ROS2 Usage Instructions

1. The ros2_ws package supports the use of the dexterous hand and mechanical gripper from Inspire Robots Company on the ROS 2.0 platform.

2. To ensure the program runs properly, the ROS 2.0 environment needs to be configured, and the corresponding dependencies must be installed.

1) Create a new folder named ros2_ws, and within this folder, create a src folder.
2) Extract the three folders from the inspire_hand.zip archive and place them under the src folder.
3) Open a new terminal in ros2_ws, and run the command colcon build. If the build is successful, proceed to the next step.

3. Usage of the Dexterous Hand

1) Connect the dexterous hand to the computer and check the connection port name.

Terminal command: ls/dev/tty* (The port name should display as /dev/ttyUSB0, /dev/ttyUSB1, /dev/ttyUSB2, etc.). The package defaults to using the serial port /dev/ttyUSB0. If the port name does not match, you can modify the program accordingly.

Note: Before using it, you need to grant executable permissions to the serial port.

2) Below is the command to officially start the program. Use the launch command to start the program:

(1) If the serial port name is the default ttyUSB0. Terminal commands as follows:

```
cd ~/ros2_ws
colcon build
```

3) If the previous step is successful, you can proceed with the following commands:

Note: You need to open a new terminal and execute the following terminal commands:

```
cd ~/ros2_ws
source install/setup.bash
ros2 run inspire_hand
Hand_control_node
```

4) Open a new terminal under ros2_ws and first run the command:

```
source install/setup.bash
```

Then execute the following commands to control the dexterous hand:

(1)ros2 service call /Setpos service_interfaces/srv/Setpos "{pos0: num0,pos1: num2,pos2: num2,pos3: num3,pos4: num4,pos5: num5,hand_id: 1, status: 'set_pos'}"

Set the positions of the six actuators------Parameters num0-num5 range from 0-2000, and the default dexterous hand ID is 1.

```
(2)ros2 service call /Setangle service_interfaces/srv/Setangle "{angle0: num0,angle1: num2,angle2: num2,angle3: num3,angle4: num4,angle5: num5,hand_id: 1, status: 'set_angle'}"
```

Set the angles of the dexterous hand------Parameters num0-num5range from -1-1000

```
(3)ros2 service call /Setforce service_interfaces/srv/Setforce "{force0: num0,force1: num2,force2: num2,force3: num3,force4: num4,force5: num5,hand_id: 1, status: 'set_force'}"
```

Set the force control threshold------Parameters num0-num5range from -1000

```
(4)ros2 service call /Setspeed service_interfaces/srv/Setspeed "{speed0: num0,speed1: num2,speed2: num2,speed3: num3,speed4: num4,speed5: num5,hand_id: 1, status: 'set_speed'}"
```

Set the speed ------Parameters num0-num5range from -1000

```
(5)ros2 service call /Getposact service_interfaces/srv/Getposact "{hand_id: 1, status: 'get_posact'}"
```

Read the actual position values of the actuator.

```
(6)ros2 service call /Getangleact service_interfaces/srv/Getangleact "{hand_id: 1, status: 'get_angleact'}"
```

Read the actual angle values.

```
(7)ros2 service call /Getforceact service_interfaces/srv/Getforceact "{hand_id: 1, status: 'get_forceact'}"
```

Read the actual force.

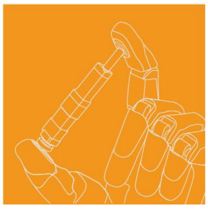
```
(8)ros2 service call /Getposset service_interfaces/srv/Getposset "{hand_id: 1, status: 'get_posset'}"
```

Read the set position values of the actuators.

```
(9)ros2 service call /Getangleset service_interfaces/srv/Getangleset "{hand_id: 1, status: 'get_angleset'}"
```

Read the set angle values.

```
(10)ros2 service call /Getforceset service_interfaces/srv/Getforceset "{hand_id: 1, status: 'get_forceset'}"
```



Read the set force control threshold.

```
(11)ros2 service call /Geterror service_interfaces/srv/Geterror "{hand_id: 1, status: 'get_error'}"
```

Read error information.

```
(12)ros2 service call /Getspeedset service_interfaces/srv/Getspeedset "{hand_id: 1, status: 'get_speedset'}"
```

Read the set speed.

```
(13)ros2 service call /Gettemp service_interfaces/srv/Gettemp "{hand_id: 1, status: 'get_temp'}"
```

Read temperature information.

```
(14)ros2 service call /Getcurrentact service_interfaces/srv/Getcurrentact "{hand_id: 1, status: 'get_currentact'}"
```

Read the current.

```
(15)ros2 service call /Setgestureno service_interfaces/srv/Setgestureno "{gesture_no: 1,hand_id: 1, status: 'set_gesture_no'}"
```

Gesture sequence, where the range of gesture_no is from 1 to 13.