Industrial Temperature Measurement SDK User Manual V0.5

IRay Technology Co., Ltd.

www.infiray.com

Version History

Version	Modified	Date	Comments
V0.1	Sun Haifeng	2020-09-15	Initial release
V0.2	Sun Haifeng	2021-03-15	Compatible with AT20
V0.3	Sun Haifeng	2021-05-12	Support connecting multiple devices
V0.4	Sun Haifeng	2021-09-18	Add functions of capturing and analyzing state grid format JPG images
V0.5	Sun Haifeng	2021-11-03	Add function of analyzing IRG files

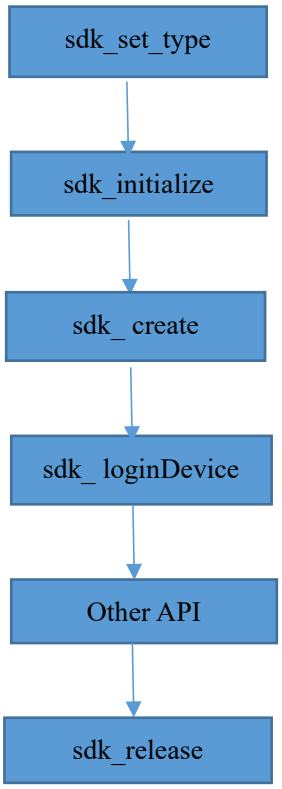
Table of Contents

1. Commands Set Overview

This SDK is applicable to three categories of products, referred to as Class A, Class B, and Class C below. Applicable models of Class A include ATF and LT series network modules, among which the level alarms are not supported between the Max temperature and the Min temperature. Applicable models of Class B include AT300, AT600, AT31, and AT61. Applicable model of Class C includes AT20.

Meaning of returned value: 0 for success, -1 for failure, 1 for not supported.

SDK calling process:



2. Detailed Commands Description

2.1 sdk_initialize ——A/B/C

【Description】

Initialize SDK

【Function】

int sdk_initialize();

【Parameters】

Parameters	Comments	Input/Output
None		

【Returned Value】

Returned value	Comments
0	Succeed to initialize
-1	Failed to initialize

2.2 sdk_create ——A/B/C

【Description】

Create Handle

【Function】

IRNETHANDLE sdk_create();

【Parameters】

Parameters	Comments	Input/Output
None		

【Returned Value】

Returned value	Comments
Handle	Handle, used to pass parameters to other API

2.3 sdk_ loginDevice —— A/B/C

【Description】

Log in device

【Function】

int sdk_loginDevice(IRNETHANDLE hHandle, ChannelInfo stinfo);

【Parameters】

Parameters	Comments	Input/Output
hHandle	sdk_create() returned value	Input
stinfo	Device information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded to log in
-1	Failed to log in

2.4 sdk_release —— A/B/C

【Description】

Release SDK

【Function】

int sdk_release(IRNETHANDLE p);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

【Returned Value】

Returned value	Comments
0	Succeeded to release
-1	Failed to release

2.5 sdk_set_type —— A/B/C

【Description】

Set device type

【Function】

void sdk_set_type(int iType, char* UserName, char* Password);

【Parameters】

Parameters	Comments	Input/Output
iType	Device type: 0： A 1: B 2:C	Input
UserName	User name for login	Input
Password	Password for login	Input

【Returned Value】

Returned value	Comments
None	

2.6 sdk_search_device —— A/B/C

【Description】

Search the IP address of the device

【Function】

int sdk_search_device(IRNETHANDLE p, DeviceList &devLst);

【Parameters】

Parameters	Comments	Input/Output

p	sdk_create() returned value	Input
devLst	Device information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The device struct is defined as follows:

```
#define MAX_DEVICE_NUM      100
struct DeviceList
{
 int iNumber;
 ChannelInfo DevInfo[MAX_DEVICE_NUM];
};
```

iNumber：the number of devices searched

DevInfo：a collection of searched device details

Please refer to demo for details.

2.7 SetMessageCallBack —— A/B

【Description】

Registered message callback.

【Function】

void __stdcall SetMessageCallBack(IRNETHANDLE p, MessageCallBack pMessageCallBack, void *pContext);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
pMessageCallBack	Message callback function	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
None	

【Notice】

The message callback function is defined as follows.

```
typedef void(*MessageCallBack)(IRNETHANDLE hHandle, WPARAM wParam, LPARAM lParam, void *context);
```

Please refer to demo for details.

2.8 SetDeviceVideoCallBack —— A/B

【Description】

Registered image data callback.

【Function】

```
int __stdcall SetDeviceVideoCallBack(IRNETHANDLE p, VideoCallBack0 pVideoCallBack, void *pContext);
```

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
pVideoCallBack	Image data callback function API	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The image data callback function is defined as follows.

```
typedef void(*VideoCallBack0)(char *pBuffer, long BufferLen, int width, int height, void *pContext);
```

Please refer to demo for details.

Image data format in callback：windows：yuv420，linux：h264

2.9 SetTempCallBack —— A/B

【Description】

Registered temperature data callback.

【Function】

int __stdcall SetTempCallBack(IRNETHANDLE p, TempCallBack pTempCallBack, void *pContext);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
pTempCallBack	Temperature data callback function API	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The temperature data callback function is defined as follows:

```
typedef void(*TempCallBack)(char *pBuffer, long BufferLen, void *pContext);
```

Please refer to demo for details.

For B-type products, the format of temperature data needs to be converted, and the conversion method is as follows:

```
memcpy(&temp_buffer[0], pBuffer, Width * Height * 2);

    for (int ii = 0; ii < Width * Height / 2; ii++)      // data conversion
    {
        temp_data_temp[ii * 2] = (unsigned short)((unsigned short)(temp_buffer[ii * 2] << 8) +
```

```
temp_buffer[ii * 2 + 1 + Width * Height]);
                temp_data_temp[ii * 2 + 1] = (unsigned short)((unsigned short)(temp_buffer[ii * 2 + 1] << 8) +
temp_buffer[ii * 2 + (Width * Height)]);
            }
```

2.10 SetSerialCallBack —— A

【Description】

Registered serial transmission callback.

【Function】

int __stdcall SetSerialCallBack(IRNETHANDLE p, ChannelInfo stinfo, SerialCallBack pSerialCallBack, void *pContext);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
pSerialCallBack	Serial transmission callback function API	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The serial callback function is defined as follows:

typedef void(*SerialCallBack)(char *pRecvDataBuff, int BuffSize, void *context);

Please refer to demo for details.

2.11 SetAlarmCallBack —— B/C

【Description】

Registered alarm callback.

【Function】

int __stdcall SetAlarmCallBack(IRNETHANDLE p, char* ip, AlarmCallBack pAlarmCallBack, void *pContext);

【Parameter】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
ip	Ip address of device	Input
pAlarmCallBack	Alarm callback function API	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The alarm callback function is defined as follows.

typedef void(*AlarmCallBack)(char* message, void *context);

Please refer to demo for details.

2.12 SetSnapCallBack —— A

【Description】

Registered snap callback.

