



睿尔曼机械臂 ROS2 使用说明书 V1.0



睿尔曼智能科技（北京）有限公司



文件修订记录：

| 版本号 | 时间 | 备注 |
|------|------------|----|
| V1.0 | 2023-11-27 | 拟制 |
| | | |
| | | |



目录

| | |
|---------------------------------|----|
| 1.package 概览 | 3 |
| 2. rm_driver 功能包介绍..... | 4 |
| 3. rm_description 功能包介绍 | 4 |
| 4. rm_ros_interfaces 功能包介绍..... | 6 |
| 5. rm_control 功能包介绍 | 6 |
| 6. rm_moveit2_config 文件介绍 | 7 |
| 7. rm_bringup 功能包介绍 | 8 |
| 8. rm_gazebo 功能包介绍 | 9 |
| 9. rm_example 功能包介绍 | 10 |
| 10. rm_doc 功能包介绍 | 12 |
| 11. rm_install 功能包介绍 | 12 |



1.package 概览

该文档将总体介绍机械臂基于 ROS2 的功能包，主要作用为帮助大家实现两个目的。

1. 了解 ROS2 的功能包作用。
2. 掌握当前 ROS 包的使用方式。

源码地址：https://github.com/RealManRobot/ros2_rm_robot.git。

下面为功能包的总体介绍。

1. 安装与环境配置(rm_install)
2. 硬件驱动(rm_driver)
3. 启动(rm_bringup)
4. 模型描述(rm_description)
5. ROS 消息接口(rm_ros_interfaces)
6. Moveit2 配置(rm_moveit_config)
7. Moveit2 与硬件驱动通信连接(rm_config)
8. Gazebo 仿真机械臂控制(rm_gazebo)
9. 使用案例(rm_examples)
10. 技术文档(rm_docs)



2. rm_driver 功能包介绍

2.1 rm_driver 功能包作用

(1)与机械臂通过 API 函数建立连接 ,机器人默认 IP 地址 :192.168.1.18 ,请保证上位机的 IP 在同一局域网内 ,使用 ROS2 控制机械臂 ,请务必确认机械臂处于以太网口通信模式 ;

(2)机械臂的底层驱动程序 ,订阅和发布各 topic 数据 ,更新 RVIZ 内机械臂各关节角度。

2.2 rm_driver 功能包使用方式

实际使用该节点时需要在终端中输入以下指令。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号 , 可选择的机械臂型号有 65、63、eco65、75。

例如使用的机械臂为 65 系列 , 则使用如下指令启动该节点。

```
ros2 launch rm_driver rm_65_driver.launch.py
```

关于该功能包更详细的介绍请参考该功能包的详细说明文档。

3. rm_description 功能包介绍

3.1 rm_description 功能包作用

(1)RM65-B、RM63-B、RMeco65-B、RM75-B 机器人描述功能包 , 提供创建好的机器人模型和配置文件。

(2)通过 rm_65.urdf、rml_63.urdf、rm_eco65.urdf、rm_75.urdf 机器人模型文件提供规划好的 TF 变换。

3.2 rm_description 功能包使用方式

实际使用该节点时需要在终端中输入以下指令。



```
ros2 launch rm_description rm_<arm_type>_display.launch.py
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

例如使用的机械臂为 65 系列，则使用如下指令启动该节点。

