

本文汇总一下自己视音频编解码学习方面的开源项目。这些开源项目大体上可以分成专业领域程序，FFmpeg示例程序，FFmpeg移植程序，多媒体项目示例程序，视音频编解码原理学习工程几个类别。这些类别的详细含义如下所列。

- 专业领域程序：一些有专业性质的程序。
- FFmpeg示例程序：FFmpeg的类库的各种示例程序。
- FFmpeg移植程序：FFmpeg项目中移植到Windows的VC平台下的程序。
- 多媒体项目示例程序：除了FFmpeg之外其他多媒体项目的示例程序。
- 视音频编解码学习工程：用于学习视音频编解码原理的示例程序。

下面按照类别分别记录每个项目。每个程序的介绍分成“简介”，“文章”，“项目主页”三个部分：

- “简介”使用几句简短的话介绍开源项目。
- “文章”则是我自己写的有关该项目的文章。
- “项目主页”则是该项目的主页地址。

此前一直以SourceForge作为各个项目的主页地址。后来受到不少人的建议，将代码在Github上也保存了一份；国内的开源中国的代码托管做的也不错，因此也在开源中国上保存了一份代码。此外，在SourceForge上做了一个简单的Wiki，在Github上做了一个简单的Blog，方便对这两个网站上的项目进行查找和访问。上述三个网站上的项目列表可以通过下列地址访问。

- 我的SourceForge主页：<https://sourceforge.net/u/leixiaohua1020/wiki/>
- 我的Github主页：<http://leixiaohua1020.github.com/>
- 我的开源中国主页：<http://git.oschina.net/leixiaohua1020>

## 1. 专业领域程序

专业领域程序指的是主要应用于科学实验方面的程序。这类程序并不是视音频技术开发必须的，主要用于研究工作。

### 1.1 VideoEye

#### 简介

VideoEye是一个开源的实时视频分析的软件。本软件可以播放和分析视频数据。它支持多种视频流输入方式：HTTP，RTMP，RTSP以及文件等等。该软件可以实时分析视频流并能以图形化的方式呈现其分析结果。



#### 文章

《开源实时视频码流分析软件：VideoEye》

#### 项目主页

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

Github：<https://github.com/leixiaohua1020/VideoEye>

开源中国：<http://git.oschina.net/leixiaohua1020/VideoEye>

### 1.2 TlandSI

#### 简介

本工具用于计算视频的时间信息(TI,也称为时间复杂度),以及空间信息(SI,也称为空间复杂度),以上两个指标取自于ITU-R BT.1788标准。



## 文章

《视频特性TI（时间信息）和SI（空间信息）的计算工具：TlandSI》

《视频特性TI（时间信息）和SI（空间信息）的计算工具：TlandSI-压缩码流版》

## 项目主页

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

Github：<https://github.com/leixiaohua1020/TlandSI>

开源中国：<http://git.oschina.net/leixiaohua1020/TlandSI>

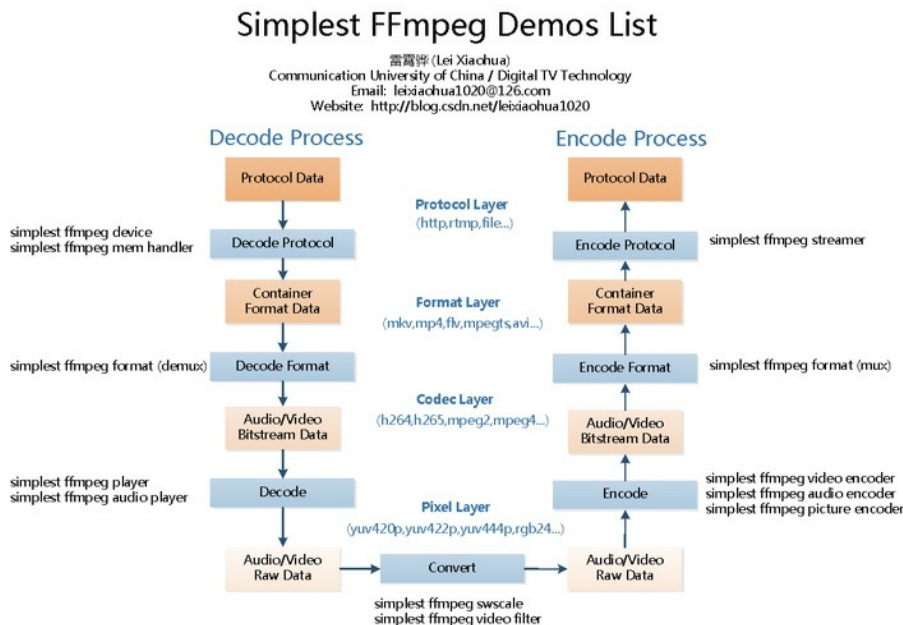
## 2.

