



# 红外热像仪 SDK 编程指南

V1.4 20210616

# 目录

- 1. SDK 简介.....6
- 2. 版本说明.....7
- 3. 调用流程.....8
- 4. 数据结构.....9
  - 4.1. 枚举.....9
    - 4.1.1. 设备类型枚举.....9
    - 4.1.2. 图像翻转类型枚举.....9
    - 4.1.3. 测温模型枚举.....9
    - 4.1.4. 增益模式枚举.....9
    - 4.1.5. 报警类型枚举.....10
    - 4.1.6. 网络透明通道命令枚举.....10
  - 4.2. 结构体.....10
    - 4.2.1. 设备信息结构体.....10
    - 4.2.2. 设备参数结构体.....11
    - 4.2.3. 点结构体.....11
    - 4.2.4. 矩形区域结构体.....11
    - 4.2.5. 温度点结构体.....12
    - 4.2.6. 温度信息结构体.....12
    - 4.2.7. 报警级别结构体.....12
    - 4.2.8. 连通区域结构体.....12

4.2.9.	线性调光结构体.....	13
4.2.10.	平台调光结构体 .....	13
4.2.11.	环境温度参数 .....	13
4.2.12.	回调数据扩展结构体.....	13
4.3.	回调函数.....	14
4.3.1.	设备搜索回调函数.....	14
4.3.2.	上传数据回调函数.....	14
4.3.3.	上传数据回调函数扩展.....	15
5.	函数说明.....	16
5.1.	设备发现.....	16
5.1.1.	开始搜索设备 .....	16
5.1.2.	停止搜索设备.....	16
5.1.3.	修改设备信息.....	17
5.1.4.	保存设备信息.....	17
5.2.	设备初始化与释放.....	18
5.2.1.	设备初始化.....	18
5.2.2.	设备释放.....	18
5.3.	设备打开与关闭.....	19
5.3.1.	打开设备.....	19
5.3.2.	打开设备扩展.....	19
5.3.3.	关闭设备.....	20

5.3.4.	设备重连.....	20
5.3.5.	获取设备运行标志.....	20
5.4.	图像配置.....	21
5.4.1.	设置调色板.....	21
5.4.2.	图像非均匀性校正.....	21
5.4.3.	快门校正.....	22
5.4.4.	图像聚焦.....	22
5.4.5.	设置图像亮度.....	22
5.4.6.	设置图像对比度.....	23
5.4.7.	设置图像增益模式.....	23
5.4.8.	设置平台调光参数.....	23
5.4.9.	Gamma 校正.....	24
5.4.10.	激活本底图像校正功能.....	24
5.4.11.	采集本底 .....	25
5.4.12.	启用/禁用黑体防遮挡算法.....	25
5.4.13.	透明通道 .....	25
5.5.	温度配置.....	26
5.5.1.	获取黑体基准温度.....	26
5.5.2.	获取黑体校正温度.....	26
5.5.3.	设置黑体校正温度.....	27
5.5.4.	设置黑体区域.....	27

5.5.5.	设置黑体校正模式.....	28
5.5.6.	设置屏蔽区域网格尺寸.....	28
5.5.7.	设置屏蔽区域.....	29
5.5.8.	设置温度曲线.....	29
5.5.9.	设置测温模式.....	30
5.5.10.	设置手动温度校正值.....	30
5.5.11.	设置距离温度校正值.....	31
5.5.12.	获取坐标点温度 .....	31
5.5.13.	获取矩形区域温度.....	31
5.5.14.	获取焦平面温度 .....	32
5.5.15.	启用/禁用环境温度矫正 .....	32
5.5.16.	获取环境温度参数.....	33
5.5.17.	获取原始测温数据.....	33
5.6.	报警配置.....	33
5.6.1.	设置报警等级.....	33
5.6.2.	设置报警最小像素值.....	34
5.7.	其他接口.....	34
5.7.1.	设置三点匹配校正矩阵.....	34
5.7.2.	获取设备软件版本号 .....	35
5.7.3.	设备升级.....	35
5.7.4.	开始模拟数据测试.....	36

5.7.5. 停止模拟数据测试.....36

## 1. SDK 简介

SDK 基于红外热像仪私有网络通信协议开发, 提供红外热像仪的搜索、配置、图像和温度数据获取、图像配置、报警配置、测温配置等功能。

SDK 文件说明如下:

Windows 版本	
文件	说明
XJTech.dll	SDK 主动态库
XIRStream.dll	SDK 视频流动态库
XJTech.lib	SDK 主静态库
XJTech.h	SDK 接口文件
XBaseDefine.h	SDK 数据接口定义文件
XJTechSDK.cs	C#接口文件
Linux 版本	
文件	说明
XJTech.so	SDK 主动态库
XIRStream.so	SDK 视频流动态库
XJTech.a	SDK 主静态库
XJTech.h	SDK 接口文件
XBaseDefine.h	SDK 数据接口定义文件

