



SDK User Manual for AT&ATF(C++)

IRay Technology Co., Ltd.

www.iraytek.com

Table of Contents

1. Manual Instruction.....	- 1 -
2. Compilation Environment.....	- 1 -
3. SDK Operation Instruction.....	- 1 -
4. Programming Instruction.....	- 6 -
5. Preview Images.....	- 6 -
6. Measuring Temperature.....	- 7 -
7. Serial Port Transmission.....	- 8 -
8. Other Functions and Interfaces.....	- 9 -

1. Manual Instruction

This SDK user manual aims to enable developers to quickly access our SDK, and use the SDK to preview device images, measure temperature, and realize serial transmission and other functions.

2. Compilation Environment

Platform: Windows 10 64-bit system

Compilation environment: Visual Studio 2015 Professional. When the SDK source code is opened with VS2015, the default compilation mode is: Debug + X86.

First, you need to configure OpenCV. For specific configuration methods, please refer to <https://blog.csdn.net/qq114480/article/details/105690376>

3. SDK Operation Instruction

Open the SDK tool → the default device type is: A, both the user name and password are 888888.

Among them:

Class A: LT, ATF series, both user name and password are 888888;

Class B: AT series, both user name and password are admin;

Class C: AT20 series, both user name and password are admin;

Click the Search button at the bottom of the panel to check the IP address of the device under the same network segment. Based on the IP information displayed in the selection box, you can perform the following operations:

Note: SDK can only operate devices under the same network segment.

The devices in the network environment must be searched through the Search button before the next operation can be performed.

