



睿尔曼机器人 rm_example 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	2023-11-22	拟制



目录

1. rm_example 功能包说明	3
2. rm_example 功能包使用	3
2.1 更换工作坐标系	3
2.2 得到当前的机械臂状态信息	4
2.3 机械臂 MoveJ 运动	4
2.4 机械臂 MoveJ_P 运动	5
2.5 机械臂 MoveL 运动	5
3. rm_example 功能包架构说明	6
3.1 功能包文件总览	6
4. rm_example 话题说明	6
4.1 rm_change_work_frame 话题说明	6
4.2 rm_get_state 话题说明	7
4.3 movej_demo 话题说明	7
4.4 movejp_demo 话题说明	7
4.5 movel_demo 话题说明	8



1. rm_example 功能包说明

rm_bringup 功能包为实现了一些基本的机械臂功能, 通过该功能包我们可以实现机械臂的一些基本的控制功能, 还可以参考代码, 实现其他的机械臂功能, 在下文中将通过以下几个方面详细介绍该功能包。

1. 功能包使用。
2. 功能包架构说明。
3. 功能包话题说明。

通过这三部分内容的介绍可以帮助大家:

1. 了解该功能包的使用。
2. 熟悉功能包中的文件构成及作用。
3. 熟悉功能包相关的话题, 方便开发和使用

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

2. rm_example 功能包使用

2.1 更换工作坐标系

首先需要运行机械臂的底层驱动节点 rm_driver。

```
rm@rm-desktop:~$ 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 65 机械臂的启动命令:

```
rm@rm-desktop:~$ ros2 launch rm_driver rm_65_driver.launch.py
```

节点启动成功后, 需要执行如下指令运行我们更换工作坐标系的节点。

```
rm@rm-desktop:~$ ros2 run rm_example rm_change_work_frame
```

弹出以下指令代表更换成功。

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

可以在终端中输入如下指令进行验证, 首先订阅当前的工作坐标系话题。

```
rm@rm-desktop:~$ ros2 topic echo /rm_driver/get_curr_workFrame_result
```

之后发布当前坐标系的请求。

```
rm@rm-desktop:~$ ros2 topic pub --once /rm_driver/get_curr_workFrame_cmd std_msgs/msg/Empty "{}"
```

可以看到终端中弹出如下界面。



```
rm@rm-desktop:~$ ros2 topic echo /rm_driver/get_curr_workFrame_result
data: Base
---
```

2.2 得到当前的机械臂状态信息

首先需要运行机械臂的底层驱动节点 `rm_driver`。

```
rm@rm-desktop:~$ 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 65 机械臂的启动命令:

```
rm@rm-desktop:~$ ros2 launch rm_driver rm_65_driver.launch.py
```

节点启动成功后, 需要执行如下指令运行获得机械臂当前状态的节点。

```
rm@rm-desktop:~$ ros2 run rm_example rm_get_state
```

弹出以下指令代表更换成功。

```
[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
```

界面中现实的为机械臂当前的角度信息, 以及机械臂当前的末端坐标位置和欧拉角姿态信息。

2.3 机械臂 MoveJ 运动

通过如下指令可以控制机械臂进行 MoveJ 关节运动。

首先需要运行机械臂的底层驱动节点 `rm_driver`。

```
rm@rm-desktop:~$ 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 65 机械臂的启动命令:

```
rm@rm-desktop:~$ ros2 launch rm_driver rm_65_driver.launch.py
```

节点启动成功后, 需要执行如下指令控制机械臂进行运动。

```
rm@rm-desktop:~$ ros2 launch rm_example rm_<dof>_movej.launch.py
```

命令中的 `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。

```
rm@rm-desktop:~$ ros2 launch rm_example rm_7dof_movej.launch.py
```

运行成功后, 机械臂的关节将发生转动, 且界面将显示如下信息。



```
[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]
```

2.4 机械臂 MoveJ_P 运动

通过如下指令可以控制机械臂进行 MoveJ_P 关节运动。

首先需要运行机械臂的底层驱动节点 rm_driver。

