

網絡測溫機芯 SDK 用戶使用手冊

目錄

1. 概述	4
2. 函數詳解	4
2.1 SDK 初始化	4
2.2 SDK 釋放	4
2.3 設備搜索	4
2.4 設備登入	5
2.5 設備登出	6
2.6 打開數據流	6
2.7 關閉數據流	7
2.8 獲取設備類型	7
2.9 查詢 SDK 支持功能	7
2.10 獲取設備信息	10
2.11 快門	11
2.12 拍照	13
2.13 Y16 錄像	16
2.14 AVI 錄像	16
2.15 設置色帶	17
2.16 獲取色帶數據	17
2.17 設置 Y16 幀率	18
2.18 調焦	19
2.19 獲取 SDK 版本信息	21
2.20 修改機芯設備 IP	22
2.21 獲取測溫參數	23
2.22 設置測溫參數	26
2.23 保存測溫參數	30
2.24 獲取實時點溫	30
2.25 獲取實時全圖溫	31
2.26 獲取實時矩形溫	33
2.27 獲取圖片溫	34
2.28 添加點線面分析對象	36
2.29 更新點線面分析對象	40
2.30 獲取點線面分析對象	41
2.31 刪除點線面分析對象	42
2.32 同步點線面分析對象	43
2.33 添加矩形分析對象	43
2.34 視頻流疊加字符串	47
2.35 電子變焦	49
2.36 調光模式	50
2.37 亮度對比度調光	51
2.38 手動溫度調光	52
2.39 機芯設備溫度報警	53
2.40 機芯設備工作狀態	55
2.41 端口號獲取與設置	56

3. 其它	58
3.1 計算機配置性能建議	58
3.2 SDK 使用注意事項	58
4. SDK 更新	58
5. SDK 版本號	61

1. 概述

本 SDK 在 Windows7 系統下開發, 使用 VS2015 開發工具。SDK 適用於 Windows 系統下調用。
本 SDK 用於測溫機芯系列。使用本 SDK 可以從設備獲取 H264 碼流、RGB 數據、溫度數據、支持拍照、錄像、IP 可配置、調焦以及從圖片獲取任意點溫度、全圖溫度和區域溫度信息等功能。
本 SDK 使用 IPv4 協議。

2. 函數詳解

2.1 SDK 初始化

Int32_t GD_MTC_SDK_Init();

功能:

SDK 初始化.此 SDK 中大部分接口調用時要求 SDK 已經初始化, 用戶在程序初始化時調用此函數初始化 SDK.

參數:

無.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.2 SDK 釋放

Int32_t GD_MTC_SDK_Release();

功能:

SDK 釋放.用戶在程序退出前調用此函數釋放 SDK.

參數:

無.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.3 設備搜索

Int32_t GD_MTC_SDK_StartSearchDevice(PROCESS_SEARCH_DEVICE_CB pSearchDeviceCallBack, int *pUser);

功能:
在網絡中搜索機芯設備.

參數:
pSearchDeviceCallback:
搜索結果回調函數. 具體參照 **PROCESS_SEARCH_DEVICE_CB** 說明.
pUser:
用戶指針.

返回值:
NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int (CALLBACK* PROCESS_SEARCH_DEVICE_CB)(const GD_MTC_IRDeviceInfo * pDeviceInfo, int  
nDeviceInfoSize, void *pUser);
```

功能:
搜索機芯設備回調函數.

參數:
pDeviceInfo:
搜索到的設備信息的地址.
nDeviceInfoSize:
搜索到的設備信息的條數.
pUser:
用戶指針.

返回值:
返回 1.

2.4 設備登入

```
Int32_t GD_MTC_SDK_Login(GD_MTC_IRDeviceInfo DeviceInfo, int *pUser);
```

功能:
設備登入.獲取數據流前需要成功登入.

參數:
DeviceInfo:
設備信息, 只需向此結構中填入 **IpAddress** 字段和 **DevType** 字段再調用此函數即可.當傳入的表示設備類型的 **DevType** 字段的值為 **DEVICE_TYPE_UNKNOWN** 時, 此函數實現中會自動識別設備類型, 否則函數認為傳入的設備類型就是將要連接的設備類型.可通過 **GD_MTC_SDK_GetDeviceType** 接口在登入前獲取設備類型.
pUser:
傳 0 值即可.

返回值:

登入 ID 值. 非 ERROR_LOGIN_FAIL 值表示調用成功, ERROR_LOGIN_FAIL 值表示調用失敗.

2.5 設備登出

Int32_t GD_MTC_SDK_Logout(const int iLoginID);

功能:

設備登出.

參數:

iLoginID:

調用設備登入函數成功時得到的登入 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.6 打開數據流

Int32_t GD_MTC_SDK_OpenStream(const int iLoginID, GD_MTC_IRLinkInfo *pIRLinkInfo, int iPaletteIndex, int iFps, int *pUser);

功能:

打開數據流.獲取測溫及圖像數據前需要打開數據流.

參數:

iLoginID:

調用設備登入函數成功時得到的登入 ID.

pIRLinkInfo:

連接信息對象地址.在此參數傳入對應的數據回調函數及獲取數據流的方式.

iPaletteIndex:

圖像的初始色帶.值的範圍為 0-7.

iFps:

Y16 數據的初始幀率.值的範圍為 1-25.

pUser:

用戶指針.傳入 0 值即可.

返回值:

數據流 ID. 非 ERROR_OPENSTREAM_FAIL 值表示調用成功, ERROR_OPENSTREAM_FAIL 值表示調用失敗.

2.7 關閉數據流

Int32_t GD_MTC_SDK_CloseStream(const int iSessionID);

功能:

關閉數據流.

參數:

iSessionID:

調用打開數據流函數成功時得到的數據流 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.8 獲取設備類型

```
Bool GD_MTC_SDK_GetDeviceType(const char * strIpAddress, GD_MTC_IRDeviceType & nDeviceType);
```

功能:

通過 IP 地址獲取設備類型.

參數:

strIpAddress:

機芯設備的 IP 地址.

nDeviceType:

獲取到的機芯的設備類型.

返回值:

如果成功獲取設備類型返回 true, 否則返回 false.

2.9 查詢 SDK 支持功能

```
Int32_t GD_MTC_SDK_SupportFunction(GD_MTC_IRDeviceType nDeviceType, PULL_DATA_TYPE nPullDataType, DEVICE_FUNCTION_SUPPORT nFunctionType);
```

功能:

查詢 SDK 對特定機型數據流模式是否支持及對特定機型的指定 SDK 功能接口是否支持.

例如, 以下函數調用分別查詢 OR2 設備是否支持 Y8 數據流, Y16 數據流及 Y8+Y16 數據流模式:

```
GD_MTC_SDK_SupportFunction(OR2, PULL_DATA_TYPE_H264_ONLY, DFS_Y8)
```

```
GD_MTC_SDK_SupportFunction(OR2, PULL_DATA_TYPE_Y16_ONLY, DFS_Y16)
```

```
GD_MTC_SDK_SupportFunction(OR2, PULL_DATA_TYPE_H264_AND_Y16, DFS_Y8_AND_Y16)
```

以下函數調用分別查詢 NET 機型在 Y8 模式是否支持 Y16 錄像, NET800x600 機型在 Y16 模式是否支持自動調焦:

```
GD_MTC_SDK_SupportFunction(NET, PULL_DATA_TYPE_H264_ONLY, DFS_Y16_RECORD)
```