### FFmpeg示例程序

FFmpeg示例程序是我做的最多的一类程序。这些示例程序展示了利用FFmpeg的类库进行视音频程序开发的方法，比较适合新手学习FFmpeg。目前这些示例程序已经覆盖了从流媒体接收，解封装，解码，像素转换，编码，封装，流媒体发送所有这些流程。



FFmpeg示例程序在视频处理的流程中的位置如下图所示。



[单击查看更清晰的图片](#)

图中显示了一个完整的视频处理流程。按照蓝色箭头指向的顺序，从外部协议接收下来的数据依次经过解协议，解封装，解码，像素转换，编码，封装，发送这几个步骤。这些步骤按照处理数据类型不同，可以分成几个层次：

- 协议层 (Protocol Layer)：该层处理的数据为符合特定流媒体协议规范的数据，例如http, rtmp, file等。
- 封装层 (Format Layer)：该层处理的数据为符合特定封装格式规范的数据，例如mkv, mp4, flv, mpegts, avi等。
- 编码层 (Codec Layer)：该层处理的数据为符合特定编码标准规范的数据，例如h264, h265, mpeg2, mpeg4等。
- 像素层 (Pixel Layer)：该层处理的数据为符合特定像素格式规范的数据，例如yuv420p, yuv422p, yuv444p, rgb24等。

下面我们上述几个层次对FFmpeg示例程序进行归类。

协议层处理包含以下项目：

- simplest ffmpeg device：使用FFmpeg读取设备中的媒体数据。
- simplest ffmpeg mem handler：使用FFmpeg读取内存中的数据。
- simplest ffmpeg streamer：使用FFmpeg发送流媒体数据。

封装层处理包含以下项目：

- simplest ffmpeg format：使用FFmpeg分离音视频（demux），封装音视频（mux），转换封装格式（remux）。

编码层处理包含以下项目：

- simplest ffmpeg player：使用FFmpeg解码播放视频数据。
- simplest ffmpeg audio player：使用FFmpeg解码播放音频数据。
- simplest ffmpeg video encoder：使用FFmpeg编码视频数据。
- simplest ffmpeg audio encoder：使用FFmpeg编码音频数据。
- simplest ffmpeg picture encoder：使用FFmpeg编码图片。

像素层处理包含以下项目：

- simplest ffmpeg swscale：使用FFmpeg转换像素格式。
- simplest ffmpeg video filter：使用FFmpeg给视频添加滤镜效果。

《FFmpeg示例程序合集》可以通过Git批量获取：

[FFmpeg示例程序合集-Git批量获取脚本](#)

《FFmpeg示例程序合集》也可以在VC++/CL.exe/MinGW/Linux GCC/Mac OS等多个平台上编译通过：

[FFmpeg示例程序合集-批量编译脚本](#)

下面逐一列出这些项目的信息。

## 2.1

### Simplest ffmpeg player

*最简单的基于FFMPEG的视频播放器*

#### 简介

本程序实现了视频文件的解码和显示。是最简单的FFmpeg视频解码方面的教程。通过学习本例子可以了解FFmpeg的解码流程。

项目包含6个工程：

- simplest\_ffmpeg\_player：标准版，FFmpeg学习的开始。
- simplest\_ffmpeg\_player\_su：SU（SDL Update）版，加入了简单的SDL的Event。
- simplest\_ffmpeg\_decoder：一个包含了封装格式处理功能的解码器。使用了libavcodec和libavformat。
- simplest\_ffmpeg\_decoder\_pure：一个纯净的解码器。只使用libavcodec（没有使用libavformat）。
- simplest\_video\_play\_sdl2：使用SDL2播放YUV的例子。
- simplest\_ffmpeg\_helloworld：输出FFmpeg类库的信息。

#### 文章

[《100行代码实现最简单的基于FFMPEG+SDL的视频播放器（SDL1.x）》](#)

[《最简单的基于FFMPEG+SDL的视频播放器 ver2（采用SDL2.0）》](#)

[《最简单的基于FFmpeg的解码器-纯净版（不包含libavformat）》](#)

[《最简单的基于FFMPEG+SDL的视频播放器：拆分-解码器和播放器》](#)

[《最简单的基于FFMPEG的Helloworld程序》](#)

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_player](https://github.com/leixiaohua1020/simplest_ffmpeg_player)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_player](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_player)

## 2.2

### simplest ffmpeg audio player

*最简单的基于FFMPEG的音频播放器*

#### 简介

本程序实现了音频的解码和播放。是最简单的FFmpeg音频解码方面的教程。通过学习本例子可以了解FFmpeg的解码流程。

项目包含3个工程：

simplest\_ffmpeg\_audio\_player：基于FFmpeg+SDL的音频解码器

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。

#### 文章

《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》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_audio\\_player](https://github.com/leixiaohua1020/simplest_ffmpeg_audio_player)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_audio\\_player](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_audio_player)

