XQ-4 Pro

ROS is not difficult any more!

XQ-mini

用户手册

（综合版）

BWBOT

蓝鲸智能机器人（深圳）有限公司

目錄

引言 1.1
一. 开始使用 1.2

二. 小强ROS机器人教程

1. 基础操作介绍 2.1
2. 蓝鲸智能开源软件仓库的使用和ROS开机启动任务的配置 2.2
3. 在rviz中显示小强机器人模型 2.3
4. 惯性导航自主移动测试 2.4
5. 小强遥控图传app安卓版 2.5
6. 小强遥控图传windows客户端 2.6
7. 使用ps3手柄控制小强 2.7
8. kinect1代 ROS驱动测试与安装 2.8
9. 使用rostopic控制kinect的俯仰角度 2.9
10. 使用kinect进行自主移动避障 2.10
11. kinect跟随包turtlebot_follower 2.11
12. ROS显示kinect2代的点云 2.12
13. rplidar二代激光雷达的使用和利用udev给小车增加串口设备 2.13
14. 在gmapping下使用激光雷达rplidar a2进行建图 2.14
15. AMCL导航测试 2.15
16. 大范围激光雷达slam与实时回路闭合测试 2.16
17. 利用ORB_SLAM2建立环境三维模型 2.17
18. 利用DSO_SLAM建立环境三维模型 2.18
19. NLlinepatrol_planner的简单使用 2.19
21. 获取usb摄像头30fps的1080p图像流及120fps的VGA分辨率图像流 2.20
22. 操作6自由度机械臂 2.21
23. ROS入门手册 2.22
24. 在小强上使用语音识别和语音合成功能 2.23
25. 自动充电功能包的使用和实现原理 2.24
26. 使用Intel RealSense D400系列深度摄像头进行自主移动避障 2.25
27. 安装使用zed相机cpu版本ros驱动 2.26
28. 使用depthimage_to_laserscan包将kinect深度图像转换成2d激光雷达话题 2.27

29. 使用ps4手柄控制小车移动 2.28

三. 维护

充电 3.1
车轮松动打滑 3.2
小强底盘固件的自动更新升级方法 3.3
升级底盘ros驱动包xqserial_server 3.4
重新校准小车底盘IMU 3.5
小强系统镜像 3.6
如何恢复小强的软件包 3.7
如何在重装系统后恢复xiaoqiang设置 3.8
如何获取伽利略证书 3.9

四. 其他

Ubuntu设置静态IP 4.1
视觉导航路径编辑器使用教程 4.2
小强的远程协助功能 4.3
小强ROS机器人障碍物识别演示 4.4
视觉导航在履带车中的运用 4.5
Google激光雷达slam算法Cartographer的安装及bag包demo测试 4.6
原装和国产ps3手柄ros驱动程序 4.7
升级软件包以支持小强图传遥控app 4.8
小强自检程序 4.9

附件

1. 小车系统框架图 5.1
2. 电气布线图 5.2
3. 小强电脑与stm32底层通讯协议 5.3

- 引言

引言

这是小强用户手册的github版本。此文件由论坛教程自动生成。 本手册包含了小强的安装及使用的必备信息，可以通过阅读本手册来学习小强的使用方法。

以下的教程重点在于使用小强的各种功能，并不包含对ROS的教学。如果您是ROS的初学者，请在进行下面的教程的同时跟进学习ROS基础知识。 请不要只跟着我们的教程输入指令，要理解每一个命令的具体含义。遇到问题时也要善于自己寻找问题的原因，这些对于以后的工作学习都会很有帮助。如果您遇到了关于教程或者ROS不理解的地方也欢迎在我们ROS交流群(538456117)中提问。

- 开始使用
  - 设置网络
  - 产品组装
  - 远程连接
  - 状态检查
  - 远程遥控
  - 软件整体结构和说明
  - ROS入门手册

开始使用

本章介绍平台的快速使用方法，如果您对ROS系统和Ubuntu系统不熟悉，请先阅读第二章的教程一。

设置网络

首先将小强的主机连接上电脑显示器（小强pro用户请利用附赠的HDMI到VGA转接头进行连接, 小强mini用户请直接使用vga）。小强的默认密码是xiaoqiang，请及时更改默认密码。进入系统设置好小强的wifi网络连接。推荐设置路由器使小强使用静态ip,静态IP设置方法，方便以后连接。

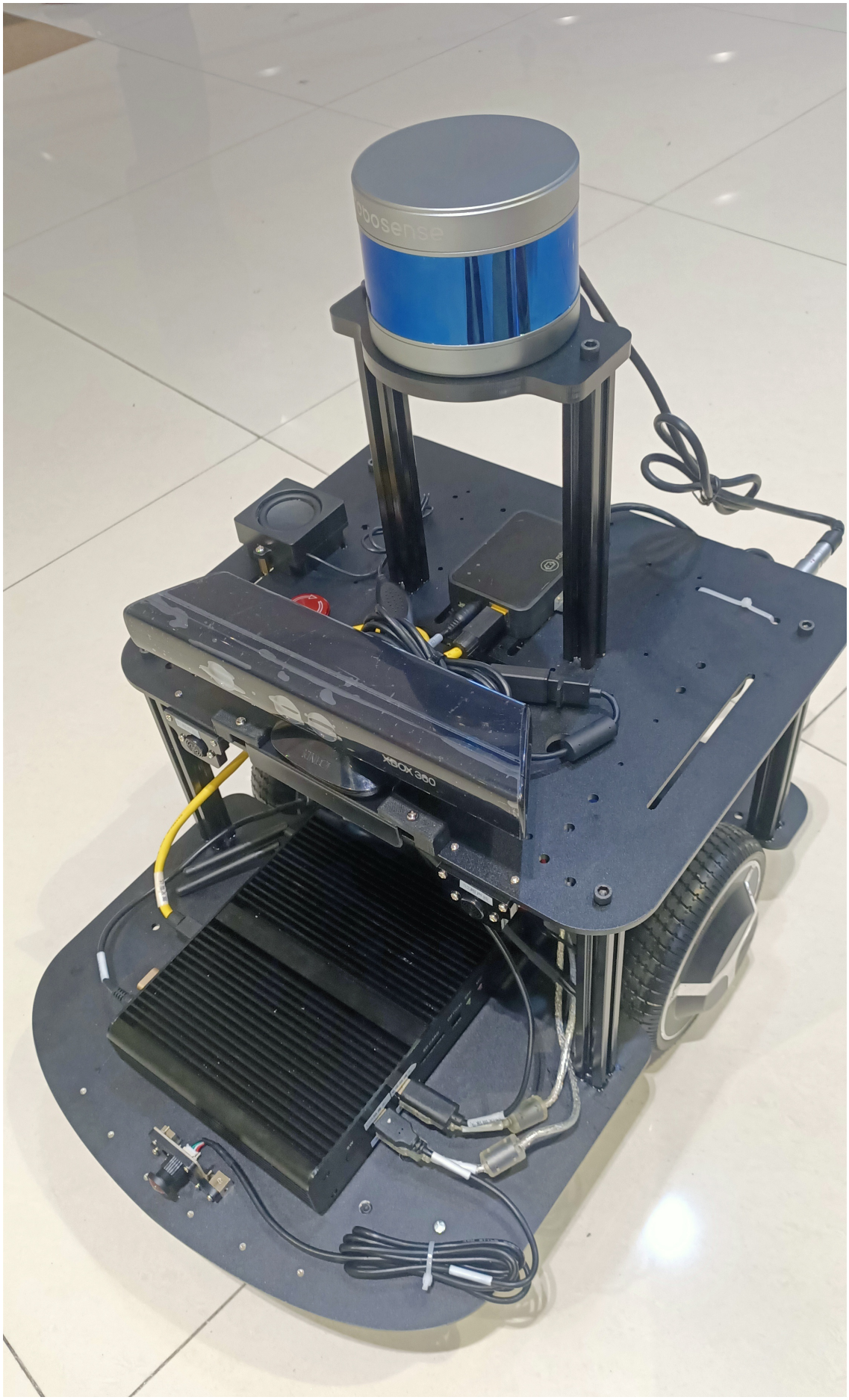
产品组装

小强的主要部分如下图所示

将电池平放在主机前方空余区域（电池靠两个小黑块堵住），根据线标提示接上底盘电源线，安装好电脑主机wifi天线，摄像头和底层USB连接模块连接上主机的USB接口即可。

注意在长时间不使用小车的时候，请将电池断开连接，避免因为电池过放产生问题。

xq5 的组装图如下


robosense
XBOX 360

远程连接

机器人默认安装了VNC远程桌面程序，用户可以方便的通过远程访问机器人系统。

VNC下载

安装软件后在软件的地址栏输入机器人ip地址即可访问。vnc密码和用户密码一样都是xiaoqiang。

状态检查

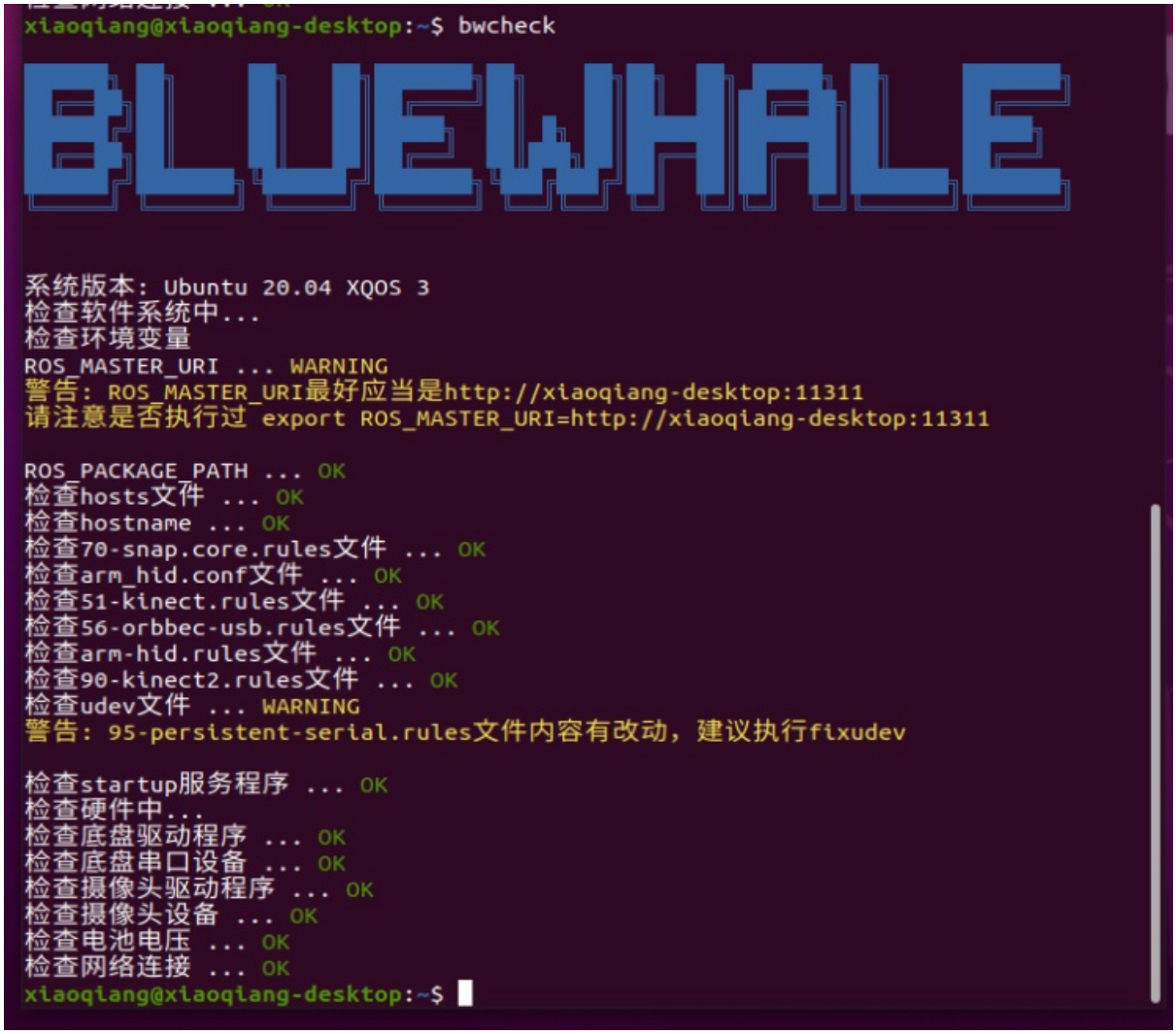
组装完成之后就可以开始使用小强了。打开小强主机开关，等待主机蓝色灯亮起。小强左侧电源数据显示正常（电池电压合理使用范围是10V以上，当电池电压小于10V时小强会自动关机）。

通过VNC远程连接，然后打开命令行终端(可以通Ctrl + Alt + T快捷键打开)。

检查程序是否正常运行，执行以下指令

```
bwcheck
```

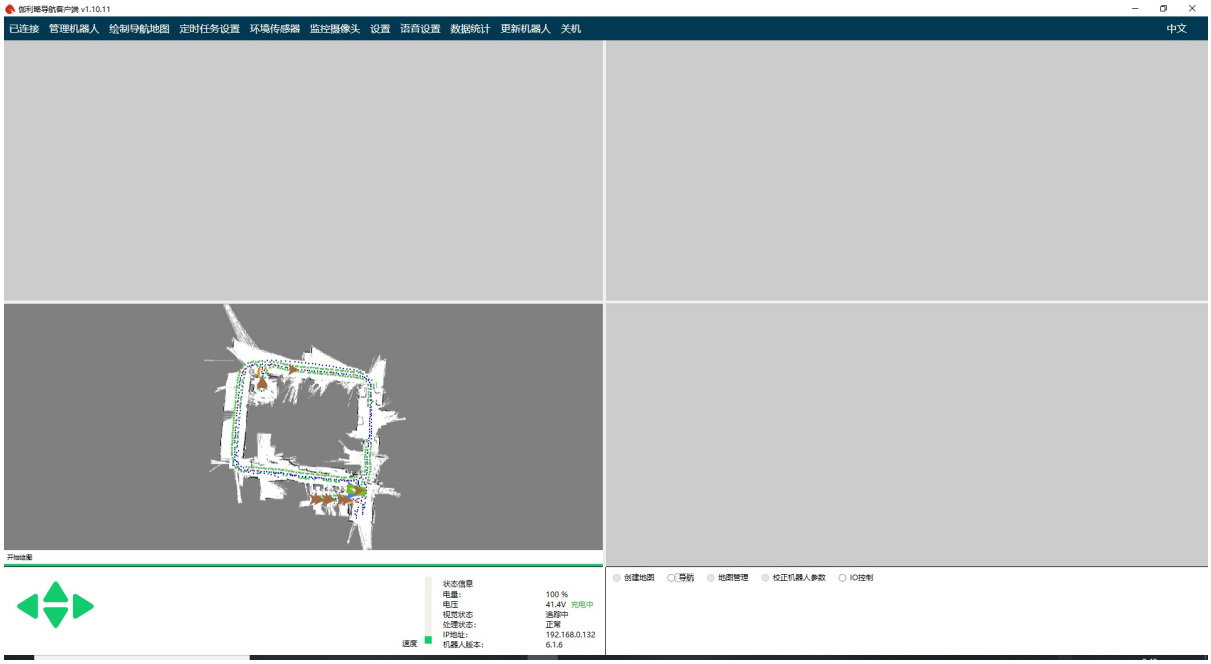
这个指令会自动进行机器人软件和硬件的自检。正常输出如下



根据硬件不同以上的程序输出可能不一样。黄色的警告可以忽略，红色的错误一般是比较重要的信息。可以根据错误信息对机器人进行调整。

远程遥控

远程遥控可以通过机器人客户端进行操作。客户端的下载和使用参见此教程。在开始进行开发前，建议先通过此教程了解和熟悉机器人的完整功能。



安装完成后，点击客户端左上角连接按钮，连接机器人。连接完成后可以通过WASD进行前后左右的遥控操作。可以多个按键同时操作以方便控制移动。上下方向键调整机器人速度。

如果发现机器人不能移动且之前的自检指令提示正常，则机器人可能触发避障。检查机器人红外传感器是否在亮，超声波传感器是否被阻挡，以及机器人急停按钮是否被按下。

软件整体结构和说明

小强的软件构建于ROS之上。程序主要包含有底层驱动，导航算法，slam算法。 对于机器人来说，软件是一个整体，各部分的依赖程度都比较高。为了方便管理，系统中的基础功能用一个service统一操作。也就是前文中提到的startup service。这个service会启动底层的驱动软件包和摄像头。 startup软件包位于/home/xiaoqiang/Documents/ros/src/startup。系统启动时会自动启动这个服务。如果想要更改这个服务的内容，可以修改这个软件包内的launch文件。具体的操作请参照这里。 系统的导航程序采用的是ROS自带的导航程序包。不过导航参数是根据小强自己的参数修改过的。导航参数对导航的性能表现影响很大，你也可以尝试自己修改参数以提高表现。

ROS入门手册

Learning ROS for Robotics Programming - Second Edition.pdf。这本教程很基础、很全面，虽然以Hydro版本为例，但是也完全兼容kinetic，noetic版本，代码实例中只需将书中的Hydro字符串替换成即可。请重点阅读本书的第二章和第三章。

- 小强ROS机器人教程(1)_基础操作介绍

小强ROS机器人教程(1)_基础操作介绍

开始之前

以下的教程重点在于使用小强的各种功能，并不包含对ROS的教学。如果您是ROS的初学者，请在进行下面的教程的同时跟进学习ROS基础知识。 请不要只跟着我们的教程输入指令，要理解每一个命令的具体含义。遇到问题时也要善于自己寻找问题的原因，这些对于以后的工作学习都会很有帮助。如果您遇到了关于教程或者ROS不理解的地方也欢迎在我们ROS交流群(538456117)中提问。

小强主页

小强pro淘宝购买链接

小强mini淘宝购买连接

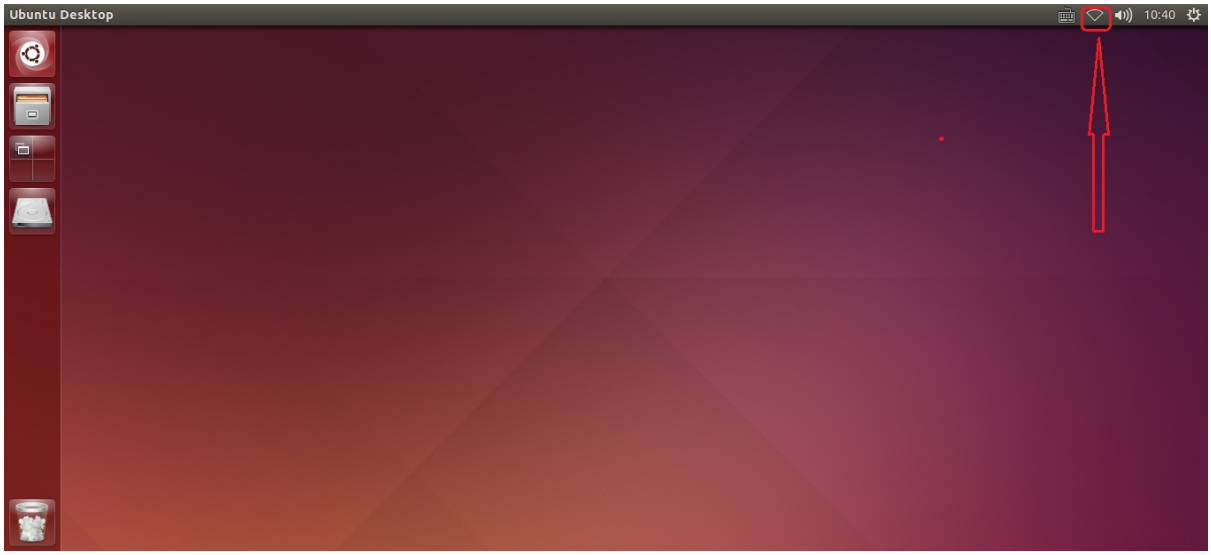
首先请您自行依据线标提示将小强接线连接好，完整结构如下图所示

1.配置小强网络

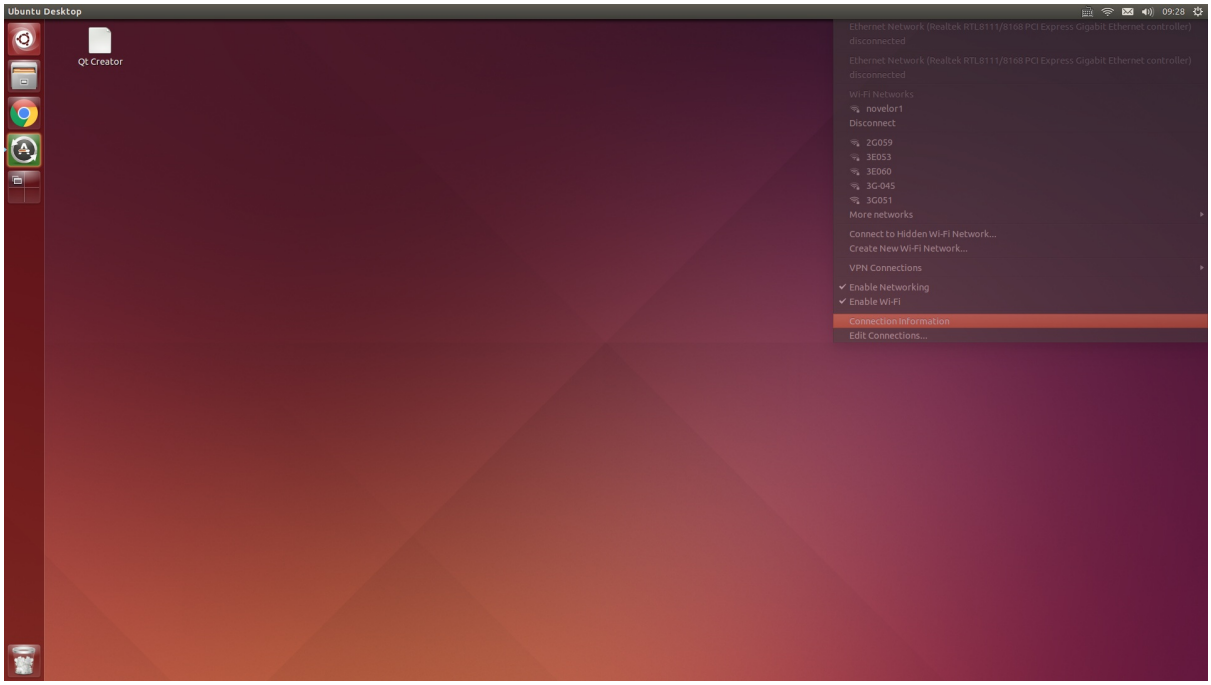
使用赠送的hdmi转vga插头(小强mini可以直接使用vga)，将显示器和键盘鼠标接入小强主机后开机，进入主机ubuntu系统。

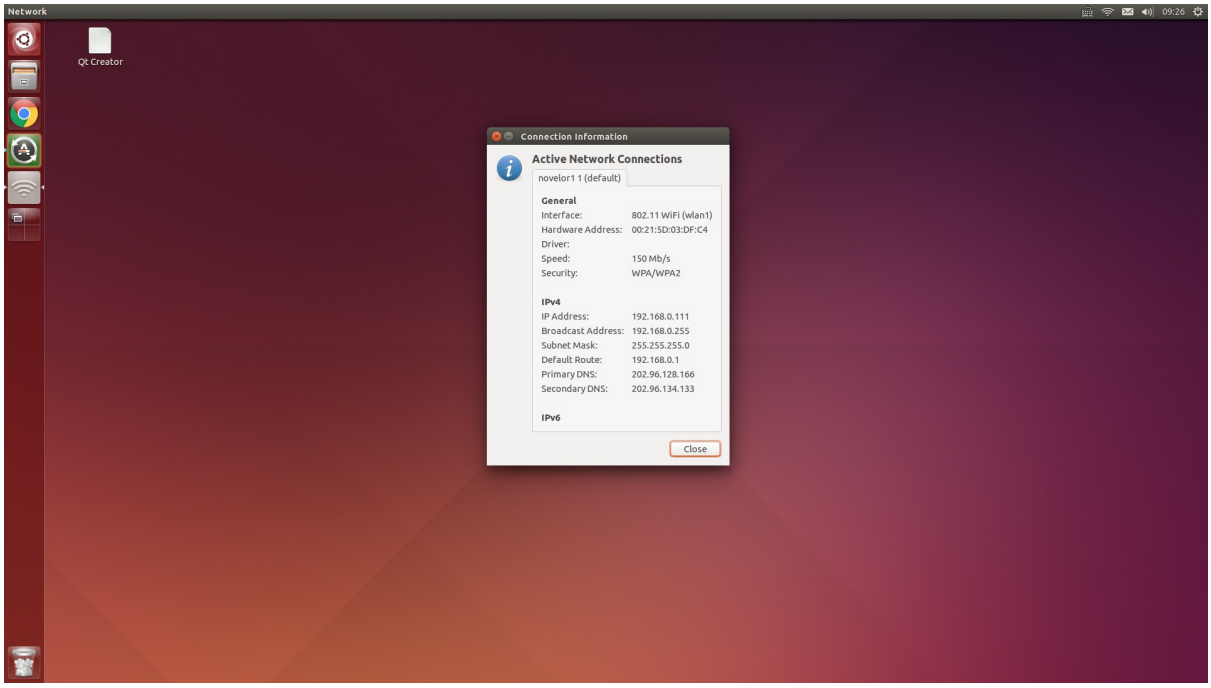
`小强默认密码是 xiaoqiang`

点击下图位置，选择需要接入的无线网络

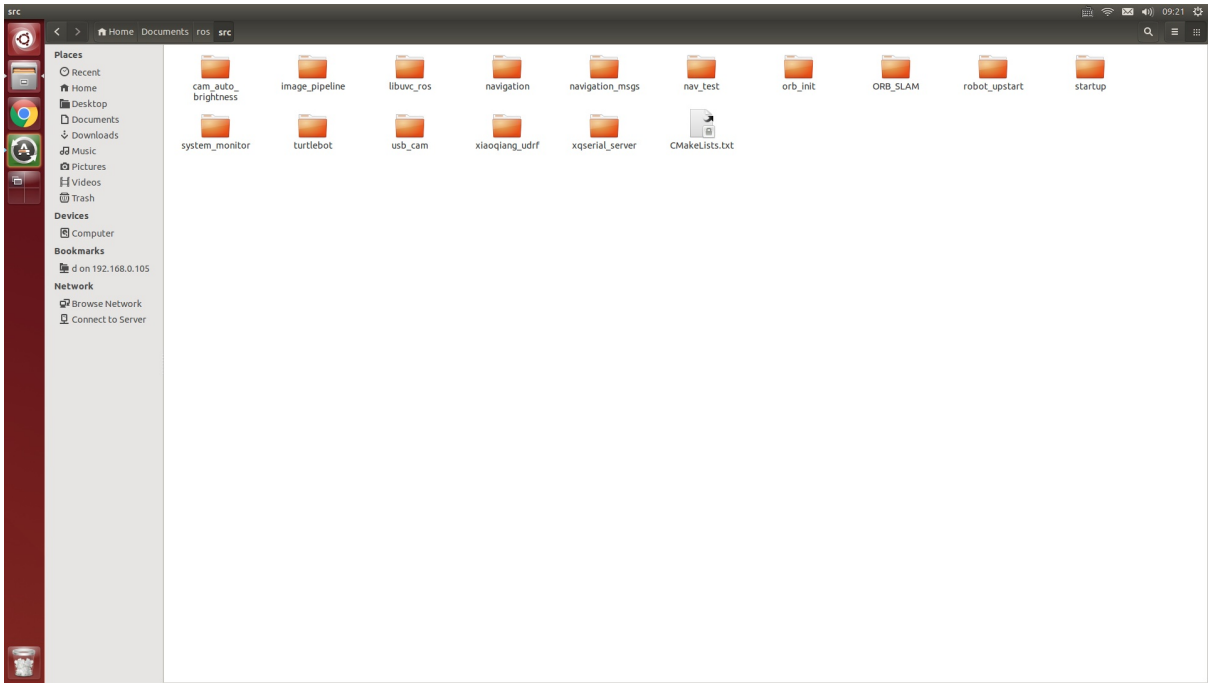


接入网络后，点击下图位置得到小强的实际ip地址，后续教程会频繁使用到这个ip信息





小强的ROS默认工作目录在这里



现在小强已经配置完成了，关机后拔掉键盘鼠标和显示器，下次小强直接开机就能使用

2.本地遥控端配置

由于小强安装了VNC远程桌面，本地遥控端只需要安装一个VNC Viewerhttps://www.realvnc.com/en/connect/download/viewer/即可。 同时保证本地遥控端必须和小强在同一个局域网下

后续由于需要频繁的更改机器人上的文件，建议在本地电脑上安装VSCode。同时安装ROS插件和远程连接插件。通过远程连接方便的对机器人进行修改。

现在您已经具备小强的完整开发使用条件，例如ssh登录小强主机，输出机器人当前电压

通过VNC在机器人上打开一个命令行终端，输入以下指令

```
rostopic echo /xqserial_server/Power
```

正常输入如下

```
---
data: 40.11030960083008
---
data: 40.1114387512207
---
data: 40.1114387512207
---
data: 40.108055114746094
---
data: 40.102413177490234
---
data: 40.09339141845703
---
data: 40.09113311767578
---
data: 40.08774948120117
---
data: 40.08436584472656
---
data: 40.08662414550781
---
data: 40.08549880981445
---
data: 40.09000778198242
---
data: 40.09000778198242
---
data: 40.08775329589844
---
data: 40.09226608276367
---
data: 40.09564971923828
---
data: 40.09790802001953
```

不同型号机器人的电压可能会不同。

小强主页 返回目录

- 小强ROS机器人教程(2)_蓝鲸智能开源软件仓库的使用和ROS开机启动任务的配置
  - 一、startup软件包功能介绍
  - 二、修改软件包中launch文件夹内的startup.launch文件
  - 三、将startup.launch文件在小强主机中注册为startup 开机启动服务
  - 四、远程重启小强主机，查看开机启动项是否正常加载

小强ROS机器人教程(2)_蓝鲸智能开源软件仓库的使用和ROS开机启动任务的配置

小强主页

小强的所有软件源码都共享在蓝鲸智能的开源仓库里，任何人任何组织都可以自由下载使用或进行二次开发,Github仓库 Gitee仓库

对于小强用户，开源仓库中的软件可以直接git clone到小强的ROS工作目录里，然后就可以直接用ROS的工具链 `catkin_make` 编译使用。小强的ROS工作目录为: `/home/xiaoqiang/Documents/ros/src`

下文将以开源仓库中的startup软件包为例，演示开源仓库的完整使用过程。

一、startup软件包功能介绍

小强主机开机后，会自动启动名字为startup的linux服务脚本，startup服务脚本运行时会去启动 startup 软件包中的startup.launch文件 在Ubuntu系统中注册的复制品。因此我们通过修改startup软件包中的startup.launch文件，然后将这个文件在ubuntu系统中注册为startup服务，就能控制小强主机的开机启动任务了。

二、修改软件包中launch文件夹内的startup.launch文件

利用上篇教程安装的vscode编辑器，在本地遥控端直接编辑这个文件（需要远程访问小强的主机文件目录，请参考上篇基础操作教程进行配置)

在上图箭头区域，添加或删除你需要启动的ROS launch文件及ROS node，这些项目在下文将被添加进小强主机的开机启动项里，小强下次开机会自动运行这些项目。最后保存退出

三、将startup.launch文件在小强主机中注册为startup 开机启动服务

接着二中的机器人命令行窗口输入

a.首先将之前注册的startup服务停止和删除

```
sudo service startup stop
rosrun robot_upstart uninstall startup
```

b.重新注册startup 开机启动服务

```
rosrun robot_upstart install startup/launch/startup.launch
sudo systemctl daemon-reload && sudo systemctl start startup
```

四、远程重启小强主机，查看开机启动项是否正常加载

接着上文的ssh窗口输入

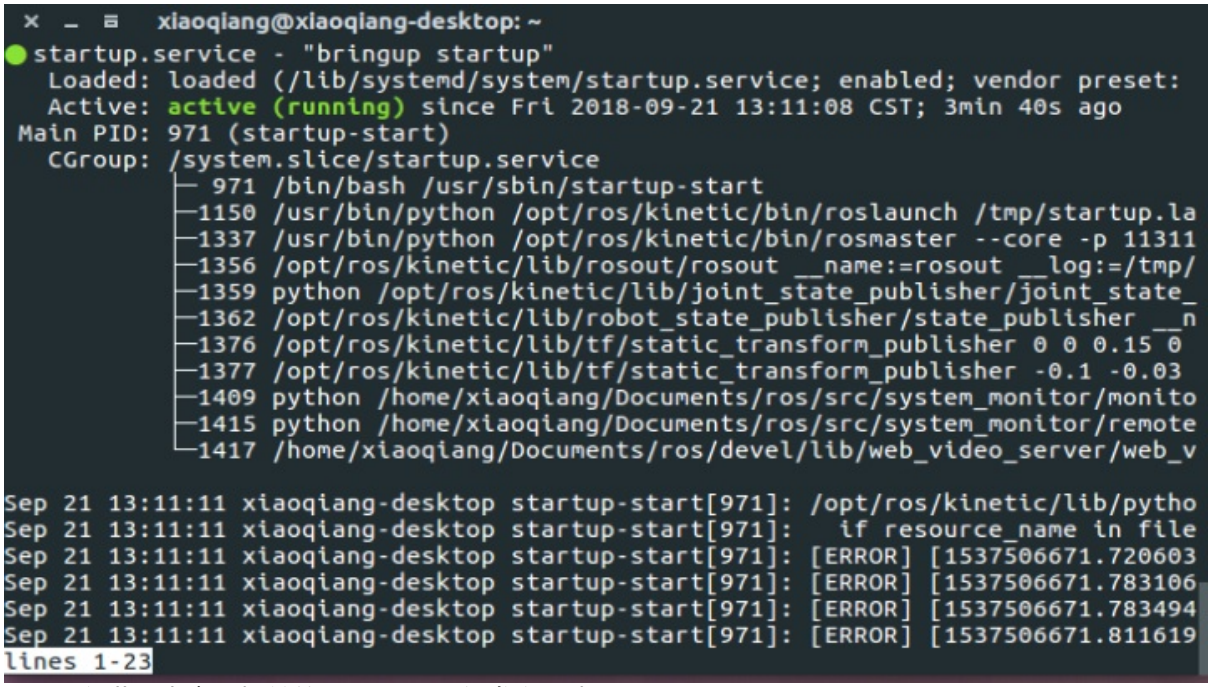
下发重启命令

```
sudo reboot
```

查看startup服务状态

```
sudo service startup status
```

正常的话会显示startup start/running 如下图所示



d.还可以进一步查看相关的topic 是否已经发布出来

```
rostopic list
```

小强主页 返回目录

- 小强ROS机器人教程(3)_在rviz中显示小强机器人模型

小强ROS机器人教程(3)_在rviz中显示小强机器人模型

小强主页

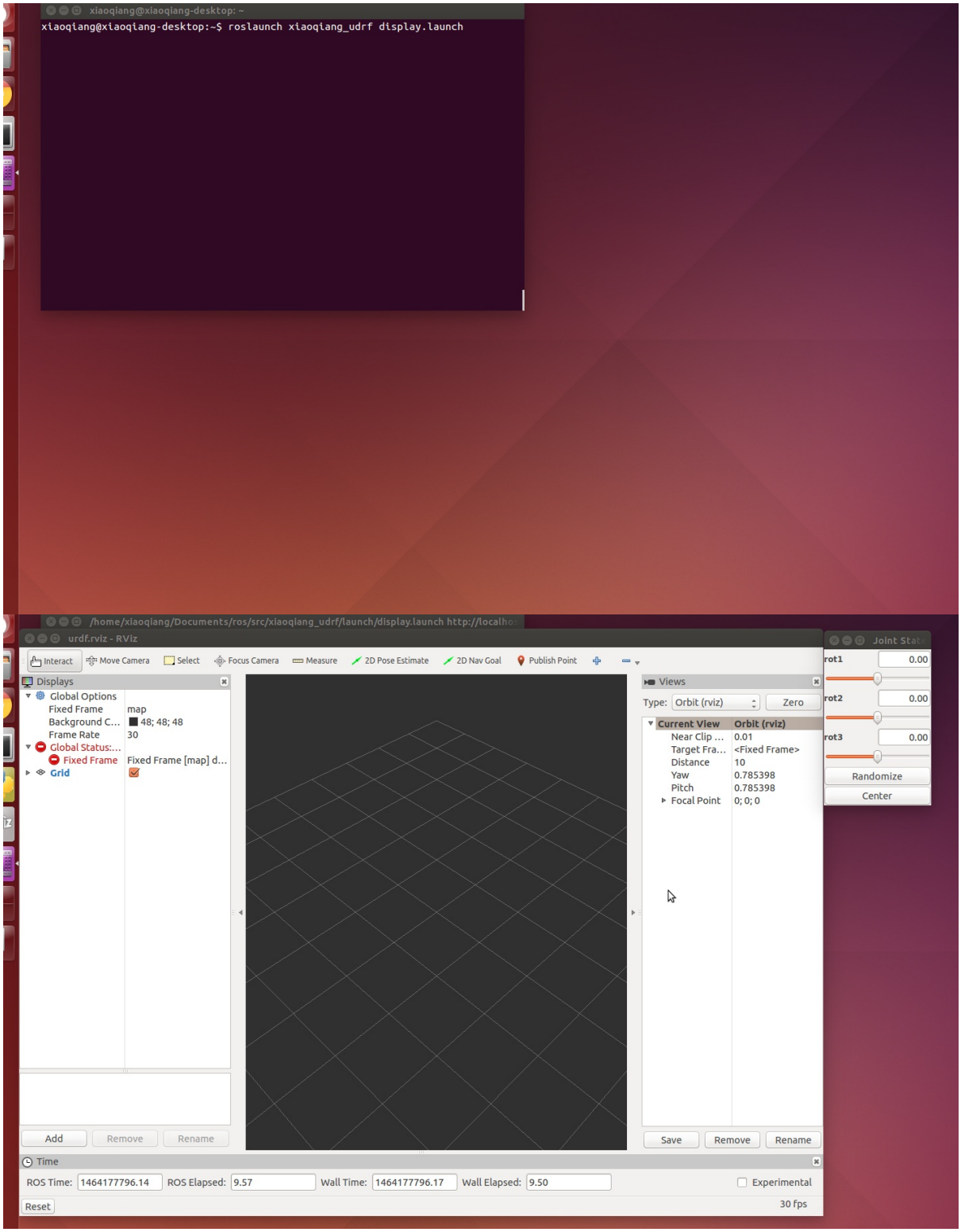
实时显示机器人当前姿态是一件很酷的事情，在ROS中借助rviz可以轻松实现这个目标。先看最后的 效果

00:00 / 00:00

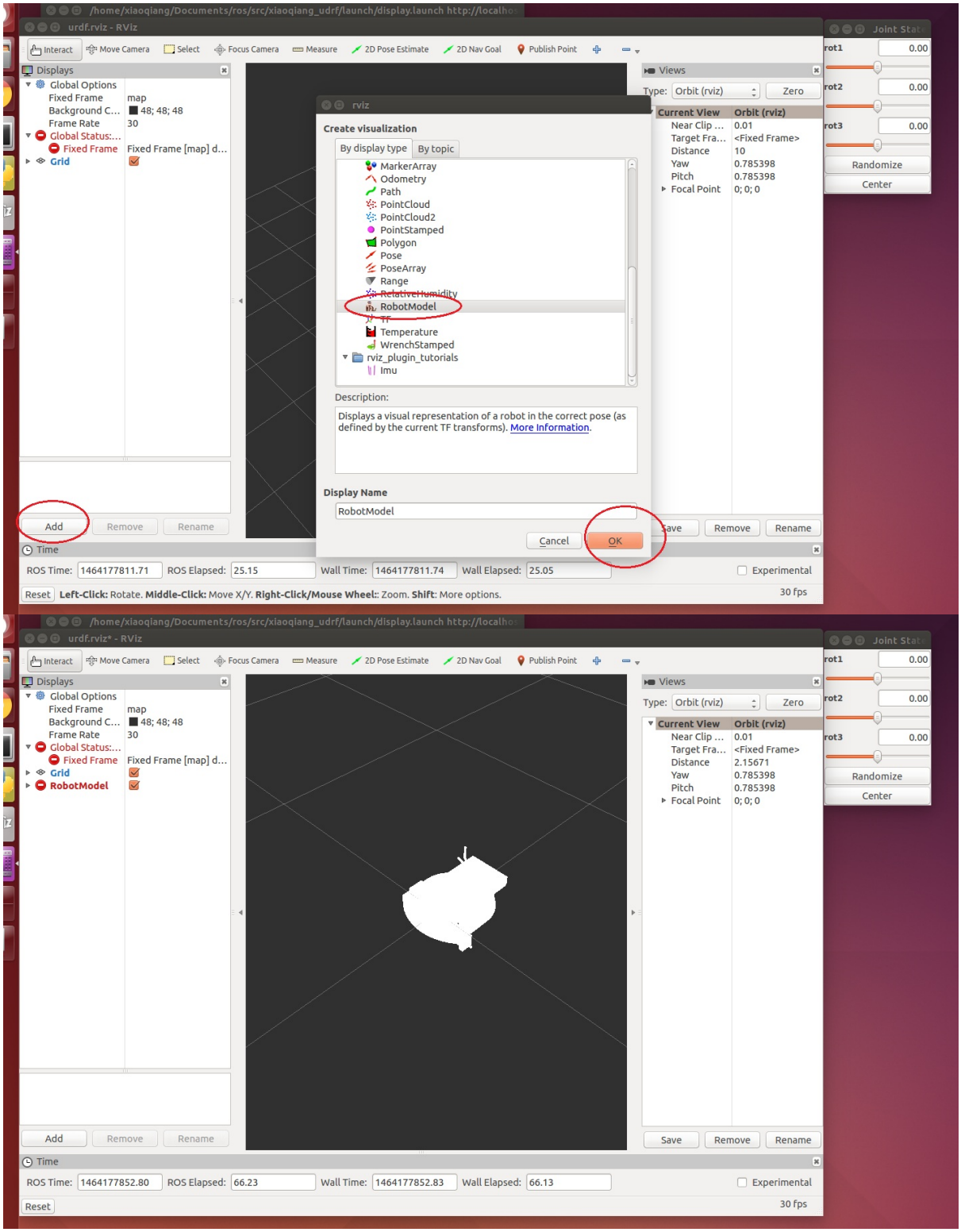
小强模型的软件包在 `Documents/ros/src/xiaoqiang_udrf` 也可以到我们的开源仓库去下载 Github

```
# 将小强主机接入显示器和键盘，开机后，打开终端
# 先关闭开机任务
sudo service startup stop
roscore
# 在小强主机上新开一个终端，启动这个软件包
roslaunch  xiaoqiang_urdf display.launch
```

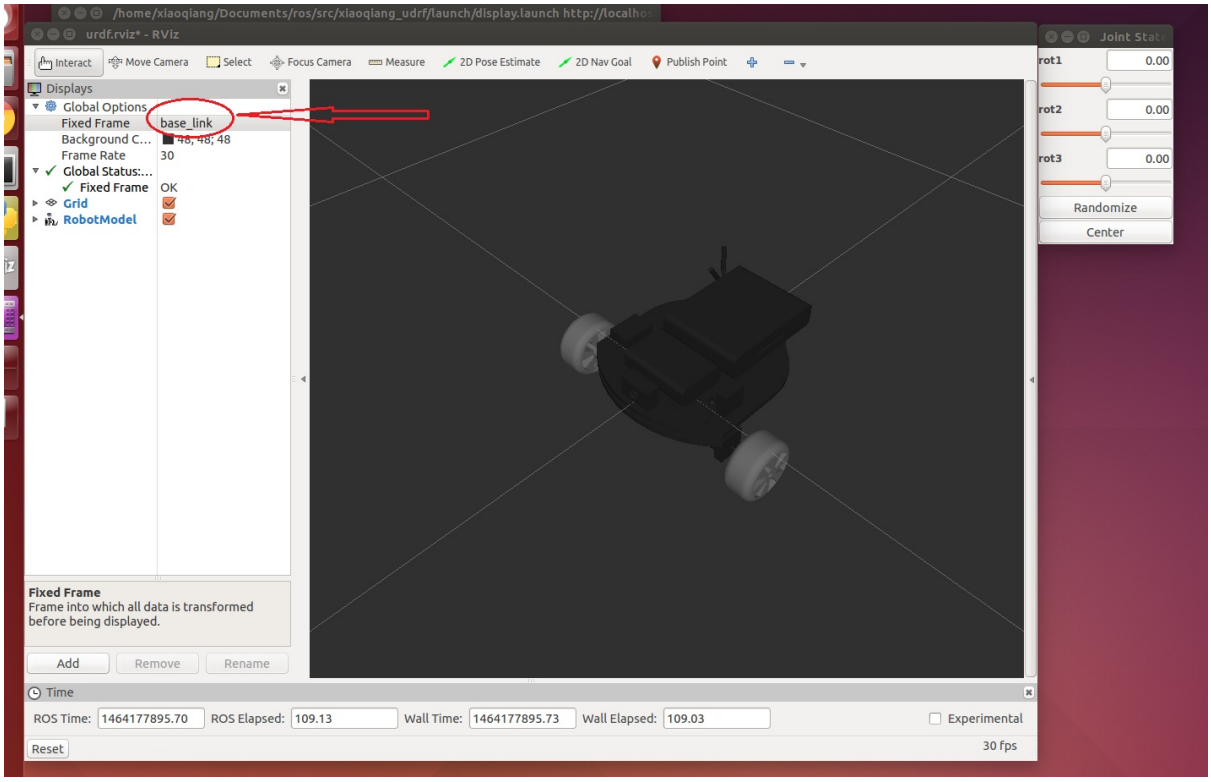
注意这个过程需要图形化操作，所以没办法通过远程ssh连接操作。如果一定要ssh连接，可以在连接时加上-X（注意是大写的）选项，但是效果不好，并不推荐。推荐通过VNC进行操作。VNC的安装方法可以参照这篇文章 `如果是远程打开模型包，需要先在本地安装xiaoqiang的模型包。对于使用xiaoqiang镜像系统的用户，xiaoqiang的模型包已经提前安装好。`



此时发现没有任何显示，需要添加rviz显示项目



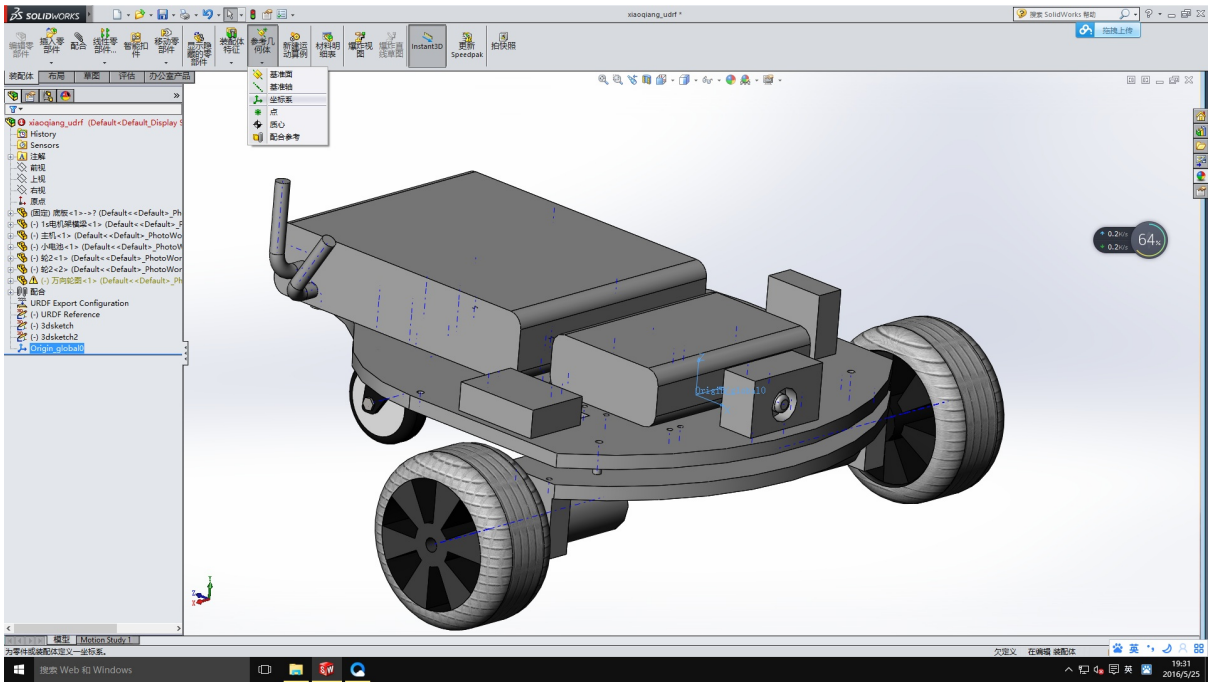
还是有问题，整个模型透明发白，这是因为rivz中的全局坐标系“fixed frame”设置的不合适，将map改成base_link后即可正常显示



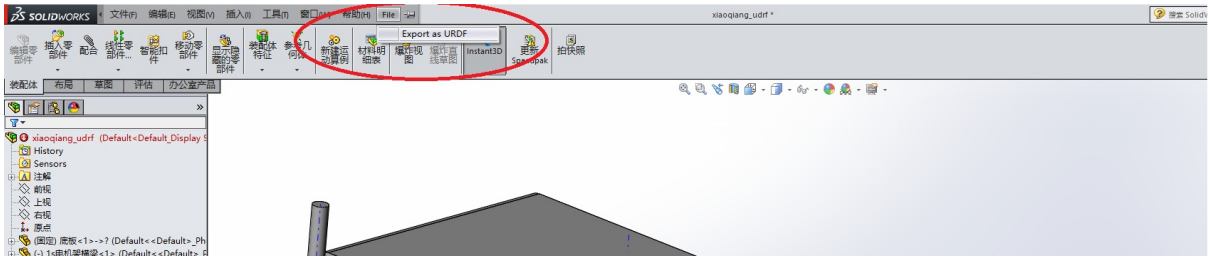
现在操作右上角的滑动条就可以使相应的轮子转动，over！

上面我们简单演示了rviz显示udrf模型的使用方法，下文将详细介绍在windows系统下用solidworks制作udrf模型的整个过程。

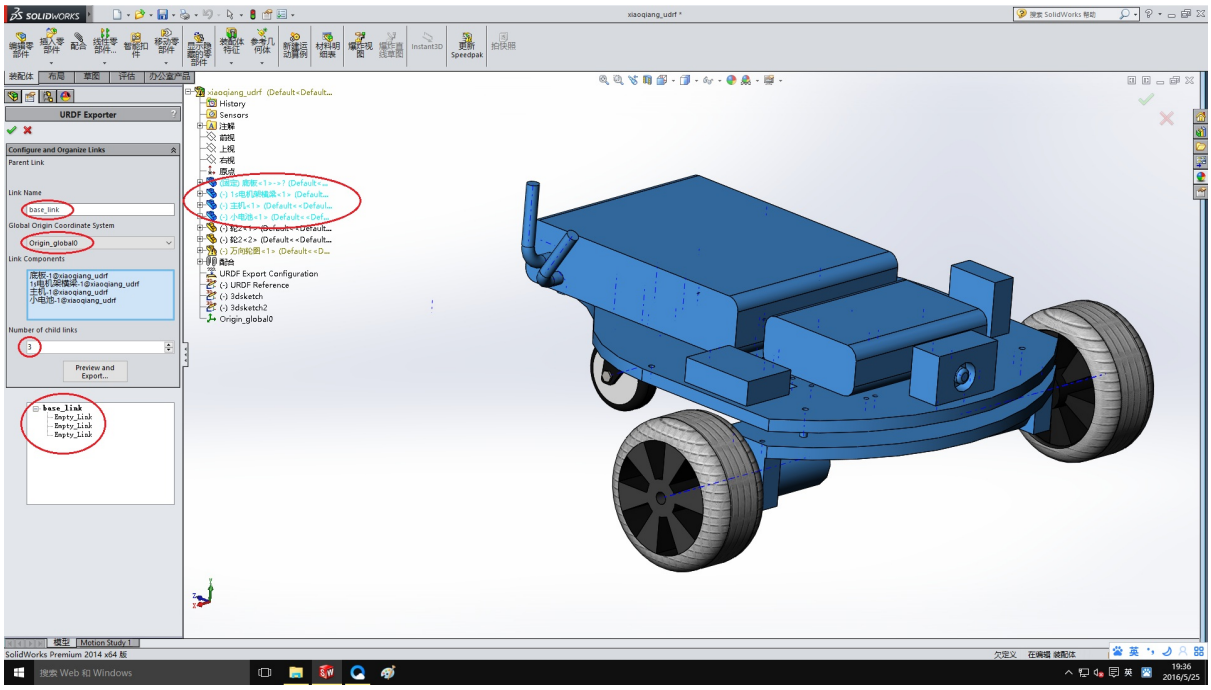
先用solidworks 建立小车模型，并下载安装好solidwork转udrf插件下载地址. 制作好模型后，需要补加一个坐标系，这个坐标系在下文将作为整个udrf模型的基准坐标系（即ROS中的base_link frame）。



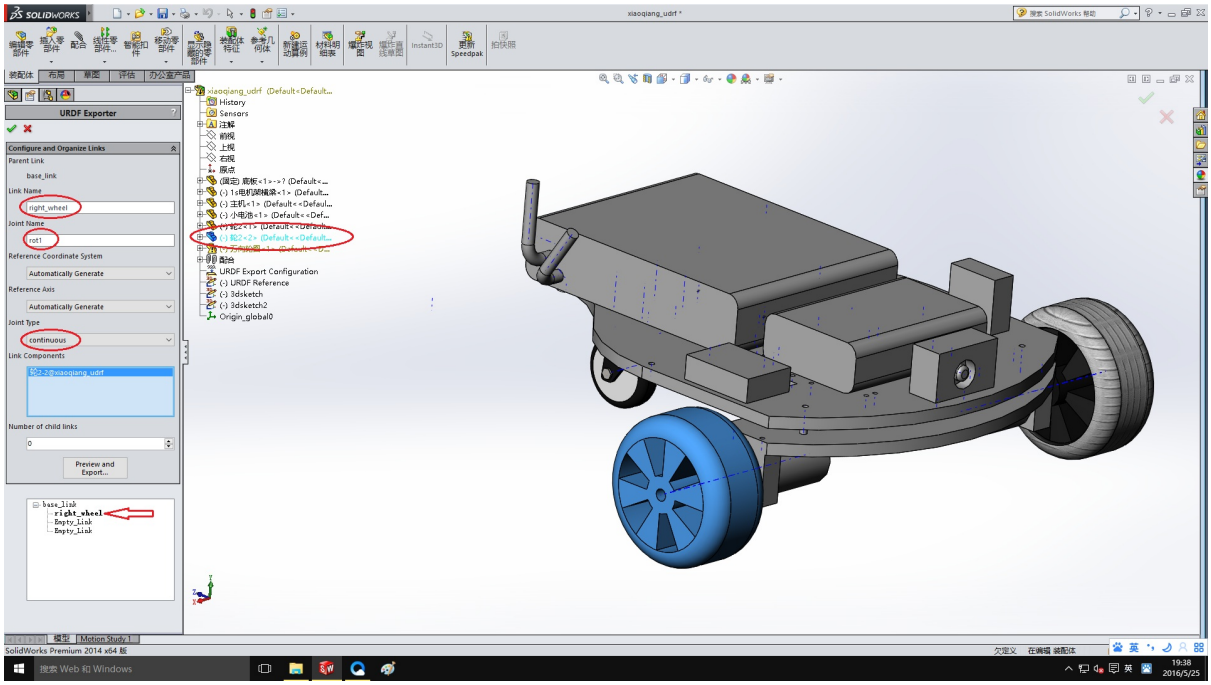
打开udrf插件



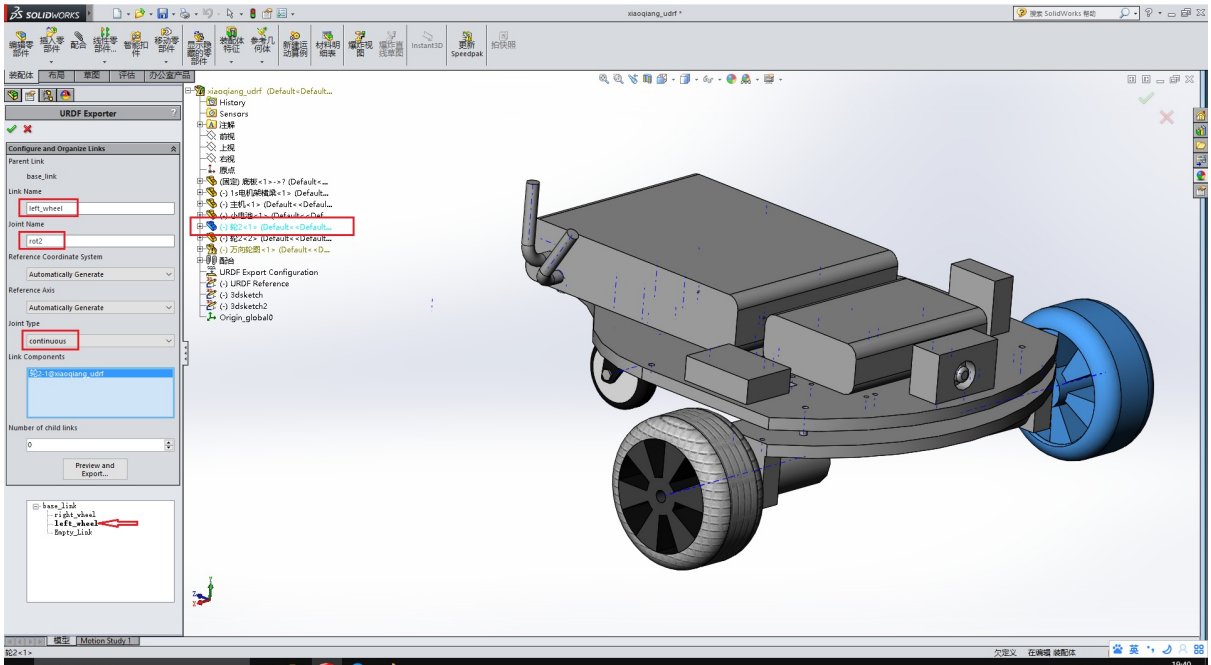
小强有两个驱动轮和一个从动轮，所以整个模型需要3个link，3个joint.首先编辑base_link,留意上文中全局坐标系，图片中红色区域就是需要自己点击或者修改的项目

