

## 附录 ROS 的附件及使用 USB 设备： 创建 udev Rules

许多机器人都使用 USB 串行设备，如微型控制器、伺服电机控制器、激光扫描器或其他外设。当这些设备插入运行着 Ubuntu 的电脑上时，它们会被赋予形如 `/dev/ttyUSBn`、`/dev/ttyACMn` 或 `/dev/ttySn` 的名称，其中 `n` 是一个从 0 开始的整数，而名称中 USB、ACM 或 S 取决于硬件设备的具体情况（如 UART）。如果超过一个同类型设备插入，则这些设备会按顺序被编号，如 `/dev/ttyUSB0` 和 `/dev/ttyUSB1`。

问题在于 Ubuntu 分配名字给设备的时候遵循的是先到先得的原则，所以如果我在插入设备 A 后插入设备 B，那么第一个设备对应于 `/dev/ttyUSB0`，而第二个设备将对应于 `/dev/ttyUSB1`。但是，若先插入 B，则名称分配反过来。

这意味着将这些设备名称硬编码到配置文件或者启动文件中是不合适的。因为设备可能会在插入再被拔出之后又重新以不同的顺序插入。

幸运的是，Ubuntu 提供了一种方法来解决这个问题。可以将唯一的硬件序列号映射为我们自己选择的文件名。操作起来非常简单，只需几个步骤即可。

### 1 将你的账户加入 dialout 组

在开始其他操作之前，将你的账户加入 Linux dialout 组中是必要的。这将保证你拥有读写所有串口的权限。运行下面的指令：

```
$ sudo adduser your_login_name dialout
```

举例来说，如果你的账户名为 `robomeister`，那么用命令：

```
$ sudo adduser robomeister dialout
```

然后退出你的账户，再重新登入。或者直接重启电脑。

### 2 删除设备的序列号

假设有一块 Arduino 板是我们想插入机器人的 USB 口上的。先在插入设备之前，在终端中运行下面的指令：

```
$ tail -f /var/log/syslog | grep tty
```

起初，这条命令将不会产生任何输出。

现在插入设备（这里即 Arduino 板），然后在终端中观察输出。在笔者这里，短暂延迟之后看到了如下输出：

```
Oct 9 18:50:21 pi - robot - z935 kernel:[75067.120457] cdc_acm 2 - 1.3:1.0:
ttyACM0: USB ACM device
```

在输出中出现的 `ttyACM0` 告诉我们 Arduino 的设备名称被分配为 `/dev/ttyACM0`。还可以用下面的命令来检验：

```
$ ls -l /dev/ttyACM0
```

这将产生类似下面的输出：

```
crw-rw-- 1 root dialout 166, 0 Oct 9 18:50 /dev/ttyACM0
```

（注意时间戳应该与你插入设备的时间是一致的。）

现在我们知道了 Arduino 的设备名称为 `/dev/ttyACM0`，可以用功能来获取它的序列号：

```
$ udevadm info -q all -n /dev/ttyACM0 | grep -w ID_SERIAL_SHORT
```

在这里的输出是：

```
E: ID_SERIAL_SHORT = 74133353437351200150
```

等号后面的那串数字 74133353437351200150 便是序列号。

现在有了 Arduino 的序列号，让我们试试把它映射到一个设备的文件名。

### 3 UDEV 规则

序列号与设备文件名之间的映射可用一组 UDEV 规则来创建，我们将它保存在目录 `/etc/udev/rules.d` 下。首先要做的事情就是在这个目录下创建一个文件来保存这些规则。

从进入这个目录和查看目录下已有文件开始：

```
$ cd /etc/udev/rules.d
$ ls -l
```

输出的结果应该像下面这样：

```
-rw-r--r-- 1 root root 862 Nov 18 2013 70-persistent-cd.rules
-rw-r--r-- 1 root root 716 Nov 17 2013 70-persistent-net.rules
-rwxr-xr-x 1 root root 66 Jul 31 2014 99-android.rules
```

```
lrwxrwxrwx 1 root root 1157 Apr  5 2012 README
```

用你喜欢的文本编辑器创建一个文件名为 40-my-robot.rules 的文件，这个文件名必须是唯一的。注意因为要在系统目录下创建新文件，所以我们需要用 sudo：

```
$ sudo gedit 40-my-robot.rules
```

现在在打开的文本中添加下面的内容：

```
SUBSYSTEM = "tty", ENV { ID_SERIAL_SHORT } = "74133353437351200150",  
MODE = "0666", OWNER = "robomeister", GROUP = "robomeister",  
SYMLINK + = "arduino"
```

这里我们将 ID\_SERIAL\_SHORT 变量的值设置为前面我们找到的代表 Arduino 设备的序列号。MODE 一般都设置为 0666，OWNER 和 GROUP 应该设为你的账号。

SYMLINK 变量定义了我们想要的设备名称。在这里，设备名为 /dev/arduino，因为“/dev”是自动加在我们定义的名称之前的。

保存并退出文本编辑器。然后运行以下命令重新加载 UDEV 规则：

```
$ sudo service udev restart
```

这样便可以了，下面可以测试这个新的规则。

## 4 测试一条 UDEV 规则

如果设备还在插着，那么现在先拔掉它。等待几秒后再次插入。现在来看看是否能显示设备的新文件名。

```
$ ls -l /dev/arduino
```

```
在笔者这里，输出结果为：lrwxrwxrwx 1 root root 7 Oct 10 18:21 /dev/arduino  
-> ttyACM0
```

如你所见，在 /dev/arduino 和真正的设备文件 /dev/ttyACM0 之间被创建了符号链接。如果在插入之前有另外一个 ACM 设备插入的话，/dev/ttyACM0 就会被占用转而被链接到 /dev/ttyACM1。但不管是这二者的哪种情况，我们都在 ROS 配置文件中使用时即可，无需再担心底层的物理设备的名称。

## 5 在 ROS 配置文件中使用时设备名

在本书前面的内容中，我们运行过很多使用 ArbotiX Dynamixel 驱动和 USB2Dynamixel 控制器的例子。在那些例子中，我们将配置文件中的端口名称设置为

/dev/ttyUSB0。如果创建了 UDEV 规则，那么 USB2Dynamixel 的序列号就被映射到了符号链接 /dev/usb2dynamixel，所以我们可以配置文件中转而使用这一端口名称了。例如，在 rbx2\_dynamixel/config/arbotix 目录下的 pi\_robot\_head\_only.yaml 文件可以被写成下面的样子：

```
port: /dev/usb2dynamixel

baud: 1000000

rate: 100

syncwrite: True

syncread: False

readrate: 10

writerate: 10

joints: {
    headpanjoint: {id: 1, neutral: 512, minangle: -145, maxangle: 145},
    headtiltjoint: {id: 2, neutral: 512, minangle: -90, maxangle: 90}
}

controllers: {
    headcontroller: {onboard: False, actionname:
        headcontroller/follow_jointtrajectory, type: followcontroller, joints:
        [headpanjoint, headtiltjoint]}
```

注意我们仅仅是把 port 的值 /dev/ttyUSB0 替换成了 /dev/usb2dynamixel。现在操作系统对 USB2Dynamixel 控制器分配不同的 USB 号已经对我们没有影响了——此配置文件将不需要任何修改就可以正常工作。