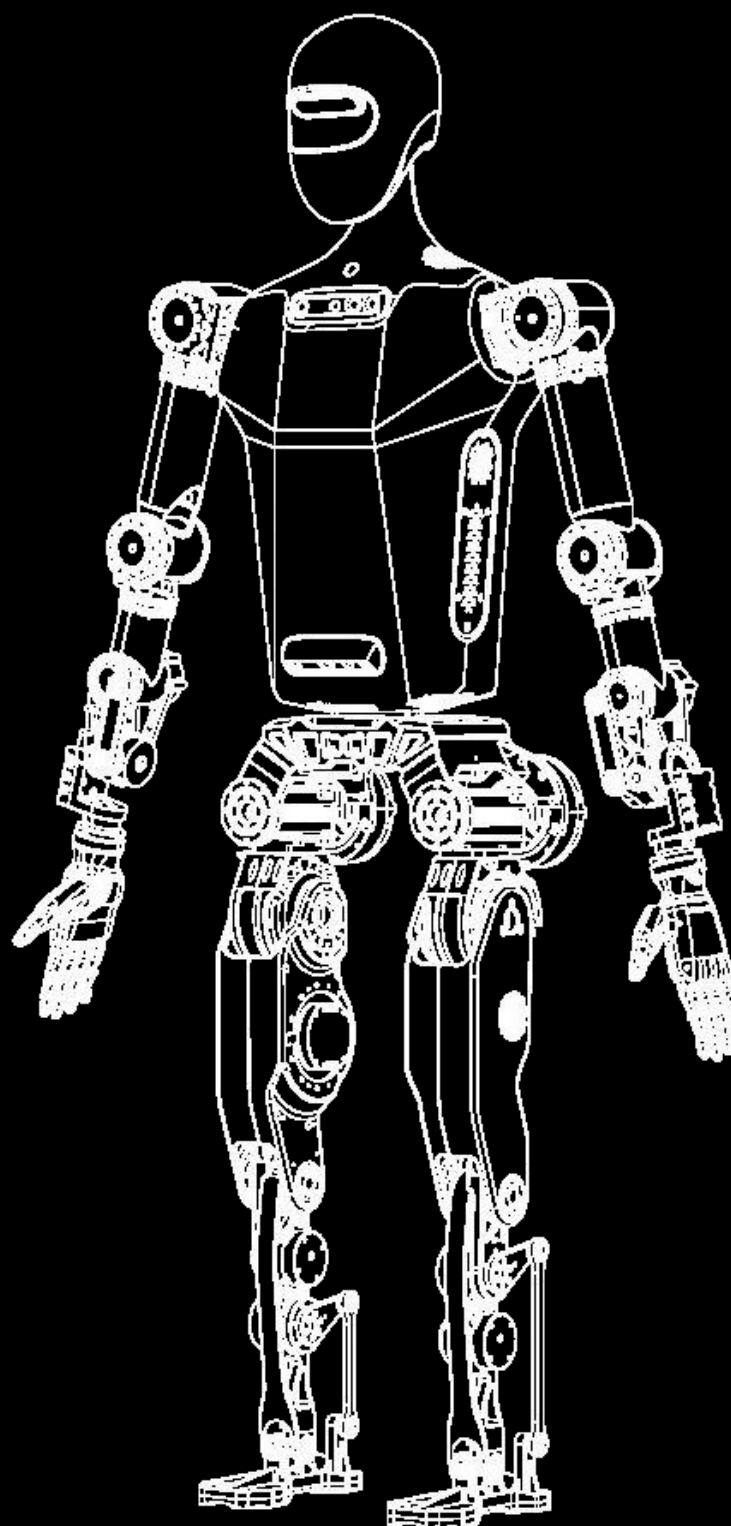




HUMANOID

# 天工通用人形机器人(Pro)

## SDK 二次开发文档



本文档内容可能会因产品更新而有所变动，请访问

<https://zitd5je6f7j.feishu.cn/wiki/CCLvwBgP5iedKAKnt8tc5nadnKc> 获取最新版本。

# 1. 关于天工

完整版天工整机分为上半身和下半身，具备多个自由度。头部拥有 3 个自由度。单手臂拥有 7 个自由度，包括肩关节、肘关节和腕关节。单灵巧手拥有自由度 6 个。腰部有 1 个自由度。单腿拥有 6 个自由度，包括髋关节、膝关节和踝关节。整机共有 42 个自由度，由 42 个一体化关节电机组成，使得机器人能够实现精确的运动和姿态控制。

## 1.1. 总体规格参数

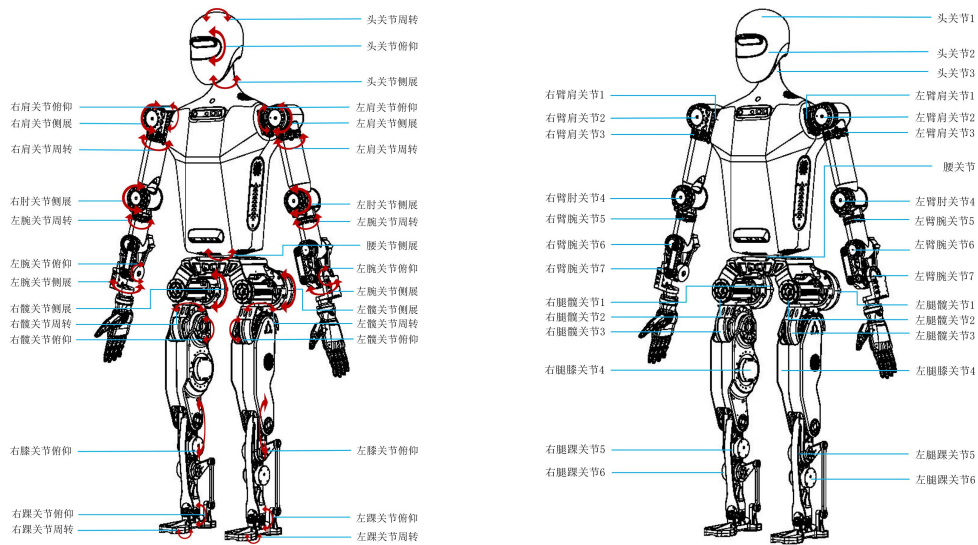
项目	规格参数
产品高度	1630mm
产品净重	56kg
产品颜色	黑色
产品材料	铝合金架构，工程塑料合金外壳
电池	电池类型：三元锂； 容量：15Ah； 电压：48V
充电时间	4 小时
语音模块	MIC 阵列：线性 MIC*4； 扬声器*2； 声卡*1； 720P RGB 单目相机*1
摄像头	RGBD 摄像头*4（头部，胸腔，腰部，背部）
传感器	高精度 IMU； 高精度六维力
自由度	42 个一体化关节（头 3 个，臂 7 个*2，手 6 个*2，腰 1 个，腿 6 个*2）

