

1 How to use the Astra camera

1 How to use the Astra camera

1.1 SDK usage - Linux

1.1.1 dependent environment

1.1.2 Camera SDK&Samples

1.1.3 OpenNI camera test tool

1.2 AstraSDK-win

1.2.1 Install the driver

1.2.2 Download SDK

1.3 OrbbecViewer-win

1.4 Web monitoring

Official website link: <http://www.orbbec.com.cn/>

Developer Community: <https://developer.orbbec.com.cn/>

Astra Camera: https://github.com/orbbec/ros_astra_camera

Normal camera: https://github.com/bosch-ros-pkg/usb_cam.git

Astra SDK: <https://developer.orbbec.com.cn/download.html?id=53>

Basic use of Astra SDK for Windows: https://developer.orbbec.com.cn/course_details.html?id=53

Astra SDK environment setup: https://developer.orbbec.com.cn/course_details.html?id=16

- Create astra udev rule

In the new environment, you need to execute the [create_udev_rules] file in the [scripts] folder of the [astra_camera] function package, enter the directory where the file is located, and execute the command

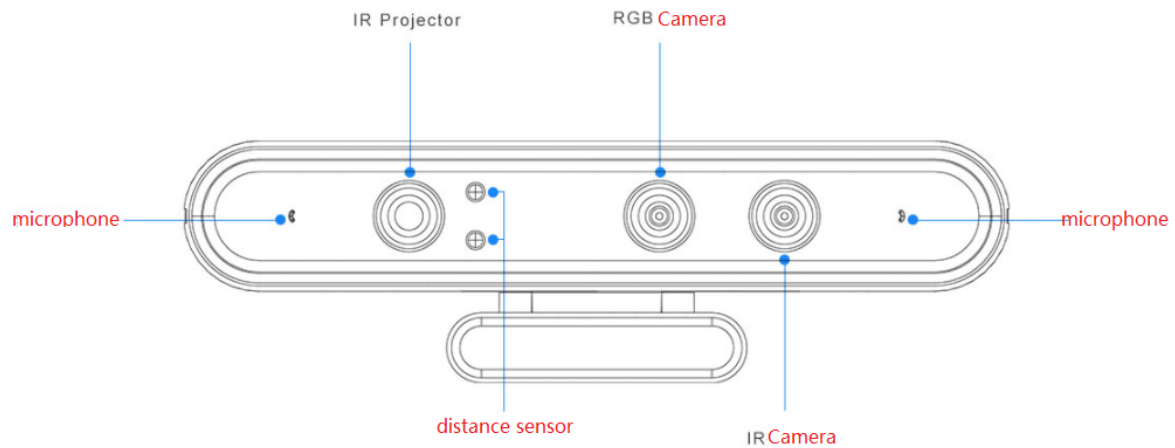
```
./create_udev_rules
```

launch start command

Launch file	Start the camera model
astra.launch	Astra, Astra S, Astra mini, Astra mini S
astraplus.launch	more
astrapro.launch	Astra pro
embedded_s.launch	Deeyea
dabai_u3.launch	Nature
gemini.launch	Gemini

Camera hardware structure diagram:

Hardware Introduction



1.1 SDK usage - Linux

Operating environment: virtual machine or dual system

Developer Community: <https://developer.orbbec.com.cn/download.html?id=53>

1.1.1 dependent environment

```
sudo apt-get install ros-melodic-serial ros-melodic-bfl ros-melodic-mbf-msgs ros-melodic-pointcloud-to-laserscan ros-melodic-rgbd-launch ros-melodic-libuvc-* ros-melodic-uvic-camera ros-melodic-usb-cam ros-melodic-ar-track-alvar ros-melodic-camera-calibration build-essential freeglut3 freeglut3-dev libsfm1-dev
```

Go to the developer community to download the SDK file, that is(Astra SDK and OpenNI2 SDK, the version and system architecture must match).

