5. Enter the robot's docker container

The operating environment and software and hardware 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 (no version 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 a docker host?

```
The host is the server where we call the command to create the container using
the image. This refers to the main control on our car (jetson or Raspberry Pi,
etc.). The hosts mentioned below all refer to this.
```

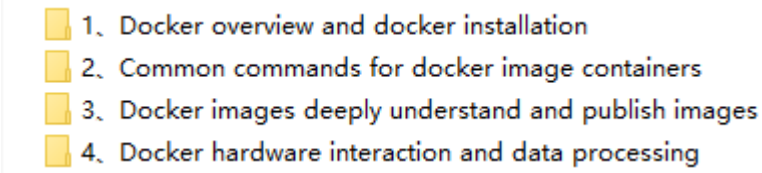
2. What is GUI?

```
GUI is the graphical user interface, which mainly refers to: the image window
displayed by opencv, rviz interface, rqt interface, etc.
```

3. What is the robot's docker container?

```
The robot here is the Rosmaster car, which is the Rosmaster car container that
has been configured with various development dependency environments.
```

4. Before operating the tutorial in this chapter, please make sure that you have mastered the knowledge of the following chapters, otherwise you may find it difficult to learn. If this happens, please review the following pre-knowledge content repeatedly. Once you master it, you will feel very relaxed. Come on, you are the best!



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

1. The docker image version used by the robot is also the image version used on the car. After the user burns the system image of the car and starts it, execute:

```
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
```

You will see multiple docker image versions. Please select the name [yahboomtechnology/ros-foxy]. The version with the highest tag is the latest image version of the robot. As queried here, use the [yahboomtechnology/ros-foxy:3.5.4] version, and ignore the image named [192.168.2.51:5000/ros-foxy].

2. Why can't we just put one docker image in the car system?

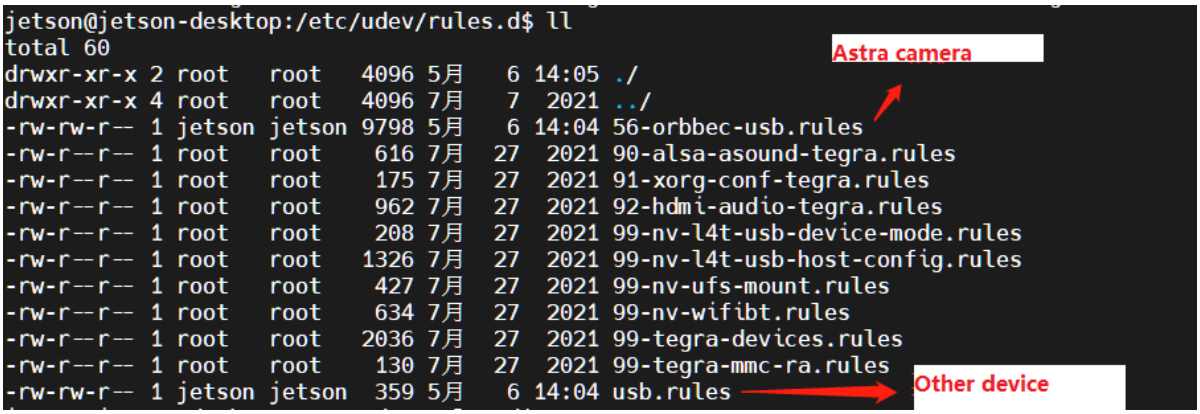
If you have read the tutorial in this chapter [07. Docker ------- 3. In-depth understanding of docker images and publishing images], you should know that docker images are a layered mechanism, that is, the image of a subsequent tag depends on the image of the previous tag. Mirror. Therefore, there may be multiple versions of docker images in the host machine, and the tags of these images will be updated incrementally.

In the future, we will update new courses and update functions by releasing new docker images.

