

## 1. 音频简介

经常见到这样的描述：44100HZ 16bit stereo 或者 22050HZ 8bit mono 等等。

44100HZ 16bit stereo：每秒钟有 44100 次采样，采样数据用 16 位(2 字节)记录，双声道(立体声)；

22050HZ 8bit mono：每秒钟有 22050 次采样，采样数据用 8 位(1 字节)记录，单声道；

当然也可以有 16bit 的单声道或 8bit 的立体声，等等。

采样率是指：声音信号在“模→数”转换过程中单位时间内采样的次数。采样值是指每一次采样周期内声音模拟信号的积分值。

对于单声道声音文件，采样数据为八位的短整数 (short int 00H-FFH)；

而对于双声道立体声声音文件，每次采样数据为一个 16 位的整数 (int)，高八位(左声道)和低八位(右声道)分别代表两个声道。

人对频率的识别范围是 20HZ - 20000HZ，如果每秒钟能对声音做 20000 个采样，回放时就足可以满足人耳的需求。所以 22050 的采样频率是常用的，44100 已是 CD 音质，超过 48000 的采样对人耳已经没有任何意义。这和电影的每秒 24 帧图片的道理差不多。

每个采样数据记录的是振幅，采样精度取决于储存空间的大小：

1 字节(也就是 8bit) 只能记录 256 个数，也就是只能将振幅划分成 256 个等级；

2 字节(也就是 16bit) 可以细到 65536 个数，这已是 CD 标准了；

4 字节(也就是 32bit) 能把振幅细分到 4294967296 个等级，实在是没必要了。

如果是双声道(stereo)，采样就是双份的，文件也差不多要大一倍。

这样我们就可以根据一个 wav 文件的大小、采样频率和采样大小估算出一个 wav 文件的播放长度。

譬如 "Windows XP 启动.wav" 的文件长度是 424,644 字节，它是 "22050HZ / 16bit / 立体声" 格式(这可以从其 "属性->摘要" 里看到)，

那么它的每秒的传输速率(位速,也叫比特率、取样率)是  $22050 \times 16 \times 2 = 705600(\text{bit/s})$ , 换算成字节单位就是  $705600/8 = 88200(\text{字节/秒})$ , 播放时间:  $424644(\text{总字节数}) / 88200(\text{每秒字节数}) \approx 4.8145578(\text{秒})$ 。

但是这还不够精确, 包装标准的 PCM 格式的 WAVE 文件(\*.wav)中至少带有 42 个字节的头信息, 在计算播放时间时应该将其去掉, 所以就有:  $(424644-42) / (22050 \times 16 \times 2 / 8) \approx 4.8140816(\text{秒})$ . 这样就比较精确了.

关于声音文件还有一个概念: "位速", 也有叫做比特率、取样率, 譬如上面文件的位速是 705.6kbps 或 705600bps, 其中的 b 是 bit, ps 是每秒的意思;

压缩的音频文件常常用位速来表示, 譬如达到 CD 音质的 MP3 是: 128kbps / 44100HZ.

## 2. wave 文件格式

### 2.1 概述

WAVE 文件是计算机领域最常用的数字化声音文件格式之一，它是微软专门为 Windows 系统定义的波形文件格式 (Waveform Audio)，由于其扩展名为 "\*.wav"。

WAVE 是录音时用的标准的 WINDOWS 文件格式，文件的扩展名为 "WAV"，数据本身的格式为 PCM 或压缩型。

WAV 文件格式是一种由微软和 IBM 联合开发的用于音频数字存储的标准，它采用 RIFF 文件格式结构，非常接近于 AIFF 和 IFF 格式。符合 PIFF Resource Interchange File Format 规范。所有的 WAV 都有一个文件头，这个文件头音频流的编码参数。

WAV 对音频流的编码没有硬性规定，除了 PCM 之外，还有几乎所有支持 ACM 规范的编码都可以为 WAV 的音频流进行编码。

多媒体应用中使用了多种数据，包括位图、音频数据、视频数据以及外围设备控制信息等。RIFF 为存储这些类型的数据提供了一种方法，RIFF 文件所包含的数据类型由该文件的扩展名来标识，能以 RIFF 文件存储的数据包括：

音频视频交错格式数据 (.AVI)、波形格式数据 (.WAV)、位图格式数据 (.RDI)、MIDI 格式数据 (.RMI)、调色板格式 (.PAL)、多媒体电影 (.RMN)、动画光标 (.ANI)、其它 RIFF 文件 (.BND)。

wave 文件有很多不同的压缩格式，所以，正确而详细地了解各种 WAVE 文件的内部结构是成功完成压缩和解压缩的基础，也是生成特有音频压缩格式文件的前提。

最基本的 WAVE 文件是 PCM（脉冲编码调制）格式的，这种文件直接存储采样的声音数据没有经过任何的压缩，是声卡直接支持的数据格式，要让声卡正确播放其它被压缩的声音数据，就应该先把压缩的数据解压缩成 PCM 格式，然后再让声卡来播放。