```
GD_MTC_SDK_SupportFunction(NET800x600, PULL_DATA_TYPE_Y16_ONLY, DFS_AUTO_FOCUS)
```

參數:

nDeviceType:

設備型號.

nPullDataType:

獲取數據流類型.

nFunctionType:

功能項.

返回值:

1 表示支持此項功能, 0 表示不支持.

說明:

以下是 SDK 各函數對應的功能支持枚舉項

枚舉項 函數名

DFS_Y8 GD_MTC_SDK_OpenStream

DFS_Y16 GD_MTC_SDK_OpenStream

DFS_Y8_AND_Y16 GD_MTC_SDK_OpenStream

DFS_SHUTTER GD_MTC_SDK_Shutter

DFS_SHUTTER_CLOSE GD_MTC_SDK_ShutterClose

DFS_SHUTTER_OPEN GD_MTC_SDK_ShutterOpen

DFS_SNAP GD_MTC_SDK_SnapPicture

DFS_SNAP_NO_CB GD_MTC_SDK_SnapPictureNoCb

DFS_SNAP_WITHOUT_TEMP_NO_CB GD_MTC_SDK_SnapPictureWithoutTempNoCb

DFS_SNAP_WITHOUT_IMG_NO_CB GD_MTC_SDK_SnapPictureWithoutImgNoCb

DFS_SNAP_UNIFORM GD_MTC_SDK_SnapPictureUniform

DFS_AVI_RECORD GD_MTC_SDK_StartRecordAvi

GD_MTC_SDK_StopRecordAvi

DFS_Y16_RECORD GD_MTC_SDK_StartRecord

GD_MTC_SDK_StopRecord

DFS_SET_PALETTE GD_MTC_SDK_SetPalette

DFS_GET_PALETTE_DATA_SIZE GD_MTC_SDK_GetPaletteDataSize

DFS_GET_PALETTE_DATA GD_MTC_SDK_GetPaletteData

DFS_SET_FRAME_RATE GD_MTC_SDK_setY16FramePerSecond

DFS_NEAR_FOCUS GD_MTC_SDK_FocusNear

GD_MTC_SDK_FocusStop

DFS_FAR_FOCUS GD_MTC_SDK_FocusFar

GD_MTC_SDK_FocusStop

DFS_AUTO_FOCUS GD_MTC_SDK_FocusAuto

DFS_NEAR_FOCUS_FINE GD_MTC_SDK_FocusNearFine

DFS_FAR_FOCUS_FINE GD_MTC_SDK_FocusFarFine

DFS_GET_MOTOR_POS GD_MTC_SDK_getMotorPostition

DFS_SET_MOTOR_POS GD_MTC_SDK_setMotorPostition

DFS_SET_UART_CHECK_ADDRESS GD_MTC_SDK_SetUartCheckAddress

DFS_SET_UART_RATE GD_MTC_SDK_SetUartRate

DFS_DISABLE_AUTO_SHUTTER GD_MTC_SDK_DisableShutter

DFS_ENABLE_AUTO_SHUTTER GD_MTC_SDK_EnableShutter
DFS_SET_AUTO_SHUTTER_TIME GD_MTC_SDK_SetShutterTime

DFS_ADD_RECT_ANALYSER GD_MTC_SDK_AddRectAnalyser
DFS_DELETE_RECT_ANALYSER GD_MTC_SDK_DeleteRectAnalyser
DFS_MAX_MIN_AVG_TYPE GD_MTC_SDK_ChangeAnalyserTemperatureShowType
DFS_ENABLE_DIGITAL_TEMP_WITH_CURSOR
 GD_MTC_SDK_EnableRectAnalyserDigitalTempWithCursor
GD_MTC_SDK_DisableRectAnalyserDigitalTempWithCursor

DFS_ADD_STRING GD_MTC_SDK_AddStringToStream
DFS_ENABLE_ADD_STRING GD_MTC_SDK_EnableAddStringToStream
DFS_DISABLE_ADD_STRING GD_MTC_SDK_DisableAddStringToStream

DFS_GET_DEVICE_SERIAL_NUMBER GD_MTC_SDK_GetDeviceSerialNumber
DFS_GET_ARM_VERSION GD_MTC_SDK_GetArmVersion
DFS_GET_FPGA_VERSION GD_MTC_SDK_GetFpgaVersion
DFS_GET_SDK_VERSION GD_MTC_SDK_GetSDKVersion

DFS_CHANGE_IP GD_MTC_SDK_ChangeDeviceAddress
DFS_CHANGE_IP_NO_CONNECT GD_MTC_SDK_ChangeDeviceAddressWithoutConnection

DFS_GET_EMISS GD_MTC_SDK_getEmiss
DFS_SET_EMISS GD_MTC_SDK_setEmiss
DFS_GET_DISTANCE GD_MTC_SDK_getDistance
DFS_SET_DISTANCE GD_MTC_SDK_setDistance
DFS_GET_HUMIDITY GD_MTC_SDK_getHumidity
DFS_SET_HUMIDITY GD_MTC_SDK_setHumidity
DFS_GET_BACKGROUND_TEMP GD_MTC_SDK_getBackgroundTemp
DFS_SET_BACKGROUND_TEMP GD_MTC_SDK_setBackgroundTemp
DFS_BACKGROUND_TEMP_SWITCH_OPEN GD_MTC_SDK_getBackgroundTempSwitch
DFS_BACKGROUND_TEMP_SWITCH_CLOSED GD_MTC_SDK_setBackgroundTempSwitch
DFS_GET_B2 GD_MTC_SDK_getB2
DFS_SET_B2 GD_MTC_SDK_setB2
DFS_GET_KF GD_MTC_SDK_GetGearKf
DFS_SET_KF GD_MTC_SDK_SetGearKf
DFS_SET_VSK_GEAR GD_MTC_SDK_SetVskGear
DFS_SAVE_ALL_PARM GD_MTC_SDK_saveCalcParm

DFS_GET_BRIGHTNESS GD_MTC_SDK_GetBrightness
DFS_SET_BRIGHTNESS GD_MTC_SDK_SetBrightness
DFS_GET_CONTRAST GD_MTC_SDK_GetContrast
DFS_SET_CONTRAST GD_MTC_SDK_SetContrast


```

DFS_GET_Y16_POINT_TEMP GD_MTC_SDK_GetPointTemp
DFS_GET_Y16_MATRIX_TEMP GD_MTC_SDK_GetTempMatrix
DFS_GET_Y16_MAX_TEMP GD_MTC_SDK_GetHighTemp
DFS_GET_Y16_MIN_TEMP GD_MTC_SDK_GetLowTemp
DFS_GET_Y16_AVG_TEMP GD_MTC_SDK_GetAvegTemp
DFS_GET_Y16_RECT_MAX_TEMP GD_MTC_SDK_GetRectHighTemp
DFS_GET_Y16_RECT_MIN_TEMP GD_MTC_SDK_GetRectLowTemp
DFS_GET_Y16_RECT_AVG_TEMP GD_MTC_SDK_GetRectAvegTemp

DFS_GET_PIC_POINT_TEMP GD_MTC_SDK_GetPointTemperatureFromPicture
GD_MTC_SDK_GetTempPictureDemension
DFS_GET_PIC_MATRIX_TEMP GD_MTC_SDK_GetGlobalTemperatureFromPicture
DFS_GET_PIC_MAX_MIN_AVG_TEMP GD_MTC_SDK_GetRectTemperatureFromPicture

DFS_GET_DEVICE_ALARM_STATE GD_MTC_SDK_GetDeviceAlarmState
DFS_SET_DEVICE_ALARM_STATE GD_MTC_SDK_SetDeviceAlarmState
DFS_GET_DEVICE_ALARM_MAX_TEMP GD_MTC_SDK_GetDeviceAlarmMax
DFS_SET_DEVICE_ALARM_MAX_TEMP GD_MTC_SDK_SetDeviceAlarmMax
DFS_GET_DEVICE_ALARM_MIN_TEMP GD_MTC_SDK_GetDeviceAlarmMin
DFS_SET_DEVICE_ALARM_MIN_TEMP GD_MTC_SDK_SetDeviceAlarmMin

DFS_GET_RECT_ANALYSER_INFO GD_MTC_SDK_GetRectAnalyserInfo
DFS_GET_ADD_STRING_INFO GD_MTC_SDK_GetAddStringInfo

