



Using YUAN's Captured Card on Nvidia's Clara IGX

For Holohub v2.0.0

Install Guide v2.0.0_2

Content

USING YUAN'S CAPTURED CARD ON NVIDIA'S CLARA AGX	3
1. YUAN'S PACKAGES	3
2. INTRODUCE.....	3
3. PREPARE HARDWARE	5
3.1 SETUP HDD OR SSD	5
3.2 INSTALL CAPTURED CARD.....	5
3.2.1 VERIFY - THE INSTALLATION OF CAPTURED CARD.....	7
3.3 ADDITIONAL SETUP – ENABLING EXCLUSIVE DISPLAY MODE	8
4. YUAN'S PACKAGE DELOPY.....	9
4.1 PREREQUIREMENT.....	9
4.2 INSTALL THE DRIVER OF CAPTURE CARD AND THE QCAP SDK.....	10
4.3 VERIFY – INSTALLATION OF CAPTURE DRIVER.....	11
5. DOWNLOAD AND BUILD HOLOHUB	12
5.1 DOWNLOAD HOLOHUB	12
6. RUN HOLOHUB APPLICATIONS.	13
6.1 COLONOSCOPY SEGMENTATION	13
6.1.1 BUILD COLONOSCOPY SEGMENTATION	13
6.1.2 RUN COLONOSCOPY SEGMENTATION (PYTHON)	14
6.2 ENDOSCOPY TOOL TRACKING	15
6.2.1 BUILD ENDOSCOPY TOOL TRACKING.....	15
6.2.2 RUN ENDOSCOPY TOOL TRACKING (CPP).....	16
6.2.3 RUN ENDOSCOPY TOOL TRACKING (PYTHON).....	17
6.3 MULTIAI ULTRASOUND	18
6.3.1 BUILD MULTIAI ULTRASOUND	18
6.3.2 RUN MULTIAI ULTRASOUND (CPP)	19
6.3.3 RUN MULTIAI ULTRASOUND (PYTHON)	20
6.4 ULTRASOUND SEGMENTATION.....	21
6.4.1 BUILD ULTRASOUND SEGMENTATION.....	21
6.4.2 RUN ULTRASOUND SEGMENTATION (CPP).....	22
6.4.3 RUN ULTRASOUND SEGMENTATION (PYTHON)	23
7. CHANGE VIDEO RESOLUTION OR SELECT INPUT TYPE (VIDEO: CH7_SWITCH_INPUT_TYPE.MP4).....	24
8. TROUBLESHOOTINEG	27
8.1 CASE 1: SHOW "No QCAP SDK"	27
8.2 CASE 2: A DIALOG SHOW "HOLOVIZ" IS NOT RESPONDING.....	28
8.3 CASE 3: SHOW "No Device"	29
8.4 CASE 4: NO VIDEO AND 'FIND_VIDEO_DEV_PATH FAIL'	30
8.5 CASE 5: SHOW "NO SIGNAL"	31
8.6 CASE 6: NO VIDEO AND 'NO SIGNAL DETECTED'	32

Using YUAN's Captured Card on Nvidia's Clara AGX/IGX

1. YUAN's Packages

No.	File name	Purpose
1	qcap_holoscan_igx-sw_pack_v2.0.0_20240705.tar.gz	Pack zip contains below files
2	lxv4l2d-sc0400-sc0710_v1330-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb	Driver for Holoscan IGX
3	qcap-sdk_1.88.7_1-igx-r36.1.0_aarch64.deb	QCAP SDK

2. Introduce

2.1 The big picture of this guide

Need follow this guide step by step to setup whole software stack. Chapter 3 and chapter 4 has a verification section, please make sure verification is pass and then goto next chapter.

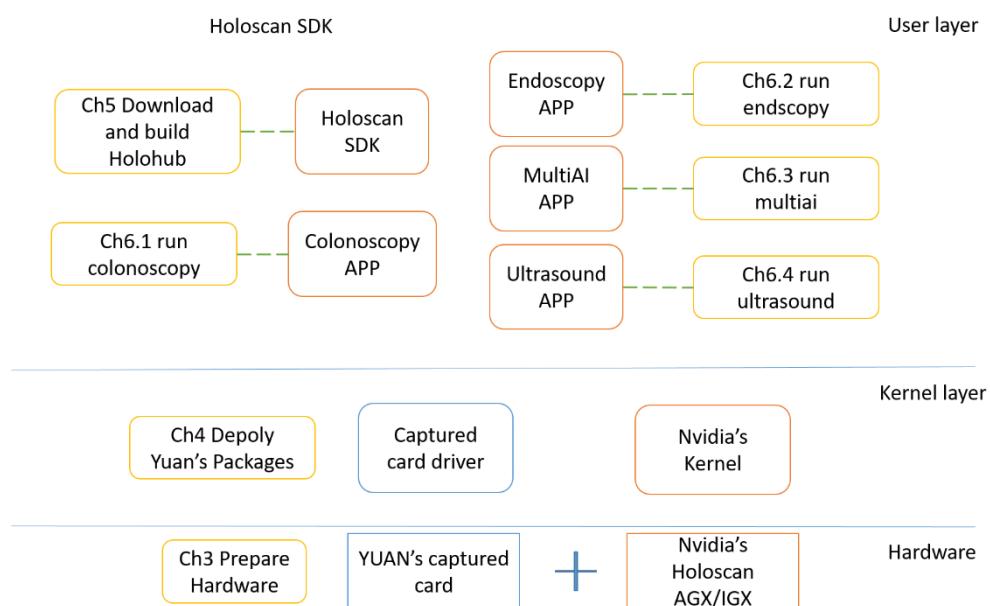
Ch3 Prepare hardware - introduce how to install YUAN's captured card into Nvidia's Clara AGX/IGX.

Ch4 Deploy YUAN's packages - to install YUAN captured card driver and QCAP SDK.

Ch5 Download and install Holohub - download and setup Holohub

Ch6 Run Holohub applications - run applications

Ch7 Select input type - select different input type of capture card.



Ch8 Troubleshooting - how to fix some errors and get help from YUAN.

2.2 Tutorial Videos

- ch6.1 build and run colonoscopy_segmentation - How to build and run colonoscopy python demo.
- ch6.2 build and run endoscopy_tool_tracking cpp - How to build and run endoscopy cpp demo.
- ch6.2 build and run endoscopy_tool_tracking python - How to build and run endoscopy python demo.
- ch6.3 build and run multiai_ultrasound cpp - How to build and run multiai cpp demo.
- ch6.3 build and run multiai_ultrasound python - How to build and run multiai python demo.
- ch6.4 build and run ultrasound_segmentation cpp - How to build and run ultrasound cpp demo.
- ch6.4 build and run ultrasound_segmentation python - How to build and run ultrasound python demo.

3. Prepare Hardware

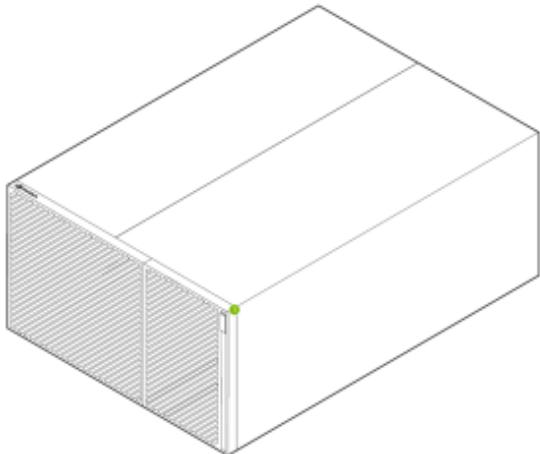
3.1 Setup HDD or SSD

Must follow Holoscan SDK user guide to setup HDD or SSD and docker.

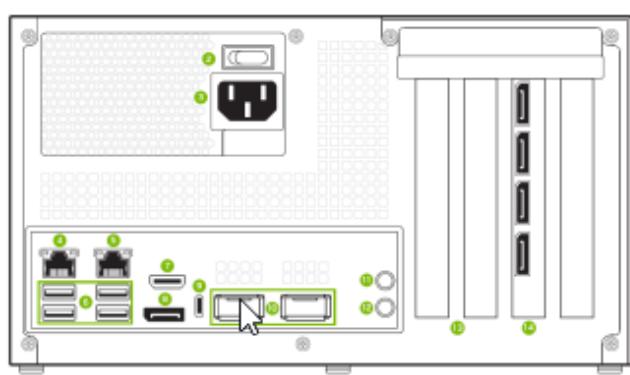
3.2 Install captured card

Install captured card on PCIe Gen5 single width slots (item 13)

Front View



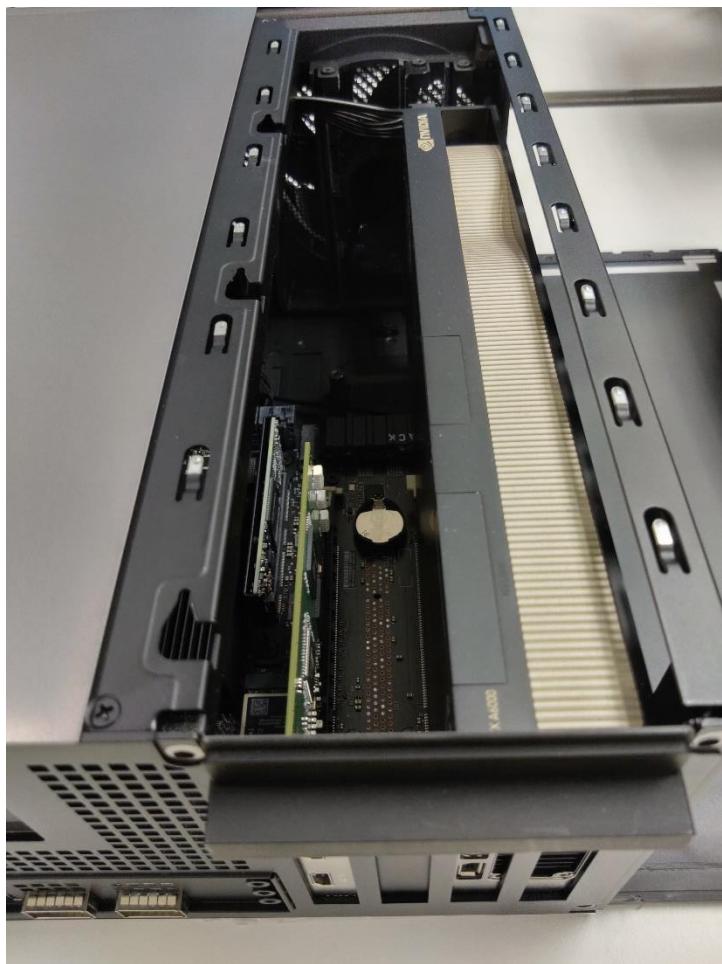
Back View



1. Power button
2. Power switch
3. Power cable connection
4. 1GbE RJ45 with connection to the Orin CCPLEX
5. 1GbE RJ45 with connection to the Orin CCPLEX and BMC
6. (4x) USB 3.2 Gen2 Type A
7. HDMI-In (There is no display output from the HDMI port)
8. DisplayPort output from Tegra
9. USB 3.2 Gen2 Type C connector
10. (2x) QSFP28 ports (up to 100GbE per port)
11. Line-Out
12. MIC
13. PCIe Gen5 single width slots(x8 connected)
14. PCIe Gen5 double width slot(x16 connected) for Optional RTX A6000.
 - There will be (4x) DisplayPorts on this slot if you selected the additional RTX A6000 when ordering from one of our distributors.



Back-view of the capture card installation.



Top-view of the capture card instation.

3.2.1 Verify - the installation of captured card.

Use lspci command to check the captured card has been successfully installed. Must have one PCI device has the name, “YUAN High-Tech Development Co., LTD”.

If can't find any captured cards, try to power off and re-insert the card again. Or bring the captured card to another machine, use the same command to make sure this card is good. If still cannot fix it, please contact YUAN to get help.

```
yuan@ubuntu:~$ lspci
0000:00:00.0 PCI bridge: NVIDIA Corporation Device 1ad0 (rev a1)
0000:01:00.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:02:00.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:02:01.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:02:02.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:03:00.0 Ethernet controller: Mellanox Technologies MT28908 Family [ConnectX-6]
0000:03:00.1 Ethernet controller: Mellanox Technologies MT28908 Family [ConnectX-6]
0000:04:00.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:05:08.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:06:00.0 Multimedia controller: YUAN High-Tech Development Co., Ltd. Device 0710
0000:07:00.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:08:00.0 PCI bridge: Mellanox Technologies MT28908 Family [ConnectX-6 PCIe Bridge]
0000:09:00.0 VGA compatible controller: NVIDIA Corporation TU102GL [Quadro RTX 6000/8000] (rev a1)
0000:09:00.1 Audio device: NVIDIA Corporation TU102 High Definition Audio Controller (rev a1)
0000:09:00.2 USB controller: NVIDIA Corporation TU102 USB 3.1 Host Controller (rev a1)
0000:09:00.3 Serial bus controller [0c80]: NVIDIA Corporation TU102 USB Type-C UCSI Controller (rev a1)
0001:00:00.0 PCI bridge: NVIDIA Corporation Device 1ad2 (rev a1)
0001:01:00.0 SATA controller: Marvell Technology Group Ltd. Device 9171 (rev 13)
```

3.3 Additional Setup – Enabling Exclusive Display Mode

To enable exclusive display mode, connect two displays to Clara AGX. And one display switch to disable in Nvidia settings.

