

Brief introduction and use of rviz2

rviz is a three-dimensional visualization platform in ROS. On the one hand, it can realize the graphical display of external information, and on the other hand, it can also release control information to the object through rviz, so as to realize the monitoring and control of the robot.

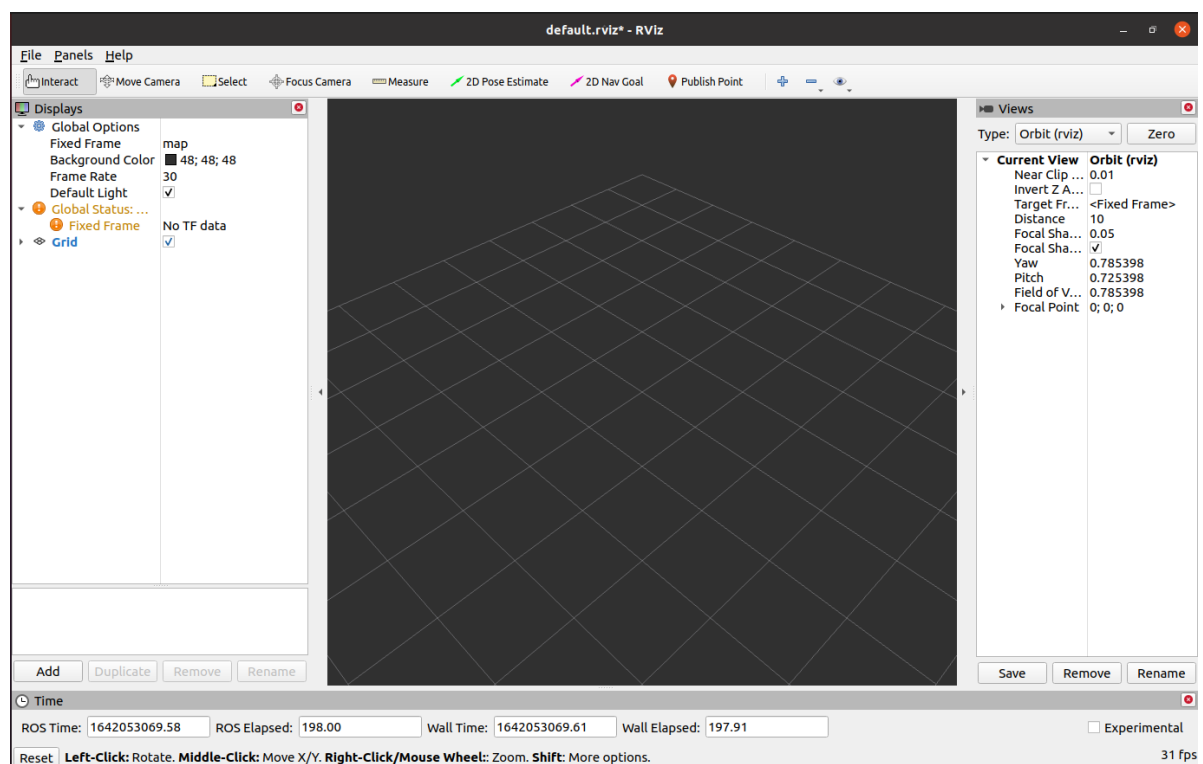
1 Introduction to rviz2

The successful installation of ros2 indicates that rviz2 is also successfully installed together, because the installation of ros2 includes rviz2.

Open a new terminal (shortcutCtrl+Alt+T)enter the command to open rviz2

```
rviz2
```

Open rviz2 and display the following interface:



2 Introduction of each area

- On the left is the list of monitors, a monitor is something that draws something in the 3D world and may have some options available in the display list.
- Above is the toolbar, which allows users to use various function buttons to select tools with multiple functions
- The middle part is the 3D view: it is the main screen where various data can be viewed in 3D. The background color, fixed frame, grid, etc. of the 3D view can be set in detail in the Global Options and Grid items displayed on the left.
- Below is the time display area, including system time and ROS time.
- The right side is the observation angle setting area, and different observation angles can be set.

We only give a rough introduction in this part. If you want to know more detailed content, you can go to the [user guide](#) to view it.

3 mycobot_ros2 installation and update

- **M5 version:** Please refer to the end of the [Installation of ROS2]chapter.

Launch via launch.py file

This example is based on the fact that you have completed the [environment setup](#) and successfully copied the company's code from GitHub..

