1. ROS操作系统安装与配置
2. Cartographer安装与配置
3. 激光雷达配置

安装urg\_node：sudo apt-get install ros-kinetic-urg-node

获取端口权限：sudo chmod a+rw /dev/ttyACM0

1. IMU配置（\*）

安装xsense\_driver、获取端口权限

1. 配置.launch和.lua文件

.launch：

1. Use sim time参数

<param name="/use\_sim\_time" value="true" />

通过去bag文件建图是，value=true；在实时建图时，value=false

1. Cartographer node节点

<node name="cartographer\_node" pkg="cartographer\_ros"

type="cartographer\_node" args="

-configuration\_directory $(find cartographer\_ros)/configuration\_files

-configuration\_basename demo\_hokuyo.lua"

output="screen">

<remap from="scan" to="/scan" />

</node>

Cartographer的启动节点，args中为.lua文件的路径

1. Rviz节点

<node name="rviz" pkg="rviz" type="rviz" required="true"

args="-d $(find cartographer\_ros)/configuration\_files/demo\_hokuyo.rviz" />

rviz启动节点，用于浏览建图过程，args中为.rviz文件的路径，.rviz文件可由rviz软件生成

1. 激光雷达启动节点

<node name="hokuyo" pkg="urg\_node" type="urg\_node">

<param name="port" type="string" value="/dev/ttyACM0"/>

<param name="frame\_id" value="base\_laser\_link" />

</node>

北洋激光雷达的启动节点urg\_node，其中port参数为激光雷达的端口，frame\_id参数为激光雷达的输出topic /scan在tf中对应的frame

1. IMU启动节点（\*）
2. Flat world imu节点

<node name="flat\_world\_imu\_node" pkg="cartographer\_turtlebot"

type="cartographer\_flat\_world\_imu\_node" output="screen">

<remap from="imu\_in" to="/mobile\_base/sensors/imu\_data\_raw" />

<remap from="imu\_out" to="/imu" />

</node>

用于将平面gyro的二维输入数据转换为三维imu数据的节点，用于turtlebot自带的imu（只有x轴y轴的数据，z轴为null），使用输出符合**[sensor\_msgs/Imu](http://docs.ros.org/api/sensor_msgs/html/msg/Imu.html)**的imu时不需要使用这个节点

1. Playbag节点

<node name="playbag" pkg="rosbag" type="play"

args="--clock $(arg bag\_filename)" />

用于读取bag文件的节点，只在读取bag文件时使用

.lua文件：

配置主要参数

**map\_frame**：发布子图（submap）的ROS框架

**tracking\_frame**：建图算法追踪的ROS框架，即作为子图坐标基准的框架，如果使用了IMU，则必须设置为IMU输入所在的框架

**published\_frame**：用于发布位姿的ROS框架，在使用外部里程计时设置为里程计输入的odom框架，否则一般设置为base\_link

**odom\_frame**：位于published\_frame和map\_frame之间，用于发布SLAM结果的框架，只在使用SLAM算法导出的虚拟里程计信息（provide\_odom\_frame为真）时有效

**provide\_odom\_frame**：启用时在map\_frame下发布一个odom\_frame，提供开环连续的位姿输出

**use\_odometry**：启用时，订阅ROS系统中发布的odom消息，输入数据格式为nav\_msgs/Odometry，使用外部里程计输入时启用，SLAM算法会读取这部分数据

**num\_laser\_scans**：订阅的激光扫描消息数量，使用一个激光雷达时订阅scan消息，使用多个激光雷达时分别订阅scan\_1、scan\_2等消息

**num\_point\_clouds**：订阅的点云消息数量，使用输入为sensor\_msgs/PointCloud2的测距仪时启用设置

**lookup\_transform\_timeout\_sec**：寻找变换的最大时间，超过这个时间则反馈未找到变换

**submap\_publish\_period\_sec**：子图发布周期，一般设置为0.3秒

**pose\_publish\_period\_sec**：位姿发布周期，一般设置为200赫兹，即5e-3秒

设置使用trajectory\_builder\_2d为真，并进行相关配置

按照激光雷达的说明书设置**max\_laser\_range**、**min\_laser\_range**等参数；设置是否使用imu数据，动作滤波motion\_filter的相关参数，sparse\_pose\_graph建图包的相关参数等

一切cartographer相关包的配置都在.lua文件中设置

1. 录制和读取bag文件
2. 录制bag文件：

记录所有topic rosbag record -a

记录一部分topic

rosbag record -O subset /turtle1/cmd\_vel /turtle1/pose

bag文件将被命名为subset.bag，并记录后面列出的两个topic

也可以把rosbag record写入.launch文件（这个方法我还没有尝试过）

<node pkg="rosbag" type="record" name="bag\_record" arg="/temp /accl"/>

1. 读取bag文件：

roslaunch cartographer\_ros demo\_hokuyo.launch

bag\_filename:=${HOME}/Downloads/demo\_hokuyo.bag

在命令行输入这个命令即可读取demo\_hokuyo.bag并启动建图程序demo\_hokuyo.launch，注意.bag文件的存放路径即可

1. 通过ssh转存bag文件：

首先通过命令行“ssh 对方ip”的方式联通两台电脑，再使用scp命令传输文件

scp file username@hostIP:文件地址 例:

scp demo\_hokuyo.bag myc@192.168.0.xx:/home/Downloads/demo\_hokuyo.bag

1. 其他注意
2. 修改.launch和.lua文件后，须重新在catkin\_ws 目录下进行编译

cd catkin\_ws

catkin\_make\_isolated --install --use-ninja

source install\_isolated/setup.bash

1. 可将每次启动终端都需要使用的命令写入.bashrc中，.bashrc位于Home文件夹中，为隐藏文件，可以在Home文件夹下按ctrl+h即可看到