

# 香橙派系统配置以及功能包启动

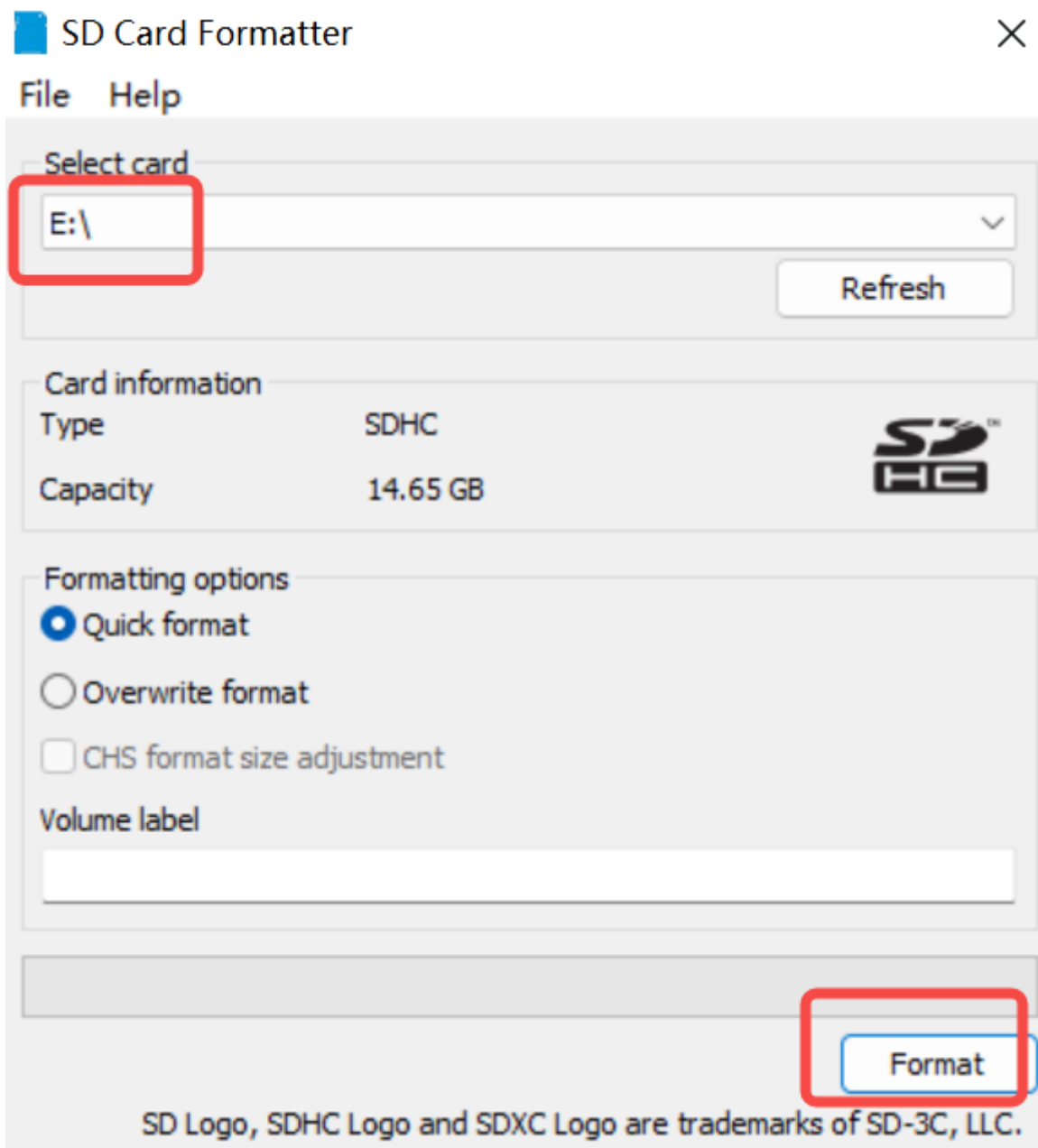
## 1.系统镜像下载

1. 这里我们需要安装ROS2的**humble**版本，需要**ubuntu 22.04**版本的系统。
2. 香橙派镜像下载：<http://www.orange-pi.cn/html/hardware/computerAndMicrocontrollers/service-and-support/Orange-Pi-3-LTS.html>
3. 点击Ubuntu镜像，选择jammy桌面版或者服务器版，如下图所示