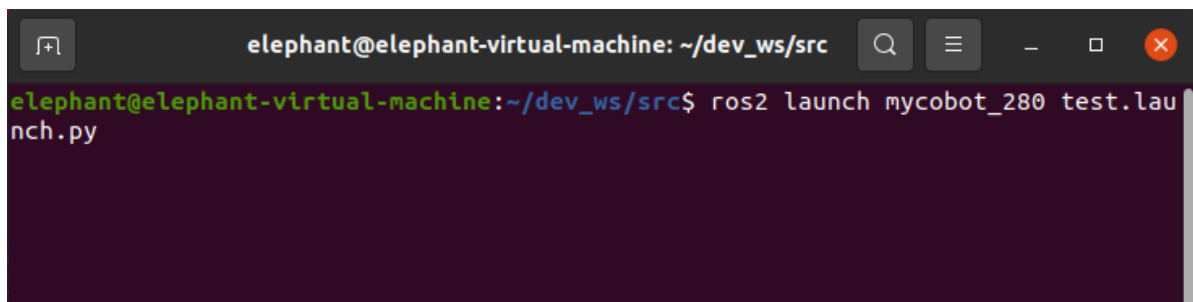
Open a console terminal (shortcutCtrl+Alt+T) Enter the following command to **configure the ROS2 environment** .

```
cd ~/colcon_ws
colcon build --symlink-install
source install/setup.bash
```

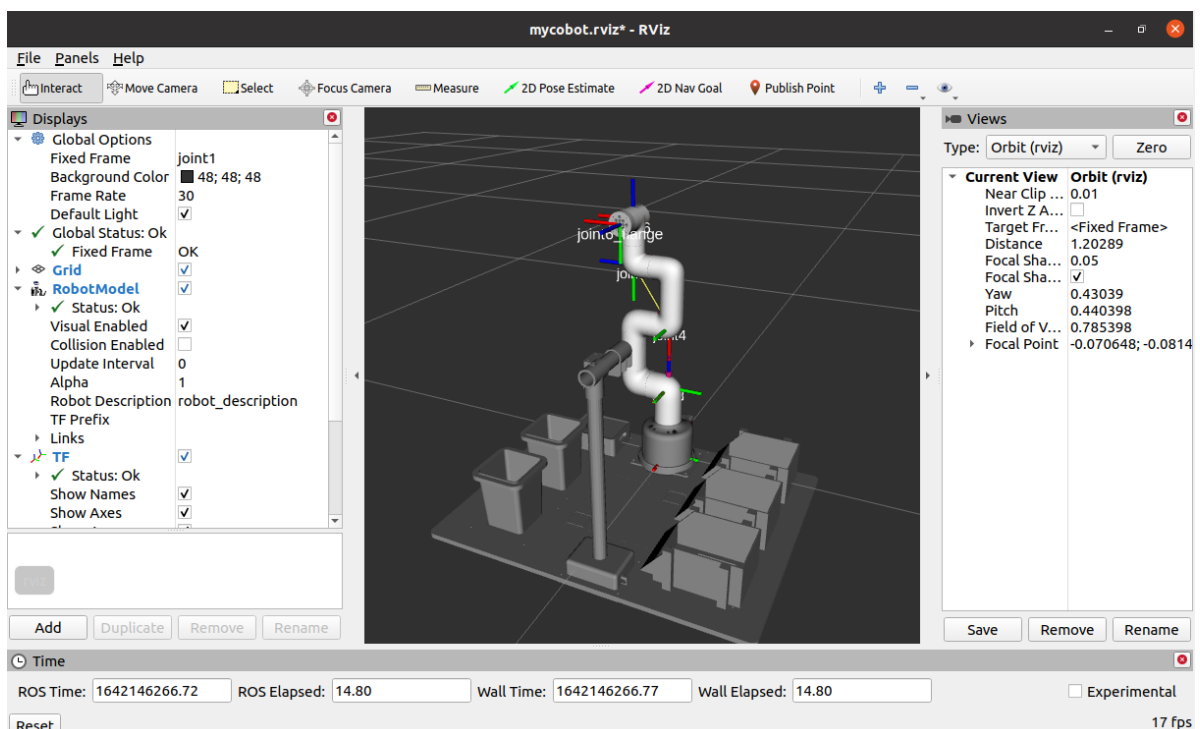
Enter again:

- mycobot 280-M5 version:

```
ros2 launch mycobot_280 test.launch.py
```



Open rviz and get the following result:



If you want to know more information about rviz, you can go to the [official documentation](#) to view it.