Product Center Developer Center open platform Auditorium blog 🔍 👤 [creation](#)

forum Activity

development board >

SDK ▾

Orbbec SDK >

Astra SDK 2.1 ▾

Android

Linux

Unity

Windows

OpenNI2 SDK >

drive

Application SDK >

Linux

file name	release date	Version size
AstraSDK-v2.1.3-Ubuntu16.04-x86_64.tar	2021-07-12	23.96MB ▾
AstraSDK-v2.1.3-Ubuntu18.04-x86_64.tar	2021-07-12	23.9MB ▾
AstraSDK-v2.1.3-Linux-arm.tar	2021-07-12	23.87MB ▾
AstraSDK-v2.1.3-Linux-aarch64.tar	2021-07-12	24.11MB ▾

Note: All searches on the Internet are for the latest version. The versions in our supporting materials include [v2.1.2], [v2.1.3], etc. The following takes [v2.1.2] as an example, other versions are similar.

1.1.2 Camera SDK&Samples

The folder name and file path may not be the same, change them according to your needs.

```
tar -zxvf AstraSDK-v2.1.2-Ubuntu18.04-x86_64.tar.gz
cd AstraSDK-v2.1.2-Ubuntu18.04-x86_64/install # Go to the install folder
sudo sh ./install.sh
```

The output contains the following two lines, pay attention to delete the install in the penultimate path:

```
export ASTRA_SDK_INCLUDE=/home/yahboom/software/AstraSDK-v2.1.2-Ubuntu18.04-
x86_64/install/include
export ASTRA_SDK_LIB=/home/yahboom/software/AstraSDK-v2.1.2-Ubuntu18.04-
x86_64/install/lib
```

After deleting install:

```
export ASTRA_SDK_INCLUDE=/home/yahboom/software/AstraSDK-v2.1.2-Ubuntu18.04-
x86_64/include
export ASTRA_SDK_LIB=/home/yahboom/software/AstraSDK-v2.1.2-Ubuntu18.04-
x86_64/lib
```

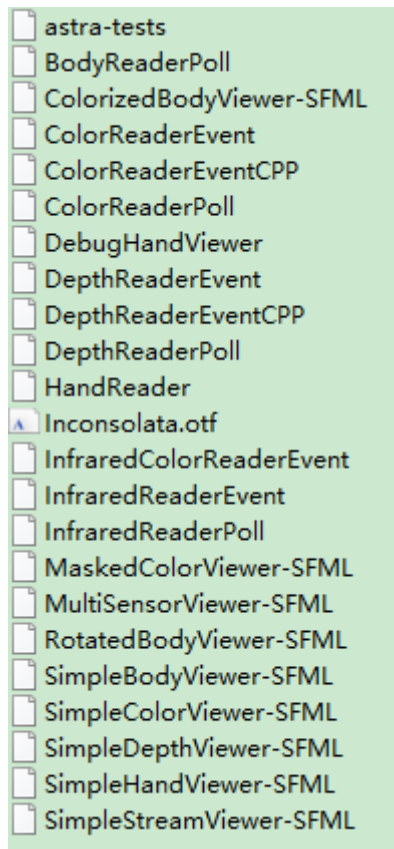
Copy the output to the end of ~/.bashrc

```
gedit ~/.bashrc
source ~/.bashrc
```

The amples directory is the sample program, which needs to depend on the include and lib directories

