SGP_SDK 開發手冊(C++版)

概述

歡迎使用 SGP_SDK 開發手冊，本文檔詳細描述了開發包中各個函數實現功能、接口及其函數之間的調用關係和示例實現。
本開發包主要包含業務操作和設備管理兩大部分：
業務操作：實時預覽、拍照、溫度獲取等功能。
設備管理：設備參數配置(系統通用參數配置、圖像配置、視頻配置、網絡配置、電機配置)等功能。

Windows 下，SDK 包括的文件有

功能库	SgpApi.h	SDK 对外头文件
	SgpParam.h	结构体定义头文件
	SgpApi.lib	lib 功能库
	SgpApi.dll	功能库
依赖库	avcodec-58.dll	
	avdevice-58.dll	
	avfilter-7.dll	
	avformat-58.dll	
	avutil-56.dll	
	cairo.dll	
	GFileAnalysis.dll	
	GuideSDK.dll	
	PocoFoundation.dll	
	PocoJSON.dll	
	postproc-55.dll	
	swresample-3.dll	
	swscale-5.dll	
配置文件	config.cfg	

Linux 下，SDK 包括的文件有		
功能库	SgpApi.h	SDK 对外头文件
	SgpParam.h	结构体定义头文件
	libSgpApi.so	功能库
依赖库	libavcodec.so	
	libacdevice.so	
	libavfilter.so	
	libavformat.so	
	libavfilter.so	
	libavformat.so	
	libavutil.so	
	libGuideSDK.so	
	ibMeasureStream.so	
	libPocoFoundation.so	
	libPocoJSON.so	
	libpostproc.so	
	libswresample.so	
	libswscale.so	
	libva-drm.so	
	libva-xzz.so	
	libva.so	
	libx264.so	
配置文件	config.cfg	

本 SDK 的功能库和依赖库都是必须，缺少依赖库会导致某些功能运行异常。

log_out_file 为 SDK 日志文件夹。

版權所有©翌宙科技股份有限公司

SDK 支持系统
Windows 32/64 位网络 SDK： Windows 10/Windows 8/Windows 7 以及 Windows Server 2012/2008 x86 Linux 32/64 位设备网络 SDK： 已测系统：CentOS 7、Redhat、Ubuntu 12、Ubuntu 14、Ubuntu 16、Ubuntu 18、Ubuntu 20 Arm Linux 32/64 位设备网络 SDK： 需提供交叉编译环境，定制化编译

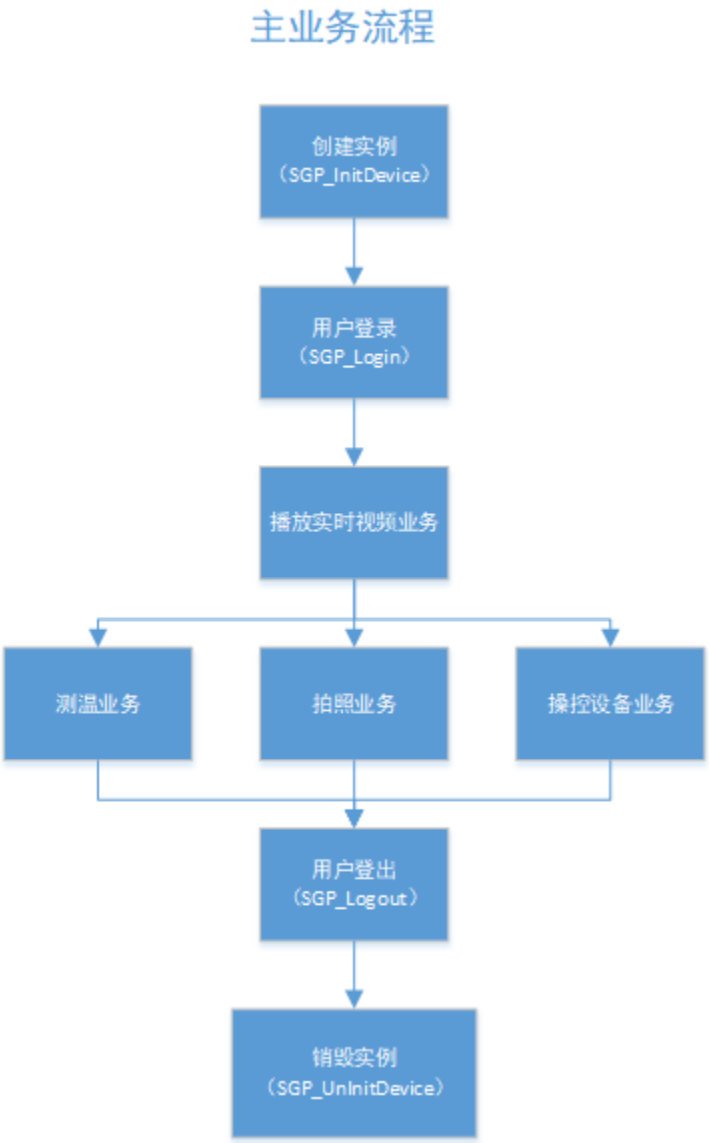
开发说明	
Window 开发环境	SDK: 根据平台选择 32 位或者 64 位 SDK，SDK 包可单放在某个文件夹下或者将 SDK 包内容拷贝到应用程序 exe 同级别目录下。
Linux 开发环境	SDK: 根据平台选择 32 位或者 64 位 SDK，解压命令 tar -xzf lib.tar.gz。开使用的库在 SDK\lib\目录下。SDK 包可单放在某个文件夹下或者将 SDK 包内容拷贝到应用程序同级别目录下。

版權所有©翌宙科技股份有限公司

修订记录

发布时间	版本号	修订内容
2022.10	V1.0.0	新增

版權所有©翌宙科技股份有限公司



流程说明：

创建实例会返回实例序号值，之后的所有函数都需要用到这个序号值。

操作任何业务之前首先要登陆平台

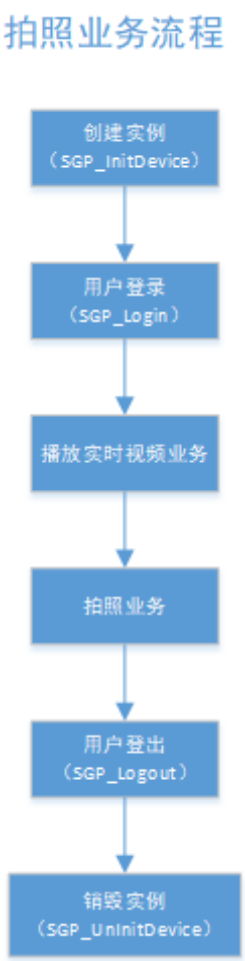
创建实例（SGP_InitDevice）：生成一个 SGPSDK 的独立实例，之后所有函数操作都作用于这个实例。可多次调用此函数生成多个实例，多个实例之间的操作互不影响。

登陆平台（SGP_Login）： 实现用户登录服务器功能。平台中设置用户为复用时，在可以同时间多次登陆。

登出平台（SGP_Logout）:登出平台

销毁实例（SGP_UnInitDevice）：

版權所有©翌宙科技股份有限公司



流程说明：

创建实例会返回实例序号值，之后的所有函数都需要用到这个序号值。

操作任何业务之前首先要登陆到平台

创建实例（SGP_InitDevice）：生成一个 SGPSDK 的独立实例，之后所有函数操作都作用于这个实例。可多次调用此函数生成多个实例，多个实例之间的操作互不影响。

