DUCO Core 用户手册

4.3

中科新松有限公司

2025 年 09 月 25 日

Contents

CHAPTER 1

安全

1.1 概述

本章介绍了在安装、使用和维护机器人及其部件时需要注意的安全事项与风险评估等重要信息，在机器人第一次通电前，用户必须阅读并理解这些信息。

在执行任何操作前，确保已经阅读随设备提供的所有操作说明，特别是危险、警告和注意等可能危及人身安全和设备安全的说明，以最大限度减少事故发生几率。当本文内容与随设备发货的文档有差异时，请以随设备发货的文档为准。

负责安装和维护设备的人员必须是经过培训并且已经掌握了正确操作方法和所有安全预防措施的专业人员。只有经过培训的合格人员才能执行设备安装和维护操作。

1.2 有效范围与责任说明

该信息不包含如何设计、安装和操作完整的机器人系统，也不包含可以影响整个系统的安全的所有外围设备。为了保护人员安全，必须设计完善的系统，并且必须按照机器人安装所在国家/地区的标准和法规中规定的安全要求进行安装。

机器人集成商应负责确保机器人系统遵守所在国家或地区的适用安全法律和法规，并且用于保护机器人系统操作者的必要安全设备设计合理且安装正确。

具体包括但不限于以下几点：

- 确保机器人系统符合所有基本要求；
- 对完整的系统执行风险评估；
- 确保整个系统的设计安装准确无误；
- 在软件中进行合适的安全设置，并确保不会被用户修改；
- 制定详细的操作说明；
- 出具符合性声明；

- 将所有信息收集到技术文件中;
- 在安装的机器人系统上标贴集成商的标志和联系信息。

中科新松有限公司致力于提供可靠的安全信息，除非中科新松有限公司在提供可靠安全信息方面存在故意或重大过失，否则中科新松有限公司对此不承担责任。需要明确的是，即使一切操作都按照安全操作进行，也无法确保机器人系统不会造成用户的人身和财产损失。

因以下原因造成的用户损失，中科新松有限公司将不对此承担责任:

- **不可抗力事件（例如，自然灾害、火灾、战争等）;**
- **机器人系统自然损坏或磨损;**
- **现场运行环境（例如，电压、温度、湿度等）或外部因素（例如，外部干扰等）不能满足已提示的正常运行的环境要求;**
- **机器人系统未正确安装（包括搬迁后未重新正确安装）;**
- **由于用户或第三方的故意或疏忽、使用不当（包括用户未按本用户手册和/或中科新松有限公司其他要求使用）或蓄意破坏行为。**

除非另有约定，因使用机器人系统所造成的间接、特殊、偶发损失，包括但不限于收入损失、实际或预期收益损失、业务损失、机会损失、商誉损失、名誉损失、数据的丢失、损坏或泄露等，中科新松有限公司均不对此承担责任。

1.3 风险评估

风险评估是集成商必须完成的最重要任务之一。机器人本身是一个部分完成的机械，而机器人安装的安全性取决于该机器人是如何集成的（例如：工具、障碍物及其他机械）。

建议集成商按照标准 ISO12100（GB 15706）和 ISO10218-2（GB 11291.2）中的规定执行风险评估。另外，可选择技术规范 ISO/TS 15066 （GB/ T 36008）作为附加指引。集成商执行风险评估时应考虑机器人整个应用寿命期间的所有工作程序，包括但不限于:

- 在开发机器人安装时示教机器人;
- 故障诊断和维护;
- 机器人安装的正常操作。

风险评估必须在机器人手臂第一次通电之前进行。由集成商执行的风险评估的一部分就是识别正确的安全配置设置，以及确定是否需要额外的紧急停止按钮和其他保护措施。

下列明确了集成商必须要考虑的重大危险。请注意，特定机器人设备可能还存在其他重大危险。

- 手指被夹在机器人关节 4 和关节 5 之间。
- 工具或工具连接器上的锐边和尖点刺伤皮肤。
- 机器人轨迹附近障碍物上的锐边和尖点刺伤皮肤。
- 因机器人有效负载与坚固表面之间的冲击而导致扭伤或骨折。
- 因用于固定机器人手臂或工具的螺栓松动而导致的后果。
- 物品从工具上掉落，例如因夹持不到位或断电。
- 因不同机器上紧急停机按钮不同而出现的操作错误。

如果将机器人安装在无法使用其内部安全功能（例如使用危险工具）充分消除风险的非协作机器人应用中，则系统集成商必须根据风险评估安装其他保护装置（例如，使用能够在安装和编程期间对集成商提供保护的保护装置）。因未安装保护装置所导致的损失，中科新松有限公司将不对此承担责任。

1.4 安全操作

1.4.1 紧急停机

紧急停机是优先于所有其他机器人控制操作的状态，将会导致所有受控的危险停止，从机器人驱动器消除电机供电，在重置前一直保持有效，并且只能通过手动操作来复位。

紧急停止状态意味着动力系统断开，机器人无法运动。用户必须执行还原步骤，即复位紧急停止按钮并按下示教器上的“开机”按钮，以恢复正常操作。紧急停机不可用作风险降低措施，但是可作为次级保护设备。

紧急停机不得用于正常的程序停止，因为这可能会给机器人带来额外的不必要磨损。

1.5 安全相关的功能

1.5.1 安全功能简介

GCR 系列机器人搭载多种内置安全功能以及紧急电气接口的安全 I/O、数字和模拟控制信号，用于连接其他机器人及附加的保护装置。

小心： 安全功能和接口的使用和配置必须遵循每个机器人应用程序的风险评估程序。

如果机器人发现安全系统中存在的故障或违例（例如紧急停止电路中的一条线被切断或发生安全极限违例），将启动 0 类停机

停止时间应考虑作为应用风险评估的一部分。

警告： 使用的安全配置参数与风险评估所确定的不同可导致无法合理消除的危险或无法充分减少的风险。

确保工具和夹持器连接正确，以避免在电源中断情况下发生危险。

末端执行器不受 GCR 安全系统保护。末端执行器和/或连接电缆的功能不受监控

1.5.2 停机类别

根据具体情况，机器人可以启动三种根据 IEC60204-1 定义的停机类别。这些类别在下列描述中定义。

0 (SS0)

立即切断机器人动力电

1 (SS1)

立即将各个关节以最快的加速度降速为 0，关节静止后弹起抱闸，切断机器人电源

2 (SS2)

在保持轨迹的同时将机器人减速至静止，静止后各个关节保持使能状态，抱闸无动作

停机类别之间的切换:

当执行 1 类停机时，会同时触发计时器。如果到达 500ms 之后，机器人的速度仍然超过设定的安全速度，会转而执行 0 类停机。

1.5.3 安全功能

下表列出的新松多可协作机器人安全功能位于机器人中，其目的是控制机器人系统，即机器人及连接的工具/末端执行器。机器人安全功能用于减少由风险评估确定的机器人系统风险。

Emergency Stop (ES)

执行 SS1。

Protective Stop

执行 SS2。

Safe Operating Stop (SOS)

ss2 执行完成后会触发 SOS 监控，监控当前机器人位置偏移，如果违例触发 SS0。

Joint Safe Limited Position (SLP)

根据关节位置门限值设置，当关节位置达到门限值时，触发 SS2。如果触发关节位置限制，则直接触发 SS0。

Joint Safe Limited Speed (SLS)

根据关节速度门限值设置，当关节速度达到门限值时，触发 SS2。如果触发关节速度限制，则直接触发 SS0。

TCP Position Limit

可以设置安全区域限制机器人的作业区域，根据门限值设置，当到达门限值时，触发 SS2。如果离开安全空间后，安全控制器触发 SS0。最多允许设置 6 个安全区域、3 个 TCP 坐标系。

Tcp Speed Limit

根据 TCP 速度门限值设置，当 TCP 速度达到门限值时，触发 SS2。如果触发 Tcp 速度限制，安全控制器直接触发 SS0。

Elbow Position Limit

根据门限值设置，当达到门限值时，触发 SS2。如果触发 elbow 位置限制，安全控制器直接触发 SS0。

Elbow Speed Limit

根据肘部速度门限值设置，当肘部速度达到门限值时，触发 SS2。如果触发肘部速度限制，安全控制器直接触发 SS0。

Joint Torque Limit

根据关节力矩门限值设置，当关节力矩达到门限值时，触发 SS2。如果触发关节力矩限制，安全控制器直接触发 SS0。

Tcp Force Limit

根据 Tcp 力门限值设置，当 Tcp 力达到门限值时，触发 SS2。如果触发 Tcp 力限制，安全控制器直接触发 SS0。

Elbow Force Limit

根据肘部力门限值设置，当肘部力达到门限值时，触发 SS2。如果触发肘部力限制，安全控制器直接触发 SS0。

Power Limit

根据功率门限值设置，当功率达到门限值时，触发 SS2。如果触发功率限制，安全控制器直接触发 SS0。

Mode Switch Input

可以选择是否启动模式切换硬件输入，禁用该硬件输入时，可以通过 UI 切换，但不可以两者同时有效。模式切换时触发 SS2，如果当前正在运行脚本，脚本将进入暂停状态，后续可以继续运行。

Enable Device Input

可以选择是否启用该硬件输入。该输入只在手动模式下有效，自动模式下无效。违例触发 SS2。

Protective Stop Input

该输入各种模式下均有效。该输入会触发 SS2。如果未激活防护 reset 输入，输入信号消失后，SS2 自动复位，否则需要触发防护 reset 输入才可以复位。

Protective Stop Reset Input

可以选择是否激活该输入信号。如果激活了防护 reset 输入，且触发防护停止输入后触发 SS2，当防护停止输入信号消失后，需要触发防护 reset 输入信号才会复位 SS2。该输入信号上升沿有效，并且高电平需要保持 500ms。

Automatic Protective Stop Input

可以选择是否启用该硬件输入。该输入仅在自动模式下有效，该输入在自动模式下会触发 SS2。如果未激活防护 reset 输入且未激活自动模式防护 reset 输入，输入信号消失后，SS2 自动复位，否则需要触发防护 reset 输入或自动模式防护 reset 输入才可以复位。

Automatic Protective Stop Reset Input

可以选择是否启用该硬件输入。和防护 reset 输入信号相似，但仅针对自动模式下自动防护停止输入触发的 SS2 有效。

System Emergency Stop Output

该输出各种模式下均有效。当系统急停触发时会输出该信号。

Protective Stop Output

可以选择是否启用该硬件输出。当触发防护停止输入并触发 SS2 时会输出该信号。

Automatic Protective Stop Output

可以选择是否启用该硬件输出。仅自动模式下触发防护停止或自动模式防护停止输入并触发 SS2 时会输出该信号。

Reduce Mode

可以选择是否启用该硬件输入。该输入会触发进入缩减模式，并使安全参数采用缩减模式相关参数。

Reduce Mode Output

可以选择是否启用该硬件输出。当触发缩减模式输入并进入缩减模式时会输出该信号。

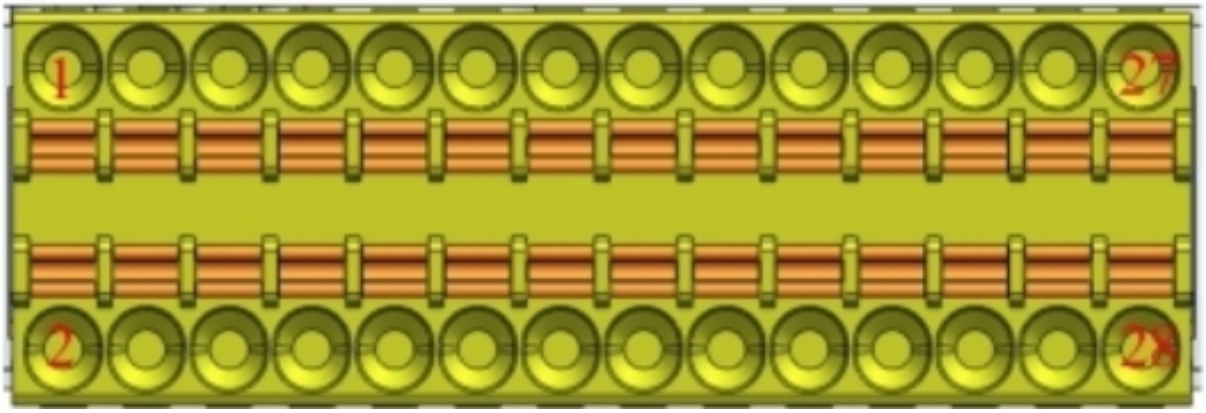
Recovery Mode

当触发关节位置限制、Tcp 位置限制或肘部位置限制并触发 SS0 后，在复位 SS0 并重新对机器人进行上电后，此时由于机器人关节、Tcp 或肘部仍然处于触发位置限制的状态，机器人系统会进入恢复模式。恢复模式下，机器人仅可进行关节 Jog，且关节速度不超过 30deg/s，末端速度不超过 250mm/s。在机器人关节、Tcp 及肘部都回到位置限制门限范围内后，会自动退从恢复模式退出并回到正常或缩减模式。

1.5.4 安全 IO 接口

安全 IO 接口是控制柜提供的外部急停及安全输入输出接口，包括 1 路急停信号输入（无源信号）、1 路急停反馈输出（有源信号）、1 路保护性停止输入（无源信号）、2 路可配置安全输入（无源信号）、2 路可配置安全输出（有源信号）。

可配置安全输入可配置为：防护 reset 输入、自动模式防护停止输入、自动模式防护 reset 输入、reduce 模式输入。可配置安全输出可配置为：防护停止输出、自动模式防护停止输出、reduce 模式输出。其接口定义如下表：



编号	信号定义	编号	信号定义
1	EI1+ (急停信号输入 1+)	2	EI1- (急停信号输入 1-)
3	EI2+ (急停信号输入 2+)	4	EI2- (急停信号输入 2-)
5	PSI1+ (保护性停止输入 1+)	6	PSI1- (保护性停止输入 1-)
7	PSI2+ (保护性停止输入 2+)	8	PSI2- (保护性停止输入 2-)
9	CSI1_1+ [可配置安全输入 1(1+)]	10	CSI1_1- [可配置安全输入 1(1-)]
11	CSI1_2+ [可配置安全输入 1(2+)]	12	CSI1_2- [可配置安全输入 1(2-)]
13	CSI2_1+ [可配置安全输入 2(1+)]	14	CSI2_1- [可配置安全输入 2(1-)]
15	CSI2_2+ [可配置安全输入 2(2+)]	16	CSI2_2- [可配置安全输入 2(2-)]
17	EO1+ (急停反馈输出 1+)	18	EO1-(急停反馈输出 1-)
19	EO2+ (急停反馈输出 2+)	20	EO2-(急停反馈输出 2-)
21	CSO1_1+[可配置安全输出 1 (1+)]	22	CSO1_1-[可配置安全输出 1 (1-)]
23	CSO1_2+[可配置安全输出 1 (2+)]	24	CSO1_2-[可配置安全输出 1 (2-)]
25	CSO2_1+[可配置安全输出 2 (1+)]	26	CSO2_1-[可配置安全输出 2 (1-)]
27	CSO2_2+[可配置安全输出 2 (2+)]	28	CSO2_2-[可配置安全输出 2 (2-)]

1.6 夹伤与碰撞风险

机器人实际运行过程中仍然存在碰撞检测功能盲区，用户务必需要注意在特殊工况下碰撞检测失效或夹伤风险。典型的三类工况如下所述。

工况一：机器人末端位置距离机器人基座中心超过机器人最大工作空间 80% 范围以外（机器人 3 关节位置约小于 20°）时，此时若机器人按照图 2.6.1 与图 2.6.2 所示红色箭头方向移动，机器人对运动方向上外力敏感度较低，较易发生不受控的碰撞危险；当机器人按照图 2.6.1 与图 2.6.2 所示绿色箭头方向产生移动，此时若机器人与外界环境发生碰撞，则对碰撞产生的外力较为灵敏。

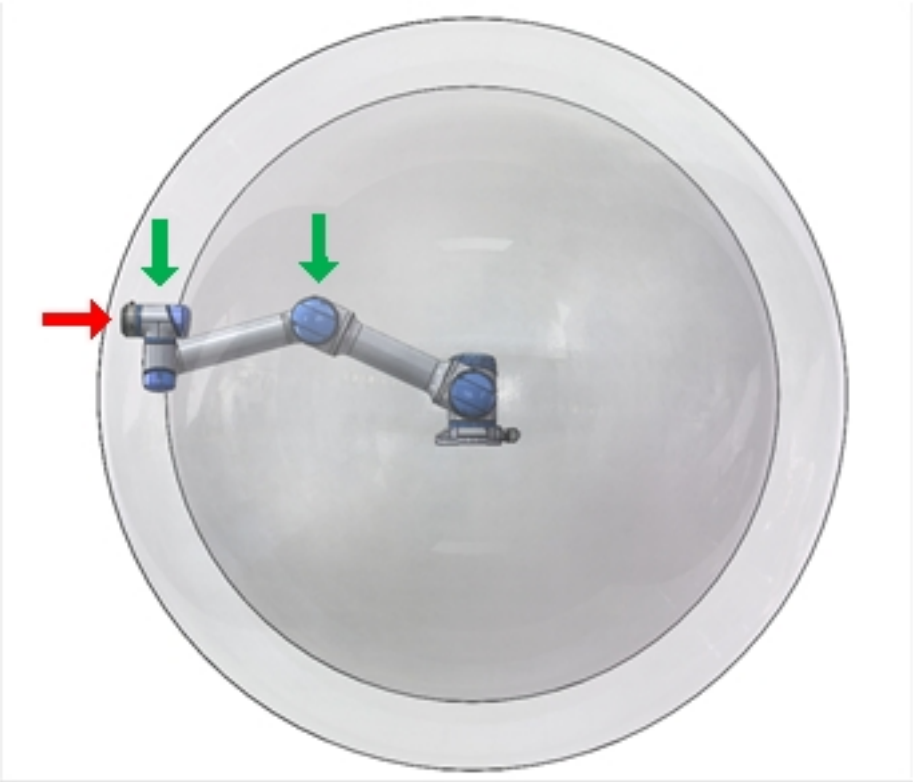


图 2.6.1 工况一机器人正视图

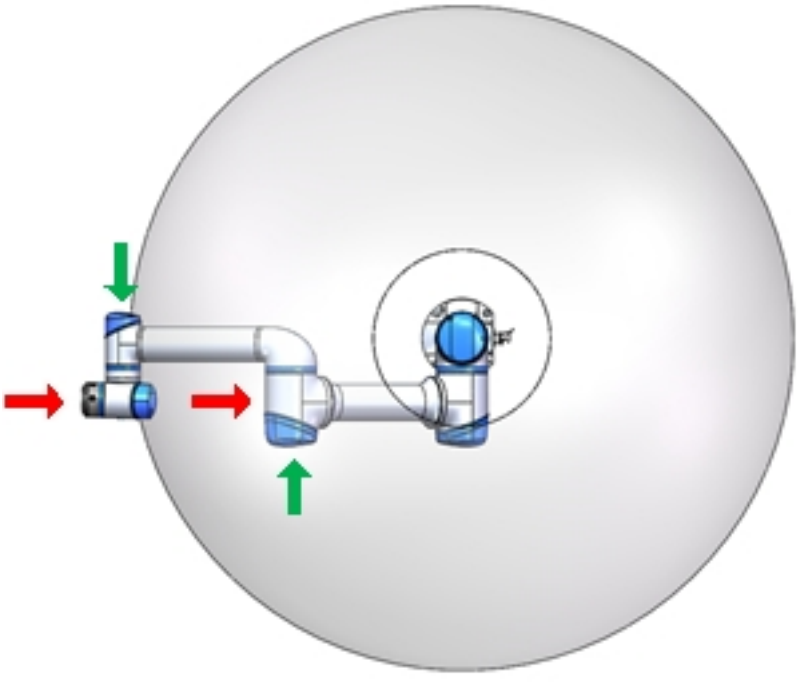


图 2.6.2 工况一机器人俯视图

工况二：以机器人基座 Z 方向为中心，半径约为 150-300mm（具体以机器人型号参数为准）的范围内，若接触点在该范围内且接触力方向与关节三连杆和关节四连杆所构成平面垂直，则碰撞检测功能较难检测机器人与外界发生的碰撞。如图 2.6.3 与图 2.6.4 中红色箭头所

示；此时若机器人与外界接触力方向与基座标 z 方向较一致，则机器人对碰撞产生的外力较为灵敏，如图 2.6.3 中绿色箭头所示。

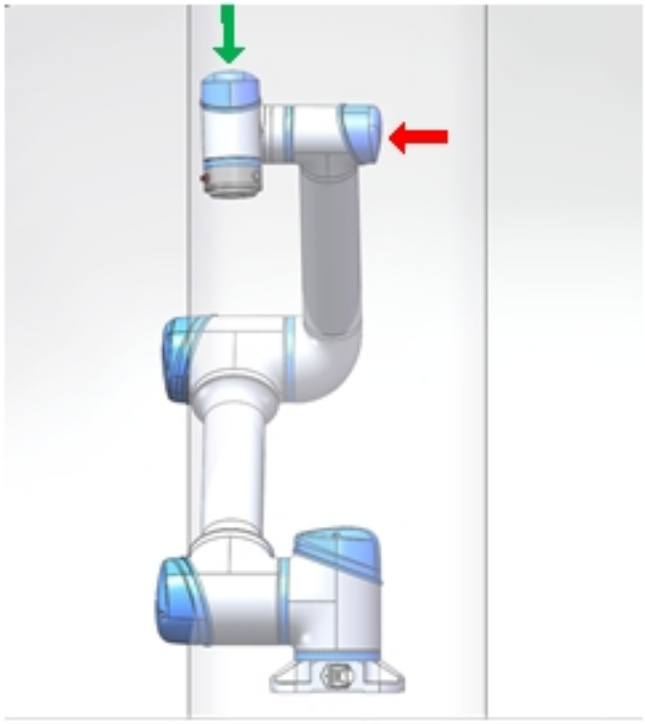


图 2.6.3 工况二正视图

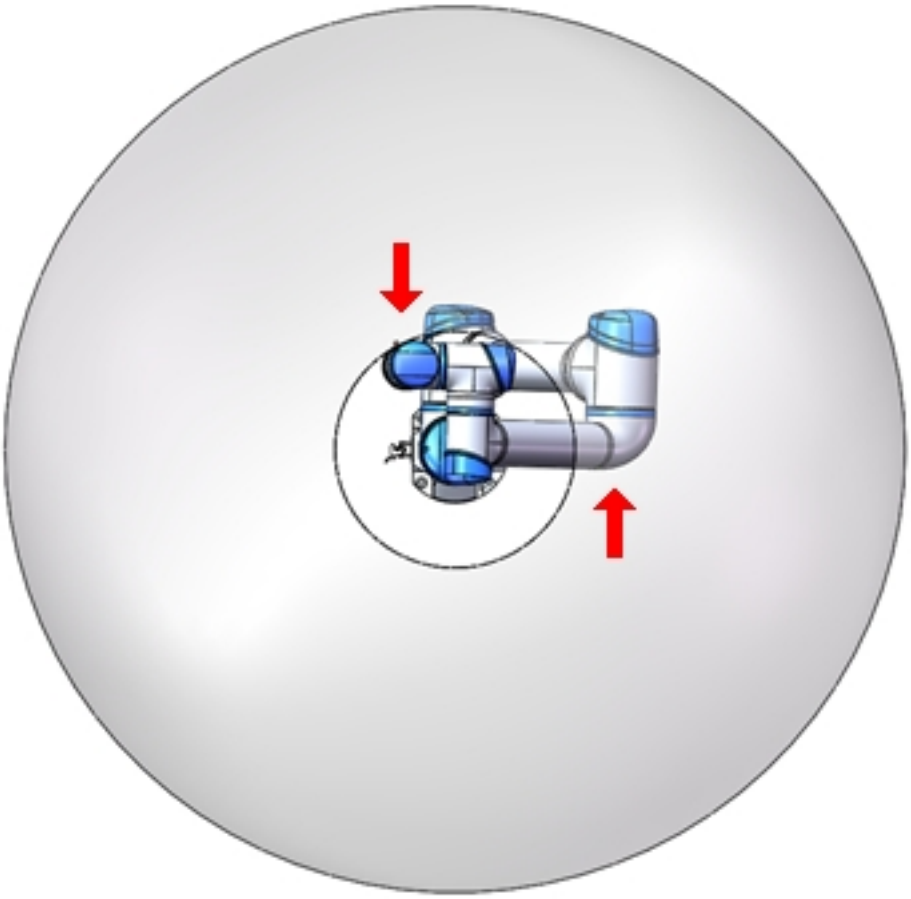


图 2.6.4 工况二俯视图

工况三：无论机器人处于何种位姿与何种运动状态，当机器人与外界发生碰撞时，若碰撞点位于以机器人基坐原点为中心，半径约为 350mm 的范围内，则机器人对该类碰撞较难检测，较易发生夹伤危险，如图 2.6.5 与图 2.6.6 中红色箭头所示；当碰撞点位于该范围以外，并且不满足工况一与工况二中所描述的碰撞检测盲区条件时，机器人较易对与外界产生的碰撞进行检测，如图 2.6.5 与图 2.6.6 中绿色箭头所示。

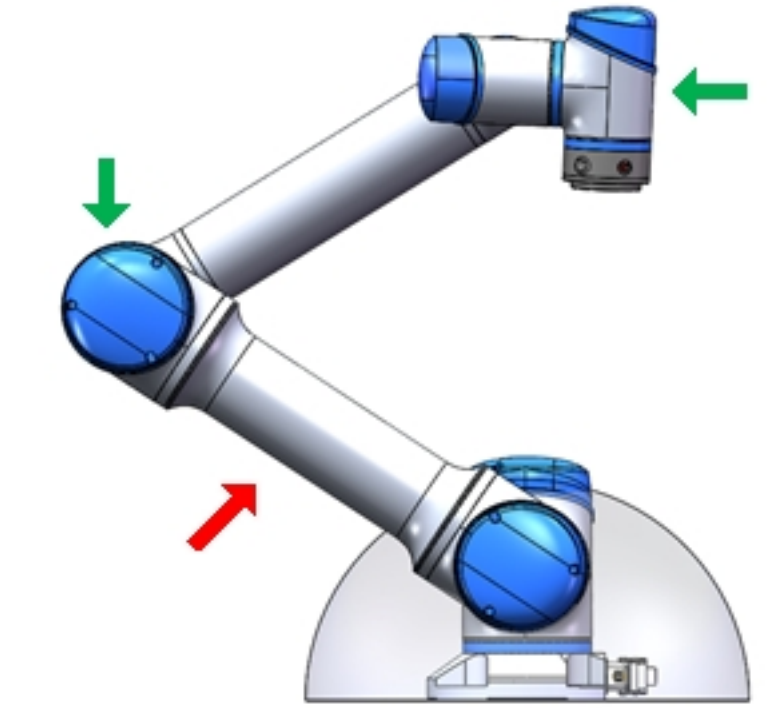


图 2.6.5 工况三侧视图

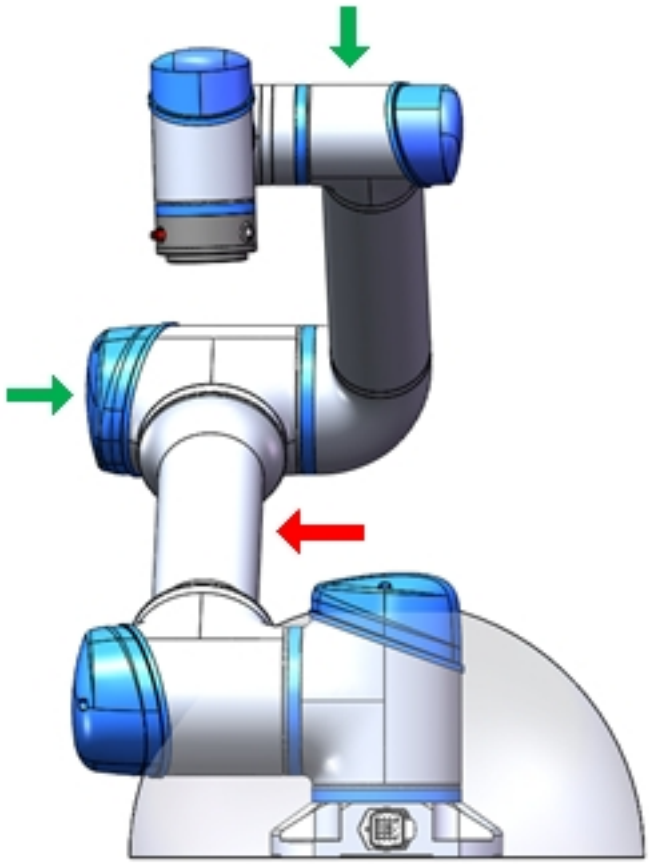
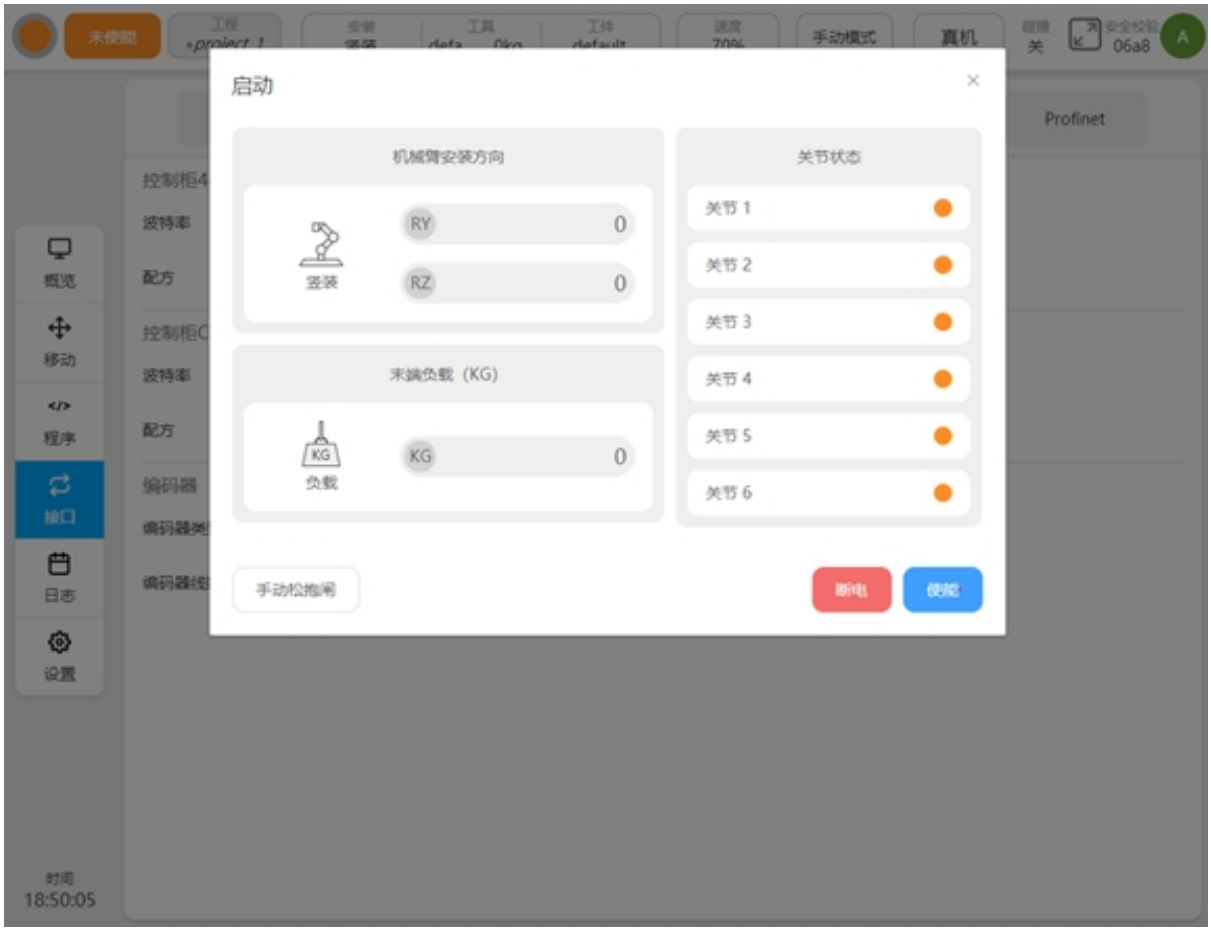


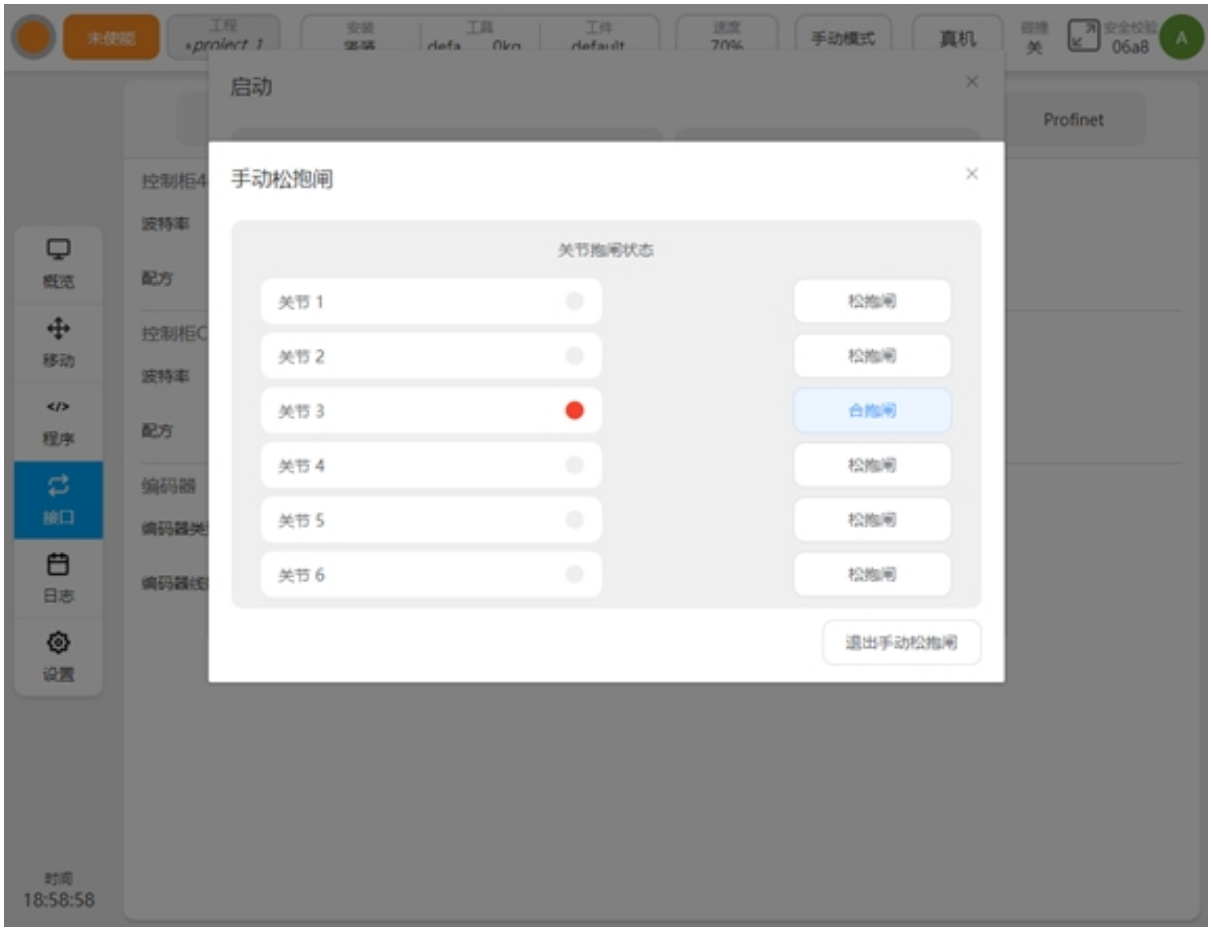
图 2.6.6 工况三正视图

针对以上所有描述的工况，若机器人在对外部碰撞检测不灵敏的方向上移动，考虑到机器人与外界协作力的限制需求，应尽可能减小此时的运行速度。

当夹伤意外不幸发生后，可以使用手动松抱闸功能，减轻意外所带来的损失。



在机器人得电但未使能的状态下，界面左下角可以启动“手动松抱闸功能”。点击“手动松抱闸”按钮后，界面切换如下图所示：



点击“松抱闸”按钮，可将对应关节的抱闸松开，此时允许对该关节进行无动力驱动。界面左侧的红色指示灯用于提示抱闸状态。点击“合抱闸”则将对应关节的抱闸重新闭合。

1.7 机器人奇异点失速风险

机器人在奇异点附近进行运动规划（直线、圆弧等，不包括关节运动）时会自动降速，示教时应避开奇异点或以关节运动通过奇异点。针对 GCR 系列构型，存在肩部奇异点、肘部奇异点以及腕部奇异点。

1.7.1 肩部奇异

当腕关节中心（五关节 J5 与六关节 J6 轴线的交点）处于一关节轴线 J1 上时，此时造成肩部奇异，导致一关节 J1 会产生运动位置突变。当腕关节中心位于很接近 J1 的位置时，也会受到奇异的影响，此时移动末端可能导致 1 关节不受控的高速运动。参考下图为临近肩部奇异位姿。

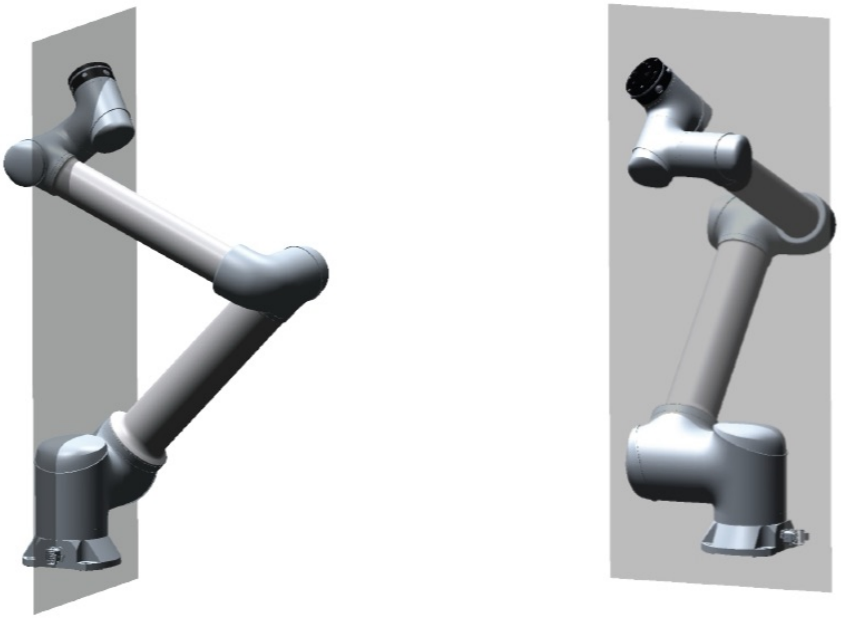


图 2.7.1 肩部奇异参考位姿

1.7.2 肘部奇异

当机器人二关节 J2、三关节 J3、四关节 J4 三轴轴线共面时，此时沿着共面所在平面方向进行末端运动会使三关节 J3 产生位置突变。简单的，当三关节 J3 临近 0° 或 ±180° 时，机器人临近奇异位姿，此时移动末端可能造成二关节 J2，三关节 J3，四关节 J4 产生不受控的高速运动。参考下图临近肘部奇异位姿：

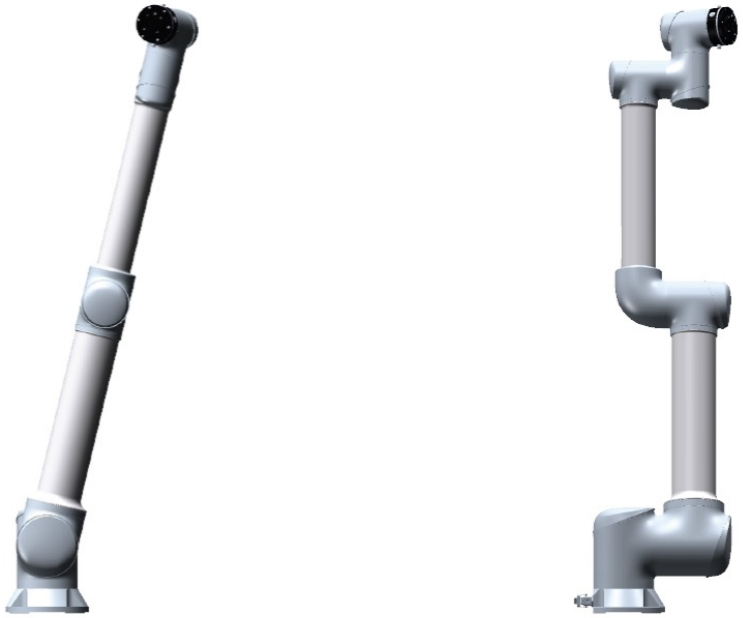


图 2.7.2 肘部奇异参考位姿

1.7.3 腕部奇异

当机器人四关节 J4 与六关节 J6 轴线平行时，会发生碗部奇异。此时沿着四关节 J4 与六关节 J6 轴线所在平面方向运动机器人末端时会使四关节 J4 与六关节 J6 产生位置突变。简单地，当五关节 J5 接近 0° 或 ±180° 时，此时四关节 J4 与六关节 J6 会产生不受控的高速运动。参考下图:

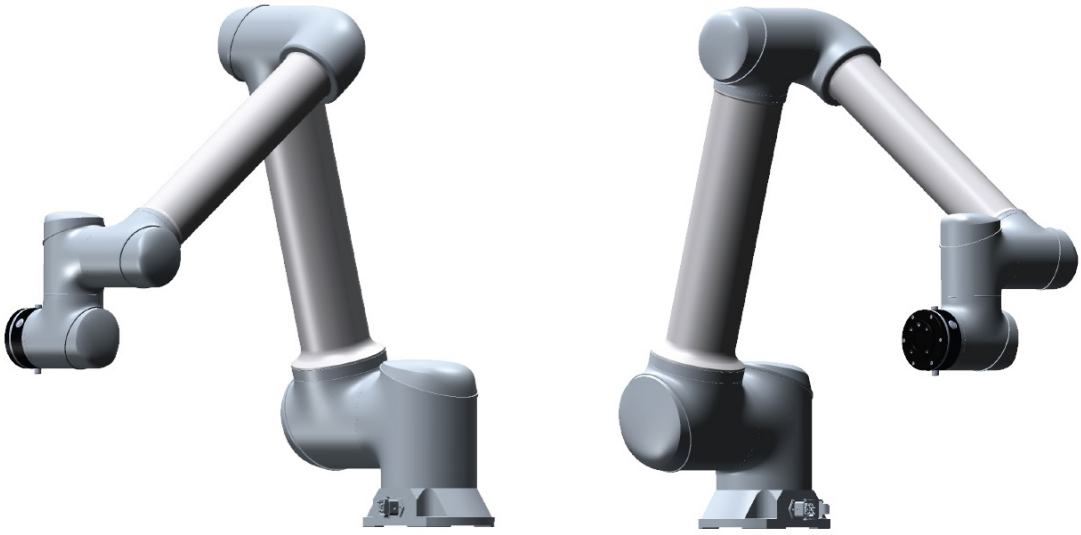


图 2.7.3 腕部奇异参考位姿

机器人运行到达或接近上述奇异点时，基于笛卡尔坐标的规划运动无法正确的逆解为各轴的关节运动，将无法正确的进行运动规划，可采用关节点动运动或 move j 运动指令。

警告： 请避免在奇异点附近使用直线、圆弧、沿 X、Y、Z、RX、RY、RZ 方向移动末端等指令，机器人存在失速风险。

对于存在奇异风险的轨迹，必须经过充分安全评估后再运行。

CHAPTER 2

用户手册

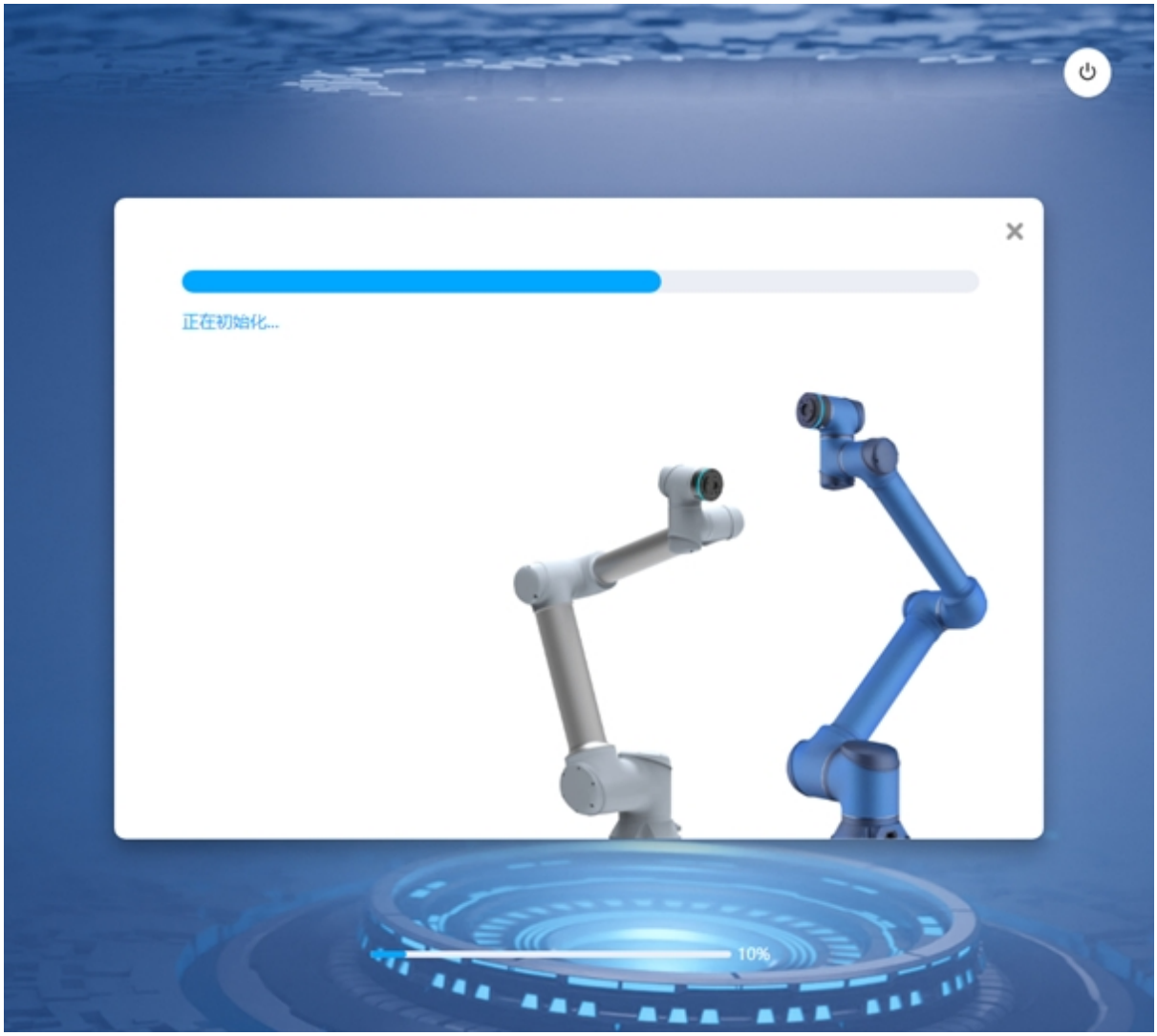
2.1 快速启动

2.1.1 首次开机

首次开机时，需要对机器人系统进行初始化并同步所连接机械臂的参数，具体操作步骤如下：

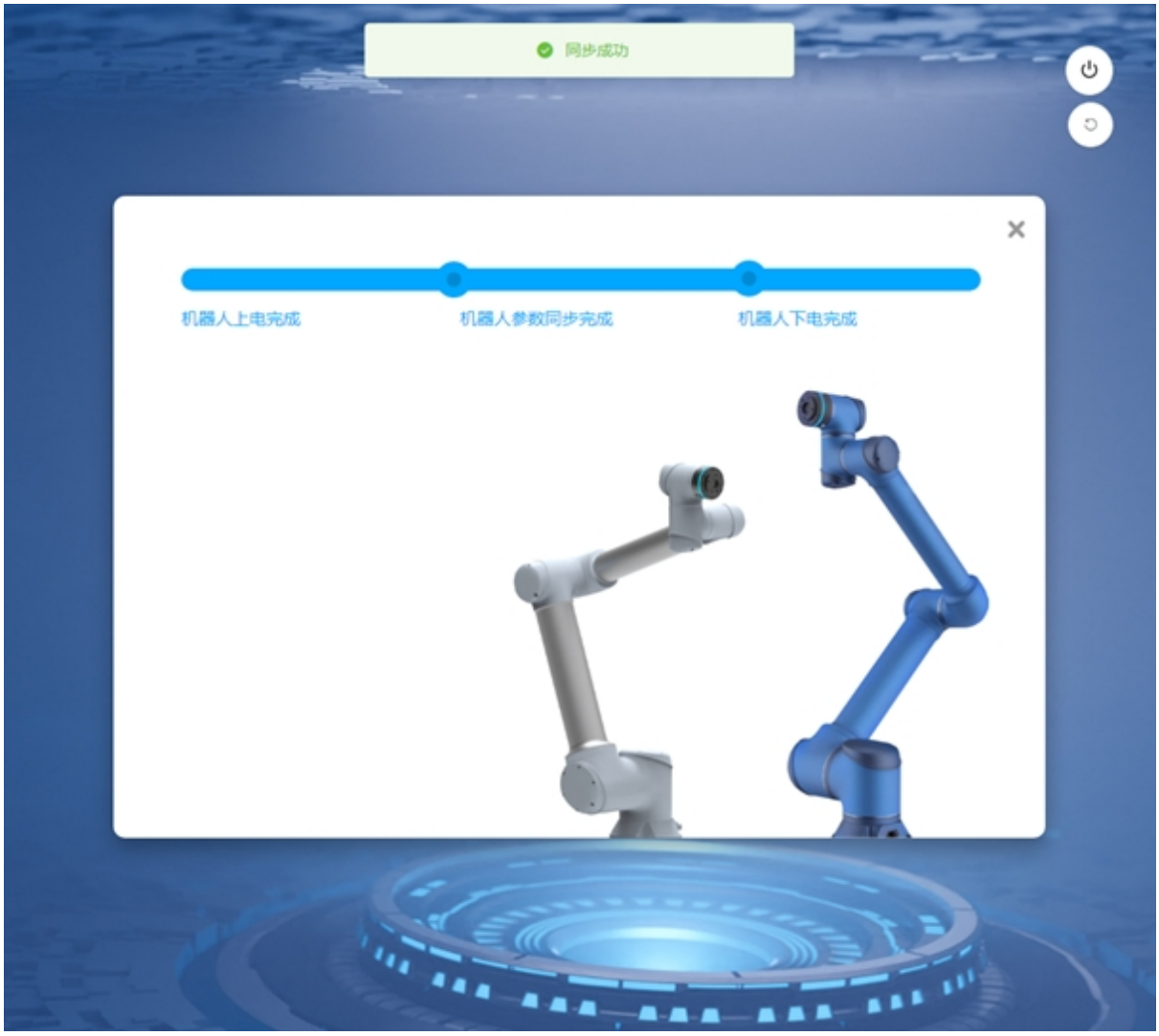
1. 开机系统默认显示“欢迎使用 DUCO 协作机器人”，进入“开始”页面。

2．点击“开始”按钮，进行机器人系统初始化。显示“正在初始化…”进度条，初始化完成显示“初始化完成”。



3．机器人初始化完成，进入机器人参数同步页面，点击“参数同步”。机器人会进行上电、参数同步、下电操作。


机器人参数同步
点击“参数同步”开始进行，同步过程机器人会自动上下电，请勿切断电源。
Click "SyncParam" to start robot parameters sync process.
Robot will automatically power on and off. Please do not cut off the power.
参数同步



4．提示同步成功后，进入安全参数设置页面。

5．确认安全参数正常后，点击“设置安全参数”，设置安全参数完成后，机器人初始化完成，根据提示重启控制柜。


设置安全参数完成

2.1.2 开机登录

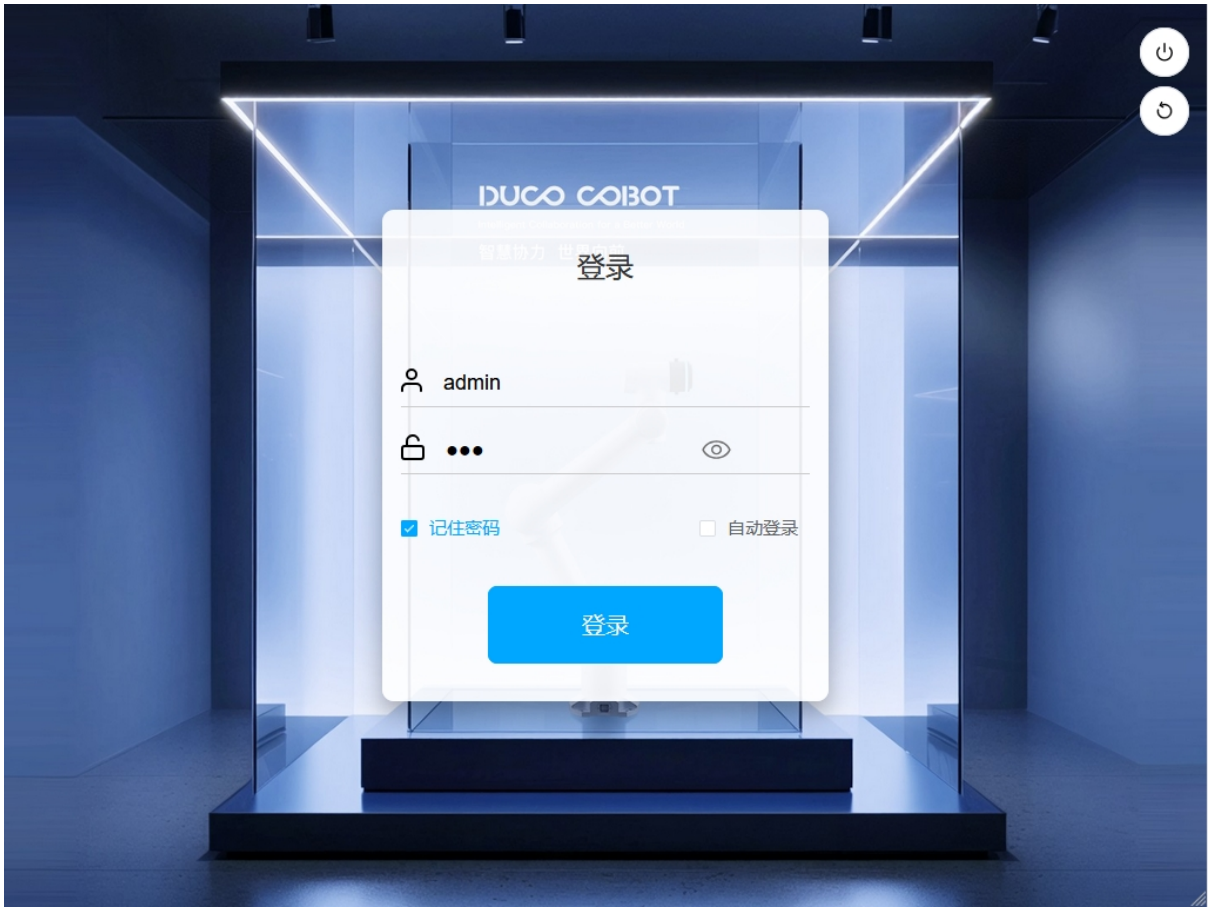
用户可以使用示教器或者自己的移动设备连接控制柜。使用移动设备时可通过浏览器（推荐使用 Chrome 浏览器）访问机器人。连接方式有:

示教器连接：直接插入示教器，系统启动时会自动启动界面程序。

有线连接：使用网线连接 PC 和控制柜后，在浏览器地址栏输入 IP 地址及端口号 7000，如：192.168.1.10:7000（IP 地址可在“系统设置”中修改），默认 IP 地址为 DHCP 动态分配。

无线连接:启动控制柜电源后,使用移动设备或者 PC 查找无线网络“DucoCobot_xxxxx”(控制柜 SN 号后五位)，输入默认密码“1234567890”,(无线网络名称和密码可在“系统设置”中修改）连接到该网络。在浏览器地址栏输入 http:duco-cobot.com:7000。

输入地址确定后，等待几秒后即可进入登陆界面，如图所示。



登陆界面右上方是关机按钮和重启系统按钮（重启功能仅 J9 及 J9-ZII 系列控制柜支持），中间是用户输入用户名和密码区域。登录按钮上方是记住密码和自动登录选项，用户可选择记住密码或自动登录功能。若勾选“记住密码”选项，用户注销后再登录不需要输入密码，点击“登录”按钮直接登录即可；若勾选“自动登录”选项，默认会自动勾选“记住密码”选项，用户关机或者重启后进入登陆页面再登陆时不需点击“登录”按钮就可以直接自动登录，进入系统概览页面。用户注销或恢复出厂设置后进入登录页面会取消“自动登录”选项，需要用户手动点击“登录”按钮登录进入系统。

机器人系统默认拥有两个系统默认账户，默认账户 `default`，密码 `123`，以及默认管理员账户 `admin`，密码 `123`。点击用户名或密码输入框，界面下方会弹出输入键盘，如图所示。输入用户名和密码（通过外部实体键盘也可以正常输入）后，点击“登录”按钮，即可登陆成功。

2.1.3 机器人启动

用户登陆成功后，默认进入概览界面。在机器人未上电情况下，会自动弹出机器人启动界面，如图所示。

启动界面左侧显示机器人安装方向和末端负载信息，右侧显示机器人关节状态。机器人初始未上电时，各关节状态指示灯颜色为红色。用户在进行机器人上电操作前，需要先确认机器人当前安装方向以及安装在机器人末端的负载是否正确配置。错误的安装方向与末端负载配置可能会引起机器人上电及上使能过程中产生异常报错。

点击启动界面右下角“上电”按钮，机器人开始上电过程。待机器人上电就绪后，关节状态指示灯颜色变为橙色，表示关节已上电并进入待使能状态，机器人处于上电且未使能状态。界面右下角会出现“断电”和“使能”按钮，如图所示：

点击“断电”按钮，机器人会断开动力电并回到未上电状态。点击“使能”按钮，机器人会对所有关节进行使能，使能过程中各关节可能会产生微小的移动。

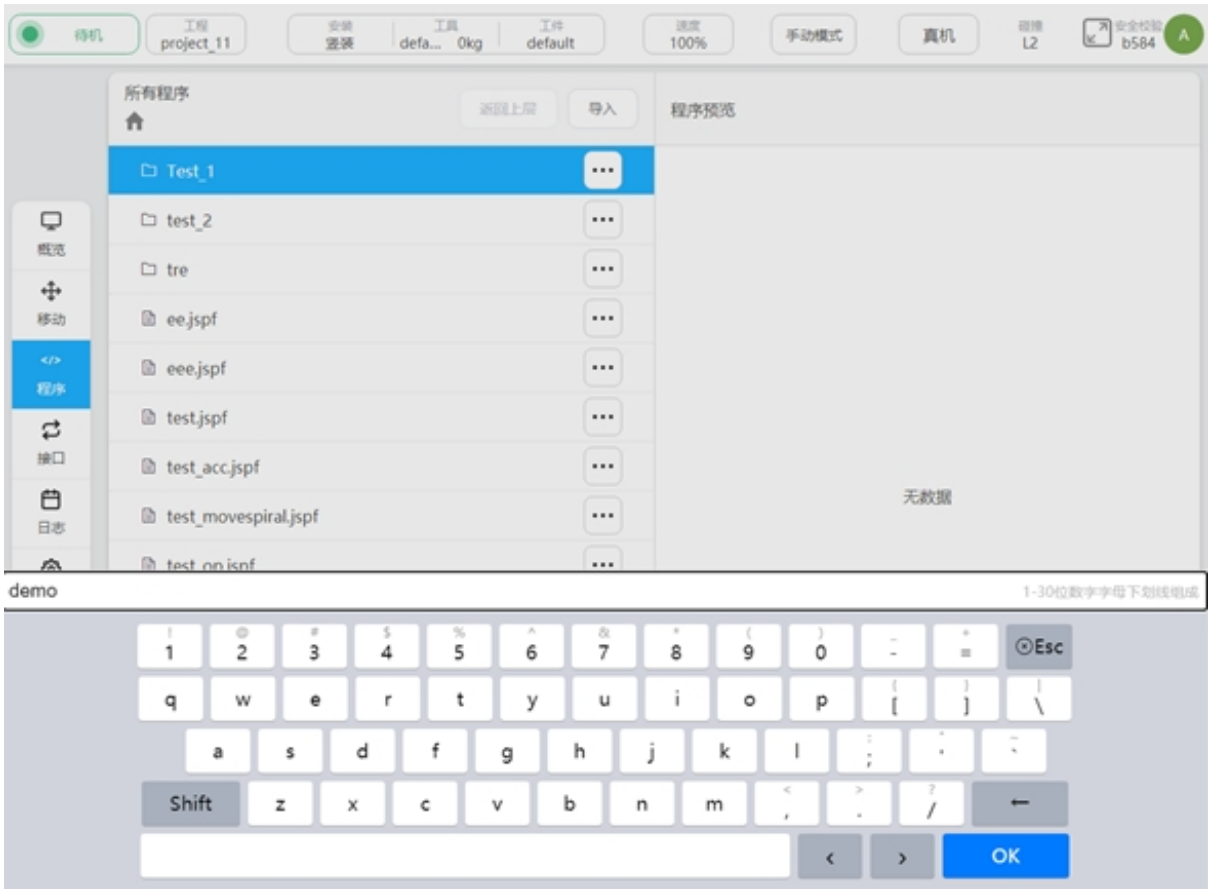
备注： 注意：使能阶段产生响动属正常现象。

待各个关节使能完成后，关节状态指示灯颜色变为绿色，此时机器人进入使能状态，机器人完成上电及上使能过程，如图所示：

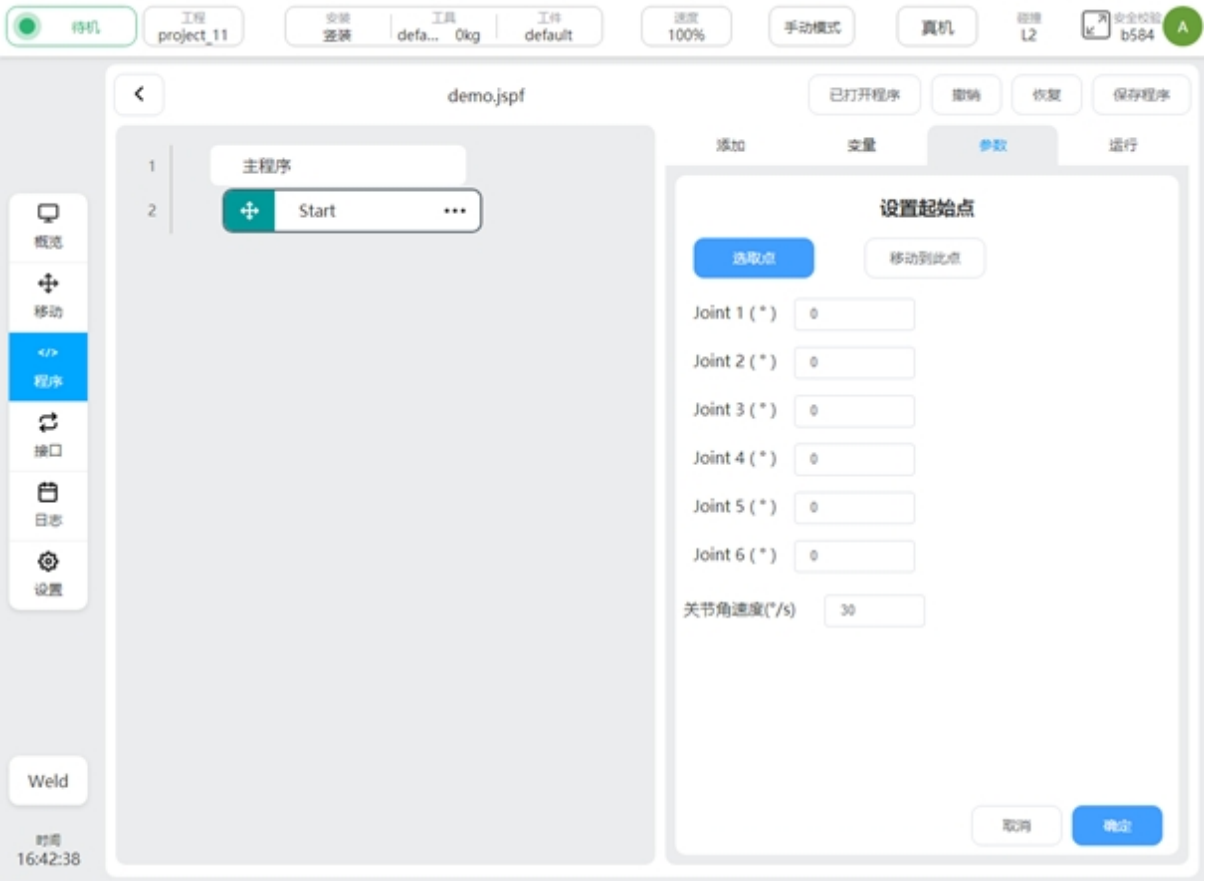
机器人完成使能后，启动界面会自动消失。

2.1.4 创建一个程序

点击左侧导航栏“程序”，切换到程序页面，点击下方的“新建程序”，输入程序名：demo。点击“Enter”，即创建了一个新程序 demo.jspf，并进入了编程页面。

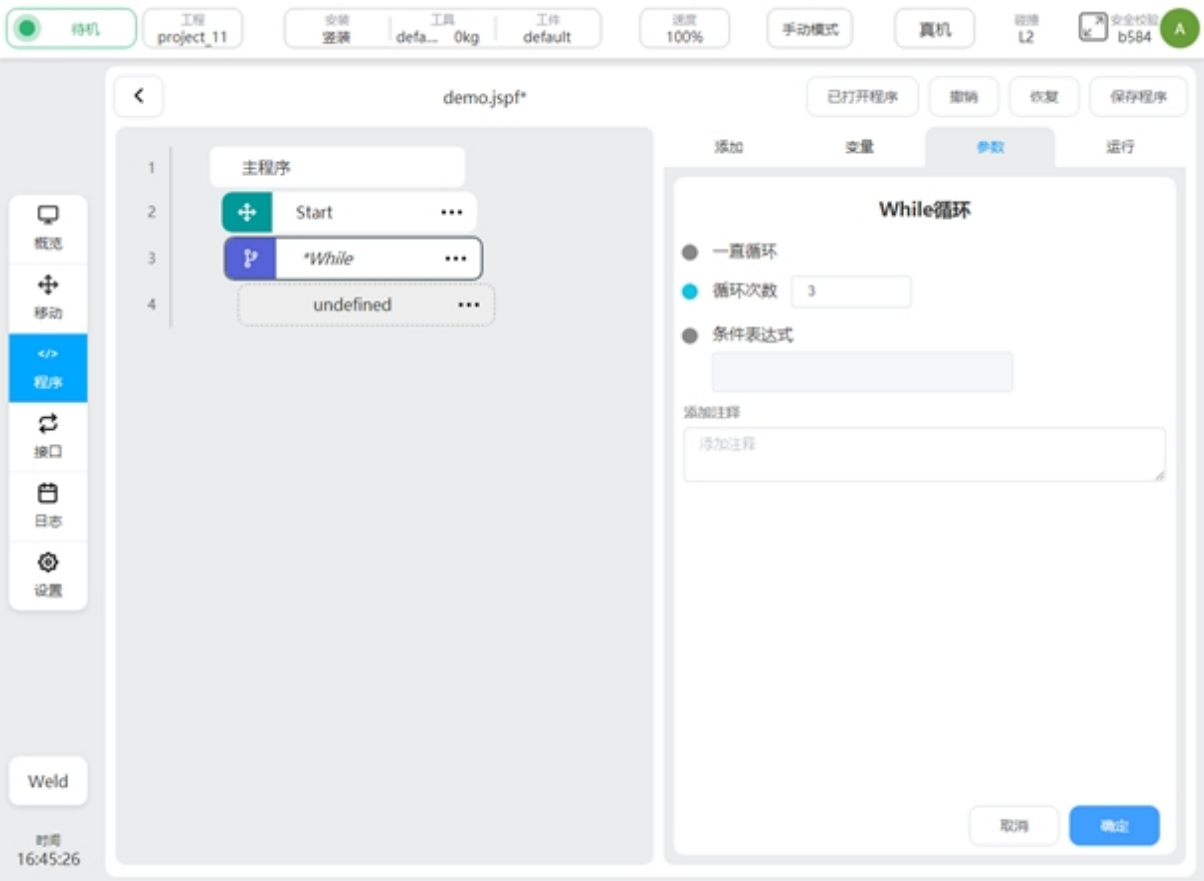


demo 程序默认包含了一个 Start 功能块来设置机器人的起始路点，双击 Start，或者点击右侧“参数”可以查看及设置起始点。

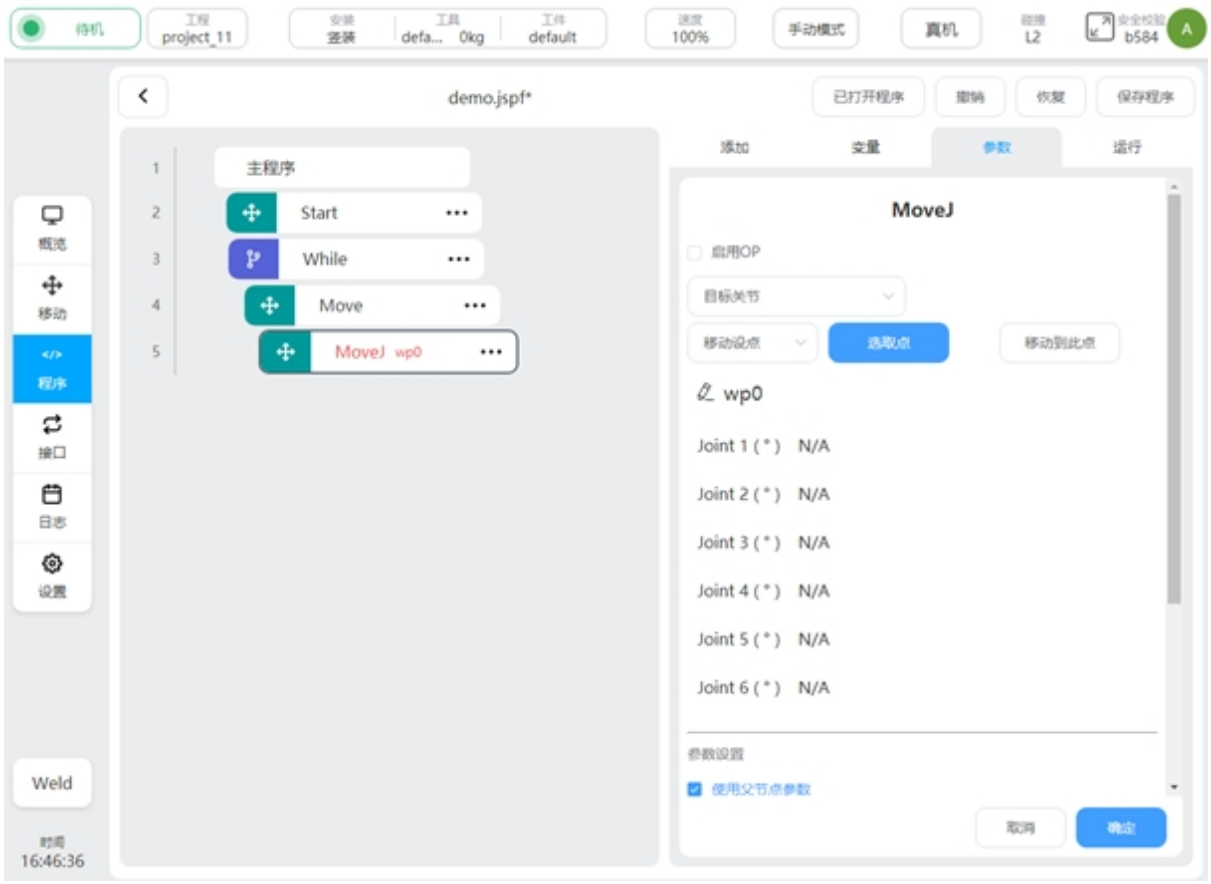


用户通过程序编程实现如下功能：机器人移动到准备点，向下移动到一个工作点，等待 2s，设置数字输出端口 1 为高电平，等待 1s，将数字输出端口 1 恢复为低电平，该过程循环 3 次。

