

# 播放 SDK 编程手册

**VERSION 1.0.1(Build 090905)**

**2009-09-05**

版权所有 侵权必究

## 前 言

非常感谢您使用我们公司的设备，我们将为您提供最好的服务。

本手册可能包含技术上不准确的地方或印刷错误。本手册的内容将做不定期的更新，恕不另行通知；更新的内容将会在本手册的新版本中加入。我们随时会改进或更新本手册中描述的产品或程序。

修订记录

日期	修订内容	作者
2009-09-05	创建	
2011-09-28	增 加 设 置 信 息 帧 回 调 接 口 H264_PLAY_SetInfoFrameCallBack	孙杰
2013-09-02	补全文档	LXQ

# 目 录

VERSION 1.0.1(BUILD 090905) .....	1
2009-09-051	
1 简 介 12	
1.1 概述.....	12
1.2 适用性.....	13
设计原则 14	
1.3 典型调用顺序 .....	14
回调及功能设置.....	14
各功能操作及信息获取 .....	14
H264_PLAY_STOP .....	14
H264_PLAY_CLOSEFILE .....	14
H264_PLAY_FREEPORT .....	14
流模式下: .....	14
回调及功能设置.....	14
H264_PLAY_SETSTREAMOPENMODE.....	14
H264_PLAY_OPENSTREAM.....	14
H264_PLAY_INPUTDATA 或者各功能操作及信息获取 .....	14
1.4 功能和回调说明 .....	15
A、    多显卡初始化 .....	15
B、    回调及功能设置 .....	15
C、    打开文件或数据流.....	16
D、    播放.....	16
E、    各功能操作及信息获取.....	16
F、    停止.....	18
G、    结束.....	18
H、    可随时调用的函数.....	18
1.5 编程补充说明 .....	18
2 数据常量定义 .....	19

2.1 最大通道数 .....	19
2.2 声音波形范围 .....	19
2.3 定时器类型 .....	19
2.4 缓冲类型 .....	19
2.5 错误类型 .....	20
2.6 最大区域显示数 .....	21
2.7 显示类型 .....	21
2.8 解码缓冲数 .....	21
2.9 定位类型 .....	21
2.10 数据流原始缓冲大小 .....	21
2.11 数据流播放模式 .....	21
2.12 解码回调音频帧类型 .....	21
2.13 解码回调视频帧类型 .....	22
2.14 系统功能 .....	22
2.15 对讲端口 .....	22
3 数据结构定义 .....	23
3.1 录像文件类型 .....	23
3.2 本地抓图类型 .....	23
3.3 帧位置 .....	23
3.4 帧信息 .....	23
3.5 帧类型 .....	24
3.6 码流类型 .....	24

3.7 图像帧类型 .....	24
3.8 图像帧子类型 .....	25
3.9 帧数据信息 .....	25
3.10 字体类型.....	26
3.11 OSD 信息 .....	26
3.12 媒体数据类型（标准） .....	27
3.13 标准 H264 结构.....	27
3.14 回调函数.....	28
3.14.1 DECODECALLBACK.....	28
3.14.2 DISPLAYCALLBACK.....	28
3.14.3 SOURCEBUFCALLBACK.....	29
3.14.4 FILECALLBACK .....	29
3.14.5 DRAWCALLBACK.....	30
3.14.6 VERIFYCALLBACK.....	30
3.14.6 AUDIOCALLBACK.....	31
3.14.7 IMAGESIZECHANGECALLBACK .....	32
3.14.8 DEMUXCALLBACK.....	32
3.14.10 AUDIOCAPTURECALLBACK .....	32
3.14.11 INFOFRAMCALLBACK .....	33
3.14.12 INFOFRAMCALLBACK .....	33
3.14.12 WATERMARK_CALLBACK.....	34
API 函数说明.....	35
3.15 打开播放文件 H264_PLAY_OPENFILE .....	35
3.16 开始播放 H264_PLAY_PLAY.....	35
3.17 停止播放 H264_PLAY_STOP.....	36
3.18 关闭播放文件 H264_PLAY_CLOSEFILE .....	36

3.19 打开媒体流 H264_PLAY_OPENSTREAM.....	37
3.20 输入待解码数据 H264_PLAY_INPUTDATA.....	38
3.21 关闭数据流 H264_PLAY_CLOSESTREAM.....	38
3.22 输入视频数据 H264_PLAY_INPUTVIDEODATA.....	39
3.23 输入音频数据 H264_PLAY_INPUTAUDIODATA.....	39
3.24 播放暂停/恢复 H264_PLAY_PAUSE .....	40
3.25 加快播放速度 H264_PLAY_FAST .....	40
3.26 反向回放 H264_PLAY_BACK.....	41
3.27 减慢播放速度 H264_PLAY_SLOW.....	41
3.28 单帧往后播放 H264_PLAY_NEXTFRAME.....	42
3.29 单帧往前播放 H264_PLAY_PREVFRAME.....	42
3.30 打开播放声音 H264_PLAY_PLAYSOUND.....	43
3.31 关闭播放声音 H264_PLAY_STOPSOUND.....	43
3.32 共享方式打开播放声音 H264_PLAY_PLAYSOUNDSHARE .....	44
3.33 共享方式关闭播放声音 H264_PLAY_STOPSOUNDSHARE .....	44
3.34 设置播放声音 H264_PLAY_SETVOLUME.....	45
3.35 获取播放声音 H264_PLAY_GETVOLUME .....	45
3.36 调整 WAVE 波形 H264_PLAY_ADJUSTWAVEAUDIO .....	46
3.37 设置解码回调函数 H264_PLAY_SETDECCALLBACK.....	46
3.38 设置解码回调的码流类型 H264_PLAY_SETDECCBSTREAM.....	47
3.39 设置图像显示回调函数 H264_PLAY_SETDISPLAYCALLBACK.....	47

3.40 设置音频解码后的数据回调函 H264_PLAY_SETAUDIOCALLBACK .....	48
3.41 源数据分析完的数据回调 H264_PLAY_SETDEMUXCALLBACK .....	49
3.42 设置数据校验 H264_PLAY_SETVERIFYCALLBACK.....	50
3.43 设置水印数据回调 H264_PLAY_SETWATERMARKCALLBACK .....	50
3.44 编码格式改变消息 H264_PLAY_SETIMAGESIZECHANGEMSG .....	51
3.45 设置文件结束时要发送的消息 H264_PLAY_SETFILEENDMSG.....	52
3.46 设置源缓冲区阈值及回调指 H264_PLAY_SETSOURCEBUF_CALLBACK.....	53
3.47 重置回调标志为有效状态 H264_PLAY_RESETSOURCEBUF_FLAG.....	53
3.48 分辨率改变回调设 H264_PLAY_SETIMAGESIZECHANGE_CALLBACK.....	54
3.49 设置文件索引建立后回调 H264_PLAY_SETFILEREF_CALLBACK .....	55
3.50 设置文件结束回调 H264_PLAY_SETFILEEND_CALLBACK .....	55
3.51 设置文件索引 H264_PLAY_SETREFVALUE.....	56
3.52 获取文件索引信息 H264_PLAY_GETREFVALUE .....	57
3.53 设置当前播放帧号 H264_PLAY_SETCURRENTFRAMENUM .....	57
3.54 设置当前播放时间 H264_PLAY_SETPLAYEDTIMEEX .....	58
3.55 设置文件播放指针的相对位置 H264_PLAY_SETPLAYPOS .....	59
3.56 获取文件播放指针的相对位置 H264_PLAY_GETPLAYPOS .....	60
3.57 设置图象的视频参数 H264_PLAY_SETCOLOR.....	60
3.58 设置播放缓冲区大小 H264_PLAY_SETDISPLAYBUF .....	61
3.59 设置图象的视频参数 H264_PLAY_SETDISPLAYTYPE .....	62
3.60 设置图象的流畅性 H264_PLAY_ADJUSTFLUENCY .....	62



3.61 设置图象播放的帧率 H264_PLAY_SETPLAYFPS .....	63
3.62 打开音频采集功能 H264_PLAY_STARTAUDIOCAPTURE .....	63
3.63 H264_PLAY_STOPAUDIOCAPTURE .....	64
3.64 设置 OVERLAY 模式 H264_PLAY_SETOVERLAYMODE .....	65
3.65 设置图像质量 H264_PLAY_SETPICQUALITY .....	66
3.66 设置播放模式 H264_PLAY_SETSTREAMOPENMODE .....	67
3.67 设置播放使用的定时器类型 H264_PLAY_SETTIMERTYPE .....	67
3.68 获取当前系统信息 H264_PLAY_GETCAPS .....	68
3.69 获得颜色值 H264_PLAY_GETCOLOR .....	69
3.70 获得播放缓冲的帧数 H264_PLAY_GETDISPLAYBUF .....	69
3.71 获得目前设置的显示模式 H264_PLAY_GETDISPLAYTYPE .....	70
3.72 获得 OVERLAY 层使用的锁结色 H264_PLAY_GETCOLORKEY .....	70
3.73 获得当前使用的 OVERLAY 模式 H264_PLAY_GETOVERLAYMODE .....	71
3.74 获得当前设置的图像质量 H264_PLAY_GETPICTUREQUALITY .....	71
3.75 获得流模式类型 H264_PLAY_GETSTREAMOPENMODE .....	72
3.76 获得定时器类型 H264_PLAY_GETTIMERTYPE .....	72
3.77 获取播放器中的缓冲区大小 H264_PLAY_GETBUFFERVALUE .....	73
3.78 获取当前播放的帧序号 H264_PLAY_GETCURRENTFRAMENUM .....	73
3.79 获取当前码流中编码时的帧率 H264_PLAY_GETPLAYFPS .....	74
3.80 获取文件头长度 H264_PLAY_GETFILEHEADLENGTH .....	74
3.81 获得文件总的时间长度 H264_PLAY_GETFILETIME .....	75
3.82 获得播放文件总的时间帧数 H264_PLAY_GETFILETOTALFRAMES .....	75

3.83 获取指定位置关键帧 H264_PLAY_GETKEYFRAMEPOS .....	76
3.84 指定位置之后关键帧位置 H264_PLAY_GETNEXTKEYFRAMEPOS.....	77
3.85 获得原始图像的大小 H264_PLAY_GETPICTURESIZE .....	78
3.86 获得播放的帧数 H264_PLAY_GETPLAYEDFRAMES .....	78
3.87 获得文件当前播放的时间 H264_PLAY_GETPLAYEDTIME .....	79
3.88 获得文件当前播放的时间 H264_PLAY_GETPLAYEDTIMEEX.....	79
3.89 获得源缓冲剩余数据 H264_PLAY_GETSOURCEBUFFERREMAIN .....	80
3.90 初始化多显卡显示设备 H264_PLAY_INITDDRAWDEVICE.....	80
3.91 H264_PLAY_RELEASEDDRAWDEVICE .....	80
3.92 转换 BMPH264_PLAY_CONVERTTOBMPFILE.....	81
3.93 H264_PLAY_CATCHPIC .....	82
3.94 画图回调 H264_PLAY_RIGISTERDRAWFUN.....	82
3.95 设置或增加显示区域 H264_PLAY_SETDISPLAYREGION.....	83
3.96 刷新显示 H264_PLAY_REFRESHPLAYEX.....	83
3.97 开始流数据录像 H264_PLAY_STARTDATARECORD.....	84
3.98 停止流数据录像 H264_PLAY_STOPDATARECORD.....	85
3.99 清除源缓冲区剩余数据 H264_PLAY_RESETSOURCEBUFFER .....	85
3.100 清空播放器中的缓冲区 H264_PLAY_RESETBUFFER .....	86
3.101 设置搜索区域及范围 H264_PLAY_SETMDRANGE .....	86
3.102 设置智能搜索的阈值 H264_PLAY_SETMDTHRESHOLD .....	87
3.103 获取搜索到的数据帧的帧序号 H264_PLAY_GETMDPOSITION .....	87
3.104 得到播放 SDK 版本号 H264_PLAY_GETSDKVERSION .....	88

