

CSDN

首页 博客 学院 下载 GitChat TinyMind 论坛 问答 商城 ...

搜博主文章

写博客

发Ch...

雷霄骅 (leixiaohua1020)的专栏

一个广院工... 音视频技术笔记

目录

收藏

27

HELLO WORLD! KEEP CODING!

RSS订阅

最简单的基于FFMPEG+SDL的音频播放器 ver2 （采用SDL2.0）

2014年07月14日 14:26:55

评论

阅读量: 27969

- =====
- 最简单的FFmpeg的音频播放器系列文章列表:
- [《最简单的基于FFMPEG+SDL的音频播放器》](#)
- [《最简单的基于FFMPEG+SDL的音频播放器 ver2 （采用SDL2.0）》](#)
- [《最简单的基于FFMPEG+SDL的音频播放器：拆分-解码器和播放器》](#)
- =====

简介

之前做过一个简单的音频播放器：《最简单的基于FFMPEG+SDL的音频播放器》，采用的是SDL1.2。前两天刚把原先做的《最简单的基于FFMPEG+SDL的视频播放器》更新采用了SDL2.0，于是顺手也把音频播放器更新成为SDL2.0。

需要注意的是，与播放视频有很大的不同，SDL2.0播放音频的函数相对于SDL1.2来说变化很小。基本上保持了不变。

除了使用SDL2.0之外，修改了如下地方：

- *重建了工程，删掉了不必要的代码，把代码修改得更规范更易懂。
- *可以通过宏控制是否使用SDL，以及是否输出PCM。
- *支持MP3,AAC等多种格式

源代码

```
[cpp]
1.  /**
2.   * 最简单的基于FFmpeg的音频播放器 2
3.   * Simplest FFmpeg Audio Player 2
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   * leixiaohua1020@126.com
7.   * 中国传媒大学/数字电视技术
8.   * Communication University of China / Digital TV Technology
9.   * http://blog.csdn.net/leixiaohua1020
10.  *
11.  * 本程序实现了音频的解码和播放。
12.  * 是最简单的FFmpeg音频解码方面的教程。
13.  * 通过学习本例子可以了解FFmpeg的解码流程。
14.  *
15.  * 该版本使用SDL 2.0替换了第一个版本中的SDL 1.0。
16.  * 注意：SDL 2.0中音频解码的API并无变化。唯一变化的地方在于
17.  * 其回调函数的中的Audio Buffer并没有完全初始化，需要手动初始化。
18.  * 本例子中即SDL_memset(stream, 0, len);
19.  *
20.  * This software decode and play audio streams.
21.  * Suitable for beginner of FFmpeg.
22.  *
23.  * This version use SDL 2.0 instead of SDL 1.2 in version 1
24.  * Note:The good news for audio is that, with one exception,
25.  * it's entirely backwards compatible with 1.2.
26.  * That one really important exception: The audio callback
27.  * does NOT start with a fully initialized buffer anymore.
28.  * You must fully write to the buffer in all cases. In this
```

```

29.  * example it is SDL_memset(stream, 0, len);
30.  *
31.  * ``ersion 2.0
32.
33.  27  lude <stdio.h>
34.  #include <stdlib.h>
35.  lude <string.h>
36.
37. 目录 ine __STDC_CONSTANT_MACROS
38.
39.  ef _WIN32
40.  ndows
41.  rn "c"
42.
43. 评论 lude "libavcodec/avcodec.h"
44.  #include "libavformat/avformat.h"
45.  lude "libswresample/swresample.h"
46.  lude "SDL2/SDL.h"
47. 微信
48.  e
49.  nux...
50. 微博 ef __cplusplus
51.  rn "c"
52.
53.  QQ if
54.  #include <libavcodec/avcodec.h>
55.  #include <libavformat/avformat.h>
56.  #include <libswresample/swresample.h>
57.  #include <SDL2/SDL.h>
58.  #ifdef __cplusplus
59.  };
60.  #endif
61.  #endif
62.
63.  #define MAX_AUDIO_FRAME_SIZE 192000 // 1 second of 48khz 32bit audio
64.
65.  //Output PCM
66.  #define OUTPUT_PCM 1
67.  //Use SDL
68.  #define USE_SDL 1
69.
70.  //Buffer:
71.  //|-----|-----|
72.  //chunk-----pos---len-----|
73.  static Uint8 *audio_chunk;
74.  static Uint32 audio_len;
75.  static Uint8 *audio_pos;
76.
77.  /* The audio function callback takes the following parameters:
78.  * stream: A pointer to the audio buffer to be filled
79.  * len: The length (in bytes) of the audio buffer
80.  */
81.  void fill_audio(void *udata, Uint8 *stream, int len){
82.  //SDL 2.0
83.  SDL_memset(stream, 0, len);
84.  if(audio_len==0)
85.  return;
86.
87.  len=(len>audio_len?audio_len:len); /* Mix as much data as possible */
88.
89.  SDL_MixAudio(stream, audio_pos, len, SDL_MIX_MAXVOLUME);
90.  audio_pos += len;
91.  audio_len -= len;
92.  }
93.  //-----
94.
95.
96.  int main(int argc, char* argv[])
97.  {
98.  AVFormatContext *pFormatCtx;
99.  int i, audioStream;
100.  AVCodecContext *pCodecCtx;
101.  AVCodec *pCodec;
102.  AVPacket *packet;
103.  uint8_t *out_buffer;
104.  AVFrame *pFrame;
105.  SDL_AudioSpec wanted_spec;
106.  int ret;
107.  uint32_t len = 0;
108.  int got_picture;
109.  int index = 0;
110.  int64_t in_channel_layout;
111.  struct SwrContext *au_convert_ctx;
112.
113.  FILE *pFile=NULL;
114.  char url[]="xiaoqingge.mp3";
115.