DFS_SET_EX_LEVEL GD_MTC_SDK_SetExLevel
GD_MTC_SDK_GetExLevel

```

2.10 獲取設備信息

```
Int32_t GD_MTC_SDK_GetDeviceInfo(const int iLoginID, GD_MTC_IRDeviceInfo *pIRDevice, int *pUser);
```

功能:

獲取機芯設備信息.目前支持的字段包括: 設備 IP, 設備類型, 設備版本號, 圖像寬, 圖像高.GD_MTC_IRDeviceInfo 結構體中 DeviceVersion 字段表示設備獲取到的設備版本號, 此字符串以一個空格為分隔, 空格前為 ARM 版本號, 空格後為 FPGA 版本號.

參數:

iLoginID:

調用打開數據流成功時得到的數據流 ID.

pIRDevice:

設備信息對象地址.如果調用成功, 通過此參數讀取設備信息.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.11 快門

Int32_t GD_MTC_SDK_Shutter(const int iLoginID, int *pUser);

功能:

快門. 注意, 在成功調用 GD_MTC_SDK_FocusAuto 接口且機芯在調焦過程動作結束前調用此接口會失敗.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_ShutterClose(const int iLoginID, int *pUser);

功能:

關閉快門. 注意, 在成功調用 GD_MTC_SDK_FocusAuto 接口且機芯在調焦過程動作結束前調用此接口會失敗. 注意, 如果需要在讓快門一直保持閉合狀態, 在調用此接口前可調用 GD_MTC_SDK_DisableShutter 接口禁用自動快門.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_ShutterOpen(const int iLoginID, int *pUser);

功能:

彈開快門. 注意, 在成功調用 GD_MTC_SDK_FocusAuto 接口且機芯在調焦過程動作結束前調用此接口會失敗.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_EnableShutter(const int iLoginID, int *pUser);

功能:

啓用機芯自動快門.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_DisableShutter(const int iLoginID, int *pUser);

功能:

禁用機芯自動快門. 禁用機芯自動快門後機芯設備不會自動打快門, 但程序仍可調用 GD_MTC_SDK_Shutter 接口打快門.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetShutterTime(const int iLoginID, int nMin, int *pUser);

功能:

設置機芯自動快門的時間間隔.

參數:

iLoginID:

登入 ID.

nMin:

機芯自動快門的時間間隔, 單位分鐘, 值的範圍為 1~255.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.12 拍照

```
Int32_t GD_MTC_SDK_SnapPictureUniform(const int iSessionID, const char * strFilePath, bool  
bFullPath, bool bImagData, bool bTempData, PROCESS_EDIT_SNAP_PIC_CB pSnapPicCb = 0, int *  
pUser = 0);
```

功能:

通用版本的拍照接口函數. 可指定保存圖片的文件夾名稱或文件全路徑, 可指定是否包含圖像信息, 是否包含溫度信息, 可在保存圖片前修改或自定義圖像數據.

參數:

iSessionID:

數據流 ID.

strFilePath:

保存拍照圖片的文件夾路徑或圖片的全路徑. 當參數 **bFullPath** 為 **false** 時, **strFilePath** 參數為保存圖片的文件夾路徑, 當 **bFullPath** 參數為 **true** 時, **strFilePath** 參數為保存圖片的全路徑.

bFullPath:

指示 **strFilePath** 參數是否是保存的全路徑.

bImagData:

當此參數為 **true** 時表示保存的圖片中包含圖像數據, 否則不包含.

bTempData:

當此參數為 **true** 時表示保存的圖片中包含溫度數據, 否則不包含.

pSnapPicCb :

在保存圖像前修改或定義圖像數據的回調函數. 具體參照的 **PROCESS_EDIT_SNAP_PIC_CB** 說明.

pUser :

用戶指針. SDK 內部在調用 **pSnapPicCb** 回調函數時反饋給調用者.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Typedef int (CALLBACK* PROCESS_EDIT_SNAP_PIC_CB)(unsigned char * pRgbData, int  
nRgbDataLen, int nRgbWidth, int nRgbHeight, void * pUser);
```

功能:

修改或自定義拍照圖像數據回調函數. 通過自定義此函數實現在保存圖片前修改或指定圖片的圖像部分數據.

參數:

pRgbData:

圖像的 RGB 數據.

nRgbDataLen:

RGB 數據的字節長度.

nRgbWidth:

圖像的寬度.

nRgbHeight:

圖像的高度.

pUser:

用戶指針.

返回值:

返回 1.

Int32_t GD_MTC_SDK_SnapPicture(void * pOpque, const char * strFilePath);

功能:

回調形式的含溫度信息拍照.在自定義的 Y16 回調中調用此函數.使用此函數得到的圖片文件中同時包含圖像數據和溫度數據.

參數:

pOpque:

Y16 回調函數傳入的對應值.

strFilePath:

保存拍照圖片的路徑.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SnapPictureNoCb(const int iSessionID, const char * strFilePath);

功能:

非回調形式的含溫度信息拍照.使用此函數得到的圖片文件中同時包含圖像數據和溫度數據.

參數:

iSessionID:

數據流 ID.

保存拍照圖片的路徑.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

`Int32_t GD_MTC_SDK_SnapPictureWithoutTempNoCb(const int iSessionID, const char * strFilePath);`

功能:

非回調形式的不含溫度信息拍照.使用此函數得到的圖片文件中僅含圖像數據.

參數:

`iSessionID`:

數據流 ID.

`strFilePath`:

保存拍照圖片的路徑.

返回值:

`NO_ERROR` 值表示調用成功, 其它值表示調用失敗.

`Int32_t GD_MTC_SDK_SnapPictureWithoutImgNoCb(const int iSessionID, const char * strFilePath);`

功能:

非回調形式的不含圖像信息拍照.使用此函數得到的圖片文件中僅含溫度數據.

參數:

`iSessionID`:

數據流 ID.

`strFilePath`:

保存拍照圖片的路徑.

返回值:

`NO_ERROR` 值表示調用成功, 其它值表示調用失敗.

2.13 Y16 錄像

`Int32_t GD_MTC_SDK_StartRecord(const int iSessionID, const char * strFilePath);`

功能:

開始 Y16 錄像.調用此函數生成的包含溫度信息的 Y16 視頻文件, 文件格式是 `irgd`, 此類型文件需要用專門的紅外分析工具查看. 注意, 在關閉數據流前如果正在進行 Y16 錄像, 請調用 `GD_MTC_SDK_StopRecord` 函數停止錄像.注意,Y16 錄像時不能調用接口獲取視頻的中心溫.

參數:

`iSessionID`:

數據流 ID.

`strFilePath`:

保存錄像文件的路徑.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_StopRecord(const int iSessionID);

功能:

停止 Y16 錄像. 注意, 在關閉數據流前如果正在進行 Y16 錄像, 請調用此函數停止錄像.

參數:

iSessionID:

數據流 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.14 AVI 錄像

Int32_t GD_MTC_SDK_StartRecordAvi(const int iSessionID, const char * strFilePath);

功能:

開始 AVI 錄像.調用此函數生成的 AVI 文件可用支持此格式的視頻播放器播放. 注意, 在關閉數據流前如果正在進行 AVI 錄像, 請調用 GD_MTC_SDK_StopRecordAvi 函數停止錄像.

參數:

iSessionID:

數據流 ID.

strFilePath:

保存錄像文件的路徑.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_StopRecordAvi(const int iSessionID);

功能:

停止 AVI 錄像. 注意, 在關閉數據流前如果正在進行 AVI 錄像, 請調用此函數停止錄像.

參數:

iSessionID:

數據流 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.15 設置色帶

`Int32_t GD_MTC_SDK_SetPalette(const int iSessionID, int iPaletteIndex, int *pUser);`

功能:

設置圖像的色帶.

參數:

iSessionID:

數據流 ID.

iPaletteIndex:

要設置的色帶號.值的範圍為 0-7.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.16 獲取色帶數據

`Int32_t GD_MTC_SDK_GetPaletteDataSize(const int iSessionID, int * pValue);`

功能:

獲取色帶數據的字節長度.

參數:

iSessionID:

數據流 ID.

pValue:

接收數據長度的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

`Int32_t GD_MTC_SDK_GetPaletteData(const int iSessionID, unsigned char * pPaletteData, int * pDemension0, int * pDemension1, int * pDemension2, char * strDataFormat);`

功能:

獲取色帶數據.

參數:

iSessionID:

數據流 ID.

pPaletteData:

色帶數據, 一個三維數組.

pDemension0:

色帶數據第一個維度的大小.即色帶的序號.

pDimension1:

色帶數據第二個維度的大小.即一條色帶中包含的顏色點的個數.

pDimension2:

色帶數據第三個維度的大小.即一個顏色點的數據.

strDataFormat:

色帶中單個顏色點的格式, 例如「**bgra**」表示 4 個字節, 依次是藍,綠,紅,透明.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.17 設置 Y16 幀率

Int32_t GD_MTC_SDK_setY16FramePerSecond(const int iLoginID, int iFps, int *pUser);

功能:

設置 Y16 數據的幀率.

