



播放库

SDK 编程指南

(for iOS)

V7.3.2.X

HIKVISION

杭州海康威视数字技术股份有限公司

<http://www.hikvision.com>

技术热线：400-700-5998

非常感谢您购买我公司的产品，如果您有什么疑问或需要请随时联系我们。

本手册可能包含技术上不准确的地方、或与产品功能及操作不相符的地方、或印刷错误。我司将根据产品功能的增强而更新本手册的内容，并将定期改进或更新本手册中描述的产品或程序。更新的内容将会在本手册的新版本中加入，恕不另行通知。

目 录

| | |
|---------------------------------------------------------|----|
| 目 录 | 1 |
| 1 产品简介..... | 5 |
| 2 SDK 版本更新..... | 6 |
| 版本号说明..... | 6 |
| Version 7.3.2.X..... | 6 |
| 3 错误代码及说明..... | 8 |
| 3.1 基本错误码说明..... | 8 |
| 3.2 鱼眼部分错误码说明..... | 9 |
| 4 函数调用顺序..... | 10 |
| 4.1 播放库调用主要流程..... | 10 |
| 4.2 文件模式..... | 11 |
| 4.3 流模式..... | 13 |
| 4.4 同步回放功能模块..... | 14 |
| 4.5 文件定位模块..... | 14 |
| 4.6 鱼眼功能模块..... | 15 |
| 4.6.1 7.3.1 之前版本鱼眼接口调用流程..... | 15 |
| 4.6.2 7.3.2 版本以后鱼眼接口调用流程..... | 16 |
| 4.7 抓图功能模块..... | 18 |
| 4.8 预录功能模块..... | 18 |
| 4.9 硬解码功能模块..... | 19 |
| 5 函数说明..... | 21 |
| 5.1 系统操作及错误号获取..... | 21 |
| 5.1.1 获取播放库 SDK 版本号和 build 号 PlayM4_GetSdkVersion | 21 |
| 5.1.2 获取错误号 PlayM4_GetLastError | 21 |
| 5.1.3 获取未使用的通道号 PlayM4_GetPort..... | 21 |
| 5.1.4 释放已使用的通道号 PlayM4_FreePort..... | 21 |
| 5.2 文件操作..... | 23 |
| 5.2.1 打开文件 PlayM4_OpenFile | 23 |
| 5.2.2 关闭文件 PlayM4_CloseFile..... | 23 |
| 5.3 流操作..... | 24 |
| 5.3.1 设置流播放模式 PlayM4_SetStreamOpenMode | 24 |
| 5.3.2 获取流播放模式 PlayM4_GetStreamOpenMode* | 24 |
| 5.3.3 打开流 PlayM4_OpenStream | 24 |
| 5.3.4 关闭流 PlayM4_CloseStream..... | 25 |
| 5.3.5 输入流数据 PlayM4_InputData | 25 |
| 5.4 播放控制..... | 26 |
| 5.4.1 开启播放 PlayM4_Play | 26 |
| 5.4.2 关闭播放 PlayM4_Stop..... | 26 |
| 5.4.3 暂停/恢复播放 PlayM4_Pause..... | 26 |

| | | |
|--------|-------------------------------------------------|----|
| 5.4.4 | 快速播放 PlayM4_Fast | 26 |
| 5.4.5 | 慢速播放 PlayM4_Slow | 27 |
| 5.4.6 | 单帧前进 PlayM4_OneByOne* | 27 |
| 5.4.7 | 设置音量 PlayM4_SetVolume* | 27 |
| 5.4.8 | 以独占方式打开声音 PlayM4_PlaySound | 28 |
| 5.4.9 | 关闭声音（独占方式） PlayM4_StopSound | 28 |
| 5.4.10 | 音视频同步 PlayM4_SyncToAudio | 28 |
| 5.4.11 | 设置图像垂直翻转 PlayM4_SetVerticalFlip | 28 |
| 5.4.12 | 设置文件当前播放时间（毫秒） PlayM4_SetPlayedTimeEx* | 29 |
| 5.4.13 | 设置文件当前播放位置 PlayM4_SetPlayPos* | 29 |
| 5.4.14 | 获取文件当前播放位置 PlayM4_GetPlayPos* | 29 |
| 5.4.15 | 设置文件当前播放帧号 PlayM4_SetCurrentFrameNum* | 30 |
| 5.4.16 | 获取文件当前播放帧号 PlayM4_GetCurrentFrameNum* | 30 |
| 5.4.17 | 获取当前播放帧的全局时间 PlayM4_GetSystemTime | 30 |
| 5.5 | 获取播放或解码信息 | 31 |
| 5.5.1 | 获取文件总时间 PlayM4_GetFileTime* | 31 |
| 5.5.2 | 获取文件总帧数 PlayM4_GetFileTotalFrames* | 31 |
| 5.5.3 | 获取当前帧率 PlayM4_GetCurrentFrameRateEx* | 31 |
| 5.5.4 | 获取已播放时间 PlayM4_GetPlayedTime* | 31 |
| 5.5.5 | 获取已解码的视频帧数 PlayM4_GetPlayedFrames* | 32 |
| 5.5.6 | 获取原始图像大小 PlayM4_GetPictureSize | 32 |
| 5.5.7 | 获取当前显示帧的全局时间 PlayM4_GetSpecialData* | 32 |
| 5.6 | 解码操作及控制 | 34 |
| 5.6.1 | 设置解码回调流类型 PlayM4_SetDecCBStream | 34 |
| 5.6.2 | 设置视频帧解码类型 PlayM4_SetDecodeFrameType | 34 |
| 5.6.3 | 解码回调 PlayM4_SetDecCallBack | 34 |
| 5.6.4 | 解码回调（增加用户传递参数） PlayM4_SetDecCallBackMend | 35 |
| 5.6.5 | 文件结束回调 PlayM4_SetFileEndCallback | 35 |
| 5.6.6 | 解码密钥 PlayM4_SetSecretKey | 36 |
| 5.6.7 | 加密码流回调 PlayM4_SetEncryptTypeCallBack* | 36 |
| 5.7 | 显示操作 | 38 |
| 5.7.1 | 设置或增加显示区域 PlayM4_SetDisplayRegion | 38 |
| 5.7.2 | 设置显示窗口 PlayM4_SetVideoWindow | 38 |
| 5.7.3 | 设置同步回放组 PlayM4_SetSycGroup | 39 |
| 5.8 | 缓冲区操作 | 40 |
| | 解码前源缓冲区 | 40 |
| 5.8.1 | 获取源缓冲区剩余数据大小 PlayM4_GetSourceBufferRemain | 40 |
| | 解码后播放缓冲区 | 40 |
| 5.8.2 | 设置播放缓冲区最大缓冲帧数 PlayM4_SetDisplayBuf | 40 |
| 5.8.3 | 获取播放缓冲区最大缓冲帧数 PlayM4_GetDisplayBuf | 41 |
| | 源、播放缓冲区 | 41 |
| 5.8.4 | 清空所有缓冲区的剩余数据 PlayM4_ResetSourceBuffer | 41 |
| 5.8.5 | 清空指定缓冲区的剩余数据 PlayM4_ResetBuffer | 41 |
| 5.8.6 | 获取指定缓冲区的大小 PlayM4_GetBufferValue | 42 |

| | | |
|---------|-----------------------------------------------------------------|----|
| 5.9 | 索引..... | 43 |
| 5.9.1 | 设置建立索引回调 PlayM4_SetFileRefCallBack | 43 |
| 5.10 | 抓图..... | 44 |
| 5.10.1 | 抓图回调 PlayM4_SetDisplayCallBack..... | 44 |
| 5.10.2 | 抓图回调 PlayM4_SetDisplayCallBackEx | 44 |
| 5.10.3 | 直接抓取 BMP 图像 PlayM4_GetBMP..... | 45 |
| 5.10.4 | 直接抓取 JPEG 图像 PlayM4_GetJPEG | 45 |
| 5.11 | 硬解码..... | 46 |
| 5.11.1 | 设置硬解码优先 PlayM4_SetHDPriority | 46 |
| 5.11.2 | 获取当前解码类型 PlayM4_GetDecodeEngine | 46 |
| 5.12 | 预录像功能..... | 47 |
| 5.12.1 | 设置预录像数据回调 PlayM4_SetPreRecordCallBack..... | 47 |
| 5.12.2 | 设置预录像开关 PlayM4_SetPreRecordFlag | 49 |
| 5.13 | 鱼眼功能..... | 51 |
| | 7.3.2 及以后版本鱼眼接口..... | 51 |
| 5.13.1 | 开启鱼眼功能 PlayM4_FEC_Enable..... | 51 |
| 5.13.2 | 关闭鱼眼功能 PlayM4_FEC_Disable..... | 51 |
| 5.13.3 | 获取鱼眼矫正处理子端口 PlayM4_FEC_GetPort..... | 51 |
| 5.13.4 | 获取鱼眼矫正处理子端口 PlayM4_FEC_DelPort..... | 52 |
| 5.13.5 | 设置鱼眼矫正参数 PlayM4_FEC_SetParam..... | 52 |
| 5.13.6 | 获取鱼眼矫正参数 PlayM4_FEC_GetParam | 52 |
| 5.13.7 | 设置显示窗口 PlayM4_FEC_SetWnd..... | 52 |
| 5.13.8 | 获取当前选择鱼眼子端口 PlayM4_FEC_GetCurrentPTZPort | 54 |
| 5.13.9 | 设置当前选择鱼眼子端口 PlayM4_FEC_SetCurrentPTZPort | 54 |
| 5.13.10 | 获取当前子端口号对应的当前 PTZ 坐标值 PlayM4_FEC_SetPTZOutLineShowMode | 55 |
| 5.13.11 | 获取当前子端口号对应的当前 PTZ 坐标值 PlayM4_FEC_PTZ2Window | 55 |
| | 7.3.1 及以前版本鱼眼接口..... | 56 |
| 5.13.12 | 广角图像矫正 PlayM4_SetImageCorrection | 56 |
| 5.13.13 | 设置鱼眼矫正类型 PlayM4_SetFECDisplayEffect..... | 56 |
| 5.13.14 | 设置鱼眼矫正参数 PlayM4_SetFECDisplayParam | 56 |
| 5.13.15 | 获取鱼眼矫正参数 PlayM4_GetFECDisplayParam..... | 57 |
| 5.14 | 私有数据..... | 58 |
| 5.14.1 | 渲染私有数据 PlayM4_RenderPrivateData | 58 |
| 5.14.2 | 渲染私有数据 PlayM4_RenderPrivateDataEx | 58 |
| 6 | 数据结构描述..... | 59 |
| 6.1 | 图像和声音信息结构体 FRAME_INFO | 59 |
| 6.2 | 显示信息结构体 DISPLAY_INFO | 59 |
| 6.3 | 全局时间结构体 PLAYM4_SYSTEM_TIME..... | 60 |
| 6.4 | 区域参数结构体 HKRECT..... | 60 |
| 6.5 | 加密信息结构体 ENCRYPT_INFO..... | 61 |
| 6.6 | 私有数据显示信息类型结构体 PLAYM4_PRIDATA_RENDER | 61 |
| 6.7 | 热成像信息结构体 PLAYM4_FIRE_ALARM | 61 |
| 6.8 | 测温信息结构体 PLAYM4_TEM_FLAG | 62 |

| | | |
|------|---------------------------------------------------|----|
| 6.9 | 鱼眼矫正显示效果结构体 VRDISPLAYEFFECT..... | 62 |
| 6.10 | 鱼眼矫正参数结构体 VRFISHPARAM..... | 62 |
| 6.11 | 鱼眼安装方式 FECPLACETYPE..... | 63 |
| 6.12 | 鱼眼矫正方式 FECCORRECTTYPE..... | 63 |
| 6.13 | PTZ 在原始鱼眼图上轮廓的显示模式 FECSHOWMODE..... | 63 |
| 6.14 | 鱼眼图像圆心参数 CYCLEPARAM..... | 64 |
| 6.15 | PTZ 校正的参数结构体 PTZPARAM..... | 64 |
| 6.16 | 鱼眼矫正参数结构体 FISHEYEPARAM..... | 64 |
| 7 | 常见问题解答..... | 65 |
| | 移动播放库特有问题介绍..... | 65 |
| | 关于流模式接口..... | 67 |
| | Question 1 送入到播放库 PlayM4_OpenStream 接口报错 | 67 |
| | 流模式预览..... | 67 |
| | Question 2 实时流预览有卡顿..... | 67 |
| | Question 3 实时流预览不显示图像..... | 67 |
| | Question 4 实时流预览延时..... | 68 |
| | Question 5 实时流预览显示第一帧画面比较慢..... | 68 |
| | Question 6 加密码流无法正常播放..... | 69 |
| | Question 7 如何将捕获的图片保存在内存中以供后续处理..... | 69 |
| | Question 8 音频播放卡顿，视频播放流畅..... | 69 |
| | Question 9 复合流文件，播放音频时画面卡顿，不播放音频。则画面正常..... | 69 |
| | Question 10 播放文件，无法听到声音..... | 69 |
| | Question 11 实时流播放正常，文件播放有快放现象..... | 70 |
| | Question 12 文件打开失败..... | 70 |
| | Question 13 文件模式如何定位..... | 70 |
| | Question 14 PlayM4_SetCurrentFrameNum 定位不准确 | 70 |
| | Question 15 如何获取多路数据流解码后的数据..... | 70 |
| | Question 16 实时流数据用解码回调有丢帧现象..... | 70 |
| | Question 17 有些码流解码回调函数中获取的每帧视频高度只有显示原来的一半..... | 71 |

1 产品简介

海康威视播放库 SDK（以下简称“播放库 SDK”）是海康威视嵌入式网络硬盘录像机、视频服务器、IP 设备的配套产品的播放相关的二次开发包，适用于海康编码产品数据流的解码与播放。

播放库 SDK 主要功能：

主要用于实时码流预览，录像文件回放，播放控制如：暂停；获取码流基本信息，如文件索引、解码帧信息，分辨率、帧率；支持 JPG 和 BMP 两种形式下的播放截图。

系统要求：

支持 iOS6.0 及以上版本 iOS 系统。

2 SDK 版本更新

版本号说明

播放库 SDK 版本号自 V7.0.2.X 起，规定版本号定义如下：

V 主版本号 . 子版本号 . 修正版本号 . 保留版本号

- 主版本号升级：工程作大规模改动、重构或优化
- 子版本号升级：功能增加
- 修正版本号升级：局部修改，bug 修正

Version 7.3.2.X

Version 7.3.2.X (2016-04)

- 支持鱼眼多画面 ptz 矫正；
- 支持 pos 信息、图片叠加外的私有数据显示；
- 支持音视频同步播放；
- 支持 bitcode 编译；

Version 7.3.1.X (2015-11)

- 增加设置同步回放组
[PlayM4_SetSyncGroup](#)
- 增加获取当前解码类型
[PlayM4_GetDecodeEngine](#)
- 修改 PlayM4_GetSdkVersion 的说明
- 删除获取音量和设置音量接口
PlayM4_GetVolume、PlayM4_SetVolume
- 增加常见问题解答

Version 7.3.0.X (2015-04)

- 增加硬解码功能：[PlayM4_SetHDPriority](#)。
- 新增鱼眼矫正相关配置功能：
[PlayM4_SetFECDisplayEffect](#)、
[PlayM4_GetFECDisplayParam](#)、
[PlayM4_SetFECDisplayParam](#)。
- 新增预录像功能：
[PlayM4_SetPreRecordCallBack](#)
[PlayM4_SetPreRecordFlag](#)

Version 7.2.1.X (2014-08)

- 增加广角图像矫正功能。

Version 7.2.0.X (2014-06)

- 修改播放模式全改成时间戳播放，保证实时性。
- 修改简化和文件相关的接口，这次的更新主要是针对流式播放。

Version 7.0.2.X (2012-12-14)

- 增加对 mpeg2 视频的支持
- 增加对 g726, amr, mpeg, aac 音频的支持
- IOS 视频解码只对标准 264 进行过优化，其他为进行优化