```

```

116.     av_register_all();
117.     avformat_network_init();
118.     pFormatCtx = avformat_alloc_context();
119.     //Open
120. 27 if(avformat_open_input(&pFormatCtx,url,NULL,NULL)!=0){
121.     printf("Couldn't open input stream.\n");
122.     return -1;
123. }
124. // Retrieve stream information
125. if(avformat_find_stream_info(pFormatCtx,NULL)<0){
126.     printf("Couldn't find stream information.\n");
127.     return -1;
128. }
129. // Dump valid information onto standard error
130. av_dump_format(pFormatCtx, 0, url, false);
131.
132. // Find the first audio stream
133. audioStream=-1;
134. for(i=0; i < pFormatCtx->nb_streams; i++)
135.     if(pFormatCtx->streams[i]->codec->codec_type==AVMEDIA_TYPE_AUDIO){
136.         audioStream=i;
137.         break;
138.     }
139.
140. QQ if(audioStream==-1){
141.     printf("Didn't find a audio stream.\n");
142.     return -1;
143. }
144.
145. // Get a pointer to the codec context for the audio stream
146. pCodecCtx=pFormatCtx->streams[audioStream]->codec;
147.
148. // Find the decoder for the audio stream
149. pCodec=avcodec_find_decoder(pCodecCtx->codec_id);
150. if(pCodec==NULL){
151.     printf("Codec not found.\n");
152.     return -1;
153. }
154.
155. // Open codec
156. if(avcodec_open2(pCodecCtx, pCodec,NULL)<0){
157.     printf("Could not open codec.\n");
158.     return -1;
159. }
160.
161.
162. #if OUTPUT_PCM
163.     pFile=fopen("output.pcm", "wb");
164. #endif
165.
166.     packet=(AVPacket *)av_malloc(sizeof(AVPacket));
167.     av_init_packet(packet);
168.
169.     //Out Audio Param
170.     uint64_t out_channel_layout=AV_CH_LAYOUT_STEREO;
171.     //nb_samples: AAC-1024 MP3-1152
172.     int out_nb_samples=pCodecCtx->frame_size;
173.     AVSampleFormat out_sample_fmt=AV_SAMPLE_FMT_S16;
174.     int out_sample_rate=44100;
175.     int out_channels=av_get_channel_layout_nb_channels(out_channel_layout);
176.     //Out Buffer Size
177.     int out_buffer_size=av_samples_get_buffer_size(NULL,out_channels ,out_nb_samples,out_sample_fmt, 1);
178.
179.     out_buffer=(uint8_t *)av_malloc(MAX_AUDIO_FRAME_SIZE*2);
180.     pFrame=av_frame_alloc();
181.     //SDL-----
182.     #if USE_SDL
183.         //Init
184.         if(SDL_Init(SDL_INIT_VIDEO | SDL_INIT_AUDIO | SDL_INIT_TIMER)) {
185.             printf("Could not initialize SDL - %s\n", SDL_GetError());
186.             return -1;
187.         }
188.         //SDL_AudioSpec
189.         wanted_spec.freq = out_sample_rate;
190.         wanted_spec.format = AUDIO_S16SYS;
191.         wanted_spec.channels = out_channels;
192.         wanted_spec.silence = 0;
193.         wanted_spec.samples = out_nb_samples;
194.         wanted_spec.callback = fill_audio;
195.         wanted_spec.userdata = pCodecCtx;
196.
197.         if (SDL_OpenAudio(&wanted_spec, NULL)<0){
198.             printf("can't open audio.\n");
199.             return -1;
200.         }
201.     #endif
202.