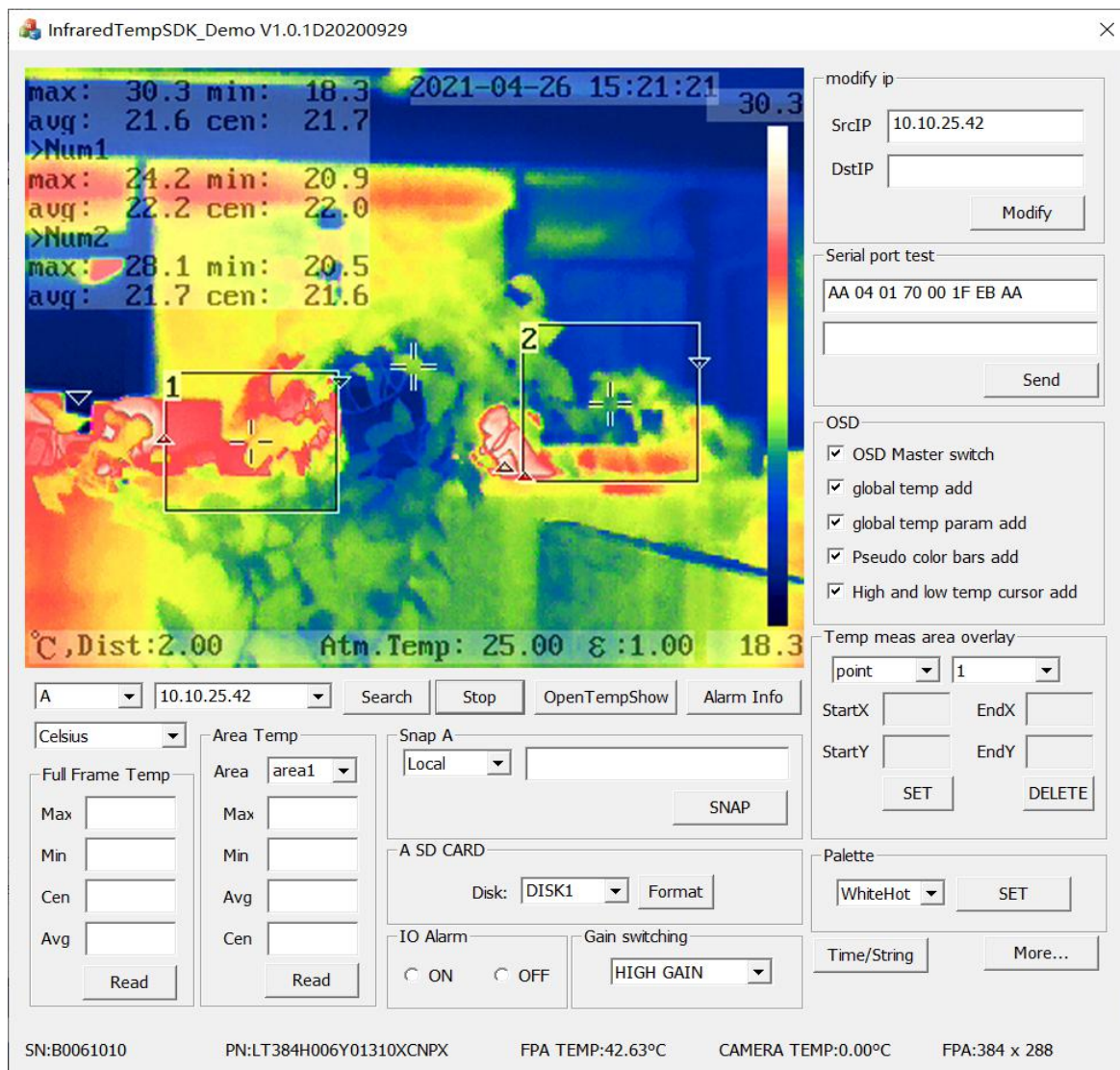


Figure 1

1. Click the “Preview” button, the image information of the device can be imaged through “Opencv”.
2. Click the “OpenTempShow” button to display temperature information with the mouse. As shown in Figure 2:

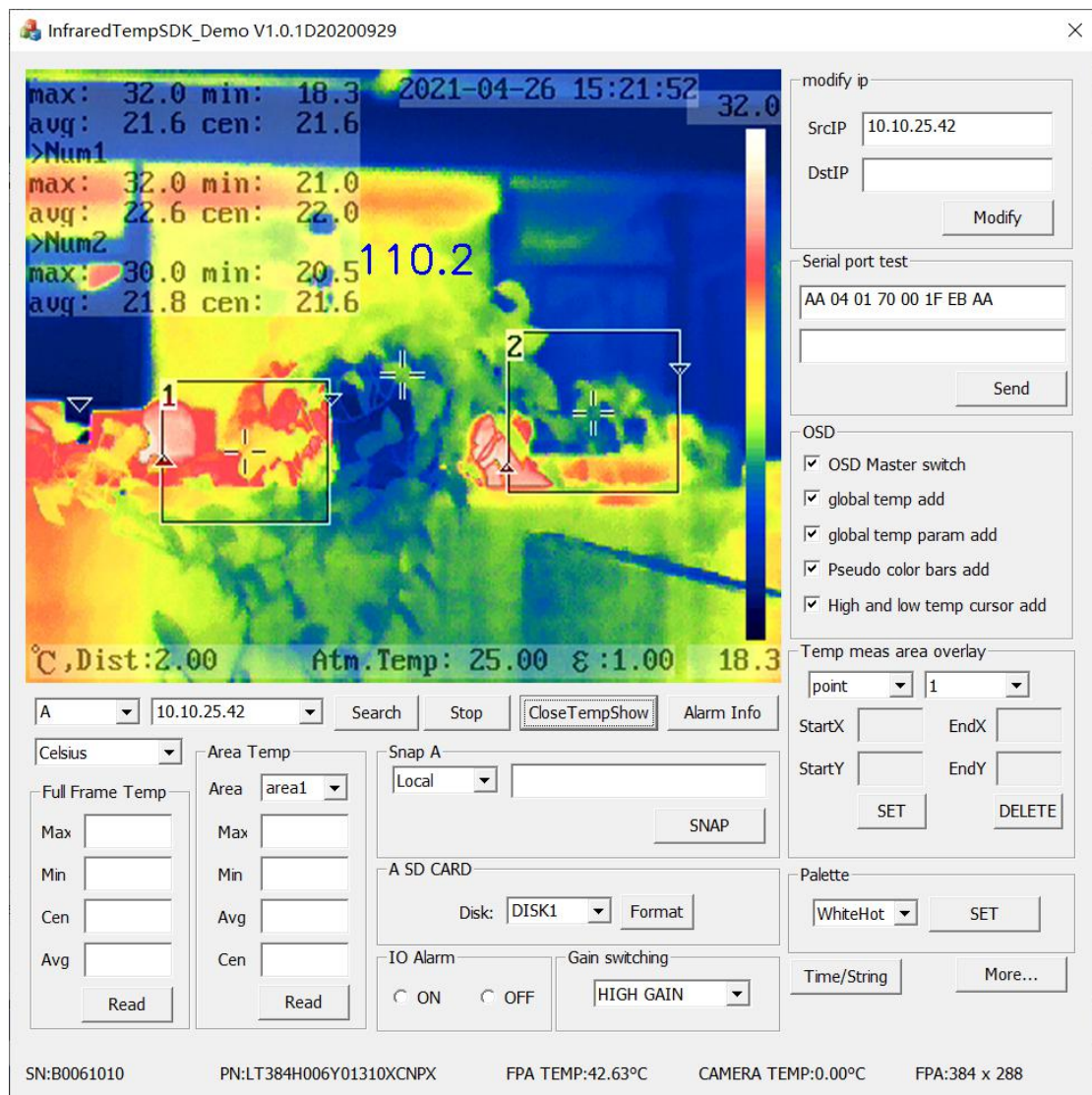


Figure 2

The conversion formula of temperature is different for industrial temperature measurement and skin temperature measurement. The default SDK is the conversion mode of industrial temperature measurement. If you are using a device for skin temperature measurement, you will need to refer to the formula in figure 3 for replacement. Refer to figure 4 for the specific interface code that needs to be replaced.