參數:

iLoginID:

登入 ID.

iFps:

新的 Y16 數據幀率.幀率的範圍為 1~25.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.18 調焦

Int32_t GD_MTC_SDK_FocusNear(const int iSessionID, int *pUser);

功能:

調近焦. 調用此接口後機芯設備會持續將鏡頭拉近, 直至最近或者調用 **GD_MTC_SDK_FocusStop** 停止調焦. 注意, 在成功調用 **GD_MTC_SDK_Shutter** 接口且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iSessionID:

數據流 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_FocusFar(const int iSessionID, int *pUser);

功能:

調遠焦. 調用此接口後機芯設備會持續將鏡頭拉遠, 直至最遠或者調用 GD_MTC_SDK_FocusStop 停止調焦. 注意, 在成功調用 GD_MTC_SDK_Shutter 接口且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iSessionID:

數據流 ID.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_FocusStop(const int iSessionID, int *pUser);

功能:

停止調焦. 注意, 在成功調用 GD_MTC_SDK_Shutter 操作且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iSessionID:

數據流 ID.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_FocusAuto(const int iSessionID, int *pUser);

功能:

自動調焦. 調用此函數後機芯設備自動動作調焦至最合適狀態. 注意, 在成功調用 GD_MTC_SDK_Shutter 接口且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iSessionID:

數據流 ID.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_FocusFarFine(const int iSessionID, int *pUser);

功能:

遠焦微調. 注意, 在成功調用 GD_MTC_SDK_Shutter 接口且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iSessionID:

數據流 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_FocusNearFine(const int iSessionID, int *pUser);

功能:

近焦微調. 注意, 在成功調用 GD_MTC_SDK_Shutter 接口且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iSessionID:

數據流 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_setMotorPostition(const int iLoginID, int iMotorPos, int *pUser);

功能:

設置調焦電機位置.注意, 在成功調用 GD_MTC_SDK_Shutter 接口且機芯正在執行快門動作期間調用, 調用此接口會失敗.

參數:

iLoginID:

登入 ID.

iMotorPos:

電機將要移動到的位置.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_getMotorPostition(const int iLoginID, int * pMotorPos, int *pUser);
```

功能:

獲取調焦電機位置.

參數:

iLoginID:

登入 ID.

pMotorPos:

接收當前電機位置的地址.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.19 獲取 SDK 版本信息

```
Int32_t GD_MTC_SDK_GetSDKVersion(const int iLoginID, char *SDKVersion, int *pUser);
```

功能:

獲取 SDK 版本號.

參數:

iLoginID:

傳入 0 值即可.

SDKVersion:

函數向此地址處寫入 SDK 版本號.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

獲取機芯序列號

```
Int32_t GD_MTC_SDK_GetDeviceSerialNumber(const int iLoginID, char* serialNum, int *pUser);
```

功能:

獲取機芯設備的序列號.

參數:

iLoginID:

登入 ID.

serialNum:

函數向此地址處寫入序列號.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.20 修改機芯設備 IP

```
Int GD_MTC_SDK_ChangeDeviceAddressWithoutConnection(const char* sNewIP, const char* sNetMask, const char* sNewGateWay, const char* sNewDNS, const char* sOriIP);
```

功能:

修改設備的 IP 地址.調用函數不要求已經登入和已打開數據流, 如果已經打開數據流, 請先關閉之. 此函數與 GD_MTC_SDK_ChangeDeviceAddress 函數相比, 建議優先使用此函數.在設備是未登入狀態時調用此函數.

參數:

sNewIP:

新的 IP 地址.

sNetMask:

子網掩碼.

sNewGateWay:

新的網關.

sNewDNS:

新的 DNS.

sOriIP:

原來的 IP.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int GD_MTC_SDK_ChangeDeviceAddress(const int iSessionID, const char* sNewIP, const char* sNetMask, const char* sNewGateWay, const char* sNewDNS, const char* sOriIP);
```

功能:

修改設備的 IP 地址.調用函數要求已經登入和已打開數據流.此函數與 GD_MTC_SDK_ChangeDeviceAddressWithoutConnection 函數相比, 建議優先使用後者.

參數:

iSessionID:

數據流 ID.

sNewIP:

新的 IP 地址.

sNetMask:

子網掩碼.

sNewGateWay:

新的網關.

sNewDNS:

新的 DNS.

sOriIP:

原來的 IP.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.21 獲取測溫參數

Int32_t GD_MTC_SDK_getEmiss(const int iLoginID, float * pEmiss, int *pUser);

功能:

獲取發射率.

參數:

iLoginID:

登入 ID.

pEmiss:

接收當前發射率的地址.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_getDistance(const int iLoginID, float * pDistance, int *pUser);

功能:

獲取距離.

參數:

iLoginID:

登入 ID.

pDistance:

接收當前距離的地址.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_getHumidity(const int iLoginID, int * pHumidity, int *pUser);

功能:

獲取濕度.

參數:

iLoginID:

登入 ID.

pHumidity:

接收當前濕度的地址.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_getBackgroundTemp(const int iLoginID, float * pBgTemp, int *pUser);

功能:

獲取背景溫度.

參數:

iLoginID:

登入 ID.

pBgTemp:

接收當前背景溫度的地址. 背景溫度的單位是攝氏度.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_getBackgroundTempSwitch(const int iLoginID, int * pEnable, int *pUser);

功能:

獲取背景溫度開關.

參數:

iLoginID:

登入 ID.

pEnable:

接收當前背景溫度開關的地址. 1 表示開狀態, 0 表示關狀態.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_getB2(const int iLoginID, float * pB2, int *pUser);
```

功能:

獲取 B2 參數.

參數:

iLoginID:

登入 ID.

pB2:

接收當前 B2 參數的地址. B2 參數的單位是攝氏度.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetGearKf(const int iSessionID, int * pKf);
```

功能:

獲取 KF 參數.

參數:

iSessionID:

數據流 ID.

pKf:

接收當前 KF 參數的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.22 設置測溫參數

```
Int32_t GD_MTC_SDK_setEmiss(const int iLoginID, float fEmiss, int *pUser);
```

功能:

設置發射率. 注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設

備.

參數:

iLoginID:

登入 ID.

fEmiss:

新的發射率. 值的範圍為 0 到 1.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_setDistance(const int iLoginID, float fDistance, int *pUser);
```

功能:

設置距離. 注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設備.

參數:

iLoginID:

登入 ID.

fDistance:

距離, 單位米, 值的範圍為 1~10.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_setHumidity(const int iLoginID, int iHumidity, int *pUser);
```

功能:

設置濕度. 注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設備.

參數:

iLoginID:

登入 ID.

iHumidity:

濕度, 值的範圍為 0~100.

pUser:

用戶指針. 傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

`Int32_t GD_MTC_SDK_setBackgroundTemp(const int iLoginID, float fBgTemp, int *pUser);`

功能:

設置背景溫度.注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設備.

參數:

iLoginID:

登入 ID.

fBgTemp:

背景溫度. 單位攝氏度

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

`Int32_t GD_MTC_SDK_BackgroundTempSwitch(const int iLoginID, int nEnable, int *pUser);`

功能:

背景溫度開關.注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設備.

參數:

iLoginID:

登入 ID.

nEnable:

傳入 1 表示開, 0 表示關.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

`Int32_t GD_MTC_SDK_setB2(const int iLoginID, float fB2, int *pUser);`

功能:

設置 B2 參數. 可通過設置 B2 參數以線性調整 SDK 輸出的溫度值, 例如將此參數增大 1, SDK 輸出的溫度值會增加 1 攝氏度.一般先通過 GD_MTC_SDK_getB2 獲取當前的 B2 數值, 再根據

需要調整 B2 參數數值, 再調用此函數將新的 B2 參數設置下去.注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設備.

參數:

iLoginID:

登入 ID.

fB2:

B2 參數的值.單位攝氏度.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetGearKf(const int iSessionID, int nKf);

功能:

設置 KF 參數. 注意修改此參數後只對設備此次開機有效, 設備斷電重啓後此設置無效, 如需此修改在設備斷電重啓後仍然有效, 請調用 GD_MTC_SDK_saveCalcParm 函數保存修改到設備.

參數:

iSessionID:

數據流 ID.

nKf:

KF 參數的值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetVskGear(const int iSessionID, int nGear);

功能:

設置當前的測溫檔位.

參數:

iSessionID:

數據流 ID.

nGear:

當前的檔位值, 目前支持 1, 2.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetUartCheckAddress(const int iLoginID, int iValue);

功能:

設置 UART 校驗地址.

