**USB接口软件开发资料说明**

**USB Interface Development Instructions**

目录

[SDK构成The SDK components 2](#_Toc114750969)

[Libir\_sample源码The Libir\_sample source code 2](#_Toc114750970)

[模组SDK接口V1.5.xlsx Interface Description 5](#_Toc114750971)

[用户开发标定User calibration instructions 5](#_Toc114750972)

[开发使用前提Premise 5](#_Toc114750973)

[获取数据流Get data stream 5](#_Toc114750974)

[UVC协议介绍UVC protocol 5](#_Toc114750975)

[Libiruvc库接口介绍Libiruvc library interface introduction 8](#_Toc114750976)

[不同模式数据流Different data flows 8](#_Toc114750977)

[USB控制接口说明USB control description 9](#_Toc114750978)

[1.概述overview 9](#_Toc114750979)

[2.特殊命令接口Special command interface 9](#_Toc114750980)

[3. 其他命令接口Other command interface 10](#_Toc114750981)

[SDK其他功能介绍Other Functions 10](#_Toc114750982)

[固件升级功能Firmware Upgrade 10](#_Toc114750983)

[测温二次标定Secondary calibration 10](#_Toc114750984)

[盲元标定Dead pixel correction 10](#_Toc114750985)

[测温环境变量修正功能Ambient variable correction 11](#_Toc114750986)

[点线框测温说明Description of point/line/box temperature measurement 11](#_Toc114750987)

[其他库的功能介绍Other Functions 12](#_Toc114750988)

[数据格式说明Data Format Description 12](#_Toc114750989)

[1.图像数据Image data 12](#_Toc114750990)

[2.温度数据Temperature data 12](#_Toc114750991)

该版本的SDK主要针对ASIC\_384\_640及G\_1280系列产品在linux平台下通过USB接口进行图像/温度数据传输和命令控制。该SDK的例程已经在树莓派4B平台开发板实现，客户可参考该开发例程进行应用层的修改。

This version of SDK mainly aims at image/temperature data transmission and command control through USB interface for ASIC\_384\_640 and G\_1280 series products on Linux platform. The SDK routines have been implemented in the Raspberry PI 4B platform development board, and customers can refer to the development routines to modify the application layer.

## SDK构成The SDK components

### Libir\_sample源码The Libir\_sample source code

图示, 文本

描述已自动生成

#### Update\_fw\_sample源码The Update\_fw\_samplesource code

1. sample.cpp: 包含sample的主函数，主要包含用户命令注册，打印等级设置，固件更新等功能。

sample.cpp: The main function of sample application, mainly including user command registration, print level setting, and update firmware.

1. readme.md: sample软件框架的说明。

readme. md: Description of the sample software framework in Chinese.

1. readme\_EN.md：sample软件框架的英文说明。

Readme\_en. md: sample Software framework description in English

1. CMakeLists.txt：客户可根据该文件利用cmake工具对sample进行编译。

CMakeLists.txt：From this file, customers can compile sample using the cmake tool.

#### Temp\_sample源码The Temp\_sample source code

通过使用libirtemp库进行测温及温度距离修正

By using the LibirTemp library for temperature measurement and temperature distance correction

1. sample.cpp: 包含sample的主函数，主要包含用户命令注册，打印等级设置，出图及进行测温显示和分析等功能。

sample.cpp: The main function of sample application, mainly including user command registration, print level setting, display the image, and display and analyze temperature measurement.

1. readme.md: sample软件框架的说明。

readme. md: Description of the sample software framework in Chinese.

1. readme\_EN.md：sample软件框架的英文说明。

Readme\_en. md: sample Software framework description in English

1. CMakeLists.txt：客户可根据该文件利用cmake工具对sample进行编译。

CMakeLists.txt：From this file, customers can compile sample using the cmake tool.

#### Cmd\_sample及double\_camera\_sample源码The Cmd \_sample and double\_camera\_sample source code

Cmd\_sample及double\_camera\_sample分别指单USB及多USB示例

Cmd\_sample and double\_camerA\_sample indicate single-USB and multi-USB sample

1. sample.cpp: 包含sample的主函数，主要包含用户命令注册，打印等级设置，可设置数据采集的分辨率/帧率，以及伪彩/翻转/镜像/旋转/输出格式等功能。

sample.cpp: The main function of sample application, mainly including user command registration, print level setting, data collection resolution/frame rate setting, and pseudo-color/flip/mirror/rotate/output format.