## 2.3

### simplest ffmpeg picture encoder

*最简单的基于FFMPEG的图像编码器*

#### 简介

本程序实现了YUV420P像素数据编码为JPEG图片。是最简单的FFmpeg编码方面的教程。通过学习本例子可以了解FFmpeg的编码流程。

#### 文章

《最简单的基于FFMPEG的图像编码器（YUV编码为JPEG）》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_picture\\_encoder](https://github.com/leixiaohua1020/simplest_ffmpeg_picture_encoder)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_picture\\_encoder](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_picture_encoder)

## 2.4

### simplest ffmpeg video encoder

*最简单的基于FFMPEG的视频编码器*

#### 简介

本程序实现了YUV像素数据编码为视频码流（H.265，H264，MPEG2，VP8等等）。是最简单的FFmpeg视频编码方面的教程。通过学习本例子可以了解FFmpeg的编码流程。

它包含以下两个子项目：

simplest\_ffmpeg\_video\_encoder：普通版编码器。使用libavcodec和libavformat编码并且封装视频。

simplest\_ffmpeg\_video\_encoder\_pure：纯净版编码器。仅使用libavcodec编码视频，不使用libavformat。

## 文章

《最简单的基于FFMPEG的视频编码器（YUV编码为H.264）》

《最简单的基于FFmpeg的视频编码器-更新版（YUV编码为HEVC(H.265)）》

《最简单的基于FFmpeg的编码器-纯净版（不包含libavformat）》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_video\\_encoder](https://github.com/leixiaohua1020/simplest_ffmpeg_video_encoder)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_video\\_encoder](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_video_encoder)

## 2.5

### simplest ffmpeg audio encoder

*最简单的基于FFMPEG的音频编码器*

#### 简介

本程序实现了音频PCM采样数据编码为压缩码流（MP3，WMA，AAC等）。是最简单的FFmpeg音频编码方面的教程。通过学习本例子可以了解Fmpeg的编码流程。

## 文章

《最简单的基于FFMPEG的音频编码器（PCM编码为AAC）》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_audio\\_encoder](https://github.com/leixiaohua1020/simplest_ffmpeg_audio_encoder)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_audio\\_encoder](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_audio_encoder)

## 2.6

### simplest ffmpeg format

*最简单的基于FFMPEG的封装格式处理*

#### 简介

该解决方案包含了使用FFmpeg进行封装格式处理的各种例子：

simplest\_ffmpeg\_demuxer：视音频分离器。

simplest\_ffmpeg\_demuxer\_simple：视音频分离器（简化版）。

simplest\_ffmpeg\_muxer：视音频复用器。

simplest\_ffmpeg\_remuxer：封装格式转换器。

## 文章

《最简单的基于FFmpeg的封装格式处理：视音频分离器简化版（demuxer-simple）》

《最简单的基于FFmpeg的封装格式处理：视音频分离器（demuxer）》

《最简单的基于FFmpeg的封装格式处理：视音频复用器（muxer）》

《最简单的基于FFMPEG的封装格式处理：封装格式转换（remuxer）》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_format](https://github.com/leixiaohua1020/simplest_ffmpeg_format)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_format](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_format)

## 2.7

### simplest ffmpeg streamer

*最简单的基于FFmpeg的推流器（推送RTMP）*

#### 简介

本例子实现了推送本地视频至流媒体服务器（以RTMP为例）。是使用FFmpeg进行流媒体推送最简单的教程。

#### 文章

《最简单的基于FFmpeg的推流器（以推送RTMP为例）》

《最简单的基于FFMPEG的推流器附件：收流器》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_streamer](https://github.com/leixiaohua1020/simplest_ffmpeg_streamer)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_streamer](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_streamer)

## 2.8

### simplest ffmpeg video filter

*最简单的基于FFmpeg的AVfilter例子（水印叠加）*

#### 简介

本程序使用包含下面两个项目：

simplest\_ffmpeg\_video\_filter：可以将一张PNG图片作为水印叠加到视频上，结合使用了libavfilter，libavcodec等类库。

simplest\_ffmpeg\_video\_filter\_pure：可以给YUV像素数据加特效，只用了libavfilter库。

#### 文章

《最简单的基于FFmpeg的AVfilter例子（水印叠加）》

《最简单的基于FFmpeg的AVfilter的例子-纯净版》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_video\\_filter](https://github.com/leixiaohua1020/simplest_ffmpeg_video_filter)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_video\\_filter](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_video_filter)

## 2.9

### simplest ffmpeg swscale

*最简单的基于FFmpeg的libswscale的示例*

#### 简介

本程序是最简单的基于FFmpeg的libswscale进行像素处理的示例。它包含了两个工程：

simplest\_ffmpeg\_swscale: 最简单的libswscale的教程。

simplest\_pic\_gen: 生成各种测试图片的工具。

#### 文章

《最简单的基于FFmpeg的libswscale的示例（YUV转RGB）》

《最简单的基于FFmpeg的libswscale的示例附件：测试图片生成工具》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_swscale](https://github.com/leixiaohua1020/simplest_ffmpeg_swscale)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_swscale](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_swscale)

