5.Enter docker container

5.Enter docker container

- 5.1. Related concepts
- 5.2. How to query the docker image version used by the robot
- 5.3. Binding peripherals
- 5.4. Check the peripheral connection status
- 5.5. Edit script
- 5.6. Execute script
- 5.7. Switch lidar model
- 5.8. Multiple terminals enter the same docker container
- 5.8. How to open a container that is already in the [Exited] state
  - 5.8.1. Need to use camera
  - 5.8.2. No need to use camera
  - 5.8.3. Containers that enter the [Exited] closed state again

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 Muto (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. 3. What is the docker container of the robot?

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

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 Muto. After the user burns the Muto system image and starts it, execute:

```
jetson@unbutu:~$ docker images
REPOSITORY                          TAG       IMAGE ID       CREATED        SIZE
192.168.2.51:5000/ros-foxy-muto     2.5.0     47a3378633af   7 hours ago    7.55GB
yahboomtechnology/ros-foxy-muto     2.5.0     47a3378633af   7 hours ago    7.55GB
```

You will see multiple docker image versions. Please select the name [yahboomtechnology/ros-foxy-muto]. The version with the highest tag is the latest image version of the robot. As queried here, use the [yahboomtechnology/ros-foxy-muto:2.5.0] version, and ignore the image named [192.168.2.51:5000/ros-foxy-muto].

Note: The tag queried above is version [2.5.0], which is the latest version at the time of writing this document.Since our docker image will be continuously updated in an incremental manner, the built-in version when purchased by users may be higher than version 2.5.0.Users only need to open the terminal, execute the [docker images] command, and then use the docker image with the highest tag to learn the course.

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

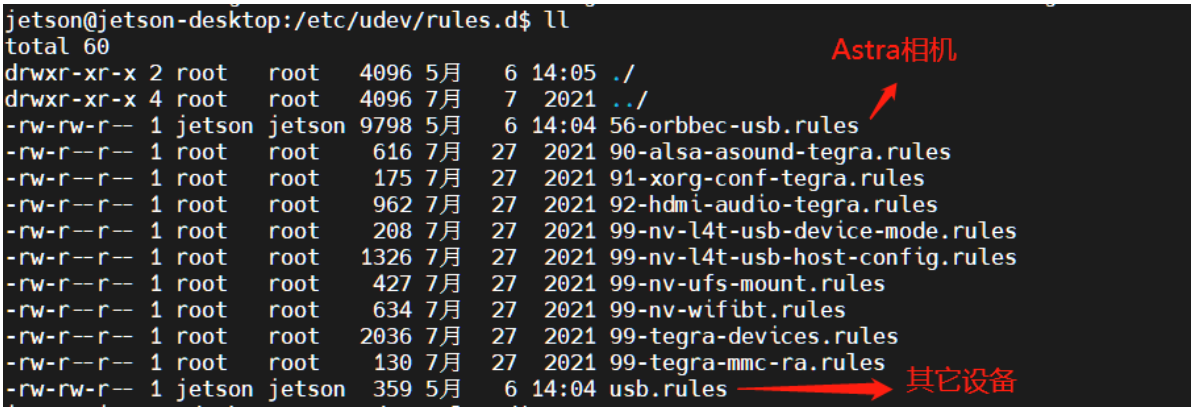
If you have read the tutorial in this chapter [12.Docker course ------ 3.Deeply understand and publish images], you should know that docker images are a layered mechanism.That is, the image of the next tag depends on the image of the previous tag.Therefore, there may be multiple versions of docker images 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 Muto has connected various peripherals and has done port binding on the peripherals. Port binding is processed on the docker host (Muto)
- Common peripherals include: serial port equipment, laser radar, RGBD camera, voice control module, joystick remote control, etc.
- **Muto has been bound to Astra cameras, lidar and serial devices by default**. If you need to bind other devices, please refer to the port binding tutorial.
- For the steps of port binding, please refer to the [Linux operating system ------ Bind device ID] tutorial chapter