2.2 Wave 文件的内部结构

注：由于 WAV 格式源自 Windows/Intel 环境，因而采用 Little-Endian 字节顺序进行存储。

WAVE 文件是以 RIFF(Resource Interchange File Format, "资源交互文件格式")格式来组织内部结构的。

RIFF 文件结构可以看作是树状结构，其基本构成是称为"块"(Chunk)的单元，最顶端是一个"RIFF"块，下面的每个块有"类型块标识(可选)"、"标志符"、"数据大小"及"数据"等项所组成。块的结构如表 1 所示：

名称	Size	备注
块标志符	4	4 个小写字符(如

		"fmt ", "fact", "data" 等)
数据大小	4	DWORD 类型,表示 后接数据的大小(N Bytes)
数据	N	本块中正式数据部 分

表 1：基本 chunk 的内部结构

上面说到的“类型块标识”只在部分 chunk 中用到，如 "WAVE" chunk 中，这时表示下面嵌套有别的 chunk。

当使用了 "类型块标识" 时，该 chunk 就没有别的项（如块标志符，数据大小等），它只作为文件读取时的一个标识。先找到这个“类型块标识”，再以它为起点读取它下面嵌套的其它 chunk。

每个文件最前端写入的是 RIFF 块，每个文件只有一个 RIFF 块。从 Wave 文件格式详细说明 中可以看到这一点。

非 PCM 格式的文件会至少多加入一个 "fact" 块，它用来记录数据(注意是数据而不是文件)解压缩后的大小。这个 "fact" 块一般加在 "data" 块的前面。

WAVE 文件是由若干个 Chunk 组成的。按照在文件中的出现位置包括：RIFF  
WAVE Chunk, Format Chunk, Fact Chunk(可选), Data Chunk。具体  
见下图：

