ROS 2-based Control of Robotis Motor Units

Version: 1.0, 2021/06/21

Author: Qiu zhe

PREFACE

This introduction aims to provide a tutorial about how to control Robotis Motor Units by using the Robot Operating System 2 (ROS2). The introduction is compatible across the X-series Robotis motor units. Especially, XM-430-W210-R is used for explanation.

Content

1. Preliminaries

1.1 ROS 2 distribution

ROS 2 Foxy is highly recommended.

(ROS 2 Foxy Fitzroy, released June 5th, 2020, supported until May 2023)

Installation: https://docs.ros.org/en/foxy/Installation/Ubuntu-Development-Setup.html

1.2 Ubuntu system

To match ROS 2 Foxy distribution, Ubuntu 20.04 is required.

Installation: https://ubuntu.com/download/desktop

2. Introduction

This introduction is based on the official Dynamixel SDK:
https://github.com/ROBOTIS-GIT/DynamixelSDK/releases/tag/3.7.40
In addition, an official video tutorial can be referred as follows:
https://www.youtube.com/watch?v=E8XPqDjof4U

2.1 Download and build ROS 2 packages

Follow the steps below to build the required ROS 2 packages:

Step 1:

```
# create a new ROS 2 workspace
$ mkdir -p ~/robotis_ws/src
```

Step 2:

```
# download the Dynamixel SDK
$ cd ~/robotis_ws/src
$ git clone -b $ROS_DISTRO-devel https://github.com/ROBOTIS-GIT/DynamixelSDK
```

Step 3:

```
# Build and setup Dynamixel SDK
$ cd ~/robotis_ws && colcon build -symlink-install
```

Step 4:

```
# Source ROS 2 bash file
$ source /opt/ros/foxy/setup.bash
# Source Dynamixel SDK bash file
$ cd ~/robotis_ws
$ . install/local_setup.bash
```

2.2 Hardware setup: setup a Robotis motor unit

2.2.1 Hardware preparation

The following components should be prepared.

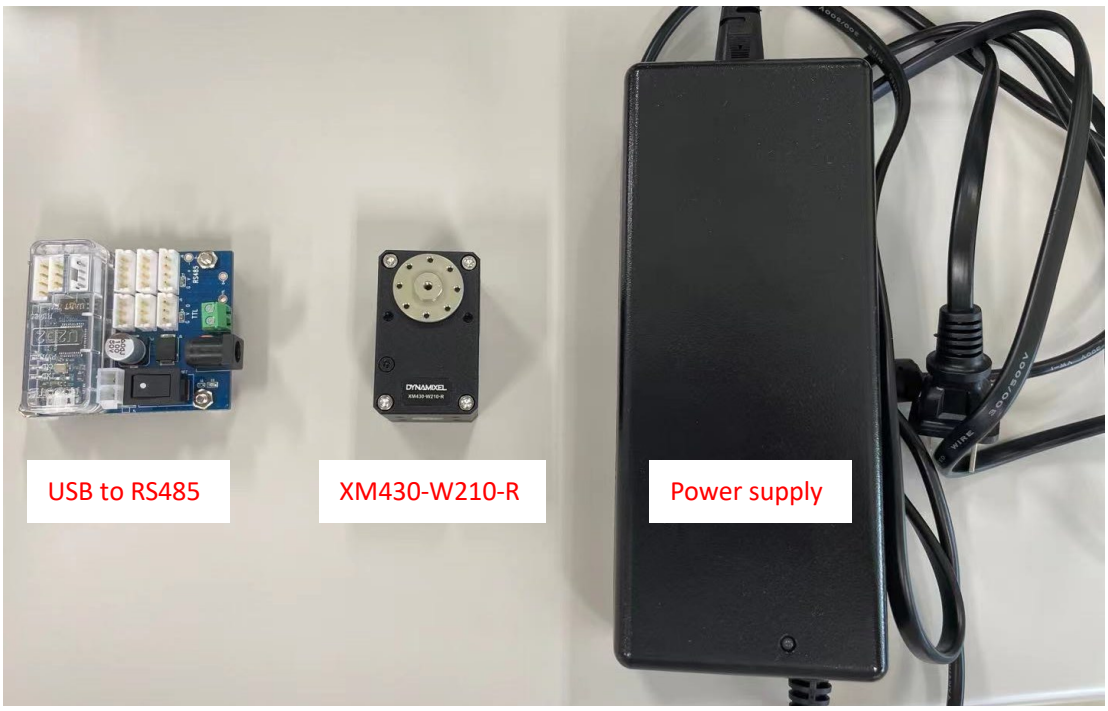

Figure 1: U2D2 Power Hub Board (left), XM430-W210-R (mid) and power supply (right).