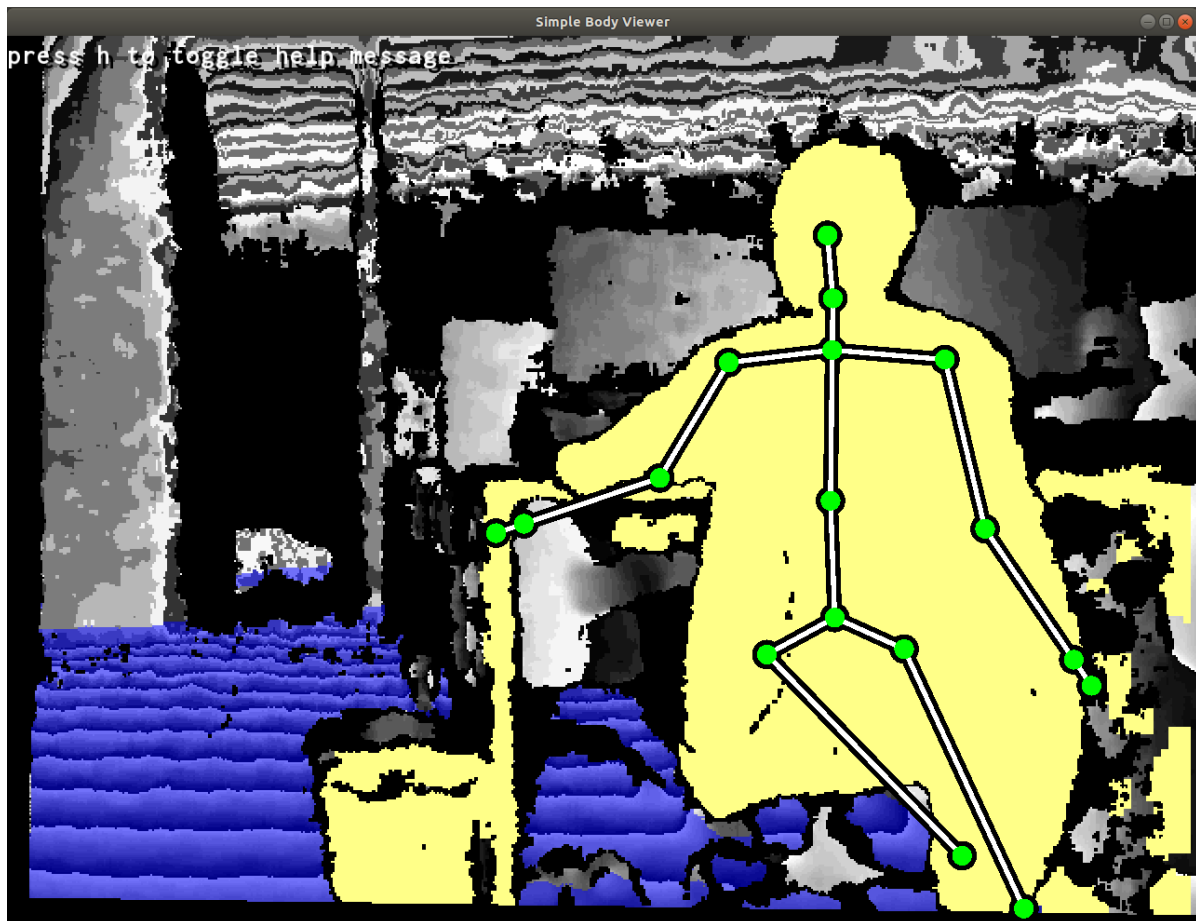
- SFML effect demo

The bin folder is as follows:



Note: `sudo ./` or `./` can be used to start the bin folder, and the files with the suffix -SFML will be displayed on the screen; the methods are similar, and other effects can be tested. If the virtual machine fails to start, please try several times, it is easier to start under the dual system.

```
cd ~/AstraSDK-v2.1.2-Ubuntu18.04-x86_64/bin/  
./SimpleBodyViewer-SFML      # skeleton detection  
./SimpleHandViewer-SFML     # finger following
```