## 2.10

### simplest ffmpeg device

*最简单的基于FFmpeg的AVDevice例子*

#### 简介

本工程包含两个基于FFmpeg的libavdevice的例子：

simplest\_ffmpeg\_grabdesktop：屏幕录制。

simplest\_ffmpeg\_readcamera：读取摄像头。

#### 文章

《最简单的基于FFmpeg的AVDevice例子（屏幕录制）》

《最简单的基于FFmpeg的AVDevice例子（读取摄像头）》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_device](https://github.com/leixiaohua1020/simplest_ffmpeg_device)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_device](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_device)

## 2.11

### simplest ffmpeg mem handler

*最简单的基于FFmpeg的内存读写例子*

#### 简介

本程序实现了对内存中的视频数据的读写。包含两个工程：

simplest\_ffmpeg\_mem\_player：播放内存中视频数据的播放器。

simplest\_ffmpeg\_mem\_transcoder：转码内存中数据的转码器。

#### 文章

《最简单的基于FFmpeg的内存读写的例子：内存播放器》

《最简单的基于FFmpeg的内存读写的例子：内存转码器》

#### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_mem\\_handler](https://github.com/leixiaohua1020/simplest_ffmpeg_mem_handler)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_mem\\_handler](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_mem_handler)

## 2.12

### simplest ffmpeg avutil

*最简单的基于FFmpeg的AVUtil例子 (AVLog, AVOption等)*

#### 简介

本程序是FFmpeg中的libavutil的示例，目前包含：

AVLog

AVOption (AVClass)

AVDictionary

## 文章

《最简单的基于FFmpeg的AVUtil例子 (AVLog, AVOption等)》

## 项目主页

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

Github : [https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_avutil](https://github.com/leixiaohua1020/simplest_ffmpeg_avutil)

开源中国 : [http://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_avutil](http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_avutil)

## 2.12

### simplest ffmpeg mobile

*最简单的基于FFmpeg的移动端的例子*

## 简介

本解决方案包含了使用FFmpeg在移动端处理多媒体的各种例子：

[Android]

simplest\_android\_player: 基于安卓接口的视频播放器

simplest\_ffmpeg\_android\_helloworld: 安卓平台下基于FFmpeg的HelloWorld程序

simplest\_ffmpeg\_android\_decoder: 安卓平台下最简单的基于FFmpeg的视频解码器

simplest\_ffmpeg\_android\_decoder\_onelib: 安卓平台下最简单的基于FFmpeg的视频解码器-单库版

simplest\_ffmpeg\_android\_streamer: 安卓平台下最简单的基于FFmpeg的推流器

simplest\_ffmpeg\_android\_transcoder: 安卓平台下移植的FFmpeg命令行工具

simplest\_sdl\_android\_helloworld: 移植SDL到安卓平台的最简单程序

[IOS]

simplest\_ios\_player: 基于IOS接口的视频播放器

simplest\_ffmpeg\_ios\_helloworld: IOS平台下基于FFmpeg的HelloWorld程序

simplest\_ffmpeg\_ios\_decoder: IOS平台下最简单的基于FFmpeg的视频解码器

simplest\_ffmpeg\_ios\_streamer: IOS平台下最简单的基于FFmpeg的推流器

simplest\_ffmpeg\_ios\_transcoder: IOS平台下移植的ffmpeg.c命令行工具

simplest\_sdl\_ios\_helloworld: 移植SDL到IOS平台的最简单程序

[Windows]

simplest\_ffmpeg\_windowsphone\_helloworld: Windows Phone平台下基于FFmpeg的HelloWorld程序

## 文章

最简单的基于FFmpeg的移动端例子：Android HelloWorld

最简单的基于FFmpeg的移动端例子：Android 视频解码器

最简单的基于FFmpeg的移动端例子：Android 视频解码器-单个库版

最简单的基于FFmpeg的移动端例子：Android 推流器

最简单的基于FFmpeg的移动端例子：Android 视频转码器

最简单的基于FFmpeg的移动端例子附件：Android 自带播放器

最简单的基于FFmpeg的移动端例子附件：SDL Android HelloWorld

最简单的基于FFmpeg的移动端例子：IOS HelloWorld

最简单的基于FFmpeg的移动端例子：IOS 视频解码器

最简单的基于FFmpeg的移动端例子：IOS 推流器

最简单的基于FFmpeg的移动端例子：IOS 视频转码器

最简单的基于FFmpeg的移动端例子附件：IOS自带播放器

最简单的基于FFmpeg的移动端例子：Windows Phone HelloWorld

## 项目主页

Github : [https://github.com/leixiaohua1020/simplest\\_ffmpeg\\_mobile](https://github.com/leixiaohua1020/simplest_ffmpeg_mobile)



开源中国：[https://git.oschina.net/leixiaohua1020/simplest\\_ffmpeg\\_mobile](https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_mobile)

### 3.

## FFmpeg移植程序

FFmpeg移植程序是将FFmpeg项目中的命令行工具“ffplay”和“ffmpeg”的源代码（ffplay.c和ffmpeg.c）移植到VC下后编写的程序。移植后的程序保留了原本命令行工具的大部分功能，并且使用MFC做了一套界面，使用起来相对方便一些。其实相比与实际应用来说，这两个工程更加适合作为学习材料。可以通过它们研究播放器，转码器的原理。

