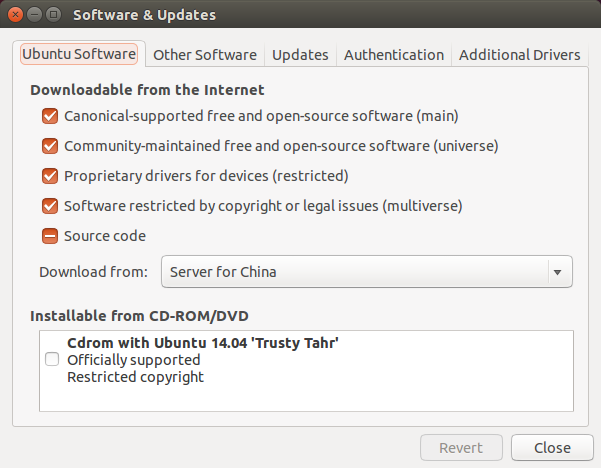
操作系统：Ubuntu 16.04

ROS kinetic

## ros安装

1. 安装并配置ROS环境

配置Ubuntu仓库，"restricted," "universe," and "multiverse." 可界面软件更新那进行设置。



1. 安装源

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

1. 增加Key

sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 0xB01FA116

1. 更新

sudo apt-get update

1. 安装:

ROS, rqt, rviz, robot-generic libraries, 2D/3D simulators, navigation and 2D/3D perception

1. 解决依赖

sudo rosdep init

rosdep update

1. 环境设置

echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc

source ~/.bashrc

1. 安装rosinstall

sudo apt-get install python-rosinstall

1. catkin方式创建ROS工作空间

$ mkdir -p ~/catkin\_ws/src

$ cd ~/catkin\_ws/src

## Turtlebot包安装

sudo apt-get update

sudo apt-get install ros-indigo-turtlebot ros-indigo-turtlebot-apps ros-indigo-turtlebot-interactions ros-indigo-turtlebot-simulator ros-indigo-kobuki-ftdi ros-indigo-rocon-remocon ros-indigo-rocon-qt-library ros-indigo-ar-track-alvar-msgs

## 测试Turtlebot安装

1. **名称概念**

* Turtlebot只指底盘+kinect+托盘+支架的合体。
* 跟turtlebot相连的电脑称为：上网本
* 远程到上网本的称为：工作站、工作机

1. **测试**

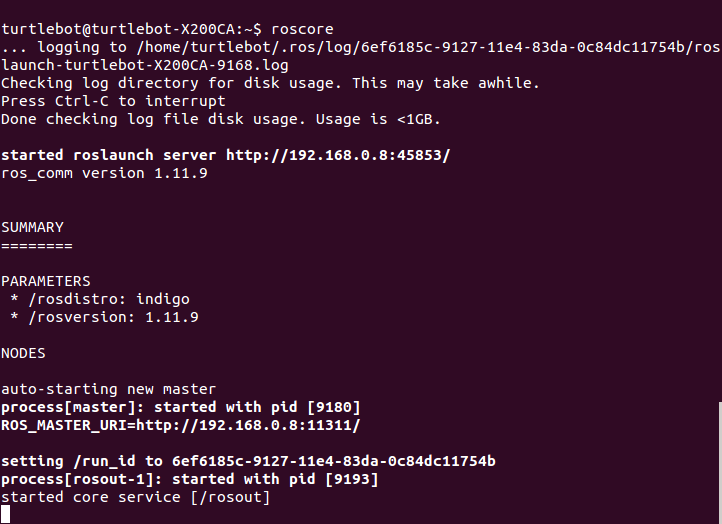
在上网本，新开终端输入

roscore

如果ROS安装正确显示

started core service [/rosout]

如图：



通过遥控移动（通过Turtlebot的上网本操作）

1. 打开Turtlebot开关，状态点亮。
2. 启动Turtlebot，在上网本，打开一个新终端，运行如下命令：

roslaunch turtlebot\_bringup minimal.launch

1. 启动键盘遥控，在上网本，打开另一个新终端，运行如下命令：

roslaunch turtlebot\_teleop keyboard\_teleop.launch

1. 点击键盘“i”前进。

## 安装rviz

1. 新建目录

$ mkdir -p ~/catkin\_ws/src

1. 下载turtlebot\_viz包并编译

$ cd ~/catkin\_ws/src

$ git clone https://github.com/ncnynl/turtlebot\_viz

$ cd ..