## 2. 版本说明

SDK 版本	更新说明
V1.0 20201122	初版 SDK 编程指南
V1.1 20201206	增加接口: 5.4.10 激活本底图像校正功能; 5.4.11 采集本底; 5.5.3 获取黑体校正温度;
V1.2 20210110	增加接口: 4.1.6 网络透明通道命令枚举; 5.4.12 启用/禁用黑体防遮挡算法; 5.4.13 透明通道; 5.5.11 设置距离温度校正值
V1.3 20210615	增加接口与数据结构: 4.2.11 环境温度参数; 4.2.12 回调数据扩展结构体; 4.3.3 上传数据回调函数扩展; 5.3.2 打开设备扩展; 5.5.15 启用/禁用环境温度矫正; 5.5.16 获取环境温度参数
V1.4 20210616	增加接口: 5.5.17 获取原始测温数据



### 3. 调用流程

SDK 调用主流程如图 1 所示，实线框为必须流程，虚线框为可选流程：

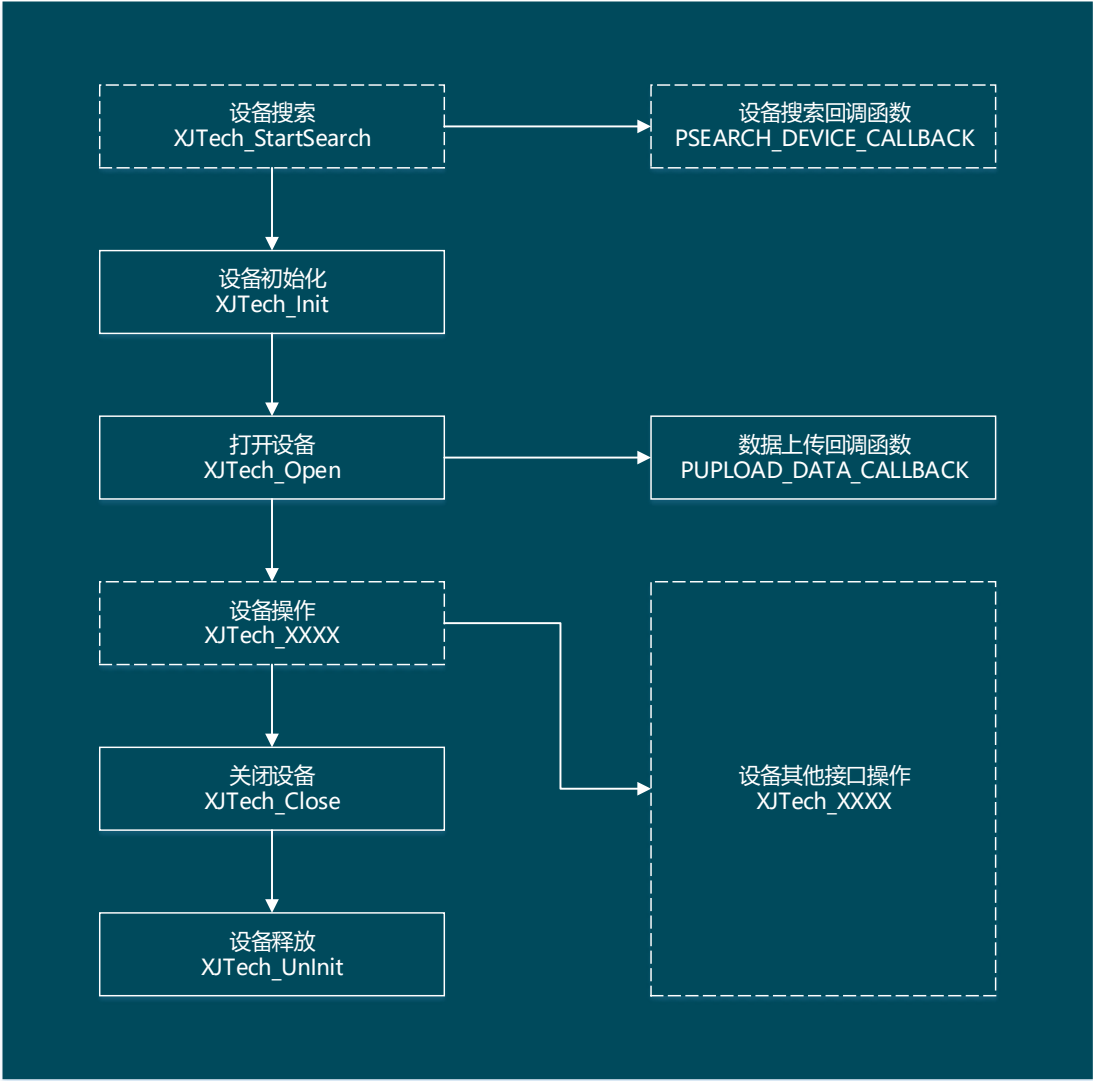


图 1 SDK 接口调用流程

## 4. 数据结构

### 4.1. 枚举

#### 4.1.1. 设备类型枚举

```
enum X_DEVICE_TYPE
{
    E_NONE = 0,    // 无
    E_UDP,         // UDP设备
    E_TCP_HISI,    // TCP_HISI设备（红外热像仪）
    E_RTSP,        // RTSP设备
    E_TCP_SAM,     // TCP_SAM设备
};
```

