

扫描仪动态库版本 SDK 接口说明文档

一. SDK 说明

本 SDK 适用于华高 G100、G200、G400 系列扫描仪调用，可通过 API 进行扫描仪的初始化、配置、图像获取、存储等调用，供第三方应用程序需要调用扫描仪模块进行相应的纸质文档电子化操作。

二. 变更说明

版本号	时间	编辑	更新说明
V2.0	2022-07-18	PM	初始版本
V2.1	2022-09-17	PM	调整简化参数配置项接口，调整协议类型

三. 系统适配说明

本 SDK 目前仅支持 Windows 以及 AMD64、ARM64 架构的银河麒麟 V10 以上及统信 UOS 操作系统，后续将补充其他国产平台下的版本。

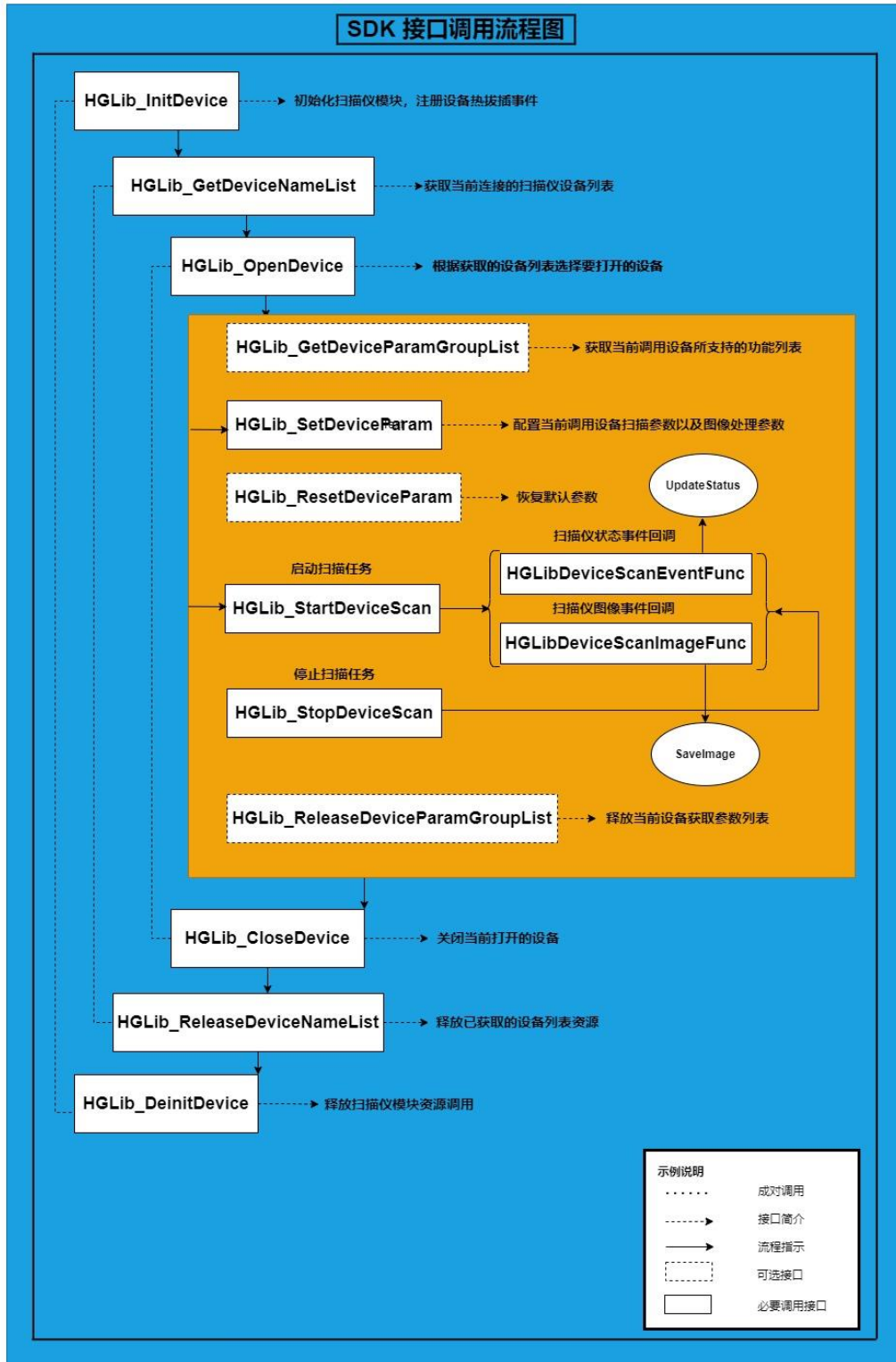
四. SDK 目录结构说明

目录编号	文件夹/文件名	文件说明	备注
1	bin	测试程序	需自行编译构建 (cmake)
2	doc	接口说明文档目录	
3	example	示例 demo 文件夹	C++ demo 示例程序源码以及 cmake 构建规则；以及 python 调用 demo 程序
4	include	C++ 开发所依赖的头文件	
5	lib	依赖库存放目录	调用程序需将相关动态库放置在调用程序运行目录下；同