3 错误代码及说明

3.1 基本错误码说明

| 错误名称 | 代码 | 说明 |
|------------------------------|----|-----------------|
| PLAYM4_NOERROR | 0 | 没有错误 |
| PLAYM4_PARA_OVER | 1 | 输入参数错误; |
| PLAYM4_ORDER_ERROR | 2 | 调用接口顺序错误 |
| PLAYM4_TIMER_ERROR | 3 | 创建多媒体时钟失败 |
| PLAYM4_DEC_VIDEO_ERROR | 4 | 视频解码失败 |
| PLAYM4_DEC_AUDIO_ERROR | 5 | 音频解码失败 |
| PLAYM4_ALLOC_MEMORY_ERROR | 6 | 分配内存失败 |
| PLAYM4_OPEN_FILE_ERROR | 7 | 打开文件失败 |
| PLAYM4_CREATE_OBJ_ERROR | 8 | 创建线程事件等失败 |
| PLAYM4_BUF_OVER | 11 | 缓冲区满, 输入流失败 |
| PLAYM4_CREATE_SOUND_ERROR | 12 | 创建音频设备失败 |
| PLAYM4_SET_VOLUME_ERROR | 13 | 设置音量失败 |
| PLAYM4_SUPPORT_FILE_ONLY | 14 | 只能在播放文件时才能使用此接口 |
| PLAYM4_SUPPORT_STREAM_ONLY | 15 | 只能在播放流时才能使用此接口 |
| PLAYM4_SYS_NOT_SUPPORT | 16 | 系统不支持 |
| PLAYM4_FILEHEADER_UNKNOWN | 17 | 没有文件头 |
| PLAYM4_VERSION_INCORRECT | 18 | 解码器和编码器版本不对应 |
| PLAYM4_INIT_DECODER_ERROR | 19 | 初始化解码器失败 |
| PLAYM4_CHECK_FILE_ERROR | 20 | 文件太短或码流无法识别 |
| PLAYM4_INIT_TIMER_ERROR | 21 | 初始化多媒体时钟失败 |
| PLAYM4_BLT_ERROR | 22 | 显示错误 |
| PLAYM4_OPEN_FILE_ERROR_MULTI | 24 | 打开文件错误,数据类型为混合流 |
| PLAYM4_OPEN_FILE_ERROR_VIDEO | 25 | 打开文件错误,数据类型为视频流 |
| PLAYM4_JPEG_COMPRESS_ERROR | 26 | JPEG 压缩错误 |
| PLAYM4_EXTRACT_NOT_SUPPORT | 27 | 不支持此文件 |
| PLAYM4_EXTRACT_DATA_ERROR | 28 | 提取视频数据错误 |
| PLAYM4_SECRET_KEY_ERROR | 29 | 密码错误 |
| PLAYM4_DECODE_KEYFRAME_ERROR | 30 | 解码关键帧错误 |
| PLAYM4_NEED_MORE_DATA | 31 | 需要更多的数据 |
| PLAYM4_INVALID_PORT | 32 | 无效端口号 |
| PLAYM4_NOT_FIND | 33 | 查找失败 |
| PLAYM4_NEED_LARGER_BUFFER | 34 | 需要更大的缓冲区 |
| PLAYM4_FAIL_UNKNOWN | 99 | 错误 |

3.2 鱼眼部分错误码说明

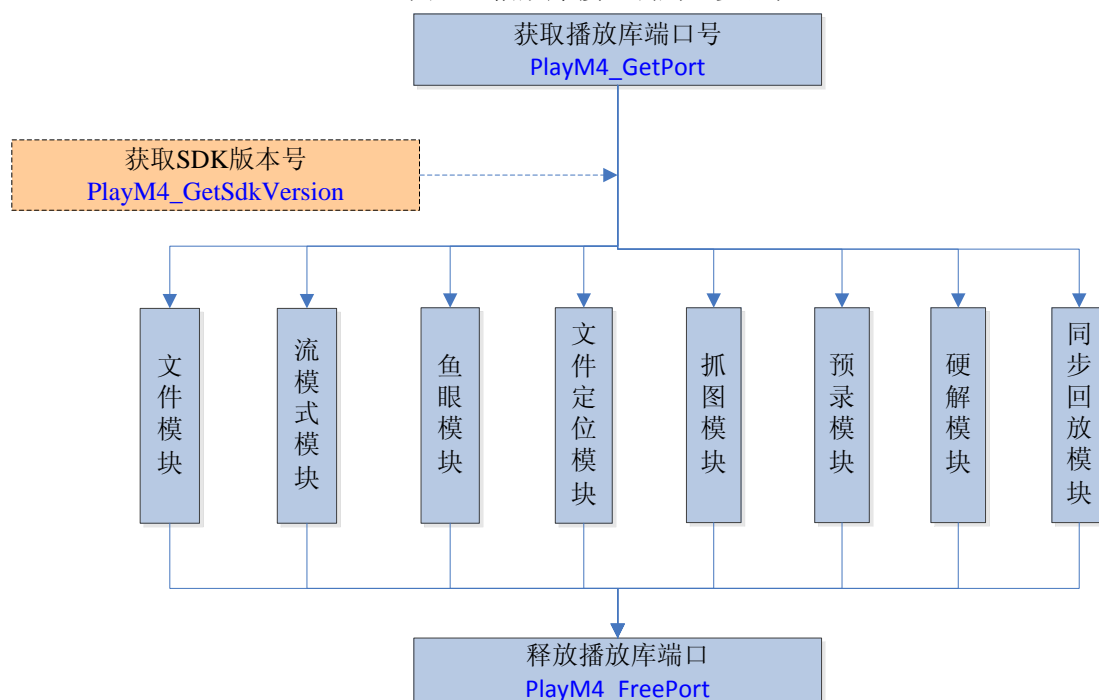
| 错误名称 | 代码 | 说明 |
|---------------------------------|-----|-----------------------|
| PLAYM4_FEC_ERR_ENABLEFAIL | 100 | 鱼眼模块加载失败 |
| PLAYM4_FEC_ERR_NOTENABLE | 101 | 鱼眼模块没有加载 |
| PLAYM4_FEC_ERR_NOSUBPORT | 102 | 子端口没有分配 |
| PLAYM4_FEC_ERR_PARAMNOTINIT | 103 | 没有初始化对应端口的参数 |
| PLAYM4_FEC_ERR_SUBPORTOVER | 104 | 子端口已经用完 |
| PLAYM4_FEC_ERR_EFFECTNOTSUPPORT | 105 | 该安装方式下这种效果不支持 |
| PLAYM4_FEC_ERR_INVALIDWND | 106 | 非法的窗口 |
| PLAYM4_FEC_ERR_PTZOVERFLOW | 107 | PTZ 位置越界 |
| PLAYM4_FEC_ERR_RADIUSINVALID | 108 | 圆心参数非法 |
| PLAYM4_FEC_ERR_UPDATENOTSUPPORT | 109 | 指定的安装方式和矫正效果，该参数更新不支持 |
| PLAYM4_FEC_ERR_NOPLAYPORT | 110 | 播放库端口没有启用 |
| PLAYM4_FEC_ERR_PARAMVALID | 111 | 参数为空 |
| PLAYM4_FEC_ERR_INVALIDPORT | 112 | 非法子端口 |
| PLAYM4_FEC_ERR_PTZZOOMOVER | 113 | PTZ 矫正范围越界 |
| PLAYM4_FEC_ERR_OVERMAXPORT | 114 | 矫正通道饱和，最大支持的矫正通道为四个 |
| PLAYM4_FEC_ERR_ENABLED | 115 | 该端口已经启用了鱼眼模块 |
| PLAYM4_FEC_ERR_D3DACCENOTENABLE | 116 | D3D 加速没有开启 |

[返回目录](#)

4 函数调用顺序

4.1 播放库调用主要流程

图 4.1 播放库接口调用主要流程



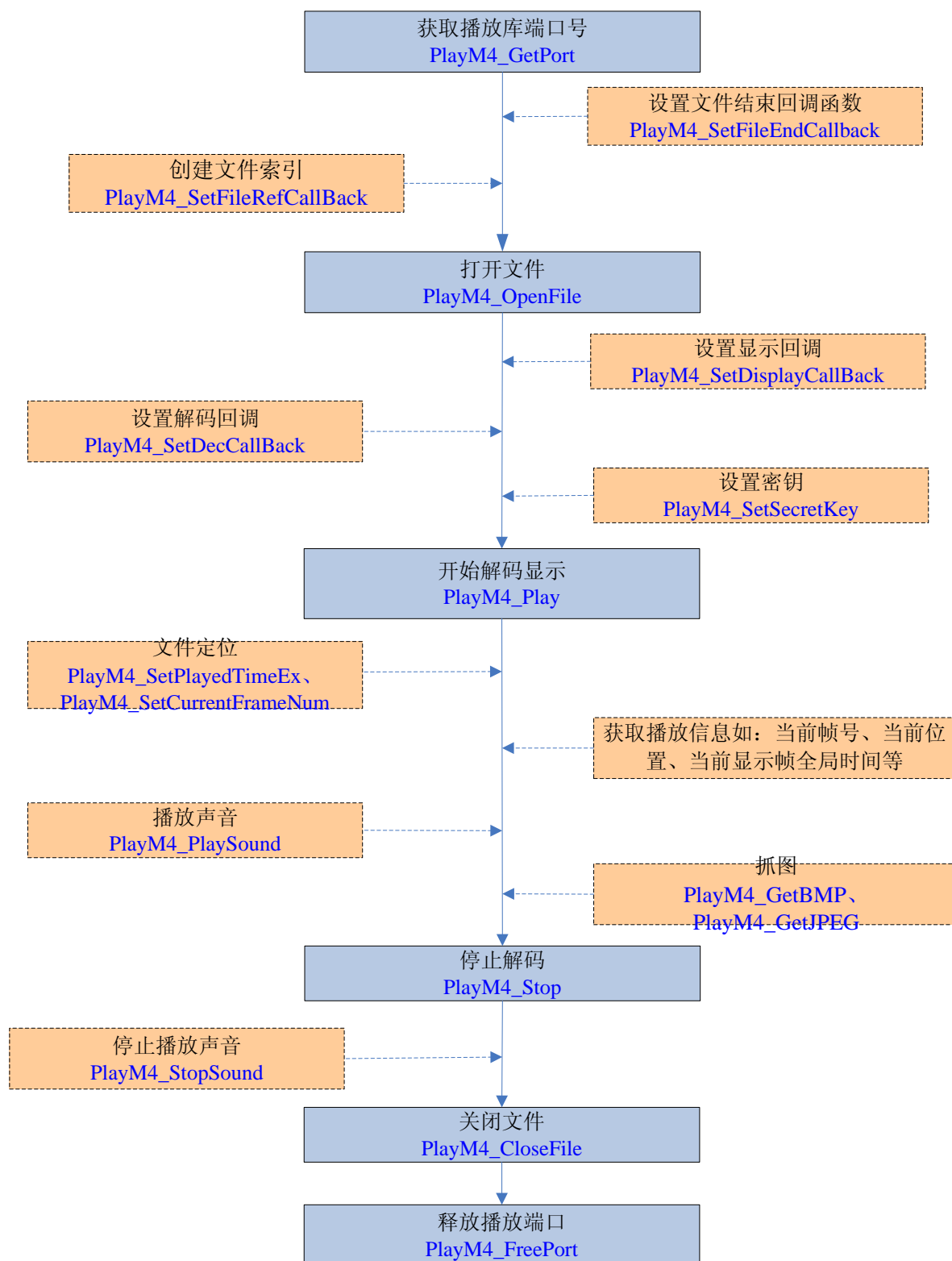
- 文件模式模块:打开待解码的文件,开始解码并显示,且可以获取解码后的音视频数据、定位和抓图等功能。具体接口调用流程请参考【[4.2 文件模式](#)】。
- 流模式模块:传入待解码的实时流或文件流数据,解码并显示。获取解码后的音视频数据、且可以通过抓图回调实现抓图。具体接口调用流程请参考【[4.3 流模式](#)】。
- 同步回放模块:同步回放是指多个文件如果有相同的时间段,多个文件按照相同的时间戳进行播放。具体接口调用流程请参考【[4.4 同步回放功能模块](#)】。
- 文件定位模块:文件方式下播放本地录像文件,播放库 SDK 按照一定的方式如帧号或者时间对文件进行定位。具体接口调用流程请参考【[4.5 文件定位模块](#)】。
- 鱼眼控制模块:鱼镜头与人们眼中真实世界的景象存在很大差别,因此在实际生活中看到的景物是有规则的固定形态,而通过鱼镜头产生的画面效果则超出了这一范畴。实现对鱼镜头这边的画面畸形实现校正。具体接口调用流程请参考【[4.6 鱼眼功能模](#)】。

块】

- 抓图模块：适用于获取当前解码后需要送入到显卡显示的数据。具体接口调用流程请参考【[4.7 抓图功能模块](#)】。
- 预录模块：设备通过网络传输过来的码流，通过回调函数给用户，方便用户保存为本地文件查看。具体接口调用流程请参考【[4.8 预录功能模块](#)】。
- 硬解模块：设定硬解码开关使得播放库优先采用 GPU 解码，替换原先的视频解码模块。具体接口调用流程请参考【[4.9 硬解码功能模块](#)】。

4.2 文件模式

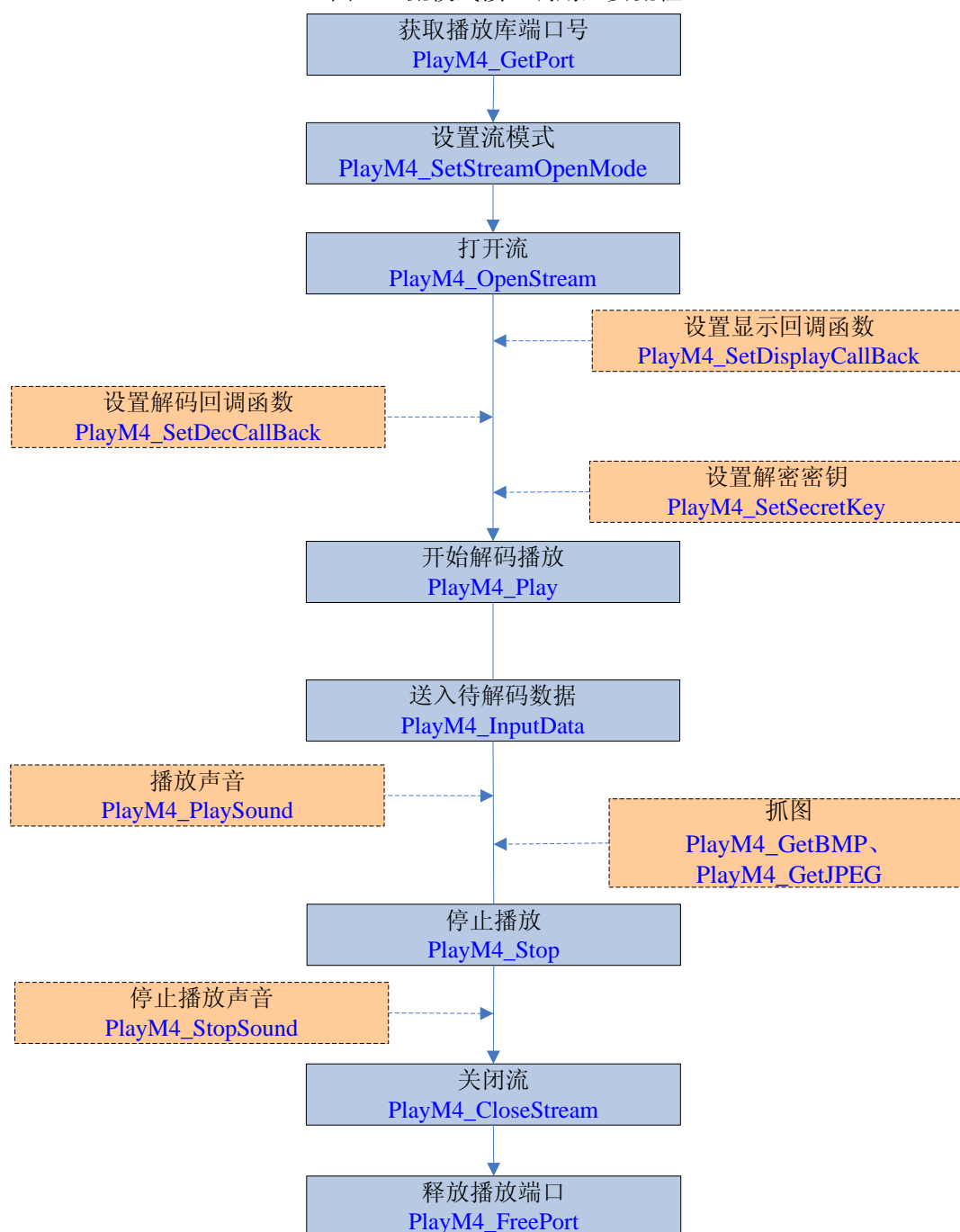
图 4.2 文件模式接口调用主要流程



- 实现文件模块需要先获取播放库端口号 [PlayM4_GetPort](#)，且需要确认文件是海康设备的录像文件。需要注意播放库需要回收端口资源，调用接口 [PlayM4_FreePort](#)。
- 打开文件 [PlayM4_OpenFile](#)，关闭文件调用接口 [PlayM4_CloseFile](#)。
- 打开文件成功，则可以调用播放接口 [PlayM4_Play](#) 进行播放。停止播放调用接口 [PlayM4_Stop](#)。

4.3 流模式

图 4.3 流模式接口调用主要流程



其中虚线框的流程是可选部分，不会影响其他流程和模块的功能使用。

- 流模式模块需要先获取播放库通道号，然后设置流模式（相关接口 [PlayM4_GetPort](#)、[PlayM4_SetStreamOpenMode](#)），送入正确的海康媒体信息头打开数据流（相关接口 [PlayM4_OpenStream](#)），开始解码播放（相关接口 [PlayM4_Play](#)），然后送入正确的海康数据（相关接口 [PlayM4_InputData](#)）。
- 停止解码播放（[PlayM4_Stop](#)），需要关闭流 [PlayM4_CloseStream](#)，需要注意播放库需要回收端口资源，调用接口 [PlayM4_FreePort](#)。