```

```

203. //FIX:Some Codec's Context Information is missing
204. in_channel_layout=av_get_default_channel_layout(pCodecCtx->channels);
205. //Swr
206.
207. 27 au_convert_ctx = swr_alloc();
208. au_convert_ctx=swr_alloc_set_opts(au_convert_ctx,out_channel_layout, out_sample_fmt, out_sample_rate,
209. in_channel_layout,pCodecCtx->sample_fmt , pCodecCtx->sample_rate,0, NULL);
210. swr_init(au_convert_ctx);
211.
212. //Play
213. SDL_PauseAudio(0);
214.
215. while(av_read_frame(pFormatCtx, packet)>=0){
216.     if(packet->stream_index==audioStream){
217.         ret = avcodec_decode_audio4( pCodecCtx, pFrame,&got_picture, packet);
218.         if ( ret < 0 ) {
219.             printf("Error in decoding audio frame.\n");
220.             return -1;
221.         }
222.         if ( got_picture > 0 ){
223.             swr_convert(au_convert_ctx,&out_buffer, MAX_AUDIO_FRAME_SIZE,(const uint8_t **)pFrame->data , pFrame->nb_samp
224.             ;
225.             printf("index:%5d\t pts:%lld\t packet size:%d\n",index,packet->pts,packet->size);
226.             if
227.
228.
229. #if OUTPUT_PCM
230.             //Write PCM
231.             fwrite(out_buffer, 1, out_buffer_size, pFile);
232. #endif
233.             index++;
234.         }
235.
236. #if USE_SDL
237.         while(audio_len>0)//Wait until finish
238.             SDL_Delay(1);
239.
240.         //Set audio buffer (PCM data)
241.         audio_chunk = (Uint8 *) out_buffer;
242.         //Audio buffer length
243.         audio_len =out_buffer_size;
244.         audio_pos = audio_chunk;
245.
246. #endif
247.     }
248.     av_free_packet(packet);
249. }
250.
251. swr_free(&au_convert_ctx);
252.
253. #if USE_SDL
254.     SDL_CloseAudio();//Close SDL
255.     SDL_Quit();
256. #endif
257.
258. #if OUTPUT_PCM
259.     fclose(pFile);
260. #endif
261.     av_free(out_buffer);
262.     avcodec_close(pCodecCtx);
263.     avformat_close_input(&pFormatCtx);
264.
265.     return 0;
266. }

```

下载

Simplest FFmpeg audio player 2

SourceForge: <https://sourceforge.net/projects/simplestffmpegaudioplayer/>

Github: https://github.com/leixiaohua1020/simplest_ffmpeg_audio_player

开源中国: http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_audio_player

修正版 下载地址：<http://download.csdn.net/detail/leixiaohua1020/7853285>

*注：²⁷ 中又修正了以下问题：

1. PCMA 内fwrite()的size有错误
2. PCMA 内fclose()外面添加了宏定义
3. 部分器（例如WMA）的AVCodecContext中的channel_layout没有进行初始化。会导致SwrContext初始化失败。改为通过channels（一定化）计算channel_layout而不是直接取channel_layout的值。