			时提供了 amd64 以及 arm64 动态库。
6	60-lanxumsane.rules	USB 权限配置文件	<ol style="list-style-type: none"> 1. rules 文件复制到 /etc/udev/rules.d 目录 2. 执行以下指令或者重启 PC 均可： <ol style="list-style-type: none"> 1) sudo udevadm control --reload 2) sudo udevadm trigger
7	README	readme	相关权限以及使用说明

五. 接口说明

1. 调用流程



2. 接口说明

1) 加载图像

接口说明：加载指定路径的图像，转换为 `HGLibImage` 类型，支持的图像格式为 jpg、png、bmp、tiff、gif、pdf 和 ofd；

传参说明：

filePath: [in] 文件路径，UTF8 格式

返回值：成功则返回 `HGLibImage` 对象；失败则返回 `NULL`；

```
HGEXPORT HGLibImage HGAPI HGLib_LoadImage(const HGChar *filePath);
```

2) 保存图像

接口说明：将 `HGLibImage` 类型，存储为支持的图像格式为 jpg、png、bmp、tiff、gif、pdf 和 ofd；

传参说明：

image: [in] `HGLibImage` 对象；

savePath: [in] 待存储图像的完整类型，包含文件路径以及后缀；

saveParam: 待保存图像控制参数，`NULL` 表示使用默认参数；

返回值：调用成功返回 `HGTRUE`；

失败则返回 `HGFALSE`；

```
HGEXPORT HGBool HGAPI HGLib_SaveImage(HGLibImage image, const HGChar* savePath, const HGLibSaveImageParam* saveParam);
```

3) 释放图像

接口说明：释放 `HGLibImage` 类型对象句柄

传参说明：

image: [in] `HGLibImage` 对象；

返回值：调用成功返回 `HGTRUE`；

失败则返回 `HGFALSE`；

```
HGEXPORT HGBool HGAPI HGLib_ReleaseImage(HGLibImage image);
```

4) 初始化扫描仪

接口说明：初始化设备相关的调用资源，调用设备前必须调用此接口，与 `HGLib_DeinitDevice` 接口成对调用；

传参说明：

func: [in] 待绑定的回调函数，可以为 `NULL`；

Param: [in] 回调参数

返回值：调用成功返回 `HGTRUE`；

失败则返回 `HGFALSE`；

```
HGEXPORT HGBool HGAPI HGLib_InitDevice(HGLibDeviceHotPlugEventFunc func, HGPointer param);
```

5) 反初始化扫描仪

接口说明：释放扫描仪模块资源，最终退出时调用此接口；

传参说明：

无

返回值： 调用成功返回 HGTRUE；
失败则返回 HGFALSE；

```
HGEXPORT HGBool HGAPI HGLib_DeinitDevice();
```

6) 获取扫描仪名称列表

接口说明：返回扫描仪的名称列表，不需要使用的时候需要调用

HGLib_ReleaseDeviceNameList 销毁；

返回值： 调用成功，返回已连接的设备列表，以字符数组的形式返回；
失败则返回 NULL；

```
HGEXPORT HGChar** HGAPI HGLib_GetDeviceNameList();
```

7) 销毁扫描仪名称列表

接口说明：释放获取到的扫描仪列表资源占用；

传参说明：

deviceName: [in] 扫描仪的名称列表

返回值： 调用成功返回 HGTRUE；
失败则返回 HGFALSE；

```
HGEXPORT HGBool HGAPI HGLib_ReleaseDeviceNameList(HGChar** deviceName);
```

8) 打开设备

接口说明：根据 ReleaseDeviceNameList 调用返回的在线设备列表，传入待打开的设备名；

传参说明：

deviceName: [in] 待打开的设备名；

返回值： 调用成功返回 HGLibDevice 设备句柄；
失败则返回 NULL；

```
HGEXPORT HGLibDevice HGAPI HGLib_OpenDevice(const HGChar *deviceName);
```