```
ros2 launch rm_description rm_65_display.launch.py
```

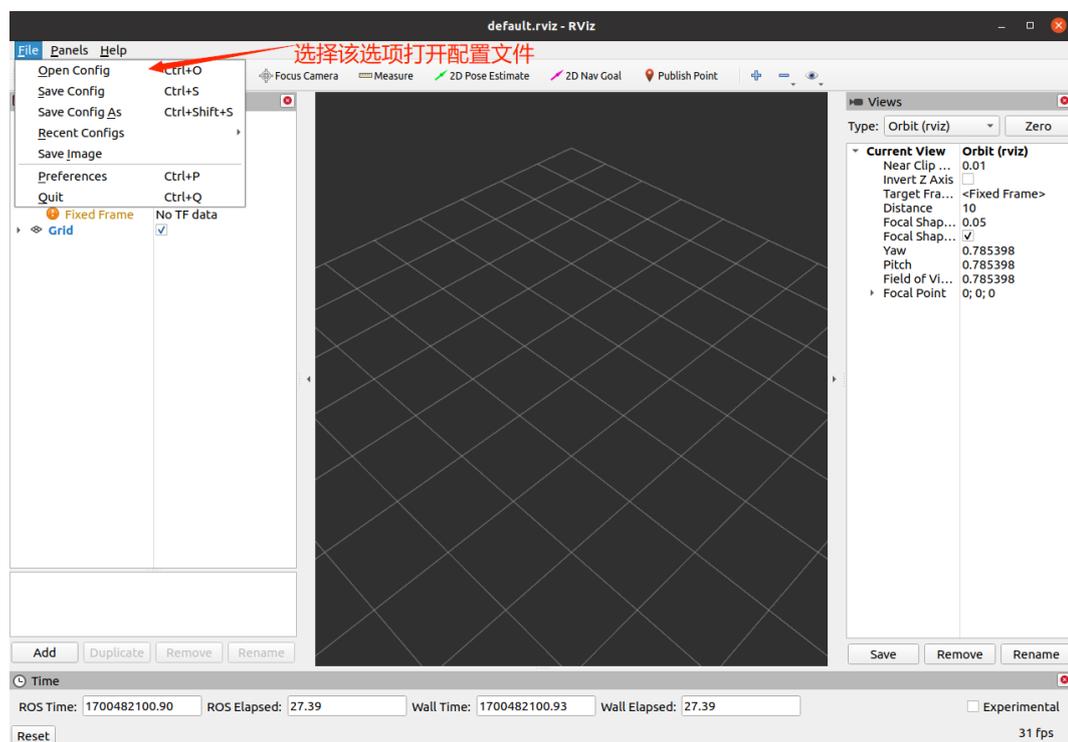
启动成功后继续启动 rm_driver 节点。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py
```

之后启动 rviz2 节点。

```
rviz2
```

可以调用之前设计好的 rviz 配置文件直接加载模型。



在功能包的 rviz 文件夹下找到对应的配置文件。



我们可以在 rviz2 的界面中调出机械臂模型进行查看，且机械臂的位姿与实际机械臂相同。

4. rm_ros_interfaces 功能包介绍

4.1 rm_ros_interfaces 功能包作用

(1) RM65、RM63、RMECO65、RM75 所用到的所有控制消息和状态消息。

4.2 rm_ros_interfaces 功能包使用方式

该功能包主要为其它功能包提供启动运行需要的消息文件，并不执行任何指令。

5. rm_control 功能包介绍

5.1 rm_control 功能包作用

(1) 提供一个与 moveit2 对接的动作，为 moveit2 实现真实机械臂的控制提供通信桥梁。

(2) 机器人控制器，将 Moveit 规划的机械臂轨迹，通过三次样条插值细分，按照 20ms 的控制周期发给 rm_driver 节点。

5.2 rm_control 功能包使用方式

实际使用该节点时需要在终端中输入以下指令。

```
ros2 launch rm_control rm_<arm_type>_control.launch.py
```



该节点本身并不实现具体可见的功能，需要配合 `rm_moveit2_config` 功能包的相关功能进行使用。

6. `rm_moveit2_config` 文件介绍

6.1 `rm_moveit2_config` 文件作用

(1) 使用 Setup Assistant 工具根据机械臂 URDF 模型 `rm_65.urdf` 创建生成的一个 Moveit2 配置的功能包，它包含了大部分 Moveit2 启动所需的配置文件和启动文件，以及包含一个简单的演示 demo。

