Z1 机械臂

开发文档

Unitree

www.unitree.com

目录

产品概述

本章节主要介绍 Z1 机械臂安装方法

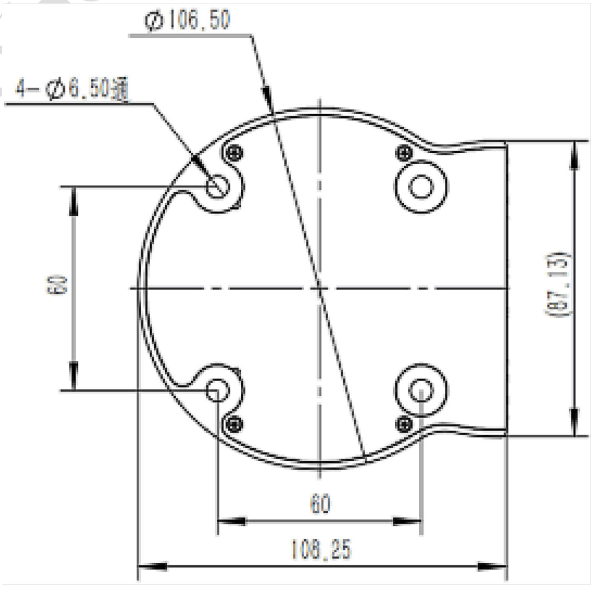
机械臂安装

物品清单

类型	数量	备注
机械臂	1	/
电源适配器	1	DC24V
机械臂固定板	1	安装固定机械臂
G 型夹	2	用于将机械臂固定板与桌面固定
网线	1	2m
M6 内六角螺丝	4	M6X16，固定机械臂
M2.5 内六角螺丝	2	M2.5X8，用于固定电源线缆
2mm 内六角扳手	1	用于拧紧 M2.5 螺丝
5mm 内六角扳手	1	用于拧紧 M6 螺丝

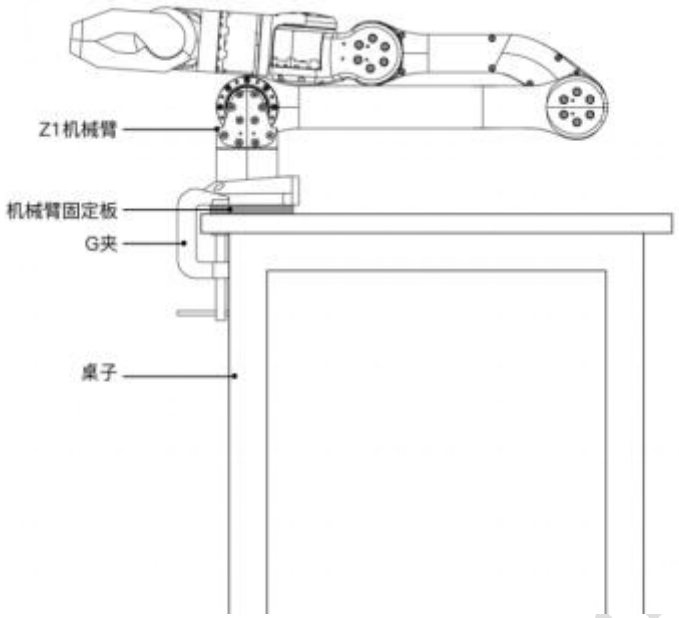
固定机械臂

用户在固定机械臂时可根据机械臂底座孔位尺寸及真实环境自行设计安装台架，机械臂的固定台架不仅需要承受机械臂自身的重量，还要承受最大加速度运动时的瞬时动态作用力。机械臂使用 4 颗 M6 螺栓，用内六角扳手安装机械臂。底座安装图如下



机械臂底座螺丝安装图

当然，我们提供了固定板及 G 夹，方便直接固定到桌面上。

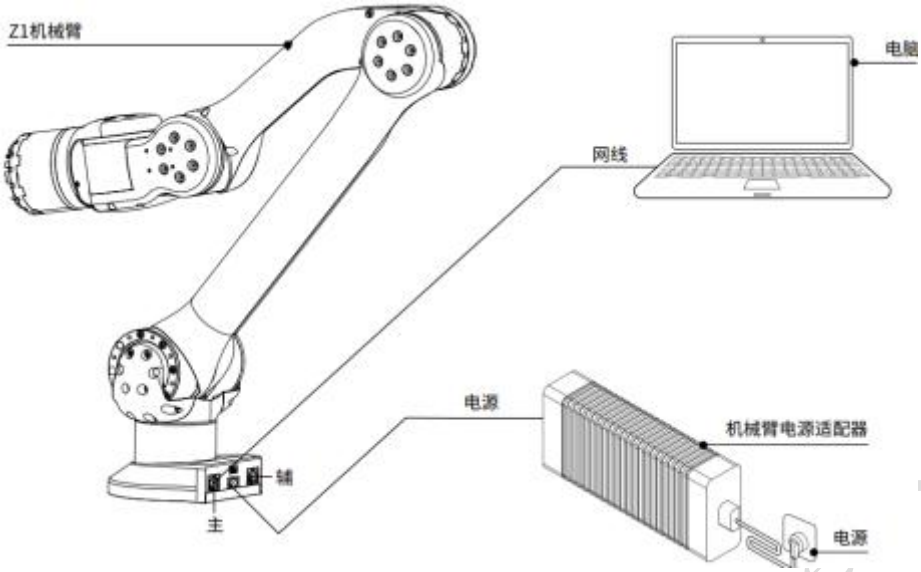


机械臂安装图

线缆连接

机械臂线缆主要有两种：供电线和通信线。机械臂供电线的接头具备防呆功能，供电线插入下图所示的机械臂供电接口。同时将通信线(即网线)的一端插入下图所示的机械臂网口并锁紧即可，通信线的另一端连接电脑。 需要注意将网线插入到机械臂的**主网口**。

- 机械臂的主网口用于控制机械臂，副网口用于更改默认 IP，使用时不可插反。

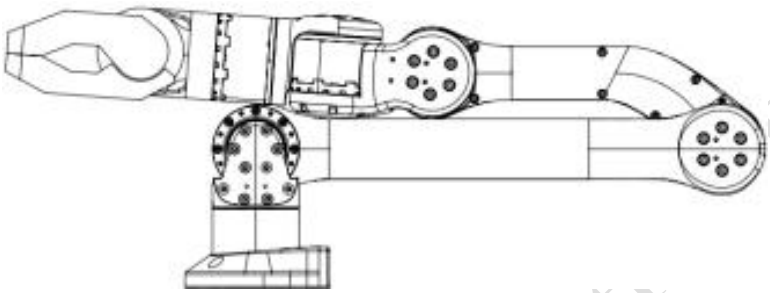


机械臂线缆连接图

开机前准备

零位摆放

机械臂重新上电前一定要确保运动程序关闭，否则会有危险发生的可能。并让机械臂处于零位， 机械臂零位姿态如下图，J1、J6 关节缝隙两侧的线完全对应，其他关节摆放到位即可。