9) 关闭设备

接口说明：关闭已打开的设备句柄；

传参说明：

device: [in] 已打开的设备句柄；

返回值： 调用成功返回 HGTRUE；
失败则返回 HGFALSE；

```
HGEXPORT HGBool HGAPI HGLib_CloseDevice(HGLibDevice device);
```

10) 设置设备参数

接口说明：配置已打开设备的扫描参数，详见常见附表参数；

传参说明：

device: [in] 已打开的设备句柄；

option: [in] 待配置的参数协议项 ID，见头文件定义；

data: [in] 配置的值，只能为 HGInt、HGUInt、HGDouble 或 HGBool 之一的指针，根据 option 决定；

返回值： 调用成功返回 HGTRUE；

失败则返回 HGFALSE；

```
HGEXPORT HGBool HGAPI HGLib_SetDeviceParam(HGLibDevice device, HGUInt option, const HGVoid* data);
```

11) 获取所有的扫描参数项信息

接口说明：获取所有的扫描参数项信息，详见常见附表参数；不使用时需要调用 HGLib_ReleaseDeviceParamGroupList 销毁；

传参说明：

device: [in] 已打开的设备句柄；

count: [out] 支持的设备参数组列表个数；

返回值： 调用成功返回指向 HGLibDeviceParamGroup 的指针；

失败则返回 NULL；

```
HGEXPORT HGLibDeviceParamGroup* HGAPI HGLib_GetDeviceParamGroupList (HGLibDevice device, HGUInt *count);
```

12) 获取指定配置参数项

接口说明：获取指定的扫描参数项信息，详见常见附表参数；不使用时需要调用 HGLib_ReleaseDeviceParam 销毁；

传参说明：

device: [in] 已打开的设备句柄；

option: [in] 协议项 ID；

返回值： 调用成功返回指向指定 option 的 HGLibDeviceParam 的指针；

失败则返回 NULL；

```
HGEXPORT HGLibDeviceParam* HGAPI HGLib_GetDeviceParam(HGLibDevice device, HGUInt *count);
```

13) 销毁设备参数组

接口说明：释放 HGLib_GetDeviceParamGroupList 调用所占用的资源；

传参说明：

paramGroup: [in] 已打开的设备句柄；

count: [in] 支持的设备参数组列表个数；

返回值： 调用成功返回 HGTRUE；

失败则返回 HGFALSE；

```
HGEXPORT HGBool HGAPI HGLib_ReleaseDeviceParamGroupList (HGLibDeviceParamGroup* paramGroup, HGUInt count);
```

14) 销毁设备参数

接口说明：释放 HGLib_GetDeviceParam 调用所占用的资源；

传参说明：

param: [in] 已打开的设备句柄;
返回值: 调用成功返回 HGTRUE;
失败则返回 HGFALSE;

```
HGEXPORT HGBool HGAPI HGLib_ReleaseDeviceParam(HGLibDeviceParam* param);
```

15) 重置设备参数

接口说明: 恢复默认扫描参数 (即 200dpi、彩色、匹配原始尺寸、自动纠偏、消除黑框、双面扫描);

传参说明:

device: [in] 已打开的设备句柄;
返回值: 调用成功返回 HGTRUE;
失败则返回 HGFALSE;

```
HGEXPORT HGBool HGAPI HGLib_ResetDeviceParam(HGLibDevice device);
```

16) 检查是否有纸

接口说明: 检查设备上是否有纸;

传参说明:

device: [in] 已打开的设备句柄;
返回值: 有纸返回 HGTRUE;
device 无效或无纸返回 HGFALSE;

```
HGEXPORT HGBool HGAPI HGLib_DeviceIsPaperOn(HGLibDevice device);
```

17) 开始扫描

接口说明: 启动一次扫描任务, 待扫描完成后再次调用, 扫描启动期间, 图像以及扫描状态会通过接口注册的扫描事件回调以及图像事件回调通知回传, 调用者需要在此回调函数中响应扫描状态的变更 (例如扫描开始、扫描停止、扫描异常消息) 以及有图回调函数 (在回调中存储图像);

传参说明:

device: [in] 已打开的设备句柄;
eventFunc: [in] 待绑定的扫描事件回调函数实例;
eventParam: [in] 指向调用者的指针, 一般类中传递 this 指针;
imageFunc: [in] 待绑定的扫描图像回调函数实例;
imageParam: [in] 指向调用者的指针, 一般类中传递 this 指针;
返回值: 调用成功返 HGTRUE;
失败则返回 HGFALSE;

```
HGEXPORT HGBool HGAPI HGLib_StartScan(HGLibDevice device, HGLibDeviceScanEventFunc eventFunc, HGPointer eventParam, HGLibDeviceScanImageFunc imageFunc, HGPointer imageParam);
```

