

# ROS2 Modbusの使用方法

## 目次

- 1 概要
- 2 環境設定
  - 2.1 ROS2 Humble環境のインストール
  - 2.2 modbusライブラリのインストール
  - 2.3 Catkinワークスペースの作成
  - 2.4 パッケージの展開
  - 2.5 パッケージの再コンパイル
- 3 5指ハンドの使用方法
  - 3.1 ハード接続
  - 3.2 inspire\_hand\_modbus\_ros2 パッケージの実行
    - 3.2.1 IDの設定
    - 3.2.2 ボーレートの設定
    - 3.2.3 6軸ドライバの位置設定
    - 3.2.4 速度の設定
    - 3.2.5 5指ハンドの角度設定
    - 3.2.6 力制御の閾値設定
    - 3.2.7 電流の閾値設定
    - 3.2.8 電源投入時の速度設定（再起動後に有効）
    - 3.2.9 電源投入時の力制御閾値設定（再起動後に有効）
    - 3.2.10 電源投入時の電流閾値設定（再起動後に有効）
    - 3.2.11 力覚センサーの校正
    - 3.2.12 エラーのクリア
    - 3.2.13 工場出荷時設定へのリセット
    - 3.2.14 パラメータのFLASHメモリへの保存
    - 3.2.15 設定されたアクチュエータの位置値の読み取り
    - 3.2.16 設定されたハンドの角度値の読み取り
    - 3.2.17 設定された力制御の閾値の読み取り
    - 3.2.18 現在の電流値の読み取り
    - 3.2.19 アクチュエータの実際の位置値の読み取り
    - 3.2.20 実際のハンドの角度値の読み取り
    - 3.2.21 実際に受けている力の読み取り
    - 3.2.22 温度情報の読み取り
    - 3.2.23 故障情報の読み取り
    - 3.2.24 設定された速度値の読み取り
    - 3.2.25 状態情報の読み取り
    - 3.2.26 ジェスチャーシーケンスの実行
  - 3.3 ROSトピックの使用例: 触覚センサーデータのリアルタイム読み取り
    - 3.3.1 ノードの立ち上がり
    - 3.3.2 トピックの発行
  - 3.4 ROSサービスの使用例
    - 3.4.1 スクリプトからのサービス呼び出し
- 4 まとめ

Generated: 2025/6/19

## 著作権表示 (Copyright):

© Inspire-Robots. All Rights Reserved.

オリジナルのスクリプトは Inspire-Robots 社によって提供されました。

翻訳およびフォーマットは TechShare 株式会社によって行われました。

## 1. 概要

inspire\_hand パッケージ (ROS2版) は、Inspire-Robots社の多指ハンドおよびロボットグリッパーをROSプラットフォーム上で使用するためのものです。

現時点では、Ubuntu 22.04 ROS2 Humble 環境でのみ検証済みです。他のROS環境については、今後の開発をお待ちください。

## 2. 環境設定

プログラムを正常に実行するためには、以下の環境設定が必要です（初回のみ。一度設定すれば再度行う必要はありません）。

### 2.1 ROS2 Humble環境のインストール

インストール手順の詳細は、以下のリンクを参照してください。

[ROS2 Humble Install](#)

### 2.2 modbusライブラリのインストール

ターミナルで以下のコマンドを実行します。

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

**注意：**他に不足している依存関係がある場合は、cmakeコンパイル時のターミナルのエラーメッセージに従って、不足している項目をダウンロードしてください。

### 2.3 Catkinワークスペースの作成

ターミナルで以下のコマンドを順に実行します。

```
mkdir -p ~/inspire_hand_ws/src
cd ~/inspire_hand_ws
colcon build
source install/setup.bash # このコマンドは、ROSのインストールディレクトリを見つけるため、新しいターミナルを開くたびに実行してください。
```

## 2.4 パッケージの展開

inspire\_hand\_ros2.zip を `~/inspire_hand_ws/src` ディレクトリに配置し、解凍します。

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

解凍後、inspire\_hand\_modbus\_ros2 と service\_interfaces の2つのフォルダを `~/inspire_hand_ws/src` に移動し、元の inspire\_hand\_ros2 フォルダは削除してください。

## 2.5 パッケージの再コンパイル

ターミナルで以下のコマンドを実行します。

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

**注意：** 環境変数の衝突を避けるため、`sudo gedit ~/.bashrc` を使ってbashファイルに過剰なsourceコマンドを追加することは極力避けてください。また、"service\_interfaces"のような重複したパッケージ名も避けるようにしてください。これらの操作によって、メッセージの参照エラーやノード起動エラーの原因となる可能性があります。