机械臂上电零位连接图

- 需先将机械臂端电源线插入机械臂 XT60 口并固定后再将适配器端插入电源。严禁将适配器插入电源后再将电源线插入机械臂。
- 需要特别注意的是，机械臂电机编码器为电机端单圈绝对值编码器，输出端没有编码器，每次使用前要将机械臂摆到指定位置（零位）。当然，每次使用完成通过程序控制机械臂回到零位，下一次使用前并不需要手动摆放。

成功上电时，自检通过机械臂绿灯会常亮，蓝灯闪烁。

需注意，每次使用前将机械臂各关节转到零位，这是为了让控制算法中的理论零位与实际机械零位重合。另外，机械臂末端电机的黑白电源线是有 DC24V 电的，如果不需要，请务必把黑白电源线用绝缘胶带缠绕固定，防止短路等发生危险。

网络配置

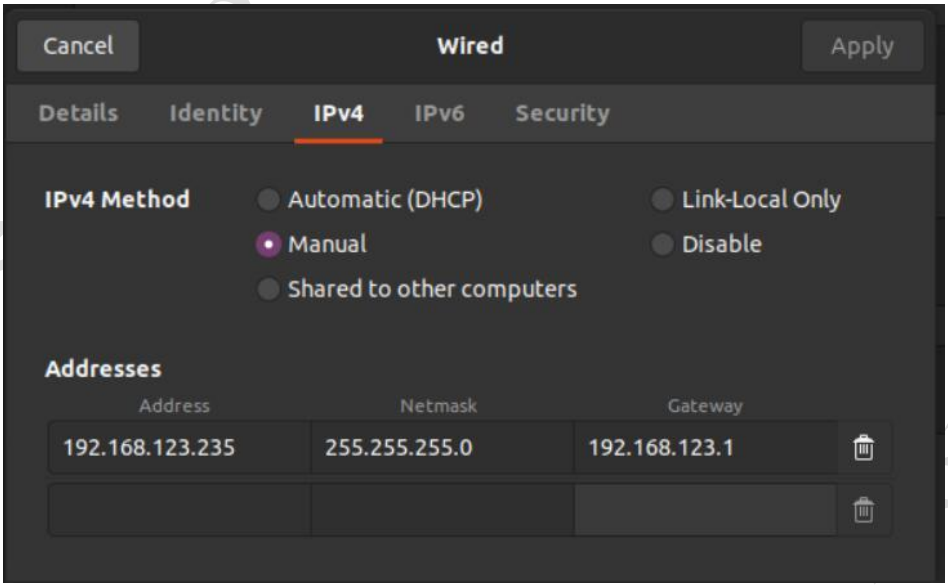
机械臂的默认 IP 为 **192.168.123.110**，用户需要在同一网段才可与机械臂正常通信。

1. 更改 PC 网段

完成以下设置后通过 ping 192.168.123.110 指令检测与机械臂连接是否正常。

方法一：图形界面操作

直接在系统网络设置的如下界面更改： Settings -> Network -> PCI/USB Ethernet (点击需要使用的网口后面的齿轮图标)



固定本机 IP 地址

方法二：命令行操作

以更改为 192.168.123.162 示例，在终端中运行 ifconfig，您将找到您的端口名称（可通过插拔网线，查看增加/减少的端口，来查找对应的端口名称）。例如，enp000ec626053d.

以下命令中的 **enpxxx 不可直接输入**，均需要替换成自己电脑中通过 ifconfig 所显示出来的具体名称。

```shell
sudo ifconfig enpxxx down           #enpxxx is your PC port
sudo ifconfig enpxxx 192.168.123.162/24
sudo ifconfig enpxxx up
```

以上方式为临时更改 IP 使用，您也可以将永久更改 PC 的 IP，操作如下

```shell
sudo vim /etc/network/interfaces
```

添加一下文本至上述打开的 interfaces 文件中

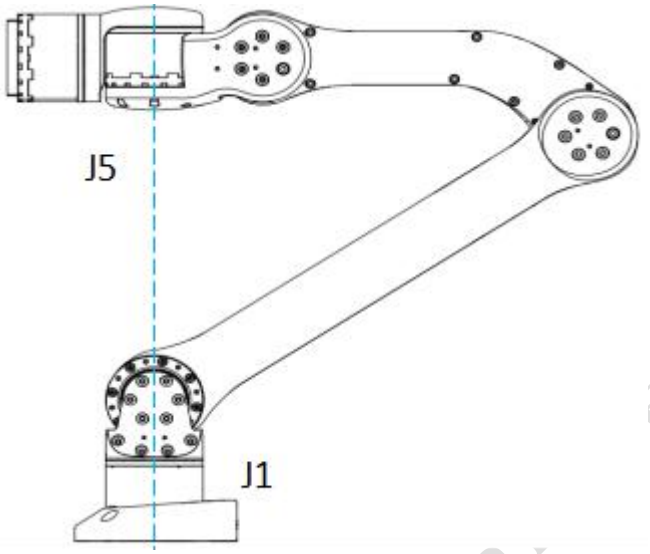
```shell
auto enpxxx
iface enpxxx inet static
address 192.168.123.162
netmask 255.255.255.0
```

2. 默认 IP 更改

如需更改机械臂默认 IP，可以通过网线连接机械臂**辅助网口**，在终端运行 z1_controller/unitreeArmTools.py 程序，根据提示输入信息即可。

