

瑞芯微 | I2S-音频基础 -1

原创

一口Linux

于 2024-03-08 20:53:56 发布


阅读量3.2k

收藏 44

点赞数 25

分类专栏: 瑞芯微 文章标签: 音视频

版权

 瑞芯微 专栏收录该内容

208 订阅 29 篇文章 订阅专栏

最近调试音频驱动，顺便整理学习了一下i2s、alsa相关知识，整理成了几篇文章，后续会陆续更新。

喜欢嵌入式、Li怒晓得老铁可以关注一口君账号。

1. 音频常用术语

名称	含义
ADC (Analog to Digit Conversion)	模拟信号转换为数字信号
AEC (Acoustic Echo Cancellor)	回声消除
AGC (Automatic Gain Control)	自动增益补偿，调整MIC收音量
ALSA (Advanced Linux Sound Architecture)	高级Linux声音架构
ANS (Automatic Noise Suppression)	背景噪音抑制，ANS可探测出背景固定频率的杂音并消除背景噪音
BCK (Bit Clock Line)	位时钟，对应数字音频的每一位数据。标准称为SCK (Serial Clock)，串行时钟。SCK=2x采样频率x采样位数
Codec	Coder/Decoder
DAC (Digit to Analog Conversion)	数字信号转换为模拟信号
DAI (Digital Audio Interface)	数字音频接口
DAPM (Dynamic Audio Power Management)	动态电源管理，DAPM可使基于linux的移动设备上的音频子系统，在任何时候都工作在最小功耗状态
DRC (Dynamic Range Control)	动态压缩，将音频输出控制在一定范围内
DSP	Digital Signal Processor
EQ (Equaliser)	均衡器，通过对

觉得还不错? 一键收藏

名称	含义
I2S (Inter-IC Sound)	IC间传输数字音频资料的一种接口标准，采用序列的方式传输2组（左右声道），Codec与CPU间音频的通信协议/接口/总线
LRCK (Left-Right Clock)	帧时钟，用于切换左右声道数据，0：左声道；1：右声道。标准称为WS(World Select)，声道选择；或称为FS (Frame Sync)，帧同步；LRCK的频率=采样频率
MCLK (Master Clock)	主时钟，一般MCLK=256*LRCK。不是I2S标准中的一部分，主要用来同步模拟/数字转换器的内部操作
Mixer	混音器，将来自不同通道的几种音频模拟信号混合成一种模拟信号
Mono	单声道
Mute	消音，屏蔽信号通道
OSS(Open Sound System)	开放声音系统，老的Linux音频体系结构，被ASLA取代并兼容
PCM (Pulse Code Modulation)	脉冲编码调制，一种从音频模拟信号转换成数字信号的技术，区别于PCM音频通信协议；I2S是PCM的子集
ramp	逐步增加或减少音量等级，避免声音急速变化，用于暂停或恢复音乐
SSI (Serial Sound Interface)	
Stereo	双声道
TDM (Time Division Multiplexing)	时分复用。I2S最多只能传2声道数据，TDM最多支持16通道

2. Pcm (playback、capture)

PCM 是英文 Pulse-code modulation 的缩写，中文译名是脉冲编码调制。PCM就是要把声音从模拟转换成数字信号的一种技术，简单的来说就是利用一个固定的频率对模拟信号进行采样，采样后的信号的幅值按一定的采样精度进行量化，量化后的数值被连续地输出、传输、处理或记录到存储介质中。

PCM 信号的两个重要指标是 采样频率 和 量化精度。

通常，播放音乐时，用程序从存储介质中读取音频数据(MP3、WMA、AAC...)，经过解码后，最终送到音频驱动程序中的就是PCM数据，反过来，在录音时，音频驱动不停地把采样所得的PCM数据送回给应用程序，由应用程序完成压缩、存储等任务。所以，音频驱动的两大核心任务就是：

- playback

如何把用户空间的应用程序发过来的PCM数据,转化为人耳可以辨别的模拟音频；

- capture

把mic拾取到得模拟信号,经过采样、量化,转换为PCM

觉得还不错？ 一键收藏

3. 声音要素

声音三要素 - 音调、响度、音色

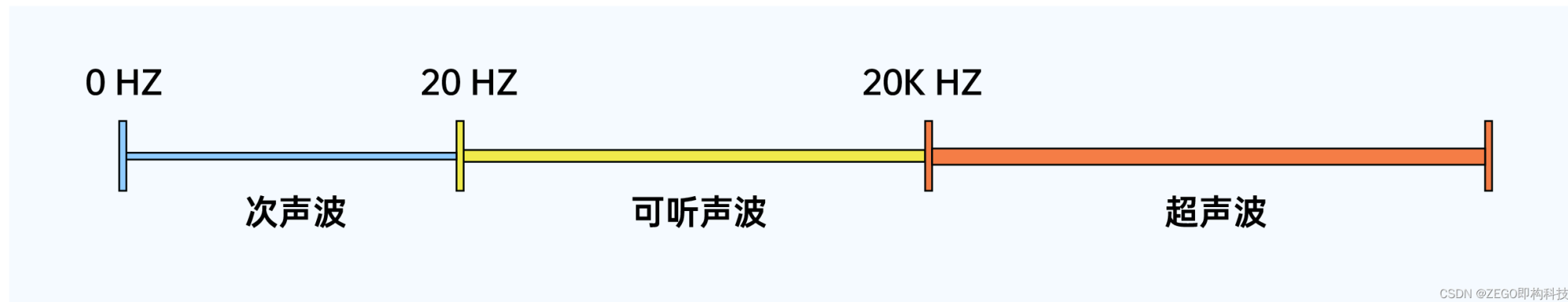
- 1、音调

“刺耳、低沉”，这其实是我们对声音高低的感受描述，这一特征我们称之为音调。

在物理定义上，声音是物体振动（比如我们的声带）产生的波，而音调由发声体振动的频率决定，频率越高（振动越快）则音调越高，听起来就越“刺耳”，反之音调越低、听起来就越低沉。

我们声带的振动频率，约在**100Hz~10KHz**之间，基本对应于常说的男低音至女高音的频率。

而我们耳朵的听力范围仅限于频率20Hz ~ 20KHz，低于或者高于这个频率范围的声音，分别被称为次声波（< 20Hz）和超声波（> 20KHz），无法被人耳感知。不难发现，虽然人耳的感知范围有限，但人类的发声频率完全包含于人耳的感知范围之内，这意味着任何人说的话，总会被耳朵捕捉到，每个人都有发声的权力，也总有一双耳朵能倾听到你的声音。



- 2、响度

“响亮、微弱”，是我们对声音强弱的感受描述，这种特征我们称之为响度。响度由发声体振动的幅度决定，当传播的距离相同时，振动幅度越大、则响度越大；相反，当振幅一定时，传播距离越远，响度越小，就是我们常说的“距离太远了，听不见”的原因。

- 3、音色

“风声、雨声、人声”，是我们对各种音调、各种响度声音的综合感受，这种特征我们称之为“音色”。音色是一种“感官属性”，我们利用这种“感官属性”，能区分发声的物体，发声的状态，还能评价听感上的优劣，比如“钢琴声、二胡声”，比如“只闻其声，如见其人”，比如“悦耳、动听”等等。那么音色是怎么“产生”的，又由什么“决定”呢？前面我们了解到，声音是由物体振动产生的波，而物体整体振动发出的只是基音，其各部分还有复合的振动，这些复合的振动也会发出声音并形成泛音，基音+泛音的不同组合就产生了多样化的音色，声音世界才变得丰富多彩起来。我们一般认为音色由发声体的材质决定。

