5. Enter the docker container of the robot

The operating environment and hardware and software reference configuration are as follows:

- Reference model: ROSMASTER X3
- Robot hardware configuration: Arm series main control, Silan A1 LiDAR, AstraPro Plus depth camera.
- Robot system: Ubuntu (version not required) + docker (version 20.10.21 and above)
- PC virtual machine: Ubuntu (20.04) + ROS2 (Foxy)
- Usage scenario: use on a relatively clean 2D plane

5.1, related concepts

1. what is docker's host

```
宿主机就是我们调用命令使用镜像创建容器的服务器。这里是指我们小车上的主控（jetson或树莓派等），以下提到的宿主机都是指这个。
The host is the server where we invoke the command to create the container using the image. In this case it is the host on our cart (jetson or raspberry pi, etc.), and all references to hosts below refer to this.
```

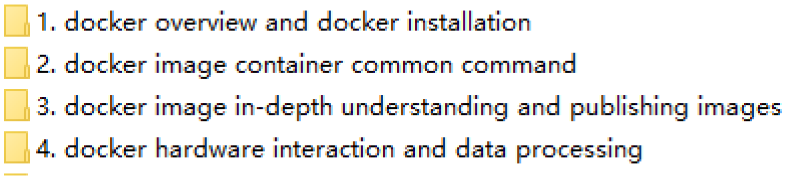
2. What is GUI

```
GUI即图形用户界面，这里主要是指：opencv显示的图像窗口、rviz界面、rqt界面等。
GUI is graphical user interface, here mainly refers to: opencv display image window, rviz interface, rqt interface and so on.
```

3. What is a docker container for robots

```
这里的机器人就是Rosmaster小车，也就是已配置好各种开发依赖环境的Rosmaster小车容器
The robot here is the Rosmaster cart, which is the Rosmaster cart container that
has been configured with various development dependencies
```

4, in the operation of this chapter of the tutorial before, please ensure that you have mastered the following knowledge of these chapters, otherwise you may feel that learning is more strenuous. In this case, please repeatedly check the content of the following pre-knowledge, skilled mastery will feel very easy, Come on, you are the best!



1. docker overview and docker installation
2. docker image container common command
3. docker image in-depth understanding and publishing images
4. docker hardware interaction and data processing

5.2. How to query the docker image version used by the robot

1, the version of the docker image used by the robot is also the version of the image used on the cart, which the user executes after burning the cart's system image to boot:

```
jetson@jetson-desktop:~$ docker images
REPOSITORY                   TAG       IMAGE ID       CREATED              SIZE
yahboomtechnology/ros-foxy   3.5.4     d307ad9f2cda   About a minute ago
14.2GB
192.168.2.51:5000/ros-foxy   1.0.0     31e97028c1c0   3 days ago
14.2GB
yahboomtechnology/ros-foxy   3.5.3     31e97028c1c0   3 days ago
14.2GB
```

Will see that there are multiple docker image version, please select [yahboomtechnology/ros-foxy] name, tag the highest version, that is, the latest version of the robot image. Such as the query here on the use of [yahboomtechnology/ros-foxy: 3.5.4] version, [192.168.2.51:5000/ros-foxy] the name of the mirror do not care.

2, why can not just put a docker image in the cart system?