- 实时流需要获取解码后的数据，则可以调用解码回调接口（相关接口 [PlayM4_SetDecCallBack](#)、[PlayM4_SetDecCallBackMend](#)）。
- 多路实时流若都需要获取解码后的数据，每个需要获取解码数据的通道都可以调用解码回调接口（[PlayM4_SetDecCallBack](#)），在解码回调函数中根据 nPort 号判断是具体哪路数据。

4.4 同步回放功能模块

图 4.4 同步回放基本调用过程



- 同步回放是指多个文件在有相同的时间段，然后多个窗口按照相同的时间戳播放。其中需要调用接口（[PlayM4_SetSycGroup](#)）。
- 打开每个待回放的文件 [PlayM4_OpenFile](#)，加入到同步回放组 [PlayM4_SetSycGroup](#) 开始同步回放 [PlayM4_Play](#)。
- 播放结束调用接口 [PlayM4_Stop](#)，关闭文件 [PlayM4_CloseFile](#)。

4.5 文件定位模块

图 4.5 文件定位调用过程



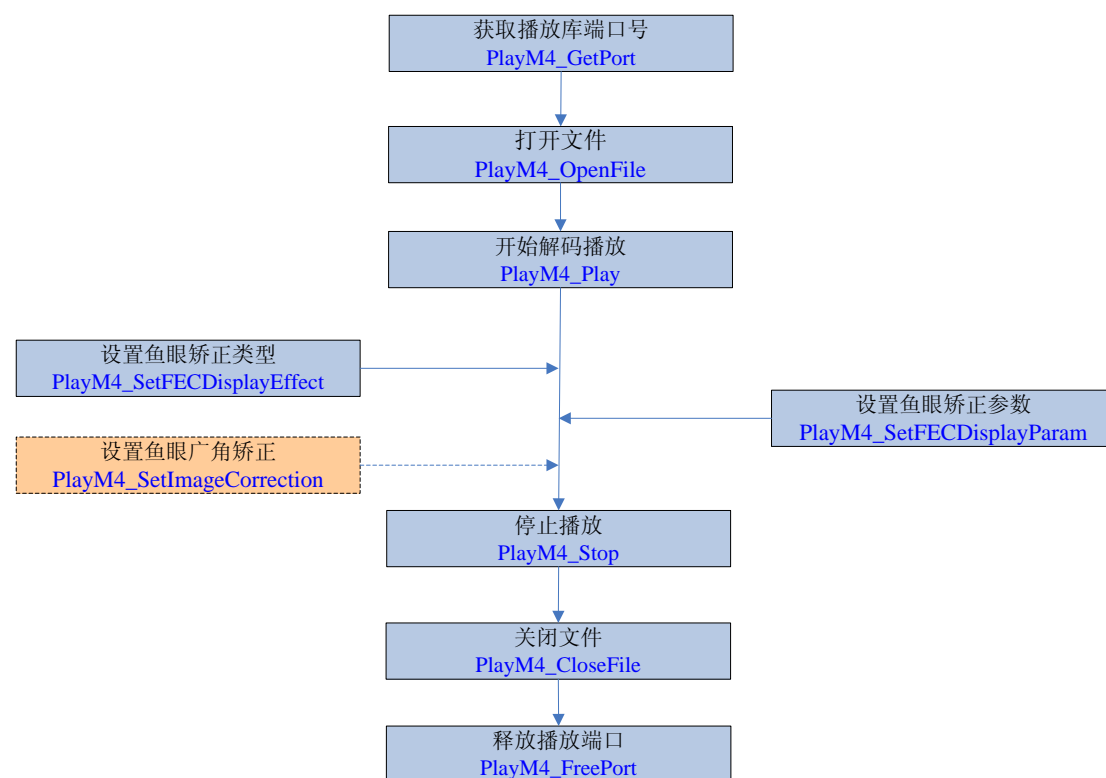
播放本地 PC 上的录像文件，播放库 SDK 支持按照帧号或者时间对文件进行定位，流程如下：

- 先获取播放库端口号（[PlayM4_GetPort](#)）；
- 创建文件索引（[PlayM4_SetFileRefCallBack](#)）；
- 打开待播放的文件（[PlayM4_OpenFile](#)）；
- 开始播放解码（[PlayM4_Play](#)）；
- 按照时间或者帧号进行定位（[PlayM4_SetCurrentFrameNum](#)、[PlayM4_SetPlayPos](#)、[PlayM4_SetPlayedTimeEx](#)）；

4.6 鱼眼功能模块

4.6.1 7.3.1 之前版本鱼眼接口调用流程

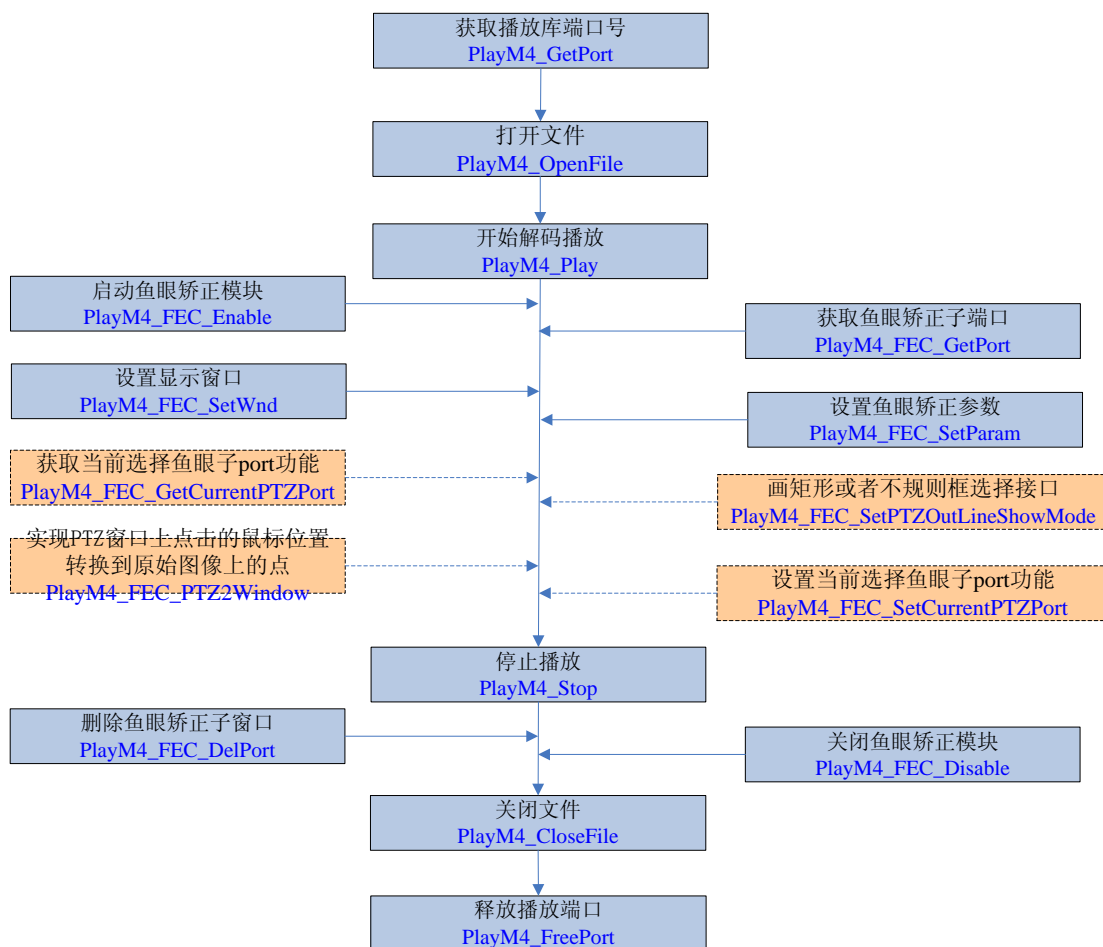
图 4.6 7.3.1 及以前版本鱼眼模块接口调用流程



- 设置 / 获取鱼眼矫正参数：调用的接口有 [PlayM4_GetFECDisplayParam](#)、[PlayM4_SetFECDisplayParam](#)。
- 设置鱼眼矫正类型：[PlayM4_SetFECDisplayEffect](#)。
- 设置广角矫正：调用的接口有 [PlayM4_SetImageCorrection](#)。
- 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。建议使用 7.3.2 及其以后版本的鱼眼接口。

4.6.2 7.3.2 版本以后鱼眼接口调用流程

图 4.7 7.3.2 版本鱼眼模块接口调用流程



鱼眼功能模块是对鱼眼摄像机图像的矫正，使用时需要开启预览模块、获取鱼眼矫正处理子端口、设置鱼眼矫正参数、设置显示窗口、设置鱼眼窗口的绘图回调。

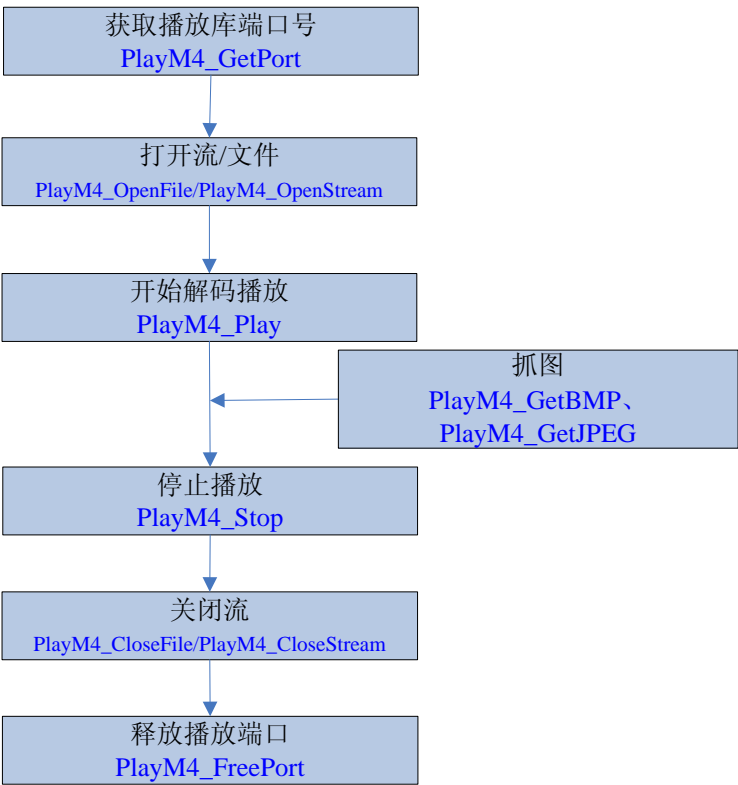
- 开启和关闭鱼眼模块：开启鱼眼模块基本参数，调用的接口 [PlayM4_FEC_Enable](#)、[PlayM4_FEC_Disable](#)。
- 获取/删除鱼眼矫正处理子端口：调用的接口有 [PlayM4_FEC_GetPort](#)、[PlayM4_FEC_DelPort](#)。
- 设置/获取鱼眼矫正参数：调用的接口有 [PlayM4_FEC_SetParam](#)、[PlayM4_FEC_GetParam](#)。
- 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。建议使用 7.3.2 及其以后版本的鱼眼接口。
- 设置显示窗口：调用的接口有 [PlayM4_FEC_SetWnd](#)。

| 安装方式 | 矫正模式 |
|-------------|---------|
| 壁装（Wall） | 纬度拉伸模式 |
| | PTZ 模式 |
| 顶装（Ceiling） | 360 度模式 |
| | 180 度模式 |
| | PTZ 模式 |
| 底装（Floor） | 360 度模式 |
| | 180 度模式 |

| | |
|--|--------|
| | PTZ 模式 |
|--|--------|

4.7 抓图功能模块

图 4.8 抓图功能模块接口调用流程



- 抓图模块适用于获取当前解码后送入到显卡显示的数据（解码后的数据）功能。
- 抓图可保存为两种格式：JPEG 和 BMP。
- 保存 JPEG 调用接口：[PlayM4_GetJPEG](#)。
- 保存 BMP 调用接口：[PlayM4_GetBMP](#)。

4.8 预录功能模块

图 4.9 预录功能模块接口调用流程



- 预录功能模块是设备通过网络传输过来的码流通过回调函数给用户,方便用户保存为本地文件查看。
- 设置预录开关: 设置预录开关标志位,调用的接口 [PlayM4_SetPreRecordFlag](#);
- 设置预录数据回调函数: 调用的接口 [PlayM4_SetPreRecordCallBack](#)。

4.9 硬解码功能模块

图 4.10 硬解码功能模块接口调用流程



- 硬解模块: 设定硬解码开关使得播放库优先采用 GPU 解码, 替换原先的视频解码模块。
- 开启硬解码优先调用接口: [PlayM4_SetHDPriority](#), 符合硬解码条件的码流会优先进行硬解码, 硬解码失败时会主动切换到软解。

5 函数说明

5.1 系统操作及错误号获取

5.1.1 获取播放库 SDK 版本号和 build 号 **PlayM4_GetSdkVersion**

| | | |
|------|---------------------------------------------------------------|--|
| 函 数: | unsigned int PlayM4_GetSdkVersion() | |
| 参 数: | 无 | |
| 返回值: | 主版本号升级：工程做大规模改动、重构或优化； 子版本号升级：功能增加； 修改版本号升级：局部修改，bug 修正 | |
| 说 明: | 如果只是修改 bug，我们只升级后面两位版本号。 例如：07030004 表示 7.3.0.4 | |

[返回目录](#)

5.1.2 获取错误号 **PlayM4_GetLastError**

| | | |
|------|---------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetLastError(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 错误码，详见 错误码宏定义 | |
| 说 明: | 获得当前错误的错误码。用户在调用某个函数返回失败时，调用此函数获得错误的详细信息。 | |
| 注 意: | | |

[返回目录](#)

5.1.3 获取未使用的通道号 **PlayM4_GetPort**

| | | |
|------|---------------------------------|----------------------------|
| 函 数: | int PlayM4_GetPort(int * nPort) | |
| 参 数: | [out] int * nPort | 播放通道号，指向用于获取端口号的 int 型变量指针 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 获取未使用的通道号,通道号范围[0,31]。 | |
| 注 意: | | |

[返回目录](#)

5.1.4 释放已使用的通道号 **PlayM4_FreePort**

| | | |
|------|--------------------------------|-------|
| 函 数: | int PlayM4_FreePort(int nPort) | |
| 参 数: | int nPort | 播放通道号 |

| | |
|------|---------------------------------|
| 返回值: | 成功返回 1；失败返回 0 |
| 说 明: | 释放已使用的通道号，释放成功后，最好将 nPort 置为-1。 |
| 注 意: | |

[返回目录](#)

5.2 文件操作

5.2.1 打开文件 **PlayM4_OpenFile**

| | | |
|------|------------------------------------------------|--------------|
| 函 数: | int PlayM4_OpenFile(int nPort,char* sFileName) | |
| 参 数: | int nPort char* sFileName | 播放通道号 文件名 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | | |
| 注 意: | 文件不能超过 4G 或小于 4K | |

[返回目录](#)

5.2.2 关闭文件 **PlayM4_CloseFile**

| | | |
|------|---------------------------------|-------|
| 函 数: | int PlayM4_CloseFile(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | | |
| 注 意: | | |

[返回目录](#)

5.3 流操作

5.3.1 设置流播放模式 **PlayM4_SetStreamOpenMode**

| | | |
|------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | int PlayM4_SetStreamOpenMode(int nPort,unsigned int nMode) | |
| 参 数: | int nPort unsigned int nMode <i>STREAME_REALTIME</i> <i>STREAME_FILE</i> | 播放通道号 流播放模式: STREAME_REALTIME 、 STREAME_FILE 实时流模式, 会尽量保正实时性, 防止数据阻塞; 而且数据检查严格; (适合播放网络实时数据, 解码器会立刻解码)默认以此模式播放 文件流模式, 适合于文件回放方式, 偏重于流畅性而不注重实时性; (按时间戳播放) |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | 必须在播放之前设置。 当前版本只支持按照时间戳播放, 帧率播放不支持, 设置为 STREAME_FILE 后依然是按照时间戳播放 | |

[返回目录](#)

5.3.2 获取流播放模式 **PlayM4_GetStreamOpenMode***

| | | |
|------|-----------------------------------------------|-------|
| 函 数: | int PlayM4_GetStreamOpenMode(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | <i>STREAME_REALTIME</i> 或 <i>STREAME_FILE</i> | |
| 说 明: | | |
| 注 意: | 当前版本暂不支持此接口 | |

[返回目录](#)