3.105 获取错误号 H264_PLAY_GETLASTERROR.....	89
3.106 刷新显示图像 H264_PLAY_REFRESHPLAY .....	89
3.107 设置信息帧回调 H264_PLAY_SETINFOFRAMECALLBACK.....	89
3.108 获取显卡数目 H264_PLAY_GETDDRAWDEVICETOTALNUMS .....	90
3.109 抓图到内存中 H264_PLAY_CATCHPICBUF .....	90
3.110 停止声音采集 H264_PLAY_STOPAUDIOCAPTURE.....	91
3.111 获取空闲通道号 H264_PLAY_GETPORT .....	91
3.112 释放通道号 H264_PLAY_FREEPORT .....	91
3.113 获得正在显示帧的时间戳 H264_PLAY_GETCURTIMESTAMP.....	92
3.114 设置 OSD 叠加信息 H264_PLAY_SETOSDTEX .....	92
3.115 删除 OSD 叠加信息 H264_PLAY_DELETEOSDTEX .....	92
3.116 录像文件格式转换 H264_PLAY_CONVERTFILE .....	93
3.117 只播放 I 帧 H264_PLAY_ONLYIFRAME .....	93
3.118 标准 H264 写入 H264_PLAY_INPUTSTDH264DATA.....	94
3.119 标准 H264 写入 H264_PLAY_INPUTSTDH264DATA.....	94

## 1 简介

### 1.1 概述

解码 SDK 是硬盘录像机的配套产品，支持 h264 码流。本文档详细描述了开发包中各函数实现的功能及接口

播放 SDK 的主要功能有：支持文件或流数据的播放、回放控制（如暂停\恢复、快放慢放）、音频控制、流数据录像、多区域显示、按帧序号或按时间定位、数据回调、消息回调、字符叠加、抓图.....

开发包中包括的文件有：`DLLDeinterlace.dll`、`hi_h264dec_v.dll`、`StreamReader.dll`、`ToJpg.dll`。

## 1.2 适用性

- ❖ 支持h264码流的解码。

## 设计原则

### 1.3 典型调用顺序

应用程序初始化

文件模式：

H264\_PLAY\_GetPort

回调及功能设置

H264\_PLAY\_OpenFile

H264\_PLAY\_Play

各功能操作及信息获取

H264\_PLAY\_Stop

H264\_PLAY\_CloseFile

H264\_PLAY\_FreePort

流模式下：

H264\_PLAY\_GetPort

回调及功能设置

H264\_PLAY\_SetStreamOpenMode

H264\_PLAY\_OpenStream

H264\_PLAY\_Play

H264\_PLAY\_InputData 或者各功能操作及信息获取

H264\_PLAY\_Stop

H264\_PLAY\_CloseStream

H264\_PLAY\_FreePort

应用程序结束

## 1.4 功能和回调说明

### A、多显卡初始化

初始化多显卡设备	<b>H264_PLAY_InitDDrawDevice</b>
获取显卡数目	<b>H264_PLAY_GetDDrawDeviceTotalNums</b>
设置播放窗口使用的显卡	<b>H264_PLAY_SetDDrawDevice</b>
获取显卡信息	<b>H264_PLAY_GetDDrawDeviceInfo</b>
获取显卡功能	<b>H264_PLAY_GetCapsEx</b>
关闭多显卡设备	<b>H264_PLAY_RealeseDDraw</b>

### B、回调及功能设置

设置文件结束消息回调	<b>H264_PLAY_SetFileEndMsg</b>
设置文件结束函数回调	<b>H264_PLAY_SetFileEndCallBack</b>
设置图像格式变动消息回调	<b>H264_PLAY_SetEncChangeMsg</b>
设置图像格式变动函数回调	<b>H264_PLAY_SetEncTypeChangeCallBack</b>
设置文件索引建立后回调	<b>H264_PLAY_SetFileRefCallBack</b>
设置流缓冲小于阈值回调	<b>H264_PLAY_SetSourceBufCallBack</b>
设置数据解码回调	<b>H264_PLAY_SetDecCallBack</b>
设置视频解码回调	<b>H264_PLAY_SetDisplayCallBack</b>
设置音频解码回调	<b>H264_PLAY_SetAudioCallBack</b>
设置水印数据回调	<b>H264_PLAY_SetVerifyCallBack</b>
设置分析数据回调	<b>H264_PLAY_SetDemuxCallBack</b>
设置文件索引	<b>H264_PLAY_SetRefValue</b>
设置播放缓冲帧数	<b>H264_PLAY_SetDisplayBuf</b>
设置 overlay 模式及关键色	<b>H264_PLAY_SetOverlayMode</b>
设置流打开模式	<b>H264_PLAY_SetStreamOpenMode</b>
设置定时器类型	<b>H264_PLAY_SetTimerType</b>
设置水印数据回调	<b>H264_PLAY_SetWaterMarkCallBack</b>

## C、打开文件或数据流

打开文件	<b>H264_PLAY_OpenFile</b>
打开流	<b>H264_PLAY_OpenStream</b> <b>H264_PLAY_OpenStreamEx</b>

## D、播放

播放开始	<b>H264_PLAY_Play</b>
------	-----------------------

## E、各功能操作及信息获取

播放声音	<b>H264_PLAY_PlaySound</b> <b>H264_PLAY_PlaySoundShare</b>
设置音量	<b>H264_PLAY_SetVolume</b>
获取音量	<b>H264_PLAY_GetVolume</b>
关闭声音	<b>H264_PLAY_StopSound</b> <b>H264_PLAY_StopSoundShare</b>
开始流数据录像	<b>H264_PLAY_StartDataRecord</b>
输入流数据	<b>H264_PLAY_InputData</b> <b>H264_PLAY_InputVideoData</b> <b>H264_PLAY_InputAudioData</b>
结束流数据录像	<b>H264_PLAY_StopDataRecord</b>
重置流缓冲回调标志	<b>H264_PLAY_ResetSourceBufFlag</b>
快放	<b>H264_PLAY_Fast</b>
慢放	<b>H264_PLAY_Slow</b>
暂停	<b>H264_PLAY_Pause</b>
按帧号定位	<b>H264_PLAY_SetCurrentFrameNum</b>
按时间定位	<b>H264_PLAY_SetPlayedTimeEx</b>
按文件偏移定位	<b>H264_PLAY_SetPlayPos</b>
刷新	<b>H264_PLAY_RefreshPlay</b> <b>H264_PLAY_RefreshPlayEx</b>
单帧播放	<b>H264_PLAY_OneByOne</b>
单帧倒放	<b>H264_PLAY_OneByOneBack</b>



抓图	<b>H264_PLAY_CatchPic</b>
清缓冲	<b>H264_PLAY_ResetBuffer</b>
颜色调整	<b>H264_PLAY_SetColor</b>
多区域显示	<b>H264_PLAY_SetDisplayRegion</b>
显示模式设置	<b>H264_PLAY_SetDisplayType</b>
调整图象播放的流畅性	<b>H264_PLAY_AdjustFluency</b>
改变图象播放的帧率	<b>H264_PLAY_ChangeRate</b>
打开音频采集功能	<b>H264_PLAY_StartAudioCapture</b>
关闭音频采集功能	<b>H264_PLAY_StopAudioCapture</b>
获取颜色参数	<b>H264_PLAY_GetColor</b>
获取关键色	<b>H264_PLAY_GetColorKey</b>
获取当前帧数	<b>H264_PLAY_GetCurrentFrameNum</b>
获取当前帧率	<b>H264_PLAY_GetCurrentFrameRate</b>
获取缓冲帧数	<b>H264_PLAY_GetDisplayBuf</b>
获取显示类型	<b>H264_PLAY_GetDisplayType</b>
获取当前时间	<b>H264_PLAY_GetPlayedTime</b> <b>H264_PLAY_GetPlayedTimeEx</b>
获取当前信息状态	<b>H264_PLAY_QueryInfo</b>
获取总帧数	<b>H264_PLAY_GetFileTotalFrames</b>
获取总时间	<b>H264_PLAY_GetFileTime</b>
获取文件偏移	<b>H264_PLAY_GetPlayPos</b>
获取索引信息	<b>H264_PLAY_GetRefValue</b>
获取流剩余缓冲	<b>H264_PLAY_GetSourceBufferRemain</b>
获取流打开模式	<b>H264_PLAY_GetStreamOpenMode</b>
获取关键帧信息	<b>H264_PLAY_GetKeyFramePos</b> <b>H264_PLAY_GetNextKeyFramePos</b>
获取 OVERLAY 模式	<b>H264_PLAY_GetOverlayMode</b>
获取图像大小	<b>H264_PLAY_GetPictureSize</b>
获取已播放的帧数	<b>H264_PLAY_GetPlayedFrames</b>

## F、 停止

播放停止	<b>H264_PLAY_Stop</b>
------	-----------------------

## G、 结束

关闭文件	<b>H264_PLAY_CloseFile</b>
关闭流	<b>H264_PLAY_CloseStream</b>
释放多显卡资源	<b>H264_PLAY_ReleaseDDrawDevice</b>

## H、 可随时调用的函数

获取系统功能	<b>H264_PLAY_GetCaps</b>
转 BMP 文件	<b>H264_PLAY_ConvertToBmpFile</b>
获取文件头长度	<b>H264_PLAY_GetFileHeadLength</b>
获取版本信息	<b>H264_PLAY_GetSdkVersion</b>

## 1.5 编程补充说明

- 步骤 A 中的操作用于支持多显卡，目前播放库暂不支持，采用默认的显卡。
- 步骤 B 中的功能设置操作只能在 **H264\_PLAY\_Play** 之前设置一次。如果用户不调用这些函数，播放库会采用默认值，如播放缓冲帧数默认为 15 帧，默认用 **overlay** 显示，默认流打开模式为最实时模式。
- 要使播放 SDK 能正常工作，步骤 C 和步骤 D 是必须的

大部分操作集中在步骤 E 中，即在 **H264\_PLAY\_Play** 和 **H264\_PLAY\_Stop** 之间调用。但其中播放声音的函数 **H264\_PLAY\_PlaySound**、**H264\_PLAY\_PlaySoundShare** 可以在 **H264\_PLAY\_Play** 之前调用，以防止部分声音数据不能播放，这在播放一个纯音频文件时较为明显，对 D、E、F、G、H 中的功能大部分之间没有调用先后顺序，当然对同一个功能还是应该按先打开再操作最后关闭的顺序。

## 2 数据常量定义

### 2.1 最大通道数

常量名	常量值	描述
FUNC_MAX_PORT	500	最大播放通道

### 2.2 声音波形范围

常量名	常量值	描述
MIN_WAVE_COEF	-100	最小波形调整因子
MAX_WAVE_COEF	100	最大波形调整因子

### 2.3 定时器类型

常量名	常量值	描述
TIMER_1	1	默认定时器，精确定时，但一个进程中只有 16 个
TIMER_2	2	不精确定时、个数不限制

### 2.4 缓冲类型

常量名	常量值	描述
BUF_VIDEO_SRC	1	视频数据源缓冲区，缓冲解码之前视频数据，只对流模式有效，单位 byte
BUF_AUDIO_SRC	2	音频数据源缓冲区，缓冲解码之前音频数据，只对流模式有效，单位 byte
BUF_VIDEO_RENDER	3	解码后视频数据缓冲区，单位帧数
BUF_AUDIO_RENDER	4	解码后音频数据缓冲区，单位帧数，音频 40ms 数据定为一帧

