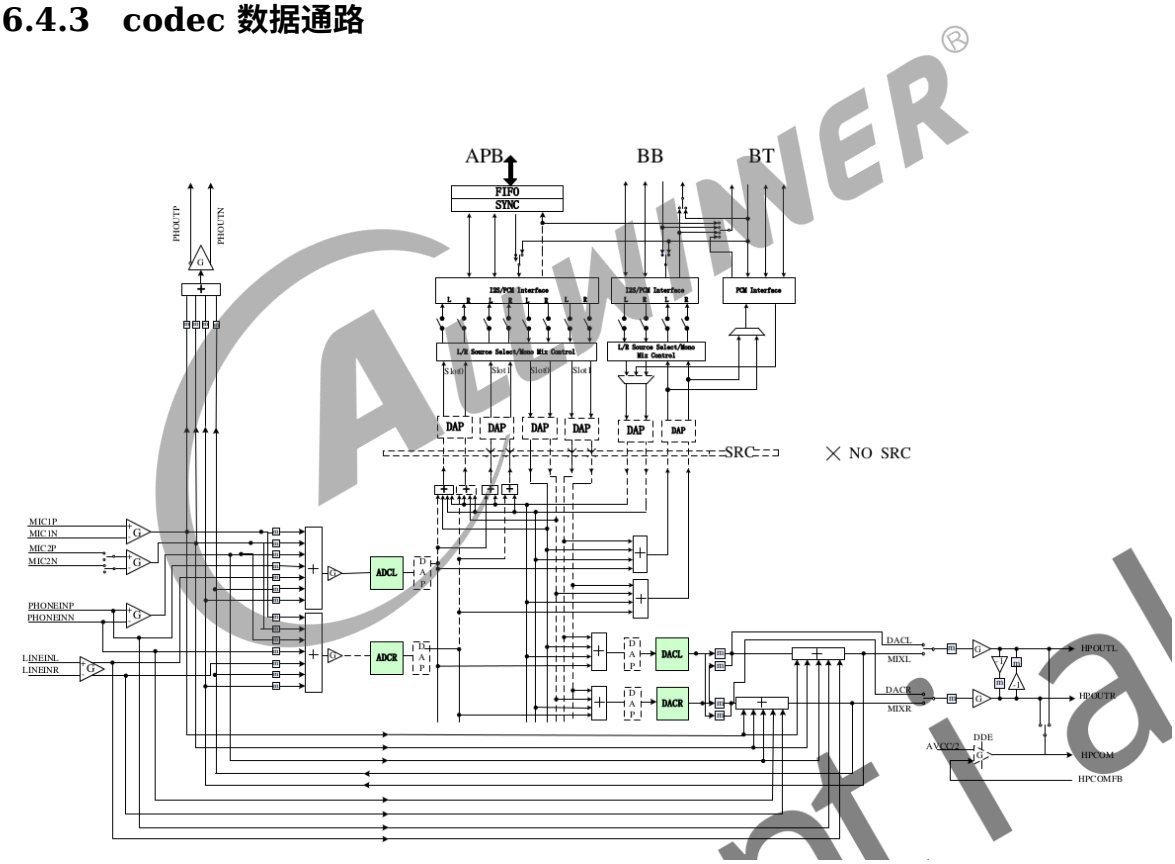


图 2-13: R16 音频通路

```
通过HPOUTL/R播歌

AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> Left Output Mixer  --> HP_L Mux --> HPOUTL
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> Right Output Mixer --> HP_R Mux --> HPOUTR

通过MIC1录音
AIF1ADCL <-- AIF1OUT0L Mux <-- AIF1 AD0L Mixer <-- ADCL Mux <-- LEFT ADC input Mixer <--
```

```
    MIC1 PGA <-- MIC1P/N
```

R16 相关控件如下表：

控件名称	功能	数值
Headphone Switch	Headphone 通路使能	0: 关闭; 1: 开启
ADC input gain	ADC 增益	0–7, 表示-4.5–6dB
ADC volume	ADCL/ADCR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
ADCL Mux	ADCL Mux 设置, 只支持 0:ADC	0:ADC
ADCR Mux	ADCR Mux 设置, 只支持 0:ADC	0:ADC
AIF1 AD0L Mixer ADCL Switch	AIF1 AD0L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF1 DA0L Switch	AIF1 AD0L Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACL Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACR Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer ADCR Switch	AIF1 AD0R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF1 DA0R Switch	AIF1 AD0R Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACL Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACR Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer ADCL Switch	AIF1 AD1L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer AIF2 DACL Switch	AIF1 AD1L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1 AD1R Mixer ADCR Switch	AIF1 AD1R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer AIF2 DACR Switch	AIF1 AD1R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 ADC timeslot 0 mixer gain	AIF1 ADC0L/ADC0R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC0L Mixer, bit0:AIF2 DACR; bit1:ADCL; bit2:AIF2 DACL; bit3:AIF2 DA0L; 对于 ADC0R Mixer, bit0:AIF2 DACL; bit1:ADCR; bit2:AIF2 DACR; bit3:AIF2 DA0R;
AIF1 ADC timeslot 0 volume	AIF1 ADC0L/ADC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 ADC timeslot 1 mixer gain	AIF1 ADC1L/ADC1R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC1L Mixer, bit0:ADCL; bit1:AIF2 DACL; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACR;
AIF1 ADC timeslot 1 volume	AIF1 ADC1L/ADC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 0 volume	AIF1 DAC0L/DAC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 1 volume	AIF1 DAC1L/DAC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1IN0L Mux	AIF1IN0L Mux 设置	0:AIF1_DA0L; 1:AIF1_DA0R; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R
AIF1IN0R Mux	AIF1IN0R Mux 设置	0:AIF1_DA0R; 1:AIF1_DA0L; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R
AIF1IN1L Mux	AIF1IN1L Mux 设置	0:AIF1_DA1L; 1:AIF1_DA1R; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1IN1R Mux	AIF1IN1R Mux 设置	0:AIF1_DA1R; 1:AIF1_DA1L; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1OUT0L Mux	AIF1OUT0L Mux 设置	0:AIF1_AD0L; 1:AIF1_AD0R; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT0R Mux	AIF1OUT0R Mux 设置	0:AIF1_AD0R; 1:AIF1_AD0L; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT1L Mux	AIF1OUT1L Mux 设置	0:AIF1_AD1L; 1:AIF1_AD1R; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
AIF1OUT1R Mux	AIF1OUT1R Mux 设置	0:AIF1_AD1R; 1:AIF1_AD1L; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
DAC mixer gain	DAC mixer 增益	0:0dB; 1:-6dB; 对于 DACL Mixer, bit0:ADCL; bit1:AIF2 DACL; bit2:AIF1 DAC1L; bit3:AIF1 DAC0L; 对于 DACR Mixer, bit0:ADCR; bit1:AIF2 DACR; bit2:AIF1 DAC1R; bit3:AIF1 DAC0R;
DAC volume	DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
DACL Mixer ADCL Switch	DACL Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
DACL Mixer AIF1DA0L Switch	DACL Mixer 设置，使能 AIF1DA0L 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA1L Switch	DACL Mixer 设置，使能 AIF1DA1L 通路	0: 关闭; 1: 开启
DACL Mixer AIF2DACL Switch	DACL Mixer 设置，使能 AIF2DACL 通路	0: 关闭; 1: 开启
DACR Mixer ADCR Switch	DACR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA0R Switch	DACR Mixer 设置，使能 AIF1DA0R 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA1R Switch	DACR Mixer 设置，使能 AIF1DA1R 通路	0: 关闭; 1: 开启
DACR Mixer AIF2DACR Switch	DACR Mixer 设置，使能 AIF2DACR 通路	0: 关闭; 1: 开启
External Speaker Switch	使能 Headphone 以及 PA	0: 关闭; 1: 开启
HP_L Mux	HP_L Mux 设置	0:DACL ; 1:Left Output Mixer
HP_R Mux	HP_R Mux 设置	0:DACR ; 1:Right Output Mixer
LEFT ADC input Mixer Lout_Mixer_Switch	LEFT ADC input Mixer 设置，使能 Lout Mixer 通路	0: 关闭; 1: 开启
LEFT ADC input Mixer MIC1 boost Switch	LEFT ADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LEFT ADC input Mixer MIC2 boost Switch	LEFT ADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LEFT ADC input Mixer Rout_Mixer_Switch	LEFT ADC input Mixer 设置，使能 Rout Mixer 通路	0: 关闭; 1: 开启
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
MIC1 boost amplifier gain	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost amplifier gain	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
RIGHT ADC input Mixer Lout_Mixer_Switch	RIGHT ADC input Mixer 设置，使能 Lout Mixer 通路	0: 关闭; 1: 开启
RIGHT ADC input Mixer MIC1 boost Switch	RIGHT ADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RIGHT ADC input Mixer MIC2 boost Switch	RIGHT ADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RIGHT ADC input Mixer Rout_Mixer_Switch	RIGHT ADC input Mixer 设置，使能 Rout Mixer 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

2.6.5 Daudio

硬件特性

- 两路 I2S/PCM；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，支持 8 通道输出和 2 通道输入
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.6.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   SoC i2s0 interface for SUNXI chips
         <*>   SoC i2s1 interface for SUNXI chips
```

2.6.5.2 sys_config 配置

I2S0,I2S1 的配置方法是一样的，下面仅描述 I2S0

```
[i2s0]
i1s0_used           = 1
i2s0_channel        = 2
i2s0_master                     = 4
i2s0_select            = 1
audio_format           = 1
signal_inversion       = 1
over_sample_rate       = 512
sample_resolution   = 16
word_select_size    = 32
pcm_sync_period     = 256
msb_lsb_first       = 0
slot_index          = 0
slot_width          = 16
frame_width         = 1
tx_data_mode           = 1
rx_data_mode           = 1
i2s0_mclk            =
i2s0_bclk            = port:PB05<2><1><default><default>
i2s0_lrclk           = port:PB04<2><1><default><default>
i2s0_dout0           = port:PB06<2><1><default><default>
i2s0_dout1           =
i2s0_dout2           =
i2s0_dout3           =
i2s0_din             = port:PB07<2><1><default><default>
```

i2s0 配置	i2s0 配置说明
i2s0_used	是否使用 i2s 驱动。0：不使用；1：使用
i2s0_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
over_sample_rate	支持 128fs/192fs/256fs/384fs/512fs/768fs
sample_resolution	采样精度,16bit, 24bit,32bit
word_select_size	支持 16bits/20bits/24bits/32bits
pcm_sync_period	16/32/64/128/256
msb_lsb_first	0: msb first; 1: lsb first
slot_index	0: 1st slot; 1: 2nd slot; 2: 3th slot; 3:4th slot
slot_width	8: 8 clocks width; 16: 16 clocks width
frame_width	0: long frame sync; 1: short frame sync
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
i2s0_mclk	i2s0_mclk 引脚
i2s0_bclk	i2s0_bclk 引脚
i2s0_lrclk	i2s0_lrclk 引脚
i2s0_dout0	i2s0_dout 引脚
i2s0_din	i2s0_din 引脚

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.6.6 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

2.6.6.1 播放

```
通过speaker播放
amixer cset name='AIF1IN0L Mux' 'AIF1_DA0L';
amixer cset name='AIF1IN0R Mux' 'AIF1_DA0R';
amixer cset name='DACL Mixer AIF1DA0L Switch' 1;
amixer cset name='DACR Mixer AIF1DA0R Switch' 1;
amixer cset name='HP_L Mux' 'DACL HPL Switch' ;
amixer cset name='HP_R Mux' 'DACR HPR Switch';
amixer cset name='External Speaker Switch' 1;

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

```
mixer cset name='AIF1IN0L Mux' 'AIF1_DA0L';
amixer cset name='AIF1IN0R Mux' 'AIF1_DA0R';
amixer cset name='DACL Mixer AIF1DA0L Switch' 1;
amixer cset name='DACR Mixer AIF1DA0R Switch' 1;
amixer cset name='HP_L Mux' 'DACL HPL Switch' ;
amixer cset name='HP_R Mux' 'DACR HPR Switch';
amixer cset name='Headphone Switch' 1;

aplay -Dhw:sndcodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:sndcodec cset name='headphone volume' 58
```

2.6.6.2 录音

表示下使用 audiocodec 进行单声道录音

```
amixer cset name='LEFT ADC input Mixer MIC1 boost Switch' 1
amixer cset name='AIF1 AD0L Mixer ADCL Switch' 1
amixer cset name='AIF1OUT0L Mux' 'AIF1_AD0L'
amixer cset name='MIC1 boost amplifier gain' 4

arecord -Dhw:sndcodec -f S16_LE -r 16000 -c 1 /tmp/test.wav
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.7 R18 音频接口

2.7.1 硬件资源

R18 包含 4 个音频模块，分别是内置 AudioCodec 以及 Daudio0,Daudio1,Daudio2。

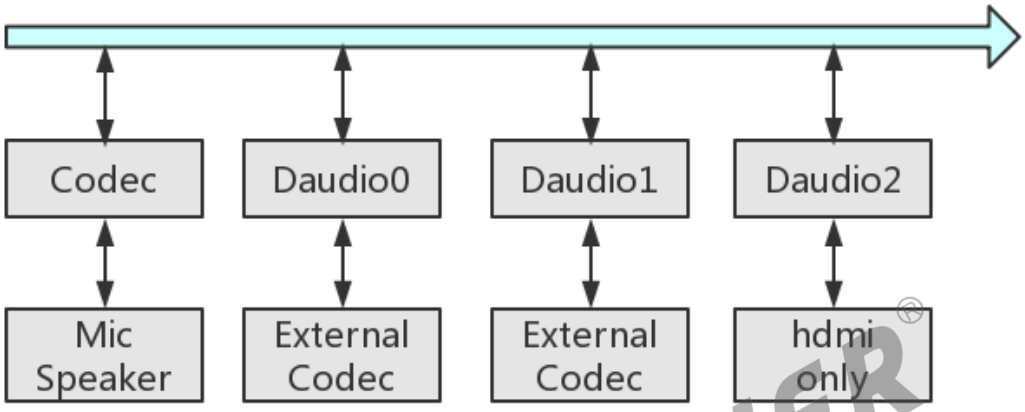


图 2-14: R18 音频硬件框图

2.7.2 时钟源

R18 中，4 个音频模块的时钟源均来自 pll_audio

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

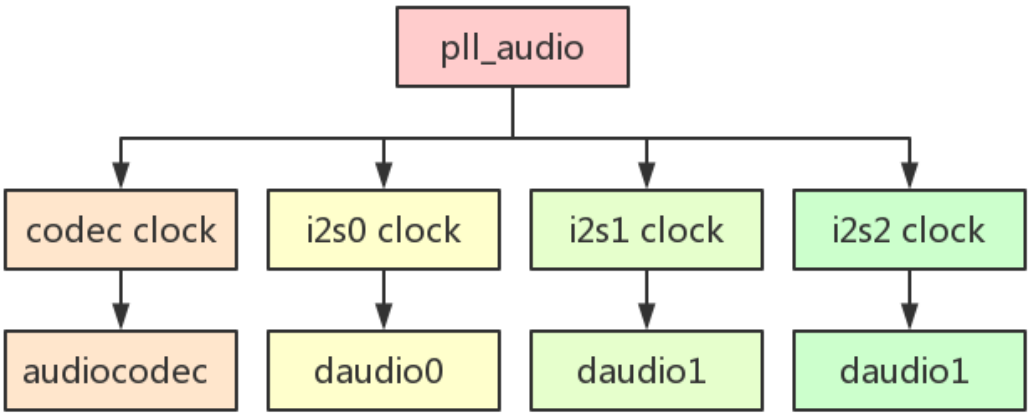


图 2-15: R18 时钟源

2.7.3 代码结构

```
linux-4.4/sound/soc/sunxi/
├── sun50iw1-codec.c                    // codec驱动
├── sun50iw1-codec.h
├── sun50iw1-sndcodec.c                 // codec machine驱动
├── sunxi-inter-i2s.c                   // codec platform驱动
├── sunxi-daudio.c                      // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-snddaudio.c                   // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-pcm.c                         // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi_rw_func.c                     // 通用文件，读写模拟/数字寄存器的接口
├── sunxi_rw_func.h
├── spdif-utils.c                       // spdif codec驱动
├── sunxi-sndspdif.c                    // spdif machine驱动
├── sunxi-spdif.c                       // spdif platform驱动
├── sunxi-spdif.h
├── sunxi-hdmi.c                        // hdmi codec驱动
└── sunxi-sndhdmi.c                     // hdmi machine驱动


hdmi platform模型使用的是sunxi-daudio.c
linux-4.4/sound/soc/soc-utils.c         // snd-soc-dummmy驱动,可用于daudio codec模型
linux-4.4/sound/soc/codecs/ac108.c      // ac108 codec驱动
linux-4.4/sound/soc/codecs/tas5731.c    // tas5731数字功放 codec驱动
```

2.7.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 四路模拟输出:
  - 一路立体声 earpiece 输出 (EAROUTP,EAROUTN)
  - 一路立体声 phoneout 输出 (PHONEOUTP,PHONEOUTN)
  - 一路立体声 headphone 输出 (HPOUTL,HPOUTR)
  - 一路立体声 lineout 输出 (LINEOUTL,LINEOUTR)
- 四路路模拟输入：MIC1,MIC2,linein,phonein
- 支持 headphone 驱动
- 支持 earpiece 驱动
- 支持同时 playback 和 record(全双工模式)
- 支持适用于 DAC 的 DRC 功能
- 支持适用于 ADC 的 AGC,DRC 功能

2.7.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun50iw1 Codec Support
```

2.7.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used    = 0x1
aif2fmt          = 0x3
aif3fmt          = 0x3
aif2master       = 0x1
linein_detect    = port:PH05<6><default><default><default>
hp_detect_case   = 0x1
;-------------------------------------------------------------------------------
[i2s]
i2s_used = 0x1
;-------------------------------------------------------------------------------
[codec]
codec_used = 0x1
headphonevol =   0x38
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
spkervol =   0x1d
earpiecevol =   0x1e
maingain =   0x4
headsetmicgain =  0x4
adcagc_cfg =   0x0
adcdrc_cfg =   0x0
adchpf_cfg =   0x1
dacdrc_cfg =   0x0
dachpf_cfg =   0x0
aif2config =  0x0
aif3config =  0x0
aif1_lrlk_div = 0x40
aif2_lrlk_div = 0x40
pa_sleep_time = 0x0a
dac_digital_vol = 0x9898
gpio-spk =
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用
linein_detect	linein 检测引脚
hp_detect_case	jack irq level, 0:low; 1:high

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
headphonevol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
spkervol	spk(lineout) volume, 可设定范围 0~0x1f, 0 或者 1 表示 mute, 2~31 表示-43.5dB~0dB, 1.5dB/step
earpiecevol	earpiece volume, 可设定范围 0~0x1f, 0 或者 1 表示 mute, 2~31 表示-43.5dB~0dB, 1.5dB/step
maingain	MIC1 默认增益, 0–7, 0:0dB, 1~7:24–42dB,3dB/step
headsetmicgain	MIC2 默认增益, 0–7, 0:0dB, 1~7:24–42dB,3dB/step
adcagc_cfg	是否使用 adcagc. 0: 不适用；1：使用
adcdrc_cfg	是否使用 adcdrc. 0: 不适用；1：使用
adchpf_cfg	是否使用 adchpf. 0: 不适用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不适用；1：使用
dachpf_cfg	是否使用 dachpf. 0: 不适用；1：使用
aif1_lrlk_div	aif1 的 lrck 分频系数
pa_sleep_time	使能 pa 之前等待的时间，单位 ms
dac_digital_vol	DACL/DACR 数字音量,0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB, 0x98 表示-6dB

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
gpio-spk	PA 使能引脚

2.7.4.3 codec 数据通路

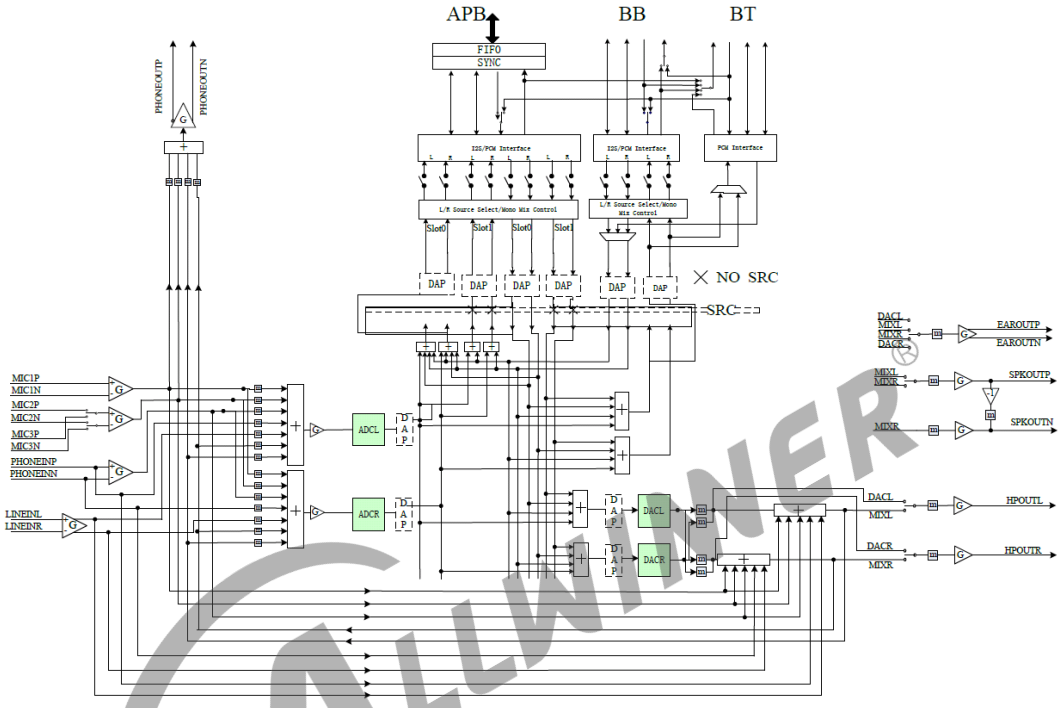


图 2-16: R18 音频通路

```
通过HPOUTL/R播歌

AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> HP_L Mux --> HPOUTL
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> HP_R Mux --> HPOUTR

通过MIC1,2录音
AIF1ADCL <-- AIF1OUT0L Mux <-- AIF1 AD0L Mixer <-- ADCL Mux <-- LADC input Mixer <-- MIC1
    PGA <-- MIC1P/N
AIF1ADCR <-- AIF1OUT0R Mux <-- AIF1 AD0R Mixer <-- ADCR Mux <-- RADC input Mixer <-- MIC2
    PGA <-- MIC2P/N
```

R18 相关控件如下表:

控件名称	功能	数值
Headphone Switch	Headphone 通路使能	0: 关闭; 1: 开启
Linein_detect Switch	Linein 检测使能	0: 关闭; 1: 开启
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB

控件名称	功能	数值
ADC volume	ADCL/ADCR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
ADCL Mux	ADCL Mux 设置, 只支持 0:ADC	0:ADC
ADCR Mux	ADCR Mux 设置, 只支持 0:ADC	0:ADC
AIF1 AD0L Mixer ADCL Switch	AIF1 AD0L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF1 DA0L Switch	AIF1 AD0L Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACL Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACR Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer ADCR Switch	AIF1 AD0R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF1 DA0R Switch	AIF1 AD0R Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACL Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACR Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer ADCL Switch	AIF1 AD1L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer AIF2 DACL Switch	AIF1 AD1L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer ADCR Switch	AIF1 AD1R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer AIF2 DACR Switch	AIF1 AD1R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1 ADC timeslot 0 mixer gain	AIF1 ADC0L/ADC0R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC0L Mixer, bit0:AIF2 DACR; bit1:ADCL; bit2:AIF2 DACL; bit3:AIF2 DA0L; 对于 ADC0R Mixer, bit0:AIF2 DACL; bit1:ADCR; bit2:AIF2 DACR; bit3:AIF2 DA0R;
AIF1 ADC timeslot 0 volume	AIF1 ADC0L/ADC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 ADC timeslot 1 mixer gain	AIF1 ADC1L/ADC1R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC1L Mixer, bit0:ADCL; bit1:AIF2 DACL; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACR;
AIF1 ADC timeslot 1 volume	AIF1 ADC1L/ADC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 0 volume	AIF1 DAC0L/DAC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 1 volume	AIF1 DAC1L/DAC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1IN0L Mux	AIF1IN0L Mux 设置	0:AIF1_DA0L; 1:AIF1_DA0R; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R
AIF1IN0R Mux	AIF1IN0R Mux 设置	0:AIF1_DA0R; 1:AIF1_DA0L; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1IN1L Mux	AIF1IN1L Mux 设置	0:AIF1_DA1L; 1:AIF1_DA1R; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1IN1R Mux	AIF1IN1R Mux 设置	0:AIF1_DA1R; 1:AIF1_DA1L; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1OUT0L Mux	AIF1OUT0L Mux 设置	0:AIF1_AD0L; 1:AIF1_AD0R; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT0R Mux	AIF1OUT0R Mux 设置	0:AIF1_AD0R; 1:AIF1_AD0L; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT1L Mux	AIF1OUT1L Mux 设置	0:AIF1_AD1L; 1:AIF1_AD1R; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
AIF1OUT1R Mux	AIF1OUT1R Mux 设置	0:AIF1_AD1R; 1:AIF1_AD1L; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
DAC mixer gain	DAC mixer 增益	0:0dB; 1:-6dB; 对于 DACL Mixer, bit0:ADCL; bit1:AIF2 DACL; bit2:AIF1 DAC1L; bit3:AIF1 DAC0L; 对于 DACR Mixer, bit0:ADCR; bit1:AIF2 DACR; bit2:AIF1 DAC1R; bit3:AIF1 DAC0R;
DAC volume	DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
DACL Mixer ADCL Switch	DACL Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA0L Switch	DACL Mixer 设置，使能 AIF1DA0L 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA1L Switch	DACL Mixer 设置，使能 AIF1DA1L 通路	0: 关闭; 1: 开启
DACL Mixer AIF2DACL Switch	DACL Mixer 设置，使能 AIF2DACL 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
DACR Mixer ADCR Switch	DACR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA0R Switch	DACR Mixer 设置，使能 AIF1DA0R 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA1R Switch	DACR Mixer 设置，使能 AIF1DA1R 通路	0: 关闭; 1: 开启
DACR Mixer AIF2DACR Switch	DACR Mixer 设置，使能 AIF2DACR 通路	0: 关闭; 1: 开启
EAR Mux	EAR Mux 设置	0:DACR; 1:DACL; 2:Right Analog Mixer; 3:Left Analog Mixer
Earpiece Switch	Earpiece 通路使能	0: 关闭; 1: 开启
External Speaker Switch	使能 Headphone 以及 PA	0: 关闭; 1: 开启
HP_L Mux	HP_L Mux 设置	0:DACL ; 1:Left Output Mixer
HP_R Mux	HP_R Mux 设置	0:DACR ; 1:Right Output Mixer
LADC input Mixer LINEINL	LADC input Mixer 设置，使能 LINEINL 通路	0: 关闭; 1: 开启
LADC input Mixer MIC1 boost Switch	LADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LADC input Mixer MIC2 boost Switch	LADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LADC input Mixer l_output mixer Switch	LADC input Mixer 设置，使能 l_output mixer 通路	0: 关闭; 1: 开启
LADC input Mixer r_output mixer Switch	LADC input Mixer 设置，使能 r_output mixer 通路	0: 关闭; 1: 开启
LINEINL/R to L_R output mixer gain		
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer LINEINL Switch	Left Output Mixer 设置，使能 LINEINL 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
MIC1 boost amplifier gain	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC1_G boost stage output mixer control	MIC1 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 BST stage to L_R outp mixer gain	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
RADC input Mixer LINEINR Switch	RADC input Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
RADC input Mixer MIC1 boost Switch	RADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 l_output mixer 通路	0: 关闭; 1: 开启
RADC input Mixer l_output Switch	RADC input Mixer 设置，使能 l_output mixer 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer LINEINR Switch	Right Output Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
SPK_L Mux	SPK_L Mux 设置	0:MIXEL Switch; 1:MIXL MIXR Switch
SPK_R Mux	SPK_R Mux 设置	0:MIXER Switch; 1:MIXR MIXL Switch
digital volume	数字音量设置	0–63, 表示-73.08–0dB
earpiece volume	earpiece 音量设置	0–31, 表示-43.5–0dB

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB
speaker volume	speaker(lineout) 音量设置	0–31, 表示-43.5–0dB

2.7.5 Daudio

硬件特性

- 三路 I2S/PCM;
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.7.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
```

2.7.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used = 1
;-----------------------------------------------------------------------------
[daudio0]
daudio0_used = 1
pcm_lrck_period =   0x60
pcm_lrckr_period =   0x01
slot_width_select =   0x18
pcm_lsb_first =   0x0
tx_data_mode =   0x0
rx_data_mode =   0x0
daudio_master =   0x04
audio_format =   0x01
signal_inversion =   0x01
```

```
frametype =   0x0
tdm_config =   0x01
clk_active = 0x0
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
frametype	0: short frame = 1 clock width; 1: long frame = 2 clock width
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

版权所有 © 珠海全志科技股份有限公司。保留一切权利

daudio 配置	daudio 配置说明
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

具体 Daudio 外接 codec, 数字功放的配置，可参考《R18 外挂 codec:ac108》《R18 外挂数字功放 TAS5731》

2.7.6 SPDIF

硬件特性

- 支持 S/PDIF_OUT
- 支持 mono 和 stereo 模式
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 支持 16bit,24bit 采样精度

2.7.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner SPDIF Support
```

2.7.6.2 sys_config 配置

```
[spdif]
spdif_used          = 0
[sndspdif]
sndspdif_used       = 0
```

spdif 配置，即 platform 驱动的相关配置

spdif 配置	spdif 配置说明
spdif_used	是否使用 spdif 驱动。0：不使用；1：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

sndspdif 配置，即 machine 驱动的相关配置

sndspdif 配置	sndspdif 配置说明
sndspdif_used	是否使用 sndspdif 驱动。0：不使用；1：使用

- sys_config 中不需要配置 codec 驱动相关信息

因为 machine 驱动代码中默认配置了"spdif-utils" 作为 codec 驱动, 代码路径:

```
linux-4.4/sound/soc/sunxi/spdif-utils.c
```

2.7.7 外挂 codec:AC108

R18 标案 tulip-noma 搭配了 MIC 子板，含有两片 AC108, 每片最高可录 4 通道

下面对 R18 如何配置使用 AC108 作简单介绍

2.7.7.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
           <*> Allwinner Digital Audio Support
           CODEC drivers  --->
           <*> Sunxi AC108 Codec
```

2.7.7.2 sys_config&dts 配置

R18 通过 twi1 控制 AC108, 而 i2s0 用于音频数据的传输

twi 部分配置, 可通过 dts 进行配置:

```
twi1: twi@0x01c2b000 {
    status = "okay";
    ac108@35{
            compatible = "Allwinnertech,MicArray_1";
            debug_mode = <0>;
            pga_gain = <0x32>;
            ref_pga_gain = <0x08>;
            ref_chip_addr = <0x3b>;
            ref_channel_num = <0x2>;
            pa_double_used = <0x1>;
```

```
            codec_mic_used = <0x0>;
            gpio-power = <&r_pio PL 12 1 1 1 1>;
            twi_bus = <1>;
            voltage_enable = "nocare";
            power_vol = <0x0>;
            slot_width = <0x18>;
            reg = <0x35>;
    };
    ac108@3b{
            compatible = "Allwinnertech,MicArray_0";
            reg = <0x3b>;
            debug_mode = <0>;
            pga_gain = <0x32>;
            ref_pga_gain = <0x08>;
            ref_chip_addr = <0x3b>;
            ref_channel_num = <0x2>;
            pa_double_used = <0x1>;
            codec_mic_used = <0x0>;
            twi_bus = <1>;
            voltage_enable = "nocare";
            gpio-power = <&r_pio PL 12 1 1 1 1>;
            power_vol = <0x0>;
            slot_width = <0x18>;
    };
};
```

I2S 部分需要配置 sys_config 以及 dts

- sys_config 部分主要涉及 i2s 相关格式，需要根据 AC108 spec 进行配置，sdk 默认 daudio0 配置可正常运行 AC108
- dts 部分主要需要指定 ASOC codec 以及 codec-dai 驱动的名称，如

```
snddaudio0:sound@1 {
    sunxi,snddaudio-codec = "ac108.1-0035";
    sunxi,snddaudio-codec-dai = "ac108-pcm1";
};
```

2.7.7.3 使用

进入系统后，通过命令 cat /proc/asound/cards 列出当前声卡信息，如果发现 ac108 相关声卡，说明已经正常加载驱动

无需额外设置音频通路，可直接用下面命令进行录音:

```
arecord -Dhw:sndac10810035 -f S16_LE -r 16000 -c 8 /tmp/test.wav
```

2.7.8 外挂数字功放 TAS5731

R18 标案 tulip-noma 搭配了一片数字功放 TAS5731

下面对 R18 如何配置使用 tas5731 作简单介绍

2.7.8.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
         CODEC drivers  --->
         <*> TAS5731 PA
```

2.7.8.2 sys_config&dts 配置

R18 通过 TWI0 控制数字功放, 而 I2S1 用于音频数据的传输

twi 部分配置, 可通过 dts 进行配置:

```
twi0: twi@0x01c2ac00 {
    status = "okay";
    tas5731-codec@1b{
        compatible = "Allwinnertech,tas5731_PA";
        tas5731_power = <&pio PH 8 1 1 1 1>;
        tas5731_reset = <&pio PB 2 1 1 1 1>;
        amp_poweren = <&r_pio PL 7 1 1 1 1>;
        regulator_name = "vcc-amp";
        reg = <0x1b>;
    };
};
```

I2S 部分需要配置 sys_config 以及 dts

- sys_config 部分主要涉及 i2s 相关格式，需要根据具体数字功放进行配置，sdk 默认 daudio1 配置可正常运行 tas5731
- dts 部分主要需要指定 ASOC codec 以及 codec-dai 驱动的名称，如

```
snddaudio1:sound@2 {
    sunxi,snddaudio-codec = "tas5731-codec.0-001b";
    sunxi,snddaudio-codec-dai = "tas5731_audio";
};
```

2.7.8.3 使用

进入系统后, 通过命令 cat /proc/asound/cards 列出当前声卡信息，如果发现 tas5731 相关声卡，说明已经正常加载驱动

版权所有 © 珠海全志科技股份有限公司。保留一切权利

无需额外设置音频通路，可直接用下面命令进行播歌:

```
aplay -Dhw:sndtas5731codec /mnt/UDISK/16000-stere-10s.wav
```

2.7.9 HDMI 音频接口

R18 使用 I2S2 将音频数据传输到 HDMI 模块，并且 I2S2 也只能用于 HDMI。

2.7.9.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
           <*> Allwinner HDMI Audio Support
           <*> Allwinner Digital Audio Support
```

2.7.9.2 sys_config 配置

```
[daudio2]
daudio2_used        = 1
[sndhdmi]
sndhdmi_used        = 1
```

daudio2 配置，即 daudio2 platform 驱动的相关配置

daudio2 配置	daudio2 配置说明
daudio2_used	是否使用 daudio2 驱动。0：不使用；1：使用

sndhdmi 配置，即 sndhdmi machine 驱动的相关配置

sndhdmi 配置	sndhdmi 配置说明
sndhdmi_used	否使用 sndhdmi 驱动。0：不使用；1：使用 是

2.7.10 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

2.7.10.1 播放

```
amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L'
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R'
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1
amixer -Dhw:audiocodec cset name='HP_R Mux' 'DACR HPR Switch'
amixer -Dhw:audiocodec cset name='HP_L Mux' 'DACL HPL Switch'
amixer -Dhw:audiocodec cset name='Headphone Switch' 1

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='headphone volume' 60
```

2.7.10.2 录音

```
amixer -Dhw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -Dhw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD0L Mixer ADCL Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD0R Mixer ADCR Switch' 1
amixer -Dhw:audiocodec cset name='AIF1OUT0L Mux' 'AIF1_AD0L'
amixer -Dhw:audiocodec cset name='AIF1OUT0R Mux' 'AIF1_AD0R'

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.8 R30 音频接口

2.8.1 硬件资源

R30 包含 5 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Daudio2 以及 Dmic

版权所有 © 珠海全志科技股份有限公司。保留一切权利

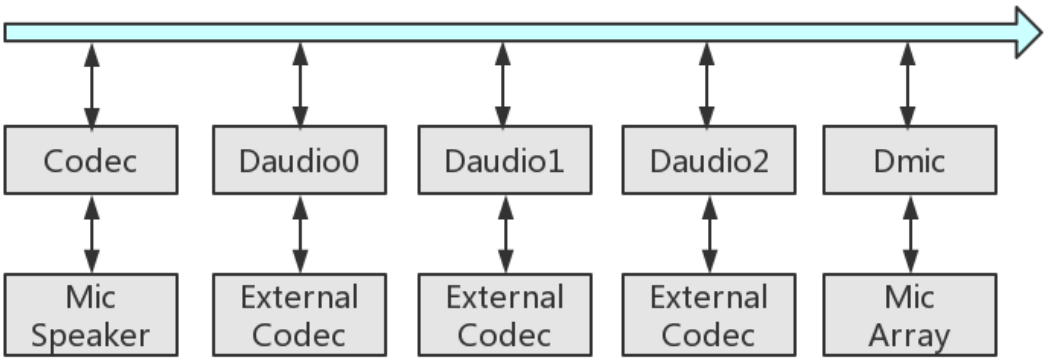


图 2-17: R30 音频硬件框图

2.8.2 时钟源

R30 中，5 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

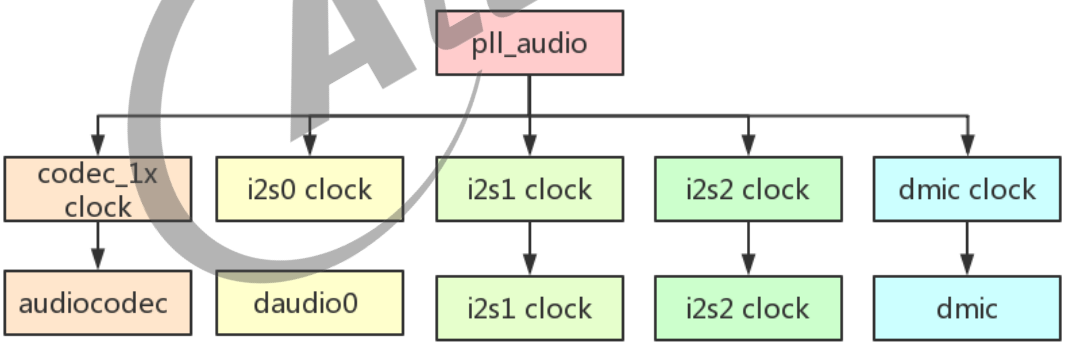


图 2-18: R30 时钟源

2.8.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sun50iw3-codec.c                // codec驱动
├── sun50iw3-codec.h
├── sun50iw3-sndcodec.c             // codec machine驱动
├── sunxi-inter-i2s.c               // codec platform驱动
```

```
├── sunxi-inter-i2s.h
├── sunxi-daudio.c                  // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-dmic.c                    // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-pcm.c                     // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi_rw_func.c                 // 通用文件，读写模拟/数字寄存器的接口
├── sunxi_rw_func.h
├── sunxi-snddaudio.c               // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-snddmic.c                 // dmic machine驱动
└── sunxi-snddmic.h

linux-4.9/sound/soc/codecs/dmic.c   // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c     // daudio codec驱动
```

2.8.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 四路模拟输出:
  - 一路立体声 earpiece 输出 (EAROUTP,EAROUTN)
  - 一路立体声 phoneout 输出 (PHONEOUTP,PHONEOUTN)
  - 一路立体声 headphone 输出 (HPOUTL,HPOUTR)
  - 一路立体声 lineout 输出 (LINEOUTL,LINEOUTR)
- 四路路模拟输入：MIC1,MIC2,linein,phonein
- 支持 headphone 驱动
- 支持 earpiece 驱动
- 支持同时 playback 和 record(全双工模式)
- 支持适用于 DAC 的 DRC 功能
- 支持适用于 ADC 的 AGC,DRC 功能

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.8.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun50iw3 Codec Support
```

2.8.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used = 0x1
aif2fmt = 0x3
aif3fmt = 0x3
aif2master = 0x1
hp_detect_case = 0x0
;-----------------------------------------------------------------------------
[i2s]
i2s_used = 0x1
;-----------------------------------------------------------------------------
[codec]
codec_used = 0x1
headphonevol =   0x3b
spkervol =   0x1b
maingain =   0x4
headsetmicgain =  0x4
adcagc_cfg =   0x0
adcdrc_cfg =   0x0
adchpf_cfg =   0x0
dacdrc_cfg =   0x0
dachpf_cfg =   0x0
aif2config =  0x0
aif3config =  0x0
gpio-spk = port:PB3<1><default><default><0>
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用
aif2fmt	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif3fmt	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format)

版权所有 © 珠海全志科技股份有限公司。保留一切权利

sndcodec 配置	sndcodec 配置说明
	3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif2master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 aif 接口选择 master 模式 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 aif 接口选择 slave 模式
hp_detect_case	jack irq level, 0:low; 1:high

I2S 配置，即 audiocodec platform 驱动的相关配置, 内部 aif 接口用的 I2S(与 I2S0,I2S1 接口无关)

i2s 配置	i2s 配置说明
i2s_used	是否使用 i2s 驱动。0：不使用；1：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
headphonevol	初始化 headphone volume，可设定范围 0~0x3f, 表示 0~-62dB, -1dB/step
spkervol	初始化 speaker volume，可设定范围 0~0x1f, 0 或者 1 表示 mute, 2~31 表示-43.5dB~0dB, 1.5dB/step
headsetmicgain	指的是 MIC2 增益, 可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
adcinputgain	adc 增益，可设定范围 0~0x7, 表示-4.5~6dB, 1.5dB/step, 一般设置 0x3, 即 0dB
adcagc_cfg	是否使用 adcagc. 0: 不使用；1：使用
adcdrc_cfg	是否使用 adcdrc. 0: 不使用；1：使用
adchpf_cfg	是否使用 adchpf. 0: 不使用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不使用；1：使用
aif2config	是否使用 aif2. 0: 不使用；1：使用
aif3config	是否使用 aif3. 0: 不使用；1：使用
gpio-spk	PA 使能引脚

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.8.4.3 codec 数据通路

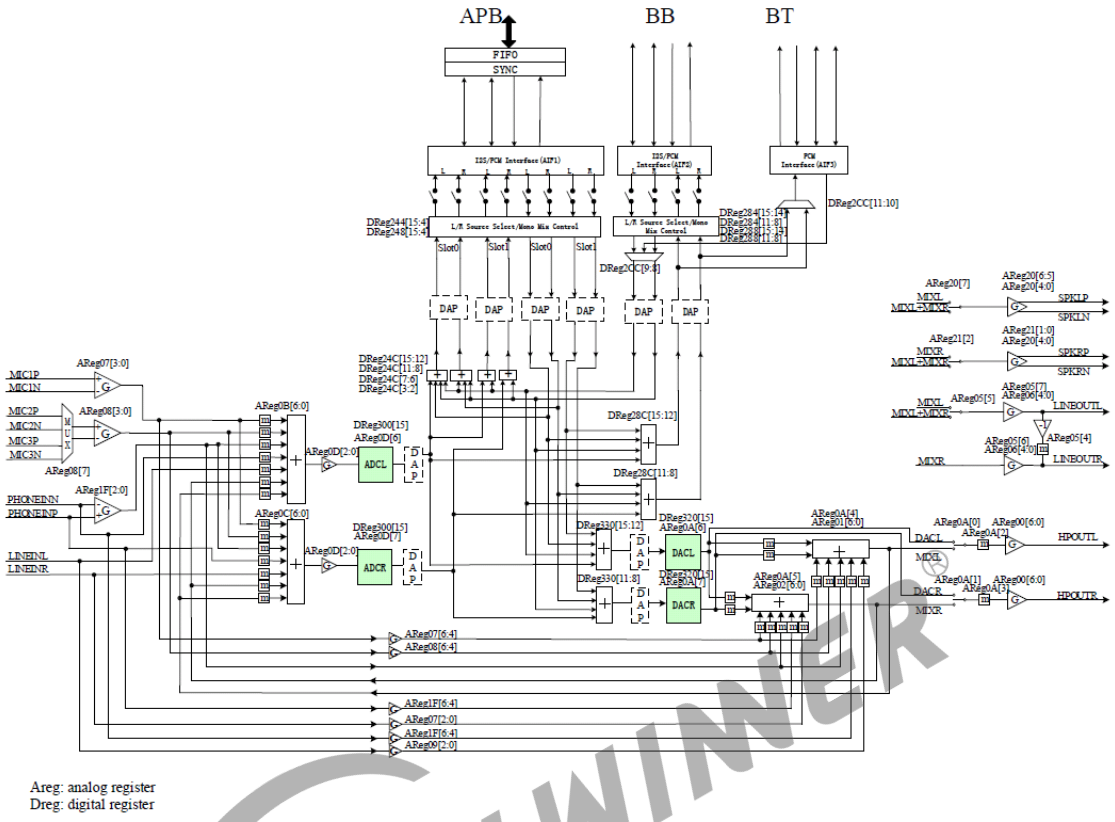


图 2-19: R30 音频通路

```
通过SPKL/R播歌
AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> Left Output Mixer --> SPK_L Mux --> SPKL
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> Right Output Mixer --> SPK_R Mux --> SPKR

通过LINEOUTL/R播歌
AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> Left Output Mixer --> LINEOUTL Mux -->
    LINEOUTL
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> Right Output Mixer --> LINEOUTR Mux -->
    LINEOUTR

通过HPOUTL/R播歌
AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> HP_L Mux --> HPOUTL
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> HP_R Mux --> HPOUTR

通过MIC1,2录音
AIF1ADCL <-- AIF1OUT0L Mux <-- AIF1 AD0L Mixer <-- LADC input Mixer <-- MIC1 PGA <-- MIC1P/
    N
AIF1ADCR <-- AIF1OUT0R Mux <-- AIF1 AD0R Mixer <-- RADC input Mixer <-- MIC2 PGA <-- MIC2P/
    N

通过LINEINL/R录音
AIF1ADCL <-- AIF1OUT0L Mux <-- AIF1 AD0L Mixer <-- LADC input Mixer <-- LINEINN
AIF1ADCR <-- AIF1OUT0R Mux <-- AIF1 AD0R Mixer <-- RADC input Mixer <-- LINEINP
```