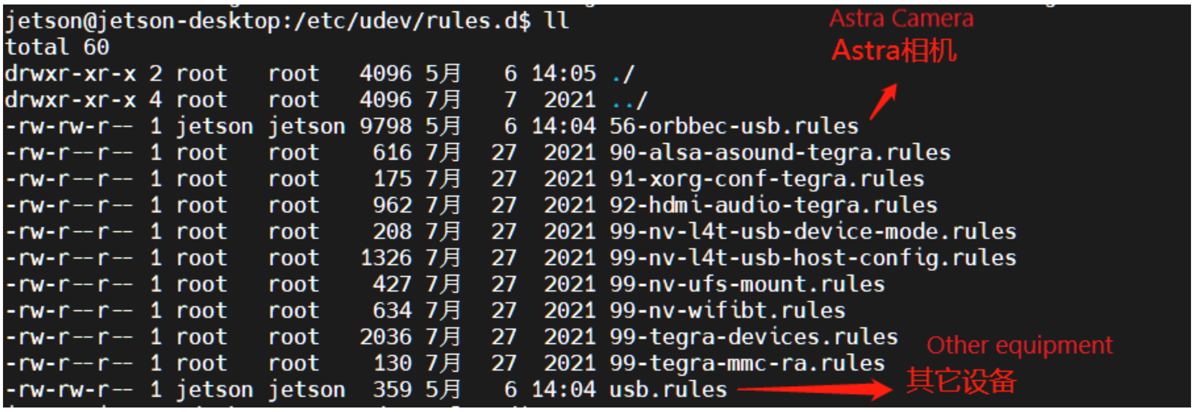
If you have read [07, Docker ------ 3, docker image in-depth understanding of the tutorials in this chapter and release images], you should know that docker image is a layered mechanism, that is, the back of a tag of the image depends on the front of a tag of the image. So there may be multiple versions of docker images in the host, and the tags of these images are updated in an incremental manner.

When we update the course, we will also update the functionality by releasing a new docker image.

5.3 Binding peripherals

- First of all, make sure that the cart is connected to various peripherals, and that the peripherals have been port-bound, which is handled on the docker's host (the cart).
- Common peripherals are: serial devices, LIDAR, RGBD camera, voice control module, joystick remote control, etc.
- **Default trolley has bound Astra camera, LIDAR and serial devices**, if you need to bind other devices, please refer to the port binding tutorial.
- For the steps of port binding, please refer to [VI. Linux OS ------ 06. Binding Device ID] tutorial section.

Port binding has been configured within the host, if you need to modify it, you can check the content inside to modify it:



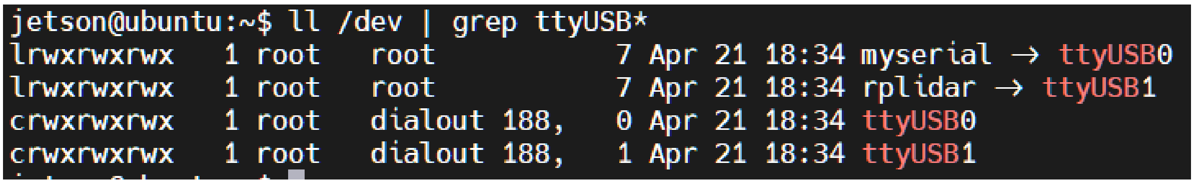
```
jetson@jetson-desktop:/etc/udev/rules.d$ ll
total 60
drwxr-xr-x 2 root   root   4096 5月   6 14:05 ./
drwxr-xr-x 4 root   root   4096 7月   7  2021 ../
-rw-rw-r-- 1 jetson jetson 9798 5月   6 14:04 56-orbbec-usb.rules
-rw-r--r-- 1 root   root    616 7月  27  2021 90-alsa-asound-tegra.rules
-rw-r--r-- 1 root   root    175 7月  27  2021 91-xorg-conf-tegra.rules
-rw-r--r-- 1 root   root    962 7月  27  2021 92-hdmi-audio-tegra.rules
-rw-r--r-- 1 root   root    208 7月  27  2021 99-nv-l4t-usb-device-mode.rules
-rw-r--r-- 1 root   root   1326 7月  27  2021 99-nv-l4t-usb-host-config.rules
-rw-r--r-- 1 root   root    427 7月  27  2021 99-nv-ufs-mount.rules
-rw-r--r-- 1 root   root    634 7月  27  2021 99-nv-wifibt.rules
-rw-r--r-- 1 root   root   2036 7月  27  2021 99-tegra-devices.rules
-rw-r--r-- 1 root   root    130 7月  27  2021 99-tegra-mmc-ra.rules
-rw-rw-r-- 1 jetson jetson  359 5月   6 14:04 usb.rules
```

5.4. Checking Peripheral Connections

This step is operated on the host computer:

1, here is to view the peripherals in addition to the camera, here is not connected to the voice control module, connected will show [myspeech] device

```
ll /dev | grep ttyUSB*
```



```
jetson@ubuntu:~$ ll /dev | grep ttyUSB*
lrwxrwxrwx   1 root   root          7 Apr 21 18:34 myserial -> ttyUSB0
lrwxrwxrwx   1 root   root          7 Apr 21 18:34 rplidar -> ttyUSB1
crwxrwxrwx   1 root   dialout 188,  0 Apr 21 18:34 ttyUSB0
crwxrwxrwx   1 root   dialout 188,  1 Apr 21 18:34 ttyUSB1
```

2. Check the ports of the AstraPro Plus camera as follows:

```
jetson@ubuntu:~$ ll /dev/astra*
lrwxrwxrwx 1 root root 15 May  5 17:42 /dev/astradepth -> bus/usb/001/007
lrwxrwxrwx 1 root root 15 May  5 17:42 /dev/astrauvc -> bus/usb/001/009
```

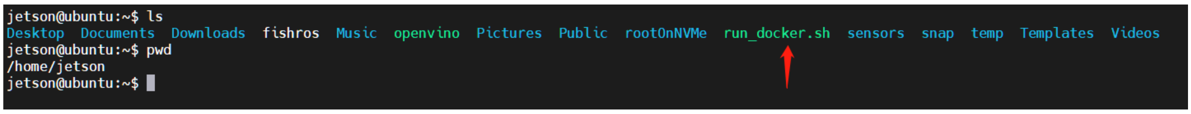
5.5. Editing Scripts

Since many times the port number changes after the AstraPro Plus camera is plugged in, you need to re-edit the script to configure the AstraPro Plus camera port.

Edit the script to run docker, this step is operated on the host computer:

1, the script to run docker [run_docker.sh] will usually be placed in the root directory of the main directory of the cart , I here is in the following path. If you don't have it, you can create the file by yourself, and remember to give the script executable permissions after creation.

```
chmod +x run_docker.sh  #给脚本赋予可执行权限
 # Give scripts executable permissions
```



```
jetson@ubuntu:~$ ls
Desktop  Documents  Downloads  fishros  Music  openvino  Pictures  Public  rootOnNVMe  run_docker.sh  sensors  snap  temp  Templates  Videos
jetson@ubuntu:~$ pwd
/home/jetson
jetson@ubuntu:~$ 
```

The contents of the [run_docker.sh] script are as follows:

You can copy it without comments and modify it as needed.

Note: When you add a host device to a container, if the device is not connected to the host, you need to remove the corresponding add operation before you can open the container.

```
#!/bin/bash
xhost +

docker run -it \
--net=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
-v /tmp/.X11-unix:/tmp/.X11-unix \
-v /home/jetson/temp:/root/yahboomcar_ros2_ws/temp \
-v /home/jetson/rosboard:/root/rosboard \
-v /home/jetson/maps:/root/maps \
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \
--device=/dev/astradepth \
--device=/dev/astrauvc \
--device=/dev/video0 \
--device=/dev/myserial \
--device=/dev/rplidar \
--device=/dev/input \
-p 9090:9090 \
-p 8888:8888 \
yahboomtechnology/ros-foxy:3.5.4 /bin/bash
```

Script instructions with comments:

Note: When you add a host device to a container below, if the device is not connected to the host, you need to remove the corresponding add operation to open the container

```
#!/bin/bash
xhost +                                                # xhost 是用来支持docker内
显示GUI的
# xhost is used to support displaying the GUI within docker

docker run -it \                                       # 交互式运行docker镜像
```

```
# Run docker images interactively
--net=host \                                          # 容器网络设置为host模式
# Container network set to host mode
--env="DISPLAY" \                                     # 开启显示GUI界面
    # Open the display GUI interface
--env="QT_X11_NO_MITSHM=1" \                          # 采用X11的端口1进行显示
# Use port 1 of X11 for display
-v /tmp/.X11-unix:/tmp/.X11-unix \                    # 映射显示服务节点目录
    # Map to show service node catalog
-v /home/jetson/temp:/root/yahboomcar_ros2_ws/temp \  # 作为宿主机和容器临时传输文
件的目录，有需要传输文件，可以使用这个目录
    # As a directory for the temporary transfer of files between the host and the
container, if there is a need to transfer files, you can use this directory
-v /home/jetson/rosboard:/root/rosboard \             # app建图导航用到的目录
    # Catalogs used for app building navigation
-v /home/jetson/maps:/root/maps \                     # app建图导航用到的目录
    # Catalogs used for app building navigation
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \        # 添加主机设备给容器，这里是
astrpro plus设备端口,如何小车没有连接摄像头，请去掉这行
   # Add the host device to the container, here is the astrpro plus device port,
if the cart is not connected to the camera, please remove this line.
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \        # 添加主机设备给容器，这里是
astrpro plus设备端口,如何小车没有连接摄像头，请去掉这行
    # Add the host device to the container, here is the astrpro plus device port,
if the cart is not connected to the camera, please remove this line.
--device=/dev/astradepth \                            # 添加主机设备给容器，这里是
astrpro plus设备端口,如何小车没有连接摄像头，请去掉这行
# Add the host device to the container, here is the astrpro plus device port, if
the cart is not connected to the camera, please remove this line.
--device=/dev/astrauvc \                              # 添加主机设备给容器，这里是
astrpro plus设备端口,如何小车没有连接摄像头，请去掉这行
# Add the host device to the container, here is the astrpro plus device port, if
the cart is not connected to the camera, please remove this line.
--device=/dev/video0 \                                # 添加主机设备给容器，这里是
astrpro plus设备端口,如何小车没有连接摄像头，请去掉这行
# Add the host device to the container, here is the astrpro plus device port, if
the cart is not connected to the camera, please remove this line.
--device=/dev/myserial \                              # 添加主机设备给容器，这里是
串口设备端口,如何小车没有连接串口，请去掉这行
# Add the host device to the container, here is the serial device port, if the
cart is not connected to the serial port, please remove this line.
--device=/dev/rplidar \                               # 添加主机设备给容器，这里是
雷达设备端口,如何小车没有连接雷达，请去掉这行
# Add the host device to the container, here is the lidar device port, if the car
is not connected to the radar, please remove this line.
--device=/dev/myspeech \                              # 添加主机设备给容器，这里是
语音控制设备端口,如何小车没有连接语音控制设备，请去掉这行
# Add the host device to the container, here is the voice control device port, if
the cart is not connected to the voice control device, please remove this line.
--device=/dev/input \                                 # 添加主机设备给容器，这里是
手柄设备端口,如何小车没有连接手柄，请去掉这行
# Add the host device to the container, here is the joystick device port, if the
cart is not connected to the joystick, please remove this line.
-p 9090:9090 \                                        # 开放的端口# Open ports
-p 8888:8888 \
yahboomtechnology/ros-foxy:3.3.9 /bin/bash            # 要启动的镜像名称，根据5.2步
骤中查询到的修改；在容器内执行/bin/bash命令
```

```
#注意：以上添加主机设备给容器器时，如果宿主机没有连接该设备，需要去掉相应的添加操作，才能开启容器
# The name of the image to be started, modified according to the query in step 5.2; execute the /bin/bash command inside the container.

# Note: When you add a host device to a container, if the host machine is not connected to the device, you need to remove the corresponding add operation to open the container.
```

2, modify the above script, these two lines are AstraPro Plus camera port number, due to the camera plug and play, the port number will change, you need to reconfigure the camera port!

```
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \      # 给容器挂载存储卷，挂载到容器的某个目录，这里挂载的是摄像头的rgb和depth端口
# Mount the storage volume for the container, to a directory in the container, here the rgb and depth ports for the camera are mounted
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \
```

