

Ros三轮小车操作指南

一、启动流程介绍

1.1 准备设备

Ros 三轮小车：一台

显示屏：一台

HDMI 接口线：一根

USB 线：一根

USB 鼠标：一个

USB 键盘：一个

1.2 连接原理图

显示屏的连接方法：HDMI 接口线接在工控机的 HDMI 接口，USB 线接在小车工控机上面的 HUB 上。若要控制机器人，直接在输入相关启动指令运行即可。



图！

1.3 机器人启动

1. 按下机器人启动面板上的电源开关。电源是小车上面的红色按钮，关闭的状态是凸起，打开的状态是按下。打开之后，小车上的电路板会有灯光点亮并且显示

屏会显示电量。

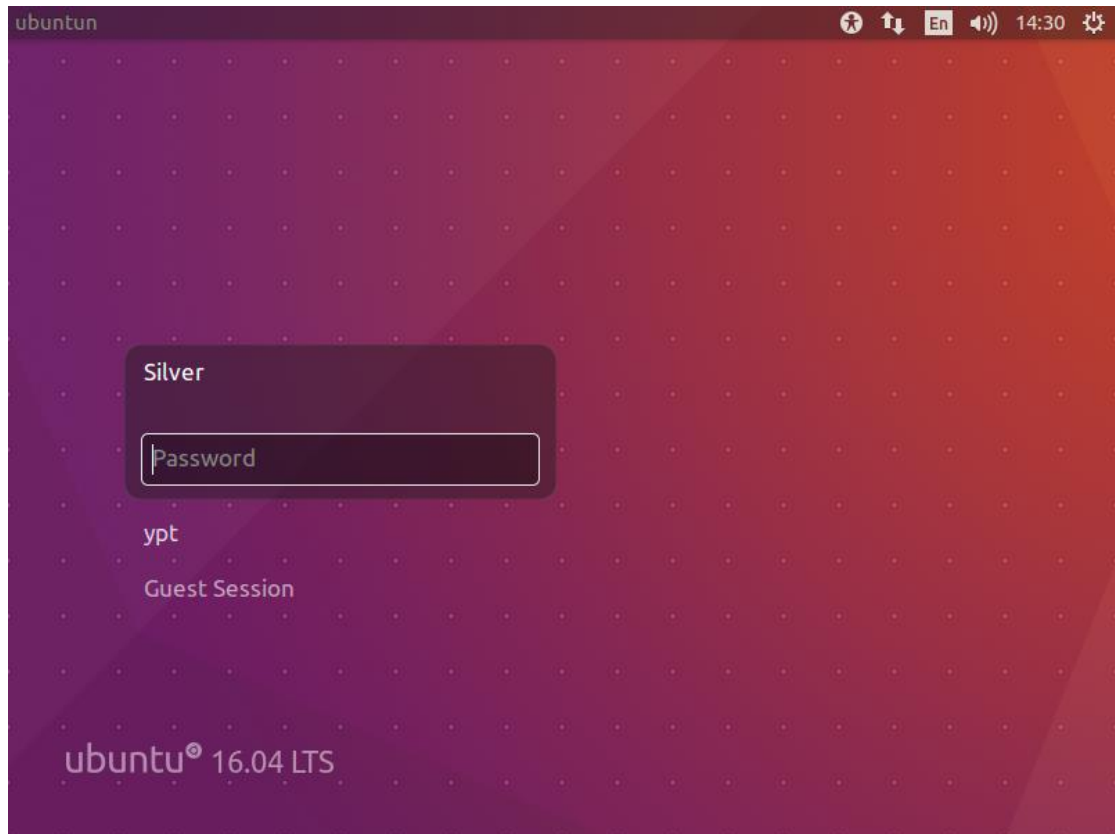
图！

2. 打开工控机，按下下图的白色按钮。



3. 等待机器人开机，显示屏出现如下画面。进入登陆页面，默认密码为：12345

6，输入密码后回车。

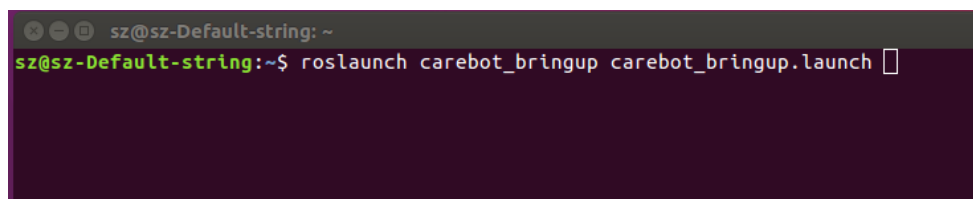


二、测试机器人

2.1 键盘控制

1. 机器人启动完毕后，按 Ctrl+Alt+T 或者点击 Terminal 图标，弹出终端窗口。
2. 在终端口输入启动命令

```
$ roslaunch carebot_bringup carebot_bringup.launch
```



3. 回车，出现如下画面。

```
/home/sz/omniWheelCareRobot/rosCode/src/carebot_bringup/launch/carebot_bringup.la
ACCEL_Z_MIN:-250.00
ACCEL_Z_MAX:250.00

MAGN_X_MIN:-600.00
MAGN_X_MAX:600.00
MAGN_Y_MIN:-600.00
MAGN_Y_MAX:600.00
MAGN_Z_MIN:-600.00
MAGN_Z_MAX:600.00

MAGN_USE_EXTENDED:false
magn_ellipsoid_center:[0.0000,0.0000,0.0000]
magn_ellipsoid_transform:[[0.0000000,0.0000000,0.0000000],[0.0000000,0.0000000,0
.0000000],[0.0000000,0.0000000,0.0000000]]

GYRO_AVERAGE_OFFSET_X:0.00
GYRO_AVERAGE_OFFSET_Y:0.00
GYRO_AVERAGE_OFFSET_Z:0.00

[INFO] [1527053811.322688]: Flushing first 200 IMU entries...
[INFO] [1527053815.423192]: Publishing IMU data...
[ INFO] [1527053815.443853411]: Initializing Imu sensor
[ INFO] [1527053815.485029365]: Imu sensor activated
```

4. 再次打开新的命令窗口，输入以下指令，执行 launch 命令，启动键盘控制节点

```
$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