觉得还不错？ [一键收藏](#)



一口Linux

关注

👍 25



★ 44

💬 0



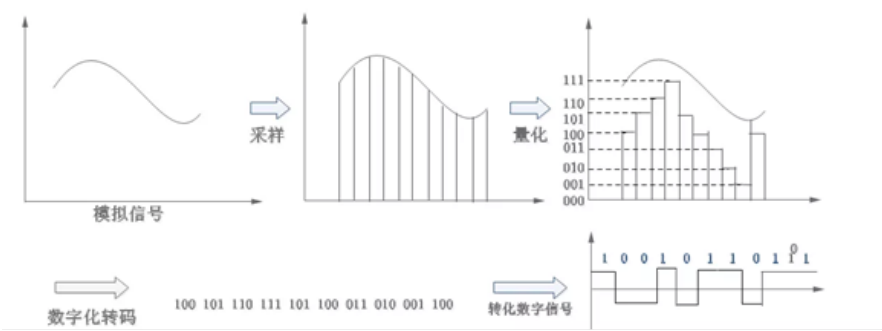
💰 打赏



	音调	响度	音色
概念	人耳对声音高低的感觉	人耳对声音强弱的感觉	人耳对各种音调,各种响度声波的综合反应,用于区分声音归属
决定因素	声波振动的频率	声波振动的幅度	取决于发声体的材质
形容词	低沉/ 刺耳/ 尖锐	震耳欲聋	风声/ 雨声/ 读书声

CSDN @ZEGO即构科技


4. 声音采样-ADC/DAC



处理器要想“听到”外界的声音必须要把外界的声音转化为自己能够理解的“语言”，处理器能理解的就是 0 和 1，也就是二进制数据。

所以我们需要先把外界的声音转换为处理器能理解的 0 和 1，在信号处理领域，外界的声音是模拟信号，处理器能理解的是数字信号，因此这里就涉及到一个模拟信号转换为数字信号的过程，而完成这个功能的就 是 ADC 芯片。

觉得还不错? 一键收藏

 一口Linux

关注

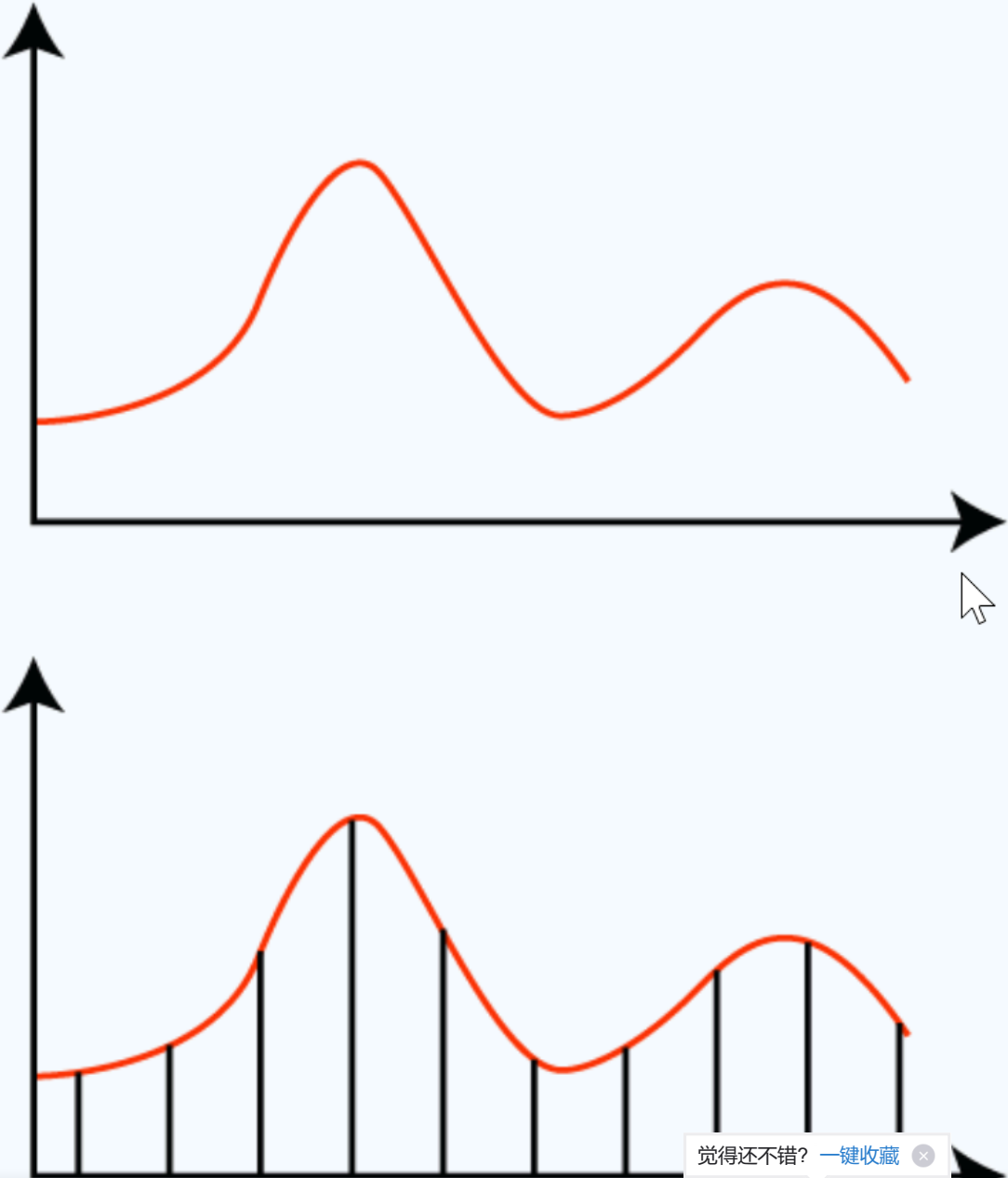
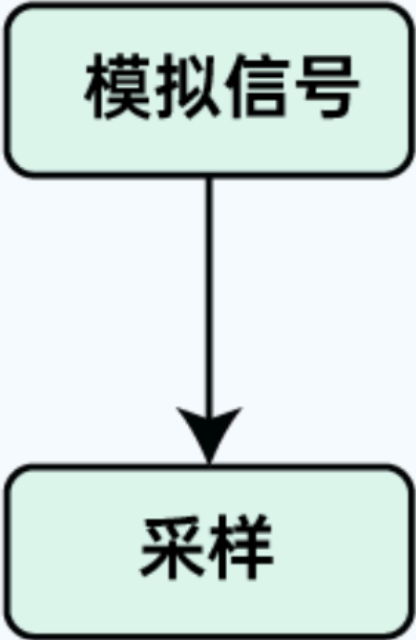
25

44

0

分享

打赏



一口Linux

关注

25



44



0



分享



打赏



觉得还不错? 一键收藏



1) 采样

以一定采样率，在时间轴上对模拟信号进行数字化。

首先，我们沿着时间轴，按照固定的时间间隔 T （假设 $T=0.1s$ ），依次取多个点（如图中 1~10 所对应波上的点）。

此时 T 称为取样周期， T 的倒数为本次取样的采样率（ $f=1/T=10Hz$ ）， f 即表示每秒钟进行采样的次数，单位为赫兹（Hz）。显然，采样率越高、单位时间的采样点越多，就能越好的表示原波形（如果高频率、密集地采集无数个点，就相当于完整地记录了原波形）。