本文档内容可能会因产品更新而有所变动，请访问

<https://zitd5je6f7j.feishu.cn/wiki/CCLvwBgP5iedKAKNt8tc5nadnKc> 获取最新版本。

运控计算单元	型号：Intel x86； CPU 10 核 12 线程； 主频最高达 4.7GHz； 内存：16GB； SSD：256GB
开发计算单元	型号：Orin AGX 64 *2 算力：275 TOPS *2
系统平台/软件	操作系统：Ubuntu22.04.4 LTS； 中间件：ROS 1
外部通讯能力	WIFI6, Ethernet, Bluetooth；
内部通信网络	全身 CAN/EtherCAT 网络：500~1kHz； 灵巧手：20Hz； 机械臂：≥200Hz； IMU：>400Hz； 相机（4 个 RGBD）：30Hz

## 1.2. 关节参数



所属部位	名称	运动范围 (°)	额定转速 (rpm)	最大转速 (rpm)	额定力矩 (Nm)	峰值力矩 (Nm)	备注
头部	头关节侧展 Head Roll	-90°~+90°	30	40	1.6	3.5	/
	头关节俯仰 Head Pitch	-30°~+20°	30	40	1.6	3.5	/
	头关节周转 Head Yaw	-26°~+26°	30	40	1.6	3.5	/
腰部	腰关节侧展 Waist Roll	-90°~+90°	/	30	31	143	/
臂	肩关节俯仰 Shoulder Pitch	-170°~+45°	/	40	4.1	140	左臂向后方为正
	肩关节侧展 Shoulder Roll	0°~+100°	/	40	4.1	140	左臂向外侧为正

	肩关节周转 Shoulder Yaw	-130°~+130°	/	54	3.2	80	左臂向外转为正
	肘关节侧展 Elbow Roll	-130°~+0°	/	54	3.2	80	左小臂向外侧为正
	腕关节周转 Wrist Yaw	-130°~+130°	/	40	2.4	40	左手向外转为正
	腕关节俯仰 Wrist Pitch	-75°~+75°	/	40	2.4	40	左手向后方为正
	腕关节侧展 Wrist Roll	-75°~+75°	/	40	2.4	40	左手向外侧为正

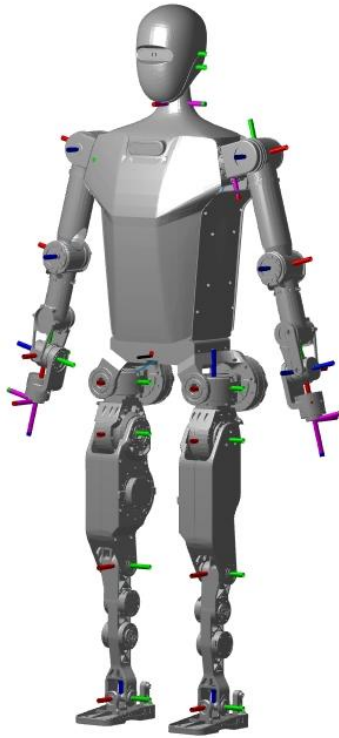
注：右臂关节的运动范围为左臂的镜像，即右肩关节俯仰运动范围为-45°~+170°，依此类推。

腿部	髋关节周转 Hip Yaw	-45°~+45°	121	140	50	150	/
	髋关节俯仰 Hip Pitch	-90°~+45°	121	140	50	150	/
	髋关节侧展 Hip Roll	-60°~+60°	142	150	30	94	/
	膝关节俯仰 Knee Pitch	0°~+140°	121	140	50	150	/
	踝关节俯仰 Ankle Pitch	-70°~+30°	78	85	12	36	/
	踝关节周转 Ankle Yaw	-25°~+25°	78	85	12	36	/

角度	运动范围 (°)	图例说明
小拇指 无名指 中指 食指	+19°~+176.7°	
大拇指弯曲角度	-13°~+53.6°	
大拇指旋转角度	+90°~+165°	

### 1.3. 坐标系，关节旋转轴与关节零点

当各个关节均为零度时，各坐标系如下图。红色为 x 轴，绿色为 y 轴，蓝色为 z 轴。



## 2. 快速操作指南

1. 确认机器人悬吊于保护支架之上；
2. 将遥控器上的电源键先短按后长按，打开遥控器；
3. 开机上电：
  - (1) 按下电池开关键；
  - (2) 按下主控开关键；
  - (3) 顺时针旋转急停按钮直到弹出。
4. 主控板登录和连接：
  - **配置 Wi-Fi 步骤：**
    - (1) 拆开机器人背板的六个螺丝，取下背板；
    - (2) 用以太网线连接机器人背后的调试以太网接口和用户电脑端；
    - (3) 用户电脑配置以太网端口地址为 192.168.41.xx/255.255.255.0；
    - (4) 打开终端中输入 `ssh ubuntu@192.168.41.x` 登录需要配置的主控板（运控 x86、

大模型 orin、导航 orin 分别为 192.168.41.1、192.168.41.2、192.168.41.3) ;

(5) 在上一步的终端中继续依次输入如下命令配置 IP:

```
sudo nmcli device wifi list #查看 Wi-Fi 热点的 SSID
sudo nmcli device wifi connect 'Wi-Fi SSID' password 'Wi-Fi 密码' #连接 Wi-Fi
iwconfig #检查是否连接成功
```

(6) Wi-Fi 配置完成, 拔掉网线。

● **Wi-Fi 配置完成的前提下, 每次机器人无需再配置 Wi-Fi, 直接用 Wi-Fi 连接的步骤如下:**

(1) 输入 `ifconfig`, 查看给主控板配置的 IP 地址;

(2) 输入 `ssh ubuntu@x.x.x.x` 登录连接主控板服务器, 其中 `x.x.x.x` 为上一步获取的 IP 地址。随后可进行下一步程序启动。

5. 程序启动:

(1) 首先, 打开第一个终端, 依次输入如下命令启动本体驱动程序:

```
cd ros_ws #切换到 ros_ws 目录
sudo -s #切换到 root 权限
(输入密码)
source install_isolated/setup.bash #执行环境变量脚本
roslaunch body_control motion_pro.launch #启动本体驱动程序
```

(2) 其次其次, 打开第二个终端, 依次输入如下命令启动遥控器通信节点:

```
cd ros_ws #切换到 ros_ws 目录
python3 sbus_3_mod.py #启动遥控器通信节点
```

(3) 最后, 打开第三个终端, 依次输入如下命令:

```
cd ros_ws #切换到 ros_ws 目录
source install_isolated/setup.bash #执行环境变量脚本
```

(4) 继续在打开的第三个终端中, 输入如下命令启动运控驱动程序:

◆ 如果想使用强化学习运控模式, 则启动:

```
roslaunch rl_control_new rl.launch
```



```
(enter 1:) 1 #确认所有 current pos 的绝对值都小于 1, 才能输入 1 启动, 否则  
结束程序检查电机是否位置偏差过大
```

◆ 如果想使用传统运控模式, 则启动:

```
roslaunch motion_control motion.launch
```

```
(enter 1:) 1 #确认所有 current pos 的绝对值都小于 1, 才能输入 1 启动, 否则  
结束程序检查电机是否位置偏差过大
```

