Linux SDK FAQ

1. GigE Vision Camera

Question 1: The interfaces and GigE cameras cannot be enumerated by GalaxyView or GxGigeIPConfig

Possible reasons:

1) There is no camera connected to the PC or the PC does not connect to the network.
2) The user does not set the PC's IP manually, but get the PC's IP automatically.
3) Network firewall is enabled.

Solutions:

1) Check whether the host (PC) and the camera are connected properly and whether the camera LED indicator turns green.
2) Set the IP manually with the following method:
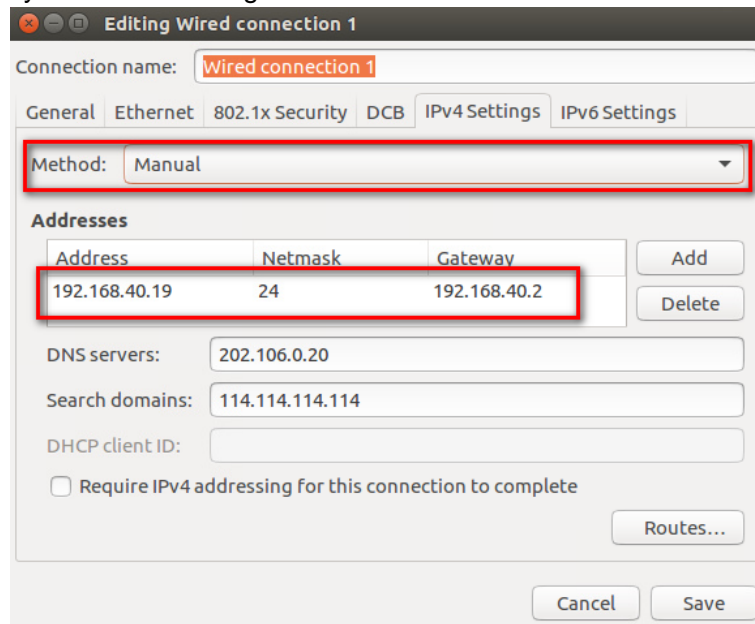

Figure 1-1

3) Please confirm that the network firewall is disabled, otherwise the device enumeration and receiving data may fail.

Question 2: When using the GxGigeIPConfig tool, the cameras of different subnets cannot be enumerated, or some cameras cannot be enumerated when there are many network cards in the PC

Possible reasons:

The cameras cannot be enumerated with GxGigeIPConfig tool due to the reverse-path filtering technology of the Linux system.

Solutions:

When you are sure that the camera is connected properly, you can check to see whether the reverse-path filtering function is enabled by the following command:

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

The following outputs can be seen in the terminal:

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

The eth0 and eth1 are the network cards that connected to the camera. When the value is 1, the reverse-path filtering function is enabled.

Command net.ipv4.conf.all.rp_filter is a global switch that controls all the network cards.

Command net.ipv4.conf.eth0(eth1).rp_filter controls each single network card.

To disable the reverse-path filtering function, the "all" should be disabled first. You can use the following commands to close the reverse-path filtering function:

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

Then re-enumerate the camera with the GxGigeIPConfig tool, the camera can be enumerated.

When the system is restarted, the reverse-path filtering function also will be enabled, and if you want to disable this function permanently, you should edit the above settings to the /etc/sysctl.conf files.

Question 3: When the reverse-path filtering function is disabled, the GigE cameras still cannot be enumerated

Solutions:

You should check whether the camera's IP address is the same as the network card's IP address. You can change the network card's IP or change another network card to connect with the camera, and try enumerating again.

Question 4: The acquisition frame rate and display frame rate are 0.0 after GigE cameras start acquisition

Solutions:

Please try to check it by the following steps:

1) Please check whether the camera and the network card are connected properly and the power supply is working properly.
2) Check the statistics in the Stream Features in GalaxyView to see whether there are many incomplete frames, if there are, try to increase the BlockTimeout in the settings.

Question 5: There are incomplete frames in GigE cameras acquisition process

Solutions:

1) If the network adapter supports jumbo frames, we recommend you to set the packet size to 8192.