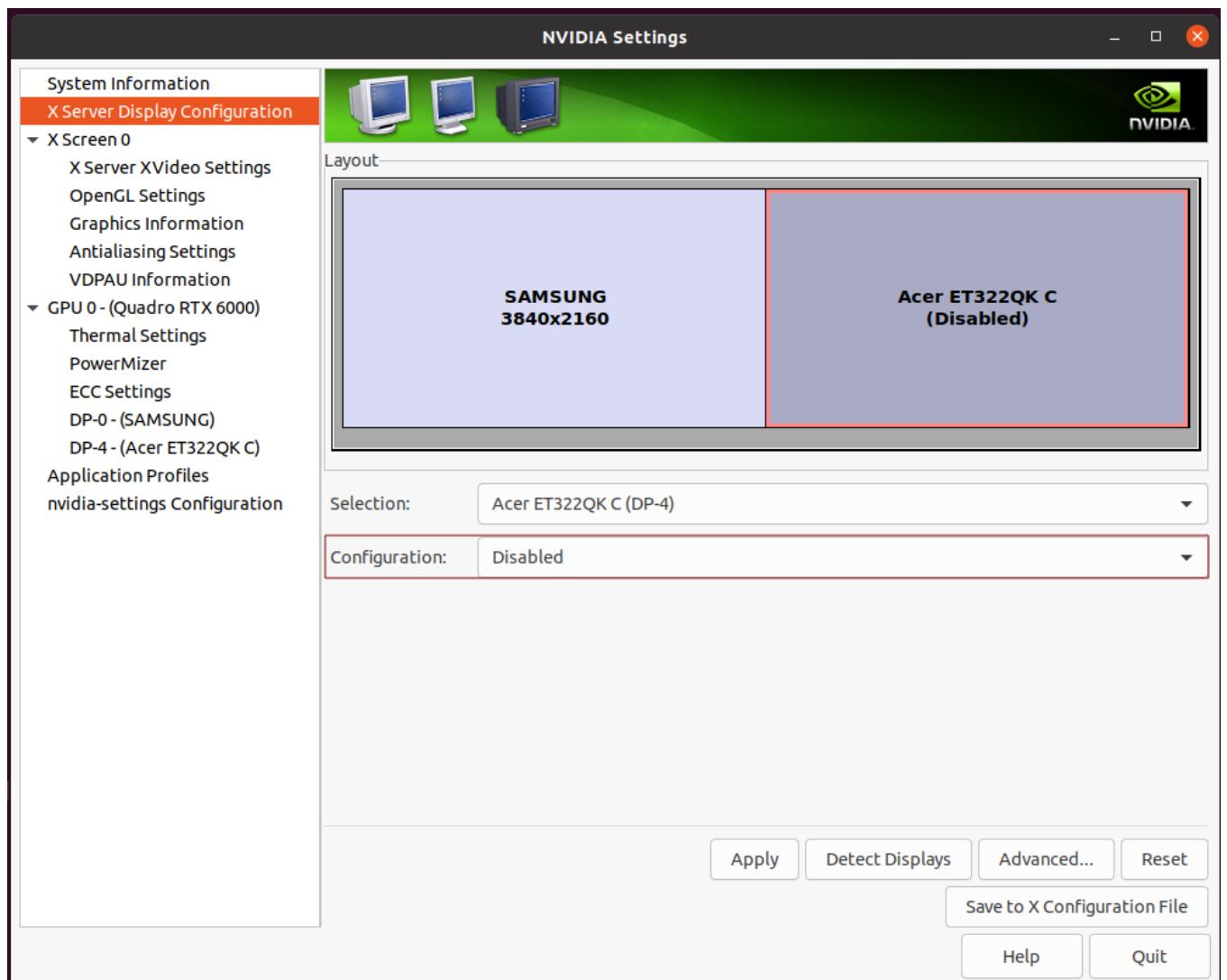


Figure: Two displays and Acer ET322QK is disable in nvidia settings.

Refer to the [Additional Setup - NVIDIA Docs](#) for Detail

4. Yuan's Package Deploy.

4.1 Prerequisite

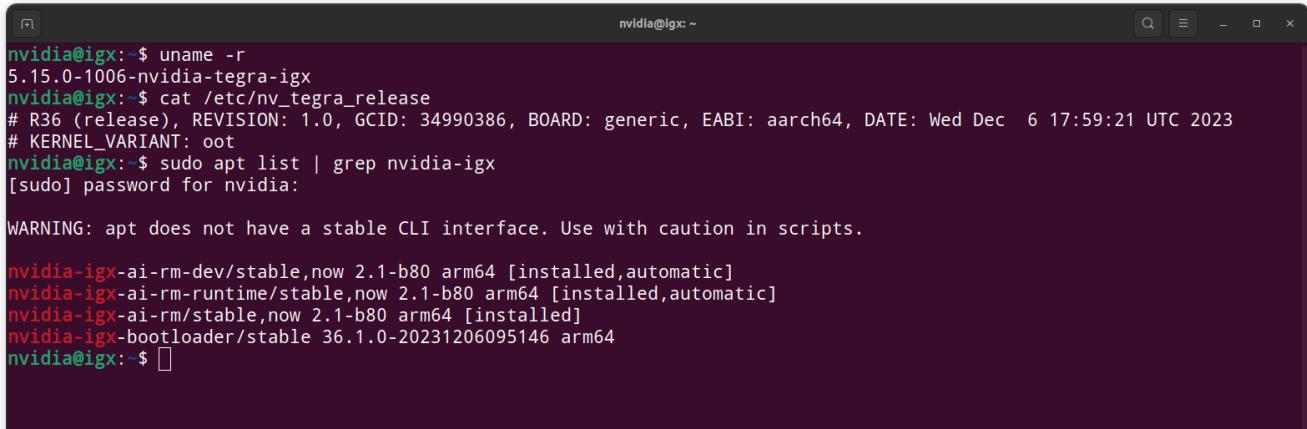
This version need IGX-SW 1.0 Devepoler Preview.

Follow this URL to install this OS, <https://developer.nvidia.com/igx-downloads>

Kernel version: 5.15.0-1006-nvidia-tegra-igx

L4T version: R36.1.0

Has an nvidia-igx-ai-rm package.

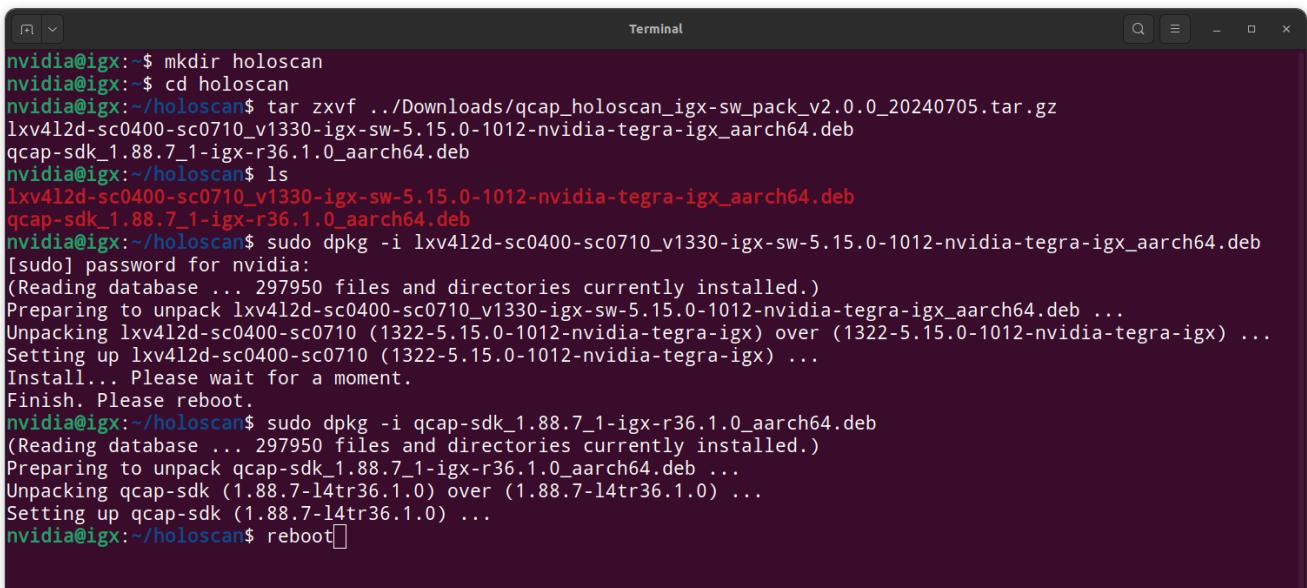


```
nvidia@igx:~$ uname -r
5.15.0-1006-nvidia-tegra-igx
nvidia@igx:~$ cat /etc/nv_tegra_release
# R36 (release), REVISION: 1.0, GCID: 34990386, BOARD: generic, EABI: aarch64, DATE: Wed Dec  6 17:59:21 UTC 2023
# KERNEL_VARIANT: oot
nvidia@igx:~$ sudo apt list | grep nvidia-igx
[sudo] password for nvidia:
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

nvidia-igx-ai-rm-dev/stable,now 2.1-b80 arm64 [installed,automatic]
nvidia-igx-ai-rm-runtime/stable,now 2.1-b80 arm64 [installed,automatic]
nvidia-igx-ai-rm/stable,now 2.1-b80 arm64 [installed]
nvidia-igx-bootloader/stable 36.1.0-20231206095146 arm64
nvidia@igx:~$
```

4.2 Install the driver of capture card and the QCAP SDK.

```
$ mkdir holoscan  
$ cd holoscan  
$ tar zxvf ~/Downloads/qcap_holoscan_igx-sw_pack_v2.0.0_20240624.tar.gz  
# kernel must be 5.15.0-1012-nvidia-tegra-igx  
$ sudo dpkg -i lxv4l2d-sc0400-sc0710_v1322-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb  
$ sudo dpkg -i qcap-sdk_1.88.7-igx-r36.1.0_aarch64  
$ sudo reboot
```



The screenshot shows a terminal window with the title 'Terminal'. The terminal output is as follows:

```
nvidia@igx:~$ mkdir holoscan  
nvidia@igx:~$ cd holoscan  
nvidia@igx:~/holoscan$ tar zxvf ../Downloads/qcap_holoscan_igx-sw_pack_v2.0.0_20240705.tar.gz  
lxv4l2d-sc0400-sc0710_v1330-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb  
qcap-sdk_1.88.7_1-igx-r36.1.0_aarch64.deb  
nvidia@igx:~/holoscan$ ls  
lxv4l2d-sc0400-sc0710_v1330-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb  
qcap-sdk_1.88.7_1-igx-r36.1.0_aarch64.deb  
nvidia@igx:~/holoscan$ sudo dpkg -i lxv4l2d-sc0400-sc0710_v1330-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb  
[sudo] password for nvidia:  
(Reading database ... 297950 files and directories currently installed.)  
Preparing to unpack lxv4l2d-sc0400-sc0710_v1330-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb ...  
Unpacking lxv4l2d-sc0400-sc0710 (1322-5.15.0-1012-nvidia-tegra-igx) over (1322-5.15.0-1012-nvidia-tegra-igx) ...  
Setting up lxv4l2d-sc0400-sc0710 (1322-5.15.0-1012-nvidia-tegra-igx) ...  
Install... Please wait for a moment.  
Finish. Please reboot.  
nvidia@igx:~/holoscan$ sudo dpkg -i qcap-sdk_1.88.7_1-igx-r36.1.0_aarch64.deb  
(Reading database ... 297950 files and directories currently installed.)  
Preparing to unpack qcap-sdk_1.88.7_1-igx-r36.1.0_aarch64.deb ...  
Unpacking qcap-sdk (1.88.7-14tr36.1.0) over (1.88.7-14tr36.1.0) ...  
Setting up qcap-sdk (1.88.7-14tr36.1.0) ...  
nvidia@igx:~/holoscan$ reboot
```

4.3 Verify – Installation of capture driver

The modprobe command is pass and lsmod can find the driver, LXV4L2D_SC0710. Has a device node, /dev/video0 (on Holoscan AGX) or /dev/video1 (on Holoscan IGX), and v4l2-ctl -DVI can get information from the captured card. If command is failed or no device node, /dev/video0 or /dev/video1, please follow the sections of [Install captured card](#).

```
$ sudo apt install -y v4l-utils  
# If platform is Holoscan AGX, use device 0  
$ v4l2-ctl -d 0 -DVI  
# If platform is Holoscan IGX, use device 1  
$ v4l2-ctl -d 1 -DVI
```

```
nvidia@igx:~/holoscan$ sudo apt install -y v4l-utils  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
v4l-utils is already the newest version (1.22.1-2build1).  
0 upgraded, 0 newly installed, 0 to remove and 329 not upgraded.  
nvidia@igx:~/holoscan$ v4l2-ctl -d 0 -DVI  
Driver Info:  
    Driver name      : LXV4L2D_SC0710  
    Card type       : SC0710:RAW 00.00 0201f718  
    Bus info        : PCI Bus 0005:06 12ab0710  
    Driver version   : 13.3.0  
    Capabilities    : 0x84220001  
        Video Capture  
        Audio  
        Streaming  
        Extended Pix Format  
        Device Capabilities  
    Device Caps     : 0x04220001  
        Video Capture  
        Audio  
        Streaming  
        Extended Pix Format  
Video input : 4 (SDI INPUT(4): ok)  
Format Video Capture:  
    Width/Height     : 3840/2160  
    Pixel Format    : 'Y210'  
    Field           : None  
    Bytes per Line  : 9600  
    Size Image      : 20736000  
    Colorspace       : Rec. 709  
    Transfer Function: Default (maps to Rec. 709)  
    YCbCr/HSV Encoding: Default (maps to Rec. 709)  
    Quantization     : Default (maps to Full Range)  
    Flags            :  
nvidia@igx:~/holoscan$
```

If driver still fail to run, please contact YUAN with below information to get help.

