Preparation before use

1. Install sdk driver

In the provided source code package, unzip YDLidar-SDK.zip and get the YDLidar-SDK folder. This folder is the SDK file of this radar, because using the ros function package of this radar requires installing the sdk and YDLidar-SDK files in advance. The folder contains the driver files of the radar. We open the terminal in this folder and enter,

```
mkdirbulid
cd build
cmake..
make -j4
sudo make install
```

If no errors are reported during operation, the driver is successfully installed.

2. Create a new workspace and compile function packages

- The first method is to decompress ydlidar_ws in the source code to your own root directory, and then compile it directly using catkin_make.

```
cd ydlidar_ws
catkin_make
```

After the compilation is passed, add the path of the workspace to .bashrc.

```
sudo gedit ~/.bashrc
```

Copy the following content to the end of the file,

```
source ~/ydlidar_ws/devel/setup.bash --extend
```

- The second method is to create a self-named workspace. Take the name oradar_ws as an example and enter it in the terminal.

```
mkdir oradar_ws
cd oradar_ws
mkdir src
cd src
catkin_init_workspace
```

Then copy the decompressed source code function package under ydlidar_ws/src to the oradar_ws/src directory, and then use catkin_make to compile in the oradar_ws directory.

```
cd oradar_ws
catkin_make
```

After the compilation is passed, add the path of the workspace to .bashrc.

```
sudo gedit ~/.bashrc
```

Copy the following content to the end of the file,

```
source ~/oradar_ws/devel/setup.bash --extend
```

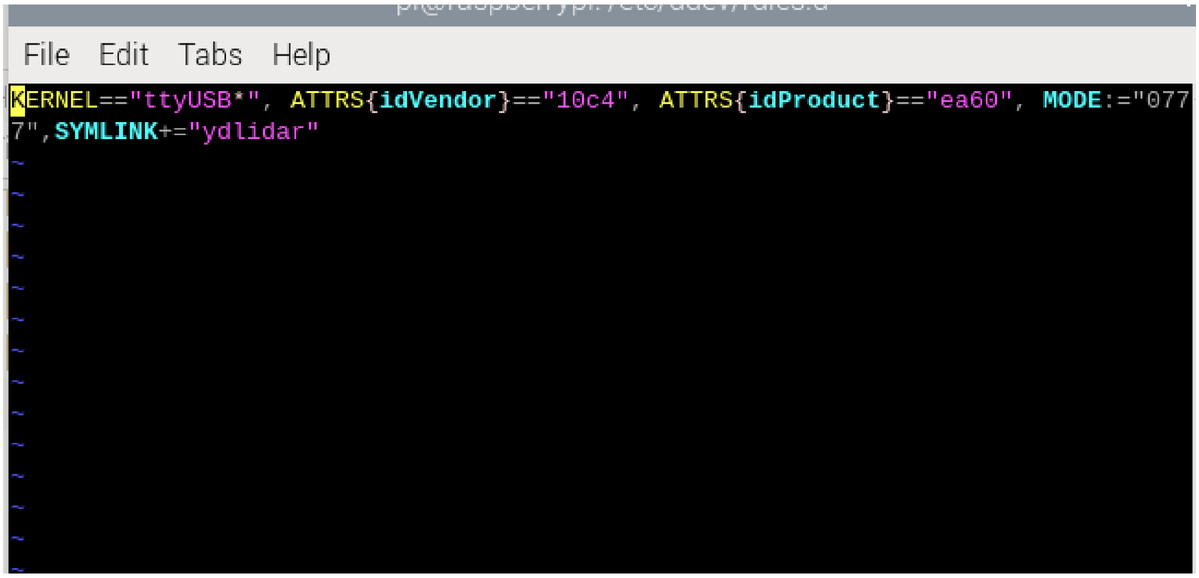
3. Bind radar port name

Open the terminal in the root directory of the Raspberry Pi and enter the following command,

```
cd /etc/udev/rules.d/
sudo vi usb.rules
```

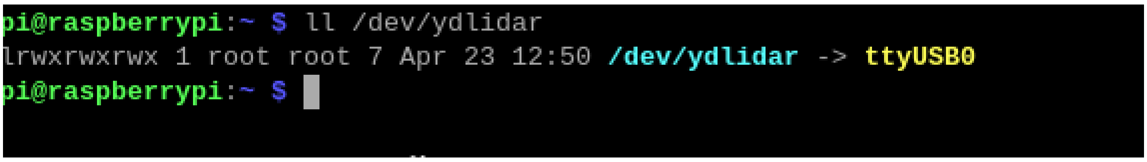
Then copy the following content into it,

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",
MODE:="0777", SYMLINK+="ydlidar"
```



```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", MODE:="077
7",SYMLINK+="ydlidar"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Then re-plug the radar serial port and enter ll /dev/ydlidar in the terminal.