5. 回车，出现如下画面。

```
/home/sz/omniWheelCareRobot/rosCode/src/carebot_bringup/launch/carebot_bringup.la
ACCEL_Z_MIN:-250.00
ACCEL_Z_MAX:250.00

MAGN_X_MIN:-600.00
MAGN_X_MAX:600.00
MAGN_Y_MIN:-600.00
MAGN_Y_MAX:600.00
MAGN_Z_MIN:-600.00
MAGN_Z_MAX:600.00

MAGN_USE_EXTENDED:false
magn_ellipsoid_center:[0.0000,0.0000,0.0000]
magn_ellipsoid_transform:[[0.0000000,0.0000000,0.0000000],[0.0000000,0.0000000,0
.0000000],[0.0000000,0.0000000,0.0000000]]

GYRO_AVERAGE_OFFSET_X:0.00
GYRO_AVERAGE_OFFSET_Y:0.00
GYRO_AVERAGE_OFFSET_Z:0.00

[INFO] [1527053811.322688]: Flushing first 200 IMU entries...
[INFO] [1527053815.423192]: Publishing IMU data...
[ INFO] [1527053815.443853411]: Initializing Imu sensor
[ INFO] [1527053815.485029365]: Imu sensor activated

sz@sz-Default-string:~$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
Reading from the keyboard and Publishing to Twist!
-----
Moving around:
  u  i  o
  j  k  l
  m  ,  .

For Holonomic mode (strafing), hold down the shift key:
-----
  U  I  O
  J  K  L
  M  <  >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:  speed 0.17    turn 1.0
```

6. 通过控制键盘上的 u、i、o、j、k、l、m、U、I、O、J、K、L、M 按键去控制机器人的移动。

2.2 视觉跟随

1. 打开新终端，输入以下指令。

```
$ roslaunch carebot_follower follower.launch
```

```
silver@ubuntu: ~  
silver@ubuntu:~$ roslaunch carebot_follower follower.launch
```

2. 回车，出现如下画面。

图！

3. 输入完成后，若无报错，站在机器人的正面（摄像头方向为正面），等待识别，识别之后机器人会随着人的移动而跟随移动。

注意：人到摄像头的跟随距离大约半米左右。

三、地图构建

1、启动机器人，输入如下指令，运行 SLAM 功能包

