

使用手册 – G602 机械臂

Revision 3.2.2

2021-02-19

适用于 G602 机器人

（硬件版本号 V3.0.0）

安诺机器人（深圳）有限公司

www.robotanno.com

敬告

感谢您购买和使用 PROBOT G602 产品。为了您的安全和利益，在使用产品前请您仔细阅读本产品用户使用手册及随机附带的全部资料。如果您未按照用户使用手册操作和使用产品，而导致任何的人身伤害、财产或其他损失，安诺机器人（深圳）有限公司将不承担责任。

版权说明

对于您将阅览的以下信息（包括但不限于文字表述及其组合、图标、图片及图表、版面设计、编排方式、数据及软件介绍、色彩搭配等），安诺机器人（深圳）有限公司特发表以下声明：

本文档系安诺机器人（深圳）有限公司（以下简称：本公司）创作并对其享有完全的、完整的版权，未经本公司书面同意，任何单位或个人均不得以任何形式进行转载、复制、编辑、修改，或以其他方式违法使用。

本文档中可能产生的著作权、硬件、软件及专有技术的所有权、或某项技术的专利申请权、专利权等全部权利皆为本公司所有。

未经本公司书面同意，其他单位或个人使用该信息资料而影响自身或第三方权益的，或第三方未同本公司联系、核实而与其他单位或个人进行交易并造成损失的，本公司不承担任何赔偿或补偿责任。

安诺机器人（深圳）有限公司

地址：广东省深圳市宝安区西乡街道固戍南昌第一工业区 1 栋 705

销售邮箱：robotanno@robotanno.com

售后邮箱：support@robotanno.com

官方网站：www.robotanno.com

强制

- 本说明书对 G602 机械臂的使用进行了全面说明，请务必在认真阅读并充分理解的基础上操作机械臂。
- 机械臂上电工作前，请将机械臂调整到靠近**初始状态**，并且操作者处于机械臂工作范围之外的安全位置。
- 紧急情况下，马上按下急停键，若不能及时制动机械臂，则可能引发人身伤害或设备损坏事故。

注意

- 说明书中的图及照片，为代表性示例，可能与所购买产品不同。
- 说明书有时由于产品改进、规格变更及自身更便于使用等原因而进行适当的修改，修改后的说明书将更新封面下边版本号，并以修订版发行。
- 由于破损、丢失等原因需订购说明书时，请与本公司销售部联系，按封面的版本号订购。
- 客户擅自进行产品改造，不在本公司保修范围之内，本公司概不负责。

目录

目录.....	4
文档版本.....	6
1 安全注意事项.....	7
1.1 符号及其含义.....	7
1.2 危险事项.....	7
1.3 注意事项.....	8
1.4 使用环境.....	9
2 系统说明.....	11
2.1 产品概述.....	11
2.2 控制箱说明.....	15
2.3 IO 接口说明.....	16
2.4 硬件参数.....	18
2.5 结构尺寸.....	18
3 快速上手指南.....	21
3.1 硬件接口与连线.....	21
3.2 启动 ROS2GO.....	22
3.3 系统上电启动.....	23
3.4 设置网络连接.....	24
3.5 启动 ROS 上位机.....	25
3.6 仿真/在线切换.....	26
3.7 机械臂使能.....	26
3.8 调整机械臂初始位姿.....	27
3.9 机械臂点动控制.....	27
4 机械臂的使用.....	28
4.1 基本操作.....	28
4.2 上位机 ROS 环境搭建.....	29
4.3 上位机功能介绍.....	34
4.4 单机仿真.....	44
4.5 真机运行.....	45
4.6 使用夹具.....	45
4.7 例程演示.....	47

5 MoveIt!编程接口.....	48
5.1 设置运动状态.....	48
5.2 设置目标位姿.....	52
5.3 笛卡尔空间运动.....	57
6 一键固件升级.....	61
6.1 使用须知.....	61
6.2 升级前准备.....	61
6.3 一键升级.....	62
7 Blockly 图形化编程.....	65
7.1 启用图形化编程.....	65
7.2 编程界面简介.....	66
7.3 Blockly 代码块.....	68
7.4 Blockly 编程示例.....	71
7.5 常见问题.....	73
8 常见故障诊断.....	75
8.1 无法开机.....	75
8.2 机械臂无法回到初始位姿.....	75
8.3 机械臂抓不紧物体.....	75
9 附录.....	76
9.1 参考资料.....	76
9.2 联系方式.....	76

文档版本

日期	版本	作者	概要
2019-4-13	1.0	安诺机器人	初始版本
2019-10-8	2.0	安诺机器人	添加施教点保存与运行功能
2019-12-15	2.1	安诺机器人	添加零位标定功能
2020-2-23	2.2.0	安诺机器人	新增一键固件更新功能；更新零位标定功能
2020-3-11	2.3.0	安诺机器人	内容整理；新增第 7 章 Blockly 图形化编程
2020-05-05	2.3.1	安诺机器人	零位标定功能升级，更新相关参数说明
2020-7-16	3.0.0	安诺机器人	添加 V3.0.0 更新
2020-12-24	3.2.1	安诺机器人	对应软硬件系统版本 V3.2.1
2021-1-16	3.2.2	安诺机器人	对应软硬件系统版本 v3.2.2

Table 1: 版本历史

1 安全注意事项

感谢您购买 PROBOT G602 机械臂。为了您的安全和防止损坏机械臂，请在使用机械臂前熟读并掌握本说明书和其他附属资料，在熟知全部设备知识、安全知识及注意事项后再开始使用，并特别注意以下安全标识。

1.1 符号及其含义



危险

误操作时有危险，可能发生死亡或重伤事故



注意

误操作时有危险，可能发生中等程度伤害、轻伤事故或物件损坏



强制

手册和文档中必须遵守的事项



禁止

手册和文档中明确禁止的事项

即使是属于“注意”类的事项，也会因情况不同而产生严重后果，故任何一条“注意”事项都极为重要，请务必严格遵守。

1.2 危险事项



危险

(1) 紧急情况下，马上按下急停键，若不能及时制动机械臂，则可能引发人身伤害或设备损坏事故。



急停键

(2) 解除急停后再接通伺服电源时，要解除造成急停的事故后再启动急停键，由于误操作造成的机械臂动作，可能引发人身伤害事故。



急停状态解除

(3) 在机械臂动作范围内运动时，请遵守以下事项：

- 1) 考虑机械臂突然向自己所处方位运动时的应变方案。
- 2) 确保设置躲避场所，以防万一。

注意

由于误操作造成的机械臂动作，可能引发人身伤害事故。

(4) 进行以下作业时，请确认机械臂的动作范围内没人，并且操作者处于安全位置操作：

- 1) G602 机械臂接通电源时；
- 2) 试运行；
- 3) 示教再现时。

(5) 请不要在机械臂工作状态下移动和维修机械臂，如要移动和维修请先关闭机械臂电源，断电之后再执行此项操作。

注意

不慎进入机械臂动作范围内或与机械臂发生接触，都有可能引发人身伤害事故。如发现异常时，请立即按下急停键。

急停键位于 **PROBOT G602** 机械臂控制箱正面右侧。

1.3 注意事项

(1) PROBOT G602 机械臂所有者、操作者必须对自己的安全负责。请勿改造机械臂，因擅自进行产品改造而产生的事故或故障，不在本公司保修范围之内，本公司概不负责。

(2) 在理解 PROBOT G602 机械臂使用说明书的“警告标志”的基础上，使用机械臂。

(3) 使用 PROBOT G602 机械臂前要检查以下事项，如有异常则应及时修理或采取其他必要措施。

- 1) 机械臂各轴是否处于零位；
- 2) 电气线缆是否已正确连接；
- 3) 电气线缆外皮有无破损；
- 4) 急停开关是否处于解除状态；
- 5) 机械臂动作有无异常、异响。

(4) 控制机械臂运动时切勿靠近，并且：

- 1) 在操作机械臂前要采用较低的倍率速度调试机械臂的运动，以增加对机械臂的有效控制；
- 2) 在按下电源键之前要考虑到机械臂的运动趋势，预先考虑好避让机械臂的运动轨迹，并确认该线路不受干涉；

(5) 生产运行

- 1) 在开机运行前，须知道机械臂根据所编程序将要执行的全部任务。
- 2) 须知道所有会影响机械臂移动的开关、传感器和控制信号的位置和状态。
- 3) 须知道机械臂控制设备上的紧急停止按钮的位置，准备在紧急情况下按下这些按钮。
- 4) 永远不要认为机械臂没有移动其程序就已经完成。因为这时机械臂很有可能是在等待让它继续移动的输入信号。

1.4 使用环境

PROBOT G602 机械臂使用安全，能最大限度的适应环境，使用时请遵照说明书的指示。请务必遵守本手册中的注意事项。

- (1) 绝不强制地扳动机械臂的轴，否则可能会造成人身伤害和设备损坏。
- (2) 使用完机械臂，应将电源线插头拔掉，并将机械臂放于干燥、常温之处。
- (3) 避免处于泥土和多灰尘的环境，避免长期放置于阳光直射位置，远离高湿度以及较强的振动环境，泥土、废屑、高温会损坏内部器件。
- (4) 以下场合不可使用 PROBOT G602 机械臂：

- 1) 靠近可燃性物质的环境
- 2) 有爆炸可能的环境
- 3) 水中或其他液体中
- 4) 存在腐蚀性、易燃性气体的环境内
- 5) 温度超过 40 摄氏度的环境
- 6) 其他恶劣使用环境

2.1 产品概述

安诺机器人（深圳）有限公司成立于 2017 年 4 月，是一家桌面机械臂解决方案的专业研发和生产商。通过持续的创新，安诺机器人致力于为教育机器人、轻工业机器人和商用机器人提供高性价比、易操作、体验佳的桌面型机械臂产品和解决方案。

PROBOT G602 是安诺机器人推出的一款面向科研和教育领域的六轴机械臂产品，简单易用，配套教学课程，能帮助您提高研究与学习的效率。

2.1.1 产品介绍

PROBOT G602 是一款重量轻、速度快、重复定位精度高、高性价比的消费级桌面型机械臂。外形小巧、体积小，能够高速、高精度的完成上下料、分拣、装配等各项工作。

PROBOT G602 提供串口、网络、IO 等丰富的通讯方式，支持 ROS、Python、C++ 进行二次开发，重复定位精度可达 0.1mm，搭配简单易用的可视化交互软件和仿真平台，支持多种便捷切换的末端执行器，满足不同任务需求。

- 集成化

采用一体化伺服电机，体积小、工作灵活、速度快、精度高；

- 高性能

控制器使用性能强劲的 Xilinx Zynq 芯片，ARM+FPGA 异构 SoC 让二次开发灵活高效，同时采用多轴联动插补控制算法，提高系统的精度和稳定性；

- 扩展性强

提供 ROS/Python/C++ SDK，支持 RS-232/Ethernet/Digital IO 通信扩展，充分发挥 PROBOT G602 的最大潜力；

- 易用性

支持 ROS、ROS-I 框架，丰富的开源功能包，帮助用户快速上手机器人的使用和开发；

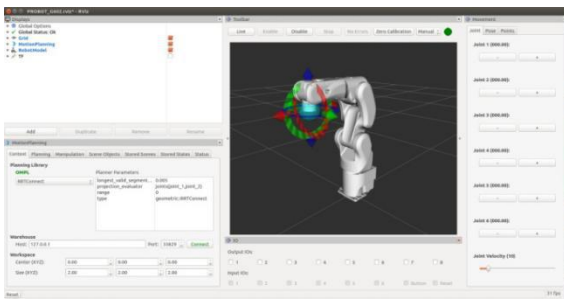
- 配套课程

选配“古月居”出品的 ROS 机械臂开发视频课程与教材，理论与实践结合，快速上手机械臂开发；同时可选课程定制、机器人线上/线下培训等增值服务，提供教育综合解决方案；

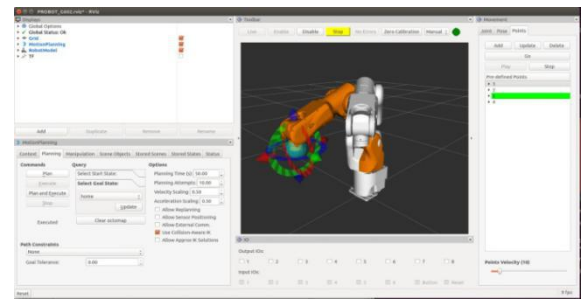
- 持续更新

支持固件更新，软件持续迭代优化，配套应用和教程不断升级。

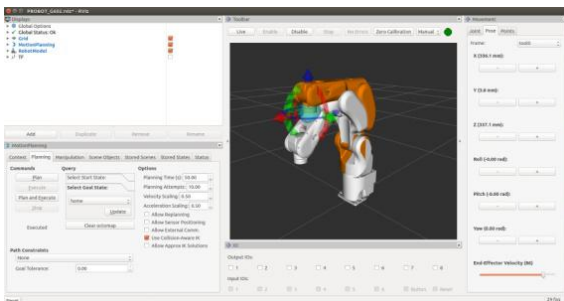
2.1.2 功能特性



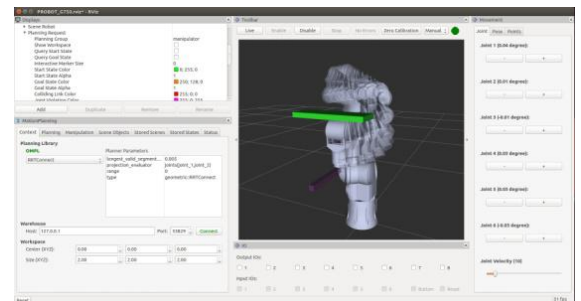
虚拟化平台支持 – PROBOT 机械臂具备完整 URDF 模型描述，支持 ROS、Matlab 和 CoppeliaSim 进行仿真分析和实时监控。



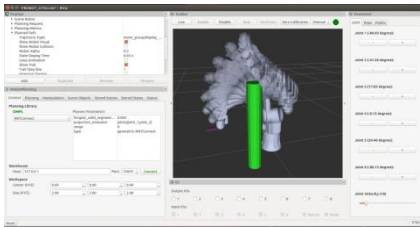
拖动示教 – PROBOT 机械臂可通过拖动三维模型关节实现在线示教功能，并且机械臂位置姿态可以在 Rivz 中实时显示。



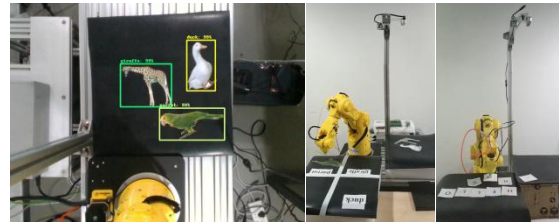
点动控制 – PROBOT 机械臂上位机支持点动控制，可实现对关节及姿态的点动，自由调整机械臂的姿态。



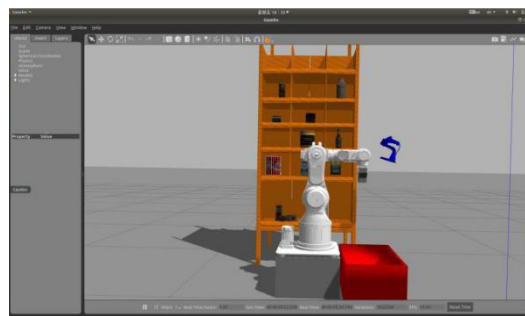
避障规划 – PROBOT 机械臂支持动态避障，通过路径规划实现自动避开机械臂周围的障碍物。