R30 相关控件如下表：

控件名称	功能	数值
Headphone Switch	Headphone 通路使能	0: 关闭; 1: 开启
Lineout Switch	Lineout 通路使能	0: 关闭; 1: 开启
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB
ADC volume	ADCL/ADCR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 AD0L Mixer ADCL Switch	AIF1 AD0L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF1 DA0L Switch	AIF1 AD0L Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACL Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACR Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer ADCR Switch	AIF1 AD0R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF1 DA0R Switch	AIF1 AD0R Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACL Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACR Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer ADCL Switch	AIF1 AD1L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer AIF2 DACL Switch	AIF1 AD1L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer ADCR Switch	AIF1 AD1R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer AIF2 DACR Switch	AIF1 AD1R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1 ADC timeslot 0 mixer gain	AIF1 ADC0L/ADC0R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC0L Mixer, bit0:AIF2 DACR; bit1:ADCL; bit2:AIF2 DACL; bit3:AIF2 DA0L; 对于 ADC0R Mixer, bit0:AIF2 DACL; bit1:ADCR; bit2:AIF2 DACR; bit3:AIF2 DA0R;
AIF1 ADC timeslot 0 volume	AIF1 ADC0L/ADC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 ADC timeslot 1 mixer gain	AIF1 ADC1L/ADC1R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC1L Mixer, bit0:ADCL; bit1:AIF2 DACL; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACR;
AIF1 ADC timeslot 1 volume	AIF1 ADC1L/ADC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 0 volume	AIF1 DAC0L/DAC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 1 volume	AIF1 DAC1L/DAC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1IN0L Mux	AIF1IN0L Mux 设置	0:AIF1_DA0L; 1:AIF1_DA0R; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R
AIF1IN0R Mux	AIF1IN0R Mux 设置	0:AIF1_DA0R; 1:AIF1_DA0L; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1IN1L Mux	AIF1IN1L Mux 设置	0:AIF1_DA1L; 1:AIF1_DA1R; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1IN1R Mux	AIF1IN1R Mux 设置	0:AIF1_DA1R; 1:AIF1_DA1L; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1OUT0L Mux	AIF1OUT0L Mux 设置	0:AIF1_AD0L; 1:AIF1_AD0R; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT0R Mux	AIF1OUT0R Mux 设置	0:AIF1_AD0R; 1:AIF1_AD0L; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT1L Mux	AIF1OUT1L Mux 设置	0:AIF1_AD1L; 1:AIF1_AD1R; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
AIF1OUT1R Mux	AIF1OUT1R Mux 设置	0:AIF1_AD1R; 1:AIF1_AD1L; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
DAC mixer gain	DAC mixer 增益	0:0dB; 1:-6dB; 对于 DACL Mixer, bit0:ADCL; bit1:AIF2 DACL; bit2:AIF1 DAC1L; bit3:AIF1 DAC0L; 对于 DACR Mixer, bit0:ADCR; bit1:AIF2 DACR; bit2:AIF1 DAC1R; bit3:AIF1 DAC0R;
DAC volume	DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
DACL Mixer ADCL Switch	DACL Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA0L Switch	DACL Mixer 设置，使能 AIF1DA0L 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA1L Switch	DACL Mixer 设置，使能 AIF1DA1L 通路	0: 关闭; 1: 开启
DACL Mixer AIF2DACL Switch	DACL Mixer 设置，使能 AIF2DACL 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
DACR Mixer ADCR Switch	DACR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA0R Switch	DACR Mixer 设置，使能 AIF1DA0R 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA1R Switch	DACR Mixer 设置，使能 AIF1DA1R 通路	0: 关闭; 1: 开启
DACR Mixer AIF2DACR Switch	DACR Mixer 设置，使能 AIF2DACR 通路	0: 关闭; 1: 开启
External Speaker Switch	使能 Headphone 以及 PA	0: 关闭; 1: 开启
HP_L Mux	HP_L Mux 设置	0:DACL ; 1:Left Output Mixer
HP_R Mux	HP_R Mux 设置	0:DACR ; 1:Right Output Mixer
LADC input Mixer LINEINL	LADC input Mixer 设置，使能 LINEINL 通路	0: 关闭; 1: 开启
LADC input Mixer MIC1 boost Switch	LADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LADC input Mixer MIC2 boost Switch	LADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LADC input Mixer l_output mixer Switch	LADC input Mixer 设置，使能 l_output mixer 通路	0: 关闭; 1: 开启
LADC input Mixer r_output mixer Switch	LADC input Mixer 设置，使能 r_output mixer 通路	0: 关闭; 1: 开启
LINEINL/R to L_R output mixer gain	LINEINL/R to L or R output Mixer 增益	0–7, 表示-4.5–6dB
LINEOUTL Mux	LINEOUTL Mux 设置	0:left output mixer; 1:left+right output mixer
LINEOUTR Mux	LINEOUTR Mux 设置	0:right output mixer; 1:left+right output mixer
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer LINEINL Switch	Left Output Mixer 设置，使能 LINEINL 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
MIC1 boost AMP gain control	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC1_G boost stage output mixer control	MIC1 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 BST stage to L_R outp mixer gain	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
RADC input Mixer LINEINR Switch	RADC input Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
RADC input Mixer MIC1 boost Switch	RADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 l_output mixer 通路	0: 关闭; 1: 开启
RADC input Mixer l_output Switch	RADC input Mixer 设置，使能 l_output mixer 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer LINEINR Switch	Right Output Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
SPK_L Mux	SPK_L Mux 设置	0:MIXEL Switch; 1:MIXL MIXR Switch

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
SPK_R Mux	SPK_R Mux 设置	0:MIXER Switch; 1:MIXR MIXL Switch
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB
lineout volume	lineout 音量设置	0–31, 表示-43.5–0dB
speaker volume	speaker(lineout) 音量设置	0–31, 表示-43.5–0dB

2.8.5 Daudio

硬件特性

- 三路 I2S/PCM;
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.8.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
```

2.8.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used = 1

[daudio0]
pcm_lrck_period =   0x80
slot_width_select =   0x20
pcm_lsb_first =   0x0
tx_data_mode =   0x0
rx_data_mode =   0x0
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
daudio_master =   0x04
audio_format =   0x01
signal_inversion =   0x01
frametype =   0x0
tdm_config =   0x01
mclk_div = 0x1
daudio0_used = 1
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
frametype	0: short frame = 1 clock width; 1: long frame = 2 clock width
tdm_config	0: pcm mode; 1: i2s mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

daudio 配置	daudio 配置说明
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

2.8.6 DMIC

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.8.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.8.6.2 sys_config 配置

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

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
dmic_used	是否使用 dmic 驱动。0：不使用；1：使用

snddmic 配置，即 machine 驱动的相关配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

snddmic 配置	snddmic 配置说明
snddmic_used	是否使用 snddmic 驱动。0：不使用；1：使用

2.8.7 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

2.8.7.1 播放

```
通过SPKL/R播歌,例如喇叭播歌
amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L'
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R'
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1
amixer -Dhw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -Dhw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -Dhw:audiocodec cset name='SPK_R Mux' 'MIXER_Switch'
amixer -Dhw:audiocodec cset name='SPK_L Mux' 'MIXEL_Switch'
amixer -Dhw:audiocodec cset name='External Speaker Switch' 1
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav

可通过下面命令调节硬件上的模拟音量:
amixer -Dhw:audiocodec cset name='speaker volume' 28
```

```
通过LINEOUTL/R播歌
amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L'
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R'
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1
amixer -Dhw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -Dhw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -Dhw:audiocodec cset name='LINEOUTL Mux' 'LOMIX'
amixer -Dhw:audiocodec cset name='LINEOUTR Mux' 'ROMIX'
amixer -Dhw:audiocodec cset name='Lineout Switch' 1
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav

可通过下面命令调节硬件上的模拟音量:
amixer -Dhw:audiocodec cset name='lineout volume' 28
```

```
通过HPOUTL/R播歌,例如耳机
amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L'
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R'
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1
amixer -Dhw:audiocodec cset name='HP_R Mux' 'DACR HPR Switch'
amixer -Dhw:audiocodec cset name='HP_L Mux' 'DACL HPL Switch'
amixer -Dhw:audiocodec cset name='Headphone Switch' 1
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav


可通过下面命令调节硬件上的模拟音量:
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
amixer -Dhw:audiocodec cset name='headphone volume' 60
```

2.8.7.2 录音

```
通过MIC1,MIC2录音
amixer -Dhw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -Dhw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD0L Mixer ADCL Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD0R Mixer ADCR Switch' 1
amixer -Dhw:audiocodec cset name='AIF1OUT0L Mux' 'AIF1_AD0L'
amixer -Dhw:audiocodec cset name='AIF1OUT0R Mux' 'AIF1_AD0R'

amixer -Dhw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -Dhw:audiocodec cset name='MIC2 boost AMP gain control' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.9 R58 音频接口

2.9.1 硬件资源

R58 只含有一个 I2S(TDM) 接口，但是方案上会搭配 AXP813,AXP813 除了提供 PMU 相关功能外，还内置有 AudioCodec。

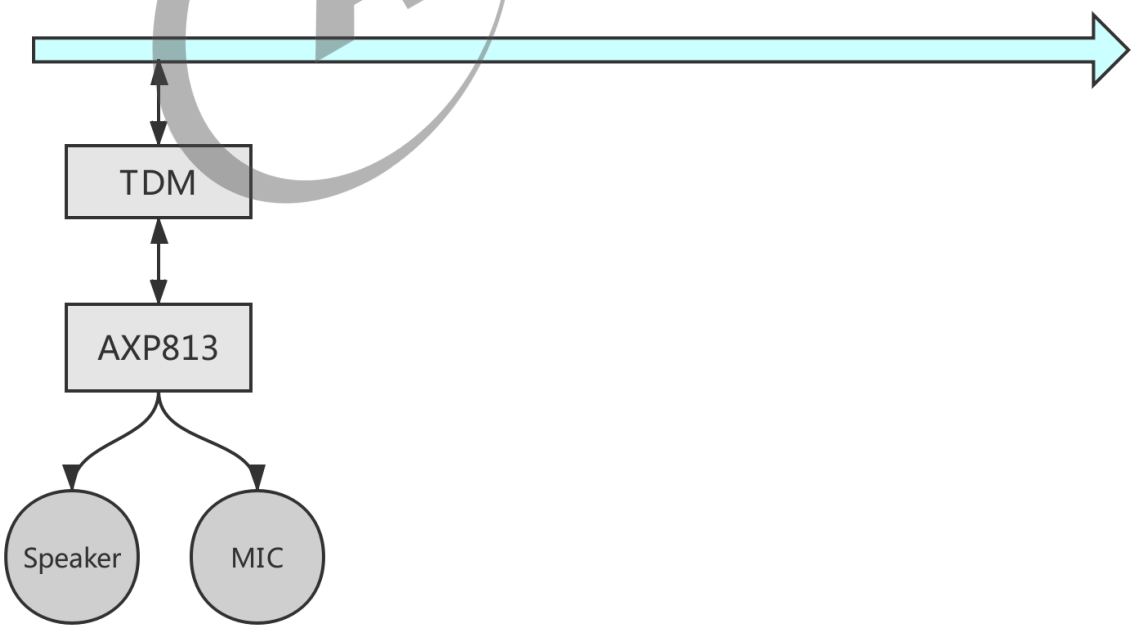


图 2-20: R58 音频硬件框图

2.9.2 时钟源

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

2.9.3 代码结构

```
linux-3.4/sound/soc/sunxi/daudio0
├── a83_ac100.c                // ac100 machine 驱动
├── a83_ac200.c                // ac200 machine 驱动
├── bb_dai.c                   // cpudai(bb-voice-dai) 驱动
├── sunxi-daudio0.c            // cpudai(pri_dai) 驱动
├── sunxi-daudio0.h
├── sunxi-daudiodma0.c         // platform 驱动
└── sunxi-daudiodma0.h

linux-3.4/sound/soc/codecs/
├── ac100_dapm.c               // ac100 codec驱动
├── acx00_dapm.c               // ac200 codec驱动
└── acx00_dapm.h
```

2.9.4 Daudio

硬件特性

- 一路 I2S/PCM;
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode,TDM mode
- 支持 i2s,pcm 协议格式配置
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.9.4.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
           <*>   SoC daudio0 tdm interface for SUNXI chips
           <*>   Machine for A83 ACX00
           <*>   virtual bb interface
           <*>   SUNXI AudioCodec ACX00 DAPM
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.9.4.2 sys_config 配置

```
[acx0]
ac100_used                      = 1
ac200_used                      = 0
```

acx0 配置, 即配置是否使用 ac100/ac200(均在 AXP813 中)

acx0 配置	acx0 配置说明
ac100_used	是否使用 ac100 驱动。0：不使用；1：使用
ac200_used	是否使用 ac100 驱动。0：不使用；1：使用

```
[tdm0]
daudio_used         = 1
daudio_master           = 4
daudio_select           = 1
audio_format            = 1
signal_inversion        = 1
mclk_fs                         = 512
sample_resolution   = 16
slot_width_select       = 16
;pcm_sync_period        = 256
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>
```

tdm0 配置	tdm0 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
daudio_select	0: pcm mode; 1: i2s mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

tdm0 配置	tdm0 配置说明
audio_format	1：SND_SOC_DAIFMT_I2S(standard i2s format) 2：SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3：SND_SOC_DAIFMT_LEFT_J(left justfied format) 4：SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
sample_resolution	采样精度,16bit, 24bit,32bit
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	可配置 16/32/64/128/256 个 bclk
pcm_lrckr_period	可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0：msb first; 1：lsb first
sign_extend	0：zero pending; 1：sign extend
tx_data_mode	0：16bit linear PCM;1：reserved;2：8bit u-law;3：8bit a-law
rx_data_mode	0：16bit linear PCM;1：reserved;2：8bit u-law;3：8bit a-law
i2s_bclk	i2s_bclk 引脚
i2s_lrclk	i2s_lrclk 引脚
i2s_dout0	i2s_dout 引脚
i2s_din	i2s_din 引脚

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.9.4.3 codec 数据通路

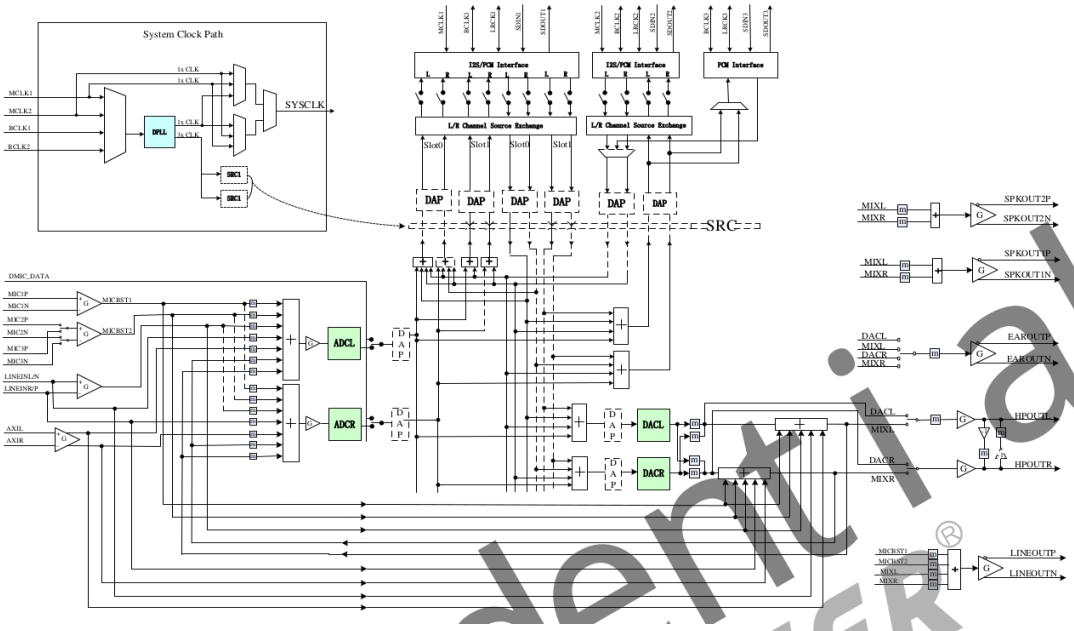


图 2-21: R58 ac100 音频通路

```
通过HPOUTL/R播歌
AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> Left Output Mixer  --> HP_L Mux --> HPOUTL -->
     Headphone
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> Right Output Mixer --> HP_R Mux --> HPOUTR -->
     Headphone

通过MIC1,MIC2录音
AIF1ADCL <-- AIF1OUT0L Mux <-- AIF1 AD0L Mixer <-- ADCL Mux <-- LEFT ADC input Mixer <--
    MIC1 PGA <-- MIC1P/N
AIF1ADCR <-- AIF1OUT0R Mux <-- AIF1 AD0R Mixer <-- ADCR Mux <-- RIGHT ADC input Mixer <--
    MIC2 PGA <-- MIC2 SRC <-- MIC2
```

ac100 相关控件如下表:

控件名称	功能	数值
Headphone Switch	Headphone 通路使能	0: 关闭; 1: 开启
ADC volume	ADCL/ADCR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
ADCL Mux	ADCL Mux 设置, 只支持 0:ADC	0:ADC
ADCR Mux	ADCR Mux 设置, 只支持 0:ADC	0:ADC

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1 AD0L Mixer ADCL Switch	AIF1 AD0L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF1 DA0L Switch	AIF1 AD0L Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACL Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACR Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer ADCR Switch	AIF1 AD0R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF1 DA0R Switch	AIF1 AD0R Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACL Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACR Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer ADCL Switch	AIF1 AD1L Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer AIF2 DACL Switch	AIF1 AD1L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer ADCR Switch	AIF1 AD1R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer AIF2 DACR Switch	AIF1 AD1R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 ADC timeslot 0 mixer gain	AIF1 ADC0L/ADC0R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC0L Mixer, bit0:AIF2 DACR; bit1:ADCL; bit2:AIF2 DACL; bit3:AIF2 DA0L; 对于 ADC0R Mixer,

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
		bit0:AIF2 DACL; bit1:ADCR; bit2:AIF2 DACR; bit3:AIF2 DA0R;
AIF1 ADC timeslot 0 volume	AIF1 ADC0L/ADC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 ADC timeslot 1 mixer gain	AIF1 ADC1L/ADC1R Mixer, 数字增益	0:0dB; 1:-6dB;
		对于 ADC1L Mixer, bit0:ADCL; bit1:AIF2 DACL; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACR;
AIF1 ADC timeslot 1 volume	AIF1 ADC1L/ADC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 0 volume	AIF1 DAC0L/DAC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 1 volume	AIF1 DAC1L/DAC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1IN0L Mux	AIF1IN0L Mux 设置	0:AIF1_DA0L; 1:AIF1_DA0R; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R
AIF1IN0R Mux	AIF1IN0R Mux 设置	0:AIF1_DA0R; 1:AIF1_DA0L; 2:SUM_AIF1DA0L_AIF1DA0R; 3:AVE_AIF1DA0L_AIF1DA0R
AIF1IN1L Mux	AIF1IN1L Mux 设置	0:AIF1_DA1L; 1:AIF1_DA1R; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1IN1R Mux	AIF1IN1R Mux 设置	0:AIF1_DA1R; 1:AIF1_DA1L; 2:SUM_AIF1DA1L_AIF1DA1R; 3:AVE_AIF1DA1L_AIF1DA1R
AIF1OUT0L Mux	AIF1OUT0L Mux 设置	0:AIF1_AD0L; 1:AIF1_AD0R; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1OUT0R Mux	AIF1OUT0R Mux 设置	0:AIF1_AD0R; 1:AIF1_AD0L; 2:SUM_AIF1AD0L_AIF1AD0R; 3:AVE_AIF1AD0L_AIF1AD0R
AIF1OUT1L Mux	AIF1OUT1L Mux 设置	0:AIF1_AD1L; 1:AIF1_AD1R; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
AIF1OUT1R Mux	AIF1OUT1R Mux 设置	0:AIF1_AD1R; 1:AIF1_AD1L; 2:SUM_AIF1AD1L_AIF1AD1R; 3:AVE_AIF1AD1L_AIF1AD1R
DAC mixer gain	DAC mixer 增益	0:0dB; 1:-6dB; 对于 DACL Mixer, bit0:ADCL; bit1:AIF2 DACL; bit2:AIF1 DAC1L; bit3:AIF1 DAC0L; 对于 DACR Mixer, bit0:ADCR; bit1:AIF2 DACR; bit2:AIF1 DAC1R; bit3:AIF1 DAC0R;
DAC volume	DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
DACL Mixer ADCL Switch	DACL Mixer 设置，使能 ADCL 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA0L Switch	DACL Mixer 设置，使能 AIF1DA0L 通路	0: 关闭; 1: 开启
DACL Mixer AIF1DA1L Switch	DACL Mixer 设置，使能 AIF1DA1L 通路	0: 关闭; 1: 开启
DACL Mixer AIF2DACL Switch	DACL Mixer 设置，使能 AIF2DACL 通路	0: 关闭; 1: 开启
DACR Mixer ADCR Switch	DACR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA0R Switch	DACR Mixer 设置，使能 AIF1DA0R 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA1R Switch	DACR Mixer 设置，使能 AIF1DA1R 通路	0: 关闭; 1: 开启
DACR Mixer AIF2DACR Switch	DACR Mixer 设置，使能 AIF2DACR 通路	0: 关闭; 1: 开启
External Speaker Switch	使能 Headphone 以及 PA	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
HP_L Mux	HP_L Mux 设置	0:DACL ; 1:Left Output Mixer
HP_R Mux	HP_R Mux 设置	0:DACR ; 1:Right Output Mixer
LEFT ADC input Mixer Lout_Mixer_Switch	LEFT ADC input Mixer 设置，使能 Lout Mixer 通路	0: 关闭; 1: 开启
LEFT ADC input Mixer MIC1 boost Switch	LEFT ADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LEFT ADC input Mixer MIC2 boost Switch	LEFT ADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LEFT ADC input Mixer Rout_Mixer_Switch	LEFT ADC input Mixer 设置，使能 Rout Mixer 通路	0: 关闭; 1: 开启
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
MIC1 boost amplifier gain	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost amplifier gain	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
RIGHT ADC input Mixer Lout_Mixer_Switch	RIGHT ADC input Mixer 设置，使能 Lout Mixer 通路	0: 关闭; 1: 开启
RIGHT ADC input Mixer MIC1 boost Switch	RIGHT ADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RIGHT ADC input Mixer MIC2 boost Switch	RIGHT ADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RIGHT ADC input Mixer Rout_Mixer_Switch	RIGHT ADC input Mixer 设置，使能 Rout Mixer 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

2.9.5 标案音频测试方法

2.9.5.1 播放

```
通过耳机播放
amixer -Dhw:sndac100 cset name='headphone volume' 60
amixer -Dhw:sndac100 cset name='AIF1IN0L Mux' 'AIF1_DA0L'
amixer -Dhw:sndac100 cset name='AIF1IN0R Mux' 'AIF1_DA0R'
amixer -Dhw:sndac100 cset name='DACL Mixer AIF1DA0L Switch' 1
amixer -Dhw:sndac100 cset name='DACR Mixer AIF1DA0R Switch' 1
amixer -Dhw:sndac100 cset name='HP_R Mux' 'DACR HPR Switch'
amixer -Dhw:sndac100 cset name='HP_L Mux' 'DACL HPL Switch'
amixer -Dhw:sndac100 cset name='Headphone Switch' 1

aplay -Dhw:sndac100 /mnt/UDISK/1KHz_0dB_16000.wav
```

2.9.5.2 录音

```
通过MIC1,MIC2录音
amixer -Dhw:sndac100 cset name='MIC2 SRC' 'MIC3'
amixer -Dhw:sndac100 cset name='LEFT ADC input Mixer MIC1 boost Switch' 1
amixer -Dhw:sndac100 cset name='RIGHT ADC input Mixer MIC2 boost Switch' 1
amixer -Dhw:sndac100 cset name='AIF1 AD0L Mixer ADCL Switch' 1
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
amixer -Dhw:sndac100 cset name='AIF1 AD0R Mixer ADCR Switch' 1
amixer -Dhw:sndac100 cset name='AIF1OUT0L Mux' 'AIF1_AD0L'
amixer -Dhw:sndac100 cset name='AIF1OUT0R Mux' 'AIF1_AD0R'

amixer -Dhw:sndac100 cset name='MIC1 boost amplifier gain' 4
amixer -Dhw:sndac100 cset name='MIC2 boost amplifier gain' 4

arecord -Dhw:sndac100 -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.10 R332 音频接口

2.10.1 硬件资源

R332 包含 2 个音频模块，分别是内置 AudioCodec 以及 Daudio0。

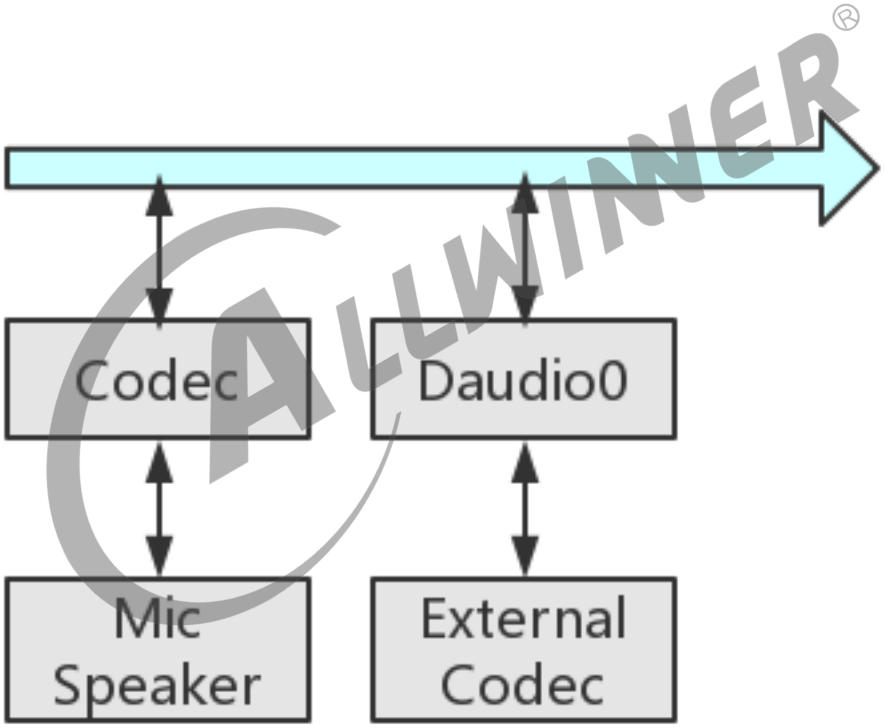


图 2-22: R332 音频硬件框图

2.10.2 时钟源

R332 中，2 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

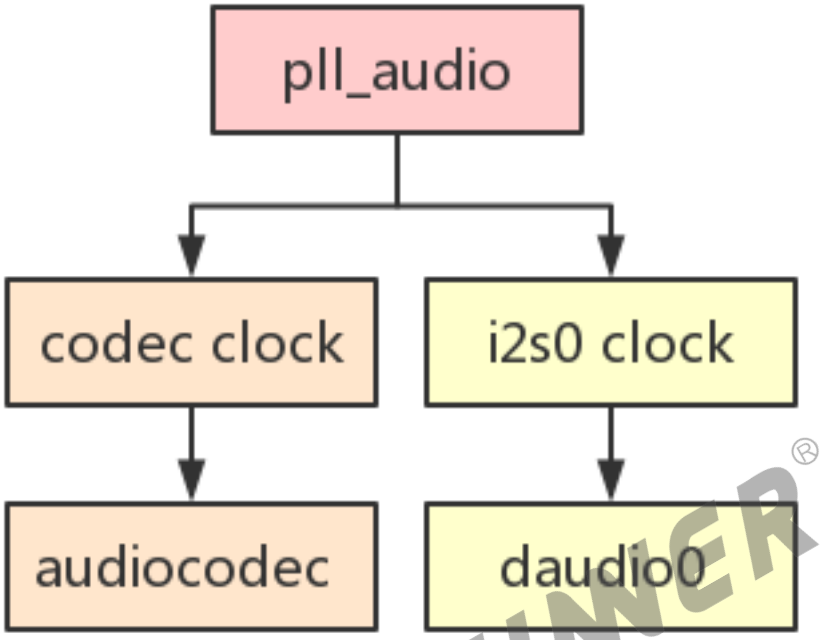


图 2-23: R332 时钟源

2.10.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sunxi-pcm.c                    // 提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sun8iw8
│   ├── sunxi_codec.c              // cpudai 驱动
│   ├── sunxi_codecdma.c           // codec platform驱动
│   ├── sun8iw8_sndcodec_new.c     // codec 驱动
│   └── sunxi_sndcodec.c           // codec machine驱动
├── sunxi-daudio.c                 // daudio platform驱动
└── sunxi-snddaudio.c              // daudio machine驱动

linux-4.9/sound/soc/soc-utils.c    // daudio codec驱动
```

2.10.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出:
  - 一路立体声 LINEOUT 输出 (LINEOUTP, LINEOUTN)
  - 一路立体声 headphone 输出 (HPOUTL, HPOUTR)
- 两路模拟输入：MIC1,MIC2
- 支持同时 playback 和 record(全双工模式)
- 支持 ADC 的 AGC,DRC 功能
- 支持 DAC 的 DRC 功能

2.10.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Audiocodec for the SUNXI chips
```

2.10.4.2 sys_config 配置

```
[codec]
headphone_vol           = 0x3b
lineout_vol             = 0x1a
audio_pa_ctrl           = port:PB05<1><default><default><0>
adcagc_used             = 0
adcdrc_used             = 0
dacdrc_used             = 0
adchpf_used             = 0
dachpf_used             = 0
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
headphone_vol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
audio_pa_ctrl	PA 使能引脚
adcagc_used	1:use adcagc 0:no use
adcdrc_used	1:use adcdrc 0:no use
dacdrc_used	1:use dacdrc 0:no use
adchpf_used	1:use adchpf 0:no use
dachpf_used	1:use dachpf 0:no use

2.10.4.3 codec 数据通路

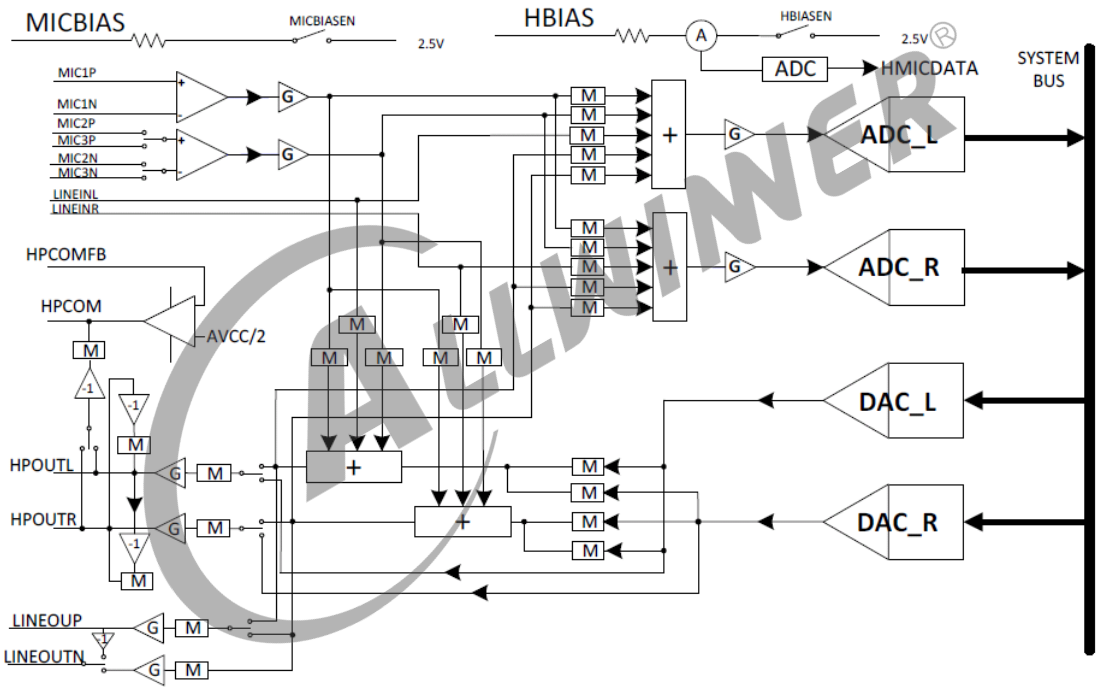


图 2-24: R332 音频通路

```
播歌
DACL --> Left Output Mixer --> LINEOUTL
DACR --> Right Output Mixer --> LINEOUTR

录音
MIC1P --> LADC input Mixer --> ADCL
MIC2P --> RADC input Mixer --> ADCR
```

控件名称	功能	数值
Lineout volume	lineout 音量设置	0–31, 表示-43.5–0dB
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB
HP_L Mux	HP_L Mux 设置	0:DACL HPL Switch; 1:MIXER_L Switch
HP_R Mux	HP_R Mux 设置	0:DACR HPR Switch; 1:MIXER_R Switch
LADC input Mixer MIC1 boost Switch	LADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LADC input Mixer MIC2 boost Switch	LADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LADC input Mixer l_output mixer Switch	LADC input Mixer 设置，使能 l_output 通路	0: 关闭; 1: 开启
LADC input Mixer r_output mixer Switch	LADC input Mixer 设置，使能 r_output 通路	0: 关闭; 1: 开启
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
MIC1 boost AMP gain control	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC1_G boost stage output mixer control	MIC1 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC2_G boost stage output mixer control	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
RADC input Mixer MIC1 boost Switch	RADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 l_output 通路	0: 关闭; 1: 开启
RADC input Mixer r_output mixer Switch	RADC input Mixer 设置，使能 r_output 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
SPK_L Mux	SPK_L Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
SPK_R Mux	SPK_R Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

通路设置举例：

1. 播放通路

```
通过lineout播放:
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 6
```

2. 录音通路

```
通过MIC1,MIC2录音:
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC2 SRC' 0
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -D hw:audiocodec cset name='MIC2 boost AMP gain control' 4
```

2.10.5 Daudio

硬件特性

- 一路 I2S/PCM；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

2.10.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Allwinner Digital Audio Support
```

2.10.5.2 sys_config 配置

```
[tdm0]
daudio_used          = 0
daudio_master            = 4
daudio_select            = 1
audio_format             = 1
signal_inversion         = 1
sample_resolution    = 16
slot_width_select        = 16
pcm_lrck_period          = 32
pcm_lrckr_period         = 1
msb_lsb_first        = 0
sign_extend              = 0
tx_data_mode             = 0
rx_data_mode             = 0
;i2s_mclk             = port:PB08<2><1><default><default>
i2s_bclk             = port:PG11<2><1><default><default>
i2s_lrclk            = port:PG10<2><1><default><default>
i2s_dout0            = port:PG12<2><1><default><default>
i2s_dout1            =
i2s_dout2            =
i2s_dout3            =
i2s_din              = port:PG13<2><1><default><default>
```

tdm0 配置	tdm0 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
daudio_select	0: pcm mode; 1: i2s mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

tdm0 配置	tdm0 配置说明
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
sample_resolution	采样精度,16bit, 24bit,32bit
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	可配置 16/32/64/128/256 个 bclk
pcm_lrckr_period	可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
i2s_bclk	i2s_bclk 引脚
i2s_lrclk	i2s_lrclk 引脚
i2s_dout0	i2s_dout 引脚
i2s_din	i2s_din 引脚

2.10.6 外挂 codec:AC100

2.10.6.1 内核配置

2.10.6.2 sys_config 配置

2.10.7 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.10.7.1 播放

```
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Lineout volume' 24
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 0

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='Lineout volume' 50
```

2.10.7.2 录音

表示下使用 audiocodec 进行 MIC1,MIC2 录音

```
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -D hw:audiocodec cset name='MIC2 SRC' 0
amixer -D hw:audiocodec cset name='MIC2 boost AMP gain control' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.11 R333 音频接口

2.11.1 硬件资源

R333 包含 2 个音频模块，分别是内置 audiocodec 以及 daudio0。

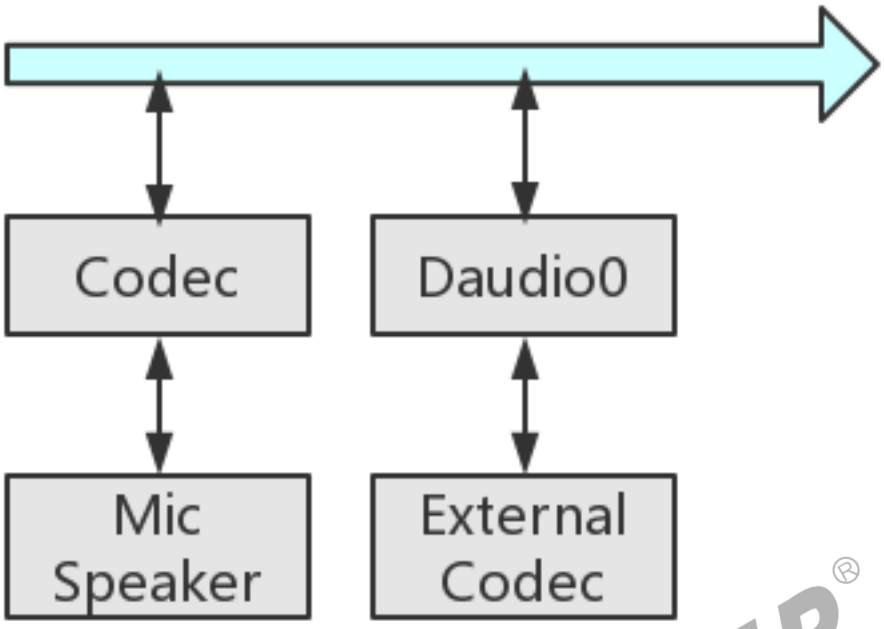


图 2-25: R333 音频硬件框图

2.11.2 时钟源

R333 中，2 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

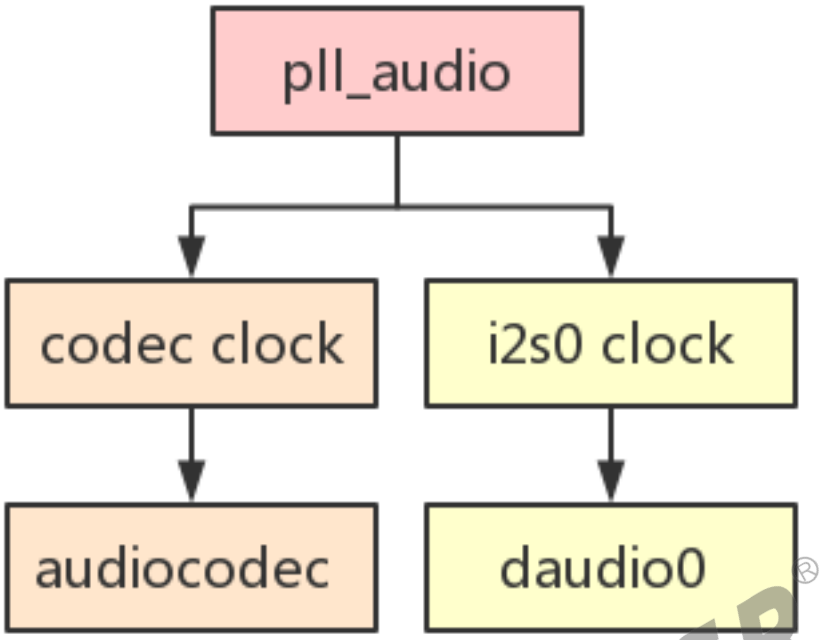


图 2-26: R333 时钟源

2.11.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sunxi-pcm.c                  // 提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sun8iw8
│   ├── sunxi_codec.c            // cpudai 驱动
│   ├── sunxi_codecdma.c         // codec platform驱动
│   ├── sun8iw8_sndcodec_new.c   // codec 驱动
│   └── sunxi_sndcodec.c         // codec machine驱动
├── sunxi-daudio.c               // daudio platform驱动
└── sunxi-snddaudio.c            // daudio machine驱动

linux-4.9/sound/soc/soc-utils.c  // daudio codec驱动
```

2.11.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出:
  - 一路立体声 LINEOUT 输出 (LINEOUTP, LINEOUTN)
  - 一路立体声 headphone 输出 (HPOUTL, HPOUTR)
- 两路模拟输入：MIC1,MIC2
- 支持同时 playback 和 record(全双工模式)
- 支持 ADC 的 AGC,DRC 功能
- 支持 DAC 的 DRC 功能

2.11.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Audiocodec for the SUNXI chips
```

2.11.4.2 sys_config 配置

```
[codec]
headphone_vol           = 0x3b
lineout_vol             = 0x1a
audio_pa_ctrl           = port:PB05<1><default><default><0>
adcagc_used             = 0
adcdrc_used             = 0
dacdrc_used             = 0
adchpf_used             = 0
dachpf_used             = 0
```

codec 配置	codec 配置说明
headphone_vol	headphone volume，可设定范围 0~0x3f, 0 表示 mute, 1~63 表示-62dB~0dB, 1dB/step
audio_pa_ctrl	PA 使能引脚
adcagc_used	1:use adcagc 0:no use
adcdrc_used	1:use adcdrc 0:no use

codec 配置	codec 配置说明
dacdrc_used	1:use dacdrc 0:no use
adchpf_used	1:use adchpf 0:no use
dachpf_used	1:use dachpf 0:no use

2.11.4.3 codec 数据通路

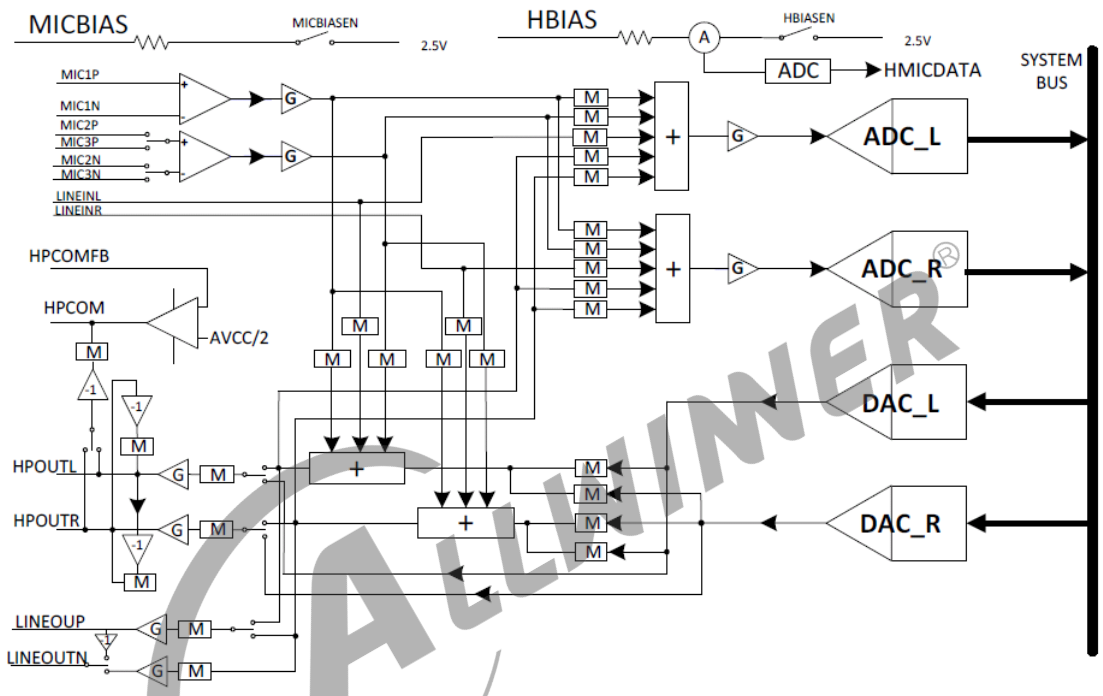


图 2-27: R333 音频通路

```
播歌
DACL --> Left Output Mixer --> LINEOUTL
DACR --> Right Output Mixer --> LINEOUTR

录音
MIC1P --> LADC input Mixer --> ADCL
MIC2P --> RADC input Mixer --> ADCR
```

控件名称	功能	数值
Lineout volume	lineout 音量设置	0–31, 表示-43.5–0dB
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB
HP_L Mux	HP_L Mux 设置	0:DACL HPL Switch; 1:MIXER_L Switch
HP_R Mux	HP_R Mux 设置	0:DACR HPR Switch; 1:MIXER_R Switch

控件名称	功能	数值
LADC input Mixer MIC1 boost Switch	LADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
LADC input Mixer MIC2 boost Switch	LADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
LADC input Mixer l_output mixer Switch	LADC input Mixer 设置，使能 l_output 通路	0: 关闭; 1: 开启
LADC input Mixer r_output mixer Switch	LADC input Mixer 设置，使能 r_output 通路	0: 关闭; 1: 开启
Left Output Mixer DACL Switch	Left Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1Booststage Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Booststage Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
MIC1 boost AMP gain control	MIC1 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC1_G boost stage output mixer control	MIC1 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:24–42dB,3dB/step
MIC2_G boost stage output mixer control	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
RADC input Mixer MIC1 boost Switch	RADC input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 l_output 通路	0: 关闭; 1: 开启
RADC input Mixer r_output mixer Switch	RADC input Mixer 设置，使能 r_output 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC1Booststage Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
SPK_L Mux	SPK_L Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
SPK_R Mux	SPK_R Mux 设置	0:MIXER_L Switch; 1:MIXR+MIXL
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

通路设置举例：

1. 播放通路

```
通过lineout播放:
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 6
```

2. 录音通路

```
通过MIC1,MIC2录音:
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC2 SRC' 0
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -D hw:audiocodec cset name='MIC2 boost AMP gain control' 4
```

2.11.5 Daudio

硬件特性

- 一路 I2S/PCM；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.11.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         <*>   Allwinner Digital Audio Support
```

2.11.5.2 sys_config 配置