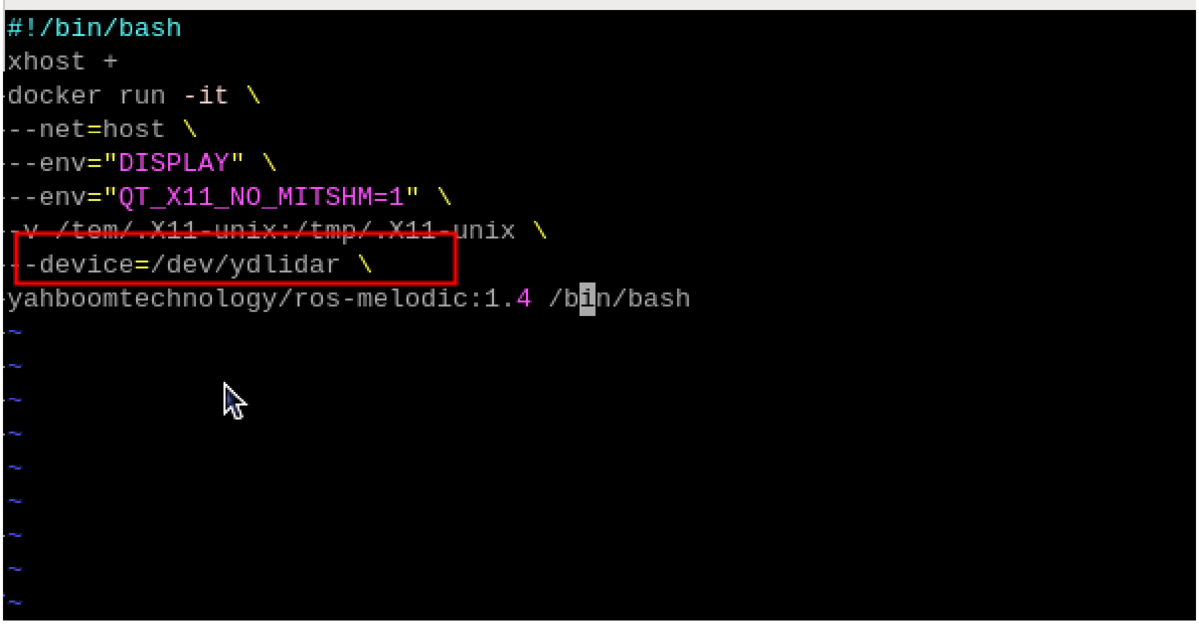


```
pi@raspberrypi:~ $ ll /dev/ydlidar
lrwxrwxrwx 1 root root 7 Apr 23 12:50 /dev/ydlidar -> ttyUSB0
pi@raspberrypi:~ $
```

The above content indicates that the binding is successful. The end is not necessarily 0 and changes according to the order in which the devices are inserted.

Then open our docker script and add it

```
vi run_ros1.sh
```

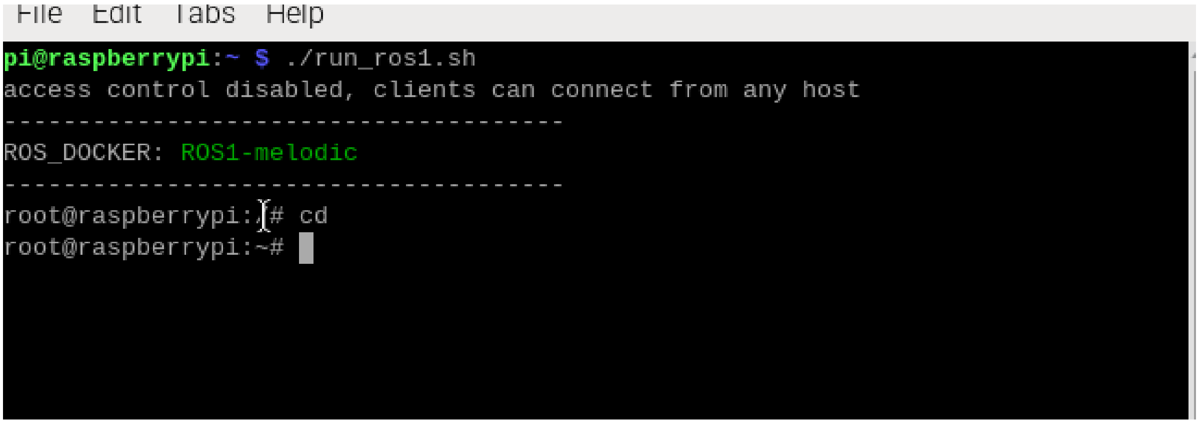


```
#!/bin/bash
xhost +
docker run -it \
--net=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
-v /tem/.X11-unix:/tmp/.X11-unix \
--device=/dev/ydlidar \
yahboomtechnology/ros-melodic:1.4 /bin/bash
~
~
~
~
~
~
~
~
~
```

