# 1.[WAVE 文件格式分析](http://www.cnblogs.com/liyiwen/archive/2010/04/19/1715715.html)

<http://www.cnblogs.com/liyiwen/archive/2010/04/19/1715715.html>

WAVE 文件作为多媒体中使用的声音波形文件格式之一，它是以RIFF（Resource Interchange File Format）格式为标准的。每个WAVE文件的头四个字节便是“RIFF”。WAVE 文件由文件头和数据体两大部分组成。其中文件头又分为 RIFF／WAV 文件标识段和声音数据格式说明段两部分。WAVE文件各部分内容及格式见后文。

常见的声音文件主要有两种，分别对应于单声道（11.025KHz 采样率、8Bit 的采样值）和双声道（44.1KHz 采样率、16Bit 的采样值）。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），高八位和低八位分别代表左右两个声道。

WAVE 文件数据块包含以脉冲编码调制（PCM）格式表示的样本。WAVE 文件是由样本组织而成的。在单声道 WAVE 文件中，声道0代表左声道，声道1代表右声道。在多声道WAVE文件中，样本是交替出现的。

WAVE 文件除了前面一小段文件头对数据组织进行说明之外，Data 块就是声音的原始采样数据，WAVE 文件虽然可以压缩，但一般都使用不压缩的格式。44.1KHz 采样率、16Bit的分辨率、双声道，所以WAVE可以保存音质要求非常高的声音文件，CD 采用的也是这种格式，声音方面的专家或是音乐发烧友们应该非常熟悉。但这种文件的体积也非常大，以 44.1KHz 16bit 双声道的数据为例，一分钟的声音数据量为：4100\*2byte\*2channel\*60s/1024/1024=10.09M 。所以不合适在网上传送。

下面我们具体地分析 WAVE 文件的格式

|  |  |  |  |
| --- | --- | --- | --- |
| **endian** | **field name** | **Size** |  |
| big | ChunkID | 4 | 文件头标识，一般就是" RIFF" 四个字母 |
| little | ChunkSize | 4 | 整个数据文件的大小，不包括上面ID和Size本身 |
| big | Format | 4 | 一般就是" WAVE" 四个字母 |
| big | SubChunk1ID | 4 | 格式说明块，本字段一般就是"fmt " |
| little | SubChunk1Size | 4 | 本数据块的大小，不包括ID和Size字段本身 |
| little | AudioFormat | 2 | 音频的格式说明 |
| little | NumChannels | 2 | 声道数 |
| little | SampleRate | 4 | 采样率 |
| little | ByteRate | 4 | 比特率，每秒所需要的字节数 |
| little | BlockAlign | 2 | 数据块对齐单元 |
| little | BitsPerSample | 2 | 采样时模数转换的分辨率 |
| big | SubChunk2ID | 4 | 真正的声音数据块，本字段一般是"data" |
| little | SubChunk2Size | 4 | 本数据块的大小，不包括ID和Size字段本身 |
| little | Data | N | 音频的采样数据 |

以下是对各个字段的详细解说：

|  |  |  |
| --- | --- | --- |
| ChunkID | 4bytes | ASCII 码表示的“RIFF”。（0x52494646） |
| ChunkSize | 4bytes | 36+SubChunk2Size，或是  4 + ( 8 + SubChunk1Size ) + ( 8 + SubChunk2Size )，  这是整个数据块的大小（不包括ChunkID和ChunkSize的大小） |
| Format | 4bytes | ASCII 码表示的“WAVE”。（0x57415645） |
|  |  |  |
| SubChunk1ID |  | 新的数据块（格式信息说明块）  ASCII 码表示的“fmt ”——最后是一个空格。（0x666d7420） |
| SubChunk1Size | 4bytes | 本块数据的大小（对于PCM，值为16）。 |
| AudioFormat | 2bytes | PCM = 1 （比如，线性采样），如果是其它值的话，则可能是一些压缩形式 |
| NumChannels | 2bytes | 1 => 单声道  |  2 => 双声道 |
| SampleRate | 4bytes | 采样率，如 8000，44100 等值 |
| ByteRate | 4bytes | 等于： SampleRate \* numChannels \* BitsPerSample / 8 |
| BlockAlign | 2bytes | 等于：NumChannels \* BitsPerSample / 8 |
| BitsPerSample | 2bytes | 采样分辨率，也就是每个样本用几位来表示，一般是 8bits 或是 16bits |
|  |  |  |
| SubChunk2ID | 4bytes | 新数据块，真正的声音数据  ASCII 码表示的“data ”——最后是一个空格。（0x64617461） |
| SubChunk2Size | 4bytes | 数据大小，即，其后跟着的采样数据的大小。 |
| Data | N bytes | 真正的声音数据 |