登陆平台（SGP_Login）： 实现用户登录服务器功能。平台中设置用户为复用时，在可以同时间多次登陆。

拍照业务：调用 SGP_GetHeatMap 函数获得 jpeg 国网格式热图，调用 SGP_GetFirHeatMap 获得 FIR 格式国网格式热图。

登出平台（SGP_Logout）：登出平台

销毁实例（SGP_UnInitDevice）：

版權所有©翌宙科技股份有限公司

连接设备接口

接口描述	功能描述	功能详细描述
SGP_InitDevice	初始化一个设备对象	
SGP_UnInitDevice	释放设备对象	
SGP_Login	用户登录	
SGP_Logout	用户登出	
SGP_GetGeneralInfo	获取通用信息	

初始化一个设备对象 SGP_InitDevice

选项：	说明
描述：	初始化一个设备对象
详细描述：	
函数：	SGP_HANDLE SGP_InitDevice();
参数：	[in] 无
返回值：	成功 返回设备对象句柄
备注：	首次调用的函数，与 SGP_UnInitDevice 成对使用。同一台设备最多支持 20 路访问

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象句柄。
	**/
	void Init()
	{
	SGP_HANDLE handle = 0;
	handle = SGP_InitDevice();
	if (handle)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

版權所有©翌宙科技股份有限公司

释放设备对象 SGP_UnInitDevice

选项：	说明
描述：	释放设备对象
详细描述：	
函数：	void SGP_UnInitDevice(SGP_HANDLE handle);
参数：	handle [in] 设备对象句柄
返回值：	无
备注：	与 SGP_InitDevice 成对使用

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象句柄。
	**/
	void Uninit()
	{
	SGP_UnInitDevice(handle);
	}

版權所有©翌宙科技股份有限公司

用户登录 SGP_Login

选项：	说明
描述：	用户登录
详细描述：	需要登录成功以后才能访问其他接口
函数：	int SGP_Login(SGP_HANDLE handle, const char *server, const char *username, const char *password, int port);
参数：	handle [in] 设备对象句柄 server [in] 设备 ip 地址，设备 ip 初始默认为“192.168.1.168” username [in] 用户名，管理员账号默认为“admin” password [in] 密码，admin 账号密码默认为“admin123”，明文方式 port [int] 端口，端口默认为 80
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	需要登录成功以后才能访问其他接口，与 SGP_Logout 成对使用； username、password 传空字符串，port 传 0 即可
使用示例：	/** 示例中部分类、变量、函数的解释： 1，handle 设备对象句柄。

```
 **/
 void Init()
 {
     SGP_HANDLE handle = 0;
     handle = SGP_InitDevice();
     if (handle)
     {
         const char *server = "192.168.1.168";
         const char *username = "admin";
         const char *password = "admin123";
         int port = 80;
         int ret = SGP_Login(handle, server,
username, password, port);
         if (ret == SGP_OK)
         {
              //登录成功，TODO......
         }
         else
         {
              SGP_UnInitDevice(handle)
              //登录失败，TODO......
         }
      }
      else
      {
          //失败，TODO......
      }
 }
```

版權所有©翌宙科技股份有限公司

用户登出 SGP_Logout

选项：	说明
描述：	用户登出
详细描述：	
函数：	int SGP_Logout(SGP_HANDLE handle);
参数：	handle [in] 设备对象句柄
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	与 SGP_Login 成对使用

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象句柄。
	**/
	void Init()
	{
	int ret = SGP_Logout(handle);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

版權所有©翌宙科技股份有限公司

获取通用信息 SGP_GetGeneralInfo

选项:	说明
描述：	获取通用信息
详细描述：	
函数：	int SGP_GetGeneralInfo(SGP_HANDLE handle, SGP_GENERAL_INFO *output);
参数：	handle [in] 设备对象句柄 output [out] 输出信息，获取的通用信息
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在使用前先初始化 仅 ir_output_w、ir_output_h 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象句柄。
	**/
	void Init()
	{
	SGP_GENERAL_INFO info;
	memset(&info, 0x00, sizeof(SGP_GENERAL_INFO));
	int ret = SGP_GetGeneralInfo(handle,&info);
	if (ret ==SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

测温接口

接口描述	功能描述	功能详细描述
SGP_GetAnalyticObjectsTemp	获取分析对象温度	
SGP_GetImageTemps	获取温度矩阵	

版權所有@翌宙科技股份有限公司

获取分析对象温度 SGP_GetAnalyticObjectsTemp

选项：	说明
描述：	获取分析对象温度
详细描述：	
函数：	int SGP_GetAnalyticObjectsTemp(SGP_HANDLE handle, SGP_ANALYTIC_TEMPS *output);
参数：	handle [in] 传入设备对象 output [out] 输出信息
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在使用前先初始化 仅工业机芯有效
使用示例：	/** 示例中部分类、变量、函数的解释： 1，handle 设备对象。 **/ void Init() { SGP_ANALYTIC_TEMPS array;

	memset(&array, 0x00, sizeof(SGP_ANALYTIC_TEMPS));
	int ret = SGP_GetAnalyticObjectsTemp(handle,&array);
	if (ret == SGP_OK)
	{
	//成功，TODO......
	}
	else
	{
	//失败，TODO......
	}
	}

版權所有©翌宙科技股份有限公司

获取温度矩阵 SGP_GetImageTemps

选项：	说明
描述：	获取温度矩阵
详细描述：	
函数：	int SGP_GetImageTemps(SGP_HANDLE handle, float *output, int length, int type);
参数：	handle [in] 传入设备对象 output [out] 输出温度矩阵 length [in] output 大小 type [in] 0 为推流红外分辨率，1 为设备红外原始分辨率
返回值：	成功返回 SGP_OK，失败返回错误码

备注：	温度矩阵值为 float 类型，4 个字节长度，调用 SGP_GetGeneralInfo 函数获取红外模组宽和红外模组高，第三个参数 length 值传入红外模组宽*红外模组高*4

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_GENERAL_INFO info;
	memset(&info, 0x00, sizeof(info));
	int ret = SGP_GetGeneralInfo(handle,&info);
	if (ret == SGP_OK)
	{
	int heigth = info.ir_model_h;
	int width = info.ir_model_w;
	int length = heigth*width;
	int type = 1;
	float *output = (float *)calloc(length, sizeof(float));
	if(output!=NULL)
	{
	ret = SGP_GetImageTemps(handle,output,length*4,type);
	if (ret == SGP_OK)
	{
	//成功，TODO......
	}
	else
	{
	//失败，TODO......
	}
	}
	free(output);
	output=NULL;
	}
	}

版權所有©翌宙科技股份有限公司

拍照接口

接口描述	功能描述	功能详细描述
SGP GetHeatMap	获取热图	

版權所有@翌宙科技股份有限公司

获取热图 SGP_GetHeatMap

选项：	说明
描述：	获取热图
详细描述：	热图文件格式为国网格式
函数：	int SGP_GetHeatMap(SGP_HANDLE handle, const char *input);
参数：	handle [in] 传入设备对象 input [in] 保存文件路径+文件名+.jpg
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	工业机芯拍国网格式、非工业机芯拍高德格式

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	const char path[] = "./screenpic.jpg";
	int ret = SGP_GetHeatMap (handle,path);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

