ROS2 Modbus Usage Guide

TechShare

1. 1. Overview

The inspire_hand package (ROS2 version) is designed for using multi-fingered hands and robot grippers from Inspire-Robots on the ROS platform.

Currently, it has only been verified in Ubuntu 22.04 ROS2 Humble environment. Please await future development for other ROS environments.

2. 2. Environment Setup

To run the program normally, the following environment setup is required (only for the first time; once set up, it doesn't need to be done again).

2.1 1) ROS2 Humble Environment Installation

For detailed installation instructions, please refer to the following link.

ROS2 Humble Install

2.2 2) Modbus Library Installation

Run the following command in the terminal.

```
sudo apt-get install libmodbus-dev
```

Note: If there are other missing dependencies, please download the missing items according to the error messages in the terminal during cmake compilation.

2.3 3) Catkin Workspace Creation

Execute the following commands in sequence in the terminal.

```
mkdir -p ~/inspire_hand_ws/src
cd ~/inspire_hand_ws
colcon build
source install/setup.bash  # This command needs to be run every time you open a new terminal to find the ROS installation directory.
```

2.4 4) Package Extraction

Place inspire_hand_ros2.zip in the `~/inspire_hand_ws/src` directory and extract it.

```
cd ~/inspire_hand_ws/src
unzip inspire_hand_ros2.zip
```

After extraction, move the two folders inspire_hand_modbus_ros2 and service_interfaces to `~/inspire_hand_ws/src`, and delete the original inspire_hand_ros2 folder.

2.5 5) Package Recompilation

Run the following commands in the terminal.

```
colcon build --packages-select service_interfaces
colcon build --packages-select inspire_hand_modbus_ros2
```

> **Note:** To avoid environment variable conflicts, please avoid adding excessive source commands to the bash file using `sudo gedit ~/.bashrc`. Also, avoid duplicate package names like "service_interfaces". These operations may cause message reference errors or node startup errors.

3. 3. 5-Finger Hand Usage Guide

3.1 1) Hardware Connection

Connect the Inspire Hand and host PC with a LAN cable. Change your PC's IPv4 settings as follows:

Setting	Value
IP Address	192.168.11.222
Subnet Mask	255.255.255.0

Run the following command in the terminal; if data returns, the connection is successful.

```
ping 192.168.11.210
```

If there is no response, check the cable connection.

3.2 2) Running the inspire_hand_modbus_ros2 Package

Open a new terminal and first run the following commands.

```
source install/setup.bash
ros2 run inspire_hand_modbus_ros2 hand_modbus_control_node
```

Below are examples of various operations using service calls.

3.2.1 (1) ID Setting

id range: 1-254

```
ros2 service call /Setid service_interfaces/srv/Setid "{id: 2, status: 'set_id'}"
```

3.2.2 (2) Baud Rate Setting

redu_ratio range: 0-4

```
ros2 service call /Setreduratio service_interfaces/srv/Setreduratio "{redu_ratio: 0, id: 1,
status: 'set_reduratio'}"
```

3.2.3 (3) 6-Axis Driver Position Setting

pos range: 0-2000

```
ros2 service call /Setpos service_interfaces/srv/Setpos "{pos0: 1000, pos1: 1000, pos2: 1000,
pos3: 1000, pos4: 1000, pos5: 1000, id: 1, status: 'set_pos'}"
```

3.2.4 (4) Speed Setting

speed range: 0-1000

```
ros2 service call /Setspeed service_interfaces/srv/Setspeed "{speed0: 50, speed1: 50, speed2:
50, speed3: 50, speed4: 50, speed5: 50, id: 1, status: 'set_speed'}"
```

3.2.5 (5) 5-Finger Hand Angle Setting

angle range: 0-1000

```
ros2 service call /Setangle service_interfaces/srv/Setangle "{angle0: 1000, angle1: 1000,
angle2: 1000, angle3: 1000, angle4: 1000, angle5: 1000, id: 1, status: 'set_angle'}"
```

3.2.6 (6) Force Control Threshold Setting

force range: 0-1000

```
ros2 service call /Setforce service_interfaces/srv/Setforce "{force0: 0, force1: 0, force2: 0,
force3: 1000, force4: 0, force5: 0, id: 1, status: 'set_force'}"
```

3.2.7 (7) Current Threshold Setting

current range: 0-1500

```
ros2 service call /Setcurrentlimit service_interfaces/srv/Setcurrentlimit "{current0: 1500,
current1: 1500, current2: 1500, current3: 1500, current4: 1500, current5: 1500, id: 1, status:
'set_currentlimit'}"
```

3.2.8 (8) Power-On Speed Setting (Effective After Restart)

speed range: 0-1000

```
ros2 service call /Setdefaultspeed service_interfaces/srv/Setdefaultspeed "{speed0: 1000,
speed1: 1000, speed2: 1000, speed3: 1000, speed4: 1000, speed5: 100, id: 1, status:
'set_defaultspeed'}"
```

3.2.9 (9) Power-On Force Control Threshold Setting (Effective After Restart)

force range: 0-1000

```
ros2 service call /Setdefaultforce service_interfaces/srv/Setdefaultforce "{force0: 1000,
force1: 1000, force2: 1000, force3: 1000, force4: 1000, force5: 1000}"
```

