## **Frame & Period**

后面章节将分析 dma buffer 的管理，其中细节需要对音频数据相关概念有一定的了解。因此本章说明下音频数据中的几个重要概念：

* Sample：样本长度，音频数据最基本的单位，常见的有 8 位和 16 位；
* Channel：声道数，分为单声道 mono 和立体声 stereo；
* Frame：帧，构成一个完整的声音单元，所谓的声音单元是指一个采样样本，Frame = Sample \* channel；
* Rate：又称 sample rate，采样率，即每秒的采样次数，针对帧而言；
* Period Size：周期，每次硬件中断处理音频数据的帧数，对于音频设备的数据读写，以此为单位；
* Buffer Size：数据缓冲区大小，这里指 runtime 的 buffer size，而不是结构图 snd\_pcm\_hardware 中定义的 buffer\_bytes\_max；一般来说 buffer\_size = period\_size \* period\_count， period\_count 相当于处理完一个 buffer 数据所需的硬件中断次数。 下面一张图直观的表示 buffer/period/frame/sample 之间的关系：

read/write pointer

|

v

+-----------------------------+--------------+--------------+

| | | | |buffer = 4 periods

+--------------+--------------+--------------+--------------+

^

|

+---+---+------+

| | | ... |period = 1024 frames

+---+---+------+

^

|

+---+

|L|R|frame = 2 samples (left + right)

+---+

sample = 2 bytes (16bit)

这个 buffer 中有 4 个 period，每当 DMA 搬运完一个 period 的数据就会出生一次中断，因此搬运这个 buffer 中的数据将产生 4 次中断。ALSA 为什么这样做？因为数据缓存区可能很大，一次传输可能会导致不可接受的延迟；为了解决这个问题，alsa 把缓存区拆分成多个周期，以周期为单元传输数据。

root@jb1505:/data # tinyplay fly-2ch-8k-125s.wav  -D 0 -d 2 -p 160 -n 2

size:1280 bytes

Playing sample: 2 ch, 8000 hz, 16 bit

Channels = 2，rate = 8000hz,period\_size = 160,period\_count = 2 求buffer size大小

Size = period\_size \* period\_count = 160帧 \* 2 = 320帧= 320 \* 2 samples (left + right) = 320 \* 2 \* 16 bit = 1280B

size = pcm\_frames\_to\_bytes(pcm, pcm\_get\_buffer\_size(pcm));

## **7.1. Frames & Periods**

敏感的读者会察觉到 period 和 buffer size 在 PCM 数据搬运中扮演着非常重要的角色。下面引用两段来自 alsa 官网对 Period 的详细解释：

**Period**

The interval between interrupts from the hardware. This defines the input latency, since the CPU will not have any idea that there is data waiting until the audio interface interrupts it. The audio interface has a “pointer” that marks the current position for read/write in its h/w buffer. The pointer circles around the buffer as long as the interface is running. Typically, there are an integral number of periods per traversal of the h/w buffer, but not always. There is at least one card (ymfpci) that generates interrupts at a fixed rate indepedent of the buffer size (which can be changed), resulting in some “odd” effects compared to more traditional designs. Note: h/w generally defines the interrupt in frames, though not always. Alsa’s period size setting will affect how much work the CPU does. if you set the period size low, there will be more interrupts and the work that is done every interrupt will be done more often. So, if you don’t care about low latency, set the period size large as possible and you’ll have more CPU cycles for other things. The defaults that ALSA provides are in the middle of the range, typically. (from an old AlsaDevel thread[1], quoting Paul Davis) Retrieved from “[http://alsa.opensrc.org/Period](https://link.zhihu.com/?target=http://alsa.opensrc.org/Period" \t "https://zhuanlan.zhihu.com/p/_blank)”

**FramesPeriods**

A frame is equivalent of one sample being played, irrespective of the number of channels or the number of bits. e.g.

这里不做翻译了，简单说下 Frame 和 Period 要点：

* Frame：帧，构成一个完整的声音单元，它的大小等于 sample\_bits \* channels；
* Peroid：周期大小，即每次 dma 运输处理音频数据的帧数。如果周期大小设定得较大，则单次处理的数据较多，这意味着单位时间内硬件中断的次数较少，CPU 也就有更多时间处理其他任务，功耗也更低，但这样也带来一个显著的弊端——数据处理的时延会增大。 再说说 period bytes，对于 dma 处理来说，它直接关心的是数据大小，而非 period\_size（一个周期的帧数），有个转换关系：period\_bytes = period\_size \* sample\_bits \* channels / 8

由于 I2S 总线采样率是稳定的，我们可以计算 I2S 传输一个周期的数据所需的时间：transfer\_time = 1 \* period\_size / sample\_rate, in second

例如 period\_size = 1024，sample\_rate = 48KHz ，那么一个周期数据的传输时间是： 1 \* 1024 / 48000 = 21.3 (ms)。