```
[tdm0]
daudio_used         = 0
daudio_master           = 4
daudio_select           = 1
audio_format            = 1
signal_inversion        = 1
sample_resolution   = 16
slot_width_select       = 16
pcm_lrck_period         = 32
pcm_lrckr_period        = 1
msb_lsb_first       = 0
sign_extend             = 0
tx_data_mode            = 0
rx_data_mode            = 0
;i2s_mclk            = port:PB08<2><1><default><default>
i2s_bclk            = port:PG11<2><1><default><default>
i2s_lrclk           = port:PG10<2><1><default><default>
i2s_dout0           = port:PG12<2><1><default><default>
i2s_dout1           =
i2s_dout2           =
i2s_dout3           =
i2s_din             = port:PG13<2><1><default><default>
```

tdm0 配置	tdm0 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
daudio_select	0: pcm mode; 1: i2s mode
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge)

tdm0 配置	tdm0 配置说明
signal_inversion	5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge) 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)
sample_resolution	采样精度,16bit, 24bit,32bit
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	可配置 16/32/64/128/256 个 bclk
pcm_lrckr_period	可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
i2s_bclk	i2s_bclk 引脚
i2s_lrclk	i2s_lrclk 引脚
i2s_dout0	i2s_dout 引脚
i2s_din	i2s_din 引脚

2.11.6 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.11.6.1 播放

```
amixer -D hw:audiocodec cset name='SPK_L Mux' 1
amixer -D hw:audiocodec cset name='SPK_R Mux' 1
amixer -D hw:audiocodec cset name='Lineout volume' 24
amixer -D hw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -D hw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 0

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='Lineout volume' 50
```

2.11.6.2 录音

表示下使用 audiocodec 进行 MIC1,MIC2 录音

```
amixer -D hw:audiocodec cset name='LADC input Mixer MIC1 boost Switch' 1
amixer -D hw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost AMP gain control' 4
amixer -D hw:audiocodec cset name='MIC2 SRC' 0
amixer -D hw:audiocodec cset name='MIC2 boost AMP gain control' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.12 R311 音频接口

2.12.1 硬件资源

R311 包含 4 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Dmic。

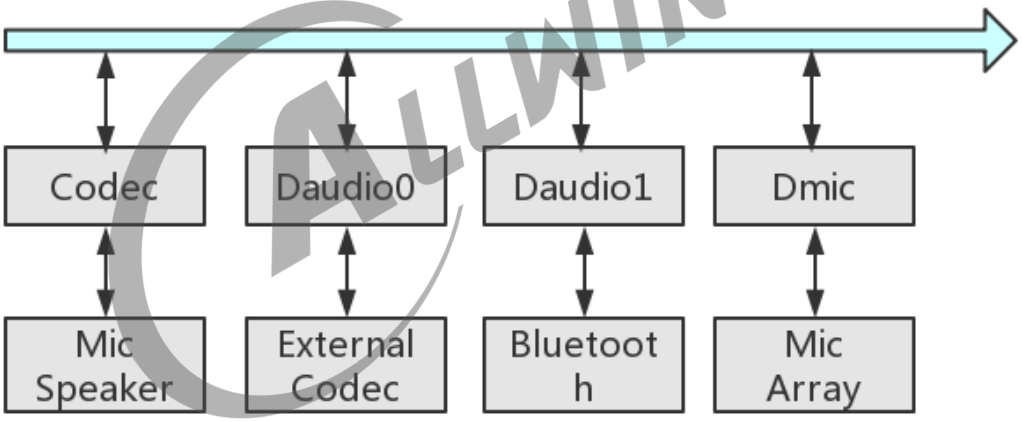


图 2-28: R311 音频硬件框图

2.12.2 时钟源

R311 中，4 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

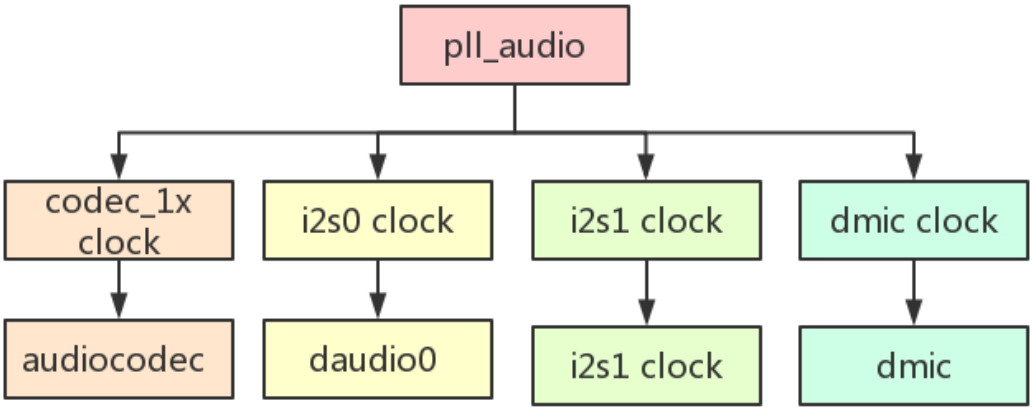


图 2-29: R311 时钟源

2.12.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sun8iw15-codec.c                // codec驱动
├── sun8iw15-codec.h
├── sun8iw15-sndcodec.c             // codec machine驱动
├── sunxi-inter-i2s.c               // codec platform驱动
├── sunxi-inter-i2s.h
├── sunxi-daudio.c                  // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-dmic.c                    // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-pcm.c                     // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi_rw_func.c                 // 通用文件，读写模拟/数字寄存器的接口
├── sunxi_rw_func.h
├── sunxi-snddaudio.c               // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-snddmic.c                 // dmic machine驱动
└── sunxi-snddmic.h

linux-4.9/sound/soc/codecs/dmic.c   // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c     // daudio codec驱动
```

2.12.4 AudioCodec

硬件特性

- 两路 DAC

  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 一路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 一路模拟输出：一路立体声 headphone 输出 (HPL, HPR)
- 三路模拟输入：MIC2,MIC3,LINEINR
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 5 段 DRC
- DAC FIFO 长度 128*24bits, ADC FIFO 长度 64*24bits
- 支持耳机及其按键检测

2.12.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun8iw15 Codec Support
```

2.12.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used = 0x1
aif2fmt = 0x3
aif3fmt = 0x3
aif2master = 0x1
hp_detect_case = 0x0
;-------------------------------------------------------------------------------
[i2s]
i2s_used = 0x1
;-------------------------------------------------------------------------------
[codec]
codec_used = 0x1
headphonevol =   0x2f
maingain =   0x4
headsetmicgain =  0x4
adcagc_cfg =   0x0
adcdrc_cfg =   0x0
adchpf_cfg =   0x0
dacdrc_cfg =   0x0
dachpf_cfg =   0x0
aif2config =  0x0
aif3config =  0x0
gpio-spk =
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用
aif2fmt	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif3fmt	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif2master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 aif 接口选择 master 模式 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 aif 接口选择 slave 模式
hp_detect_case	jack irq level, 0:low; 1:high

i2s 配置，即 audiocodec platform 驱动的相关配置, 内部 aif 接口用的 i2s(与 i2s0,i2s1 接口无关)

i2s 配置	i2s 配置说明
i2s_used	是否使用 i2s 驱动。0：不使用；1：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
headphonevol	初始化 headphone volume，可设定范围 0~0x3f, 表示 0~-62dB, -1dB/step
headsetmicgain	指的是 MIC2 增益, 可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
adcinputgain	adc 增益，可设定范围 0~0x7, 表示-4.5~6dB, 1.5dB/step, 一般设置 0x3, 即 0dB
adcagc_cfg	是否使用 adcagc. 0: 不使用；1：使用
adcdrc_cfg	是否使用 adcdrc. 0: 不使用；1：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
adchpf_cfg	是否使用 adchpf. 0: 不使用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不使用；1：使用
aif2config	是否使用 aif2. 0: 不使用；1：使用
aif3config	是否使用 aif3. 0: 不使用；1：使用
gpio-spk	PA 使能引脚

说明

- **如果想要正常加载 *audiocodec* 声卡，需要把 *codec,platform,machine* 驱动都选上，即 *codec_used,i2s_used,sndcodec_used* 都置为 *1*；**
- ***headphonevol,headsetmicgain* 等值会在驱动初始化的时候设置，进入系统后还可以通过 *amixer* 工具对应控件进行再次修改；**
- **注意 *gpio-spk* 是否配置正确，是否有其他模块复用了该 *gpio;***

2.12.4.3 codec 数据通路

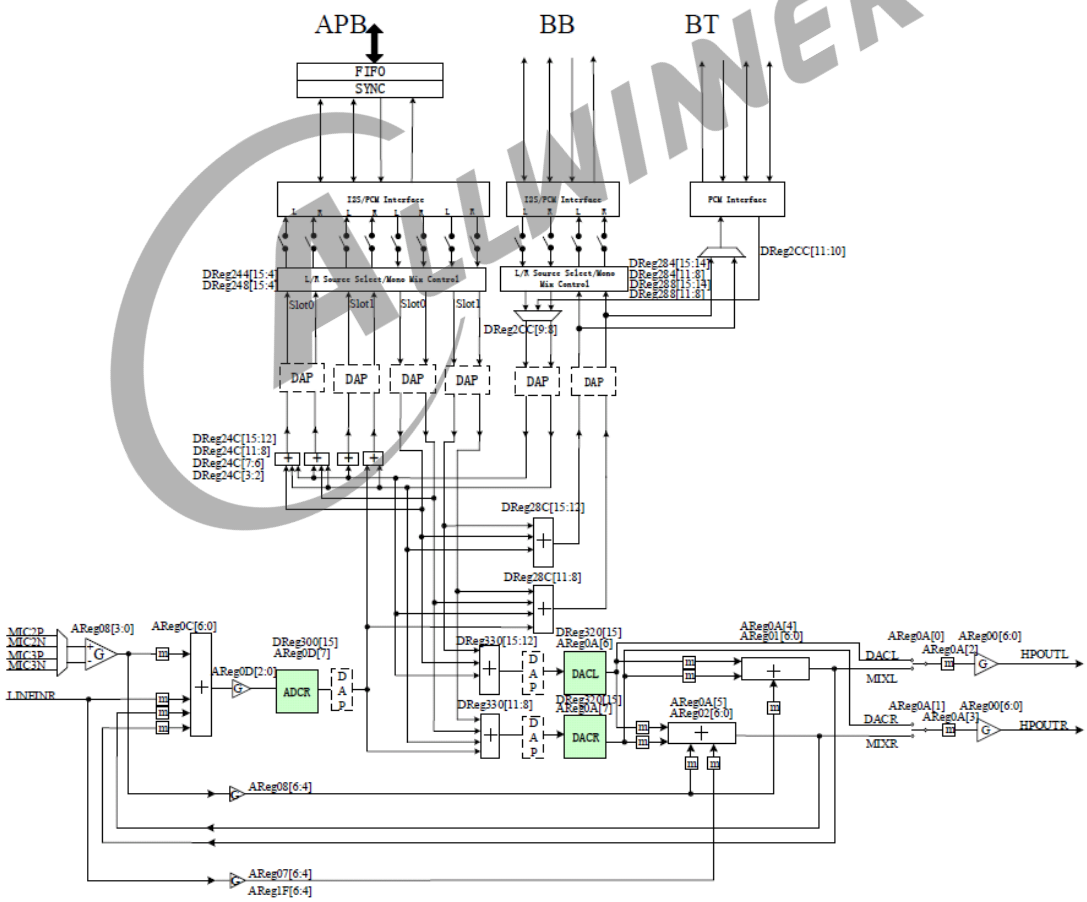


图 2-30: R311 音频通路

```
播歌

AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> HP_L Mux --> HPOUTL --> Headphone
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> HP_R Mux --> HPOUTR --> Headphone

录音
MIC2 --> MIC2 SRC --> MIC2 PGA --> RADC input Mixer --> AIF1 AD0R Mixer --> AIF1OUT0L Mux
    --> AIF1ADCL
MIC2 --> MIC2 SRC --> MIC2 PGA --> RADC input Mixer --> AIF1 AD0R Mixer --> AIF1OUT0R Mux
    --> AIF1ADCR
```

R311 所有控件如下表：

控件名称	功能	数值
Headphone Switch	使能 Headphone	0: 关闭; 1: 开启
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB, 具体计算请看注释 1
ADC volume	ADC 左/右通道音量	0–0xff, 0 表示 mute, 0x1～0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
I2S HUB FUNC	使能 hub 功能	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF1 DA0L Switch	AIF1 AD0L Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACL Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACR Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer ADCR Switch	AIF1 AD0R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF1 DA0R Switch	AIF1 AD0R Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACL Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACR Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer AIF2 DACL Switch	AIF1 AD1L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer ADCR Switch	AIF1 AD1R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer AIF2 DACR Switch	AIF1 AD1R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 ADC timeslot 0 mixer gain	AIF1 ADC0L/ADC0R Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC0L Mixer, bit0:AIF2 DACR; bit1: 保留位; bit2:AIF2 DACL;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
		bit3:AIF2 DA0L; 对于 ADC0R Mixer, bit0:AIF2 DACL; bit1:ADCR; bit2:AIF2 DACR; bit3:AIF2 DA0R;
AIF1 ADC timeslot 0 volume	AIF1 ADC0L/ADC0R 音量设置	0–0xff, 0 表示 mute, 0x1～0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 ADC timeslot 1 mixer gain	AIF1 ADC1L/ADC1R Mixer, 数字增益	0:0dB; 1:-6dB;
		对于 ADC1L Mixer, bit0: 保留位; bit1:AIF2 DACL; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACR;
AIF1 ADC timeslot 1 volume	AIF1 ADC1L/ADC1R 音量设置	0–0xff, 0 表示 mute, 0x1～0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 0 volume	AIF1 DAC0L/DAC0R 音量设置	0–0xff, 0 表示 mute, 0x1～0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 1 volume	AIF1 DAC1L/DAC1R 音量设置	0–0xff, 0 表示 mute, 0x1～0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1IN0L Mux	AIF1IN0L Mux 设置	0:AIF1_DA0L; 1:AIF1_DA0R; 2:AIF1_DA0L+AIF1_DA0R; 3:AIF1_DA0L+AIF1_DA0R 的平均值

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1IN0R Mux	AIF1IN0R Mux 设置	0:AIF1_DA0R; 1:AIF1_DA0L; 2:AIF1_DA0L+AIF1_DA0R; 3:AIF1_DA0L+AIF1_DA0R 的平均值
AIF1IN1L Mux	AIF1IN1L Mux 设置	0:AIF1_DA1L; 1:AIF1_DA1R; 2:AIF1_DA1L+AIF1_DA1R; 3:AIF1_DA1L+AIF1_DA1R 的平均值
AIF1IN1R Mux	AIF1IN1R Mux 设置	0:AIF1_DA1R; 1:AIF1_DA1L; 2:AIF1_DA1L+AIF1_DA1R; 3:AIF1_DA1L+AIF1_DA1R 的平均值
AIF1OUT0L Mux	AIF1OUT0L Mux 设置	0:AIF1_AD0L; 1:AIF1_AD0R; 2:AIF1_AD0L+AIF1_AD0R; 3:AIF1_AD0L+AIF1_AD0R 的平均值
AIF1OUT0R Mux	AIF1OUT0R Mux 设置	0:AIF1_AD0R; 1:AIF1_AD0L; 2:AIF1_AD0L+AIF1_AD0R; 3:AIF1_AD0L+AIF1_AD0R 的平均值
AIF1OUT1L Mux	AIF1OUT1L Mux 设置	0:AIF1_AD1L; 1:AIF1_AD1R; 2:AIF1_AD1L+AIF1_AD1R; 3:AIF1_AD1L+AIF1_AD1R 的平均值
AIF1OUT1R Mux	AIF1OUT1R Mux 设置	0:AIF1_AD1R; 1:AIF1_AD1L; 2:AIF1_AD1L+AIF1_AD1R; 3:AIF1_AD1L+AIF1_AD1R 的平均值
AIF2 ADC mixer gain	AIF2 ADCL/ADCR Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC1L Mixer, bit0: 保留位; bit1:AIF2 DACR; bit2:AIF2 DAC1L;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
		bit3:AIF2 DAC0L; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACL; bit2:AIF2 DAC1R; bit3:AIF2 DAC0R;
AIF2 ADC volume	AIF2 ADCL/ADCR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF2 ADL Mixer AIF1 DA0L Switch	AIF2 ADL Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF2 ADL Mixer AIF1 DA1L Switch	AIF2 ADL Mixer 设置，使能 AIF1 DA1L 通路	0: 关闭; 1: 开启
AIF2 ADL Mixer AIF2 DACR Switch	AIF2 ADL Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer ADCR Switch	AIF2 ADR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer AIF1 DA0R Switch	AIF2 ADR Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer AIF1 DA1R Switch	AIF2 ADR Mixer 设置，使能 AIF1 DA1R 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer AIF2 DACL Switch	AIF2 ADR Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF2 DAC volume	AIF2 DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF2INL Mux	AIF2INL Mux 设置	0:AIF2_DACL; 1:AIF2_DACR; 2:AIF2_DACL+AIF2_DACR; 3:AIF2_DACL+AIF2_DACR 的平均值
AIF2INL Mux VIR switch aif2inl aif3	使能 aif3 -> aif2 通路	0: 关闭; 1: 开启
AIF2INL Mux switch aif2inl aif2	使能 aif2 -> aif2 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF2INR Mux	AIF2INR Mux 设置	0:AIF2_DACR; 1:AIF2_DACL; 2:AIF2_DACL+AIF2_DACR; 3:AIF2_DACL+AIF2_DACR 的平均值
AIF2INR Mux VIR switch aif2inl aif3	使能 aif3 -> aif2 通路	0: 关闭; 1: 开启
AIF2INR Mux switch aif2inl aif2	使能 aif2 -> aif2 通路	0: 关闭; 1: 开启
AIF2OUTL Mux	AIF2OUTL Mux 设置	0:AIF2_ADCL; 1:AIF2_ADCR; 2:AIF2_ADCL+AIF2_ADCR; 3:AIF2_ADCL+AIF2_ADCR 的平均值
AIF2OUTR Mux	AIF2OUTR Mux 设置	0:AIF2_ADCR; 1:AIF2_ADCL; 2:AIF2_ADCL+AIF2_ADCR; 3:AIF2_ADCL+AIF2_ADCR 的平均值
AIF3OUT Mux	AIF3OUT Mux 设置	0:NULL; 1:AIF2_ADC_Left_Channel; 2:AIF2_ADC_Right_Channel
DAC mixer gain	DAC mixer 增益	0:0dB; 1:-6dB; 对于 DACL Mixer, bit0: 保留位; bit1:AIF2 DACL; bit2:AIF1 DAC1L; bit3:AIF1 DAC0L; 对于 DACR Mixer, bit0:ADCR; bit1:AIF2 DACR; bit2:AIF1 DAC1R; bit3:AIF1 DAC0R;
DAC volume	DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
DACL Mixer AIF1DA0L Switch	DACL Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
DACL Mixer AIF1DA1L Switch	DACL Mixer 设置，使能 AIF1 DA1L 通路	0: 关闭; 1: 开启
DACL Mixer AIF2DACL Switch	DACL Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
DACR Mixer ADCR Switch	DACR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA0R Switch	DACR Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA1R Switch	DACR Mixer 设置，使能 AIF1 DA1R 通路	0: 关闭; 1: 开启
DACR Mixer AIF2DACR Switch	DACR Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
HP_L Mux	HP_L Mux 设置	0:DACL_HPL_Switch; 1:LOMixer_HPL_Switch
HP_R Mux	HP_R Mux 设置	0:DACR_HPR_Switch; 1:ROMixer_HPR_Switch
LINEINL/R to L_R output mixer gain	LINEINR to R output mixer 增益	0–7, 表示-4.5–6dB
Left Output Mixer DACL Switch	Left Output Mixer 设置, 使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置, 使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Boststage Switch	Left Output Mixer 设置, 使能 MIC2 通路	0: 关闭; 1: 开启
MIC2 BST stage to L_R outp mixer gain	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC Mux 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:15–33dB
RADC input Mixer LINEINR Switch	RADC input Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 left output mixer 通路	0: 关闭; 1: 开启
RADC input Mixer r_output mixer Switch	RADC input Mixer 设置，使能 right output mixer 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Right Output Mixer LINEINR Switch	Right output Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Speaker Switch	Speaker 开关	0: 关闭; 1: 开启
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

通路设置举例：

1. 播放通路

```
通过headphone耳机播放:

amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L';
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R';
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1;
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1;
amixer -Dhw:audiocodec cset name='HP_L Mux' 'DACL HPL Switch' ;
amixer -Dhw:audiocodec cset name='HP_R Mux' 'DACR HPR Switch';
amixer -Dhw:audiocodec cset name='Headphone Switch' 1;
```

2. 录音通路

```
通过模拟MIC2录音:

amixer -Dhw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD0R Mixer ADCR Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD1R Mixer ADCR Switch' 1
amixer -Dhw:audiocodec cset name='AIF1OUT0L Mux' 'AIF1_AD0L'
amixer -Dhw:audiocodec cset name='AIF1OUT0R Mux' 'AIF1_AD0R'

amixer -Dhw:audiocodec cset name='MIC2 boost AMP gain control' 4
```

上面的播放, 录音功能主要用到 AIF1 接口。而 AIF2 主要用于语音通话,AIF3 主要用于蓝牙通话, 这类功能多用于通话平板, 这里不作介绍。

2.12.5 Daudio

硬件特性

- 两路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 2 路 MCLK 输出

2.12.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
```

2.12.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used          = 1

[daudio0]
daudio0_used             = 1
daudio_master            = 4
tdm_config               = 1
mclk_div                 = 1
audio_format             = 1
signal_inversion         = 1
pcm_lrck_period          = 32
;msb_lsb_first           = 0
slot_width_select        = 32
frametype                = 1
;tx_data_mode            = 1
;rx_data_mode            = 1
daudio_regulator         =
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

daudio 配置	daudio 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
daudio_regulator	daudio 供电配置

- daudio machine 驱动的配置 (snddaudio), 一般来说还需要配置 codec name 以及 codec dai name

  1. 例如 daudio0 使用了 AC108 作为外挂 codec:

```
[snddaudio0]
snddaudio0_used = 1
sunxi,snddaudio-codec = "ac108.1-003b";
sunxi,snddaudio-codec-dai = "ac108-pcm0";
     [daudio0]
daudio0_used            = 1
slot_width_select       = 32
pcm_lrck_period         = 128
msb_lsb_first           = 0
sign_extend             = 0
frametype               = 0
mclk_div                = 1
tdm_config              = 1
tx_data_mode            = 0
rx_data_mode            = 0
     注意名称需要与codec驱动中配置的名称一致，如ac108驱动，路径：
linux-4.9/sound/soc/codecs/ac108.c
代码中snd_soc_register_codec注册codec驱动，其中codec device name为ac108.1-003b,codec
dai name为ac108-pcm0
```

2. 例如 daudio2 与 bluetooth 模组相连 (没有实际的 codec 驱动)，那么这时候 codec name, codec dai name 需要配置为 dummy codec, 可以如下配置：

```
[snddaudio1]
snddaudio1_used = 1
sunxi,snddaudio-codec = "snd-soc-dummy"
sunxi,snddaudio-codec-dai = "snd-soc-dummy-dai"
               因为驱动中解析snddaudio-codec等字段时，判断出错的时候则使用默认codec"snd-soc
-dummy",
所以如果sunxi,snddaudio-codec和sunxi,snddaudio-codec-dai不配置，或者配置为空的时候，
则默认使用dummy codec:
[snddaudio1]
snddaudio1_used = 1
       或者
[snddaudio1]
snddaudio1_used = 1
sunxi,snddaudio-codec =
sunxi,snddaudio-codec-dai =
daudio_master           = 1
audio_format            = 5
signal_inversion        = 2
```

2.12.6 Dmic

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.12.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.12.6.2 sys_config 配置

配置如下:

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

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
dmic_used	是否使用 dmic 驱动。0：不使用；1：使用

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
snddmic_used	是否使用 snddmic 驱动。0：不使用；1：使用

- sys_config 中不需要配置 codec 驱动相关信息

因为 machine 驱动代码中默认配置了"dmic-codec" 作为 codec 驱动, 代码路径:

```
linux-4.9/sound/soc/codecs/dmic.c
```

2.12.7 标案音频测试方法

该章节主要介绍在标案上进行播歌的测试命令。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.12.7.1 播放

```
amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L';
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R';
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1;
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1;
amixer -Dhw:audiocodec cset name='HP_L Mux' 'DACL HPL Switch' ;
amixer -Dhw:audiocodec cset name='HP_R Mux' 'DACR HPR Switch';
amixer -Dhw:audiocodec cset name='Headphone Switch' 1;

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='headphone volume' 50
```

2.13 MR133 音频接口

2.13.1 硬件资源

MR133 包含 4 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Dmic。

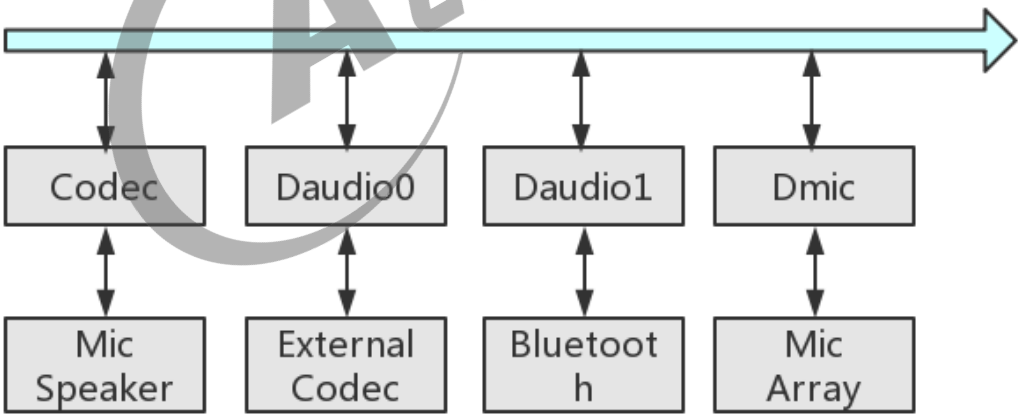


图 2-31: MR133 音频硬件框图

2.13.2 时钟源

MR133 中，4 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

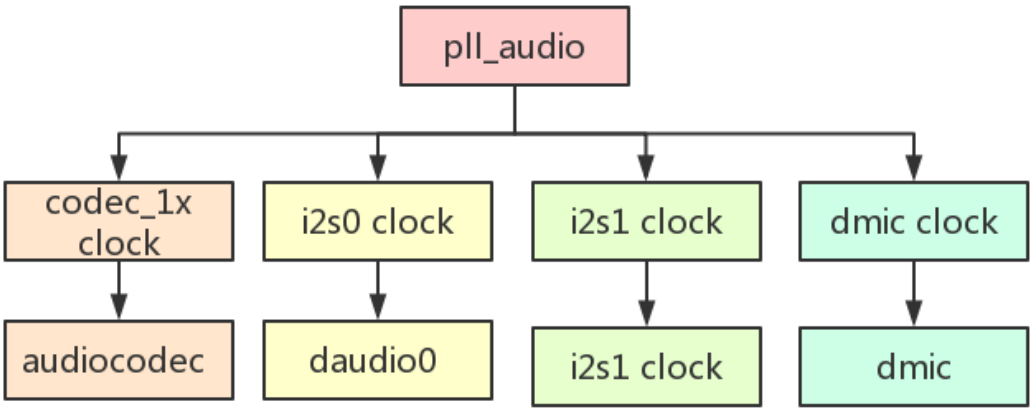


图 2-32: MR133 时钟源

2.13.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sun8iw15-codec.c              // codec驱动
├── sun8iw15-codec.h
├── sun8iw15-sndcodec.c           // codec machine驱动
├── sunxi-inter-i2s.c             // codec platform驱动
├── sunxi-inter-i2s.h
├── sunxi-daudio.c                // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-dmic.c                  // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-pcm.c                   // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi_rw_func.c               // 通用文件，读写模拟/数字寄存器的接口
├── sunxi_rw_func.h
├── sunxi-snddaudio.c             // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-snddmic.c               // dmic machine驱动
└── sunxi-snddmic.h

linux-4.9/sound/soc/codecs/dmic.c   // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c     // daudio codec驱动
```

2.13.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 一路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 一路模拟输出：一路立体声 headphone 输出 (HPL, HPR)
- 三路模拟输入：MIC2,MIC3,LINEINR
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 5 段 DRC
- DAC FIFO 长度 128*24bits, ADC FIFO 长度 64*24bits
- 支持耳机及其按键检测

2.13.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun8iw15 Codec Support
```

2.13.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used = 0x1
aif2fmt = 0x3
aif3fmt = 0x3
aif2master = 0x1
hp_detect_case = 0x0
;-------------------------------------------------------------------------------
[i2s]
i2s_used = 0x1
;-------------------------------------------------------------------------------
[codec]
codec_used = 0x1
headphonevol =   0x2f
maingain =   0x4
headsetmicgain =  0x4
adcagc_cfg =   0x0
adcdrc_cfg =   0x0
adchpf_cfg =   0x0
dacdrc_cfg =   0x0
dachpf_cfg =   0x0
aif2config =  0x0
aif3config =  0x0
gpio-spk =
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用
aif2fmt	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif3fmt	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
aif2master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 aif 接口选择 master 模式 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 aif 接口选择 slave 模式
hp_detect_case	jack irq level, 0:low; 1:high

i2s 配置，即 audiocodec platform 驱动的相关配置, 内部 aif 接口用的 i2s(与 i2s0,i2s1 接口无关)

i2s 配置	i2s 配置说明
i2s_used	是否使用 i2s 驱动。0：不使用；1：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
headphonevol	初始化 headphone volume，可设定范围 0~0x3f, 表示 0~-62dB, -1dB/step
headsetmicgain	指的是 MIC2 增益, 可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
adcinputgain	adc 增益，可设定范围 0~0x7, 表示-4.5~6dB, 1.5dB/step, 一般设置 0x3, 即 0dB
adcagc_cfg	是否使用 adcagc. 0: 不使用；1：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
adcdrc_cfg	是否使用 adcdrc. 0: 不使用；1：使用
adchpf_cfg	是否使用 adchpf. 0: 不使用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不使用；1：使用
aif2config	是否使用 aif2. 0: 不使用；1：使用
aif3config	是否使用 aif3. 0: 不使用；1：使用
gpio-spk	PA 使能引脚

说明

- **如果想要正常加载 *audiocodec* 声卡，需要把 *codec,platform,machine* 驱动都选上，即 *codec_used,i2s_used,sndcodec_used* 都置为 *1*；**
- ***headphonevol,headsetmicgain* 等值会在驱动初始化的时候设置，进入系统后还可以通过 *amixer* 工具对应控件进行再次修改；**
- **注意 *gpio-spk* 是否配置正确，是否有其他模块复用了该 *gpio;***

2.13.4.3 codec 数据通路

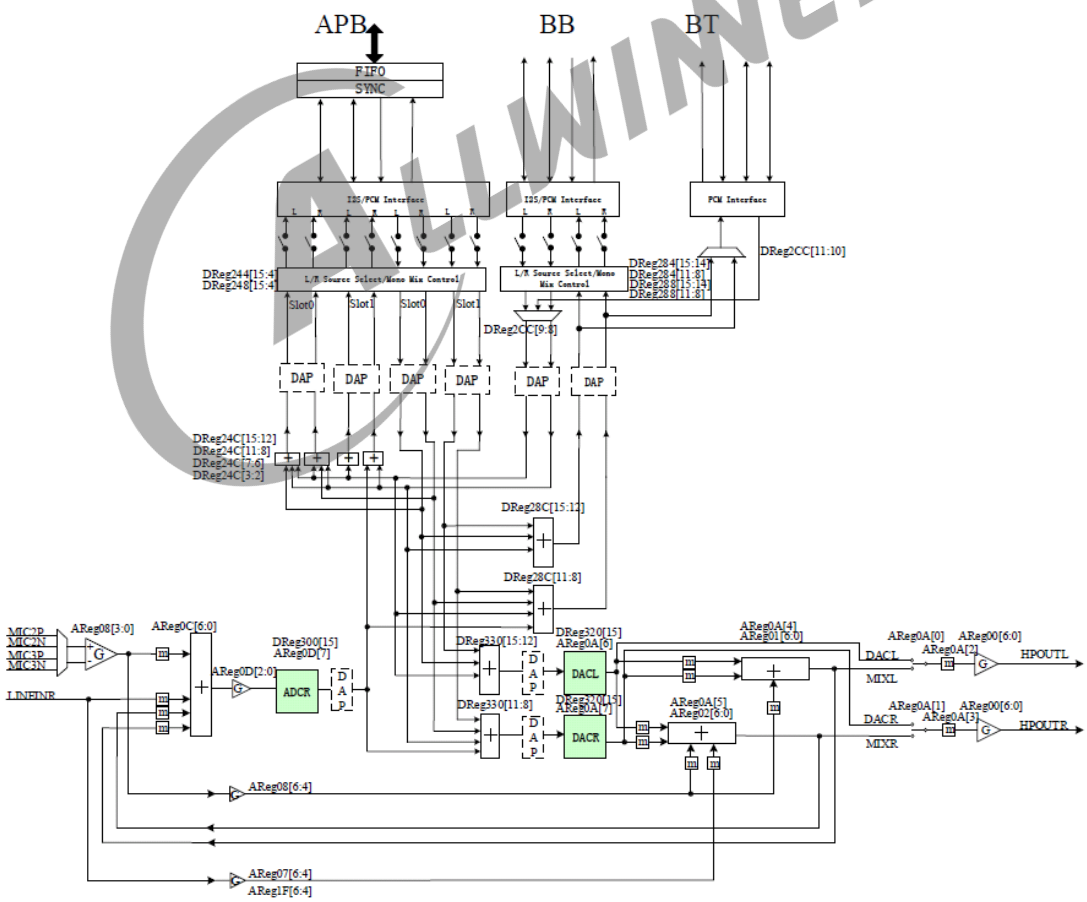


图 2-33: MR133 音频通路

```
播歌

AIF1DACL --> AIF1IN0L Mux --> DACL Mixer --> HP_L Mux --> HPOUTL --> Headphone
AIF1DACR --> AIF1IN0R Mux --> DACR Mixer --> HP_R Mux --> HPOUTR --> Headphone

录音
MIC2 --> MIC2 SRC --> MIC2 PGA --> RADC input Mixer --> AIF1 AD0R Mixer --> AIF1OUT0L Mux
    --> AIF1ADCL
MIC2 --> MIC2 SRC --> MIC2 PGA --> RADC input Mixer --> AIF1 AD0R Mixer --> AIF1OUT0R Mux
    --> AIF1ADCR
```

MR133 所有控件如下表：

控件名称	功能	数值
Headphone Switch	使能 Headphone	0: 关闭; 1: 开启
ADC input gain control	ADC 增益	0–7, 表示-4.5–6dB, 具体计算请看注释 1
ADC volume	ADC 左/右通道音量	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
I2S HUB FUNC	使能 hub 功能	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF1 DA0L Switch	AIF1 AD0L Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACL Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0L Mixer AIF2 DACR Switch	AIF1 AD0L Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer ADCR Switch	AIF1 AD0R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF1 DA0R Switch	AIF1 AD0R Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACL Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD0R Mixer AIF2 DACR Switch	AIF1 AD0R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 AD1L Mixer AIF2 DACL Switch	AIF1 AD1L Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer ADCR Switch	AIF1 AD1R Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF1 AD1R Mixer AIF2 DACR Switch	AIF1 AD1R Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF1 ADC timeslot 0 mixer gain	AIF1 ADC0L/ADC0R Mixer, 数字增益	0:0dB; 1:-6dB;
		对于 ADC0L Mixer, bit0:AIF2 DACR; bit1: 保留位;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
		bit2:AIF2 DACL; bit3:AIF2 DA0L; 对于 ADC0R Mixer, bit0:AIF2 DACL; bit1:ADCR; bit2:AIF2 DACR; bit3:AIF2 DA0R;
AIF1 ADC timeslot 0 volume	AIF1 ADC0L/ADC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 ADC timeslot 1 mixer gain	AIF1 ADC1L/ADC1R Mixer, 数字增益	0:0dB; 1:-6dB;
		对于 ADC1L Mixer, bit0: 保留位; bit1:AIF2 DACL; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACR;
AIF1 ADC timeslot 1 volume	AIF1 ADC1L/ADC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 0 volume	AIF1 DAC0L/DAC0R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1 DAC timeslot 1 volume	AIF1 DAC1L/DAC1R 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF1IN0L Mux	AIF1IN0L Mux 设置	0:AIF1_DA0L; 1:AIF1_DA0R; 2:AIF1_DA0L+AIF1_DA0R; 3:AIF1_DA0L+AIF1_DA0R 的平均值

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF1IN0R Mux	AIF1IN0R Mux 设置	0:AIF1_DA0R; 1:AIF1_DA0L; 2:AIF1_DA0L+AIF1_DA0R; 3:AIF1_DA0L+AIF1_DA0R 的平均值
AIF1IN1L Mux	AIF1IN1L Mux 设置	0:AIF1_DA1L; 1:AIF1_DA1R; 2:AIF1_DA1L+AIF1_DA1R; 3:AIF1_DA1L+AIF1_DA1R 的平均值
AIF1IN1R Mux	AIF1IN1R Mux 设置	0:AIF1_DA1R; 1:AIF1_DA1L; 2:AIF1_DA1L+AIF1_DA1R; 3:AIF1_DA1L+AIF1_DA1R 的平均值
AIF1OUT0L Mux	AIF1OUT0L Mux 设置	0:AIF1_AD0L; 1:AIF1_AD0R; 2:AIF1_AD0L+AIF1_AD0R; 3:AIF1_AD0L+AIF1_AD0R 的平均值
AIF1OUT0R Mux	AIF1OUT0R Mux 设置	0:AIF1_AD0R; 1:AIF1_AD0L; 2:AIF1_AD0L+AIF1_AD0R; 3:AIF1_AD0L+AIF1_AD0R 的平均值
AIF1OUT1L Mux	AIF1OUT1L Mux 设置	0:AIF1_AD1L; 1:AIF1_AD1R; 2:AIF1_AD1L+AIF1_AD1R; 3:AIF1_AD1L+AIF1_AD1R 的平均值
AIF1OUT1R Mux	AIF1OUT1R Mux 设置	0:AIF1_AD1R; 1:AIF1_AD1L; 2:AIF1_AD1L+AIF1_AD1R; 3:AIF1_AD1L+AIF1_AD1R 的平均值
AIF2 ADC mixer gain	AIF2 ADCL/ADCR Mixer, 数字增益	0:0dB; 1:-6dB; 对于 ADC1L Mixer, bit0: 保留位; bit1:AIF2 DACR; bit2:AIF2 DAC1L;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
		bit3:AIF2 DAC0L; 对于 ADC1R Mixer, bit0:ADCR; bit1:AIF2 DACL; bit2:AIF2 DAC1R; bit3:AIF2 DAC0R;
AIF2 ADC volume	AIF2 ADCL/ADCR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF2 ADL Mixer AIF1 DA0L Switch	AIF2 ADL Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启
AIF2 ADL Mixer AIF1 DA1L Switch	AIF2 ADL Mixer 设置，使能 AIF1 DA1L 通路	0: 关闭; 1: 开启
AIF2 ADL Mixer AIF2 DACR Switch	AIF2 ADL Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer ADCR Switch	AIF2 ADR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer AIF1 DA0R Switch	AIF2 ADR Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer AIF1 DA1R Switch	AIF2 ADR Mixer 设置，使能 AIF1 DA1R 通路	0: 关闭; 1: 开启
AIF2 ADR Mixer AIF2 DACL Switch	AIF2 ADR Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
AIF2 DAC volume	AIF2 DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
AIF2INL Mux	AIF2INL Mux 设置	0:AIF2_DACL; 1:AIF2_DACR; 2:AIF2_DACL+AIF2_DACR; 3:AIF2_DACL+AIF2_DACR 的平均值
AIF2INL Mux VIR switch aif2inl aif3	使能 aif3 -> aif2 通路	0: 关闭; 1: 开启
AIF2INL Mux switch aif2inl aif2	使能 aif2 -> aif2 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
AIF2INR Mux	AIF2INR Mux 设置	0:AIF2_DACR; 1:AIF2_DACL; 2:AIF2_DACL+AIF2_DACR; 3:AIF2_DACL+AIF2_DACR 的平均值
AIF2INR Mux VIR switch aif2inl aif3	使能 aif3 -> aif2 通路	0: 关闭; 1: 开启
AIF2INR Mux switch aif2inl aif2	使能 aif2 -> aif2 通路	0: 关闭; 1: 开启
AIF2OUTL Mux	AIF2OUTL Mux 设置	0:AIF2_ADCL; 1:AIF2_ADCR; 2:AIF2_ADCL+AIF2_ADCR; 3:AIF2_ADCL+AIF2_ADCR 的平均值
AIF2OUTR Mux	AIF2OUTR Mux 设置	0:AIF2_ADCR; 1:AIF2_ADCL; 2:AIF2_ADCL+AIF2_ADCR; 3:AIF2_ADCL+AIF2_ADCR 的平均值
AIF3OUT Mux	AIF3OUT Mux 设置	0:NULL; 1:AIF2_ADC_Left_Channel; 2:AIF2_ADC_Right_Channel
DAC mixer gain	DAC mixer 增益	0:0dB; 1:-6dB; 对于 DACL Mixer, bit0: 保留位; bit1:AIF2 DACL; bit2:AIF1 DAC1L; bit3:AIF1 DAC0L; 对于 DACR Mixer, bit0:ADCR; bit1:AIF2 DACR; bit2:AIF1 DAC1R; bit3:AIF1 DAC0R;
DAC volume	DACL/DACR 音量设置	0–0xff, 0 表示 mute, 0x1~0xff 表示-119.25dB~71.25dB, 0.75dB/step, 如 0xA0 表示 0dB
DACL Mixer AIF1DA0L Switch	DACL Mixer 设置，使能 AIF1 DA0L 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
DACL Mixer AIF1DA1L Switch	DACL Mixer 设置，使能 AIF1 DA1L 通路	0: 关闭; 1: 开启
DACL Mixer AIF2DACL Switch	DACL Mixer 设置，使能 AIF2 DACL 通路	0: 关闭; 1: 开启
DACR Mixer ADCR Switch	DACR Mixer 设置，使能 ADCR 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA0R Switch	DACR Mixer 设置，使能 AIF1 DA0R 通路	0: 关闭; 1: 开启
DACR Mixer AIF1DA1R Switch	DACR Mixer 设置，使能 AIF1 DA1R 通路	0: 关闭; 1: 开启
DACR Mixer AIF2DACR Switch	DACR Mixer 设置，使能 AIF2 DACR 通路	0: 关闭; 1: 开启
HP_L Mux	HP_L Mux 设置	0:DACL_HPL_Switch; 1:LOMixer_HPL_Switch
HP_R Mux	HP_R Mux 设置	0:DACR_HPR_Switch; 1:ROMixer_HPR_Switch
LINEINL/R to L_R output mixer gain	LINEINR to R output mixer 增益	0–7, 表示-4.5–6dB
Left Output Mixer DACL Switch	Left Output Mixer 设置, 使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Switch	Left Output Mixer 设置, 使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC2Boststage Switch	Left Output Mixer 设置, 使能 MIC2 通路	0: 关闭; 1: 开启
MIC2 BST stage to L_R outp mixer gain	MIC2 to L or R output Mixer 增益	0–7, 表示-4.5–6dB
MIC2 SRC	MIC2 SRC Mux 设置	0:MIC3; 1:MIC2
MIC2 boost AMP gain control	MIC2 增益	0–7, 0:0dB, 1~7:15–33dB
RADC input Mixer LINEINR Switch	RADC input Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
RADC input Mixer MIC2 boost Switch	RADC input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
RADC input Mixer l_output mixer Switch	RADC input Mixer 设置，使能 left output mixer 通路	0: 关闭; 1: 开启
RADC input Mixer r_output mixer Switch	RADC input Mixer 设置，使能 right output mixer 通路	0: 关闭; 1: 开启
Right Output Mixer DACL Switch	Right output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Switch	Right output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Right Output Mixer LINEINR Switch	Right output Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2Booststage Switch	Right output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Speaker Switch	Speaker 开关	0: 关闭; 1: 开启
digital volume	数字音量设置	0–63, 表示-73.08–0dB
headphone volume	headphone 音量设置	0–63,0 表示 mute; 1~63 表示-62dB–0dB

通路设置举例：

1. 播放通路

```
通过headphone耳机播放:

amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L';
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R';
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1;
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1;
amixer -Dhw:audiocodec cset name='HP_L Mux' 'DACL HPL Switch' ;
amixer -Dhw:audiocodec cset name='HP_R Mux' 'DACR HPR Switch';
amixer -Dhw:audiocodec cset name='Headphone Switch' 1;
```

2. 录音通路

```
通过模拟MIC2录音:

amixer -Dhw:audiocodec cset name='RADC input Mixer MIC2 boost Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD0R Mixer ADCR Switch' 1
amixer -Dhw:audiocodec cset name='AIF1 AD1R Mixer ADCR Switch' 1
amixer -Dhw:audiocodec cset name='AIF1OUT0L Mux' 'AIF1_AD0L'
amixer -Dhw:audiocodec cset name='AIF1OUT0R Mux' 'AIF1_AD0R'

amixer -Dhw:audiocodec cset name='MIC2 boost AMP gain control' 4
```

上面的播放, 录音功能主要用到 AIF1 接口。而 AIF2 主要用于语音通话,AIF3 主要用于蓝牙通话, 这类功能多用于通话平板, 这里不作介绍。

2.13.5 Daudio

硬件特性

- 两路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 2 路 MCLK 输出

2.13.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
```

2.13.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used          = 1

[daudio0]
daudio0_used             = 1
daudio_master            = 4
tdm_config               = 1
mclk_div                 = 1
audio_format             = 1
signal_inversion         = 1
pcm_lrck_period          = 32
;msb_lsb_first           = 0
slot_width_select        = 32
frametype                = 1
;tx_data_mode            = 1
;rx_data_mode            = 1
daudio_regulator         =
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

daudio 配置	daudio 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
daudio_regulator	daudio 供电配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.13.6 Dmic

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.13.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.13.6.2 sys_config 配置

配置如下:

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

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
dmic_used	是否使用 dmic 驱动。0：不使用；1：使用

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
snddmic_used	是否使用 snddmic 驱动。0：不使用；1：使用

- sys_config 中不需要配置 codec 驱动相关信息

因为 machine 驱动代码中默认配置了”dmic-codec” 作为 codec 驱动, 代码路径:

```
linux-4.9/sound/soc/codecs/dmic.c
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.13.7 标案音频测试方法

该章节主要介绍在标案上进行播歌的测试命令

2.13.7.1 播放