## 2.5 错误类型

常量名	常量值	描述
H264_PLAY_NOERROR	0	没有错误
H264_PLAY_PARA_OVER	1	输入参数非法
H264_PLAY_ORDER_ERROR	2	调用顺序不对
H264_PLAY_TIMER_ERROR	3	多媒体时钟设置失败
H264_PLAY_DEC_VIDEO_ERROR	4	视频解码失败
H264_PLAY_DEC_AUDIO_ERROR	5	音频解码失败
H264_PLAY_ALLOC_MEMORY_ERROR	6	分配内存失败
H264_PLAY_OPEN_FILE_ERROR	7	文件操作失败
H264_PLAY_CREATE_OBJ_ERROR	8	创建线程事件等失败
H264_PLAY_CREATE_DDRAW_ERROR	9	创建 directDraw 失败
H264_PLAY_CREATE_OFFSCREEN_ERROR	10	创建后端缓存失败
H264_PLAY_BUF_OVER	11	缓冲区满, 输入流失败
H264_PLAY_CREATE_SOUND_ERROR	12	创建音频设备失败
H264_PLAY_SET_VOLUME_ERROR	13	设置音量失败
H264_PLAY_SUPPORT_FILE_ONLY	14	只能在播放文件时才能使用
H264_PLAY_SUPPORT_STREAM_ONLY	15	只能在播放流时才能使用
H264_PLAY_SYS_NOT_SUPPORT	16	系统不支持, 解码器只能工作在 Pentium 3 以上
H264_PLAY_FILEHEADER_UNKNOWN	17	没有文件头
H264_PLAY_VERSION_INCORRECT	18	解码器和编码器版本不对应
H264_PLAY_INIT_DECODER_ERROR	19	初始化解码器失败
H264_PLAY_CHECK_FILE_ERROR	20	文件太短或码流无法识别
H264_PLAY_INIT_TIMER_ERROR	21	初始化多媒体时钟失败
H264_PLAY_BLT_ERROR	22	位拷贝失败
H264_PLAY_UPDATE_ERROR	23	显示 overlay 失败

## 2.6 最大区域显示数

常量名	常量值	描述
MAX_DISPLY_WND	4	同时最多打开 4 个区域显示窗口

## 2.7 显示类型

常量名	常量值	描述
DISPLAY_NORMAL	1	以正常分辨率显示
DISPLAY_QUARTER	2	以四分之一分辨率显示

## 2.8 解码缓冲数

常量名	常量值	描述
MAX_DIS_FRAMES	50	最大解码缓冲帧数
MIN_DIS_FRAMES	6	最小解码缓冲帧数

## 2.9 定位类型

常量名	常量值	描述
BY_FRAMENUM	1	按帧号
BY_FRAME_TIME	2	按时间

## 2.10 数据流原始缓冲大小

常量名	常量值	描述
SOURCE_BUF_MAX	1024*100000	最大原始缓冲
SOURCE_BUF_MIN	1024*50	最小原始缓冲

## 2.11 数据流播放模式

常量名	常量值	描述
STREAM_REALTIME	0	数据流模式
STREAM_FILE	1	文件模式

## 2.12 解码回调音频帧类型

常量名	常量值	描述
-----	-----	----

T_AUDIO16	101	
T_AUDIO8	100	

## 2.13 解码回调视频帧类型

常量名	常量值	描述
T_UYVY	1	
T_YV12	3	
T_RGB32	7	

## 2.14 系统功能

常量名	常量值	描述
SUPPORT_DDRAW	1	支持 DIRECTDRAW; 如果不支持, 则播放器不能工作
SUPPORT_BLT	3	显卡支持 BLT 操作; 如果不支持, 则播放器不能工作
SUPPORT_BLTFOURCC	4	显卡 BLT 支持颜色转换
SUPPORT_BLTSHRINKX	8	显卡 BLT 支持 X 轴缩小
SUPPORT_BLTSHRINKY	16	显卡 BLT 支持 Y 轴缩小
SUPPORT_BLTSTRETCHX	32	显卡 BLT 支持 X 轴放大
SUPPORT_BLTSTRETCHY	64	显卡 BLT 支持 Y 轴放大
SUPPORT_SSE	128	CPU 支持 SSE 指令, Intel Pentium3 以上支持 SSE 指令
SUPPORT_MMX	256	CPU 支持 MMX 指令集

## 2.15 对讲端口

常量名	常量值	描述
TALK_PORT	499	音频数据为 G711A
TALK_PCM_PORT	500	音频数据为 PCM

### 3 数据结构定义

#### 3.1 录像文件类型

```
typedef enum _MEDIA_FILE_TYPE
{
    MEDIA_FILE_NONE = 0,        //
    MEDIA_FILE_H264 = 1,        //H264 文件
    MEDIA_FILE_AVI = 2,         //AVI 文件
    MEDIA_FILE_RMVB = 3,        //暂时不支持
    MEDIA_FILE_MPG4 = 4,        //暂时不支持
    MEDIA_FILE_NUM
}MEDIA_FILE_TYPE;
```

#### 3.2 本地抓图类型

```
typedef enum _PICTURE_FILE_TYPE
{
    PICTURE_FILE_BMP = 0,        //图片格式 BMP
    PICTURE_FILE_JPG = 1         //图片格式 JPG
}PICTURE_FILE_TYPE;
```

#### 3.3 帧位置

```
typedef struct{
    long nFilePos;                //指定帧在文件中的偏移位置
    long nFrameNum;               //帧序号
    long nFrameTime;              //帧时间
    long nErrorFrameNum;          //错误帧号
    SYSTEMTIME *pErrorTime;       //错误帧时间
    long nErrorLostFrameNum;      //错误帧帧号
    long nErrorFrameSize;         //错误帧大小
}FRAME_POS,*PFRAME_POS;
```

#### 3.4 帧信息

```
typedef struct{
    long nWidth;                  //画面宽,单位像素。如果是音频数据则为 0;
```

```
long nHeight;                //画面高。如果是音频数据则为 0；
long nStamp;                 //时标信息，单位毫秒。
long nType;                  //数据类型，T_AUDIO16，T_RGB32，T_YV12，详见宏定义说明。
long nFrameRate;             //编码时产生的图像帧率。
}FRAME_INFO;
```

### 3.5 帧类型

```
typedef struct {
    char *pDataBuf;           //帧数据
    long nSize;               //帧大小
    long nFrameNum;           //帧序号
    BOOL bIsAudio;            //是否音频帧
    long nReserved;           //保留字
}FRAME_TYPE;
```

### 3.6 码流类型

```
enum ENCODE_TYPE{
    StreamTypeEmpty = 0,    //
    StreamTypeH264 = 2,     //H264
    StreamTypeJpeg = 3,     //JPEG
    StreamTypeGeneral = 4,  //General
    StreamTypePCM8 = 7,     // PCM8
    StreamTypeStd = 8       //Std
};
```

### 3.7 图像帧类型

```
enum VIDEO_FRAME_TYPE{
    FrameTypeUnknown = 0, //未知
    FrameTypeVideo = 1,   //视频
    FrameTypeAudio = 2,   //音频
    FrameTypeData = 3,    //数据
};
```



### 3.8 图像帧子类型

```
enum VIDEO_FRAME_SUB_TYPE{
    FrameSubTypeIFrame = 0,
    FrameSubTypePFrame = 1,
    FrameSubTypeBFrame = 2,
    FrameSubTypeSFrame = 3,
    FrameSubTypeDataText = 5,
    FrameSubTypeDataIntl = 6,
};
```

### 3.9 帧数据信息

```
typedef struct
{
    unsigned char *pHeader;           ///< 帧数据缓冲起始地址
    unsigned char *pContent;          ///< 帧数据缓冲地址（不包含私有头）
    unsigned long nLength;             ///< pHeader 长度
    unsigned long nFrameLength;       ///< pContent 长度

    unsigned int nType;                ///< 见 VIDEO_FRAME_TYPE
    unsigned int nSubType;             ///< 见 VIDEO_FRAME_SUB_TYPE

    unsigned int nEncodeType;          ///< 见 ENCODE_TYPE

    unsigned long nYear;               ///< 帧时间戳
    unsigned long nMonth;
    unsigned long nDay;
    unsigned long nHour;
    unsigned long nMinute;
    unsigned long nSecond;
    unsigned long nTimeStamp;

    unsigned int nFrameRate;           ///< 视频图像帧率
    int nWidth;                       ///< 视频图像大小，宽度和高度
    int nHeight;
    unsigned long nRequence;

    unsigned int nChannels;
```

```
    unsigned int nBitsPerSample;        ///< 音频采样位深
    unsigned int nSamplesPerSecond;      ///< 音频采样率

    unsigned long nParam1;              // 扩展用
    unsigned long nParam2;              // 扩展用
} VIDEO_FRAME_INFO;
```

### 3.10 字体类型

```
typedef enum
{
    OSD_TXT_FONT_ARIAL = 1,
    OSD_TXT_FONT_SERIF,
    OSD_TXT_FONT_SANS,

    OSD_TXT_FONT_SIMSUN = 101,  // 中文字体, 宋体
    OSD_TXT_FONT_SIMHEI

} DEFAULT_OSD_TXT_FONT;
```

### 3.11 OSD 信息

```
typedef struct
{
    int pos_x;                //相对位置(百分比)
    int pos_y;
    COLORREF color;          //字体颜色
    char text[256];
    DEFAULT_OSD_TXT_FONT font_type;
    int font_size;           //以窗口的百分比来表示

    HFONT out_ft;            //创建的字体对象

    int isBold;               //是否粗体
    int isTransparent;        //是否透明底色
    COLORREF bkColor;         //背景色
} OSD_INFO_TXT;
```

### 3.12 媒体数据类型（标准）

```
typedef enum
{
    STD_FILE_HEAD = 0,           // 文件头
    STD_VIDEO_I_FRAME = 1,       // 视频 I 帧
    STD_VIDEO_B_FRAME = 2,       // 视频 B 帧
    STD_VIDEO_P_FRAME = 3,       // 视频 P 帧
    STD_VIDEO_BP_FRAME = 4,      // 视频 BP 帧
    STD_VIDEO_BBP_FRAME = 5,     // 视频 B 帧 B 帧 P 帧
    STD_VIDEO_J_FRAME = 6,       // 图片帧
    STD_AUDIO_PACKET = 10,       // 音频包
}MEDIA_DATA_TYPE;
```

### 3.13 标准 H264 结构

```
typedef struct
{
    int      nPacketType;         // 包类型, MEDIA_DATA_TYPE
    char*    pPacketBuffer;      // 缓存区地址
    unsigned int dwPacketSize;    // 包的大小

    // 绝对时标
    int      nYear;              // 时标:年
    int      nMonth;             // 时标:月
    int      nDay;               // 时标:日
    int      nHour;              // 时标:时
    int      nMinute;            // 时标:分
    int      nSecond;            // 时标:秒
    unsigned int dwTimeStamp;     // 相对时标低位, 单位为毫秒
    unsigned int dwFrameNum;      // 帧序号
    unsigned int dwFrameRate;     // 帧率
    unsigned short uWidth;        // 图像宽度
    unsigned short uHeight;       // 图像高度
    unsigned int nAudioEncodeType; // 音频编码类型 IMA 9 PCM8_VWIS 12
    MS_ADPCM 13 G711A 14
    unsigned int nBitsPerSample;  // 音频采样位深
    unsigned int nSamplesPerSecond; // 音频采样率
    unsigned int Reserved[6];     // 保留
```

```
} STDH264_PACKET_INFO;
```

3.14 回调函数

3.14.1 DecodeCallBack

```
typedef void (CALLBACK* DecodeCallBack)(  
    LONG nPort,  
    PBYTE pBuf,  
    LONG nSize,  
    FRAME_INFO * pFrameInfo,  
    LONG nPlayedFrames);
```

【目的】解码回调

【描述】解码回调,建议不要在回调函数里面做长时间的逻辑处理,以免增加显示的延时

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	播放通道号
pBuf	-	-	输出	解码后的音视频数据
nSize			输出	数据长度
pFrameInfo	-	-	输出	帧信息,详见FRAME_INFO结构体
nPlayedFrames	-	-	输出	帧计数

