06. Bind device ID

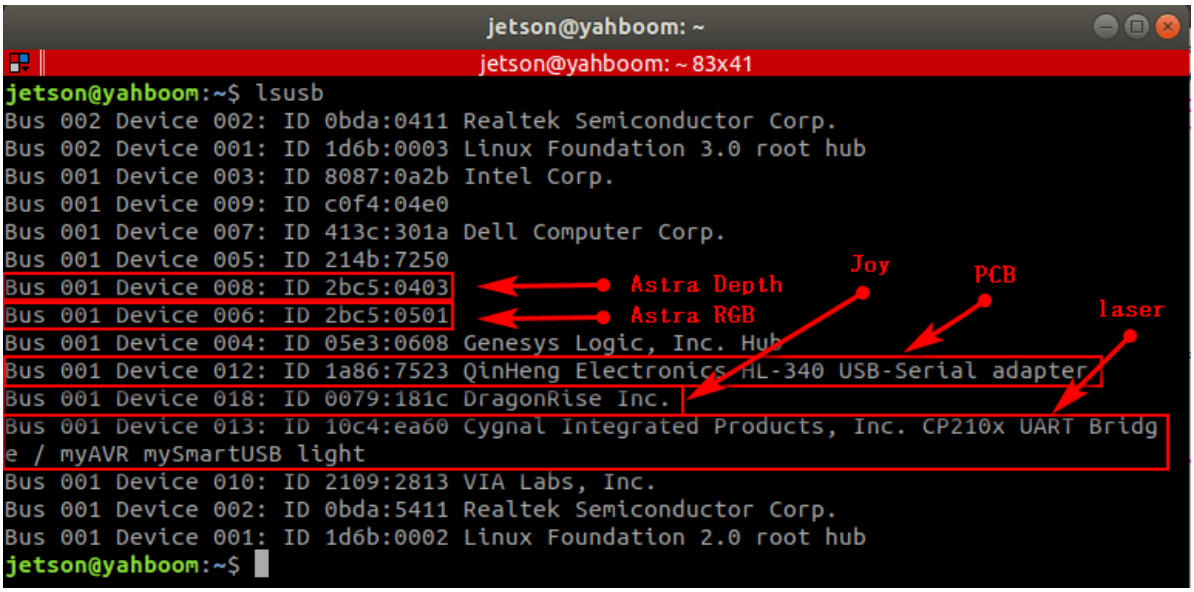
When the robot uses two or more USB serial devices, the correspondence between the device name and the device is not fixed, but is assigned in the order in which the devices are plugged into the system. Plugging in one device first and then another can determine the relationship between the device and the device name, but it is troublesome to plug and unplug the devices every time the system starts. You can map the serial port to a fixed device name, regardless of the order of insertion, the device will be mapped to the new device name, we just need to use the new device name to read and write to the device can be.