此时可以开始使用 SDK 进行开发调试。

## 6. 关机:

- (1) 确认机器人已停止并返回站立状态;
- (2) 按下遥控器上的“C”键使机器人僵停;
- (3) 将机器人固定在支架上, 并向上吊起;
- (4) 将程序启动中打开的运控驱动程序、遥控器通信节点和本体驱动程序按照该顺序依次按 control+C 停止;
- (5) 待上述三个程序都成功停止后, 在本体驱动程序的终端界面里输入命令 `poweroff;`
- (6) 按下急停按钮;
- (7) 按下主控开关键;
- (8) 按下电池开关键;
- (9) 将遥控器上的电源键先短按后长按, 关闭遥控器。

# 3. 应用开发

## 3.1. SDK 概述

完整版天工 SDK 由北京人形机器人创新中心开发, 提供了丰富的接口, 涵盖头部、臂部、手部、腰部和腿部的电机控制, 以及六维力、IMU (惯性测量单元) 和相机的使用, 用于编写和部署机器人应用程序, 旨在帮助开发人员快速灵活地构建自己的应用程序来精确控制和使用机器人, 以满足在不同应用场景下的需求。

## 3.2. 环境依赖

- 系统环境

为了确保最佳的开发体验和兼容性, 建议在 Ubuntu 20.04 系统下进行开发。当前暂不支持在 Mac 和 Windows 系统下开发。

- 框架环境

ROS1

## 4. 全局值说明 (头、腿)

- 消息名称: `MotorName`
- 数据定义位置: `~/rosws/src/bodyctrl_msgs/msg/MotorName.msg`
- 说明: 完整版天工头部和腿部的电机的命名 ID 定义如下。

```
uint16 MOTOR_LEG_LEFT_1 = 1
uint16 MOTOR_LEG_LEFT_2 = 2
uint16 MOTOR_LEG_LEFT_3 = 3
uint16 MOTOR_LEG_LEFT_4 = 4
uint16 MOTOR_LEG_LEFT_5 = 5
uint16 MOTOR_LEG_LEFT_6 = 6
uint16 MOTOR_LEG_RIGHT_1 = 7
uint16 MOTOR_LEG_RIGHT_2 = 8
uint16 MOTOR_LEG_RIGHT_3 = 9
uint16 MOTOR_LEG_RIGHT_4 = 10
uint16 MOTOR_LEG_RIGHT_5 = 11
uint16 MOTOR_LEG_RIGHT_6 = 12
uint16 MOTOR_HEAD_TOP = 19
uint16 MOTOR_HEAD_LEFT = 20
uint16 MOTOR_HEAD_RIGHT = 21
```

注: 关节电机 ID 对应的具体关节请参考 [1.2 关节参数](#)。

## 5. 接口说明

### 5.1. IMU

#### 5.1.1. 状态获取接口

##### 1. 获取 IMU 传感器信息

- 说明：获取 IMU 传感器数据信息，其中包含加速度、角速度、位姿四元素和欧拉角。
- 控制方式：topic
- 话题名称：/BodyControl/imu
- 数据定义位置：~/rosws/src/bodyctrl\_msgs/msg/Imu.msg
- 数据格式：

```
std_msgs/Header header
geometry_msgs/Quaternion orientation
geometry_msgs/Vector3 angular_velocity
geometry_msgs/Vector3 linear_acceleration
bodyctrl_msgs/Euler euler

float64[3] angular_velocity_covariance
float64[3] orientation_covariance
float64[3] linear_acceleration_covariance
```

### 5.2. 头部电机

#### 5.2.1. 状态获取接口

##### 1. 获取头部电机信息

- 说明：获取头部电机的状态信息，其中包含电机的当前位置、速度、电流和温度。
- 控制方式：topic
- 话题名称：/BodyControl/ey/status
- 数据定义位置：~/rosws/src/bodyctrl\_msgs/msg/MotorStatusMsg.msg
- 数据格式：

```
std_msgs/Header header
MotorStatus[] status
uint16 name      # MotorName
float32 pos      # rad
float32 speed    # rad
float32 current  # A
float32 temperature
```

## 5.2.2. 控制接口

### 1. 控制头部电机位置

- 说明：头部电机的位置控制接口，需要提供期望位置、期望速度、最大电流。
- 控制方式：topic
- 话题名称：/BodyControl/ey/set\_pos
- 数据定义位置：~/ros\_ws/src/bodyctrl\_msgs/msg/CmdSetMotorPosition.msg
- 数据格式：

```
std_msgs/Header header
SetMotorPosition[] cmds
uint16 name # MotorName
float32 pos # rad
float32 spd # rpm
float32 cur # A
```

## 5.3. 腰部电机

### 5.3.1. 状态获取接口

#### 1. 获取腰部电机信息

- 说明：获取腰部电机的状态信息，其中包含电机的当前位置、速度、电流和温度。
- 控制方式：topic
- 话题名称：/BodyControl/ze/status
- 数据定义位置：~/ros\_ws/src/bodyctrl\_msgs/msg/MotorStatusMsg.msg

- 数据格式:

```
std_msgs/Header header
MotorStatus[] status
uint16 name      # MotorName
float32 pos      # rad
float32 speed    # rad
float32 current  # A
float32 temperature
```

## 5.3.2. 控制接口

### 1. 控制腰部电机位置

- 说明: 腰部电机的位置控制接口, 需要提供期望位置、期望速度、最大电流。
- 控制方式: topic
- 话题名称: `/BodyControl/ze/set_pos`
- 数据定义位置: `~/ros_ws/src/bodyctrl_msgs/msg/CmdSetMotorPosition.msg`
- 数据格式:

```
std_msgs/Header header
SetMotorPosition[] cmds
uint16 name  # MotorName
float32 pos  # rad
float32 spd  # rpm
float32 cur  # A
```

## 5.4. 腿部电机

### 5.4.1. 状态获取接口

#### 1. 获取腿部电机信息

- 说明: 获取腿部电机的状态信息, 包含电机的当前位置、速度、电流和温度。
- 控制方式: topic
- 话题名称: `/BodyControl/motor_state`

- 数据定义位置: `~/ros_ws/src/bodyctrl_msgs/msg/MotorStatus.msg`
- 数据格式:

```
std_msgs/Header header
MotorStatus[] status
uint16 name      # MotorName
float32 pos      # rad
float32 speed    # rad
float32 current  # A
float32 temperature
```

### 5.4.2. 控制接口

**注:** 完整版天工的腿部支持三种电机控制模式: 力位混合控制、速度控制、位置控制。每种控制模式针对多个电机均**独立运行**, 互不干扰。但对于单个电机而言, 这些模式是**互斥的**, 后一个模式的指令会覆盖前一个模式的指令。

#### 1. 力位混合控制腿部电机

- 说明: 腿部电机的力位混合控制接口, 需要提供 `kp`、`kd`、期望位置、最大电流和前馈扭矩。
- 控制方式: `topic`
- 话题名称: `/BodyControl/motor_ctrl`
- 数据定义位置: `~/ros_ws/src/bodyctrl_msgs/msg/CmdSetMotorCtrl.msg`
- 数据格式:

```
std_msgs/Header header
MotorCtrl[] cmds
uint16 name  # MotorName
float32 kp
float32 kd
float32 pos  # rad
float32 spd  # rad
```

```
float32 tor # Nm
```