奇异

机械臂处于奇异位置，自由度将发生退化，会造成某些关节的角速度无限大，机械臂有失控风险。z1 机械臂具有肩关节奇异：第五个关节轴线与第一个关节轴线同轴，如下图。



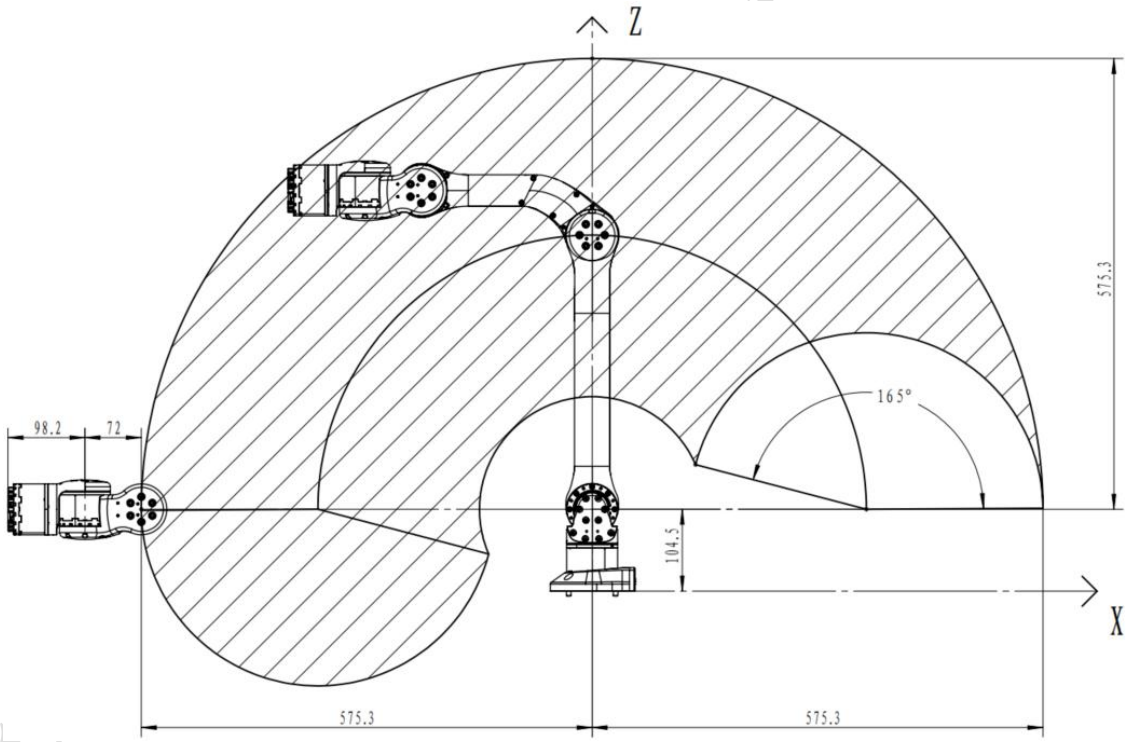
机械臂奇异位置

对于逆运动学问题，在奇异区域外，采取解析解方式；在奇异区域内，采用 QP 求近似解。建议机械臂上电后运行至 forward 点位，避开该区域。

规格参数

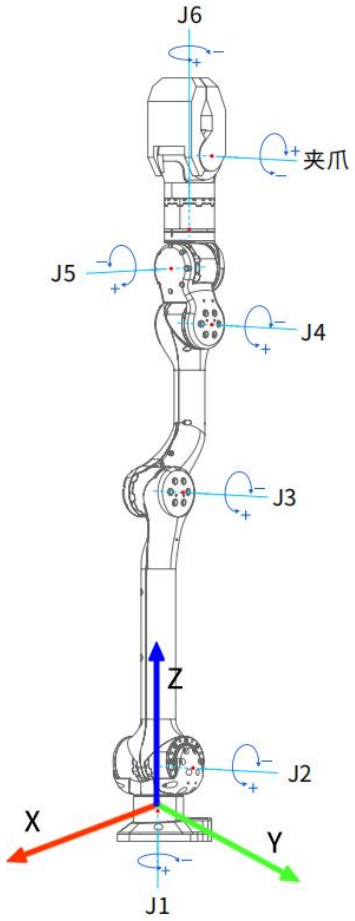
型号	Z1	电源需求	电压 24V 电流>20A
自由度	6 轴	接口	Ethernet
自重	4.1kg	用户控制系统	Ubuntu
负载	3-5kg	功率	峰值 500W
最大臂展	740mm	力反馈和碰撞检测	有
重复定位精度	~0.1mm	接口控制	位置+力控

机械臂运动范围



旋转方向及坐标系

关节坐标系



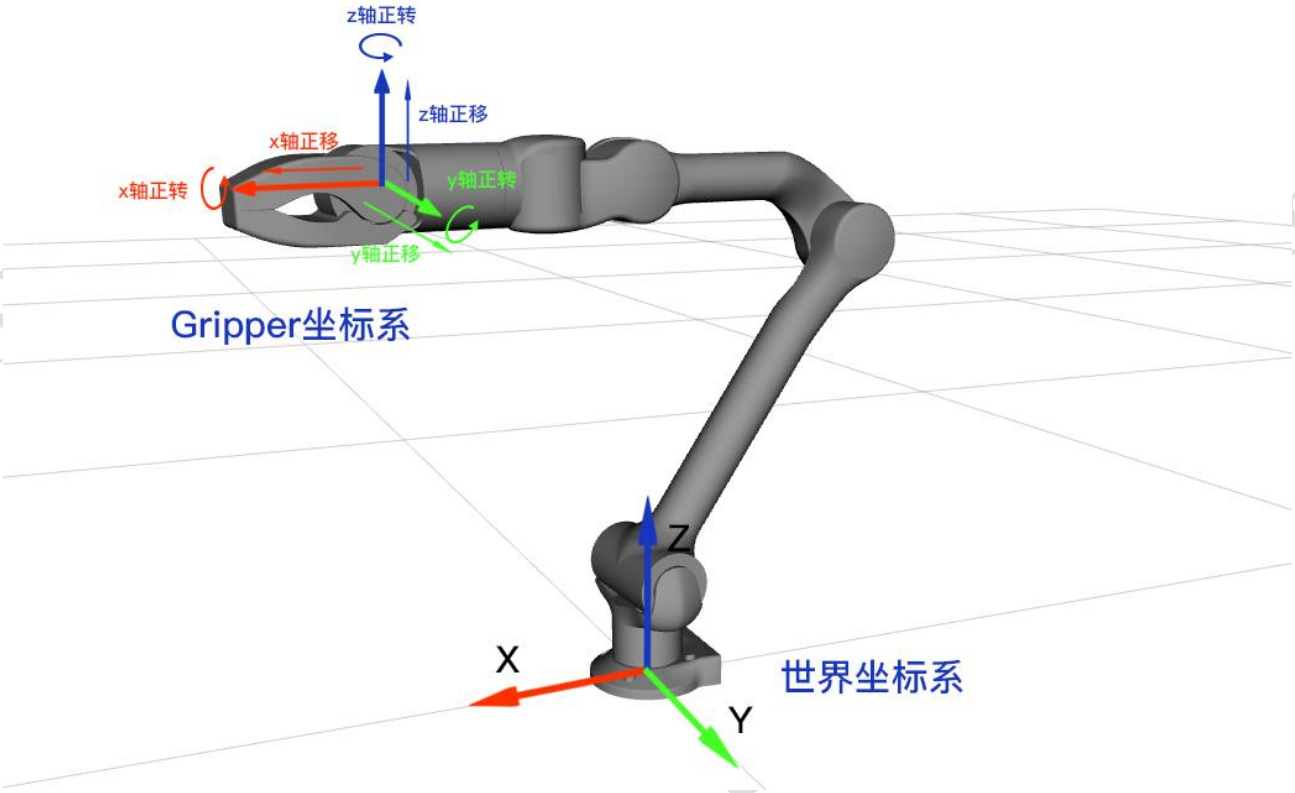
z1 机械臂关节序号及关节转动正方向定义

各关节序号从 J1 开始，逐个递增至 J6。在上图中 + 键表示关节转动的正方向，- 键表示关节转动的负方向。

关节	各关节旋转法向	关节坐标
J1	[0，0，1]	[0，0，0.065]
J2	[0，1，0]	[0，0，0.1115]
J3	[0，1，0]	[-0.35，0. 0.1115]
J4	[0，1，0]	[-0.132，0，0.1685]
J5	[0，0，1]	[-0.06，0，0.1685]
J6	[1，0，0]	[-0.0128，0，0.1685]