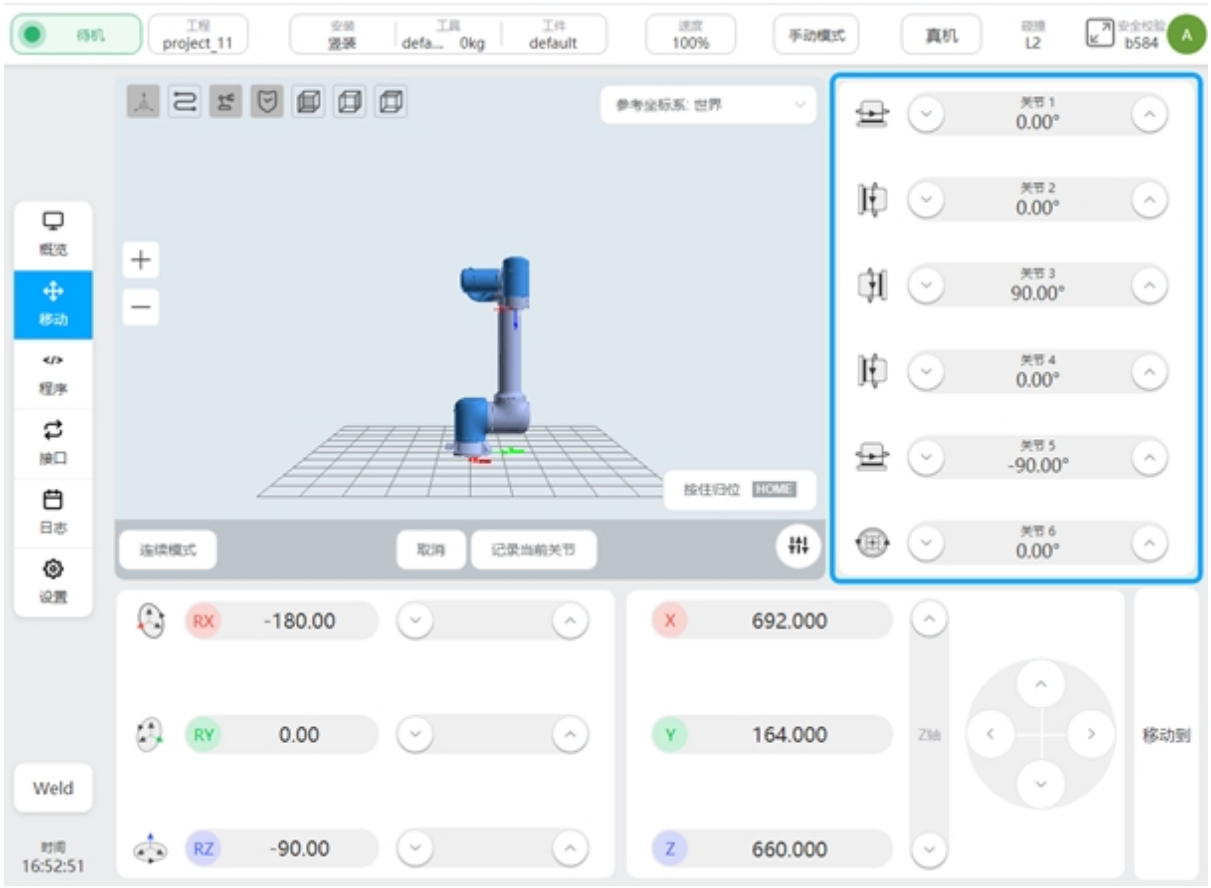
在“添加”标签页——“流程控制”中找到 While 功能块，该功能块可实现循环流程，按住将其拖动到左侧程序树中，释放即可将 While 功能块添加到程序树中。双击 While 功能块，可配置其参数，选择“循环次数”，输入 3。表示该 While 功能块下的所有程序段将循环执行 3 次。



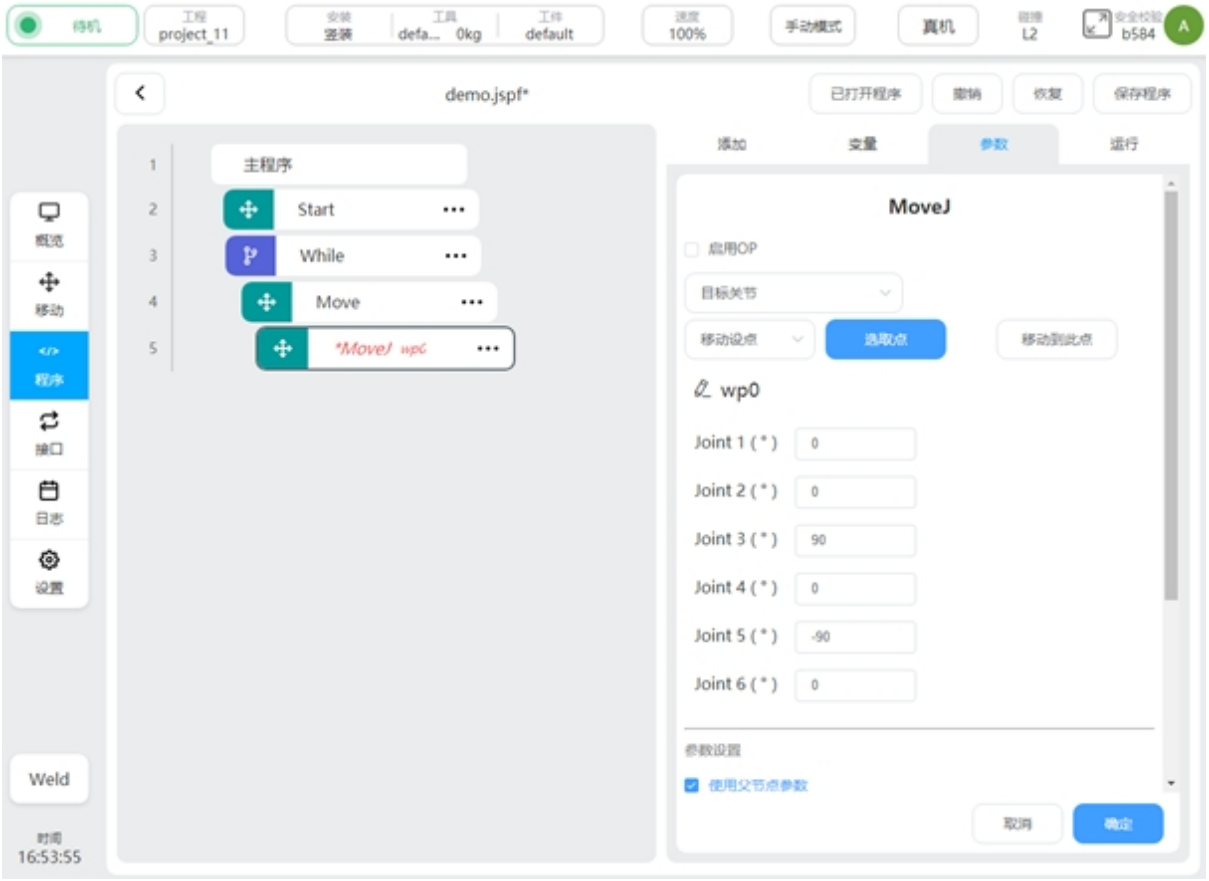
在“添加”标签页——“移动”中找到 MoveJ 功能块，将其拖动到程序树中 While 的子功能块中，如图，双击 MoveJ 功能块，配置其参数。点击“选取点”按钮，可切换到移动界面示教机器人的点位。



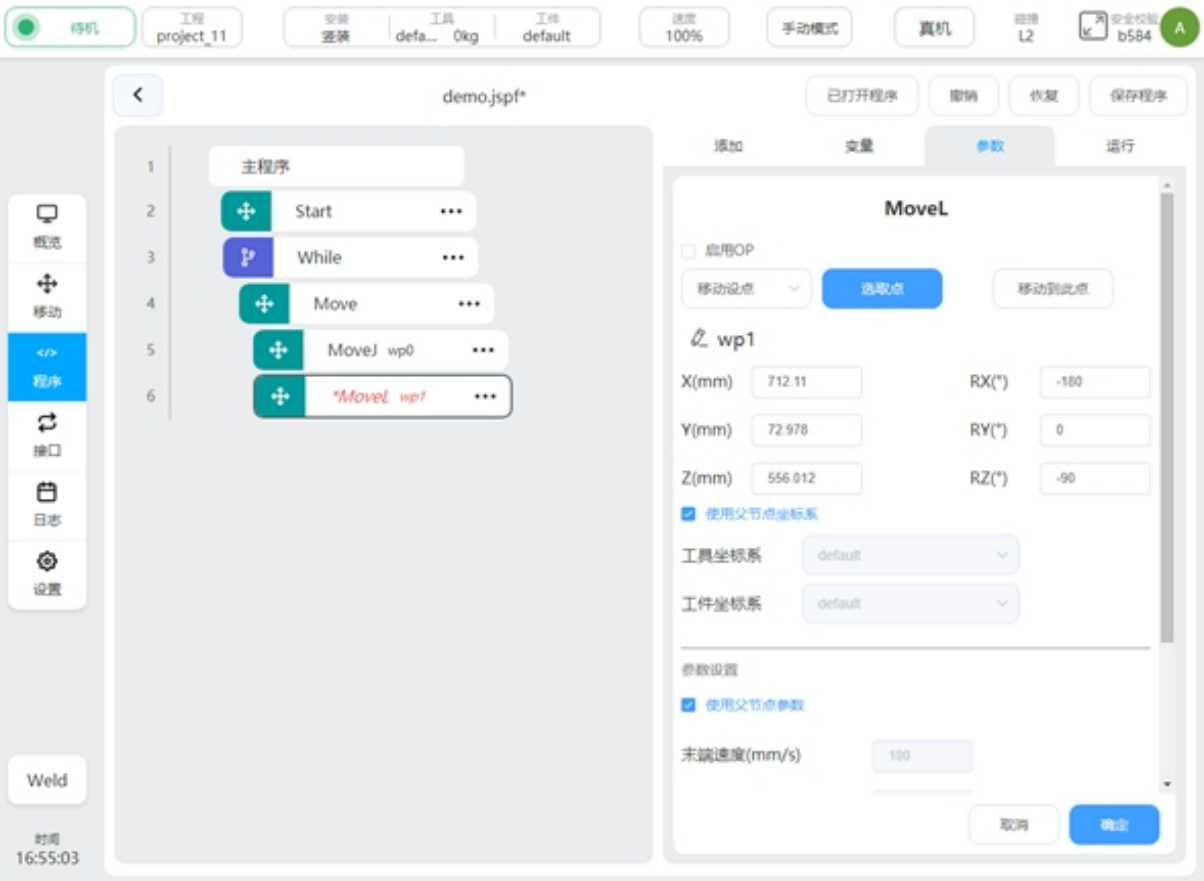
点位设定好后，点击“记录当前关节”可记录当前的机器人位置。



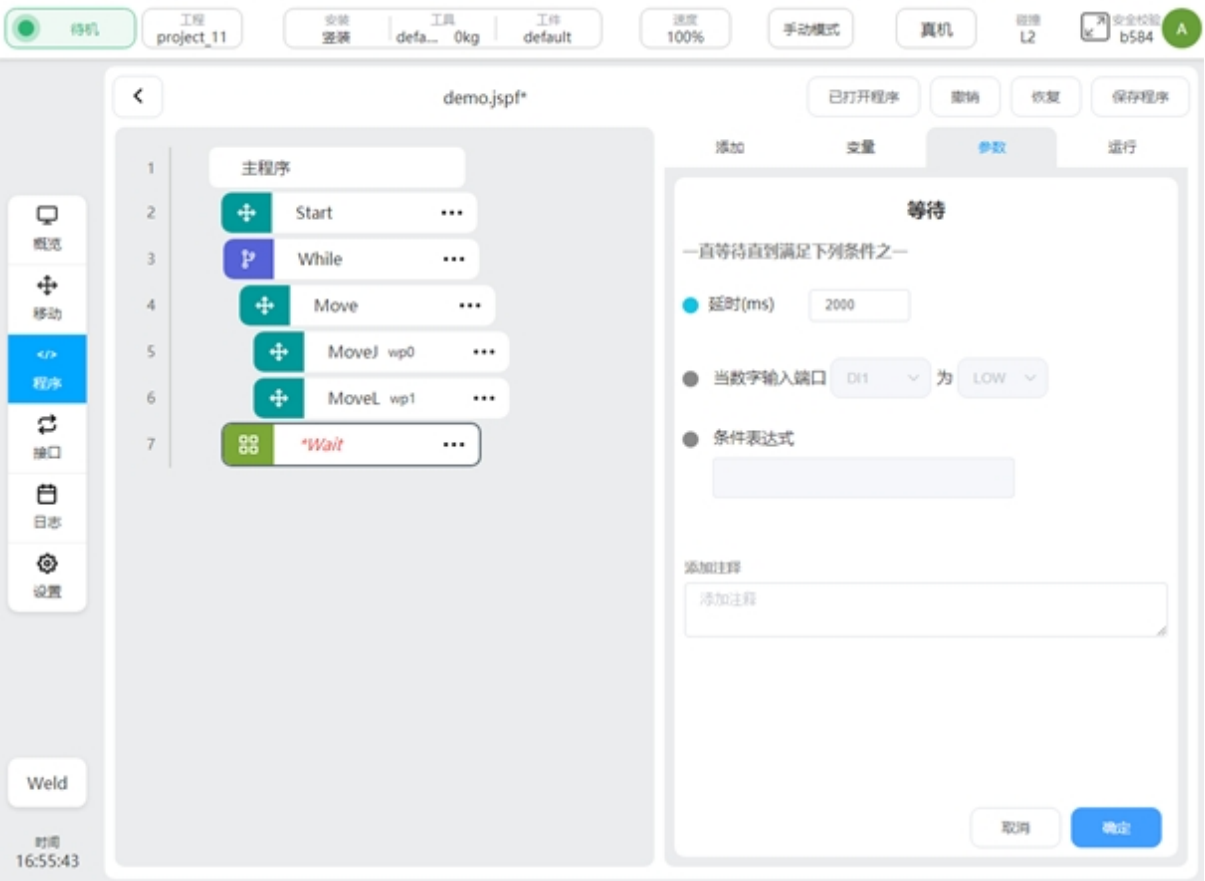
在 MoveJ 功能块的参数配置区可看到设置的点位信息。可进一步设置运动的关节角速度、关节角加速度。点击下方的“确定”按钮，保存该功能块的参数。



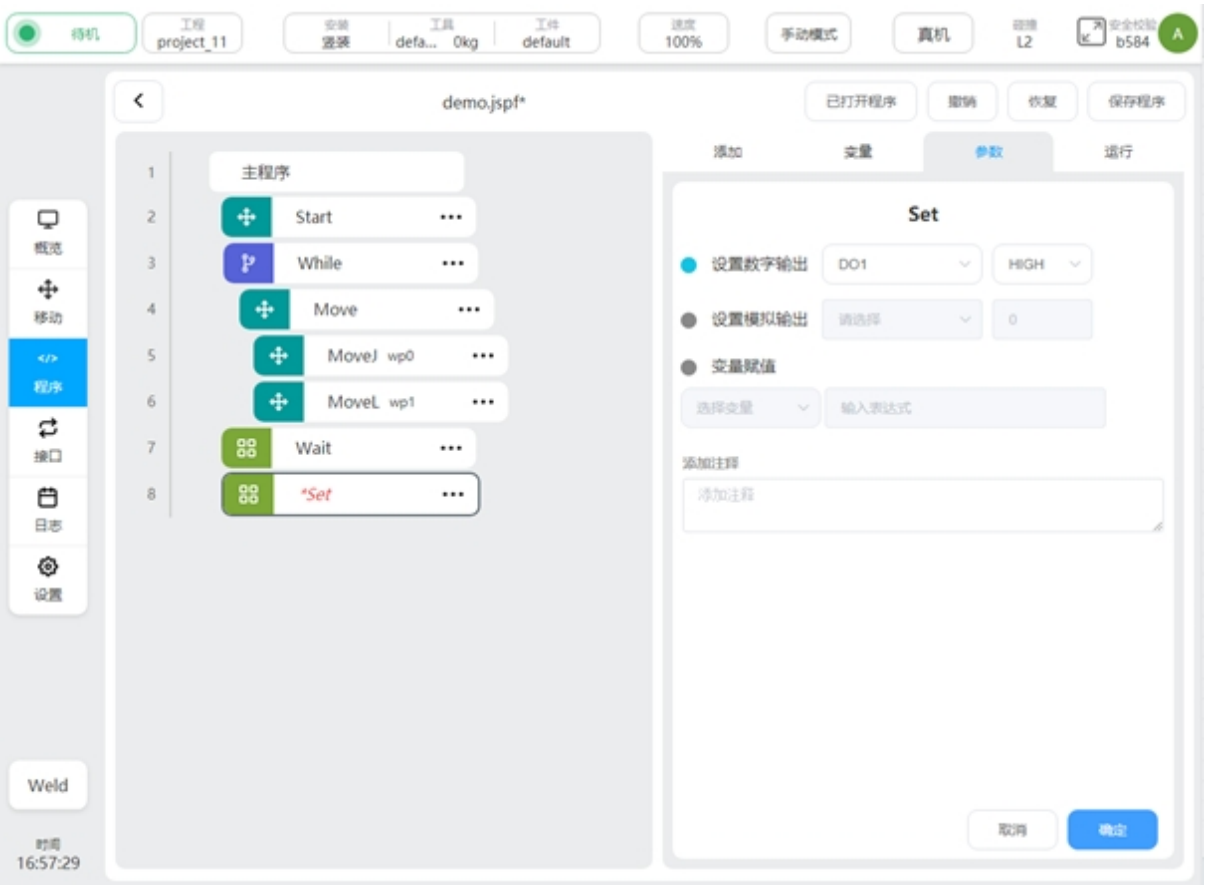
同样的方式，添加 MoveL 功能块到 MoveJ 功能块的下方，设定点位。



在“添加”标签页——“基础”区找到 `Wait` 功能块，拖动到如图位置，在其参数配置区选择“延时”，输入参数 `2000`，表示程序执行到此将延时 `2s` 执行后续功能块。



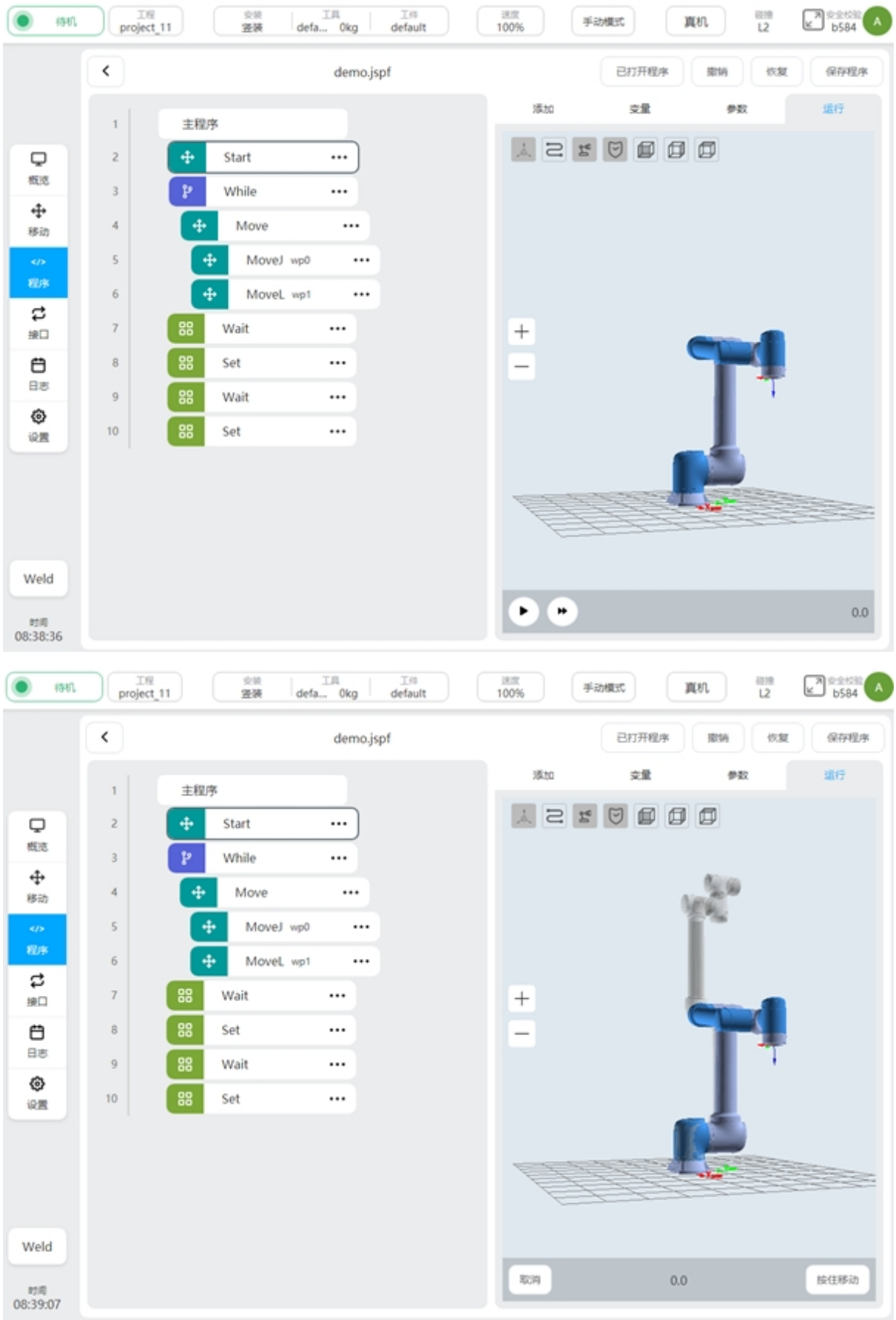
在“添加”标签页——“基础”区选择 Set 功能块将其拖动至 Wait 下方。在其参数配置区选择“设置数字输出”项，选择 DO1、HIGH。表示程序执行到该功能块时，将数字输出端口 1 置为高电平。



同样的，添加一个 Wait 功能块，设定延时 1s，添加一个 Set 功能块，设定将 DO1 置为低电平。以上即完成该功能的程序编写，点击上方的“保存程序”按钮，保存该程序

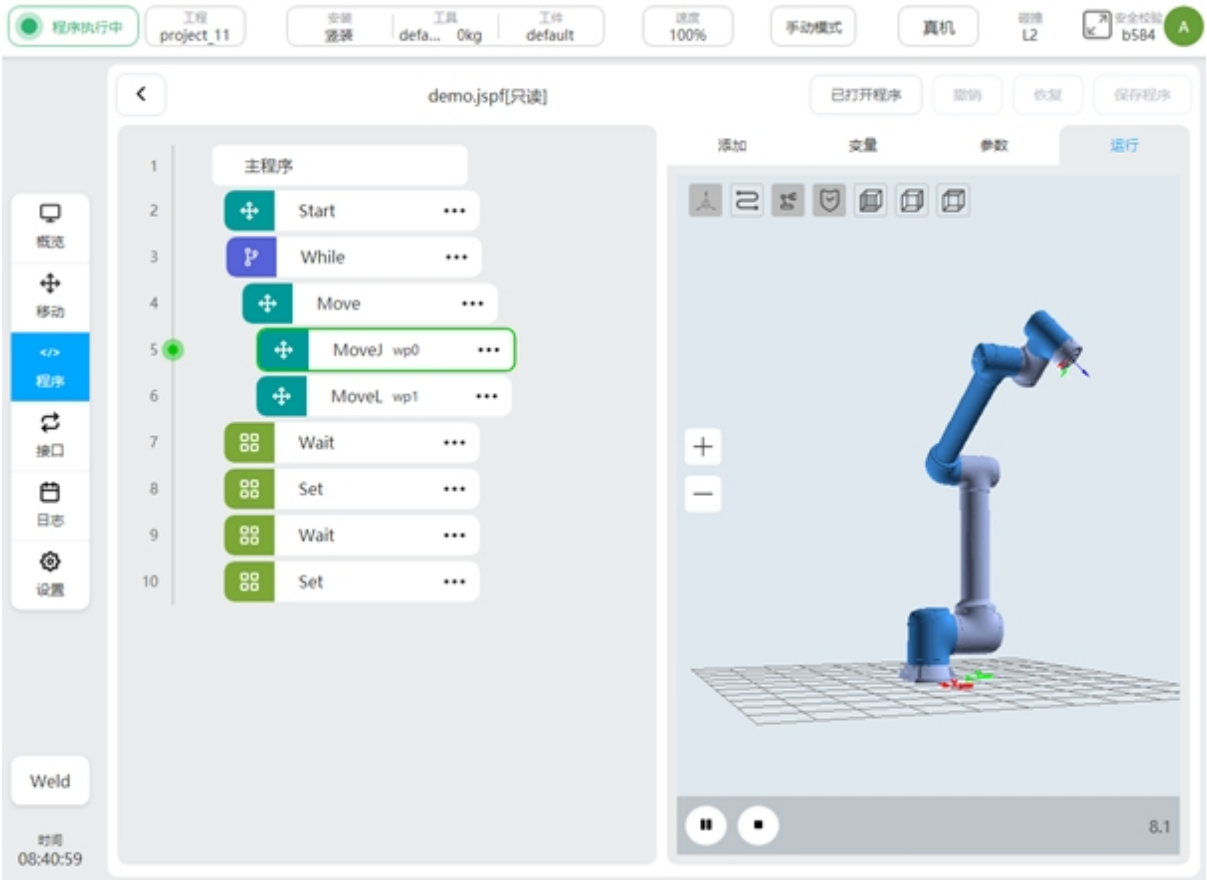
2.1.5 运行程序

切换到“运行”标签页，点击下方的运行按钮，若机器人的当前位置为程序的起始点不同，将显示“按住移动”按钮，并在 3D 图显上显示机器人的起始位置。长按“按住移动”按钮将机器人移动到起始位置后。再次点击运行按钮即可运行该程序。



程序执行时，如下图所示，程序树前面的绿色圆点用来指示当前正在执行的功能块，且当前执

行的功能块显示绿色高亮边框，3D 模型中可以显示机器人的实时姿态。在程序运行过程中，可暂停或者停止程序。



2.2 系统概述

2.2.1 工程管理

系统支持创建多个工程并对其进行管理，工程数据包含机器人程序、工程全局变量、工程设置等信息。系统启动加载时，会根据系统设置加载当前工程数据，其他工程数据不可以在当前工程中使用。

小心： 所有的工程数据，如接口页面内的设置，设置页面内的设置等，在设置完后都需要保存工程，以便在下次开机时被正确加载。

2.2.2 用户管理

系统支持创建多个用户并对其进行管理，用户权限等级分为 Operator、Programmer、Maintainer、Admin。具体权限说明如下：

Operator

允许选择工程、运行程序、手动 jog 机器人、查看机器人状态等

Programmer

Operator 用户权限

编程机器人程序、工程配置

默认用户名 default，初始密码 123

Maintainer

Programmer 用户权限

系统更新

Admin

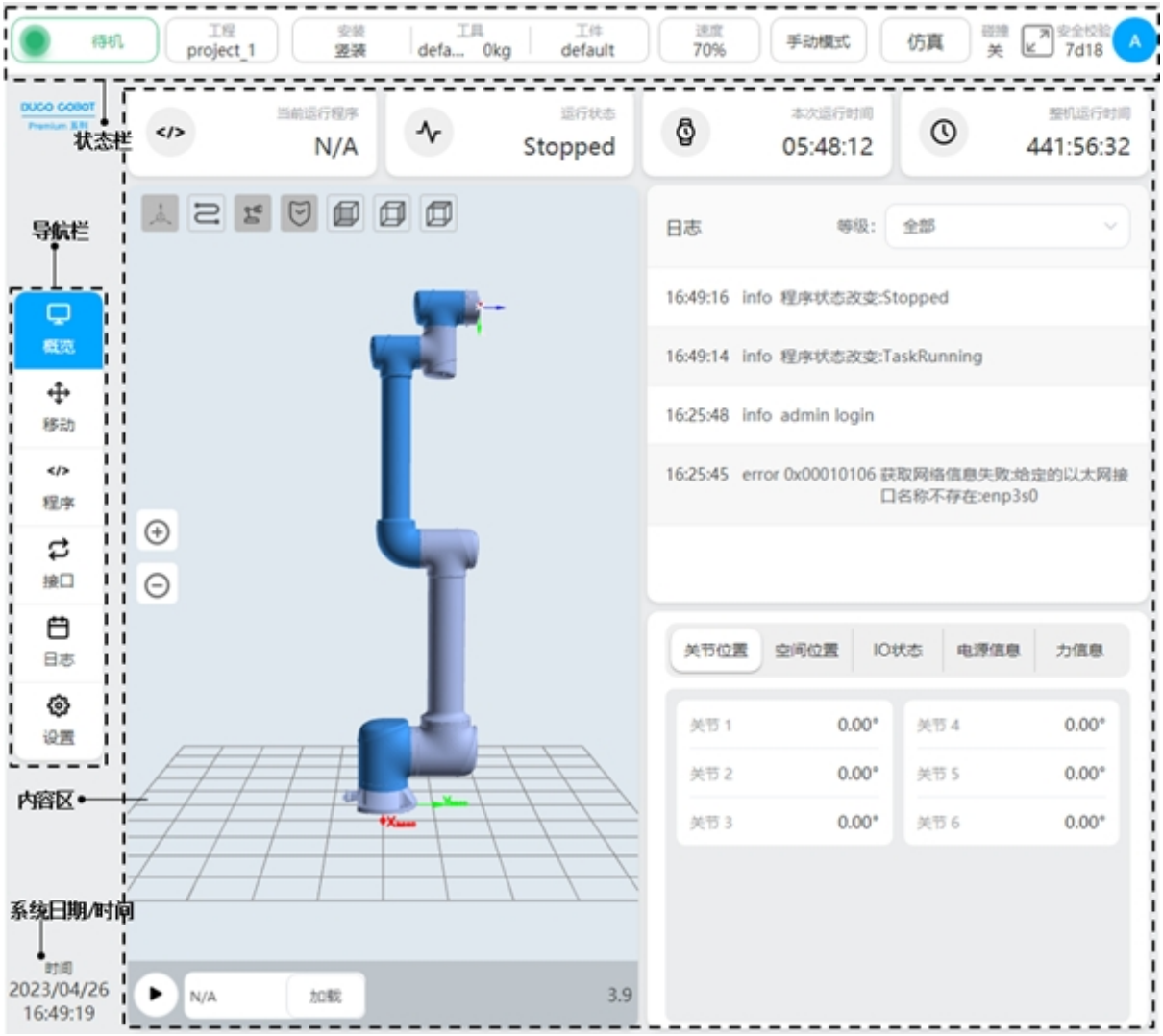
Maintainer 用户权限

用户管理

单一用户名 admin，初始密码 123

2.2.3 用户页面概览

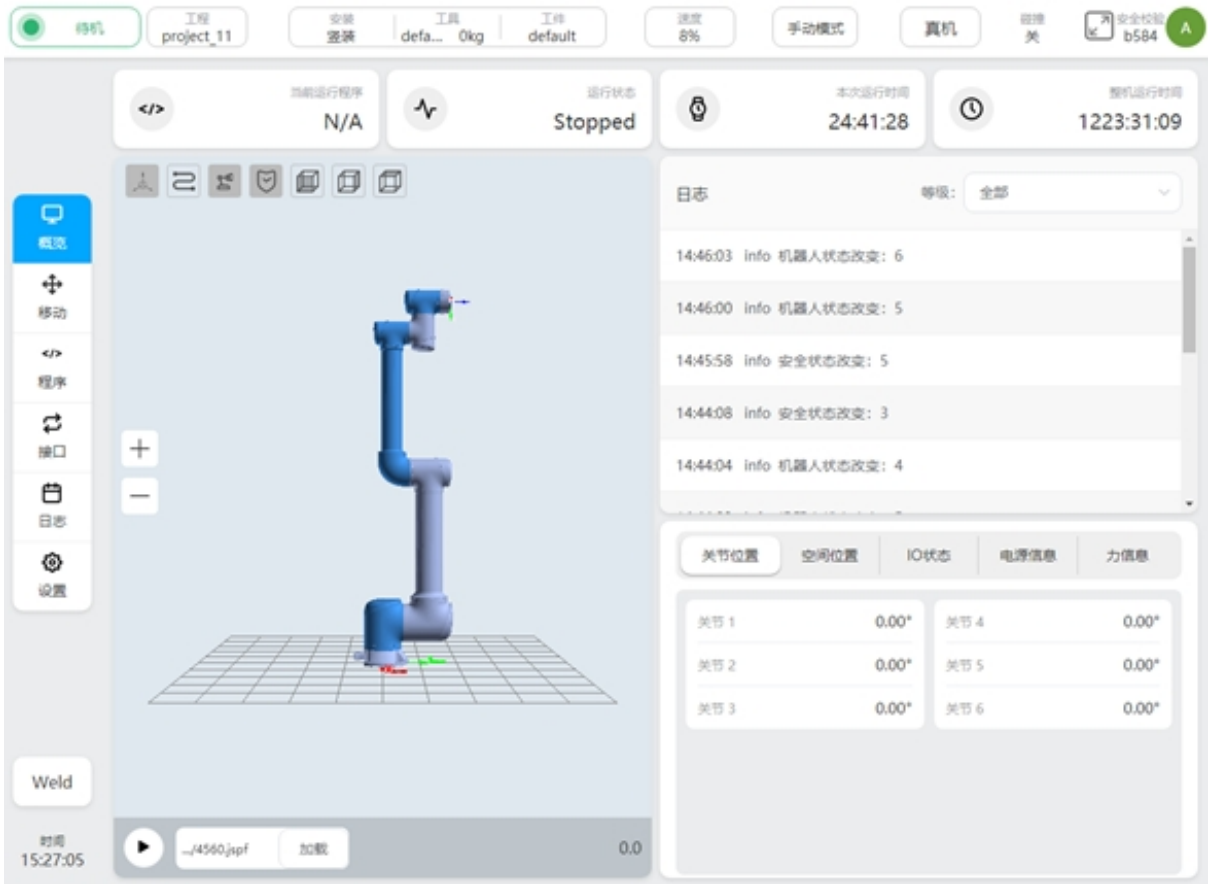
如图所示，界面整体主要分三部分区域，顶部状态栏，左侧导航栏，右侧内容区，其中内容区是根据选择左侧导航栏的不同而呈现不同内容（子页面）。导航栏下方是插件入口及系统日期和时间。



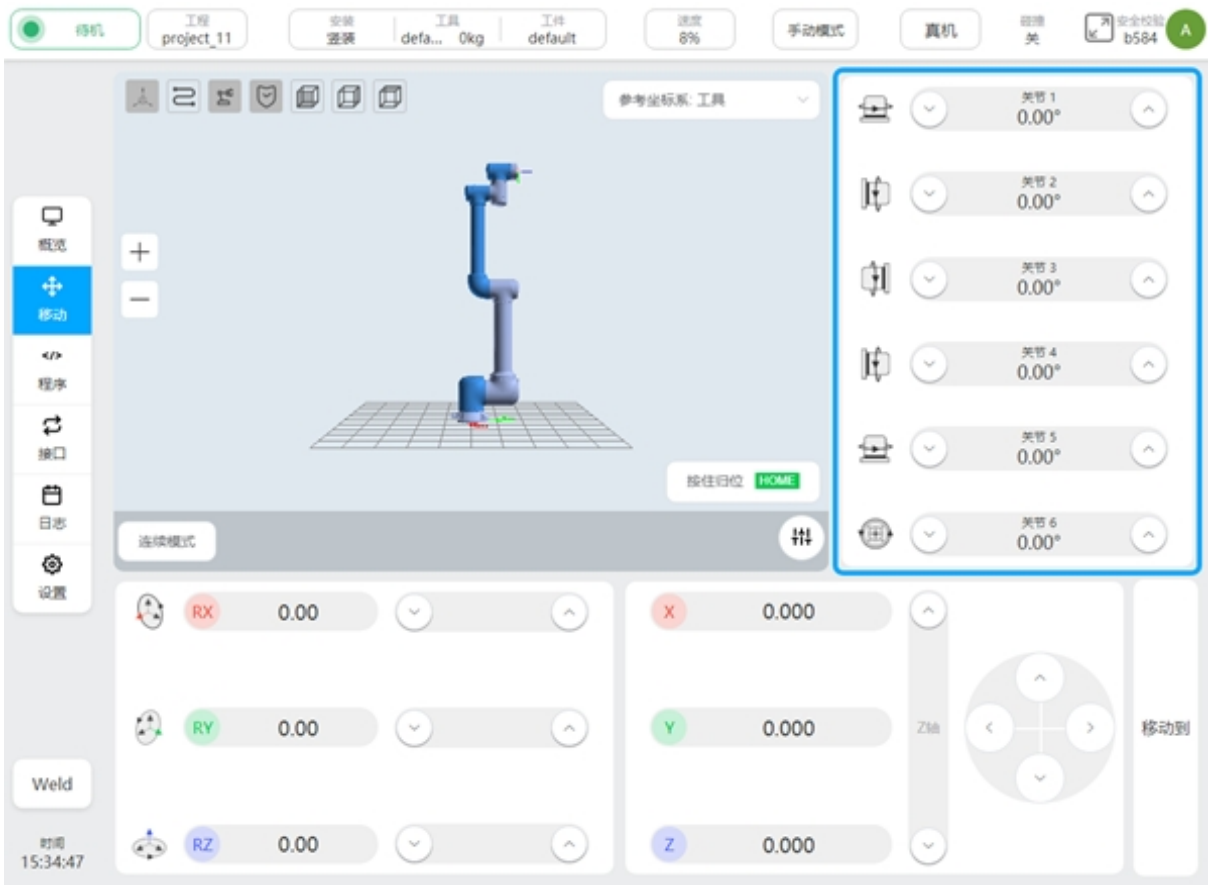
状态栏里主要展示信息有：机器人状态指示灯、当前工程、机器人安装方向、当前工具坐标系和工件坐标系相关信息、速度百分比、操作模式、碰撞检测设置信息、安全校验码、用户信息。

导航栏包含概览、移动、程序、接口、日志和设置 6 个图标导航，单击导航栏图标会对应切换内容区里显示的内容，各页面主要内容如下：

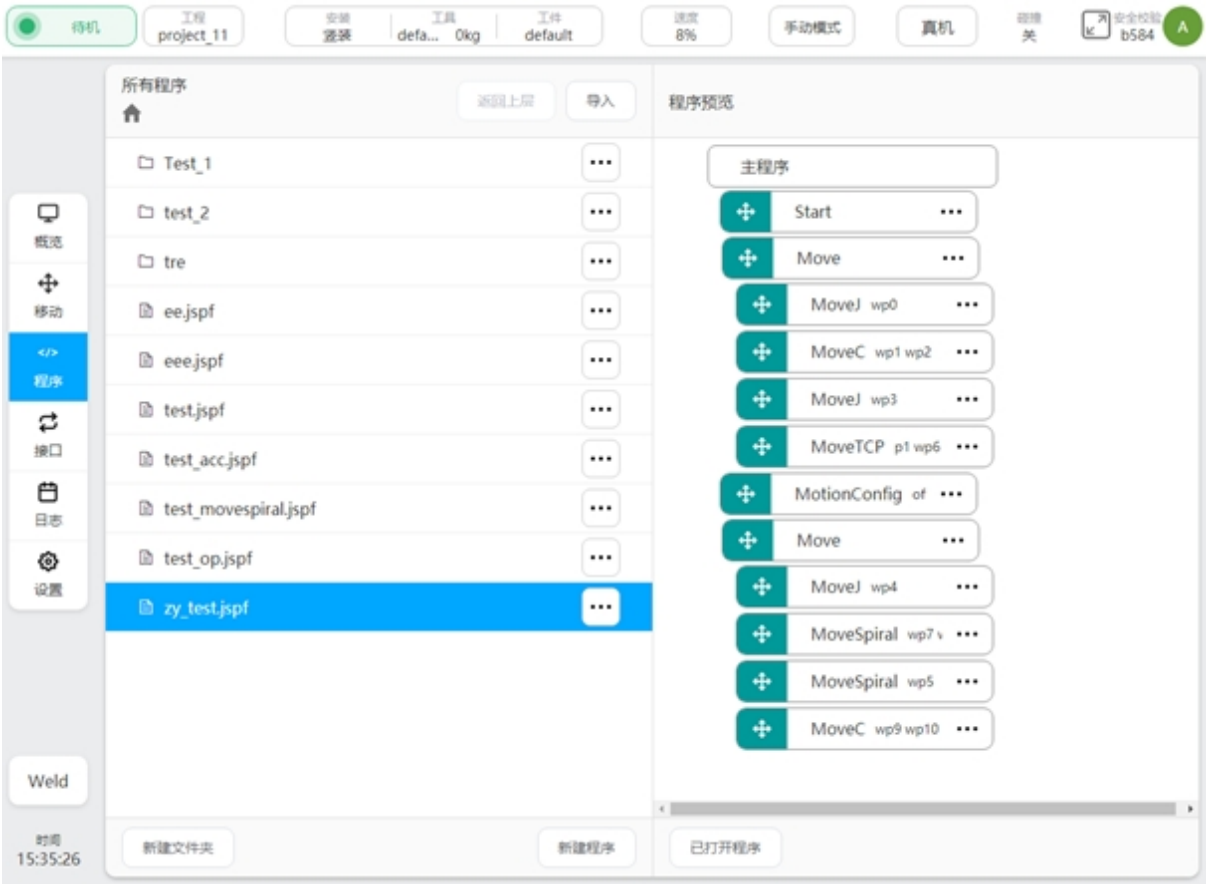
概览页面是生产视图页面，可以选择要运行的程序，显示程序状态、系统运行时间、系统日志以及机器人系统基本信息。



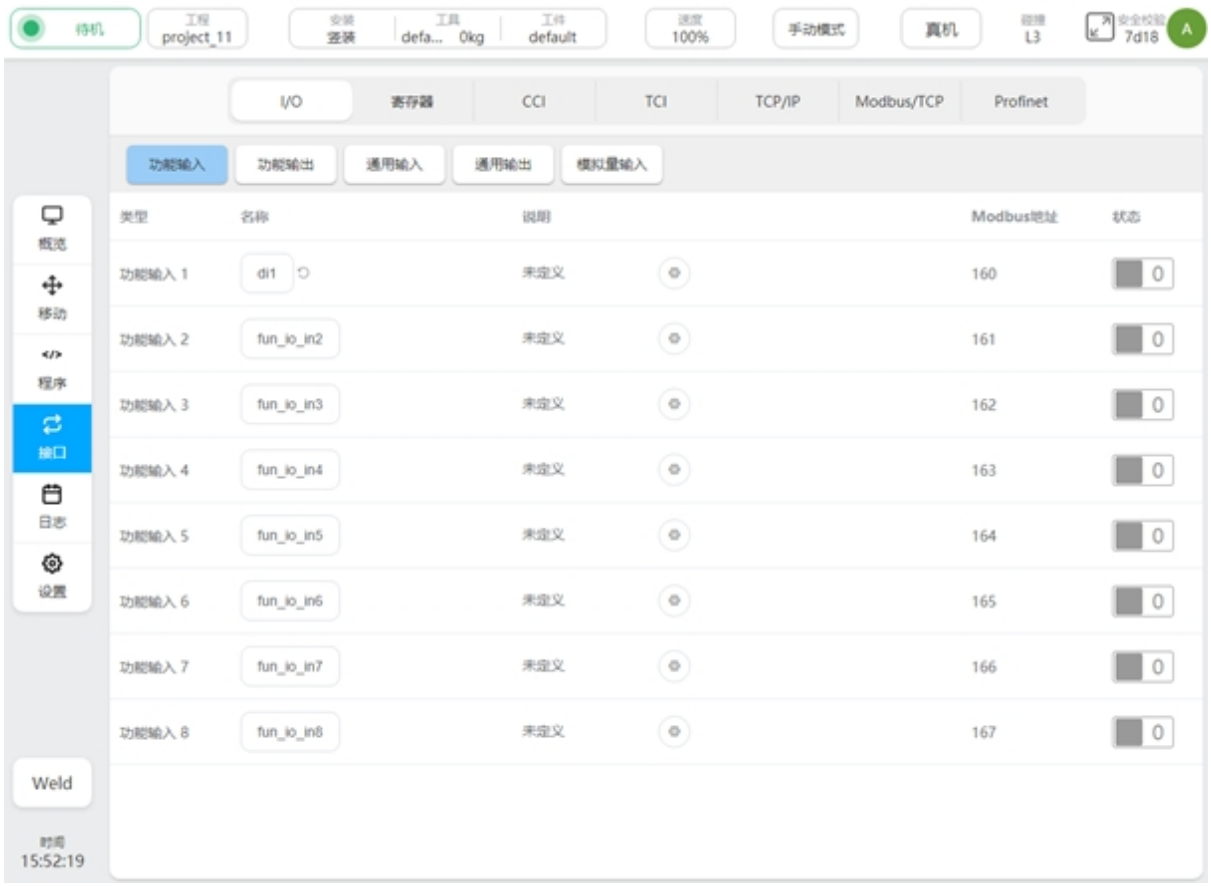
移动界面允许用户手动操作机器人，设置手动移动的坐标系及运动方式。



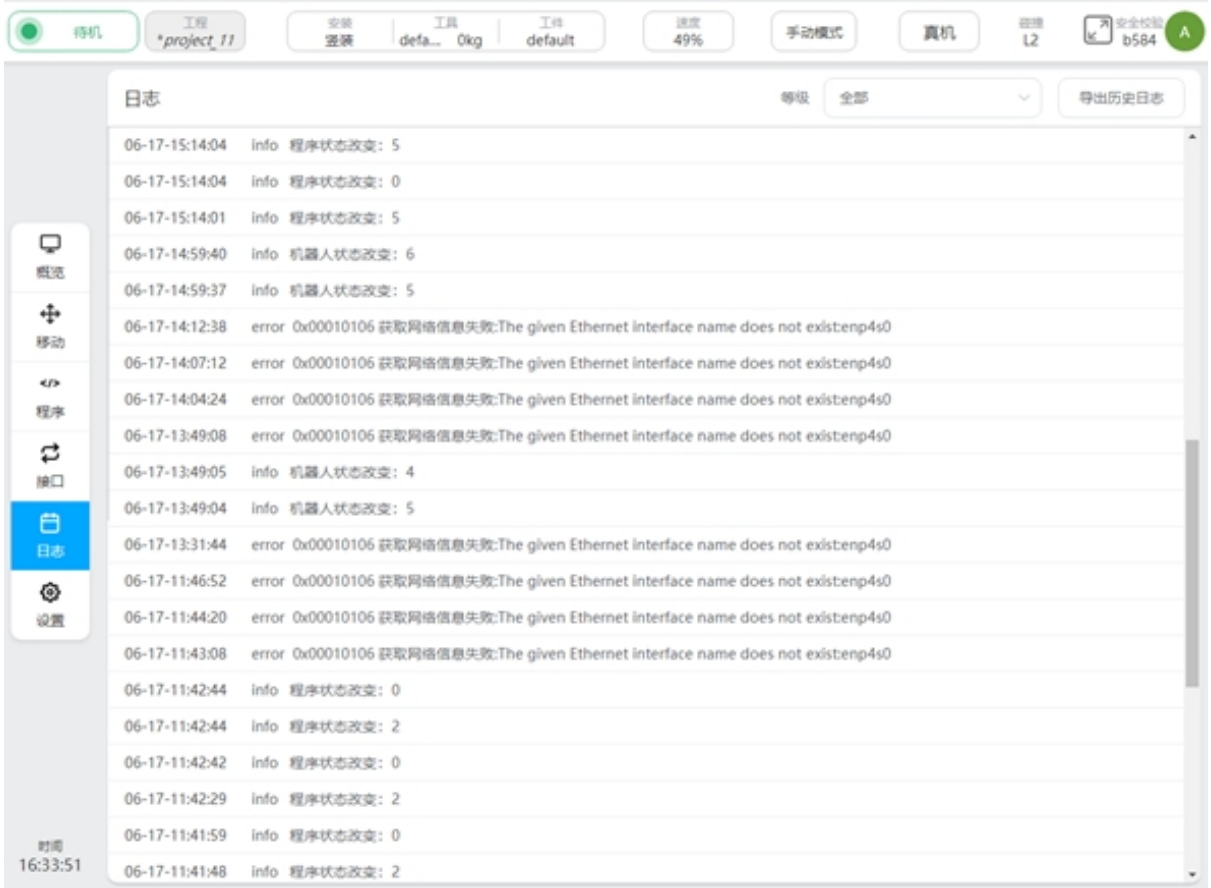
程序界面是机器人的主要编程页面，用户可以管理工程中的程序文件，以及通过提供的图形化编程环境编写机器人任务。



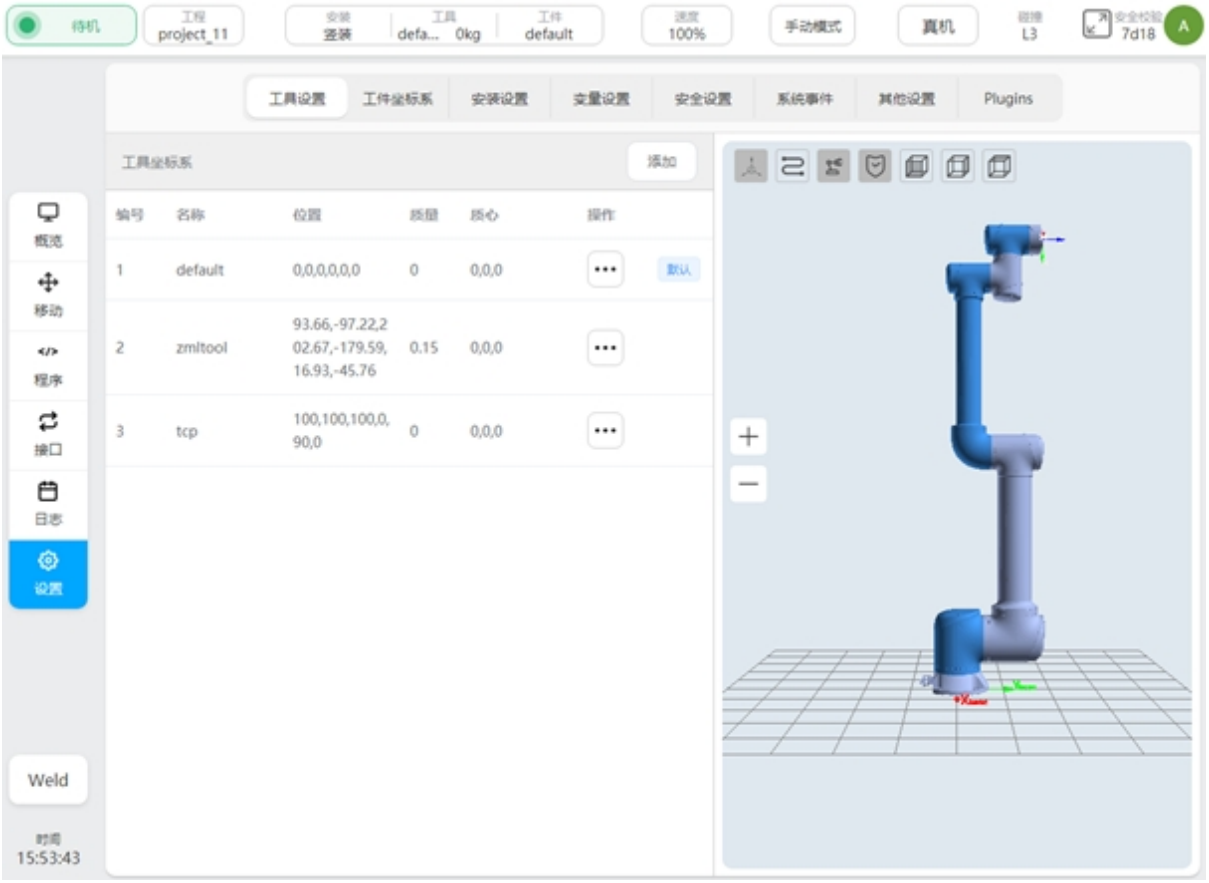
接口界面允许用户在此页面下配置机器人所有对外的交互接口，包括数字输入输出、模拟输入输出、TCP/IP 接口，以及其他工业现场总线接口。



日志界面显示本次开机系统的日志信息。

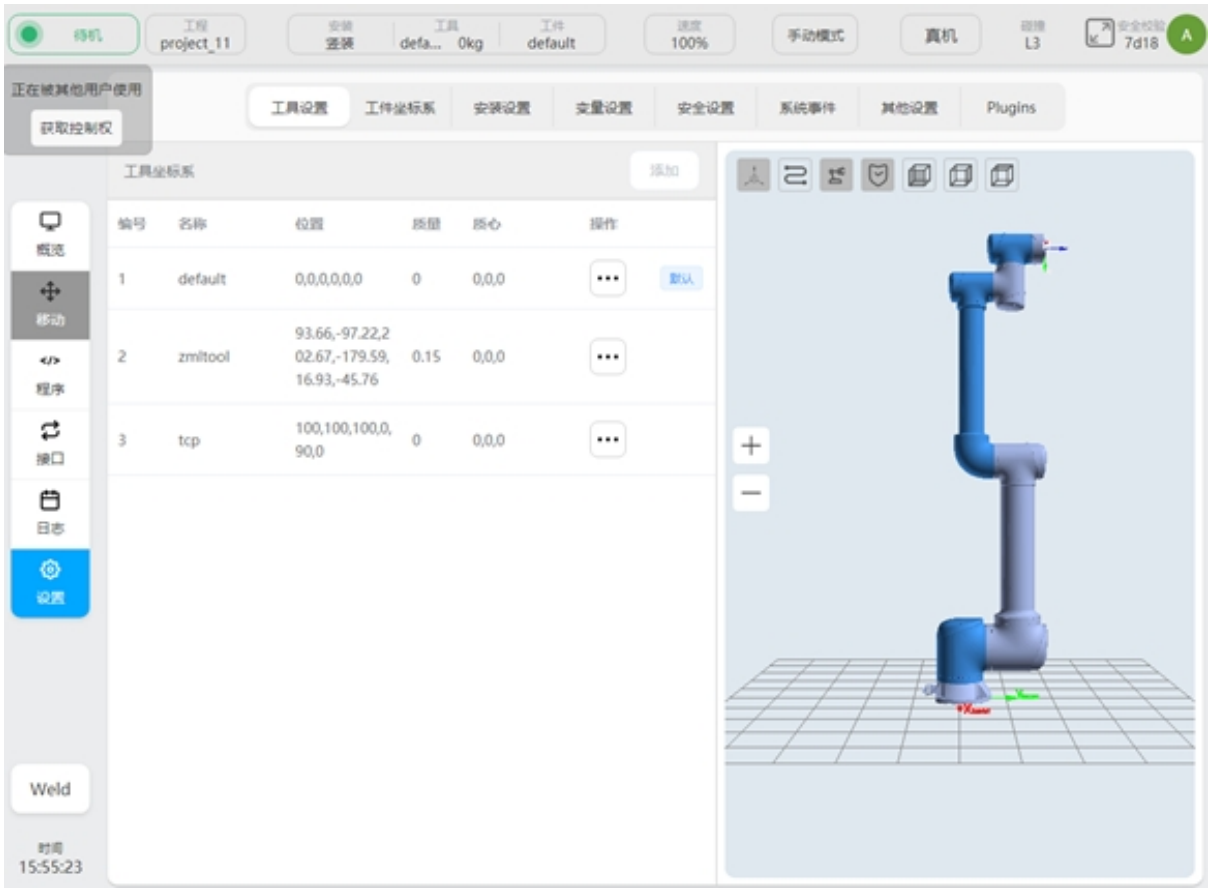


设置界面，当前工程配置设置，包含坐标系统的设置，安装方向的设置，以及安全设置等。

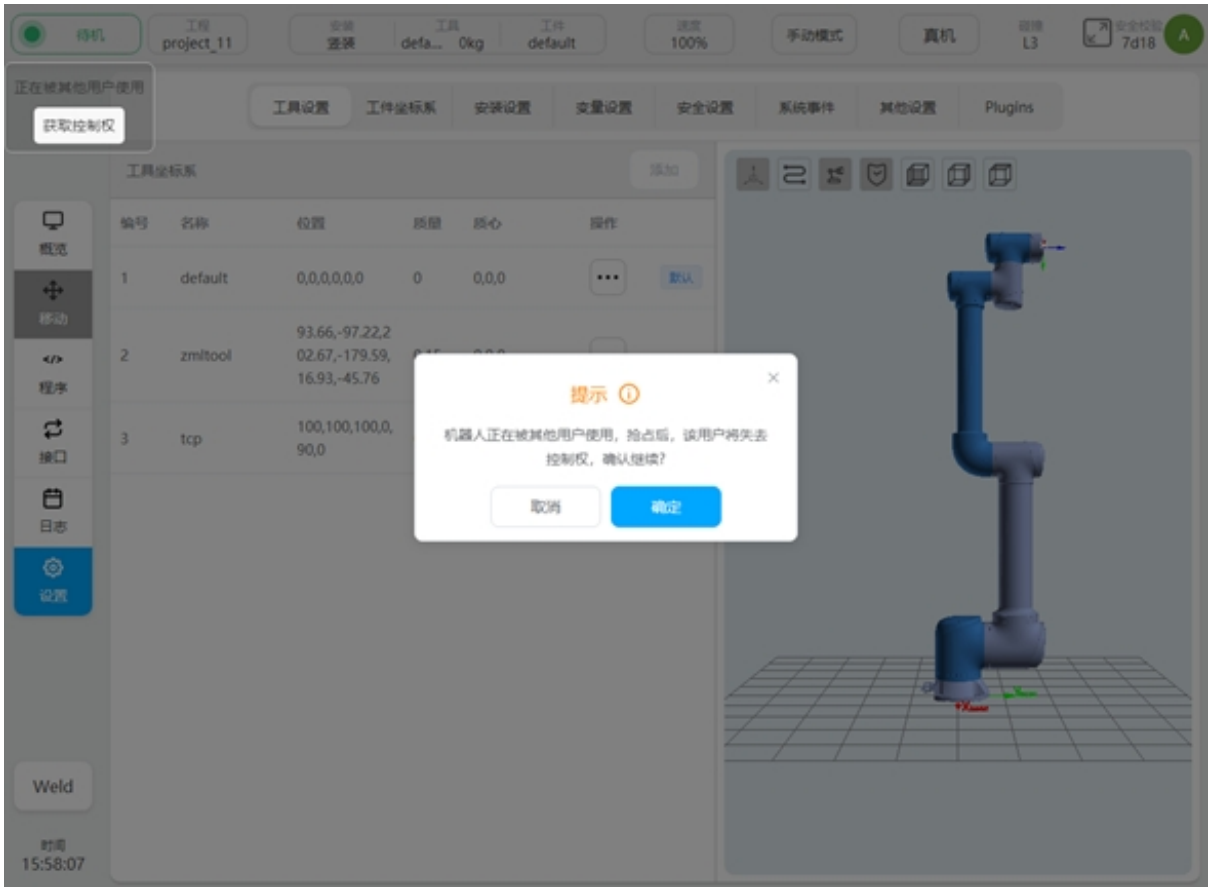


2.2.4 多终端连接

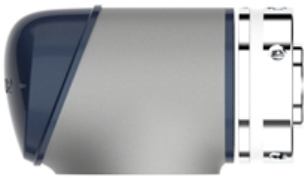
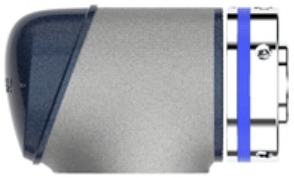
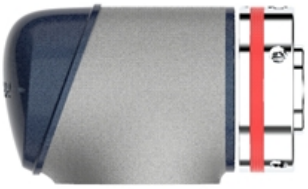
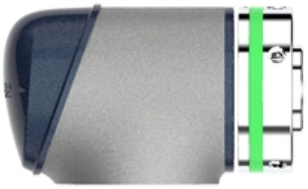
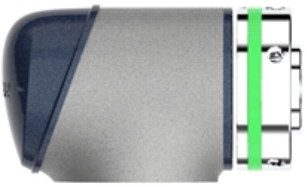
系统支持多台终端同时连接，但是只有一台终端拥有控制权，可以进行程序的运行、系统的配置等全部功能。其他终端只有查看的权限，不能运行机械臂及修改设置信息等。无控制权的终端显示如图：



当其他终端要获取控制权时，可以点击左上角的“获取控制权”按钮。弹出如下对话框，点击“确定”按钮获取系统的控制权，此时原来有控制权的终端将失去控制权。


待机
project_11
0kg
default
100%
手动模式
真机
L3
7d18
正在被其他用户使用
获取控制权
工具设置
工件坐标系
安装设置
变量设置
安全设置
系统事件
其他设置
Plugins
工具坐标系
添加
编号
名称
位置
质量
质心
操作
1
default
0,0,0,0,0,0
0
0,0,0
默认
2
zmltool
93.66,-97.22,2
02.67,-179.59,
16.93,-45.76
3
tcp
100,100,100,0,
90,0
概览
移动
程序
接口
日志
设置
Weld
时间
15:58:07
提示
机器人正在被其他用户使用，抢占后，该用户将失去
控制权，确认继续?
取消
确定

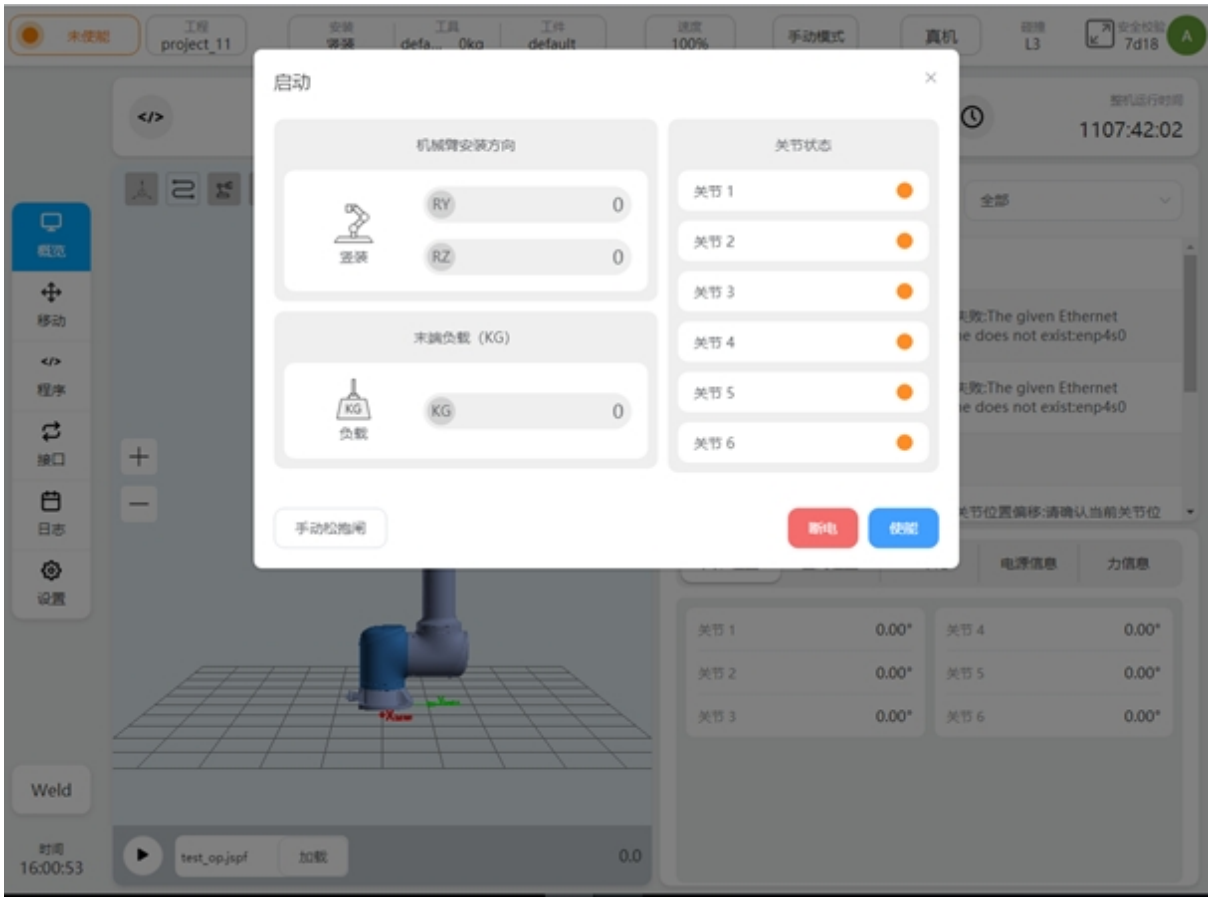
2.2.5 末端灯带指示

状态定义	颜色定义	示意图
1. 机器人上电 2. 机器人建立通讯	恒白色	
1. 机器人待机	恒蓝色	
1. 安全停止触发	恒红色	
1. 程序运行 2. 机器人回零 3. 手动移动到某一点	恒绿色	
1. 牵引示教	闪烁绿色	