3.14.2 DisplayCallBack

```
typedef void (CALLBACK* DisplayCallBack)(  
    LONG nPort,  
    PBYTE *pBuf,  
    LONG nSize,  
    LONG nWidth,  
    LONG nHeight,  
    LONG nStamp,  
    LONG nType,
```

LONG nUser);

【目的】抓图回调函数（显示回调，无音频）

【描述】抓图回调函数,任何时候调用。设置成功一直有效，不需要设置为 NULL，

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
pBuf	-	-	输出	图像数据
nSize			输出	图像数据长度
nWidth	-	-	输出	画面宽度（像素）
nHeight	-	-	输出	画面高度
nStamp				时间标 （毫秒）
nType				类型 T_YV12
nUser				用户数据

### 3.14.3 SourceBufCallBack

```
typedef void (CALLBACK * SourceBufCallBack)(  
    LONG nPort,  
    DWORD nBufSize,  
    LONG nUser);
```

【目的】源缓冲区阈值回调

【描述】只有在数据量小于等于指定阈值的时候，才会触发回调。一次触发后需要调用 H264\_PLAY\_ResetSourceBufFlag 接口重置参数，使再次有效。

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
nBufSize	-	-	输出	缓冲中数据大小
nUser			输出	用户数据

### 3.14.4 FileCallBack

```
typedef void (CALLBACK * FileCallBack)(DWORD nPort, LONG nUser);
```

【目的】文件播放结束回调或者建立文件索引成功，则回调

【描述】文件播放结束回调或者建立文件索引成功，则回调。

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
nUser			输出	用户数据

### 3.14.5 DrawCallBack

```
typedef void (CALLBACK* DrawCallBack)(LONG nPort,HDC hDc,LONG nUser);
```

【目的】注册画图板回调

【描述】获得当前表面的 device context, 可以在这个 DC 上画图（或写字），就好像在窗口的客户区 DC 上绘图，但这个 DC 不是窗口客户区的 DC，而是 DirectDraw 里的 Off-Screen 表面的 DC。

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
HDC				表面设备上下文，可以像操作显示窗口客户区 DC 那样操作
nUser			输出	用户数据

【注意】如果是使用 overlay 表面，这个接口无效，可以直接在窗口上绘图，只要不是透明色就不会被覆盖。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

### 3.14.6 VerifyCallBack

```
typedef void (CALLBACK * VerifyCallBack)(  
    LONG nPort,  
    FRAME_POS * pFilePos,  
    DWORD bIsVideo,  
    LONG nUser);
```

【目的】水印验证回调

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
pFilePos				帧信息
bIsVideo				是否是视频
nUser			输出	用户数据

### 3.14.6 AudioCallBack

```
typedef void (CALLBACK * AudioCallBack)(
```

```
    LONG nPort,  
    PBYTE pAudioBuf,  
    LONG nSize,  
    LONG nStamp,  
    LONG nType,  
    LONG nUser);
```

【目的】解码音频回调

【描述】解码音频回调

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	播放通道号
pAudioBuf	-	-	输出	wave 格式音频数据
nSize			输出	音频长度
nStamp	-	-	输出	时标 (ms)
nType	-	-	输出	音频类型 T_AUDIO16, 采样率 8000, 单声道, 每个采样点道, 每个采样点道, 每个采样点 16 位表示
nUser				用户数据

### 3.14.7 ImageSizeChangeCallBack

```
typedef void(CALLBACK * ImageSizeChangeCallBack)(LONG nPort, LONG nUser);
```

【目的】图像分辨率改变通知回调

【描述】设置解码时图象格式发生改变通知用户的回调函数；打开文件前使用

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
nUser			输出	用户数据

### 3.14.8 DemuxCallBack

```
typedef void (CALLBACK* DemuxCallBack)(  
    LONG nPort,  
    VIDEO_FRAME_INFO *pFrameInfo,  
    LONG nUser);
```

【目的】源数据分析完的数据回调

【描述】码流分析后的一帧数据

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输出	通道号
pFrameInfo				帧信息
nUser			输出	用户数据

### 3.14.10 AudioCaptureCallBack

```
typedef void (CALLBACK *AudioCaptureCallBack)(  
    LPBYTE pDataBuffer,  
    DWORD DataLength,  
    LONG nUser);
```

【目的】音频采集回调

【描述】对讲采集电脑音频

【参数】

参数	成员	取值范围	输入/输出	描述
----	----	------	-------	----



pDataBuffer	-		输出	回调数据指针
DataLength				回调数据长度
nUser			输出	用户数据

### 3.14.11 InfoFramCallBack

```
typedef void (CALLBACK *InfoFramCallBack)(
    LONG nPort,
    LONG nType, (0x03 代表 gprs 信息帧)
    LPCSTR pBuf,
    LONG nSize,
    LONG nUser);
```

【目的】源数据分析完的数据回调

【描述】码流分析后的一帧数据

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-		输出	端口
nType				类型
pBuf				数据指针
nSize			输出	数据长度
nUser				用户数据

### 3.14.12 InfoFramCallBack

```
typedef void (CALLBACK *CoverFileCallBack)(
    DWORD CurrentPos,
    DWORD TotoalPos,
    long dwUser);
```

【目的】H264 转换 AVI 转换进度

【描述】H264 转换 AVI 转换进度

【参数】

参数	成员	取值范围	输入/输出	描述
CurrentPos	-		输出	当前大小
TotoalPos				总的大小
nUser				用户数据

### 3.14.12 Watermark\_callback

```
typedef int (__stdcall* Watermark_callback)(  
    char* buf,  
    long key,  
    long len,  
    long reallen,  
    long reserved,  
    long nUser);
```

【目的】水印数据回调

【描述】水印数据回调

【参数】

参数	成员	取值范围	输入/输出	描述
buf	-		输出	水印数据 buffer 指针
Key				区分不同水印信息
len				缓冲的最大长度
reallen				缓冲的实际长度
reserved				保留
nUser				用户数据

## API 函数说明

### 3.15 打开播放文件 H264\_PLAY\_OpenFile

【目的】打开播放文件

【语法】BOOL H264\_PLAY\_OpenFile(LONG nPort, LPCSTR sFileName)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
sFileName	-	-	输入	播放文件名

【返回值】

返回值	宏定义	描述
1	TRUE	打开播放文件成功
0	FALSE	打开播放文件失败

【注意】

文件不能大于 4G Byte 或者小于 4K Byte

### 3.16 开始播放 H264\_PLAY\_Play

【目的】开始播放

【语法】BOOL H264\_PLAY\_Play(LONG nPort, HWND hWnd)

【描述】开始播放文件，如果已经在播放的情况下调用该接口则改变当前播放速度为正常播放速度。

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
hWnd	-	-	输入	播放窗口句柄

【返回值】

返回值	宏定义	描述
1	TRUE	开始播放成功
0	FALSE	开始播放失败

**【注意】**

无。

**3.17 停止播放 H264\_PLAY\_Stop**

**【目的】** 停止播放

**【语法】** BOOL H264\_PLAY\_Stop(LONG nPort)

**【描述】** 停止播放。

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
1	TRUE	停止播放成功
0	FALSE	停止播放失败

**【注意】**

无。

**3.18 关闭播放文件 H264\_PLAY\_CloseFile**

**【目的】** 关闭播放文件

**【语法】** BOOL H264\_PLAY\_CloseFile(LONG nPort)

**【描述】**

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
1	TRUE	关闭播放文件成功
0	FALSE	关闭播放文件失败

**【注意】**

无。

### 3.19 打开媒体流 H264\_PLAY\_OpenStream

【目的】打开媒体流

【语法】**BOOL** H264\_PLAY\_OpenStream(**LONG** nPort,  
PBYTE pFileHeadBuf,  
DWORD nSize,  
DWORD nBufPoolSize)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pFileHeadBuf	-	NULL	输入	目前不使用
nSize	-	0	输入	目前不使用
nBufPoolSize	-	[SOURCE_BUF_MIN, SOURCE_BUF_MAX]	输入	设置播放器中存放数据流的缓冲区大小。一般设为 900K，如果数据送过来相对均匀，可调小该值，如果数据传输不均匀，可增大该值

【返回值】

返回值	宏定义	描述
1	TRUE	打开媒体流成功
0	FALSE	打开媒体流失败

【注意】

pFileHeadBuf和nSize目前不使用。

### 3.20 输入待解码数据 H264\_PLAY\_InputData

【目的】输入待解码数据

【语法】BOOL H264\_PLAY\_InputData(

LONG nPort,

PBYTE pBuf,

DWORD nSize)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pBuf	-	-	输入	缓冲区地址
nSize	-	-	输入	缓冲区长度

【返回值】

返回值	宏定义	描述
1	TRUE	输入数据成功
0	FALSE	输入数据失败，一般是缓冲区以满

【注意】

打开流并调用 H264\_PLAY\_Play 之后才能输入数据。

### 3.21 关闭数据流 H264\_PLAY\_CloseStream

【目的】关闭数据流

【语法】BOOL H264\_PLAY\_CloseStream(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
1	TRUE	关闭数据流成功

0	FALSE	关闭数据流失败
---	-------	---------

**【注意】**

无。

### 3.22 输入视频数据 H264\_PLAY\_InputVideoData

**【目的】** 输入视频数据

**【语法】** BOOL H264\_PLAY\_InputVideoData(LONG nPort,  
PBYTE pBuf,  
DWORD nSize)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pBuf	-	-	输入	缓冲区地址
nSize	-	-	输入	缓冲区长度

**【返回值】**

返回值	宏定义	描述
1	TRUE	输入数据成功
0	FALSE	输入数据失败，一般是缓冲区已满

**【注意】**

打开流并调用 H264\_PLAY\_Play 之后才能输入数据。

### 3.23 输入音频数据 H264\_PLAY\_InputAudioData

**【目的】** 输入音频数据

**【语法】** BOOL H264\_PLAY\_InputAudioData(LONG nPort,  
PBYTE pBuf,  
DWORD nSize)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pBuf	-	-	输入	缓冲区地址
nSize	-	-	输入	缓冲区长度

**【返回值】**

返回值	宏定义	描述
1	TRUE	输入音频数据成功
0	FALSE	输入音频数据失败，一般是缓冲区已满

**【注意】**

打开声音后才能输入数据。

### 3.24 播放暂停/恢复 H264\_PLAY\_Pause

**【目的】** 播放暂停/恢复

**【语法】** BOOL H264\_PLAY\_Pause(LONG nPort, BOOL bPause)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
bPause	-	[0,1]	输入	暂停或恢复

**【返回值】**

返回值	宏定义	描述
1	TRUE	暂停播放成功
0	FALSE	暂停播放失败

**【注意】**

无。

### 3.25 加快播放速度 H264\_PLAY\_Fast

**【目的】** 加快播放速度

**【语法】** BOOL H264\_PLAY\_Fast(LONG nPort)



【描述】播放速度分为九级，播放速度分别为每秒播放 1,3,6,12,25,50,75,100,125 帧图像。每次调用播放速度提升一级最多调用 4 次，要恢复正常播放调用 H264\_PLAY\_Play,从当前位置开始正常播放

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
1	TRUE	加快播放成功
0	FALSE	加快播放失败

【注意】

无。

### 3.26 反向回放 H264\_PLAY\_Back

【目的】反向回放

【语法】BOOL H264\_PLAY\_Back(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

【注意】

无。

### 3.27 减慢播放速度 H264\_PLAY\_Slow

【目的】减慢播放速度

【语法】BOOL H264\_PLAY\_Slow(LONG nPort)

【描述】播放速度分为九级，播放速度分别为每秒播放 1,3,6,12,25,50,75,100,125 帧图像。每次调用播放速度降低一级最多调用 4 次，要恢复正常播放调用 H264\_PLAY\_Play,从当前位置开始正常播放。