1. cmd.cpp:主要包含用户发送命令的各种demo。

cmd.cpp: mainly contains various demos of commands sent by users.

1. display.cpp:获取图像帧信息之后，根据之前frame\_info里参数的设定，做图像数据格式转换、翻转/镜像、旋转等处理，最后调用opencv显示出来图像。

display.cpp: after obtaining image frame information, according to the Settings of parameters in the previous frame\_info, image data format conversion, flip/mirror, rotation and other processing, and finally call OpencV to display the image.

1. camera.cpp:用于获取红外模组数据流，当stream线程获取到原始红外帧信息的时候，会将红外帧信息raw frame切分为图像信息image frame和温度信息temp frame，并发送信号，传递给对应的模块做相应的处理，当image frame和temp frame处理完成后发送信号给camera线程，camera线程继续下一次循环。

camera. cpp: used to obtain infrared module data flow. When the stream thread obtains the original infrared frame information, it will slice the raw frame information into image frame information and temperature information temp frame, and send signals to the corresponding module for corresponding processing. When the image frame and Temp frame processing are completed, signals are sent to the camera thread, which continues the next loop.

1. readme.md: sample软件框架的说明。

readme. md: Description of the sample software framework in Chinese.

1. readme\_EN.md：sample软件框架的英文说明。

Readme\_en. md: sample Software framework description in English

1. sample.cpp: 包含sample的主函数，主要包含用户命令注册，打印等级设置，可设置数据采集的分辨率/帧率，以及伪彩/翻转/镜像/旋转/输出格式等功能。

sample.cpp: The main function of sample application, mainly including user command registration, print level setting, data collection resolution/frame rate setting, and pseudo-color/flip/mirror/rotate/output format.

1. cmd.cpp:主要包含用户发送命令的各种demo。

cmd.cpp: mainly contains various demos of commands sent by users.

1. display.cpp:获取图像帧信息之后，根据之前frame\_info里参数的设定，做图像数据格式转换、翻转/镜像、旋转等处理，最后调用opencv显示出来图像。

display.cpp: after obtaining image frame information, according to the Settings of parameters in the previous frame\_info, image data format conversion, flip/mirror, rotation and other processing, and finally call OpencV to display the image.

1. camera.cpp:用于获取红外模组数据流，当stream线程获取到原始红外帧信息的时候，会将红外帧信息raw frame切分为图像信息image frame和温度信息temp frame，并发送信号，传递给对应的模块做相应的处理，当image frame和temp frame处理完成后发送信号给camera线程，camera线程继续下一次循环。

camera. cpp: used to obtain infrared module data flow. When the stream thread obtains the original infrared frame information, it will slice the raw frame information into image frame information and temperature information temp frame, and send signals to the corresponding module for corresponding processing. When the image frame and Temp frame processing are completed, signals are sent to the camera thread, which continues the next loop.

1. readme.md: sample软件框架的说明。

readme. md: Description of the sample software framework in Chinese.

1. readme\_EN.md：sample软件框架的英文说明。

Readme\_en. md: sample Software framework description in English

1. irsample\_structure.png：sample软件框架图解。客户可参考该框架进行开发。

irsample\_structure.png：Sample Software framework diagram. Customers can refer to this framework for development.

1. CMakeLists.txt：客户可根据该文件利用cmake工具对sample进行编译。

CMakeLists.txt：From this file, customers can compile sample using the cmake tool.

#### libs

linux下的动静态库、windows下64位的静态库、include：libir SDK的所有头文件

Dynamic and static libraries for Linux, 64-bit static libraries for Windows, include: all header files for the libir SDK

#### Win\_include

windows下openCV及pthread等标准库的头文件

Windows openCV and pthread standard library header files

#### x64

windows下64位的动态库、各个实例的应用程序

Windows 64-bit dynamic library, each sample of the application

#### libir\_sample.sln

windows下的sln工程

sln project in Windows

#### sample\_version.h

sample示例程序的版本号

Sample Version number of the sample program

### 模组SDK接口V1.5.xlsx Interface Description

概要说明SDK各个库中的函数接口，方便用户大致了解SDK所支持的功能。

This section describes the function interfaces in each SDK library to help users understand the functions supported by the SDK

