

# ROS 导航平台、ROS 计算单元使用说明

水若波科技有限公司

网址：[www.waterobo.com](http://www.waterobo.com)

<b>第 1 章 导航平台综述</b> .....	<b>1</b>
1.1 综述.....	1
<b>第 2 章 硬件安装</b> .....	<b>1</b>
2.1 激光雷达安装.....	1
2.2 ROS 计算单元安装.....	2
<b>第 3 章 硬件连接</b> .....	<b>2</b>
3.1 激光雷达连接.....	2
3.2 ROS 计算单元连接.....	2
<b>第 4 章 软件配置</b> .....	<b>3</b>
<b>附录：更改记录</b> .....	<b>7</b>
更改记录：.....	7
0.01 初始版本.....	7

## 第 1 章 导航平台综述

### 1.1 综述

ROS 导航平台，可以用于学习 ROS 基础知识，ROS 导航相关算法。首先 ROS 系统导航硬件基本框图如下所示，包含三大部分：

① ROS 导航平台，输入为 CMD\_VEL（速度指令），输出为 ODOM（里程计）。也就是说该平台受控于 ROS 计算单元，并向 ROS 计算单元上报里程信息。

② 激光雷达，作为导航建图的传感器，输出为 SCAN（详见 ROS Wiki），软件包启动以后激光雷达开始旋转并采集测量的距离数据。

③ ROS 计算单元，作为机器人的大脑，所有的软件和算法运行在此平台。一般搭在 Ubuntu 系统，运行 ROS 次操作系统。



图：ROS 系统导航硬件基本框图

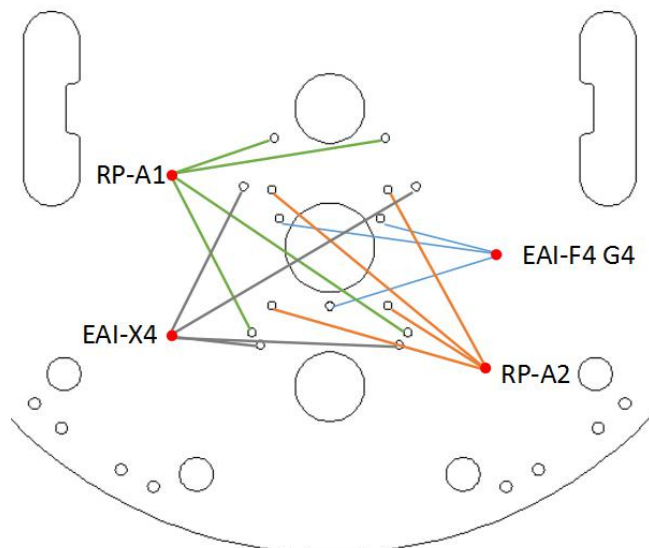
## 第 2 章 硬件安装

激光雷达和 ROS 计算单元都需要安装在 ROS 导航平台上。

### 2.1 激光雷达安装

ROS 导航平台支持两个厂家的激光雷达，上海思岚科技和深圳 EAI 科技系列激

光雷达。根据您购买的激光雷达类型对应不同的安装孔进行安装，下图是 ROS 导航平台对应激光雷达安装示意图。



图：激光雷达安装示意图