碰撞检测 – PROBOT 机械臂支持在工作空间中添加障碍物模型，并可以在 Rviz 和 Gazebo 里实时查看碰撞区域。



视觉抓取 – PROBOT 机械臂支持通过 RGBD 进行物理环境中物体的识别、定位和抓取。



应用场景 – 智能货架拣选系统支持数字孪生（Digital Twin），可以实现 PROBOT 机械臂的动态避障、货物分拣等综合应用。

2.1.3 产品外观



图 2-1 PROBOT G602 机械臂外观



图 2-2 PROBOT G602 机械臂电控箱外观图

2.2 控制箱说明



图 2-3 控制箱外观图（前）

在底部控制箱正面上，从左到右分别为：复位按钮（RESET）、急停按钮（STOP）、一键启动按钮（BUTTON）；在控制箱后面有红色的电源开关。

（1）急停按钮（STOP）：在紧急情况下，向下按压急停按钮可以及时停止机械臂的运动；再次启动机械臂时需顺时针扭动该按钮，以解除急停状态。

（2）一键启动按钮（BUTTON）、复位按钮（RESET）：分别连接到控制器的数字输入口，对应 ROS 上位机中的 Button 和 Reset 输入端口状态，供用户编程使用。

（3）电源开关：I 为开，O 为关。



图 2-3 控制箱外观图（后）

2.3 IO 接口说明

2.3.1 控制器IO 接口

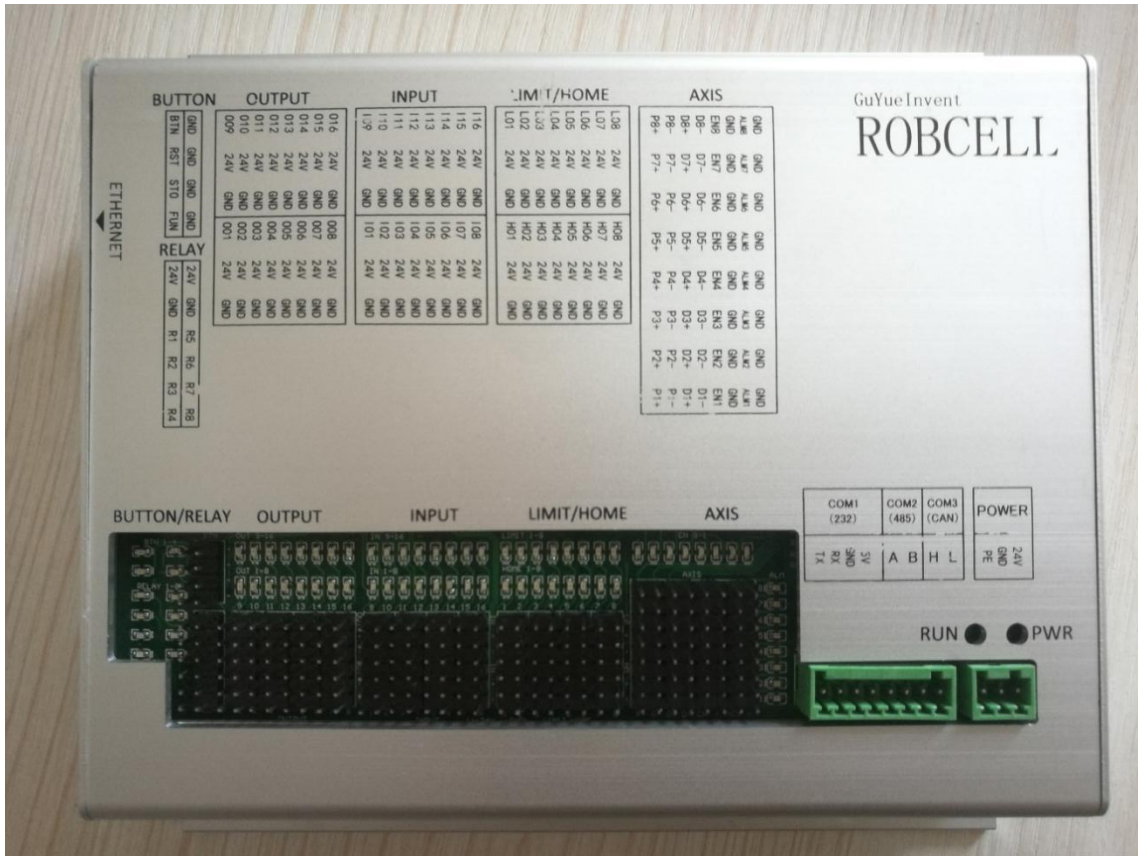


图 2-4 ROBCELL 控制器（预装于控制柜中）

- (1) **BUTTON:** 连接控制柜上的按键，共 4 位；
- (2) **RELAY:** 连接控制柜内的继电器使用，共 8 位；
- (3) **OUTPUT:** 数字 IO 输出口，连接外部输出设备，共 16 位；
- (4) **INPUT:** 数字 IO 输入口，连接外部输入设备，共 16 位；
- (5) **LIMIT/HOME:** 连接机械臂上的限位传感器，各 8 位；

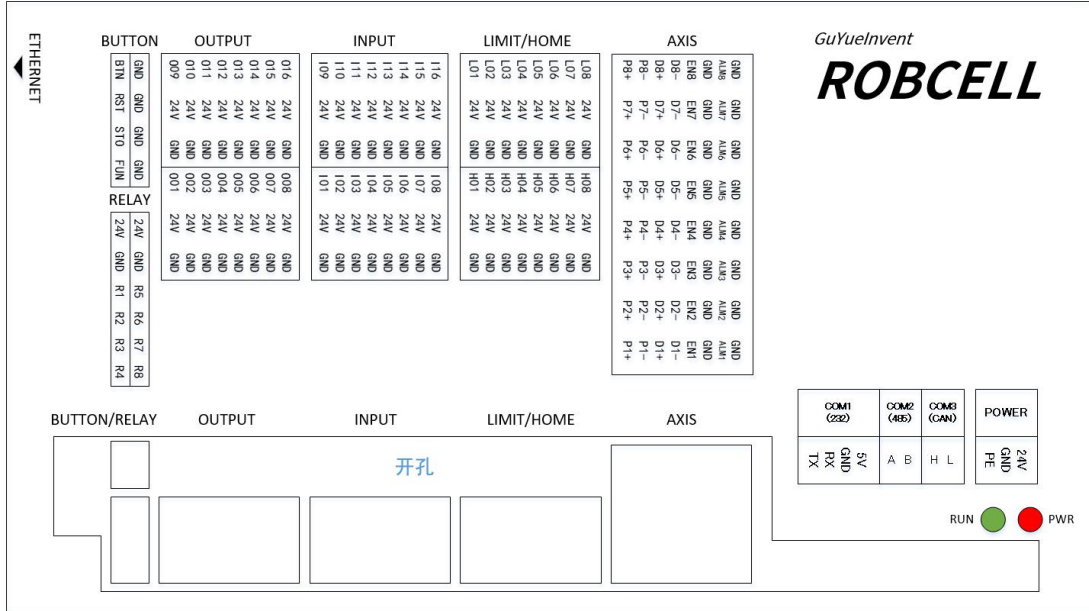


图 2-5 ROBCELL 控制器接口说明（预装于控制柜中）

2.3.2 控制柜 IO 接口

为方便连接外部设备使用，控制柜后端通过 DB9 接出 4 路数字 IO 口（2 路输出、2 路输入），具体引脚分配如下：

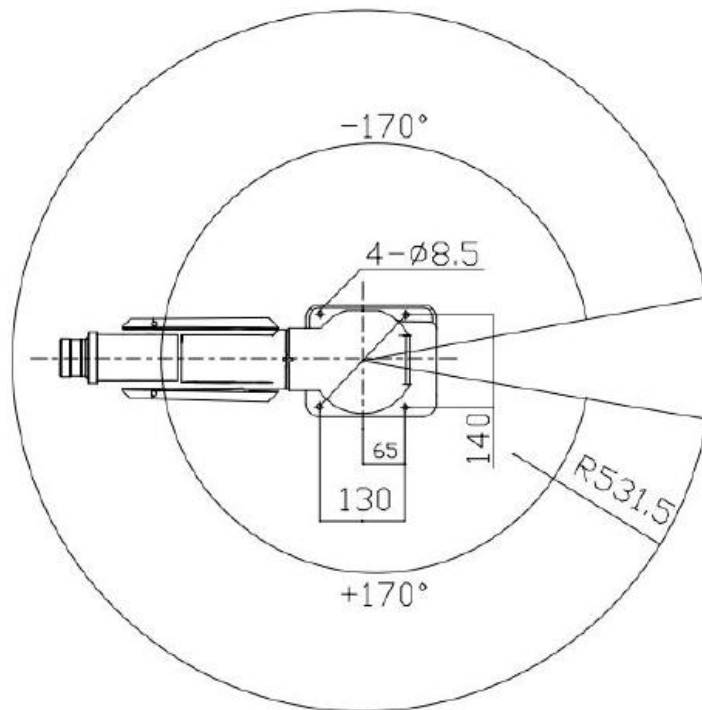
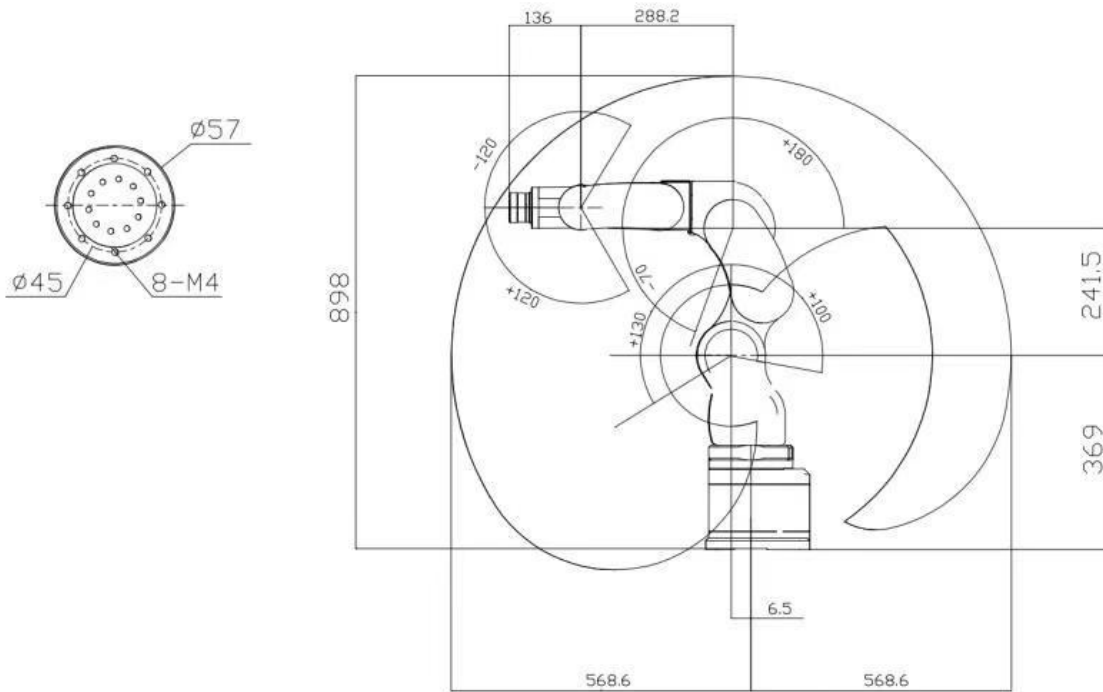
序号	定义	序号	定义	序号	定义
1	RELAY1	4	IN2	7	GND
2	RELAY2	5	+24V	8	GND
3	IN1	6	+24V	9	GND

2.4 硬件参数

机械臂类型		六轴	
轴	运动范围	最大转度	
第 1 轴	$\pm 170^\circ$	370°/s	
第 2 轴	$\pm 110^\circ$	370°/s	
第 3 轴	$+40^\circ \sim -220^\circ$	430°/s	
第 4 轴	$\pm 185^\circ$	300°/s	
第 5 轴	$\pm 125^\circ$	460°/s	
第 6 轴	$\pm 360^\circ$	600°/s	
电源电压	220V/110V	重量	28KG
重复定位精度	$\pm 0.05\text{mm}$	减速装置	谐波减速
电源电压	220V/110V	额定负载	3KG
最大工作半径	540mm	最大负载	4.5KG
电机类型	伺服电机	控制器	ROBCELL A1

2.5 结构尺寸

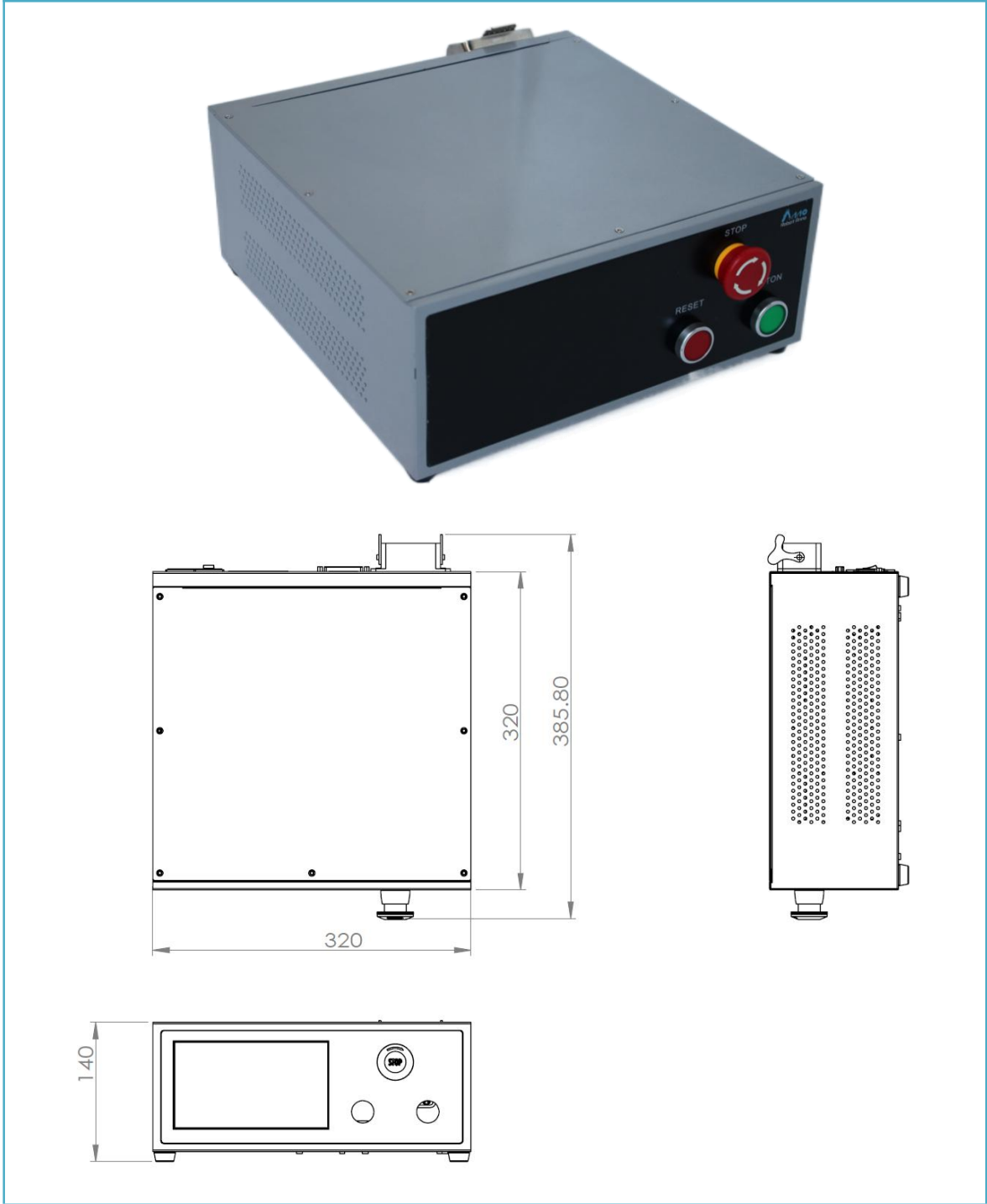
尺寸图



电控箱尺寸

320mm*320mm*140mm

电控箱尺寸图



3 快速上手指南

3.1 硬件接口与连线

使用前，请参考 **2.2 控制箱说明** 了解控制箱接口及面板功能，再按照以下说明完成硬件接线。

请按照以下图示完成控制箱与机械臂、网线、数字 IO 接口的连线：

- (1) 将控制箱端的重载接头（公头）连接至机械臂对应的接头端（母头）；
- (2) 连接机械臂控制箱的电源线至 220V 市电；
- (3) 使用网线连接 PC 机网口与机械臂控制箱网口；
- (4) 在 DB9 连接器上根据需要连接外部 IO 设备，如夹爪、吸盘；
- (5) 将圆孔充电器头与控制柜的圆头相连。