【参数】

参数	成员	取值范围	输入/输出	描述
----	----	------	-------	----

nPort	-	[0, 100]	输入	播放通道号
-------	---	----------	----	-------

**【返回值】**

返回值	宏定义	描述
1	TRUE	减慢播放成功
0	FALSE	减慢播放失败

**【注意】**

无。

### 3.28 单帧往后播放 H264\_PLAY\_NextFrame

**【目的】** 单帧往后播放

**【语法】** BOOL H264\_PLAY\_NextFrame (LONG nPort)

**【描述】** 单帧往后播放，要恢复正常播放调用 [H264\\_PLAY\\_Play](#),从当前位置开始正常播放。

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
1	TRUE	单帧播放成功
0	FALSE	单帧播放失败

**【注意】**

无。

### 3.29 单帧往前播放 H264\_PLAY\_PrevFrame

**【目的】** 单帧往前播放

**【语法】** BOOL H264\_PLAY\_PrevFrame (LONG nPort)

**【描述】** 单帧往前播放，要恢复正常播放调用 [H264\\_PLAY\\_Play](#),从当前位置开始正常播放。

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

## 【返回值】

返回值	宏定义	描述
1	TRUE	单帧播放成功
0	FALSE	单帧播放失败

## 【注意】

无。

### 3.30 打开播放声音 H264\_PLAY\_PlaySound

## 【目的】打开播放声音

## 【语法】BOOL H264\_PLAY\_PlaySound(LONG nPort)

【描述】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。

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

## 【返回值】

返回值	宏定义	描述
1	TRUE	打开声音成功
0	FALSE	打开声音失败

## 【注意】

默认情况下声音是关闭的。

### 3.31 关闭播放声音 H264\_PLAY\_StopSound

## 【目的】关闭播放声音

## 【语法】BOOL H264\_PLAY\_StopSound(void)

## 【描述】

## 【参数】

## 【返回值】

返回值	宏定义	描述
1	TRUE	关闭声音成功
0	FALSE	关闭声音失败

**【注意】**

默认情况下声音是关闭的。

### 3.32 共享方式打开播放声音 H264\_PLAY\_PlaySoundShare

**【目的】** 以共享方式打开播放声音，播放本路声音而不去关闭其它路声音

**【语法】** BOOL H264\_PLAY\_PlaySoundShare(LONG nPort)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
1	TRUE	打开声音成功
0	FALSE	打开声音失败

**【注意】**

默认情况下声音是关闭的。

### 3.33 共享方式关闭播放声音 H264\_PLAY\_StopSoundShare

**【目的】** 以共享方式关闭播放声音。

**【语法】** BOOL H264\_PLAY\_StopSoundShare(LONG nPort)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
1	TRUE	关闭声音成功
0	FALSE	关闭声音失败

**【注意】**

以共享方式关闭声音。[H264\\_PLAY\\_PlaySound](#)和[H264\\_PLAY\\_StopSound](#)是以独占方

式播放声音的，在同一个进程中，所有通道必须使用相同的方式播放或关闭声音。

### 3.34 设置播放声音 H264\_PLAY\_SetVolume

【目的】设置播放声音。

【语法】`BOOL H264_PLAY_SetVolume(LONG nPort, WORD nVolume)`

【描述】修改的是音频本身的声音，100 为原声音大小，小于 100 变小，答应 100 增加，最大 1000

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nVolume	-	[0, 1000]	输入	音量

【返回值】

返回值	宏定义	描述
1	TRUE	设置声音成功
0	FALSE	设置声音失败

【注意】

无。

### 3.35 获取播放声音 H264\_PLAY\_GetVolume

【目的】获取播放声音。

【语法】`WORD H264_PLAY_GetVolume(LONG nPort)`

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 500]	输入	播放通道号

【返回值】

返回值	宏定义	描述
-	-	当前设置的音量值

【注意】

无。

### 3.36 调整 WAVE 波形 H264\_PLAY\_AdjustWaveAudio

【目的】调整 WAVE 波形

【语法】WORD H264\_PLAY\_AdjustWaveAudio(

LONG nPort,

LONG nCoefficient)

【描述】调整 WAVE 波形，可以改变声音的大小

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nCoefficient	-	[-100,100]	输入	调整的参数， 0 是不调整

【返回值】

返回值	宏定义	描述
-	-	当前设置的音量值

【注意】

无。

### 3.37 设置解码回调函数 H264\_PLAY\_SetDecCallBack

【目的】设置解码回调函数

【语法】BOOL H264\_PLAY\_SetDecCallBack(

LONG nPort,

[DecodeCallBack](#) pDecodeProc);

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pDecodeProc	-	-	输入	设置解码回调 函数，替换掉

				解码器的解码部分，由用户自己控制解码。
--	--	--	--	---------------------

**【返回值】**

返回值	宏定义	描述
1	TRUE	设置解码回调函数成功
0	FALSE	设置解码回调函数失败

**【注意】**

无。

### 3.38 设置解码回调的码流类型 H264\_PLAY\_SetDecCBStream

**【目的】** 设置解码回调的码流类型

**【语法】** WORD H264\_PLAY\_SetDecCBStream(LONG nPort, DWORD nStream)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nStream	-	[1, 3]	输入	1、视频流，2 音频流，3、复合流。

**【返回值】**

返回值	宏定义	描述
1	TRUE	设置解码回调码流类型成功
0	FALSE	设置解码回调码流类型失败

**【注意】**

无。

### 3.39 设置图像显示回调函数 H264\_PLAY\_SetDisplayCallBack

**【目的】** 设置图像显示回调函数

**【语法】** WORD H264\_PLAY\_SetDisplayCallback(

LONG nPort,  
[DisplayCallBack](#) pDisplayProc,  
LONG nUser)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pDisplayProc	-	-	输入	视频图像显示数据回调函数
nUser	-	-	输入	用户自定义参数

**【返回值】**

返回值	宏定义	描述
1	TRUE	设置显示回调函数成功
0	FALSE	设置显示回调函数失败

**【注意】**

无。

### 3.40 设置音频解码后的数据回调函数 H264\_PLAY\_SetAudioCallBack

**【目的】** 设置音频解码后的数据回调函数

**【语法】** WORD H264\_PLAY\_SetAudioCallBack(

LONG nPort,  
[AudioCallBack](#) pAudioProc,  
LONG nUser)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pAudioProc	-	-	输入	音频解码后的 WAVE 帧数



				据回调
nUser	-	-	输入	用户自定义参数

## 【返回值】

返回值	宏定义	描述
1	TRUE	设置音频数据回调函数成功
0	FALSE	设置音频数据回调函数失败

## 【注意】

无。

### 3.41 源数据分析完的数据回调 H264\_PLAY\_SetDemuxCallBack

【目的】源数据分析完的数据回调

【语法】BOOL H264\_PLAY\_SetDemuxCallBack (

LONG nPort,

[DemuxCallBack](#) pProc,

LONG nUser)

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pProc	-	-	输入	分析数据回调函数指针
nUser	-	-	输入	用户自定义参数

## 【返回值】

返回值	宏定义	描述
1	TRUE	设置回调函数成功
0	FALSE	设置回调函数失败

## 【注意】

无。

### 3.42 设置数据校验 H264\_PLAY\_SetVerifyCallBack

【目的】设置数据校验

【语法】WORD H264\_PLAY\_SetVerifyCallBack (  
LONG nPort,  
DWORD nBeginTime,  
DWORD nEndTime,  
[VerifyCallBack](#) pProc,  
LONG nUser)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nBeginTime	-	-	输入	开始时间
nEndTime	-	-	输入	结束时间
pProc	-	-	输入	水印回调函数指针
nUser	-	-	输入	用户自定义参数

【返回值】

返回值	宏定义	描述
1	TRUE	设置回调函数成功
0	FALSE	设置回调函数失败

【注意】

无。

### 3.43 设置水印数据回调 H264\_PLAY\_SetWaterMarkCallBack

【目的】设置水印数据回调

【语法】WORD H264\_PLAY\_SetWaterMarkCallBack (  
LONG nPort,  
[WaterMarkProc](#) pWaterMarkProc,

LONG nUser)

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pWaterMarkProc	-	-	输入	水印数据回调函数指针
nUser	-	-	输入	用户自定义参数

## 【返回值】

返回值	宏定义	描述
1	TRUE	设置回调函数成功
0	FALSE	设置回调函数失败

## 【注意】

无。

### 3.44 编码格式改变消息 H264\_PLAY\_SetImageSizeChangeMsg

【目的】设置解码时编码格式发生改变时要发送的消息

【语法】BOOL H264\_PLAY\_SetImageSizeChangeMsg (

LONG nPort,

HWND hWnd,

UINT nMsg)

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0,100]	输入	播放器通道号
hWnd	-	-	输入	消息发送窗口
nMsg	-	-	输入	用户输入的消息,当播放到文件结束

				时 用 户 在 hWnd 窗 口 过程中收到 这个消息
--	--	--	--	--------------------------------------

## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

## 【注意】

无。

### 3.45 设置文件结束时要发送的消息 H264\_PLAY\_SetFileEndMsg

【目的】设置文件结束时要发送的消息

【语法】BOOL H264\_PLAY\_SetFileEndMsg(

LONG nPort,

HWND hWnd,

UINT nMsg)

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0,100]	输入	播放器通道号
hWnd	-	-	输入	消息发送窗口
nMsg	-	-	输入	用户输入的消息,当播放到文件结束时用户在 hWnd 窗口过程中收到这个消息

## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

### 3.46 设置源缓冲区阈值及回调指 H264\_PLAY\_SetSourceBufCallBack

**【目的】** 设置源缓冲区阈值和剩余数据小于等于阈值时的回调函数指针

**【语法】** BOOL H264\_PLAY\_SetSourceBufCallBack(

LONG nPort,

DWORD nThreshold,

[SourceBufCallBack](#) pBufferProc,

LONG nUser)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nThreshold	-	-	输入	阈值
pBufferProc	-	-	输入	回调函数指针
nUser	-	-	输入	用户数据

**【返回值】**

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

### 3.47 重置回调标志为有效状态 H264\_PLAY\_ResetSourceBufFlag

**【目的】** 重置回调标志位为有效状态

【语法】BOOL H264\_PLAY\_ResetSourceBufFlag(LONG nPort)

【描述】重置回调标志位为有效状态。流模式下源缓冲到达阈值时（如果用户设置了回调函数）不一定会回调，用户需要重置回调标志位后才能回调。而且每次回调后标志位都被设为无效，用户可以在适当的时候重置回调标志，这个接口的主要目的是防止重复回调（数据在阈值附近摆动因为用户输入数据时，播放器也在读走数据）。初始化状态下，回调标志位是有效的

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

无。

### 3.48 分辨率改变回调设 H264\_PLAY\_SetImageSizeChangeCallBack

【目的】解码时图象格式发生改变通知用户的回调函数；打开文件前使用

【语法】BOOL H264\_PLAY\_SetImageSizeChangeCallBack (

LONG nPort,

[ImageSizeChangeCallBack](#) pProc,

LONG nUser)

【描述】：

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pProc	-	-	输入	回调函数
nUser	-	-	输入	用户自定义数据

【返回值】

返回值	宏定义	描述
1	TRUE	成功

0	FALSE	失败
---	-------	----

**【注意】**

无。

### 3.49 设置文件索引建立后回调 H264\_PLAY\_SetFileRefCallBack

**【目的】** 设置回调函数指针，文件索引建立后回调

**【语法】** BOOL H264\_PLAY\_SetFileRefCallBack (

LONG nPort,

[FileCallBack](#) pProc,

LONG nUser)

**【描述】** 设置回调函数指针，文件索引建立后回调。为了能在文件中准确

快速的定位，我们在文件打开时生成文件索引。这个过程耗时比较长，大约每秒处理 40M 左右的数据，主要是因为从硬盘读数据比较慢。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

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pProc	-	-	输入	回调函数
nUser	-	-	输入	用户自定义数据