表 1

中文名称	英文名称	数据类型	长度字节	注　释
红外视频截图数据文件	IRImage	二进制数据		JPEG 格式的视频截图中至少需要包含柱状温度色标、公司 logo、辐射率、拍摄距离、环境温度、拍摄时间（至少精确到分钟）、现场拍摄时增加的分析区域（如点、框等分析区域）及点对应的温度和框对应的最高温度数值和位置。公司 logo、辐射率、拍摄距离、环境温度、拍摄时间等信息可放在视频截图下方
文件版本	File Version	Unsigned short	2	本标准发布时使用 1.0 版本，即 0x0100
温度点阵宽度	Width	Unsigned short	2	例如：640
温度点阵高度	Height	Unsigned short	2	例如：480
拍摄时间	DateTime	Unsigned char	14	14 字节的时间字符串，格式是 YYYYMMDDHHMMSS，例如“20150812124818”，存储每个字符对应的 ASCII 码
红外温度值点阵数据	IRData	Float		红外温度值点阵数据以 Float 类型（4 字节）的浮点数直接存储每个像素点的温度，按照从左到右、从上到下的顺序依次存储
辐射率	Emiss	Float	4	范围 0～1，例如：0.9；该字段必须支持
环境温度	Ambient Temperature	Float	4	单位：℃。例如：25.3℃；该字段必须支持
镜头度数	Len	Unsigned char	1	例如：24，代表 24°镜头； 0 代表不支持该参数
拍摄距离	Distance	Unsigned int	4	单位：m；0 代表不支持该参数
相对湿度	Relative Humidity	Unsigned char	1	存储百分比，范围 0～100，50 代表 50%；0 代表不支持该参数
反射温度	Reflective Temperature	Float	4	单位：℃。例如：25.3℃； 0 代表不支持该参数
生产厂家	Productor	Unsigned char	32	存储每个字符对应的 ASCII 码，例如：0x4D495353494F4E 代表“MISSION”不足位补 0；全部填充 ASCII 码，0 代表不支持该参数
产品型号	Type	Unsigned char	32	存储每个字符对应的 ASCII 码，全部填充 ASCII 码，0 代表不支持该参数
产品序列号	Serial NO	Unsigned char	32	存储每个字符对应的 ASCII 码，全部填充 ASCII 码，0 代表不支持该参数

17

表 C.1（续）

中文名称	英文名称	数据类型	长度字节	注　释
经度	Longitude	Double	8	例如：120.11896012； 0 代表不支持该参数
纬度	Latitude	Double	8	例如：30.1581147021； 0 代表不支持该参数
海拔	Altitude	Int	4	单位：m；例如：20； 0 代表不支持该参数
备注信息长度	Description Length	Unsigned int	4	0 代表没有存储备注信息
备注信息	Description Data	Unsigned char	由“备注信息长度”字段指定	备注信息的具体内容； 例如分析结果、诊断结果等
红外数据的起始偏移地址	IRData Offset	Unsigned int	4	记录“文件版本”字段在整个文件中的偏移地址，用于定位红外数据的起始地址
文件末尾标识	File End Type	Unsigned char	16	必须是 0x37、0x66、0x07、0x1a、0x12、0x3a、0x4e、0x9f、0xa9、0x5d、0x21、0xd2、0xda、0x7d、0x26、0xbc

版權所有©翌宙科技股份有限公司

成像接口

接口描述	功能描述	功能详细描述
SGP_OpenIrVideo	开启红外视频	
SGP_CloseIrVideo	关闭红外视频	

版權所有@翌宙科技股份有限公司

开启红外视频 SGP_OpenIrVideo

选项：	说明
描述：	开启红外视频
详细描述：	
函数：	int SGP_OpenIrVideo(SGP_HANDLE handle, SGP_RTSPCALLBACK callback, void *pUser);
参数：	handle [in] 传入设备对象 callback [in] 注册图像回调函数（RGB24 数据） pUser [in] void* 可传入窗口句柄指针
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	
使用示例：	/** 示例中部分类、变量、函数的解释： 1，handle 设备对象。 **/ /**

	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	2,以 QT 界面库为例
	**/
	static void GetIrRtsp(unsigned char *outdata, int w, int h, void *pUser)
	{
	MainWindow *pDlg = (MainWindow *)pUser;
	//TODO......
	}
	void MainWindow::Init()
	{
	SGP_OpenIrVideo(handle, GetIrRtsp, this);
	//TODO......
	}

回调函数描述		
函数名称	typedef void(*SGP_RTSPCALLBACK)(unsigned char *outdata, int w, int h, void *pUser);	
功能描述	图像数据回调函数	
参数说明	outdata	输出参数 图像数据
	w	输出参数 图像宽度
	h	输出参数 图像高度
	pUser	输出参数
返回值	无	

版權所有©翌宙科技股份有限公司

关闭红外视频 SGP_CloseIrVideo

选项：	说明
描述：	关闭红外视频
详细描述：	
函数：	void SGP_CloseIrVideo(SGP_HANDLE handle);
参数：	handle [in] 传入设备对象
返回值：	无
备注：	退出登录会自动关闭视频流

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_CloseIrVideo(handle);
	//TODO......
	}

版權所有©翌宙科技股份有限公司

操控设备接口

接口描述	功能描述	功能详细描述
SGP_DoShutter	快门操作	
SGP_GetThermometryParam	获取全局测温参数	
SGP_SetThermometryParam	设置全局测温参数	
SGP_SetColorBar	设置色带号	
SGP_SetFocus	调焦	
SGP_GetMotorPosition	获取电机位置	
SGP_SetRange	切换测温范围	
SGP_GetThermometryRule	获取分析对象	
SGP_AddThermometryRule	添加分析对象	
SGP_UpdateThermometryRule	更新分析对象	
SGP_DeleteThermometryRule	删除分析对象	
SGP_DeleteAllThermometryRule	删除全部分析对象	
SGP_SetThermometryRuleShowMode	设置分析对象温度显示类型	
SGP_GetIrImageEffectParam	获取红外图像效果参数	

SGP_SetIrImageEffectParam	设置红外图像效果参数	
SGP_GetNetInfo	获取网络信息	
SGP_SetVideoParam	设置视频参数	
SGP_GetVersionInfo	获取系统版本信息	
SGP_SetElectronicMagnification	设置电子变倍	

版權所有@翌宙科技股份有限公司

快门操作 SGP_DoShutter

选项：	说明
描述：	快门操作
详细描述：	
函数：	int SGP_DoShutter(SGP_HANDLE handle, SGP_SHUTTER_ENUM type);
参数：	handle [in] 传入设备对象 type [in] 快门操作类型
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_SHUTTER_ENUM type = SGP_SHUTTER;
	int ret = SGP_DoShutter(handle,type);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

获取全局测温参数 SGP_GetThermometryParam

选项：	说明
描述：	获取全局测温参数
详细描述：	
函数：	int SGP_GetThermometryParam(SGP_HANDLE handle, SGP_THERMOMETRY_PARAM *output);
参数：	handle [in] 传入设备对象 output [out] 输出信息
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在使用前先初始化 仅 dist、emiss、humi 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_THERMOMETRY_PARAM info;
	memset(&info, 0x00, sizeof(SGP_THERMOMETRY_PARAM));
	int ret = SGP_GetThermometryParam(handle,&info);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

设置全局测温参数 SGP_SetThermometryParam

选项：	说明
描述：	设置全局测温参数
详细描述：	
函数：	int SGP_SetThermometryParam(SGP_HANDLE handle, SGP_THERMOMETRY_PARAM input);
参数：	handle [in] 传入设备对象 input [in] 输入信息
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	使用前先调用 SGP_GetThermometryParam 获取全局测温参数，然后再修改测温参数， 结构体变量在使用前先初始化 仅 dist、emiss、humi 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	//先获取全局测温参数，再设置参数。
	SGP_THERMOMETRY_PARAM info;
	memset(&info, 0x00, sizeof(SGP_THERMOMETRY_PARAM));
	int ret = SGP_GetThermometryParam(handle,&info);
	if (ret == SGP_OK)
	{
	parm.dist = 5; //修改测温距离为 5 米
	ret = SGP_SetThermometryParam(handle,info);

```
            if (ret == SGP_OK )
            {
                //成功，TODO......
            }
            else
            {
                //失败，TODO......
            }

        }
}
```