4. Drive radar

First go into the docker we provide, open a terminal in the Raspberry Pi directory and enter,

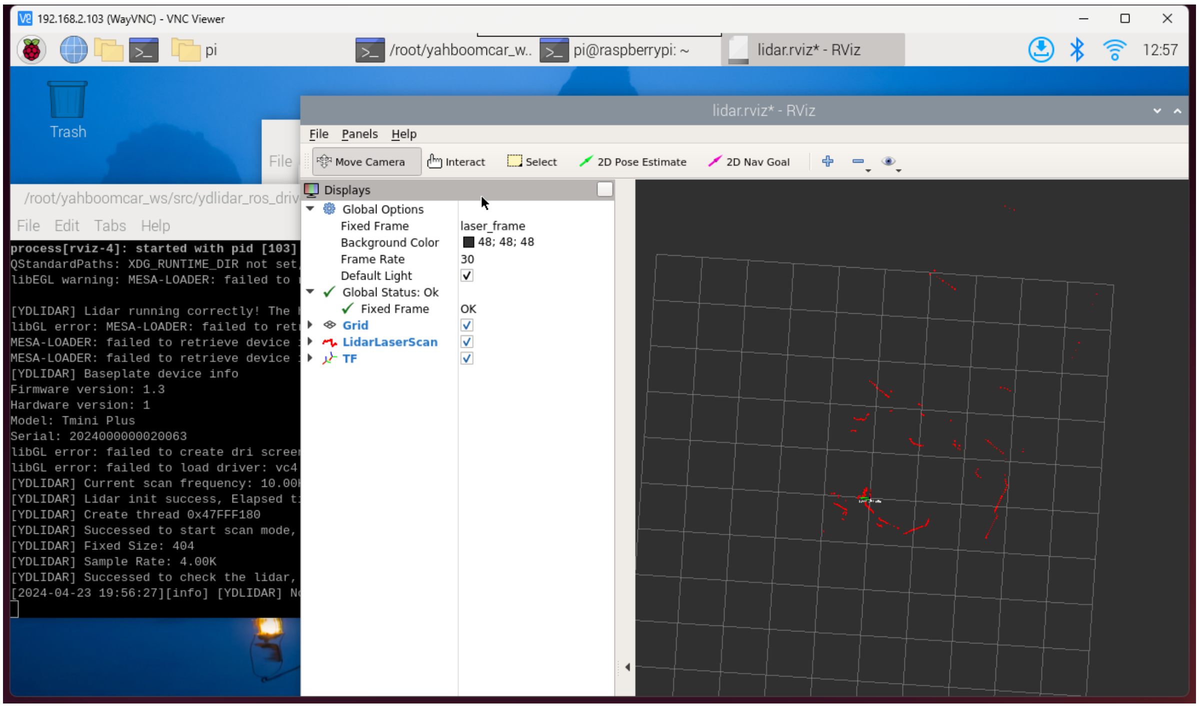
```
./run_ros1.sh
```



```
pi@raspberrypi:~ $ ./run_ros1.sh
access control disabled, clients can connect from any host
-------------------------------------
ROS_DOCKER: ROS1-melodic
-------------------------------------
root@raspberrypi:/# cd
root@raspberrypi:~# 
```

After that, enter the following statement to open the radar and display it in rviz,

```
roslaunch ydlidar_ros_driver lidar_view.launch
```



/root/yahboomcar_ws/src/ydlidar_ros_driv

```
process[rviz-4]: started with pid [103]
QStandardPaths: XDG_RUNTIME_DIR not set,
libEGL warning: MESA-LOADER: failed to 

[YDLIDAR] Lidar running correctly! The 
libGL error: MESA-LOADER: failed to ret
MESA-LOADER: failed to retrieve device 
MESA-LOADER: failed to retrieve device 
[YDLIDAR] Baseplate device info
Firmware version: 1.3
Hardware version: 1
Model: Tmini Plus
Serial: 2024000000020063
libGL error: failed to create dri screen
libGL error: failed to load driver: vc4
[YDLIDAR] Current scan frequency: 10.00
[YDLIDAR] Lidar init success, Elapsed t
[YDLIDAR] Create thread 0x47FFF180
[YDLIDAR] Successed to start scan mode,
[YDLIDAR] Fixed Size: 404
[YDLIDAR] Sample Rate: 4.00K
[YDLIDAR] Successed to check the lidar,
[2024-04-23 19:56:27][info] [YDLIDAR] N
```

- Global Options
  - Fixed Frame: laser_frame
  - Background Color: ■ 48; 48; 48
  - Frame Rate: 30
  - Default Light: ☑
- ✓ Global Status: Ok
  - ✓ Fixed Frame: OK
- Grid ☑
- LidarLaserScan ☑
- TF ☑

When the above screen appears, it means that all preparations have been completed.