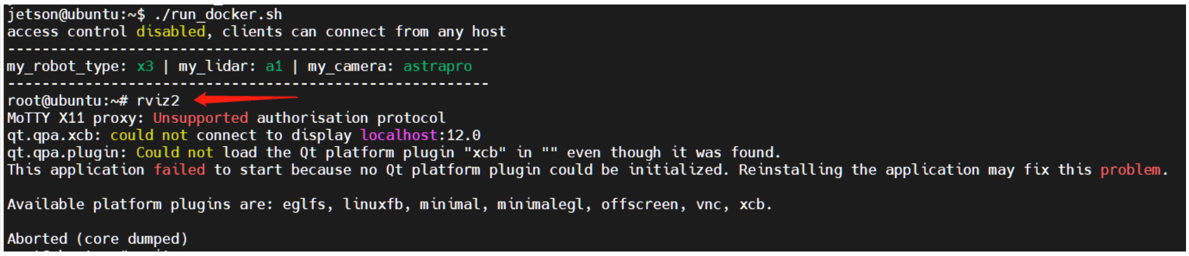
is the camera port queried in step 5.4 2, this port may change after the camera is plugged in, so everyone's is different and you need to match it yourself.

```
-v /dev/bus/usb/001/007:/dev/bus/usb/001/007 \      # 给容器挂载存储卷，挂载到容器的某个目录，这里挂载的是摄像头的rgb和depth端口
# Mount the storage volume for the container, to a directory in the container, here the rgb and depth ports for the camera are mounted
-v /dev/bus/usb/001/009:/dev/bus/usb/001/009 \
```

5.6. Execution of scripts

5.5 After executing the steps, open a terminal on the docker's host [i.e. the cart, either on the VNC or on the cart's screen].

Note: This must be executed on the VNC of the cart or on the screen of the cart, but not in the cart terminal remotely accessed through ssh (e.g., the cart terminal accessed through MobaXterm), otherwise the container may not be able to display the GUI image, as shown below: after entering the cart terminal in MobaXterm and executing run_docker.sh to enter the container, you can not Show rviz

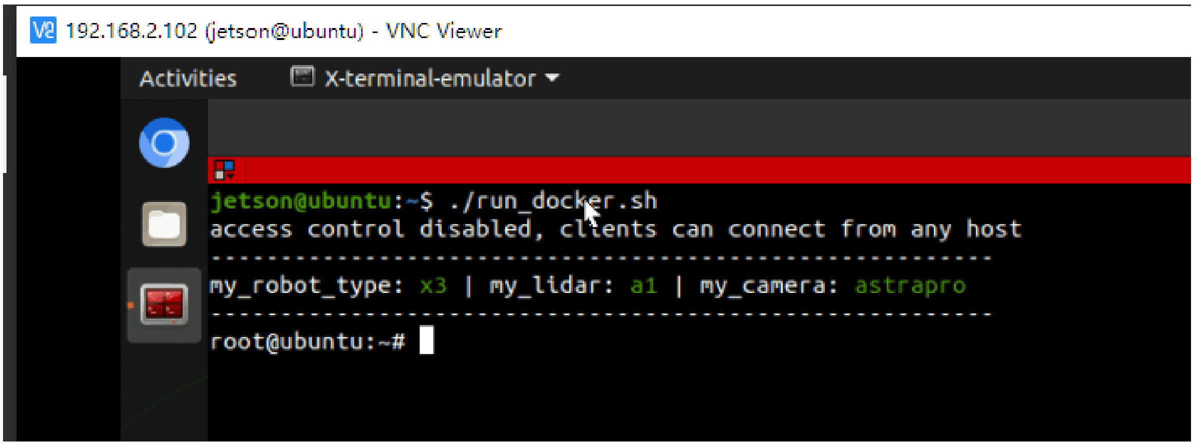


```
jetson@ubuntu:~$ ./run_docker.sh
access control disabled, clients can connect from any host
--------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
--------------------------------------------------------
root@ubuntu:~# rviz2
MoTTY X11 proxy: Unsupported authorisation protocol
qt.qpa.xcb: could not connect to display localhost:12.0
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.
This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb.

Aborted (core dumped)
```

Executed in the VNC interface of the cart or on the cart screen:

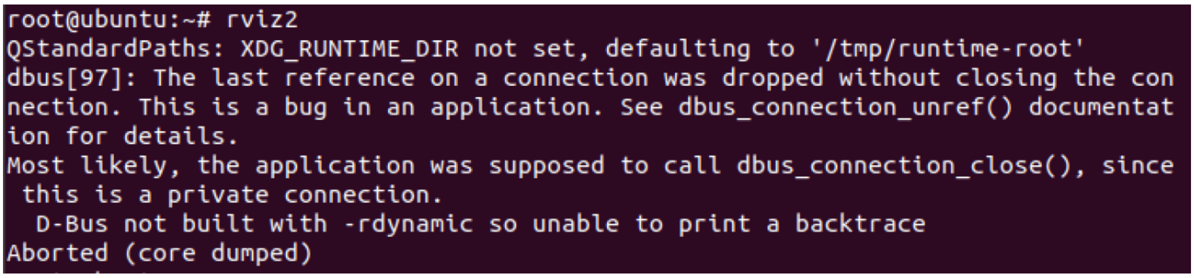
```
./run_docker.sh
```

You can enter the container correctly and display the GUI screen. you can test it by executing the rviz2 command again.



```
jetson@ubuntu:~$ ./run_docker.sh
access control disabled, clients can connect from any host
-----------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-----------------------------------------------------------
root@ubuntu:~#
```

If the GUI cannot be displayed after executing the rviz2 command, the following error is displayed: (generally possible in Raspberry Pi master)



```
root@ubuntu:~# rviz2
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
dbus[97]: The last reference on a connection was dropped without closing the con
nection. This is a bug in an application. See dbus_connection_unref() documentat
ion for details.
Most likely, the application was supposed to call dbus_connection_close(), since
 this is a private connection.
  D-Bus not built with -rdynamic so unable to print a backtrace
Aborted (core dumped)
```

```
--security-opt apparmor:unconfined
```

I.e.:

```
#!/bin/bash
xhost +

docker run -it \
--net=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
-v /tmp/.X11-unix:/tmp/.X11-unix \
--security-opt apparmor:unconfined \                    # 添加了这句参数 #
Added this parameter
-v /home/jetson/temp:/root/yahboomcar_ros2_ws/temp \
-v /home/jetson/rosboard:/root/rosboard \
-v /home/jetson/maps:/root/maps \
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \
--device=/dev/astradepth \
--device=/dev/astrauvc \
--device=/dev/myserial \
--device=/dev/rplidar \
--device=/dev/myspeech \
--device=/dev/input \
-p 9090:9090 \
-p 8888:8888 \
yahboomtechnology/ros-foxy:3.3.9 /bin/bash
```

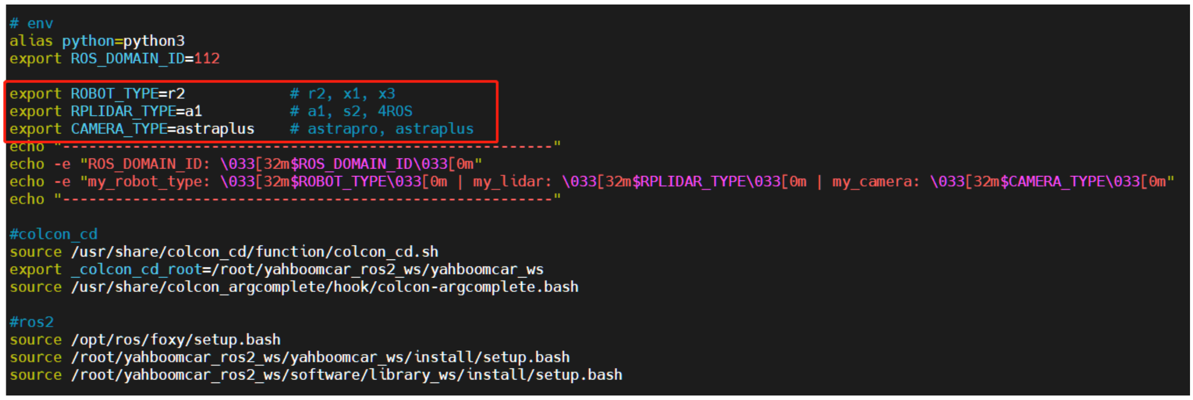
Then run the script again to access the container and be able to display the GUI screen.

