

静态平板 SDK 开发 Demo 工程说明

昊博研发部-软件组

2021-01-15

目录

一、 开发环境.....	2
二、 功能说明.....	2
三、 图像校正.....	5
四、 流程说明.....	6
五、 常见问题.....	12

目的：针对二次开发集成用户，易于掌握、方便调试和快速集成。

一、开发环境

win7 以上+独立网卡+vs2013+MFC+Opencv249

设置本地 IP 地址和防火墙，详细见附件静态 SDK API 接口开发说明书。

显示部分基于 Opencv249 开发，请先安装 Opencv249。

二、功能说明

2.1、主界面

如下图 1-1 所示，分为 4 个主要区域：菜单栏、图像显示窗口、状态和控制参数窗口以及功能触发按钮窗口。

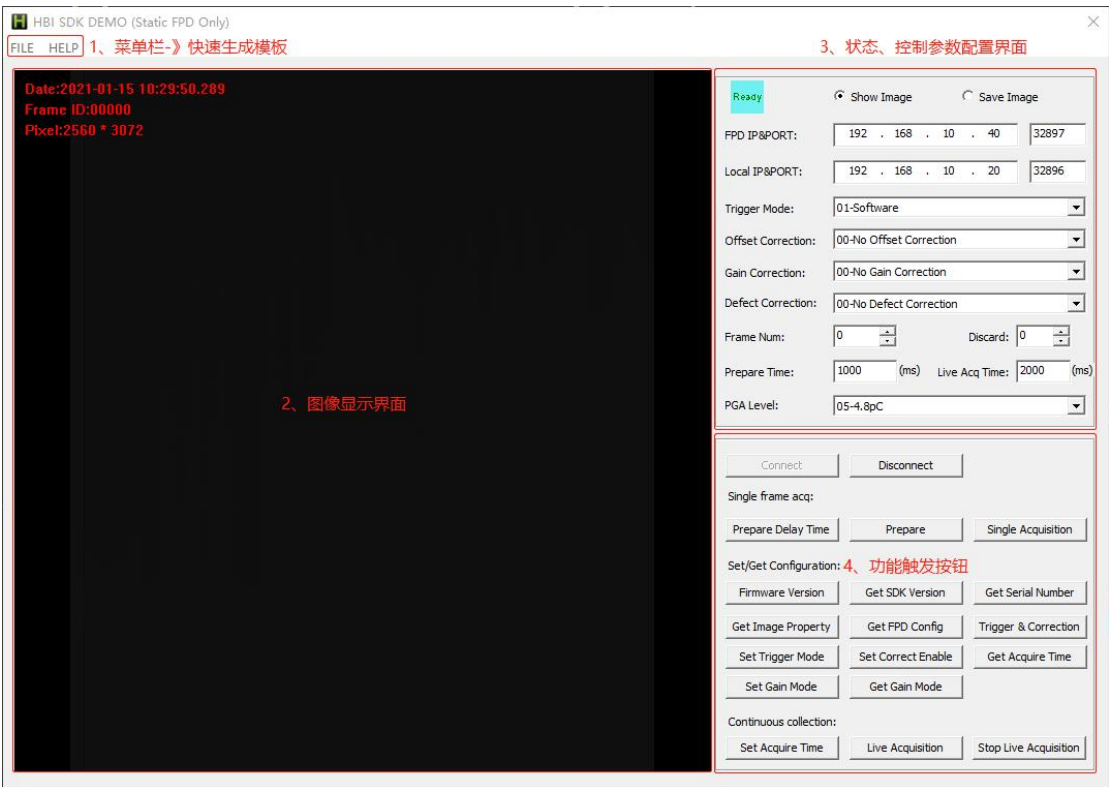


图 1-1

菜单栏-子菜单“快速生成模板”，用户可以通过该窗口生成校正模板；

图像显示窗口-当选择“show Image”显示图像，“Save Image”保存图像。目前没有进行自适应窗宽窗位算法，仅为调试；

状态、控制参数窗口-平板状态（连接、断开、ready 或 busy 等）、平板地址端口、触发模式、校正使能选项、帧率设置、PGA 档位调节等；

功能按钮窗口



序号	按钮	参数或者输出	接口函数	描述
1	Connect	<div> <div>FPD IP&PORT:</div> <div>192 . 168 . 10 . 40</div> <div>32897</div> </div>	HBI_RegEventCallBackFu n(如果没有注册, 请先注册) HBI_ConnectDetector	连接平板
2	Disconnect		HBI_DisConnectDetector	断开平板
3	Prepare Delay Time	<div> <div>Prepare Time:</div> <div>1000</div> <div>(ms)</div> </div>	HBI_SetPreAcqTm 参数: prepare 延时, 0 和 >=1000, unit: ms	设置 prepare 延时时间 参数区别: 主要是软触发模式下的单帧采集略有区别, 0-prepare+siggle shot 命令完成采集一帧图像; >=1000-prepare 命令延时后采集一帧图像。
4	Prepare	<div>Pre</div>	HBI_Prepare	例, 当 prepare 延时为 1000 时, 点击后 1000 秒之后上图, 当“prepare”延时为 0 时, 点击后显示 Pre 状态。
5	Single Acquisition	采集模式参数, 默认 STATIC_ACQ_DEFAULT_MODE 模式	HBI_SingleAcquisition	当 prepare 延时为 0 时, 点击“prepare”再点击“Single Acquisition”后上一帧图像。
6	Firmware Version	<div>OnBtnClickedButtonFirmwareVer</div> <div>szFirmVer=Firmware-2.0.1-200706-release</div>	HBI_GetFirmareVerion	获取固件版本号
7	Get SDK Version	<div>szFirmVer=Firmware-2.0.1-200706-release</div>	HBI_GetSDKVerion	获取 SDK 版本号
8	Get Serial Version	<div>HBI_GetFPDSerialNumber:szSeiralNum=243CS201021001</div>	HBI_GetFPDSerialNumbe r	获取平板序列号
9	Get Image Property	<div> HBI_GetImageProperty: nSize=2 width=2560,height=3072 datatype is unsigned char. databit is 16bits. data is little endian. packet_sum=15360 frame_size=15728640 </div>	HBI_GetImageProperty	