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

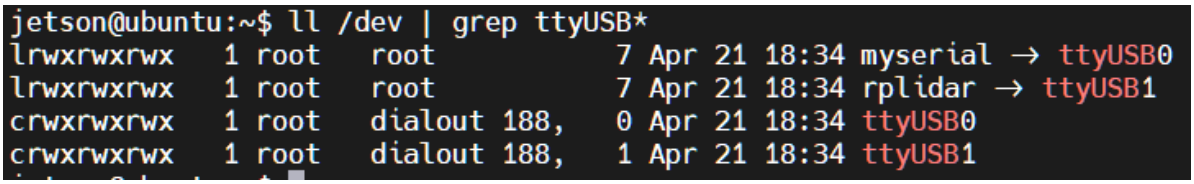


5.4. Check the peripheral connection status

This step is performed on the host machine:

1. Here is a view of peripherals other than cameras and voice modules:

```
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
```

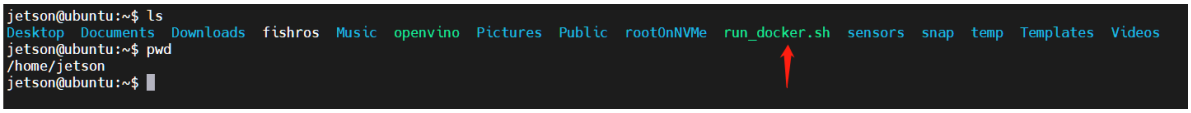
5.5. Edit script

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 Muto home directory. Here it is 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
```



【run_docker.sh】The content of the script is as follows:

Without comments, you can copy it directly and modify it 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 ~/temp:/root/yahboomcar_ros2_ws/temp \
-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-muto:2.5.0 /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 \                                        # Run docker image
interactively
--net=host \                                            # Container network is
set to host mode
--env="DISPLAY" \                                       # Turn on the display
GUI interface
--env="QT_X11_NO_MITSHM=1" \                            # Use port 1 of X11 for
display
-v /tmp/.X11-unix:/tmp/.X11-unix \                      # Mapping shows service
node directory
-v ~/temp:/root/yahboomcar_ros2_ws/temp \               # As a directory for
temporary file transfer between the host and container, you can use this
directory if you need to transfer files.
```

```
-v /dev/bus/usb/001/010:/dev/bus/usb/001/010 \          # Add the host device to
the container. Here is the astrpro plus device port. If the Muto is not connected
to the camera, please remove this line.
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \          # Add the host device to
the container. Here is the astrpro plus device port. If the Muto is not connected
to the camera, please remove this line.
--device=/dev/astradepth \                              # Add the host device to
the container. Here is the astrpro plus device port. If the Muto is not connected
to the camera, please remove this line.
--device=/dev/astrauvc \                                # Add the host device to
the container. Here is the astrpro plus device port. If the Muto is not connected
to the camera, please remove this line.
--device=/dev/video0 \                                  # Add the host device to
the container. Here is the astrpro plus device port. If the Muto is not connected
to the camera, please remove this line.
--device=/dev/myserial \                                # Add the host device to
the container. Here is the serial device port. If Muto is not connected to the
serial port, please remove this line.
--device=/dev/rplidar \                                 # Add the host device to
the container. Here is the radar device port. If Muto is not connected to the
radar, please remove this line.
--device=/dev/myspeech \                                #Add the host device to
the container. Here is the voice control device port. If Muto is not connected to
the voice control device, please remove this line.
--device=/dev/input \                                   #Add the host device to
the container. Here is the handle device port. If Muto is not connected to the
handle, please remove this line.
-p 9090:9090 \                                          # Open ports
-p 8888:8888 \
yahboomtechnology/ros-foxy-muto:2.5.0 /bin/bash             #The name of the
image to be started is based on the modification queried in step 5.2; execute the
/bin/bash command within 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 depth ports of the camera.
-v /dev/bus/usb/001/011:/dev/bus/usb/001/011 \
```