2.2.2 Hardware connection

The hardware connection should be referred as follows.

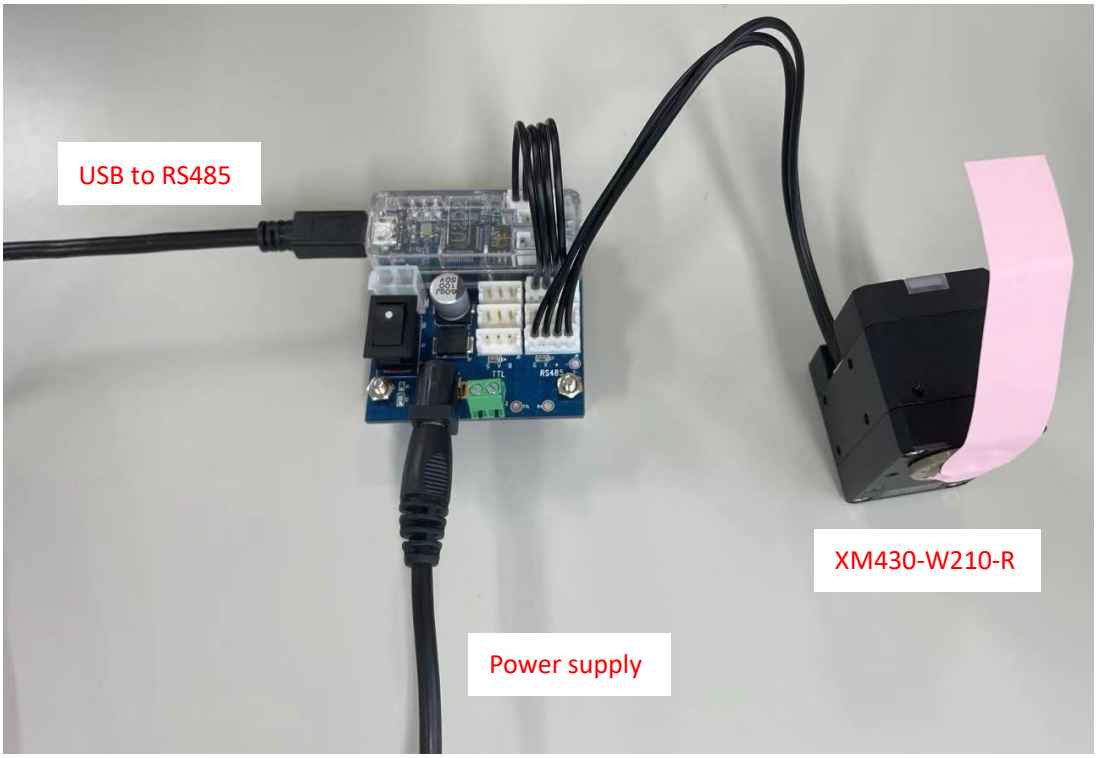

Figure 2: Hardware connection of the Robotis Motor Unit

3. Examples

3.1 Preparation

The USB port connection should be checked first.