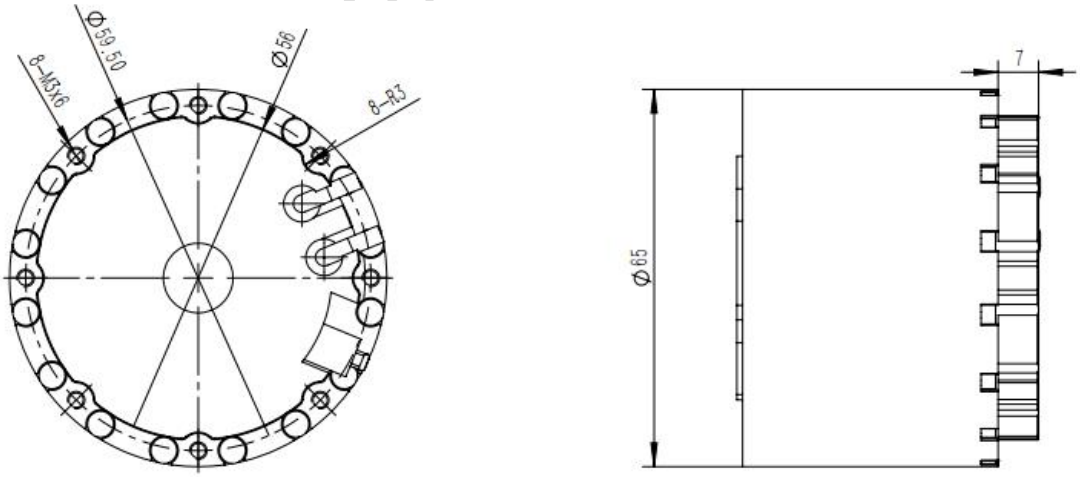
实际计算采用弧度制，还有其他如坐标轴、惯性矩阵等信息都在 unitree_ros/robots/z1_description/xacro 的文件里有描述。

笛卡尔坐标系



笛卡尔空间控制

末端关节法兰



z1 机械臂末端法兰示意图

其中手爪装载平面（即 J6 末端平面）的绝对初始位置为[0,049，0，0.1605]，即无手爪时的笛卡尔系末端计算位姿。

Unitree_gripper 电机中心（并非手爪位型中心）相对于装载平面的位置为[0.0382，0，0]，有手爪时最终的末端计算位姿为[0.0872，0，0.1605]。

机械臂使用

本章节介绍 Z1 机械臂使用

SDK介绍

z1 机械臂共提供 4 个文件夹供用户使用，分别是 z1_controller，z1_sdk 以及 unitree_ros，z1_ros。

1. z1_controller 存储着直接控制机械臂的源码。
2. z1_sdk 包含了用于控制机械臂的一些接口，用户在创建自己的程序使用 z1 机械臂时需要包含该文件夹。
3. unitree_ros 用于机械臂仿真，其中包含了宇树四足产品 Go1，A1，Aliengo，Laikago 和机械臂产品 Z1 的仿真文件。
4. z1_ros（包含 unitree_ros），提供 noetic moveit1 支持。

> 注意：z1_ros 中也包含 z1_controller 和 z1_sdk，提供的接口可以直接发送 UDP 结构体进行通信，与上述 z1_sdk 不兼容。

z1_controller

除以下提到的文件外，用户无需查看该文件夹下的其他文件。

1. build/z1_ctrl

用户创建 build 文件后编译程序，最终可执行程序名为 z1_ctrl，每次用户使用机械臂时都需要执行该程序。

- 调用./z1_ctrl -v 可以查看当前机械臂版本信息，如果此时可以正常连接机械臂，也会显示下位机版本信息。
- 调用./z1_ctrl k 可以使用键盘控制机械臂运动
- 直接调用./z1_ctrl 需使用 SDK 控制机械臂

2. CMakeLists.txt

用户在使用时需要根据自身使用方式自行选择采用实际控制或仿真，只需注释掉不需要的一行即可。更改完毕后需重新编译一下程序。

```cmake
set (COMMUNICATION UDP)
# set(COMMUNICATION ROS)
```

3. unitreeArmTools

该文件用于更改机械臂 IP（默认 192.168.123.110） 使用方法：使用网线连接好机械臂和 PC（更改 IP 需连接右端备用网口），终端执行 python3 unitreeArmTools.py，按提示输入信息即可。

4. config/config.xml

该文件只在 z1_ctrl 启动时读取一次：

(1) IP & Port

IP：此为通过 unitreeArmTools 更改的机械臂 IP，当机械臂 IP 更改后，需要更改此地址以便 z1_ctrl 和机械臂正常通信

Port：机械臂的端口为 8880，此项更改的是 PC 执行 z1_ctrl 时绑定的本机端口，默认 8881，用于同一台 PC 控制多个机械臂

(2) collision

包含碰撞检测相关的设置

open: 设置为 Y 或 N 选择是否打开碰撞检测

limitT: 为进行检测时计算扭矩与反馈扭矩的差值检测阈值，单位为 NM

load: 挂载在机械臂末端的负载重量，单位 kg，该值与 open 设置无关，会一直参与末端负载的动力学计算。

5. config/saveArmStates.csv

该文件存储 labelRun 可到达的点位，记录的数据为该位姿下关节坐标。点位可通过 labelSave 创建，亦可手动修改文件输入。

z1_sdk

1. include

该文件夹存放着 unitre_arm_sdk 的头文件，用户可以查看其中注释了解函数作用

- **unitree_arm_sdk/control**

该文件夹下共两个文件，ctrlComponents.h 和 unitreeArm.h，其中 ctrlComponents.h 主要是将所有的控制参数放在了一个类中便于调用，unitreeArm 封装了用于控制机械臂的所有接口，用户使用时只要查看该类中信息。

- **unitree_arm_sdk/model**

该文件夹下包含 armModel 类，内部包含 z1 机械臂的正逆运动学、逆动力学以及空间雅克比计算等函数。通过在 unitreeArm 类下调用_ctrlComp->armModel 使用各种方法。

2. examples

为方便用户使用，该文件夹下内含机械臂 sdk 的使用示例文件。

- **highcmd_basic**

如果用户希望简单了解如何使用机械臂，可以查看该示例。该示例下共含三种使用机械臂的方式:

① armCtrlByFSM()

该方法直接调用 unitreeArm 中提供的函数运行机械臂，如 MoveJ、MoveL、MoveC、backToStart 等。 所有函数与通过./z1_ctrl k 命令直接使用键盘控制时的效果一致。

② armCtrlInJointCtrl()

该方法相当于在 highcmd_development 的基础上再进一步封装，原本用户需要输入关节命令 \(q \And \dot{q}\)，现在只需输入希望电机运行的方向即可。函数内直接会进行如下命令计算：

\(\dot{q} = directions*\omega\)
\(q_{k} = q_{k-1} + \dot{q}*\delta t\)

采用该方法时相当于在键盘控制时按下 2 键后通过键盘直接控制机械臂。

③ armCtrlInCartsian()

与上例类似，在笛卡尔空间下，原本需要用户控制空间速度旋量 \(V =[\omega \quad v]'\)，即 unitreeArm.twist 从而控制机械臂运转，该方法在此基础上进行了封装，用户直接控制希望机械臂末端运行的方向即可。 函数内直接会进行如下命令计算： 其中 T 是由 R，p 组成的齐次变换矩阵，[ω]为ω的反对称矩阵

\(posture_k = posture_{k-1}+posture_{\Delta}\)
\([\omega] = \log{(R_{k-1}^T R_k)}\)
\(v=p_\Delta\)

