## 安装

//安装wstool、rosdep和ninjia工具

sudo apt-get update

sudo apt-get install -y python-wstool python-rosdep ninja-build

// Create a new workspace in 'catkin\_ws'.

mkdir catkin\_ws

cd catkin\_ws

wstool init src//初始化一个空的工作空间

// 将准备要下载的文件目录保存在src目录下

wstool merge -t src <https://raw.githubusercontent.com/googlecartographer/cartographer_ros/master/cartographer_ros.rosinstall>

//下载软件到src目录中

wstool update -t src

//rosdep用来安装系统的依赖项，就是这个命令帮助cartographer安//装了谷歌的各个工具：gflags、gmock等

rosdep init//安装之后需要初始化一次

rosdep update

// generate a bash script and then execute it.

//安装src文件夹下所有package的所有依赖，但是不会安装ros //package path 和src文件下的依赖

rosdep install --from-paths src --ignore-src --rosdistro=${ROS\_DISTRO} -y

//因为cartographer包含了非ros package，需要编译这些package

//就需要使用catkin\_make\_isolated –install命令，此处的编译器使用

//的是google开发的ninja编译器。因为ninja的编译速度更快。

//详见如下参考文献http://www.ros.org/reps/rep-0134.html

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

source install\_isolated/setup.bash

注：在Unix/Linux下通常使用Makefile来控制代码的编译，但是Makefile对于比较大的项目有时候会比较慢，看看上面那副漫画，代码在编译都变成了程序员放松的借口了。所以这个Google的程序员在开发Chrome的时候因为忍受不了Makefile的速度，自己重新开发出来一套新的控制编译的工具叫作Ninja，Ninja相对于Makefile这套工具更注重于编译速度。除了Chrome现在还有一些其他的比较大的项目也在开始使用Ninja，比如LLVM。我试用了一下感觉还是不错，比如编译Cmake时间大概是原来的1/4。

注：在执行wstool update -t src安装命令时，需要从谷歌的服务中下载Ceres，需要翻墙。如果翻墙还不行那直接登录<https://ceres-solver.googlesource.com/ceres-solver.git/+/1.12.0rc4>，下载对应的版本然后直接复制到src文件夹下。

## 二、配置参数选项

**map\_frame：**The ROS frame ID to use for publishing submaps, the parent frame of poses, usually “map”.

**tracking\_frame：**The ROS frame ID of the frame that is tracked by the SLAM algorithm. If an IMU is used, it should be at its position, although it might be rotated. A common choice is“imu\_link”.

**published\_frame：**The ROS frame ID to use as the child frame for publishing poses. For example “odom” if an “odom” frame is supplied by a different part of the system. In this case the pose of “odom” in the map\_frame will be published. Otherwise, setting it to “base\_link” is likely appropriate.

**odom\_frame：**Only used if provide\_odom\_frame is true. The frame between published\_frame and map\_frame to be used for publishing the (non-loop-closed) local SLAM result. Usually “odom”.

**provide\_odom\_frame：**If enabled, the local, non-loop-closed, continuous pose will be published as the odom\_frame in the map\_frame.

**use\_odometry：**If enabled, subscribes to nav\_msgs/Odometry on the topic “odom”. Odometry must be provided in this case, and the information will be included in SLAM.

**use\_laser\_scan：**If enabled, the node subscribes to sensor\_msgs/LaserScan on the “scan” topic. If 2D SLAM is used, either this or use\_multi\_echo\_laser\_scan must be enabled.

**use\_multi\_echo\_laser\_scan：**If enabled, the node subscribes to sensor\_msgs/MultiEchoLaserScan on the “echoes” topic. If 2D SLAM is used, either this or use\_laser\_scan must be enabled.在一个角度能够返回多个测量值

**num\_point\_clouds：**Number of 3D lasers to subscribe to. Must be a positive value if and only if using 3D SLAM. Subscribes to sensor\_msgs/PointCloud2 on the “points2” topic for one laser, or topics “points2\_1”, “points2\_2”, etc for multiple lasers.

**lookup\_transform\_timeout\_sec：**Timeout in seconds to use for looking up transforms using tf2.

**submap\_publish\_period\_sec：**Interval in seconds at which to publish the submap poses, e.g. 0.3 seconds.

**pose\_publish\_period\_sec：**Interval in seconds at which to publish poses, e.g. 5e-3 for a frequency of 200 Hz.

## 三、订阅/发布的topic：

**订阅的topic：**

**scan (**[**sensor\_msgs/LaserScan**](http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html)**)**

Supported in 2D and 3D (e.g. using an axially rotating planar laser scanner). If use\_laser\_scan is enabled in the [Configuration](https://google-cartographer-ros.readthedocs.io/en/latest/configuration.html), this topic will be used as input for SLAM.

**echoes (**[**sensor\_msgs/MultiEchoLaserScan**](http://docs.ros.org/api/sensor_msgs/html/msg/MultiEchoLaserScan.html)**)**

Supported in 2D and 3D. If use\_multi\_echo\_laser\_scan is enabled in the [Configuration](https://google-cartographer-ros.readthedocs.io/en/latest/configuration.html), this topic will be used as input for SLAM. Only the first echo is used.

**points2 (**[**sensor\_msgs/PointCloud2**](http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html)**)**

Only supported in 3D. If num\_point\_clouds is set to 1 in the [Configuration](https://google-cartographer-ros.readthedocs.io/en/latest/configuration.html), this topic will be used as input for SLAM. If num\_point\_clouds is greater than 1, multiple numbered points2 topics (i.e. points2\_1, points2\_2, points2\_3, ... up to and including num\_point\_clouds) will be used as inputs for SLAM.

**imu (sensor\_msgs/Imu)**

Supported in 2D (optional) and 3D (required). This topic will be used as input for SLAM.

**odom (nav\_msgs/Odometry)**

Supported in 2D (optional) and 3D (optional). If use\_odometry is enabled in the Configuration, this topic will be used as input for SLAM.

发布的topic：

**map (nav\_msgs/OccupancyGrid)**

Only supported in 2D. If subscribed to, a background thread will continuously compute and publish the map. The time between updates will increase with the size of the map. For faster updates, use the submaps APIs.

**scan\_matched\_points2 (sensor\_msgs/PointCloud2)**

Point cloud as it was used for the purpose of scan-to-submap matching. This cloud may be both filtered and projected depending on the Configuration.

**submap\_list (cartographer\_ros\_msgs/SubmapList)**

List of all submaps, including the pose and latest version number of each submap, across all trajectories.

## 四、发布/需求的tf

需求的frame：*tracking\_frame* 和 *published\_frame*

发布的frame：map\_frame and published\_frame is always provided；

provide\_odom\_frame is enabled in the Configuration, a continuous transform between the configured odom\_frame and published\_frame will be provided.