5. Web page real-time monitoring

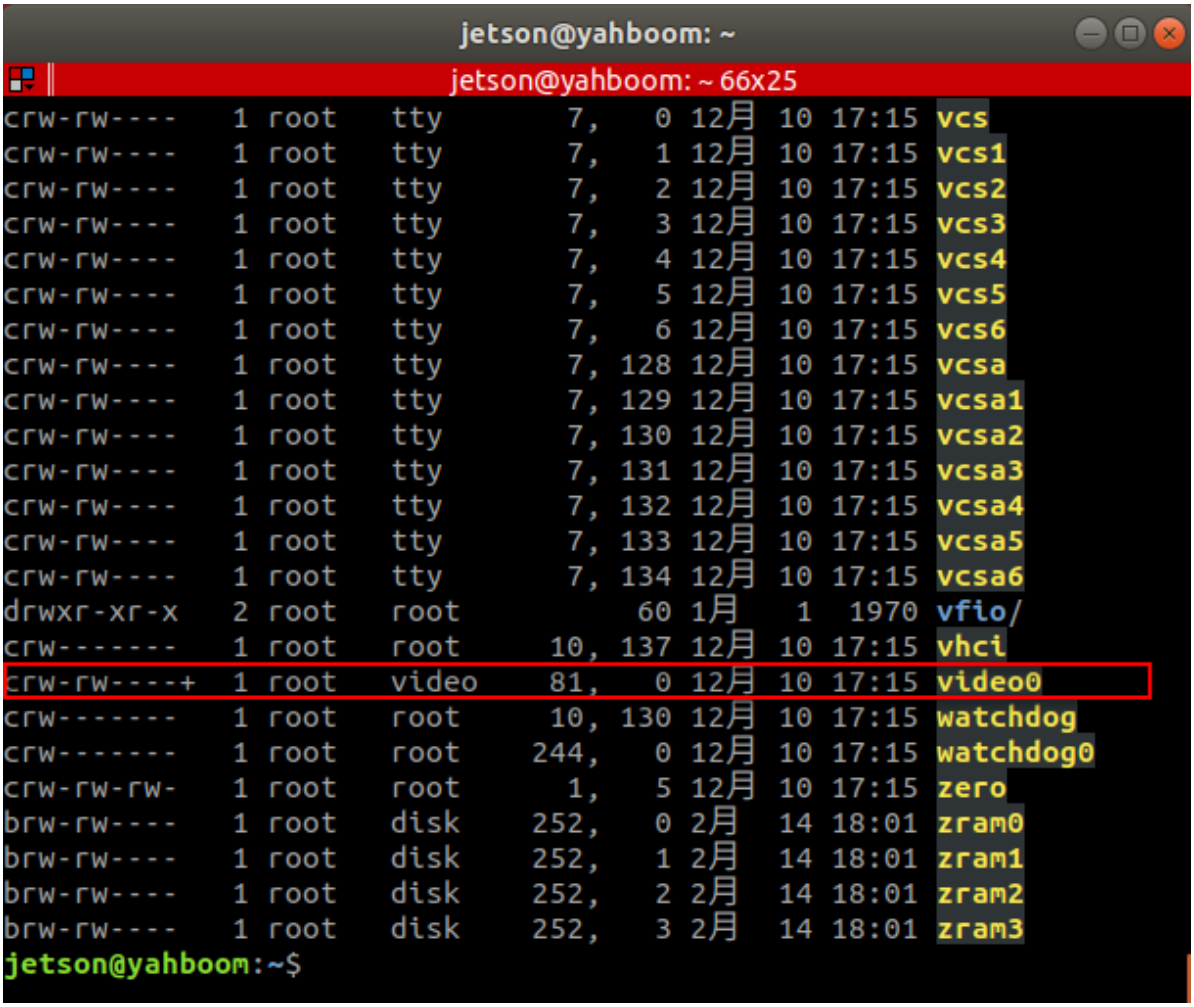
5. Web page real-time monitoring
    5.1. Environment Construction
    7.2. modify the launch file
    7.3. effect demonstration

5.1. Environment Construction

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

First make sure that the USB camera link is correct, enter the following command to check that the USB device number exists and is video0

```
ll  /dev
```



If the execute permission is not enough, you need to add the execute permission

```
sudo chmod 777 /dev/video*
```

7.2. modify the launch file

```
sudo vim /opt/ros/melodic/share/usb_cam/launch/usb_cam-test.launch
```

Keep hitting the keyboard [d] to delete everything. Click [i] on the keyboard to enter the edit mode, and write the following content into it.

```
< launch >
    < arg  name = "open_view"  default = "false" />
    < node  name = "usb_cam"  pkg = "usb_cam"  type = "usb_cam_node"  output =
"screen" >
        < param  name = "video_device"  value = "/dev/video0" />
        < param  name = "image_width"  value = "640" />
        < param  name = "image_height"  value = "480" />
        < param  name = "pixel_format"  value = "yuyv" />
        < param  name = "camera_frame_id"  value = "usb_cam" />
        < param  name = "io_method"  value = "mmap" />
    </ node >
    <!-- web_video_server -->
    < node  pkg = "web_video_server"  type = "web_video_server"  name =
"web_video_server"  output = "screen" />
    <!-- image_view -->
    < group  if = "$(arg open_view)" >
        < node  name = "image_view"  pkg = "image_view"  type = "image_view"
respawn = "false"  output = "screen" >
            < remap  from = "image"  to = "/usb_cam/image_raw" />
            < param  name = "autosize"  value = "true" />
        </ node >
    </ group >
</ launch >
```

Click the [ESC] key twice (or multiple times), click the keyboard [shift plus;], enter [wq], and click the [Enter] key.

7.3. effect demonstration

open terminal, start

```
roslaunch usb_cam usb_cam-test.launch
```

<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

```
rosrun web_video_server web_video_server
```

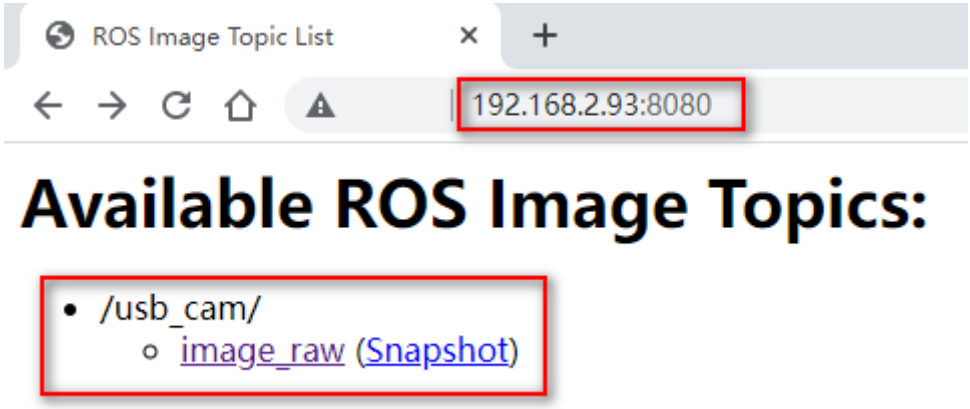
- View in local web browser

```
http://localhost:8080/
```

- View other devices (must be under the same LAN, 192.168.2.93 is the IP address of the master)

```
http://192.168.2.93:8080/
```

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



Click [image_raw] to view the camera image in real time, and click [Snapshot] to display only one frame of image.