10	Get FPD Config	<div> get all firmware config info! HBI_GetDevCfgInfo:width=1280,height=1024 SourceIP:0.0.0.0:0 DestIP:192.168.10.40:32897 PanelSize:0x03,fpd type:3543 width=1280,height=1024 </div>	HBI_GetDevCfgInfo	
11	Trigger & Correction	<div> <div>Trigger Mode:</div> <div>01-Software</div> <div>Offset Correction:</div> <div>02-Firmware PostOffset Correction</div> <div>Gain Correction:</div> <div>01-Software Gain Correction</div> <div>Defect Correction:</div> <div>01-Software Defect Correction</div> </div>	HBI_TriggerAndCorrectA pply	设置触发模式和校正 使能
12	Set Trigger Mode	<div> <div>Trigger Mode:</div> <div>01-Software</div> </div>	HBI_UpdateTriggerMode	设置触发模式

13	Set Correction Enable	Offset Correction: 02-Firmware PostOffset Correction Gain Correction: 01-Software Gain Correction Defect Correction: 01-Software Defect Correction	HBI_UpdateCorrectEnable	设置校正使能，一版工作站使用固件 offset 校正效果比较好，点料机使用软件 offset 校正。
14	Get Acquire Time		HBI_GetAcqSpanTm	获取连续采集时间间隔
15	Set Gain Mode	PGA Level: 06-7.2pC	HBI_SetGainMode	设置 PGA 档位
16	Get Gain Mode		HBI_GetGainMode	获取 PGA 档位
17	Set Acquire Time	Live Acq Time: 0 (ms)	HBI_SetAcqSpanTm	设置连续采集时间间隔
18	Live Acquisition	Frame Num: 0 Discard: 0 采集模式参数，默认 DYNAMIC_ACQ_DEFAULT_MODE 模式	HBI_LiveAcquisition	连续采集命令：Frame Number: 0一直采集，直到发送停止采集命令；n>=1, 采集 1~n 张图像。Discard: 表示抛弃前 n 张图像。
19	Stop Live Acquisition		HBI_StopAcquisition	停止连续采集

2.2、调试信息打印窗口

如下图 1-2 所示，打印当前调试信息等。

```

HBI_SDK_DEMO2008
@
Local IP Addr:192.168.10.20:32896
Remote IP Addr:192.168.10.40:32897
Image property:
    width=3072,height=3072
    datatype is unsigned char.
    databit is 16bits.
    data is little endian.
trigger_mode[0x01]:
    static software trigger.
Image Calibration:
    No Offset Correction
    No Gain Correction
    No Defect Correction
    No Dummy Correction
    Invalid Dummy Correction
    0x01:1*1
ShowThreadProcFun start!
  
