# IPC 智能高清网络摄像机 SDK 用户手册

IPC SDK

Version 3.0

# 目　录


1　概述 ..... 4

2 功能接口介绍 ..... 5

3 流程介绍 ..... 13

1.1　实时码流 ..... 14

1.2　视频设置 ..... 14

1.3　音频监听、对讲 ..... 15

1.4　网络设置 ..... 15

1.5　云台设置 ..... 16

1.6　事件设置 ..... 16

1.7　系统设置 ..... 17

1.8　SD 卡录像 ..... 18

2　结构与宏定义 ..... 19

2.1　数据结构 ..... 19

2.1.1　实时码流 ..... 19

2.1.2　视频管理 ..... 20

2.1.3　网络管理 ..... 35

2.1.4　云台管理 ..... 42

2.1.5　事件管理 ..... 47

2.1.6　系统管理 ..... 53

2.1.7　语音对讲 ..... 55

2.1.8　日志 ..... 57

2.1.9　SD 卡录像 ..... 60

2.2　宏定义 ..... 62

2.2.1　实时码流 ..... 62

2.2.2　视频管理 ..... 62

2.2.3　网络管理 ..... 64

2.2.4　云台管理 ..... 64

2.2.5　事件管理 ..... 66

2.2.6　系统管理 ..... 66

2.2.7　限制性常量定义 ..... 66

2.2.8　日志 ..... 67

2.2.9　SD 卡录像 ..... 67

3　接口定义 ..... 68

3.1　SDK 初始化 ..... 68

3.2　网络连接 ..... 72

3.3 实时浏览 ..... 77
3.4 视频设置 ..... 113
3.5 网络管理 ..... 130
3.6 云台管理 ..... 149
3.7 报警中心上传 ..... 166
3.8 事件管理 ..... 167
3.9 布防设置 ..... 190
3.10 系统管理 ..... 195
3.11 音频监听 ..... 206
3.12 语音对讲 ..... 216
3.13 日志 ..... 226
3.14 SD 卡录像 ..... 236
4 编程引导 ..... 242
4.1 视频预览及云台控制 ..... 242
4.2 视频码流回调 ..... 242
4.3 播放本地文件 ..... 242
5 帮助 ..... 243
5.1 注意事项 ..... 243
5.2 常见问题解答 ..... 243

# 1概述

智能高清网络摄像机 SDK 分为四部分：视频流控制、外部设备操作、安防设置、本地文件播放。 它专门为智能高清网络摄像机设计的接口程序，以动态库的形式提供给应用软件开发者，并同时附有演示程序及其源代码，能有效地缩短应用软件的开发周期。SDK 主要的功能包括：实时浏览、录像、截图、机械云台控制、数字云台控制、事件设置、属性设置、系统参数设置等。SDK 包含的组件说明如表 1.所示。

**表 1.** 设备端通讯端口列表

| 端口号 | 协议 | 用途 |
|---|---|---|
| 30000 | TCP | 默认信令传输端口，比如登录、云台控制，配置数据等。视频、音频、对讲端口以此为基础依次加 2 |
| 161 | UDP | Snmp 协议 |

**表 2.** SDK 端侦听端口列表

| 端口号 | 协议 | 用途 |
|---|---|---|
| 可配置端口 | TCP | 私有 DDNS 信息上报 |
| 可配置端口 | TCP | 报警信息上传报警中心 |

**表 3.** SDK 包含的组件

| **SDK 功能接口** | | |
|---|---|---|
| SDK 功能接口头文件 | IPCSdkDefines.h | SDK 数据类型定义 |
| | IPCDefines.h | SDK 数据类型定义 |
| | IPCCamera.h | SDK 摄像机部分 API 接口 |
| | IPCCameraBackup.h | SDK 摄像机部分 API 接口，此处为预留或以前的旧接口 |
| | IPCSdkDefinesBackup.h | SDK 数据类型定义，此处为预留或以前的旧类型定义 |
| | IPCPlayer.h | SDK 本地文件播放部分 API 接口,兼容以前的 IPCFilePlayer.h 的所有接口 |
| **摄像机运行模块** | | |
| 公共基础库 | IPCCamera.dll | 公共基础文件,通信控制，码流获取 |
| 码流播放库 | IPCPlayer.dll | 多媒体显示及录像回放模块 |
| | IPCQuartz.dll | |
| Sample | IPCSdkFuncDemo | 功能的综合演示程序 |
| | IPCFilePlayDemo | 本地文件播放演示程序 |

# 2 功能接口介绍

设备功能接口按照下列顺序进行阐述：初始化，用户管理， 网络管理，设备管理，视频操作，音频操作，事件管理，存储管理，系统管理。

| 功能 | 接口 | 相关参数 |
|---|---|---|
| **设备初始化** | | |
| SDK 全局初始化 | ipcInitialize | |
| SDK 全局析构 | ipcUninitialize | |
| 设备初始化,创建设备操作句柄 | ipcCreateDevice | |
| 释放设备操作句柄 | ipcDestroyDevice | |
| **网络连接** | | |
| 与设备建立网络连接 | ipcConnect | |
| 断开与设备的网络连接 | ipcDisconnect | |
| 设备链路状态 | ipcIsConnected | |
| 等待 SDK 与设备建立连接 | ipcWaitConnected | 单位毫秒，建议大于 5000ms. |
| 设置 API 执行网络操作时的超时时限 | ipcSetConnectTimeOut | 默认 10*1000 毫秒 |
| **网络管理** | | |
| 设置设备端口 | ipcSetDevicePort | |
| 设置 IPv4 地址 | ipcSetDeviceIPInfo | 参考结构体 SYSIPINFO |
| 获取当前 Ipv4 地址配置信息 | ipcGetDeviceIPInfo | 参考结构体 SYSIPINFO |
| **SMTP 设置** | | |
| 设置 SMTP 配置 | ipcSetDeviceSMTPInfo | 参考结构体 SYSSMTP |
| 获取当前 SMTP 配置信息 | ipcGetDeviceSMTPInfo | 参考结构体 SYSSMTP |
| **UPnp 设置** | | |
| 设置 upnp 配置 | ipcSetDeviceUPNPInfo | 参考结构体 SYSUPNPINFO |
| 获取当前 upnp 配置 | ipcGetDeviceUPNPInfo | 参考结构体 SYSUPNPINFO |
| **FTP 设置** | | |
| 设置 FTP 配置 | ipcSetDeviceFTPInfo | 参考结构体 SYSFTP |
| 获取当前 FTP 配置 | ipcGetDeviceFTPInfo | 参考结构体 SYSFTP |
| **DDNS 设置** | | |
| 获取 DDNS 配置 | ipcGetDDNSParam | 参考结构体 DDNSPARA |
| 设置 DDNS 配置 | ipcSetDDNSParam | 参考结构体 DDNSPARA |
| DDNS 信息回调 | ipcDDNSCallBack | 参考结构体 DDNSRecord |
| 设置 DDNS 信息回调 | ipcSetDDNSServerCallBack | 参考回调函数 ipcDDNSCallBack |
| 启动 DDNS 服务 | ipcStartDdnsListen | |
| 停止 DDNS 服务 | ipcStopDdnsListen | |
| **设备名称配置** | | |
| 设置设备名称 | ipcSetDeviceName | |
| 获取设备名称 | ipcGetDeviceName | |

| | | |
|---|---|---|
| **设备管理** | | |
| 设备搜索 | ipcScanOnlineDevices | |
| 停止设备收索 | ipcStopScanOnlineDevices | |
| 扫描回调函数 | ipcScanOnlineDeviceCallBack | |
| **视频属性管理** | | |
| 获取设备当前的码流信息 | ipcGetStreamInfo | 参考结构体 StreamInfo |
| 获取设备当前各个码流编码信息 | ipcGetCurAllStreamInfo | 参考结构体 DeviceImgInfo |
| 设置设备当前各个码流编码信息 | ipcSetCurAllStreamInfo | 参考结构体 DeviceImgInfo |
| 设置视频属性设置 | ipcSetConfig | 1 抗闪烁模式:<br>IPC_VIDEO_EXPOSURE_MODE<br>2 曝光目标系数:<br>IPC_VIDEO_AE_TARGET_RATION<br>3 SenSor 最大增益:<br>IPC_VIDEO_MAX_GAIN<br>4 白平衡设置:<br>IPC_VIDEO_WBC<br>IPC_VIDEO_WBC_CUSTOM_R_GAIN<br>IPC_VIDEO_WBC_CUSTOM_B_GAIN<br><br>5 夜晚模式:<br>IPC_VIDEO_DN_MODE<br>6 背光补偿:<br>IPC_VIDEO_BACKLIGHT_COMP<br>7 局部曝光模式:<br>IPC_VIDEO_LOCAL_EXPOSURE<br>8 3D 降噪:<br>IPC_VIDEO_MCTF_STRENGTH<br>9 自动曝光设置中 SlowShutter:<br>IPC_VIDEO_SLOW_SHUTTER<br>10 自动曝光的偏好:<br>IPC_VIDEO_AE_PREFERENCE<br>11 自动曝光设置中测光模式:<br>IPC_VIDEO_METERING_MODE<br>12 编码模式:<br>IPC_ENCODE_ENC_MODE<br>13 显示当前时间:<br>IPC_ENCODE_SN_TIME_ENABLEO<br>14 DC IRIS 模式:<br>IPC_VIDEO_DCIRIS_GROUP<br>15 电子快门时间范畴:<br>IPC_VIDEO_SHUTTER_GROUP<br>16 视频遮挡:<br>参见 SetPictMask/GetPictMask |

| | | |
|---|---|---|
| | | 17　显示信息叠加:<br>IPC_ENCODE_SN_TEXT_GROUP |
| 获取视频属性信息 | ipcGetConfig | 同上 |
| 设置视频遮挡信息 | ipcSetPictMask | |
| 获取视频遮挡信息 | ipcGetPictMask | |
| 设置 MJPEG 实时码流图像质量 | ipcSetImageQuant | 1~100 |
| 获取支持的分辨率、帧率 | ipcGetSupportedImgInfo | |
| 获取设备的码率 | ipcGetBitRate | 参考结构体 |
| 设置设备的码率 | ipcSetBitRate | |
| 设置设备的码流个数 | ipcSetSupportedStreamCount | |
| 获取设备的码流个数 | ipcGetSupportedStreamCount | |
| 获取设备音频编码状态 | ipcGetAudioState | |
| 设置设备音频编码状态 | ipcSetAudioState | 启用、禁用音频 |
| 获取电子快门时间的最小值取值范围 | ipcGetShutterTimeMinRange | |
| 获取电子快门时间中最小值所对应的最大值取值范围 | ipcGetShutterTimeMaxRange | |
| 开启图像锐化 | ipcStartImageSharpen | |
| 改变图像的锐化程度 | ipcChangeSharpenDegree | |
| 关闭图像锐化 | ipcStopImageSharpen | |
| **视频操作** | | |
| **本地抓图** | | |
| 单帧数据捕获并保存成图形存放在指定的文件中 | ipcCapturePicture | 1: BITMAP 24 bit , 2:JPEG, |
| 单帧数据捕获并保存成图形存放在指定的内存空间中 | ipcCapturePictToBuff | 只支持 1: BITMAP24 bit |
| **本地录像** | | |
| 开始录像，并保存至本地文件 | ipcStartRecord | 暂只支持 asf 格式录制同时生成名为 ipcdx 的索引文件 |
| 结束录像 | ipcStopRecord | |
| 创建 MP4 格式文件并返回操作句柄 | ipcMP4Open | |
| 关闭 MP4 格式文件 | ipcMP4Close | |
| 添加视频流 | ipcMP4AddVideoStream | |
| 添加音频流 | ipcMP4AddAudioStream | |
| 开始 MP4 格式录像 | ipcMP4Run | |
| 结束 MP4 格式录像 | ipcMP4Stop | |
| 写入音视频数据 | ipcMP4Write | 首帧必须为关键帧 |
| **实时预览** | | |
| 创建预览句柄 | ipcCreateLivePlay | 窗口填 NULL 表只获取码流 |
| 释放设备句柄 | ipcDestroyLivePlay | |
| 开始播放预览 | ipcStartLivePlay | |
| 停止预览 | ipcStopLivePlay | |

| | | |
|---|---|---|
| 设置视频的显示位置 | ipcSetDisplayRect | |
| 获取视频的显示位置 | ipcGetDisplayRect | |
| 设置播放缓存时间 | ipcSetBufferTime | |
| 获取播放缓存时间 | ipcGetBufferTime | |
| 解码前数据回调函数 | ipcRegisterLiveFrameCallBack | 只在回调的 IPCVIDEO_CALLBACK_INFO 结构中返回视频数据和视频相关信息，无时间戳。<br>推荐使用 ipcSetAvCallbackEx2 来替代此回调设置。 |
| 实时解码数据回调函数 | ipcRegisterLiveYUVCallBack | 数据为 YUY2 (422) |
| 视频解码前数据捕获回调函数 | ipcLiveFrame_CallBack | 仅回调视频流 |
| 设置音视频混合流回调 | ipcSetAVCallbackEx | 在回调函数的 FrameInfoEx 结构中返回音视频数据并提供时间戳，需根据标志来区分是音频数据或视频数据 |
| 叠加扩展文字 | ipcSetDisplayOSD | 本地播放器叠加 osd |
| 图片叠加 | ipcSetDisplayOSD | |
| 设置设备 OSD | ipcSetDeviceOSD | 摄像头叠加 OSD |
| 获取设备 OSD 信息 | ipcGetDeviceOSD | 参考结构体: DEVICEOSD |
| 设置设备时间信息显示格式 | ipcSetDeviceDateTimeOSD | 摄像头叠加时间 OSD |
| 获取设备时间信息显示格式 | ipcGetDeviceDateTimeOSD | 参考结构体: DATETIMEOSD |
| 获取视频宽度 | ipcGetImageWidth | |
| 获取视频高度 | ipcGetImageHeight | |
| 获取视频帧率 | ipcGetVideoFPS | |
| 设置视频帧率 | ipcSetVideoFPS | |
| 强制设备生成 I 帧 | ipcForceIDR | |
| **云台控制** | | |
| 数字云台控制 | ipcDPTZControl | 详见 API Remark 部分 |
| 机械云台控制 | ipcPTZControl | |
| 设置机械云台预置位 | ipcPTZSetPreset | 最多 255 个预置点 |
| 转移到机械云台预置位 | ipcPTZGotoPreset | |
| 删除机械云台预置位 | ipcPTZClearPreset | |
| 获取机械云台预置位信息 | ipcPTZGetAllPreset | 参考结构体 **IPC_PRESET,** 最多 255 个预置点 |
| 获取机械云台预置位能力 | ipcPTZGetCapacity | |
| 操作一条巡航路径 | ipcSetPTZCruiseControl | |
| 获取一条巡航路径 | ipcGetPTZCruise | |
| 设置一条巡航路径 | ipcSetPTZCruise | |
| 透明云台通道 | ipcSetPTZTrans | |
| 设置透明通道的属性 | ipcSetPTZTransProp | [协议,地址,波特率 |
| 获取云台配置信息 | ipcGetPTZTransProp | |
| 控制机械云台左上移动 | ipcPTZUpleft | |

| | | |
|---|---|---|
| 控制机械云台左下移动 | ipcPTZDownleft | |
| 控制机械云台右上移动 | ipcPTZUpright | |
| 控制机械云台右下移动 | ipcPTZDownright | |
| 控制机械云台坐标 | ipcPTZPositionControl | |
| 控制机械云台区域缩放 | ipcPTZSelZoomIn | |
| **音频监视** | | |
| 设置音视频混合流回调 | ipcSetAVCallbackEx | 在回调函数的 FrameInfoEx 结构中返回音视频数据并提供时间戳，需根据标志来区分是音频数据或视频数据 |
| 设置音视频混合流回调扩展 | ipcSetAVCallbackEx2 | 在回调数据之前包含了头信息等 IPC 流文件播放及检索用到的相关数据，可按照格式拆分出带时间戳的音频或视频数据，此数据直接写文件可生成 IPC 流播放文件<br>格式参见 IPC_STREAM_HEADER，IPC_VIDEOFRAME_HEADER，IPC_AUDIOFRAME_HEADER，IPC_FRAME_TAIL |
| 创建音频监听资源 | ipcCreateAudioLivePlay | |
| 开始音频监听 | ipcStartAudioLivePlay | |
| 停止音频监听 | ipcStopAudioLivePlay | |
| 释放音频监听资源 | ipcDestroyAudioLivePlay | |
| 配置 IPC 音频参数 | ipcSetAudioDeviceConfig | |
| 获取 IPC 音频参数 | ipcGetAudioDeviceConfig | |
| 设置音频监听的控制参数 | ipcSetAudioLivePlayControl | |
| 获取音频监听的控制参数 | ipcGetAudioLivePlayControl | |
| 静音音频监听 | ipcSetAudioLivePlayMute | |
| 获取音频监听的静音标志 | ipcGetAudioLivePlayMute | |
| **语音对讲** | | |
| 开始语音对讲 | ipcStartTalk | |
| 停止语音对讲 | ipcStopTalk | |
| 设置语音对讲的控制参数 | ipcSetTalkControl | |
| 获取语音对讲的控制参数 | ipcGetTalkControl | |
| 音频对讲数据回调函数 | ipcTalkCallback | |
| 设置语音对讲回调 | ipcSetTalkCallback | |
| 枚举声卡设备 | ipcEnumAudioDevice | |
| 设置语音对讲声卡设备 | ipcSetAudioDevice | |
| 获取语音对讲声卡设备 | ipcGetAudioDevice | |
| 直接发送音频数据到 IPC | ipcSendAudioStream | |
| 编码音频数据 | ipcEncodeAudioStream | |
| **SD 卡录像** | | |
| 控制操作录像文件 | ipcRecordControl | |
| 查询录像文件 | ipcFindRecordFile | |
| 逐个获取查找到的文件信息 | ipcFindNextRecordFile | |

| | | |
|---|---|---|
| 关闭文件查找，释放资源 | ipcFindRecordClose | |
| 录像文件下载 | ipcDownloadByRecordName | 仅支持保存 ipc 格式，不支持续传 |
| 停止录像文件下载 | ipcStopDownload | |
| 获取录像文件下载进度 | ipcDownloadPos | |
| | | |
| **事件管理** | | |
| **视频异常** | | |
| 获取视频异常事件 | ipcGetCameraUnusualEvent | |
| 设置视频异常事件 | ipcSetCameraUnusualEvent | |
| **移动侦测** | | |
| 获取移动侦测报警联动配置信息 | ipcGetMotionEventAction | |
| 设置移动侦测报警联动配置信息 | ipcSetMotionEventAction | |
| 获取移动侦测区域配置信息 | ipcGetMotionWndInfo | |
| 设置移动侦测区域配置信息 | ipcSetMotionWndInfo | |
| **报警输入** | | |
| 获取报警输入事件 | ipcGetInputEvent | |
| 设置报警输入事件 | ipcSetInputEvent | |
| **报警输入输出端口设置** | | |
| 获取报警输入端口数 | ipcGetInputPortNumber | 目前仅支持 1 个 |
| 获取报警输出端口数 | ipcGetOutputPortNumber | 目前仅支持 1 个 |
| 获取报警输入端口状态 | ipcGetInputPortStatus | 目前仅支持 1 个 |
| 设置报警输入端口触发条件 | ipcSetInputPortTriggerStatus | 目前仅支持 1 个,0 低电平 1 高电平 |
| 获取报警输入端口触发条件 | ipcGetInputPortTriggerStatus | 目前仅支持 1 个,0 低电平 1 高电平 |
| 获取报警输出端口状态 | ipcGetOutputPortStatus | 目前仅支持 1 个 |
| 设置报警输出端口状态 | ipcSetOutputPortStatus | 目前仅支持 1 个 |
| 设置输出端口报警联动行为 | ipcSetOutputPortAction | |
| 获取输出端口报警联动行为 | ipcGetOutputPortAction | |
| 启用禁用报警输出口 | ipcSetOutputPortEnable | |
| 启用禁用报警输入口 | ipcSetInputPortEnable | |
| 获取报警输入口启用禁用配置 | ipcGetInputPortEnable | |
| 获取报警输出口启用禁用配置 | ipcGetOutputPortEnable | |
| 手动触发 IPC 报警输出 | ipcTriggerAlarmOutput | 目前仅支持 1 个 |
| **事件使能设置** | | |
| 设置事件使能 | ipcSetEventEnable | |
| 获取事件使能 | ipcGetEventEnable | |
| **布防设置** | | |
| 获取视频异常布防信息 | ipcGetCameraUnusualPlan | |
| 设置定时录像布防信息 | ipcSetCameraUnusualPlan | |
| 获取移动侦测布防信息 | ipcGetMotionDetectionPlan | |
| 设置移动侦测布防信息 | ipcSetMotionDetectionPlan | |

| | | |
|---|---|---|
| 获取报警输入布防 | ipcGetInputPlan | |
| 设置报警输入布防 | ipcSetInputPlan | |
| **报警事件中心上传** | | |
| 注册报警回调函数 | ipcSetAlarmCenterCallBack | |
| 开启报警中心监听 | ipcStartListen | |
| 停止报警中心监听 | ipcStopListen | |
| 设置报警中心 | ipcSetSystemSetting | IPC_SYSTEM_GETSET_ALARMCENTER |
| 获取报警中心 | ipcGetSystemSetting | IPC_SYSTEM_GETSET_ALARMCENTER |
| **系统管理** | | |
| **升级** | | |
| 启动升级服务 | ipcUpdateServiceStart | |
| 停止升级服务 | ipcUpdateServiceStop | |
| 设备升级 | ipcUpgrade | |
| **日志** | | |
| 设置日志保存策略 | ipcSetLogConfig | |
| 获取日志保存策略 | ipcGetLogConfig | |
| 查找设备的日志信息 | ipcFindLog | |
| 逐条获取查找到的日志信息 | ipcFindNextLog | |
| 释放查找日志的资源 | ipcFindLogClose | |
| 根据日志 ID 删除指定日志 | ipcDeleteLogById | |
| 清空全部日志 | ipcClearLog | |
| 查询日志个数 | ipcQueryLogCount | |
| 删除符合条件的日志 | ipcDeleteLog | |
| 导出日志到文件 | ipcExportLog | |
| **系统维护** | | |
| 设备恢复出厂设置 | ipcSystemRestore | |
| 待机 | ipcDeviceStandby | |
| 唤醒 | ipcDeviceWakeup | |
| 重启 | ipcDeviceRestart | |
| **时间管理** | | |
| 设置设备时间 | ipcSetSystemSetting | IPC_SYSTEM_SETGET_DATEINFO<br>IPC_SYSTEM_SETGET_DATEINFOEX |
| 获取当前设备时间 | ipcGetSystemSetting | IPC_SYSTEM_SETGET_DATEINFO<br>IPC_SYSTEM_SETGET_DATEINFOEX |
| **异常处理** | | |
| **心跳** | | |
| 设置心跳及回调 | ipcSetHeartbeat<br>ipcSetHeartbeatEx | 最小心跳间隔 5S<br>扩展版本支持与设备建立连接前使用 |
| 心跳回调函数 | ipcHeartBeatCallback | |
| **设备基本信息** | | |
| 获取设备名称 | ipcGetSystemSetting | IPC_SYSTEM_GET_DEVICE_NAME |
| 获取设备制造商 | ipcGetSystemSetting | IPC_SYSTEM_GET_DEVICE_MANUFACTURE |

| | | |
|---|---|---|
| | | R |
| 获取固件 FW 版本 | ipcGetSystemSetting | IPC_SYSTEM_GET_FW_VERSION |
| 获取固件 HW 版本 | ipcGetSystemSetting | IPC_SYSTEM_GET_HW_VERSION |
| 获取设备序列号 | ipcGetDeviceSN | |

# 3 流程介绍

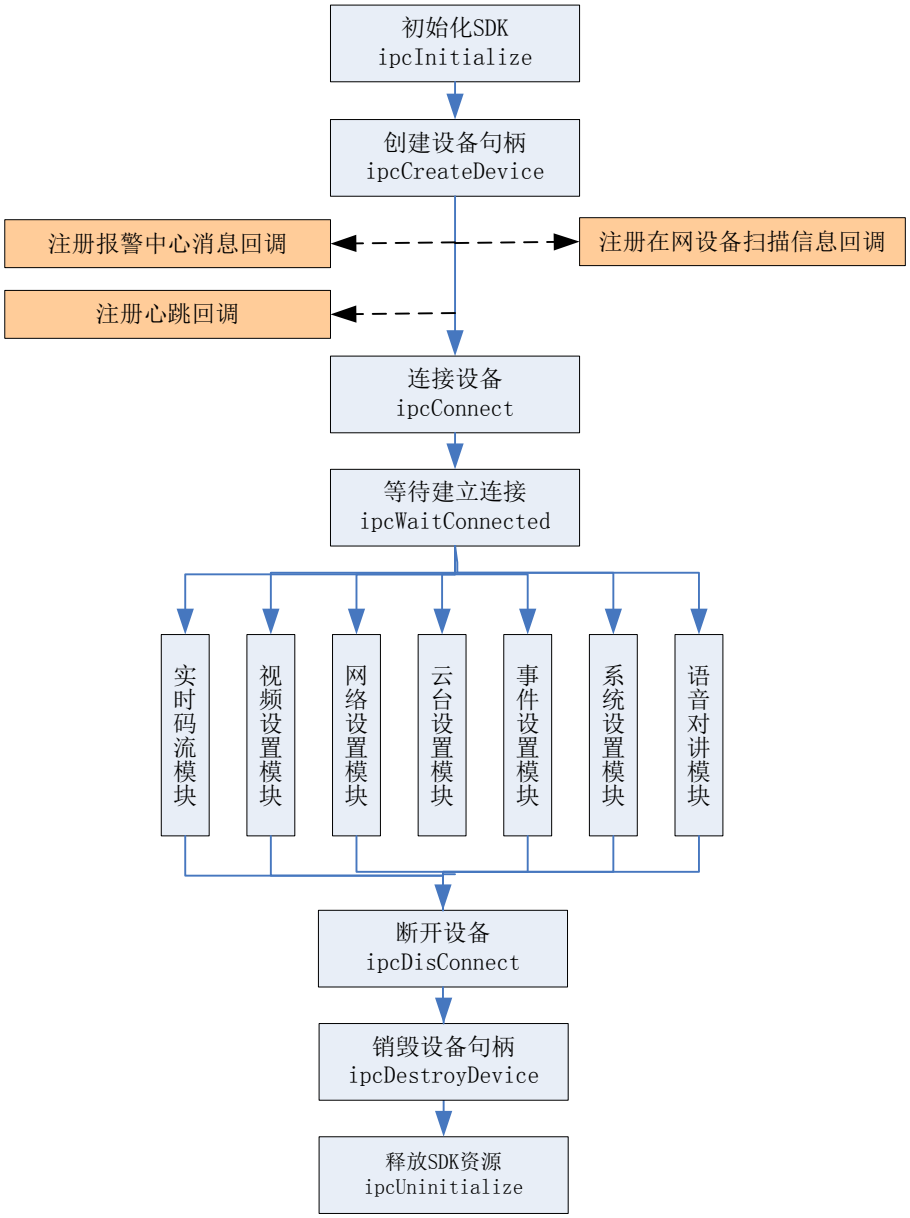

初始化SDK
ipcInitialize
创建设备句柄
ipcCreateDevice
注册报警中心消息回调
注册在网设备扫描信息回调
注册心跳回调
连接设备
ipcConnect
等待建立连接
ipcWaitConnected
实时码流模块
视频设置模块
网络设置模块
云台设置模块
事件设置模块
系统设置模块
语音对讲模块
断开设备
ipcDisConnect
销毁设备句柄
ipcDestroyDevice
释放SDK资源
ipcUninitialize

