机器人URDF模型

注：虚拟机需要与小车处在同一个局域网下，且ROS_DOMAIN_ID，需要一致，可以查看【使用前必看】来设置板子上的IP和ROS_DOMAIN_ID。

1、程序功能说明

小车连接上代理，运行程序，rviz中会显示URDF模型。

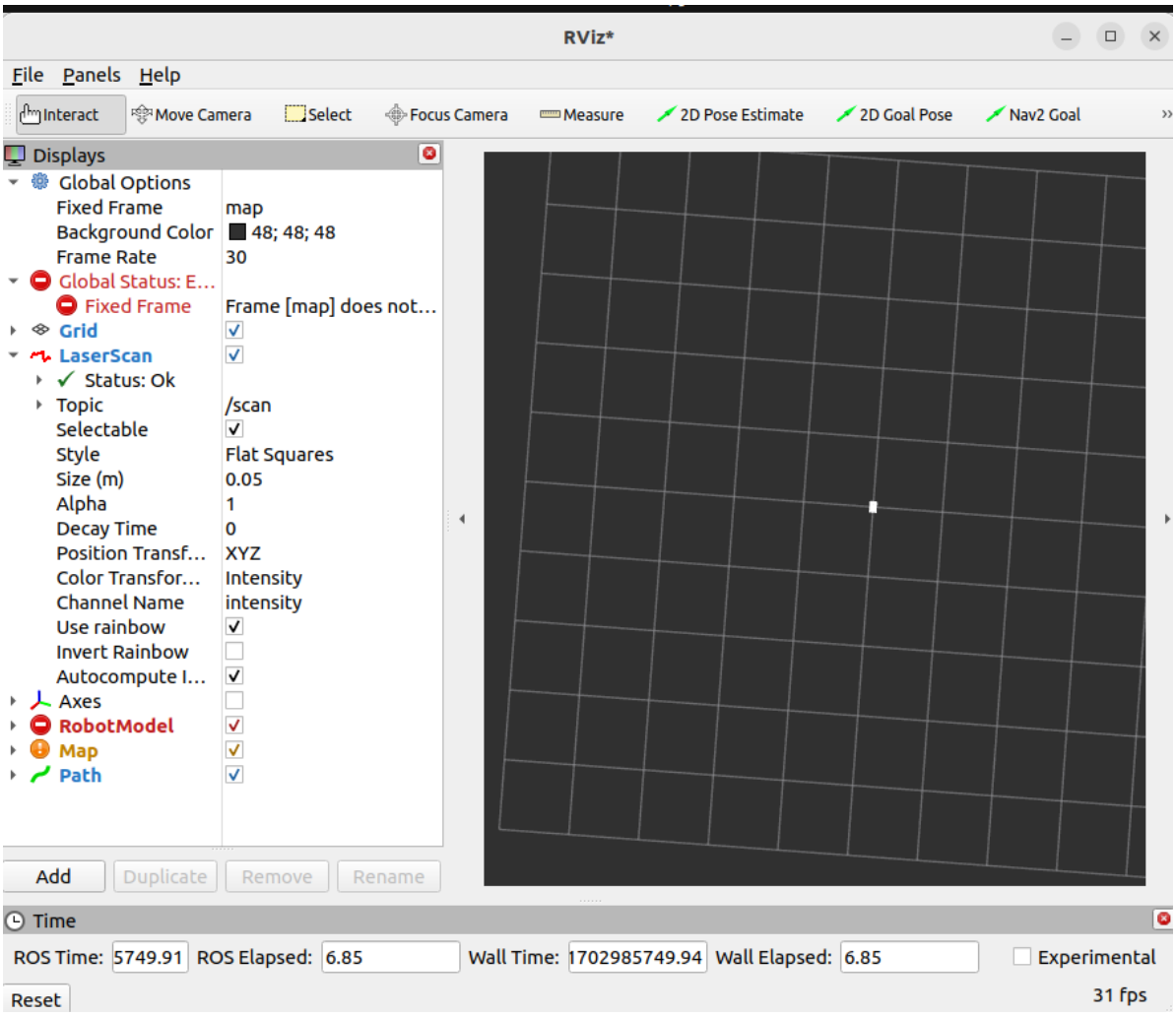