#### 3.1

### FFPLAY for MFC

#### 简介

本工程将ffmpeg项目中的ffplay播放器（ffplay.c）移植到了VC的环境下。并且使用MFC做了一套简单的界面。



#### 文章

《ffplay播放器移植VC的工程：ffplay for MFC》

#### 项目主页

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

Github：<https://github.com/leixiaohua1020/ffplaymfc>

开源中国：<http://git.oschina.net/leixiaohua1020/ffplay-for-MFC>

#### 3.2

### FFMPEG for MFC

#### 简介

本工程将ffmpeg工程中的ffmpeg转码器（ffmpeg.c）移植到了VC环境下。并且使用MFC做了一套简单的界面。



#### 文章

《ffmpeg转码器移植VC的工程：ffmpeg for MFC》

#### 项目主页

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

Github：<https://github.com/leixiaohua1020/ffmpegmfc>

开源中国：<http://git.oschina.net/leixiaohua1020/ffmpeg-for-MFC>

### 4.

## 多媒体项目示例程序

除了做了基于FFmpeg的多媒体程序之外，还做了一些基于其他多媒体项目的示例。如下文所列。

#### 4.1

## simplest media play

最简单的视音频播放示例

### 简介

本工程包含了使用各种API（Direct3D，OpenGL，GDI，DirectSound，SDL2）播放多媒体例子。其中音频输入为PCM采样数据。输出至系统的声卡播放出来。视频输入为YUV/RGB像素数据。输出至显示器上的一个窗口播放出来。

通过本工程的代码初学者可以快速学习使用这几个API播放视频和音频的技术。一共包括了如下几个子工程：

- simplest\_audio\_play\_directsound:使用DirectSound播放PCM音频采样数据。
- simplest\_audio\_play\_sdl2:使用SDL2播放PCM音频采样数据。
- simplest\_video\_play\_direct3d:使用Direct3D的Surface播放RGB/YUV视频像素数据。
- simplest\_video\_play\_direct3d\_texture:使用Direct3D的Texture播放RGB视频像素数据。
- simplest\_video\_play\_gdi:使用GDI播放RGB/YUV视频像素数据。
- simplest\_video\_play\_opengl:使用OpenGL播放RGB/YUV视频像素数据。
- simplest\_video\_play\_opengl\_texture:使用OpenGL的Texture播放YUV视频像素数据。
- simplest\_video\_play\_sdl2:使用SDL2播放RGB/YUV视频像素数据。



### 文章

- 《最简单的视音频播放示例1：总述》
- 《最简单的视音频播放示例2：GDI播放YUV, RGB》
- 《最简单的视音频播放示例3：Direct3D播放YUV, RGB（通过Surface）》
- 《最简单的视音频播放示例4：Direct3D播放RGB（通过Texture）》
- 《最简单的视音频播放示例5：OpenGL播放RGB/YUV》
- 《最简单的视音频播放示例6：OpenGL播放YUV420P（通过Texture，使用Shader）》
- 《最简单的视音频播放示例7：SDL2播放RGB/YUV》
- 《最简单的视音频播放示例8：DirectSound播放PCM》
- 《最简单的视音频播放示例9：SDL2播放PCM》

### 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_media\\_play](https://github.com/leixiaohua1020/simplest_media_play)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_media\\_play](http://git.oschina.net/leixiaohua1020/simplest_media_play)

## 4.2

### simplest encoder

最简单的编码器例子

### 简介

该解决方案包含了几个常见的编码器的使用示例：

- simplest\_vpx\_encoder：最简单的基于libvpx的视频编码器
- simplest\_x264\_encoder：最简单的基于libx264的视频编码器
- simplest\_x265\_encoder：最简单的基于libx265的视频编码器

## 文章

《最简单的视频编码器：编译（libx264，libx265，libvpx）》

《最简单的视频编码器：基于libx264（编码YUV为H.264）》

《最简单的视频编码器：基于libx265（编码YUV为H.265）》

《最简单的视频编码器：基于libvpx（编码YUV为VP8）》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_encoder](https://github.com/leixiaohua1020/simplest_encoder)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_encoder](http://git.oschina.net/leixiaohua1020/simplest_encoder)

## 4.3

### simplest libRTMP example

*最简单的基于librtmp的示例*

#### 简介

本工程包含了LibRTMP的使用示例，包含如下子工程：

simplest\_librtmp\_receive: 接收RTMP流媒体并在本地保存成FLV格式的文件。

simplest\_librtmp\_send\_flv: 将FLV格式的视音频文件使用RTMP推送至RTMP流媒体服务器。

simplest\_librtmp\_send264: 将内存中的H.264数据推送至RTMP流媒体服务器。



## 文章

《最简单的基于librtmp的示例：接收（RTMP保存为FLV）》

《最简单的基于librtmp的示例：发布（FLV通过RTMP发布）》

《最简单的基于librtmp的示例：发布H.264（H.264通过RTMP发布）》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_librtmp\\_example](https://github.com/leixiaohua1020/simplest_librtmp_example)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_librtmp\\_example](http://git.oschina.net/leixiaohua1020/simplest_librtmp_example)

