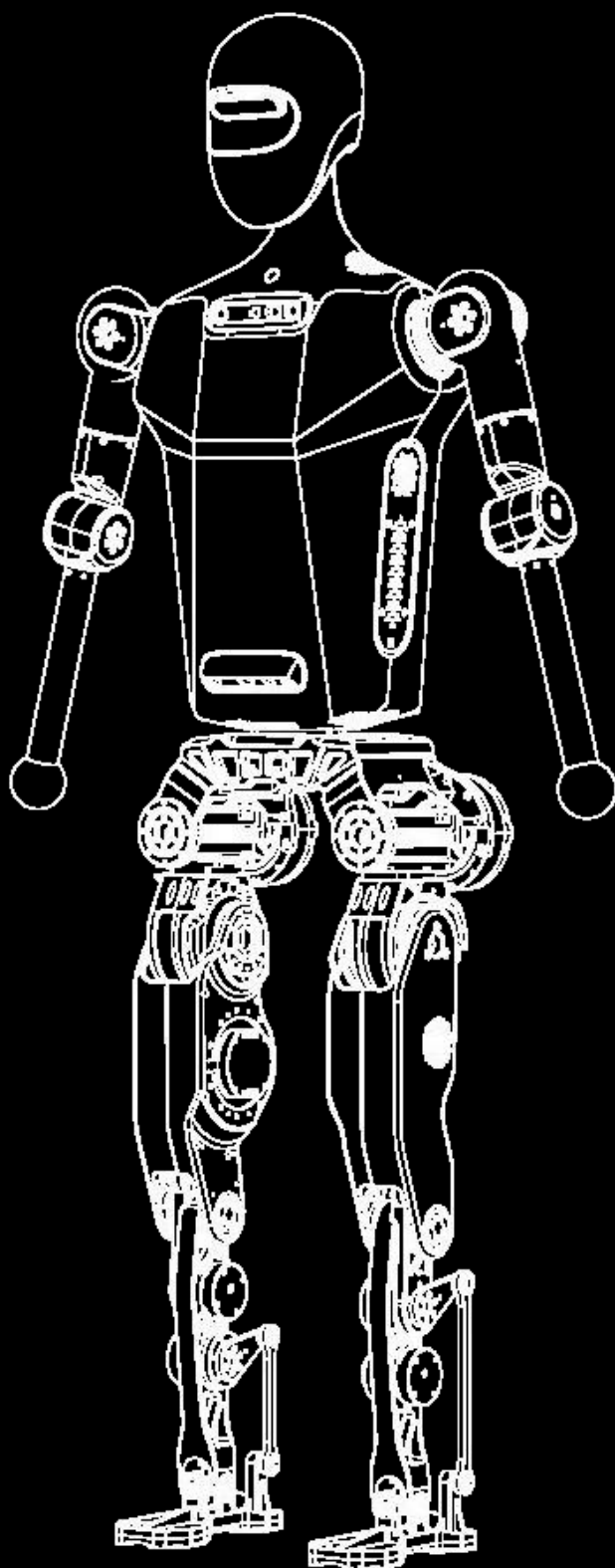




HUMANOID

# 天工通用人形机器人(Lite)

## 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/V7Zww7fUJiEINTkiRRLcsTFLneg> 获取最新版本。

# 1. 关于天工 c

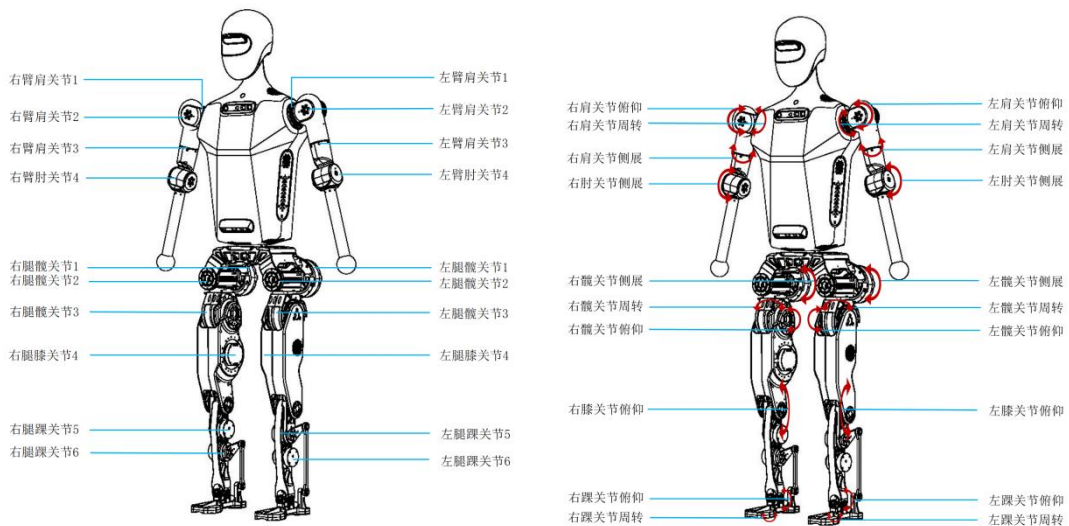
简易版天工整机分为上半身和下半身，具备多个自由度。单手臂拥有 4 个自由度，包括肩关节和肘关节。单腿拥有 6 个自由度，包括髋关节、膝关节和踝关节。整机共有 20 个自由度，由 20 个关节电机组成，使得机器人能够实现精确的运动和姿态控制。

