

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

1.2 Ubuntu system

2. Introduction

2.1 Download and build ROS 2 packages

2.2 Hardware setup: setup a Robotis motor unit

2.2.1 Hardware preparation

2.2.2 Hardware connection

3. Examples

3.1 Preparation

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

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

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.

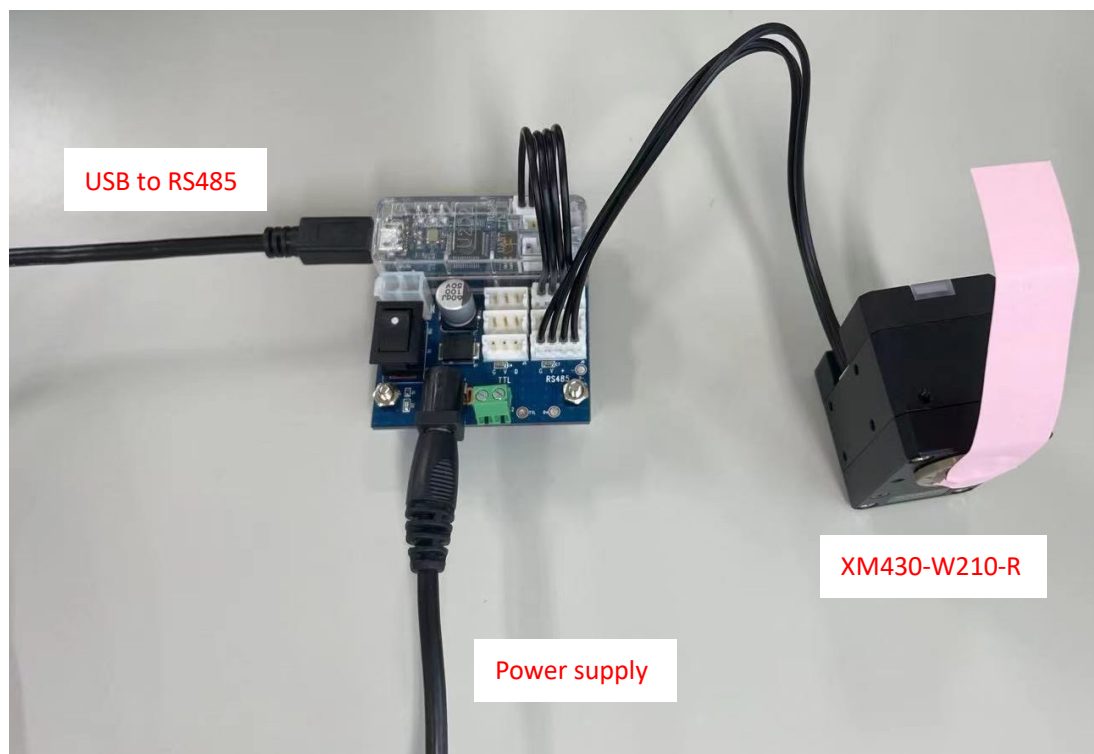


Figure 2: Hardware connection of the Robotis Motor Unit

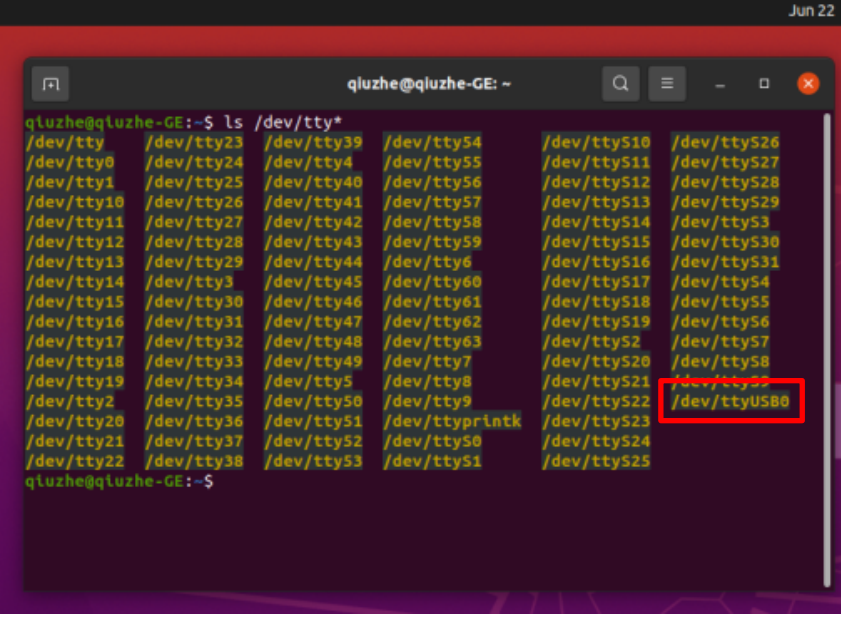
3. Examples

3.1 Preparation

The USB port connection should be checked first.

Open a Terminal,

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



```
qluzhe@qluzhe-GE:~$ ls /dev/tty*
/dev/tty      /dev/tty23  /dev/tty39  /dev/tty54  /dev/tty510 /dev/tty526
/dev/tty0     /dev/tty24  /dev/tty4    /dev/tty55  /dev/tty511 /dev/tty527
/dev/tty1     /dev/tty25  /dev/tty40  /dev/tty56  /dev/tty512 /dev/tty528
/dev/tty10    /dev/tty26  /dev/tty41  /dev/tty57  /dev/tty513 /dev/tty529