1.1.3 OpenNI camera test tool

Install OpenNI

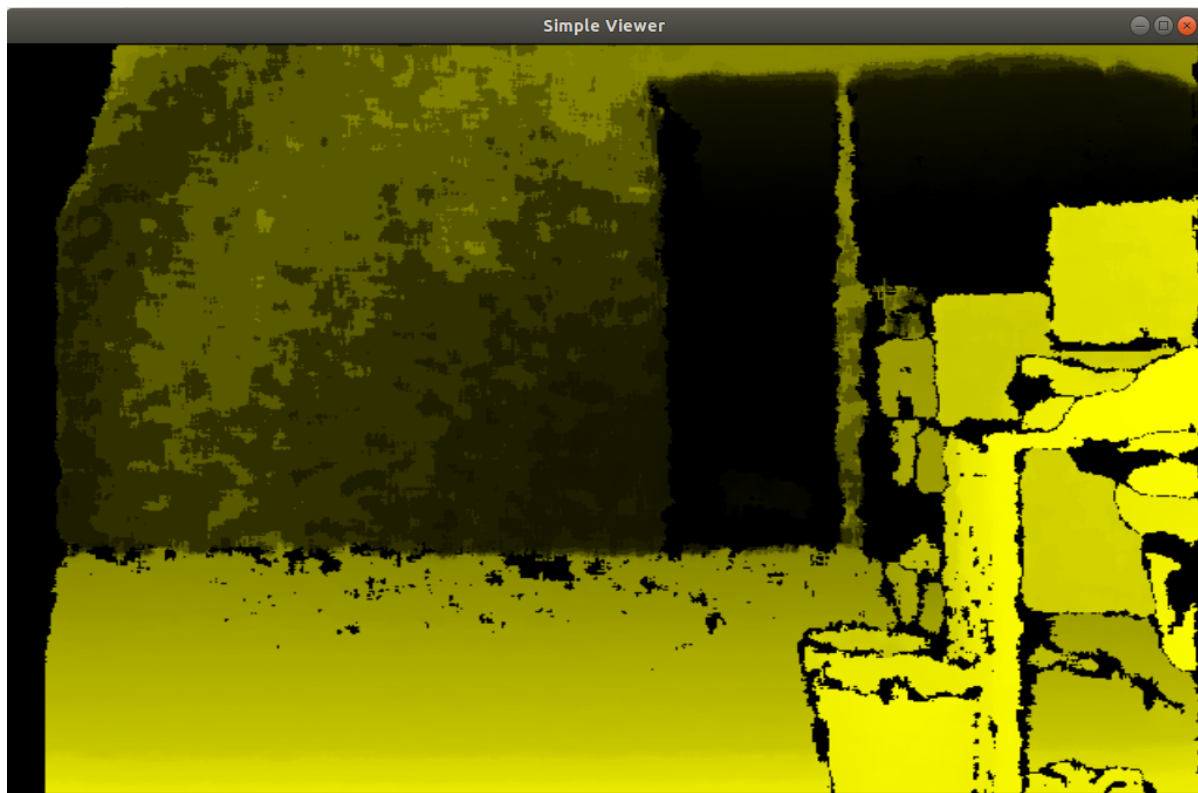
```
unzip OpenNI_2.3.0.55.zip
cd OpenNI_2.3.0.55/Linux/OpenNI-Linux-x64-2.3.0.55
chmod +x install.sh
sudo ./install.sh
```

the device to initialize the OpenNI environment

```
source OpenNIDevEnvironment
```

compile and run

```
cd Samples/Simpleviewer
make
cd Bin/x64-Release
./Simpleviewer
```



1.2 AstraSDK-win

<https://developer.orbbec.com.cn/download.html?id=32>

1.2.1 Install the driver

[Product Center](#) [Developer Center](#) [open platform](#) [Auditorium](#) [blog](#) [Q](#) [person](#) [creation](#)