图 3-1

## 1.1 实时码流

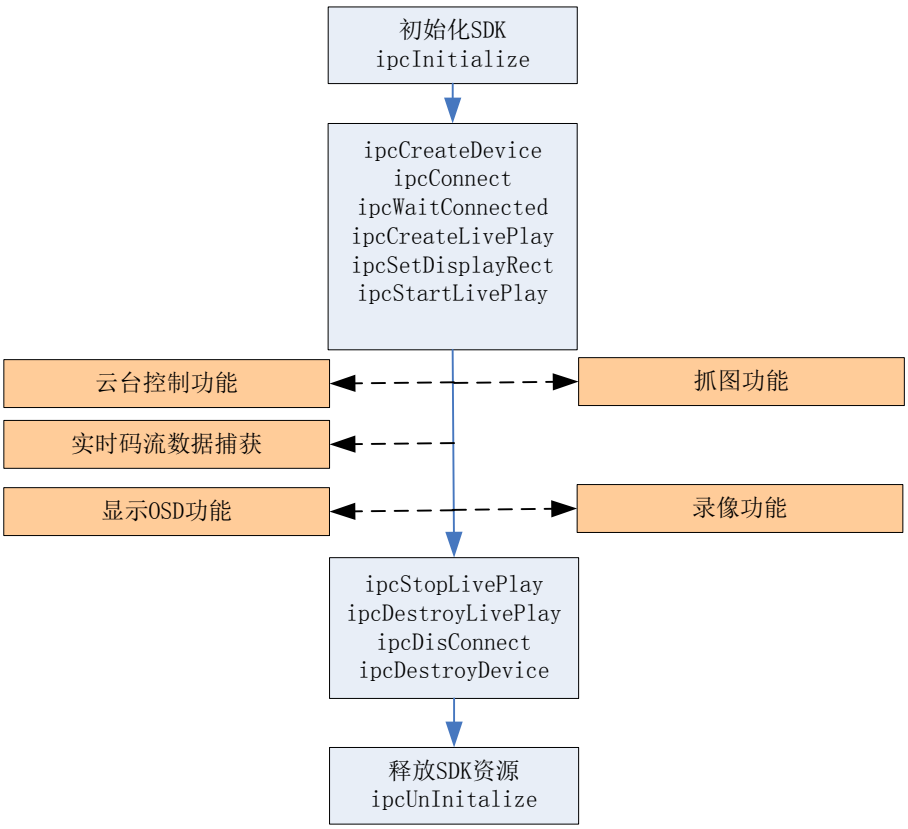



图 3-2

## 1.2 视频设置

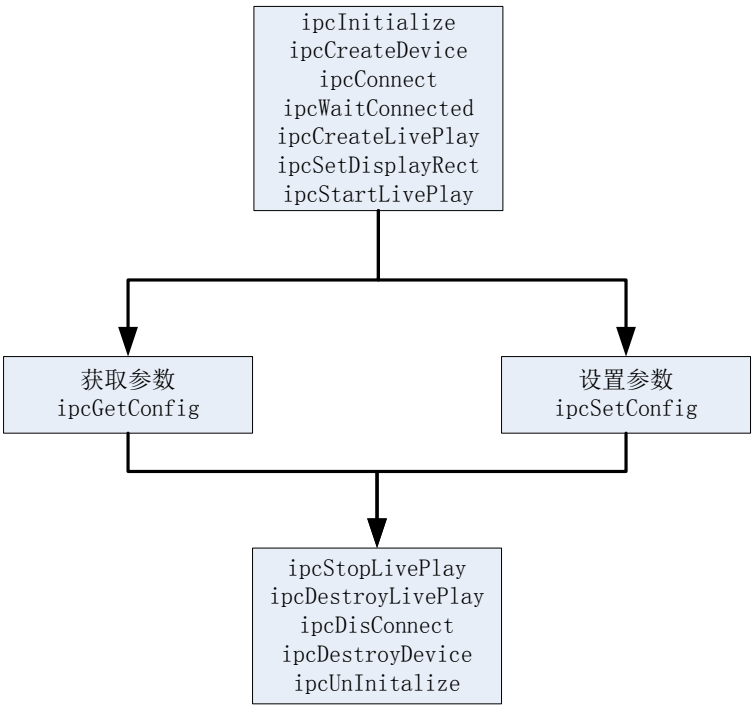



图 3-3

## 1.3 音频监听、对讲

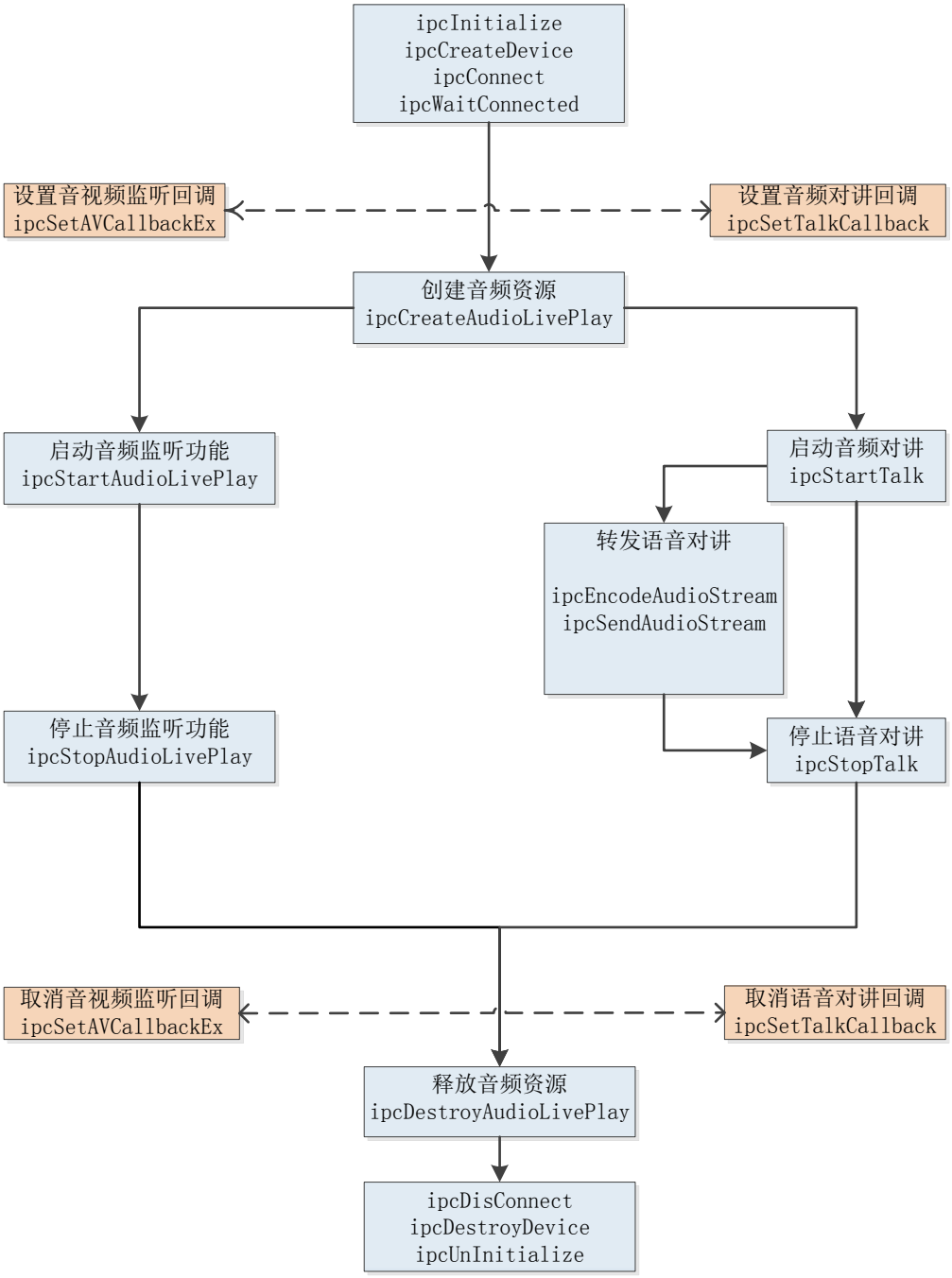



图 3-4

## 1.4 网络设置

网络设置流程与视频设置流程相同 。

## 1.5 云台设置

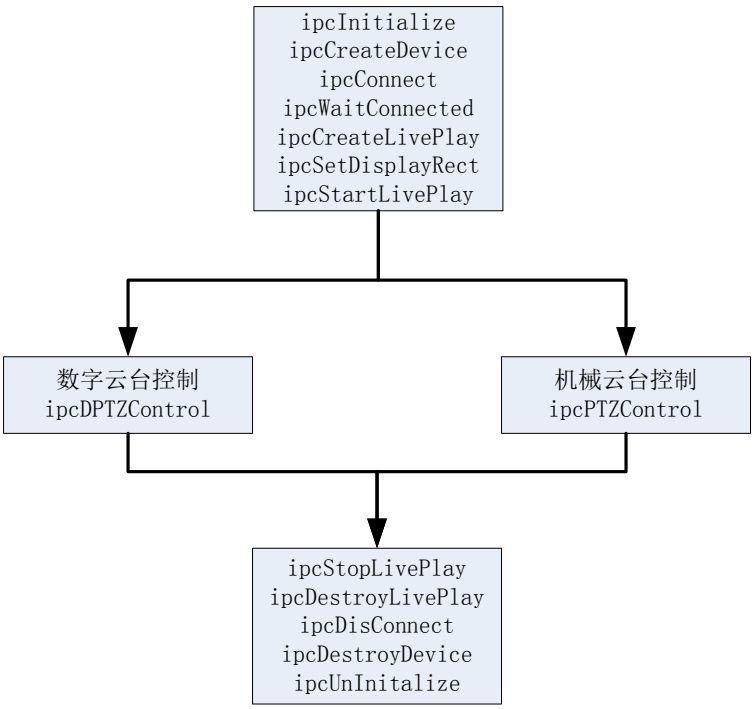



图 3-5

## 1.6 事件设置

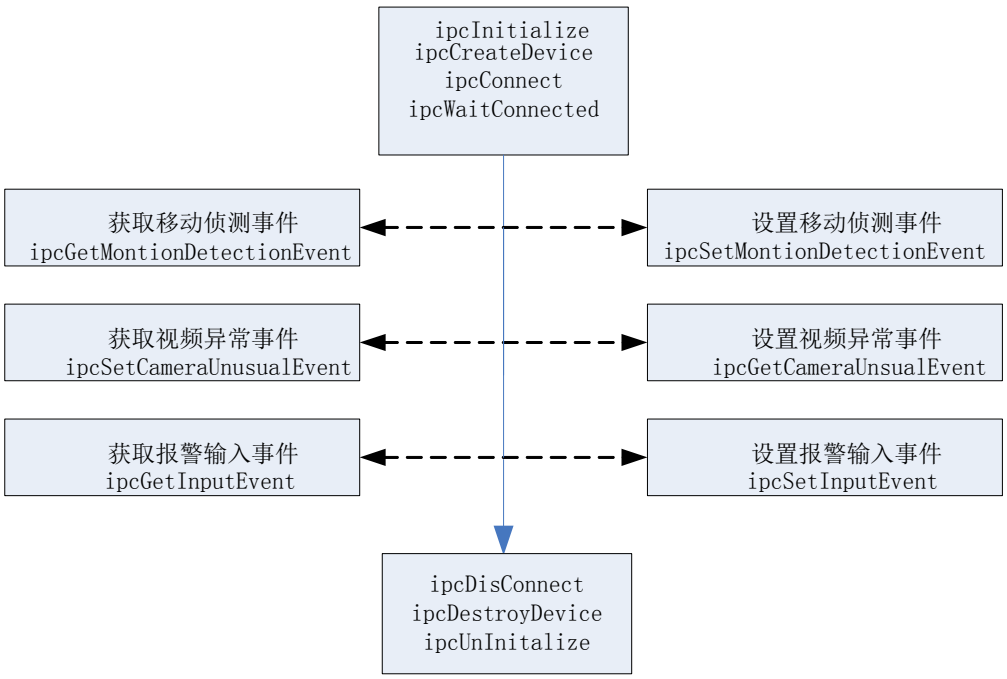



图 3-6

## 1.7 系统设置

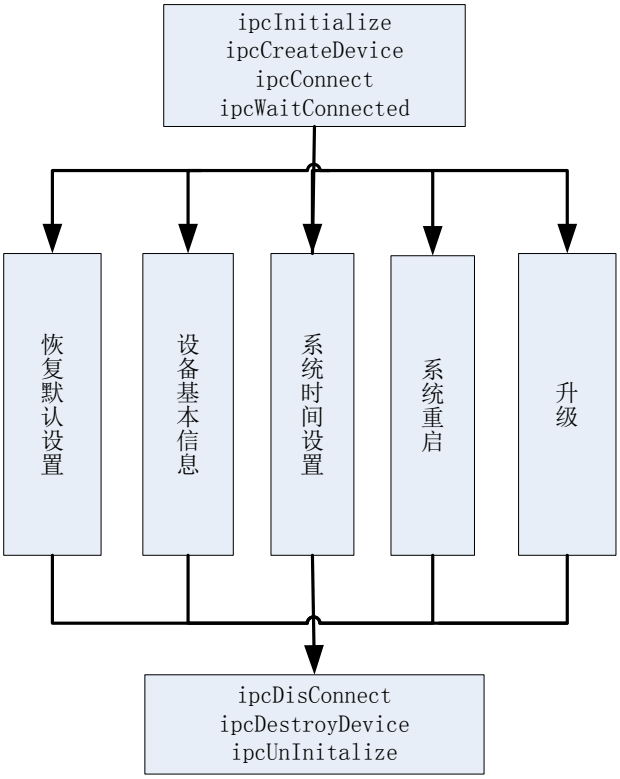

ipcInitialize
ipcCreateDevice
ipcConnect
ipcWaitConnected
恢复默认设置
设备基本信息
系统时间设置
系统重启
升级
ipcDisConnect
ipcDestroyDevice
ipcUnInitalize

图 3-7

## 1.8 SD 卡录像

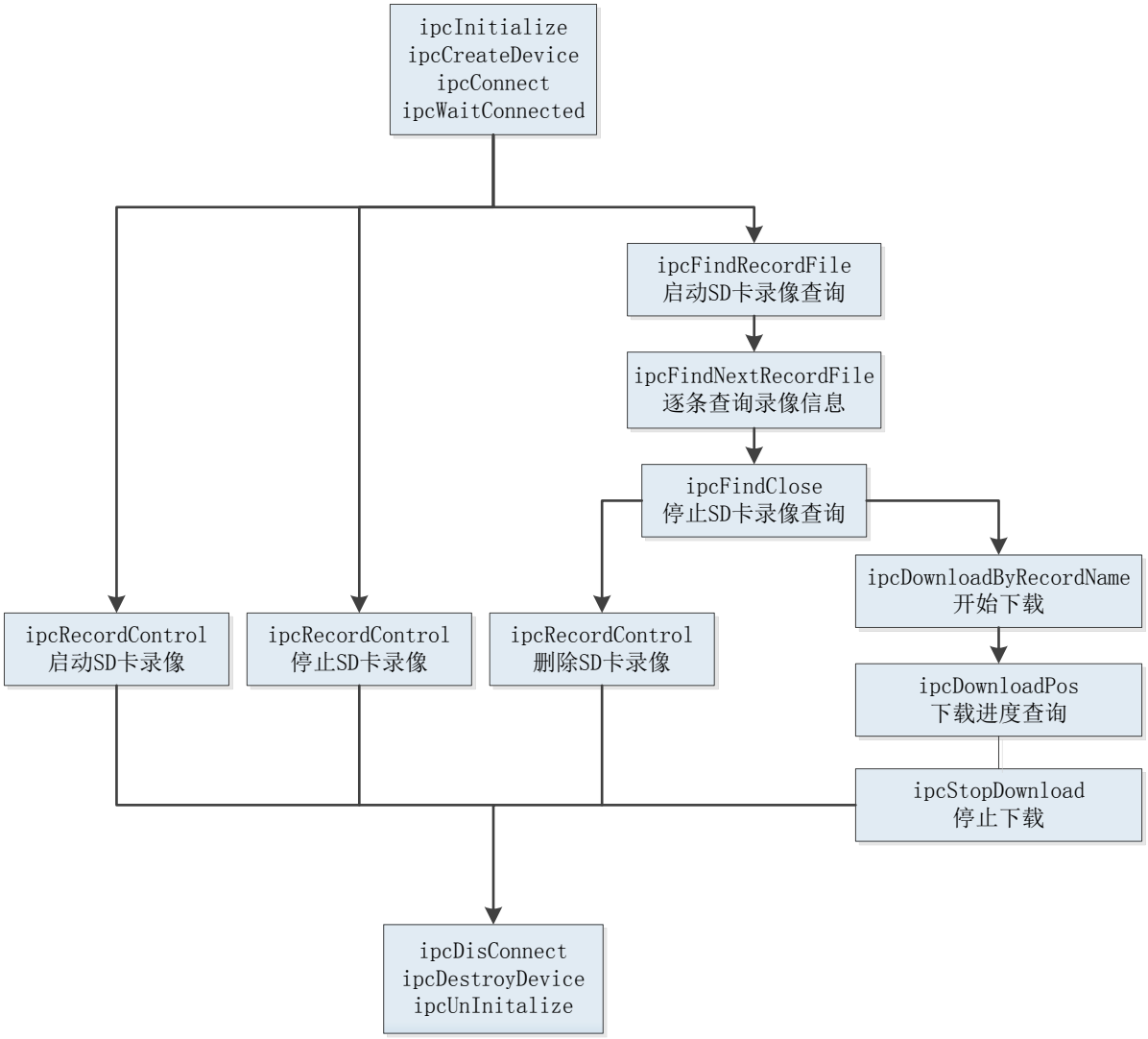

ipcInitialize
ipcCreateDevice
ipcConnect
ipcWaitConnected
ipcFindRecordFile
启动SD卡录像查询
ipcFindNextRecordFile
逐条查询录像信息
ipcFindClose
停止SD卡录像查询
ipcDownloadByRecordName
开始下载
ipcDownloadPos
下载进度查询
ipcStopDownload
停止下载
ipcRecordControl
启动SD卡录像
ipcRecordControl
停止SD卡录像
ipcRecordControl
删除SD卡录像
ipcDisConnect
ipcDestroyDevice
ipcUnInitalize

图 3-8

# 2结构与宏定义

## 2.1数据结构

### 2.1.1 实时码流

**ENCENC**

获取当前预览的码流信息

```
typedef   struct   tagENCENC
{
 DWORD enc_streamID ;
 DWORD enc_type;
 DWORD nc_fbs;
 DWORD enc_brcMode;
 DWORD enc_cbrAVG;
 DWORD enc_vbr_min;
 DWORD enc_vbr_max;
}ENCENC, *LPENCENC;
```

**Members**

enc_streamID

为当前正在预览的码流 ID，0 为码流 1，1 为码流 2，2 为码流 3，3 为码流 4

enc_type

编码格式： 1 为 H.264，2 为 MJPEG

enc_fbs

编码帧率

enc_brcMode

比特率控制，0 为 CBR，1 为 VBR，2 为 CBR (keep quality) ，3 为 VBR (keep quality)

enc_cbrAVG

比特率平均值，介于 1000000 到 8000000 之间

enc_vbr_min

最小比特率，默认为 1000000

enc_vbr_max

最大比特率，默认为 8000000

**Remarks**

编码帧率对应值： 60fps 为 8533333， 30fps 为 17066667，25fps 为 20480000

20fps 为 25600000，15fps 为 34133333，10fps 为 51200000，6fps 为 85333333

5fps 为 102400000，3fps 为 170666667，2fps 为 256000000，1fps 为 512000000

**See Also**

ipcGetConfig

**IPCVIDEO_CALLBACK_INFO_t**
视频单独回调的数据结构

```
typedef struct IPCVIDEO_CALLBACK_INFO_t
{
Long     lType ;
    Long     lWidth;
    Long     lHeight;
    Long     lFrameRate;
    const unsigned char *pBuf;
    Long     lBufSize;
}FrameInfo ,IPCVIDEO_CALLBACK_INFO,*LPIPCVIDEO_CALLBACK_INFO;
```

**Members**

lType
帧类型,I,P,B
lWidth
图像宽
lHeight
图像高
lFrameRate
帧率
pBuf
帧数据缓冲区
lBufSize
帧数据缓冲区大小

**See Also**

**ipcLiveFrame_CallBack**

## 2.1.2 视频管理

**OSDPARAM**
当前预览码流显示 OSD 信息

```
typedef struct tagOSDParam
{
 bool                          text ;
 long                          text_x;
 long                          text_y;
 long                          text_lWidth;
 long                          text_lHeight;
#ifdef UNICODE
 LOGFONTW              text_font ;
#else
 LOGFONTA              text_font ;
#endif
 COLORREF            text_crColor ;
```

```
 wchar_t                    text_context[256] ;

 bool                           picture;
 long                           pic_x;
 long                           pic_y;
 long                           pic_lWidth;
 long                           pic_lHeight;
 COLORREF            pic_crMask ;
 long                           pic_data_len ;
 BYTE*                      pic_data ;
}OSDPARAM,*LPOSDPARAM;
```

**Members**

text

是否显示文字信息，true 为显示, false 为不显示

text_x

文字显示区域的 x 坐标

text_y

文字显示区域的 y 坐标

text_lWidth

文字显示区域的宽

text_lHeight

文字显示区域的高

text_font

字体，标准的 **LOGFONT** 结构 ,SDK 自动适配用户应用程序的 UNICODE 和 MultiBytes 数据

text_crColor

字体颜色

text_context[256]

文本内容，小于 256 个字符

picture

是否显示图片，true 为显示，false 为不显示

pic_x

图片显示区域的 x 坐标

pic_y

图片显示区域的 y 坐标

pic_lWidth

图片显示区域的宽

pic_lHeight

图片显示区域的高

pic_crMask

图片底色

pic_data_len

图片数据长度

pic_data

图片数据

**Remarks**

文字信息和图片可以同时操作，也可以个别操作。

**See Also**

**ipcSetDisplayOSD**

**DEVICEOSD**

设备 OSD 信息

```
typedef struct tagDEVICEOSD
{
 BOOL       bShow;
 DWORD   streamID ;
 DWORD   x ;
 DWORD    y ;
 DWORD  textColor ;
DWORD   textBold ;
 DWORD   textSize ;
 CHAR   textContent[32];
}DEVICEOSD, *LPDEVICEOSD;
```

**Members**

bShow

是否显示文字信息，FALSE 为不显示，TRUE 为显示

streamID

码流 ID，0 为码流 1，1 为码流 2，2 为码流 3，3 为码流 4

x

显示区域起始 X 坐标[0 100]视频图像的左上角为原点、x 为在 X 轴方向上视频宽度的百分比

y

显示区域起始 Y 坐标[0 100]视频图像的左上角为原点、y 为在 Y 轴方向上视频高度的百分比

textColor

字体颜色[0 7] 0:黑色、1:红色、2:蓝色、3:绿色、4:黄色、5:洋红色、6:蓝绿色、7:白色

textBold

字体粗细 0：正常、1：粗体

textSize

字体大小,只能取 16,24,32,40,48

textContent

信息内容，最多 32 个字节

Remarks

See Also

ipcSetDeviceOSD, ipcGetDeviceOSD

DATETIMEOSD

时间 OSD 信息

```
typedef struct tagDATETIMEOSD
{
 BOOL       bShow;
 DWORD   streamID ;
 DWORD   x ;
 DWORD    y ;
 DWORD   timeColor ;
DWORD   timeBold ;
 DWORD   timeSize ;
}DATETIMEOSD, *LPDATETIMEOSD;
```