6.1 Device View Commands

Device ID View

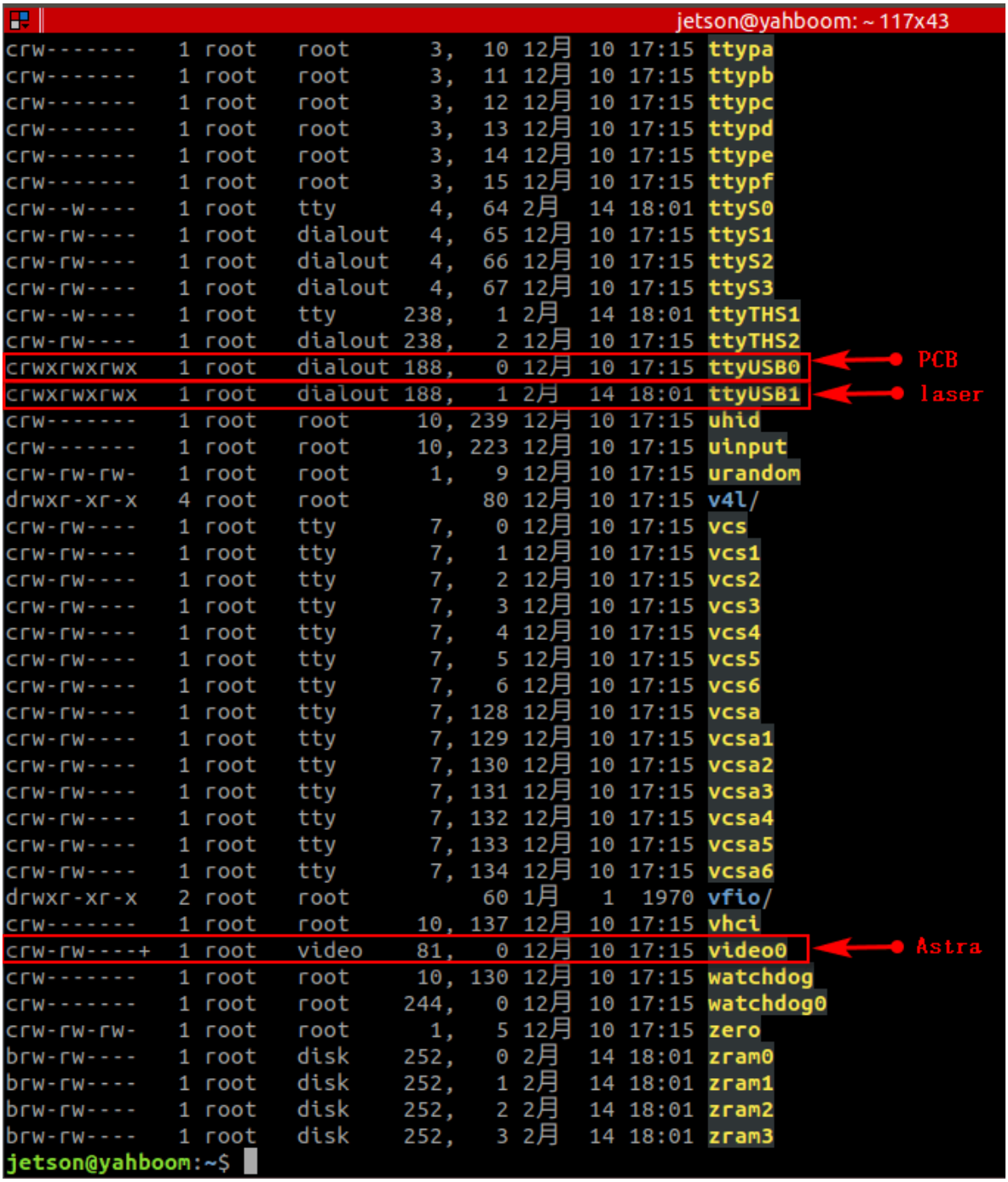
```
lsusb
```

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



Device Number View

```
ll /dev/
```



6.2 Device binding

6.2.1 Astra Camera Binding

Astra camera binding rules file is [56-orbbec-usb.rules], which is provided by Astra vendor, and is demonstrated here with AstraPro Plus.

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

I.e. the following location:

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

Execute the following command again to refresh the USB rules to bind the Astra camera in effect

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

Check to see if the binding was successful:

```
jetson@ubuntu:~$ ll /dev/astra*
lrwxrwxrwx 1 root root 15 May  5 17:42 /dev/astradepth -> bus/usb/001/007   #表示深度的端口
      # Ports that indicate depth
lrwxrwxrwx 1 root root 15 May  5 17:42 /dev/astrauvc -> bus/usb/001/009     #表示RGB的端口
    # denotes the port of RGB
```

Prints as above, indicating successful binding.

6.2.2 PCB and radar bindings

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"
```

Save and exit to make the rule effective and enforce it on the master:

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

Check to see if the binding was successful:

```
jetson@jetson-desktop:/etc/udev/rules.d$ ll /dev | grep ttyUSB*
lrwxrwxrwx   1 root   root           7 5月  18 20:13 gps1 -> ttyUSB1         #这个是系统自带的，不用管
#It comes with the system. Don't worry about it.
lrwxrwxrwx   1 root   root           7 5月  18 20:13 myserial -> ttyUSB0     #pcb绑定了ttyUSB0端口
#pcb bound to ttyUSB0 port
lrwxrwxrwx   1 root   root           7 5月  18 20:13 rplidar -> ttyUSB1      #雷达绑定了ttyUSB1端口
#Lidar bound to 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
```

Prints as above, indicating successful binding.

6.3 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          # 匹配事件的设备名
# Device name for matching events
ATTR{filename}  # 匹配事件设备的sysfs属性。
 # Match the sysfs attribute of the event device.
idVendor        # 生产商编号.
 # Manufacturer's number
idProduct       # 产品号# Product number
SYMLINK         # 为/dev/下的设备文件产生符号链接。就是给这个设备取一个别名。
# Generate symbolic links for device files under /dev/. That is, give this device
an alias.
MODE            # 为设备设定权限。
# Set permissions for the device.
```