2) 量化

以一定精度，在幅度轴上对模拟信号进行数字化。

完成采样后，我们接下来进行音频数字化的第二步，量化。采样是在时间轴上对音频信号进行数字化，得到多个采样点；而量化，则是在幅度方向上进行数字化，得到每个采样点的幅度值。

如下图所示，我们设定纵轴的坐标取值范围为 $0 \sim 8$ ，得到每个采样点的纵坐标（向上取整），这里的坐标值即为量化后的幅度值。因为我们将幅度轴分为了 8 段，有 8 个值用于量化取整，即本次量化的精度为 8。

显然，如果分段越多，则幅度的量化取值将越准确（取整带来的误差就越小），也能越好的表示原波形。对于幅度的量化精度，有一个专有术语描述 – 位深，我们后面会详细说明。

觉得还不错? 一键收藏



一口Linux

关注

👍 25



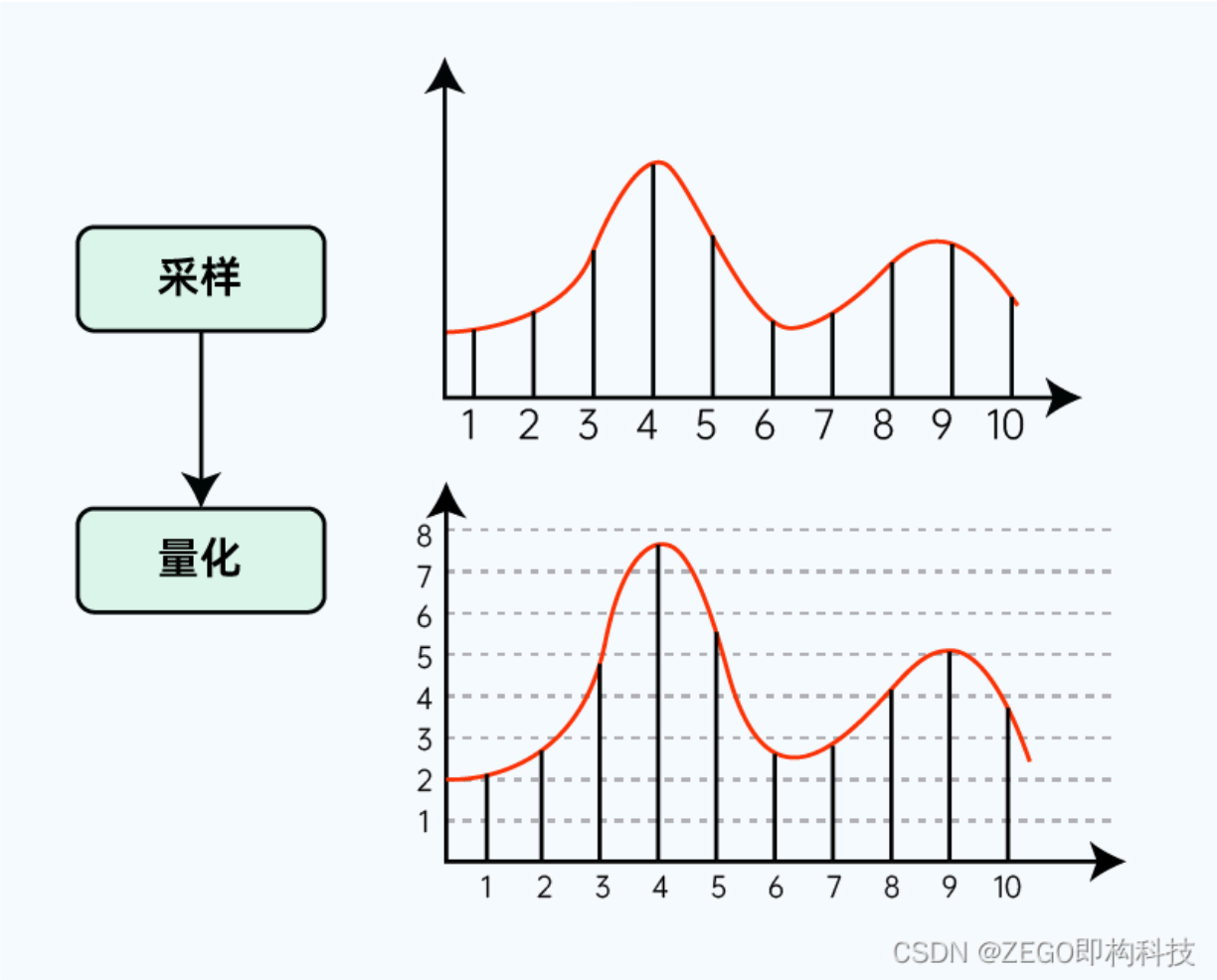
★ 44

💬 0

🔗 分享

💰 打赏





3) 编码

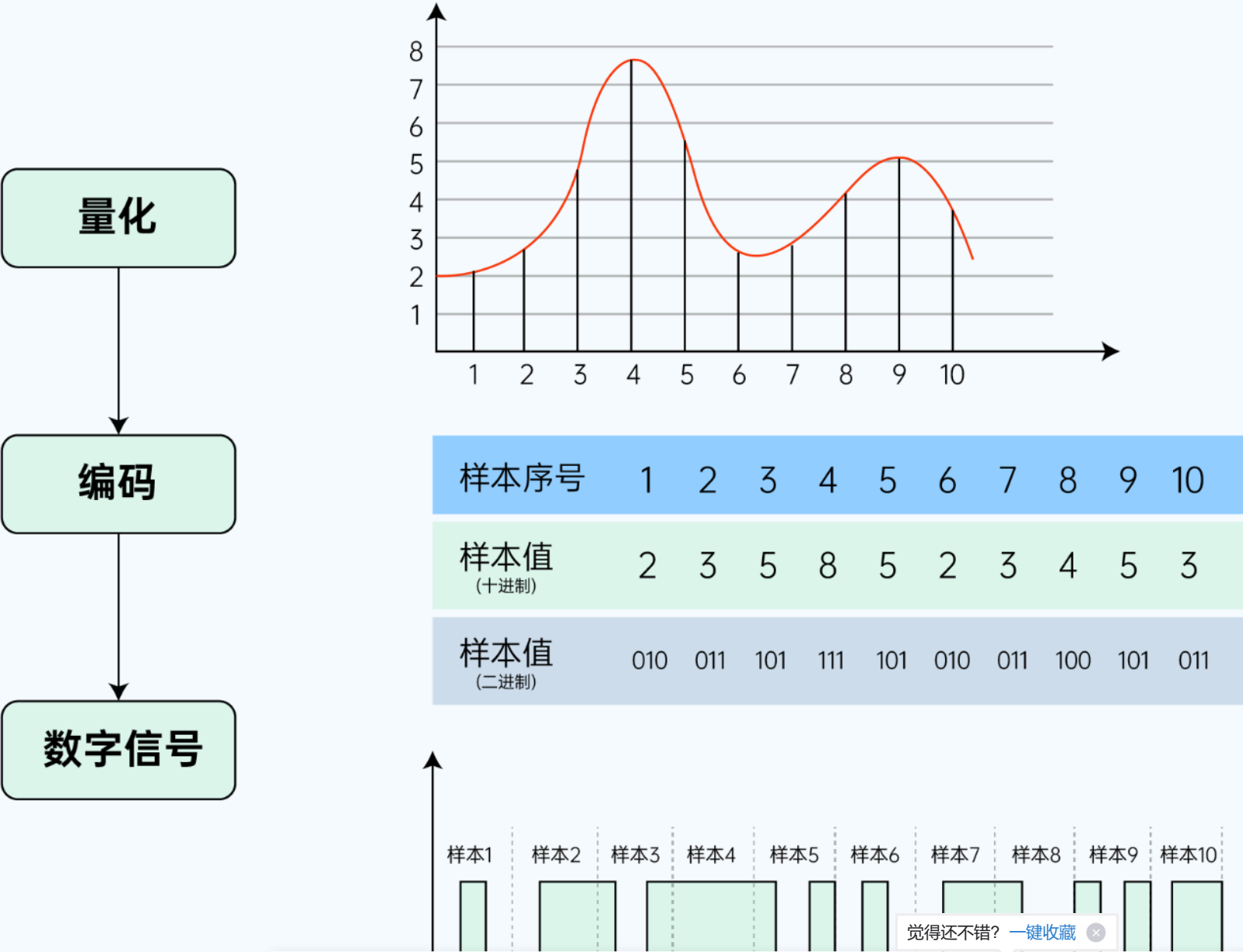
编码是整个声音数字化的最后一步，其实声音模拟信号经过采样，量化之后已经变为了数字形式，但是为了方便计算机的储存和处理，我们需要对它进行编码，以减少数据量。

