



深圳市艾利光科技有限公司

Application Note

PCIE 采集卡 Demo 程序 使用方法

— Note-2.4

版本：V1.2

版权声明

Copyright @2020 - 2022 深圳市艾利光科技有限公司

公司网站: www.aili-light.com

官方旗舰店:



邮箱: sea@ailiteam.com

电话: 0755-27218150

传真: 0755-27218150

微信公众号:



版本记录

版本号	Date	Authors	修改内容记录
V1.0	2022/06/28	jimmy.xie	初版
V1.1	2022/06/29	jimmy.xie	调整了 Windows 的部分
V1.2	2022/07/06	jimmy.xie	新增 json 文件配置 sensor 的方式 (python 接口)

说明：

目录

版权声明	2
版本记录	3
1. C 版本	5
1.1. 系统环境	5
1.2. 编译	5
1.3. 运行	5
2. Python 版本	8
2.1. 系统环境	8
2.2. 依赖库	8
2.3. 运行	9
3. 常见问题:	16

1. C 版本

1.1. 系统环境

1) Linux 系统

- Ubuntu 18.04
- CMake 3.5 或以上
- gcc version 7.5.0 (Ubuntu 7.5.0-3ubuntu1~18.04)
- opencv 3.4.9 with gtk-2.0 (图像显示)

2) Windows 系统

- Windows 10/11
- CMake 3.5 或以上
- Mingw gcc version 7.3-win32 20180312 (GCC)
- Qt-5.14.10 (建议使用)

1.2. 编译

1) Linux 系统

- mkdir build
- cd build
- cmake -DCMAKE_INSTALL_PREFIX=<install path> ..
如需要图像显示 (已安装 opencv* 和 gtk2.0), 增加选项 :
-DBUILD_IMAGE_DISP=ON
- make
- make install

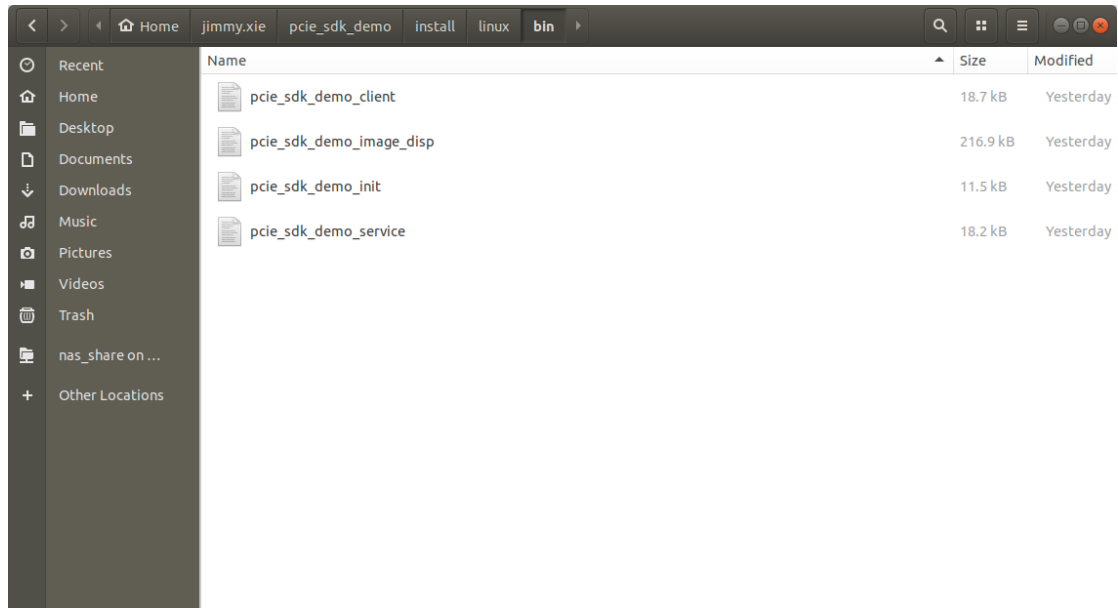
*要求 opencv 版本 < 4 (不包含 4)