$ catkin\_make

1. **测试：**
2. 运行roscore

$ roscore

1. 运行rviz

$ roslaunch turtlebot\_rviz\_launcher view\_navigation.launch

## 配置网络

1. 在工作站和Turtlebot都要同步时钟。输入如下命令：

sudo apt-get install chrony

sudo ntpdate ntp.ubuntu.com

1. 获取Turtlebot和工作站的IP地址，在终端执行：

ifconfig

1. 查看SSh状态

sudo service ssh status

如果显示ssh: unrecognized service，没安装ssh,则安装：

sudo apt-get install openssh-server

1. 工作站远程登录Turtlebot上网本，在终端执行

ssh turtlebot@[ip\_of\_turtlebot]

1. Turtlebot配置参数

echo export ROS\_MASTER\_URI=http://localhost:11311 >> ~/.bashrc

echo export ROS\_HOSTNAME=IP\_OF\_TURTLEBOT >> ~/.bashrc

1. 在工作站配置参数

echo export ROS\_MASTER\_URI=http://IP\_OF\_TURTLEBOT:11311 >> ~/.bashrc

echo export ROS\_HOSTNAME=IP\_OF\_PC >> ~/.bashrc

注意：IP\_OF\_TURTLEBOT为turtlebot的Ip地址，IP\_OF\_PC为工作站的IP地址

1. 验证从工作站到Turtlebot

在Turtlebot终端执行：

roslaunch turtlebot\_bringup minimal.launch --screen

在工作站终端执行：

roslaunch turtlebot\_teleop keyboard\_teleop.launch

可以在工作站的键盘点击“i”，控制Turtlebot前进.

在工作站终端执行：

rostopic list

如果没看到主题列表，检查turtlebot上网本和工作站的ROS\_MASTER\_URI配置

在工作站终端执行：

rostopic echo /diagnostics

如果没有得到警告：topic has not been published，检查turtlebot上网本和工作站的的ROS\_HOSTNAME

1. 验证从Turtlebot到工作站

在工作站终端执行：

rostopic pub -r10 /hello std\_msgs/String "hello"

在TurtleBot终端执行：

rostopic echo /hello

消息“你好”开始打印每秒约10次。如果没有，检查工作站的ROS\_HOSTNAME设置。

## 激光雷达(Rplidar)gmapping构建地图

1. **准备**

* 测试环境ubuntu14.04 + indigo
* 已安装Turtlebot包，具体安装：[deb包安装](http://www.ncnynl.com/archives/201611/1085.html" \t "https://www.ncnynl.com/archives/201611/_blank) | [源码安装](http://www.ncnynl.com/archives/201611/1084.html" \t "https://www.ncnynl.com/archives/201611/_blank)
* 依赖包：

Turtlebot应用包，https://github.com/ncnynl/turtlebot\_apps.git

激光雷达rplidar一代驱动，https://github.com/ncnynl/rplidar\_ros.git

1. **步骤**

**雷达配置：**

**建立工作空间（也可以利用现有的），编译包**

**$ mkdir -p ~/turtlebot\_ws/src**

**$ cd ~/turtlebot\_ws/src**

**## turtlebot建图依赖包**

**$ git clone https://github.com/turtlebot/turtlebot\_apps**

**#编译**

**$ cd ~/turtlebot\_ws**

**$ catkin\_make**

**添加环境变量，在〜/ .bashrc中最后添加一行：**

**$ source /home/ubu/turtlebot\_ws/devel/setup.bash**

**刷新配置**

**$ source ~/.bashrc**

**创建激光雷达（Rplidar）的串口别名**

**确认idVendor和idProduct，ID后面的部分idVendor：idProduct**

**$ lsusb**

**Bus 001 Device 006: ID 10c4:ea60**

**新建/etc/udev/rules.d/rplidar.rules文件，内容如下:(别名为rplidar，实际名称为：/ dev / rplidar）**

**KERNEL=="ttyUSB\*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", MODE:="0666", GROUP:="dialout", SYMLINK+="rplidar"**