### 用户开发标定User calibration instructions

包含盲元标定和测温标定的说明，以及温度的环境变量修正。

Contains the description of Secondary calibration, Dead pixel correction, Ambient variables correction

## 开发使用前提Premise

1. 客户的开发板或者目标平台芯片支持USB接口。

Customer's development board or platform supports USB interface

1. 客户开发平台最好能支持串口调试。(非必要)

Customer development platform should support serial port debugging. (Not necessary)

1. 该SDK例程采用opencv显示图像，客户平台如果需要跑sample则需要编译安装opencv库。

This SDK routine uses OpencV to display images. If the customer platform needs to run sample, it needs to compile and install opencV library.

## 获取数据流Get data stream

### UVC协议介绍UVC protocol

UVC是属于USB行业规范中的USB设备类规范，用于USB接口的视频设备的一个统一的数据交换规范。其中UVC官方协议文档 UVC 1.5 Class specification是对整个UVC协议的框架做了规范，USB\_Video\_Example则规定了描述符等的数据结构和相关的一些其他相关规定。在UVC规范中明确要求，具有实际UVC功能的视频设备只要需要两种接口：一种为VC Interface(视频控制接口)，VS Interface(视频流接口)。其中VC接口用于对UVC设备进行配置操控，而VS接口则用于负责传输视频数据流，两者相互配合完成UVC设备功能。

UVC is a unified data exchange specification for VIDEO devices with USB interface, which belongs to the USB device category specification in the USB industry specification. UVC 1.5 Class Specification is the official UVC protocol document. USB\_Video\_Example defines the data structure of descriptors and other relevant provisions. In the UVC specification, it is clearly required that the video equipment with actual UVC function only needs two interfaces: VC Interface(video control Interface) and VS Interface(video streaming Interface). The VC interface is used to configure and control the UVC device, while the VS interface is used to transmit video data flow. The two interfaces cooperate to complete the functions of the UVC device.

* **Video Control Interface（视频控制接口）**

VC主要用于控制UVC功能中的Units和Terminals，通过对不同的Units和Terminals下发不同的指令，从而影响UVC设备的当前工作状态;Host端下发的所有对UVC设备的操纵控制命令都是经由此Interface完成的。以上面拓扑图所表示的设备为例：

该设备包含了一个Processing Unit、一个Input Terminal、Output Terminal和Selector Unit，VC Interface需要对它们下发不同的命令，为此VC Interface需要包含一个控制端点(强制性要求)，它使用的就是每个USB设备中默认的端点0。另外一个Interrupt端点则是可选的，用来返回或通知Host端当前的UVC设备内部状态有变化。大部分情况下，一个UVC设备的VC Interface不需一定要实现此端点，但一旦UVC设备需要实现某些特定feature时，Spec会强制性要求实现该interrupt端点。

VC controls Units and Terminals in the UVC. You can send different commands to different Units and Terminals to affect the current operating status of the UVC. All the control commands on the UVC device delivered by the Host end are completed through this Interface. Take the devices in the topology as an example:

The device contains a Processing Unit, an Input Terminal, an Output Terminal and a Selector Unit. VC Interface needs to deliver different commands to them. For this, the VC Interface needs to include a control endpoint (mandatory) that uses the default endpoint 0 on each USB device. The other Interrupt endpoint is optional and is used to return or notify the Host of a change in the internal state of the current UVC device. In most cases, the VC Interface of a UVC device is not required to implement this endpoint, but when the UVC device needs to implement some specific feature, the Spec will make it mandatory to implement this interrupt endpoint.

* **Video Streaming Interface（视频流接口）**

VS Interface则专注与负责传输UVC设备的Video数据到Host端。若一种UVC设备支持n种格式的Video数据，则它需要实现n个VS Interface，每个Interface对应一种专门的数据格式;

VS Interface focuses on and is responsible for transmitting Video data of UVC devices to the Host end. If a UVC device supports N formats of Video data, it needs to implement N VS interfaces, and each Interface corresponds to a special data format.