【Function】

int __stdcall SetSnapCallBack(IRNETHANDLE p, ChannelInfo stinfo, SnapCallBack pSnapCallBack, void *pContext);

【Parameter】

Parameters	Comments	Input/Output

p	sdk_create() returned value	Input
stinfo	Device information struct	Input
pSnapCallBack	Snap callback function API	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The snap callback function is defined as follows.

typedef void(*SnapCallBack)(int m_ch, char *pBuffer, int size, void *context);

Please refer to demo for details.

2.13 sdk_CapSingle —— A

【Description】

Single capture.

【Function】

int sdk_CapSingle(IRNETHANDLE p, ChannelInfo stinfo);

【Parameter】

Parameters	Comments	Input/Output
p	sdk_create()returned value	Input
stinfo	Device information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.14 sdk_start_url —— B

【Description】

Class B product，start callback.

【Function】

int sdk_start_url(IRNETHANDLE p, char* ip);

【Parameter】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
ip	ip address of device	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.15 sdk_serial_cmd_send —— A/B

【Description】

Send serial commands.

【Function】

int sdk_serial_cmd_send(IRNETHANDLE p,char *pSendBuff, DWORD BuffSize);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
pSendBuff	Send commands(Hexadecimal)	Input
BuffSize	Commands length	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.16 sdk_serial_cmd_receive —— A/B

【Description】

Receive serial commands

【Function】

int sdk_serial_cmd_receive(IRNETHANDLE p,char *pRecvBuff, int *BuffSize);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
pRecvBuff	Send commands(Hexadecimal)	Output
BuffSize	Commands length	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.17 sdk_set_device_ip —— A/B/C

【Description】

Set the IP address of the device.

【Function】

int sdk_set_device_ip(IRNETHANDLE p, ChannelInfo stinfo , const char* DstIP, int port);

【Parameters】

Parameters	Comments	Input/Output

p	sdk_create() returned value	Input
stinfo	Device information struct	Input
DstIP	New IP address	Input
port	New port	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

Note: only the port of Class A can be modified.

2.18 sdk_osd_switch —— A/B/C

【Description】

OSD switch

【Function】

int sdk_osd_switch(IRNETHANDLE p, ChannelInfo stinfo, int iSwitch);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iSwitch	Switch 0:off 1:on	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.19 sdk_SetInfOsd —— A

【Description】

OSD parameter

【Function】

int sdk_SetInfOsd(IRNETHANDLE p, ChannelInfo stinfo, const INF_OSD &osd_p);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
osd_p	OSD parameter struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.20 sdk_LoadParamOsd —— A

【Description】

Load OSD parameter status

【Function】

int sdk_LoadParamOsd(IRNETHANDLE p, ChannelInfo stinfo, int* iOSD, INF_OSD *osd_p);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iOSD	Switch 0:off 1:on	Output
osd_p	OSD parameter struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.21 sdk_get_temp_data —— A/B/C

【Description】

Get the full frame and area temperature (including the maximum, minimum, average and central temperature)

【Function】

int sdk_get_temp_data(IRNETHANDLE p, ChannelInfo stinfo, int iIndex, Area_Temp &area_temp);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iIndex	region no 7-frame	Input
area_temp	Temperature struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.22 sdk_set_envir_param —— A/B/C

【Description】

Set environment parameter.

【Function】

int sdk_set_envir_param(IRNETHANDLE p,ChannelInfo stinfo, envir_param envir_data);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

stinfo	Device information struct	Input
envir_data	Environment parameter struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The variable value in the struct is the actual value*10000, for example: the reflected temperature is 25℃, and the parameter in the struct is 250,000.

```
typedef struct

    int emissivity;
    int airTemp;
    int reflectTemp;
    int humidity;
    int distance;
} envir_param; //Parameters are actual values * 10000
```

2.23 sdk_get_envir_param —— A/B/C

【Description】

Get environment parameters.

【Function】

int sdk_get_envir_param(IRNETHANDLE p,ChannelInfo stinfo, envir_param* envir_data);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
envir_data	Environment parameter struct	Output

【Returned Value】

Returned value	Comments

0	Succeeded
-1	Failed

【Notice】

The variable value in the struct is the actual value*10000, for example: the reflected temperature is 25℃, and the parameter in the struct is 250,000.

```
typedef struct
{
    int emissivity;
    int airTemp;
    int reflectTemp;
    int humidity;
    int distance;
} envir_param; //Parameters are actual values * 10000
```

2.24 sdk_envir_effect —— A/B

【Description】

Environment parameters take effect.

【Function】

int sdk_envir_effect(IRNETHANDLE p);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.25 sdk_shutter_correction —— A/B

【Description】

Correct shutter.

【Function】

int sdk_shutter_correction(IRNETHANDLE p ,int type);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
type	Default: 0	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.26 sdk_set_color_plate —— A/B/C

【Description】

Set palette.

【Function】

int sdk_set_color_plate(IRNETHANDLE p,ChannelInfo stinfo, int color_plate);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
color_plate	Palette number	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.27 sdk_get_SN_PN —— A/B/C

【Description】

Get SN and PN.

【Function】

int sdk_get_SN_PN(IRNETHANDLE p,ChannelInfo stinfo, char *strSN, char* strPN);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
strSN	SN	Output
strPN	PN	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.28 sdk_get_FPA_temp —— A/B

【Description】

Get FPA temperature.

【Function】

int sdk_get_FPA_temp(IRNETHANDLE p,float *fTemp);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
fTemp	FPA temperature	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.29 sdk_get_camera_temp —— A/B

【Description】

Get camera temperature.

【Function】

int sdk_get_camera_temp (IRNETHANDLE p,float *fTemp);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
fTemp	Camera temperature	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.30 sdk_get_width —— A/B

【Description】

Get PFA width.

【Function】

int sdk_get_width(IRNETHANDLE p,int *iValue);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iValue	FPA width	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.31 sdk_get_height —— A/B

【Description】

Get PAF height.

【Function】

int sdk_get_height(IRNETHANDLE p,int *iValue);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iValue	FPA height	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.32 sdk_get_TempImaging —— A/B

【Description】

Get the status of temperature imaging.

【Function】

int sdk_get_TempImaging(IRNETHANDLE p,int *iValue);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iValue	Temperature imaging switch 0:off 1:on	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.33 sdk_Convert_to_Celsius —— A/BC

【Description】

Convert to Celsius.

【Function】

int sdk_Convert_to_Celsius(int iType, int iTempImaging, unsigned short usValue, float* fTempC);

【Parameters】

Parameters	Comments	Input/Output
iType	Type 0:skin temperature measuring, 1:industrial temperature measuring	Input
iTempImaging	switch of temperature imaging	Input
usValue	Temperature to be converted	Input
fTempC	Temperature after conversion	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.34 sdk_get_wtr_status —— A/B

【Description】

Get wide temperature range status.

【Function】

int sdk_get_wtr_status(IRNETHANDLE p,int* iStatus);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iStatus	Switch of wide temperature range	Output

	0:off 1:on	

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.35 sdk_set_wtr_status —— A/B

【Description】

Set wide temperature range status.

【Function】

int sdk_set_wtr_status(IRNETHANDLE p ,int iStatus);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iStatus	Switch of wide temperature range 0:off 1:on	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.36 sdk_set_wtr_low_threshold —— A/B

【Description】

Set low threshold of wide temperature range.