[forum](#) [Activity](#)

development board >

SDK >

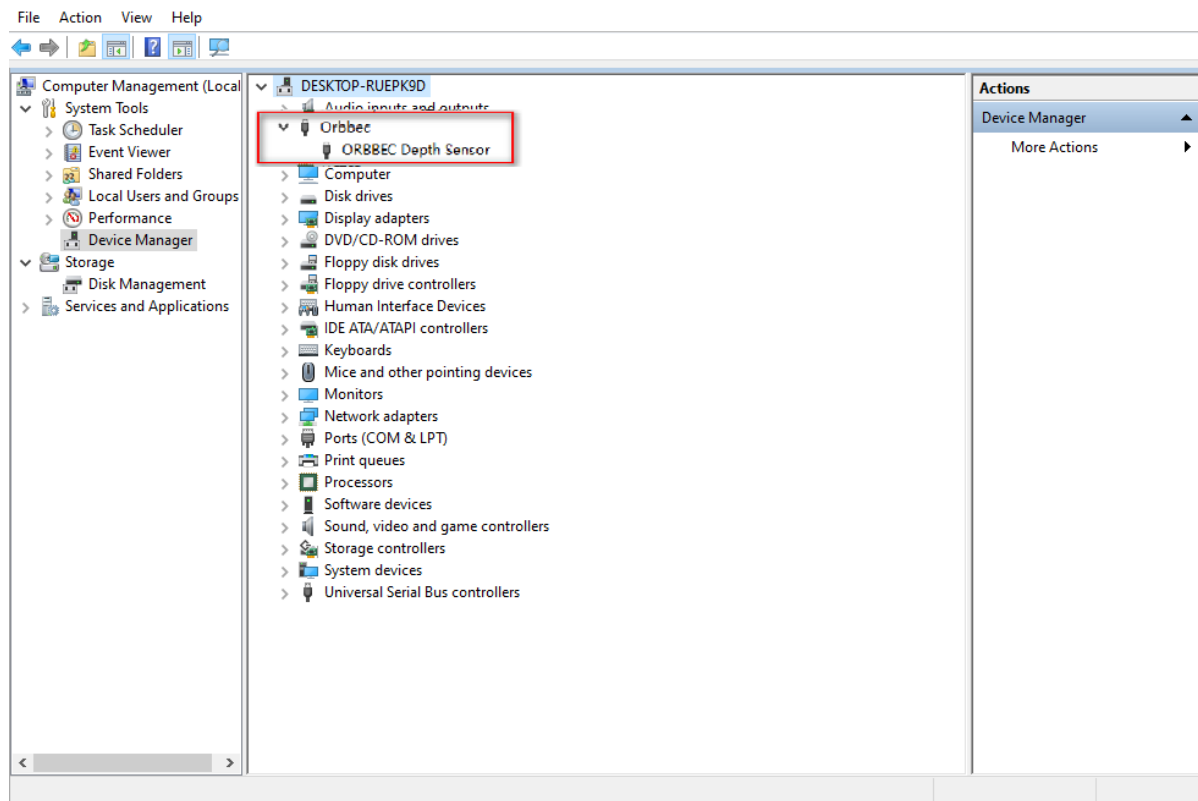
drive

Application SDK >

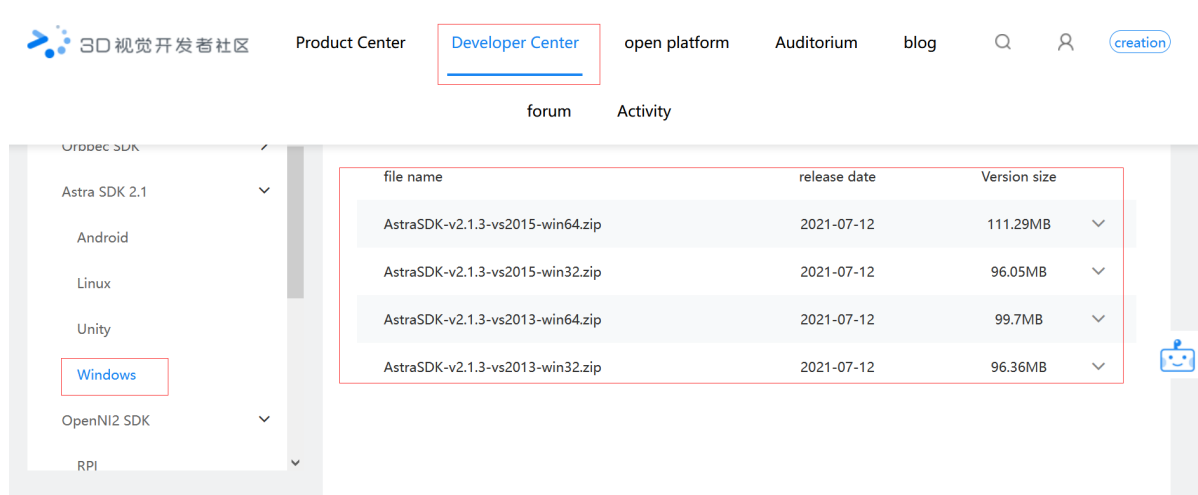
drive

file name	release date	Version size
SensorDriver_V4.3.0.17	2020-11-13	4.16M

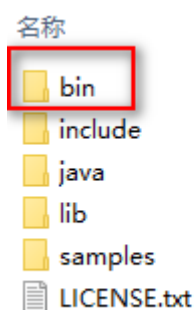
After the download is complete, double-click to install it. The signs of success are as follows