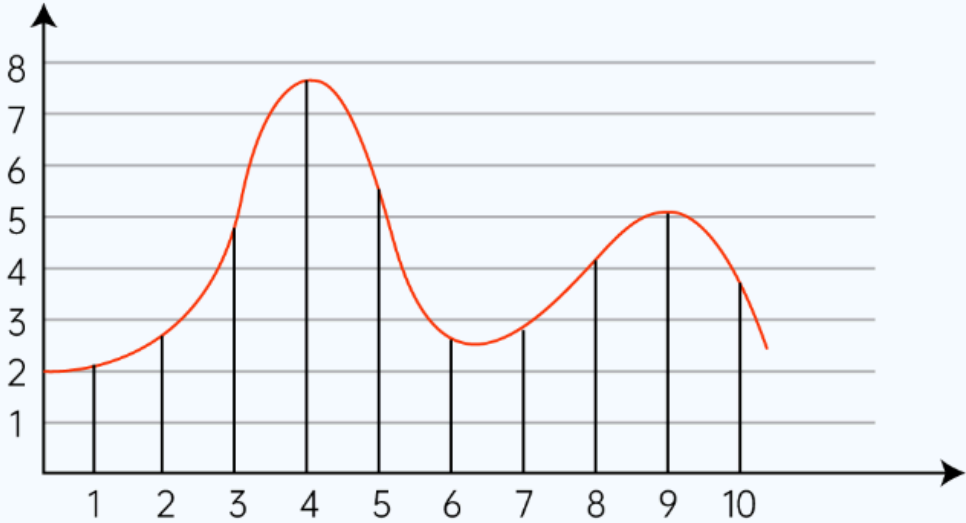
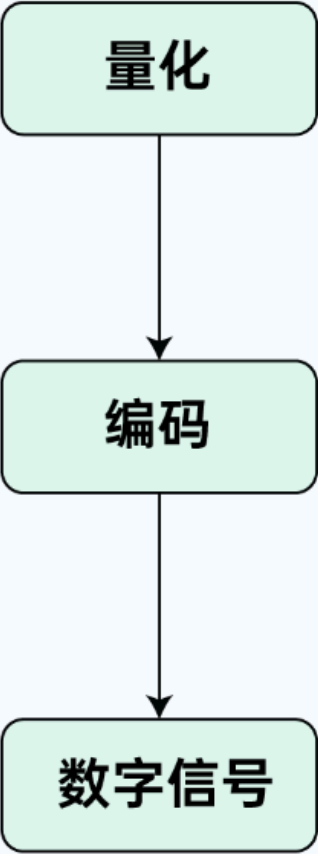
常见的音频编码格式有PCM、PDM。

经过量化后，我们得到了每个采样点的幅度值。接下来，就是音频信号数字化的最后一步，编码。编码是将每个采样点的幅度量值，转化为计算机可理解的二进制字节序列。

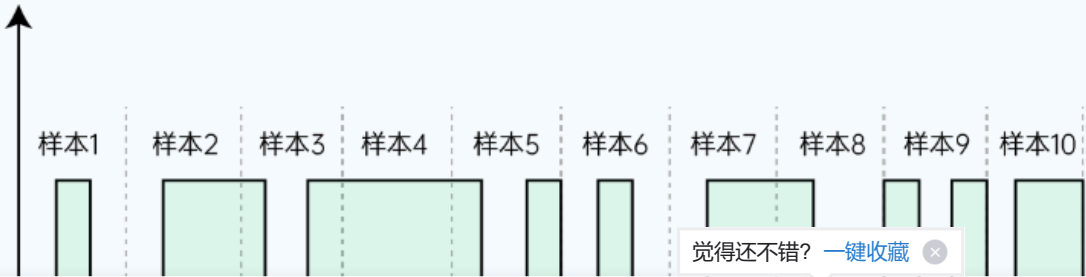
如下图所示，参照编码部分的表格，样本序号为样本采样顺序，样本值（十进制）为量化的幅度值。而样本值（二进制）即为幅度值转化的二进制字节序列，也就得到了“0”、“1”形式的二进制字节序列，也即离散的数字信号。

这里得到的，是未经压缩的音频采样数据裸流，也叫做PCM 音频数据(Pulse Code Modulation，脉冲编码调制)。实际应用中，往往还会使用其他编码 **算法** 做进一步压缩，以后的文章我们会再展开讨论。





样本序号	1	2	3	4	5	6	7	8	9	10
样本值 (十进制)	2	3	5	8	5	2	3	4	5	3
样本值 (二进制)	010	011	101	111	101	010	011	100	101	011



同理，如果处理器要向外界传达自己的“心声”，也就是放音，那么就涉及到将处理器能理解的 0 和 1 转化为外界能理解的连续变化的声音，这个过程就是将数字信号转化为模拟信号，而完成这个功能的是 DAC 芯片。

5. 音频数字信号质量三要素

1) 采样率

音频采样率，指的是单位时间内（1s）对声音信号的采样次数（参考数字化过程-采样）。常说的 44.1KHz 采样率，也即 1 秒采集了 44100 个样本。

我们前面了解到，采样率越高、采样点越多，就可以越好的表示原波形，这就是采样率的影响。

参考**奈奎斯特采样定理**：采样率 f ，必须大于原始音频信号最大振动频率 f_{max} 的 2 倍（也即 $f > 2f_{max}$ ， f_{max} 被称为奈奎斯特频率），采样结果才能用于完整重建原始音频信号；如果采样率低于 $2f_{max}$ ，那么音频采样就存在失真。

比如，要对最高频率 $f_{max}=8\text{KHz}$ 的原始音频进行采样，则采样率 f 至少为 16KHz。

对于最大频率为 f 的音频信号，当我们分别采用 f 、 $2f$ 、 $4f/3$ 的采样率进行采样时，所得到的采样结果参考下图。显然，只有当采样率为 $2f$ 时，才能有效的保留原信号特征。采样率 f 和 $3f/4$ 下得到的结果，都和原波形差别很大。