## 2. 控制腿部电机速度

- 说明：腿部电机的速度控制接口，需要提供期望速度和最大电流。
- 话题名称：/BodyControl/set\_motor\_speed
- 数据定义位置：~/rosws/src/bodyctrl\_msgs/msg/CmdSetMotorSpeed.msg
- 数据格式：

```
std_msgs/Header header
SetMotorSpeed[] cmds
uint16 name # MotorName
float32 spd # rpm
float32 cur # A
```

## 3. 控制腿部电机位置

- 说明：腿部电机的位置控制接口，需要提供期望位置、期望速度和最大电流。
- 话题名称：/BodyControl/set\_motor\_position
- 数据定义位置：~/rosws/src/bodyctrl\_msgs/msg/CmdSetMotorPosition.msg
- 数据格式：

```
std_msgs/Header header
SetMotorPosition[] cmds
uint16 name # MotorName
float32 pos # rad
float32 spd # rpm
float32 cur # A
```

## 5.5. 臂部电机

### 5.5.1. 状态获取接口

#### 1. 上报臂部电机信息

- 说明：上报臂部电机的状态信息，即从臂部关节获取数据，通过话题发布出来。

- 控制方式: topic
- 话题名称:

```
/human_arm_state_left
/human_arm_state_right
```

- 消息类型: `sensor_msgs::JointState`
- 成员:

```
Header header uint32 seq #序列号
Header header time stamp #时间戳
Header header string frame_id #帧 id

string[7] name # "1" ~ "7"

float64[7] position
float64[7] velocity
float64[7] effort
```

## 5.5.2. 控制接口

### 1. 控制臂部电机

- 说明: 接收用户的插补指令, 下发给臂部关节, 其中包含位置、速度和力矩指令。
- 控制方式: topic
- 话题名称:

```
/human_arm_ctrl_left
/human_arm_ctrl_right
```

- 消息类型: `sensor_msgs::JointState`
- 成员:

```
Header header uint32 seq #序列号
Header header time stamp #时间戳
```



Header header string frame\_id #帧 id

string[7] name # "1" ~ "7"

float64[7] position

float64[7] velocity

float64[7] effort

## 5.6. 手部电机

### 5.6.1. 状态获取接口

#### 1. 获取实际角度

- 说明：获取手爪的实际角度。其中响应值表示关节角度的比例，范围为 0 到 1，0 表示完全弯曲，1 表示完全伸展。
- 控制方式：service
- 服务名称：

/inspire\_hand/get\_angle/left\_hand

/inspire\_hand/get\_angle/right\_hand

- request:

null

- response:

float32[6] curangleRatio

#### 2. 获取实际力矩

- 说明：获取手爪的实际力矩。其中响应值表示力矩的比例，范围为 0 到 1，0 代表 0g，1 代表 1000g。
- 控制方式：service
- 服务名称：

```
/inspire_hand/get_force/left_hand  
/inspire_hand/get_force/right_hand
```

- request:

```
null
```

- response:

```
float32[6] curforceRatio
```

### 3. 获取状态

- 说明：获取手爪的当前状态。其中不同响应值分别代表的手爪状态如下表所示。

response	手爪状态
0	正在松开
1	正在抓取
2	位置到位停止
3	力控到位停止
5	电流保护停止
6	电缸堵转停止
7	电缸故障停止
0xff	空闲

- 控制方式：service
- 服务名称：

```
/inspire_hand/get_status/left_hand  
/inspire_hand/get_status/right_hand
```

- request:

```
null
```

- response:

uint32[6] statusvalue

#### 4. 获取错误

- 说明：获取手爪的错误。其中不同响应值分别代表的具体错误如下表所示。

response	错误
Bit0	堵转故障
Bit1	过温故障
Bit2	过流故障
Bit3	电机异常
Bit4	通讯故障

- 控制方式：service
- 服务名称：

/inspire\_hand/get\_error/left\_hand  
/inspire\_hand/get\_error/right\_hand

- request:

null

- response:

uint32[6] errorvalue

### 5.6.2. 控制接口

#### 1. 清除错误

- 说明：清除手爪的错误。
- 控制方式：service
- 服务名称：

/inspire\_hand/set\_clear\_error/left\_hand

```
/inspire_hand/set_clear_error/right_hand
```

- request:

```
null
```

- response:

```
bool setclear_error_accepted #是否设置成功
```

## 2. 设置角度

- 说明：设置手爪的角度。其中请求值表示手爪各关节运动角度的比例，范围为 0 到 1，0 表示完全弯曲，1 表示完全伸展。手爪各关节实际运动范围请参考 [1.2 关节参数](#) 中的表格。
- 控制方式：service
- 服务名称：

```
/inspire_hand/set_angle/left_hand
```

```
/inspire_hand/set_angle/right_hand
```

- request:

```
float32 angle0Ratio #小拇指
```

```
float32 angle1Ratio #无名指
```

```
float32 angle2Ratio #中指
```

```
float32 angle3Ratio #食指
```

```
float32 angle4Ratio #大拇指弯曲
```

```
float32 angle5Ratio #大拇指旋转
```

- response:

```
bool angle_accepted #是否设置成功
```

## 3. 灵活设置角度

- 说明：灵活设置手爪的角度。其中请求值表示手爪各关节运动角度的比例，范围为 0 到 1，0 表示完全弯曲，1 表示完全伸展。手爪各关节实际运动范围请参考 [1.2 关节参数](#) 中的表格。和设置角度接口不同，该接口不需要每次都填入所有关节的角度，最少可

只填 1 个。并且 name 字段和 angleRatio 一一对应, name 字段的值为 “1” 至 “6”。

- 控制方式: service
- 服务名称:

```
/inspire_hand/set_angle_flexible/left_hand  
/inspire_hand/set_angle_flexible/right_hand
```

- request:

```
float32 angle0Ratio #小拇指  
float32 angle1Ratio #无名指  
float32 angle2Ratio #中指  
float32 angle3Ratio #食指  
float32 angle4Ratio #大拇指弯曲  
float32 angle5Ratio #大拇指旋转
```

- response:

```
bool angle_accepted #是否设置成功
```

#### 4. 设置力矩

- 说明: 设置手爪的实际力矩。其中请求值表示力矩的比例, 范围为 0 到 1, 0 代表 0g, 1 代表 1000g。
- 控制方式: service
- 服务名称:

```
/inspire_hand/set_force/left_hand  
/inspire_hand/set_force/right_hand
```

- request:

```
float32 force0Ratio #小拇指  
float32 force1Ratio #无名指  
float32 force2Ratio #中指  
float32 force3Ratio #食指
```

```
float32 force4Ratio #大拇指弯曲
```

```
float32 force5Ratio #大拇指旋转
```

- response:

```
bool force_accepted #是否设置成功
```

## 5. 设置速度

- 说明：设置手爪的实际速度。其中请求值表示速度的比例，范围为 0 到 1，1 表示从最大角度到最小角度的运动时间为 800ms，0.5 表示 1600ms，0.25 表示 3200ms，依此类推。
- 控制方式：service
- 服务名称：

```
/inspire_hand/set_speed/left_hand
```

```
/inspire_hand/set_speed/right_hand
```