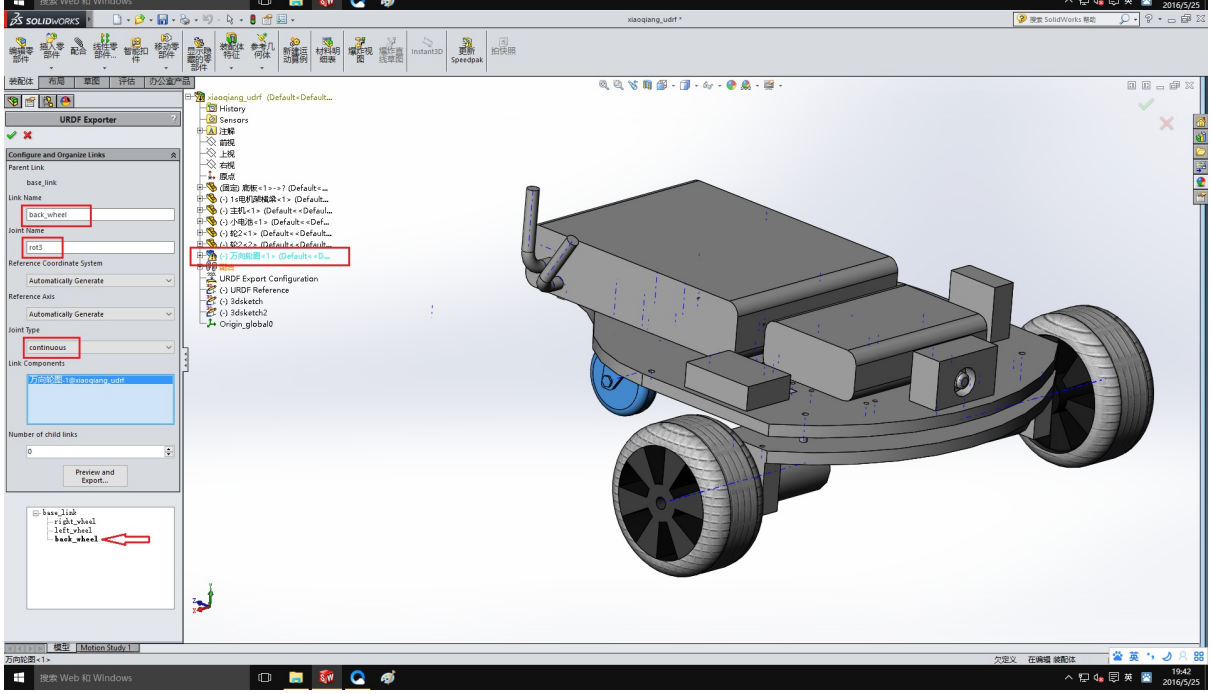


然后是右轮

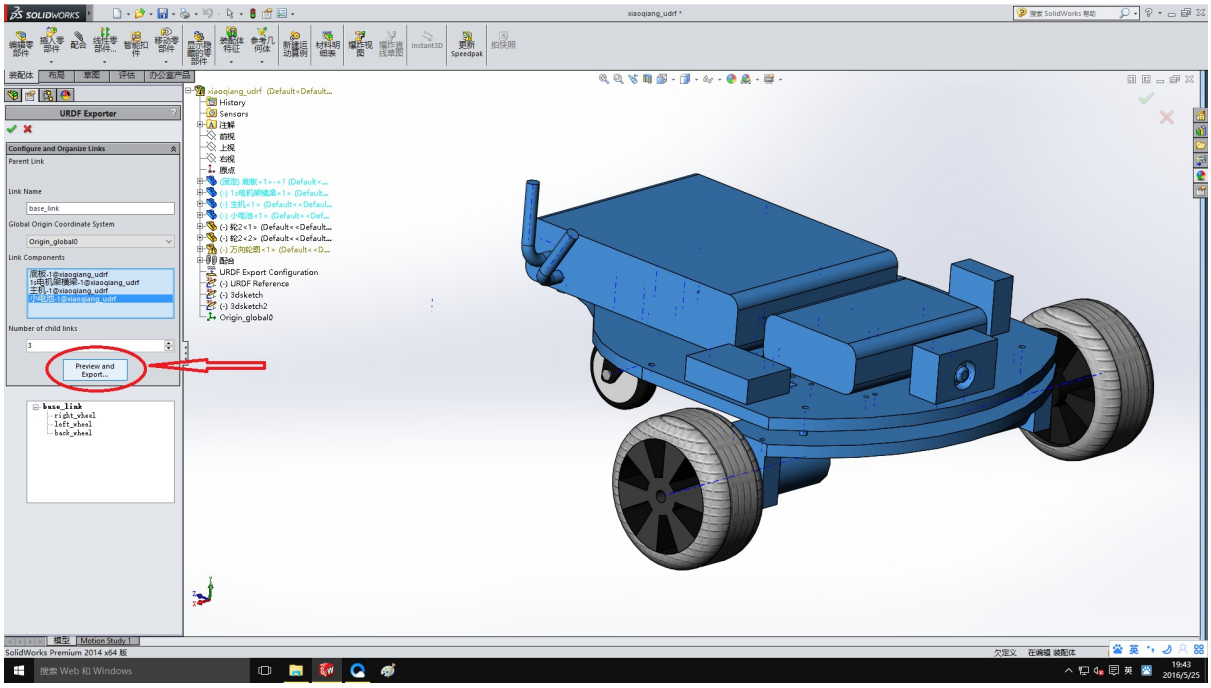


继续左轮和后轮

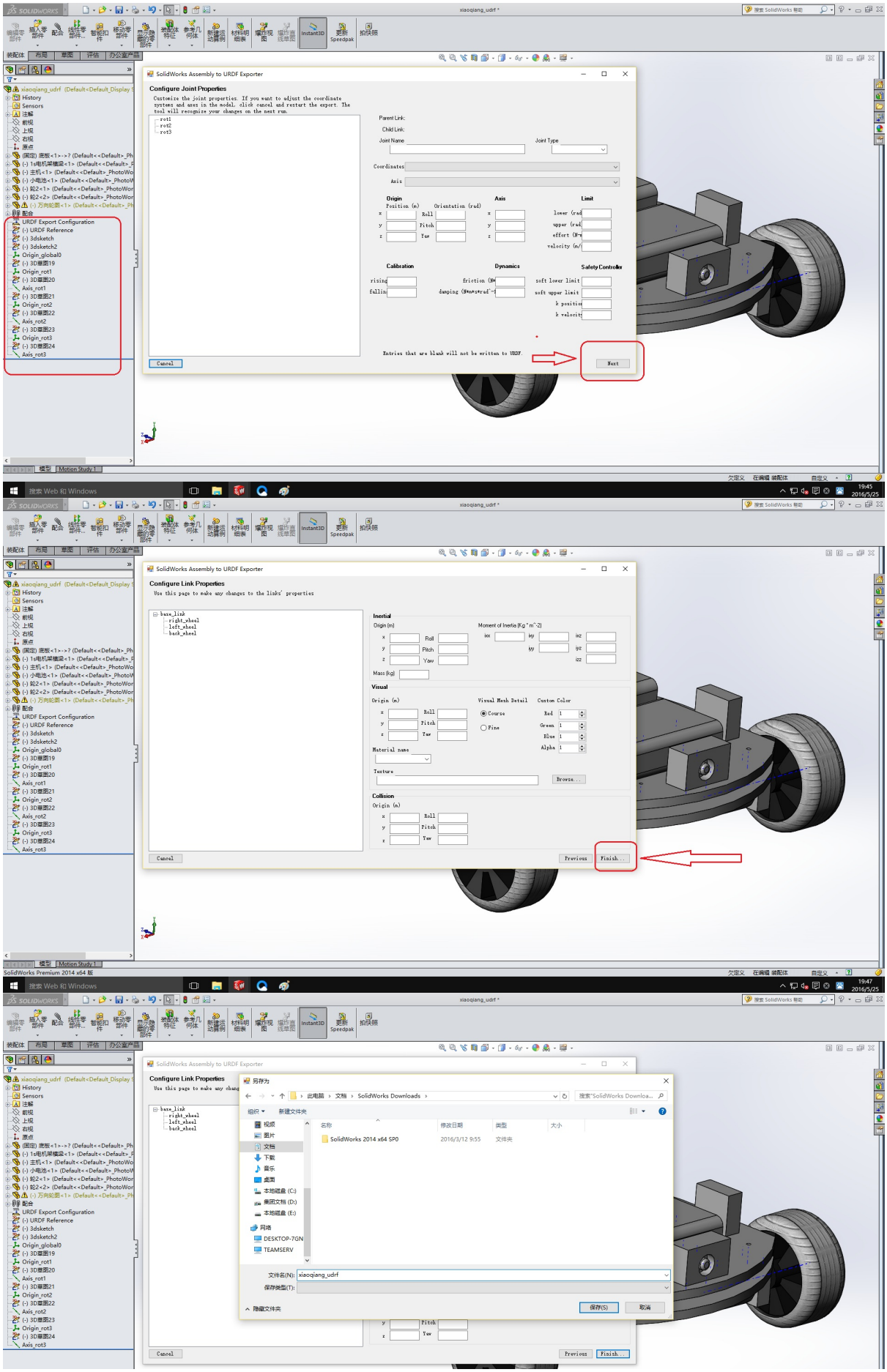


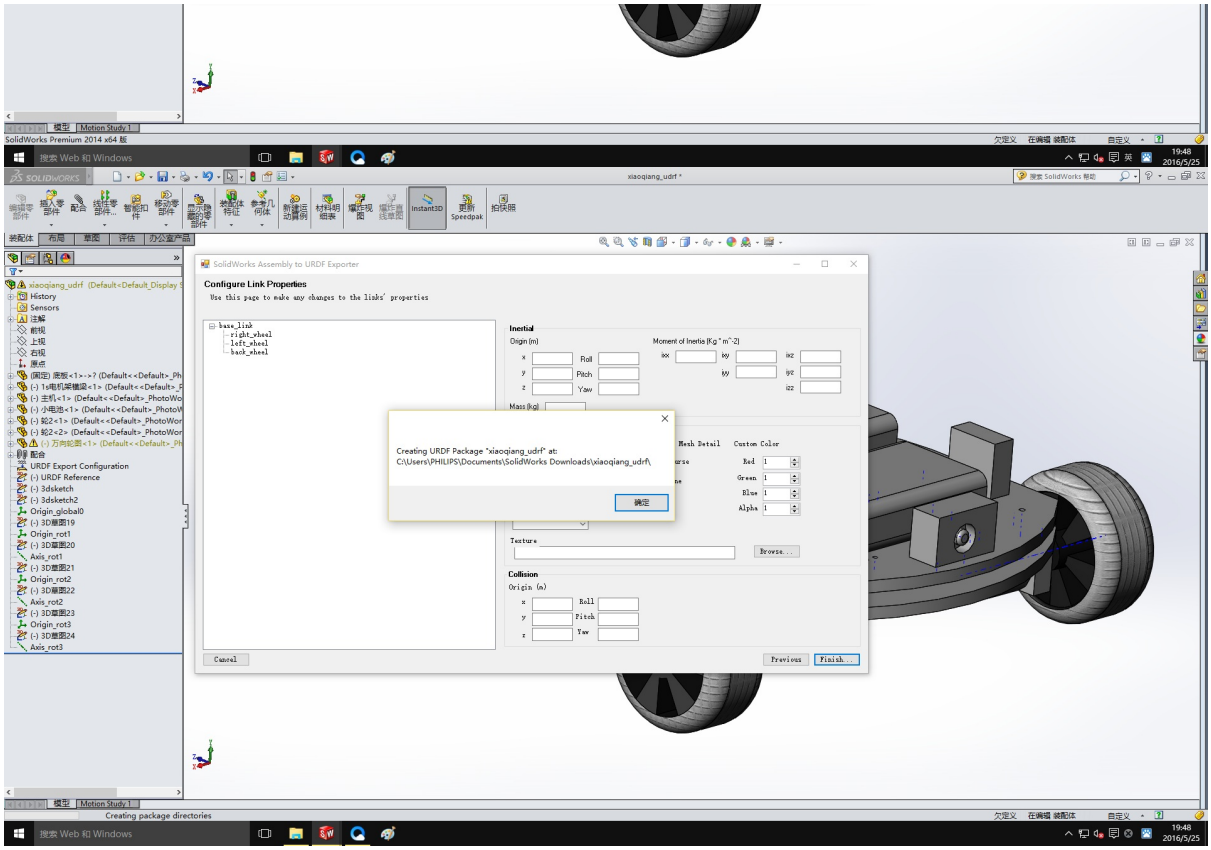


现在全部设置完成，开始导出

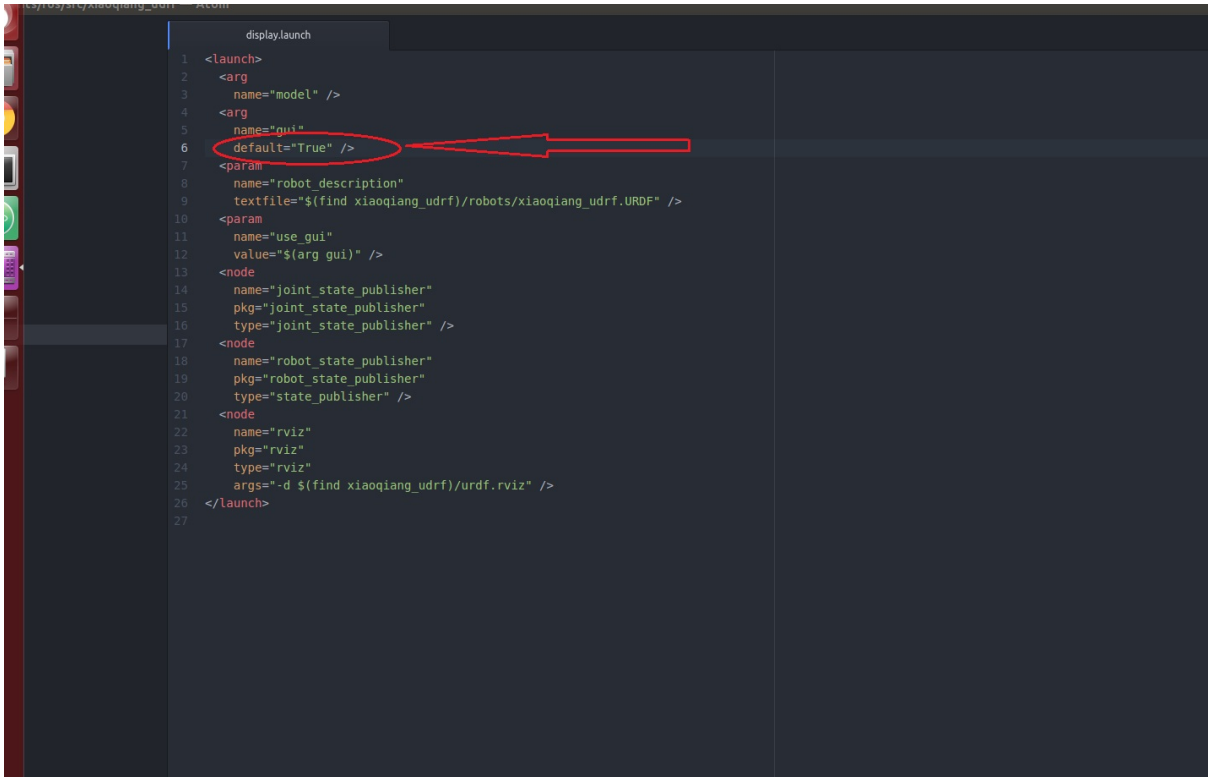


一路next和确定下去


SolidWorks Assembly to URDF Exporter
Configure Joint Properties
Customize the joint properties. If you want to adjust the coordinate systems and axes in the model, click cancel and restart the export. The tool will recognize your changes on the next run.
Parent Link:
Child Link:
Joint Name
Joint Type
Coordinates
Axis
Origin
Position (m)
Orientation (rad)
Axis
Limit
Calibration
Dynamics
Safety Controller
Entries that are blank will not be written to URDF.
Cancel
Next
Configure Link Properties
Use this page to make any changes to the links' properties
base_link
right_wheel
left_wheel
back_wheel
Inertial
Origin (m)
Moment of Inertia (Kg * m^2)
Mass (kg)
Visual
Visual Mesh Detail
Custom Color
Material name
Texture
Browse...
Collision
Previous
Finish...
另存为
SolidWorks Downloads
SolidWorks 2014 x64 SP0
文件名(N): xiaoqiang_udrf
保存类型(T):
保存(S)
取消



现在我们已经获到了机器人的udrf文件，生成的整个文件夹是一个ROS包， 修改launch文件夹内的display.launch文件 ,false改true



```
display.launch
<launch>
  <arg
    name="model" />
  <arg
    name="gui"
    default="True" />
  <param
    name="robot_description"
    textfile="$(find xiaoqiang_udrf)/robots/xiaoqiang_udrf.URDF" />
  <param
    name="use_gui"
    value="$(arg gui)" />
  <node
    name="joint_state_publisher"
    pkg="joint_state_publisher"
    type="joint_state_publisher" />
  <node
    name="robot_state_publisher"
    pkg="robot_state_publisher"
    type="state_publisher" />
  <node
    name="rviz"
    pkg="rviz"
    type="rviz"
    args="-d $(find xiaoqiang_udrf)/urdf.rviz" />
</launch>
```

将这个ros包复制到ROS工作空间中，catkin_make编译后就可以用本文开头的方法测试使用了。

小强主页 返回目录

- 小强ROS机器人教程(4)_惯性导航自主移动测试

小强ROS机器人教程(4)_惯性导航自主移动测试

小强主页

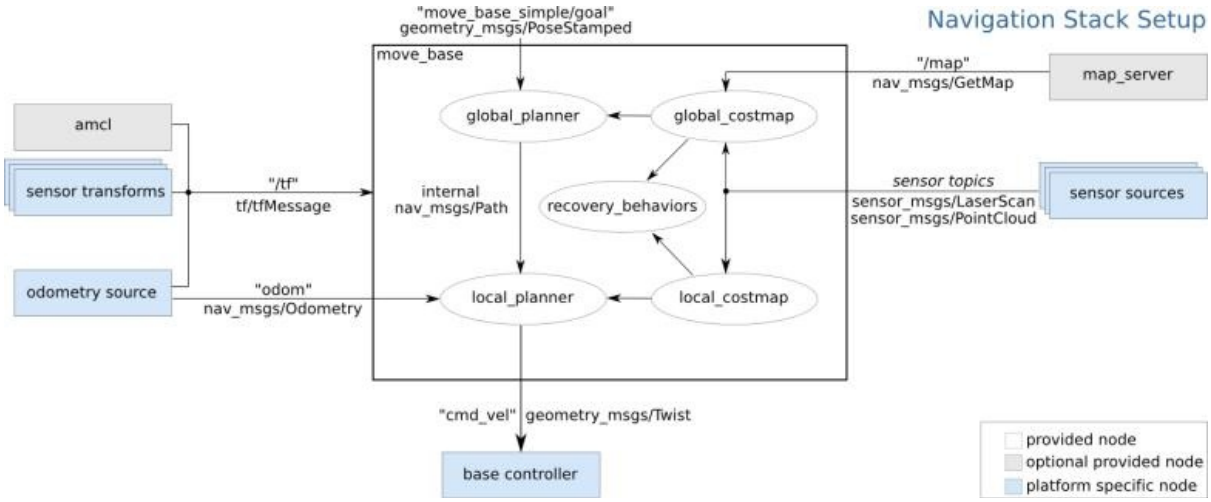
上一篇教程中，我们已经制作出了小强的3d模型，接下来开始测试小强的惯性导航功能。 这里的惯性导航，是指利用小强自身佩戴的惯性传感器（加速度和陀螺仪）和底盘编码器器信息进行定位和移动。需要的ROS软件包有：

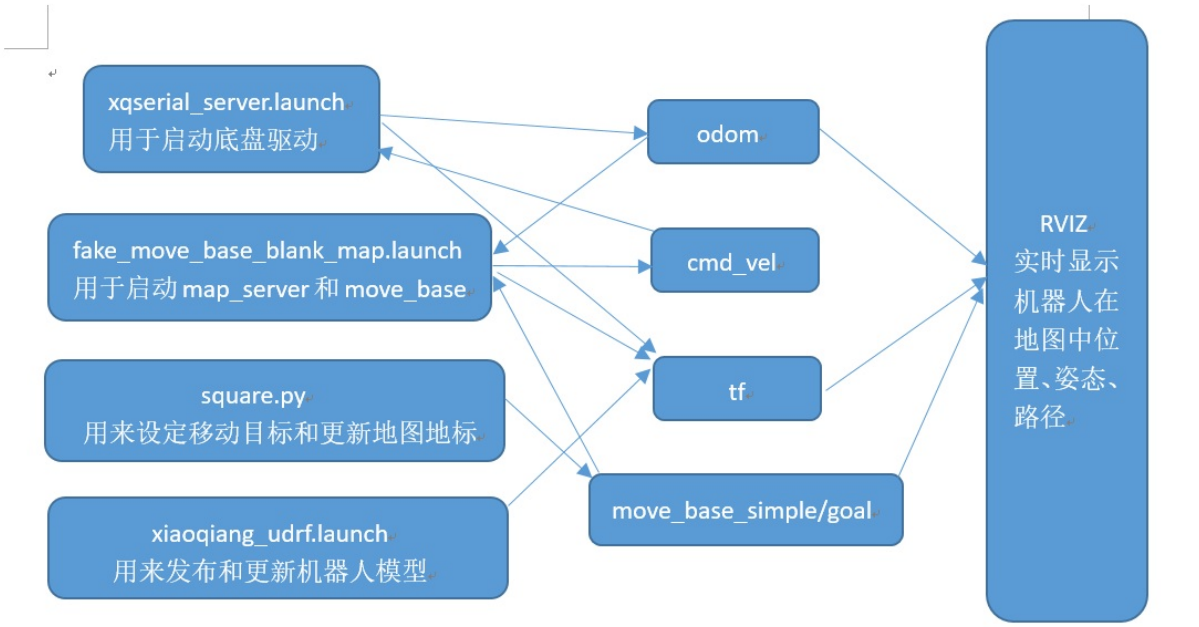
1. 底层驱动xqserial_server
2. 机器人模型包xiaoqiang_udrf
3. 惯性导航测试软件包nav_test

先上教程最后效果图

00:00 / 00:00

整个导航测试的内部框架如下两图所示





VNC方式在小强主机上完成的操作

请确保小强已经正常启动，小强主机正常启动完成后会自动运行上面提到的三个软件包，不需要手工启动相应的launch文件。如果是前期购买的用户，请自行更新安装这三个软件包，同时更新startup任务（请参考robot_upstart教程），再重新启动。

通过VNC在机器人上打开命令行

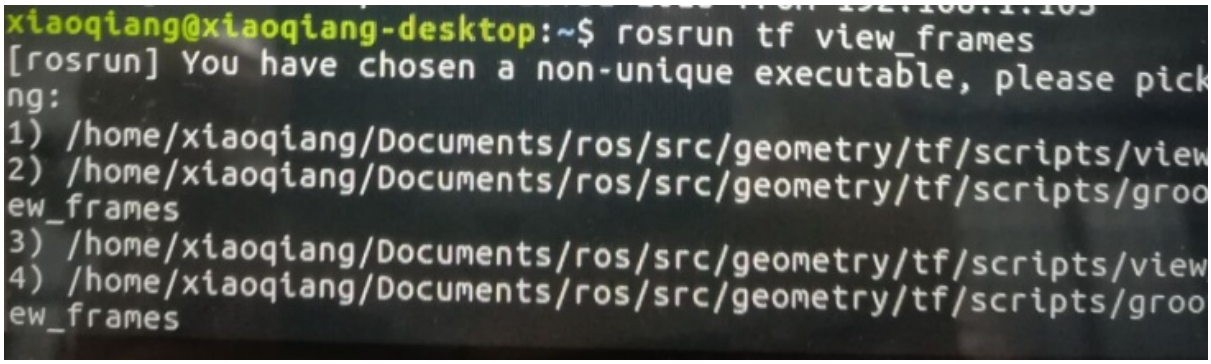
```
# 重启服务程序，有些程序基础可能没有运行
sudo service startup restart
roslaunch nav_test fake_move_base_blank_map.launch
```

`如果出现了 You must specify at least three point for the robot footprint 这样的错误可以忽略，这个并没有什么影响。这是由于foot_print要求设置一个形状，所以至少要三个点。foot_print就是机器人的轮廓形状，这个参数会用于避障。小强的foot_print设置的是圆形，所以并不需要指定三个点。如果出现的tf树和下图并不相同，请检查一下底盘驱动是不是正常运行`

在机器人上新开一个终端，检查是否所有tf都已经就位

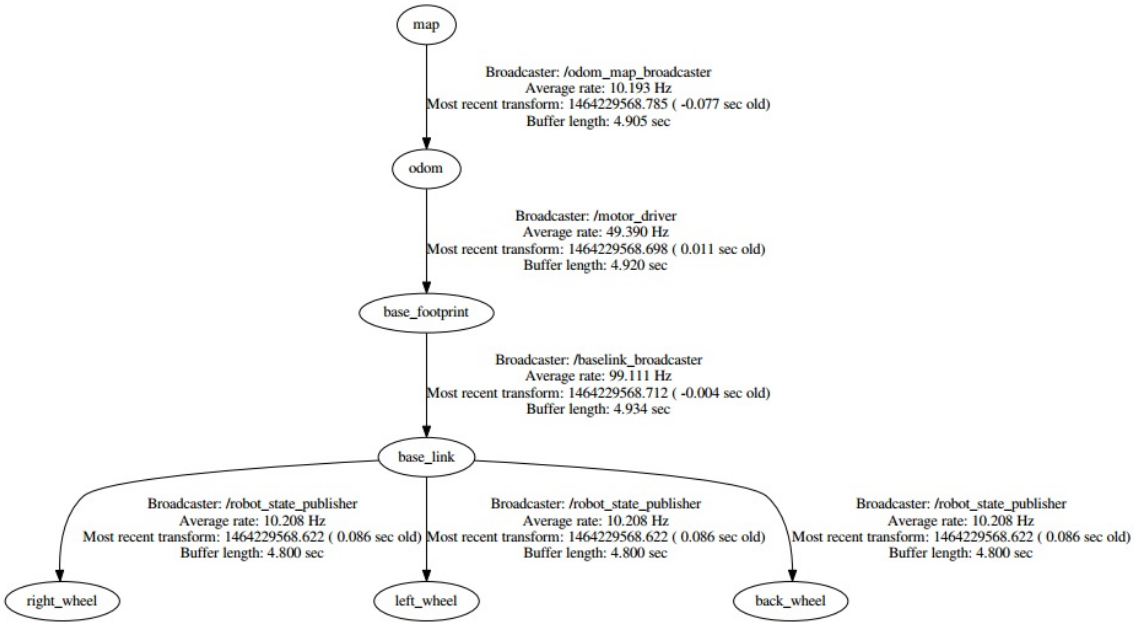
```
rosrun tf2_tools view_frames.py
evince frames.pdf
```

如果上面的指令提示如下图

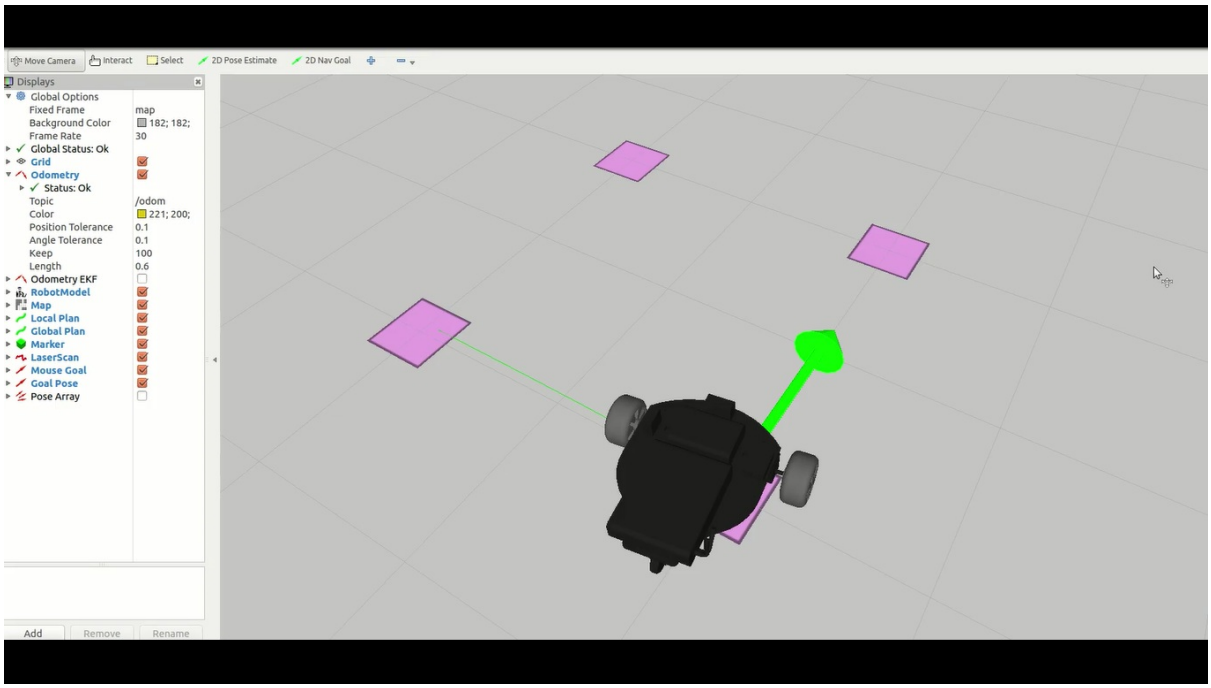


根据提示选择一个 `view_frame` 的程序。比如1

正常会显示下图



在机器人上打开rviz 当窗口打开后，点击左上角的file->open，选择小强里的/home/xiaoqiang/Documents/ros/src/nav_test/config/nav.rviz文件。这时界面应该如下图显示。



在机器人上打开一个新的窗口

```
rosrun nav_test square.py
```

如果小强无法移动，请检查底盘驱动程序是否正常运行。可以输入bwcheck进行自检。如果还是无法移动，可以再检查小强红外传感器是否被触发。触发时红外传感器会发光。

现在在rviz中就能看到文章开头的视频效果了

小强主页 返回目录

- 小强ROS机器人教程(5)_小强手机遥控app安卓版
    - 新版小强手机APP

小强ROS机器人教程(5)_小强手机遥控app安卓版

小强主页

新版小强手机APP

下载地址 说明： 增加导航控制，建图等功能

文档说明 此文档为送餐机器人说明。但是此APP与小强版本只有少量区别，可以参照使用。

使用方法

1. 保证小强和遥控用手机在同一局域网内
2. 在小强上开启服务端程序（默认自动启动）
3. 打开app，如果一切正常就可以看到小强的电压显示和小强的图像数据了。如果没有数据可以尝试点击重连按钮。

疑难解答

Q: 应用启动后无法连接
A: 可能是由于小强和你的手机没有处在同一局域网下。也有可能是小强的服务端程序没有启动。可以输入 `sudo service startup restart` 重启服务程序后再尝试。

Q: 成功连接后无法遥控
A: 检查底盘驱动程序是否正常运行。检查底盘串口USB是否正常连接。然后输入 `bwcheck` ，等待自检完成查看自检错误。如果自检正常但是还是无法移动，请检查红外是否触发。触发时红外传感器会发红光或黄光。

Q: 成功连接后没有图传视频
A: 检查摄像头USB是否连接正常。然后重启服务后再试一次 `sudo service startup restart`

小强主页 返回目录

- 小强ROS机器人教程(6)_小强图传遥控Windows客户端

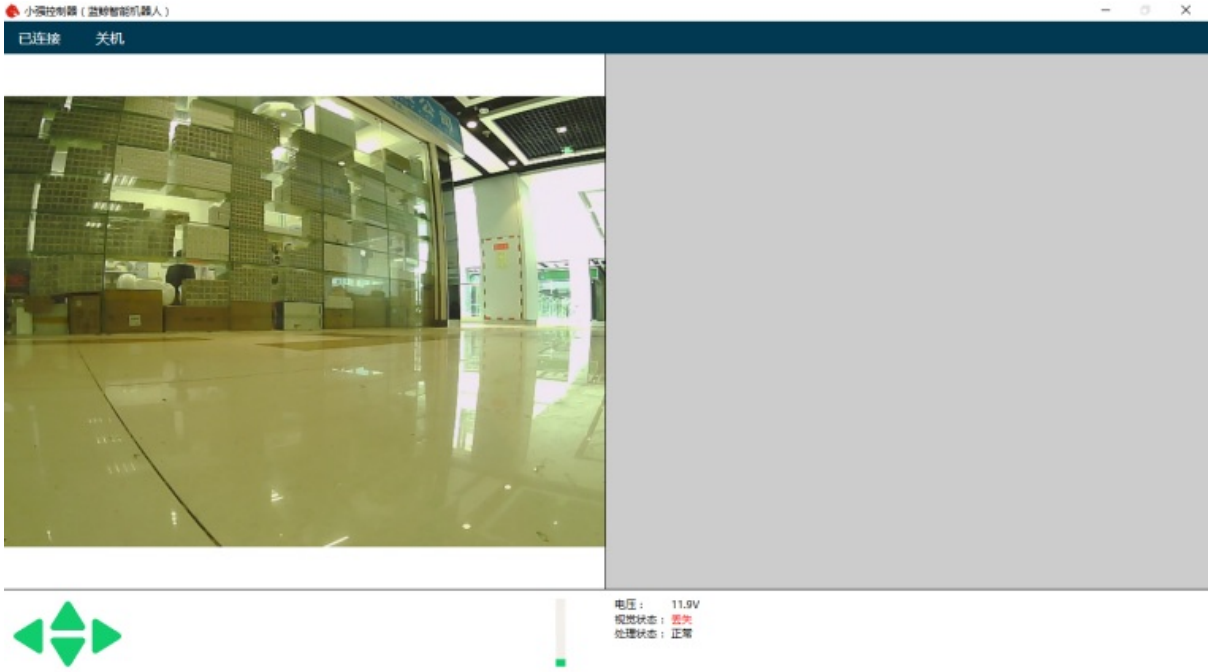
小强ROS机器人教程(6)_小强图传遥控Windows客户端

小强主页

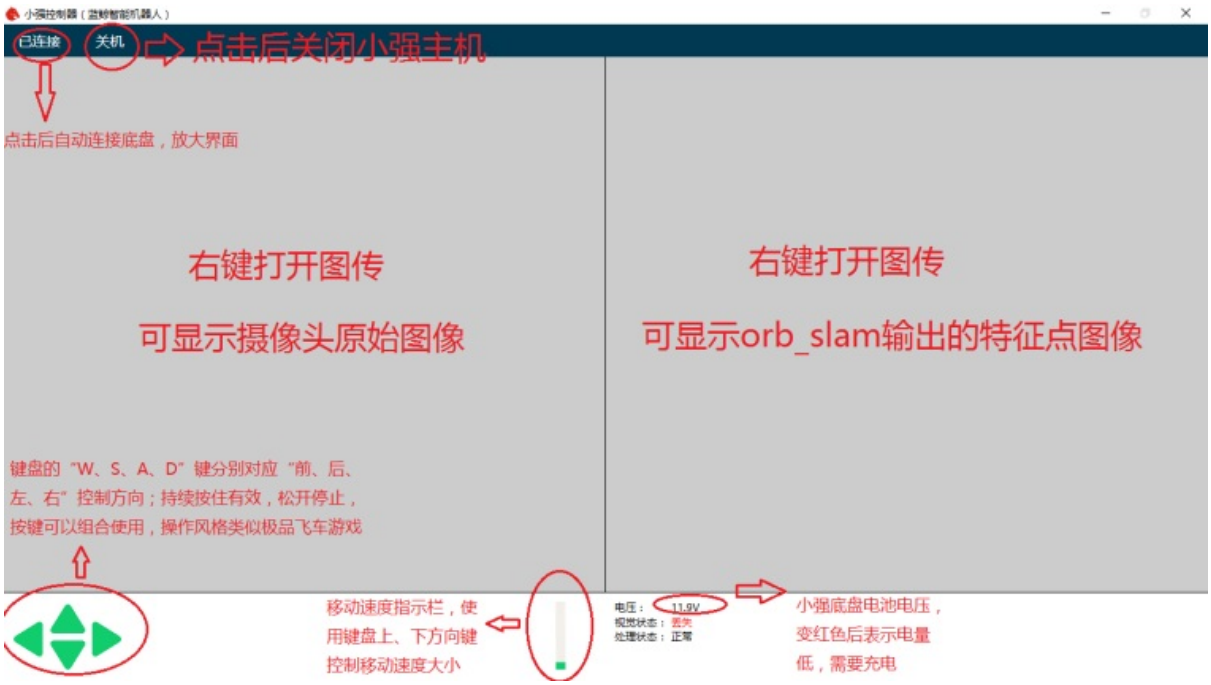
2017年3月份之前收到小强的用户，安装前请参考帖子升级软件包以支持小强图传遥控app

`目前Windows客户端已改为伽利略导航系统客户端，使用方法可以参照` 伽利略导航系统使用手册

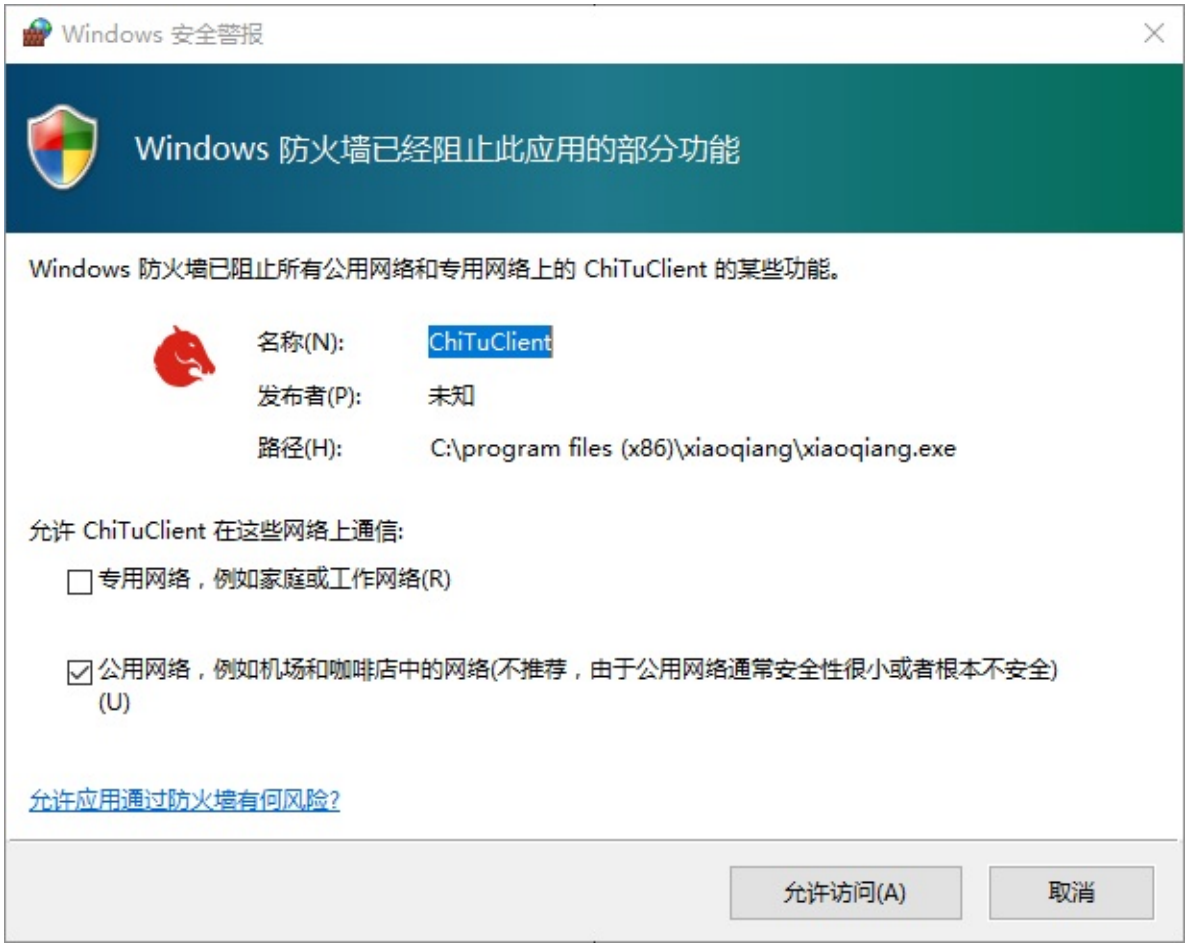
图传显示和控制界面



软件安装包下载连接 下载后双击安装，根据提示一直点继续就可以了。



初次使用电脑会有如下提示，请勾选“专有网络和公用网络”后，点击选择“允许访问”



疑难解答

Q: 应用启动后无法连接
A: 可能是由于小强和你的电脑没有处在同一局域网下。也有可能是小强的服务端程序没有启动。可以输入 sudo service startup restart重启服务程序后再尝试。

Q: 成功连接后无法遥控
A: 检查底盘驱动程序是否正常运行。检查底盘串口USB是否正常连接。然后输入 `bwcheck` ，看看自检是否有错误输出。如果自检正常但是还是无法移动，请检查红外是否触发。触发时红外传感器会发红光。

Q: 成功连接后没有图传视频
A: 检查摄像头USB是否连接正常。然后重启服务后再试一次 sudo service startup restart。输入 `bwcheck` ，如果自检数据正常，此时仍然没有图像显示则说明客户端安装异常。请检查一下客户端的安装步骤。

Q: 软件提示获取证书
A: 由于现在的服务端已经升级成伽利略导航版本，所以使用时需要证书。小强用户可以根据提示联系客服免费获取。

小强主页 返回目录

- 小强ROS机器人教程(7)_使用ps3手柄控制小强移动
  - 操作步骤
  - 花样玩法

小强ROS机器人教程(7)_使用ps3手柄控制小强移动

小强主页

原理：本教程涉及3个包 ps3joy负责将ps 3 蓝牙接受信号转换成标准的linux设备(`/dev/input/js0`) joy_node节点负责将上述joy设备数据转换成ros中的joy数据类型 turtlebot_teleop_joy负责将上述joy数据topic 转换成小车运动指令 `/cmd_vel`

操作步骤

0.第一次使用手柄，需要将手柄与蓝牙接受器进行绑定，以后可以直接从步骤1开始

如果手柄和小强同时购买，那么我们已经提前为您将手柄配对可以从步骤1开始使用。 绑定方法参考原装和国产ps3手柄ros驱动程序中的“快速使用方法步骤1”

1.启动ps3joy，将ps3手柄与蓝牙接受器配比

```
#确保蓝牙接收器已经插入主机usb口
sudo bash
rosrun ps3joy ps3joyfake_node.py
```

正常会出现下面的提示

```
root@xiaoqiang-desktop:~# rosrun ps3joy ps3joyfake_node.py
No inactivity timeout was set. (Run with --help for details.)
Waiting for connection. Disconnect your PS3 joystick from USB and press the pairing button
.
```

如果提示下列错误 按下下图中的手柄配对键

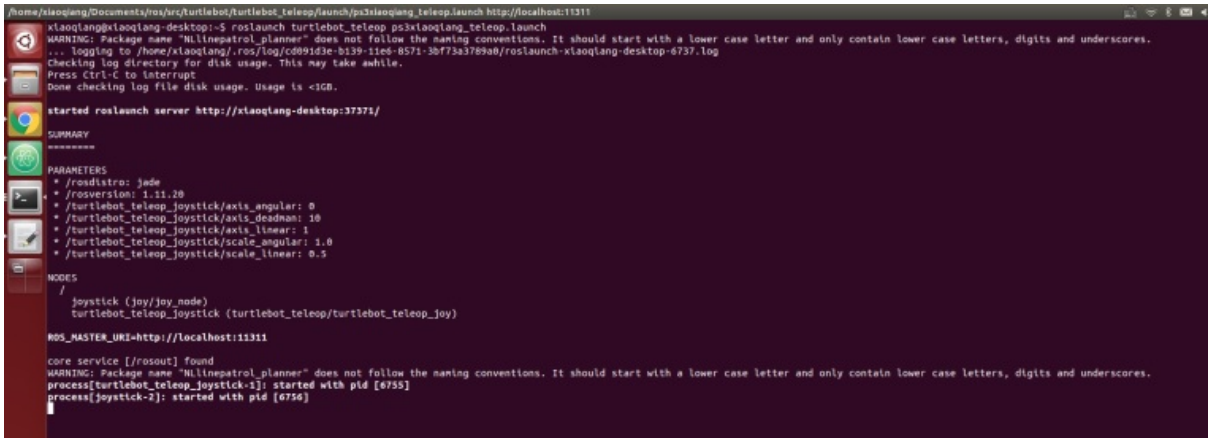
配对成功的话，上面的窗口会输出类似下面的结果

```
root@xiaoqiang-desktop:~# rosrun ps3joy ps3joyfake_node.py
No inactivity timeout was set. (Run with --help for details.)
Waiting for connection. Disconnect your PS3 joystick from USB and press the pairing button
.
Connection activated
```

2.启动joy_node和turtlebot_teleop_joy

```
roslaunch turtlebot_teleop ps3fakexiaoqiang_teleop.launch
```

正常启动后如下图所示



ps3fakexiaoqiang_teleop.launch文件内容如下

```
<launch>

  <node pkg="turtlebot_teleop" type="turtlebot_teleop_joy" name="turtlebot_teleop_joystick"
>
    <param name="scale_angular" value="0.4"/>
```

```
    <param name="scale_linear" value="0.4"/>
    <param name="axis_deadman" value="10"/>
    <param name="axis_linear" value="1"/>
    <param name="axis_angular" value="0"/>
    <param name="axis_enbar" value="12"/>
    <param name="axis_disenbar" value="14"/>
    <remap from="turtlebot_teleop_joystick/cmd_vel" to="/cmd_vel"/>
    <remap from="turtlebot_teleop_joystick/joy" to="/joy"/>
  </node>
</launch>
```

上述launch文件中的参数分别对应直线速度最大值(scale_linear)和角速度最大值(scale_angular)，油门离合键(axis_deadman)、前进后退轴(axis_linear)、左右转轴(axis_angular)，底盘红外使能键(axis_enbar)，底盘红外关闭键(axis_disenbar)，这些控制按键、摇杆的映射关系。

3.保持按住手柄油门键（下图中的１０号键），现在使用左侧的推杆可以控制小车的前后移动和转向（下图中的１＋１－摇杆）

SONY
START
SELECT
DUALSHOCK 3
SIXAXIS
1 2 3 4
11
10
9
8
18

SONY
SELECT
START
1+
0+
0-
1-
3+
2+
2-
3-
17
16
19

根据这些按键编号，可以修改launch文件中的相关参数从而改变按键映射关系

花样玩法

购买ps3手机支架, 安卓手机装上小强图传app,这样可以实现图传遥控

小强主页 返回目录

- 小强ROS机器人教程(8)___kinect1代 ROS驱动测试与安装
    - 1.libfreenect测试
    - 2.ROS驱动测试
    - 3.下文将介绍kienct1代的ros驱动安装步骤
    - 需要安装三个软件：
    - a. libfreenect
    - b. 安装rgbd_launch
    - c. 安装freenect_stack
    - d. 驱动安装完成，现在可以在ROS中使用kinect了，例如在rviz中观看kinect输出的点云,参考本节开头的步骤2

小强ROS机器人教程(8)___kinect1代 ROS驱动测试与安装

小强主页

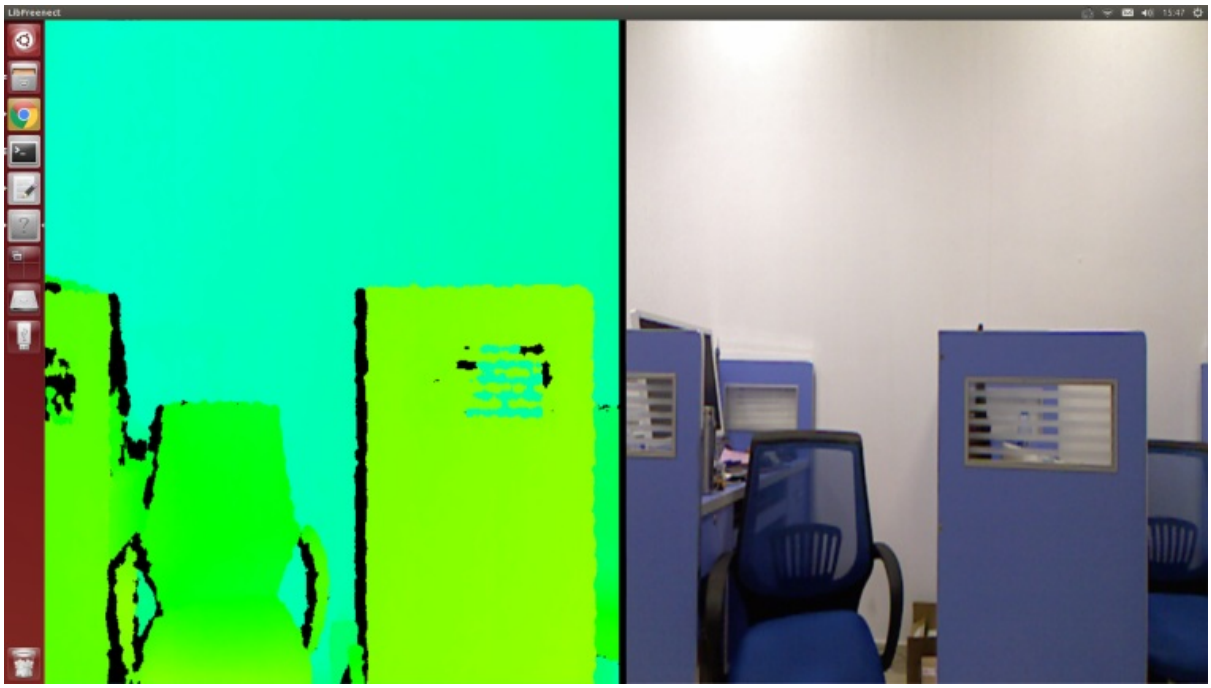
小强底盘输出一个12v电源（DC头，贴有“kinect供电”标签）用于kinect 供电。

1.libfreenect测试

将小强主机接入显示器和键盘，在小强主机上新开一个命令终端输入

```
freenect-glview
```

可以看到如下图的类似界面



2.ROS驱动测试

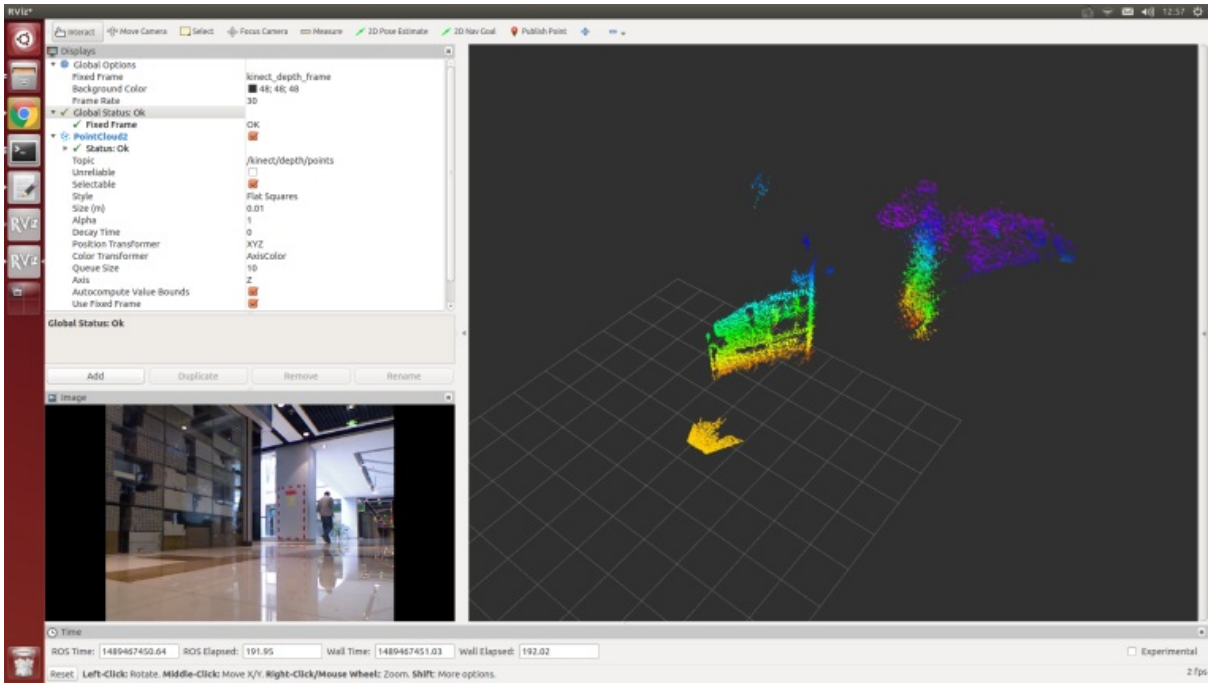
关闭步骤1中的程序，新开一个命令窗口，使用freenect_launch启动相关kinect节点

```
roslaunch freenect_launch  freenect-xyz.launch
```

新开1个窗口打开rviz

```
rviz
```

选择需要显示的内容，例如kinect的rgb图像和深度点云，显示效果如下



kienct各项功能的开启

在/home/xiaoqiang/Documents/ros/src/freenect_stack/freenect_launch/launch/examples/freenect-xyz.launch里面

```
<launch>
  <include file="$(find freenect_launch)/launch/freenect.launch">
    <arg name="camera"                          value="kinect" />
    <arg name="motor_processing"                value="true" />
    <arg name="audio_processing"                value="false" />
    <arg name="rgb_processing"                  value="true" />
    <arg name="ir_processing"                   value="false" />
    <arg name="depth_processing"                value="true" />
    <arg name="depth_registered_processing"     value="false" />
    <arg name="disparity_processing"            value="false" />
    <arg name="disparity_registered_processing" value="false" />
    <arg name="num_worker_threads"              value="4" />
  </include>
</launch>
```