5.3.3 打开流 **PlayM4_OpenStream**

| | | |
|------|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | int PlayM4_OpenStream(int nPort,unsigned char* pFileHeadBuf,unsigned int nSize, unsigned int nBufPoolSize) | |
| 参 数: | int nPort unsigned char* pFileHeadBuf unsigned int nSize unsigned int nBufPoolSize | 播放通道号 文件头数据 (用户从网络库实时流回调中得到的文件头数据) 文件头长度 设置播放器中存放数据流的缓冲区大小。 范围是 SOURCE_BUF_MIN ~ SOURCE_BUF_MAX 。该值过小会导致无法解码, 建议标清设备该值大于 |

| | | |
|------|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| | 其中: <code>#define SOURCE_BUF_MAX</code> <code>#define SOURCE_BUF_MIN</code> | 等于 200*1024, 高清设备大于等于 600*1024。 <code>1024* 100000</code> <code>1024*50</code> |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 需要送入正确的海康文件头 网络状况差, 数据不均匀或者有丢弃会引起卡顿情况; 如果在有效端口中文件或 者流已处于打开状态, 则会先关闭原有的文件或者流, 再进行后续操作 | |
| 注 意: | | |

[返回目录](#)

5.3.4 关闭流 **PlayM4_CloseStream**

| | | |
|------|------------------------------------------------|-------|
| 函 数: | <code>int PlayM4_CloseStream(int nPort)</code> | |
| 参 数: | <code>int nPort</code> | 播放通道号 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | | |

[返回目录](#)

5.3.5 输入流数据 **PlayM4_InputData**

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| 函 数: | <code>int PlayM4_InputData(int nPort,unsigned char* pBuf,unsigned int nSize)</code> | |
| 参 数: | <code>int nPort</code> <code>unsigned char* pBuf</code> <code>unsigned int nSize</code> | 播放通道号 流数据缓冲区地址 流数据缓冲区大小 |
| 返回值: | 1 表示已经输入数据到播放库缓冲。0 表示失败, 数据没有输入成功 | |
| 说 明: | 输入流数据, 需要在开启流 <code>PlayM4_OpenStream</code> 之后才能输入数据, 返回 0, 若错误码是 11 则是由于内部缓冲区满, 建议暂停输入数据线程, 再次输入数据, 确保播放库不丢失数据。 | |
| 注 意: | 实时流模式下, 先保证实时性, 尝试几次失败后再丢弃数据。 | |

[返回目录](#)

5.4 播放控制

5.4.1 开启播放 **PlayM4_Play**

| | | |
|------|-----------------------------------------------------------------------------|-----------|
| 函 数: | int PlayM4_Play(int nPort, PLAYM4_HWND hWnd) | |
| 参 数: | int nPort | 播放通道号 |
| | PLAYM4_HWND hWnd | 播放视频的窗口句柄 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 播放开始，播放视频画面大小将根据 hWnd 窗口调整，要全屏显示，只要把 hWnd 窗口放大到全屏。 如果已经播放，重置当前播放速度为正常速度。 | |
| 注 意: | 若只需要解码，不需要显示，可以将显示的窗口句柄置为 NULL。 | |

[返回目录](#)

5.4.2 关闭播放 **PlayM4_Stop**

| | | |
|------|----------------------------|-------|
| 函 数: | int PlayM4_Stop(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | | |
| 注 意: | | |

[返回目录](#)

5.4.3 暂停/恢复播放 **PlayM4_Pause**

| | | |
|------|-------------------------------------------------|-----------|
| 函 数: | int PlayM4_Pause(int nPort,unsigned int nPause) | |
| 参 数: | int nPort | 播放通道号 |
| | unsigned int nPause | 1 暂停，0 恢复 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | | |
| 注 意: | | |

[返回目录](#)

5.4.4 快速播放 **PlayM4_Fast**

| | | |
|------|-----------------------------|-------|
| 函 数: | int PlayM4_Fast(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 每次调用将使当前播放速度加快一倍，最多加速至 16 倍 | |

| | |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 注 意： | <p>实际能达到倍数取决与机器性能、视频的分辨率等多种因素。</p> <p>最大播放速度跟码流和手机 CPU 性能有很大关系，在 CPU 处于 90% 以上，或者一个核满的情况下，播放速度可能上不去，这属于硬件解码性能不足导致）。</p> <p>每次调用将使当前播放速度加快一倍，正常速度调用此接口最多调用 4 次；要恢复正常播放调用 play()，从当前位置开始正常播放；</p> <p>高清码流在高倍速播放时，由于受到解码和显示的限制，可能达不到所设置的速度。</p> <p>不支持实时流模式。</p> |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

[返回目录](#)

5.4.5 慢速播放 PlayM4_Slow

| | | |
|-------------|---------------------------------|-------|
| 函 数： | int PlayM4_Slow(int nPort) | |
| 参 数： | int nPort | 播放通道号 |
| 返回值： | 成功返回 1；失败返回 0 | |
| 说 明： | 慢速播放，每次调用将使当前播放速度慢一倍；最多减速至 1/16 | |
| 注 意： | 不支持实时流模式。 | |

[返回目录](#)

5.4.6 单帧前进 PlayM4_OneByOne*

| | | |
|-------------|--------------------------------|-------|
| 函 数： | int PlayM4_OneByOne(int nPort) | |
| 参 数： | int nPort | 播放通道号 |
| 返回值： | 成功返回 1；失败返回 0 | |
| 说 明： | 恢复正常播放需要调用 PlayM4_Play()。 | |
| 注 意： | 当前版本暂不支持此接口。 | |

[返回目录](#)

5.4.7 设置音量 PlayM4_SetVolume*

| | | |
|-------------|----------------------------------------------------------------------|-----------------|
| 函 数： | int PlayM4_SetVolume(int nPort, unsigned short nVolume); | |
| 参 数： | int nPort | 播放通道号 |
| | unsigned short nVolume | 音量值，范围 0-0xFFFF |
| 返回值： | 成功返回 1；失败返回 0 | |
| 说 明： | 可以在播放之前设置，返回值是 1，但设置的值被保存，并作为启动声音时的初始音量。这里的音量是指声卡输出的主音量，会影响到其他的声音应用。 | |
| 注 意： | 当前版本暂不支持此接口。 | |

[返回目录](#)

5.4.8 以独占方式打开声音 **PlayM4_PlaySound**

| | | |
|------|-------------------------------------------------|-------|
| 函 数: | int PlayM4_PlaySound(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 打开声音；同一时刻只能有一路声音。 如果现在已经有声音打开，则自动关闭原来已经打开的声音 | |
| 注 意: | 默认情况下是不处理声音数据的，不调用这个接口则不会解码音频数据。 | |

[返回目录](#)

5.4.9 关闭声音（独占方式） **PlayM4_StopSound**

| | | |
|------|-----------------------------------------------------------|--|
| 函 数: | int PlayM4_StopSound() | |
| 参 数: | 无 | |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 关闭最后一次使用 PlayM4_PlaySound 接口打开声音的播放通道 | |
| 注 意: | PlayM4_PlaySound 、 PlayM4_StopSound 需要配对调用。 | |

[返回目录](#)

5.4.10 音视频同步 **PlayM4_SyncToAudio**

| | | |
|------|-----------------------------------------------------|--------------------|
| 函 数: | int PlayM4_SyncToAudio(int nPort, int bSyncToAudio) | |
| 参 数: | int nPort int bSyncToAudio | 播放通道号 是否开启音视频同步 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 音视频同步需要以音视频的时间戳为基准，若时间戳有跳变，则可能会影响音视频同步的效果。 | |
| 注 意: | | |

[返回目录](#)

5.4.11 设置图像垂直翻转 **PlayM4_SetVerticalFlip**

| | | |
|------|------------------------------------------------------------|------------------------------|
| 函 数: | int PlayM4_SetVerticalFlip(unsigned int nPort , int bFlag) | |
| 参 数: | unsigned int nPort int bFlag | 播放通道号 1：图像垂直翻转；0：图像不垂直翻转。 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | | |
| 注 意: | | |

[返回目录](#)

5.4.12 设置文件当前播放时间（毫秒） **PlayM4_SetPlayedTimeEx***

| | | |
|------|----------------------------------------------------------------------------------------|------------------|
| 函 数: | int PlayM4_SetPlayedTimeEx(int nPort,unsigned int nTime) | |
| 参 数: | int nPort | 播放通道号 |
| | unsigned int nTime | 设置文件播放到指定时间，单位毫秒 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 根据时间设置文件播放位置 | |
| 注 意: | 如果在建立文件索引的前提下使用, 为精确定位, 否则即为粗略定位, 精确定位是定位到最近的一个视频帧, 粗略定位是位置之后的最近的一个关键帧) 当前版本暂不支持此接口 | |

[返回目录](#)

5.4.13 设置文件当前播放位置 **PlayM4_SetPlayPos***

| | | |
|------|------------------------------------------------------------------------------------------------------------------------|--------|
| 函 数: | int PlayM4_SetPlayPos (int nPort,float fRelativePos) | |
| 参 数: | int nPort | 播放通道号 |
| | float fRelativePos | 文件位置比例 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 设置文件播放相对位置，百分比 | |
| 注 意: | 如果在建立文件索引的前提下使用则为精确定位；否则即为粗略定位, 精确定位是定位到最近的一个视频帧, 粗略定位是位置之后的最近的一个关键帧) 当前版本暂不支持此接口 | |
| 示 例 | int bFlag = PlayM4_SetPlayPos (nPort,0.25); 注: 参数 1: nPort 表示已经获取的播放库内部分配的端口号 ; 参数 2: 0.25 表示当前设定的定位位置为 25% ; | |

[返回目录](#)

5.4.14 获取文件当前播放位置 **PlayM4_GetPlayPos***

| | | |
|------|--------------------------------------------------------------------------------------|-------|
| 函 数: | float PlayM4_GetPlayPos (int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 当前播放的进度值（百分比） | |
| 说 明: | 获取文件当前播放相对位置，百分比 | |
| 注 意: | 如果在建立文件索引的前提下使用则为精确定位；否则即为粗略定位, 精确定位是定位到最近的一个视频帧, 粗略定位是位置之后的最近的一个关键帧) 当前版本暂不支持此接口 | |
| 示 例 | | |

[返回目录](#)

5.4.15 设置文件当前播放帧号 **PlayM4_SetCurrentFrameNum***

| | | |
|------|-------------------------------------------------------------------------------------------------------------------|-------------|
| 函 数: | int PlayM4_SetCurrentFrameNum(int nPort,unsigned int nFrameNum) | |
| 参 数: | int nPort unsigned int nFrameNum | 播放通道号 帧号 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 设置当前播放位置到指定帧号；根据帧号来定位播放位置。 | |
| 注 意: | 如果在建立文件索引的前提下使用则为精确定位；否则即为粗略定位， 若客户指定的帧号存在，则精确定位是定位到指定的视频帧；若不存在则是最 近的视频帧；而粗略定位是位置之后的最近的一个关键帧 当前版本暂不支持此接口 | |

[返回目录](#)

5.4.16 获取文件当前播放帧号 **PlayM4_GetCurrentFrameNum***

| | | |
|------|-----------------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetCurrentFrameNum(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 当前播放的帧序号 | |
| 说 明: | 得到当前播放的帧序号。 | |
| 注 意: | 当前版本暂不支持此接口 | |

[返回目录](#)

5.4.17 获取当前播放帧的全局时间 **PlayM4_GetSystemTime**

| | | |
|------|----------------------------------------------------------------------------------------|---------------|
| 函 数: | int PlayM4_GetSystemTime(int nPort, PLAYM4_SYSTEM_TIME *pstSystemTime) | |
| 参 数: | int nPort [out]PLAYM4_SYSTEM_TIME * pstSystemTime | 播放通道号 全局时间 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 获取当前播放帧对应的全局时间。 获取到的系统时间是在码流封装层里面的，可能与图像 OSD 上面有一定误差。 | |

[返回目录](#)

5.5 获取播放或解码信息

5.5.1 获取文件总时间 **PlayM4_GetFileTime***

| | | |
|------|-------------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetFileTime(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 文件总时间长度，单位秒 | |
| 说 明: | | |
| 注 意: | 不支持对文件的数据追加（不支持对正在写入的文件进行时间的读取）。 当前版本暂不支持此接口 | |

[返回目录](#)

5.5.2 获取文件总帧数 **PlayM4_GetFileTotalFrames***

| | | |
|------|---------------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetFileTotalFrames(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 文件总帧数 | |
| 说 明: | | |
| 注 意: | 不支持对文件的数据追加（不支持对正在写入的文件进行帧数的读取）。 当前版本暂不支持此接口 | |

[返回目录](#)

5.5.3 获取当前帧率 **PlayM4_GetCurrentFrameRateEx***

| | | |
|------|-----------------------------------------------------------------|-------------|
| 函 数: | int PlayM4_GetCurrentFrameRateEx(int nPort, float* pfFrameRate) | |
| 参 数: | int nPort | 播放通道号 |
| | [out]float* pfFrameRate | 当前码流中当前编码帧率 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | | |
| 注 意: | 主要是为增加对帧率小于 1 的情况下获取帧率而增加的 当前版本暂不支持此接口 | |

[返回目录](#)

5.5.4 获取已播放时间 **PlayM4_GetPlayedTime***

| | | |
|------|----------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetPlayedTime(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 文件当前播放时间，单位秒 | |
| 说 明: | | |

| | |
|------|-------------|
| 注 意: | 当前版本暂不支持此接口 |
|------|-------------|

[返回目录](#)

5.5.5 获取已解码的视频帧数 **PlayM4_GetPlayedFrames***

| | | |
|------|------------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetPlayedFrames(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 已经解码的视频帧数 | |
| 说 明: | | |
| 注 意: | 当前版本暂不支持此接口 | |

[返回目录](#)

5.5.6 获取原始图像大小 **PlayM4_GetPictureSize**

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| 函 数: | int PlayM4_GetPictureSize(int nPort,int *pWidth,int *pHeight) | |
| 参 数: | int nPort, [out] int * pWidth, [out] int * pHeight | 播放通道号 原始图像的宽度 原始图像的高度 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 获得码流中原始图像的大小, 根据此大小来设置显示窗口的区域, 可以不使用显卡做缩放工作, 对于那些不支持硬件缩放的显卡来说非常有用。该函数获取的是当前播放显示的视频帧数据的宽高信息, 因此正常预览显示后才能获取准确的值。在尚未解码出第一帧图像前, 返回默认值 352x288。 获取 2cif 码流或者文件流的图片大小的时候返回的是 4cif 的分辨率大小 | |
| 注 意: | | |

[返回目录](#)

5.5.7 获取当前显示帧的全局时间 **PlayM4_GetSpecialData***

| | | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetSpecialData(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 失败返回 0, 成功则返回一个全局时间的压缩值, 精确到秒 #define GET_YEAR(_time_) (((_time_)>>26) + 2000) #define GET_MONTH(_time_) (((_time_)>>22) & 15) #define GET_DAY(_time_) (((_time_)>>17) & 31) #define GET_HOUR(_time_) (((_time_)>>12) & 31) #define GET_MINUTE(_time_) (((_time_)>>6) & 63) #define GET_SECOND(_time_) (((_time_)>>0) & 63) | |
| 说 明: | 有正常图像显示后才能调用此接口 | |
| 注 意: | 当前版本暂不支持此接口 | |

[返回目录](#)

5.6 解码操作及控制

5.6.1 设置解码回调流类型 `PlayM4_SetDecCStream`

| | | |
|------|----------------------------------------------------------|---------------------|
| 函 数: | int PlayM4_SetDecCStream(int nPort,unsigned int nStream) | |
| 参 数: | int nPort | 播放通道号 |
| | unsigned int nStream | 1 视频流, 2 音频流, 3 复合流 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 建议在解码回调函数前使用 | |
| 注 意: | | |

[返回目录](#)

5.6.2 设置视频帧解码类型 `PlayM4_SetDecodeFrameType`

| | | |
|------|------------------------------------------------------------------|----------------------------------------------------------------------------------|
| 函 数: | int PlayM4_SetDecodeFrameType(int nPort,unsigned int nFrameType) | |
| 参 数: | int nPort | 播放通道号 |
| | unsigned int nFrameType | 解码帧类型说明, 设置为: 0 正常解码 1 只解关键帧 2 不解视频帧 4 解码类型为4时表示8倍速高分辨率码流不自动抽帧, 默认全解 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |

[返回目录](#)

5.6.3 解码回调 `PlayM4_SetDecCallBack`

| | | |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| 函 数: | int PlayM4_SetDecCallBack(int nPort,void (CALLBACK* DecCBFun)(int nPort,char * pBuf,int nSize,FRAME_INFO * pFrameInfo,int nReserved1,int nReserved2)) | |
| 参 数: | int nPort | 播放通道号 |
| | void (CALLBACK* DecCBFun) | 解码回调函数指针, 不能为 NULL |
| | DecCBFun 回调函数参数说明 | |
| | int nPort | 播放器通道号 |
| | char * pBuf | 解码后的音视频数据 |
| | int nSize | 解码后的音视频数据 pBuf 的长度 |
| | FRAME_INFO *pFrameInfo | 图像和声音信息结构体指针 |
| | int nReserved1 | 保留参数 |
| | int nReserved2 | 保留参数 |

| | |
|------|------------------------------------------------------------------------------------------------------------------------|
| 返回值: | 成功返回 1；失败返回 0 |
| 说明: | 设置回调函数，替换播放器中的显示部分，由用户自己控制显示，该函数在 PlayM4_Play 之前调用，在 PlayM4_Stop 时自动失效，下次调用 PlayM4_Play 之前需要重新设置。 |
| 注意: | 解码部分不控制 播放速度 ，只要用户从回调函数中返回，解码器就会解码下一部分数据。这个功能的使用需要用户对视频显示和声音播放有足够的了解，否则请慎重使用。 |

