

# ros2指令：

---

编译指定包：

```
colcon build --packages-select <name-of-pkg>
```

编译指定包及其依赖

```
colcon build --packages-up-to <name-of-pkg>
```

创建功能包（C++）

```
ros2 pkg create --build-type ament_cmake <name-of-pkg>
```

Python

```
ros2 pkg create --build-type ament_python <name-of-pkg>
```

查看数据类型

```
ros2 interface show <数据类型>
```

## 任务一：

---

使用Publisher-Subscriber结构，完成两个node节点间传输数据功能

### (1) 文本传输

---

订阅者

```
ros2 run topic_text_video topic_text_subscriber
```

发布者

```
ros2 run topic_text_video topic_text_publisher
```

## (2) 视频传输

---

订阅者

```
ros2 run topic_text_video topic_video_subscriber
```

发布者

```
ros2 run topic_text_video topic_video_publisher
```

## 任务二：

---

完成URDF、rviz、Gazebo的探索学习

创建激光雷达，使用URDF创建自定义模型，在Gazebo中搭建仿真环境，在rviz模拟器中使用键盘完成对小车的控制，并实时在Gazebo中完成构图

### (1) 使用rviz展示URDF自定义模型

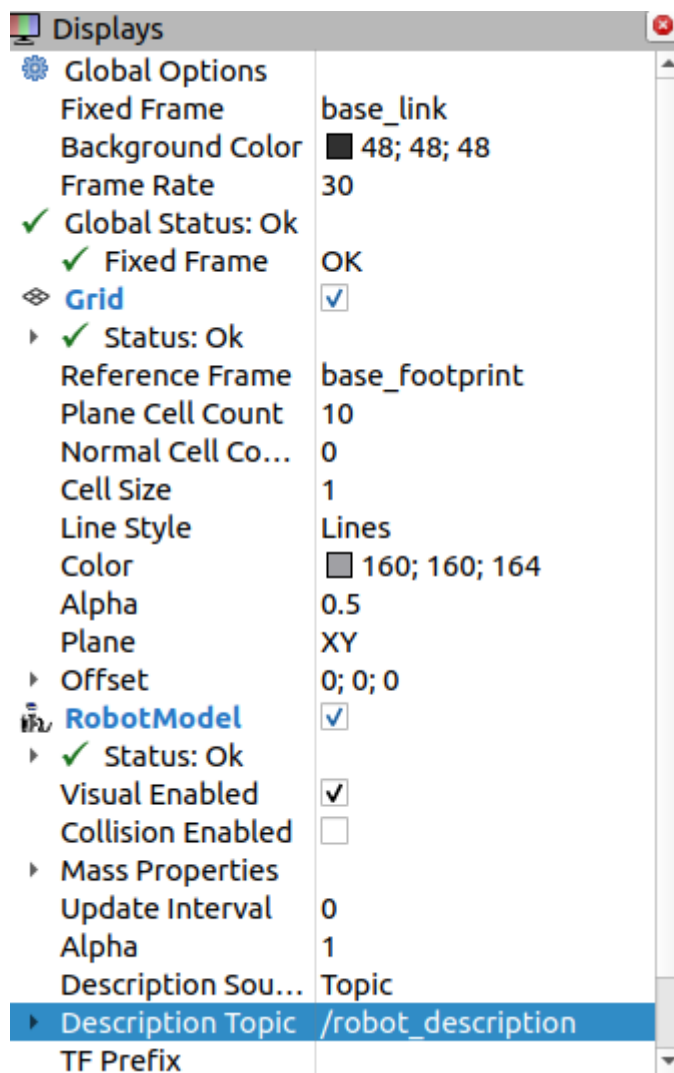
---

先启动左右轮控制器再启动主体

```
ros2 run laser_car_description rotate_wheel
```

```
ros2 launch laser_car_description display_rviz2.launch.py
```