- request:

```
float32 speed0Ratio #小拇指
```

```
float32 speed1Ratio #无名指
```

```
float32 speed2Ratio #中指
```

```
float32 speed3Ratio #食指
```

```
float32 speed4Ratio #大拇指弯曲
```

```
float32 speed5Ratio #大拇指旋转
```

- response:

```
bool speed_accepted #是否设置成功
```

## 5.7. 六维力

### 5.7.1. 状态获取接口

#### 1. 六维力信息

- 说明：获取六维力当前的信息。其中上报频率为 200Hz。

- 控制方式: topic
- 话题名称:

```
/human_arm_6dof_left
/human_arm_6dof_right
```

- 类型: geometry\_msgs::WrenchStamped
- 成员:

```
Header header uint32 seq #序列号
Header header time stamp #时间戳
Header header string frame_id #帧 id

Wrench wrench Vector3 force #3 个方向的力
Wrench wrench Vector3 torque #3 个方向的力矩
```

## 5.8. 相机

天工头部、胸部、腰部和背部各配置了一个深度相机，其不仅提供精准的深度数据流，还支持输出高质量的彩色数据流和红外数据流。其中头部配置的是奥比中光 335，用于大模型；其余部位配置的是奥比中光 335L，用于导航。

### 5.8.1. 奥比中光 335L 相机\*3 接口-导航

#### 1. 胸部相机接口

- 话题名称:

```
/ob_camera_front/accel/imu_info
/ob_camera_front/color/camera_info
/ob_camera_front/color/image_raw
/ob_camera_front/color/image_raw/compressed
/ob_camera_front/color/image_raw/compressed/parameter_descriptions
/ob_camera_front/color/image_raw/compressed/parameter_updates
/ob_camera_front/color/image_raw/compressedDepth
```

/ob\_camera\_front/color/image\_raw/compressedDepth/parameter\_descriptions  
/ob\_camera\_front/color/image\_raw/compressedDepth/parameter\_updates  
/ob\_camera\_front/color/image\_raw/theora  
/ob\_camera\_front/color/image\_raw/theora/parameter\_descriptions  
/ob\_camera\_front/color/image\_raw/theora/parameter\_updates  
/ob\_camera\_front/color/metadata  
/ob\_camera\_front/depth/camera\_info  
/ob\_camera\_front/depth/image\_raw  
/ob\_camera\_front/depth/image\_raw/compressed  
/ob\_camera\_front/depth/image\_raw/compressed/parameter\_descriptions  
/ob\_camera\_front/depth/image\_raw/compressed/parameter\_updates  
/ob\_camera\_front/depth/image\_raw/compressedDepth  
/ob\_camera\_front/depth/image\_raw/compressedDepth/parameter\_descriptions  
/ob\_camera\_front/depth/image\_raw/compressedDepth/parameter\_updates  
/ob\_camera\_front/depth/image\_raw/theora  
/ob\_camera\_front/depth/image\_raw/theora/parameter\_descriptions  
/ob\_camera\_front/depth/image\_raw/theora/parameter\_updates  
/ob\_camera\_front/depth/metadata  
/ob\_camera\_front/depth\_registered/points  
/ob\_camera\_front/depth\_to\_accel  
/ob\_camera\_front/depth\_to\_color  
/ob\_camera\_front/depth\_to\_gyro  
/ob\_camera\_front/filter\_status  
/ob\_camera\_front/gyro/imu\_info  
/ob\_camera\_front/gyro\_accel/sample

## 2. 背部相机接口

- 话题名称:



/ob\_camera\_back/accel/imu\_info  
/ob\_camera\_back/color/camera\_info  
/ob\_camera\_back/color/image\_raw  
/ob\_camera\_back/color/image\_raw/compressed  
/ob\_camera\_back/color/image\_raw/compressed/parameter\_descriptions  
/ob\_camera\_back/color/image\_raw/compressed/parameter\_updates  
/ob\_camera\_back/color/image\_raw/compressedDepth  
/ob\_camera\_back/color/image\_raw/compressedDepth/parameter\_descriptions  
/ob\_camera\_back/color/image\_raw/compressedDepth/parameter\_updates  
/ob\_camera\_back/color/image\_raw/theora  
/ob\_camera\_back/color/image\_raw/theora/parameter\_descriptions  
/ob\_camera\_back/color/image\_raw/theora/parameter\_updates  
/ob\_camera\_back/color/metadata  
/ob\_camera\_back/depth/camera\_info  
/ob\_camera\_back/depth/image\_raw  
/ob\_camera\_back/depth/image\_raw/compressed  
/ob\_camera\_back/depth/image\_raw/compressed/parameter\_descriptions  
/ob\_camera\_back/depth/image\_raw/compressed/parameter\_updates  
/ob\_camera\_back/depth/image\_raw/compressedDepth  
/ob\_camera\_back/depth/image\_raw/compressedDepth/parameter\_descriptions  
/ob\_camera\_back/depth/image\_raw/compressedDepth/parameter\_updates  
/ob\_camera\_back/depth/image\_raw/theora  
/ob\_camera\_back/depth/image\_raw/theora/parameter\_descriptions  
/ob\_camera\_back/depth/image\_raw/theora/parameter\_updates  
/ob\_camera\_back/depth/metadata  
/ob\_camera\_back/depth\_registered/points  
/ob\_camera\_back/depth\_to\_accel

/ob\_camera\_back/depth\_to\_color  
/ob\_camera\_back/depth\_to\_gyro  
/ob\_camera\_back/filter\_status  
/ob\_camera\_back/gyro/imu\_info  
/ob\_camera\_back/gyro\_accel/sample

### 3. 腰部相机接口

- 话题名称:

/ob\_camera\_waist/accel/imu\_info  
/ob\_camera\_waist/color/camera\_info  
/ob\_camera\_waist/color/image\_raw  
/ob\_camera\_waist/color/image\_raw/compressed  
/ob\_camera\_waist/color/image\_raw/compressed/parameter\_descriptions  
/ob\_camera\_waist/color/image\_raw/compressed/parameter\_updates  
/ob\_camera\_waist/color/image\_raw/compressedDepth  
/ob\_camera\_waist/color/image\_raw/compressedDepth/parameter\_descriptions  
/ob\_camera\_waist/color/image\_raw/compressedDepth/parameter\_updates  
/ob\_camera\_waist/color/image\_raw/theora  
/ob\_camera\_waist/color/image\_raw/theora/parameter\_descriptions  
/ob\_camera\_waist/color/image\_raw/theora/parameter\_updates  
/ob\_camera\_waist/color/metadata  
/ob\_camera\_waist/depth/camera\_info  
/ob\_camera\_waist/depth/image\_raw  
/ob\_camera\_waist/depth/image\_raw/compressed  
/ob\_camera\_waist/depth/image\_raw/compressed/parameter\_descriptions  
/ob\_camera\_waist/depth/image\_raw/compressed/parameter\_updates  
/ob\_camera\_waist/depth/image\_raw/compressedDepth  
/ob\_camera\_waist/depth/image\_raw/compressedDepth/parameter\_descriptions