* **Descriptor（描述符）**

描述符用来描述USB设备性能或特向的数据结构，与设备类相关的信息都是主机向设备获取描述符来得到的。在UVC所规定的描述符中除了常用的标准描述符，另外还定义了视频设备的特殊类描述符，主要如下：

Descriptors are used to describe the performance or specific data structure of USB devices. Information related to device classes is obtained by the host from the descriptors of devices. In addition to common standard descriptors, special class descriptors of video equipment are also defined in the descriptors stipulated by UVC, which are mainly as follows:

**标准描述符Standard descriptor**

设备描述符（Device Descriptor）

设备限定描述符（Device Qualifier Descriptor）

设备配置描述符（Configure Descriptor）

其他速度描述符（Other Speed Descriptor）

字符描述符​（String Descriptor）

**特殊类描述符Special class descriptor**

接口联合描述符（Interface Association Descriptor）

视频控制接口描述符（Video Control Interface Descriptor）

视频控制端点描述符（Video Control Endpoint Descriptor）

视频流接口描述符（VideoStreaming Interface Descriptor）

视频流端点描述符（VideoStreaming Endpoint Descriptor）

* **Request （请求）**

请求是主机软件与实体中控制器交流的一种机制。请求都是由主机向设备端发起的功能请求，所有的USB设备都需要支持标准的设备请求和与Class相关的特殊类请求。

标准请求：在标准协议中有相关描述，不做详细展开

特殊类请求“事实上特殊类请求也可以根据接口来分成两大类：1）、视频控制请求，2）视频流请求。

Requests are a mechanism by which the host software communicates with the controller in the entity. Requests are function requests initiated by the host to the device. All USB devices must support standard device requests and class-related special requests.

Standard request: a standard request is described in the standard protocol and is not expanded in detail

Special Class request "In fact, special class request can also be divided into two categories according to the interface: 1) video control request, 2) video streaming request.

1）视频控制请求有：Video control requests include:

接口控制请求（Interface Control Request）

Camera终端控制请求（Camera Terminal Control Request）

选择单元控制请求（Selector Unit Control Request）

处理单元控制请求（Processing Unit Control Request）

扩展单元控制请求​（Extension Unit Control Request）

2）视频流请求仅有接口控制请求(Interface Control Request)一种：

其中，Interface Control Requests因为是用来在主机和设备之间协商数据交互格式和分辨率，流量等信息的，所以一般来说是必须实现的。Camera Terminal Control Requests包含了对曝光时间，曝光模式，对焦，变焦，平移等sensor获取数据阶段时的参数控制。Processing Unit Control Requests中则包含了亮度，增益，色调，对比度，白平衡等等sensor在获取到图像数据后，图像处理阶段的相关参数。

不过实际上，以上两者的划分在硬件层次并不是绝对的。很多参数的控制在sensor硬件级别上是同一层次的。不过，将这些功能抽象成这两类，正如在硬件的拓扑结构上将功能模块抽象出来一样，有利于通用化的程序设计。红外模组因为其特殊性，与标准的可见光摄像头模组有所差别，除了接口控制请求外，并未实现其他请求，ISP参数的读写都是通过自定义的Vendor Command来实现。

Interface Control Requests are generally mandatory because they are used to negotiate data exchange formats and resolutions, traffic, and so on between hosts and devices. Camera Terminal Control Requests include parameter controls for exposure time, exposure mode, focus, zoom, pan and other sensor data acquisition stages. Processing Unit Control Requests includes brightness, gain, hue, contrast, white balance and other parameters related to image Processing after sensor obtains image data.

In fact, however, the division between the two is not absolute at the hardware level. Many parameters are controlled at the same level on the sensor hardware level. However, the abstraction of these functions into these two categories, just as the abstraction of functional modules in the topology of hardware, facilitates the design of generalized programs. Due to its particularity, Tiny1C/TinyBE infrared module is different from the standard visible light camera module. Except for interface control request, other request is not realized. The read and write of ISP parameters are realized through customized Vendor Command.

