

Linux 版 SDK 常见问题解答

1. GigE Vision 相机

问题 1：GalaxyView 或 GxGigEIPConfig 无法枚举到 Interface 和 GigE 相机

可能的原因：

- 1) 主机 (PC) 没有连接相机或网络；
- 2) 主机 (PC) 没有手动设置 IP，使用自动模式获取 IP；
- 3) 启用了网络防火墙。

解决方法：

- 1) 检查主机和相机是否连接正常，相机指示灯有没有变成绿色；
- 2) 通过如图 1-1 所示方法进行手动设置 IP；

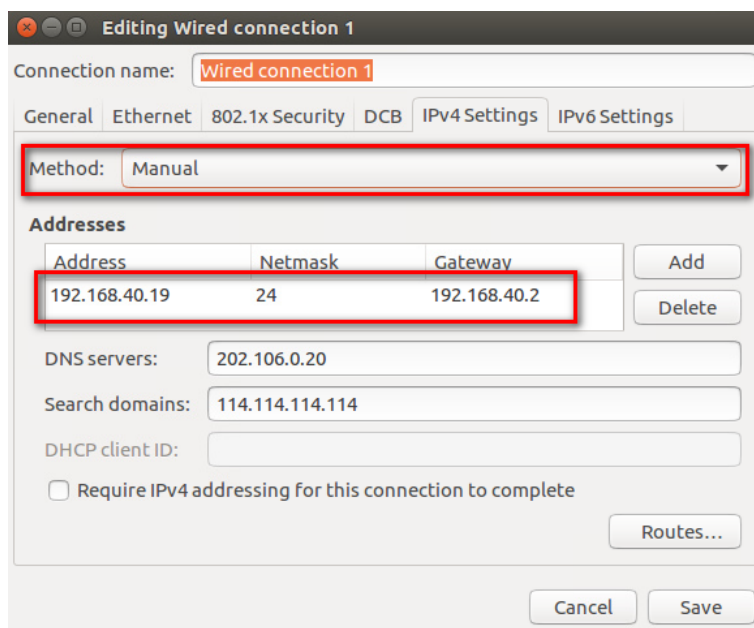


图 1-1

- 3) 请确认关闭网络防火墙，否则设备枚举和接收数据可能无法工作。

问题 2：使用 GxGigEIPConfig 工具无法枚举到不同子网的相机或多网卡下有网卡枚举不到相机

可能的原因：

由于 Linux 系统的反向过滤技术导致了 GxGigEIPConfig 工具无法枚举到相机。

解决方法：

在确定相机连接正常的情况下，可通过以下命令查看反向过滤功能是否开启：

```
sysctl -a 2>/dev/null | grep '\.rp_filter'
```

终端中可以看到如下输出：

```
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.eth0.rp_filter = 1
net.ipv4.conf.eth1.rp_filter = 1
```

其中 eth0 和 eth1 为连接相机的网卡，当值为 1 时表示开启了反向过滤功能。

命令 net.ipv4.conf.all.rp_filter 是一个全局开关，控制所有网卡。

命令 net.ipv4.conf.eth0 (eth1) .rp_filter 控制每个单独的网卡。

若要关闭反向过滤功能，应该先将“all”关闭，使用如下命令关闭反向过滤功能：

```
sudo sysctl net.ipv4.conf.all.rp_filter=0
sudo sysctl net.ipv4.conf.eth0.rp_filter=0
sudo sysctl net.ipv4.conf.eth1.rp_filter=0
```

然后使用 GxGigEIPConfig 工具重新枚举相机，相机可以被枚举到。

当重新启动系统时，反向过滤功能会被重新开启，若要永久关闭该功能，则需要将以上设置编辑到 /etc/sysctl.conf 文件。

问题 3：当关闭反向过滤功能后，仍然无法枚举到 GigE 相机

解决方法：

该情况下，需要确认相机 IP 地址与网卡 IP 地址是否相同。可通过修改网卡 IP 或者更换另外一个网卡与相机连接，再次尝试枚举。

问题 4：GigE 相机开采后，采集帧率、显示帧率为 0.0

解决方法：

请按以下步骤检查：

- 1) 检查相机与网卡的物理连接，电源是否正常；
- 2) 查看属性列表中的流标签页中的统计数据，查看是否出现大量残帧，若出现大量残帧，可尝试增大设置中的数据块超时时间。

问题 5：GigE 相机采集中出现残帧

解决方法：

- 1) 若网络适配器支持巨帧，我们建议设置为 8192，命令如下：

```
sudo ifconfig ethx mtu 8192
```

然后将相机的包长设置为 8192（通过我们的 API 修改“流通道包长”的值）。

- 2) 若仍然性能不足，可尝试增加相机的包间隔大小（通过我们的 API 修改“流通道包间隔”的值）。
- 3) 针对大分辨率相机，可尝试增大流层控制参数的数据块超时时间（通过我们的 API 修改“数据块超时时间”的值）。

问题 6：GigE 相机在 TK1 开发板上采集出现大量的残帧

可能的原因：

系统运行在低功耗模式，导致产生了大量丢包。在 Linux 中，内核的开发者定义了一套框架模型来完成 CPU 频率动态调整这一目的，它就是 CPU Freq 系统。尽管在各个 Linux 发行版中，前端软件稍有差异，但其最终都会通过 Linux 内核的 CPU Freq 系统来实现 CPU 频率动态调整的功能。这些软件都会提供如下 CPU 模式（governor 参数）：ondemand，interactive，performance，powersave 等。其中 ondemand 是系统默认方式；performance 是性能模式，系统运行在最高频率下，但是功耗较高。