通过设置true 或者false来开启、关闭相应功能

3.下文将介绍kienct1代的ros驱动安装步骤

2016年7月以后购买的用户不需要安装驱动，小强主机已经配置好kinect 1代驱动。

需要安装三个软件：

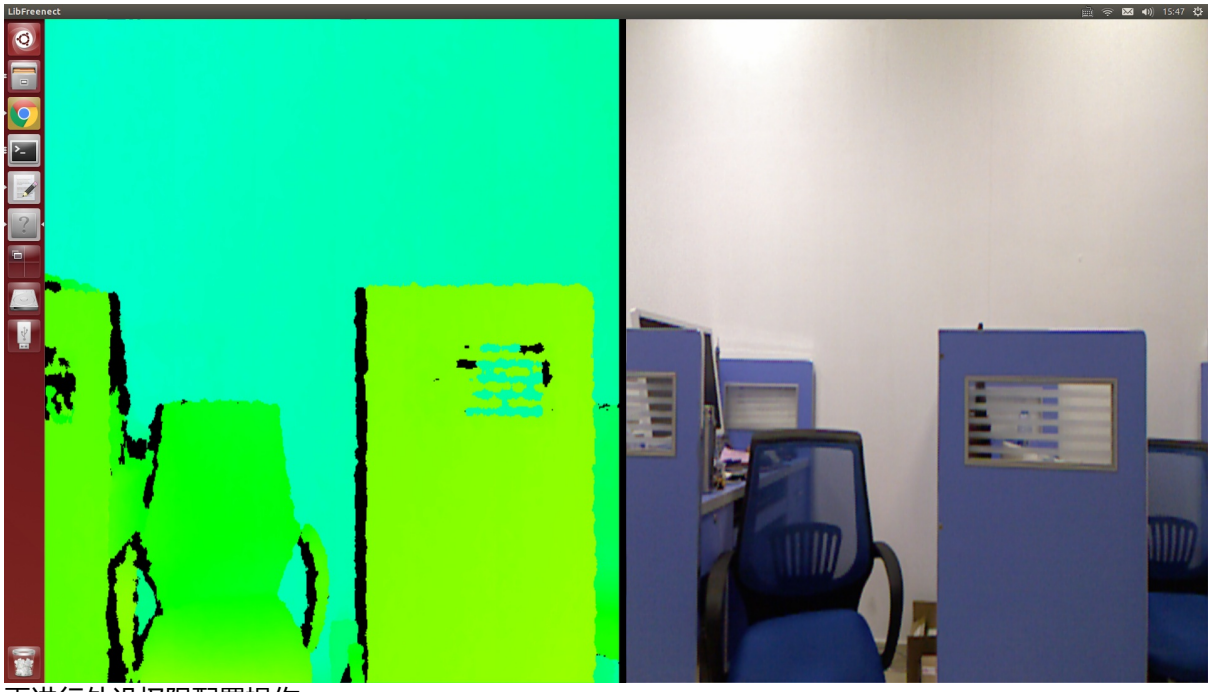
a.libfreenect b.rgbd_launch c.freenect_stack

a. libfreenect

先将kinect接入小强主机，然后打开一个命令行终端，输入下列代码

```
cd Documents

sudo apt-get install git-core cmake freeglut3-dev pkg-config build-essential libxmu-dev li
bxi-dev libusb-1.0-0-dev

git clone https://github.com/OpenKinect/libfreenect
cd libfreenect
mkdir build
cd build

# 重点来了，下面配置将使能kinect音频和解决安装路径问题
cmake .. -DCMAKE_INSTALL_RPATH:STRING="/usr/local/bin;/usr/local/lib" -DBUILD_REDIST_PACKA
GE=OFF

make
sudo make install
sudo ldconfig /usr/local/lib64/
sudo freenect-glview
```

现在应该可以看到kinect的输出图像了



再进行外设权限配置操作

```
sudo adduser $xiaoqiang video //请将xiaoqiang换成自己电脑的账户名
```

增加一个udev规则,先打开51-kinect.rules文件

```
sudo gedit /etc/udev/rules.d/51-kinect.rules
```

拷贝如下内容后保存退出

```
# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02b0", MODE="0666"
# ATTR{product}=="Xbox NUI Audio"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ad", MODE="0666"
# ATTR{product}=="Xbox NUI Camera"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02ae", MODE="0666"
# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02c2", MODE="0666"
# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02be", MODE="0666"
# ATTR{product}=="Xbox NUI Motor"
SUBSYSTEM=="usb", ATTR{idVendor}=="045e", ATTR{idProduct}=="02bf", MODE="0666"
```

注销用户后重进系统，现在就能直接启用kinect了，不用sudo了

```
freenect-glview
```

b. 安装rgbd_launch

rgbd_launch包含了驱动安装包openni_launch或 freenect_launch需要的通用launch文件。 主要有两个重要的launch文件：

1. processing.launch.xml：安装一系列nodelets去处理来自RGB-D driver（openni_camera or freenect_camera的数据，还可以设定参数简化处理nodelets图像。
2. kinect_frames.launch：为 Kinect安装tf tree。也可以从openni_launch or freenect_launch内部启动该文件。

rgbd_launch文件包含多个分散处理的launch文件。但只有 processing.launch.xml可以在外部修改使用。

```
cd ~/Documents/ros/src
git clone  https://git.bwbot.org/publish/rgbd_launch
cd ..
catkin_make
```

c. 安装freenect_stack

```
cd ~/Documents/ros/src
git clone  https://gitee.com/BlueWhaleRobot/freenect_stack
cd ..
catkin_make
```

d. 驱动安装完成，现在可以在ROS中使用kinect了，例如在rviz中观看kinect输出的点云,参考本节开头的步骤2

小强主页 返回目录

- 小强ROS机器人教程(9)_使用rostopic控制kinect的俯仰角度
    - 准备工作:
    - 操作步骤:

小强ROS机器人教程(9)_使用rostopic控制kinect的俯仰角度

小强主页

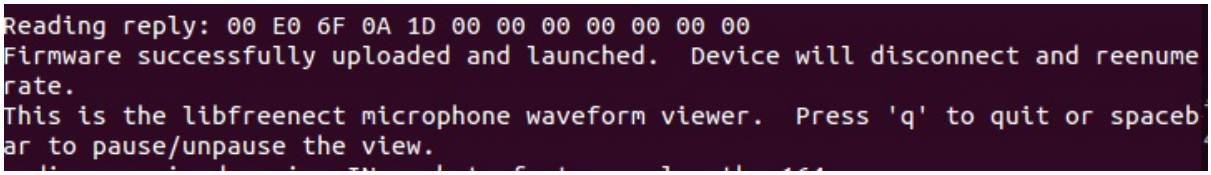
`本教程适用于2016年9月之后购买的用户，之前购买的用户请到我们开源仓库安装下载` freenect_stack，`替换教程(7)中的ros官方版freenect_stack`

准备工作:

请查看kinect版本，在kinect底座标签上有注明。对于部分model1473的用户，因为驱动的缺陷（不影响kinect其它功能,只涉及电机），需要先进行如下操作，model1414用户可以直接跳过。

```
# vnc登入小车主机
freenect-micview
```

如果出现下图，关闭上述命令，继续教程



操作步骤:

1.在本地虚拟机新开一个窗口，启动freenect_stack驱动,

```
# vnc登入小车主机
roslaunch freenect_launch freenect-xyz.launch
```

正常启动会出现下图，如果出现红色错误（驱动缺陷），请通过git pull更新freenect_stack程序。

```
[ INFO] [1477406854.865362860]: Number devices connected: 1
[ INFO] [1477406854.865453964]: 1. device on bus 000:00 is a Xbox NUI Camera (2
e) from Microsoft (45e) with serial id 'A70774703536333A'
[ INFO] [1477406854.866242209]: Searching for device with index = 1
[ INFO] [1477406855.502427696]: flushDevice

[ INFO] [1477406855.502642152]: Starting a 3s RGB and Depth stream flush.
[ INFO] [1477406855.503054171]: Opened 'Xbox NUI Camera' on bus 0:0 with serial
number 'A70774703536333A'
[ WARN] [1477406856.952469775]: Could not find any compatible depth output mode
for 1. Falling back to default depth output mode 1.
[ INFO] [1477406856.961696739]: rgb_frame_id = 'kinect_rgb_optical_frame'
[ INFO] [1477406856.961735786]: depth_frame_id = 'kinect_depth_optical_frame'
[ WARN] [1477406856.975615344]: Camera calibration file /home/xiaoqiang/.ros/ca
era_info/rgb_A70774703536333A.yaml not found.
[ WARN] [1477406856.975681775]: Using default parameters for RGB camera calibra
ion.
[ WARN] [1477406856.975777743]: Camera calibration file /home/xiaoqiang/.ros/ca
era_info/depth_A70774703536333A.yaml not found.
[ WARN] [1477406856.975820217]: Using default parameters for IR camera calibrat
on.
[ INFO] [1477406859.826134535]: Stopping device RGB and Depth stream flush.
```

2.在本地虚拟机新开一个窗口，发布电机角度控制命令

```
# vnc登入小车主机
rostopic pub /set_tilt_degree std_msgs/Int16 '{data: -20}' -1
```

如果一切正常，现在可以看到kinect的仰角不断变小，上述命令中的 `{data: -20}` 数字就代表角度，可以设置为30到-30之间的整数 `请将kinect数据线插入小车主机蓝色usb3.0端口，这样kinect才能正常工作`

小强主页 返回目录

- 小强ROS机器人教程(10)_使用kinect进行自主移动避障
  - 原理
  - 操作步骤

小强ROS机器人教程(10)_使用kinect进行自主移动避障

小强主页

`本教程适用于2016年9月20号之后购买且带kinect版的用户，对于其它用户，请联系、咨询技术支持`

先看最后效果图， 演示视频

00:00 / 00:00

原理

`freenct_stack` 包提供kinect驱动，其发布的点云通过 `image_pipeline` 转换成障碍物栅格分布图。 `nav_test` 软件包启动底盘导航程序后会自动处理分析障碍物分布图，之后根据rivz发布的目标导航点自主移动。

操作步骤

`注意在插入kinect的时候先插入在小强主机上插入kinect usb接口。等待kinect灯亮再插入kinect供电口。否则可能会烧毁主机和kinect`

在第一个机器人命令行窗口启动kinect驱动

对于kinect一代

```
roslaunch freenect_launch kinect-xyz.launch
```

对于kinect二代

```
roslaunch kinect2_bridge kinect2-xyz.launch
```

对于奥比中光astrapro

```
roslaunch astra_launch astrapro.launch
```

对于intel RealDense D400系列

```
roslaunch realsense2_camera rs_camera_xiaoqiang.launch
```

在第二个机器人命令行窗口设置kinect俯仰角，这个角度不是任意的

对于kinect一代

```
rostopic pub /set_tilt_degree std_msgs/Int16 '{data: -19}' -1
```

对于kinect二代

因为kinect二代没有俯仰角电机，因此请手动把kinect掰到最大俯角位置

对于奥比中光astrapro

因为astrapro没有俯仰角电机，因此请手动把astrapro掰到最大俯角位置

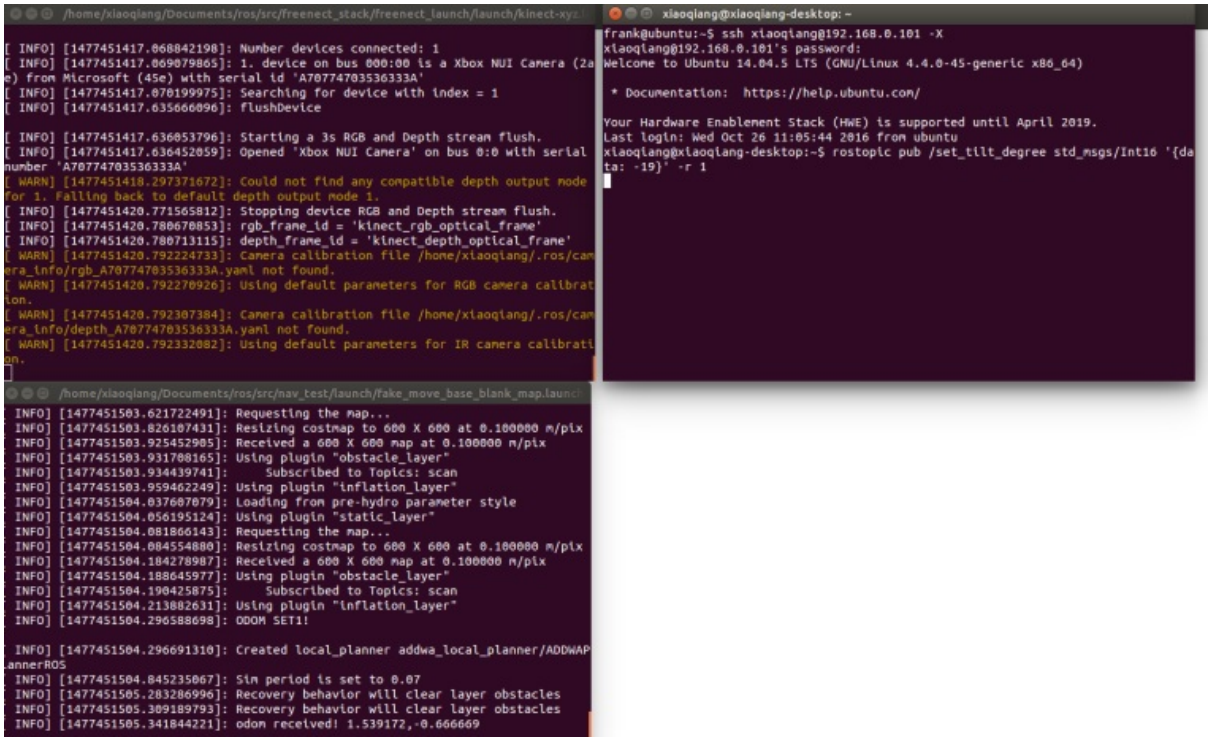
对于intel RealDense D400系列

因为RealDense D400系列没有俯仰角电机，配带的支架已经预设好角度，因此不用任何操作。

在第三个机器人命令行窗口启动底盘导航程序

```
roslaunch nav_test fake_move_base_blank_map.launch
```

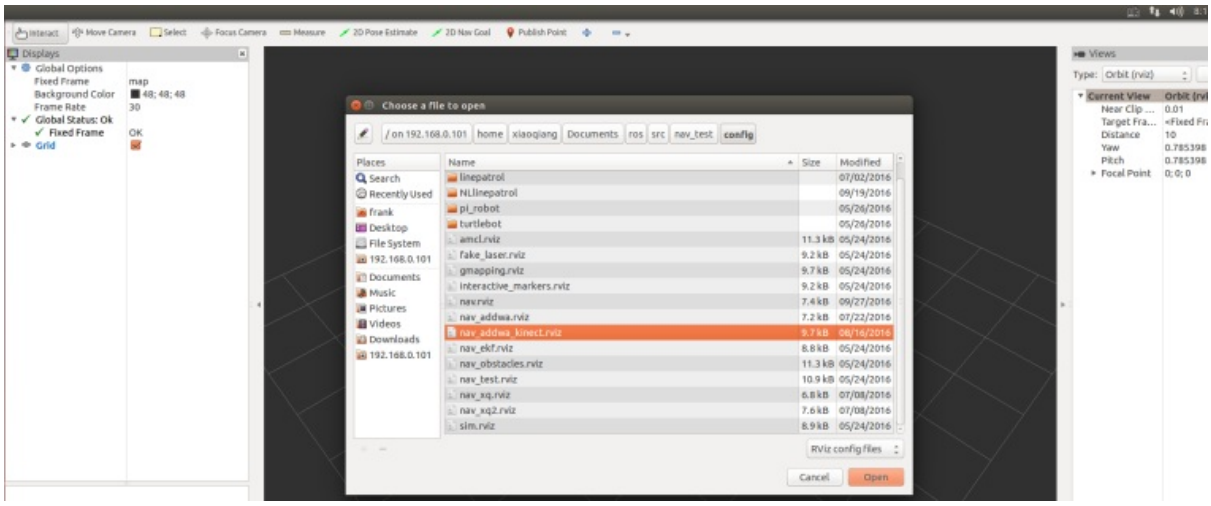
全部正常，会出现类似下图的界面。



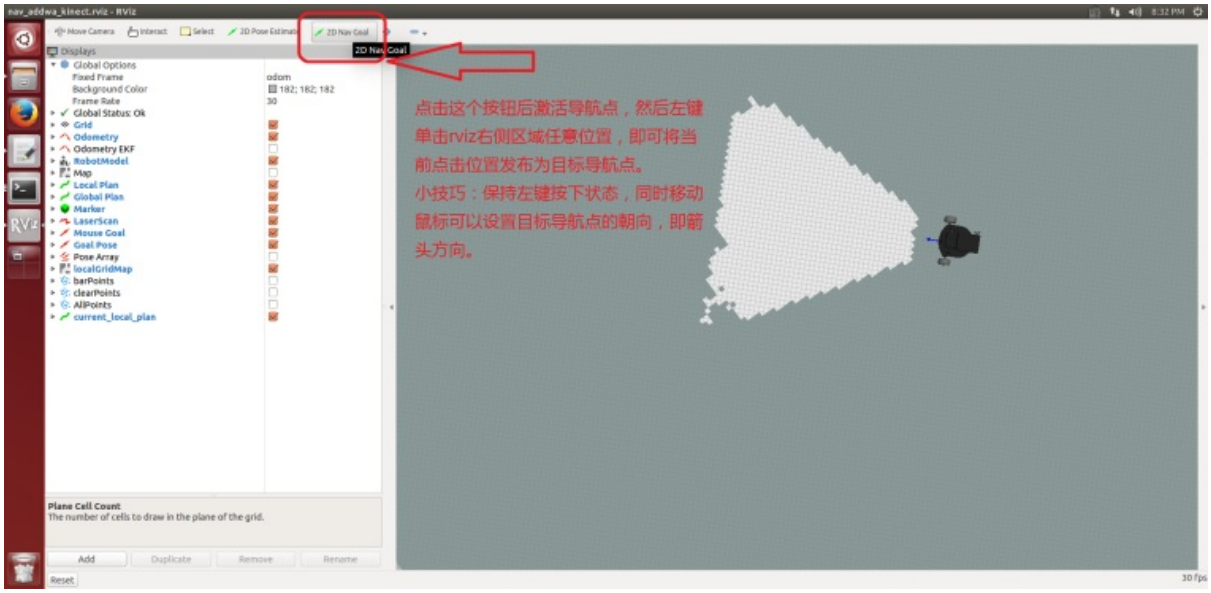
在第四个机器人命令行窗户启动rviz

```
rviz
```

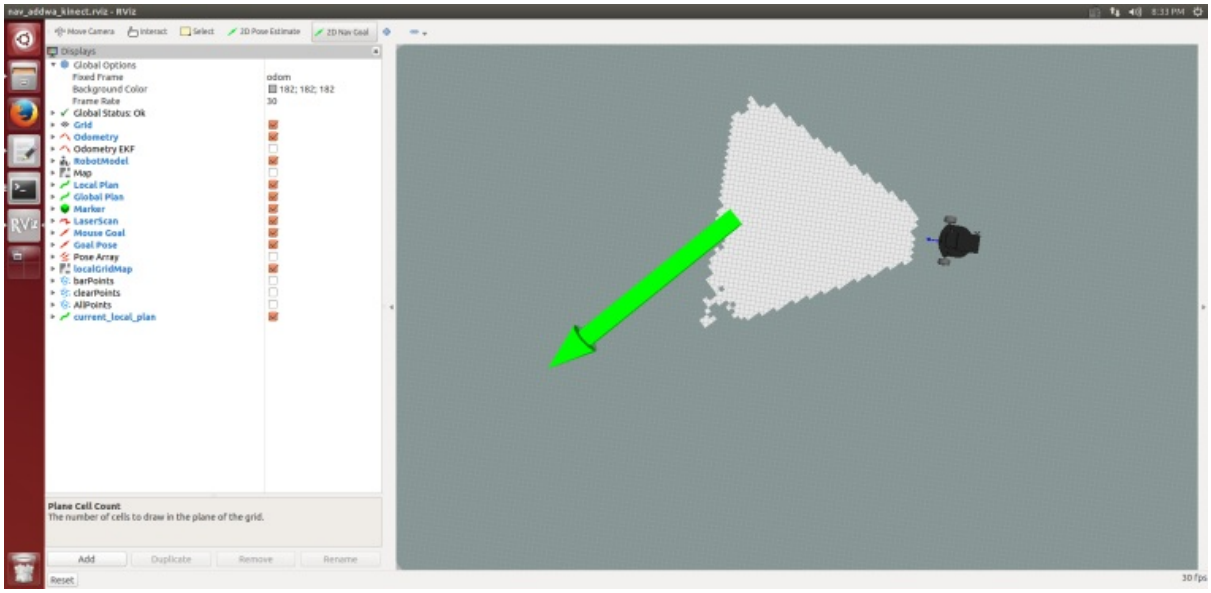
点击rviz界面左上角的open config，选择小车主机上的/home/xiaoqiang/Documents/ros/src/nav_test/config/nav_addwa_kinect.rviz配置文件



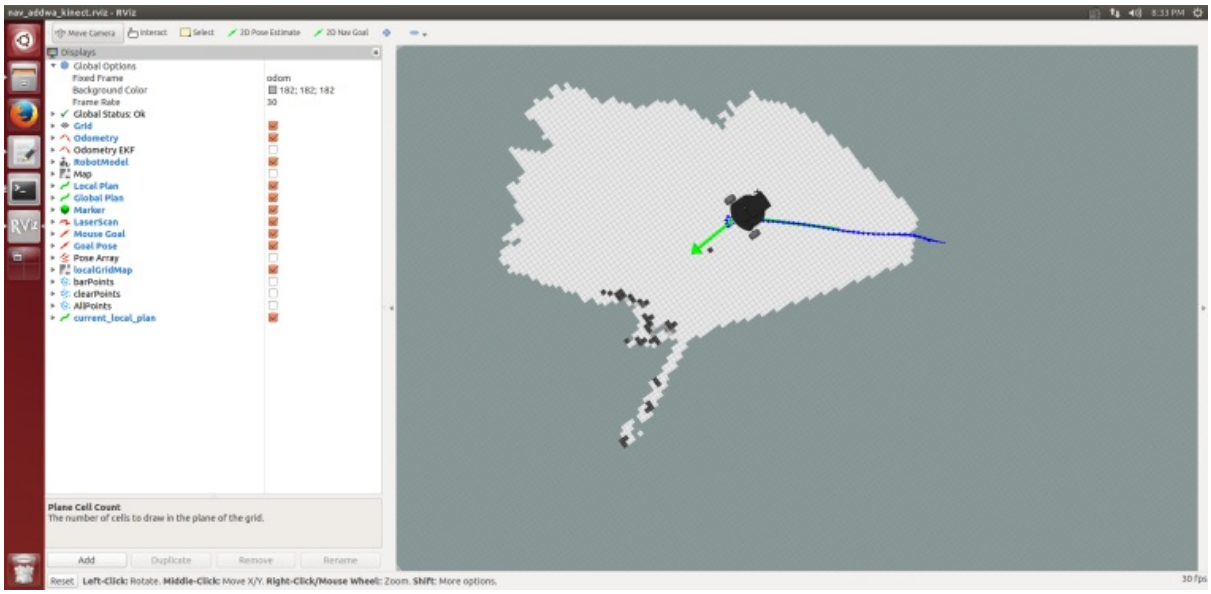
正常的话，现在rviz中将出现类似下图的画面，现在所有配置都已经完成，开始发布导航目标点



任意发布一个目标点，小车会开始自主移动



小车到达目标点，请继续尝试其它位置，本教程结束



请将kinect数据线插入小车主机蓝色usb3.0端口，才能保证驱动的正常工作。型号为1473的kinect用户请先看教程（9）中的准备办法，在成功发布角度之后（kinect低头）需要关闭这个角度发布命令，不关闭会导致kinect深度点云数据没有输出。

如果小强一直后退，rviz中可以看到小车前面有一块黑色区域，黑色区域一直跟随小车移动。这种情况说明kinect安装或者kienct低头角度没有到位，请检查kinect安装支架、去除kienct贴膜后重新执行教程中的指令

注意此处避障时的建图由于没有使用map_server,所以并不能保存出来。如果想要建立地图需要用其他的算法。

小强主页 返回目录

- 小强ROS机器人教程(11)___kinect跟随包turtlebot_follower
  - 1.安装软件包
  - 2.保证小强前方2米*2米范围空旷无杂物，先检查kinect角度是否水平，如果不处于水平状态请先关闭小车底盘电源，在小强主机上启动turtlebot_follower包
  - 3.此时人进入小强前方视野，会激活小强的跟随功能，小强开始跟随人的移动而移动
  - 常见问题：

小强ROS机器人教程(11)___kinect跟随包turtlebot_follower

小强主页

`注：2017年3月15日之后购买的用户请跳过安装步骤，直接从步骤2开始测试`

turtlebot_follower利用深度摄像头反馈的点云图，计算一定区域内的点云中心坐标作为目标跟随点，根据这个坐标和设置的安全距离控制底盘移动实现跟随功能。

下图小强识别出人的双腿后，在人腿位置设置了一个红球作为目标点

00:00 / 00:00 00:00 / 00:00

1.安装软件包

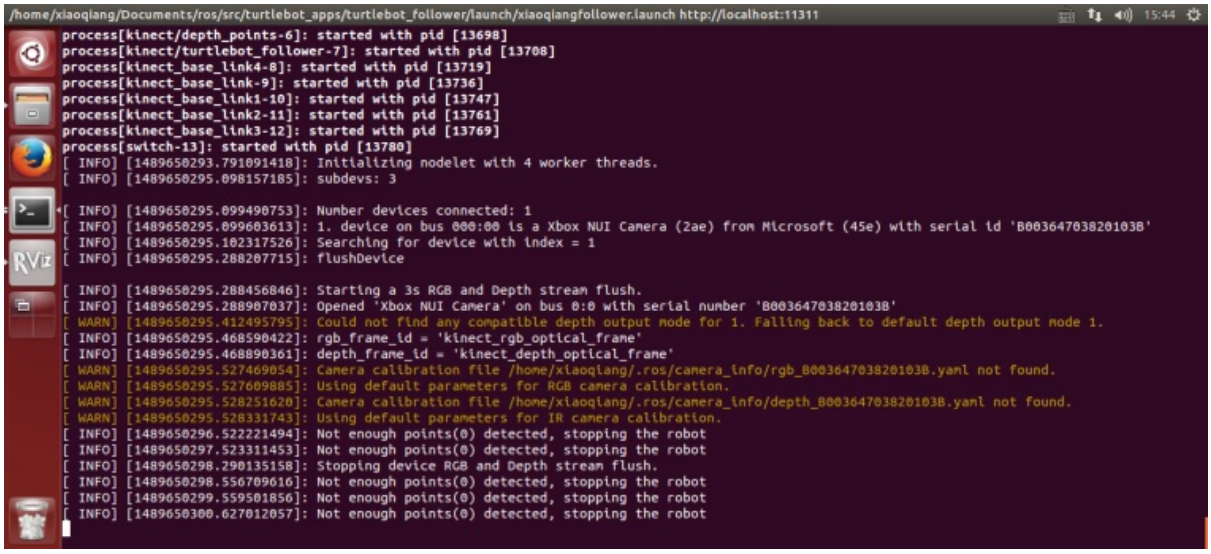
VNC登入小强主机后，进入ros工作空间，下载安装测试软件包

```
cd Documents/ros/src/
git clone https://github.com/turtlebot/turtlebot_msgs.git
git clone https://github.com/BlueWhaleRobot/turtlebot_apps.git
cd ..
catkin_make
```

2.保证小强前方2米*2米范围空旷无杂物，先检查kinect角度是否水平，如果不处于水平状态请先关闭小车底盘电源，在小强主机上启动turtlebot_follower包

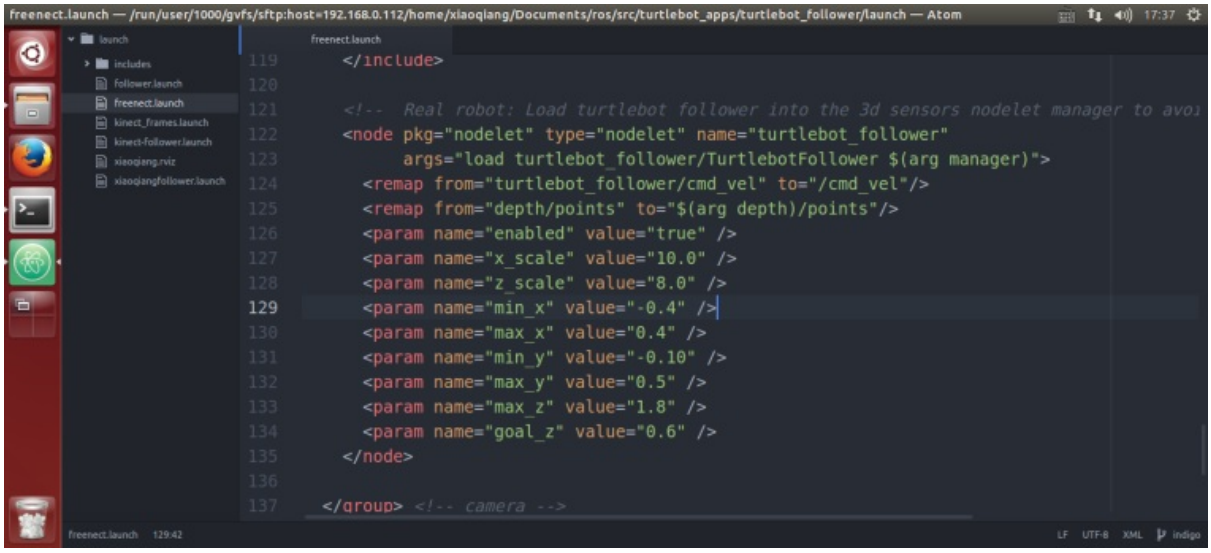
```
roslaunch turtlebot_follower xiaoqiangfollower.launch #kinect设备使用这个launch
roslaunch turtlebot_follower xiaoqiangfollower_realsense.launch #realsense设备则使用这个launch
```

正常启动后会出现下图



启动后，脚本会发布电机角度控制命令,保持kienct处于水平姿态，再打开小车底盘电源。

3.此时人进入小强前方视野，会激活小强的跟随功能，小强开始跟随人的移动而移动



常见问题：

1.开启kinect后没有响应，提示no points found之类，检查kinect型号是不是model 1473, 如果是1473需要先运行一遍freenect-micview程序激活一下固件。 小强主页 返回目录

- 小强ROS机器人教程(12)___ROS显示kinect2代的点云

小强ROS机器人教程(12)___ROS显示kinect2代的点云

小强主页

本教程适用于2016年12月之后购买的小强用户，在此之前购买的用户请自行根据这篇教程安装kinect2代驱动。

1启动kienct2代的ROS驱动

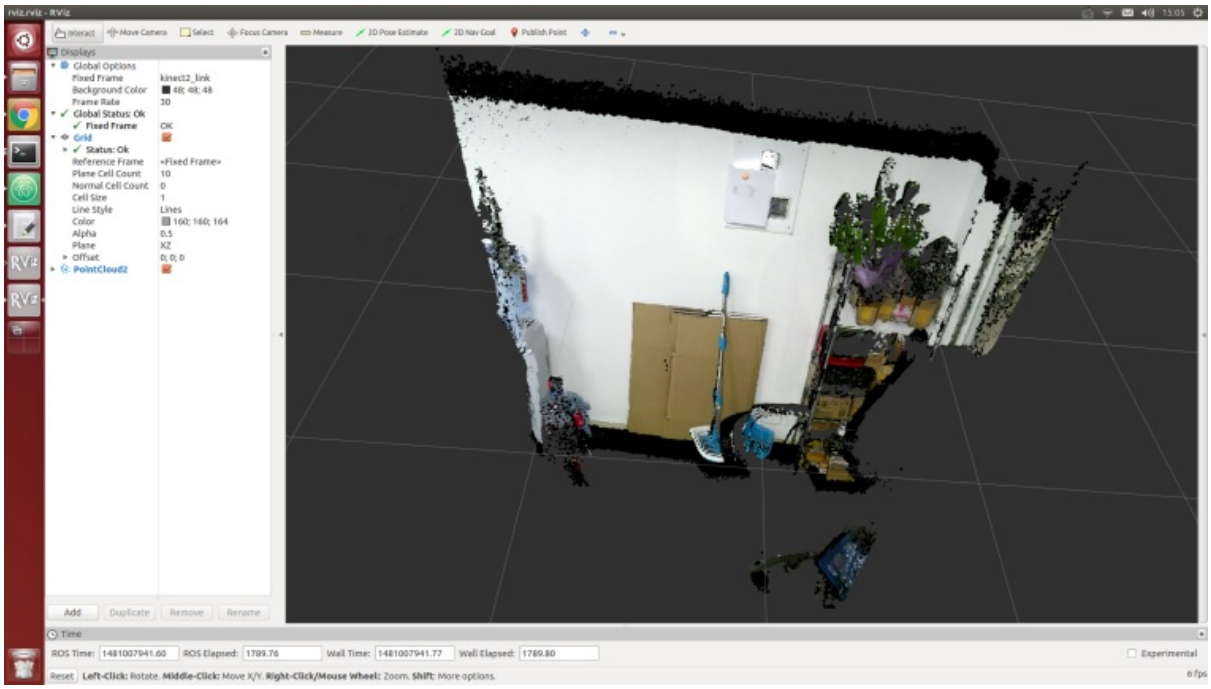
小强底盘输出一个12v电源（DC头，贴有“kinect供电”标签）用于kinect 供电，kinect2代需要插入小车主机蓝色usb 3 .0接口。 将小强主机接入显示器和键盘，在小强主机上打开一个命令行终端输入

```
roslaunch kinect2_bridge kinect2-xyz.launch
```

2 新开1个命令行终端，启动rviz

```
rviz
```

打开这个rviz配置文件 `/home/xiaoqiang/Documents/ros/src/iai_kinect2/kinect2_bridge/launch/rviz.rviz` ,点击可下载本文件 一切正常的话，可以出现类似下图的界面



小强主页 返回目录

- 小强ROS机器人教程(13)___rplidar二代激光雷达的使用和利用udev给小车增加串口设备
  - １．查看各个串口设备的ID

小强ROS机器人教程(13)___rplidar二代激光雷达的使用和利用udev给小车增加串口设备

小强主页

本节教程的1、2、3、4步骤仅用于演示给小强增加串口设备的方法，小强用户请接上硬件后直接跳到步骤5测试雷达。 小车主机和底盘的通信是通过串口实现的，在实际开发过程中我们可能会给小车增加串口外设，这会导致串口号（ttyUSB*）的混乱,引发小车底盘ros驱动和串口设备的异常。下文将以rplidar二代激光雷达为例，演示通过修改udev文件指定设备串口号的方式解决串口冲突问题。本文方法出处

１．查看各个串口设备的ID

```
sudo apt-get install expect-dev
unbuffer udevadm monitor --environment | grep 'ID_SERIAL='
```

将底盘通信ｕ转串重新插拔一下，终端会打印出此设备的ＩＤ信息，例如下图中的"Prolific_Technology_Inc._USB-Serial_Controller" 再将激光雷达的usb适配器重新插入主机，终端也会打印出激光雷达的ＩＤ信息，例如下图中的"Silicon_Labs_CP2102_USB_to_UART_Bridge_Controller_0001"

2．根据获取的串口设备的ID，建立udev规则文件，将底盘通信ｕ转串指定为ttyUSB001，将激光雷达指定为ttyUSB002

```
sudo gedit  /etc/udev/rules.d/60-persistent-serial.rules
```

输入下面内容后保存,请将文中ID_SERIAL后面的字符串换成步骤１中获取的ID

```
ACTION!="add", GOTO="persistent_serial_end"
SUBSYSTEM!="tty", GOTO="persistent_serial_end"
KERNEL!="ttyUSB[0-9]*", GOTO="persistent_serial_end"

# This is old 11.10 style: IMPORT="usb_id --export %p"
IMPORT{builtin}="path_id"
```

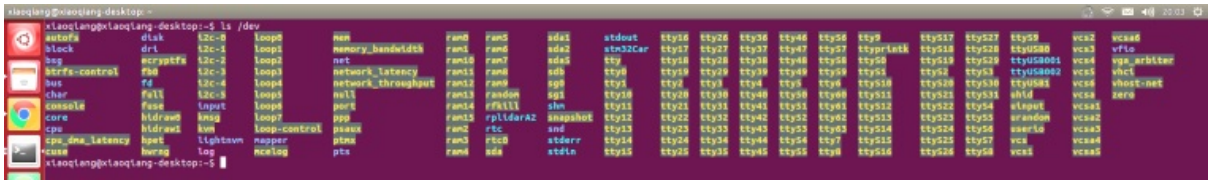
```
ENV{ID_SERIAL}=="Prolific_Technology_Inc._USB-Serial_Controller"         , SYMLINK="stm32C
ar"        , SYMLINK+="ttyUSB001" , OWNER="xiaoqiang"
ENV{ID_SERIAL}=="Silicon_Labs_CP2102_USB_to_UART_Bridge_Controller_0001" , SYMLINK="rplida
rA2"       , SYMLINK+="ttyUSB002" , OWNER="xiaoqiang"

LABEL="persistent_serial_end"
```

更新系统udev规则

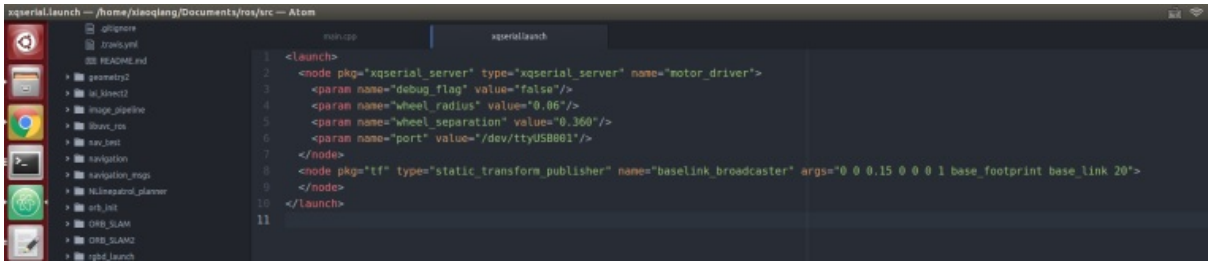
```
sudo udevadm control --reload
```

重新插拔所有usb串口设备,现在底盘通信u转串成功被识别为ttyUSB001、激光雷达被识别为ttyUSB002，与设备插入顺序和端口无关。

```
ls /dev
```

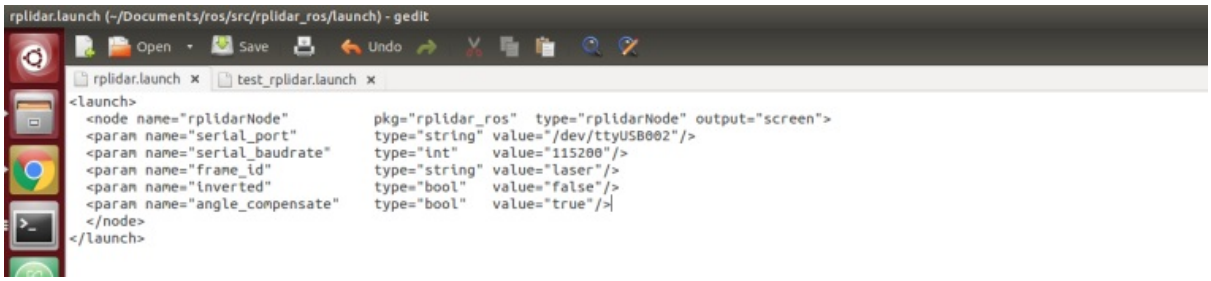


3．修改小车底盘ROS驱动节点launch文件，将通信设备指定为上文设置的ttyUSB001



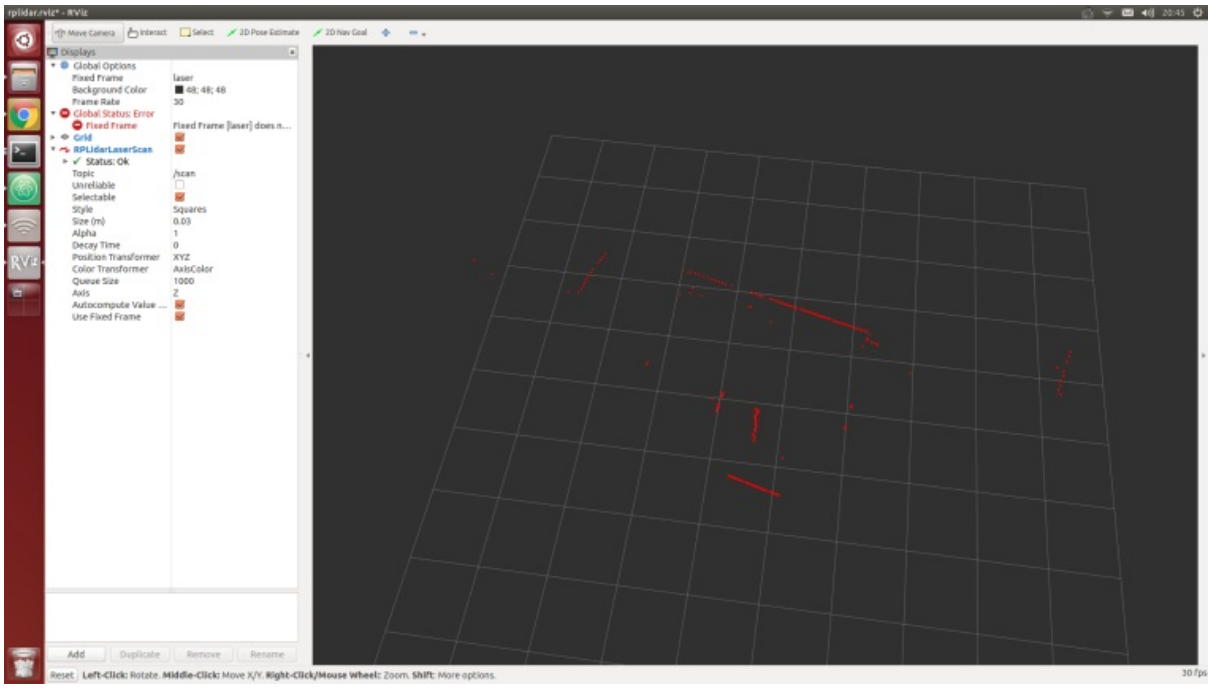
4．修改rplidar二代激光的ROS驱动节点launch文件，将通信设备指定为上文设置的ttyUSB002

```
驱动的安装

cd ~/Documents/ros/src
git clone https://git.bwbot.org/publish/rplidar_ros
cd  ..
catkin_make
```



5.重启小车，现在已经可以同时正常使用激光雷达和小车底盘，例如运行下述命令测试激光雷达

```
roslaunch rplidar_ros view_rplidar.launch
```



小强主页 返回目录

- 小强ROS机器人教程(14)_在gmapping下使用激光雷达进行建图
  - 1、安装gmapping，2017年3月3日之后收到货的用户可以跳过这个安装步骤，直接从步骤2开始
  - 2、启动gmapping节点
  - 3、遥控小强运动开始建图
  - 4、保存地图，本文结束

小强ROS机器人教程(14)_在gmapping下使用激光雷达进行建图

小强主页

1、安装gmapping，2017年3月3日之后收到货的用户可以跳过这个安装步骤，直接从步骤2开始

vnc登录小强主机,进入小强ros工作空间

```
cd Documents/ros/src/
```

下载两个ros软件包gmapping、openslam_gmapping

```
git clone https://gitee.com/bluewhaleRobot/slam_gmapping
git clone https://gitee.com/bluewhaleRobot/openslam_gmapping
```

编译、完成安装

```
cd ..
catkin_make
```

2、启动gmapping节点

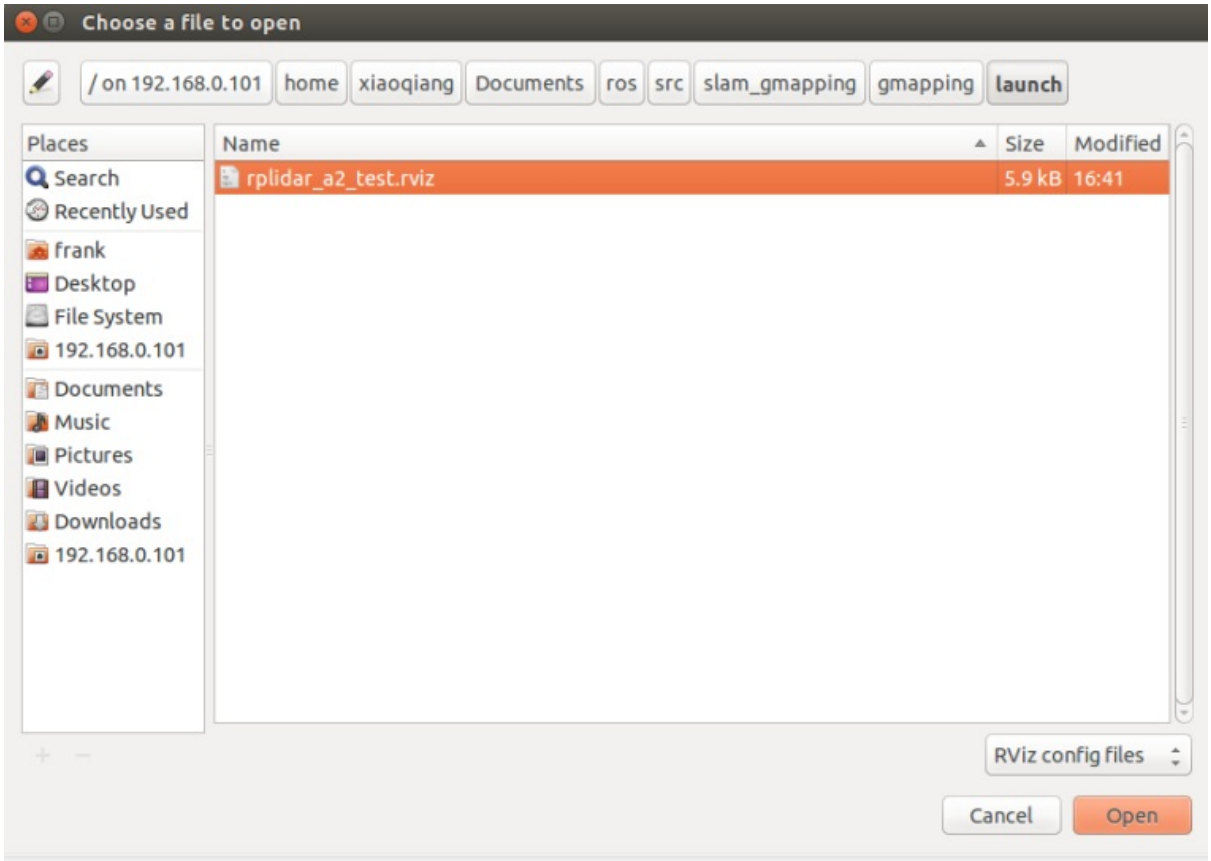
确保雷达安装正确，vnc进入小强主机后启动gmapping中的launch文件

```
如果是rplidar a1或者a2
roslaunch gmapping slam_gmapping_xiaoqiang_rplidar_a2.launch
# 如果是rplidar a3则启动
roslaunch gmapping slam_gmapping_xiaoqiang_rplidar_a3.launch
# 如果是星秒pavo激光雷达
roslaunch gmapping slam_gmapping_xiaoqiang_pavo.launch
# 如果是镭神ls01b激光雷达
roslaunch gmapping slam_gmapping_xiaoqiang_ls01b.launch
# 如果是sc-mini激光雷达
```

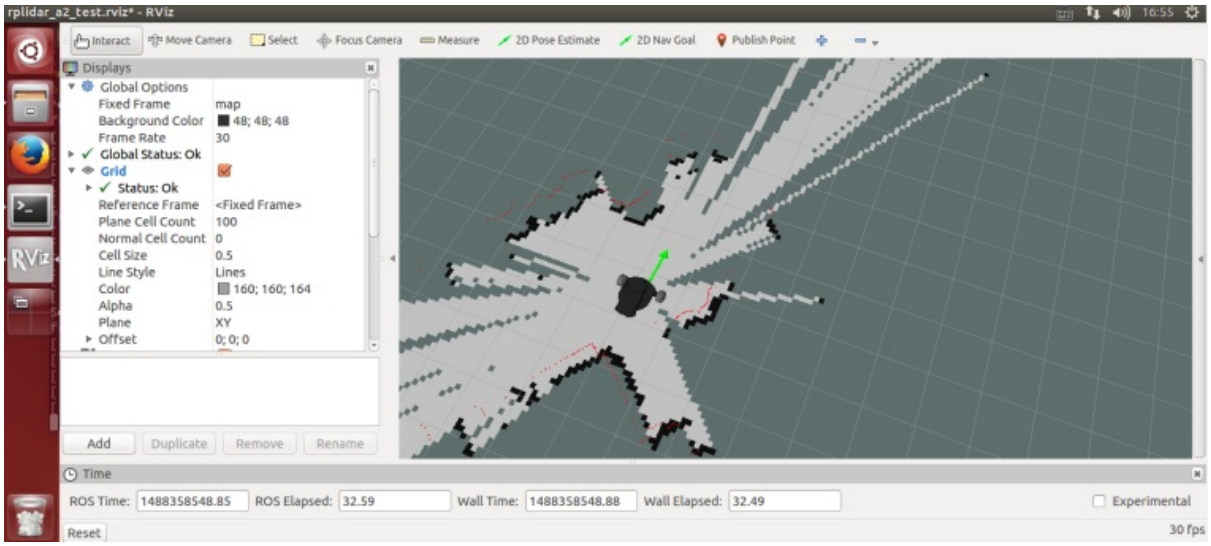
```
roslaunch gmapping slam_gmapping_xiaoqiang_scmini.launch
```

机器人打开rviz，选择打开小强ros工作目录下的slam_gmapping/gmapping/launch/rplidar_a2_test.rviz配置文件

```
rviz
```



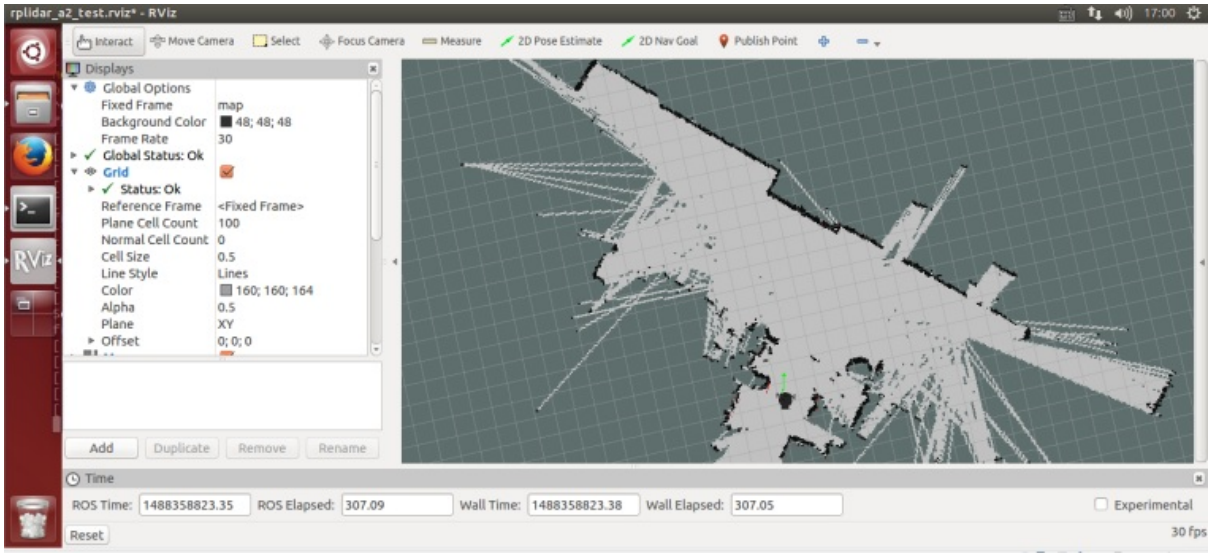
等待几秒，正常情况会出现下图的类似结果



3、遥控小强运动开始建图

第一种方式，使用windows遥控端，参考这篇帖子(小强图传遥控windows客户端)

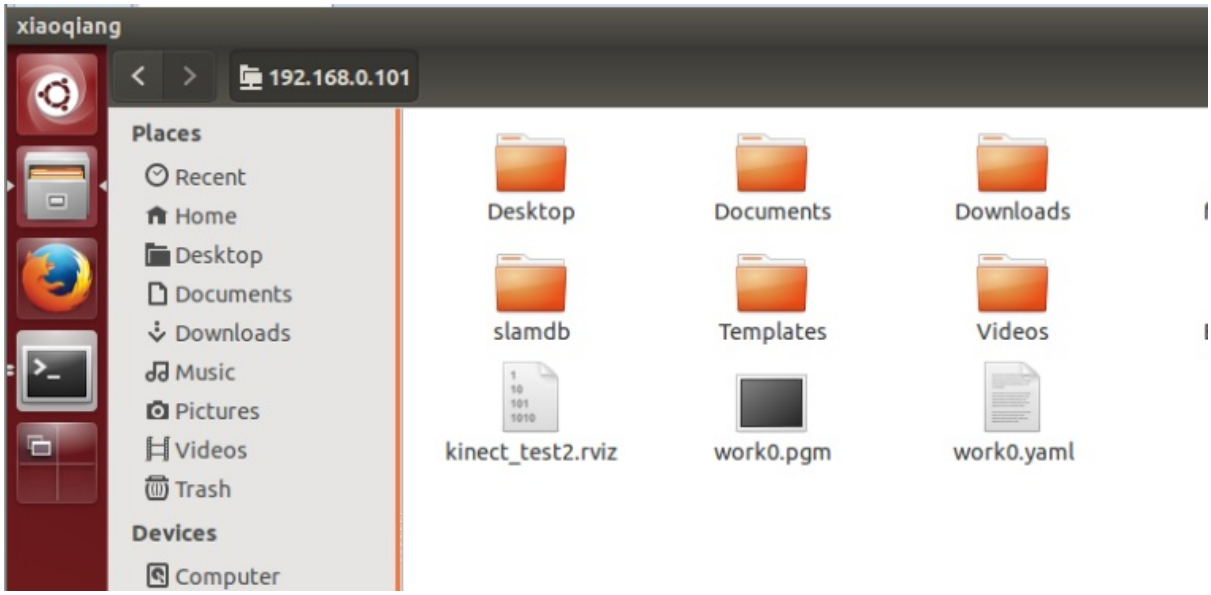
第二种方式，使用手机app,参考这篇贴子（小强手机遥控app安卓版）



4、保存地图，本文结束

vnc登录小强，在小强home目录下保存为work0开头的文件

```
rosrun map_server map_saver -f work0
```



小强主页 返回目录

- 小强ROS机器人教程(15)___AMCL导航测试

小强ROS机器人教程(15)___AMCL导航测试

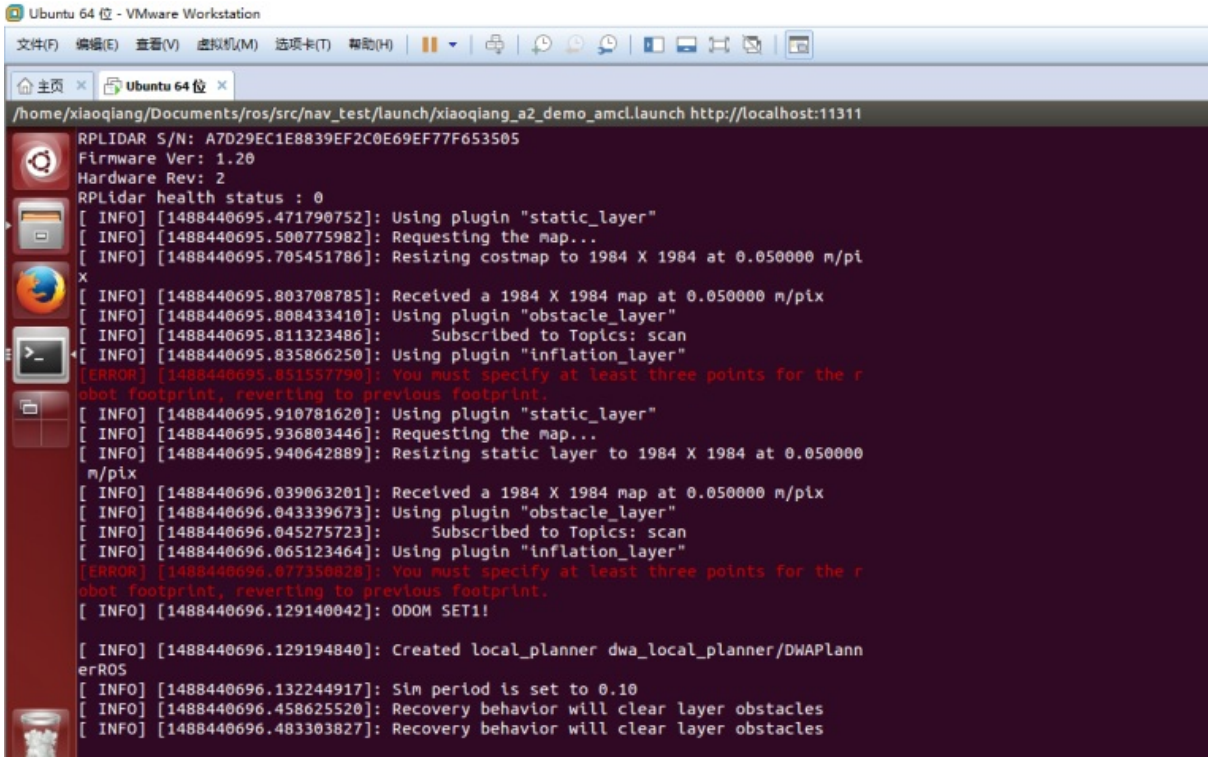
小强主页

下文将演示AMCL导航操作，使用rplidar a2作为scan输入，教程14中建立的地图文件作为全局map

启动导航节点 先将教程14中建立的两个map文件拷贝到 `/home/xiaoqiang/Documents/ros/src/nav_test/maps/` 下，覆盖同名文件即可

```
# 对于rplidar a1或者a2
roslaunch nav_test xiaoqiang_a2_demo_amcl.launch
# 对于rplidar a3
roslaunch nav_test xiaoqiang_a3_demo_amcl.launch
# 对于星秒pavo激光雷达
roslaunch nav_test xiaoqiang_pavo_demo_amcl.launch
# 对于镭神ls01b激光雷达
roslaunch nav_test xiaoqiang_ls01b_demo_amcl.launch
# 对于sc-mini激光雷达
roslaunch nav_test xiaoqiang_scmini_demo_amcl.launch
```