5.3. Binding peripherals

- First make sure that the car has connected various peripherals and has done port binding on the peripherals. The port binding is processed on the docker host (car)
- Common peripherals include: serial port equipment, laser radar, RGBD camera, voice control module, joystick remote control, etc.
- **By default, the car has been bound to Astra camera, lidar and serial device**. If you need to bind other devices, please refer to the port binding tutorial.
- For the steps of port binding, please refer to the tutorial chapter [6. Linux operating system ---- 06. Binding device ID]

  Port binding has been configured in the host. If you need to modify it, you can check the content and modify it:

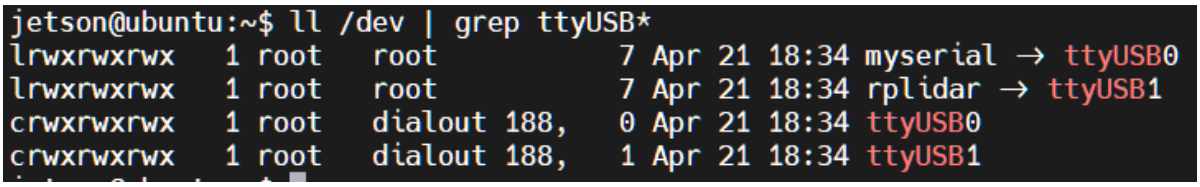


5.4. Check the peripheral connection status

This step is performed on the host machine:

1. This is to view the peripherals other than the camera. There is no voice control module connected here. If it is connected, the [myspeech] device will be displayed.

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



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
```

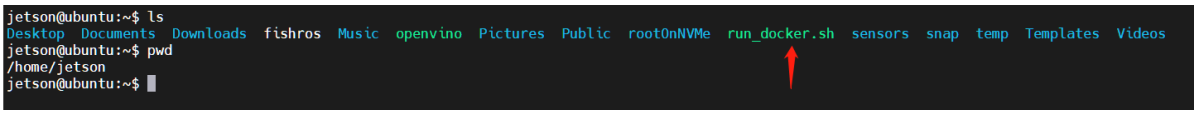
5.5. Edit script

Since the port number will often change after the AstraPro Plus camera is plugged in and unplugged, you need to re-edit the script to configure the port of the AstraPro Plus camera.

Edit the script to run docker. This step is performed on the host machine:

1. The script to run docker [run_docker.sh] is generally placed in the root directory of the car's owner directory. Here I am in the path below. If not, you can create the file yourself, and remember to give the script executable permissions after creation.

```
chmod +x run_docker.sh #Give the script executable permissions
```



The content of the [run_docker.sh] script is as follows:

Those without comments can be copied directly and modified as needed.

Note: When adding a host device to the container below, if the host is not connected to the device, you need to remove the corresponding addition operation before the container can be opened.

```
#!/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/video1 \
--device=/dev/myserial \
--device=/dev/rplidar \
--device=/dev/input \
-p 9090:9090 \
-p 8888:8888 \
yahboomtechnology/ros-foxy:3.5.4 /bin/bash
```

Annotated script description:

Note: When adding a host device to the container below, if the host is not connected to the device, you need to remove the corresponding addition operation before the container can be opened.

```
#!/bin/bash
xhost +                                         # xhost is used to support GUI
display in docker

docker run -it \                                # Interactively run the docker
image
--net=host \                                    # Container network is set to
host mode
--env="DISPLAY" \                               # Turn on the display GUI
interface
--env="QT_X11_NO_MITSHM=1" \                    # Use X11 port 1 for display
-v /tmp/.X11-unix:/tmp/.X11-unix \              # Map display service node
directory
-v /home/jetson/temp:/root/yahboomcar_ros2_ws/temp \ # As a directory for the
host and container to temporarily transfer files, you can use this directory if
you need to transfer files.
```