#### 4.1.2. 图像翻转类型枚举

```
enum X_FLIP_TYPE
{
    E_FLIP_NONE = 0,    // 不翻转
    E_FLIP_R90,         // 顺时针翻转90度
    E_FLIP_L90,         // 逆时针翻转90度
    E_FLIP_MAX          // 翻转最大值（哨兵）
};
```

#### 4.1.3. 测温模型枚举

```
enum X_MEASURE_MODE
{
    E_SURFACE = 0,    // 体表
    E_ARMPIT         // 腋下
};
```

#### 4.1.4. 增益模式枚举

```
enum X_GAIN_MODE
{
    E_MANUAL = 0,    // 手动
};
```

```
E_SEMI_AUTO,    // 半自动
E_AUTO          // 自动
};
```

#### 4.1.5. 报警类型枚举

```
enum X_ALARM_TYPE
{
    E_HIGH_ALARM = 0,    // 高温报警
    E_LOW_ALARM,         // 低温报警
    E_BOTH_ALARM         // 高低温报警
};
```

#### 4.1.6. 网络透明通道命令枚举

```
enum X_SERIAL_CMD
{
    // 无
    CMD_NONE = 0,
    // 快门补偿
    CMD_SHUTTER = 1,
    // 设置快门间隔时间
    CMD_SET_SHUTTER_INTERVAL = 2,
    // 保存快门间隔时间配置
    CMD_SAVE_SHUTTER_INTERVAL = 3,
    // 设置色带
    CMD_PALETTE = 4,
    // 增益调节
    CMD_GAIN_ADJUST = 5,
    // 自定义
    CMD_USER_DEFINE = 6,
    // 最大值
    CMD_MAX
};
```

## 4.2. 结构体

### 4.2.1. 设备信息结构体

```
struct X_DEVICE_INFO
```

```
{
    char szMac[20];    // 设备Mac地址
    char szIpAddr[16]; // 设备IP地址
    char szSubNet[16]; // 设备子网掩码
    char szGateWay[16]; // 设备网关
    int  iPort;        // 端口号
};
```

#### 4.2.2. 设备参数结构体

```
struct X_DEVICE_PARAM
{
    int          iFrameRate;    // 图像帧频
    int          iSrcWidth;     // 原始图像/温度宽度
    int          iSrcHeight;    // 原始图像/温度高度
    int          iDstWidth;     // 目标图像/温度宽度
    int          iDstHeight;    // 目标图像/温度高度
    int          iCutHorOffset; // 数据裁剪水平偏移量
    int          iCutVerOffset; // 数据裁剪垂直偏移量
    int          bCut;          // 数据裁剪标志
    X_ALARM_TYPE eAlarmType;    // 报警类型
    X_FLIP_TYPE  eFlipType;     // 数据翻转类型
};
```

#### 4.2.3. 点结构体

```
struct X_PT
{
    int iX; // X坐标
    int iY; // Y坐标
};
```

#### 4.2.4. 矩形区域结构体

```
struct X_RECT
{
    int iLeft;    // 左上角X坐标
    int iTop;     // 左上角Y坐标
    int iRight;   // 右下角X坐标
    int iBottom;  // 右下角Y坐标
};
```

#### 4.2.5. 温度点结构体

```
struct X_TEMPR_PT
{
    X_PT pt;      // 点结构体
    float fTempr; // 温度
    short shAd;   // AD值
};
```

#### 4.2.6. 温度信息结构体

```
struct X_TEMPR_INFO
{
    int iX;      // X坐标
    int iY;      // Y坐标
    int iTempr;  // 温度 (除以10为实际温度)
};
```

#### 4.2.7. 报警级别结构体

```
struct X_ALARM_LEVEL
{
    float fOne;   // 报警等级1温度
    float fTwo;   // 报警等级2温度
    float fThree; // 报警等级3温度
    float fFour;  // 报警等级4温度
};
```

#### 4.2.8. 连通区域结构体

```
struct X_ALARM_LINDED_AREA
{
    int iDstTempr;      // 最高或者最低温度
    int iAvgTempr;      // 平均温度
    short shAvgAd;      // 平均AD值
    X_PT pDstTemprPt;   // 最高温度对应的图像坐标
    X_PT pCentroidPt;   // 质心图像坐标
    int iAreaPixelCount; // 区域点个数
};
```