注意

如未使用 IO 功能，第（4）步可以跳过。

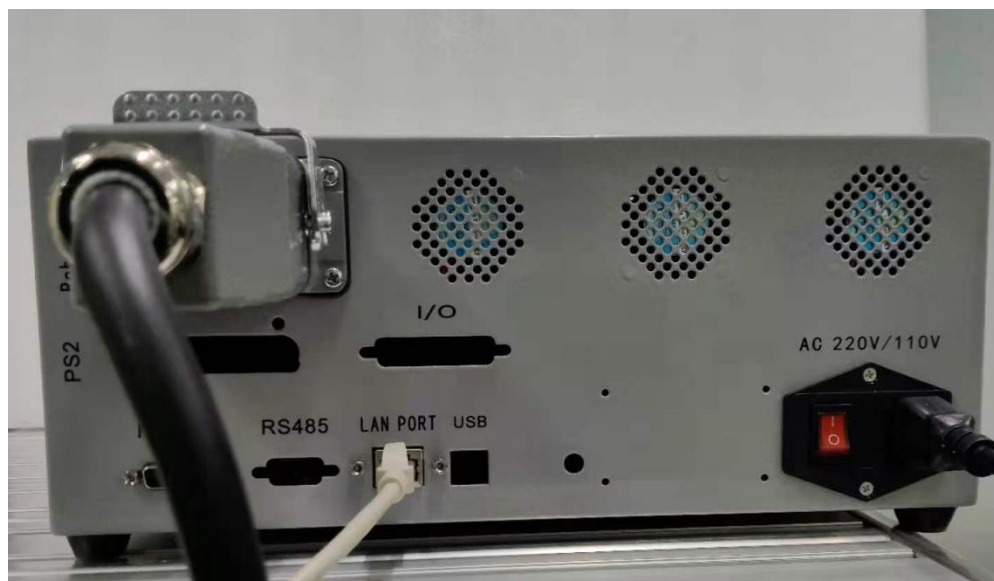


图 3-1 控制柜连线实物图

3.2 启动 ROS2GO

本公司为广大用户提供了即插即用的 ROS2GO 迷你 U 盘，并在 ROS2GO 中安装部署了本产品所需的一整套 ROS 环境和上位机，用户可直接使用 ROS2GO 开始体验之旅！



图 3-2 ROS2GO 迷你 U 盘

(1) 将 ROS2GO 插入 PC，启动/重启 PC，并进入 BIOS 设置使用 **UEFI 模式** 引导 U 盘启动，（不同型号的 PC，进入 BIOS 系统的方式有所不同，具体请参考使用 PC 的启动方式）。

注意

需要在 BIOS 中关闭 Secure Boot（安全启动）功能。

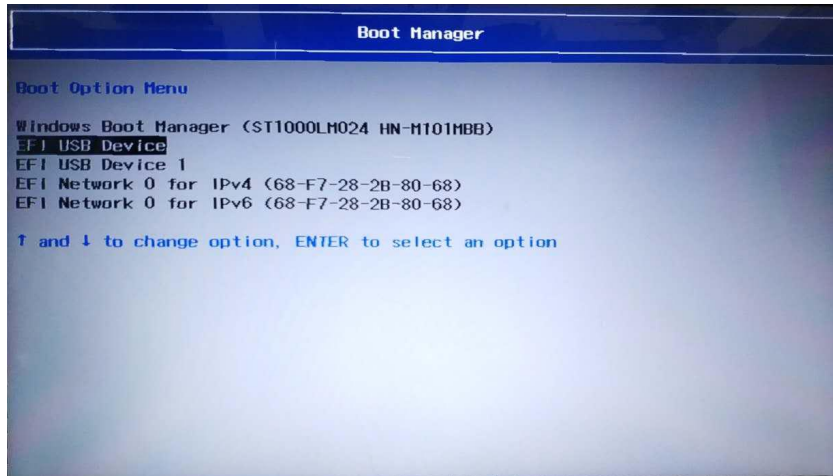


图 3-3 BIOS 设置 U 盘启动示意图

(2) 设置完成后，等待 ROS2GO 完成启动，启动成功后可以看到如下系统桌面。



图 3-4 ROS2GO 系统桌面

3.3 系统上电启动

- (1) 再次检查硬件连线，确保连接无误。
- (2) 将控制箱后端电源插座接通外部电源，并按下正面电源开关给控制箱上电。

(3) 上电后等待约 20s，将听到蜂鸣器连续响两声，持续时间较短促，表明 PROBOT G602 软硬件系统启动成功。

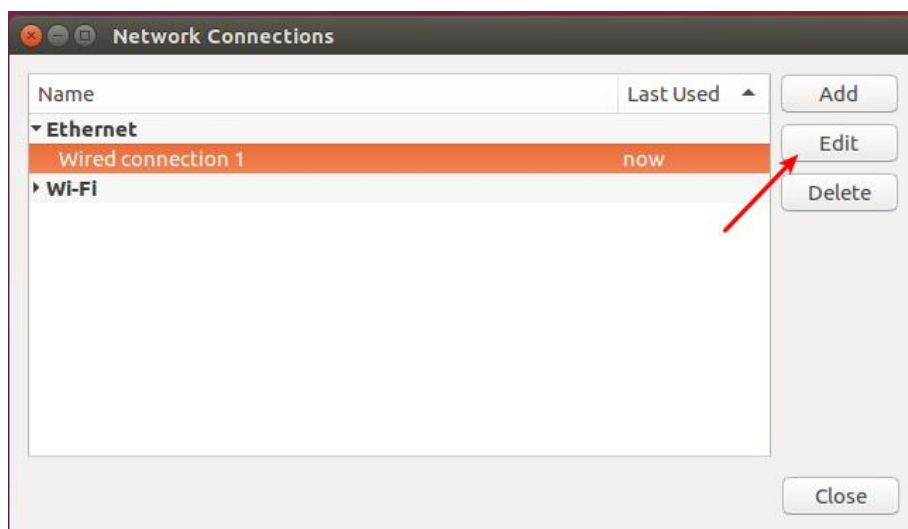
(4) 释放急停按键。

3.4 设置网络连接

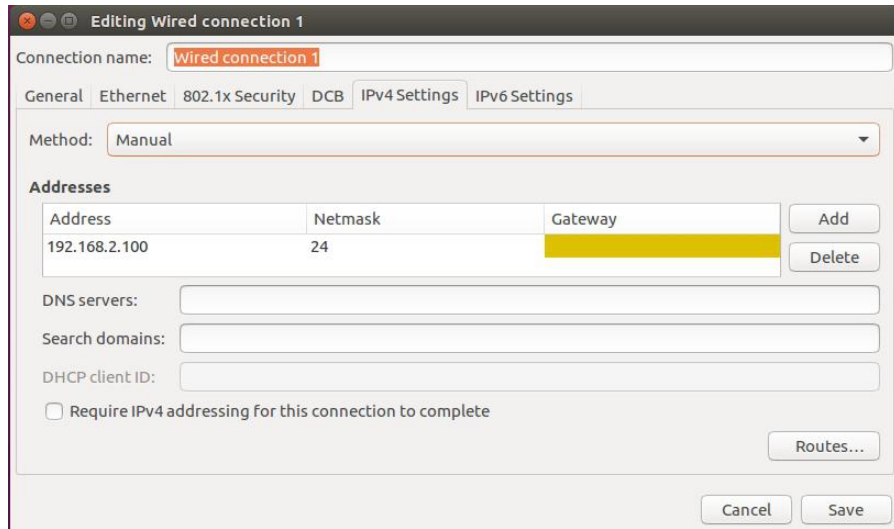
注意

控制器和上位机通过网线连接，两者需要处于同一局域网下，控制器的 IP 固定为 192.168.2.123，ROS2GO 出厂时已经配置好 IP，但是由于不同主机的网卡不同，可能会有变化，使用前请确保控制器与上位机网络连接成功。

进入 ROS2GO 桌面后，点击右上角的网络标志，选择网络配置：



进入配置界面后按照下图所示的内容进行配置：



保存配置信息并重新连接有线网络，看到连接成功的信息则说明配置完成，可以使用 ping 指令确认：

```
ping 192.168.2.123
→ ~ ping 192.168.2.123
PING 192.168.2.123 (192.168.2.123) 56(84) bytes of data.
64 bytes from 192.168.2.123: icmp_seq=1 ttl=64 time=0.254 ms
64 bytes from 192.168.2.123: icmp_seq=2 ttl=64 time=0.315 ms
64 bytes from 192.168.2.123: icmp_seq=3 ttl=64 time=0.315 ms
64 bytes from 192.168.2.123: icmp_seq=4 ttl=64 time=0.336 ms
64 bytes from 192.168.2.123: icmp_seq=5 ttl=64 time=0.341 ms
```

3.5 启动 ROS 上位机

注意

控制器上电后，PROBOT 软硬件系统启动需要约 20s，启动完成后蜂鸣器将连续响两声。
请确保听到连续响声后再启动 ROS 上位机！否则需重启 ROS 上位机。

系统启动完成后，可以双击桌面的图标“PROBOT Studio-G602”来启动 G602 机械臂 ROS 上位机。



也可以使用如下命令启动 ROS 上位机：

```
$ roslaunch probot_bringup probot_g602_bringup.launch
```

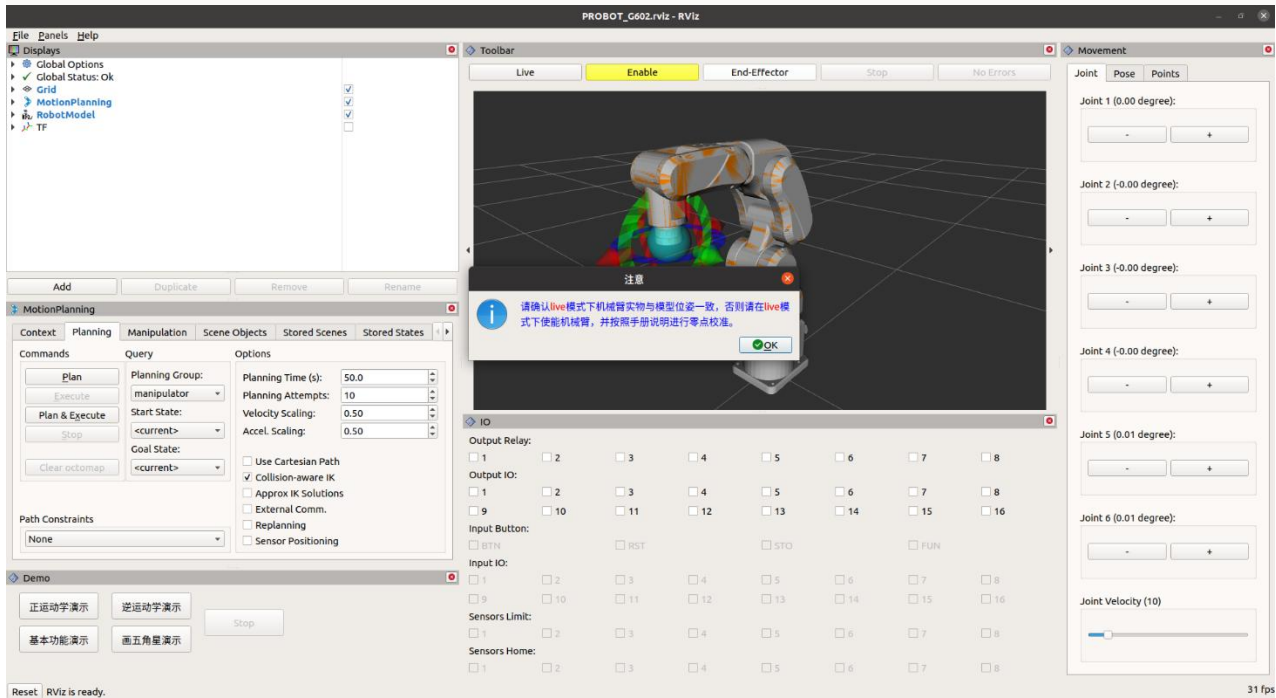


图 3-5 真机运行

3.6 仿真/在线切换

ROS 上位机启动后默认进入真机控制环境，可点击控制栏中的仿真/真机切换按键，切换到仿真环境。按键显示 Live 表示当前处于真机控制环境，Sim 则表示仿真环境：



图 3-6 切换至真机模式

3.7 机械臂使能

点击控制栏的使能/除能按键，使能机械臂控制，按键黄色指示灯变为绿色：

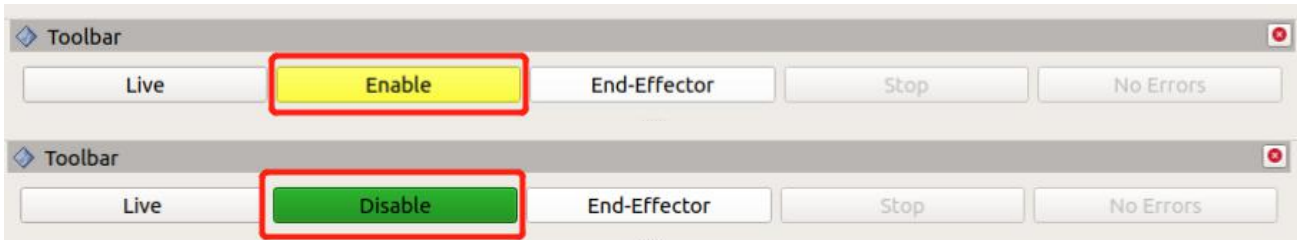


图 3-7 使能机器人控制

3.8 调整机械臂初始位姿

PROBOT G602 机械臂的关节无法手动推动，如果实物机器人与模型位姿不符，请参考零位标定部分，使用该功能重新标定有位置偏差的轴。

3.9 机械臂点动控制

点击上位机右侧的“Movement”区域里对应轴/位姿的“+”、“-”符号，即可控制机械臂在关节空间/工作空间点动运动，如果模型和实物同步运动，说明系统启动成功。

