

enzo26的ChinaUnix博客

暂无签名

首页 | 博文目录 | 关于我



enzo26

博客访问：62761
博文数量：119
博客积分：554
博客等级：中士
技术积分：1753
用户组：普通用户
注册时间：2010-03-04 12:49

加关注

短消息

论坛

加好友

文章分类

全部博文 (123)

音频/视频 (4)

uboot (1)

链接 (1)

python (2)

linux内核 (31)

理财 (4)

ARM (6)

调试技术 (14)

硬件相关 (7)

AT91SAM (5)

linux (15)

emacs (9)

s3c2440/s3c2416 (15)

单片机 (5)

未分配的博文 (4)

文章存档

2013年 (97)

2012年 (22)

我的朋友



tekkaman



zsc_eric



crowsy00

最近访客

WAV文件格式分析

2013-11-05 11:34:03

分类： 嵌入式

一、综述

WAVE文件作为多媒体中使用的声波文件格式之一，它是以RIFF格式为标准的。

RIFF是英文Resource Interchange File Format的缩写，每个WAVE文件的头四个字节便是“RIFF”。

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：

RIFF WAVE Chunk
ID = 'RIFF'
RiffType = 'WAVE'
Format Chunk
ID = 'fmt '
Fact Chunk(optional)
ID = 'fact'
Data Chunk
ID = 'data'

图1 Wav格式包含Chunk示例

其中除了Fact Chunk外，其他三个Chunk是必须的。每个Chunk有各自的ID，位于Chunk最开始位置，作为标示，而且均为4个字节。并且紧跟在ID后面的是Chunk大小（去除ID和Size所占的字节数后剩下的其他字节数目），4个字节表示，低字节表示数值低位，高字节表示数值高位。下面具体介绍各个Chunk内容。

PS：

所有数值表示均为低字节表示低位，高字节表示高位。

二、具体介绍

RIFF WAVE Chunk

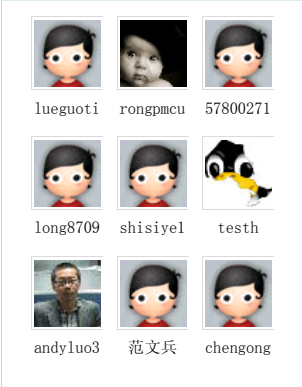
所占字节数	具体内容
ID	4 Bytes 'RIFF'
Size	4 Bytes
Type	4 Bytes 'WAVE'

图2 RIFF WAVE Chunk

以' FIFF' 作为标示，然后紧跟着为size字段，该size是整个wav文件大小减去ID和Size所占用的字节数，即FileLen - 8 = Size。然后是Type字段，为' WAVE'，表示是wav文件。

结构定义如下：

```
struct RIFF_HEADER
{
    char szRiffID[4]; // 'R','I','F','F'
    DWORD dwRiffSize;
```



订阅

- 推荐博文
- 分布式文件系统FastDFS设计原...
 - Node.js开发手册（四）-Redis...
 - P2P学习点滴录（一）
 - 解决rsyslog无法推送日志的问...
 - Linux spi驱动分析(三)——sp...

- 热词专题
- Uclinux下LED驱动
 - Fedora18下安装mysql (linux)...
 - CF0
 - xp系统用户过多
 - 以ISO文件安装Fedora 18的Gru...

```
char szRiffFormat[4]; // 'W','A','V','E'
```

```
};
```

Format Chunk

	字节数	具体内容
ID	4 Bytes	'fmt '
Size	4 Bytes	数值为16或18, 18则最后又附加信息
FormatTag	2 Bytes	编码方式, 一般为0x0001
Channels	2 Bytes	声道数目, 1--单声道; 2--双声道
SamplesPerSec	4 Bytes	采样频率
AvgBytesPerSec	4 Bytes	每秒所需字节数
BlockAlign	2 Bytes	数据块对齐单位(每个采样需要的字节数)
BitsPerSample	2 Bytes	每个采样需要的bit数
	2 Bytes	附加信息(可选, 通过Size来判断有无)

图3 Format Chunk

以 'fmt ' 作为标示。一般情况下Size为16, 此时最后附加信息没有; 如果为18则最后多了2个字节的附加信息。主要由一些软件制成的wav格式中含有该2个字节的附加信息。

```
结构定义如下:

struct WAVE_FORMAT
{
    WORD wFormatTag;
    WORD wChannels;
    DWORD dwSamplesPerSec;
    DWORD dwAvgBytesPerSec;
    WORD wBlockAlign;
    WORD wBitsPerSample;
};

struct FMT_BLOCK
{
    char szFmtID[4]; // 'f','m','t',' '
    DWORD dwFmtSize;
    WAVE_FORMAT wavFormat;
};
```

补充头文件样例说明:

```
87654321 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
1 00000000: 5249 4646 3ec1 0400 5741 5645 666d 7420 RIFF>...WAVEfmt
2 00000010: 1000 0000 0100 0200 44ac 0000 10b1 0200 .....D.....
3 00000020: 0400 1000 6461 7461 84b0 0400 d152 0000 ...data....R..
4 00000030: a14a 0000 ec37 0000 a439 0000 573a 0000 .J...7...9..W:..
5 00000040: 862e 0000 3d15 0000 0109 0000 b827 0000 ....=.....'..
6 00000050: 5b3d 0000 b025 0000 d713 0000 8319 0000 [=...%.
7 00000060: 561f 0000 4e16 0000 0fee 0000 ccc7 0000 V...N.....
8 00000070: 80c8 0000 34d1 0000 12de 0000 bffb 0000 ....4.....
9 00000080: 790a 0000 a412 0000 221b 0000 9004 0000 y.....".
10 00000090: 85f9 0000 4c0c 0000 56fd 0000 d6ed 0000 ....L...V.....
```