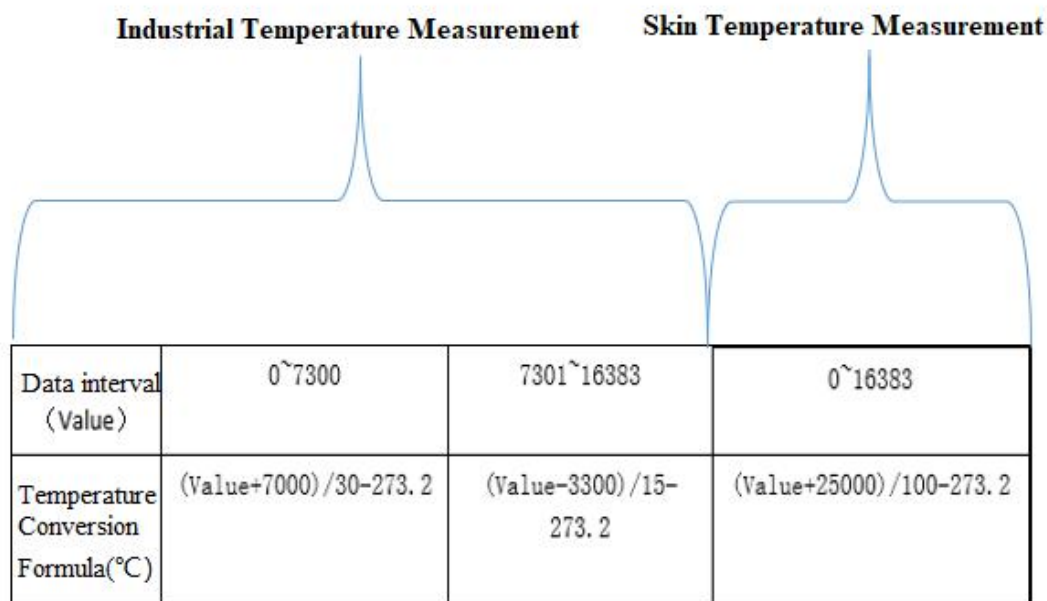


Figure 3

```

private unsafe void TempCallBackFunc(ref byte pData, int size)
{
    if (size == 384 * 288 * 2)
    {
        Width = 384;
        Height = 288;
    }
    else if (size == 640 * 512 * 2)
    {
        Width = 640;
        Height = 512;
    }
    byte[] buf = new byte[size];
    unsafe
    {
        fixed (byte* p = &pData)
        {
            using (UnmanagedMemoryStream ms = new UnmanagedMemoryStream((byte*)p, size))
            {
                ms.Read(buf, 0, buf.Length);
            }
        }
    }
    mt.WaitOne();
    if (iDeviceType == 1)
    {
        Buffer.BlockCopy(buf, 0, temp_data, 0, size);

        for (int ii = 0; ii < Width * Height / 2; ii++)
        {
            data_save[ii * 2] = (UInt16)((UInt16)(temp_data[ii * 2] << 8) + temp_data[ii * 2 + 1 + Width * Height]);
            data_save[ii * 2 + 1] = (UInt16)((UInt16)(temp_data[ii * 2 + 1] << 8) + temp_data[ii * 2 + (Width * Height)]);
        }
    }
    else
    {
        // Console.WriteLine("iDeviceType=Other.");
        Buffer.BlockCopy(buf, 0, data_save, 0, size);
    }

    for (int j = 0; j < Height; j++)
    {
        for (int i = 0; i < Width; i++)
        {
            //判断温度区间0~7300, 则温度换算公式(换算为摄氏度) (Value+7000)/30-273.2
            //7301~16383, (Value-3300)/15-273.2

            if (data_save[j * Width + i] > 7300) //7301~16383, (Value-3300)/15-273.2
            {
                fTempValue = 15.0f;
                data_show[j * Width + i] = (UInt16)(data_save[j * Width + i] - 3300);
            }
            else //0~7300, 则温度换算公式(换算为摄氏度) (Value + 7000) / 30 - 273.2
            {
                fTempValue = 30.0f;
                data_show[j * Width + i] = (UInt16)(data_save[j * Width + i] + 7000);
            }
        }
    }
    mt.ReleaseMutex();
}

1 个引用
private unsafe void SerialCallBackFunc(ref byte pData, int BufferLen)

```

Figure 4

The following operations can be performed based on the temperature information:

3. Through the Read button of the “Full Frame Temp” function, you can get the temperature information of a full frame, including Max (the highest temperature), Min (the lowest temperature), Cen (temperature at the center point), and Avg (average temperature).

Through the “Area Temp” function, the temperature information of points, lines, and frames can be obtained respectively, including Max (the highest temperature), Min (the lowest temperature), Cen (temperature at the center point), and Avg (average temperature). Among them, the switch of point, line and area is set in the “Temp meas area overlay” module.

4. Modify the IP address through “modify ip”. “SrcIP”: Source IP address; “DstIP”: the new IP address to be modified.

5. “Serial port test”: send and receive serial port commands.

6. “OSD” function: superimpose and delete OSD information.

7. “Palette” function: switch color palette.

8. The alarm information can be viewed through the “AlarmInfo” button.

9. The “Snap A” can realize the function of taking pictures, which can be stored to the local disk and SD card respectively.

The stored picture formats are *.irg, *.jpg, *.dat. Among them *.irg contains temperature information.

10. The “IO Alarm” module can output peripheral alarm information for thermal cameras with input and output functions.

11. “Gain Switching” function can switch between high and low gain.

When using a skin temperature measuring thermal module, do not adjust the high and low gains. Let it display high gain by default.

For industrial temperature measurement: high, low, and automatic gain adjustments can be performed.

4. Programming Instruction

The SDK uses a multi-threaded communication internally, and all exception notifications are fed back to the caller through a callback function. Do not call the interface that frees the resource in the callback function to avoid internal deadlock or exception.

5. Preview Images

Interface call flow diagram

First: Initialize the search button function: `sdk_set_type()` -> `sdk_initialize()` -> `sdk_search_device()`

Then: Preview to get data: reference code. This step executes the registration information callback, the registered image data callback, and the registered temperature callback.


```

1  if (strConn == "Preview")
    {
        cCInfraredTempSDK.strDeviceIP = strDeviceIP.GetBuffer();
        strDeviceIP.ReleaseBuffer();

        cCInfraredTempSDK.SetDeviceVideoCallBack(strDeviceIP.GetBuffer(), VideoCallBack, this);
        strDeviceIP.ReleaseBuffer();

        cCInfraredTempSDK.SetTempCallBack(strDeviceIP.GetBuffer(), TempCallBack, this);
        strDeviceIP.ReleaseBuffer();

        cCInfraredTempSDK.SetMessageCallBack(MessageCallBack, this);

        int iOSD = 0;
        INF_OSD osd_info = INF_OSD(0, 0, 0, 0);
        cCInfraredTempSDK.sdk.LoadParamOsd(devInfo, &iOSD, &osd_info);
    }