采用该方法时相当于在键盘控制时按下 3 键后通过键盘直接控制机械臂。

- **highcmd_development**

如果用户希望自行开发，通过**规划轨迹**运行机械臂，可以查看该示例。

当 sendRecvThread->start()执行后，该线程会以 500HZ 的频率执行 arm.sendRecv()函数。

关节下控制时，该函数会发送 unitreeArm 下的 q, qd, gripperQ, gripperW 给 z1_controller，笛卡尔下控制时，该函数会发送 unitreeArm 下的 twist 给 z1_controller，用户只需不断更改这些参数即可控制机械臂。也可自己编写线程进行调用 unitreeArm.sendRecv()进行通信。

- **lowcmd_development**

如果用户希望直接从**底层控制**电机的 \(q, \dot{q}, \tau_f, k_p, k_d\) 参数，可以查看该示例。

该文件展示了如何对电机进行直接发送 p, d 参数。用户可以通过编写自己的机械臂程序对电机直接进行控制，实现自主开发。电机最终输出的力矩如下

\[\tau = k_p * 25.6 * (q_d - q) + k_d * 0.0128 * (\dot{q_d} - \dot{q}) + \tau_f\]

25.6 与 0.0128 为与电机通信协议中的缩放倍数。

CtrlComponents 下的 sendRecvThread 是调用 unitreeArm 的函数进行指令操作，如运行至 forward 视为一条指令，而运行 lowcmd 时建议采用自己定义的线程，执行 run 函数，run 函数开始通过计算确定当前需要发给电机的命令，最后调用 sendRecv 发送 udp 报文。

- **lowcmd_multirobots**

该程序提供了控制多个机械臂的示例。详见 SDK 运行

3. examples_py

这里面包含机械臂 sdk 的 python 版本接口。

接口设计在 arm_python_interface.cpp 文件中，其中已经包含的函数及变量可直接在 python 中使用，用户也可自行更改，详情查看 pybind11 documentation。

调用方式:

编译完成后 z1_sdk/lib 中会生成一个名为 unitree_arm_interface 的库。

在第一个终端运行./z1_ctrl。

在 z1_sdk/examples_py 目录下打开第二个终端执行 python3 example_highcmd.py。

环境安装

运动控制程序可以在 Ubuntu 18.04 以上的 X86 架构下的 linux 平台运行，仿真可在 ROS melodic 或 neotic 版本上运行。

建议使用 Ubuntu20 neotic

依赖安装

- libboost-dev
- libeigen3-dev

```bash
sudo apt install -y libboost-dev libeigen3-dev
sudo ln -s /usr/include/eigen3/Eigen /usr/local/include/Eigen
sudo ln -s /usr/include/eigen3/unsupported /usr/local/include/unsupported
```

- **pybind11**

如果需要使用 python 接口，需安装 pybind11

```bash
# Install pybind11
git clone https://github.com/pybind/pybind11.git
cd pybind11
mkdir build && cd build
cmake .. -DPYBIND11_TEST=OFF
make -j
sudo make install
```

ROS 安装

中国用户可能会因为无法连接外网而发生错误，可以尝试通过国内鱼香 ROS 开发的一键安装命令执行。

```Shell
wget http://fishros.com/install -O fishros && . fishros
```

可以通过小乌龟测试 ROS 是否正常

```Shell
roscore #第一个终端
rosrun turtlesim turtlesim_node # 第二个终端
rosrun turtlesim turtle_teleop_key # 第三个终端
```

SDK运行

实机控制

1. 首先对机械臂进行通电（**摆放至零位**），并通过 ping 192.168.123.110 指令查看是否通信正常（强调：网线需要插在左边**主网口**）

2. 编译 z1_controller，在该文件夹下创建 build 文件夹

```Shell
mkdir build & cd build
cmake ..
make
```

3. 执行./z1_ctrl

当执行该条命令后，终端会不断地打印[WARNING] UDPPort::recv, unblock version, connect wit z1_sdk wait time out 语句，这是正常的，因为我们还没有启动机械臂 SDK 与机械臂控制器通信。

4. 打开第二个终端，打开 z1_sdk 文件夹，创建 build 文件并编译

5. 在 Z1_SDK 中执行./highcmd_basic

ROS 仿真

1. 打开仿真

如果用户对 ros 文件不太熟悉，请在 home 下创建 unitree_ws/src 文件并将 unitree_ros 文件移动到 /home/username/unitree_ws/src/unitree_ros 同时下载 unitree_legged_msgs 放入 unitree_ws/src/目录下

```Shell
cd ~/unitree_ws                                  #打开该文件夹
catkin_make                                      #初始化 ROS 工作空间
echo "source ~/unitree_ws/devel/setup.bash">>~/.bashrc  #将 ros 路径添加到环境变量，可由 pwd 命令获取当前路径替换该路径
source ~/.bashrc                                 #更新环境变量
```

在终端执行 roslaunch unitree_gazebo z1.launch，如果成功配置此时可以显示出 gazebo 的仿真界面。

可以通过选择 UnitreeGripperYN 配置仿真是否有无手爪（默认带手爪）。如

```Shell
roslaunch unitree_gazebo z1.launch UnitreeGripperYN:=false
```

2. 编译 z1_controller，在该文件夹下创建 build 文件夹（打开第 2 个终端）

```Shell
mkdir build & cd build
cmake ..
make
```

3. 执行 ./sim_ctrl

当执行该条命令后，终端会不断地打印[WARNING] UDPPort::recv, unblock version, connect wit z1_sdk wait time out 语句，这是正常的，因为我们还没有启动机械臂 SDK 与机械臂控制器通信。

各种信息都会在这个窗口打印，用户使用使请多观察此窗口内容。

4. 执行./highcmd_basic，会执行一个示例动作

```Shell
./highcmd_basic
```

此时我们已经完成仿真操作，整个流程为：**运行 ROS–>运行 z1_ctrl–>运行 SDK 实例。**

多机控制

对于需要控制多个 Z1 机械臂的用户，SDK 提供了 lowcmd_multirobots 示例代码。

1. 打开 z1_controller，设置编译条件为 UDP， 复制 z1_controller 文件夹，如命名为 z1_controller_111

2. 打开 z1_controller_111

① main.cpp

将 L51 的与 SDK 通信的端口设置为如下示例

```cpp
ctrlComp->cmdPanel = new ARMSDK(events, emptyAction, "127.0.0.1", 8074, 8073, 0.002);
```

② config.xml

更改 IP 为 192.168.123.111，更改端口为 8882.