Open a Terminal,

```
$ ls /dev/tty*
```

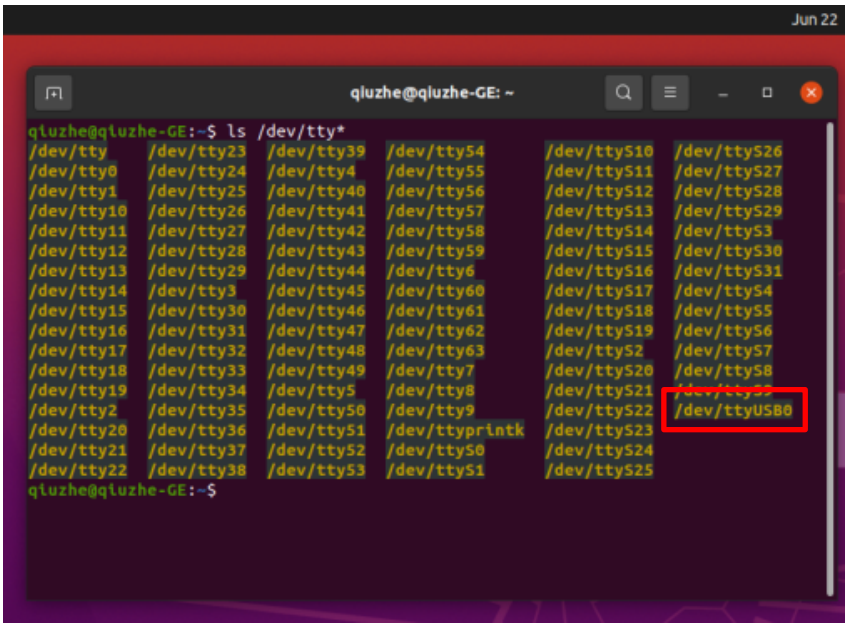

Figure 3 Port check

Then, Port permission is required.

```
$ sudo usermod -aG dialout <your linux account>
$ <your password>
```

Notice: remember to reboot the control PC (laptop), if this is the first time controlling the Robotis motor units. Otherwise, the port might not be successfully connected.

3.2 Official package: test position control of a XM430-W210-R

The XM430-W210-R is controlled via `read_write_mode` of the `Dynamixel SDK examples` by using a control PC (ROS 2 Foxy with Ubuntu 20.04).

The following steps are recommended:

Step 1:
Connect the hardware (Sec 2.2), and confirm the port connection and permission (Sec 3.1).

Step 2:
Source the bash files.

Open Terminal 1:

```
$ source /opt/ros/foxy/setup.bash
$ cd ~/robotis_ws
$ . install/local_setup.bash
```

Step 3:
Run the `read_write_node` of the `Dynamixel SDK examples`.

```
$ ros2 run dynamixel_sdk_examples read_write_node
```

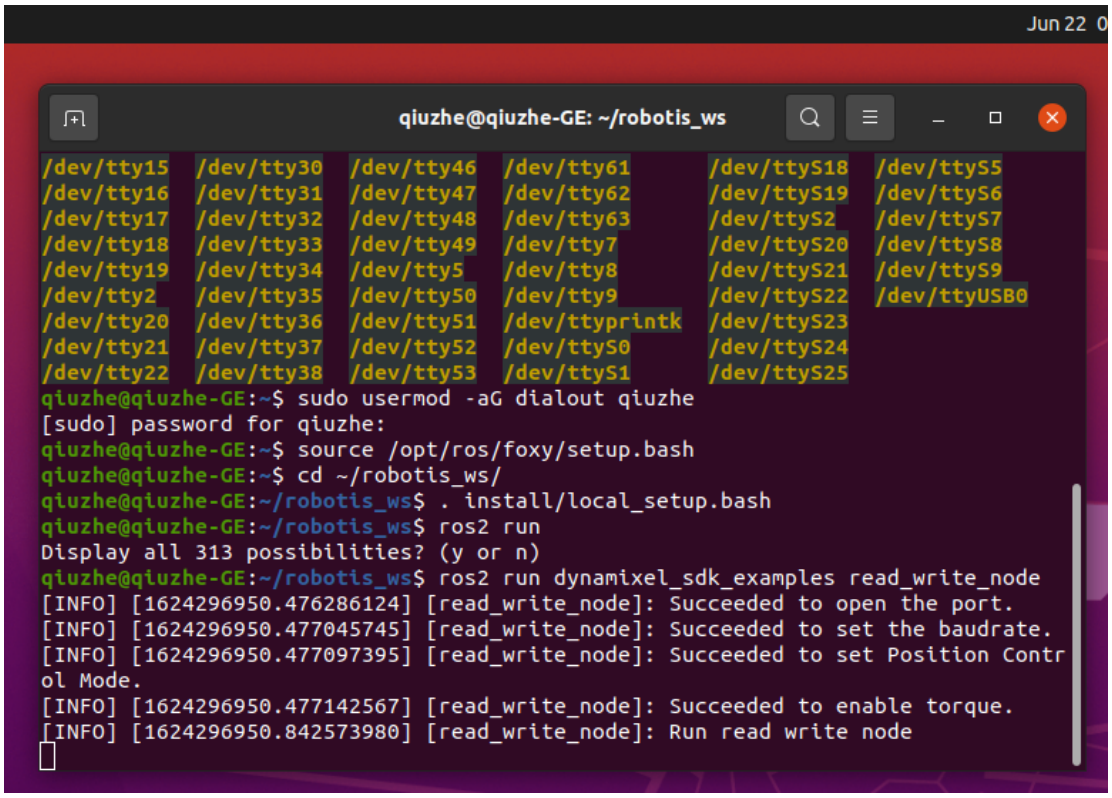

Figure 4 Run the read_write_node of dynamixel_sdk_examples

Step 4:
Send desired position to the Robotis motor unit.