**【返回值】**

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

### 3.50 设置文件结束回调 H264\_PLAY\_SetFileEndCallBack

**【目的】** 设置文件结束回调

**【语法】** BOOL H264\_PLAY\_SetFileEndCallBack (

LONG nPort,

[FileCallBack](#) pFileDoneProc,  
LONG nUser)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pFileDoneProc	-	-	输入	回调函数
nUser	-	-	输入	用户自定义数据

**【返回值】**

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

### 3.51 设置文件索引 H264\_PLAY\_SetRefValue

**【目的】** 设置文件索引**【语法】** BOOL H264\_PLAY\_SetRefValue (

LONG nPort,  
PBYTE pBuffer,  
DWORD nSize)

**【描述】** 设置文件索引。如果已经有了文件索引信息，可以不再调用生成索引的回调函数([H264\\_PLAY\\_SetFileRefCallBack](#))，直接输入索引信息

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号
pBuffer	-	-	输入	索引信息
nSize	-	-	输入	索引信息长度



## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

## 【注意】

索引信息及其长度必须准确。

### 3.52 获取文件索引信息 H264\_PLAY\_GetRefValue

【目的】获取文件索引信息

【语法】BOOL H264\_PLAY\_GetRefValue (

LONG nPort,

PBYTE pBuffer,

DWORD \*pSize)

【描述】获取文件索引。

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号
pBuffer	-	-	输出	索引信息
nSize	-	-	输出	索引信息长度

## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

## 【注意】

可以在第一次指定 pSize=0,pBuffer=NULL,从 pSize 的返回值获得需要的缓冲区大小。然后分配足够的缓冲，再调用一次。

### 3.53 设置当前播放帧号 H264\_PLAY\_SetCurrentFrameNum

【目的】设置当前播放播放位置到指定帧号

【语法】BOOL H264\_PLAY\_SetCurrentFrameNum(

LONG nPort,

DWORD nFrameNum)

【描述】设置当前播放位置到指定帧号；根据帧号来定位播放位置。此函数必须在文件索引生成之后才能调用

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

无。

### 3.54 设置当前播放时间 H264\_PLAY\_SetPlayedTimeEx

【目的】设置当前播放时间

【语法】BOOL H264\_PLAY\_SetPlayedTimeEx(

LONG nPort,

DWORD nTime)

【描述】根据时间设置文件播放位置，此接口比 H264\_PLAY\_SetPlayedPos 费时，但如果用时间来控制进度条（与 [H264\\_PLAY\\_GetPlayedTimeEx](#) 配合使用），那么可以使进度条平滑滚动

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nTime	-	-	输入	要设置的播放时间

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

**3.55 设置文件播放指针的相对位置 H264\_PLAY\_SetPlayPos**

**【目的】** 设置文件播放指针的相对位置（百分比）

**【语法】** BOOL H264\_PLAY\_SetPlayPos(LONG nPort, float fReleatePos)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
fReleatePos	-	[0.0, 100.0]	输入	文件播放相对位置

**【返回值】**

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

无。

3.56 获取文件播放指针的相对位置 H264\_PLAY\_GetPlayPos

【目的】返回文件播放指针的相对位置（百分比）

【语法】float H264\_PLAY\_GetPlayPos(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】返回文件播放指针的相对位置

【注意】

无。

3.57 设置图象的视频参数 H264\_PLAY\_SetColor

【目的】设置图象的视频参数，即时起作用

【语法】BOOL H264\_PLAY\_SetColor(  
LONG nPort,  
DWORD nRegionNum,  
LONG nBrightness,  
LONG nContrast,  
LONG nSaturation,  
LONG nHue)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nRegionNum	-	[0, 128]	输入	显示区域，参考 H264_PLAY_SetDisplayRegion;

				如果只有一个显示区域(通常情况)设为 0
nBrightness	-	[0, 128]	输入	亮度, 默认 64
nContrast	-	[0, 128]	输入	对比度, 默认 64
nSaturation	-	[0, 128]	输入	饱和度, 默认 64
nHue	-	[0, 128]	输入	色调, 默认 64

## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

## 【注意】

无。

### 3.58 设置播放缓冲区大小 H264\_PLAY\_SetDisplayBuf

【目的】设置播放缓冲区大小

【语法】BOOL H264\_PLAY\_SetDisplayBuf (LONG nPort, DWORD nNum)

【描述】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，他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 10(帧)，如果用户追求最大延时最小，可以考虑试当减小这个值

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nNum	-	-		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
--	--	--	--	-------------------

**【返回值】**

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

### 3.59 设置图象的视频参数 H264\_PLAY\_SetDisplayType

**【目的】** 设置图象的视频参数，即时起作用

**【语法】** BOOL H264\_PLAY\_SetDisplayType(LONG nPort, LONG nType);

**【描述】** 设置显示的模式，在小画面显示时，采用 DISPLAY\_QUARTER 可以减小显卡工作量，从而支持更多路显示，但画面显示质量有下降。在正常和大画面显示时应该使用 DISPLAY\_NORMAL

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nType	-	-		DISPLAY_NORMAL 或 DISPLAY_QUARTER

**【返回值】**

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

**【注意】**

无。

### 3.60 设置图象的流畅性 H264\_PLAY\_AdjustFluency

**【目的】** 设置图象的流畅性参数，即时起作用

**【语法】** BOOL H264\_PLAY\_AdjustFluency (LONG nPort, int nLevel);

**【描述】** 调整图象播放的流畅性。流畅性和实时性是一对平衡体。当 level 为 0 时，图象最流畅；当 level 为 6 时，图象最实时。Level 的默认值为 3

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nLevel	-	-	输入	

## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

## 【注意】

无。

### 3.61 设置图象播放的帧率 H264\_PLAY\_SetPlayFPS

【目的】改变图象播放的帧率

【语法】BOOL H264\_PLAY\_SetPlayFPS (LONG nPort, LONG nFPS);

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nFPS	-	-	输入	

## 【返回值】

## 【注意】

无。

### 3.62 打开音频采集功能 H264\_PLAY\_StartAudioCapture

【目的】打开音频采集功能

【语法】BOOL H264\_PLAY\_StartAudioCapture(  
AudioCaptureCallBack pProc,  
LONG nBitsPerSample,  
LONG nSamplesPerSecond,  
LONG nLength,

LONG nUser);

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
pProc	-	-	-	音频采集回调函数，当采集到音频数据时该函数将被调用
nBitsPerSample	-	-	输入	采样位深度
nSamplePerSecond	-	-	输入	采样率
nLength	-	-	输入	音频数据包长度
nUser	-	-	输入	自定义参数

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

无。

### 3.63 H264\_PLAY\_StopAudioCapture

【目的】关闭音频采集功能

【语法】BOOL H264\_PLAY\_StopAudioCapture (void)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
	-	-		

【返回值】

返回值	宏定义	描述
1	TRUE	成功



0	FALSE	失败
---	-------	----

**【注意】**

无。

### 3.64 设置 Overlay 模式 H264\_PLAY\_SetOverlayMode

**【目的】** 设置 Overlay 模式

**【语法】** BOOL H264\_PLAY\_SetOverlayMode(

LONG nPort,

BOOL bOverlay,

COLORREF colorKey)

**【描述】** 设置 OVERLAY 模式显示画面。在一块显卡中同一时刻只能有一个 OVERLAY 表面处于活动状态，如果此时系统中已经有程序使用了 OVERLAY，那么播放器就不能再创建 OVERLAY 表面，它将自动改用 Off\_Screen 表面，并不返回 FALSE。一些常用的播放器，以及我们卡的预览都可能使用了 OVERLAY 表面。同样，如果播放器使用了 OVERLAY 表面，那么，其他的程序将不能使用 OVERLAY 表面，特别注意，我们的卡在预览时可能也要使用 OVERLAY (用户可设置)，如果先打开播放器（并且使用了 OVERLAY），再启动预览，那么预览可能因为得不到 OVERLAY 而失败。使用 OVERLAY 模式的优点是：大部份的显卡都支持 OVERLAY，在一些不支持 BLT 硬件缩放和颜色转换的显卡上 (如 SIS 系列显卡)使用 OVERLAY 模式 (OVERLAY 模式下的缩放和颜色转换由显卡支持)，可以大大减小 CPU 利用率并提高画面质量 (相对于软件缩放和颜色转换)。缺点是：只能有一路播放器使用。该设置必须在 H264\_PLAY 之前使用，而且需要设置透明色。

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort		[0, 100]	输入	播放通道号
bOverlay	-	-	输入	<i>TRUE</i> 表示将首先尝试使用 OVERLAY 模式，如果不行再使用其他模式 <i>FALSE</i> 不进行 OVERLAY 模式的尝试
colorKey	-	-	输入	用户设置的锁结色，透明色相当于一层透明膜，显示的画面只

				能穿过这种颜色，而其他的颜色将挡住显示的画面。用户应该在显示窗口中涂上这种颜色，那样才能看到显示画面。一般应该使用一种不常用的颜色作为透明色。这是一个双字节值 0x00rrggbb,最高字节为 0,后三个字节分别表示 r,g,b 的值
--	--	--	--	---

【返回值】

【注意】

无。

3.65 设置图像质量 H264\_PLAY\_SetPicQuality

【目的】设置图像质量

【语法】BOOL H264\_PLAY\_SetPicQuality(LONG nPort,BOOL bHighQuality)

【描述】设置图像质量，当设置成高质量时画面效果好，但 CPU 利用率高。在支持多路播放时，可以设为低质量，以降低 CPU 利用率；当某路放大播放时将该路设置成高质量，以达到好的画面效果

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道
bHighQuality	-	[0, 1]	输入	解码图像质量

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

无。

### 3.66 设置播放模式 H264\_PLAY\_SetStreamOpenMode

【目的】设置播放模式

【语法】`BOOL H264_PLAY_SetStreamOpenMode(LONG nPort, DWORD nMode)`

【描述】设置流播放的模式。必须在播放之前设置

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道
nMode	-	[0, 1]	输入	<a href="#">播放流模式</a> STREAME_REALTIME 实时模式（默认） STREAME_FILE 文件 模式 实时模式，适合播放网 络实时数据，解码器会 立刻解码； 文件模式，适合用户把 文件数据用流方式输 入；

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

当 H264\_PLAY\_InputData() 返回 FALSE 时，用户要等一下重新输入。

### 3.67 设置播放使用的定时器类型 H264\_PLAY\_SetTimerType

【目的】设置定时器类型

【语法】`BOOL H264_PLAY_SetTimerType(  
LONG nPort,  
DWORD nTimerType)`

【描述】设置当前通道使用的定时器类型，注意：必须在 Open 之前调用

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道
nTimerType	-	-	输入	TIMER_1 多媒体定时器，精度高，但不能超过16个； TIMER_2 线程定时器，精度略低，无数量限制

## 【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

## 【注意】

无。

### 3.68 获取当前系统信息 H264\_PLAY\_GetCaps

【目的】获取当前系统信息

【语法】DWORD H264\_PLAY\_GetCaps(void)

【描述】测试播放器需要的一些系统功能

## 【参数】

参数	成员	取值范围	输入/输出	描述

## 【返回值】

返回值	宏定义	描述
-	-	<a href="#">系统功能位</a>

## 【注意】

无。

### 3.69 获得颜色值 H264\_PLAY\_GetColor

【目的】获得相应的颜色值

【语法】BOOL H264\_PLAY\_GetColor(

```
    LONG nPort,  
    DWORD nRegionNum,  
    LONG *pBrightness,  
    LONG *pContrast,  
    LONG *pSaturaation,  
    LONG *pHue)
```

【描述】

【参数】

参数	成员	取值范围	输入 / 输出	描述
nPort	-	[0, 100]	输入	
nRegionNum	-	-	输入	显示区域，参考 PLAY_SetDisplayRegion； 如果只有一个显示区域(通常情况)设为 0
pBrightness	-	[0,128]	输出	亮度
pContrast	-	[0,128]	输出	对比度
pSaturaation	-	[0,128]	输出	饱和度
pHue	-	[0,128]	输出	色调