Members

bShow

是否显示时间信息，FALSE 为不显示，TRUE 为显示

streamID

码流 ID，0 为码流 1，1 为码流 2，2 为码流 3，3 为码流 4

x

显示区域起始 X 坐标[0 100]视频图像的左上角为原点、x 为在 X 轴方向上视频宽度的百分比

y

显示区域起始 Y 坐标[0 100]视频图像的左上角为原点、y 为在 Y 轴方向上视频高度的百分比

timeColor

字体颜色[0 7] 0:黑色、1:红色、2:蓝色、3:绿色、4:黄色、5:洋红色、6:蓝绿色、7:白色

timeBold

字体粗细 0：正常、1：粗体

timeSize

字体大小,只能取 16,24,32,40,48

Remarks

See Also

ipcSetDeviceDateTimeOSD, ipcGetDeviceDateTimeOSD

**IMGIRCUT**
IrCut昼夜模式

```
typedef struct tagIRCUT
{
 DWORD    ircut_DurTime ;
 DWORD    ircut_DayToNightThr ;
 DWORD    ircut_NightToDayThr ;
}IMGIRCUT, *LPIMGIRCUT;
```

**Members**

ircut_DurTime
等待时间，即每次作出日夜切换决策的时间间隔 3s ~ 30s  单位：秒

ircut_DayToNightThr
白天模式转夜晚模式的阈值 0~100  默认 40

ircut_NightToDayThr
夜晚模式转白天模式的阈值 0 ~ 100 默认 60

**Remarks**
**See Also**
**ipcGetConfig ,ipcSetConfig**

**IMGPRO**
图像属性信息

```
typedef struct tagIMGPRO
{
bool              img_bDefault;
DWORD     img_saturation;
LONG         img_brightness;
LONG         img_hue;
DWORD     img_contrast;
DWORD     img_sharpness;
}IMGPRO, *LPIMGPRO;
```

**Members**
img_bDefault
 是否恢复到默认值，true 为是，false 为不是
img_saturation
 饱和度，取值范围为 0 到 255
img_brightness

亮度，取值范围为-255 到 255

img_hue

色度，取值范围为-15 到 15

img_contrast

对比度，取值范围为 0 到 255

img_sharpness

锐度，取值范围为 0 到 255

**Remarks**

当 img_bDefault 为 true 时，是把图像属性参数恢复的默认值，故其他参数可以缺省。

**See Also**

**ipcGetConfig ,ipcSetConfig**

**IMGIRIS**

DC IRIS 模式

```
typedef struct tagIMGIRIS
{
 DWORD    iris_type;
 DWORD    iris_quty;
}IMGIRIS, *LPIMGIRIS;
```

**Members**

iris_type

模式 0:关闭 1:开启

iris_quty

占空值 100 ~ 999

**Remarks**

**See Also**

**ipcGetConfig ,ipcSetConfig**

**IMGSHUTTER**

电子快门时间范畴

```
typedef struct tagIMGSHUTTER
{
 DWORD     shutter_min;
 DWORD     shutter_max;
}IMGSHUTTER, *LPIMGSHUTTER;
```

**Members**

shutter_min

电子快门时间的最小值，其取值范围需通过 ipcGetShutterTimeMinRange 获取

shutter_max

电子快门时间的最大值，其取值范围需通过 ipcGetShutterTimeMaxRange 获取

**Remarks**

**See Also**

**ipcGetConfig ,ipcSetConfig**

**BitRate**

编码信息

```
typedef struct _tagBitRate
{
 DWORD   sn_brc;
 DWORD   sn_cbr;
 DWORD   sn_vbr_min;
 DWORD   sn_vbr_max;
}BitRate;
```

**Members**

sn_brc

比特率控制，0 为 CBR，1 为 VBR，2 为 CBR (keep quality) ，3 为 VBR (keep quality)

sn_cbr

比特率平均值，介于到之间 1000000~8000000  sn_brc 为 0 或 2 时有效

sn_vbr_min

最小比特率（bps），默认为 1000000  sn_brc 为 1 或 3 时有效

sn_vbr_max

最大比特率（bps），默认为 8000000  sn_brc 为 1 或 3 时有效

**Remarks**

该编码参数修改后，已经建立的视频连接在接收到断线通知后需要重新请求视频

**See Also**

**ipcGetBitRate ,ipcSetBitRate**

**IMGPM**

视频遮挡

```
typedef struct tagIMGPM
{
 INT             pm_index;
 INT             enable_flag;
 DWORD     pm_left;
 DWORD     pm_top;
 DWORD     pm_w;
 DWORD     pm_h;
 DWORD     pm_color;
}IMGPM, *LPIMGPM;
```

**Members**

pm_index

索引，取值范围为 1 到 4

enable_flag

true 为启用，false 为删除

pm_left

取值范围 0~99

pm_top

取值范围 0~99

pm_w

取值范围 1~100

pm_h

取值范围 1~100

pm_color

区域颜色 0:黑色 1:红色 2:蓝色 3:绿色 4:黄色 5:洋红 6:青绿 7:白色

**Remarks**

视频图像被分成 16x16 大小的若干个格子，具体格子个数由当前图像分辨率决定 （例如 1280X720 的分辨率，可划分为 80X45 个方格）。pm_left, pm_top 可确定左上角，pm_w, pm_h 可以确定右下角。右下角的坐标一定要大于左上角的坐标

例如：{ pm_left =0，pm_top =0 ， pm_w =1，pm_h=1} 表示占用(0,0),(0,1),(1,0),(1,1)四个格子

目前支持 4 个视频遮挡区域

**See Also**

**ipcSetPictMask , ipcGetPictMask**

**SNTEXT**

显示信息叠加

```
typedef struct tagSNTEXT
```

```
{
 DWORD     text_streamID ;
 BOOL         text_enable;
 CHAR         text_content[32];
}SNTEXT, *LPSNTEXT;
```

**Members**

text_streamID

码流 ID，0 为码流 1，1 为码流 2，2 为码流 3，3 为码流 4

text_enable

是否显示文字信息，0 为不显示， 1 为显示

text_content

文字信息内容，最多 31 个字符

**Remarks**

**See Also**

**ipcGetConfig ,ipcSetConfig**

**StreamInfo**

设备码流编码信息

```
typedef struct _tagStreamInfo
{
 INT    iWidth;
 INT    iHeight;
 INT    iRate;
 INT    iEncType;
 long   lAudioFormat;
}StreamInfo;
```

**Members**

iWidth

图像宽度

iHeight

图像高度

iRate

图像帧率

iEncType

图像分编码类型 1:H264,2:MJPEG

lAudioFormat

音频编码格式 0:AMR; 1:G711

格式固定为:AMR 为 SamplesPerSec:8000,BitsPerSample:16,Channels:1,AvgBytesPerSec:16000

G711 为 SamplesPerSec:8000,BitsPerSample:8,Channels:1,AvgBytesPerSec:8000

Remarks

See Also

ipcGetStreamInfo

DeviceImgInfo

设备当前各个码流信息

```
typedef struct _tagDeviceImgInfo
{
 INT             iStreamCount;
 StreamInfo  stStreamInfo[4];
}DeviceImgInfo;
```

Members

iStreamCount

码流个数 1~4

iHeight

对应码流 id 0~3 的码流信息

Remarks

See Also

ipcGetCurAllStreamInfo,ipcSetCurAllStreamInfo, StreamInfo

IPC_MP4_VIDEOPARAM

MP4 文件中视频流数据结构

```
typedef   struct  IPC_MP4_VIDEOPARAM_t
{
    int                dwHeight;
    int                dwWidth;
    double          dwFrameInterval;
    int                dwBitRate;
    int                wAspectRatio;
    int                dwParWidth;
    int                dwParHeight;
    int                cbExtraInfo;
    void *           pExtraInfo;
} IPC_MP4_VIDEOPARAM,*LPIPC_MP4_VIDEOPARAM;
```

Members

dwHeight

高

dwWidth

宽

dwFrameInterval

帧间隔（单位：100 纳秒）

计算方法（如：10000000.0 / FrameRate）

dwBitRate

码率

wAspectRatio

比例

dwParWidth

保留默认为 0

dwParHeight

保留默认为 0

cbExtraInfo

保留默认为 0

pExtraInfo

保留默认为 0

**See Also**

**ipcMP4AddVideoStream**

**IPC_MP4_AUDIOPARAM**

MP4 文件中语音流数据结构

```
typedef struct IPC_MP4_AUDIOPARAM_t
{
    int          dwChannel;
    int          dwSamplePerSec;
    int          dwBitsPerSample ;
    int          nAvgBytesPerSec;
    int          nBlockAlign;
    int          cbExtraInfo;
    void*    pExtraInfo;
} IPC_MP4_AUDIOPARAM,*LPIPC_MP4_AUDIOPARAM;
```

**Members**

dwChannel

声道

dwSamplePerSec

采样频率

dwBitsPerSample

采样精度

nAvgBytesPerSec

平均码率

nBlockAlign

块对齐方式 保留默认为 0

cbExtraInfo

扩展数据长度 保留默认为 0

pExtraInfo

扩展数据 保留默认为0

See Also

ipcMP4AddAudioStream

IPC_STREAM_HEADER

音视频混合流回调流头数据结构

```
typedef struct
{
    unsigned long  ulFlag;
    unsigned long   ulVersion;
    long           lType;
    long           lSize;
    long           lHeaderSize;
    long           lVideoForamt;
    long           lWidth;
    long           lHeight;
    long           lFrameRate;
    long           lAudioForamt;
    long           lSamplesPerSec;
    long           lBitsPerSample;
    long           lChannels;
    long           lAvgBytesPerSec;
    unsigned char  Reserve[32];
} IPC_STREAM_HEADER,*PIPC_STREAM_HEADER
```

Members

ulFlag
帧头标识,固定值:'SIMG'

ulVersion
版本号10000

lType
帧类型,视频: 'I','P','B',音频:'A',流头:'H'

lSize
帧长度: 帧头长度+ 帧数据长度

lHeaderSize
帧头长度:sizeof(IPC_STREAM_HEADER)

lVideoForamt
视频编码方式. 0:H264 ,1:MJPEG

lWidth
视频宽

lHeight
视频高

lFrameRate
视频帧率

lAudioForamt
音频编码格式0:AMR; 1:G711

lSamplesPerSec
采样频率:8000

lBitsPerSample
采样精度:amr为16,711为8

lChannels
声道:1

lAvgBytesPerSec
平均速率:amr为16000,711为8000

Reserve
保留字段32字节

## See Also

ipcAVStreamCallBackEx2

## IPC_VIDEOFRAME_HEADER

音视频混合流回调视频头数据结构

```
typedef struct
{
    unsigned long   ulFlag;
    long             lType;
    long            lSize;
    long            lHeaderSize;
    long            lTailSize;
    ULONGLONG       ullTimeStamp;
    long            lForamt;
    long             lWidth;
    long            lHeight;
    long            lFrameRate;
    unsigned char   Reserve[12];
} IPC_VIDEOFRAME_HEADER,*PIPC_VIDEOFRAME_HEADER;
```

## Members

ulFlag
帧头标识,固定值:'FIMG'

lType
帧类型,视频: 'I','P','B',音频:'A',流头:'H'

lSize
帧长度: 帧头长度+ 帧数据长度+ 帧尾长度

lHeaderSize

帧头长度:sizeof(IPC_VIDEOFRAME_HEADER)

lTailSize
帧尾长度:sizeof(IPC_FRAME_TAIL)

ullTimeStamp
时间戳

lForamt
视频编码方式. 0:H264 ,1:MJPEG

lWidth
视频宽

lHeight
视频高

lFrameRate
视频帧率

Reserve
保留12字节

## See Also

ipcAVStreamCallBackEx2

## IPC_AUDIOFRAME_HEADER

音视频混合流回调音频头数据结构

```
typedef struct
{
    unsigned long   ulFlag;
    long            lType;
    long            lSize;
    long            lHeaderSize;
    long            lTailSize;
    ULONGLONG       ullTimeStamp;
    long            lForamt;
    long            lSamplesPerSec;
    long            lBitsPerSample;
    long            lChannels;
    long            lAvgBytesPerSec;
    unsigned char   Reserve[8];
} IPC_AUDIOFRAME_HEADER,*PIPC_AUDIOFRAME_HEADER;
```

## Members

ulFlag
帧头标识,固定值:'FIMG'

lType
帧类型,视频: 'I','P','B',音频:'A',流头:'H'

lSize
帧长度: 帧头长度+ 帧数据长度+ 帧尾长度

lHeaderSize
帧头长度:sizeof(IPC_AUDIOFRAME_HEADER)

lTailSize
帧尾长度:sizeof(IPC_FRAME_TAIL)

ullTimeStamp
时间戳

lForamt
音频编码格式0:AMR; 1:G711

lSamplesPerSec
采样频率:8000

lBitsPerSample
采样精度:amr为16,711为8

lChannels
声道:1

lAvgBytesPerSec
平均速率:amr为16000,711为8000

Reserve[8]
保留 8 字节

## See Also

ipcAVStreamCallBackEx2

### 2.1.3 网络管理

SYSSMTP

邮件传输协议设置

```
typedef   struct  tagSYSSMTP
{
 DWORD    smtp_enable;
 CHAR        smtp_addr[32];
 DWORD    smtp_port;
 DWORD    smtp_auth_enable;
 CHAR        smtp_auth_name[32];
 CHAR        smtp_auth_password[32];
 DWORD    smtp_auth_model;
 CHAR        smtp_sender[64];
 CHAR        smtp_receiver[64];
 CHAR        smtp_cc[64];
 CHAR        smtp_subject[64];
 CHAR        smtp_content[128];
}SYSSMTP, *LPSYSSMTP;
```

Members

smtp_enable

是否配置 SMTP, 1:需要配置，0:不需要配置

smtp_addr

smtp 邮件服务器地址，格式类似于 192.168.1.114

smtp_port

smtp 端口号 0 到 65535

smtp_auth_enable

是否使用权限认证登陆 1:使用， 0:不使用，则不需要填写 smtp_auth_name 和 smtp_auth_password

smtp_auth_name

用户名

smtp_auth_password

密码

smtp_auth_model

验证方式，0:LOGIN  1:PLAIN

smtp_sender

发件人地址，正确邮箱地址

smtp_receiver

收件人地址，正确邮箱地址

smtp_cc

抄送地址， 正确邮箱地址

smtp_subject

邮件主题

smtp_content

邮件内容，文本字符

**Remarks**

**See Also**

ipcGetSystemSetting,ipcSetSystemSetting, ipcSetDeviceSMTPInfo, ipcGetDeviceSMTPInfo

**SYSFTP**

文件传输协议设置

```
typedef   struct   tagSYSFTP
{
 DWORD    ftp_enable;
 CHAR        ftp_addr[32];
 DWORD    ftp_port;
 CHAR        ftp_user[32];
 CHAR        ftp_password[32];
}SYSFTP, *LPSYSFTP;
```

**Members**

ftp_enable

是否开启 ftp， 1:开启，0:关闭

ftp_addr

ftp 服务器地址，格式类似于 192.168.1.114

ftp_port

ftp 端口

ftp_user

用户名

ftp_password

密码

**Remarks**

**See Also**

ipcGetSystemSetting,ipcSetSystemSetting, ipcSetDeviceFTPInfo, ipcGetDeviceFTPInfo

**SYSIPINFO**

ipv4 地址配置

```
typedef   struct   tagSYSIPINFO
{
```

```
 DWORD     dhcp_enable;
 CHAR          ip_addr[32];
 CHAR          network_mask[32];
 CHAR          gateway_addr[32];
 CHAR          dns_addr[32];
 CHAR          dns_backup_addr[32];
 CHAR          mac_addr[32] ;
}SYSIPINFO, *LPSYSIPINFO;
```

**Members**

dhcp_enable

配置 ip 地址方式，1:采用 dhcp，0:手动配置 ip

ip_addr

ip 地址

network_mask

子网掩码

gateway_addr

默认网关

dns_addr

首选 DNS 服务器地址

dns_backup_addr

备用 DNS 服务器地址

mac_addr

mac 地址 （只读），不可设置

**Remarks**

**See Also**

ipcGetSystemSetting,ipcSetSystemSetting, ipcSetDeviceIPInfo. ipcGetDeviceIPInfo

**SYSUPNPINFO**

UPNP 设置

```
typedef   struct   tagSYSUPNPINFO
{
 DWORD      enable;
 DWORD      port;
 CHAR          name;
}SYSUPNPINFO,*LPSYSUPNPINFO;
```

**Members**

enable

是否可用，true 为可用，false 为不可用

port
端口号
name
名称

**Remarks**

**See Also**

**ipcGetSystemSetting,ipcSetSystemSetting, ipcSetDeviceUPNPInfo, ipcGetDeviceUPNPInfo**

**DDNSPARAM**

DDNS 设置

```
typedef struct _tagDDNSPARA
{
 INT       iEnableDDNS;
 INT       iDDNSType;
 INT       iInterval;
 CHAR   sUserName[64];
 CHAR   sPassword[64];
 CHAR   sDomainName[64];
 CHAR    sServerName[64];
 INT        iDDNSPort;
}DDNSPARA,*LPDDNSPARA;
```

**Members**

iEnableDDNS
是否使能：0－否，1－是
iDDNSType
0－Dyndns DNS，1---Private，2－PeanutHull(花生壳)，3-NO-IP 目前仅仅支持 1
iInterval
上报时间间隔，单位秒(10~7200)
sUserName
DDNS 账号用户名 暂时未使用
sPassword
DDNS 密码 暂时未使用
sDomainName
设备域名
sServerName
DDNS 对应的服务器地址，可以是 IP 地址或域名
iDDNSPort
DDNS 服务器端口

**Remarks**

See Also

ipcSetDDNSParam, ipcGetDDNSParam

DDNSRecord

DDNS信息

```
typedef struct _tagDDNSRecord
{
 CHAR szDomain[IPC_MAX_NAME_LEN];
 CHAR szSN[IPC_MAX_NAME_LEN];
 CHAR szIp[IPC_MAX_NAME_LEN];
 CHAR szName[IPC_MAX_NAME_LEN];
 INT      iPort;
}DDNSRecord;
```

Members

szDomain

域名

szSN

序列号

szIp

设备 IP 信息

szName

设备名称

iPort

dnns 服务登录端口

Remarks

See Also

ipcDDNSCallBack  ipcSetDDNSServerCallBack ipcStartDdnsListen ipcStopDdnsListen

ImgSize

分辨率

```
typedef enum  _tagImgSize
{
 SIZE_1080P,
 SIZE_720P,
 SIZE_576P,
 SIZE_480P,
 SIZE_288,
 SIZE_240,
```

```
}ImgSize;
```

**ImgInfo**

视频图像信息

```
typedef struct _tagImgInfo
{
 ImgSize         stSize;
 INT               iRate;
}ImgInfo;
```

**Members**

stSize

分辨率

iRate

帧率

**Remarks**

**See Also**

AllImgInfo

**AllImgInfo**

与码流个数对应的全部图像分辨率帧率信息

```
typedef struct _tagAllImgInfo
{
 INT                  iCount;
 ImgInfo          stImgInfo[IPC_MAX_IMGINFO_COUNT][4];
}AllImgInfo;
```

**Members**

iCount

分辨率帧率信息个数

stImgInfo

列对应码流id 1~4支持的图像分辨率和帧率,

**Remarks**

**See Also**

ipcGetSupportedImgInfo

**设备状态定义**

DEVICESTATE

```
typedef enum tagDEVICESTATE
{
    ds_online,
    ds_offline,
    ds_busy,
    ds_idle,
    ds_unknow
}DEVICESTATE,*LPDEVICESTATE;
```

Members

ds_online,
在线
ds_offline
离线
ds_busy
忙碌
ds_idle
休眠
ds_unknow
无法探知

Remark

See Also

IPCSCANINFO

设备信息

```
typedef struct tagIPCSCANINFO
{
    DEVICESTATE state ;
    int  port ;
    char mac[32] ;
    char IPAddress[32] ;
    char PrimaryDns[32] ;
    char SecondaryDns[32] ;
    char Netmask[32] ;
    char Gateway[32] ;
}IPCSCANINFO,*LPIPCSCANINFO ;
```

Members

State

设备当前状态

Mac

mac 地址

IPAddress

ip 地址

PrimaryDns

主 DNS

SecondaryDns

备选 DNS

Netmask

掩码

Gateway

网关

**Remark**

**See Also**

**ipcScanOnlineDeviceCallBack**

**IPC_COMM_PROP**

串口配置信息

```
typedef struct COMM_PROP_t{
BYTE byDataBit;
BYTE byStopBit;
BYTE byParity;
BYTE byBaudRate;
} IPC_COMM_PROP,*LPIPC_COMM_PROP ;
```

**Members**

byDataBit

数据位[7,8]

byStopBit

停止位[1,2]

byParity

校验位[0,2]：PARITY_NONE = 0, PARITY_ODD = 1, PARITY_EVEN = 2,

byBaudRate

波特率[0,9]：=300,1=1200,2=2400,3=4800,4=9600,5=19200,6=38400,7=115200,8=460800,9=921600

**Remarks**

**See Also**

**ipcGetPTZTransProp, ipcSetPTZTransProp**

### 2.1.4 云台管理

**PRESET_CRUISE_POINT_t**

巡航预置点

```
typedef struct
{
 INT     iPresetIdx;
 INT     iStaySec;
 INT     iMoveSpeed;
} PRESET_CRUISE_POINT_t;
```

**Members**

iPresetIdx

预置点索引,范围: 1 - 255

iStaySec

停留时间 单位秒 大于 0

iMoveSpeed

转到该预置点的速度 大于 0

**Remarks**

**See Also**

PTZCruiseParam

**PTZCruiseParam**

巡航参数

```
 struct PTZCruiseParam
{
 INT  id;
 CHAR        name[IPC_MAX_NAME_LEN];
 INT             iEnable;
 DWORD     iCount ;
 PRESET_CRUISE_POINT_t point[IPC_MAX_PRESET_CRUISE_POINTS];
};
```

**Members**

id

巡航路径 id, 范围: 1 - 32

name

巡航路径名称 ,最大长度 64

iEnable

巡航路径使能标志（1 表示 启用 ， 0 表示禁用）

iCount

巡航路径里巡航点个数

point

巡航预置点 ， 最大 32

**Remarks**

**See Also**

ipcGetPTZCruise, ipcSetPTZCruise

IPC_PRESET
预置位信息

```
 typedef struct PRESET_t
{
    DWORD index ;
    DWORD enable ;
    CHAR  name[256] ;
}IPC_PRESET ,*LPIPC_PRESET ;
```

Members

index

索引

enable

开关

name

名称

Remarks

See Also

ipcPTZGetAllPreset

IPC_PTZ_POSITION_PARAM

机械云台坐标信息，含 z 坐标

```
typedef   struct   tagIPC_PTZPositionParam
{
INT        xPostion ;
INT        yPostion ;
INT        zPostion ;
INT       xSpeed ;
INT        ySpeed ;
}IPC_PTZ_POSITION_PARAM,* LPIPC_PTZ_POSITION_PARAM;
```

Members

xPosition

水平方向 坐标:  取值范围受 ipcPTZPositionControl 函数的命令宏影响 ：

命令为 IPC_CMD_PTZ_GOTO_REL_POSITION 时候,取值范围-35999 到 35999

命令为 IPC_CMD_PTZ_GOTO_ABS_POSITION 时候,取值范围  0 到 35999

yPosition

垂直方向坐标:  取值范围受 ipcPTZPostionControl 函数的命令宏影响 ：

命令为 IPC_CMD_PTZ_GOTO_REL_POSITION 时候,取值范围-9000 到 9000

命令为 IPC_CMD_PTZ_GOTO_ABS_POSITION 时候,取值范围 0 到 9000

zPosition

放大倍数:  应大于等于 0 且是 10 的整数倍,0 表示原始大小即放大 1 倍,10 表示原始大小放大 2 倍,20 表示放大原始大小 3 倍,zPosition/10+1 为放大的倍数,zPosition 的最大值 180（该值由硬件设备决定）,传递的值超过设备支持的最大值时，设备自动设置为最大值。

xSpeed

水平方向移动的速度，取值范围为 0 到 100

ySpeed

垂直方向移动的速度，取值范围为 0 到 100

Remarks

See Also

ipcPTZPositionControl

IPC_POINT_FRAME

机械云台区域定位

```
typedef struct  tagIPC_POINT_FRAME
{
        INT     xTop ;
        INT     yTop ;
        INT     xBottom ;
        INT     yBottom ;
        INT     iType ;
}IPC_POINT_FRAME,*LPIPC_POINT_FRAME;
```

Members

xTop

水平起始坐标

yTop

垂直起始坐标:

xBottom

水平结束坐标

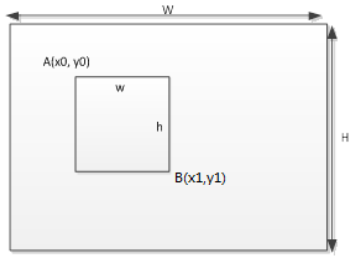
yBottom

垂直结束坐标

iType

画框方式:0-框选 1-滚轮缩小 2-滚轮放大（当类型为 1 和 2 时坐标点无效）

Remarks




A(x0,y0)表示开始点，B(x1,y1)表示结束点
w 表示框选的宽度，W 表示视频画面的宽度
h 表示框选的高度，H 表示视频画面的高度

x、y 是归一化的值

A(x,y)的计算方法为：

x = (x0 / W) * 8192

y = (y0 / H) * 8192

B(x,y) 的计算方法为：

x = (x1 / W) * 8192

y = (y1 / H) * 8192

单击的时候 A 点与 B 点重合，即 w 和 h 均为 0

正选时，w、 h 大于 0，图像放大；反选时，w、 h 小于 0，图像缩小。

**See Also**

ipcPTZSelZoomIn

### 2.1.5 事件管理

**RECORDACTION**

录像动作信息

```
typedef struct tagRecordAction
{
INT              enable_flag;
DWORD      delay;
DWORD      streamID;
DWORD      locationID;
}RECORDACTION, *LPRECORDACTION;
```