(2) 根据机械臂特性对演示 demo 进行改写，配合 `rm_control` 和 `rm_driver` 功能包实现真实机械臂的控制功能。

6.2 `rm_moveit2_config` 文件使用方式

该功能包可实现虚拟机械臂的规划控制演示，通过以下指令进行实现。

```
ros2 launch rm_<arm_type>_config demo.launch.py
```

在实际使用时需要将以上的 `<arm_type>` 更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

例如使用的机械臂为 65 系列，则使用如下指令启动该节点。

```
ros2 launch rm_65_config demo.launch.py
```

启动成功后可以通过拖动的方式控制机械臂。

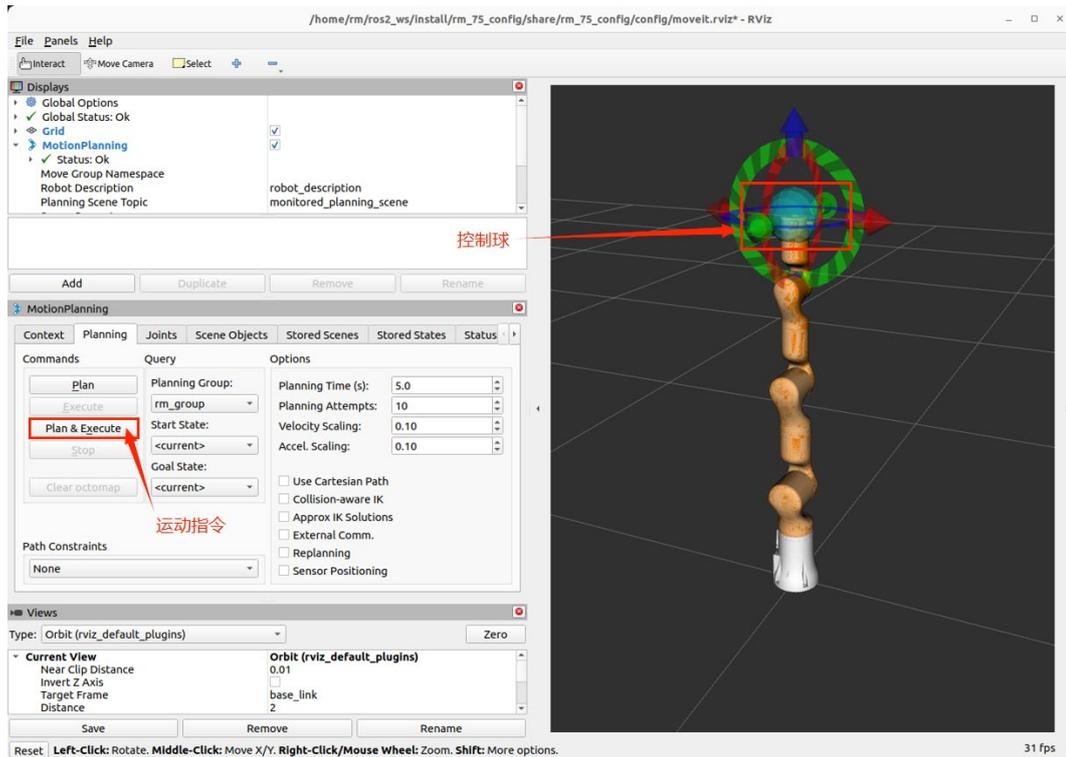
该功能包可实现真实机械臂的规划控制，通过以下指令进行实现。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py  
  
ros2 launch rm_description rm_<arm_type>_display.launch.py  
  
ros2 launch rm_control rm_<arm_type>_control.launch.py  
  
ros2 launch rm_<arm_type>_config real_moveit_demo.launch.py
```

在 `rviz2` 中会出现机械臂 `moveit2` 规划的相关信息，我们可以通过拖动控制



球的方式控制真实机械臂。



7. rm_bringup 功能包介绍

7.1 rm_bringup 功能包作用

(1) 实现机械臂复杂启动指令的集成，可通过一个 launch 文件，实现多节点配合的复杂功能。

7.2 rm_bringup 功能包使用方式

实际使用该节点时需要在终端中输入以下指令，直接启动 moveit2 实现 moveit2 对真实机械臂的控制。