/ob\_camera\_waist/depth/image\_raw/compressedDepth/parameter\_updates  
/ob\_camera\_waist/depth/image\_raw/theora  
/ob\_camera\_waist/depth/image\_raw/theora/parameter\_descriptions  
/ob\_camera\_waist/depth/image\_raw/theora/parameter\_updates  
/ob\_camera\_waist/depth/metadata  
/ob\_camera\_waist/depth\_registered/points  
/ob\_camera\_waist/depth\_to\_accel  
/ob\_camera\_waist/depth\_to\_color  
/ob\_camera\_waist/depth\_to\_gyro  
/ob\_camera\_waist/filter\_status  
/ob\_camera\_waist/gyro/imu\_info  
/ob\_camera\_waist/gyro\_accel/sample

## 5.8.2. 奥比中光 335 相机接口-大模型

### 1. 头部相机接口

- 话题名称:

/camera/accel/imu\_info  
/camera/color/camera\_info  
/camera/color/image\_raw  
/camera/color/image\_raw/compressed  
/camera/color/image\_raw/compressed/parameter\_descriptions  
/camera/color/image\_raw/compressed/parameter\_updates  
/camera/color/image\_raw/compressedDepth  
/camera/color/image\_raw/compressedDepth/parameter\_descriptions  
/camera/color/image\_raw/compressedDepth/parameter\_updates  
/camera/color/image\_raw/theora  
/camera/color/image\_raw/theora/parameter\_descriptions  
/camera/color/image\_raw/theora/parameter\_updates

/camera/color/metadata  
/camera/depth/camera\_info  
/camera/depth/image\_raw  
/camera/depth/image\_raw/compressed  
/camera/depth/image\_raw/compressed/parameter\_descriptions  
/camera/depth/image\_raw/compressed/parameter\_updates  
/camera/depth/image\_raw/compressedDepth  
/camera/depth/image\_raw/compressedDepth/parameter\_descriptions  
/camera/depth/image\_raw/compressedDepth/parameter\_updates  
/camera/depth/image\_raw/theora  
/camera/depth/image\_raw/theora/parameter\_descriptions  
/camera/depth/image\_raw/theora/parameter\_updates  
/camera/depth/metadata  
/camera/depth\_registered/points  
/camera/depth\_to\_accel  
/camera/depth\_to\_color  
/camera/depth\_to\_gyro  
/camera/filter\_status  
/camera/gyro/imu\_info  
/camera/gyro\_accel/sample

## 6. 接口运行（臂、手、六维力、相机）

### 6.1. 臂部接口

#### 6.1.1. 运行节点

##### 6.1.1.1. 启动主节点

1. 如果机器人型号为 **Demo3** 或 **Max1** 或 **Pro1**:

```
cd /home/ubuntu/catkin_ws/src/aubo_dev_node/script #对应自己的目录
./usbcan.sh #设置 can 接口

roslaunch aubo_dev_plugin aubo_dev_all.launch
```

2. 其余型号则:

```
sudo su
cd /home/ubuntu/catkin_ws/
source devel/setup.bash

roslaunch aubo_dev_plugin aubo_dev_all.launch
```

### 6.1.1.2. 启动轨迹规划的节点 (测试 demo)

```
roslaunch aearm_traj_test aearm_traj_test.launch
```

### 6.1.1.3. 启动测试节点 (测试 demo)

```
roslaunch human_arm_ae_testPub human_arm_ae_testPub.launch
```

### 6.1.1.4. 启动测试节点 (测试 demo) 指令 list

主界面 (全局按键 q: 返回上一级菜单) :

1. 按键 3: CONTROL

a. 按键 1: pos: 位置模式

(1) 选择轴号:

1) 按键 1~7: 表示单轴运动 (左右臂单轴同时)

2) 按键 8: 表示 7 轴同时运动 (左右臂)

2. 按键 4: PLAN

a. 按键 5 = auto modes (控制各轴运动角度)

(1) 输入左右

1) 输入 1~7 轴的角度

3. 按键 5: Hand (手相关的指令)
- a. 按键 1 = get angle act; (获取当前角度)
    - (1) 输入左右
      - 1) 输入 1~7 轴的角度
  - b. 按键 2 = set angle ; (设置目标角度)
    - (1) 输入左右
      - 1) 输入 1~7 轴的角度
  - c. 按键 3 = get status ; (获取当前状态)
    - (1) 输入左右
  - d. 按键 4 = get error ; (获取当前 error 信息)
    - (1) 输入左右
  - e. 按键 5 = get force act ; (获取当前力矩信息)
    - (1) 输入左右
  - f. 按键 6 = set error clear; (清楚当前 error)
    - (1) 输入左右
  - g. 按键 7 = set force ; (设置力矩限制)
    - (1) 输入左右
      - 1) 1.输入 1~7 轴的力矩
  - h. 按键 8 = set speed ; (设置速度限制)
    - (1) 输入左右
      - 1) 输入 1~7 轴的速度
  - i. 按键 9 = set angle fixible; (灵活设置目标角度)
    - (1) 输入左右 (默认程序中默认值)
4. 按键 6: 6-dof-六维力
- a. 按键 1 = force & torque; (获取当前力&力矩)
    - (1) 输入左右

### 6.1.1.5. 启动测试节点 (测试 demo) 常用操作

1. 启动之后, 显示主界面;
2. 按 4+回车进入 PLAN 模式;

3. 按 **5+回车** 进行 auto modes;
4. 依次输入 **左/右臂**和 **各轴角度**，控制各轴运动到指定位置，通常先回 0 点;
5. 按 **q+回车** 返回上一级目录（主界面）；
6. 在主界面按 **3+回车** 进入 CONTROL 模式;
7. **注意**：在各轴回 0 点的前提下，再按 **8+回车**，运行展示的程序（左右各轴同时运动）。

## 6.1.2. Publish 对应的话题和服务

### 6.1.2.1. 声明 Publisher

```
ros::Publisher pub_Control_left_dev; //给左臂位置插补点
ros::Publisher pub_Control_right_dev; //给右臂位置插补点
```

### 6.1.2.2. 声明 Publisher（自己测试用接口）

```
ros::ServiceClient client_PlanJointTraj_left; //左臂轴空间运动
ros::ServiceClient client_PlanJointTraj_right; //右臂轴空间运动
```

### 6.1.2.3. Advertise 话题

```
pub_Control_left_dev =
private_nh.advertise<sensor_msgs::JointState>("/human_arm_ctrl_left", 1);
pub_Control_right_dev =
private_nh.advertise<sensor_msgs::JointState>("/human_arm_ctrl_right", 1);

client_PlanJointTraj_left =
private_nh.serviceClient<aearm_traj_test::PlanJointTraj>("/aearm_traj_test_left/pl
an_joint_traj");
client_PlanJointTraj_right =
private_nh.serviceClient<aearm_traj_test::PlanJointTraj>("/aearm_traj_test_right/
plan_joint_traj");
```

### 6.1.2.4. Publish 话题（正式接口）

#### 6.1.2.4.1. ros::Publisher pub\_Control\_left\_dev/ros::Publisher pub\_Control\_right\_dev; //左右臂的插补

```
joint_states_msg.position[i] = dPos_temp;
pub_Control_left_dev.publish(joint_states_msg);
pub_Control_right_dev.publish(joint_states_msg);
```