相关设置：

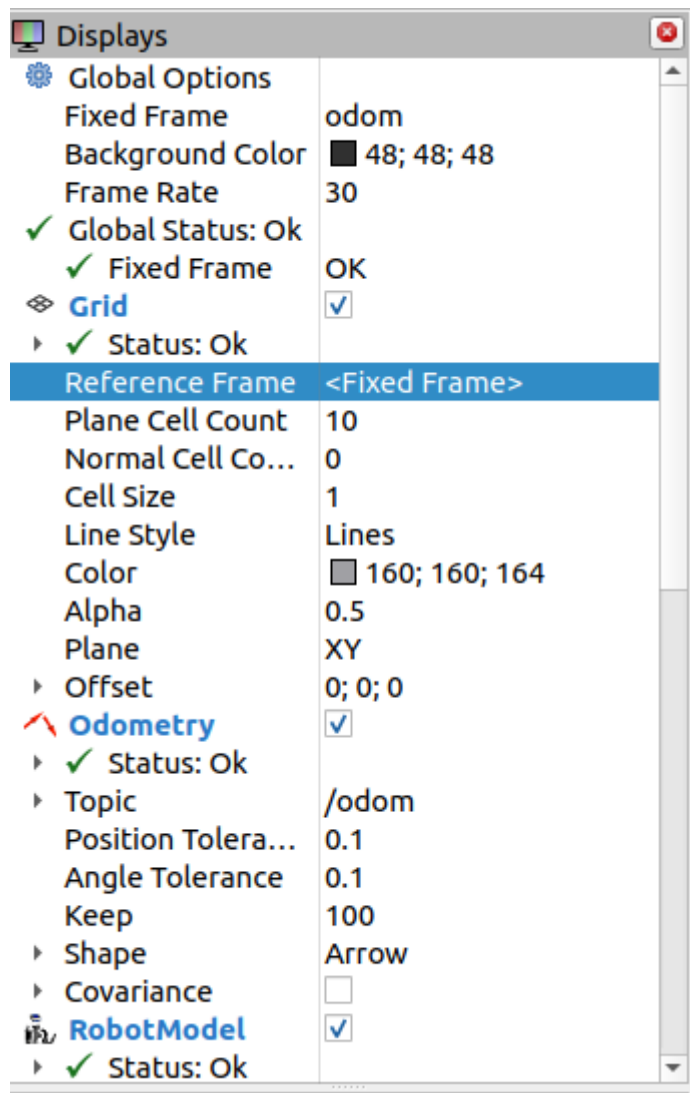


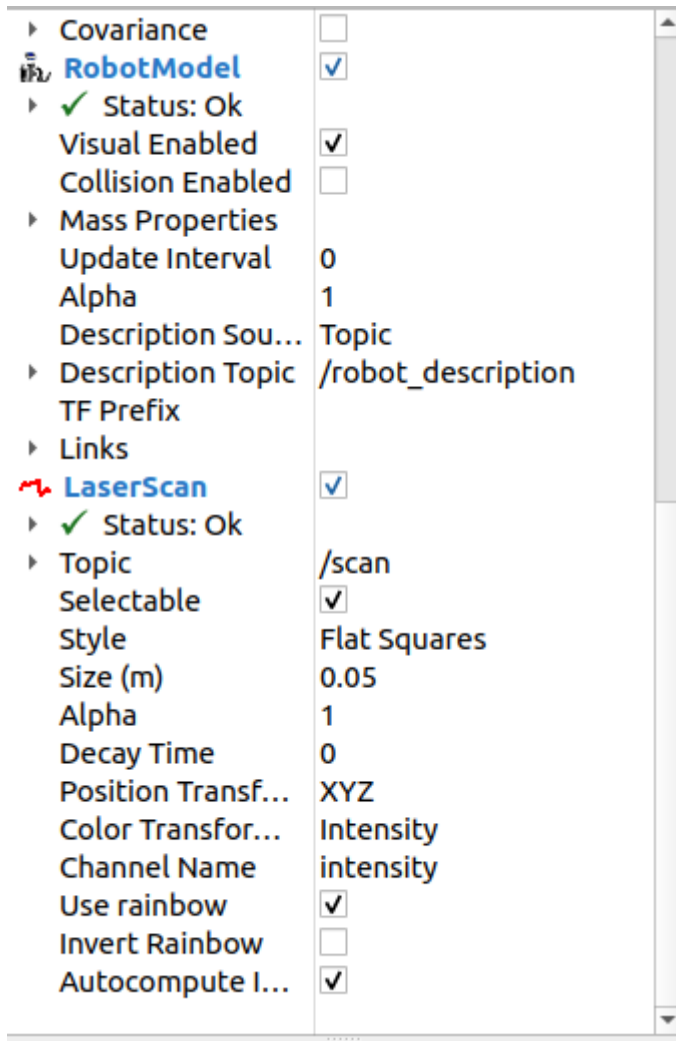
## (2) 使用gazebo和rviz联动展示小车以及激光雷达等数据

启动rviz和gazebo

```
ros2 launch laser_car_description rviz_gazebo.launch.py
```

rviz中设置如下





使用键盘控制小车运动