## 4.4

### simplest libVLC example

*最简单的基于libVLC的例子*

#### 简介

本工程是包含了一些基于libVLC的示例程序。一共包含了如下几个子程序。

playerGUI: 最简单的基于libVLC的播放器-图形界面版。

simplest\_libvlc\_example: 最简单的基于libVLC的播放器。

simplest\_libvlc\_streamer: 最简单的基于libVLC的推流器。



## 文章

《最简单的基于libVLC的例子：最简单的基于libVLC的视频播放器》

《最简单的基于libVLC的例子：最简单的基于libVLC的视频播放器（图形界面版）》

《最简单的基于libVLC的例子：最简单的基于libVLC的推流器》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_libvlc\\_example](https://github.com/leixiaohua1020/simplest_libvlc_example)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_libvlc\\_example](http://git.oschina.net/leixiaohua1020/simplest_libvlc_example)

## 4.5

### simplest directshow example

*最简单的基于DirectShow的示例*

#### 简介

本程序包含了DirectShow开发的示例程序。适合DirectShow初学者进行学习。

它包含了以下几个子程序：

simplest\_directshow\_player: 最简单的基于DirectShow的视频播放器。

simplest\_directshow\_player\_custom: 最简单的基于DirectShow的视频播放器（Custom）。

playerGUI: 最简单的基于DirectShow的播放器-图形界面版。

simplest\_directshow\_info: 最简单的Directshow信息显示例子。

simplest\_directshow\_filter: 目前还未完成。



## 文章

《最简单的基于DirectShow的示例：视频播放器》

《最简单的基于DirectShow的示例：视频播放器图形界面版》

《最简单的基于DirectShow的示例：视频播放器自定义版》

《最简单的基于DirectShow的示例：获取Filter信息》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_directshow\\_example](https://github.com/leixiaohua1020/simplest_directshow_example)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_directshow\\_example](http://git.oschina.net/leixiaohua1020/simplest_directshow_example)

## 5.

### 视音频编解码学习工程

视音频编解码学习工程主要用于学习编解码的原理。这几个工程都是简单的“码流分析”工具。他们可以分析不同格式的多媒体数据：TS封装格式，FLV封装格式，H.264码流，AAC码流，JPEG压缩图片。

这几个程序我完成的比较早，是我在学习视音频技术初期边看理论边实践着做出来的。因此功能上还有一些不足，一直也没有时间修正，有时间再把它们统一更新一下。

## 5.1

### H264 Stream Analysis

*H.264分析器*

## 文章

《视音频编解码学习工程：H.264分析器》

## 项目主页

SourceForge： <https://sourceforge.net/projects/h264streamanalysis/>

Github： [https://github.com/leixiaohua1020/h264\\_analysis](https://github.com/leixiaohua1020/h264_analysis)

开源中国： [http://git.oschina.net/leixiaohua1020/H264\\_Analysis](http://git.oschina.net/leixiaohua1020/H264_Analysis)

## 5.2

### AAC Stream Analysis

AAC格式分析器

## 文章

《视音频编解码学习工程：AAC格式分析器》

## 项目主页

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

## 5.3

### FLV Format Analysis

FLV封装格式分析器

## 文章

《视音频编解码学习工程：FLV封装格式分析器》

## 项目主页

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

## 5.4

### TS Format Analysis

TS封装格式分析器

## 文章

《视音频编解码学习工程：TS封装格式分析器》

## 项目主页

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

## 5.5

### JPEG Analysis

JPEG 分析器



#### 文章

《视音频编解码学习工程：JPEG分析器》

#### 项目主页

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

## 6.

### 其他语言工程

一些不是使用C/C++写的流媒体相关的工程。

### 6.1

#### Simplest Video Website

最简单的视频网站 (JavaEE+FFmpeg)

#### 简介

本程序是一个最简单的视频网站系统。基于JavaEE的SSH (Strut2 + Spring + Hibernate)和FFmpeg。



#### 文章

《最简单的视频网站 (JavaEE+FFmpeg) 》

#### 项目主页

开源中国 : [http://git.oschina.net/leixiaohua1020/simplest\\_video\\_website](http://git.oschina.net/leixiaohua1020/simplest_video_website)

Github : [https://github.com/leixiaohua1020/simplest\\_video\\_website](https://github.com/leixiaohua1020/simplest_video_website)

在线演示 : <http://www.velab.com.cn:8080/swf/>

### 6.2

#### Simplest flashmedia example

最简单的基于Flash的流媒体示例

#### 简介

本程序是一个Flash流媒体示例的合集。其中包含了使用Actionscript编写的推流器和播放器，以及网页播放器。本工程包含如下基于Flash技术的流媒体的例子：