#### 6.1.2.4.2. client\_PlanJointTraj\_left/client\_PlanJointTraj\_right; //左右臂的轴空间控制

```
aearm_traj_test::PlanJointTraj srv_joint_left;
srv_joint_left.request.arm_name = "left_arm"; //臂的名称: 左或者右
srv_joint_left.request.joint_pos.resize(7);
srv_joint_left.request.joint_pos[0] = 0; //各轴角度, 单位弧度
srv_joint_left.request.joint_pos[1] = -0.5;
srv_joint_left.request.joint_pos[2] = 0;
srv_joint_left.request.joint_pos[3] = 1.5;
srv_joint_left.request.joint_pos[4] = 0;
srv_joint_left.request.joint_pos[5] = 1;
srv_joint_left.request.joint_pos[6] = 0;
srv_joint_left.request.vel_percent = 0.3; //速度倍率, 范围从 0-1, 0 代表速度为 0,
1 代表对应轴的最大速度。
srv_joint_left.request.acc_percent = 0.3; //加速度倍率, 范围从 0-1, 0 代表加速度为
0, 1 代表对应轴的最大加速度。
srv_joint_left.request.jerk_percent = 0.1; //加加速度倍率, 范围从 0-1, 0 代表加加速
度为 0, 1 代表对应轴的最大加加速度。
client_PlanJointTraj_left.call(srv_joint_left);

aearm_traj_test::PlanJointTraj srv_joint_right; //同上
```



```
srv_joint_right.request.arm_name = "right_arm";
srv_joint_right.request.joint_pos.resize(7);
srv_joint_right.request.joint_pos[0] = 0;
srv_joint_right.request.joint_pos[1] = -0.5;
srv_joint_right.request.joint_pos[2] = 0;
srv_joint_right.request.joint_pos[3] = 1.5;
srv_joint_right.request.joint_pos[4] = 0;
srv_joint_right.request.joint_pos[5] = 1;
srv_joint_right.request.joint_pos[6] = 0;
srv_joint_right.request.vel_percent = 0.3;
srv_joint_right.request.acc_percent = 0.3;
srv_joint_right.request.jerk_percent = 0.1;
client_PlanJointTraj_right.call(srv_joint_right);
```

## 6.2. 手部接口

### 6.2.1. 运行节点

#### 6.2.1.1. 启动主节点

```
roslaunch aubo_dev_plugin aubo_dev_all.launch
```

#### 6.2.1.2. 启动测试节点 (测试 demo)

```
roslaunch human_arm_ae_testPub human_arm_ae_testPub.launch
```

#### 6.2.1.3. 启动测试节点 (测试 demo) 指令 list

同 6.1.1.4。

#### 6.2.1.4. 启动测试节点 (测试 demo) 常用操作

1. 启动之后，显示主界面；
2. 按 **5+回车** 进入 hand 模式 (控制手爪) ；
3. 按照提示按 1~8 对手爪进行操作。

## 6.2.2. Call 对应的服务

### 6.2.2.1. 声明服务

```
aubo_dev_plugin::get_angle_act srv_get_angleact_left;
aubo_dev_plugin::set_angle srv_set_angle_left;
aubo_dev_plugin::get_status srv_get_status_left;
aubo_dev_plugin::get_error srv_get_error_left;
aubo_dev_plugin::get_force_act srv_get_forceact_left;
aubo_dev_plugin::set_clear_error srv_set_error_clear_left;
aubo_dev_plugin::set_force srv_set_force_left;
aubo_dev_plugin::set_speed srv_set_speed_left;

aubo_dev_plugin::get_angle_act srv_get_angleact_right;
aubo_dev_plugin::set_angle srv_set_angle_right;
aubo_dev_plugin::get_status srv_get_status_right;
aubo_dev_plugin::get_error srv_get_error_right;
aubo_dev_plugin::get_force_act srv_get_forceact_right;
aubo_dev_plugin::set_clear_error srv_set_error_clear_right;
aubo_dev_plugin::set_force srv_set_force_right;
aubo_dev_plugin::set_speed srv_set_speed_right;
```

### 6.2.2.2. Subscribe 话题

```
// 订阅左和右的手爪相关服务
client_inspire_get_angle_act_left =
private_nh.serviceClient<aubo_dev_plugin::get_angle_act>("/inspire_hand/get_a
ngle/left_hand");
client_inspire_set_angle_left =
private_nh.serviceClient<aubo_dev_plugin::set_angle>("/inspire_hand/set_angle
/left_hand");
```

```
client_inspire_get_status_left =  
private_nh.serviceClient<aubo_dev_plugin::get_status>("/inspire_hand/get_status/left_hand");
```

```
client_inspire_get_error_left =  
private_nh.serviceClient<aubo_dev_plugin::get_error>("/inspire_hand/get_error/left_hand");
```

```
client_inspire_get_force_act_left =  
private_nh.serviceClient<aubo_dev_plugin::get_force_act>("/inspire_hand/get_force/left_hand");
```

```
client_inspire_set_error_clear_left =  
private_nh.serviceClient<aubo_dev_plugin::set_clear_error>("/inspire_hand/set_clear_error/left_hand");
```

```
client_inspire_set_force_left =  
private_nh.serviceClient<aubo_dev_plugin::set_force>("/inspire_hand/set_force/left_hand");
```

```
client_inspire_set_speed_left =  
private_nh.serviceClient<aubo_dev_plugin::set_speed>("/inspire_hand/set_speed/left_hand");
```

```
client_inspire_get_angle_act_right =  
private_nh.serviceClient<aubo_dev_plugin::get_angle_act>("/inspire_hand/get_angle/right_hand");
```

```
client_inspire_set_angle_right =  
private_nh.serviceClient<aubo_dev_plugin::set_angle>("/inspire_hand/set_angle/right_hand");
```

```
client_inspire_get_status_right =  
private_nh.serviceClient<aubo_dev_plugin::get_status>("/inspire_hand/get_status/right_hand");
```

```
client_inspire_get_error_right =  
private_nh.serviceClient<aubo_dev_plugin::get_error>("/inspire_hand/get_error/right_hand");
```

```

right_hand");
client_inspire_get_force_act_right =
private_nh.serviceClient<aubo_dev_plugin::get_force_act>("/inspire_hand/get_f
orce/right_hand");
client_inspire_set_error_clear_right =
private_nh.serviceClient<aubo_dev_plugin::set_clear_error>("/inspire_hand/set_
clear_error/right_hand");
client_inspire_set_force_right =
private_nh.serviceClient<aubo_dev_plugin::set_force>("/inspire_hand/set_force/
right_hand");
client_inspire_set_speed_right =
private_nh.serviceClient<aubo_dev_plugin::set_speed>("/inspire_hand/set_spee
d/right_hand");

```

### 6.2.2.3. Call 服务

#### 6.2.2.3.1. 获取角度

```

ROS_INFO("try to get angle act left arm-----");
if (client_inspire_get_angle_act_left.call(srv_get_angleact_left))
{
    for (int i = 0; i < 6; i++)
    {
        srv_get_angleact_left.response.curangleRatio[i];
        // ROS_INFO("curangleRatio %d = %f", i,
srv_get_angleact_left.response.curangleRatio[i]);
        ROS_INFO("curangleRatio %d = %f", i,
srv_get_angleact_left.response.curangleRatio[i]);
    }
}
else