### Libiruvc库接口介绍Libiruvc library interface introduction

libiruvc是由开源的libuvc库改写而成， libuvc又依赖开源的libusb库。libuvc是一个跨平台的专为uvc摄像头开发的库，该开源库具有UVC设备发现和管理API，具有异步/回调和同步/轮询模式的视频流（设备到主机），能对标准设备设置的读/写访问，能进行各种格式之间的转换：RGB、YUV、JPEG等。经过在Mac和Linux上测试，可移植到Windows和一些BSD。对于本SDK而言，Libiruvc和libusb实现了标准uvc摄像头用户态的驱动开发。

Libiruvc is adapted from the open source Libuvc library, which relies on the open source Libusb library. Libuvc is a cross-platform library developed especially for UVC camera. The open source library (1)has UVC device discovery and management API, (2)has asynchronous/callback and synchronous/polling mode video stream (device-to-host), (3)can read/write access to standard device Settings, (4)can convert between various formats: RGB, YUV, JPEG, etc.

Tested on Mac and Linux, it is portable to Windows and some BSD. For this SDK, Libiruvc and Libusb realize the driver development of standard UVC camera user mode.

libiruvc对libuvc进行了进一步封装。下面简单介绍下libuvc封装后的接口。

Libiruvc further encapsulates Libuvc. The following is a brief introduction to the libuvc encapsulated interface.

1. uvc\_camera\_init 初始化uvc服务的上下文

Initializes the context of the UVC service

1. uvc\_camera\_list 列出当前的usb设备并获取它们的枚举信息

Lists the current USB devices and gets their enumeration information

1. uvc\_camera\_info\_get 获取对应的uvc设备的视频信息

Obtain the video information of the UVC device

1. uvc\_camera\_open 打开一个uvc设备

Open a UVC device

1. uvc\_camera\_open\_same 在相同的VID和PID下打开不同的设备

Turn on different devices with the same VID and PID

1. uvc\_camera\_stream\_start 根据设定的图像格式、分辨率和帧率进行出图

According to the set image format, resolution and frame rate map, start streaming

1. uvc\_camera\_stream\_close 停止uvc设备出图

Stop camera streaming

1. uvc\_camera\_close 关闭一个uvc设备

Close a UVC device

1. uvc\_camera\_release 退出当前的uvc服务上下文

Exit the current UVC service context

1. uvc\_frame\_buf\_create 根据视频参数申请buffer空间

alloc for buffer space according to video parameters

1. uvc\_frame\_buf\_release 释放申请的buffer空间

Release the requested buffer space

### 不同模式数据流Different data flows

在SDK sample中，通过camera.h文件中修改不同的宏，可以输出不同分辨率和模式的数据流。

In the SDK sample, different macros can be modified in the camera.h file to output data flows of different resolutions and modes.

|  |  |  |
| --- | --- | --- |
| 宏 macro | 数据data | 分辨率resolution |
| IMAGE\_AND\_TEMP\_OUTPUT | 图像+温度Image + Temperature | 384\*576/640\*1024 |
| IMAGE\_OUTPUT | 图像Image | 384\*288/640\*512 |
| TEMP\_OUTPUT | 温度(通过basic\_y16\_preview切换)  Temperature(Switch by basic\_y16\_preview) | 384\*288/640\*512 |

## USB控制接口说明USB control description

### 1.概述overview

Libiruvc库中既支持USB控制命令传输，也支持I2C命令，因此需要通过vdcmd\_init\_by\_type函数注册相关的接口类型来使用不同的通信接口。该库的源码可开放给客户。也可以发送交叉编译工具链由红外厂商编译。目前已经编译好主流的嵌入式平台的库，客户可以根据需要进行选用。Tiny1BE/Tiny1C的USB控制接口实质上通过RTS5840/5830芯片实现uvc协议，并转发I2C控制命令到模组ISP芯片。

The Libiruvc library supports both USB control command and I2C commands, so you need to register the related interface types through the vdcmd\_init\_by\_type function to use different communication interfaces. The source code of the library is open to customers. It is also possible to send cross-compilation toolchains compiled by infrared vendors. At present, we have compiled the mainstream embedded platform library, and customers can choose according to their needs. The USB control interface of Tiny&Mini256 essentially realizes UVC protocol through RTS5840/5830 chip and forwards I2C control commands to module ISP chip.