③ unitreeArmTools.py

更改第二台机械臂的 IP 为 192.168.123.111

如果使用两个 USB 转网口与机械臂通信，而非使用交换机组网，那么两个机械臂必须设置在不同网段。

如第一个在 192.168.123.110，第二个在 192.168.120.110

3. 根据之前所述，进行如下操作：

① 连接实体机械臂 110 并在第一个终端执行 z1_controller 下的 z1_ctrl。

② 连接实体机械臂 111 并在第二个终端执行在 z1_controller_111 的 z1_ctrl。

③ 在第三个终端运行 z1_sdk 中的 lowcmd_multirobots。

此时可以看到两个机械臂的第一个关节都转动了一定角度。

键盘控制

z1_controller 使用有限状态机进行代码架构，每一个状态机代表具体的工作。

State	KeySwitch	Switchable
BACKTOSTART	~	1 2
PASSIVE	1	~ 2 3 =
JOINTCTRL	2	~ 1 3 4 5 6 7 8 9 0 -
CARTESIAN	3	~ 1 2 4 5 6 9
MoveJ	4	~ 1 2 3 5 6 9
MoveL	5	~ 1 2 3 4 6 9
MoveC	6	~ 1 2 3 4 5 9
TEACH	7	~ 1 2
TEACHREPEAT	8	automatically switched to 2
SAVESTATE	9	automatically switched to 2
TOSTATE	0	automatically switched to 2
CALIBRATION	=	automatically switched to 1

在键盘控制模式下，按下某个键位就可以进入该状态机，在 Switchable 列表下为从某一状态机下可以进入的其他状态机。

为了避免奇异问题，我们建议用户每次使用时先将机械臂通过 TOSTAET 状态机运行至 forward 位姿下。

关闭原来的 z1_ctrl 程序，重新执行 z1_ctrl k。

BACKTOSTART

所有电机返回初始位置。

PASSIVE

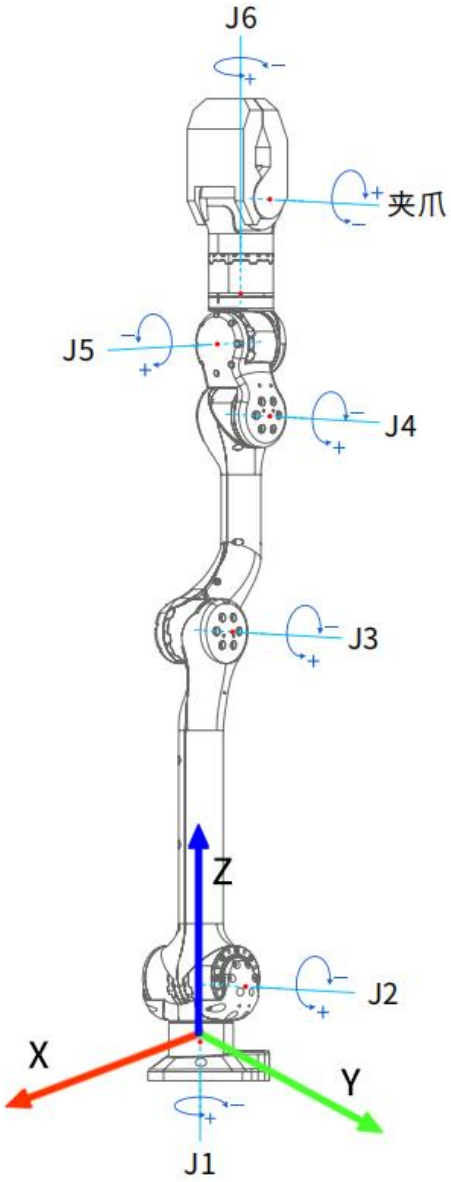
所有电机进入阻尼状态（上电后的默认状态）。

JOINTCTRL

在关节空间速度控制中，可以通过**长按**键盘直接地给定机械臂 6 个关节运动的速度，从而控制机械臂的运动。

需要再次说明的是，所有关节坐标系均是右手系，在使用前需注意个关节的正反转运动趋势，以确保安全操作。

Joint ID	0	1	2	3	4	5	Gripper
Keyboard	Q/A	W/S	D/E	R/F	T/G	Y/H	up/down
Joint ActionJoint Action (right hand)	positive/ negative	positive/ negative	positive/ negative	positive/ negative	positive/ negative	positive/ negative	positive/ negative

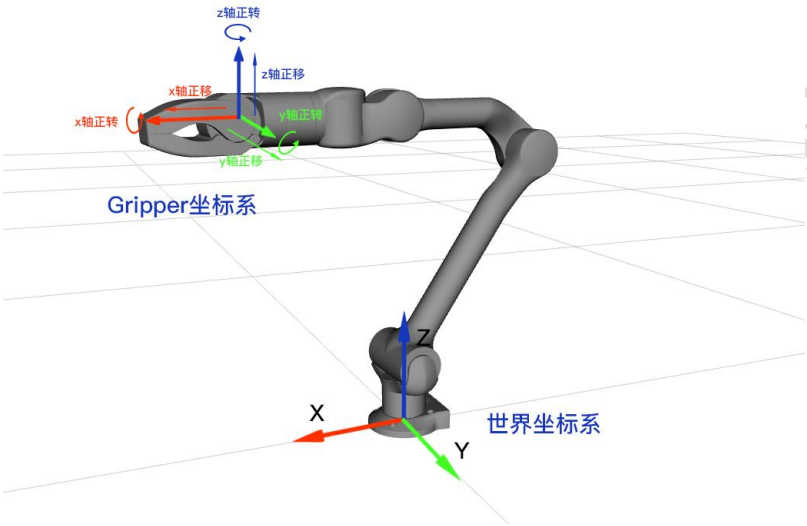


z1 机械臂关节序号及关节转动正方向定义

CARTESIAN

在笛卡尔空间控制中，可以直接地通过键盘或手柄给定机械臂末端的期望位置与姿态的运动速度，进而控制机械臂的运动。

Keyboard	Q/A	W/S	E/D	R/F	T/G	Y/H	Gripper
Key Function	forward/ backward	right/ left	up/ down	roll	pitch	yaw	up/down



笛卡尔空间控制

MOVEJ、MOVEL、MOVEC

根据提示输入末端位姿(roll pitch yaw x y z)

点的坐标可以先通过关节控制移动到目标位置，然后通过 savestate 状态机保存至 csv 文件或调用 getJointState 可执行文件获取。

示例：(sdk 中的 highcmd_basic)

1. 先将机械臂运行至 forward 状态

   按 0，输入 forward 后 enter

2. MOVEJ

   按 4，输入 0.5 0.1 0.1 0.5 -0.2 0.5

请确保输入正确，之后按两次 enter，机械臂开始运动

> ● 注：两次 enter 的目的是：在键盘控制时，我们允许用户在按一次 enter 后继续输入下一次位姿，完成输入后按两次 enter 顺序执行。

3. MOVEL

按 5， 输入 0 0 0 0.45 -0.2 0.2 后按两次 enter

4. MOVEC

按 6，此时需要输入两次位姿，分别是中间位姿和最终位姿。

输入 0 0 0 0.45 0 0.4 后 enter，

再输入 0 0 0 0.45 0.2 0.2 后按两次 enter

TEACH & TEACHREPEAT

首先按~使机械臂返回原点，或控制机械臂运动至某一初始点位后按 2 进入关节控制状态机，再按 7 进入示教状态机后，输入标签名后 enter，此时可以拖拽机械臂运动一定的轨迹，机械臂将持续记录轨迹直至用户按 2 进入关节空间控制。

该轨迹将保存在 z1_controller/config 目录下一个新的.csv 文件中。

对于已有的 csv 文件，可以按 8 进入示教重复功能，输入已保存的轨迹标签名，机械臂将重复该示教轨迹运动。

SAVESTATE & TOSTATE

键盘按 9，允许将机械臂某一时刻的各关节角度记录为一个标签，我们称此功能为 标签记录。

机械臂会自动保存当前的位姿状态，根据提示输入自定的标签名。记录的标签保存于 savedArmStates.csv 文件中。 完成后机械臂及将自动转至关节空间控制状态。

对于 savedArmStates.csv 中已有的标签，可以在键盘按 0 后输入已存储的标签名后 enter，机械臂将以 MoveJ 的方式运行至相应位姿。

当运动完成后，机械臂将自动转至关节空间控制状态。

CALIBRATION

键盘按=，设置当前位置为初始位置，完成设置后机械臂将自动转至阻尼状态。

新的机械臂需在运行程序的第一步执行该操作。此时机械臂需要在零位并处于阻尼状态，即刚执行./z1_ctrl k.