- The model of capture card: SC0710 N1 HDV or SC0710 N1 12G-SDI, etc.
- Dmesg log, by command ‘dmesg | tee dmesg.log’

5. Download and build Holohub

5.1 Download HoloHub

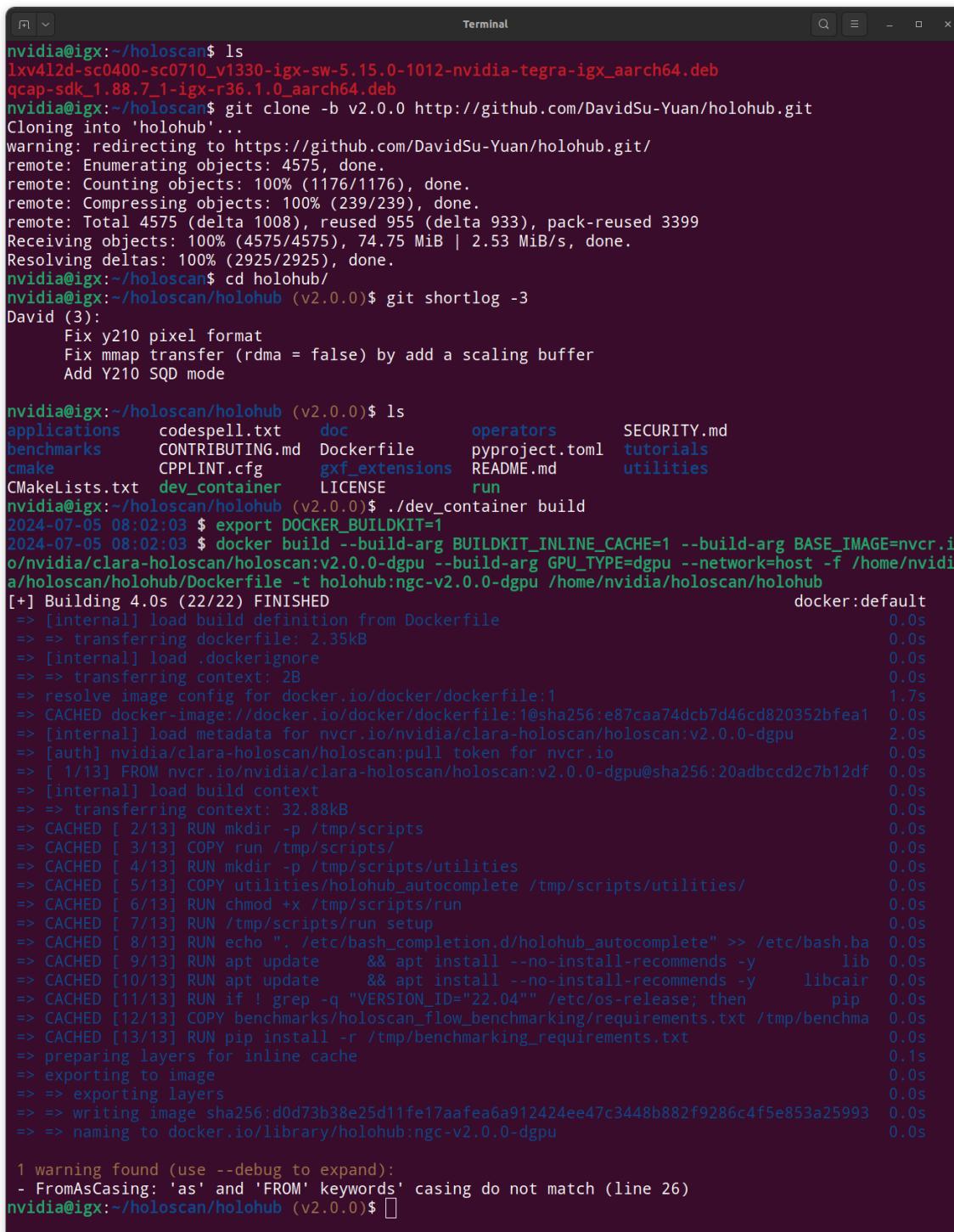
Must download from DavidSu-Yuan's repository which forked from Nvidia's repository and add the support of Yuan capture card.

```
$ git clone -b v2.0.0 http://github.com/DavidSu-Yuan/holohub.git
```

```
$ cd holohub
```

```
$ ./dev_container build
```

Make sure there are no errors when run the commands.



```
nvidia@igx:~/holoscan$ ls
lxv4l2d-sc0400-sc0710_v1330-igx-sw-5.15.0-1012-nvidia-tegra-igx_aarch64.deb
qcap-sdk_1.88.7_1-igx-r36.1.0_aarch64.deb
nvidia@igx:~/holoscan$ git clone -b v2.0.0 http://github.com/DavidSu-Yuan/holohub.git
Cloning into 'holohub'...
warning: redirecting to https://github.com/DavidSu-Yuan/holohub.git/
remote: Enumerating objects: 4575, done.
remote: Counting objects: 100% (1176/1176), done.
remote: Compressing objects: 100% (239/239), done.
remote: Total 4575 (delta 1008), reused 955 (delta 933), pack-reused 3399
Receiving objects: 100% (4575/4575), 74.75 MiB | 2.53 MiB/s, done.
Resolving deltas: 100% (2925/2925), done.
nvidia@igx:~/holoscan$ cd holohub/
nvidia@igx:~/holoscan/holohub (v2.0.0)$ git shortlog -3
David (3):
    Fix y210 pixel format
    Fix mmap transfer (rdma = false) by add a scaling buffer
    Add Y210 SQD mode

nvidia@igx:~/holoscan/holohub (v2.0.0)$ ls
applications  codespell.txt  doc          operators      SECURITY.md
benchmarks    CONTRIBUTING.md Dockerfile    pyproject.toml tutorials
cmake         CPPLINT.cfg   gxf_extensions README.md   utilities
CMakeLists.txt dev_container LICENSE      run
nvidia@igx:~/holoscan/holohub (v2.0.0)$ ./dev_container build
2024-07-05 08:02:03 $ export DOCKER_BUILDKIT=1
2024-07-05 08:02:03 $ docker build --build-arg BUILDKIT_INLINE_CACHE=1 --build-arg BASE_IMAGE=nvcr.io/nvidia/clara-holoscan:holoscan:v2.0.0-dgpu --build-arg GPU_TYPE=dgpu --network=host -f /home/nvidia/holoscan/holohub/Dockerfile -t holohub:ngc-v2.0.0-dgpu /home/nvidia/holoscan/holohub
[+] Building 4.0s (22/22) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                      0.0s
=> => transferring dockerfile: 2.35kB                                    0.0s
=> [internal] load .dockerignore                                         0.0s
=> => transferring context: 2B                                         0.0s
=> resolve image config for docker.io/docker/dockerfile:1                1.7s
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:e87caa74dc7d46cd820352bfea1 0.0s
=> [internal] load metadata for nvcr.io/nvidia/clara-holoscan:v2.0.0-dgpu        2.0s
=> [auth] nvidia/clara-holoscan:pull token for nvcr.io                         0.0s
=> [ 1/13] FROM nvcr.io/nvidia/clara-holoscan:holoscan:v2.0.0-dgpu@sha256:20adbcc2c7b12df 0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 32.88kB                                     0.0s
=> CACHED [ 2/13] RUN mkdir -p /tmp/scripts                           0.0s
=> CACHED [ 3/13] COPY run /tmp/scripts/                                0.0s
=> CACHED [ 4/13] RUN mkdir -p /tmp/scripts/utilities                  0.0s
=> CACHED [ 5/13] COPY utilities/holohub_autocomplete /tmp/scripts/utilities/ 0.0s
=> CACHED [ 6/13] RUN chmod +x /tmp/scripts/run                         0.0s
=> CACHED [ 7/13] RUN /tmp/scripts/run setup                            0.0s
=> CACHED [ 8/13] RUN echo ". /etc/bash_completion.d/holohub_autocomplete" >> /etc/bash.bashrc 0.0s
=> CACHED [ 9/13] RUN apt update && apt install --no-install-recommends -y libcairn0 0.0s
=> CACHED [10/13] RUN apt update && apt install --no-install-recommends -y libcairn0 0.0s
=> CACHED [11/13] RUN if ! grep -q "VERSION_ID='22.04'" /etc/os-release; then pip 0.0s
=> CACHED [12/13] COPY benchmarks/holoscan_flow_benchmarking/requirements.txt /tmp/benchmarks/ 0.0s
=> CACHED [13/13] RUN pip install -r /tmp/benchmarking_requirements.txt 0.0s
=> preparing layers for inline cache                                     0.1s
=> exporting to image                                                 0.0s
=> => exporting layers                                              0.0s
=> => writing image sha256:d0d73b38e25d11fe17aafea6a912424ee47c3448b882f9286c4f5e853a25993 0.0s
=> => naming to docker.io/library/holohub:ngc-v2.0.0-dgpu                 0.0s

1 warning found (use --debug to expand):
- FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 26)
nvidia@igx:~/holoscan/holohub (v2.0.0)$ 
```

6. Run Holohub applications.

6.1 Colonoscopy segmentation

6.1.1 Build colonoscopy segmentation

```
$ cd holohub
$ ./dev_container launch
$ ./run build colonoscopy_segmentation --with yuan_qcap
```

(Known Issue: Running fail with log lost holohub.qcap_source. Because colonoscopy_segmentation don't trigger build the qcap_source operator. To workaround this, please build endoscopy_tool_tracking fist and run colonoscopy again.
\$./run build endoscopy_tool_tracking --with yuan_qcap.)

```
nvidia@igx:~/workspace/holohub$ ./run build colonoscopy_segmentation --with yuan_qcap
Building with operator yuan_qcap
Building Holohub
building colonoscopy_segmentation application
[command] cmake -S . -B build --no-warn-unused-cli -DPython3_EXECUTABLE=/usr/bin/python3 -DPython3_ROOT_DIR=/usr/lib/python3 -DHOLOHUB_BUILD_OPERATOR=yuan_qcap -DHOLOHUB_DATA_DIR=$PATH=/workspace/holohub/data -DCMAKE_BUILD_TYPE=release -DAPP_colonoscopy_segmentation=1
Not searching for unused variables given on the command line.
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found version "2.0.0")
-- pybind11 v2.10.1
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found version "2.0.0")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/ holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Configuring done
-- Generating done
-- Build files have been written to: /workspace/holohub/build
[command] cmake --build build -j
Consolidate compiler generated dependencies of target gxf_qcap_source_lib
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference
Consolidate compiler generated dependencies of target visualizer_icardio
Consolidate compiler generated dependencies of target tool_tracking_postprocessor
Consolidate compiler generated dependencies of target ultrasound_segmentation_cpp_test
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference_lib
[ 2%] Built target colonoscopy_segmentation_data
[ 5%] Built target colonoscopy_segmentation_test
[ 12%] Built target ultrasound_segmentation_data
[ 12%] Built target endoscopy_data
[ 12%] Built target multiai_ultrasound_data
[ 12%] Built target endoscopy_tool_tracking_deps
[ 12%] Built target multiai_ultrasound_yaml
[ 12%] Built target mpwu_multiai_ultrasound_yaml
[ 12%] Built target ultrasound_segmentation_yaml
[ 20%] Built target gxf_qcap_source_lib
[ 20%] Built target ultrasound_segmentation_python_test
[ 25%] Built target visualizer_icardio
[ 30%] Built target lstm_tensor_rt_inference
[ 35%] Built target gxf_lstm_tensor_rt_inference_lib
Consolidate compiler generated dependencies of target gxf_qcap_source
[ 50%] Built target ultrasound_segmentation_cpp_test
[ 50%] Built target tool_tracking_postprocessor
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference_python
Consolidate compiler generated dependencies of target visualizer_icardio_python
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference
Consolidate compiler generated dependencies of target tool_tracking_postprocessor_python
[ 55%] Built target gxf_qcap_source
Consolidate compiler generated dependencies of target qcap_source
```