2) Windows 系统

请参考 Linux 系统编译

1.3. 运行

1) 打开程序所在的目录 (Windows 下为 .exe 文件) :



2) 初始化 sdk, 输入:

```
$ sudo ./pcie_sdk_demo_init -s
```

启动成功后, 可看到如下画面:

```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/install/linux/bin
File Edit View Search Terminal Help
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/install/linux/bin$ sudo ./pc
ie_sdk_demo_init -s
pcie dev index0 start open
pcie card 0 mmap success, addr 0x7f3484fd2000
pcie dev index0 open device success
pcie dev wait and clear histort irq
pcie dev 0 open, cur dev open number 1
thread fresh test start, pcie dev 0
----->
```

如果出现错误, 请检查 PCIE 采集卡的驱动是否安装正确。

3) 配置摄像头

```
$ ./pcie_demo_service -set_config
```

(无需导入配置表, 艾利光的模组)

或:

```
$ ./pcie_demo_service -set_config_by_file <filename>
```

(如果需要导入配置表)

建议这里采用 python 版本的接口 (比较灵活, 不需要重新编译), 请参考:

【配置摄像头】

4) 开始出图

```
$ ./pcie_demo_service -stream_on
```

建议这里采用 python 版本的接口（比较灵活，不需要重新编译），请参考：
【开始出图】

5) 打开 log 文件查看是否出图

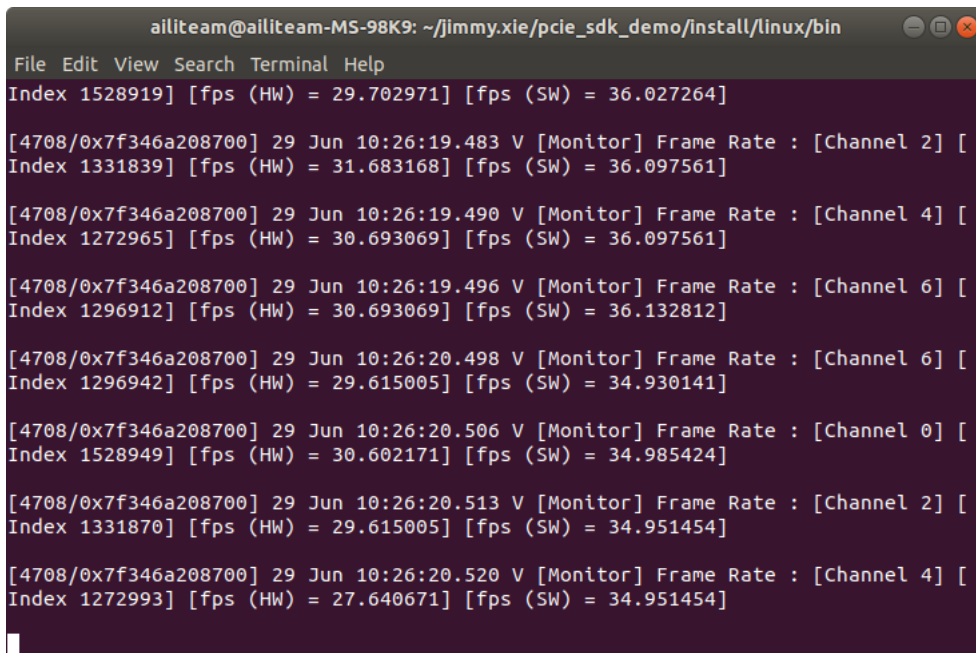
- Linux 下使用：

```
$ tail -f alg_sdk.log
```

