

Movelt drives the real machine

1. Usage environment

Motherboard: Jetson Orin Nano/Nx

ROS2: Humble

2. Drive the real machine

The real machine is driven by subscribing to the `/joint_states` topic of Moveit2 to convert the joint state information of the robot arm into the control of the real robot arm.

Note: Since the real robot arm does not have an obstacle avoidance function, some positions may encounter obstacles; so the planned robot arm movements should be as reasonable as possible and avoid positions with obstacles

(It is recommended to use preset positions to demonstrate driving the real machine)

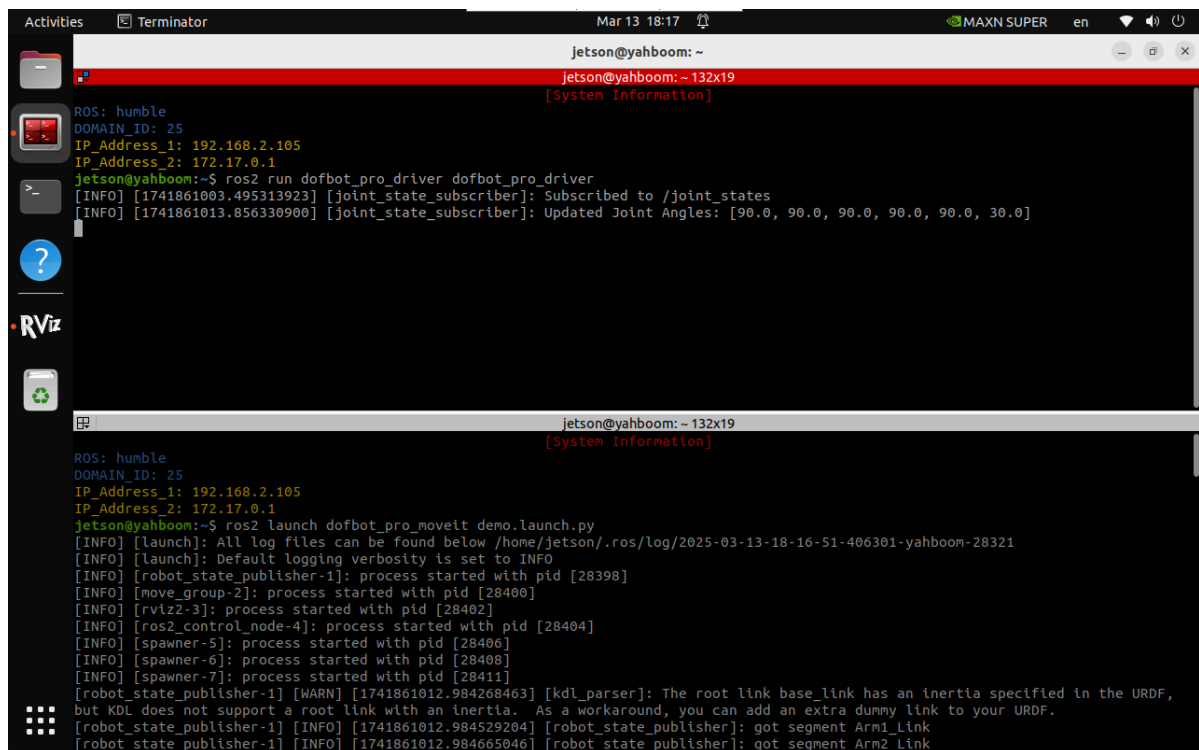
2.1. Start the real machine

If you do not drive the real machine, simulate the robot arm movements in Moveit:

```
ros2 run dofbot_pro_driver dofbot_pro_driver
```

2.2. Start Moveit2

```
ros2 launch dofbot_pro_moveit demo.launch.py
```



```
jetson@yahboom: ~  
jetson@yahboom: ~ 132x19  
[System Information]  
  
ROS: humble  
DOMAIN_ID: 25  
IP_Address_1: 192.168.2.105  
IP_Address_2: 172.17.0.1  
jetson@yahboom:~$ ros2 run dofbot_pro_driver dofbot_pro_driver  
[INFO] [1741861003.495313923] [joint_state_subscriber]: Subscribed to /joint_states  
[INFO] [1741861013.856330900] [joint_state_subscriber]: Updated Joint Angles: [90.0, 90.0, 90.0, 90.0, 90.0, 30.0]  
  
jetson@yahboom:~$ ros2 launch dofbot_pro_moveit demo.launch.py  
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-03-13-18-16-51-406301-yahboom-28321  
[INFO] [launch]: Default logging verbosity is set to INFO  
[INFO] [robot_state_publisher-1]: process started with pid [28398]  
[INFO] [move_group-2]: process started with pid [28400]  
[INFO] [rviz2-3]: process started with pid [28402]  
[INFO] [ros2_control_node-4]: process started with pid [28404]  
[INFO] [spawner-5]: process started with pid [28406]  
[INFO] [spawner-6]: process started with pid [28408]  
[INFO] [spawner-7]: process started with pid [28411]  
[robot_state_publisher-1] [WARN] [1741861012.984268463] [kdl_parser]: The root link base_link has an inertia specified in the URDF,  
but KDL does not support a root link with an inertia. As a workaround, you can add an extra dummy link to your URDF.  
[robot_state_publisher-1] [INFO] [1741861012.984529204] [robot_state_publisher]: got segment Arm1_Link  
[robot_state_publisher-1] [INFO] [1741861012.984665046] [robot_state_publisher]: got segment Arm2_Link
```

3. Real machine movement

Demonstrating driving the real robot arm to the preset initialization position:

Set the planning group: arm_group

Set the pose: init

Plan and execute pose: Plan & Execute