```

图 1-2

2.3、快速生成模板界面

如下图 1-3 所示，生成 Offset、Gain 和 Defect 模板。

✓ Offset 模板

需要采集暗场图像，只需要采集一组图像；

步骤：

Offset 模板一般需要每天重做一次(包括不存在 offset 模板)。

1》不存在 offset 模板或每天开机后先完成 offset 模板的制作。

2》中途有平板断电状况等，再次连接需要下载 offset 模板到固件，否则出现失校正现象。

3》客户可根据 offset 模板文件判断是否重做 offset 模板或者下载模板。

例如：offset 模板不存在或者文件属性创建日期为 T-1 的，重做 offset 模板，上位机发送做 offset 模板命令，软件端完成 offset 模板并在本地保存一张 offset 模板。

✓ Gain 模板

需要采集亮场图像，需要清场，不能在平板上放置物品等，采集 1 组亮场场图，整常高压，毫安秒调节正常的 50%。

步骤：

1》设置固件 Offset 使能：03-preoffset correction；

2》调节好剂量，采集一组亮场并成成模板；

3》将 gain 模板下载到固件；

✓ Defect 需要采集暗场图像，只需要采集一组图像

defect 模板需要采集 3 组亮场图像，需要清场，不能在平板上放置物品等

Group1: 正常高压，毫安秒调节正常的 10%

Group2: 正常高压，毫安秒调节正常的 50%

Group3: 正常高压，毫安秒调节正常

步骤：

1》调节剂量采集第一组亮场；

2》调节剂量采集第二组亮场；

3》调节剂量采集完第三组亮场图，将自动生成 defect 模板，并保存在本地；

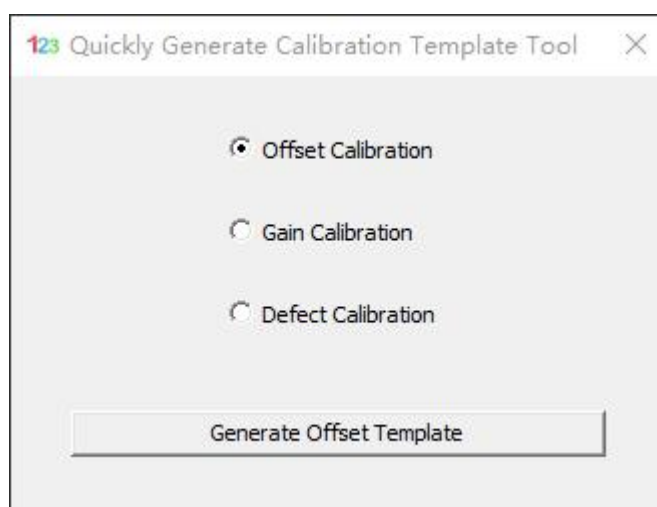


图 1-3

三、图像校正

静态平板一般使用软件校正，固件支持 offset 校正。默认校正使能如下图 1-4 所示。

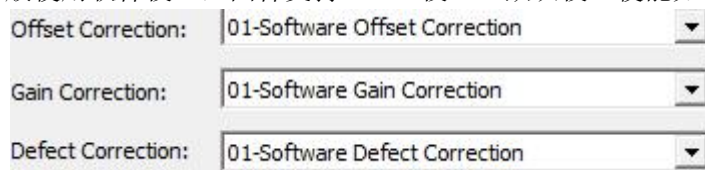


图 1-4

例如：

医用或宠物工作站：固件 offset、软件 gain 和 defect 校正，效果比较好。

点料机：软件 offset、gain 和 defect 校正，效果比较好。

四、流程说明

以本工程为例说明

HBI_DLL 为 SDK 对外接口文件。

HBI_DLL\INCLUDES：包含 HbFpdDll.h、HbDllType.h 和 HbDllError.h 一共三个头文件。

a. 《HbFpdDll.h》：导出函数以及说明，具体可参考《HBI Static FPD API Programming Reference Ver*.pdf；