**Members**

enable_flag

0 为删除，1 为有效

delay

录像持续时间动作，以秒为单位，参考 ActionOptionInfo 中的响应方式，有效期响应方式时该参数有效

streamID

0:码流 1， 1:码流 2 ，2:码流 3 ，3:码流 4

locationID

文件保存类型，有 IPC_RECORD_FILELOCATION_FTP，IPC_RECORD_FILELOCATION_SDCARD

**Remarks**

**See Also**

ActionOptionInfo

DPTZACTION

数字云台动作

```
typedef struct tagDPTZAction
{
INT              enable_flag;
DWORD      delay;
DWORD      streamID;
DWORD      preset;
}DPTZACTION, *LPDPTZACTION;
```

Members

enable_flag

0 为删除， 1 为启用

delay

转移到预置位后停留时间，以秒为单位，参考 ActionOptionInfo 中的响应方式，有效期响应方式时该参数有效

streamID

0:码流 1 ,1:码流 2, 2:码流 3, 3:码流 4

preset

预置位标识号,取值范围为 1 到 9

Remarks

See Also

ActionOptionInfo

PTZACTION

机械云台动作

```
typedef struct tagPTZAction
{
 INT              enable_flag;
 DWORD      delay;
 DWORD      preset;
}PTZACTION, *LPPTZACTION;
```

Members

enable_flag

0 为删除， 1 为启用

delay

转移到预置位后停留时间，以秒为单位，参考 ActionOptionInfo 中的响应方式，有效期响应方式时该参数有效

preset

预置位标识号,取值范围为 1 到 255

**Remarks**

**See Also**

ActionOptionInfo

**OUTPUTACTION**

输出端口动作

```
typedef struct tagOutputAction
{
 INT              enable_flag;
 DWORD      level;
}OUTPUTACTION, *LPOUTPUTACTION;
```

**Members**

enable_flag

0 为删除， 1 为启用

level

保留参数，高低电平： 0 为低电平， 1 为高电平

**Remarks**

**See Also**

ActionOptionInfo

**ActionOptionInfo**

响应动作信息

```
typedef  struct  tagActionOptionInfo
{
DWORD                        priority;
DWORD                        respond_option;
RECORDACTION      record ;
DPTZACTION            dptz
PTZACTION               ptz
OUTPUTACTION      output
} ActionOptionInfo,*LPActionOptionInfo ;
```

**Members**

priority

动作的优先级 0: Low, 1: Middle, 2: High

respond_option

响应方式，0：报警持续期间执行动作，1：报警发生和结束时执行动作(暂不支持)，2：报警开始时执行动作（暂不支持），3：报警结束时执行动作（暂不支持）

record

录像动作，见 RECORDACTION 数据结构

dptz

数字云台动作，见 DPTZACTION 数据结构

ptz

机械云台动作，见 PTZACTION 数据结构

output

报警输出端口动作，见 OUTPUTACTION 数据结构

**Remarks**

**See Also**

ipcGetCameraUnusualEvent,ipcSetCameraUnusualEvent,ipcGetMotionEventAction, ipcSetMotionEventAction, ipcGetInputEvent, ipcSetInputEvent

**MotionDetectionWindowInfoEx**

移动侦测窗口信息

```
typedef struct tagMotionDetectionWindowInfoEx
{
Int             index;
Int              enable;
Char          name[128] ;
RECT       rect ;
INT            agile;
INT            threshold;
}MotionDetectionWindowInfoEx,*LPMotionDetectionWindowInfoEx ;
```

**Members**

Index

窗口 id 范围 1~4

enable

侦测区域使能标志:0 禁用,1 启用

name

窗口名称

rect

窗口区间，为 RECT 结构体

视频图像被分成 12x8=96 个格子 。x,y 可确定左上角，right,bottom 可以确定右下角。右下角的坐标一定要大于左上角的坐标

x 取值范围[0, 11].

y 取值范围 [0, 7].

right 取值范围 [1, 12].

bottom 取值范围 [1, 8].

例如： rect 为{x=1，y=1 ，righgt=2，bottom=2} 表示占用(1,1),(1,2),(2,1),(2,2)四个格子

agile

灵敏度 范围[0, 10],值越大越灵敏，建议值 3~5,

threshold

阈值 范围[1, 100],值越小越容易触发事件，建议值 25~30

**Remarks**

**See Also**

**ipcGetMotionWndInfo, ipcSetMotionWndInfo**

**IPC_ALARMCENTER_UP_CFG**

报警上传中心

```
struct IPC_ALARMCENTER_UP_CFG{
  BYTE           byEnable[IPC_MAX_CENTER];
  WORD         wHostPort[IPC_MAX_CENTER];
  CHAR          sHostIPAddr[IPC_MAX_CENTER][IPC_MAX_IPADDR_LEN];
  BYTE           dwReserved[1];
};
```

**Members**

byEnable

上传使能 ，范围个数3

bReserverd

保留，范围个数3

wHostPort

报警中心侦听端口

sHostIPAddr

报警中心 IP

**Remarks**

**See Also**

**ipcGetSystemSetting,ipcSetSystemSetting**

**ALARMER**

报警信息源

```
struct ALARMER
```

```
{
 CHAR szIp[64];
 CHAR szDevName[64];
};
```

**Members**

szIp

报警设备ip

szDevName

设备名称

**Remarks**

**See Also**

AlarmCallBack

**ALARMINFO**

报警信息

```
typedef struct _tagAlarmInfo
{
 INT     idHeight;
 INT     idLow;
 INT     iLevel;
 INT     iState;
 CHAR     tmStamp[32];
 CHAR    szDescript[64];
 CHAR    szReserved[64];
 }ALARMINFO,*LPALARMINFO;
```

**Members**

idHeight

报警消息 id（64 位整数）高 32 位数据，该 id 为报警信息数据库的行记录 id，可以唯一地代表一条报警信息

idLow

报警消息 id（64 位整数）低位 32 数据，该 id 为报警信息数据库的行记录 id，可以唯一地代表一条报警信息

iLevel

报警等级(0,1,2….) ,0 等级最高,数据越大等级越低

iState

1 报警开始 0 报警结束

tmStamp

时间戳格式 YYYYMMDDHHMMSS;如：20120802235502

szDescript

报警信息描述

szReserved

保留数据,外部输入报警时候可以传递端口号，部分报警可以描述一个字符串

**Remarks**

**See Also**

AlarmCallBack

### 2.1.6 系统管理

**SYSDATEINFO**

系统时间设置

```
typedef struct tagSYSDATEINFO
{
 DWORD       sync_mode;
 CHAR           time_date[32];
 CHAR           time_time[32];
 CHAR           ntp_addr[32];
}SYSDATEINFO, *LPSYSDATEINFO;
```

**Members**

sync_mode

设备时间同步类型（只写），0:与 NTP 同步，ntp_addr 不能为空。1:手动设置， time_date 和 time_time 不能为空。2:与 pc 时间同步，time_date 和 time_time 不能为空

time_date

日期

time_time

时间

ntp_addr

ntp 服务器地址 （只写）

**Remarks**

**See Also**

ipcGetSystemSetting,ipcSetSystemSetting

**SYSDATEINFOEX**

系统时间设置扩展

```
typedef struct tagSYSDATEINFOEX
{
 DWORD       sync_mode;
 CHAR           time_date[32];
 CHAR           time_time[32];
INT                intval;
CHAR           timeZone[32];
 CHAR           ntp_addr[32];
}SYSDATEINFO, *LPSYSDATEINFO;
```

**Members**

sync_mode

设备时间同步类型（只写），0:与 NTP 同步，ntp_addr 和 timeZone 不能为空。1:手动设置， time_date 和 time_time 、timeZone 不能为空。2:与 pc 时间同步，time_date 和 time_time、timeZone 不能为空

time_date

日期

time_time

时间

intval

时间间隔，单位分钟，取值范围-100000

timeZone

时区，格式为 CST+HH:MM:SS 或 CST-HH:MM:SS，如东八区为 CST+08:00:00，西八区为 CST-08:00:00

ntp_addr

ntp 服务器地址 （只写）

**Remarks**

See Also

ipcGetSystemSetting,ipcSetSystemSetting

### 2.1.7 语音对讲

语音对讲配置数据结构

```
typedef struct
{
  int audio_enable;
  int codec_fmt;
  int audio_out_enable;
  int audio_out_vol;
  int audio_in_vol;
  int echo_canceller_enable;
}AUDIOCONF_t,*LPAUDIOCONF;
```

Members

audio_enable;
是否启用语音对讲功能 0:disable1:enable
codec_fmt
0:amr ,1:g711 目前暂只支持 711
audio_out_enable
允许设备输出 0:disable1:enable
audio_out_vol
设备输出音量 参数范围：eg:(50)50%
audio_in_vol
设备输入音量 参数范围：eg:(50)50%
echo_canceller_enable
回音消除 0:disable1:enable

Remarks

See Also

ipcAudioSetting,ipcSetSystemSetting

AudioDevice

声卡设备信息

```
struct AudioDevice
{
   INT   iId;
   CHAR  sDevName[128];
};
```

Members

iId

声卡设备 id

sDevName

声卡设备名字

Remarks

See Also

ipcEnumAudioDevice,ipcSetAudioDevice,ipcGetAudioDevice

IPC_CALLBACK_INFO

音视频复合流数据结构

```
typedef struct IPC_CALLBACK_INFO_t
{
  UINT32                                lType ;
  const unsigned char*        pBuf;
    UINT32                              lBufSize;
    LONGLONG                    lStamp;
    union {
        struct Video{
            UINT32                      lVideoFormat ;
            UINT32                      lWidth;
            UINT32                      lHeight;
            UINT32                      lFrameRate;
        } video ;

        struct Audio{
            UINT32                       lAudioFormat ;
            UINT32                       lSamplesPerSec ;
            UINT32                       lChannels ;
            UINT32                       lBitsPerSample ;
            UINT32                       lAvgBytesPerSec ;
        } audio ;
    } info ;
} FrameInfoEx ,IPC_CALLBACK_INFO,*LPIPC_CALLBACK_INFO;
```

Members

lType

帧类型,<video:'I','P','B'><audio:'A'>

pBuf
帧数据缓冲区
lBufSize
帧数据缓冲区大小
lStamp
时间戳<保留>
lVideoFormat
类型 0:H264, 1:MJPEG
lWidth
图像宽
lHeight
图像高
lFrameRate
帧率
lAudioFormat
类型 0:AMR, 1:G711 。目前暂只支持 G711
lSamplesPerSec
采用率 8000
lChannels
通道数 1
lBitsPerSample
位率 AMR 为 16 ， G711 为 8
lAvgBytesPerSec
码率 AMR 为 16000，G711 为 8000

**See Also**

ipcLiveFrame_CallBackEx

### 2.1.8 日志

**IPC_TIME**

日志时间

```
typedef struct tagIPC_TIME
{
  DWORD dwYear;
  DWORD dwMonth;
  DWORD dwDay;
  DWORD dwHour;
  DWORD dwMinute;
  DWORD dwSecond;
  DWORD dwRes;
 }IPC_TIME, *LPIPC_TIME;
```

**Members**

dwYear
  年
dwMonth
  月
dwDay
   日
dwHour
    时
dwMinute
  分
dwSecond
  秒
dwRes
  保留

**IPC_LOG_CONFIG**

日志配置

```
typedef struct tagIPC_LOG_CONFIG
{
    DWORD dwType;
    DWORD dwMaxValue;
}IPC_LOG_CONFIG, *LPIPC_LOG_CONFIG;
```

**Members**

dwType

日志保存策略类型：0 限制最多保存条数，1 限制最多保存天数

dwMaxValue

与保存策略对应： 当dwType为0时取值范围为 1-30000，当dwType为1时取值范围为 1-365.

**IPC_LOG**

日志信息结构体

```
typedef struct _tagIPC_LOG
{
  DWORD             dwId;
  IPC_TIME          stLogTime;
  DWORD             dwType;
  CHAR              sUser[IPC_MAX_USER_NAME_LEN];
  CHAR              sSummary[IPC_MAX_SUMMARY_LEN]
  CHAR              sDescription[IPC_MAX_DESCRIPTION_LEN];
}IPC_LOG,*LPIPC_LOG;
```

**Members**

dwId

日志id

stLogTime

日志时间

dwType

日志类型,见日志宏定义：IPC_LOG_ACCESS，IPC_LOG_ALARM，IPC_LOG_SYSTEM

sUser

用户名，GBK编码

sSummary

日志概述，GBK编码

sDescription

日志详细描述，GBK编码

### 2.1.9 SD卡录像

RECORDDOWNLOADINFO

断点续传结构

```
typedef struct tagRecordDownloadInfo
{
    unsigned long       uType ;
    unsigned __int64    ullIndex ;
    unsigned long       uReserved[2] ;
 } RECORDDOWNLOADINFO,*LPRECORDDOWNLOADINFO;
```

Members

uType

0: 按帧ID, 1: 按字节, 2:按时间

ullIndex

从此位置开始下载.

uReserved

保留字

IPC_RECORD_CTRL

录像控制结构

```
typedef struct tagIPC_RECORD_CTRL
{
 DWORD    dwFileType ;
 DWORD    dwChannel ;
} IPC_RECORD_CTRL,*LPIPC_RECORD_CTRL;
```

Members

dwFileType

文件类型，位与结果为0表示不启用，1表示启用

dwFileType & 0x0001，表示是否查询定时录像文件

dwFileType & 0x0002，表示是否查询手动录像文件

dwFileType & 0x0004，表示是否查询报警录像文件

dwFileType & 0x0008，表示是否查询移动桢测录像文件

dwFileType & 0x0010，表示是否查询断网录像文件

dwFileType & 0x0020，表示是否查询视频异常录像文件

dwChannel
通道号（码流id）范围 0~3

IPC_RECORD_CONFIG
录像属性配置结构

```
typedef struct tagIPC_RECORD_CONFIG
{
DWORD       dwFileMaxSize;
DWORD       dwStreamType;
}IPC_RECORD_CONFIG,*LPIPC_RECORD_CONFIG;
```

Members

dwFileMaxSize
录像文件分割大小单位MB 范围:32~512
dwStreamType
0:仅视频1:仅音频2:音视频复合流

IPC_FIND_DATA
录像信息结构

```
typedef struct tagIPC_FIND_DATA
{
 char        sFileID[128];
 char        sFileName[128];
 DWORD       dwFileType ;
 IPC_TIME    struStartTime;
 IPC_TIME    struStopTime;
 DWORD       dwFileSize;
} IPC_FIND_DATA,*LPIPC_FIND_DATA;
```

Members

sFileID

文件ID标识号

sFileName
文件名

dwFileType
录像文件类型，位与结果为 0 表示不启用，1 表示启用
dwFileType & 0x0001，表示是否查询定时录像文件
dwFileType & 0x0002，表示是否查询手动录像文件
dwFileType & 0x0004，表示是否查询报警录像文件
dwFileType & 0x0008，表示是否查询移动桢测录像文件
dwFileType & 0x0010，表示是否查询断网录像文件
dwFileType & 0x0020，表示是否查询视频异常录像文件

struStartTime
录像开始时间

struStopTime
录像结束时间

dwFileSize
文件大小 单位 KB

## 2.2 宏定义

### 2.2.1 实时码流

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_REALPLAY _BASE | 0x00001000 | |

### 2.2.2 视频管理

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_VIDEO_BASE | 0x00002000 | |
| IPC_VIDEO_IMGPRO_GROUP | (IPC_VIDEO_BASE+1) | 图像属性<br>参考结构体 IMGPRO |
| IPC_VIDEO_EXPOSURE_MODE | (IPC_VIDEO_BASE+2) | 抗闪烁模式<br>0 :50Hz ; 1:60Hz; 2:自动 |
| IPC_VIDEO_AE_TARGET_RATION | (IPC_VIDEO_BASE+3) | 曝光目标系数 25~400 |
| IPC_VIDEO_MAX_GAIN | (IPC_VIDEO_BASE+4) | SenSor 最大增益 30db, 36db, 42db, 48db, 54db, 60db |
| IPC_VIDEO_WBC | (IPC_VIDEO_BASE+5) | 白平衡设置： 0: 自动 ;1:incandescent;2:d4000;3: d5000;4: sunny;5: cloudy;6: flash;7: |

| | | |
|---|---|---|
| | | fluorescent;8: fluorescent high;9: under water;10:custom;11: 关闭 |
| IPC_VIDEO_DN_MODE | (IPC_VIDEO_BASE+6) | 模式设置(0 白天,1 夜晚,2 自动,3 定制) |
| IPC_VIDEO_BACKLIGHT_COMP | (IPC_VIDEO_BASE+7) | 背光补偿 0 关闭 1 开启 |
| IPC_VIDEO _LOCAL_EXPOSURE | (IPC_VIDEO_BASE+8) | 局部曝光模式 0：关闭; 1: 打开; 2: 暂停; 3: 2x; 4:3x; 5:4x |
| IPC_ VIDEO _MCTF_STRENGTH | (IPC_VIDEO_BASE+9) | 3D 降噪 0~255 |
| IPC_ VIDEO _DCIRIS_GROUP | (IPC_VIDEO_BASE+10) | DC IRIS 模式 参考结构体 IMGIRIS |
| IPC_VIDEO_SHUTTER_GROUP | (IPC_VIDEO_BASE+11) | 电子快门时间范畴 参考结构体 IMGSHUTTER |
| IPC_VIDEO_MJPEG_QUALITY | (IPC_VIDEO_BASE+12) | 图像质量 0~100 |
| IPC_VIDEO_SLOW_SHUTTER | (IPC_VIDEO_BASE+13) | 自动曝光设置中 SlowShutter 0:关闭;1:开启 |
| IPC_ VIDEO _AE_PREFERENCE | (IPC_VIDEO_BASE+14) | 自动曝光的偏好 0：normal;1 : low light;2 : traffic |
| IPC_ VIDEO _METERING_MODE | (IPC_VIDEO_BASE+15) | 自动曝光设置中测光模式 0: 聚光灯;1:center;2:average |
| IPC_ VIDEO _PM_GROUP | (IPC_VIDEO_BASE+16) | 视频遮挡 参考结构体 IMGPM |
| IPC_ENCODE_ENC_MODE | (IPC_VIDEO_BASE+17) | 编码模式 |
| IPC_ENCODE_SN_TIME_ENABLE0 | (IPC_VIDEO_BASE+18) | 显示码流 1 当前时间 |
| IPC_ENCODE_SN_TIME_ENABLE1 | (IPC_VIDEO_BASE+19) | |
| IPC_ENCODE_SN_TIME_ENABLE2 | (IPC_VIDEO_BASE+20) | |
| IPC_ENCODE_SN_TIME_ENABLE3 | (IPC_VIDEO_BASE+21) | |
| IPC_ENCODE_SN_TEXT_GROUP | (IPC_VIDEO_BASE+22) | |
| IPC_ENCODE_ENC_ENC_GROUP | (IPC_VIDEO_BASE+24) | 预览页获取当前码流信息 |
| IPC_VIDEO_IRCUT_GROUP | (IPC_VIDEO_BASE+25) | IRCUT 昼夜模式 参见 IMGIRCUT 结构 |
| IPC_VIDEO_IRCUT_NIGHT2DAY_MIN | (IPC_VIDEO_BASE+26) | 夜晚->白天模式最小阀值 0 |
| IPC_VIDEO_IRCUT_NIGHT2DAY_MAX | (IPC_VIDEO_BASE+27) | 夜晚->白天模式最大阀值 100 |
| IPC_VIDEO_IRCUT_DAY2NIGHT_MIN | (IPC_VIDEO_BASE+28) | 白天->夜晚模式最小阀值 |

| | | |
|---|---|---|
| | | 0 |
| IPC_VIDEO_IRCUT_DAY2NIGHT_MAX | (IPC_VIDEO_BASE+29) | 白天->夜晚模式最大阀值<br>100 |
| IPC_ENCODE_STR_I_INTERVAL | (IPC_VIDEO_BASE+30) | 设置 I 帧间隔 |
| IPC_ENCODE_STR_FORCE_I | (IPC_VIDEO_BASE+31) | 强制 I 帧 |
| IPC_VIDEO_WBC_CUSTOM_R_GAIN | (IPC_VIDEO_BASE+32) | 自定义 R<br>0~1023 |
| IPC_VIDEO_WBC_CUSTOM_B_GAIN | (IPC_VIDEO_BASE+33) | 自定义 B<br>0~1023 |
| IPC_VIDEO_AF_ENABLE | (IPC_VIDEO_BASE+34) | 聚焦模式<br>0:手动模式<br>1:全程自动模式<br>2:自动聚焦锁定模式（云台转动可解锁聚焦） |
| IPC_ENCODE_PROPERTY_SN_MJPEG_LEVEL_STREAM0<br>IPC_ENCODE_PROPERTY_SN_MJPEG_LEVEL_STREAM1<br>IPC_ENCODE_PROPERTY_SN_MJPEG_LEVEL_STREAM2<br>IPC_ENCODE_PROPERTY_SN_MJPEG_LEVEL_STREAM3 | (IPC_VIDEO_BASE+35)<br>(IPC_VIDEO_BASE+36)<br>(IPC_VIDEO_BASE+37)<br>(IPC_VIDEO_BASE+38) | 调整码流 MJPEG 压缩率<br>调整码流 MJPEG 压缩率<br>调整码流 MJPEG 压缩率<br>调整码流 MJPEG 压缩率<br>1(high),2(middle),3(low) |
| | | |
| | | |

### 2.2.3 网络管理

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_ NETWORK _BASE | 0x00003000 | |
| IPC_SYSTEM_SETGET_UPNPINFO | (IPC_NETWORK_BASE+2) | UPNP 信息 |
| IPC_SYSTEM_SETGET_SMTPINFO | (IPC_NETWORK_BASE+3) | SMTP 信息 |
| IPC_SYSTEM_SETGET_FTPINFO | (IPC_NETWORK_BASE+4) | FTP 信息 |
| IPC_SYSTEM_SETGET_IPINFO | (IPC_NETWORK_BASE+5) | IP 信息 |

### 2.2.4 云台管理

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_ PTZ _BASE | 0x00004000 | |
| IPC_CMD_DPTZ_ZOOM | (IPC_PTZ_BASE+1) | 控制数字云台视角 0~11 |
| IPC_CMD_DPTZ_UP | (IPC_PTZ_BASE+2) | 数字云台上移 0~1000 |
| IPC_CMD_DPTZ_DOWN | (IPC_PTZ_BASE+3) | 数字云台下移 0~1000 |
| IPC_CMD_DPTZ_LEFT | (IPC_PTZ_BASE+4) | 数字云台左移 0~1000 |
| IPC_CMD_DPTZ_RIGHT | (IPC_PTZ_BASE+5) | 数字云台右移 0~1000 |
| IPC_CMD_DPTZ_SET_PRESET | (IPC_PTZ_BASE+6) | 设置数字云台预置位 |

| | | |
|---|---|---|
| | | 预置点索引 1~9 |
| IPC_CMD_DPTZ_CLEAR_PRESET | (IPC_PTZ_BASE+7) | 清理指定数字云台预置位<br>预置点索引 1~9 |
| IPC_CMD_DPTZ_GOTO_PRESET | (IPC_PTZ_BASE+8) | 转到数字云台预置位<br>预置点索引 1~9 |
| IPC_CMD_DPTZ_CLEAR_ALL_PRESET | (IPC_PTZ_BASE+9) | 清理所有数字云台预置位 |
| IPC_CMD_DPTZ_GOTO_HOME | (IPC_PTZ_BASE+10) | 移到数字云台 |
| IPC_CMD_PTZ_FOCUS_NEAR | (IPC_PTZ_BASE+21) | 缩小机械云台焦距 |
| IPC_CMD_PTZ_FOCUS_FAR | (IPC_PTZ_BASE+22) | 放大机械云台焦距 |
| IPC_CMD_PTZ_ZOOM_WIDE | (IPC_PTZ_BASE+23) | 放大机械云台视角 |
| IPC_CMD_PTZ_ZOOM_TELE | (IPC_PTZ_BASE+24) | 缩小机械云台视角 |
| IPC_CMD_PTZ_UP | (IPC_PTZ_BASE+25) | 机械云台上移 |
| IPC_CMD_PTZ_DOWN | (IPC_PTZ_BASE+26) | 机械云台下移 |
| IPC_CMD_PTZ_LEFT | (IPC_PTZ_BASE+27) | 机械云台左转 |
| IPC_CMD_PTZ_RIGHT | (IPC_PTZ_BASE+28) | 机械云台右转 |
| IPC_CMD_PTZ_GOTO_HOME | (IPC_PTZ_BASE+29) | 移到机械云台起始位置 |
| IPC_CMD_PTZ_SET_PAN_SPEED | (IPC_PTZ_BASE+30) | 机械云台水平速度<br>1~100,0 表示停止运动 |
| IPC_CMD_PTZ_SET_TILT_SPEED | (IPC_PTZ_BASE+31) | 机械云台垂直速度 1-<br>100,0 表示停止运动 |
| IPC_CMD_PTZ_SET_LEFTBORDER | (IPC_PTZ_BASE+44) | 设置左边界 |
| IPC_CMD_PTZ_SET_RIGHTBORDER | (IPC_PTZ_BASE+45) | 设置右边界 |
| IPC_CMD_PTZ_AUTO_SCAN | (IPC_PTZ_BASE+46) | 自动左右扫描，该功能需要前端设备支持 |
| IPC_CMD_PTZ_AUXIOPEN | (IPC_PTZ_BASE+47) | 辅助点开，支持辅助点<br>1~8，该功能需要前端设备支持 |
| IPC_CMD_PTZ_AUXICLOSE | (IPC_PTZ_BASE+48) | 辅助点关 |
| IPC_CMD_PTZ_AUTOPAN | (IPC_PTZ_BASE+49) | 自动水平旋转 |
| IPC_CMD_PTZ_GOTO_REL_POSITION | (IPC_PTZ_BASE+50) | 从相对当前坐标位置运动指定的偏移量 |
| IPC_CMD_PTZ_GOTO_ABS_POSITION | (IPC_PTZ_BASE+51) | 运动到指定的绝对坐标位置 |
| IPC_CMD_PTZ_GET_ABS_POSITION | (IPC_PTZ_BASE+52) | 获取云台当前的绝对坐标 |
| IPC_CMD_PTZ_3DPOSBOX | (IPC_PTZ_BASE+55) | 设置区域进行云台定位<br>同 ipcPTZSelZoomIn |