更多使用方法请参考下文的操作说明。

4 机械臂的使用

4.1 基本操作

4.1.1 调整机械臂初始位姿

PROBOT G602 机械臂的部分关节无法手动推动，请参考零位标定部分，使用该功能重新标定有位置偏差的轴。

4.1.2 启动机械臂

- (1) 再次检查硬件连线，确保连接无误。
- (2) 将控制箱后端电源插座接通外部电源，并按下正面电源开关给控制箱上电。
- (3) 上电后等待约 20s，将听到蜂鸣器连续响两声，持续时间较短促，表明 PROBOT G602 软硬件系统启动成功。
- (4) 释放急停按键。

注意

- 启动机械臂前要检查机械臂外部电线遮盖物及外包装有无破损，如有异常则应及时修理或采取其他必要措施。
- **必须等待蜂鸣器连续响两声后，才能继续后续操作！** 否则 PROBOT G602 软硬件系统尚未启动完成，ROS 上位机将无法与控制器建立连接。
- 急停按钮被按下表示处于急停状态，此时顺时针扭动可以解除急停状态。

危险

- 请不要在机械臂工作状态下移动机械臂，如要移动请先关闭机械臂电源，断电之后再移动机械臂。

4.1.3 关闭机械臂

操作结束后，请按照下图说明，控制机械臂回到初始位姿“home”，然后点击“Disable”，按下控制柜急停按钮，关闭控制柜电源，完成关机。

4.1.4 急停

在紧急情况下，向下按压急停按钮可以及时停止机械臂的运转；再次运转机械臂时顺时针扭动该按钮解除急停状态；

危险

- 紧急情况下，马上按下急停键，若不能及时制动机器臂，则可能引发人身伤害或设备损坏事故。



急停键



解除急停状态

- 请不要在机械臂工作状态下移动和维修机械臂，如要移动和维修请先关闭机械臂电源，断电之后再执行这些操作。

4.2 上位机 ROS 环境搭建

PROBOT 机械臂的使用基于 Ubuntu 和 ROS 环境。如您使用本产品配备的定制化 ROS2GO，则可直接启动 ROS2GO，使用预安装好的上位机 ROS 环境。否则，需要在自己的 PC 端安装 Ubuntu 系统和 ROS 相关的软件，请参考以下搭建步骤。

4.2.1 安装 Ubuntu16.04/18.04 系统

如您使用本产品配备的定制化 ROS2GO，则请跳过本节，直接启动 ROS2GO 使用预安装好的上位机 ROS 环境。

(1) 可采用双系统硬盘安装或虚拟机安装，硬盘空间需在 20GB 以上，Ubuntu 镜像下载地址为：<http://www.ubuntu.com/download/>

(2) 完成安装后建议国内用户修改系统软件源为阿里云/华为云（根据具体情况选择）：

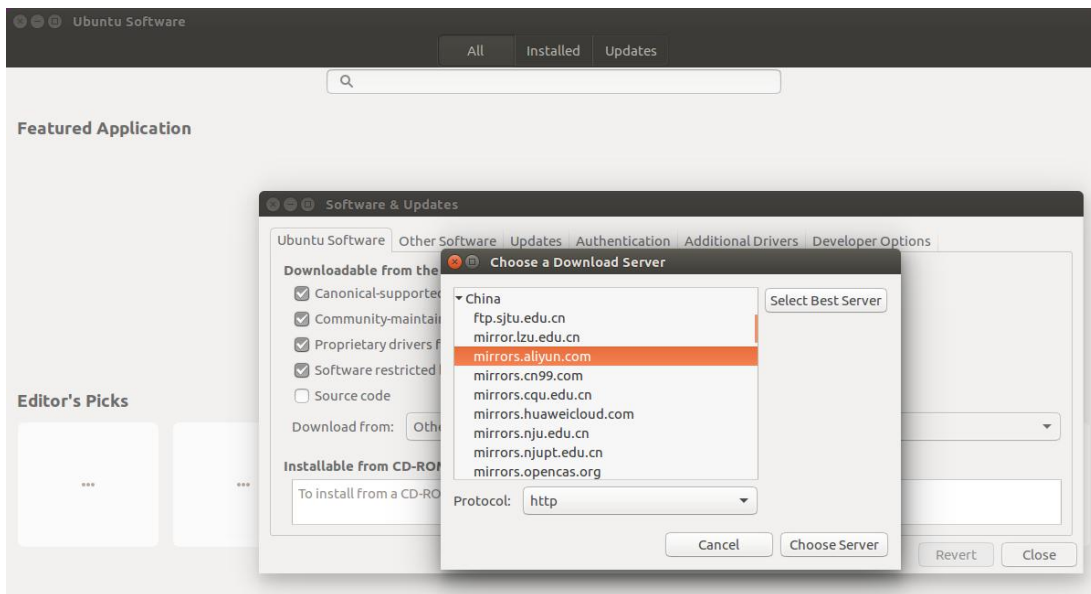


图 4-1 修改软件源

(3) 在终端输入如下命令，更新系统软件源：

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

4.2.2 使用上位机安装助手

如您使用本产品配备的定制化 ROS2GO，则请跳过本节，直接启动 ROS2GO 使用预安装好的上位机 ROS 环境。

安装本产品 ROS 上位机，请先下载/获取 PROBOT Studio 上位机安装助手 `probot_studio_setup.sh.x`，以及安装包（如：`probot_studio_v3_2_1-ubuntu18_04.zip`）。

(1) 打开上位机安装助手

将上位机安装助手及安装包文件放置到同一文件夹下。

确认安装助手具有可执行权限。选中 `probot_studio_setup.sh.x` 并点击右键，点击“属性”-“权限”，勾选“允许作为程序执行文件”。



图 4-2 查看/修改可执行权限

打开终端输入如下命令，启动上位机安装助手。首次启动，将自动安装该程序运行所需的依赖包，请耐心等待。

```
$ ./probot_studio_setup.sh.x
```

(2) 选择上位机安装包

请注意选择与当前 Ubuntu 系统版本对应的安装包，当前支持 Ubuntu 16.04/18.04。单击选中后点击“OK”。

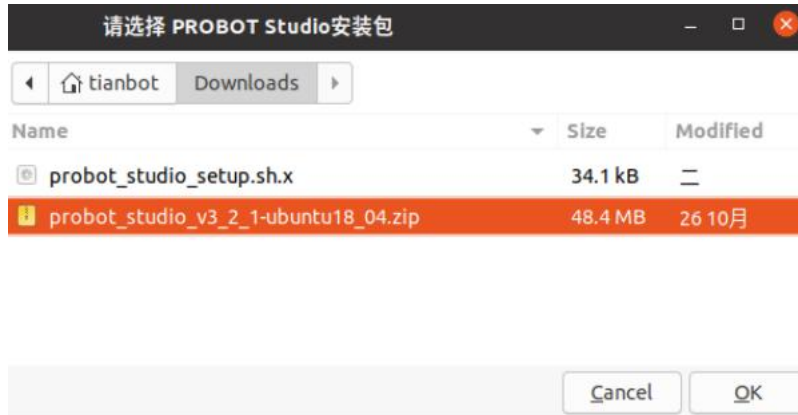


图 4-3 选择上位机安装包

(3) 选择安装项

上位机的安装项包括：

1. 仅 ROS：选择此项仅安装 ROS 操作系统。
2. 仅 PROBOT Studio 依赖包：选择此项仅安装 PROBOT Studio 上位机所需的依赖包。
3. 仅 PROBOT Studio：选择此项仅安装 PROBOT Studio 上位机。
4. 全部：选择此项将依次安装 ROS 操作系统、PROBOT Studio 上位机及其所需依赖包。

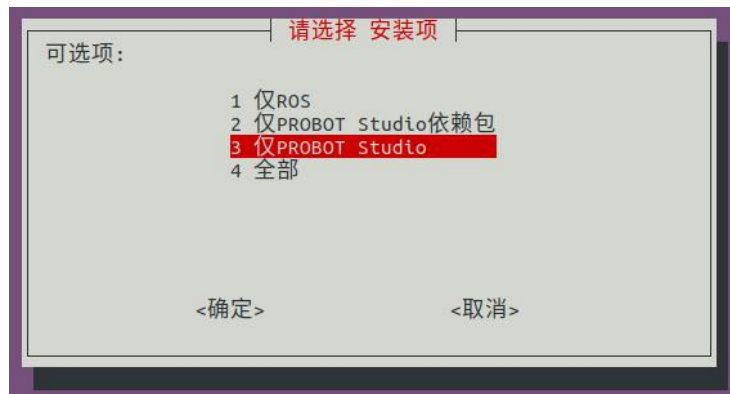


图 4-4 选择安装项

此界面仅支持通过键盘方向键移动光标。按上/下键切换可选项，按左/右键可切换至“确定/取消”，按回车键进入。

(4) 选择机械臂型号

⚠ 注意

请根据实际机械臂型号选择，安装后该上位机仅支持本次选中的机械臂。

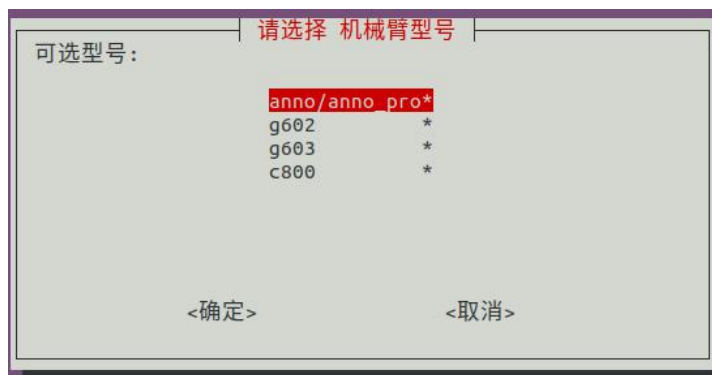


图 4-5 选择机械臂型号

(5) 开始安装

安装所需时长取决于网络环境，请耐心等待。若升级成功，终端将打印如下信息。若耗时过长，请尝试更换网络。

```
[ 94%] Linking CXX shared library /home/tianbot/probot_g602_ws/devel/lib/libgazebo_world_plugin_loader.so
[ 94%] Built target gazebo_version_helpers
Scanning dependencies of target gazebo_grasp_fix
[ 95%] Linking CXX executable /home/tianbot/probot_g602_ws/devel/lib/aruco_ros/single
[ 96%] Building CXX object probot_g602/tools/gazebo-pkgs/gazebo_grasp_plugin/CMakeFiles/gazebo_grasp_fix.dir/src/GazeboGraspFix.cpp.o
[ 96%] Building CXX object probot_g602/tools/gazebo-pkgs/gazebo_grasp_plugin/CMakeFiles/gazebo_grasp_fix.dir/src/GazeboGraspGripper.cpp.o
[ 96%] Built target gazebo_world_plugin_loader
[ 96%] Built target single
[ 97%] Linking CXX executable /home/tianbot/probot_g602_ws/devel/lib/object_color_detector/object_detector
[ 97%] Built target object_detector
[ 98%] Linking CXX executable /home/tianbot/probot_g602_ws/devel/lib/probot_grasping/graspingDemo
[ 98%] Built target graspingDemo
[100%] Linking CXX shared library /home/tianbot/probot_g602_ws/devel/lib/libgazebo_grasp_fix.so
[100%] Built target gazebo_grasp_fix
[INFO ]: 已安装 PROBOT Studio 到 /home/tianbot/probot_g602_ws, 添加快捷启动图标到桌面, 可双击启动上位机
```

图 4-6 安装成功

4.2.3 设置PC端IP地址

控制器与上位机通过网线通信，使用前需手动配置PC端IP地址，使其与控制器处于同一局域网中。控制器默认IP为192.168.2.123，PC端IP地址应配置为192.168.2.XXX，如192.168.2.101。

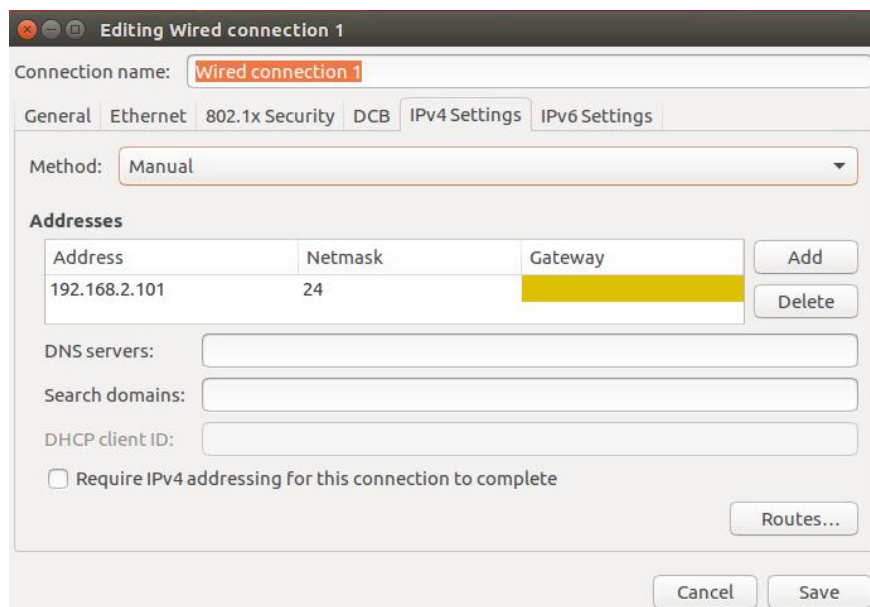


图 4-7 设置静态 IP

4.3 上位机功能介绍

首先，介绍上位机主界面分区及其功能，共分为 6 个区域：

- 可视化插件配置区：
- ROS MoveIt!可视化控制区：
- 按键控制区：

仿真/真机环境切换键、机器人控制使能键、除能键、停止键、清除错误键、零位标定键、状态指示灯；

- 模型动态显示区：

显示机器人当前实际位姿以及目标位姿，使用鼠标拖拽机器人末端拖动球可改变目标位姿；

- IO 控制区：

数字 IO 的输出控制和输入状态显示；

- 运动控制区：

关节空间点动，工作空间点动和示教点功能。

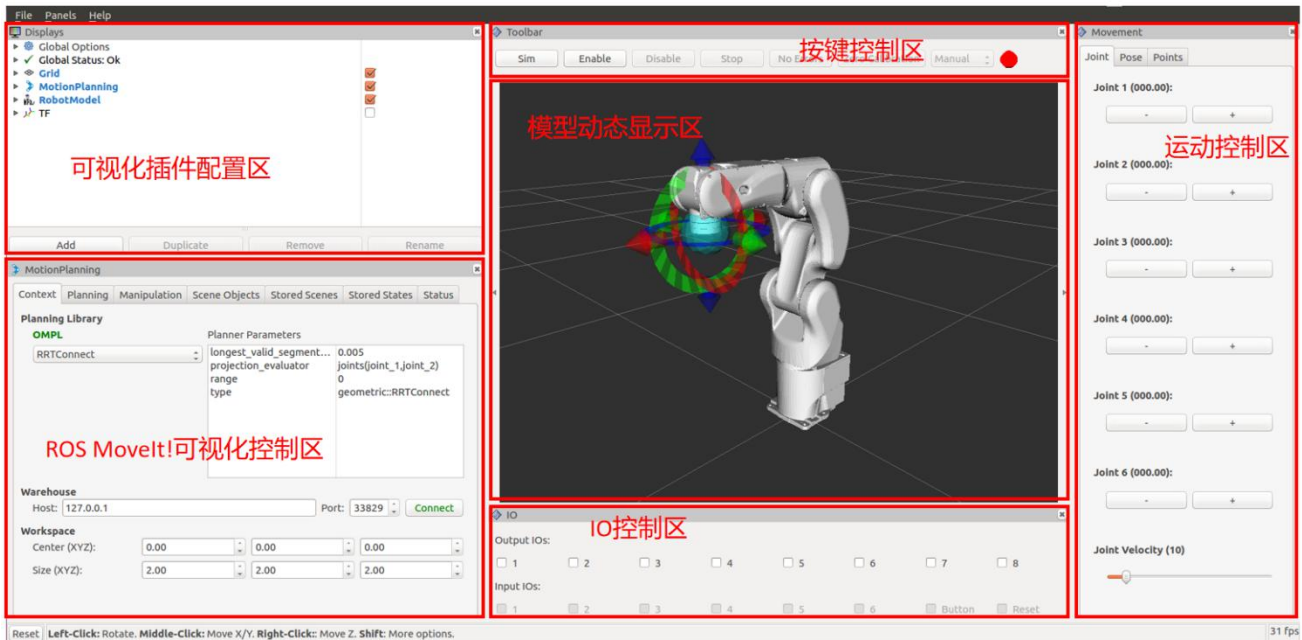


图 4-8 上位机主界面

注意

根据实际机械臂类型，模型动态显示区将显示对应的机械臂模型。但上位机主界面与上图基本一致。

4.3.1 控制栏按键功能

(1) 仿真/真机环境切换

点击该按钮可进行仿真/真机环境的切换，请务必在 **Disable** 状态下执行切换操作。**Sim** 表示当前处于仿真环境，**Live** 表示当前处于真机控制环境：