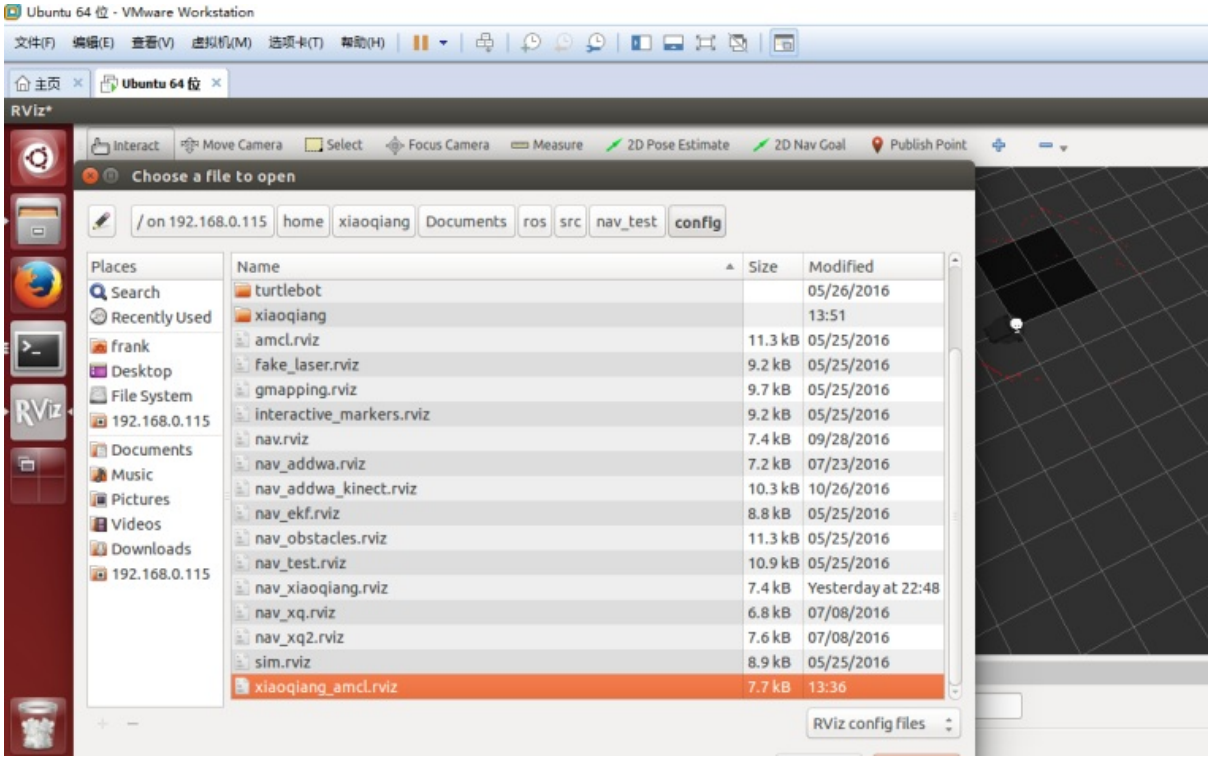
正常会出现下图的类似结果，同时雷达开始旋转



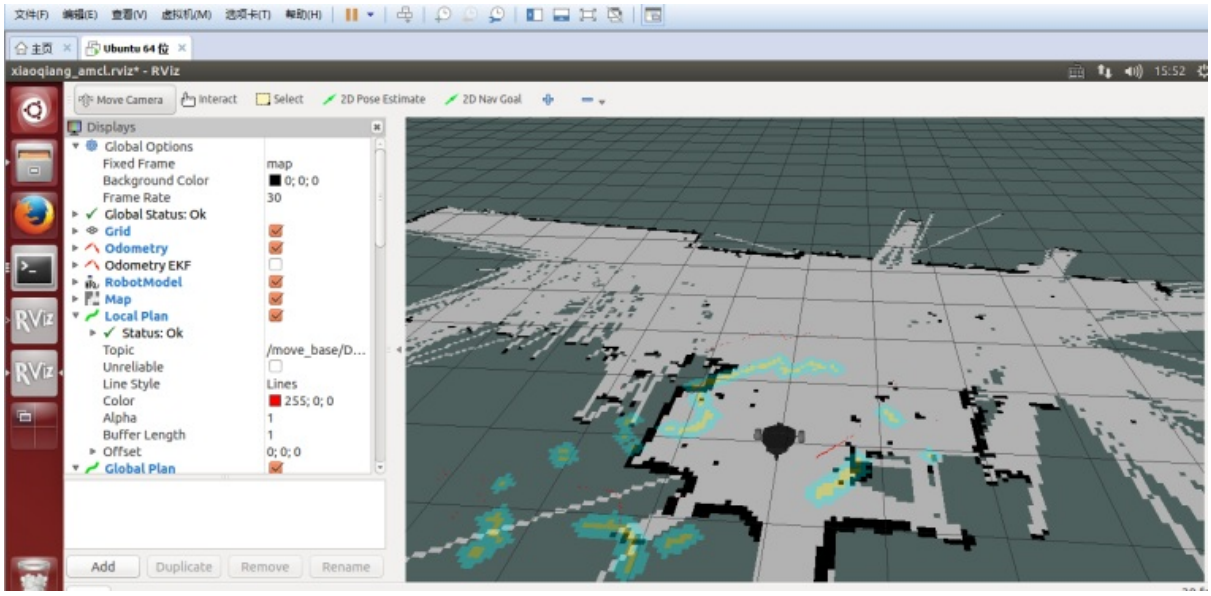
三、打开操作客户端

1.在机器人中启动rviz,选择打开小强ros工作目录下的 `nav_test/config/xiaoqiang_amcl.rviz` 配置文件

```
rviz
```

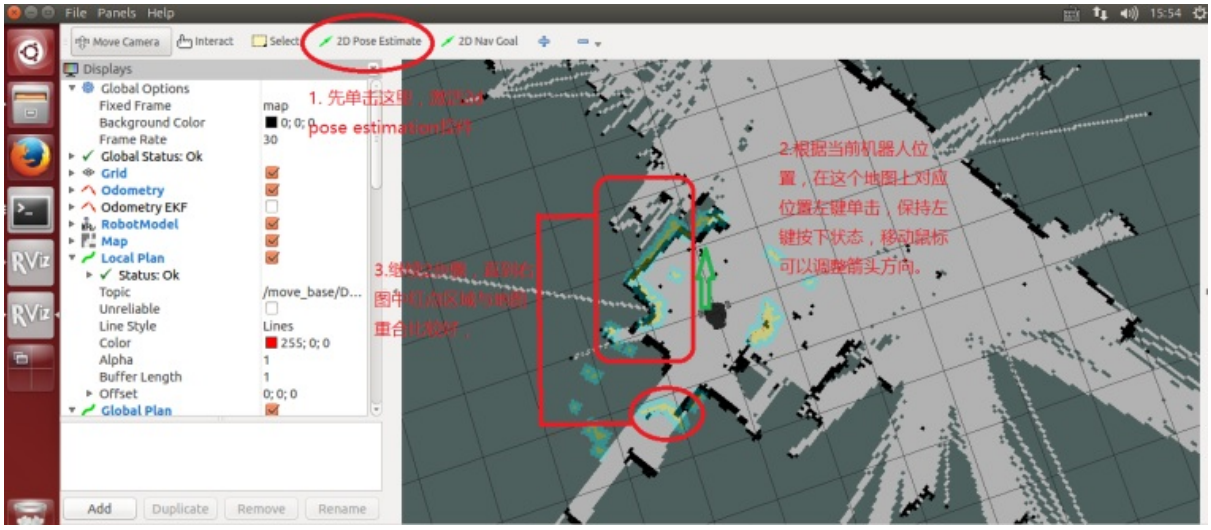


2.等待几秒后，rviz正常会出现类似下图的界面

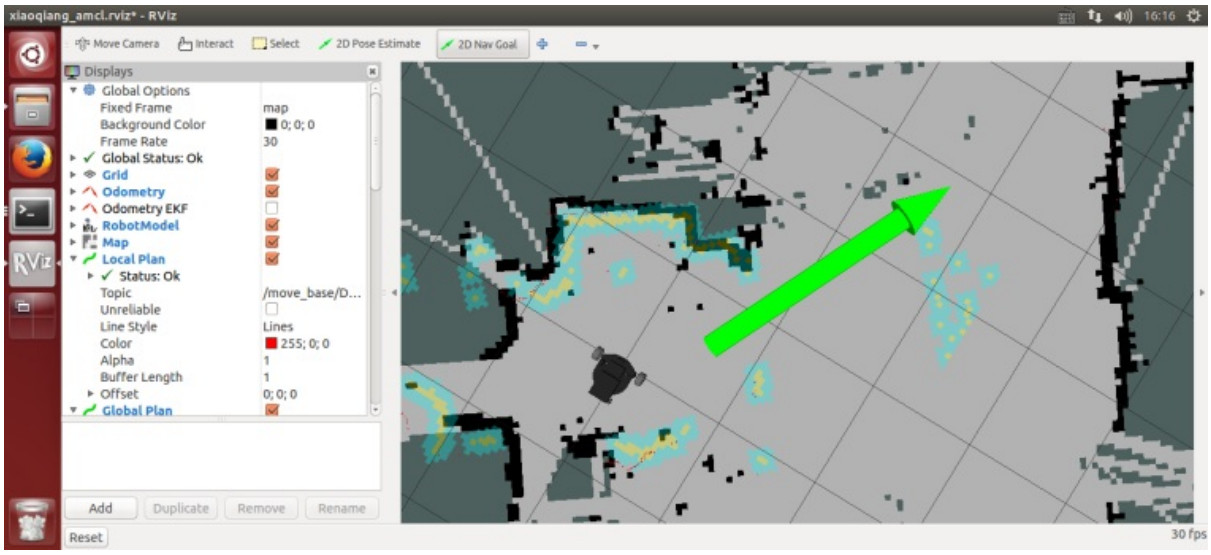


四、开始导航测试

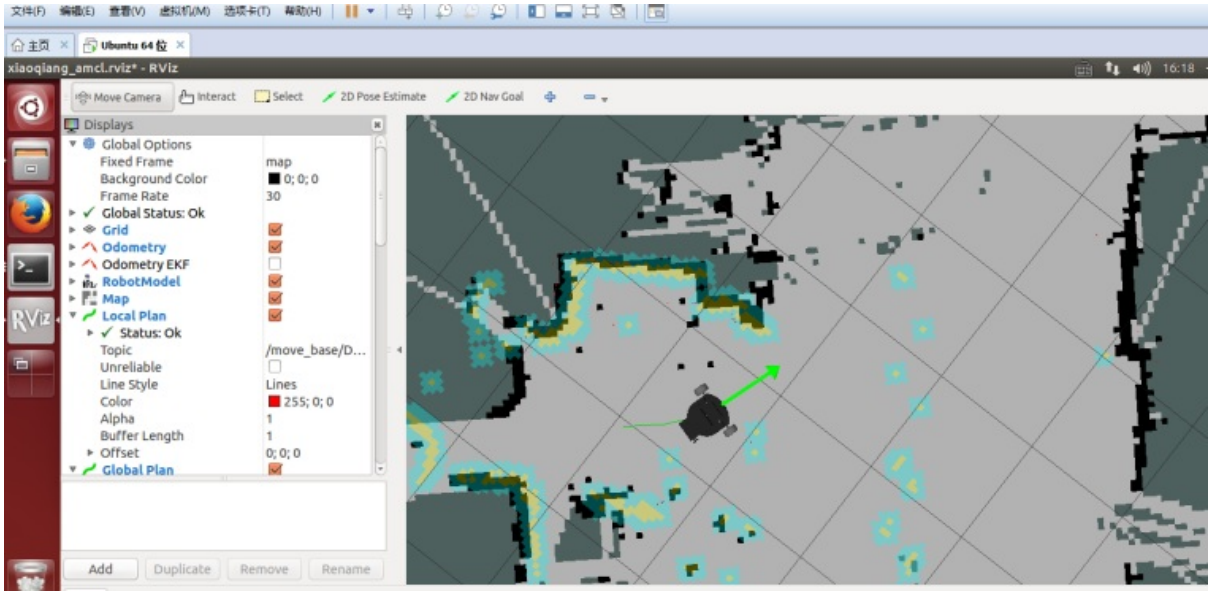
1.在rviz中使用2D Pose Estimation设置机器人的初始pose在map中的位置，因为AMCL算法需要一个较为精确的初始值，才能进一步由当前雷达扫描点阵匹配出机器人在map中的真实位置。



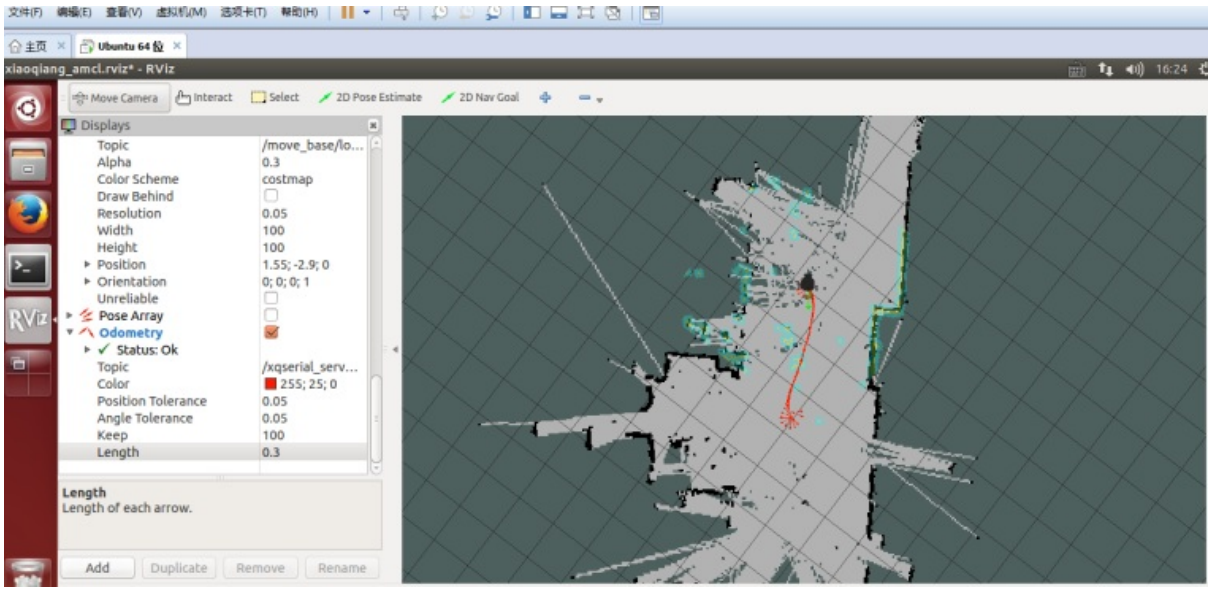
2.在rviz中使用2D Nav Goal 给小强发布目标点

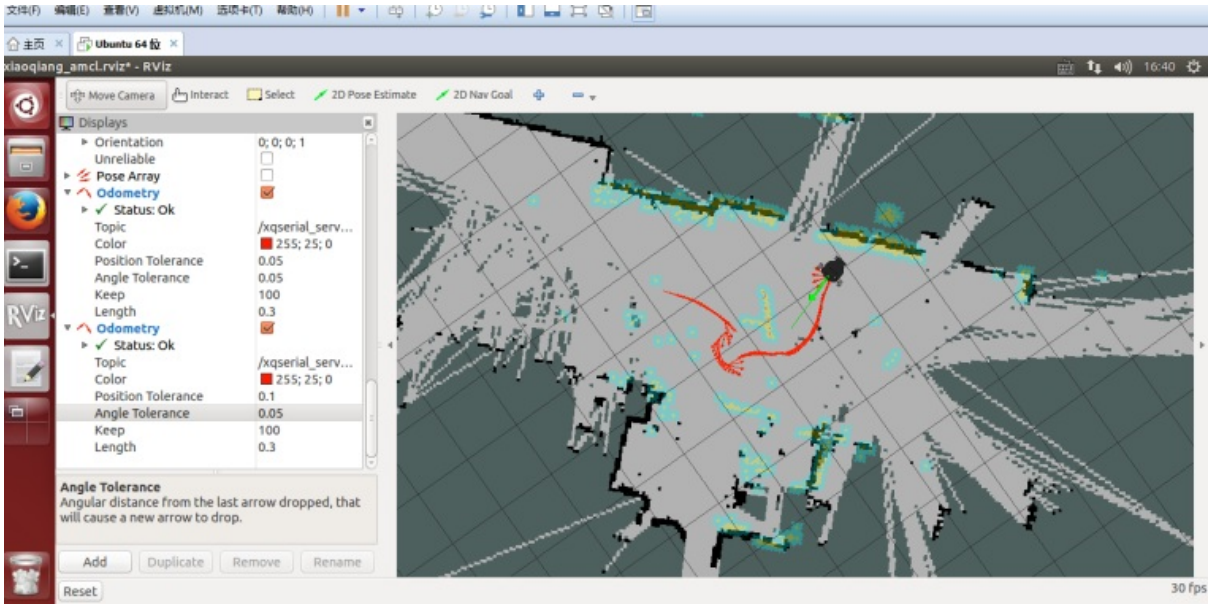


3.小强开始自主移动到指定位置



五、请自由设置小强的2D Nav Goal,观察小强的运动情况，本文结束





小强主页 返回目录

- 小强ROS机器人教程(16)_大范围激光雷达slam与实时回路闭合测试
  - 准备工作：
  - 操作步骤：

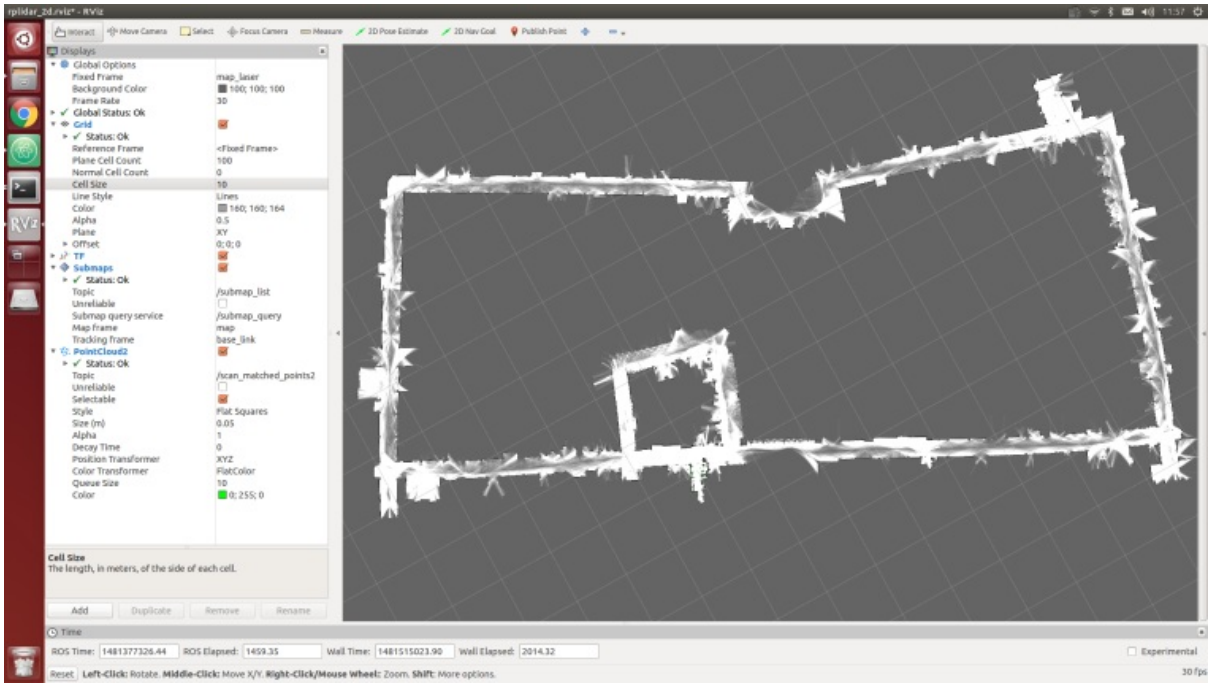
小强ROS机器人教程(16)_大范围激光雷达slam与实时回路闭合测试

小强主页

借助谷歌的Cartographer配合slamtec的激光雷达，我们可以尝试对大型建筑建立平面图。先看我们自己的demo演示效果。

00:00 / 00:00

在本demo中，小强实际运行在一个5000平米的写字楼走廊里，走廊两侧存在大量的玻璃幕墙，大楼中央存在一个大面积空旷地，加上rplidar的测距范围只有 6 米，因此下图的最终效果还算理想（只使用激光雷达，没有开启IMU和底盘odometer，大回路路径仍然成功闭合）



本文操作思路：因为是大范围建图，wifi网络覆盖是一个问题，所以我们借助蓝牙手柄来遥控小车运动。期间通过rosbag录制激光雷达数据，手柄遥控小车在建图范围内跑一圈，最后重放bag建图。

注：以下所有操作在小车主机ubuntu上完成

准备工作：

1．安装rplidar驱动

对于2016年11月15日之后购买小强开发平台的用户，rplidar驱动已经配置好。 rplidar的驱动安装请参考这篇教程

2．安装ps3手柄驱动

对于2016年11月15日之后购买小强开发平台的用户，请跳过本步骤。 请参考这篇安装教程

3．安装cartographer_ros

请参考这篇安装教程：http://community.bwbot.org/topic/136/google激光雷达slam算法cartographer的安装及bag包demo测试

操作步骤：

1．新开一个窗口启动rplidar

```
roslaunch rplidar_ros rplidar.launch
```

2．新开两个窗口启动ps3手柄遥控程序，按手柄连接键连上小车 第一个窗口

```
sudo bash
rosrun ps3joy ps3joyfake_node.py
```

第二个窗口

```
roslaunch turtlebot_teleop ps3fakexiaoqiang_teleop.launch
```

3．新开一个窗口启动rosbag录制进程,开始录制激光雷达数据/scan

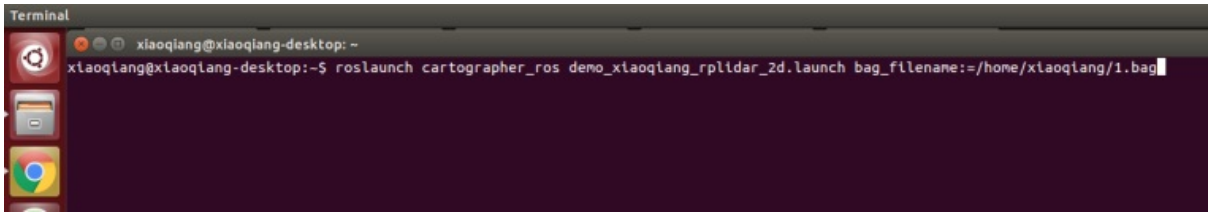
```
rosbag record /scan
```

4．用手柄遥控小车运动，绕建图区域一圈，也可以多圈

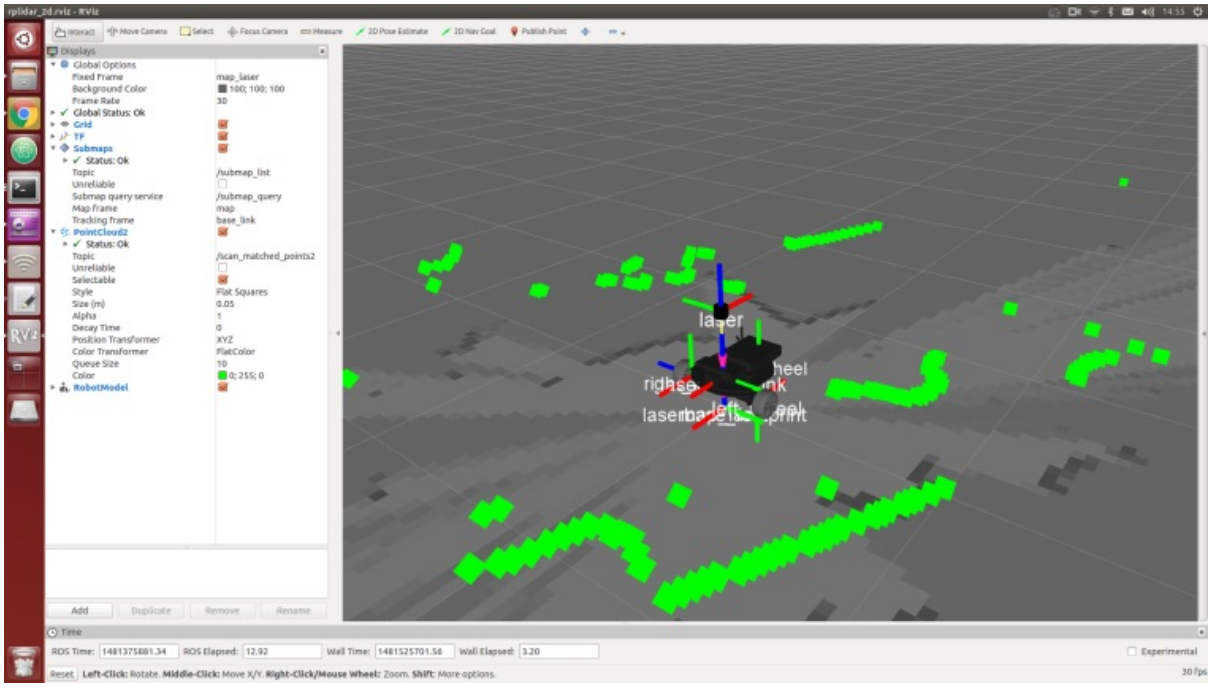
5．bag录制完成，关闭上文的１、２、３窗口 新录制的点bag文件在小强home目录下，将其重命名为１.bag

6．启动cartographer_ros开始bag回放建图

```
roslaunch cartographer_ros demo_xiaoqiang_rplidar_2d.launch bag_filename:=/home/xiaoqiang/
1.bag
```



7．一切正常的话，现在可以看到下图的类似效果，等待bag包play完



8.用map_server保存地图，本文结束

```
rosrun map_server map_saver --occ 51 --free 49 -f work0
```

小强主页 返回目录

- 小强ROS机器人教程(17)_利用ORB_SLAM2建立环境三维模型

小强ROS机器人教程(17)_利用ORB_SLAM2建立环境三维模型

小强主页

想要实现视觉导航，空间的三维模型是必须的。ORB_SLAM2就是一个非常有效的建立空间模型的算法。这个算法基于ORB特征点的识别，具有准确度高，运行效率高的特点。我们在原有算法的基础上进行了修改，增加了地图的保存和载入功能，使其更加适用于实际的应用场景。下面就介绍一下具体的使用方法。

`注意：以下方法目前程序版本已经不再支持，小强用户已经完全替换为Galileo导航系统。导航系统文档 https://doc.bwbot.org/en/books-online/galileo-servicebot-doc/`

`注意：由于小强的ORB_SLAM2版本已经升级到伽利略版本，所以运行时会检测是否有有效的证书。默认发货时小强是配置好证书的。如果没有证书也可以联系客服免费获取。如果你不是小强用户，那么下面的教程无法执行。`

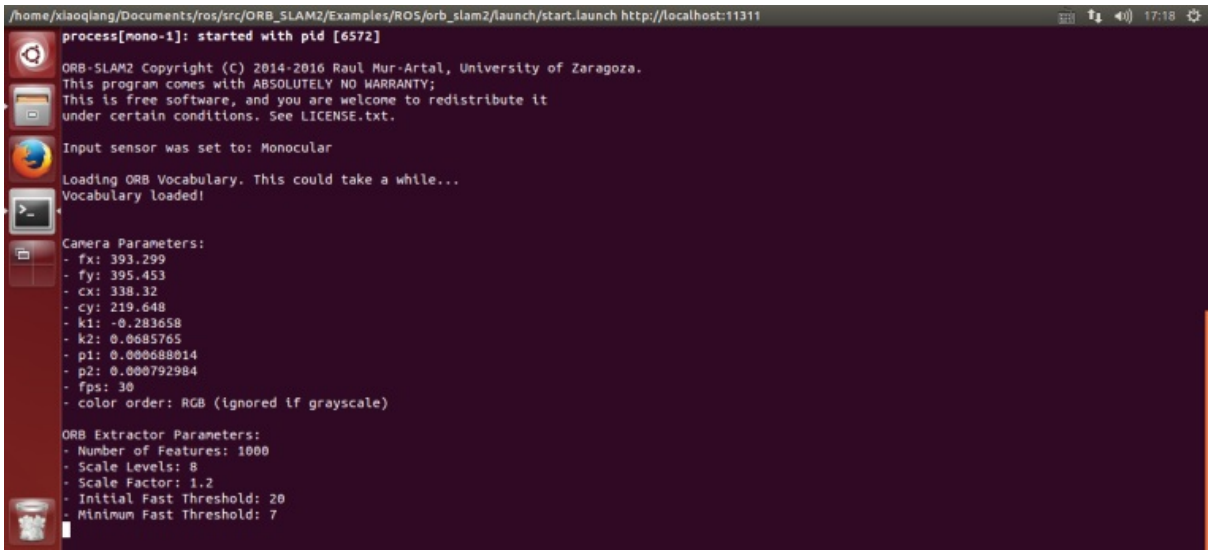
准备工作

在启动ORB_SLAM2之前，请先确认小强的摄像头工作正常。 ORB_SLAM2建图过程中需要移动小车，移动小车过程中ORB_SLAM2的运行状态不方便显示(ssh方式比较卡顿，也不可能拖着显示器)，因此请先安装好小强图传遥控windows客户端。

启动ORB_SLAM2

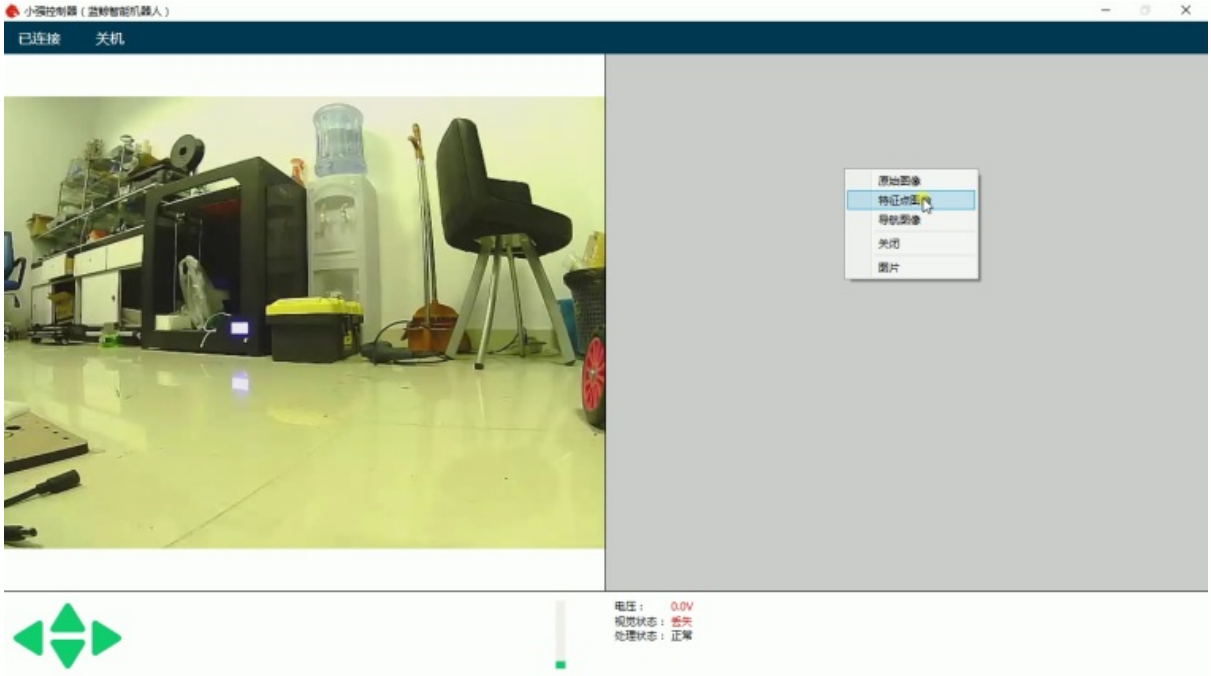
ssh方式进入小强，执行以下指令

```
roslaunch orb_slam2 map.launch
```



开始建立环境三维模型

打开小强图传遥控windows客户端，点击“未连接“按钮连接小强。在图传窗口右键打开”原始图像“和”ORB_SLAM2的特征点图像“



上图中，左侧图像是摄像头原始彩色图像，右侧是ORB_SLAM2处理后的黑白图像。当前ORB_SLAM2还没有初始化成功，所以黑白图像没有特征点。 按住”w”键开始遥控小强往前缓慢移动，使ORB_SLAM2初始化成功，即黑白图像开始出现红绿色特征点

现在就可以开始遥控小强对周围环境建图，遥控过程中需要保证黑白图像一直存在红绿色特征点，不存在则说明视觉lost了，需要遥控小强退回到上次没lost的地方找回视觉位置。

00:00 / 00:00

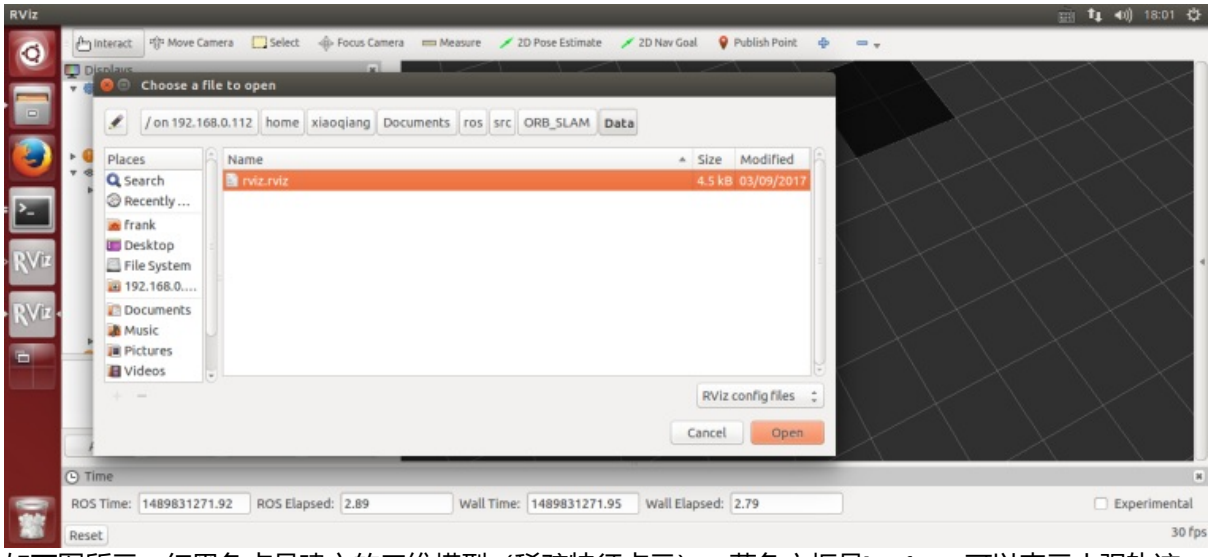
使用rviz查看建图效果

在机器人上打开rviz

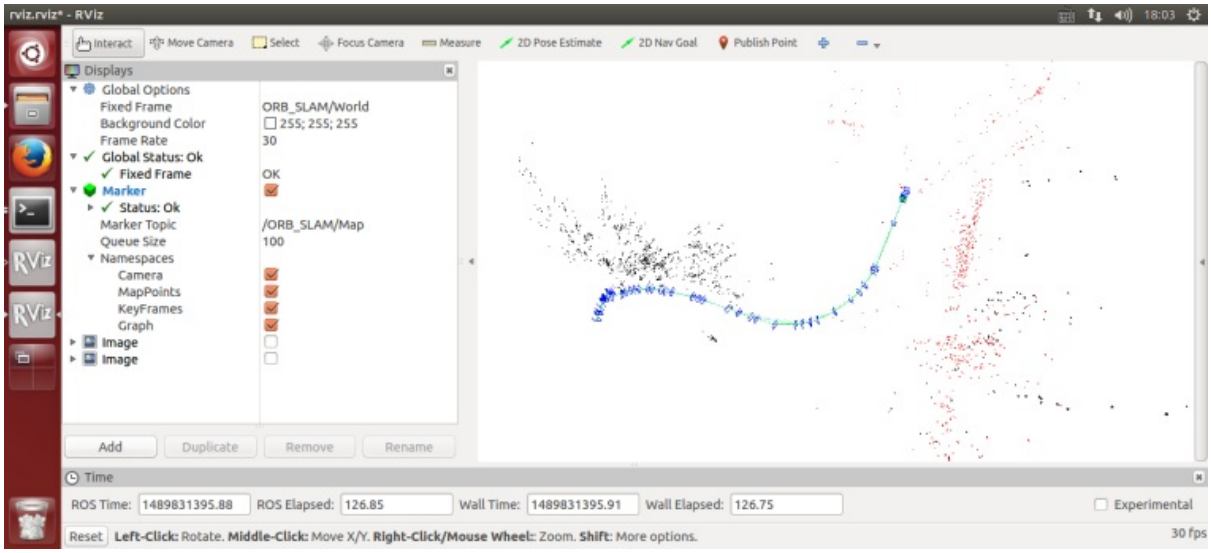
```
rviz
```

打开 `/home/xiaoqiang/Documents/ros/src/ORB_SLAM/Data/rivz.rviz` 配置文件

有些系统可能无法如此打开，可以先把文件复制到本地，然后再在本地打开



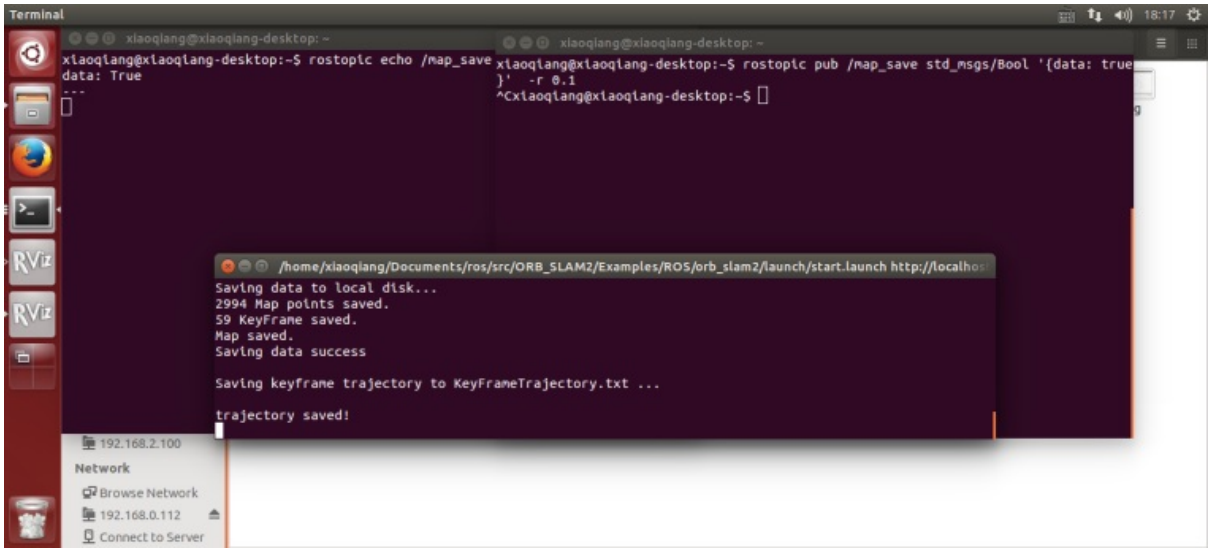
如下图所示，红黑色点是建立的三维模型（稀疏特征点云），蓝色方框是keyframe可以表示小强轨迹



保存地图

当地图建立的范围满足自己要求后，在虚拟机新开一个命令窗口，输入下列命令保存地图

```
rosrun orb_slam2 save_map.py map1
```



地图文件会被保存进用户主目录的 `slamdb` 文件夹内。

地图的载入

保存地图之后可以再次载入ORB_SLAM2程序中。地图载入后程序可以非常迅速的定位出摄像头的具体位置。这样就可以开发出基于视觉定位的导航算法了。 载入地图的方式也非常简单。

运行

```
rosparam set /galileo/galileo_map map1
roslaunch orb_slam2 run.launch
```

此处的run.launch实际上就是把LoadMap 设置成1.

地图的后期处理

在建立地图之后，想要使用这个地图进行导航，还需要对地图文件做进一步的操作。比如创建导航的线路，标记导航路径的行走方式等等。

对于小强用户可以直接使用导航客户端进行地图处理。详情可参见 导航系统文档

常见问题

Q: 可不可以不使用IMU
A: 小强的ORB_SLAM2是经过修改的版本，增加了IMU的融合。如果不想使用IMU可以在setting.yaml里面设置UseImu为0.

Q: 小强的标定文件在什么地方
A: 小强的标定文件，2018.6之后发货的，位于usb_cam包launch文件夹下ov2610.yaml。之前发货的位于ORB_SLAM2包内的setting.yaml，setting4.yaml 和setting5.yaml分别是map.launch和run.launch使用的。

Q: 小强的摄像头很不清晰
A: 可能是摄像头的镜头被碰到了导致没有聚焦。可以转动摄像头镜头，调整到图像清晰的位置。注意调整过之后摄像头参数需要重新标定。标定方法可以参见如何标定单目摄像头

小强主页 返回目录

- 小强ROS机器人教程(18)_使用DSO_SLAM进行三维建模

小强ROS机器人教程(18)_使用DSO_SLAM进行三维建模

小强主页

Direct Sparse Odometry(DSO)是业内很流行的lsd_slam系统作者的学生Jakob Engel开发的,实测性能和精度优于lsd_slam。DSO被作者开源到github，同时还一并开源了DSO在ros系统下的使用代码实例dso_ros. 本篇教程将演示如何在小强开发平台上安装DSO和dso_ros， 利用小强平台上的摄像头实时运行DSO进行三维建模。点击观看视频。

00:00 / 00:00

1．DSO的安装

注：因为小强开发平台已经提前安装好了不少DSO需要的依赖包，下文将跳过这些包的安装，请其它开发平台的读者参考github上的完整安装教程进行安装。

1.a安装依赖包

```
sudo apt-get install libsuitesparse-dev libeigen3-dev libboost-dev
sudo apt-get install libopencv-dev
```

1.b下载源代码

```
cd ~/Documents/
git clone https://git.bwbot.org/publish/dso
```

1.c继续配置依赖包

```
sudo apt-get install zlib1g-dev
cd ~/Documents/dso/thirdparty
tar -zxvf libzip-1.1.1.tar.gz
cd libzip-1.1.1/
./configure
make
sudo make install
sudo cp lib/zipconf.h /usr/local/include/zipconf.h
```

1.d编译安装

```
cd ~/Documents/dso/
mkdir build
cd build
cmake ..
make -j 2
```

2．dso_ros的安装

注：原作者提供的源代码有两个分支，master分支对应rosbuild版，catkin分支对应catkin版。对于现代ROS版本，推荐使用catkin版本，安装使用更方便。但是作者的catkin分支存在代码缺陷，实际无法安装使用，因此下文将安装我们蓝鲸智能修改之后的dso_ros版本。

```
cd ~/Documents/ros/src
git clone https://gitee.com/BluewhaleRobot/dso_ros.git
cd ..
export DSO_PATH=/home/xiaoqiang/Documents/dso
catkin_make #或者用这条命令 catkin_make -DCATKIN_WHITELIST_PACKAGES=""
```

3．开始使用

```
sudo service startup stop # 停止startup服务
roslaunch startup startup.launch # 启动摄像头等程序
roslaunch dso_ros dso_ros.launch # 启动dso程序
```

现在移动摄像头，就能开始对周围环境进行三维建模，移动过程避免急转弯和剧烈运动。

对于小强用户，可以先遥控小强运动同时用rosbag录制/camera_node/image_raw这个image topic数据，然后重放，这样可以实现大范围的建模。rosbag重放前，需要关闭usb摄像头节点（sudo service startup stop）,否则会有图像发布冲突。

小强主页 返回目录

- 小强ROS机器人教程(19)___NLlinepatrol_planner的简单使用

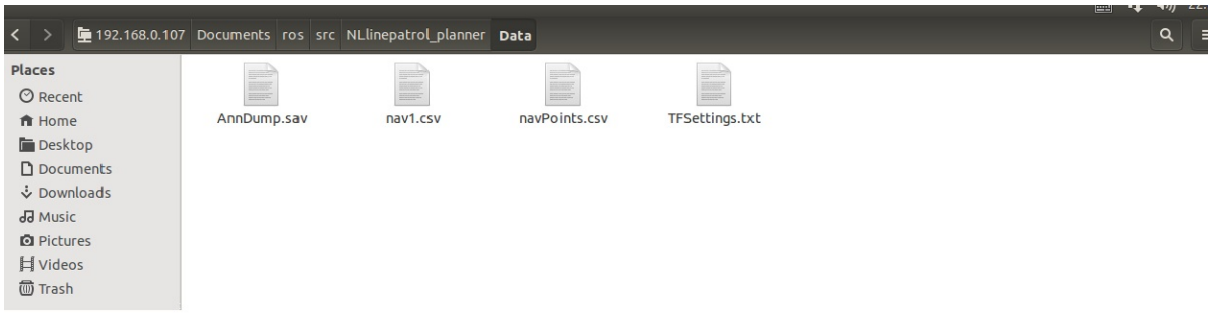
小强ROS机器人教程(19)___NLlinepatrol_planner的简单使用

小强主页

随小车主机附带的NLlinepatrol_planner是一个用于视觉导航的全局路径规划器，根据小车输出的视觉轨迹，能输出一条连接小车当前坐标和目的坐标的全局路径,下文将通过一个模拟实例来演示它的使用方式。主要思路是：一个python脚本发布虚拟的视觉里程计和相关tf树，一个python脚本给move_base节点发布目标点，最后move_base节点通过调用NLlinepatrol_planner获得一条全局路径并在rviz中显示。

1.配置NLlinepatrol_planner

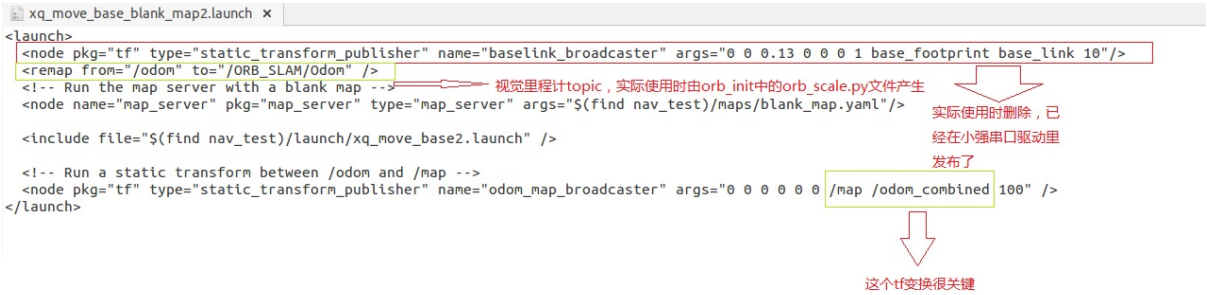
需要提供NLlinepatrol_planner待读取的视觉轨迹文件、轨迹坐标变换需要的变换参数文件，这两文件都应该放在NLlinepatrol_planner下面的data文件夹内，文件名任意，通过配置move_base中的相关参数可以指定NLlinepatrol_planner读取的文件，下文会说明。



在上图中，nav1.csv是视觉轨迹文件、TFSettings.txt为变换参数文件(第一行是旋转矩阵的9个元素、数组元素排列顺序为c语言的行排列，第二行为平移向量的xyz分量，第三行为scale因子)

2.制作move_base的launch文件

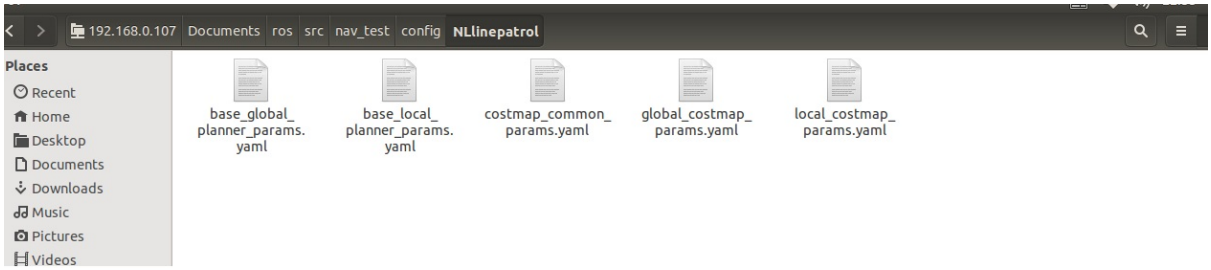
对于本教程，我们已经在nav_test软件包中的launch文件夹内提供了相关的launch文件，名字为xq_move_base_blank_map2.launch，这个launch文件在实际运用时可以作为模板，需要注意的地方请看下图



这个launch文件会调用xq_move_base2.launch文件，xq_move_base2.launch文件也在当前目录下，里面的内容如下

```
<launch>

  <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
    <param name="base_global_planner" value="NLlinepatrol_planner/NLlinepatrolPlanner"/>
    <rosparam file="$(find nav_test)/config/NLlinepatrol/costmap_common_params.yaml" comma
nd="load" ns="global_costmap" />
    <rosparam file="$(find nav_test)/config/NLlinepatrol/costmap_common_params.yaml" comma
nd="load" ns="local_costmap" />
    <rosparam file="$(find nav_test)/config/NLlinepatrol/local_costmap_params.yaml" command
="load" />
    <rosparam file="$(find nav_test)/config/NLlinepatrol/global_costmap_params.yaml" comma
nd="load" />
    <rosparam file="$(find nav_test)/config/NLlinepatrol/base_local_planner_params.yaml" c
ommand="load" />
    <rosparam file="$(find nav_test)/config/NLlinepatrol/base_global_planner_params.yaml"
command="load" />
  </node>

</launch>
```

通过上述内容，发现通过设置base_global_planner参数值来指定全局路径规划器为NLlinepatrol_planner，还可以看出move_base的其它参数配置文件放在nav_test软件包内的config/NLlinepatrol路径内



对于上图，我们需要更改的文件是base_global_planner_params.yaml,因为这个文件里的内容对应NLlinepatrol_planner运行时实际加载的参数，默认内容如下

```
NLlinepatrolPlanner:
    DumpFileName: AnnDump.sav
    strTFParsFile: TFSettings.txt
    TxtFileName: nav1.csv
    ANN_Dump_Bool: false
    connect_distance: 0.3
```

TxtFileName是加载的视觉轨迹文件名，strTFParsFile是加载的变换参数文件名，connect_distance设置视觉轨迹文件中连通点之间的最大距离（坐标变换后两个点之间的距离小于该值就认为这两点之间没有障碍物，可以直接连接），ANN_Dump_Bool值为false表示从txt文件中加载轨迹和变换参数、如果为true则从DumpFileName参数指定的dump文件加载（多次使用同一个视觉轨迹文件时，第二次以后从dump文件启动可以加速）

3.配置完成开始使用

A.因为我们这次是虚拟运行，发布的一些topic是没有实际意义的但是和小强默认ROS驱动冲突，所以现在需要停止所有ROS运行实例

```
sudo service startup stop
roscore
```

B.启动虚拟topic和小强模型文件

```
rosrun orb_init temp.py //发布odom
roslaunch xiaoqiang_udrf xiaoqiang_udrf.launch //启动模型
```

C.启动上文制作的xq_move_base_blank_map2.launch文件

```
roslaunch nav_test xq_move_base_blank_map2.launch
```

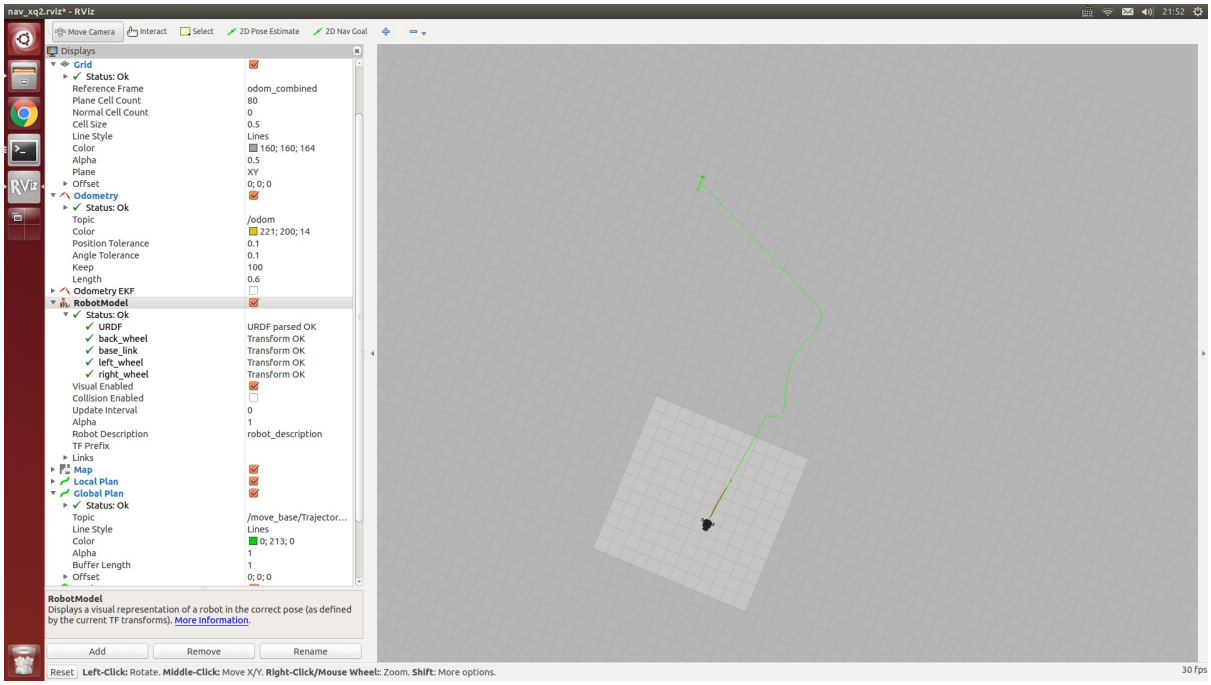
D.启动rviz，并打开ros/src/nav_test/config/nav_xq2.rviz配置文件

```
rviz
```

E.启动虚拟goal发布节点（基于惯性导航中的squre.py修改而来）

```
rosrun nav_test NLlinepatrol.py
```

4.现在rviz中就已经出现目标全局路径轨迹了(绿线)，想测试其它goal目标点，请自行修改NLlinepatrol.py中相关代码



小强主页 返回目录

- 小强ROS机器人教程(21)_获取usb摄像头30fps的1080p图像流及120fps的VGA分辨率图像流

小强ROS机器人教程(21)_获取usb摄像头30fps的1080p图像流及120fps的VGA分辨率图像流

小强主页

小车默认开机启动的usb摄像头节点，发布的是30fps的VGA（640*480）分辨率的图像流，可以满足大部分的视觉任务要求。usb摄像头硬件实际可以输出120fps的VGA图像流和高达30fps的1080p图像流，本文将介绍获取这两种规格图像的办法。

1.升级usb_cam节点源代码

由于usb2.0带宽问题，输出120fps的VGA视频和30fps的1080p视频，需要采用mjpeg格式。小车自带的usb摄像头ROS驱动包usb_cam，它不能以mjpeg方式读取小车usb摄像头，因此我们需要先升级usb_cam。升级办法请参考这篇帖子:http://community.bwbot.org/topic/144/解决ROS的usb_cam节点无法正常读取mjpeg格式摄像头的方法