6.1.2 Run colonoscopy segmentation (python)

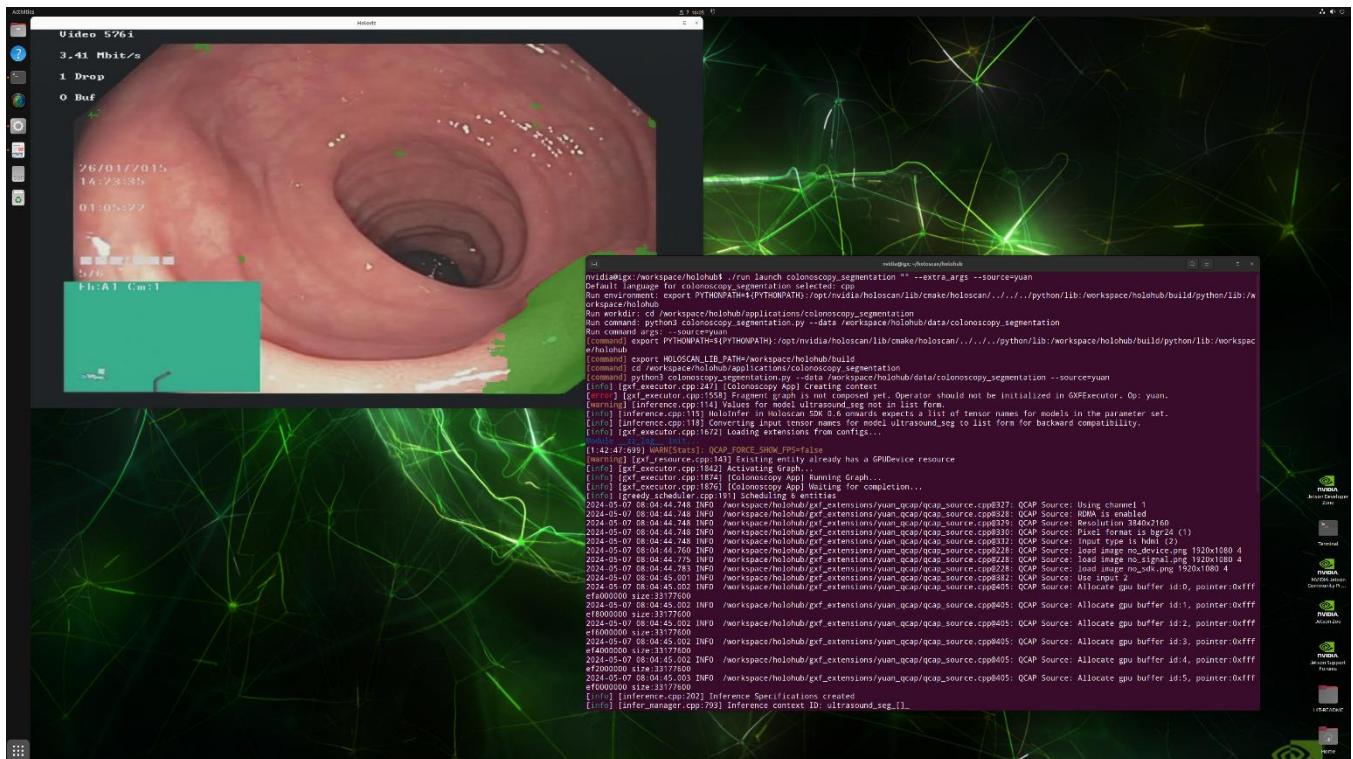
Prepare a video player to play the file in

holohub/data/colonoscopy_segmentation/colo_exam_720x576.mp4 and the video output to the capture card.

This will take long time (over 5-10 mins) in first-time running.

\$./dev_container launch

\$./run launch colonoscopy_segmentation "" --extra_args --source=yuan



6.2 Endoscopy tool tracking

6.2.1 Build endoscopy tool tracking

```
$ cd holohub
$ ./dev_container launch
$ ./run build endoscopy_tool_tracking --with yuan_qcap
```

```
nvidia@igx:~/workspace/holohub$ ./run build endoscopy_tool_tracking --with yuan_qcap
Building with operator yuan_qcap
Building Holohub
building endoscopy_tool_tracking application
[command] cmake -S . -B build --no-warn-unused-cli -DPython3_EXECUTABLE=/usr/bin/python3 -DPython3_ROOT_DIR=/usr/lib/python3 -DHOLOHUB_BUILD_OPERATOR=yuan_qcap -DHOLOHUB_DATA_DIR=PATH=/workspace/holohub/data -DCMAKE_BUILD_TYPE=release -DAPP_endoscopy_tool_tracking=1
Not searching for unused variables given on the command line.
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found version "2.0.0")
-- pybind11 v2.10.1
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found version "2.0.0")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/oloscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Configuring done
-- Generating done
-- Build files have been written to: /workspace/holohub/build
[command] cmake --build build -j
Consolidate compiler generated dependencies of target gxf_qcap_source_lib
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference
Consolidate compiler generated dependencies of target tool_tracking_postprocessor
Consolidate compiler generated dependencies of target visualizer_icardio
[ 2%] Built target colonoscopy_segmentation_data
[ 5%] Built target colonoscopy_segmentation_test
Consolidate compiler generated dependencies of target ultrasound_segmentation_cpp_test
[ 7%] Built target endoscopy_data
[10%] Built target multiai_ultrasound_data
[10%] Built target mpwu_multiai_ultrasound_yaml
[10%] Built target endoscopy_tool_tracking_deps
[10%] Built target multiai_ultrasound_yaml
[12%] Built target ultrasound_segmentation_data
[22%] Built target lstm_tensor_rt_inference
[22%] Built target gxf_qcap_source_lib
[22%] Built target ultrasound_segmentation_yaml
[25%] Built target ultrasound_segmentation_python_test
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference_lib
[30%] Built target visualizer_icardio
[37%] Built target tool_tracking_postprocessor
Consolidate compiler generated dependencies of target gxf_qcap_source
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference_python
[45%] Built target ultrasound_segmentation_cpp_test
Consolidate compiler generated dependencies of target visualizer_icardio_python
Consolidate compiler generated dependencies of target tool_tracking_postprocessor_python
[50%] Built target gxf_qcap_source
[55%] Built target gxf_lstm_tensor_rt_inference_lib
Consolidate compiler generated dependencies of target qcap_source
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference
[60%] Built target lstm_tensor_rt_inference_python
[65%] Built target tool_tracking_postprocessor_python
[70%] Built target visualizer_icardio_python
[80%] Built target gxf_lstm_tensor_rt_inference
[80%] Built target qcap_source
Consolidate compiler generated dependencies of target qcap_source_python
Consolidate compiler generated dependencies of target endoscopy_tool_tracking
Consolidate compiler generated dependencies of target multiai_ultrasound
Consolidate compiler generated dependencies of target ultrasound_segmentation
[85%] Built target qcap_source_python
[90%] Built target endoscopy_tool_tracking
[100%] Built target ultrasound_segmentation
[100%] Built target multiai_ultrasound
Holohub build done.
nvidia@igx:~/workspace/holohub$ ]
```

6.2.2 Run endoscopy tool tracking (cpp)

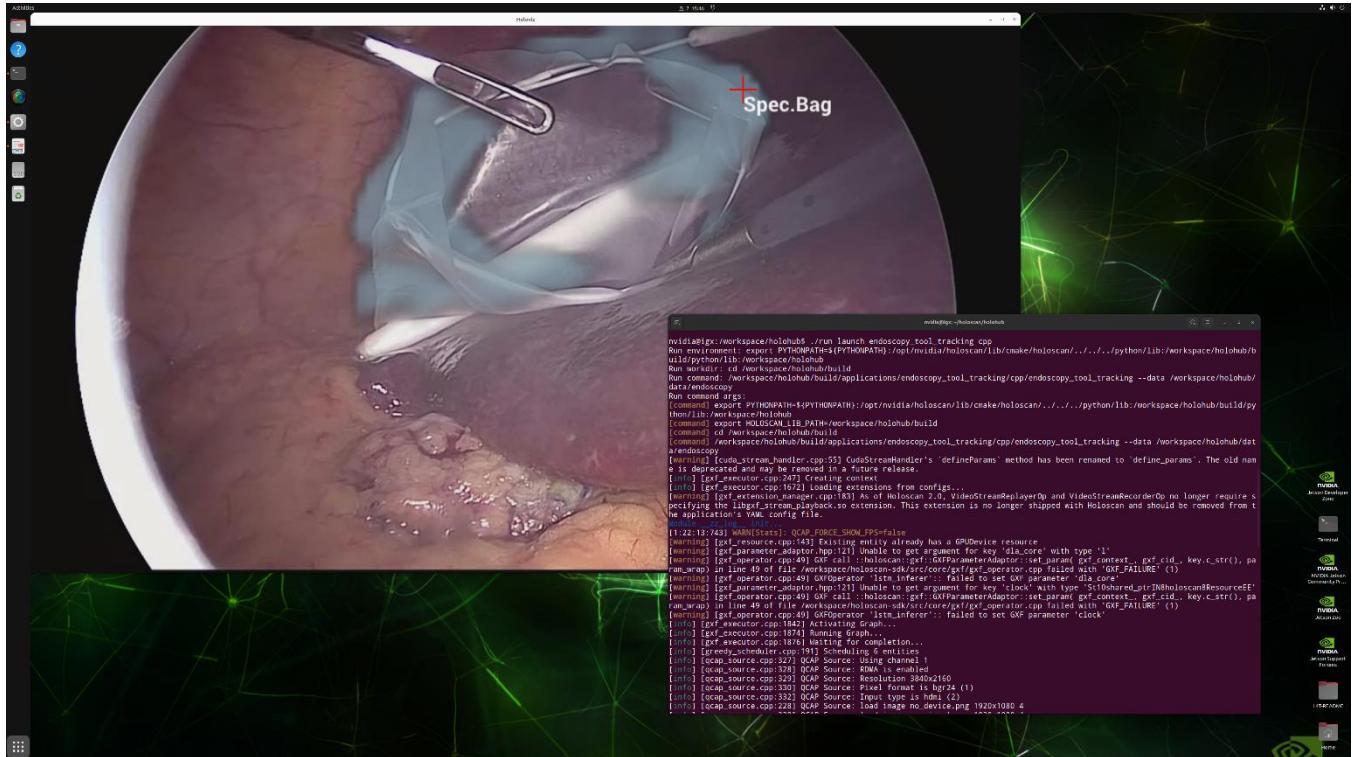
Prepare a video player to play the file in holohub/data/endoscopy/surgical_video.264 and the video output to the capture card.

#This will take long time (over 5-10 mins) in first-time running.

\$ cd holohub

\$./dev_container launch

\$./run launch endoscopy_tool_tracking cpp



6.2.3 Run endoscopy tool tracking (python)

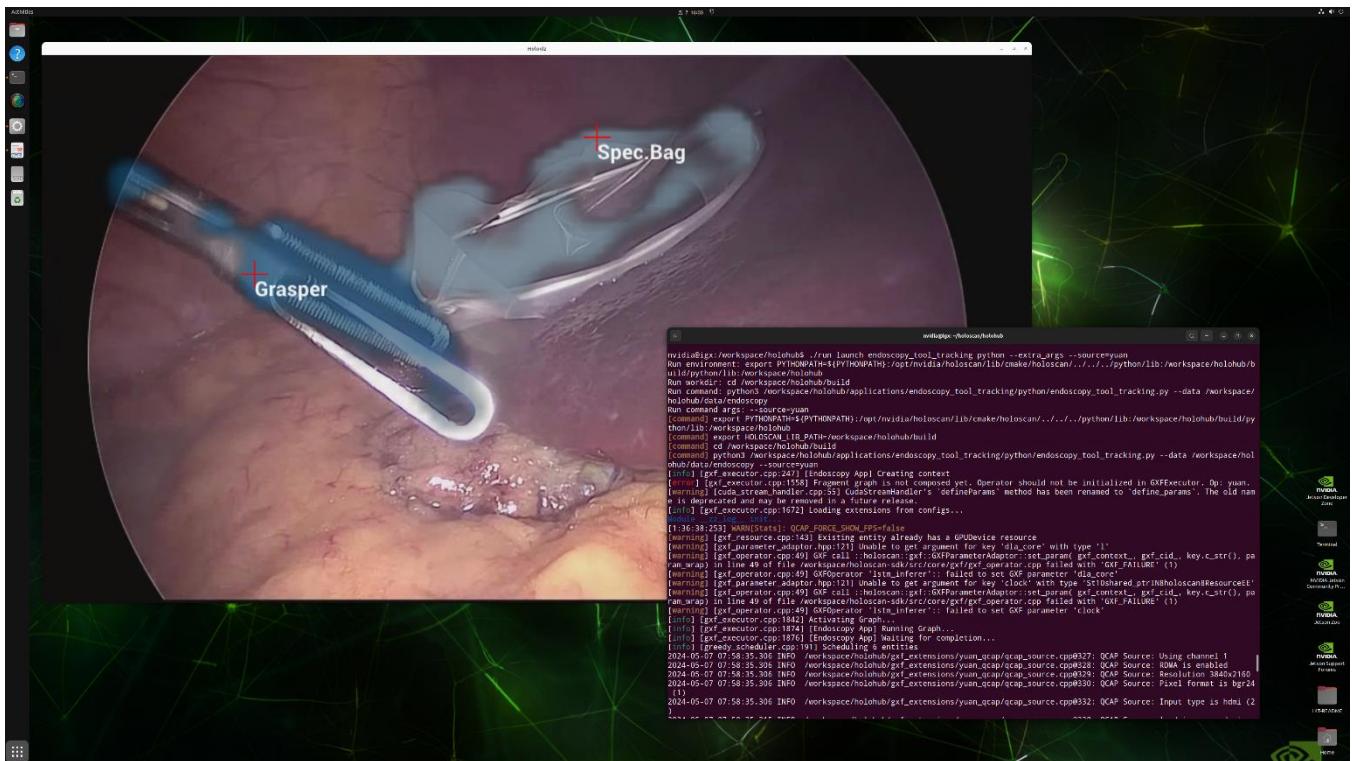
```
# Prepare a video player to play the file in holohub/data/endoscopy/surgical_video.264 and the  
video output to the capture card.
```