```
$ roslaunch carebot_navigation gmapping_ls01d_lidar.launch
```

```
sz@sz-Default-string: ~  
sz@sz-Default-string:~$ roslaunch carebot_navigation gmapping_ls01d_lidar.launch
```

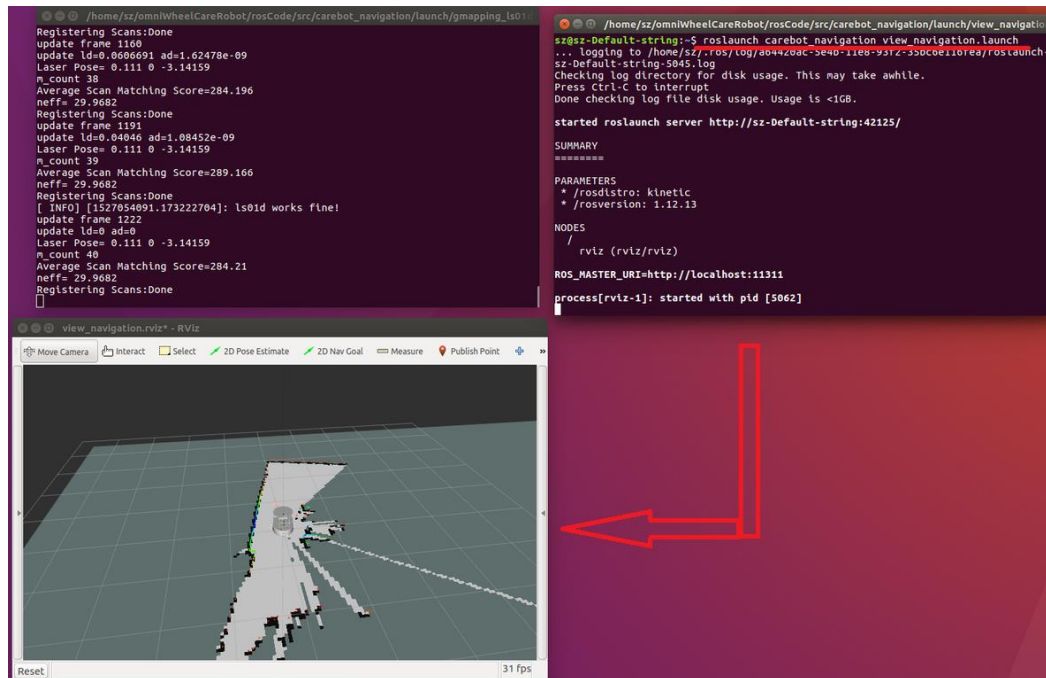
2. 回车，出现如下画面。

```
/home/sz/omniWheelCareRobot/rosCode/src/carebot_navigation/launch/gmapping_ls01d  
Registering Scans:Done  
update frame 265  
update ld=0.0411577 ad=1.08001e-09  
Laser Pose= 0.111 0 -3.14159  
m_count 9  
Average Scan Matching Score=291.024  
neff= 29.9875  
Registering Scans:Done  
update frame 296  
update ld=0.0406145 ad=1.04854e-09  
Laser Pose= 0.111 0 -3.14159  
m_count 10  
Average Scan Matching Score=289.091  
neff= 29.9875  
Registering Scans:Done  
[ INFO] [1527054000.974907881]: ls01d works fine!  
update frame 327  
update ld=0.0204262 ad=4.98617e-10  
Laser Pose= 0.111 0 -3.14159  
m_count 11  
Average Scan Matching Score=283.116  
neff= 29.9875  
Registering Scans:Done
```

3. 打开新终端，输入如下指令。