【Function】

int sdk_set_wtr_low_threshold(IRNETHANDLE p ,int iThreshold);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

iThreshold	Low threshold of wide temperature range	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

iThreshold=actual value*10000

2.37 sdk_get_wtr_low_threshold —— A/B

【Description】

Get low threshold of wide temperature range.

【Function】

int sdk_get_wtr_low_threshold(IRNETHANDLE p ,int* iThreshold);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iThreshold	Low threshold of wide temperature range	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

iThreshold=actual value*10000

2.38 sdk_set_wtr_high_threshold —— A/B

【Description】

Set high threshold of wide temperature range.

【Function】

int sdk_set_wtr_high_threshold(IRNETHANDLE p ,int iThreshold);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iThreshold	High threshold of wide temperature range	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

iThreshold=actual value*10000

2.39 sdk_get_wtr_high_threshold —— A/B

【Description】

Get high threshold of wide temperature range.

【Function】

int sdk_get_wtr_high_threshold(IRNETHANDLE p ,int* iThreshold);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
iThreshold	High threshold of wide temperature range	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

iThreshold=actual value*10000

2.40 sdk_get_image_framerate —— A/B

【Description】

Get image frame rate.

【Function】

int sdk_get_image_framerate(IRNETHANDLE p,ChannelInfo stinfo, int* iFrameRate);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iFrameRate	Image frame rate	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.41 sdk_set_image_framerate —— A/B

【Description】

Set image frame rate.

【Function】

int sdk_set_image_framerate(IRNETHANDLE p,ChannelInfo stinfo, int iFrameRate);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iFrameRate	Image frame rate	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.42 sdk_get_temp_framerate —— B/C

【Description】

Get temperature frame rate.

【Function】

int sdk_get_temp_framerate(IRNETHANDLE p,ChannelInfo stinfo, int* iFrameRate);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iFrameRate	Temperature frame rate	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.43 sdk_set_temp_framerate —— B/C

【Description】

Set temperature frame rate.

【Function】

int sdk_set_temp_framerate(IRNETHANDLE p,ChannelInfo stinfo, int iFrameRate);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iFrameRate	Temperature frame rate	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.44 sdk_set_hw_io_output —— A

【Description】

Set hardware IO output interface

【Function】

int sdk_set_hw_io_output(IRNETHANDLE p,ChannelInfo stinfo, int iSwitch);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iSwitch	Switch 0:off 1:on	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.45 sdk_set_osd_display —— A/B/C

【Description】

Set OSD display

【Function】

int sdk_set_osd_display(IRNETHANDLE p,ChannelInfo stinfo, Custom_String osdContent);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
osdContent	OSD struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

The struct is defined as follows.

```
typedef struct
{
    int iFormat;//1:Center 2:Align left   3:Align right
    int iFormatTime;//0:off
                    //1:2020 - 07 - 20 16 : 18 : 30
                    //2 : 2020 - 07 - 20 FRI 16 : 18 : 30
                    //3 : 07 - 20 - 2020 16 : 18 : 30
                    //4 : 07 - 20 - 2020 FRI 16 : 18 : 30
                    //5 : 20 - 07 - 2020 16 : 18 : 30
                    //6 : 20 - 07 - 2020 FRI 16 : 18 : 30
                      //C:TimeEnable   0:off   1:on
    int iShow;   //0:Do not show   1:show    C:TitleEnable    0:off   1:on
    int iIndex; //0:time 1/2/3:Custom string
    char m_szString[200];//Content    B:utf-8
    int iWidth; //String width
    int iDeviceWidth; //Area array width
    int iDeviceHeight;   //Area array height
    int iX; //time coordinate
    int iY; //time coordinate
    int iStringX; //string coordinate
    int iStringY; //string coordinate
}Custom_String;//Overlay Custom string
```

2.46 sdk_get_osd_display ——C

【Description】

Get OSD display

【Function】

int sdk_get_osd_display(IRNETHANDLE p,ChannelInfo stinfo, Custom_String osdContent);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
osdContent	OSD struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.47 sdk_synchronised_time —— A/B/C

【Description】

Synchronize time.

【Function】

int sdk_synchronised_time(IRNETHANDLE p,ChannelInfo stinfo, Time_Param timeData);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
timeData	Time struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.48 sdk_set_DHCP_on_off —— A/B

【Description】

Set the switch of DHCP.

【Function】

int sdk_set_DHCP_on_off(IRNETHANDLE p,ChannelInfo stinfo, int iSwitch);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iSwitch	Switch 0:off 1:on	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.49 sdk_set_capture_format —— A

【Description】

Set capture image format.

【Function】

int sdk_set_capture_format(IRNETHANDLE p, ChannelInfo stinfo, int iFormat);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iFormat	3:jpg 4:jpg+irg	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.50 sdk_snapshot —— A

【Description】

Snapshot images.

【Function】

int sdk_snapshot(IRNETHANDLE p, ChannelInfo stinfo, int iLocation, char* strPath);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iLocation	0：SD card 1：local	Input
strPath	iLocation=1，local pathe.g.：C:\	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】

Before capturing, you can call sdk_set_capture_format to set the format of the

captured image

2.51 sdk_get_timing_recording —— B

【Description】

Get timed recording parameters.

【Function】

int sdk_get_timing_ recording(IRNETHANDLE p, ChannelInfo stinfo, Recoding* data);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
data	Timed video parameter struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.52 sdk_set_timing_ recording —— B

【Description】

Set timed recording parameters.

【Function】

int sdk_set_timing_recording(IRNETHANDLE p, ChannelInfo stinfo, Recoding data);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

stinfo	Device information struct	Input
data	Timed video parameter struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.53 sdk_get_timing_capture —— AB

【Description】

Get timed capture.

【Function】

int sdk_get_timing_capture(IRNETHANDLE p,ChannelInfo stinfo, Encoding_Format* data);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
data	Timed capture struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.54 sdk_set_timing_capture —— AB

【Description】

Set timed capture.

【Function】

int sdk_set_timing_capture(IRNETHANDLE p, ChannelInfo stinfo,

Encoding_Format data);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
data	Timed capture struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.55 sdk_set_temp_alarm —— AB

【Description】

Set temperature alarm.

【Function】

int sdk_set_temp_alarm(IRNETHANDLE p,ChannelInfo stinfo, int m_regionIndex, Alarm_Config alarm_config);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
m_regionIndex	0:frame; >0 region	Input
alarm_config	Temperature alarm detailed information	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.56 sdk_disk_format —— A

【Description】

Format SD card.

【Function】

int sdk_disk_format(IRNETHANDLE p,ChannelInfo stinfo, int iDiskNo);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iDiskNo	Disk No.	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.57 sdk_get_temp_unit —— AB

【Description】

Get temperature unit.

【Function】

int sdk_get_temp_unit(IRNETHANDLE p, ChannelInfo stinfo, int *iUnit);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iUnit	0:Celsius 1:Kelvin 2:Fahrenheit	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.58 sdk_set_temp_unit —— AB

【Description】

Set temperature unit.

【Function】

int sdk_set_temp_unit(IRNETHANDLE p, ChannelInfo stinfo, int iUnit);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iUnit	0:Celsius 1:Kelvin 2:Fahrenheit	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.59 sdk_get_temp_configuration —— ABC

【Description】

Get temperature configuration.