The command is as follows:

sudo ifconfig ethx mtu 8192

Then set the camera's packet size to 8192. (You can change "PacketSize" value by setting the corresponding value via the API.)

2) If the network transmission performance is still insufficient, try to increase the camera's packet delay. (You can change "PacketDelay" value by setting the corresponding value via the API.)

3) For large resolution cameras, you can try to increase the BlockTimeout in the Stream Features in GalaxyView. (You can change "BlockTimeout" value by setting the corresponding value via the API.)

Question 6: Many incomplete frames appear when GigE cameras acquire images on the TK1 development board

Possible reasons:

The system is running in low power mode, resulting in a lot of packet loss. In Linux, the kernel developers defined a set of framework models, which is CPU Freq system, to adjust the CPU frequency dynamically. Though the front-end software is slightly different in each Linux release version, it will eventually implement the CPU frequency dynamic adjustment function through the CPU Freq system of the Linux kernel. This software will provide CPU modes (governor parameters) such as **ondemand**, **interactive**, **performance**, **powersave** and so on.

The **ondemand** is the default mode of the system; Performance is a performance mode in which the system runs at the highest frequency but with high power consumption.

Solutions:

When this problem happens, try setting the CPU mode to **performance**.

Take the ubuntu system as an example, adjust the working mode of the CPU through the cpufreq management software. The steps are as follows:

1) Install cpufrequtils management software.

The command is as follows:

sudo apt-get install cpufrequtils

2) Modify the configuration file **/etc/init.d/cpufrequtils**.

Modify **GOVERNOR="ondemand"** to **GOVERNOR="performance"**, and save it.

Then it will be taken effect when execute the following command:

/etc/init.d/cpufrequtils restart

3) Finally, you need to disable the ondemand daemon, otherwise the settings will be overwritten after reboot.

The command is as follows:

sudo update-rc.d ondemand disable

2. USB3 Vision Camera

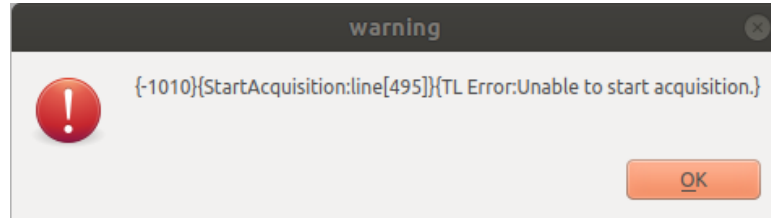Question 1: When using more than four cameras, the failure to start acquisition is shown in the Figure 2-1

Figure 2-1

Solutions:

1) Execute the command in the terminal:

   "echo 1000 > /sys/module/usbcore/parameters/usbfs_memory_mb"

2) Execute the script file **SetUSBStack.sh** in the installation package.

The settings of the above two methods are only valid for this startup, if you need to be valid all the time, you can write to the startup script.
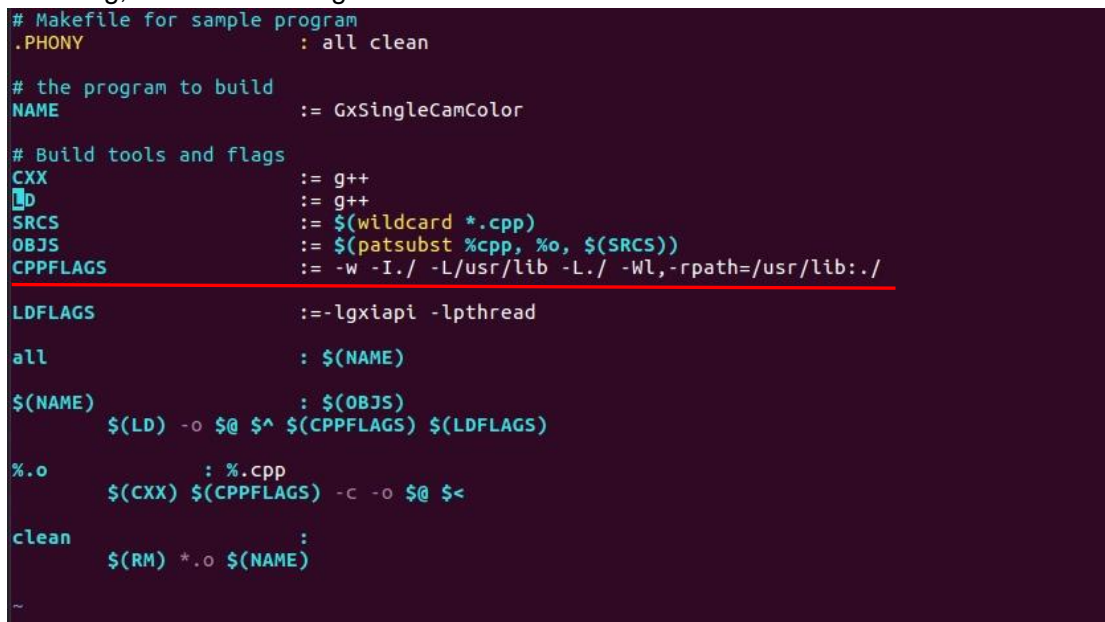
3. Installation package