參數:

iLoginID:

登入 ID.

iValue:

UART 校驗地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetUartRate(const int iLoginID, int iValue);

功能:

設置 UART 傳輸速率.

參數:

iSessionID:

數據流 ID.

iValue:

UART 傳輸速率.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.23 保存測溫參數

Int32_t GD_MTC_SDK_saveCalcParm(const int iLoginID, int *pUser);

功能:

將測溫參數的修改保存到機芯設備. 測溫相關參數成功保存到機芯設備後, 設備斷電重啓後相關參數的修改仍然有效. 保存過程需要 2 分鐘, 請在調用此函數後的 2 分鐘之內不要操作設備, 保持設備與電腦的連接狀態同時不再調用 SDK 中的接口函數. 測溫參數包括: 發射率, 距離, 濕度, 背景溫度, 背景溫度開關, B2 參數, KF 參數.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_saveB2(const int iLoginID, int *pUser);

功能:

保存 B2 參數. 將 B2 參數的修改保存到機芯設備.B2 參數成功保存到機芯設備後, 設備斷電重啓後 B2 參數的修改仍然有效. 保存過程需要 2 分鐘, 請在調用此函數後的 2 分鐘之內不要操作設備, 保持設備與電腦的連接狀態同時不再調用 SDK 中的接口函數.

參數:

iLoginID:

登入 ID.

pUser:

用戶指針.傳入 0 值即可.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.24 獲取實時點溫

Int32_t GD_MTC_SDK_GetPointTemp(void * pOpque, GD_MTC_TempPointInfo *pTempPoint);

功能:

獲取點溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

pTempPoint:

表示點和溫度的對象地址.向此對象傳入要獲取點的坐標值,函數調用成功後讀取此點的溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.25 獲取實時全圖溫

Int32_t GD_MTC_SDK_GetRgbTempMatrixMaxMinAvgTemp(const int iSessionID, unsigned char * pRgbData, int * pRgbWidth, int * pRgbHeight, float * pTempMatrix, int * pMatrixWidth, int * pMatrixHeight, GD_MTC_TempPointInfo * pTempMax, GD_MTC_TempPointInfo * pTempMin, GD_MTC_TempPointInfo * pTempAvg)

功能:

獲取 RGB 數據, 溫度矩陣, 全圖最高溫, 最低溫, 平均溫.

參數:

iSessionID:

數據流 ID.

pRgbData:

傳入用於接收 RGB 數據的緩衝區地址,函數調用成功後讀取對應的 RGB 數據.緩衝區中能夠容納的元素個數為圖像的寬與高的乘積.

pRgbWidth:

傳入用於接收 RGB 數據寬度的地址.

pRgbHeight:

傳入用於接收 RGB 數據高度的地址.

pTempMatrix:

傳入用於接收溫度數據的緩衝區地址,函數調用成功後讀取對應的溫度值.緩衝區中能夠容納的元素個數為圖像的寬與高的乘積.

pMatrixWidth:

傳入用於接收溫度數據寬度的地址.

pMatrixHeight:

傳入用於接收溫度數據高度的地址.

pTempMax:

最高溫.表示最點和溫度的對象地址.函數調用成功後讀取對應的坐標和溫度值.

pTempMin:

最低溫.表示點和溫度的對象地址.函數調用成功後讀取對應的坐標和溫度值.

pTempAvg:

平均溫.表示點和溫度的對象地址.函數調用成功後讀取對應的溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetTempMatrix(void * pOpque, float *pTempMatrix);
```

功能:

獲取全圖溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

pTempMatrix:

傳入用於接收溫度數據的緩衝區地址,函數調用成功後讀取對應的溫度值.緩衝區中能夠容納的元素個數為圖像的寬與高的乘積.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetHighTemp(void * pOpque, GD_MTC_TempPointInfo *pTempPoint);
```

功能:

獲取全圖最高溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

pTempPoint:

表示點和溫度的對象地址.函數調用成功後讀取對應的坐標和溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetLowTemp(void * pOpque, GD_MTC_TempPointInfo *pTempPoint);
```

功能:

獲取全圖最低溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

pTempPoint:

表示點和溫度的對象地址.函數調用成功後讀取對應的坐標和溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetAvegTemp(void * pOpque, GD_MTC_TempPointInfo *pTempPoint);
```

功能:

獲取全圖平均溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

pTempPoint:

表示點和溫度的對象地址.函數調用成功後讀取對應的溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.26 獲取實時矩形溫

```
Int32_t GD_MTC_SDK_GetRectHighTemp(void * pOpque, GD_MTC_RectangleInfo RectInfo,  
GD_MTC_TempPointInfo *pTempPoint);
```

功能:

獲取指定矩形區域的最高溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

RecInfo:

表示一個矩形區域. 調用此函數前設置好此結構體中的各成員.

pTempPoint:

表示點和溫度的對象地址.函數調用成功後讀取對應的坐標和溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetRectLowTemp(void * pOpque, GD_MTC_RectangleInfo RecInfo,  
GD_MTC_TempPointInfo *pTempPoint);
```

功能:

獲取指定矩形區域的最低溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

RecInfo:

表示一個矩形區域. 調用此函數前設置好此結構體中的各成員.

pTempPoint:

表示點和溫度的對象地址.函數調用成功後讀取對應的坐標和溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetRectAvegTemp(void * pOpque, GD_MTC_RectangleInfo RecInfo,  
GD_MTC_TempPointInfo *pTempPoint);
```

功能:

獲取指定矩形區域的平均溫.在自定義的 Y16 回調函數中調用此函數.

參數:

pOpque:

Y16 回調函數中傳入的對應值.

RecInfo:

表示一個矩形區域. 調用此函數前設置好此結構體中的各成員.

pTempPoint:

表示點和溫度的對象地址.函數調用成功後讀取對應的溫度值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.27 獲取圖片溫

Int32_t GD_MTC_SDK_GetTempPictureDemension(const char* filename, int * pWidth, int * pHeight);

功能:

獲取溫度圖片的寬和高.

參數:

Filename:

溫度圖片的路徑.

pWidth:

傳入用於接收溫度圖片的寬度的地址.

pHeight:

傳入用於接收溫度圖片的高度的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetPointTemperatureFromPicture(const char* filename, int x, int y, float* point);

功能:

獲取溫度圖片中任意點的溫度.

參數:

Filename:

溫度圖片的路徑.

X:

圖片中的點相對圖片左上角的水平坐標.

Y:

圖片中的點相對圖片左上角的垂直坐標.

point:

傳入用於接收溫度圖片中指定點溫度值的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetGlobalTemperatureFromPicture(const char* filename, float* imageT);

功能:

獲取溫度圖片的溫度矩陣.在調用此函數前可調用 `GD_MTC_SDK_GetTempPictureDemension` 接口來獲取溫度圖片的寬和高.

參數:

Filename:

溫度圖片的路徑.

imageT:

傳入用於接收溫度矩陣的地址.溫度矩陣的字節長度是寬乘高個 `float` 的長度.

返回值:

`NO_ERROR` 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetRectTemperatureFromPicture(const char* filename, int x, int y, int width,
int height, POINT_INFO* maxInfo, POINT_INFO* minInfo, float* avgT);
```

功能:

獲取溫度圖片中指定矩形區域的最高溫及坐標, 最低溫及坐標, 平均溫. 在調用此函數前可調用 `GD_MTC_SDK_GetTempPictureDemension` 接口來獲取溫度圖片的寬高.

參數:

Filename:

溫度圖片的路徑.

X:

指定矩形區域左上角相對圖片左上角的水平坐標.

Y:

指定矩形區域左上角相對圖片左上角的垂直坐標.

Width:

指定矩形區域的寬度.

Height:

指定矩形區域的高度.

maxInfo:

傳入用於接收指定矩形區域中最高溫信息的結構體地址. 如果此函數調用成功,此結構體中包含對應點的坐標及溫度值.

minInfo:

傳入用於接收指定矩形區域中最低溫信息的結構體地址. 如果此函數調用成功,此結構體中包含對應點的坐標及溫度值.

avgT:

傳入用於接收指定矩形區域平均溫的地址..

返回值:

`NO_ERROR` 值表示調用成功, 其它值表示調用失敗.

2.28 添加點線面分析對象

```
Int32_t GD_MTC_SDK_AddPlpPoint(const int iLoginID, GD_MTC_PLP_PointInfo * pPoint, int nNum = 1);
```

功能:

在視頻流中添加點分析對象. 添加後會 SDK 輸出的圖像上顯示此對象及其溫度信息. 可調用此接口一次添加多個對象.

參數:

iLoginID:

登入 ID.

pPoint:

要添加的點對象起始地址.

nNum :

對象的個數.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

GD_MTC_PLP_PointInfo 結構體說明:

Struct GD_MTC_PLP_PointInfo

```
{
    GD_MTC_PLP_PointInfo()
    {
        PointX = 0;
        PointY = 0;
        nHide = 0;
        nCursorType = CURSOR_MODE_MAX;
        nPlpId = 0;
        PointTemp = 0.0f;
    }

    Int nPlpId;
    Int nHide;
    Int nCursorType;

    Short PointX;    // 溫度點 X 坐標
    Short PointY;    // 溫度點 Y 坐標
    Float PointTemp; // 溫度點溫度
};
```

成員:

nPlpId:

對象的 PLP 編號.

nHide:

對象是否隱藏.

nCursorType:

光標類型. 目前未使用.

PointX:

對象的水平坐標.

PointY:

對象的垂直坐標.

PointTemp:

對象的溫度.