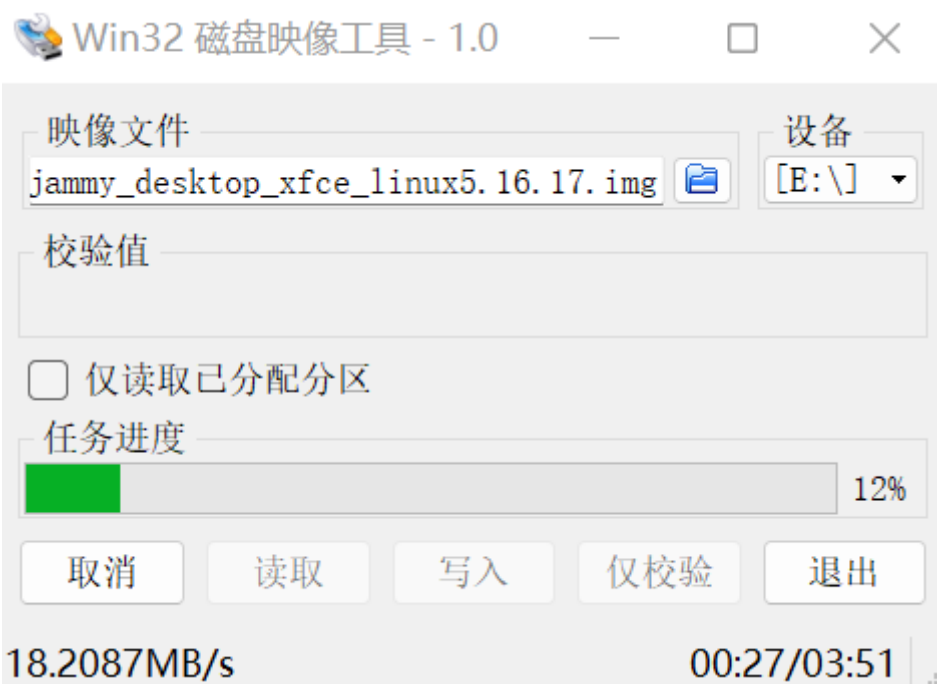
Ubuntu		订阅链接
2021-12-27 10:46 过期时间: 永久有效		
返回上一级   全部文件 > Ubuntu		
<input type="checkbox"/>	文件名	大小
<input type="checkbox"/>	backup	-
<input type="checkbox"/>	Orangepi3-lts_3.0.8_ubuntu_jammy_server_linux5.16.17.7z	358.9M
<input type="checkbox"/>	Orangepi3-lts_3.0.8_ubuntu_jammy_desktop_xfce_linux5.16.17.7z	827.2M
<input type="checkbox"/>	Orangepi3-lts_2.2.2_ubuntu_focal_server_linux5.10.75.7z	278.9M
<input type="checkbox"/>	Orangepi3-lts_2.2.2_ubuntu_focal_desktop_linux5.10.75.7z	715.8M
<input type="checkbox"/>	Orangepi3-lts_2.2.2_ubuntu_bionic_desktop_linux4.9.118.7z	660.4M

## 2.系统镜像安装

1. 插入读卡器，使用SD Card Formatter软件 格式化SD卡
2. 选好盘符，点击Format，点击弹窗确认，如下图所示：



3. 打开Win32DiskImager软件，然后选择 你需要安装的\*.img文件，点击写入，等待写入完成，如下图所示：



### 3.配网以及xshell连接香橙派

1. 香橙派连接wifi，获得IP地址
2. 客户端和香橙派连接同一个局域网
3. 打开xshell，新建一个链接
4. 用户名：orangeypi，密码：orangeypi

### 4.环境准备

# 0. 可通过配置选择关闭桌面版 （非必选）

```
sudo orangepi-config
```

# 1. ROS2安装

```
wget http://fishros.com/install -O fishros && . fishros # 1、1、2、3、1、2
```

# 2. 安装键盘控制功能包

```
sudo apt-get install ros-humble-teleop-twist-keyboard
```

# 3. 安装机器人描述功能包需要使用的相关功能包

```
sudo apt-get install ros-humble-xacro
```

```
sudo apt-get install ros-humble-joint-state-publisher
```

# 4. 卸载brltty程序，防止ttyUSB端口被占用

```
sudo apt-get purge brltty # 删除端口占用
```

# 5. 安装boost，串口通信程序需要，香橙派版默认没有安装

```
sudo apt-get install libboost-dev
```

# 6. reboot，到这里环境就准备ok了。

```
reboot
```

```
## -----  
-----
```

## 若出现包更新错误

E: 读错误 - read (5: 输入/输出错误)

E: 无法解析或打开软件包的列表或是状态文件。

# 英文是：

E: Read error - read (5 Input/output error),

E:The package lists or status file could not be parsed or opened.