- Windows 可使用：

```
$ Get-Content .\alg_sdk.log -Wait
```

看到如下字样，说明出图成功：



```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/install/linux/bin
File Edit View Search Terminal Help
Index 1528919] [fps (HW) = 29.702971] [fps (SW) = 36.027264]

[4708/0x7f346a208700] 29 Jun 10:26:19.483 V [Monitor] Frame Rate : [Channel 2] [
Index 1331839] [fps (HW) = 31.683168] [fps (SW) = 36.097561]

[4708/0x7f346a208700] 29 Jun 10:26:19.490 V [Monitor] Frame Rate : [Channel 4] [
Index 1272965] [fps (HW) = 30.693069] [fps (SW) = 36.097561]

[4708/0x7f346a208700] 29 Jun 10:26:19.496 V [Monitor] Frame Rate : [Channel 6] [
Index 1296912] [fps (HW) = 30.693069] [fps (SW) = 36.132812]

[4708/0x7f346a208700] 29 Jun 10:26:20.498 V [Monitor] Frame Rate : [Channel 6] [
Index 1296942] [fps (HW) = 29.615005] [fps (SW) = 34.930141]

[4708/0x7f346a208700] 29 Jun 10:26:20.506 V [Monitor] Frame Rate : [Channel 0] [
Index 1528949] [fps (HW) = 30.602171] [fps (SW) = 34.985424]

[4708/0x7f346a208700] 29 Jun 10:26:20.513 V [Monitor] Frame Rate : [Channel 2] [
Index 1331870] [fps (HW) = 29.615005] [fps (SW) = 34.951454]

[4708/0x7f346a208700] 29 Jun 10:26:20.520 V [Monitor] Frame Rate : [Channel 4] [
Index 1272993] [fps (HW) = 27.640671] [fps (SW) = 34.951454]
```

6) 显示图像

```
$ sudo ./pice_demo_image_disp -all
```

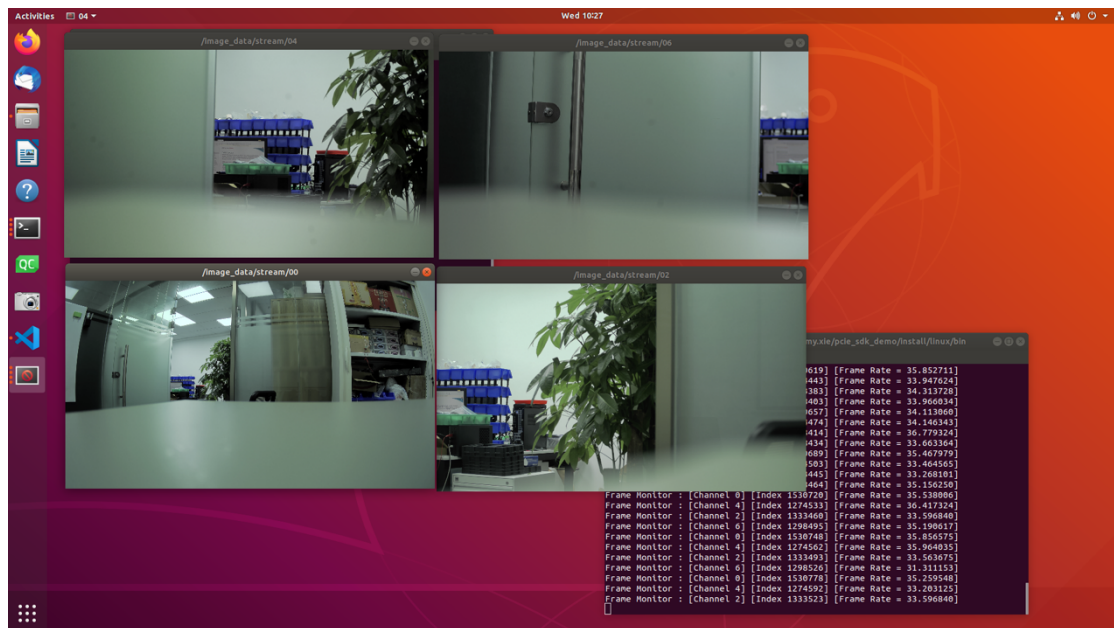
显示所有通道的图像

或：

```
$ sudo ./pice_demo_image_disp -c /image_data/stream/xx
```

显示 xx 通道（xx = [00 - 07]）的图像

如果配置正确，会弹出图像窗口：



如果图像没有显示, 请检查通道号是否正确, 以及 PCIE 采集卡是否蓝灯慢闪, 并查看 log 文件是否显示正在出图。

2. Python 版本

2.1. 系统环境

1) Linux 系统

Linux 系统要求为:

- Ubuntu 18.04
- Python 2.7.17 (default, Mar 18 2022) [GCC 7.5.0]
或 Python 3.6.9 (default, Mar 15 2022) [GCC 8.4.0]

2) Windows 系统

- Windows 10/11
- Python 3.10.5 (Jun 6 2022) [MSC v.1929 64 bit (AMD64)] on win32

2.2. 依赖库

请使用 pip 安装所需的依赖库:

```
$ pip install <lib-name>
```

如需使用其他的源 (例如清华源), 可以加上:

```
$ pip install <lib-name> -i https://pypi.tuna.tsinghua.edu.cn/simple
```

如需指定版本号, 可以加上:

```
$ pip install <lib-name==xx.xx.xx.xx>
```

xx.xx.xx.xx 为库的版本号, 例如 3.4.9.31

1) Linux

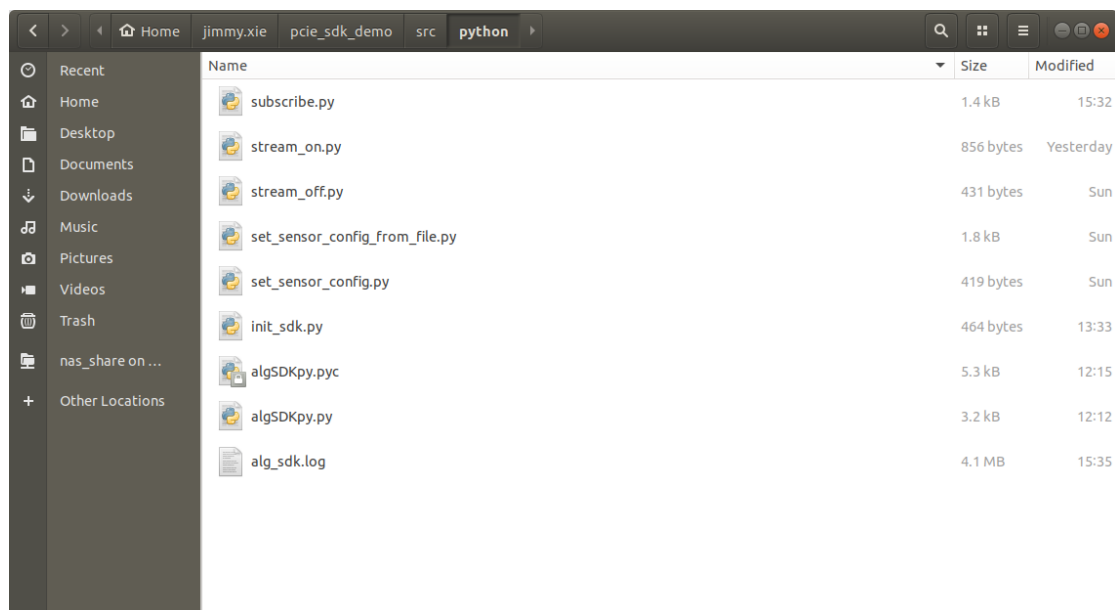
Python	2. 7. 17
numpy	1. 16. 6
opencv-python	3. 4. 9

2) Windows

Python	3. 10. 5
numpy	1. 23. 0
opencv-python	4. 6. 0

2.3. 运行

1) 打开包含脚本的文件夹：



2) 打开终端 Terminal （Windows 下可用 PowerShell）

如果是 Linux 系统，输入：

```
$ sudo python init_sdk.py
```

可以看到运行结果如下：

```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/src/python
File Edit View Search Terminal Help
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$ sudo python init_sdk.py
[sudo] password for ailiteam:
pcie dev index0 start open
pcie card 0 mmap success, addr 0x7f247ef86000
pcie dev index0 open device success
pcie dev wait and clear histort irq
pcie dev 0 open, cur dev open number 1
thread fresh test start, pcie dev 0
----->
```