```
Int32_t GD_MTC_SDK_AddPlpLine(const int iLoginID, GD_MTC_PLP_LineInfo * pLine, int nNum = 1);
```

功能:

在視頻流中添加線分析對象. 添加後會 SDK 輸出的圖像上顯示此對象及其溫度信息. 可調用此接口一次添加多個對象.

參數:

iLoginID:

登入 ID.

pLine:

要添加的線對象起始地址.

nNum :

對象元素個數.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

GD_MTC_PLP_LineInfo 結構體說明:

Struct GD_MTC_PLP_LineInfo

```
{
    GD_MTC_PLP_LineInfo()
    {
        nHide = 0;
        nCursorType = CURSOR_MODE_MAX;
        nPlpId = 0;
    }
}
```

Int nPlpId;

Int nHide;

Int nCursorType;

GD_MTC_PointInfo PointBegin;

GD_MTC_PointInfo PointEnd;

GD_MTC_TempPointInfo PointTempMax; // 溫度點溫度

```

        GD_MTC_TempPointInfo PointTempMin;    // 溫度點溫度
        GD_MTC_TempPointInfo PointTempAvg;    // 溫度點溫度
};

```

成員:

nPlpId:

對象的 PLP 編號.

nHide:

對象是否隱藏.

nCursorType:

光標類型. 目前未使用.

PointBegin:

線的起點.

PointEnd:

線的終點.

PointTempMax:

線上的最高溫點信息.

PointTempMin:

線上的最低溫點信息.

PointTempAvg:

線上的平均溫信息.

```

Int32_t GD_MTC_SDK_AddPlpRect(const int iLoginID, GD_MTC_PLP_RectangleInfo * pRect, int
nNum = 1);

```

功能:

在視頻流中添加矩形分析對象. 添加後會 SDK 輸出的圖像上顯示此對象及其溫度信息. 可調用此接口一次添加多個對象.

參數:

iLoginID:

登入 ID.

pRect:

要添加的矩形對象起始地址.

nNum :

對象的個數.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

GD_MTC_PLP_RectangleInfo 結構體說明:

```

Struct GD_MTC_PLP_RectangleInfo
{
    GD_MTC_PLP_RectangleInfo()
    {

```

```

        nHide = 0;
        nCursorType = CURSOR_MODE_MAX;
        nPlpId = 0;
    }

    Int nPlpId;
    Int nHide;
    Int nCursorType;

    Short X; // 矩形左上角 X 坐標
    Short Y; // 矩形左上角 Y 坐標
    Int Width; // 矩形寬度
    Int Height; // 矩形高度

    GD_MTC_TempPointInfo PointTempMax; // 溫度點溫度
    GD_MTC_TempPointInfo PointTempMin; // 溫度點溫度
    GD_MTC_TempPointInfo PointTempAvg; // 溫度點溫度
};

```

成員:

nPlpId:

對象的 PLP 編號.

nHide:

對象是否隱藏.

nCursorType:

光標類型. 目前未使用.

X:

矩形的左上角頂點水平坐標.

Y:

矩形的左上角頂點垂直坐標.

Width:

矩形的寬度.

Height:

矩形的高度.

PointTempMax:

矩形中的最高溫點信息.

PointTempMin:

矩形中的最低溫點信息.

PointTempAvg:

矩形中的平均溫信息.

2.29 更新點線面分析對象

```

Int32_t GD_MTC_SDK_UpdatePlpPoint(const int iLoginID, GD_MTC_PLP_PointInfo * pPoint, int
nNum = 1);

```

功能:

更新視頻流中已添加的點分析對象.

參數:

iLoginID:

登入 ID.

pPoint:

要添加的點對象起始地址.指定對象的 PLP 編號必須已存在.

nNum :

對象的個數.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_UpdatePlpLine(const int iLoginID, GD_MTC_PLP_LineInfo * pLine, int nNum  
= 1);
```

功能:

更新視頻流中已添加的線分析對象.

參數:

iLoginID:

登入 ID.

pLine:

要添加的線對象起始地址.指定對象的 PLP 編號必須已存在.

nNum :