```
ros2 launch rm_bringup rm_<arm_type>_bringup.launch.py
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

例如使用的机械臂为 65 系列，则使用如下指令启动该节点。

```
ros2 launch rm_bringup rm_65_bringup.launch.py
```



实际使用该节点时需要在终端中输入以下指令，直接启动 gazebo 实现 moveit2 对仿真机械臂的控制。

```
ros2 launch rm_bringup rm_<arm_type>_gazebo.launch.py
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

例如使用的机械臂为 65 系列，则使用如下指令启动该节点。

```
ros2 launch rm_bringup rm_65_gazebo.launch.py
```

有关该功能包更详细的介绍请参考《ROS2 机械臂 rm_bringup 功能包详解》。

8. rm_gazebo 功能包介绍

8.1 rm_gazebo 功能包作用

(1) 实现仿真机械臂控制，在没有机械臂实体，而且想要使用 Moveit2 进行机械臂规划并查看机械臂运动轨迹时可以使用。

8.2 rm_gazebo 功能包使用方式

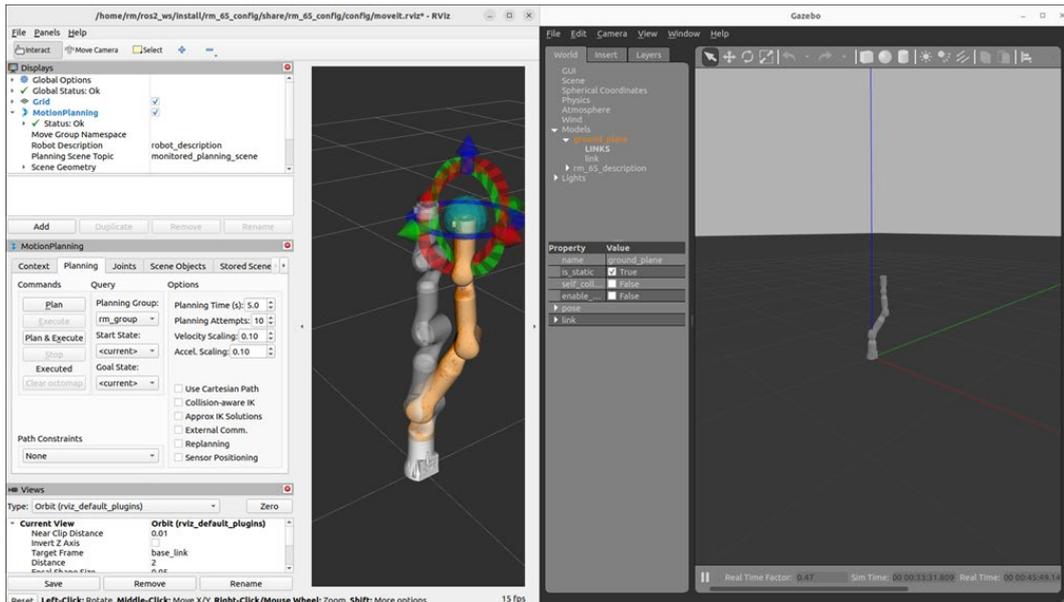
实际使用该节点时需要在终端中输入以下指令，启动 gazebo 仿真和 moveit2 实现 moveit2 对仿真机械臂的控制，指令如下。

```
ros2 launch rm_gazebo gazebo_<arm_type>_demo.launch.py
```

```
ros2 launch rm_<arm_type>_config gazebo_moveit_demo.launch.py
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

启动成功后，界面将出现如下画面。



关于该功能包更详细的介绍请参考该功能包的详细说明文档。

9. rm_example 功能包介绍

9.1 rm_example 功能包作用

(1) 实现机械臂一些常用功能的使用和代码开发，主要包含更换工作坐标系、获得机械臂当前状态、moveJ 角度规划运动、moveJP 位姿规划运动、moveL 直线运动。

