# 使用说明

**概述**

在Linux下，本摄像头使用的是UVC协议，使用V4L2可以打开使用本摄像头，在编译Linux系统时，需添加V4L2模块。数据传输量大，Vmware和HUB会导致数据传输错误，不推荐使用，但是部分电脑使用Vmware设置USB3.0后可以使用。

**Linux PC快捷使用本例子**

安装opencv 4.x版本

在以交叉编译器命名的文件夹里，把对应平台的的库文件，复制到此文件目录下。

在teminal使用以下命令可演示例子：

g++ main.cpp Usb.cpp -L. -lthermometry -lSimple -lm -lpthread -lpot -lusb-1.0 -o main1 `pkg-config opencv4 --libs --cflags`

export LD\_LIBRARY\_PATH=./

./main1

在teminal输入a（不需要回车）打开设备，显示图像并开始测温，输入s关闭设备。

其中Opencv在本例中使用了yuyv转rgb和显示等功能，并不是移植必需的。

**模块介绍**

libthermometry.so thermometry.h 测温模块

libSimple.so SimplePictureProcessing.h 图像处理模块

libpot.so pot.h Usb.cpp Usb.h 锅盖处理模块（适用于384×288面阵并且机芯输出yuyv数据）

**Main.cpp代码介绍**

主函数main():创建界面及交互按钮，监听键盘按键，并发送给handlerThread处理

处理线程handlerThread：根据键盘按键，进行具体处理

显示线程renderThread：按照uvc传输，使用V4L2进行显示的线程

Bulk线程bulkThread：进行锅盖处理的线程，（适用于384×288面阵并且机芯输出yuyv数据），使用bulk传输，跟uvc传输互斥

**handlerThread介绍**

回车键:打快门，设备挡片会进行一次打开关闭的操作，可听到咔咔的声音，定时打快门可使图像清晰，测温准确。

0：启动bulk线程，并进行锅盖标定

1：标定锅盖前进行数据采集

2：使标定锅盖生效

3：是标定锅盖无效

4：保存参数：设置测温参数后，使用保存参数，断电不丢失，否则断电恢复上一次状态。

5: 8005模式设置第一个额外点的例子（8004和8005的区别参照renderThread介绍）

6: 8005模式设置第二个额外点的例子

7: 8005模式设置第三个额外点的例子

q: 8004模式下切换为白热色板

w: 8004模式下切换为黑热色板

e: 8004模式下切换为铁虹色板

r: 8004模式下切换为彩虹1色板

t: 8004模式下切换为彩虹2色板

y: 8004模式下切换为高动态彩虹色板

u: 8004模式下切换为高对比彩虹色板

i: 8004模式下切换为熔岩彩虹色板

o: 8004模式下切换为第二种铁虹色板

p: 8004模式下computeMethod使用ComputeDivTempType2时，切换displayMode（参照renderThread介绍里的ComputeDivTempType2）

a: 创建renderThread，打开UVC设备，并显示

b: 关闭UVC设备

d: 8004模式下循环切换computeMethod（参照renderThread介绍里的computeMethod）

f: 切换到高温度段测温

g: 切换到常温度段测温

h: 设置测温参数（参数作用，参照代码注释）

z: 8005模式下切换色板，色板样式跟8004模式下有差异

**renderThread介绍**

**V4L2相关：**

int init\_v4l2(void)；//初始化v4l2，此处需要注意打开的是否是红外设备

int v4l2\_grab(void); //采集数据

int v4l2\_control(int); //摄像头控制

int v4l2\_release();//释放

*tips关于长跟宽：*

图像最后4行用来传递参数，例如对T3系列，本来的分辨率为384×288，采集时要设置为384×292，292-288=4这4行是用来传递参数的。

**摄像头控制及相关配置：**

机芯控制指令采用UVC协议中的Zoom (Absolute)控制通道，控制指令共2个字节，其中低字节为参数值，高字节为寄存器地址；

例如：发送快门校正指令 0x8000时调用v4l2\_control(0x8000)。

详细指令好如下（更多参数请参照产品规格书）：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 寄存器地址 | 参数 | 描述 |
| 1 | 0x80 | 0x00 | 快门校正 |
| 2 | 0x01 | 背景校正 |
| 3 | 0x02 | 探测器原始输出 |
| 4 | 0x04 | 16bit NUC原始数据 |
| 5 | 0x05 | YUYV数据输出 |
| 6 | 0x20 | 常温段测温 |
| 7 | 0x21 | 高温度段测温 |
| 8 | 0xFE | 保存配置参数(数据源切换、模组色板) |
| 9 | 0xFF | 保存测温参数 |
| 10 | 0x88 | 00-07 | 色板 |

