

# Ubuntu下绑定串口设备

## 一、背景

在日常的开发中，ubuntu主机可能会同时连接多个串口设备，每当重新上电或者是重新插拔之后，各个串口设备在ubuntu下分配的名字就可能产生变化，（比如原来的ttyUSB0,现在变为了ttyUSB1）为了解决这种分配名字的不确定性，这里将使用绑定串口号方法。想要实现这种绑定方法，那么就需要找到各个串口设备之间的区别。此处，依据的是串口芯片的ID来作为区分依据的。说明如下：

打开Ubuntu终端，插上所有USB串口设备，然后输入如下命令，然后类似反馈如下信息：

```
lee@lee-desktop: ~  
lee@lee-desktop:~$ lsusb  
Bus 001 Device 020: ID 2341:0042 Arduino SA Mega 2560 R3 (CDC ACM)  
Bus 001 Device 021: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter  
Bus 001 Device 022: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP210x UART Bridge / myAVR mySmartUSB light  
Bus 001 Device 007: ID 0424:7800 Standard Microsystems Corp.  
Bus 001 Device 003: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub  
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
lee@lee-desktop:~$
```

解释如下：

Bus 001 : 指明设备连接到哪条总线  
Device 020 : 表明这是连接到总线上的第二十台设备  
ID 1a86:7523: 设备的ID #不同串口芯片之间的唯一标识符  
QinHeng Electronics HL-340 USB-Serial adaptor :生产商名字和设备名，#此处为我的IMU模块的串口芯片名

## 二、根据串口ID绑定串口号

新打开终端，然后输入如下指令：

```
lee@lee-desktop:~$ cd /etc/udev/rules.d/  
lee@lee-desktop:/etc/udev/rules.d$ sudo vim yzrobot.rules
```

然后在新创建的文件中输入如下内容，并退出保存：

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="2a86", ATTRS{idProduct}=="7523",  
MODE=="0777", SYMLINK+="IMU_PORT"  
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",  
MODE=="0777", SYMLINK+="LIDAR_PORT"
```

和第一张图对应，其imu串口的绑定语句解释如下：

KERNEL=="ttyUSB*"	#表示的是ttyUSB类型的串口设备
ATTRS{idVendor}=="2a86"	#表示该usb串口设备的芯片ID号
ATTRS{idProduct}=="7523"	#表示该usb串口设备的厂商ID号
MODE=="0777"	#表示的给该串口赋予的权限
SYMLINK+="IMU_PORT"	#该串口的重命名为：IMU_PORT

### 三、重启服务和重新插拔usb串口设备

重启服务：

```
sudo service udev reload
sudo service udev restart
```

重新插拔串口设备，然后在终端中输入如下指令，并查看反馈：

```
ls -l /dev |grep ttyUSB
```

```
lee@lee-desktop:~$ ls -l /dev |grep ttyUSB
lrwxrwxrwx 1 root root      7 1月 20 11:40 IMU_PORT -> ttyUSB0
lrwxrwxrwx 1 root root      7 1月 20 11:40 LIDAR_PORT -> ttyUSB1
crwxrwxrwx 1 root dialout 188, 0 1月 20 11:40 ttyUSB0
crwxrwxrwx 1 root dialout 188, 1 1月 20 11:40 ttyUSB1
lee@lee-desktop:~$
```

可以看见，分别被重命名为了：IMU\_PORT、LIDAR\_PORT了。

至此，以后，在程序打开相应串口设备时，就用类似如下的方法：

```
mpu6050_ros_node.cpp X
src > mpu6050_ros > src > mpu6050_ros_node.cpp
12 unsigned char tag=0x00;
13 unsigned char Angle_initialization[3]={0xFF,0xAA,0x52}; //角度初始化校准指令，使陀螺角度归零
14 unsigned char Acceleration_calibration[3]={0xFF,0xAA,0x67}; //加速度计校准指令，校准加速度零偏
15 bool DecodeIMUData(double result[], unsigned char chrTemp[]);
16
17 int main(int argc, char** argv)
18 {
19     serial::Serial ser;
20     std::string port;
21     std::string imu_frame_id;
22     int baudrate;
23
24
25     ros::init(argc, argv, "mpu6050_ros_node");
26
27     ros::NodeHandle private_node_handle("~");
28     private_node_handle.param<std::string>("port", port, "/dev/IMU_PORT");
29     private_node_handle.param<int>("baudrate", baudrate, 115200);
30     private_node_handle.param<std::string>("imu_frame_id", imu_frame_id, "imu");
31
32
33     ros::NodeHandle nh;
34     ros::Publisher imu_pub = nh.advertise<sensor_msgs::Imu>(imu_frame_id, 100);
35     geometry_msgs::Quaternion Qua;
36
37     // publish imu message
38     sensor_msgs::Imu imu;
39
40     ros::Rate r(200); // 1000 hz
41
42     unsigned char chrBuffer[1000];
43     unsigned char chrTemp[1000];
```