- simplest\_as3\_rtmp\_player: 最简单的RTMP播放器 (基于ActionScript)
- simplest\_as3\_rtmp\_streamer: 最简单的RTMP推流器 (基于ActionScript)
- rtmp\_sample\_player\_adobe: 从Adobe Flash Media Sever提取出来的测试播放器
- rtmp\_sample\_player\_wowza: 从Wowza服务器中提取出来的测试播放器
- rtmp\_sample\_player\_flowplayer: 基于FlowPlayer的RTMP/HTTP播放器 (添加RTMP plugin)
- rtmp\_sample\_player\_videosjs: 基于VideoJS的RTMP/HTTP播放器
- rtmp\_sample\_player\_jwplayer: 基于JWplayer的RTMP/HTTP播放器
- hls\_sample\_player\_flowplayer: 基于FlowPlayer的HLS播放器 (添加HLS plugin)

hls\_video\_player\_html5: 基于HTML5的HLS/HTTP播放器

activex\_vlc\_player: 基于VLC的ActiveX控件的播放器



## 文章

《最简单的基于Flash的流媒体示例：RTMP推送和接收（ActionScript）》

《最简单的基于Flash的流媒体示例：网页播放器（HTTP，RTMP，HLS）》

## 项目主页

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

Github：[https://github.com/leixiaohua1020/simplest\\_flashmedia\\_example](https://github.com/leixiaohua1020/simplest_flashmedia_example)

开源中国：[http://git.oschina.net/leixiaohua1020/simplest\\_flashmedia\\_example](http://git.oschina.net/leixiaohua1020/simplest_flashmedia_example)

## 7.

### 附录：程序汇总表

#### 1.

##### 专业领域程序

##### 1.1

VideoEye

##### 1.2

TlAndSI

#### 2.

##### FFmpeg示例程序

##### 2.1

##### Simplest ffmpeg player | 最简单的基于FFMPEG的视频播放器

包括了如下几个子工程：

simplest\_ffmpeg\_player：标准版，FFmpeg学习的开始。

simplest\_ffmpeg\_player\_su：SU（SDL Update）版，加入了简单的SDL的Event。

simplest\_ffmpeg\_decoder\_pure：一个纯净的解码器只使用libavcodec（没有使用libavformat）。

##### 2.2

##### simplest ffmpeg audio player | 最简单的基于FFMPEG的音频播放器

##### 2.3

##### simplest ffmpeg picture encoder | 最简单的基于FFMPEG的图像编码器

##### 2.4

##### simplest ffmpeg video encoder | 最简单的基于FFMPEG的视频编码器

包括了如下几个子工程：

simplest\_ffmpeg\_video\_encoder：普通版编码器。使用libavcodec和libavformat编码并且封装视频。

simplest\_ffmpeg\_video\_encoder\_pure：纯净版编码器。仅使用libavcodec编码视频，不使用libavformat。

##### 2.5

##### simplest ffmpeg audio encoder | 最简单的基于FFMPEG的音频编码器

##### 2.6

##### simplest ffmpeg format | 最简单的基于FFMPEG的封装格式处理

包括了如下几个子工程：

simplest\_ffmpeg\_demuxer：视音频分离器。

simplest\_ffmpeg\_demuxer\_simple：视音频分离器（简化版）。

simplest\_ffmpeg\_muxer：视音频复用器。

simplest\_ffmpeg\_remuxer：封装格式转换器。

##### 2.7

##### simplest ffmpeg streamer | 最简单的基于FFmpeg的推流器（推送RTMP）

##### 2.8

##### simplest ffmpeg video filter | 最简单的基于FFmpeg的AVfilter例子（水印叠加）

##### 2.9

##### simplest ffmpeg swscale | 最简单的基于FFmpeg的libswscale的示例

包括了如下几个子工程：

simplest\_ffmpeg\_swscale: 最简单的libswscale的教程。

simplest\_pic\_gen: 生成各种测试图片的工具。

### 2.10 simplest ffmpeg device | 最简单的基于FFmpeg的AVDevice例子

包括了如下几个子工程：

simplest\_ffmpeg\_grabdesktop：屏幕录制。

simplest\_ffmpeg\_readcamera：读取摄像头。

### 2.11 simplest ffmpeg mem handler | 最简单的基于FFmpeg的内存读写例子

包括了如下几个子工程：

simplest\_ffmpeg\_mem\_player：播放内存中视频数据的播放器。

simplest\_ffmpeg\_mem\_transcoder：转码内存中数据的转码器。

### 2.12 simplest ffmpeg avutil | 最简单的基于FFmpeg的AVUtil例子

### 2.13 simplest ffmpeg mobile | 最简单的基于FFmpeg的移动端的例子

本解决方案包含了使用FFmpeg在移动端处理多媒体的各种例子：

[Android]