```
rm@rm-desktop:~$ 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 65 机械臂的启动命令:

```
rm@rm-desktop:~$ ros2 launch rm_driver rm_65_driver.launch.py
```

节点启动成功后, 需要执行如下指令控制机械臂进行运动。

```
rm@rm-desktop:~$ ros2 run rm_example movejp_demo
```

执行成功后界面将出现如下提示, 并且机械臂运动到指定位姿。

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

2.5 机械臂 MoveL 运动

通过如下指令可以控制机械臂进行 MoveL 关节运动。

首先需要运行机械臂的底层驱动节点 rm_driver。

```
rm@rm-desktop:~$ 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 65 机械臂的启动命令:

```
rm@rm-desktop:~$ ros2 launch rm_driver rm_65_driver.launch.py
```

节点启动成功后, 需要执行如下指令控制机械臂进行运动。

```
rm@rm-desktop:~$ ros2 run rm_example movel_demo
```

执行成功后界面将出现如下提示, 并且机械臂将进行两次运动, 首先通过 MoveJP 运动到指定位姿, 之后通过 MoveL 进行关节运动。

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



3. rm_example 功能包架构说明

3.1 功能包文件总览

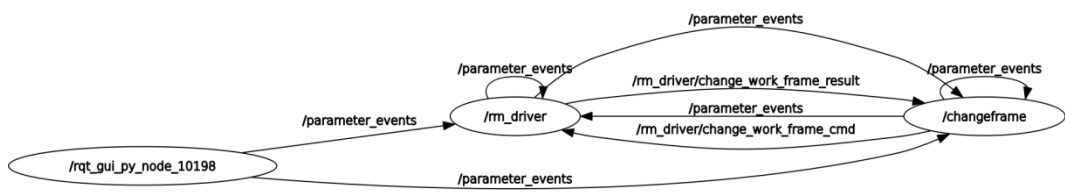
当前 rm_driver 功能包的文件构成如下。

└─ CMakeLists.txt	#编译规则文件
└─ include	
└─ rm_example	
└─ launch	
└─ rm_6dof_movej.launch.py	#6 自由度 MoveJ 运动启动文件
└─ rm_7dof_movej.launch.py	#7 自由度 MoveJ 运动启动文件
└─ package.xml	
└─ src	
└─ api_ChangeWorkFrame_demo.cpp	#更换工作坐标系源文件
└─ api_Get_Arm_State_demo.cpp	#获得机械臂状态源文件
└─ api_MoveJ_demo.cpp	#MoveJ 运动源文件
└─ api_MoveJP_demo.cpp	#MoveJP 运动源文件
└─ api_MoveL_demo.cpp	#MoveL 运动源文件

4. rm_example 话题说明

4.1 rm_change_work_frame 话题说明

以下为该节点的数据通信图：

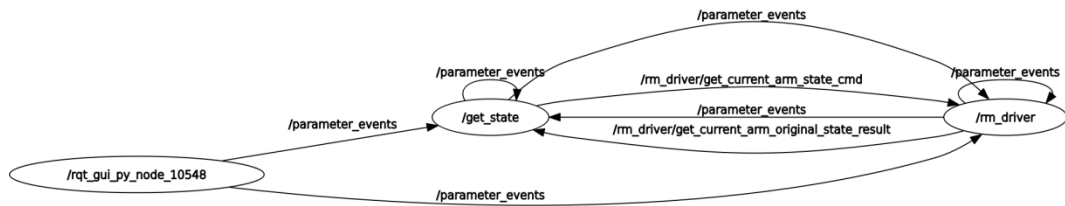




