



# 自定义视频播放操作手册

——基于秉火 STM32F429 开发板

修订历史

日期	版本	更新内容
2015/11/7	1.0.0	-

## 文档说明

本手册旨在帮助用户正确使用秉火 STM32F429 开发板视频播放例程，最终实现播放自定义视频文件。

# 目录

自定义视频播放操作手册 .....	1
文档说明.....	2
目录.....	3
1. AVI 格式视频文件.....	4
1.1 AVI 的主要参数 .....	4
1.1.1 视频参数.....	4
1.1.2 音频参数.....	4
1.1.3 压缩算法.....	5
1.1.4 图像深度: .....	5
1.1.5 压缩质量: .....	5
1.1.6 关键帧: .....	5
1.1.7 数据率: .....	5
1.2 AVI 格式 .....	6
1.2.1 RIFF 结构.....	6
1.2.2 LIST 结构.....	6
1.2.3 CHUNK 结构.....	7
1.2.4 AVI 文件格式结构图 .....	8
2. 使用开发板播放视频.....	9
2.1 播放自定义视频操作步骤.....	9
2.2 视频播放不成功可能问题及解决方法 .....	14

# 1. AVI 格式视频文件

AVI 是音频视频交错(Audio Video Interleaved)的英文缩写,它是 Microsoft 公司开发的一种符合 RIFF 文件规范的数字音频与视频文件格式。AVI 文件格式,多用于音视频捕捉、编辑、回放等应用程序中。通常情况下,一个 AVI 文件可以包含多个不同类型的媒体流(典型的情况下有一个音频流和一个视频流),不过含有单一音频流或单一视频流的 AVI 文件也是合法的。AVI 可以算是 Windows 操作系统上最基本的、也是最常用的一种媒体文件格式。AVI 格式允许视频和音频交错在一起同步播放,支持 256 色和 RLE 压缩,但 AVI 文件并未限定压缩标准,因此,AVI 文件格式只是作为控制界面上的标准,不具有兼容性,用不同压缩算法生成的 AVI 文件,必须使用相应的解压缩算法才能播放出来。

## 1.1 AVI 的主要参数

### 1.1.1 视频参数

- 1) 视窗尺寸 (Video size): AVI 的视窗大小可按 4:3 的比例或随意调整,视窗越大,数据量越大。
- 2) 帧率 (Frames per second): 帧率也可以调整,而且与数据量成正比。不同的帧率会产生不同的效果。

表格 1-1 帧率对视频效果影响

帧率 (fps)	文件大小 (KB)	效果
6	217	画面出现跳动的不连续感
15	637	画面基本连续,是实际应用中较常使用的参数
25	1134	理想的帧率,但数据量太大

### 1.1.2 音频参数

在 AVI 文件中,视频和音频是分别存储的,因此可以把一段视频中的图像与另一段视频中的声音组合在一起。AVI 文件与 WAV 文件密切相关,因为 WAV 文件是 AVI 文件中音频信号的来源,音频的基本参数也即 WAV 格式的参数。除此以外,AVI 文件还包括与音频有关的其他参数:

- 1) 视频与音频的交织参数 (Interlace Audio Every X Frames): AVI 格式中每 X 帧交织存储的音频信号,也即音频和图像交替的频率。X 是可调参数,X 的最小值是一帧,即每个视频帧与音频数据交织组织,这是 CD-ROM 上使用的默认值。交织参数越小,回放 AVI 文件时读到内存中的数据流越少,回放越连续。因此,如果 AVI 文件的存储平台的数据传输率较大,则交错参数可设置得高一些,如几帧,甚至 1 秒。

- 2) 同步控制 (Synchronization): 在 AVI 文件中, 图像和音频是同步得很好的。但实际上由于 CPU 处理能力的不够, 回放 AVI 时有可能出现图像和音频不同步的现象。当 AVI 文件的数据率较高, 而 MPC 的处理速度不够时, 容易出现图像和音频不同步的现象。如视频中人张嘴说话, 但声音并没有发出来。设置同步控制可保证在不同的 MPC 环境下播放该 AVI 文件时都能同步。此时播放程序自动地丢掉一些中间帧以保证视频和音频的同步。