/dev/tty11    /dev/tty27  /dev/tty42  /dev/tty58  /dev/tty514 /dev/tty53
/dev/tty12    /dev/tty28  /dev/tty43  /dev/tty59  /dev/tty515 /dev/tty530
/dev/tty13    /dev/tty29  /dev/tty44  /dev/tty6    /dev/tty516 /dev/tty531
/dev/tty14    /dev/tty3   /dev/tty45  /dev/tty60  /dev/tty517 /dev/tty54
/dev/tty15    /dev/tty30  /dev/tty46  /dev/tty61  /dev/tty518 /dev/tty55
/dev/tty16    /dev/tty31  /dev/tty47  /dev/tty62  /dev/tty519 /dev/tty56
/dev/tty17    /dev/tty32  /dev/tty48  /dev/tty63  /dev/tty52  /dev/tty57
/dev/tty18    /dev/tty33  /dev/tty49  /dev/tty7    /dev/tty520 /dev/tty58
/dev/tty19    /dev/tty34  /dev/tty5   /dev/tty8    /dev/tty521 /dev/tty59
/dev/tty2     /dev/tty35  /dev/tty50  /dev/tty9    /dev/tty522 /dev/ttyUSB0
/dev/tty20    /dev/tty36  /dev/tty51  /dev/ttyprintk /dev/tty523
/dev/tty21    /dev/tty37  /dev/tty52  /dev/tty50   /dev/tty524
/dev/tty22    /dev/tty38  /dev/tty53  /dev/tty51   /dev/tty525
qluzhe@qluzhe-GE:~$
```

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.