1. 快门校正：进行非均匀性校正，发送指令后会打快门，快门校正可以使图像均匀、测温准确。工业类测温一般每三分钟打一次，体温类测温一般一分钟打一次快门，打完快门1秒钟（红外相机帧率是25帧）进行计算表的操作（详见下方测温）。
2. 背景校正：相机面对均匀的场景实现快门校正的功能。

3、0x8004与0x8005的区别：

⑴、在0x8004状态下，虽然采用yuyv的输出格式，但实际上输出的是16bit NUC（RAW）数据，每个连续的16bit中用14bit存储一个点的宽动态灰度值。在0x8005状态下，输出的就是yuyv的数据。

⑵、在0x8004状态下，原始数据经过libthermometry.so的处理，可以转化为全局的温度数据，经过libSimple.so或者线性算法（详见代码main）的处理可以转变为rgba数据。在0x8005状态下，输出的yuyv数据，后面的4行参数经过libthermometry.so的处理可以得到：中心点温度，最高温点温度及位置，最低温点温度及位置，3个自定义点温度，无法得到全局温度数据。

**测温库相关：**

测温分两步：

计算温度表，然后用16bit RAW数据去查表。thermometryT4Line作用计算温度表，thermometrySearchXXX为查表操作。详见thermometry.h注释。

设置测温参数：

handlerThread里的h，当设置完参数后，需要重新计算温度表，确认无误后保存参数，保存参数后设置的参数断电不丢失，否则断电丢失

常温段与高温度段：

工业类测温，常温段为-20摄氏度到120摄氏度，高温度段为120摄氏度到400（不同产品略有不同）摄氏度，高温度段部分产品不支持，具体请以规格书为准。切换到高温段测温需要10分钟左右稳定时间（不同产品存在差异），如果想较快获得清晰图像和温度，除了handlerThread里g的打快门和计算表的方式外，切换后前三分钟一般每分钟打一次快门并且计算表，然后再按照三分钟的频率打快门计算表。切换高温度段测温无法断电保存，需要每次启动后再切换。

**图像处理相关：**

使用libSimple.so里的图像算法需要先初始化高性能算法，最后释放资源。

8004模式下：

computeMethod==1时，使用效果最好的图像算法Compute，推荐主频大于1.8ghz。推荐参数设置SetParameter(100,0.5f,0.1f,0.1f,1.0f,3.5f)，不建议修改;

computeMethod==2时,使用效果一般的图像算法ComputeMethodTwo，对主频没有硬性要求，推荐参数设置SetMethodTwoParameter(13,0,3,3,50,50);

computeMethod==3时，使用ComputeDivTemp，是在ComputeMethodTwo和SetMethodTwoParameter基础上，结合测温SetDivTemp(divTmpNuc1,divTmpNuc2)，在divTmp1摄氏度以下黑白显示图像，divTmp1摄氏度到divTmp2摄氏度之间从黄到红过渡显示，divTmp2摄氏度以上红色显示。由于跟温度相关，所以每次计算表thermometryT4Line后，都需要重新SetDivTemp。

computeMethod==4时，使用ComputeDivTempType2，是在ComputeDivTemp基础上进行的改进，divTmp1以下，显示色板最低温，divTmp2以上显示色板最高温，divTmp1到divTmp2之间，图像呈色板过渡显示。通过lowTmpPaletteIndex，可设置最低温处在色板的位置。displayMode==1时，通过divTmp1二值化显示。displayMode==2时，通过divTmp1、divTmp2三值化显示。

computeMethod==0时，使用线性算法进行显示。

关于色板：

⑴、在0x8005状态，发送0x88（00-07）可切换色板，八种色板分别为：白热、 黑热、铁虹、熔岩、彩虹、铁灰、红热、彩虹2。

⑵、在0x8004状态下，色板类型参考handlerThread里的色板切换。

**bulkThread介绍**

进行锅盖处理的线程，适用于384×288面阵并且机芯输出yuyv数据，使用bulk传输，跟uvc传输互斥。

1、首先需要获得访问usb的权限

sudo vi /etc/udev/rules.d/50-myusb.rules

SUBSYSTEMS=="usb", ATTRS{idVendor}=="1514", ATTRS{idProduct}=="ffff", GROUP="jun", MODE="0666"