```
-v /home/jetson/rosboard:/root/rosboard \       # Directory used for app mapping
navigation
-v /home/jetson/maps:/root/maps \               # Directory used for app mapping
navigation
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \  # Add host device to the
container. This is the astrpro plus device port. If the car is not connected to
the camera, please remove this line.
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \  # Add host device to the
container. This is the astrpro plus device port. If the car is not connected to
the camera, please remove this line.
--device=/dev/astradepth \                      # Add host device to the
container. Here is the astrpro plus device port. If the car is not connected to
the camera, please remove this line.
--device=/dev/astrauvc \                        # Add host device to the
container. Here is the astrpro plus device port. If the car is not connected to
the camera, please remove this line.
--device=/dev/video0 \                          # Add host device to the
container. Here is the astrpro plus device port. If the car is not connected to
the camera, please remove this line.
--device=/dev/video1 \                          # Add host device to the
container. Here is the astrpro plus device port. If the car is not connected to
the camera, please remove this line.
--device=/dev/myserial \                        # Add host device to the
container. Here is the serial device port. If the car is not connected to the
serial port, please remove this line.
--device=/dev/rplidar \                         # Add host device to the
container. Here is the radar device port. If the car is not connected to the
radar, please remove this line.
--device=/dev/myspeech \                        # Add host device to the
container. Here is the voice control device port. If the car is not connected to
the voice control device, please remove this line.
--device=/dev/input \                           # Add host device to the
container. Here is the handle device port. If the car is not connected to the
handle, please remove this line.
-p 9090:9090 \ # Open port
-p 8888:8888 \
yahboomtechnology/ros-foxy:3.3.9 /bin/bash      # The name of the image to be
started, based on the modification queried in step 5.2; execute the /bin/bash
command in the container

#Note: When adding the host device to the container above, if the host is not
connected to the device, you need to remove the corresponding addition operation
before the container can be opened.
```

2. Modify the above script. These two lines are the port numbers of the AstraPro Plus camera. Since the port number will change after the camera is plugged in and out, you need to reconfigure the camera port.

```
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \ # Mount the storage volume to the
container and mount it to a directory in the container. What is mounted here is
the RGB and RGB of the camera. depth port
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \
```

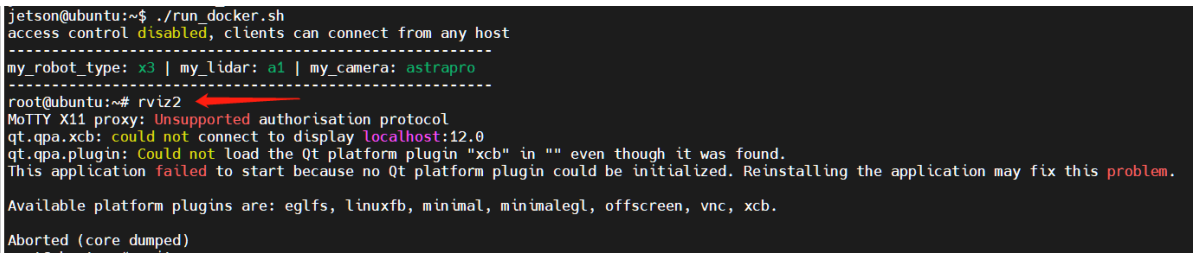
It is the camera port queried in step 5.4 2. This port may change after the camera is plugged in and out, so everyone's port is different and needs to be configured by yourself.