simplest\_android\_player: 基于安卓接口的视频播放器

simplest\_ffmpeg\_android\_helloworld: 安卓平台下基于FFmpeg的HelloWorld程序

simplest\_ffmpeg\_android\_decoder: 安卓平台下最简单的基于FFmpeg的视频解码器

simplest\_ffmpeg\_android\_decoder\_onelib: 安卓平台下最简单的基于FFmpeg的视频解码器-单库版

simplest\_ffmpeg\_android\_streamer: 安卓平台下最简单的基于FFmpeg的推流器

simplest\_ffmpeg\_android\_transcoder: 安卓平台下移植的FFmpeg命令行工具

simplest\_sdl\_android\_helloworld: 移植SDL到安卓平台的最简单程序

[IOS]

simplest\_ios\_player: 基于IOS接口的视频播放器

simplest\_ffmpeg\_ios\_helloworld: IOS平台下基于FFmpeg的HelloWorld程序

simplest\_ffmpeg\_ios\_decoder: IOS平台下最简单的基于FFmpeg的视频解码器

simplest\_ffmpeg\_ios\_streamer: IOS平台下最简单的基于FFmpeg的推流器

simplest\_ffmpeg\_ios\_transcoder: IOS平台下移植的ffmpeg.c命令行工具

simplest\_sdl\_ios\_helloworld: 移植SDL到IOS平台的最简单程序

[Windows]

simplest\_ffmpeg\_windowsphone\_helloworld: Windows Phone平台下基于FFmpeg的HelloWorld程序

## 3.

### FFmpeg移植程序

#### 3.1

#### FFPLAY for MFC

#### 3.2

#### FFMPEG for MFC

## 4.

### 多媒体项目示例程序

#### 4.1

#### simplest media play | 最简单的视音频播放示例

包括了如下几个子工程：

simplest\_audio\_play\_directsound:使用DirectSound播放PCM音频采样数据。

simplest\_audio\_play\_sdl2:使用SDL2播放PCM音频采样数据。

simplest\_video\_play\_direct3d:使用Direct3D的Surface播放RGB/YUV视频像素数据。

simplest\_video\_play\_direct3d\_texture:使用Direct3D的Texture播放RGB视频像素数据。

simplest\_video\_play\_gdi:使用GDI播放RGB/YUV视频像素数据。

simplest\_video\_play\_opengl:使用OpenGL播放RGB/YUV视频像素数据。

simplest\_video\_play\_opengl\_texture:使用OpenGL的Texture播放YUV视频像素数据。

simplest\_video\_play\_sdl2:使用SDL2播放RGB/YUV视频像素数据。

#### 4.2

#### simplest encoder | 最简单的编码器例子

包括了如下几个子工程：

simplest\_vpx\_encoder：最简单的基于libvpx的视频编码器

simplest\_x264\_encoder：最简单的基于libx264的视频编码器

simplest\_x265\_encoder：最简单的基于libx265的视频编码器

#### 4.3

#### simplest libRTMP example | 最简单的基于librtmp的示例

包括了如下几个子工程：

simplest\_librtmp\_receive: 接收RTMP流媒体并在本地保存成FLV格式的文件。

simplest\_librtmp\_send\_flv: 将FLV格式的视音频文件使用RTMP推送至RTMP流媒体服务器。

simplest\_librtmp\_send264: 将内存中的H.264数据推送至RTMP流媒体服务器。



#### 4.4

##### simplest libVLC example | 最简单的基于libVLC的例子

包括了如下几个子工程：

playerGUI: 最简单的基于libVLC的播放器-图形界面版。

simplest\_libvlc\_example: 最简单的基于libVLC的播放器。

simplest\_libvlc\_streamer: 最简单的基于libVLC的推流器。

#### 4.5

##### simplest directshow example | 最简单的基于DirectShow的示例

包括了如下几个子工程：

simplest\_directshow\_player: 最简单的基于DirectShow的视频播放器。

simplest\_directshow\_player\_custom: 最简单的基于DirectShow的视频播放器（Custom）。

playerGUI: 最简单的基于DirectShow的播放器-图形界面版。

simplest\_directshow\_info: 最简单的Directshow信息显示例子。

simplest\_directshow\_filter: 目前还未完成。

### 5. 音视频编解码学习工程

#### 5.1

##### H264 Stream Analysis | H.264分析器

#### 5.2

##### AAC Stream Analysis | AAC格式分析器

#### 5.3

##### FLV Format Analysis | FLV封装格式分析器

#### 5.4

##### TS Format Analysis | TS封装格式分析器

#### 5.5

##### JPEG Analysis | JPEG分析器

### 6. 其他语言工程

#### 6.1

##### Simplest Video Website | 最简单的视频网站（JavaEE+FFmpeg）

#### 6.2

##### Simplest flashmedia example | 最简单的基于Flash的流媒体示例

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/42658139>

文章标签： [开源项目](#) [ffmpeg](#) [VLC](#) [视频](#) [解码](#)

个人分类：[我的开源项目](#)

此PDF由spygg生成, 请尊重原作者版权!!!

我的邮箱: liushidc@163.com