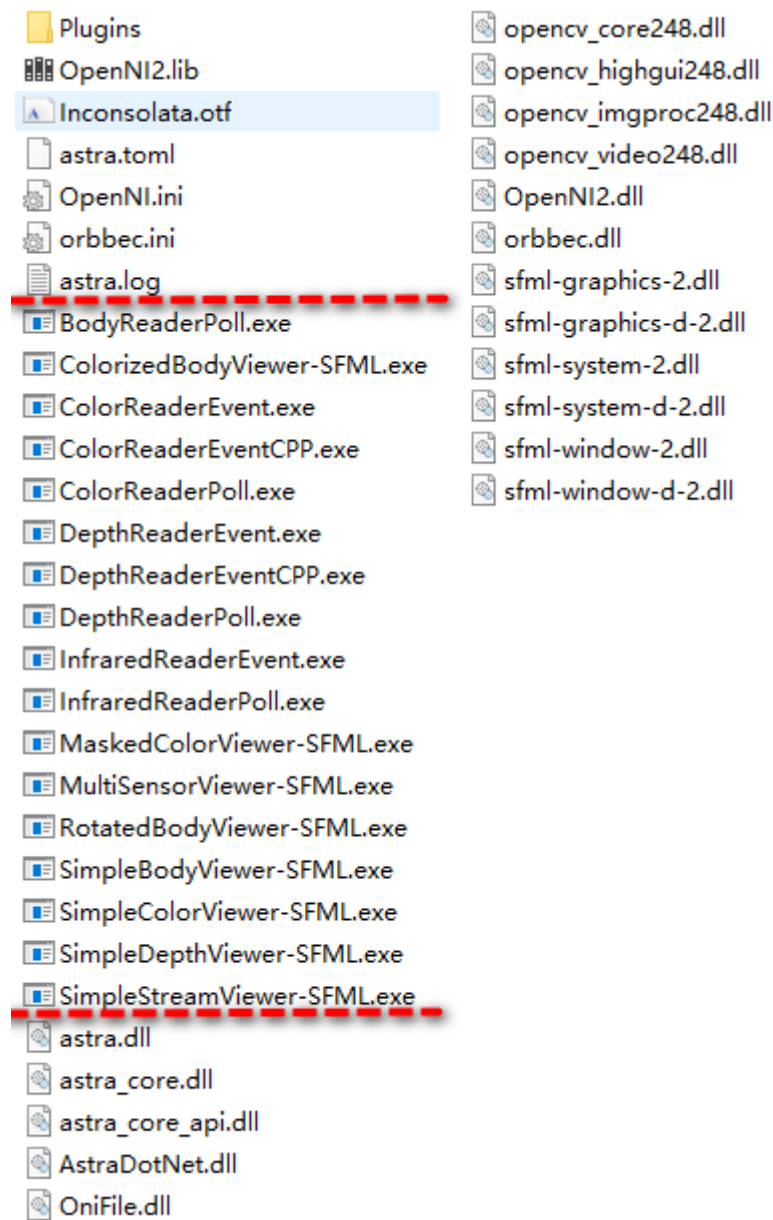
1.2.2 Download SDK



After the download is complete, unzip the folder,

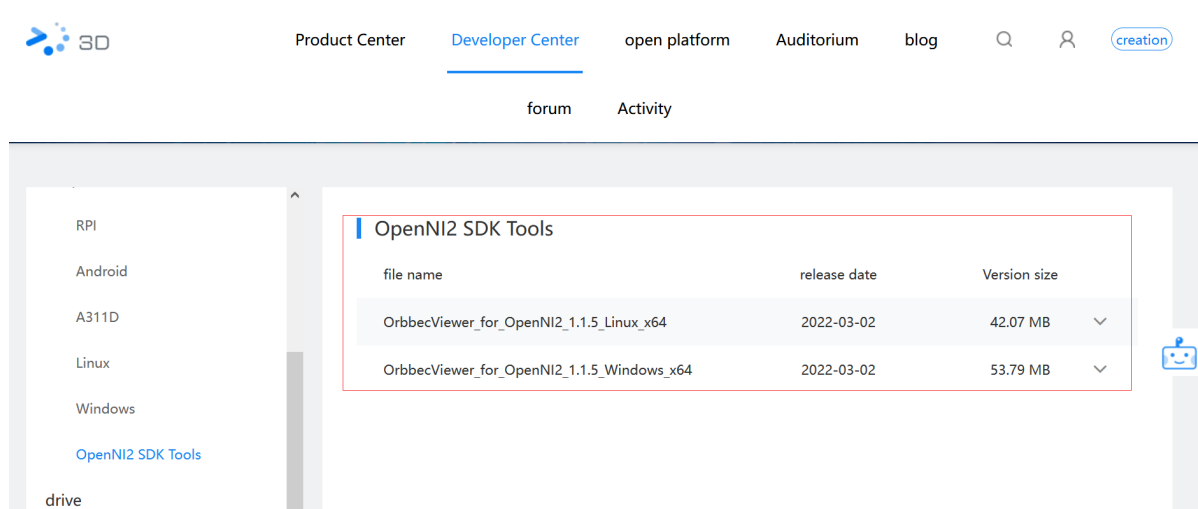



Enter the bin folder and double-click any file with the suffix exe to test it.