更新-2.2 (2015.2.13)=====

这次考虑跨平台的要求，调整了源代码。经过这次调整之后，源代码可以在以下平台编译通过：
VC++：打开sln文件即可编译，无需配置。
GCC：打开compile_cl.bat即可命令行下使用cl.exe进行编译，注意可能需要按照VC的安装路径调整脚本里面的参数。编译命令如QQ

```
[plain]
1.  ::VS2010 Environment
2.  call "D:\Program Files\Microsoft Visual Studio 10.0\VC\vcvarsall.bat"
3.  ::include
4.  @set INCLUDE=include;%INCLUDE%
5.  ::lib
6.  @set LIB=lib;%LIB%
7.  ::compile and link
8.  cl simplest_ffmpeg_audio_player.cpp /MD /link SDL.lib SDLmain.lib avcodec.lib ^
9.  avformat.lib avutil.lib avdevice.lib avfilter.lib postproc.lib swresample.lib swscale.lib ^
10. /SUBSYSTEM:WINDOWS /OPT:NOREF
```

MinGW：MinGW命令行下运行compile_mingw.sh即可使用MinGW的g++进行编译。编译命令如下。

```
[plain]
1.  g++ simplest_ffmpeg_audio_player.cpp -g -o simplest_ffmpeg_audio_player.exe \
2.  -I /usr/local/include -L /usr/local/lib \
3.  -lmingw32 -lSDL2main -lSDL2 -lavformat -lavcodec -lavutil -lswresample
```

GCC：Linux或者MacOS命令行下运行compile_gcc.sh即可使用GCC进行编译。编译命令如下。

```
[plain]
1.  gcc simplest_ffmpeg_audio_player.cpp -g -o simplest_ffmpeg_audio_player.out -I /usr/local/include -L /usr/local/lib \
2.  -lSDL2main -lSDL2 -lavformat -lavcodec -lavutil -lswresample
```

PS：相关的编译命令已经保存到了工程文件夹中

CSDN下载地址：<http://download.csdn.net/detail/leixiaohua1020/8444761>

SourceForge、Github等上面已经更新。

更新-2.2 (2015.7.17)=====

增加了下面工程：

- simplest_ffmpeg_audio_decoder：音频解码器。使用了libavcodec和libavformat。
- simplest_audio_play_sdl2：使用SDL2播放PCM采样数据的例子。

CSDN下载地址：<http://download.csdn.net/detail/leixiaohua1020/8924329>

SourceForge、Github等上面已经更新。

所属专栏：[FFmpeg](#)

27

目录

收藏

评论

微信



想对作者说点什么？

我来说一句



M

2016-08-27 00:29:48 #20楼

雷 微博



pl QQ 2016-04-07 19:17:34 #19楼

[查看回复\(3\)](#)

老师，为什么程序运行到av_samples_get_buffer_size总是返回负值，我是直接将您的代码给复制 过来的，ffmpeg现在的版本是3.0.1，SDL2.0

```
[cpp]
01. //Out Audio Param
02. uint64_t out_channel_layout=AV_CH_LAYOUT_STEREO;
03. //nb_samples: AAC-1024 MP3-1152
04. int out_nb_samples=pCodecCtx->frame_size;
05. AVSampleFormat out_sample_fmt=AV_SAMPLE_FMT_S16;
06. int out_sample_rate=44100;
07. int out_channels=av_get_channel_layout_nb_channels(out_channel_layout);
08. //Out Buffer Size
09. int out_buffer_size=av_samples_get_buffer_size(NULL,out_channels ,out_nb_samples,out_sample_fmt, 1);
10. if(out_buffer_size < 0)
11. {
12.     printf("Could not calculate required memory.\n");
13.     return -1;
14. }
```



TaiJi1985

2016-03-12 21:06:10 #18楼

[查看回复\(2\)](#)