```
$ roslaunch carebot_navigation view_navigation.launch
```

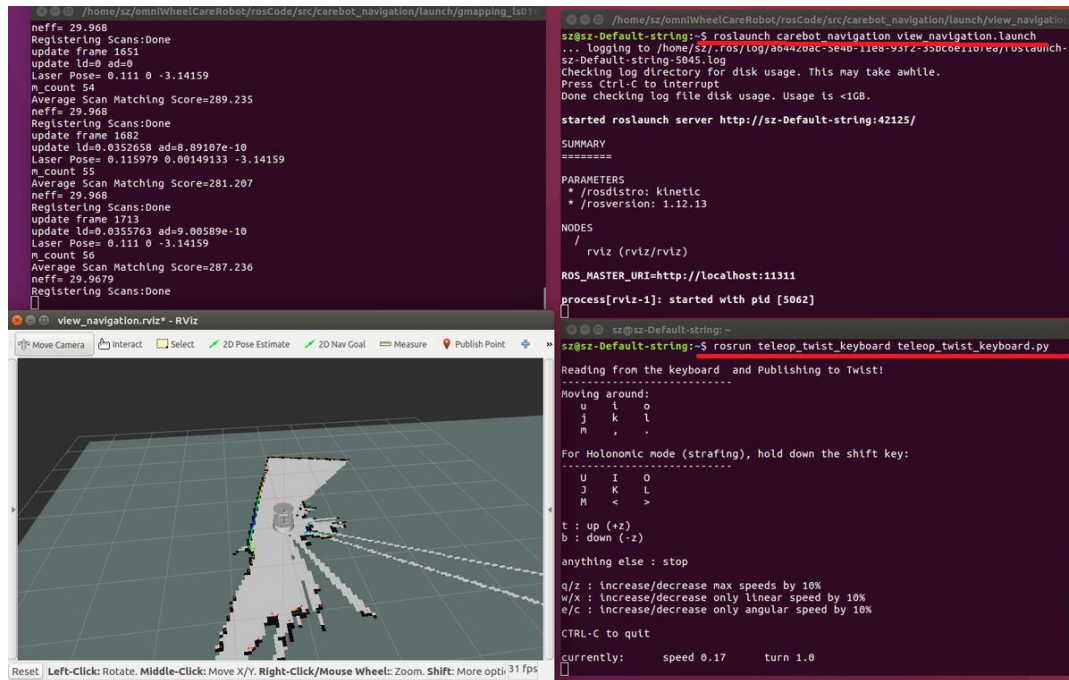
4. 回车，出现如下画面，准备进行地图构建。



5. 打开新终端，输入如下指令。

```
$ rosrn teleop_twist_keyboard teleop_twist_keyboard.py
```

6. 出现如下画面。这时可以通过 i、l、j 等键来控制机器人行走，完成地图的构建。



7、完成地图的构建需要保存地图，另起终端，输入如下命令，保存地图。

```
$ roslaunch map_server map_saver -f mymap
```

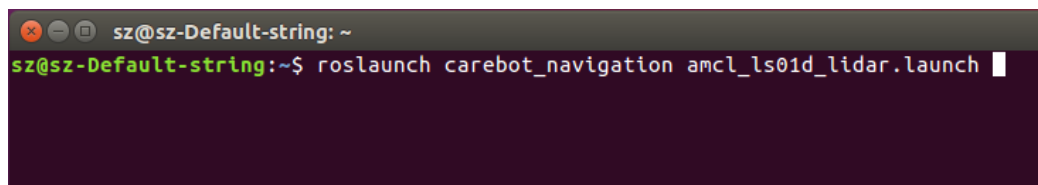
8. 输入如下指令，将地图放到指定的路径下面。

```
$ mv ~/mymap.* omniWheelCareRobot/rosCode/src/carebot_navigation/maps/
```

四、导航

1. 打开终端，输入如下命令。

```
$ roslaunch carebot_navigation amcl_ls01d_lidar.launch
```



2. 回车，出现如下画面。


```
/home/sz/omniWheelCareRobot/rosCode/src/carebot_navigation/launch/amcl_ls01d_lida