18) 停止扫描

接口说明: 停止正在进行的扫描任务, 停止后将会清除底层可能尚未传输的图像;

传参说明:

device: [in] 已打开的设备句柄;
返回值: 调用成功返 HGTRUE;
失败则返回 HGFALSE;

```
HGEXPORT HGBool HGAPI HGLib_StopScan(HGLibDevice device);
```

六. 参数说明

1. 宏定义

1) TIFF 压缩方式

```
#define HGLIB_TIFFCOMPRESSION_NONE 1L /* 不压缩 */  
#define HGLIB_TIFFCOMPRESSION_CCITTFAX4 2L /* CCITTFAX4 压缩 */  
#define HGLIB_TIFFCOMPRESSION_LZW 3L /* LZW 压缩 */  
#define HGLIB_TIFFCOMPRESSION_JPEG 4L /* JPEG 压缩 */
```

2) 设备热拔插事件

```
#define HGLIB_DEVHOTPLUG_EVENT_ARRIVE 1L /* 设备拔走 */  
#define HGLIB_DEVHOTPLUG_EVENT_REMOVE 2L /* 设备插上 */
```

3) 设备扫描事件

```
#define HGLIB_DEVSCAN_EVENT_BEGIN 1L /* 扫描开始 */  
#define HGLIB_DEVSCAN_EVENT_END 2L /* 扫描结束 */  
#define HGLIB_DEVSCAN_EVENT_INFO 3L /* 扫描信息 */
```

4) 设备参数类型

```
#define HGLIB_OPTION_VALUETYPE_INT 1L /* 整型 */  
#define HGLIB_OPTION_VALUETYPE_ENUM2L /* 宏/枚举值 */  
#define HGLIB_OPTION_VALUETYPE_DOUBLE 3L /* 浮点型 */  
#define HGLIB_OPTION_VALUETYPE_BOOL4L /* 布尔型 */
```

5) 设备参数取值范围类型

```
#define HGLIB_OPTION_VALUERANGETYPE_NULL 0L  
#define HGLIB_OPTION_VALUERANGETYPE_INTLIST 1L /* 整型列表 */  
#define HGLIB_OPTION_VALUERANGETYPE_ENUMLIST 2L /* 宏/枚举列表 */  
#define HGLIB_OPTION_VALUERANGETYPE_DOUBLELIST 3L /* 浮点型列表 */  
#define HGLIB_OPTION_VALUERANGETYPE_INTRANGE 4L /* 整型范围 */  
#define HGLIB_OPTION_VALUERANGETYPE_DOUBLERANGE 5L /* 浮点型范围 */
```

2. 参数定义

1) 图像保存参数

```
typedef struct  
{  
    HGUInt size; /* 结构体大小, 必须初始化为 sizeof(HGLibSaveImageParam) */
```



```
    HGUInt jpegQuality; /* jpeg 下有效, 0-100 */
    HGUInt tiffCompression; /* tiff 下有效, HGLIB_TIFFCOMPRESSION_*/
    HGUInt tiffJpegQuality; /* tiff 且 HGLIB_TIFFCOMPRESSION_JPEG 下有效, 0-100 */
    HGBool ocr; /* 是否 OCR, pdf 和 ofd 格式有效 */
}HGLibSaveImageParam;
```

2) 设备参数整型列表

```
typedef struct
{
    HGInt* value;
    HGUInt count;
}HGLibDeviceIntValueList;
```

3) 设备参数宏/枚举列表

```
typedef struct
{
    HGInt* value;
    HGUInt count;
}HGLibDeviceEnumValueList;
```

4) 设备参数浮点型列表

```
typedef struct
{
    HGDouble* value;
    HGUInt count;
}HGLibDeviceDoubleValueList;
```

5) 设备参数整型范围

```
typedef struct
{
    HGInt minValue;
    HGInt maxValue;
}HGLibDeviceIntValueRange;
```

6) 设备参数浮点型范围

```
typedef struct
{
    HGDouble minValue;
    HGDouble maxValue;
}HGLibDeviceDoubleValueRange;
```

7) 获取设备参数

```
typedef struct
{
    HGUInt option; /* 配置名, 参见 HGLIB_OPTION_NAME_*/

    HGUInt type; /* 值类型, 参见 HGLIB_OPTION_VALUETYPE_*/
    union
    {
        HGInt intValue;
        HGUInt enumValue;
        HGDouble doubleValue;
        HGBool boolValue;
    };

    HGUInt rangeType; /* 值范围类型, 参见 HGLIB_OPTION_VALUERANGETYPE_*/
    union
    {
        HGLibDeviceIntValueList intValueList;
        HGLibDeviceEnumValueList enumValueList;
        HGLibDeviceDoubleValueList doubleValueList;
        HGLibDeviceIntValueRange intValueRange;
        HGLibDeviceDoubleValueRange doubleValueRange;
    };
}HGLibDeviceParam;
```