【Function】

int sdk_get_temp_configuration(IRNETHANDLE p, ChannelInfo stinfo, int iIndex, Alarm_Config& temp_config);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iIndex	Area index	Input
temp_config	Temperature configuration struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.60 sdk_set_area_pos —— AB

【Description】

Set area position.

【Function】

int sdk_set_area_pos (IRNETHANDLE p, ChannelInfo stinfo, int iIndex, Area_pos area_pos);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iIndex	Area index	Input
area_pos	Area parameter struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.61 sdk_remove_area_pos —— ABC

【Description】

Remove area position

【Function】

int sdk_remove_area_pos (IRNETHANDLE p, ChannelInfo stinfo, int iIndex, int iMode);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iIndex	Area index	Input
iMode	0:point 1:line 2:area	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.62 sdk_close_alarm —— BC

【Description】

Close alarm.

【Function】

void sdk_close_alarm(IRNETHANDLE p);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

【Returned Value】

Returned value	Comments
None	

2.63 sdk_analyze_alarm_info —— BC

【Description】

Analyze alarm information.

【Function】

int sdk_analyze_alarm_info(IRNETHANDLE p, char* strAlarm, Alarm_Info* alarm_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
strAlarm	Alarm character string	Input
alarm_info	Alarm information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.64 sdk_ reset_param —— A

【Description】

Reset parameters.

【Function】

int sdk_reset_param(IRNETHANDLE p, ChannelInfo stinfo);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.65 sdk_get_wlan —— C

【Description】

Get WLAN.

【Function】

int sdk_get_wlan(IRNETHANDLE p,ChannelInfo stinfo, Wlan_Config*

wlan_config);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
wlan_config	WLAN information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.66 sdk_set_wlan —— C

【Description】

Set WLAN.

【Function】

int sdk_set_wlan(IRNETHANDLE p,ChannelInfo stinfo, Wlan_Config wlan_config);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
wlan_config	WLAN information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.67 sdk_get_all_user_info —— C

【Description】

Get all user information.

【Function】

int sdk_get_all_user_info(IRNETHANDLE p,ChannelInfo stinfo, User_Info user_info[USER_NUM]);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
user_info	User information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

Note: Currently, the maximum number of users supported is 100.

2.68 sdk_create_new_user —— C

【Description】

Create new user.

【Function】

Int sdk_create_new_user(IRNETHANDLE p,ChannelInfo stinfo, User_New user_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

stinfo	Device information struct	Input
user_info	New user information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.69 sdk_get_user_online —— C

【Description】

Get list of online users.

【Function】

int sdk_get_user_online(IRNETHANDLE p,ChannelInfo stinfo, User_Online user_info[USER_NUM]);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
user_info	Online user information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.70 sdk_get_user_info —— C

【Description】

Get information of specified ID user.

【Function】

int sdk_get_user_info(IRNETHANDLE p,ChannelInfo stinfo, int id, User_Info* user_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
id	User ID	Input
user_info	User information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.71 sdk_modify_user_info —— C

【Description】

Modify user information.

【Function】

int sdk_modify_user_info(IRNETHANDLE p,ChannelInfo stinfo, User_Modify user_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
user_info	User information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.72 sdk_delete_user —— C

【Description】

Delete user.

【Function】

int sdk_delete_user(IRNETHANDLE p,ChannelInfo stinfo, int id);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
id	User ID	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.73 sdk_get_no_opr_timeout —— C

【Description】

Get no operation timeout.

【Function】

int sdk_get_no_opr_timeout(IRNETHANDLE p,ChannelInfo stinfo, int* timeout);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
timeout	No operation timeout	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.74 sdk_set_no_opr_timeout —— C

【Description】

Set no operation timeout..

【Function】

int sdk_set_no_opr_timeout(IRNETHANDLE p,ChannelInfo stinfo, int timeout);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
timeout	No operation timeout	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.75 sdk_get_all_group_info —— C

【Description】

Get all group information.

【Function】

int sdk_get_all_group_info(IRNETHANDLE p,ChannelInfo stinfo, Group_Info group_info [USER_NUM]);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
group_info	Group information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

Note: Currently, the maximum number of groups supported is 100.

2.76 sdk_create_new_group —— C

【Description】

Creat new group.

【Function】

int sdk_create_new_group(IRNETHANDLE p,ChannelInfo stinfo, Group_Info group_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
group_info	Group information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.77 sdk_get_group_info —— C

【Description】

Get information of specified group.

【Function】

int sdk_get_group_info(IRNETHANDLE p,ChannelInfo stinfo, int id, Group_Info* group_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
id	User ID	Input
group_info	Group information struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.78 sdk_modify_group_info —— C

【Description】

Modify group information.

【Function】

int sdk_modify_group_info(IRNETHANDLE p,ChannelInfo stinfo, int id, Group_Info group_info);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

stinfo	Device information struct	Input
id	User ID	Input
group_info	Group information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.79 sdk_delete_group —— C

【Description】

Delete group.

【Function】

int sdk_delete_group(IRNETHANDLE p,ChannelInfo stinfo, int id);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
id	User ID	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.80 sdk _get_device_setting —— C

【Description】

Get setting of device.

【Function】

int sdk_get_device_setting(IRNETHANDLE p,ChannelInfo stinfo,

Device_Setting* device_setting);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
device_setting	Device setting struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.81 sdk_set_device_setting —— C

【Description】

Set device setting.

【Function】

int sdk_set_device_setting(IRNETHANDLE p,ChannelInfo stinfo,

Device_Setting device_setting);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
device_setting	Device setting struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.82 sdk_get_record_param —— C

【Description】

Get record parameters.

【Function】

int sdk_get_record_param(IRNETHANDLE p,ChannelInfo stinfo, Record_Param* record_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
record_param	Record parameter struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.83 sdk_set_record_param —— C

【Description】

Set record parameters.

【Function】

int sdk_set_record_param(IRNETHANDLE p,ChannelInfo stinfo, Record_Param record_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input

record_param	Record parameter struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.84 sdk_get_record_path —— C

【Description】

Get record storage path.

【Function】

int sdk_get_record_path(IRNETHANDLE p,ChannelInfo stinfo, Record_Path*

record_path);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
record_path	Record storage path struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.85 sdk_search_record_file —— C

【Description】

Search record files.

【Function】

int sdk_search_record_file(IRNETHANDLE p,ChannelInfo stinfo,

Record_Search record_search, list<char*> &file);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
record_search	Recording retrieval condition struct	Input
file	Record files list	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.86 sdk_delete_record_file —— C

【Description】

Delete recorded files.

【Function】

int sdk_delete_record_file(IRNETHANDLE p,ChannelInfo stinfo, char* filePath);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
filePath	Absolute path of recording file	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

Note: Call sdk_get_record_path to get the storage path, and then call

sdk_search_record_file to get the file list. The combination of the two is filePath.

2.87 sdk_get_snap_param —— C

【Description】

Get snap parameters.

【Function】

int sdk_get_snap_param(IRNETHANDLE p,ChannelInfo stinfo, Snap_Param* snap_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
snap_param	Snap parameters struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.88 sdk_set_snap_param —— C

【Description】

Set snap parameters.