b. 《HbDllType.h》：命令、回调函数定义返回事件类型以及固件参数结构体；

c. 《HbDllError.h》：错误以及返回码信息表；

HBI_DLL\BIN：动态库库文件，注意 32bits 和 64bits；

HBI_DLL\DOC：API 接口函数说明文档。

1、添加头文件

将"HBI_DLL\INCLUDES"头文件拷贝的用户开发指定目录下；

将动态库文件"HBI_DLL\BIN"拷贝到用户工程目录或者系统目录下，注意 32bits 和 64bits；

2、接口调用流程

```
#define _DLL_EX_IM 0
#include "HbFpdDll.h"
#pragma comment(lib, "HBI_FPD.lib")

// 基本数据类型
void* m_pFpd;
RegCfgInfo* m_pLastRegCfg; // 固件配置
FPD_AQC_MODE aqc_mode;     // 采集模式和参数，单帧采集、多帧采集、暗场图以及亮场图采集
bool IsOpen;
HWND m_templateHwnd;       //快速生成模板窗口句柄
int m_imgW, m_imgH;        // 图像分辨率
char m_path[MAX_PATH];
fpd_base_cfg_st *fpd_base;
IMAGE_PROPERTY img_pro;
char szSdkVer[128];
char szFirmVer[128];
char szSeiralNum[16];
CALIBRATE_INPUT_PARAM m_calibrate_param;

// 初始化 DLL 资源
void *m_pFpd = HBI_Init();
if (m_pFpd == NULL) {
    printf("HBI_Init init failed!\n");
    return false;
}
```

```

// 设置注册回调函数
// myCallbackFun 回调函数
int ret = HBI_RegEventCallBackFun(m_pFpd, myCallbackFun);
if (ret != 0) {
    printf("HBI_RegEventCallBackFun failed!\n");
    HBI_Destroy(m_pFpd);
    m_pFpd = NULL;
    return false;
}
// 连接
// timeout 参数系统已默认, 改为 offset 模板控制参数
// 1-表示初始化做 offset 模板, 非 1 不做 offset 模板
ret = HBI_ConnectDetector(m_pFpd, "192.168.10.40", 32897, "192.168.10.20", 32896, 0);
if (ret != 0) {
    printf("连接失败!\n");
    HBI_Destroy(m_pFpd);
    m_pFpd = NULL;
    return false;
}
// 设置触发模式和图像矫正使能
int _triggerMode = 3; // 1-软触发, 3-高压触发, 4-freeAED
IMAGE_CORRECT_ENABLE* pCorrect = new IMAGE_CORRECT_ENABLE;
if (pCorrect != NULL) {
    pcorrect->bFeedbackCfg = true;          //true-ECALLBACK_TYPE_ROM_UPLOAD
Event,false-ECALLBACK_TYPE_SET_CFG_OK Event
    pcorrect->ucOffsetCorrection = 0x02; //00-"Do nothing";01-"prepare Offset Correction";
02-"post Offset Correction";
    pcorrect->ucGainCorrection = 0x01; //00-"Do nothing";01-"Software Gain Correction";
02-"Hardware Gain Correction"
    pcorrect->ucDefectCorrection = 0x01; //00-"Do nothing";01-"Software Defect Correction";
02-"Software Defect Correction"
    pcorrect->ucDummyCorrection = 0x01; //00-"Do nothing";01-"Software Dummy Correction";
02-"Software Dummy Correction"
    ret=HBI_TriggerAndCorrectApplay(m_pFpd,_triggerMode, pCorrect);
    if (ret != 0) {
        printf("连接失败!\n");
    }
}
// 断开连接
//HBI_DisConnectDetector(m_pFpd));
// 回收资源(包括断开连接和资源释放)
HBI_Destroy(m_pFpd);
m_pFpd = NULL;
注意:

```

- 1》连接成功后，平板会自动反馈 ROM 参数。
- 2》HBI_Init 和 HBI_Destroy：连接和断开平板对应；
- 3》HBI_ConnectDetector 和 HBI_DisConnectDetector：初始化和释放设备对应；
- 4》HBI_SetSysParamCfg 和 HBI_GetSDKVerion：都是读取平板固件参数，HBI_SetSysParamCfg 是向固件发请求获取参数，异步函数；HBI_GetSDKVerion 是连接成功或设置成功后获取参数，同步函数。
- 5》HBI_ConnectDetector 连接平板，回调事件 ECALLBACK_TYPE_ROM_UPLOAD 反馈当前固件的参数，这里基本信息已固化好，用户可以直接使用
- 6》HBI_TriggerAndCorrectApplay，根据参数反馈 ECALLBACK_TYPE_ROM_UPLOAD 反馈当前固件的参数和 ECALLBACK_TYPE_SET_CFG_OK 事件确认成功，用户根据实际情况设置参数。
- 7》HBI_Destroy 释放资源，句柄为 NULL，如果直接关闭，调用 HBI_Destroy 即可，HBI_Destroy 中已包含 HBI_DisConnectDetector 的调用。