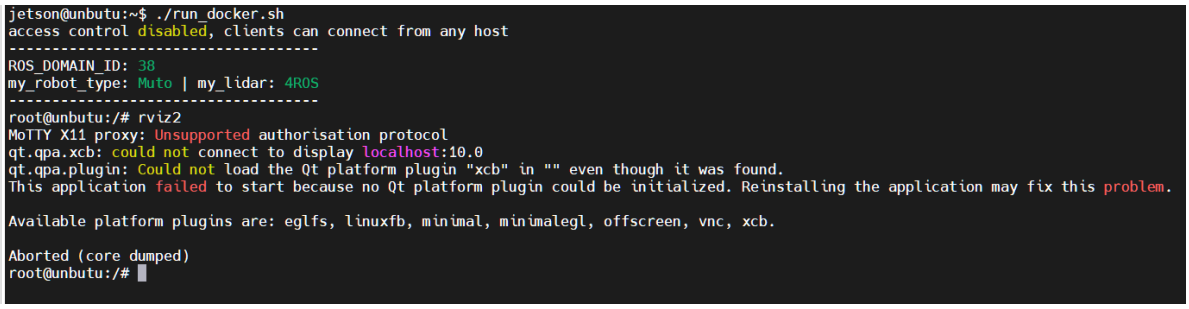
It is the camera port queried in step 5.4 2. This port may change after the camera is plugged in and out, so everyone's port is different and needs to be configured by yourself.

```
-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 depth ports of the camera.
-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 [i.e. Muto, which can be on VNC or on the Muto screen]

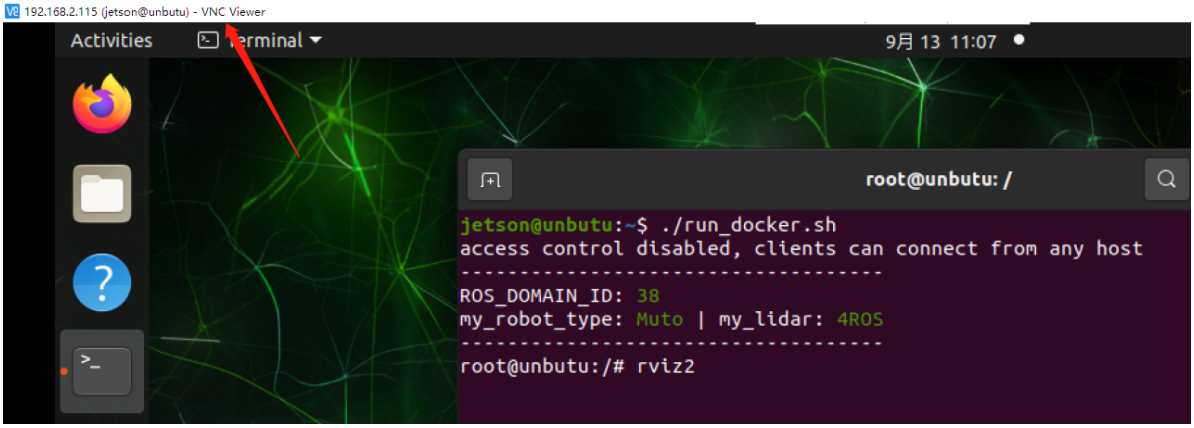
Note: This must be executed on Muto's VNC or on the Muto screen. It cannot be executed in the Muto terminal remotely entered through ssh (such as the Muto terminal entered through MobaXterm), otherwise the GUI image may not be displayed in the container.As follows, enter the Muto terminal in MobaXterm and execute run_docker.sh to enter the container. rviz cannot be displayed.



Execute in Muto's VNC interface or on the Muto 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 will be displayed: (generally likely to occur 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
```

That is:

```
#!/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 ~/temp:/root/yahboomcar_ros2_ws/temp \
-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/myspeech \
--device=/dev/input \
-p 9090:9090 \
-p 8888:8888 \
yahboomtechnology/ros-foxy-muto:2.5.0 /bin/bash
```

Then run the script again to enter the container and display the GUI screen.

5.7. Switch lidar model

Note: Since the Muto series robots are equipped with multiple radar devices, the factory system has been configured with routines for multiple devices. However, since the product cannot be automatically recognized, the lidar model needs to be manually set.

After entering the container: Make the following modifications according to the radar type:

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

```
# env
alias python=python3
export ROS_DOMAIN_ID=26

export ROBOT_TYPE=Muto
export RPLIDAR_TYPE=a1          # a1, 4ROS
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"
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@jetson-desktop:~# source .bashrc
-----------------------------------
ROS_DOMAIN_ID: 26
my_robot_type: Muto | my_lidar: a1
-----------------------------------
root@jetson-desktop:~#
```

You can see the current modified radar type.