2.测试120fps的VGA视频输出

首先关闭小车的开机启动任务

```
sudo service startup stop
```

修改usb_cam包中的ov2610mjpg.launch文件内容为下文所示

```
<launch>
  <node name="camera_node" pkg="usb_cam" type="usb_cam_node">
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="framerate" value="120" />
    <param name="pixel_format" value="mjpeg" />
    <param name="camera_frame_id" value="ov2610" />
    <param name="io_method" value="mmap"/>
  </node>
</launch>
```

启动上述launch文件

```
roslaunch usb_cam ov2610mjpg.launch
```

新开一个窗口打印发布的图像topic帧率

```
rostopic hz /camera_node/image_raw
```

正常会出现类似下图的输出

```
subscribed to [/camera_node/image_raw]
average rate: 99.497
    min: 0.004s max: 0.016s std dev: 0.00246s window: 98
average rate: 99.324
    min: 0.004s max: 0.016s std dev: 0.00240s window: 198
average rate: 99.385
    min: 0.004s max: 0.016s std dev: 0.00236s window: 297
average rate: 99.247
    min: 0.004s max: 0.016s std dev: 0.00236s window: 396
average rate: 99.302
    min: 0.004s max: 0.016s std dev: 0.00232s window: 495
average rate: 99.296
    min: 0.004s max: 0.016s std dev: 0.00231s window: 595
average rate: 99.249
    min: 0.004s max: 0.016s std dev: 0.00230s window: 694
```

上文输出显示摄像头帧率大概是99fps，并没有达到120fps，这是因为环境光照的影响，如果环境光照亮度充足时是可以达到120帧的。

2.测试30fps的1080p视频输出

首先关闭小车的开机启动任务

sudo service startup stop 修改usb_cam包中的ov2610mjpg.launch文件内容为下文所示

```
<launch>
  <node name="camera_node" pkg="usb_cam" type="usb_cam_node">
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="1920" />
    <param name="image_height" value="1080" />
    <param name="framerate" value="30" />
    <param name="pixel_format" value="mjpeg" />
    <param name="camera_frame_id" value="ov2610" />
    <param name="io_method" value="mmap"/>
  </node>
</launch>
```

启动上述launch文件

```
roslaunch usb_cam ov2610mjpg.launch
```

新开一个窗口打印发布的图像topic帧率

```
rostopic hz /camera_node/image_raw
```

正常会出现类似下图的输出

```
xiaoqiang@xiaoqiang-desktop:~$ rostopic hz /camera_node/image_raw
subscribed to [/camera_node/image_raw]
```

```
average rate: 29.986
    min: 0.028s max: 0.039s std dev: 0.00318s window: 30
average rate: 29.917
    min: 0.028s max: 0.039s std dev: 0.00296s window: 59
average rate: 29.912
    min: 0.028s max: 0.039s std dev: 0.00292s window: 89
average rate: 29.853
    min: 0.027s max: 0.042s std dev: 0.00308s window: 119
average rate: 29.874
    min: 0.027s max: 0.042s std dev: 0.00283s window: 149
average rate: 29.839
    min: 0.027s max: 0.042s std dev: 0.00282s window: 179
^Caverage rate: 29.849
    min: 0.027s max: 0.042s std dev: 0.00282s window: 202
```

小强主页 返回目录

- 小强ROS机器人教程(22)_操作6自由度机械臂

小强ROS机器人教程(22)_操作6自由度机械臂

小强主页

6自由度机械臂的资料在这个百度云盘链接里：http://pan.baidu.com/s/1nuBpDaD 请仔细阅读二次开发部分的串口通信协议、虽然下文使用的是usb hid通信方式，但命令格式是一样

下文将操作6自由度机械臂完成主控板上存储的3组动作。

控制原理： 小车主机使用usb连接机械臂主控板，用户发送名为robot_arm/cmdstring的topic，这个topic内容为控制命令，最后由robot_arm节点负责将这个topic内容经由usb hid协议发送给机械臂主控板。

1.运行robot_arm节点

```
rosrun robot_arm move.py
#robot_arm包代码在这里 http://git.bwbot.org/publish/robot_arm
```

启动正常，机械臂会执行动作0（默认状态），同时命令窗口会显示如下内容

```
xiaoqiang@xiaoqiang-desktop:~/Documents/ros$ rosrun robot_arm move.py
Opening robot arm device
Manufacturer: MyUSB_HID
Product: LOBOT
Serial No: 8D9823654852
Run the zero group action
```

2.构造robot_arm/cmdstring topic

robot_arm/cmdstring 这个topic 类型是std_msgs.msg中的String 即字符串。根据上文提供的机械臂二次开发资料，我们可以知道6自由度的控制协议是由一个无符号byte数组表示，因此我们在这里稍加改造，直接将这个数组的hex值变成字符串打包成topic命令。变换方式是采用python的byte array表示方式，将数组中每个元素的hex编码串在一起，然后将其中的0x换成\x。

```
例如： [0x55,0x55,0x05,0x06,0x00,0x01,0x00] 这个控制命令数组，转换成robot_arm/cmdstring 内容是，'
\x55\x55\x05\x06\x00\x01\x00'
小技巧：看不懂的话，还可以用python的map函数帮忙转换，找规律。map(ord,'\x55\x55\x05\x06\x00\x01\x00'
)
```

我们要控制机械臂完成3个动作，这三个字符串内容如下

```
'\x55\x55\x05\x06\x00\x01\x00'

'\x55\x55\x05\x06\x01\x01\x00'

'\x55\x55\x05\x06\x02\x01\x00'
```

```
找到规律没...
```

3.发布运动命令

```
新开一个命令行终端，因为是演示，所以直接使用rostopic 的pub功能，将上面的字符串命令打包成topic发给robot_arm节点
动作1:  rostopic pub robot_arm/cmdstring std_msgs/String '\x55\x55\x05\x06\x00\x01\x00'
动作2:  rostopic pub robot_arm/cmdstring std_msgs/String '\x55\x55\x05\x06\x01\x01\x00'
动作3:  rostopic pub robot_arm/cmdstring std_msgs/String '\x55\x55\x05\x06\x02\x01\x00'
```

4.运动结果暂时没有视频，需要自己实际操作，觉得运动不满意的话请参考我们这个舵机运动控制帖子：电机控制与缓动函数

小强主页 返回目录

- 小强ROS机器人教程(23)___ROS入门手册

小强ROS机器人教程(23)___ROS入门手册

小强主页

Learning ROS for Robotics Programming - Second Edition.pdf这本教程很基础，虽然以Hydro版本为例，但是也完全兼容kinetic版本，代码实例中只需将书中的Hydro字符串替换成kinetic即可。 在完成小强ROS机器人教程(1)后，如果您不熟悉ROS，请重点阅读本书的第二章和第三章。

小强主页 返回目录

- 在小强中使用语音识别和语音合成
  - 安装，小强已经默认安装好了
  - 使用
    - 使用语音合成功能
    - 使用语音识别功能
    - 同时启动语音识别和语音合成
    - 和机器人对话
    - 参数说明

在小强中使用语音识别和语音合成

小强主页

为小强添加语音识别和语音合成功能，安装之后小强就能说会道了。当然首先要给小强添加话筒和喇叭。

处理后端可以设置为科大迅飞语音或百度语音。也可以利用audio_capture实时的处理从话筒中收到的数据。

安装，小强已经默认安装好了

```
cd [到你的工作空间的src文件夹中]
git clone https://gitee.com/BluewhaleRobot/xiaoqiang_tts
# 对于python3环境需要切换至python3分支
cd xiaoqiang_tts
git checkout python3
git clone https://gitee.com/BluewhaleRobot/xiaoqiang_audio
cd xiaoqiang_audio
git checkout noetic
cd ../..
catkin_make -DCATKIN_WHITELIST_PACKAGES="xiaoqiang_tts"
```

程序使用的是我自己申请的百度和迅飞参数，最好自己再去百度和迅飞的官网申请一个。然后在launch文件中设置自己的appid，再catkin_make一下就可以了。

使用

使用语音合成功能

启动百度语音合成

```
roslaunch xiaoqiang_tts tts_baidu.launch
```

输入Topic	消息类型

/xiaoqiang_tts/text	std_msgs/String

输出Topic	消息类型
/xiaoqiang_audio/audio	audio_common_msgs/AudioData

发布消息，测试语音

```
rostopic pub /xiaoqiang_tts/text std_msgs/String 测试一下语音合成 -1
```

此时如果正常应该能够听到"测试一下语音合成"的声音

启动科大迅飞语音合成

`注意要先关闭百度tts节点`

```
roslaunch xiaoqiang_tts tts_xunfei.launch
```

测试方法和上面一样。正常应该会听到合成的声音。

使用语音识别功能

启动百度语音识别

```
roslaunch xiaoqiang_tts asr_baidu.launch
```

新开一个终端接收语音识别结果

```
rostopic echo /xiaoqiang_tts/text
```

现在可以开始说话了，程序会自动监听环境声音并进行分句。当你停止说话时会开始处理你的这一句的结果。

输入Topic	消息类型
/xiaoqiang_audio/audio	audio_common_msgs/AudioData

输出Topic	消息类型
/xiaoqiang_tts/text	std_msgs/String

启动迅飞语音识别

```
roslaunch xiaoqiang_tts asr_xunfei.launch
```

使用方法和上面的一样

同时启动语音识别和语音合成

使用科大迅飞语音合成和语音识别

```
roslaunch xiaoqiang_tts tts_xunfei.launch
roslaunch xiaoqiang_tts asr_xunfei.launch
```

现在你说一句话机器人就会跟着你说一句话。

和机器人对话

```
roslaunch xiaoqiang_nlp talk_bot.launch
```

执行这个指令后机器人会和你对话交流，快和它说说话吧。

参数说明

详细的参数说明请参照launch文件内的注释

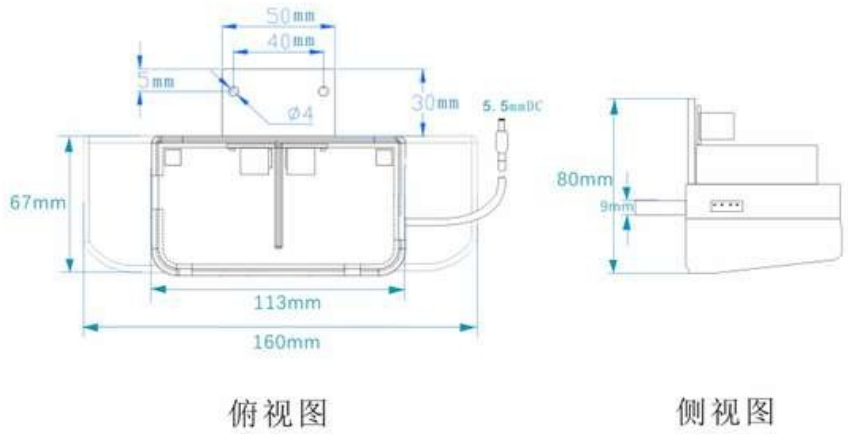
小强主页 返回目录

- 小强ROS机器人教程(27)___bw_auto_dock自动充电功能包的使用和实现原理
    - 点击此处购买自动充电模块套件
    - 自动充电模块串口通信协议v1.2.pdf，2020年新版
    - 自动充电模块串口通信协议V1.0.pdf，2020年之前老版本
    - ROS驱动包
    - 【使用与实现原理】
    - 1. 将自动充电模块安装在小车尾部，把充电桩固定好，接好电源。小车充电过程会接触充电桩，因此充电桩需要固定不能左右、前后移动，最好是靠墙放置。充电桩前面需要至少预留1米乘以1.5米的自由活动空间。
    - 2. 在主机ros catkin工作空间中下载安装自动充电模块的ros驱动包
    - 3.将模块通过usb转串口模块接入主机usb，添加udev规制，将串口号映射为/dev/ttyUSB004，同时根据插入的usb端口号进行端口绑定。参考：这里
    - 4.启动ros驱动节点，开始测试自动充电功能
    - 5.自动充电原理介绍
    - 6.常见问题
    - 7.升级控制算法

小强ROS机器人教程(27)___bw_auto_dock自动充电功能包的使用和实现原理

点击此处购买自动充电模块套件

尺寸图： （尾部两个4mm通孔用于固定小车底板尾部，5.5mmDC头接车载电池）



自动充电模块串口通信协议v1.2.pdf，2020年新版

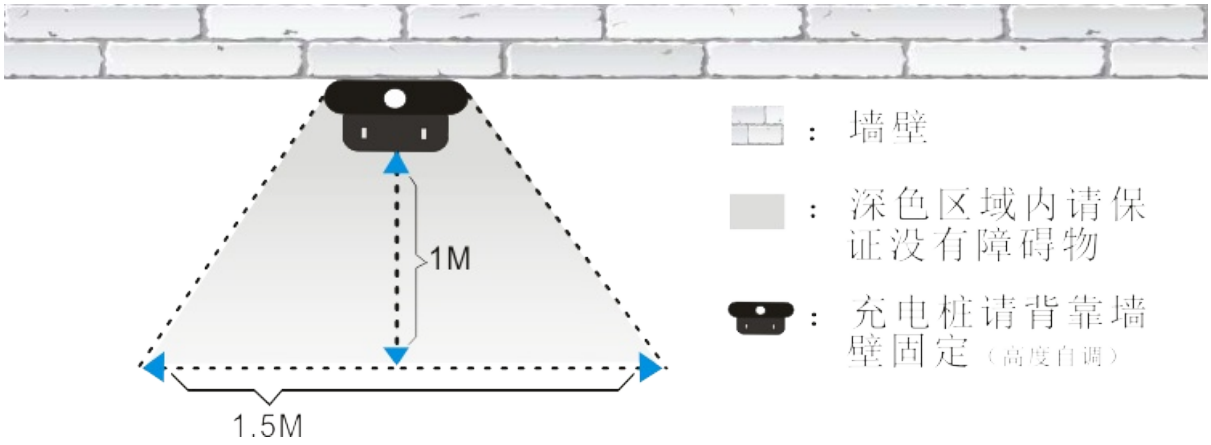
自动充电模块串口通信协议V1.0.pdf，2020年之前老版本

ROS驱动包

【使用与实现原理】

2018年12月之后购买的用户，充电模块全部升级到了2代，2代相比一代使用超声波替换了碰撞开关，减少了模块机械损坏机率。2代支持同时测量电池电压和充电输入电压，1代只支持测量充电电压。电压测量值以ros话题形式发布，具体请看驱动包readme文件。

1. 将自动充电模块安装在小车尾部，把充电桩固定好，接好电源。小车充电过程会接触充电桩，因此充电桩需要固定不能左右、前后移动，最好是靠墙放置。充电桩前面需要至少预留1米乘以1.5米的自由活动空间。



2. 在主机ros catkin工作空间中下载安装自动充电模块的ros驱动包

如果是已经配置好的小强用户，可以直接跳过本步骤。

```
#假设catkin工作空间在~/Documents/ros
cd  ~/Documents/ros
cd src/
git clone https://gitee.com/BlueWhaleRobot/bw_auto_dock.git
#切换到对应分支，1代切换到master，2代切换到version2分支，如果ros版本是melodic以上版本用noetic-version2分支
git checkout version2 #语法“git checkout 分支名字”
cd ..
catkin_make
```

3.将模块通过usb转串口模块接入主机usb，添加udev规制，将串口号映射为/dev/ttyUSB004，同时根据插入的usb端口号进行端口绑定。参考：这里

如果是已经配置好的小强用户，可以直接跳过本步骤。

```
#先安装串口设置包
sudo apt-get install setserial
```

```
#将当前计算机用户名添加进dailout用户组，以xiaoqiang为例
sudo adduser xiaoqiang dialout
```

```
#查看串口所在usb端口，即下条命令输出结果中/ttyUSBXXX字符前面的数字例如2-2:1.0，/ttyUSBXXX需要替换成实际端口号
udevadm info --name=/dev/ttyUSBXXX --attribute-walk
```

```
#根据这个端口号和映射信息/dev/ttyUSB004建立udev规则文件/etc/udev/rules.d/95-persistent-serial.rules，下面内容可供参考。对于非小强用户，xiaoqiang需要更换成自己的计算机用户名
ACTION!="add", GOTO="persistent_serial_end"
SUBSYSTEM!="tty", GOTO="persistent_serial_end"
KERNEL!="ttyUSB[0-9]*", GOTO="persistent_serial_end"

# This is old 11.10 style: IMPORT="usb_id --export %p"
IMPORT{builtin}="path_id"
KERNELS=="2-2:1.0"         ,SYMLINK+="ttyUSB004" , OWNER="xiaoqiang" ,RUN+="/bin/setserial /dev/ttyUSB004 low_latency"
LABEL="persistent_serial_end"
```

```
#最后重新加载udev规则
udevadm control --reload-rules
#重新插拔u转串模块 如果 ls /dev可以看到ttyUSB004设备，说明操作成功。
```

4.启动ros驱动节点，开始测试自动充电功能

对于小强用户，可以直接启动xiaoqiang_local.launch进行测试。 对于其它平台的用户，请根据xiaoqiang_local.launch里面的参数注释，把里程计和安装参数修改一下。

```
roslaunch bw_auto_dock xiaoqiang_local.launch
```

手动设置充电桩位置：节点启动成功后，先遥控小车到充电桩附近，小车可以不用接触上充电桩，保证充电桩已经很接近小车充电部位（相差10cm左右），现在可以通过运行下列命令手动发布充电桩位置保存指令，这样小车会把当前位置记录为充电桩位置。

```
rostopic pub  /bw_auto_dock/dockposition_save   std_msgs/Bool   '{data: true}'  -1
#保存成功后会在launch文件设置的目录里面生成一个文件，里面有位置数据。
#只有在红外传感器收到数据而且里程计消息有效时才会保存数据
```

遥控小车到一个随机位置，运行下列命令后，小车会自动进入自动充电模式，然后尝试自动对准充电桩。

```
rostopic pub  /bw_auto_dock/EnableCharge   std_msgs/Bool   '{data: true}'  -1
```

运行下列命令可以使小车退出自动充电模式，现在又可以返回到上一步，继续发布自动充电命令。

```
rostopic pub  /bw_auto_dock/EnableCharge   std_msgs/Bool   '{data: false}'  -1
```

00:00 / 00:00

5.自动充电原理介绍

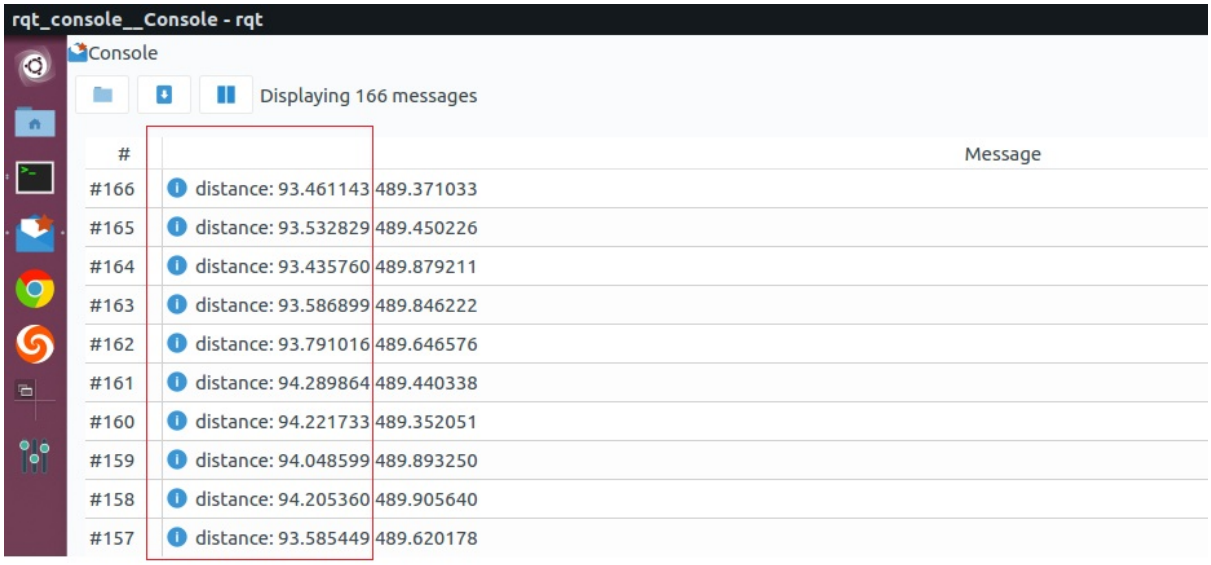
充电桩会广播一组红外信号，小车上的自动充电模块有4个红外接收器。通过分析这四个红外接收器接收的红外信号，小车可以确定自己相对充电桩的位置。

安装在小车上的自动充电模块有充电电压、充电电流检测功能，还有两个碰撞检测开关。 整个自动充电控制逻辑是这样的：

a. 设置充电桩位置后，bw_auto_dock会自动设定两个参考点，充电桩位于这两个参考点连线一侧的中间位置。 b. 每次接收到自动充电指令后，小车会自动在这两个参考点之间循环运动。 c. 当L1或者R1接收器接收到充电桩的中间位置信号后，小车会判断已经到达充电桩正前方，会先后退一段距离（back_distance长度），然后原地旋转使R2、L2接收器正对充电桩。 d. 小车开始直线后退对准充电桩，期间根据R2、L2接收的信号进行pid对准控制。 e. 如果侦测到充电电压、电流，就停止移动，开始充电。如果触发了碰撞开关，同时还没有侦测到电压和电流就回到步骤b。

6.常见问题

```
a.back_distance该如何设置?
    back_distance对应的是超声波模块到base_link的距离，这个距离决定了上文5.c步骤小车后退距离。如果发现小车每次5.c旋转后都没有正对充电桩，说明这个参数设置的过大或过小，手动调一下back_distance参数（一般是减去过调的距离）。

b.在步骤5.e中机器人经常没有碰到充电桩就不后退了反而立即离开，即crash_distance该如何设置?
    crash_distance是一个碰撞检测阈值，在5.e步骤中，如果超声波探测的值小于crash_distance就认为是触发了碰撞开关。
    如果5.e中机器人经常没有碰到充电桩就不后退了反而立即离开，说明crash_distance值设置的太大了，尝试把它改小一些（每次减少10）。
    如果5.e中机器人已经碰到墙了，还在不断后退，说明crash_distance值设置的太小了，尝试把它改大一些（每次增加10）。
    在rqt_console里面打开dock_driver节点的debug输出，可以查看超声波模块的当前测量值。
```



7.升级控制算法

xq5分支代码是控制算法升级后的版本(ros melodic以上版本需要用noetic分支代码)，同时整合提供了service和action接口，还支持手动对接充电，具体请阅读代码了解。

```
#编译xq5分支代码(或者noetic分支)，所依赖的ros包从这里git clone下载后catkin_make编译
https://gitee.com/BluewhaleRobot/galileo_serial_server.git
https://gitee.com/BluewhaleRobot/galileo_msg.git
```

- 小强ROS机器人教程(28)_使用Intel RealSense D400系列深度摄像头进行自主移动避障
    - 1.驱动测试
    - 2.关闭步骤1中命令，开始自主移动避障测试

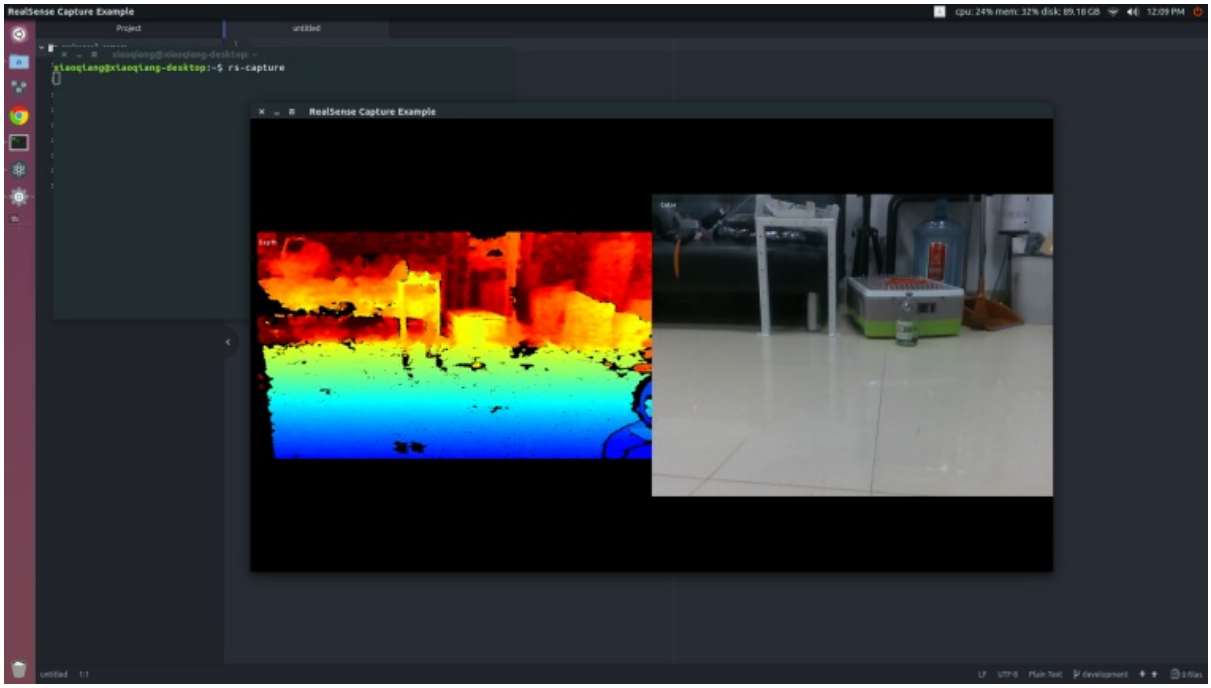
小强ROS机器人教程(28)_使用Intel RealSense D400系列深度摄像头进行自主移动避障

1.驱动测试

连接好RealSense摄像头后，运行下述命令可以测试sdk有没有安装成功

```
#sdk的安装教程 https://github.com/IntelRealSense/librealsense/blob/master/doc/distribution_linux.md#installing-the-packages
rs-capture
```

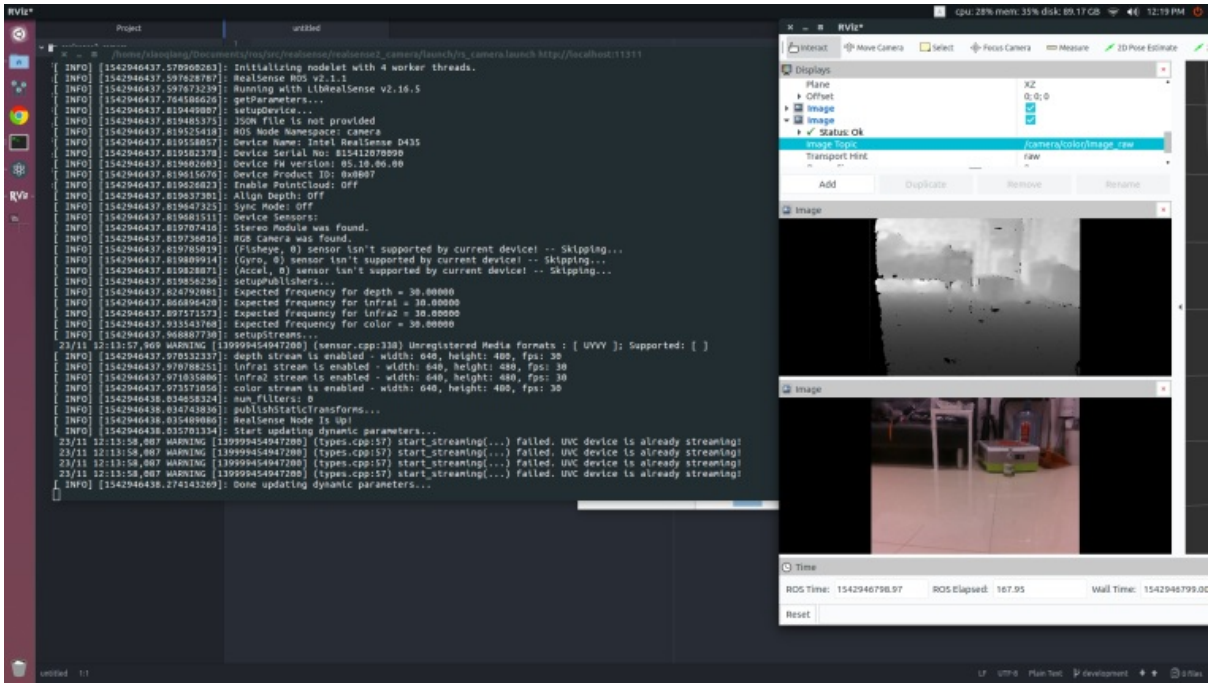
正常的话可以看到类似下图的图像



关闭上述命令，继续测试ros驱动

```
#ros驱动的安装教程 https://gitee.com/bluewhalerobot/realsense
roslaunch realsense2_camera rs_camera.launch
```

如果正常，在rviz里面可以订阅显示深度和color图像话题。



2.关闭步骤1中命令，开始自主移动避障测试

intel RealsSense和kinect是兼容的，使用它来自主移动避障的教程也和kinect一样，区别是教程2.b、2.c步骤需要选择realsense。

```
roslaunch realsense2_camera rs_camera_xiaoqiang.launch
```

kinect避障教程： https://community.bwbot.org/topic/115/小强ros机器人教程-10-___使用kinect进行自主移动避障

- 小强ROS机器人教程(29)_安装使用zed相机cpu版本ros驱动
  - 1. 在主机ros catkin工作空间中下载安装zed的ros驱动包
  - 2.将zed接入主机usb，添加udev规制，将设备号映射为/dev/video003
  - 3.启动ros驱动节点，正常的话可以在rviz中订阅显示zed的图像数据。

小强ROS机器人教程(29)_安装使用zed相机cpu版本ros驱动

官方sdk和ros驱动需要使用cuda，如果主机不支持cuda就无法使用这些软件包，也就无法使用zed相机。很幸运，zed相机是一个uvc标准的摄像头，有第三方的ros驱动可以支持无nvidia显卡的主机使用。因为只使用cpu，因此只能获取摄像头的双目图像，官方的slam和深度信息处理功能是无法使用的。 下文将详细介绍如何在小强主机上安装使用zed相机cpu版本ros驱动。对于一些出厂已经配置过zed的小强用户，请跳过驱动安装步骤，可以直接启动相关launch文件进行测试。

1. 在主机ros catkin工作空间中下载安装zed的ros驱动包

如果是已经配置好的小强用户，可以直接跳过本步骤。

```
#假设catkin工作空间在~/Documents/ros
cd  ~/Documents/ros
cd src/
git clone https://github.com/BluewhaleRobot/zed_cpu_ros.git
cd ..
catkin_make
```

2.将zed接入主机usb，添加udev规制，将设备号映射为/dev/video003

如果是已经配置好的小强用户，可以直接跳过本步骤。

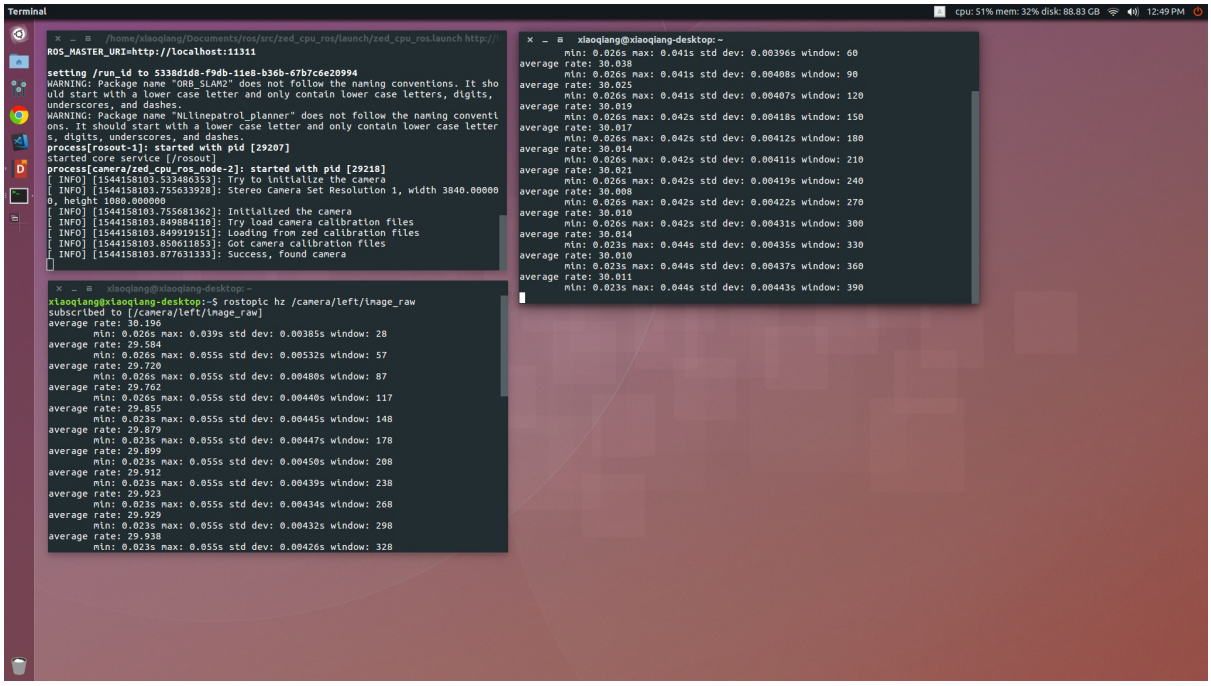
```
#查看zed相机的id信息，即下条命令输出结果中以{SUBSYSTEM=="video4linux"}开头的部分内容，/videoXXX需要替换成实际设备号
udevadm info --name=/dev/videoXXX --attribute-walk
```

```
#根据id信息，建立udev规则文件/etc/udev/rules.d/56-zed.rules，下面内容可供参考。对于非小强用户，xiaoqiang需要更换成自己的计算机用户名
SUBSYSTEM=="video4linux", ATTR{name}=="ZED: ZED", MODE:="0666", OWNER:="xiaoqiang", SYMLINK+="video003"
```

```
#最后重新加载udev规则
udevadm control --reload-rules
#重新插拔zed相机，如果 ls /dev可以看到video003设备，说明操作成功。
```

3.启动ros驱动节点，正常的话可以在rviz中订阅显示zed的图像数据。

```
roslaunch zed_cpu_ros zed_cpu_ros.launch
```



- 使用depthimage_to_laserscan包将kinect深度图像转换成2d激光雷达话题
  - 1.安装depthimage_to_laserscan包
  - 2.启动转换节点开始测试
  - 4.在rviz中查看2d雷达数据

使用depthimage_to_laserscan包将kinect深度图像转换成2d激光雷达话题

kinect 1代的驱动包freenct_stack可以输出深度图像和点云，理论上可以使用pointcloud_to_laserscan包将点云转换成scan数据，但是freenct_stack包输出的点云数据所在的坐标系不是常用的base_link坐标系，要先进行坐标变换才能转换，这导致pointcloud_to_laserscan包效率很低。

```
#运行下列命令可以测试pointcloud_to_laserscan包的转换功能，用“rostopic hz /kinect/scan”查看输出的/k
inect/scan话题发布频率，可以发现值很低，即转换效率太低不实用
roslaunch pointclound_to_laserscan xiaoqiang_lungu_kinect.launch
```

下文将介绍使用depthimage_to_laserscan包将kinect深度图像转换成2d激光雷达话题的详细步骤，这种方法输出的雷达话题/kinect/scan，频率高达20HZ，满足实时性要求。

1.安装depthimage_to_laserscan包

```
#需要安装到ros工作空间，以小强主机为例
cd ~/Documents/ros/src
git clone https://github.com/BluewhaleRobot/depthimage_to_laserscan.git
cd  ..
catkin_make
```

2.启动转换节点开始测试

freenct_stack驱动发布的深度图像/kinect/depth/image被 depthimage_to_laserscan节点订阅后将其转换成laserscan话题/kinect/scan，所在的frame_id则是“kinect_link”

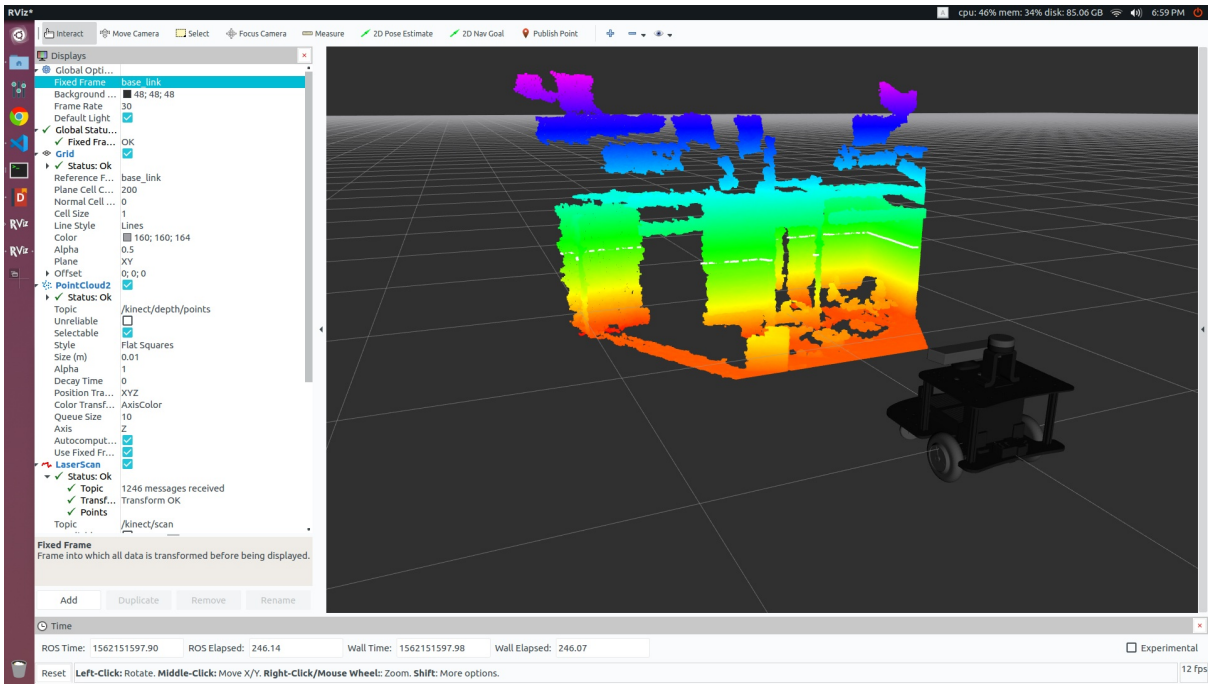
```
#新开一个终端
#如果是小强xq5轮毂电机版本，请启动这个launch文件
roslaunch depthimage_to_laserscan xiaoqiang_lungu_kinect.launch
#如果是小强xq4-pro普通电机版本，请启动这个launch文件
roslaunch depthimage_to_laserscan xiaoqiang_pro_kinect.launch

#新开一个终端将kinect角度设为水平方向
rostopic pub /set_tilt_degree std_msgs/Int16 '{data: 0}' -1
```

正常的话，现在就可以获取10hz的laserscan类型话题数据了

```
#新开一个终端,查看转换后的/kinect/scan话题发布频率
rostopic hz /kinect/scan
```

4.在rviz中查看2d雷达数据

```
roscd depthimage_to_laserscan/launch/
rviz -d  kinect.rviz
#图中白线就是转换输出的2d雷达数据
```



- 小强ROS机器人教程(33)_使用ps4手柄控制小车移动
  - 操作步骤
    - 0.先要安装对应的ros驱动同时将手柄与ubuntu进行蓝牙配对
    - 1.将ps4手柄与蓝牙接受器进行连接

小强ROS机器人教程(33)_使用ps4手柄控制小车移动

小强主页

原理：本教程涉及2个包 ds4_driver节点负责将蓝牙手柄设备数据转换成ros中的joy数据类型 turtlebot_teleop_joy负责将上述joy数据topic 转换成小车运动指令 `/cmd_vel`

操作步骤

0.先要安装对应的ros驱动同时将手柄与ubuntu进行蓝牙配对

参考ps4手柄ros驱动安装和蓝牙配对教程

如果手柄和小强同时购买，那么我们已经提前为您将手柄适配好了，可以从步骤1开始使用。

1.将ps4手柄与蓝牙接受器进行连接

只需要按一下ps键，让手柄进入连接模式，此时手柄会闪白光。

ubuntu不用进行任何操作，手柄就会连上ubuntu，连上后手柄led会停止闪烁，同时颜色变成蓝色。

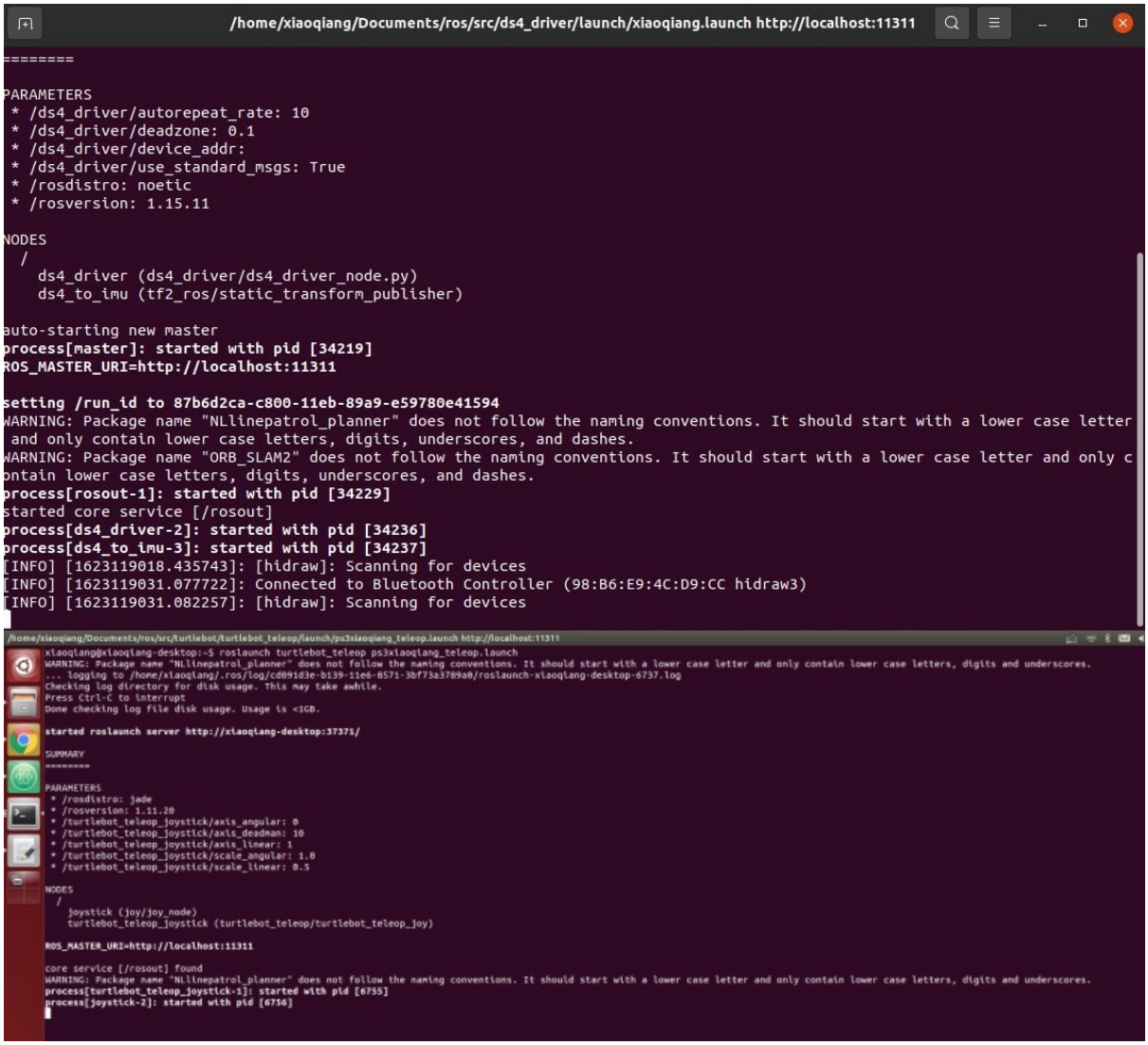
如果白灯熄灭前，也没连上ubuntu，可以等白灯熄灭后再次按ps键，再次尝试连接。蓝牙适配器性能不同，有时需要按两三次才能连上。

2.启动ds4_driver和turtlebot_teleop_joy

已经适配好小强的用户，这些节点默认开机自动启动，不用手动执行

```
#启动ds4_driver
roslaunch ds4_driver xiaoqiang.launch
#启动 turtlebot_teleop
roslaunch turtlebot_teleop ps3fakexiaoqiang_teleop.launch
```

正常启动后如下图所示



ps3fakexiaoqiang_teleop.launch文件内容如下

```
<launch>

  <node pkg="turtlebot_teleop" type="turtlebot_teleop_joy" name="turtlebot_teleop_joystick"
>
    <param name="scale_angular" value="0.4"/>
    <param name="scale_linear" value="0.4"/>
    <param name="axis_deadman" value="10"/>
    <param name="axis_linear" value="1"/>
    <param name="axis_angular" value="0"/>
    <remap from="turtlebot_teleop_joystick/cmd_vel" to="/cmd_vel"/>
    <remap from="turtlebot_teleop_joystick/joy" to="/joy"/>
  </node>
</launch>
```

上述launch文件中的参数分别对应直线速度最大值(scale_linear)和角速度最大值(scale_angular)，油门离合键(axis_deadman)、前进后退轴(axis_linear)、左右转轴(axis_angular)，这些控制按键、摇杆的映射关系。

3.保持按住手柄油门键（下图中的１０号键），现在使用左侧的推杆可以控制小车的前后移动和转向（下图中的１＋１－摇杆）

遥控使能10号键
遥控使能11号键
前进
加速
左转
右转
前进
后退
左转
右转
减速
后退

0
3
12
4
17
7
5
15
13
6
14
1
16
2

根据这些按键编号，可以修改launch文件中的相关参数从而改变按键映射关系

4.使用手柄对主机进行关机

同时按住手柄的12和14号键,手柄会振动，然后松手，在振动消失前再次长按，会触发主机关机同时振动消失。

5.常见问题

```
1.ps4手柄和ubuntu无法连接
答：之前有没有配对成功，没有配对成功要先确保配对成功。
2.ps4手柄和ubuntu无法连接
答：等白灯熄灭后，再次按ps键尝试连接，因为蓝牙适配器性能原因，有时要尝试两三次才能连上，这是正常现象。
3.ps4手柄和ubuntu无法连接
答：配对成功后，如果再次同时按了share和ps键，会清除手柄的配对信息，此时需要先在ubuntu里面删除叫wireless Controller的蓝牙设备，然后重启执行步骤0的配对。
```

小强主页 返回目录

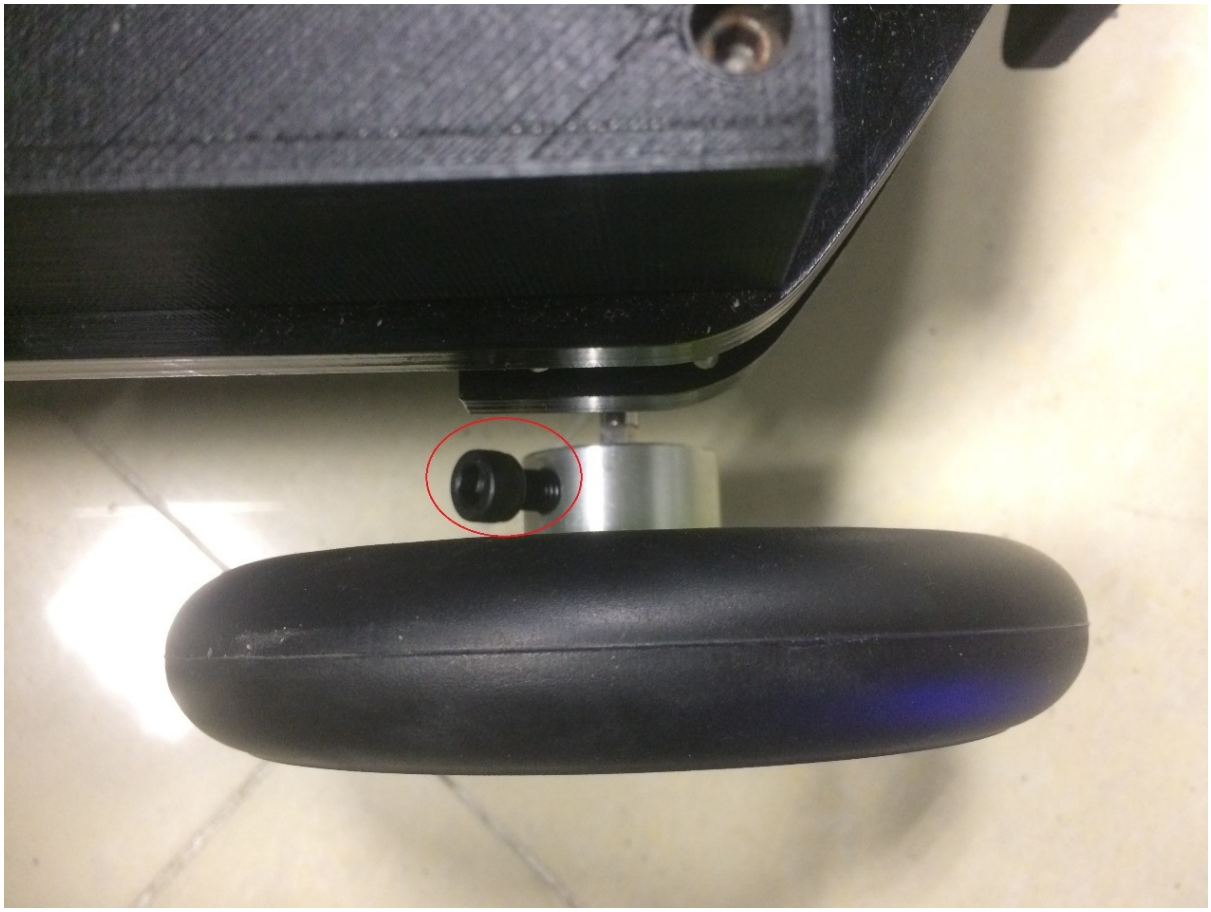
- 充电

充电

将电池与车底盘的连接断开后，用配送的电池专用充电器充电，充满电需要5个小时左右，充满后指示灯会由红色变青色。 电池的两根输出线内部是并联在一起的，因此两根头都可以用于充电和放电（充电器只有一个公头），电池支持同时充放电

- 车轮松动打滑

车轮松动打滑



解决办法：请重新拧紧上图中的螺丝

- 小强底层固件下载和升级办法
  - 小强v4_3.hex ，右键"另存为..."下载后重名为xiaoqiang.hex
    - 固件是普通hex文件，支持任何stm32下载器。stm32烧录后，请根据升级底盘ros驱动包xqserial_server完成上位机更新操作。最后根据这篇教程重新校准底盘IMU。
    - 下文介绍使用小强自带的u转串烧写stm32的方法，经过本次升级，以后可以使用这篇帖子中的简便升级方法。

小强底层固件下载和升级办法

2017年3月11日之后收到小强的用户，请参考这篇帖子升级固件，无法升级时请询问客服后再使用下面的方法。手动升级有风险，操作前请咨询客服。

小强v4_3.hex ，右键"另存为..."下载后重名为xiaoqiang.hex

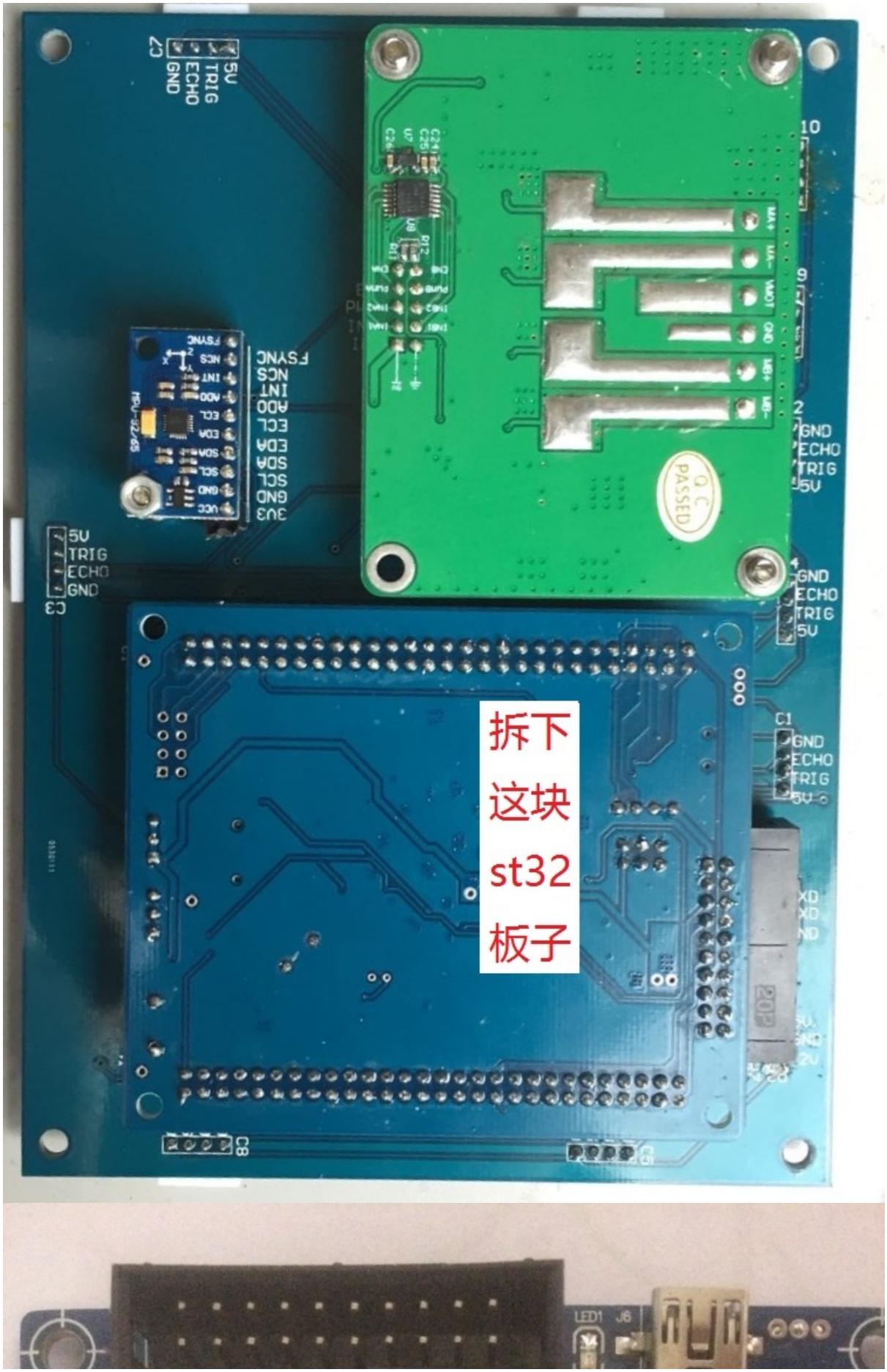
固件是普通hex文件，支持任何stm32下载器。stm32烧录后，请根据升级底盘ros驱动包xqserial_server完成上位机更新操作。最后根据这篇教程重新校准底盘IMU。