對象的個數.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_UpdatePlpRect(const int iLoginID, GD_MTC_PLP_RectangleInfo * pRect, int  
nNum = 1);
```

功能:

更新視頻流中已添加的矩形分析對象.

參數:

iLoginID:

登入 ID.

pRect:

要添加的矩形對象起始地址.指定對象的 PLP 編號必須已存在.

nNum :
對象的個數.

返回值:
NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.30 獲取點線面分析對象

Int32_t GD_MTC_SDK_GetPlpPoint(const int iLoginID, GD_MTC_PLP_PointInfo * pPoint, int * pNum);

功能:
獲取視頻流中已添加的點分析對象相關信息.

參數:
iLoginID:
登入 ID.
pPoint:
接收點分析對象信息的起始地址.
nNum :
獲取到的對象的個數.

返回值:
NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetPlpLine(const int iLoginID, GD_MTC_PLP_LineInfo * pLine, int * pNum);

功能:
獲取視頻流中已添加的線分析對象相關信息.

參數:
iLoginID:
登入 ID.
pLine:
接收線分析對象信息的起始地址.
nNum :
獲取到的對象的個數.

返回值:
NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetPlpRect(const int iLoginID, GD_MTC_PLP_RectangleInfo * pRect, int * pNum);

功能:

獲取視頻流中已添加的矩形分析對象相關信息.

參數:

iLoginID:

登入 ID.

pRect:

接收矩形分析對象信息的起始地址.

nNum :

獲取到的對象的個數.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.31 刪除點線面分析對象

Int32_t GD_MTC_SDK_DeletePlp(const int iLoginID, int nPlpId);

功能:

刪除已添加到視頻流中的分析對象.

參數:

iLoginID:

登入 ID.

nPlpId:

PLP 對象的 ID 號.此參數為 0 表示刪除所有點線面分析對象.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_DeleteAllPlp(const int iLoginID);

功能:

刪除已添加到視頻流中的所有分析對象.

參數:

iLoginID:

登入 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.32 同步點線面分析對象

Int32_t GD_MTC_SDK_SyncPlpToLocal(const int iLoginID, int nFlag);

功能:

相機與庫之間的點線面分析對象同步.

參數:

iLoginID:

登入 ID.

nFlag:

同步標誌, 1 表示從相機同步到庫, 0 表示從庫同步到相機.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.33 添加矩形分析對象

Int32_t GD_MTC_SDK_AddRectAnalyser(const int iLoginID, int x, int y, int width, int height, int type);

功能:

在視頻流中添加矩形溫度分析區域. 根據函數中的標記類型, 會標記顯示矩形域內的最高溫及最低溫點. 目前僅部分機型在 Y8 數據流模式支持.

參數:

iLoginID:

登入 ID.

X:

以視頻圖像左上角頂點為起始點, 矩形左上角頂點相對於此點的 x 值.

Y:

以視頻圖像左上角頂點為起始點, 矩形左上角頂點相對於此點的 y 值.

Width:

矩形的寬.

Height:

矩形的高.

Type:

標記類型. 如果標記類型中包含最高溫或最低溫, 則會在指定的矩形內用十字光標標記最高溫或最低溫點. 標記類型數值的具體含義如下:

0 表示最高溫,

1 表示最低溫,

2 表示平均溫,

3 表示最高溫+最低溫,

4 表示最高溫+平均溫,

5 表示最低溫+平均溫,

6 表示最高溫+最低溫+平均溫,

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_DeleteRectAnalyser(const int iLoginID);

功能:

刪除視頻流中的矩形溫度分析區域。

參數:

iLoginID:

登入 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_ChangeAnalyserTemperatureShowType(const int iLoginID, int type);

功能:

在視頻流中添加矩形溫度分析區域. 根據矩形分析區域的標記類型, 會標記顯示域內的最高溫及最低溫點. 目前僅部分機型在 Y8 數據流模式支持.

參數:

iLoginID:

登入 ID.

Type:

標記類型. 如果標記類型中包含最高溫或最低溫或平均溫, 則會在視頻流的左下方顯示最高溫或最低溫或平均溫的溫度數值. 標記類型數值的具體含義如下:

0 表示最高溫,

1 表示最低溫,

2 表示平均溫,

3 表示最高溫+最低溫,

4 表示最高溫+平均溫,

5 表示最低溫+平均溫,

6 表示最高溫+最低溫+平均溫,

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_EnableRectAnalyserDigitalTempWithCursor(const int iLoginID);

功能:

在矩形分析對象中的光標旁邊顯示數字溫度.

參數:

iLoginID:

登入 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_DisableRectAnalyserDigitalTempWithCursor(const int iLoginID);

功能:

在矩形分析對象中的光標旁邊不顯示數字溫度.

參數:

iLoginID:

登入 ID.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetRectAnalyserInfo(const int iLoginID, GD_MTC_RECT_ANALYSER_INFO *
pRectAnalyser)

功能:

獲取矩形溫度分析區域信息.

參數:

iLoginID:

登入 ID.

pRectAnalyser:

矩形溫度分析區域信息, 具體查看 GD_MTC_RECT_ANALYSER_INFO 說明.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

GD_MTC_RECT_ANALYSER_INFO 結構體說明:

Struct GD_MTC_RECT_ANALYSER_INFO

{

Int nAnalyseMode = 0;

Int nCursorMode = 0;

Int x = 0;

Int y = 0;

Int nWidth = 0;

Int nHeight = 0;

Int nEnableDigitalTempWithCursor = 0;

};

成員:

nAnalyseMode :

分析模式.

- 0 表示關閉;
- 1 表示點分析;
- 2 表示區域分析

nCursorMode :

光標模式.

- 0 表示最高溫;
- 1 表示最低溫;
- 2 表示平均溫;
- 3 表示最高溫和最低溫;
- 4 表示最高溫和平均溫;
- 5 表示最低溫和平均溫;
- 6 表示最高溫和最低溫和平均溫

X :

矩形區域左上角水平坐標.

Y :

矩形區域左上角垂直坐標.

nWidth :

矩形區域寬度.

nHeight :

矩形區域高度.

nEnableDigitalTempWithCursor :

是否在矩形分析對象中光標旁邊顯示數字溫度.

2.34 視頻流疊加字符串

Int32_t GD_MTC_SDK_AddStringToStream(const int iLoginID, int x, int y, const void* pStrData, int nStrByteLen, const char * strEncodeName);

功能:

在指定的坐標處疊加指定的字符串到視頻流中. 目前僅部分機型在 Y8 數據流模式支持.

參數:

iLoginID:

登入 ID.

X:

以視頻圖像左上角頂點為起始點, 字符串矩形區域左上角頂點相對於此點的 x 值.

Y:

以視頻圖像左上角頂點為起始點, 字符串矩形區域左上角頂點相對於此點的 y 值.

pStrData:

要疊加到視頻流中的字符串數據的地址.

nStrByteLen:

字符串數據的字節長度.

strEncodeName:

字符串數據的編碼方式, 在調用此接口時傳入「utf16」。目前此接口僅支持「utf16」方式編碼的字符串。

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗。

備注:

接口變更(2018.06.18), 接口參數變化, 添加字符串原有接口不再支持。

Int32_t GD_MTC_SDK_EnableAddStringToStream(const int iLoginID);

功能:

啓用字符串疊加功能。目前僅部分機型在 Y8 數據流模式支持。

參數:

iLoginID:

登入 ID。

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗。

Int32_t GD_MTC_SDK_DisableAddStringToStream(const int iLoginID);

功能:

禁用字符串疊加功能。目前僅部分機型在 Y8 數據流模式支持。

參數:

iLoginID:

登入 ID。

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗。

Int32_t GD_MTC_SDK_GetAddStringInfo(const int iLoginID, GD_MTC_ADD_STRING_INFO * pAddStringInfo)

功能:

獲取字符串疊加信息。目前僅部分機型在 Y8 數據流模式支持。

參數:

iLoginID:

登入 ID。

pAddStringInfo:

字符串疊加信息, 具體查看 GD_MTC_ADD_STRING_INFO 說明。

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

GD_MTC_ADD_STRING_INFO 結構體說明:

Struct GD_MTC_ADD_STRING_INFO

```
{
    Int nEnabled;
    Int x;
    Int y;
    Unsigned char strData[128];
    Int nStrByteLen;
    GD_MTC_ADD_STRING_INFO()
    {
        nEnabled = 0;
        x = 0;
        y = 0;
        memset(strData, 0, sizeof(strData));
        nStrByteLen = 0;
    }
};
```

成員:

nEnabled:

當前是否啓用字符疊加狀態, 1 表示開啓, 0 表示關閉.

X:

疊加字符串區域左上角的水平坐標.

Y:

疊加字符串區域左上角的垂直坐標.

strData:

疊加的字符串數據, 數據是「utf16」編碼.

nStrByteLen:

疊加的字符串數據字節長度.

2.35 電子變焦

Int32_t GD_MTC_SDK_SetExLevel(const int iSessionID, GD_MTC_EX nExLevel, int nCenterX, int nCenterY);

功能:

設置溫度原始數據電子變焦.

參數:

iSessionID:

數據流 ID.

nExLevel:

電子變焦放大倍率.

GD_MTC_EX_1 表示原始圖像;

GD_MTC_EX_2 表示 2 倍放大;

GD_MTC_EX_4 表示 4 倍放大;

nCenterX:

電子變焦中心點水平坐標.

nCenterY:

電子變焦中心點垂直坐標.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetExLevel(const int iSessionID, GD_MTC_EX_INFO * pExInfo)

功能:

獲取電子變焦狀態.

參數:

iSessionID:

數據流 ID.

pExInfo:

電子變焦狀態, 具體查看 GD_MTC_EX_INFO 說明.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

GD_MTC_EX_INFO 說明

Struct GD_MTC_EX_INFO

```
{  
    GD_MTC_EX nExLevel;  
    Int nCenterX;  
    Int nCenterY;  
};
```

成員:

nExLevel:

電子變焦放大倍率.

GD_MTC_EX_1 表示原始圖像;

GD_MTC_EX_2 表示 2 倍放大;

GD_MTC_EX_4 表示 4 倍放大;

nCenterX:

電子變焦中心點水平坐標.

nCenterY:

電子變焦中心點垂直坐標.

2.36 調光模式

Int32_t GD_MTC_SDK_SetAdjustLightMode(const int iSessionID, int nMode);

功能:

設置調光模式.只有部分數據流模式支持, 具體查看功能支持表.

參數:

iSessionID:

數據流 ID.

nMode:

調光模式.目前調光模式有三種, 亮度對比度調光, 自動溫度調光, 手動溫度調光, 分別對應

ADJUST_LIGHT_MODE_BRIGHTNESS_CONTRAST

ADJUST_LIGHT_MODE_TEMPERATURE_AUTO

ADJUST_LIGHT_MODE_TEMPERATURE_MANUAL

當為亮度對比度調光時調用 GD_MTC_SDK_SetBrightness 和 GD_MTC_SDK_SetContrast 接口設置亮度對比度改變調光效果.

當為自動溫度調光時 SDK 自動根據當前溫度改變調光效果,無需要手動設置其它參數.

當為手動溫度溫度調光時調用 GD_MTC_SDK_SetSpanMin 和 GD_MTC_SDK_SetSpanMax 接口指定溫度高值和溫度低值來改變調光效果.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.37 亮度對比度調光

Int32_t GD_MTC_SDK_SetBrightness(const int iSessionID, int nValue);

功能:

設置圖像的亮度. 只有部分數據流模式支持, 具體查看功能支持表.

參數:

iSessionID:

數據流 ID.

nValue:

圖像的亮度, 範圍為 1 到 100.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetBrightness(const int iSessionID, int * pValue);

功能:

獲取圖像的亮度。只有部分數據流模式支持，具體查看功能支持表。

參數：

iSessionID:

數據流 ID。

pValue:

接收圖像亮度的地址。亮度值的範圍為 1 到 100。

返回值：

NO_ERROR 值表示調用成功，其它值表示調用失敗。