### 2.2.5 事件管理

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_EVENT_BASE | 0x00005000 | |
| IPC_EVENT_SCHEDULED | (IPC_EVENT_BASE+14) | 事件类型 |
| IPC_EVENT_INPUTPORT | (IPC_EVENT_BASE+15) | |
| IPC_EVENT_CAMERATAMPERING | (IPC_EVENT_BASE+16) | |
| IPC_EVENT_MOTIONDETECTION | (IPC_EVENT_BASE+17) | |
| IPC_RECORD_FILELOCATION_FTP | (IPC_EVENT_BASE+23) | 录像文件保存方式 |
| IPC_RECORD_FILELOCATION_SDCARD | (IPC_EVENT_BASE+24) | |
| IPC_RECORD_FILELOCATION_EMAIL | (IPC_EVENT_BASE+25) | |
| IPC_EVENT_REBOOT | (IPC_EVENT_BASE+31) | |
| IPC_EVENT_OUTPUT | (IPC_EVENT_BASE+48) | |

### 2.2.6 系统管理

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_ SYSTEM _BASE | 0x00007000 | |
| IPC_SYSTEM_GET_DEVICE_NAME | (IPC_SYSTEM_BASE+5) | 设备基本信息 |
| IPC_SYSTEM_GET_DEVICE_MANUFACTURER | (IPC_SYSTEM_BASE+6) | 厂商信息 |
| IPC_SYSTEM_GET_FW_VERSION | (IPC_SYSTEM_BASE+7) | 嵌入式软件版本 |
| IPC_SYSTEM_GET_HW_VERSION | (IPC_SYSTEM_BASE+8) | 硬件版本 |
| IPC_SYSTEM_SETGET_DATEINFO | (IPC_SYSTEM_BASE+9) | 系统时间 |
| IPC_SYSTEM_SETGET_DATEINFOEX | (IPC_SYSTEM_BASE+30) | 系统时间信息扩展 |
| IPC_SYSTEM_AUDIO_SETTING | (IPC_SYSTEM_BASE+15) | 设置语音属性 |
| IPC_SYSTEM_GETSET_ALARMCENTER | (IPC_SYSTEM_BASE+18) | 报警中心属性 |
| IPC_SYSTEM_GET_DEVICE_SN | (IPC_SYSTEM_BASE+19) | 设备 SN 编号,最大 64 字节 |
| IPC_SYSTEM_GET_SUPPORT_GB | (IPC_SYSTEM_BASE+31) | 获取设备支持国标能力情况 |
| IPC_SYSTEM_GET_SUPPORT_RTSP | (IPC_SYSTEM_BASE+32) | 获取设备支持 RTSP 能力情况 |

### 2.2.7 限制性常量定义

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_MAX_NAME_LEN | 64 | 名称最大长度 |
| IPC_MAX_PRESET_COUNT | 255 | 最多支持的预置点个数 |
| IPC_MAX_CRUISE_COUNT | 32 | 最多支持的巡航路径个数 |
| IPC_MAX_IP_LEN | 64 | IP 地址长度 |
| IPC_MAX_LOGINFO_LEN | 1024 | 最大日志信息长度 |
| IPC_MEMO_LENGTH | 64 | 备注信息最大长度 |
| IPC_MAX_IMGINFO_COUNT | 32 | 最多支持的分辨率帧率信息个数 |

### 2.2.8 日志

| 宏定义 | 宏定义值 | 含义 |
| --- | --- | --- |
| IPC_LOG_MAX_USER_NAME_LEN | 20 | 日志查询用户名称最大长度 |
| IPC_LOG_MAX_SUMMARY_LEN | 128 | 日志查询概述最大长度 |
| IPC_LOG_MAX_DESCRIPTION_LEN | 512 | 日志查询详细描述最大长度 |
| IPC_LOG_SUCCESS | 1000 | 获取日志信息成功 |
| IPC_LOG_NOFIND | 1001 | 未查找到日志 |
| IPC_LOG_ISFINDING | 1002 | 正在查找请等待 |
| IPC_LOG_NOMORERECORD | 1003 | 没有更多的日志，查找结束 |
| IPC_LOG_EXCEPTION | 1004 | 查找日志时异常 |
| IPC_LOG_ALL | 0x00000000 | 全部日志 |
| IPC_LOG_ACCESS | 0x00000001 | 访问日志 |
| IPC_LOG_ALARM | 0x00000002 | 报警日志 |
| IPC_LOG_SYSTEM | 0x00000004 | 系统日志 |

### 2.2.9 SD 卡录像

| 宏定义 | 宏定义值 | 含义 |
| --- | --- | --- |
| IPC_RECORD_CTRL_STRLEN | 128 | 录像查询字符串最大长度 |

# 3接口定义

## 3.1 SDK初始化

### ipcInitialize

SDK全局初始化

```
INT ipcInitialize(
 void
);
```

**Parameters**

无

**Return Values**

0表示成功，小于0表示失败

**Remarks**

函数的功能为创建设备控制句柄，并初始化，开辟内存空间
若失败表示SDK依赖的相关库文件缺失。

### ipcUnInitialize

SDK 全局析构

```
INT ipcUnInitialize(
 void
);
```

#### Parameters

无

#### Return Values

0 表示成功，小于 0 表示失败

#### Remarks

函数的功能为释放设备控制句柄，并归还初始化开辟的内存空间

### ipcCreateDevice

设备初始化,创建设备操作句柄

```
ipcHandle ipcCreateDevice(
 void
);
```

#### Parameters

无

Return Values

成功返回设备句柄 ， 失败返回 0

Remarks

## ipcDestroyDevice

释放设备操作句柄

```
INT ipcDestroyDevice(
 ipcHandle handle
);
```

### Parameters

handle
[in] 设备操作句柄

### Return Values

成功返回 0 ， 失败返回-1

### Remarks

## 3.2 网络连接

### ipcIsConnected

判断设备是否已经建立连接

```
INT ipcIsConnected(
 ipcHandle handle,
);
```

**Parameters**

handle
[in] 设备操作句柄

**Return Values**

返回状态
0-链接　非 0-断开

**Remarks**

若失败则表示设备与 SDK 的通信断开，此函数在设备登陆时使用。（登陆成功后建议用心跳检测设备状态）

ipcDisConnect

断开与设备的网络连接

```
INT ipcDisConnect (
 ipcHandle handle,
);
```

Parameters

handle

[in] 设备操作句柄

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcConnect

与设备建立网络连接

```
INT ipcConnect (
 ipcHandle handle,
 const CHAR * accout,
 const CHAR * password,
 const CHAR * ip,
 INT port
);
```

Parameters

handle

[in] 设备操作句柄

accout

[in] 设备登陆账号

password

[in] 设备登陆密码

ip

[in] 设备的 ip 地址

port

[in] 设备信令通信端口 默认端口号 30000

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

非阻塞模式连接设备

ipcSetConnectTimeOut

设置网络连接超时时限

```
INT ipcSetConnectTimeOut (
 ipcHandle handle,
DWORD timeout
);
```

Parameters

handle

[in] 设备操作句柄

timeout

[in] 超时时间（单位：毫秒）系统默认超时时间为 10*1000 毫秒

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

此函数主要应用场景为通过 API 设置，获取像机参数时若发生断网等异常事件，API 最大超时时限。

**ipcWaitConnected**

等待设备连接成功

```
INT ipcWaitConnected (
ipcHandle handle ,
DWORD dwWaitTime
);
```

**Parameters**

handle

[in] 设备操作句柄

dwWaitTime

[in] 等待时间（单位：毫秒）

建议等待时间大于 5000 毫秒

**Return Values**

错误代码 0 表示 SDK 与设备连接成功 ,小于 0 表示在 dwWaitTime 时限内未能连接成功失败，若失败请调用 ipcDisconnect 然后重新连接。

**Remarks**

SDK 的 ipcConnect 执行后在后台与设备建立连接，意思是此 API 返回后不表示建立连接成功。通过此函数可以等到连接成功才返回。用户也可自行循环检测 ipcIsConnected 来替代此过程。

## 3.3 实时浏览

### ipcCreateLivePlay

创建预览句柄

```
playHandle ipcCreateLivePlay(
 ipcHandle handle,
 displayHandle displayhandle,
 int channel,
 int streamID,
);
```

**Parameters**

handle
[in] 设备操作句柄,
displayhandle
[in]显示播放的窗口句柄，如无需显示，可为 NULL，表示只需要回调实时帧数据，不需要解码播放显示
channel
[in]通道号，此值始终为 0，保留。
streamID
[in] 码流 ID，0~3 共 4 路码流

**Return Values**

返回预览句柄，涉及预览的配置和控制，均通过此句柄进行

**Remarks**

函数的功能为创建设备浏览实例

## ipcSetDisplayRect

设置视频的显示位置，如不设置默认填满窗口。

```
INT ipcSetDisplayRect(
 playHandle handle,
 RECT rect,
);
```

### Parameters

handle
[in] 预览句柄

rect
[in] 视频显示画面在窗口的 top(左上角 y 坐标),left(左上角 x 坐标),bottom(右下角 y 坐标),right(右下角 x 坐标)
若 rect 全部填 0（{0，0，0，0}）则表示充满当前窗口显示。

### Return Values

错误代码 0 表示成功 ,小于 0 失败

### Remarks

设置视频的显示位置，如不设置则默认填满窗口

## ipcSetBufferTime

设置播放缓冲区缓冲时间。

```
INT ipcSetBufferTime(
 playHandle handle,
 DWROD     time,
);
```

### Parameters

handle
[in] 预览句柄 ipcCreateLivePlay 的返回值。

time
[in]  缓冲时间单位毫秒，取值区间[0--1000]。

### Return Values

正确返回 0，错误返回非 0

### Remarks

### ipcGetBufferTime

获取播放缓冲区缓冲时间。

```
INT ipcGetBufferTime(
 playHandle handle,
 DWROD*     time,
);
```

**Parameters**

handle
[in] 预览句柄 ipcCreateLivePlay 的返回值。

time
[out]  缓冲时间单位毫秒，取值区间[0--1000]。

**Return Values**

正确返回 0，错误返回非 0

**Remarks**

ipcForceIDR

强制设备生成 I 帧

```
INT ipcForceIDR (
ipcHandle handle,
DWORD dwStreamId
);
```

Parameters

handle

[in] 设备句柄

[in] 通道 id

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

## ipcGetDisplayRect

获取视频的显示位置

```
INT ipcGetDisplayRect(
 playHandle handle,
 RECT* rect,
);
```

### Parameters

handle
[in] 预览句柄
rect
[out] 视频显示画面在窗口的 top(起始 x 坐标),left(起始 y 坐标),bottom(宽度),right(高度)

### Return Values

错误代码 0 表示成功 ,小于 0 失败

### Remarks

**ipcStartLivePlay**

开始播放预览

```
INT ipcStartLivePlay(
 playHandle handle
);
```

**Parameters**

handle
[in] 预览句柄

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

ipcStopLivePlay
停止预览

```
INT ipcStopLivePlay(
 playHandle handle
);
```

Parameters

handle
[in] 预览句柄

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks
该函数将停止预览数据的接受与解码播放，但与这预览句柄相关的回调，OSD 配置仍有效

**ipcDestroyLivePlay**
释放预览句柄

```
INT ipcDestroyLivePlay(
 ipcHandle handle,
    playHandle hPlayer
);
```

**Parameters**

handle
[in] 设备操作句柄
hPlayer
[in]预览句柄

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**
该函数将释放与此句柄相关的回调，OSD 配置等信息。

**ipcLiveFrame_CallBack**

视频解码前数据捕获回调函数

```
VOID ipcLiveFrame_CallBack (
    playHandle hPlayer,
    const FrameInfo* lpFrameInfo,
    const CHAR* lpData,
    INT iDataLen ,
    LPVOID lpContext
);
```

**Parameters**

hPlayer

[out]预览句柄。

lpFrameInfo

[out] 当前帧信息

参见FrameInfo结构定义

lpData

[out]帧数据

iDataLen

[out]帧数据长度

lpContext

[out]用户自定义数据。

**Return Values**

**Remarks**

注册回调函数的区别

1）ipcRegisterLiveFrameCallBack

只在回调的 IPCVIDEO_CALLBACK_INFO 结构中返回视频数据和视频相关信息，无时间戳。

2）ipcSetAVCallbackEx

在回调函数的 FrameInfoEx 结构中返回音视频数据并提供时间戳，需根据标志来区分是音频数据或视频数据

3）ipcSetAVCallbackEx2
在回调数据之前包含了头信息等 IPC 流文件播放及检索用到的相关数据，可按照格式拆分出带时间戳的音频或视频数据，此数据直接写文件可生成 IPC 流播放文件
格式参见 IPC_STREAM_HEADER，IPC_VIDEOFRAME_HEADER，IPC_AUDIOFRAME_HEADER，IPC_FRAME_TAIL

### ipcRegisterLiveFrameCallBack

实时解码数据回调函数，通过该回调函数获得解码后的实时视频数据

```
INT ipcRegisterLiveFrameCallBack(
    playHandle hPlayer,
    ipcLiveFrame_CallBack pCallBack,
    LPVOID context
);
```

### Parameters

hPlayer
[in]预览句柄。

pCallBack
[in]回调函数指针。
参见ipcLiveFrame_CallBack

context
[in]用户自定义数据。

### Return Values

错误代码 0 表示成功 ,小于 0 失败

### Remarks

实时解码数据回调函数，通过该回调函数获得解码前的实时视频数据。该函数需在 ipcStartLivePlay 之前调用

**ipcLiveYUV_CallBack**

视频解码后数据捕获回调函数,解码后数据为 YUY2

;

```
typedef void ipcLiveYUV_CallBack( playHandle hPlayer , const FrameInfo * lpFrameInfo,const CHAR* lpData,INT iDataLen, LPVOID lpContext )
```

hPlayer
[out]预览句柄

lpFrameInfo
[out]媒体数据元信息

lpData
[out]存放数据的缓冲区指针

iDataLen
数据大小

lpContext
用户数据

**ipcRegisterLiveYUVCallBack**

视频解码后数据捕获回调函数

```
INT ipcRegisterLiveYUVCallBack(
    playHandle hPlayer,
    ipcLiveYUV_CallBack pCallBack,
    LPVOID context
);
```

**Parameters**

hPlayer
[in]预览句柄。

pCallBack
[in]回调函数指针。

context
[in]用户自定义数据。

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

解码后数据为 YUY2。该函数需在 ipcStartLivePlay 之前调用

**ipcStartRecord**

开始录像，并保存至本地文件

```
INT ipcStartRecord (
 playHandle hPlayer,
 const WCHAR * filePath
);
```

**Parameters**

hPlayer
[in]预览句柄。

filePath
[in]保存的文件名。文件名名后缀为 ASF。需填写全路径(如: d:\record\xxxx.asf)

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

本函数为开始录像，并将文件保存至本地,文件类型为 ASF 文件。该函数可在预览句柄有效时调用。 本录像是客户端录像，在形成录像时，同时生成扩展名为*.ipcdx 的同名索引文件。 用户也可以通过回调获取到视频流后，自己保存私有格式的文件，通过自己的私有播放器播放。

*.ipcdx 文件只有在使用本公司提供的录像播放库及程序时才需要。任何支持 asf 的第三方程序均可不依赖 ipcdx 文件播放此录像。

该函数可在预览句柄有效时调用，并需要与 ipcStopRecord 成对使用。其他要求请参考 ipcStopRecord 的函数说明。

**ipcStopRecord**
结束录像

```
INT ipcStopRecord (
 playHandle hPlayer
);
```

**Parameters**

hPlayer
[in]预览句柄。

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**
结束录像，该函数需在 ipcStopLivePlay 之前调用，否则 ipcStopLivePlay 会强制调用本函数，确保录像文件的完整性。

**ipcCapturePicture**

单帧数据捕获并保存成图形存放在指定的文件中

```
INT ipcCapturePicture(
 playHandle hPlayer,
 int picType,
 const WCHAR* file_name
);
```

**Parameters**

hPlayer
[in]预览句柄。

picType
[in]保存的图像类型，1 表示 bmp 格式，2 表示 jpeg 格式
保存 bmp 时格式为 bitmap 24bit
file_name
[in]文件名

**Return Values**

正确返回保存图像的大小，否则返回错误代码小于 0 失败

**Remarks**

ipcCapturePictToBuff

单帧数据捕获并保存成图形存放在指定的内存空间中

```
INT ipcCapturePictToBuff(
 playHandle hPlayer,
 int picType,
 BYTE* buff,
 LONG* len
);
```

Parameters

hPlayer
[in]预览句柄。

picType
[in]保存的图像类型，1 表示 bmp 格式 只支持 bmp24

buff
[out]位图数据
数据为 bitmap 24bit
len
[out]文图数据长度

Return Values

正确返回保存图像的大小，否则返回错误代码

Remarks

**ipcSetDisplayOSD**

设置 OSD 显示信息

```
INT ipcSetDisplayOSD(
 playHandle hPlayer,
 LPOSDPARAM osd_info
);
```

**Parameters**

hPlayer

[in]预览句柄。

osd_info

[in] 图片/文字叠加的数据结构

**LPOSDPARAM 的定义**

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

LPOSDPARAM 含有详情见结构体说明

**ipcSetDeviceOSD**

设置设备 OSD

```
INT ipcSetDeviceOSD(
 ipcHandle handle,
 IN LPDEVICEOSD lpOSD
)
```

**Parameters**

handle

[in] 设备操作句柄。

lpOSD
[in] DEVICEOSD 数据结构

```
DEVICEOSD 的定义
```

### Return Values

错误代码 0 表示成功 ，小于 0 失败

### Remarks

DEVICEOSD 含有详情见结构体说明

## ipcGetDeviceOSD

获取设备 OSD

```
INT ipcGetDeviceOSD(
 ipcHandle handle,
 OUT LPDEVICEOSD lpOSD
)
```

### Parameters

handle
[in] 设备操作句柄。

lpOSD
[out] DEVICEOSD 数据结构

```
DEVICEOSD 的定义
```

### Return Values

错误代码 0 表示成功 ，小于 0 失败

### Remarks

DEVICEOSD 含有详情见结构体说明

ipcSetDeviceDateTimeOSD

设置设备时间信息显示格式

```
INT ipcSetDeviceDateTimeOSD(
ipcHandle handle,
IN LPDATETIMEOSD lpTimeOSD
)
```

Parameters

handle

[in] 设备操作句柄。

lpOSD

[in] DATETIMEOSD 数据结构

DATETIMEOSD 的定义

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

DATETIMEOSD 含有详情见结构体说明

ipcGetDeviceDateTimeOSD

获取设备时间信息显示格式

```
INT ipcGetDeviceDateTimeOSD(
ipcHandle handle,
OUT LPDATETIMEOSD lpTimeOSD
)
```

Parameters

handle

[in] 设备操作句柄。

lpOSD

[out] DATETIMEOSD 数据结构

DATETIMEOSD 的定义

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks
DATETIMEOSD 含有详情见结构体说明

ipcGetImageWidth
获取图像宽度

```
INT ipcGetImageWidth(
 playHandle hPlayer,
 LONG* width
);
```

Parameters

hPlayer
[in]预览句柄。

width
[out] 图片宽度

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcGetImageHeight**

设置图像高度

```
INT ipcGetImageHeight(
 playHandle hPlayer,
 LONG* height
);
```

**Parameters**

hPlayer
[in]预览句柄。

height
[out] 图片高度

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetVideoFPS**
获取帧率

```
ipcGetVideoFPS(
ipcHandle handle,
int streamID,
OUT DWORD* fps
)
```

**Parameters**

handle
[in]设备操作句柄

streamID
[in]码流 id 0~3

fps
[out] 帧率

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetVideoFPS**
设置帧率

```
ipcSetVideoFPS(
ipcHandle handle,
int streamID,
IN DWORD fps
)
```

Parameters

handle
[in]设备操作句柄

streamID
[in]码流 id 0~3

fps
[in] 帧率 :

可以设置的帧率值受设备当前的分辨率、总共支持的码流个数影响，具体参考下表，需通过 ipcGetSupportedImgInfo 获取设备当前支持的分辨率帧率列表。

**参考值如下表（以实际设备查询列表值为准）：**@前面是分辨率，后面是对应的帧率，+前后代表不同的码流 id 的数据

1080P 摄像机分辨率组合

| 单码流 | | |
|---|---|---|
| 1080P@25 | | |
| 1080P@30 | | |
| 720P@25 | | |
| 720P@30 | | |
| 720P@60 | | |
| 双码流 | | |
| 1080P@25 + 576P@25 | | |
| 1080P@30 + 480P@30 | | |
| 1080P@15 + 1080P@15 | | |
| 1080P@15 + 720P@15 | | |
| 720P@60 + CIF@60 | | |
| 720P@30 + 720P@30 | | |
| 720P@30 + CIF@30 | | |
| 720P@25 + 576P@25 | | |
| 576P@25 + 576P@25 | | |
| 三码流 | | |
| 1080P@25+ 576P@25+ CIF@25 | | |
| 1080P@30 + CIF@30 + CIF@30 | | |
| 1080P@20 + 720P@20 + 576@20 | | |
| 1080P@10 + 1080P@10 + 1080@10 | | |
| 720P@25+ 720P@25+ 720@25 | | |
| 720P@25 + 720P@25 +576@25 | | |
| 576P@25 + 576P@25 +576@25 | | |

| 四码流 | | |
|---|---|---|
| 1080P@25+ CIF@25+ CIF@25+ CIF@25 | | |
| 1080P@30 + CIF@30 + CIF@30 + CIF@30 | | |
| 1080P@20 + 720P@20 + 576@20 + CIF@20 | | |
| 1080P@15 + 576P@15 + 576@15 + 576@15 | | |
| 720P@30 + 720P@30 + CIF@20 + CIF@30 | | |
| 720P@20+ 720P@20+ 576@20+ 576@20 | | |
| 720P@15+ 720P@15+ 720@15+ 720@15 | | |
| 576P@25+ 576P@25+ 576@15+ 576@25 | | |

720P 摄像机分辨率组合

| 单码流 | | |
|---|---|---|
| 720P@25 | | |
| 720P@30 | | |
| 720P@60 | | |
| 双码流 | | |
| 720P@25 + 576P@25 | | |
| 720P@30 + 720P@30 | | |
| 720P@60 + CIF@60 | | |
| 720P@30 + CIF@30 | | |
| 576P@25 + 576P@25 | | |
| 三码流 | | |
| 720P@25+ 720P@25+ 720@25 | | |
| 720P@25+ 720P@25+ 576@25 | | |
| 576P@25+ 576P@25+ 576@25 | | |
| 四码流 | | |
| 720P@25+ 720P@25+ CIF@25+CIF@25 | | |
| 720P@30+ 720P@30+ CIF@30+CIF@30 | | |
| 720P@20+ 720P@20+ 576@20+576@20 | | |
| 720P@10+ 720P@10+ 720@10+720@10 | | |
| 576P@25+ 576P@25+ 576@25+576@25 | | |

D1 标清摄像机分辨率组合

| 单码流 | | |
|---|---|---|
| 576P@25 | | |
| 480P@30 | | |
| 双码流 | | |
| 576P@25+ CIF@25 | | |
| 480P@30+ CIF@30 | | |
| 576P@25+ QVGA@20 | | |

Return Values

错误代码 0 表示成功 , 小于 0 失败

Remarks

ipcMP4Open

创建 MP4 文件并返回文件句柄

```
HANDLE ipcMP4Open (
 const char *pFileName
);
```

Parameters

pFileName

[in]文件全路径

Return Values

文件句柄

Remarks

**ipcMP4Close**

关闭 MP4 格式文件

```
LONG  ipcMP4Close (
 HANDLE hMP4
);
```

**Parameters**

hMP4

[in] 文件句柄

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcMP4AddVideoStream**
添加视频流

```
LONG ipcMP4AddVideoStream (
HANDLE                                  hMP4,
Long                                        lStreamID,
IPC_MP4_VIDEOPARAM        Param
);
```

**Parameters**

hMP4
[in] 文件句柄

lStreamID
[in] 码流 ID 。同一个文件中的所有流 ID 不能重复 视频为 0 ， 音频为 1

Param
[in]视频流数据结构
dwHeight
高

dwWidth
宽

dwFrameInterval
帧间隔（单位：100 纳秒）
计算方法（如：10000000.0 / FrameRate）
dwBitRate
码率

wAspectRatio
比例

dwParWidth
保留默认为 0

dwParHeight
保留默认为 0

cbExtraInfo
保留默认为 0

pExtraInfo
保留默认为 0

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcMP4AddAudioStream**
添加音频流

```
LONG  ipcMP4AddAudioStream (
HANDLE                                    hMP4,
long                                              lStreamID,
IPC_MP4_AUDIOPARAM       Param
);
```

**Parameters**

hMP4
[in] 文件句柄

lStreamID
[in] 码流 ID

Param
[in]音频流数据

dwChannel
声道
dwSamplePerSec
采样频率
dwBitsPerSample
采样精度
nAvgBytesPerSec
平均码率，计算方法： dwSamplePerSec * dwBitsPerSample * dwChannel / 8
nBlockAlign
块对齐方式
cbExtraInfo
扩展数据长度
pExtraInfo
扩展数据