[返回目录](#)

5.6.4 解码回调（增加用户传递参数） **PlayM4_SetDecCallBackMend**

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| 函数: | int PlayM4_SetDecCallBackMend(int nPort,void (CALLBACK* DecCBFun)(int nPort,char * pBuf, int nSize, FRAME_INFO * pFrameInfo, int nUser, int nReserved2), int nUser) | |
| 参数: | int nPort void (CALLBACK* DecCBFun) int nUser DecCBFun 回调函数参数 int nPort char * pBuf int nSize FRAME_INFO * pFrameInfo int nUser int nReserved2 | 播放通道号 解码回调函数，不能为 NULL 用户数据 播放通道号 解码后的视音频数据 解码后的视音频数据长度 图像与声音信息 用户数据 保留参数 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说明: | 设置回调函数，替换播放器中的显示部分，由用户自己控制显示，该函数在 PlayM4_Play 之前调用，在 PlayM4_Stop 时自动失效，下次调用 PlayM4_Play 之前需要重新设置。 注意 解码部分不控制 速度 ，只要用户从回调函数中返回，解码器就会解码下一部分数据。这个功能的使用需要用户对视频显示和声音播放有足够的了解，否则请慎重使用。和解码回调函数 PlayM4_SetDecCallBack 区别在于增加了用户传递参数。 | |
| 注意: | | |

[返回目录](#)

5.6.5 文件结束回调 **PlayM4_SetFileEndCallback**

| | | |
|-----|----------------------------------------------------------------------------------------------------------------|-------|
| 函数: | int PlayM4_SetFileEndCallback(int nPort, void(CALLBACK*FileEndCallback)(int nPort, void *pUser), void *pUser) | |
| 参数: | int nPort | 播放通道号 |

| | | |
|------|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| | void(CALLBACK* FileEndCallback) void * pUser FileEndCallback 回调函数参数 int nPort void * pUser | 文件结束回调函数指针 用户数据 播放通道号 用户数据 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说明: | 设置文件播放结束回调函数。在 PlayM4_OpenFile 之前调用才有效 | |
| 注意: | 只适用于文件模式。 | |

[返回目录](#)

5.6.6 解码密钥 **PlayM4_SetSecretKey**

| | | |
|------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| 函数: | int PlayM4_SetSecretKey(int nPort, int lKeyType, char *pSecretKey, int lKeyLen) | |
| 参数: | int nPort int lKeyType char * pSecretKey int lKeyLen | 播放通道号 密钥类型（取值为 0 不加密；取值为 1 标识 AES 加密） 密钥串首地址 密钥长度，单位为 bit（1 字节为 8 bit） |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说明: | 如果在编码时设置了密钥，那么在解码之前需要调用该接口设置密钥才能正常解码。该接口在 PlayM4_OpenStream/PlayM4_OpenFile 之前调用才有效。 | |
| 注意: | 编码只支持标准 264 格式的水印。 | |

[返回目录](#)

5.6.7 加密码流回调 **PlayM4_SetEncryptTypeCallBack***

| | | |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| 函数: | int PlayM4_SetEncryptTypeCallBack(int nPort,unsigned int nType,void (CALLBACK* EncryptTypeCBFun)(int nPort,ENCRYPT_INFO* pEncryptInfo,void* nUser,int nReserved2), void* nUser); | |
| 参数: | int nPort unsigned int nType void (CALLBACK* EncryptTypeCBFun) void* nUser EncryptTypeCBFun 回调函数参数 int nPort ENCRYPT_INFO * pEncryptInfo void* nUser int nReserved2 | 播放通道号 nType=0，表示码流加密标记位发生变化就回调； nType=1，表示码流有加密位发生回调 加密码流回调函数 用户指针 播放通道号 加密数据信息 用户数据 保留数据 |
| 返回值: | 成功返回 1；失败返回 0 | |

| | |
|-------------|------------------------------------------------------------------------------------------|
| 说 明： | 仅仅支持加密码流，通过回调函数将加密的码流数据回调给用户。若无加密信息的码流调用此接口无效果。 回调函数中不能调用参数中带 nPort 的播放库接口，否则有可能造成死锁。 |
| 注 意： | 当前版本暂不支持此接口 |

[返回目录](#)

5.7 显示操作

5.7.1 设置或增加显示区域 **PlayM4_SetDisplayRegion**

| | | | | | | | | | | | | | | | | | | | | |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-------|-------|--------------|------------|-------------------------------------------------------------------------------------------------------------------|--------------------------|----------|------------------------------------------------------------------------------------------------------|-------------|----------|---------------------------------------|-----|---------|------------------|-----------------------------------------|--|---|--|
| 函 数: | int PlayM4_SetDisplayRegion(int nPort,unsigned int nRegionNum, HKRECT *pSrcRect, PLAYM4_HWND hDestWnd, int bEnable) | | | | | | | | | | | | | | | | | | | |
| 参 数: | <table><tr><td>int</td><td>nPort</td><td>播放通道号</td></tr><tr><td>unsigned int</td><td>nRegionNum</td><td>显示区域序号, 0~(MAX_DISPLAY_WND-1)。如果 nRegionNum 为 0, 表示对主要显示窗口 (PlayM4_Play 中设置的窗口) 进行设置, 将忽略 hDestWnd 和 bEnable 的设置。</td></tr><tr><td>HKRECT*</td><td>pSrcRect</td><td>设置在要显示的原始图像上的区域, 如: 如果原始图像是 352*288, 那么 pSrcRect 可设置的范围只能在(0,0,352,288)之中。如果 pSrcRect=NULL, 将显示整个图像。</td></tr><tr><td>PLAYM4_HWND</td><td>hDestWnd</td><td>设置显示窗口。如果该区域的窗口已经设置过 (打开过), 那么该参数被忽略。</td></tr><tr><td>int</td><td>bEnable</td><td>打开 (设置) 或关闭显示区域。</td></tr><tr><td colspan="2">#define MAX_DISPLAY_WND</td><td>4</td></tr></table> | int | nPort | 播放通道号 | unsigned int | nRegionNum | 显示区域序号, 0~(MAX_DISPLAY_WND-1)。如果 nRegionNum 为 0, 表示对主要显示窗口 (PlayM4_Play 中设置的窗口) 进行设置, 将忽略 hDestWnd 和 bEnable 的设置。 | HKRECT * | pSrcRect | 设置在要显示的原始图像上的区域, 如: 如果原始图像是 352*288, 那么 pSrcRect 可设置的范围只能在(0,0,352,288)之中。如果 pSrcRect=NULL, 将显示整个图像。 | PLAYM4_HWND | hDestWnd | 设置显示窗口。如果该区域的窗口已经设置过 (打开过), 那么该参数被忽略。 | int | bEnable | 打开 (设置) 或关闭显示区域。 | #define MAX_DISPLAY_WND | | 4 | |
| int | nPort | 播放通道号 | | | | | | | | | | | | | | | | | | |
| unsigned int | nRegionNum | 显示区域序号, 0~(MAX_DISPLAY_WND-1)。如果 nRegionNum 为 0, 表示对主要显示窗口 (PlayM4_Play 中设置的窗口) 进行设置, 将忽略 hDestWnd 和 bEnable 的设置。 | | | | | | | | | | | | | | | | | | |
| HKRECT * | pSrcRect | 设置在要显示的原始图像上的区域, 如: 如果原始图像是 352*288, 那么 pSrcRect 可设置的范围只能在(0,0,352,288)之中。如果 pSrcRect=NULL, 将显示整个图像。 | | | | | | | | | | | | | | | | | | |
| PLAYM4_HWND | hDestWnd | 设置显示窗口。如果该区域的窗口已经设置过 (打开过), 那么该参数被忽略。 | | | | | | | | | | | | | | | | | | |
| int | bEnable | 打开 (设置) 或关闭显示区域。 | | | | | | | | | | | | | | | | | | |
| #define MAX_DISPLAY_WND | | 4 | | | | | | | | | | | | | | | | | | |
| 返回值: | 成功返回 1; 失败返回 0 | | | | | | | | | | | | | | | | | | | |
| 说 明: | 设置或增加显示区域。可以做局部放大显示。 | | | | | | | | | | | | | | | | | | | |
| 注 意: | 对于设置子窗口, RECT 的宽或者高小于 16 个像素点, 不能做放大处理, 只能显示原始图像的大小。 | | | | | | | | | | | | | | | | | | | |

[返回目录](#)

5.7.2 设置显示窗口 **PlayM4_SetVideoWindow**

| | | | | | | | | | | | |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------|--------|--------------|------------|--------|-------------|------|---------|--|
| 函 数: | int PlayM4_SetVideoWindow(int nPort, unsigned int nRegionNum, PLAYM4_HWND hWnd) | | | | | | | | | | |
| 参 数: | <table><tr><td>int</td><td>nPort</td><td>播放库端口号</td></tr><tr><td>unsigned int</td><td>nRegionNum</td><td>显示区域序号</td></tr><tr><td>PLAYM4_HWND</td><td>hWnd</td><td>显示的窗口句柄</td></tr></table> | int | nPort | 播放库端口号 | unsigned int | nRegionNum | 显示区域序号 | PLAYM4_HWND | hWnd | 显示的窗口句柄 | |
| int | nPort | 播放库端口号 | | | | | | | | | |
| unsigned int | nRegionNum | 显示区域序号 | | | | | | | | | |
| PLAYM4_HWND | hWnd | 显示的窗口句柄 | | | | | | | | | |
| 返回值: | 成功返回 1, 失败返回 0 | | | | | | | | | | |
| 说 明: | 电子放大时调用该接口电子放大效果会消失。 | | | | | | | | | | |
| 注 意: | | | | | | | | | | | |

[返回目录](#)

5.7.3 设置同步回放组 **PlayM4_SetSycGroup**

| | | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| 函 数: | int PlayM4_SetSycGroup(int nPort, int dwGroupIndex) | |
| 参 数: | int nPort | 播放库端口 |
| | int dwGroupIndex | 同步回放分组号，取值在 0~3 之间 |
| 返回值: | 成功返回 1，失败返回 0 | |
| 说 明: | <p>可以多次调用将指定的 nPort 加入到同步组中；</p> <p>PlayM4_SetSycGroup 在 PlayM4_Play 之前调用；</p> <p>dwGroupIndex 一个序号被视为一个同步组，每个同步组最多 16 个播放通道，若超过 16 给播放通道则返回失败，且必须 PlayM4_FreePort 此播放通道才能退出同步组；</p> <p>目前只支持海康设备码流。</p> <p>dwGroupIndex 暂约定取值 0~3，第一版本取消同步只能同个 PlayM4_CloseStream 处理</p> | |
| 注 意: | | |

[返回目录](#)

5.8 缓冲区操作

解码前源缓冲区

5.8.1 获取源缓冲区剩余数据大小 **PlayM4_GetSourceBufferRemain**

| | | |
|------|------------------------------------------------------|--------|
| 函 数: | unsigned int PlayM4_GetSourceBufferRemain(int nPort) | |
| 参 数: | int nPort | 播放库端口号 |
| 返回值: | 返回源缓冲区剩余数据大小（单位：BYTE） | |
| 说 明: | 成功返回 1；失败返回 0 | |
| 注 意: | 获取源缓冲区剩余数据大小，如果返回-1，则说明所有数据已播放完成。 | |

[返回目录](#)

解码后播放缓冲区

5.8.2 设置播放缓冲区最大缓冲帧数 **PlayM4_SetDisplayBuf**

| | | |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | int PlayM4_SetDisplayBuf(int nPort,unsigned int nNum) | |
| 参 数: | int nPort unsigned int nNum | 播放通道号 播放缓冲区最大缓冲帧数。 范围：MIN_DIS_FRAMES~MAX_DIS_FRAMES。一帧 352*288 图像的所需内存最小值是 352*288*3/2 大约 150K 。最大值是 352*288*4 大约 405K。 <code>#define MIN_DIS_FRAMES</code> 1 <code>#define MAX_DIS_FRAMES</code> 50 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 设置播放缓冲区（即解码后的图像缓冲区）大小； 这个缓冲区比较重要，可以直接影响播放的流畅性和延时性。在一定范围内缓冲越大越流畅，同时延时越大。 在播放文件时用户最好可以考虑开大缓冲（如果内存足够大），我们的默认值是 15（帧），在 25 帧/秒的情况下即 0.6 秒的数据。 在播放流时我们的默认值是 15(帧) ，如果用户追求较小的延时，可以考虑适当减小这个值。 | |
| 注 意: | 建议在 PlayM4_OpenStream 之前配置，如果要改变这个配置，必须 PlayM4_CloseStream。 重置回调标志位为有效状态。 若设置的值超过有效范围，播放库会内部自适应为 15，且会返回成功。 | |

| | |
|--|---------------|
| | 在 play 之间调用有效 |
|--|---------------|

[返回目录](#)

5.8.3 获取播放缓冲区最大缓冲帧数 **PlayM4_GetDisplayBuf**

| | | |
|------|------------------------------------------------------|-------|
| 函 数: | unsigned int PlayM4_GetDisplayBuf(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 播放缓冲区最大缓冲帧数 | |
| 说 明: | | |
| 注 意: | 未创建缓冲 buf 的时候是返回 0 的,只有在创建了缓冲(就是有显示画面的之后)才会返回缓冲的节点个数 | |

[返回目录](#)

源、播放缓冲区

5.8.4 清空所有缓冲区的剩余数据 **PlayM4_ResetSourceBuffer**

| | | |
|------|-----------------------------------------|-------|
| 函 数: | int PlayM4_ResetSourceBuffer(int nPort) | |
| 参 数: | int nPort | 播放通道号 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 清空数据缓冲 (同时清除帧解析缓存) 的剩余数据 | |
| 注 意: | | |

[返回目录](#)

5.8.5 清空指定缓冲区的剩余数据 **PlayM4_ResetBuffer**

| | | |
|------|---------------------------------------------------------|---------------------------------------------------------|
| 函 数: | int PlayM4_ResetBuffer(int nPort,unsigned int nBufType) | |
| 参 数: | int nPort | 播放通道号 |
| | unsigned int nBufType | 缓冲区类型 |
| | 缓冲区宏定义 | |
| | BUF_VIDEO_SRC | 视频数据源缓冲区 (同时清除帧解析缓存), 只对流模式有效, 单位 byte。 |
| | BUF_AUDIO_SRC | 音频数据源缓冲区 (同时清除帧解析缓存), 只对流模式有效, 单位 byte。且音视频数据分开送到情况下才有效 |
| | BUF_VIDEO_RENDER | 视频播放缓冲区剩余数据, 单位帧数。 |
| | BUF_AUDIO_RENDER | 音频播放缓冲区剩余数据, 单位帧数。 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | | |

[返回目录](#)

5.8.6 获取指定缓冲区的大小 **PlayM4_GetBufferValue**

| | | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | unsigned int PlayM4_GetBufferValue(int nPort,unsigned int nBufType) | |
| 参 数: | <div><div>int nPort</div><div>unsigned int nBufType</div><div>缓冲区宏定义</div><div>BUF_VIDEO_SRC</div><div>BUF_AUDIO_SRC</div><div>BUF_VIDEO_RENDER</div><div>BUF_AUDIO_RENDER</div></div> | <div>播放通道号</div> <div>缓冲区类型</div> <div>视频数据源缓冲区大小，只对流模式有效，单位 byte。</div> <div>音频数据源缓冲区大小，只对流模式有效，单位 byte。且音视频数据分开送到情况下才有效</div> <div>视频播放缓冲区剩余数据，单位帧数。</div> <div>音频播放缓冲区剩余数据，单位帧数，音频 40ms 数据定为一帧。</div> |
| 返回值: | 根据参数不同，返回缓冲区值，源缓冲区返回 byte,解码后缓冲区返回帧数。 | |
| 说 明: | 获取播放器中的缓冲区大小（帧数或者 byte）。BUF_VIDEO_RENDER 可在网络延时方面有所估计。 | |
| 注 意: | ios 的音频缓冲节点是 60 个，而视频缓冲节点是 15。 | |

[返回目录](#)