觉得还不错？ 一键收藏



一口Linux

关注

👍 25



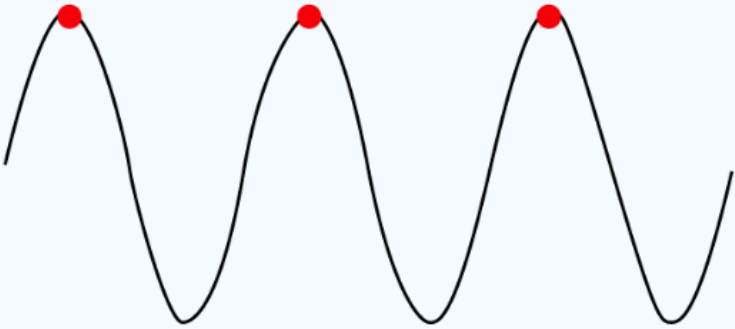
★ 44

💬 0

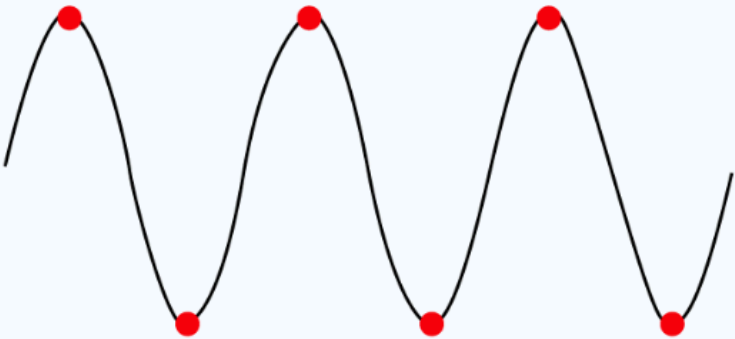
🔗 分享

💰 打赏

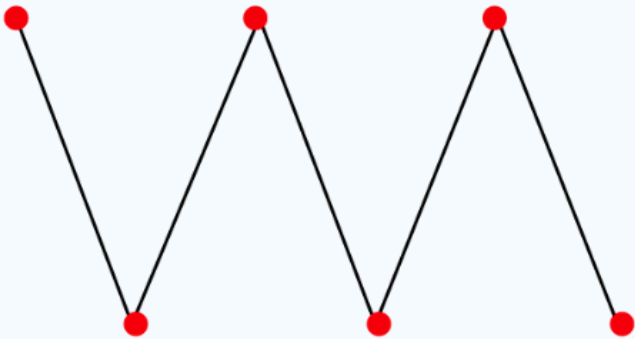




A
→
Sampled at f



B
→
Sampled at $2f$



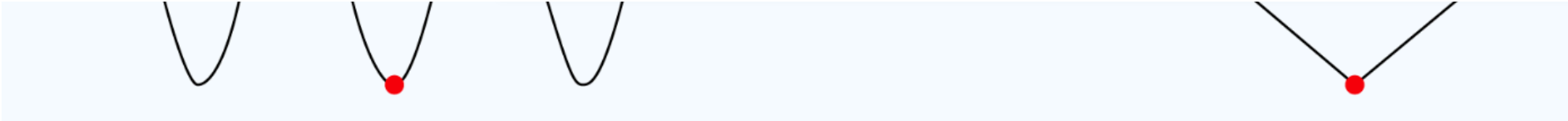
C
→



觉得还不错? 一键收藏

Linux 关注

25 44 0 分享 打赏



那么，我们需要多大的采样率？

按前面的讨论，采样率似乎越大越好，是否如此呢？理论上来说，最低采样率需要满足奈奎斯特采样定理，在该前提下，采样率越高则保留的原始音频信息越多，声音自然就越真实。但需要注意的是，采样率越高则采样得到的数据量越大，对存储和带宽的要求也就越高。在实际应用中，我们为了平衡带宽和音质，不同场景往往会有不同的选择。常见的选择如下：

采样率	描述
8KHz	在语聊、通话场景，满足基本的沟通目的，同时有效减少数据量、兼容各种传输/存储环境。人说话声音频率一般在300-700Hz之间，最大区间一般为60Hz-2000Hz，参考奈奎斯特定理，8KHz采样率完全足够
16KHz、32KHz	在保证基本沟通的基础上，进一步提升音质，同时平衡带宽、存储的压力。某些音频处理算法会要求使用32KHz的采样率
44.1KHz、48KHz	在比如在线KTV、音乐教学等场景，对音质要求比较高，可考虑进一步提升采样率。人耳可识别的声音频率范围为 20Hz ~ 20KHz，根据奈奎斯特采样定理，理论上采样率大于40KHz则完全足够。实际应用中，44.1KHz 可满足绝大多数的音视频应用场景。我们一般将 44.1kHz作为CD音质的采样标准
96KHz、192KHz	更特殊的应用，比如需要对采集的音频进行后期加工、二次处理等。96KHz、192KHz等采样率对于人耳听感来说已无明显的提升，反而会增大存储、带宽的压力，对采集/播放设备也有较高要求，RTC场景一般不考虑

当采样率从 8KHz 翻倍至 16KHz 时，听感明显变得更清晰、空灵和舒适。

此时，采样率的提升带来了明显的音质提升。而采样率从 16KHz 提升至 44.1KHz 时，实际听感却好像没有太大的变化，这是因为采样率到达一定程度后，音频质量已经比较高，再往上提升带来的优化已经很细微。

借助专业的频谱分析软件，或许可以观察到高频谱区域的能量差异，但对于人耳来说，已经很难进行区分。所以实际应用中，我们不需要一味追求高采样率，而是要综合带宽、性能、实际听感，选择合适的配置即可。


2) 采样位深/量化精度

位深度，也叫位宽，量化精度。

指的是在音频采集量化过程中，每个采样点幅度值的取值精度，一般使用bit作为单位。

常见的位宽有：8bit或者16bit。

比如，当采样位深为 8bit，则每个采样点的幅度值可以用 2

 一口Linux

关注

觉得还不错? 一键收藏

25

44

0

分享

打赏

显然，16bit 比 8bit 可存储、表示的数据更多、更精细，量化时产生的误差损失就越小。位深影响声音的解析精度、细腻程度，我们可以将其理解为声音信号的“分辨率”，位深越大，音色也越真实、生动。

采样位深选择

和采样率的选择类似，虽然理论上来说位深越大越好，但是综合带宽、存储、实际听感的考虑，我们应该为不同场景选用不同的位深。

采样率	描述
8bit	早期常用的位深精度，可满足基础的通话音质需求
16bit	被认为是达到专业音频质量的位深标准，足够完整地收录绝大多数音频场景的动态变化，适用范围广。和44.1KHz采样率一起，被作为CD音质的标准
24bit、32bit、64bit	对于使用常见播放设备（手机、普通音箱）的用户来说，32bit与16bit的感官差异很细微，音质上的提升不明显，反而带来了更大的带宽、存储压力，更不用说64bit。并不需要盲目追求

3) 声道数

我们常说的单声道、双声道，其实就是在描述一个音频信号的声道数（分别对应于声道数 1 和 2）。

声波是可以叠加的，音频的采集和播放自然也如此，我们可以同时从多个音频源采集声音，也可以分别输出到多个扬声器，声道数一般指声音采集录制时的音源数量或播放时的扬声器数量。

除了常见的声道数1、2，PC上还有4，6，8等声道的扩展。一般来说声道数越多，声音的方向感、空间感越丰富，听感也就越好。

目前很多手机厂商已经将双声道扬声器作为旗舰标配。在RTC音乐场景，越来越多的应用也开始采用双声道配置，其目的也是进一步提高听感，给用户更好的体验。

声道数的选择

实时音视频场景下，声道的选择受限于编解码器、前处理算法的能力，一般仅支持单、双声道。而双声道配置主要在语音电台、音乐直播、乐器教学、ASMR 直播等场景使用，其它场景单声道即可满足。