刚开始是以为分区有错，所以用ubuntu的启动盘运行fsck -f /dev/sdx，检查根目录分区，但是问题依旧。

# 解决办法是：

```
sudo rm /var/lib/dpkg/status
```

```
sudo cp /var/lib/dpkg/status-old /var/lib/dpkg/status
```

```
sudo rm -rf /var/lib/apt/lists/*
```

```
sudo apt-get update
```

# 问题解决。

```
## -----
```

## 5.阿克曼小车驱动功能包相关准备

### 1.打开一个终端，新建ROS2工作空间文件夹

```
mkdir -p ~/ros2_ws/src
```

### 2.将分享的ROS2相应的功能包文件拷贝到src目录下

## 拷贝后文件结构如下

```
.
└─ src
    ├── mbot_bringup
    ├── mbot_description
    └─ sllidar_ros2-humble
```

### 3.打开终端，进入工作空间目录下，编译功能包

```
cd ~/ros2_ws
colcon build # 一起编译，若出现编译错误，可以单个单个编译功能包

colcon build --packages-select mbot_description
colcon build --packages-select mbot_bringup
colcon build --packages-select sllidar_ros2
```

编译成功后，终端将打印类似如下的信息

```
Starting >>> mbot_bringup
Starting >>> mbot_description
Starting >>> sllidar_ros2

Finished <<< mbot_bringup [2.16s]
          Finished <<< sllidar_ros2 [2.46s]

Finished <<< mbot_description [5.35s]

Summary: 3 packages finished [7.67s]
```

### 4.刷新环境

```
source ~/ros2_ws/install/setup.bash
```

### 5.设置工作空间刷新环境（一劳永逸）

```
echo "source ~/ros2_ws/install/setup.bash" >> ~/.bashrc
```

## 6.香橙派端口号配置

### 1.打开终端，在/etc/udev/rules.d/目录下，新建一个名为99-robot-usb.rules文件

```
cd /etc/udev/rules.d/  
sudo vi 99-robot-usb.rules
```

## 2.将如下内容复制到文件中

```
KERNELS=="1-1", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",  
SYMLINK+="mbot",MODE=="0777"  
KERNELS=="6-1", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",  
SYMLINK+="lidar",MODE=="0777"  
KERNELS=="5-1", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",  
SYMLINK+="camera",MODE=="0777"
```

**一定要注意：**否则运行程序时，会提示找不到端口

单片机串口的端口：命名为mbot

激光雷达驱动的端口：命名为lidar

## 3.然后根据你的实际情况更改，KERNELS、idVendor、idProduct对应的值。

这里主要参考我的这篇文章：linux系统下串口设备和串口号绑定

链接如下：[https://blog.csdn.net/zhao\\_ke\\_xue/article/details/108700080?spm=1001.2014.3001.5501](https://blog.csdn.net/zhao_ke_xue/article/details/108700080?spm=1001.2014.3001.5501)

详细的细节如下：

### 1.问题由来

在ROS小车的开发过程中，需要使用ROS通信串口模块、激光雷达的串口模块、IMU的串口模块等，为了防止每次开机这些设备的串口号发生变动，所以需要对接口号与串口设备绑定；为了更好的识别串口，我们也可以对接口号进行设置静态名称；为了更好的使用串口，我们也可以对接口进行权限设置。

**一次设置，终身受益。**

### 2.原理

在linux系统下，虽然每一个固定的端口可能由于外接设备启动的时间差异分配的ttyUSB\*不同，但是对于每一个固定的端口，都有固定的 KERNELS、idProduct、idVendor，并且每一个端口的KERNELS都不相同。

所以，我们只要固定串口设备的安装位置，还是可以从端口物理性质的不同解决这个问题。

### 3.方法

#### 第一步：如何查询每一个端口的KERNELS、idProduct、idVendor？

这里需要使用下面这条命令

```
udevadm info -a -p $(udevadm info -q path -n /dev/ttyUSB0)
```

并且**当每次独立插入一个串口设备时**，串口号总为/dev/ttyUSB0，若不放心，可以使用下面命令查询

```
ls /dev | grep ttyUSB
```