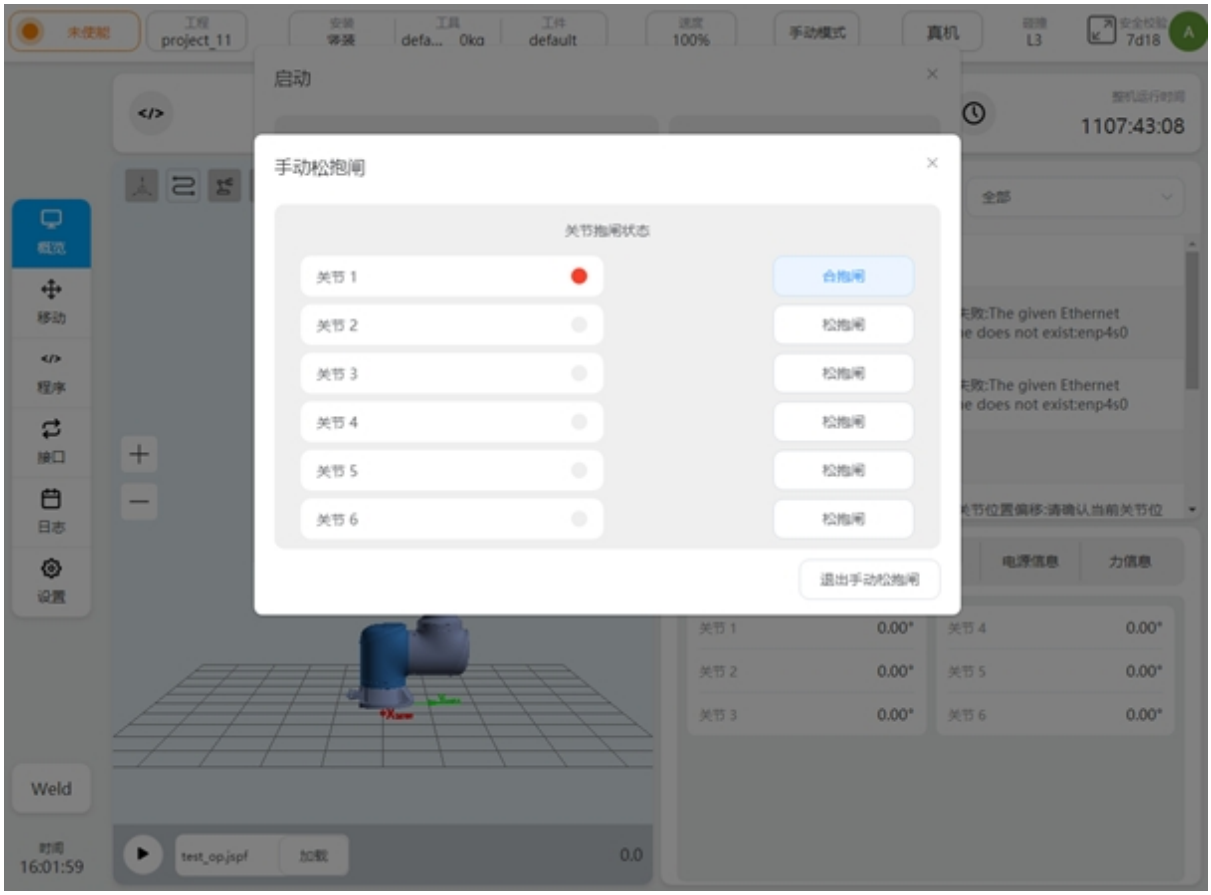
2.3 安全维护

2.3.1 手动松抱闸

在某些情况下，用户需要手动松开单个关节的抱闸。在机械臂上电未使能的情况下，启动页面会出现手动松抱闸的入口，如图



点击可进入手动松抱闸页面，如图。该页面左侧显示各个关节的抱闸状态，灰色表示抱闸闭合，红色表示抱闸松开；右侧为抱闸控制按钮，点击可控制抱闸的分合。

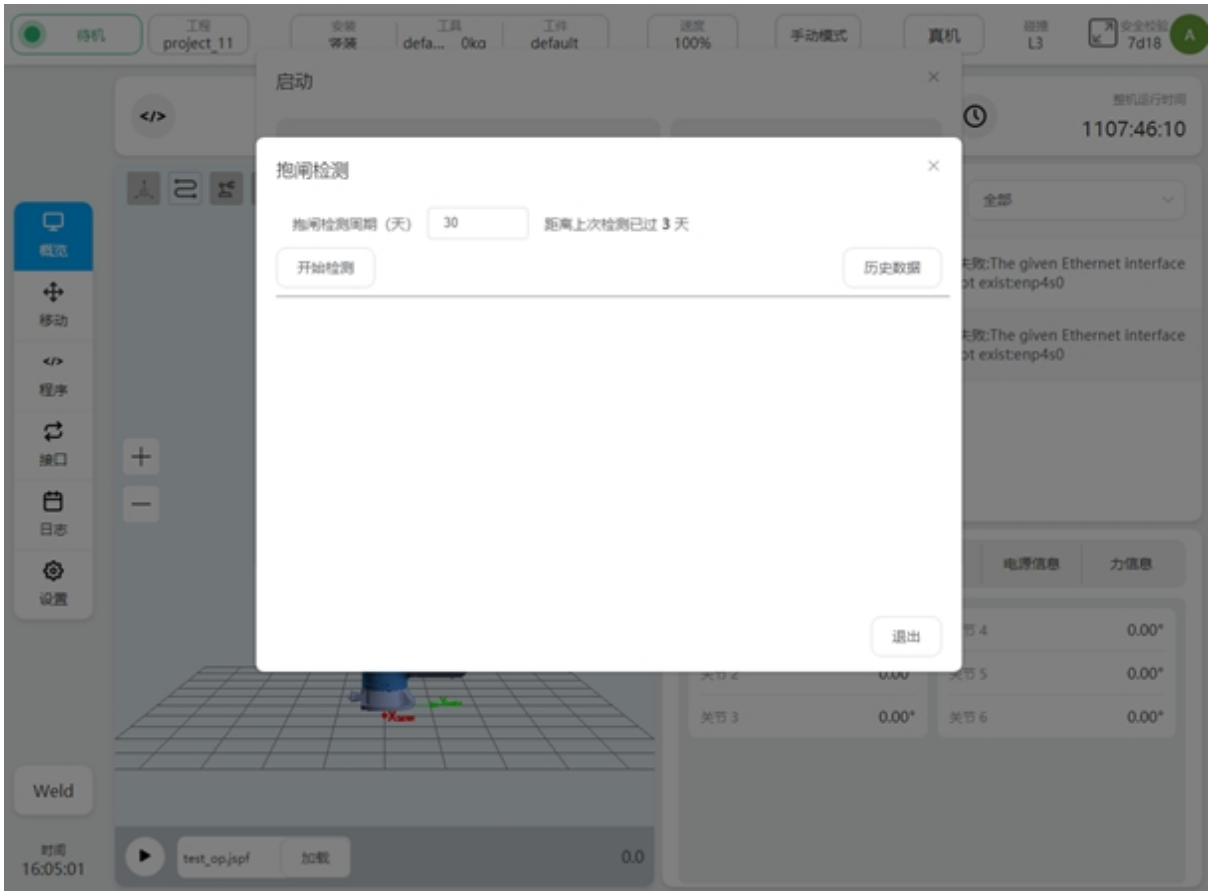


2.3.2 抱闸检测

用户需要按期进行抱闸检测，防止机械臂因抱闸失效带来风险。系统会周期性的提醒用户需要进行抱闸检测，出厂默认每隔 30 天提醒。

当距离上次抱闸检测超过设定时间后，系统将会弹出如下对话框，提示用户需要执行抱闸检测。启动页面的抱闸检测入口将显示小红点提示用户。

在机械臂使能的情况下，启动页面会出现抱闸检测功能的入口。点击可进入抱闸检测页面，如图。该页面可显示抱闸检测周期和距离上次抱闸检测过去的天数。



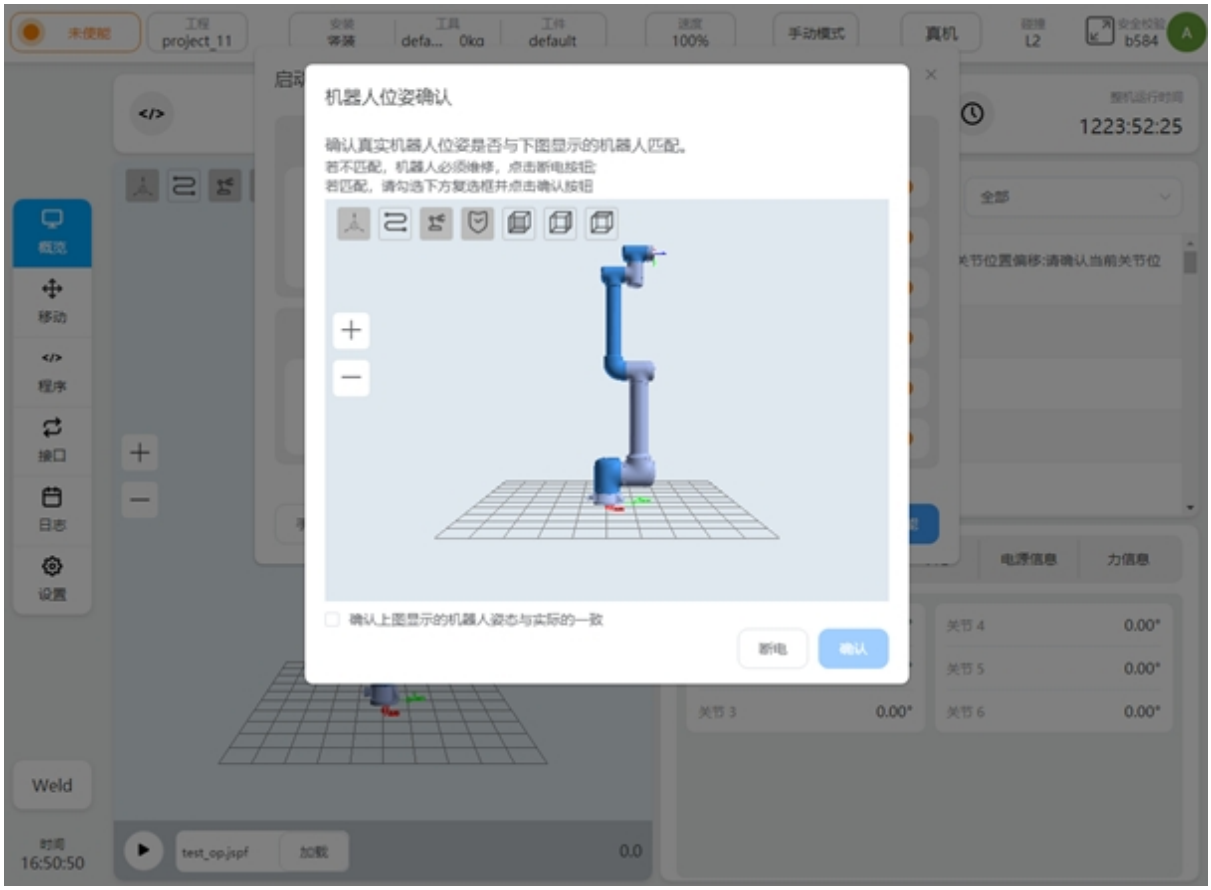
点击“开始检测”，弹出如下对话框，按照对话框提示进行操作。在检测的过程中可以暂停、停止检测过程。检测完成后，显示本次抱闸检测的结果。

点击“历史数据”可查看历次抱闸检测的时间及检测结果。

2.3.3 上电关节位置检测

机械臂在上电时，会检测关节是否在断电状态产生过异常移动，以保证使用安全。

若系统检测到至少一个机器人关节在下电状态下产生过异常移动，会弹出如下对话框。用户需要确认机器人实际位姿与界面 3D 模型是否一致，若一致，勾选“确认上图显示的机器人姿态与实际的一致”，点击“确认”按钮继续上电操作即可；若不一致，点击“断电”按钮断开机械臂动力电并联系维护人员。



2.4 安全设置

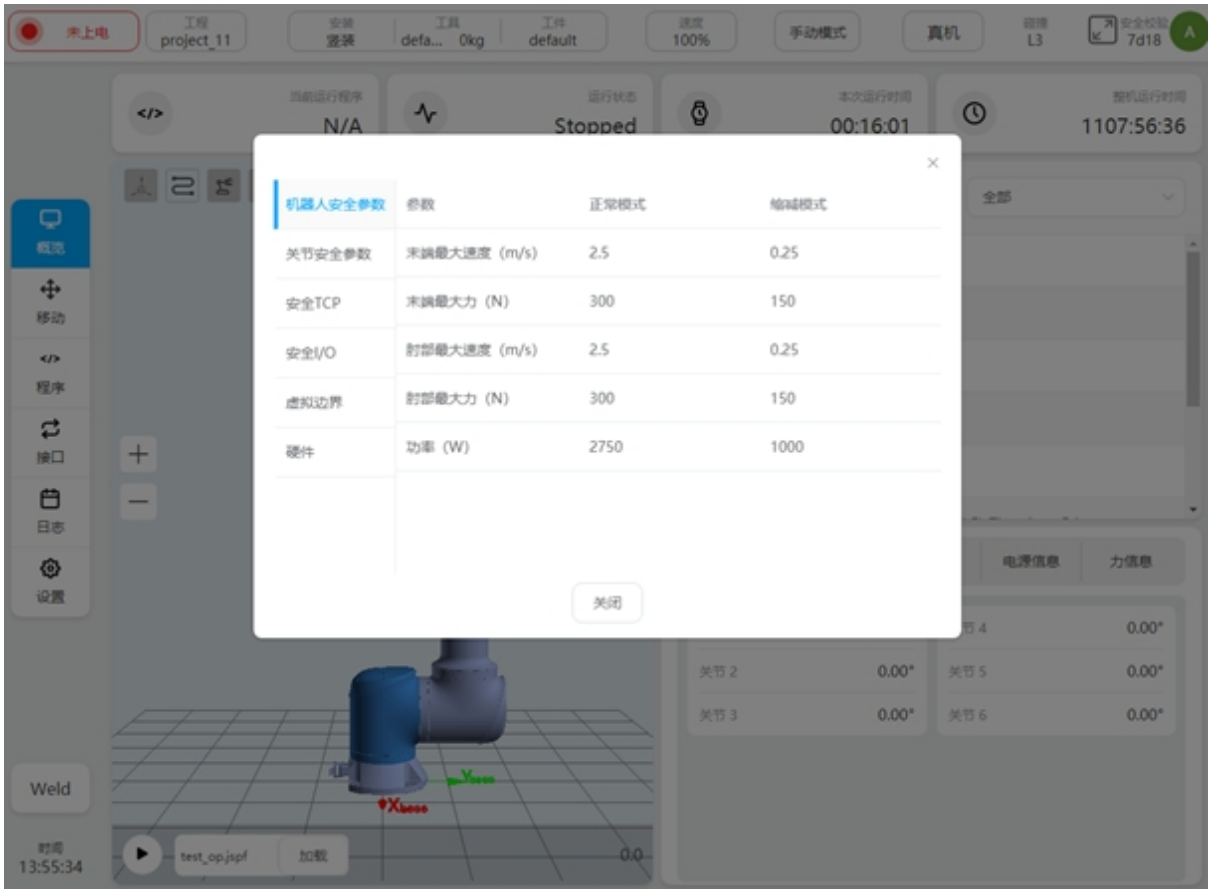
本小节介绍如何查看安全配置，更改应用安全配置。目前存在两种型号的安全控制器，DucoSafetyV1.0 适配 DC30 控制柜。DucoSafetyV2.0 适配 J9 及 J9-ZII 系列控制柜。

2.4.1 DucoSafety V1.0

查看安全配置

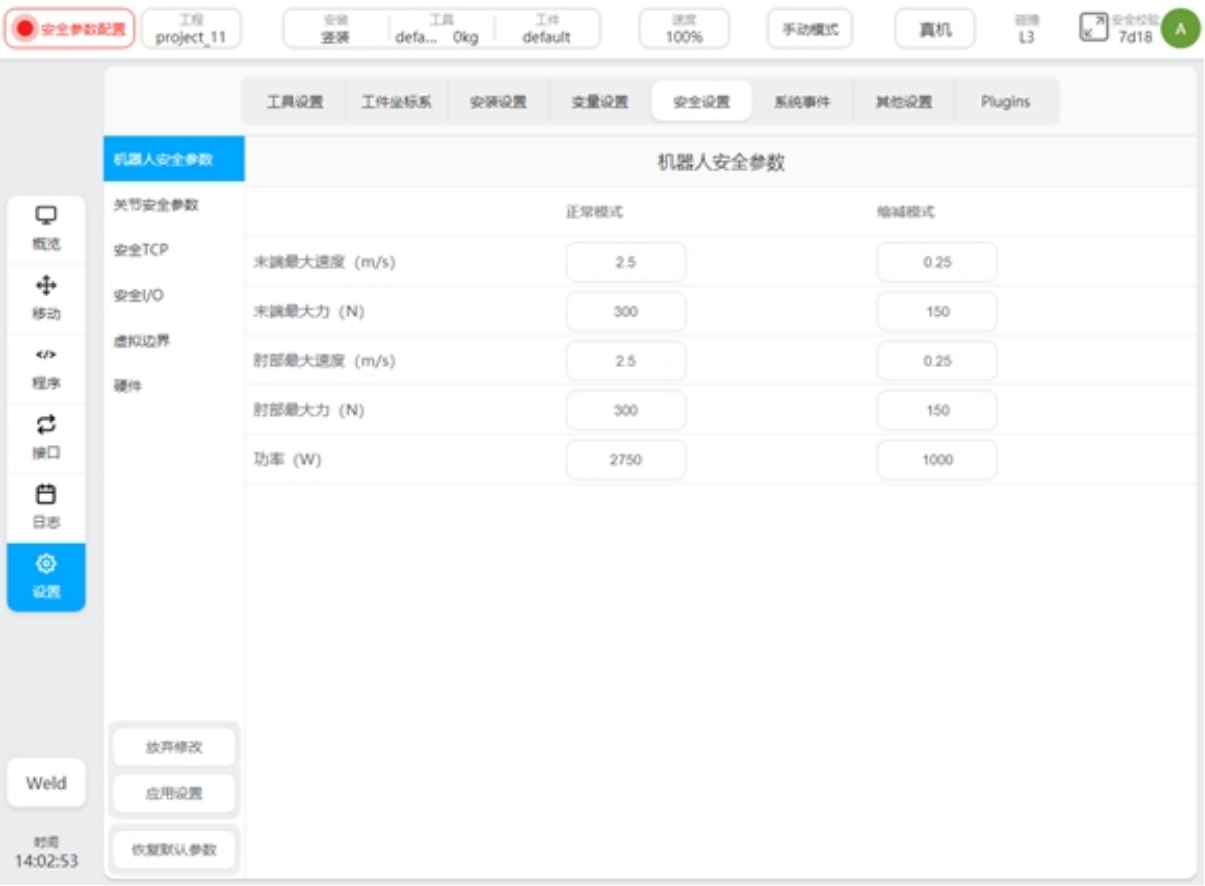
点击上方状态栏上的“安全校验”，弹出如下对话框，可查看当前激活的安全配置参数。

或者在设置页面——安全设置中也可以查看安全配置参数

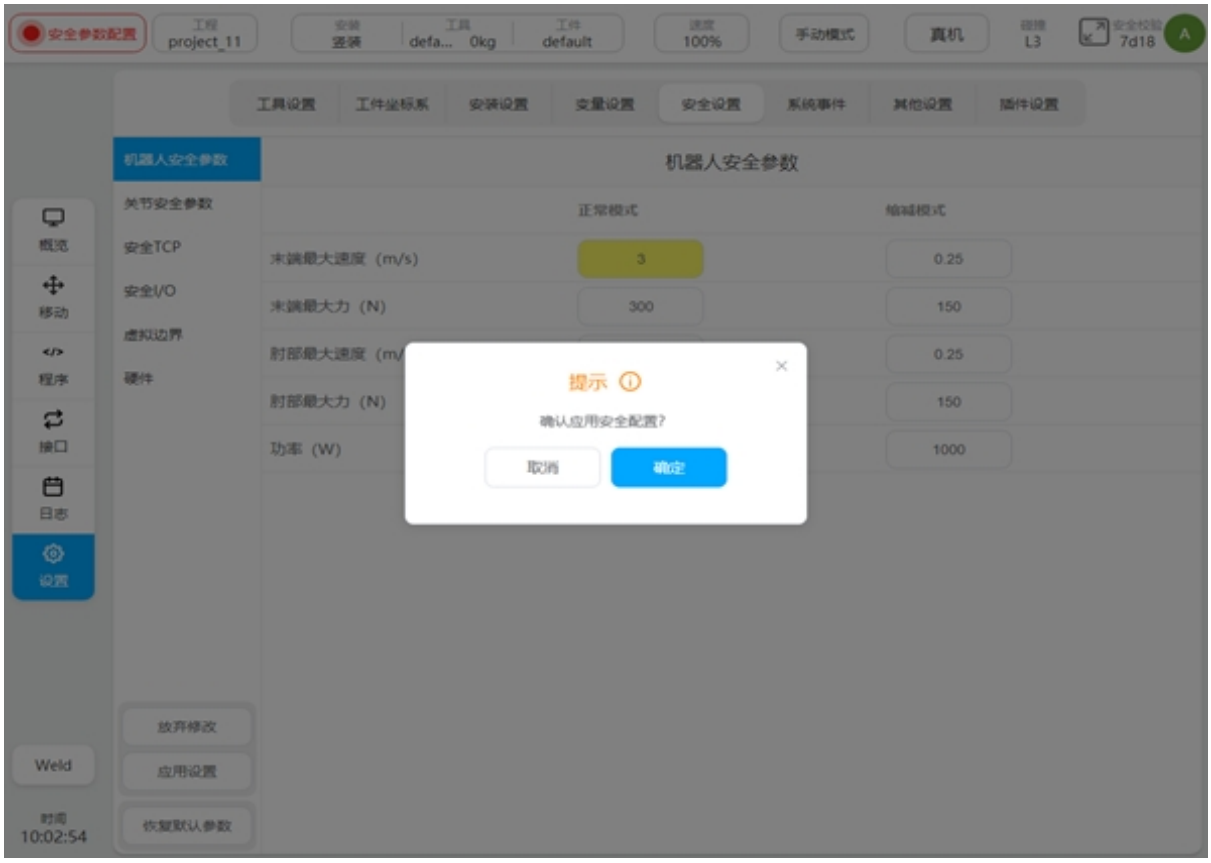


安全配置更改应用

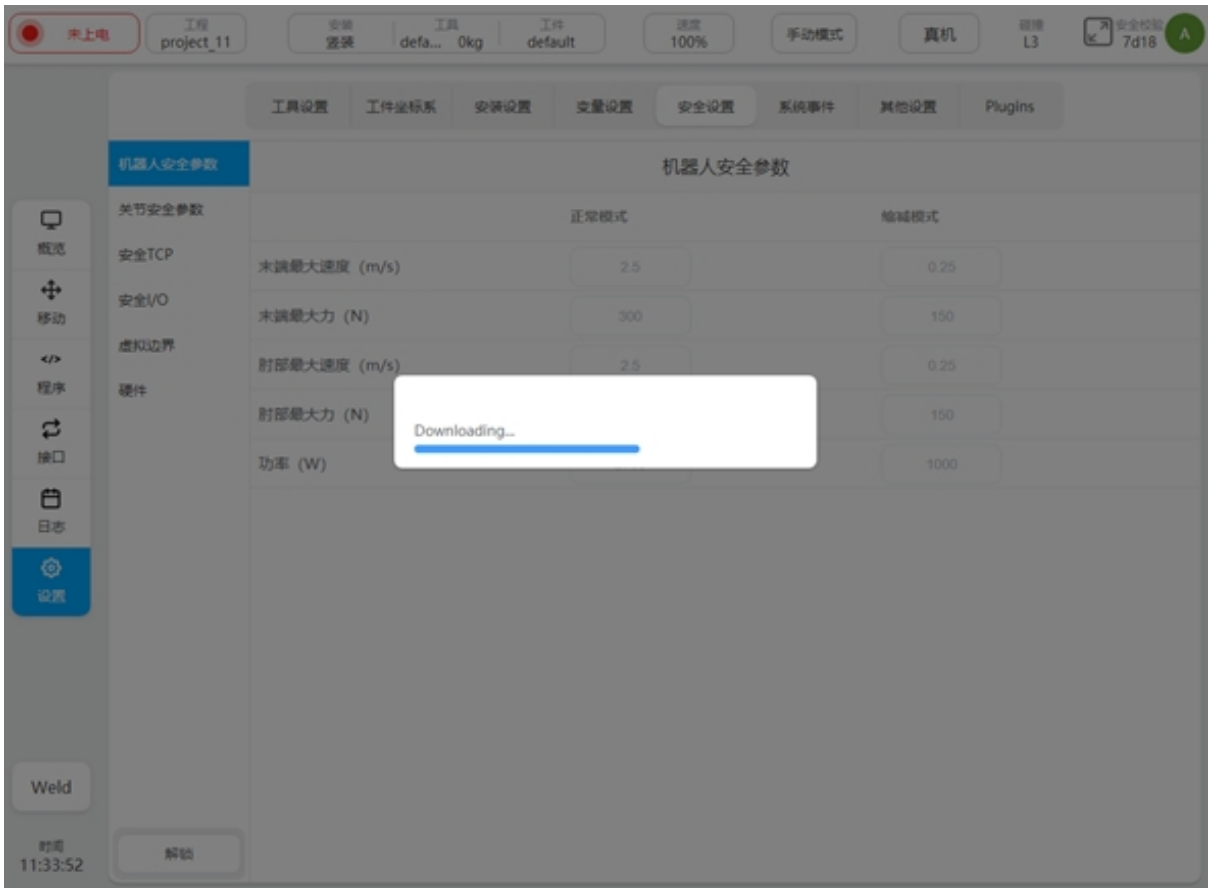
更改安全配置前必须使用密码解锁。进入设置页面——安全配置，点击左下方的“解锁”按钮，仅在机器人断电的情况下可以解锁，输入密码（当前登陆用户的登陆密码），验证通过后进入安全参数配置模式。此时状态栏上的状态显示区显示为“安全参数配置”。



更改配置时，本次的修改内容将标记为高亮黄色显示出来。所做的所有更改只有在点击“应用设置”按钮时才会生效。点击“恢复默认设置”，将会将所有的安全参数恢复到默认的出厂设置值。点击“应用设置”按钮后，会弹出提示框，如图所示。



单击“确认”按钮后，会显示加载安全参数过程的提示框，如图所示。



安全参数配置完成后，系统会将配置的安全参数再次以弹出窗口的方式显示，供用户进行检查。如下图，检查确认无误后。点击“确定”按钮进行安全参数的配置。配置成功后，状态栏上方的安全校验将发生更改。

安全参数说明

本小节将介绍机器人的各项安全配置参数。

安全模式

正常模式： 默认激活的安全模式

缩减模式： 可以使用安全输入 IO 激活缩减模式

恢复模式： 当机器人的实际运动参数超出关节位置或 Tcp 位置安全限制范围导致机器人触发 SS0 并停机时。当复位 SS0 并重新使机器人上电后，由于机器人仍然处于关节位置或 Tcp 位置违例的状态，此时恢复模式将激活。恢复模式下，用户仅可以通过关节 Jog 操作机械臂，将机器人移动到安全限制范围内后恢复模式讲过自动退出

机器人安全参数

机器人参数用来限制一般的机器人运动。可以配置其在正常模式和缩减模式下的参数值。

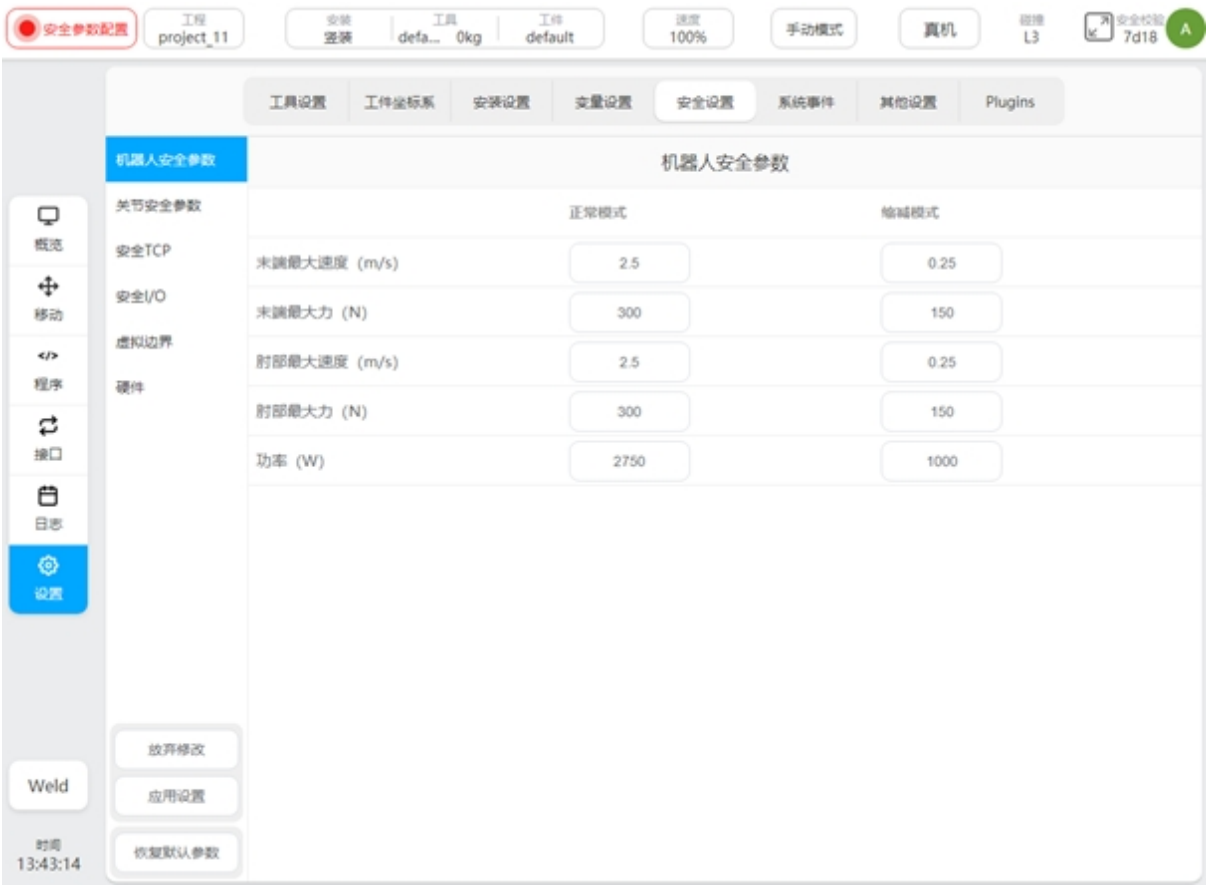
末端最大速度限制机器人末端 Tcp 的最大线速度

末端最大力限制机器人末端 Tcp 允许对外施加的最大力

肘部最大速度限制机器人肘部的最大线速度

肘部最大力限制机器人肘部对外施加的最大力

功率限制机器人产生的最大的机械功



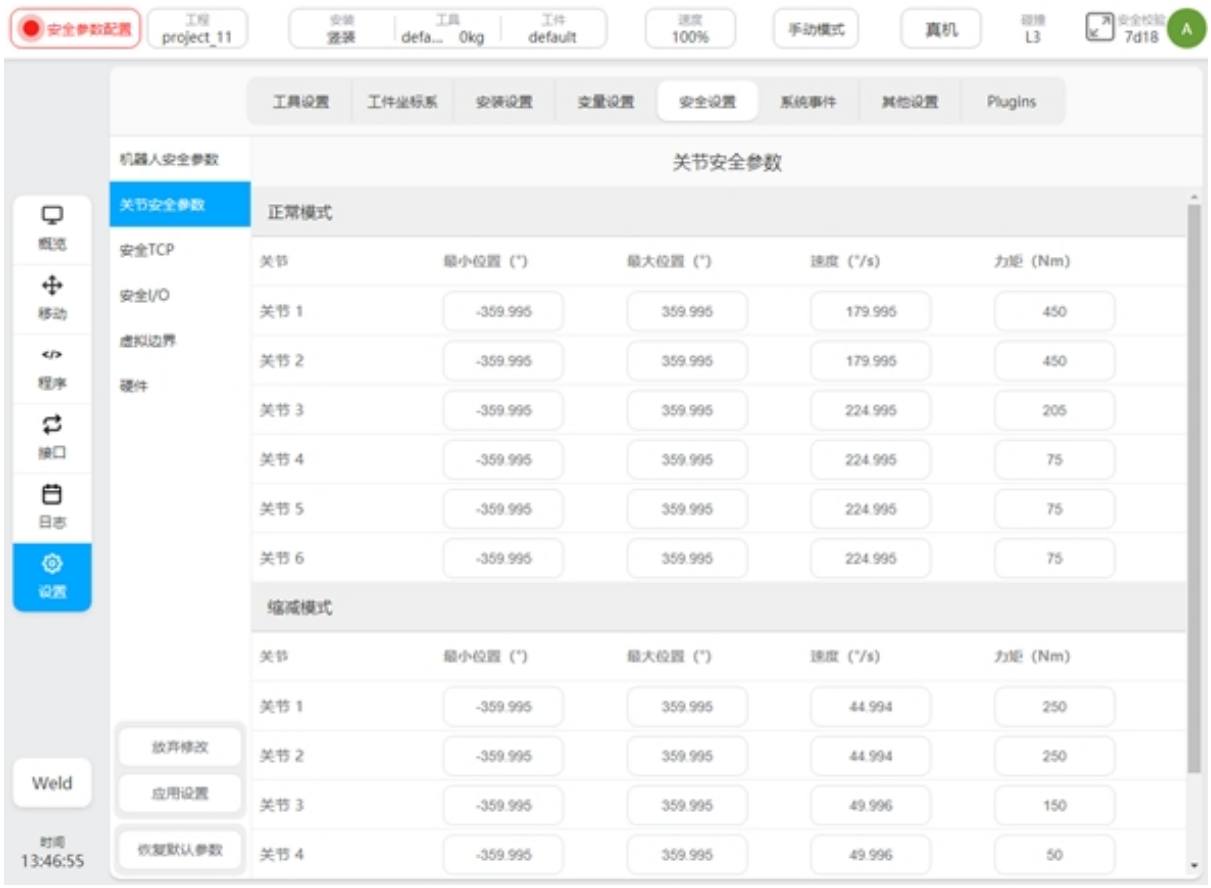
关节安全参数

关节参数限制用来限制机器人的每个关节的位置范围、最大速度、最大力矩。可以配置其在正常模式和缩减模式下的参数值

位置范围： 定义各个关节的最小位置和最大位置

最大速度： 定义各个关节的最大角速度

最大力矩： 定义各个关节最大的力矩

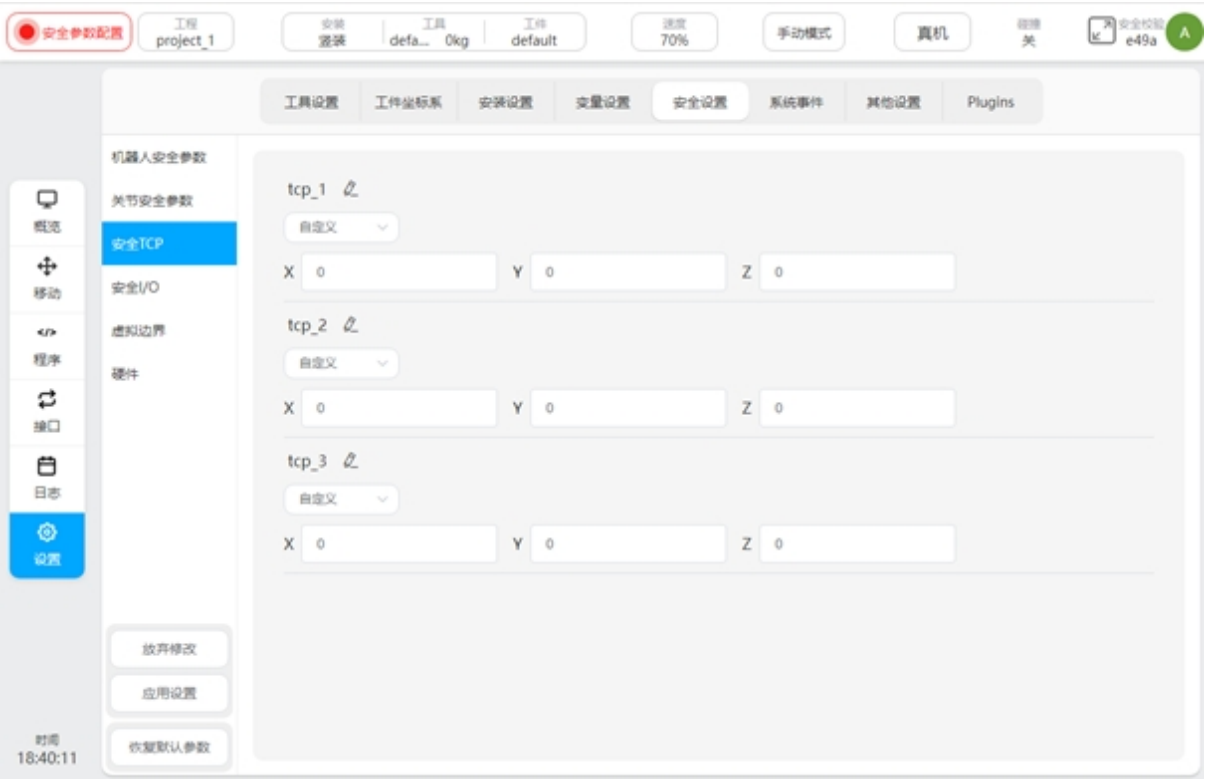


安全 TCP

安全系统可定义三组 TCP 偏移量，配置后，机器人将以这三组 TCP 来做速度监控和位置监控。任意一个安全 TCP 的位置、速度超过安全设置，均会触发安全违例。

速度监控阈值参考“机器人安全参数”中的工具最大速度。位置监控参考“虚拟边界”中所定义的机器人安全空间。在下拉框中可以选择现有的 TCP 或者自定义直接设定 XYZ 的值，当选择现有 TCP 时，更改 X、Y、Z 输入字段中的值时，下拉菜单中的 TCP 名称将变为自定义，表明是新定义的 TCP。

在完成了安全参数配置并应用后，再次更改系统中的工具坐标系 TCP 的设置信息，不会影响已经配置好的安全参数。



安全 IO

安全 IO 模块上包含两路可配置安全输入端口和两路可配置安全输出端口。

安全输入功能包含：

防护 reset 输入： 当防护停止发生时，触发该端口，机器人恢复正常状态

自动模式防护停止输入：机器人在自动模式下触发该端口时，机器人执行防护性停止

自动模式防护 reset 输入：当自动模式防护停止发生后，触发该端口，机器人恢复正常状态

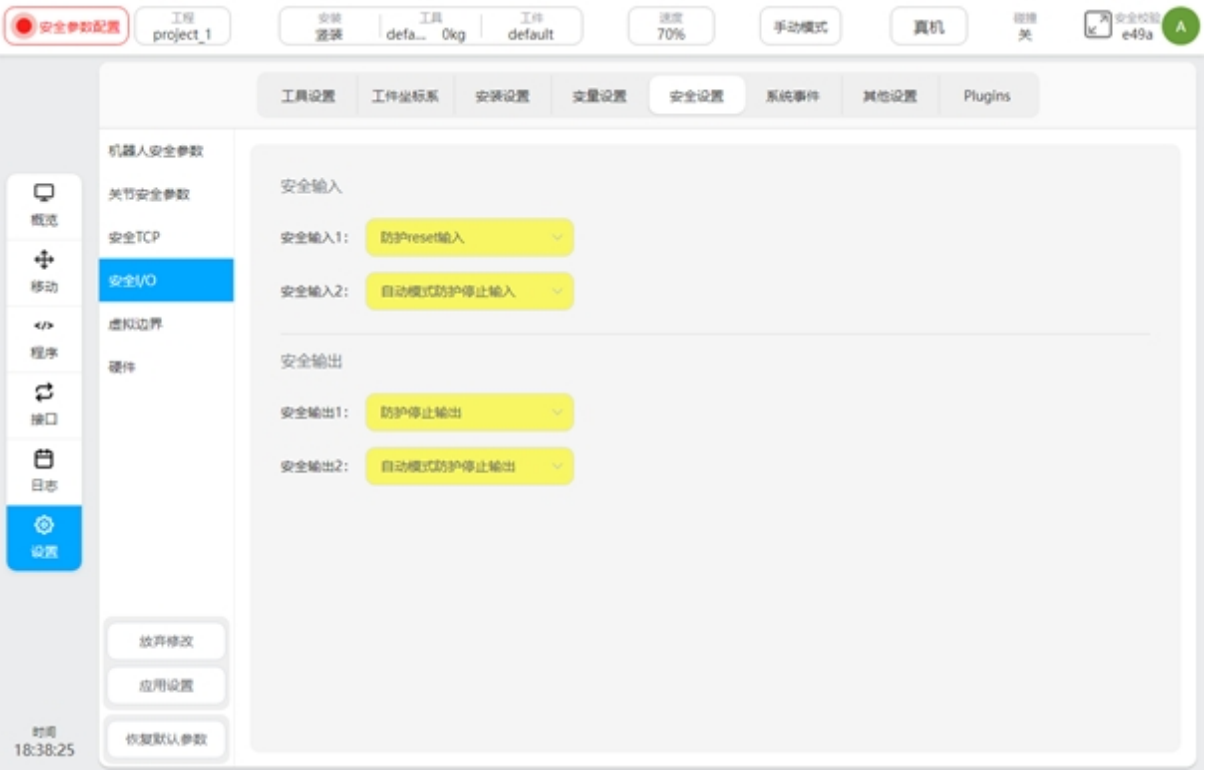
Reduce 模式输入：触发该端口，机器人将过渡到缩减模式，缩减模式对应的安全参数将激活。机器人可能会在过渡过程中减速以适应缩减模式所设置的安全参数

安全输出功能包含：

防护停止输出：当机器人处于防护停止状态时，触发该端口

自动模式防护停止输出：当机器人处于自动模式防护停止时，触发该端口

Reduce 模式输出：当机器人处于缩减模式时，触发该端口

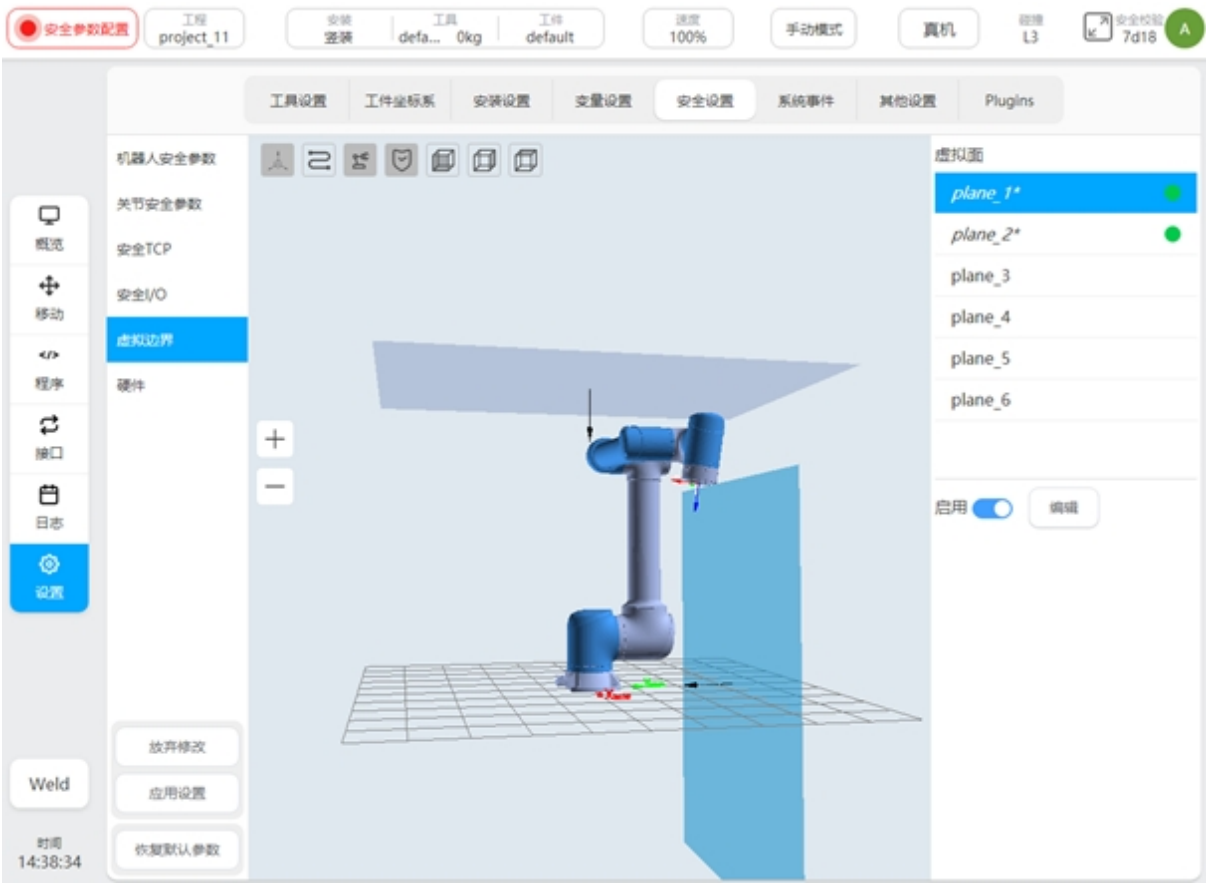


虚拟边界

虚拟边界用来定义机器人的安全工作空间。可定义六个虚拟边界来限制机器人的安全工具 Tcp 和肘部运动所能到达的空间位置。

当机器人的工具和肘部触碰到虚拟边界时，机器人将执行防护性停止。在此基础上，若机器人进一步产生移动使安全工具 Tcp 或肘部超出了虚拟边界所限制的安全工作空间，则会触发 SS0。

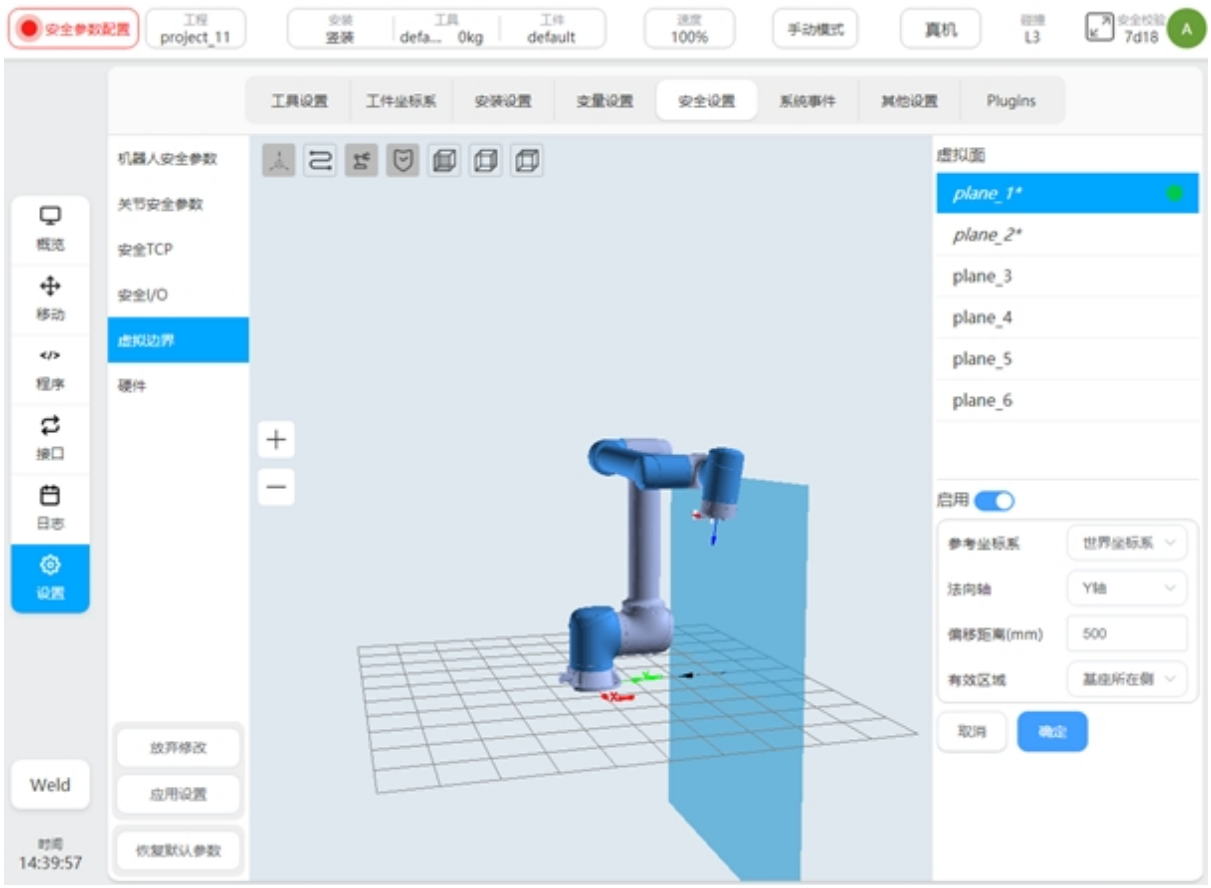
界面上右侧绿色指示灯表示该平面被激活，同时 3D 图显上可以显示激活的虚拟边界的位置和虚拟边界的有效区域。选中一个平面，3D 图显上对应的平面将高亮显示。



虚拟边界通过以下方式定义：

选择一个参考坐标系，可选择世界坐标系、基座坐标系或者设置中定义的工件坐标系；选择参考坐标系的一个坐标轴（X、Y、Z）作为虚拟平面的法向轴，设定沿着该坐标轴的偏移距离，偏移距离为正值时表示沿着坐标轴正向偏移，为负值时表示沿着坐标轴负向偏移。如此则确定了一个平面，然后选择机械臂的有效活动区域参考定义好的平面是否与机器人基座所在位置处于同一侧。例如选择参考坐标系为基座坐标系，选择 Z 轴为法向轴，偏移距离设定为 600mm，则虚拟平面为基座坐标系 XoY 平面向 Z 轴正向偏移 600mm 形成。

下图为相应的交互，选中一个平面，点击“启用”、“编辑”按钮，选择参考坐标系、法向轴、输入偏移距离、选择有效区域。点击“确定”按钮即可定义该虚拟边界，虚拟平面上显示的箭头表示机械臂的活动区域。

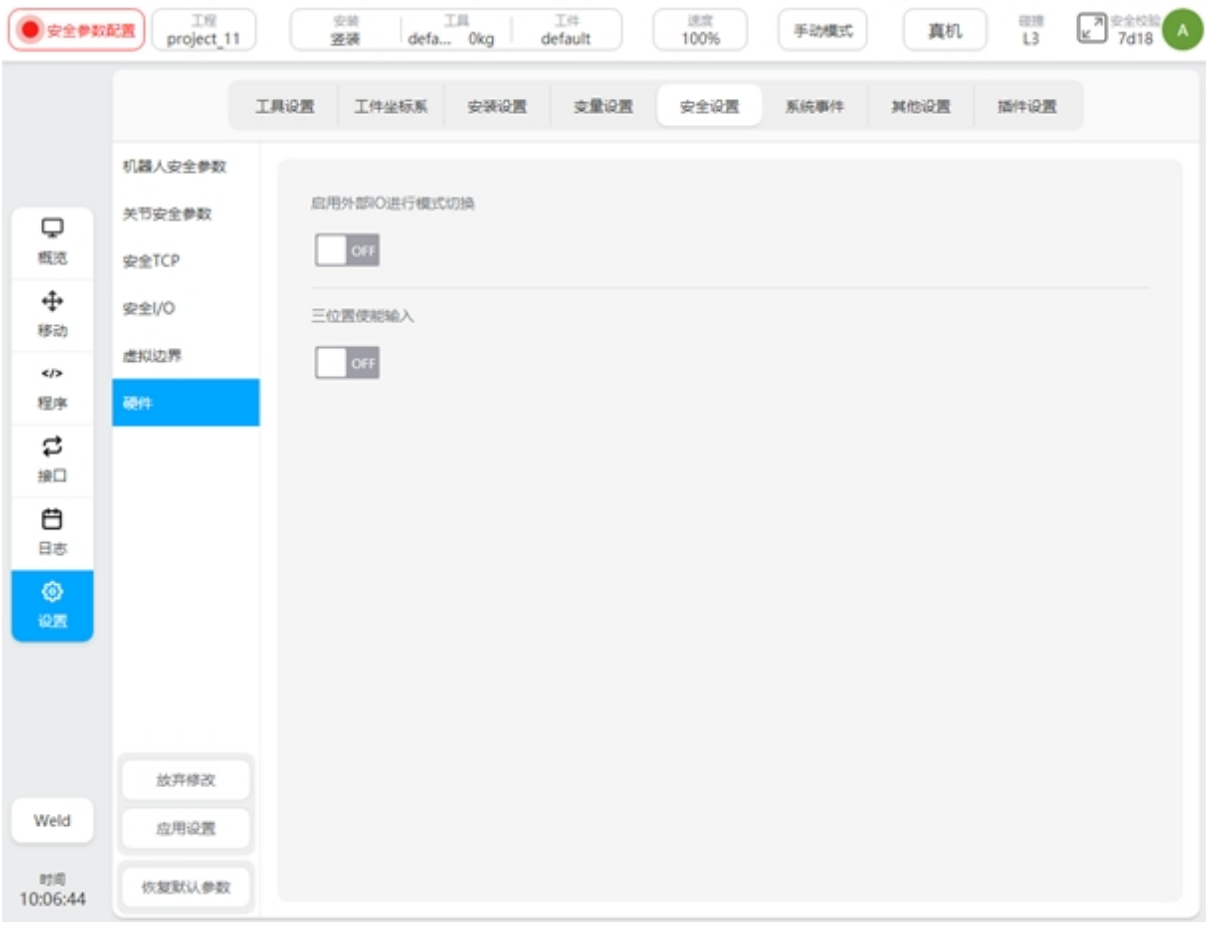


硬件

包含启用外部 IO 进行模式切换和三位置使能输入

启用外部 IO 进行模式切换： 启用此项，可以通过外部 IO 进行手自动模式的切换，此时界面状态栏上的模式切换功能无效。

三位置使能输入： 启用此项，当机器人处于手动模式下，仅当示教器上的三位置开关处于中间位置才可以移动机器人，机器人在移动过程中任意时刻三位置开关处于非中间位置时都会触发机器人的暂停。



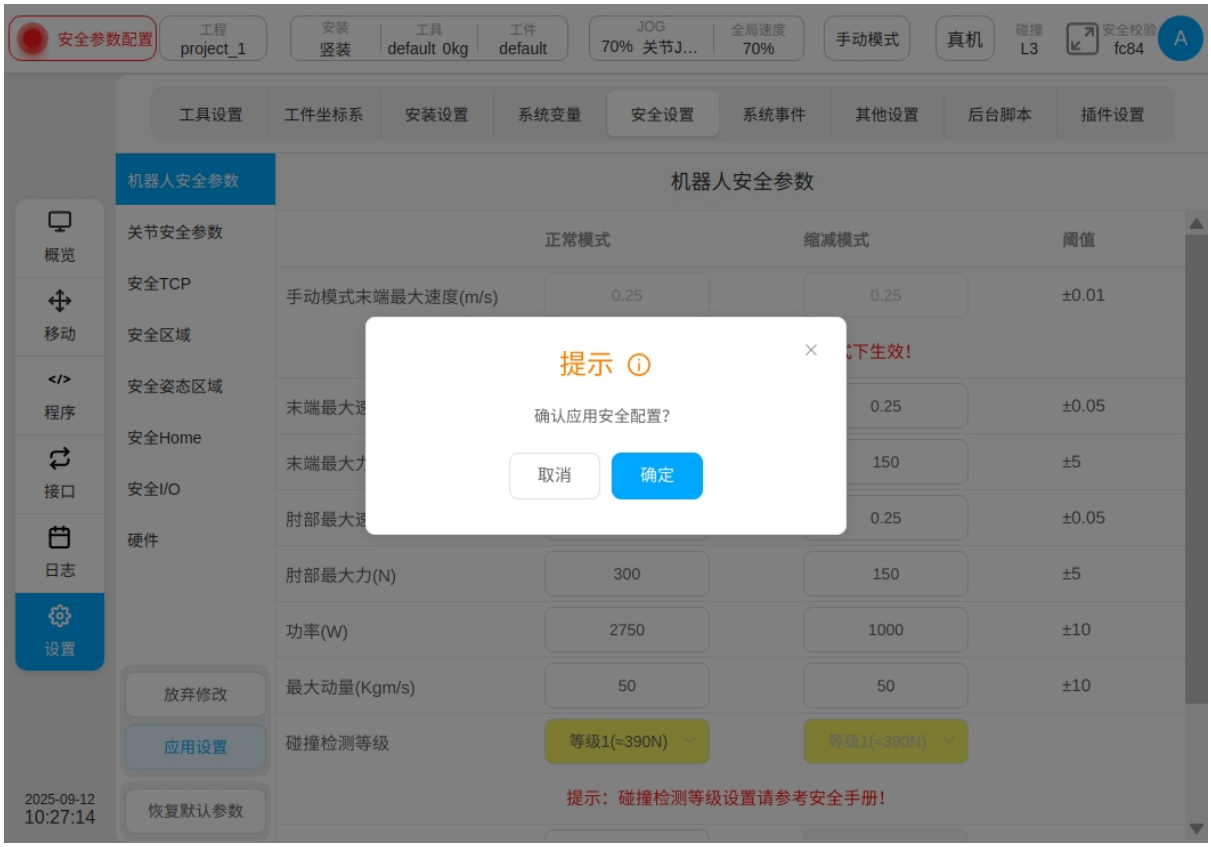
2.4.2 DucoSafety V2.0

查看安全配置

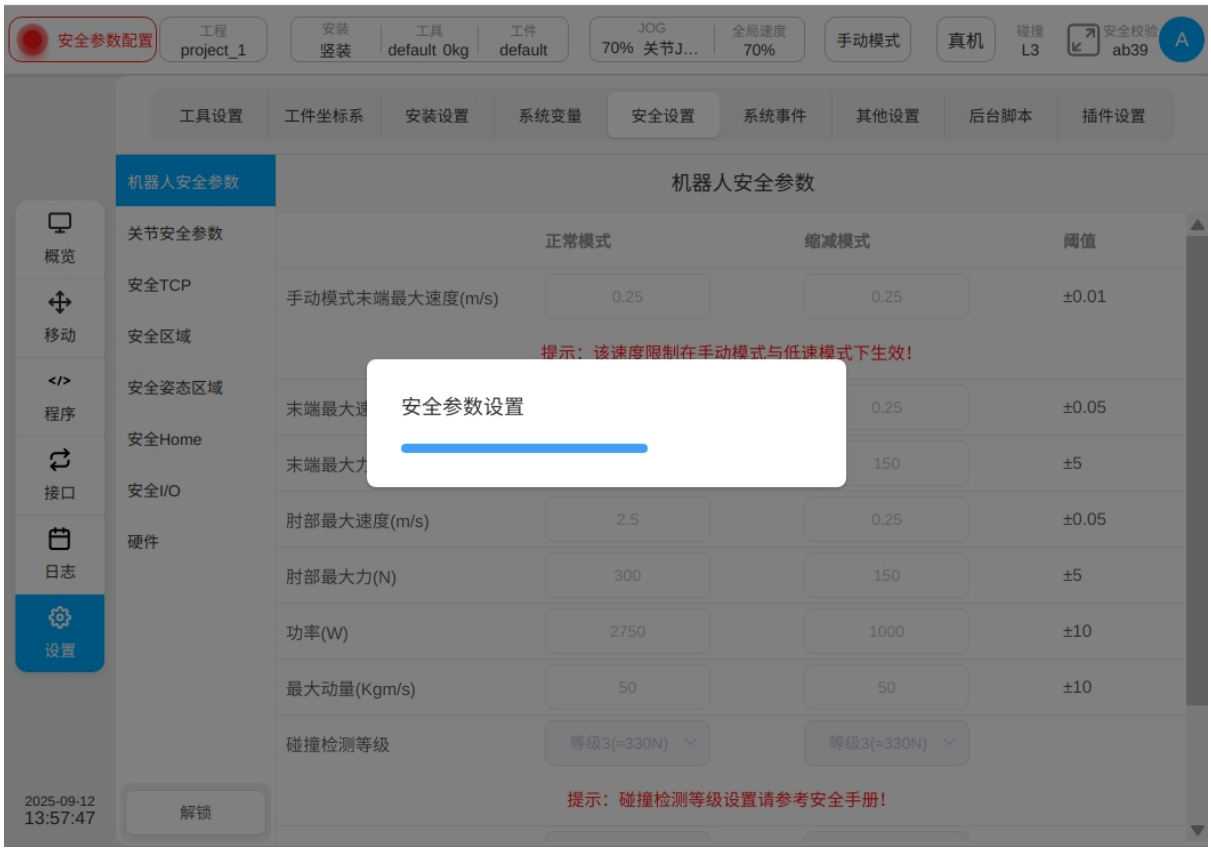
点击上方状态栏上的“安全校验”，弹出如下对话框，可查看当前激活的安全配置参数。
或者在设置页面——安全设置中也可以查看安全配置参数

安全配置更改应用

更改安全配置前必须使用密码解锁。进入设置页面——安全配置，点击左下方的“解锁”按钮，仅在机器人断电的情况下可以解锁，输入密码（当前登陆用户的登陆密码），验证通过后进入安全参数配置模式。此时状态栏上的状态显示区显示为“安全参数配置”。



单击“确认”按钮后，会显示加载安全参数过程的提示框，如图所示。



安全参数配置完成后，系统会将配置的安全参数再次以弹出窗口的方式显示，供用户进行检查。如下图，检查确认无误后。点击“确定”按钮进行安全参数的配置。配置成功后，状态栏上方的安全校验将发生更改。

安全参数说明

本小节将介绍机器人的各项安全配置参数。

安全模式

正常模式： 默认激活的安全模式

缩减模式： 可以使用安全输入 IO 激活缩减模式

恢复模式： 当机器人的实际运动参数超出关节位置或 Tcp 位置安全限制范围导致机器人触发 SS0 并停机时。当复位 SS0 并重新使机器人上电后，由于机器人仍然处于关节位置或 Tcp 位置违例的状态，此时恢复模式将激活。恢复模式下，用户仅可以通过关节 Jog 操作机械臂，将机器人移动到安全限制范围内后恢复模式讲过自动退出

机器人安全参数

机器人参数用来限制一般的机器人运动。可以配置其在正常模式和缩减模式下的参数值。

手动模式末端最大速度限制手动模式（低速模式）机器人末端 Tcp 的最大线速度，不可修改，可通过功能输入配置激活，功能输出反馈激活状态

末端最大速度限制机器人末端 Tcp 的最大线速度

末端最大力限制机器人末端 Tcp 允许对外施加的最大力

肘部最大速度限制机器人肘部的最大线速度

肘部最大力限制机器人肘部对外施加的最大力

功率限制机器人产生的最大机械功

最大动量限制机器人产生的最大动量，机器人末端负载视为机器人本体的一部分

碰撞检测等级 机器人检测与外界发生碰撞的灵敏度，等级越高灵敏度越高

碰撞复位模式 机器人发生碰撞后复位的方式

碰撞响应模式 机器人发生碰撞时响应方式，其中“零力回弹”响应模式会在机器人检测到碰撞后响应碰撞力并产生回弹移动。

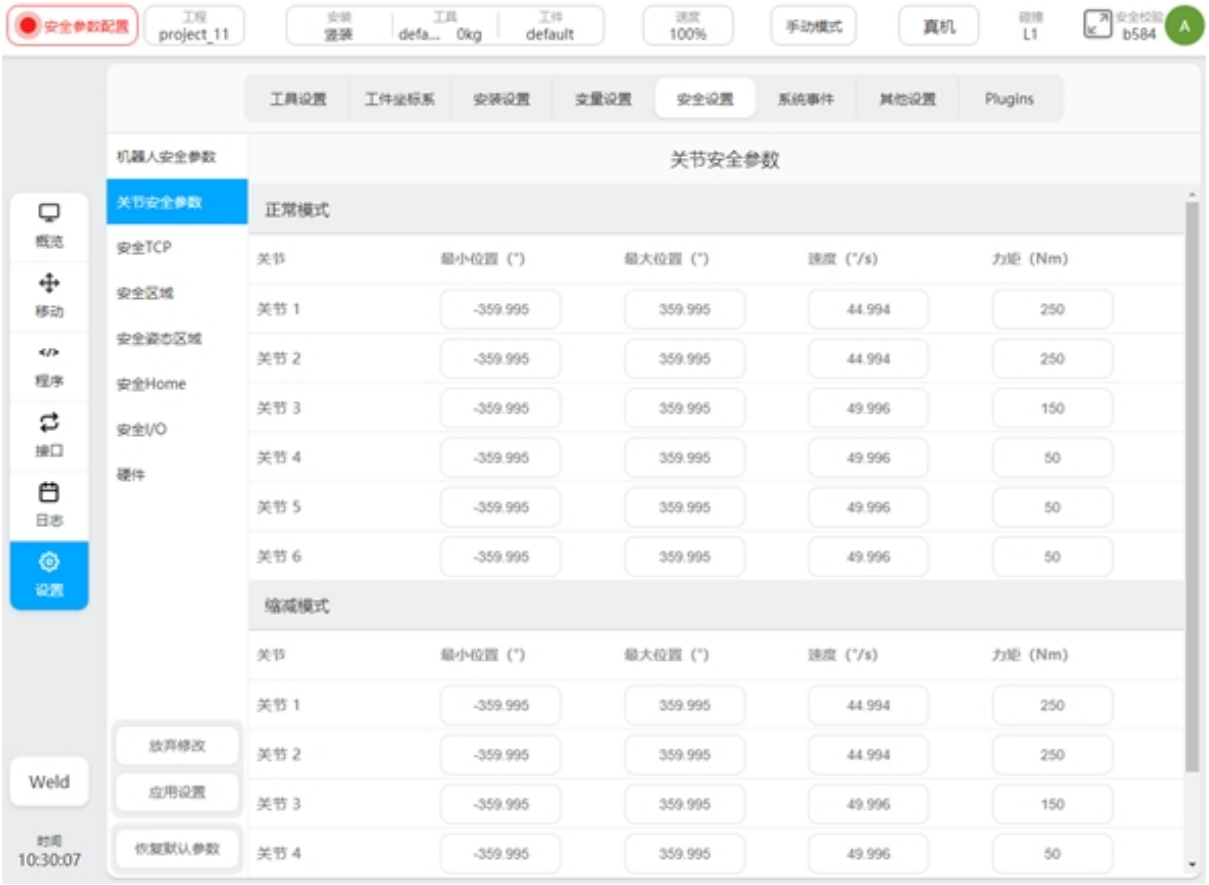
关节安全参数

关节参数限制用来限制机器人的每个关节的位置范围、最大速度、最大力矩。可以配置其在正常模式和缩减模式下的参数值

位置范围： 定义各个关节的最小位置和最大位置

最大速度： 定义各个关节的最大角速度

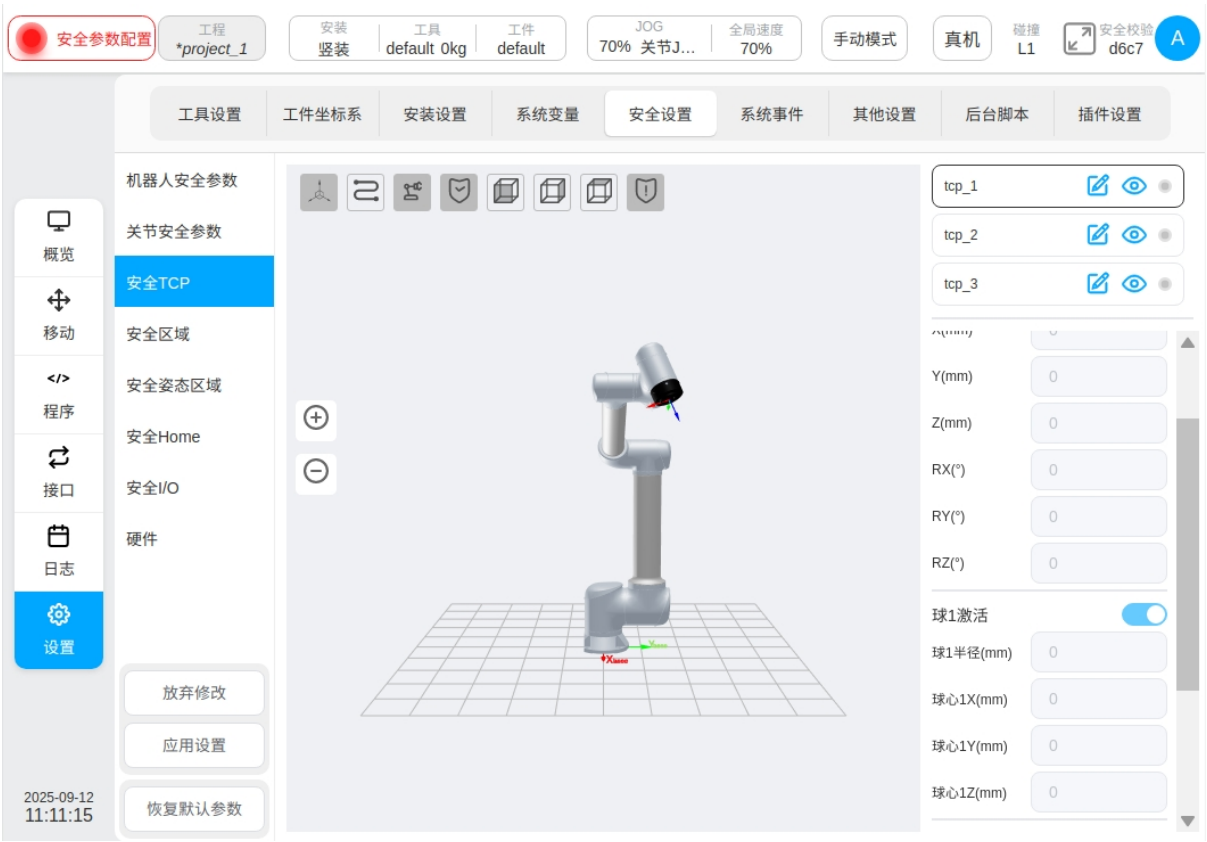
最大力矩： 定义各个关节最大的力矩



安全 TCP 及工具

安全系统可定义三组 TCP 偏移量，配置后，机器人将以这三组 TCP 来做速度监控和位置监控。任意一个安全 TCP 的位置、速度超过安全设置，均会触发安全违例。

速度监控阈值参考“机器人安全参数”中的工具最大速度。位置监控参考“虚拟边界”中所定义的机器人安全空间。在下拉框中可以选择现有的 TCP 或者自定义直接设定 XYZ 的值，当选择现有 TCP 时，更改 X、Y、Z 输入字段中的值时，下拉菜单中的 TCP 名称将变为自定义，表明是新定义的 TCP。如果全局变量中对应坐标系中的值被修改，和当前安全设置中用到的坐标系的值不一致，会在工具定义处显示"!" 提醒图标。当选择自定义输入数据，直接编辑 X、Y、Z、Rx、Ry、Rz 的值以及可分别设置两个包络球的球半径、球心距离 TCP 的 X、Y、Z 方向偏移量。



每个安全 TCP 可以被配置为五类激活模式：禁用、一直有效、自动模式有效、安全组合配置 1、安全组合配置 2。当三个 TCP 均是禁用状态时，安全控制系统默认使用法兰坐标系。在 TCP 配置列表上，显示对应 TCP 的配置状态，如果某 TCP 被禁用，则对应状态灰显。配置的安全 TCP 可以在 3D 显示区中显示坐标系以及包络球，且可以通过对应 TCP 名称后的 图标进行显示和隐藏的切换。点击 TCP 名称后的 图标可以修改 TCP 默认名称。未被禁用的安全 TCP 会显示 ，否则会显示 。

安全区域

安全区域用来定义机器人的安全工作空间。共可定义六个安全区域来限制机器人的安全工具 Tcp 和肘部运动所能到达的空间位置或在安全空间中的运动状态。

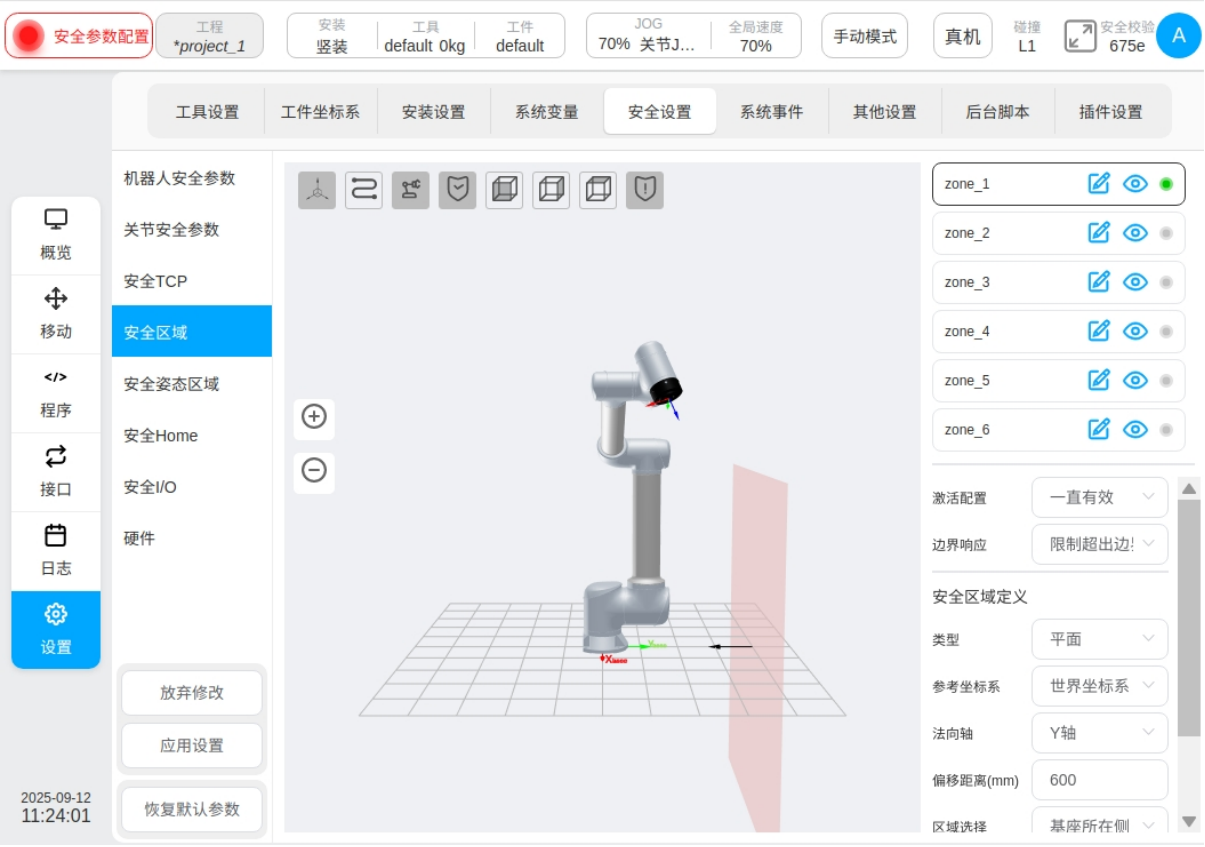
安全区域定义类型包括平面、长方形、圆柱体三种形式。用户可以最多设置 6 个相互独立的空间区域。通过点击安全区域名称后的 图标可以对配置的安全区域在 3D 显示区中进行显示和隐藏的切换，且可以点击区域名称（如：默认名称 zone_1）后的 图标修改安全区域名称。未被禁用的安全区域会显示 ，否则会显示 。

