Upano

全景相机二次开发 SDK 手册

ClientSDK v1.0

北京极图科技有限公司

目 录

1. SDK 简介.....3
1.1 SDK 模块划分.....3
1.2 目录结构.....3
1.3 库依赖关系.....3
2. SDK 详细文档.....4
2.1 UDeviceCtl 模块.....4
2.1.1 函数.....4
2.1.2 数据类型.....6
2.1.3 错误码.....8
2.2 UPreviewPlayer 模块.....9
2.2.1 函数.....9

1.SDK 简介

本 SDK 用于 WINDOWS 系统的 WIN32 平台，提供 32 位动态库。支持 win7 win8 win10 操作系统。

1.1 SDK 模块划分

本 SDK 分为两个模块，分别是 UDeviceCtl 模块和 UPreviewPlayer 模块。

UDeviceCtl 模块负责对设备进行控制。例如：录制、直播、ISO 设置，快门速度设置等等。

UPreviewPlayer 模块负责预览视频画面，有两种预览模式：全景预览、互动预览。

注：UPreviewPlayer 模块用 DX9 开发完成，所以需要 DX9 运行环境。本 SDK 携带 DX9 环境安装包 directx_Jun2010_redist.exe。 安装方法：双击运行 directx_Jun2010_redist.exe，选择 YES，选择文件解压目录。在解压目录中双击文件 DXSETUP.exe 进行安装。

1.2 目录结构

(1)include 目录 --- 包含头文件。

(2)lib 目录 ---- 包含开发所用*.lib

(3)bin 目录 --- 包含运行所需*.dll

(4)Samples 目录 --- 包含两个 SDK 使用例子。UDeviceCtlSample 目录是 UDeviceCtl 模块的例子。UPreviewSample 是 UPreviewPlayer 模块的例子。这两个例子使用 vs2013 开发，用 vs2013 可以直接打开项目运行实例。

提示：在运行实例前，请先配置好网络。使得电脑可以 Ping 通设备。

注：在运行实例前，需要修改代码中的宏。需要修改代码中的两处地方。

（1）UDeviceCtlSample 中的 sample.cpp 文件，把宏 SAMPLE_DEVICE_IPADDR 改为自己设备的 IP 地址

（2）UPreviewSample 中的 UPreviewSample.cpp 文件，把宏 SAMPLE_STREAM_RTSP 和 SAMPLE_STREAM_RTMP 改为自己需要的流地址

1.3 库依赖关系

(1)UDeviceCtl.lib

头文件：commontypes.h 、UDeviceCtl.h

依赖 dll：UDeviceCtl.dll

(2)UPreviewPlayer.lib

头文件：UPreviewPlayer.h

依赖 dll：UPreviewPlayer.dll、avcodec-57.dll、avformat-57.dll、avutil-55.dll
libgcc_s_dw2-1.dll、libwinpthread-1.dll、swresample-2.dll

2.SDK 详细文档

2.1 UDeviceCtl 模块

函数使用流程介绍：可参考实例 UDeviceCtlSample

第一步：UDeviceRegisterAll() 全局注册

第二步：UCreateDevice() 创建 device,获取返回的 deviceID

第三步：URegisterMessageCallBack() 和 URegisterCallBackSpecific() 注册回调函数。
UDeviceCtl 模块有一个用来接收设备消息的 recv 线程。当 recv 线程解析完设备发过来的消息之后，会调用相应的回调函数来反馈。回调函数的类型是 typedef void(__stdcall *UCALLBACK_PF)(void *pData, int length);
其中 pData 是反馈的数据，length 是反馈数据的长度。消息类型和数据类型的对应关系见 2.1.2 -3 和 2.1.2 - 5

第四步：UConnectDevice() 和设备建立连接。

第五步：调用 USendMessageSet()和 USendMessageGet()和设备进行交互。

第六步：UDisconnectDevice() 断开和设备的连接。

第七步：UDeleteDevice() 删除设备。

2.1.1 函数

1. void UDeviceRegisterAll()
   函数介绍：对 UDeviceCtl 模块进行全局初始化操作，只可调用一次。
   参数说明：无
   返回值：无
2. UDeviceID UCreateDevice (const char *pIpAddr)
   函数介绍：创建设备。
   参数说明：pIpAddr 设备 IP 地址
   返回值：成功 -- 返回创建的设备 ID。
   失败 -- 返回 U_INVALID_DEVICEID
3. void UDeleteDevice(UDeviceID deviceID)
   函数介绍：删除设备
   参数说明：deviceID 设备 ID
   返回值：无