【返回值】

返回值	宏定义	描述
1	TRUE	成功
0	FALSE	失败

【注意】

无。

### 3.70 获得播放缓冲的帧数 H264\_PLAY\_GetDisplayBuf

【目的】获得播放缓冲区最大缓冲的帧数

【语法】int H264\_PLAY\_GetDisplayBuf(LONG nPort)

【描述】测试播放器需要的一些系统功能

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】所获得的播放缓冲区最大缓冲帧数

【注意】

无。

### 3.71 获得目前设置的显示模式 H264\_PLAY\_GetDisplayType

【目的】获得目前设置的显示模式

【语法】int H264\_PLAY\_GetDisplayType(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
1	<a href="#"><u>DISPLAY_NORMAL</u></a>	正常显示模式
2	<a href="#"><u>DISPLAY_QUARTER</u></a>	1/4 大小显示模式

【注意】

无。

### 3.72 获得 overlay 层使用的锁结色 H264\_PLAY\_GetColorKey

【目的】获得 overlay 层使用的锁结色

【语法】COLORREF H264\_PLAY\_GetColorKey(LONG nPort)

【描述】获得 overlay 层使用的锁结色

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号

## 【返回值】

返回值	宏定义	描述
-	-	锁结色的颜色值

## 【注意】

无。

### 3.73 获得当前使用的 overlay 模式 H264\_PLAY\_GetOverlayMode

【目的】获得当前使用的 overlay 模式

【语法】LONG H264\_PLAY\_GetOverlayMode(LONG nPort)

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号

## 【返回值】

返回值	宏定义	描述
0	-	没有使用 overlay 模式
1	-	使用 overlay 模式

## 【注意】

无。

### 3.74 获得当前设置的图像质量 H264\_PLAY\_GetPictureQuality

【目的】获得当前设置的图像质量

【语法】BOOL H264\_PLAY\_GetPictureQuality(LONG nPort, BOOL \*pHighQuality)

## 【描述】

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pHighQuality	-	[0, 1]	输出	当前设置的图像质量

## 【返回值】

返回值	宏定义	描述
0	-	获取失败
1	-	获取成功

## 【注意】

无。

### 3.75 获得流模式类型 H264\_PLAY\_GetStreamOpenMode

【目的】获得流模式类型

【语法】LONG H264\_PLAY\_GetStreamOpenMode(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号

## 【返回值】

返回值	宏定义	描述
0	<a href="#">STREAM_FILE</a>	文件流播放模式
1	<a href="#">STREAM_STREAM</a>	数据流播放模式

## 【注意】

无。

### 3.76 获得定时器类型 H264\_PLAY\_GetTimerType

【目的】获得定时器类型

【语法】BOOL H264\_PLAY\_GetTimerType(  
LONG nPort,  
DWORD \*pTimerType)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号



pTimerType	-	-	输出	<a href="#">定时器类型</a>
------------	---	---	----	-----------------------

【返回值】

【注意】

无。

### 3.77 获取播放器中的缓冲区大小 H264\_PLAY\_GetBufferValue

【目的】获取播放器中的缓冲区大小（帧数或者 byte）。这个接口可以帮助用户了解缓冲区中的数据，从而在网络延时方面有所估计

【语法】LONG H264\_PLAY\_GetBufferValue(LONG nPort, DWORD nBufType)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号
nBufType	-	-	-	<a href="#">缓冲区类型</a>

【返回值】

返回值	宏定义	描述
		根据不同参数返回缓冲区值，源缓冲区返回 byte,解码后缓冲区返回帧数。

【注意】

无。

### 3.78 获取当前播放的帧序号 H264\_PLAY\_GetCurrentFrameNum

【目的】得到当前播放的帧序号。而 H264\_PLAY\_GetPlayedFrames 是总共解码的帧数。如果文件播放位置不被改变，那么这两个函数的返回值应该非常接近，除非码流丢失数据

【语法】LONG H264\_PLAY\_GetCurrentFrameNum(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道

				号
--	--	--	--	---

**【返回值】**

返回值	宏定义	描述
-	-	当前播放的帧序号

**【注意】**

无。

### 3.79 获取当前码流中编码时的帧率 H264\_PLAY\_GetPlayFPS

**【目的】** 得到当前码流中编码时的帧率

**【语法】** LONG H264\_PLAY\_GetPlayFPS (LONG nPort)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放器通道号

**【返回值】**

返回值	宏定义	描述
-	-	当前码流编码时的帧率

**【注意】**

无。

### 3.80 获取文件头长度 H264\_PLAY\_GetFileHeadLength

**【目的】** 得到当前版本播放器能播放的文件的文件头长度

**【语法】** DWORD H264\_PLAY\_GetFileHeadLength(void)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
-----	-----	----

-	-	此版本播放器对应的文件头长度
---	---	----------------

**【注意】**

无。

### 3.81 获得文件总的时间长度 H264\_PLAY\_GetFileTime

**【目的】** 获得文件总的时间长度，单位秒

**【语法】** DWORD H264\_PLAY\_GetFileTime(LONG nPort)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
-	-	文件总的时间长度

**【注意】**

无。

### 3.82 获得播放文件总的时间帧数 H264\_PLAY\_GetFileTotalFrames

**【目的】** 获得播放文件总的时间帧数

**【语法】** DWORD H264\_PLAY\_GetFileTotalFrames(LONG nPort)

**【描述】****【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
-	-	文件总的帧数

**【注意】**

无。

### 3.83 获取指定位置关键帧 H264\_PLAY\_GetKeyFramePos

【目的】查找指定位置前的关键帧

【语法】**BOOL H264\_PLAY\_GetKeyFramePos(  
LONG nPort,  
DWORD nValue,  
DWORD nType,  
PFRAME\_POS pFramePos)**

【描述】查找指定位置之前的关键帧位置。图像解码必须从关键帧开始，如果用户保存的文件不是从关键帧开始的，那么倒下一个关键帧之前的数据会被忽略。如果用户要截取文件中的一段数据，则应该考虑从关键帧开始截取。结束位置则关系不大，最多丢失 3 帧数据

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nValue	-	-	输入	当前位置，可以是时间或帧号，类型由 nType 指定
nType	-		输入	如果 nType 是 BY_FRAMENUM，则 nValue 表示帧号，如果 nType 是 BY_FRAMTIME，则 nValue 表示时间，单位 ms
pFramePos	-		输入	查找到的关键帧的文件位置，帧序号，时标信息

【返回值】

返回值	宏定义	描述
0	FALSE	失败
1	TRUE	成功

【注意】

无。

### 3.84 指定位置之后关键帧位置 H264\_PLAY\_GetNextKeyFramePos

【目的】查找指定位置之后的关键帧位置

【语法】**BOOL H264\_PLAY\_GetNextKeyFramePos(**

**LONG nPort,**

**DWORD nValue,**

**DWORD nType,**

**PFRAME\_POS pFramePos)**

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nValue	-	-	输入	当前位置, 可以是时间或帧号, 类型由 nType 指定
nType	-		输入	如果 nType 是 BY_FRAMENUM, 则 nValue 表示帧号, 如果 nType 是 BY_FRAMETIME, 则 nValue 表示时间, 单位 ms
pFramePos	-		输入	查找到的关键帧的文件位置, 帧序号, 时标信息

【返回值】

返回值	宏定义	描述
0	FALSE	失败
1	TRUE	成功

【注意】

无。

### 3.85 获得原始图像的大小 H264\_PLAY\_GetPictureSize

【目的】获得码流中原始图像的大小

【语法】**BOOL H264\_PLAY\_GetPictureSize (**

**LONG nPort,**

**LONG \*pWidth,**

**LONG \*pHeight)**

【描述】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

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pWidth	-	-	输出	原始图像的宽度
pHeight	-	-	输出	原始图像的高度

【返回值】

返回值	宏定义	描述
0	FALSE	失败
1	TRUE	成功

【注意】

无。

### 3.86 获得播放的帧数 H264\_PLAY\_GetPlayedFrames

【目的】得到播放的帧数

【语法】**DWORD H264\_PLAY\_GetPlayedFrames (LONG nPort)**

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
-	-	当前正在播放的帧数

**【注意】**

无。

**3.87 获得文件当前播放的时间 H264\_PLAY\_GetPlayedTime**

**【目的】** 得到文件当前播放的时间，单位秒

**【语法】** DWORD H264\_PLAY\_GetPlayedTime(LONG nPort)

**【描述】**

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
-	-	文件当前播放的时间

**【注意】**

无。

**3.88 获得文件当前播放的时间 H264\_PLAY\_GetPlayedTimeEx**

**【目的】** 得到文件当前播放的时间，单位毫秒

**【语法】** DWORD H264\_PLAY\_GetPlayedTimeEx(LONG nPort)

**【描述】**

**【参数】**

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

**【返回值】**

返回值	宏定义	描述
-	-	文件当前播放的时间

**【注意】**

无。

### 3.89 获得源缓冲剩余数据 H264\_PLAY\_GetSourceBufferRemain

【目的】获得流播放模式下源缓冲剩余数据

【语法】DWORD H264\_PLAY\_GetSourceBufferRemain (LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
-	-	源源缓冲的大小(BYTE)

【注意】

无。

### 3.90 初始化多显卡显示设备 H264\_PLAY\_InitDDrawDevice

【目的】初始化多显卡显示设备

【语法】BOOL H264\_PLAY\_InitDDrawDevice(void)

【描述】

【参数】

【返回值】

【注意】此接口无效

无。

### 3.91 H264\_PLAY\_ReleaseDDrawDevice

【目的】释放枚举显示设备的过程中分配的资源

【语法】BOOL H264\_PLAY\_ReleaseDDrawDevice(void)

【描述】

【参数】

【返回值】

【注意】此接口无效



### 3.92 转换 BMPH264\_PLAY\_ConvertToBmpFile

【目的】将抓图得到的图像数据保存成 BMP 文件

【语法】BOOL H264\_PLAY\_ConvertToBmpFile(

PBYTE \* pBuf,

LONG nSize,

LONG nWidth,

LONG nHeight,

LPCSTR sFileName)

【描述】将抓图得到的图像数据保存成 BMP 文件。转换函数占用的 cpu 资源，如果不需要保存图片，则不要调用

【参数】

参数	成员	取值范围	输入/输出	描述
pBuf	-	-	输入	图像的 YUV 缓冲指针
nSize	-	-	输入	
nWidth	-	-	输入	
nHeight	-	-	输入	
sFileName	-			要保存的文件名。最好以 BMP 作为文件扩展名

【返回值】

返回值	宏定义	描述
1	TRUE	保存成 BMP 文件成功
0	FALSE	参数错误或者打开文件失败，具体错误请看 <a href="#">H264_PLAY_GetLastError</a>

【注意】

无。

### 3.93 H264\_PLAY\_CatchPic

【目的】直接抓取图像(默认 BMP)

【语法】`BOOL H264_PLAY_CatchPic(LONG nPort, LPCSTR sFileName, long ntype=0)`

【描述】抓图。[H264\\_PLAY\\_SetDisplayCallBack](#) 设置的视频数据回调函数，只有在有视频数据解码出来时才调用，并由用户处理视频数据（如抓图），如果不断有解码的数据，就不断调用这个回调函数。而 `H264_PLAY_CatchPic` 一次只抓一幅图，并能在暂停和单帧播放时实现抓图。建议：如果用户想实现抓图（一次抓一幅图），调用 `H264_PLAY_CatchPic`，而如果想得到一段时间内的视频数据，调用 [H264\\_PLAY\\_SetDisplayCallBack](#)

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	-	输入	播放通道号
sFileName	-	-	输入	文件名称
ntype				文件类型 (bmp,jpg)

【返回值】

【注意】

无。

### 3.94 画图回调 H264\_PLAY\_RigisterDrawFun

【目的】画图回调

【语法】`BOOL H264_PLAY_RigisterDrawFun (LONG nPort,  
DrawCallBack pDrawProc,  
LONG nUser)`