## 3. 5指ハンドの使用法

### 3.1 ハード接続

Inspire HandとホストPCをLANケーブルで接続します。PCのIPv4設定を以下のように変更してください。

設定	値
IPアドレス	192.168.11.222
サブネットマスク	255.255.255.0

ターミナルで以下のコマンドを実行し、データが返ってくれば接続は成功です。

```
ping 192.168.11.210
```

応答がない場合は、ケーブルの接続を確認してください。

## 3.2 inspire\_hand\_modbus\_ros2 パッケージの実行

新しいターミナルを開き、まず以下のコマンドを実行してください。

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

以下に、サービスコールを使用した各種操作のコマンド例を示します。

### 3.2.1 IDの設定

id の範囲: 1-254

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

### 3.2.2 ボーレートの設定

redu\_ratio の範囲: 0-4

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

### 3.2.3 6軸ドライバの位置設定

pos の範囲: 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 速度の設定

speed の範囲: 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指ハンドの角度設定

angle の範囲: 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 力制御の閾値設定

force の範囲: 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 電流の閾値設定

current の範囲: 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 電源投入時の速度設定（再起動後に有効）

speed の範囲: 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_defaultspeak'}"
```

### 3.2.9 電源投入時の力制御閾値設定（再起動後に有効）

force の範囲: 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 電源投入時の電流閾値設定（再起動後に有効）

current の範囲: 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 力覚センサーの校正

このコマンドは2回実行する必要があります。実行後、ハンドは完全に開き、その後、力覚センサーの校正が行われます。

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

### 3.2.12 エラーのクリア

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

### 3.2.13 工場出荷時設定へのリセット

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

### 3.2.14 パラメータのFLASHメモリへの保存

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

### 3.2.15 設定されたアクチュエータの位置値の読み取り

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

### 3.2.16 設定されたハンドの角度値の読み取り



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

### 3.2.17 設定された力制御の閾値の読み取り

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

### 3.2.18 現在の電流値の読み取り

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

### 3.2.19 アクチュエータの実際の位置値の読み取り

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

### 3.2.20 実際のハンドの角度値の読み取り

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

### 3.2.21 実際に受けている力の読み取り

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

### 3.2.22 温度情報の読み取り

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

### 3.2.23 故障情報の読み取り

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

### 3.2.24 設定された速度値の読み取り

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

### 3.2.25 状態情報の読み取り

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

### 3.2.26 ジェスチャーシーケンスの実行

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

### 3.3 ROSトピックの使用例: 触覚センサーデータのリアルタイム読み取り

新たな2つのターミナルを開き、`source install/setup.bash` を実行してから、以下のコマンドを実行します。

```
# 1つ目のターミナルで実行
ros2 run inspire_hand_modbus_ros2 handcontrol_topic_publisher_modbus.py

# 2つ目のターミナルで実行
ros2 run inspire_hand_modbus_ros2 handcontrol_topic_subscriber_modbus.py
```

この例では、送信周波数と、ハンド全体の現在の触覚センサーデータがターミナルにリアルタイムで表示されます。

#### 3.3.1 ノードの立ち上がり

以下のコマンドを実行すると、角度、速度、力閾値の設定、および角度、触覚、力、シリンダー温度の読み取りを行うトピックが発行されます。

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

#### 3.3.2 トピックの発行

- 角度の設定:

```
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]}"
```

- 角度の読み取り:

```
ros2 topic echo /angle_data
```

## 3.4 ROSサービスの使用例

### 3.4.1 スクリプトからのサービス呼び出し

`service_interfaces/srv` に含まれる Setpos サービスをスクリプトから呼び出す例です。新しいターミナルを開き、`source install/setup.bash` を実行してから、以下のコマンドを実行します。

```
ros2 run inspire_hand_modbus_ros2 hand_control_client_modbus_node
```

## 4. まとめ

このドキュメントでは、ROS2環境におけるInspire-Robots社の多指ハンドの設定方法と使用方法を解説しました。以下の点について説明しています：

- 環境設定とセットアップ手順
- ハードウェア接続とネットワーク設定
- ROSサービスを使用したハンドの制御方法
- ROSトピックを使用したハンドのデータ取得方法
- スクリプトからのサービス呼び出し例

詳細な情報や更新については、[Inspire-Robots公式サイト](#)および[TechShare-Inspire](#)を参照してください。