Open Terminal 2:

```
$ source /opt/ros/foxy/setup.bash
$ cd ~/robotis_ws
$ . install/local_setup.bash
$ ros2 topic pub -1 /set_position
Dynamixel_sdk_custom_interfaces/msg/SetPosition “{id: 1, position: 1000}”
```

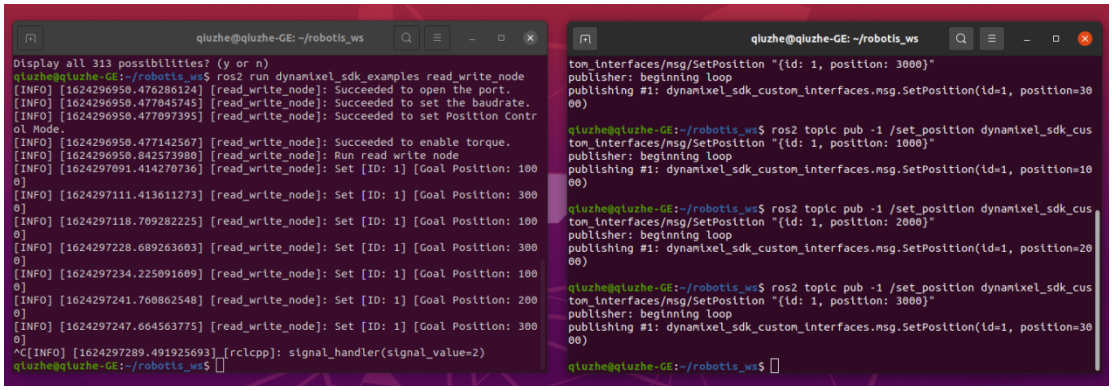

Figure 5 Send desired positions to the Robotis Motor Unit

3.3 Modified package: test position control of two XM430-W210-R

Two XM430-W210-R motors are controlled via modified ROS 2 package by using a control PC (ROS 2 Foxy with Ubuntu 20.04).

The following steps are recommended:

Step 1:
Connect the hardware (Sec 2.2), and confirm the port connection and permission (Sec 3.1).

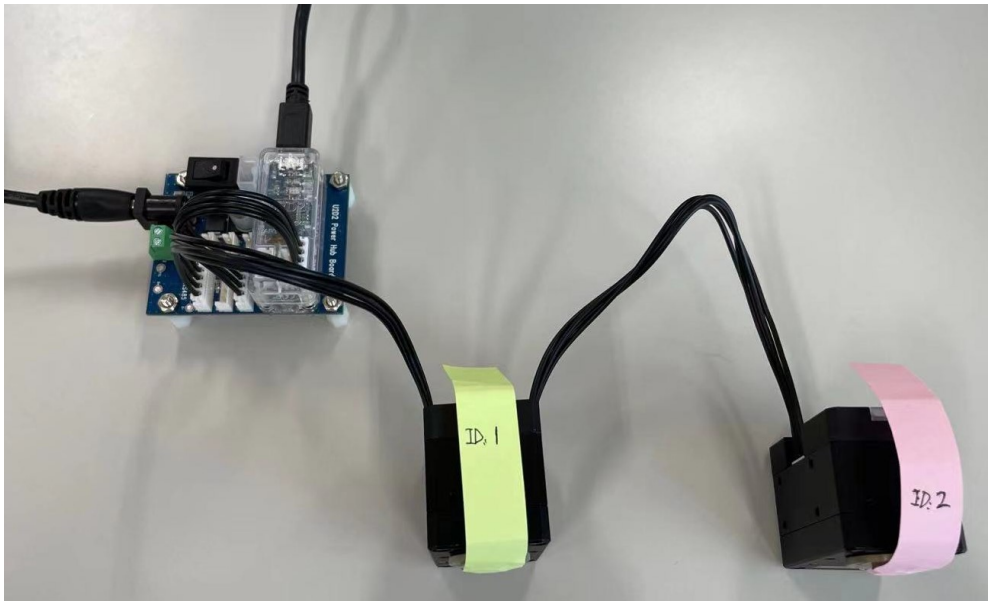

Figure 6 Hardware connections of two Robotis Motor Units

Step 2:
Source the bash files.