Question 1: The user packages the library files in the MER SDK package into their own installation package, and the program cannot be run after installation

Solutions:

1) Add in the makefile of the user program:

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

   After adding, as shown the Figure 3-1:


```
# 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)

~
```

Figure 3-1

2) Recompile the user program.
3) Copy the installation package library provided by DAHENG IMAGING to the user installation package program directory. There are three files "**libgxiapi.so**", "**GxU3VTL.cti**", "**GxGVTL.cti**" that need to be copied.

Note: When users build their own installation package, in addition to operating the camera driver files according to the above two methods, they also need to copy the two files under the config folder generated by DAHENG IMAGING's Linux version installation package to the specified directory.

Explain the files in the config directory and the locations to be copied:

1) Under the config directory. The 99-galaxy-u3v.rules file is used to enumerate U3 cameras in non-privileged mode. The lack of this file may cause enumeration to U3 cameras only in privileged mode (sudo or root identity). The 99-galaxy-u3v.rules file needs to be copied to the system's /etc/udev/rules.d directory to take effect.

The directory of 99-galaxy-u3v.rules file copy is shown in the Figure 3-2:

Figure 3-2

2) The conf file in the galaxy-limits.conf directory is used to improve the collection performance. The file needs to be copied to the system's /etc/security/limits.d directory to take effect.

The galaxy-limits.conf file copy directory is shown in the Figure 3-3:

Figure 3-3

4. System Problem

Question 1: In the Jetson TX2 / nano R32.2.3 version system (linux 4.9 kernel), if you repeatedly start and stop the acquisition (GXStreamOn/GxStreamOff), the GXStreamOn operation will fail after a certain number of times, and it can be restored after restarting the program

Reasons:

This problem is due to the bug of TX2 R32.2.3 version system. In this version system, if thread priority is raised when the thread is starting, a memory leak will occur when the thread exits, which is manifested as a VIRT (virtual memory) leak in Linux. The top command can show you this phenomenon. When the VIRT value exceeds the allowable threshold of the system, there will be a problem that the thread cannot be started. The virtual memory is the applied memory, corresponding to the real memory, which represents the actual memory used. Real memory may affect the performance of other processes while virtual memory will only affect the application of memory in this process.

This problem occurs only when repeatedly starting and stopping acquisition scenes, about 3000 times in TX2. If encountering this problem, you can refer to the following solutions.

Solution I (recommended):

After testing, this problem only appears in the Jetson TX2/nano R32.2.3 system, and other new or old versions do not have this problem. You can avoid this problem by upgrading or reducing the system version.

Solution II:

Remove galaxy-limits.conf file by executing the following command and restart the system to solve the VIRT leak problem.

sudo rm -rf /etc/security/limits.d/galaxy-limits.conf

Note: This solution will not raise the priority of the acquisition threads inside the SDK. If the system starts too many processes or threads, the acquisition frame rate may be unstable. Please use it with caution.