This will take long time (over 5-10 mins) in first-time running.

```
$ cd holohub
```

```
$ ./dev_container launch
```

```
$ ./run launch endoscopy_tool_tracking python --extra_args --source=yuan
```

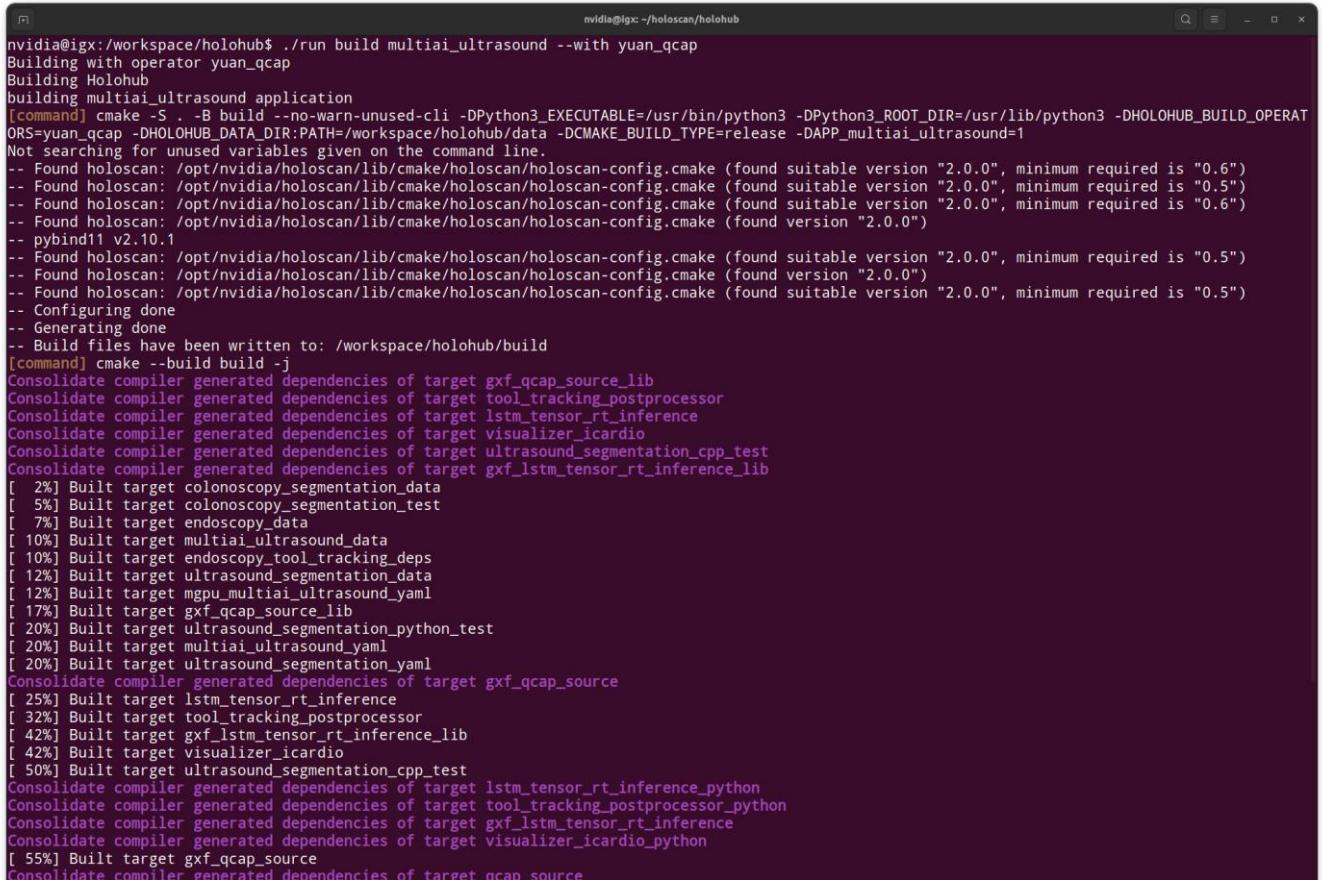


This information contained herein is the exclusive property of YUAN and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of YUAN.

6.3 Multiai ultrasound

6.3.1 Build multiai ultrasound

```
$ cd holohub
$ ./dev_container launch
$ ./run build multai_ultrasound --with yuan_qcap
```



The screenshot shows a terminal window titled 'nvidia@igx:~/workspace/holohub\$' displaying the output of a cmake build command. The command is:

```
./run build multaii_ultrasound --with yuan_qcap
```

The output shows the build process for the 'multaii_ultrasound' application, which includes linking against various libraries and tools. The progress bar indicates the completion of the build process.

```
Building with operator yuan_qcap
Building Holohub
building multaii_ultrasound application
[command] cmake -S . -B build --no-warn-unused-cli -DPython3_EXECUTABLE=/usr/bin/python3 -DPython3_ROOT_DIR=/usr/lib/python3 -DHOLOHUB_BUILD_OPERATOR=yuan_qcap -DHOLOHUB_DATA_DIR:PATH=/workspace/holohub/data -DCMAKE_BUILD_TYPE=release -DAPP_multaii_ultrasound=1
Not searching for unused variables given on the command line.
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found version "2.0.0")
-- pybind11 v2.10.1
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found version "2.0.0")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Configuring done
-- Generating done
-- Build files have been written to: /workspace/holohub/build
[command] cmake --build build -j
Consolidate compiler generated dependencies of target gxf_qcap_source_lib
Consolidate compiler generated dependencies of target tool_tracking_postprocessor
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference
Consolidate compiler generated dependencies of target visualizer_icardio
Consolidate compiler generated dependencies of target ultrasound_segmentation_cpp_test
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference_lib
[ 2%] Built target colonoscopy_segmentation_data
[ 5%] Built target colonoscopy_segmentation_test
[ 7%] Built target endoscopy_data
[ 10%] Built target multaii_ultrasound_data
[ 10%] Built target endoscopy_tool_tracking_deps
[ 12%] Built target ultrasound_segmentation_data
[ 12%] Built target mgpu_multaii_ultrasound_yaml
[ 17%] Built target gxf_qcap_source_lib
[ 20%] Built target ultrasound_segmentation_python_test
[ 20%] Built target multaii_ultrasound_yaml
[ 20%] Built target ultrasound_segmentation_yaml
Consolidate compiler generated dependencies of target gxf_qcap_source
[ 25%] Built target lstm_tensor_rt_inference
[ 32%] Built target tool_tracking_postprocessor
[ 42%] Built target gxf_lstm_tensor_rt_inference_lib
[ 42%] Built target visualizer_icardio
[ 50%] Built target ultrasound_segmentation_cpp_test
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference_python
Consolidate compiler generated dependencies of target tool_tracking_postprocessor_python
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference
Consolidate compiler generated dependencies of target visualizer_icardio_python
[ 55%] Built target gxf_qcap_source
Consolidate compiler generated dependencies of target qcap_source
```

6.3.2 Run multiai ultrasound (cpp)

Prepare a video player to play the file in holohub/data/multiai_ultrasound/icardio_input1.avi and the video output to the capture card.

This will take long time (over 5-10 mins) in first-time running.

\$ cd holohub

\$./dev_container launch

\$./run launch multiai_ultrasound cpp

```

nvidia@lgx:~/workspace/holohub$ ./run launch multiai_ultrasound cpp
Run environment: export PYTHONPATH=${PYTHONPATH}:/opt/nvidia/holoscan/lib/cmake/holoscan/../../../../../python/lib:/workspace/holohub/build/python/lib:/workspace/holohub
Run workdir: cd /workspace/holohub/build/applications/multiai_ultrasound/cpp
Run command: /workspace/holohub/build/applications/multiai_ultrasound/cpp/multiai_ultrasound --data /workspace/holohub/data/multiai_ultrasound
Run command args:
[command] export PYTHONPATH=${PYTHONPATH}:/opt/nvidia/holoscan/lib/cmake/holoscan/../../../../../python/lib:/workspace/holohub/build/python/lib:/workspace/holohub
[command] export HOLOSCAN_LIB_PATH=/workspace/holohub/build
[command] cd /workspace/holohub/build/applications/multiai_ultrasound/cpp
[command] /workspace/holohub/build/applications/multiai_ultrasound/cpp/multiai_ultrasound --data /workspace/holohub/data/multiai_ultrasound
[info] [gxf_executor.cpp:247] Creating context
[info] [gxf_executor.cpp:1672] Loading extensions from config...
Module: >= log_init
[2:27:33:983] [WARN][Stats]: QCAP_FORCE_SHOW_FPS=false
[warning] [inference.cpp:88] Values for model plax_chamber not in a vector form.
[info] [inference.cpp:89] HoloInfer in Holoscan SDK 0.6 onwards expects tensor names for models in a vector form in the parameter set.
[warning] [inference.cpp:93] Converting input tensor names for model plax_chamber to vector form for backward compatibility mode.
[warning] [inference.cpp:88] Single I/O per model supported in backward compatibility mode.
[warning] [inference.cpp:89] Values for model aortic_stenosis not in a vector form.
[info] [inference.cpp:89] HoloInfer in Holoscan SDK 0.6 onwards expects tensor names for models in a vector form in the parameter set.
[warning] [inference.cpp:93] Converting input tensor names for model aortic_stenosis to vector form for backward compatibility mode.
[warning] [inference.cpp:97] Single I/O per model supported in backward compatibility mode.
[warning] [inference.cpp:88] Values for model bmode_perspective not in a vector form.
[info] [inference.cpp:89] HoloInfer in Holoscan SDK 0.6 onwards expects tensor names for models in a vector form in the parameter set.
[warning] [inference.cpp:97] Single I/O per model supported in backward compatibility mode.
[info] [inference_processor.cpp:85] Values for Tensor plax_cham_infer not in a vector form.
[info] [inference_processor.cpp:86] HoloInfer in Holoscan SDK 0.6 onwards expects mapped tensor names in a vector form.
[info] [inference_processor.cpp:89] Converting mappings for tensor plax_cham_infer to vector for backward compatibility.
[info] [gxf_executor.cpp:1842] Activating Graph...
[info] [gxf_executor.cpp:1874] Running Graph...
[info] [gxf_executor.cpp:1876] Waiting for completion...
[info] [greedy_scheduler.cpp:191] Scheduling 10 entities
[info] [qcap_source.cpp:327] QCAP Source: Using channel 1
[info] [qcap_source.cpp:328] QCAP Source: RDMA is enabled
[info] [qcap_source.cpp:329] QCAP Source: Resolution 3840x2160
[info] [qcap_source.cpp:330] QCAP Source: Pixel format is bgr24 (1)
[info] [qcap_source.cpp:332] QCAP Source: Input type is hdmi (2)
[info] [qcap_source.cpp:228] QCAP Source: load image no_device.png 1920x1080 4
[info] [qcap_source.cpp:228] QCAP Source: load image no_signal.png 1920x1080 4
[info] [qcap_source.cpp:228] QCAP Source: load image no_sdk.png 1920x1080 4
[info] [qcap_source.cpp:382] QCAP Source: Use input 2
[info] [qcap_source.cpp:405] QCAP Source: Allocate gpu buffer id:0, pointer:0xffffe1e00000 size:33177600
[info] [qcap_source.cpp:405] QCAP Source: Allocate gpu buffer id:1, pointer:0xffffeca00000 size:33177600
[info] [qcap_source.cpp:405] QCAP Source: Allocate gpu buffer id:2, pointer:0xffffec8000000 size:33177600
[info] [qcap_source.cpp:405] QCAP Source: Allocate gpu buffer id:3, pointer:0xffffec6000000 size:33177600
[info] [qcap_source.cpp:405] QCAP Source: Allocate gpu buffer id:4, pointer:0xffffec4000000 size:33177600
[info] [qcap_source.cpp:405] QCAP Source: Allocate gpu buffer id:5, pointer:0xffffec2000000 size:33177600
[info] [inference.cpp:202] Inference Specifications created
[info] [infer_manager.cpp:793] Inference context ID: aortic_stenosis_[]_bmode_perspective_[]_plax_chamber_[]
[info] [core.cpp:46] TRT Inference: converting ONNX model at /workspace/holohub/data/multiai_ultrasound/aortic_stenosis.onnx
[info] [utils.cpp:76] Cached engine found: /workspace/holohub/data/multiai_ultrasound/aortic_stenosis.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32
[info] [core.cpp:79] Loading Engine: /workspace/holohub/data/multiai_ultrasound/aortic_stenosis.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32
[info] [core.cpp:122] Engine loaded: /workspace/holohub/data/multiai_ultrasound/aortic_stenosis.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32
[info] [infer_manager.cpp:359] HoloInfer buffer created for aortic_infer
[info] [core.cpp:46] TRT Inference: converting ONNX model at /workspace/holohub/data/multiai_ultrasound/bmode_perspective.onnx
[info] [utils.cpp:76] Cached engine found: /workspace/holohub/data/multiai_ultrasound/bmode_perspective.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32
[info] [core.cpp:79] Loading Engine: /workspace/holohub/data/multiai_ultrasound/bmode_perspective.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32
[info] [core.cpp:122] Engine loaded: /workspace/holohub/data/multiai_ultrasound/bmode_perspective.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32
[info] [infer_manager.cpp:359] HoloInfer buffer created for bmode_infer
[info] [core.cpp:46] TRT Inference: converting ONNX model at /workspace/holohub/data/multiai_ultrasound/plax_chamber.onnx
[info] [utils.cpp:76] Cached engine found: /workspace/holohub/data/multiai_ultrasound/plax_chamber.NVIDIARTXA6000.8.6.84.trt.8.6.1.6.engine.fp32