## 1.1. 总体规格参数

项目	规格参数
产品高度	1630mm
产品净重	43kg
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
续航时间	纯行走 2 小时，纯站立 4 小时
摄像头	RGBD 摄像头*4 (头部，胸腔，腰部，背部)
传感器	高精度 IMU
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； 算力：275 TOPS

自由度	20 个一体化关节 (臂 4 个*2, 腿 6 个*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: 1kHz; IMU: $\geq 400\text{Hz}$ ; 相机 (2 个 RGBD) : 30Hz

## 1.2. 关节参数

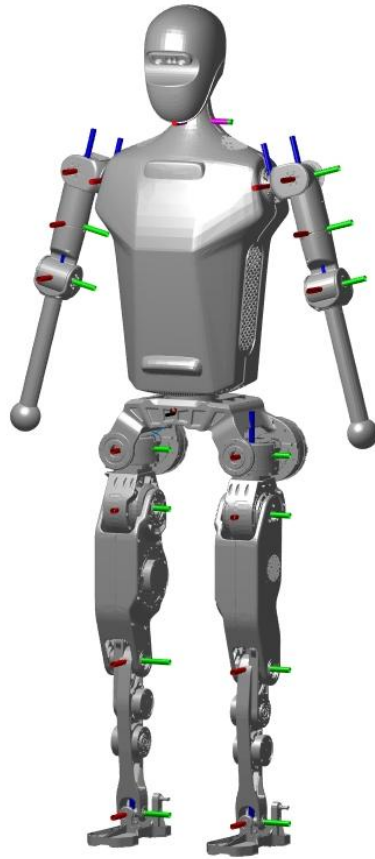


所属部位	名称	运动范围 (°)	额定转速 (rpm)	最大转速 (rpm)	额定力矩 (Nm)	峰值力矩 (Nm)
臂部	肩关节周转 Shoulder Yaw	-20°~+190°	78	93	12	36
	肩关节俯仰 Shoulder Pitch	-90°~+180°	78	93	12	36
	肩关节侧展 Shoulder Roll	-120°~+120°	78	93	12	36
	肘关节侧展	-30°~+120°	78	93	12	36

	Elbow Roll					
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

### 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) 用以太网线连接机器人背后的调试用以太网接口和用户电脑端；
    - (2) 用户电脑配置以太网端口地址为 192.168.41.xx/255.255.255.0；

(3) 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）；

(4) 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
```

(5) 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_simple.launch #启动本体驱动程序
```

(2) 其次，打开第二个终端，依次输入如下命令启动运控驱动程序：

```
cd ros_ws #切换到 ros_ws 目录
source install_isolated/setup.bash #执行环境变量脚本
roslaunch rl_control_new rl.launch #启动运控驱动程序
(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和本体驱动程序按照该顺序依次按 control+C 停止；
- (5) 待上述两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

简易版天工 SDK 由北京人形机器人创新中心开发，提供了丰富的接口，涵盖臂部电机控制、腿部电机控制以及 IMU（惯性测量单元）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. 电机控制模式说明

简易版天工共提供三种的电机控制模式，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**：结合力和位置传感器数据，实现精确的力和位置控制，适用于需要精确控制接触力的任务。
2. **速度控制**：通过设定电机的转速，实现对机器人的平滑运动控制，适用于连续运动任务。
3. **位置控制**：通过设定目标位置，使电机移动到指定位置，适用于需要精确定位的任务。

## 5. 全局值说明

- 消息名称：MotorName
- 数据定义位置：~/rosws/src/bodyctrl\_msgs/msg/MotorName.msg
- 说明：简易版天工所有电机的命名 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_ARM_LEFT_1 = 13
uint16 MOTOR_ARM_LEFT_2 = 14
uint16 MOTOR_ARM_LEFT_3 = 15
uint16 MOTOR_ARM_RIGHT_1 = 16
uint16 MOTOR_ARM_RIGHT_2 = 17
uint16 MOTOR_ARM_RIGHT_3 = 18
```



```
uint16 MOTOR_ARM_LEFT_4 = 23
uint16 MOTOR_ARM_RIGHT_4 = 24
```

注：关节电机 ID 对应的具体关节请参考 [1.2 关节参数](#)。

## 6. 接口说明

### 6.1. IMU

#### 6.1.1. 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/lmu.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
```

### 6.2. 臂/腿部电机

#### 6.2.1. 状态获取接口

##### 1. 获取头部电机信息

- 说明：获取臂/腿部电机的状态信息，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/motor\_state
- 数据定义位置：~/rosws/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
```

## 6.2.2. 控制接口

### 1. 力位混合控制臂/腿部电机位置

- 说明：臂/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：~/rosws/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. 控制臂/腿部电机速度

- 说明：臂/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. 控制臂/腿部电机位置

- 说明：臂/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
```

## 6.3. 运控命令接口

- 参数说明：

控制	命令参数	范围	备注
前进后退	线速度 x 向	-0.5~1.0m/s	前向为正
旋转	角速度 z 向	-0.2~0.2rad/s	左向为正
左右移动	线速度 y 向	-0.1~0.1m/s	左向为正

- 数据格式:

```
#速度矢量控制, twist 格式
```

```
/cmd-vel
```

- 特别说明:

请注意, 只有当航模遥控器按键 F 向上拨杆时, 机器人将使用上面列出的 `cmd_vel` 进行控制; 若按键 F 不处于向上拨杆的状态, 则机器人将处于遥控器控制模式。