【Function】

int sdk_set_snap_param(IRNETHANDLE p,ChannelInfo stinfo, Snap_Param snap_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
snap_param	Snap parameters struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.89 sdk_get_GB28181_config —— C

【Description】

Get GB28181 configuration.

【Function】

int sdk_get_GB28181_config(IRNETHANDLE p,ChannelInfo stinfo, GB28181_Param* config_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
config_param	GB28181 parameters struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.90 sdk_set_GB28181_config —— C

【Description】

Set GB28181 configuration.

【Function】

int sdk_set_GB28181_config(IRNETHANDLE p,ChannelInfo stinfo, GB28181_Param config_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
config_param	GB28181 parameters struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.91 sdk_system_upgrade —— C

【Description】

System upgrade, and upload remote upgrade package.

【Function】

int sdk_system_upgrade(IRNETHANDLE p,ChannelInfo stinfo, int format, char* file);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
file	Absolute path of upgrade package	Input

【Returned Value】

Returned value	Comments
0	Succeeded

-1	Failed
1	Not supported

2.92 sdk_set_area_pos_new —— A

【Description】

Set area position of Class A.

【Function】

int sdk_set_area_pos_new (IRNETHANDLE p, ChannelInfo stinfo, int iIndex,

Area_pos area_pos);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
iIndex	Area index	Input
area_pos	Area parameter struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.93 sdk_start_record —— A

【Description】

Start recording (Class A)

【Function】

int sdk_start_record(IRNETHANDLE p, ChannelInfo stinfo, char* file);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

stinfo	Device information struct	Input
file	Complete record file path	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
1	Not supported

2.94 sdk_stop_record —— A

【Description】

Stop recording (Class A)

【Function】

int sdk_stop_record(IRNETHANDLE p, ChannelInfo stinfo);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.95 sdk_get_temp_offline —— A

【Description】

Get temperature offline (get the temperature of the specified area from the IRG file)

【Function】

int sdk_get_temp_offline(char* file, Position_info& pos_info);

【Parameters】

Parameters	Comments	Input/Output
file	Complete path of irg files	Input
pos_info	Parameter struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.96 sdk_snapshot_jpg —— B

【Description】

Capture state grid format jpg

【Function】

int sdk_snapshot_jpg(IRNETHANDLE p, ChannelInfo stinfo, char* strPath);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
strPath	Save complete path	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.97 sdk_get_temp_data_param —— B

【Description】

Get temperature data.

【Function】

int sdk_get_temp_data_param(IRNETHANDLE p, ChannelInfo stinfo,

TempData_Param* tempdata_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
tempdata_param	Temperature date struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.98 sdk_set_temp_data_param —— B

【Description】

Set temperature data.

【Function】

int sdk_set_temp_data_param(IRNETHANDLE p, ChannelInfo stinfo,

TempData_Param* tempdata_param);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
tempdata_param	Temperature date struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.99 sdk_open_jpg_param —— B

【Description】

Get parameter from state grid jpg.

【Function】

int sdk_open_jpg_param(char* file, JPG_Param* jpg_param);

【Parameters】

Parameters	Comments	Input/Output
file	Complete path of jpg file	Input
jpg_param	Parameter struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
2	Not captured by AT61P

2.100 sdk_open_jpg_data —— B

【Description】

Get temperature data and image data from state grid jpg

【Function】

int sdk_open_jpg_param(char* file, float* temp_data, unsigned char* image_data);

【Parameters】

Parameters	Comments	Input/Output
file	Complete path of jpg file	Input
temp_data	Temperature data of full frame	Output
image_data	Image data of full frame	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed
2	Not captured by AT61P

2.101 sdk_get_temp_offline_jpg —— B

【Description】

Get the maximum, minimum and average temperature of some area from state grid jpg

【Function】

int sdk_get_temp_offline_jpg(float* temp_data, int width, int height, Position_info& pos_info);

【Parameters】

Parameters	Comments	Input/Output
temp_data	Temeprature date of full frame	Input
width	FPA width	Input
height	FPA height	Input
pos_info	Temperature struct	Input / Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.102 sdk_temp_data_correction —— B

【Description】

Perform temperature correction according to environmental variables for state grid format jpg.

【Function】

int sdk_temp_data_correction(float srcTempBuffer, JPG_envir_param srcEvn_param, JPG_envir_param dstEvn_param, float* dstTempBuffer);

【Parameters】

Parameters	Comments	Input/Output
srcTempBuffer	Original temperature	Input
srcEvn_param	Original environment struct	Input
dstEvn_param	New environment struct	Input
dstTempBuffer	Temperature after correction	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.103 sdk_stretch_temp —— B

【Description】

Stretch temperature for state grid format jpg.

【Function】

int sdk_stretch_temp(Stretch_param stretch_param, unsigned char* image_data, unsigned char* out_data);

【Parameters】

Parameters	Comments	Input/Output
stretch_param	Parameter struct	Input
image_data	Original image data	Input
out_data	Image data after stretch	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.104 sdk_ get_irg_param—— A

【Description】

Get parameter from irg file

【Function】

int sdk_get_irg_param(char* file, IRG_Param* irg_param);

【Parameters】

Parameters	Comments	Input/Output
file	Complete path of irg file	Input
irg_param	Parameter struct	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.105 sdk_ get_irg_data—— A

【Description】

Get image data and temperature of irg file

【Function】

int sdk_get_irg_data(char* file, int colorIndex, unsigned short* temp_data, unsigned char* image_data);

【Parameters】

Parameters	Comments	Input/Output
file	Complete path of irg file	Input
colorIndex	Color index	Input
temp_data	Temperature data	Output
image_data	Image data	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

Note: The image data format is YUYV, and the temperature data format is K*10.

2.106 SetSnapGeneralCallBack—— A

【Description】

Register snap callback (regular version)

【Function】

int __stdcall SetSnapGeneralCallBack(IRNETHANDLE p, ChannelInfo stinfo, SnapCallBack pSnapCallBack, void *pContext);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
pSnapCallBack	Snap callback function API	Input
pContext	User context	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

【Notice】 The snapshot callback function is defined as follows.

typedef void(*SnapCallBack)(int m_ch, char *pBuffer, int size, void *context);

After the registration callback is successful, use the sdk_CapSingle APP to trigger.

For detailed usage, please refer to the demo.

2.107 sdk_stop_url —— B

【Description】

Stop url (Class B)

【Function】

int sdk_stop_url(IRNETHANDLE p);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.108 sdk_ get_onvif_port —— A

【Description】

Get Onvif port (Class A)

【Function】

int sdk_get_onvif_port(IRNETHANDLE p, ChannelInfo stinfo, unsigned short* port);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
port	Onvif port	Output

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.109 sdk_ set_onvif_port —— A

【Description】

Set Onvif port (Class A)

【Function】

int sdk_set_onvif_port(IRNETHANDLE p, ChannelInfo stinfo, unsigned short* port);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input
port	Onvif port	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.110 sdk_ save_param —— A

【Description】

Save parameters (Class A)

【Function】

int sdk_save_param(IRNETHANDLE p, ChannelInfo stinfo);

【Parameters】

Parameters	Comments	Input/Output
p	sdk_create() returned value	Input
stinfo	Device information struct	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

2.111 sdk_ get_pseudo_color_pic—— A/B/C

【Description】

Get palette picture.