```

6.3.3 Run multiai ultrasound (python)

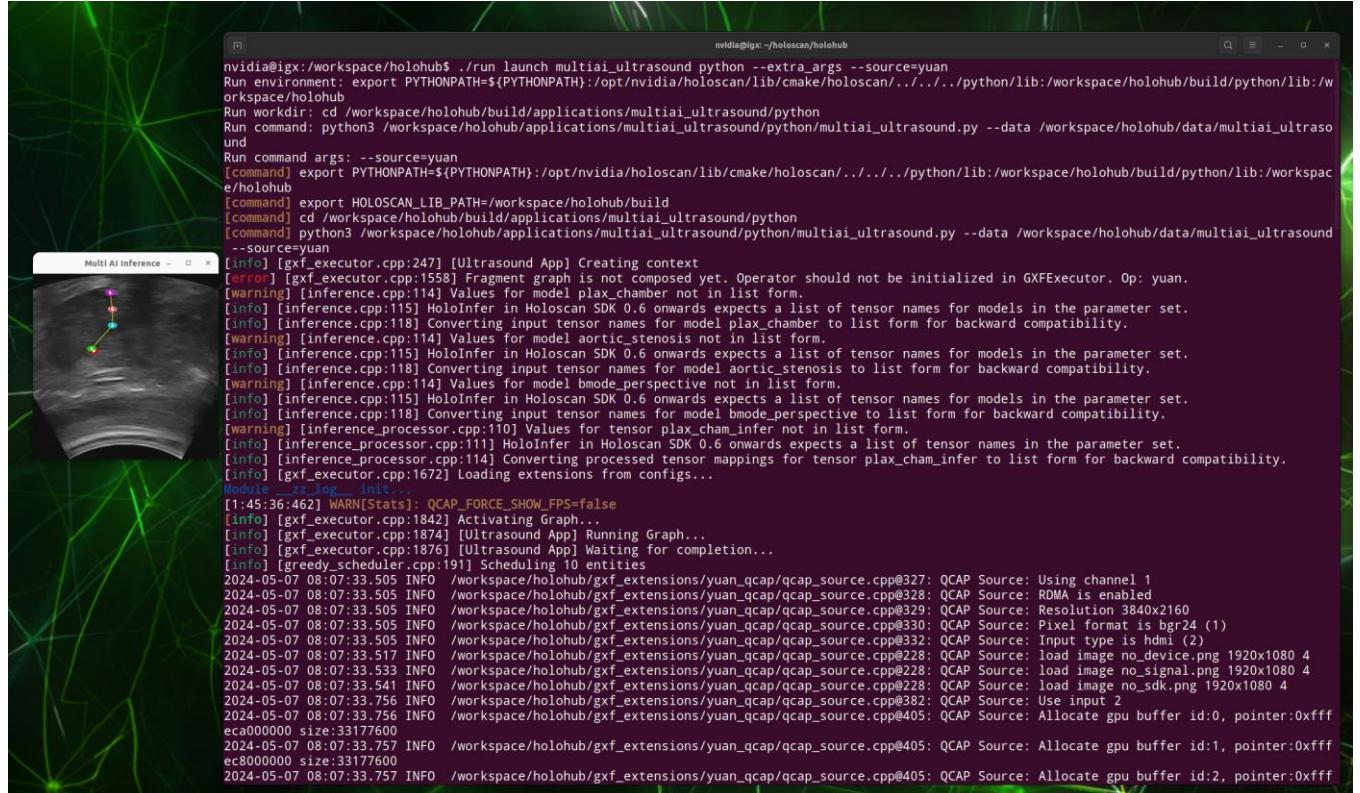
Prepare a video player to play the file in holohub/data/multiai_ultrasound/icardio_input1.avi and the video output to the capture card.

This will take long time (over 5-10 mins) in first-time running.

\$ cd holohub

\$./dev_container launch

\$./run launch multiai_ultrasound python --extra_args --source=yuan



6.4 Ultrasound segmentation

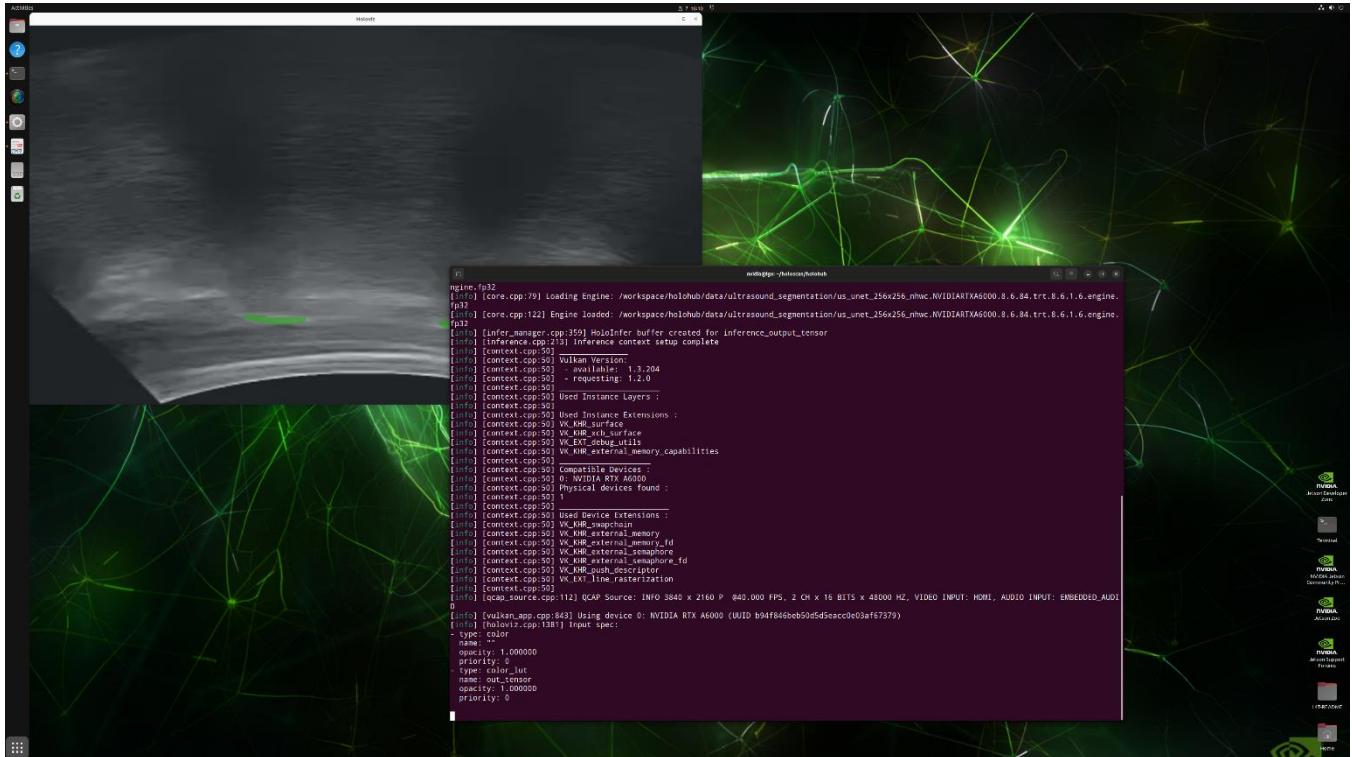
6.4.1 Build Ultrasound segmentation

```
$ cd holohub
$ ./dev_container launch
$ ./run build ultrasound_segmentation --with yuan_qcap
```

```
nvidia@igx:~/workspace/holohub$ ./run build ultrasound_segmentation --with yuan_qcap
Building with operator yuan_qcap
Building Holohub
building ultrasound_segmentation application
[command] cmake -S . -B build --no-warn-unused-cli -DPython3_EXECUTABLE=/usr/bin/python3 -DPython3_ROOT_DIR=/usr/lib/python3 -DHOLOHUB_BUILD_OPERATOR=yuan_qcap -DHOLOHUB_DATA_DIR:PATH=/workspace/holohub/data -DCMAKE_BUILD_TYPE=release -DAPP_ultrasound_segmentation=1
Not searching for unused variables given on the command line.
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.6")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found version "2.0.0")
-- pybind11 v2.10.1
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found version "2.0.0")
-- Found holoscan: /opt/nvidia/holoscan/lib/cmake/holoscan/holoscan-config.cmake (found suitable version "2.0.0", minimum required is "0.5")
-- Configuring done
-- Generating done
-- Build files have been written to: /workspace/holohub/build
[command] cmake --build build -j
Consolidate compiler generated dependencies of target gxf_qcap_source_lib
Consolidate compiler generated dependencies of target visualizer_icardio
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference
Consolidate compiler generated dependencies of target tool_tracking_postprocessor
[ 2%] Built target colonoscopy_segmentation_data
Consolidate compiler generated dependencies of target ultrasound_segmentation_cpp_test
[ 5%] Built target endoscopy_data
[ 7%] Built target colonoscopy_segmentation_test
[ 10%] Built target multiai_ultrasound_data
[ 10%] Built target endoscopy_tool_tracking_deps
[ 12%] Built target ultrasound_segmentation_data
[ 15%] Built target multiai_ultrasound_yaml
[ 15%] Built target ultrasound_segmentation_python_test
[ 15%] Built target mpwu_multiai_ultrasound_yaml
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference_lib
[ 15%] Built target ultrasound_segmentation_yaml
[ 25%] Built target gxf_qcap_source_lib
[ 25%] Built target lstm_tensor_rt_inference
[ 30%] Built target visualizer_icardio
[ 37%] Built target tool_tracking_postprocessor
[ 45%] Built target ultrasound_segmentation_cpp_test
Consolidate compiler generated dependencies of target gxf_qcap_source
Consolidate compiler generated dependencies of target lstm_tensor_rt_inference_python
Consolidate compiler generated dependencies of target visualizer_icardio_python
[ 50%] Built target gxf_lstm_tensor_rt_inference_lib
Consolidate compiler generated dependencies of target tool_tracking_postprocessor_python
Consolidate compiler generated dependencies of target gxf_lstm_tensor_rt_inference
[ 55%] Built target gxf_qcap_source
Consolidate compiler generated dependencies of target qcap_source
```

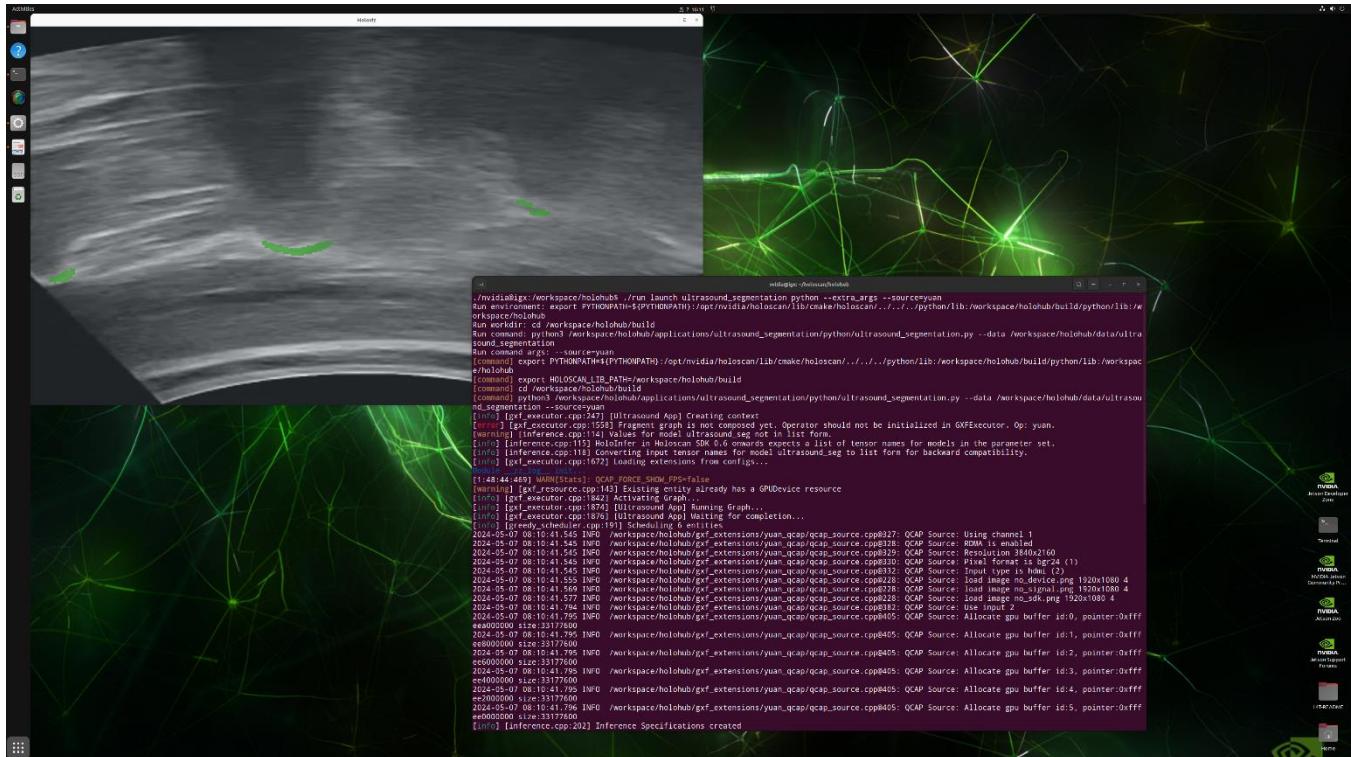
6.4.2 Run Ultrasound segmentation (cpp)

```
# Prepare a video player to play the file in
holohub/data/ultrasound_segmentation/ultrasound_256x256.avi and the video output to the capture card.
# This will take long time (over 5-10 mins) in first-time running.
$ cd holohub
$ ./dev_container launch
$ ./run launch ultrasound_segmentation cpp
```