2、程序启动

加载URDF和生成一个模拟控制器，终端输入，

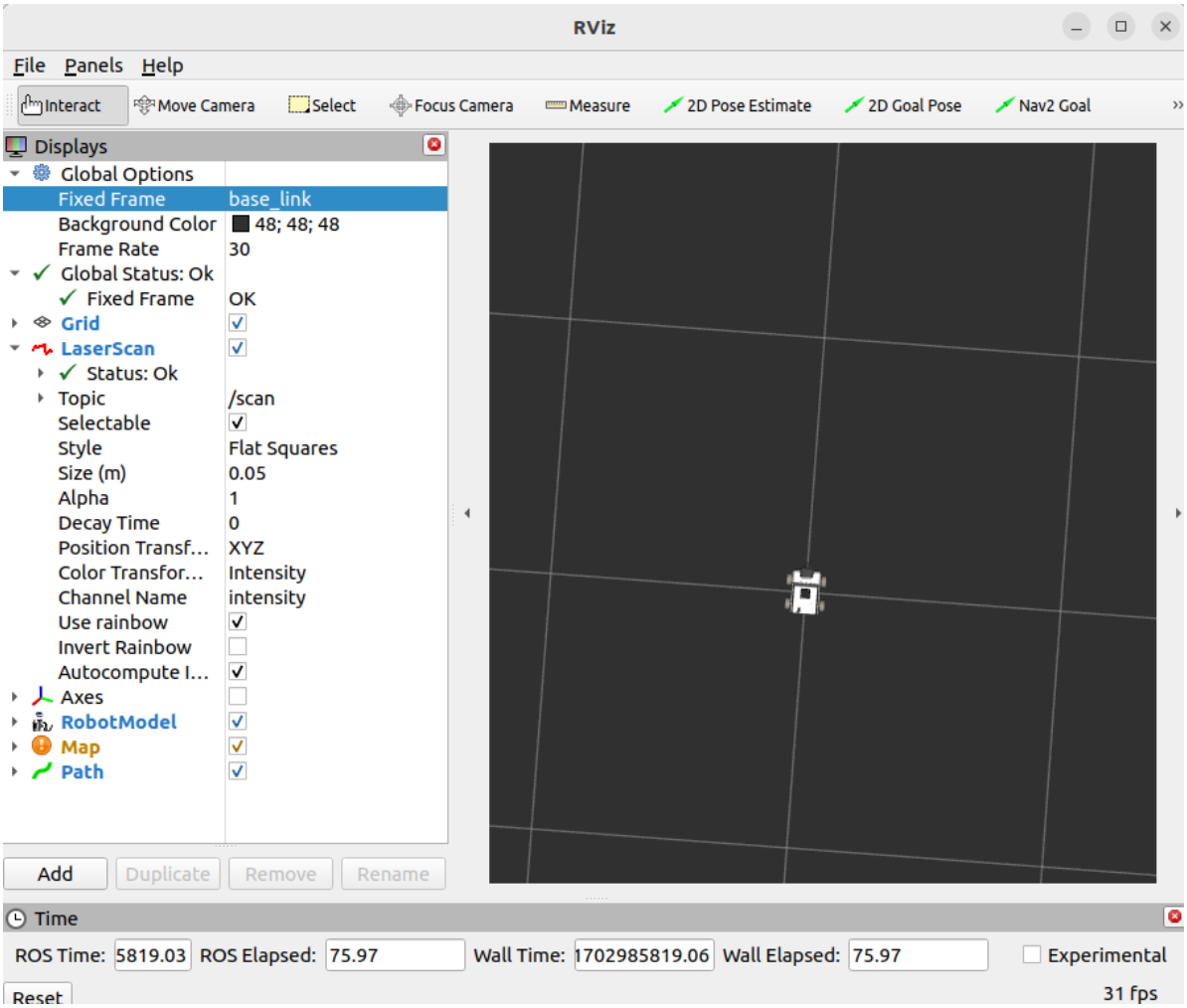
```
ros2 launch yahboomcar_description display_launch.py
```