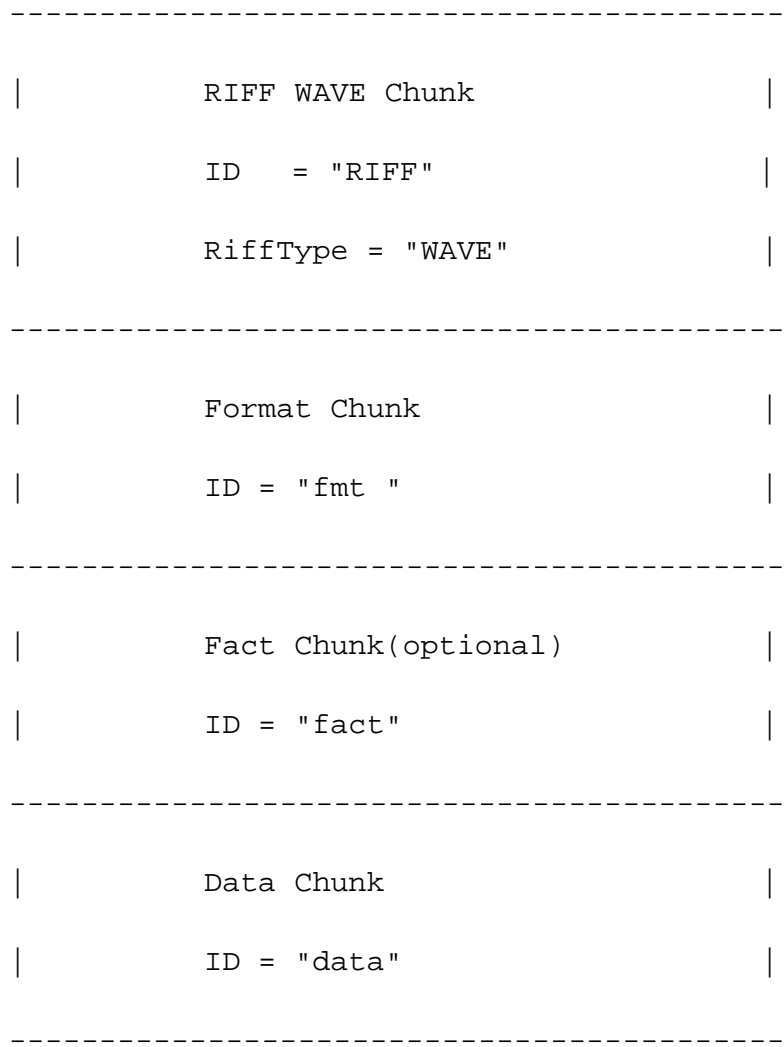


图 wav 格式包含 Chunk 示例

**Fact Chunk**



图 Fact Chunk

2.3 wave 文件格式详细说明

别名	字节数	类型	注释
ckid	4	char	"RIFF" 标志，大写
cksize	4	int32	文件长度。这个长度不包括 "RIFF"标志 和



文件长度 本身所占字

节，下面的

子块大小也是这样。

fcc type	4	char	"WAVE" 类型块标识，大 写。
----------	---	------	----------------------

ckid	4	char	表示"fmt" chunk 的开始。 此块中包括文
------	---	------	------------------------------

件内部格式信息。小写，  
最后一个

字符是空格。

cksize	4	int32	文件内部格式信息数据的大 小。
--------	---	-------	--------------------

FormatTag	2	int16	音频数据的编码方式。1 表 示是 PCM 编码
-----------	---	-------	----------------------------

Channels	2	int16	声道数，单声道为 1，双声 道为 2
----------	---	-------	-----------------------

SamplesPerSec	4	int32	采样率(每秒样本数)，比如 44100 等
---------------	---	-------	--------------------------

BytesPerSec	4	int32	音频数据传送速率，单位是 字节。其值为
-------------	---	-------	------------------------

采样率×每次采样大小。

播放软件

利用此值可以估计缓冲

区的大小。

`BlockAlign`     2     `int16`     每次采样的大小 = 采样精度\*声道数/8(单位是字节)；这也是字节对齐的最小单位，譬如

16bit 立体声在这里的值是 4 字节。播放软件需要

一次处理多个该值大小的字节数

据，以便将其值用于缓冲区的调整。

`BitsPerSample` 2     `int16`     每个声道的采样精度；譬如

16bit 在这里的值就是 16。如果有多个声道，则

每个声道的采样精度

大小都一样的。

[cbsize]	2	int16	[可选]附加数据的大小。
[...]	x		
[ckid]	4	char	"fact".
[cksize]	4	int32	"fact" chunk data size.
[fact data]	4	int32	解压后的音频数据的大小 (Bytes).
ckid	4	char	表示 "data" chunk 的开始。此块中包含

音频数据。小写。

cksize	4	int32	音频数据的长度
.....			文件声音信息数据(真正声音存储部分)
[.....]			其它 chunk

2.4 Windows 平台上 WAVEFORMAT 结构的认识

PCM 和非 PCM 的主要区别是声音数据的组织不同，这些区别可以通过两者的 WAVEFORMAT 结构来区分。

下面以 PCM 和 IMA-ADPCM 来进行对比。

WAVE 的基本结构 WAVEFORMATEX 结构定义如下：

```
1 typedef struct
2 {
3     WORD    wFormatTag; //编码格式，包括 WAVE_FORMAT_PCM，
WAVEFORMAT_ADPCM 等
4     WORD    nChannels; //声道数，单声道为 1，双声道为 2;
5
6     DWORD    nSamplesPerSec; //采样频率;
7
8     DWORD    nAvgBytesperSec; //每秒的数据量;
9
10    WORD    nBlockAlign; //块对齐;
11
12    WORD    wBitsPerSample; //WAVE 文件的采样大小;
13
14    WORD    cbSize; // The count in bytes of the size of extra
15                  // information(after cbSize). PCM 中忽略此值
16 } WAVEFORMATEX;
```

IMAADPCMWAVEFORMAT 结构定义如下：

```
1 Typedef struct
2 {
3     WAVEFORMATEX    wfmt;
4
5     WORD    nSamplesPerBlock;
6
7 } IMAADPCMWAVEFORMAT;
```

IMA-ADPCM 中的 `wfmt->cbsize` 不能忽略，一般取值为 2，表示此类型的 `WAVEFORMAT` 比一般的 `WAVEFORMAT` 多出 2 个字节。这两个字节也就是 `nSamplesPerBlock`。

### "fact" chunk 的内部组织

在非 PCM 格式的文件中，一般会在 `WAVEFORMAT` 结构后面加入一个 "fact" chunk，结构如下：

[view source](#)[print?](#)

```
1  typedef struct{
2
3      char[4]; // “fact” 字符串
4
5      DWORD chunksize;
6
7      DWORD datafactsize; // 音频数据转换为 PCM 格式后的大小。
8
9  } factchunk;
```

