

Compilation Environment

- Ubuntu 22.04
- ROS2 Humble

ROS 2 requires manual installation: It can be deployed via the one-click installation script below using wget. `wget http://fishros.com/install -O fishros && . fishros`

Sample Project Structure

```
lhandpro_ws                                # Project Workspace
├── src                                    # Source Code Directory
│   ├── lhandpro_description              # Includes URDF and mesh files
│   │   ├── .....
│   │   └── .....                        # Monitors real-time joint angles and publishes the
│   │       /joint_states topic
│   │   └── .....                        # Launch Script for rviz2, robot_state_publisher, and
│   └── lhandpro_state_publisher
│       ├── lhandpro_interfaces           # Define the Service Interface
│       ├── lhandpro_service              # The Service Execution Node hosts the mapped service and
│       │   controls the dexterous hand via the API
│       ├── sequence_demo_cpp             # C++ Example of Executing the Dexterous Hand's Motion
│       │   Sequence via the Service
│       │   ├── sequence_demo_py          # Python Example of Executing the Dexterous Hand's Motion
│       │   │   Sequence via the Service
│       │   └── ros2_lhandpro.conf        # Preconfigured Configuration Files
```

Compilation Steps

1. lhandpro_ws Compilation of the ROS Sample Project

```
cd lhandpro_ws
colcon build
source install/setup.bash
# Barring unforeseen issues, all components will compile successfully
```

2. Configure the Environment

```
# Copy the SO files to the system library directory to facilitate subsequent use
sudo cp src/lhandpro_service/thirdparty/lib/libLHandProLib.so /usr/local/lib/
# Refresh the Dynamic Linker Cache
sudo ldconfig
# Grant Raw Socket and Network Management Permissions
```

```
sudo setcap cap_net_raw,cap_net_admin+ep
./install/lhandpro_service/lib/lhandpro_service/lhandpro_service
```

handpro_ws/src/ros2_lhandpro.conf and edit the .conf file

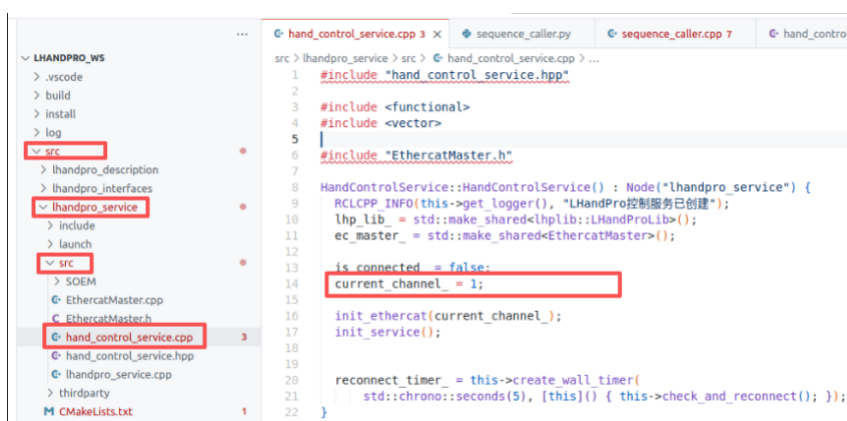
```
# Modify the content of the ros2_lhandpro.conf file to the actual local path
/home/plf/Project/RosProject/lhandpro_ws/install/lhandpro_service/lib
/home/plf/Project/RosProject/lhandpro_ws/install/lhandpro_interfaces/lib
/home/plf/Project/RosProject/lhandpro_ws/install/lhandpro_description/lib
/home/plf/Project/RosProject/lhandpro_ws/install/sequence_demo_cpp/lib
/home/plf/Project/RosProject/lhandpro_ws/install/sequence_demo_py/lib
```

Configure the Execution Environment

```
# Copy the .conf file to the system directory
sudo cp src/ros2_lhandpro.conf /etc/ld.so.conf.d/
# Refresh the Dynamic Linker Cache
sudo ldconfig
```

3. Execute the Control Service

```
ros2 launch lhandpro_service lhandpro.launch.py
# If there are no configuration errors, the console will normally output the
network interface list.
# It will then automatically connect to the default network interface specified in
the code
# Please ensure it matches the actual network interface used for EtherCAT
```



Once you modify the designated location of the default network interface specified in the code, you will need to re-run colcon build

4. Execute the Sample Control

```
# Once launched, it will automatically execute the predefined motion sequence
ros2 launch sequence_demo_py sequence.launch.py
# Or execute the C++-version of the sample program
```

```
ros2 launch sequence_demo_cpp sequence.launch.py
```

5. Execute the RViz2 Simulation Visualization

```
ros2 launch lhandpro_description display_lhandpro.launch.py
# Running this script will launch the robot_state_publisher node
# Launch the lhandpro_state_publisher node, which is used to monitor angle changes
# via the service and publish data to the /joint_states topic for RViz2 to update its
# visualization
# Launch the RViz2 node and load the predefined RViz configuration file
# If the loading completes successfully, the dexterous hand model will be viewable
# in RViz2
# At this point, running the control sample from Step 4 will display the physical
# dexterous hand in motion, with the RViz2 model moving synchronously
```