使用第一条命令，窗口将打印一堆设备信息，我们只需要KERNELS、idProduct、idVendor，我们向下滑找到类似下图的部分找到该端口的信息。

```

looking at parent device '/devices/pci0000:00/0000:00:15.0/0000:03:00.0/usb3/3-2':
  KERNELS=="3-2"
  SUBSYSTEMS=="usb"
  DRIVERS=="usb"
  ATTRS{authorized}=="1"
  ATTRS{avoid_reset_quirk}=="0"
  ATTRS{bConfigurationValue}=="1"
  ATTRS{bDeviceClass}=="ff"
  ATTRS{bDeviceProtocol}=="00"
  ATTRS{bDeviceSubClass}=="00"
  ATTRS{bMaxPacketSize0}=="8"
  ATTRS{bMaxPower}=="98mA"
  ATTRS{bNumConfigurations}=="1"
  ATTRS{bNumInterfaces}==" 1"
  ATTRS{bcdDevice}=="0263"
  ATTRS{bmAttributes}=="80"
  ATTRS{busnum}=="3"
  ATTRS{configuration}==" "
  ATTRS{devnum}=="5"
  ATTRS{devpath}=="2"
  ATTRS{idProduct}=="7523"
  ATTRS{idVendor}=="1a86"
  ATTRS{ltm_capable}=="no"
  ATTRS{maxchild}=="0"
  ATTRS{product}=="USB2.0-Serial"
  ATTRS{quirks}=="0x0"
  ATTRS{removable}=="unknown"
  ATTRS{speed}=="12"
  ATTRS{urbnum}=="15"
  ATTRS{version}==" 1.10"

```

注意：查询端口信息时，请单独插入串口设备。

## 第二步：在/etc/udev/rules.d/目录下创建自己的99-\*.rules文件

比如这里创建99-robot-usb.rules文件，使用如下命令

```
sudo vim /etc/udev/rules.d/99-robot-usb.rules
```

填入如下内容，保存

```
KERNELS=="3-2", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523",
SYMLINK+="robot",MODE=="0777"
```

对上述的解释

“==”：比较键、值，若等于，则该条件满足；

“+=”：为一个表示多个条目的键赋值。

“:="：对一个键赋值，并拒绝之后所有对该键的改动。目的是防止后面的规则文件对该键赋值。

## 第三步：重启并查看配置是否生效

重启

```
sudo reboot
```

等待开机，查看端口

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

中断打印出如下信息，设置成功

```
lrwxrwxrwx  1 root root          7 Sep 20 22:30 robot -> ttyUSB0
crwxrwxrwx  1 root dialout 188,  0 Sep 20 22:30 ttyUSB0
```

之后我们再使用该物理位置的端口时，它的串口号就固定为 `/dev/robot`，并且具有可读可写可执行的权限。

#### 注意事项：

以上只是对一个端口设置的流程，大家在多数情况下都是多个设备。聪明的你，一定知道怎么举一反三。

每次独立插入一个串口设备，获得它的KERNELS、idProduct、idVendor值，记录下来；在`/etc/udev/rules.d/`目录下创建自己的99-\*.rules文件，一次性填入这些设备的信息以及你需要的映射的名字；然后重启即可。

## 7.启动功能包测试

此时，需要具备的条件

- 1.线路都链接正确且正常（单片机->香橙派）（激光雷达->香橙派）
- 2.单片机正常且烧录好程序

这里我们准备了两个launch文件，一个含激光雷达，一个不含激光雷达

```
mbot_with_lidar_launch.py
mbot_launch.py
```

以运行含激光雷达的launch文件举例，在终端输入如下指令：

```
ros2 launch mbot_bringup mbot_with_lidar_launch.py
```

终端打印如下就是正常启动了

```
[mbot_bringup-1] [INFO] [1666446425.232926736] [mbot_bringup]: y_-0.006942
[mbot_bringup-1]
[mbot_bringup-1] [INFO] [1666446425.233090570] [mbot_bringup]: th_:338.204911
[mbot_bringup-1]
[mbot_bringup-1] [INFO] [1666446425.253658500] [mbot_bringup]: dt:0.021006
[mbot_bringup-1]
[mbot_bringup-1] [INFO] [1666446425.253865001] [mbot_bringup]: x_-0.004255
[mbot_bringup-1]
[mbot_bringup-1] [INFO] [1666446425.253914877] [mbot_bringup]: y_-0.006942
```