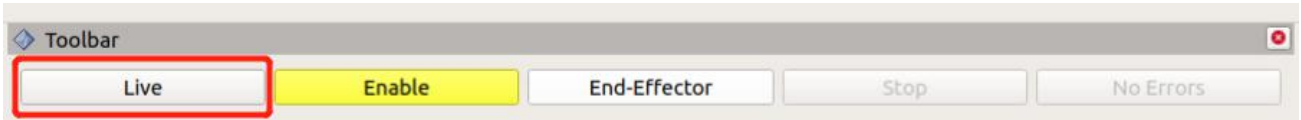


图 4-9 仿真/真机环境切换

(2) 机器人控制使能/除能

点击 **Enable** 可使能机器人控制，点击 **Disable** 除能（在仿真环境下无需使能）。

状态指示灯显示红灯表示当前机器人处于控制除能状态，显示绿灯表示当前机器人处于控制使能状态：

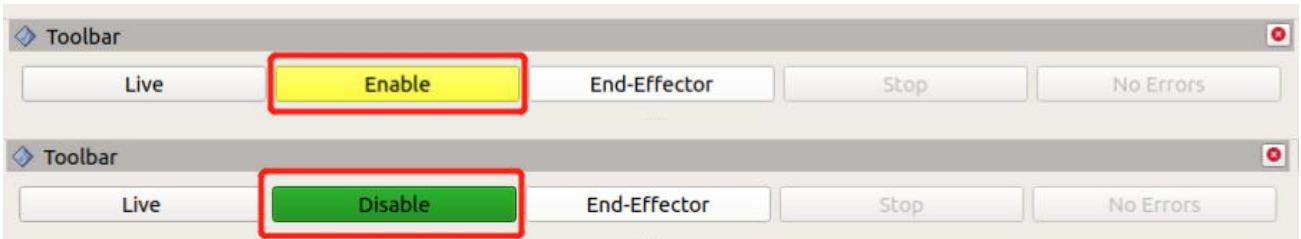


图 4-10 机器人控制使能/除能

(3) 终端执行器

点击 **End-Effector** 即可控制机器人终端执行器-夹爪或吸盘的工作状态如机器人末端安装夹爪，点击 **End-Effector**，夹爪闭合，再次点击，夹爪张开（如夹爪未动作，可先打开气泵，再打开电磁阀）。

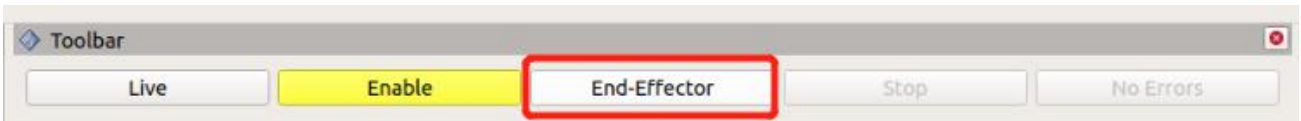


图 4-6 机器人终端执行器

使用前需在 **ConfigPanel** 页面下，设置气泵和电磁阀对应的 IO 口，出厂默认设置气泵连接 IO1，电磁阀连接 IO2，如下图。

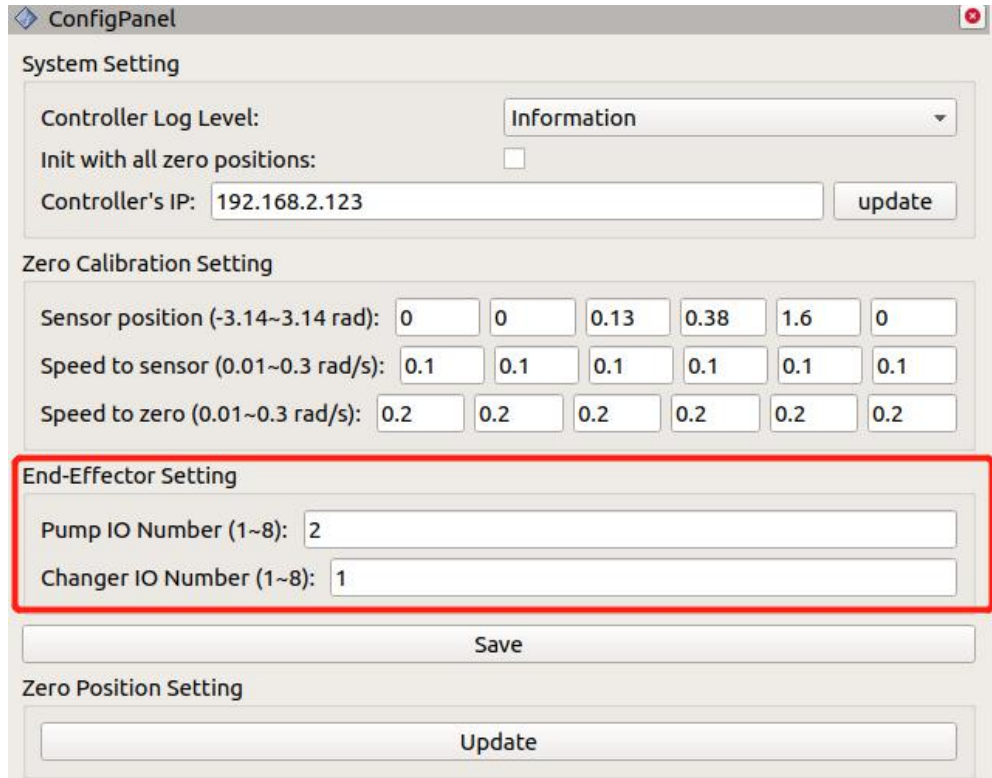


图 4-7 机器人终端执行器 IO 设置

4.3.2 拖动示教

(1) 将鼠标定位在机器人模型终端的控制球上，长按鼠标左键可拖拽机器人，鼠标松开时显示黄色的机器人模型即为目标位姿，而银白色的机器人模型为当前实际位姿。

(2) 设置规划的最大时间限制和机器人运动的速度和加速度（百分比，1 表示 100%，即设置为最大值）。

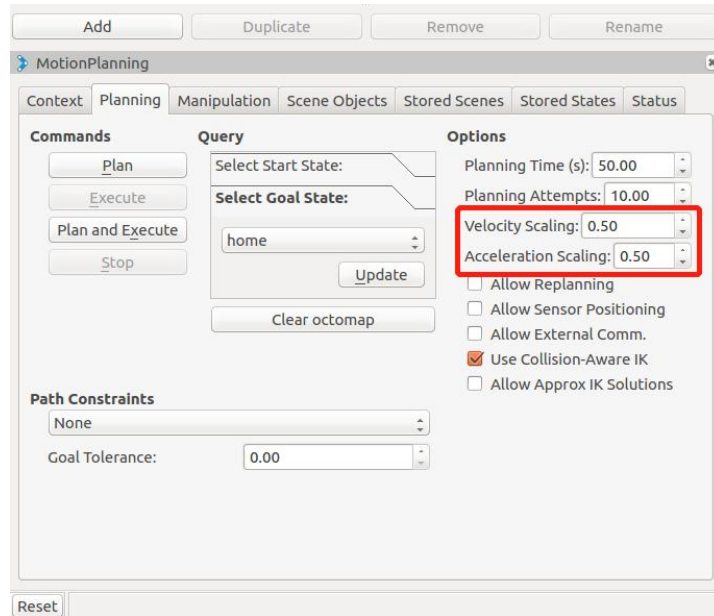


图 4-8 拖动示教的速度与加速度设置

(3) 点击 ROS MoveIt! 可视化控制区的 Planning 标签页中的 Plan and Execute 按键，机器人模型即开始运动（运动过程中，Plan and Execute 按键变灰并且不可操作，需等待当前动作完成）。

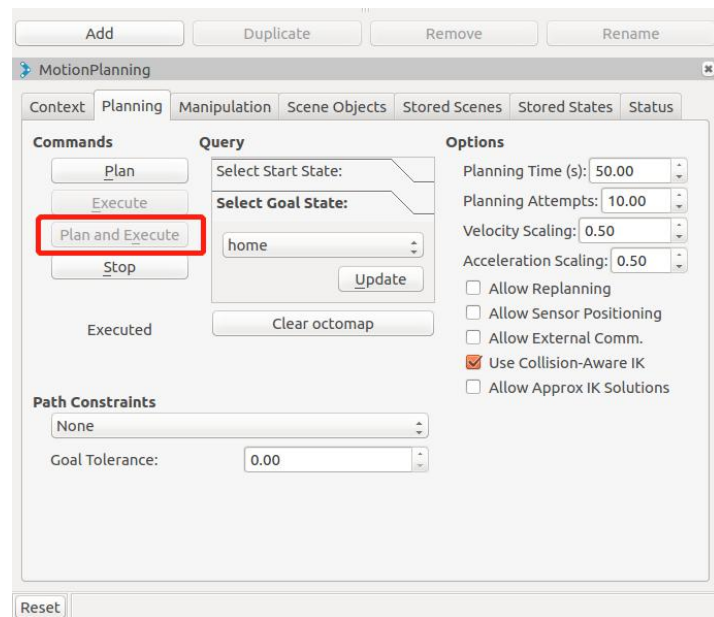


图 4-9 拖动示教执行

(4) 机器人从当前位姿运动到目标位姿，直到界面显示两个模型完全重合，机器人运动结束。

4.3.3 关节空间点动

- (1) 关节点动的所有功能在右侧运动控制区的 **Joint** 标签页中。
- (2) 通过长按+、-可控制关节运动，下侧的 **Joint Velocity** 滑条可调节运动速度。

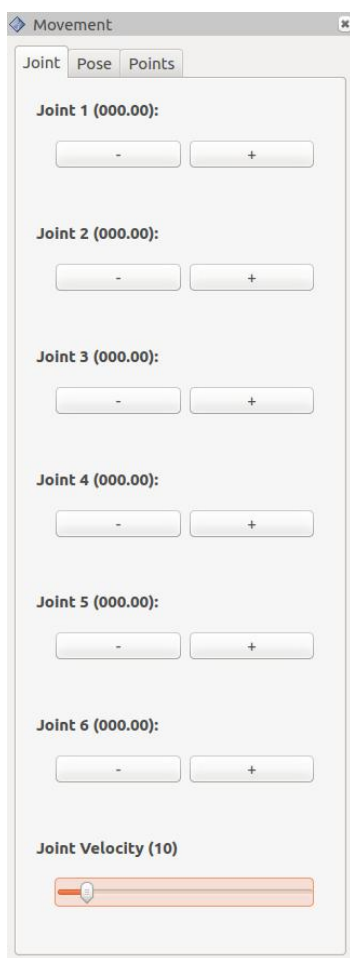


图 4-10 关节空间点动控制

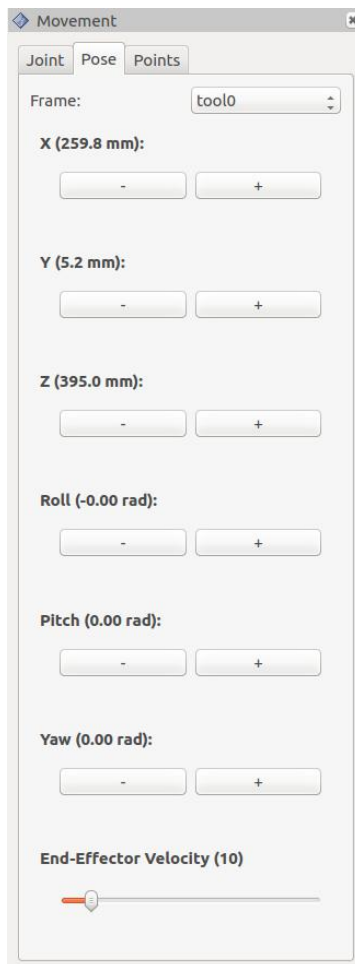


图 4-11 工作空间点动控制

4.3.4 工作空间点动

- (1) 空间点动的所有功能在右侧运动控制区的 **Pose** 标签页中。
- (2) 通过长按+、-可控制机器人末端执行工作空间点动，下侧的 **End-Effector Velocity** 滑条可调节运动速度。

4.3.5 示教点保存与运行

示教点的保存与运行功能在右侧运动控制区的 **Points** 标签页中。

(1) **Add**: 添加示教点。

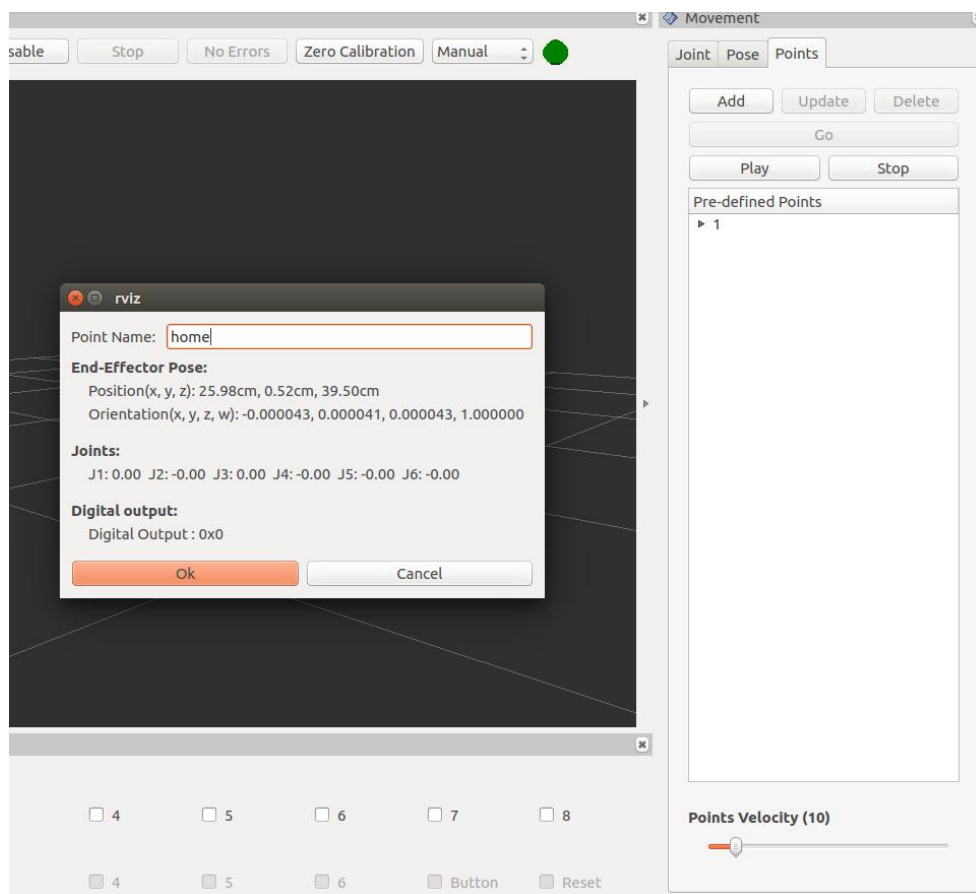


图 4-12 添加示教点

点击 **Add** 按键后会弹出添加示教点的对话框，自动检测机械臂的当前位姿和 IO 状态，同时能添加操作前的延时，并保存为自定义名称的示教点，保存成功后会放置下方的示教点列表中。