6.4.3 Run ultrasound segmentation (python)

```
# Prepare a video player to play the file in  
holohub/data/ultrasound_segmentation/ultrasound_256x256.avi and the video output to the capture card.  
# This will take long time (over 5-10 mins) in first-time running.  
$ cd holohub  
$ ./dev_container launch  
$ ./run launch multiai_ultrasound python --extra_args --source=yuan
```

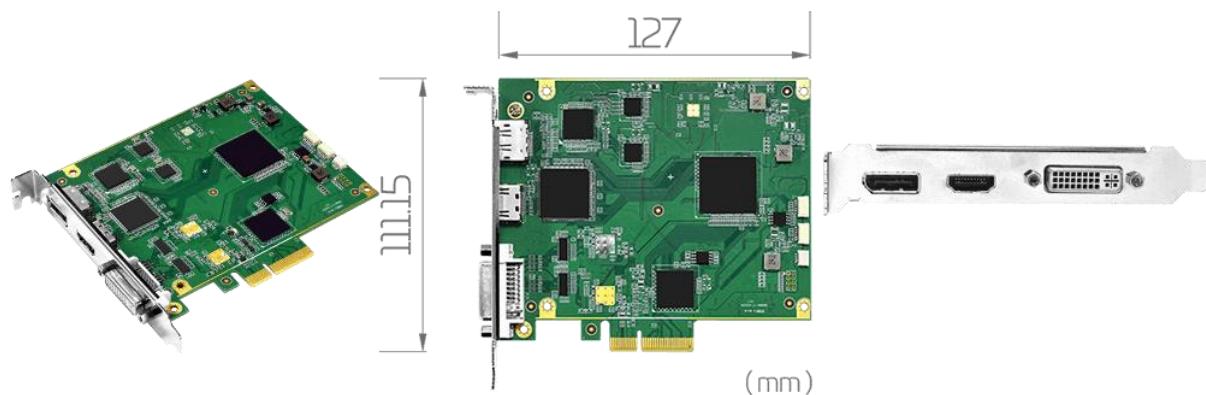


This information contained herein is the exclusive property of YUAN and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of YUAN.

7. Change video resolution or select input type (Video: Ch7_switch_input_type.mp4)
You may need to switch to different input type for some capture card has multiple input type.

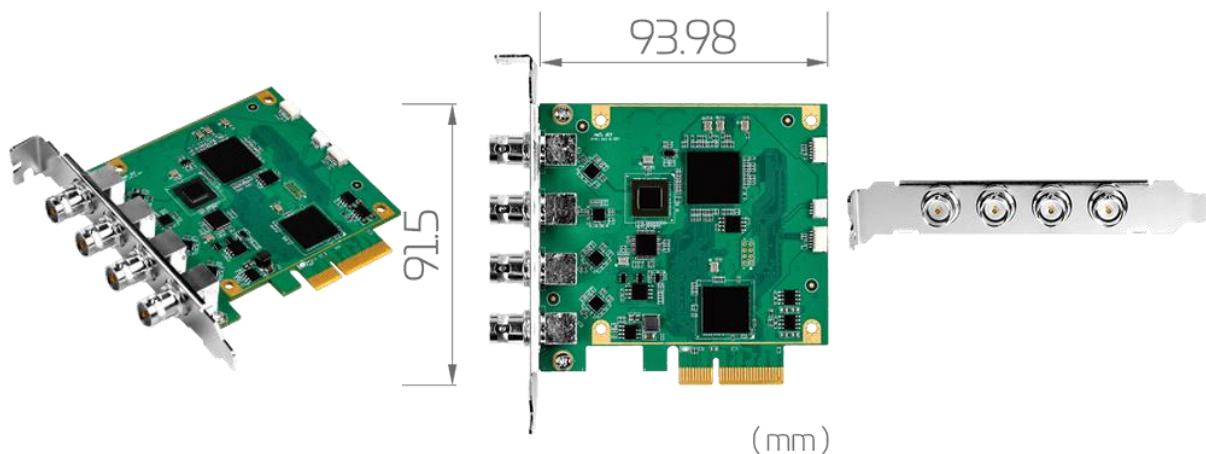
SC710N1 HDV has 3 different input type, HDMI, DVI-D and DisplayPort.

Refer: [SC710N1 HDV](#)



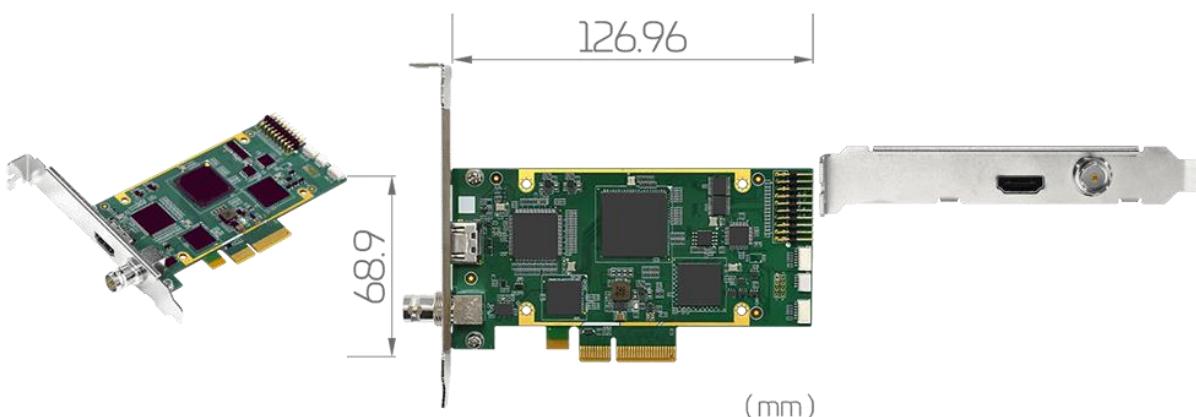
SC710N1 12G-SDI has one input type, SDI.

Refer: [SC710N1 12G-SDI](#)



SC710N1-L 12G-SDI + HDMI2.0 has two input type, SDI and HDMI.

Refer: [SC710N1-L 12G-SDI + HDMI2.0](#)



7.1 Modify config file to change video resolution.

Modify the entry ‘width’ and ‘height’ of qcap section in the config file (e.g. endoscopy_tool_tracking.yaml).

```

</home/yuan/holoscan/holohub/
> .git/
└── applications/
    ├── adv_networking_bench/
    ├── basic_networking_ping/
    ├── colonoscopy_segmentation/
    ├── cunumeric_integration/
    ├── deltacast_endoscopy_tool_tracking/
    ├── deltacast_transmitter/
    ├── endoscopy_out_of_body_detection/
    └── endoscopy_tool_tracking/
        └── cpp/
            ├── CMakeLists.txt
            ├── endoscopy_tool_tracking.yaml
            ├── main.cpp
            ├── metadata.json
            └── README.md
    ├── docs/
    ├── python/
        ├── CMakeLists.txt
        └── README.md
    └── h264_endoscopy_tool_tracking/
        ├── h264_video_decode/
        ├── high_speed_endoscopy/
        ├── monai_endoscopic_tool_seg/
        ├── multiai_ultrasound/
        ├── sdr_fm_demodulation/
        ├── simple_pdw_pipeline/
        ├── simple_radar_pipeline/
        └── speech_to_text_llm/

```

```

22   - libgxf_stream_playback.so
23   - gxf_extensions/lstm_tensor_rt_inference/libgxf_lstm_tensor_rt_i
nference.so
24   - gxf_extensions/qcap_source/libgxf_qcap_source.so
25
26 source: "replayer" # or "aja" or "qcap"
27 record_type: "none" # or "input" if you want to record input video
stream, or "visualizer" if you want
28 | | | | # to record the visualizer output.
29
30 aja:
31   width: 1920
32   height: 1080
33   rdma: true
34   enable_overlay: false
35   overlay_rdma: true
36
37 qcap:
38   width: 3840
39   height: 2160
40   rdma: true
41
42 replayer:
43   basename: "surgical_video"
44   frame_rate: 0 # as specified in timestamps
45   repeat: true # default: false
46   realtime: true # default: true
47   count: 0 # default: 0 (no frame count restriction)
48
49 recorder_format_converter:

```

7.2 Modify config file to change input type.

Add a new entry ‘input_type’ of qcap section in the config file (e.g. endoscopy_tool_tracking.yaml). The valid value are ‘dvi_d’, ‘dp’, ‘sdi’ or ‘hdmi’.

```

/media/m2/home/yuan/holoscan/holohub/
> .git/
└── applications/
    ├── adv_networking_bench/
    ├── basic_networking_ping/
    ├── colonoscopy_segmentation/
    ├── cunumeric_integration/
    ├── deltacast_endoscopy_tool_tracking/
    ├── deltacast_transmitter/
    ├── endoscopy_out_of_body_detection/
    └── endoscopy_tool_tracking/
        └── cpp/
            ├── CMakeLists.txt
            ├── endoscopy_tool_tracking.yaml
            ├── main.cpp
            ├── metadata.json
            └── README.md
    ├── docs/
    ├── python/
        ├── CMakeLists.txt
        └── README.md
    └── h264_endoscopy_tool_tracking/
        ├── h264_video_decode/
        ├── high_speed_endoscopy/
        ├── monai_endoscopic_tool_seg/
        ├── multiai_ultrasound/
        ├── sdr_fm_demodulation/
        ├── simple_pdw_pipeline/
        ├── simple_radar_pipeline/
        ├── speech_to_text_llm/
        ├── ssd_detection_endoscopy_tools/
        ├── ultrasound_segmentation/
        └── usb_hdmi_video_capture/
            ├── CMakeLists.txt
            └── README.md

```

```

22   - libgxf_stream_playback.so
23   - gxf_extensions/lstm_tensor_rt_inference/libgxf_lstm_tensor_rt_i
nference.so
24   - gxf_extensions/qcap_source/libgxf_qcap_source.so
25
26 source: "replayer" # or "aja" or "qcap"
27 record_type: "none" # or "input" if you want to record input vid
28 | | | | # to record the visualizer output.
29
30 aja:
31   width: 1920
32   height: 1080
33   rdma: true
34   enable_overlay: false
35   overlay_rdma: true
36
37 qcap:
38   width: 3840
39   height: 2160
40   rdma: true
41   input_type: "hdmi" # "dvi_d", "dp", "sdi" or "hdmi"
42
43 replayer:
44   basename: "surgical_video"
45   frame_rate: 0 # as specified in timestamps
46   repeat: true # default: false
47   realtime: true # default: true
48   count: 0 # default: 0 (no frame count restriction)
49
50 recorder_format_converter:
51   in_dtype: "rgba8888"
52   out_dtype: "rgb888"
53
54 recorder:
55   directory: "/tmp"
56   basename: "tensor"