当然，最终能否使用哪一种声道配置，还是由实际采集、播放设备的能力决定。解码音频数据时，可以获取数据的声道数，在实际播放时，也要先获取设备属性。

如果设备支持双声道，但待播放数据是单声道的，就需要将单声道数据转成双声道数据再播放；同理，如果设备只支持单声道，但数据是双声道的，也需要将双声道数据转换成单声道数据再播放。

目前，CD音频的采样频率通常为 **44100** Hz,量化精度是 **16bit**。

6. 音频码率

数字音频的三要素不仅影响音频质量，也会影响音频存储、传输所需的空间、带宽。而实际应用场景下，音质决定用户体验、带宽决定成本，都是我们必须考虑到。

音质可能更多是主观上的感受，但带宽、空间是比较容易量

音频码率，又称为比特率，指的是单位时间内（一般为1s）所包含的音频数据量，可以通过公式计算。

- 1 | 数据传输率：数据传输率（bps） = 采样频率 × 量化位数 × 声道数。
- 2 |
- 3 | 声音信号的数据量：数据量（byte） = 数据传输率 * 持续时间 / 8。

比如采样率 44.1K Hz，位深16bit、双声道音频PCM数据，它的原始码率为：

- 1 | 原始码率 = 采样率/s × 位深/bit × 声道数 × 时长(1s)
- 2 |
- 3 | $44.1 * 1000 * 16 * 2 * 1 = 1411200 \text{ bps} = 1411.2 \text{ kbps} = 1.411 \text{ Mbps}$ （需要注意单位之间的差异和转换，b=bit）

如果一个PCM文件时长为1分钟，则传输/存储这个文件需要的数据量为：

- 1 | $1.411 \text{ Mbps} * 60\text{s} = 86.46\text{Mb}$

上述计算结果是未经压缩的、原始音频PCM数据的码率。

RTC场景下，往往还需要再使用 AAC、OPUS 等编码算法做编码压缩，进一步减小带宽、存储的压力。码率的选择也是一个综合质量和成本的博弈。

7. 噪声抑制

觉得还不错？ [一键收藏](#) ×



一口Linux

关注

👍 25



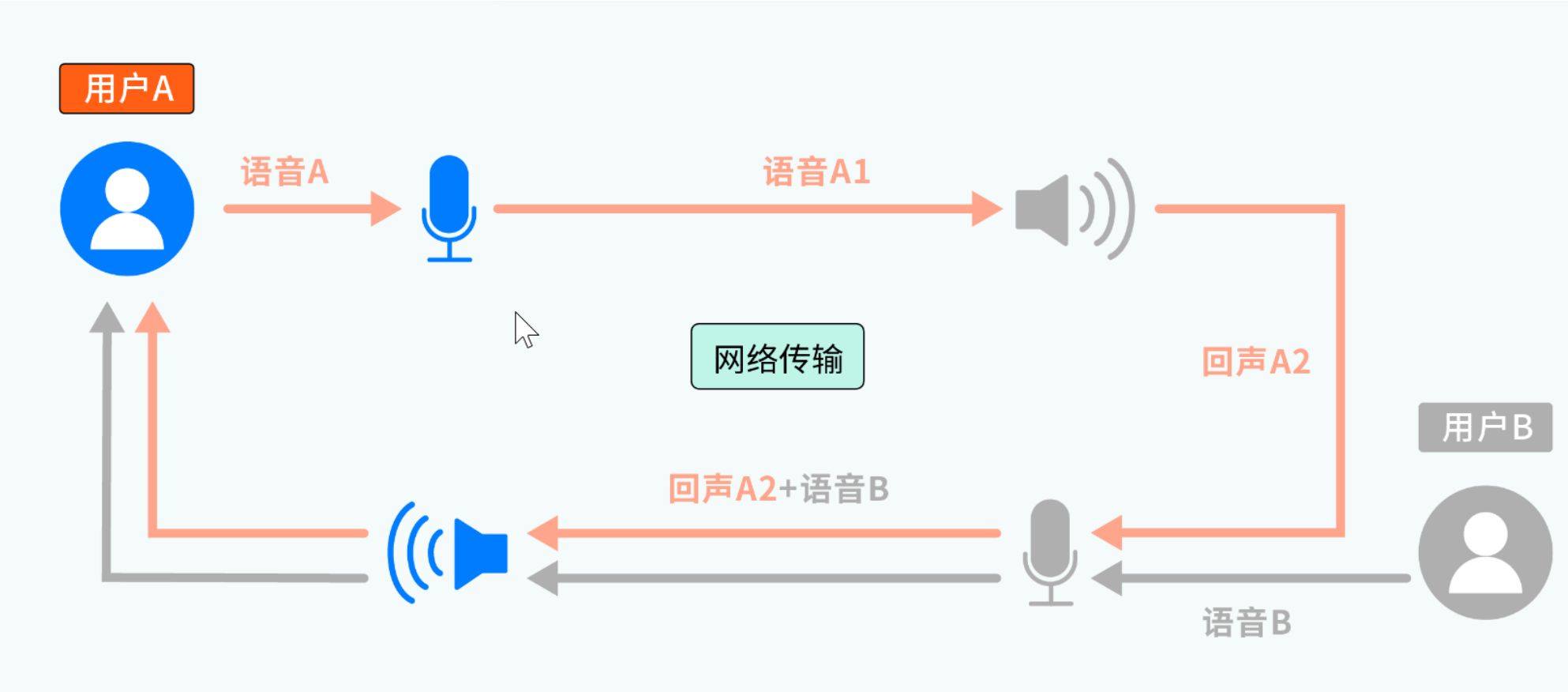
★ 44

💬 0

🔗 分享

💰 打赏





有上图所示，

- 1、某一时刻，用户 A 开始说话，产生的语音 A 被麦克风 A 采集、并通过网络传输给用户 B，成为待播放的语音 A1
- 2、语音 A1 被扬声器 B 播放后，通过直射、周围环境反射等方式，最终又被麦克风 B 采集为语音 A2（图中的回声 A2）
- 3、被回采的语音 A2，会通过网络再传输给用户 A，并通过扬声器 A 播放出来

经过上述过程，用户 A 会发现：自己刚说完一句话，过一会儿居然又听到了自己的“复述”，这就是 RTC 场景下的“回声”现象。

噪声抑制技术用于消除背景噪声，改善语音信号的信噪比和可懂度，让人和机器听得更清楚。

8. I2S

I2S(Inter-IC Sound)总线有时候也写作 IIS，I2S 是飞利浦公司提出的一种用于数字音频设备之间进行音频数据传输的总线

I2S 总线用于主控制器（譬如 ZYNQ 7010/7020）和音频

觉得还不错? 一键收藏

在I2S总线上，只能同时存在一个主设备和发送设备，主设备可以是发送设备或接收设备。

I2S是PCM的一个分支，接口定义相同。

I2S的采样率一般为44.1/48KHZ，PCM采样频率一般为8/16KHZ等。

I2S接口有4组信号：SCK(位时钟)、LRCK(帧时钟)、SDI/SDO（数据）。

LRCLK

采样时钟，也叫WS(Word Select)：字段选择线，帧时钟 (LRC)线

用于切换左右声道数据，

- 1：传输左声道的数据
- 0：表示正在传输右声道的数据

WS的频率等于采样频率，比如采样率为44.1KHz的音频，WS=44.1KHz；

SCLK/BCLK

串行时钟信号，也叫位时钟（BCLK）、CK(Serial Clock)