5.9 索引

5.9.1 设置建立索引回调 **PlayM4_SetFileRefCallBack**

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| 函 数: | int PlayM4_SetFileRefCallBack(int nPort, void (__stdcall *pFileRefDone) (unsigned int nPort, unsigned int nUser), unsigned int nUser) | |
| 参 数: | int nPort void (__stdcall * pFileRefDone) unsigned int nUser pFileRefDone 回调函数参数说明 unsigned int nPort unsigned int nUser | 播放通道号 索引回调函数 用户数据 播放器通道号 用户数据 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 文件索引建立后回调。 为了能在文件播放中精确定位，在文件打开的时候建议生成文件索引。（目前测试的情况是 100M-200M 的录像文件，建议索引时间约 1s），建立索引的过程是在后台完成，需要使用索引的函数要等待这个过程结束，而其他接口不会受到影响。 | |
| 注 意: | | |

[返回目录](#)

5.10 抓图

5.10.1抓图回调 **PlayM4_SetDisplayCallBack**

| | | |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | int PlayM4_SetDisplayCallBack(int nPort,void (CALLBACK* DisplayCBFun) (int nPort,char * pBuf,int nSize, int nWidth, int nHeight, int nStamp, int nType, int nReceaved)) | |
| 参 数: | int nPort void (CALLBACK* DisplayCBFun) DisplayCBFun 抓图回调函数参数说明 int nPort char * pBuf int nSize int nWidth int nHeigh int nStamp int nType int nReceaved | 播放通道号 抓图回调函数，可以为 NULL 播放通道号 返回图像数据指针 返回图像数据大小 画面宽，单位像素 画面高 时标信息，单位毫秒 数据类型， T_YV12, T_UYVY， 详见 PlayM4_SetDecCallBack 宏定义 保留 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 设置抓图回调函数；注意要尽快返回，如果要停止回调，可以把回调函数指针 DisplayCBFun 设为 NULL。一旦设置回调函数，则一直有效，直到 PlayM4_Stop 。该函数可以在任何时候调用 回调在时钟线程中触发,不能出现耗时操作,否则会打乱时钟脉冲,影响显示 | |
| 注 意: | | |

[返回目录](#)

5.10.2抓图回调 **PlayM4_SetDisplayCallBackEx**

| | | |
|------|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| 函 数: | int PlayM4_SetDisplayCallBackEx(int nPort,void (CALLBACK*DisplayCBFun) (DISPLAY_INFO *pstDisplayInfo),void* nUser); | |
| 参 数: | int nPort void (CALLBACK* DisplayCBFun) void* nUser DisplayCBFun 抓图回调函数参数说明 DISPLAY_INFO * pstDisplayInfo | 播放通道号 抓图回调函数，可以为 NULL 用户指针 播放通道号 当前显示信息结构体指针 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | ➤ 设置抓图回调函数；注意要尽快返回，如果要停止回调，可以把回调函数指针 DisplayCBFun 设为 NULL。一旦设置回调函数，则一直有效，直到 | |

| | |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>PlayM4_Stop。该函数可以在任何时候调用</p> <ul style="list-style-type: none"> ➤ 回调在时钟线程中触发,不能出现耗时操作,否则会打乱时钟脉冲,影响显示 ➤ 当前获取到的图像数据类型为 YV12 格式。 ➤ 回调函数中不能调用参数中带 nPort 的播放库接口, 否则有可能造成死锁。 |
| 注 意: | |

[返回目录](#)

5.10.3 直接抓取 BMP 图像 **PlayM4_GetBMP**

| | | |
|------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | int PlayM4_GetBMP(int nPort,unsigned char* pBitmap,unsigned int nBufSize, unsigned int* pBmpSize) | |
| 参 数: | <div>int nPort</div> <div>unsigned char* pBitmap</div> <div>unsigned int nBufSize</div> <div>[out]unsigned int * pBmpSize</div> | <div>播放通道号</div> <div>存放 BMP 图像数据地址, 由用户分配, 不得小于 bmp 图像大小, 即 sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER)+ w * h * 4, 其中 w 和 h 分别为图像宽高。</div> <div>申请的缓冲区大小</div> <div>获取到的实际 bmp 图像大小</div> |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | 2CIF 分辨率的图像抓图后的分辨率为 4CIF。 | |

[返回目录](#)

5.10.4 直接抓取 JPEG 图像 **PlayM4_GetJPEG**

| | | |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | int PlayM4_GetJPEG(int nPort,unsigned char* pJpeg,unsigned int nBufSize, unsigned int* pJpegSize) | |
| 参 数: | <div>int nPort</div> <div>unsigned char* pJpeg</div> <div>unsigned int nBufSize</div> <div>[out]unsigned int * pJpegSize</div> | <div>播放通道号</div> <div>存放 JPEG 图像数据地址, 由用户分配, 因为无法预算出编码后的数据大小, 建议申请大小为 w * h * 3/2, 其中 w 和 h 分别为图像宽高。</div> <div>申请的缓冲区大小</div> <div>获取到的实际 JPEG 图像数据大小</div> |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | <p>2CIF 分辨率的图像抓图后的分辨率为 4CIF。</p> <p>获取到的 JPEG 图片的大小最终会扩成宽高都是 16 的整数倍, 比如播放 1920 * 1080 的录像, 抓出来的 JPEG 图片的分辨率就是扩成 1920 * 1088(1080 不是 16 的整数倍)。</p> | |

[返回目录](#)

5.11 硬解码

5.11.1 设置硬解码优先 **PlayM4_SetHDPriority**

| | | |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 函 数: | int PlayM4_SetHDPriority(int nPort) | |
| 参 数: | int nport | 播放通道号 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 设置硬解码优先后，符合硬解码的码流优先进行硬解码，不符合的会自动切换至软解，设置后无法关闭。接口在 PlayM4_OpenStream 之后、PlayM4_Play 之前调用。 | |
| 注 意: | iOS 版本 8.0 及以上，暂时支持分辨率为 720P、1080P、704*576、704*480、2048*1536、2560*1920、640*480、1024*768、1280*960，新增 512*288、768*432。硬解码支持 H264 编码和 Smart264 编码。 | |

[返回目录](#)

5.11.2 获取当前解码类型 **PlayM4_GetDecodeEngine**

| | | |
|------|---------------------------------------|-------|
| 函 数: | int PlayM4_GetDecodeEngine(int nPort) | |
| 参 数: | int nport | 播放通道号 |
| 返回值: | 1-表示软解；2 表示硬解 | |
| 说 明: | 需要在 PlayM4_Play 之后，视频正常播放过程中调用。 | |
| 注 意: | 在码流解码正常播放后调用，否则返回的解码类型无意义 | |

[返回目录](#)

5.12 预录像功能

5.12.1 设置预录像数据回调 `PlayM4_SetPreRecordCallBack`

| | | |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 函 数: | <code>int PlayM4_SetPreRecordCallBack(int nPort, void (CALLBACK* PreRecordCBfun)(int nPort, void* pData, unsigned int nDataLen, void *pUser), void *pUser)</code> | |
| 参 数: | <div><div><code>int</code> <code>void (CALLBACK* PreRecordCBfun)</code> <code>void</code> <code>PreRecordCBfun</code> 回调函数参数说明 <code>int</code> <code>void*</code> <code>unsigned int</code> <code>void*</code></div><div><code>nPort</code> <code>nPort</code> <code>pData</code> <code>nDataLen</code> <code>pUser</code></div></div> | <div>播放通道号 数据回调函数 用户指针 播放通道号 音视频数据 数据长度 用户指针</div> |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 该接口需要在 <code>PlayM4_Play</code> 之前调用。回调函数中不能调用参数中带 <code>nPort</code> 的播放库接口，否则有可能造成死锁。另外，回调出来的数据需要及时处理，不能阻塞回调函数，否则会造成录像丢帧。 | |
| 注 意: | 当前版本 <code>setPreRecordCallBack</code> 这个接口要在第一个 I 帧显示之后调用才行 所以建议播放一段时间再调用。播放的窗口句柄不能为 <code>NULL</code> ，必须是有效的显示窗口句柄 | |
| 示例代码 | <pre>//预录数据保存的本地文件 HANDLE g_hVidPreFile = CreateFile("Prereord.mp4",GENERIC_WRITE,FILE_SHARE_WRITE,NULL,CREATE_ALWAYS,FILE_ATTRIBUTES_NORMAL,NULL); //预录回调函数 void CALLBACK fPreRecordCBfun(long nPort, RECORD_DATA_INFO *pRecordDataInfo,void* pUser) { DWORD dwSize = 0; if (pRecordDataInfo == NULL) { return; } if (pRecordDataInfo->nType) { if (g_hVidPreFile != NULL) { //保存预录数据 WriteFile(g_hVidPreFile,pRecordDataInfo->pBuf,pRecordDataInfo->nBufLen,&dwSize,NULL); } } }</pre> | |

```

    }
}
}
BOOL g_bDecFlag = FALSE;
//解码回调函数
void CALLBACK DecCBFun(long nPort,char * pBuf,long nSize,
                        FRAME_INFO * pFrameInfo,
                        long nReserved1,long /*nReserved2*/)
{
    CPlayerDlg* pDlg = (CPlayerDlg *)nReserved1;
    DWORD dwSize = 0;
    if (pFrameInfo->nType == T_AUDIO16)
    {
        OutputDebugString("Zytest:: get audio data !\n");
        WriteFile(g_hAudFile,pBuf,nSize,&dwSize,NULL);
    }
    else if ( pFrameInfo->nType == T_YV12 )
    {
        //成功解码出第一帧数据，设置标记位置
        g_bDecFlag = TRUE;
    }
}

long lPort = -1;
BOOL g_bFlag = FALSE;
HWND g_hWnd = NULL;
//网络 SDK 的实时流回调函数（以海康设备网络 SDK 实时流回调为例）
void CALLBACK fRealDataCallBack_V30(LONG lRealHandle, DWORD dwDataType, BYTE *pBuffer,
DWORD dwBufSize, void* pUser)
{
    BOOL bFlag = TRUE;
    CPlayerDlg* pDlg = (CPlayerDlg*)pUser;

    switch (dwDataType)
    {
    case NET_DVR_SYSHEAD://当前类型为文件头
        //获取当前播放库端口
        bFlag = PlayM4_GetPort(&lPort);
        //设置实时流打开模式
        bFlag = PlayM4_SetStreamOpenMode(lPort,1);
        //打开实时流数据
        bFlag = PlayM4_OpenStream(lPort,pBuffer,dwBufSize,4*1024*1024);
        //设置预录标记

```

| | |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre>bFlag = PlayM4_SetPreRecordFlag(IPort,TRUE); //设置解码回调, 此处用来确认是否有成功解码出第一帧数据 bFlag = PlayM4_SetDecCallBackExMend(IPort,DecCBFun,NULL,0,NULL); //开始解码, 此处的显示窗口必须为有效的句柄, 否则预录回调函数不能设置成功 bFlag = PlayM4_Play(IPort,g_hWnd); break; case NET_DVR_STREAMDATA: //送入待解码的数据 bFlag = PlayM4_InputData(IPort,pBuffer,dwBufSize); //若未成功设置过预录数据回调函数, 且成功解码出第一帧数据, 则设置预录数据回调函数 if ((g_bFlag== FALSE)&&(g_bDecFlag == TRUE)) { Sleep(100); OutputDebugString("zytest:: set dec callback success !"); //设置预录数据回调函数 g_bFlag = PlayM4_SetPreRecordCallBack(IPort,fPreRecordCBfun,pUser); if (g_bFlag == FALSE) { DWORD dwErr = PlayM4_GetLastError(IPort); } else { OutputDebugString("zytest:: set pre record callback success !"); } } break; }</pre> <p>注:</p> <p>参数 1: nPort 表示已经获取的播放库内部分配的端口号 ;</p> <p>参数 2: PreRecordCBfun 表示设置预录像回调函数;</p> <p>参数 3: NULL 表示用户数据为 NULL;</p> |
| | |

[返回目录](#)

5.12.2 设置预录像开关 **PlayM4_SetPreRecordFlag**

| | | |
|------|----------------------------------------------------|----------------------------|
| 函 数: | int PlayM4_SetPreRecordFlag(int nPort, bool bFlag) | |
| 参 数: | int nPort | 播放通道号 |
| | bool bFlag | 设置预录像开关: bFlag = 1 表示开启 |

| | | |
|------|----------------------------------------------------------|----------------|
| | | bFlag = 0 表示关闭 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 预录功能是把通过网络传输过来的设备码流转换为 PS 封装，通过回调函数返回给用户，方便用户保存文件或者其他处理。 | |
| 注 意: | | |

[返回目录](#)

5.13 鱼眼功能

7.3.2 及以后版本鱼眼接口

5.13.1 开启鱼眼功能 **PlayM4_FEC_Enable**

| | | |
|------|--------------------------------------------------------------------------------------|--------|
| 函 数: | int PlayM4_FEC_Enable(int nPort) | |
| 参 数: | int nport | 播放通道号; |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.2 关闭鱼眼功能 **PlayM4_FEC_Disable**

| | | |
|------|--------------------------------------------------------------------------------------|--------|
| 函 数: | int PlayM4_FEC_Disable(int nPort) | |
| 参 数: | int nport | 播放通道号; |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.3 获取鱼眼矫正处理子端口 **PlayM4_FEC_GetPort**

| | | |
|------|--------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 函 数: | int PlayM4_FEC_GetPort(int nPort , int* nSubPort, FECPLACETYPE emPlaceType , FECCORRECTTYPE emCorrectType) | |
| 参 数: | int nport | 播放通道号; |
| | int* nSubPort | 处理子窗口 |
| | FECPLACETYPE emPlaceType | 安装方式 |
| | FECCORRECTTYPE emCorrectType | 矫正方式 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 三代鱼眼支持 4 路展开，子端口取值为[2~5]。 | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.4 获取鱼眼矫正处理子端口 **PlayM4_FEC_DelPort**

| | | |
|------|--------------------------------------------------------------------------------------|--------|
| 函 数: | int PlayM4_FEC_DelPort(int nPort , int nSubPort) | |
| 参 数: | int nport | 播放通道号; |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | 三代鱼眼支持 4 路展开, 子端口取值为[2~5]。 | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.5 设置鱼眼矫正参数 **PlayM4_FEC_SetParam**

| | | |
|------|------------------------------------------------------------------------------------------|----------------------------------|
| 函 数: | int PlayM4_FEC_SetParam(int nPort , int nSubPort , FISHEYEPARAM * pPara) | |
| 参 数: | int nport int* nSubPort FISHEYEPARAM * pPara | 播放通道号; 处理子窗口 鱼眼矫正参数结构体参数指针 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.6 获取鱼眼矫正参数 **PlayM4_FEC_GetParam**

| | | |
|------|------------------------------------------------------------------------------------------|----------------------------------|
| 函 数: | int PlayM4_FEC_GetParam(int nPort , int nSubPort , FISHEYEPARAM * pPara) | |
| 参 数: | int nport int* nSubPort FISHEYEPARAM * pPara | 播放通道号; 处理子窗口 鱼眼矫正参数结构体参数指针 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.7 设置显示窗口 **PlayM4_FEC_SetWnd**

| | |
|------|---------------------------------------------------------------|
| 函 数: | int PlayM4_FEC_SetWnd(int nPort , int nSubPort , void * hWnd) |
|------|---------------------------------------------------------------|

| | | | |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|-----------------------------|
| 参 数: | int int* void* | nport nSubPort hWnd | 播放通道号; 处理子窗口 设置显示窗口句柄 |
| 返回值: | 成功返回 1; 失败返回 0 | | |
| 说 明: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 可随时切换。 | | |
| 注 意: | | | |
| 示例代码 | <pre> //启动鱼眼功能 PlayM4_FEC_Enable(m_IPort); PlayM4_FEC_GetPort(m_IPort,&m_nSubPort1,FEC_PLACE_CEILING,FEC_CORRECT_PTZ)); //同时更新位置和Zoom 值 //更新的类型 stFEPara1.nUpDateType = FEC_UPDATE_PTZPARAM FEC_UPDATE_PTZZOOM; //PTZ 显示的中心坐标 stFEPara1.stPTZParam.fPTZPositionX = 0.2; stFEPara1.stPTZParam.fPTZPositionY = 0.3; //PTZ 显示的范围参数 stFEPara1.fZoom = 0.1; //stCycleParam 用于圆心对当前鱼眼所有子端口都有影响 PlayM4_FEC_SetParam(m_IPort,m_nSubPort1,&stFEPara1); //子端口 2 顶装, PTZ 矫正 获取鱼眼矫正处理子端口 PlayM4_FEC_GetPort(m_IPort,&m_nSubPort2,FEC_PLACE_CEILING,FEC_CORRECT_PTZ); //更新位置 stFEPara2.nUpDateType = FEC_UPDATE_PTZPARAM; stFEPara2.stPTZParam.fPTZPositionX = 0.7; stFEPara2.stPTZParam.fPTZPositionY = 0.8; PlayM4_FEC_SetParam(m_IPort,m_nSubPort2,&stFEPara2); //Zoom stFEPara3.nUpDateType = FEC_UPDATE_PTZZOOM; stFEPara3.fZoom = 0.9; PlayM4_FEC_SetParam(m_IPort,m_nSubPort2,&stFEPara3); //更新位置 stFEPara3.nUpDateType = FEC_UPDATE_PTZPARAM; stFEPara3.stPTZParam.fPTZPositionX = 0.7; </pre> | | |

| | |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre>stFEPara3.stPTZParam.fPTZPositionY = 0.8; stFEPara3.fZoom = 0.9; PlayM4_FEC_SetParam(m_IPort,m_nSubPort2,&stFEPara3) //Zoom stFEPara3.nUpDateType = FEC_UPDATE_PTZZOOM; PlayM4_FEC_SetParam(m_IPort,m_nSubPort2,&stFEPara3); PlayM4_FEC_SetWnd(m_IPort,m_nSubPort1,hWnd1);</pre> <p>注:</p> <p>参数 1: nPort 表示已经获取的播放库内部分配的端口号 ;</p> <p>参数 2: 1 表示子端口号;</p> <p>参数 3: hWnd 表示窗口句柄;</p> |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