大神你好。我根据你的例子，做了一些修改。这里贴不开就发到我自己的博客上了。http://blog.csdn.net/taiji1985/article/details/50867480 1 将 视频例子和音频例子进行了合并 2 使用循环缓冲区替换了原本的缓冲机制（从网上找的缓冲区代码，抱歉忘记从哪里拷贝的了，不然一定会注明的） 3 尝试播放了rtmp 4 修正了声音有断续杂音的情况。关键使用swr_convert转换声音后，声音数据的大小是swr_convert返回值*声道数*每帧数据字节数



fxStilling

2016-02-17 14:18:00 #17楼

[查看回复\(3\)](#)

[查看 41 条热评](#)

SDL编程三（音频）

SDL播放声音的基本流程如下：（1）创建一个回调函数用于混合音频数据，并放入音频流；（2）设置音频参数，调用SDL_OpenAudio，打开音频设备；（3）调用SDL_Paus...



zqf_office

2014-06-13 11:36:18 阅读数：2816

最简单的视音频播放示例9：SDL2播放PCM

本文记录SDL播放音频的技术。在这里使用的版本是SDL2。实际上SDL本身并不提供视音频播放的功能，它只是封装了视音频播放的底层API。在Windows平台下，SDL封装了Direct3D这类的API...



leixiaohua1020

2014-10-31 00:23:11 阅读数：23913

[5] ffmpeg+SDL2实现的音频播放器V2.0(无杂音) - CSDN博客

日期:2016.10.4 作者:isshe github:github.com/isshe 邮箱:i.sshe@outlook.com 平台:ubuntu16.04 64bit1. 前言目前为止,学习了并记录了ffmpeg+SDL2显示视频... 2018-5-23

最简单的基于FFMPEG+SDL的视频播放器 ver2 (采用SDL2.0) - CSDN博客

之前做过一个FFMPEG+SDL的简单播放器:《100行代码实现最简单的基于FFMPEG+SDL的视频播放器》。最近有不少人反映SDL已经升级到2.0版本了,甚至官网的Wiki上都有...

2018-6-7

耳聩，鸟、耳朵嗡嗡响，试试这个小妙招，一招解决!

君旺生

目录



SDL：多窗口播放音频打开方式

在实现播放控件中要求可以多个窗口播放，注意这里的需求不是分屏。官方例子参考：SDL_OpenAudioDeviceSDL_AudioSpec want, have; SDL_AudioDeviceID

评论

2018-03-19 15:10:16 阅读数：110

ffmpeg+SDL2实现的视频播放器「视音频同步」 - CSDN博客

最简单的FFMPEG+SDL的视频播放器 ver2 (采用SDL2.0) 之前做过一个FFMPEG+SDL的简单播放器：《100行代码实现最简单的基于FFMPEG+SDL的视频播放器》近有不...

2018-3

QQ

ffmpeg+SDL2实现的视频播放器

ffmpeg+SDL2实现的视频播放器 2016-10-09 上传大小:5.22MB ffmpeg 博文:http://blog.csdn.net/i_scream_/article/details/52760033 中的代码。实现了音视频的...

2018-5-8

SDL 与 FFMPEG 音乐播放器开发（2）——混播多个音频

第一篇总体提了一下SDL，完全没有提到FFMPEG。我的思路是，在说解码之前，你起码要知道怎么使用解码后的文件。相信大家如果看了网上的一些教程，应该已经能够播放出PCM文件。今天我来谈谈如何播放多个...

u013080313

2015-12-23 15:16:25 阅读数：1183

ffmpeg+SDL能同时播放多路视频么？

转自： <http://www.chinavideo.org/archiver/?tid-17184.html> ffmpeg+SDL能同时播放多路视频么？ 如题，有能实现的大侠么？ ...

xiaoxianerqq

2015-11-10 17:12:16 阅读数：1481

ffmpeg+SDL2实现的视频播放器(windows版)

被举报人: i_scream_ 举报的资源分: 5 *类型: *详细原因: 取消提交 ffmpeg+SDL2实现的视频播放器(windows版) 5积分 立即下载 ...

2018-5-9

[8] ffmpeg + SDL2 实现的视频播放器「快进、快退」

举报的资源分: 5 *类型: *详细原因: 取消提交 [8] ffmpeg + SDL2 实现的视频播放器「快进、快退」 5积分 立即下载 ...

