Environment build

This section introduces how to build a camera running environment on your own controller. Take the workspace orbbec_ws as an example, the workspace directory is under the ~ directory, and modify it according to the actual situation.

1、Install related dependencies

```
sudo apt install libgflags-dev  ros-$ROS_DISTRO-image-geometry ros-$ROS_DISTRO-camera-info-manager ros-$ROS_DISTRO-image-transport ros-$ROS_DISTRO-image-publisher libgoogle-glog-dev libusb-1.0-0-dev libeigen3-dev
```

```
git clone https://github.com/libuvc/libuvc.git
cd libuvc
mkdir build && cd build
cmake .. && make -j4
sudo make install
sudo ldconfig
```

2、Compile feature pack

Unzip the file orbbec_ws_src.zip to get a src folder, then copy orbbec-ros-sdk to the orbbec_ws/src directory (the previous operation of initializing the workspace will not be repeated here), then open the terminal and enter the following command to compile

```
cd ~/orbbec_ws
catkin_make
echo "source ~/orbbec_ws/devel/setup.bash" >> ~/.bashrc
```

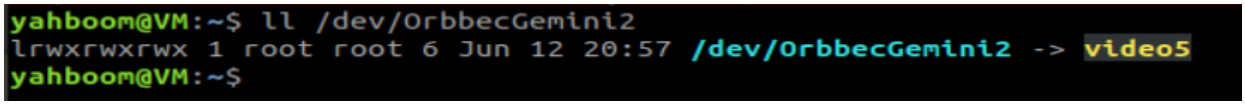
3、Create a serial port rule file

Enter the following command to create，

```
cd ~/orbbec_ws/src/orbbec-ros-sdk/script
sudo chmod 777 *
sudo sh install.sh
```

Then re-plug the camera, enter the following command to check whether the creation is successful,

```
ll /dev/OrbbecGemini2
```

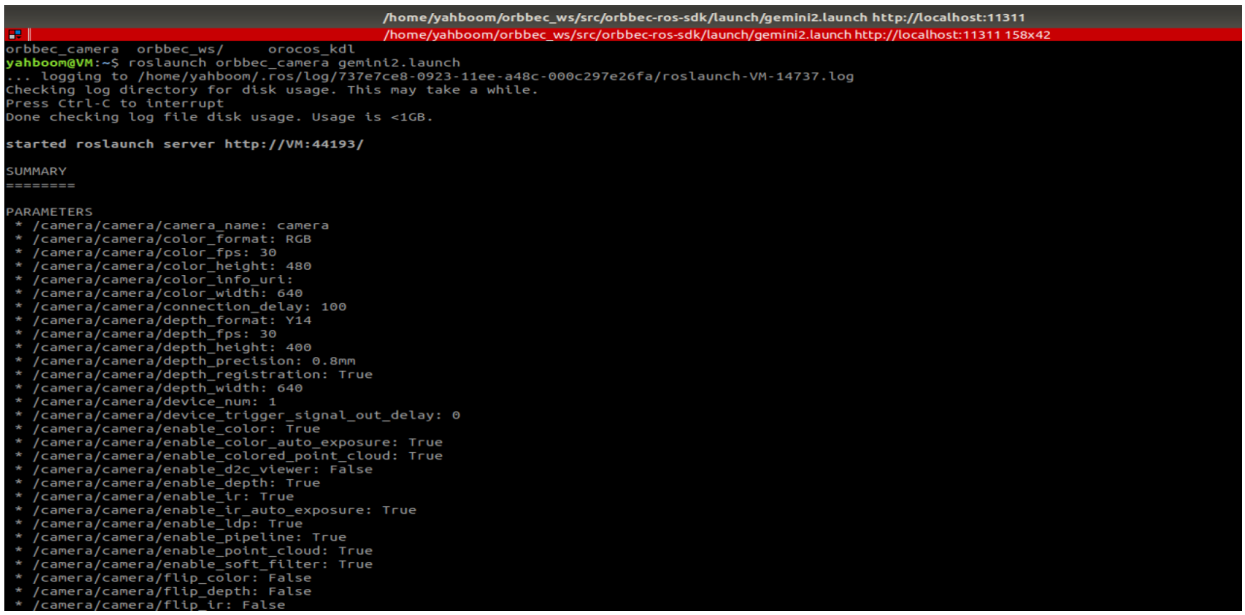


If the above screen appears, it means that the creation is successful. As long as there is ll /dev/OrbbecGemin2, it means that the binding is successful.

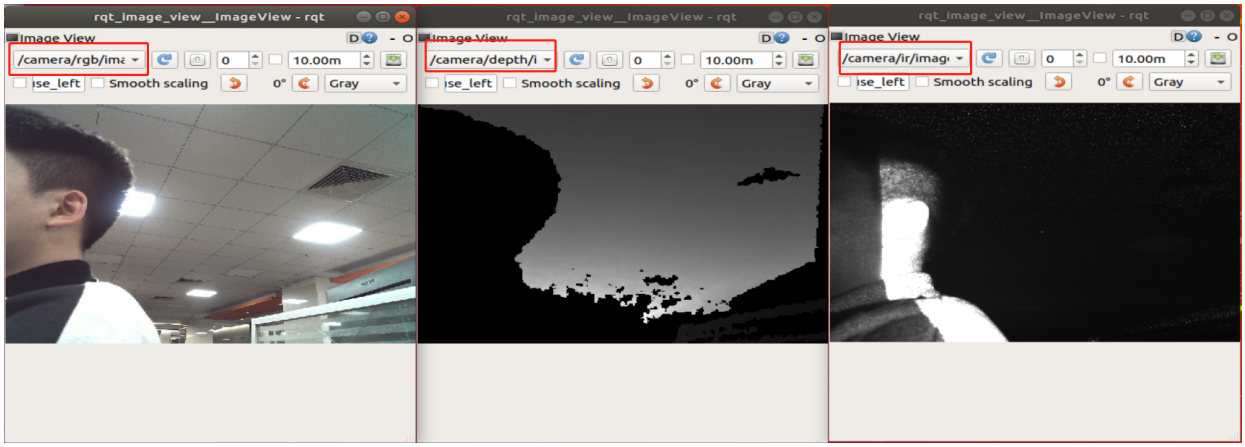
4、run camera

Gemini2 camera running：

```
roslaunch orbbec_camera gemini2.launch
```



Enter rqt_image_view to see if there is image data,



If there is an image display, it means that the environment is built successfully.