对于Data块，根据声道数和采样率的不同情况，布局如下（每列代表8bits）：

1. 8 Bit 单声道：

|  |  |
| --- | --- |
| 采样1 | 采样2 |
| 数据1 | 数据2 |

2. 8 Bit 双声道

|  |  |  |  |
| --- | --- | --- | --- |
| 采样1 |  | 采样2 |  |
| 声道1数据1 | 声道2数据1 | 声道1数据2 | 声道2数据2 |

1. 16 Bit 单声道：

|  |  |  |  |
| --- | --- | --- | --- |
| 采样1 |  | 采样2 |  |
| 数据1低字节 | 数据1高字节 | 数据1低字节 | 数据1高字节 |

2. 16 Bit 双声道

|  |  |  |  |
| --- | --- | --- | --- |
| 采样1 |  |  |  |
| 声道1数据1低字节 | 声道1数据1高字节 | 声道2数据1低字节 | 声道2数据1高字节 |
| 采样2 |  |  |  |
| 声道1数据2低字节 | 声道1数据2高字节 | 声道2数据2低字节 | 声道2数据2高字节 |

下面我们看一个具体的例子，声音文件如下：

52 49 46 46 24 08 00 00 57 41 56 45

66 6d 74 20 10 00 00 00 01 00 02 00

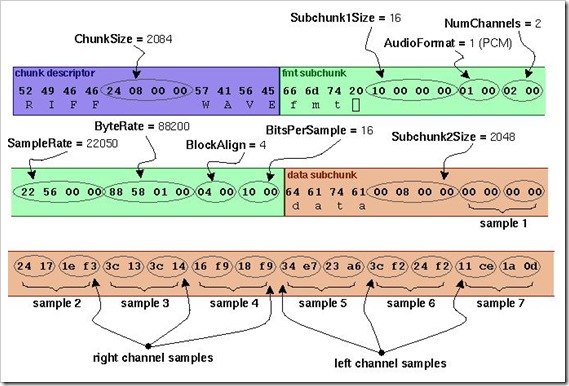
22 56 00 00 88 58 01 00 04 00 10 00

64 61 74 61 00 08 00 00 00 00 00 00

24 17 1e f3 3c 13 3c 14 16 f9 18 f9

34 e7 23 a6 3c f2 24 f2 11 ce 1a 0d

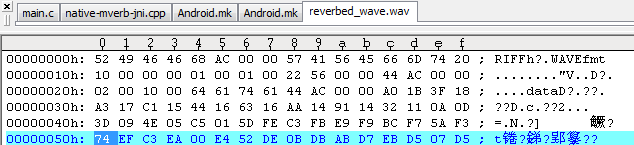
对应的分析如下图所示：

[](http://images.cnblogs.com/cnblogs_com/liyiwen/WindowsLiveWriter/WAVE_133DA/untitled.jpg)

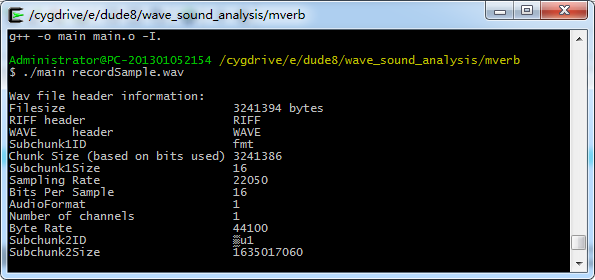
# 2.wave 文件格式的理解

针对Wave文件的数据部分的分析：

下图为声音为1秒的Wave文件头部。可以看出RIF WAV fmt等标识



此sample为采样频率为22050，16位，单声道的音频文件。



采样频率和声道信息就可以计算出相应的wave文件大小

44header + 1second \* 22050 freq/sec \*2 bytes = 44144 bytes.