【Function】

int sdk_get_pseudo_color_pic(char *filename, int index, int width, int height);

【Parameters】

Parameters	Comments	Input/Output
filename	Complete path of picture e.g. C:\1.jpg	Input
index	Palette index	Input
width	Palette width	Input
height	Palette height	Input

【Returned Value】

Returned value	Comments
0	Succeeded
-1	Failed

3. Programming Example

3.1 Log in

```
#include "stdafx.h"
#include <iostream >
#include <string>
#include "atlstr.h"
#include "time.h"

#include "InfraredTempSDK.h"
#include "InfEntity.h"

using namespace std;
    //设备类型
    enum DeviceType { DEVICE_TYPE_A = 0, DEVICE_TYPE_B = 1, DEVICE_TYPE_C = 2 };
    enum SdkReturnType { SDK_RETURN_SUCCESS = 0, SDK_RETURN_FAIL = -1, SDK_RETURN_NULL
 = 1 };

    IRNETHANDLE _sdkHandle;
```

```
bool _isLogin = false;     //是否已登录
DeviceType _deviceType;            //设备类型
ChannelInfo _deviceInfo;           //设备信息

void device_login()
{
        _deviceType = DEVICE_TYPE_A;        //设备类型
        string userName = "888888";         //账号
        string userPwd = "888888";          //密码

        //1、设置设备类型
        sdk_set_type((int)_deviceType, (char*)(userName.c_str()), (char*)(userPwd.c_str()));

        //2、初始化，A 类、B 类
        if (_deviceType != DEVICE_TYPE_C)
        {
                int iRes = sdk_initialize();
                if (iRes != 0)
                {
                        cout << "初始化失败!" << endl;
                        return;
                }
        }

        //3、创建 handle
        _sdkHandle = sdk_create();

        //4、登录
        _deviceInfo.channel = 0;
        _deviceInfo.wPortNum = 3000;
        strcpy(_deviceInfo.szIP, "10.10.25.36");
        strcpy(_deviceInfo.szServerName, "IRCAM");
        memcpy(_deviceInfo.szUserName, userName.c_str(), sizeof(userName));
        memcpy(_deviceInfo.szPWD, userPwd.c_str(), sizeof(userPwd));

        _isLogin = sdk_loginDevice(_sdkHandle, _deviceInfo) == SDK_RETURN_SUCCESS;
        cout << (_isLogin ? "\n\n 登录成功！ " : "\n\n 登录失败！ ") << endl;
}

int main()
```

```
{
        device_login();

        cin.get();
        sdk_release(_sdkHandle);
        return 0;
}
```

3.2 Message Callback

```
void showMsgLink(int type)
{
        string msg;
        switch (type)
        {
        case 0:
                msg = "连接成功";
                break;
        case 1:
                msg = "用户停止了连";
                break;
        case 2:
                msg = "连接失败";
                break;
        case 3:
                msg = "连接断开";
                break;
        case 4:
                msg = "端口冲突";
                break;
        case 5:
                msg = "分配内存失败";
                break;
        case 6:
                msg = "连接域名服务器失";
                break;
        case -102:
                msg = "用户名密码错";
```

```
                break;
        case -103:
                msg = "系统用户满员";
                break;
        case -105:
                msg = "通道用户满员";
                break;
        case -106:
                msg = "没有指定的通道";
                break;
        case -112:
                msg = "没有找到服务";
                break;
        default:
                msg = "未知";
        }

        cout << "showMsgLink=" << msg << endl;
}

void  setAlarmTempGlobal(unsigned long value)
{
        for (int i = 0; i < 7; i++)
                value = value >> 4;

        int alarmLevel = 0;
        switch (value)
        {
        case 0x01:
                alarmLevel = 0;
                break;
        case 0x02:
                alarmLevel = 1;
                break;
        case 0x04:
                alarmLevel = 2;
                break;
        case 0x08:
                alarmLevel = 3;
                break;
```

```
        }

        SYSTEMTIME st;
        GetLocalTime(&st);

        SYSTEM_INFO_LOG info;
        info.strMain = "global temp alarm";
        info.strChild = "global";
        info.strLevel.Format("%d level", alarmLevel);
        info.strTime.Format(_T("%04hu-%02hu-%02hu %02hu:%02hu:%02hu"), st.wYear, st.wMonth, st.wDay, st.wHour, st.wMinute, st.wSecond);

        //显示
}

void  setAlarmTempRegion(int value)
{
        int MAX_REGION = 6;/*区域上限暂为 6 个*/
        int areaindex = -10, idxtmp = value;
        for (int regidxcy = 0, regidxbd = MAX_REGION; regidxcy < regidxbd; ++regidxcy)
        {
                if (idxtmp &(0x1 << regidxcy))
                {
                        areaindex = regidxcy + 1;
                        break;
                }
        }

        unsigned long iValue = value;
        for (int i = 0; i < 7; i++)
                iValue = iValue >> 4;

        int alarmLevel = 0;
        switch (iValue)
        {
        case 0x01:
                alarmLevel = 0;
                break;
        case 0x02:
                alarmLevel = 1;
```

```
                break;
        case 0x04:
                alarmLevel = 2;
                break;
        case 0x08:
                alarmLevel = 3;
                break;
        }

        SYSTEMTIME st;
        GetLocalTime(&st);

        SYSTEM_INFO_LOG info;
        info.strMain = "region temp alarm";
        info.strChild.Format("region %d", areaindex);
        info.strLevel.Format("%d level", alarmLevel);
        info.strTime.Format(_T("%04hu-%02hu-%02hu %02hu:%02hu:%02hu"), st.wYear, st.wMonth, st.wDay, st.wHour, st.wMinute, st.wSecond);

        //显示
        cout << "====================" << endl
                << info.strMain << endl
                << info.strChild << endl
                << info.strLevel << endl
                << info.strTime << endl;
}

HANDLE _hMutex = CreateMutex(NULL, FALSE, NULL);
void WINAPI MessageCallBackReceive(IRNETHANDLE hHandle, WPARAM wParam, LPARAM lParam, void *context)
{
        //cout << "\n messageCallBackReceive:  wParam=" + std::to_string(wParam) + "    lParam=" + std::to_string(lParam) << endl;

        WaitForSingleObject(_hMutex, INFINITE);
        switch (_deviceType)
        {
        case DEVICE_TYPE_A:
        {
                switch (wParam)
```

```
                {
                case LAUMSG_LINKMSG:
                        showMsgLink(lParam);
                        break;
                case LAUMSG_ALARMMSG_GLOBAL_TEMP:
                        setAlarmTempGlobal(lParam);
                        break;
                case LAUMSG_ALARMMSG_REGION_TEMP:
                        setAlarmTempRegion(lParam);
                        break;
                }
                break;
        }
        case DEVICE_TYPE_B:
        {
                string msg = "";
                switch (wParam)
                {
                case MSG_INIT:
                        msg = (lParam) ? "decode ok!\n" : "decode fail!\n";
                        break;
                case MSG_PLAY:
                        msg = (lParam) ? "play video ok!\n" : "play video fail!\n";
                        break;
                }
                cout <<  msg << endl;
                break;
        }
        }
        ReleaseMutex(_hMutex);
}

void MessagCallBackRegion()
{
        //设备类型：A 类、B 类
        if (_deviceType != DEVICE_TYPE_A &&_deviceType != DEVICE_TYPE_B)
                return;

        SetMessageCallBack(_sdkHandle, MessageCallBackReceive, NULL);
}
```

```
int main()
{
	device_login();	//本方法及头文件、公用参数，参照 3.1 登录

	if (_isLogin)
	{
		MessagCallBackRegion();		//A 类、B 类

		if (_deviceType == DEVICE_TYPE_B)
			sdk_start_url(_sdkHandle, _deviceInfo.szIP);
	}

	cin.get();
	sdk_release(_sdkHandle);
	return 0;
}
```