## 2.2 ROS 计算单元安装

由于计算单元各异，预留安装孔复杂。所以一般采用橡皮筋或魔术贴进行固定。

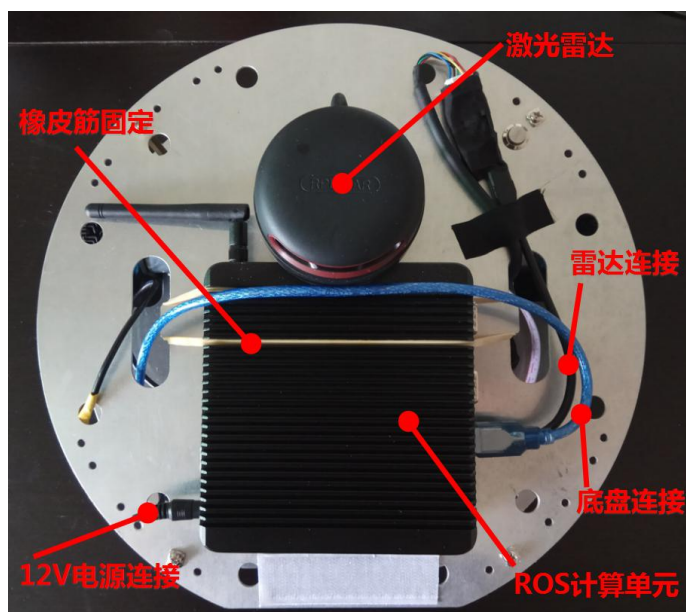
# 第 3 章 硬件连接

## 3.1 激光雷达连接

使用思岚 A2 激光雷达，激光雷达安装完成后将 USB 插头连接 ROS 计算单元的一个 USB 口。

## 3.2 ROS 计算单元连接

ROS 计算单元连接包括 12V 电源和底盘控制接口。首先，连接 12V-DC 插头至 ROS 计算单元，然后连接 USB 插头连接 ROS 计算单元的一个 USB 口。如下图所示所有硬件和连接安装完成的效果图。

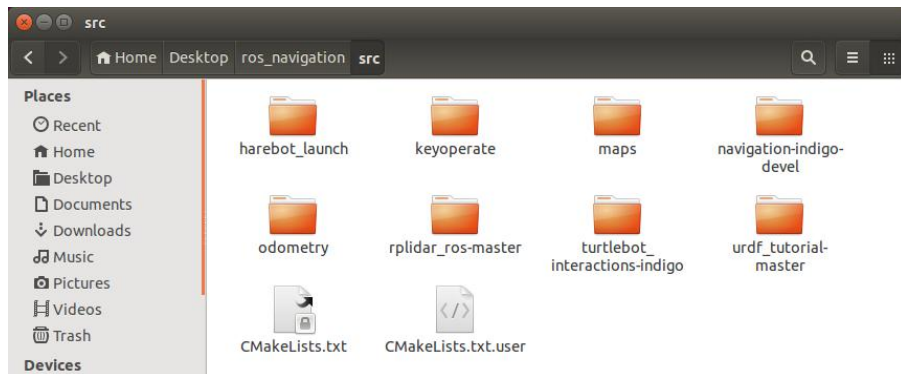


图：硬件连接安装完成图

## 第 4 章 软件配置

① 连接 ROS 计算单元的 HDMI 或 VGA 接口到显示器，连接无线鼠标和键盘到 ROS 计算单元，供电并开机启动 ROS 计算单元，默认密码：aa，进入 ubuntu 系统桌面。

② 在官网下载软件包 `ros_navigation.tar.gz`，并在复制到 ubuntu 桌面上，将软件包解压到该目录下，如下图所示是整个目录结构。其中 `src` 目录下文件夹的作用如下，`harebot_launch`：该文件夹下主要是启动文件（具体介绍见下文）；`keyoperate`：该文件夹下主要包含使用键盘控制底盘运动的源代码；`maps`：该文件夹用来放置地图；`navigation-indigo-devel`：是 ROS 导航包，详细信息请搜索 `ros navigation`；`odometry`：该文件夹包含与底盘通讯的相关源代码，包括控制底盘协议和接收底盘数据协议封装解析；`rplidar_ros-master`：包含激光雷达相关源代码（若使用其他雷达，需要替换该包）；`turtlebot_interactions-indigo`：该文件夹下主要包含 `rviz` 显示相关源代码；`urdf_tutorial-master`：该文件夹主要是机器人模型相关参数和描述文件。