sudo udevadm control --reload

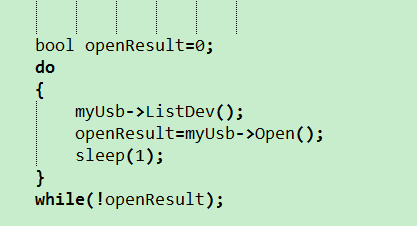
如果后面提示没有获得权限打不开，需要插拔设备或者断电重启。

Group按照实际情况填写。

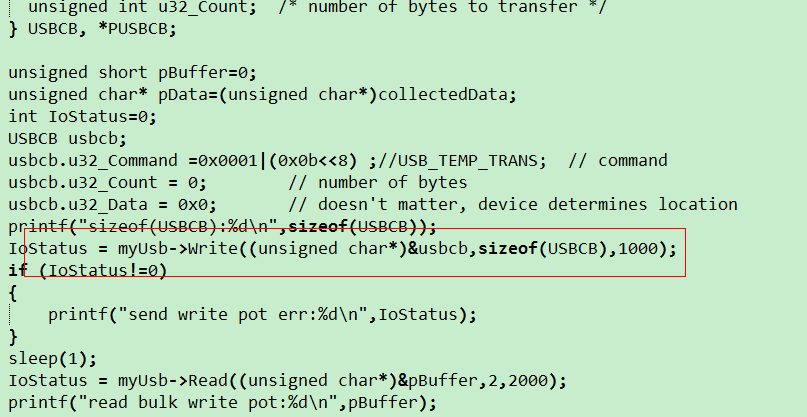
设备在uvc状态下vid=1514 pid=0001

在bulk状态台下，vid=1514 pid=ffff

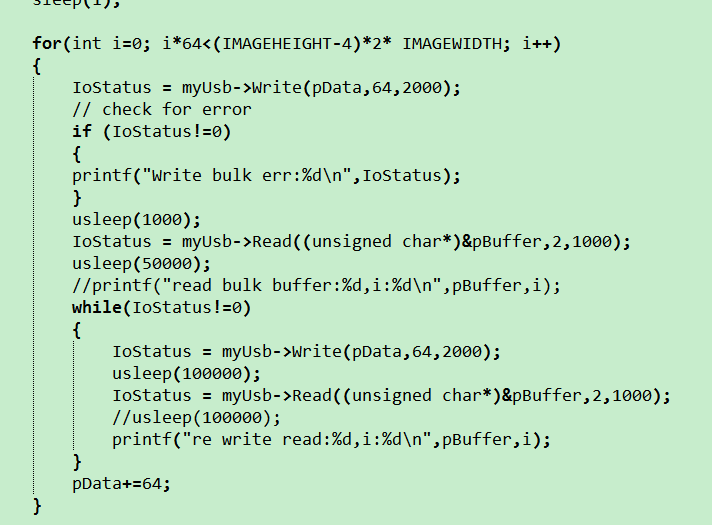
2、uvc模式下，在8004模式下，terminal输入1（不要带回车），采集数据并处理16帧数据（collectData函数）。

输入0，切换到bulk模式，并退出释放uvc资源。

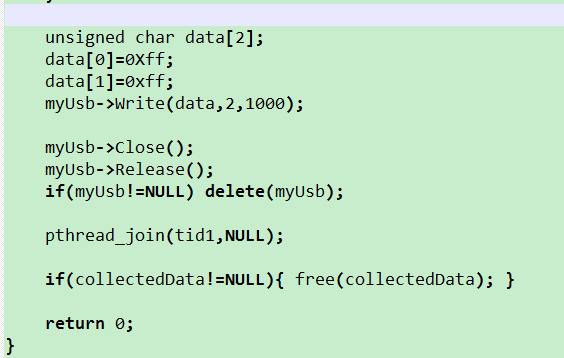
由于切换到bulk需要时间，所以循环直到打开bulk.（切换中有极小几率设备挂载不上，长时间没有找到1514设备，需要重新插拔或者断电重启，可以在这个循环里加计数器，计数大于多少认为没有挂载成功）。



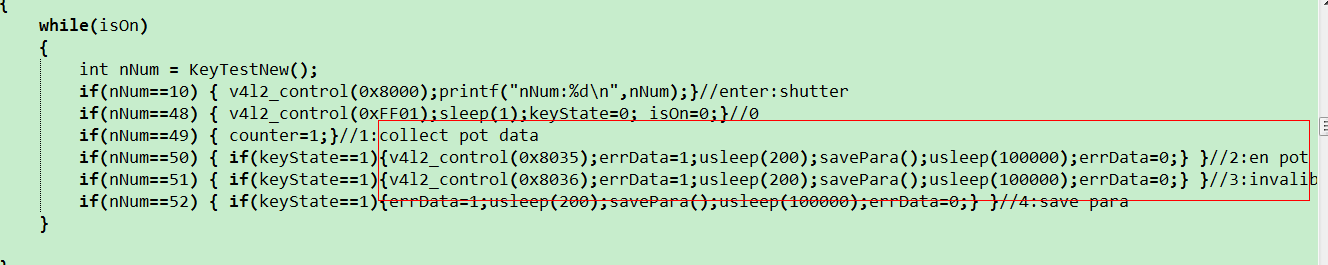
Bulk发送指令，告诉设备要写锅盖数据。



pBuffer返回的数值跟i一样。



Bulk传输指令，通知切回uvc，释放所有bulk资源



断电重启，在uvc 8005状态设备下打开显示，输入2，使能去锅盖功能，并保存，断电不丢失。

输入3，使去锅盖功能失效，并保存