手柄使用

控制模式

手柄控制固定在机器狗上的机械臂，有三种控制模式。

1.关节空间控制。当处于此模式时，手柄可控制机械臂单个关节转动。

2.笛卡尔空间控制。当处于此模式时，机械臂作为一个整体，手柄控制机械臂在工具坐标系下沿 XYZ 轴方向移动以及绕 XYZ 轴转动。

3.固定轨迹运动。机械臂内置演示用的固定动作。

手柄键位介绍

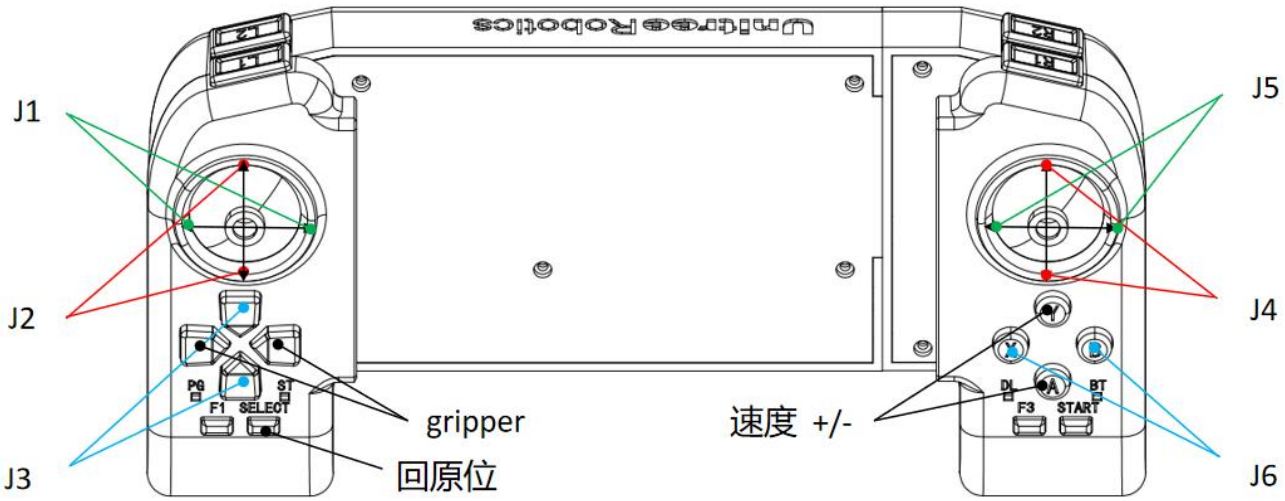
关节空间控制

- B1

当机器狗停止时，同时按住 **L1+L2** 键，切换到手柄控制机械臂，机器狗不再运动。此时按下 **R2** 键则进入关节空间控制模式，用手柄可控制单个关节转动，对应键位如下图所示。运动结束后，如果想让机械臂保持当前姿态不动，可同时按下 **L2+R2**，机械臂不再接收手柄的控制信息，此时按下 **L1+L2**，可使机械臂保持当前姿态控制机器狗运动。机器狗停止后，同时按住 **L1+L2** 键，切换到手柄控制机械臂，按下 **R2** 可继续控制机械臂动作，运动结束后，按下 **SELECT** 键可使机械臂回到原位。

- Aliengo

运动模式启动后，**L2+A** 锁定机器狗，此时按下 **R2** 键则进入关节空间控制模式，用手柄可控制单个关节转动，对应键位如下图所示。运动结束后，按下 **SELECT** 键可使机械臂回到原位。手柄控制可参考 B1。



关节空间手柄控制

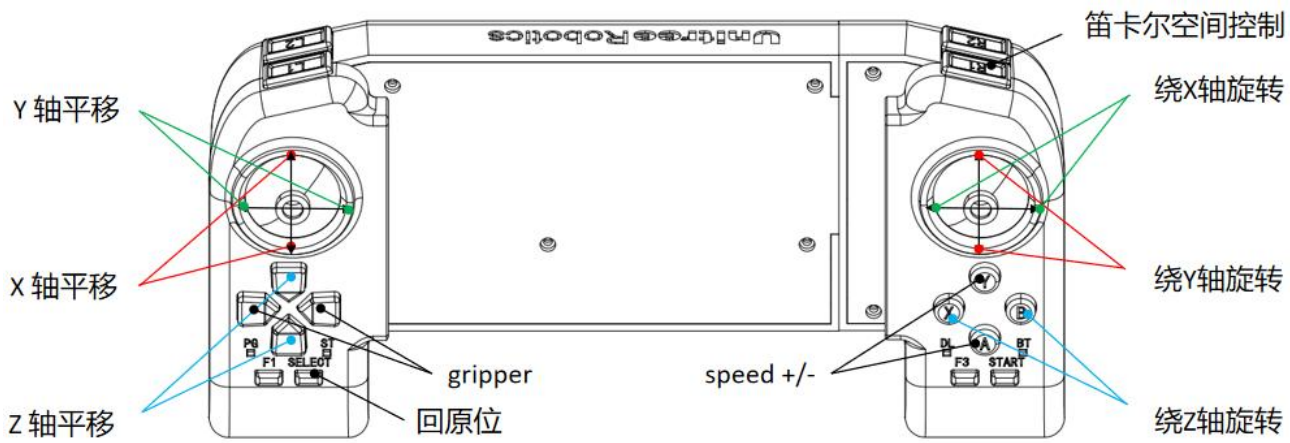
笛卡尔空间控制

- B1

当机器狗停止时，同时按住 **L1+L2** 键，切换到要控制控制机械臂，机器狗不再运动。此时按下 **R1** 键则进入笛卡尔空间控制模式，或者当机械臂处于关节空间控制模式下，按下 **R1** 键位也可进入笛卡尔空间控制模式，对应键位如下图所示。运动结束后，如果想让机械臂保持当前姿态不动，可同时按下 **L2+R2**，机械臂不再接收手柄的控制信息，此时按下 **L1+L2**，可使机械臂保持当前姿态控制机器狗运动。机器狗停止后，同时按住 **L1+L2** 键，切换到手柄控制机械臂，按下 **R2** 或 **R1** 可继续控制机械臂动作，运动结束后，按下 **SELECT** 键可使机械臂回到原位。

- Aliengo

运动模式启动后，**L2+A** 锁定机器狗，此时按下 **R1** 键则进入笛卡尔空间控制模式，或者当机械臂处于关节空间控制模式下，按下 **R1** 键位也可进入笛卡尔空间控制模式，对应键位如下图所示。运动结束后，按下 **SELECT** 键可使机械臂回到原位。手柄控制可参考 B1。



笛卡尔空间手柄控制

固定轨迹运动

- B1

当机器狗停止时，同时按住 **L1+L2** 键，切换到要控制控制机械臂，机器狗不再运动。按下 **R2** 键后进入关节空间控制，此时同时按下 **R2+A** 键机械臂会进入预设固定轨迹运动。若让机械臂回到原位，则按下 **SELECT** 键即可。

