

Industrial Temperature Measurement SDK(Linux x64)

Quick Development Manual

1. Brief Introduction

This document aims to enable developers to quickly access the industrial temperature measurement SDK, and use the SDK to realize the functions of device images preview, temperature measurement, serial transmission and others. For specific implementation code, please refer to the Qt Demo attached to the SDK. For details about the API, see Industrial Temperature Measurement SDK User Manual.

2. SDK Contents

Catalog	Description
SDK/x64/includes	SDK header folder
SDK/x64/libs	SDK library folder
Demo/Qt/bin	Qt Demo executable program folder
Demo/Qt/InfraredTempSDK_Demo	Qt Demo source code

3. Compile Environment

3.1 Platform

Please use Ubuntu 18.04.1 and later version.

3.2 Compiler

Use gcc and g++ 9.3.0 and later version

3.3 Environment Variable Configuration

- 1) Put the library files in the SDK/x64/libs folder in the appropriate directory, e.g. /usr/local/lib/IRTInfraredSDK/
- 2) Add a configuration file under /etc/ld.so.conf.d/ directory with root permission, e.g. IRTInfraredSDK.conf, and the content is the directory path of the library file in 1), for example /usr/local/lib/IRTInfraredSDK/. Save the file.
- 3) Run the ldconfig command with root permission to make the configuration take effect.

Compile and install libjpeg turbo library. Take libjpeg-turbo-1.5.2.tar.gz as an example

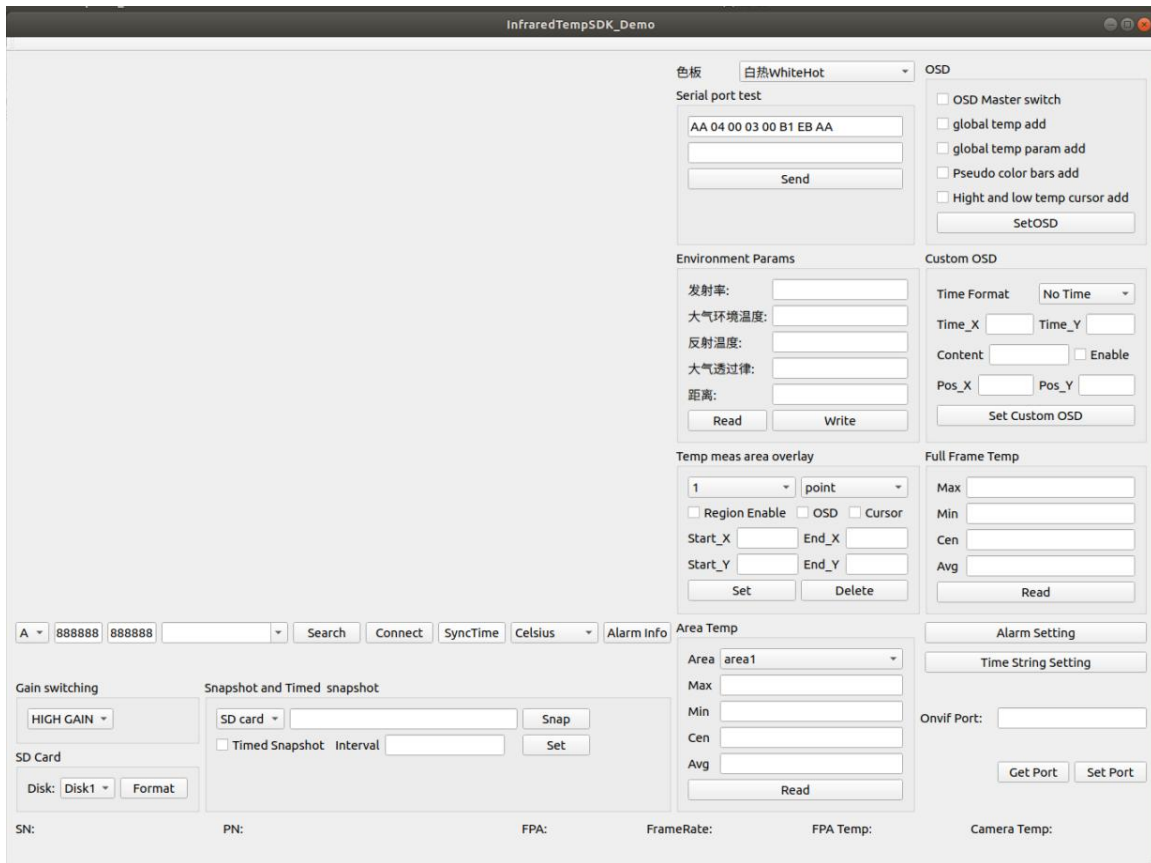
1. Install and compile dependent packages \$ sudo apt-get install nasm
2. Move it to the source file directory and execute the autoreconf -fiv command
\$ cd libjpeg-turbo \$ autoreconf -fiv
3. Run the configure script and add the configuration parameters \$ sudo rm -rf /usr/local/lib/libjpeg.so* \$./configure --prefix=/usr/local --mandir=/usr/local/share/man --with-jpeg8 --disable-static --docdir=/usr/local/share/doc/libjpeg-turbo-1.5.2
4. Compile and install \$make \$sudo make install
- 5) Install the vlc related library for imaging. The reference command is as follows:
\$ sudo apt-get install vlc libvlc-dev

4. SDK Qt Demo Introduction

4.1 Demo Instructions

SDK Qt Demo is a sample program developed with Qt to show how to use the SDK. Based on Qt 5.12.8, Demo has functions such as device search, device connection, image preview, temperature data acquisition and calculation.