### 2.特殊命令接口Special command interface

可参考《模组SDK接口V1.5.xlsx》表格，因SDK的产品兼容性考虑，部分接口并非本产品使用，有些接口存在冗余。用户只需使用《模组SDK接口V1.5.xlsx》表格中的指令即可，**表格中未列出的指令请不要使用，尤其是擦写flash的函数**。下表重点介绍几个常用的命令接口。

Refer to “模组SDK接口V1.5.xlsx”. Due to the compatibility of SDK products, some interfaces are not used by this product, and some interfaces are redundant. Users only need to use the instructions in the tabl e“模组SDK接口说明Interface Description1.5”, please do not use the instructions not listed in the table, especially the function to erase and write the flash. The following table focuses on several common command interfaces.

|  |  |  |
| --- | --- | --- |
| 函数接口Function | 功能Description | 说明Notes |
| basic\_y16\_preview | 用于切换出图数据源  (可切换成温度数据输出)  Use to switch out the graph data source(Can be switched to temperature data output) | 仅适用于384\*288/640\*512分辨率下使用  384\*288/640\*512 Only |
| adv\_dpc\_auto\_calc | 用于模组内置自动盲元标定  Auto Dead Pixel correction | 设定时间越长，效果越好  By increasing the setting time, the effect can be improved |
| adv\_tpd\_ktbt\_recal\_1point | 用于测温二次标定(单点标定)  One-point Secondary calibration | 需要等待较长时间，约10s  about 10s |
| adv\_tpd\_ktbt\_recal\_2point | 用于测温二次标定(两点标定) Two-point Secondary calibration | 需要等待较长时间，约15s  about 15s |

### 3. 其他命令接口Other command interface

可参考《模组SDK接口V1.5.xlsx》表格

Refer to libirSDK.chm and “模组SDK接口说明Interface Description1.5”.

## SDK其他功能介绍Other Functions

### 固件升级功能Firmware Upgrade

因为模组的固件还在不断的开发和完善中，如果出现固件bug或者需要使用固件支持的新功能，用户可以通过该功能进行固件升级。调用libircmd库中的basic\_update\_fw函数，参考update\_fw\_sample，打开后，在sample.h中选择装置类型，并在sample.cpp的update\_fw\_cmd接口中选择需要更新的固件文件，即可进行固件升级，用户可将固件升级的相关代码移植到自己的应用软件中。

Because the firmware of the module is still under constant development and improvement, users can use this function to upgrade the firmware if there are firmware bugs or new features supported by the firmware. Call basic\_update\_fw in libircmd library and refer to update\_fw\_sample. After opening update\_fw ，select the device type in sample.h and select the firmware file to be updated in the update\_fw\_cmd interface of sample.cpp to perform the firmware upgrade. You can transplant firmware upgrade codes to your own application software.

### 测温二次标定Secondary calibration

非制冷红外热像仪对温度敏感，无论模组出厂测温标定如何精确，当用户将模组集成到整机中后，由于热分布的变化、光学结构的变化（加窗口片等），必然引起一定的测温偏差。因此需要在整机端进行二次标定。该功能在模组固件中实现，详细的使用说明见《客户开发标定》目录下的《二次标定》。

When the user integrates the module into the whole machine, it will inevitably cause a certain temperature measurement deviation due to the changes of thermal distribution and optical structure (add window, etc.). Therefore, Secondary calibration is required at the end of the machine. This function is realized in the module firmware. For detailed instructions, see "Secondary calibration " in the directory of "User calibration instructions".

### 盲元标定Dead pixel correction

模组在使用过程中，受到比较严重的机械冲击或静电放电后，有极低的概率出现新增盲元，用户可将新增盲元添加到盲元表中。该功能有多种实现方法：1. 调用固件中的自动标定功能；2. 手动输入新增盲元坐标。详细的使用说明见《客户开发标定》目录下的《盲元标定》。

If a module is hit by a serious mechanical shock or electrostatic discharge (ESD), there is a very low probability that new dead pixels will appear. You can add new dead pixels to the dead pixel table. There are several ways to implement this function: 1. Call the automatic calibration function in firmware; 2. Manually enter the coordinates of the new Dead pixel. For detailed instructions, see Dead pixel correction in the catalogue of "User calibration instructions"

### 测温环境变量修正功能Ambient variable correction

红外测温的精度受到很多环境参数的影响，例如距离、环境温度、反射温度、湿度、目标发射率等等。Libirtemp SDK为用户提供了一种测温修正的方法，可以使测温精度更加接近真实值，接口为enhance\_distance\_temp\_correct。详细的修正方案见《客户开发标定》目录下的《环境变量修正》。

The temperature measurement of the module is affected by Emissivity, Atmospheric transmissivity, ambient reflection temperature, ambient atmospheric temperature, target distance, et Libirtemp SDK provides users with a correction method, which can make the temperature measurement accuracy closer to the real value，the interface is enhance\_distance\_temp\_correct. For detailed instructions, see "Ambient variable correction" in the directory of "User calibration instructions".