(2) **Update**: 更新当前示教点。

在机械臂任意位姿下，选中示教点列表中的任意示教点，点击 **Update** 按键，可以将该示教点更新为机器人的当前位姿和 IO 状态。

(3) **Delete**: 删除当前示教点

选中示教点列表中的任意示教点，点击 **Delete** 按键，可以删除该示教点。

(4) **Go**: 运动到指点的示教点。

选中示教点列表中的任意示教点，点击 **Go** 按键，可以让机械臂运动到该示教点保存的位姿及 IO 状态。

(5) **Play**: 循环运行示教点列表。

点击 **Play** 按键后，从第一个示教点开始循环运行列表中的示教点，点击 **Stop** 按键之后停止运行，当前运行的目标点会变成绿色背景。

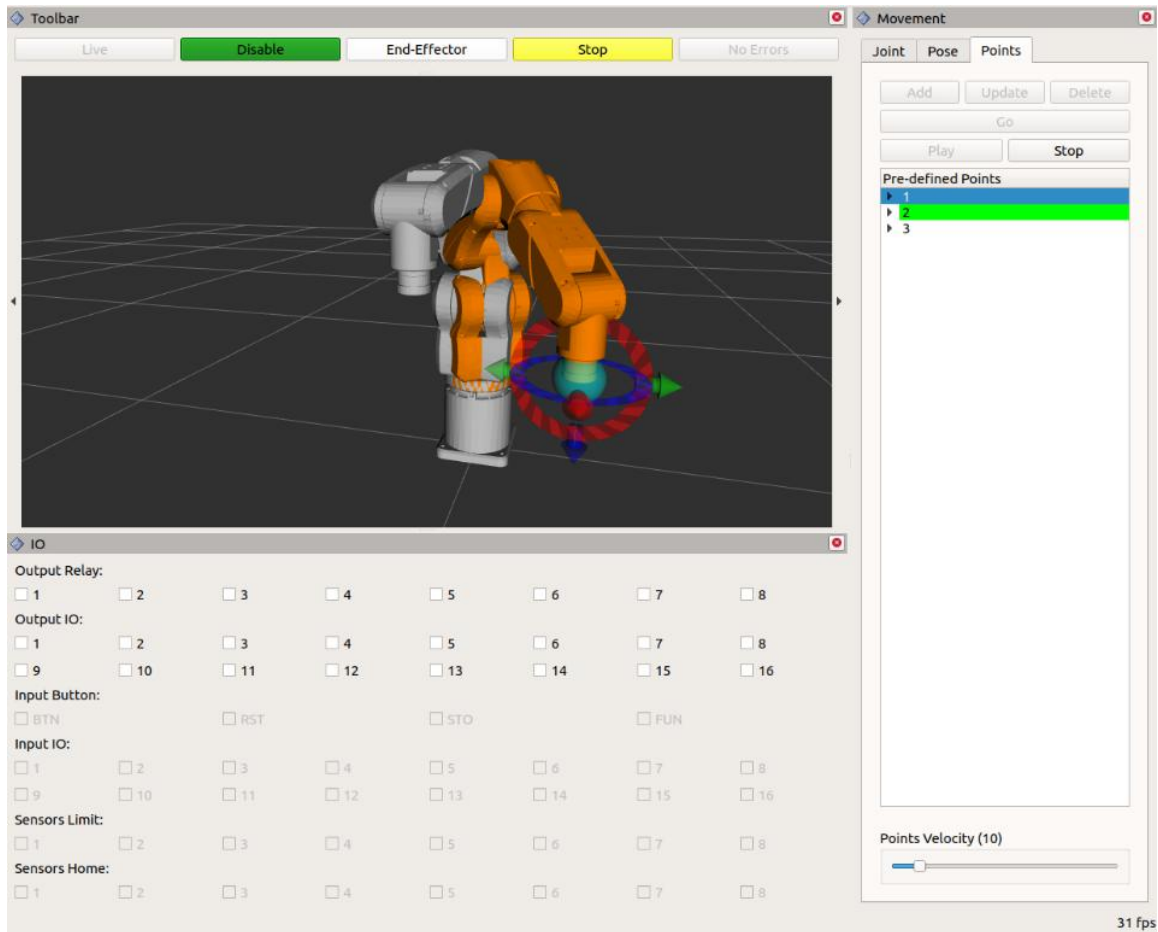


图 4-13 运行示教点

(6) **Stop**: 停止循环运行示教点列表。

(7) **Points Velocity**: 调整示教点运动的速度。

4.3.6 示教点离线运行

示教点运行支持离线模式。参考上一小节，通过上位机添加并保存示教点。保存示教点信息后，即可支持示教点离线运行。具体操作如下：

- (1) 控制箱上电，无需启动上位机。
- (2) 长按（约 3s）控制箱一键启动按钮（**BUTTON**），即可使能机械臂控制（和上位机控制栏的使能按键功能一致）。
- (3) 短按控制箱一键启动按钮（**BUTTON**），即可启动示教点离线运行功能。系统将自动读取离线保存的示教点信息，并逐个执行，实现示教点列表循环运行（和上位机右侧运动控制区的 **Points** 标签页的 **Play** 功能一致）。
- (4) 短按控制箱复位按钮（**RESET**），机械臂运动减速停止。在紧急状况下，可按下急停按钮（**STOP**），机械臂将立即停止运动。

4.3.7 IO 控制

IO 控制功能在 IO 控制区，数字 IO 分为以下几类：

- (1) **Output Relay**: 连接控制柜内的继电器使用，共 8 位；
- (2) **Output IO**: 数字 IO 输出口，连接外部输出设备，共 16 位；
- (3) **Input Button**: 连接控制柜上的按键，共 4 位；
- (4) **Input IO**: 数字 IO 输入口，连接外部输入设备，共 16 位；
- (5) **Sensors Limit/Home**: 连接机械臂上的限位传感器，各 8 位；



图 4-14 IO 控制

鼠标左击想要控制的输出端口，将改变其输出电平，端口号前的方框中打钩代表使能数字输出口输出（低电平有效）；

在 Input 一栏，可看到数字输入口的电平信号，端口号前的方框中打钩代表数字输入端口为有效电平（低电平有效）。

4.3.8 零位标定

出厂前将对机械臂预先标定零位。如发现上位机模型显示区的机械臂位姿与实际不符，可使用零位标定功能进行同步。如有疑问，请立即与售后部联系，勿擅自拆卸机器。

零位标定的具体步骤如下：

- 1.使用运动控制区的关节空间点动控制功能（Joint），长按+/-按钮控制轴移动，使机械臂各关节都处在零位位置。

- 2.点击上位机左上角的 Panels，点击 Add new panel，选择 ConfigPanel 打开参数配置界面，点击 Update，零位标定完成，**重启控制柜即可生效。**

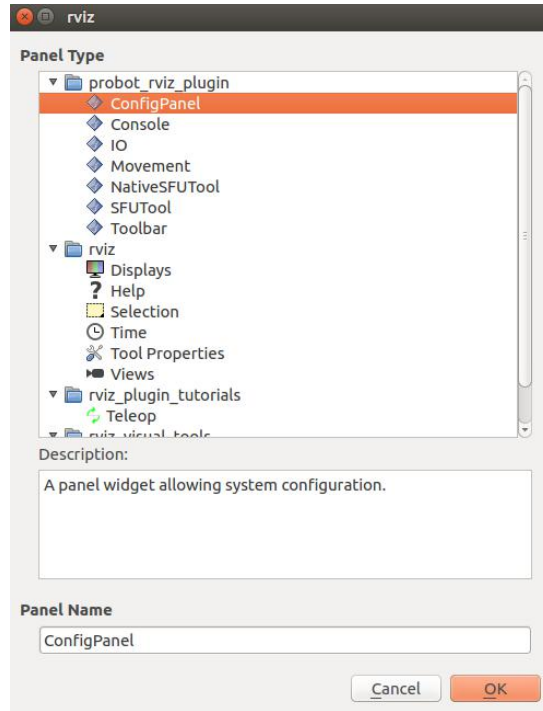


图 4-15 打开 ConfigPanel 配置面板

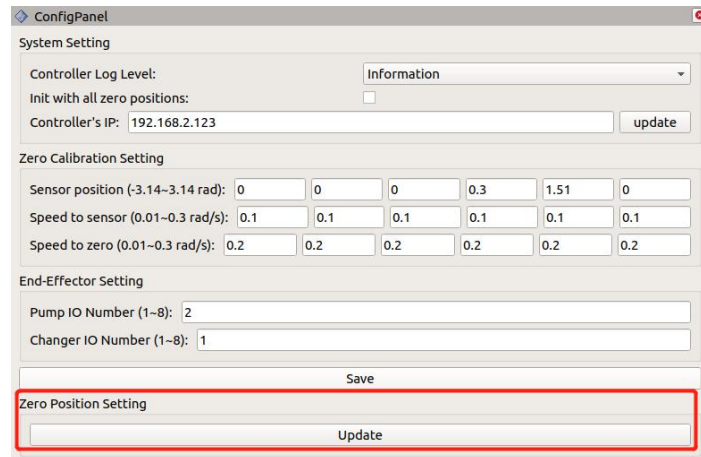


图 4-16 ConfigPanel 配置界面

4.4 单机仿真

上位机 ROS 环境搭建完成后即可在 PC 上进行单机仿真，无需连接控制器和机器人。

使用以下命令，启动上位机并进入仿真控制环境：

```
$ roslaunch probot_bringup probot_g602_bringup.launch sim:=true
```

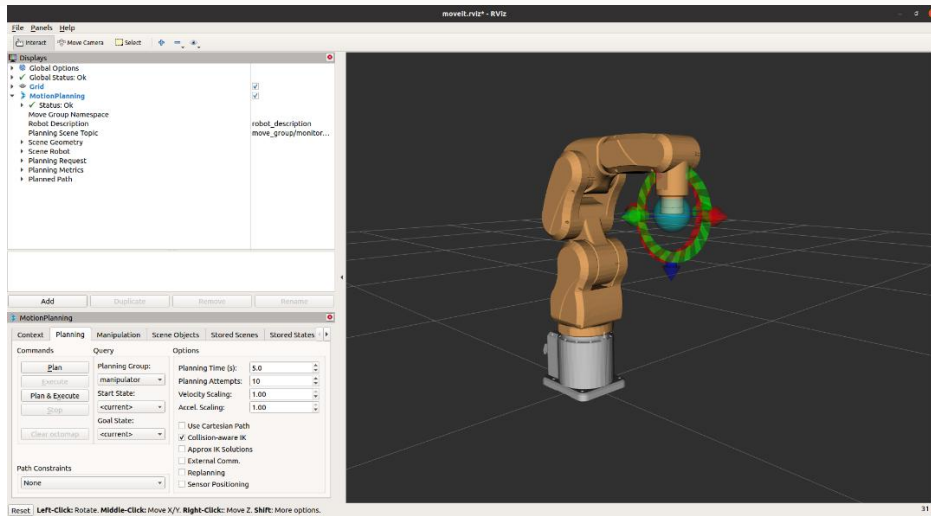


图 4-17 单机仿真启动

⚠ 注意

使用以上命令启动的上位机界面和真机运行下启动的界面有所差别，更加精简。

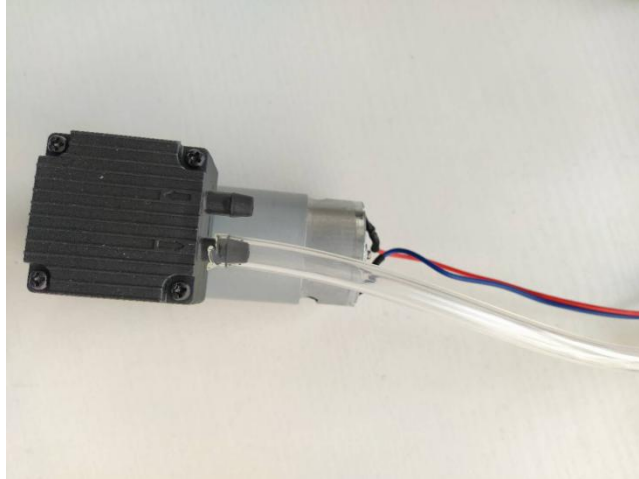
4.5 真机运行

请参考 [3 快速上手指南](#)

更多例程请参考 `probot_g602_demo` 功能包中的程序。

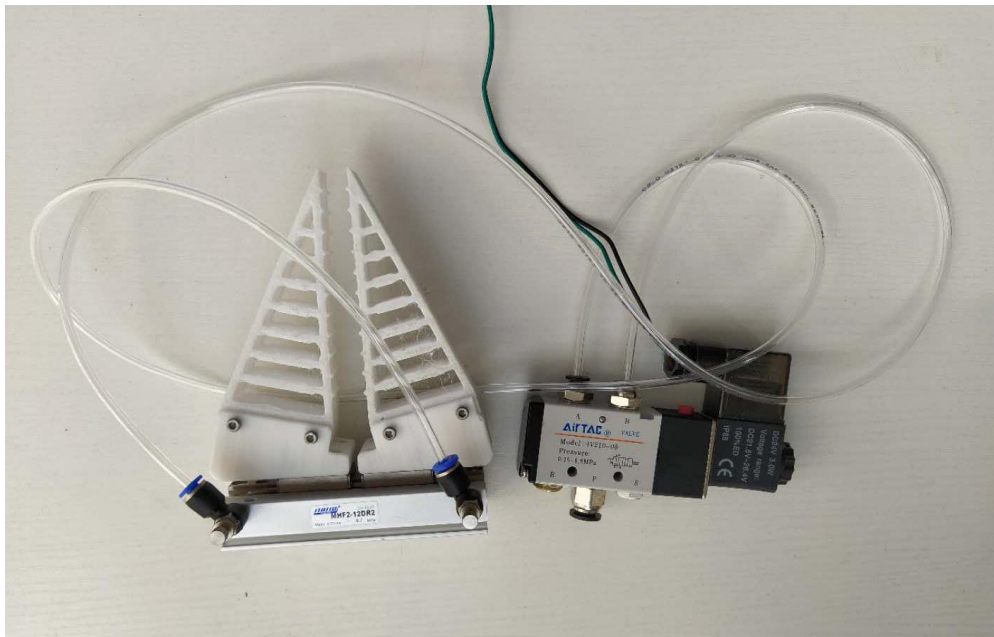
4.6 使用夹爪

- (1) 将气泵两极焊上电源线，在箭头朝上的出气口处插入 8mm 气管：

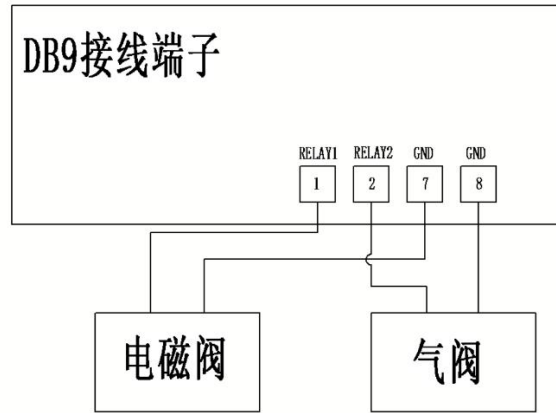


(2) 另一端与变径接头相连，接头另一端与 6mm 气管相连，再将 6mm 气管插入电磁阀 P 口，两根 2.5mm 气管插入 A、B 口，

(3) 将电磁阀 A 口的气管与夹爪右侧接口相连，B 口与夹爪左侧接口相连，如图所示，



(4) 按照下图将电磁阀、气泵、接线端子连接起来：



连接成功后请按照 5.5 节方式运行真机，并参考 IO 控制部分，输出 IO 1 和 2 设置为高电平时，夹爪闭合；输出 IO 1 和 2 设置为低电平时，夹爪张开。