数字音频的每一位数据都对应有一个CK脉冲，它的频率为：2* 采样频率 * 量化位数，2代表左右两个通道数据

比如采样频率为44.1KHz、16位的立体声音频，那么

1 | $SCK=2\times44100\times16=1411200\text{Hz}=1.4112\text{MHz}$ ；

SD/DATA


SD(Serial Data)：串行数据线

用于发送或接收两个时分复用的数据通道上的数据 (仅半双工模式)，如果是全双工模式，该信号仅用于发送数据。

MCLK

MCLK（Master/System clock input）也叫做主时钟或系统时钟，音频 CODEC 芯片与主控制器之间同步用，一般是采样率的 256 倍或 384 倍。

之所以引入MCLK。这是由CODEC内部基于Delta -Sigma ($\Delta\Sigma$)的架构设计要求使然，其主要原因是因为这类的CODEC没有所谓提供码芯片提供的系统时钟。

 一口Linux

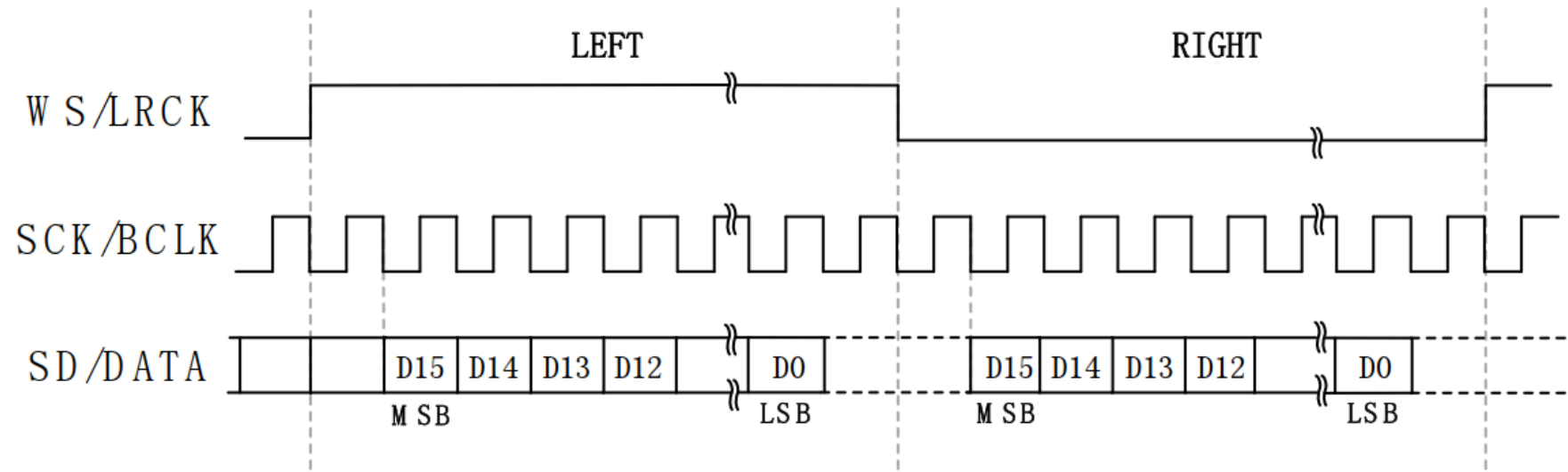
关注

觉得还不错? 一键收藏

25 44 0 分享 打赏


如果使用MCLK时钟的话，MCLK时钟频率一般为采样频率的256倍或384倍，具体参考特定器件手册。

下图为一帧立体声音频时序图



逻辑分析仪抓到的数据帧：

觉得还不错? 一键收藏

一口Linux

关注

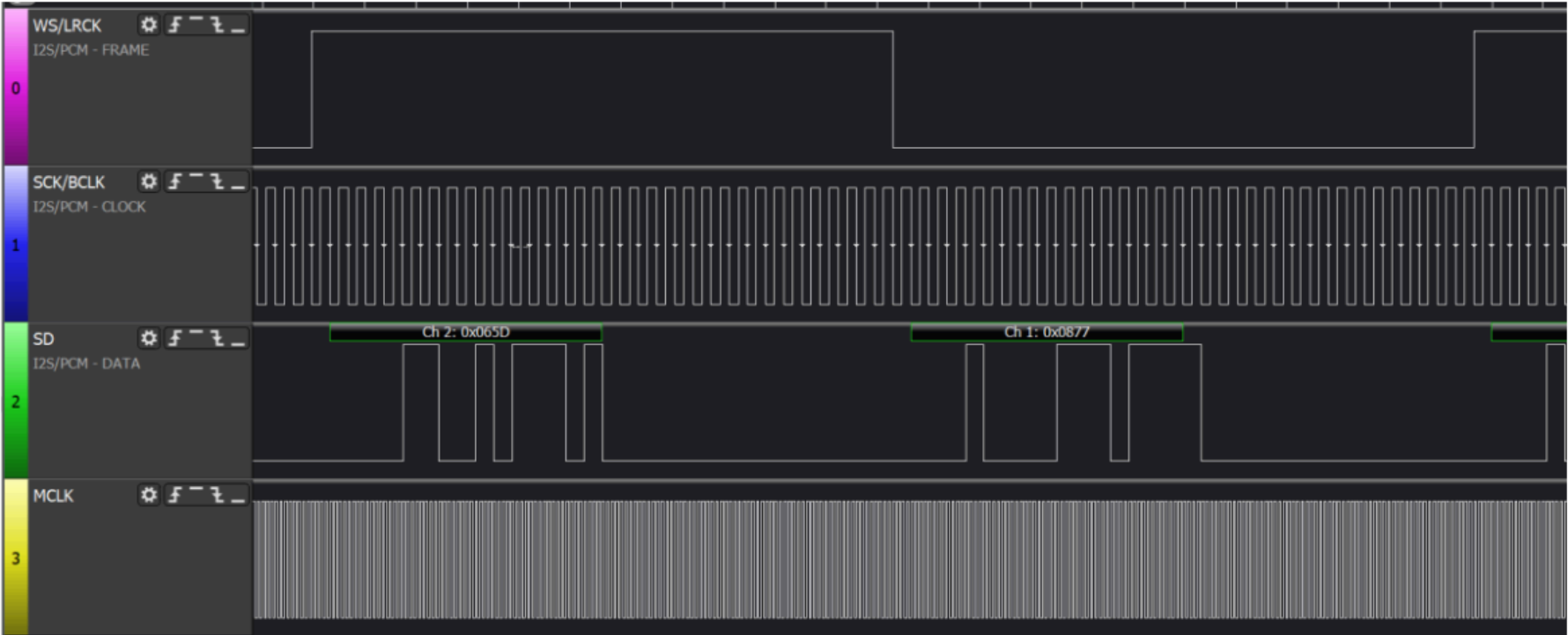
25

44

0

分享

打赏



通道 0 是 LRCK 时钟，通道 1 为 BCLK，通道 2 是 DACDATA，通道 3 是MCLK。

9. codec

处理器如果想要播放或者采集声音， 需要用到 DAC 和 ADC 这两款芯片。

那是不是买两颗 DAC 和 ADC 芯片就行了呢？

答案肯定是可以的，但是音频不单单是能出声、能听到就行。

我们往往需要听到的声音动听、录进去的语音贴近真实、可以调节音效、对声音能够进行一些处理(需要 DSP 单元)、拥有统一的标准接口，方便开发等等。


将这些针对声音的各种要求全部叠加到 DAC 和ADC 芯片上，那么就会得到一个专门用于音频的芯片，也就是音频编解码芯片，英文名字就是 **Audio CODEC**。