如下图所示：

1. 对于16位的wave文件，每两个字节代表一次采样。采样频率越高，声音还原的越清楚。
2. 声音的音调如：人说话声，乐器声。各种音调的产生是由于物体振动频率不同产生的。
3. 物体的震动频率与wave文件中的采样频率无关。但是采样频率高，可以还原物体的震动频率，即可以还原更好的声音。
4. 一个wave文件中，连续存放的采样内容可以构成完整的声音文件或音乐。所以如果要保持音乐和声音文件的完整性，需要在固定采样频率下，增加采样时间。足够长的时间才能将现实世界中的音乐频率记录下来。
5. wave中对于16位的采样，每2个字节记录一个采样的点。这个字节的内容记录的是当前采样的声音的振幅。声音的振幅决定了记录的声音的音量大小。所以可以通过调整每2个字节的数值来改变记录声音的音量。
6. 声音的音调变化只能通过更改声音的频率才能实现。



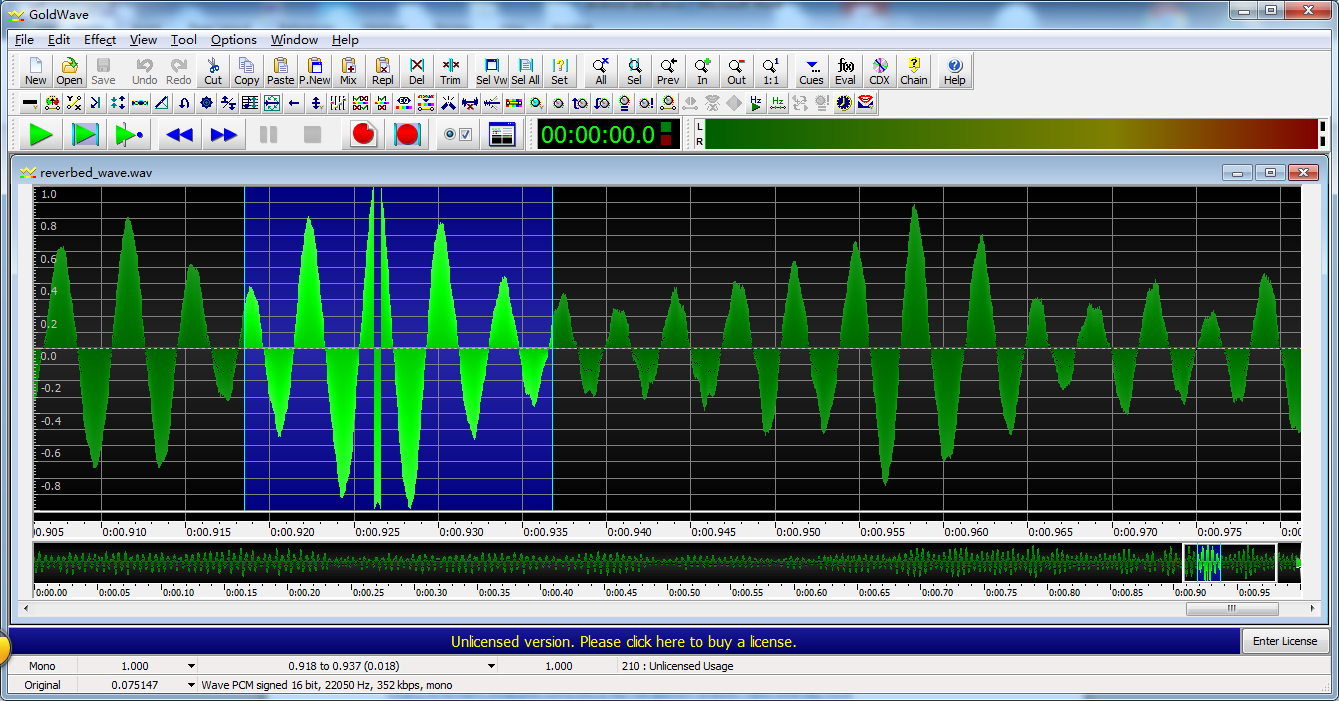
1.文件reverbed\_wave.wav 的波形如下：

当wave文件进行Mverb混响的时候，中间有noise，其中的原因是下图高亮部分。

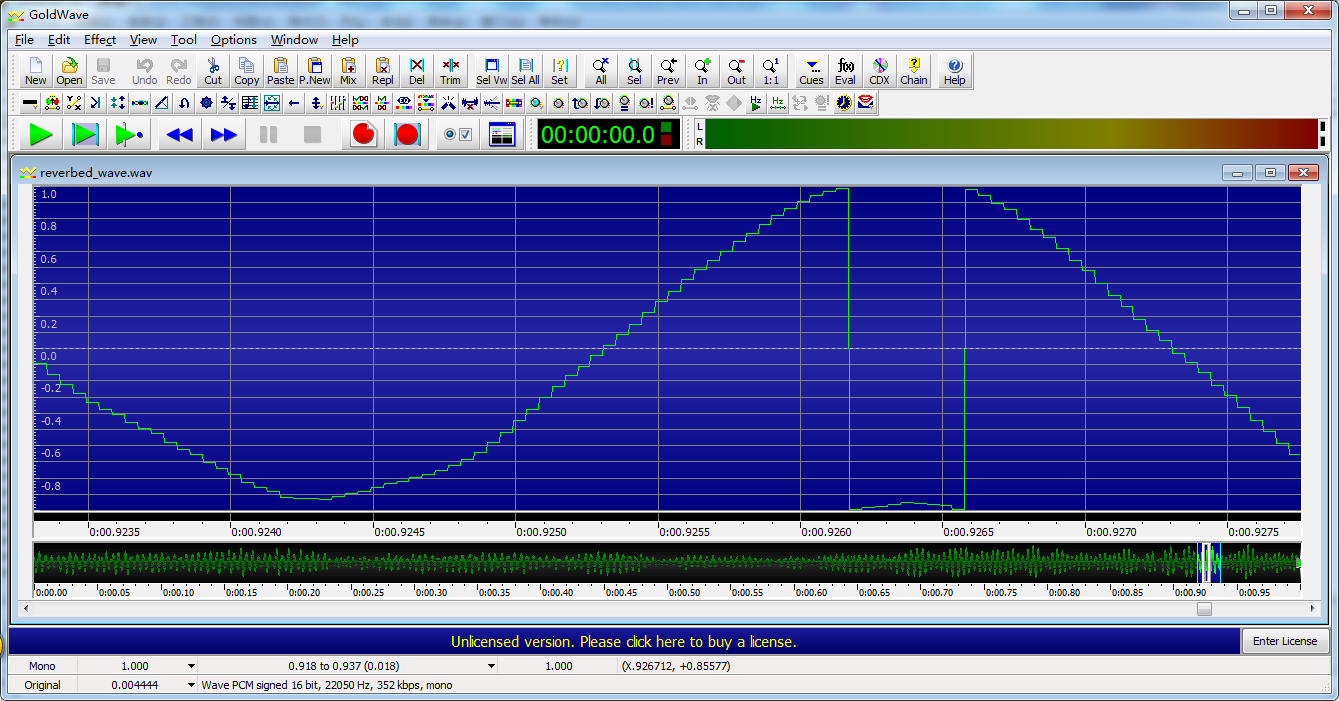
在使用GoldWave查看音频波形的时候，对于正常的音频波形应该是连续的类似sin曲线。

但是在有noise的音频波形中，在波形中突然出现振幅反向跳变。这样导致声音不正常。

在925ms到930ms的时候，出现振幅跳变。



局部放大：



从波形上可以看出来，在GoldWave中最大值达到1.0后，方向从负最大值开始减小。

可以从中看出应该是波形的振幅正向产生了溢出。