[INFO] [1527054374.217623]: Flushing first 200 IMU entries...
[ INFO] [1527054374.355840500]: Loading from pre-hydro parameter style
[ INFO] [1527054374.411763401]: Using plugin "static_layer"
[ INFO] [1527054374.428560651]: Requesting the map...
[ INFO] [1527054374.634511624]: Resizing costmap to 192 X 416 at 0.050000 m/pix
[ INFO] [1527054374.734391741]: Received a 192 X 416 map at 0.050000 m/pix
[ INFO] [1527054374.750078892]: Using plugin "obstacle_layer"
[ INFO] [1527054374.758922444]: Subscribed to Topics: scan
[ INFO] [1527054374.793760010]: Using plugin "inflation_layer"
[ INFO] [1527054374.846634752]: Loading from pre-hydro parameter style
[ INFO] [1527054374.863683066]: Using plugin "obstacle_layer"
[ INFO] [1527054374.865613194]: Subscribed to Topics: scan
[ INFO] [1527054374.887051196]: Using plugin "inflation_layer"
[ INFO] [1527054374.937576973]: Created local_planner dwa_local_planner/DWAPlann
erROS
[ INFO] [1527054374.939717336]: Sim period is set to 0.05
[ INFO] [1527054375.616478884]: Recovery behavior will clear layer obstacles
[ INFO] [1527054375.624239186]: Recovery behavior will clear layer obstacles
[ INFO] [1527054375.681088632]: odom received!
[INFO] [1527054378.319628]: Publishing IMU data...
[ INFO] [1527054378.340515607]: Initializing Imu sensor
[ INFO] [1527054378.442098622]: Imu sensor activated
```

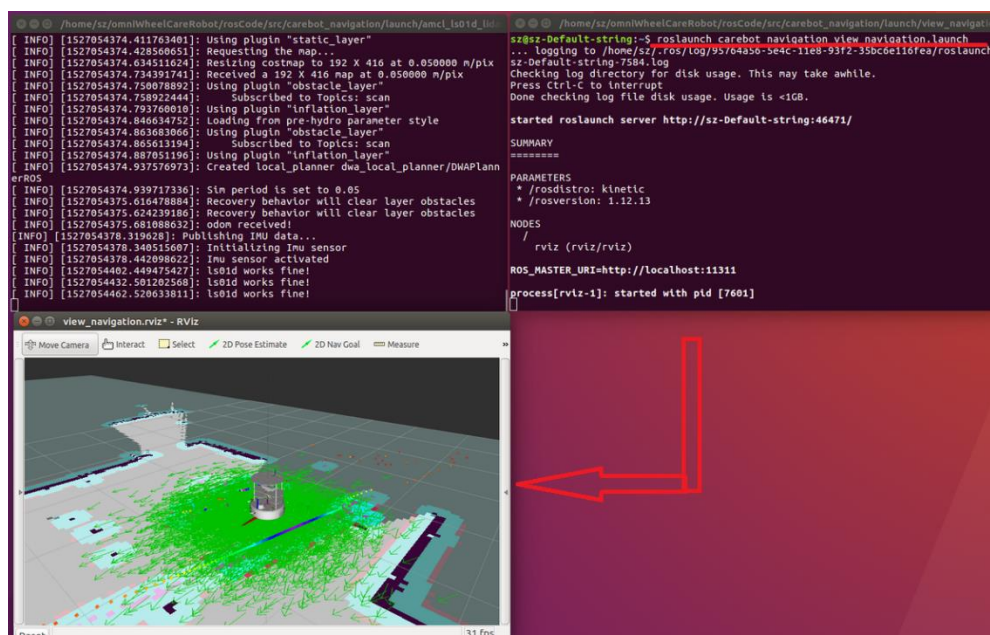
3. 打开新终端，输入如下命令。