5 M5 Version Prerequisites

- Type Ctrl+Shift+T in the command terminal to open another terminal window in the same directory to view the device name:

```
# View the device name of the robotic arm
ls /dev/ttyUSB* # old version myCobot280 M5

# If the terminal does not display the /dev/ttyUSB related name, you need to use
the following command
ls /dev/ttyACM* # new version myCobot280 M5
```

- Grant the serial port permission to the robotic arm:

```
# The default device name is /dev/ttyUSB0, if the device name is not the default
value, it needs to be modified.
sudo chmod 777 /dev/ttyUSB0 # old version myCobot280 M5

sudo chmod 777 /dev/ttyACM0 # new version myCobot280 M5
```

Then enter the user password(**Note:** The password is not displayed, just enter it correctly).

Control and follow-up of the robotic arm

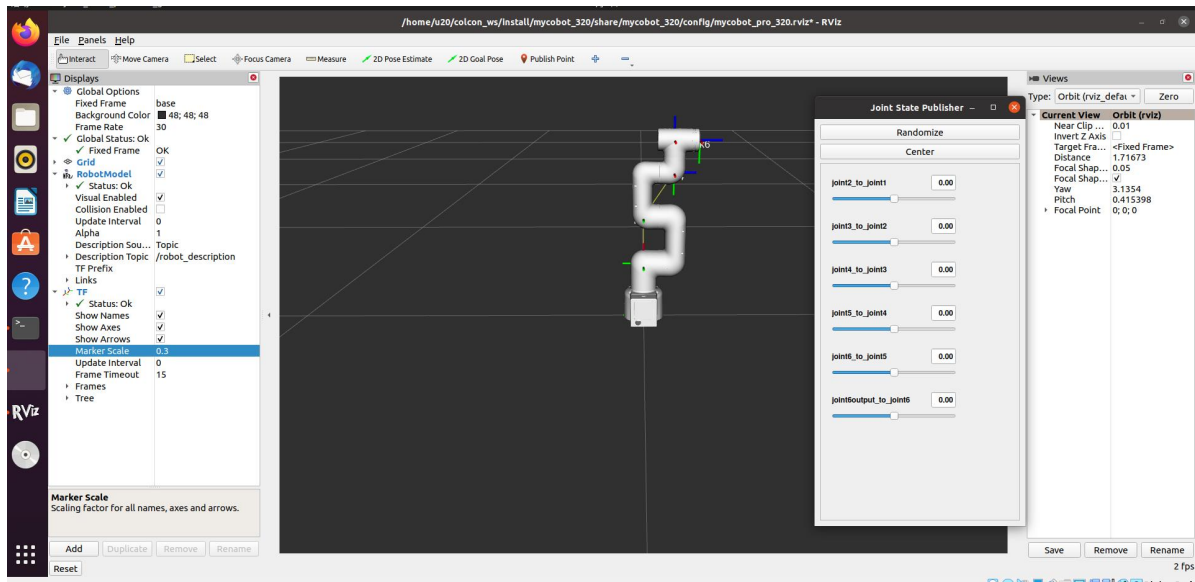
1 Slider Control

Open a command line and run:

- mycobot 320-M5 2022 version:

```
# The default serial port name of mycobot 320-M5 version is "/dev/ttyUSB0", and
the baud rate is 115200. The serial port name of some models is "dev/ttyACM0". If
the default serial port name is wrong, you can change the serial port name to "
/dev/ttyACM0".
ros2 launch mycobot_320 slider_control1.launch.py
```

It will **open rviz and a slider component** and you will see something like this:



Then you can **control the movement of the model in rviz by dragging the slider** . The real mycobot will move with it.

Please note: Since the robot arm will move to the current position of the model when the command is input, please make sure that the model in rviz does not appear to be worn out before you use the command Do not quickly drag the slider after connecting the robotic arm to prevent damage to the robotic arm

2 Model Follow

In addition to the above controls, we can also **make the model follow the movement of the real robotic arm** . Open a command line and run:

- mycobot 320-M5 2022 version:

```
# The default serial port name of mycobot 320-M5 version is "/dev/ttyUSB0", and
the baud rate is 115200. The serial port name of some models is "dev/ttyACM0". If
the default serial port name is wrong, you can change the serial port name to "
/dev/ttyACM0".
ros2 launch mycobot_320 mycobot_follow.launch.py
```

It will **open rviz to show the model following effect** .