如果是 Windows 系统，输入：
\$ python init_sdk.py

```
Windows PowerShell
Windows 终端可在设置中设置为默认终端应用程序。 打开设置

Windows PowerShell
版权所有 (C) Microsoft Corporation. 保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS D:\jimmy.xie\pcie_sdk_demo\src\python> python .\init_sdk.py
pcie dev index0 start open
Devices found: 1
Device base path: \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}
Device node: h2c_0, path \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}\h2c_0
device h2c_0 handle 23c
Devices found: 1
Device base path: \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}
Device node: c2h_0, path \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}\c2h_0
device c2h_0 handle 200
Devices found: 1
Device base path: \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}
Device node: h2c_1, path \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}\h2c_1
device h2c_1 handle 204
Devices found: 1
Device base path: \\?\pci#ven_10ee&dev_9034&subsys_000710ee&rev_00#4&2f6c0d12&0&0008#{74c7e4a9-6d5d-4a70-bc0d-20691dff9e9d}
```

3) 配置摄像头

- 方式一：

\$ python set_sensor_config.py

该方法仅针对艾利光的模组

- 方式二：

\$ python set_sensor_config_from_file.py

该方法通过读取配置文件导入 sensor 的配置表，适用于所有类型的模组（包括非艾利光的模组）

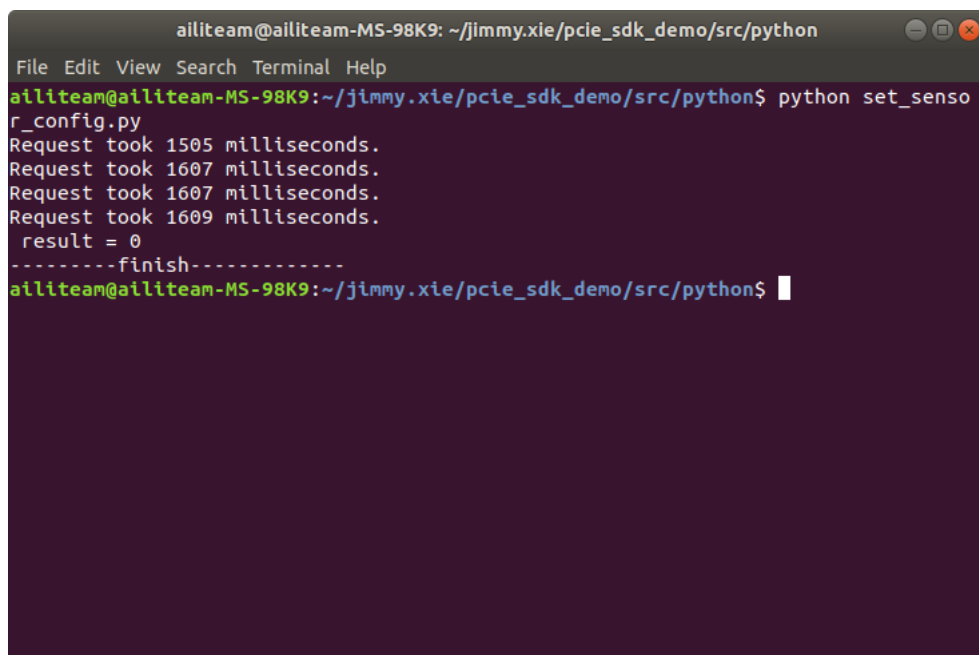
- 方式三:

```
$ python set_sensor_from_json.py --json_file=<path_to_json_file>  
--channel=xx
```

该方式通过 json 文件导入配置, <path_to_json_file> 即 json 文件的地址 (可用相对路径), xx 表示配置的解串器通道号(这里的设置优先级高于 json 文件中的解串器通道号)。