3.3 Video Callback

```
int _width = 384;
int _height = 288;
void VideoCallBackReceive(char *pBuffer, long BufferLen, int width, int height, void* pContext)
{
	short videoData[1310720];//1280*1024*1.5  图像数据
	_width = width;
	_height = height;
	memcpy(videoData, pBuffer, width*height *1.5);

	cout << "VideoCallBackReceive:  _width=" << _width << "    _height=" << _height << end
l;
	//示例显示 100 个前数据
	for (int i = 0; i < 100; i++)
		cout << videoData[i] << ",";
	cout << endl;
}

void VideoCallBackRegion()
```

```
{
        //设备类型：A 类、B 类
        if (_deviceType != DEVICE_TYPE_A &&_deviceType != DEVICE_TYPE_B)
                return;

        int result = SetDeviceVideoCallBack(_sdkHandle, VideoCallBackReceive, NULL);
        if (result != SDK_RETURN_SUCCESS)
                cout << "设置【视频回调】失败!" << endl;
        else
                cout << "设置【视频回调】成功!" << endl;
 }

int main()
{
        device_login();      //本方法及头文件、公用参数，参照 3.1 登录

        if (_isLogin)
        {
                VideoCallBackRegion();                  //A 类、B 类

                if (_deviceType == DEVICE_TYPE_B)
                        sdk_start_url(_sdkHandle, _deviceInfo.szIP);
        }

        cin.get();
        sdk_release(_sdkHandle);
        return 0;
}
```

3.4 Temperature Callback

```
void TempCallBackReceive(char *pBuffer, long BufferLen, void* pContext)
{
        unsigned char _tempBuffer[1280 * 1024 * 2];
        unsigned short _temp_data[1280 * 1024];
        if (_deviceType == DEVICE_TYPE_B)
        {
                memcpy(_tempBuffer, pBuffer, BufferLen);
```

```
                    for (int ii = 0; ii < BufferLen / 4; ii++)    //数据转换
                    {

            _temp_data[ii * 2] = (unsigned short)((unsigned short)(_tempBuffer[ii * 2] << 8) + _tempBu
ffer[ii * 2 + 1 + _width * _height]);

            _temp_data[ii * 2 + 1] = (unsigned short)((unsigned short)(_tempBuffer[ii * 2 + 1] << 8) + _
tempBuffer[ii * 2 + (_width * _height)]);
                    }
            }
            else
                    memcpy(_temp_data, pBuffer, BufferLen);

            //示例显示 100 个前数据
            for (int i = 0; i < 100; i++)
                    cout << _temp_data[i] << ",";
            cout << endl;
}

void TempCallBackRegion()
{
            //设备类型：A 类、B 类
            if (_deviceType != DEVICE_TYPE_A &&_deviceType != DEVICE_TYPE_B)
                    return;

            int result = SetTempCallBack(_sdkHandle, TempCallBackReceive, NULL);
            if (result != SDK_RETURN_SUCCESS)
                    cout << "设置【温度回调】失败!" << endl;
            else
                    cout << "设置【温度回调】成功!" << endl;
}

int main()
{
            device_login(); //本方法及头文件、公用参数，参照 3.1 登录

            if (_isLogin)
            {
                    TempCallBackRegion();                    //A 类、B 类
```

```
                if (_deviceType == DEVICE_TYPE_B)
                        sdk_start_url(_sdkHandle, _deviceInfo.szIP);
        }

        cin.get();
        sdk_release(_sdkHandle);
        return 0;
}
```

3.5 Serial Port Callback

```
void SerialCallBackReceive(char *pRecvDataBuff, int BuffSize, void* pContext)
{
        if (BuffSize < 0)
        {
                cout << "SerialCallBackReceive：连接断开";
                return;
        }

        CString showData;
        int serialDataSize = BuffSize;
        unsigned char serialData[512];

        for (int i = 0; i < serialDataSize; ++i)
        {
                showData.AppendFormat(_T("%02X "), ((UCHAR*)pRecvDataBuff)[i]);
                serialData[i] = (unsigned char)pRecvDataBuff[i];
        }

        cout << " 数据大小=" + to_string(serialDataSize) << "\n 数据=" + showData << endl;
}

void SerialCallBackRegion()
{
        //设备类型：A 类
        if (_deviceType != DEVICE_TYPE_A)
                return;
```

```
        int result = SetSerialCallBack(_sdkHandle, _deviceInfo, SerialCallBackReceive, NULL);
        if (result != SDK_RETURN_SUCCESS)
                cout << "设置【串口透传回调】失败!" << endl;
        else
                cout << "设置【串口透传回调】成功!" << endl;
}

int main()
{
        device_login();  //本方法及头文件、公用参数，参照 3.1 登录

        if (_isLogin)
        {
                SerialCallBackRegion();                         //A 类

                //发送串口指令
                char sendCmd[] = { 0xAA, 0x04, 0x01, 0x70, 0x00, 0x1F, 0xEB, 0xAA };
                int length = sizeof(sendCmd);
                sdk_serial_cmd_send(_sdkHandle, sendCmd, length);
        }

        cin.get();
        sdk_release(_sdkHandle);
        return 0;
}
```