安全区域可以被配置为五类激活模式：禁用、一直有效、自动模式有效、安全组合配置 1、安全组合配置 2，共 5 种。

安全区域的边界响应是指机器人安全工具 Tcp 包络球从安全区域内部运动至外侧或从外侧运动至内部时所产生的响应动作，共有 2 种响应模式，即：限制超出边界、进入区域触发 reduce 模式。当选择限制超出边界时，安全区域所定义的安全工作空间被定义为机器人唯一允许运动作业的空间。任何使机器人安全工具 Tcp 向安全区域外侧运动的操作，在接近安全区域边界时都会触发防护性停止。若进一步使机器人安全工具 Tcp 运动并超出安全区域边界，

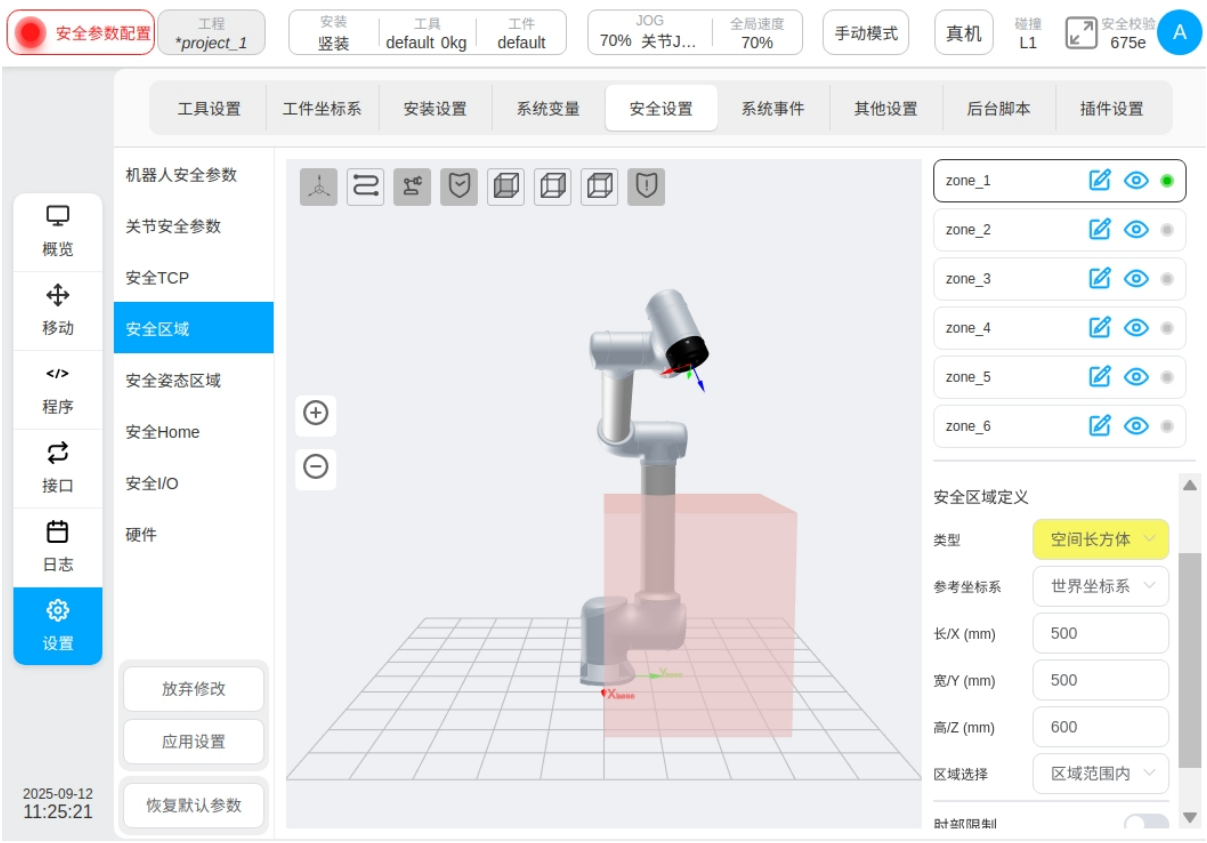
则会触发 SS0；当选择触发 reduce 模式时，安全区域所定义的安全工作空间则被定义为需要以缩减模式安全参数进行安全监控的限制下进行运动。机器人安全工具 Tcp 离开安全区域内部后，安全模式会自动切换回正常模式，直到安全工具 Tcp 再次回到安全区域内部。

空间区域中还可以设置是否包含肘部限制，肘部的空间范围以球半径形式进行设置。若激活肘部限制，则面向安全工具 Tcp 的所有安全区域相关限制同样适用于肘部。

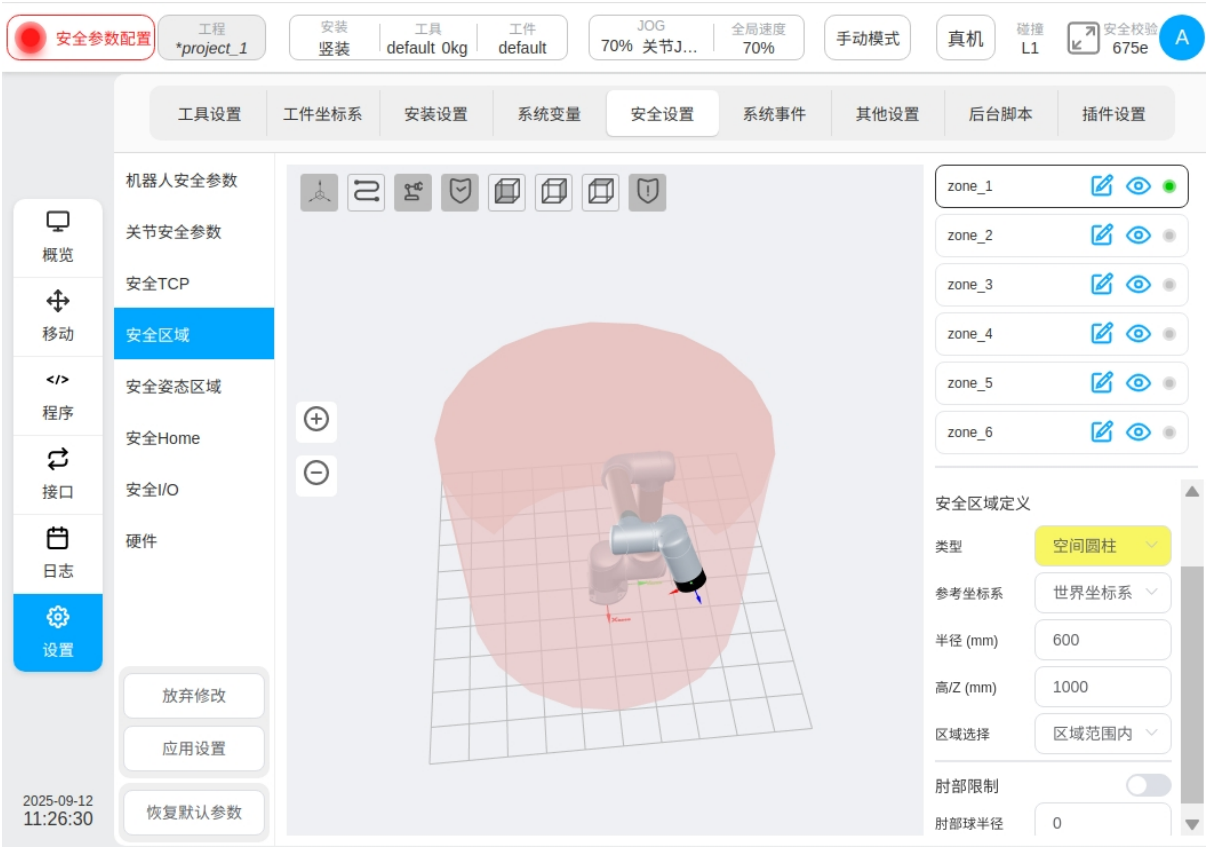


当配置区域类型为平面时，平面区域的参考基准是基于世界坐标系/基座/设定的工件坐标系，通过设定平面法向、法向偏移量从而确定平面。平面的设置可以通过“区域选择”配置生效区域，且对应的方向在 3D 区域中通过黑色箭头显示。

当配置区域类型为空间长方体时，长方体区域的设置基准是基于世界坐标系/基座/设定的工件坐标系，以工件坐标系作为长方体的一个角点，三个坐标轴方向分别对应长（X）、宽（Y）、高（Z）。长宽高的设置范围为-3000mm—3000mm。



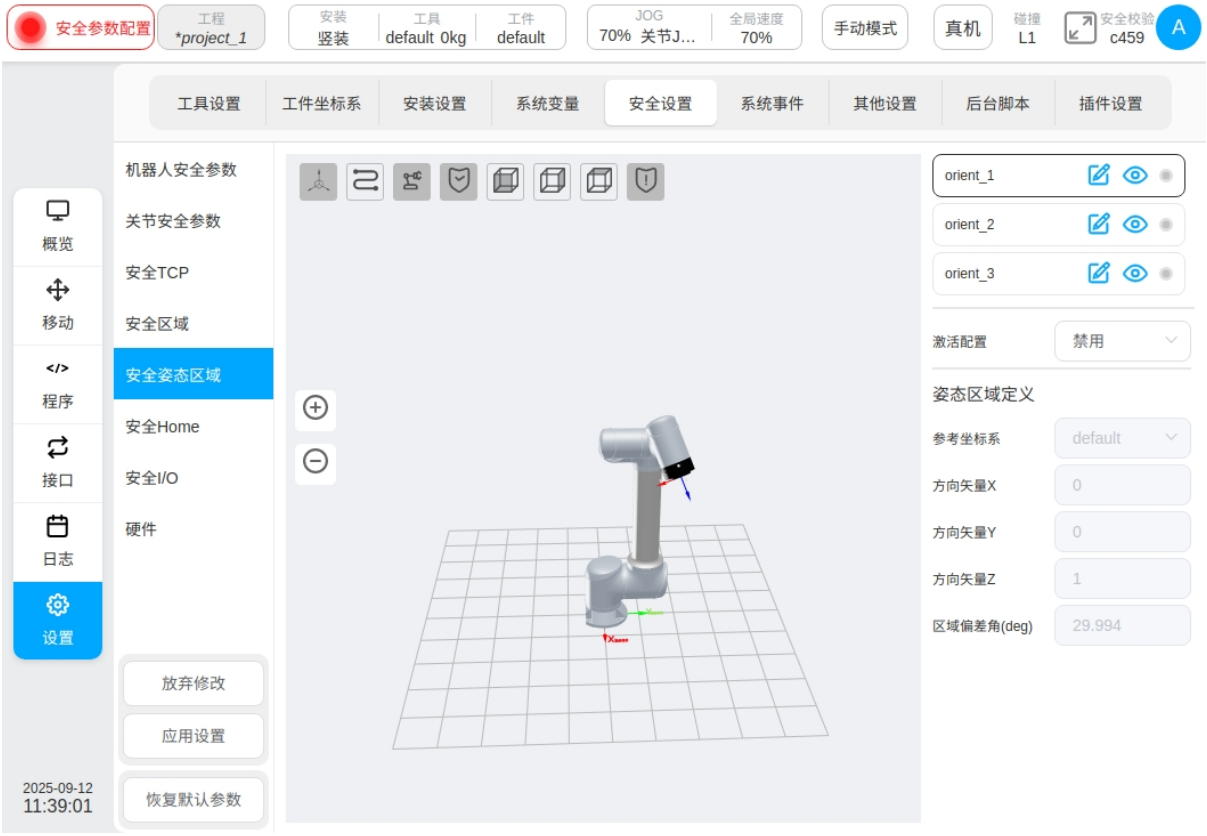
当配置区域类型为空间圆柱体时，其设置基准是基于世界坐标系/基座/设定的工件坐标系，以工件坐标系作为圆平面的圆心，z 方向指向高度方向，可以设置半径、高度。半径取值范围为0-3000mm，高度取值范围为-3000mm—3000mm。



除平面安全区域外，空间长方体和空间圆柱体的区域选择参数可选项为“区域范围内”和“区域范围外”，参数默认为“区域范围内”。当用户选择“区域范围内”作为安全区域选择类型时，用户所定义空间几何体内部为安全区域；相反地，用户选择“区域范围外”作为安全区域选择类型时，用户所定义空间几何体外部为安全区域。

安全姿态区域

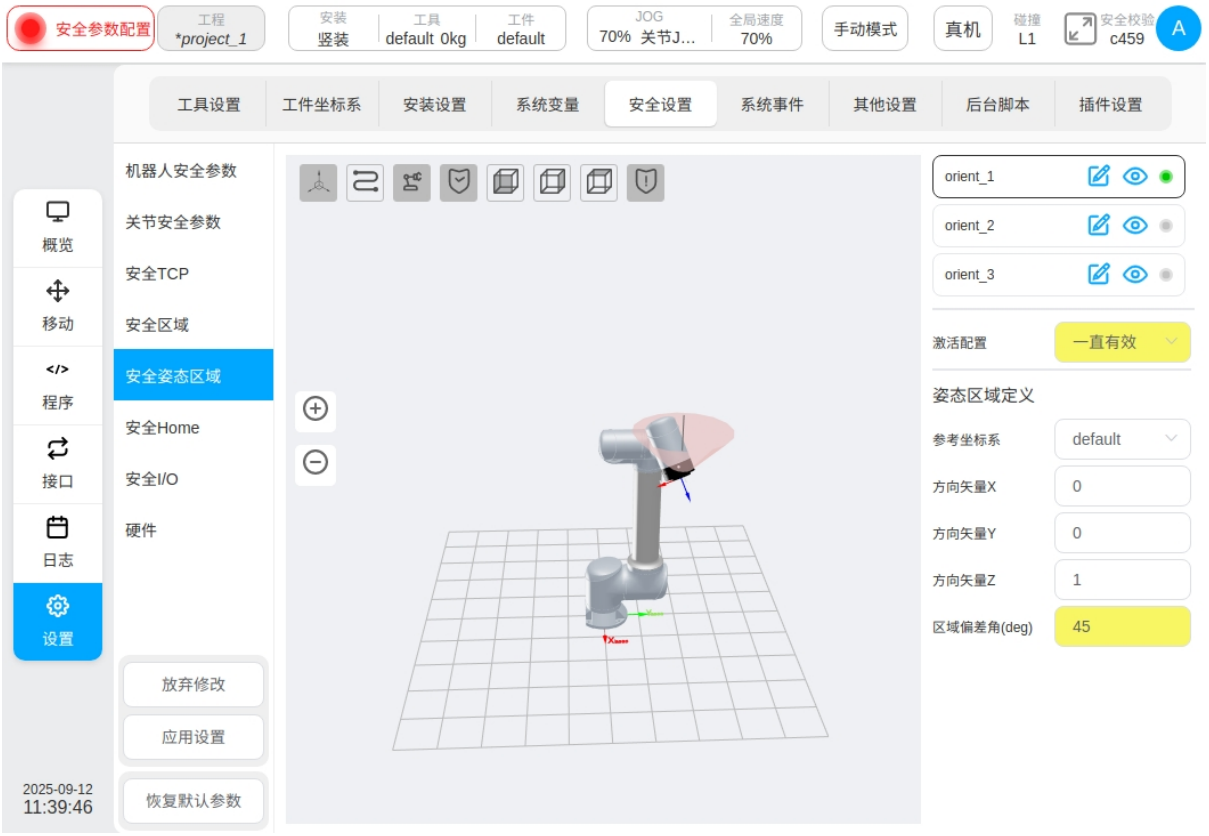
安全姿态区域指在机器人基坐标系下，围绕某个方向矢量形成的一个圆锥角。安全姿态区域限制了机器人安全工具 TCP 的 Z 轴在空间内的姿态允许活动区域。用户最多可以设置三个姿态区域。通过点击姿态区域右侧名称后的 图标可以显示或隐藏单个姿态区域，默认为显示。且可以点击名称后的 图标修改姿态区域名称。未被禁用的姿态区域会显示 "，否则会显示 。



安全姿态区域的激活配置有：禁用、一直有效、自动模式有效、安全组合配置 1、安全组合配置 2，共 5 种方式。安全姿态区域不同于安全区域，只约束 TCP 的 Z 轴方向。任何使安全工具 TCP Z 轴向安全姿态区域外侧旋转的运动，在接近安全姿态区域边界时会触发防护性停止。若进一步运动使安全工具 TCP Z 轴超出安全姿态区域范围，则会触发 SS0。

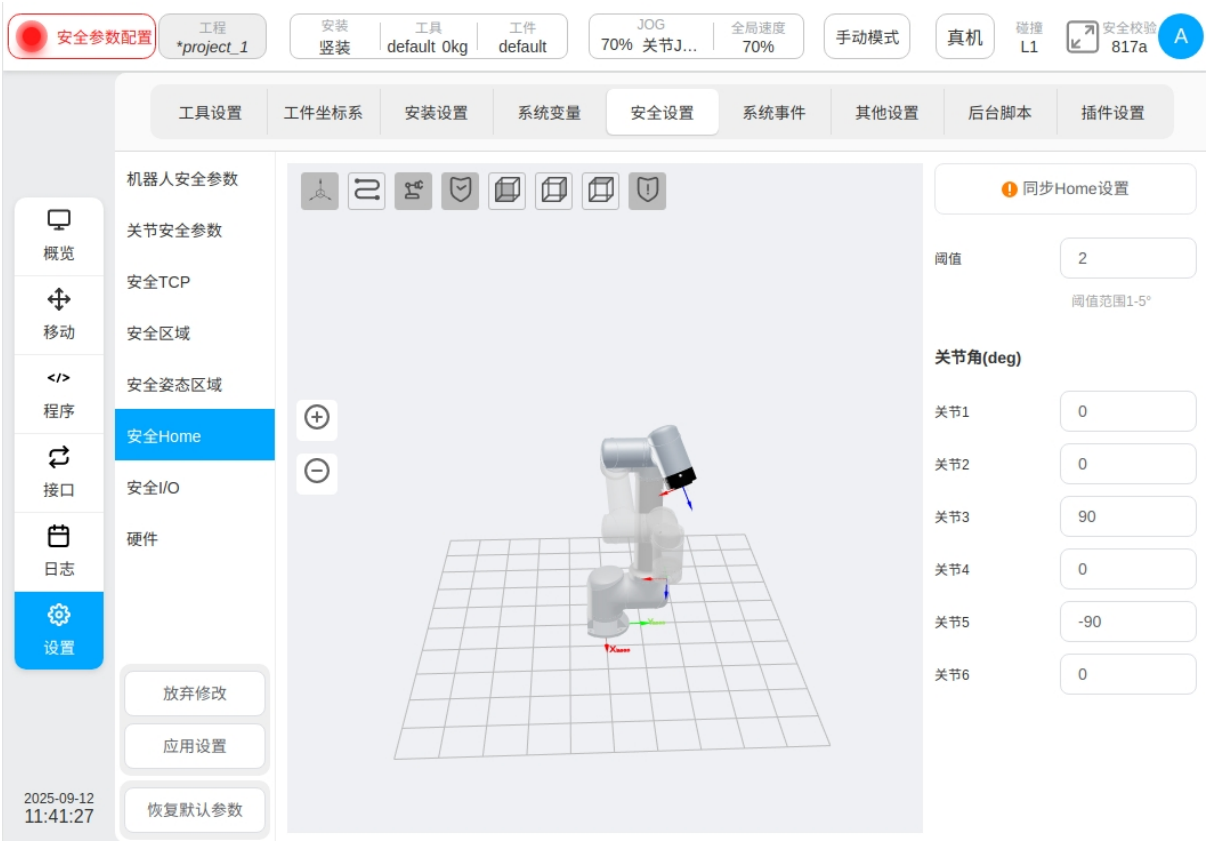
安全姿态区域定义的参考坐标系有 3 种方式：自定义、获取当前 TCP 姿态、通过预先定义的工件坐标系定义。当选择自定义时，手动修改方向矢量 X、Y、Z 的值，该值描述的是在机器人基坐标系下的值；当选择获取当前 TCP 姿态时，以当前 TCP 的 Z 轴方向作为姿态区域的基准方向，且将方向转换到机器人基坐标系下描述，显示到矢量方向 X、矢量方向 Y、矢量方向 Z 上，若手动修改值后，会变成自定义方式.；当通过工件坐标系定义时，以工件坐标系 Z 轴方向作为姿态区域的基准方向，转换到机器人基坐标系下，同样地手动修改值后，会变成自定义方式。

如果在外部对工件坐标系方向做了修改，在参考坐标系处显示提醒 提醒图标。区域偏差角取值 5-180 度。例如：



安全 Home

安全 Home 设置，安全 Home 监控将同步“其他设置”页面中的 Home 位置设置。如果安全 Home 设置中的位置和“其他设置”中的 Home 位置不一致，在“同步 Home 设置”处显示提醒 提醒图标。若不同步设置，系统的 Home 位置以安全设置中为准。移动界面中“按主归位”回 Home 点操作，及机器人其他端口（如 2001 端口）Home 信号的输出，均以安全控制器的设置为准。“其他设置”页面中的 Home 点仅作为记录，不作为 Home 点判断依据。可设置 Home 点的监控阈值，范围 1-5°。



当选择同步 Home 位置后，显示 Home 位置的关节角度。且 3D 模型显示区机器人模型刷新到对应位置。

当设置了安全 home 位置，且在安全 I/O 中，配置了 home 位置输出（详见第 6.2.3.8 章节），则当机器人所有的关节角度，在所设定的 home 位置范围内（设定值-阈值，设定值 + 阈值）并保持 500 毫秒（系统预定义且无法配置），对应的安全 I/O 通道会输出安全 home 信号。

安全 IO

安全 IO 模块上包含两路可配置安全输入端口和两路可配置安全输出端口。

安全输入功能包含：

防护 reset 输入： 当防护停止发生时，触发该端口，机器人恢复正常状态

自动模式防护停止输入：机器人在自动模式下触发该端口时，机器人执行防护性停止

自动模式防护 reset 输入：当自动模式防护停止发生后，触发该端口，机器人恢复正常状态

Reduce 模式输入：触发该端口，机器人将过渡到缩减模式，缩减模式对应的安全参数将激活。机器人可能会在过渡过程中减速以适应缩减模式所设置的安全参数

安全组合配置 1 / 2：当安全组合配置 1 或者安全组合配置 2 端口被触发时，所有被配置属于安全组合配置 1 或者安全组合配置 2 的安全特征，包括安全工具、安全区域、安全姿态区域将会被激活并开始监控

安全输出功能包含：

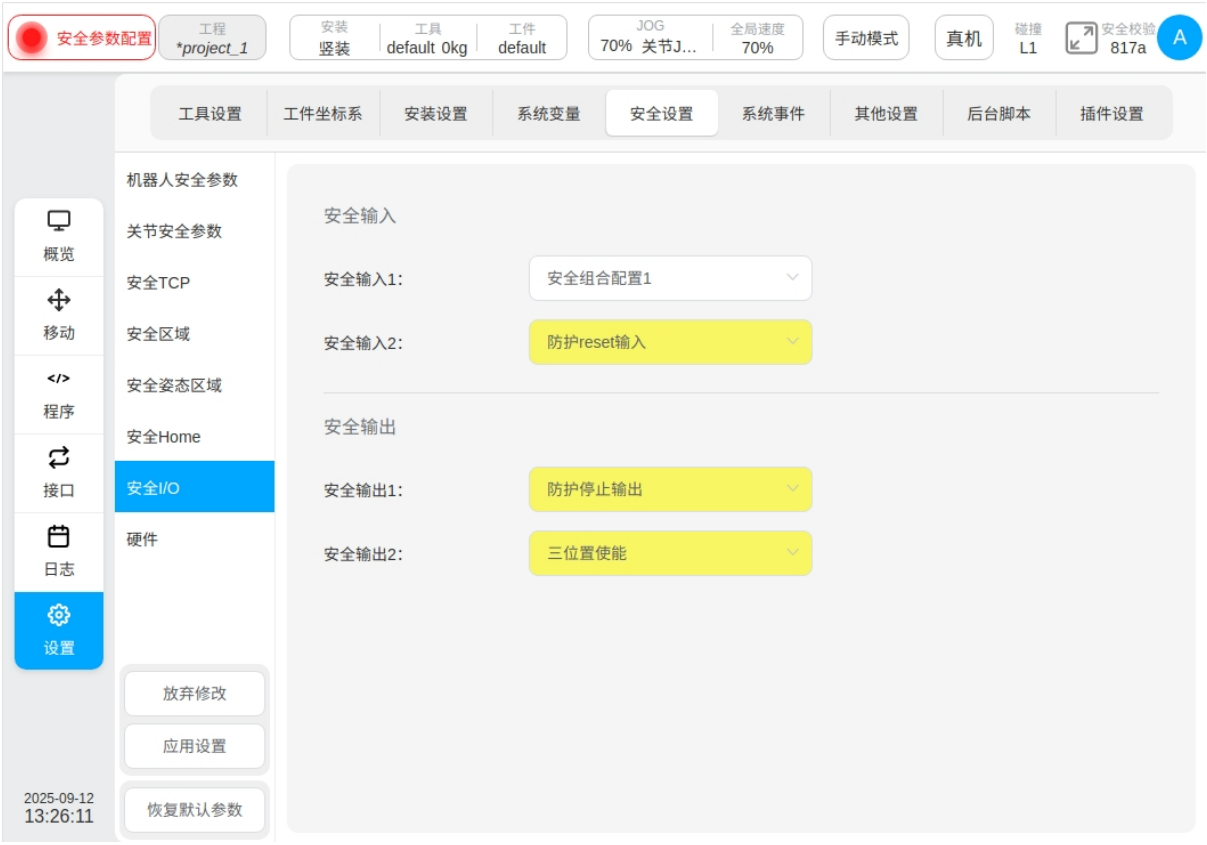
防护停止输出：当机器人处于防护停止状态时，触发该端口

自动模式防护停止输出：当机器人处于自动模式防护停止时，触发该端口

Reduce 模式输出：当机器人处于缩减模式时，触发该端口

HOME 位置：当机器人处于安全 HOME 位置时，触发该端口

三位置使能：当示教器上的三位置开关处于中间位置时，触发该端口，并且不管三位置使能输入功能是否启用，都会触发该端口输出

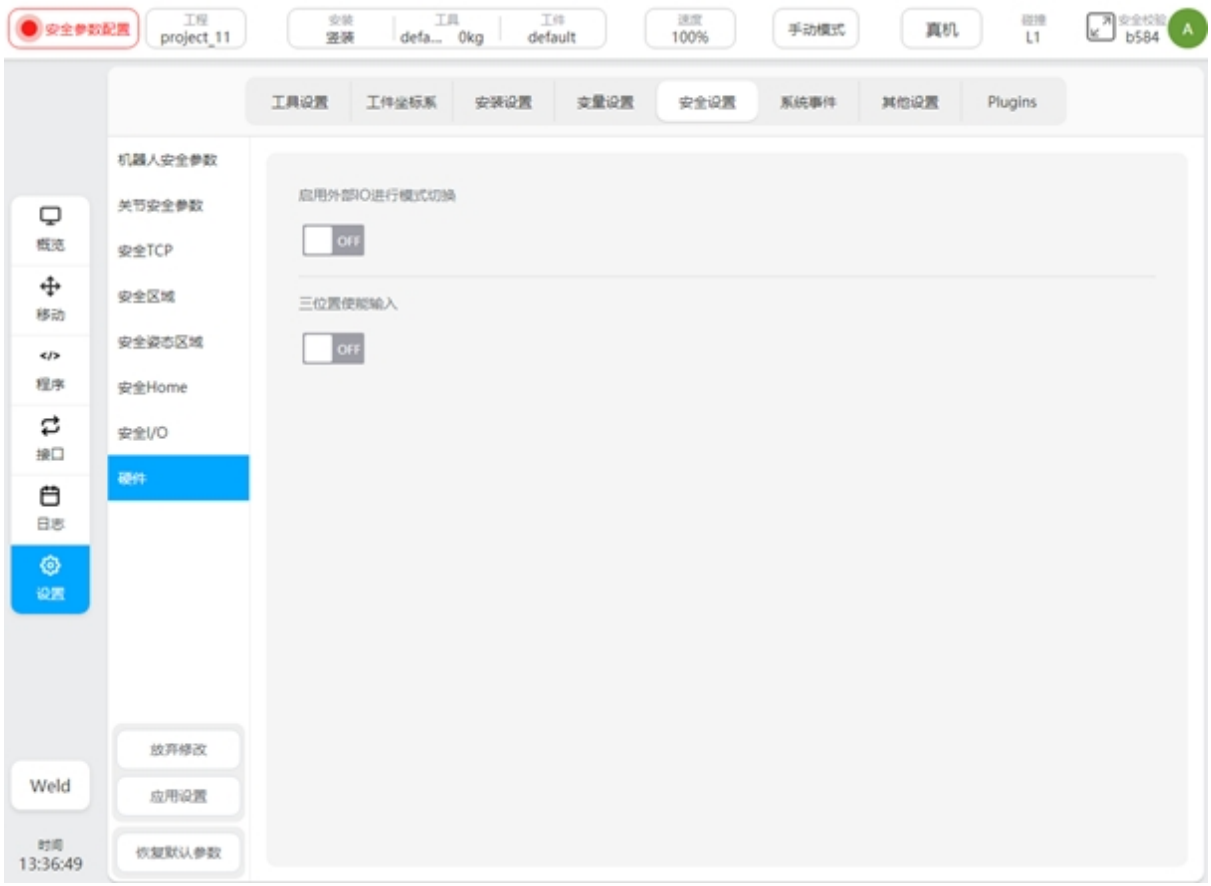


硬件

包含启用外部 IO 进行模式切换和三位置使能输入

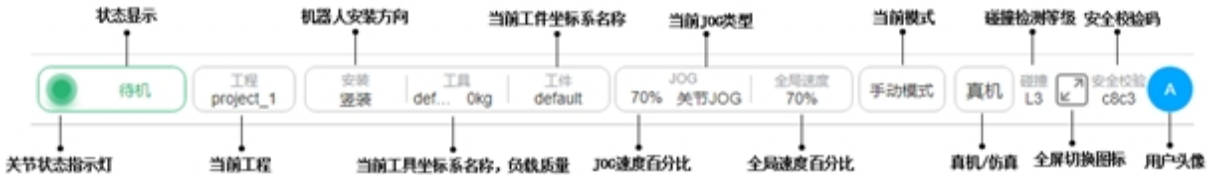
启用外部 IO 进行模式切换：启用此项，可以通过外部 IO 进行手自动模式的切换，此时界面状态栏上的模式切换功能无效

三位置使能输入：启用此项，当机器人处于手动模式下，仅当示教器上的三位置开关处于中间位置才可以移动机器人，机器人在移动过程中任意时刻三位置开关处于非中间位置时都会触发机器人的暂停



2.5 状态栏

界面头部状态栏如图所示。



机器人状态指示灯：

机器人初始未上电状态，指示灯为红色；机器人上电未使能状态，指示灯为橙色；机器人使能状态，指示灯绿色。该指示灯的颜色与启动界面中关节状态指示灯颜色定义保持一致。

状态显示区：

该区域会显示安全状态、程序状态、机器人状态三类信息。

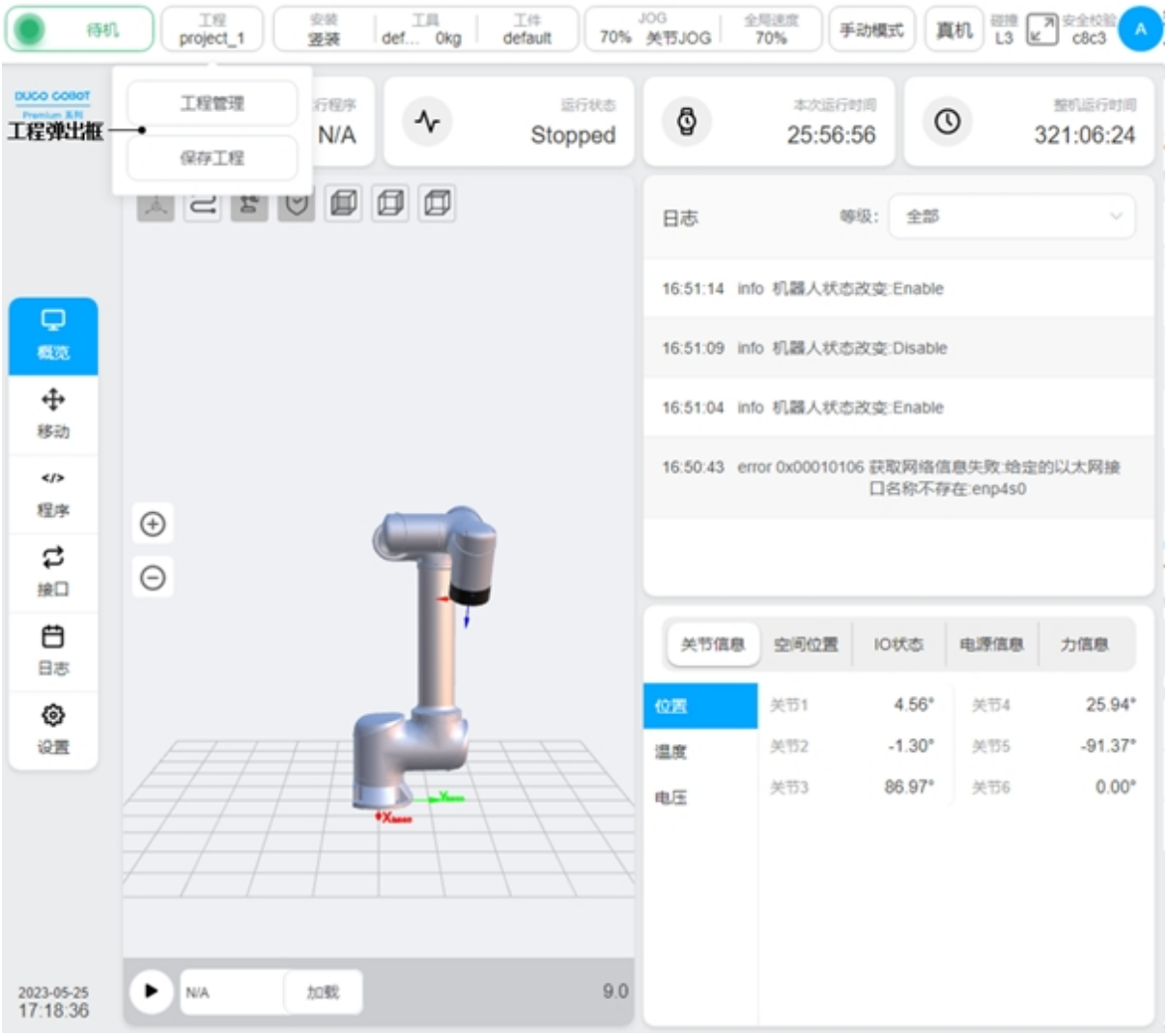
安全状态显示字样有：缩减模式、恢复模式、安全参数配置、安全板固件升级、Stop0、Stop1、Stop2、安全板错误、机械臂固件升级；

程序状态显示字样有：程序停止中、程序执行中、程序暂停、程序暂停中、任务执行中；

机器人状态显示字样有：未上电、未使能、待机、固件更新。

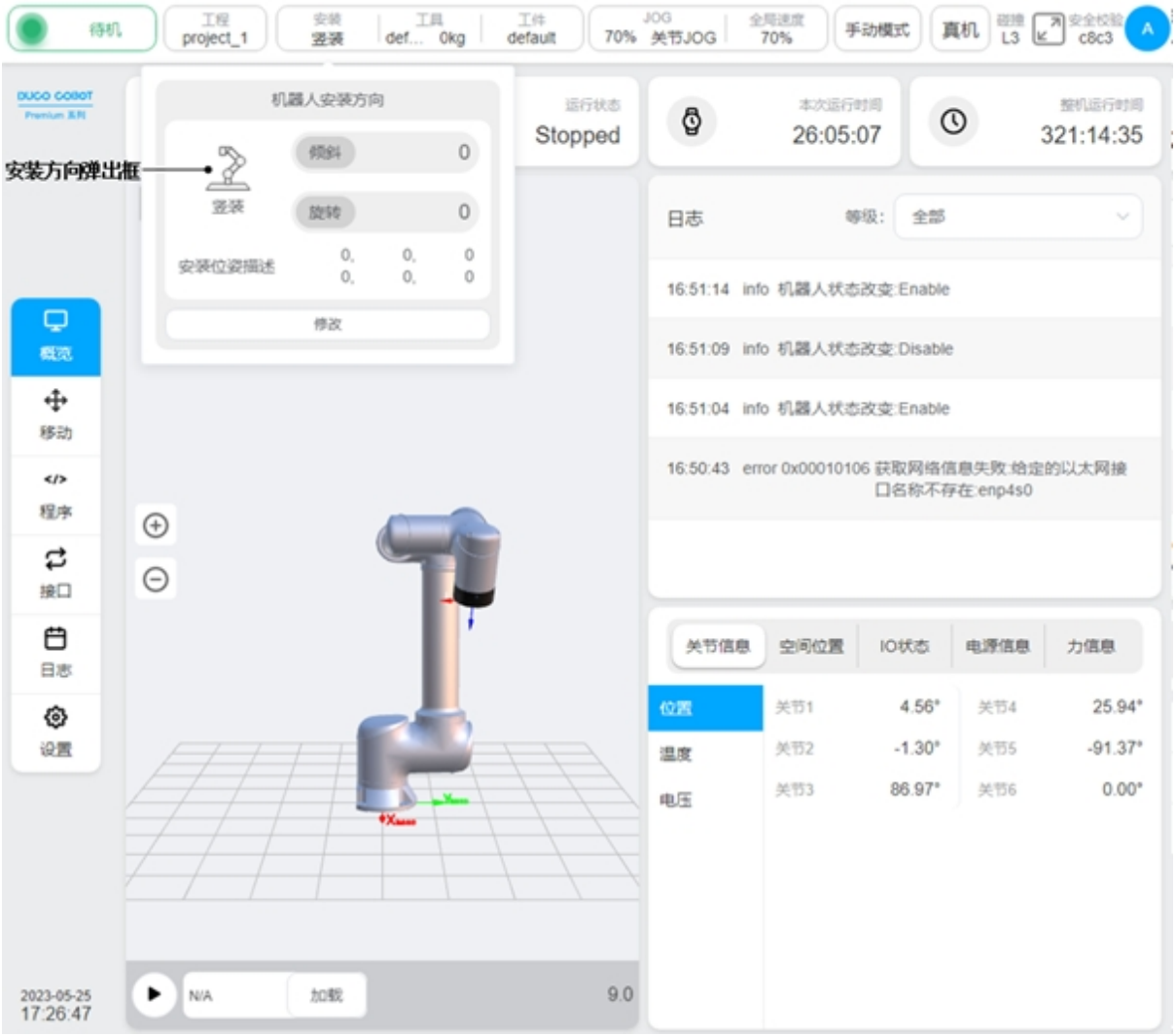
工程：

显示当前工程名。单击“工程”，会显示包含“工程管理”和“保存工程”按钮的弹出框，如图所示。其中，工程管理部分，可以进行对工程进行新建、导入、导出、恢复、保存到本地、备份到云端、删除、修改工程名等操作。

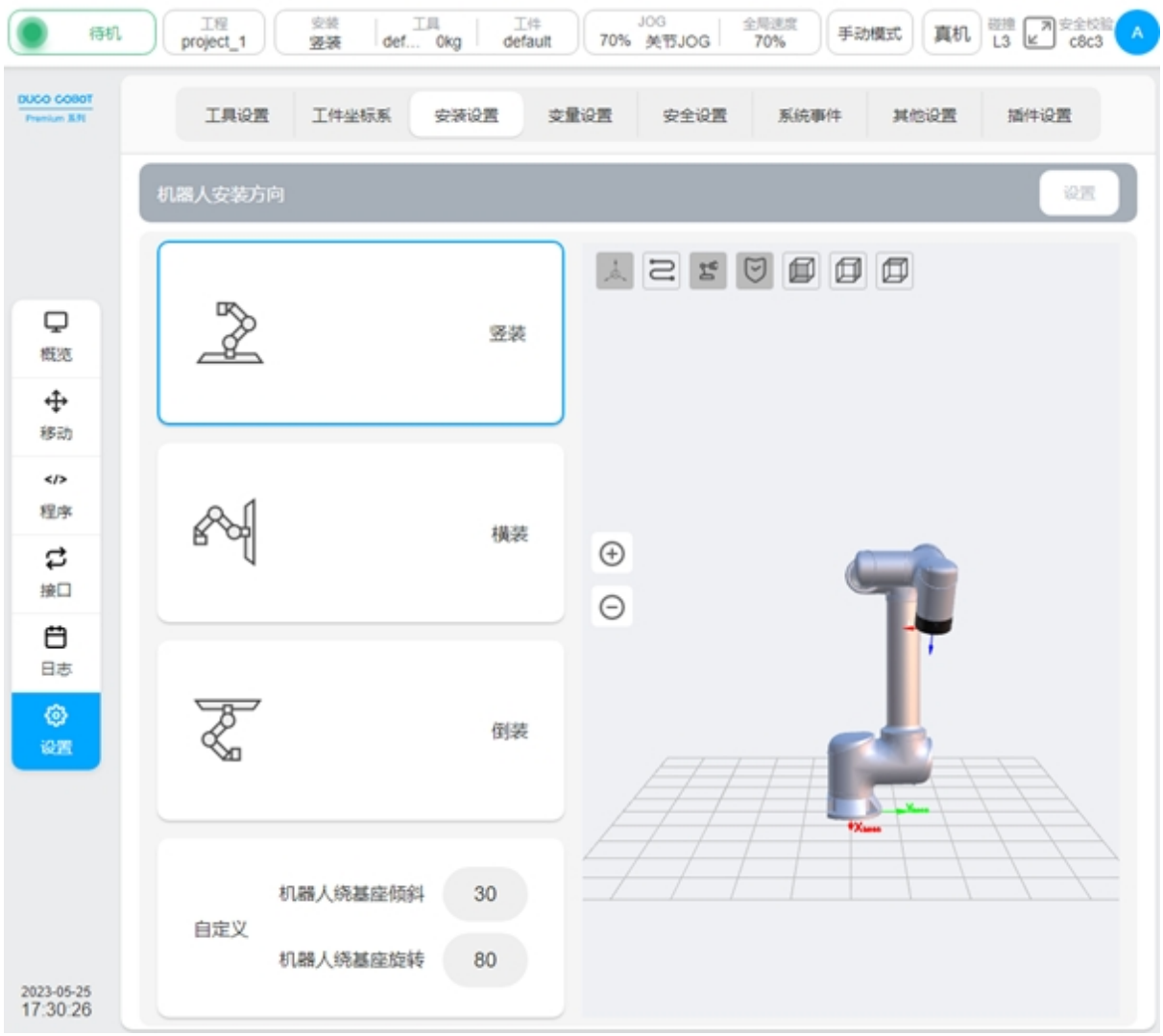


安装：

显示机器人当前安装方向以及安装位姿描述。单击“安装”显示框，显示机器人安装方向和安装位姿描述信息弹出框，如图所示。

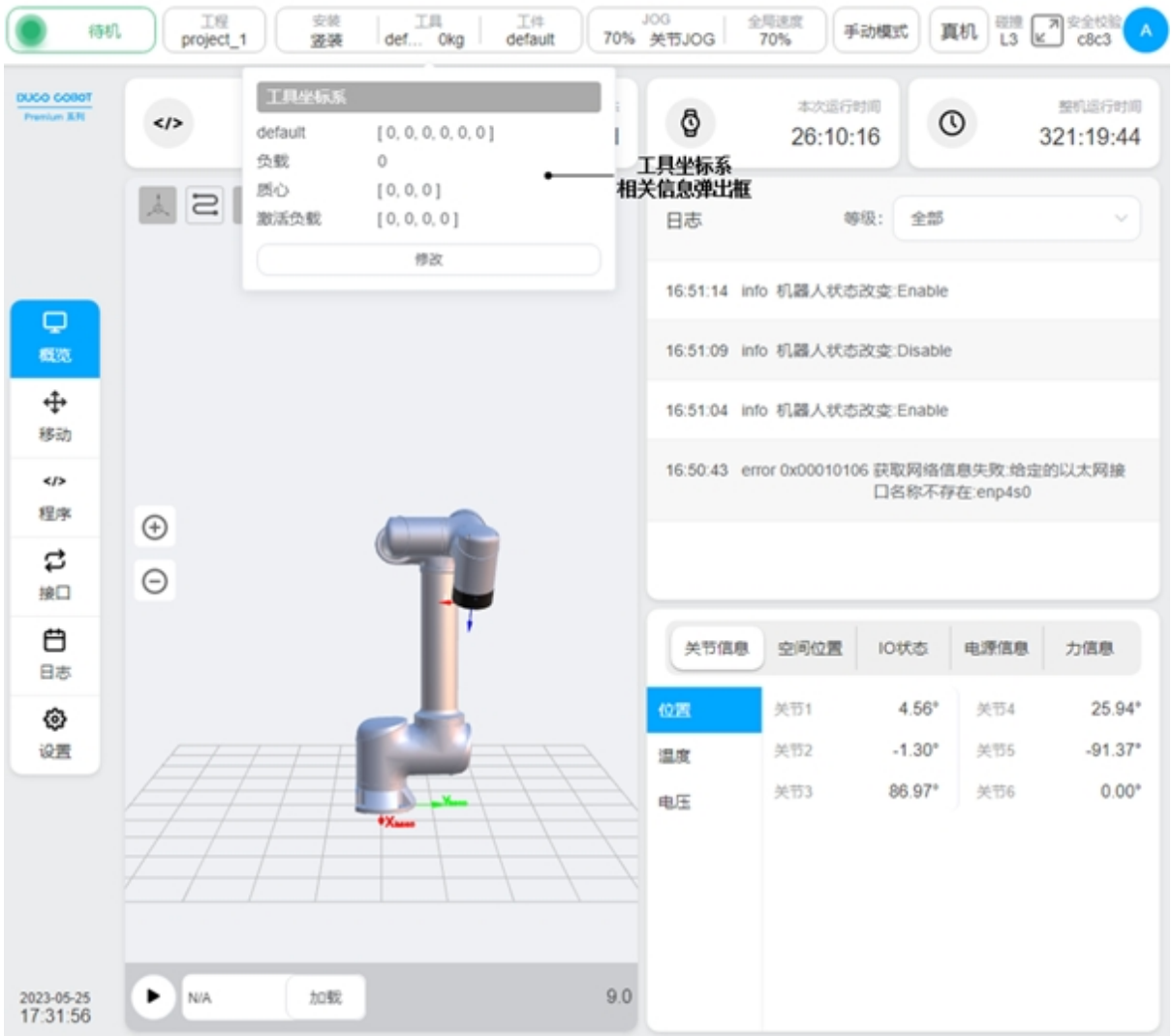


单击“修改”按钮，会跳转到设置界面里安装设置子页面，如图所示。用户可在该页面设置机器人安装方向操作。



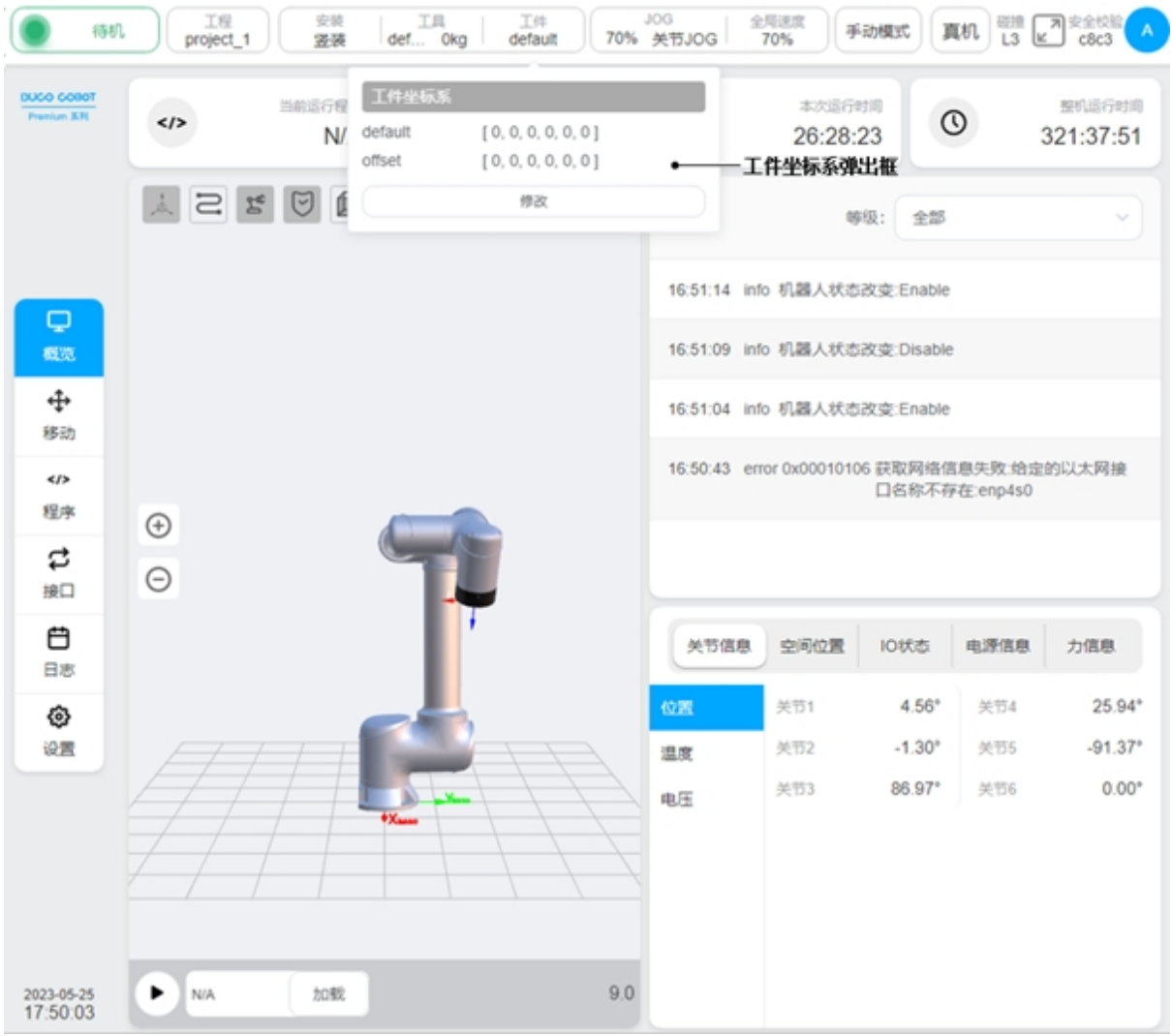
工具：

显示机器人当前工具坐标系名称和负载质量。单击“工具”显示框，显示机器人工具坐标系相关信息的弹出框，如图所示。单击“修改”按钮，会进入设置界面里工具设置子页面，与上述跳转到安装设置界面类似。



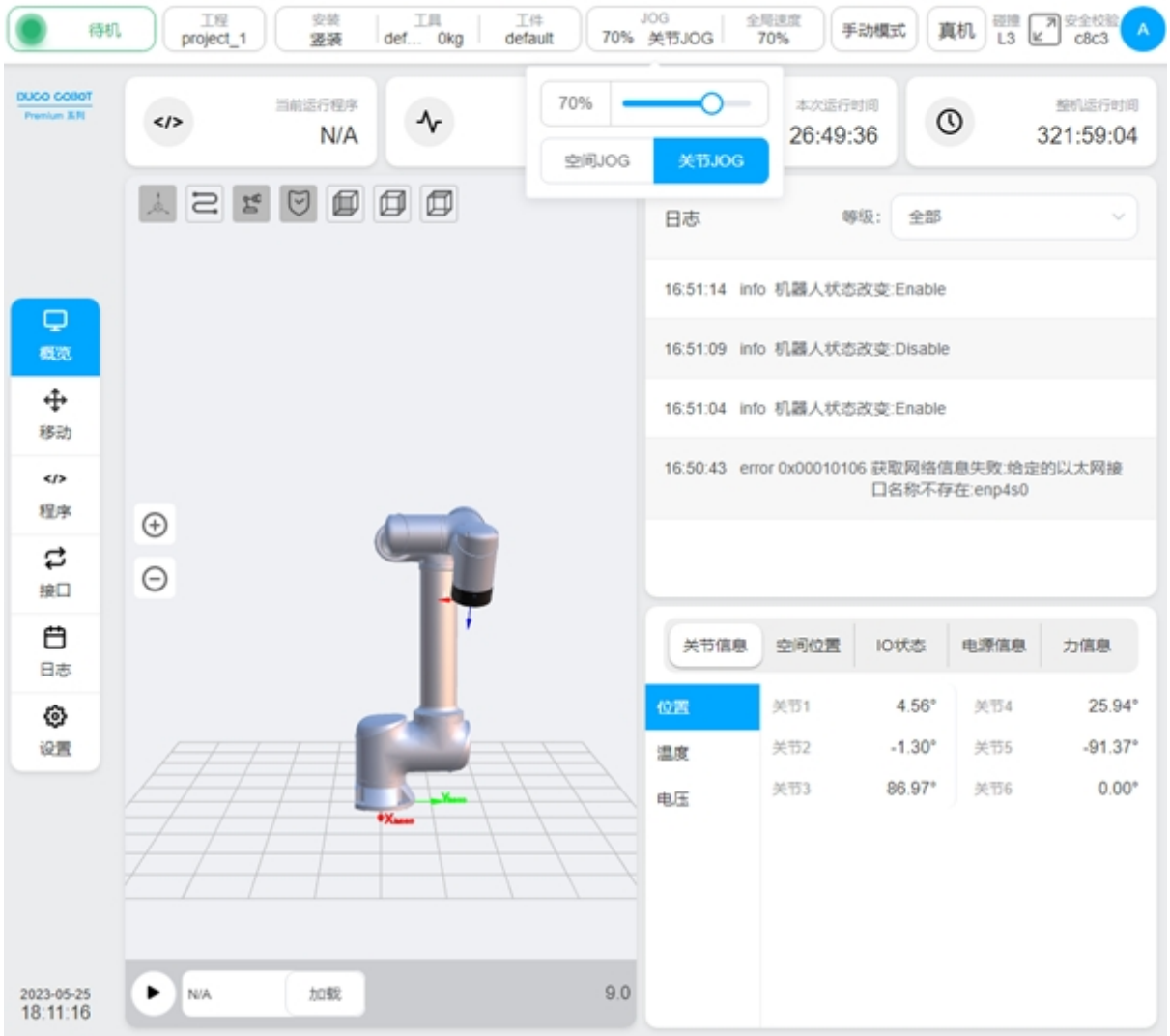
工件：

显示机器人当前工件坐标系名称。单击“工件”显示框，显示机器人工件坐标系信息的弹出框，如图所示。单击“修改”按钮，会进入设置界面里工件设置子页面，与上述跳转到安装设置界面类似。



JOG 速度/类型:

显示机器人当前 JOG 速度的百分比。单击“JOG”显示框，显示 JOG 速度调节滑块和 JOG 类型选择按钮，可以调节 JOG 速度或手动输入以及对空间 JOG 或关节 JOG 进行切换。注意：仅影响本次开机时的机器人 JOG 速度，下次开机失效，若需要更改每次开机时的 JOG 速度，请在“设置”——“其他设置”——“启动设置”中设置默认 JOG 速度。



全局速度：

显示机器人当前全局速度的百分比。单击“速度”显示框，显示速度调节滑块，可以调节全局速度或手动输入。注意：仅影响本次开机时的机器人全局速度，下次开机失效，若需要更改每次开机时的全局速度，请在“设置”——“其他设置”——“启动设置”中设置默认全局速度。

操作模式：

显示机器人操作模式，有手动模式和自动模式两种。单击该显示框，会显示包含“手动模式”和“自动模式”按钮的弹出框，通过单击对应按钮进行操作模式的切换。当用户进行手动/自动模式切换时，用户需要输入当前登陆密码，输入正确密码后，才可以进行模式的切换。当启用物理硬件模式切换信号后，界面中的模式切换按钮无效。

真机：

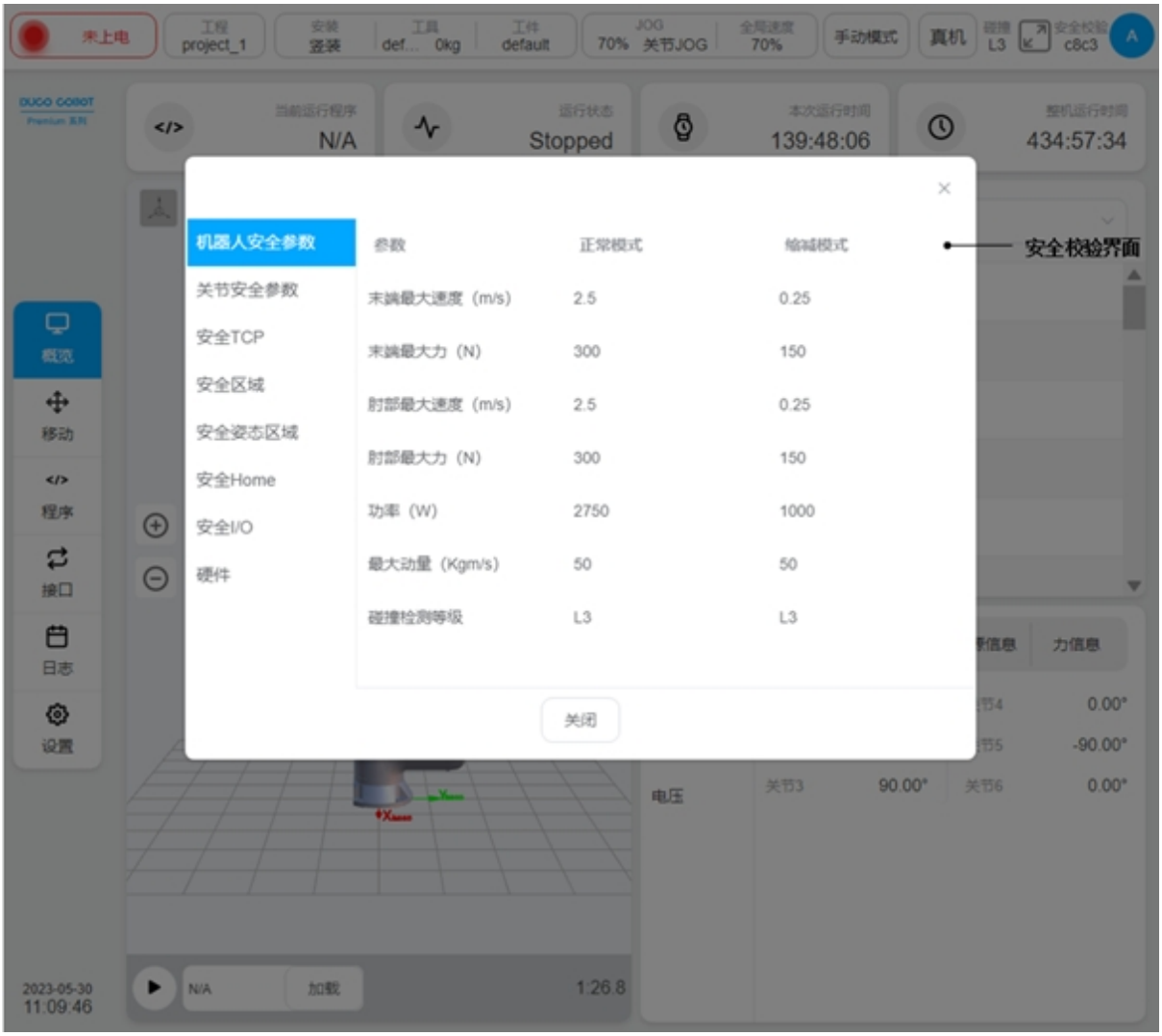
切换真机和仿真模式。仿真模式只能在机器人处于使能待机且程序停止状态下设置。当机器人由于任何原因产生过断使能行为后，机器人将自动切换回真机模式。

碰撞检测：

碰撞检测显示当前的碰撞检测设置状态，以及安全等级。安全等级设置详见第 6.3.2 章节。

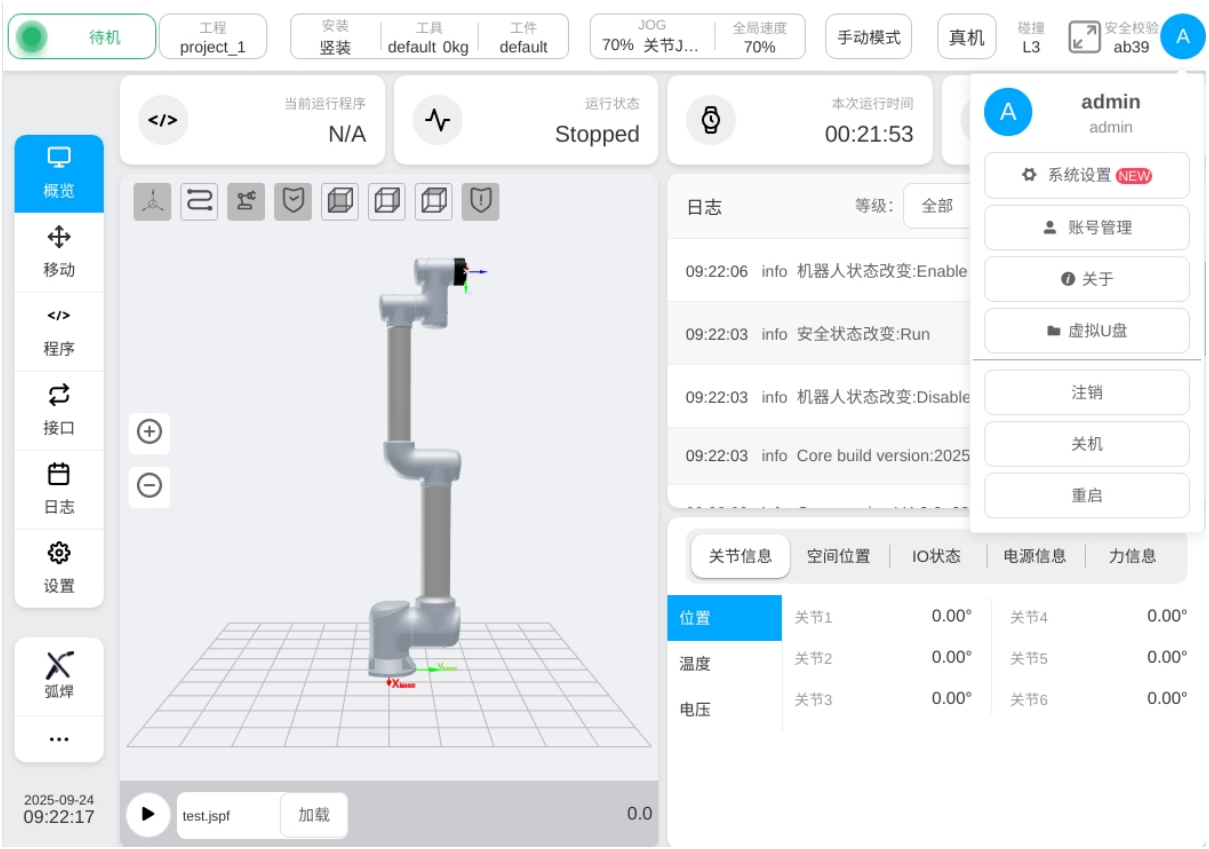
安全校验码：

显示安全校验码。单击“安全校验”区域，会弹出安全校验界面，如图所示。该界面显示的是设置界面里安全设置子页面下设置的安全配置参数。单击安全校验界面下方“关闭”按钮，或者界面右上角叉号，可以关闭页面。

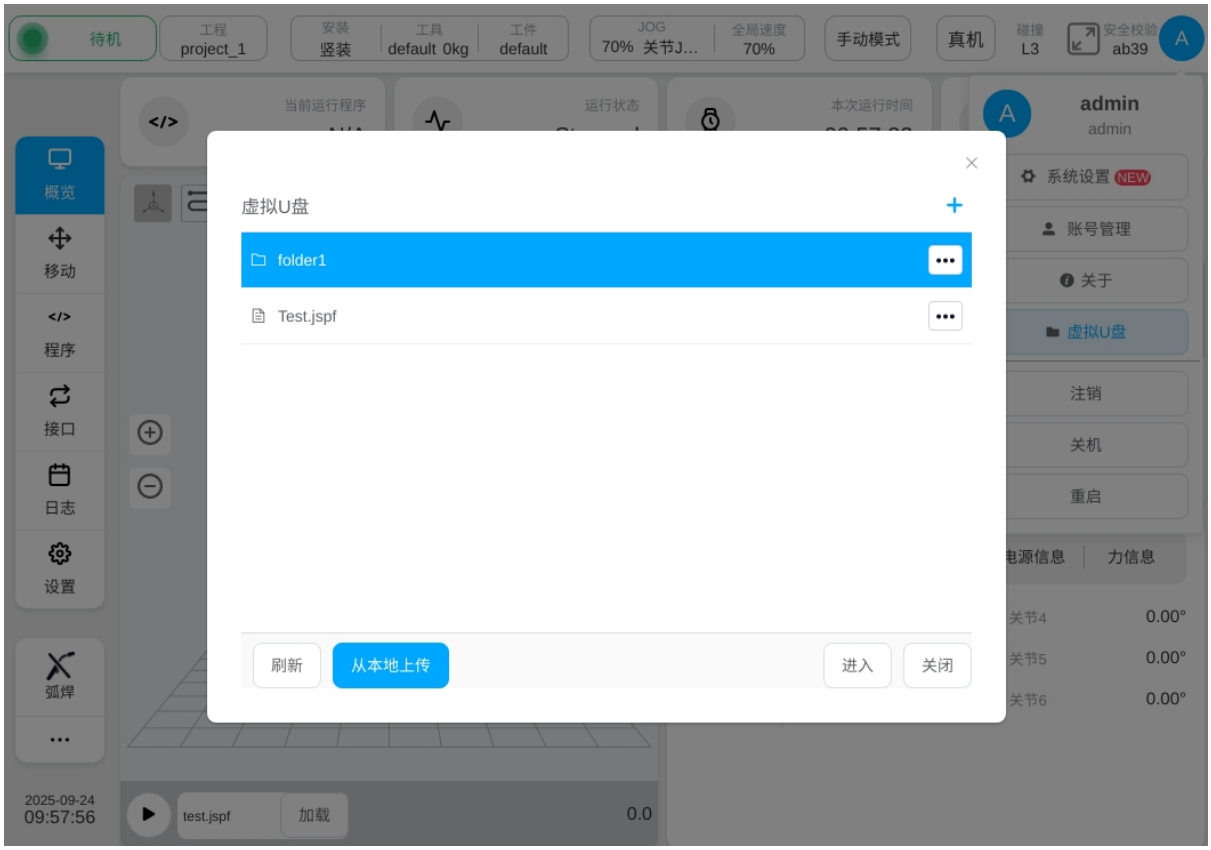


用户头像：

显示登录用户名称首字母的大写字母。单击用户头像，显示如图所示弹出框。其中，包含“更改密码”（除 admin 权限以外用户显示）、“系统设置”、“账号管理”（只有 admin 权限用户显示）、“关于”、“虚拟 U 盘”、“注销”、“关机”和“重启”按钮。



点击“虚拟 U 盘”按钮，会显示虚拟 U 盘弹框，如图所示。可查看当前虚拟 U 盘中文件夹及文件，也可以进行文件的上传、保存至本地及删除。点击左下角”从本地上传“按钮，可将本地文件上传至该虚拟 U 盘，在系统更新、机器人参数导入、工程导入、程序导入等功能处选择上传至该虚拟 U 盘的文件。



