Robot URDF model

Note: The virtual machine needs to be in the same LAN as the car, and the ROS_DOMAIN_ID needs to be consistent. You can view [Expansion Board Parameter Configuration] to set the IP and ROS_DOMAIN_ID on the board.

1. Program function description

The car connects to the proxy, runs the program, and the URDF model will be displayed in rviz.

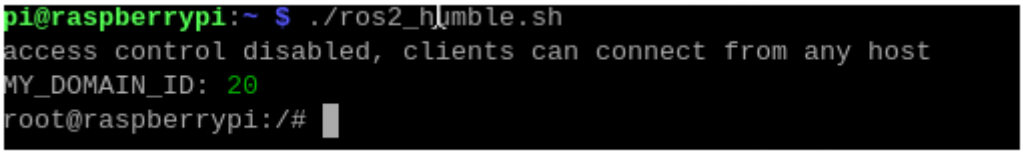
2. Program startup

2.1 Run command

If it is the desktop version of Raspberry Pi and the desktop version of Jetson Nano and X5, you need to enter Docker in advance and enter in the terminal,

```
sh ~/ros2_humble.sh
```

When the following interface appears, it means that you have successfully entered Docker,



Then enter in the Docker terminal separately, **(See the [How to enter the same Docker terminal] section)**

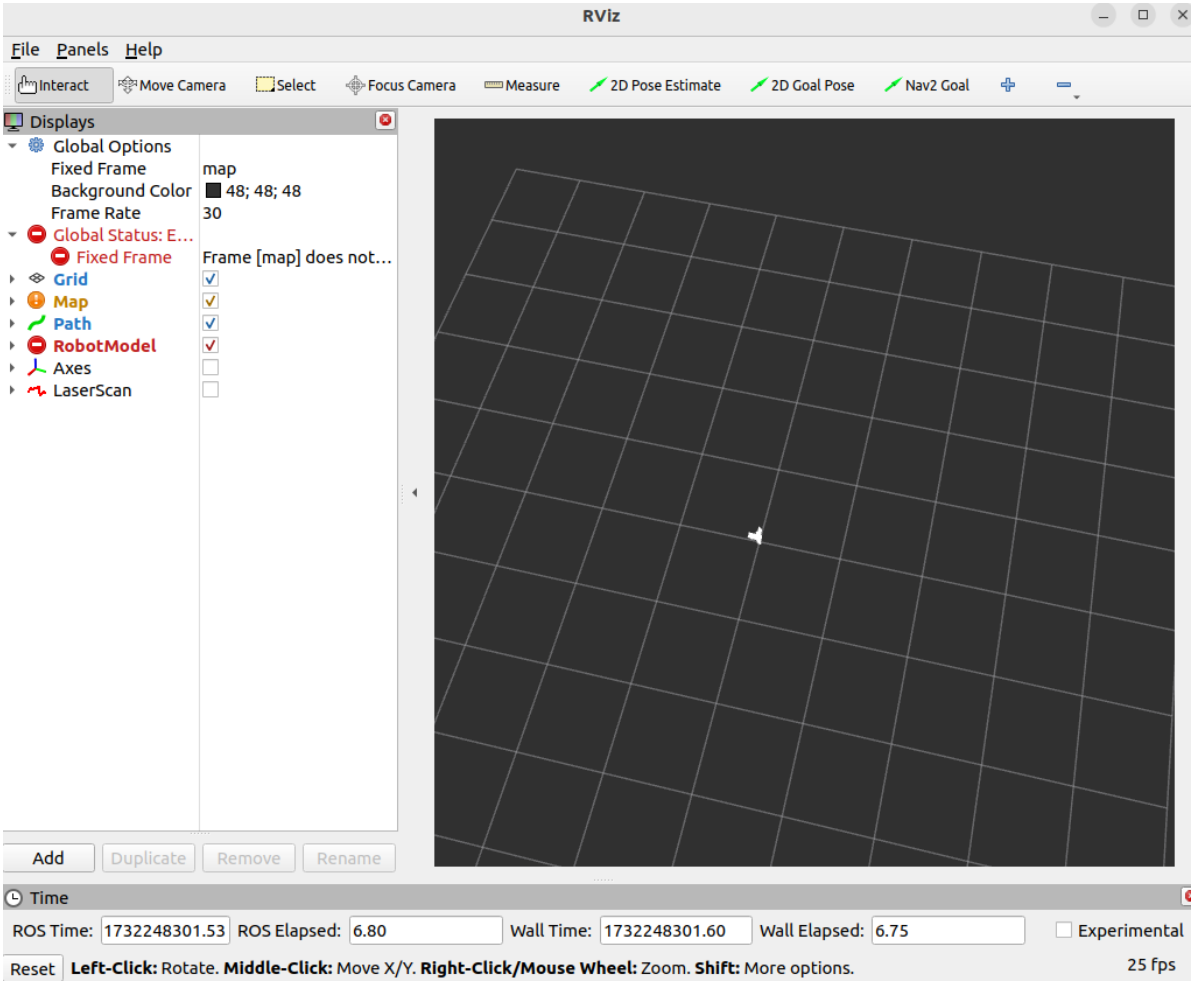
```
ros2 launch yahboomcar_description display_launch.py          #Generate a
controller
rviz2
```