打开rivz显示模型，终端输入，

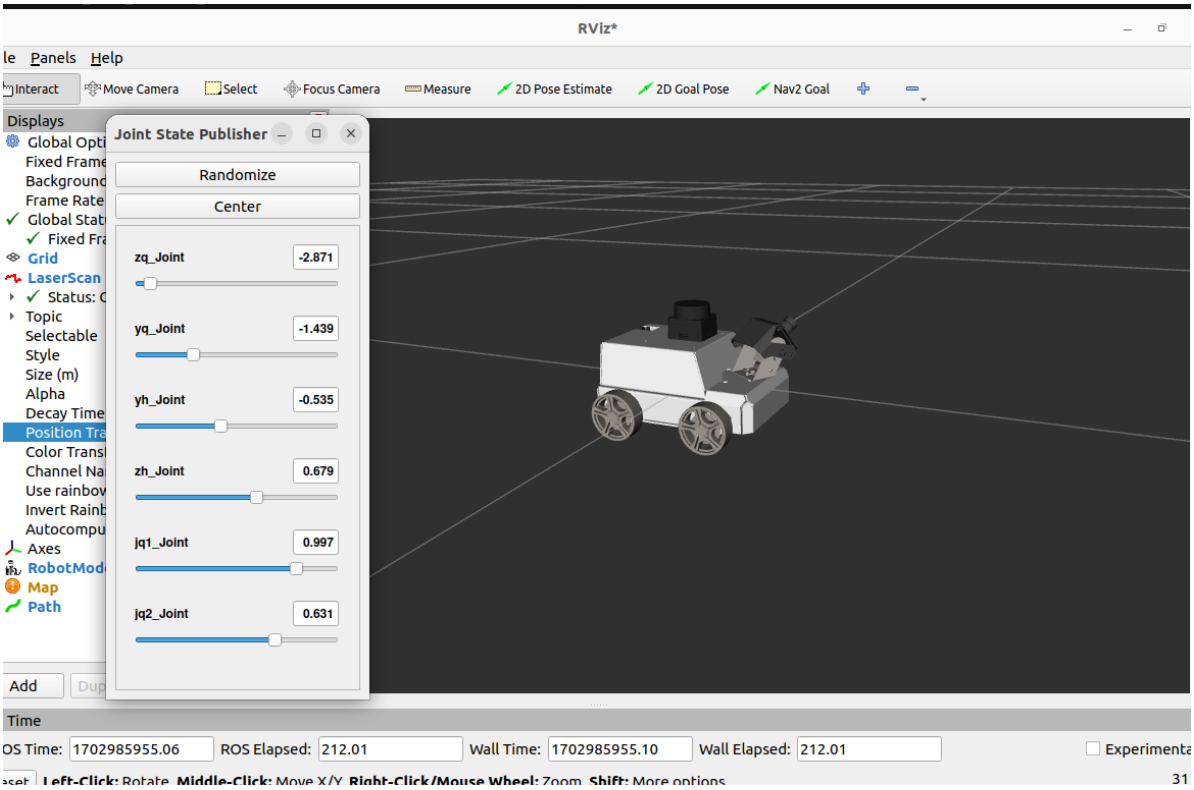
```
rviz2
```



将【Fixed Frame】改成base_linke，即可显示小车模型，



然后，用鼠标调整视角，滑动刚才生成的模拟控制器，即可看的小车的轮胎/相机在变化，



zq_Joint：控制左前轮

yq_Joint：控制右前轮

yh_Joint：控制右后轮

zh_Joint：控制左后轮

jq1_Joint：控制云台1

jq2_Joint：控制云台2

Randomize：随机发布数值到各个joint

Center：所有的Joint归中

3、代码解析

代码位置（以虚拟机为例），

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_description/launch
```

display_launch.py

```
from ament_index_python.packages import get_package_share_path

from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch.substitutions import Command, LaunchConfiguration

from launch_ros.actions import Node
from launch_ros.parameter_descriptions import ParameterValue

def generate_launch_description():
    urdf_tutorial_path = get_package_share_path('yahboomcar_description')
    default_model_path = urdf_tutorial_path / 'urdf/MicroROS.urdf'

    model_arg = DeclareLaunchArgument(name='model',
default_value=str(default_model_path),
                                      description='Absolute path to robot urdf
file')
    robot_description = ParameterValue(Command(['xacro ',
LaunchConfiguration('model')]),
                                       value_type=str)

    robot_state_publisher_node = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        parameters=[{'robot_description': robot_description}]
    )

    joint_state_publisher_gui_node = Node(
        package='joint_state_publisher_gui',
        executable='joint_state_publisher_gui'
    )


    tf_base_footprint_to_base_link = Node(
        package='tf2_ros',
        executable='static_transform_publisher',
        arguments=['0', '0', '0.05', '0.0', '0.0', '0.0', 'base_footprint',
'base_link'],
```

```
    )

    return LaunchDescription([
        model_arg,
        joint_state_publisher_gui_node,
        robot_state_publisher_node,
        tf_base_footprint_to_base_link
    ])
```

- model_arg：加载模型参数，加载的模型为MicroROS.urdf，位置在`/home/yahboom/yahboomcar_ws/src/yahboomcar_description/urdf`
- joint_state_publisher_gui_node：发布sensor_msgs/JointState消息
- robot_state_publisher_node：机器人状态发布
- tf_base_footprint_to_base_link：发布base_footprint到base_link的静态变换

4、URDF模型

URDF，全称为 Unified Robot Description Format ，翻译为中文为 统一机器人描述性格式，是一种使用xml格式描述的机器人模型文件，类似于D-H参数。