可以看到/changeframe 节点和/rm_driver 之间的主要通信话题为 /rm_driver/change_work_frame_result 和 /rm_driver/change_work_frame_cmd。 /rm_driver/change_work_frame_cmd 为切换请求和切换目标坐标的发布， /rm_driver/change_work_frame_result 为切换结果。

4.2 rm_get_state 话题说明

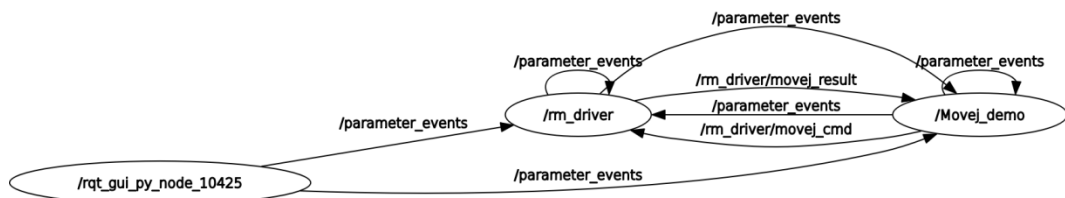
以下为该节点的数据通信图：



可以看到/get_state 节点和/rm_driver 之间的主要通信话题为 /rm_driver/get_current_arm_state_cmd 和 /rm_driver/get_current_arm_original_state_result。 /rm_driver/get_current_arm_state_cmd 为获取机械臂当前状态请求， /rm_driver/get_current_arm_original_state_result 为切换结果。

4.3 movej_demo 话题说明

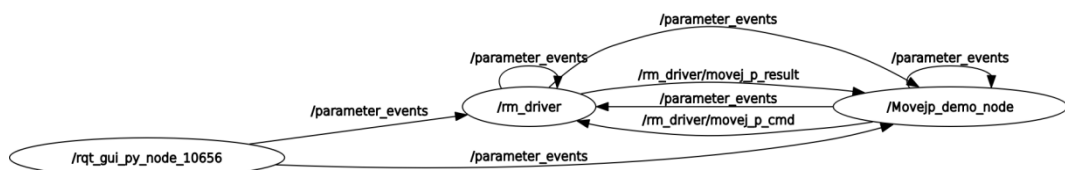
以下为该节点的数据通信图：



可以看到/Movej_demo 节点和/rm_driver 之间的主要通信话题为 /rm_driver/movej_cmd 和 /rm_driver/movej_result。 /rm_driver/movej_cmd 为控制机械臂运动的请求，将发布需要运动到的各关节的弧度信息， /rm_driver/movej_result 为运动结果。

4.4 movejp_demo 话题说明

以下为该节点的数据通信图：

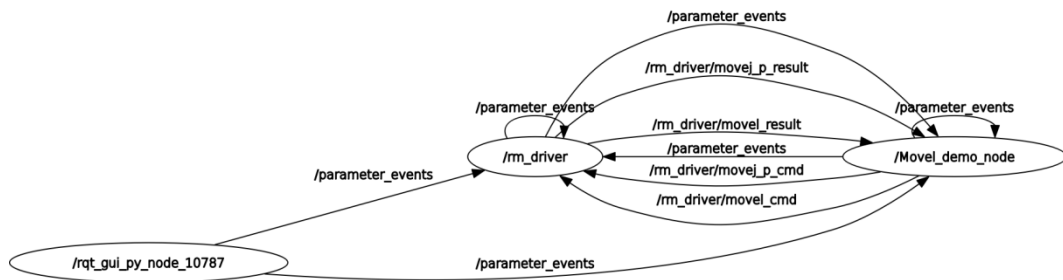




可以看到/Movejp_demo_node 节点和/rm_driver 之间的主要通信话题为 /rm_driver/movej_p_cmd 和/rm_driver/movej_p_result。/rm_driver/movej_p_cmd 为控制机械臂运动规划的请求，将发布需要运动到的目标点的坐标，/rm_driver/movej_p_result 为运动结果。

4.5 movej_demo 话题说明

以下为该节点的数据通信图：



可以看到/Movej_demo_node 节点和/rm_driver 之间的主要通信话题为 /rm_driver/movej_p_cmd 和/rm_driver/movej_p_result 还有/rm_driver/movel_cmd 和/rm_driver/movel_result。/rm_driver/movej_p_cmd 为控制机械臂运动规划的请求，将发布机械臂首先需要运动到的目标点的坐标，/rm_driver/movej_p_result 为运动结果，到达第一个点位后我们通过直线运动到达第二个点位，就可以通过/rm_driver/movel_cmd 发布第二个点位的位姿，/rm_driver/movel_result 话题代表运动的结果。