4.URESULT UConnectDevice(UDeviceID deviceID)

函数介绍：连接设备。

参数说明：deviceID 设备 ID。

返回值：成功 -- U_SUCCESS

失败 -- 其它

5.URESULT UDisconnectDevice(UDeviceID deviceID)

函数介绍：和设备断开连接。

参数说明：deviceID 设备 ID。

返回值：成功 -- U_SUCCESS

失败 -- 其它

6. URESULT USendMessageSet (UDeviceID deviceID,
UMESSAGETYPESET_E messageTypeSet,
UMESSAGEDATA_ST *pMessageData);

函数介绍：向设备发送设置类型的消息，对设备进行配置。

参数说明：deviceID -- 设备 ID。

messageTypeSet -- 设置消息类型。详见 2.1.2 - 1

pMessageData -- 消息包含的内容，可以为 NULL；

返回值：成功 -- U_SUCCESS

失败 -- 其它

注：messageTypeSet 和 pMessageData 有一种对应关系。详见 2.1.2 - 2 中的表格。

(1)当 messageTypeSet 对应的类型是整形（int）时，应当填充 pMessageData->data->dataInt;

函数用法示例：对应关系详见 2.1.2 - 2 中的表格

把设备的 ISO 值设置为 10. 由于 ISO 值的类型是整形，所以赋值给 data.dataInt;

```
UMESSAGEDATA_ST messageData;
messageData.data.dataInt = 10;
messageData.length = sizeof(int);
USendMessageSet(deviceID, SENSOR_ISO_SET,  &messageData);
```

(2)当 messageTypeSet 对应的类型是字符串(char*)时，应当填充 pMessageData->data->dataCharArray;

函数用法示例：对应关系详见 2.1.2 - 2 中的表格

设置设备的推流地址。由于推流地址的类型是 char*所以赋值给 data.dataCharArray.

```
char *pStreamAddr = “rtmp://192.168.1.1/live/stream”
UMESSAGEDATA_ST messageData;
strcpy(messageData.data.dataCharArray, pStreamAddr );
messageData.length = strlen(pStreamAddr )+1;
USendMessageSet(deviceID, STREAMLIVE_ADDR_SET,  &messageData);
```

(3)当 messageTypeSet 对应的类型不需要填充数据时，pMessageData=NULL;

函数用法示例：对应关系详见 2.1.2 - 2 中的表格
让设备开始录制全景视频。此消息不需要参数，所以 pMessageData=NULL
USendMessageSet(deviceID, RECORD_PANOSTART_SET, NULL);

7. URESULT USendMessageGet(UDeviceID deviceID,
UMESSAGETYPEGET_E messageTypeGet);
函数介绍：向设备发送获取类型的消息，用来获取设备参数。本函数需要和函数 URegisterMessageCallBack()配合使用。对应关系见 2.1.2 -3
参数说明：deviceID -- 设备 ID。
messageTypeGet -- 获取消息类型
返回值：成功 -- U_SUCCESS
失败 -- 其它

8.URESULT URegisterMessageCallBack(UDeviceID deviceID,
UMESSAGETYPEGET_E messageTypeGet,
UCALLBACK_PF callBackFunc);
函数介绍：此函数用来注册 UMESSAGETYPEGET_E 类型消息的回调函数。
对应关系见 2.1.2 -3

参数说明：deviceID -- 设备 ID。
messageTypeGet -- 获取消息类型
callBackFunc -- 回调函数
返回值：成功 -- U_SUCCESS
失败 -- 其它

9.URESUL URegisterCallBackSpecific(UDeviceID deviceID,
UCALLBACKSPECIFICTYPE_E callBackSpecificType,
UCALLBACK_PF callBackFunc);
函数介绍：此函数用来注册 UCALLBACKSPECIFICTYPE_E 类型消息的回调函数。
对应关系见 2.1.2 - 5
参数说明：deviceID -- 设备 ID。
callBackSpecificType-- 消息类型
callBackFunc -- 回调函数
返回值：成功 -- U_SUCCESS
失败 -- 其它