方式三的好处是无需提前修改 python 文件, 只要找到对应的 json 文件, 把路径提供给 json_file 这个参数即可。

运行结果如下 (方式一):



```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/src/python  
File Edit View Search Terminal Help  
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$ python set_sens  
r_config.py  
Request took 1505 milliseconds.  
Request took 1607 milliseconds.  
Request took 1607 milliseconds.  
Request took 1609 milliseconds.  
result = 0  
-----finish-----  
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$
```

方式三:

```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/src/python
File Edit View Search Terminal Help
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$ python set_sens
r_from_json.py --json_file="../../config/alg_isx_021_2_ch.json" --channel=0
Set for channel 0
Request took 3007 milliseconds.
result = 0
-----finish-----
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$
```

如果看到 `result = 0` 的返回值，说明配置下发成功。

此时可以检查采集卡的通道指示灯是否为蓝灯慢闪，如果是，说明配置执行成功，否则可能是 PCIE 采集卡的硬件问题，或者通道配置错误。

4) 开始出图

- 方式一：

```
$ python stream_on.py
```

该方法需要提前修改好 `stream_on.py` 文件中对应的通道编号

- 方式二：

```
$ python stream_on_by_channel.py --channel='x,y,z,w'
```

该方法无需修改 python 文件，只要在命令中加 `channel` 参数即可，这里的 `x, y, z, w` 表示要出图的通道号，例如 `'0, 2, 4, 6'` 等等。

可以看到结果如下（方法一）：

```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/src/python
File Edit View Search Terminal Help
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$ python set_sens
r_config.py
Request took 1505 milliseconds.
Request took 1607 milliseconds.
Request took 1607 milliseconds.
Request took 1609 milliseconds.
result = 0
-----finish-----
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$ python stream_on
.py
Request took 101 milliseconds.
result = 0
-----finish-----
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$
```

方式二:

```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/src/python
File Edit View Search Terminal Help
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$ python stream_on
_by_channel.py --channel='0,1'
stream on channel [0]
stream on channel [1]
Request took 101 milliseconds.
result = 0
-----finish-----
ailiteam@ailiteam-MS-98K9:~/jimmy.xie/pcie_sdk_demo/src/python$
```

如果看到 `result = 0` 的返回值, 说明出图指令下发成功。

5) 打开 log 文件查看是否出图

- Linux 下使用:

```
$ tail -f alg_sdk.log
```

- Windows 可使用:

```
$ Get-Content .\alg_sdk.log -Wait
```

看到如下字样, 说明出图成功:

```
ailiteam@ailiteam-MS-98K9: ~/jimmy.xie/pcie_sdk_demo/install/linux/bin
File Edit View Search Terminal Help
Index 1528919] [fps (HW) = 29.702971] [fps (SW) = 36.027264]

[4708/0x7f346a208700] 29 Jun 10:26:19.483 V [Monitor] Frame Rate : [Channel 2] [
Index 1331839] [fps (HW) = 31.683168] [fps (SW) = 36.097561]

[4708/0x7f346a208700] 29 Jun 10:26:19.490 V [Monitor] Frame Rate : [Channel 4] [
Index 1272965] [fps (HW) = 30.693069] [fps (SW) = 36.097561]

[4708/0x7f346a208700] 29 Jun 10:26:19.496 V [Monitor] Frame Rate : [Channel 6] [
Index 1296912] [fps (HW) = 30.693069] [fps (SW) = 36.132812]

[4708/0x7f346a208700] 29 Jun 10:26:20.498 V [Monitor] Frame Rate : [Channel 6] [
Index 1296942] [fps (HW) = 29.615005] [fps (SW) = 34.930141]

[4708/0x7f346a208700] 29 Jun 10:26:20.506 V [Monitor] Frame Rate : [Channel 0] [
Index 1528949] [fps (HW) = 30.602171] [fps (SW) = 34.985424]

[4708/0x7f346a208700] 29 Jun 10:26:20.513 V [Monitor] Frame Rate : [Channel 2] [
Index 1331870] [fps (HW) = 29.615005] [fps (SW) = 34.951454]

[4708/0x7f346a208700] 29 Jun 10:26:20.520 V [Monitor] Frame Rate : [Channel 4] [
Index 1272993] [fps (HW) = 27.640671] [fps (SW) = 34.951454]
```