```
-v /dev/bus/usb/001/007:/dev/bus/usb/001/007 \ # Mount the storage volume to the container and mount it to a directory in the container. What is mounted here is the RGB and RGB of the camera. depth port
-v /dev/bus/usb/001/009:/dev/bus/usb/001/009 \
```

5.6. Execute script

After step 5.5 is completed, open the terminal on the docker host machine [i.e. the car, which can be on VNC or on the car screen]

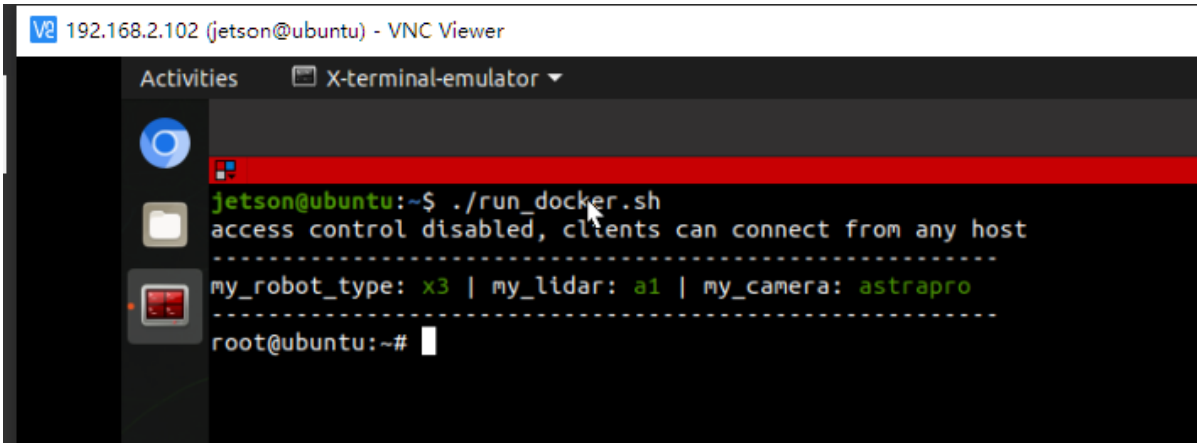
Note: This must be executed on the VNC of the car or on the car screen. It cannot be executed in the car terminal remotely entered through ssh (such as the car terminal entered through MobaXterm). Otherwise, the GUI image may not be displayed in the container, as shown below in MobaXterm After entering the car terminal and executing run_docker.sh to enter the container, rviz cannot be displayed.



Execute in the VNC interface of the car or on the car screen:

```
./run_docker.sh
```

You can correctly enter the container and display the GUI screen. You can execute the rviz2 command test again.



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

You need to add another parameter to the startup script:

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

Right now:

```
#!/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 \ # 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 enter the container and display the GUI screen.

5.7. Switch models, radars and cameras

Note: Since the ROSMASTER series robots are divided into multiple types of robots and multiple types of equipment, the factory system has been configured with routines for multiple types of equipment. However, since the product cannot be automatically identified, the machine type and radar model need to be manually set.

After entering the container: Make the following modifications according to the car model, radar type and camera type:

```
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
```

After the modification is completed, save and exit vim, and 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 modified car model, radar type and camera type

Robot project files are stored in the following directory:

```
/root/yahboomcar_ros2_ws
```

5.8. Multiple terminals 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.

Robot project files are stored in the following directory:

```
/root/yahboomcar_ros2_ws
```

3. Note:

(1) When executing the command in step 2, make sure the container is in the [UP] state

```
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, please refer to step 1.6 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] state

There are two situations: still need to use the camera and no longer need to use the camera

5.8.1. Need to use camera

First, you need to check whether the port of the AstraPro Plus camera has changed according to the guidance in the above step [5.3. Check the peripheral connection status].

1. If the port of the Astra Pro camera is changed, it will not be possible to enter the container again.

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

```
Submit an image from the container:
docker commit container id Target image name to be created: [label name]
For example: docker commit 66c40ede8c68 yahboomtechnology/ros-foxy:1.1 #The label
name is incremented according to your own situation

Then run this new image into the container: refer to the steps [5.2 to 5.5] in
this chapter to perform
```

(2) If there are no modifications that need to be retained, directly refer to the steps [5.2 to 5.5] in this chapter to enter the container.

2. If the port of the AstraPro Plus camera has not changed, then directly refer to the steps of [5.7.3, Entering the [Exited] Closed State Container Again].

5.8.2. No need to use camera

Directly refer to the steps of [5.7.3, Entering the [Exited] Closed State Container Again] to perform.

5.8.3. Containers that enter the [Exited] closed state again

Open the terminal on the docker host machine [that is, the car, which can be on VNC or on the car screen]

Note: This must be executed on the VNC of the car or on the car screen. It cannot be executed in the car terminal remotely entered through ssh (such as the car terminal entered through MobaXterm). Otherwise, the GUI image may not be displayed in the container. Of course, how can you There is no need to display the GUI image, that's fine.

1. First check the status of the container

```
  docker ps -a
```

2. Enable GUI access permissions

```
xhost+
```

3. Open the container [The ID of the container here can be abbreviated, as long as it can uniquely identify the currently existing container]

```
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
```