```
$ roslaunch carebot_navigation view_navigation.launch
```

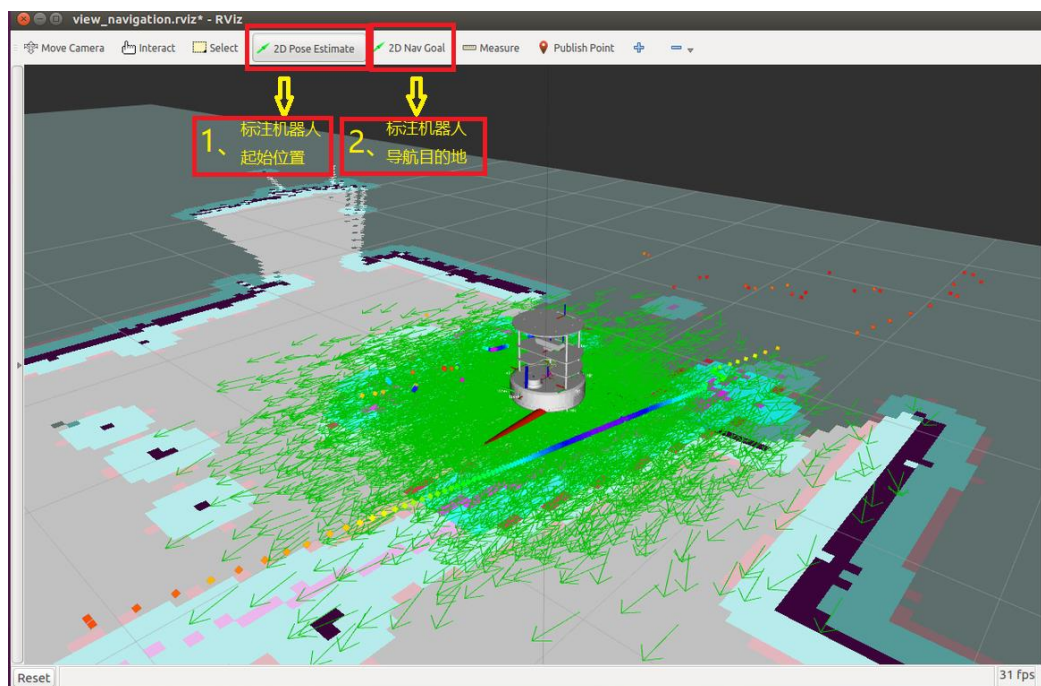
```
/home/sz/omniWheelCareRobot/rosCode/src/carebot_navigation/launch/amcl_ls01d_lida
[INFO] [1527054374.217623]: Flushing first 200 IMU entries...
[ INFO] [1527054374.355840500]: Loading from pre-hydro parameter style
[ INFO] [1527054374.411763401]: Using plugin "static_layer"
[ INFO] [1527054374.428560651]: Requesting the map...
[ INFO] [1527054374.634511624]: Resizing costmap to 192 X 416 at 0.050000 m/pix
[ INFO] [1527054374.734391741]: Received a 192 X 416 map at 0.050000 m/pix
[ INFO] [1527054374.750078892]: Using plugin "obstacle_layer"
[ INFO] [1527054374.758922444]: Subscribed to Topics: scan
[ INFO] [1527054374.793760010]: Using plugin "inflation_layer"
[ INFO] [1527054374.846634752]: Loading from pre-hydro parameter style
[ INFO] [1527054374.863683066]: Using plugin "obstacle_layer"
[ INFO] [1527054374.865613194]: Subscribed to Topics: scan
[ INFO] [1527054374.887051196]: Using plugin "inflation_layer"
[ INFO] [1527054374.937576973]: Created local_planner dwa_local_planner/DWAPlann
erROS
[ INFO] [1527054374.939717336]: Sim period is set to 0.05
[ INFO] [1527054375.616478884]: Recovery behavior will clear layer obstacles
[ INFO] [1527054375.624239186]: Recovery behavior will clear layer obstacles
[ INFO] [1527054375.681088632]: odom received!
[INFO] [1527054378.319628]: Publishing IMU data...
[ INFO] [1527054378.340515607]: Initializing Imu sensor
[ INFO] [1527054378.442098622]: Imu sensor activated
[ INFO] [1527054402.449475427]: ls01d works fine!