图：目录结构

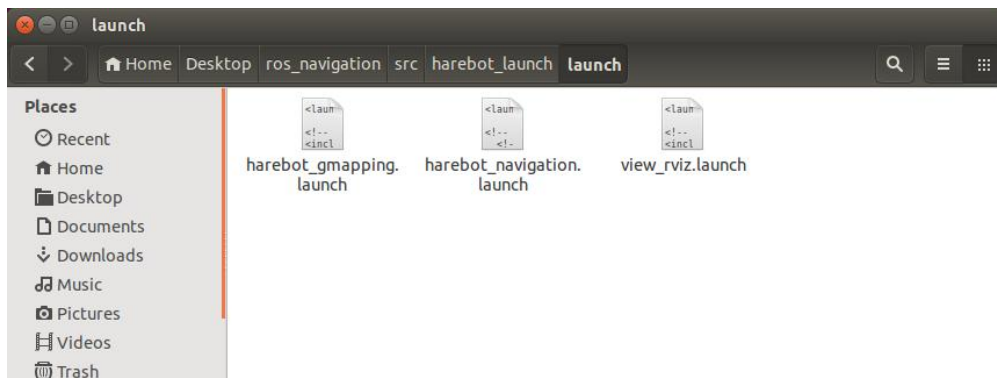
③ 编译程序,打开终端运行 `cd ~/Desktop/ros_navigation;` 然后运行 `catkin_make` 就可以编译文件夹下的所有代码。

④ 环境变量设置，在 `~/.bashrc` 文件中加入编译生成的 `devel` 文件夹下的 `setup.bash` 路径，加入该环境变量可以识别 ROS 的包文件。（如果是使用自己的电脑，编译过程中可能会遇到错误，在网上查找一下依赖库并安装就好了）。



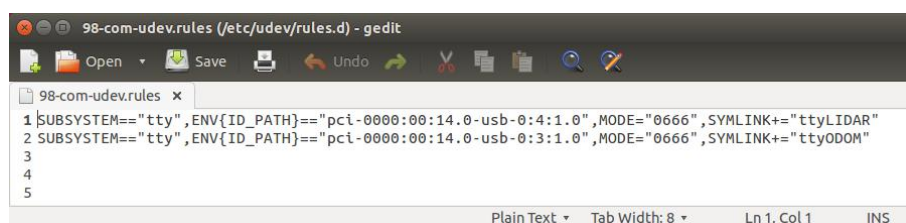
图：用户文件

⑤ `harebot_launch` 的 `launch` 文件夹下有三个启动文件，如下图所示：其中 `harebot_gmapping.launch` 是建图启动文件，`harebot_navigation.launch` 是导航启动文件，`view_rviz.launch` 是 RVIZ 显示启动文件，后面会用到。



图：文件描述

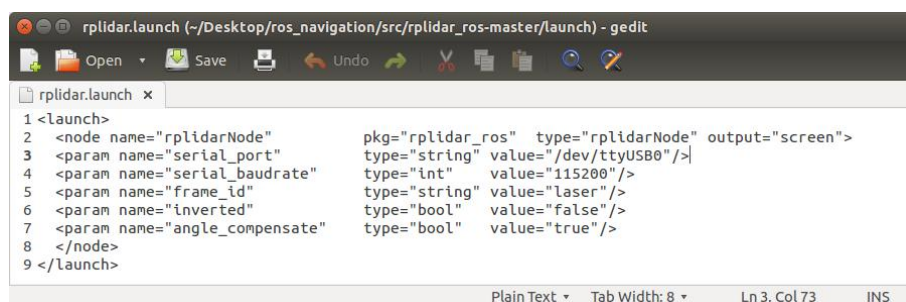
⑥ 绑定激光雷达和底盘的串口号：由于底盘和雷达都使用了 USB-TTL 转换芯片，Ubuntu 系统会将先插入的 USB-TTL 串口识别成 /dev/ttyUSB0，后插入的识别成 /dev/ttyUSB1，以此类推。为了使用方便，我们需要给雷达和底盘固定 USB 口和名称。详细参考博客 <https://blog.csdn.net/xiangbin099/article/details/79676053>。固定 USB 口和名称以后，只要不改变接口都可以直接使用。如下图所示，在 /etc/udev/rules.d/ 目录下的文件 98-com-udev.rules 的内容，这里我们把雷达串口号绑定为 ttyLIDAR，底盘串口号绑定为 ttyODOM。



```
98-com-udev.rules (etc/udev/rules.d) - gedit
1 | $SUBSYSTEM=="tty", ENV{ID_PATH}=="pci-0000:00:14.0-usb-0:4:1.0", MODE="0666", SYMLINK+="ttyLIDAR"
2 | SUBSYSTEM=="tty", ENV{ID_PATH}=="pci-0000:00:14.0-usb-0:3:1.0", MODE="0666", SYMLINK+="ttyODOM"
3
4
5
```