#### 4.2.9. 线性调光结构体

```
struct LinearDimmerParam
{
    X_GAIN_MODE eGainMode;           // 增益模式
    double dLowerDiscardRatio;        // 下抛点率 (默认值1%)
    double dUpperDiscardRatio;        // 上抛点率 (默认值1%)
    double dBrightness;               // 亮度
    double dContrast;                 // 对比度
};
```

#### 4.2.10. 平台调光结构体

```
struct PlatHistDimmerParam
{
    int      iPlatThresholdValue;      // 平台阈值 (范围1 ~ 200, 默认值100)
    int      iMappingMidValue;         // 映射中间值, 调整亮度 (范围0 ~ 255, 默认值128)
    double   dLowerDiscardRatio;       // 下抛点率 (默认值1%)
    double   dUpperDiscardRatio;       // 上抛点率 (默认值1%)
    int      iDynamicRangeCoef;        // 动态范围系数
    int      iMappingRange;            // 映射范围
};
```

#### 4.2.11. 环境温度参数

```
struct X_ENV_PARAM
{
    float      fCurrEnvTempr;          // 当前环境温度
    float      fBaseEnvTempr;          // 基准环境温度
    float      fEnvCorreStep;          // 环境温度矫正系数
    unsigned char chEnvCorreSwitch;    // 下位机环温矫正开关 (0代表关, 1代表开)
};
```

#### 4.2.12. 回调数据扩展结构体

```
struct X_CALLBACK_DATA
{
    unsigned char*      pImage;          // 红外图像数据 (RGB)
    float*              pSrcTemperature; // 原始温度数据 (全图)
    float*              pOptTemperature; // 优化温度数据 (全图)
};
```

X_TEMPR_INFO*	pMaxTemprInfo;	// 高温信息
X_TEMPR_INFO*	pMinTemprInfo;	// 低温信息
X_ALARM_LINDED_AREA*	pAlarmLinkedArea;	// 连通区域
int	iAlarmAreaCount;	// 连通区域个数
unsigned char*	pAlarmMask;	// 报警模板 (全图)
void*	pUserData;	// 用户数据
};		

### 4.3. 回调函数

#### 4.3.1. 设备搜索回调函数

<ul style="list-style-type: none"><li>● 函数: typedef void (STD_CALL</li></ul>
<div><div>*PSEARCH_DEVICE_CALLBACK)(X_DEVICE_INFO* pDeviceInfo, void</div><div>*pUserData);</div></div>
<ul style="list-style-type: none"><li>● 参数:</li></ul>
<div><div>➤ pDeviceInfo: 设备信息结构体</div><div>➤ pUserData: 用户数据</div></div>
<ul style="list-style-type: none"><li>● 返回值: 无</li></ul>
<ul style="list-style-type: none"><li>● 说明: 用户在调用 XJTech_StartSearch 函数时传入该回调函数, 搜索到设备后会触发该回调, 用户在该回调函数中处理设备信息</li></ul>

#### 4.3.2. 上传数据回调函数

<ul style="list-style-type: none"><li>● 函数: typedef void (STD_CALL *PUPLOAD_DATA_CALLBACK)(unsigned char*</li></ul>
<div><div>pImage, X_TEMPR_INFO* pMaxTemprInfo, X_TEMPR_INFO* pMinTemprInfo,</div><div>X_ALARM_LINDED_AREA* pAlarmLinkedArea, int iAlarmAreaCount, unsigned</div></div>

```
char* pAlarmMask, void* pUserData);
```

- 参数：
  - pImage: 图像数据
  - pMaxTemprInfo: 全图最高温结构体（当报警类型为高温或者高低温时，该参数有效，否则为 NULL）
  - pMinTemprInfo: 全图最低温结构体（当报警类型为低温或者高低温时，该参数有效，否则为 NULL）
  - pAlarmLinkedArea: 连通区域数组
  - iAlarmAreaCount: 连通区域个数
  - pAlarmMask: 报警图像模板（用户可依据该模板进行图像渲染）
  - pUserData: 用户数据
- 返回值: 无
- 说明: 用户在调用 XJTech\_Open 函数时传入该回调函数, 有红外数据上传会触发该回调, 用户在该回调函数中处理图像和温度信息

#### 4.3.3. 上传数据回调函数扩展

```
函数: typedef void (STD_CALL *PUPLOAD_DATA_CALLBACK_EX)
(X_CALLBACK_DATA* pCallbackData);
```

- 参数：
  - pCallbackData: 回调数据结构体
- 返回值: 无



- 说明：用户在调用 XJTech\_OpenEx 函数时传入该回调函数，有红外数据上传会触发该回调，用户在该回调函数中处理图像和温度信息

## 5. 函数说明

### 5.1. 设备发现

#### 5.1.1. 开始搜索设备

- 函数：int XJTech\_StartSearch(X\_DEVICE\_TYPE eDeviceType, PSEARCH\_DEVICE\_CALLBACK pSearchDeviceCallback, void\* pUserData)
- 参数：
  - eDeviceType：设备类型（红外设备必须为 E\_TCP\_HISI）
  - pSearchDeviceCallback：设备搜索回调函数
  - pUserData：用户数据
- 返回值：< 0 代表失败，> 0 代表成功
- 说明：调用该函数后，设备信息通过 pSearchDeviceCallback 回调函数推送给调用者