```
<?xml version="1.0" encoding="utf-8"?>
```

第一行为xml必填项，描述了xml的版本信息。

```
<robot name="name="micro4.0">
</robot>
```

第二行描述了当前的机器人名称；当前机器人所有信息均包含在【robot】标签内。

4.1、组成部分

- link，连杆，可以想象成是人的手臂
- joint，关节，可以想象成是人的手肘关节

link与joint的关系：两个link之间通过关节连接起来，想象成手臂有小臂(link)和大臂(link)通过肘关节(joint)连接。

4.1.1、link

1）、简介

在URDF描述性语言中，link是用来描述物理特性的，

- 描述视觉显示， 标签。
- 描述碰撞属性， 标签。
- 描述物理惯性， 标签不常用。

Links还可以描述连杆尺寸(size)\颜色(color)\形状(shape)\惯性矩阵(inertial matrix)\碰撞参数(collision properties)等，每个Link会成为一个坐标系。

2）、示例代码

```
<link
    name="base_link">
    <inertial>
```

```
      <origin
        xyz="-0.0038187037800903 -0.000532399212617988 -0.00668209865413972"
        rpy="0 0 0" />
      <mass
        value="0.222555109690442" />
      <inertia
        ixx="0.000160582675692647"
        ixy="-8.18530574494391E-07"
        ixz="-2.74575507729664E-06"
        iyy="0.000176217109527607"
        iyz="1.64721285063183E-07"
        izz="0.000302441932451338" />
    </inertial>
    <visual>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <geometry>
        <mesh
          filename="package://yahboomcar_description/meshes/base_link.STL" />
      </geometry>
      <material
        name="">
        <color
          rgba="1 1 1 1" />
      </material>
    </visual>
    <collision>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <geometry>
        <mesh
          filename="package://yahboomcar_description/meshes/base_link.STL" />
      </geometry>
    </collision>
  </link>
```

3）、标签介绍

- origin：描述的是位姿信息； xyz 属性描述的是在大环境中的坐标位置， rpy 属性描述的是自身的姿态。
- mess：描述的是link的质量。
- inertia：惯性参考系，由于转动惯性矩阵的对称性，只需要6个上三角元素 ixx, ixy, ixz, iyy, iyz, izz 作为属性。
- geometry：标签描述的是形状； mesh 属性主要的功能是去加载纹理文件的， filename 属性纹理路径的文件地址
- material：标签描述的是材质； name 属性为**必填项**，可以为空，可以重复 。通过【color】标签中的 rgba

  属性来描述红、绿、蓝、透明度，中间用空格分隔。

4.1.2、joints

1）、简介

描述两个关节之间的关系，运动位置和速度限制，运动学和动力学属性。关节类型有以下几种：

- fixed：固定关节。不允许运动，起连接作用。
- continuous：旋转关节。可以持续旋转，没有旋转角度的限制。
- revolute：旋转关节。类似于continuous，有旋转角度的限制。
- prismatic：滑动关节。沿某一轴线移动，有位置限制。
- floating：悬浮关节。具备六个自由度，3T3R。
- planar：平面关节。允许在平面正交上方平移或者旋转。

2）、示例代码

```
<joint
    name="zq_Joint"
    type="continuous">
    <origin
      xyz="0.0455 0.0675 -0.02125"
      rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="zq_Link" />
    <axis
      xyz="0 1 0" />
</joint>
```

在【joint】标签中 name 属性是 **必填项**，描述关节的名称,并且是唯一。在【joint】标签中 type 属性，对应填写六大关节类型。

3）、标签介绍

- origin：子标签,指的是旋转关节于 parent 所在坐标系的相对位置。
- parent，child：parent，child子标签代表的是两个要连接的link；parent是参照物，child围绕着praent旋转。
- axis：子标签表示child对应的link围绕哪一个轴（xyz）旋转和述绕固定轴的旋转量。
- limit：子标签主要是限制child的。 lower 属性和 upper 属性限制了旋转的弧度范围， effort 属性限制的是转动过程中的受力范围。(正负value值，单位为牛或N)， velocity 属性限制了转动时的速度，单位为米/秒或m/s。
- mimic：描述该关节与已有关节的关系。
- safety_controller：描述安全控制器参数。保护机器人关节的运动。