5.7. Switching Models, Lidar and Cameras

Description: Since the ROSMASTER series robots are divided into multiple robots as well as multiple devices, the factory system has been configured with routines for multiple devices, but since it is not possible to automatically recognize the products, you need to manually set up the machine type and LIDAR model.

After entering the container: according to the model of the trolley, the type of LIDAR and the type of CAMERA make the following changes:

```
root@ubuntu:/# cd
root@ubuntu:~# vim .bashrc
```



```
# env
alias python=python3
export ROS_DOMAIN_ID=112

export ROBOT_TYPE=r2            # r2, x1, x3
export RPLIDAR_TYPE=a1          # a1, s2, 4ROS
export CAMERA_TYPE=astraplus    # astrapro, astraplus
echo "--------------------------------------------------------"
echo -e "ROS_DOMAIN_ID: \033[32m$ROS_DOMAIN_ID\033[0m"
echo -e "my_robot_type: \033[32m$ROBOT_TYPE\033[0m | my_lidar: \033[32m$RPLIDAR_TYPE\033[0m | my_camera: \033[32m$CAMERA_TYPE\033[0m"
echo "--------------------------------------------------------"

#colcon_cd
source /usr/share/colcon_cd/function/colcon_cd.sh
export _colcon_cd_root=/root/yahboomcar_ros2_ws/yahboomcar_ws
source /usr/share/colcon_argcomplete/hook/colcon-argcomplete.bash

#ros2
source /opt/ros/foxy/setup.bash
source /root/yahboomcar_ros2_ws/yahboomcar_ws/install/setup.bash
source /root/yahboomcar_ros2_ws/software/library_ws/install/setup.bash
```

When the changes are complete, save and exit vim, then execute:

```
root@ubuntu:~# source .bashrc
--------------------------------------------------------
ROS_DOMAIN_ID: 12
my_robot_type: x3 | my_lidar: a1 | my_camera: astraplus
--------------------------------------------------------
root@ubuntu:~#
```

You can see the current model of the modified cart, the lidar type and the camera type.

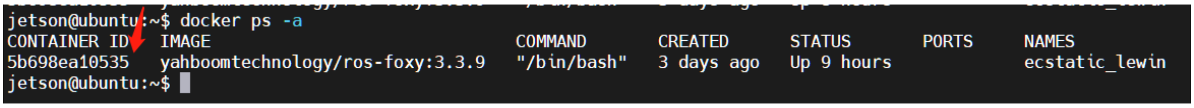
The robot project files are stored in the following directory:

```
/root/yahboomcar_ros2_ws
```

5.8 Multi-terminal access to the same docker container

1, in the above steps have opened a docker container, you can open another terminal on the host (cart) to view:

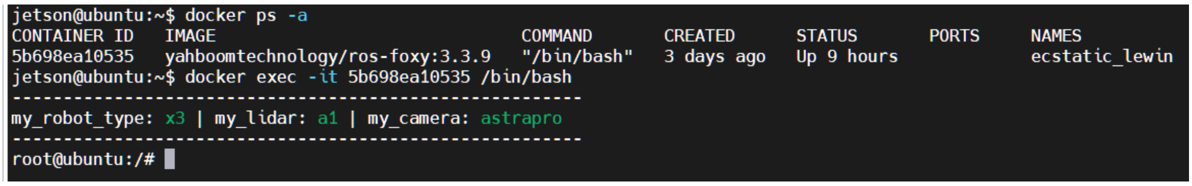
```
docker ps -a
```



```
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 that docker container in this 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:/#
```

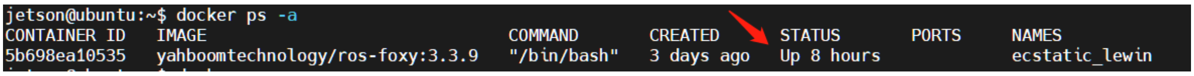
Successfully enter the container and also open numerous more terminals to enter that container.