2.1.2 数据类型

1. UMESSAGETYPESET_E
类型：枚举
说明：对应想要设置的参数。每个参数都有取值范围和对应的数据类型。
详见 2.1.2 - 2 的表格
2.
typedef struct UMessageData

```
{
    union DataType
    {
        int dataInt;
        char dataCharArray[256];
    };
    DataType data;
    int length;
}UMESSAGEDATA_ST;
```

结构介绍：此结构用来保存想要配置的参数。

参数：data -- 如果枚举值 UMESSAGETYPESET_E 对应类型是 int,则赋值给 data.dataInt.如果对应类型是 char* 则赋值给 data.dataCharArray. 对应关系如下

length -- 数据长度

UMESSAGETYPESET_E	对应参数	对应类型	取值范围
SENSOR_ISO_SET	设置 iso	int	[-1,481]
SENSOR_SHUTTERLINE_SET	设置快门速度	int	[-1,33334]
SCHEME_SELECT_SET	选择方案	int	[1,3]
RECORD_PANOSTART_SET	开始全景录制	NULL	
RECORD_PANOSTOP_SET	停止全景录制	NULL	
RECORD_ORGSTART_SET	开始原始录制	NULL	
RECORD_ORGSTOP_SET	停止原始录制	NULL	
STREAMLIVE_SET	开始（停止）推流直播	int	[0,1] 0 -- 停止 1 -- 开始
STREAMLIVE_ADDR_SET	设置推流地址	Char *	
STREAMLIVE_PUSHMODE_SET	设置推流模式（有线、4G）	int	[1,2] 1 -- 有线 2 -- 4G

3. UMESSAGETYPEGET_E

类型：枚举

说明：对应想要获取的信息，可通过此枚举注册回调函数

UMESSAGETYPEGET_E	说明	回调数据类型
SDCARD_STATUS_GET	获取 SD 卡状态，如果 SD 卡状态异常，设备将会一直自动反馈此消息。所以要注意此类型注册的回调函数可能不发送 GET 消息也会被调用。	USDCARDDATA_ST，回调函数会反馈 7 个 sd 卡状态。即反馈数据长度为 7*sizeof(USDCARDDATA_ST)

4. typedef void(__stdcall *UCALLBACK_PF)(void *pData, int length);

类型：函数指针
说明：回调函数。pData 反馈数据，length 反馈数据长度

5. UCALLBACKSPECIFICTYPE_E
类型：枚举
说明：通过此枚举注册回调函数

UCALLBACKSPECIFICTYPE_E	说明	回调数据类型
DEVICE_DISCONNECT_SPECIFIC	设备网络连接异常断开后（例如拔掉网线），调用其注册的回调函数	NULL

6.USDCARDSTATUS_E
类型：枚举
说明：SD 卡状态

USDCARDSTATUS_E	说明
SDCARD_STATUS_NORMAL	SD 卡正常
SDCARD_STATUS_NOCARD	没有插入 SD 卡
SDCARD_STATUS_FULL	SD 卡满
SDCARD_STATUS_ABNORMAL	SD 卡异常

7. typedef struct USDCardData
```
{
    int sdCardID;
    USDCARDSTATUS_E sdCardStatus;
    int sdUnusedMemorySize;
    int sdTotalMemorySize;
}USDCARDDATA_ST;
```
结构介绍：此结构是 SDCARD_STATUS_GET 注册的回调函数的反馈数据类型。
字段介绍：sdCardID -- 取值范围 [0,6] ,
0 -- 表示全景 SD 卡
1-6 -- 表示其它 6 路 SD 卡
sdCardStatus -- sd 卡状态
sdUnusedMemorySize -- sd 卡剩余空间大小， 单位(MB)
sdTotalMemorySize -- sd 卡总容量，单位(MB)

2.1.3 错误码

错误类型	含义

U_SUCCESS	执行成功
U_FAILED	执行失败
UERROR_INVALID_DEVICEID	无效的 deviceID
UERROR_MESSAGE_LENGTH	消息长度太长或错误
UERROR_INVALID_PARAM	无效参数。可能超出有效范围
UERROR_INVALID_MESSAGETYPE	无效的消息类型
U_INVALID_DEVICEID	获取 deviceID 失败