备注： 文件的上传、保存至本地的功能仅能在 PC 浏览器端操作，示教器上仅能查看以及进行文件的删除操作，不可上传及下载。

2.6 输入键盘

2.6.1 普通输入键盘

在点击输入框时，会弹出如下输入键盘

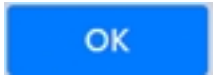
键盘输入框的左侧显示输入提示，右侧是输入的格式要求。

按键可分为字符按键和功能按键

字符按键与常用键盘的布局基本一致，可通过”Shift”按键切换大小写及字符

功能按键作用有：

退出键盘

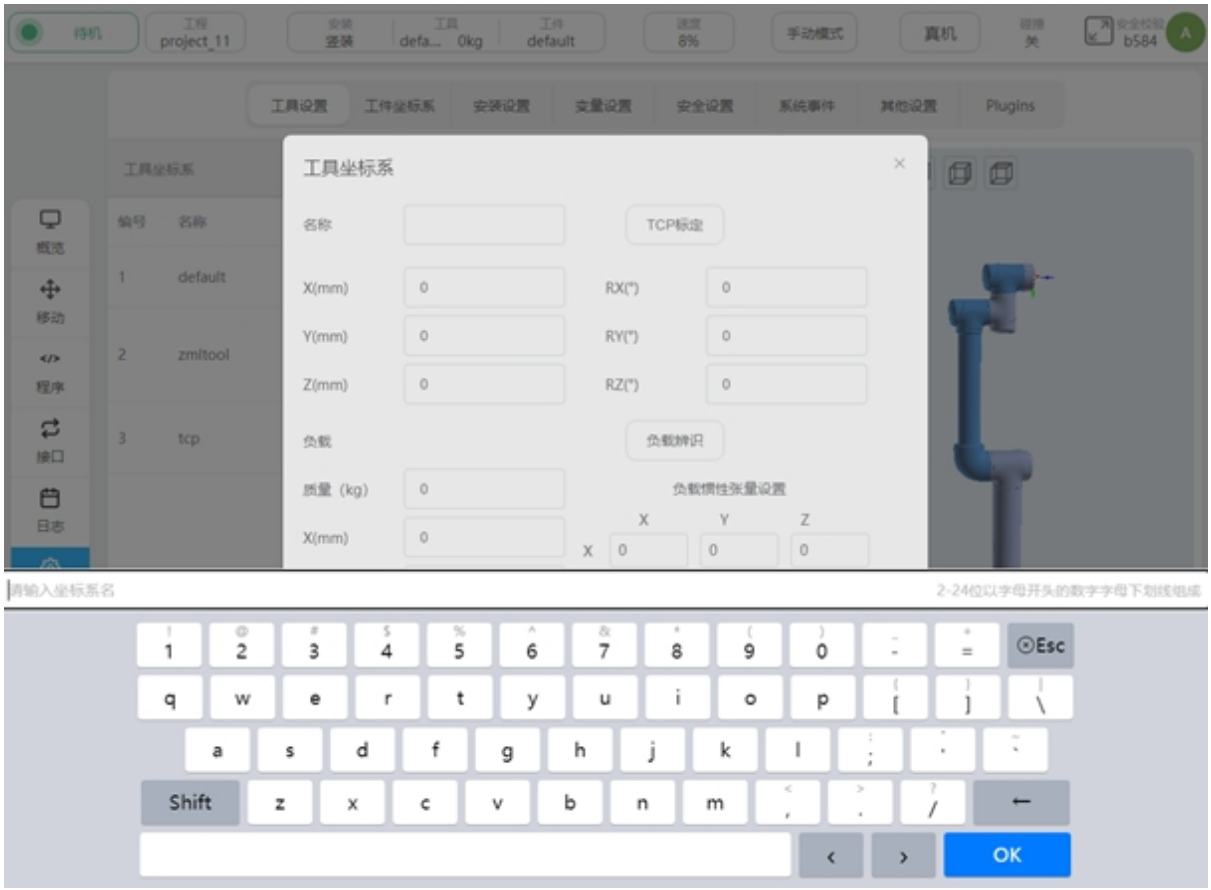


确定输入，若不符合输入格式则无效

切换大小写及字符

删除当前光标前的字符，长按可连续删除当前光标前的字符

光标左右移动，长按可连续移动



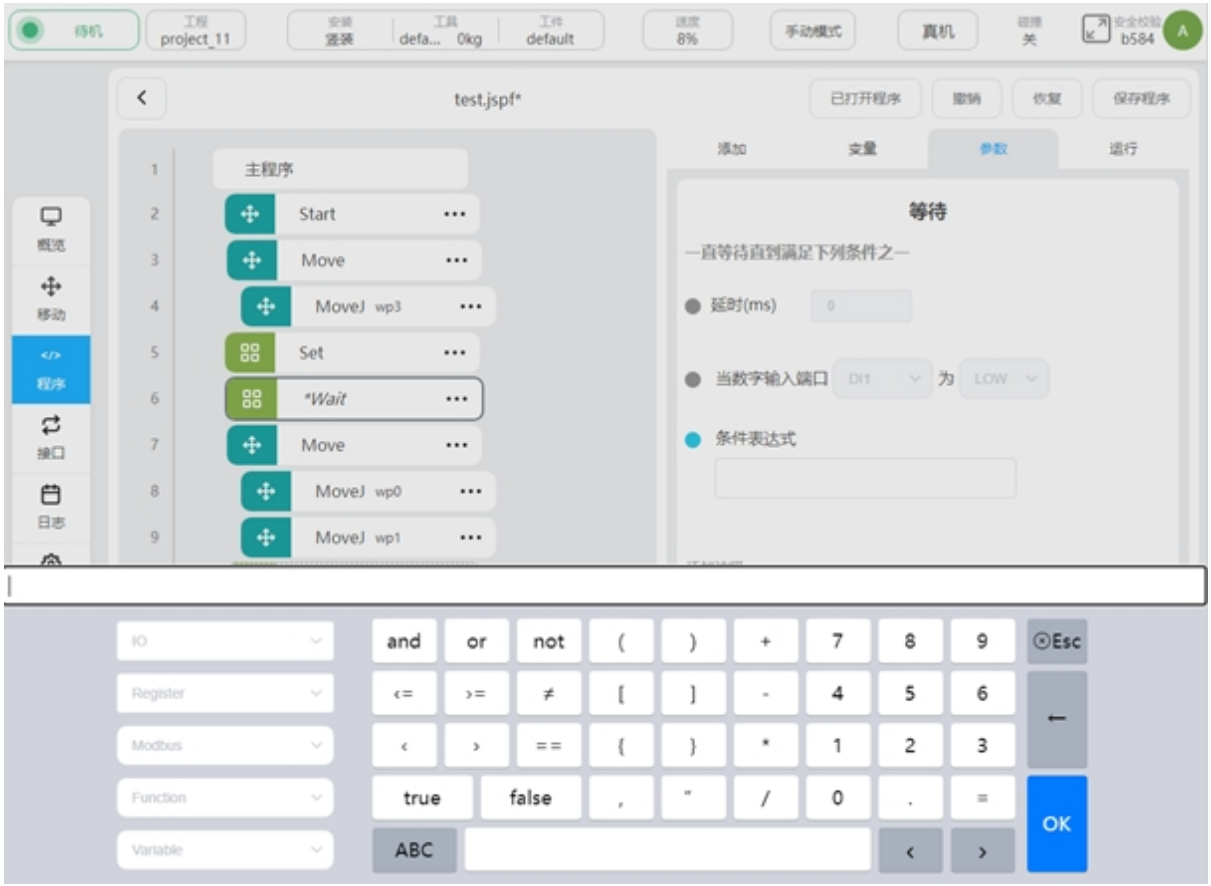
弹出输入键盘后，用户仍可以使用实体键盘输入，点击实体键盘上的“Enter”为确认输入，点击“Esc”为直接退出键盘，点击左右方向键为向左右移动光标，点击上下方向键为将光标移动到最前或最后。

2.6.2 表达式输入键盘

在编程时，某些功能块需要输入表达式，此时会弹出表达式键盘。表达式键盘主要用来快速输入表达式，并且表达式键盘会对表达式做自动转换生成机器人的脚本语句，表达式键盘也会检查表达式是否合法。

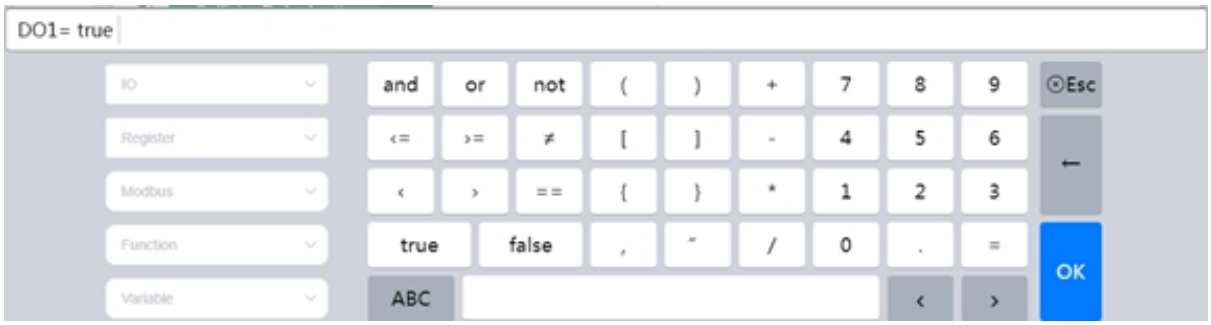
表达式键盘左侧下拉输入框可进行选择，快速输入 IO、寄存器、Modbus、函数及变量。右侧可以输入常用的逻辑运算符：and、or、not，关系运算符：<=、>=、<、>、==、≠，算术运算符：+、-、*、/，boolean 量：true、false，以及常用的符号与数字。

表达式键盘的功能按键与普通输入键盘类似。可以通过 ABC 按键来切换到普通输入键盘，如图



区别于机器人脚本，表达式键盘的输入较为简单，可以自动将其转换为机器人的脚本。例如给数字输出端口 1 置为高电平，表达式键盘如下输入即可。实际执行时会自动将其转换为机器人的脚本：

```
set_standard_digital_out(1,true)
```



以下是一些表达式的输入示例

IO

条件表达式 DI1 为高电平且 DI2 为低电平:

```
DI1 == true and DI2 == false
```

条件表达式末端数字输入口 1 为高电平:

```
Tool_DI1 == true
```

DO1 置为高电平:

```
DO1 = true
```

DO2 置为低电平:

```
DO2 = false
```

寄存器

条件表达式 bool 输入寄存器 1 为 true:

```
bool_reg_in1 == true
```

条件表达式 word 输入寄存器 2 为 32:

```
word_reg_in2 == 32
```

条件表达式 float 输入寄存器 1 小于等于 2.5:

```
float_reg_in1 <= 2.5
```

bool 输出寄存器 1 设为 true:

```
bool_reg_out1 = true
```

word 输出寄存器 1 设为 16:

```
word_reg_out1 = 16
```

float 输出寄存器 1 设为 1.2:

```
float_reg_out1 = 1.2
```

Modbus

条件表达式可读写线圈 mb1 值为 1:

```
mb1 == 1
```

条件表达式可读写寄存器 mb3 值大于 10:

```
mb3 > 10
```

可读写线圈 mb1 设为 1:

```
mb1 = 1
```

可读写寄存器 mb3 设为 12:

```
mb3 = 12
```

变量

条件表达式全局变量 g_bo 值为 true:

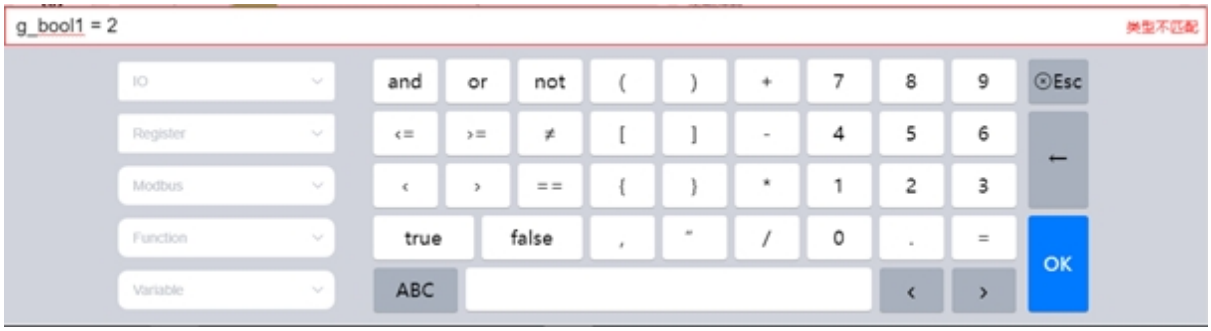
```
g_bo == true
```

全局变量 g_num 赋值 3:

```
g_num = 3
```

输入完成，点击“OK”时，会检测表达式的合法性。

如图，给一个全局 boolean 类型的变量赋值一个数字，会报错



2.6.3 数字输入键盘

用在点击数字输入框时，会在输入框附件弹出数字键盘，如图

数字键盘用来输入数字，在弹出键盘后，用户仍可通过实体键盘输入。

数字键盘上各个功能按键的作用：

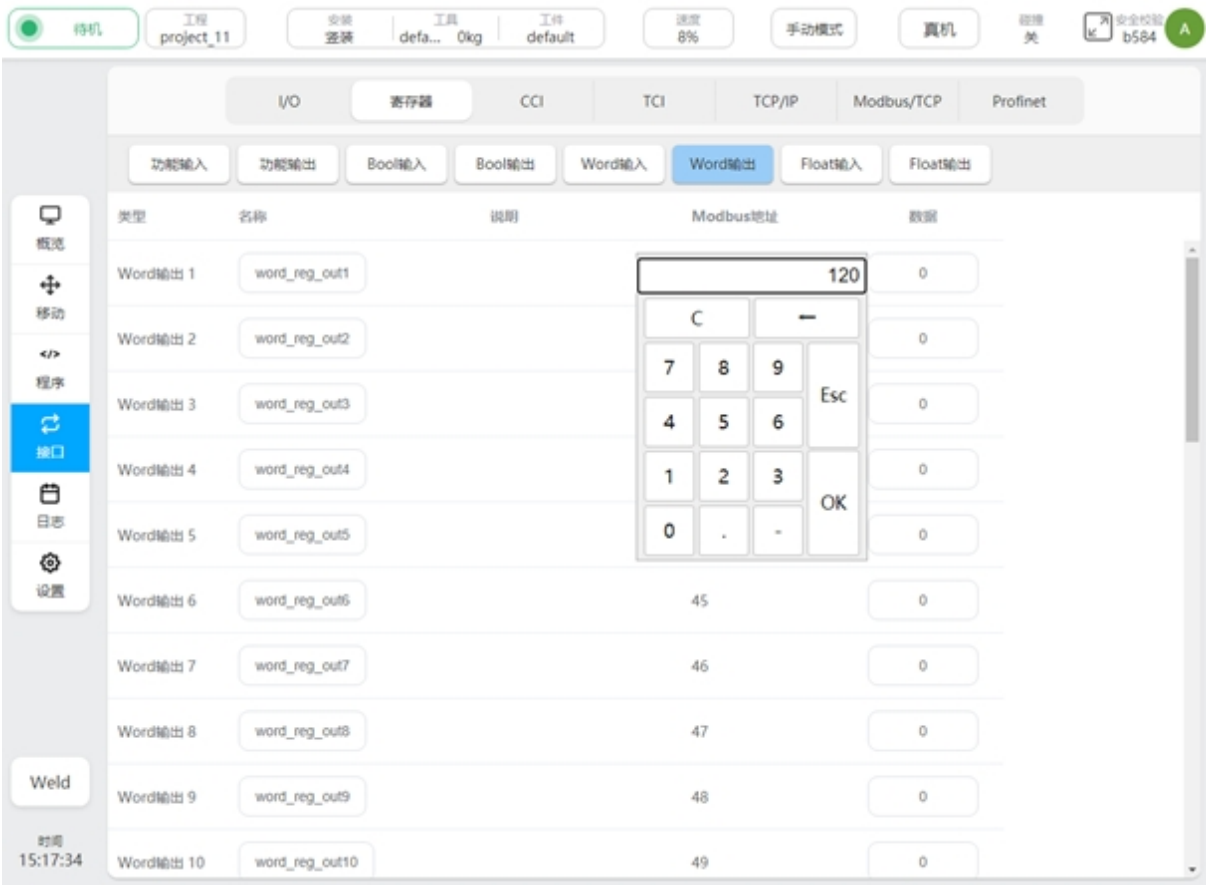
从清空输入框

删除字符

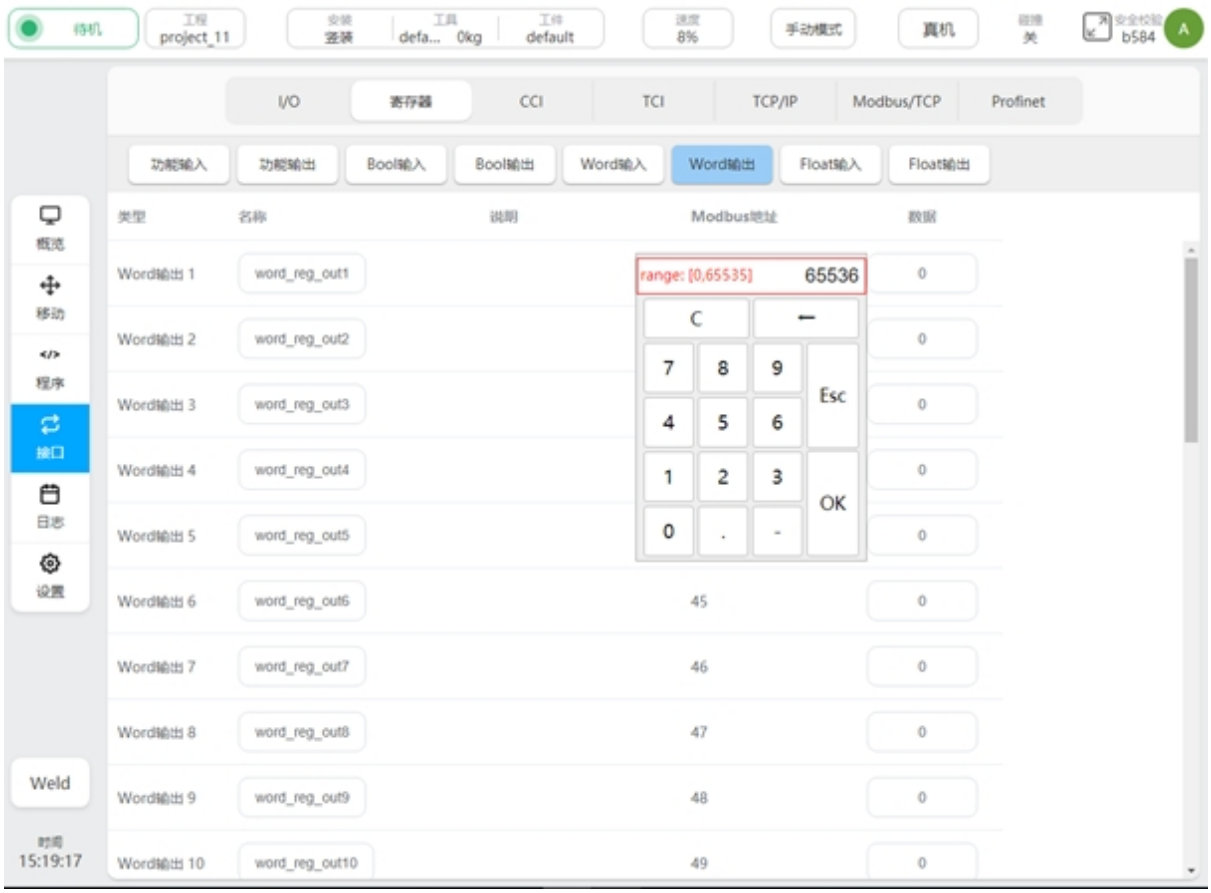
改变数字的符号

取消键盘输入

确定键盘输入

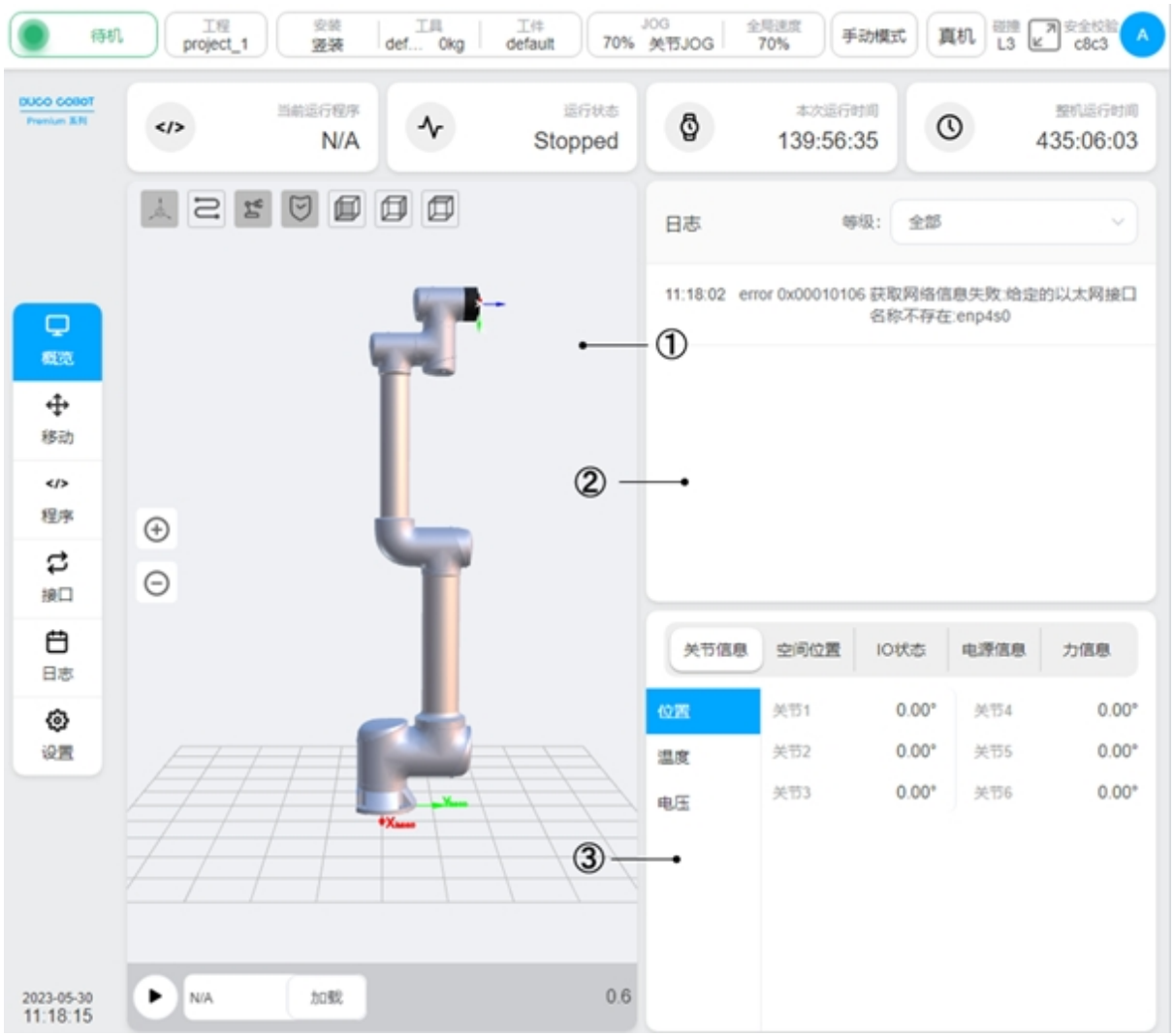


数字键盘会检测输入的数字是否合法，如图，当输入的数字超过范围，输入框左侧将会报错，并提示有效的输入范围

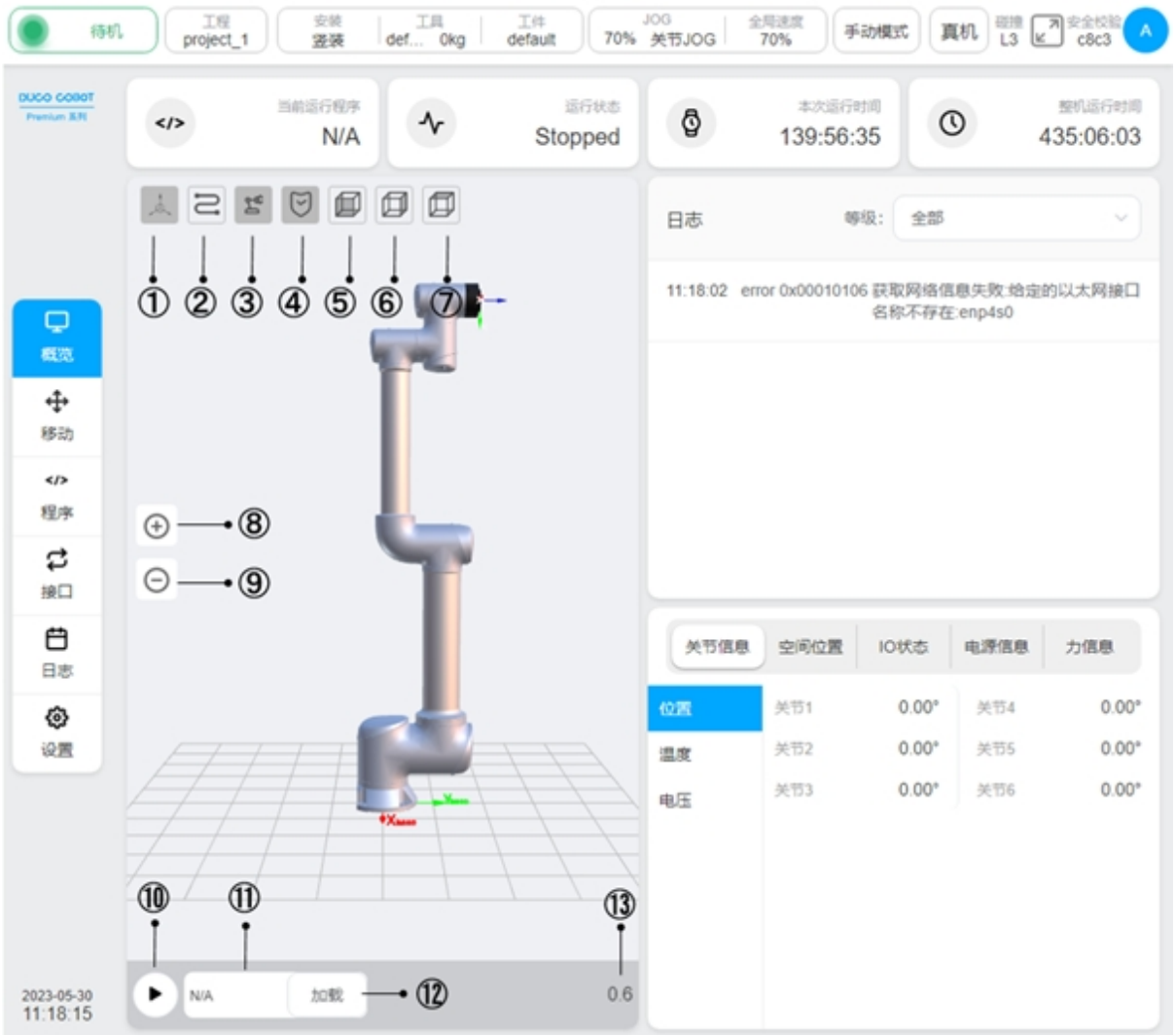


2.7 概览页面

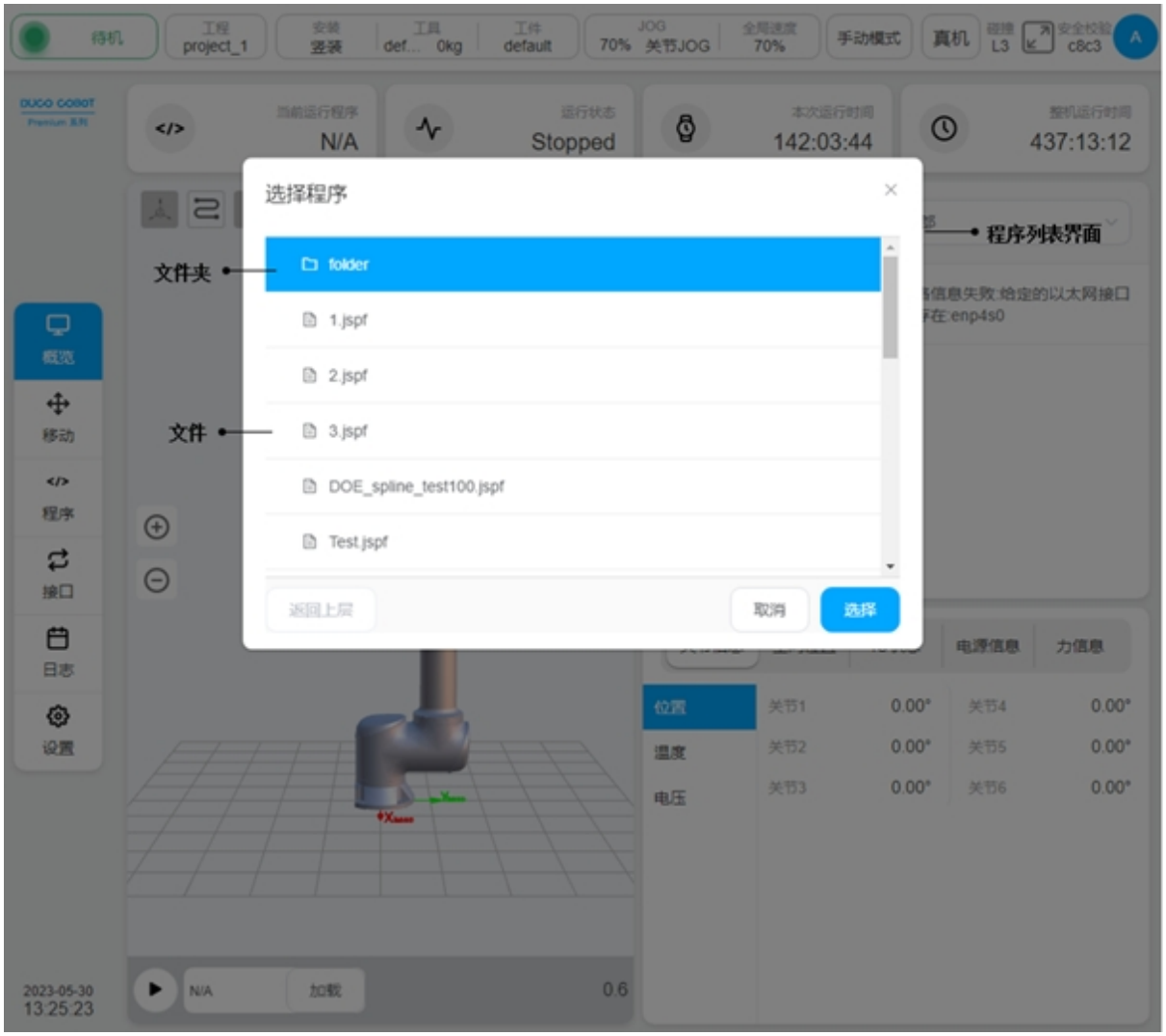
单击导航栏中“概览”会进入概览页面，如图所示。页面内容区上方显示：当前运行程序、程序运行状态、本次运行时间和整机运行时间。页面左侧 ① 显示 3D 仿真模型；右侧上方 ② 显示日志信息，日志可通过下拉框选择全部、致命、错误、警告、信息 5 个不同等级进行查看；右侧下方 ③ 显示机器人关节信息（位置/温度/电压）、空间位置（基座坐标系下）、IO 状态、电源信息（控制柜）及力信息。



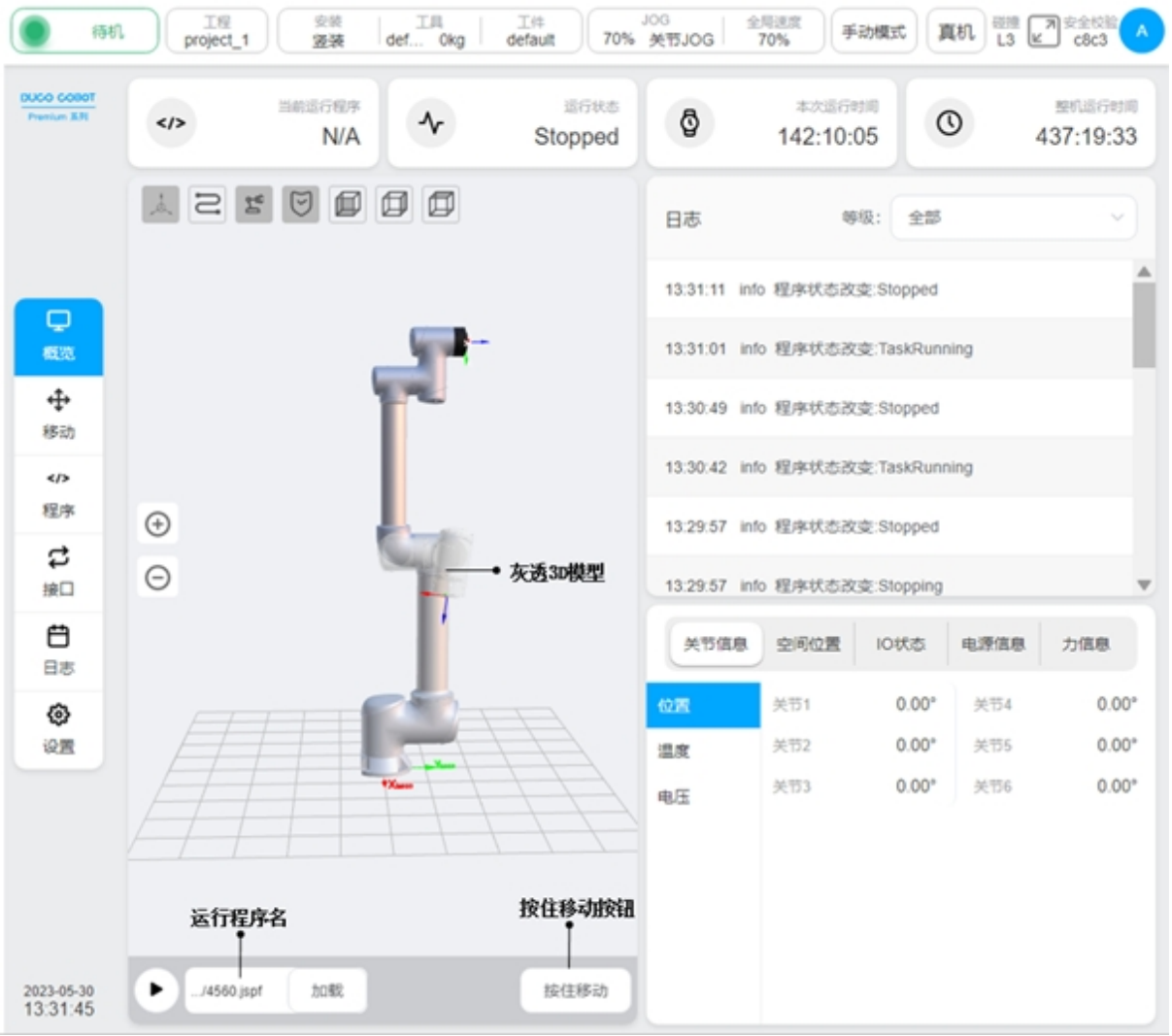
3D 仿真模型区如图所示，左上方图标 ①-⑦ 依次表示是否显示：工件坐标系和工具坐标系、末端轨迹、机器人 3D 仿真模型、安全区域、正视图、右视图和俯视图。图标 ⑧ 和 ⑨ 分别是操作 3D 模型放大和缩小的按钮。图标 ⑩-⑫ 分别是程序运行按钮、当前运行程序显示框、加载程序按钮和程序运行时间显示。



单击“加载”按钮，会弹出程序列表界面，如图所示。程序列表罗列出所有文件夹（如：Test_1）和后缀名为.jspf 的程序文件（如：ee.jspf）。界面下方有“返回上层”、“取消”、“选择”三个按钮。

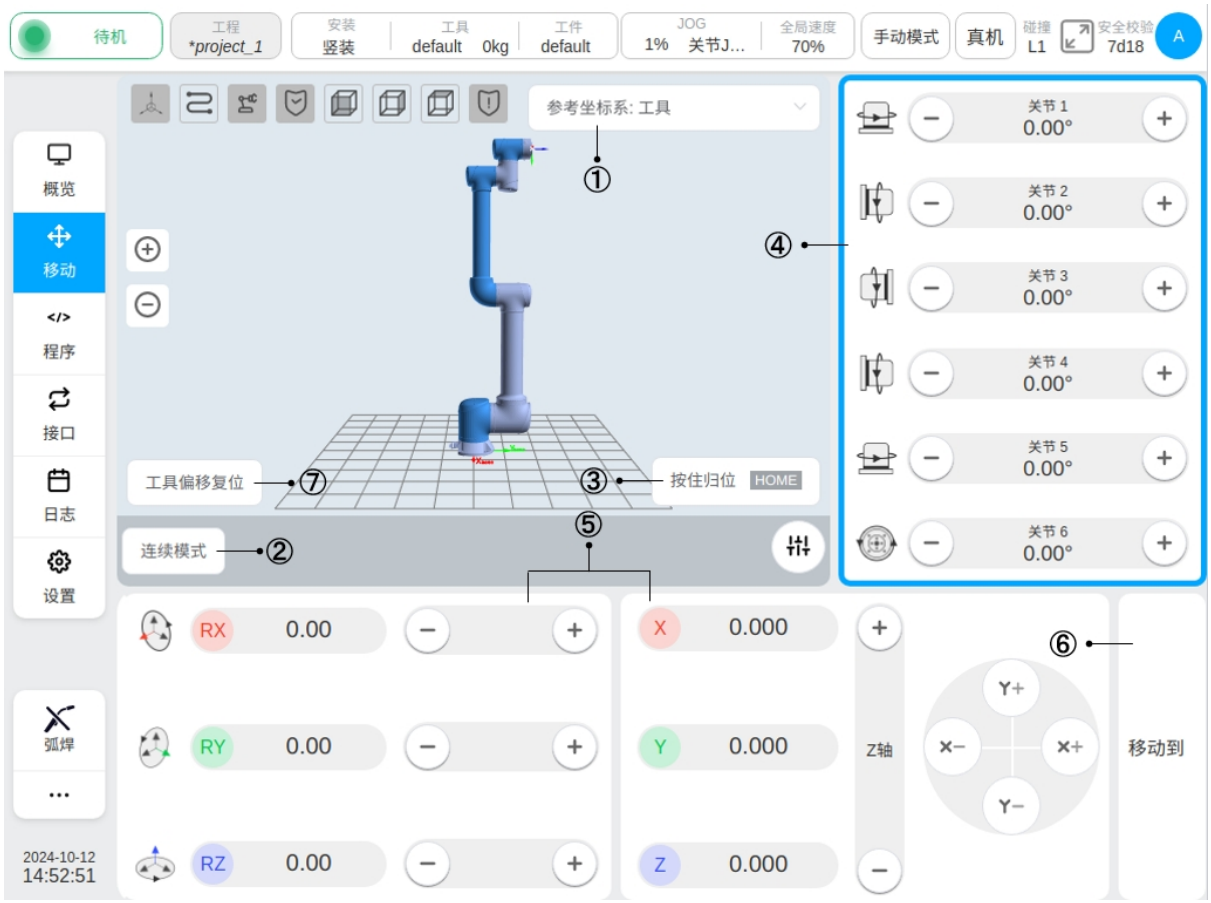


选择好需要加载运行的程序文件，单击界面“选择”按钮后，程序列表界面会自动消失，同时图标 ⍰ 处会显示被选择程序文件的文件名或者带有“…/”的文件名（存在文件夹嵌套，如：…/4560.jspf），然后单击“运行”按钮，如果程序配置了 Start 点，会出现“按住移动”按钮和灰色透视 3D 模型（后文统称灰透 3D 模型），如图所示。按住“按住移动”按钮，机器人会往程序里设置的初始位置移动，即灰透 3D 模型处，松开该按钮会停止移动。当机器人运动到灰透 3D 模型处，“按住移动”按钮会消失。此时再单击“运行”按钮，即可运行程序。如果程序没有配置 Start 点，将从当前位置直接运行。



2.8 移动页面

单击导航栏中“移动”会进入移动页面，如图所示。



页面左上方显示 3D 仿真模型，单击图标 ① 选择器，可选择当前机器人使用的坐标系和手动操作机器人的参考坐标系，默认会使用世界坐标系（此系统中同基座坐标系），表示以机器人底座为坐标基点。还可选择工具坐标系和工件坐标系，工具坐标系表示以末端工具的末端点为坐标基点，工件坐标系由用户自由设定。切换参考坐标系后，界面上显示的机器人笛卡尔坐标下的位置姿态数据会相应改变。

图标 ② 是用户可进行模式切换的按钮，按钮显示模式即为当前选择的移动模式。单击“连续模式”按钮，可切换到“步进模式”，且界面会显示步进角度和步进距离的调节按钮和数据显示框。用户自定义移动的步进角度和步进距离默认值为 0.5deg 和 0.5mm，最小允许设置为 0.1deg 和 0.1mm，最大允许设置为 5deg 和 5mm。用户可通过“-”“+”进行调节步进角度和步进距离。

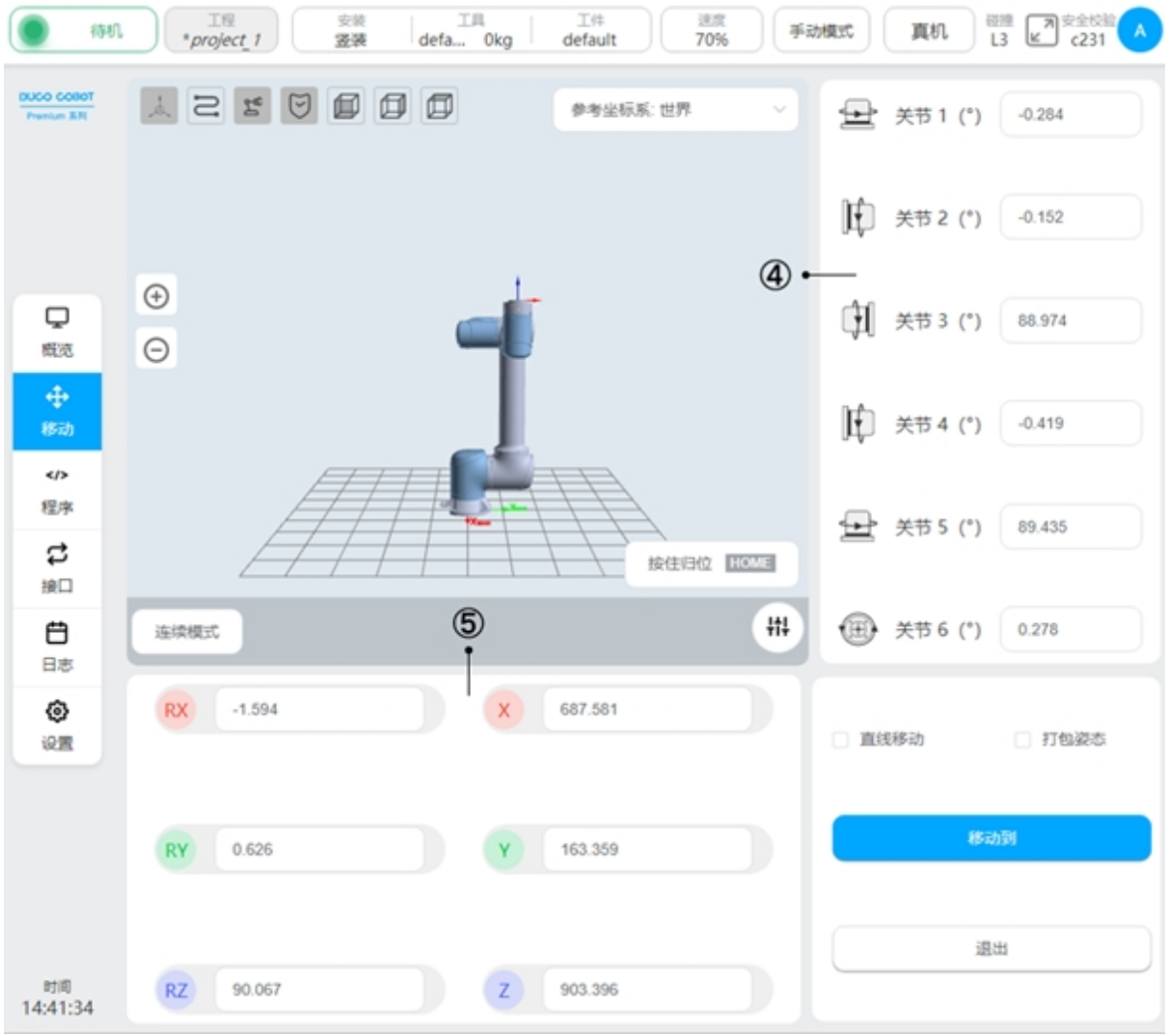
图标 ③ 是“按住归位”按钮，按住该按钮机器人会移动到设定的 HOME 点，松开按钮会停止移动。默认 HOME 点是每个关节角度值均为 0°，机器人呈直立状态。当机器人移动至“按住归位”按钮里“HOME”字样背景显示为绿色，说明机器人已达到 HOME 位置，如图所示。

区域 ④ 是手动操作机器人关节移动的区域，当单击该区域任意地方，区域会显示蓝色边框，代表当前手动操作机器人关节移动。按住某个关节的左边箭头或右边箭头按钮可移动该关节向正方向或反方向移动，松开按钮机器人关节会停止移动。用户也可以通过按住示教器上右侧“+”“−”按钮移动关节，按钮从上到下分别对应关节 1~6。

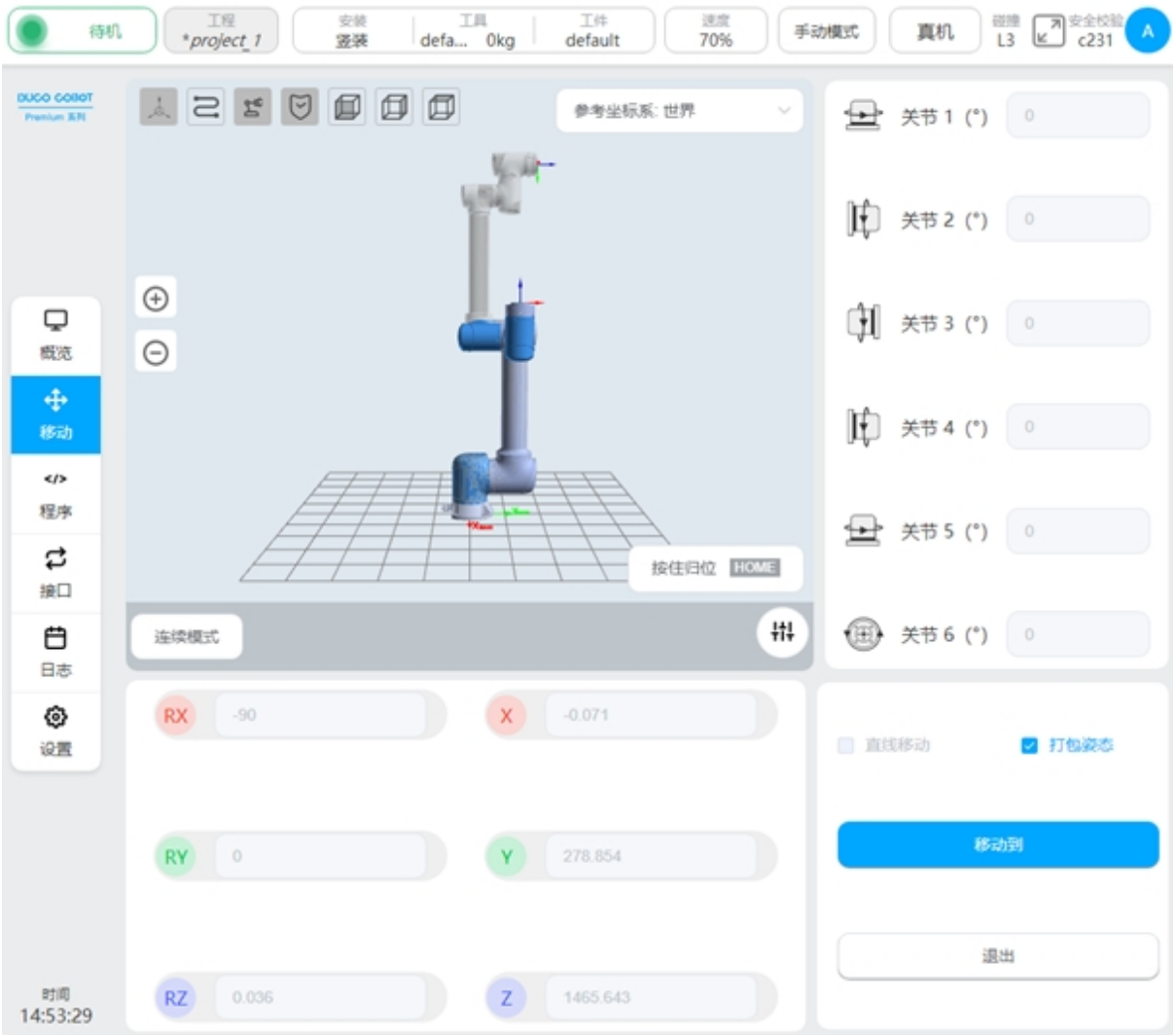
区域 ⑤ 是手动操作机器人末端在笛卡尔坐标系下移动的区域，同样当单击该区域任意地方，区域会显示蓝色边框，代表当前手动操作机器人末端移动。按住 X、Y、Z 轴方向按钮改变机器人在笛卡尔坐标系下位置，按住 RX、RY、RZ 方向按钮改变机器人在笛卡尔坐标系下的姿态，松开对应按钮机器人会停止移动。同样，用户可通过按住示教器上右侧“+”“−”按钮改变机器人的位置姿态，按钮从上到下分别对应 X、Y、Z、RX、RY、RZ 六个移动方向。

小心： 当使用示教器上的物理按钮 Jog 机器人时，务必根据当前激活的状态（蓝色边框），确认控制的是关节空间运动还是笛卡尔空间运动。

图标 ⑥ 是用户进行手动输入关节角度或位姿值和按住按钮操作机器人移动两种方式的切换按钮，单击“移动到”按钮，区域 ④ 和区域 ⑤ 会分别显示输入关节角度和位姿值的输入框，且 3D 模型区会显示一个目标点位置的灰透机器人模型。如图所示：



单击关节角度和位姿值的输入框，弹出虚拟数字键盘，在虚拟数字键盘中输入目标位置数据后，点击“Esc”键关闭虚拟数字键盘。然后，按住“移动到”按钮，机器人会移动到目标位置，松开该按钮机器人会停止移动。当勾选“直线移动”单选框，表示机器人是以直线移动到目标位置；否则，表示机器人以关节移动到目标位置。当勾选“打包姿态”单选框时，“直线移动”单选框会被禁用勾选，关节角度和笛卡尔位置姿态输入框分别被禁用输入，对于已勾选的“直线移动”单选框会被取消勾选，关节角度和位置姿态值会显示打包姿态时固定值，表示机器人会以关节移动到打包位置，单击“退出”按钮，会退出手动输入目标位置的界面，回到原来的移动界面。如图所示：



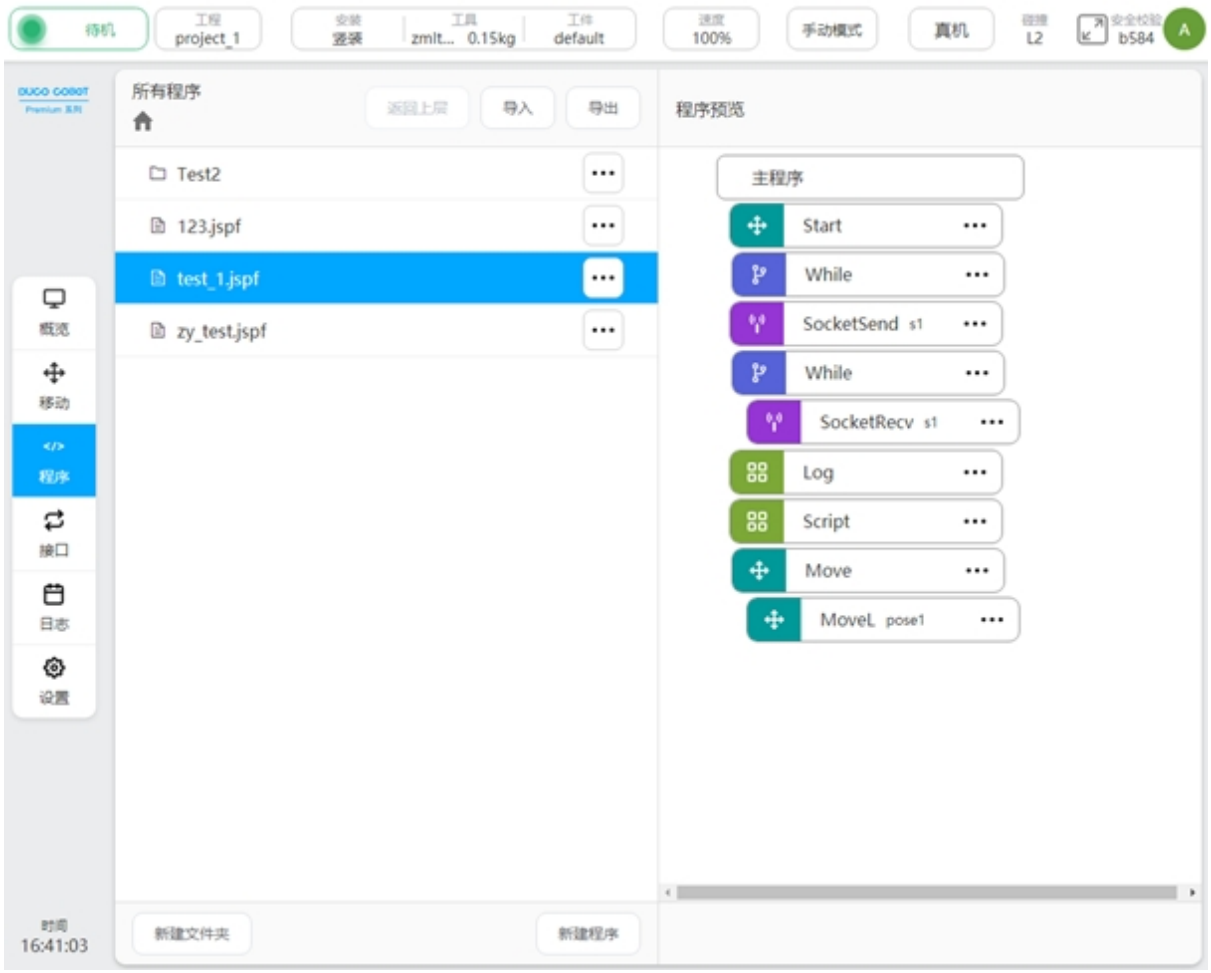
图标 ⑦ 是“工具偏移复位”按钮，只有当选择移动页面参考坐标系为工具坐标系时，才会显示该按钮。用户单击该按钮时，会以单击时刻的位置为参考起始点，在操作机器人运动时会在区域 ⑤ 实时显示机器人位置参考单击该按钮时的参考起始点的偏移量。直到用户再次单击“工具偏移复位”按钮才会更新 TCP 偏移量的计算参考起始点。

2.9 程序页面

点击导航栏“程序”进入程序页面。程序页面可以管理程序列表、编辑程序、运行程序。

2.9.1 程序列表页

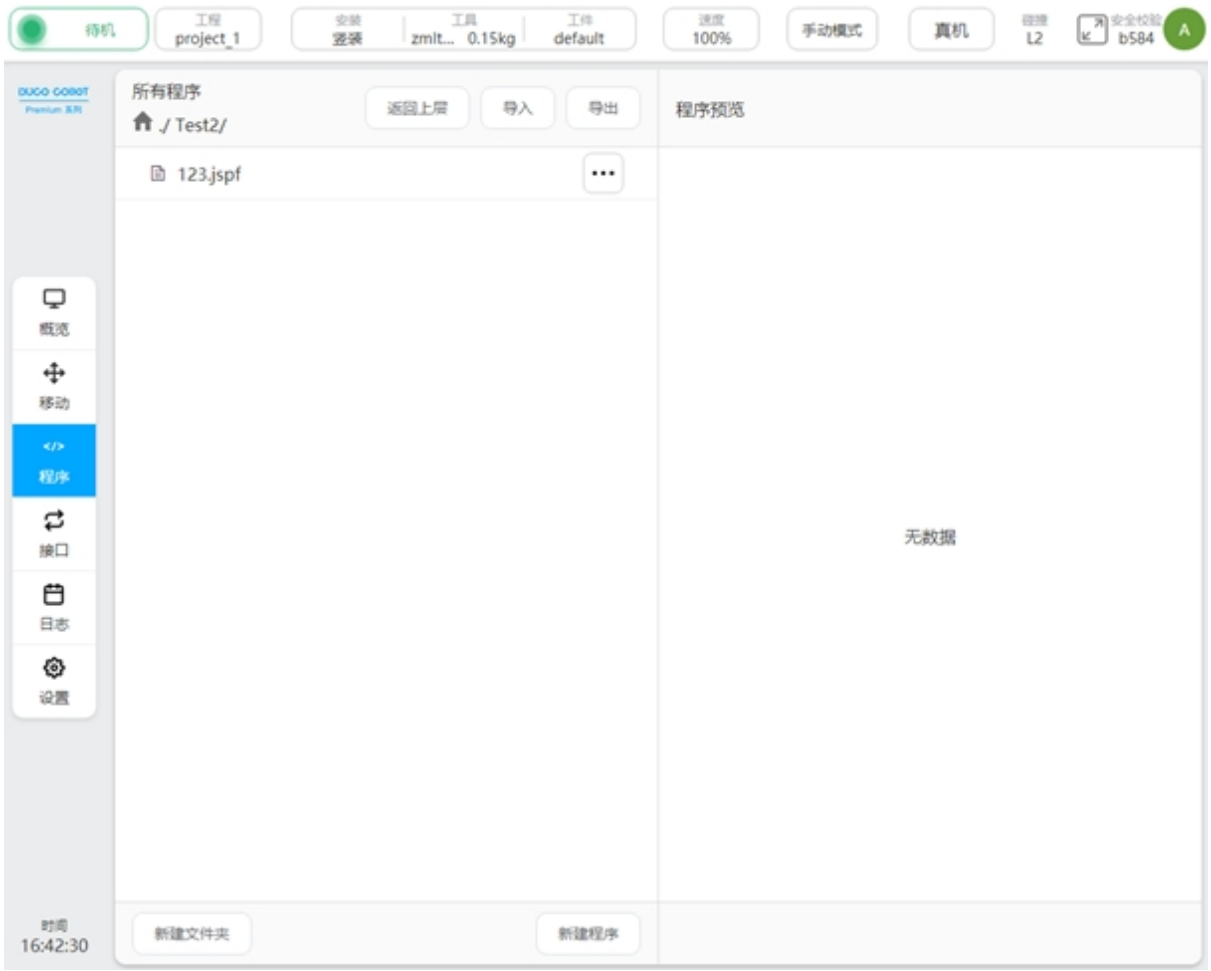
列表页用来管理当前激活工程内的所有程序。如图为程序列表页面，显示了程序根目录下的所有程序及文件夹。



选中文件夹，点击右边的 ••• 按钮，点击“进入程序”，可切换到该文件目录，查看该文件夹下的所有程序及文件夹。

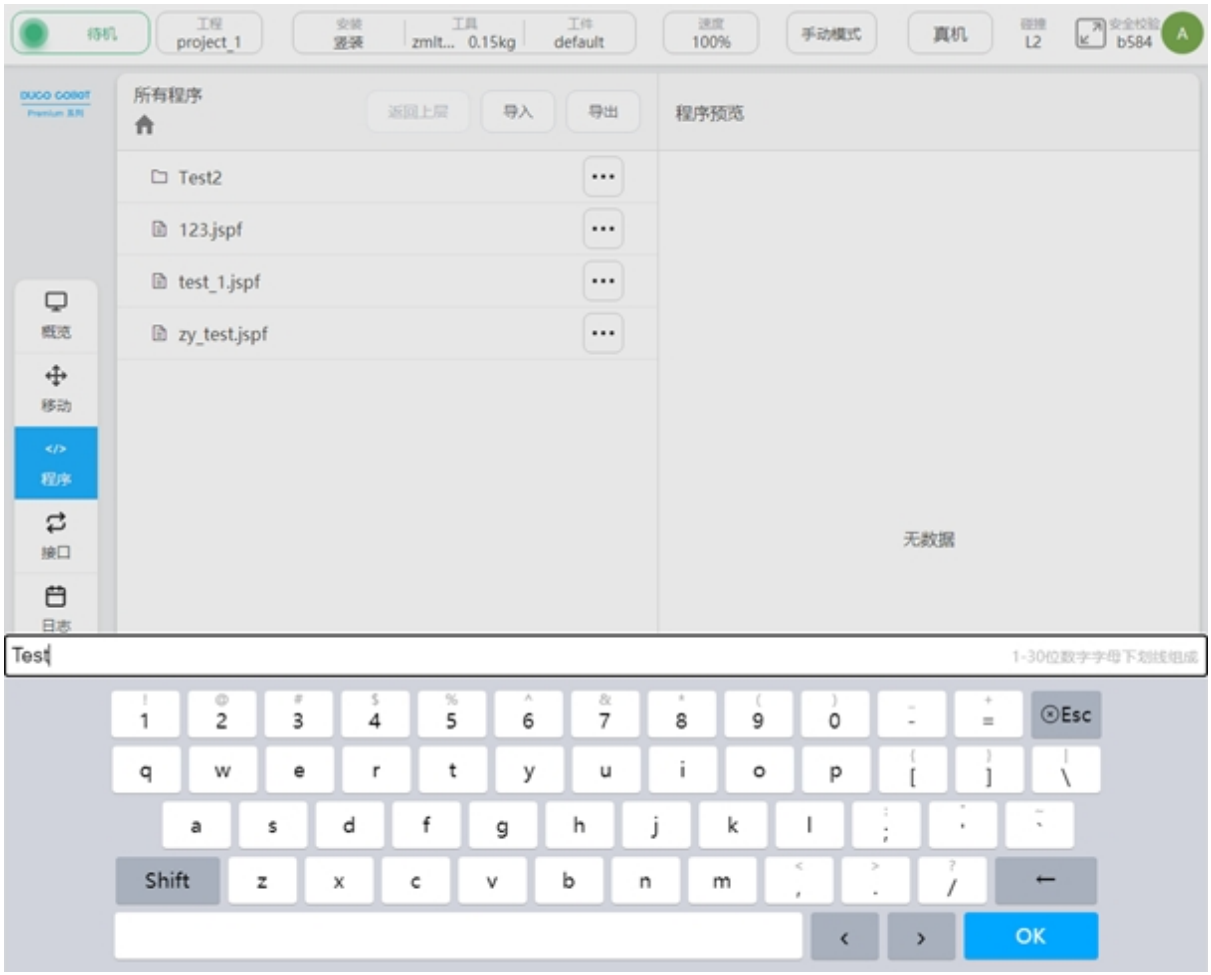
选中程序，右侧“程序预览”区可以显示该程序的内容，点击“进入程序”可打开程序。

左上方可显示当前所在的目录，如图。点击“返回上层”按钮可返回上层目录，点击路径可切换目录，点击  可切换到根目录。



新建程序

点击“新建程序”按钮，可弹出键盘，输入程序名，点击“OK”即在当前文件夹下创建了一个程序。

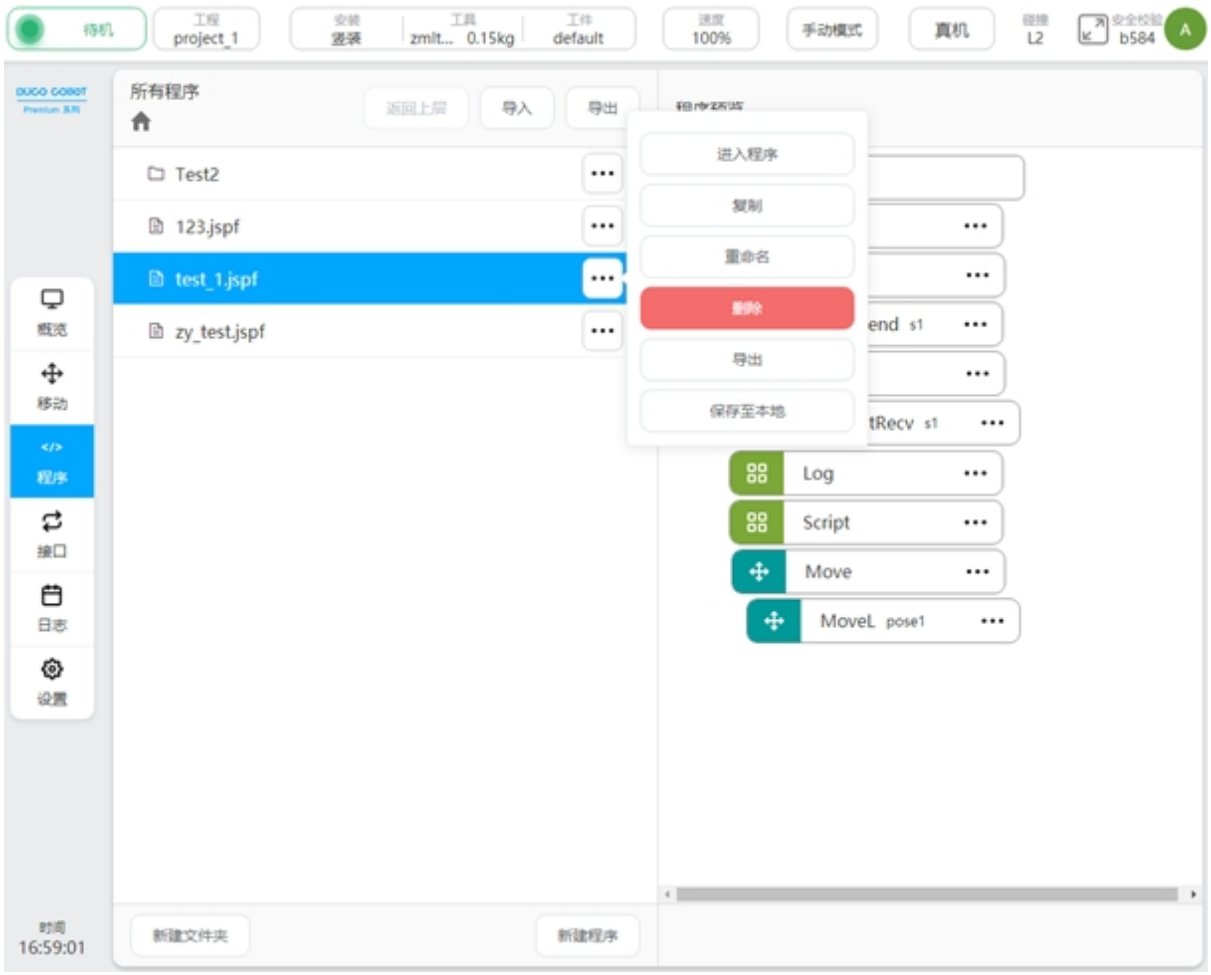


新建文件夹

点击“新建文件夹”按钮，可弹出键盘，输入文件夹名，点击“OK”即在当前文件夹下创建了一个文件夹。

程序文件操作

点击列表项右边的 ••• 按钮，可弹出菜单框如下图所示：



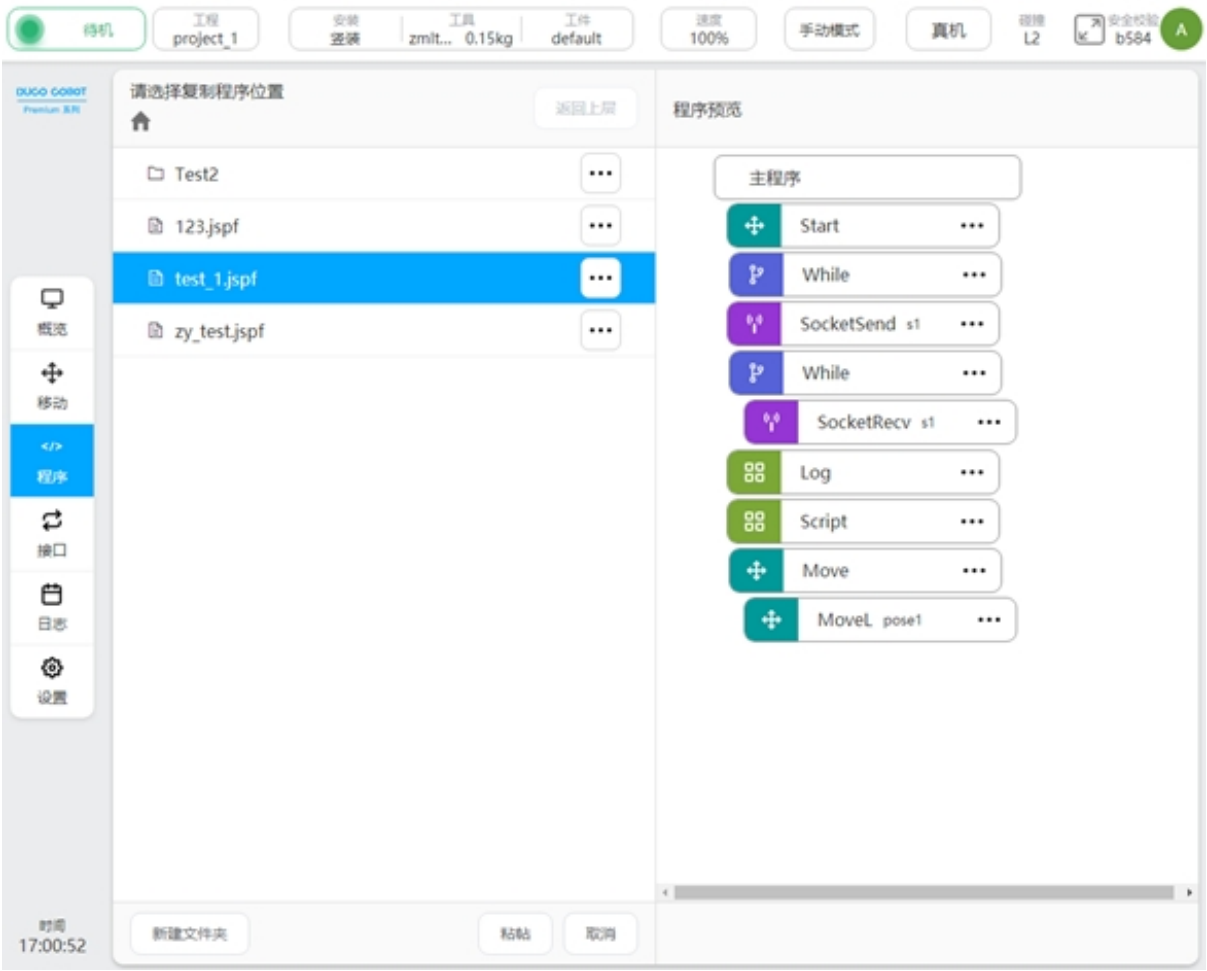
对于程序文件来说，可执行如下操作：

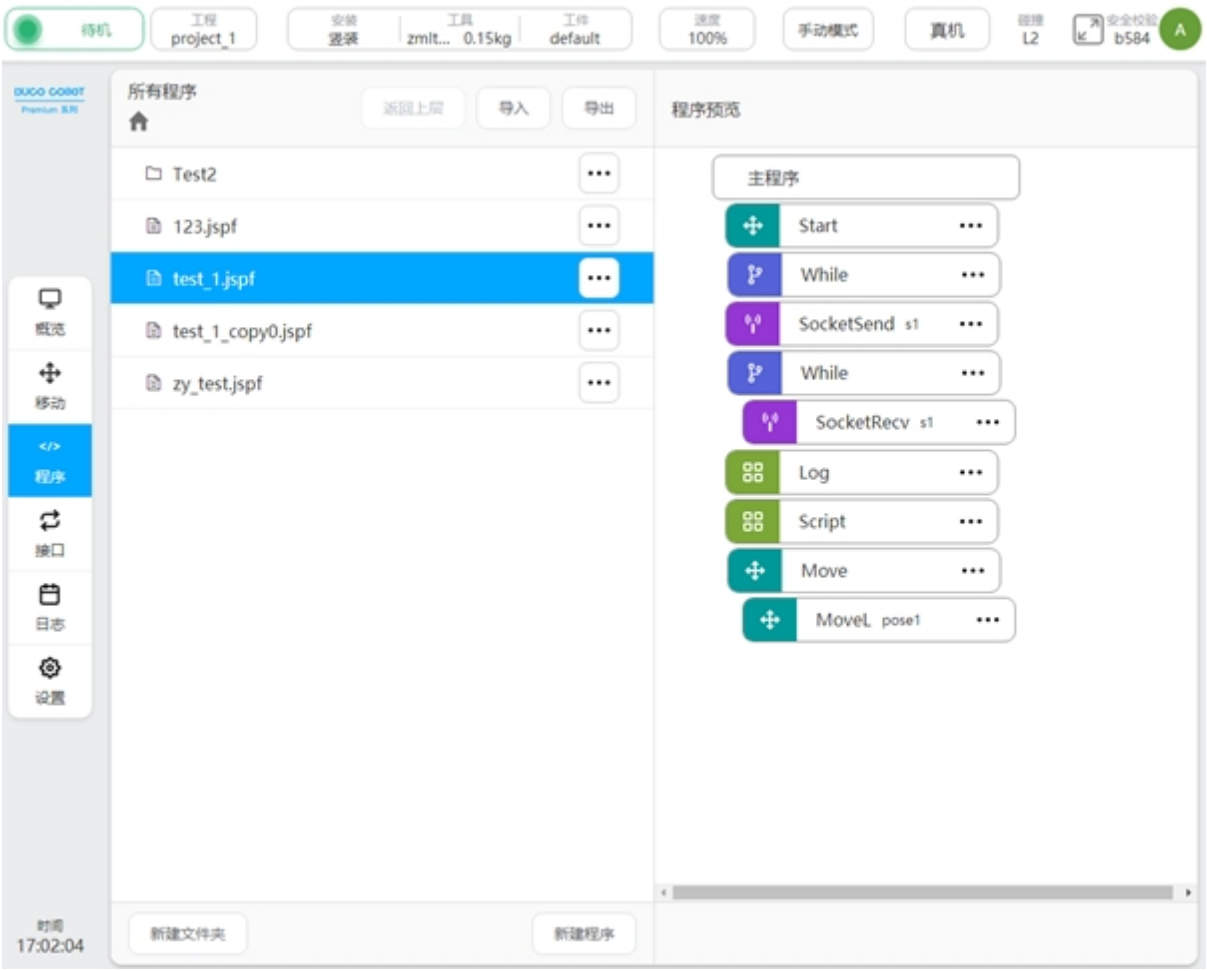
进入程序

点击可打开程序。

复制

点击“复制”按钮，将复制该程序文件，程序列表页面如下，此时可选择粘帖的位置，点击“粘帖”按钮，即可将该程序复制到当前文件夹，点击“取消”按钮取消复制。粘帖时，若当前文件夹有同名文件，将自动在后面加“_copyXX”后缀