3、注册回调函数

#####

#回调函数及事件说明：

#pObj: 对象指针

#command: 事件 ID

#buff: 缓冲区地址

#len: 缓冲区长度

#nid: 图像数据为帧号

```
// 以下 theDemo 指针是自定义的全局指针，见 SDK 工程源码
CHB_SDK_DEMO2008D1g *theDemo = (CHB_SDK_DEMO2008D1g *)this;
// 回调函数
int CHB_SDK_DEMO2008D1g::handleCommandEvent(unsigned char command, void *buff, int len, int nId)
{
    int ret = 0;
    if ((command == ECALLBACK_TYPE_SINGLE_IMAGE) || (command == ECALLBACK_TYPE_MULTIPLE_IMAGE) ||
        (command == ECALLBACK_TYPE_ROM_UPLOAD) || (command == ECALLBACK_TYPE_RAM_UPLOAD))
    {
        if (buff == NULL || len == 0) {
            printf("注册回调函数参数异常!\n");
            return ret;
        }
    }

    //
    switch (command)
    {
        case ECALLBACK_TYPE_NET_ERR_MSG: // 平板状态：连接/断开/ready/busy
            break;
        case ECALLBACK_TYPE_ROM_UPLOAD: // 更新配置
            break;
        case ECALLBACK_TYPE_SET_CFG_OK: // 参数设置成功，不反馈全部参数
            break;
        case ECALLBACK_TYPE_SINGLE_IMAGE: // 单帧采集上图
        case ECALLBACK_TYPE_MULTIPLE_IMAGE: // 连续采集上图
            break;
        case ECALLBACK_TYPE_THREAD_EVENT: // 主线程监控
            break;
        case ECALLBACK_TYPE_ACQ_SUCCESS: // 连续采集保存成功
            break;
        case ECALLBACK_TYPE_GENERATE_TEMPLATE: // 生成模板结果信息
            break;
        case ECALLBACK_TYPE_FILE_NOTEXIST: // 反馈文件是否存在
```



```

        break;
    default:
        printf("ECALLBACK_TYPE_INVALID, command=%02x\n", command);
        break;
    }
    return ret;
}

```

4、制作模板

1> Offset 模板

```

int CTemplateTool::DoOffsetTemp()
{
    if (theDemo == NULL) {
        ::AfxMessageBox("err:theDemo is NULL!");
        return -1;
    }
    if (!theDemo->m_pFpd) {
        ::AfxMessageBox("warning:请初始化动态库!");
        return -1;
    }
    if (!theDemo->IsOpen) {
        ::AfxMessageBox("warning:请连接平板!");
        return -1;
    }
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(false);
    EnumIMAGE_ACQ_MODE enumTemplateType = OFFSET_TEMPLATE_MODE;
    int ret = HBI_ImmediateGenerateTemplate(theDemo->m_pFpd, enumTemplateType);
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(true);
    return ret;
}

```

2> Gain 模板

```

int CTemplateTool::DoGainTemp()
{
    if (theDemo == NULL) {
        ::AfxMessageBox("err:theDemo is NULL!");
        return -1;
    }
    if (!theDemo->m_pFpd) {
        ::AfxMessageBox("warning:请初始化动态库!");
        return -1;
    }
    if (!theDemo->IsOpen) {
        ::AfxMessageBox("warning:请连接平板!");
        return -1;
    }
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(false);
    EnumIMAGE_ACQ_MODE enumTemplateType = GAIN_TEMPLATE_MODE;
}

```

```

        int ret = HBI_ImmediateGenerateTemplate(theDemo->m_pFpd, enumTemplateType);
        GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(true);
        return ret;
    }