下文介绍使用小强自带的u转串烧写stm32的方法，经过本次升级，以后可以使用这篇帖子中的简便升级方法。

1.找到1台windows电脑，下载安装这份stm32flash软件

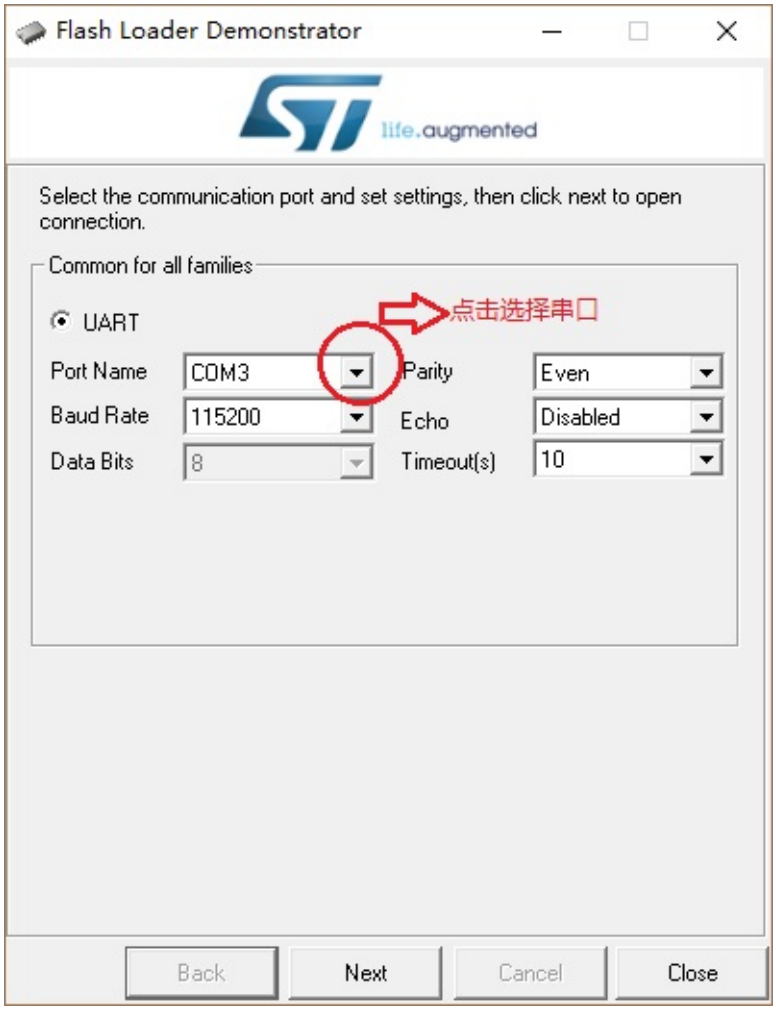
flash_loader_demo_v2.8.0.exe ,右键选择“另存为...”启动下载

2.小强底盘断电后，先拔下绿色电机驱动板上中间的“VMOT GND”接线，慢慢拔下底盘上的stm32核心板，然后按照下图调整板子上的两个短接帽

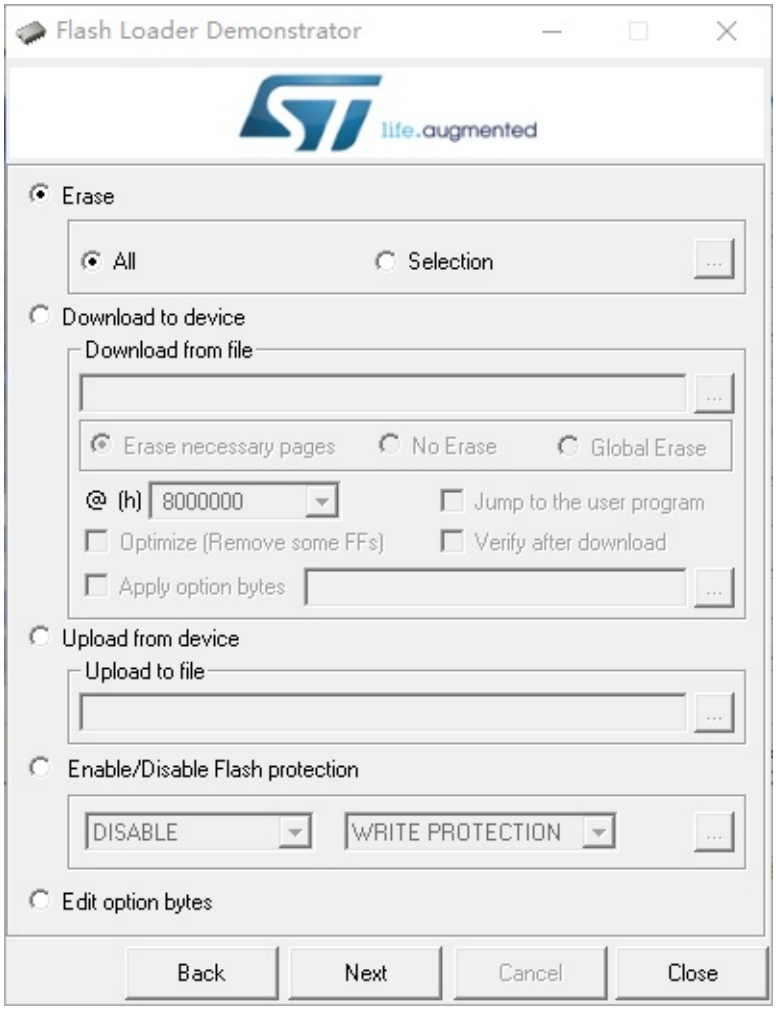

拆下
这块
st32
板子

3.将2中调整完后的stm32板子插回底盘，注意引脚需要全部对准插入基座，然后给底盘上电。

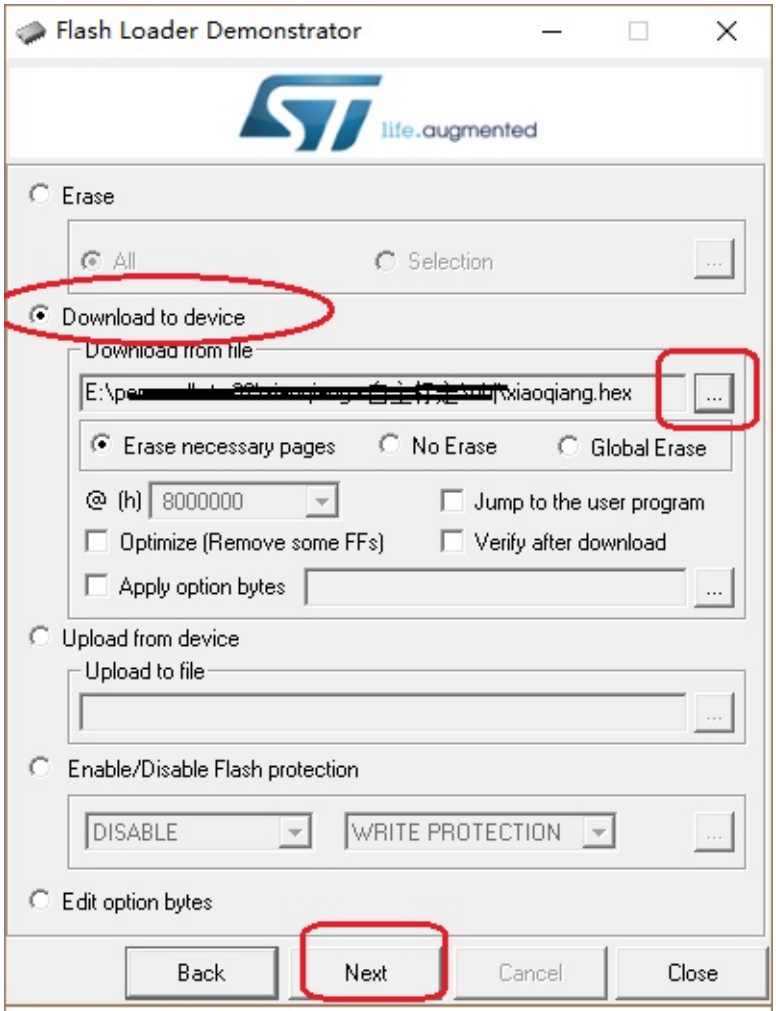
4.将小车上的u转串模块usb头从小车主机上拔下后插入windows电脑，从开始程序菜单打开Flash Loader Demo程序，所有程序->STMicroelectronics-> Demonstrator GUI，出现如下界面



5.选择默认设置即可，一直点next(如果点击next失败，重新给底盘上电后再试)，出现如下界面：



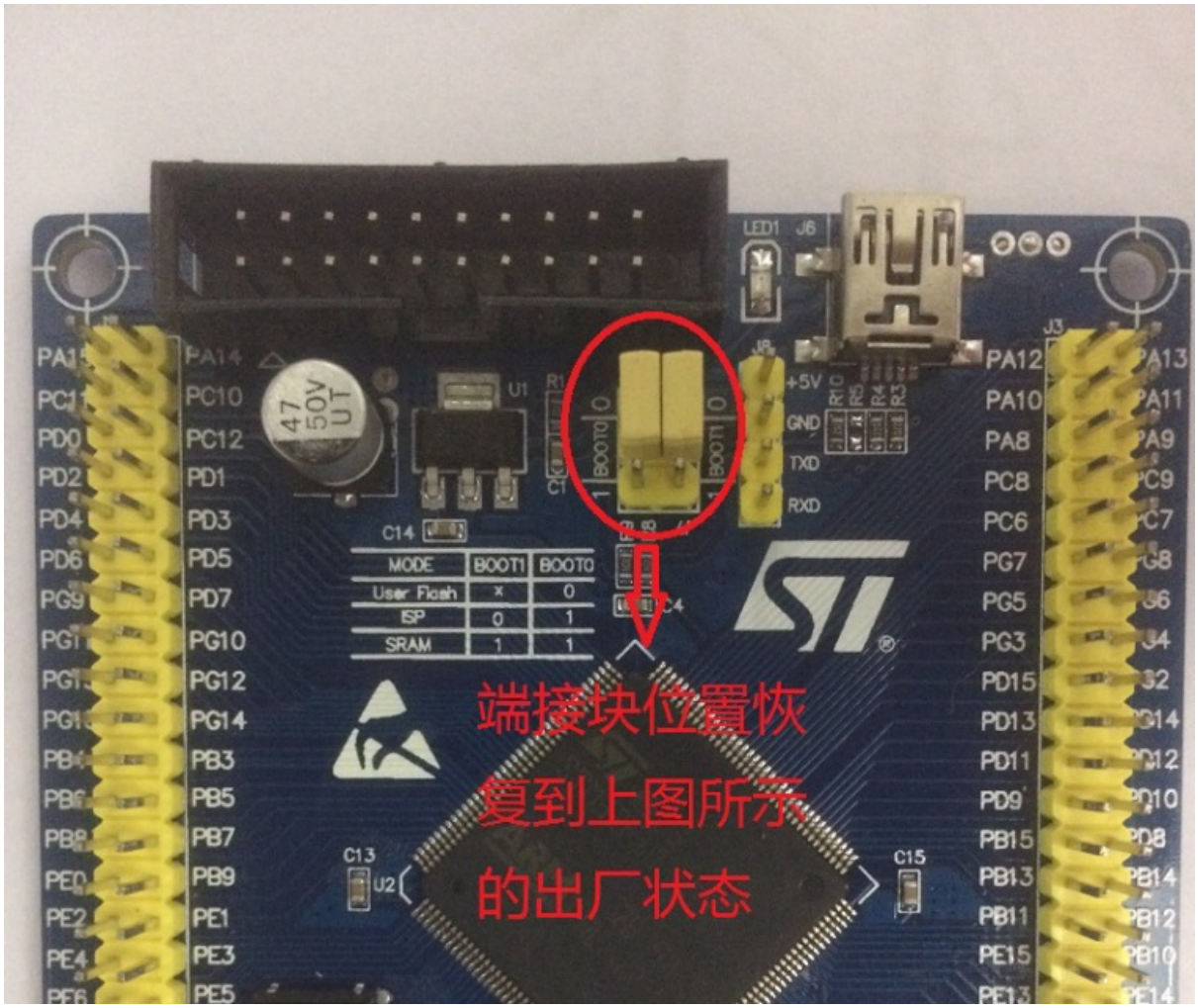
6.选择Download to device，选择上面提供的 hex文件，点Next开始下载


Flash Loader Demonstrator
life.augmented
Erase
All
Selection
Download to device
Download from file
E:\pe[redacted]\xiaoqiang.hex
Erase necessary pages
No Erase
Global Erase
@ (h)
8000000
Jump to the user program
Optimize (Remove some FFs)
Verify after download
Apply option bytes
Upload from device
Upload to file
Enable/Disable Flash protection
DISABLE
WRITE PROTECTION
Edit option bytes
Back
Next
Cancel
Close

Flash Loader Demonstrator
life.augmented
Target STM32F1_High-density_512K
Map file STM32F1_High-density_512K.STmap
Operation DOWNLOAD
File name E:\personal\stm32\xiaoqiang - 自主标定\obj\xiaoqiang.hex
File size 65.29 KB (66860 bytes)
Status 65.29 KB (66860 bytes) of 65.29 KB (66860 bytes)
Time 00:11
Download operation finished successfully

7.烧录完成，关闭软件，小车底盘断电，将u转串插回小车主机。

8.再次慢慢拔下底盘上的stm32核心板，调整板子上的两个短接帽恢复到初始状态后，插回底盘（引脚需要全部对准插入基座），将绿色电机驱动板上中间的“VMOT GND”接线插上，stm32升级完成！



9.小车底盘重新上电，根据文章开头提示完成上位机的升级。

- 升级底盘ros驱动包xqserial_server

升级底盘ros驱动包xqserial_server

1.ssh登录小强主机，进入小强ros工作目录

```
ssh xiaoqiang@192.168.xxx.xxx   #请将xxx.xxx换成实际ip
cd Documents/ros/src/
```

2.进入ros驱动包xqserial_server,更新软件

```
cd xqserial_server/
git stash
git pull
cd ..
cd ..
catkin_make -DCMAKE_BUILD_TYPE=Release -DCATKIN_WHITELIST_PACKAGES="xqserial_server"
```

3.重启ros节点，更新完成

```
sudo service startup stop
sudo service startup start
```

- 重新校准小车底盘IMU

重新校准小车底盘IMU

Sorry：本教程适用2016年9月之后购买的用户

适用情况：

发货前，每台小车的底盘IMU都已经校准好，理论上正常使用是不用重新校准地。如果小车经过长期使用后，发现底盘输出的odom角度开始存在严重飘逸现象，请按下述步骤重新校准底盘IMU

操作步骤：

0.把小车水平静止放置好，下面标定过程不能移动、碰撞小车

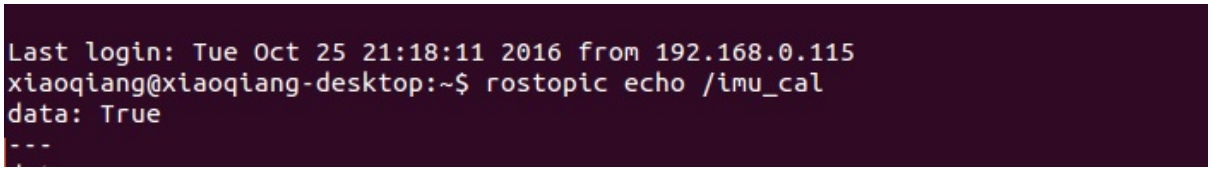
1.在本地虚拟机中ssh登录小车，输入下述命令

```
ssh xiaoqiang@192.168.0.xxx   -X
rostopic echo /imu_cal
```

2.在本地虚拟机中新开一个窗口再次ssh登录小车，输入下述命令

```
ssh xiaoqiang@192.168.0.xxx   -X
rostopic pub /imu_cal std_msgs/Bool '{data: true}' -1
```

3.等待10s，当步骤1中的窗口出现下图时，说明小车已经启动校准程序，终止步骤2中的topic发布命令



4.等待2分钟，IMU重新标定完成，现在无需重启小车即可继续正常使用

- 蓝鲸ROS镜像发布
  - 系统配置要求
  - Noetic 最新版本
  - Kinetic 最新版本
  - Jade最新版本
    - 在虚拟机中安装
    - 在实际的电脑上安装

蓝鲸ROS镜像发布

本文列出所有已发布的蓝鲸ROS系统镜像，用户可以根据自己需要进行选择。 蓝鲸ROS是蓝鲸智能机器人公司在Ubuntu ROS的基础上进行修改而制作的ROS系统镜像。其中包含了ROS系统和常用的一些ROS软件包。非常适合ROS学习和开发人员使用。系统安装完成后不需要再安装配置ROS，即可直接使用。蓝鲸ROS也是小强的系统镜像，可以安装在小强上直接使用。

注意在虚拟机上安装推荐使用Vmware Player

系统配置要求

内存最低1.5G推荐2G 存储空间至少30G

Noetic 最新版本

xq4 pro 和xq5通用版本：xiaoqiang-noetic_2021-11-12.iso 下载地址

版本 `xq5-noetic_2020-10-29` 下载地址

版本 `xq5-noetic_2020-07-16` 下载地址 md5: 3EF9EE329283B3009B22EF1E7C9A856D

已知问题:

在未插显示器和视频转接头的情况下VNC远程会黑屏。这个是Gnome的问题，无法很好的解决。 如果想在无显示器和转接头情况下VNC访问可以试试这里的方法。这个方法也有一些问题，比如无法播放声音。

Kinetic 最新版本

版本 `xq_os_v2.0.10-2018-11-08.iso(kinetic版本)`
`可以在小强Pro上使用，也可以在自己电脑上使用`
下载地址
md5: 6E093DDF851852C50316F86A279DB22B

说明： `注意此镜像更改了默认的udev规则，设备和usb端口绑定。串口插在对应的USB位置上才能识别。如果安装完成镜像后无法遥控，查看/dev文件夹下是否有ttyUSB001,如果没有则把U转串换一个USB接口。直到出现为止。` 增加奥比中光支持

版本 `xq_os_v2.0.11_xq5-2019-12-21.iso.` (kinetic版本)
`可以在XQ5上使用，也可以在自己的电脑上使用`
下载地址
md5: 1E9DB219A28936CA2366BAB2CC4D41F9

版本 `xq_os_v2.0.7_mini_2019_01_15.iso(kinetic版本)`
`可以在小强mini上使用，也可以在自己电脑上使用`
下载地址
md5: 4A37DA0FCC09866615A5CAEBE91B4CE9

Jade最新版本

版本 `xq_os_v1.0.3_2017-10-31`
`可以在小强Pro上使用，也可以在自己电脑上使用`
下载地址
md5: afbf1c8026733c4f8063778846c84e9b

版本 `xq_os_v1.0.4_mini_2017-11-17`
`只能在mini上使用`
下载地址
md5: e874c78088211dff3feabd92f86275e7

注意事项

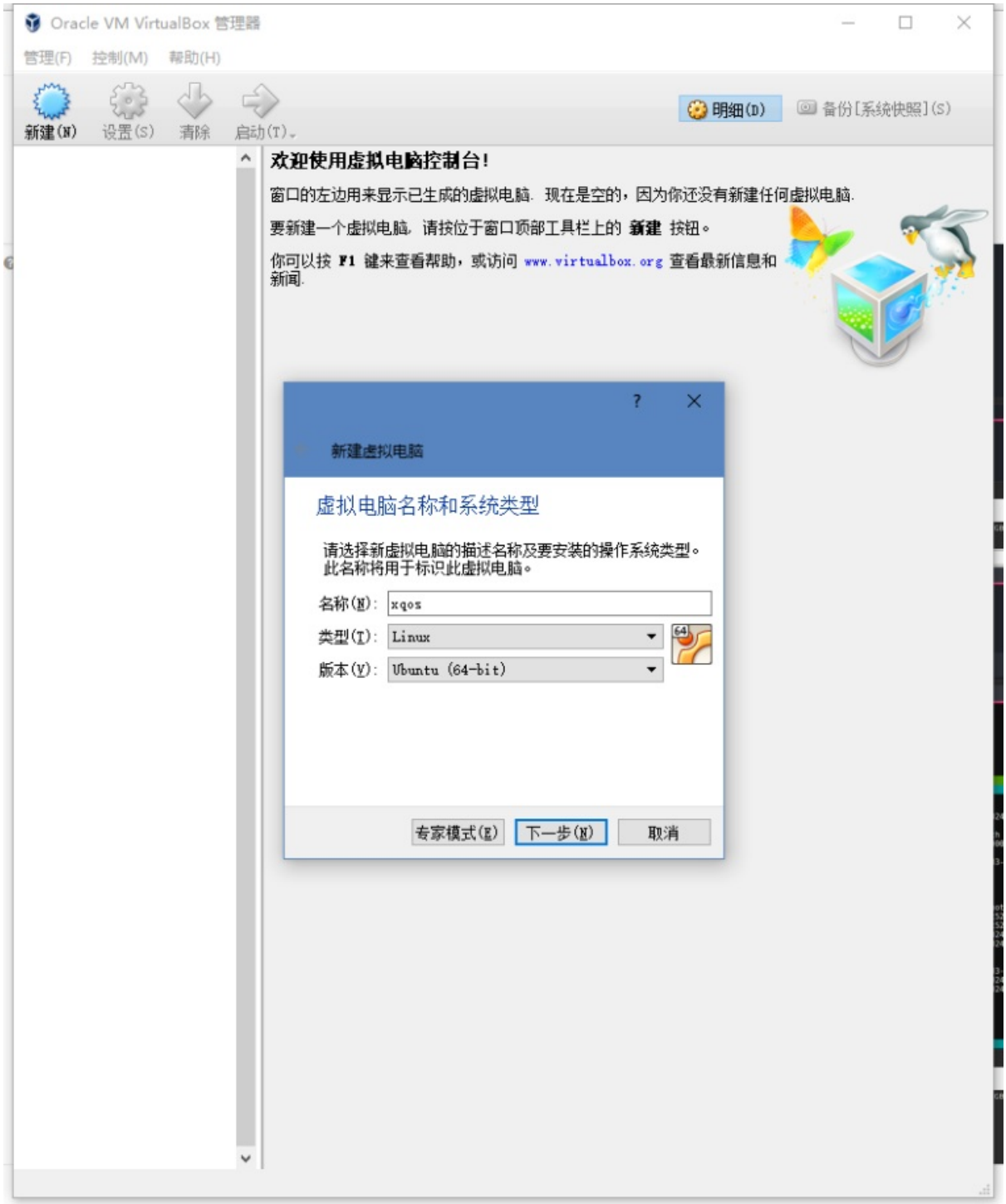
下载完成之后一定要做一下md5的检查，确认文件完整性 注意安装时的用户名只能是xiaoqiang,否则会有问题

镜像安装方法

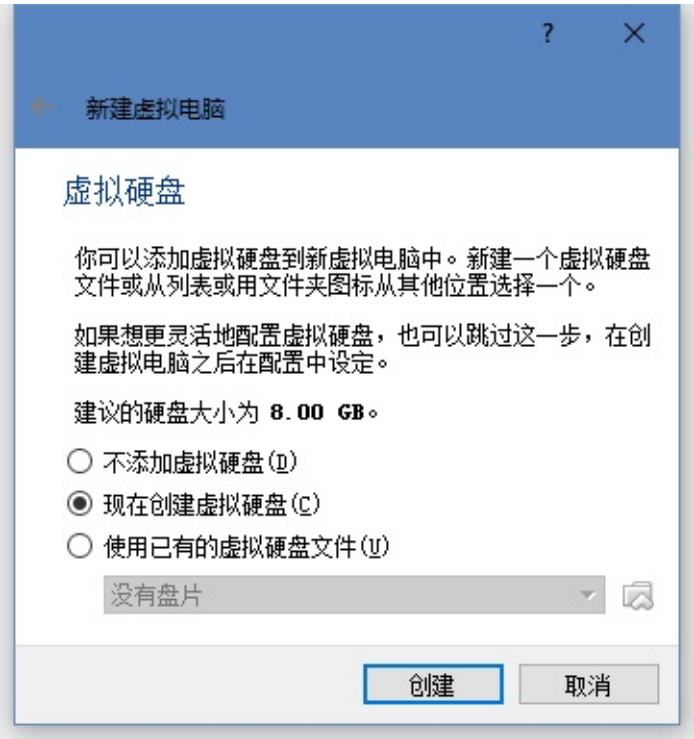
在虚拟机中安装

打开虚拟机软件，这里使用的是Virtualbox。下载地址

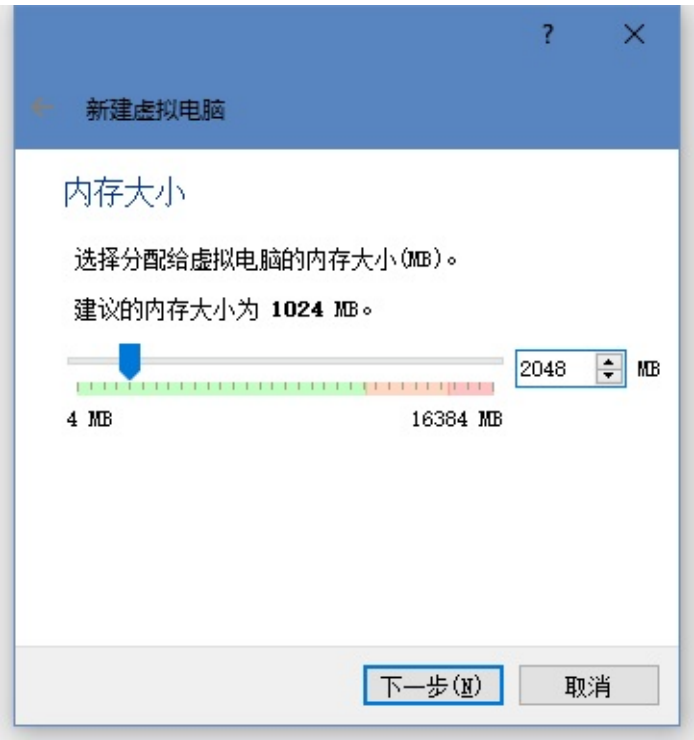
点击创建按钮，开始创建虚拟机 注意在系统选择列表中选择 `Ubuntu 64`

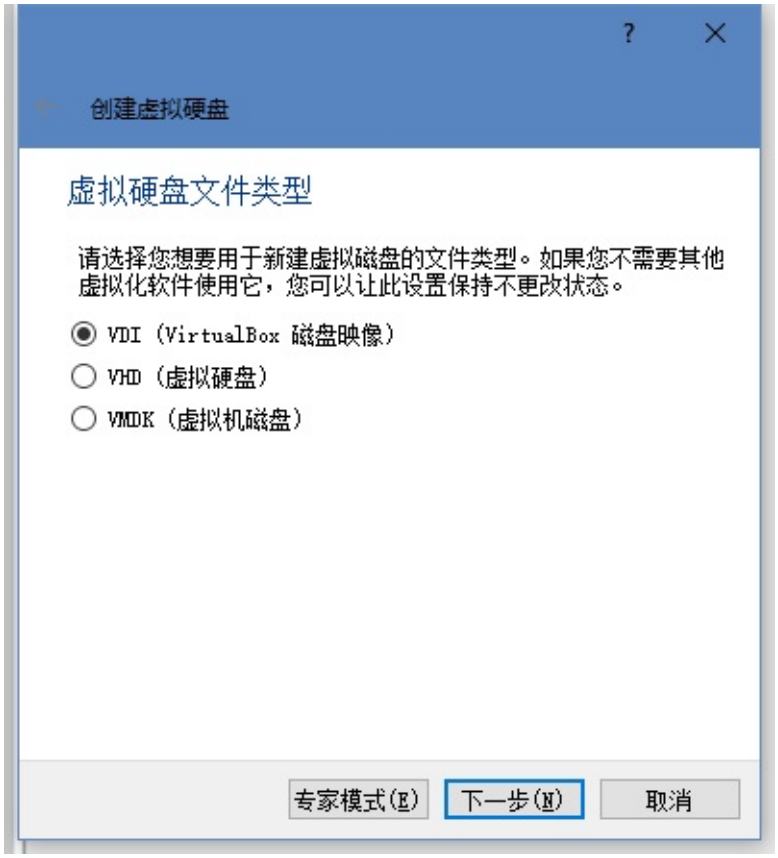


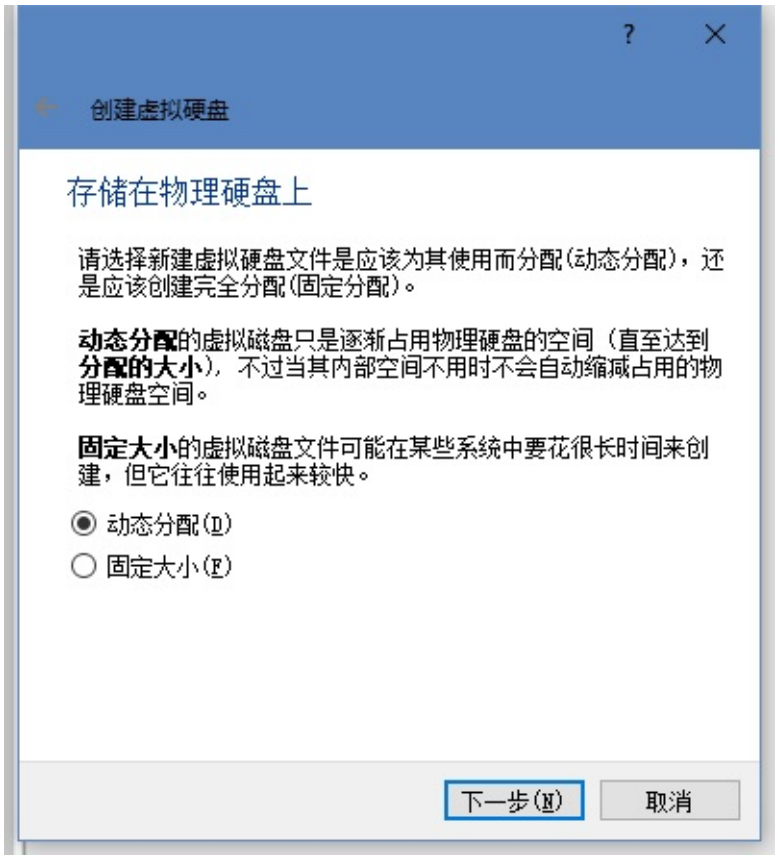
按照图上设置，完成后点击下一步



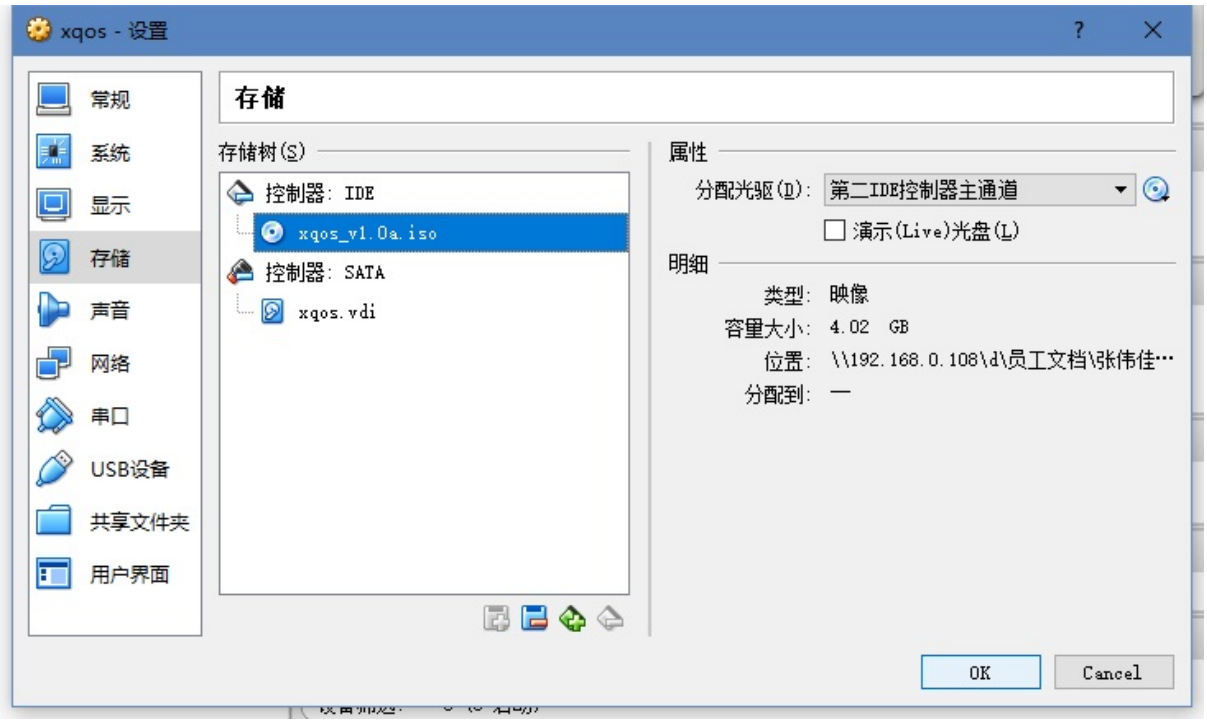
内存建议设置成2G，内存太小可能会导致无法启动



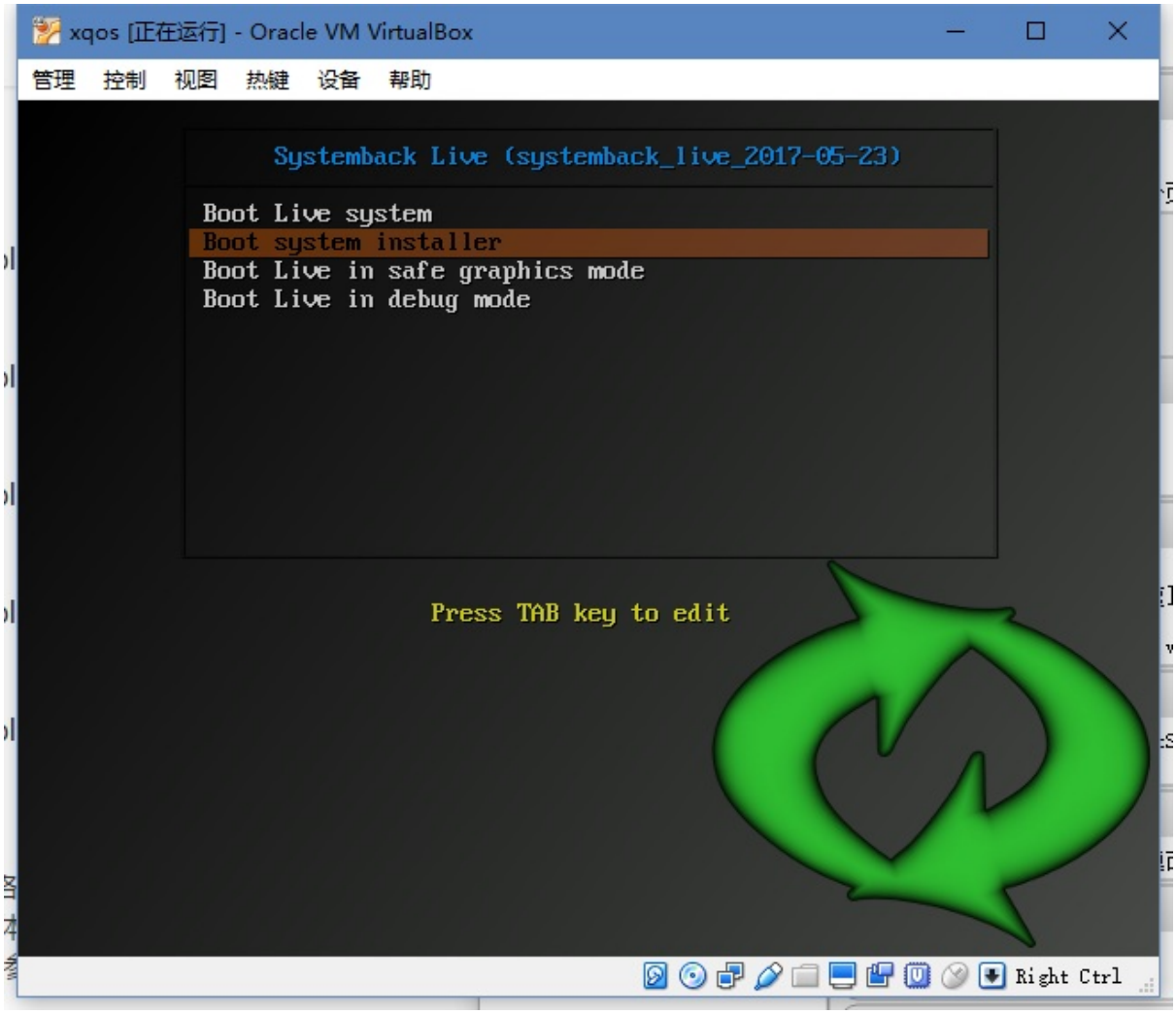

创建虚拟硬盘
虚拟硬盘文件类型
请选择您想要用于新建虚拟磁盘的文件类型。如果您不需要其他虚拟化软件使用它，您可以让此设置保持不更改状态。
VDI (VirtualBox 磁盘映像)
VHD (虚拟硬盘)
VMDK (虚拟机磁盘)
专家模式(E)
下一步(N)
取消


创建虚拟硬盘
存储在物理硬盘上
请选择新建虚拟硬盘文件是应该为其使用而分配(动态分配)，还是应该创建完全分配(固定分配)。
动态分配的虚拟磁盘只是逐渐占用物理硬盘的空间（直至达到分配的大小），不过当其内部空间不用时不会自动缩减占用的物理硬盘空间。
固定大小的虚拟磁盘文件可能在某些系统中要花很长时间来创建，但它往往使用起来较快。
动态分配(D)
固定大小(F)
下一步(N)
取消

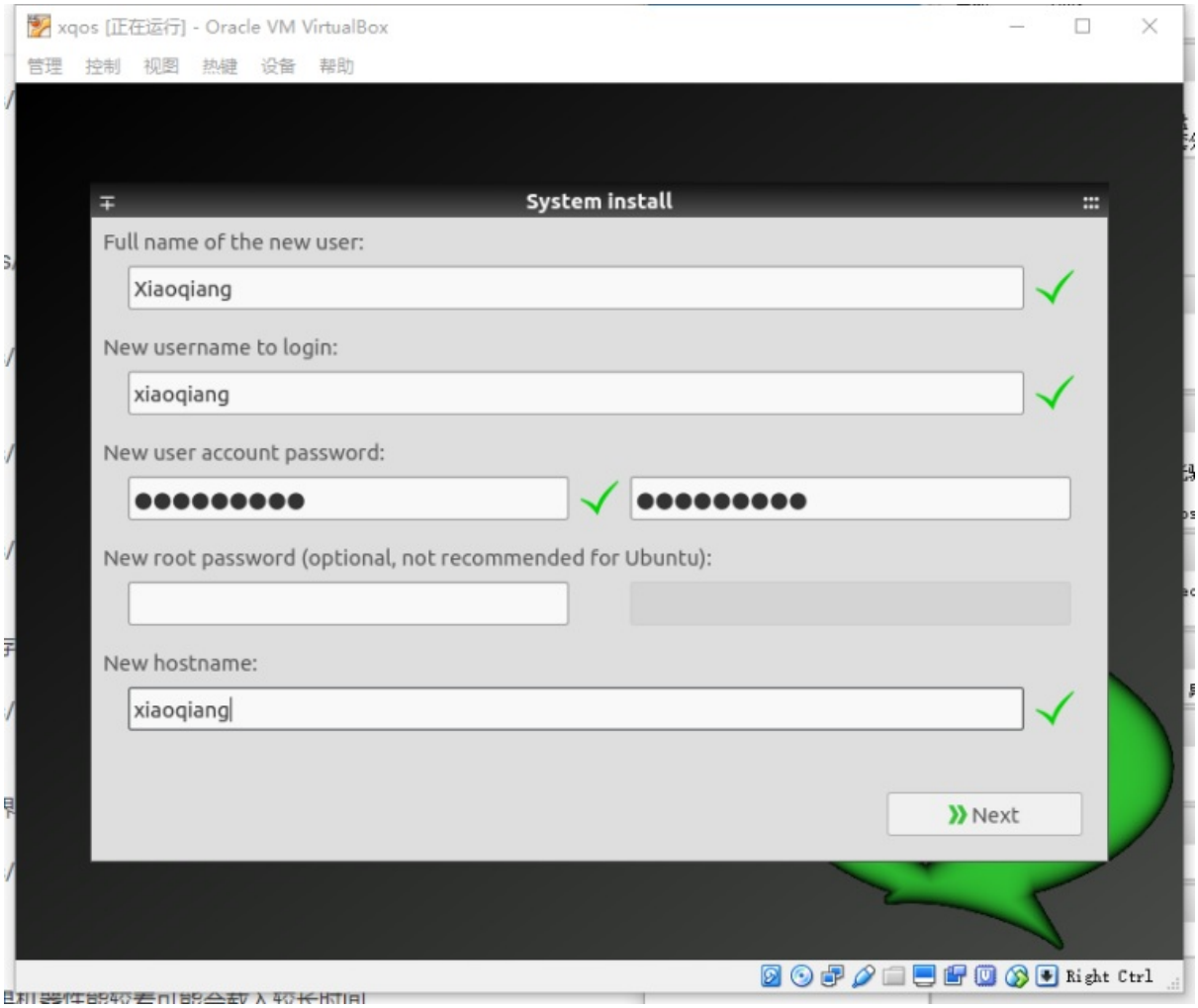
设置完成后，点击上面的设置按钮。在弹出的对话框中选择存储。点击存储界面右侧的光盘图标设置镜像文件



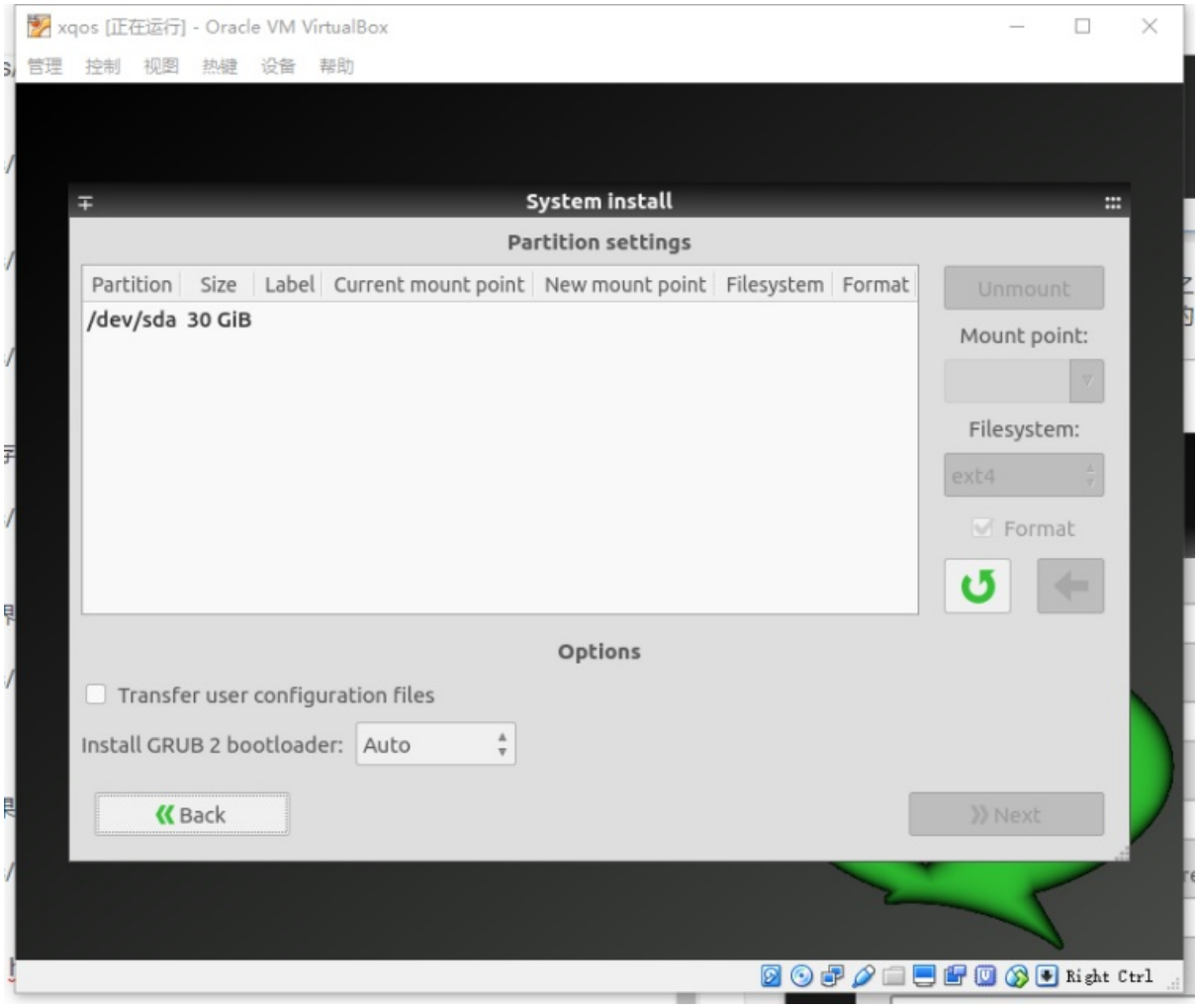
设置完成之后点击主界面的启动按钮。等待载入系统的选择界面。



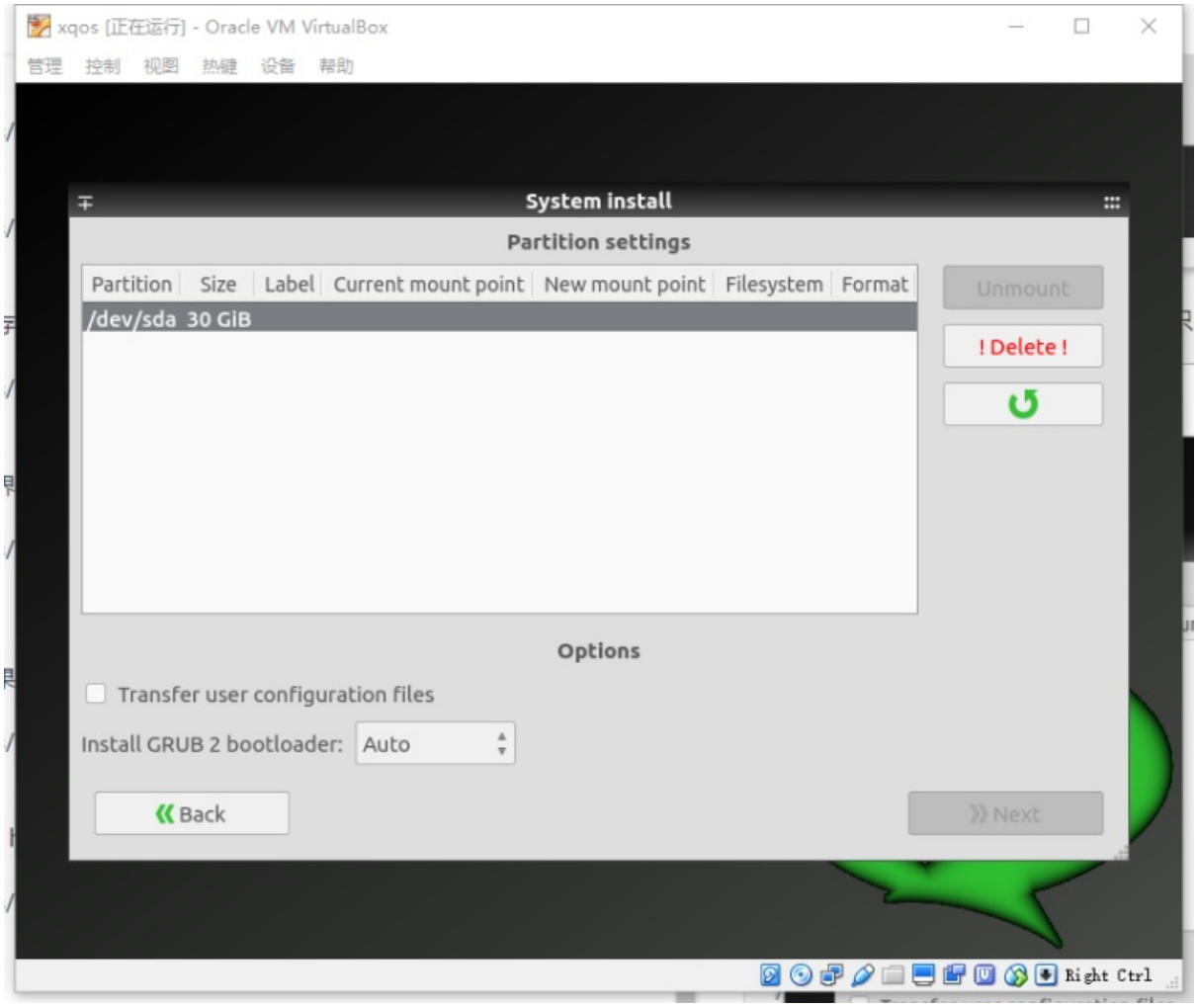
如果想要安装系统则选择第二个选项，之后按回车确认。 等待系统安装程序载入，这个过程和你的机器性能相关，如果机器性能较差可能会载入较长时间。



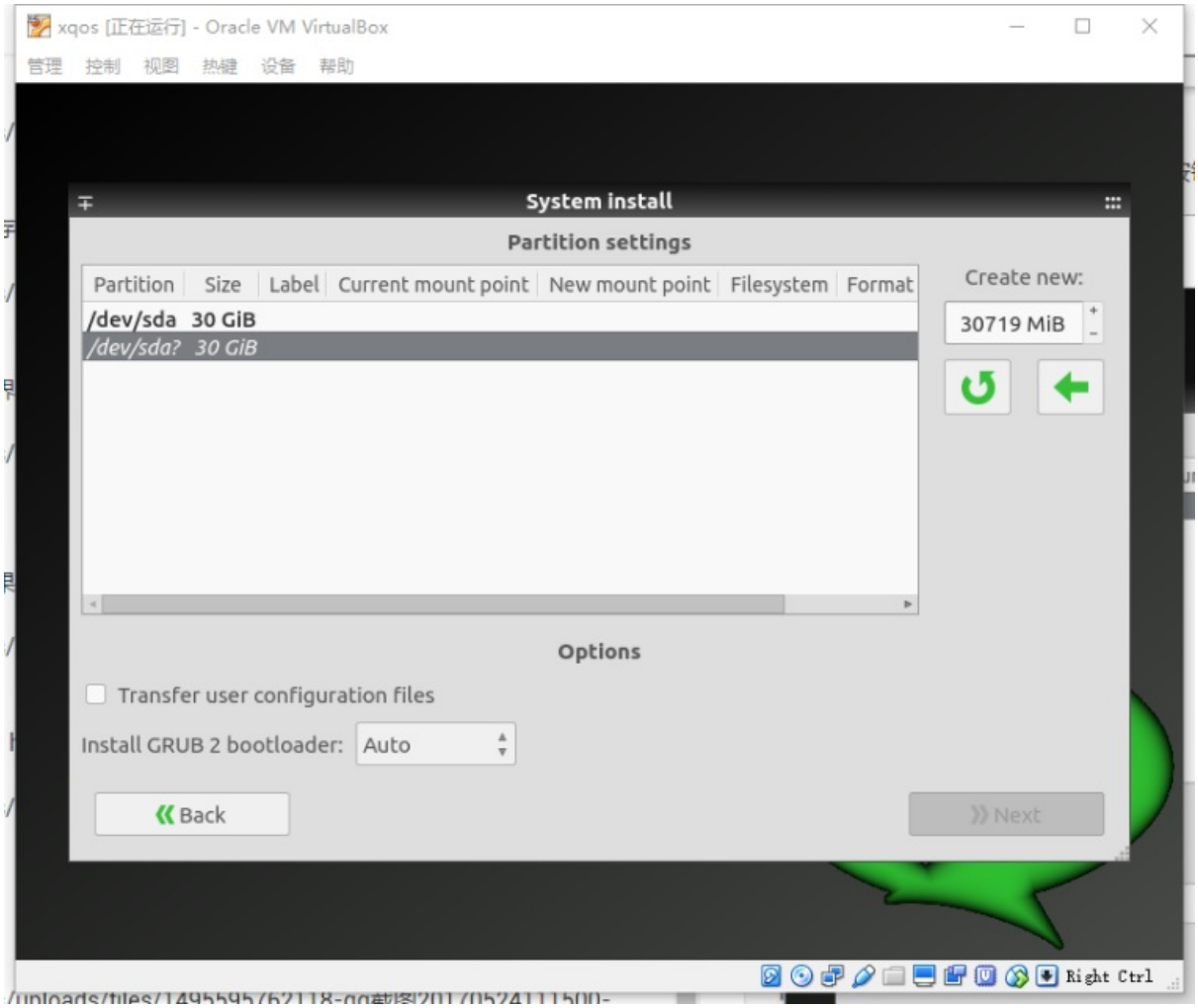
载入完成之后设置用户信息，注意用户只能是 `xiaoqiang` ，hostname可以随意设置。



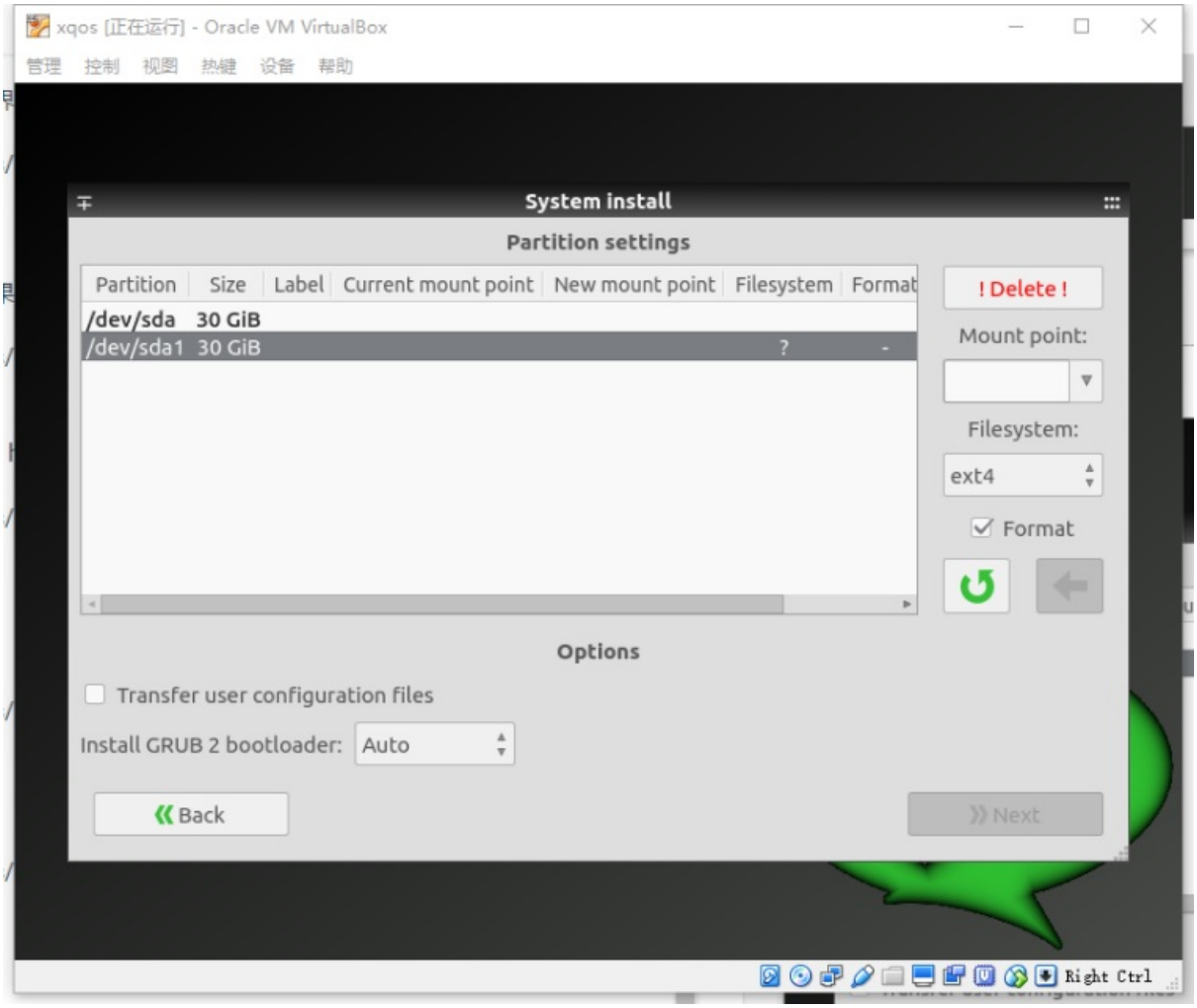
点击下一步进入分区界面。 选中你要安装的硬盘，然后点击Delete按钮



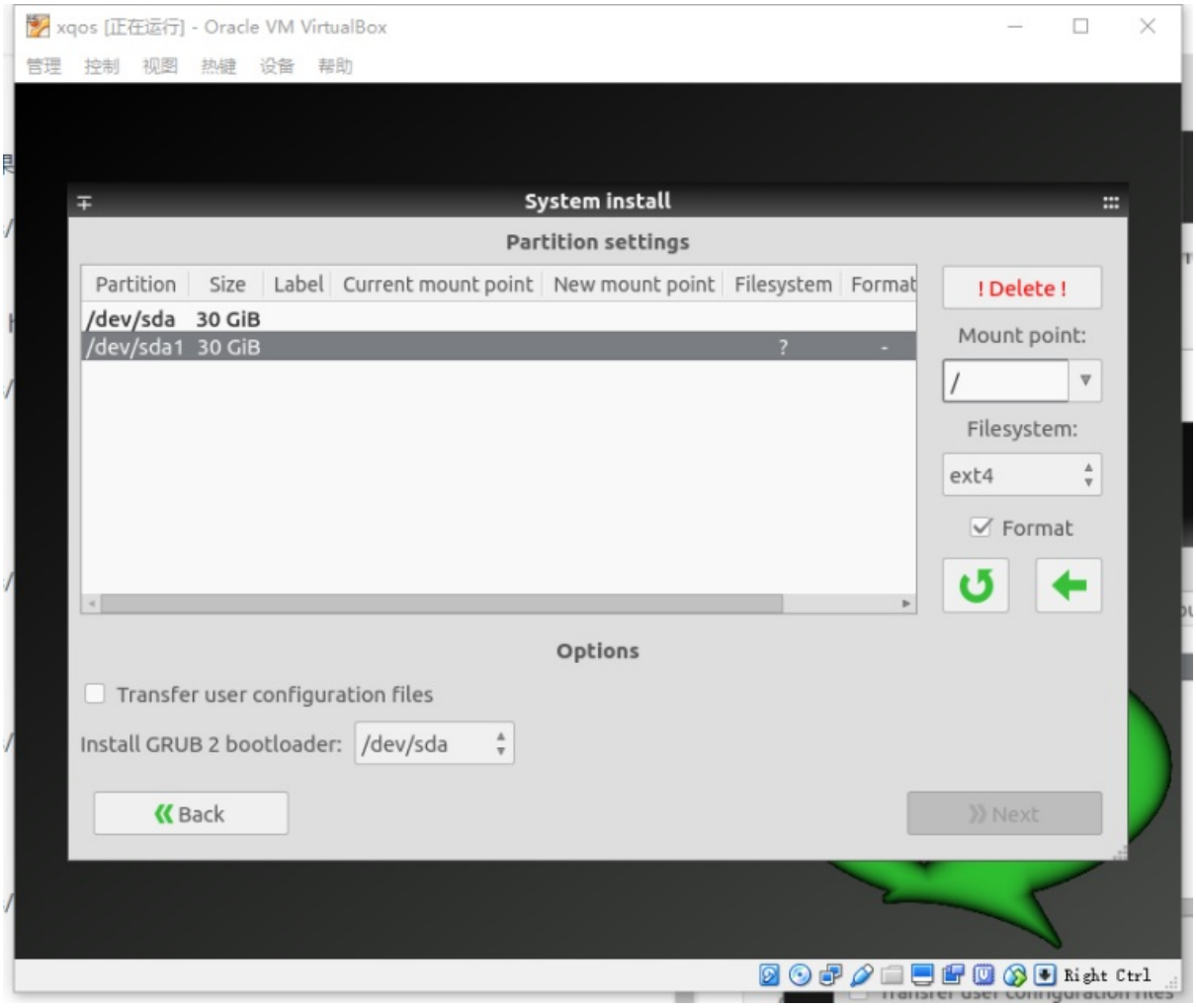
再次选择硬盘中的新分区


xqos [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
System install
Partition settings
Partition Size Label Current mount point New mount point Filesystem Format
/dev/sda 30 GiB
/dev/sda2 30 GiB
Create new:
30719 MiB
Options
Transfer user configuration files
Install GRUB 2 bootloader: Auto
Back
Next
Right Ctrl