⚠ 注意

请确保气管插入至气密螺丝最深处，否则系统漏气无法正常控制夹爪。

4.7 例程演示

上位机支持四个演示例程的快速启动，分别是正运动学演示，逆运动学演示，总体功能演示和画五角星演示。点击任意一个 demo 按键，机械臂便开始执行对应例程。



若想停止运动，点击 stop 按键。



5 MoveIt!编程接口

PROBOT 机械臂兼容 ROS MoveIt!编程接口，可以在 ROS 环境下使用 MoveIt!实现机器人的应用功能开发。

注意

(1) 以下函数接口均具备多态特性，详细函数说明请见 MoveIt!官方介绍：

https://docs.ros.org/api/moveit_ros_planning_interface/html/namespacemoveit_1_1planning_interface.html

(2) 完整例程请参见：

<https://github.com/ps-micro>，

如 https://github.com/ps-micro/PROBOT_G602/tree/master/probot_demo/src

5.1 设置运动状态

(1) void

`moveit::planning_interface::MoveGroupInterface::MoveGroupInterfaceImpl::setMaxAccelerationScalingFactor (double max_acceleration_scaling_factor)`

函数功能	设置运动加速度系数		
函数参数	类型	名称	描述
	double	max_acceleration_scaling_factor	加速度上限系数，范围： 0~1
返回值	值	描述	

	-	-
说明		

例程:

```
moveit::planning_interface::MoveGroupInterface arm("manipulator");
arm.setMaxAccelerationScalingFactor(0.2);
```

(2) void

```
moveit::planning_interface::MoveGroupInterface::MoveGroupInterfaceImpl::setMaxVelocityScalingFactor ( double max_velocity_scaling_factor )
```

函数功能	设置运动速度系数		
函数参数	类型	名称	描述
	double	max_velocity_scaling_factor	速度上限系数，范围：0~1
返回值	值	描述	
	-	-	
说明			

例程:

```
moveit::planning_interface::MoveGroupInterface arm("manipulator");
arm.setMaxVelocityScalingFactor(0.2);
```

(3) void

moveit::planning_interface::MoveGroupInterface::MoveGroupInterfaceImpl::setGoalJointTolerance(double tolerance)

函数功能	设置关节运动的允许误差		
函数参数	类型	名称	描述
	double	tolerance	允许误差，单位：Rad
返回值	值	描述	
	-	-	
说明			

例程：

```
moveit::planning_interface::MoveGroupInterface arm("manipulator");  
arm.setGoalJointTolerance(0.001);
```

(4) void

moveit::planning_interface::MoveGroupInterface::MoveGroupInterfaceImpl::setGoalOrientationTolerance(double tolerance)

函数功能	设置运动姿态的允许误差		
函数参数	类型	名称	描述

	double	tolerance	允许误差
返回值	值	描述	
	-	-	
说明			

例程：

```
moveit::planning_interface::MoveGroupInterface arm("manipulator");
arm.setGoalOrientationTolerance(0.001);
```

(5) void

moveit::planning_interface::MoveGroupInterface::MoveGroupInterfaceImpl::setGoalPositionTolerance (double tolerance)

函数功能	设置运动位置的允许误差		
函数参数	类型	名称	描述
	double	tolerance	允许误差
返回值	值	描述	
	-	-	
说明			

例程：

```

moveit::planning_interface::MoveGroupInterface arm("manipulator");
arm.setGoalPositionTolerance (0.001);

```

(6) void moveit::planning_interface::MoveGroupInterface::stop (void)

函数功能	停止机器人的所有运动		
函数参数	类型	名称	描述
	-	-	-
返回值	值	描述	
	-	-	
说明			

5.2 设置目标位姿

(1) bool moveit::planning_interface::MoveGroupInterface::setJointValueTarget (const std::vector< double > & group_variable_values)

函数功能	设置各关节目标位姿		
函数参数	类型	名称	描述
	std::vector< double > &	group_variable_values	各关节位置值，单位：Rad
返回值	值	描述	

	True	设置成功
	False	设置失败
说明		

例程：

```

moveit::planning_interface::MoveGroupInterface arm("manipulator");

double targetPose[6] = {0.391410, -0.676384, -0.376217, 0.0, 1.052834, 0.454125};
std::vector<double> joint_group_positions(6);
joint_group_positions[0] = targetPose[0];
joint_group_positions[1] = targetPose[1];
joint_group_positions[2] = targetPose[2];
joint_group_positions[3] = targetPose[3];
joint_group_positions[4] = targetPose[4];
joint_group_positions[5] = targetPose[5];

arm.setJointValueTarget(joint_group_positions);
arm.move();

```

(2) `bool moveit::planning_interface::MoveGroupInterface::setPoseTarget (const geometry_msgs::Pose & end_effector_pose, const std::string & end_effector_link = "")`

函数功能	设置机器人终端的目标位姿		
函数参数	类型	名称	描述

	geometry_msgs::Pose &	end_effector_pose	终端的目标姿态
	std::string &	end_effector_link	终端名称，可缺省
返回值	值	描述	
	True	设置成功	
	False	设置失败	
说明			

例程：

```

moveit::planning_interface::MoveGroupInterface arm("manipulator");

// 设置机器人终端的目标位置
geometry_msgs::Pose target_pose;
target_pose.orientation.x = 0.70692;
target_pose.orientation.y = 0.0;
target_pose.orientation.z = 0.0;
target_pose.orientation.w = 0.70729;

target_pose.position.x = 0.2593;
target_pose.position.y = 0.0636;
target_pose.position.z = 0.1787;

arm.setPoseTarget(target_pose);
arm.move();

```

(3) void moveit::planning_interface::MoveGroupInterface::setRandomTarget()

函数功能	设置随机目标位姿		
函数参数	类型	名称	描述
	-	-	-
返回值	值	描述	
	-	-	
说明			

例程：

```

moveit::planning_interface::MoveGroupInterface arm("manipulator");

// 随机产生一个目标位置
arm.setRandomTarget();

// 开始运动规划，并且让机械臂移动到目标位置
arm.move();

```

(4) bool moveit::planning_interface::MoveGroupInterface::setPositionTarget (double x, double y, double z, const std::string & end_effector_link = "")

函数功能	设置机器人终端的目标位置
------	--------------

	类型	名称	描述
函数参数	double	x	X 坐标值
	double	y	Y 坐标值
	double	z	Z 坐标值
	std::string &	end_effector_link	终端名称, 可缺省
返回值	值	描述	
	True	设置成功	
	False	设置失败	
说明			

(5) `bool moveit::planning_interface::MoveGroupInterface::setRPYTarget (double roll, double pitch, double yaw, const std::string & end_effector_link = "")`

函数功能	设置机器人终端的目标姿态		
	类型	名称	描述
函数参数	double	roll	X 轴旋转值
	double	pitch	Y 轴旋转值
	double	yaw	Z 轴旋转值
	std::string &	end_effector_link	终端名称, 可缺省

	值	描述
返回值	True	设置成功
	False	设置失败
说明		

5.3 笛卡尔空间运动

```
(1) double moveit::planning_interface::MoveGroupInterface::computeCartesianPath (
const std::vector< geometry_msgs::Pose > & waypoints,
double eef_step,
double jump_threshold,
moveit_msgs::RobotTrajectory & trajectory,
bool avoid_collisions = true,
moveit_msgs::MoveItErrorCodes * error_code = NULL )
```

函数功能	设置机器人笛卡尔路径		
函数参数	类型	名称	描述
	std::vector< geometry_msgs::Pose > &	waypoints	笛卡尔路径点
	double	eef_step	终端的步进值
	double	jump_threshold	跳跃阈值
	moveit_msgs::RobotTrajectory &	trajectory	规划得到的笛卡尔路径

	bool	avoid_collisions	是否允许碰撞检测
	moveit_msgs::MoveItErrorCodes *	error_code	返回的错误码
返回值	值	描述	
	double	可规划路径的覆盖率，范围：0 ~ 1	
说明			

示例：

```

moveit::planning_interface::MoveGroupInterface arm("manipulator");

//获取终端 link 的名称
std::string end_effector_link = arm.getEndEffectorLink();

// 获取当前位姿数据最为机械臂运动的起始位姿
geometry_msgs::Pose start_pose = arm.getCurrentPose(end_effector_link).pose;

std::vector<geometry_msgs::Pose> waypoints;

//将初始位姿加入路点列表
waypoints.push_back(start_pose);

start_pose.position.z -= 0.2;
waypoints.push_back(start_pose);

start_pose.position.x += 0.1;
waypoints.push_back(start_pose);

```

```

start_pose.position.y += 0.1;
waypoints.push_back(start_pose);

// 笛卡尔空间下的路径规划
moveit_msgs::RobotTrajectory trajectory;
const double jump_threshold = 0.0;
const double eef_step = 0.01;
double fraction = 0.0;
int maxtries = 100; //最大尝试规划次数
int attempts = 0;   //已经尝试规划次数

while(fraction < 1.0 && attempts < maxtries)
{
    fraction = arm.computeCartesianPath(waypoints, eef_step, jump_threshold, trajectory);
    attempts++;

    if(attempts % 10 == 0)
        ROS_INFO("Still trying after %d attempts...", attempts);
}

if(fraction == 1)
{
    ROS_INFO("Path computed successfully. Moving the arm.");

    // 生成机械臂的运动规划数据
    moveit::planning_interface::MoveGroupInterface::Plan plan;
    plan.trajectory_ = trajectory;

    // 执行运动

```

```
        arm.execute(plan);
    sleep(1);
}
else
{
    ROS_INFO("Path planning failed with only %0.6f success after %d attempts.", fraction,
maxtries);
}
```

6 一键固件升级

ROS 上位机集成 PROBOT SFU 固件升级工具，可在网络环境下，为 ROBCELL 控制器远程提供一键固件更新服务。

按照以下步骤，可通过上位机对控制器固件进行升级。**请注意以下步骤中均以 PROBOT G750 为例，其他型号机械臂操作一致。**

6.1 使用须知

注意

- (1) 请严格按照本说明进行固件升级，否则可能导致无法开机、控制器无法启动等问题。
- (2) 请勿对正在工作状态的控制器（如正在控制机械臂运动）执行固件升级操作。**升级完成后将自动重启控制器，极可能引发危险！**
- (3) 在固件升级过程中，请确保控制器的供电正常，并且保持与 PC 的网络连接（不掉电、不断网），否则可能导致升级过程故障而无法开机。
- (4) 升级成功后，控制器将自动重启，可能导致蜂鸣器一直响。无需担心！只需将控制器重新上电即可。

6.2 升级前准备

启动控制器，并确认 PC 与控制器的网络连接稳定。

- (1) 使用如下命令启动 ROS 上位机，末尾添加附加参数 `sfu:=enable`，使能固件升级功能。

```
$ roslaunch probot_bringup probot_g750_bringup.launch sfu:=enable
```

- (2) 点击上位机 **Panels** 选项，选择 **SFUTool**，打开固件升级工具面板。

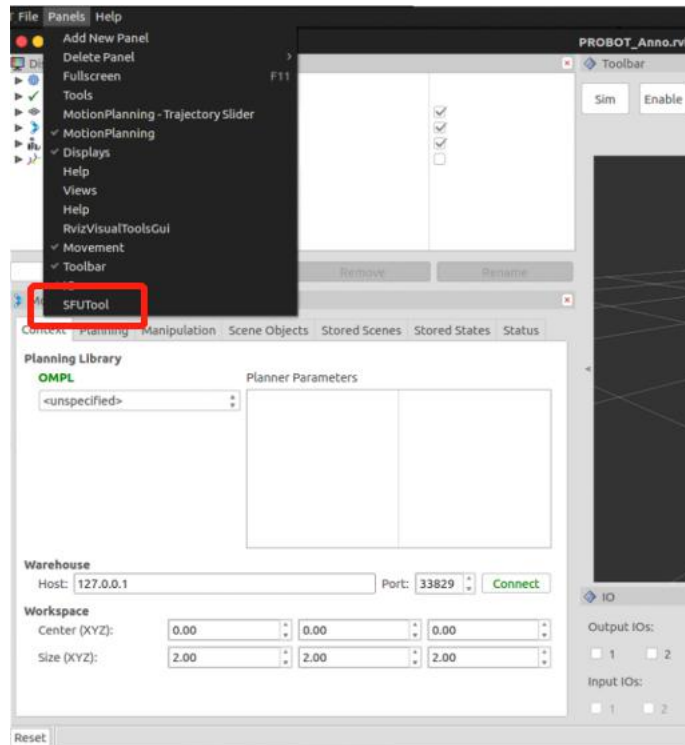


图 6-1 上位机中打开 SFUTool

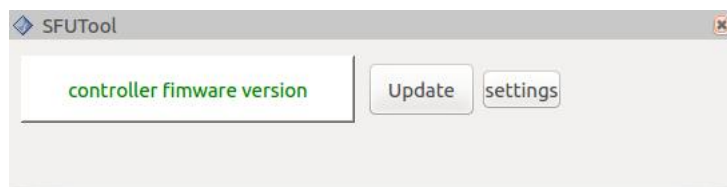


图 6-2 固件升级工具面板

controller firmware version 一栏将显示控制器固件版本。Update 为一键固件升级按钮。点击 settings 可打开配置面板。

6.3 一键升级

点击 SFUTool 面板的 Update 按键，系统将自动完成以下动作：

1. 获取控制器固件版本信息；
2. 根据以上得到的版本信息，查询对应的最新固件版本，以确认当前控制器固件是否需要更新。版本显示栏将显示控制器当前固件的版本；



图 6-3 查询控制器当前版本

3. 若需要更新，则弹出以下提示框。点击 Yes 确定进行升级；

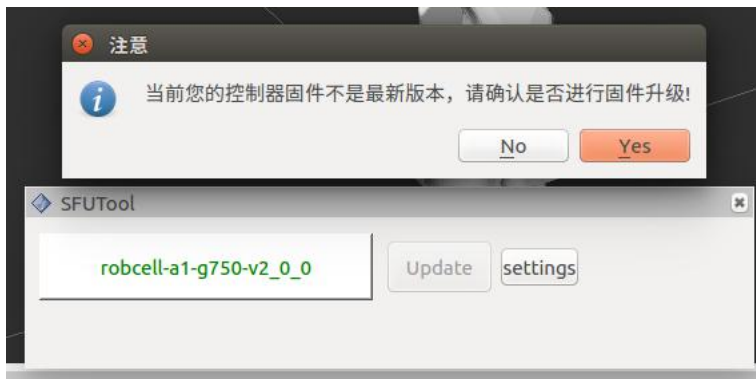


图 6-4 固件升级确认

4. 固件升级启动后，将下载官方最新的固件包，下载完成后自动完成固件更新操作；

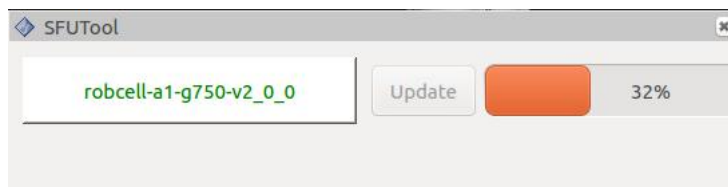


图 6-5 固件升级正在进行

5. 固件升级完成。若升级成功，版本显示栏将刷新固件版本号，并自动重启控制器。此时上位机和控制器的连接将断开，请重启上位机。



图 6-6 固件升级成功

7 Blockly 图形化编程

ROS 上位机集成 PROBOT Blockly 图形化编程工具，在仿真或真机模式下，都可通过拖拽代码块的方式编程控制机械臂，无需手动编写代码。

7.1 启用图形化编程

7.1.1 单机仿真

上位机 ROS 环境搭建完成后即可在 PC 上进行单机仿真，无需连接控制器和机器人。

使用以下命令，启动上位机并启用 Blockly 图形化编程功能：

```
$ roslaunch probot_bringup probot_g602_bringup.launch sim:=true blockly:=enable
```

