# [ALSA音频驱动框架](https://www.cnblogs.com/jiangjh/p/10994488.html)

1. 关键字
2. 代码路径

**sound目录**

core              该目录包含了ALSA驱动的中间层，它是整个ALSA驱动的核心部分

core/oss      包含模拟旧的OSS架构的PCM和Mixer模块

core/seq      有关音序器相关的代码

include         ALSA驱动的公共头文件目录，该目录的头文件需要导出给用户空间的应用程序使用，通常，驱动模块私有的头文件不应放置在这里

drivers          放置一些与CPU、BUS架构无关的公用代码

i2c                 ALSA自己的I2C控制代码

pci                 pci声卡的顶层目录，子目录包含各种pci声卡的代码

isa                 isa声卡的顶层目录，子目录包含各种isa声卡的代码

soc                针对system-on-chip体系的中间层代码

soc/codecs  针对soc体系的各种codec的代码，与平台无关

sound/core/sound.c

- 实现了最顶层的file\_operations，它起中转作用

sound/core/control.c

- 实现了控制接口的file\_operations

sound/core/pcm\_native.c

- 实现了playback，capture的file\_operations

1. 架构图

3.1. Linux音频框架图

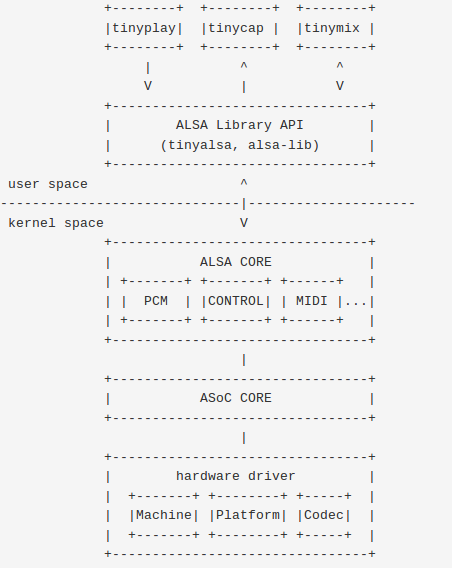


图1 Linux音频框架图

3.1. Hardware Driver：音频硬件设备驱动，由三大部分组成，分别是 Machine、Platform、Codec

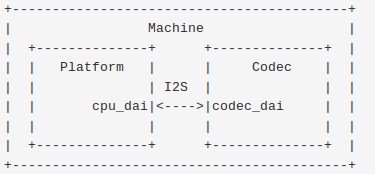


图2 音频硬件设备驱动结构

3.2播放音频的数据流向图

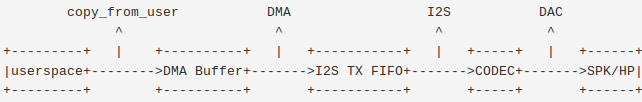


图3 数据流向图

1. ASoC ALSA
   1. Machine

- 单板相关，表明platform是哪个，cup DAI是哪个，DMA是哪个，Codec是哪个，Codec DAI是哪个

sound\soc\imx\imx-3stack-sgtl5000.c

1. **static** **struct** platform\_driver imx\_3stack\_sgtl5000\_audio\_driver = {
2. .probe = imx\_3stack\_sgtl5000\_probe,
3. .remove = imx\_3stack\_sgtl5000\_remove,
4. .driver = {
5. .name = "imx-3stack-sgtl5000",
6. },
7. };

arch\arm\mach-s3c2440\mach-mini2440.c

1. **static** **struct** platform\_device mini2440\_audio = {
2. .name       = "s3c24xx\_uda134x",
3. .id     = 0,
4. .dev        = {
5. .platform\_data  = &mini2440\_audio\_pins,
6. },
7. };

sound\soc\soc-core.c

1. /\* ASoC platform driver \*/
2. **static** **struct** platform\_driver soc\_driver = {
3. .driver     = {
4. .name       = "soc-audio",
5. .owner      = THIS\_MODULE,
6. .pm     = &soc\_pm\_ops,
7. },
8. .probe      = soc\_probe,
9. .remove     = soc\_remove,
10. };

sound\soc\soc-core.c

1. **struct** snd\_soc\_card \*card = socdev->card;
3. /\* Bodge while we push things out of socdev \*/
4. card->socdev = socdev;
6. /\* Bodge while we unpick instantiation \*/
7. card->dev = &pdev->dev;
8. ret = snd\_soc\_register\_card(card);

sound\soc\imx\imx-3stack-sgtl5000.c

1. **static** **struct** snd\_soc\_card snd\_soc\_card\_imx\_3stack = {
2. .name = "imx-3stack",
3. .platform = &imx\_soc\_platform,
4. .dai\_link = &imx\_3stack\_dai,
5. .num\_links = 1,
6. .remove = imx\_3stack\_card\_remove,
7. };
   1. Platform

- CPU DAI：设置接口，snd\_soc\_dai\_driver

- DMA：传输数据，snd\_soc\_platform\_driver

* 1. Codec

- DAI：设置接口，snd\_soc\_dai\_driver

- 控制接口，snd\_soc\_codec\_driver