设置色带号 SGP_SetColorBar

选项：	说明
描述：	设置色带号
详细描述：	
函数：	int SGP_SetColorBar(SGP_HANDLE handle, int input);
参数：	handle [in] 传入设备对象 input [in] 色带号
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	色带号范围 1-26

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	int colorbar = 2;
	int ret = SGP_SetColorBar(handle,colorbar);
	if (ret == SGP_OK)
	{
	//success, TODO......
	}
	else
	{
	//fail, TODO ......
	}
	}

调焦 SGP_SetFocus

选项：	说明
描述：	调焦
函数：	int SGP_SetFocus(SGP_HANDLE handle, SGP_FOCUS_TYPE type, int value);
参数：	handle [in] 传入设备对象 type [in] 操作类型 value [in] 电机位置值 0~750，当 type 传入 SGP_FOCUS_PLACE 有效
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	电机位置值范围 0~750

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_FOCUS_TYPE type = SGP_FOCUS_AUTO;
	int value =0;
	int ret = SGP_SetFocus(handle,type,value);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

获取电机位置 SGP_GetMotorPosition

选项：	说明
描述：	获取电机位置
详细描述：	
函数：	int SGP_GetMotorPosition(SGP_HANDLE handle, int *output);
参数：	handle [in] 传入设备对象 output [out] 电机位置
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	int value =0;
	int ret =
	SGP_GetMotorPosition(handle,&value);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

切换测温范围 SGP_SetRange

选项：	说明
描述：	切换测温范围
函数：	int SGP_SetRange(SGP_HANDLE handle, int input);
参数：	handle [in] 传入设备对象 input [in] 0~2，部分设备只有 1 个档位，目前最多有 3 个档位
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	例如对 IPT640M，0 表示-20℃~150℃，1 表示 100℃~350℃，2 表示 100℃~550℃（如果设备支持则包含 2）

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_GENERAL_INFO info;
	memset(&info, 0x00, sizeof(SGP_GENERAL_INFO));
	int ret = SGP_GetGeneralInfo(handle,&info);
	if (ret ==SGP_OK)
	{
	int range = info.range_num;
	ret = SGP_SetRange(handle,range -1);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

获取分析对象 SGP_GetThermometryRule

选项：	说明
描述：	获取分析对象
函数：	int SGP_GetThermometryRule(SGP_HANDLE handle, SGP_RULE_ARRAY *output);
参数：	handle [in] 传入设备对象 output [out] 全部分析对象
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在使用前先初始化 仅工业机芯有效；仅 type、points 有效，且 type 中 4 无效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_RULE_ARRAY array;
	memset(&array, 0x00, sizeof(SGP_RULE_ARRAY));
	int ret = SGP_GetThermometryRule(handle,&array);
	if (ret == SGP_OK)
	{
	//成功，TODO......
	}
	else
	{
	//失败，TODO......
	}
	}

添加分析对象 SGP_AddThermometryRule

选项：	说明
描述：	添加分析对象
函数：	int SGP_AddThermometryRule(SGP_HANDLE handle, SGP_RULE input);
参数：	handle [in] 传入设备对象 input [in] 分析对象类型
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在使用前先初始化 仅工业机芯有效；仅 type、points 有效，且 type 中 4 无效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_RULE rulePoint;
	memset(&rule, 0x00, sizeof(SGP_RULE));
	rulePoint.points_num = 1; //点个数是 1
	rulePoint.points[0].x = 200;
	rulePoint.points[0].y = 200;
	rulePoint.type = 1; //类型是 1
	
	int ret = SGP_AddThermometryRule(handle,rulePoint);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....

```
        }
    }
```

更新分析对象 SGP_UpdateThermometryRule

选项：	说明
描述：	更新分析对象
详细描述：	
函数：	int SGP_UpdateThermometryRule(SGP_HANDLE handle, SGP_RULE input);
参数：	handle [in] 传入设备对象 input [in] 分析对象
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	先调用 SGP_GetThermometryRule 函数获取，再更新，结构体变量在使用前先初始化 仅工业机芯有效；仅 type、points 有效，且 type 中 4 无效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_RULE_ARRAY array;
	memset(&array, 0x00, sizeof(SGP_RULE_ARRAY));
	int ret = SGP_GetThermometryRule(handle,&array);
	if (ret == SGP_OK)
	{
	if(array.rule_num>0)
	{
	SGP_RULE rule;
	memset(&rule, 0, sizeof(SGP_RULE));
	memcpy(&rule,&array.rule[0],sizeo

```
f(SGP_RULE));//取第一组分析对象值
              SGP_POINT point1;
              point1.x = 10;
              point1.y = 10;
              rule.points[0] = point1;
              SGP_POINT point2;
              point2.x = 100;
              point2.y = 10;
              rule.points[1] = point2;
              SGP_POINT point3;
              point3.x = 100;
              point2.y = 100;
              rule.points[2] = point3;
              SGP_POINT point4;
              point4.x = 10;
              point4.y = 100;
              rule.points[3] = point4;
             int ret =
SGP_UpdateThermometryRule(handle,rule);
              if (ret == SGP_OK )
              {
                  //成功，TODO......
              }
              else
              {
                  //失败，TODO......
              }
          }
          //成功，TODO......
     }
     else
     {
          //失败，TODO......
     }
 }
```

删除分析对象 SGP_DeleteThermometryRule

选项：	说明
描述：	删除分析对象
详细描述：	
函数：	int SGP_DeleteThermometryRule(SGP_HANDLE handle, int input);
参数：	handle [in] 传入设备对象 input [in] 分析对象 id
返回值：	成功返回 SGP_OK,失败返回错误码
备注：	仅工业机芯有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_RULE_ARRAY array;
	memset(&array, 0x00, sizeof(SGP_RULE_ARRAY));
	int ret = SGP_GetThermometryRule(handle,&array);
	if (ret == SGP_OK)
	{
	if(array.rule_num>0)
	{
	int ret = SGP_DeleteThermometryRule(handle,array.rule[0].id);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}

```
                else
                {
                    //失败，TODO......
                }
            }
            //成功，TODO......
        }
        else
        {
            //失败，TODO......
        }
    }
```

删除全部分析对象 SGP_DeleteAllThermometryRule

选项：	说明
描述：	删除全部分析对象
详细描述：	
函数：	int SGP_DeleteAllThermometryRule(SGP_HANDLE handle);
参数：	handle [in] 传入设备对象
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	仅工业机芯有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	int ret =
	SGP_DeleteAllThermometryRule(handle);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

设置分析对象温度显示类型

SGP_SetThermometryRuleShowMode

选项：	说明
描述：	设置分析对象温度显示类型
函数：	int SGP_SetThermometryRuleShowMode(SGP_HANDLE handle, int input);
参数：	handle [in] 传入设备对象 input [in] 对象温度显示:1 最高温;2 最低温;3 平均温;4 仅名称;5 不显示
返回值：	成功返回 SGP_OK,失败返回错误码
备注：	需要分析对象开启配置，设置分析对象温度显示才会生效。 仅工业机芯有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	int showtype = 1;
	int ret = SGP_SetThermometryRuleShowMode(handle,showtype);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

获取红外图像效果参数 SGP_GetIrImageEffectParam

选项：	说明
描述：	获取红外图像效果参数
详细描述：	
函数：	int SGP_GetIrImageEffectParam(SGP_HANDLE handle, SGP_IAMGE_EFFECT_PARAM_IR_CONFIG *output);
参数：	handle [in] 传入设备对象 output [out] 输出信息
返回值：	成功返回 SGP_OK,失败返回错误码
备注：	结构体变量在使用前先初始化 仅 SGP_IR_BRIGHTNESS、SGP_IR_CONTRAST 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_IAMGE_EFFECT_PARAM_IR_CONFIG info;
	memset(&info, 0x00, sizeof(info));
	int ret =
	SGP_GetIrImageEffectParam(handle,&info);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