#### 5.1.2. 停止搜索设备

- 函数：void XJTech\_StopSearch(X\_DEVICE\_TYPE eDeviceType)
- 参数：
  - eDeviceType：设备类型（红外设备必须为 E\_TCP\_HISI）

- 返回值：无
- 说明：该函目前无实际作用，可忽略

### 5.1.3. 修改设备信息

- 函数：int XJTech\_ModifyDeviceInfo(X\_DEVICE\_TYPE eDeviceType, X\_DEVICE\_INFO\* pOldDeviceInfo, X\_DEVICE\_INFO\* pNewDeviceInfo, void\* pUserData)
- 参数：
  - eDeviceType：设备类型（红外设备必须为 E\_TCP\_HISI）
  - pOldDeviceInfo：旧设备信息
  - pNewDeviceInfo：新设备信息
  - pUserData：用户数据
- 返回值：< 0 代表失败，> 0 代表成功
- 说明：调用该函数必须在设备处于连接状态，即必须能都依据旧设备信息打开设备后才能使用新设备信息更新设备配置

### 5.1.4. 保存设备信息

- 函数：int XJTech\_SaveDeviceInfo(X\_DEVICE\_TYPE eDeviceType, X\_DEVICE\_INFO\* pDeviceInfo, void\* pUserData)
- 参数：
  - eDeviceType：设备类型（红外设备必须为 E\_TCP\_HISI）

- pDeviceInfo: 设备信息
- pUserData: 用户数据
- 返回值: < 0 代表失败, > 0 代表成功
- 说明: 该函目前无实际作用, 可忽略

## 5.2. 设备初始化与释放

### 5.2.1. 设备初始化

- 函数: `int XJTech_Init(const X_DEVICE_TYPE eDeviceType, const X_DEVICE_PARAM &pDeviceParam)`
- 参数:
  - eDeviceType: 设备类型 (红外设备必须为 E\_TCP\_HISI)
  - pDeviceParam: 设备参数
- 返回值: 设备 ID
- 说明: 设备初始化, 该函数调用后会生成一个对应设备对象, 要使用设备功能, 该函数必须第一个调用

### 5.2.2. 设备释放

- 函数: `void XJTech_UnInit(int iUserId)`
- 参数:
  - iUserId: 设备 ID
- 返回值: 无

- 说明：设备资源释放，设备使用关闭后，需调用该函数释放相关资源，设备使用完毕后最后调用该函数

### 5.3. 设备打开与关闭

#### 5.3.1. 打开设备

- 函数：int XJTech\_Open(int iUserId, X\_DEVICE\_INFO\* pDeviceInfo, PUPLOAD\_DATA\_CALLBACK pUploadDataCallback, void\* pUserData)
- 参数：
  - iUserId：设备 ID
  - pDeviceInfo：设备信息
  - pUploadDataCallback：图像与温度数据回调函数
  - pUserData：用户数据
- 返回值：< 0 代表失败，> 0 代表成功
- 说明：打开设备

#### 5.3.2. 打开设备扩展

- 函数：int XJTech\_OpenEx(int iUserId, X\_DEVICE\_INFO\* pDeviceInfo, PUPLOAD\_DATA\_CALLBACKEX pUploadDataCallbackEx, void\* pUserData)
- 参数：
  - iUserId：设备 ID
  - pDeviceInfo：设备信息

- pUploadDataCallbackEx: 图像与温度数据回调函数
- pUserData: 用户数据
- 返回值: < 0 代表失败, > 0 代表成功
- 说明: 打开设备

### 5.3.3. 关闭设备

- 函数: void XJTech\_Close(int iUserId)
- 参数:
  - iUserId: 设备 ID
- 返回值: 无
- 说明: 关闭设备, 需与 XJTech\_Open 函数配套使用

### 5.3.4. 设备重连

- 函数: int XJTech\_Reconnect(int iUserId, X\_DEVICE\_INFO\* pDeviceInfo)
- 参数:
    - iUserId: 设备 ID
    - pDeviceInfo: 设备信息
  - 返回值: < 0 代表失败, > 0 代表成功
  - 说明: SDK 内部处理设备重连逻辑, 外部调用无效, 永远返回-1

### 5.3.5. 获取设备运行标志

函数: bool XJTech\_IsRunning (int iUserId)