```
amixer -Dhw:audiocodec cset name='AIF1IN0L Mux' 'AIF1_DA0L';
amixer -Dhw:audiocodec cset name='AIF1IN0R Mux' 'AIF1_DA0R';
amixer -Dhw:audiocodec cset name='DACL Mixer AIF1DA0L Switch' 1;
amixer -Dhw:audiocodec cset name='DACR Mixer AIF1DA0R Switch' 1;
amixer -Dhw:audiocodec cset name='HP_L Mux' 'DACL HPL Switch' ;
amixer -Dhw:audiocodec cset name='HP_R Mux' 'DACR HPR Switch';
amixer -Dhw:audiocodec cset name='Headphone Switch' 1;

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

可通过下面命令调节硬件上的模拟音量:

```
amixer -Dhw:audiocodec cset name='headphone volume' 50
```

2.14 R328 音频接口

2.14.1 硬件资源

R328 音频接口丰富，包含 6 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Daudio2,Dmic,Spdif。

另外还支持 MAD 作语音唤醒检测 (详细请看R328 MAD 章节)。

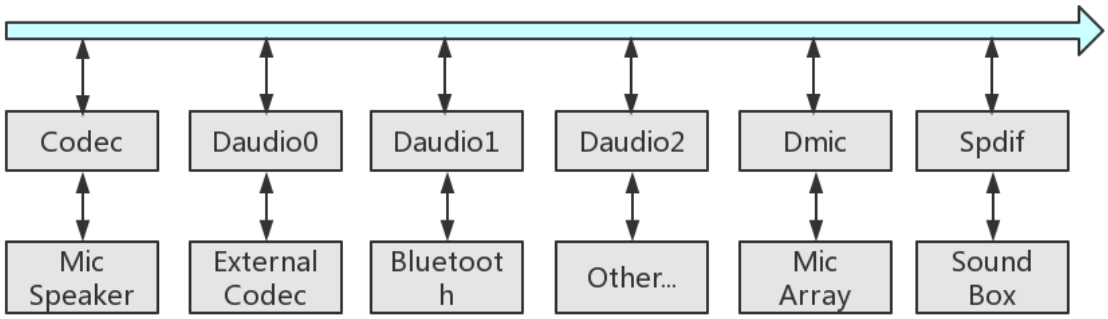


图 2-34: R328 音频硬件框图

2.14.2 时钟源

R328 中，6 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

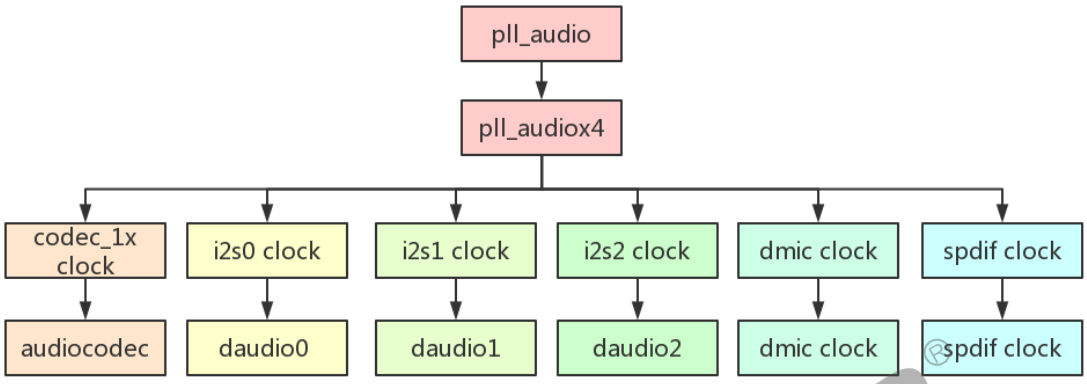


图 2-35: R328 时钟源

2.14.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── spdif-utils.c                  // spdif codec驱动
├── sun8iw18-codec.c               // codec驱动
├── sun8iw18-codec.h
├── sun8iw18-sndcodec.c            // codec machine驱动
├── sunxi-cpudai.c                 // codec platform驱动
├── sunxi-daudio.c                 // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-dmic.c                   // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-mad.c                    // 提供MAD相关功能接口
├── sunxi-mad.h
├── sunxi-pcm.c                    // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi_rw_func.c                // 通用文件，读写模拟/数字寄存器的接口
├── sunxi_rw_func.h
├── sunxi-snddaudio.c              // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-snddmic.c                // dmic machine驱动
├── sunxi-snddmic.h
├── sunxi-sndspdif.c               // spdif machine驱动
├── sunxi-spdif.c                  // spdif platform驱动
└── sunxi-spdif.h

linux-4.9/sound/soc/codecs/dmic.c  // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c    // daudio codec驱动
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.14.4 AudioCodec

硬件特性

- 一路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 三路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 一路模拟输出：一路差分输出 lineoutP/N，支持单端 lineout 输出
- 三路模拟输入：MIC1,MIC2,MIC3
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 5 段 DRC
- DAC FIFO 长度 128*24bits, ADC FIFO 长度 128*24bits

2.14.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun8iw18 Codec Support
```

2.14.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used = 0x1
;-----------------------------------------------------------------------------
[cpudai]
cpudai_used = 0x1
;-----------------------------------------------------------------------------
[codec]
codec_used = 0x1
digital_vol = 0x0
lineout_vol =0x1a
mic1gain = 0x4
mic2gain = 0x4
mic3gain = 0x0
adcgain = 0x3
adcagc_cfg = 0x0
adcdrc_cfg = 0x0
adchpf_cfg = 0x0
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
dacdrc_cfg = 0x0
dachpf_cfg = 0x0
pa_ctl_level = 0x1
pa_msleep_time = 160
gpio-spk = port:PH9<1><1><1><1>
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用

cpudai 配置，即 platform 驱动的相关配置

cpudai 配置	cpudai 配置说明
cpudai_used	是否使用 cpudai 驱动。0：不使用；1：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
digital_vol	初始化 digital volume，可设定范围 0~0x3f, 表示 0~-73.08dB, -1.16dB/step
lineout_vol	lineout volume，可设定范围 0~0x1f, 表示-43.5dB~0dB, 1.5dB/step
mic1gain	mic1 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
mic2gain	mic2 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
mic3gain	mic3 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB. 如果作为 aec 回路，则需要设置为 0dB
adcgain	adc 增益，可设定范围 0~0x7, 表示-4.5~6dB, 1.5dB/step, 一般设置 0x3, 即 0dB
adcdrc_cfg	是否使用 adcdrc. 0: 不适用；1：使用
adchpf_cfg	是否使用 adchpf. 0: 不适用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不适用；1：使用
dachpf_cfg	是否使用 dachpf. 0: 不适用；1：使用
pa_ctl_level	PA 引脚使能方式。0: 低电平有效；1：高电平有效
pa_msleep_time	操作 PA 之后的延时时间 (用来避免 pop 音)
gpio-spk	PA 使能引脚

版权所有 © 珠海全志科技股份有限公司。保留一切权利

说明

- 如果想要正常加载 *audiocodec* 声卡，需要把 *codec,platform,machine* 驱动都选上，即 *codec_used,cpudai_used,sndcodec_used* 都置为 *1*；
- *digital_vol,lineout_vol* 等值会在驱动初始化的时候设置，进入系统后还可以通过 *amixer* 工具对应控件进行再次修改；
- 注意 *gpio-spk* 是否配置正确，是否有其他模块复用了该 *gpio;*
- 注意 *pa_ctl_level*，实际功放的 *PA* 引脚是高电平有效，还是低电平有效

2.14.4.3 codec 数据通路

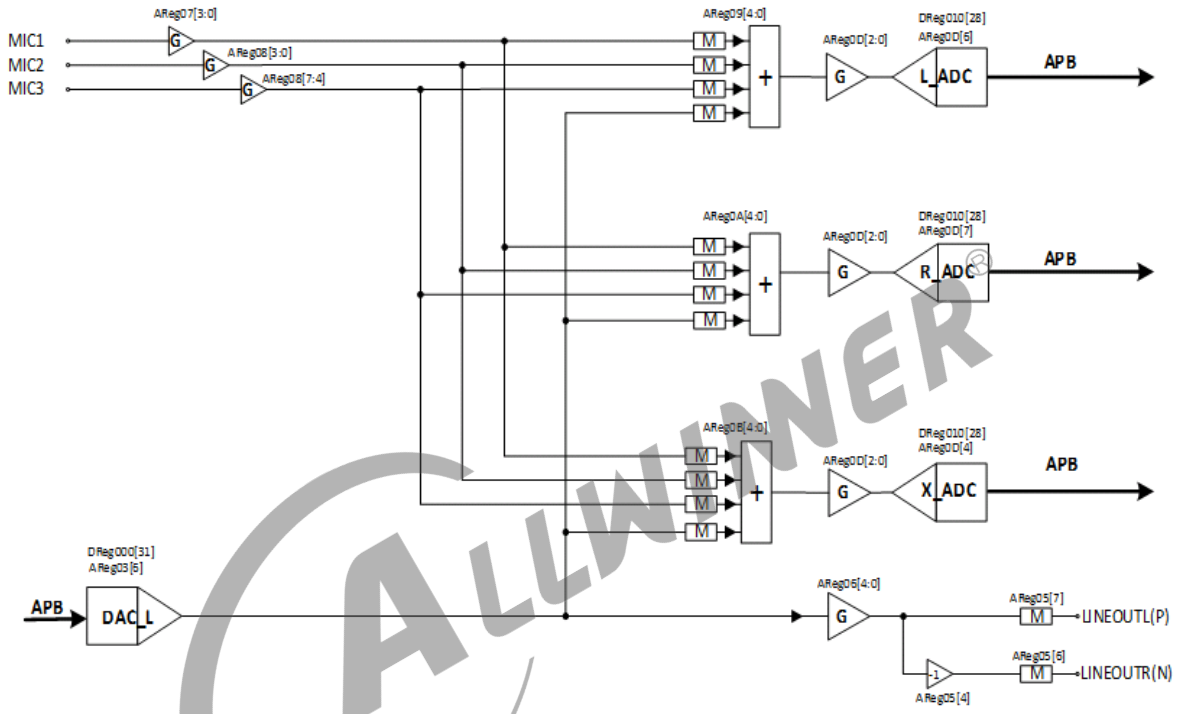


图 2-36: R328 音频通路

```
播歌
Playback --> DACL --> Left LINEOUT Mux --> LINEOUTL --> External Speaker
Playback --> DACR --> Right LINEOUT Mux --> LINEOUTR --> External Speaker

录音
MIC1 --> MIC1 PGA ---> Left Input Mixer --> ADCL --> Capture
MIC2 --> MIC2 PGA ---> Right Input Mixer --> ADCL --> Capture
MIC3 --> MIC3 PGA ---> Xadc Input Mixer --> ADCL --> Capture
```

R328 所有控件如下表：

控件名称	功能	数值
Lineout Switch	使能 lineout	0: 关闭; 1: 开启
ADC gain volume	ADC 增益	0–7, 表示-4.5–6dB, 具体计算请看注释 1
External Speaker Switch	使能 lineout 以及 PA	0: 关闭; 1: 开启

控件名称	功能	数值
LINEOUT volume	lineout 音量设置	0–31, 表示-43.5–0dB, 具体计算请看注释 2
Left Input Mixer DACL Switch	Left Input Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Input Mixer MIC1 Boost Switch	Left Input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Input Mixer MIC2 Boost Switch	Left Input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Left Input Mixer MIC3 Boost Switch	Left Input Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
Left LINEOUT Mux	Left Lineout Mux 设置	0:DACL; 1:NULL(空)
Right LINEOUT Mux	Right Lineout Mux 设置	0:NULL(空); 1:DACL
MIC1 gain volume	MIC1 Boost AMP gain	0–7, 0:0dB, 1~7:15–33dB, 具体计算请看注释 3
MIC2 gain volume	MIC2 Boost AMP gain	与 MIC1 gain volume 设置一样
MIC3 gain volume	MIC3 Boost AMP gain	与 MIC1 gain volume 设置一样
Right Input Mixer DACL Switch	Right Input Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Input Mixer MIC1 Boost Switch	Right Input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Input Mixer MIC2 Boost Switch	Right Input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Right Input Mixer MIC3 Boost Switch	Right Input Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
Xadc Input Mixer DACL Switch	Xadc Input Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Xadc Input Mixer MIC1 Boost Switch	Xadc Input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Xadc Input Mixer MIC2 Boost Switch	Xadc Input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Xadc Input Mixer MIC3 Boost Switch	Xadc Input Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
codec hub mode	使能 audiocodec hub 功能	0: 关闭; 1: 开启
digital volume	数字端音量设置	0–63, 表示-73.08–0dB, 具体计算请看注释 4

- 注释 1

```
ADC gain volume计算方法:
应用层可设置范围:0～7
对应实际硬件设置的范围:-4.5~6dB, step: 1.5dB
换算方法:-4.5+(n\*1.5)
举例，设置0dB:
-4.5+(n\*1.5) = 0
n = 3
所以应用层上输入下面命令设置为0dB:
amixer -D hw:audiocodec cset name='ADC gain volume' 3
```

- 注释 2

```
LINEOUT volume计算方法:
应用层可设置范围: 0～31 (设置为0或者1时，就是mute)
对应实际硬件设置的范围: -43.5~0dB, step: 1.5dB
换算方法:-43.5+((n-2)\*1.5)
举例1，设置0dB:
-43.5+((n-2)\*1.5) = 0
n = 31
所以应用层上输入下面命令设置为0dB:
amixer -D hw:audiocodec cset name='LINEOUT volume' 31
举例2，设置-6dB:
-43.5+((n-2)\*1.5) = -6
n = 27
所以应用层上输入下面命令设置为-6dB:
amixer -D hw:audiocodec cset name='LINEOUT volume' 27
```

- 注释 3

```
MIC1 gain volume计算方法:
应用层可设置范围: 0～7 (设置为0时，就是0dB)
对应实际硬件设置的范围: 0dB或者15~33dB, step: 3dB
换算方法:15+((n-1)*3)
举例，设置24dB:
15+((n-1)*3) = 24
n = 4
所以应用层上输入下面命令设置为24dB:
amixer -D hw:audiocodec cset name='MIC1 gain volume' 4
```

- 注释 4

```
digital volume计算方法:
应用层可设置范围: 0～63
对应实际硬件设置的范围: -73.08~0dB, step: 1.16dB
换算方法:-73.08+(n*1.16)
举例1，设置0dB:
-73.08+(n*1.16) = 0
n = 63
所以应用层上输入下面命令设置为0dB:
amixer -D hw:audiocodec cset name='digital volume' 63
举例2，设置-5.8dB:
-73.08+(n*1.16) = -5.8
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
n = 58
所以应用层上输入下面命令设置为-5.8dB:
amixer -D hw:audiocodec cset name='digital volume' 58
```

通路设置举例：

1. 播放通路

```
通过Speaker播放，差分输出:
amixer -D hw:audiocodec cset name='External Speaker Switch' 1
amixer -D hw:audiocodec cset name='digital volume' 63
amixer -D hw:audiocodec cset name='LINEOUT volume' 25
amixer -D hw:audiocodec cset name='Right LINEOUT Mux' 1
```

2. 录音通路

```
通过模拟MIC1, MIC2录音:
amixer -D hw:audiocodec cset name='Left Input Mixer MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='Right Input Mixer MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 gain volume' 4
amixer -D hw:audiocodec cset name='MIC2 gain volume' 4
```

3. 常用 AEC 通路

```
有两种AEC回路方式，具体看硬件如何设计

  1) 外部AEC

  MIC1,MIC2录音; MIC3作为AEC,外部SPKP/N连接到MIC3.
amixer -D hw:audiocodec cset name='Left Input Mixer MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='Right Input Mixer MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='Xadc Input Mixer MIC3 Boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 gain volume' 4
amixer -D hw:audiocodec cset name='MIC2 gain volume' 4
amixer -D hw:audiocodec cset name='MIC3 gain volume' 0

  2)内部AEC(可省去外部AEC电路)

  MIC1,MIC2录音; MIC3作为AEC,使能内部DACL到MIC3的通路.
amixer -D hw:audiocodec cset name='Left Input Mixer MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='Right Input Mixer MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='Xadc Input Mixer DACL Switch' 1
amixer -D hw:audiocodec cset name='MIC1 gain volume' 4
amixer -D hw:audiocodec cset name='MIC2 gain volume' 4
```

2.14.5 Daudio

硬件特性

- 三路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 3 路 MCLK 输出

2.14.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
```

2.14.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used          = 0
daudio_master            = 4
audio_format             = 1
signal_inversion         = 1

[daudio0]
daudio0_used             = 0
slot_width_select        = 32
pcm_lrck_period          = 128
msb_lsb_first            = 0
sign_extend              = 0
frametype                = 0
mclk_div                 = 1
tdm_config               = 1
tx_data_mode             = 0
rx_data_mode             = 0
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

snddaudio 配置	snddaudio 配置说明
	4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

注意事项：

- daudio machine 驱动的配置 (snddaudio), 一般来说还需要配置 codec name 以及 codec dai name

1. 例如 daudio0 使用了 AC108 作为外挂 codec:

```
[snddaudio0]
snddaudio0_used = 1
sunxi,snddaudio-codec = "ac108.1-003b";
sunxi,snddaudio-codec-dai = "ac108-pcm0";
daudio_master           = 4
audio_format            = 1
signal_inversion        = 2

[daudio0]
daudio0_used            = 1
slot_width_select       = 32
pcm_lrck_period         = 128
msb_lsb_first           = 0
sign_extend             = 0
frametype               = 0
mclk_div                = 1
tdm_config              = 1
tx_data_mode            = 0
rx_data_mode            = 0

注意名称需要与codec驱动中配置的名称一致，如ac108驱动，路径：
linux-4.9/sound/soc/codecs/ac108.c
代码中snd_soc_register_codec注册codec驱动，其中codec device name为ac108.1-003b,
codec dai name为ac108-pcm0
```

2. 例如 daudio2 与 bluetooth 模组相连 (没有实际的 codec 驱动)，那么这时候 codec name, codec dai name 需要配置为 dummy codec, 可以如下配置：

```
[snddaudio2]
snddaudio2_used = 1
sunxi,snddaudio-codec = "snd-soc-dummy"
sunxi,snddaudio-codec-dai = "snd-soc-dummy-dai"
daudio_master           = 1
audio_format            = 5
signal_inversion        = 2


因为驱动中解析snddaudio-codec等字段时，判断出错的时候则使用默认codec"snd-soc-dummy",
所以如果sunxi,snddaudio-codec和sunxi,snddaudio-codec-dai不配置，或者配置为空的时候，
则默认使用dummy codec:
[snddaudio2]
snddaudio2_used = 1
daudio_master           = 1
audio_format            = 5
signal_inversion        = 2

或者
[snddaudio2]
snddaudio2_used = 1
sunxi,snddaudio-codec =
sunxi,snddaudio-codec-dai =
daudio_master           = 1
audio_format            = 5
signal_inversion        = 2
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.14.6 Dmic

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.14.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.14.6.2 sys_config 配置

配置如下:

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

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
dmic_used	是否使用 dmic 驱动。0：不使用；1：使用

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
snddmic_used	是否使用 snddmic 驱动。0：不使用；1：使用

- sys_config 中不需要配置 codec 驱动相关信息

因为 machine 驱动代码中默认配置了”dmic-codec” 作为 codec 驱动, 代码路径:

```
linux-4.9/sound/soc/codecs/dmic.c
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.14.7 SPDIF

硬件特性

- 支持 S/PDIF_OUT 和 S/PDIF_IN
- 支持 mono 和 stereo 模式 (mono 模式下由硬件自动拓展为 stereo)
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 输入支持 44.1KHz,48KHz 采样率
- 输出和输入支持 16bit,24bit 采样精度

2.14.7.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner SPDIF Support
```

2.14.7.2 sys_config 配置

```
[sndspdif]
sndspdif_used = 0
[spdif]
spdif_used = 0
```

spdif 配置，即 platform 驱动的相关配置

spdif 配置	spdif 配置说明 |
spdif_used	是否使用 spdif 驱动。0：不使用；1：使用 |

sndspdif 配置，即 machine 驱动的相关配置

sndspdif 配置	sndspdif 配置说明
sndspdif_used	是否使用 sndspdif 驱动。0：不使用；1：使用

- sys_config 中不需要配置 codec 驱动相关信息

版权所有 © 珠海全志科技股份有限公司。保留一切权利

因为 machine 驱动代码中默认配置了”spdif-utils” 作为 codec 驱动, 代码路径：

```
linux-4.9/sound/soc/sunxi/spdif-utils.c
```

2.14.8 MAD

硬件特性

- 支持三路 I2S，一路 DMIC PCM 音频传输接口，时分复用，固定 16bit
- 支持 16KHz,48KHz 采样率
- 支持基于能量识别的语音检测模块 LPSD
- 支持一块 128KB 的 SRAM, 可用于保存音频数据

2.14.8.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Mad Support
```

2.14.8.2 sys_config 配置

```
[mad]
mad_used = 1
lpsd_clk_src_cfg = 0
standby_sram_io_type = 1
```

mad 配置	mad 配置说明
mad_used	是否使用 mad 驱动。0：不使用；1：使用

2.14.8.3 mixer 控件说明

1. mad 绑定到 I2S

控件名称	功能	数值
daudio bind mad Function	是否绑定 MAD 功能	0: 不绑定; 1: 绑定
lpsd channel sel Function	选择作为能量唤醒的通道	0: 通道 0; 1: 通道 1;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
mad_standby channel sel Function	设定休眠时 mad 录音通道数	如此类推，最高可指定通道 7 0: 表示使用实际录音通道数; 1: 表示只录制两通道; 2: 表示只录制四通道;

2. mad 绑定到 dmic

控件名称	功能	数值
dmic bind mad Function	是否绑定 MAD 功能	0: 不绑定; 1: 绑定
lpsd channel sel Function	选择作为能量唤醒的通道	0: 通道 0; 1: 通道 1; 如此类推，最高可指定通道 7
mad_standby channel sel Function	设定休眠时 mad 录音通道数	0: 表示使用实际录音通道数; 1: 表示只录制两通道 2: 表示只录制四通道

2.14.8.4 使用说明

固件上的配置，只要修改 sys_config 以及内核配置即可。

应用上需要使能 MAD 相关的 mixer control。

I2S 设置举例，例如使用的是 AC108:

```
mad使能,绑定mad到daudio中
amixer -Dhw:sndac1081003b cset name='daudio bind mad Function' 1
设置通道0作为唤醒通道
amixer -Dhw:sndac1081003b cset name='lpsd channel sel Function' 0
设定mad standby时,录音的通道数
amixer -Dhw:sndac1081003b cset name='mad_standby channel sel Function' 2
```

DMIC 设置举例:

```
amixer -Dhw:snddmic cset name='dmic bind mad Function' 1
amixer -Dhw:snddmic cset name='lpsd channel sel Function' 0
amixer -Dhw:snddmic cset name='mad_standby channel sel Function' 0
```

然后应用正常进行录音即可，如果需要进入休眠，有下面几点必须实现的:

1. 暂定录音、播放。snd_pcm_pause 将 playback,capture 均暂停;

2. 设置 wakeup_count。更新当前唤醒次数;
3. 进入休眠。写 mem 到/sys/power/state 即可;

Tina SDK 中有一个能量唤醒 demo 可供参考。

make menuconfig 选中 mad-demo 软件包

```
Allwinner  --->
  <*> mad-demo
```

执行 mad-ac108-demo，默认配置 (脚本/usr/bin/mad-ac108-demo 上设定了默认配置):

- 使用通道 0 作为唤醒通道;
- 录制 4 通道，16bit, 16K;
- 每次录音 5s 后进入休眠，可通过语音能量唤醒;

执行 mad-dmic-demo，默认配置 (脚本/usr/bin/mad-dmic-demo 上设定了默认配置):

- 使用通道 0 作为唤醒通道;
- 录制 4 通道，16bit, 16K;
- 每次录音 5s 后进入休眠，可通过语音能量唤醒;

如果想查看录音数据，可以增加 dump 参数，例如 mad-dmic-demo dump，录音文件保存在/mnt/UDISK/目录下。

2.14.8.5 能量唤醒阈值参数

能量唤醒模块 lpsd，识别能量主要有两个方向，瞬时能量和累计能量（前者比如是关门声，后者比如是不断说话）能量检测参数配置均在/sys/module/sunxi_mad/parameters/目录下

lpsd_rrun 和 lpsd_rstop 的推荐值：

lpsd_rrun	lpsd_rstop
77	88
77	108
77	128
77	148

1. 瞬时能量检测参数，主要是 lpsd_rrun 和 lpsd_rstop。

- 一般我们只对 stop 值进行修改;
- 如果录音数据经常缺少唤醒词的第一个字，则可以尝试降低 stop 值，可以有效提高唤醒词数据的完整性。但同时会提高误唤醒率，环境噪音也会很容易触发能量检测，唤醒系统;
- 如果想要降低误唤醒率 (环境噪音造成唤醒)，则可以尝试提高 stop 值。同样的，这会导致一些唤醒词录音数据不完整，例如一些音量较低，音调较低的语料;
- 唤醒词识别率以及误唤醒率无法同时兼得，客户需要根据实际需求、场景，权衡配置参数;

2. 累积能量检测参数，主要是 lpsd_th。

- 我们建议使用默认值 1200。建议修改范围 50~1200;

2.14.8.6 注意事项

1. MAD 绑定动作，需要在应用打开声卡前就设置好;
2. 应用操作上的一些要求，具体请查看《MAD 使用说明》章节;
3. 如果读取 wakeup_count 时一直阻塞，说明当前仍有 wake_lock 处于激活状态，例如 usb 线连接着 PC，usb 驱动会保持一个 wake_lock，不让系统进入休眠，所以需要拔掉 usb 或者连接到 usb 适配器上，或者改动代码，去掉 usb 驱动中 wake_lock 的使用;

2.14.9 VAD

VAD 是基于 MAD 实现的，可以通过内部 AudioCodec 的 ADC 采集音频数据，并作能量唤醒。由于硬件上 MAD 功能只能用于 I2S 或者 DMIC，内部 codec 无法直接关联到 MAD，因此通过 I2S 作为音频数据的桥梁，实现了 VAD 功能，使得模拟 MIC 也可以利用 MAD 功能作能量唤醒。

VAD 完整的数据通路：

```
ADC RxFiFo ---> I2S TxFiFo ---> I2S RxFiFo ---> MAD SRAM ---> MEM
```

2.14.9.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         [*] Allwinner I2S PCM DMA MAP Support
         <*> Allwinner Mad Support
         <*> Allwinner Sun8iw18 Codec Support
         <*> Allwinner Digital Audio Support
```

2.14.9.2 sys_config 配置

需要使能 MAD 配置:

```
[mad]
mad_used = 1
lpsd_clk_src_cfg = 0
standby_sram_io_type = 1
```

mad 配置	mad 配置说明
mad_used	是否使用 mad 驱动。0：不使用；1：使用

2.14.9.3 mixer 控件说明

控件名称	功能	数值
codec I2S Port	指定 VAD 使用的 I2S	0: 不适用; 1: 使用 I2S0; 2: 使用 I2S1; 3: 使用 I2S2
sndcodec bind mad Function	是否绑定 MAD 功能	0: 不绑定; 1: 绑定
lpsd channel sel Function	选择作为能量唤醒的通道	0: 通道 0; 1: 通道 1; 如此类推，最高可指定通道 7
mad_standby channel sel Function	设定休眠时 mad 录音通道数	0: 表示使用实际录音通道数; 1: 表示只录制两通道 2: 表示只录制四通道

注意:

对于控件 “codec I2S Port”，需要指定实际没有使用 (sys_config 没有使能的) 的一路 I2S。设置举例:

- 实际没有使用 I2S0(sys_config 中 snddaudio0,audio0 均没有配置), 那么这里可以设置为 1, 表示 VAD 使用 I2S0;
- 实际没有使用 I2S1(sys_config 中 snddaudio1,audio1 均没有配置), 那么这里可以设置为 2, 表示 VAD 使用 I2S1;

设置举例:

```
amixer -Dhw:audiocodec cset name='codec I2S Port' 2
amixer -Dhw:audiocodec cset name='sndcodec bind mad Function' 1
amixer -Dhw:audiocodec cset name='lpsd channel sel Function' 0
amixer -Dhw:audiocodec cset name='mad_standby channel sel Function' 0
```

2.14.9.4 使用说明

VAD 的使用与 MAD 类似。

固件上的配置，只要修改 sys_config 以及内核配置即可。

应用上除了打开内部 audiocodec 的录音通路之外，还需要下面一些配置：

```
vad需要使用一路i2s,这里指定使用i2s1
amixer -Dhw:audiocodec cset name='codec I2S Port'  2

使能mad,绑定mad到audiocodec中
amixer -Dhw:audiocodec cset name='sndcodec bind mad Function' 1

设置通道0作为唤醒通道
amixer -Dhw:audiocodec cset name='lpsd channel sel Function' 0
```

然后应用正常进行录音即可，如果需要进入休眠，有下面几点必须实现的：

1. 暂定录音、播放。snd_pcm_pause 将 playback,capture 均暂停
2. 设置 wakeup_count。更新当前唤醒次数
3. 进入休眠。写 mem 到/sys/power/state 即可

Tina SDK 中有一个能量唤醒 demo 可供参考

make menuconfig 选中 mad-demo 软件包

```
Allwinner  --->
  <*> mad-demo
```

执行 vad-demo，默认配置 (脚本/usr/bin/vad-demo 上设定了默认配置):

- 使用 I2S1
- 录制 2 通道，16bit, 16K
- 每次录音 5s 后进入休眠，可通过语音能量唤醒

如果想查看录音数据，可以执行 vad-demo dump, 录音文件保存在/mnt/UDISK/vad-test.wav

能量唤醒阈值的调整，可以参考《能量唤醒阈值参数》

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.14.9.5 注意事项

VAD 同样需要注意MAD 注意事项章节中提到的几点。

另外需要注意，VAD 隐式使用了一路 I2S，所以硬件上需要保留一路 I2S，并且 sys_config 中不能使能该 I2S 配置

2.14.10 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.14.10.1 播放

```
通过Speaker播放

amixer -D hw:audiocodec cset name='External Speaker Switch' 1
amixer -D hw:audiocodec cset name='Right LINEOUT Mux' 1

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

2.14.10.2 录音

```
通过MIC1,MIC2录制两通道
amixer -D hw:audiocodec cset name='Left Input Mixer MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='Right Input Mixer MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost volume' 4
amixer -D hw:audiocodec cset name='MIC2 boost volume' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.15 T7 音频接口

T7 包含 6 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Daudio2,Dmic,Spdif。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.1 硬件资源

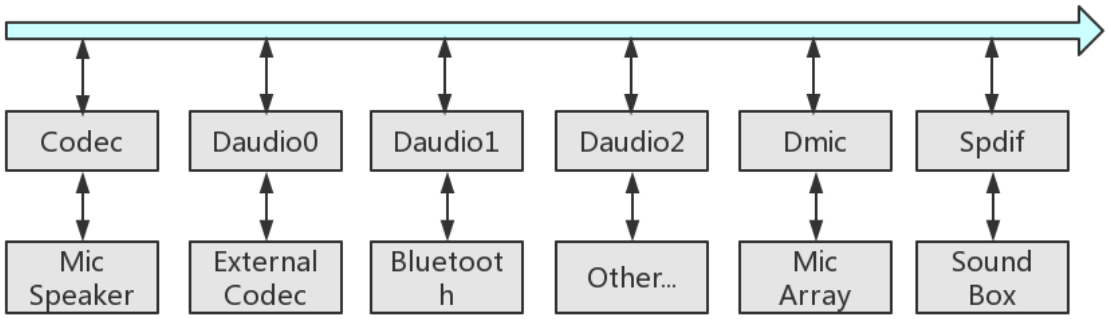


图 2-37: T7 音频硬件框图

2.15.2 时钟源

T7 中，6 个音频模块的时钟源均来自 pll_audio。

pll_audio 可以输出 24.576M 或者 22.5792M 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

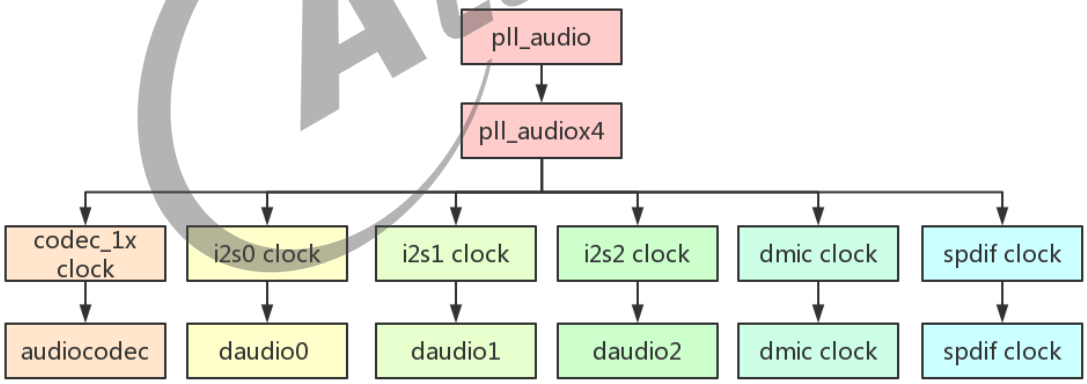


图 2-38: T7 时钟源

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.3 代码结构

```
linux-4.9/sound/soc/sunxi/
├── spdif-utils.c                    // spdif codec驱动
├── sun8iw17-codec.c                 // codec驱动
├── sun8iw17-codec.h
├── sun8iw17-sndcodec.c              // codec machine驱动
├── sunxi-cpudai.c                   // codec platform驱动
├── sunxi-daudio.c                   // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-dmic.c                     // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-mad.c                      // 提供MAD相关功能接口
├── sunxi-mad.h
├── sunxi-pcm.c                      // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi_rw_func.c                  // 通用文件，读写模拟/数字寄存器的接口
├── sunxi_rw_func.h
├── sunxi-snddaudio.c                // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-snddmic.c                  // dmic machine驱动
├── sunxi-snddmic.h
├── sunxi-sndspdif.c                 // spdif machine驱动
├── sunxi-spdif.c                    // spdif platform驱动
└── sunxi-spdif.h

linux-4.9/sound/soc/codecs/dmic.c    // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c      // daudio codec驱动
```

2.15.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 三路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出：
  - 一路立体声输出 LINEOUTL,LINEOUTR，支持单端 lineout 输出
  - 一路立体声输出 PHONEOUTP,PHONEOUTN
- 三路模拟输入：MIC1,MIC2,MIC3
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 DRC

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun8iw17 Codec Support
```

2.15.4.2 sys_config 配置

```
[sndcodec]
sndcodec_used = 0x1
;-----------------------------------------------------------------------------
[cpudai]
cpudai_used = 0x1
;-----------------------------------------------------------------------------
[codec]
codec_used = 0x1
digital_vol = 0x0
lineout_vol =0x1a
mic1gain = 0x4
mic2gain = 0x4
mic3gain = 0x0
adcgain = 0x3
adcagc_cfg = 0x0
adcdrc_cfg = 0x0
adchpf_cfg = 0x0
dacdrc_cfg = 0x0
dachpf_cfg = 0x0
pa_ctl_level = 0x1
pa_msleep_time = 160
gpio-spk = port:PH9<1><1><1><1>
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
sndcodec_used	是否使用 sndcodec 驱动。0：不使用；1：使用

cpudai 配置，即 platform 驱动的相关配置

cpudai 配置	cpudai 配置说明
cpudai_used	是否使用 cpudai 驱动。0：不使用；1：使用

codec 配置，即内置 audiocodec 驱动的相关配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
codec_used	是否使用 codec 驱动。0：不使用；1：使用
digital_vol	初始化 digital volume，可设定范围 0~0x3f, 表示 0~-73.08dB, -1.16dB/step
lineout_vol	lineout volume，可设定范围 0~0x1f, 表示-43.5dB~0dB, 1.5dB/step
mic1gain	mic1 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
mic2gain	mic2 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB
mic3gain	mic3 增益，可设定范围 0~0x7, 0:0dB, 1~7:15~33dB, 3dB/step, 一般设置 0x4, 即 24dB. 如果作为 aec 回路，则需要设置为 0dB
adcgain	adc 增益，可设定范围 0~0x7, 表示-4.5~6dB, 1.5dB/step, 一般设置 0x3, 即 0dB
adcdrc_cfg	是否使用 adcdrc. 0: 不适用；1：使用
adchpf_cfg	是否使用 adchpf. 0: 不适用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不适用；1：使用
dachpf_cfg	是否使用 dachpf. 0: 不适用；1：使用
pa_ctl_level	PA 引脚使能方式。0: 低电平有效；1：高电平有效
pa_msleep_time	操作 PA 之后的延时时间 (用来避免 pop 音)
gpio-spk	PA 使能引脚

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.4.3 codec 数据通路

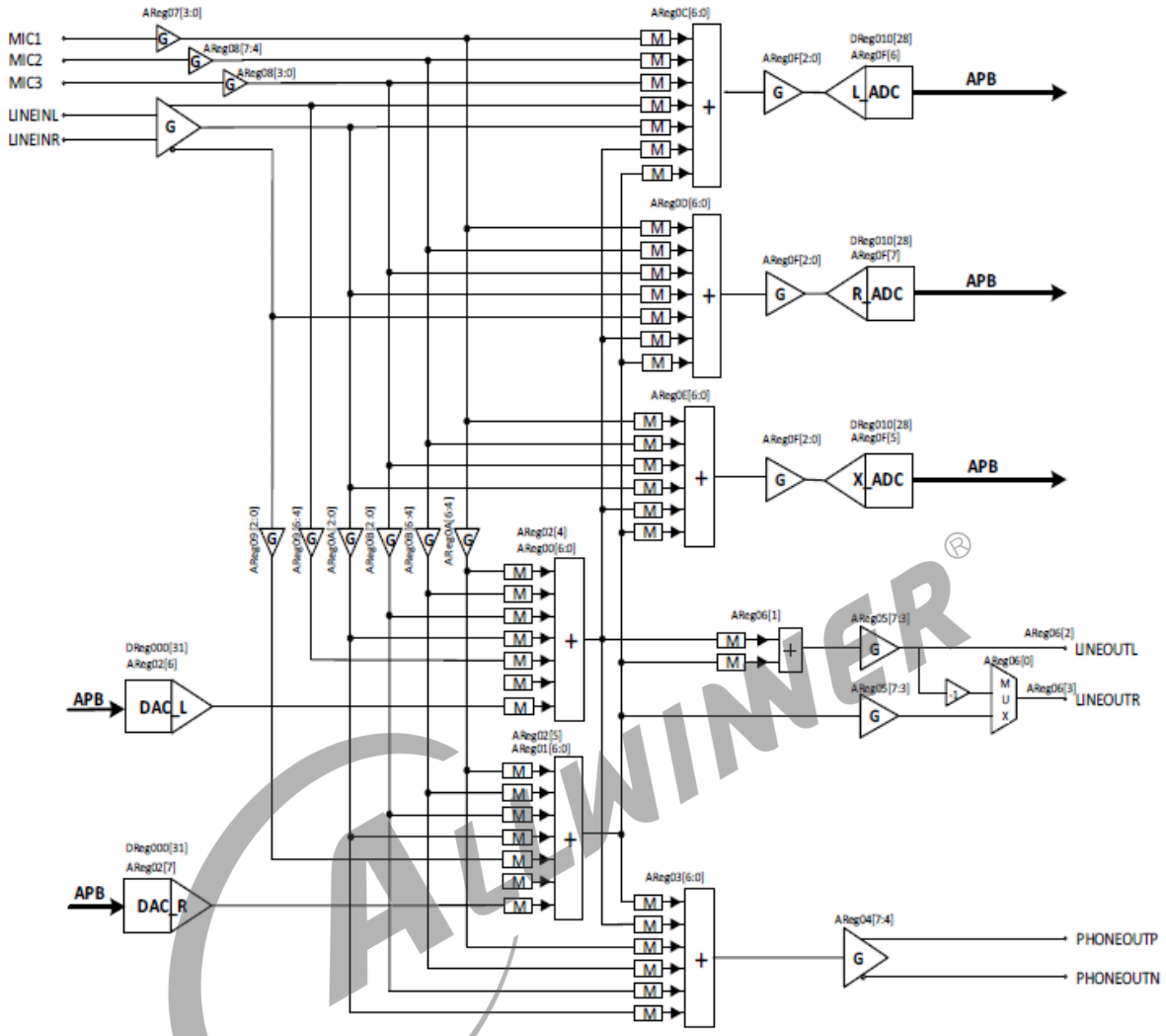


图 2-39: T7 音频通路

```
播歌
Playback --> DACL --> Left Output Mixer --> Left LINEOUT Mux --> LINEOUT
Playback --> DACR --> Right Output Mixer --> Right LINEOUT Mux --> LINEOUT

Playback --> DACL --> Left Output Mixer --> Phone Out Mixer --> SPKPA DRV --> PHONEOUTP -->
     Phoneout Speaker
Playback --> DACR --> Right Output Mixer --> Phone Out Mixer --> SPKPA DRV --> PHONEOUTN
    --> Phoneout Speaker

录音
MIC1 --> MIC1 PGA ---> Left Input Mixer --> ADCL --> Capture
MIC2 --> MIC2 PGA ---> Right Input Mixer --> ADCL --> Capture
MIC3 --> MIC3 PGA ---> Xadc Input Mixer --> ADCL --> Capture
```

T7 所有控件如下表：

控件名称	功能	数值
Phone Out Mixer LOMIX Switch	Phone Out Mixer 设置, 使能 loutput	0: 关闭; 1: 开启
Phone Out Mixer MIC1 Switch	Phone Out Mixer 设置, 使能 MIC1 通路	0: 关闭; 1: 开启
Phone Out Mixer MIC2 Switch	Phone Out Mixer 设置, 使能 MIC2 通路	0: 关闭; 1: 开启
Phone Out Mixer MIC3 Switch	Phone Out Mixer 设置, 使能 MIC3 通路	0: 关闭; 1: 开启
Phone Out Mixer ROMIX Switch	Phone Out Mixer 设置, 使能 routput	0: 关闭; 1: 开启
Phoneout Speaker Switch	使能 Phoneout Speaker	0: 关闭; 1: 开启
ADC gain volume	ADC 增益	0–7, 表示-4.5–6dB
LINEIN Mixer volume	Linein mixer volume	0–7, 表示-4.5–6dB
LINEIN gain volume	Linein gain volume	0–7, 表示-4.5–6dB
LINEOUT volume	lineout 音量设置	0–31, 表示-43.5–0dB
Left Input Mixer LINEINL Switch	Left Input Mixer 设置，使能 LINEINL	0: 关闭; 1: 开启
Left Input Mixer LINEINLR Switch	Left Input Mixer 设置，使能 LINEINLR	0: 关闭; 1: 开启
Left Input Mixer LOMIX Switch	Left Input Mixer 设置，使能 loutput	0: 关闭; 1: 开启
Left Input Mixer ROMIX Switch	Left Input Mixer 设置，使能 routput	0: 关闭; 1: 开启
Left Input Mixer MIC1 Boost Switch	Left Input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Left Input Mixer MIC2 Boost Switch	Left Input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Left Input Mixer MIC3 Boost Switch	Left Input Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
Left LINEOUT Mux	Left Lineout Mux 设置	0:Left OMixer; 1:LR OMixer
Left Output Mixer DACL Boost Switch Left	Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Left Output Mixer DACR Boost Switch Left	Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Left Output Mixer LINEINL Boost Switch	Left Output Mixer 设置，使能 LINEINL 通路	0: 关闭; 1: 开启
Left Output Mixer LINEINLR Boost Switch	Left Output Mixer 设置，使能 LINEINLR 通路	0: 关闭; 1: 开启
Left Output Mixer MIC1 Boost Switch	Left Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Left Output Mixer MIC2 Boost Switch	Left Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Left Output Mixer MIC3 Boost Switch	Left Output Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
MIC1 boost volume	MIC1 Boost AMP gain	0–7, 0:0dB, 1~7:24–42dB
MIC2 boost volume	MIC2 Boost AMP gain	0–7, 0:0dB, 1~7:24–42dB
MIC2 boost volume	MIC2 Boost AMP gain	0–7, 0:0dB, 1~7:24–42dB
MIC1 gain volume	MIC1 to L/R output mixer gain	0–7, 表示-4.5–6dB
MIC2 gain volume	MIC2 to L/R output mixer gain	0–7, 表示-4.5–6dB
MIC2 gain volume	MIC3 to L/R output mixer gain	0–7, 表示-4.5–6dB
Right Input Mixer LINEINLR Switch	Right Input Mixer 设置，使能 LINEINLR	0: 关闭; 1: 开启
Right Input Mixer LINEINR Switch	Right Input Mixer 设置，使能 LINEINR	0: 关闭; 1: 开启
Right Input Mixer LOMIX Switch	Right Input Mixer 设置，使能 LOMIX 通路	0: 关闭; 1: 开启
Right Input Mixer ROMIX Switch	Right Input Mixer 设置，使能 ROMIX 通路	0: 关闭; 1: 开启
Right Input Mixer MIC1 Boost Switch	Right Input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Input Mixer MIC2 Boost Switch	Right Input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Right Input Mixer MIC3 Boost Switch	Right Input Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
Right LINEOUT Mux	Right Lineout Mux 设置	0:Right OMixer; 1:LR OMixer
Right Output Mixer DACL Boost Switch	Right Output Mixer 设置，使能 DACL 通路	0: 关闭; 1: 开启
Right Output Mixer DACR Boost Switch	Right Output Mixer 设置，使能 DACR 通路	0: 关闭; 1: 开启
Right Output Mixer LINEINR Boost Switch	Right Output Mixer 设置，使能 LINEINR 通路	0: 关闭; 1: 开启
Right Output Mixer LINEINLR Boost Switch	Right Output Mixer 设置，使能 LINEINLR 通路	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
Right Output Mixer MIC1 Boost Switch	Right Output Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Right Output Mixer MIC2 Boost Switch	Right Output Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Right Output Mixer MIC3 Boost Switch	Right Output Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
Xadc Input Mixer LINEINLR Switch	Xadc Input Mixer 设置，使能 LINEINLR 通路	0: 关闭; 1: 开启
Xadc Input Mixer LOMIX Switch	Xadc Input Mixer 设置，使能 LOMIX 通路	0: 关闭; 1: 开启
Xadc Input Mixer ROMIX Switch	Xadc Input Mixer 设置，使能 ROMIX 通路	0: 关闭; 1: 开启
Xadc Input Mixer MIC1 Boost Switch	Xadc Input Mixer 设置，使能 MIC1 通路	0: 关闭; 1: 开启
Xadc Input Mixer MIC2 Boost Switch	Xadc Input Mixer 设置，使能 MIC2 通路	0: 关闭; 1: 开启
Xadc Input Mixer MIC3 Boost Switch	Xadc Input Mixer 设置，使能 MIC3 通路	0: 关闭; 1: 开启
digital volume	数字端音量设置	0–63, 表示-73.08–0dB
phoneout volume	phone volume 设置	0–7, 表示-4.5–6dB

2.15.5 Daudio

硬件特性

- 三路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 3 路 MCLK 输出

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Digital Audio Support
```

