6. Bind the device ID

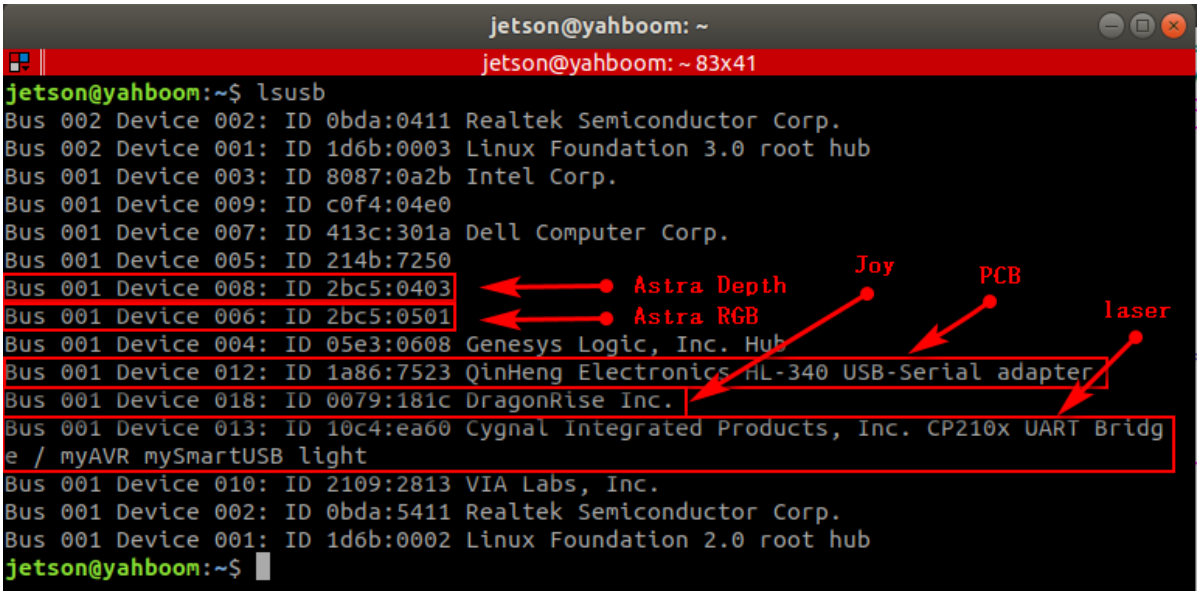
When the robot uses two or more USB serial devices, the corresponding relationship between the device name and the device is not fixed, but is allocated according to the order in which the devices are connected to the system. Inserting one device first and then another device can determine the relationship between the device and the device name, but it is very troublesome to plug and unplug the device every time the system starts. The serial port can be mapped to a fixed device name. No matter what the insertion order is, the device will be mapped to the new device name. We only need to use the new device name to read and write the device.