- 参数：
  - iUserId: 设备 ID
- 返回值: true 代表正在运行, false 代表处于非运行状态
- 说明: 获取设备运行标志

## 5.4. 图像配置

### 5.4.1. 设置调色板

- 函数: void XJTech\_ChangePalette(int iUserId, int iPaletteIndex)
- 参数：
  - iUserId: 设备 ID
  - iPaletteIndex: 调色板索引 (范围: 0~27)
- 返回值: 无
- 说明: 设置调色板

### 5.4.2. 图像非均匀性校正

- 函数: void XJTech\_DoFFC(int iUserId)
- 参数：
  - iUserId: 设备 ID
- 返回值: 无
- 说明: 该函数目前无实际作用, 可忽略

### 5.4.3. 快门校正

- 函数: void XJTech\_Shutter(int iUserId)
- 参数:
  - iUserId: 设备 ID
- 返回值: 无
- 说明: 快门校正, 设备会定时自动打快门, 该函数提供手动打快门功能

### 5.4.4. 图像聚焦

- 函数: void XJTech\_Focus (int iUserId)
- 参数:
  - iUserId: 设备 ID
- 返回值: 无
- 说明: 该函目前无实际作用, 可忽略

### 5.4.5. 设置图像亮度

- 函数: void XJTech\_SetBrightness(int iUserId, int iBrightness)
- 参数:
  - iUserId: 设备 ID
  - iBrightness: 亮度值 (范围: 0~255)
- 返回值: 无
- 说明: 设置图像亮度

#### 5.4.6. 设置图像对比度

- 函数: void XJTech\_SetContrast(int iUserId, int iContrast)
- 参数:
  - iUserId: 设备 ID
  - iContrast: 对比度值 (范围: 0~255)
- 返回值: 无
- 说明: 设置图像对比度

#### 5.4.7. 设置图像增益模式

- 函数: void XJTech\_SetGainMode(int iUserId, bool bGainMode)
- 参数:
  - iUserId: 设备 ID
  - bGainMode: true 代表自动增益, false 代表手动增益
- 返回值: 无
- 说明: 默认为自动增益模式, 自动增益模式 SDK 自动计算图像亮度和对比度, 手动增益模式下, SDK 使用用户设置的图像亮度和对比度

#### 5.4.8. 设置平台调光参数

- 函数: void XJTech\_SetPlatDimmerParam(int iUserId, PlatHistDimmerParam\* pParam)
- 参数:



- iUserId: 设备 ID
- pParam: 平台调光参数
- 返回值: 无
- 说明: 该函数提供图像高级设置功能, SDK 内部包含默认的平台调光参数, 用户如需调节该参数, 请在专业人士指导下进行

#### 5.4.9. Gamma 校正

- 函数: void XJTech\_GammaCorrection(int iUserId, int iGammaTableIndex)
- 参数:
  - iUserId: 设备 ID
  - iGammaTableIndex: Gamma 表索引 (范围: 0~22)
- 返回值: 无
- 说明: 该函数提供图像高级设置功能, SDK 内部包含默认的 Gamma 表索引, 用户如需调节该参数, 请在专业人士指导下进行

#### 5.4.10. 激活本底图像校正功能

- 函数: void XJTech\_EnableBackgroundCorrect(int iUserId, bool bEnable)
- 参数:
  - iUserId: 设备 ID
  - bEnable: 激活标志
- 返回值: 无

- 说明：激活本底图像校正功能

#### 5.4.11. 采集本底

- 函数：void XJTech\_CaptureBackground (int iUserId)
- 参数：
  - iUserId：设备 ID
- 返回值：无
- 说明：当本底图像校正功能开启时，调用该函数才有效，调用该函数前，需使用均匀物体遮挡热像仪镜头，再调用该函数采集本底图像。该函数提供图像高级设置功能，SDK 内部包含默认的 Gamma 表索引，用户如需使用该功能，请在专业人士指导下进行

#### 5.4.12. 启用/禁用黑体防遮挡算法

- 函数：void XJTech\_EnableBlackBodyAvoidCover(int iUserId, bool bEnable)
- 参数：
  - iUserId：设备 ID
  - bEnable：true 代表启用，false 代表禁用
- 返回值：无
- 说明：在黑体被遮挡情况下，保证测温准确度，请在专业人士指导下进行设置

#### 5.4.13. 透明通道

- 函数：void XJTech\_SerialSend(int iUserId, X\_SERIAL\_CMD emCmdType,

char \*pData, int iSize)

- 参数：
  - iUserId: 设备 ID
  - emCmdType: 4.1.6 网络透明通道命令枚举
  - pData: 协议数据
  - iSize: 协议长度
- 返回值: 无
- 说明: 网络透明通道, 请在专业人士指导下进行调用

## 5.5. 温度配置

### 5.5.1. 获取黑体基准温度

- 函数: float XJTech\_GetBlackBodyBaseTempr(int iUserId)
- 参数：
  - iUserId: 设备 ID
- 返回值: 黑体基准温度 (默认为 37.0°C)
- 说明: 获取黑体基准温度