The robot project files are stored in the following directory:

```
/root/yahboomcar_ros2_ws
```

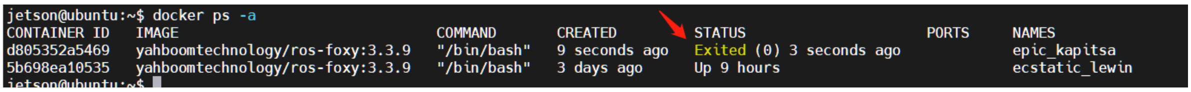
3. Note:

(1) Ensure that the container is turned on [UP] when executing the commands in step 2.



```
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 8 hours             ecstatic_lewin
```

(2) If the container is in the [Exited] closed state, see step 5.8 below.



```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                             COMMAND       CREATED         STATUS                     PORTS     NAMES
d805352a5469   yahboomtechnology/ros-foxy:3.3.9  "/bin/bash"   9 seconds ago   Exited (0) 3 seconds ago             epic_kapitsa
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9  "/bin/bash"   3 days ago      Up 9 hours                           ecstatic_lewin
jetson@ubuntu:~$
```

5.8 How to open a container that is already in the [Exited] closed state

There are two scenarios: you still need to use the camera and you no longer need to use the camera.

5.8.1. Need to use the webcam

First of all, you need to check whether the port of AstraPro Plus camera has been changed according to the guideline in [5.4 Checking the Peripheral Connection] above.

1. If the port of Astra Pro camera has changed, it is not possible to access that container again.

(1) If there are some modifications in that container that need to be kept, you can refer to the following command to generate a new image that

```
从容器中提交一个镜像：
docker commit 容器id 要创建的目标镜像名:[标签名]
例如：docker commit 66c40ede8c68 yahboomtechnology/ros-foxy:1.1  # 标签名根据自己的情况递增

然后再运行这个新的镜像进入容器：参见本章节【5.2 到 5.6】步骤执行
Commit an image from a container:
docker commit container id name of target image to create:[tag name]
Example: docker commit 66c40ede8c68 yahboomtechnology/ros-foxy:1.1 # increment the tag name as you see fit

Then run this new image into the container: see steps [5.2 through 5.6] in this section.
```

(2) If there is no modification to be kept, then directly refer to steps [5.2 to 5.6] in this section to enter the container.

2. If there is no change in the AstraPro Plus camera port, then see [5.8.3, Entering the [Exited] closed container again].

5.8.2. Camera is not required.

Refer to [5.8.3, Re-entering a container in the [Exited] closed state].

5.8.3 Re-entering an [Exited] shutdown container

Open a terminal on the docker's host [i.e., the cart, either on VNC or on the cart's screen].

Note: Here must be in the cart VNC above or on the cart screen, not in the ssh remote access to the cart terminal (such as through the MobaXterm into the cart terminal), otherwise the container may not be able to display the GUI image, of course, how you do not need to display the GUI image, then you can.

1. First check the status of the container

```
docker ps -a
```

2. Enable GUI access

```
xhost +
```

3. Open the container [the ID of the container can be abbreviated, as long as it can uniquely identify the container that currently exists].

```
docker start 5b
```

4. Enter the container again

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

5. Open rviz to see if the GUI screen can be opened.

```
rviz2
```

6. The specific implementation is as follows:

```
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
Exited (0) 8 seconds ago             ecstatic_lewin
jetson@ubuntu:~$ xhost +
access control disabled, clients can connect from any host
jetson@ubuntu:~$ docker start 5b
5b
jetson@ubuntu:~$ docker exec -it 5b /bin/bash
--------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
--------------------------------------------------------
root@ubuntu:/# rviz2
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
[INFO] [1682298616.634096279] [rviz2]: Stereo is NOT SUPPORTED
[INFO] [1682298616.634576375] [rviz2]: OpenGl version: 3.1 (GLSL 1.4)
[INFO] [1682298617.959654036] [rviz2]: Stereo is NOT SUPPORTED
```