6.1. Device view command

Device ID view

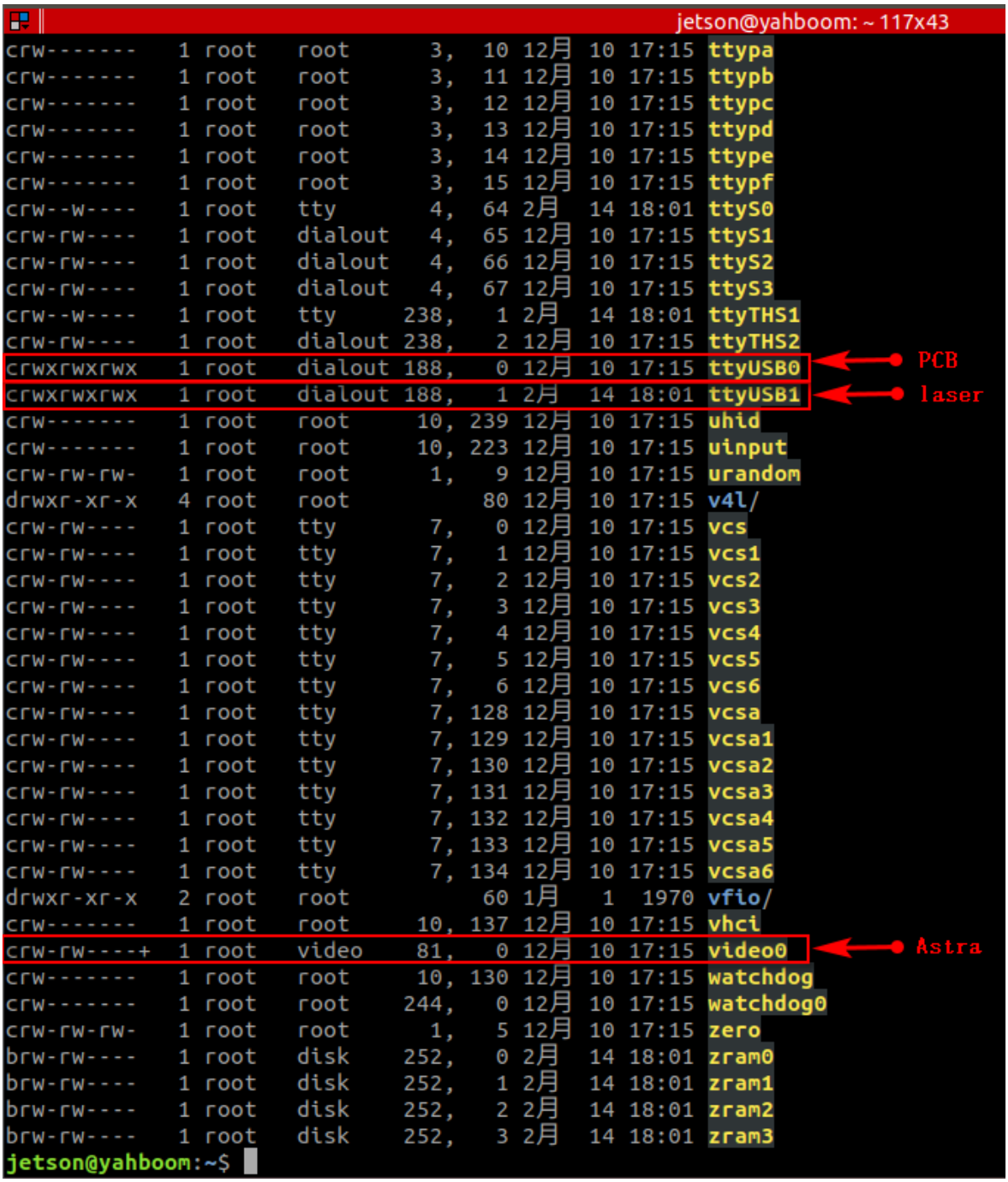
```
lsusb
```

As can be seen from the figure below, the ID number of each device, Astra has the official file for binding the device, the handle generally does not need to be bound, and the main binding is PCB and radar.



Device number view

```
ll /dev/
```



6.2. Establish port mapping relationship

6.2.1. Device binding

- **Astra binding**

The binding rules file for Astra camera is [56-orbbec-usb.rules], It is provided by Astra manufacturers and demonstrated here with AstraPro Plus.

Place the [56-orbbec-usb.rules] file in the /etc/udev/rules.d directory of the main control

```
/etc/udev/rules.d/56-orbbec-usb.rules
```

You can find the [56-orbbec-usb.rules] file, which is the Astra camera device binding file.

Run the following command to refresh the USB rules to bind the Astra camera

```
sudo udevadm control --reload-rules && sudo udevadm trigger
```

Check whether the binding is successful:

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

If the preceding information is displayed, the binding is successful.

- **PCB and Radar Bonding**

Go to the rules.d directory

```
cd  /etc/udev/rules.d/
```

Create a new [usb.rules] file and edit it

```
sudo vim usb.rules
```

write the following

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",
MODE:="0777", SYMLINK+="myserial"
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",
MODE:="0777", SYMLINK+="rplidar"
```

Exit to make the rules take effect

```
sudo chmod 777 usb.rules
sudo udevadm control --reload-rules && sudo udevadm trigger
```

Check whether the binding is successful:

```
jetson@ubuntu:~$ ll /dev | grep ttyUSB*
lrwxrwxrwx   1 root   root   7 5月  18 20:13 gps1 -> ttyUSB1
lrwxrwxrwx   1 root   root   7 5月  18 20:13 myserial -> ttyUSB0     #pcb binding
ttyUSB0 port
lrwxrwxrwx   1 root   root   7 5月  18 20:13 rplidar -> ttyUSB1      #radar
binding ttyUSB1 port
crwxrwxrwx   1 root   dialout 188,   0 5月  18 20:13 ttyUSB0
crwxrwxrwx   1 root   dialout 188,   1 5月  18 20:13 ttyUSB1
```

If the preceding information is displayed, the binding is successful.

6.2.2. Introduction to rule file syntax

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",
MODE:="0777", SYMLINK+="myserial"
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",
MODE:="0777", SYMLINK+="rplidar"
```

Parse

```
KERNEL          # The device name that matches the event
ATTR{filename}  # Match the sysfs attribute of the event device.
idVendor        # Vendor ID
idProduct       # product number
SYMLINK         # Generate symbolic links for device files under /dev/. Just
give this device an alias.
MODE            # Set permissions for the device.
```

From [6.1], it is easy to see that the device number of the PCB is [ttyUSB0], and the ID number is [1a86, 7523], which is fixed. 0, 1, 2, 3, 4, ...] are all bound to [myserial]; the same is true for radar device [ttyUSB1]; the same is true for other devices that need to be bound.