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

### (3) 在gazebo中搭建仿真环境并通过SLAM在RVIZ中完成建图

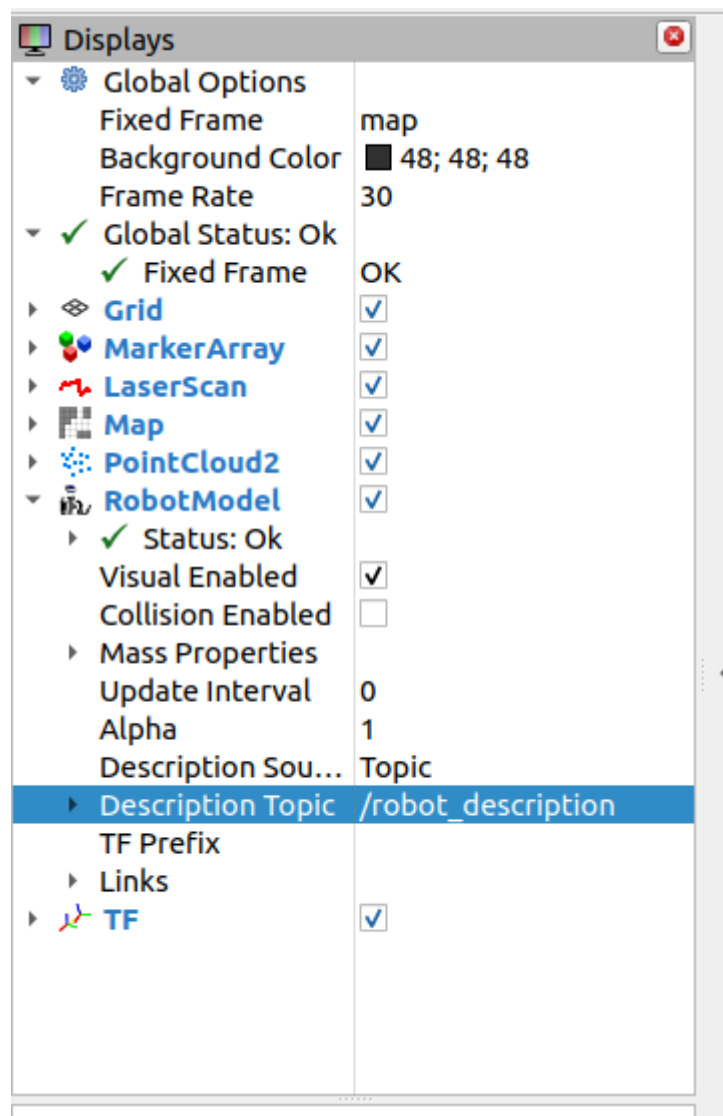
启动建图前，需要先启动gazebo仿真环境，因为我们的建图程序依赖于Gazebo提供雷达和里程计等数据。

```
ros2 launch laser_car_description rviz_gazebo_world.launch.py
```