**增加当前用户对串口的默认访问权限：**

**$ sudo usermod -a -G dialout 用户名**

**使UDEV配置生效:(使串口的默认访问权限生效，需要重启机器）**

**$ sudo service udev reload**

**$ sudo service udev restart**

**制作雷达驱动启动文件**

**复制rplidar.launch到rplidar-laser.launch，并增加TF定义**

**$ roscd turtlebot\_navigation**

**$ mkdir -p laser/driver**

**$ sudo cp ~/turtlebot\_ws/src/rplidar\_ros/launch/rplidar.launch laser/driver/rplidar\_laser.launch**

**打开rplidar\_laser.launch，并修改**

**$ rosed turtlebot\_navigation rplidar\_laser.launch**

**检查frame\_id是否指定为激光**

**<param name="frame\_id" type="string" value="laser"/>**

**查看serial\_port是否指定正确端口，使用别名**

**检查端口：<param name="serial\_port" type="string" value="/dev/rplidar"/>，设置好别名，或直接端口/ dev / ttyUSB0**

**增加TF：**

**<node pkg="tf" type="static\_transform\_publisher" name="base\_to\_laser" args="0.0 0.0 0.18 0 0.0 0.0 base\_link laser 100"/>**

**修改为args =“0.0 0.0 0.18 0 0.0 0.0为自己的实际安装位置。**

**static\_transform\_publisher部分**

**static\_transform\_publisher xyz qx qy qz qw frame\_id child\_frame\_id period\_in\_ms**

**<launch>**

**<node name="rplidarNode" pkg="rplidar\_ros" type="rplidarNode" output="screen">**

**<param name="serial\_port" type="string" value="/dev/ttyUSB0"/>**

**<param name="serial\_baudrate" type="int" value="115200"/>**

**<param name="frame\_id" type="string" value="laser"/>**

**<param name="inverted" type="bool" value="false"/>**

**<param name="angle\_compensate" type="bool" value="true"/>**

**</node>**

**<node pkg="tf" type="static\_transform\_publisher" name="base\_to\_laser" args="0.0 0.0 0.18 0 0.0 0.0 base\_link laser 100"/>**

**</launch>**

**检查turtlebot\_navigation包**

**增加rplidar\_gmapping\_demo.launch文件，用于启动gmapping。**

**$ roscd turtlebot\_navigation**

**$ touch launch/rplidar\_gmapping\_demo.launch**

**$ rosed launch/rplidar\_gmapping\_demo.launch**

**输入内容：**

**<launch>**

**<!-- Define laser type-->**

**<arg name="laser\_type" default="rplidar" />**

**<!-- laser driver -->**

**<include file="$(find turtlebot\_navigation)/laser/driver/$(arg laser\_type)\_laser.launch" />**

**<!-- Gmapping -->**

**<arg name="custom\_gmapping\_launch\_file" default="$(find turtlebot\_navigation)/launch/includes/gmapping/$(arg laser\_type)\_gmapping.launch.xml"/>**