设置红外图像效果参数 SGP_SetIrImageEffectParam

选项：	说明
描述：	设置红外图像效果参数
函数：	int SGP_SetIrImageEffectParam(SGP_HANDLE handle, SGP_IR_IMAGE_EFFECT_ENUM type, int value);
参数：	handle [in] 传入设备对象 type [in] 参数类型 value [in] 参数值
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	仅 SGP_IR_BRIGHTNESS、SGP_IR_CONTRAST 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_IR_IMAGE_EFFECT_ENUM type = SGP_IR_BRIGHTNESS;
	int value = 50;
	int ret = SGP_SetIrImageEffectParam(handle,type,value);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

获取网络信息 SGP_GetNetInfo

选项：	说明
描述：	获取网络信息
函数：	int SGP_GetNetInfo(SGP_HANDLE handle, SGP_NET_INFO *output);
参数：	handle [in] 传入设备对象 output [out] 输出信息
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在定义后先初始化 仅 dns1、gateway、ipaddr、netmask、mac 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_NET_INFO info;
	memset(&info, 0x00, sizeof(SGP_NET_INFO));
	int ret = SGP_GetNetInfo(handle,&info);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

设置视频参数 SGP_SetVideoParam

选项：	说明
描述：	设置视频参数
详细描述：	
函数：	int SGP_SetVideoParam(SGP_HANDLE handle, SGP_VIDEO_PARAM_ENUM type, SGP_VIDEO_PARAM input);
参数：	handle [in] 传入设备对象 type [in] 视频类别 input [in] 参数值
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	仅 fps 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_GENERAL_INFO info;
	memset(&info, 0x00, sizeof(SGP_GENERAL_INFO));
	int ret = SGP_GetGeneralInfo(handle,&info);
	
	if (ret == SGP_OK)
	{
	SGP_VIDEO_PARAM_ENUM type = SGP_IR;
	SGP_VIDEO_PARAM param;
	memset(&param, 0x00, sizeof(SGP_VIDEO_PARAM));
	ret = SGP_GetVideoParam(handle,type,&param);

```
          param.fps = 25;
          ret =
SGP_SetVideoParam(handle,type,param);
          if (ret == SGP_OK )
          {
               //成功，TODO......
          }
          else
          {
               //失败，TODO......
          }
     }
     else
     {
          //失败，TODO......
     }
 }
```

获取系统版本信息 SGP_GetVersionInfo

选项：	说明
描述：	获取系统版本信息
详细描述：	
函数：	int SGP_GetVersionInfo(SGP_HANDLE handle, SGP_VERSION_INFO *output);
参数：	handle [in] 传入设备对象 output [out] 版本信息
返回值：	成功返回 SGP_OK，失败返回错误码
备注：	结构体变量在使用前先初始化 仅 serial、fpga_version、sdk_version 有效

使用示例：	/**
	示例中部分类、变量、函数的解释：
	1，handle 设备对象。
	**/
	void Init()
	{
	SGP_VERSION_INFO info;
	memset(&info, 0x00, sizeof(SGP_VERSION_INFO));
	int ret = GetVersionInfo(handle,&info);
	if (ret == SGP_OK)
	{
	//成功，TODO.....
	}
	else
	{
	//失败，TODO.....
	}
	}

设置电子变倍 SGP_SetElectronicMagnification

选项：	说明
描述：	设置电子变倍，只对主码流有效
详细描述：	
函数：	int SGP_SetElectronicMagnification(SGP_HANDLE handle, SGP_VIDEO_PARAM_ENUM type, int magnification);
参数：	handle [in] 传入设备对象 type [in] 视频类型值 input [in] 1：红外原始，可见光原始 2：红外 2 倍，可见光 4 倍 3：红外 3 倍，可见光 16 倍
返回值：	成功返回 SGP_OK,失败返回错误码
备注：	
使用示例：	/** 示例中部分类、变量、函数的解释： 1，handle 设备对象。 **/ void Init() {

```
     int magnification = 2;
     SGP_VIDEO_PARAM_ENUM type = 3;
     int ret =
SGP_SetElectronicMagnification(handle,type,magnificatio
n);
     if (ret == SGP_OK )
     {
          //成功，TODO......
     }
     else
     {
          //失败，TODO......
     }
 }
```

结构体定义描述

```
struct SGP_ACCESS_VIOLATION_INFO
{
    int audio_flag;//是否音频联动 0:否; 1:是
    int audio_index;//音频文件索引 0-2
    int audio_mode;//音频模式 0:持续时间; 1:播放次数
    int audio_value;//音频模式值 0-100(次/秒)
    int allow_count;//允许登录次数 3-10 次
    int flag;//是否开启 0:不开启; 1:开启
    int sendmail;//是否发送邮件 0:否; 1:是
    int light_flag;//是否闪光灯 0:否; 1:是
    int light_hold;//闪光灯持续时间 10-300s
    int output_flag;//是否外部输出 0:不输出 1:输出
    int output_hold;//外部输出持续时间 10-300s
};
```

结构体定义描述

```
struct SGP_ACCESSVIOLATIONNOTIFY
{
    char user[STRING LENGH];//异常登录用户
    char ip[STRING LENGH];  //异常登录 IP
    char time[STRING LENGH];//异常登录时间
};
```

结构体定义描述

```
struct SGP_ALARM_INPUT_INFO
{
    int flag;//是否开启 0 不开启 1 开启
    int alarm_shake;//报警抖动 0-100s
    int type;//输入类型：0 常开型  1 常闭型
    int record_delay;//录制延时 10-300
    int record_flag;//是否录制 0:不录制; 1:录制
    int record_stream;//录制类型 0:不录制; 1:只录制可见光; 2:只录制红外; 3:录制红外和可见光
    int capture_flag;//是否截图 0:否; 1:是
    int capture_stream;//截图类型 0:不截图; 1:只截图可见光; 2:只截图红外; 3:截图红外和可见光
    int sendmail;//是否发送邮件 0:不发送; 1:发送
    int light_flag;//是否开启闪光灯 0:否; 1:是
    int light_hold;//闪光灯持续时间，10-300s
    int output_flag;//是否外部输出 0:不输出 1:输出
    int output_hold;//外部输出持续时间 10-300s
    int audio_flag;//是否音乐提醒 1:是; 0:否
    int audio_index;//音乐文件索引，0-2
    int audio_mode;//音乐播放模式 1:播放次数; 2:持续时间
    int audio_value;//音乐播放值,随模式定义:(持续时间:秒数)(播放次数:播放次数 0-100)
    int effect_day_num;//时间数组数量
    SGP_EFFECT_DAY effect_day[7];//时间数组
};
```

结构体定义描述

```
struct SGP_ALARMINPUTNOTIFY
{
    char time[STRING LENGH];//报警时间，格式为 2020-05-21 12:22:33
    char vl_image_url[STRING LENGH];//报警记录可见光截图，http jpeg 路径
    char ir_image_url[STRING LENGH];//报警记录红外截图，http jpeg 路径
    char vl_image_content[STRING LENGH];//可见光 JPEG 图片得 BASE64 格式
    char ir_image_content[STRING LENGH];//红外 JPEG 图片得 BASE64 格式
    char vl_video_url[STRING LENGH];//可见光录像地址
    char ir_video_url[STRING LENGH];//红外录像地址
};
```