首先是一串“52 49 46 46”这个是Ascii字符“RIFF”，这部分是固定格式，表明这是一个WAVE文件头。

然后是“3ec1 0400”，这个是我这个WAV文件的数据大小，记住这个大小是包括头文件的一部分的，包括除了前面8个字节的的所有字节，也就等于文件总字节数减去8。这是一个DWORD，我这个文件对应是311614+8 = 311622。

然后是“57 41 56 45 66 6D 74 20”，也是Ascii字符“WAVEfmt”，这部分是固定格式。

然后是PCMWAVEFORMAT部分，可以对照一下上面的struct定义，首先就是一个WAVEFORMAT的struct。

随后是“10 00 00 00”，这是一个DWORD，对应数字16，这个对应定义中的Sizeof (PCMWAVEFORMAT)，后面我们可以看到这个段内容正好是16个字节。

随后的字节是“01 00”，这是一个WORD，对应定义为编码格式“WAVE_FORMAT_PCM”，我们一般用的是这个。

随后的是“02 00”，这是一个WORD，对应数字1，表示声道数为1，2表示双声道，这是个双声道Wav。

随后的是“44 AC 00 00”，这是一个DWORD，对应数字44100，代表的是采样率。

随后的是“22 56 00 00”，这是一个DWORD，对应数字22050，代表的是采样频率22050。

然后是“04 00”，这是一个WORD，对应数字是2，表示块对齐的内容，每个采样对应的字节数。

然后是“10 00”，这是一个WORD，对应WAVE文件的采样大小，数值为16，采样位数16Bits。

然后是一串“64 61 74 61”，这个是Ascii字符“data”，标示头结束，开始数据区域。

而 后是数据区的开头，有一个DWORD，我这里的字符是“C0 3C 00 00”，对应的十进制数为15552，看一下前面正好可以看到，文件大小是15596，其中到“data”标志出现为止的头是40个字节，再减去这个标志 的4个字节正好是15552，再往后面就是真正的Wave文件的数据体了，头文件的解析就到这里。

所占字节数		具体内容	
ID	4 Bytes	'fact'	
Size	4 Bytes	数值为4	
data	4 Bytes		

图4 Fact Chunk

Fact Chunk是可选字段，一般当wav文件由某些软件转化而成，则包含该Chunk。

结构定义如下：

```
struct FACT_BLOCK
{
    char  szFactID[4]; // 'f','a','c','t'
    DWORD dwFactSize;
};
```

Data Chunk

图5 Data Chunk

Data Chunk是真正保存wav数据的地方，以'data'作为该Chunk的标示。然后是数据的大小。紧接着就是wav数据。根据Format Chunk中的声道数以及采样bit数，wav数据的bit位置可以分成以下几种形式：

单声道	取样1	取样2	取样3	取样4
8bit量化	声道0	声道0	声道0	声道0
双声道	取样1	取样2		
8bit量化	声道0(左)	声道1(右)	声道0(左)	声道1(右)
单声道	取样1	取样2		
16bit量化	声道0 (低位字节)	声道0 (高位字节)	声道0 (低位字节)	声道0 (高位字节)
双声道	取样1			
16bit量化	声道0(左) (低位字节)	声道0(左) (高位字节)	声道1(右) (低位字节)	声道1(右) (高位字节)

图6 wav数据bit位置安排方式

Data Chunk头结构定义如下：

```
struct DATA_BLOCK
{
    char  szDataID[4]; // 'd','a','t','a'
    DWORD dwDataSize;
};
```

三、小结

因此，根据上述结构定义以及格式介绍，很容易编写相应的wav格式解析代码。这里具体的代码就不给出了。

阅读 (846) | 评论 (4) | 转发 (2) |

上一篇: [alsa-lib 交叉编译以及声卡驱动测试](#)
下一篇: [用户模式I2C驱动](#)

0

相关热门文章

大数据的冲击

你的2014愿望是什么?

九问“小牛犊”一启动资产翻倍...

国际投行集体“砸”2014年国际...

电子控制汽油喷射系统主要元件...

shell中字符串操作

shell中的特殊字符

linux守护进程的几个关键地方...

stagefright与opencore对比

嵌入式Linux之我行——u-boot-...

kvm上的两台windows虚拟机不能...

关于ip_route_output_key和ip...

test a tests

linux inode中磁盘地址列表有...

LNMP 老是会出现502?

给主人留下些什么吧!~~



ilfmonday 2013-11-16 22:33:02

enzo26: [emacs](#) 做的

了解。。。

[回复](#) | [举报](#)



ilfmonday 2013-11-16 22:32:59

enzo26: [emacs](#) 做的

了解。。。

[回复](#) | [举报](#)



enzo26 2013-11-06 12:20:25

ilfmonday: 麻烦问博主, 这个ascii表格是用什么工具做的呢? vimwiki?

[emacs](#) 做的

[回复](#) | [举报](#)



ilfmonday 2013-11-05 15:37:38

麻烦问博主, 这个ascii表格是用什么工具做的呢? vimwiki?

[回复](#) | [举报](#)

评论热议

登录后评论。

[登录](#) [注册](#)

[关于我们](#) | [关于IT168](#) | [联系方式](#) | [广告合作](#) | [法律声明](#) | [免费注册](#)

Copyright 2001-2010 ChinaUnix.net All Rights Reserved 北京皓辰网域网络信息技术有限公司. 版权所有

感谢所有关心和支持过ChinaUnix的朋友们

京ICP证041476号 京ICP证060528号