【描述】注册一个回调函数，获得当前表面的 **device context**，你可以在这个 DC 上画图（或写字），就好像在窗口的客户区 DC 上绘图，但这个 DC 不是窗口客户区的 DC，而是 **DirectDraw** 里的 *Off-Screen* 表面的 DC。注意，如果是使用 *overlay* 表面，这个接口无效，你可以直接在窗口上绘图，只要不是透明色就不会被覆盖。

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pDrawProc	-	-	输入	回调函数指

				针
nUser	-	-	输入	用户自定义数据

【返回值】

【注意】

无。

### 3.95 设置或增加显示区域 H264\_PLAY\_SetDisplayRegion

【目的】设置或增加显示区域。可以做局部放大显示

【语法】BOOL H264\_PLAY\_SetDisplayRegion(  
LONG nPort,  
DWORD nRegionNum,  
RECT \*pSrcRect,  
HWND hDestWnd,  
BOOL bEnable)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nRegionNum	-	[0, MAX_DISPLAY_WND]	输入	显示区域序号
pSrcRect	-	-	输入	源显示区域
hDestWnd	-	-	输入	显示区域句柄
bEnable	-	[0, 1]	输入	打开关闭显示区域

【返回值】

【注意】

无。

### 3.96 刷新显示 H264\_PLAY\_RefreshPlayEx

【目的】刷新显示，刷新多区域显示的窗口

【语法】BOOL H264\_PLAY\_RefreshPlayEx(LONG nPort, DWORD nRegionNum)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nRegionNum	-	[0, MAX_DISPLAY_WND]	输入	显示区域序号

【返回值】

【注意】

无。

### 3.97 开始流数据录像 H264\_PLAY\_StartDataRecord

【目的】：开始流数据录像。只对流模式有用

【语法】BOOL H264\_PLAY\_StartDataRecord(

LONG nPort,

LPSTR sFileName,

LONG nType)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
sFileName	-	-	输入	录像文件名 如果文件名中有不存在的文件夹，就创建该文件夹
nType	-	MEDIA_FILE_H264 = 1, MEDIA_FILE_AVI = 2,	输入	=1 H264 格式 =2 AVI 格式

				式
--	--	--	--	---

【返回值】

【注意】

在调用 H264\_PLAY\_Play 之后调用。

3.98 停止流数据录像 H264\_PLAY\_StopDataRecord

【目的】：停止流数据录像，只对流模式有用

【语法】BOOL H264\_PLAY\_StopDataRecord(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

【注意】

在调用 H264\_PLAY\_StartDataRecord 之后调用。

3.99 清除源缓冲区剩余数据 H264\_PLAY\_ResetSourceBuffer

【目的】清除流播放模式下源缓冲区剩余数据

【语法】BOOL H264\_PLAY\_ResetSourceBuffer(LONG nPort)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

返回值	宏定义	描述
1	TRUE	清除数据成功
0	FALSE	清除数据失败

【注意】

无。

### 3.100 清空播放器中的缓冲区 H264\_PLAY\_ResetBuffer

【目的】清空播放器中的缓冲区

【语法】BOOL H264\_PLAY\_ResetBuffer(LONG nPort, DWORD nBufType)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nBufType	-	-	输入	<a href="#">缓冲类型</a>

【返回值】

返回值	宏定义	描述
1	TRUE	清除数据成功
0	FALSE	清除数据失败

【注意】

无。

### 3.101 设置搜索区域及范围 H264\_PLAY\_SetMDRange

【目的】设置动检区域及范围

【语法】BOOL H264\_PLAY\_SetMDRange(

LONG nPort,

RECT \*pRect,

DWORD nValueBegin,

DWORD nValueEnd,

DWORD nType)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pRect	left	-	输入	区域左坐标

	top	-	输入	上坐标
	right	-	输入	右坐标
	bottom	-	输入	下坐标
nValueBegin	-	-	输入	搜索范围上限，可以是时间或帧号，具体取决于 nType
nValueEnd	-	-	输入	搜索范围下限，可以是时间或帧号
nType	-	-	输入	BY_FRAMEENUM 或 BY_FRAME TIME

【返回值】

【注意】

无。

### 3.102 设置智能搜索的阈值 H264\_PLAY\_SetMDThreShold

【目的】设置动检搜索的阈值

【语法】BOOL H264\_PLAY\_SetMDThreShold(LONG nPort, DWORD nThreshold)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nThreshold	-	-	输入	阈值

【返回值】

【注意】

无。

### 3.103 获取搜索到的数据帧的帧序号 H264\_PLAY\_GetMDPosition

【目的】获得搜索到的动检帧的帧序号

【语法】 BOOL H264\_PLAY\_GetMDPosition(LONG nPort, DWORD nDirection, DWORD nFrame, DWORD \*MDValue)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nDirection	-	[0,1]	输入	0, 向前搜索, 1 像后搜索
nFrame	-	-	输入	参考帧
MDValue	-	-	输出	搜索到的动检帧的阈值

【返回值】

返回值	宏定义	描述
-	-	搜索到的动检帧的帧序号

【注意】

无。

### 3.104 得到播放 SDK 版本号 H264\_PLAY\_GetSdkVersion

【目的】得到播放 SDK 版本号

【语法】 DWORD H264\_PLAY\_GetSdkVersion(void)

【描述】获取播放库 **SDK** 版本号、次版本号和补丁号

【参数】

【返回值】

返回值	宏定义	描述
-	-	高 16 位表示当前的主版本号。9~16 位表示次版本号，1~8 位表示次补丁号。如：返回值 0x00010101 表示：主版本号是 1，次版本号是 1，补丁号是 1

【注意】

无。



### 3.105 获取错误号 H264\_PLAY\_GetLastError

【目的】获得最后一次出错的错误码

【语法】`BOOL H264_PLAY_GetLastError(LONG nPort)`

【描述】获得当前错误的错误码。用户应该在调用某个函数失败时，调用此函数以获得错误的详细信息。

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

[见错误类型](#)。

### 3.106 刷新显示图像 H264\_PLAY\_RefreshPlay

【目的】刷新显示的图像

【语法】`BOOL H264_PLAY_RefreshPlay(LONG nPort)`

【描述】刷新显示。当用户暂停时如果刷新了窗口，则窗口中的图像因为刷新而消失，此时调用这个接口可以重新把图像显示出来。只有在暂停和单帧播放时才会执行，其它情况会直接返回

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

【注意】

无。

### 3.107 设置信息帧回调 H264\_PLAY\_SetInfoFrameCallBack

【目的】设置信息帧回调

【语法】`BOOL H264_PLAY_SetInfoFrameCallBack(LONG nPort, InfoFramCallBack pProc, LONG nUser)`

【描述】设置信息帧回调，这个接口在调用 `H264_PLAY_OpenStream` 或者 `H264_PLAY_OpenFile` 后使用，如果有信息帧数据，从回调返回给用户

## 【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pProc			输入	回调
nUser			输入	用户参数

## 【返回值】

## 【注意】

无。

### 3.108 获取显卡数目 H264\_PLAY\_GetDDrawDeviceTotalNums

【目的】设置信息帧回调

【语法】DWORD H264\_PLAY\_GetDDrawDeviceTotalNums()

【描述】

【参数】

【返回值】

### 3.109 抓图到内存中 H264\_PLAY\_CatchPicBuf

【目的】直接抓取图像到内存中(默认 BMP)

【语法】DWORD H264\_PLAY\_CatchPicBuf(LONG nPort, char\* pBuf, LONG len, LONG \*width, LONG \*height, long ntype=0)

【描述】直接抓取图像到用户开辟的内存中(默认 BMP),

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
pBuf			输入	写入图像的指针
width				图像宽
height				图像高
ntype				图像类型 0 bmp 1`jpg

【返回值】

【注意】

### 3.110 停止声音采集 H264\_PLAY\_StopAudioCapture

【目的】停止声音采集

【语法】BOOL CALLMETHOD H264\_PLAY\_StopAudioCapture()

【描述】

【参数】

【返回值】

### 3.111 获取空闲通道号 H264\_PLAY\_GetPort

【目的】获取空闲通道号

【语法】BOOL H264\_PLAY\_GetPort(LONG\* nPort)

【描述】获取空闲通道号，上限为 500。与 H264\_PLAY\_FreePort 成对使用

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

【注意】

### 3.112 释放通道号 H264\_PLAY\_FreePort

【目的】释放通道号

【语法】BOOL H264\_PLAY\_FreePort(LONG nPort);

【描述】释放通道号 H264\_PLAY\_GetPort 得到的通道号

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

【注意】

### 3.113 获得正在显示帧的时间戳 H264\_PLAY\_GetCurTimeStamp

【目的】获得在显示帧的时间戳,单位 s

【语法】DWORD H264\_PLAY\_GetCurTimeStamp(LONG nPort)

【描述】获得在显示帧的时间戳

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号

【返回值】

【注意】

### 3.114 设置 OSD 叠加信息 H264\_PLAY\_SetOsdTex

【目的】设置 osd 叠加信息

【语法】int H264\_PLAY\_SetOsdTex(LONG nPort, OSD\_INFO\_TXT \* txt)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
txt				OSD 信息

【返回值】

【注意】必须在 openstream 后，play 之前设置，否则无效

### 3.115 删除 OSD 叠加信息 H264\_PLAY\_DeleteOsdTex

【目的】删除 osd 叠加信息

【语法】int H264\_PLAY\_DeleteOsdTex(LONG nPort, int nIndex)

【描述】

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	播放通道号
nIndex				H264_PLAY_SetOsdTex 返回值

【返回值】

【注意】

### 3.116 录像文件格式转换 H264\_PLAY\_ConvertFile

【目的】录像文件格式转换

【语法】int H264\_PLAY\_ConvertFile(const char\* pSourceFile, const char\* pDestFile, LONG nType, CoverFileCallBack coverPosCallback, DWORD dwUser)

【描述】H264 录像文件转 Avi 录像文件，现在只支持 H264 到 AVI 转换

【参数】

参数	成员	取值范围	输入/输出	描述
pSourceFile	-		输入	源文件
pDestFile				目的文件
nType				转换类型，只能是 MEDIA_FILE_AVI
coverPosCallback				进度回调函数
dwUser				用户自定义数据

【返回值】

【注意】

### 3.117 只播放 I 帧 H264\_PLAY\_OnlyIFrame

【目的】只播放 I 帧

【语法】BOOL H264\_PLAY\_OnlyIFrame(LONG nPort, bool bPlayOnlyIFrame);

【描述】只播放 I 帧，后面的 P 帧不解码和显示

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	源文件
bPlayOnlyIFrame				是否只播放 I 帧

【返回值】

【注意】

### 3.118 标准 H264 写入 H264\_PLAY\_ InputStdH264Data

【目的】把标准的 H264 写入到源缓存

【语法】`BOOL H264_PLAY_ InputStdH264Data(LONG nPort,  
STD_H264_PACKET_INFO *infor)`

【描述】把标准的 H264 写入到源缓存，进行界面播放，和 H264\_PLAY\_InputData 一样的功能和调用顺序，区别在于一个是带私有头的，一个是标准的

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	源文件
infor				标准 H264

【返回值】

【注意】

### 3.119 标准 H264 写入 H264\_PLAY\_ InputStdH264Data

【目的】把标准的 H264 写入到源缓存

【语法】`BOOL H264_PLAY_ InputStdH264Data(LONG nPort,  
STD_H264_PACKET_INFO *infor)`

【描述】把标准的 H264 写入到源缓存，进行界面播放，和 H264\_PLAY\_InputData 一样的功能和调用顺序，区别在于一个是带私有头的，一个是标准的

【参数】

参数	成员	取值范围	输入/输出	描述
nPort	-	[0, 100]	输入	源文件
infor				标准 H264

【返回值】

【注意】