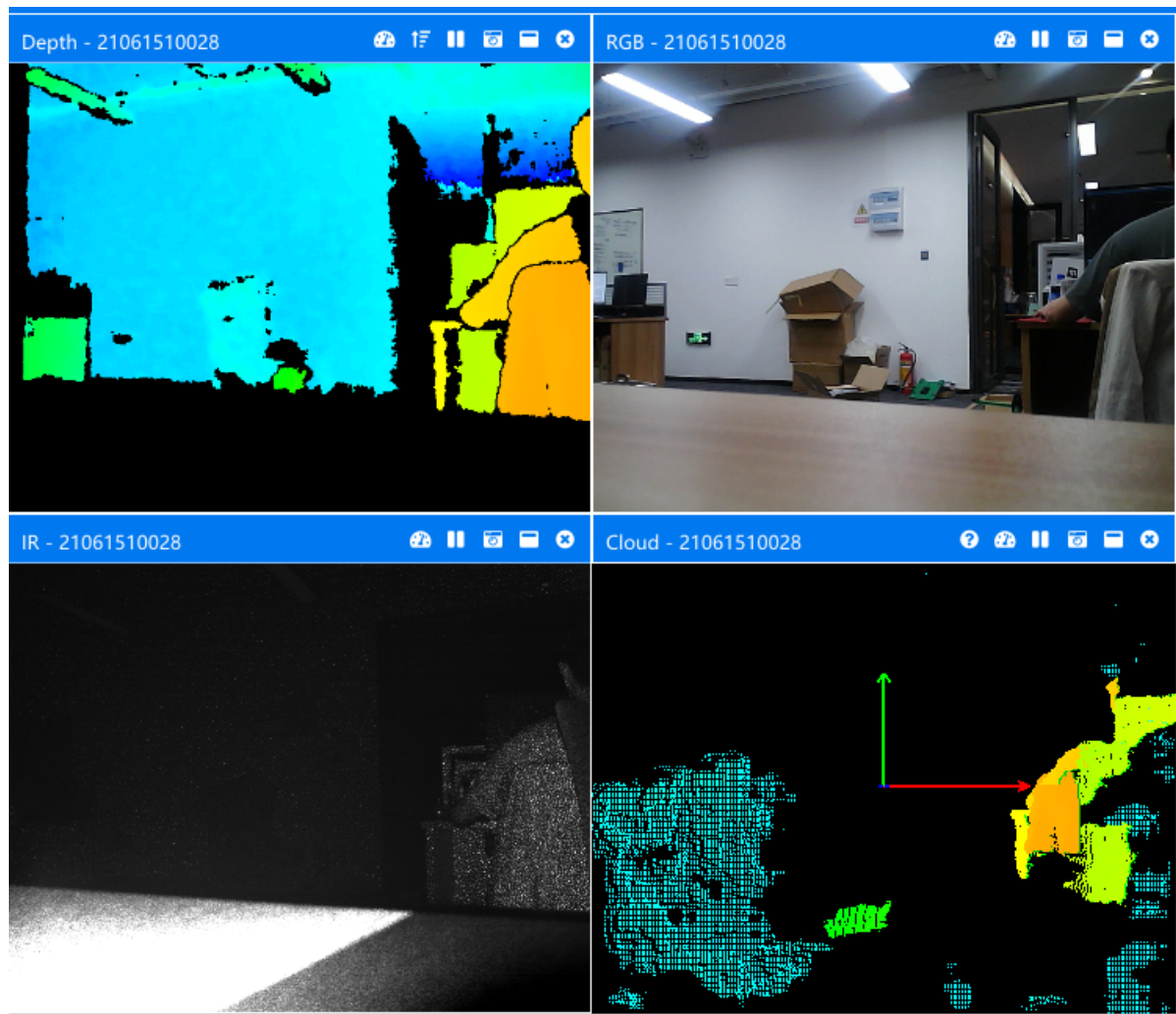


1.3 OrbbecViewer-win

<https://developer.orbbec.com.cn/download.html?id=77>



Unzip, enter the OrbbecViewer_v1.1.1 folder, double-click  **OrbbecViewer.exe** That's it.



1.4 Web monitoring

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS2/07, Docker
tutorial
~/run_docker.sh
```

Environment construction

```
sudo apt-get install ros-melodic-async-web-server-cpp ros-melodic-web-video-
server ros-melodic-usb-cam
```

Start the camera

```
roslaunch astra_camera astraproplus.launch      # Astra
roslaunch usb_cam usb_cam-test.launch           # USB
```

<PI5 needs to open another terminal and enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

```
docker ps -a
```

```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9   "/bin/bash"            3 days ago    Up 9 hours                   ecstatic_lewin
jetson@ubuntu:~$
```

2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```

```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9   "/bin/bash"            3 days ago    Up 9 hours                   ecstatic_lewin
jetson@ubuntu:~$ docker exec -it 5b698ea10535 /bin/bash
-----
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-----
root@ubuntu:/#
```

After successfully entering the container, you can open countless terminals to enter the container.

start web_video_server

```
roslaunch web_video_server web_video_server
```

Check

view in local web browser

http://localhost:8080/

It must be under the same local area network, and other devices can view it

http://192.168.2.103:8080/

(192.168.2.103 is the IP address of the master)

Note: It is recommended to use Google Chrome or mobile QQ browser, other browsers may not be able to open the image