2.15.5.2 sys_config 配置

```
[snddaudio0]
snddaudio0_used         = 0
daudio_master           = 4
audio_format            = 1
signal_inversion        = 1

[daudio0]
daudio0_used            = 0
slot_width_select       = 32
pcm_lrck_period         = 128
msb_lsb_first           = 0
sign_extend             = 0
frametype               = 0
mclk_div                = 1
tdm_config              = 1
tx_data_mode            = 0
rx_data_mode            = 0
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge)

版权所有 © 珠海全志科技股份有限公司。保留一切权利

snddaudio 配置	snddaudio 配置说明
signal_inversion	5: SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge) 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)

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
daudio0_used	是否使用 daudio 驱动。0：不使用；1：使用
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

2.15.6 DMIC

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.15.6.2 sys_config 配置

配置如下:

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

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
dmic_used	是否使用 dmic 驱动。0：不使用；1：使用

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
snddmic_used	是否使用 snddmic 驱动。0：不使用；1：使用

2.15.7 SPDIF

硬件特性

- 支持 S/PDIF_OUT 和 S/PDIF_IN
- 支持 mono 和 stereo 模式 (mono 模式下由硬件自动拓展为 stereo)
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 输入支持 44.1KHz,48KHz 采样率
- 输出和输入支持 16bit,24bit 采样精度

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.15.7.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner SPDIF Support
```

2.15.7.2 sys_config 配置

```
[sndspdif]
sndspdif_used = 0
[spdif]
spdif_used = 0
```

spdif 配置，即 platform 驱动的相关配置

spdif 配置	spdif 配置说明 |
spdif_used	是否使用 spdif 驱动。0：不使用；1：使用 |

sndspdif 配置，即 machine 驱动的相关配置

sndspdif 配置	sndspdif 配置说明
sndspdif_used	是否使用 sndspdif 驱动。0：不使用；1：使用

2.15.8 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令。

2.15.8.1 播放

```
通过Speaker播放
amixer -Dhw:audiocodec cset name='Left Output Mixer DACL Switch' 1
amixer -Dhw:audiocodec cset name='Right Output Mixer DACR Switch' 1
amixer -Dhw:audiocodec cset name='Phone Out Mixer LOMIX Switch' 1
amixer -Dhw:audiocodec cset name='Phone Out Mixer ROMIX Switch' 1
amixer -Dhw:audiocodec cset name='Phoneout Speaker Switch' 1
amixer -Dhw:audiocodec cset name='digital volume' 63
amixer -Dhw:audiocodec cset name='phoneout volume' 4
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

2.15.8.2 录音

```
通过MIC1,MIC2录制两通道
amixer -D hw:audiocodec cset name='Left Input Mixer MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='Right Input Mixer MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 boost volume' 4
amixer -D hw:audiocodec cset name='MIC2 boost volume' 4

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.16 R329 音频接口

2.16.1 硬件资源

R329 音频接口丰富，其中 System 域 (A53) 有三路 I2S, 一个 SPDIF 接口; 而 DSP 域有 AudioCodec,DMIC 以及两路 I2S。

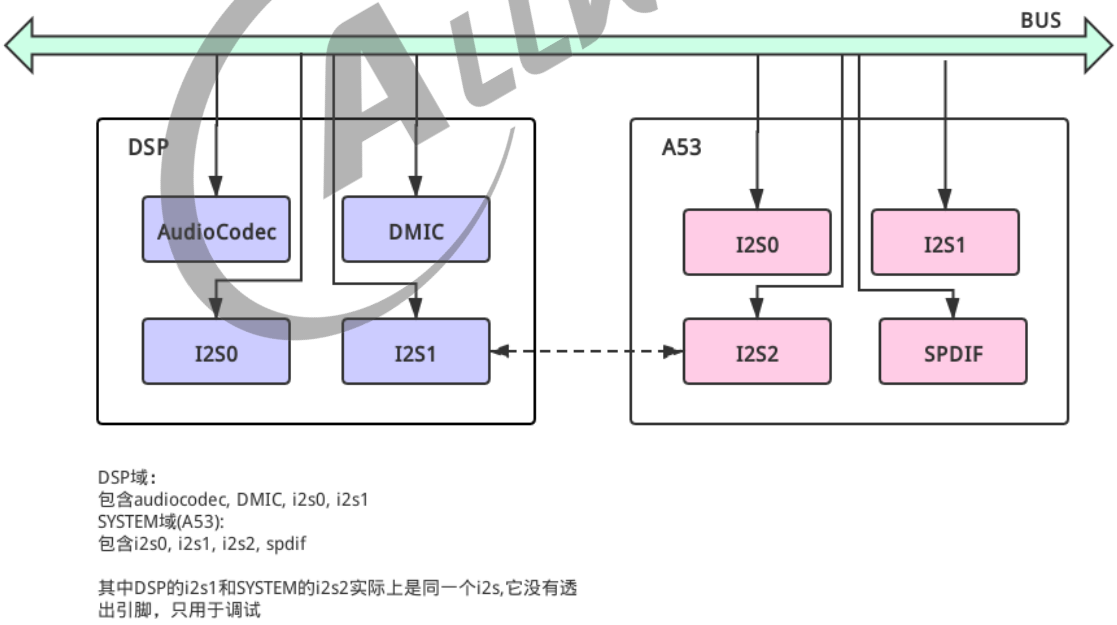


图 2-40: R329 音频硬件框图

其中 DSP 的 I2S1 与 System 域的 I2S2 实际上是同一个 I2S，它对外没有透出 PIN 脚，是一个内联的 I2S，主要用于 CPU 与 DSP 之间的调试。

另外还支持 VAD 作语音唤醒检测 (详细请看R329 VAD 章节), 休眠时 CPU 可以被关闭，而 DSP 可以处于 IDLE 状态等待 VAD 语音能量唤醒，实现低功耗场景。

2.16.2 时钟源

R329 的音频模块的时钟源来自 pll_audio0 或者 pll_audio1，如果需要使用 48K 系列的采样率，则驱动中会选用 pll_audio0 输出 24.576M 的时钟，如果需要使用 44.1K 系列的采样率，则选用 pll_audio1 输出 22.5792M 的时钟。

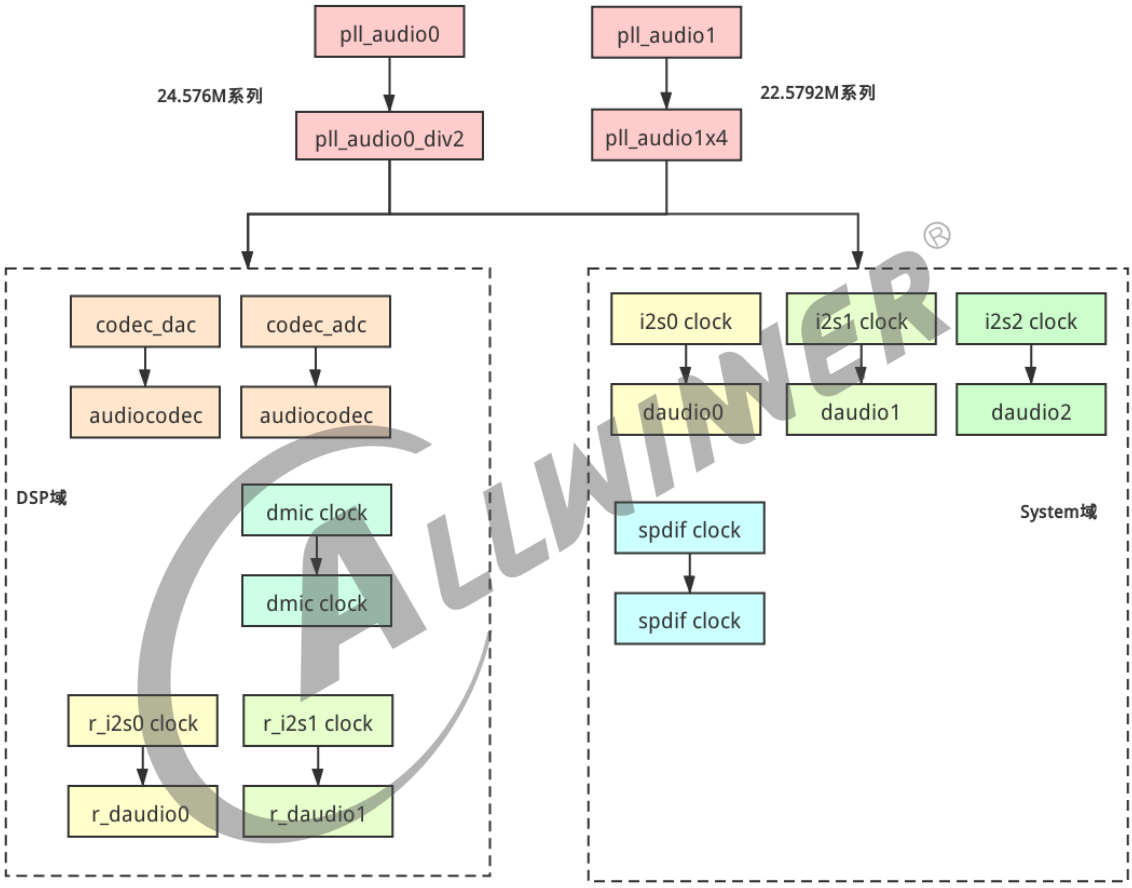


图 2-41: R329 时钟源

2.16.3 代码结构

R329 上分两套 Audio 代码，一套是 Linux-Audio，完全由 CPUX 去控制音频硬件资源，使用的传统 ALSA 驱动；而另一套是 Linux-DSP-Audio，DSP 域上的音频硬件资源 (AudioCodec,DMIC,I2S) 由 DSP 去控制，代码上使用 AW-RPAF 驱动，这是基于 ALSA 驱动开发的一套框架，如下图所示：

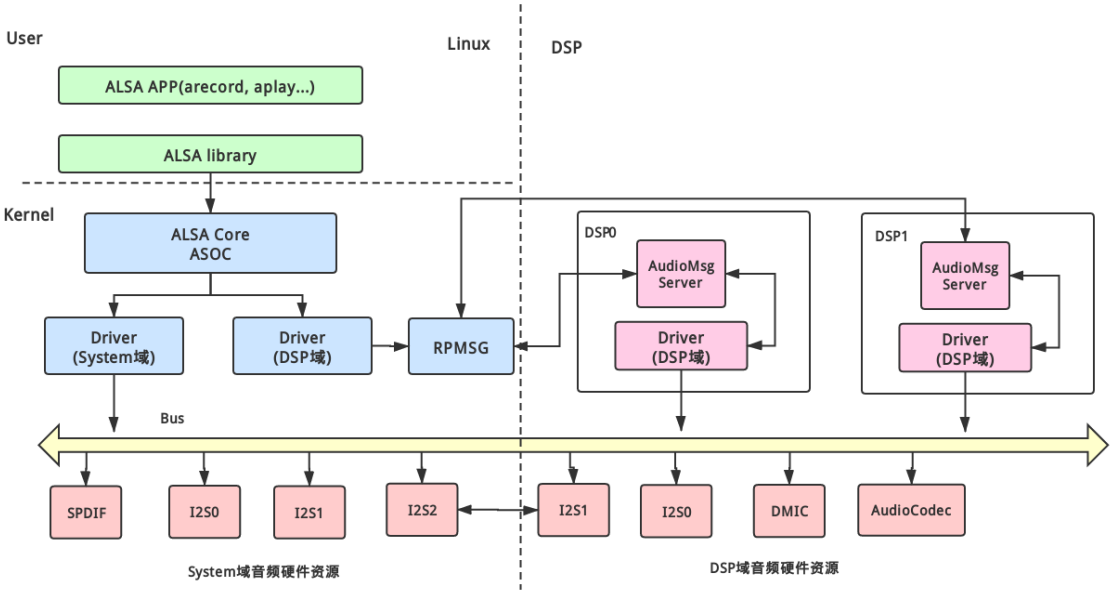


图 2-42: R329 软件驱动框架

对于应用层，使用的同样是 alsa-lib，客户应用无需进行代码调整即可正常进行录音、播放。而内核驱动中操作音频硬件资源的地方都会被替换成与 DSP 的通信接口，由 DSP 完成底层硬件的操作。

代码上默认选用 Linux-DSP-Audio 这套代码，而 Linux-Audio 这套代码只作备用，后续没有特殊描述的都是基于 Linux-DSP-Audio 下进行的开发说明。

2.16.3.1 Linux-Audio 下的代码结构

```
linux-4.9/sound/soc/sunxi/sun50iw11/
├── spdif-utils.c                // spdif codec驱动
├── sunxi-codec.c                // AudioCodec驱动
├── sunxi-codec.h
├── sunxi-sndcodec.c             // AudioCodec machine驱动
├── sunxi-cpudai.c               // AudioCodec platform驱动
├── sunxi-daudio.c               // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-dmic.c                 // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-pcm.c                  // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi-snddaudio.c            // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-snddmic.c              // dmic machine驱动
├── sunxi-snddmic.h
├── sunxi-sndspdif.c             // spdif machine驱动
├── sunxi-spdif.c                // spdif platform驱动
└── sunxi-spdif.h

linux-4.9/sound/soc/codecs/dmic.c   // dmic codec驱动
```

```
linux-4.9/sound/soc/soc-utils.c     // daudio codec驱动
```

对应的内核配置如下所示，注意不要选中 “ALSA for Sunxi HiFi”：

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     < > ALSA for Sunxi HiFi  --->
```

2.16.3.2 Linux-DSP-Audio 下的代码结构

```
linux-4.9/sound/sunxi-rpaf
├── component
│   ├── component-core.c                    // RPAF算法组件核心文件，生成节点用于与DSP的算法组件交
    互
│   └── component-driver.c                  // RPAF算法组件驱动
├── core                                    // 原生ALSA驱动的核心代码目录，这部分没有作修改
│   ├── pcm_lib.c
│   ...
└── soc
    ├── codecs                              // ASOC框架下的codec驱动,这里主要是外挂codec的驱动
    │   ├──dmic.c                           // dmic codec驱动
    │   ├──ac107.c                          // AC107驱动，含有2个ADC
    │   └──ac108.c                          // AC108驱动，含有4个ADC
    └── sunxi
        ├── spdif-utils.c                   // spdif codec驱动
        ├── sunxi-spdif.c                   // spdif platform 驱动
        ├── sunxi-sndspdif.c                // spdif machine 驱动
        ├── sunxi-pcm.c                     // 通用文件，提供注册platform驱动的接口及相关函数集
        ├── sunxi-daudio.c                  // System域的I2S, daudio platform驱动
        ├── sunxi-snddaudio.c               // System域的I2S, daudio machine驱动
        └── hifi-dsp
            ├── rpmsg_hifi.c                // RPAF框架下的RPMSG驱动，提供了与DSP通信的接口
            ├── sun50iw11-cpudai.c          // HIFI驱动，AudioCodec platform驱动
            ├── sun50iw11-codec.c           // HIFI驱动，AudioCodec codec驱动
            ├── sun50iw11-sndcodec.c        // HIFI驱动，AudioCodec machine驱动
            ├── sunxi-daudio.c              // HIFI驱动，daudio platform驱动
            ├── sunxi-snddaudio.c           // HIFI驱动，daudio machine驱动
            ├── sunxi-hifi-pcm.c            // HIFI驱动，提供注册platform驱动的接口及相关函数集
            ├── sunxi-dmic.c                // HIFI驱动，dmic platform驱动
            └── sunxi-snddmic.c             // HIFI驱动，dmic platform驱动
```

对应的内核配置如下所示，注意需要选中 “ALSA for Sunxi HiFi”：

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi  --->
```

2.16.4 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 六路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出：支持 SPKL 与 SPKR 的差分输出
- 六路模拟输入：MIC1,MIC2,MIC3,MIC4,MIC5,MIC6
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 5 段 DRC
- DAC FIFO 长度 128*24bits, ADC FIFO 长度 256*24bits

2.16.4.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
   <*> ALSA for Sunxi HiFi --->
      <*> ALSA for SoC Sunxi HiFi Support --->
         <*> Allwinner Audio Codec HiFi Support
```

2.16.4.2 DTS 配置

```
codec:codec@0x07032000 {
    pa_level = <0x1>;
    digital_vol = <0x00>;
    gpio-pa-power = <&r_pio PN 22 1 1 1 1>;
    gpio-boost-en = <&r_pio PN 20 1 1 1 1>;
    gpio-spk = <&r_pio PM 8 1 1 1 1>;
    /* dsp config */
    playback_cma = <32>;
    capture_cma = <64>;
    pa_msleep_time = <0x64>;
    spk_vol = <0x14>;
    mic1gain = <0x13>;
    mic2gain = <0x13>;
    mic3gain = <0x13>;
    mic4gain = <0x13>;
    mic5gain = <0x0>;
    mic6gain = <0x0>;
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    mic_num = <0x6>;
    adcdrc_cfg = <0x00>;
    adchpf_cfg = <0x01>;
    dacdrc_cfg = <0x00>;
    dachpf_cfg = <0x00>;
    rx_sync_en = <0x0>;
    status = "okay";
};

cpudai:cpudai-controller@0x07032000 {
    dsp_card = <0x0>;
    status = "okay";
};

sndcodec:sound@0 {
    status = "okay";
};
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
status	是否使用 sndcodec 驱动。disabled：不使用；okay：使用

cpudai 配置，即 platform 驱动的相关配置

cpudai 配置	cpudai 配置说明
status	是否使用 cpudai 驱动。disabled：不使用；okay：使用
dsp_card	DSP 中的声卡号，0: AudioCodec; 1: DMIC; 2: r_i2s0

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
digital_vol	初始化 digital volume，可设定范围 0~0x3f 表示 0~-73.08dB, -1.16dB/step
gpio-pa-power	PA Power 使能引脚
gpio-boost-en	PA Boost 使能引脚
gpio-spk	PA 使能引脚
playback_cma	配置 playback 的 dma buffer 大小，单位 KB
capture_cma	配置 capture 的 dma buffer 大小，单位 KB
pa_msleep_time	操作 PA 之后的延时时间 (用来避免 pop 音)
spk_vol	speaker volume，可设定范围 0~0x1f 表示-43.5dB~0dB, 1.5dB/step
mic1gain	mic1 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
mic2gain	mic2 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
mic3gain	mic3 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
mic4gain	mic4 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB 如果作为 aec 回路，则需要设置为 0dB
mic5gain	mic5 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB 如果作为 aec 回路，则需要设置为 0dB
mic6gain	mic6 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB 如果作为 aec 回路，则需要设置为 0dB
mic_num	MIC 个数,R329 含有 6 个 MIC, 所以固定为 6
adcdrc_cfg	是否使用 adcdrc. 0: 不适用；1：使用
adchpf_cfg	是否使用 adchpf. 0: 不适用；1：使用
dacdrc_cfg	是否使用 dacdrc. 0: 不适用；1：使用
dachpf_cfg	是否使用 dachpf. 0: 不适用；1：使用
rx_sync_en	是否使用 RX_SYNC 功能 (用于双声卡同步). 0: 关闭; 1: 使能
status	是否使用 codec 驱动。disabled：不使用；okay：使用

2.16.4.3 AudioCodec 控件

R329 所有控件如下表：

控件名称	功能	数值
Speaker volume	Speaker 音量设置	0~31, 表示-43.5~0dB
MIC1 gain volume	MIC1 Boost AMP gain	0~31, 0:0dB, 1~3:6dB, 4~31:9~36dB
MIC2 gain volume	MIC2 Boost AMP gain	0~31, 0:0dB, 1~3:6dB, 4~31:9~36dB
MIC3 gain volume	MIC3 Boost AMP gain	0~31, 0:0dB, 1~3:6dB, 4~31:9~36dB
MIC4 gain volume	MIC4 Boost AMP gain	0~31, 0:0dB, 1~3:6dB, 4~31:9~36dB
MIC5 gain volume	MIC5 Boost AMP gain	0~31, 0:0dB, 1~3:6dB, 4~31:9~36dB
MIC6 gain volume	MIC6 Boost AMP gain	0~31, 0:0dB, 1~3:6dB, 4~31:9~36dB
codec vad support	使能 VAD 功能	0: 关闭; 1: 开启

版权所有 © 珠海全志科技股份有限公司。保留一切权利

注意事项：

默认 DSP 配置 AudioCodec 的 dma buffer 比较小 (因为 SRAM 比较有限), 所以如果应用上不能设置较大的 period_size, 以及 buffer_size. 默认/etc/asound.conf 中配置了 default 声卡的 period_size 为 1024, 所以直接 aplay test.wav 就可以正常播放了；如果使用 hw 方法进行播放，则需要指定 period_size 以及 buffer_size，如下：

```
aplay -Dhw:audiocodec /tmp/test.wav --period-size 1024 --buffer-size 4096
```

2.16.5 Daudio

硬件特性

- System 域下三路 I2S/PCM
  - I2S0 含有 4 线 DOUT
  - I2S1 含有 2 线 DIN/OUT
  - I2S2 不出 pin, 与 DSP 域下的 R_I2S1 片内互联
- DSP 域下两路 I2S/PCM
  - R_I2S02 线 DIN/OUT
  - R_I2S1 不出 pin，与 System 域下的 I2S2 片内互联
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 I2S0,I2S1, R_I2S0 均支持 MCLK 输出

2.16.5.1 内核配置

```
System域的I2S:

  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi --->
        <*> ALSA for SoC Sunxi HiFi Support --->
           <*> ALSA for SoC Sunxi Audio Support --->
              <*> Allwinner Digital Audio Support
DSP域的I2S:
```

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi --->
        <*> ALSA for SoC Sunxi HiFi Support --->
           <*> Allwinner Digital HiFi Support
```

2.16.5.2 DTS 配置

System 域下的 I2S 配置:

```
daudio0:daudio@0x02004000 {
    mclk_div = <0x00>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    dsp_card = <0x1>;
    status = "disabled";
};
snddaudio0:sound@1 {
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "disabled";
};
```

注意：audio_format, daudio_master, signal_inversion 的配置是放在 snddaudio 中。

DSP 域下的 I2S 配置：

```
r_daudio0:daudio@0x07033000 {
    compatible = "allwinner,sunxi-hifi-daudio";
    mclk_div = <0x01>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    capture_cma = <64>;
    playback_cma = <32>;
    tx_num = <1>;
    tx_chmap1 = <0x76543210>;
    tx_chmap0 = <0xFEDCBA98>;
    rx_num = <4>;
    rx_chmap3 = <0x03020100>;
    rx_chmap2 = <0x07060504>;
    rx_chmap1 = <0x0B0A0908>;
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    rx_chmap0 = <0x0F0E0D0C>;
    rx_sync_en = <0x0>;
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};
```

注意：r_i2s 的配置均放在 r_daudio 中，包括 audio_format, daudio_master, signal_inversion。

配置说明:

I2S 配置	I2S 配置说明
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode

版权所有 © 珠海全志科技股份有限公司。保留一切权利

I2S 配置	I2S 配置说明
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
playback_cma	配置 playback 的 dma buffer 大小，单位 KB
capture_cma	配置 capture 的 dma buffer 大小，单位 KB
rx_sync_en	是否使用 RX_SYNC 功能 (用于双声卡同步). 0: 关闭; 1: 使能
tx_num	默认配置的 tx 个数 (I2S-DOUT 个数)
txchmapX	设置通道映射，txchmap0 设置 8 15 通道，txchmap1 设置 0 7 通道
rx_num	默认配置的 rx_chmap 个数，默认为 4
rxchmapX	设置通道映射及 DIN 选择，每个 rxchmapX 设置 4 通道, 如 rx_chmap3=0x03020100, 表示使用 I2S-DIN0, 并且通道映射为 0ch->0ch, 1ch->1ch, 2ch->2ch, 3ch->3ch

注意事项:

- daudio machine 驱动的配置 (snddaudio), 一般来说还需要配置 codec name 以及 codec dai name

  1. 例如 r_daudio0 使用了 AC107 作为外挂 codec:

```
snddaudio3:sound@4 {
    sunxi,snddaudio-codec = "ac107.2-0036";
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
};

r_daudio0:daudio@0x07033000 {
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
};

注意名称需要与codec驱动中配置的名称一致，如ac107驱动，路径:
linux-4.9/sound/sunxi-rpaf/soc/codecs/ac107.c
代码中snd_soc_register_codec注册codec驱动，其中codec device name为ac107.2-0036,codec
dai name为ac107-pcm0
```

  2. 例如 daudio1 与 bluetooth 模组相连 (没有实际的 codec 驱动)，那么这时候 codec name, codec dai name 需要配置为 dummy codec, 可以如下配置:

```
snddaudio1:sound@2 {
    audio_format = <0x05>;
    daudio_master = <0x01>;
    signal_inversion = <0x02>;
};

或者
snddaudio1:sound@2 {
    audio_format = <0x05>;
```

```
        daudio_master = <0x01>;
        signal_inversion = <0x02>;
        sunxi,snddaudio-codec;
        sunxi,snddaudio-codec-dai;
    };
```

2.16.6 Dmic

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.16.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
   <*> ALSA for Sunxi HiFi --->
      <*> ALSA for SoC Sunxi HiFi Support --->
         <*> Allwinner DMIC Support
```

2.16.6.2 DTS 配置

配置如下:

```
r_dmic:dmic-controller@0x07031000 {
    capture_cma = <64>;
    data_vol = <0xB0>;
    rx_sync_en = <0x00>;
    status = "disabled";
};

snddmic:sound@7 {
    status = "disabled";
};
```

r_dmic 配置	r_dmic 配置说明
status	是否使用 dmic 驱动。disabled：不使用；okay：使用
capture_cma	配置 capture 的 dma buffer 大小，单位 KB
data_vol	DATA0~DATA3 的数字增益, 默认配置 0xB0 即 12dB
rx_sync_en	是否使用 RX_SYNC 功能 (用于双声卡同步). 0: 关闭; 1: 使能
rx_chmap	通道映射, 默认配置 0x76543210 表示按照默认通道顺序

版权所有 © 珠海全志科技股份有限公司。保留一切权利

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
status	是否使用 snddmic 驱动。disabled：不使用；okay：使用

2.16.7 SPDIF

硬件特性

- 支持 S/PDIF_OUT 和 S/PDIF_IN
- 支持 mono 和 stereo 模式 (mono 模式下由硬件自动拓展为 stereo)
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 输入支持 44.1KHz,48KHz 采样率
- 输出和输入支持 16bit,24bit 采样精度

2.16.7.1 内核配置

```
  Device Drivers  --->

  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi --->
        <*> ALSA for SoC Sunxi Audio Support --->
          <*> Allwinner SPDIF Support
          <*> Allwinner SPDIF RX IEC61937 Support
```

2.16.7.2 DTS 配置

```
spdif:spdif-controller@0x02002c00 {
    clk_parent = <0x3>;
    status = "disabled";
};

sndspdif:sound@6 {
    status = "disabled";
};
```

spdif 配置，即 platform 驱动的相关配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

spdif 配置	spdif 配置说明
status	是否使用 spdif 驱动。disabled：不使用；okay：使用

sndspdif 配置，即 machine 驱动的相关配置

sndspdif 配置	sndspdif 配置说明
status	是否使用 sndspdif 驱动。disabled：不使用；okay：使用

2.16.8 VAD

硬件特性

- 支持 AudioCodec, I2S，DMIC 音频接口，时分复用，固定 16bit
- 支持 16KHz,48KHz 采样率
- 支持基于能量识别的语音检测模块 LPSD

2.16.8.1 内核配置

无需内核配置。

2.16.8.2 使用说明

默认 SDK 可以直接使用 VAD 功能。

使用流程:

```
开机后通过amixer命令设置:
amixer -Dhw:audiocodec cset name='codec vad support' 1

然后应用正常进行录音即可，如果需要进入休眠，有下面几点需要实现的:

 1. 暂定录音、播放。snd_pcm_drop将playback,capture均暂停
 2. 设置wakeup_count。更新当前唤醒次数
 3. 进入休眠。写mem到/sys/power/state即可
```

tina SDK 中有一个能量唤醒 demo 可供参考

make menuconfig 选中 mad-demo 软件包

```
Allwinner  --->
  <*> mad-demo
```

R329 上执行 vad-demo, 它会每录制 5s 都进入一次休眠，等待语音能量唤醒。

如果想查看录音数据，可以执行 vad-demo dump, 录音文件保存在/mnt/UDISK/vad-test.wav

注意：VAD 目前仅支持 1,2,4 通道录音。

2.16.8.3 能量唤醒阈值参数

能量唤醒模块 lpsd，识别能量主要有两个方向，瞬时能量和累计能量（前者比如是关门声，后者比如是不断说话）能量检测参数配置在 DTS 中配置：

```
mad:mad@0x07097000{
    lpsd_th = <0x4b0>;
    lpsd_rrun = <0x4d>;
    lpsd_rstop = <0x80>;
};
```

lpsd_rrun 和 lpsd_rstop 的推荐值：

lpsd_rrun	lpsd_rstop
77	88
77	108
77	128
77	148

1. 瞬时能量检测参数，主要是 lpsd_rrun 和 lpsd_rstop

- 一般我们只对 stop 值进行修改
- 如果录音数据经常缺少唤醒词的第一个字，则可以尝试降低 stop 值，可以有效提高唤醒词数据的完整性。但同时会提高误唤醒率，环境噪音也会很容易触发能量检测，唤醒系统。
- 如果想要降低误唤醒率 (环境噪音造成唤醒)，则可以尝试提高 stop 值。同样的，这会导致一些唤醒词录音数据不完整，例如一些音量较低，音调较低的语料。
- 唤醒词识别率以及误唤醒率无法同时兼得，客户需要根据实际需求、场景，权衡配置参数

2. 累积能量检测参数，主要是 lpsd_th

- 我们建议使用默认值 1200。建议修改范围 50~1200

注意事项:

1. 应用操作上的一些要求，具体请查看《2.16.8.2 使用说明》章节
2. 如果读取 wakeup_count 时一直阻塞，说明当前仍有 wake_lock 处于激活状态，例如 usb 线连接着 PC，usb 驱动会保持一个 wake_lock，不让系统进入休眠，所以需要拔掉 usb 或者连接到 usb 适配器上，或者改动代码，去掉 usb 驱动中 wake_lock 的使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.16.9 标案音频 ALSA 配置

该章节主要介绍在标案上的 alsa 配置 (/etc/asound.conf), 以 SDK 中 r329-evb5 方案为例：

```
配置文件在SDK路径: package/allwinner/alsa-conf-aw/files/r329/r329-evb5
如果客户增加新方案，请注意自行添加自己的asound.conf
```

```
default:
    1)playback, 使用AudioCodec的DAC,默认输出48K,两声道,S16_LE
     2)capture, 使用AudioCodec的ADC,默认录制16K,6通道,S16_LE

通过Speaker播放:
aplay /mnt/UDISK/1KHz_0dB_16000.wav
录制六通道:
arecord -f S16_LE -r 16000 -c 6 /tmp/test.wav
```

```
CaptureMic:
     用于录制麦克风数据，标案上只有3个麦克风，所以这里只取3通道数据。
     可以使用该配置的场景,如: 蓝牙通话, 声波联网

录制单声道(实际会将MIC1, MIC2, MIC3数据合并):
arecord -DCaptureMic -f S16_LE -r 16000 -c 1 /tmp/test.wav
```

```
其他双声卡配置:
CaptureDmicAec:
     用于DMIC与AudioCodec的双声卡，组成6+2方案(6DMIC, + 2内部ADC)

DSPI2S6Mic2Ref:
     用于AC108与AudioCodec的双声卡，组成6+2方案

DSPI2S6Mic1Ref:
     用于AC107与AudioCodec的双声卡，组成6+1方案
```

注意事项：

- 使用双声卡时，需要注意配置 DTS 中两个声卡的 rx_sync_en，如 r_i2s0 与 audiocodec 的 rx_sync_en
- 注意配置的 period_size, buffer_size 不能太大 (DSP 中 SRAM 有限), 建议分别配置 1024, 6144 即可

2.16.10 DSP 音频相关

DSP 运行了 FreeRTOS 系统，实际音频相关的硬件均由 DSP 去控制，而 Linux 只是与 DSP 进行通信，不会操作实际音频硬件模块。

DSP 中可独立的进行播歌、录音等操作 (测试用)，下面介绍这些工具的操作方法 (一般客户不需要改动或操作 DSP，这里的命令也只用作调试)。

2.16.10.1 默认支持的命令

DSP FreeRTOS 系统默认支持命令行控制，音频相关的命令如下所示：

命令	命令描述
aplay	作用与 Linux 的 aplay 一样，用于播歌测试
arecord	作用与 Linux 的 arecord 一样，用于录音测试
amxier	作用与 Linux 的 amixer 一样，用于设置控件信息

1. aplay 命令

   对于一般开发者，建议在 Linux 端使用 aplay 工具进行播歌测试。

   由于 DSP FreeRTOS 中没有文件系统，所以无法 aplay 指定 wav 文件的方式去进行播歌测试。但可以通过生成正弦波去进行播放功能的测试。

   命令参数说明如下：

命令	命令描述
-D	指定声卡，如 hw:audiocodec(默认), hw:snddaudio1
-c	通道数
-r	采样率
-f	采样精度
-p	period size
-b	buffer size
-t	播放时间
-s	播放正弦波

   举例：

```
播放48K,16bit,2通道的正弦波5s:
aplay  -f 16 -r 48000 -c 2 -t 5 -s

(注意，DSP执行aplay命令时，Linux端不能同时进行播歌操作)
```

2. arecord 命令

   对于一般开发者，建议在 Linux 端使用 arecord 工具进行录音测试。

   由于 DSP FreeRTOS 中没有文件系统，所以无法 arecord 将录音文件保存起来，但可以将数据发送到 i2s,

   举例：

```
arecord -D hw:audiocodec -c 4 -r 16000 -f 16 -p 1024 -b 4096 -t 20 -H hw:snddaudio1
从audiocodec录音并同时将数据送给snddaudio1, 运行20秒
同时,可以在linux端用arecord 从snddaudio2上录音(实际上录制的是dsp上的snddaudio1给过来的数据)

(注意，DSP执行arecord命令时，Linux端不能同时对该声卡进行录音操作)
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3. amixer 命令

用于设定指定声卡的控件信息

命令参数说明如下：

命令	命令描述
-l	列出当前所有的声卡
-c	声卡号
controls	列出指定声卡的所有控件的具体信息
cget	获取指定控件的信息
cset	设定指定控件的值

举例：

```
列出当前所有的声卡:
amixer -l
======== Sound Card list ========
card_num            card_name
       0           audiocodec
       1           snddaudio1

列出声卡0的所有控件信息:
amixer -c 0 controls

获取声卡0，numid=13的信息:
amixer cget numid=13

设定声卡0，numid=13的值为20:
amixer cset numid=13 20
```

2.16.11 双声卡配置

R329 内部含有 6 个 ADC，如果有其他语音方案要求更多的 ADC，那么就需要外挂 ADC 或者使用 DMIC, 与内部 ADC 组成双声卡。下面介绍下 R329 下如何配置使用双声卡。

2.16.11.1 外挂 AC107

AC107 含有两个 ADC，常用组合方式：

- R329 + 单片 AC107，实现 7+1 方案 (7MIC=R329 的 5 个 MIC + AC107 的两个 MIC, 1REF=R329 的最后一个 MIC)
- R329 + 三片 AC107，实现 6+2 方案 (6MIC= 三片 AC107 共 6 个 MIC, 2REF=R329 的两个 MIC)

配置方法:

1. 内核配置

需要选中 i2s 驱动

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi --->
        <*> ALSA for SoC Sunxi Audio Support --->
           CODEC drivers --->
              <*> Sunxi AC107 Codec
        <*> ALSA for SoC Sunxi HiFi Support --->
           <*> Allwinner Digital HiFi Support
```

2.DTS 配置

配置 I2S, 其中 r_daudio0 里面的 audio_format, daudio_master, signal_inversion 是会传递给 DSP 中设置到 R329 的 I2S 硬件上, 而 snddaudio3 里面的配置会传递给 AC107 驱动中，所以注意要同时配置上，并且格式要一致。

I2C 配置上 AC107 设备，如挂在 twi2(r_twi0) 上，配置如下：

```
注意pcm_lrck_period = <0x20>;
r_daudio0:daudio@0x07033000 {
    mclk_div = <0x02>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x20>;
    slot_width_select = <0x20>;
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};

snddaudio3:sound@4 {
    sunxi,snddaudio-codec = "ac107.2-0036";
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};

&twi2{
    ac107@36 {
        compatible = "Allwinnertech,ac107_0";
        reg = <0x36>;
        status = "okay";
    };
};
```

如果是三片 AC107 则按如下配置：

```
注意pcm_lrck_period = <0x60>;
r_daudio0:daudio@0x07033000 {
    mclk_div = <0x02>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x60>;
    slot_width_select = <0x20>;
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};

snddaudio3:sound@4 {
    sunxi,snddaudio-codec = "ac107.2-0036";
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};
&twi2{
    ac107@36 {
        compatible = "Allwinnertech,ac107_0";
        reg = <0x36>;
        status = "okay";
    };
    ac107@37 {
        compatible = "Allwinnertech,ac107_1";
        reg = <0x37>;
        status = "okay";
    };
    ac107@38 {
        compatible = "Allwinnertech,ac107_2";
        reg = <0x38>;
        status = "okay";
    };
};
```

注意！AC107 代码默认配置的是单片 AC107, 如果需要支持三片，请修改代码, 将 sound/sunxi-rpaf/soc/codecs/ac107.c 的宏 AC107_CHIP_NUMS 修改为 3

另外，双声卡存在同步问题，录音数据有可能存在较大的相位差，请使能 AudioCodec 以及 I2S 的 rx_sync_en, 配置如下:

```
r_daudio0:daudio@0x07033000 {
    rx_sync_en = <0x1>;
};
codec:codec@0x07032000 {
    rx_sync_en = <0x1>;
};
注意，使能rx_sync_en之后，不能单独对AudioCodec或者AC107进行录音了，必须两者同时进行录音。
```

最终测试结果两个声卡的录音数据相位差为 6 个采样点 (该相位差是由于两种类型 ADC 设计上的

差异引起的).

3. 测试

单片 AC107 组成 7+1:

```
设置各个MIC增益均为24dB,REF
amixer -Dhw:audiocodec cset name='MIC1 gain volume' 0x13
amixer -Dhw:audiocodec cset name='MIC2 gain volume' 0x13
amixer -Dhw:audiocodec cset name='MIC3 gain volume' 0x13
amixer -Dhw:audiocodec cset name='MIC4 gain volume' 0x13
amixer -Dhw:audiocodec cset name='MIC5 gain volume' 0x13
amixer -Dhw:audiocodec cset name='MIC6 gain volume' 0
amixer -Dhw:sndac10720036 cset name='Channel 1 PGA Gain' 25
amixer -Dhw:sndac10720036 cset name='Channel 2 PGA Gain' 25

(没有使能rx_sync_en的情况下)可以先尝试下只录制ac107的情况:
arecord -Dhw:sndac10720036 -f S16_LE -r 16000 -c 2 --period-size 1024 --buffer-size 4096 /
    tmp/test.wav

录制8通道:
arecord -DCapture7Mic1Ref -f S16_LE -r 16000 -c 8 --period-size 1024 --buffer-size 4096 /
    tmp/test.wav

录制得到的7通道数据排列:
ch0:R329 MIC1
ch1:R329 MIC2
ch2:R329 MIC3
ch3:R329 MIC4
ch4:R329 MIC5
ch5:AC107 MIC1
ch6:AC107 MIC2
ch7:R329 MIC6

其中R329 MIC6作为REF参考回路
```

三片 AC107 组成 6+2:

```
设置各个MIC增益均为24dB,REF
amixer -Dhw:sndac10720036 cset name='Channel 1 PGA Gain' 25
amixer -Dhw:sndac10720036 cset name='Channel 2 PGA Gain' 25
amixer -Dhw:sndac10720036 cset name='Channel 3 PGA Gain' 25
amixer -Dhw:sndac10720036 cset name='Channel 4 PGA Gain' 25
amixer -Dhw:sndac10720036 cset name='Channel 5 PGA Gain' 25
amixer -Dhw:sndac10720036 cset name='Channel 6 PGA Gain' 25
amixer -Dhw:audiocodec cset name='MIC4 gain volume' 0
amixer -Dhw:audiocodec cset name='MIC5 gain volume' 0

(没有使能rx_sync_en的情况下)可以先尝试下只录制ac107的情况:
arecord -Dhw:sndac10720036 -f S16_LE -r 16000 -c 6 --period-size 1024 --buffer-size 4096 /
    tmp/test.wav

录制8通道:
arecord -DCapture6Mic2Ref -f S16_LE -r 16000 -c 8 --period-size 1024 --buffer-size 4096 /
    tmp/test.wav

录制得到的8通道数据排列:
ch0:AC107 MIC1
ch1:AC107 MIC2
```

```
ch2:AC107 MIC3
ch3:AC107 MIC4
ch4:AC107 MIC5
ch5:AC107 MIC6
ch6:R329 MIC4
ch7:R329 MIC5

其中R329 MIC4 MIC5作为REF参考回路
```

2.16.11.2 外挂 AC108

通过外挂两片 AC108 可实现 6+2 方案：6MIC=AC108 的 6 个 MIC, 2REF=R329 的两个 MIC

配置方法:

1. 内核配置

需要选中 i2s 驱动

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi --->
        <*> ALSA for SoC Sunxi Audio Support --->
           CODEC drivers --->
              <*> Sunxi AC108 Codec
        <*> ALSA for SoC Sunxi HiFi Support --->
           <*> Allwinner Digital HiFi Support
```

2.DTS 配置

配置 I2S, 其中 r_daudio0 里面的 audio_format, daudio_master, signal_inversion 是会传递给 DSP 中设置到 R329 的 I2S 硬件上, 而 snddaudio3 里面的配置会传递给 AC108 驱动中，所以注意要同时配置上，并且格式要一致。

I2C 配置上 AC108 设备，如挂在 twi2(r_twi0) 上，配置如下：

两片 AC108 则按如下配置：

```
注意pcm_lrck_period = <0x80>;
r_daudio0:daudio@0x07033000 {
    mclk_div = <0x01>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    audio_format = <0x01>;
    daudio_master = <0x04>;
```

```
    signal_inversion = <0x01>;
    status = "okay";
};

snddaudio3:sound@4 {
    sunxi,snddaudio-codec = "ac108.2-003b";
    sunxi,snddaudio-codec-dai = "ac108-pcm0";
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};
&twi2{
    ac108@3B {
        compatible = "Allwinner,MicArray_0";
        device_type = "MicArray_0";
        reg = <0x3B>;
        regulator_used = <0x0>;
        power_voltage = <3300000>;
        regulator_name = "vcc-3v3";
        power_gpio_used = <0x0>;
        power-gpio = <&pio PH 7 1 1 1 1>;
        reset_gpio_used = <0x0>;
        reset-gpio = <&pio PH 6 1 1 1 1>;
        twi_bus = <0x1>;
        pga_gain = <0x1F>;
        slot_width = <0x20>;
        lrck_period = <0x80>;
        ref_pga_used = <0x1>;
        ref_pga_gain = <0x10>;
        ref_channel = <0x3>;
        debug_mode = <0x0>;
    };
    ac108@35 {
        compatible = "Allwinner,MicArray_1";
        reg = <0x35>;
        device_type = "MicArray_1";
        regulator_used = <0x0>;
        power_voltage = <3300000>;
        regulator_name = "vcc-3v3";
        power_gpio_used = <0x0>;
        power-gpio = <&pio PH 5 1 1 1 1>;
        reset_gpio_used = <0x0>;
        reset-gpio = <&pio PH 4 1 1 1 1>;
        twi_bus = <0x1>;
        pga_gain = <0x1E>;
        slot_width = <0x20>;
        lrck_period = <0x80>;
        ref_pga_used = <0x0>;
        ref_pga_gain = <0x12>;
        ref_channel = <0x0>;
        debug_mode = <0x0>;
    };
};
```

另外，双声卡存在同步问题，录音数据有可能存在较大的相位差，请使能 AudioCodec 以及 I2S 的 rx_sync_en, 配置如下:

```
r_daudio0:daudio@0x07033000 {
    rx_sync_en = <0x1>;
};
codec:codec@0x07032000 {
    rx_sync_en = <0x1>;
};
注意，使能rx_sync_en之后，不能单独对AudioCodec或者AC108进行录音了，必须两者同时进行录音。
```

3. 测试

两片 AC108 组成 6+2:

```
录制8通道:
arecord -DDSPI2S6Mic2Ref -f S16_LE -r 16000 -c 8 --period-size 1024 --buffer-size 4096 /tmp
    /test.wav
```

2.16.11.3 使用 DMIC

使用 6 个数字麦可搭配组成 6+2 方案：6DMIC, 2REF=R329 的两个 MIC

配置方法:

1. 内核配置

需要选中 i2s 驱动

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
     <*> ALSA for Sunxi HiFi --->
        <*> ALSA for SoC Sunxi HiFi Support --->
           <*> Allwinner DMIC Support
```

2.DTS 配置

```
data_vol为DMIC的数字增益，0xB0=0.75*(0xB0-0xA0)=12dB
r_dmic:dmic-controller@0x07031000 {
    data_vol = <0xB0>;
    status = "okay";
};
snddmic:sound@7 {
    status = "okay";
};
```

另外，双声卡存在同步问题，录音数据有可能存在较大的相位差，请使能 AudioCodec 以及 DMIC 的 rx_sync_en, 配置如下:

```
r_dmic:dmic-controller@0x07031000 {
    rx_sync_en = <0x1>;
};
codec:codec@0x07032000 {
    rx_sync_en = <0x1>;
```

```
};
注意，使能rx_sync_en之后，不能单独对AudioCodec或者DMIC进行录音了，必须两者同时进行录音。
```