6) 显示图像

- Linux 下使用:

\$ sudo python subscribe.py

注意需要订阅的图像话题，需要在 subscribe.py 文件中修改，默认是全部订阅：

```
Open subscribe.py
~/jimmy.xie/pcie_sdk_demo/src/python
Save

import signal, sys
import time
import numpy as np
import ctypes
from ctypes import *
import cv2

import algSDKpy
from algSDKpy import algSDKClient
from algSDKpy import pcie_image_data_t
from algSDKpy import callbackFunc_t

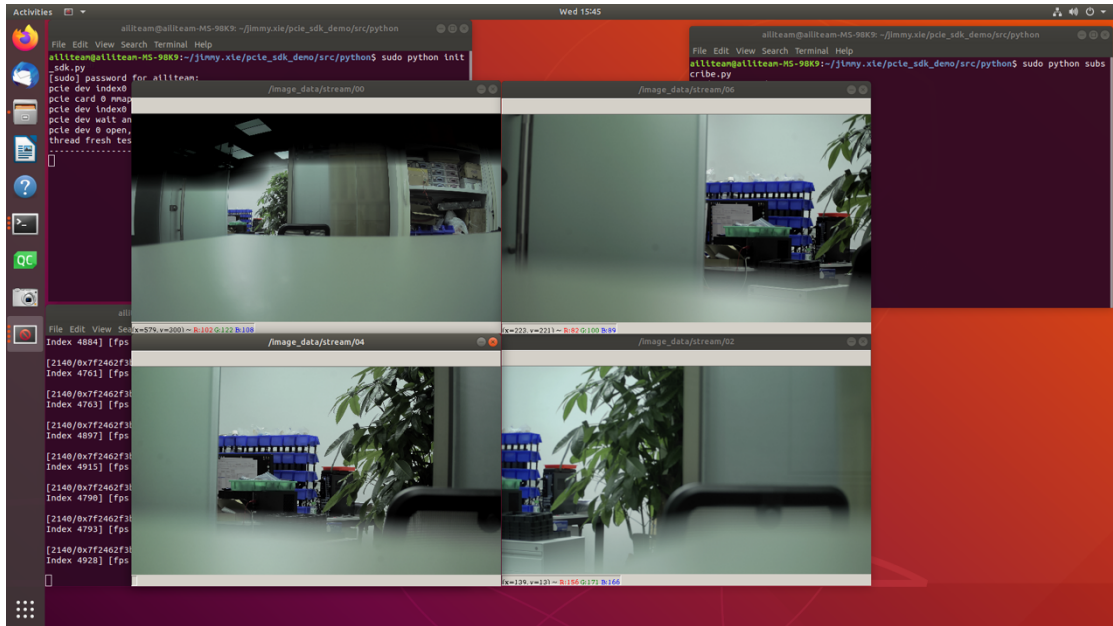
client = algSDKClient()
topic = b"/image_data/stream"

def array2mat(w, h, payload, image_name):
    p_array = np.frombuffer(np.ctypeslib.as_array(payload, shape=((h*w*2, 1, 1))), dtype=np.uint8)
    img_in = p_array.reshape((h, w, 2))
    img_out = cv2.cvtColor(img_in, cv2.COLOR_YUV2BGR_YUYV)
    img_disp = cv2.resize(img_out, (640, 360))
    cv2.imshow(image_name, img_disp)
    cv2.waitKey(1)
    # print(image_name)

def CallbackFunc(ptr):
    p = ctypes.cast(ptr, ctypes.POINTER(pcie_image_data_t))
    payload = ctypes.cast(p.contents.payload, ctypes.POINTER(c_uint8))
    # print('frame = %d, time %d, byte_0 = %d, byte_end = %d' % (p.contents.image_info_meta.frame_index,
    p.contents.image_info_meta.img_size, payload[0], payload[p.contents.image_info_meta.img_size-1]))
    array2mat(p.contents.image_info_meta.width, p.contents.image_info_meta.height, payload, p.contents.common_head.topic_name)

if __name__ == '__main__':
    callback_func = callbackFunc_t(CallbackFunc)
    client.Subscribe(topic, callback_func)
    ret = client.InitClient()
```