Take the matching virtual machine as an example, load URDF and generate a simulation controller, enter in the terminal,

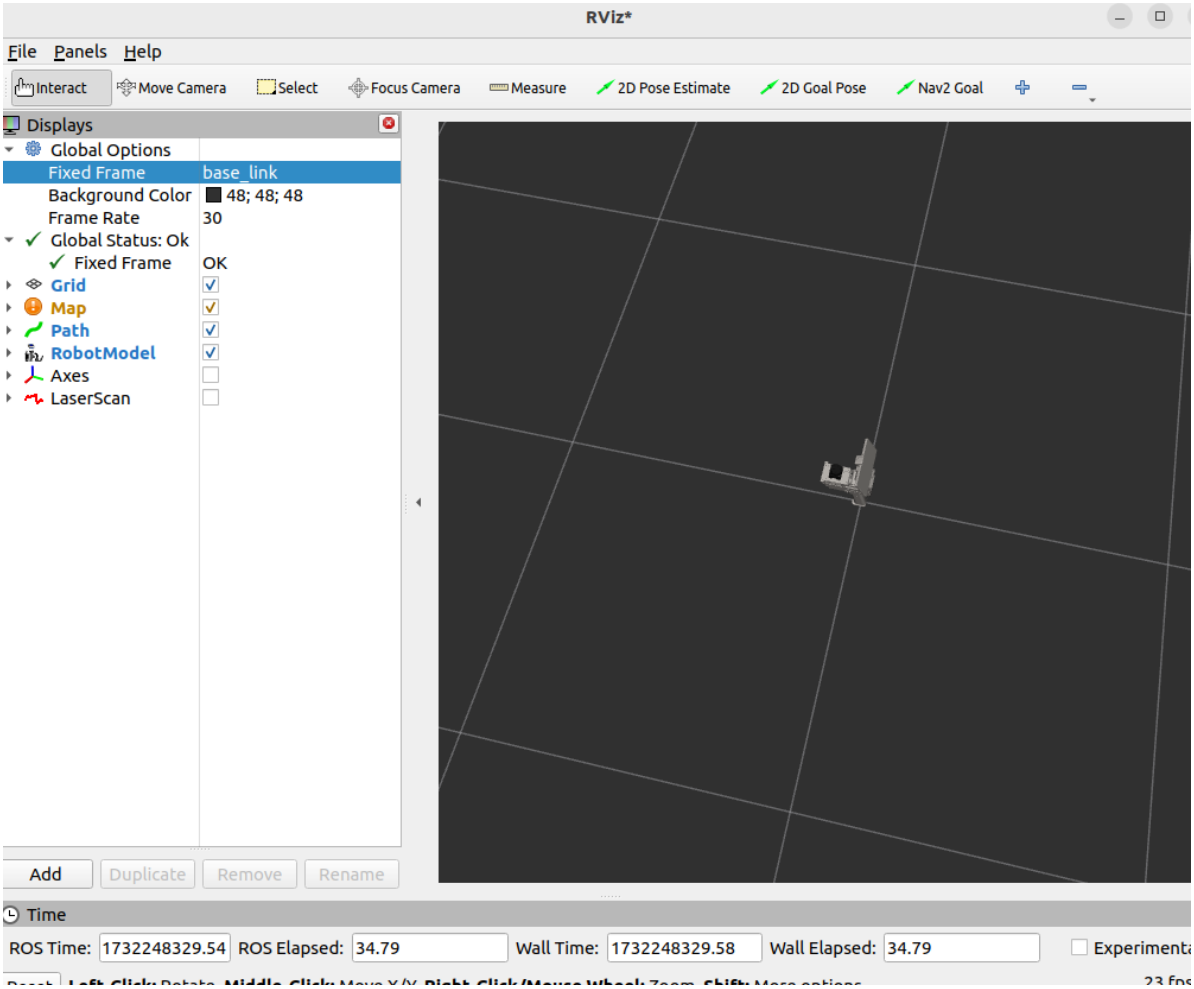
```
ros2 launch yahboomcar_description display_launch.py
```

Open rivz to display the model, input in the terminal,

```
rviz2
```



Change [Fixed Frame] to base_linke to display the car model,



Then you can use the mouse to adjust the viewing angle.