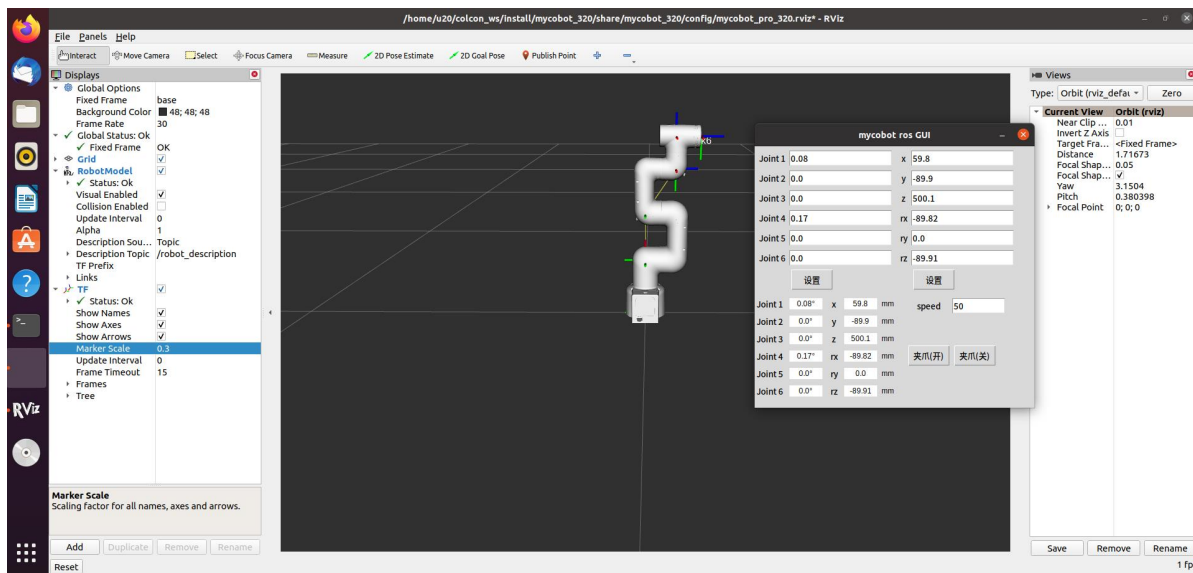
3 GUI Control

On the basis of the previous, this package also **provides a simple GUI control interface** . This method means that the real robotic arms are linked with each other, please connect to mycobot.

Open the command line::

- mycobot 320-M5 2022 version:

```
# The default serial port name of mycobot 320-M5 version is "/dev/ttyUSB0", and
the baud rate is 115200. The serial port name of some models is "dev/ttyACM0". If
the default serial port name is wrong, you can change the serial port name to "
/dev/ttyACM0".
ros2 launch mycobot_320 simple_gui.launch.py
```



4 Keyboard Control

The function of keyboard control is added in the package of `mycobot_320`, and it is synchronized in real time in rviz. This function relies on pythonApi, so make sure to connect with the real robotic arm.

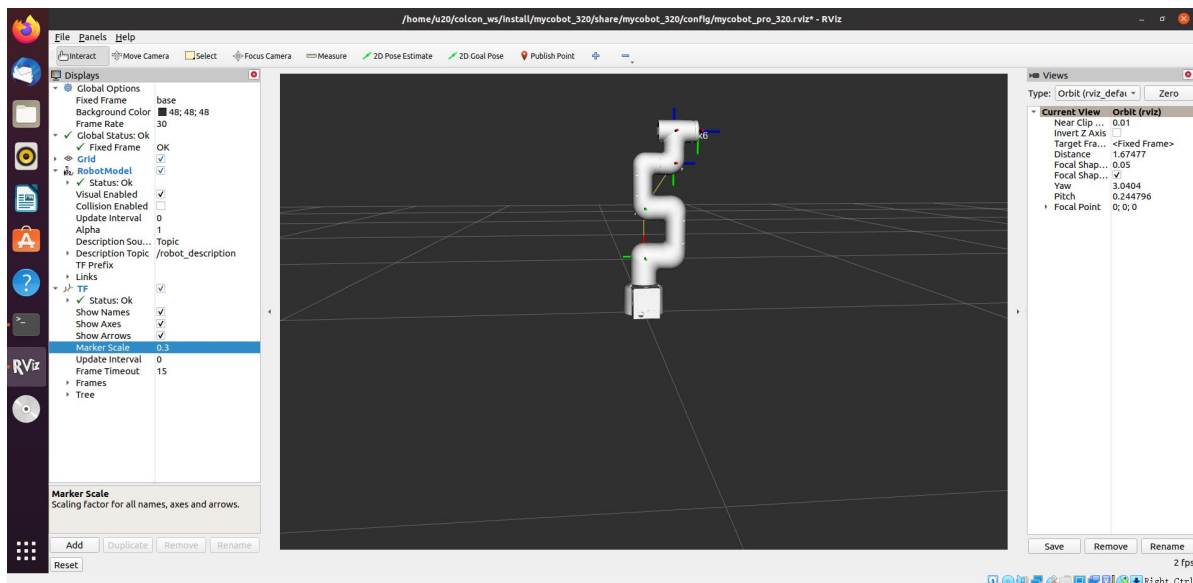
Open a command line and run:

- mycobot 320-M5 2022 version:

The default serial port name of mycobot 320-M5 version is `"/dev/ttyUSB0"`, and the baud rate is 115200. The serial port name of some models is `"/dev/ttyACM0"`. If the default serial port name is wrong, you can change the serial port name to `"/dev/ttyACM0"`.

```
ros2 launch mycobot_320 teleop_keyboard.launch.py
```

The operation effect is as follows:



The command line will output mycobot information, as follows:

```
[INFO] [launch]: All log files can be found below /home/u20/.ros/log/2022-08-05-10-43-12-301971-u20-virtualBox-6290
[INFO] [launch]: Default logging verbosity is set to INFO
```

```
[INFO] [robot_state_publisher-1]: process started with pid [6293]
[INFO] [rviz2-2]: process started with pid [6295]
[INFO] [follow_display-3]: process started with pid [6297]
[robot_state_publisher-1] Parsing robot urdf xml string.
[robot_state_publisher-1] Link link1 had 1 children
[robot_state_publisher-1] Link link2 had 1 children
[robot_state_publisher-1] Link link3 had 1 children
[robot_state_publisher-1] Link link4 had 1 children
[robot_state_publisher-1] Link link5 had 1 children
[robot_state_publisher-1] Link link6 had 0 children
[robot_state_publisher-1] [INFO] [1659667392.703132973] [robot_state_publisher]:
got segment base
[robot_state_publisher-1] [INFO] [1659667392.703485410] [robot_state_publisher]:
got segment link1
[robot_state_publisher-1] [INFO] [1659667392.703545198] [robot_state_publisher]:
got segment link2
[robot_state_publisher-1] [INFO] [1659667392.703571119] [robot_state_publisher]:
got segment link3
[robot_state_publisher-1] [INFO] [1659667392.703587512] [robot_state_publisher]:
got segment link4
[robot_state_publisher-1] [INFO] [1659667392.703603744] [robot_state_publisher]:
got segment link5
[robot_state_publisher-1] [INFO] [1659667392.703619685] [robot_state_publisher]:
got segment link6
[rviz2-2] [INFO] [1659667393.588026632] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-2] [INFO] [1659667393.588472253] [rviz2]: OpenGL version: 3.1 (GLSL 1.4)
[rviz2-2] [INFO] [1659667393.766777360] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-2] Parsing robot urdf xml string.
[follow_display-3] [INFO] [1659667394.310152595] [follow_display]:
port:/dev/ttyUSB0, baud:115200
```

Next, open another command line and run:

- mycobot 320-M5 2022 version:

```
ros2 run mycobot_320 teleop_keyboard
```

You will see the following output on the command line:

```
Mycobot Teleop Keyboard Controller
-----
Moving options(control coordinations [x,y,z,rx,ry,rz]):
    w(x+)

    a(y-)    s(x-)    d(y+)

    z(z-) x(z+)

u(rx+)  i(ry+)  o(rz+)
j(rx-)  k(ry-)  l(rz-)

Gripper control:
  g - open
  h - close
```

Other:

- 1 - Go to init pose
- 2 - Go to home pose
- 3 - Resave home pose
- q - Quit

currently: speed: 30 change percent: 2

In this terminal, you can control the state of the robot arm and move the robot arm through the keys in the command line.