2018-5-7

SDL 混音同时播放多个媒体文件

转自http://blog.csdn.net/u013080313/article/details/50375098 首先，跟学习任何东西一样，官方的资料永远是最应该看的。传送门：sdl官方...

tropicofcancer9

2017-01-03 18:45:24 阅读数：1421

老中医说：男人多吃它，性生活时间延长5倍

国瑞兴邦·顶新



FFMPEG+SDL2 实现播放器功能 - CSDN博客

FFMPEG+SDL2 实现播放器功能2018年02月21日 16:58:32 阅读数:79 一、界面 1 打开的pushbutton 2 停止/开始的pushbutton 3...

2018-5-21

[8] ffmpeg + SDL2 实现的视频播放器「快进、快退」 - CSDN博客

前言:实现暂停、播放的时候就想,快进快退要如何实现呢?没想到ffmpeg提供了这个av_seek_frame()这么方便的函数。相关知识: ffmpeg的一些define:AV_TIME_BASE : ...

2018-5-22

最简单. 基于FFMPEG+SDL的视频播放器 ver2 （采用SDL2.0）

之前做过一个FFMPEG+SDL的简单播放器：《100行代码实现最简单的基于FFMPEG+SDL的视频播放器》。最近有不少人反映SDL已经升级到2.0版本了，甚至官网上也只有SDL2.0的文...

leixiaohua1020 2014-08-31 01:20:14 阅读数：78780

FFmpeg+SDL2.0+SDL2.0 视频播放Demo

2016年 52.62MB 下载



基于FFmpeg+SDL2 实现简单rtsp播放器 - CSDN博客

基于ffmpeg+SDL2 实现简单rtsp播放器2014年07月09日 15:58:30 阅读数:3395 参考资料: 编译参考: http://...

2018-6 微博

undefined

[6] ffmpeg+SDL2 实现的视频播放器「视音频同步」

日期：2016.10.8 作者：isshe github: github.com/isshe 邮箱：i.sshe@outlook.com 平台：ubuntu16.04 64bit前言 这个...

i_scream_ 2016-10-09 00:40:27 阅读数：4352

FFmpeg+SDL2音频播放器

基于雷神最简单的音频播放器修改。/** * 最简单的基于FFmpeg的音频播放器 2 * Simplest FFmpeg Audio Player 2 * 雷霄骅 Lei Xiaohua * le...

bixinwei22 2018-01-11 16:53:22 阅读数：192

[5] ffmpeg+SDL2实现的音频播放器V2.0（无杂音）

日期：2016.10.4 作者：isshe github: github.com/isshe 邮箱：i.sshe@outlook.com 平台：ubuntu16.04 64bit1. 前...

i_scream_ 2016-10-05 01:31:03 阅读数：3291

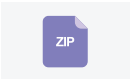
最简单的基于FFMPEG+SDL的音频播放器

FFMPEG工程浩大，可以参考的书籍又不是很多，因此很多刚学习FFMPEG的人常常感觉到无从下手。在此我把自己做项目过程中实现的一个非常简单的音频播放器（大约100行代码）源代码传上来，以作备忘，同...

leixiaohua1020 2013-08-29 17:11:09 阅读数：63608

ffmpeg+SDL2实现的音频播放器V2.0（无杂音）

2016年10月05日 11.44MB 下载



外语好课！特价1元你想学的都在这！抢课>

1元学外语！点击抢课>7天说流利小语种！



FFmpeg+SDL2.0 音频解码播放（部分格式杂音处理）

还是参照雷大神的例子改写得，但是改写过程中碰到几个问题，纠结的比较久的问题是解码MP4，avi文件，播放出来的音频不仅很多杂音而且断断续续。因为是刚接触这块，没什么音频方面得知识，出了问题也找...

u012853439 2016-11-18 16:41:02 阅读数：1246


ffmpeg+sdl教程----编写一个简单的播放器3(为视频加入音频)

上个教程实现了视频的简单播放，但那是个哑巴电影，完全没有声音，下面的程序会为视频加入相应的音频：// ffmpegExe.cpp: 主项目文件。#include "stdafx.h" #...