### 5.5.2. 获取黑体校正温度

- 函数: float XJTech\_GetBlackBodyCorrTempr(int iUserId)
- 参数：
  - iUserId: 设备 ID

- 返回值：黑体校正温度
- 说明：获取黑体校正温度

### 5.5.3. 设置黑体校正温度

- 函数：bool XJTech\_SetBlackBodyBaseTempr(int iUserId, float fTempr)
- 参数：
  - iUserId：设备 ID
  - fTempr：黑体基准温度
- 返回值：true 代表设置成功，false 代表设置失败
- 说明：默认黑体基准温度为 37.0℃，如果调整了黑体温度，需调用该接口重置黑体基准温度

### 5.5.4. 设置黑体区域

- 函数：void XJTech\_SetBlackBodyRegion(int iUserId, unsigned char\* pRegion, int iSize)
- 参数：
  - iUserId：设备 ID
  - pRegion：代表黑体区域的左上角和右下角坐标，总共 4 个点，依次代表左上角 X 坐标，左上角 Y 坐标，右下角 X 坐标，右下角 Y 坐标，每个坐标值占 2 个字节，总共 8 个字节，将 8 个字节转换成 unsigned char 数组传入（**注意：该坐标系基于实际图像区域**）

- iSize: 固定值 8
- 返回值: 无
- 说明: 设置黑体区域

#### 5.5.5. 设置黑体校正模式

- 函数: void XJTech\_StartBdCorrectMode(int iUserId, bool bBdCorrect)
- 参数:
  - iUserId: 设备 ID
  - bBdCorrect: true 代表使用黑体进行温度校正, false 代表不使用黑体进行温度校正
- 返回值: 无
- 说明: SDK 内部默认为黑体校正模式

#### 5.5.6. 设置屏蔽区域网格尺寸

- 函数: void XJTech\_SetShieldGridSize(int iUserId, int iHorGridNum, int iVerGridNum)
- 参数:
  - iUserId: 设备 ID
  - iHorGridNum: 横向网格个数 (默认为 45)
  - iVerGridNum: 纵向网格个数 (默认为 40)
- 返回值: 无

- 说明：设置屏蔽区域前需调用该函数设置屏蔽区域的网格个数

#### 5.5.7. 设置屏蔽区域

- 函数：void XJTech\_SetShieldArea(int iUserId, unsigned char\* pArea, int iSize)
- 参数：
  - iUserId：设备 ID
  - pArea：代表屏蔽区域网格数组，pArea 数组中 1 代表是屏蔽区域，0 代表非屏蔽区域，该数组大小应为上节中 iHorGridNum \* iVerGridNum (**注意：该坐标系基于实际图像区域**)
  - iSize：pArea 数组大小
- 返回值：无
- 说明：设置屏蔽区域

#### 5.5.8. 设置温度曲线

- 函数：void XJTech\_SetTemprCurve(int iUserId, unsigned char\* pCurve, int iSize)
- 参数：
  - iUserId：设备 ID
  - pCurve：代表温度曲线数组，该数组包含 9 组温度点，每组包含两个温度点，分别代表原始温度和映射温度，总共 18 个温度点，每个温度点为 float 值，将 18 个 float 值转换为 unsigned char 数组传入

- iSize: 固定值 18 \* sizeof(float)
- 返回值: 无
- 说明: 设置温度曲线, SDK 内部包含默认温度曲线, 如需调整温度曲线, 需调用该接口, 在专业人士指导下进行

#### 5.5.9. 设置测温模式

- 函数: void XJTech\_SetMeasureMode(int iUserId, X\_MEASURE\_MODE eMode)
- 参数:
  - iUserId: 设备 ID
  - eMode: E\_SURFACE 代表体表测温模式, E\_ARMPIT 代表腋下测温模式, 如果设置为腋下测温模式, SDK 将使用温度曲线将体表温度映射为腋下温度
- 返回值: 无
- 说明: 设置测温模式

#### 5.5.10. 设置手动温度校正

- 函数: void XJTech\_SetManualTemprCorrect(int iUserId, float fCorrect)
- 参数:
  - iUserId: 设备 ID
  - fCorrect: 温度校正, 可以为正数或者为负数
- 返回值: 无
- 说明: 设置该值后, 所测腋下温度将加上该值校正

### 5.5.11. 设置距离温度校正

- 函数: void XJTech\_SetDistanceTemprCorrect (int iUserId, float fCorrect)
- 参数:
  - iUserId: 设备 ID
  - fCorrect: 温度校正, 可以为正数或者为负数
- 返回值: 无
- 说明: 设置该值后, 所测体表温度将加上该值校正

### 5.5.12. 获取坐标点温度

- 函数: float XJTech\_GetTemprByPoint(int iUserId, int iX, int iY)
- 参数:
  - iUserId: 设备 ID
  - iX: 坐标点 X 值 (基于真实图像坐标系)
  - iY: 坐标点 Y 值 (基于真实图像坐标系)
- 返回值: 坐标点温度值
- 说明: 获取坐标点温度值