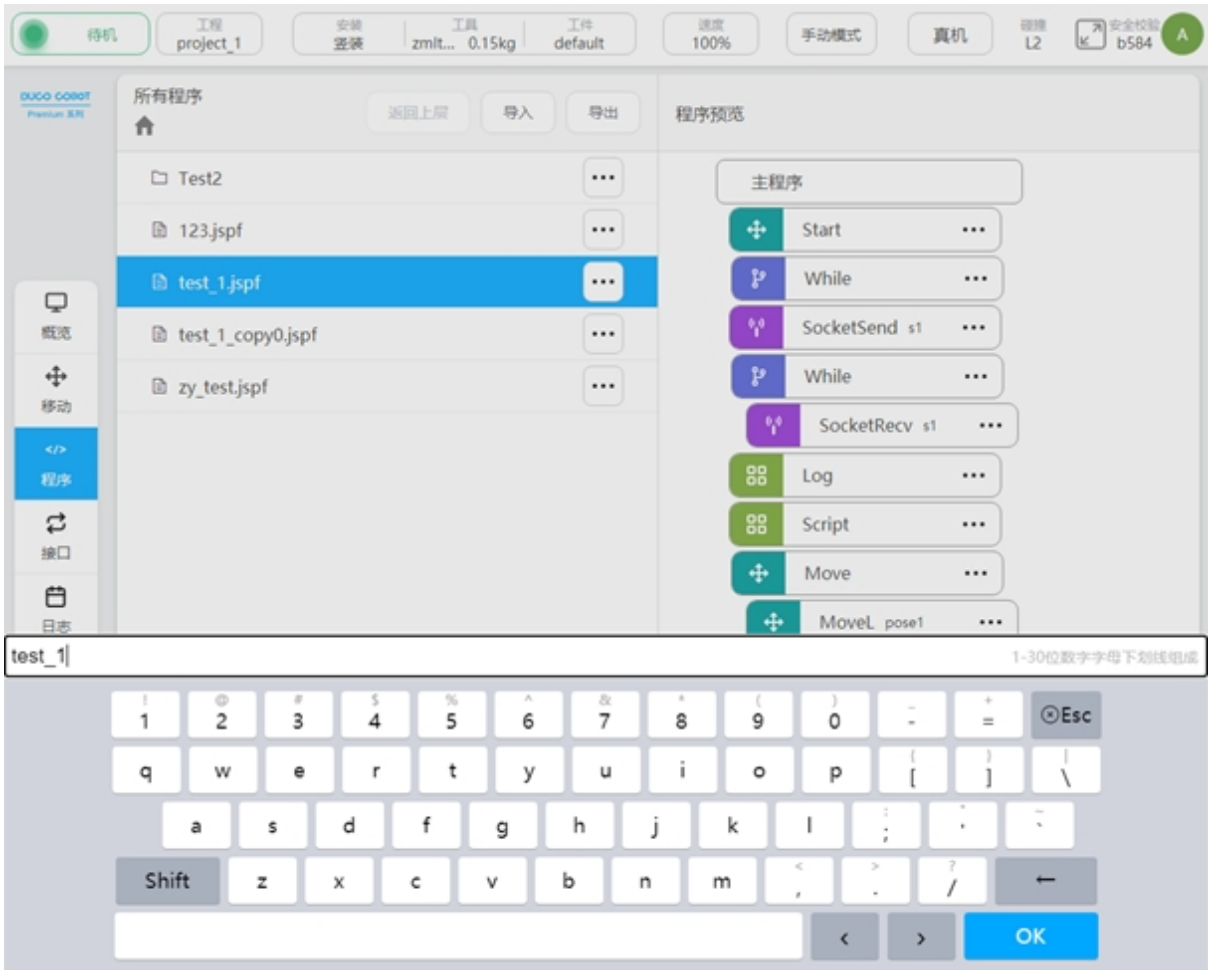

待机
project_1
竖装
zmlt... 0.15kg
default
100%
手动模式
真机
L2
b584
请选择复制程序位置
返回上层
程序预览
Test2
123.jspf
test_1.jspf
zy_test.jspf
主程序
Start
While
SocketSend s1
While
SocketRecv s1
Log
Script
Move
MoveL pose1
概览
移动
程序
接口
日志
设置
时间
17:00:52
新建文件夹
粘贴
取消



重命名

点击“重命名”按钮，会弹出键盘，输入新的名称后，确定即可修改。

备注： 注意：已经打开的程序或正在运行的程序不能重命名。



删除

点击“删除”按钮，弹出确定对话框，确定后即可删除该程序

备注： 注意：已经打开的程序或正在运行的程序不能删除。

导出

可以将程序文件导出到 U 盘上，详见程序的导入导出

保存至本地

可以将程序文件通过浏览器下载到本地

文件夹操作

点击文件夹项右边的按 ••• 钮，弹出的菜单可执行如下操作：进入程序、重命名、导出文件夹、删除。

进入程序

点击可切换到该文件夹，查看该文件夹下的所有程序及子文件夹。

重命名

可进行文件夹的重命名。注意：若该文件夹下有已打开的程序或正在运行的程序，则不能重命名。

导出文件夹

可以将整个程序文件夹压缩成压缩包导出到 U 盘。

删除

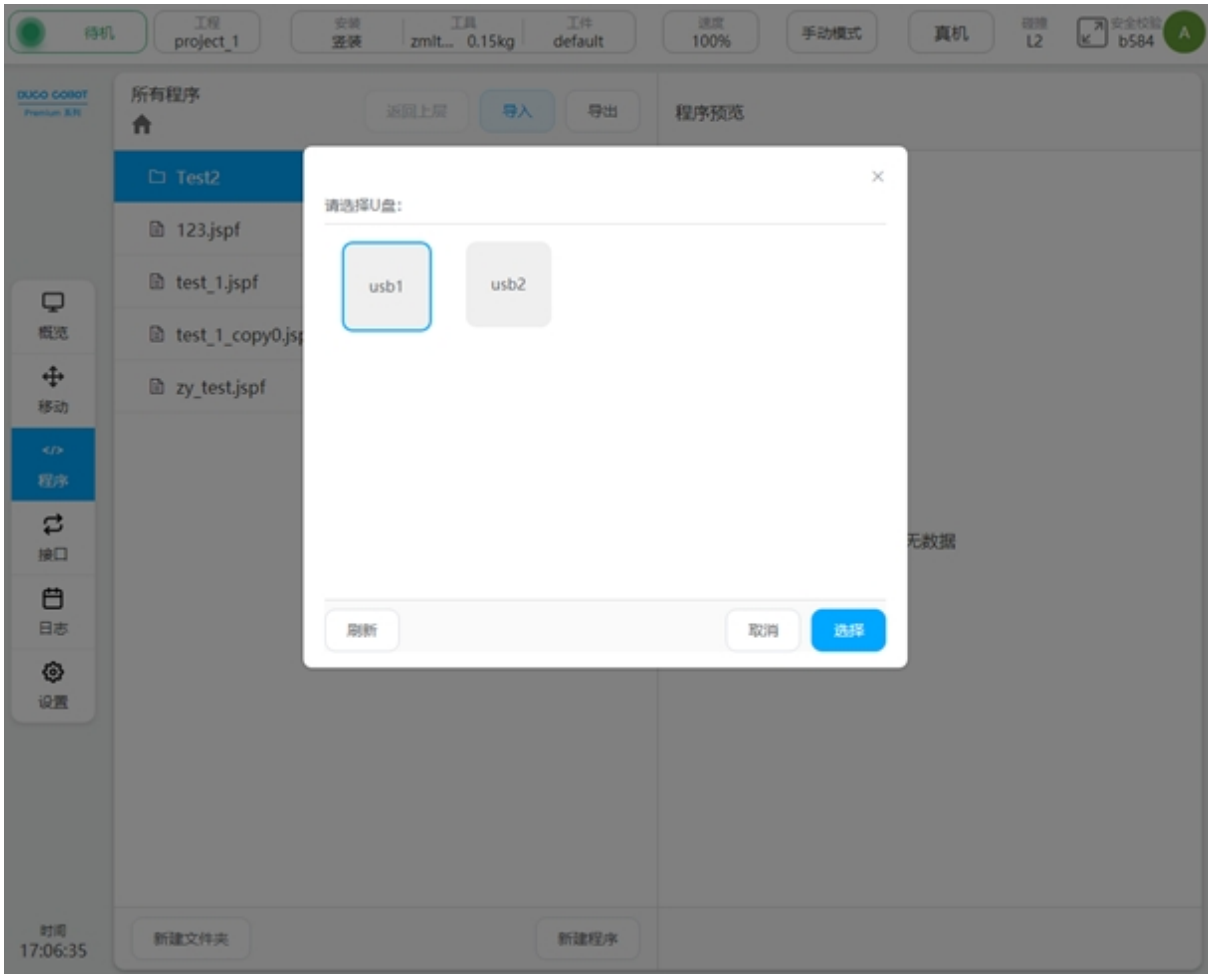
可删除文件夹下的所有内容。如该文件夹下有已打开的程序或正在运行的程序，则不能删除。

程序的导入导出

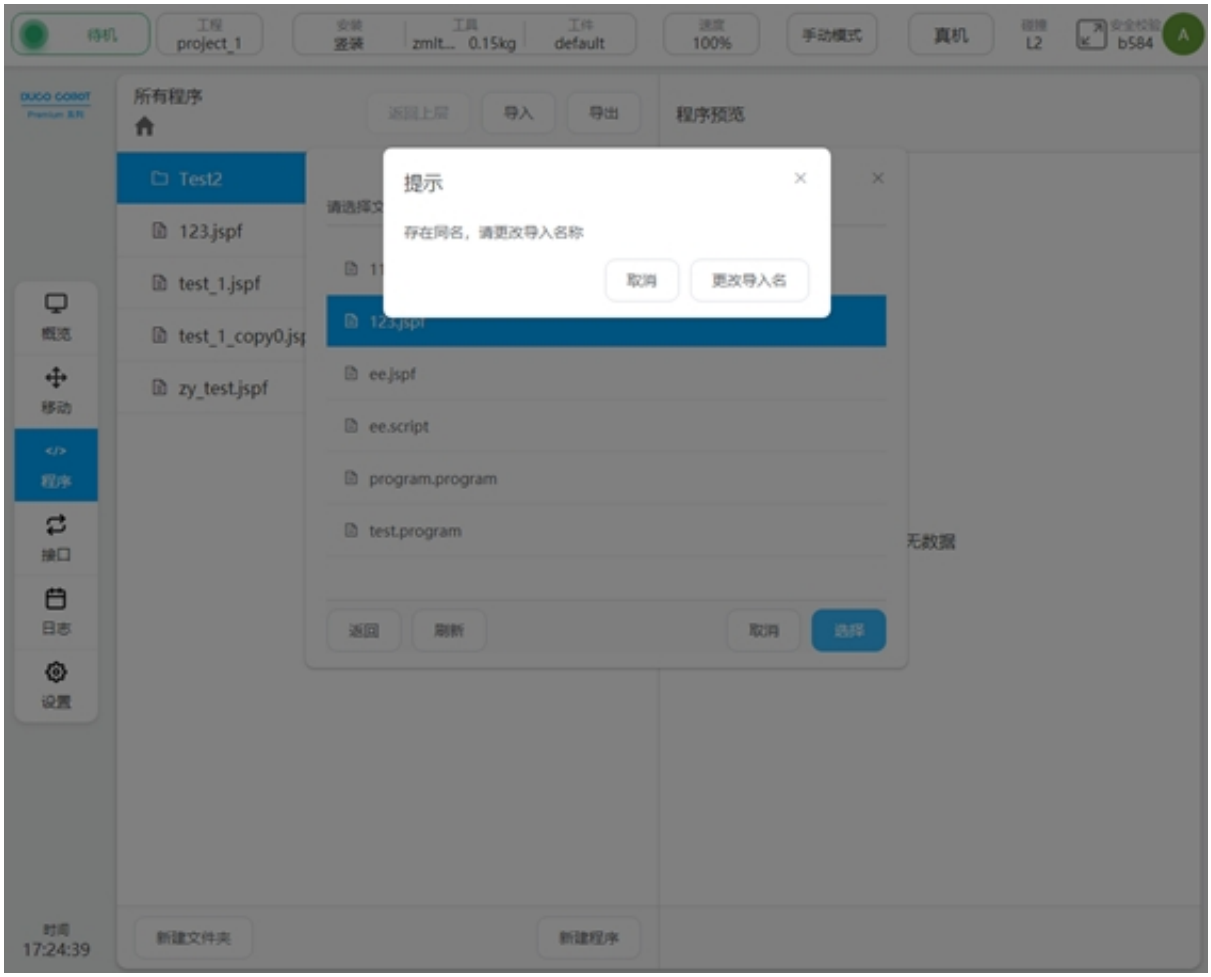
支持从外部设备导入程序、文件夹或者导出程序、文件夹到外部设备中，外部设备指接入控制柜 USB 端口的存储设备，如 U 盘、移动硬盘等。

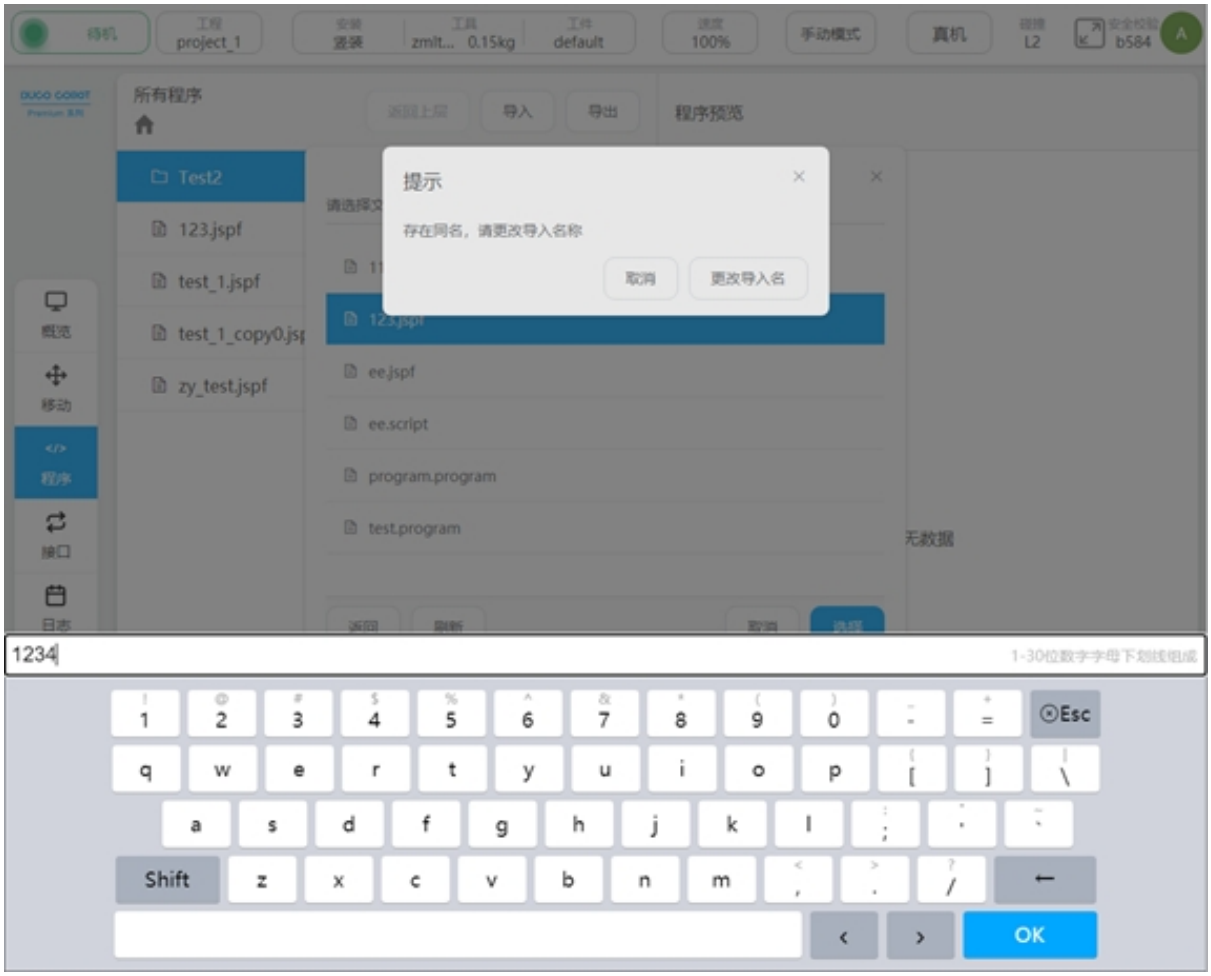
程序/文件夹导入

点击“导入”按钮，弹出以下窗口，显示当前控制柜上挂载的 USB 存储设备，选择一个设备。

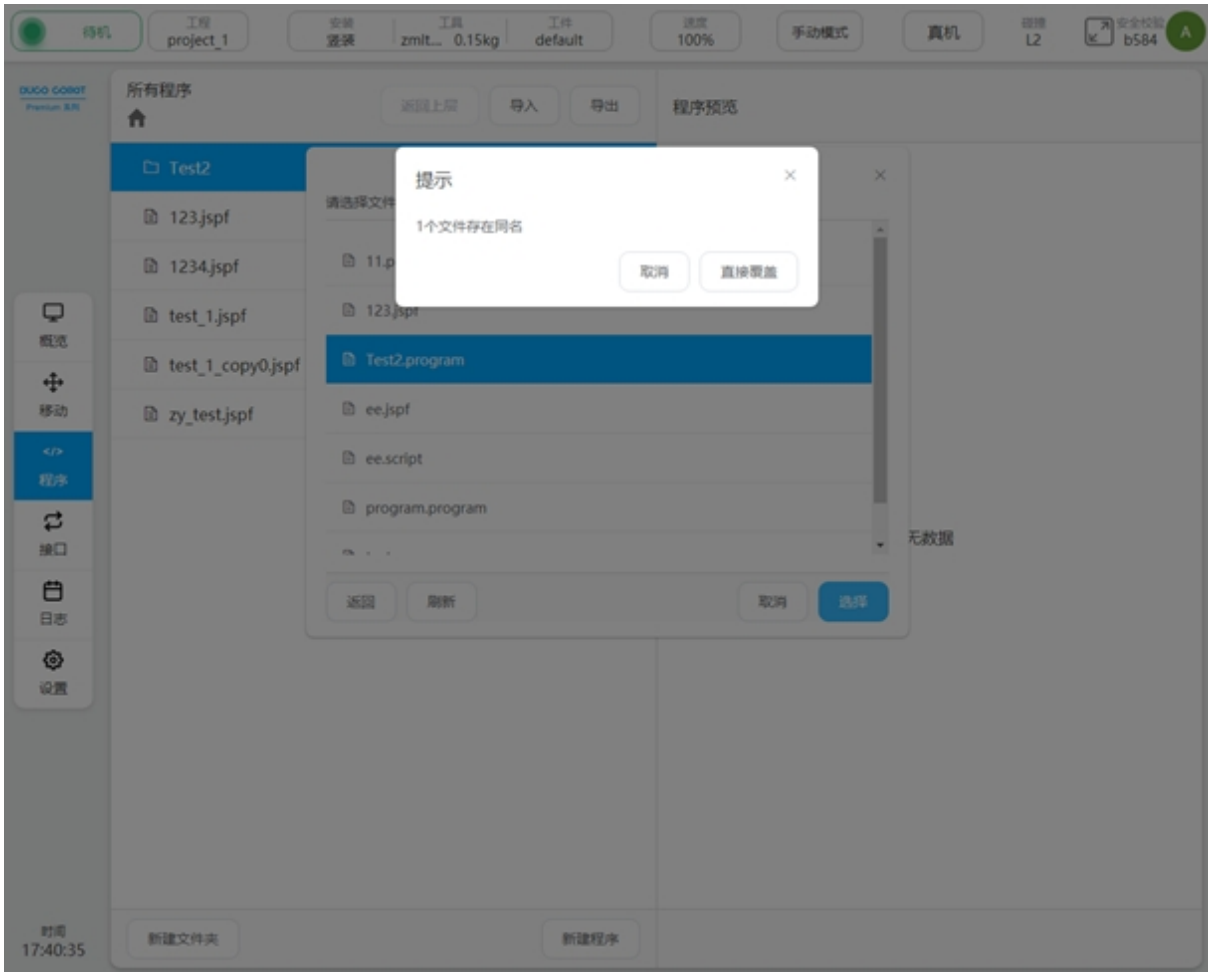


页面上将显示该设备中的文件夹及符合条件的文件（后缀名为 jspf 或 script 或 program），选中一个文件，点击“选择”按钮，则会将该文件或文件压缩包（后缀名为 program）解压后，导入到控制柜中。当导入文件时，若检查到与控制柜上的程序文件刚好重名，则会弹出重命名提示框，重命名后导入到控制柜中。


待机
project_1
0.15kg
default
100%
手动模式
真机
L2
所有程序
返回上层
导入
导出
程序预览
Test2
123.jspf
test_1.jspf
test_1_copy0.jsp
zy_test.jspf
提示
存在同名，请更改导入名称
取消
更改导入名
ee.jspf
ee.script
program.program
test.program
无数据
返回
刷新
取消
选择
概览
移动
程序
接口
日志
设置
时间
17:24:39
新建文件夹
新建程序



当导入文件夹（后缀名为 program）时，有重名文件，会提示重名文件数量，用户可操作“取消”或“直接覆盖”。



程序/文件夹导出

在程序列表页选中一个程序，在其交互操作菜单中选择“导出”，则将执行导出操作。与导入类似，此时将弹出窗口显示当前控制柜上挂载的 USB 存储设备。选中设备后，可以查看该设备的文件及文件夹，选择导出的文件夹，确定后即可将程序导出到设备的该文件夹下。若导出的程序文件与设备中的文件刚好同名，则会弹出如下窗口。用户可选择取消导出，或者修改导出名称后重新导出，或者直接覆盖设备中的文件。若选择的是文件夹，则会以该文件夹名称为命名，打包文件夹并压缩成“文件夹名.program”，该压缩包包含本级文件夹。

点击程序列表页上方“导出”按钮，会将当前目录下的所有文件与文件夹打包到一个压缩包中，压缩包名称为“当前程序列表名.program”并导出到外部设备的指定路径下。该压缩包不包含本级文件夹。与导出程序类似，存在同名文件时，同样会弹同名操作提示框。

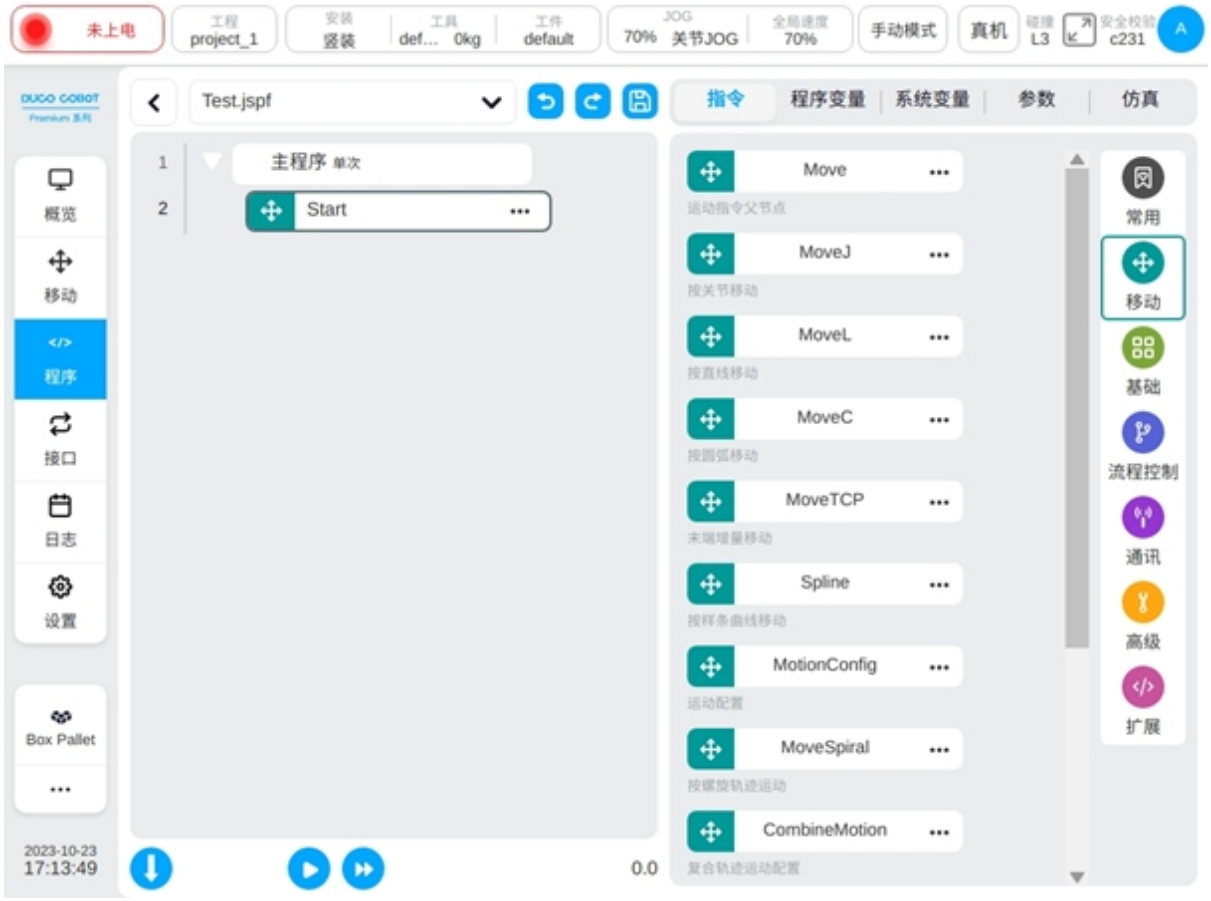
2.9.2 编程

整体布局

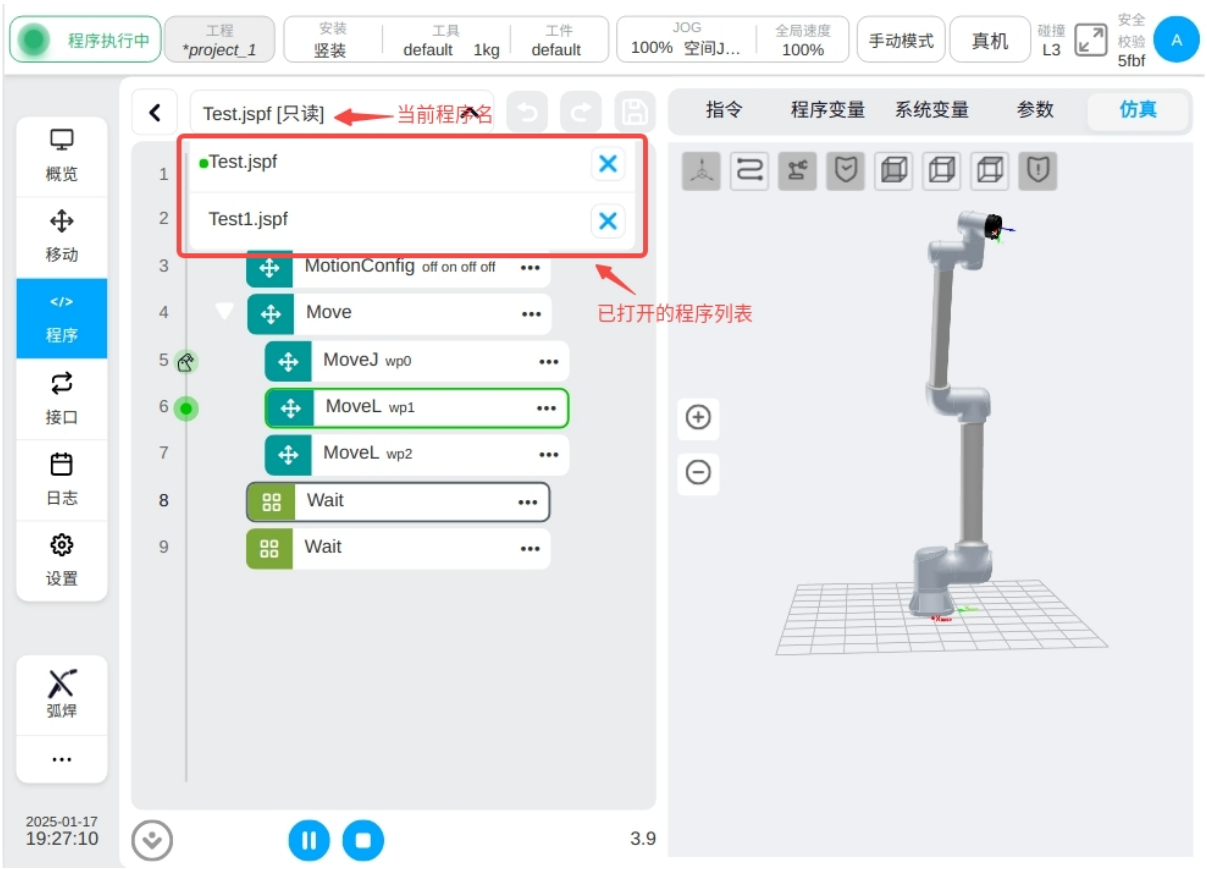
在程序列表页新建程序或者打开程序后，可进入编程页面，如下图。

编程页面左侧为编程区，可进行程序树的编辑。右侧可分为五个部分：“指令”标签页是功能块列表区，显示了可用的图形化编程功能块；“程序变量”标签页是程序变量区，可创建程序变量，程序运行时，监控程序变量，还可对程序变量名称按照字母进行排序，以及选择是否进行变量监控；“系统变量”标签页是显示系统变量的初始值，以及程序运行时监控系统变量的当前值，同样地可对系统变量名称按照字母进行排序；“参数”标签页是参数配置区，查看及编辑程序树中选中的功能块参数；“仿真”标签页是显示机器人的 3D 模型。

其中，点击“指令”标签页中“移动”、“基础”、“流程控制”、“通讯”、“高级”、“扩展”任意一类功能块列表区的功能块右边 ••• 图标可弹出操作对话框，点击该操作对话框中“添加到常用”的按钮，可将该指令块添加到“常用”类；点击“常用”类功能块列表区的功能块右边 ••• 图标同样会弹出操作对话框，点击弹出操作对话框中“移除”按钮，可将该指令块移除“常用”类。

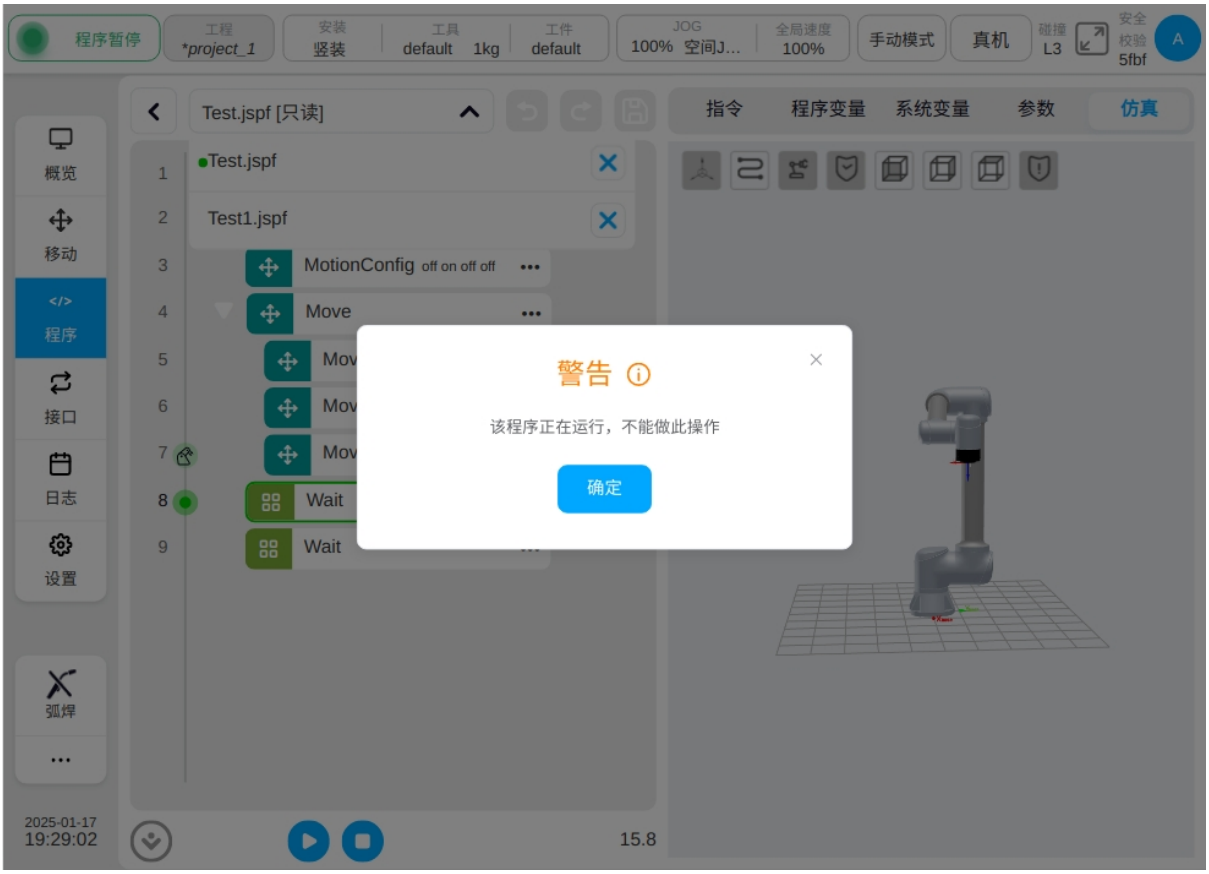


左侧上方显示当前程序名称，程序名称后面的 * 表示该程序有改动。点击程序名所在显示框任意区域，显示框下方会出现下拉框显示已打开的程序列表。

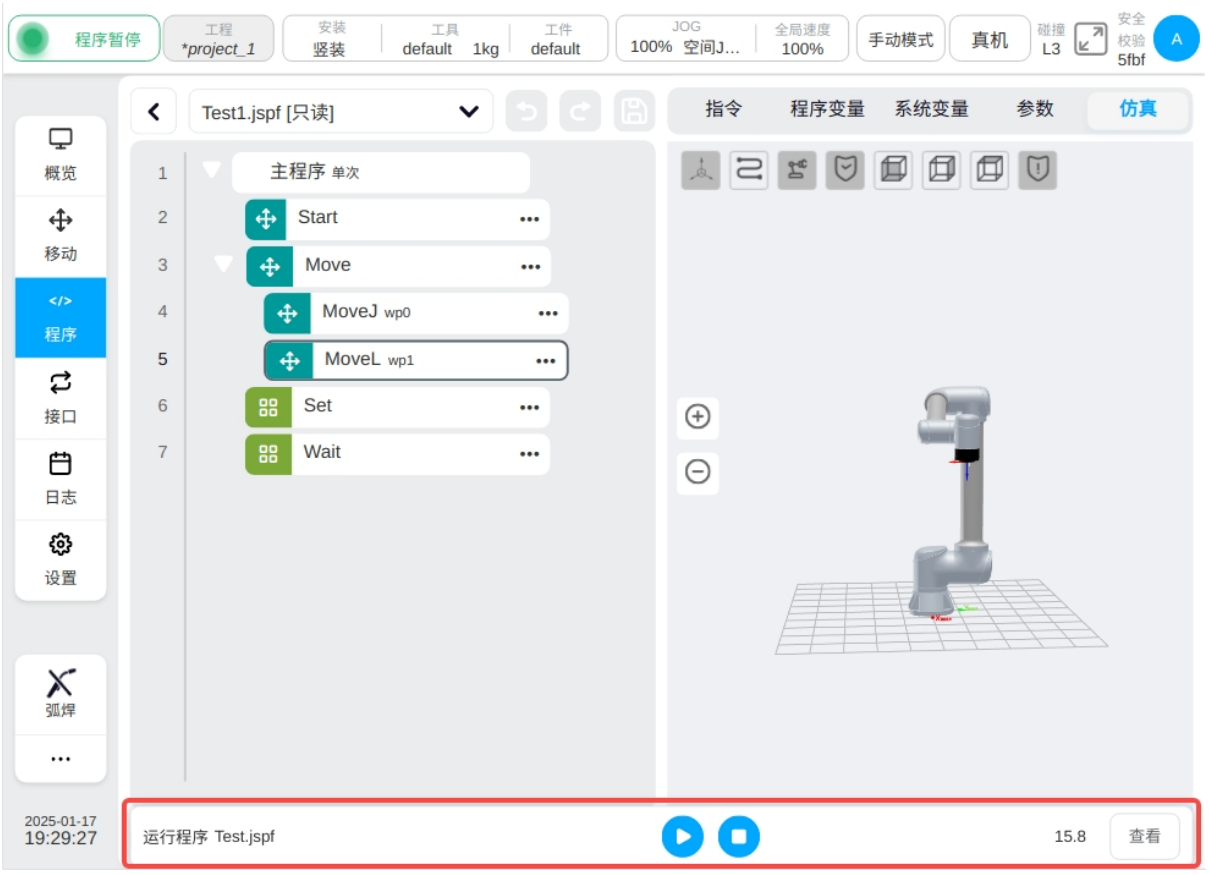


如上图所示，已打开的程序列表显示了当前系统中有两个打开的程序，其中，程序名前显示绿色实心圆的程序是正在运行的程序。点击下拉框里名称右侧的 按钮可以关闭该程序；点击下拉框列表项栏可切换至对应程序。

备注： 注意：正在运行的程序不可操作关闭程序！会弹窗提示如下图。



关闭程序时，若程序有改动，会弹出对话框提示保存程序或者放弃修改。切换查看程序运行过程中的其他程序时，页面下方显示如下图所示。



左侧显示当前正在运行的程序名称，中间是程序暂停、继续、停止等操作按钮，点击右侧“查看”按钮可返回当前正在运行程序。

编程操作

编程采用拖拽和双击添加的方式。

拖动的方式： 可以按住功能块列表区的功能块，拖动到编程区，程序树对应的位置将会显示蓝色横线表示功能块将插入该处。松开即在该处添加了一个功能块。当拖动功能块到程序树对应位置显示红色横线时，表示该功能块不能插入该处，且松开后会弹框提示。

双击添加： 点击程序树的任一功能块可选中该功能块，此时双击功能块列表区中的功能块，列表区中的功能块将添加到程序树选中功能块的下方。（注意：移动设备端不支持双击操作）

支持直接拖动程序树中的功能块到其他位置。

选中某个功能块，手动切换到参数配置区，配置该功能块的参数，各个功能块及其对应的参数配置页面参见后面内容。

程序树中的功能块有以下五种状态，红色代表该功能块未配置有效的参数；斜体加 * 代表该功能块的参数发生改动，但是未保存到程序树中；黑色正体代表该功能块是参数有效、无改动；功能块部分灰显代表已选中可进行拖拽的状态；功能块全部灰显代表该功能块设置为不启用。

点击程序树功能块右边 ••• 可弹出如下操作对话框。包含以下操作

剪切： 剪切该功能块。

复制： 复制该功能块。

粘帖： 可以将复制的功能块粘帖到选中功能块的下方。

不启用/启用： 点击可以选择是否启用该功能块，选择不启用时，该功能块灰显，程序不会执行该功能块。

运行此行： 点击可以直接运行选中的程序行。

备注： 注意：使用运行此行功能时，由于所选中程序之前的程序不会被执行，因此若所选中程序中包含变量，变量的值会以使用运行此行功能时变量当前值为准。

批量操作： 可以进行功能块的批量复制、批量剪切、创建组、批量启用、批量禁用、批量删除。

选中任意一个功能块批量操作，该功能块行号处会出现  被选中图标，且与该功能块属于同一个层级的功能块行号处出现多选框图标，选中其中任意一个功能块，会自动选中该功能块和初始功能块之间的所有功能块。且，只能选中同一层级的功能块，不能选中跨越父级的功能块。

批量复制：点击后，选择需要粘贴的位置，粘贴

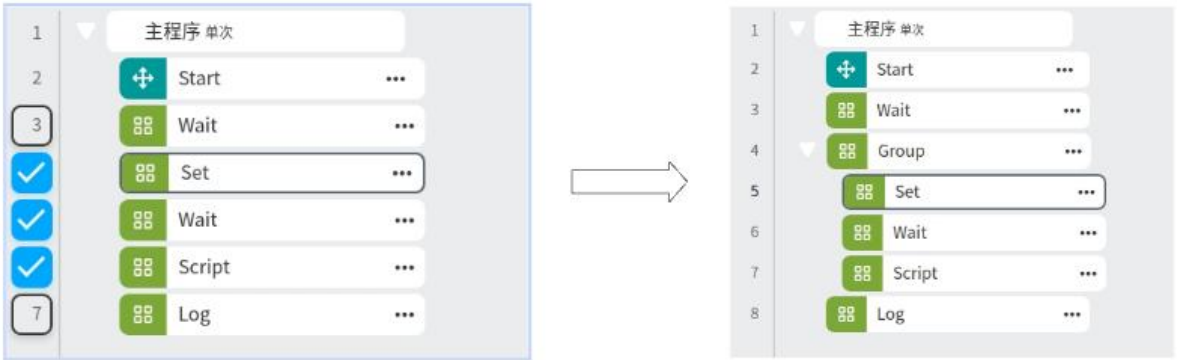
批量剪切：点击后，选择需要粘贴的位置，粘贴

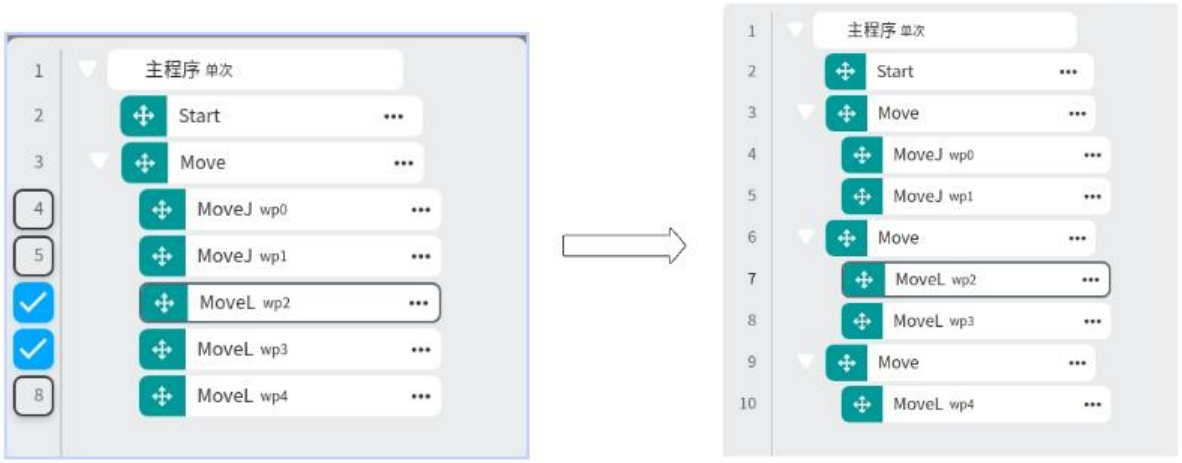
批量启用：对选中的功能块都启用

批量禁用：对选中的功能块都禁用

批量删除：删除选中的功能块

创建组：将选中的功能块创建为一个组：若选中的都为 Move 子块，则将创建新的 Move 块，Move 块参数和原 Move 块参数一致；否则将创建一个 Group 块





删除： 可以删除该功能块。

选中此行为运行起始行： 点击可以选中该功能块为运行起始行，此时该功能块的前方将会有图标 此时运行程序，程序将从该功能块处往下执行。点击右侧上方的“取消选中行”可以取消起始行的设定。

在程序编辑过程中，可随时通过页面左侧上方的 和 图标来撤销修改或者恢复修改。注意：撤销修改最多支持撤销 10 次。

程序树一旦发生了改动，当前程序名称后面会显示 *，表示该程序有改动，此时可以通过页面左侧上方 图标将程序保存到控制器。

2.9.3 功能块及参数配置

功能块列表区显示了当前可用的图形化编程功能块，这些功能块根据机器人编程常用的场景封装而来。本系统提供的功能块及其配置参数如下。

Start 功能块

机器人初始关节位置设置。当开启该功能块，程序运行时，会进行初始关节位置判断。若禁用该功能块，则会从当前所在的位置直接运行程序。

建议主程序始终开启该功能块，子程序可根据情况关闭该功能块。

移动功能块

在拖动 MoveJ、MoveL 等运动类功能块到程序树中，若没有 Move 节点，将自动创建一个 Move 功能块。这个功能块可以用来批量设置其子节点的运动参数和坐标系，含有如下参数：

末端速度：单位 mm/s，取值范围为 [0.01,5000]；

末端加速度：单位 mm/s2，取值范围为 [0.01,Infinity]；

关节角速度：单位 °/s，取值范围为 [0.01,225]；

关节角加速度：单位 °/s2，取值范围为 [0.01,Infinity]；

融合半径：单位 mm，取值范围为 [0,Infinity]；

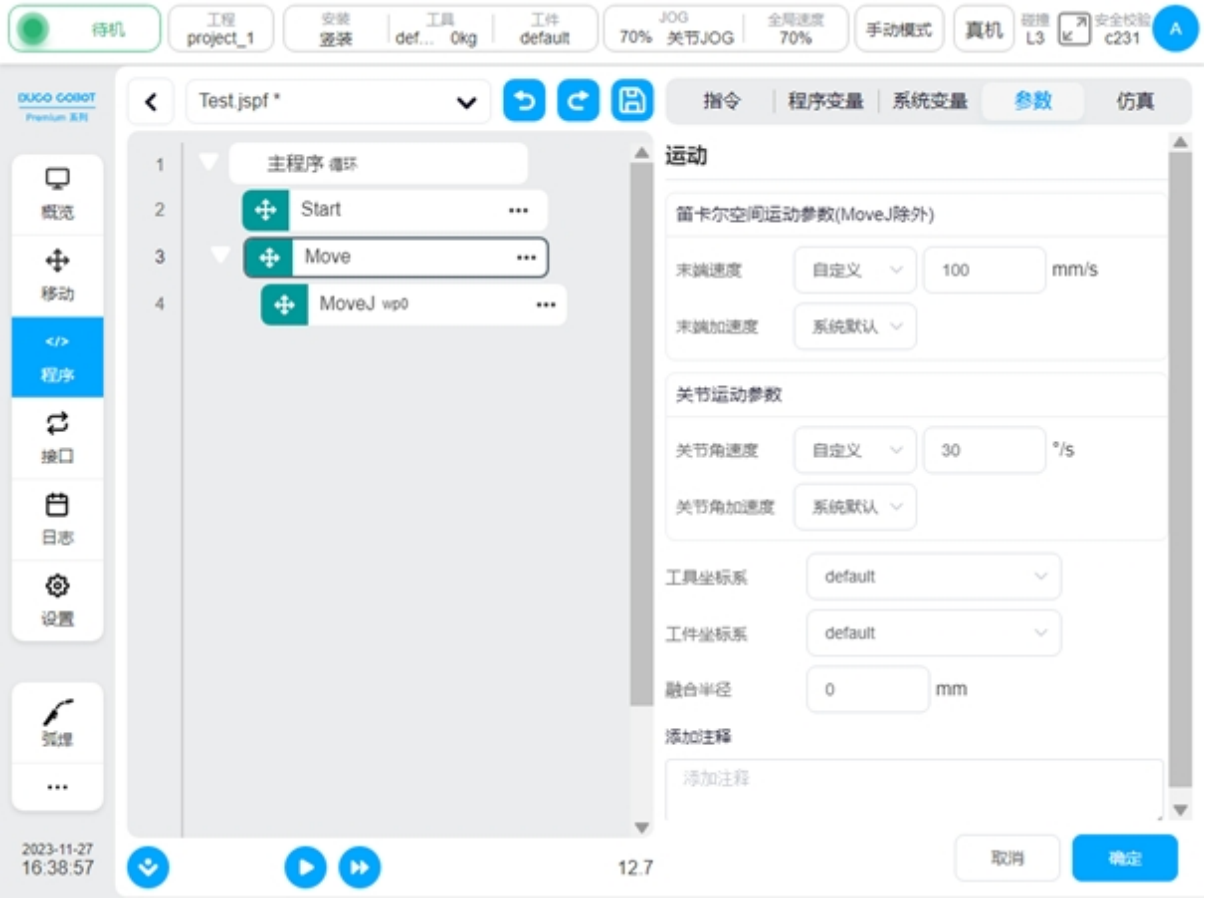
参考坐标系：工具坐标系、工件坐标系，可手动更改。

其中，末端速度、末端加速度、关节角速度和关节角加速度均可以使用自定义直接输入数值的方式或选择变量，末端加速度和关节角加速度还可以选择系统默认。当末端加速度或关节角加速度选择为系统默认时，运动规划将采用算法计算的最优规划。注意：“自定义”时才会显示对应参数的单位；选择变量时，只能选择特定类型变量，末端速度只能选择 pose_speed 类型变量，末端加速度只能选择 pose_acc 类型变量，关节角速度只能选择 joint_speed 类型变量，关节角加速度只能选择 joint_acc 类型变量。

在生成 Move 功能块时，默认将获取 Move 的所有父节点是否有设置参考坐标系，若无将使用系统当前设置的坐标系。例如传送带功能块有设置的参考坐标系，即标定的传送带坐标系，所以拖动 Move 类功能块到传送带子节点时，生成的 Move 功能块默认使用传送带的坐标系。

Move 功能块设置的笛卡尔空间运动参数只针对 MoveL、MoveC、MoveTCP、MoveWobj、MoveSpiral 功能块，对其他子功能块无影响；设置的关节运动参数只针对 MoveJ 功能块，对其他子功能块无影响。

Move 功能块设置的参考坐标系只针对 MoveJ、MoveL、MoveC、MoveTCP、MoveWobj、MoveSpiral 功能块，对其他功能块无影响。



具体的控制机器人运动的功能块，有如下类型：

MoveJ

机器人按照关节运动的方式移动，可以选择移动到目标关节或者目标位置姿态。可设参数：

目标关节：目标使用关节位置，可以通过示教的方式设置或者设置为变量，示教设置后可手动更改；

目标位置姿态：目标使用末端位姿，可以通过示教的方式设置或者设置为变量，示教设置后可手动更改；

示教：记录目标点位，点击将跳转到移动页面，在移动页面移动机器人后，点击“记录当前关节”按钮，将跳回程序页面，并设置点位；

记录当前点：点击后，将当前点位设置为目标点；

移动到此点：点击“移动到此点”，程序页面右侧将跳转到“仿真”页面。在仿真页面右下方将出现“按住移动”按钮。功能与概览页面的“按住移动”相同，详情请参见概览页面。

备注： 注意：当使用目标位置姿态时，从不同的机器人初始位姿使用“移动到此点”功能，会出现最终位姿一致，但机器人构型不一致，即关节位置不一致。可以通过观察灰透 3D 模型判

断构型是否正确

备注： 注意：当使用目标位置姿态时，若脚本运行过程中运行到此脚本时机器人实际关节位置与示教时机器人的关节位置处于不同的选解空间，则会报错“机器人当前关节位置与试教参考关节位置选解不一致”

使用父节点坐标系：选择目标位置姿态时可设，勾选时，该功能块使用父节点 Move 功能块设置的参考坐标系，默认勾选；

参考坐标系：选择目标位置姿态时可设，不勾选使用父节点坐标系时，可单独为该功能块设置其参考坐标系；

使用父节点参数：勾选时，该功能块使用父节点 Move 功能块设置的关节角速度、关节角加速度参数；不勾选时，需要单独为该功能块设置关节角速度、关节角加速度，默认勾选；

关节角速度：单位 °/s，可以自定义或选择变量，自定义取值范围为 [0.01,225]；

关节角加速度：单位 °/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

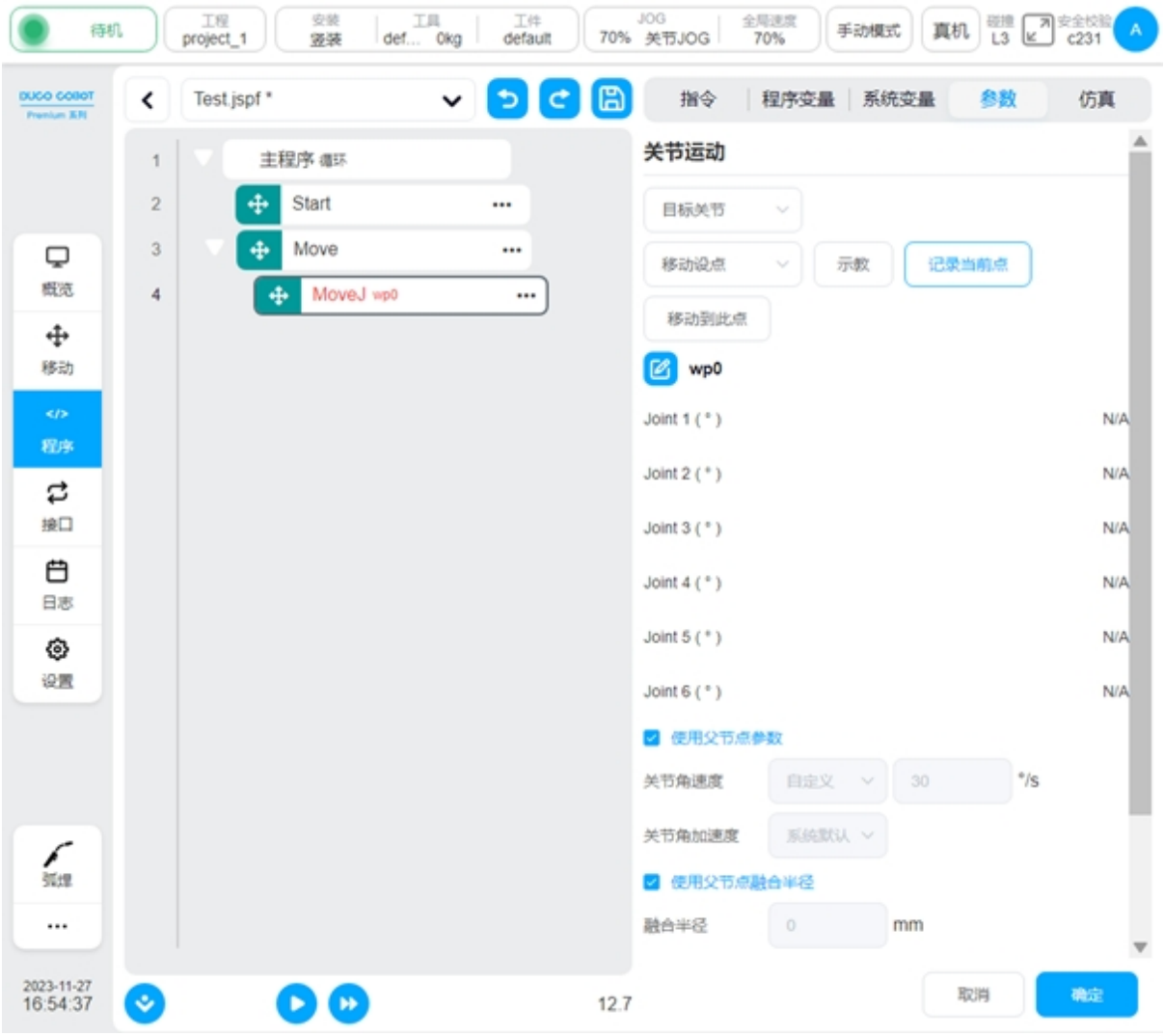
使用父节点融合半径：勾选时，该功能块使用父节点 Move 功能块设置的融合半径参数；不勾选时，需要单独为该功能块设置融合半径，默认勾选；

融合半径：单位 mm，取值范围为 [0,Infinity]，0 表示不融合；

备注： 注意：不合适的融合半径，会导致轨迹抖动，需要根据实际情况调整

备注： 注意：过小的融合半径长度可能会导致轨迹无法正常融合，若出现融合取消的情况，在确保不影响运动正常逻辑的情况下，优先使用 MotionConfig 色块配置融合预读取参数为“运动起点”。若依旧无法实现预期的融合效果，则应适当降低当前轨迹运行速度或增大当前轨迹的路径长度，保证轨迹路径长度运行时长至少不小于 100ms

启用 OP：OP 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。



若启用 OP，则需要做如下配置：

可以在轨迹开始后触发和轨迹结束前/结束后触发

触发类型：可选择不触发、时间触发、距离触发

触发延时：设置时间，单位 ms

触发操作：选择端口及端口状态，或操作自定义事件

备注： 对于 moveJ 运动，OP 操作仅可以通过时间控制。

MoveL

机器人按照直线移动到目标位姿，可设参数：

目标位姿：可以通过示教的方式设置或者设置为变量，示教设置后可手动更改；

示教：记录目标点位，点击将跳转到移动页面，在移动页面移动机器人后，点击“记录当前关节”按钮，将跳回程序页面，并设置点位；

记录当前点：点击后，将当前点位设置为目标点；

移动到此点：点击“移动到此点”，程序页面右侧将跳转到“仿真”页面。在仿真页面右下方将出现“按住移动”按钮。功能与概览页面的“按住移动”相同，详情请参见概览页面。

备注： 注意：从不同的机器人初始位姿使用“移动到此点”功能，会出现最终位姿一致，但机器人构型不一致，即关节位置不一致。可以通过观察灰透 3D 模型判断构型是否正确

备注： 注意：若脚本运行过程中运行到此脚本时机器人实际关节位置与试教时机器人的关节位置处于不同的选解空间，则会报错“机器人当前关节位置与试教参考关节位置选解不一致”

使用父节点坐标系：勾选时，该功能块使用父节点 Move 功能块设置的参考坐标系，默认勾选；

参考坐标系：不勾选使用父节点坐标系时，可单独为该功能块设置其参考坐标系；

使用父节点参数：勾选时，该功能块使用父节点 Move 功能块设置的末端速度、末端加速度参数；不勾选时，需要单独为该功能块设置末端速度、末端加速度，默认勾选；

末端速度：单位 mm/s，可以自定义或选择变量，自定义取值范围为 [0.01,5000]；

末端加速度：单位 mm/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

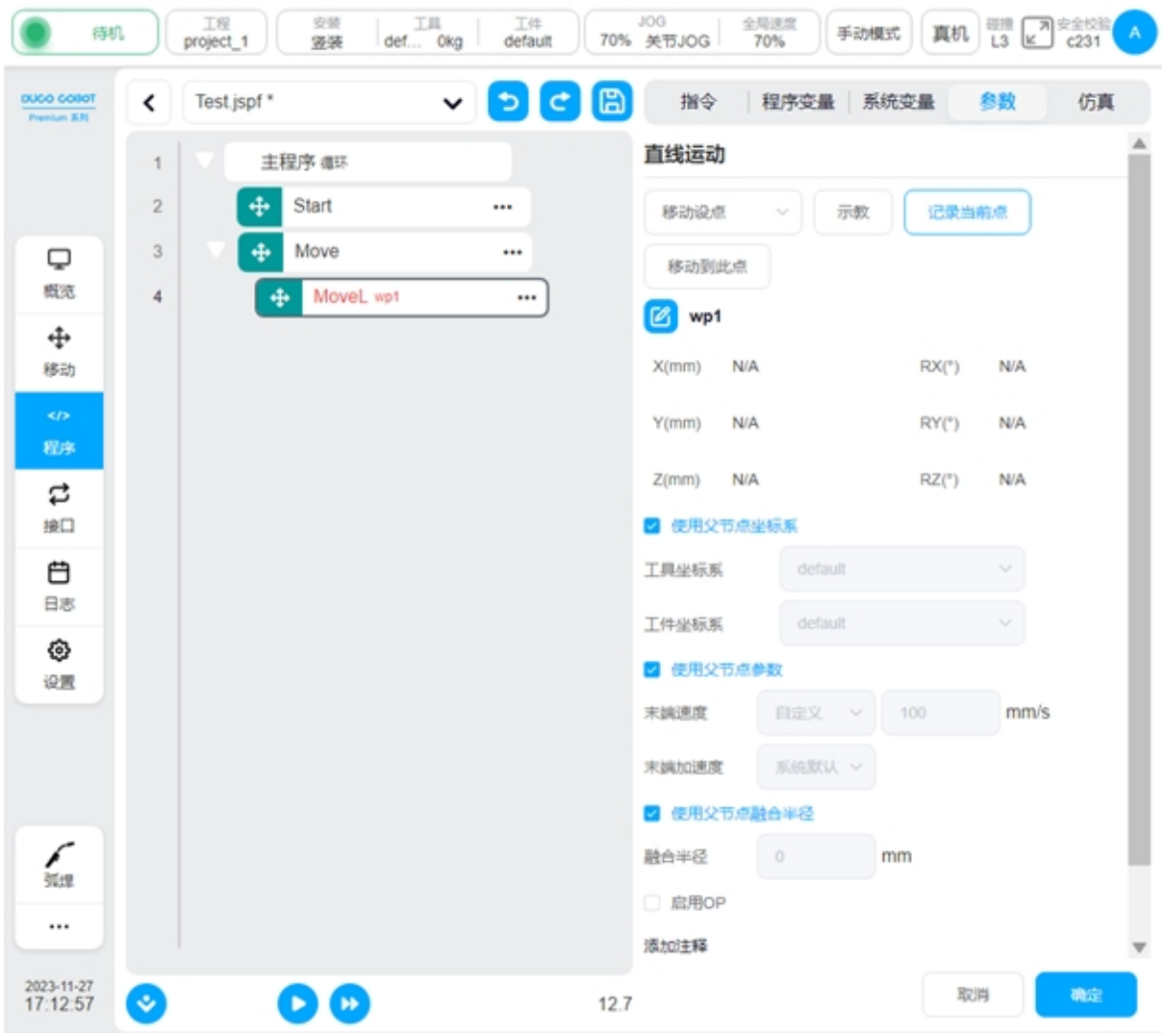
使用父节点融合半径：勾选时，该功能块使用父节点 Move 功能块设置的融合半径参数；不勾选时，需要单独为该功能块设置融合半径，默认勾选；

融合半径：单位 mm，取值范围为 [0,Infinity]，0 表示不融合；

备注： 注意：不合适的融合半径，会导致轨迹抖动，需要根据实际情况调整

备注： 注意：过小的融合半径长度可能会导致轨迹无法正常融合，若出现融合取消的情况，在确保不影响运动正常逻辑的情况下，优先使用 MotionConfig 色块配置融合预读取参数为“运动起点”。若依旧无法实现预期的融合效果，则应适当降低当前轨迹运行速度或增大当前轨迹的路径长度，保证轨迹路径长度运行时长至少不小于 100ms

启用 OP：OP 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。



若启用 OP，则需要做如下配置：

可以在轨迹开始后触发和轨迹结束前/结束后触发。

触发类型：可选择不触发、时间触发、距离触发

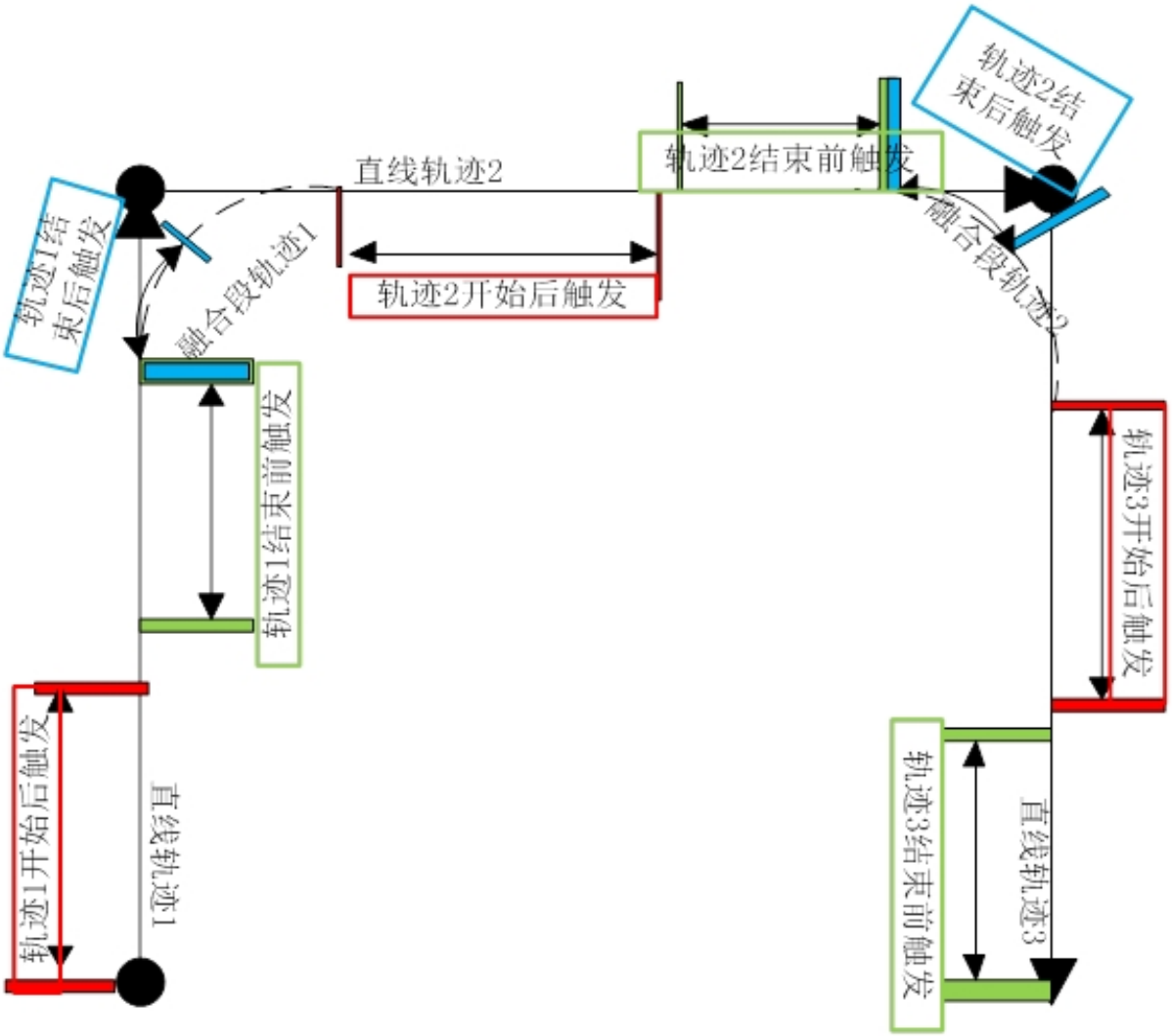
触发延时：设置时间，单位 ms

触发距离：设置距离，单位 mm

触发操作：选择端口及端口状态，或操作自定义事件

融合中的 OP 操作：

机器人在轨迹执行过程中设置通用数字输出口状态。当与融合一起使用时，以三条 moveL 融合为例，其效果如下：



MoveC

机器人按照圆弧或者整圆移动，可设参数：

模式：圆弧、整圆、圆弧角；

圆弧角：若模式选择圆弧角时，需设置圆弧角，当前点与参考点 1、参考点 2 三点确定圆弧路径与起点，由圆弧角决定终点，只允许取正值，即取值范围大于 0，默认为 0，路径按圆弧模式处理；

中间点/中间点 1/参考点 1 位姿：可以通过示教的方式设置或者设置为变量，示教设置后可手动更改；

目标点/中间点 2/参考点 2 位姿：可以通过示教的方式设置或者设置为变量，示教设置后可手动更改；

示教：记录目标点位，点击将跳转到移动页面，在移动页面移动机器人后，点击“记录当前关节”按钮，将跳回程序页面，并设置点位；

记录当前点：点击后，将当前点位设置为目标点；

移动到此点：点击“移动到此点”，程序页面右侧将跳转到“仿真”页面。在仿真页面右下方将出现“按住移动”按钮。功能与概览页面的“按住移动”相同，详情请参见概览页面。

备注： 注意：从不同的机器人初始位姿使用“移动到此点”功能，会出现最终位姿一致，但机

器人构型不一致，即关节位置不一致。可以通过观察灰透 3D 模型判断构型是否正确

备注： 注意：若脚本运行过程中运行到此脚本时机器人实际关节位置与试教时机器人的关节位置处于不同的选解空间，则会报错“机器人当前关节位置与试教参考关节位置选解不一致”