3. 测试

6 个 DMIC 组成 6+2:

```
录制8通道:

arecord -DCaptureDmicAec -f S16_LE -r 16000 -c 8 --period-size 1024 --buffer-size 4096 /tmp
    /test.wav
```

2.17 MR813 音频接口

MR813 包含 7 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Daudio2,Daudio3,Dmic,Spdif。

2.17.1 时钟源

MR813 中，7 个音频模块的时钟源来自 pll_audio 以及 pll_com_div5。

pll_audio 输出 24.576M, 而 pll_com_div5 输出 90.3168M(再由具体模块，如 dac 的 4 分频得到 22.5792M) 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

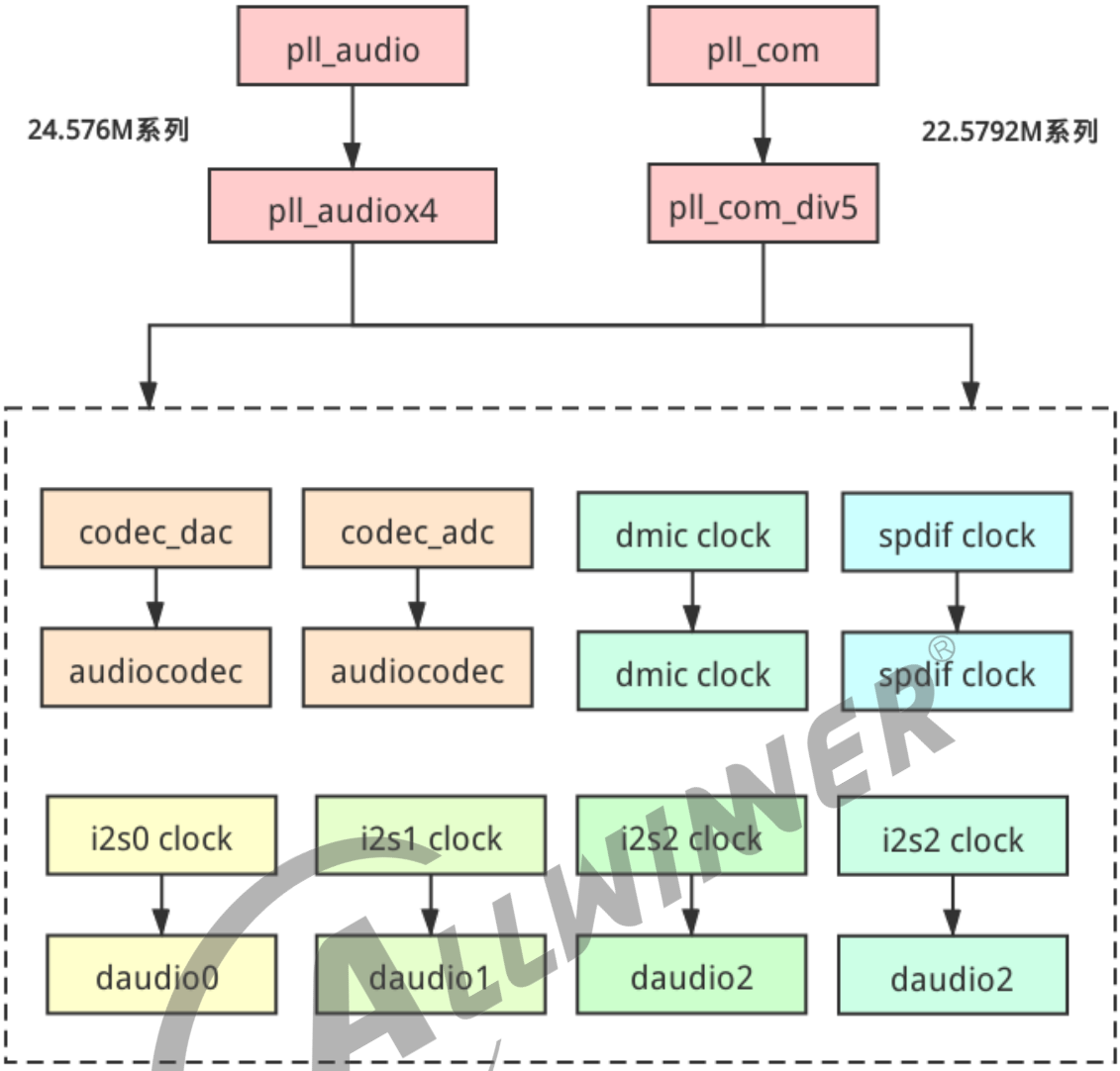


图 2-43: MR813 时钟源

2.17.2 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sun50iw10-codec.c              // codec驱动
├── sun50iw10-codec.h
├── sun50iw10-sndcodec.c           // codec machine驱动
├── sunxi-cpudai.c                 // codec platform驱动
├── sunxi-daudio.c                 // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-snddaudio.c              // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-dmic.c                   // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-snddmic.c                // dmic machine驱动
├── sunxi-snddmic.h
├── sunxi-pcm.c                    // 通用文件，提供注册platform驱动的接口及相关函数集
```

```
├── sunxi-pcm.h
├── sunxi-sndspdif.c                // spdif machine驱动
├── spdif-utils.c                   // spdif codec驱动
├── sunxi-spdif.c                   // spdif platform驱动
└── sunxi-spdif.h

linux-4.9/sound/soc/codecs/dmic.c   // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c     // daudio codec驱动
```

2.17.3 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出：
  - 一路单声道输出 LINEOUTLP/N
  - 一路立体声输出 HPOUTL,HPOUTR
- 两路模拟差分输入：MIC1P/N,MIC2P/N
- 支持耳机驱动电路
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 DRC

2.17.3.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun50iw10 Codec Support
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.17.3.2 DTS 配置

```
codec:codec@0x05096000 {
    /* MIC and headphone gain setting */
    mic1gain    = <0x17>;
    mic2gain    = <0x17>;
    headphonegain   = <0x00>;
    /* adc/dac DRC/HPF func enabled */
    adcdrc_cfg  = <0x00>;
    adchpf_cfg  = <0x00>;
    dacdrc_cfg  = <0x00>;
    dachpf_cfg  = <0x00>;
    /* Volume about */
    digital_vol     = <0x00>;
    lineout_vol     = <0x1a>;
    dac_digital_vol = <0x1A0A0>;
    /* Pa enabled about */
    pa_level    = <0x01>;
    pa_msleep_time  = <0xa0>;
    gpio-spk = <&pio PF 2 1 1 1 1>;
    /* regulator about */
    avcc-supply = <&reg_bldo1>;
    cpvin-supply = <&reg_bldo3>;
    status = "okay";
};

cpudai:cpudai-controller@0x050906000 {
    status = "okay";
};

sndcodec:sound@0 {
    status = "okay";
};
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
status	是否使用 sndcodec 驱动。disabled：不使用；okay：使用

cpudai 配置，即 platform 驱动的相关配置

cpudai 配置	cpudai 配置说明
status	是否使用 cpudai 驱动。disabled：不使用；okay：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
status	是否使用 codec 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
mic1gain	mic1 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
mic2gain	mic1 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
headphonegain	headphone 增益，可设定范围 0~0x07, -6dB/stop, 0: -0dB, 1: -6dB
digital_vol	初始化 digital volume，可设定范围 0~0x3f, 表示 0~-73.08dB, -1.16dB/step
lineout_vol	lineout volume，可设定范围 0~0x1f, 表示-43.5dB~0dB, 1.5dB/step
dac_digital_vol	初始化 dac digital volume，bit16 表示是否使能,bit[7:0] 以及 bit[15:8] 分别表示 DACL, DACR 的数字增益, 可设定范围 0~0xff, 表示-119.25~71.25dB, 0.75dB/step,0:Mutex, 1:-119.25dB
pa_level	PA 引脚使能方式。0: 低电平有效；1：高电平有效
pa_msleep_time	操作 PA 之后的延时时间 (用来避免 pop 音)
gpio-spk	PA 使能引脚
avcc-supply	AVCC 供电配置, 选择对应的 regulator 配置
cpvin-supply	CPVIN 供电配置, 选择对应的 regulator 配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.17.3.3 codec 数据通路

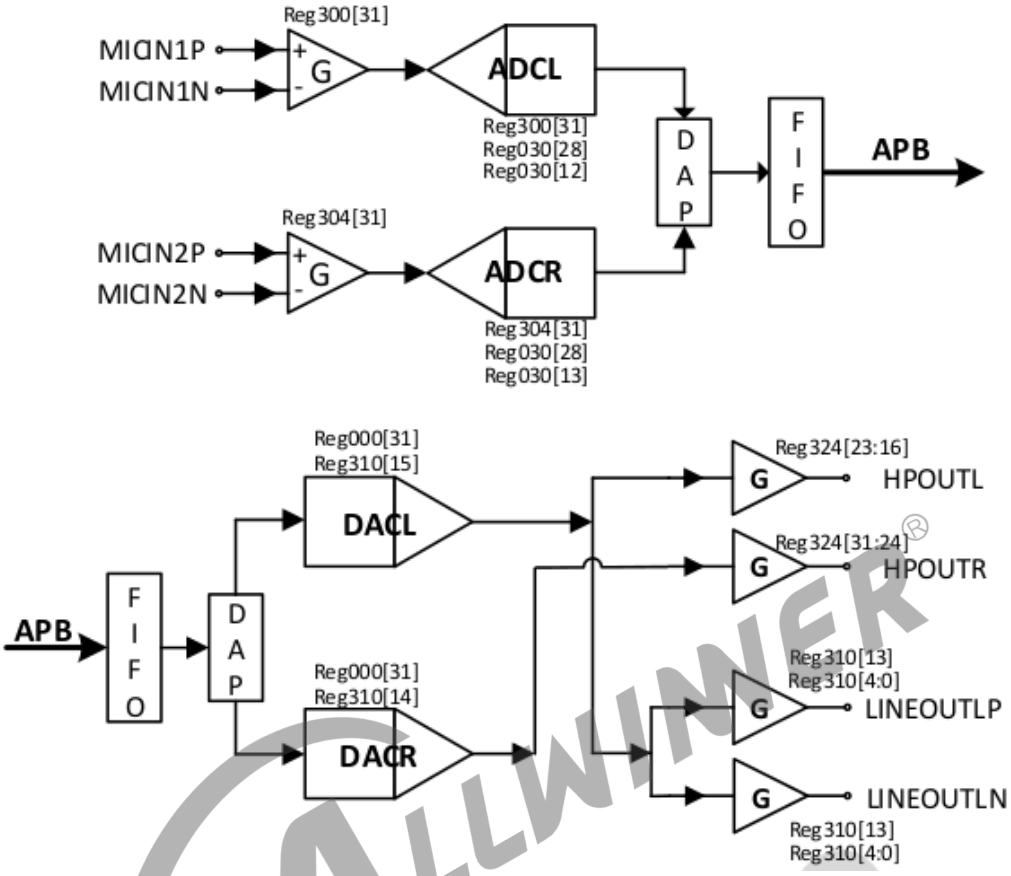


图 2-44: MR813 音频通路

```
通过Lineout播歌
Playback --> DACL --> LINEOUT Output Select --> LINEOUT

通过HPOUT播歌
Playback --> DACL --> HPOUTL --> Headphone
Playback --> DACR --> HPOUTR --> Headphone

如果HPOUT输出到模拟功放:
Playback --> DACL --> HPOUTL --> HpSpeaker
Playback --> DACR --> HPOUTR --> HpSpeaker

录音
MIC1 --> ADCL Input --> ADCL --> Capture
MIC2 --> ADCR Input --> ADCR --> Capture
```

MR813 所有控件如下表：

控件名称	功能	数值
digital volume	数字端音量设置	0~63, 表示-73.08~0dB

控件名称	功能	数值
MIC1 gain volume	MIC1 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
MIC2 gain volume	MIC1 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
LINEOUT volume	Lineout 音量设置	0~31, 表示-43.5~0dB
DAC volume	DACL,DACR 音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
ADC volume	ADCL,ADCR 音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
Headphone Volume	Headphone 音量设置	0~7, 表示-0dB~-42dB, -6dB/step
LINEOUT Output Select	Lineout 输出选择	0: 单端; 1: 差分
ADCL Input MIC1 Boost Switch	是否使能 MIC1 通路	0: 关闭; 1: 使能
ADCR Input MIC2 Boost Switch	是否使能 MIC2 通路	0: 关闭; 1: 使能
Headphone Switch	是否使能 Headphone 通路	0: 关闭; 1: 使能
HpSpeaker Switch	是否使能 Speaker 通路 (使用功放)	0: 关闭; 1: 使能
LINEOUT Switch	是否使能 Lineout 通路	0: 关闭; 1: 使能

2.17.4 Daudio

硬件特性

- 四路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 4 路 MCLK 输出

2.17.4.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
           <*> Allwinner Digital Audio Support
```

2.17.4.2 DTS 配置

```
daudio0:daudio@0x05090000 {
    mclk_div = <0x02>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    status = "okay";
};

snddaudio0:sound@3 {
    sunxi,snddaudio-codec = "ac107.1-0036";
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

snddaudio 配置	snddaudio 配置说明
	2: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.17.5 DMIC

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.17.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.17.5.2 DTS 配置

配置如下:

```
dmic:dmic-controller@0x05095000{
    status = "okay";
};
snddmic:sound@2{
    status = "okay";
};
```

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
status	是否使用 dmic 驱动。disabled：不使用；okay：使用

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
status	是否使用 snddmic 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.17.6 SPDIF

硬件特性

- 支持 S/PDIF_OUT 和 S/PDIF_IN
- 支持 mono 和 stereo 模式
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 输入支持 44.1KHz,48KHz 采样率
- 输出和输入支持 16bit,24bit 采样精度

2.17.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner SPDIF Support
```

2.17.6.2 DTS 配置

```
spdif:spdif-controller@0x05094000{
    status = "okay";
};
sndspdif:sound@1{
    status = "okay";
};
```

spdif 配置，即 platform 驱动的相关配置

spdif 配置	spdif 配置说明 |
status	是否使用 spdif 驱动。disabled：不使用；okay：使用

sndspdif 配置，即 machine 驱动的相关配置

sndspdif 配置	sndspdif 配置说明
status	是否使用 sndspdif 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.17.7 外挂 codec:ac107

MR813 标案 mr813-evb2 开发板上默认搭配了 AC107, 最高可录两通道

下面对 MR813 如何配置使用 ac107 作简单介绍

2.17.7.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
              <*> Allwinner Digital Audio Support
           CODEC drivers  --->
           <*> Sunxi AC107 Codec
```

2.17.7.2 DTS 配置

MR813 通过 twi1 控制 ac107, 而 i2s0 用于音频数据的传输

twi 配置:

```
twi1: twi@0x05002400{
    status = "okay";
    ac107@36 {
        compatible = "Allwinnertech,ac107_0";
        gpio-reset = "null";
        reg = <0x36>;
        status = "okay";
    };
};
```

i2s 配置:

```
daudio0:daudio@0x05090000 {
    mclk_div     = <0x02>;
    frametype    = <0x00>;
    tdm_config   = <0x01>;
    sign_extend      = <0x00>;
    tx_data_mode     = <0x00>;
    rx_data_mode     = <0x00>;
    msb_lsb_first    = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    status = "okay";
};

snddaudio0:sound@3 {
    sunxi,snddaudio-codec = "ac107.1-0036";
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
    audio_format     = <0x01>;
    daudio_master    = <0x04>;
    signal_inversion  = <0x01>;
    status = "okay";
};

其中注意配置mclk_div=2,ac107要求MCLK为11.288M或者11.2896M;
audio_format=1表示标准i2s格式;
daudio_master=4表示AC107作为从设备
signal_inversion=1表示BCLK,LRCK都是normal模式(BCLK下降沿采样,LRCK低电平为左声道)
```

2.17.7.3 使用

进入系统后，通过命令 cat /proc/asound/cards 列出当前声卡信息，如果发现 ac107 相关声卡，说明已经正常加载驱动

无需额外设置音频通路，可直接用下面命令进行录音:

```
arecord -Dhw:sndac10710036 -f S16_LE -r 16000 -c 2 /tmp/test.wav

另外可以通过下面命令调整增益
amixer -D hw:sndac10710036 cset name='Channel 1 PGA Gain' 25
amixer -D hw:sndac10710036 cset name='Channel 2 PGA Gain' 25
```

2.17.8 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.17.8.1 播放

```
通过HPOUT->Speaker播放
amixer -Dhw:audiocodec cset name='Headphone Switch' 1
amixer -Dhw:audiocodec cset name='Headphone Volume' 3
amixer -Dhw:audiocodec cset name='HpSpeaker Switch' 1

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
或者利用默认/etc/asound.conf配置的pcm设备进行播放:
aplay -Ddefault /mnt/UDISK/1KHz_0dB_16000.wav

通过Lineout播放
amixer -D hw:audiocodec cset name='LINEOUT Output Select' 'DAC_DIFFER'
amixer -D hw:audiocodec cset name='LINEOUT Switch' 1
amixer -D hw:audiocodec cset name='LINEOUT volume' 20
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav

通过HPOUT播放
amixer -Dhw:audiocodec cset name='Headphone Switch' 1
amixer -Dhw:audiocodec cset name='Headphone Volume' 3
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

2.17.8.2 录音

```
通过内部的MIC1,MIC2录制两通道
amixer -D hw:audiocodec cset name='ADCL Input MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='ADCR Input MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 gain volume' 19
amixer -D hw:audiocodec cset name='MIC2 gain volume' 19

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav


通过外部AC107的MIC1,MIC2录制两通道
amixer -D hw:sndac10710036 cset name='Channel 1 PGA Gain' 25
amixer -D hw:sndac10710036 cset name='Channel 2 PGA Gain' 25

arecord -Dhw:sndac10710036 -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.18 R818 音频接口

R818 包含 7 个音频模块，分别是内置 AudioCodec,Daudio0,Daudio1,Daudio2,Daudio3,Dmic,Spdif。

2.18.1 时钟源

R818 中，7 个音频模块的时钟源来自 pll_audio 以及 pll_com_div5。

pll_audio 输出 24.576M, 而 pll_com_div5 输出 90.3168M(再由具体模块，如 dac 的 4 分频得到 22.5792M) 的时钟，分别支持 48k 系列，44.1k 系列的播放录音。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

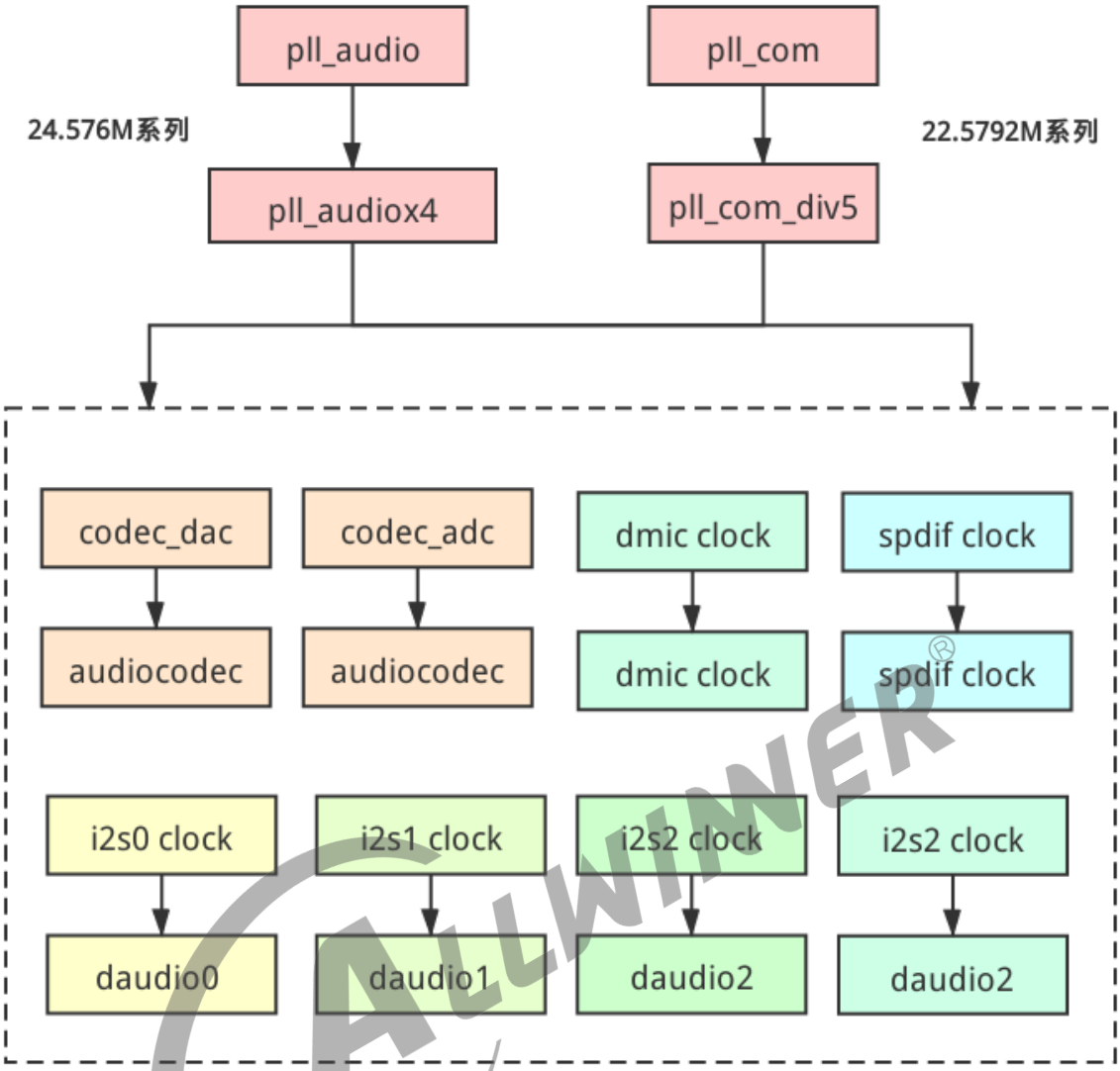


图 2-45: R818 时钟源

2.18.2 代码结构

```
linux-4.9/sound/soc/sunxi/
├── sun50iw10-codec.c              // codec驱动
├── sun50iw10-codec.h
├── sun50iw10-sndcodec.c           // codec machine驱动
├── sunxi-cpudai.c                 // codec platform驱动
├── sunxi-daudio.c                 // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-snddaudio.c              // daudio machine驱动
├── sunxi-snddaudio.h
├── sunxi-dmic.c                   // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-snddmic.c                // dmic machine驱动
├── sunxi-snddmic.h
├── sunxi-pcm.c                    // 通用文件，提供注册platform驱动的接口及相关函数集
```

```
├── sunxi-pcm.h
├── sunxi-sndspdif.c                // spdif machine驱动
├── spdif-utils.c                   // spdif codec驱动
├── sunxi-spdif.c                   // spdif platform驱动
└── sunxi-spdif.h

linux-4.9/sound/soc/codecs/dmic.c   // dmic codec驱动
linux-4.9/sound/soc/soc-utils.c     // daudio codec驱动
```

2.18.3 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,24bit 采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出：
  - 一路单声道输出 LINEOUTLP/N
  - 一路立体声输出 HPOUTL,HPOUTR
- 两路模拟差分输入：MIC1P/N,MIC2P/N
- 支持耳机驱动电路
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 DRC

2.18.3.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun50iw10 Codec Support
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.3.2 DTS 配置

```
codec:codec@0x05096000 {
    /* MIC and headphone gain setting */
    mic1gain    = <0x17>;
    mic2gain    = <0x17>;
    headphonegain   = <0x00>;
    /* adc/dac DRC/HPF func enabled */
    adcdrc_cfg  = <0x00>;
    adchpf_cfg  = <0x00>;
    dacdrc_cfg  = <0x00>;
    dachpf_cfg  = <0x00>;
    /* Volume about */
    digital_vol     = <0x00>;
    lineout_vol     = <0x1a>;
    dac_digital_vol = <0x1A0A0>;
    /* Pa enabled about */
    pa_level    = <0x01>;
    pa_msleep_time  = <0xa0>;
    gpio-spk = <&pio PF 2 1 1 1 1>;
    /* regulator about */
    avcc-supply = <&reg_bldo1>;
    cpvin-supply = <&reg_bldo3>;
    status = "okay";
};

cpudai:cpudai-controller@0x050906000 {
    status = "okay";
};

sndcodec:sound@0 {
    status = "okay";
};
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
status	是否使用 sndcodec 驱动。disabled：不使用；okay：使用

cpudai 配置，即 platform 驱动的相关配置

cpudai 配置	cpudai 配置说明
status	是否使用 cpudai 驱动。disabled：不使用；okay：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
status	是否使用 codec 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

codec 配置	codec 配置说明
mic1gain	mic1 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
mic2gain	mic1 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
headphonegain	headphone 增益，可设定范围 0~0x07, -6dB/stop, 0: -0dB, 1: -6dB
digital_vol	初始化 digital volume，可设定范围 0~0x3f, 表示 0~-73.08dB, -1.16dB/step
lineout_vol	lineout volume，可设定范围 0~0x1f, 表示-43.5dB~0dB, 1.5dB/step
dac_digital_vol	初始化 dac digital volume，bit16 表示是否使能,bit[7:0] 以及 bit[15:8] 分别表示 DACL, DACR 的数字增益, 可设定范围 0~0xff, 表示-119.25~71.25dB, 0.75dB/step,0:Mutex, 1:-119.25dB
pa_level	PA 引脚使能方式。0: 低电平有效；1：高电平有效
pa_msleep_time	操作 PA 之后的延时时间 (用来避免 pop 音)
gpio-spk	PA 使能引脚
avcc-supply	AVCC 供电配置, 选择对应的 regulator 配置
cpvin-supply	CPVIN 供电配置, 选择对应的 regulator 配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.3.3 codec 数据通路

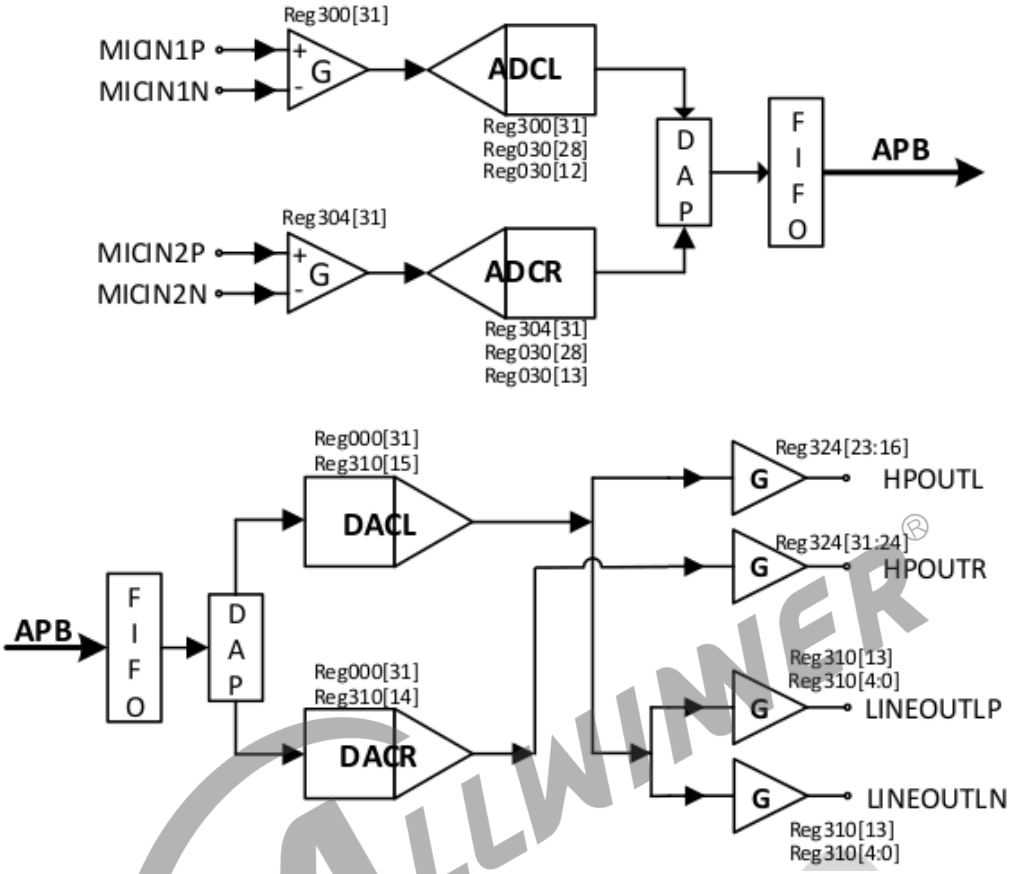


图 2-46: R818 音频通路

```
通过Lineout播歌
Playback --> DACL --> LINEOUT Output Select --> LINEOUT

通过HPOUT播歌
Playback --> DACL --> HPOUTL --> Headphone
Playback --> DACR --> HPOUTR --> Headphone

如果HPOUT输出到模拟功放:
Playback --> DACL --> HPOUTL --> HpSpeaker
Playback --> DACR --> HPOUTR --> HpSpeaker

录音
MIC1 --> ADCL Input --> ADCL --> Capture
MIC2 --> ADCR Input --> ADCR --> Capture
```

R818 所有控件如下表：

控件名称	功能	数值
digital volume	数字端音量设置	0~63, 表示-73.08~0dB

控件名称	功能	数值
MIC1 gain volume	MIC1 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
MIC2 gain volume	MIC1 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
LINEOUT volume	Lineout 音量设置	0~31, 表示-43.5~0dB
DAC volume	DACL,DACR 音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
ADC volume	ADCL,ADCR 音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
Headphone Volume	Headphone 音量设置	0~7, 表示-0dB~-42dB, -6dB/step
LINEOUT Output Select	Lineout 输出选择	0: 单端; 1: 差分
ADCL Input MIC1 Boost Switch	是否使能 MIC1 通路	0: 关闭; 1: 使能
ADCR Input MIC2 Boost Switch	是否使能 MIC2 通路	0: 关闭; 1: 使能
Headphone Switch	是否使能 Headphone 通路	0: 关闭; 1: 使能
HpSpeaker Switch	是否使能 Speaker 通路 (使用功放)	0: 关闭; 1: 使能
LINEOUT Switch	是否使能 Lineout 通路	0: 关闭; 1: 使能

2.18.4 Daudio

硬件特性

- 四路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 4 路 MCLK 输出

2.18.4.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
           <*> Allwinner Digital Audio Support
```

2.18.4.2 DTS 配置

```
daudio0:daudio@0x05090000 {
    mclk_div = <0x02>;
    frametype = <0x00>;
    tdm_config = <0x01>;
    sign_extend = <0x00>;
    tx_data_mode = <0x00>;
    rx_data_mode = <0x00>;
    msb_lsb_first = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    status = "okay";
};

snddaudio0:sound@3 {
    sunxi,snddaudio-codec = "ac107.1-0036";
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
    audio_format = <0x01>;
    daudio_master = <0x04>;
    signal_inversion = <0x01>;
    status = "okay";
};
```

snddaudio0 配置，即 daudio0 machine 驱动的相关配置

snddaudio 配置	snddaudio 配置说明
snddaudio0_used	是否使用 snddaudio 驱动。0：不使用；1：使用
daudio_master	1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master), 即 daudio 接口作为 slave, codec 作为 master 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master), 一般不用

snddaudio 配置	snddaudio 配置说明
	2: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave), 一般不用 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave), 即 daudio 接口作为 master, codec 作为 slave
audio_format	1: SND_SOC_DAIFMT_I2S(standard i2s format) 2: SND_SOC_DAIFMT_RIGHT_J(right justfied format) 3: SND_SOC_DAIFMT_LEFT_J(left justfied format) 4: SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge) 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)

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
slot_width_select	支持 8bit, 16bit, 32bit 宽度
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
msb_lsb_first	0: msb first; 1: lsb first
sign_extend	0: zero pending; 1: sign extend
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: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div
tdm_config	0: pcm mode; 1: i2s mode
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.5 DMIC

硬件特性

- 支持 8 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.18.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner DMIC Support
```

2.18.5.2 DTS 配置

配置如下:

```
dmic:dmic-controller@0x05095000{
    status = "okay";
};
snddmic:sound@2{
    status = "okay";
};
```

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
status	是否使用 dmic 驱动。disabled：不使用；okay：使用

snddmic 配置，即 machine 驱动的相关配置

snddmic 配置	snddmic 配置说明
status	是否使用 snddmic 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.6 SPDIF

硬件特性

- 支持 S/PDIF_OUT 和 S/PDIF_IN
- 支持 mono 和 stereo 模式
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 输入支持 44.1kHz,48kHz 采样率
- 输出和输入支持 16bit,24bit 采样精度

2.18.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner SPDIF Support
```

2.18.6.2 DTS 配置

```
spdif:spdif-controller@0x05094000{
    status = "okay";
};
sndspdif:sound@1{
    status = "okay";
};
```

spdif 配置，即 platform 驱动的相关配置

spdif 配置	spdif 配置说明 |
status	是否使用 spdif 驱动。disabled：不使用；okay：使用

sndspdif 配置，即 machine 驱动的相关配置

sndspdif 配置	sndspdif 配置说明
status	是否使用 sndspdif 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.7 外挂 codec:ac107

R818 标案 r818-evb2 开发板上默认搭配了 AC107, 与内部两个 MIC 搭配使用，可以实现 2+1 或者 2+2 方案。

下面对 R818 如何配置使用 ac107 作简单介绍

2.18.7.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
              <*> Allwinner Digital Audio Support
           CODEC drivers  --->
           <*> Sunxi AC107 Codec
```

2.18.7.2 DTS 配置

R818 通过 twi1 控制 ac107, 而 i2s0 用于音频数据的传输

twi 配置:

```
twi1: twi@0x05002400{
    status = "okay";
    ac107@36 {
        compatible = "Allwinnertech,ac107_0";
        gpio-reset = "null";
        reg = <0x36>;
        status = "okay";
    };
};
```

i2s 配置:

```
daudio0:daudio@0x05090000 {
    mclk_div     = <0x02>;
    frametype    = <0x00>;
    tdm_config   = <0x01>;
    sign_extend      = <0x00>;
    tx_data_mode     = <0x00>;
    rx_data_mode     = <0x00>;
    msb_lsb_first    = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    status = "okay";
};

snddaudio0:sound@3 {
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    sunxi,snddaudio-codec = "ac107.1-0036";
    sunxi,snddaudio-codec-dai = "ac107-pcm0";
    audio_format     = <0x01>;
    daudio_master    = <0x04>;
    signal_inversion  = <0x01>;
    status = "okay";
};

其中注意配置mclk_div=2,ac107要求MCLK为11.288M或者11.2896M;
audio_format=1表示标准i2s格式;
daudio_master=4表示AC107作为从设备
signal_inversion=1表示BCLK,LRCK都是normal模式(BCLK下降沿采样,LRCK低电平为左声道)
```

2.18.7.3 使用

进入系统后，通过命令 cat /proc/asound/cards 列出当前声卡信息，如果发现 ac107 相关声卡，说明已经正常加载驱动

无需额外设置音频通路，可直接用下面命令进行录音:

```
arecord -Dhw:sndac10710036 -f S16_LE -r 16000 -c 2 /tmp/test.wav

另外可以通过下面命令调整增益
amixer -D hw:sndac10710036 cset name='Channel 1 PGA Gain' 25
amixer -D hw:sndac10710036 cset name='Channel 2 PGA Gain' 25
```

2.18.8 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.18.8.1 播放

```
通过HPOUT->Speaker播放
amixer -Dhw:audiocodec cset name='Headphone Switch' 1
amixer -Dhw:audiocodec cset name='Headphone Volume' 3
amixer -Dhw:audiocodec cset name='HpSpeaker Switch' 1

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
或者利用默认/etc/asound.conf配置的pcm设备进行播放:
aplay -Ddefault /mnt/UDISK/1KHz_0dB_16000.wav
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.18.8.2 录音

```
通过内部的MIC1,MIC2录制两通道
amixer -D hw:audiocodec cset name='ADCL Input MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='ADCR Input MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='MIC1 gain volume' 19
amixer -D hw:audiocodec cset name='MIC2 gain volume' 19

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav


通过外部AC107的MIC1,MIC2录制两通道
amixer -D hw:sndac10710036 cset name='Channel 1 PGA Gain' 25
amixer -D hw:sndac10710036 cset name='Channel 2 PGA Gain' 25

arecord -Dhw:sndac10710036 -f S16_LE -r 16000 -c 2 /tmp/test.wav


2+2方案，录制R818 2MIC + AC107 2MIC:
arecord -DCaptureAec  -f S16_LE -r 16000 -c 4 /tmp/test.wav
```

2.19 R528 音频接口

R528 包含多个音频模块，分别是内置 AudioCodec,I2S0,I2S1,I2S2,DMIC,SPDIF。

2.19.1 时钟源

R528 中，音频模块的时钟源来自 pll_audio0 以及 pll_audio1_div5。

pll_audio0 可以输出 22.5792M 的时钟, 而 pll_audio1_div5 输出 24.576M 的时钟，分别支持 44.1k 系列，48k 系列的播放录音。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

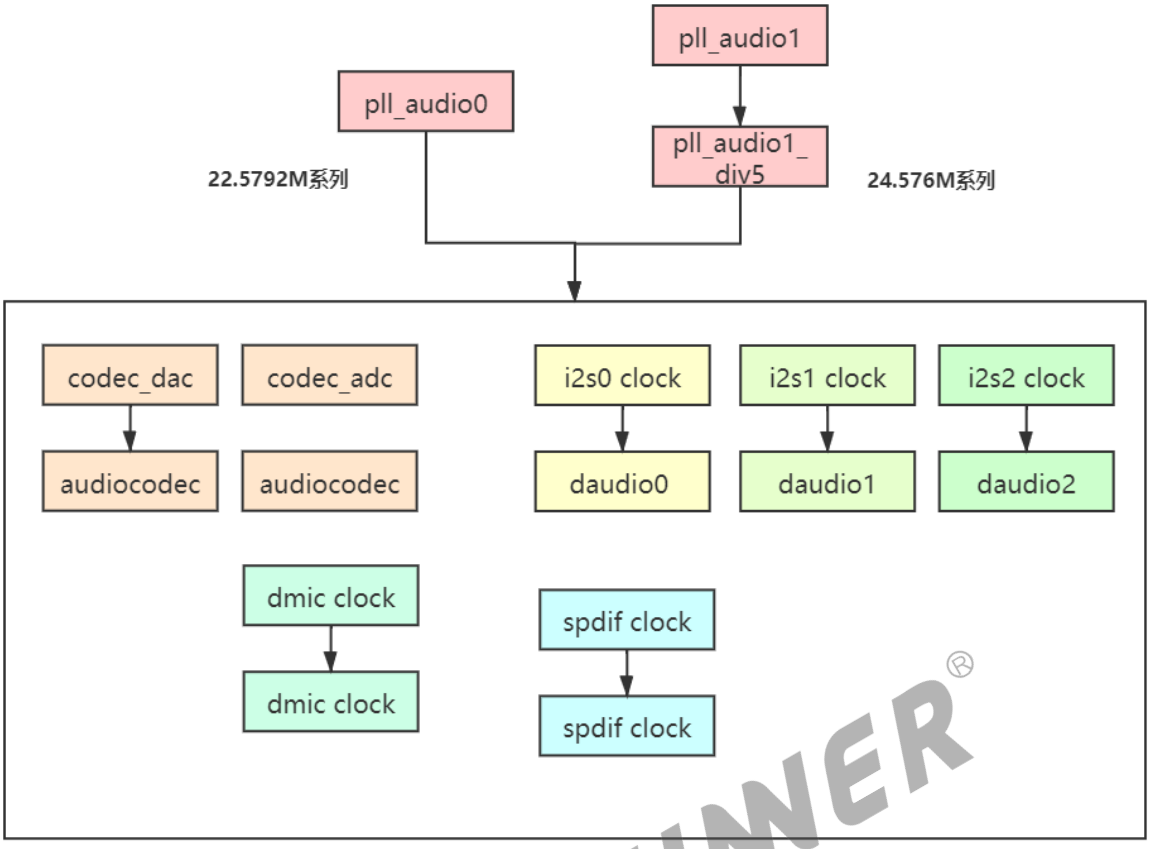


图 2-47：R528 时钟源

2.19.2 代码结构

```
linux-5.4/sound/soc/sunxi/
├── sun8iw20-codec.c              // codec驱动
├── sun8iw20-codec.h
├── sun8iw20-sndcodec.c           // codec machine驱动
├── sunxi-dummy-cpudai.c          // codec platform驱动
├── sunxi-daudio.c                // daudio platform驱动
├── sunxi-daudio.h
├── sunxi-simple-card.c           // 通用machine驱动
├── sunxi-dmic.c                  // dmic platform驱动
├── sunxi-dmic.h
├── sunxi-pcm.c                   // 通用文件，提供注册platform驱动的接口及相关函数集
├── sunxi-pcm.h
├── sunxi-spdif.c                 // spdif platform驱动
└── sunxi-spdif.h

linux-5.4/sound/soc/codecs/dmic.c   // dmic codec驱动
linux-5.4/sound/soc/soc-utils.c     // daudio codec驱动(snd-soc-dummy)
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.19.3 AudioCodec

硬件特性

- 两路 DAC
  - 支持 16bit,20bit 有效采样精度
  - 支持 8KHz~192KHz 采样率
- 三路 ADC
  - 支持 16bit,20bit 有效采样精度
  - 支持 8KHz~48KHz 采样率
- 两路模拟输出：
  - 一路立体声输出 HPOUTL,HPOUTR
  - 一路立体声差分输出 LINEOUTLP/N,LINEOUTRP/N
- 五路模拟差分输入：MIC1P/N,MIC2P/N
  - 三路差分麦克风输入 MIC1P/N,MIC2P/N,MIC3P/N
  - 一路立体声 line-in 输入 LINEINL,LINEINR
  - 一路立体声 FM-in 输入 FMINL,FMINR
- 支持耳机驱动电路
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 DRC

2.19.3.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
         <*> Allwinner Sun8iw20 Codec Support
```

2.19.3.2 DTS 配置

```
&codec {
        /* MIC and headphone gain setting */
        mic1gain        = <0x1F>;
        mic2gain        = <0x1F>;
        mic3gain        = <0x1F>;
        /* ADC/DAC DRC/HPF func enabled */
        /* 0x1:DAP_HP_EN; 0x2:DAP_SPK_EN; 0x3:DAP_HPSPK_EN */
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
        adcdrc_cfg      = <0x0>;
        adchpf_cfg      = <0x1>;
        dacdrc_cfg      = <0x0>;
        dachpf_cfg      = <0x0>;
        /* Volume about */
        digital_vol     = <0x00>;
        lineout_vol     = <0x1a>;
        headphonegain   = <0x03>;
        /* Pa enabled about */
        pa_level        = <0x01>;
        pa_pwr_level    = <0x01>;
        pa_msleep_time  = <0x78>;
        gpio-spk        = <&pio PF 2 GPIO_ACTIVE_HIGH>;
        gpio-spk-pwr    = <&pio PF 4 GPIO_ACTIVE_HIGH>;
        /* regulator about */
        /* avcc-supply  = <&reg_aldo1>; */
        /* hpvcc-supply = <&reg_eldo1>; */
        status = "okay";
};

&sndcodec {
        hp_detect_case  = <0x00>;
        jack_enable     = <0x01>;
        status = "okay";
};

&dummy_cpudai {
        /* CMA config about */
        playback_cma    = <128>;
        capture_cma     = <256>;
        status = "okay";
};
```

sndcodec 配置，即 machine 驱动的相关配置

sndcodec 配置	sndcodec 配置说明
status	是否使用 sndcodec 驱动。disabled：不使用；okay：使用
hp_detect_case	耳机检测电平，0: 低电平；1: 高电平
jack_enable	是否初始化耳机相关代码。0: 不使用; 1: 使用

dummy_cpudai 配置，即 platform 驱动的相关配置

dummy_cpudai 配置	dummy_cpudai 配置说明
playback_cma	配置 playback 的 dma buffer 大小，单位 KB
capture_cma	配置 capture 的 dma buffer 大小，单位 KB
status	是否使用 cpudai 驱动。disabled：不使用；okay：使用

codec 配置，即内置 audiocodec 驱动的相关配置

codec 配置	codec 配置说明
status	是否使用 codec 驱动。disabled：不使用；okay：使用
mic1gain	mic1 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
mic2gain	mic2 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
mic3gain	mic3 增益，可设定范围 0~0x1f, 0:0dB, 1~0x3:6dB, 0x4~0x1f:9~36dB, 1dB/step, 一般设置 0x13, 即 24dB
adchpf_cfg	是否使用 ADC HPF 功能，1: 使用; 0: 不使用
digital_vol	初始化 digital volume，可设定范围 0~0x3f, 表示 0~-73.08dB, -1.16dB/step
lineout_vol	lineout volume，可设定范围 0~0x1f, 表示-43.5dB~0dB, 1.5dB/step
headphonegain	headphone 增益，可设定范围 0~0x07, 表示 0dB~-42dB, -6dB/step
pa_level	PA 引脚使能方式。0: 低电平有效；1：高电平有效
pa_pwr_level	PA 供电引脚使能方式。0: 低电平有效；1：高电平有效
gpio-spk	PA 使能引脚
gpio-spk-pwr	PA 供电使能引脚

2.19.3.3 codec 数据通路

R528 音频通路请参考《R528_User_Manual》中 AudioCodec 章节的《8.4.3.4Data Path Diagram》

```
通过Lineout播歌
Playback --> DACL --> LINEOUTL Output Select --> LINEOUTL --> LINEOUT
Playback --> DACR --> LINEOUTR Output Select --> LINEOUTR --> LINEOUT

通过HPOUT播歌
Playback --> DACL --> HPOUTL --> Headphone
Playback --> DACR --> HPOUTR --> Headphone

如果HPOUT输出到模拟功放:
Playback --> DACL --> HPOUTL --> HpSpeaker
Playback --> DACR --> HPOUTR --> HpSpeaker

录音
MIC1 --> MIC1 Input Select --> ADC1 Input --> ADC1 --> Capture
MIC2 --> MIC2 Input Select --> ADC2 Input --> ADC2 --> Capture
MIC3 --> MIC3 Input Select --> ADC3 Input --> ADC3 --> Capture

LINE-in录音
LINEINL --> ADC1 Input --> ADC1 --> Capture
LINEINR --> ADC2 Input --> ADC2 --> Capture

FM-in录音
```

```
FMINL --> ADC1 Input --> ADC1 --> Capture
FMINR --> ADC2 Input --> ADC2 --> Capture
```