```
Int32_t GD_MTC_SDK_SetContrast(const int iSessionID, int nValue);
```

功能：

設置圖像的對比度。只有部分數據流模式支持，具體查看功能支持表。

參數：

iSessionID:

數據流 ID。

nValue:

圖像的對比度，範圍為 1 到 100。

返回值：

NO_ERROR 值表示調用成功，其它值表示調用失敗。

```
Int32_t GD_MTC_SDK_GetContrast(const int iSessionID, int * pValue);
```

功能：

獲取圖像的對比度。只有部分數據流模式支持，具體查看功能支持表。

參數：

iSessionID:

數據流 ID。

pValue:

接收圖像對比度的地址。對比度值的範圍為 1 到 100。

返回值：

NO_ERROR 值表示調用成功，其它值表示調用失敗。

2.38 手動溫度調光

```
Int32_t GD_MTC_SDK_SetSpanMin(const int iSessionID, float nValue);
```

功能：

設置手動調光溫度低值。只有部分數據流模式支持，具體查看功能支持表。

參數:

iSessionID:

數據流 ID.

nValue:

手動調光溫度低值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetSpanMin(const int iSessionID, float * pValue);
```

功能:

獲取手動調光溫度低值. 只有部分數據流模式支持, 具體查看功能支持表.

參數:

iSessionID:

數據流 ID.

pValue:

接收溫度低值的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_SetSpanMax(const int iSessionID, float nValue);
```

功能:

設置手動調光溫度高值. 只有部分數據流模式支持, 具體查看功能支持表.

參數:

iSessionID:

數據流 ID.

nValue:

手動調光溫度高值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

```
Int32_t GD_MTC_SDK_GetSpanMax(const int iSessionID, float * pValue);
```

功能:

獲取手動調光溫度高值. 只有部分數據流模式支持, 具體查看功能支持表.

參數:

iSessionID:

數據流 ID.

pValue:

接收溫度高值的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.39 機芯設備溫度報警

Int32_t GD_MTC_SDK_GetDeviceAlarmState(const int iSessionID, int * pState);

功能:

獲取機芯設備溫度報警的當前開關狀態. 如果觸發機芯設備溫度報警, 不管是高溫報警還是低溫報警, 機芯設備後端對應針角會輸出報警電平.

參數:

iSessionID:

數據流 ID.

pState:

接收開關狀態的地址. 值的含義具體如下:

0 表示高溫報警關, 低溫報警關;

1 表示高溫報警開, 低溫報警關;

2 表示高溫報警關, 低溫報警開;

3 表示高溫報警開, 低溫報警開;

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetDeviceAlarmState(const int iSessionID, int nState);

功能:

設置機芯設備溫度報警的當前開關狀態.

參數:

iSessionID:

數據流 ID.

nState:

要設置的溫度報警開關狀態. 值的含義具體如下:

0 表示高溫報警關, 低溫報警關;

1 表示高溫報警開, 低溫報警關;

2 表示高溫報警關, 低溫報警開;

3 表示高溫報警開, 低溫報警開;

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetDeviceAlarmMax(const int iSessionID, float * pTemp);

功能:

獲取機芯設備溫度報警的高溫報警閾值.

參數:

iSessionID:

數據流 ID.

pTemp:

接收高溫報警閾值的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetDeviceAlarmMax(const int iSessionID, float fTemp);

功能:

設置機芯設備溫度報警的高溫報警閾值.

參數:

iSessionID:

數據流 ID.

fTemp:

要設置的高溫報警閾值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetDeviceAlarmMin(const int iSessionID, float * pTemp);

功能:

獲取機芯設備溫度報警的低溫報警閾值.

參數:

iSessionID:

數據流 ID.

pTemp:

接收低溫報警閾值的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetDeviceAlarmMin(const int iSessionID, float fTemp);

功能:

設置機芯設備溫度報警的低溫報警閾值.

參數:

iSessionID:

數據流 ID.

fTemp:

要設置的低溫報警閾值.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.40 機芯設備工作狀態

Int32_t GD_MTC_SDK_GetDeviceState(const int iSessionID, int * pState, int * pLen);

功能:

獲取機芯設備工作狀態.

參數:

iSessionID:

數據流 ID.

pState:

接收狀態信息的地址.此參數為 0 時 pLen 參數將返回需要接收狀態信息的字節長度.異常狀態用特定的位表示, 用接收到的狀態信息按位與特定的位來判別當前是否異常, 各位的含義已在 SDK 中定義, 具體含義如下:

DEVICE_STATE_ARM_ERROR_NO_CURVE_FILE:

ARM 錯誤, 沒有曲線.

DEVICE_STATE_ARM_ERROR_Y16_ABNORMAL:

ARM 錯誤, Y16 數據異常.

DEVICE_STATE_ARM_ERROR_NO_MEASURE_PARAM:

ARM 錯誤, 沒有測溫參數.

DEVICE_STATE_ARM_ERROR_NO_CURVE_FILE:

ARM 錯誤, 字體庫缺失.

DEVICE_STATE_FPGA_ERROR_INVALID_SENSOR_CONFIG:

FPGA 錯誤, 無效的探測器配置.

DEVICE_STATE_FPGA_ERROR_LOAD_FLASH_PARAM_FAILED:

FPGA 錯誤, 加載 FLASH 參數失敗.

DEVICE_STATE_FPGA_ERROR_DEVICE_WORK_TEMP_OVER_RANGE:

FPGA 錯誤, 設備工作溫度超出範圍.

pLen:

接收狀態信息的地址字節長度.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

2.41 端口號獲取與設置

Int32_t GD_MTC_SDK_GetArmPort(const char * strIpAddress, int * pPort);

功能:

獲取 ARM 端口號.目前僅部分機型支持.

參數:

strIpAddress:

機芯 IP.

pPort:

接口端口號的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetArmPort(const char * strIpAddress, int nPort);

功能:

設置 ARM 端口號.目前僅部分機型支持.

參數:

strIpAddress:

機芯 IP.

nPort:

要設置的新的端口號.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetFpgaPort(const char * strIpAddress, int * pPort);

功能:

獲取 FPGA 端口號.目前僅部分機型支持.

參數:

strIpAddress:

機芯 IP.

pPort:

接口端口號的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetFpgaPort(const char * strIpAddress, int nPort);

功能:

設置 FPGA 端口號.目前僅部分機型支持.

參數:

strIpAddress:

機芯 IP.

nPort:

要設置的新的端口號.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_GetY16Port(const char * strIpAddress, int * pPort);

功能:

獲取 Y16 端口號.目前僅部分機型支持.

參數:

strIpAddress:

機芯 IP.

pPort:

接口端口號的地址.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

Int32_t GD_MTC_SDK_SetY16Port(const char * strIpAddress, int nPort);

功能:

設置 Y16 端口號.目前僅部分機型支持.

參數:

strIpAddress:

機芯 IP.

nPort:

要設置的新的端口號.

返回值:

NO_ERROR 值表示調用成功, 其它值表示調用失敗.

3. 其它

3.1 計算機配置性能建議

由於 SDK 在特定模式時需要對大量的溫度信息進行運算，因此對計算機的硬件性能有一定要求，建議計算機達到或超過如下配置：

1. CPU: Intel i5, 第五代, 雙核, 3.2GHz
2. 內存: 2G
3. 硬盤: 1T
4. 顯卡: GTX960, 顯存 2G

3.2 SDK 使用注意事項

1. 目前 SDK 保存圖片及視頻時存儲路徑中不能包含中文字符，路徑中只能包含英文符號及數字。
2. 針對部分支持修改端口號的海思平台相機，修改了 ARM 端口號後，需要重新編輯與庫在同一目錄的 MeasureCoreNet.cfg 文件，將其中的 HsArmPort 字段的數值改為新的端口號。FPGA 端口號和 Y16 端口號目前無此要求。