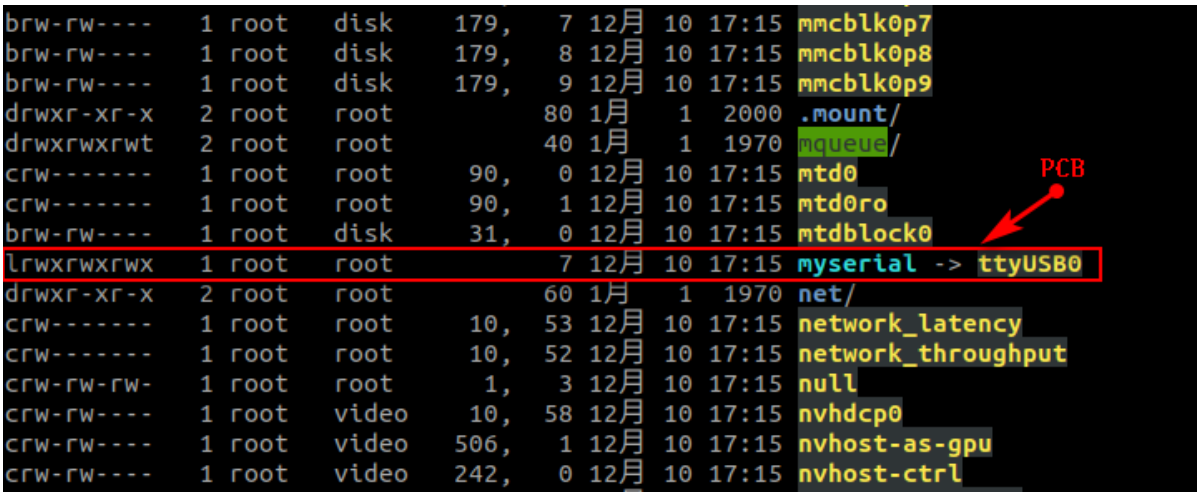
Note: When taking an alias, do not take some device names that already exist in the system, otherwise it will fail.