启动建图

```
ros2 launch laser_car_cartographer cartographer.launch.py
```

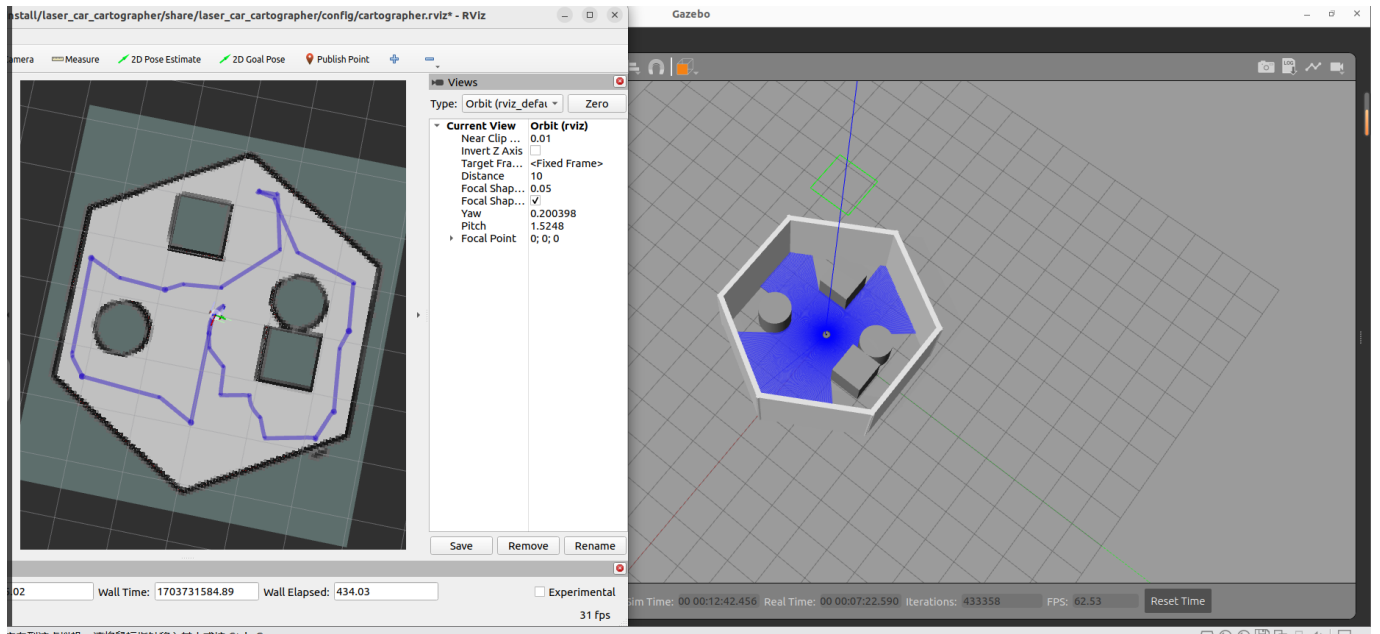
rviz配置如下



启动键盘，开始建图

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

结果如下



保存地图

在map目录下执行

```
ros2 run nav2_map_server map_saver_cli -t map -f <地图名>
```

加载保存的地图

在map目录下执行

```
ros2 run nav2_map_server map_server --ros-args --param yaml_filename:=<参数文件名>
```

新开一个终端

```
rviz2
```

新开终端进行配置

```
ros2 lifecycle set /map_server configure
```

激活配置

```
ros2 lifecycle set /map_server activate
```

在rviz2中通过添加map组件

若仍然不显示图片，则先取消激活，再激活

```
ros2 lifecycle set /map_server deactivate
```

```
ros2 lifecycle set /map_server activate
```