3. Code analysis

Code location (take virtual machine as an example),

```
/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/ShouChiLeiDa_V1.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',
        executable='joint_state_publisher'
    )


    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: load model parameters, the loaded model is ShouChiLeiDa_V1.urdf, the location is `/home/yahboom/yahboomcar_ws/src/yahboomcar_description/urdf`
- joint_state_publisher_gui_node: publish sensor_msgs/JointState message
- robot_state_publisher_node: robot state publishing
- tf_base_footprint_to_base_link: publish static transformation from base_footprint to base_link
-

4. URDF model

URDF, the full name is Unified Robot Description Format, translated into Chinese as Unified Robot Description Format, is a robot model file described in XML format, similar to D-H parameters.

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

The first line is a required XML field, describing the XML version information.

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

The second line describes the current robot name; all information about the current robot is included in the [robot] tag.

4.1. Components

- Link, connecting rod, can be imagined as a human arm
- Joint, joint, can be imagined as a human elbow joint

Relationship between link and joint: two links are connected by joints, imagine that the arm has a small arm (link) and a large arm (link) connected by an elbow joint (joint).

4.1.1. Link

1), Introduction

In the URDF descriptive language, link is used to describe physical properties,

- Describe visual display, label.
- Describe collision properties, label.
- Describe physical inertia, label is not commonly used.

Links can also describe the size of the connecting rod (size)\color (color)\shape (shape)\inertial matrix (inertial matrix)\collision parameters (collision properties), etc. Each Link will become a coordinate system.

2), Sample code

```
<link
    name="base_link">
    <inertial>
      <origin
        xyz="0.0157964627714118 0.000203656131254087 0.174023147288787"
        rpy="0 0 0" />
      <mass
        value="0.314338879377736" />
      <inertia
```

```
        ixx="0.000381597601327499"
        ixy="-2.50848479375714E-07"
        ixz="-3.69141941471681E-05"
        iyy="0.000306254045050102"
        iyz="-1.87740203934869E-06"
        izz="0.000441541214660635" />
    </inertial>
    <visual>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <geometry>
        <mesh
          filename="package://yahboomcar_description/meshes/hand_base_link.STL"
/>
      </geometry>
      <material
        name="">
        <color
          rgba="0.776470588235294 0.756862745098039 0.737254901960784 1" />
      </material>
    </visual>
    <collision>
      <origin
        xyz="0 0 0"
        rpy="0 0 0" />
      <geometry>
        <mesh
          filename="package://yahboomcar_description/meshes/hand_base_link.STL"
/>
      </geometry>
    </collision>
  </link>
```

3) Tag introduction

- origin: describes the pose information; the xyz attribute describes the coordinate position in the environment, and the rpy attribute describes its own posture.
- mess: describes the quality of the link.
- inertia: inertial reference system. Due to the symmetry of the rotational inertia matrix, only 6 upper triangular elements ixx, ixy, ixz, iyy, iyz, izz are required as attributes.
- geometry: the tag describes the shape; the main function of the mesh attribute is to load the texture file, and the filename attribute is the file address of the texture path
- material: the tag describes the material; the name attribute is a **required item**, which can be empty and repeated. The rgba attribute in the [color] tag is used to describe red, green, blue, and transparency, separated by spaces.

4.1.2, joints

1) Introduction

Describes the relationship between two joints, motion position and speed limits, kinematic and dynamic properties. There are several types of joints:

- fixed: fixed joints. No movement is allowed, it serves as a connection.
- continuous: revolute joint. It can rotate continuously, without rotation angle limit.
- revolute: revolute joint. Similar to continuous, with rotation angle limit.

- prismatic: sliding joint. It moves along a certain axis, with position limit.
- floating: floating joint. It has six degrees of freedom, 3T3R.
- planar: planar joint. It allows translation or rotation above the plane orthogonal to the plane.

2), sample code

```
<joint
    name="leida_Joint"
    type="fixed">
    <origin
      xyz="0.088935 0 0.19696"
      rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="radar_Link" />
    <axis
      xyz="0 0 0" />
  </joint>
```

In the [joint] tag, the name attribute is a **required item**, describing the name of the joint, and it is unique. In the [joint] tag, the type attribute is filled in with the six major joint types.

3), Tag introduction

- origin: sub-tag, refers to the relative position of the rotation joint in the parent coordinate system.
- parent, child: parent, child sub-tags represent two links to be connected; parent is a reference object, and child rotates around the parent.
- axis: sub-tag indicates which axis (xyz) the link corresponding to the child rotates around and the amount of rotation around the fixed axis.
- limit: sub-tag is mainly used to limit the child. The lower attribute and upper attribute limit the range of rotation, the effort attribute limits the force range during the rotation process. (positive and negative value, in Newton or N), the velocity attribute limits the speed of rotation, in meters/second or m/s.
- mimic: describes the relationship between this joint and the existing joints.
- safety_controller: describes the safety controller parameters. Protect the movement of the robot joints.