点击箭头进入下一步



再次选择刚才新建的分区 设置右侧的Mount point，再次点击箭头进入下一步

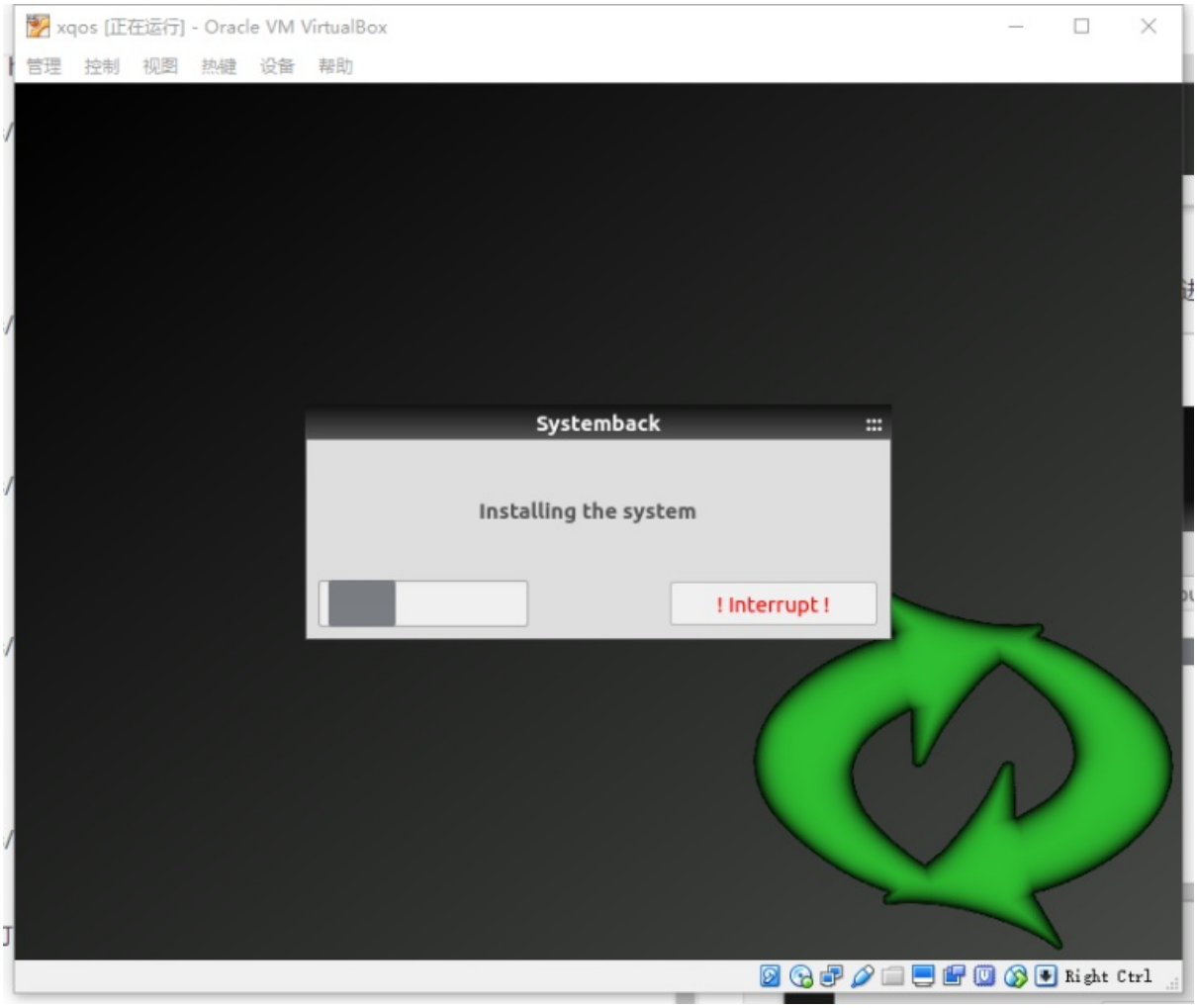


勾选transfer user configuration files

注意在实际的电脑上安装时，分区的时候，要给boot单独的分区，然后选择下面的grub安装位置。否则系统安装完成后是无法启动的。对于不同的启动方式分区也有些不同。以grub方式启动的要给/boot单独分区，以uefi方式启动的要给/boot/efi单独分区。具体的分区信息可以看Install GRUB 2 bootloader的右侧的提示信息。注意一定要保证Install GRUB 2 bootloader不是disable状态。

在给实际的电脑安装的时候，推荐的分区方式是 1G给 /boot/efi, 4G给swap，剩下的全部分给/。

点击右下角的next就可以继续了。如果你还想设置其他分区可以按照以上的步骤进行设置。



等待系统安装完成后重新启动就可以了。

在实际的电脑上安装

安装过程和一般的安装Ubuntu是一样的。下面介绍一下如何用U盘安装系统。 首先你要在自己的硬盘上准备一个空白的分区用来安装系统。 然后下载ultroiso软件。 下载地址 在电脑上插入自己的U盘。 用软件打开iso文件。在菜单栏中选择启动 -> 写入硬盘镜像，弹出界面如下

点击写入，等待写入完成。

写入完成之后，把U盘插入想要安装系统的电脑上。启动时进入BIOS选项，选择从U盘启动。后面的安装过程就基本上是一样的了。

注意在分区的时候，要给boot单独的分区，然后选择下面的grub安装位置。否则系统安装完成后是无法启动的。对于不同的启动方式分区也有些不同。以grub方式启动的要给/boot单独分区，以uefi方式启动的要给/boot/efi单独分区。一般情况下用U盘安装都需要给/boot/efi单独分区。

虚拟机软件下载 Vmware Player

- 如何恢复小强中被修改的软件包

如何恢复小强中被修改的软件包

在使用小强的过程中可能会对小强中的相关软件进行修改。很容易出现修改后小强不能正常工作的情况。下面介绍如何恢复小强的相关软件。

小强的相关软件都是通过git进行代码管理的。任何对代码的更改都是有迹可循的，也是可以恢复的。下面以startup软件包为例进行说明。

假设我们修改了小强的startup软件包，导致有些启动项没有正常运行。如何查看我们做了哪些修改呢？可以通过下面的指令

```
# 进入startup 软件包所在目录
roscd startup
# 查看我们进行了哪些修改
git diff
```

如果有修改我们可以看到下面的输出

```
xiaoqiang@xiaoqiang-desktop:~/Documents/ros/src/startup$ git diff
diff --git a/launch/startup.launch b/launch/startup.launch
index 5fa2900..814d037 100755
--- a/launch/startup.launch
+++ b/launch/startup.launch
@@ -2,7 +2,7 @@
   <include file="$(find xiaoqiang_udrf)/launch/xiaoqiang_udrf.launch" />
   <include file="$(find xqserial_server)/launch/xqserial.launch" />
   <include file="$(find galileo_serial_server)/launch/galileo_serial_server.lau
nch" />
-  <include file="$(find usb_cam)/launch/ov2610.launch" />
+  <!-- <include file="$(find usb_cam)/launch/ov2610.launch" /> -->
   <node pkg="system_monitor" type="monitor.py" name="system_monitor" respawn="t
rue" respawn_delay="10">
     <remap from="/usb_cam" to="/camera_node" />
     <remap from="/usb_cam/camera_info" to="/camera_node/camera_info" />
xiaoqiang@xiaoqiang-desktop:~/Documents/ros/src/startup$
```

从这里我们可以看出usb_cam包的launch文件被注释掉了。这样小强就不会开机自动启动摄像头了。

如果我们想要恢复到原来的文件状态，可以执行下面的指令。

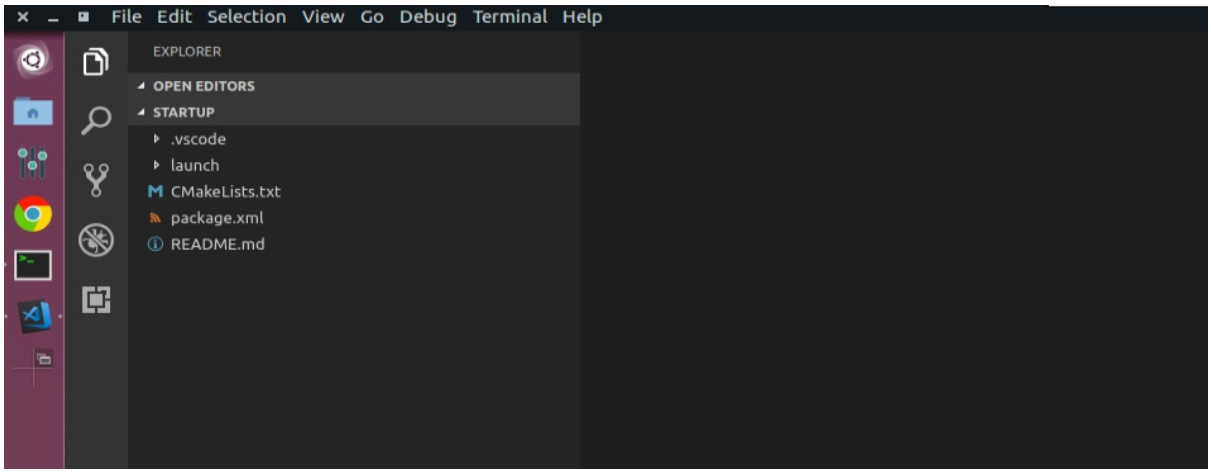
```
# 进入startup 软件包
roscd startup
# 撤销所有更改
git stash
```

```
xiaoqiang@xiaoqiang-desktop:~/Documents/ros/src/startup$ git stash
Saved working directory and index state WIP on service_bot: 0670ba3 add camera i
nfo
HEAD is now at 0670ba3 add camera info
xiaoqiang@xiaoqiang-desktop:~/Documents/ros/src/startup$ git diff
xiaoqiang@xiaoqiang-desktop:~/Documents/ros/src/startup$
```

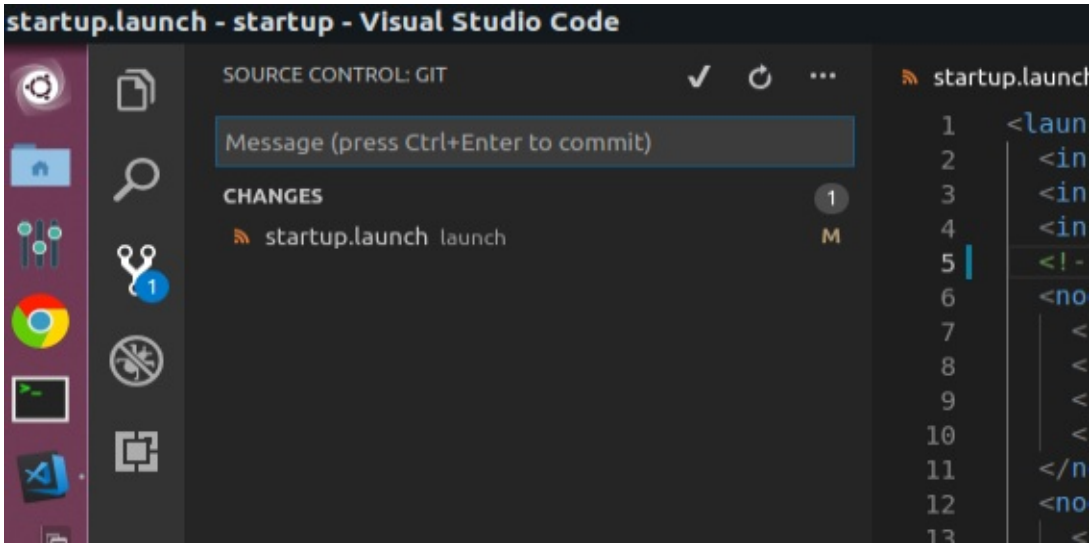
可以看到刚才的更改已经被撤销了。

如果你不习惯使用命令行进行操作，我们也可以使用vscode进行操作。

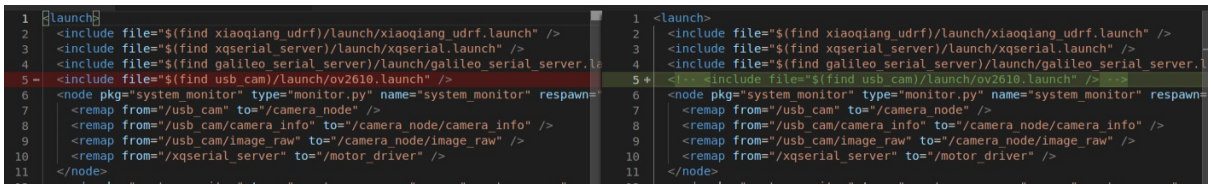
通过vscode打开startup软件包。



点击左侧第三个按钮，进入代码管理界面

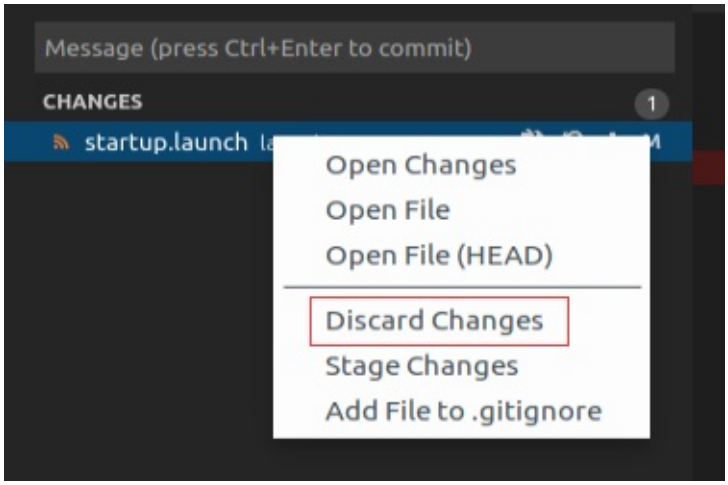


可以看到这里列出了startup.launch文件被修改 想要查看修改内容，只需要点击对应的文件即可。



可以看到usb_cam那一行被标记修改过了。

如果想要撤销修改可以右键单击文件，在弹出的菜单中选择撤销更改。



以上是代码的恢复。但是仅恢复代码有时小强还是无法正常工作。对于startup软件包来说，要需要重新注册启动项后，更改才能生效。对于启动项的注册可以查看小强用户手册。

对于需要编译的代码，代码需要在重新编译后才会生效。比如 `xqserial_server` 在进行更改后需要

```
# 进入小强工作空间
cd ~/Documents/ros
# 编译xqserial_server软件包
catkin_make -DCATKIN_WHITELIST_PACKAGES="xqserial_server"
```

编译成功后 `sudo service startup restart` 这样更改才能生效。

对于进行了很多修改，而以上恢复办法仍无法让小强正常工作时，那只能用重装系统大法了

- 如何在重装系统后恢复xiaoqiang设置

如何在重装系统后恢复xiaoqiang设置

请在重装或重置完系统后，打开终端然后执行以下操作（如果只是想更新小强系统可以直接执行bwupdate）：

```
sudo apt-get update
sudo apt-fast upgrade -y
cd Documents/ros/src/bwbotupdater
git pull
# 若是noetic版本的系统镜像则需要切换至v4分支
git checkout v3
cd ~
```

接下来切换到xiaoqiang-cmds执行下一步操作。如果你是xq4，请切换到xq4分支；如果是xq5用户请切换到xq5（这一步为16.04系统和20.04系统通用）：

```
cd Documents/xiaoqiang-cmds
git pull
git checkout 机器型号
sudo ./fixudev
```

最后执行如下指令（如果你是xq4用户，则为bwupdate xq4；是xq5则为bwupdate xq5。如果你使用的是20.04系统版本的小强，请使用bwupdate xq4-noetic或bwupdate xq5-noetic）

```
bwupdate 机器型号
```

耐心等待系统更新完成，当终端显示更新完成且指令完成后即更新完成。完成更新后，如果是重装系统的用户请咨询客服重新获取伽利略凭证。

常见问题： Q：sudo apt-get update以及sudo apt-fast upgrade -y执行过慢。 A：有可能是没有执行加速指令，在论坛中搜索“加速”即可；也有可能是单纯的速度慢，请耐心等待。

Q：bwupdate 过程中报错，例如“**Install python dependencies failed**”。 A：示例错误是最常见的报错，意为“**ros package安装python依赖项失败**”。如果是安装某个包出现这个问题，请重新执行bwupdate指令至安装成功；如果仍有这个问题至ros目录下单独编译这个包，然后再继续bwupdate。其他编译错误也可以尝试切换至ros目录下单独编译这个包再继续bwupdate。

- 如何获取伽利略证书
  - 获取证书
  - 证书的使用
  - 常见的问题

如何获取伽利略证书

伽利略导航系统在运行时会检查证书。小强用户可以向管理员免费索取。下面介绍一下证书的获取和使用方法

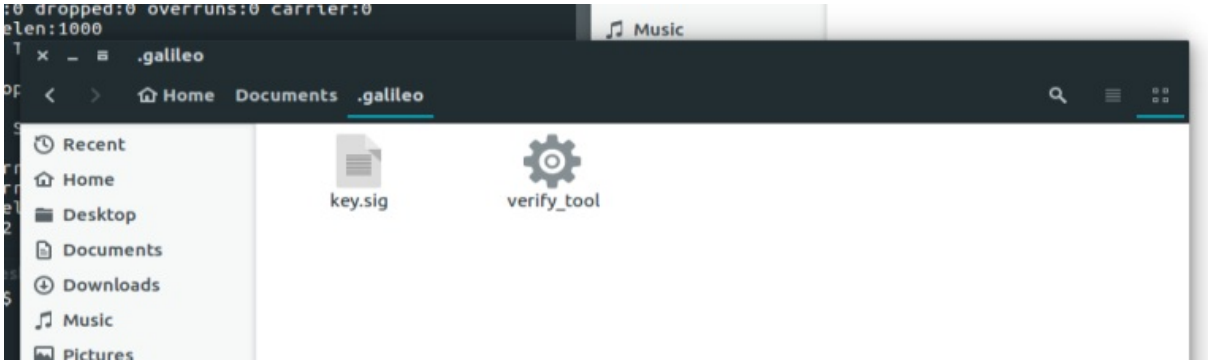
获取证书

通过QQ联系管理员(1024313822),在自己的小强上运行ifconfig,并把结果截图发给管理员

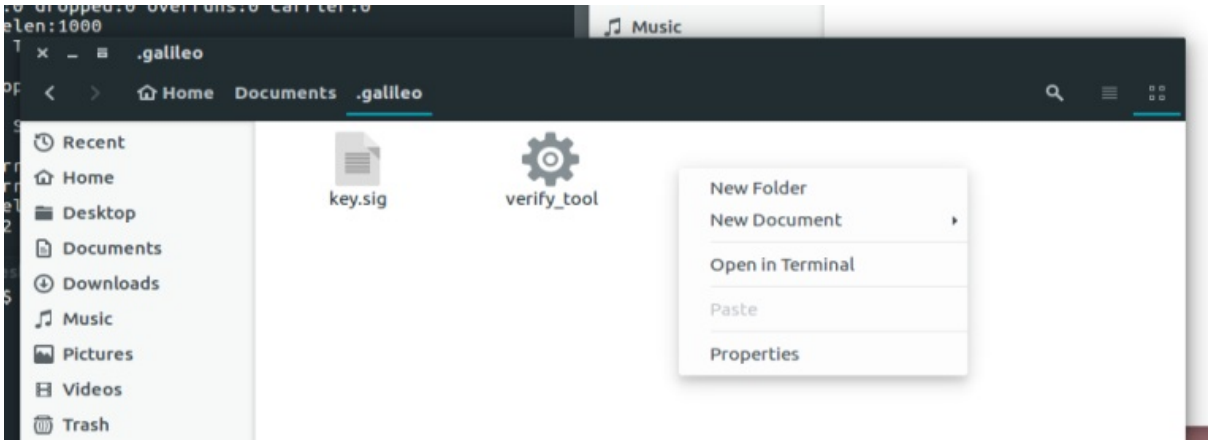
管理员之后会给你发送证书文件

证书的使用

把证书文件 `key.sig` 文件放入 `Documents/.galileo` 文件夹里面。然后运行此文件夹中的 `verify_tool` 文件。注意 `.galileo` 是一个隐藏文件夹。可以通过按 Ctrl + H显示隐藏文件夹。

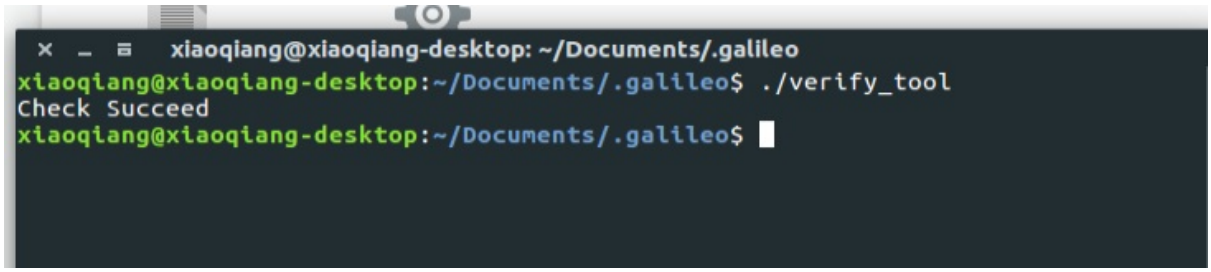


在此文件夹按住shift同时点击鼠标右键,在菜单中选择Open in Terminal



输入

```
./verify_tool
```



如果证书没问题，就会有如上的输出。

常见的问题

Q: 证书验证失败 A: 小强Pro用户需要保证的自己的网络设备有 `enp2s0` 设备。在有些特殊情况下，有时候会没有对应的网络设备。可以参照这里的文档添加udev规则修改网络设备名称。

比如对于小强mini用户可以创建 `/etc/udev/rules.d/70-persistent-net.rules` 文件。在文件中添加

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:e0:4c:68:4a:ff", ATTR{dev_id}=="0x0", ATTR{type}=="1", NAME="enp2s0"
```

其中 `00:e0:4c:68:4a:ff` 替换成自己的mac地址。修改完后重启生效。

Q: 需要什么版本的系统 A: 目前只支持Ubuntu 16.04。对于14.04以及比较老的系统需要更新系统后使用。更新系统请参考这里

- Ubuntu 如何设置静态IP

Ubuntu 如何设置静态IP

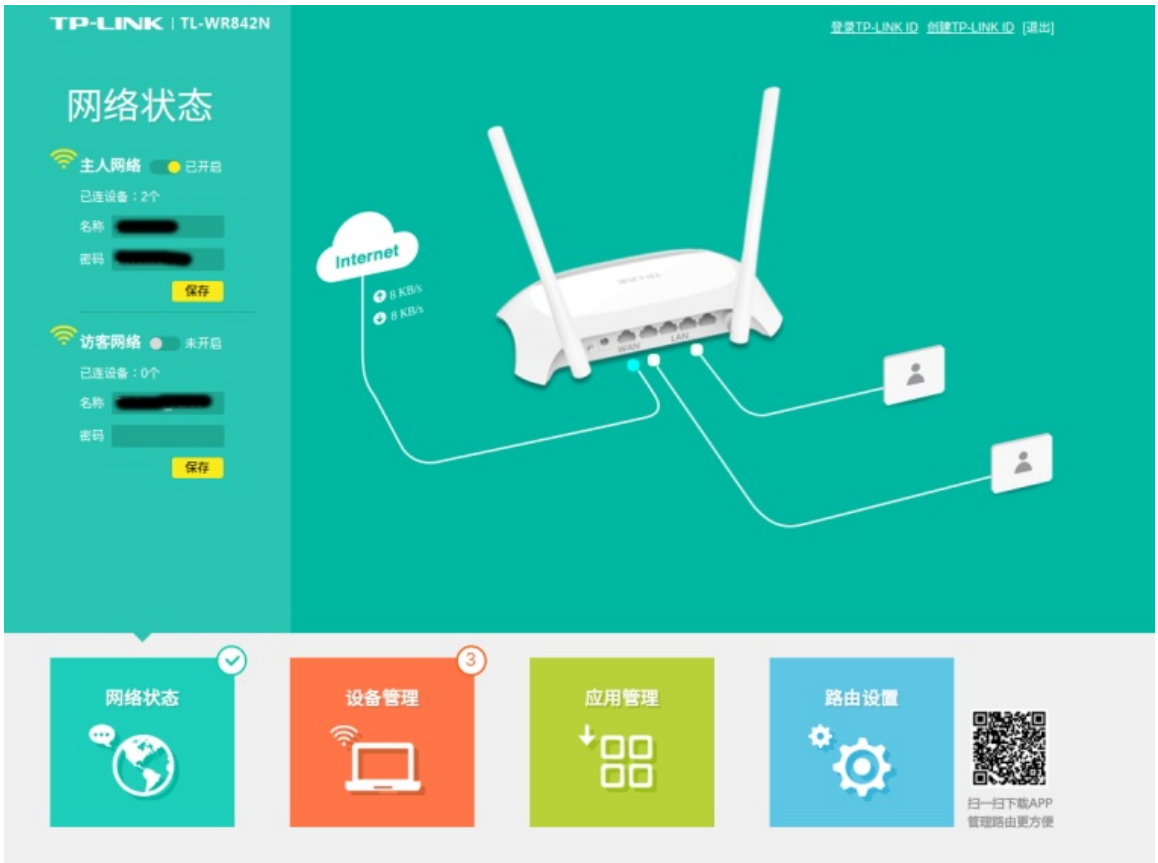
在网络上搜索设置静态IP出现的都是通过修改文件进行设置的。这样的方式很容易让电脑的网络不能使用。推荐使用下面的方法进行修改

IP分配的基本方式

IP分配的方式一般是由路由器决定的。路由器有DHCP方式和静态IP两种方式。DHCP就是动态的分配IP方式。一般路由器默认的就是这种模式。在这种模式下电脑也可以设置自己的静态IP。当然并不能保证一定会成功，比如自己设定的IP可能被别人占用。也有可能路由器由于迷之原因不给你分配自己设定的IP。下面就具体说一下各种设置静态IP的方法。

设置静态IP的几种方式

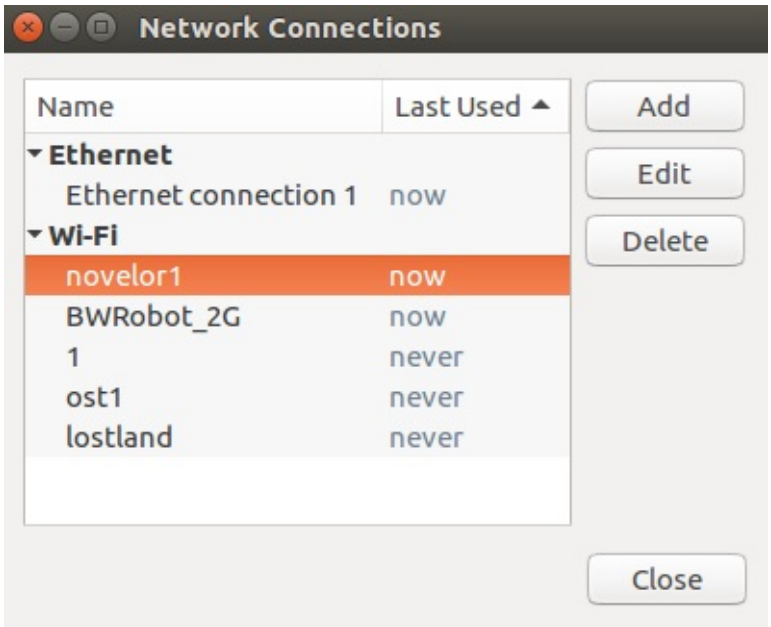
1. 路由器设置法 通过路由器设置静态IP是最简单的方法。不过前提是你要有路由器的管理权限，路由器也要支持这种设置功能。 首先在浏览器中输入路由的地址，进入路由器的管理界面。



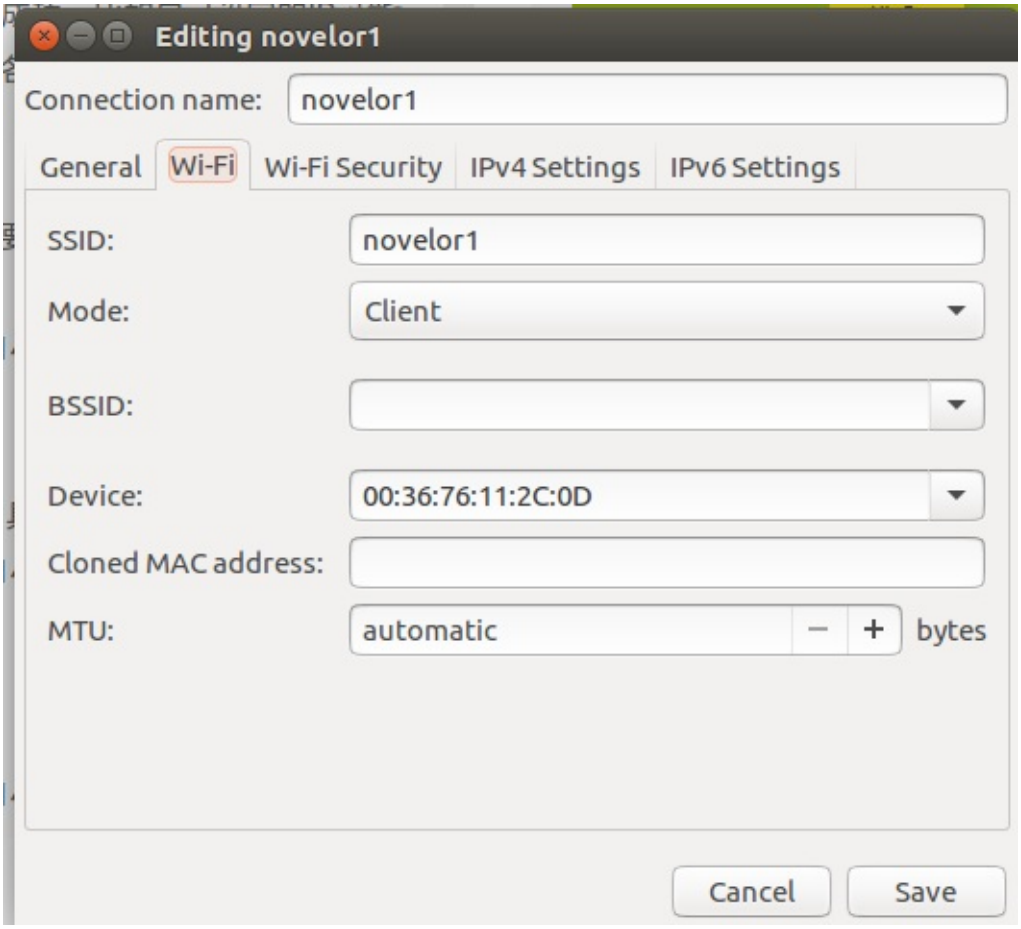
在应用管理中有IP和MAC绑定。不同的路由器设置界面不一样，你可以找一找这个功能的具体位置。

在这里就可以设置静态IP了

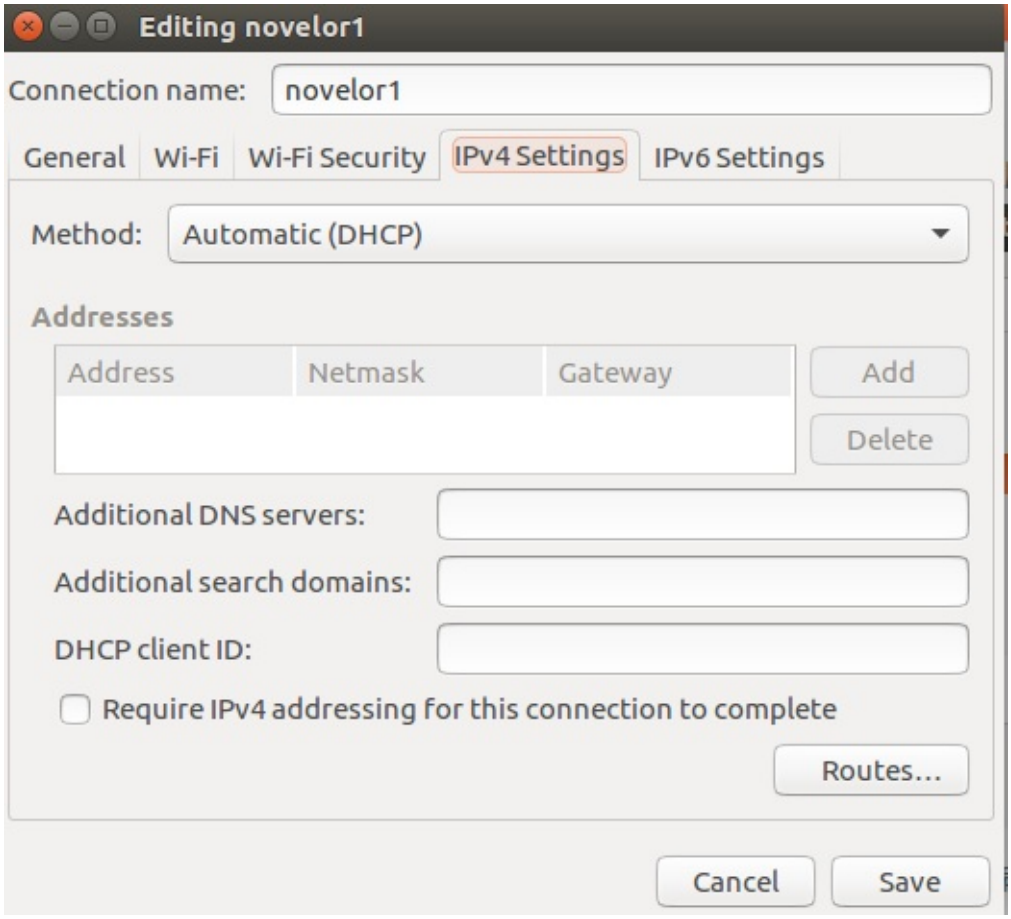
1. 在电脑上进行设置 电脑连接上网络后，打开电脑的网络管理器



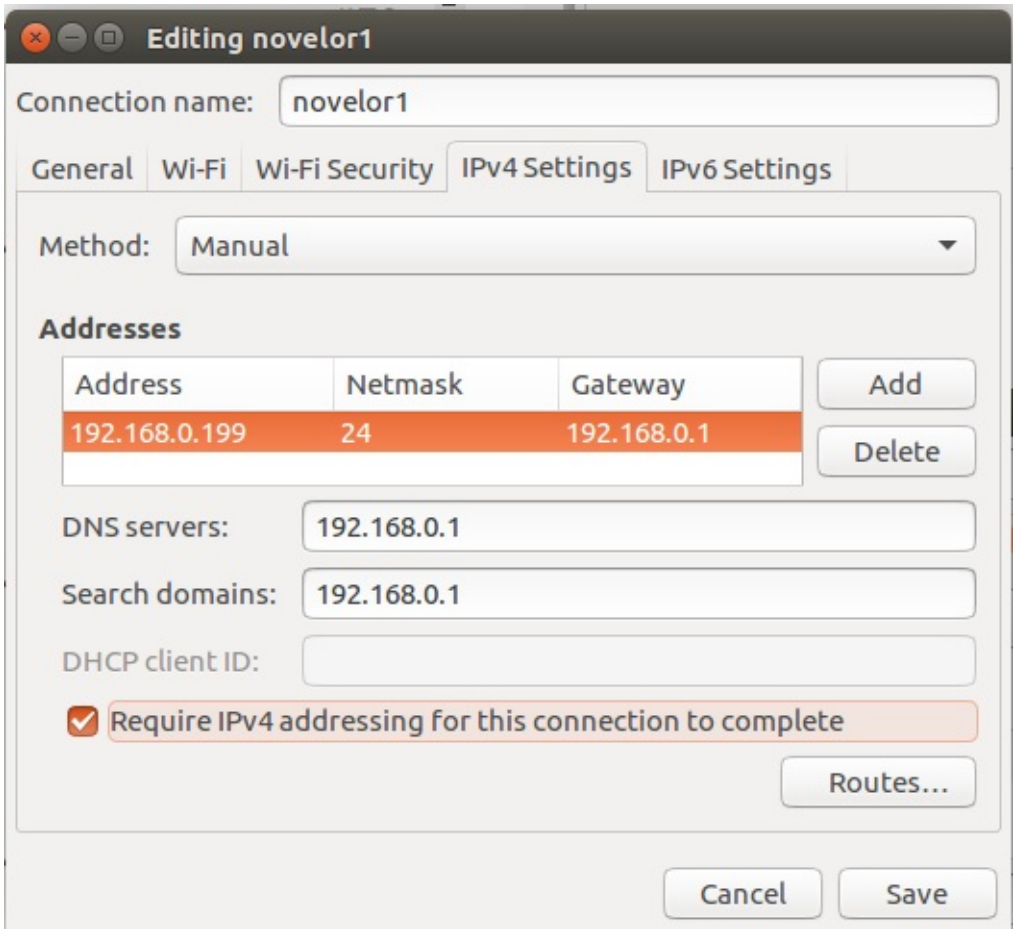
选中自己当前的网络链接，选择编辑


Editing novelor1
Connection name:
novelor1
General
Wi-Fi
Wi-Fi Security
IPv4 Settings
IPv6 Settings
SSID:
novelor1
Mode:
Client
BSSID:
Device:
00:36:76:11:2C:0D
Cloned MAC address:
MTU:
automatic
bytes
Cancel
Save

点击IPv4设置


Editing novelor1
Connection name:
novelor1
General
Wi-Fi
Wi-Fi Security
IPv4 Settings
IPv6 Settings
Method:
Automatic (DHCP)
Addresses
Address
Netmask
Gateway
Add
Delete
Additional DNS servers:
Additional search domains:
DHCP client ID:
Require IPv4 addressing for this connection to complete
Routes...
Cancel
Save

按照下图这样进行设置



其中192.168.0.199是你想要设置的本机IP，要保证不能和局域网中的其他IP重复。192.168.0.1是路由器的地址，这个要根据自己的网络情况进行调整。 设置完成后点击保存就可以了

然后断开网络重新连接 一定要重新连接后才能生效

```
randoms@nowhere:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr fc:aa:14:a1:af:f0
          inet addr:192.168.0.104  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::feaa:14ff:fea1:aff0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:109449 errors:0 dropped:0 overruns:0 frame:0
          TX packets:111052 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53269931 (53.2 MB)  TX bytes:30415250 (30.4 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:27928 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27928 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:8926946 (8.9 MB)  TX bytes:8926946 (8.9 MB)

wlan1     Link encap:Ethernet  HWaddr 00:36:76:11:2c:0d
          inet addr:192.168.0.199  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::b4e4:1ad2:575d:d856/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:468 errors:0 dropped:0 overruns:0 frame:0
          TX packets:293 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:85815 (85.8 KB)  TX bytes:49309 (49.3 KB)

randoms@nowhere:~$
```

输入ifconfig查看当前的网络信息，可以发现我的无线网络已经设置成刚才的那个IP了。

- 视觉导航路径编辑器使用教程

视觉导航路径编辑器使用教程

利用小强可以建立出周围环境的三维地图，但是如何利用这个地图实现视觉循迹呢？视觉导航路径编辑器就是为了实现这个功能而编写的。通过这个软件你可以在三维空间中标记你想要小强运动的轨迹。然后将生成的轨迹文件导出给小强，小强就能按照你标定的轨迹进行运动了。下面就详细介绍一下软件的使用方法。

安装

软件提供了Ubuntu 的deb安装包 64位deb包下载 源代码在这里

下载完成后执行一下指令安装 `注意这个路径编辑器已经和新版本的导航程序不兼容了，对于新版本的导航程序请使用Windows客户端(目前仅对巡检版开放)`