[返回目录](#)

5.13.8 获取当前选择鱼眼子端口 **PlayM4_FEC_GetCurrentPTZPort**

| | | | | | | | | | | | | | | | | | |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|-------|--------|------|-----------|------------------------------------|-------|------------|------------------|-------|------------|------------------|--------------------|--------|------|--|
| 函 数: | int PlayM4_FEC_GetCurrentPTZPort(int nPort, bool bPanorama, float fPositionX,float fPositionY, unsigned int *pnPort) | | | | | | | | | | | | | | | | |
| 参 数: | <table><tr><td>int</td><td>nport</td><td>播放通道号;</td></tr><tr><td>bool</td><td>bPanorama</td><td>取值说明如下: false表示操作原图; true则表示操作全景图。</td></tr><tr><td>float</td><td>fPositionX</td><td>鼠标x位置 (接触屏幕点x坐标)</td></tr><tr><td>float</td><td>fPositionY</td><td>鼠标y位置 (接触屏幕点y坐标)</td></tr><tr><td>[out]unsigned int*</td><td>pnPort</td><td>子端口号</td></tr></table> | int | nport | 播放通道号; | bool | bPanorama | 取值说明如下: false表示操作原图; true则表示操作全景图。 | float | fPositionX | 鼠标x位置 (接触屏幕点x坐标) | float | fPositionY | 鼠标y位置 (接触屏幕点y坐标) | [out]unsigned int* | pnPort | 子端口号 | |
| int | nport | 播放通道号; | | | | | | | | | | | | | | | |
| bool | bPanorama | 取值说明如下: false表示操作原图; true则表示操作全景图。 | | | | | | | | | | | | | | | |
| float | fPositionX | 鼠标x位置 (接触屏幕点x坐标) | | | | | | | | | | | | | | | |
| float | fPositionY | 鼠标y位置 (接触屏幕点y坐标) | | | | | | | | | | | | | | | |
| [out]unsigned int* | pnPort | 子端口号 | | | | | | | | | | | | | | | |
| 返回值: | 成功返回 1; 失败返回 0 | | | | | | | | | | | | | | | | |
| 说 明: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持 4 路展开, 子端口取值为[2~5]。 | | | | | | | | | | | | | | | | |
| 注 意: | | | | | | | | | | | | | | | | | |

[返回目录](#)

5.13.9 设置当前选择鱼眼子端口 **PlayM4_FEC_SetCurrentPTZPort**

| | | | | | | | | |
|--------------|-------------------------------------------------------------------------------------------------------------------------------|--------|-------|--------|--------------|----------|------|--|
| 函 数: | int PlayM4_FEC_SetCurrentPTZPort(int nPort, unsigned int nSubPort) | | | | | | | |
| 参 数: | <table><tr><td>int</td><td>nport</td><td>播放通道号;</td></tr><tr><td>unsigned int</td><td>nSubPort</td><td>子端口号</td></tr></table> | int | nport | 播放通道号; | unsigned int | nSubPort | 子端口号 | |
| int | nport | 播放通道号; | | | | | | |
| unsigned int | nSubPort | 子端口号 | | | | | | |
| 返回值: | 成功返回 1; 失败返回 0 | | | | | | | |
| 说 明: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | | | | | | | |

| | |
|------|------------------------------------|
| | 7.3.2 及以后版本鱼眼支持 4 路展开，子端口取值为[2~5]。 |
| 注 意： | |

[返回目录](#)

5.13.10 获取当前子端口号对应的当前 PTZ 坐标值

PlayM4_FEC_SetPTZOutLineShowMode

| | | | |
|------|----------------------------------------------------------------------------------------------------------------------------|----------------------|----------------|
| 函 数： | int PlayM4_FEC_SetPTZOutLineShowMode(int nPort, FECSHOWMODE nPTZShowMode) | | |
| 参 数： | int FECSHOWMODE | nport PTZShowMode | 播放通道号； 展现模式 |
| 返回值： | 成功返回 1；失败返回 0 | | |
| 说 明： | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 7.3.2 及以后版本鱼眼支持 4 路展开，子端口取值为[2~5]。 | | |
| 注 意： | | | |

[返回目录](#)

5.13.11 获取当前子端口号对应的当前 PTZ 坐标值

PlayM4_FEC_PTZ2Window

| | | | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 函 数： | int PlayM4_FEC_PTZ2Window(int nPort , int nSubPort , PTZPARAM stPTZRefOrigin , PTZPARAM stPTZRefWindow , PTZPARAM stPTZWindow , float * fxWindow , float * fyWindow) | | |
| 参 数： | int int PTZPARAM PTZPARAM PTZPARAM [out]float * [out]float* | nport nSubPort stPTZRefOrigin stPTZRefWindow stPTZWindow fxWindow fyWindow | 播放通道号； 设置期望的子port号 当前PTZ坐标值 手指或鼠标触碰控制区域的初始坐标值 手指或鼠标触碰控制区域的当前坐标值 计算出新的PTZ坐标X值 计算出新的PTZ坐标Y值 |
| 返回值： | 成功返回 1；失败返回 0 | | |
| 说 明： | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | | |

| | |
|------|------------------------------------|
| | 7.3.2 及以后版本鱼眼支持 4 路展开，子端口取值为[2~5]。 |
| 注 意： | |

[返回目录](#)

7.3.1 及以前版本鱼眼接口

5.13.12 广角图像矫正 **PlayM4_SetImageCorrection**

| | | |
|------|--------------------------------------------------------------------------------------|------------------------------------------------|
| 函 数： | int PlayM4_SetImageCorrection(int nPort, bool bFlag) | |
| 参 数： | int nport bool bFlag | 播放通道号； bFlag = true 开启； bFlag = false 关闭 |
| 返回值： | 成功返回 1；失败返回 0 | |
| 说 明： | 120° 广角图像矫正，bFlag = true 开启，bFlag = false 关闭 | |
| 注 意： | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.13 设置鱼眼矫正类型 **PlayM4_SetFECDisplayEffect**

| | | |
|------|------------------------------------------------------------------------------------------------------------|---------------------------|
| 函 数： | int PlayM4_SetFECDisplayEffect(int nPort, int nRegionNum, VRDISPLAYEFFECT enDisplayEffect) | |
| 参 数： | int nport int nRegionNum VRDISPLAYEFFECT enDisplayEffect | 播放通道号； 窗口编号； 鱼眼矫正类型 |
| 返回值： | 成功返回 1；失败返回 0 | |
| 说 明： | 鱼眼全景图像矫正后播放画面为正常图像。接口在 PlayM4_Play 之后调用。 | |
| 示 例 | | |
| 注 意： | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.14 设置鱼眼矫正参数 **PlayM4_SetFECDisplayParam**

| | | |
|------|-----------------------------------------------------------------------------------------------------|-----------------|
| 函 数： | int PlayM4_SetFECDisplayParam(int nPort, int nRegionNum, VRFISHPARAM *pstFishParam) | |
| 参 数： | int nport int nRegionNum | 播放通道号； 窗口编号； |

| | | |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| | VRFISHPARAM *pstFishParam | 鱼眼矫正参数 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 说 明: | 参数需要满足以下条件： $fRXLeft < fRXRight$; $fRYTop < fRYBottom$; PTZ 矫正中心坐标需要位于点 (0.5, 0.5) 为圆心和半径为 0.4 的圆内。 | |
| 示例代码: | <pre>int PlayM4_SetFECDisplayParam (int nPort, int nRegionNum ,VRFISHPARAM *pstFishParam); VRFISHPARAM pPara = new VRFISHPARAM(); pPara.fRXLeft = 0.0; pPara.fRXRight = 1.0; pPara.fRYTop = 0.1; pPara.fRYBottom = 0.9; pPara.fAngle = 100; pPara.fZoom = 0.0; pPara.fPTZX = 0.5; pPara.fPTZY = 0.5; int bFlag = setParam(nPort,1, &pPara);</pre> <p><u>注:</u> 参数 1: <i>nPort</i> 表示已经获取的播放库内分配的端口号 ; 参数 2: <i>1</i> 表示鱼眼矫正处理子端口号; 参数 3: <i>pPara</i> 表示设置的参数;</p> | |
| 注 意: | 需要新的窗口实现鱼眼效果需要先使用电子放大创建对应的新的显示窗口。 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.13.15 获取鱼眼矫正参数 `PlayM4_GetFECDisplayParam`

| | | |
|------|------------------------------------------------------------------------------------------------------------------------|---------------------------|
| 函 数: | int PlayM4_GetFECDisplayParam(int nPort, int nRegionNum, VRFISHPARAM *pstFishParam) | |
| 参 数: | int nport int nRegionNum VRFISHPARAM * pstFishParam | 播放通道号; 窗口编号; 鱼眼矫正参数 |
| 返回值: | 成功返回 1；失败返回 0 | |
| 注 意: | 7.3.1 及之前版本和 7.3.2 及以后版本鱼眼接口不可同时使用。 7.3.2 及以后版本鱼眼支持硬解码和翻转。建议使用 7.3.2 及其以后版本的鱼眼接口。 | |

[返回目录](#)

5.14 私有数据

5.14.1 渲染私有数据 `PlayM4_RenderPrivateData`

| | | |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| 函 数: | <code>int PlayM4_RenderPrivateData(int nPort, int nIntelType, int bTrue)</code> | |
| 参 数: | <code>int nport</code> <code>bool nIntelType</code> <code>int bTrue</code> | 播放通道号; 内部类型, 按位表示, 每一位表示一种私有数据类型, 具体类型请参考 PLAYM4_PRIDATA_RENDER 是否显示 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | <ul style="list-style-type: none">➤ 此接口适用于带有智能信息的码流, 若无智能信息的码流则调用此接口没有效果。POS 信息叠加和图片叠加显示是默认关闭的, 其他三个类型显示是默认开启的。➤ 硬解码支持私有数据显示。 | |

[返回目录](#)

5.14.2 渲染私有数据 `PlayM4_RenderPrivateDataEx`

| | | |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 函 数: | <code>int PlayM4_RenderPrivateDataEx(int nPort, int nIntelType, int nSubType, int bTrue)</code> | |
| 参 数: | <code>int nport</code> <code>bool nIntelType</code> <code>int nSubType</code> <code>int bTrue</code> | 播放通道号; 内部类型, 按位表示, 每一位表示一种私有数据类型, 具体类型请参考 PLAYM4_PRIDATA_RENDER 子类型。 当 <code>nIntelType</code> 为 <code>PLAYM4_RENDER_FIRE_DETCET</code> , 则子类型取值为 PLAYM4_FIRE_ALARM ; 当 <code>nIntelType</code> 为 <code>PLAYM4_RENDER_TEM</code> , 则子类型取值为 PLAYM4_TEM_FLAG 。 是否显示 |
| 返回值: | 成功返回 1; 失败返回 0 | |
| 说 明: | | |
| 注 意: | <ul style="list-style-type: none">➤ 此接口适用于带有智能信息的码流, 若无智能信息的码流则调用此接口没有效果。POS 信息叠加和图片叠加显示是默认关闭的, 其他三个类型显示是默认开启的。➤ 硬解码支持私有数据显示。 | |

[返回目录](#)

6 数据结构描述

6.1 图像和声音信息结构体 **FRAME_INFO**

```
struct{
    int          nWidth;           //画面宽，单位为像素；若是音频数据则为音频声道数
    int          nHeight;         //画面高，若为音频数据则为样位率
    int          nStamp;          //时标信息，单位毫秒
    int          nType;           //数据类型，详见下表说明
    int          nFrameRate;      //编码时产生的图像帧率，若是音频数据则为采样率
    unsigned int dwFrameNum;      //帧号
}FRAME_INFO;
```

nType 类型说明表格：

| nType 宏定义 | nFrameType 含义 |
|-----------|----------------------------------------------------------------------------------------------|
| T_AUDIO16 | 音频数据；采样率 16Khz，单声道，每个采样点 16 位表示；具体以解码出的音频数据信息为准 |
| T_RGB32 | 视频数据。每个像素点 4 个字节，排列方式与位图相似。“G-G-R-0...”，第一个像素点位于图像左下角；保留 |
| T_UYVY | 视频数据，uyvy 格式。“U0-Y0-V0-Y1-U2-Y2-V2-V3...”，第一个像素点位于图像左上角，保留 |
| T_YV12 | 视频数据，yv12 格式，当前解码后的视频格式。排列顺序“Y0-Y1-.....”，“V0-V1-.....”，“U0-U1-.....”。关于图像格式详细信息，用户可以查阅相关资料。 |

使用接口如下：

[PlayM4_SetDecCallBack](#)、[PlayM4_SetDecCallBackMend](#)

6.2 显示信息结构体 **DISPLAY_INFO**

```
struct{
    int    nPort;           //通道号
    char*  pBuf;            //返回图像数据指针
    int    nBufLen;         //返回图像数据大小
    int    nWidth;          //画面宽
    int    nHeight;         //画面高
    int    nStamp;          //时标信息，单位毫秒
    int    nType;           //数据类型，具体说明如下表所示：
```

```
void* nUser; //用户数据
}DISPLAY_INFO;
```

nType 类型说明表格:

| nType 宏定义 | nFrameType 含义 |
|-----------|---------------------------------------------------------------------------------------------------|
| T_AUDIO16 | 音频数据; 采样率 16Khz, 单声道, 每个采样点 16 位表示; 具体以解码出的音频数据信息为准 |
| T_RGB32 | 视频数据。每个像素点 4 个字节, 排列方式与位图相似。“G-G-R-0...”, 第一个像素点位于图像左下角; 保留 |
| T_UYVY | 视频数据, uyvy 格式。“U0-Y0-V0-Y1-U2-Y2-V2-V3...”, 第一个像素点位于图像左上角, 保留 |
| T_YV12 | 视频数据, yv12 格式, 当前解码后的视频格式。排列顺序“Y0-Y1-.....”, “V0-V1-.....”, “U0-U1-.....”。关于图像格式详细信息, 用户可以查阅相关资料。 |

使用接口如下:

[PlayM4_SetDisplayCallBackEx](#)

6.3 全局时间结构体 **PLAYM4_SYSTEM_TIME**

```
struct{
    unsigned int dwYear; //年
    unsigned int dwMon; //月
    unsigned int dwDay; //日
    unsigned int dwHour; //时
    unsigned int dwMin; //分
    unsigned int dwSec; //秒
    unsigned int dwMs; //毫秒
}PLAYM4_SYSTEM_TIME;
```

使用接口如下:

[PlayM4_GetSystemTime](#)

6.4 区域参数结构体 **HKRECT**

```
typedef struct tagHKRect
{
    unsigned long nLef; //左
    unsigned long nTop; //上
    unsigned long nRight; //右
    unsigned long nBottom; //下
}
```

}HKRECT;

使用接口如下:

[PlayM4_SetDisplayRegion](#)

6.5 加密信息结构体 ENCRYPT_INFO

```
typedef struct{
    long nVideoEncryptType;           //视频加密类型
    long nAudioEncryptType;          //音频加密类型
    long nSetSecretKey;               //是否设置, 1 表示设置密钥, 0 表示没有设置密钥
}ENCRYPT_INFO;
```

使用接口如下:

[PlayM4_SetEncryptTypeCallBack](#)

6.6 私有数据显示信息类型结构体 PLAYM4_PRIDATA_RENDER

```
enum _PLAYM4_PRIDATA_RENDER{
    PLAYM4_RENDER_ANA_INTEL_DATA    = 0x00000001,    //智能分析
    PLAYM4_RENDER_MD                = 0x00000002,    //移动侦测
    PLAYM4_RENDER_ADD_POS            = 0x00000004,    //POS 信息后叠加
    PLAYM4_RENDER_ADD_PIC           = 0x00000008,    //图片叠加
    PLAYM4_RENDER_FIRE_DETECT       = 0x00000010,    //热成像信息
    PLAYM4_RENDER_TEM                = 0x00000020,    //温度信息
}PLAYM4_PRIDATA_RENDER
```

使用接口如下:

[PlayM4_RenderPrivateData](#)