结构体定义描述

```
struct SGP_ANALYTIC_TEMP
{
    int rule_id;     //规则 id
    char rule_name[STRING LENGH];  //规则名称 32 字符以内
    int type;//对象类型 1 点; 2 线; 3 矩形; 4 多边形; 5 圆形
    float max_temp;//最高温度值
    float min_temp;//最低温度值
    float avg_temp;//平均温度值
};
```

结构体定义描述

```
struct SGP_ANALYTIC_TEMPS
{
    int analytic_num;
    SGP_ANALYTIC_TEMP analytic[ANALYTIC_MAX_NUM];
};
```

结构体定义描述

```
struct SGP_COLD_HOT_TRACE_INFO
{
    int light_hold;//闪光灯持续时间 10-300s
    int light_flag;//是否开启闪光灯 0:否; 1:是
    int alarm_shake;//报警抖动,单位 s,0-100
    int capture_flag;//是否截图 0:否; 1:是
    int capture_stream;//截图类型 1:只截图可见光; 2:只截图红外;
3:截图红外和可见光
    char high_color[STRING_LENGH];//高温颜色:0xRGB
    int high_flag;//高温是否检测 0:不检测; 1:检测
    float high_temp;//高温温度，-40~2000
    char low_color[STRING_LENGH];//低温颜色:0xRGB
    int low_flag;//低温是否检测 0:不检测; 1:检测
    float low_temp;//低温温度，-40~2000
    int record_delay;//录制延时 10-300s
    int record_flag;//是否录制 0:不录制; 1:录制
    int record_stream;//录制类型 1:只录制可见光; 2:只录制红外;
3:录制红外和可见光
    int sendmail;//是否发送邮件 0:不发送; 1:发送
    int trace_flag;//是否开启 0:不开启; 1:开启
    int effect_day_num;//时间数组数量
```

```
    int output_flag;//是否外部输出 0:不输出 1:输出
    int output_hold;//外部输出持续时间 10-300s
    int audio_flag;//是否音乐提醒 1:是; 0:否
    int audio_index;//音乐文件索引,0-2
    int audio_mode;//音乐播放模式 1:播放次数; 2:持续时间
    int audio_value;//音乐播放值,随模式定义:(持续时间:秒数)(播放次数:播放次数 0-100)
    SGP_EFFECT_DAY effect_day[7];//时间数组
};
```

结构体定义描述

```
struct SGP_CONFIG
{
    int type;//报警类型 1:高温报警; 2:低温报警; 3:平均温报警; 4:高低温报警
    int condition;//条件 1:高于; 2:低于 3:匹配;
    float high_temp;//配置高温
    float low_temp;//配置低温
    float avg_temp;//配置平均温
    int objtype;//类型 0:冷热点; 1:点; 2:线; 3:矩形; 4:多边形;5:圆形
    SGP_POINT points[7];
};
```

结构体定义描述

```
struct SGP_EFFECT_DAY
{
    int day;//1-7,星期几
    int period_num;//时间段数量
    SGP_PERIOD period[6];//时间段
};
```

结构体定义描述

```
struct SGP_EMAIL_INFO
{
    int alarm;//是否使用报警邮件 0:否;1:是
    int alarm_value;//报警邮件间隔 1-3600 秒
    int enclosure;//是否带附件 0:否; 1:是
    int encry_type;//加密方式 0:none; 1:tls; 2:ssl
    char from[STRING_LENGH];//发件人
    int health;//是否使用健康邮件 0:否; 1:是
    int health_value;//健康邮件间隔 1-3600 秒
    int is_anon;//是否匿名 0:否; 1:是
    char password[STRING_LENGH];//登录密码，密文传输
    int smtp_port;//smtp 服务端口，默认 25
    char smtp_server[STRING_LENGH];//smtp 服务器,默认空
xxx.xxx.xxx.xxx
    char subject[STRING_LENGH];//主题
    char username[STRING_LENGH];//登录服务器用户名
    int mailto_num;//收件人数量
    char mailto[5][STRING_LENGH];//收件人列表
};
```

结构体定义描述

```
struct SGP_FILL_LIGHT_INFO
{
    int brightness;/*亮度 0-100,0 - 20 一档;21 - 40 二档;41 - 60 三档;61 - 80 四档;81 - 100 五档*/
    int light;//灯开启状态 0:关闭; 1:开启
    int mode;//灯模式 0:手动; 1:自动
};
```

结构体定义描述

```
enum SGP_FOCUS_TYPE
{
    SGP_FOCUS_STOP = 0,        //电机停止
    SGP_FOCUS_FAR = 1,         //远焦
    SGP_FOCUS_NEAR = 2,        //近焦
    SGP_FOCUS_FAR_FINE = 3,    //远焦微调
    SGP_FOCUS_NEAR_FINE = 4,   //近焦微调
    SGP_FOCUS_AUTO = 5,        //自动聚焦
    SGP_FOCUS_PLACE = 6,       //设置位置
};
```

结构体定义描述

```
struct SGP_GENERAL_INFO
{
```

```
        char datetime[STRING LENGH];//系统时间，格式为 2020-05-21 12:55:12
        char ir_rtsp_url[STRING LENGH];//红外主码流 rtsp 地址
        char ir_sub_rtsp_url[STRING LENGH];//红外辅码流 rtsp 地址
        int ir_model_w;//红外模组宽
        int ir_model_h;//红外模组高
        int ir_output_w;//红外通道输出宽
        int ir_output_h;//红外通道输出高
        int range_num;//测温范围数量
        SGP RANGE range[RANGE MAX NUM];//测温范围
        char vl_rtsp_url[STRING LENGH];//可见光主码流 rtsp 地址
        char vl_sub_rtsp_url[STRING LENGH];//可见光辅码流 rtsp 地址
};
```

类型定义描述

```
typedef unsigned long long SGP_HANDLE;
```

结构体定义描述

```
struct SGP_IAMGE_EFFECT_PARAM_IR_CONFIG
{
    int auto_shutter;//快门自动补偿时间 1-20(单位分钟)
    int brightness;//亮度，取值范围 0-100
```

```
    int contrast;//对比度，取值范围 0-100
    int reverse;//是否反转，0:不反转 1 反转
    int time_flag;//降噪时域滤波开关:0 关闭;1 开启
    int time_value;//降噪时域滤波值 0-100
    int space_flag;//降噪空域滤波开关:0 关闭;1 开启
    int space_value;//降噪空域滤波值 0-100
    int iee_flag;//细节增强开关:0 关闭;1 开启
    int iee_value;//细节增强值 0-100
    int saturation;//饱和度，取值范围 0-100
    int sharpness;//锐度，取值范围 0-100
};
```

结构体定义描述

```
struct SGP_IAMGE_EFFECT_PARAM_VL_CONFIG
{
    int blc;//背光补偿:0 关闭; 1 上; 2 下; 3 左; 4 右; 5 中; 6 自动
    int brightness;//亮度，取值范围 0-100
    int contrast;//对比度，取值范围 0-100
    int exp;//曝光补偿：0-100
    int hlc;//强光抑制:0 关闭;1 开启
    int reverse;//是否反转，0:不反转 1 反转
    int saturation;//饱和度，取值范围 0-100
    int sharpness;//锐度，取值范围 0-100
    int wdr;//宽动态 0:关闭; 1:20%; 2:40%; 3:60%; 4:80%;
5:100%
};
```