### 5.5.13. 获取矩形区域温度

- 函数: void XJTech\_GetRectangleTempr(int iUserId, X\_RECT\* pRect, X\_TEMPR\_INFO\* pTemprInfo, float fRatio = 1.0F)
- 参数:



- iUserId: 设备 ID
- pRect: 矩形区域
- pTemprInfo: 矩形温度信息 (输出值)
- fRatio: 缩放率 (默认不缩放)
- 返回值: 无
- 说明: 获取矩形区域温度

#### 5.5.14. 获取焦平面温度

- 函数: int XJTech\_GetFPATempr(int iUserId)
- 参数:
  - iUserId: 设备 ID
- 返回值: 焦平面温度 (温度值以整数传出, 实际温度需除以 10.0)
- 说明: 该接口暂时无效

#### 5.5.15. 启用/禁用环境温度矫正

- 函数: void XJTech\_EnableEnvTemprCorrect (int iUserId, bool bEnable)
- 参数:
  - iUserId: 设备 ID
  - bEnable: 启用/禁用标志
- 返回值: 无)
- 说明: 依据环境温度自动矫正测温精度

### 5.5.16. 获取环境温度参数

- 函数: void XJTech\_GetEnvParam(int iUserId, X\_ENV\_PARAM\* pEnvParam)
- 参数:
  - iUserId: 设备 ID
  - pEnvParam: 环境温度参数结构体指针
- 返回值: 无
- 说明: 获取环境温度参数

### 5.5.17. 获取原始测温数据

- 函数: void XJTech\_GetSrcTemperatureData(int iUserId, short\* pData)
- 参数:
  - iUserId: 设备 ID
  - pData: 用户传入的数据指针, 数据大小为 “4.2.2 设备参数结构体” 中  
iDstWidth \* iDstHeight
- 返回值: 无
- 说明: pData 不能为空, 原始测温数据会拷贝至 pData

## 5.6. 报警配置

### 5.6.1. 设置报警等级

- 函数: void XJTech\_SetAlarmLevel(int iUserId, X\_ALARM\_LEVEL pAlarmLevel)
- 参数:

- iUserId: 设备 ID
- pAlarmLevel: 报警温度段结构体
- 返回值: 无
- 说明: 设置报警等级

### 5.6.2. 设置报警最小像素值

- 函数: void XJTech\_SetAlarmMinPixelsValue(int iUserId, int iAlarmMinPixelsValue)
- 参数:
  - iUserId: 设备 ID
  - iAlarmMinPixelsValue: 报警最小像素个数
- 返回值: 无
- 说明: 报警区域大于 iAlarmMinPixelsValue 值才会触发报警

## 5.7. 其他接口

### 5.7.1. 设置三点匹配校正矩阵

- 函数: bool XJTech\_GetTransformMatrix(X\_PT irPointF[3], X\_PT vlPointF[3], double TranformMatrix[3][3])
- 参数:
  - irPointF[3]: 红外图像三个坐标点数组
  - vlPointF[3]: 可见光图像三个坐标点数组

- TranformMatrix[3][3]: 匹配矩阵 (输出参数)
- 返回值: true 代表成功, false 代表失败
- 说明: 该函数用于配准红外图像和可见光图像坐标系

### 5.7.2. 获取设备软件版本号

- 函数: int XJTech\_GetDeviceSoftwareVersion(int iUserId, unsigned int\* uiVersion, unsigned int\* uiDate)
- 参数:
  - iUserId: 设备 ID
  - uiVersion: 软件版本, 可直接转换为字符串使用 (输出参数)
  - uiDate: 软件日期, 可直接转换为字符串使用 (输出参数)
- 返回值: < 0 代表失败, > 0 代表成功
- 说明: 获取设备软件版本号

### 5.7.3. 设备升级

- 函数: int XJTech\_UpdateProgram(int iUserId, char\* strFileName, unsigned char ucType, unsigned char ucIsReboot)
- 参数:
  - iUserId: 设备 ID
  - strFileName: 升级文件路径
  - ucType: 升级类型

- uclsReboot: 重启标志
- 返回值: < 0 代表失败, > 0 代表成功
- 说明: 请在专业人士指导下对设备进行升级

#### 5.7.4. 开始模拟数据测试

- 函数: void XJTech\_StartSimDataTest(int iUserId, PUPLOAD\_DATA\_CALLBACK pUploadDataCallback)
- 参数:
  - iUserId: 设备 ID
  - pUploadDataCallback: 图像与温度数据回调函数
- 返回值: 无
- 说明: 该版本 SDK 不提供该功能

#### 5.7.5. 停止模拟数据测试

- 函数: void XJTech\_StopSimDataTest(int iUserId)
- 参数:
  - iUserId: 设备 ID
- 返回值: 无
- 说明: 该版本 SDK 不提供该功能