2.2 UPreviewPlayer 模块

函数使用流程介绍：可参考实例 UPreviewSample
第一步：UVideoDecodeShowRegister() 全局注册函数。
第二步：UVideoDecodeShowStart() 开始解码显示图像。
第三步：USetViewMode() 设置预览模式。
当预览模式是互动模式时，USphereSetupMatrics()用来设置角度。
USetFrameCaptureCallBack() 设置用来获取视频帧的回调函数。
USetFrameCaptureMode() 设置获取视频帧的模式。
第四步：UVideoDecodeShowStop() 停止解码显示。

2.2.1 函数

1. void UVideoDecodeShowRegister()
函数介绍：对 UPreviewPlayer 模块进行全局初始化操作，只可调用一次。
参数说明：无
返回值：无

2. int UVideoDecodeShowStart(char* streamAddr, HWND hWnd, int streamType);
函数介绍：开始对设备图像进行解码显示
参数说明：streamAddr -- 流地址，和参数 streamType 相关.
当 streamType 等于 USTREAM_TYPE_RTSP 时,streamAddr 填入设备 IP 地址。 例如：”172.16.0.1”
当 streamType 等于 USTREAM_TYPE_RTMP 时，streamAddr 填入 rtmp 地址 -- 例如：”rtmp://127.0.0.1:1935/myapp/stream”
hWnd -- 要显示图像的窗口句柄
streamType -- 流类型。（USTREAM_TYPE_RTSP 或 USTREAM_TYPE_RTMP)
返回值：成功 -- 返回流 ID
失败 -- 返回 -1

3. void UVideoDecodeShowStop(int streamID);
函数介绍：停止对设备图像进行解码显示。
参数说明：streamID -- 流 ID

返回值：无

4.void USphereSetupMatrics(int streamID, float yaw, float pitch, float s);
函数介绍：当显示模式是互动模式时，此函数用来设置角度和远近。
参数说明：streamID -- 流 ID
yaw -- 设置水平方向视角
pitch -- 设置竖直方向视角
s -- 设置视角远近
返回值：无

5.void USetViewMode(int streamID, int viewMode);
函数介绍：设置预览模式。
参数说明：streamID -- 流 ID
viewMode 预览模式。取值范围[0,1]
viewMode 等于 UVIEWM0DE_PANO， 全景预览模式.
UVIEWMODE_INTERACTION, 互动预览模式
返回值：无

/*****************************类型说明*************************/
typedef struct
{
int width;
int height;
unsigned char *data;
int dataSize;
}UFrame;
类型说明：图像帧类型。
结构说明：width -- 图像的宽。
height -- 图像的高
data -- 指向图像数据的指针。
dataSize -- 图像数据的大小。

typedef void(_stdcall *UFRAMEARRIVED_PF)(UFrame *pFrame, void *pUserData);
类型说明：用来捕获图像帧的回调函数。
参数说明：pFrame -- 捕获到的图像帧。图像格式为 yuv420p
pUserData -- 用户数据，注册回到函数时 USetFrameCaptureCallBack
传入的 pUserData.
/***/

6.void USetFrameCaptureCallBack(int streamID,
UFRAMEARRIVED_PF pFrameArrivedPf,
void *pUserData);
函数介绍：设置用来捕获视频帧的回调函数。捕获到的图像格式为 yuv420p
参数说明：streamID -- 流 ID

pFrameArrivedPf -- 回调函数指针

void *pUserData -- 用户数据，会传入回调函数 pFrameArrivedPf 的 pUserData 参数。

返回值：无

7.void USetFrameCaptureMode(int streamID, UFrameCaptureMode frameCaptureMode);

函数介绍：设置捕获图像帧的模式。捕获的图像帧将通过回调函数传出。图像格式为 yuv420p

参数说明：streamID -- 流 ID

frameCaptureMode -- 捕获图像帧的模式：

DoNothing -- 不捕获图像帧。

OnlyOnce -- 只捕获一帧。

KeepWorking -- 持续捕获图像帧。

返回值：无