4.2 Demo Functions



4.2.1 SDK Initialization

1) First select the device type (A/B/C) and enter the user name and password. The demo implements the function of switching device types and filling in the default user name and password. Call the interface `sdk_set_type` to set the above information. The code in Demo is implemented as follows.

```
m_deviceType = ui->comboBoxDeviceType->currentIndex();  
m_strUsrName = ui->lineEditUsrName->text();  
m_strPwd = ui->lineEditPwd->text();  
const char* pUsrName = m_strUsrName.toStdString().c_str();  
const int lenUsrName = strlen(pUsrName);  
char tmpUsrName[lenUsrName] = {0};  
memcpy(tmpUsrName, pUsrName, lenUsrName);  
const char* pUsrPwd = m_strUsrName.toStdString().c_str();
```

```

const int lenUsrPwd = strlen(pUsrPwd);

char tmpUsrPwd[lenUsrPwd] = {0};

memcpy(tmpUsrPwd, pUsrPwd, lenUsrPwd);

sdk_set_type(m_deviceType, tmpUsrName, tmpUsrPwd);

```

- 2) Call the interface `sdk_initialize` to initialize the SDK. After initialization, you can add an appropriate delay to ensure that initialization is complete as much as possible.
- 3) Call the interface `sdk_create` to create an SDK instance. Many subsequent operations need to be based on this instance.

4.2.2 Search Device

Call interface `sdk_search_device` to search for devices. The code in Demo is as follows.

```

if (NULL != pHandle) {
    m_deviceList = {0};
    ui->comboBoxIPAddr->clear();

    if (sdk_search_device(pHandle, m_deviceList) == -1) {
        QMessageBox::information(NULL, "Error", QString("Search device failed"), QMessageBox::Ok);
        return;
    }
    else {
        for(int i = 0; i < m_deviceList.iNumber; i++)
        {
            qDebug("%s\n", m_deviceList.DevInfo[i].szIP);
            ui->comboBoxIPAddr->addItem(QString("%1").arg(m_deviceList.DevInfo[i].szIP));
        }
    }
}

```

4.2.3 Device Connection

SDK does not provide a separate interface for connecting devices. The device connection function in Demo is mainly to register callback function, call preview image function, and obtain device SN, PN and other parameters. Note that different types of devices require different registration callback functions. Class A devices need to call the registered serial port transparent transmission callback function SetSerialCallBack. See specific code in Demo regCallbackAndPreview.

4.2.4 Image Preview

The way to realize image preview in Demo is to play the rtsp video stream output by the device with vlc. It can be done in other ways.

- 1) If vlc related libraries are not installed, install them. For details, see the following command.

```
$ sudo apt-get install vlc libvlc-dev
```

- 2) Determine the url of rtsp video stream according to the device type and ip address. The code in Demo is implemented as follows.

```
QString url;
switch (m_deviceType) {
    case DeviceTypeA:
        url = QString("rtsp://%1:554/stream0").arg(QLatin1String(m_ip));
        break;
    case DeviceTypeB:
        url = QString("rtsp://%1/cam/realmonitor?channel=1&subtype=0").arg(QLatin1String(m_ip));
        break;
    case DeviceTypeC:
        url = QString("rtsp://%1:9554/live?channel=1&subtype=0").arg(QLatin1String(m_ip));
        break;
}
```

- 3) Call the vlc related interface to play the rtsp video stream. Please refer to doPreview in Demo for details.

4.2.5 Temperature Data Acquisition and Calculation

- 1) Register the temperature data callback function by using the interface SetTempCallBack, read the temperature data in the callback function, and calculate.
- 2) The temperature conversion formula is different for industrial temperature measurement and skin temperature measurement. The default conversion method in Demo is for industrial temperature measurement.

	Industrial temperature measurement	Industrial temperature measurement	Skin temperature measurement
Data interval(Value)	0~7300	7301~16383	0~16383
Temperature conversion formula (Kelvin → Celsius)	$(\text{Value} + 7000) / 30 - 273.2$	$(\text{Value} - 3300) / 15 - 273.2$	$(\text{Value} + 25000) / 100 - 273.2$

4.2.6 Gain Switching

Refer to Gain switching in Demo to switch between high and low gain. Do not switch gains when using the skin temperature measuring modules. Display high gain by default. For industrial temperature measurement, high, low and automatic gain can be adjusted.

4.2.7 Set Palette

Refer to the color palette in Demo and call the interface sdk_set_color_plate to switch the palette.

4.2.8 Other Functions

For other functions, please refer to the code in Demo and Industrial Temperature Measurement SDK User Manual. After the image and temperature are sorted out, it is easier to sort out the business of other interfaces.

5. Q&A

1. Q: Class A products (ATF series) always fail to send and receive serial port commands.

A: Register the serial port transparent callback. Only Class A devices have this callback. Refer to the demo source code for the registration method.

2. Q: Frequently call to open or close the interface, resulting in a crash.

A: The callback function cannot contain time-consuming operations. It is recommended to start the thread separately.

3. Q: How to parse the IRG format?

A: Header (8Byte+128Byte)+ grayscale image (1Byte per pixel, array size * 1)+temperature data (2Byte per pixel, low byte first, array size * 2).