Robot project files are stored in the following directory:

```
/root/yahboomcar_ros2_ws
```

5.8. Multiple terminals enter the same docker container

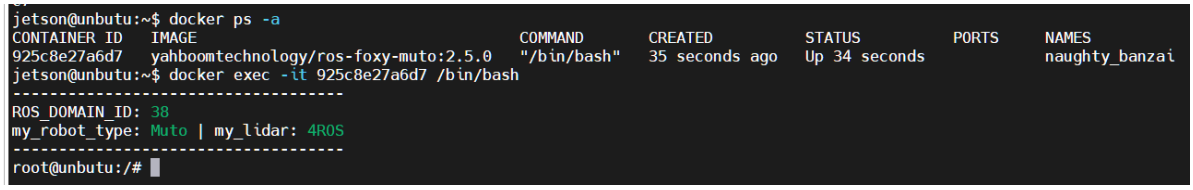
1. In the above steps, a docker container has been opened. You can open another terminal on the host (Muto) to view:

```
docker ps -a
```

```
jetson@unbutu:~$ docker ps -a
CONTAINER ID   IMAGE                                    COMMAND       CREATED          STATUS          PORTS     NAMES
925c8e27a6d7   yahboomtechnology/ros-foxy-muto:2.5.0   "/bin/bash"   35 seconds ago   Up 34 seconds             naughty_banzai
jetson@unbutu:~$
```

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

```
docker exec -it 925c8e27a6d7 /bin/bash
```



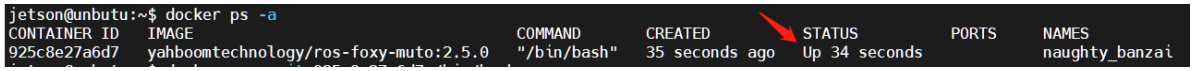
After successfully entering the container, you can open countless terminals to enter the container.

Robot project files are stored in the following directory:

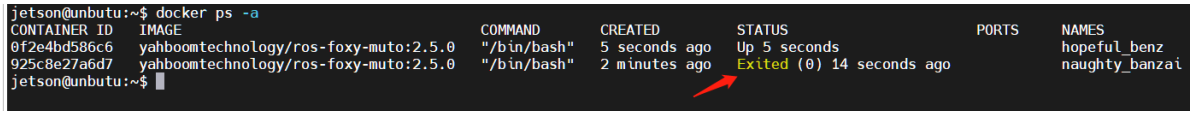
```
/root/yahboomcar_ros2_ws
```

3. Note:

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



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



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 step [5.4. Check the peripheral connection status] above.

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-muto:1.1 #The label name is incremented according to your own situation

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

(2) If there are no modifications and need to be retained, directly refer to the steps [5.2 to 5.6] 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 [5.8.3, Entering the [Exited] Closed State Container Again].

5.8.2. No need to use camera

Directly refer to the steps of [5.8.3, 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 [i.e. Muto, on VNC or on the Muto screen]

Note: This must be executed on Muto's VNC or on the Muto screen.It cannot be executed in a Muto terminal remotely entered through ssh (such as a Muto terminal entered through MobaXterm), otherwise the GUI image may not be displayed in the container. Of course, if you do not need to display the GUI image, then it is OK.

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 detailed execution is as follows:

```
jetson@unbutu:~$ docker ps -a
CONTAINER ID   IMAGE                                     COMMAND       CREATED
    STATUS                      PORTS     NAMES
925c8e27a6d7   yahboomtechnology/ros-foxy-muto:2.5.0   "/bin/bash"   2 minutes
ago   Exited (0) 14 seconds ago             naughty_banzai
jetson@ubuntu:~$ xhost +
access control disabled, clients can connect from any host
```

```
jetson@ubuntu:~$ docker start 92
92
jetson@unbutu:~$ docker exec -it 925c8e27a6d7 /bin/bash
-------------------------------------
ROS_DOMAIN_ID: 38
my_robot_type: Muto | my_lidar: 4ROS
-------------------------------------
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
```