By [6.1], see the PCB device number is [ttyUSB0] easy to jump, ID number is [1a86, 7523] fixed, [ttyUSB *] on behalf of whatever the device number becomes [ttyUSB] followed by [0, 1, 2, 3, 4, ...] are bound to [myserial]; lidar equipment [ttyUSB1] is the same; the need to bind other devices is also the same.

Note: When taking aliases, don't take some device names that already exist in the system, or it will fail. Note: When taking aliases, do not take some device names that already exist in the system, or they will fail.

6.4. Binding USB ports

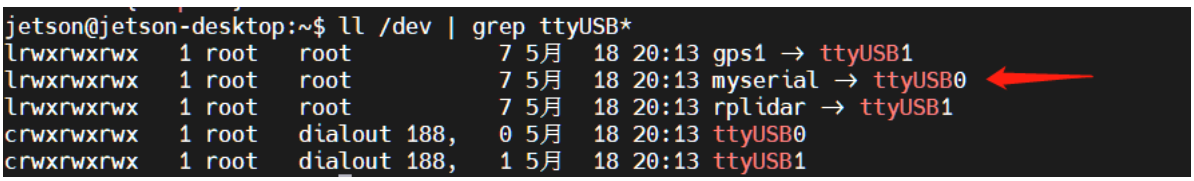
The above cases are all different ID numbers, if the lidar and PCB have the same ID number, or if there are two or more PCBs (lidars) with the same ID, the above bindings will be confused. E.g.**If we need to bind the voice control board when we have already bound the lidar and PCB, this situation will occur**

Then, we need to bind the USB port, after binding **can not be replaced at will** USB port, each device **can only be linked to a fixed** USB port.

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

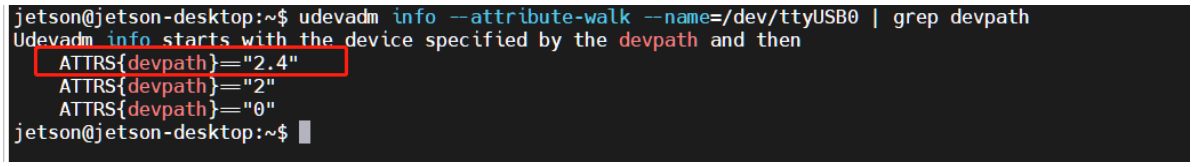
First view ttyUSB0 corresponding to the device:

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



The corresponding device for ttyUSB0 is: myserial

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



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

```
# 修改前:
# Before modification:
# KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",
MODE:="0777", SYMLINK+="myserial"
# 修改后:
# Modified:
KERNEL=="ttyUSB*", ATTRS{devpath}=="2.4",  ATTRS{idVendor}=="1a86",
ATTRS{idProduct}=="7523", MODE:="0777", SYMLINK+="myserial"
```

Save and exit to make the rule effective and enforce it on the master:

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

Check to see if the binding was successful:

```
jetson@jetson-desktop:/etc/udev/rules.d$ ll /dev | grep ttyUSB*
lrwxrwxrwx   1 root   root           7 5月  18 20:13 gps1 -> ttyUSB1         #这个
是系统自带的，不用管
 #It comes with the system. Don't worry about it.
lrwxrwxrwx   1 root   root           7 5月  18 20:13 myserial -> ttyUSB0     #pcb
绑定了ttyUSB0端口
 #pcb bound to ttyUSB0 port
lrwxrwxrwx   1 root   root           7 5月  18 20:13 rplidar -> ttyUSB1      #雷达
绑定了ttyUSB1端口
#Lidar is bound to the ttyUSB1 port
lrwxrwxrwx   1 root   root           7 5月  18 20:13 myspeech -> ttyUSB2     #语音
控制板绑定了ttyUSB2端口
#Voice control board bundled with 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
```

Printing as above indicates that the lidar, PCB and voice control board are all bound successfully.

If you do not understand any of the above binding methods, please refer to the contents of Chapter 14 of the course document **"14, Voice Control Series Courses \2, Voice Control Module Port Binding "** course.