图：绑定串口号

⑦ 单独调试激光雷达，打开雷达源文件夹 rplidar\_ros-master/launch 下的 rplidar.launch 文件，修改 serial\_port 的 value 值，改成我们绑定的雷达名称 /dev/ttyLIDAR。然后打开终端输入：roslaunch rplidar\_ros rplidar.launch 启动雷达。如果雷达正常旋转起来，就证明雷达调试完成。



```
rplidar.launch (~/Desktop/ros_navigation/src/rplidar_ros-master/launch) - gedit
1 <launch>
2   <node name="rplidarNode"          pkg="rplidar_ros" type="rplidarNode" output="screen">
3     <param name="serial_port"      type="string" value="/dev/ttyUSB0"/>|
4     <param name="serial_baudrate"  type="int"    value="115200"/>
5     <param name="frame_id"         type="string" value="laser"/>
6     <param name="inverted"         type="bool"   value="false"/>
7     <param name="angle_compensate" type="bool"   value="true"/>
8   </node>
9 </launch>
```

图：雷达调试

⑧ 单独调试底盘运动，打开底盘 odometry/src 下的 odometry\_node.cpp，找到 main 函数如下图，将 /dev/ttyODOM，将名称改成您自己绑定的底盘串口名称（若您绑定为 ttyODOM 那么不用修改）。再一次编译整个包，然后打开终端运行 rosrund odometry odometry\_node，打开另一个终端运行 rosrund keyoperate keyoperate\_node，然

后使用键盘的“上下左右”键就可以控制机器人运动，使用“0”可以使机器人立刻停止运动，使用“+”可以调节机器人运行速度。

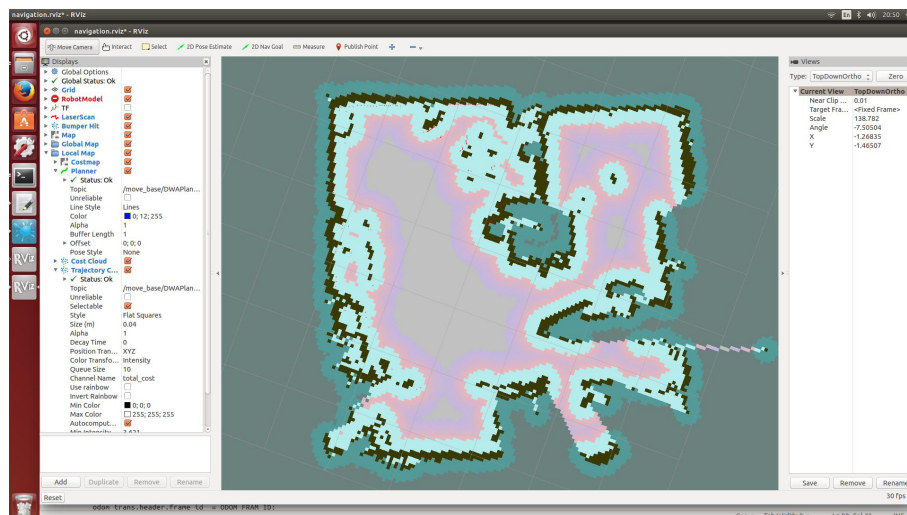
```

140 }
141 /*****
142 * 函数名:
143 *****/
144 int main(int argc, char** argv)
145 {
146     ros::init(argc, argv, "odometry_node");
147     ros::NodeHandle nh;
148     ROS_INFO("Odometry node start!");
149     odom_broadcaster = new tf::TransformBroadcaster();
150     sub_cmdvel = nh.subscribe(SUB_CMD_VEL, 100, &cmd_vel_callback);
151     pub_odom = nh.advertise<nav_msgs::Odometry>(PUB_ODOM, 100);
152     try{
153         chassis_uart_device.open("/dev/ttyODOM", 115200);|
154     }
155     catch(cereal::Exception& e){
156         ROS_INFO("fail to open serial port!");