```

3> Defect 模板

// 第一组亮场

```

int CTemplateTool::DoDefectGroup1()
{
    if (theDemo == NULL) {
        ::AfxMessageBox("err:theDemo is NULL!");
        return -1;
    }
    if (!theDemo->m_pFpd) {
        ::AfxMessageBox("warning:请初始化动态库!");
        return -1;
    }
    if (!theDemo->IsOpen) {
        ::AfxMessageBox("warning:请连接平板!");
        return -1;
    }
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(false);
    EnumIMAGE_ACQ_MODE enumTemplateType = DEFECT_ACQ_GROUP1;
    int ret = HBI_ImmediateGenerateTemplate(theDemo->m_pFpd, enumTemplateType);
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(true);
    return ret;
}

```

// 第二组亮场

```

int CTemplateTool::DoDefectGroup2()
{
    if (theDemo == NULL) {
        ::AfxMessageBox("err:theDemo is NULL!");
        return -1;
    }
    if (!theDemo->m_pFpd) {
        ::AfxMessageBox("warning:请初始化动态库!");
        return -1;
    }
    if (!theDemo->IsOpen) {
        ::AfxMessageBox("warning:请连接平板!");
        return -1;
    }
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(false);
    EnumIMAGE_ACQ_MODE enumTemplateType = DEFECT_ACQ_GROUP2;
    int ret = HBI_ImmediateGenerateTemplate(theDemo->m_pFpd, enumTemplateType);
}

```

```

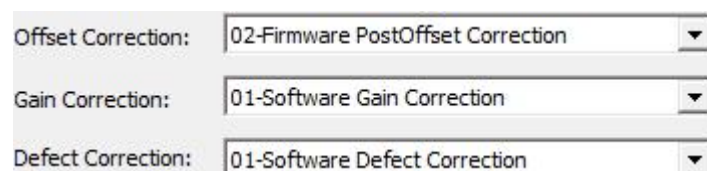
        GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(true);
        return ret;
    }
// 第三组亮场
int CTemplateTool::DoDefectGroup3()
{
    if (theDemo == NULL) {
        ::AfxMessageBox("err:theDemo is NULL!");
        return -1;
    }
    if (!theDemo->m_pFpd) {
        ::AfxMessageBox("warning:请初始化动态库!");
        return -1;
    }
    if (!theDemo->IsOpen) {
        ::AfxMessageBox("warning:请连接平板!");
        return -1;
    }
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(false);
    EnumIMAGE_ACQ_MODE enumTemplateType = DEFECT_ACQ_AND_TEMPLATE;
    int ret = HBI_ImmediateGenerateTemplate(theDemo->m_pFpd, enumTemplateType);
    GetDlgItem(IDC_BTN_TEMPLATE_GENERATE)->EnableWindow(true);
    return ret;
}

```

具体请参考 Demo 源码例子工程。

5、校正使能

医用或宠物工作站：固件 offset、软件 gain 和 defect 校正，效果比较好。



点料机：软件 offset、gain 和 defect 校正，效果比较好。



以点料机为例：

```

IMAGE_CORRECT_ENABLE *pcorrect = new IMAGE_CORRECT_ENABLE;
if (pcorrect != NULL) {
    #if 0
        pcorrect->bFeedbackCfg = false; // false-ECALLBACK_TYPE_SET_CFG_OK Event
    #else
        pcorrect->bFeedbackCfg = true; // true-ECALLBACK_TYPE_ROM_UPLOAD Event,

```

```

#endif
pcorrect->ucOffsetCorrection = 1;
pcorrect->ucGainCorrection   = 1;
pcorrect->ucDefectCorrection = 1;
pcorrect->ucDummyCorrection  = 0;
int ret = HBI_UpdateCorrectEnable(m_pFpd, pcorrect);
if (ret == 0)
    printf("OnBnClickedBtnCorrectEnable:\n\tHBI_UpdateCorrectEnable
success!\n");
else
    ::AfxMessageBox("HBI_QuickInitDllCfg failed!");
delete pcorrect;
pcorrect = NULL;
}
else {
    ::AfxMessageBox("malloc IMAGE_CORRECT_ENABLE failed!");
}
}

```

五、常见问题

依赖库：目前 SDK 使用了 Opencv249，将 Opencv249 的库文件拷贝系统库目录或者 exe 目录下即可，注意 32 位和 64 位库。

有些 Window 系统缺少 vs 库，请安装 vs 环境库。

结束语：由于时间匆忙，文档可能会存在个别问题，望见谅！