使用父节点坐标系：勾选时，该功能块使用父节点 Move 功能块设置的参考坐标系，默认勾选；

参考坐标系：不勾选使用父节点坐标系时，可单独为该功能块设置其参考坐标系；

使用父节点参数：勾选时，该功能块使用父节点 Move 功能块设置的末端速度、末端加速度参数；不勾选时，需要单独为该功能块设置末端速度、末端加速度，默认勾选；

末端速度：单位 mm/s，可以自定义或选择变量，自定义取值范围为 [0.01,5000]；

末端加速度：单位 mm/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

使用父节点融合半径：勾选时，该功能块使用父节点 Move 功能块设置的融合半径参数；不勾选时，需要单独为该功能块设置融合半径，默认勾选；

融合半径：单位 mm，取值范围为 [0,Infinity]，0 表示不融合；

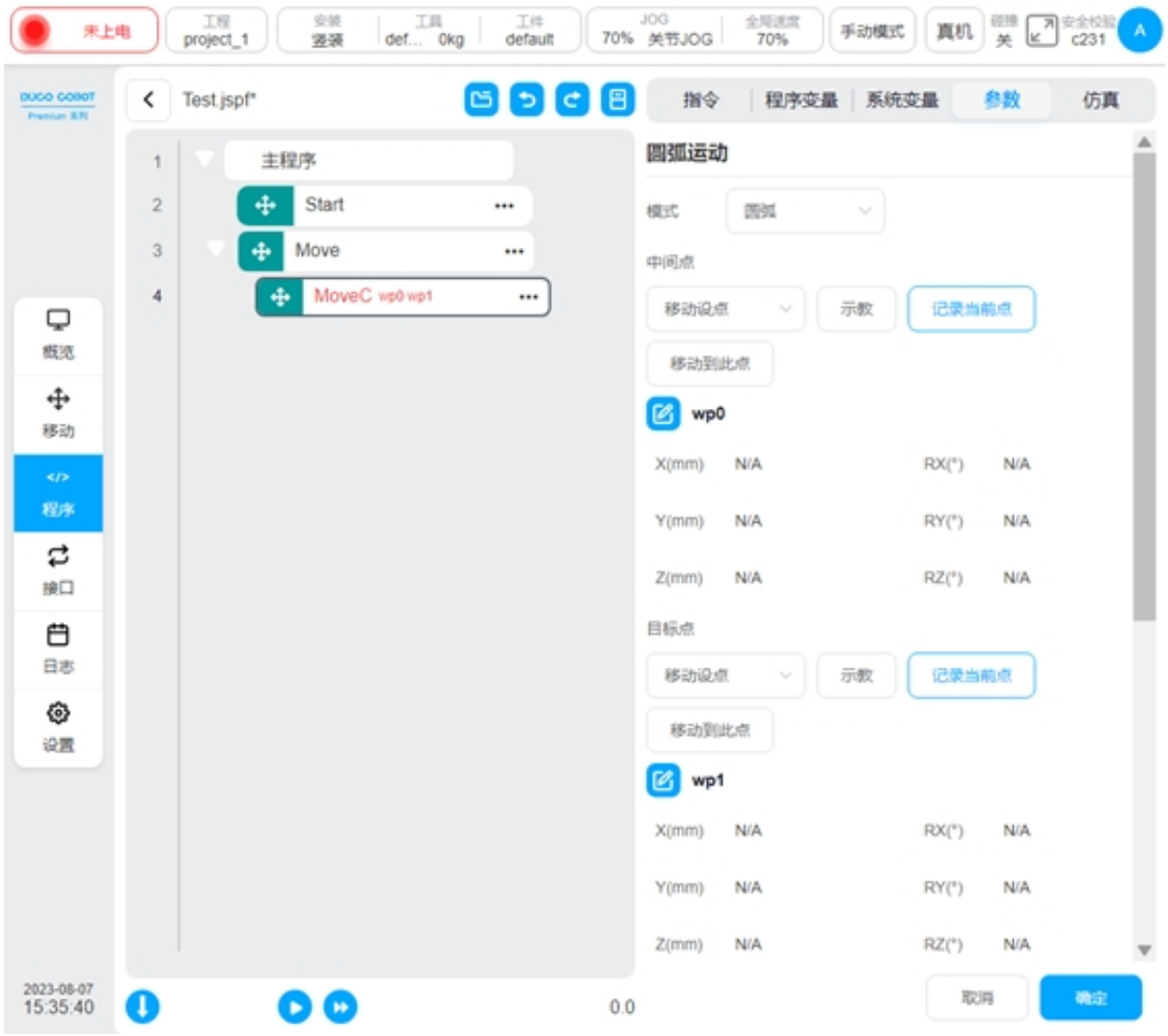
备注： 注意：不合适的融合半径，会导致轨迹抖动，需要根据实际情况调整

备注： 注意：过小的融合半径长度可能会导致轨迹无法正常融合，若出现融合取消的情况，在确保不影响运动正常逻辑的情况下，优先使用 MotionConfig 色块配置融合预读取参数为“运动起点”。若依旧无法实现预期的融合效果，则应适当降低当前轨迹运行速度或增大当前轨迹的路径长度，保证轨迹路径长度运行时长至少不小于 100ms

姿态控制模式：若选择“与终点一致”，机器人的姿态根据终点姿态对圆弧路径中的姿态进行规划；若选择“与起点一致”，机器人的姿态根据起点姿态对圆弧路径中的姿态进行规划，路径过程中姿态与起点一致；若选择“受圆心约束”，机器人的姿态相对于圆弧运动产生的姿态变化对姿态进行约束；

启用 OP：OP 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。

OP 参数配置同 MoveL



MoveTCP

机器人沿工具坐标系移动。

可以自定义各个方向的偏移量，或者使用变量。可以示教两个点位，以这两个点位之间的偏移量作为移动的偏移值。其他可设参数：

使用父节点坐标系：勾选时，该功能块使用父节点 Move 功能块设置的参考坐标系，默认勾选；

参考坐标系：不勾选使用父节点坐标系时，可单独为该功能块设置其参考坐标系；若选择设置增量方式，只需选择工具坐标系；若选择设置两点方式，则需设置工具坐标系和工件坐标系；

使用父节点参数：勾选时，该功能块使用父节点 Move 功能块设置的末端速度、末端加速度参数；不勾选时，需要单独为该功能块设置末端速度、末端加速度，默认勾选；

末端速度：单位 mm/s，可以自定义或选择变量，自定义取值范围为 [0.01,5000]；

末端加速度：单位 mm/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

使用父节点融合半径：勾选时，该功能块使用父节点 Move 功能块设置的融合半径参数；不勾选时，需要单独为该功能块设置融合半径，默认勾选；

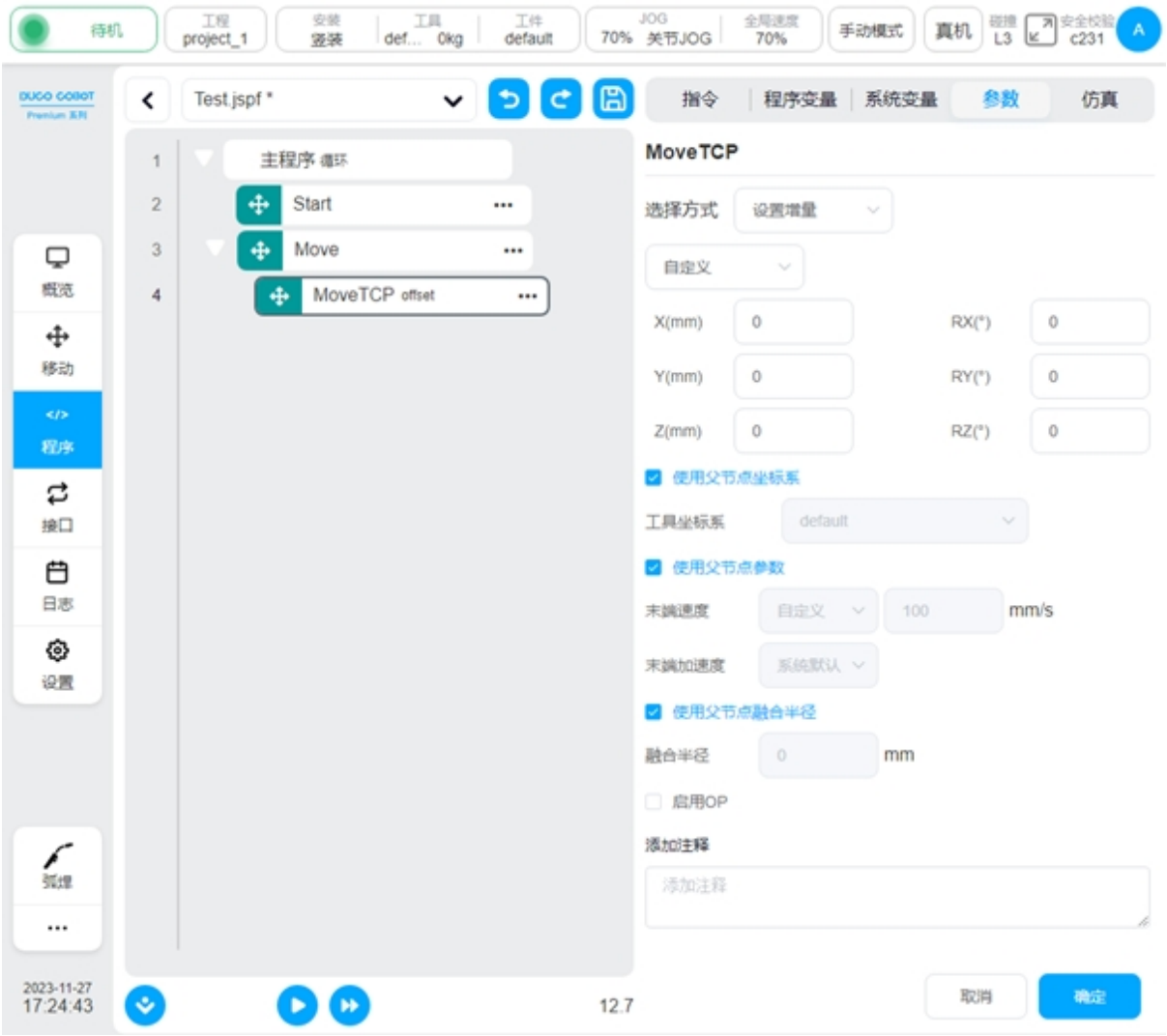
融合半径：单位 mm，取值范围为 [0,Infinity]，0 表示不融合；

备注： 注意：不合适的融合半径，会导致轨迹抖动，需要根据实际情况调整

备注： 注意：过小的融合半径长度可能会导致轨迹无法正常融合，若出现融合取消的情况，在确保不影响运动正常逻辑的情况下，优先使用 `MotionConfig` 色块配置融合预读取参数为“运动起点”。若依旧无法实现预期的融合效果，则应适当降低当前轨迹运行速度或增大当前轨迹的路径长度，保证轨迹路径长度运行时长至少不小于 `100ms`

启用 `OP: OP` 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。

`OP` 参数配置同 `MoveL`



MoveWobj

使机器人末端工具沿工件坐标系轴向进行增量运动，运动方向与工件坐标系轴所在方向平行。

可以自定义各个方向的偏移量，或者使用变量。可以示教两个点位，以这两个点位之间的偏移量作为移动的偏移值。其他可设参数：

使用父节点坐标系：勾选时，该功能块使用父节点 `Move` 功能块设置的参考坐标系，默认勾

选；

参考坐标系：不勾选使用父节点坐标系时，可单独为该功能块设置其参考坐标系；若选择设置增量方式，只需选择工件坐标系；若选择设置两点方式，则需设置示教点所对应的工具坐标系和工件坐标系；

使用父节点参数：勾选时，该功能块使用父节点 Move 功能块设置的末端速度、末端加速度参数；不勾选时，需要单独为该功能块设置末端速度、末端加速度，默认勾选；

末端速度：单位 mm/s，可以自定义或选择变量，自定义取值范围为 [0.01,5000]；

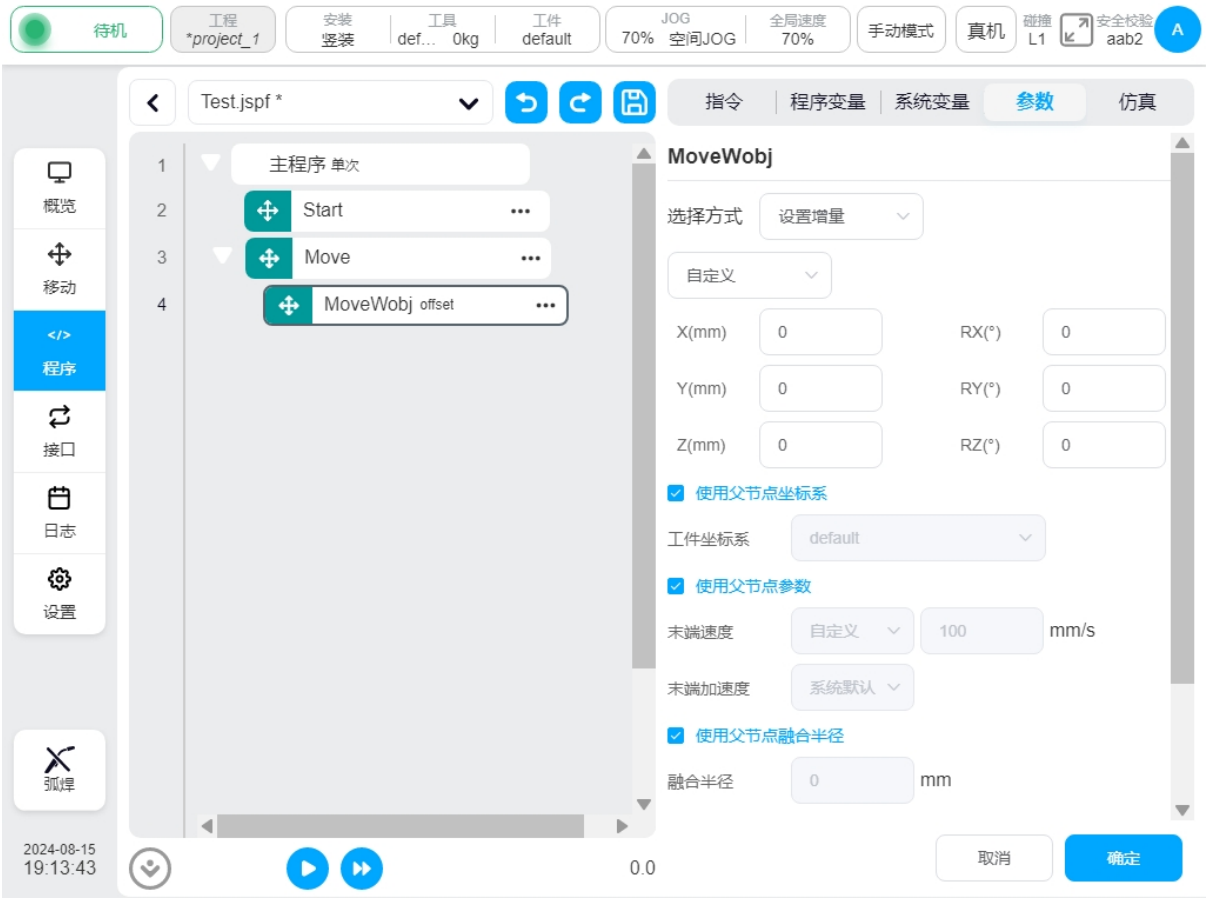
末端加速度：单位 mm/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

使用父节点融合半径：勾选时，该功能块使用父节点 Move 功能块设置的融合半径参数；不勾选时，需要单独为该功能块设置融合半径，默认勾选；

融合半径：单位 mm，取值范围为 [0,Infinity]，0 表示不融合；

启用 OP：OP 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。

OP 参数配置同 MoveL



Spline

控制机器人末端按照样条曲线移动。可以选择自定义即直接示教点位，或者选择变量，且选择变量时，只能选择 pose_list 类型变量，其他可设参数：

样条路点：添加、编辑、删除路点，根据路点生成样条曲线；

参考坐标系：示教点位时，默认为当前的工具和工件坐标系，可手动更改；

末端速度：单位 mm/s，可以自定义或选择变量，自定义取值范围为 [0.01,5000]；

末端加速度：单位 mm/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

融合半径：单位 mm，取值范围为 [0,Infinity]，0 表示不融合；

备注： 注意：不合适的融合半径，会导致轨迹抖动，需要根据实际情况调整

备注： 注意：过小的融合半径长度可能会导致轨迹无法正常融合，若出现融合取消的情况，在确保不影响运动正常逻辑的情况下，优先使用 MotionConfig 色块配置融合预读取参数为“运动起点”。若依旧无法实现预期的融合效果，则应适当降低当前轨迹运行速度或增大当前轨迹的路径长度，保证轨迹路径长度运行时长至少不小于 100ms

启用 OP：OP 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。

OP 参数配置同 MoveL

备注： 连续两条 spline 一起使用时，需要注意前一条的 spline 结束点与后一条的 spline 起始点，不能是同一个点。

MotionConfig

打开或关闭运动相关配置。可选配置：

速度优化：打开速度优化后，机械臂将在满足系统约束的前提下，以尽可能高的速度执行运动指令；

振动控制：打开振动控制后，将会针对机器人末端振动进行优化；

奇异规避：打开奇异规避后，将会在机器人经过奇异空间附近时，对轨迹自动优化；

备注： 奇异规避功能启用时，如果经过奇异点，将影响原有设定路径，而且会使得个别关节速度变快，可能因为关节超速而导致机械臂报错斯电。

姿态约束：打开姿态约束后，机器人运动姿态将会与路径间关系始终保持一致；

融合预读取：可选运动中间点或运动起点，配置了融合半径的运动脚本会根据此设定参数在对应的位置预读取后续脚本以进行融合半径。

备注： 对于短线段与长线段的融合，通常需要将短线段的融合预读取位置设为运动起点，以保证程序有足够的时间进行融合的轨迹规划。但这并不意味着，设运动起点为融合预读取点位后，一定可以执行短线段融合长线段。

预读取数量：当且仅当融合预读取被配置为从运动起点获取时，此时机器人由于无需等待当前运动指令执行过半后读取下一条指令，因此可以进一步选择想要预读取的指令数量，当前最多支持预读取 3 行运动指令。当使用机器人以较高速度运行连续且路径较短的运动轨迹时，通过配置融合预读取模式为从运动起点并提升运动指令预读取数量，可以较大程度的提升融合运动规划的成功率以及一致性。

备注： 当使用融合功能并且配置了融合预读取多条运动指令时，若在融合运动指令间存在其他类型的机器人指令时，如 IO 读写、寄存器读写、变量操作、接口操作等，该指令会先于机器人实际运动到目标示教点位执行，因此在确认使用多行运动指令的融合预读取功能后，需要额外试运行以确认所有寄存器、IO、变量及程序逻辑状态跳转是否正常。

MoveSpiral

机器人做螺旋轨迹运动，可设参数：

螺旋示教方式：参数设置或者结束点设置；

总圈数：机器人总旋转圈数，正数为顺时针旋转，负数为逆时针旋转；

螺旋线中心点：可以通过示教的方式设置或者设置为变量，示教设置后可手动更改；

轴向移动距离：示教方式为参数设置时，需配置的参数，正负号遵循右手定则，单位 mm；

结束点半径：示教方式为参数设置时，需配置的参数，单位 mm；

轨迹结束点：示教方式为结束点设置时，可以通过示教的方式设置，或者设置为变量，示教设置后可手动更改；

使用父节点坐标系：勾选时，该功能块使用父节点 Move 功能块设置的参考坐标系，默认勾选；

参考坐标系：不勾选使用父节点坐标系时，可单独为该功能块设置其参考坐标系；

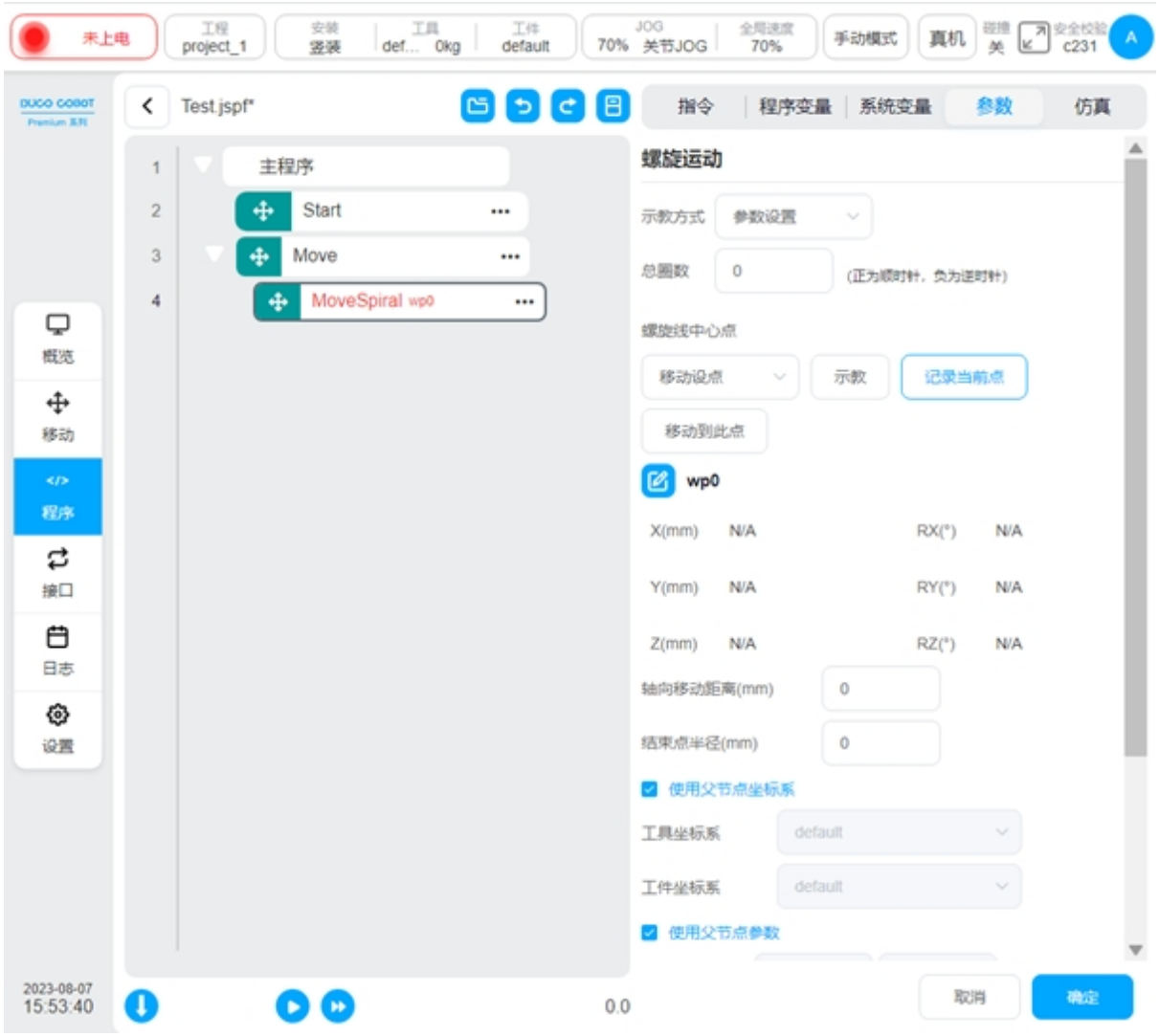
使用父节点参数：勾选时，该功能块使用父节点 Move 功能块设置的末端速度、末端加速度参数；不勾选时，需要单独为该功能块设置末端速度、末端加速度，默认勾选；

末端速度：单位 mm/s，可以自定义或选择变量，自定义取值范围为 [0.01,5000]；

末端加速度：单位 mm/s2，可以自定义或选择变量，自定义取值范围为 [0.01,Infinity]；

启用 OP：OP 功能可以在轨迹执行过程中设置通用数字输出口状态或设置工具输出状态或操作自定义事件。

OP 参数配置同 MoveL。



CombineMotion

仅针对直线及圆弧运动控制机器人做复合轨迹运动，可设参数：

轨迹类型：默认为三角，可选三角、正弦、圆弧、梯形、8 字；

参考平面：默认为工具 XOY，可选工具 XOY、工具 XOZ、工具 YOZ、工件 XOY、工件 XOZ、工件 YOZ；

频率：默认值为 1，单位 Hz；

振幅：默认值为 1，单位 mm；

主路径同步停留：仅轨迹类型为正弦或梯形时，需要配置的参数，默认为不勾选；当勾选该参数时，能够在左右停留过程中，机器人运动主路径也同时停止；

左停留时间：仅轨迹类型选为正弦或梯形时，需配置的参数，单位 ms；

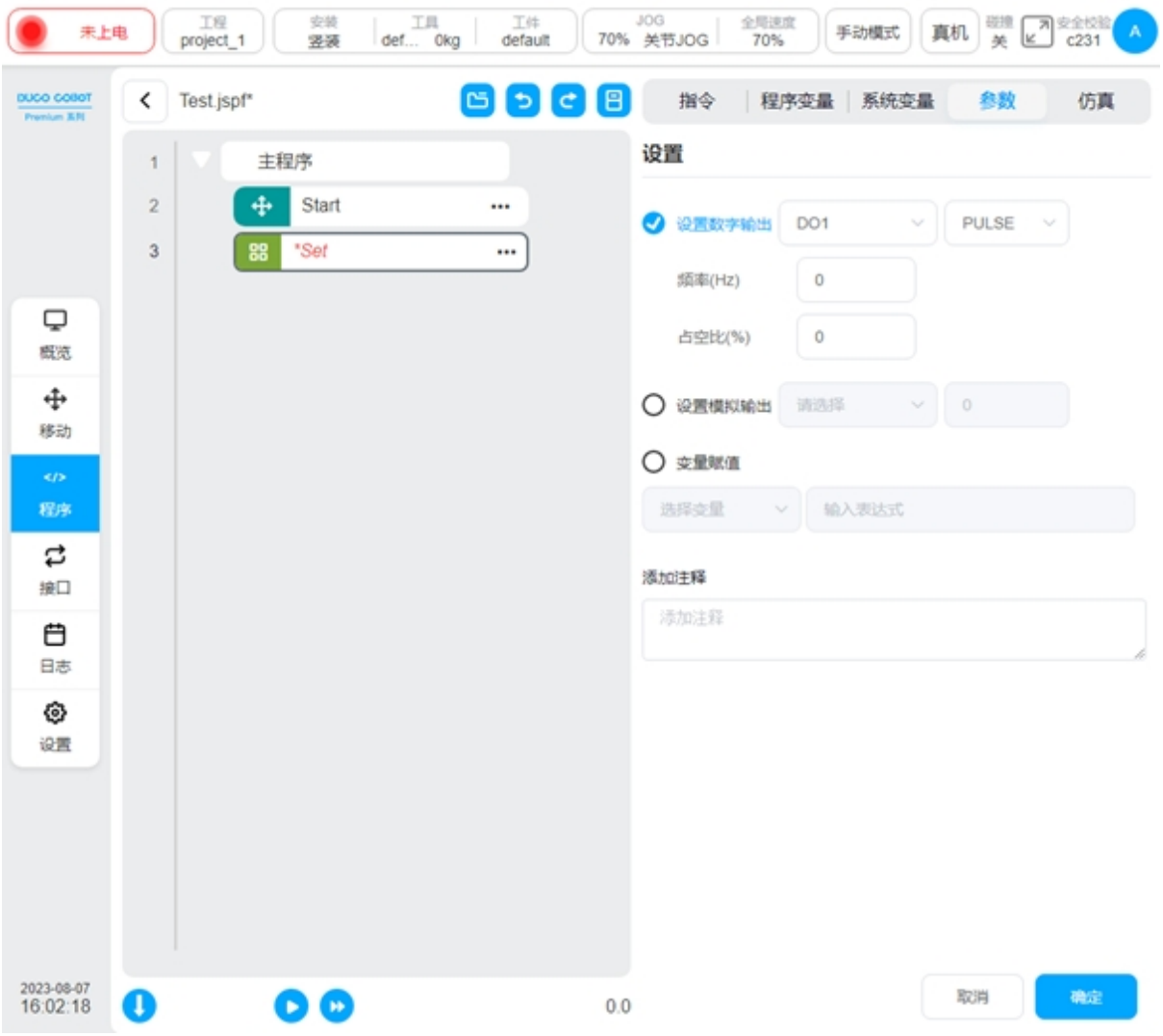
右停留时间：仅轨迹类型选为正弦或梯形时，需配置的参数，单位 ms；

基础功能块

Set

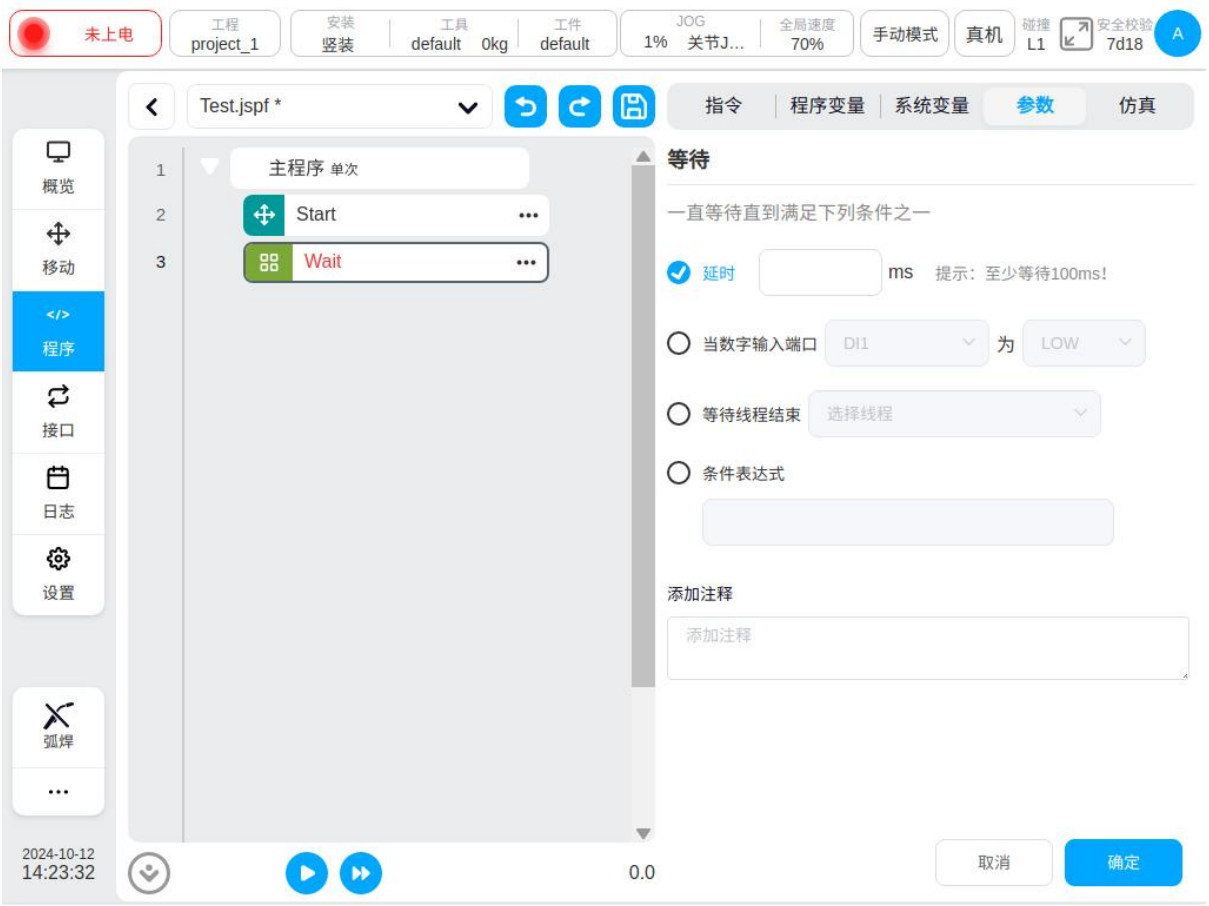
设置功能块。可设定控制柜数字输出端口和机械臂末端数字输出端口为高电平或低电平或脉冲信号，且设为脉冲信号时，需要设置脉冲的频率和占空比；为程序变量或系统变量赋值。

备注： 为变量赋值时，将弹出表达式键盘，请按照表达式的语法写赋值内容。如为字符串变量赋值“Hello”，键盘需输入“Hello”（双引号不能省略）



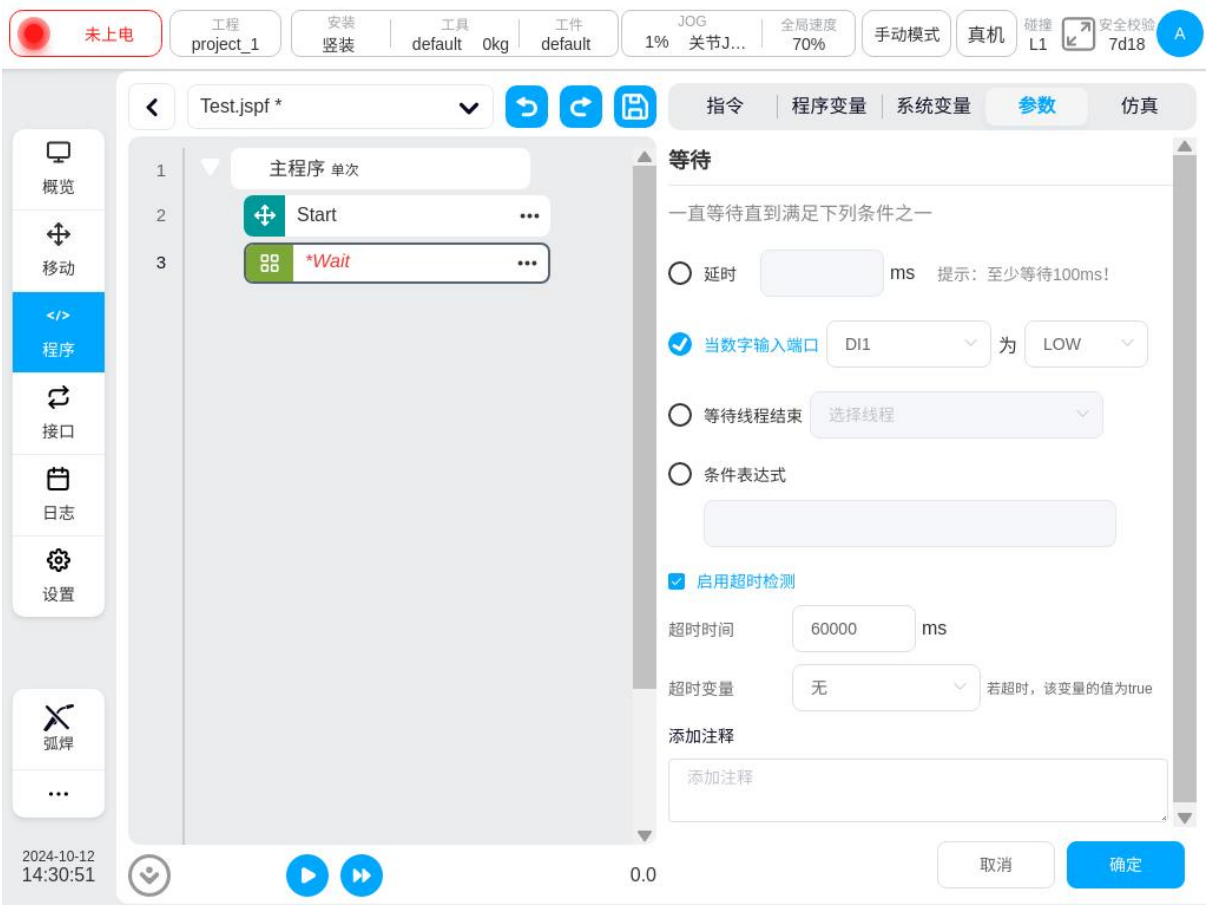
Wait

等待功能块。可选等待一段时间、等待 DI 信号、等待某一线程结束或表达式，程序执行到此功能块时会一直等待直到满足设定的条件。



备注： 注意：延时设置至少 100ms！

当选中除“延时”以外其他选项时，会显示是否启用超时检测的勾选框。若勾选“启用超时检测”，可输入超时时间和超时变量，如下图。



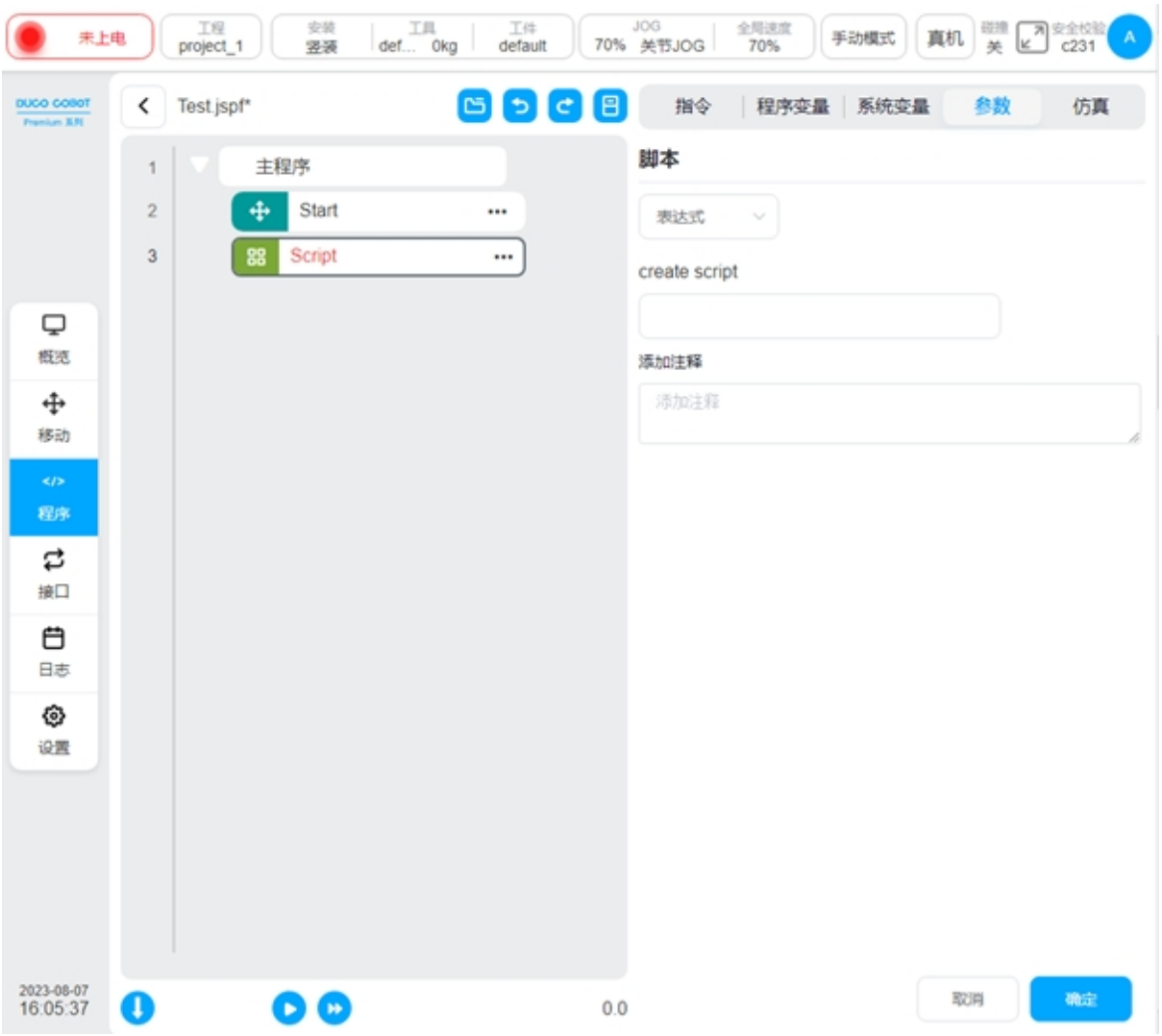
用户可手动输入超时时间，默认超时时间 60000ms；用户也可以使用超时变量判断是否超时，且超时变量为可选项，可以不选择变量。

当等待条件不满足且超时后，会输出超时信号，若机器人在自动模式时，会继续等待直到满足条件；若机器人在手动模式时，超时后会暂停程序并弹窗如下图。

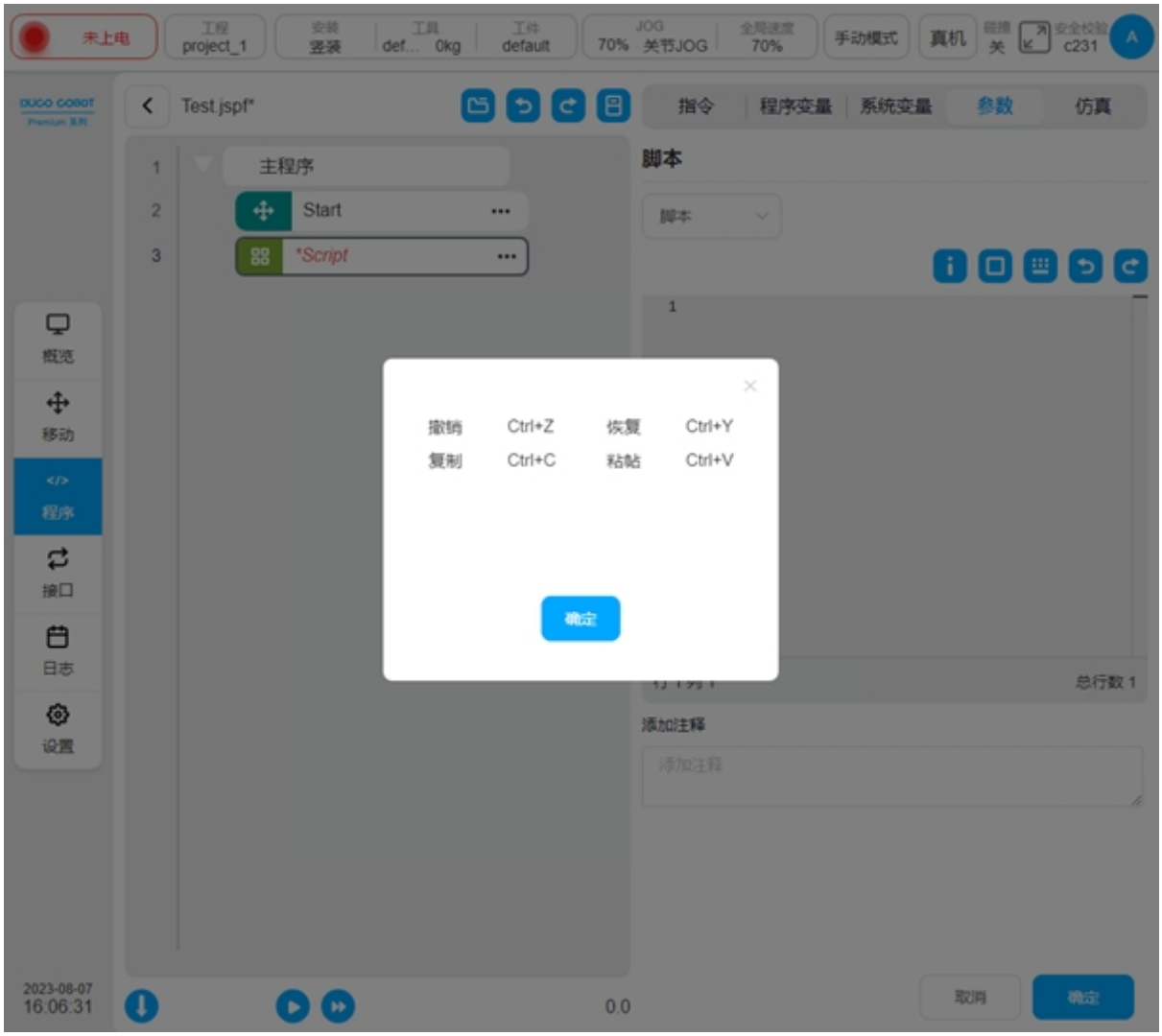
当用户选择“跳过并继续”时，程序会跳过 `Wait` 等待并继续运行程序；当用户选择“暂停并关闭”时，程序继续暂停且弹窗被关闭。

`Script`

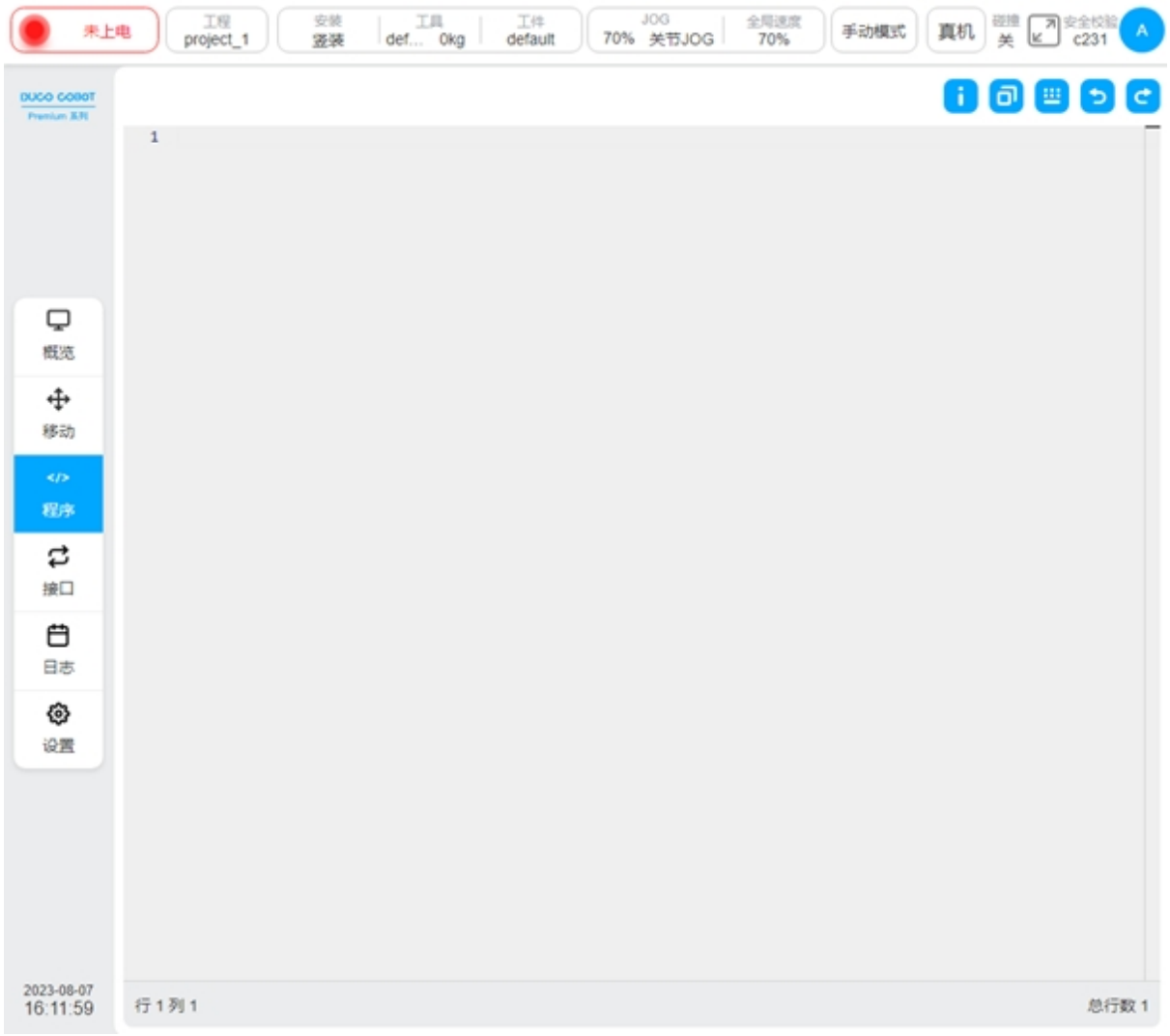
脚本功能块。可以选择表达式、脚本或者脚本文件。表达式可使用表达式编辑器创建一行脚本，表达式的输入请参见第 8 节输入键盘（表达式输入键盘）；脚本可用于编写整段的脚本代码；脚本文件可以从文件中选择脚本一个脚本文件。脚本代码的说明参见脚本手册。



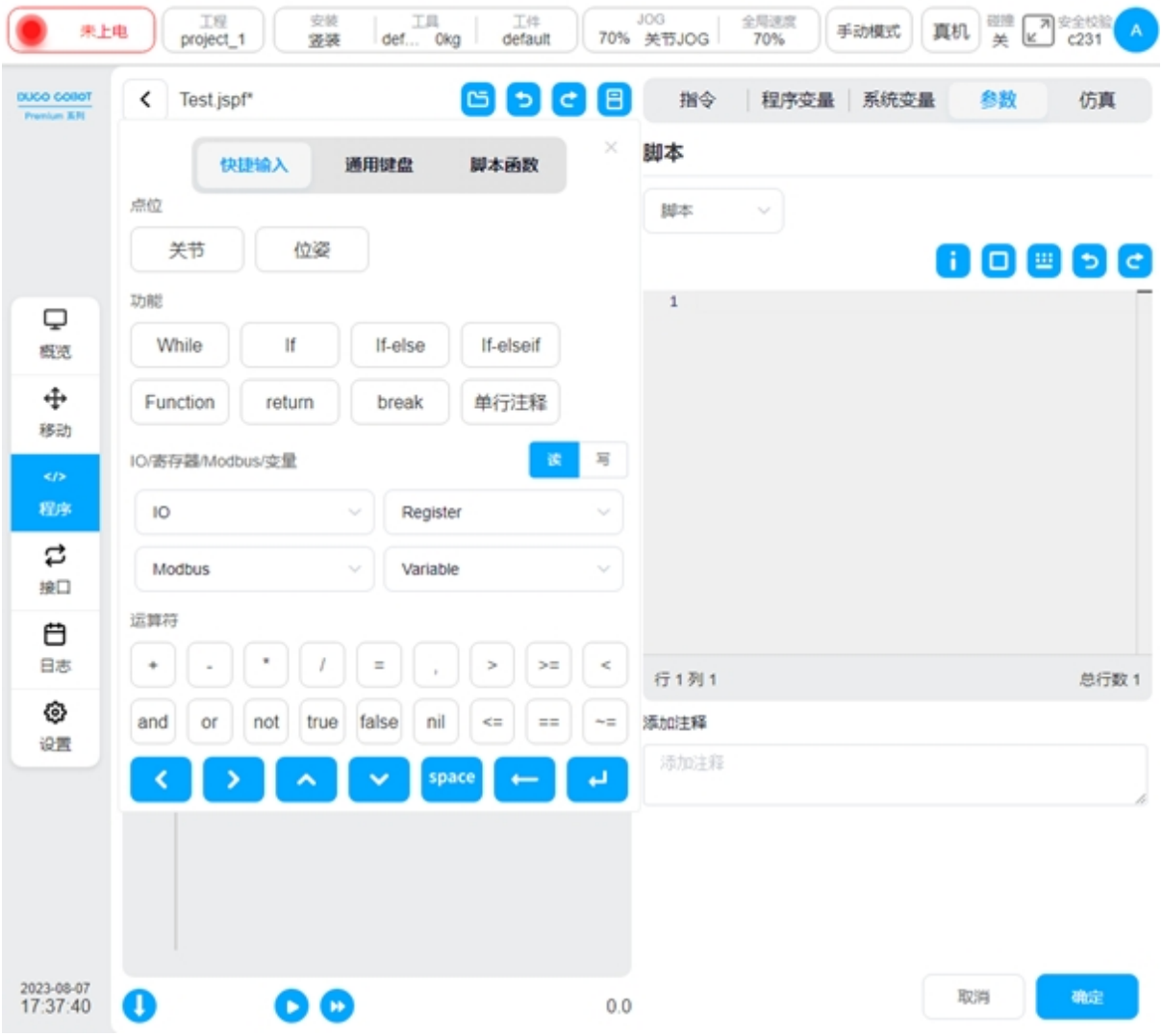
选择脚本编写整段代码时，点击编辑区上方  提示图标会弹出编辑快捷键提示框，如下图。



点击编辑区上方  最大化图标，脚本编辑区会最大化显示如下，点击最大化界面右上方  还原图标即可还原到原始界面大小。  为撤销图标，  为恢复图标。编辑区左下方会显示光标所在的行数和列数，右下方会显示脚本总行数。



使用脚本编写方式可以使用系统自带的快捷输入框和外部键盘输入。点击脚本编写区上方的  图标弹出系统快捷输入框，悬浮在编辑器上方如下图，且该弹框可被按住拖动。

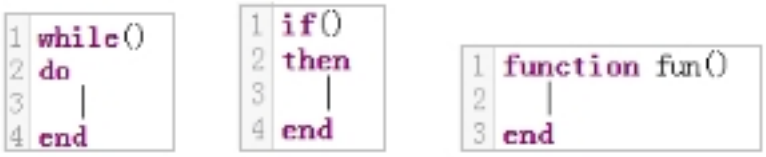


快捷输入框有三个标签，分别为快捷输入、通用键盘、脚本函数。输入框下方栏为一些常用的控制按键，功能分别为光标左移、光标右移、光标上移、光标下移、输入空格、删除、换行。

快捷输入会针对一些常用的功能进行快捷输入。主要功能有

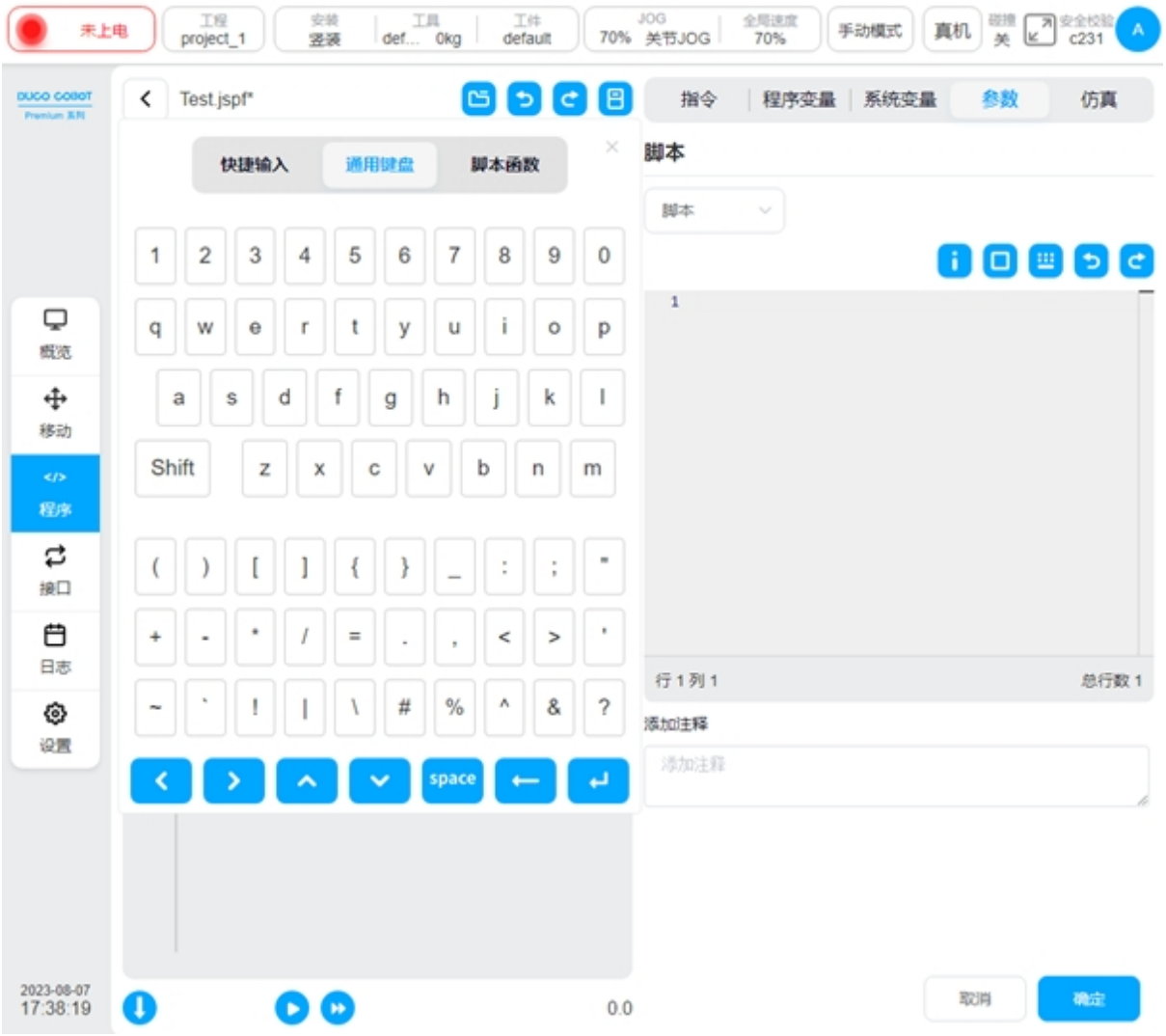
点位： 点击“关节”可输入当前机械臂的关节值，单位是 rad；点击“位姿”，可输入当前机械臂工具在设定工件上的位置姿态，单位 m、rad。

功能： 可以输入 While 循环、条件分支、函数定义、单行注释等



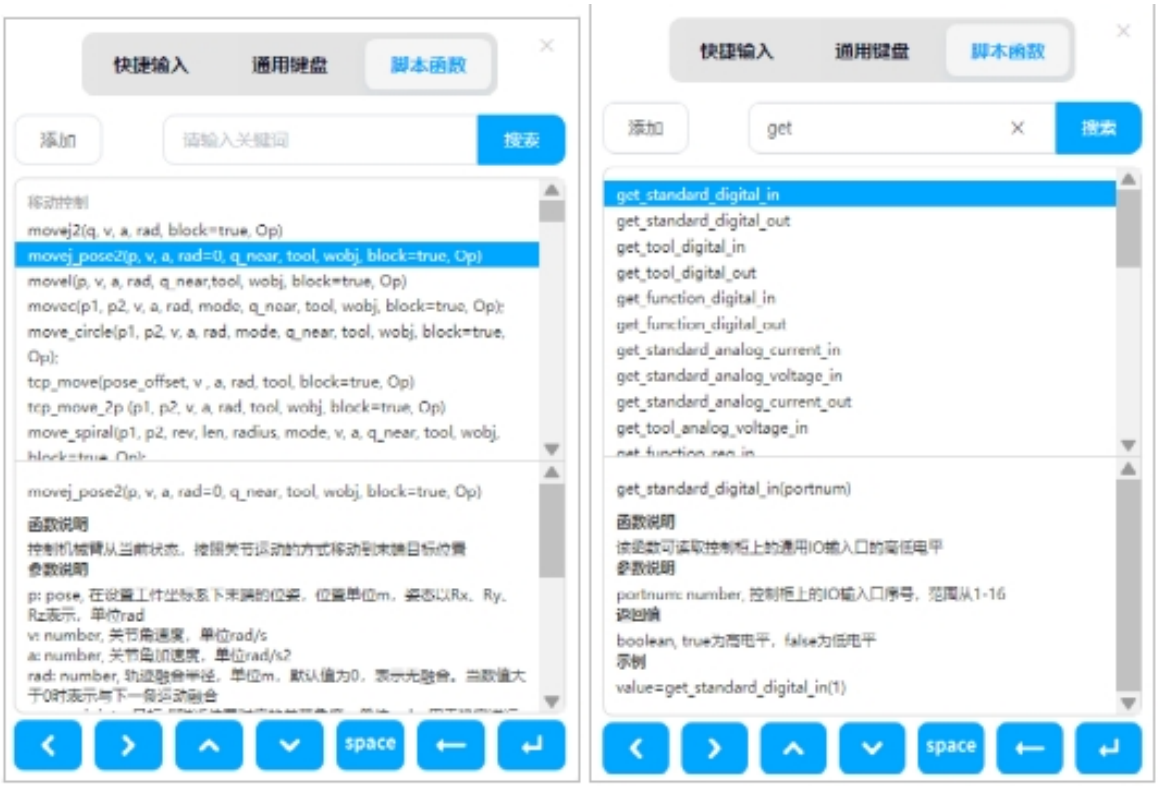
IO/寄存器/Modbus/变量： 可以选择是读取还是写入，系统会自动转换为相应的脚本函数。如选择“读”时，选择通用输入 DI1，系统自动输入对应的脚本函数 get_standard_digital_in(1)