3.2.10 (10) Power-On Current Threshold Setting (Effective After Restart)

current range: 0-1500

```
ros2 service call /Setdefaultcurrentlimit service_interfaces/srv/Setdefaultcurrentlimit "
{current0: 1500, current1: 1500, current2: 1500, current3: 1500, current4: 1500, current5:
1500}"
```

3.2.11 (11) Force Sensor Calibration

This command needs to be executed twice. After execution, the hand will fully open, and then the force sensor will be calibrated.

```
ros2 service call /Setforceclb service_interfaces/srv/Setforceclb "{id: 1, status:
'set_forceclb'}"
```

3.2.12 (12) Clear Errors

```
ros2 service call /Setclearerror service_interfaces/srv/Setclearerror "{id: 1, status:
'set_clearerror'}"
```

3.2.13 (13) Reset to Factory Settings

```
ros2 service call /Setresetpara service_interfaces/srv/Setresetpara "{id: 1, status:
'set_resetpara'}"
```

3.2.14 (14) Save Parameters to FLASH Memory

```
ros2 service call /Setsaveflash service_interfaces/srv/Setsaveflash "{id: 1, status:
'set_saveflash'}"
```

3.2.15 (15) Read Set Actuator Position Value

```
ros2 service call /Getposset service_interfaces/srv/Getposset "{id: 1, status: 'get_posset'}"
```

3.2.16 (16) Read Set Hand Angle Value

```
ros2 service call /Getangleset service_interfaces/srv/Getangleset "{id: 1, status:
'get_angleset'}"
```

3.2.17 (17) Read Set Force Control Threshold

```
ros2 service call /Getforceset service_interfaces/srv/Getforceset "{id: 1, status:
'get_forceset'}"
```

3.2.18 (18) Read Current Value

```
ros2 service call /Getcurrentact service_interfaces/srv/Getcurrentact "{id: 1, status:
'get_currentact'}"
```

3.2.19 (19) Read Actual Actuator Position Value

```
ros2 service call /Getposact service_interfaces/srv/Getposact "{id: 1, status: 'get_posact'}"
```

3.2.20 (20) Read Actual Hand Angle Value

```
ros2 service call /Getangleact service_interfaces/srv/Getangleact "{id: 1, status:
'get_angleact'}"
```

3.2.21 (21) Read Actual Force

```
ros2 service call /Getforceact service_interfaces/srv/Getforceact "{id: 1, status:
'get_forceact'}"
```

3.2.22 (22) Read Temperature Information

```
ros2 service call /Gettemp service_interfaces/srv/Gettemp "{id: 1, status: 'get_temp'}"
```

3.2.23 (23) Read Fault Information

```
ros2 service call /Geterror service_interfaces/srv/Geterror "{id: 1, status: 'get_error'}"
```

3.2.24 (24) Read Set Speed Value

```
ros2 service call /Getspeedset service_interfaces/srv/Getspeedset "{id: 1, status:
'get_speedset'}"
```

3.2.25 (25) Read Status Information

```
ros2 service call /Getstatus service_interfaces/srv/Getstatus "{id: 1, status: 'get_status'}"
```

3.2.26 (26) Execute Gesture Sequence

```
ros2 service call /Setgestureno service_interfaces/srv/Setgestureno "{gesture_no: 1, id: 1,
status: 'setgesture'}"
```

3.3 3) ROS Topic Usage Example: Real-time Tactile Sensor Data Reading

Open two new terminals, run `source install/setup.bash`, and then execute the following commands.

```
# Execute in the first terminal
ros2 run inspire_hand_modbus_ros2 handcontrol_topic_publisher_modbus.py

# Execute in the second terminal
ros2 run inspire_hand_modbus_ros2 handcontrol_topic_subscriber_modbus.py
```

In this example, the transmission frequency and current tactile sensor data of the entire hand are displayed in real-time in the terminal.

3.3.1 Node Launch

Execute the following command to publish topics for setting angle, speed, force threshold, and reading angle, tactile, force, and cylinder temperature.

```
ros2 run inspire_hand_modbus_ros2 inspire_hand_modbus_topic.py
```

3.3.2 Topic Publishing

- **Angle Setting:**

```
ros2 topic pub -1 /set_angle_data service_interfaces/msg/SetAngle1 "{finger_ids:
[1,2,3,4,5,6], angles: [1000,1000,1000,1000,1000,1000]}"
```

- **Angle Reading:**

```
ros2 topic echo /angle_data
```

3.4 4) ROS Service Usage Example

3.4.1 Service Call from Script

This is an example of calling the Setpos service included in service_interfaces/srv from a script. Open a new terminal, run `source install/setup.bash`, and then execute the following command.

```
ros2 run inspire_hand_modbus_ros2 hand_control_client_modbus_node
```

4. 4. Summary

This document explains the setup and usage of Inspire-Robots' multi-fingered hand in the ROS2 environment. The following points are covered:

- Environment setup and configuration procedures
- Hardware connection and network settings
- Hand control methods using ROS services
- Data acquisition methods using ROS topics
- Examples of service calls from scripts

For more detailed information and updates, please refer to the Inspire-Robots Official Website and TechShare-Inspire.