6.7 热成像信息结构体 PLAYM4_FIRE_ALARM

```
typedef enum _PLAYM4_FIRE_ALARM
{
    PLAYM4_FIRE_FRAME_DIS          = 0x00000001,    //火点框显示
    PLAYM4_FIRE_MAX_TEMP           = 0x00000002,    //最高温度
    PLAYM4_FIRE_MAX_TEMP_POSITION  = 0x00000004,    //最高温度位置显示
    PLAYM4_FIRE_DISTANCE           = 0x00000008,    //最高温度距离
}PLAYM4_FIRE_ALARM;
```

使用接口如下:

[PlayM4_RenderPrivateDataEx](#)6.8 测温信息结构体 **PLAYM4_TEM_FLAG**

```
typedef enum _PLAYM4_TEM_FLAG
{
    PLAYM4_TEM_REGION_BOX           = 0x00000001,    //框测温
    PLAYM4_TEM_REGION_LINE         = 0x00000002,    //线测温
    PLAYM4_TEM_REGION_POINT        = 0x00000004,    //点测温
}PLAYM4_TEM_FLAG;
```

使用接口如下：

[PlayM4_RenderPrivateDataEx](#)6.9 鱼眼矫正显示效果结构体 **VRDISPLAYEFFECT**

```
typedef enum tagVRDisplayEffect
{
    VR_ET_NULL                     = 0x100,          //不矫正
    VR_ET_FISH_PTZ_CEILING         = 0x101,          //应用于顶装鱼眼
    VR_ET_FISH_PTZ_FLOOR           = 0x102,          //应用于地面安装鱼眼
    VR_ET_FISH_PTZ_WALL            = 0x103,          //应用于壁装鱼眼
    VR_ET_FISH_PANORAMA_CEILING360 = 0x104,          //应用于顶装鱼眼 1P
    VR_ET_FISH_PANORAMA_CEILING180 = 0x105,          //应用于顶装鱼眼 2P
    VR_ET_FISH_PANORAMA_FLOOR360   = 0x106,          //应用于地面安装鱼眼 1P
    VR_ET_FISH_PANORAMA_FLOOR180   = 0x107,          //应用于地面安装鱼眼 2P
    VR_ET_FISH_LATITUDE_WALL        = 0x108,          //应用于壁装维度展开(广角)
    VR_ET_REDBLUE_3D               = 0x109,          //红蓝 3D 效果
}VRDISPLAYEFFECT;
```

使用接口如下：

[PlayM4_SetFECDisplayEffect](#)6.10 鱼眼矫正参数结构体 **VRFISHPARAM**

```
typedef struct tagVRFishParam
{
    float fRXLeft;                //水平坐标(min)
    float fRXRight;               //水平坐标(max)
    float fRYTop;                 //垂直坐标(min)
    float fRYBottom;              //垂直坐标(max)
    float fAngle;                 //180° 矫正中心角
```



```

float fZoom;           //PTZ 矫正放大系数
float fPTZX;           //PTZ 矫正的中心坐标
float fPTZY;           //PTZ 矫正的中心坐标
}VRFISHPARAM;

```

使用接口如下：

[PlayM4_SetFECDisplayParam](#)、[PlayM4_GetFECDisplayParam](#)

6.11 鱼眼安装方式 **FECPLACETYPE**

```

typedef enum tagFECPlaceType
{
    FEC_PLACE_WALL      = 0x1,      //壁装方式 (法线水平)
    FEC_PLACE_FLOOR     = 0x2,      //地面安装 (法线向上)
    FEC_PLACE_CEILING   = 0x3,      //顶装方式 (法线向下)
}FECPLACETYPE;

```

使用接口如下：

[PlayM4_FEC_GetPort](#)

6.12 鱼眼矫正方式 **FECCORRECTTYPE**

```

typedef enum tagFECCorrectType
{
    FEC_CORRECT_PTZ = 0x100,      // PTZ
    FEC_CORRECT_180 = 0x200,      // 180 度矫正 (对应 2P)
    FEC_CORRECT_360 = 0x300,      // 360 全景矫正 (对应 1P)
}FECCORRECTTYPE;

```

使用接口如下：

[PlayM4_FEC_GetPort](#)

6.13 PTZ 在原始鱼眼图上轮廓的显示模式 **FECSHOWMODE**

```

typedef enum tagFECShowMode
{
    FEC_PTZ_OUTLINE_NULL,      //不显示
    FEC_PTZ_OUTLINE_RECT,      //矩形显示
    FEC_PTZ_OUTLINE_RANGE,      //真实区域显示
}FECSHOWMODE;

```

使用接口如下：

[PlayM4_FEC_SetPTZOutLineShowMode](#)

6.14 鱼眼图像圆心参数 **CYCLEPARAM**

```
typedef struct tagCycleParam
{
    float    fRadiusLeft;           //圆的最左边 X 坐标
    float    fRadiusRight;          //圆的最右边 X 坐标
    float    fRadiusTop;            //圆的最上边 Y 坐标
    float    fRadiusBottom;         //圆的最下边 Y 坐标
}CYCLEPARAM;
```

使用结构体如下：

[FISHEYEPARAM](#)

6.15 PTZ 校正的参数结构体 **PTZPARAM**

```
typedef struct tagPTZParam
{
    float fPTZPositionX;           // PTZ 显示的中心位置 X 坐标
    float fPTZPositionY;           // PTZ 显示的中心位置 Y 坐标
}PTZPARAM;
```

使用结构体如下：

[FISHEYEPARAM](#)

6.16 鱼眼矫正参数结构体 **FISHEYEPARAM**

```
typedef struct tagFECParam
{
    unsigned int  nUpDateType;       //更新的类型
    unsigned int  nPlaceAndCorrect;  //安装方式和矫正方式，只能用于获取，SetParam
                                     //的时候无效,该值表示安装方式和矫正方式的和
    PTZPARAM      stPTZParam;         // PTZ 校正的参数
    CYCLEPARAM    stCycleParam;       //鱼眼图像圆心参数
    Float         fZoom;              // PTZ 显示的范围参数
    Float         fWideScanOffset;    // 180 或者 360 度校正的偏移角度
    Int           nResver[16];        // 保留字段
}FISHEYEPARAM;
```

使用接口如下：

[PlayM4 FEC SetParam](#)、[PlayM4 FEC GetParam](#)

7 常见问题解答

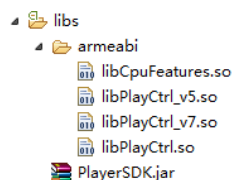
移动播放库特有问题的介绍

Q: 移动播放库涵盖哪几个平台？系统版本版本要求分别是多少？

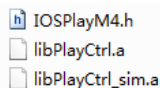
- A: 1、Android 平台播放库，要求 Android 2.3 及以上；
2、iOS 平台播放库，要求 iOS 6.0 及以上；
3、Windows Phone 平台播放库，要求 Wp8 及以上，目前未对外提供。

Q: 移动播放库各平台库文件组织形式是怎么样的？

- A: 1、Android 平台播放库，共包含一个 jar 包，4 个库文件，其中 PlayCtrl 可根据 cpu 类型只取其一，默认全部放在 app 工程目录：



- 2、iOS 平台播放库，包含一个头文件，两个库文件，其中后缀 sim 的库文件为模拟器版本，可选；



Q: 预览、回放播放画面黑屏无图像怎么办？

- A: 1、查看确认操作系统版本是否满足要求；
2、确认码流是否为加密码流，密钥是否有误；
3、使用海康 PC 端客户端、播放器确认码流是否有误；
4、使用 logcat 工具查看是否存在 “No Configs for your device!” 日志。

Q: 碰到特有机型才有的问题怎么处理？

- A: 1、记录手机系统版本；
2、记录手机型号信息，包括品牌、发行地区、芯片信息等，越详细越好。
3、如果是花屏问题，保存码流、截屏。

Q: Android 播放库闪退怎么办？

- A: 1、确认播放库文件有没有遗漏，尤其注意 libCpuFreatrue.so 是否遗漏；
2、确认手机 cpu 为 arm 架构，非 x86 等；
3、使用 logcat 工具查看闪退日志；

Q: 打开播放没有声音怎么办？

- A: 1、使用海康 PC 端客户端、播放器确认码流是否包含音频;
- 2、如果是 G722 音频格式，则可以在录像后查看 40 字节文件头，第 13、14 个字节表示音频格式（下图 c、d），合起来是 0x7221，部分老设备可能会打成 0x1011;

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00000000h: | 49 | 4D | 4B | 48 | 01 | 01 | 00 | 00 | 02 | 00 | 00 | 01 | 21 | 72 | 01 | 10 |
| 00000010h: | 80 | 3E | 00 | 00 | 80 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00000020h: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | BA | 6E | C0 | 75 | 9D |
| 00000030h: | 84 | 01 | 02 | 3D | 77 | FE | FF | FF | 00 | 3E | 49 | 90 | 00 | 00 | 01 | BC |

关于流模式接口

Question 1 送入到播放库 PlayM4_OpenStream 接口报错

Answer 1

1. 检查参数是否正确，调用接口 [PlayM4_OpenStream](#)，需要确认送入正确的文件头（通过网络 SDK 中的实时流回调函数中获取的 NET_DVR_SYSHEAD，或者板卡中编码回调函数中获取的 PktSysHeader 类型的即为正确的文件头内容以及文件头长度）；
2. 调用失败后，若返回错误码是 PLAYM4_ALLOC_MEMORY_ERROR，说明内存不足，可以查看 pc 当前内存情况使用情况排查；
3. 此路流数据操作结束后，可以调用 [PlayM4_CloseStream](#) 接口关闭流回收系统资源；[PlayM4_CloseStream](#) 和 [PlayM4_OpenStream](#) 需要配对出现。

流模式预览

Question 2 实时流预览有卡顿

Answer 2

1. 检查数据是否流畅，若数据不流畅则检查网络部分；
4. 由于不同的分辨率对设备的性能要求不一样，如高清分辨率对设备性能要求比较高，多路高清实时预览时可能会有卡顿现象，此时可以适当降低预览的路数；
5. 检查视频当前的分辨率，缓冲区若开的太小，则可能送入数据到播放库缓冲区 [PlayM4_InputData](#) 满，此时需要重复送入未送成功的数据；
建议：若为 4CIF 分辨率，则建议 [PlayM4_OpenStream](#) 缓冲开到 1.5M 到 2M，1.5M 即 $1.5 * 1024 * 1024$ 。若 720P 则建议开到 3~4M；若 1080P 则建议开大到 6M 左右；
6. 建议检查 [PlayM4_SetDisplayBuf](#)，设置为 15 个节点即可；
7. 可以将数据保存下来用播放器验证，若播放卡顿，则确认在网络部分是否丢数据，若播放正常则检查送数据部分是否正常。

Question 3 实时流预览不显示图像

Answer 3

1. 确认是否有正确的数据送入到播放库缓冲，且调用过 [PlayM4_Play](#) 开始解码；

2. 确认是否注册了解码回调不显示的回调函数 [PlayM4_SetDecCallBack](#);
3. 确认是否可以正常抓取到图片; 若不能抓图成功则转 1, 若能抓图则转 4;
4. 抓图成功, 则连续抓图, 若图片随场景和 OSD 有变化, 则表示数据正常, 则检查显示的窗口句柄是否正确;
5. 可以将需要解码的数据保存为本地文件, 然后用播放器进行播放验证, 若能播放则转步骤 6, 若不能播放则转步骤 7;
6. 若可以正常播放, 则检查 [PlayM4_InputData](#) 送数据部分是否正确;
7. 若不能正常播放, 则检查数据码流是否正确, 若是标准码流则可以用 Elecard 或者 VLC 等标准第三方解码器进行验证; 若第三方的解码器可以解码则可以将测试码流发给我们分析; 若第三方的解码器不能解码则检查是否有网络丢包的现象;
8. 如果实时码流是 RTP 封装的码流, 则最好提供网络抓包文件, 或者如下格式: 4 字节 RTP 包的长度+RTP 数据包保存, 然后发给我们确认。

Question 4 实时流预览延时

Answer 4

1. 一般设备延时在 100 到 200ms 之间, 延时和网络的流畅性有很大关系。如在全帧率下延时正常, 而在低帧率情况下比较明显, 可以反馈给我们确认。6.2.2.2 版本以后, 延时会达到最小;
2. 如果是多路预览才出现延时的现象, 则请查看当前 CPU 的使用情况, 可能是 CPU 已经达到极限 (85%以上), 无法支持这么多路的实时流解码, 可以尝试减少几路播放, 或者提高硬件性能;
3. 若客户使用的是客户端软件, 则可以设置为流畅性好, 若是调用 SDK 进行二次开发, 则可以调用接口 [PlayM4_SetDisplayBuf](#) 设置解码后缓冲区帧数;
4. 确定接口 [PlayM4_SetStreamOpenMode](#) 设定为 STREAM_REALTIME;
5. 确定网络良好, 收发网络包延时较小。

Question 5 实时流预览显示第一帧画面比较慢

Answer 5

在低帧率下比较常见, 检查第一帧是否 I 帧, 若第一帧不是 I 帧则需要到 I 帧才能正常解码, 所以时间可能比较长。

实时流预览时，网络环境较差也有可能引起该问题；

开启硬解码优先时，码流符合硬解码要求，但是解码过程中出错需要切换软解时，也会出现该现象。

Question 6 加密码流无法正常播放

Answer 6

1. 确认码流是否是基线的码流，若是定制的码流则需要联系分公司确认加密方式和订单号；
2. 确认密钥和调用设置密钥接口被正确调用，且密钥正确 [PlayM4_SetSecretKey](#)；
3. 可以将码流发给我们确认。

Question 7 如何将捕获的图片保存在内存中以供后续处理

Answer 7

在开始正常解码播放 [PlayM4_Play](#) 以后，可以通过播放库函数 [PlayM4_GetBMP](#) 和 [PlayM4_GetJPEG](#) 把图片保存在内存中。

Question 8 音频播放卡顿，视频播放流畅

Answer 8

确认音频数据是否均匀。

Question 9 复合流文件，播放音频时画面卡顿，不播放音频。则画面正常

Answer 9

一般可能存在于低帧率文件，或者音频分布不均匀导致。

Question 10 播放文件，无法听到声音

Answer 10

1. 需要确定码流中是否有音频数据，音频源是有声音的。
2. 音频封装相关表示类型的字段不符合规定，如 RTP 封装 payload 错误，PS 封装 stream type 错误等。
3. 音频未加密而在封装层为加密，在播放库 6.2 系列版本后开始关注音频加密字段。如果之前版本可以正常播放，之后版本不行，则可能是设备问题，需要和设备确认。

Question 11 实时流播放正常，文件播放有快放现象

Answer 11

确认码流中帧率或者时间戳是是否正常；另外可以检查是否有被抽帧的情况。

Question 12 文件打开失败

Answer 12

检查保存文件时是否有保存正确的文件头（设备网络 SDK 的实时流回调函数中码流类型为 NET_DVR_SYSHEAD，或者编码卡编码回调函数中当前帧类型为 PktSysHeader）；文件是否小于 4K 或者大于 4G；或者数据错误；或者没有关键帧（视频解码需要依赖于关键帧）。

Question 13 文件模式如何定位

Answer 13

是否成功创建文件索引；若索引创建成功，检查设定的定位参数（如时间或者帧号）是否正确；

Question 14 PlayM4_SetCurrentFrameNum 定位不准确

Answer 14

是否成功创建文件索引；文件帧号是否乱序；

Question 15 如何获取多路数据流解码后的数据

Answer 15

1. 一路数据对应一个播放库 Port（通过 [PlayM4_GetPort](#) 获取）；
2. 调用 [PlayM4_SetStreamOpenMode](#) 设定正确的播放模式；
3. 送入这路数据对应的文件头 [PlayM4_OpenStream](#)；
4. 调用 [PlayM4_SetDecCallBack](#) 注册解码回调函数，获取解码后的数据；
5. 调用 [PlayM4_Play](#) 开始解码线程（若不需要显示图像，则可以将窗口句柄置为 NULL）；
6. 调用 [PlayM4_InputData](#) 送入每路数据正确的数据；播放库支持多路并发解码，当前版本最多支持 500 路。

Question 16 实时流数据用解码回调有丢帧现象

Answer 16

1. 实时录文件，先排除数据流是否有问题。
2. 如果数据没有问题，注释掉解码回调里面所有的实现，查看是否还有丢帧情况。解码回调里面实现耗时比较多的操作，也有可能出现此类的问题。
3. 请查看在实际调用过程中，检查 [PlayM4 InputData](#) 是否有失败，若有失败，请尝试将数据重新送入，错误发生频率比较高，建议 [PlayM4 OpenStream](#) 中设置缓冲区大小设置大一些。

Question 17 有些码流解码回调函数中获取的每帧视频高度只有显示原来的一半

Answer 17

若原始图像分辨率是 2CIF（PAL 制为 704*288，NTSC 制为 704*240），实际显示是按照 PAL 制 704*576，NTSC 制是 704*480 进行显示，而解码回调函数中获取的是码流实际的视频分辨率，所以会出现这种情况。

科技呵护未来

First Choice for Security Professionals