- Aliengo

运动模式启动后，**L2+A** 锁定机器狗，按下 **R2** 键后进入关节空间控制，此时同时按下 **R2+X** 键机械臂会进入预设固定轨迹运动。若让机械臂回到原位，则按下 **SELECT** 键即可。

注意事项

- B1

由于机器狗和机械臂用同一个手柄进行控制，存在部分键位共用，因此，控制机器狗移动时，要同时按下 **R1+R2** 将机械臂切入阻尼状态或者同时按下 **L2+R2** 机械臂姿态固定，手柄可控制机器狗运动。判断机械臂是否在阻尼状态，可用手拖动机械臂，若能拖动则表明已切入阻尼。控制机械臂运动时，则需要机器狗处于停止状态。

- Aliengo

由于机器狗和机械臂用同一个手柄进行控制，存在部分键位共用，因此，控制机器狗移动时，要同时按下 **R1+R2** 将机械臂切入阻尼状态或者同时按下 **L2+R2** 机械臂姿态固定。判断机械臂是否在阻尼状态，可用手拖动机械臂，若能拖动则表明已切入阻尼。控制机械臂运动时，则需要机器狗处于停止状态。

常见异常

机械臂抖动

为了保证整个机械臂的力控性能，每个关节采用的谐波减速器减速比相对较低，使得关节刚度相对较低；并且本机械臂是轻量级机械臂，各连杆也比较轻量，机械臂整机机械刚度也相对较低。

因此，如果采用的控制方式不够优化，机械臂会有抖动的情况。

机械臂零位摆放

由于目前 Z1 机械臂电机没有输出端编码器，所以不能在任意位置开机使用，上电使用前需要将机械臂摆到指定位置（零位）。

电机过温保护

如果机械臂末端负载较大、持续运行时间太长，在机械臂不同的位姿下，负载比较大的关节，会导致关节电机过热而进入保护状态，电机不再输出扭矩，需要将机械臂断电后重新上电。

上电时关节未复位

机械臂开机使用时如果发生如下报错，是由于机械臂在上电时 joint1 没有在零位导致。 因此，上电前务必将机械臂各个关节摆放到零位。

```text
[ERROR] The No.1 term of joint angle: -0.024 does not between [0.000 3.142]
```

不断电，直接更换电脑不能控制机械臂

机械臂上电与上位机通讯后会自动绑定端口号，所以要更换电脑控制机械臂需重新对机械臂进行断电上电。

上电后机械臂指示灯不亮

上电后机械臂指示灯不亮，主控板保险丝烧坏，需联系售后返厂维修。

机械臂指示灯红灯闪烁

机械臂关节通讯出现异常。终端会提示出现异常的关节序号。

附录

售后服务及政策

保修期限的说明

1. 您购买 Z1 整机及其他相关产品后，整机保修一年，保修期限自您收到货物当日算起。
2. 如果您购买的宇树 Unitree 产品已经超过保修期，您也可以通过购买服务的方式来延长原有维保时长。
3. 维保期 1 年内客户有权向宇树 Unitree 获取相应服务。
4. 在质保期内，未经宇树 Unitree 允许，发生私自改造、拆装、开壳等行为，质保期将直接失效。
5. 收到客户坏件后 20 天内将修复或更换件送达。

服务范畴

根据具体的情况，我们会为您购买的宇树 Unitree 产品进行相应的维修或者部件的更换。

下列情况可能导致宇树科技服务不能按要求提供：

1. 不可抗力（如：火灾、水灾、地震、雷击等）引起的意外情况。
2. 社会性问题（如：动乱、战争、罢工、政府管制等）引起的服务条件恶化。
3. 能量供应中断（如：电力、供水、油料等）引起的服务无法实施。
4. 由以下原因对宇树科技生产设备造成的损坏，不属于服务承诺范畴。
5. 由于不可抗力事件（自然灾害、火灾、战争等）对宇树科技生产的设备造成损坏。
6. 自然损耗或磨损造成设备损坏。
7. 因现场设备运行环境（比如潮湿）或外部因素（比如外部电磁干扰、内部互联设备的故障等）不能满足设备已书面提示的正常运行的环境要求，所造成的直接损坏。
8. 由于故意或疏忽、使用不当或蓄意破坏行为对宇树科技生产设备造成大规模的硬件或数据损坏。
9. 没有根据设备的操作手册运行宇树科技公司生产设备，所造成的损坏。
10. 因客户或第三方所造成的系统损坏，包括未按照宇树科技的要求擅自对系统重新搬迁、安装；未按照宇树 Unitree 要求擅自对识别标志进行调整、修改或删除所造成的损坏。
11. 未按宇树科技要求擅自对产品设备本身进行涂改、标记。
12. 由于客户自身基础设施的原因造成的系统损坏。
13. 未经宇树科技授权，硬件或软件已被修改的设备。

免责声明

1. 宇树 Unitree 不提供本文不涉及的任何明确或隐含的商业和技术保证。

2. 宇树 Unitree 不保证所提供的产品/服务是完全无缺陷的，完全达到客户要求的，使用该产品/服务不会遇到任何问题和中断的，也不保证宇树 Unitree 能完全修复这些缺陷。

3. 任何情形下，宇树 Unitree 都不因本服务说明书对客户的直接或间接经济损失承担法律责任，宇树 Unitree 对由于其自身产品责任所导致的客户损失的最大赔偿额不高于客户购买该产品/服务所支付的金额。

4. 外购件不在本服务说明书所含的服务范围内。

5. 终端产品及附件类不提供现场服务。

6. 宇树 Unitree 提供的超过 1 年的维保服务是一项可以选择的服务，客户可以选择是否购买相关的服务并选择何时终止。如果客户选择购买相关的服务，则表示客户允许宇 Unitree 在提供服务时访问、采集和处理故障、检测、定位、调试相关的信息。宇树 Unitree 将在客户同意的前提下，遵从客户的要求访问和处理相关信息，该信息将仅用于提供维保服务。由于客户是这些信息的控制者，宇树 Unitree 无法确认此类信息是否包含客户机密信息或个人数据，客户应当保证其将根据所适用的法律要求，获得或保留所有必要的同意、许可、授权（"同意"）用于让宇树 Unitree 提供此服务，使宇树 Unitree 在提供相关服务时不会违反适用的法律要求、客户的隐私政策、或者客户与用户的协议。宇树 Unitree 将采取合理的措施保障此类客户信息的安全，但宇树 Unitree 不对提供服务过程中获取和处理此类信息的行为导致的直接或间接责任负责。

其他细则

1. 您在将产品寄往宇树 Unitree 时需先行承担邮寄费用。

2. 宇树 Unitree 收到您需要保修的问题产品后，将对产品进行检测以确定问题和责任。若属于产品本身质量缺陷，宇树 Unitree 将负责承担检测费、材料费、人工费及寄回的快递费。

3. 如果检测产品不符合免费维修条件，您可选择付费维修或原机寄回，原机寄回的快递费将由您承担。

4. 如果产品的问题超过保修范围，我们会根据具体问题收取相应的检测费、更换零件费、测试费、人工费及运输费用。