```

7.3 Modify config file to change pixel format.

Add a new entry “pixel_format” of qcap section in the config file (e.g. endoscopy_tool_tracking.yaml). The valid value are “y210”, “bgr”, “yuy2” or “nv12”.

```
/home/nvidia/holoscan/holohub/
├ .git/
├ .github/
└ applications/
  ├ adv_networking_bench/
  ├ asr_to_llm/
  ├ basic_networking_ping/
  ├ body_pose_estimation/
  ├ colonoscopy_segmentation/
  ├ cuda_quantum/
  ├ cunumeric_integration/
  ├ cvcuda_basic/
  ├ deltacast_transmitter/
  ├ endoscopy_depth_estimation/
  ├ endoscopy_out_of_body_detection/
  └ endoscopy_tool_tracking/
    └ cpp/
      CMakeLists.txt
      endoscopy_tool_tracking.yaml
      main.cpp
      metadata.json
      README.md
    └ docs/
    └ python/
    └ testing/
      27   - gxf_extensions/yuan_qcap/libgxf_qcap_source.so
      28
      29 source: "yuan" # "replayer" or "aja" or "deltacast" or "yuan"
      30 visualizer: "holoviz" # "holoviz" or "vtk"
      31 record_type: "none" # or "input" if you want to record input video str
        eam, or "visualizer" if you want
      32   | | | | | | | | | # to record the visualizer output.
      33
      34 external_source:
      35   rdma: true
      36   enable_overlay: false
      37
      38 aja:
      39   width: 1920
      40   height: 1080
      41   overlay_rdma: true
      42
      43 yuan:
      44   width: 3840
      45   height: 2160
      46   rdma: false
      47   input_type: sdi # dvi_d, dp, sdi or hdmi
      48   pixel_format: yuy2 # y210, bgr, yuy2, nv12
      49   sdi12g_mode: 1 # 1 for SQD, 2 for SI
      50
```

7.4 Modify config file to change sdi 12g mode.

Add a new entry “” of qcap section in the config file (e.g. endoscopy_tool_tracking.yaml). The valid value are “1” for SQD or “2” for SI.

```
/home/nvidia/holoscan/holohub/
├ .git/
├ .github/
└ applications/
  ├ adv_networking_bench/
  ├ asr_to_llm/
  ├ basic_networking_ping/
  ├ body_pose_estimation/
  ├ colonoscopy_segmentation/
  ├ cuda_quantum/
  ├ cunumeric_integration/
  ├ cvcuda_basic/
  ├ deltacast_transmitter/
  ├ endoscopy_depth_estimation/
  ├ endoscopy_out_of_body_detection/
  └ endoscopy_tool_tracking/
    └ cpp/
      CMakeLists.txt
      endoscopy_tool_tracking.yaml
      main.cpp
      metadata.json
      README.md
    └ docs/
    └ python/
    └ testing/
      27   - gxf_extensions/yuan_qcap/libgxf_qcap_source.so
      28
      29 source: "yuan" # "replayer" or "aja" or "deltacast" or "yuan"
      30 visualizer: "holoviz" # "holoviz" or "vtk"
      31 record_type: "none" # or "input" if you want to record input video str
        eam, or "visualizer" if you want
      32   | | | | | | | | | # to record the visualizer output.
      33
      34 external_source:
      35   rdma: true
      36   enable_overlay: false
      37
      38 aja:
      39   width: 1920
      40   height: 1080
      41   overlay_rdma: true
      42
      43 yuan:
      44   width: 3840
      45   height: 2160
      46   rdma: false
      47   input_type: sdi # dvi_d, dp, sdi or hdmi
      48   pixel_format: yuy2 # y210, bgr, yuy2, nv12
      49   sdi12g_mode: 1 # 1 for SQD, 2 for SI
      50
```

8. TroubleShooting

8.1 Case 1: Show “No QCAP SDK”.

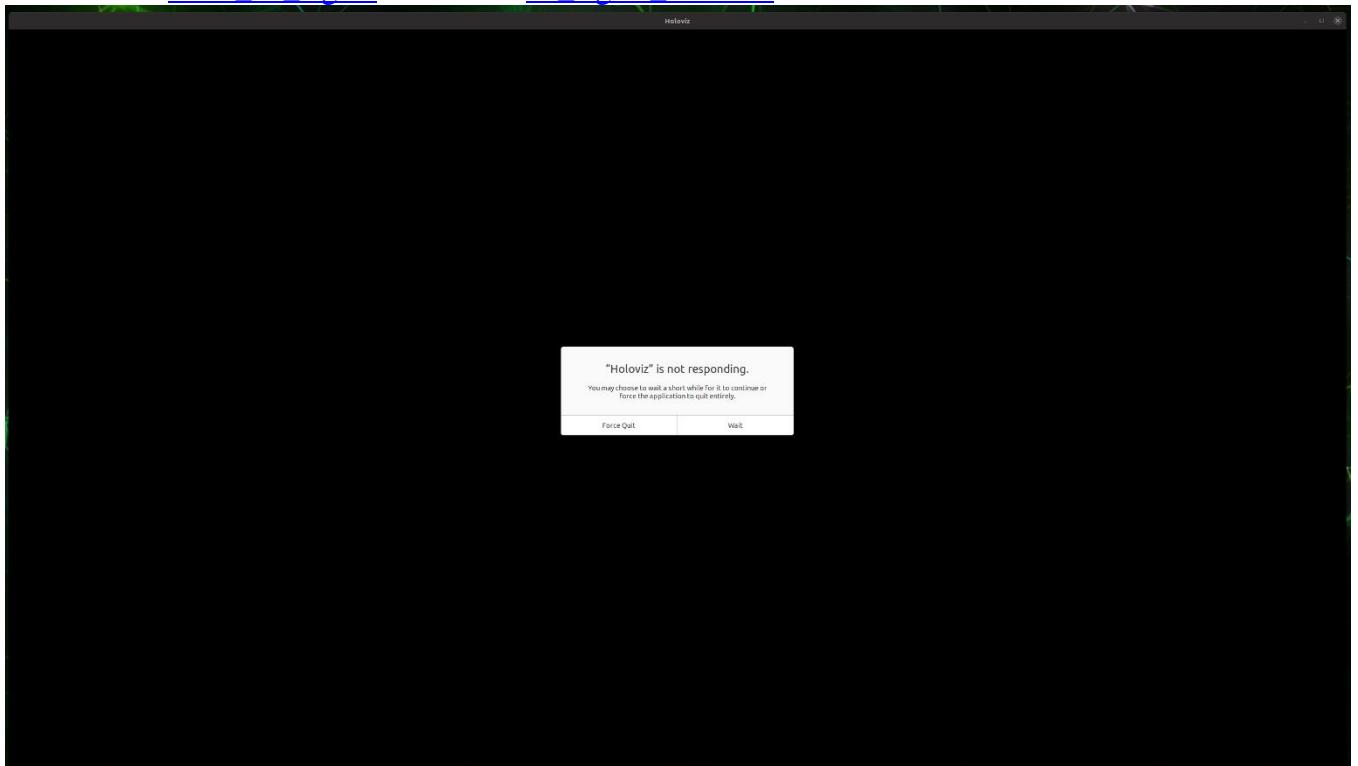
This dialog is shown when holohub build without QCAP SDK, please check [Unpack_QCAP_SDK](#).



No QCAP SDK

8.2 Case 2: A dialog show “Holoviz” is not responding.

This dialog is show when no video, please do advanced chacking as the section of [change resolution or select input type](#), case 3 [show no capture card](#), case 4 [find video dev path fail](#), case 5 [show no signal](#) and case 6 [no signal detected](#).



8.3 Case 3: Show “No Device”

Application is running but no video show up and log has string ‘find_video_dev_path failed’. Please follow the sections of [Install captured card](#) and [Installation of capture driver](#) to check. If application still failed to run, please contact YUAN with below information to get help.

- The model of capture card: SC0710 N1 HDV or SC0710 N1 12G-SDI, etc.
- Input type: HDMI or DVI, etc.
- Input resolution: 3840x2160 60 fps, 1920x1080 60 fps, etc.
- Color Format: RGB24, YUY2, NV12, etc.
- Dmesg log, by command ‘dmesg | tee dmesg.log’
- Application log by command ‘[application_name] | tee app.log’.



No Device

8.4 Case 4: No video and ‘find_video_dev_path fail’

Application is running but no video show up and log has string ‘find_video_dev_path failed’. Please follow the sections of [Install captured card](#) and [Installation of capture driver](#) to check. If application still failed to run, please contact YUAN with below information to get help.

- The model of capture card: SC0710 N1 HDV or SC0710 N1 12G-SDI, etc.
- Input type: HDMI or DVI, etc.
- Input resolution: 3840x2160 60 fps, 1920x1080 60 fps, etc.
- Color Format: RGB24, YUY2, NV12, etc.
- Dmesg log, by command ‘dmesg | tee dmesg.log’
- Application log by command ‘[application_name] | tee app.log’.

```
yuan@ubuntu: ~/clara/qcap_gpudirect_holoviz/build
[0:02:23:881] DEBUG[ZzGPUDirect]: QCAP_GPU_TYPE=1
Module __v4l2_general2__ init...
[0:02:23:881] DEBUG[V4L2_GENERAL2]: QCAP_MAX_VIDEO_FRAME_COUNT = 4
[0:02:23:882] DEBUG[V4L2_GENERAL2]: QCAP_V4L2_VIDEO_BUFFER_COUNT = 10
[0:02:23:882] DEBUG[V4L2_GENERAL2]: QCAP_VIDEO_FRAME_ALIGN=16
[0:02:23:882] DEBUG[V4L2_GENERAL2]: QCAP_VIDEO_FRAME_VALIGN=1
[0:02:23:882] DEBUG[V4L2_GENERAL2]: QCAP_V4L2_VIDEO_DELAY_COUNT=0
[0:02:23:882] DEBUG[V4L2_GENERAL2]: QCAP_V4L2_AUDIO_DELAY_COUNT=0
[0:02:23:882] DEBUG[V4L2_GENERAL2]: QCAP_V4L2_BUFFER_CONTROL=0
Module __webcam_device__ init...
Module __qcap_linux__ init...
[0:02:23:887] DEBUG[AVCommunicationEmulator]: dl open ok~~~~~
Device Number: 0
Device name: Quadro RTX 6000
Integrated: 0
Memory Clock Rate (KHz): 7001000
Memory Bus Width (bits): 384
Peak Memory Bandwidth (GB/s): 672.096000

[0:02:24:124] INFO[QCAPDevice]: QCAP_CREATE: "SC0710 PCI", 0, (nil), 0xfffffdb830168,
1, 0
[0:02:24:125] DEBUG[V4L2_GENERAL2]: Create(630): "SC0710 PCI", 0, (nil), 1, 0
[0:02:24:125] ERROR[V4L2_GENERAL2]: Create(825): [yuancap::find_video_dev_path failed]
[0:02:24:125] ERROR[QCAPDevice]: QCAP_CREATE(2027): pDev->Create() failed, qres=4
```

8.5 Case 5: Show “no signal”

Application is running but no video show up and log has ‘no signal detected’.

Please check video source and make sure cable has connect to the captured card. And check the section of [change resolution or select input type](#). If application still failed to capture video, please contact YUAN with below information to get help.

- The model of capture card: SC0710 N1 HDV or SC0710 N1 12G-SDI, etc.
- Input type: HDMI or DVI, etc.
- Input resolution: 3840x2160 60 fps, 1920x1080 60 fps, etc.
- Color Format: RGB24, YUY2, NV12, etc.
- Dmesg log, by command ‘dmesg | tee dmesg.log’
- Application log by command ‘[application_name] | tee app.log’



No Signal

8.6 Case 6: No video and ‘no signal detected’

Application is running but no video show up and log has ‘no signal detected’.

Please check video source and make sure cable has connect to the captured card. And check the section of [select_input_type](#). If application still failed to capture video, please contact YUAN with below information to get help.

- The model of capture card: SC0710 N1 HDV or SC0710 N1 12G-SDI, etc.
- Input type: HDMI or DVI, etc.
- Input resolution: 3840x2160 60 fps, 1920x1080 60 fps, etc.
- Color Format: RGB24, YUY2, NV12, etc.
- Dmesg log, by command ‘dmesg | tee dmesg.log’
- Application log by command ‘[application_name] | tee app.log’.

```
yuan@ubuntu: ~/clara/qcap_gpudirect_holoviz/build
```

```
Used Instance Extensions :  
VK_KHR_surface  
VK_KHR_xcb_surface  
VK_EXT_debug_utils  
VK_KHR_external_memory_capabilities  
  
Compatible Devices :  
0: Quadro RTX 6000  
Physical devices found : 1  
[0:01:40:933] DEBUG[V4L2_GENERAL2]: VI[0]: NO SIGNAL  
[0:01:40:933] DEBUG[ZzV4L2]: VIDIOC_S_INPUT: 0x80000000  
  
Used Device Extensions :  
VK_KHR_swapchain  
VK_KHR_external_memory  
VK_KHR_external_memory_fd  
  
ERROR: Loader Message  
--> libVkLayer_MESA_device_select.so: cannot open shared object file: No such f  
ile or directory  
[0:01:41:256] DEBUG[ZzV4L2]: ~VIDIOC_S_INPUT: 0x80000000  
no signal Detected
```

9. Version:

V1.00 : Initial version

V1.10 : Update for Holoscan SDK from v0.4.0 to v0.5.0

V1.20 :

1. Refine for Clara Holoscan IGX.
2. Add a new capture card, SC710N1-L 12G-SDI + HDMI 2.0
3. Add a section for change the video resolution.

V1.30:

1. Align to Holohub v0.6

V1.40:

1. Align to Holohub v0.6 20231101

V1.41:

1. Fix problem of can't use nsys.

V1.50:

1. Update to new OS, IGX-SW 1.0.

V2.0.0

1. Update to new OS and Holoscan v2.0.0.

2. Align documentation's version to Holoscan's version.

V2.0.0_1 - 20240624

1. Update for Y210 format.

V2.0.0_2 - 20240705

1. Add Y210 SQD format when enabling GPUDirect.