**<include file="$(arg custom\_gmapping\_launch\_file)"/>**

**<!-- Move base -->**

**<include file="$(find turtlebot\_navigation)/launch/includes/move\_base.launch.xml"/>**

**</launch>**

**设置laser\_type为rplidar。**

**增加rplidar\_gmapping.launch.xml文件，执行gmapping建图**

**$ roscd turtlebot\_navigation**

**$ touch launch/includes/gmapping/rplidar\_gmapping.launch.xml**

**$ rosed launch/includes/gmapping/rplidar\_gmapping.launch.xml**

**输入内容：**

**<launch>**

**<arg name="scan\_topic" default="scan" />**

**<arg name="base\_frame" default="base\_footprint"/>**

**<arg name="odom\_frame" default="odom"/>**

**<node pkg="gmapping" type="slam\_gmapping" name="slam\_gmapping" output="screen">**

**<param name="base\_frame" value="$(arg base\_frame)"/>**

**<param name="odom\_frame" value="$(arg odom\_frame)"/>**

**<param name="map\_update\_interval" value="0.01"/>**

**<param name="maxUrange" value="4.0"/>**

**<param name="maxRange" value="5.0"/>**

**<param name="sigma" value="0.05"/>**

**<param name="kernelSize" value="3"/>**

**<param name="lstep" value="0.05"/>**

**<param name="astep" value="0.05"/>**

**<param name="iterations" value="5"/>**

**<param name="lsigma" value="0.075"/>**

**<param name="ogain" value="3.0"/>**

**<param name="lskip" value="0"/>**

**<param name="minimumScore" value="30"/>**

**<param name="srr" value="0.01"/>**

**<param name="srt" value="0.02"/>**

**<param name="str" value="0.01"/>**

**<param name="stt" value="0.02"/>**

**<param name="linearUpdate" value="0.05"/>**

**<param name="angularUpdate" value="0.0436"/>**

**<param name="temporalUpdate" value="-1.0"/>**

**<param name="resampleThreshold" value="0.5"/>**

**<param name="particles" value="8"/>**

**<!--**

**<param name="xmin" value="-50.0"/>**

**<param name="ymin" value="-50.0"/>**

**<param name="xmax" value="50.0"/>**

**<param name="ymax" value="50.0"/>**

**make the starting size small for the benefit of the Android client's memory...**

**-->**

**<param name="xmin" value="-1.0"/>**

**<param name="ymin" value="-1.0"/>**

**<param name="xmax" value="1.0"/>**

**<param name="ymax" value="1.0"/>**

**<param name="delta" value="0.05"/>**

**<param name="llsamplerange" value="0.01"/>**

**<param name="llsamplestep" value="0.01"/>**

**<param name="lasamplerange" value="0.005"/>**

**<param name="lasamplestep" value="0.005"/>**

**<remap from="scan" to="$(arg scan\_topic)"/>**

**</node>**

**</launch>**

**测试激光雷达gmapping构建地图：**

**新开端口，打开roscore**

**$ roscore**

**新开端口，启动turtlebot**

**$ roslaunch turtlebot\_bringup minimal.launch**

**新开端口，启动gmapping，用于构建地图**

**$ roslaunch turtlebot\_navigation rplidar\_gmapping\_demo.launch**

**新开端口，启动rviz，实时查看建图情况**

**$ roslaunch turtlebot\_rviz\_launchers view\_navigation.launch**

**构建地图结束保存地图**

**新开端口，建立目录，保存地图**

**$ mkdir -p ~/map**

**$ rosrun map\_server map\_saver -f ~/map/rplidar\_gmapping**

**$ ls ~/map #查看内容，包含rplidar\_gmapping.pgm rplidar\_gmapping.yaml**

**查看地图，已经生成rplidar\_gmapping.pgm文件，可以用图像浏览器（gimp，eog，gthumb，等等）打开查看。**

**AMCL：**

**首先运行机器人节点：**

**roslaunch rbx1\_bringup fake\_turtlebot.launch**

**然后运行AMCL节点，使用测试地图：**

**roslaunch rbx1\_nav fake\_amcl.launch map:=test\_map.yaml**

**然后运行rviz：**

**rosrun rviz rviz -d `rospack find rbx1\_nav`/nav\_fuerte.vcg**

**这时在rviz中就应该显示出了地图和机器人，通过rviz在地图上选择目标位置了，然后就会看到机器人自动规划出一条全局路径，并且导航前进。**