R528 所有控件如下表：

控件名称	功能	数值
MIC1 gain volume	MIC1 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
MIC2 gain volume	MIC2 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
MIC3 gain volume	MIC3 增益	0~31, 表示 0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
FMINL gain volume	FMINL 增益	0:0dB; 1:6dB
FMINR gain volume	FMINR 增益	0:0dB; 1:6dB
LINEINL gain volume	LINEINL 增益	0:0dB; 1:6dB
LINEINR gain volume	LINEINR 增益	0:0dB; 1:6dB
MIC1 Input Select	MIC1 输入模式	0: 差分输入; 1: 单端输入
MIC2 Input Select	MIC2 输入模式	0: 差分输入; 1: 单端输入
MIC3 Input Select	MIC3 输入模式	0: 差分输入; 1: 单端输入
ADC1 volume	ADC1 数字音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
ADC2 volume	ADC2 数字音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
ADC3 volume	ADC3 数字音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
ADC1 Input MIC1 Boost Switch	是否使能 ADC1->MIC1 的通路	0: 关闭; 1: 使能

版权所有 © 珠海全志科技股份有限公司。保留一切权利

控件名称	功能	数值
ADC2 Input MIC2 Boost Switch	是否使能 ADC2->MIC2 的通路	0: 关闭; 1: 使能
ADC3 Input MIC3 Boost Switch	是否使能 ADC3->MIC3 的通路	0: 关闭; 1: 使能
ADC1 Input FMINL Switch	是否使能 ADC1->FMINL 的通路	0: 关闭; 1: 使能
ADC2 Input FMINR Switch	是否使能 ADC2->FMINR 的通路	0: 关闭; 1: 使能
ADC1 Input LINEINL Switch	是否使能 ADC1->LINEINL 的通路	0: 关闭; 1: 使能
ADC2 Input LINEINR Switch	是否使能 ADC2->LINEINR 的通路	0: 关闭; 1: 使能
LINEOUT volume	Lineout 音量设置	0~31, 表示-43.5~0dB
digital volume	数字端音量设置	0~63, 表示-73.08~0dB
DAC volume	DACL,DACR 音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~71.24dB, 0.75dB/step, 默认 0xA0=0dB
Headphone volume	Headphone 音量设置	0~7, 表示-0dB~-42dB, -6dB/step
LINEOUTL Output Select	Lineout left 输出选择	0: 单端; 1: 差分
LINEOUTR Output Select	Lineout right 输出选择	0: 单端; 1: 差分
LINEOUT Switch	是否使能 Lineout 通路	0: 关闭; 1: 使能
Headphone Switch	是否使能 Headphone 通路	0: 关闭; 1: 使能
HpSpeaker Switch	是否使能 Speaker 通路 (使用功放)	0: 关闭; 1: 使能

2.19.4 Daudio

硬件特性

- 四路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持 mono 和 stereo 模式，最高支持 8 通道
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 16,24,32bit 采样精度
- 支持 3 路 MCLK 输出

2.19.4.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support  --->
             <*> Allwinner Audio Simple Card
             <*>   Allwinner Digital Audio Support
```

2.19.4.2 DTS 配置

```
&daudio0 {
        mclk_div        = <0x01>;
        frametype       = <0x00>;
        tdm_config      = <0x01>;
        sign_extend     = <0x00>;
        msb_lsb_first   = <0x00>;
        pcm_lrck_period = <0x80>;
        slot_width_select = <0x20>;
        pinctrl-names   = "default", "sleep";
        pinctrl-0       = <&daudio0_pins_a>;
        pinctrl-1       = <&daudio0_pins_b>;
        pinctrl_used    = <0x0>;
        status = "disabled";
};

&sounddaudio0 {
        status = "disabled";
        simple-audio-card,name = "snddaudio0";
        /* simple-audio-card,frame-master = <&daudio0_master>; */
        /* simple-audio-card,bitclock-master = <&daudio0_master>; */
        /* simple-audio-card,bitclock-inversion; */
        /* simple-audio-card,frame-inversion; */
        daudio0_master: simple-audio-card,codec {
                /* sound-dai = <&ac108>; */
        };
};
```

daudio0 配置，即 daudio0 platform 驱动的相关配置

daudio 配置	daudio 配置说明
mclk_div	0: not output(normal setting this); 1/2/4/6/8/12/16/24/32/48/64/96/128/176/192: 给外部 codec 提供时钟，频率是 pll_audio/mclk_div

daudio 配置	daudio 配置说明
frametype	0: short frame = 1 clock width; 1: long frame = 2 clock width
tdm_config	0: pcm mode; 1: i2s mode
sign_extend	0: zero pending; 1: sign extend
msb_lsb_first	0: msb first; 1: lsb first
pcm_lrck_period	一般可配置 16/32/64/128/256 个 bclk
slot_width_select	支持 8bit, 16bit, 32bit 宽度
tx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
rx_data_mode	0: 16bit linear PCM;1: reserved;2: 8bit u-law;3: 8bit a-law
playback_cma	配置 playback 的 dma buffer 大小，单位 KB
capture_cma	配置 capture 的 dma buffer 大小，单位 KB

sounddaudio0 配置，即 daudio0 machine 驱动的相关配置

sounddaudio 配置	sounddaudio 配置说明
status	是否使用 snddaudio 驱动。disabled：不使用；okay：使用
simple-audio-card,name	声卡名称
simple-audio-card,format	i2s,right_j,left_j,dsp_a,dsp_b
simple-audio-card,frame-master	配置 frame clk 主从关系, 不配置则是 SoC 作为主,codec 作为 slave; 如果配置了 codec 节点，则 codec 作为主，SoC 作为从
simple-audio-card,bitclock-master	配置 bit clk 主从关系, 不配置则是 SoC 作为主,codec 作为 slave; 如果配置了 codec 节点，则 codec 作为主，SoC 作为从
simple-audio-card,bitclock-inversion	配置 bit clk 极性取反; 不配置则是正常极性
simple-audio-card,frame-inversion	配置 frame clk 极性取反; 不配置则是正常极性
simple-audio-card,capture_only	仅支持录音流
simple-audio-card,playbcak_only	仅支持播放流

2.19.5 DMIC

硬件特性

- 支持 8 路输入

- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.19.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
           <*> Allwinner Audio Simple Card
           <*>   Allwinner DMIC Support
```

2.19.5.2 DTS 配置

配置如下:

```
&dmic {
        pinctrl-names   = "default","sleep";
        pinctrl-0       = <&dmic_pins_a>;
        pinctrl-1       = <&dmic_pins_b>;
        status = "okay";
};

&dmic_codec {
        status = "okay";
};

&sounddmic {
        status = "okay";
};
```

dmic 配置，即 platform 驱动的相关配置

dmic 配置	dmic 配置说明
status	是否使用 dmic 驱动。disabled：不使用；okay：使用
capture_cma	配置 capture 的 dma buffer 大小，单位 KB
data_vol	DATA0~DATA3 的数字增益, 默认配置 0xB0 即 12dB
rx_chmap	通道映射, 默认配置 0x76543210 表示按照默认通道顺序

dmic_codec 配置，即 codec 驱动的相关配置

dmic_codec 配置	dmic_codec 配置说明
status	是否使用 dmic_codec 驱动。disabled：不使用；okay：使用

版权所有 © 珠海全志科技股份有限公司。保留一切权利

sounddmic 配置，即 machine 驱动的相关配置

sounddmic 配置	sounddmic 配置说明
status	是否使用 sounddmic 驱动。disabled：不使用；okay：使用
simple-audio-card,name	声卡名称
simple-audio-card,capture_only	仅支持录音流

2.19.6 SPDIF

硬件特性

- 支持 S/PDIF_OUT 和 S/PDIF_IN
- 支持 mono 和 stereo 模式
- 输出支持 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz, 192kHz 采样率
- 输入支持 44.1KHz,48KHz 采样率
- 输出和输入支持 16bit,24bit 采样精度

2.19.6.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
           <*> Allwinner Audio Simple Card
           <*>   Allwinner SPDIF Support
```

2.19.6.2 DTS 配置

```
&spdif {
        pinctrl-names   = "default","sleep";
        pinctrl-0       = <&spdif_pins_a>;
        pinctrl-1       = <&spdif_pins_b>;
        status = "okay";
};

&soundspdif {
        status = "okay";
};
```

spdif 配置，即 platform 驱动的相关配置

spdif 配置	spdif 配置说明 |
status	是否使用 spdif 驱动。disabled：不使用；okay：使用

soundspdif 配置，即 machine 驱动的相关配置

soundspdif 配置	soundspdif 配置说明
status	是否使用 sndspdif 驱动。disabled：不使用；okay：使用

2.19.7 外挂 codec:ac107

下面对 R528 如何配置使用 ac107 作简单介绍

2.19.7.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
           <*> Allwinner Audio Simple Card
           <*>   Allwinner Digital Audio Support
         CODEC drivers  --->
         <*> Sunxi AC107 Codec
```

2.19.7.2 DTS 配置

R528 通过 twi0 控制 ac107, 而 i2s2 用于音频数据的传输

twi 配置:

```
&twi0 {
        clock-frequency = <400000>;
        pinctrl-0 = <&twi0_pins_a>;
        pinctrl-1 = <&twi0_pins_b>;
        pinctrl-names = "default", "sleep";
        status = "okay";

        ac107: ac107@36 {
                #sound-dai-cells = <0>;
                compatible = "Allwinnertech,ac107_0";
                /*compatible = "ac107_0";*/
                reg = <0x36>;
```

```
                status = "okay";
        };
};
```

i2s 配置:

```
&daudio2 {
        mclk_div        = <0x02>;
        frametype       = <0x00>;
        tdm_config      = <0x01>;
        sign_extend     = <0x00>;
        tx_data_mode    = <0x00>;
        rx_data_mode    = <0x00>;
        msb_lsb_first   = <0x00>;
        pcm_lrck_period = <0x80>;
        slot_width_select = <0x20>;
        asrc_function_en  = <0x00>;
        pinctrl-names   = "default", "sleep";
        pinctrl-0       = <&daudio2_pins_a &daudio2_pins_b &daudio2_pins_c>;
        pinctrl-1       = <&daudio2_pins_d>;
        pinctrl_used    = <0x1>;
        daudio_type     = <0x0>;
        status = "okay";
};

&sounddaudio2 {
        status = "okay";
        simple-audio-card,name = "ac107";
        simple-audio-card,format = "i2s";
        simple-audio-card,capture_only;
        daudio2_master: simple-audio-card,codec {
                sound-dai = <&ac107>;
        };
};

其中注意配置mclk_div=2,ac107要求MCLK为11.288M或者11.2896M;
使用标准i2s格式;
AC107作为从设备
BCLK,LRCK都是normal模式，即不用配置bitclock-inversion,frame-inversion
simple-audio-card,capture_only表示只注册录音流，因为ac107声卡仅支持录音流
simple-audio-card,name声卡名字是ac107
```

2.19.7.3 使用

进入系统后，通过命令 cat /proc/asound/cards 列出当前声卡信息，如果发现 ac107 相关声卡，说明已经正常加载驱动

无需额外设置音频通路，可直接用下面命令进行录音:

```
arecord -Dhw:ac107 -f S16_LE -r 16000 -c 2 /tmp/test.wav

另外可以通过下面命令调整增益
amixer -D hw:ac107 cset name='Channel 1 PGA Gain' 25
amixer -D hw:ac107 cset name='Channel 2 PGA Gain' 25
```

2.19.8 RX_SYNC 多声卡同步功能

RX_SYNC 功能用于同时使用到两个录音声卡 (不同音频硬件接口)，可以保证两个声卡同时开始录音，保证延迟稳定不变。

例如 R528 内部 3ADC+ 外部 ADC(使用 I2S) 组成 3+1 的语音方案，它就可以保证内部 ADC 和 I2S RX 的同步性。

R528 中，AudioCodec, I2S, DMIC 均可以使用 RX_SYNC 功能，除了它们对应的驱动配置外，还需要额外配置内核,dts 等地方。

2.19.8.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
           [*] Allwinner RX SYNC Support
```

2.19.8.2 dts 配置

在希望使用到 rx_sync 功能的音频模块上添加 rx_sync_en = <1> 字段:

```
codec:codec@2030000 {
    rx_sync_en  = <0x01>;
};

dmic:dmic@2031000 {
    rx_sync_en  = <0x01>;
};

daudio0:daudio@2032000 {
    rx_sync_en  = <0x01>;
};
```

注意，配置了 rx_sync_en 字段的模块, 需要都开启录音后，才会真正开始录音。

举例，4DMIC + 内置 3ADC 录音:

```
dts只需要将codec和dmic的rx_sync_en置一;

进入系统，确认DMIC和AudioCodec声卡都正确加载后，输入下面命令开始录音:
arecord -Dhw:snddmic -f S16_LE -r 16000 -c 4 /tmp/test1.wav &
arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 3 /tmp/test2.wav &

测试时可以发现，执行第一条arecord时，/tmp/test1.wav文件中并没有录音数据，
因为需要等第二个声卡也开始录音后，才会开始录制得到录音数据。
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
一般我们会使用multi插件将两个声卡数据合并(默认asound.conf中添加了CaptureMulti1可供参考)
arecord -DCaptureMulti1 -f S16_LE -r 16000 -c 7 /tmp/test.wav &
```

2.19.9 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.19.9.1 播放

```
通过LINEOUT->Speaker播放
amixer -Dhw:audiocodec cset name='LINEOUTL Output Select' 1
amixer -Dhw:audiocodec cset name='LINEOUTR Output Select' 1
amixer -Dhw:audiocodec cset name='LINEOUT Switch' 1
amixer -D hw:audiocodec cset name='LINEOUT volume' 15

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
或者利用默认/etc/asound.conf配置的pcm设备进行播放:
aplay -Ddefault /mnt/UDISK/1KHz_0dB_16000.wav

通过Headphone播放
amixer -D hw:audiocodec cset name='Headphone Switch' 1
amixer -D hw:audiocodec cset name='Headphone volume' 3

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

2.19.9.2 录音

```
通过内部的MIC1,MIC2,MIC3录制三通道
amixer -D hw:audiocodec cset name='MIC1 Input Select' 0
amixer -D hw:audiocodec cset name='MIC2 Input Select' 0
amixer -D hw:audiocodec cset name='MIC3 Input Select' 0
amixer -D hw:audiocodec cset name='ADC1 Input MIC1 Boost Switch' 1
amixer -D hw:audiocodec cset name='ADC2 Input MIC2 Boost Switch' 1
amixer -D hw:audiocodec cset name='ADC3 Input MIC3 Boost Switch' 1

amixer -D hw:audiocodec cset name='MIC1 gain volume' 19
amixer -D hw:audiocodec cset name='MIC2 gain volume' 19
amixer -D hw:audiocodec cset name='MIC3 gain volume' 19

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 3 /tmp/test.wav
```

2.20 V853 音频接口

V853 包含多个音频模块，分别是内置 AudioCodec,I2S0,I2S1,DMIC。

2.20.1 时钟源

V853 中，音频模块的时钟源来自 pll_audio0。

pll_audio0 可输出 22.5792M 和 24.576M 频率的时钟，分别支持 44.1k 系列、48k 系列的播放录音，但无法同时输出。

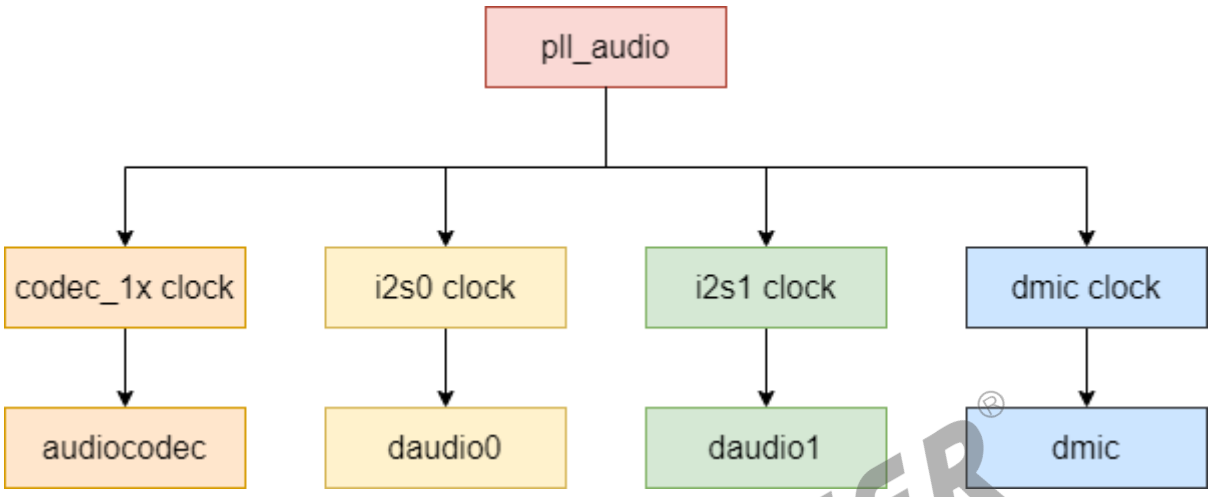


图 2-48: V853 时钟源

2.20.2 代码结构

```
linux-4.9/sound/soc/sunxi_v2/
├── snd_sun8iw21_codec.c          // codec驱动
├── snd_sun8iw21_codec.h
├── snd_sunxi_aaudio.c
├── snd_sunxi_common.c
├── snd_sunxi_common.h
├── snd_sunxi_daudio.c          //daudio platform驱动
├── snd_sunxi_daudio.h
├── snd_sunxi_dmic.c            // dmic platform驱动
├── snd_sunxi_dmic.h
├── snd_sunxi_log.h
├── snd_sunxi_mach.c
├── snd_sunxi_mach.h
├── snd_sunxi_mach_utils.c
├── snd_sunxi_mach_utils.h
├── snd_sunxi_pcm.c
├── snd_sunxi_pcm.h
├── snd_sunxi_rxsync.c
├── snd_sunxi_rxsync.h
├── snd_sunxi_txhub.c
└── snd_sunxi_txhub.h
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.20.3 AudioCodec

2.20.3.1 硬件特性

- 一路 DAC
  - 支持 16bit,20bit 有效采样精度
  - 支持 8KHz~192KHz 采样率
- 两路 ADC
  - 支持 16bit,20bit 有效采样精度
  - 支持 8KHz~48KHz 采样率
- 一路模拟输出：
  - 一路立体声输出 LINEOUT，支持 single/differ
- 两路模拟输入：MIC1,MIC2
  - MIC 支持 single/differ,
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 DRC
- 支持 mono 模式，最高支持 2 通道

2.20.3.2 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support V2  --->
         <*> Allwinner AAUDIO support
         <*> Allwinner DAUDIO Support
         <*> Allwinner Function Components
         <*>   Components Rx Sync
```

2.20.3.3 DTS 配置

2.20.3.3.1 DeviceTree 配置说明 设备树为芯片平台的模块配置，面对芯片特性进行配置，设备树文件的路径为：`lichee/linux-4.9/arch/arm/boot/dts/sun8iw21p1.dtsi`

```
codec:codec@0x02030000 {
            #sound-dai-cells = <0>;
            compatible  = "allwinner,sunxi-snd-codec";
            reg     = <0x0 0x02030000 0x0 0x34C>;
            clocks      = <&clk_pll_audio>,
```

```
                    <&clk_codec_dac>,
                    <&clk_codec_adc>;
            status = "disabled";
        };

        codec_plat:codec_plat {
            #sound-dai-cells = <0>;
            compatible  = "allwinner,sunxi-snd-aaudio";
            playback_cma    = <128>;
            capture_cma = <128>;
            tx_fifo_size    = <128>;
            rx_fifo_size    = <128>;
            dac_txdata  = <0x02030020>;
            adc_txdata  = <0x02030040>;
            status = "disabled";
        };

        codec_mach:codec_mach {
            compatible  = "allwinner,sunxi-snd-mach";
            soundcard-mach,name = "audiocodec";
            soundcard-mach,pin-switches = "MIC1", "MIC2", "LINEIN",
                              "LINEOUT", "SPK";
            soundcard-mach,routing = "MIC1_PIN",    "MIC1",
                         "MIC2_PIN",    "MIC2",
                         "LINEINL_PIN", "LINEIN",
                         "LINEINR_PIN", "LINEIN",
                         "LINEOUT", "LINEOUTL_PIN",
                         "SPK",     "LINEOUTL_PIN";
            status = "disabled";
            soundcard-mach,cpu {
                /* pll freq = 24.576M or 22.5792M * pll-fs */
                soundcard-mach,pll-fs = <1>;
                sound-dai = <&codec_plat>;
            };
            soundcard-mach,codec {
                sound-dai = <&codec>;
            };
        };
```

配置项说明（仅对常用项进行展开）：

AudioCodec 模块由 3 个设备树节点构建。

1. ASoC 层 codec: codec

表 2-128: AudioCodec codec 节点配置项 (linux4.9)

配置项名称	配置项说明
`#sound-dai-cells`	machine 层检测 codec 和 platform 节点的标志
reg	设置 audiocodec 寄存器起始地址和地址长度
clocks	设置 audiocodec 所需的时钟源和模块时钟

2. ASoC 层 platform: codec_plat

表 2-129: AudioCodec codec_plat 节点配置项 (linux4.9)

配置项名称	配置项说明
#sound-dai-cells	machine 层检测 codec 和 platform 节点的标志
playback_cma	设置播放流 DMA 申请的 size 大小，必须为 (2^n)Kbyte，默认 128
capture_cma	设置录音流 DMA 申请的 size 大小，必须为 (2^n)Kbyte，默认 128
tx_fifo_size	设置播放流 snd_pcm_runtime 的 fifo_size 大小，用于声卡硬件参数限定，默认 128
rx_fifo_size	设置录音流 snd_pcm_runtime 的 fifo_size 大小，用于声卡硬件参数限定，默认 128
dac_txdata	设置播放流 DMA 搬运地址（audiocodec 模块的 tx_fifo 寄存器地址）
adc_txdata	设置录音流 DMA 搬运地址（audiocodec 模块的 rx_fifo 寄存器地址）

3. ASoC 层 machine: codec_mach

表 2-130: AudioCodec codec_mach 节点配置项 (linux4.9)

配置项名称	配置项说明
soundcard-mach	machine 层配置前缀
name	声卡名字
pin-switches	用于定义模块接口开关（需参考驱动代码 dapm 进行设定）
routing	用于定义模块接口开关所链接的 dapm 通路（需参考驱动代码 dapm 进行设定）
cpu	machine 层所绑定的 cpu 节点（即 platform 层），用 sound-dai 属性指定节点
codec	machine 层所绑定的 codec 节点（即 codec 层），用 sound-dai 属性指定节点
pll-fs	指定模块时钟源频率（24.576M or 22.5792M * pll-fs）

2.20.3.3.2 board.dts 板级配置说明 board.dts 用于保存板级平台设备差异化的信息的补充，面对**板型特性**进行配置，其配置信息会覆盖 device tree 默认配置信息，board.dts 文件的路径为：`/device/config/chips/v853/configs/{BOARD}/board.dts`

```
&codec {
    /* external-avcc; */
    /* avcc-supply  = <&reg_aldo1>; */
    avcc-vol     = <1800000>;    /* uv */
    lineout_vol = <31>;
    mic1gain     = <31>;
    mic2gain     = <31>;
```

```
    adcdelaytime    = <0>;
    /* lineout-single; */
    /* mic1-single; */
    /* mic2-single; */
    pa_pin_max  = <1>;      /* set pa */
    pa_pin_0    = <&pio PH 4 1 1 1 1>;
    pa_pin_level_0  = <1>;
    pa_pin_msleep_0 = <0>;
    tx_hub_en;
    rx_sync_en;
    status      = "okay";
};

&codec_plat {
    status      = "okay";
};

&codec_mach {
    status      = "okay";
    soundcard-mach,cpu {
        sound-dai = <&codec_plat>;
    };
    soundcard-mach,codec {
        sound-dai = <&codec>;
    };
};
```

配置项介绍：

表 2-131：AudioCodec 模块板级配置项

配置项名称	配置值范围	配置项说明
status	“okay”, “disabled”	使能或关闭该节点驱动
avcc-external	注释为 false, 反之为 ture	avcc 电源是否为外围电路提供
avcc-supply	注释, 引用 pmu 提供的电源节点	avcc 若为外部 pmu 供电，可选择该项指定对应的 pmu 电源
avcc-vol	u32（缺省值: 1800000）	avcc 电压值设定，单位 uV，需符合实际硬件需求
dvcc-external	注释为 false, 反之为 ture	dvcc 电源是否为外围电路提供
dvcc-supply	注释, 引用 pmu 提供的电源节点	dvcc 若为外部 pmu 供电，可选择该项指定对应的 pmu 电源
dvcc-vol	u32（缺省值: 1800000）	dvcc 电压值设定，单位 uV，需符合实际硬件需求
adc-dig-vol-(n)	0~255	adc(n) 数字音量，(n) 代表 adc 序号，从 1 开始递增
mic(n)_vol	0~31	mic(n) 默认输入音量（增益），(n) 代表 mic 序号，从 1 开始递增
dac-dig-vol	0~63	dac 数字总音量

版权所有 © 珠海全志科技股份有限公司。保留一切权利

配置项名称	配置值范围	配置项说明
dac-dig-vol-(n)	0~255	dac(n) 数字音量，(n) 代表 dac 序号，从 1 开始递增
lineout-vol	0~31	lineout 默认输出音量（增益）
hp-vol	0~7	耳机默认输出音量（增益）
pa-pin-max	u32（正常为 1 或 2）	标定外部功放芯片使能引脚数量
pa-pin-(n)	pio 提供的引脚节点	指定第 (n) 个功放使能引脚
pa-pin-level-(n)	0~1	指定功放芯片使能电平
pa-pin-msleep-(n)	u32（正常小于 200）	设置功放芯片使能所需的 sleep 时长，用于规避 pop 声，单位 ms
adcdelaytime	u32（需符合 IC 规格）	设置 adc 录音延迟时长，单位 ms
tx-hub-en	注释为 false, 反之为 ture	选择是否注册 txhub 控件
rx-sync-en	注释为 false, 反之为 ture	选择是否注册 rxsync 控件

2.20.3.4 codec 数据通路

```
通过Lineout播歌
Playback --> DACL --> LINEOUTL Output Select --> LINEOUTL --> LINEOUT
Playback --> DACR --> LINEOUTR Output Select --> LINEOUTR --> LINEOUT

录音
MIC1 --> MIC1 Input Select --> ADC1 Input --> ADC1 --> Capture
MIC2 --> MIC2 Input Select --> ADC2 Input --> ADC2 --> Capture

LINE-in录音
LINEINL --> ADC1 Input --> ADC1 --> Capture
LINEINR --> ADC2 Input --> ADC2 --> Capture
```

V853 所有控件如下表：

表 2-132: amixer 控件表

控件名称	功能	数值
ADC1 ADC2 swap	将 adc1 和 adc2 进行通道交换	
ADC1 volume	ADC1 数字音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~ 71.24dB, 0.75dB/step, 默认 0xA0=0dB
ADC2 volume	ADC2 数字音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~ 71.24dB, 0.75dB/step, 默认 0xA0=0dB

控件名称	功能	数值
ADCDRC	开启 ADC DRC 功能	
ADCHPF	开启 adc hpf 功能	
DAC volume	DACL,DACR 音量设置	0~0xFF, 0:Mute; 1~0xFF:-119.25dB~ 71.24dB, 0.75dB/step, 默认 0xA0=0dB
DACDRC	开启 dac drc 功能	
DACHPF	开启 dac hpf 功能	
LINEIN Switch	是否使能 ADC->LINEIN 的通路	0: 关闭; 1: 使能
LINEINL gain volume	LINEINL 增益	0:0dB; 1:6dB
LINEINR gain volume	LINEINR 增益	0:0dB; 1:6dB
LINEOUT Output Select	Lineout 输出选择	0: 单端; 1: 差分
LINEOUT Switch	是否使能 Lineout 通路	0: 关闭; 1: 使能
LINEOUT volume	Lineout 音量设置	0~31, 表示-43.5~0dB
MIC1 Input Select	MIC1 输入模式	0: 差分输入; 1: 单端输入
MIC1 Switch	是否使能 ADC1->MIC1 的通路	0: 关闭; 1: 使能
MIC1 gain volume	MIC1 增益	0~31, 表示0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
MIC2 Input Select	MIC2 输入模式	0: 差分输入; 1: 单端输入
MIC2 Switch	是否使能 ADC2->MIC2 的通路	0: 关闭; 1: 使能
MIC2 gain volume	MIC2 增益	0~31, 表示0~36dB, 0:0dB, 1~3:6dB, 4~31:9~36dB, 1dB/step
SPK Switch	是否使能 Speaker 通路 (使用功放)	0: 关闭; 1: 使能
digital volume	数字端音量设置	0~63, 表示-73.08~0dB
rx sync mode	使能同步录音（和其它开启 rx sync mode 的声卡）	
tx hub mode	使能同源播放（和其它开启 tx hub mode 的声卡）	

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.20.4 Daudio

2.20.4.1 硬件特性

- 两路 I2S/PCM, 可用于蓝牙通话，语音采集，数字功放；
- 支持主从模式
- 支持 Left-justified,Right-justified,Standar mode I2S,PCM mode
- 支持 i2s,pcm 协议格式配置
- 支持同时 playback 和 record(全双工模式)
- 支持 8~192KHz 采样率
- 支持 16,24,32bit 采样精度
- 支持 2 路 MCLK 输出

2.20.4.2 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support V2  --->
           <*> Allwinner AAUDIO support
           <*> Allwinner DAUDIO Support
           <*> Allwinner Function Components
           <*>   Components Rx Sync
```

2.20.4.3 DTS 配置

2.20.4.3.1 DeviceTree 配置说明 设备树为芯片平台的模块配置，面对芯片特性进行配置，设备树文件的路径为：`lichee/linux-4.9/arch/arm/boot/dts/sun8iw21p1.dtsi`

```
daudio0_plat:daudio0_plat@0x02032000 {
    #sound-dai-cells = <0>;
    compatible  = "allwinner,sunxi-snd-plat-daudio";
    reg     = <0x0 0x02032000 0x0 0x7c>;
    clocks      = <&clk_pll_audio>, <&clk_i2s0>;
    playback_cma    = <128>;
    capture_cma = <128>;
    tx_fifo_size    = <128>;
    rx_fifo_size    = <128>;
    status = "disabled";
};

daudio0_mach:daudio0_mach{
    compatible  = "allwinner,sunxi-snd-mach";
    soundcard-mach,format = "i2s";
    soundcard-mach,name = "snddaudio0";
```

```
    status = "disabled";
    soundcard-mach,cpu {
        sound-dai = <&daudio0_plat>;
    };
    soundcard-mach,codec {
    };
};

daudio1_plat:daudio1_plat@0x02033000 {
    #sound-dai-cells = <0>;
    compatible  = "allwinner,sunxi-snd-plat-daudio";
    reg     = <0x0 0x02033000 0x0 0x7c>;
    clocks      = <&clk_pll_audio>, <&clk_i2s1>;
    playback_cma    = <128>;
    capture_cma = <128>;
    tx_fifo_size    = <128>;
    rx_fifo_size    = <128>;
    status = "disabled";
};

daudio1_mach:daudio1_mach{
    compatible  = "allwinner,sunxi-snd-mach";
    soundcard-mach,format = "i2s";
    soundcard-mach,name = "snddaudio1";
    status = "disabled";
    soundcard-mach,cpu {
        sound-dai = <&daudio1_plat>;
    };
    soundcard-mach,codec {
    };
};
```

配置项说明（仅对常用项进行展开）：

I2S/PCM 模块由 2 个或 3 个设备树节点构建。

1. ASoC 层 codec: 非必须节点，若无，则绑定虚拟 codec 节点。
2. ASoC 层 platform: daudio(n)_plat

表 2-133: I2S/PCM daudio(n)_plat 节点配置项 (linux4.9)

配置项名称	配置项说明
#sound-dai-cells	machine 层检测 codec 和 platform 节点的标志
reg	设置 I2S/PCM 寄存器起始地址和地址长度
clocks	设置 I2S/PCM 所需的时钟源和模块时钟
playback_cma	设置播放流 DMA 申请的 size 大小，必须为 (2^n)Kbyte，默认 128
capture_cma	设置录音流 DMA 申请的 size 大小，必须为 (2^n)Kbyte，默认 128
tx_fifo_size	设置播放流 snd_pcm_runtime 的 fifo_size 大小，用于声卡硬件参数限定，默认 128
rx_fifo_size	设置录音流 snd_pcm_runtime 的 fifo_size 大小，用于声卡硬件参数限定，默认 128

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3. ASoC 层 machine: daudio(n)_mach

表 2-134: I2S/PCM daudio(n)_mach 节点配置项 (linux4.9)

配置项名称	配置项说明
soundcard-mach	machine 层配置前缀
name	声卡名字
cpu	machine 层所绑定的 cpu 节点（即 platform 层），用 sound-dai 属性指定节点
codec	machine 层所绑定的 codec 节点（即 codec 层），用 sound-dai 属性指定节点。若该子节点下无 sound-dai 属性，即代表使用虚拟 codec，用于辅助生成声卡
pll-fs	指定模块时钟源频率（24.576M or 22.5792M * pll-fs）

说明

- ***daudio(n)_plat* 代表 *daudio0_plat, daudio1_plat,* ···（取决于芯片规格）；**
- ***daudio(n)_mach* 代表 *daudio0_mach, daudio1_mach,* ···（取决于芯片规格）；**

2.20.4.3.2 board.dts 板级配置说明 board.dts 用于保存板级平台设备差异化的信息的补充，面对**板型特性**进行配置，其配置信息会覆盖 device tree 默认配置信息，board.dts 文件的路径为：`/device/config/chips/v853/configs/{BOARD}/board.dts`

```
&daudio0_plat {
    tdm_num     = <0>;
    tx_pin      = <0>;
    rx_pin      = <0>;
    /* pinctrl_used; */
    /* pinctrl-names= "default","sleep"; */
    /* pinctrl-0    = <&daudio0_pins_a>; */
    /* pinctrl-1    = <&daudio0_pins_b>; */
    tx_hub_en;
    rx_sync_en;
    status      = "okay";
};

&daudio0_mach {
    soundcard-mach,format       = "i2s";
    soundcard-mach,frame-master = <&daudio0_cpu>;
    soundcard-mach,bitclock-master  = <&daudio0_cpu>;
    /* soundcard-mach,frame-inversion; */
    /* soundcard-mach,bitclock-inversion; */
    soundcard-mach,slot-num     = <2>;
    soundcard-mach,slot-width   = <32>;
    status      = "okay";
    daudio0_cpu: soundcard-mach,cpu {
        sound-dai = <&daudio0_plat>;
        soundcard-mach,pll-fs   = <1>; /* pll freq = 24.576M or 22.5792M * pll-fs */
        soundcard-mach,mclk-fs  = <0>; /* mclk freq = pcm rate * mclk-fs */
    };
    daudio0_codec: soundcard-mach,codec {
```

```
    };
};

&daudio1_plat {
    tdm_num     = <1>;
    tx_pin      = <0>;
    rx_pin      = <0>;
    /* pinctrl_used; */
    /* pinctrl-names= "default","sleep"; */
    /* pinctrl-0    = <&daudio1_pins_a>; */
    /* pinctrl-1    = <&daudio1_pins_b>; */
    tx_hub_en;
    rx_sync_en;
    status      = "disabled";
};

&daudio1_mach {
    soundcard-mach,format       = "i2s";
    soundcard-mach,frame-master = <&daudio1_cpu>;
    soundcard-mach,bitclock-master  = <&daudio1_cpu>;
    /* soundcard-mach,frame-inversion; */
    /* soundcard-mach,bitclock-inversion; */
    soundcard-mach,slot-num     = <2>;
    soundcard-mach,slot-width   = <32>;
    status      = "disabled";
    daudio1_cpu: soundcard-mach,cpu {
        sound-dai = <&daudio1_plat>;
        soundcard-mach,pll-fs   = <1>; /* pll freq = 24.576M or 22.5792M * pll-fs */
        soundcard-mach,mclk-fs  = <0>; /* mclk freq = pcm rate * mclk-fs */
    };
    daudio1_codec: soundcard-mach,codec {
    };
};
```

配置项介绍：

表 2-135: I2S/PCM 模块板级配置项

配置项名称	配置值范围	配置项说明
status	“okay”, “disabled”	使能或关闭该节点驱动
tdm-num	0~3	指定 I2S 序号，需和 daudio(n)_plat 的 (n) 对应
tx-pin	0~3	指定 I2S 所使用的 DOUT 引脚序号
rx-pin	0~3	指定 I2S 所使用的 DIN 引脚序号
tx-hub-en	注释为 false, 反之为 ture	选择是否注册 txhub 控件
rx-sync-en	注释为 false, 反之为 ture	选择是否注册 rxsync 控件
format	“i2s”,“right_j”,“left_j”, “dsp_a”,“dsp_b”	选择 tdm 协议格式
frame-master	cpu 子节点，codec 子节点	选择 LRCK 信号主模式

配置项名称	配置值范围	配置项说明
bitclock-master	cpu 子节点，codec 子节点	选择 BCLK 信号主模式
frame-inversion	注释为 false, 反之为 ture	LRCK 信号是否翻转
bitclock-inversion	注释为 false, 反之为 ture	BCLK 信号是否翻转
slot-num	1~16	slot 数量（可简单理解为支持最大通道数）
slot-width	8, 16, 24, 32	单个 slot 宽度（可简单理解为支持最大数据精度）
mclk-fp	注释为 false, 反之为 ture	ture: mclk 以固定频段输出；false: mclk 以采样率倍数输出
mclk-fs	u32	固定频段：`mclk = mclk-fs * 12.288M or 11.2896M` 采样率倍数：`mclk = mclk-fs * pcm rate`

2.20.5 DMIC

硬件特性

- 支持 4 路输入
- 支持 8~48KHz 采样率
- 支持 16/24bit 采样精度

2.20.5.1 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support V2  --->
         <*> Allwinner DMIC support
```

2.20.5.2 DTS 配置

2.20.5.2.1 DeviceTree 配置说明 设备树为芯片平台的模块配置，面对芯片特性进行配置，设备树文件的路径为：`lichee/linux-4.9/arch/arm/boot/dts/sun8iw21p1.dtsi`

```
dmic_plat:dmic_plat@0x02031000 {
    #sound-dai-cells = <0>;
    compatible  = "allwinner,sunxi-snd-plat-dmic";
```

```
    reg     = <0x0 0x02031000 0x0 0x50>;
    clocks      = <&clk_pll_audio>, <&clk_dmic>;
    capture_cma = <128>;
    rx_fifo_size    = <128>;
    status = "disabled";
};
dmic_mach:dmic_mach{
    compatible  = "allwinner,sunxi-snd-mach";
    soundcard-mach,name = "snddmic";
    soundcard-mach,capture_only;
    status = "disabled";
    soundcard-mach,cpu {
        sound-dai = <&dmic_plat>;
    };
    soundcard-mach,codec {
    };
};
```

配置项说明（仅对常用项进行展开）：

DMIC 模块由 2 个设备树节点构建。

1. ASoC 层 codec: 无，绑定虚拟 codec 节点。
2. ASoC 层 platform: dmic_plat

表 2-136: DMIC dmic_plat 节点配置项 (linux4.9)

配置项名称	配置项说明
`#sound-dai-cells`	machine 层检测 codec 和 platform 节点的标志
reg	设置 DMIC 寄存器起始地址和地址长度
clocks	设置 DMIC 所需的时钟源和模块时钟
capture_cma	设置录音流 DMA 申请的 size 大小，必须为 (2^n)Kbyte，默认 128
rx_fifo_size	设置录音流 snd_pcm_runtime 的 fifo_size 大小，用于声卡硬件参数限定，默认 128

3. ASoC 层 machine: dmic_mach

表 2-137: DMIC dmic_mach 节点配置项 (linux4.9)

配置项名称	配置项说明
soundcard-mach	machine 层配置前缀
name	声卡名字
cpu	machine 层所绑定的 cpu 节点（即 platform 层），用 sound-dai 属性指定节点
codec	machine 层所绑定的 codec 节点（即 codec 层），用 sound-dai 属性指定节点（使用虚拟 codec）

版权所有 © 珠海全志科技股份有限公司。保留一切权利

配置项名称	配置项说明
capture_only	设置仅录音，不进行播放流设备创建
pll-fs	指定模块时钟源频率（24.576M or 22.5792M * pll-fs）

2.20.5.2.2 board.dts 板级配置说明 board.dts 用于保存板级平台设备差异化的信息的补充，面对**板型特性**进行配置，其配置信息会覆盖 device tree 默认配置信息，board.dts 文件的路径为：`/device/config/chips/v853/configs/{BOARD}/board.dts`

```
&dmic_plat {
    rx_chmap    = <0x76543210>;
    data_vol    = <0xB0>;
    rxdelaytime = <0>;
    pinctrl_used;
    pinctrl-names   = "default","sleep";
    pinctrl-0   = <&dmic_pins_a>;
    pinctrl-1   = <&dmic_pins_b>;
    rx_sync_en;
    status      = "disabled";
};

&dmic_mach {
    status      = "disabled";
    soundcard-mach,cpu {
        sound-dai = <&dmic_plat>;
        soundcard-mach,pll-fs   = <1>; /* pll freq = 24.576M or 22.5792M * pll-fs */
    };
    soundcard-mach,codec {
    };
};
```

配置项介绍：

表 2-138: DMIC 模块板级配置项

配置项名称	配置值范围	配置项说明
status	“okay”, “disabled”	使能或关闭该节点驱动
rx-sync-en	注释为 false, 反之为 ture	选择是否注册 rxsync 控件
avcc-supply	注释, 引用 pmu 提供的电源节点	avcc 若为外部 pmu 供电，可选择该项指定对应的 pmu 电源

2.20.6 RX_SYNC 多声卡同步功能

RX_SYNC 功能用于同时使用到两个录音声卡 (不同音频硬件接口)，可以保证两个声卡同时开始录音，保证延迟稳定不变。

例如内部 ADC+ 外部 ADC(使用 I2S) 的语音方案，它就可以保证内部 ADC 和 I2S RX 的同步性。

AudioCodec, I2S, DMIC 均可以使用 RX_SYNC 功能，除了它们对应的驱动配置外，还需要额外配置内核,dts 等地方。

2.20.6.1 内核配置

```
  Device Drivers  --->
  <*> Sound card support  --->
     <*> Advanced Linux Sound Architecture  --->
        <*> ALSA for SoC audio support  --->
           Allwinner SoC Audio support V2  --->
           <*> Allwinner AAUDIO support
           <*> Allwinner DAUDIO Support
           <*> Allwinner Function Components
           <*>   Components Rx Sync
```

2.20.6.2 dts 配置

在希望使用到 rx_sync 功能的音频模块上添加 rx_sync_en = <1> 字段:

```
codec:codec@2030000 {
    rx_sync_en  = <0x01>;
};

dmic:dmic@2031000 {
    rx_sync_en  = <0x01>;
};

daudio0:daudio@2032000 {
    rx_sync_en  = <0x01>;
};
```

注意，配置了 rx_sync_en 字段的模块，需要都开启录音后，才会真正开始录音。

一般我们会使用 multi 插件将两个声卡数据合并 (默认 asound.conf 中添加了 CaptureMulti1 可供参考)，`arecord -DCaptureMulti1 -f S16_LE -r 16000 -c 7 /tmp/test.wav &`。

2.20.7 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.20.7.1 播放

```
通过LINEOUT->Speaker播放
amixer -Dhw:audiocodec cset name='LINEOUTL Output Select' 1
amixer -Dhw:audiocodec cset name='LINEOUTR Output Select' 1
amixer -Dhw:audiocodec cset name='LINEOUT Switch' 1
amixer -D hw:audiocodec cset name='LINEOUT volume' 15

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
或者利用默认/etc/asound.conf配置的pcm设备进行播放:
aplay -Ddefault /mnt/UDISK/1KHz_0dB_16000.wav

通过Headphone播放
amixer -D hw:audiocodec cset name='Headphone Switch' 1
amixer -D hw:audiocodec cset name='Headphone volume' 3

aplay -Dhw:audiocodec /mnt/UDISK/1KHz_0dB_16000.wav
```

2.20.7.2 录音

```
通过内部的MIC1,MIC2录制双通道
amixer -D hw:audiocodec cset name='MIC1 Input Select' 0
amixer -D hw:audiocodec cset name='MIC2 Input Select' 0
amixer -D hw:audiocodec cset name='MIC1 Switch' 1
amixer -D hw:audiocodec cset name='MIC2 Switch' 1

amixer -D hw:audiocodec cset name='MIC1 gain volume' 19
amixer -D hw:audiocodec cset name='MIC2 gain volume' 19

arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 /tmp/test.wav
```

2.21 F133 音频接口

F133 包含多个音频模块，分别是内置 AudioCodec,I2S1,I2S2,DMIC,SPDIF。

2.21.1 时钟源

F133 中，音频模块的时钟源来自 pll_audio0 以及 pll_audio1_div5。

pll_audio0 可以输出 22.5792M 的时钟, 而 pll_audio1_div5 输出 24.576M 的时钟，分别支持 44.1k 系列，48k 系列的播放录音。

2.21.2 AudioCodec

2.21.2.1 硬件特性

- 两路 DAC
  - 支持 16bit,20bit 有效采样精度
  - 支持 8KHz~192KHz 采样率
- 三路 ADC
  - 支持 16bit,20bit 有效采样精度
  - 支持 8KHz~48KHz 采样率
- 模拟输出：
  - 一路立体声输出 HPOUTL,HPOUTR
- 模拟差分输入：
  - 一路差分麦克风输入 MIC3P/N
  - 一路立体声 line-in 输入 LINEINL,LINEINR
  - 一路立体声 FM-in 输入 FMINL,FMINR
- 支持同时 playback 和 record(全双工模式)
- DAC 及 ADC 均支持 DRC

2.21.2.2 内核配置

kernel_menuconfig 配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
          <*> Allwinner Sun20iw1 Codec Support
          <*> Allwinner Audio Simple Card
          [ ] Allwinner RX SYNC Support
```

配置项名称	配置项说明
Allwinner Sun20iw1 Codec Support	audiocodec
Allwinner Audio Simple Card	绑定声卡
Allwinner RX SYNC Support	同步功能组件

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.21.2.3 DTS 配置

board.dts 板级配置