mu399 2010-08-16 10:10:00 阅读数：17496

ffmpeg+sdl教程----编写一个简单的播放器5(同步视频到音频)

个人认为，这部分教程的新增代码量虽然不是最多的，难度却是最大的，重复看了多次才明白，因为有两个问题的困扰，搞得还不清楚： 1.音频和视频既然都有各自的时间戳，各自按各自的时间戳来播放不就行了，...

 lihuibo128

2014-08-27 15:31:10

阅读数：1145

FFMPEG+SDL2.0 RTMP视频流播放器

由于工作需要要对RTMP视频直播流进行接收和播放。研究一段时间，终于有所成效。下面对在研究过程进行一些总结，方便其他人员少走一些弯路。 一、简介 视频流播放开始音频和视频数据，服务器在推送这些原始数...

 qq_3383399

2018-03-23 15:51:09

阅读数：38

AAC流如何解码播放

AAC原始数据流，让ffmpeg软解或系统硬解，最最简单的处理可在前面加上ADTS头，那么这个ADTS头如何添加呢？其实就是七个字节，下面分享一个我使用的代码样例，评论回复数据都是以一个个sample为单元...

 acn123

2015-01-21 18:02:09

阅读数：1482

正版AVI文件下载

免费mp4格式下载

百度广告
QQ



FFMpeg+SDL2.0 播放本地AVI文件练手示例程序

FFMpeg+SDL2播放本地AVI视频文件

 humadivinity

2017-12-13 16:20:19

阅读数：331

[2] SDL的基础知识以及利用SDL播放视频

SDL 的基础知识以及利用SDL播放视频此博文相关知识点从雷神的博客以及视频学习，截图也是用了他的课件, 雷神博客地址：http://blog.csdn.net/leixiaohua1020/日期...

 i_scream_

2016-10-01 01:07:54

阅读数：2904

个人资料



雷霄骅

关注

原创	粉丝	喜欢	评论
373	1万+	453	7315

等级： 博客 9

访问： 1039万+

积分： 6万+

排名： 50

勋章：



道路检测



关于我

姓名：雷霄骅

网名：leixiaohua1020

本科：中国传媒大学-广播电视工程

硕士：中国传媒大学-数字电视技术

博士：

- 27
- 目录
- 收藏
- 评论
- 微信
- 微博
- QQ

中国传媒大学-数字视频技术
Email:
leixiaohua1020@126.com
QQ:
494085803

[注1：QQ消息较多，难以一一回复，见谅]
[注2：CSDN私信功能使用很少，有问题可以直接在博客评论处留言]

奖项：
2014年度 - CSDN博客之星
2015年度 - 微软MVP
2015年度 - CSDN博客之星
简介：
主要从事与广播电视有关的视音频技术的研究。包括视音频质量评价，视音频编解码，流媒体，媒资检索等。

最新文章

- [投稿] Speex回声消除原理深度解析
- [投稿]房间声学原理与Schroeder混响算法实现
- [投稿]一个频域语音降噪算法实现及改进方法
- 最简单的基于FFmpeg的AVfilter的例子-纯净版
- 视音频数据处理入门：UDP-RTP协议解析

博主专栏

-  视频质量评价
阅读量：433208 41 篇
-  开源多媒体项目源代码分析
阅读量：1156747 91 篇
-  FFmpeg
阅读量：4841230 135 篇

归档

2016年4月	1篇
2016年3月	2篇
2016年2月	1篇
2016年1月	7篇
2015年12月	1篇

展开

最新评论

- 视频编码器评测系统：VideoCo...
tqs_1220：天妒英才
- FFmpeg获取DirectSho...
a137748099：[reply]liangqingzhi[reply] 大佬，有的电脑使用regsvr32注册不成...
- [总结]FFMPEG视音频编解码零...
qq_17276615：每次视频编解码都会搜到你的文章，感谢您给我们这些菜鸟一些指引。谢谢！
- [投稿] Speex回声消除原理深...
jwq1220：厉害！

- 27
- 目录
- 收藏
- 评论
- 微信
- 微博
- QQ

最简单的基于librtmp的示例： ...
qq_32245927: [reply]zzqgtt[/reply] 你好，你的Q
Q多少