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcMP4Run**

开始 MP4 录像

```
LONG ipcMP4Run (
 HANDLE        hMP4
 );
```

**Parameters**

hMP4

[in] 文件句柄

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcMP4Stop**

停止 MP4 录像

```
LONG ipcMP4Stop (
 HANDLE        hMP4
 );
```

**Parameters**

hMP4

[in] 文件句柄

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcMP4Write**
写入音视频数据

```
LONG ipcMP4Write (
HANDLE                          hMP4,
long                                    lStreamID,
const unsigned char*     pData,
long                                    lSize,
int                                      nFrameType,
__int64                               llTimeStamp
 );
```

**Parameters**

hMP4
[in] 文件句柄

lStreamID
[in]码流 ID

pData
[in]数据

lSize
[in]数据长度

nFrameType
[in]帧类型

llTimeStamp
[in]时间戳，单位：100 纳秒

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**
写入文件的第一帧视频帧必须为 I 帧

## 3.4 视频设置

**ipcGetStreamInfo**

获取设备当前的码流信息

```
INT ipcGetStreamInfo(
ipcHandle handle,
INT iStream,
StreamInfo* pStreamInfo
)
```

**Parameters**

handle
[in] 设备操作句柄

iStream
[in]码流 ID  0~3

pStreamInfo
[out] 码流信息数据

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**ipcGetCurAllStreamInfo**

获取设备当前各个码流信息

```
INT ipcGetCurAllStreamInfo(
ipcHandle handle,
DeviceImgInfo* pDevImgInfo
)
```

**Parameters**

handle
[in] 设备操作句柄

pDevImgInfo

[out] 全部码流编码信息

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**ipcSetCurAllStreamInfo**

设置设备各个码流信息

```
INT ipcSetCurAllStreamInfo(
ipcHandle handle,
DeviceImgInfo* pDevImgInfo
) ;
```

**Parameters**

handle
[in] 设备操作句柄

pDevImgInfo
[in] 全部码流信息

**Remarks**

参考函数 ipcGetSupportedImgInfo 查询设备支持的视频配置信息
参考函数 ipcSetVideoFPS 的文档部分附带码流个数、分辨率 、帧率关系表。

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**ipcSetConfig**

设置指定类型的参数

```
INT ipcSetConfig(
 ipcHandle handle,
 INT command,
 INT param,
 CONST CHAR* inBuffer,
);
```

Parameters

handle
[in]操作句柄。

command
[in] 系统参数设置控制码（见配置命令宏定义）
**定义及取值范围请参见接口定义中视频设置部分最后面的命令码详细说明**

param
[in] 不同的命令，有时需要额外的参数，比如制定通道号，或者流类型

inBuffer
[in]存放参数数据的缓存指针
**定义请参见数据结构中视频部分**

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks
param 表示对某属性进行设置时需要附加对其进行赋值操作
inBuffer 表示对某属性进行设置时需要传入数据结构
两个参数有可能会同时使用

IPC_ENCODE_ENC_SN_GROUP
此处数据结构比较复杂，并且需要全部输入正确
请参见 IPCSdkFuncDemo 例子中的 CameraContfiSetting.cpp

**ipcGetConfig**

获取指定类型的参数

```
INT ipcGetConfig(
 ipcHandle  handle,
 INT        command,
    INT         param,
    VOID *      outBuffer,
);
```

**Parameters**

handle

[in]设备操作句柄。

command

[in] 系统参数设置控制码（见配置命令宏定义）

**定义及取值范围请参见接口定义中视频设置部分最后面的命令码详细说明**

param

[in] 不同的命令，有时需要额外的参数，比如制定通道号，或者流类型

[out]outBuffer

特别指令需要返回的复杂数据

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

此处数据结构比较复杂，请参见 IPCSdkFuncDemo

**视频属性说明**

| 名称 | 控制码 | 取值范围及说明 |
|---|---|---|
| | | |
| 抗闪烁模式 | IPC_VIDEO_EXPOSURE_MODE | 0：抗闪烁 50Hz<br>1：抗闪烁 60Hz |

| | | |
|---|---|---|
| | | 2：自动 |
| 曝光目标系数 | IPC_VIDEO_AE_TARGET_RATION | 25 ~ 400 |
| SenSor 最大增益 | IPC_VIDEO_MAX_GAIN | 30、36、42<br>48、54、60 |
| 白平衡设置 | IPC_VIDEO_WBC | 0：自动<br>1：白炽 2：D4000<br>3：D5000 4：晴朗<br>5：多云 6：闪光<br>7：荧光 8：高荧光<br>9：水下 |
| 夜晚模式 | IPC_VIDEO_DN_MODE | 0：白天<br>1：夜晚<br>2：自动 |
| 背光补偿 | IPC_VIDEO_BACKLIGHT_COMP | 0：关闭<br>1：开启 |
| 局部曝光模式 | IPC_VIDEO_LOCAL_EXPOSURE | 0：关闭<br>1：开启<br>2：暂停<br>3：2x(曝光模式)<br>4：3X<br>5：4X |
| 3D 降噪 | IPC_VIDEO_MCTF_STRENGTH | 0 ~ 255 |
| 图像属性 | IPC_VIDEO_IMGPRO_GROUP | 参考结构体 IMGPRO<br>配置饱和度、亮度、<br>色度、 对比度、锐度 |
| 自动曝光设置中 SlowShutter | IPC_VIDEO_SLOW_SHUTTER | 1 开启<br>0 关闭 |
| 自动曝光的偏好 | IPC_VIDEO_AE_PREFERENCE | 0 正常<br>1 低标准<br>2 Traffic |
| 自动曝光设置中测光模式 | IPC_VIDEO_METERING_MODE | 0 点测光<br>1 局部测光<br>2 中央平均测光<br>3 分割测光 |
| | | |
| 编码模式 | IPC_ENCODE_ENC_MODE | 0：正常模式<br>1：低延时模式 |
| 显示当前时间 | IPC_ENCODE_SN_TIME_ENABLE0 | 码流 1<br>0：不显示<br>1：显示 |
| 显示当前时间 | IPC_ENCODE_SN_TIME_ENABLE1 | 码流 2<br>0：不显示<br>1：显示 |

| | | |
|---|---|---|
| 显示当前时间 | IPC_ENCODE_SN_TIME_ENABLE2 | 码流 3<br>0：不显示<br>1：显示 |
| 显示当前时间 | IPC_ENCODE_SN_TIME_ENABLE3 | 码流 4<br>0：不显示<br>1：显示 |
| | | |
| DC IRIS 模式 | IPC_VIDEO_DCIRIS_GROUP | IMGIRIS 结构体 |
| 电子快门时间范畴 | IPC_VIDEO_SHUTTER_GROUP | IMGSHUTTER 结构体 |
| 视频遮挡 | ipcSetPictMask<br>ipcGetPictMask | 变更为 API 直接设置<br>IMGPM 结构体 |
| 显示信息叠加 | IPC_ENCODE_SN_TEXT_GROUP | SNTEXT 结构体 |
| 预览页获取当前码流信息 | IPC_ENCODE_ENC_ENC_GROUP | ENCENC 结构体 |

ipcSetPictMask

**设置视频遮挡信息**

```
INT ipcSetPictMask(
 ipcHandle handle,
 IMGPM pm[4]
);
```

Parameters

handle
[in]设备句柄。

pm
[in]视频遮挡数组
**详见 IMGPM 数据结构定义**

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcGetPictMask

获取视频遮挡信息

```
INT ipcGetPictMask(
 ipcHandle handle,
 IMGPM* ipm
);
```

Parameters

handle

[in]设备句柄。

ipm

[out]视频遮挡数组，大小为

详见 IMGPM 数据结构定义

Return Values

错误代码 0 表示成功 ，小于 0 失败

Remarks

ipcSetImageQuant

设置 MJPEG 实时码流图像质量

```
INT ipcSetImageQuant (
 ipcHandle handle,
INT streamID,
INT param
);
```

Parameters

[in] handle
设备操作句柄
[in] streamID
码流ID
[in] param
编码质量 1-100

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcGetSupportedImgInfo

获取设备支持的分辨率和帧率信息

```
INT ipcGetSupportedImgInfo(
ipcHandle handle,
INT iStreanCount,
AllImgInfo* pImgInfo
)
```

Parameters

[in] handle
设备操作句柄

[in] iStreanCount
码流数1-4,分别代表获取单码流,双码流,3码流,4码流对应的每个码流支持的分辨率，帧率信息

[out] pImgInfo
分辨率帧率信息

Return Values

错误代码 0 表示成功 ,小于 0 失败

ipcGetBitRate
获取设备的码率

```
INT ipcGetBitRate(
ipcHandle handle,
INT iStream,
BitRate* pBitRate
)
```

Parameters

[in] handle
设备操作句柄

[in] iStream
码流id 0~3

[out] pBitRate
码率信息

Return Values

错误代码 0 表示成功 ,小于 0 失败

ipcSetBitRate

获取设备的码率

```
INT ipcSetBitRate(
ipcHandle handle,
INT iStream,
BitRate* pBitRate
)
```

Parameters

[in] handle
设备操作句柄

[in] iStream
码流id 0~3

[in] pBitRate
码率信息

Remarks
该编码参数修改后，已经建立的视频连接在接收到断线通知后需要重新请求视频

Return Values

错误代码 0 表示成功 ,小于 0 失败

ipcGetSupportedStreamCount
获取设备的码率

```
INT ipcGetSupportedStreamCount(
ipcHandle handle,
INT* iStreamCount
)
```

Parameters

[in] handle
设备操作句柄

[out] iStreamCount
码流个数 1~4

Return Values

错误代码 0 表示成功 ,小于 0 失败

**ipcSetSupportedStreamCount**

获取设备的码率

```
INT ipcSetSupportedStreamCount(
ipcHandle handle,
INT iStreamCount
)
```

**Parameters**

[in] handle
设备操作句柄

[in] iStreamCount
码流个数 1~4

**Remarks**

该编码参数修改后，已经建立的视频连接在接收到断线通知后需要重新请求视频

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**ipcGetAudioState**

获取设备音频编码编码状态

```
INT ipcGetAudioState(
ipcHandle    handle,
DWORD*     dwState
) ;
```

**Parameters**

[in] handle
设备操作句柄

[out] dwState
当前设备音频编码状态 0 禁用音视频编码; 1 启用音频编码

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetAudioState**

设置设备码流的编码状态

```
INT ipcSetAudioState(
ipcHandle        handle,
DWORD          dwState
) ;
```

**Parameters**

[in] handle
设备操作句柄

[in] dwState
当前设备音频编码状态 0 禁用音视频编码; 1 启用音频编码

**Remarks**

当音频编码禁用时，不能语音对讲和实时音频监听；
当音频启用时，语音对讲或者音频监听需要重新连接
才能获取到音频数据。

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**ipcGetShutterTimeMinRange**

获取电子快门时间的最小值取值范围

```
INT ipcGetShutterTimeMinRange (
ipcHandle        handle,
DWORD* pMinRangeArray,
DWORD* pArraySize
) ;
```

**Parameters**

[in] handle

设备操作句柄

[out] pMinRangeArray

存放电子快门时间的一维数组地址，数组大小为 30

[in/out] pArraySize

输入为 pMinRangeArray 数组的大小，输出为数组里有效元素的个数

**Remarks**

用户可通过本函数获取设备当前所支持的电子快门时间的最小值的组合，然后再通过 ipcGetShutterTimeMaxRange 获取电子快门时间的最大值组合，最后可通过 ipcSetConfig 设置电子快门时间

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**ipcGetShutterTimeMaxRange**

获取电子快门时间的最大值取值范围

```
INT ipcGetShutterTimeMaxRange (
ipcHandle          handle,
DWORD              dwMinValue,
DWORD*             pMaxRangeArray,
DWORD*             pArraySize
) ;
```

**Parameters**

[in] handle

设备操作句柄

[in] dwMinValue

电子快门时间最小值，可从ipcGetShutterTimeMinRange获取

[out] pMaxRangeArray

存放电子快门时间的一维数组地址，数组大小为 30

[in/out] pArraySize

输入为 pMinRangeArray 数组的大小，输出为数组里有效元素的个数

**Remarks**

用户可通过 ipcGetShutterTimeMinRange 获取设备当前所支持的电子快门时间的最小值的组合，然后再通过本函数获取电子快门时间的最大值组合，最后可通过 ipcSetConfig 设置电子快门时间

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

ipcStartLiveImageSharpen

开启图像锐化

```
INT  ipcStartLiveImageSharpen (
playHandle     hPlayer,
float          amount  =  1.2f
);
```

**Parameters**

[in] handle
预览句柄

[in] amount
图像锐化程度，默认为1.2f，范围0.0f-5.0f

Remarks

Return Values

错误代码 0 表示成功 ,小于 0 失败

ipcChangeLiveSharpenDegree

改变图像的锐化程度

```
INT   ipcChangeLiveSharpenDegree(
playHandle     hPlayer,
float          amount
);
```

Parameters

[in] handle
预览句柄

[in] amount
图像锐化程度，范围0.0f-5.0f

Remarks

Return Values

错误代码 0 表示成功 ,小于 0 失败

ipcStopLiveImageSharpen

停止图像锐化

```
INT    ipcStopLiveImageSharpen(
playHandle      hPlayer
);
```

Parameters

[in] handle
预览句柄

Remarks

Return Values

错误代码 0 表示成功 ,小于 0 失败

## 3.5 网络管理

**ipcSetDevicePort**

设置设备端口

```
INT ipcSetDevicePort(
 ipcHandle                      handle,
 DWORD                          dwPort
);
```

**Parameters**

handle
[in]设备操作句柄。

dwPort
[in] 设备通信端口

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetDeviceIPInfo**

设置 IPv4 地址

```
INT ipcSetDeviceIPInfo (
 ipcHandle                      handle,
 CONST LPSYSIPINFO          lpIpInfo,
);
```

**Parameters**

handle
[in]设备操作句柄。

lpIpInfo
[in] ip 地址保存结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetDeviceIPInfo**
获取 IPv4 地址

```
INT ipcSetDeviceIPInfo (
 ipcHandle                          handle,
 CONST LPSYSIPINFO            lpIpInfo,
);
```

**Parameters**

handle
[in]设备句柄。

lpIpInfo
[out] ip 地址保存结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetDeviceFTPInfo**
设置设备 FTP 信息

```
INT   ipcSetDeviceFTPInfo(
ipcHandle  handle,
  CONST  LPSYSFTP  lpFtpInfo
)
```

**Parameters**

handle
[in]设备操作句柄。

lpFtpInfo
[in]  设备 FTP 信息结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetDeviceFTPInfo**
获取设备 FTP 信息

```
INT   ipcGetDeviceFTPInfo(
ipcHandle  handle,
  CONST  LPSYSFTP  lpFtpInfo
)
```

**Parameters**

handle
[in]设备句柄。

lpFtpInfo
[out]  设备 FTP 信息结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetDeviceSMTPInfo**
设置设备 SMTP 信息

```
INT    ipcSetDeviceSMTPInfo(
ipcHandle  handle,
IN  CONST  LPSYSSMTP  lpSmtpInfo
)
```

**Parameters**

handle
[in]设备句柄。

lpSmtpInfo
[in]  设备 SMTP 信息结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetDeviceSMTPInfo**
获取设备 SMTP 信息

```
  INT      ipcGetDeviceSMTPInfo(
ipcHandle  handle,
OUT  LPSYSSMTP  lpSmtpInfo
)  ;
```

**Parameters**

handle
[in]设备句柄。

lpSmtpInfo
[out]  设备 SMTP 信息结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetDeviceUPNPInfo**
设置设备 UPNP 信息

```
INT       ipcSetDeviceUPNPInfo(
ipcHandle  handle,
IN  CONST  LPSYSUPNPINFO  lpUpnpInfo
)
```

**Parameters**

handle
[in]设备句柄。

lpUpnpInfo
[in]  设备 UPNP 信息结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetDeviceUPNPInfo**
设置设备 UPNP 信息

```
  INT     ipcGetDeviceUPNPInfo(
ipcHandle  handle,
OUT  LPSYSUPNPINFO  lpUpnpInfo
)  ;
```

**Parameters**

handle
[in]设备句柄。

  lpUpnpInfo
[out]  设备 UPNP 信息结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetPTZProtocol**

获取云台协议

```
INT ipcGetPTZProtocol(
 ipcHandle      handle,
  DWORD*       ptzProtocol
);
```

**Parameters**

handle

[in]设备句柄。

ptzProtocol

[out]云台控制协议类型

**Return Values**

-1 出错

**Remarks**

ipcGetPTZTransProp

获取云台协议

```
INT ipcGetPTZTransProp (
 ipcHandle handle,
DWORD* protocol_type,
DWORD* address,
LPIPC_COMM_PROP prop
);
```

Parameters

handle
[in] 设备操作句柄

protocol_type
[out]云台协议类型

address
[out]地址

prop
[out]云台属性配置

Return Values

-1 出错 0 成功

Remarks

ipcScanOnlineDevices
检索在线设备

```
INT ipcScanOnlineDevices(
 ipcScanOnlineDeviceCallBack callback,
 LPVOID context
);
```

Parameters

callback
[in]扫描结果回调

context
[in]回调函数的参数

Return Values

失败返回 NULL ， 成功返回服务描述句柄

Remarks

ipcStopScanOnlineDevices
停止检索在线设备

```
INT ipcStopScanOnlineDevices(
 HANDLE handle

);
```

Parameters

handle
[in] 服务描述句柄。

Return Values

-1 出错 ， 0 成功

Remarks

ipcScanOnlineDeviceCallBack
扫描回调函数

```
INT ipcScanOnlineDeviceCallBack(
    HANDLE handle,
    LPIPCSCANINFO info,
    INT infoLen,
    LPVOID context,
);
```

Parameters

handle
[in]设备句柄。

info
[in]扫描的设备信息 ， 参见 IPCSCANINFO 结构

infoLen
[in]
结构长度

context
[in]
用户设置的回调参数

Return Values

无

Remarks

**ipcSetSystemSetting**
设置指定的系统参数类型

```
INT ipcSetSystemSetting(
 ipcHandle handle,
 INT command,
 INT param,
 CONST CHAR* inBuffer
);
```

**Parameters**

handle
[in]设备句柄。

command
[in] 系统参数设置控制码
**定义及取值范围请参见接口定义中视频设置部分最后面的命令码详细说明**

param
[in] 附带的参数

inBuffer
[in]存放参数数据的缓存指针
**定义请参见数据结构中视频部分**

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**
param 表示对某属性进行设置时需要附加对其进行赋值操作
inBuffer 表示对某属性进行设置时需要传入数据结构
两个参数有可能会同时使用

IPC_SYSTEM_GETSET_ALARMCENTER
参见名为 IPCAlarmCenter 的 sample.

**ipcGetSystemSetting**
获取指定类型的系统参数

```
INT ipcGetSystemSetting(
 ipcHandle        handle,
 INT              command,
 INT              param,
 CONST CHAR*      outBuffer
);
```

**Parameters**

handle
[in]设备句柄。

command
[in] 系统参数设置控制码
**定义及取值范围请参见接口定义中视频设置部分最后面的命令码详细说明**

param
[in] 附带的参数

outBuffer
[out]存放返回数据的缓存指针
**定义请参见数据结构中视频部分**

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**
此处数据结构比较复杂，请参见 IPCSdkFuncDemo

IPC_SYSTEM_GETSET_ALARMCENTER
参见名为 IPCAlarmCenter 的 sample.

**系统管理属性说明**

| 控制码 | 名称 | 读写属性 | 附带参数说明 |
|---|---|---|---|
| IPC_SYSTEM_SETGET_SMTPINFO | 邮件传输协议设置 | 见 SYSSMTP 结构体 | 见 SYSSMTP 结构体 |
| IPC_SYSTEM_SETGET_FTPINFO | FTP 设置 | 见 SYSFTP 结构体 | 见 SYSFTP 结构体 |
| IPC_SYSTEM_SETGET_IPINFO | ipv4 地址配置 | 见 SYSIPINFO 结构体 | 见 SYSIPINFO 结构体 |
| | | | |
| IPC_SYSTEM_GET_DEVICE_NAME | 设备名称 | 只读（get） | 字符串 string |
| IPC_SYSTEM_GET_DEVICE_MANUFACTURER | 设备制造商 | 只读（get） | 字符串 string |
| IPC_SYSTEM_GET_FW_VERSION | fw 版本号 | 只读（get） | 字符串 string |
| IPC_SYSTEM_GET_HW_VERSION | hw 版本号 | 只读（get） | 字符串 string |
| IPC_SYSTEM_GET_DEVICE_SN | SN 编码 | 只读(get) | 64 字节字符串 string |
| IPC_ SYSTEM _UPNP_ENABLE | 即插即用是否可用 | 读写 | 0：不可用<br>1：可用 |
| IPC_ SYSTEM_ UPNP_PORT | 即插即用端口号 | 读写 | 0：自动模式<br>或其他值 |
| IPC_ SYSTEM_ UPNP_NAME | 即插即用名字 | 读写 | 字符串 string |
| IPC_SYSTEM_DEV_CTRL_CMD | 恢复默认设置 | 只写 | 0： 重启<br>1：恢复出厂设置 |
| IPC_SYSTEM_SETGET_DATEINFO | 设备时间设置 | 见 SYSDATEINFO 结构体 | 见 SYSDATEINFO 结构体 |
| IPC_SYSTEM_SETGET_DATEINFOEX | 更新设备时间 | 见 SYSDATEINFOEX 结构体 | 见 SYSDATEINFOEX 结构体 |
| IPC_SYSTEM_GETSET_ALARMCENTER | 报警中心设置 | 读写 | 见 IPC_ALARMCENTER_UP_CFG 结构体 |
| IPC_SYSTEM_GET_SUPPORT_GB | 设备支持国标能力 | 只读 | 0：不支持<br>1：支持 |
| IPC_SYSTEM_GET_SUPPORT_RTSP | 设备支持 RTSP 能力 | 只读 | 0：不支持<br>1：支持 |

**ipcGetDeviceSN**

获取设备序列号

```
INT ipcGetDeviceSN(
ipcHandle handle,
OUT CHAR* pStrSn,
INT nLen) ;
```

**Parameters**

handle
[in] 设备操作句柄。

pStrSn
[out] 序列号，序列号,最长 63 个字符

nLen
[out] pStrSn 内存块长度

**Return Values**

非 0 失败

**Remarks**

**ipcGetDDNSParam**

获取 DDNS 服务器配置数据

```
INT ipcGetDDNSParam (
ipcHandle  handle,
DDNSPARA *  pDdnsParam,
);
```

**Parameters**

handle
[in] 设备操作句柄。

pDdnsParam
[out] DDNS 配置数据

**Return Values**

非 0 失败

**Remarks**

**ipcSetDDNSParam**

设置 DDNS 服务器配置数据

```
INT ipcGetDDNSParam (
ipcHandle  handle,
DDNSPARA *  pDdnsParam,
);
```

**Parameters**

handle
[in] 设备操作句柄。

pDdnsParam
[in] DDNS 配置数据

Return Values

非 0 失败

Remarks

ipcDDNSCallBack

DDNS 信息回调

```
typedef void(CALLBACK * ipcDDNSCallBack)(
DDNSRecord * pDdnsRecord,
VOID *pUser
);
```

Parameters

pDdnsRecord
[in] DDNS 信息。

pUser
[in] 用户自定义数据

Return Values

Remarks

ipcSetDDNSServerCallBack

注册 DDNS 信息回调函数

```
INT ipcSetDDNSServerCallBack(
ipcDDNSCallBack fDdnsCallBack,
VOID *pUser
);
```

Parameters

fDdnsCallBack

[in] DDNS 信息会回调函数。

pUser
[in] 用户自定义数据

**Return Values**

非 0 失败

**Remarks**

**ipcStartDdnsListen**
监听本地，等待 IPC 主动链接上报 DDNS 信息

```
INT ipcStartDdnsListen (
 const CHAR *   ip,
INT                    port
) ;
```

[in] ip
　　DDNS 服务侦听的地址,null 则使用本机 ip 地址
[in] port
　　DDNS 服务侦听的端口

**Return Values**
　　成功返回 0，失败返回非 0

**ipcStopDdnsListen**
停止接收 DDNS 信息

```
INT ipcStopDdnsListen()
```

**Return Values**
　　失败返回非 0

**ipcSetDeviceName**
　设置设备名称

```
INT IPCAPI ipcSetDeviceName(
ipcHandle            handle,
IN CHAR*          deviceName
)
```

**Parameters**

handle
[in] 设备操作句柄。

deviceName
[in] 设备名称最长 32 字节

**Return Values**

非 0 失败

**Remarks**

**ipcGetDeviceName**

设置设备名称

```
INT IPCAPI ipcGetDeviceName(
ipcHandle            handle,
OUT CHAR*          deviceName
)
```

**Parameters**

handle
[in] 设备操作句柄。

deviceName
[out] 设备名称最长 32 字节

**Return Values**

非 0 失败

**Remarks**

## 3.6 云台管理

**ipcPTZControl**

控制机械云台

```
INT ipcPTZControl(
 ipcHandle    handle,
 INT          channel,
 INT          PTZCommand,
 INT          PTZControl,
 INT          hSpeed ,
 INT          vSpeed
);
```

**Parameters**

handle
[in]设备句柄。

channel
[in] 通道号

PTZCommand
[in] 云台控制码,参考宏定义的云台控制部分定义

PTZControl
[in] 控制状况，0 表示开始，1 表示停止。
当 PTZCommand 为 IPC_CMD_PTZ_AUXIOPEN 和 IPC_CMD_PTZ_AUXICLOSE 时 PTZControl 传递辅助点号

hSpeed
[in] 水平方向移动速度，取值范围 1~100,0 表示停止

vSpeed
[in] 垂直方向移动速度，取值范围 1~100,0 表示停止

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

参见名为 IPCVideoDemo 的 sample

**Remarks**

**ipcPTZSetPreset**
设置机械云台预置位

```
INT ipcPTZSetPreset(
 ipcHandle     handle,
 INT           channel,
 INT           index,
 const CHAR *   name
)
```