如果配置正确，可以看到图像数据已显示：

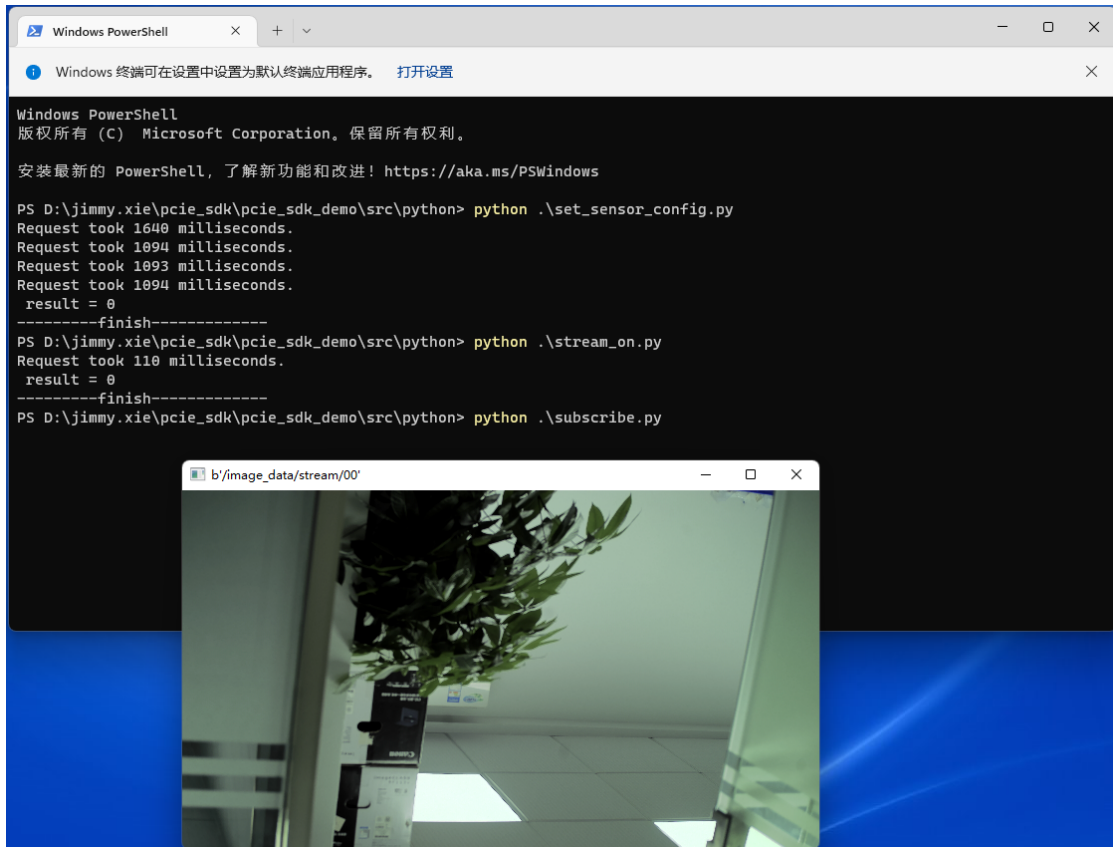


如果图像没有显示, 请检查通道号是否正确, 以及 PCIE 采集卡是否蓝灯慢闪, 并查看 log 文件是否显示正在出图。

- Windows 下使用:

\$ python subscribe.py

显示结果如下:




```
{
  "sensor_cfg": {
    "sensor_name": "isx_021",
    "sensor_width": "1920",
    "sensor_height": "1280",
    "config_table": "../../../config/table/ALG_made_9296_ISX021_9295a_MIPI_4LANES_2_ch.txt"
  },
  "deserdes_cfg": {
    "channel_id": "0",
    "des_mode": "1",
    "camera_num": "2",
    "data_type": "0"
  }
}
```