### 1.1.3 压缩算法

压缩算法是首先要确定的一个参数。与 MPEG 标准不同的是, AVI 采用的压缩算法并无统一的标准。也就是说, 同样是以 AVI 为后缀的视频文件, 其采用的压缩算法可能不同, 需要相应的解压缩软件才能识别和回放该 AVI 文件。Microsoft 公司推出 AVI 文件格式和 VFW 软件时, 同时也推出了一种压缩算法, 由于 AVI 和 VFW 的开放性, 其它的公司也相应推出了其它压缩算法, 只要把该算法的驱动加到 Windows 系统中, 就可以在 Windows 系统中播放用该算法压缩的 AVI 文件。

### 1.1.4 图像深度:

与静态图像一样, 视频的图像深度决定其可以显示的颜色数。某些编码 (压缩算法) 使用固定的图像深度, 在这种情况下该参数不可调整。较小的图像深度可以减小文件的容量, 但同时也降低了图像的质量。

### 1.1.5 压缩质量:

选择了一种压缩算法后还可以调整压缩质量, 这个参数常用百分比来表示, 100% 表示最佳效果压缩。同一种压缩算法下, 压缩质量越低, 文件容量越小, 丢失信息越多。

### 1.1.6 关键帧:

关键帧 (Key Frame) 是其他帧压缩时与之比较并产生差值的基准。关键帧可以不压缩, 而中间帧 (也称作差值帧) 是根据其与关键帧的差异来压缩的。采用关键帧压缩可以使压缩比更小而回放速率更快, 但在一段视频文件中访问某一帧的时间将延长。该参数只有在使用帧间压缩编码如帧间差值编码时才起作用。如果不设置关键帧, 则编码器默认每一帧都是关键帧。

### 1.1.7 数据率:

根据其他参数, 可以计算出 AVI 文件的数据率, 一般以每秒兆比特计 (MB/s)。数据率是 AVI 文件的一个重要参数。实际播放 AVI 文件时, 从某种意义上说文件的数据率只能

起到为播放平台设置初始的数据传输率的作用。如果 AVI 文件的数据率过高，而播放该 AVI 文件 MPC 达不到要求，则播放时可能出现不同步或者丢帧现象。因此，要根据播放环境的要求确定 AVI 的数据率，然后根据数据率的要求再确定其它参数。

## 1.2 AVI 格式

AVI 文件采用的是 RIFF 文件结构方式，RIFF（Resource Interchange File Format，资源互换文件格式）是微软公司定义的一种用于管理 windows 环境中多媒体数据的文件格式，RIFF 文件使用四字符码 FOURCC，来表征数据类型，比如‘RIFF’、‘AVI’、‘LIST’等。注意，Windows 操作系统使用的字节顺序是 little-endian，因此一个四字符码‘abcd’实际的 DWORD 值应为 0x64636261。另外，四字符码中像‘AVI’一样含有空格也是合法的。波形音频 wave，MIDI 和数字视频 AVI 都采用这种格式存储。

### 1.2.1 RIFF 结构

RIFF 文件的实际数据中，通常还使用了列表（List）和块（Chunk）的形式来组织。列表可以嵌套子列表和块。

- 1) “RIFF”表示字串
- 2) RIFF 文件大小
- 3) 形式类型或者列表类型“AVI”、“WAVE”...
- 4) RIFF 数据



图 1-1 RIFF 结构

```
00000000h: 52 49 46 46 48 51 D5 02 41 56 49 20 4C 49 53 54 ; RIFFHQ?AVI LIST
00000010h: E0 01 00 00 68 64 72 6C 61 76 69 68 38 00 00 00 ; ...hdrlavih8...
```