**Parameters**

handle
[in]设备句柄。

channel
[in] 通道号

index
[in] 预置位标号，取值范围 1 到  255 。具体由 IPC 设备连接的云台设备决定。

name
[in] 预置位名称：最多 31 个非中文字符

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

ipcPTZGotoPreset
转移机械云台到预置位

```
INT ipcPTZGotoPreset(
 ipcHandle   handle,
 INT         channel,
 INT         index,
 INT         hSpeed ,
 INT         vSpeed
)
```

Parameters

handle
[in]设备句柄。

channel
[in] 通道号

index
[in] 预置位标号，取值范围 1 到 255

hSpeed
[in] 水平方向移动速度，取值范围 0 到 100

vSpeed
[in] 垂直方向移动速度，取值范围 0 到 100

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcPTZGotoPreset
删除机械云台预置位

```
INT ipcPTZClearPreset(
 ipcHandle   handle,
INT                index,
)
```

Parameters

handle
[in]设备句柄。

index
[in] 预置位标号，取值范围 1 到 255

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcPTZGetAllPreset

获取机械云台预置位信息

```
INT ipcPTZGetAllPreset (
 ipcHandle handle,
DWORD count,
LPIPC_PRESET preset
)
```

Parameters

handle

[in]设备句柄。

count

[in] preset 结构指针包含的预置位空间的数量　　此处与 ipcPTZGetCapacity 值相同为 255

preset

[in/out]预置位结构内存指针，调用者分配内存

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

通过此函数可以读取已经设置的预置位信息。读取到的预置位数量由 count 决定。

ipcPTZGetCapacity

获取机械云台预置位能力

```
INT ipcPTZGetCapacity (
 ipcHandle handle,
DWORD* capacity
)
```

Parameters

handle

[in]设备句柄。

preset

[out]预置位数量

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcGetPTZCruise

获取机械云台一条巡航路径

```
INT ipcGetPTZCruise(
 ipcHandle handle,
INT iCruiseId,
struct PTZCruiseParam *config,
INT* count)
```

Parameters

handle

[in]设备句柄。

iCruiseId

[in] 巡航路径 iCruiseId,0 表示获取全部巡航路径

config

[out] 存放巡航路径的结构体指针，该内存块应为支持的最多巡航路大小

count

[out]返回获取到的巡航路个数

PTZCruiseParam 结构说明

Return Values

失败返回-1,成功返回巡航参数结构体的个数

Remarks

ipcSetPTZCruise

设置机械云台一条巡航路径

```
INT ipcSetPTZCruise(
 ipcHandle handle,
INT dwCruiseCmd,
INT iCruiseId,
VOID *config
)
```

Parameters

handle
[in]设备句柄。

dwCruiseCmd
[in]巡航配置命令

iCruiseId
[in]巡航路径iCruiseId

config
[in]该结构体与命令对应

| dwCruiseCmd | iCruiseId | config | |
|---|---|---|---|
| SET_CRUISE_ALLDATA | 0 | PTZCruiseParam | 包含设置的巡航路径的全部信息 |
| SET_PRESET_CRUISE | 巡航路径id | PRESET_CRUISE_POINT_t | 要添加或者修改的巡航点信息 |
| DEL_PRESET_CRUISE | 巡航路径id | PRESET_CRUISE_POINT_t | 巡航点 仅预置点id有效 |
| CLEAR_CRUISE | 巡航路径id | NULL | |
| CLEAR_ALL_CRUISE | 巡航路径 | NULL | |

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcSetPTZCruiseControl

控制机械云台一条巡航路径的状态

```
INT ipcSetPTZCruiseControl(
 ipcHandle handle,
INT iCruiseId,
INT ctrl
)
```

Parameters

handle

[in]设备句柄。

iCruiseId

[in] 巡航路径 id

ctrl

[in] 开启或关闭巡航路径，0 表示关闭，1 表示开始

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcSetPTZTrans

机械云台透明数据,使用该接口能直接通过设备将云台控制码信息直接传输给云台设备，而无需配置解码器

```
INT ipcSetPTZTrans(
 ipcHandle     handle,
 CHAR *         ptzcode,
 INT                size
)
```

Parameters

handle
[in]设备句柄。

ptzcode
[in] 数据缓冲指针

size
[in] 数据大小

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcSetPTZTransProp

设置机械云台透明通道的属性[协议,地址,波特率...]

```
INT ipcSetPTZTransProp (
 ipcHandle handle,
DWORD protocol_type,
DWORD address,
CONST LPIPC_COMM_PROP prop
)
```

Parameters

handle

[in]设备句柄。

protocol_type

[in] 协议类型[1,2] : PROTOCOL_D = 1,PROTOCOL_P = 2

透明通道时填写 1

address

[in]起始地址[0,255]

prop

[in]云台配置信息

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcDPTZControl**

控制数字云台

```
INT ipcDPTZControl(
 playHandle      hPlayer,
 INT                  DPTZCommand,
 INT                  value,
 const CHAR*  buff
)
```

**Parameters**

hPlayer

[in] 预览句柄。

DPTZCommand

[in] 数字云台控制码

value

[in] 控制参数

buff

[in] 预置位的设定需要传入字符串的话使用此参数。

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

1) IPC_CMD_DPTZ_ZOOM : 控制数字云台视角　　缩放范围: 0~11
2) IPC_CMD_DPTZ_UP :数字云台上移　　步长范围: 0~1000
3) IPC_CMD_DPTZ_DOWN :数字云台下移　　步长范围: 0~1000
4) IPC_CMD_DPTZ_LEFT :数字云台左移　　步长范围: 0~1000
5) IPC_CMD_DPTZ_RIGHT :数字云台右移　　步长范围: 0~1000
6) IPC_CMD_DPTZ_SET_PRESET :设置数字云台预置位　　预置位范围: 1~9
7) IPC_CMD_DPTZ_CLEAR_PRESET :清理指定数字云台预置位　　预置位范围: 1~9
8) IPC_CMD_DPTZ_GOTO_PRESET :转到数字云台预置位　　预置位范围: 1~9
9) IPC_CMD_DPTZ_CLEAR_ALL_PRESET :清理所有数字云台预置位
10) IPC_CMD_DPTZ_GOTO_HOME :移到数字云台起始位
11) IPC_CMD_DPTZ_FOCUS_FAR :数字聚焦远　　开关标志 0:on, 1:off
12) IPC_CMD_DPTZ_FOCUS_NEAR :数字聚焦近　　开关标志 0:on, 1:off

参见：名为 IPCVideoDemo 的 sample

### ipcPTZUpleft

控制机械云台左上移动

```
INT ipcPTZUpleft(
ipcHandle handle,
int control,
int hSpeed,
int vSpeed)
```

#### Parameters

handle
[in]设备句柄。

control
[in] 开关控制，：开始移动 1：停止移动

hSpeed
[in] 水平移动速度[1~ 63]

vSpeed
[in] 垂直移动速度[1~ 63]

#### Return Values

错误代码 0 表示成功 ,小于 0 失败

#### Remarks

### ipcPTZDownleft

控制机械云台左下移动

```
INT ipcPTZDownleft(
ipcHandle handle,
int control,
int hSpeed,
int vSpeed)
```

**Parameters**

handle
[in]设备句柄。

control
[in] 开关控制，：开始移动 1：停止移动

hSpeed
[in] 水平移动速度[1~ 63]

vSpeed
[in] 垂直移动速度[1~ 63]

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcPTZUpright**
控制机械云台右上移动

```
INT ipcPTZUpright(
ipcHandle handle,
int control,
int hSpeed,
int vSpeed)
```

**Parameters**

handle
[in]设备句柄。

control
[in] 开关控制，：开始移动 1：停止移动

hSpeed
[in] 水平移动速度[1~ 63]

vSpeed

[in] 垂直移动速度[1~ 63]

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcPTZDownright

控制机械云台右下移动

```
INT ipcPTZDownright(
ipcHandle handle,
int control,
int hSpeed,
int vSpeed)
```

Parameters

handle
[in]设备句柄。

control
[in] 开关控制，：开始移动 1：停止移动

hSpeed
[in] 水平移动速度[1~ 63]

vSpeed
[in] 垂直移动速度[1~ 63]

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcPTZPositionControl**
控制机械云台坐标

```
INT ipcPTZPositionControl (
 ipcHandle                              handle,
INT                                         cmdType,
LPIPC_PTZ_POSITION_PARAM  pPosition
)
```

**Parameters**

handle
[in]设备句柄

cmdType
[in] 控制命令类型如
IPC_CMD_PTZ_GOTO_REL_POSITION　　从当前坐标位置运动指定的偏移量

IPC_CMD_PTZ_GOTO_ABS_POSITION　　运动到指定的绝对坐标位置

IPC_CMD_PTZ_GET_ABS_POSITION　　获取云台当前的绝对坐标

pPostion
[in] 坐标参数，参考 IPC_PTZ_POSITION_PARAM 说明

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

目前控制命令类型有：
IPC_CMD_PTZ_GOTO_REL_POSITION
IPC_CMD_PTZ_GOTO_ABS_POSITION
IPC_CMD_PTZ_GET_ABS_POSITION

**ipcPTZSelZoomIn**
控制机械云台区域缩放

```
LONG ipcPTZSelZoomIn
(
ipcHandle                               handle,
LONG                                      lChannel,
LPIPC_POINT_FRAME    pStruPointFrame
)
```

**Parameters**

handle
[in]设备句柄

lChannel
[in] 码流 id（默认填 0，保留）

pStruPointFrame
[in] 坐标参数结构，参考 IPC_POINT_FRAME 说明

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

## 3.7 报警中心上传

ipcSetAlarmCenterCallBack

注册报警回调函数

```
INT ipcSetAlarmCenterCallBack(
AlarmCallBack    fAlarmCallBack,
VOID *                pUser
) ;
```

Parameters

[in] fAlarmCallBack
　　报警信息会回调函数
[in] pUser
　　用户自定义数据

Return Values
　　成功返回 0，失败返回非0

```
typedef void(CALLBACK * AlarmCallBack)(LONG lType,ALARMER *pAlarmer,CHAR
*pAlarmInfo,DWORD dwBufLen,VOID *pUser);
```

lType
　　[out] 报警类型

| 报警类型 | 定义 |
| --- | --- |
| IPC_EVENT_CAMERATAMPERING | 视频异常报警 |
| IPC_EVENT_MOTIONDETECTION | 移动侦测报警 |
| IPC_EVENT_INPUTPORT | 外部输入报警 |
| IPC_EVENT_REBOOT | 将要重启报警 |

pAlarmer
　　[out] 报警设备信息
pAlarmInfo
　　[out] 报警信息 ALARMINFO
dwBufLen
　　[out] 报警信息缓存大小
pUser

[out] 用户数据

Return Values

ipcStartListen

监听本地，等待 IPC 主动链接。IPC 可以主动的方式连接一个或多个 SDK 主机，连接的主机地址在 IPC 报警中心参数中设置

```
INT ipcStartListen(
 const CHAR *   ip,
INT                    port
) ;
```

[in] ip

报警中心侦听的地址,null 则使用本机 ip 地址

[in] port

报警中心侦听的端口

Return Values

成功返回 0，失败返回非 0

ipcStopListen

停止报警中心监听

```
INT ipcStopListen()
```

Return Values

失败返回非 0

## 3.8 事件管理

ipcGetCameraUnusualEvent

获取视频异常事件

```
INT ipcGetCameraUnusualEvent(
 ipcHandle          handle,
    DWORD*           delayTime,
    ActionOptionInfo*   action_option
```

```
)
```

**Parameters**

handle
[in] 设备操作句柄。

delayTime
[out] 延时多少时间再进行报警，以秒为单位

action_option
[out] 动作响应设置信息

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcSetCameraUnusualEvent**
设置视频异常事件

```
INT ipcSetCameraUnusualEvent(
 ipcHandle        handle,
 DWORD          delayTime,
    ActionOptionInfo  action_option
)
```

**Parameters**

handle
[in] 设备操作句柄。

delayTime
[in] 延时多少时间再进行报警，以秒为单位

action_option
[in] 响应动作设置，见 **ActionOptionInfo** 数据结构

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

**ipcGetMotionAction**
获取移动侦测报警联动配置信息

```
INT ipcGetMotionEventAction(
 ipcHandle                          handle,
ActionOptionInfo*                   action_option
)
```

Parameters

handle
[in] 设备操作句柄。
action_option
[out] 响应动作设置，ActionOptionInfo 数据结构的指针

Return Values

Return Values

错误代码 0 表示成功 ，小于 0 失败

Remarks

ipcSetMotionEventAction
设置移动侦测报警联动配置信息

```
INT ipcSetMotionEventAction (
 ipcHandle                          handle,
ActionOptionInfo                    action_option
)
```

Parameters

handle
[in] 设备操作句柄。
action_option
[in] 响应动作设置，见 ActionOptionInfo 数据结构
Return Values

错误代码 0 表示成功 ，小于 0 失败

Remarks

ipcGetMotionWndInfo
获取移动侦测区域配置信息

```
INT ipcGetMotionWndInfo (
 ipcHandle handle,
 MotionDetectionWindowInfoEx * mdi,
)
```

Parameters

handle
[in] 设备操作句柄。

mdi
[out] 移动侦测区域信息

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks
mdi 为 MotionDetectionWindowInfoEx[4]一维数组的指针

**ipcSetMotionWndInfo**
设置移动侦测区域配置信息

```
INT ipcSetMotionWndInfo (
 ipcHandle handle,
 MotionDetectionWindowInfoEx mdi[4],
)
```

**Parameters**

handle
[in] 设备操作句柄。

mdi
[in] 移动侦测区域信息

**Return Values**

错误代码 0 成功  小于 0 失败

**Remarks**
mdi 为 MotionDetectionInfo [4]为 MotionDetectionInfo 的一维数组

**ipcGetInputEvent**
获取报警输入事件

```
INT ipcGetInputEvent (
 ipcHandle            handle,
    DWORD*             level,
 ActionOptionInfo*     action_option
)
```

**Parameters**

handle
[in] 设备操作句柄

level
[out] 报警电平，0：低电平，1：高电平

action_option
[out] 响应动作设置，**ActionOptionInfo** 数据结构的指针

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

**ipcSetInputEvent**
设置报警输入事件

```
INT ipcSetInputEvent (
 ipcHandle           handle,
    DWORD             level,
 ActionOptionInfo     action_option
)
```

**Parameters**

handle
[in] 设备操作句柄。

level
[in] 报警电平，0：低电平，1：高电平

action_option
[in] 响应动作设置，见 **ActionOptionInfo** 数据结构

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

ipcGetInputPortNumber
获取报警输入端口数

```
INT ipcGetInputPortNumber (
 ipcHandle      handle,
    INT *         number,
);
```

Parameters

handle
[in] 设备操作句柄

number
[out] 报警输入端口数 ，目前只支持 1 个端口

Return Values

正确返回值大于等于 0，错误返回值小于 0

Remarks

**ipcGetInputPortStatus**

获取报警输入端口状态

```
INT ipcGetInputPortStatus(
 ipcHandle      handle,
    INT *         status
);
```

**Parameters**

handle

[in] 设备操作句柄

status

[out] 端口状态, 目前仅支持 1 个端口

**Return Values**

正确返回值大于等于 0，错误返回值小于 0

**Remarks**

**ipcGetOutputPortNumber**

获取报警输出端口数

```
INT ipcGetOutputPortNumber(
 ipcHandle      handle,
    INT *         param
);
```

**Parameters**

handle

[in] 设备操作句柄

param

[out] 端口数,目前仅支持 1 个

**Return Values**

正确返回值大于等于 0，错误返回值小于 0

**Remarks**

**ipcGetInputPortTriggerStatus**
获取报警输入端口触发条件

```
INT ipcGetInputPortTriggerStatus(
 ipcHandle    handle,
    INT *       triggerStatus
);
```

**Parameters**

handle
[in] 设备操作句柄

triggerStatus
[out] 报警输入端口触发条件，0 为低电平，1 为高电平
目前仅支持 1 个

**Return Values**

正确返回值等于 0，错误返回值小于 0

**Remarks**

ipcSetInputPortTriggerStatus
设置报警输入端口触发条件

```
INT ipcSetInputPortTriggerStatus(
 ipcHandle  handle,
    INT       param
);
```

Parameters

handle
[in] 设备操作句柄

param
[in] 电平值，目前仅支持 1 个,0 低电平 1 高电平

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcGetOutputPortStatus**
获取报警输出端口状态

```
INT ipcGetOutputPortStatus(
 ipcHandle      handle,
    INT *         status
);
```

**Parameters**

handle
[in] 设备操作句柄

status
[out] 报警输出端口状态，0 为低电平，1 为高电平
目前仅支持 1 个

**Return Values**

正确返回值等于 0，错误返回值小于 0

**Remarks**

ipcSetOutputPortAction
设置输出端口报警联动行为

```
INT ipcSetOutputPortAction(
    ipcHandle handle,
    INT param
);
```

Parameters

handle
[in] 设备操作句柄

param
[in] 高低电平值

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcGetOutputPortAction
获取输出端口报警联动行为

```
INT ipcGetOutputPortAction(
    ipcHandle handle,
    INT* param
);
```

Parameters

handle
[in] 设备操作句柄

param
[in] 高低电平值

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

ipcSetOutputPortStatus
设置报警输出端口状态

```
INT ipcSetOutputPortStatus(
    ipcHandle handle,
    INT param
);
```

Parameters

handle
[in] 设备操作句柄

param
[in] 高低电平值
目前仅支持 1 个
Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcTriggerAlarmOutput**
触发 IPC 报警输出

```
INT ipcTriggerAlarmOutput (
ipcHandle handle,
IN LONG  port,
IN LONG  duration
) ;
```

**Parameters**

**Parameters**

handle
[in] 设备操作句柄

port
[in] 输出端口
目前仅支持1个, 索引值从零开始

duration
[in]输出持续时间
单位为毫秒

**Return Values**
错误代码 0表示成功,小于0失败

**Remarks**

**ipcSetEventEnable**

配置事件使能

```
INT ipcSetEventEnable (
ipcHandle                handle,
IN DWORD             dwEvent,
IN DWORD             dwEnable
) ;
```

**Parameters**

[in] handle

设备句柄

[in] dwEvent

报警事件，参考事件类型定义

[in] dwEnable

报警使能标志 0 禁用 1 启用

**Return Values**

成功返回 0，失败返回非0

**ipcGetEventEnable**

获取事件使能

```
INT ipcGetEventEnable(
ipcHandle               handle,
IN DWORD            dwEvent,
OUT DWORD*      dwEnable
) ;
```

**Parameters**

[in] handle

设备句柄

[in] dwEvent

报警事件，参考事件类型定义

[out] dwEnable

报警使能标志 0 禁用 1 启用

**Return Values**

成功返回 0，失败返回非0

ipcSetOutputPortEnable
启用禁用报警输出口

```
INT ipcSetOutputPortEnable(
ipcHandle handle,
IN DWORD dwEnable
)
```

Parameters

[in] handle
    设备句柄
 [in] dwEnable
    启用禁用标志：0 禁用 1 启用

Return Values
        成功返回 0，失败返回非0

ipcSetInputPortEnable
启用禁用报警输入口

```
INT ipcSetInputPortEnable(
ipcHandle handle,
IN DWORD dwEnable
)
```

Parameters

[in] handle
    设备句柄
 [in] dwEnable
    启用禁用标志：0 禁用 1 启用

Return Values
成功返回 0，失败返回非0

ipcGetInputPortEnable

获取报警输入口启用禁用配置

```
INT ipcGetInputPortEnable(
ipcHandle handle,
IN DWORD* dwEnable
)
```

Parameters

[in] handle

设备句柄

[out] dwEnable

启用禁用标志：0 禁用 1 启用

Return Values

成功返回 0，失败返回非0

ipcGetOutputPortEnable

获取报警输出口启用禁用配置

```
INT ipcGetOutputPortEnable(
ipcHandle handle,
IN DWORD* dwEnable
)
```

Parameters

[in] handle

设备句柄

[out] dwEnable

启用禁用标志：0 禁用 1 启用

Return Values

成功返回 0，失败返回非0

## 3.9 布防设置

**ipcGetCameraUnusualPlan**

获取视频异常布防信息

```
INT ipcGetCameraUnusualPlan(
 ipcHandle            handle,
DWORD**          start_time,
DWORD**          end_time
)
```

**Parameters**

handle

[in] 设备操作句柄。

start_time

[out] 开始时间的 7*16 二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

end_time

[out] 结束时间的 7*16 二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

ipcSetCameraUnusualPlan
设置定时录像布防信息

```
INT ipcSetCameraUnusualPlan (
 ipcHandle       handle,
 DWORD          start_time[7][16],
 DWORD          end_time[7][16]
)
```

Parameters

handle
[in] 设备操作句柄。

start_time[7][16]
[in] 开始时间的二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间
end_time[7][16]
[in] 结束时间的二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

Return Values

错误代码 0 表示成功 ，小于 0 失败

Remarks

**ipcGetMotionDetectionPlan**
获取移动侦测布防信息

```
INT ipcGetMotionDetectionPlan (
ipcHandle            handle,
DWORD**          start_time,
DWORD**          end_time
)
```

**Parameters**

handle
[in] 设备操作句柄。

start_time
[out] 开始时间的 7*16 二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间
end_time
[out] 结束时间的 7*16 二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

**ipcSetMotionDetectionPlan**

设置移动侦测布防信息

```
INT ipcSetMotionDetectionPlan (
 ipcHandle       handle,
 DWORD          start_time[7][16],
 DWORD          end_time[7][16]
)
```

**Parameters**

handle

[in] 设备操作句柄。

start_time[7][16]

[in] 开始时间的二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

end_time[7][16]

[in] 结束时间的二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

**ipcGetInputPlan**
获取报警输入布防

```
INT ipcGetInputPlan (
ipcHandle          handle,
DWORD**          start_time,
DWORD**          end_time
)
```

**Parameters**

handle
[in] 设备操作句柄。

start_time
[out] 开始时间的 7*16 二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间
end_time
[out] 结束时间的 7*16 二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

**Return Values**

错误代码 0 表示成功 ，小于 0 失败

**Remarks**

ipcSetInputPlan
设置报警输入布防

```
INT ipcSetInputPlan (
 ipcHandle       handle,
 DWORD          start_time[7][16],
 DWORD          end_time[7][16]
)
```

Parameters

handle
[in] 设备操作句柄。

start_time[7][16]
[in] 开始时间的二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间
end_time[7][16]
[in] 结束时间的二维数组，一维元素表示日期 0：星期日、1：星期一、2：星期二、3：星期三、4：星期四、5：星期五、6：星期六，二维元素表示时间段 00:00 到 23:59 之间

Return Values

错误代码 0 表示成功 ，小于 0 失败

Remarks

## 3.10 系统管理

ipcUpdateServiceStart
启动升级服务

```
INT ipcUpdateServiceStart(
 short port
);
```

Parameters

port
[in]服务端口号。

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks
此函数将在本地计算机上创建网络服务，端口号请勿使用已经被占用的端口。

ipcUpdateServiceStop

停止升级服务

```
INT ipcUpdateServiceStop(

);
```

Parameters

Return Values

错误代码 0 表示成功 ,小于 0 失败

Remarks

**ipcUpgrade**

设备升级

```
INT ipcUpgrade(
 ipcHandle handle,
 const CHAR * inPathFileName
);
```

**Parameters**

handle

[in]设备操作句柄

inPathFileName

[in]需要升级的文件地址

**Return Values**

错误代码 0 表示成功 ,小于 0 失败

**Remarks**

建议不要使用目录层级过深的路径以免数据超长被截断

ipcSystemRestore
恢复到出厂设置。将所有参数恢复到出厂设置

```
INT ipcSystemRestore(
    ipcHandle handle
) ;
```

Parameters

handle
[in]设备句柄

Return Values

大于 0 表示成功，小于 0 表示失败

Remarks

**ipcDeviceRestart**

实现设备重启功能

```
INT ipcDeviceRestart (
    ipcHandle handle
) ;
```

**Parameters**

handle

[in]设备句柄

**Return Values**

大于 0 表示成功，小于 0 表示失败

**Remarks**

**ipcDeviceStandby**

实现设备待机功能

```
INT ipcDeviceStandby(
    ipcHandle handle
) ;
```

**Parameters**

handle

[in]设备句柄

**Return Values**

大于 0 表示成功，小于 0 表示失败

**Remarks**

**ipcDeviceWakeup**
唤醒设备

```
INT ipcDeviceWakeup (
    ipcHandle handle
) ;
```

**Parameters**

handle
[in]设备句柄

**Return Values**

大于 0 表示成功，小于 0 表示失败

**Remarks**

ipcSetHeartbeat

设置心跳间隔，需已经建立网络连接。

```
INT ipcSetHeartbeat(
    ipcHandle handle,
    DOWRD interval,
    ipcHeartBeatCallback callback,
    LPVOID context
) ;
```

Parameters

handle
[in]设备句柄

interval
[in]snmp 心跳间隔（不小于 5000 毫秒）单位：毫秒

callback
[in]回调函数

context
[in]
用户参数

Return Values

大于 0 表示成功，小于 0 表示失败

Remarks

ipcSetHeartbeatEx

设置心跳间隔

```
INT ipcSetHeartbeatEx (
    ipcHandle handle,
    DOWRD interval,
    CONST CHAR* ip,
    ipcHeartBeatCallback callback,
    LPVOID context
) ;
```