3.6 Alarm Callback

```
SYSTEM_INFO_LOG _alarmShow;
void AlarmCallBackReceive(char*  message, void *context)
{
        //解析报警信息
        Alarm_Info alarmInfo;
        int result = sdk_analyze_alarm_info(_sdkHandle, message, &alarmInfo);
        if (result != SDK_RETURN_SUCCESS)
        {
                cout << "【解析报警信息】失败!" << endl;
                return;
```

```
	}

	if (_deviceType == DEVICE_TYPE_B)//B 类
	{
		switch (alarmInfo.iType)
		{
		case 0://point
			_alarmShow.strMain = "Point temperature alarm";
			_alarmShow.strChild.Format("Point %d", alarmInfo.iIndex + 1);
			break;
		case 1://line
			_alarmShow.strMain = "Line temperature alarm";
			_alarmShow.strChild.Format("Line %d", alarmInfo.iIndex + 1);
			break;
		case 2://area
			_alarmShow.strMain = "Area temperature alarm";
			_alarmShow.strChild.Format("Area %d", alarmInfo.iIndex + 1);
			break;
		case 3://frame
			_alarmShow.strMain = "global alarm";
			_alarmShow.strChild = "global";
			break;
		}

		switch (alarmInfo.iAlarmType)
		{
		case 0://high

	_alarmShow.strLevel.Format("High temp alarm %d Level", alarmInfo.iLevel);
			break;
		case 1://low

	_alarmShow.strLevel.Format("Low temp alarm %d Level", alarmInfo.iLevel);
			break;
		}

		time_t now;
		now = alarmInfo.iTime;
		struct tm ltm;                                                    //tm
结构指针
```

```
            localtime_s(&ltm, &now);                    //获取当地日期和时间
            _alarmShow.strTime.Format(_T("%04hu-%02hu-%02hu %02hu:%02hu:%02hu"),
                    1900 + ltm.tm_year, 1 + ltm.tm_mon, ltm.tm_mday,
                    ltm.tm_hour, ltm.tm_min, ltm.tm_sec);
    }
    else//C 类
    {
            switch (alarmInfo.iType)
            {
            case 0://point
                    _alarmShow.strMain = "Area temperature alarm";
                    _alarmShow.strChild.Format("Area %d", alarmInfo.iIndex + 1);
                    break;
            case 1://line
                    _alarmShow.strMain = "Line temperature alarm";
                    _alarmShow.strChild.Format("Line %d", alarmInfo.iIndex + 1);
                    break;
            case 2://area
                    _alarmShow.strMain = "Point temperature alarm";
                    _alarmShow.strChild.Format("Point %d", alarmInfo.iIndex + 1);
                    break;
            }

            switch (alarmInfo.iAlarmType)
            {
            case 0:
                    _alarmShow.strLevel.Format("The Max is less than alarm");
                    break;
            case 1:
                    _alarmShow.strLevel.Format("The Max is more than alarm");
                    break;
            case 2:
                    _alarmShow.strLevel.Format("The Min is less than alarm");
                    break;
            case 3:
                    _alarmShow.strLevel.Format("The Min is more than alarm");
                    break;
            case 4:
                    _alarmShow.strLevel.Format("The Avg is less than alarm");
                    break;
```

```
                case 5:
                        _alarmShow.strLevel.Format("The Avg is more than alarm");
                        break;
                }

                _alarmShow.strTime.Format(_T("%s"), alarmInfo.alarmTime);
        }
}

void AlarmCallBackRegion()
{
        //设备类型：B 类、C 类
        if (_deviceType != DEVICE_TYPE_C && _deviceType != DEVICE_TYPE_B)
                return;

        int result = SetAlarmCallBack(_sdkHandle, _deviceInfo.szIP, AlarmCallBackReceive, NULL);
        if (result != SDK_RETURN_SUCCESS)
                cout << "设置【报警回调】失败!" << endl;
        else
                cout << "设置【报警回调】成功!" << endl;
}

int main()
{
        device_login(); //本方法及头文件、公用参数，参照 3.1 登录

        if (_isLogin)
        {
                AlarmCallBackRegion();                  //B 类、C 类

                if (_deviceType == DEVICE_TYPE_B)
                        sdk_start_url(_sdkHandle, _deviceInfo.szIP);
        }

        cin.get();
        sdk_release(_sdkHandle);
        return 0;
}
```

3.7 Snap Callback

```
void SnapCallBackReceive(int m_ch, char *pBuffer, int size, void *context)
{
	if (pBuffer)
	{
		char filePath[MAX_PATH + 25] = { 0 };
		GetModuleFileName(NULL, filePath, sizeof(filePath));

		SYSTEMTIME st;
		GetLocalTime(&st);

		char fileName[50] = { 0 };

	sprintf(fileName, "_%04hu%02hu%02hu%02hu%02hu%02hu%03hu.JPG", st.wYear, st.wMo
nth, st.wDay, st.wHour, st.wMinute, st.wSecond, st.wMilliseconds);
		strcat_s(filePath, fileName);
		FILE*pFile = fopen(filePath, "wb");
		if (!pFile) //打开文件失败
			return;

		if (!fwrite(pBuffer, size, 1, pFile)) //写文件失败
			cout << "【抓拍】保存文件失败！";
		else
			cout << "【抓拍】保存文件=" << filePath;
		fclose(pFile);
	}
}

void SnapCallBackRegion()
{
	//设备类型：A 类
	if (_deviceType != DEVICE_TYPE_A)
		return;

	int result = SetSnapCallBack(_sdkHandle, _deviceInfo, SnapCallBackReceive, NULL);
	if (result != SDK_RETURN_SUCCESS)
		cout << "设置【抓拍回调】失败!" << endl;
	else
```

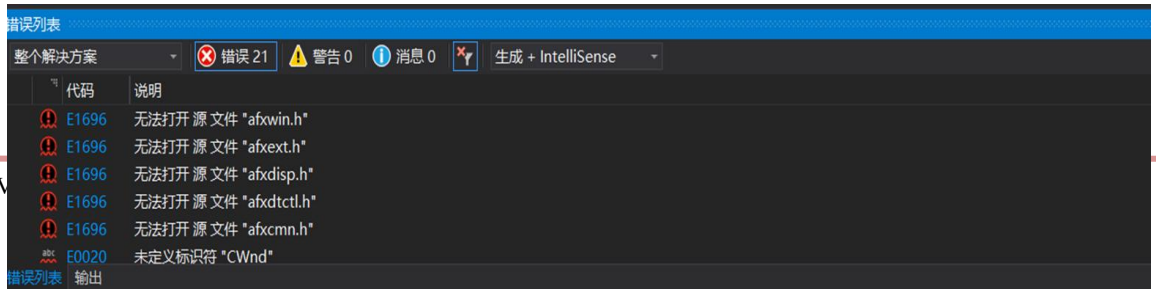
```
                cout << "设置【抓拍回调】成功!" << endl;
}

int main()
{
        device_login(); //本方法及头文件、公用参数，参照 3.1 登录

        if (_isLogin)
        {
                SnapCallBackRegion();                          //A 类

                //抓拍一次
                int imgType = 3;//3=jpg，4=jpg+irg
                if (sdk_set_capture_format(_sdkHandle, _deviceInfo, imgType) == 0) //设置抓拍格
式
                {
                        int result = sdk_CapSingle(_sdkHandle, _deviceInfo);
                        if (result != SDK_RETURN_SUCCESS)
                                cout << "【抓拍】失败!" << endl;
                        else
                                cout << "【抓拍】成功!" << endl;
                }
        }

        cin.get();
        sdk_release(_sdkHandle);
        return 0;
}
```

4. Common Problems

4.1 Can’t compile the demo with a version of VS higher than 2015

C++ module of VS should be completely installed.

Screenshot of the error.



4.2 Developed with QT, and a lot of type errors are reported

Include header file，#include<windows.h>

4.3 Developed with C# or JAVA, failed to call API

When defining the structure, be sure to keep the byte alignment with the SDK structure. For the specific definition method, please refer to the demo source code.

4.4 Always failed to send and receive serial port commands for Class A products (ATF series)

Register the serial port transparent transmission callback. Only Class A devices have this callback. For the registration method, please refer to section 3.5 or the demo source code.

4.5 Frequent calls to open and close the interface, resulting in a crash

There cannot be time-consuming operations in the callback function, and it is recommended to start the thread separately.

4.6 C++ Demo configure opencv

The demo uses opencv for imaging. Opencv is only a tool we choose for imaging. If not necessary, we can choose imaging tools freely. Our SDK provides original image

data in YUV420 format.

4.7 Report socket redefinition for QT accesss

Add DEFINES += WIN32_LEAN_AND_MEAN in pro file.