```

图：mian 函数

⑨ 创建地图，如果以上步骤都已完成。那么就可以遥控机器人创建地图了。关掉所有的终端。打开新的终端，运行 `roslaunch harebot_launch harebot_gmapping.launch` 就可以开始创建地图了，运行 `roslaunch harebot_launch view_rviz.launch` 可以使用 RVIZ 观察机器人位置和建图的效果，如下图所示。使用步骤 7 遥控机器人运动，地图就会慢慢增加大小，直到创建完所有地方的地图。

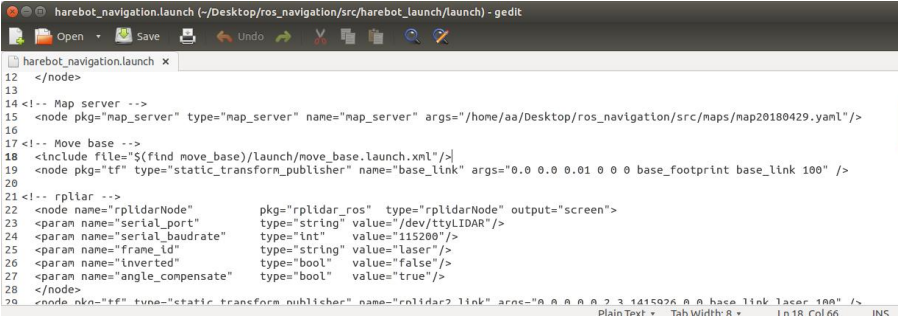


图：创建地图



⑩ 保存地图，当地图创建完成后。打开新的终端运行 `roslaunch map_server map_saver -f ~/map` 就会在 `home` 目录下生产 `map` 文件，包括 `map.pgm` 和 `map.yaml`，将两个文件拷贝到 `maps` 文件夹下，就可以使用了。

⑪ 导航，为了使导航能够找到该 `map` 文件，需要打开 `harebot_navigation.launch` 文件修改 `args` 中的地图路径，修改完成后。关闭所有终端，启动新的终端，运行 `roslaunch harebot_launch harebot_navigation.launch`，就可以开启导航了。运行 `roslaunch harebot_launch view_rviz.launch` 可以显示机器人的地图、当前位置等信息。首先需要使用 `RVIZ` 上的按钮手动给机器人定位，定位完成后，就可以用 `RVIZ` 发送目标点，让机器人导航到该目标点了。



```
harebot_navigation.launch x
12 </node>
13
14 <!-- Map server -->
15 <node pkg="map_server" type="map_server" name="map_server" args="/home/aa/Desktop/ros_navigation/src/maps/map20180429.yaml"/>
16
17 <!-- Move base -->
18 <include file="$(find move_base)/launch/move_base.launch.xml"/>
19 <node pkg="tf" type="static_transform_publisher" name="base_link" args="0.0 0.0 0.01 0 0 0 base_footprint base_link 100" />
20
21 <!-- rplidar -->
22 <node name="rplidarNode" pkg="rplidar_ros" type="rplidarNode" output="screen">
23 <param name="serial_port" type="string" value="/dev/ttyLIDAR"/>
24 <param name="serial_baudrate" type="int" value="115200"/>
25 <param name="frame_id" type="string" value="laser"/>
26 <param name="inverted" type="bool" value="false"/>
27 <param name="angle_compensate" type="bool" value="true"/>
28 </node>
29 <node pkg="tf" type="static_transform_publisher" name="rplidar2_link" args="0.0 0.0 0.2 3.1415926 0.0 base_link laser 100" />
```

图：更改 map 路径

## 附录：更改记录

### 更改记录：

#### 0.01 初始版本