Parameters

handle
[in]设备句柄

interval
[in]snmp 心跳间隔，（不小于 5000 毫秒）单位：毫秒

ip
[in]设备 ip 地址

callback
[in]回调函数

context
[in]
用户参数

Return Values

大于 0 表示成功，小于 0 表示失败

Remarks

相对于非扩展版本，此函数可以在 sdk 与设备未建立时使用。

ipcHeartBeatCallback

心跳回调通知

```
VOID ipcHeartBeatCallback(
    ipcHandle handle,
    BOOL isOnline,
    LPVOID context
) ;
```

Parameters

handle

[in]

设备句柄

isOnline

[in]

是否在线

context

[in]

心跳包中的信息

Return Values

Remarks

待定义

## 3.11 音频监听

**ipcLiveFrame_CallBackEx**
音视频解码前数据捕获回调函数

```
VOID (WINAPI *ipcLiveFrame_CallBackEx)( LPVOID hStream, const FrameInfoEx*
avFrameInfo , LPVOID lpContext ) ;
```

**Parameters**

hStream
句柄(传回的音视频hStream不同，可根据当前帧类型判断当前返回的是音频hStream,还是视频handle。
视频为播放句柄(playHandle) ，音频为设备句柄(ipcHandle)

avFrameInfo
音视频数据结构

lpContext
用户数据

**Return Values**
返回：无含义

**Remarks**
若需要区分码流来自哪个设备，视频请根据播放句柄来对应，音频请根据设备句柄来对应即可（需调用方自行映射hStream的对应关系）

**ipcAVStreamCallBackEx2**
音视频解码前数据捕获回调扩展2函数

```
VOID (WINAPI *ipcAVStreamCallBackEx2)( LPVOID hStream, long lType,const unsigned char
*pBuf,long lSize,LPVOID pContext ) ;
```

**Parameters**

hStream
句柄(传回的音视频hStream不同，可根据当前帧类型判断当前返回的是音频hStream,还是视频handle。
视频为播放句柄(playHandle) ，音频为设备句柄(ipcHandle)

lType
帧类型，H流头 ,视频I B P,音频A

pBuf
数据，第一次回调必为流头(参考结构体 IPC_STREAM_HEADER),之后根据帧类型判断(数据结构(
参考结构体IPC_VIDEOFRAME_HEADER、IPC_AUDIOFRAME_HEADER)+数据+帧尾)

lSize
数据长度

pContext

用户数据

**Return Values**
无

**Remarks**
若需要区分码流来自哪个设备，视频请根据播放句柄来对应，音频请根据设备句柄来对应即可（需调用方自行映射hStream的对应关系）

**ipcSetAVCallbackEx2**
设置音视频数据流扩展回调

```
INT ipcSetAVCallbackEx2 (
playHandle                              handle ,
ipcAVStreamCallBackEx2  func,
LPVOID                                  lpContext
)
```

**Parameters**
handle
[in]播放句柄

func
[in]回调函数

lpContext
[in]回调参数

**Return Values**
成功返回0，小于0为失败。
Remarks
视频和音频一起使用时只需要设置1次即可生效。只有在视频和音频都停止后才可反注册此回调，以免导致无法收到回调数据。

注册回调函数的区别
1）ipcRegisterLiveFrameCallBack
只在回调的IPCVIDEO_CALLBACK_INFO结构中返回视频数据和视频相关信息，无时间戳。

2）ipcSetAVCallbackEx
在回调函数的FrameInfoEx结构中返回音视频数据并提供时间戳，需根据标志来区分是音频数据或视频数据

3）ipcSetAVCallbackEx2
在回调数据之前包含了头信息等IPC流文件播放及检索用到的相关数据，可按照格式拆分出带时间戳的音频或视频数据，此数据直接写文件可生成IPC流播放文件
格式参见IPC_STREAM_HEADER，IPC_VIDEOFRAME_HEADER，IPC_AUDIOFRAME_HEADER，IPC_FRAME_TAIL

ipcSetAVCallbackEx

设置音视频数据流回调

```
INT ipcSetAVCallbackEx (
playHandle                              handle ,
ipcLiveFrame_CallBackEx  func,
LPVOID                                   lpContext
)
```

Parameters

handle
[in]播放句柄

func
[in]回调函数

lpContext
[in]回调参数

Return Values

成功返回0，小于0为失败。

Remarks

视频和音频一起使用时只需要设置1次即可生效。只有在视频和音频都停止后才可反注册此回调，以免导致无法收到回调数据。

注册回调函数的区别

1）ipcRegisterLiveFrameCallBack
只在回调的IPCVIDEO_CALLBACK_INFO结构中返回视频数据和视频相关信息，无时间戳。

2）ipcSetAVCallbackEx
在回调函数的FrameInfoEx结构中返回音视频数据并提供时间戳，需根据标志来区分是音频数据或视频数据

3）ipcSetAVCallbackEx2
在回调数据之前包含了头信息等IPC流文件播放及检索用到的相关数据，可按照格式拆分出带时间戳的音频或视频数据，此数据直接写文件可生成IPC流播放文件
格式参见IPC_STREAM_HEADER，IPC_VIDEOFRAME_HEADER，IPC_AUDIOFRAME_HEADER，IPC_FRAME_TAIL

ipcCreateAudioLivePlay
创建音频监听资源

```
INT ipcCreateAudioLivePlay (
ipcHandle                  handle,
int                              device_in_port,
int                              device_out_port,
)
```

Parameters
handle
[in] 设备句柄

device_in_port
[in] 音频通信端口（IPC->PC） 默认填0

device_out_port

[in] 音频通信端口（PC->IPC） 默认填0
bRequestRemoteData

Return Values
成功返回0，小于0为失败。

ipcStartAudioLivePlay
开始音频监听

```
INT ipcStartAudioLivePlay(
ipcHandle handle
) ;
```

Parameters
handle
[in] 设备句柄

Return Values
成功返回0，小于0为失败。

Remarks

音频编码禁用时不能音频监听。
音频编码禁用后再启用时，已经存在音频监听需要重新连接。参考函数 ipcSetAudioState

ipcStopAudioLivePlay
停止音频监听

```
INT ipcStopAudioLivePlay(
ipcHandle handle
)
```

Parameters

handle
[in] 设备句柄

Return Values
成功返回0，小于0为失败。

ipcDestroyAudioLivePlay

释放音频监听资源

```
INT ipcDestroyAudioLivePlay(
ipcHandle handle
)
```

Parameters
handle
[in] 设备句柄

Return Values
成功返回0，小于0为失败。

ipcSetAudioDeviceConfig

配置 IPC 音频参数

```
INT ipcSetAudioDeviceConfig(
ipcHandle    handle ,
LPAUDIOCONF  lpAudioConf
)
```

Parameters
handle
[in] 设备句柄

lpAudioConf
[in] 音频数据结构

Return Values
成功返回0，小于0为失败。

ipcGetAudioDeviceConfig

获取 IPC 音频参数

```
INT ipcGetAudioDeviceConfig(
ipcHandle    handle ,
LPAUDIOCONF  lpAudioConf
)
```

Parameters
handle
[in] 设备句柄

lpAudioConf
[out] 音频数据结构

**Return Values**

成功返回0，小于0为失败。

### ipcSetAudioLivePlayControl

设置音频监听的控制参数

```
INT ipcSetAudioLivePlayControl(
ipcHandle handle,
BOOL      bEnableIPCData,
BOOL      bEnablePlayer
)
```

### Parameters

handle
[in] 设备句柄

bEnableIPCData
[in] 控制标志，接收IPC声音

bEnablePlayer
[in] 控制标志，播放IPC声音

### Return Values

成功返回0，小于0为失败。

### Remark

ipcSetAudioLivePlayControl(TRUE,TRUE) ;表示自动播放IPC的监控数据
ipcSetAudioLivePlayControl(TRUE,FALSE) ;表示只通过回调获取IPC的监控数据
ipcSetAudioLivePlayControl(FALSE,TRUE) ;此方式不支持
ipcSetAudioLivePlayControl(FALSE, FALSE) ;禁用监控的功能（此方式通常为配合ipcSetTalkControl使用）

### ipcGetAudioLivePlayControl

获取音频监听的控制参数

```
INT ipcGetAudioLivePlayControl(
ipcHandle handle,
BOOL* bEnableIPCData,
BOOL* bEnablePlayer
)
```

#### Parameters

handle
[in] 设备句柄

bEnableIPCData
[out] 控制标志，接收IPC声音

bEnablePlayer
[out] 控制标志，播放IPC声音

#### Return Values

成功返回0，小于0为失败。

#### Remark

### ipcSetAudioLivePlayMute

静音音频监听

```
INT ipcSetAudioLivePlayMute (
ipcHandle handle,
BOOL bMute
)
```

#### Parameters

handle
[in] 设备句柄

bMute
[in] 是否静音

#### Return Values

成功返回0，小于0为失败。

#### Remark

### ipcGetAudioLivePlayMute

获取音频监听的静音标志

```
INT ipcGetAudioLivePlayMute (
ipcHandle handle,
BOOL* bMute
)
```

### Parameters

handle
[in] 设备句柄

bMute
[out] 是否静音

### Return Values

成功返回0，小于0为失败。

### Remark

## 3.12 语音对讲

**ipcStartTalk**
开始语音对讲

```
INT ipcStartTalk(
ipcHandle            handle,
)
```

**Parameters**

handle
[in]设备句柄

**Return Values**

成功返回0，小于0为失败。

**Remarks**

音频编码禁用时不能语音对讲。
音频编码禁用后再次启用时，已经存在语音对讲需要重新连接。参考函数 **ipcSetAudioState**

**ipcStopTalk**
停止语音对讲

```
INT ipcStopTalk(
ipcHandle            handle,
)
```

**Parameters**

handle
[in]设备句柄

**Return Values**
成功返回0，小于0为失败。

**ipcTalkCallback**
音频数据回调函数

```
INT (WINAPI * ipcTalkCallback)(ipcHandle hAudio, const unsigned char* Data ,INT iDataLen,LPVOID lpContext)
```

**Parameters**

hAudio
[out] 设备句柄

Data
[out]数据指针 ， 回调数据

iDataLen
[out]数据长度 ， 回调数据长度

lpContext
[out]回调参数 ， 用户传入的自定义参数

**Return Values**
暂时忽略默认返回0

**Remarks**

语音对讲中回调本地未编码数据，采样率固定。

**ipcSetTalkCallback**
设置语音对讲回调

```
INT ipcSetTalkCallback (
ipcHandle            handle,
ipcTalkCallback func,
LPVOID                lpContext
)
```

**Parameters**

handle
[in]设备句柄

func
[in] 未编码数据，格式: SamplesPerSec:8000,BitsPerSample:16,Channels:1,AvgBytesPerSec:16000

lpContext
[in]

**Return Values**
成功返回0，小于0为失败。

**ipcGetTalkControl**
获取语音对讲的控制参数

```
INT ipcGetTalkControl (
ipcHandle            handle,
BOOL*                 bEnableCapture,
BOOL*                 bEnableTransfer
)
```

**Parameters**

handle
[in]设备句柄

bEnableCapture
[out]是否允许采集本机音频

bEnableTransfer
[out]是否允许传输到IPC端

**Return Values**
成功返回0，小于0为失败。

**ipcSetTalkControl**
设置语音对讲的控制参数

```
INT ipcSetTalkControl (
ipcHandle         handle,
BOOL                 bEnableCapture,
BOOL                 bEnableTransfer
)
```

**Parameters**

handle
[in]设备句柄

bEnableCapture
[in]是否允许采集本机音频

bEnableTransfer
[in]是否允许传输到IPC端

**Return Values**
成功返回0，小于0为失败。

**Remark**

ipcSetTalkControl(TRUE,TRUE) ;//自动采集本机音频并发送到IPC端
ipcSetTalkControl(TRUE,FALSE) ;//只采集本机音频
ipcSetTalkControl(FALSE,TRUE) ;//只发送音频到IPC端（平台转发用）
ipcSetTalkControl(FALSE, FALSE) ;//禁用PC-->IPC通路（配合ipcSetAudioLiveControl使用）

**ipcSendAudioStream**
发送编码后音频数据到IPC

```
INT ipcSendAudioStream (
ipcHandle       handle,
int                     type,
unsigend char* data,
unsigned int      data_len
)
```

**Parameters**

handle
[in] 设备句柄

type
[in] 类型 1表示 711 ,0表示 amr(暂不支持amr)

data
[in] 数据 AMR为SamplesPerSec:8000,BitsPerSample:16,Channels:1,AvgBytesPerSec:16000
G711为SamplesPerSec:8000,BitsPerSample:8,Channels:1,AvgBytesPerSec:8000
data_len
[in] 数据长度

**Return Values**
成功返回0，小于0为失败。

**ipcEncodeAudioStream**
将音频数据进行编码

```
INT ipcEncodeAudioStream(
ipcHandle handle ,
int               type,
unsigned char* src_data,
unsigned int      src_data_len,
unsigned char* enc_data,
unsigned int*    enc_data_len
)
```

**Parameters**

handle
[in] 设备句柄

type
[in] 类型1表示 711 ,0表示 amr(暂不支持amr)

src_data

[in] 源音频数据 AMR为SamplesPerSec:8000,BitsPerSample:16,Channels:1,AvgBytesPerSec:16000
G711为SamplesPerSec:8000,BitsPerSample:8,Channels:1,AvgBytesPerSec:8000

src_data_len
[in] 源数据长度

enc_data
[out]编码后数据

enc_data_len
[out]编码后数据长度

**Return Values**

成功返回0，小于0为失败。

**ipcEnumAudioDevice**
枚举声卡设备

```
INT ipcEnumAudioDevice (
   ipcHandle     handle,
  AudioDevice* pstDevice,
  LONG*        pSize
);
```

**Parameters**

handle
[in] 设备句柄

pstDevice
[in/out] 声卡设备信息，如果为NULL，则 pSize 返回声卡设备个数

pSize
[in/out]  返回声卡设备个数

**Return Values**
成功返回0，小于0为失败。

**Remarks**
使用时需要第一次调用pstDevice传递NULL，获取个数，根据pSize返回的个数跟配内存，再次调用函数获取设备信息。

**ipcSetAudioDevice**
设置语音对讲声卡设备

```
INT ipcSetAudioDevice
 (
  ipcHandle     handle
    AudioDevice  stDevice
);
```

**Parameters**

handle
[in] 设备句柄

stDevice
[in] 声卡设备信息

**Return Values**
成功返回0，小于0为失败。

**Remarks**

该函数需要在语音对讲开始前调用生效

**ipcGetAudioDevice**

获取语音对讲声卡设备

```
INT ipcGetAudioDevice
 (
  ipcHandle      handle
    AudioDevice*  pstDevice
);
```

**Parameters**

handle

[in] 设备句柄

pstDevice

[out] 获取语音对讲使用的声卡设备信息，用户未设置语音对讲声卡设备是返回缺省声卡设备

**Return Values**

成功返回0，小于0为失败。

**Remarks**

## 3.13 日志

**ipcExportLog**
导出设备日志

```
INT ipcExportLog(
ipcHandle handle,
const WCHAR* lpFilename
) ;
```

**Parameters**

handle
[in]设备句柄

lpFilename
[in] 日志存放文件全路径

**Return Values**

0 表示成功，小于 0 表示失败

**Remarks**

**ipcFindLog**
查找设备的日志信息

```
LONG ipcFindLog(
  ipcHandle         handle,
  DWORD             dwLogType,
  LPIPC_TIME        lpStartTime,
  LPIPC_TIME        lpStopTime,
```

```
)
```

**Parameters**

handle
[in] ipcCreateDevice 的返回值

dwLogType
[in] 日志类型：0－全部；1－访问日志；2－报警日志；4－系统日志，可按位组合

lpStartTime
[in] 开始时间，传 NULL,将忽略该参数

lpStopTime
[in]结束时间，传 NULL,将忽略该参数

**Return Values**
-1 表示失败，其值作为 ipcFindNextLog 等函数的参数

**Remarks**

**ipcFindNextLog**
逐条获取查找到的日志信息。

```
LONG ipcFindNextLog(
  LONG                lLogHandle,
  LPIPC_LOG     lpLogData
)
```

Parameters

lLogHandle
[in]  日志查找句柄，ipcFindLog 的返回值

lpLogData
[out]保存日志信息的指针

Return Values
-1 表示失败，其他值表示当前的获取状态等信息（参考日志查找结果宏定义）。

Remarks
在调用该接口获取查找日志之前，必须先调用 ipcFindLog 得到当前的查找句柄。

ipcFindLogClose
释放查找日志的资源。

```
BOOL ipcFindLogClose(
  LONG   lLogHandle
)
```

Parameters

lLogHandle

[in] 日志查找句柄，ipcFindLog()的返回值

Return Values

0 表示成功，-1 表示失败。

ipcSetLogConfg

设置日志保存策略。

```
BOOL ipcSetLogConfig(
  ipcHandle          handle,
 LPIPC_LOG_CONFIG    lpLogConfig
);
```

Parameters

[in]  handle
ipcCreateDevice 的返回值

[in] lpLogConfig
 日志配置信息

Return Values
0 表示成功，-1 表示失败

ipcGetLogConfg
获取日志保存策略。

```
BOOL ipcGetLogConfig(
  ipcHandle          handle,
 LPIPC_LOG_CONFIG    lpLogConfig
);
```

**Parameters**

handle
[in]  ipcCreateDevice 的返回值

lpLogConfig
[in] 日志配置信息

**Return Values**
0 表示成功，-1 表示失败

**ipcDeleteLog**
删除日志

```
INT ipcDeleteLog(
 ipcHandle              handle,
  DWORD                 dwLogType,
  LPIPC_TIME        lpStartTime,
  LPIPC_TIME        lpStopTime,
) ;
```

**Parameters**

handle
[in] ipcCreateDevice 的返回值

dwLogType
[in] 日志类型：0－全部；1－访问日志；2－报警日志；4－系统日志，可按位组合

lpStartTime
[in] 开始时间，传 NULL,将忽略该参数

lpStopTime
[in] 结束时间，传 NULL,将忽略该参数

**Return Values**

大于 0 表示成功，小于 0 表示失败

**Remarks**

**ipcDeleteLogById**

删除日志

```
INT ipcDeleteLog(
 ipcHandle          handle,
  INT*                    pID,
  INT                      nCount,
) ;
```

**Parameters**

handle

[in]  ipcCreateDevice 的返回值

pID

[in]日志 ID 数组

nCount

[in] pID 里面包含的日志 ID 个数

**Return Values**

大于 0 表示成功，小于 0 表示失败

**Remarks**

**ipcClearLog**

清空日志

```
INT ipcClearLog(
 ipcHandle          handle,
 ) ;
```

**Parameters**

handle

[in] ipcCreateDevice 的返回值

**Return Values**

大于 0 表示成功，小于 0 表示失败

**Remarks**

**ipcQueryLogCount**

查找符合条件的日志信息条数

```
INT ipcQueryLogCount(
  ipcHandle         handle,
  DWORD             dwLogType,
  LPIPC_TIME        lpStartTime,
  LPIPC_TIME        lpStopTime,
```

```
  INT*                          pRetCount
);
```

**Parameters**

handle
[in]  ipcCreateDevice 的返回值

dwLogType
[in]日志类型：0－全部；1－访问日志；2－报警日志；4－系统日志，可按位组合

lpStartTime
[in]开始时间，传 NULL,将忽略该参数

lpStopTime
[in]结束时间，传 NULL,将忽略该参数

pRetCount
[out]返回符合条件的日志条数

**Return Values**
-1 表示失败，0 成功

**Remarks**

## 3.14 SD卡录像

**ipcRecordControl**

控制操作录像文件

```
INT ipcRecordControl (
ipcHandle handle,
DWORD dwControlCode,
char* lpInBuffer,
DWORD dwInLen,
char* lpOutBuffer,
DWORD* lpOutLen
) ;
```

**Parameters**

handle

[in] ipcCreateDevice 的返回值

dwControlCode

[in] 控制录像命令， 见下表

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_RECORDSTART | 1 | 开始录像 |
| IPC_RECORDSTOP | 2 | 停止录像 |
| IPC_DELETEFILE | 3 | 删除录像文件 |
| IPC_LOCKFILE | 4 | 锁定录像文件（暂不支持） |
| IPC_UNLOCKFILE | 5 | 解锁录像文件（暂不支持） |
| IPC_GET_RECORDCONFIG | 6 | 获取录像文件分割大小信息 |
| IPC_SET_RECORDCONFIG | 7 | 设置录像文件分割大小 |

lpInBuffer

[in] 指向输入参数的指针

dwInLen

[in] 输入参数的长度

lpOutBuffer

[out] 指向输出参数的指针

lpOutLen

[out] 输出参数的长度

**Return Values**

大于等于 0 表示成功，小于 0 表示失败

**Remarks**

该接口中的第三个参数和第五个参数是否需要输入数值与控制命令有关，详见下表所示

| 状态命令宏定义 | 状态命令说明 | lpInBuf | lpOutBuf |
|---|---|---|---|
| IPC_RECORDSTART | 开始录像 | IPC_RECORD_CTRL 的指针 | 无 |
| IPC_RECORDSTOP | 停止录像 | IPC_RECORD_CTRL 的指针 | 无 |
| IPC_DELETEFILE | 删除录像文件 | 长度小于 128 个字符串的录像文件名 | 无 |
| IPC_GET_RECORDCONFIG | 获取录像文件分割大小信息 | 无 | IPC_RECORD_CONFIG 的指针 |
| IPC_SET_RECORDCONFIG | 设置录像文件分割大小 | IPC_RECORD_CONFIG 的指针 | 无 |

**ipcFindRecordFile**

根据文件类型、时间查找设备录像文件

```
INT ipcFindRecordFile (
ipcHandle handle,
LONG lChannel,
DWORD dwFileType,
LPIPC_TIME lpStartTime,
LPIPC_TIME lpStopTime
)
```

**Parameters**

handle

[in] ipcCreateDevice 的返回值

lChannel

[in] 通道号

dwFileType

[in] 要查找的文件类型：

位与结果为 0 表示不启用，1 表示启用

dwFileType & 0x0001，表示是否查询定时录像文件

dwFileType & 0x0002，表示是否查询手动录像文件

dwFileType & 0x0004，表示是否查询报警录像文件

dwFileType & 0x0008，表示是否查询移动桢测录像文件

dwFileType & 0x0010，表示是否查询断网录像文件

dwFileType & 0x0020，表示是否查询视频异常录像文件

lpStartTime

[in] 文件的开始时间

lpStopTime

[in] 文件的结束时间
返回：

**Return Values**
小于 0 表示失败，其他值作为 ipcFindRecordClose 等函数的参数

**Remarks**

**ipcFindNextRecordFile**
查找设备的日志信息

```
INT ipcFindNextRecordFile(
LONG lFindHandle,
LPIPC_FIND_DATA lpFindData
)
```

**Parameters**

lFindHandle
[in] 文件查找句柄，ipcFindRecordFile 的返回值
lpFindData
[out] 保存文件信息的指针，详情见 IPC_FIND_DATA

**Return Values**
小于 0 表示失败，其他值表示当前的获取状态等信息如获取文件信息成功、没有更多的文件，查找结束、查找文件时异常，详见下表

| 宏定义 | 宏定义值 | 含义 |
|---|---|---|
| IPC_FINDFILE_SUCCESS | 1000 | 获取录像文件信息成功 |
| IPC_FINDFILE_NOFIND | 1001 | 未查找到录像文件 |
| IPC_FINDFILE_ISFINDING | 1002 | 正在查找请等待 |
| IPC_FINDFILE_NOMORERECORD | 1003 | 没有更多的录像文件，查找结束 |
| IPC_FINDFILE_EXCEPTION | 1004 | 查找录像文件时异常 |

**Remarks**

ipcFindRecordClose
关闭文件查找，释放资源

```
INT ipcFindRecordClose(
LONG lFindHandle
)
```

Parameters

lFindHandle
[in] 文件查找句柄 ipcFindRecordFile 的返回值

Return Values
小于 0 表示失败，大于等于 0 表成功

Remarks

ipcDownloadByRecordName
录像文件下载

```
LONG ipcDownloadByRecordName (
ipcHandle handle,
char* recordName,
char* sSaveName ,
DWORD dwFileType = 0 ,
LPRECORDDOWNLOADINFO pRDInfo = NULL
)
```

Parameters

handle

[in] 设备句柄
[in] 设备录像的文件名小于字节
[in] 本地保存的文件名绝对路径
[in] 保存的文件格式 0:ipc (暂时仅支持ipc) 1：264裸流
[in] 端点续传结构[保留接口,暂未支持]

Return Values
正确返回下载句柄，错误返回小于

Remarks

ipcStopDownload
停止录像文件下载

```
INT ipcStopDownload (
  LONG    handle,
)
```

Parameters

handle
[in] ipcDownloadByRecordName 的返回值

Return Values
正确返回大于等于，错误返回小于 0
Remarks

ipcDownloadPos
获取录像文件下载进度

```
INT ipcDownloadPos (
LONG    handle
)
```

Parameters

handle
[in] ipcDownloadByRecordName 的返回值

Return Values

0～100 表示下载的进度；100 表示下载结束；正常范围 0-100。错误返回小于 0

Remarks

# 4编程引导

## 4.1视频预览及云台控制

参考：

/sample/vc6/IPCVideoDemo_VC6.0

/sample/vs2005/IPCVideoDemo_VS2005

/sample/vs2008/IPCVideoDemo_VS2008

## 4.2视频码流回调

参考 ：

/sample/vc6/IPCStreamDemo_VC6.0

/sample/vs2005/IPCStreamDemo_VS2005

/sample/vs2008/IPCStreamDemo_VS2008

## 4.3播放本地文件

参考：(例子中使用的 API 文档说明参考 IPC 播放库 SDK 用户手册.pdf)

/sample/vc6/IPCFilePlayDemo_VC6.0

/sample/vs2008/IPCFilePlpayer_VS2008

# 5帮助

## 5.1注意事项

待完善(略)

## 5.2常见问题解答

待完善(略)