```

```
{  
    ROS_INFO("get angle act left arm failed!!");  
}
```

#### 6.2.2.3.2. 设定角度

```
ROS_INFO("try to get angle act left arm-----");  
  
srv_set_angle_left.request.angle0Ratio = dPos1;  
srv_set_angle_left.request.angle1Ratio = dPos2;  
srv_set_angle_left.request.angle2Ratio = dPos3;  
srv_set_angle_left.request.angle3Ratio = dPos4;  
srv_set_angle_left.request.angle4Ratio = dPos5;  
srv_set_angle_left.request.angle5Ratio = dPos6;  
  
if (client_inspire_set_angle_left.call(srv_set_angle_left))  
{  
    if (srv_set_angle_left.response.angle_accepted)  
    {  
        ROS_INFO("angle_accepted");  
    }  
    else  
    {  
        ROS_INFO("angle_not_accepted");  
    }  
}
```

#### 6.2.2.3.3. 获取状态

```
ROS_INFO("try to get status left arm-----");
```

```

if (client_inspire_get_status_left.call(srv_get_status_left))
{
    for (int i = 0; i < 6; i++)
    {
        srv_get_status_left.response.statusvalue[i];
        // ROS_INFO("curangleRatio %d = %f", i,
srv_get_angleact_left.response.curangleRatio[i]);
        ROS_INFO("status %d = %u", i,
srv_get_status_left.response.statusvalue[i]);
    }
}

```

#### 6.2.2.3.4. 获取错误

```

ROS_INFO("try to get error left arm-----");
if (client_inspire_get_error_left.call(srv_get_error_left))
{
    for (int i = 0; i < 6; i++)
    {
        srv_get_error_left.response.errorvalue[i];
        // ROS_INFO("curangleRatio %d = %f", i,
srv_get_angleact_left.response.curangleRatio[i]);
        ROS_INFO("error %d = %u", i, srv_get_error_left.response.errorvalue[i]);
    }
}

```

#### 6.2.2.3.5. 获取力矩

```

ROS_INFO("try to get force act left arm-----");
if (client_inspire_get_force_act_left.call(srv_get_forceact_left))

```

```

{
    for (int i = 0; i < 6; i++)
    {
        srv_get_forceact_left.response.curforceRatio[i];
        // ROS_INFO("curangleRatio %d = %f", i,
srv_get_angleact_left.response.curangleRatio[i]);
        ROS_INFO("curforceRatio %d = %f", i,
srv_get_forceact_left.response.curforceRatio[i]);
    }
}

```

#### 6.2.2.3.6. 清除错误

```

ROS_INFO("try to set error clear left arm-----");
if (client_inspire_set_error_clear_left.call(srv_set_error_clear_left))
{
    if (srv_set_error_clear_left.response.setclear_error_accepted)
    {
        ROS_INFO("set error clear_accepted");
    }
    else
    {
        ROS_INFO("set error clear_not_accepted");
    }
}
}

```

#### 6.2.2.3.7. 设置力矩

```

ROS_INFO("try to set force left arm-----");

```

```

srv_set_force_left.request.force0Ratio = dPos1;
srv_set_force_left.request.force1Ratio = dPos2;
srv_set_force_left.request.force2Ratio = dPos3;
srv_set_force_left.request.force3Ratio = dPos4;
srv_set_force_left.request.force4Ratio = dPos5;
srv_set_force_left.request.force5Ratio = dPos6;

if (client_inspire_set_force_left.call(srv_set_force_left))
{
    if (srv_set_force_left.response.force_accepted)
    {
        ROS_INFO("force_accepted");
    }
    else
    {
        ROS_INFO("force_not_accepted");
    }
}

```

#### 6.2.2.3.8. 设置速度

```

ROS_INFO("try to set speed left arm-----");

srv_set_speed_left.request.speed0Ratio = dPos1;
srv_set_speed_left.request.speed1Ratio = dPos2;
srv_set_speed_left.request.speed2Ratio = dPos3;
srv_set_speed_left.request.speed3Ratio = dPos4;
srv_set_speed_left.request.speed4Ratio = dPos5;
srv_set_speed_left.request.speed5Ratio = dPos6;

```



```

if (client_inspire_set_speed_left.call(srv_set_speed_left))
{
    if (srv_set_speed_left.response.speed_accepted)
    {
        ROS_INFO("speed_accepted");
    }
    else
    {
        ROS_INFO("speed_not_accepted");
    }
}
}

```

#### 6.2.2.3.9. 灵活设置角度

```

ROS_INFO("try to get angle act left arm-----");
srv_set_angle_flexible_left.request.name.resize(3);
srv_set_angle_flexible_left.request.name[0] = "1";
srv_set_angle_flexible_left.request.name[1] = "4";
srv_set_angle_flexible_left.request.name[2] = "6";
srv_set_angle_flexible_left.request.angleRatio.resize(3);
srv_set_angle_flexible_left.request.angleRatio[0] = 0.5;
srv_set_angle_flexible_left.request.angleRatio[1] = 0.5;
srv_set_angle_flexible_left.request.angleRatio[2] = 0.5;

if (client_inspire_set_angle_flexible_left.call(srv_set_angle_flexible_left))
{
    if (srv_set_angle_flexible_left.response.angle_accepted)
    {

```

```
        ROS_INFO("angle_accepted");
    }
    else
    {
        ROS_INFO("angle_not_accepted");
    }
}
```

## 6.3. 六维力接口

### 6.3.1. 运行节点

#### 6.3.1.1. 启动主节点

```
roslaunch aubo_dev_plugin aubo_dev_all.launch
```

#### 6.3.1.2. 启动测试节点 (测试 demo)

```
roslaunch human_arm_ae_testPub human_arm_ae_testPub.launch
```

#### 6.3.1.3. 启动测试节点 (测试 demo) 指令 list

同 6.1.1.4。

#### 6.3.1.4. 启动测试节点 (测试 demo) 常用操作

1. 启动之后, 显示主界面;
2. 按 **6+回车** 进入 dof 模式;
3. 按 **1+回车** 进行 get force & torque;
4. 依次输入 **左/右臂**, 获取一次 6dof 读数。

### 6.3.2. Subscribe 对应的话题

#### 6.3.2.1. 声明 Subscriber

```
ros::Subscriber sub_6dof;
```

#### 6.3.2.2. Subscribe 话题

```
if (iL_R == 1)
{
    sub_6dof = private_nh.subscribe("/human_arm_6dof_left", 10,
    &TestPubNodelet::callback_6dof, this); // 订阅 CMD 的简化指令
}
else
{
    sub_6dof = private_nh.subscribe("/human_arm_6dof_right", 10,
    &TestPubNodelet::callback_6dof, this); // 订阅 CMD 的简化指令
}
```

## 6.4. 相机接口

### 6.4.1. 启动奥比中光 335L 相机\*3-导航

```
source orbbec_ros/devel/setup.bash
roslaunch orbbec_camera list_devices_node
roslaunch orbbec_camera slam_330.launch
```

### 6.4.2. 启动奥比中光 335 相机\*1-大模型

```
cd ~/orbbec_ros
source devel/setup.bash
roslaunch orbbec_camera gemini_330_series.launch
```