```

Finally, the callback image and temperature data are respectively parsed through the following two callback functions.

```

void CInfraredTempSDK_DemoDlg::VideoCallBack(char *pBuffer, long BufferLen, int width, int height)
{
    CInfraredTempSDK_DemoDlg* pDlg = (CInfraredTempSDK_DemoDlg*)AfxGetApp()->GetMainWnd();

    Width = width;
    Height = height;

    memcpy(pBufShow, pBuffer, Width*Height *1.5);
    ::SendMessage(m_AppShow, UM_SHOW_VIDEO, 0, LPARAM(pBufShow));
}

void CInfraredTempSDK_DemoDlg::TempCallBack(char *pBuffer, long BufferLen)
{
    if (iDeviceType == 1)
    {
        memcpy(&temp_buffer[0], pBuffer, Width * Height * 2);

        for (int ii = 0; ii < Width * Height / 2; ii++) //数据转换
        {
            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)]);
        }
    }
    else
    {
        memcpy(&temp_data_temp[0], pBuffer, Width * Height * 2);
    }

    for (int j = 0; j < Height; j++)
    {
        for (int i = 0; i < Width; i++)
        {

```

6. Measuring Temperature

The effect of temperature measurement is to return the temperature values of all pixels in each frame of the image, that is, a temperature matrix. What you get from the callback function is the temperature data matrix of the full frame, and every two bytes represents the temperature of a pixel (this temperature is

Kelvin*10). The specific conversion code for Celsius is as follows:

```
for (int j = 0; j < Height; j++)
{
    for (int i = 0; i < Width; i++)
    {
        //判断温度区间0~7300, 则温度换算公式(换算为摄氏度) (Value+7000)/30-273.2
        //7301~16383, (Value-3300)/15-273.2
        if (iTEn&&bIfNewNios)
        {
            if (temp_data_temp[j * Width + i] > 7300) //7301~16383, (Value-3300)/15-273.2
            {
                fTempValue = 15.0;
                temp_data_temp[j * Width + i] = temp_data_temp[j * Width + i] - 3300;
            }
            else //0~7300, 则温度换算公式(换算为摄氏度) (Value + 7000) / 30 - 273.2
            {
                fTempValue = 30.0;
                temp_data_temp[j * Width + i] = temp_data_temp[j * Width + i] + 7000;
            }
        }
        else
        {
            fTempValue = 10.0;
        }
    }
}
```

data interval (value)	0~7300	7301~16383
temperature conversion formula(°C)	$(\text{Value}+7000)/30-273.2$	$(\text{Value}-3300)/15-273.2$

7. Serial Port Transmission

Class A needs to register the serial port transmission interface, and Class B/C does not need to perform the registration serial port transmission.

```
if (iDeviceType == 0) //判断A类设备
{
    checkBoxGlobalTempAdd.Enabled = true;
    checkBoxGlobalTempParamAdd.Enabled = true;
    checkBoxPseudoColorAdd.Enabled = true;
    checkBoxHighLowPosAdd.Enabled = true;
    SetSerialCallBack(_ip, serialcallback, this.Handle);

    checkBoxOSDSwitch.Checked = Convert.ToBoolean(iOSD);
    checkBoxGlobalTempAdd.Checked = osd_info.m_g_temp_add;
    checkBoxGlobalTempParamAdd.Checked = osd_info.m_g_temp_param_add;
    checkBoxPseudoColorAdd.Checked = osd_info.m_pseudo_add;
    checkBoxHighLowPosAdd.Checked = osd_info.m_low_high_temp_pos;
}
else
```

8. Other Functions and Interfaces

Please refer to the industrial temperature measurement SDK user manual and SDK Demo code for study. After the process of temperature and image is sorted out, it is easier to sort out the process of other interfaces.