```
&codec {
    /* MIC and headphone gain setting */
    mic1gain    = <0x1F>;
    mic2gain    = <0x1F>;
    mic3gain    = <0x1F>;
    /* ADC/DAC DRC/HPF func enabled */
    ¦   /* 0x1:DAP_HP_EN; 0x2:DAP_SPK_EN; 0x3:DAP_HPSPK_EN */
    adcdrc_cfg  = <0x0>;
    adchpf_cfg  = <0x1>;
    dacdrc_cfg  = <0x0>;
    dachpf_cfg  = <0x0>;
    /* Volume about */
    digital_vol     = <0x00>;
    lineout_vol     = <0x1a>;
    headphonegain   = <0x03>;
    /* Pa enabled about */
    pa_level    = <0x01>;
    pa_pwr_level    = <0x01>;
    pa_msleep_time  = <0x78>;
    /* gpio-spk = <&pio PF 2 GPIO_ACTIVE_HIGH>;*/
    /* gpio-spk-pwr = <&pio PF 4 GPIO_ACTIVE_HIGH>; */
    /* CMA config about */
    playback_cma    = <128>;
    capture_cma = <256>;
    /* regulator about */
    /* avcc-supply  = <&reg_aldo1>; */
    /* hpvcc-supply = <&reg_eldo1>; */
    status = "okay";
};
```

部分配置项名称	配置取值	配置项说明
mic3gain	0-31	mic3 增益
digital_vol	0-63	DAC 数字音量
headphonegain	0-7	耳机播放增益
pa_level	0-1	功放芯片使能电平
pa_pwr_level	0-1	功放供电使能电平
pa_msleep_time	u32, 一般小于 200	设置功放芯片使能所需 sleep 时间
status	“okay”/“disable”	使能或者关闭本节点

2.21.3 Daudio

2.21.3.1 硬件特性

- 两路 I2S/PCM I2S1 I2S2

版权所有 © 珠海全志科技股份有限公司。保留一切权利

- 支持 8-192k 采样率
- 支持 16 24 32 采样精度
- 支持 1-16 多通道录音播放
- 支持 5 种 TDM 模式
  - I2S standard mode
  - Left-justified mode
  - Right-justified mode
  - DSP-A mode(short frame PCM mode)
  - DSP-B mode(long frame PCM mode)
- 支持回环模式
- 支持同时 playback 和 capture
- 支持多声卡同步录音

2.21.3.2 内核配置

kernel_menuconfig 配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
          <*> Allwinner Audio Simple Card
          <*>   Allwinner Digital Audio Support
          [ ] Allwinner RX SYNC Support
```

配置项名称	配置项说明
Allwinner Sun20iw1 Codec Support	I2S/PCM
Allwinner Audio Simple Card	绑定声卡
Allwinner RX SYNC Support	同步功能组件

2.21.3.3 DTS 配置

board.dts 板级配置

```
&daudio1 {
    mclk_div     = <0x01>;
    frametype    = <0x00>;
    tdm_config   = <0x01>;
    sign_extend      = <0x00>;
    msb_lsb_first    = <0x00>;
    pcm_lrck_period = <0x80>;
    slot_width_select = <0x20>;
    pinctrl-names    = "default", "sleep";
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    pinctrl-0       = <&daudio1_pins_a>;
    pinctrl-1       = <&daudio1_pins_b>;
    pinctrl_used    = <0x0>;
    status = "disabled";
};

&sounddaudio1 {
    status = "disabled";
    daudio1_master: simple-audio-card,codec {
        /* sound-dai = <&ac108>; */
    };
};
```

设备树配置 sun20iw1p1.dtsi

```
daudio1:daudio@2033000 {
    #sound-dai-cells = <0>;
    compatible = "allwinner,sunxi-daudio";
    reg = <0x0 0x02033000 0x0 0xa0>;
    clocks = <&ccu CLK_PLL_AUDIO0>,
        ¦<&ccu CLK_I2S1>,
        ¦<&ccu CLK_BUS_I2S1>;
    clock-names = "pll_audio", "i2s1", "i2s1_bus";
    resets = <&ccu RST_BUS_I2S1>;
    dmas        = <&dma 4>, <&dma 4>;
    dma-names   = "tx", "rx";
    interrupts-extended = <&plic0 43 IRQ_TYPE_LEVEL_HIGH>;
    sign_extend     = <0x00>;
    tx_data_mode        = <0x00>;
    rx_data_mode        = <0x00>;
    msb_lsb_first       = <0x00>;
    pcm_lrck_period     = <0x80>;
    slot_width_select   = <0x20>;
    frametype       = <0x00>;
    tdm_config      = <0x01>;
    tdm_num         = <0x01>;
    mclk_div        = <0x00>;
    clk_parent      = <0x01>;
    capture_cma     = <128>;
    playback_cma        = <128>;
    tx_num          = <4>;
    tx_chmap1       = <0x76543210>;
    tx_chmap0       = <0xFEDCBA98>;
    rx_num          = <4>;
    rx_chmap3       = <0x03020100>;
    rx_chmap2       = <0x07060504>;
    rx_chmap1       = <0x0B0A0908>;
    rx_chmap0       = <0x0F0E0D0C>;
    asrc_function_en    = <0x00>;
    rx_sync_en              = <0x00>;
    device_type = "daudio1";
    status = "disabled";
};

sounddaudio1: sounddaudio1@20330a0 {
    reg = <0x0 0x020330a0 0x0 0x4>;
    compatible = "sunxi,simple-audio-card";
    simple-audio-card,name = "snddaudio1";
    simple-audio-card,format = "i2s";
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    status = "disabled";
    simple-audio-card,cpu {
        sound-dai = <&daudio1>;
    };
};
```

部分配置项名称	配置取值	配置项说明
status	“okay” “disable”	使能或者关闭 snddaudio 驱动
mclk_div	0-192	mclk 分频系数，取值为 0/1/2/4/8/12/16/24/32/48/64/96/128/176/192
frametype	0-1	0:short frame(1 clock width) 1:long frame(2 clock width)
tdm_config	0-1	0:PCM mode 1:I2S mode
sign_extend	0-1	0:zero pending 1:sign extend
msb_lsb_first	0-1	0:msb first 1:lsb first
pcm_lrck_period	16/32/64/128/256	一般可配置为 16/32/64/128/256 个 bclk
slot_width_select	8/16/32	slot 支持 8/16/32bit 宽度
tx_data_mode	0/1/2/3	0:16bit linear PCM 1:reserved 2:8bit u-law 3:8bit a-law
rx_data_mode	0/1/2/3	0:16bit linear PCM 1:reserved 2:8bit u-law 3:8bit a-law
simple-audio-card,name	string	声卡名称
simple-audio-card,format	string	I2S tdm 模式, 取值为 “i2s”/“right_j”/“left_j”/“dsp_a”/“dsp_b”
simple-audio-card,frame-master	注释为 false, 反之为 true	配置 frame clk 主从关系，不配置则 soc 为主，反之 soc 为从
simple-audio-card,bitclock-master	注释为 false, 反之为 true	配置 bit clk 主从关系，不配置则 soc 为主，反之 soc 为从
simple-audio-card,frame-inversion	注释为 false, 反之为 true	配置 frame clk 极性取反; 不配置则是正常极性
simple-audio-card,bitclock-inversion	注释为 false, 反之为 true	配置 bit clk 极性取反; 不配置则是正常极性
simple-audio-card,capture_only	注释为 false, 反之为 true	仅支持录音
simple-audio-card,playback_only	注释为 false, 反之为 true	仅支持播放

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2.21.4 DMIC

2.21.4.1 硬件特性

- 支持 8-48k 采样率
- 支持 16 24 采样精度
- 支持 1-8 多通道录音播放
- 支持多声卡同步录音

2.21.4.2 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
          <*> Allwinner Audio Simple Card
          <*>   Allwinner DMIC Support
          [ ] Allwinner RX SYNC Support
```

配置项名称	配置项说明
Allwinner Sun20iw1 Codec Support	dmic
Allwinner Audio Simple Card	绑定声卡
Allwinner RX SYNC Support	同步功能组件

2.21.4.3 DTS 配置

```
&dmic {
    pinctrl-names   = "default","sleep";
    pinctrl-0       = <&dmic_pins_a>;
    pinctrl-1       = <&dmic_pins_b>;
    rx_sync_en      = <0x00>;
    status = "okay";
};

&dmic_codec {
    status = "okay";
};

&sounddmic {
    status = "okay";
};
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

部分配置项名称	配置取值	配置项说明
status	“okay”/“disable”	使能或者关闭本节点
rx_sync_en	注释为 false 反之为 true	是否注册 rx_sync 组件

2.21.5 SPDIF

2.21.5.1 硬件特性

- 支持 22.05-192k 采样率
- 支持 16 24 采样精度
- 支持 1-2 多通道录音播放
- 支持多声卡同步录音
- 支持回环模式
- 支持 IEC-60958 协议
- 支持 IEC-61937 协议

2.21.5.2 内核配置

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      <*> ALSA for SoC audio support  --->
         Allwinner SoC Audio support  --->
          <*> Allwinner Audio Simple Card
          <*>   Allwinner SPDIF Support
          [ ] Allwinner RX SYNC Support
```

配置项名称	配置项说明
Allwinner Sun20iw1 Codec Support	spdif
Allwinner Audio Simple Card	绑定声卡
Allwinner RX SYNC Support	同步功能组件

2.21.5.3 DTS 配置

```
&spdif {
    pinctrl-names   = "default","sleep";
    pinctrl-0       = <&spdif_pins_a>;
    pinctrl-1       = <&spdif_pins_b>;
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
    rx_sync_en      = <0x00>;
    status = "okay";
};

&soundspdif {
    status = "okay";
};
```

部分配置项名称	配置取值	配置项说明
status	“okay” “disable”	使能或者关闭本节点
rx_sync_en	注释为 false 反之为 true	是否注册 rx_sync 组件

2.21.6 标案音频测试方法

该章节主要介绍在标案上进行播歌，录音的测试命令

2.21.6.1 播放

```
通过Headphone播放
1.开机后推送测试音频48000.wav到小机端,pc命令: adb push 48000.wav /tmp/
2.播放该音频文件, aplay /tmp/48000.wav
```

2.21.6.2 录音

```
mic:
1.使能mic3通路: amixer set "ADC3 Input MIC3 Boost" on
2.设置mic3增益: amixer cset name="MIC3 gain volume" 31
3.录音5s: arecord -D hw:audiocodec -f S16_LE -r 44100 -c 1 /tmp/test.wav --period-size 1024
    --buffer-size 4096 -d 5

linein:
1.使能linein通路: amixer set "ADC1 Input LINEINL" on;amixer set "ADC2 Input LINEINR" on
2.设置linein增益: amixer cset name="LINEINL gain volume" 1;amixer cset name="LINEINR gain
    volume" 1
3.录音5s: arecord -D hw:audiocodec -f S16_LE -r 44100 -c 2 /tmp/test.wav --period-size 1024
    --buffer-size 4096 -d 5

fmin:
1.使能fmin通路: amixer set "ADC1 Input FMINL" on;amixer set "ADC2 Input FMINR" on
2.设置fmin增益: amixer cset name="FMINL gain volume" 1;amixer cset name="FMINR gain volume" 1
3.录音5s: arecord -D hw:audiocodec -f S16_LE -r 44100 -c 2 /tmp/test.wav --period-size 1024
    --buffer-size 4096 -d 5

I2S:
1.推送音频文件到小机端:adb push 16000.wav /tmp/16000.wav
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
2.配置I2S Loopback功能:amixer  -D hw:snddaudio2 cset name='sunxi daudio loopback debug' 1
3.开始录音: arecord -D hw:snddaudio2 -f S16_LE -r 44100 -c 2 /tmp/test_su.wav --period-size
    1024 --buffer-size 4096 &
4.开始播放: aplay -D hw:snddaudio2 /tmp/16000.wav
5.aplay播放结束后，killall arecord结束录音任务
6.通过adb命令把test_su.wav拉出来，在PC端查看音频数据是否跟播放的内容一致
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3 3 常用工具及调试方法

3.1 3.1 alsa-utils

标准 ALSA 工具, 它使用到 alsa-lib 标准库，一般常用到的有 amixer,aplay,arecord 等。

3.1.1 3.1.1 amixer

amixer 是命令行的 ALSA 声卡驱动调节器工具，用于设置 mixer control。

使用方法:

- 常用选项

选项	功能
`-D,--device`	指定声卡设备, 默认使用 default

- 常用命令

命令	功能
controls	列出指定声卡的所有控件
contents	列出指定声卡的所有控件的具体信息
cget	获取指定控件的信息
cset	设定指定控件的值

举例:

```
获取audiocodec声卡的所有控件名
amixer -Dhw:audiocodec controls

获取当前硬件音量
amixer -Dhw:audiocodec cget name='LINEOUT volume'

设置当前硬件音量
amixer -Dhw:audiocodec cget name='LINEOUT volume' 25
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

3.1.2 3.1.2 aplay

aplay 是命令行的 ALSA 声卡驱动的播放工具，用于播放功能。

使用方法：

选项	功能
-D,--device	指定声卡设备, 默认使用 default
-l,--list-devices	列出当前所有声卡
-t,--file-type	指定播放文件的格式, 如 voc,wav,raw, 不指定的情况下会去读取文件头部作识别
-c,--channels	指定通道数
-f,--format	指定采样格式
-r,--rate	采样率
-d,--duration	指定播放的时间
--period-size	指定 period size
--buffer-size	指定 buffer size

如果播放的是 wav 文件，可以解析头部，识别通道数，采样率等参数。

举例：

```
aplay -Dhw:audiocodec /mnt/UDISK/test.wav
```

3.1.3 3.1.3 arecord

arecord 是命令行的 ALSA 声卡驱动的录音工具，用于录音功能。

使用方法：

选项	功能
-D,--device	指定声卡设备, 默认使用 default
-l,--list-devices	列出当前所有声卡
-t,--file-type	指定播放文件的格式, 如 voc,wav,raw, 不指定的情况下会去读取文件头部作识别
-c,--channels	指定通道数
-f,--format	指定采样格式
-r,--rate	采样率
-d,--duration	指定播放的时间
--period-size	指定 period size
--buffer-size	指定 buffer size

版权所有 © 珠海全志科技股份有限公司。保留一切权利

举例：

```
    录制5s,通道数为2, 采样率为16000, 采样精度为16bit, 保存为wav文件
    arecord -Dhw:audiocodec -f S16_LE -r 16000 -c 2 -d 5 /mnt/UDISK/test.wav
```

3.1.4 3.1.4 alsaconf

alsaconf 指的是 ALSA configuration file，使用 alsa-lib 打开声卡，操作 pcm, mixer 时，会加载相关位置上的配置文件，用于指导操作 pcm,mixer 设备。

首先会读取配置文件/usr/share/alsa/alsa.conf，其中有下面一段 hooks。

```
@hooks [
    {
        func load
        files [
            {
                @func concat
                strings [
                    { @func datadir }
                    "/alsa.conf.d/"
                ]
            }
            "/etc/asound.conf"
            "~/.asoundrc"
        ]
        errors false
    }
]
```

这里设定了一个钩子，去读取相关目录配置文件：

```
/usr/share/alsa/alsa.conf.d/
/etc/asound.conf
~/.asoundrc
```

这些配置文件可以设定 defaut 声卡，自定义 pcm 设备,alsa 插件等功能，具体可以参考:

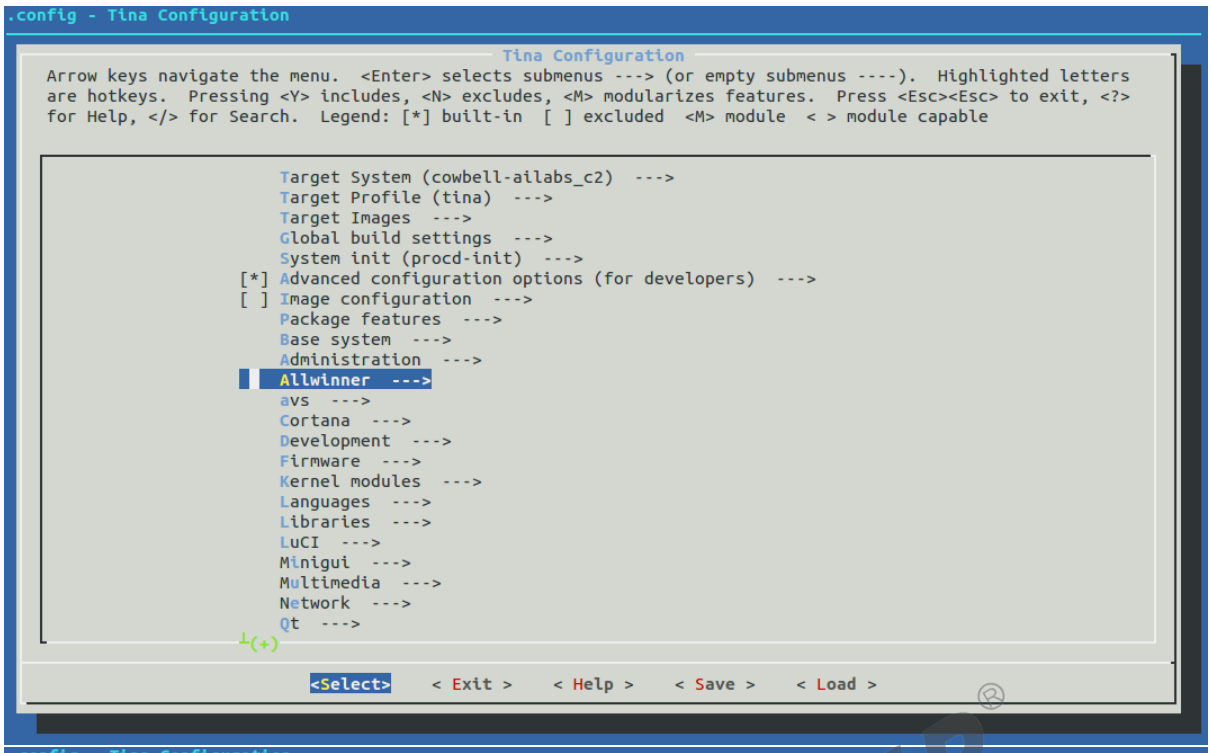
https://www.alsa-project.org/alsa-doc/alsa-lib/conf.html

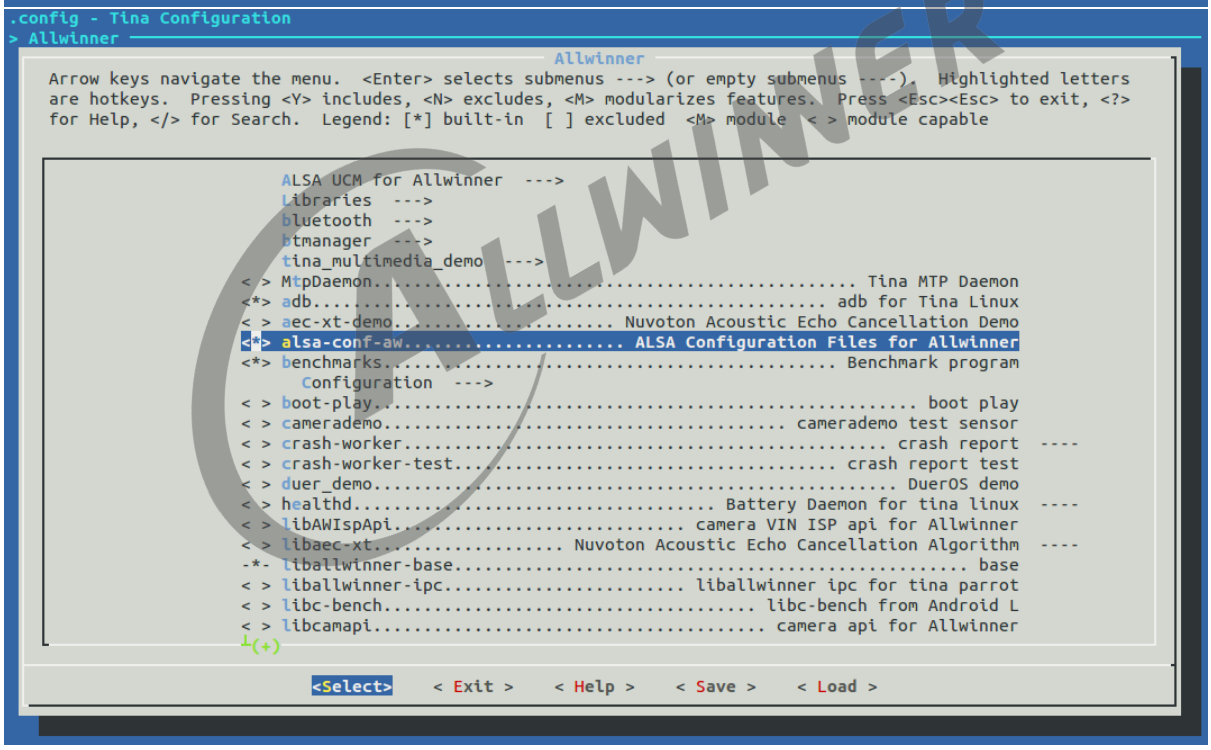
https://www.alsa-project.org/alsa-doc/alsa-lib/pcm_plugins.html

tina sdk 下有相关软件包会设置/etc/asound.conf，可以用作参考。

使用方法：

tina 根目录下执行 make menuconfig, 选择 alsa-conf-aw 软件包。





它会生成/etc/asound.conf 文件，下面作简单介绍：

```
设定amixer操作的defautl声卡(执行snd_hctl_open会获取该配置)
ctl.!default {
    type hw
    card audiocodec
}

设定default声卡(执行snd_pcm_open会获取该配置)
pcm.!default {
```

```
    type asym
    playback.pcm "PlaybackDmix"
    capture.pcm "CaptureDsnoop"
}

使用dmix插件，可以混合播歌，即支持多次打开声卡进行播歌
pcm.PlaybackDmix {
        type plug
        slave.pcm {
            type dmix
            ipc_key 1111
            ipc_perm 0666
            slave {
                    pcm "hw:audiocodec"
                    rate 48000
                    channels 2
            }
        }
}

使用dsnoop插件，可以混合录音，即支持多次打开声卡进行录音
pcm.CaptureDsnoop {
    type plug
    slave.pcm {
        type dsnoop
        ipc_key 1111
        ipc_perm 0666
        slave {
            pcm "hw:audiocodec,0"
            rate 48000
            channels 2
        }
    }
}

使用dmix插件以及softvol插件，softvol插件可以增加一个control，用于控制音量(软件上作调节)
pcm.PlaybackDmix {
    type plug
    slave.pcm {
        type softvol
        slave.pcm {
            type dmix
            ipc_key 1111
            ipc_perm 0666
            slave {
                pcm "hw:audiocodec,0"
                rate 48000
                channels 1
            }
        }
        control {
            name "Soft Volume Master"
            card audiocodec
        }
        min_dB -51.0
        max_dB 0.0
        resolution 256
    }
}
```

3.2 3.2 tinyalsa-utils

tinyalsa 是 alsa-lib 的一个简化版。它提供了 pcm 和 control 的基本接口；没有太多太复杂的操作、功能。可以按需使用接口。tinyalsa-utils 是基于 tinyalsa 的一些工具，下面对几个常用的工具作介绍。

3.2.1 3.2.1 tinymix

与 amixer 作用类似，用于操作 mixer control。

- 常用选项

选项	功能
-D,–card	指定声卡设备, 默认使用 card0

- 常用命令

命令	功能
controls	列出指定声卡的所有控件
contents	列出指定声卡的所有控件的具体信息
get	获取指定控件的信息
set	设定指定控件的值

举例:

```
获取card0的所有控件名
tinymix -D 0 controls

获取card0当前硬件音量
tinymix -D 0 get 'LINEOUT volume'

设置card0当前硬件音量
tinymix -D 0 set 'LINEOUT volume' 25
```

3.2.2 3.2.2 tinyplay

与 aplay 作用类似, 用于操作声卡设备进行播放

- 常用选项

选项	功能
-D,–card	指定声卡设备, 默认使用 card0
-p,–period-size	指定 period 大小, 单位为帧
-c,–channels	指定通道数
-r,–rate	指定采样率
-b,–bits	指定采样精度

如果播放的是 wav 文件, 可以解析头部, 识别通道数, 采样率等参数

举例:

```
tinyplay -D 0 /tmp/16000-stere-10s.wav
```

3.2.3 3.2.3 tinycap

与 arecord 作用类似，用于操作声卡进行录音功能

- 常用选项

选项	功能
-D,–device	指定声卡设备, 默认使用 card0
-p,–period-size	指定 period 大小, 单位为帧
-c,–channels	指定通道数
-r,–rate	指定采样率
-b,–bits	指定采样精度

举例:

```
    录制通道数为2, 采样率为16000, 采样精度为16bit, 保存为wav文件
    tinycap -D 0 -b 16 -r 16000 -c 2 /mnt/UDISK/test.wav
```

3.3 3.3 dump 寄存器

我们 sunxi 平台均提供了 sunxi_dump 驱动，用于查看读写寄存器。

节点位于/sys/class/sunxi_dump 目录。

但是 audiocodec 模拟寄存器的操作会有些特殊，我们一般在 audio 驱动中都会增加相关调试节点，去操作自己模块的寄存器，以便调试。

3.3.1 3.3.1 dump audiocodec 寄存器

audiocodec 驱动的寄存器调试节点位于:

```
/sys/devices/platform/soc/codec/audio_reg_debug/audio_reg
```

使用方法:

通过 echo 写入下列参数

参数 1：0-read; 1-write

参数 2：1-digitar reg; 2-analog reg

参数 3：reg value

参数 4：write value

举例：

查看所有寄存器状态:

```
cat /sys/devices/platform/soc/codec/audio_reg_debug/audio_reg

打印如下(其中地址为0x300以上的寄存器为模拟寄存器，其他均为数字寄存器):
dump audio reg:
SUNXI_DAC_DPC       [0x000]: 0x0            Save:0x0
SUNXI_DAC_FIFO_CTL  [0x010]: 0x3004000      Save:0x0
SUNXI_DAC_FIFO_STA  [0x014]: 0x80800c       Save:0x0
SUNXI_DAC_CNT       [0x024]: 0xb4014        Save:0x0
SUNXI_DAC_DG        [0x028]: 0x0            Save:0x0
SUNXI_ADC_FIFO_CTL  [0x030]: 0xe000400      Save:0x0
SUNXI_ADC_FIFO_STA  [0x038]: 0x0            Save:0x0
SUNXI_ADC_CNT       [0x044]: 0x0            Save:0x0
SUNXI_ADC_DG        [0x04c]: 0x0            Save:0x0
SUNXI_DAC_DAP_CTL   [0x0f0]: 0x0            Save:0x0
SUNXI_ADC_DAP_CTL   [0x0f8]: 0x0            Save:0x0
SUNXI_HP_CTL        [0x300]: 0x0            Save:0x0
SUNXI_MIX_DAC_CTL   [0x303]: 0x0            Save:0x0
SUNXI_LINEOUT_CTL0  [0x305]: 0x10           Save:0x0
SUNXI_LINEOUT_CTL1  [0x306]: 0x19           Save:0x0
SUNXI_MIC1_CTL      [0x307]: 0x34           Save:0x0
SUNXI_MIC2_MIC3_CTL [0x308]: 0x4            Save:0x0
SUNXI_LADCMIX_SRC   [0x309]: 0x4            Save:0x0
SUNXI_RADCMIX_SRC   [0x30a]: 0x8            Save:0x0
SUNXI_XADCMIX_SRC   [0x30b]: 0x10           Save:0x0
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
SUNXI_ADC_CTL       [0x30d]: 0x3          Save:0x0
SUNXI_MBIAS_CTL     [0x30e]: 0x21         Save:0x0
SUNXI_APT_REG       [0x30f]: 0xd6         Save:0x0
SUNXI_OP_BIAS_CTL0  [0x310]: 0x55         Save:0x0
SUNXI_OP_BIAS_CTL1  [0x311]: 0x55         Save:0x0
SUNXI_ZC_VOL_CTL    [0x312]: 0x2          Save:0x0
SUNXI_BIAS_CAL_CTRL [0x315]: 0x0          Save:0x0
```

查看某个数字寄存器状态：

```
echo 0,1,0x10 > /sys/devices/platform/soc/codec/audio_reg_debug/audio_reg

打印如下:
[  127.036609] sunxi-internal-codec codec: ret:3, reg_group:1, reg_offset:16, reg_val:0x0
[  127.045557] sunxi-internal-codec codec:
[  127.045557]
[  127.045557]  Reg[0x10] : 0x03004000
[  127.045557]

表示0x10数字寄存器的值为0x03004000
```

查看某个模拟寄存器状态：

```
echo 0,2,0x5 > /sys/devices/platform/soc/codec/audio_reg_debug/audio_reg

打印如下:
[  306.126103] sunxi-internal-codec codec: ret:3, reg_group:2, reg_offset:5, reg_val:0x0
[  306.134971] sunxi-internal-codec codec:
[  306.134971]
[  306.134971]  Reg[0x05] : 0x10
[  306.134971]

表示0x05模拟寄存器的值为0x10
```

改写某个数字寄存器:

```
echo 1,1,0x24,0 > /sys/devices/platform/soc/codec/audio_reg_debug/audio_reg
表示将0x24数字寄存器写为0x0
```

改写某个模拟寄存器:

```
echo 1,2,0x3,0x1 > /sys/devices/platform/soc/codec/audio_reg_debug/audio_reg
表示将0x03模拟寄存器写为0x1
```

3.3.2 3.3.2 dump daudio 寄存器

查看 spec 可以知道 i2s 模块的寄存器基地址

```
i2s0: 0x05090000
i2s1: 0x05091000
i2s2: 0x05092000
```

可以通过 sunxi_dump 节点查询寄存器状态，例如查看 i2s0 的寄存器:

```
cd /sys/class/sunxi_dump
echo 0x05090000,0x050900a0 > dump
cat dump
```

3.3.3 3.3.3 dump dmic 寄存器

查看 spec 可以知道 dmic 模块的寄存器基地址

```
dmic: 0x05095000
```

可以通过 sunxi_dump 节点查询寄存器状态:

```
cd /sys/class/sunxi_dump
echo 0x05095000,0x05095050 > dump
cat dump
```

3.3.4 3.3.4 dump spdif 寄存器

查看 spec 可以知道 spdif 模块的寄存器基地址

```
spdif: 0x05093000
```

可以通过 sunxi_dump 节点查询 spdif 寄存器状态:

```
cd /sys/class/sunxi_dump
echo 0x05093000,0x05093040 > dump
cat dump
```

3.4 3.4 sound procfs

通过 procfs 文件系统下面的声卡相关节点，可以得到各个声卡各个音频流的状态。实际调试中会非常有用。

内核需要选中下面选项才能在 procfs 下生成对应节点：

```
Device Drivers  --->
<*> Sound card support  --->
   <*> Advanced Linux Sound Architecture  --->
      [*]   Sound Proc FS Support
      [*]     Verbose procfs contents
```

以 card0 为例看下提供的节点信息：

```
ddd
/proc/asound/card0/
├── id              /* 声卡名称 */
├── pcm0c           /* pcm0 录音流 */
│   ├── info        /* pcm信息 */
│   └── sub0
│       ├── hw_params  /* 硬件参数信息 */
│       ├── info       /* pcm信息 */
│       ├── status     /* pcm流运行状态 */
│       └── sw_params  /* 软件参数信息 */
└── pcm0p           /* pcm0 播放流 */
    ├── info
    └── sub0
```

其中，hw_params, status 都能拿到比较有用的信息：

```
cat /proc/asound/card0/pcm0c/sub0/hw_params
access: RW_INTERLEAVED                  /* 交错模式排列通道 */
format: S16_LE                          /* 当前音频流的采样精度 */
subformat: STD
channels: 2                             /* 通道数 */
rate: 16000 (16000/1)                   /* 采样率 */
period_size: 320                        /* 周期(决定dma中断时间,例如这里period_time=320/16000=20ms)
     */
buffer_size: 2560                       /* 内核ALSA框架中环形缓冲区大小,决定能够缓存多少个period */

cat /proc/asound/card0/pcm0c/sub0/status
state: RUNNING                          /* 音频流运行状态,RUNNING, SETUP等状态 */
owner_pid   : 22653
trigger_time: 81828.078175765
tstamp      : 82373.796969347           /* 开始运行后的时间戳信息 */
delay       : 256
avail       : 256                       /* 当前可用音频数据帧数 */
avail_max   : 320
-----
hw_ptr      : 8731456                   /* 硬件逻辑指针，单位(帧) */
appl_ptr    : 8731200                   /* 应用逻辑指针，单位(帧) */
```

- 从 period_size 可以知道当前 dma 中断频率，太快会影响系统响应速度，太慢可能就存在一定延时。
- buffer_size 可以知道缓存区大小，太小容易因调度不及时出现 xrun, 太大同样存在一定延时。
- 从 hw_ptr, appl_ptr 可以知道当前录音/播音的帧数，是否发生过 xrun 等。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4 4 常用接口说明

这里主要介绍 alsa-lib 中的常用接口

4.1 4.1 control 接口

为了方便操作访问，alsa-lib 中封装了相关接口, 通过 control 节点 (/dev/snd/controlCX) 去获取、设置 control elements

主要涉及到的接口：

```
snd_ctl_open
snd_ctl_elem_info_get_id
snd_ctl_elem_info_set_id
snd_ctl_elem_info
snd_ctl_ascii_value_parse
snd_ctl_elem_read
snd_ctl_elem_write
snd_ctl_close
```

详细 control 接口说明请查阅：

https://www.alsa-project.org/alsa-doc/alsa-lib/control.html

https://www.alsa-project.org/alsa-doc/alsa-lib/group___control.html

下面是一个设置音量接口的例子：

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <stdint.h>
#include <unistd.h>
#include <string.h>

#include <alsa/asoundlib.h>

#define DEV_NAME    "hw:audiocodec"
#define VOLUME_CONTROL  "name='LINEOUT volume'"

/* Fuction to convert from percentage to volume. val = volume */
static int convert_volume(int percent, long min, long max)
{
    long range = max - min;
```

```
    if (range == 0)
        return 0;
    return (int)((range * percent / 100) + min);

}

bool controlVolume(int volume_percent)
{
    int err = -1;
    snd_ctl_t *handle = NULL;
    char *card = DEV_NAME;
    char *volume_control = VOLUME_CONTROL;
    char volume_string[4];
    long min, max, raw;

    snd_ctl_elem_info_t *info = NULL;
    snd_ctl_elem_id_t *id = NULL;
    snd_ctl_elem_value_t *control = NULL;

    if (volume_percent > 100 || volume_percent < 0)
        return false;

    snd_ctl_elem_info_alloca(&info);
    snd_ctl_elem_id_alloca(&id);
    snd_ctl_elem_value_alloca(&control);

    err = snd_ctl_ascii_elem_id_parse(id, volume_control);
    if (err < 0) {
        fprintf(stderr, "Wrong control identifier: %s\n", volume_control);
        goto failed;
    }
    err = snd_ctl_open(&handle, card, 0);
    if (err < 0) {
        fprintf(stderr, "Control device %s open error:%s\n", card, snd_strerror(err));
        goto failed;
    }
    snd_ctl_elem_info_set_id(info, id);
    err = snd_ctl_elem_info(handle, info);
    if (err < 0) {
        fprintf(stderr, "Cannot find the given element from control %s\n", card);
        goto failed;
    }
    snd_ctl_elem_info_get_id(info, id);
    snd_ctl_elem_value_set_id(control, id);
    err = snd_ctl_elem_read(handle, control);
    if (err < 0) {
        fprintf(stderr, "Cannot read the given element from control %s\n", card);
        goto failed;
    }
    min = snd_ctl_elem_info_get_min(info);
    max = snd_ctl_elem_info_get_max(info);
    snprintf(volume_string, sizeof(volume_string), "%d", convert_volume(volume_percent, min
    , max));
    /*printf("set volume %s, [%u%%]\n", volume_string, volume_percent);*/
    err = snd_ctl_ascii_value_parse(handle, control, info, volume_string);
    if (err < 0) {
        fprintf(stderr, "Control %s parse error: %s\n", card, snd_strerror(err));
        goto failed;
    }
    err = snd_ctl_elem_write(handle, control);
```

```
    if (err < 0) {
        fprintf(stderr, "Control %s write error: %s\n", card, snd_strerror(err));
        goto failed;
    }
failed:
    if (info)
        snd_ctl_elem_info_free(info);
    if (id)
        snd_ctl_elem_id_free(id);
    if (control)
        snd_ctl_elem_value_free(control);
    if (handle)
        snd_ctl_close(handle);

    return ((err < 0) ? false : true);
}
```

4.2 4.2 PCM 接口

为了方便操作访问，alsa-lib 中封装了相关接口，通过 pcmCXDXp/pcmCXDXc 节点 (/dev/snd/pcmCXDXx) 去实现播放、录音功能。

主要涉及到的接口：

```
snd_pcm_open
snd_pcm_info
snd_pcm_hw_params_any
snd_pcm_hw_params_set_access
snd_pcm_hw_params_set_format
snd_pcm_hw_params_set_channels
snd_pcm_hw_params_set_rate_near
snd_pcm_hw_params_set_buffer_size_near
snd_pcm_hw_params
snd_pcm_sw_params_current
snd_pcm_sw_params
snd_pcm_readi
snd_pcm_writei
snd_pcm_close
```

详细 pcm 接口说明请查阅：

https://www.alsa-project.org/alsa-doc/alsa-lib/pcm.html

https://www.alsa-project.org/alsa-doc/alsa-lib/group___p_c_m.html

接口使用例子可以参考 aplay,arecord 的实现，代码可以在 alsa-utils 中找到 (dl/alsa-utils-1.1.0.tar.bz2)

版权所有 © 珠海全志科技股份有限公司。保留一切权利

5 开机音乐功能

这里开机音乐功能，指的是在 uboot 阶段开始启动 dma 将音频数据搬运到 AudioCodec 的 FIFO 中进行播歌，同时 CPU 继续开机流程进入内核。进入系统后，在合适的启动脚本中加载音频驱动模块 (如果 builtin，那么音乐会提前中止), 这样开机音乐可以大大的提前，给用户一种迅速开机的错觉。

目前 SDK 代码中支持 uboot 开机音乐功能的平台有:R6,R7s,R11,R16,R328,R311,MR133

配置使用方法都比较类似，下面以 R328 为例进行说明。

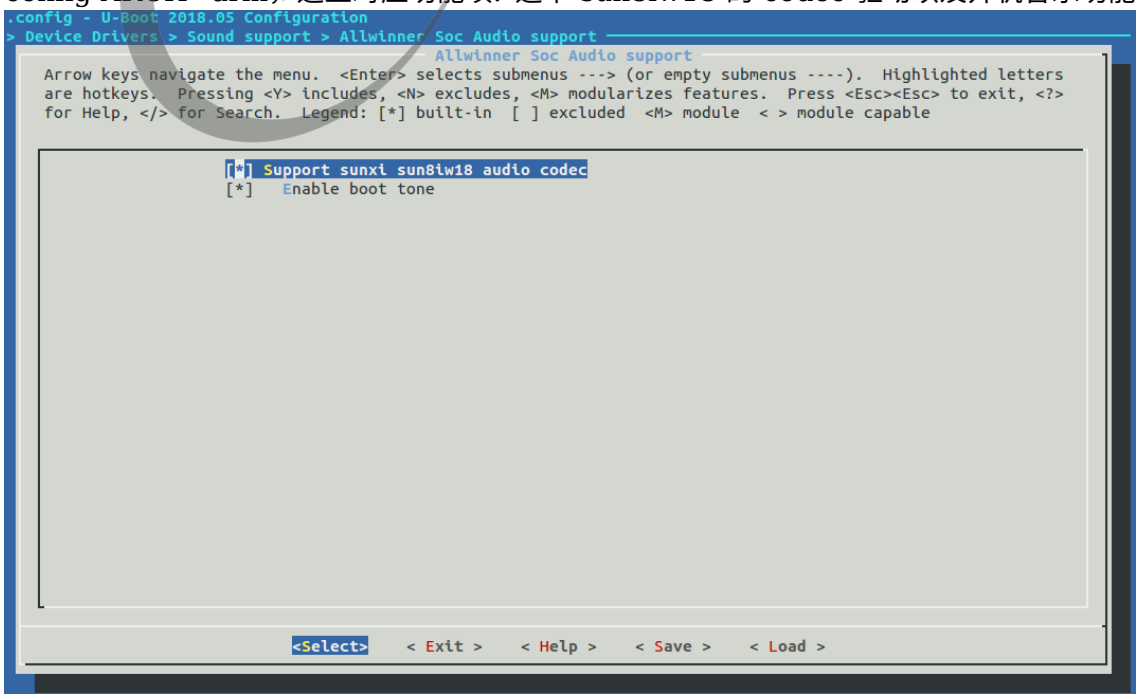
5.1 配置方法

1. 修改 uboot 配置文件

- 可修改默认配置 configs/sun8iw18p1_defconfig 增加下面几项

```
CONFIG_SOUND_SUNXI_SOC_RWFUNC=y
CONFIG_SOUND_SUNXI_SUN8IW18_CODEC=y
CONFIG_SOUND_SUNXI_BOOT_TONE=y
```

- 也可以通过 kconfig 进行配置 lichee/brandy-2.0/u-boot-2018 目录下执行 make menuconfig ARCH=arm，选上对应功能项：选中 sun8iw18 的 codec 驱动以及开机音乐功能



2. 修改 sys_config 文件

   配置 boottone_used 为 1, 表示使用开机音乐功能

```
;-----------------------------------------------------------------------------
;boot tone configuration
;
;boottone_used ---boot tone enable
;len_limit ---set size in bytes, normally do not need to set it and driver will use
  file size
;-----------------------------------------------------------------------------
[boottone]
boottone_used = 1
```

   另外 codec 节点也需要配置正确, 下面是部分重要配置

```
[codec]
codec_used = 0x1
lineout_vol =0x1a
gpio-spk = port:PH9<1><1><1><1>
```

3. 创建 boottone 分区

   开机音乐的音频文件需要放置在单独一个分区中，例如 boottone 分区，修改 sys_partition.fex 文件：

```
[partition]
   name            = boottone
   size               = 2048
   downloadfile = "boottone.fex"
   user_type       = 0x8000
```

   上述 2048 表示 1M 的空间, 注意根据实际音频文件大小填写合适的 size

   然后将音频文件重命名为 boottone.fex, 并放置到方案配置目录下，以 cowbell-perf1 方案为例：

```
mv kaiji.wav target/allwinner/cowbell-perf1/configs/boottone.fex
```

4. 修改 env 配置文件

   主要增下面几行：

```
uboot_tone_addr=0x4327ffd4
boottone_partition=boottone
load_boottone=sunxi_flash read ${uboot_tone_addr} ${boottone_partition}
```

   uboot_tone_addr 用于指定音频文件加载到 dram 的地址

   boottone_partition 用于指定音频文件所在分区名

   load_boottone 用于加载音频文件到 dram 的命令

5. 将内核 AudioCodec 驱动编译成模块

   make kernel_menuconfig 去除下面几个配置：

```
@@ -702,14 +705,9 @@ CONFIG_SND_SOC_I2C_AND_SPI=y
 # CONFIG_SND_SOC_WM8960 is not set
 # CONFIG_SND_SOC_WM8974 is not set
 # CONFIG_SND_SOC_WM8985 is not set
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

```
-CONFIG_SND_SUN8IW18_CODEC=y

 # CONFIG_SND_SUNXI_MAD is not set
-CONFIG_SND_SUNXI_SOC=y
-CONFIG_SND_SUNXI_SOC_CPUDAI=y
-CONFIG_SND_SUNXI_SOC_DAUDIO=y
-CONFIG_SND_SUNXI_SOC_RWFUNC=y
-CONFIG_SND_SUNXI_SOC_SUN8IW18_CODEC=y
-CONFIG_SND_SUNXI_SOC_SUNXI_DAUDIO=y
+# CONFIG_SND_SUNXI_SOC_SUN8IW18_CODEC is not set
+# CONFIG_SND_SUNXI_SOC_SUNXI_DAUDIO is not set
 # CONFIG_SND_SUNXI_SOC_SUNXI_DMIC is not set
 # CONFIG_SND_SUNXI_SOC_SUNXI_SPDIF is not set
 CONFIG_SND_SUPPORT_OLD_API=y
```

make menuconfig 增加驱动模块配置

```
@@ -2923,7 +2929,7 @@ CONFIG_PACKAGE_kmod-sound-core=y
 # CONFIG_PACKAGE_kmod-sound-soc-ac97 is not set
 # CONFIG_PACKAGE_kmod-sound-soc-core is not set
 # CONFIG_PACKAGE_kmod-sound-via82xx is not set
-# CONFIG_PACKAGE_kmod-sunxi-sound is not set
+CONFIG_PACKAGE_kmod-sunxi-sound=y
```

5.2 注意事项

1. uboot 退出时，不能关掉 dma。如果发生在跳转 kenrel 时声音变化或消失，请确认跳转到内核前是否调用了 sunxi_dma_exit();
2. 内核并不知道 uboot 将音频加载到 dram 的位置。理论上,kernel 可能会在初始化过程中，用到那片内存，导致音乐播放不正常。
   - 可以在 dts 中，将该片内存改为 reserve 属性，可以确保 kernel 不会使用到。
   - 选用 kernel 初始化不会用到的内存块
3. 可以在 uboot 命令行下执行 boottone 命令进行调试

版权所有 © 珠海全志科技股份有限公司。保留一切权利

6 各平台音频模块注意事项

6.1 R328

1. 只有一个 DAC，所以在播放两通道数据的时候，硬件上会将第二个通道的数据丢掉。如果想要将两通道数据都完成播放出来，需要在软件上将两通道合成，可利用 alsa 插件实现, 例如下面配置：

```
pcm.playback {
        type plug
        slave {
                pcm "hw:audiocodec,0"
                rate 48000
                channels 1
        }
}
```

   alsa 插件会将两通道数据的幅度衰减为由原来的一半 (如果直接相加，幅度为原来的两倍，有可能造成削顶)，再组合为一通道写入声卡中。

2. 使用模拟 mic 作 AEC 时，需要将该 MIC gain 设置为 0dB. 如果有增益，则会导致录音开始及结束时产生 pop 音 (开关 PA 产生的 pop 音，经过差分 MIC 消除后仍然存在极小的杂音，如果再经过 MIC 增益放大，则会变为明显的 pop 音)

版权所有 © 珠海全志科技股份有限公司。保留一切权利

著作权声明

版权所有 ©2022 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、、全志科技、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的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所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。