结构体定义描述

```
struct SGP_IMAGE_FUSION
{
    int percent;//融合比例值 0-100
    int ir_left;//红外图像左边裁剪像素值 0～50
    int ir_right;//红外图像右边裁剪像素值 0～50
    int ir_top;//红外图像上边裁剪像素值 0～50
    int ir_botton;//红外图像下边裁剪像素值 0～50
    int vl_left;//可见光图像左边裁剪像素值 0～1000
    int vl_right;//可见光图像右边裁剪像素值 0～1000
    int vl_top;//可见光图像上边裁剪像素值 0～1000
    int vl_botton;//可见光图像下边裁剪像素值 0～1000
    SGP_IMAGE_FUSION_MATCH_POINTS ir_match_points;//红外校准点
    SGP_IMAGE_FUSION_MATCH_POINTS vl_match_points;//可见光校准点
};
```

结构体定义描述

```
struct SGP_IMAGE_FUSION_MATCH_POINTS
{
    SGP_POINT point1;//第 1 个校准点
    SGP_POINT point2;//第 2 个校准点
    SGP_POINT point3;//第 3 个校准点
    SGP_POINT point4;//第 4 个校准点
    SGP_POINT point5;//第 5 个校准点
};
```

结构体定义描述

```
enum SGP_IMAGE_TYPE
{
    SGP_VL_IMAGE = 1,//可见光图片
    SGP_IR_IMAGE = 2,//红外图片
};
```

结构体定义描述

```
enum SGP_IR_IMAGE_EFFECT_ENUM
{
    SGP_IR_AUTO_SHUTTER = 1,//快门自动补偿时间 1-20(单位分钟)
    SGP_IR_BRIGHTNESS = 2,//亮度，取值范围 0-100
    SGP_IR_CONTRAST = 3,//对比度，取值范围 0-100
    SGP_IR_REVERSE = 4,//是否反转，0:不反转 1 反转
    SGP_IR_TIME_FLAG = 5,//降噪时域滤波开关:0 关闭;1 开启
    SGP_IR_TIME_VALUE = 6,//降噪时域滤波值 0-100
    SGP_IR_SPACE_FLAG = 7,//降噪空域滤波开关:0 关闭;1 开启
    SGP_IR_SPACE_VALUE = 8,//降噪空域滤波值 0-100
    SGP_IR_IEE_FLAG = 9,//细节增强开关:0 关闭;1 开启
    SGP_IR_IEE_VALUE = 10,//细节增强值 0-100
    SGP_IR_SATURATION = 11,//饱和度，取值范围 0-100
    SGP_IR_SHARPNESS = 12,//锐度，取值范围 0-100
};
```

结构体定义描述

```
struct SGP_MEMORYFULLNOTIFY
{
    int total;//总存储，单位 M
    int free;//可用大小，单位 M
    int limit;//报警阈值，可用小于报警阈值时报警，单位 M
};
```

结构体定义描述

```
struct SGP_NET_EXCEPTION_INFO
{
    int audio_flag;//是否音频联动 0:否; 1:是
    int audio_index;//音频文件索引 0-2
    int audio_mode;//音频模式 0:持续时间; 1:播放次数
    int audio_value;//音频模式值 0-100(次/秒)
    int flag;//是否开启 0:不开启; 1:开启
    int light_flag;//是否闪光灯 0:否; 1:是
    int light_hold;//闪光灯持续时间 10-300s
    int output_flag;//是否外部输出 0:不输出 1:输出
    int output_hold;//外部输出持续时间 1-300s
};
```

结构体定义描述

```
struct SGP_NET_INFO
```

```
{
    int card;//网卡类型:0 有线网卡
    char dns1[STRING LENGH];//dns 服务器 xxx.xxx.xxx.xxx
    char dns2[STRING LENGH];//dns 服务器 xxx.xxx.xxx.xxx
    char gateway[STRING LENGH];//网关 xxx.xxx.xxx.xxx
    char host_name[STRING LENGH];//主机名
    int ip_version;//版本 0:ipv4; 1:ipv6
    char ipaddr[STRING LENGH];//网络 ip 地址 xxx.xxx.xxx.xxx
    char mac[STRING LENGH];//Mac 地址
    int mode;//模式 0:静态; 1:动态
    char netmask[STRING LENGH];//子网掩码 xxx.xxx.xxx.xxx
};
```

结构体定义描述

```
struct SGP_NETWORKERRORNOTIFY
{
    int type;//类型 1:ip 冲突; 2:ping 不通网关，ip 为网关
    char ip[STRING LENGH];//ip 地址
};
```

结构体定义描述

```
struct SGP_PERIOD
{
    char start[STRING LENGH];//开始时间，格式 HH:mm:ss
```

```
    char end[STRING LENGH];//结束时间，格式 HH:mm:ss
};
```

结构体定义描述

```
struct SGP_POINT
{
    int x;//x 坐标 范围 参照红外图像
    int y;//y 坐标 范围 参照红外图像
};
```

结构体定义描述

```
struct SGP_PORT_INFO
{
    int http_port;//http 服务器端口，默认端口 80 保留设置
    int max_connectios;//最大 web 连接数， 1-20
    int onvif_check;//Onvif 登录校验 0:不校验; 1:校验
    int rtsp_port;//红外 rtsp 端口，1024-65535，端口用于 rtsp 流服务，默认端口 554 保留设置
    int tcp_port;//web 消息交互端口，不可设置
};
```

结构体定义描述

```
struct SGP_RANGE
{
    int min;//测温范围最低温
    int max;//测温范围最高温
};
```

结构体定义描述

```
struct SGP_RECORD_INFO
{
    int record_interval;//延时录制时间 1-3600 秒
    int record_max_size;//录制文件最大大小，单位 M，1-1000
    int record_time;//录制时长，单位秒，1-3600 分钟
};
```

结构体定义描述

```
struct SGP_RECT
{
    int x;//x 坐标， 1-图像宽
    int y;//y 坐标， 1-图像高
    int w;//区域宽，与坐标共同作用，取值范围 1-图像宽
    int h;//区域高，与坐标共同作用，取值范围 1-图像高
};
```

结构体定义描述

```
struct SGP_RULE
{
    int id;//分析对象 id，内部分配，无需设置。
    int alarm_condition;//报警条件:1 高于;2 低于;3 匹配;4 高于和低于
    int alarm_flag;//是否报警:0 不需要;1 需要
    int alarm_time;//持续时间,0-100 秒
    int alarm_type;//报警类型:1 高温报警;2 低温报警;3 平均温报警;4 最高温+最低温报警
    float avg_temp;//平均温-20~550
    int flag;//是否启用配置:0 不启用;1 启用
    float high_temp;//报警高温阈值,-20~550
    float low_temp;//报警低温阈值,-20~550
    int points_num;
    SGP_POINT points[7];//矩形，圆是四个点，顺时针顺序，且构成的矩形应用图片的边缘平行
    char rule_name[STRING_LENGH];//规则名称，支持 50 字符
    int show_location;//名称显示位置:1 上方;2 下方;3 左方;4 右方;5 中间
    float temp_mod;//温度误差
    int type;//对象类型:1 点;2 线;3 矩形;4 多边形;5 圆
    float atmo_trans;//大气透过率 0.01-1
    float dist;//距离，单位米，0.1-20.0
    float emiss;//发射率 0.1-1.0
    int emiss_mode;//发射率类型:1 标准;2 自定义
    int humi;//湿度,范围 1-100
    float opti_trans;//光学透过率 0.01-1
    float ref_temp;//反射温度-20~550,单位摄氏度
};
```

结构体定义描述

```
struct SGP_RULE_ARRAY
{
    int rule_num;
    SGP_RULE rule[ANALYTIC_MAX_NUM];//规则列表
};
```

结构体定义描述

```
struct SGP_SHIELD_AREA_INFO
{
    int rect_num;
    SGP_RECT rect[SHIELD_AREA_MAX_NUM];//区域数组,左上角坐标 0,0 标准,最多支持两个
};
```