```
sudo dpkg -i path-drawer_1.0.0_amd64.deb
```

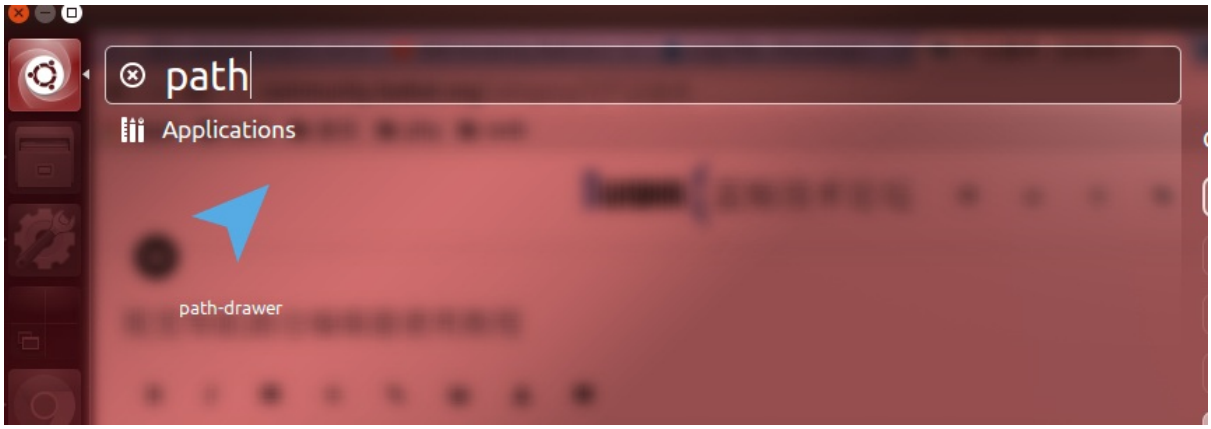
等待安装完成即可

采集空间数据

路径编辑器需要载入小强采集的空间数据才能够进行操作，详细的操作可以参考这一篇。 点击保存按钮之后，地图信息会被保存到 `/home/xiaoqiang/slamdb` 文件夹内。

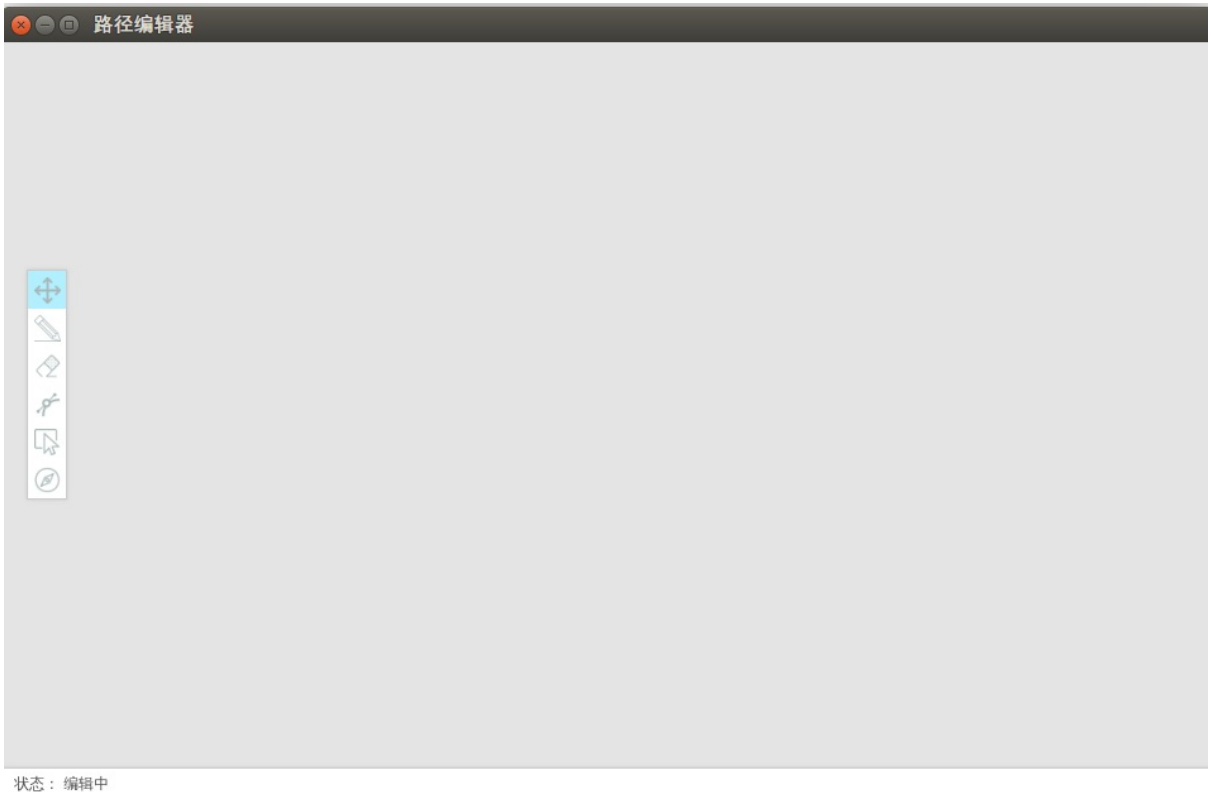
启动软件

安装完成后可以在Ubuntu的Dash菜单中找到名为Path Drawer的程序，点击启动即可。

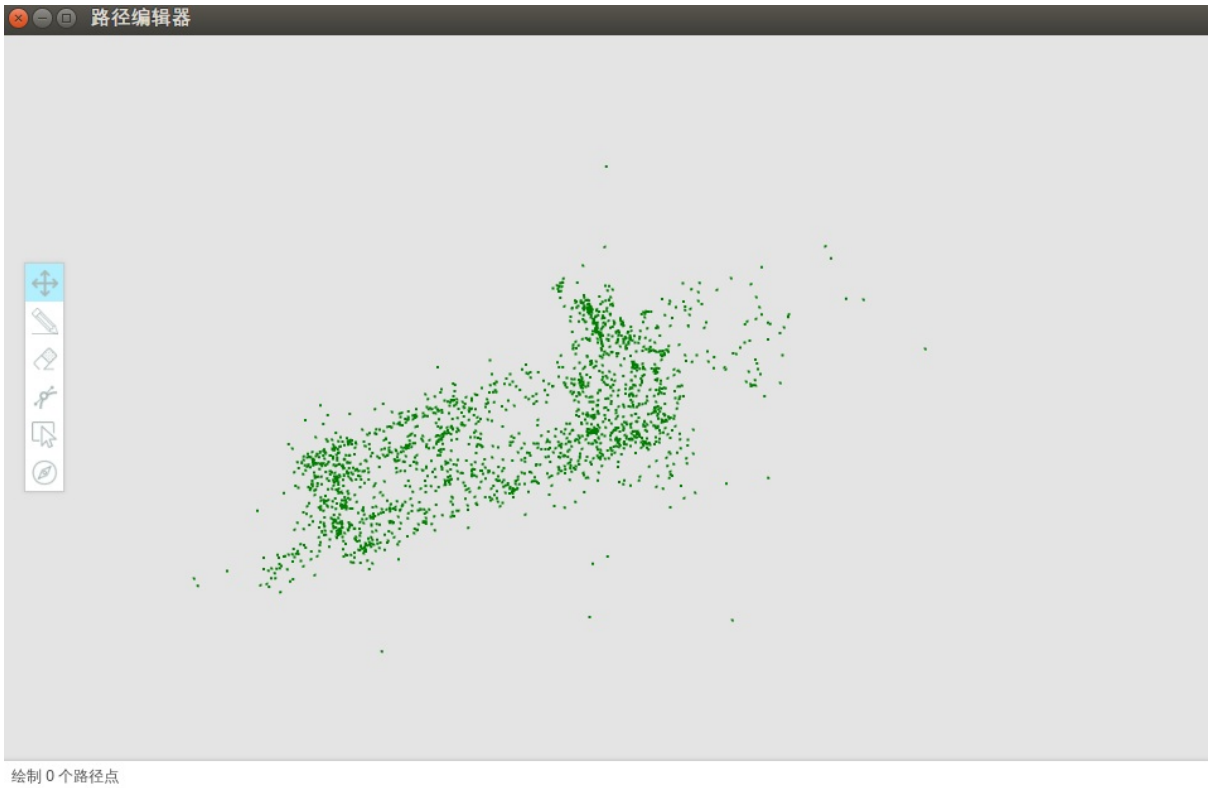


导入数据

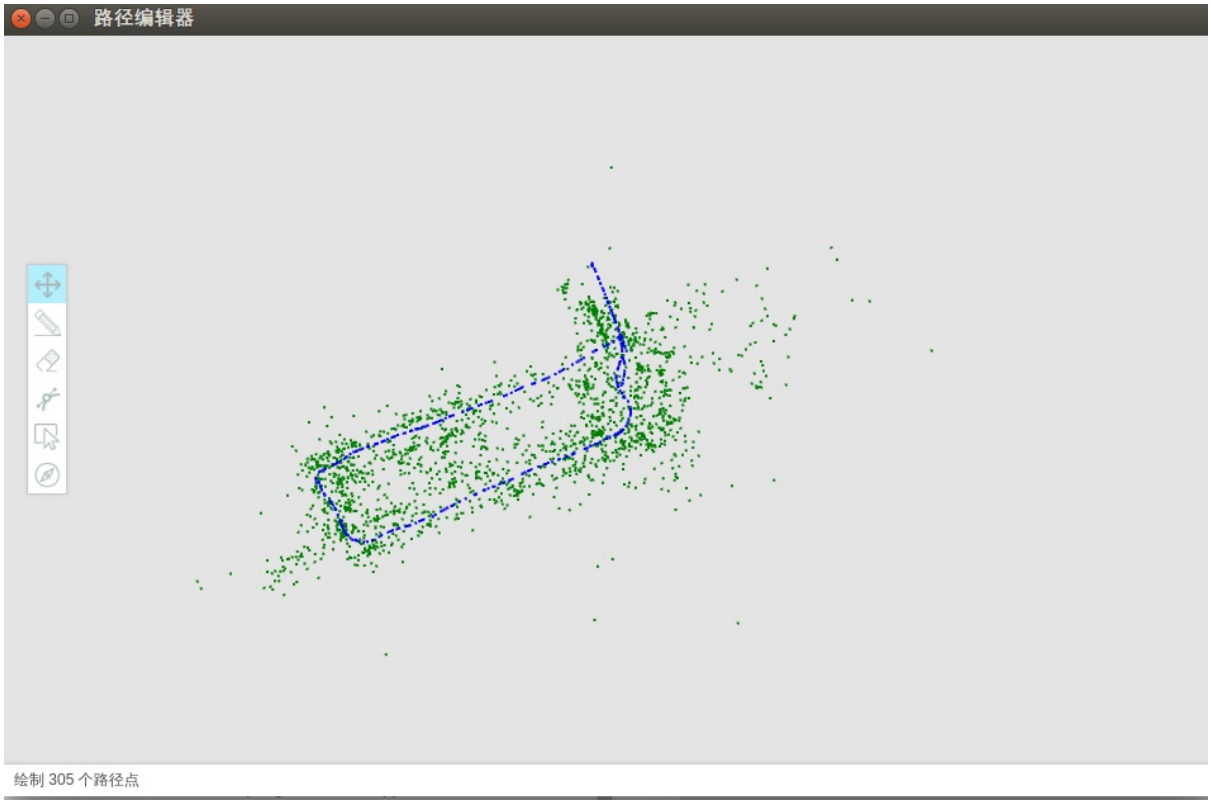
启动后的软件界面如图所示



在左上角的菜单中选择文件->导入地图数据。在弹出的文件选择对话框中选择 `/home/xiaoqiang/slamdb/mappoints.bson` 文件。 成功导入后就能在软件中看到地图点的数据。这是从上向下的俯视图。



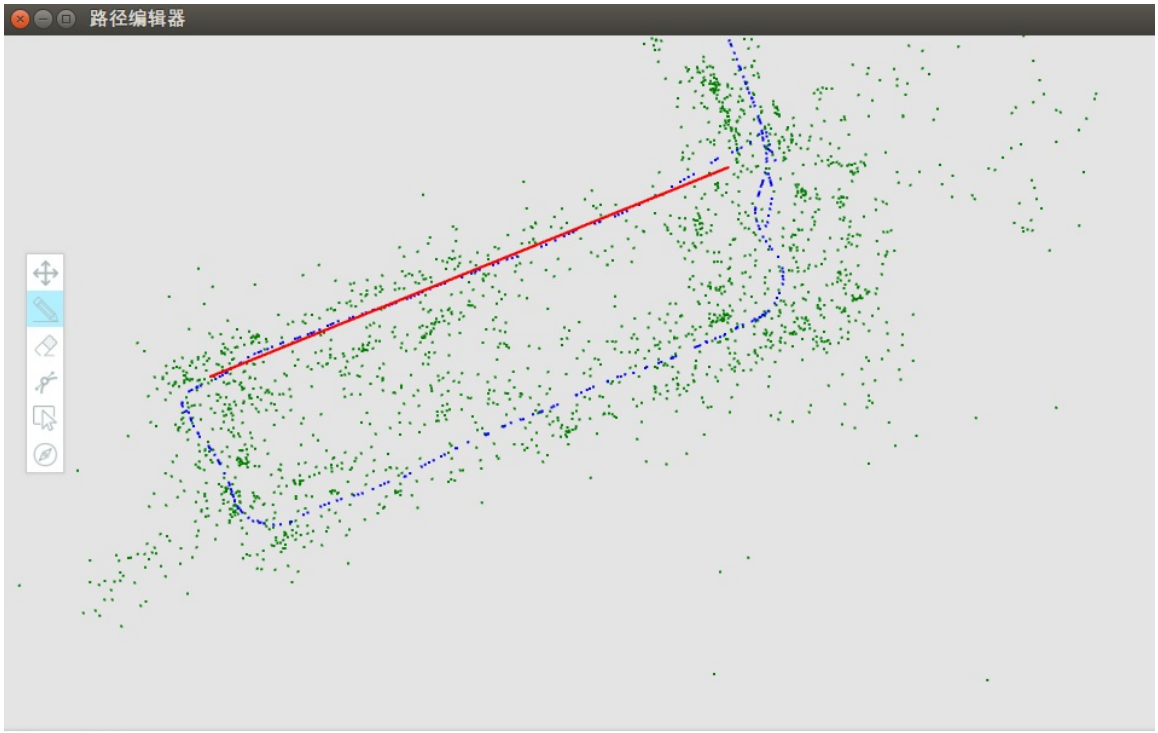
然后继续在左上角的菜单中选择文件->导入路径文件。在弹出的文件选择对话框中选择 `/home/xiaoqiang/slamdb/keyframes.bson` 文件。 成功导入后能在软件中看到之前小车行走的路径。



绘制导航路线

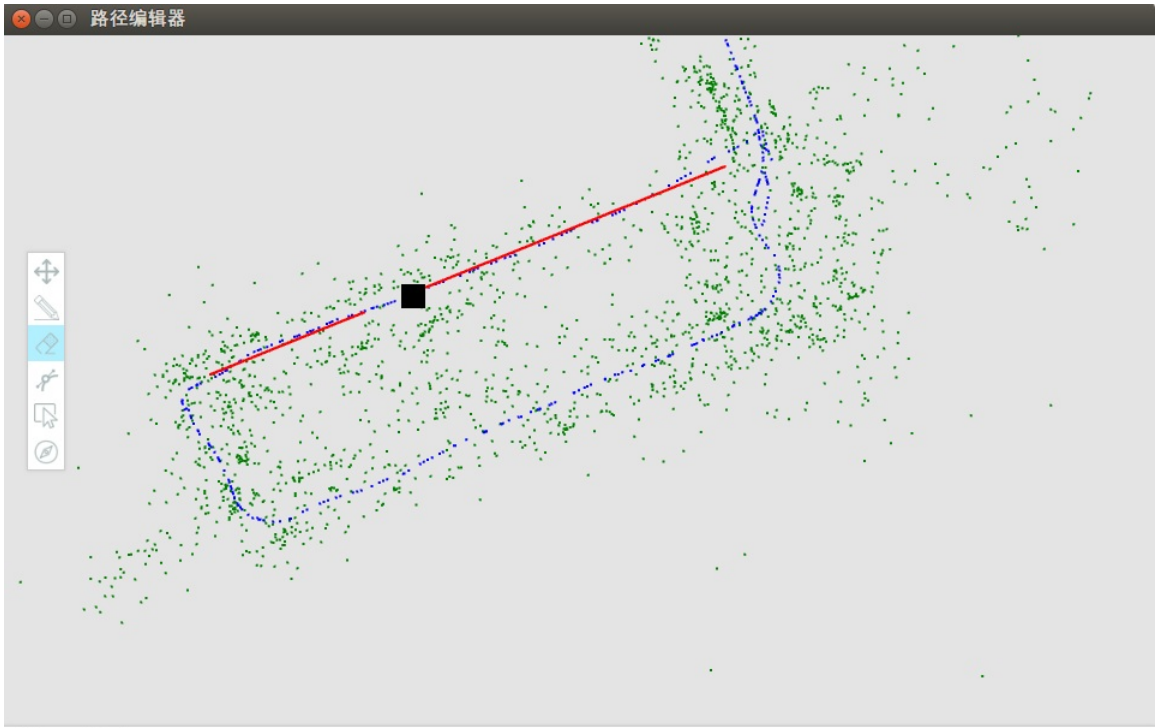
导航路线就是你想要小强行走的路径。当数据导出到小强后，小强就会按照你画的路径进行移动。下面介绍一下路径绘图工具的使用方法。

1. 基本操作 基本操作包括平移和缩放。如果鼠标左键拖动地图可以实现地图的平移。鼠标滚轴前后滚动可以实现地图的缩放，这在绘制路径的过程中非常的有用。对于对运动要求比较细致的地方可以放大后进行绘图。
2. 直线工具 点击左侧工具栏里的铅笔一样的图标。这就是直线工具。鼠标左键点击图上任意一点，然后移动鼠标就会出现一条红色直线。移动鼠标到想要的终止位置，再次点击鼠标左键，一条直线就绘制完成了。在点击一次左键之后，点击右键就可以取消此次绘图。



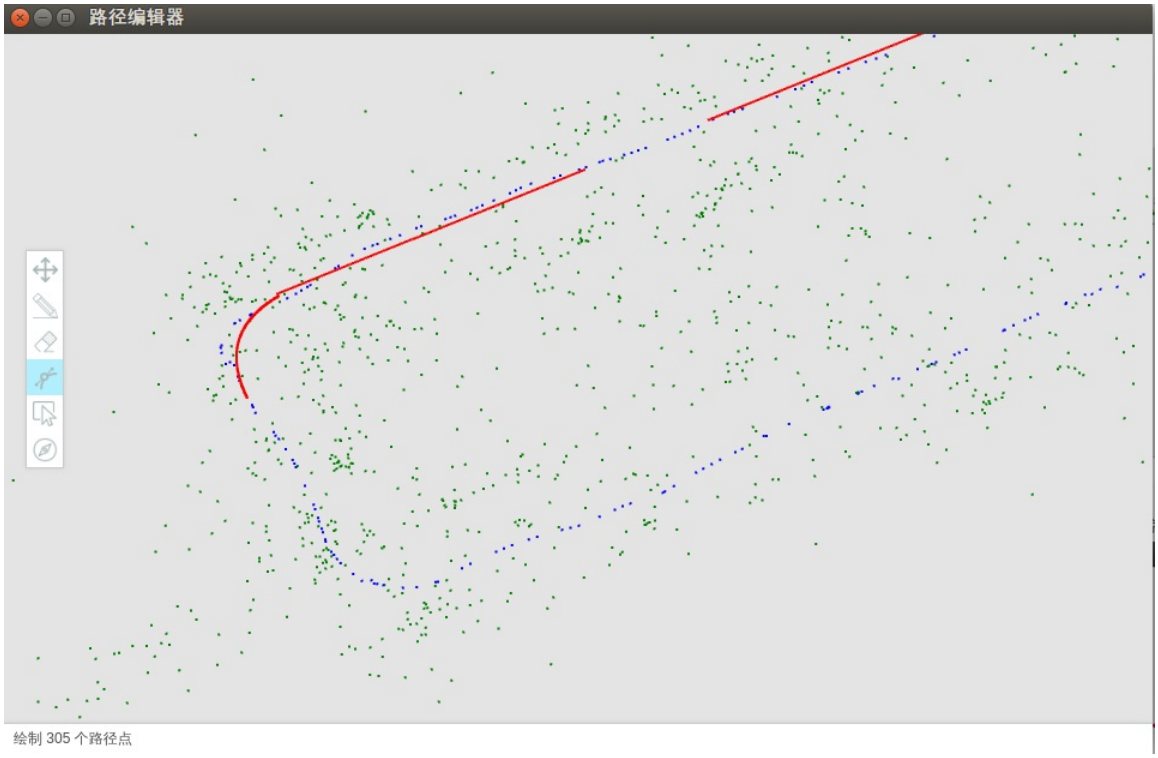
绘制 305 个路径点

3. 橡皮擦工具 点击左侧工具栏中的橡皮擦工具，然后按下鼠标左键进行拖动就可以擦除之前绘制的点。



绘制 305 个路径点

4. 曲线工具 点击左侧曲线工具，在曲线的起始点点击鼠标左键，然后在曲线的中间的再次点击一次鼠标左键，最后在曲线的结束点点击鼠标左键。这样一条曲线就绘制完成了。

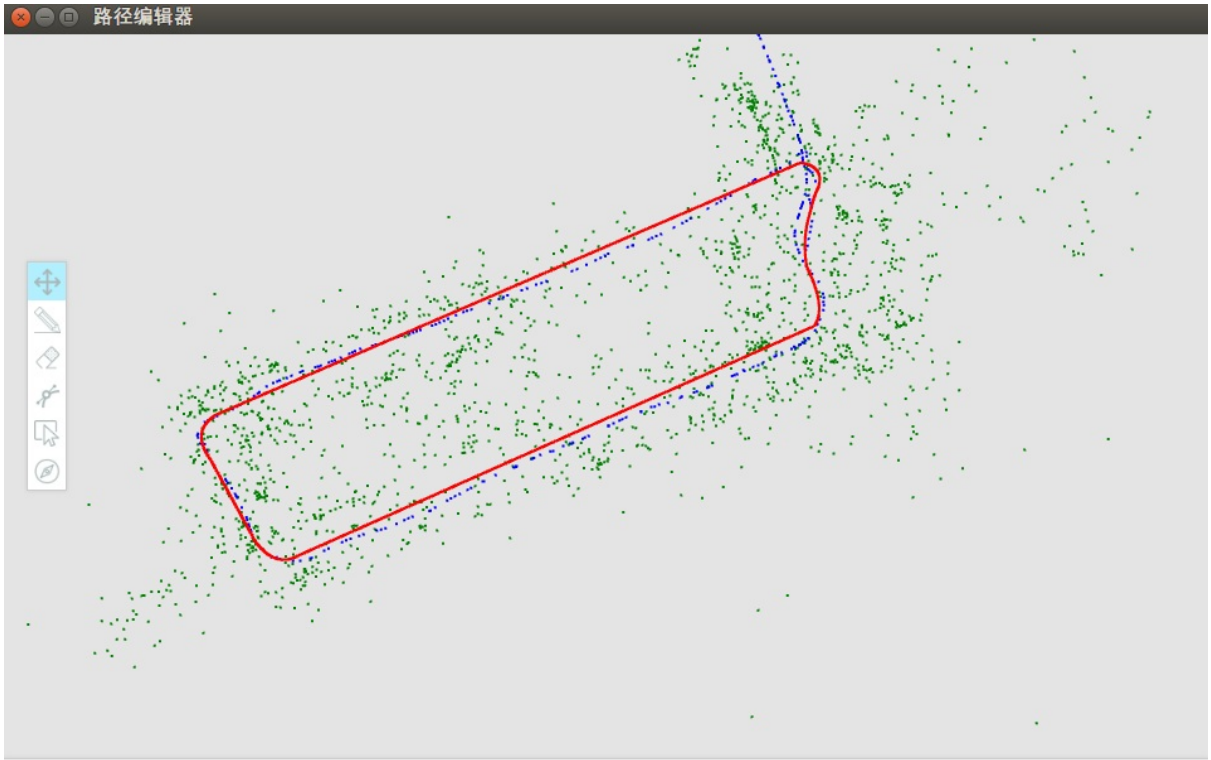


5. 删除工具 如果想要大范围的删除之前绘制的点，那么就可以利用这个删除工具。点击左侧的删除工具然后鼠标左键点击删除的起始点，可以看到在鼠标的移动过程中有一个矩形一直在跟随。再次点击鼠标左键就可以删除矩形选中的范围。右键可以取消选择。

利用这几个工具就可以绘制出小强的导航路径了。注意要尽量沿着原有的轨迹进行来画线，这样可以保证在运动过程中路线是畅通的。从绿色的地图点可以大致看出地形，根据这些信息画出运动所范围允许的点。

设置导航关键点

对于比较复杂的图形可能运动的方式有很多种。比如一个8字形路径，小强可能先绕其中的一个圆运动，然后再绕另一个圆运动，也可以两个圆交叉的运动。所以很有必要指明小强运动的具体方式。 下面以一个圆形轨迹为例。在圆形轨迹中，小强可以顺时针运动，同时也可能是逆时针运动。



导航路径文件载入成功: /home/randoms/Desktop/nav1.csv

点击鼠标右键可以删除最近添加的一个导航点。同样也可以利用橡皮擦和删除工具来删除导航点。小强会按照关键点标记的顺序进行运动。

导出数据

1. 导出导航路径文件 当导航路径绘制完成之后，在左上角的菜单中点击文件->导出导航路径文件，选择文件保存的位置即可。在导出文件后还可以从菜单中导入，进行二次编辑。
2. 导出导航关键点文件 当导航关键点绘制完成之后，在左上角的菜单中点击文件->保存导航点，选择文件的保存位置即可。 在导出文件后同样也可以再次从菜单中导入，进行二次编辑。注意：只能在导航路径文件导入成功之后才能导入导航关键点。

导出的数据放入小强的对应文件夹内就可以开始视觉导航了。

Enjoy it!

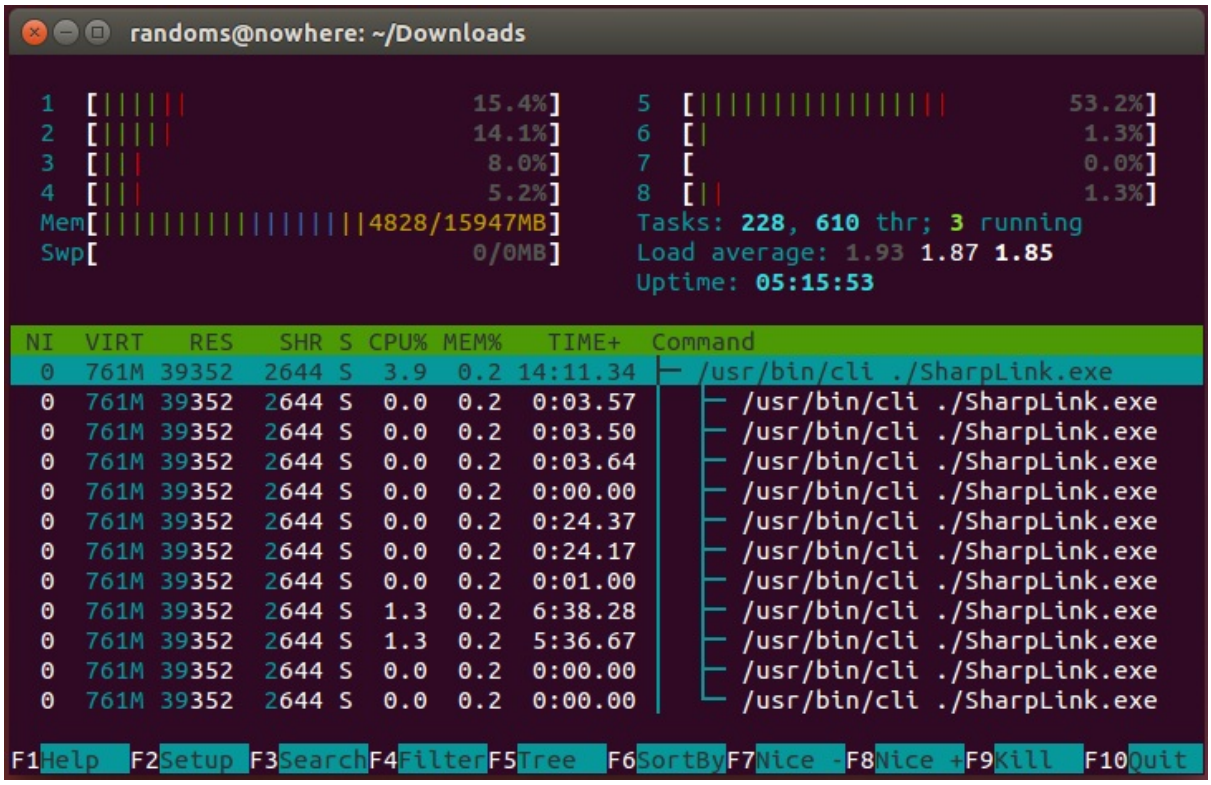
- 小强的远程协助功能

小强的远程协助功能

为了为您提供更好的服务，在2016年6月之后发售的小强都默认安装了远程协助软件。通过这个软件我们的技术人员能够直接连接到您的小强，为您解决技术问题。当然您也可以通过远程协助去更加方便的远程遥控自己的小强。下面介绍一下小强的远程协助功能的具体使用方法。

确认远程协助是否已经启动

在终端执行 `htop`



如果能够看到叫做 `SharpLink` 的线程正在执行说明远程协助已经开始执行了。 还可以通过更简单的方法确认程序的状态，执行一下指令

```
sudo service sharplink status
```

如果终端显示

```
xiaoqiang@xiaoqiang-desktop: ~
xiaoqiang@xiaoqiang-desktop:~$ sudo service sharplink status
[sudo] password for xiaoqiang:
● sharplink.service - Start sharplink service at startup
   Loaded: loaded (/lib/systemd/system/sharplink.service; enabled; vendor preset
   Active: active (running) since Tue 2018-09-25 15:12:37 CST; 22s ago
 Main PID: 2388 (cli)
   CGroup: /system.slice/sharplink.service
           └─2388 /usr/bin/cli /home/xiaoqiang/Documents/ros/devel/lib/sharplink

Sep 25 15:12:37 xiaoqiang-desktop systemd[1]: Started Start sharplink service at
Sep 25 15:12:37 xiaoqiang-desktop SharpLink.exe[2388]: ID: 4D87B61F3A66826911045
Sep 25 15:12:37 xiaoqiang-desktop SharpLink.exe[2388]: Server listening on 36063
Sep 25 15:12:53 xiaoqiang-desktop SharpLink.exe[2388]: From Server 36063:tox is
xiaoqiang@xiaoqiang-desktop:~$
```

则说明程序已经正常执行了。

远程连接小强

查看自己的ID

`每台小强都有一个自己的ID，请不要轻易的告诉别人。因为如果你没有更改默认密码，别人又知道你的ID的话，那么他就可以轻易的对你的小强进行操作。` 查看ID的方法也非常简单，在终端执行

```
sudo service sharplink restart
bwgetid
```

此时终端的输出为

```
D70101972AB9B7E674290A25485B3752EDA236F65EF2C4AC7E738390DA61903565E68B8C431B
```

这个很长的字符串就是您的ID。 如果您想要我们提供远程协助，只需要把这个ID告诉我们的工作人员就可以了。

远程连接自己的小强 在记住自己的ID之后，只要您的小强连接上互联网，您就可以在任意地方随时的控制它，不会受到路由器，局域网的限制。

首先在您的电脑上安装SharpLink，这个软件是跨平台的，无论是Linux还是Windows都可以安装。安装的具体方法在项目的介绍里面。

安装完成之后，在终端执行

```
./SharpLink.exe 9999 你的ID 127.0.0.1 22
```

这个指令会把本地9999端口和小强的22端口映射起来。只要连接本地的9999端口你就可以和小强的22端口相连了。当然你也可以把9999换成自己喜欢的端口。

此时在本地电脑上执行

```
ssh -p 9990 xiaoqiang@127.0.0.1
```

等待连接完成就可以控制小强了。

打开和关闭远程协助

如果你想要关闭远程协助也是非常简单的。在终端删除对应的服务文件就可以了。

```
sudo systemctl disable sharplink
```

如果你想重新打开远程协助服务

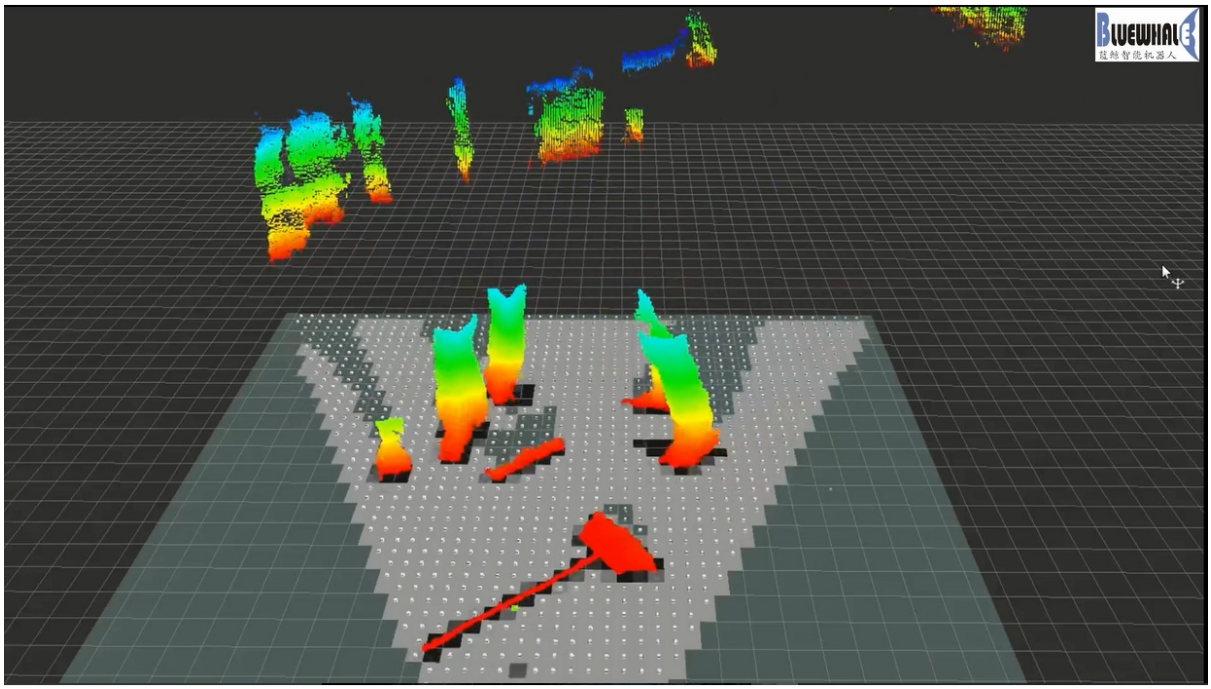
```
sudo systemctl enable sharplink
```

Enjoy it!

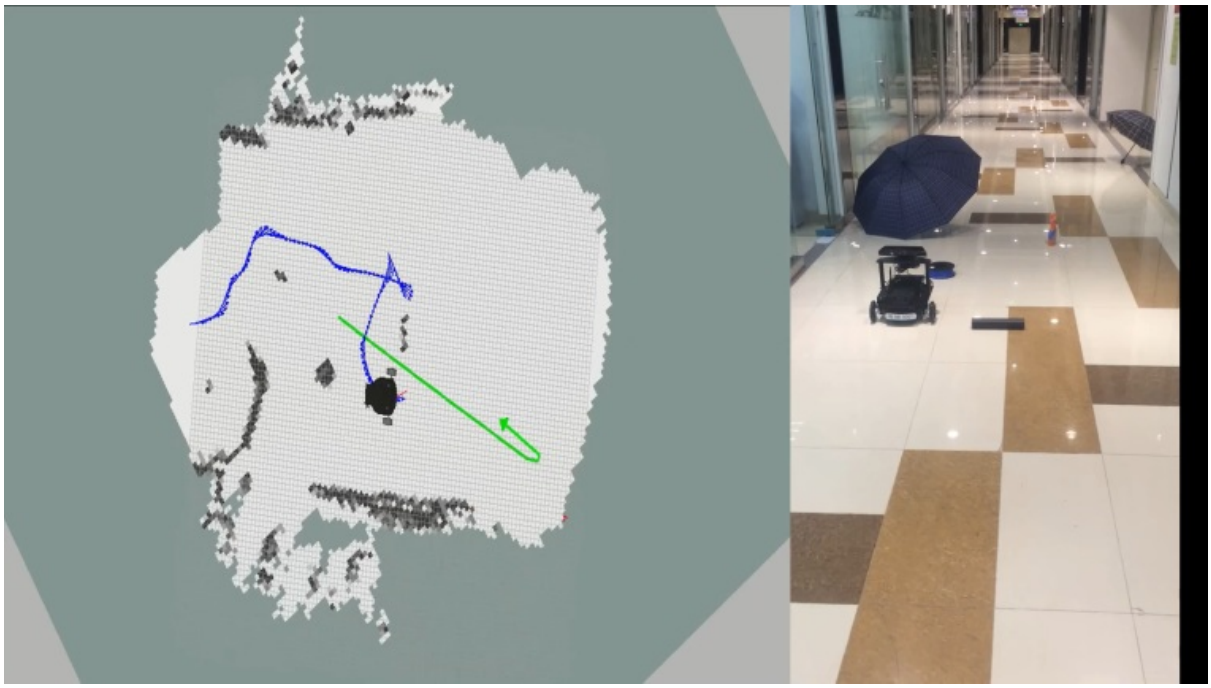
- 小强ROS机器人障碍物识别演示视频

小强ROS机器人障碍物识别演示视频

下图是视频截图，完整演示视频请点击这里,平台测试代码请关注蓝鲸开源仓库



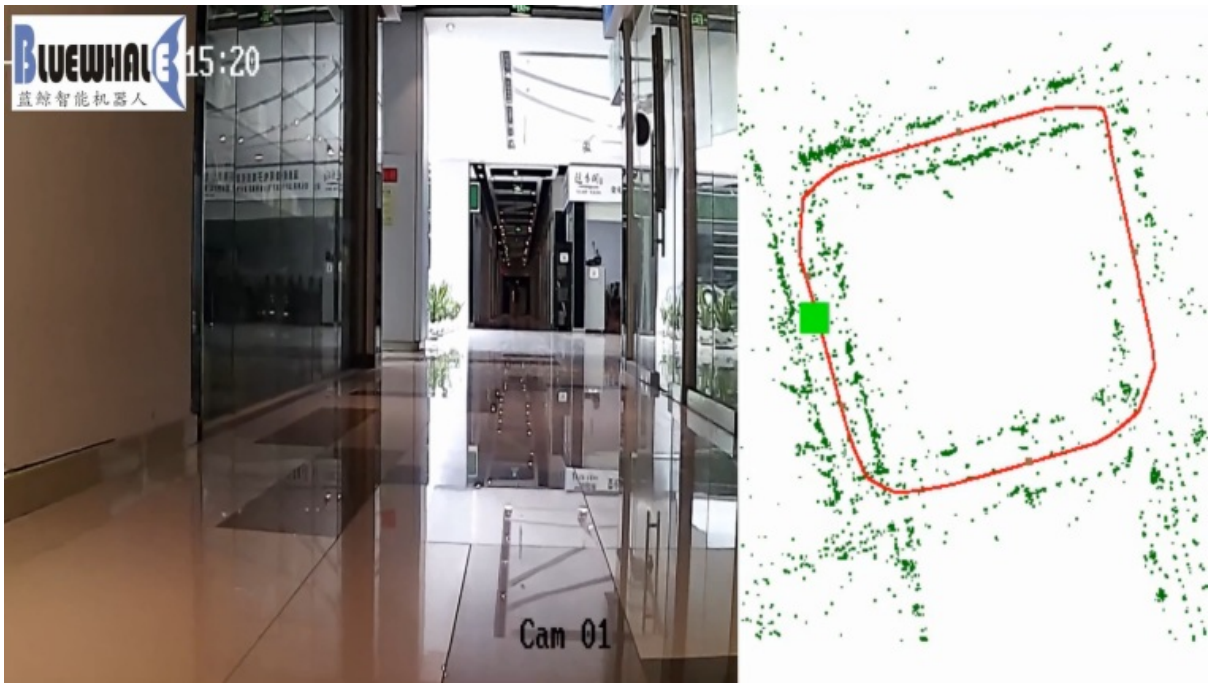
附上第二个小测试视频 自动驾驶测试视频



- 视觉导航在履带车中的运用

视觉导航在履带车中的运用

测试视频中的履带车采用了小强开发平台中的技术，点击观看完整视频



- google激光雷达slam算法Cartographer的安装及bag包demo测试
  - 操作步骤：

google激光雷达slam算法Cartographer的安装及bag包demo测试

Cartographer 是google于2016年９月开源的一套激光雷达slam算法,精度和效果在业界处于领先水准。点击查看最终效果。

00:00 / 00:00

操作步骤：

１．安装依赖包

```
# Install the required libraries that are available as debs.
sudo apt-get update
sudo apt-get install -y \
    cmake \
    g++ \
    git \
    google-mock \
    libboost-all-dev \
    libcairo2-dev \
    libeigen3-dev \
    libgflags-dev \
    libgoogle-glog-dev \
    liblua5.2-dev \
    libprotobuf-dev \
    libsuitesparse-dev \
    libwebp-dev \
    ninja-build \
    protobuf-compiler \
    python-sphinx
```

```
python-sphinx   可以换成 sphinx-common
```

2．安装ceres solver

```
cd  ~/Documents
```

```
git clone https://github.com.cnpmjs.org/ceres-solver/ceres-solver.git
cd ceres-solver
git checkout 1.14.x
mkdir build
cd build
cmake ..  #git hook时间比较长，耐心等待
make -j
sudo make install
```

3．安装prtobuf 3.0

```
cd  ~/Documents
git clone https://github.com.cnpmjs.org/google/protobuf.git
cd protobuf
git checkout v3.6.1
mkdir build
cd build
cmake  \
  -DCMAKE_POSITION_INDEPENDENT_CODE=ON \
  -DCMAKE_BUILD_TYPE=Release \
  -Dprotobuf_BUILD_TESTS=OFF \
  ../cmake
make -j 2
sudo make install
```

4．安装cartographer

```
cd  ~/Documents
git clone https://gitee.com/BlueWhaleRobot/cartographer.git
cd cartographer
mkdir build
cd build
cmake ..
make -j
sudo make install
```

5．安装cartographer_ros

```
cd ~/Documents/ros/src   #请修改路径到自己的ROS catkin工作空间
git clone https://gitee.com/BlueWhaleRobot/cartographer_ros.git
cd ..
catkin_make
```

6．安装已完成，开始下载测试用的bag文件

点击下述链接下载文件，保存到桌面 下载链接

7．启动demo演示，正常可以看到rviz启动并开始建图

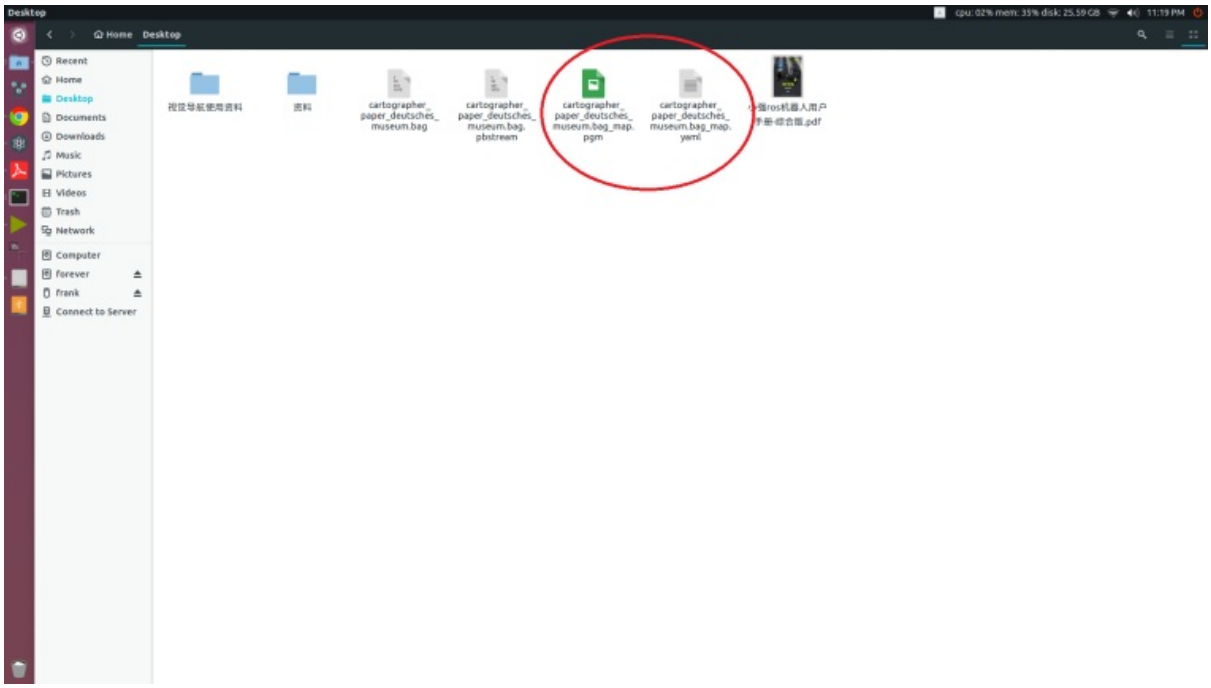
根据个人平台计算能力不同，本demo完整运行时间一般为半个小时到1个小时之间

```
roslaunch cartographer_ros offline_backpack_2d.launch  bag_filenames:=${HOME}/Desktop/cart
ographer_paper_deutsches_museum.bag
```

8．提取建立的地图，结束测试

上面步骤7会生成一个pbstream文件，用cartographer_ assets_writer可以转换成栅格地图

```
roslaunch cartographer_ros assets_writer_ros_map.launch bag_filenames:=${HOME}/Desktop/car
tographer_paper_deutsches_museum.bag pose_graph_filename:=${HOME}/Desktop/cartographer_pap
er_deutsches_museum.bag.pbstream
```



现在在home目录下的Desktop文件夹内会生成建立的地图文件，这两个文件（pgm 和 yaml）在ros中的map_server中可以加载使用

- 原装和国产ps3手柄ros驱动程序
  - 安装步骤
  - 快速使用方法
  - 购买带手柄的小强用户，还可以根据这篇教程使用手柄

原装和国产ps3手柄ros驱动程序

小强主页

ps3手柄的ros驱动程序为joystick_drivers包中的ps3joy,这份驱动对索尼原装手柄支持较好，但是对国产ps3手柄的支持存在问题。我们在ps3joy的基础上进行了修改，增加了一个ps3joyfake_node.py脚本作为国产手柄的驱动，包源代码地址在这里:。下文以小强为例，演示这个包的安装步骤和简略的使用方法

安装步骤

ssh进入小强ros工作空间，下载源码后编译，完成安装 小强镜像中默认已经安装此软件包，可以跳过这一步骤。

```
ssh xiaoqiang@192.168.xxx.xxx
cd Documents/ros/src/
git clone https://gitee.com/BlueWhaleRobot/joystick_drivers.git
cd ..
catkin_make
# 如果提示下列错误
error spnav.h no such file
# 先安装下面这个包，然后重新执行catkin_make
sudo apt-get install libspnav-dev
```

快速使用方法

`ps3joyfake_node.py` 启动后，它会将蓝牙接收的手柄按键数据转换成标准的joy msg，同时以 `/joy` 为话题在ros中发布，即 `ps3joyfake_node.py` 一个文件相当于 `ps3joy.py + joy_node` 两个文件，在实际使用中不必再开启 `joy_node` 节点。

1.将手柄与usb蓝牙适配器进行绑定，只需绑定一次，下次直接跳过这个步骤

`如果是手柄是和小强同时购买，那么我们已经提前为您配对，可以跳过此步骤。` 将手柄通过usb数据线接入主机，usb蓝牙适配器也插上主机

```
sudo bash
rosrun ps3joy sixpair
```

此时会得到类似下图的输出,current和setting的mac地址一样说明绑定成功

```
Current Bluetooth master: 00:22:b0:d0:5a:09
```

```
Setting master bd_addr to 00:22:b0:d0:5a:09

# 如果出现下述错误
Current Bluetooth master: 00:1b:dc:00:07:3c
Unable to retrieve local bd_addr from `hcitool dev`.
Please enable Bluetooth or specify an address manually.
# 先运行
hciconfig hci0 reset
# 如果运行
hciconfig hci0 reset
# 出现错误
# Can't init device hci0: Operation not possible due to

# 运行
rfkill unblock all
# 然后运行
hciconfig hci0 reset
# 重新运行
rosrun ps3joy sixpair
```

绑定设置完成,断开手柄与主机的usb连接

ctrl+D　退出root模式

2.将手柄与usb蓝牙适配器配对

```
#确保蓝牙接收器已经插入主机usb口
sudo bash
rosrun ps3joy ps3joyfake_node.py
```

正常会出现下面的提示

```
root@xiaoqiang-desktop:~# rosrun ps3joy ps3joyfake_node.py
No inactivity timeout was set. (Run with --help for details.)
Waiting for connection. Disconnect your PS3 joystick from USB and press the pairing button
.
#如果提示下列错误
ImportError: No module named bluethooth
#请先安装下列包,然后重新运行
sudo apt-get install libbluetooth-dev
sudo pip install PyBluez
```

按一下下图中的手柄配对键

配对成功的话，上面的窗口会输出类似下面的结果

```
root@xiaoqiang-desktop:~# rosrun ps3joy ps3joyfake_node.py
No inactivity timeout was set. (Run with --help for details.)
Waiting for connection. Disconnect your PS3 joystick from USB and press the pairing button
.
Connection activated
```

配对完成

3.查看手柄输出

```
新开一个窗口，打印按键输出
rostopic echo /joy
```

正常会出现下面的类似结果

```
header:
  seq: 297
  stamp:
    secs: 1488877867
    nsecs: 535818099
  frame_id: ''
axes: [0.0, 0.0, 0.0, 0.0, -1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.21316899359226227, 0.0]
buttons: [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

从上面可以看到第 4 个button被按下了

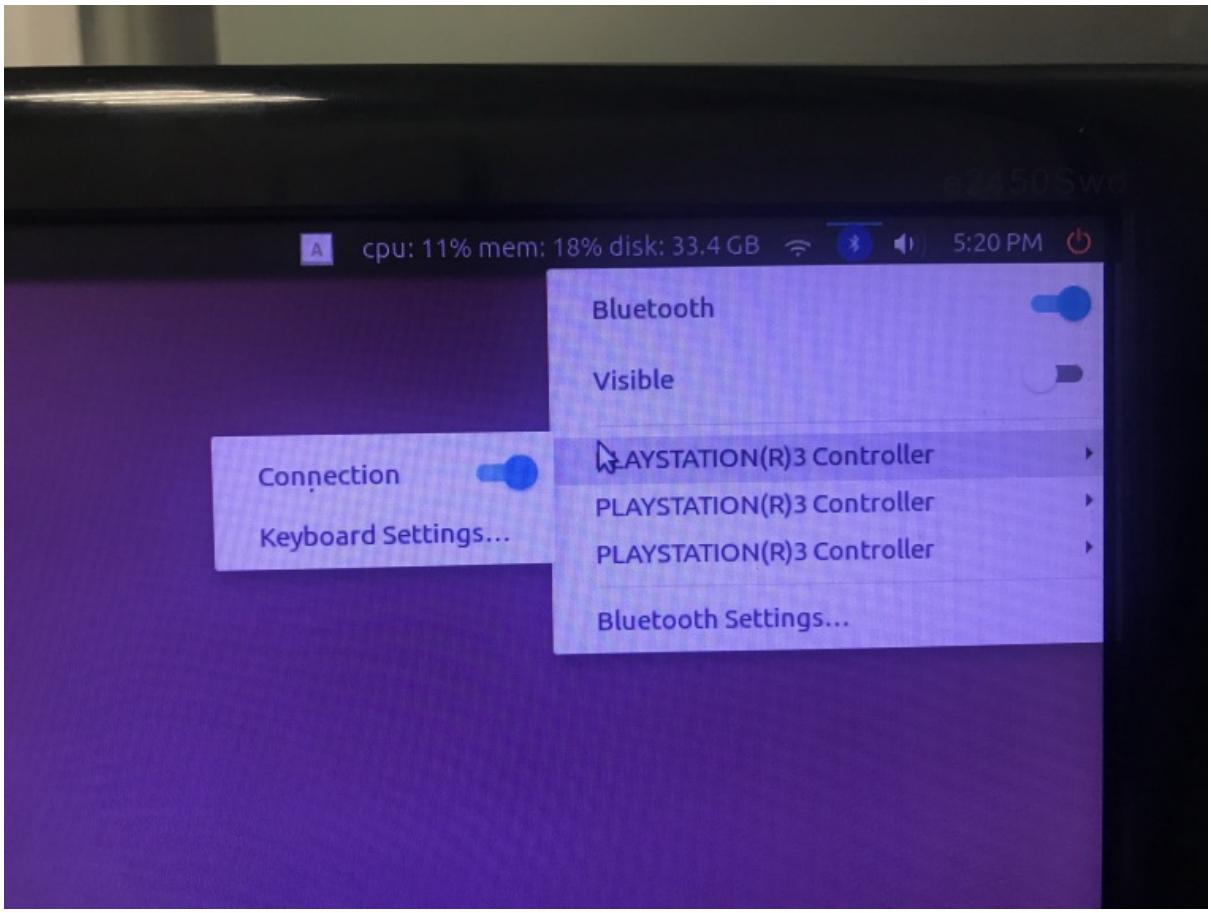
4.启动相关的joy msg处理节点

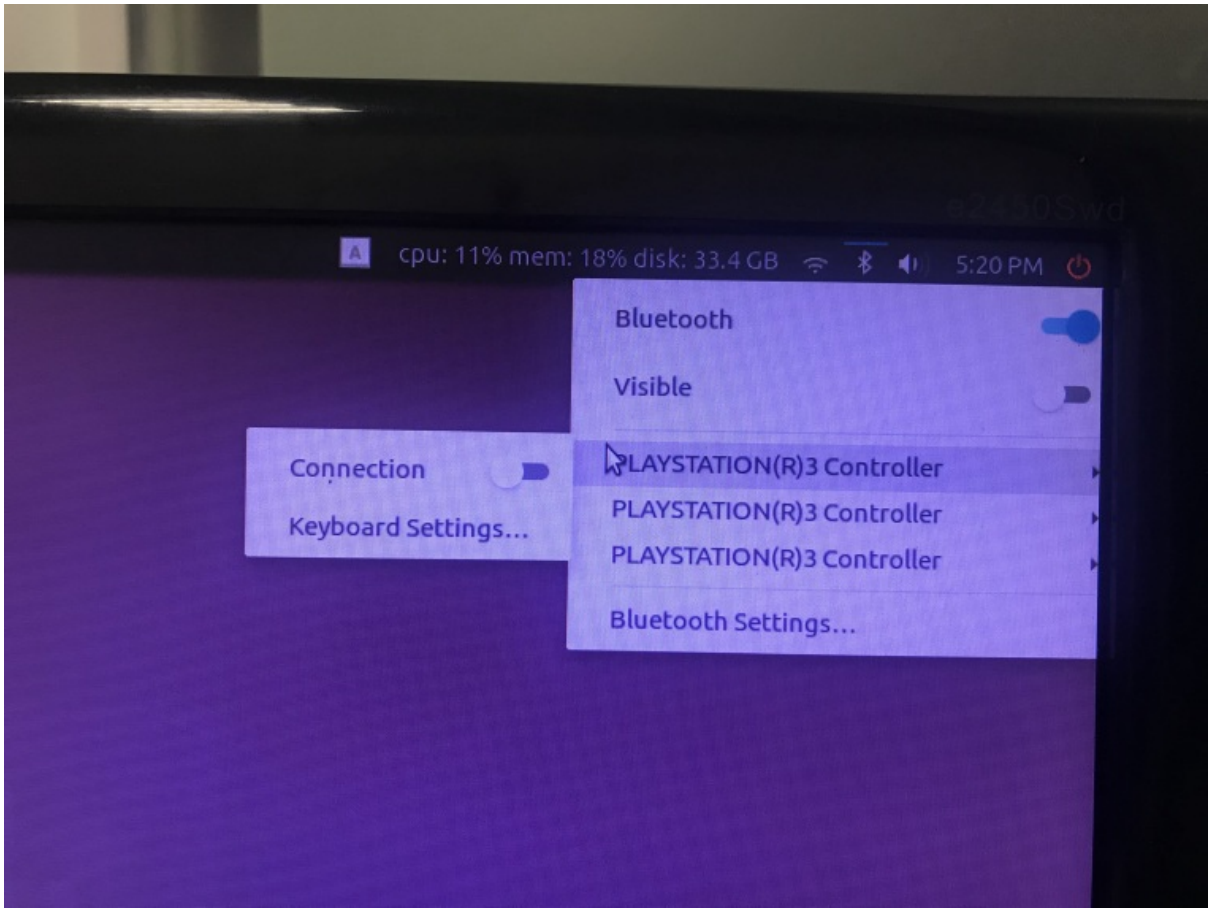
注意不要再启动joy_node

```
#以小强为例,启动下列launch文件后就可以遥控小强移动了
roslaunch turtlebot_teleop ps3fakexiaoqiang_teleop.launch
```

购买带手柄的小强用户，还可以根据这篇教程使用手柄

进行步骤1绑定时，如果提示找不到手柄设备，无法绑定，可以执行下面两个步骤以解决问题： 1.点击系统桌面右上角的蓝牙图标，将连接着的蓝牙手柄服务关闭





2.重新插拔手柄，然后重新运行绑定指令

小强主页 返回目录

- 升级软件包以支持小强图传遥控app

升级软件包以支持小强图传遥控app

小强主页

`注：本教程适用于2017年3月份之前收到小强的用户，2017年3月份之后的用户请跳过本教程，直接参考这两篇帖子的使用方法` — 小强手机遥控app安卓版、小强图传遥控windows客户端

小强手机遥控app需要小强开机自动启动一些配套服务后才能使用，下文将演示升级小强主机上的软件包地具体步骤，完成这些步骤后重启小强主机即可开始使用手机端app。

1.ssh登陆小强主机

```
ssh xiaoqiang@192.168.xxx.xxx -X
```

2.进入小强主机ros工作空间，升级替换两个软件包

```
cd Documents/ros/src/
rm -rf system_monitor
rm -rf startup
git clone https://github.com/BlueWhaleRobot/system_monitor.git
cd system_monitor
git checkout kinetic_service_bot
cd ..
git clone https://github.com/BlueWhaleRobot/startup.git
cd startup
git checkout service_bot
cd ../..
catkin_make
```

3.更新小强开机启动项

```
rosrun robot_upstart uninstall startup
rosrun robot_upstart install startup/launch/startup.launch
sudo systemctl daemon-reload && sudo systemctl start startup
```

小强主页 返回目录

- 小强自检程序发布
    - 安装方法
    - 使用方法

小强自检程序发布

为了方便各位用户使用小强，我们开发了小强的自检程序。当小强无法正常工作时可以通过这个程序获取到有用的信息，方便调试和解决问题。

安装方法

使用 `xq_os` 的用户可以通过 `bwupdate` 指令安装。使用旧版本系统的用户需要先安装 `xiaoqiang-cmds` 软件包。安装方法可以参照这里.

使用方法

执行

```
bwcheck
```

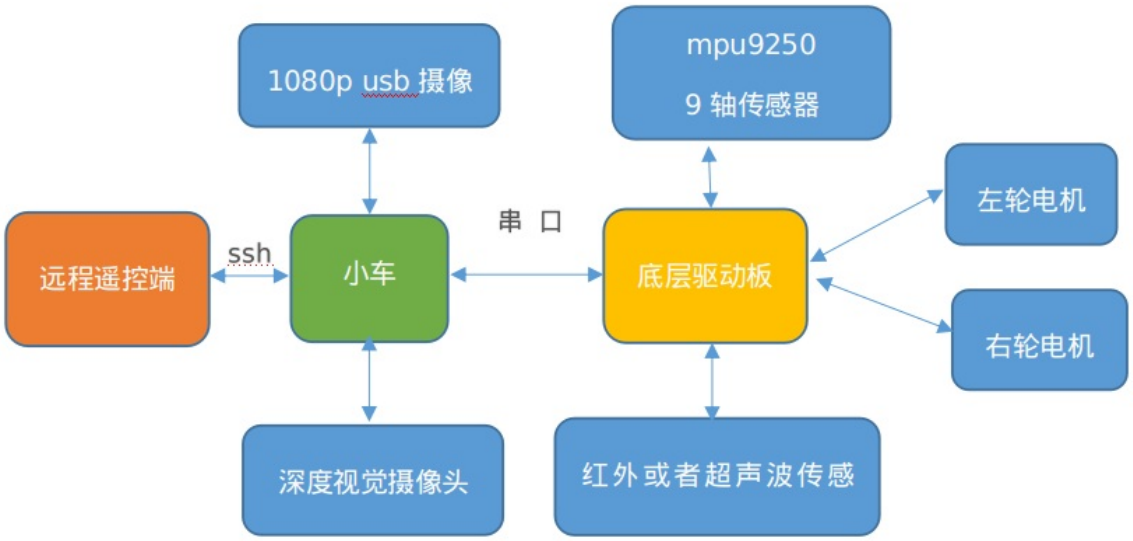
输出结果如下

```
xiaoqiang@xiaoqiang-desktop:~$ bwcheck

╔═══╗ ╔═╗   ╔═╗ ╔═══╗╔═╗   ╔═╗╔═╗ ╔═╗ ╔═══╗ ╔═╗   ╔═══╗
╚═╗ ╚═╗╚═╗   ╔═╗ ╚═╗ ╚═╗   ╔═╗╚═╗ ╚═╗ ╚═╗ ╚═╗ ╚═╗   ╚═╗ ╚═╗
╔═══╗ ╚═╗   ╔═╗ ╚═╗ ╚═╗   ╔═╗╚═╗ ╔═╗ ╚═╗ ╚═╗ ╚═╗   ╔═╗ ╚═╗
╚═╗ ╚═╗ ╚═╗ ╔═╗ ╚═╗ ╚═╗   ╚═╗╚═╗ ╚═╗ ╚═╗ ╚═╗ ╚═╗   ╚═╗ ╚═╗
╔═══╗ ╚═══╗ ╔═══╗ ╚═══╗   ╔═╗╚═╗ ╔═╗ ╚═╗ ╚═╗ ╚═╗   ╔═══╗ ╚═╗
╚═══╝ ╚═══╝ ╚═══╝ ╚═══╝   ╚═╝ ╚═╝ ╚═╝ ╚═╝ ╚═╝ ╚═╝   ╚═══╝ ╚═╝


系统版本: Ubuntu 16.04 XQOS 2
检查软件系统中...
检查环境变量
ROS_MASTER_URI ... WARNING
警告: ROS_MASTER_URI最好应当是http://xiaoqiang-desktop:11311
请注意是否执行过 export ROS_MASTER_URI=http://xiaoqiang-desktop:11311

ROS_PACKAGE_PATH ... OK
addwa_local_planner ... OK
autoWifi ... OK
BwbotCenter ... OK
bwbotupdater ... OK
cam_auto_brightness ... OK
freenect_stack ... OK
iai_kinect2 ... OK
image_pipeline ... OK
joystick_drivers ... OK
libuvc_ros ... OK
nav_test ... OK
```

```
NLlinepatrol_planner ... OK
openslam_gmapping ... OK
orb_init ... OK
ORB_SLAM ... OK
ORB_SLAM2 ... OK
rgbd_launch ... OK
robot_arm ... OK
rplidar_ros ... OK
rqt_common_plugins ... OK
SharpLink ... OK
slam_gmapping ... OK
startup ... OK
stm32loader ... OK
system_monitor ... OK
turtlebot_apps ... OK
turtlebot_msgs ... OK
usb_cam ... OK
web_video_server ... OK
xiaoqiang_udrf ... OK
xqserial_server ... OK
audio_common ... OK
cartographer_ros ... OK
geometry ... OK
geometry2 ... OK
navigation_msgs ... OK
robot_upstart ... OK
turtlebot ... OK
turtlebot_apps ... OK
turtlebot_msgs ... OK
laser_filters ... OK
navigation ... OK
bwbot_check ... OK
检查hosts文件 ... OK
检查hostname ... OK
检查95-persistent-serial.rules文件 ... OK
检查startup服务程序 ... OK
检查硬件中...
检查底盘驱动程序 ... ERROR
错误: 底盘驱动程序没有启动

检查底盘串口设备 ... ERROR
错误: 未发现底盘串口设备

检查摄像头驱动程序 ... ERROR
错误: 摄像头驱动程序未启动

检查摄像头设备 ... ERROR
错误: 未发现摄像头设备

检查kinect ros驱动程序 ... ERROR
错误: kinect ros 驱动未启动

检查kinect设备 ... ERROR
错误: 未发现kinect设备

检查电池电压 ... ERROR
错误: 获取电池电压失败
```

根据错误提示就能找到小强出现异常的原因了。

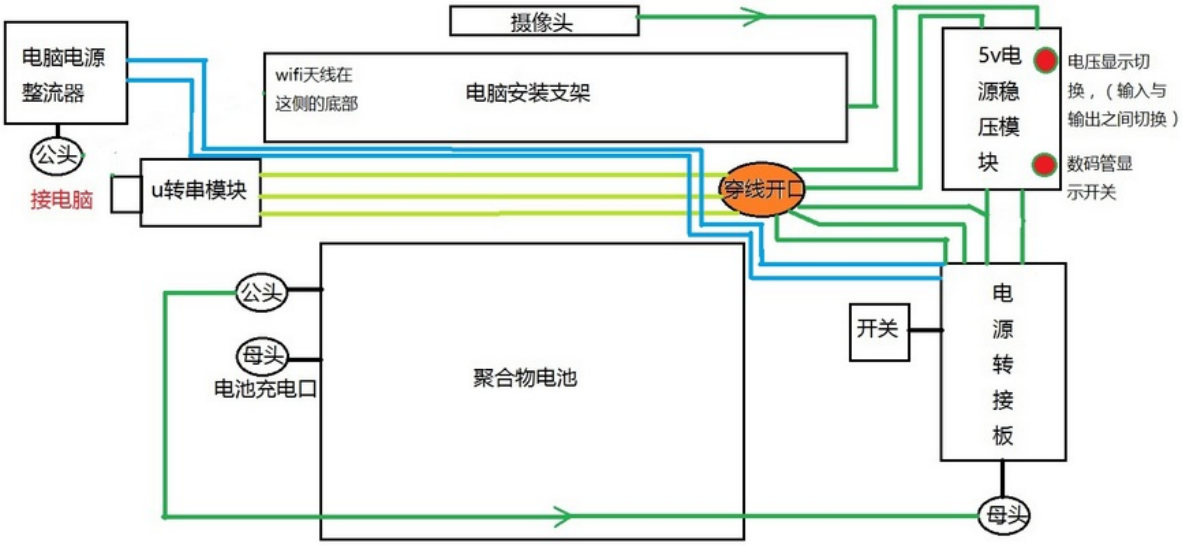
小车系统框架图 小车主机基本配置为intel I7 4560U、64gb固态硬盘、8gb内存。20AH聚合物电池，整机续航7小时以上。小车移动速度最大0.8m/s，电机自带编码器配有速度控制，自适应多种承重负载和地面路况。



- 电气布线图

电气布线图

小强底盘上的电线头部都带有线标，各种模块pcb上也有引脚标识，接线时只需要将线标与引脚标识对应起来就可以了，对应关系是名字一样。 这是小强底盘上端面俯视图：



- 小强电脑与stm32底层串口通讯协议

小强电脑与stm32底层串口通讯协议

串口波特率为115200，8个数据位，1个停止位，无奇偶校验。

第一部分是上位机对底层下发的控制指令：

每个指令构成一个数据帧，可以同时下发多个数据帧，帧内容是：包头+长度+内容，包头为3个u8字符：205 235 215 长度是1个u8字符，长度不包括包头和长度字符自身，内容为命令+参数。

单字符命令，即一个字符完成的命令： 调试模式 ‘T’ 运行模式 ‘R’ 软件复位 ‘I’

```
例如，让stm32软件复位，下发的数据帧hex内容为：0xcd 0xeb 0xd7 0x01 0x49
```

双字符命令，即一个字符加一个参数X完成的命令： 前进 'f‘X , X数值范围为0~100
后退 'b’X , X数值范围为0~100 刹车 's'X , X数值任意 左转 'c'X , X数值范围为0~100 右转 'd'X , X数值范围为0~100

```
例如：
以80%的制动量制动小车：  0xcd 0xeb 0xd7 0x02  0x73 0x50
以16%的速度左转：        0xcd 0xeb 0xd7 0x02  0x63 0x10
以16%的速度右转：        0xcd 0xeb 0xd7 0x02  0x64 0x10
```

多字符带参数命令： 设置电机 't'XXXXxxxx 命令说明：完整命令格式tABCDabcd，总共9个字符。t为命令标识，表示设置电机命令 ABCD分别为1--4个电机的状态，每个字符有三种状态，F：向前，B：向后，S：刹车， 其他字符无效。abcd为四个电机的速度等级，范围0～100,即以字符的数值表示速度

```
例如：

让四个电机全部以0%的制动量刹车 tSSSS0000
对应帧内容为 0xcd 0xeb 0xd7 0x09 0x74 0x53 0x53 0x53 0x53 0x00 0x00 0x00 0x00

对于小强硬件，只有前两个电机有效，第一个电机对应右轮，第二个电机对应左轮
```

第二部分是底层给上位机上传的数据：

也是数据帧的形式上传，上传频率为50HZ。帧格式和上面的命令帧一样，只是把命令内容换成了数据。 数据内容是由23个4字节小端模式二进制表示的数字组成，数字之间用空格0x20分开。这23个数字可以构成如下结构体，建议使用内存拷贝的方式直接将数据内容拷贝给这个结构体，然后以结构体形式访问数据，这样可以避免浮点数转换问题。具体代码请参考我们git开源仓库中的上位机ros驱动。

```
typedef struct {
    int status;//小车状态，0表示未初始化，1表示正常，-1表示error
    float power;//电源电压【9 13】v
    float theta;//方位角，【0 360）°
    int encoder_ppr;//车轮1转对应的编码器个数
    int encoder_delta_r;//右轮编码器增量，  个为单位
```

```
    int encoder_delta_l;//左轮编码器增量,  个为单位
    int encoder_delta_car;//两车轮中心位移, 个为单位
    int omga_r;//右轮转速 个每秒
    int omga_l;//左轮转速 个每秒
    float distance1;//第一个超声模块距离值 单位cm
    float distance2;//第二个超声模块距离值 单位cm
    float distance3;//第三个超声模块距离值 单位cm
    float distance4;//第四个超声模块距离值 单位cm
    float IMU[9];//mpu9250 9轴数据
    unsigned int time_stamp;//时间戳
}UPLOAD_STATUS;
```