codec芯片举例：

1 | wolfson(欧胜) 的WM8960；顺芯的es8388、es8396；瑞芯微的rk809

觉得还不错？ 一键收藏

Codec里面包含了I2S接口、D/A、A/D、Mixer、PA（功放）

 一口Linux

关注

25

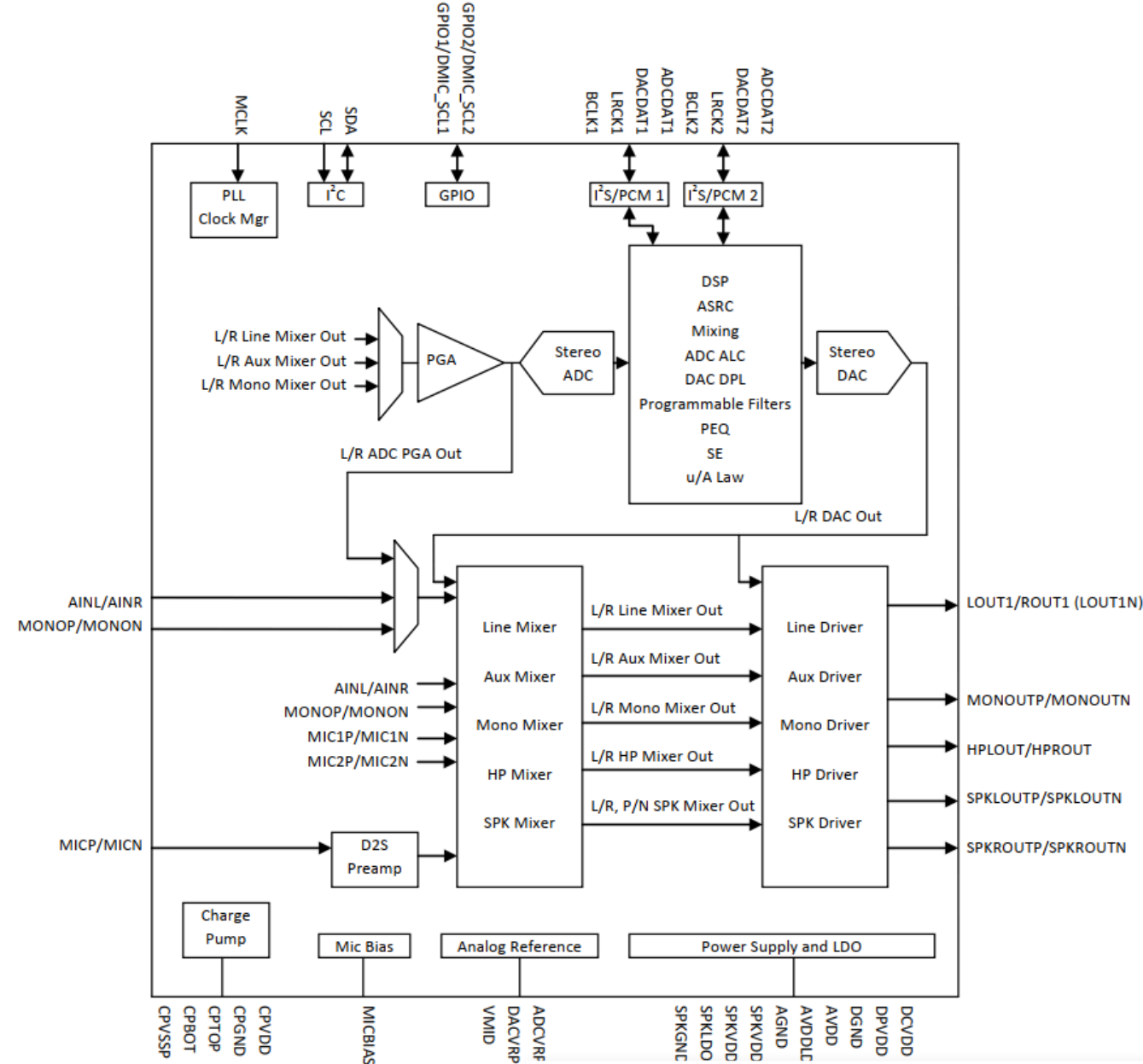
44

0

分享

打赏

下图是codec es8396芯片的模块图。



觉得还不错? 一键收藏

10. dai

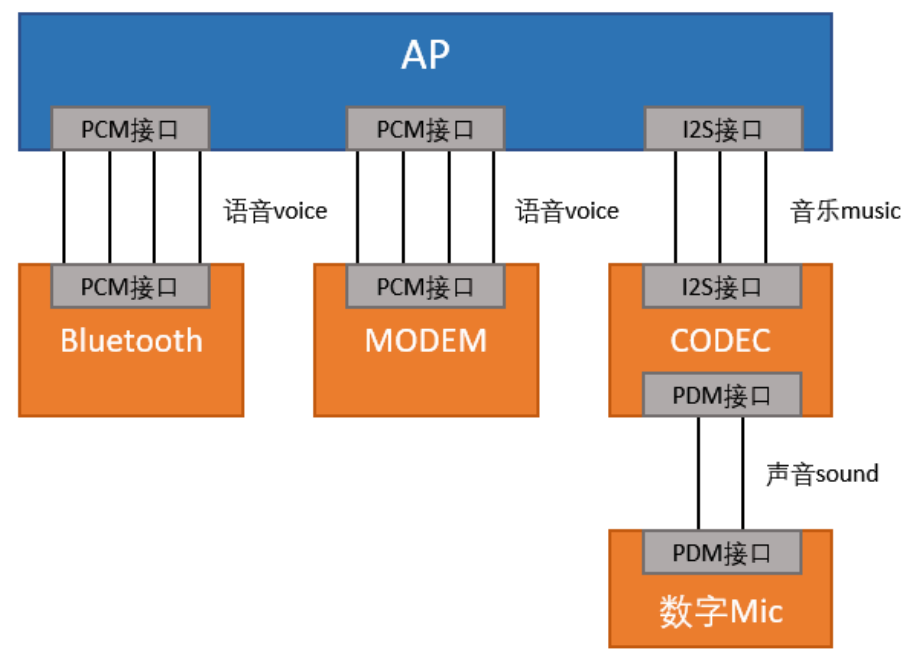
dai(digital audio interface)

数字音频接口全部是 硬件 接口，是实实在在的物理连线方式，即同一个PCB板上IC芯片和IC芯片之间的通讯协议。和音频编码格式完全是两回事。

数字音频接口有PCM、I2S、AC97、PDM；

- I2S和PCM（TDM）接口传输的数据是PCM编码格式的音频数据；
- PDM接口传输的数据是PDM编码格式的音频数据；


下图是数字音频接口硬件接线的一般场景：



 一口Linux
嵌入式Linux硬核号主！

 微信公众号 >

瑞芯微 | I2S-音频基础分享
音色是一种“感官属性”，我们利用这种“感官属性”，能区分发声的物

 一口Linux

关注

觉得还不错？ 一键收藏

嵌入式技术开发 662

 25   44  0  分享  打赏 ...