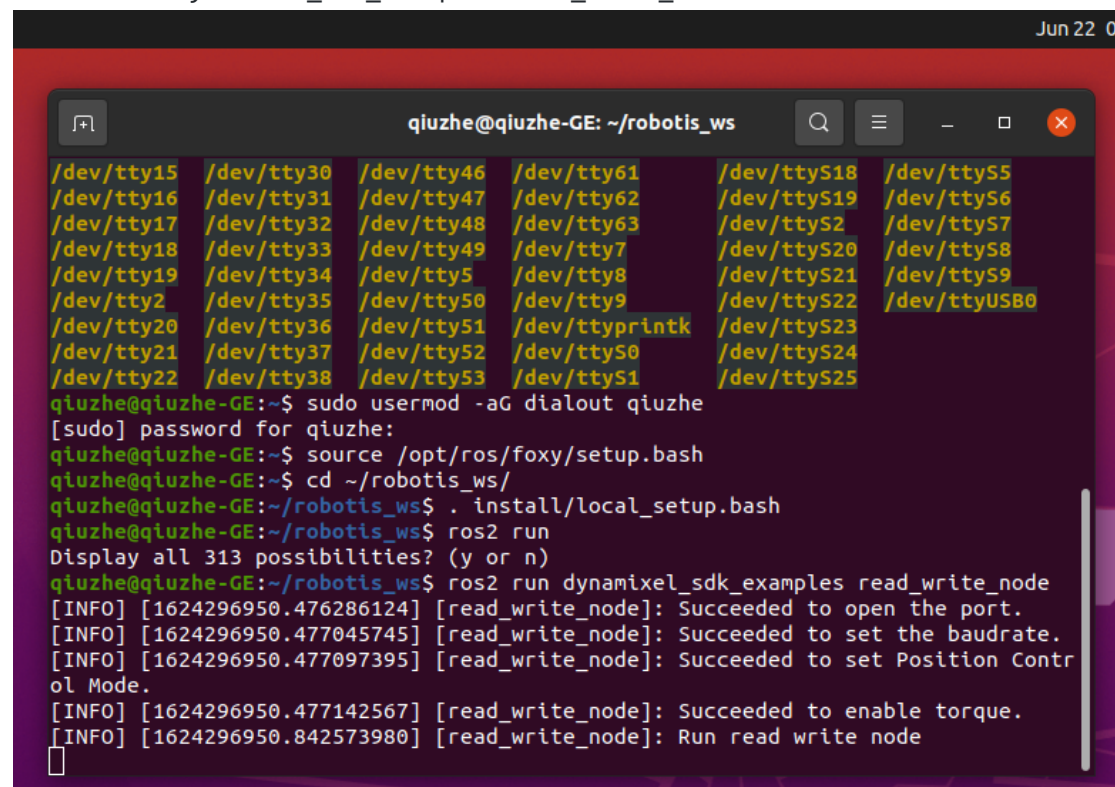
Open Terminal 1:

```
$ 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
```



```
Jun 22 0
qiuzhe@qiuzhe-GE: ~/robotis_ws
/dev/tty15 /dev/tty30 /dev/tty46 /dev/tty61 /dev/ttyS18 /dev/ttyS5
/dev/tty16 /dev/tty31 /dev/tty47 /dev/tty62 /dev/ttyS19 /dev/ttyS6
/dev/tty17 /dev/tty32 /dev/tty48 /dev/tty63 /dev/ttyS20 /dev/ttyS7
/dev/tty18 /dev/tty33 /dev/tty49 /dev/tty7 /dev/ttyS21 /dev/ttyS8
/dev/tty19 /dev/tty34 /dev/tty5 /dev/tty8 /dev/ttyS22 /dev/ttyS9
/dev/tty2 /dev/tty35 /dev/tty50 /dev/tty9 /dev/ttyS23 /dev/ttyUSB0
/dev/tty20 /dev/tty36 /dev/tty51 /dev/ttyprintk /dev/ttyS24
/dev/tty21 /dev/tty37 /dev/tty52 /dev/ttyS0 /dev/ttyS25
/dev/tty22 /dev/tty38 /dev/tty53 /dev/ttyS1
qiuzhe@qiuzhe-GE:~$ sudo usermod -aG dialout qiuzhe
[sudo] password for qiuzhe:
qiuzhe@qiuzhe-GE:~$ source /opt/ros/foxy/setup.bash
qiuzhe@qiuzhe-GE:~$ cd ~/robotis_ws/
qiuzhe@qiuzhe-GE:~/robotis_ws$ . install/local_setup.bash
qiuzhe@qiuzhe-GE:~/robotis_ws$ ros2 run
Display all 313 possibilities? (y or n)
qiuzhe@qiuzhe-GE:~/robotis_ws$ ros2 run dynamixel_sdk_examples read_write_node
[INFO] [1624296950.476286124] [read_write_node]: Succeeded to open the port.
[INFO] [1624296950.477045745] [read_write_node]: Succeeded to set the baudrate.
[INFO] [1624296950.477097395] [read_write_node]: Succeeded to set Position Contr
ol Mode.
[INFO] [1624296950.477142567] [read_write_node]: Succeeded to enable torque.
[INFO] [1624296950.842573980] [read_write_node]: Run 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:

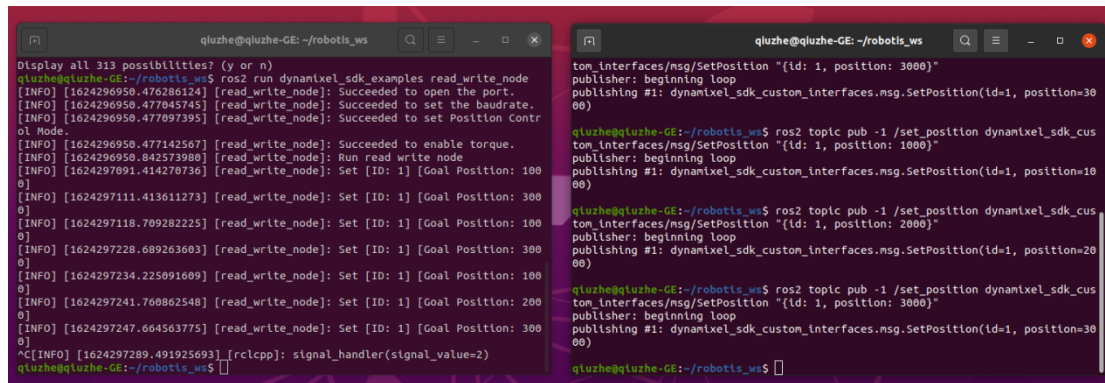
```
$ source /opt/ros/foxy/setup.bash
```