6.3. Verify View

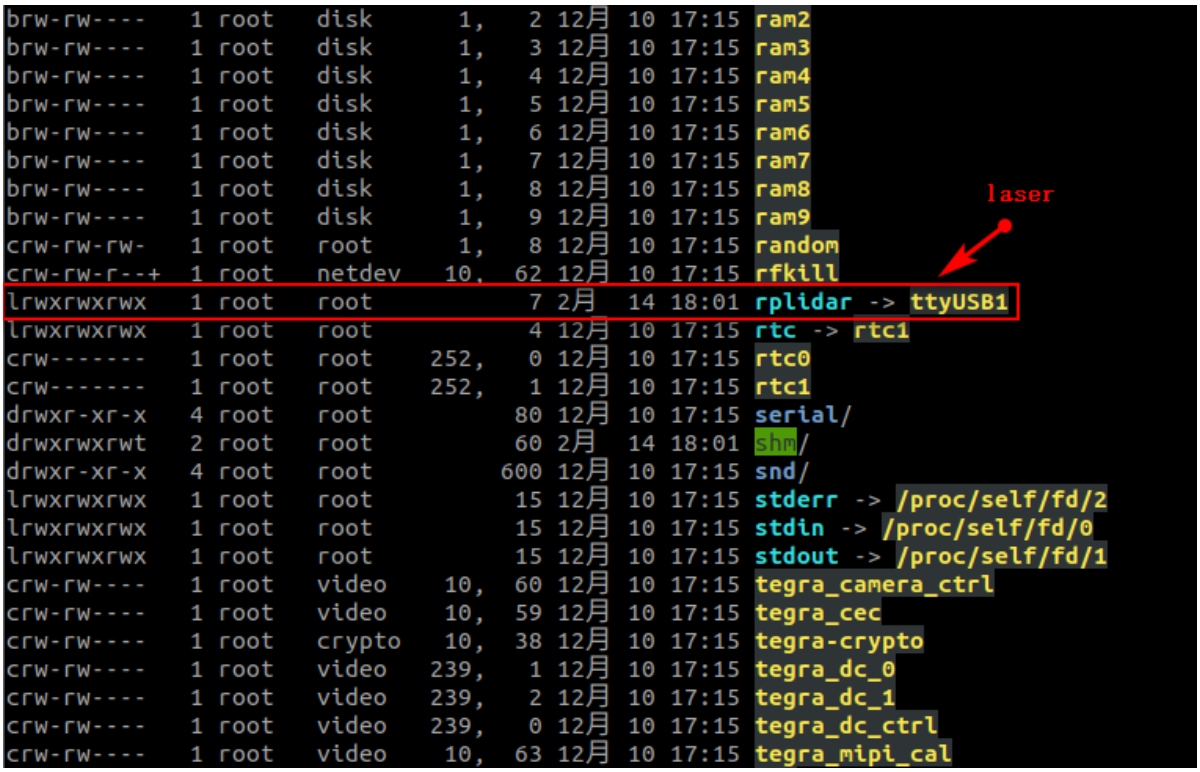
Device number view

```
ll /dev/
```

PCB



laser



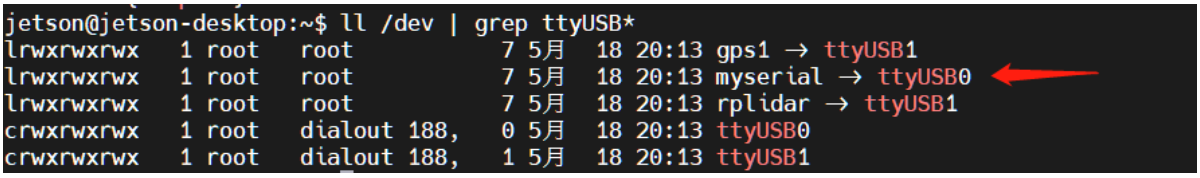
6.4. Binding the USB port

The above situations are all different ID numbers. If the ID numbers of the radar and the PCB are the same, or there are two or more PCBs (radars) with the same ID, the above binding will be confused.

Then, we need to bind the USB port. After binding, the **cannot be changed at will** , and each device **can only be linked to a fixed** USB port.

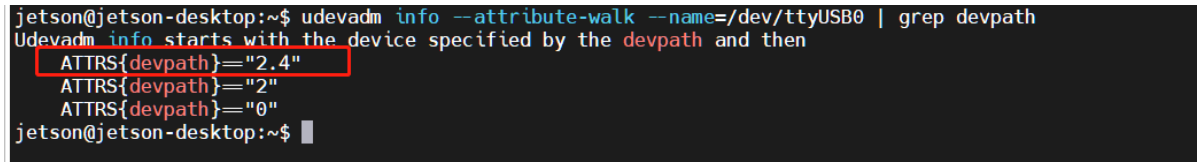
Binding method, take [ttyUSB0] as an example, check the port of the device at this time

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



The ttyUSB0 corresponding device is myserial

```
udevadm info --attribute-walk --name=/dev/ttyUSB0 | grep devpath
```



What we need is to modify the rules for myserial in the Rules file:

```
# before modification:
# KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",
MODE:="0777", SYMLINK+="myserial"
# after modification:
KERNEL=="ttyUSB*", ATTRS{devpath}=="2.4",  ATTRS{idVendor}=="1a86",
ATTRS{idProduct}=="7523", MODE:="0777", SYMLINK+="myserial"
```

Save and exit to make the rule take effect. Execute on the master:

```
sudo udevadm control --reload-rules && sudo udevadm trigger
```

Check whether the binding is successful:

```
jetson@ubuntu:~$ ll /dev | grep ttyUSB*
lrwxrwxrwx   1 root   root     7 5月  18 20:13 gps1 -> ttyUSB1
lrwxrwxrwx   1 root   root     7 5月  18 20:13 myserial -> ttyUSB0   #pcb binding
ttyUSB0 port
lrwxrwxrwx   1 root   root     7 5月  18 20:13 rplidar -> ttyUSB1    #radar
binding ttyUSB1 port
lrwxrwxrwx   1 root   root     7 5月  18 20:13 myspeech -> ttyUSB2   #Voice
control panel binding ttyUSB2 port
crwxrwxrwx   1 root   dialout 188,   0 5月  18 20:13 ttyUSB0
crwxrwxrwx   1 root   dialout 188,   1 5月  18 20:13 ttyUSB1
crwxrwxrwx   1 root   dialout 188,   1 5月  18 20:13 ttyUSB2
```

If the preceding information is displayed, the binding is successful.

If you do not understand the binding method above, you can also refer to the content of Chapter 14 of the course document <14.Voice control courses \ Voice control module port binding>