图 1-2 RIFF 结构示例

RIFF 文件大小 = 时间资料长度 + 4（形式类型或者列表类型的大小）；也就是说，档大小的值不包括‘RIFF’（‘LIST’）域和“档大小”域本身的大小。（如该位置数据为 28 69 6B 00，则实际为 0x006B6928，转换成十进制为 7039272，实际在 windows 下看到的文件大小为 7039280 bytes，即换算成实际大小时要加上 8）。

### 1.2.2 LIST 结构

RIFF 块中包含一系列的子块，其中有一种字块的 ID 为“LIST”，称为 LIST，LIST 块中可以再包含一系列的子块，但除了 LIST 块外的其他所有的子块都不能再包含子块。

- 1) “LIST”表示字串

- 2) LIST 文件大小
- 3) 形式类型或者列表类型
- 4) LIST 数据

LIST 文件大小 = 实际的列表数据长度 + 4（形式类型或者列表类型的大小）；也就是说 listSize 值不包括‘LIST’域和 listSize 域本身的大小。（如该位置数据为 5E 00 00 00，即 0x0000005E，转换成十进制为 94，实际总长为 102 bytes，即换算成列表总长是要加上 8。但是有一点要特别注意的是，当前 List 具体包括到哪里，可能会有 List 嵌套。如截图中，选中的部分为一个 hdrI 的 List；其中包含两个 strI 的 List，一个音频，一个视频；每个 strI List 又包含一个 strh 和一个 strf）。

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
00000000h:	52	49	46	46	FE	95	B1	00	41	56	49	20	4C	49	53	54	; RIFF?AVI LIST
00000010h:	72	01	00	00	68	64	72	6C	61	76	69	68	38	00	00	00	; r...hdrIavih8...
00000020h:	6A	04	01	00	00	00	00	00	00	00	00	00	10	09	00	00	; j.....
00000030h:	E8	0B	00	00	00	00	00	00	02	00	00	00	00	00	00	00	; ?.....
00000040h:	40	01	00	00	F0	00	00	00	00	00	00	00	00	00	00	00	; @...?.....
00000050h:	00	00	00	00	00	00	00	00	4C	49	53	54	C0	00	00	00	; .....LIST?..
00000060h:	73	74	72	6C	73	74	72	68	38	00	00	00	76	69	64	73	; strIstrh8...vids
00000070h:	58	56	49	44	00	00	00	00	00	00	00	00	00	00	00	00	; XVID.....
00000080h:	01	00	00	00	0F	00	00	00	00	00	00	00	E8	0B	00	00	; .....?..
00000090h:	6E	68	00	00	00	00	00	00	00	00	00	00	00	00	00	00	; nh.....
000000a0h:	40	01	F0	00	73	74	72	66	28	00	00	00	28	00	00	00	; @.?strf{...(...
000000b0h:	40	01	00	00	F0	00	00	00	01	00	0C	00	58	56	49	44	; @...?.....XVID
000000c0h:	00	C2	01	00	00	00	00	00	00	00	00	00	00	00	00	00	; .?.....
000000d0h:	00	00	00	00	76	70	72	70	44	00	00	00	00	00	00	00	; ....vprpD.....
000000e0h:	00	00	00	00	0F	00	00	00	40	01	00	00	F0	00	00	00	; .....@...?..
000000f0h:	03	00	04	00	40	01	00	00	F0	00	00	00	01	00	00	00	; ....@...?.....
00000100h:	F0	00	00	00	40	01	00	00	F0	00	00	00	40	01	00	00	; ?..@...?..@...
00000110h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	; .....
00000120h:	4C	49	53	54	5E	00	00	00	73	74	72	6C	73	74	72	68	; LIST^...strIstrh
00000130h:	38	00	00	00	61	75	64	73	50	00	00	00	00	00	00	00	; 8...audsP.....
00000140h:	00	00	00	00	14	00	00	00	80	04	00	00	44	AC	00	00	; .....€...D?..
00000150h:	00	00	00	00	58	1E	00	00	1B	00	00	00	00	00	00	00	; ....X.....
00000160h:	00	00	00	00	00	00	00	00	00	00	00	00	73	74	72	66	; .....strf
00000170h:	12	00	00	00	50	00	02	00	44	AC	00	00	80	3E	00	00	; ....P...D?..€>..
00000180h:	80	04	00	00	00	00	4C	49	53	54	30	00	00	00	49	4E	; €.....LISTO...IN
00000190h:	46	4F	49	53	46	54	24	00	00	00	4D	45	6E	63	6F	64	; FOISFT\$...MEncod
000001a0h:	65	72	20	53	68	65	72	70	79	61	2D	72	32	30	32	35	; er Sherpya-r2025
000001b0h:	38	2D	4D	69	6E	47	57	2D	34	2E	31	2E	31	00	4A	55	; 8-MinGW-4.1.1.JU
000001c0h:	4E	4B	3A	0E	00	00	5B	3D	20	4D	50	6C	61	79	65	72	; NK:...[= MPlayer
000001d0h:	20	6A	75	6E	6B	20	64	61	74	61	21	20	3D	5D	5B	3D	; junk data! =][=
000001e0h:	20	4D	50	6C	61	79	65	72	20	6A	75	6E	6B	20	64	61	; MPlayer junk da