## 任务三：

### (1) 基础：在rviz中选定起点和终点，完成自主导航

运行仿真

```
ros2 launch laser_car_description gazebo_world.launch.py
```

运行导航

```
ros2 launch laser_car_navigation2 laser_car_nav2.launch.py
```

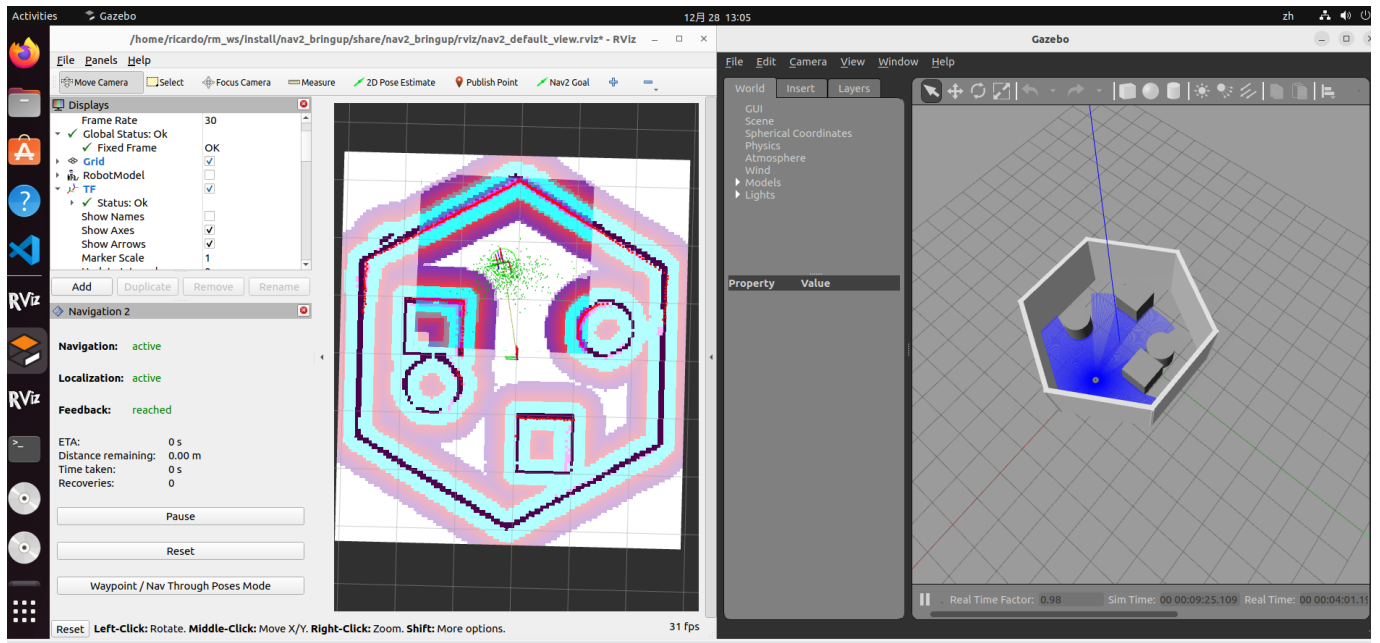
初始化位置

```
ros2 run navigation2_cmd init_robot_pose
```

或者使用rviz中的“2D Pose Estimate”给激光雷达小车确定一个初始的方位

然后通过“Nav2 Goal”确定目标点，让小车自主寻路到达目的地。



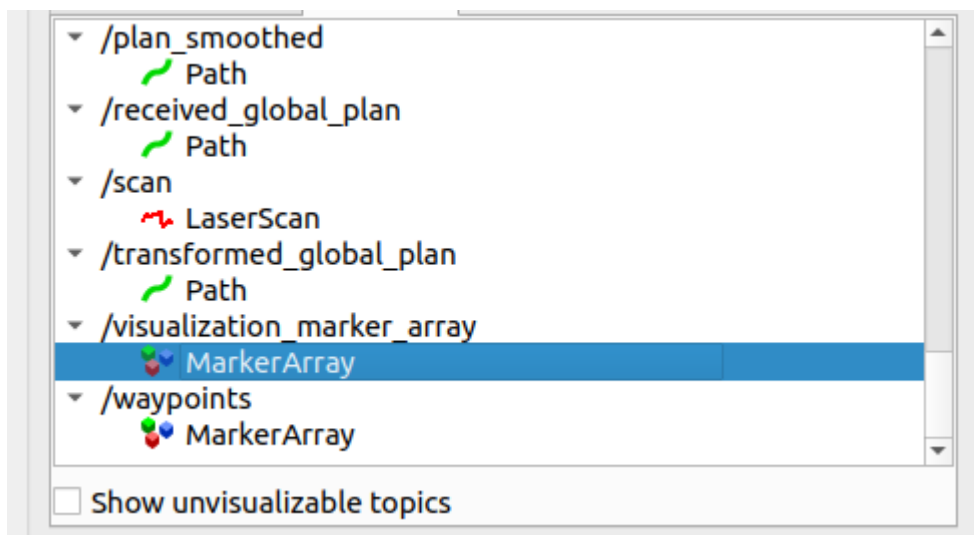


## (2) 在rviz中指定路径，使小车能够循环运行实现巡视功能

获取指定点序列

```
ros2 run navigation2_cmd set_up_inspection_points
```

在rviz中通过topic添加MarkerArray序列展示标记点



使用publishpoint发布点

启动巡视

```
ros2 run navigation2_cmd start_patrol
```