

Bind device ID

Bind device ID

5.1. Device View

5.2. Device binding

5.2.1, Astra camera binding

5.2.2, PCB and radar binding

5.3, Syntax of rule file

When the robot uses two or more USB serial port devices, the correspondence between the device name and the device is not fixed, but is assigned in the order in which the devices are connected to the system. The relationship between the device and the device name can be determined by plugging one device first and then another device, but the device needs to be plugged in and out every time the system starts, which is very troublesome. The serial port can be mapped to a fixed device name. Regardless of the insertion order, 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.

Note: The ID of the supporting device has been bound to the system image. The following steps can be used as a reference for binding new devices.

5.1. Device View

After SSH connects to the car, enter in the terminal,

```
lsusb
```

Astra has an official file for binding the ID number of each device. The handle generally does not need to be bound. The main binding is PCB and radar.

```
root@ubuntu:/etc/udev/rules.d# lsusb
Bus 002 Device 002: ID 2109:0817 VIA Labs, Inc. USB3.0 Hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 007: ID 2109:8817 VIA Labs, Inc.
Bus 001 Device 025: ID 2bc5:0403 ← Astra depth
Bus 001 Device 024: ID 2bc5:0501 USB 2.0 Hub ← Astra RGB
Bus 001 Device 023: ID 05e3:0608 Genesys Logic, Inc. Hub
Bus 001 Device 027: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter ← PCB
Bus 001 Device 004: ID 2109:2817 VIA Labs, Inc.
Bus 001 Device 028: ID 045e:028e Microsoft Corp. Xbox360 Controller ← Joy
Bus 001 Device 026: ID 1a86:55d4 QinHeng Electronics USB 2.0 Hub ← Oradar
Bus 001 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Device number view

```
11 /dev/
```

```
crw--w---- 1 root tty      4,   7 May 29 16:10 tty7
crw--w---- 1 root tty      4,   8 May 29 16:10 tty8
crw--w---- 1 root tty      4,   9 May 29 16:10 tty9
crwxrwxrwx 1 root dialout 166,  0 May 29 16:10 ttyACM0 ← Oradar
crw----- 1 root root      5,   3 May 29 16:10 ttyprintk
crw--w---- 1 root tty     240,  0 May 29 16:10 ttyS0
crw-rw---- 1 root dialout 240,  1 May 29 16:10 ttyS1
crw-rw---- 1 root dialout 240,  3 May 29 16:10 ttyS3
crwxrwxrwx 1 root dialout 188,  0 May 29 16:10 ttyUSB0 ← PCB
crw----- 1 root root     236,  0 May 29 16:10 ubi0
crw----- 1 root root     480,  0 May 29 16:10 ubi10

crw-rw----+ 1 root video    81,  8 May 29 16:10 video3
crw-rw----+ 1 root video    81,  9 May 29 16:10 video4
crw-rw----+ 1 root video    81, 10 May 29 16:10 video5
crw-rw----+ 1 root video    81, 11 May 29 16:10 video6
crw-rw----+ 1 root video    81, 12 May 29 16:10 video7 ← Astra
crw-rw----+ 1 root video    81, 13 May 29 16:10 video8
crw-rw---- 1 root root     504,  0 May 29 16:10 vio_bind_info
```

5.2, Device binding

5.2.1, Astra camera binding

The binding rule file of Astra camera is [56-orbbec-usb.rules], provided by Astra manufacturer, and Astra Pro is used for demonstration here.

Put the [56-orbbec-usb.rules] file in the following directory of the car:

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

Execute the following command in the terminal to refresh the USB rules to bind the Astra camera.

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

After binding, enter the following command,

```
ll /dev/astra*
```

```
root@ubuntu:~# ll /dev/astra*
lrwxrwxrwx 1 root root 6 May 29 15:16 /dev/astra -> video8
lrwxrwxrwx 1 root root 15 May 29 15:16 /dev/astra_pro -> bus/usb/001/012
lrwxrwxrwx 1 root root 15 May 29 15:16 /dev/astrauvc -> bus/usb/001/011
```

The output is as above, indicating that the binding is successful.

5.2.2, PCB and radar binding

The car terminal executes the following command,

```
#Enter the rules.d directory
cd /etc/udev/rules.d/
#Create a rules file and edit it
sudo vim yahboomcar.rules
```

Write the following content (see the tutorial [Linux Basics] for vim command usage)

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",  
MODE:="0777", SYMLINK+="myserial"  
KERNEL=="ttyACM*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="55d4",  
MODE:="0777", SYMLINK+="oradar"
```

Save and exit, make the rules effective, execute:

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

After binding, enter the following command,

```
ll /dev/myserial #PCB  
ll /dev/oradar #Radar
```

```
root@ubuntu:/etc/udev/rules.d# ll /dev/myserial  
lrwxrwxrwx 1 root root 7 May 29 17:13 /dev/myserial -> ttyUSB0  
root@ubuntu:/etc/udev/rules.d# ll /dev/oradar  
lrwxrwxrwx 1 root root 7 May 29 17:13 /dev/oradar -> ttyACM0
```

The output is as above, indicating that the binding is successful.

5.3, Syntax of rule file

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",  
MODE:="0777", SYMLINK+="myserial"  
KERNEL=="ttyACM*", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="55d4",  
MODE:="0777", SYMLINK+="oradar"
```

Analysis

```
KERNEL # Match the device name of the event  
ATTR{filename} # Match the sysfs attribute of the event device.  
idVendor # Manufacturer number  
idProduct # Product number  
SYMLINK # Generate a symbolic link for the device file under /dev/. Just give  
this device an alias.  
MODE # Set permissions for the device.
```

From [6.1], we can see that the device number of PCB is [ttyUSB0] which is easy to change, and the ID number is [1a86, 7523] which is fixed. [ttyUSB*] means that no matter the device number becomes [ttyUSB] followed by [0, 1, 2, 3, 4, ...], it will be bound to [myserial]; the same is true for radar device [ttyACM0]; 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.