### 点线框测温说明Description of point/line/box temperature measurement

本产品模组内置点线框测温功能，可以分别调用libircmd库中的basic\_tpd\_get\_point\_temp\_info、basic\_tpd\_get\_line\_temp\_info，basic\_tpd\_get\_rect\_temp\_info来获取模组内的特定的点线框温度信息，也可以调用函数basic\_tpd\_get\_frame\_temp\_info函数获取整帧的温度最大/最小值信息。

同时SDK libirtemp中也提供了点线框的测温功能，使用前提是获取到了一整帧的温度数据，可以分别调用get\_point\_temp、get\_line\_temp、get\_rect\_temp函数获取特定的点线框温度信息。为了减小空域噪声的影响，其中点测温的功能是取点周围3\*3领域内的像素温度值去掉最大最小值后的平均值，线测温和框测温均以点测温为基础进行开发。若用户不满意该SDK的功能，可自行开发。

This product module has built-in temperature measurement function of point, line and box. You can call basic\_tpd\_get\_point\_temp\_info、basic\_tpd\_get\_line\_temp\_info、basic\_tpd\_get\_rect\_temp\_info to get the specific temperature information. You can also call the basic\_tpd\_get\_frame\_temp\_info function to obtain maximum/minimum temperature information for the entire frame.

At the same time, the SDK libirtemp also provides the temperature measurement function of point, line and box. The prerequisite is to obtain the temperature data of a whole frame. You can call get\_point\_temp, get\_line\_temp, and get\_rect\_temp respectively to obtain the specific temperature information. In order to reduce the influence of spatial noise, the function of point temperature measurement is to take the average value of pixel temperature values in 3\*3 fields around the point after removing the maximum and minimum values. Both line and box temperature measurement are developed on the basis of point temperature measurement. If users are not satisfied with the functions of the SDK, they can develop it by themselves.

### 其他库的功能介绍Other Functions

libirprocess、libirparse、libirtemp三个库文件均是软件层面的数据处理函数库，客户可根据具体的需求选用相关的函数接口。

libirprocess、libirparse、libirtemp are all data processing function libraries at the software layer. Customers can choose relevant function interfaces according to their specific needs.

libirprocess:实现了图像的旋转、镜像、翻转、图像增强、空域滤波等功能

The functions of image rotation, mirror, flip, image enhancement, Spatial Noise Reduction and so on are realized

libirparse:实现了图像数据格式转换功能

The functions of Data conversion are realized

libirtemp:实现了点、线、框测温和相关的测温修正函数

The functions of point/line/box temperature measurement are realized

## 数据格式说明Data Format Description

### 1.图像数据Image data

图像数据的输出格式为YUV422格式，即YUYV格式。

The output format of image data is YUV422 format, namely YUYV format.

### 2.温度数据Temperature data

#### 获得一整帧的温度数据Get a whole frame of temperature data

每个像素的温度需要2个byte存储，放在高14位，按照小端存放。如下图。

The temperature data of each pixel needs to be stored in 2 bytes, which are placed in the upper 14 bits and stored as the little end.

|  |  |
| --- | --- |
| 低地址 Low address | 高地址 High address |
| 低字节 Low byte | 高字节 high byte |

计算公式为The calculation formula：Celsius temp = Count/64-273.15

#### 发命令获取点线框温度数据Send the command to get the temperature data

每个像素的温度需要2个byte存储，放在低14位，按照小端存放。如下图。

The temperature data of each pixel needs to be stored in 2 bytes, which are placed in the upper 14 bits and stored as the little end.

|  |  |
| --- | --- |
| 低地址 Low address | 高地址 High address |
| 低字节 Low byte | 高字节 high byte |

计算公式为The calculation formula：Celsius temp = Count/16-273.15