图 1-3 LIST 数据示例

## 1.2.3 CHUNK 结构

- 1) Chunk ID - 表示块类型的四字符码
- 2) Chunk 文件大小 - 记录了整个块的大小
- 3) Chunk 数据

Chunk 文件大小 = 实际的块数据长度，而不包括 ckID 域和 ckSize 域本身的大小。

## 1.2.4 AVI 文件格式结构图



图 1-4 AVI 文件格式结构图



## 2. 使用开发板播放视频

因 AVI 压缩算法不同对于的解码算法也不同，为获取合适的 AVI 格式文件，我们使用视频格式转换器见任意格式的视频文件转成合适秉火 STM32F429 开发板播放。秉火 STM32F429 开发板例程只编程了使用 libjpeg 解码库解码 jpeg 格式图片以及支持 PCM 数据的音频输出，所以为保证视频在秉火 STM32F429 开发板上支持播放需要将视频格式设置为符合的格式。这里我们使用狸窝全能视频转换器作为视频格式转换工具。

### 2.1 播放自定义视频操作步骤

- 1) 打开狸窝全能视频转换器软件，进入主界面，见图 2-1 狸窝全能视频转换器主界面图 2-1；



图 2-1 狸窝全能视频转换器主界面

- 2) 点击左上角添加视频，选择待转换视频，设置输出目录路径，然后打开输出视频格式设置界面，见图 2-2；



图 2-2 添加视频文件

- 3) 在输出视频格式设置窗口中的预置方案中选择：常规视频->AVI-Audio-Video Interleaved(\*.avi)，见图 2-3；



图 2-3 选择 AVI 格式输出文件

- 4) 设置视频编码器为：MJPEG，视频尺寸为：480\*272，比特率为：1000，帧率为 10；设置音频编码器为：PCM S16LE，采样率为 11025；然后点击确定按钮返回主界面，见图 2-4；