sz@sz-Default-string:~$ roslaunch carebot_navigation view_navigation.launch
```

4. 回车，出现如下画面。



5. 点击下图所示红框 1 里的按键，然后在地图上标注机器人的起始位置和方向；
然后按红框 2 里的按键，在地图上标注机器人要前往的目的地。标注好位置后机器人将自动移动到之前所标的目标位置。



五、多点导航

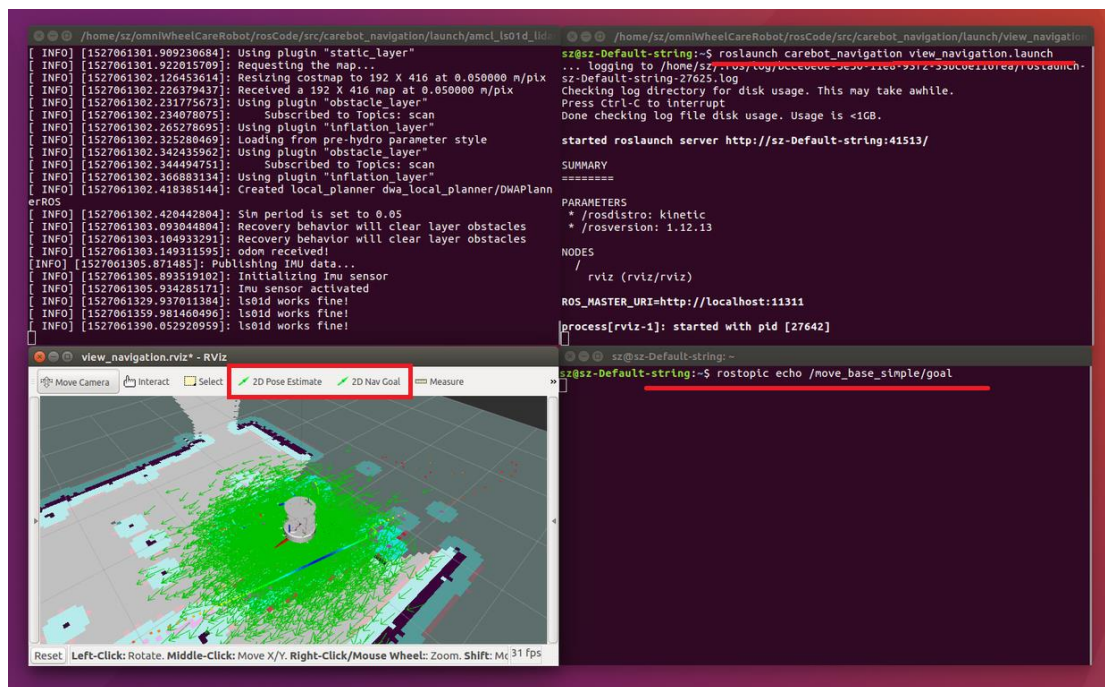
1. 重复第四章 1、2 小节，打开导航。

```
$ roslaunch carebot_navigation amcl_ls01d_lidar.launch
```

```
$ roslaunch carebot_navigation view_navigation.launch
```

2、打开新终端，输入如下指令，当选定小车的移动位置后将会通过这个终端返回小车的位置坐标以及姿态信息。（如果没有移动小车将没有返回值）

```
$ rostopic echo /move_base_simple/goal
```



3. 在地图上标注小车的起始位置和方向，然后再标注小车的第一个目标位置，当小车移动到所标第一目标位置后，继续标注小车的第二个目标位置（此时不需要再标注起始位置，而只需再次点击目标位置键设定第二个目标位置），依次标注第三个目标位置、第四个.....

4、在标注小车目标位置的之后，小车每移动到一个目标位置 echo 界面就会相继返回一个位置的坐标点和小车此时的姿态信息。



5. 根据下一条提示信息找到该路径下的文件并双击打开。

omniWheelCareRobot/rosCode/src/carebot_navigation/nodes # 文件路径

6. 将 echo 界面的返回值和下图（来自所打开的文件）进行比对，将文件中的数值改为返回值，其中 echo 界面的 x、y、z 和图片中的 point 对应，orientation

的 4 位与 Quaternion 对应。

```
# Pose coordinates are then displayed in the terminal
# that was used to launch RViz.
locations = dict()
locations['one'] = Pose(Point(0.582, -0.903, 0.000), Quaternion(0.000, 0.000, -0.630, 0.776))
locations['two'] = Pose(Point(1.281, 1.210, 0.000), Quaternion(0.000, 0.000, 0.643, 0.766))
locations['three'] = Pose(Point(1.579, 1.906, 0.000), Quaternion(0.000, 0.000, 0.897, -0.442))
locations['four'] = Pose(Point(0.618, 0.563, 0.000), Quaternion(0.000, 0.000, -0.032, 0.999))
```

7、重启导航步骤，重复第四章 1、2 小节，打开导航。

```
$ roslaunch carebot_navigation amcl_ls01d_lidar.launch
```

```
$ roslaunch carebot_navigation view_navigation.launch
```

8. 在文件夹中找到此目录

```
omniWheelCareRobot/rosCode/src/carebot_navigation/node
```

然后在这个目录的空白处重新打开一个新终端，在新终端加上如下指令，结果如下图所示

```
$ ./ position_nav.py
```

图！

9. 在地图上标注小车的初始位置即可，小车将开始多点导航。