解决方法：

当出现此问题时，可尝试将 CPU 模式设置为 performance。

以 ubuntu 系统为例，通过 cpufreq 管理软件调整 CPU 的工作模式，步骤如下：

- 1) 安装 cpufrequtils 管理软件：

```
sudo apt-get install cpufrequtils
```

- 2) 修改配置文件/etc/init.d/cpufrequtils：

将 GOVERNOR="ondemand" 改为：GOVERNOR="performance"

保存后执行如下命令即可生效：

```
/etc/init.d/cpufrequtils restart
```

- 3) 最后需要禁用 ondemand 守护程序，否则重启后设置将被覆盖：

```
sudo update-rc.d ondemand disable
```

2. USB3 Vision 相机

问题 1：使用 4 台以上相机，开采失败问题，提示开采失败如图 2-1

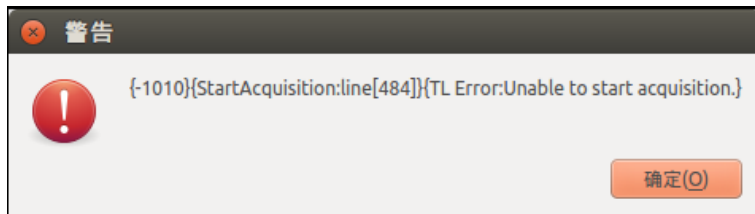


图 2-1

解决方法：

1) 在终端中执行命令：

```
“echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb”
```

2) 执行安装包中的脚本文件 SetUSBStack.sh。

以上两种方法的设置仅对本次开机启动有效，如果需要一直有效，可以写入开机脚本中。

3. 安装包

问题 1：用户将水星相机 SDK 包中的库文件打包到自己的安装包内，安装后无法运行程序

解决方法：

1) 在用户程序的 makefile 中添加：

CPPFLAGS := -w -I./ -L/usr/lib -L./ -Wl,-rpath=/usr/lib:./

添加后如图 3-1 所示：

```
# Makefile for sample program
.PHONY      : all clean

# the program to build
NAME        := GxSingleCamColor

# Build tools and flags
CXX         := g++
LD          := g++
SRCS        := $(wildcard *.cpp)
OBJS        := $(patsubst %cpp, %o, $(SRCS))
CPPFLAGS    := -w -I./ -L/usr/lib -L./ -Wl,-rpath=/usr/lib:./

LDFLAGS     := -lgxiapi -lpthread

all          : $(NAME)

$(NAME)      : $(OBJS)
              $(LD) -o $@ $^ $(CPPFLAGS) $(LDFLAGS)

%.o          : %.cpp
              $(CXX) $(CPPFLAGS) -c -o $@ $<

clean        :
              $(RM) *.o $(NAME)
```

图 3-1

2) 重新编译用户程序；

3) 将我公司提供的安装包的库，拷贝到用户安装包中用户程序同一目录下。需要拷贝的库有 libgxiapi.so, GxU3VTL.cti, GxGVTL.cti 这三个文件。



注意：用户构建自己的安装包时，除了需要将相机驱动文件按照上述两种方式进行操作之外，还需要将我公司 Linux 版本安装包安装后生成的 config 文件下的两个文件拷贝到指定目录。

对 config 目录下的文件及需要拷贝到的位置进行说明：

1) config 目录下的 99-galaxy-u3v.rules 文件用于在非特权模式下枚举 U3 相机，缺少该文件可能造成只能在特权模式(sudo 或 root 身份)下枚举到 U3 相机。99-galaxy-u3v.rules 文件需要被拷贝到系统的/etc/udev/rules.d 目录中生效。

99-galaxy-u3v.rules 文件拷贝目录如图 3-2 所示：

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
wangdh@wangdh-System-Product-Name:~$ ls /etc/udev/rules.d/
70-snap.core.rules  99-galaxy-u3v.rules
wangdh@wangdh-System-Product-Name:~$
```

图 3-2

2) config 目录下的 galaxy-limits.conf 文件用于提升采集性能，文件需要被拷贝到系统的 /etc/security/limits.d 目录下生效。

galaxy-limits.conf 文件拷贝目录如图 3-3 所示：

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
wangdh@wangdh-System-Product-Name:~$ ls /etc/security/limits.d/
galaxy-limits.conf
wangdh@wangdh-System-Product-Name:~$
```

图 3-3

4. 系统问题

问题 1：在 Jetson TX2/nano R32.2.3 版本系统中（linux 4.9 内核），如果反复进行开停采（GXStreamOn/GxStreamOff）操作，到达一定次数后会出现开采失败的问题，重启程序后可恢复

问题根因：

该问题属于 TX2 R32.2.3 版本系统的 bug。在该版本系统中，如果启动线程时对线程优先级进行提升，线程退出时会出现内存泄露，实际为 VIRT（虚内存）泄露，通过 top 命令可以观察到该现象。当 VIRT 值超出系统允许临界值时，会出现无法启动线程的问题。虚内存是申请的内存，对应的还有实内存，表示实际使用的内存。实内存属于真正的内存占用，可能会影响其他进程的执行性能。虚内存溢出只会影响本进程内存的申请，其他进程则不受影响。

只有在大量的反复开停采场景中才会出现此问题，TX2 中约为 3000 次左右。如果您遇到了该问题，可以参考以下方案进行解决。

方案一（推荐方案）：

经测试该问题仅出现在 Jetson TX2/nano R32.2.3 版本系统中，其他新旧版本均无此问题，用户可以通过升级或者降低系统版本来避免此问题。

方案二：

执行以下命令删除 galaxy-limits.conf 文件，重启系统即可解决 VIRT 泄漏问题。

```
sudo rm -f /etc/security/limits.d/galaxy-limits.conf
```



注意：方案二将不会对 SDK 内部的采集线程进行优先级提升，如果系统启动的进程或线程过多，可能会导致采集帧率不稳定，请谨慎使用。