Open Terminal 1:

```
$ source /opt/ros/foxy/setup.bash
$ cd ~/robotis_ws
$ . install/local_setup.bash
```

Step 3:
This example is realized by modifying the `read_write_node` of the `Dynamixel SDK examples`, thus, we still run the read_write_node

```
$ ros2 run dynamixel_sdk_examples read_write_node
```

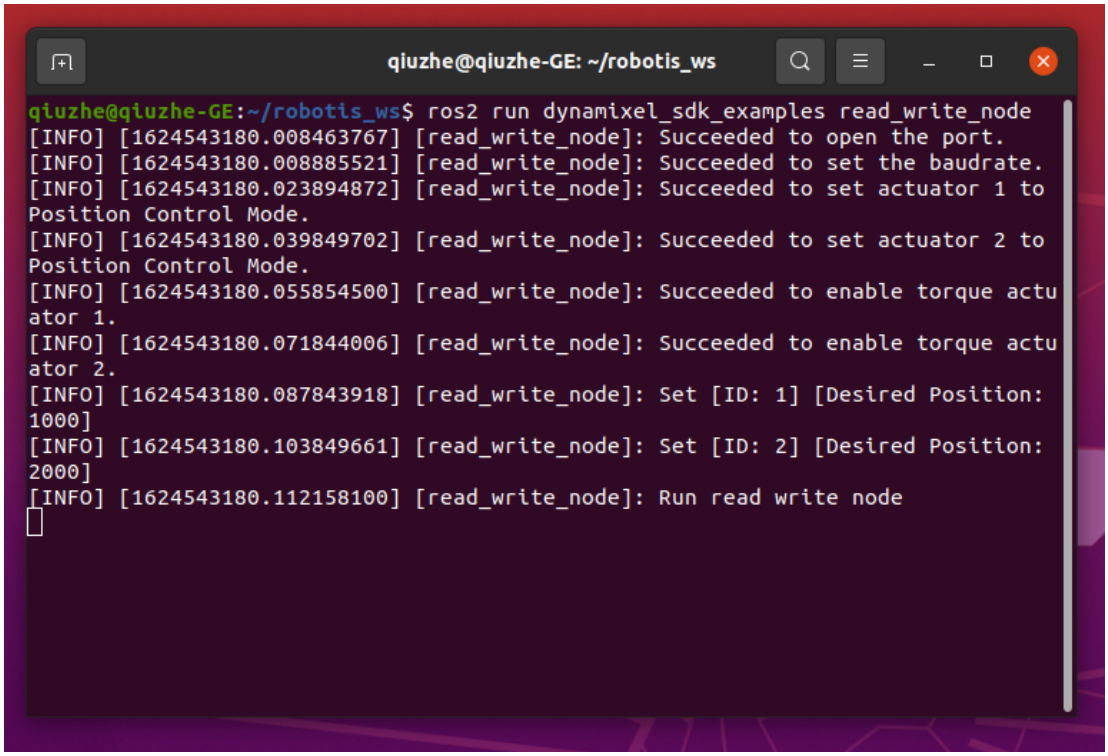

Figure 7 Run the modified read_write_node

The desired position of ID 1 is 1000, and the desired position of ID 2 is 2000. The desired position of each motor can be changed in the program, and then the package should be recompiled.

The programming codes of some key functions are given as follows:

```
// Use Position Control Mode
dxl_comm_result1 = packetHandler->write1ByteTxRx(
    portHandler,
    DXL1_ID,
    ADDR_OPERATING_MODE,
    3,
    &dxl_error
    );

  // Enable Torque of DYNAMIXEL
    dxl_comm_result1 = packetHandler->write1ByteTxRx(
    portHandler,
    DXL1_ID,
    ADDR_TORQUE_ENABLE,
    1,
    &dxl_error
    );

// Write Goal Position of actuator 1
```

```
dxl_comm_result1 =
  packetHandler->write4ByteTxRx(
  portHandler,
  DXL1_ID,
  ADDR_GOAL_POSITION,
  desired_position1,
  &dxl_error
);

// Disable first actuator Torque of DYNAMIXEL
  packetHandler->write1ByteTxRx(
  portHandler,
  DXL1_ID,
  ADDR_TORQUE_ENABLE,
  0,
  &dxl_error
  );
```