8) 获取设备参数组

```
typedef struct
{
    HGUInt group; /* 组名, 参见 HGLIB_GROUP_NAME_*/
    HGLibDeviceParam* param;
    HGUInt paramCount;
}HGLibDeviceParamGroup;
```

3. 回调函数定义

1) 设备热拔插回调函数定义

```
/* event: HGLIB_DEVHOTPLUG_EVENT_*
 * deviceName: 设备名
 * param: 回调参数
 */
```

```
typedef void (*HGLibDeviceHotPlugEventFunc)(HGUInt event, const HGChar *deviceName, HGPointer param);
```

2) 设备扫描事件回调

```
/* device: 设备句柄
 * event: HGLIB_DEVSCAN_EVENT_*
 * err: 是否是错误信息, HGLIB_DEVSCAN_EVENT_INFO 下有效
 * info: 扫描信息, HGLIB_DEVSCAN_EVENT_INFO 下有效
 * param: 回调参数
 */
```

```
typedef void (*HGLibDeviceScanEventFunc)(HGLibDevice device, HGUInt event, HGBool err, const
HGChar *info, HGPointer param);
```

3) 设备扫描图像回调

```
/* device: 设备句柄
 * image: 图像句柄, 不要调用 HGLib_ReleaseImage 释放, 回调执行后会自动释放
 * param: 回调参数
 */
```

```
typedef void (*HGLibDeviceScanImageFunc)(HGLibDevice device, HGLibImage image, HGPointer
param);
```

七. 附表

此表中罗列了部分常用的一些配置的协议名以及所支持的值, 具体见 HGSscannerLib.h 所定义的协议名以及协议值, 配置类型见 demo 中配置部分协议写法。

编号	协议 ID(宏定义 int 值)	支持值 (宏定义 int 值)	备注
1	HGLIB_OPTION_NAME_YSMS (颜色模式)	HGLIB_OPTION_ENUMVALUE_YSMS_24WCS	24 位彩色
		HGLIB_OPTION_ENUMVALUE_YSMS_256JHD	256 级灰度
		HGLIB_OPTION_ENUMVALUE_YSMS_HB	黑白图像
		HGLIB_OPTION_ENUMVALUE_YSMS_YSZDSB	自动颜色识别
2	HGLIB_OPTION_NAME_ZZCC	HGLIB_OPTION_ENUMVALUE_ZZCC_A3	A3
		HGLIB_OPTION_ENUMVALUE_ZZCC_8K	8K 纸
		HGLIB_OPTION_ENUMVALUE_ZZCC_A4	A4
		HGLIB_OPTION_ENUMVALUE_ZZCC_A4HX	A4 横向
		HGLIB_OPTION_ENUMVALUE_ZZCC_16K	16K
		HGLIB_OPTION_ENUMVALUE_ZZCC_16KHX	16K

			横向
		HGLIB_OPTION_ENUMVALUE_ZZCC_A5	A5
		HGLIB_OPTION_ENUMVALUE_ZZCC_A5HX	A5 横向
		HGLIB_OPTION_ENUMVALUE_ZZCC_PPYSCC	匹配原始尺寸
		HGLIB_OPTION_ENUMVALUE_ZZCC_ZDSMCCZDCQ	最大扫描尺寸自动裁切
3	HGLIB_OPTION_NAME_SMYM	HGLIB_OPTION_ENUMVALUE_SMYM_DM	单面
		HGLIB_OPTION_ENUMVALUE_SMYM_SM	双面
		HGLIB_OPTION_ENUMVALUE_SMYM_TGKBYTY	跳过空白页通用
		HGLIB_OPTION_ENUMVALUE_SMYM_TGKBYFPZ	跳过空白页（发票纸）
		HGLIB_OPTION_ENUMVALUE_SMYM_DZ	对折
4	HGLIB_OPTION_NAME_WGFX	HGLIB_OPTION_ENUMVALUE_WGFX_0	0°
		HGLIB_OPTION_ENUMVALUE_WGFX_90	旋转90°
		HGLIB_OPTION_ENUMVALUE_WGFX_180	旋转180°
		HGLIB_OPTION_ENUMVALUE_WGFX_90	旋转-90°
		HGLIB_OPTION_ENUMVALUE_WGFX_ZDWBFXS B	自动文本方向识别