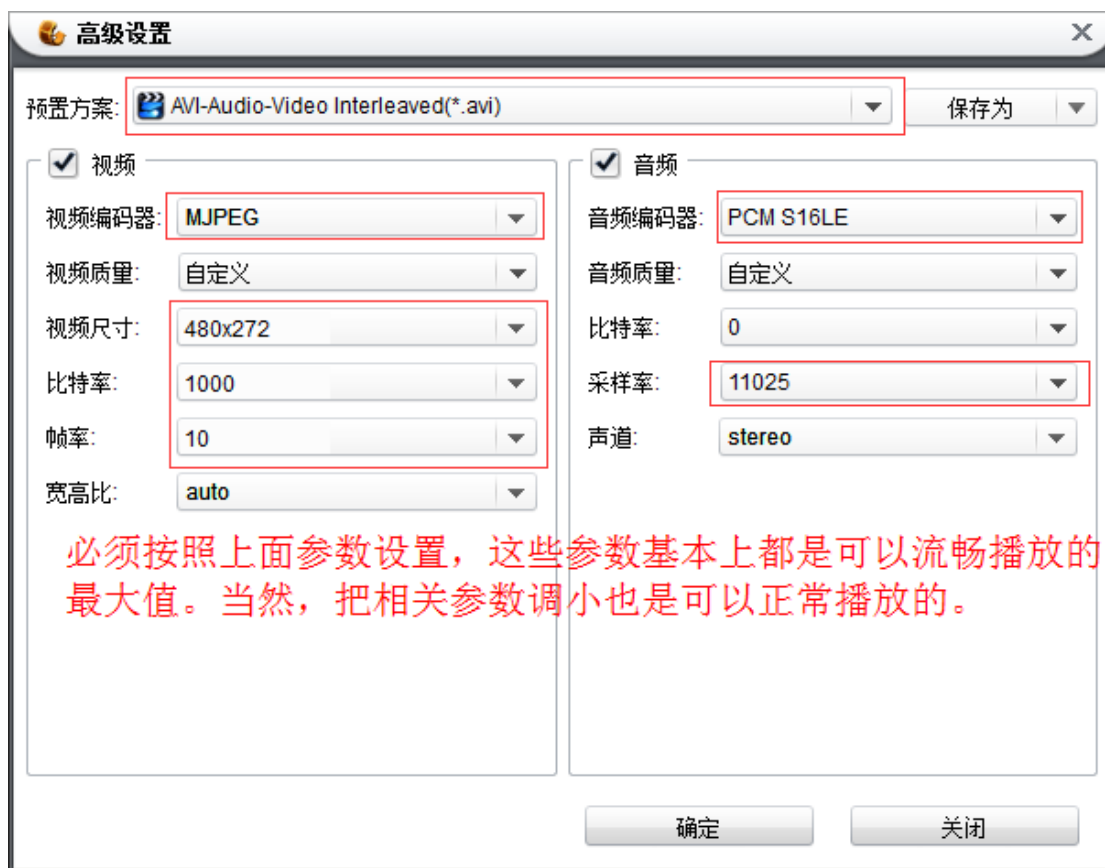


图 2-4 自定义视频和音频格式

5) 点击开始转换按钮，启动转换，见图 2-5；



图 2-5 启动转换

- 6) 等待转换完成，见图 2-6；打开输出目录，拷贝\*.avi 格式文件到 SD 卡根目录下 (SD 卡，读卡器自备)，等待拷贝完成，把 SD 卡插入到开发板上。



图 2-6 转换完成

- 7) 使用 keil 软件打开配套例程：02. AVI Player(libjpeg)，在 keil 软件中打开 main.c 文件；在 main 函数的 while(1)循环中修改 AVI\_play("0:/Thank you.avi");函数中为 SD 卡内视频文件完整路径，见图 2-7。