9.2 rm_example 功能包使用方式

1. 更换工作坐标系。

更换机械臂当前的工作坐标系，需要执行以下两条指令实现该功能。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py
ros2 run rm_example rm_change_work_frame
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

运行成功后，终端将出现如下语句。

```
[INFO] [1701416861.664450164] [changeframe]: *****Switching the tool coordinate system succeeded
```



2. 得到当前的机械臂状态信息。

获得机械臂当前状态，需要执行以下两条指令实现该功能。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py  
  
ros2 run rm_example rm_get_state
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

运行成功后，终端将出现如下语句。

```
[INFO] [1701417669.964643809] [get_state]: joint state is: [-0.001000, 0.002000, 0.000000, 0.000000, 0.000000, -0.001000, -0.001000]  
[INFO] [1701417669.964801453] [get_state]: pose state is: [0.000025, 0.000000, 0.850500, 0.000000, 0.000000, 0.000000]  
[INFO] [1701417669.964838239] [get_state]: arm_err is: 0  
[INFO] [1701417669.964852780] [get_state]: sys_err is: 0
```

3. 机械臂 MoveJ 运动。

控制机械臂进行关节运动，需要执行以下两条指令实现该功能。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py  
  
ros2 launch rm_example rm_<dof>_movej.launch.py
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

dof 代表机械当前的自由度信息，可以选的参数有 6dof 和 7dof。

例如启动 7 轴的机械臂时需要使用如下指令。

```
ros2 launch rm_example rm_7dof_movej.launch.py
```

运行成功后，终端将出现如下语句。

```
[INFO] [launch]: All log files can be found below /home/rm/.ros/log/2023-12-01-16-31-59-669068-rm-desktop-9816  
[INFO] [launch]: Default logging verbosity is set to INFO  
[INFO] [movej_demo-1]: process started with pid [9817]  
[movej_demo-1] [INFO] [1701419519.880260228] [Movej_demo]: arm_dof is 7  
[movej_demo-1]  
[movej_demo-1] [INFO] [1701419520.691271905] [Movej_demo]: *****Movej succeeded  
[movej_demo-1]
```

4. 机械臂 MoveJ_P 运动。

控制机械臂进行 MoveJ_P 关节运动，需要执行以下两条指令实现该功能。



```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py  
  
ros2 run rm_example movejp_demo
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

运行成功后，终端将出现如下语句。

```
[INFO] [1701421702.758055237] [Movejp_demo_node]: *****MoveJP succeeded
```

5. 机械臂 MoveL 运动。

控制机械臂进行 MoveL 关节运动，需要执行以下两条指令实现该功能。

```
ros2 launch rm_driver rm_<arm_type>_driver.launch.py  
  
ros2 run rm_example movel_demo
```

在实际使用时需要将以上的<arm_type>更换为实际的机械臂型号，可选择的机械臂型号有 65、63、eco65、75。

运行成功后，终端将出现如下语句。

```
[INFO] [1701422229.234605201] [MoveL_demo_node]: *****MoveJP succeeded  
[INFO] [1701422230.095942637] [MoveL_demo_node]: *****MoveL succeeded
```

10. rm_doc 功能包介绍

10.1 rm_doc 功能包作用

(1)rm_doc 为文档介绍功能包，主要包括各个功能包的介绍文档和使用方式的文档。

11. rm_install 功能包介绍

11.1 rm_install 功能包作用

(1) rm_install 为安装介绍功能包，主要包含三个脚本 1.ROS2 安装脚本



ros2_install.sh ; 2.moveit2 安装脚本 moveit2_install.sh ; 3.功能包安装脚本 lib_install.sh ,和一个安装说明的文档《睿尔曼 I ROS2 机械臂 rm_install 包详解 V1.0》 , 在实际使用时需要参考该文档进行脚本使用。