```
$ cd ~/robotis_ws
```

```
$ . install/local_setup.bash
```

```
$ ros2 topic pub -1 /set_position
```

```
Dynamixel_sdk_custom_interfaces/msg/SetPosition "{id: 1, position: 1000}"
```



The image shows two terminal windows side-by-side. The left window displays the output of running `ros2 run dynamixel_sdk_examples read_write_node`, showing logs for opening the port, setting the baudrate, and setting the position control mode. The right window shows the output of running `ros2 topic pub -1 /set_position dynamixel_sdk_custom_interfaces/msg/SetPosition "{id: 1, position: 1000}"`, showing the publisher beginning a loop and publishing the message.

```
qluzhe@qluzhe-GE: ~/robotis_ws
Display all 313 possibilities? (y or n)
qluzhe@qluzhe-GE:~/robotis_ws$ ros2 run dynamixel_sdk_examples read_write_node
[INFO] [1624296950.476886124] [read_write_node]: Succeeded to open the port.
[INFO] [1624296950.477045745] [read_write_node]: Succeeded to set the baudrate.
[INFO] [1624296950.477097395] [read_write_node]: Succeeded to set Position Control Mode.
[INFO] [1624296950.477142567] [read_write_node]: Succeeded to enable torque.
[INFO] [1624296950.842573980] [read_write_node]: Run read write node
[INFO] [1624297091.414270736] [read_write_node]: Set [ID: 1] [Goal Position: 1000]
[INFO] [1624297111.413611273] [read_write_node]: Set [ID: 1] [Goal Position: 3000]
[INFO] [1624297118.709282225] [read_write_node]: Set [ID: 1] [Goal Position: 1000]
[INFO] [1624297228.689263603] [read_write_node]: Set [ID: 1] [Goal Position: 3000]
[INFO] [1624297234.225091609] [read_write_node]: Set [ID: 1] [Goal Position: 1000]
[INFO] [1624297241.760862548] [read_write_node]: Set [ID: 1] [Goal Position: 2000]
[INFO] [1624297247.664563775] [read_write_node]: Set [ID: 1] [Goal Position: 3000]
^C[INFO] [1624297289.491925693] [rclcpp]: signal_handler(signal_value=2)
qluzhe@qluzhe-GE:~/robotis_ws$