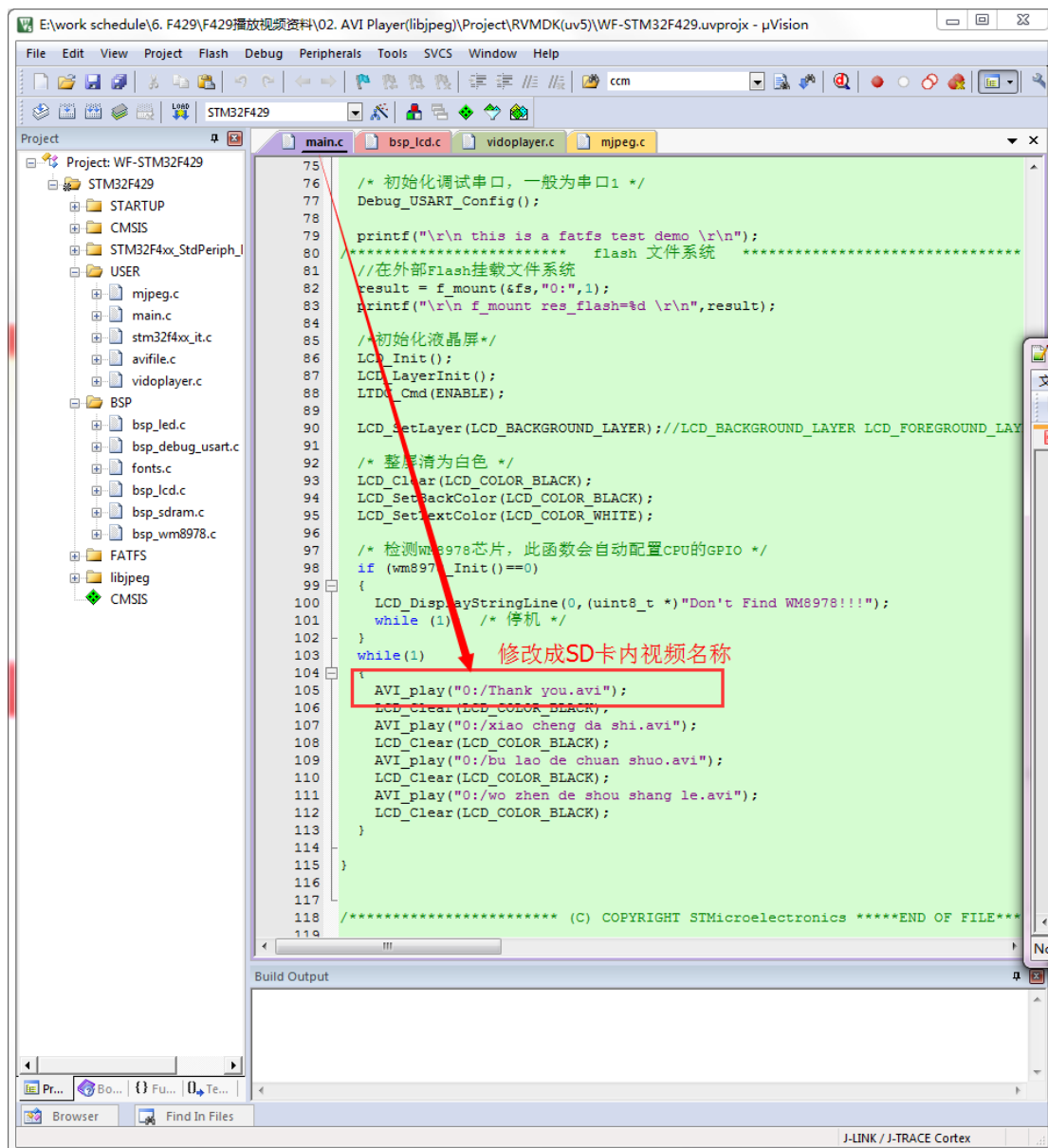


图 2-7 修改程序

- 8) 最后就是编译工程，下载程序。接下来就是观看视频了，可在开发板左上角的音频接口插入耳机(左边那个)亦可欣赏视频配音。

## 2.2 视频播放不成功可能问题及解决方法

情况一：配套例程视频文件播放不了

- 1) 确保视频文件正确拷贝到 SD 卡根目录下，并插入到开发板上；
- 2) 打开串口调试助手软件，例程中会把相关错误信息打印到串口，可以了解错误信息；

- 3) 还是解决不了，把例程视频文件重新拷贝到 SD 卡，再试试。(在测试中发现偶尔播放不了时候重新拷贝文件就可以了)

情况二：自己转换的视频播放不了

- 1) 请严格按照上面步骤转换视频；
- 2) 重新拷贝文件都 SD 卡内。