注：默认不开启 Blockly 图形化编程功能，需指定启动参数 blockly 为 enable。

7.1.2 真机运行

注意

请检查硬件连接后给控制箱上电，确保 PROBOT G602 软硬件系统启动成功，并且机械臂已处于初始位姿后，才可使用该功能。

使用以下命令，启动上位机并启用 Blockly 图形化编程功能：

```
$ roslaunch probot_bringup probot_g602_bringup.launch blockly:=enable
```

注：默认不开启 Blockly 图形化编程功能，需指定启动参数 blockly 为 enable。

以上命令将默认同时启动 rviz 上位机界面，可指定启动参数 rviz 为 disable，启动时将不会打开 rviz 界面：

```
$ roslaunch probot_bringup probot_g602_bringup.launch rviz:=disable blockly:=enable
```

7.1.3 打开编程界面

以上两小节已在启动上位机控制的同时，开启了图形化编程工具，该程序运行于后台。可打开浏览器，网址栏输入 **127.0.0.1:1234**，按下回车即可访问 **Blockly** 编程界面。

7.2 编程界面简介

网页打开后，将看到如下界面：



图 7-1 Blockly 图形化编程界面

编程界面主要分为控制栏、编程视图区和 Python 视图区。

7.2.1 编程视图区

默认将进入编程视图区，该区左侧为 **Blockly** 代码块菜单。如点击 **机械臂运动** 菜单，将打开机械臂运动控制相关的 **Blockly** 代码块。鼠标左击单击选中所需要的代码块，直接拖拽到右侧空白区，即可完成代码块的图形化编程。



图 7-2 Blockly 图形化编程界面

7.2.2 Python 视图区

点击界面中的 *Python* 将进入 Python 视图区，可查看图形化编程自动生成的 Python 源码。

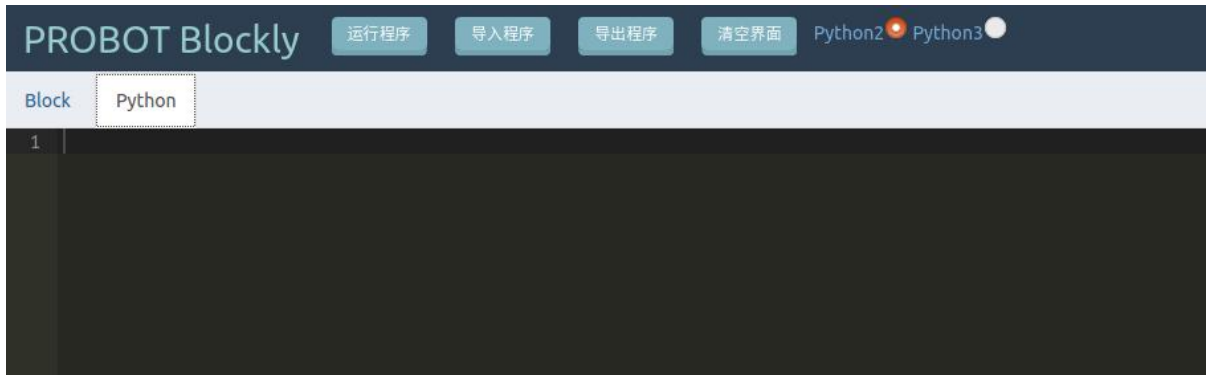


图 7-3 Python 视图区

7.2.3 控制栏

在编程视图区完成编程后，通过点击控制栏按键对程序进行处理。包括：

- (1) 运行程序：运行 Blockly 编程自动生成的程序，实现对机械臂和系统的控制；
- (2) 导入程序：导入 Blockly 程序到编程视图区；
- (3) 导出程序：将编程视图区完成的 Blockly 程序导出文件，可保存到本地。该文件可用于导入程序；
- (4) 清空界面：清空编程视图区的所有 Blockly 程序。

7.3 Blockly 代码块

7.3.1 机械臂设置



(1) 机械臂初始化。可设置真机或仿真控制环境，并选择是否使能机械臂控制。在真机环境下，必须使能机械臂控制，否则程序将无法控制机械臂运动；

(2) 设置运动速度系数（百分比）。用于调整关节运动、笛卡尔运动等所有机器人运动控制的速度；

(3) 设置运动加速度系数（百分比）。用于调整关节运动、笛卡尔运动等所有机器人运动控制的加速度；

(4) 设置关节运动的允许误差（角度）。用于调整关节运动的允许误差；

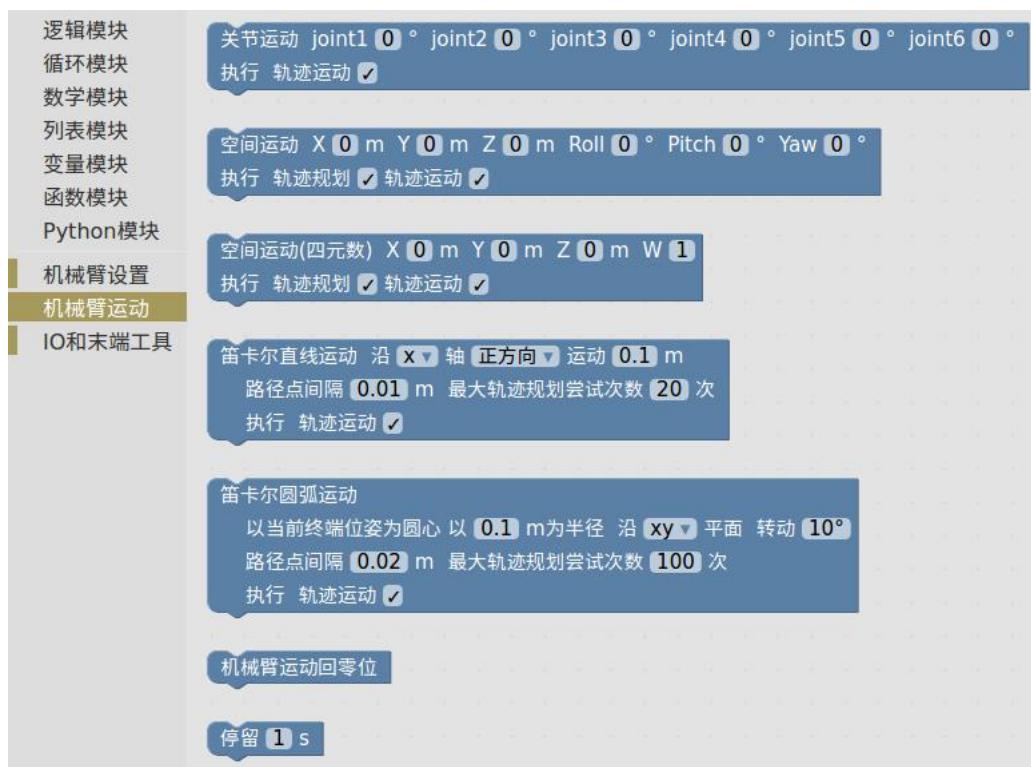
(5) 设置空间运动是否允许多次轨迹规划。勾选即为允许多次轨迹规划，即规划失败后，是否允许再次尝试；

(6) 设置空间运动的目标姿态允许误差（角度）；

(7) 设置空间运动的目标位置允许误差（米）；

(8) 机械臂运行环境卸载。可强制机械臂停止所有运动，并选择是否回到初始位姿，最后关闭机械臂控制功能。

7.3.2 机械臂运动



(1) 关节运动（度）。设置目标位置的各个关节角度值；

(2) 空间运动。设置目标位置的笛卡尔坐标值 xyz（米）和终端旋转角度（度），并选择是否执行轨迹规划和运动。如勾选 轨迹运动，则一定会执行轨迹规划；

(3) 空间运动（四元数）。设置目标位置的笛卡尔坐标值 xyz（米）和终端姿态（四元数），并选择是否执行轨迹规划和运动。如勾选 轨迹运动，则一定会执行轨迹规划；

(4) 笛卡尔直线运行。沿着某个坐标轴的正/负方向直线运动一段距离（米）。允许设置直线中规划路径点的间隔，以及轨迹规划的最大次数；

(5) 笛卡尔圆弧运动。以当前终端位姿为圆心，以设定半径（m）沿着某两个坐标轴形成的平面进行圆环运动。允许设置转动角度、路径点间隔，以及轨迹规划的最大次数；

(6) 机械臂运动回零位。控制机械臂回到各关节为 0 角度的位姿；

(7) 停留 n 秒。系统休眠一段时间，机械臂保持静止不动。

7.3.3 IO 和末端工具



(1) 设置控制器数字 IO OUTPUT。设置输出 IO 电平（高/低）；

(2) 如果 读取控制器数字 IO INPUT。读取输入 IO，如果电平值为高或低，则执行某个动作；

(3) 读取控制器数字 IO INPUT。返回值为高/低电平；

(4) 高/低电平值。

7.3.4 其他代码块

除机械臂控制之外，PROBOT Blockly 功能包还提供了程序逻辑、循环、数学、列表等代码块，使图形化编程功能更加强大。

7.4 Blockly 编程示例

Blockly 编程官方示例放置于 `packages/functions/probot_blockly/blockly_demo` 文件夹下，可在编程视图区导入和使用这些示例程序。

注意

使用 PROBOT Blockly 进行图形化编程时，请使用 机械臂初始化 作为程序的第一个代码块，用以初始化机械臂运行及控制环境。否则，将无法使用其他代码块控制机械臂（运行时终端将报错，并提示您使用 机械臂设置--机械臂初始化）。

在程序最末尾，建议使用 机械臂运行环境卸载 作为程序的最后一个代码块，但并不强制。

7.4.1 关节运动

jog_joint_demo:



图 7-4 Blockly 图形化编程示例-关节运动

7.4.2 空间运动

jog_pose_demo:



图 7-5 Blockly 图形化编程示例-笛卡尔运动

7.4.3 IO 控制

io_control_demo:



7.5 常见问题

7.5.1 启动脚本无可执行权限

报错如下：

```
ERROR: cannot launch node of type [ares_blockly/ares_blockly_backend.py]: can't locate
node [ares_blockly_backend.py] in package [ares_blockly]
ERROR: cannot launch node of type [ares_blockly/pywebserver.py]: can't locate node
[pywebserver.py] in package [ares_blockly]
```

该问题是由于启动脚本无可执行权限。执行以下命令为其添加可执行权限：

```
$ cd packages/functions/probot_blockly/scripts
$ sudo chmod +x . -R
```

7.5.2 启动脚本换行符格式错误

报错如下：

```
process[ares_blockly_backend-2]: started with pid [901]
/usr/bin/env: \u2018python3\r\u2019: No such file or directory
...
process[ares_blockly_webserver-3]: started with pid [904]
/usr/bin/env: \u2018python3\r\u2019: No such file or directory
```

该问题是由于启动脚本换行符格式错误。执行以下命令将 windows 换行符替换为 unix 格式：

```
$ cd packages/functions/probot_blockly/scripts
$ dos2unix ./*
```

在使用该命令前，需安装 dos2unix 工具。

7.5.3 缺少 python3 版本的依赖包

报错如下：

```
Traceback (most recent call last):
```

```
File "/blockly_ws/src/ares_blockly/scripts/ares_blockly_backend.py", line 35, in
<module>
...
import yaml
ImportError: No module named 'yaml'
```

该问题是由于 PC 缺少 Python3 版本的 `yaml`、`rospkg` 等依赖包。执行以下命令安装缺失的依赖包：

```
$ sudo apt-get -y install python3-pip ros-melodic-mavros
$ sudo pip3 install pyyaml numpy rospkg autobahn
```

8 常见故障诊断

8.1 无法开机

检查电源输入线缆是否已接好，电源插头是否插到有效市电网络。中国大陆电源电压 220V，中国台湾、日本、欧美等地区电源电压 110V。

电源开关是否处于开启状态（电源开关处于开启状态时，电源开关指示灯亮。）急停开关是否处于急停状态，如果是以上原因，可以自行诊断解决；否者，请及时和我们的维修部联系，请不要在不熟悉的情况下自己拆装机器。售后问题联系邮箱：support@robotanno.com

8.2 机械臂无法回到初始位姿

处理方法：如果在操作过程中，发现机械臂回不了初始位姿，请参考关节空间点动部分，使用关节空间点动功能将机械臂调整到初始位姿。然后再打开电源，重新设置初始位姿，调试运转几次，查看该问题是否排除。如果多次出现该问题，请及时与我们的售后部联系。请不要在不熟悉的情况下自己擅自拆动机器。

8.3 机械臂抓不紧物体

机械臂如果出现抓不紧物体等，可能因为机械臂抓取物体超载或者机械臂使用时间长引起的零件磨损等问题。如果多次出现该问题，请及时和我们的维修部联系，请不要在不熟悉的情况下自己拆机器。

注意

如果不是因为被抓物体超重引起的，请找修理人员处理，不要自己私自处理，否则出现其他问题我们概不负责。

9.1 参考资料

1. Github - PROBOT: https://github.com/ps-micro/PROBOT_G602
2. MoveIt! Tutorials: http://docs.ros.org/melodic/api/moveit_tutorials/html/index.html
3. 古月学院: <https://class.guyuehome.com/>
4. 古月居: <https://www.guyuehome.com/>
5. 《ROS 机器人开发实践》，胡春旭编著，机械工业出版社
6. 《ROBOT SFU 固件升级工具用户手册》，精锋微控

9.2 联系方式

邮箱: support@robotanno.com

网站: www.robotanno.com

www.guyuehome.com