`datafactsize` 是这个 chunk 中最重要的数据，如果这是某种压缩格式的声音文件，那么从这里就可以知道他解压缩后的大小。对于解压时的计算会有很大的好处！

### 2.5 "data" chunk 的内部组织

从 "data" chunk 的第 9 个字节开始, 存储的就是声音信息的数据了, (前八个字节存储的是标志符 "data" 和后接数据大小 size(DWORD))。这些数据可以是压缩的, 也可以是没有压缩的。

### 3. PCM 数据格式

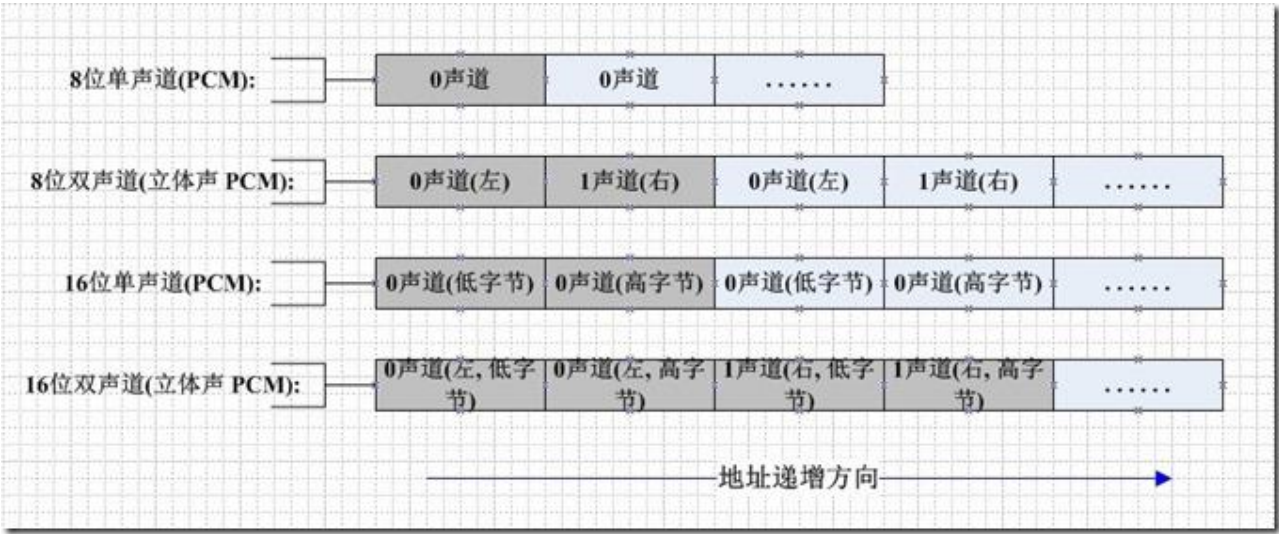
PCM(Pulse Code Modulation)也被称为 脉码编码调制。PCM 中的声音数据没有被压缩, 如果是单声道的文件, 采样数据按时间的先后顺序依次存入。(它的基本组织单位是 BYTE(8bit)或 WORD(16bit))

一般情况下, 一帧 PCM 是由 2048 次采样组成的(参

考 <http://discussion.forum.nokia.com/forum/showthread.php?>

129458-请问 PCM 格式的音频流, 每次读入或输出的块的大小是必须固定为 4096B 么&s=e79e9dd1707157281e3725a163844c49 )。

如果是双声道的文件, 采样数据按时间先后顺序交叉地存入。如图所示:



PCM 的每个样本值包含在一个整数  $i$  中,  $i$  的长度为容纳指定样本长度所需的最小字节数。

首先存储低有效字节, 表示样本幅度的位放在  $i$  的高有效位上, 剩下的位置为 0,

这样 8 位和 16 位的 PCM 波形样本的数据格式如下所示。

样本大小	数据格式	最小值	最大值
8 位 PCM	unsigned int	0	225
16 位 PCM	int	-32767	32767

参考资料:

[1]<http://redsoft.ycool.com/post.2232742.html>

[2][http://dev.firnow.com/course/3\\_program/hb/hbxi/20100803/518348.html](http://dev.firnow.com/course/3_program/hb/hbxi/20100803/518348.html)

[3]<http://hi.baidu.com/kindyb/blog/item/0a314f8859489c93a4c27297.html>

[4]<http://hi.baidu.com/kindyb/blog/item/353f4813df8799055aaf5397.html>

[5]<http://hi.baidu.com/kindyb/blog/item/2f31daa93f5ed4fb1e17a291.html>

[6]<http://hi.baidu.com/bigbigant/blog/item/7b91aa01e46dd4021d958317.html>