qluzhe@qluzhe-GE:~/robotis_ws
tom_interfaces/msg/SetPosition "{id: 1, position: 3000}"
publisher: beginning loop
publishing #1: dynamixel_sdk_custom_interfaces.msg.SetPosition(id=1, position=3000)
qluzhe@qluzhe-GE:~/robotis_ws$ ros2 topic pub -1 /set_position dynamixel_sdk_custom_interfaces/msg/SetPosition "{id: 1, position: 1000}"
publisher: beginning loop
publishing #1: dynamixel_sdk_custom_interfaces.msg.SetPosition(id=1, position=1000)
qluzhe@qluzhe-GE:~/robotis_ws$ ros2 topic pub -1 /set_position dynamixel_sdk_custom_interfaces/msg/SetPosition "{id: 1, position: 2000}"
publisher: beginning loop
publishing #1: dynamixel_sdk_custom_interfaces.msg.SetPosition(id=1, position=2000)
qluzhe@qluzhe-GE:~/robotis_ws$ ros2 topic pub -1 /set_position dynamixel_sdk_custom_interfaces/msg/SetPosition "{id: 1, position: 3000}"
publisher: beginning loop
publishing #1: dynamixel_sdk_custom_interfaces.msg.SetPosition(id=1, position=3000)
qluzhe@qluzhe-GE:~/robotis_ws$
```

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).

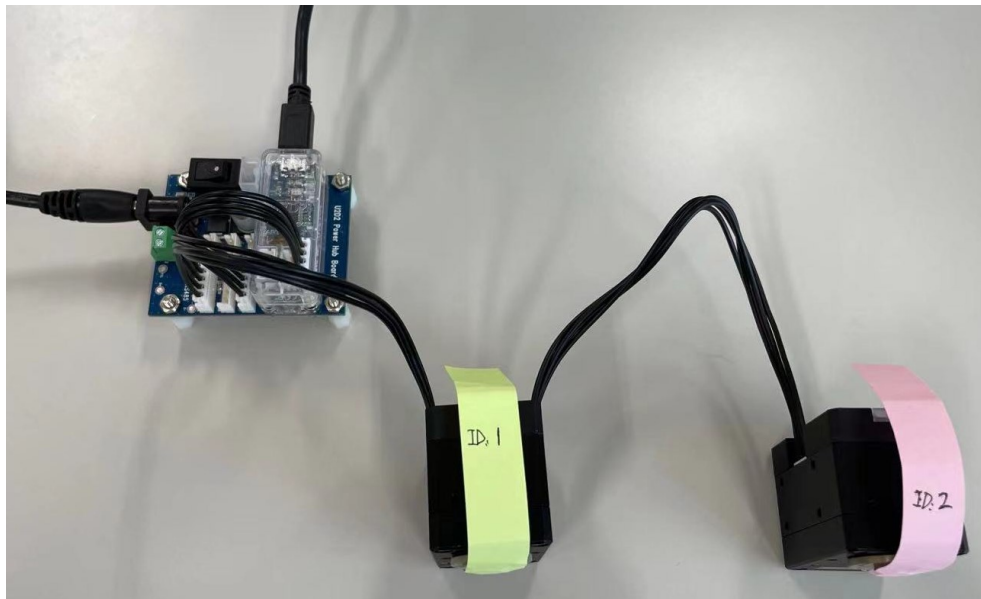


Figure 6 Hardware connections of two Robotis Motor Units

Step 2:

Source the bash files.

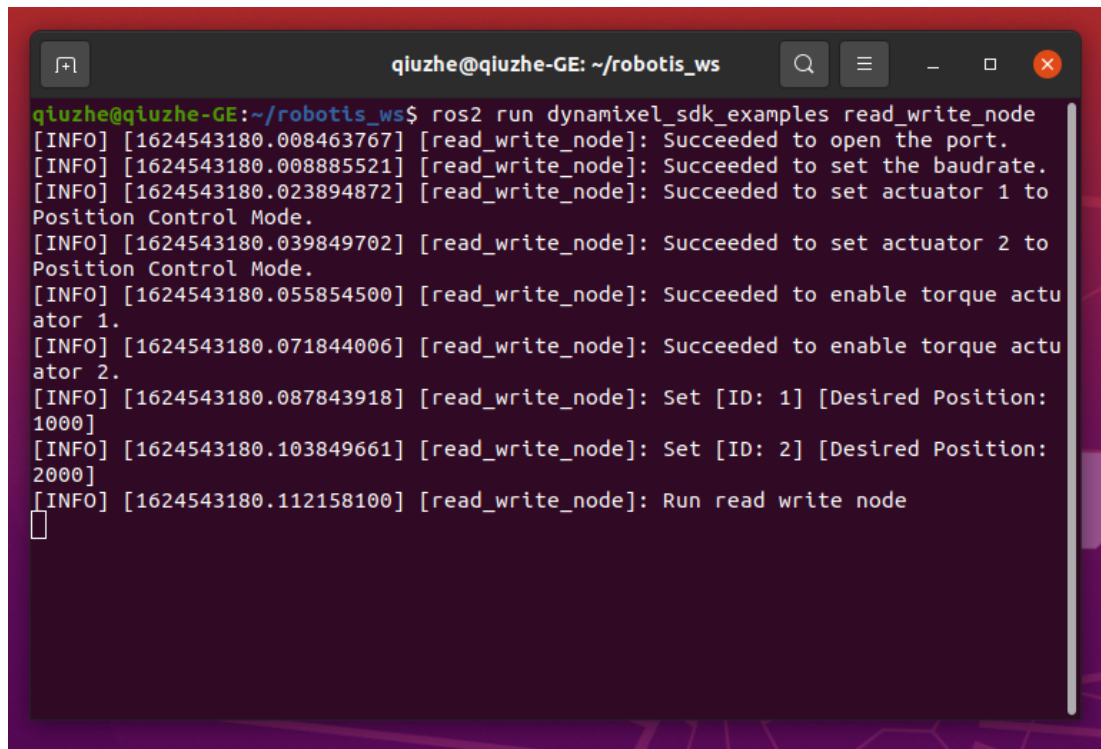
Open Terminal 1:

```
$ 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
```

A terminal window titled 'qiuzhe@qiuzhe-GE: ~/robotis_ws' showing the execution of the 'read_write_node' program. The output consists of several log messages with timestamps and status reports. The messages indicate successful operations: opening the port, setting the baudrate, setting actuators 1 and 2 to Position Control Mode, enabling torque for both actuators, and setting desired positions for ID 1 (1000) and ID 2 (2000). The terminal ends with a cursor on a new line.

```
qiuzhe@qiuzhe-GE:~/robotis_ws$ ros2 run dynamixel_sdk_examples read_write_node
[INFO] [1624543180.008463767] [read_write_node]: Succeeded to open the port.
[INFO] [1624543180.008885521] [read_write_node]: Succeeded to set the baudrate.
[INFO] [1624543180.023894872] [read_write_node]: Succeeded to set actuator 1 to
Position Control Mode.
[INFO] [1624543180.039849702] [read_write_node]: Succeeded to set actuator 2 to
Position Control Mode.
[INFO] [1624543180.055854500] [read_write_node]: Succeeded to enable torque actu
ator 1.
[INFO] [1624543180.071844006] [read_write_node]: Succeeded to enable torque actu
ator 2.
[INFO] [1624543180.087843918] [read_write_node]: Set [ID: 1] [Desired Position:
1000]
[INFO] [1624543180.103849661] [read_write_node]: Set [ID: 2] [Desired Position:
2000]
[INFO] [1624543180.112158100] [read_write_node]: Run 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
    );

```