结构体定义描述

```
enum SGP_SHUTTER_ENUM
{
    SGP_SHUTTER = 1,//快门操作
    SGP_SHUTTER_OPEN = 2,//快门常开
    SGP_SHUTTER_CLOSE = 3,//快门常闭
    SGP_SHUTTER_AUTO = 4,//自动快门
};
```

结构体定义描述

```
struct SGP_TEMP_ALARM_INFO
{
    int audio_flag;//是否音乐提醒 1:是; 0:否
    int audio_index;//音乐文件索引,0-2
    int audio_mode;//音乐播放模式 1:播放次数; 2:持续时间
    int audio_value;//音乐播放值,随模式定义:(持续时间:秒数)(播放次数:播放次数 0-100)
    int alarm_flag;//是否开启报警 1:开启; 0:不开启
    int light_hold;//闪光灯持续时间,10-300s
    int light_flag;//是否开启闪光灯 0:否; 1:是
    int alarm_shake;//报警抖动 0-100s
    int capture_flag;//是否截图 0:否; 1:是
    int capture_stream;//截图类型 0:不截图; 1:只截图可见光; 2:只截图红外; 3:截图红外和可见光
    int record_delay;//录制延时 10-300
    int record_flag;//是否录制 0:不录制; 1:录制
    int record_stream;//录制类型 0:不录制; 1:只录制可见光; 2:只录制红外; 3:录制红外和可见光
    int sendmail;//是否发送邮件 0:不发送; 1:发送
    int effect_day_num;//时间数组数量
    int output_flag;//是否外部输出 0:不输出 1:输出
    int output_hold;//外部输出持续时间 10-300s
    SGP_EFFECT_DAY effect_day[7];//时间数组
};
```

结构体定义描述

```
struct SGP_TEMPALARMNOTIFY
{
    char vl_image_url[STRING LENGH];//报警记录可见光截图
    char vl_video_url[STRING LENGH];//可见光视频地址
    char ir_image_url[STRING LENGH];//报警记录红外截图
    char ir_video_url[STRING LENGH];//红外视频地址
    float high_temp;//高温温度,高温报警时有效
    float low_temp;//低温温度,低温报警时有效
    float avg_temp;//平均温度,平均温报警时有效
    int temp_flag;//报警类型，0 代表平均温，1 代表高温报警，2 代表低温报警，3 代表高低温报警
    int type;//1:温度报警； 2:热点报警； 3:冷点报警
    char name[STRING LENGH];//名称
    char time[STRING LENGH];//报警时间，格式为 2020-05-21 12:22:33
    SGP CONFIG config;//配置
};
```

结构体定义描述

```
struct SGP_THERMOMETRY_PARAM
{
    int color_bar;//色带 1-26
    int color_show;//色带显示 0~1
    int flag;//测温开关 0~1
    float mod_temp;//温度修正
    int show_mode;//温度显示方式：1 最高温 2 最低温 3 平均温 4 最
```

```
高温 + 最低温
                                5 最高温 + 平均温 6 平均温 + 最低温 7 最高温
+ 最低温 + 平均温 8 不显示
    int gear;//测温范围
    int show_string; //是否使用字符叠加 其他机型 1:关闭; 2,4,5:
右下; 3:右上
                                IPT640M 1:关闭; 2:左上; 3:右上; 4:左下;
5:右下
    char show_desc[STRING LENGH];//显示字符串
    float atmo_trans;//大气透过率 0.01-1
    float dist;//距离,单位米,0.1-20.0
    float emiss;//发射率 0.1-1.0
    int emiss_mode;//发射率类型:1 标准;2 自定义
    int humi;//湿度,范围 1-100
    float opti_trans;//光学透过率 0.01-1
    float ref_temp;//反射温度-20~550,单位摄氏度
    int isot_flag;//等温线开关 0:关闭;1 开启
    float isot_high;//高温阈值 0~400
    char isot_high_color[STRING LENGH];//高温颜色,十六进制值,如
红色:0xff0000
    int isot_low;//低温阈值-50~-100
    char isot_low_color[STRING LENGH];//低温颜色,十六进制值,如
红色:0xff0000
    int isot_type;//范围类型:1 关闭等温线效果 2 开启高等温线 3 开
启低等温线 4 开启区间内等温线 5 开启区间外等温线
    float ambient;//环境温度
};
```

结构体定义描述

```
struct SGP_VERSION_INFO
{
    char model[STRING LENGH];            //设备型号
    char version[STRING LENGH];          //系统版本
    char serial[STRING LENGH];           //序列号
    char fpga_version[STRING LENGH];     //FPGA 版本
    char measure_version[STRING LENGH]; //测温版本
    char sdk_version[STRING LENGH];      //sdk 版本
};
```

结构体定义描述

```
struct SGP_VIDEO_PARAM
{
    int bit_size;//主码流固定码流值，
可变码流时也需设置（宽*高*1.5*fps*8/压缩率）其中压缩率范围（18-500）
如分辨率 1280x720    取值范围：540Kb/s - 15000Kb/s
    int encodec;//主码流编码 0:h264; 1:h265; 2:mjpeg
    int fps;//主码流帧率 1-25
    int gop_size;//主码流帧间隔 1-50
    int level;//编码质量等级，等级效果随实际变化，如使用 ffmpeg，需
服务端自映射（
默认 medium， 可以上下延续几个等级）1:最好 2:更好 3:好 4:差 5:更差
6:最差
    int rate_control;//主码流控制 0:可变码流;1:固定码流
    char ratio[STRING LENGH];/*主码流分辨率
                                1920x1080
                                1280x960
                                1280x720
```

```
                                        可见光辅码流分辨率
                                        704x576
                                        640x480
                                        红外主码流分辨率
                                        512x384(640*512)
                                        红外辅码流分辨率
                                        (384*288)256x192*/
    int svc;//帧率可分层编码，H264 有效，其他格式也需传入 0:分层编码; 1:不分层编码
};
```

结构体定义描述

```
enum SGP_VIDEO_PARAM_ENUM
{
    SGP_VL = 1,         //可见光主码流
    SGP_VL_SUB = 2,     //可见光辅码流
    SGP_IR = 3,         //红外主码流
    SGP_IR_SUB = 4,     //红外辅码流
};
```

结构体定义描述

```
enum SGP_VIDEO_TYPE
{
    SGP_VL_VIDEO = 1,//可见光录像
    SGP_IR_VIDEO = 2,//红外录像
};
```

结构体定义描述

enum SGP_VL_IMAGE_EFFECT_ENUM
{
SGP_VL_BLC = 1,//背光补偿:0 关闭; 1 上; 2 下; 3 左; 4 右; 5 中; 6 自动
SGP_VL_BRIGHTNESS = 2,//亮度，取值范围 0-100
SGP_VL_CONTRAST = 3,//对比度，取值范围 0-100
SGP_VL_EXP = 4,//曝光补偿：0-100
SGP_VL_HLC = 5,//强光抑制:0 关闭;1 开启
SGP_VL_REVERSE = 6,//是否反转，0:不反转 1 反转
SGP_VL_SATURATION = 7,//饱和度，取值范围 0-100
SGP_VL_SHARPNESS = 8,//锐度，取值范围 0-100
SGP_VL_WDR = 9,//宽动态 0:关闭; 1:20%; 2:40%; 3:60%; 4:80%; 5:100%
};

宏定义

定义	数值	描述
STRING_LENGH	50	一般长度
RANGE_MAX_NUM	3	测温范围值
ANALYTIC_MAX_NUM	21	分析对象个数
SHIELD_AREA_MAX_NUM	2	屏蔽区域个数

错误码

错误码	定义	描述
0	SGP_OK	正常
1	SGP_ERR	错误

版權所有©翌宙科技股份有限公司