运算符： 输入一些常用的运算符



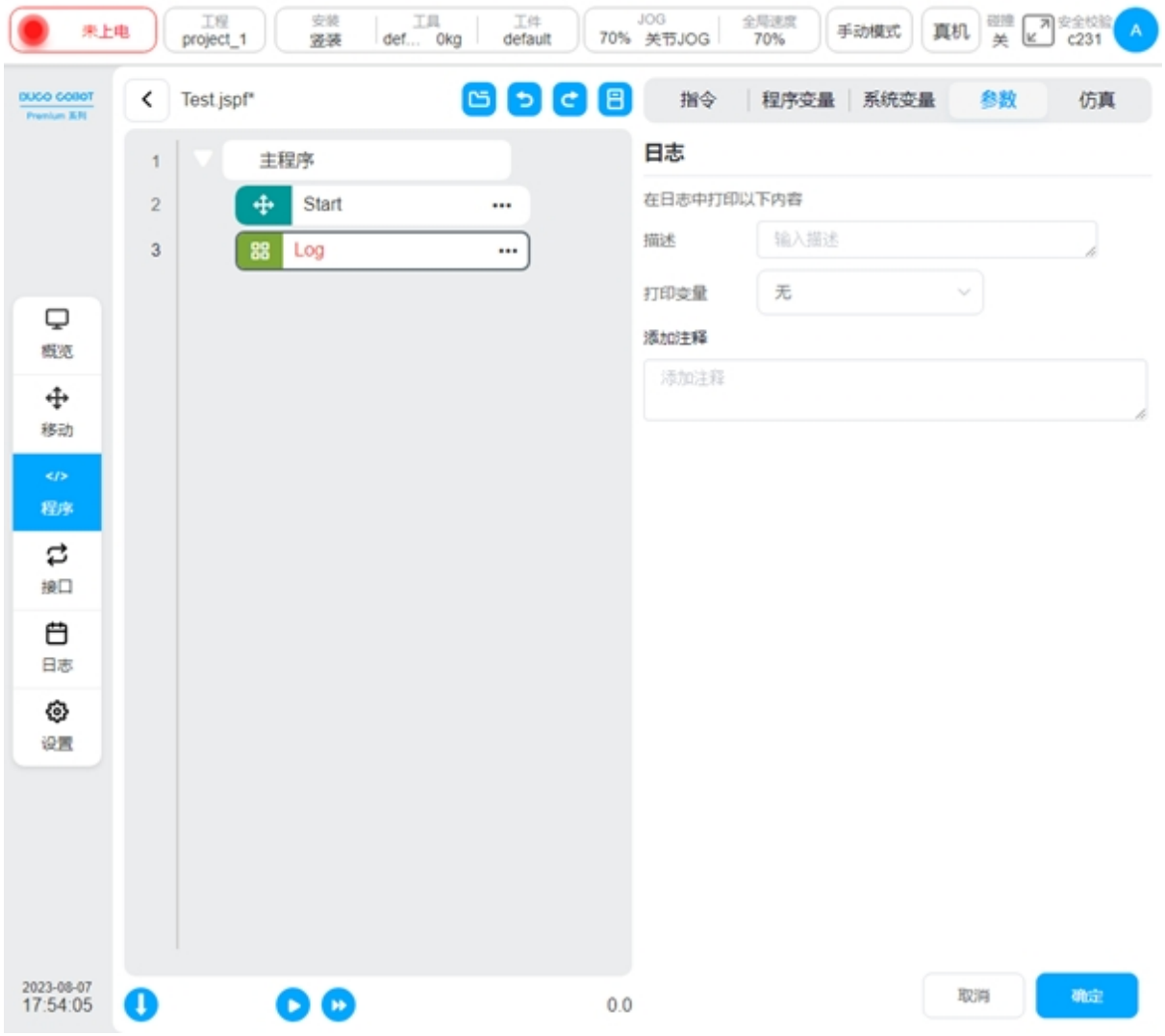
通用键盘与实体键盘类似，可以输入大小写字母、数字、符号。

脚本函数可以从列表中选择脚本函数，选中后，在下方会显示该脚本函数的说明及示例，双击或者点击左上方的“添加”按钮可以将该脚本输入到脚本区。支持脚本的搜索。



Log

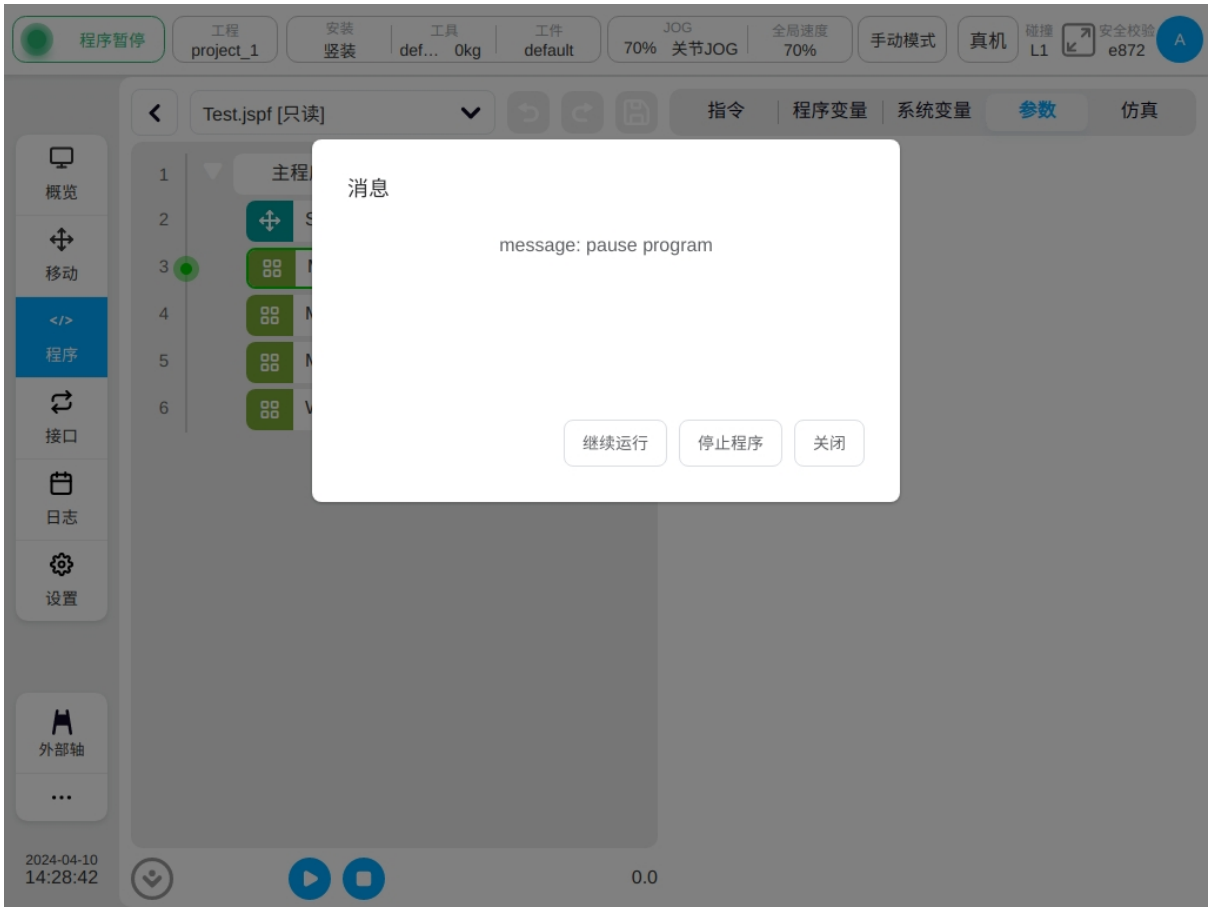
日志功能块。可以打印消息或者变量的值到日志文件中。



Message

消息弹窗功能块。可对消息弹框进行配置，即可选暂停程序、仅弹窗或停止程序。

当选择暂停程序时，设定一条消息，程序运行到此功能块时可弹出一个对话框显示该消息并且程序暂停运行，用户可选择继续运行、停止程序或者关闭弹窗。

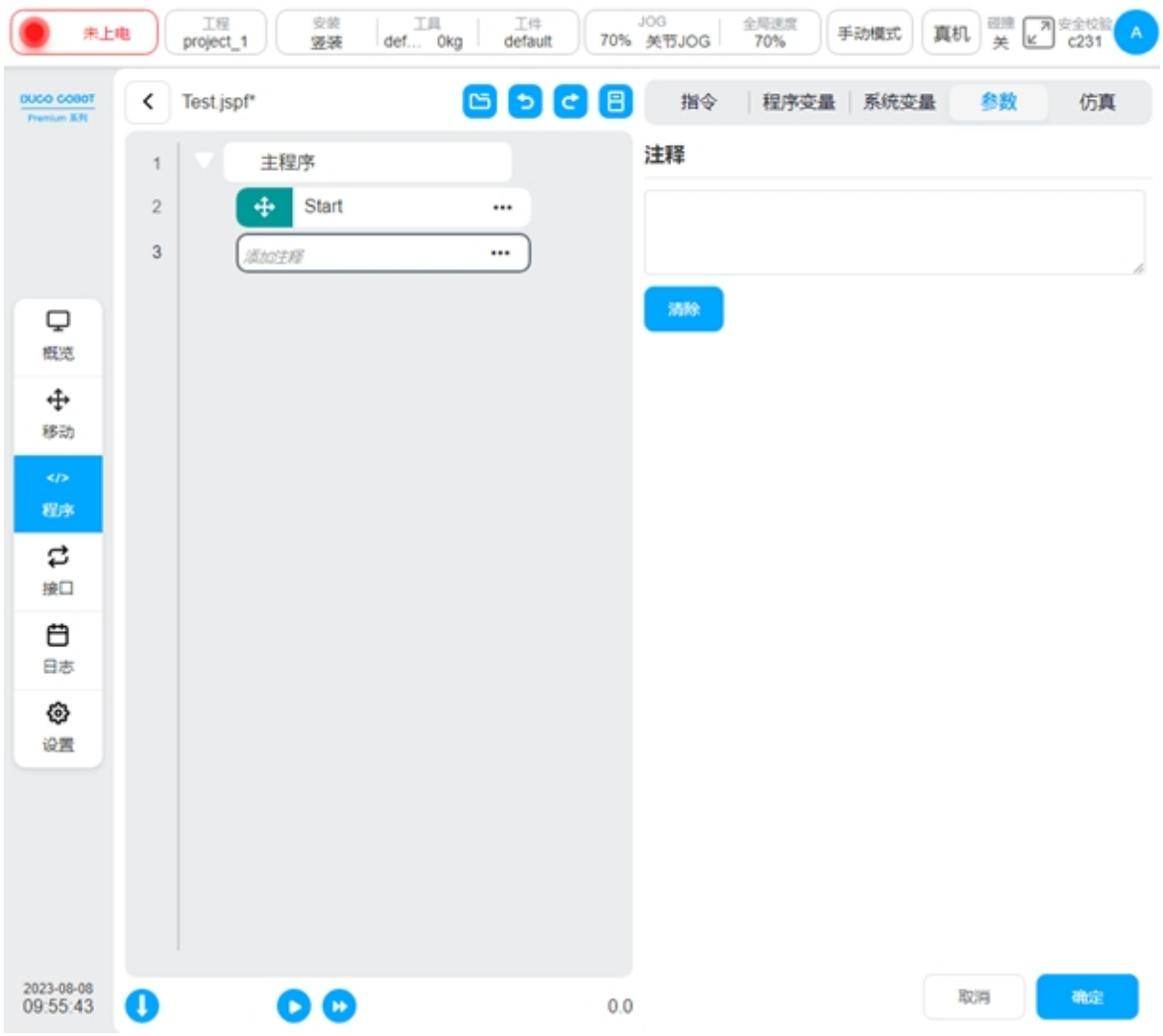


当选择仅弹窗时，设定一条消息，程序运行到此功能块时弹出一个对话框显示该消息，但程序会继续运行，用户可选择停止程序或者关闭弹窗。

当选择停止程序时，设定一条消息，程序运行到此功能块时弹出一个对话框显示该消息并且程序直接停止运行，用户可选择关闭弹窗。

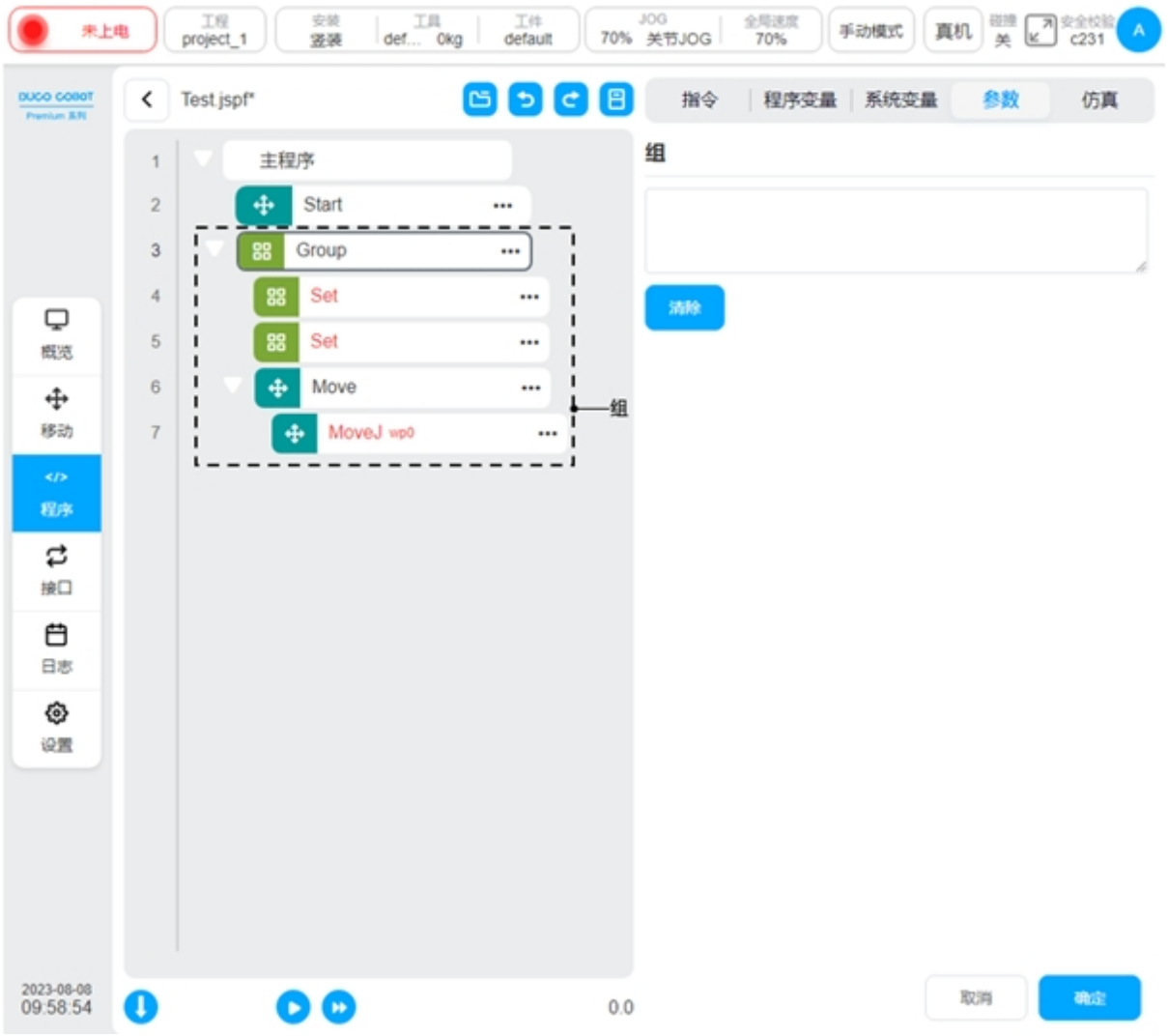
Comment

注释功能块。向程序树中添加一个注释。程序运行时，此功能块不会执行任何操作。



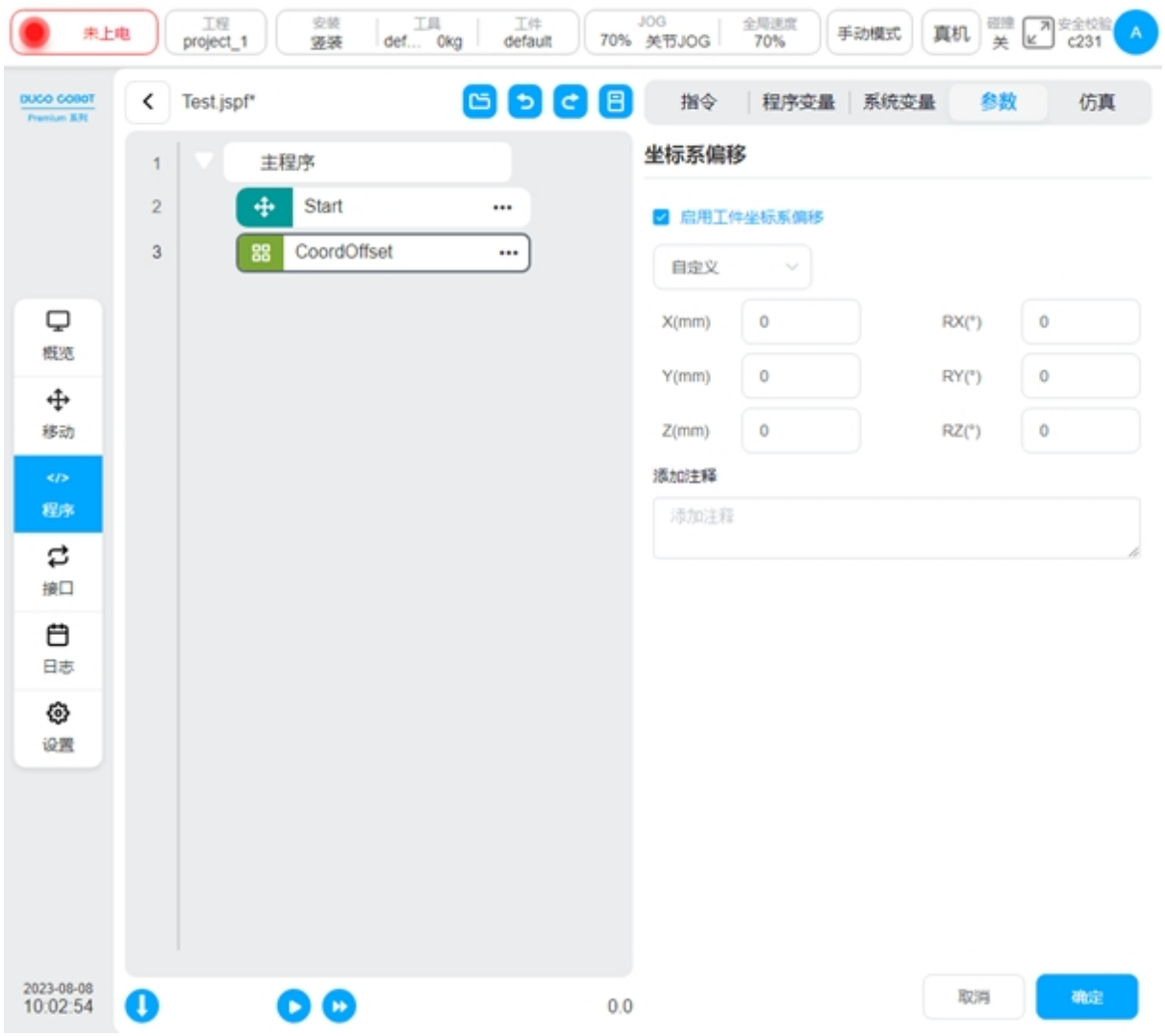
Group

组功能块。用于对程序进行整理，可以将某些功能块放在一个 Group 下面，方便程序的组织和阅读。对程序的执行无影响。



CoordOffset

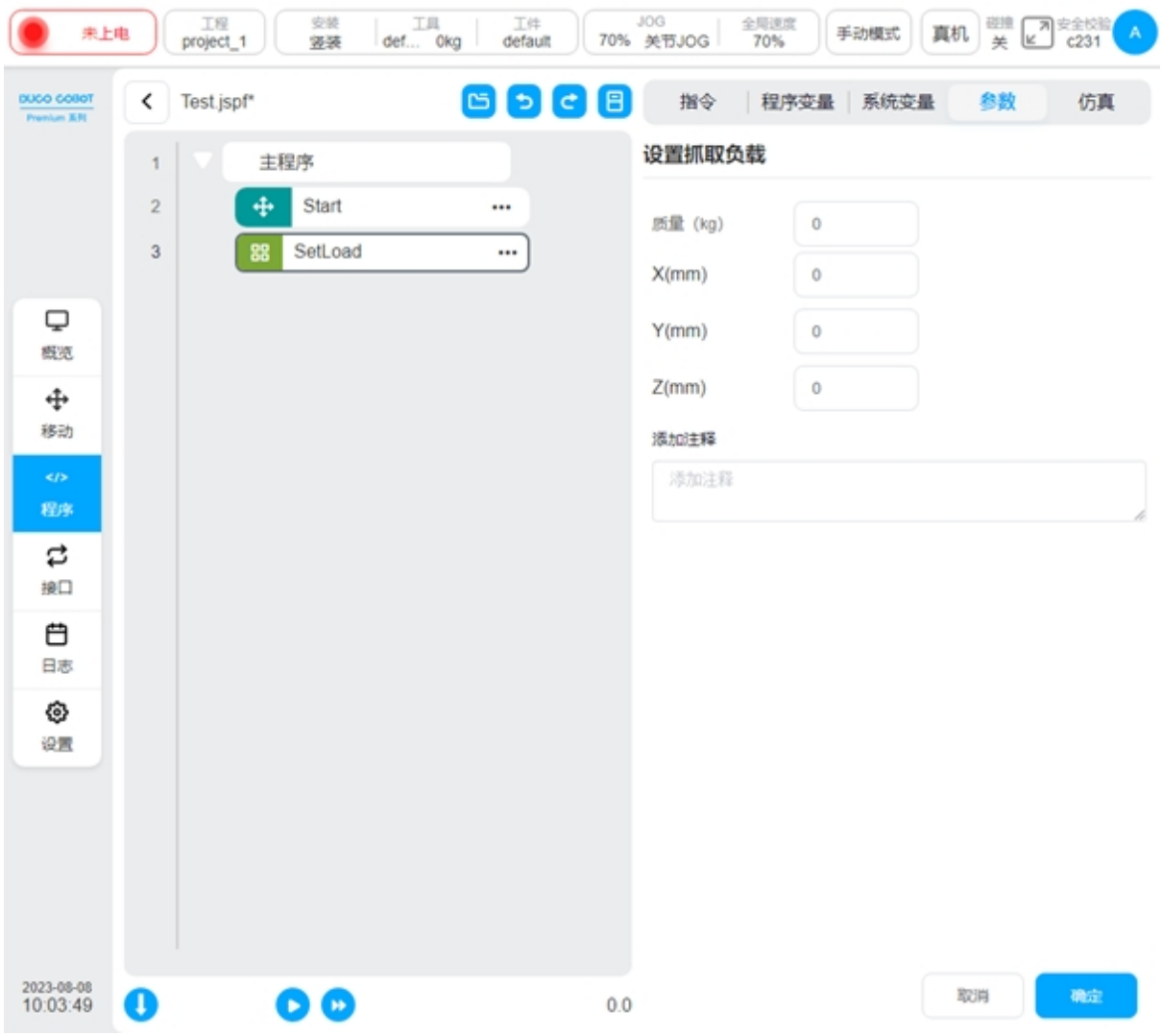
坐标系偏移功能块，基于工件坐标系设置一个偏移量，可以自定义或设定变量。后续的 Move 类功能块的参考工件坐标系上都将添加这个偏移量。在程序运行过程中生效，程序停止坐标系偏移取消。



备注： 使用 `CoordOffset` 功能块对工件坐标系进行偏移，在程序停止时会自动对其清空。在程序执行过程中，若对程序进行暂停并手动示教机器人，此时机器人针对笛卡尔运动所参考的工件坐标系为使用 `CoordOffset` 功能块偏移后的工件坐标系，该偏移将保持到程序手被手动停止或完成运行。

SetLoad

设置抓取负载功能块。可以在程序运行过程中设置机器人当前的负载（质量、质心）



GetListElement

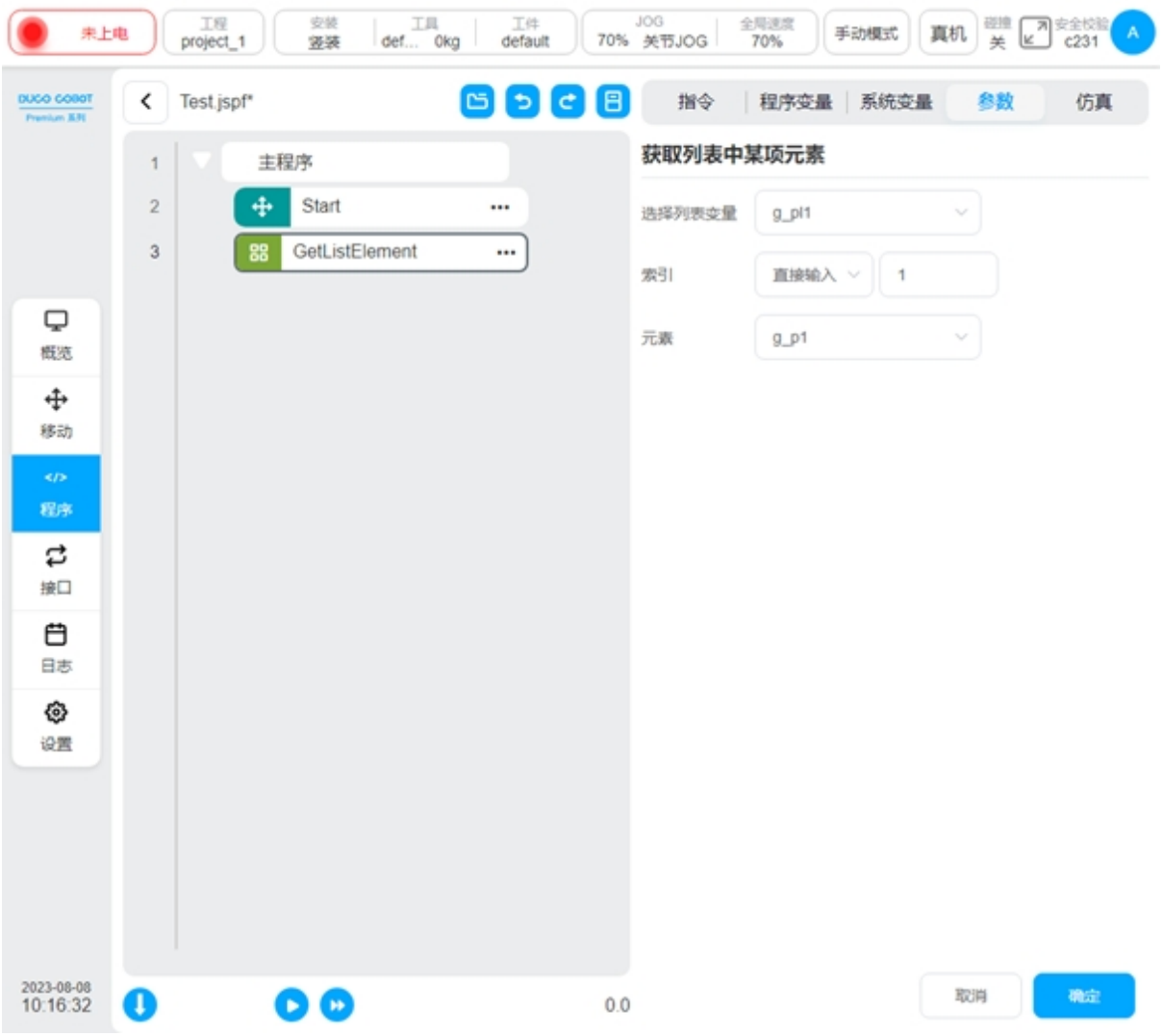
获取列表元素功能块。该功能块用于从 `list` 类型变量中获取其中的一个元素，并将该元素的值转存至指定的变量中。

选择列表类型：当前支持 `num_list,pose_list,joints_list,` 三种类型的变量。使用时，需要先建立期望的 `list` 类变量。

索引：即期望获取的元素在列表中的位置，从 `1` 开始计数。支持直接输入索引值，或从变量获取索引值。

元素：指定获取的元素需要转存到的对应变量。支持 `number, pose, joints` 三种类型的变量。使用时，需要先建立期望的变量。

备注： `GetListElement` 功能块，不会改变原 `list` 类型变量中的数据。即不会把获取的元素删除，做的是复制操作，不是剪切操作。转存的变量类型需要与 `list` 类型变量中的元素类型一致。例如，从 `pose_list` 变量中，获取的 `pose` 类型变量，无法存入 `joints` 类型变量中.



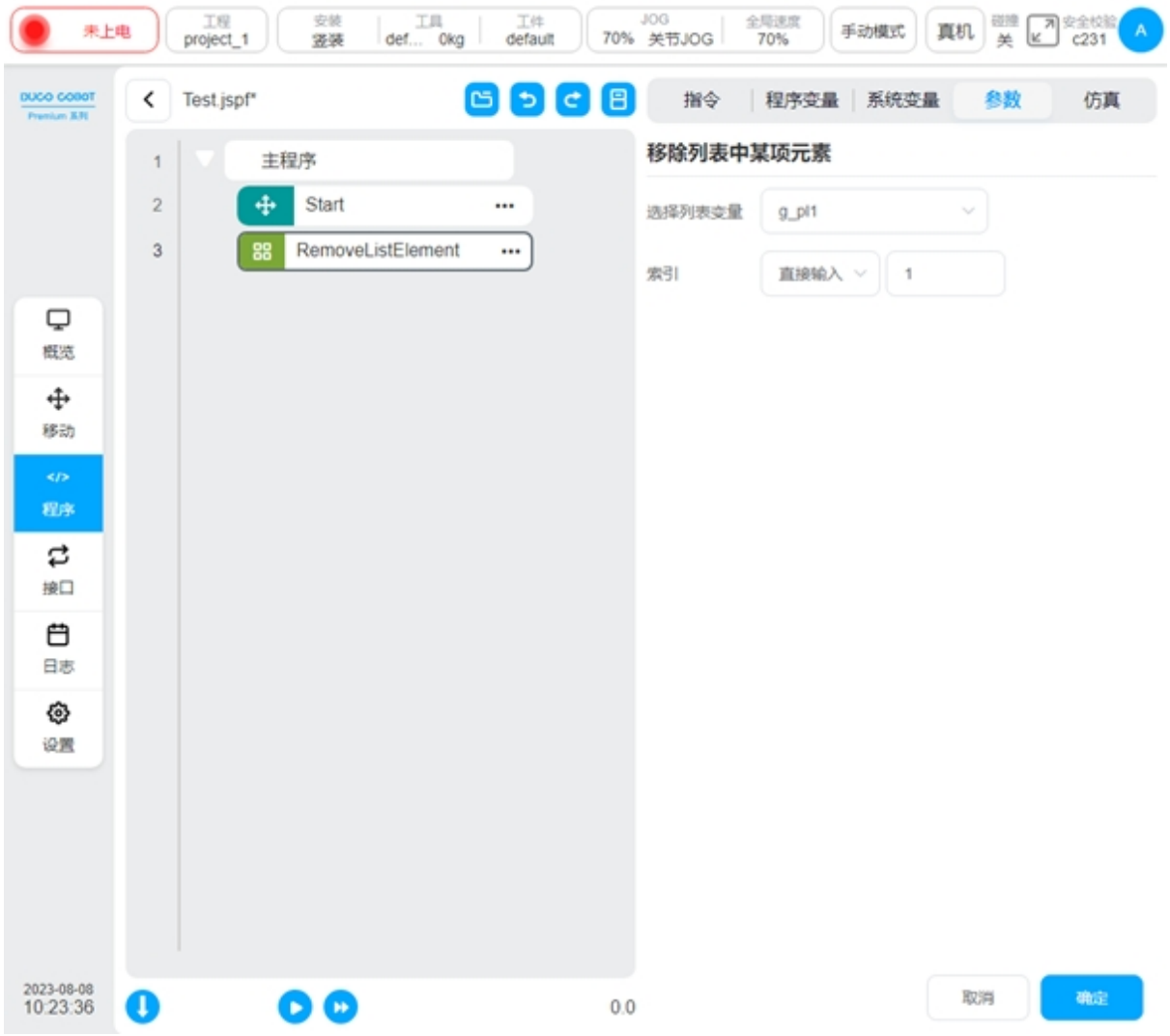
RemoveListElement

移除列表元素功能块。该功能块用于将 list 类型变量中的某一个元素，从该 list 种删除。

选择列表类型：当前支持 num_list,pose_list,joints_list, 三种类型的变量。使用时，需要先建立期望的 list 类变量。

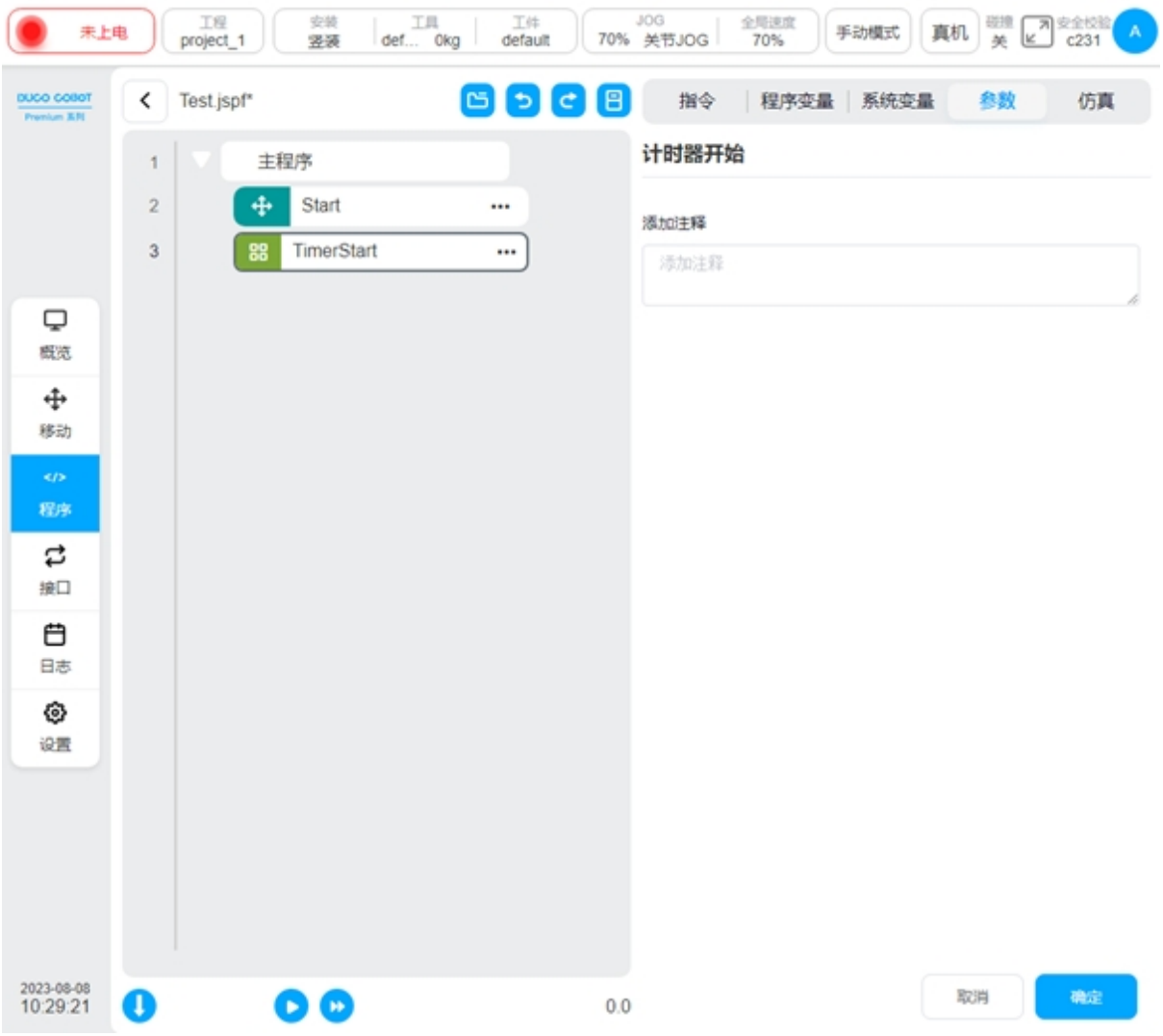
索引：即期望移除的元素在列表中的位置，从 1 开始计数。支持直接输入索引值，或从变量获取索引值。

备注： RemoveListElement 功能块，会改变原 list 类型变量中的数据，及索引号之后的元素位置。例如，pose_list 中含有 pose 类变量 {A,B,C,D}，当使用 RemoveListElement 功能块，移除了其中的变量 B，则新的 pose_list 中的元素变为 {A,C,D}，而不是 {A, null, C, D} 其中 null 表示有 pose 类型变量，但值为空。



TimerStart

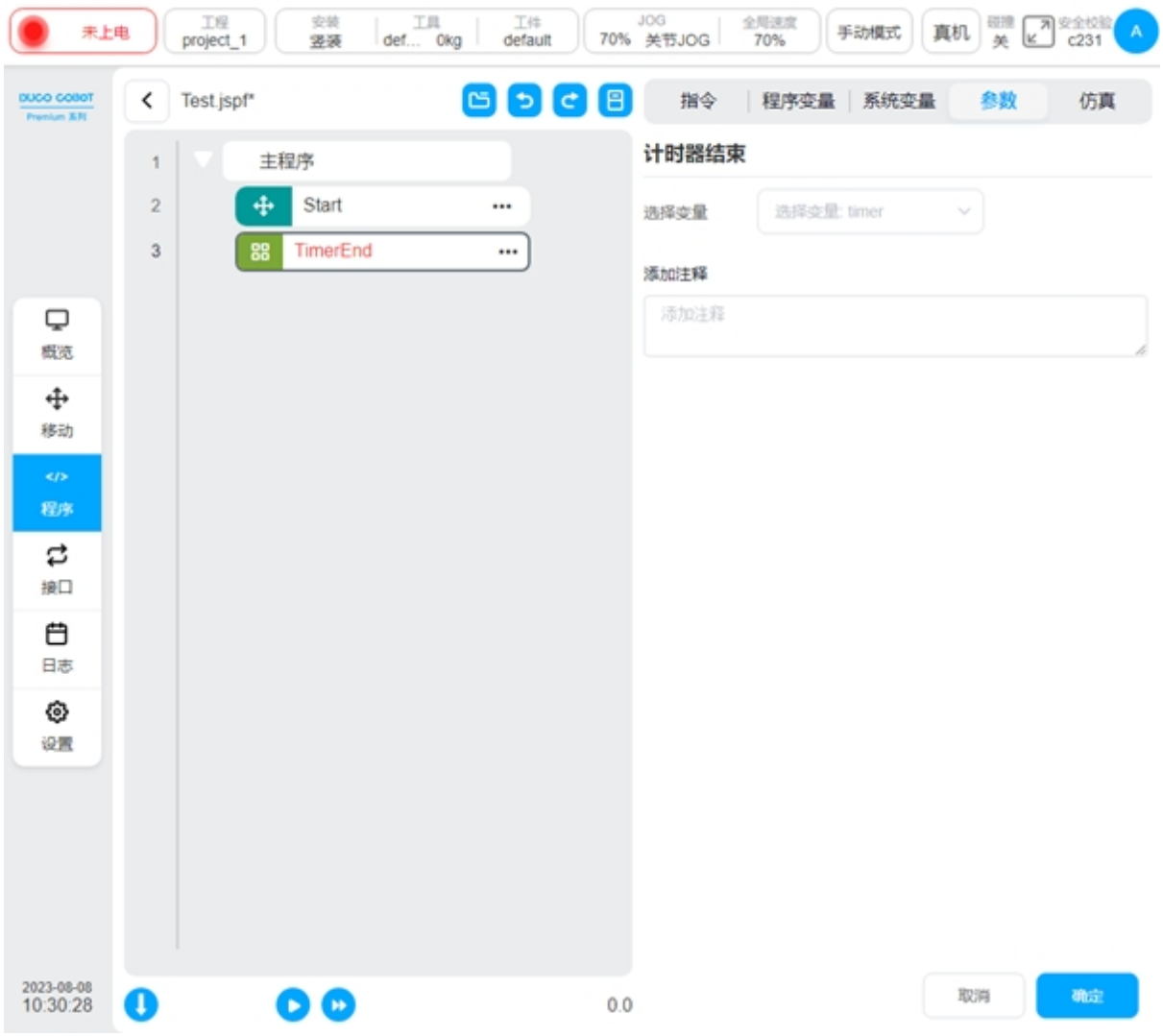
计时开始功能块。该功能块用于计时器的开始计时



TimerEnd

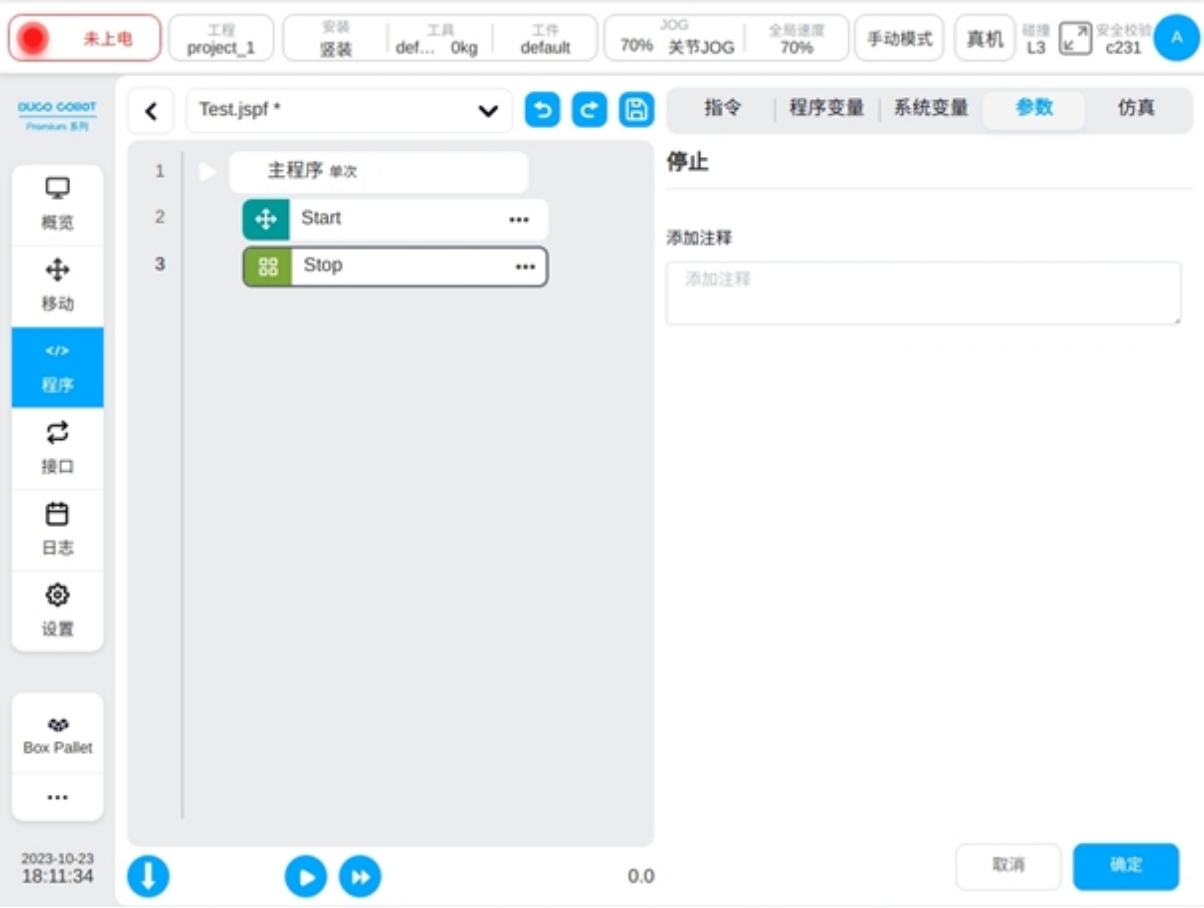
计时结束功能块。该功能块用于计时器的结束计时。

选择变量：仅支持全局 timer 类型的变量。使用时，需要先建立全局的 timer 类变量。



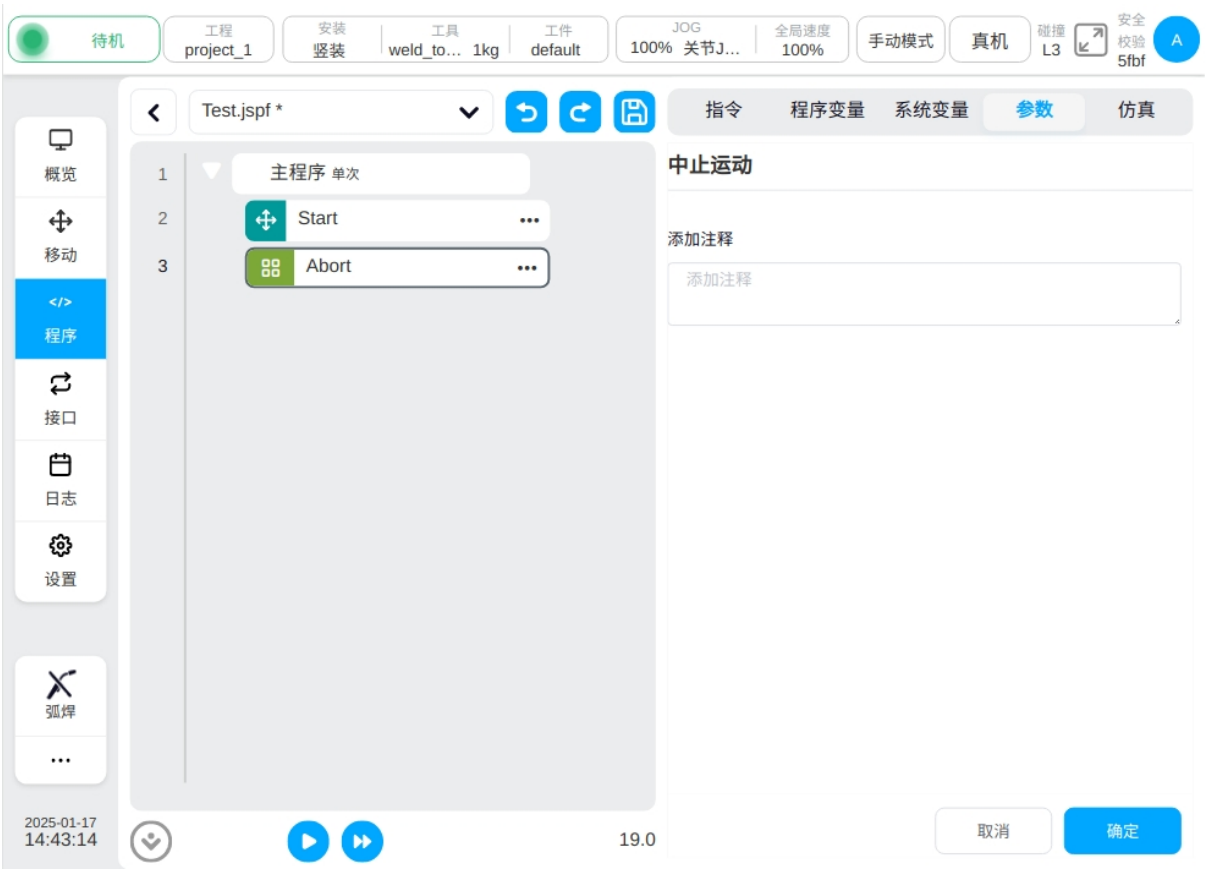
Stop

停止功能块。该功能块用于发出程序停止指令。



Abort

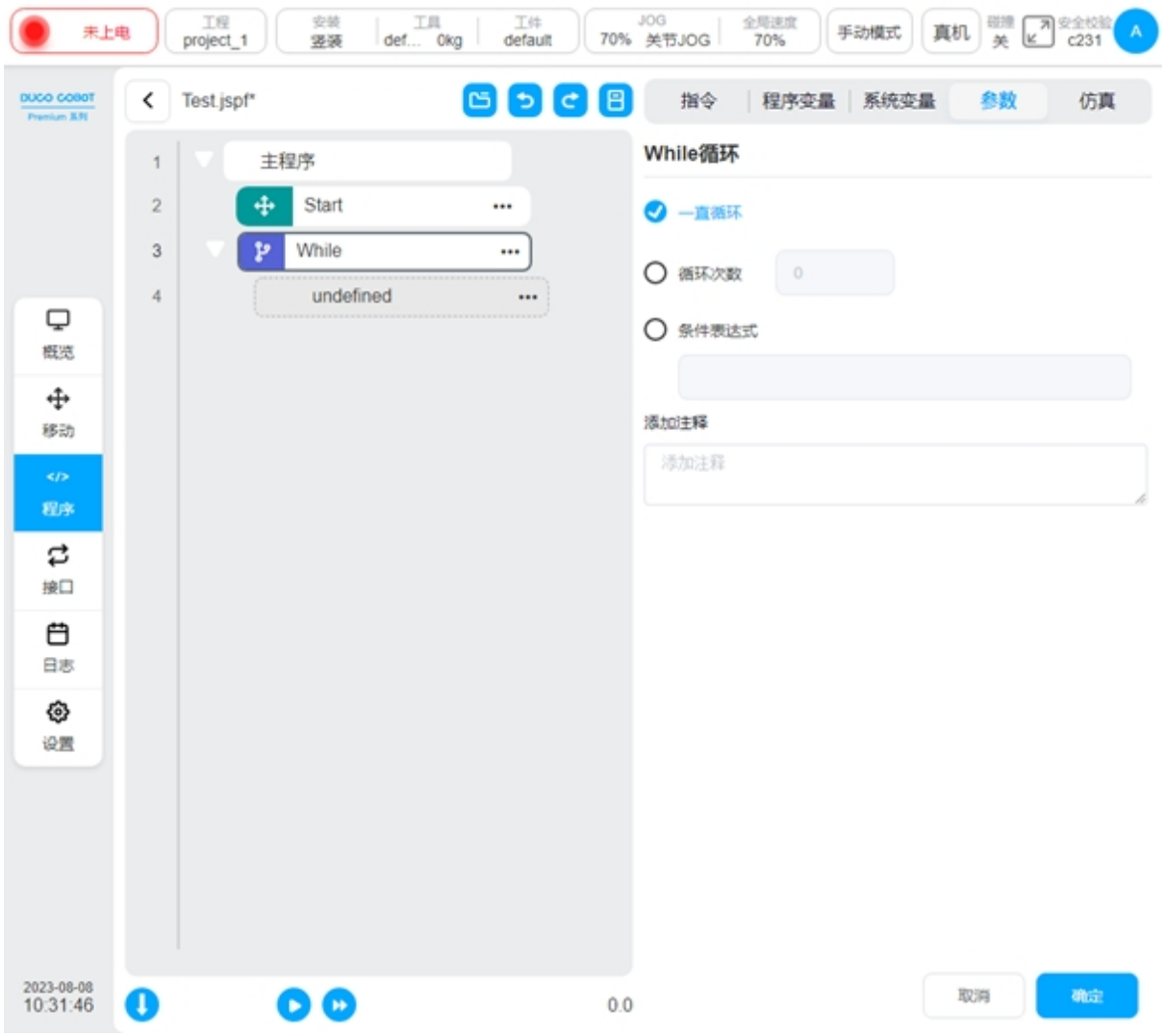
中止运动任务功能块。该功能块用于发出中止当前运动任务指令。



流程控制

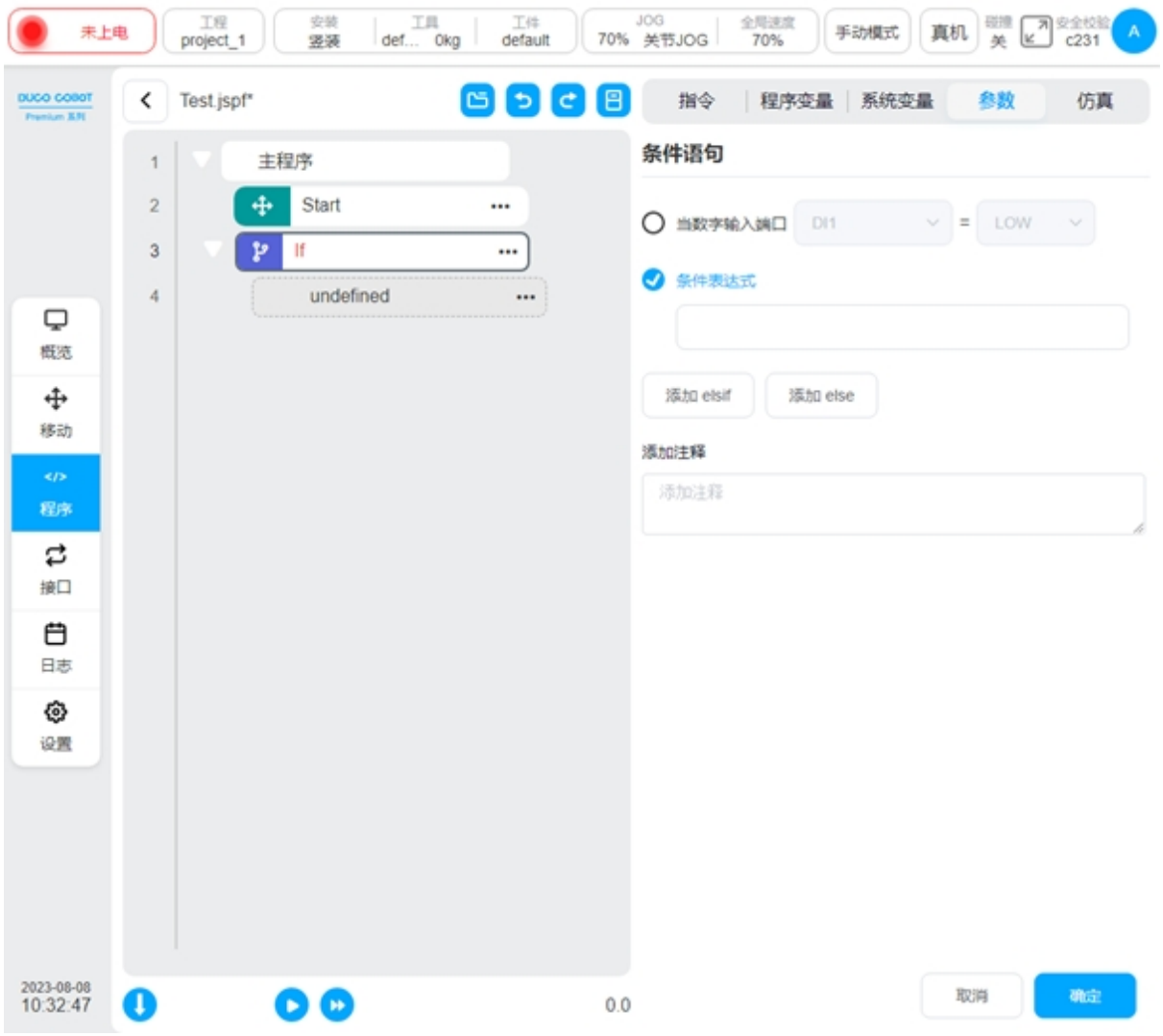
While

循环功能块。循环执行其内部的功能块。可以设定一直循环；指定循环次数；指定循环条件，只要循环条件为真，就循环运行。



If

条件功能块。可以设定数字输入端口条件或者条件表达式，满足该条件则执行 `If` 内的功能块，可以添加后续的 `elsif` 或者 `else` 功能块

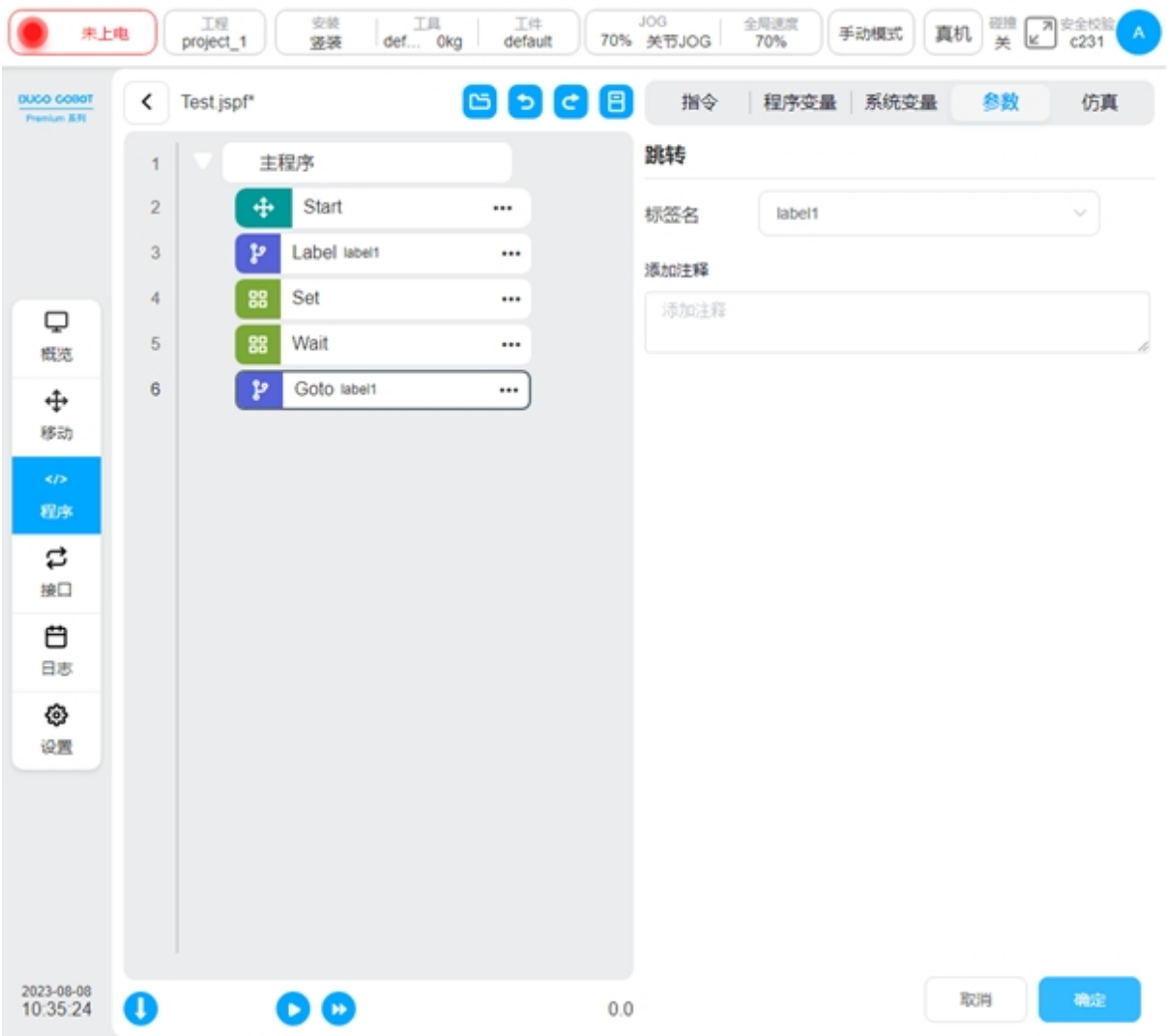


Goto&Label

Goto 功能，和 Label 配合使用，可以将程序的控制点转移到 Label 处。

首先创建 Label 功能块，标记跳转点，并对该跳转点命名。再使用 goto 功能块，选择要跳转到的位置。

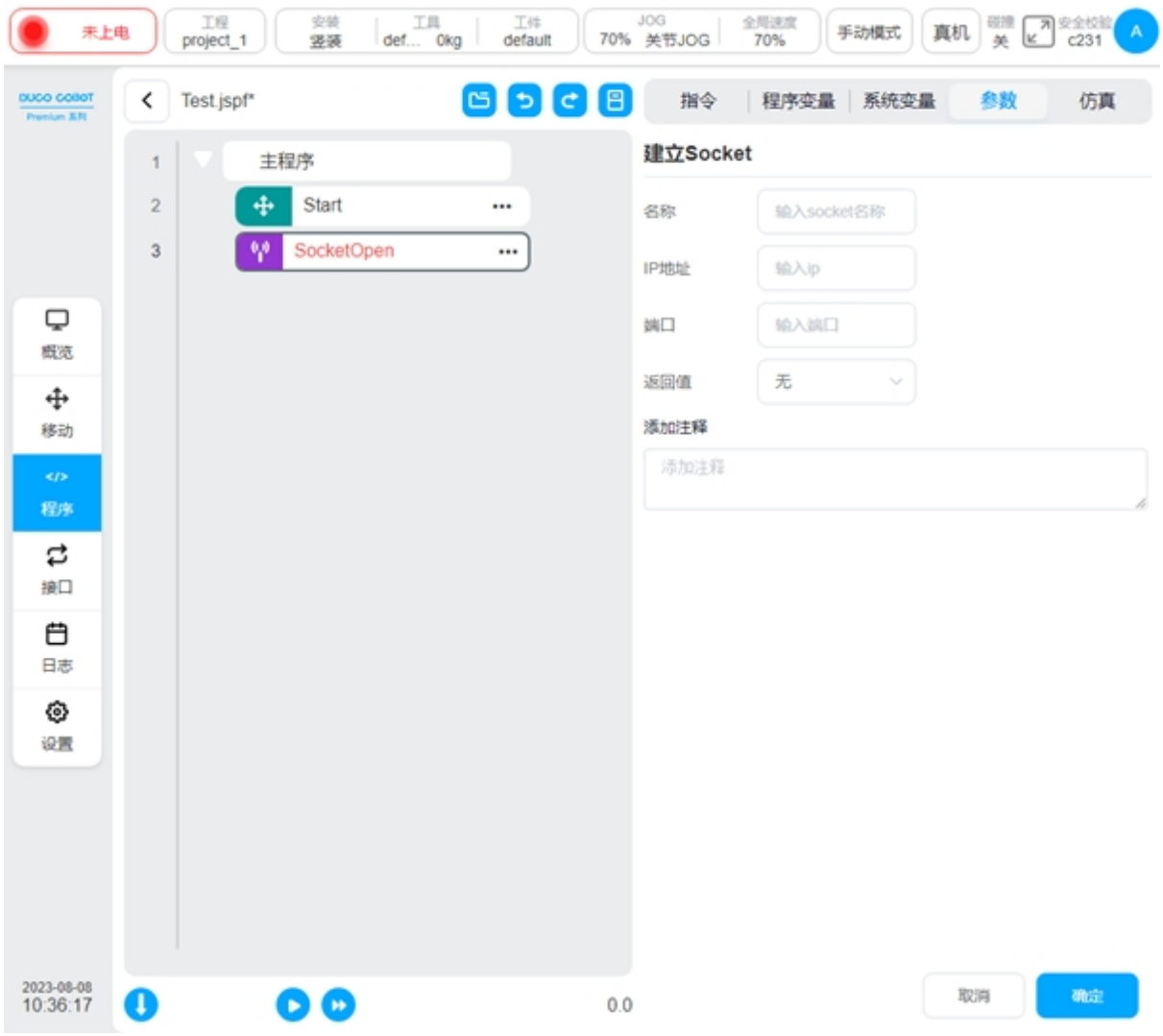
如图的示例程序表示程序运行到第 6 行时将跳转到第 3 行的 label1 处再向下执行，即第四行 Set 与第五行 wait 功能块将会运行两次。



通讯

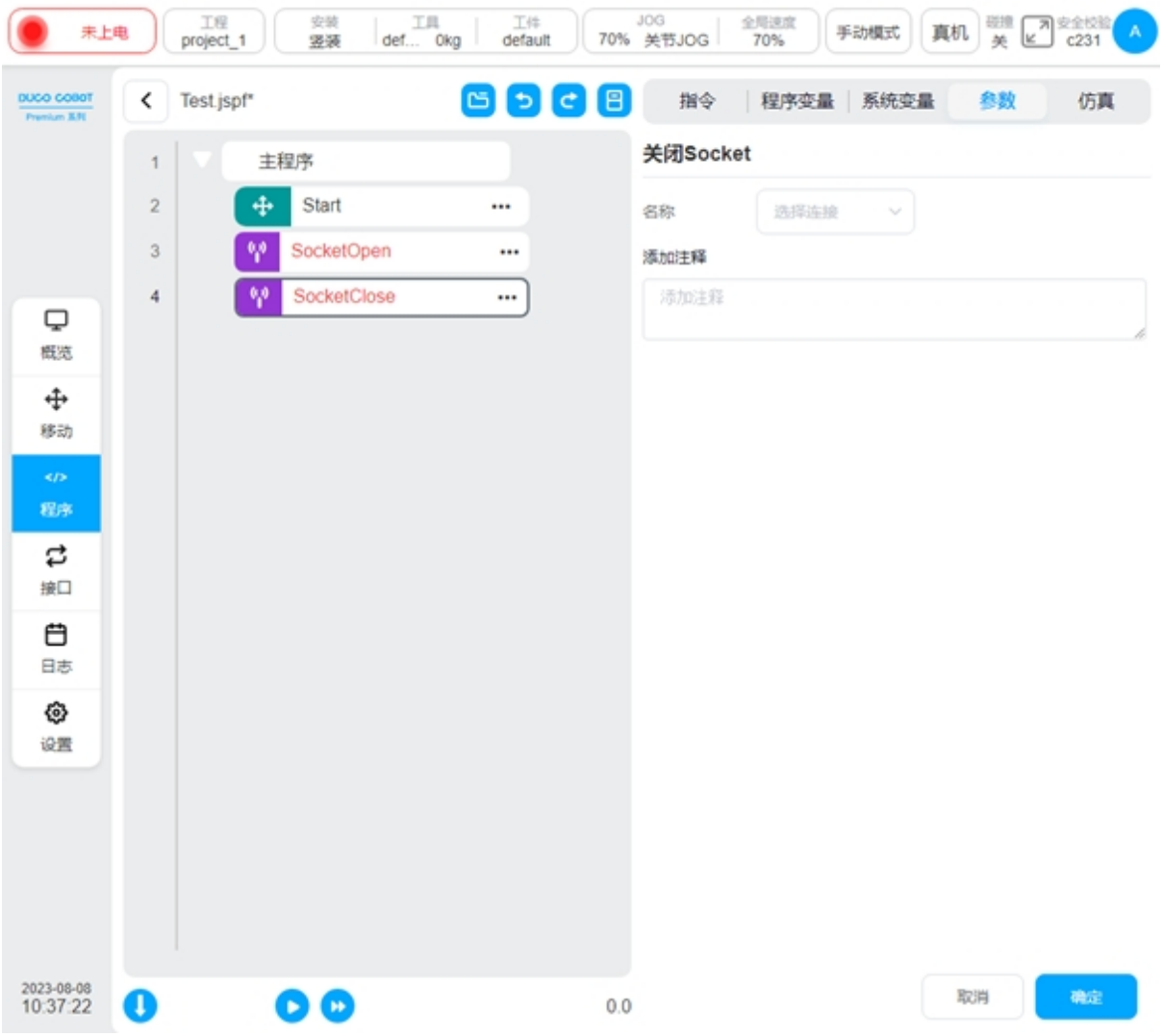
SocketOpen

建立 Socket，参数页面需要设置连接的名称，配置目标 Server 的 IP 地址及端口号，选择是否将返回值绑定到变量



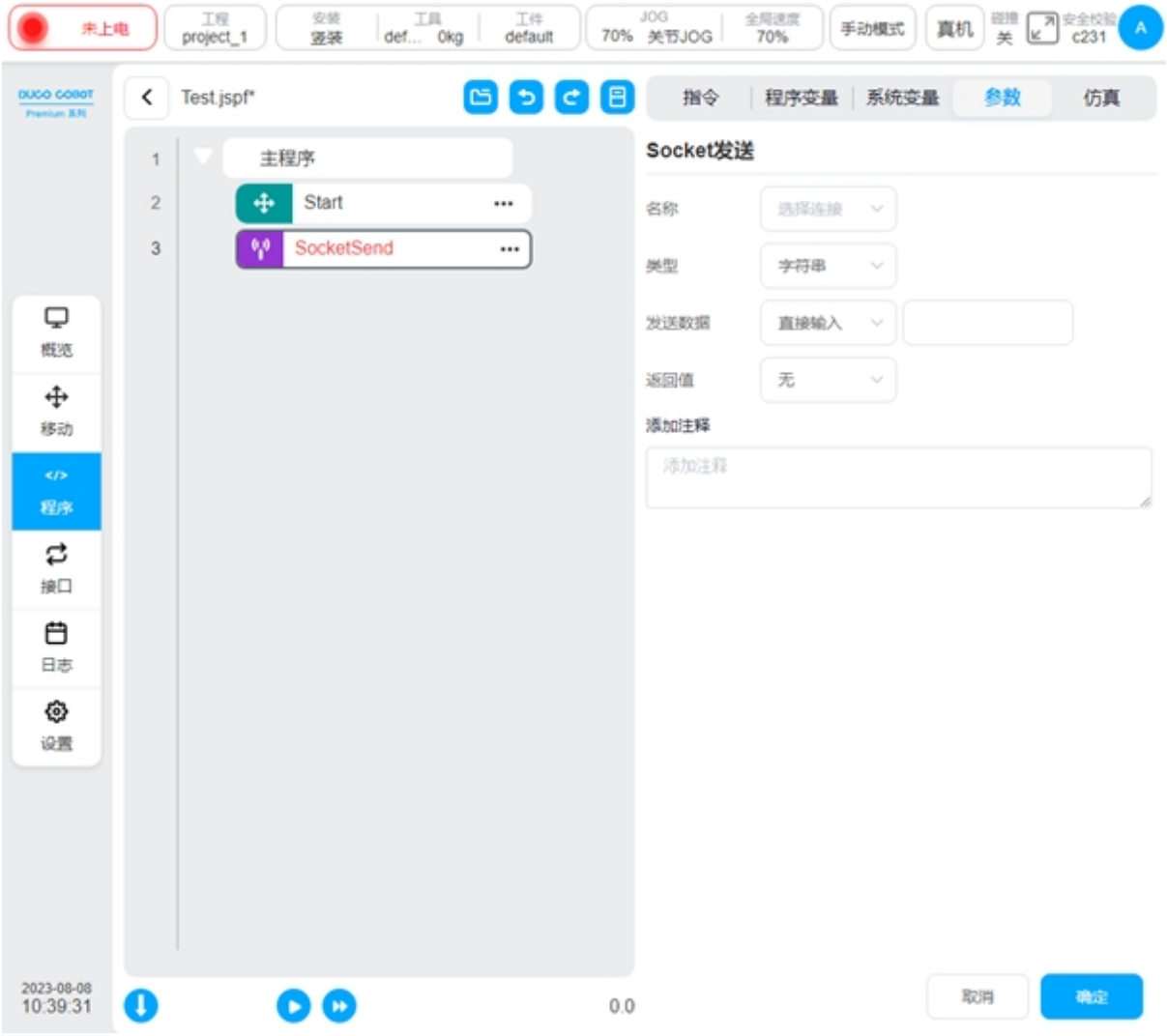
SocketClose

关闭 socket 连接，需要选择关闭连接的名称



SocketSend

socket 发送数据。向已建立的 socket 连接发送数据，发送类型可以选择发送字符串或者浮点数数组，发送数据可以选择变量或者直接输入，可以将返回值绑定到变量以获取发送状态。



SocketRecv

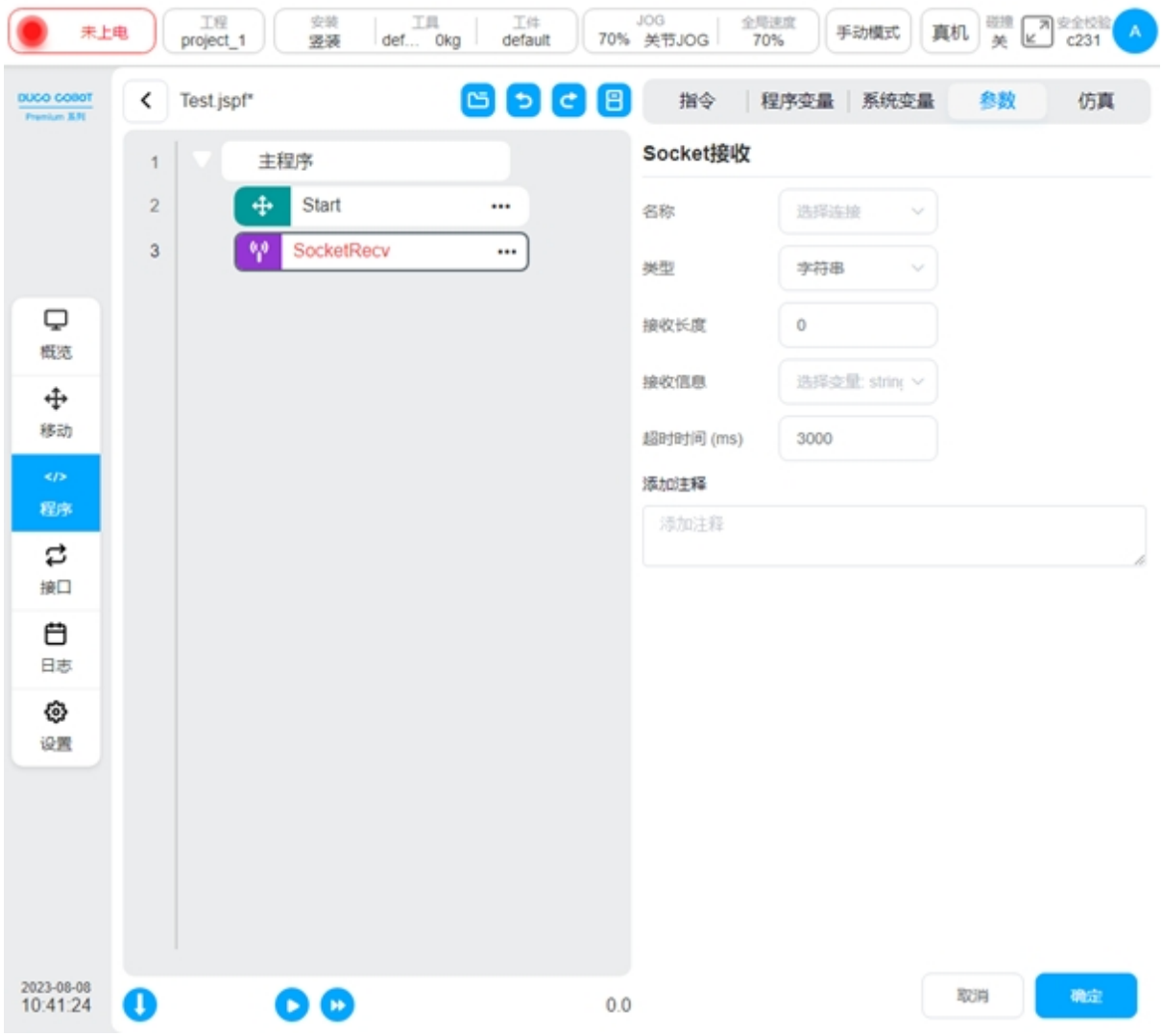
socket 接收数据。从已建立的 socket 连接中接收数据，接收类型可选字符串、字符串数组、浮点数数组，接收信息配置接收变量。

字符串类型时，此时需要给定接收长度，程序会接收该长度的数据并按照字符串来处理，将其保存到“接收信息”配置的变量。

字符串数组类型时，此时程序会将收到的字符串解析成数值，所有数值在“()”内，数值之间使用“,”隔开。例如，从 socket 中收到了一串字符串“(12,1.23)”，该功能块会将其转换为 num_list 类型，值为 {12,1.23}，并将其保存到“接收信息”配置的变量。

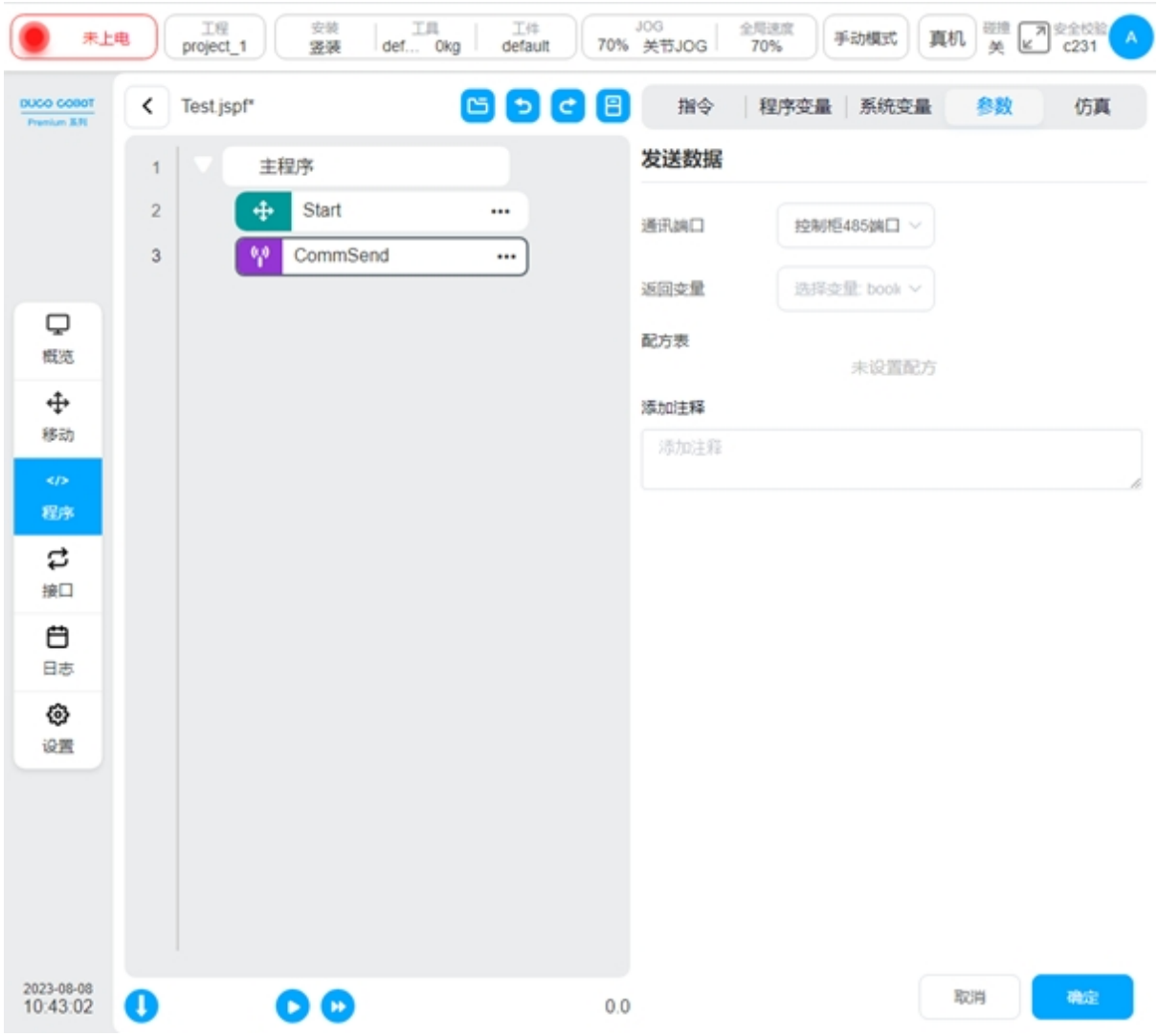
浮点数数组类型时，此时需要给定接收长度，程序会将收到数据转换为一组单精度浮点数，(按照 IEEE 754 标准转换) 并将其保存到“接收信息”配置的变量。

可以配置接收超时时间，在超时时间内未收到符合规则的数据则执行下一条语句。



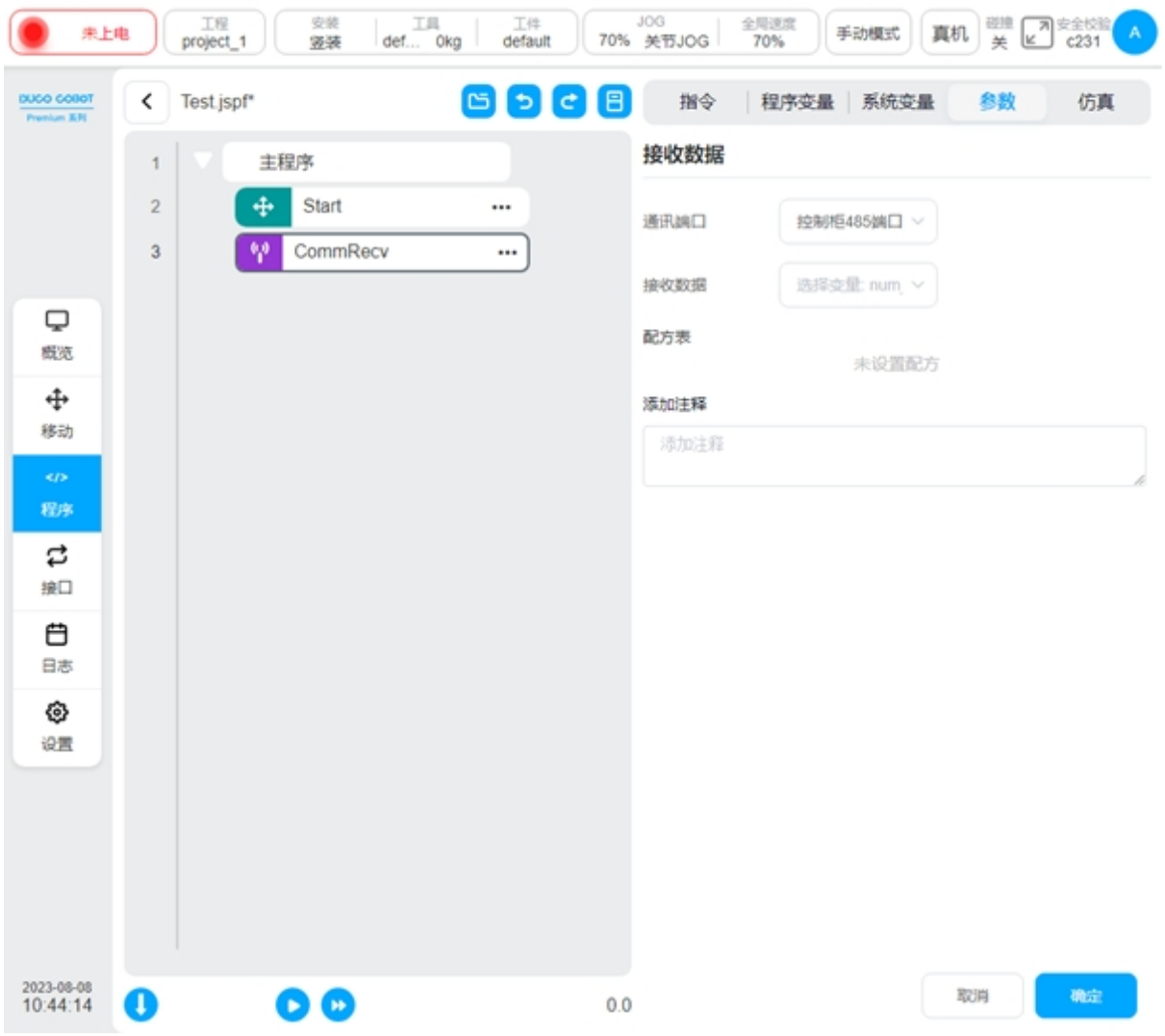
CommSend

配方数据发送。当机器人的 485 端口或者 CAN 端口设置了配方时，使用该功能块可以为配方中的数据设置值，并发送。可以设置返回变量获取配方数据发送的状态。



CommRecv

配方数据接收。当机器人的 485 端口或者 CAN 端口设置了配方时，使用该功能块接收数据，并按照配方处理后，将得到的 num_list 数据赋值到接收变量上。



高级

Subprogram

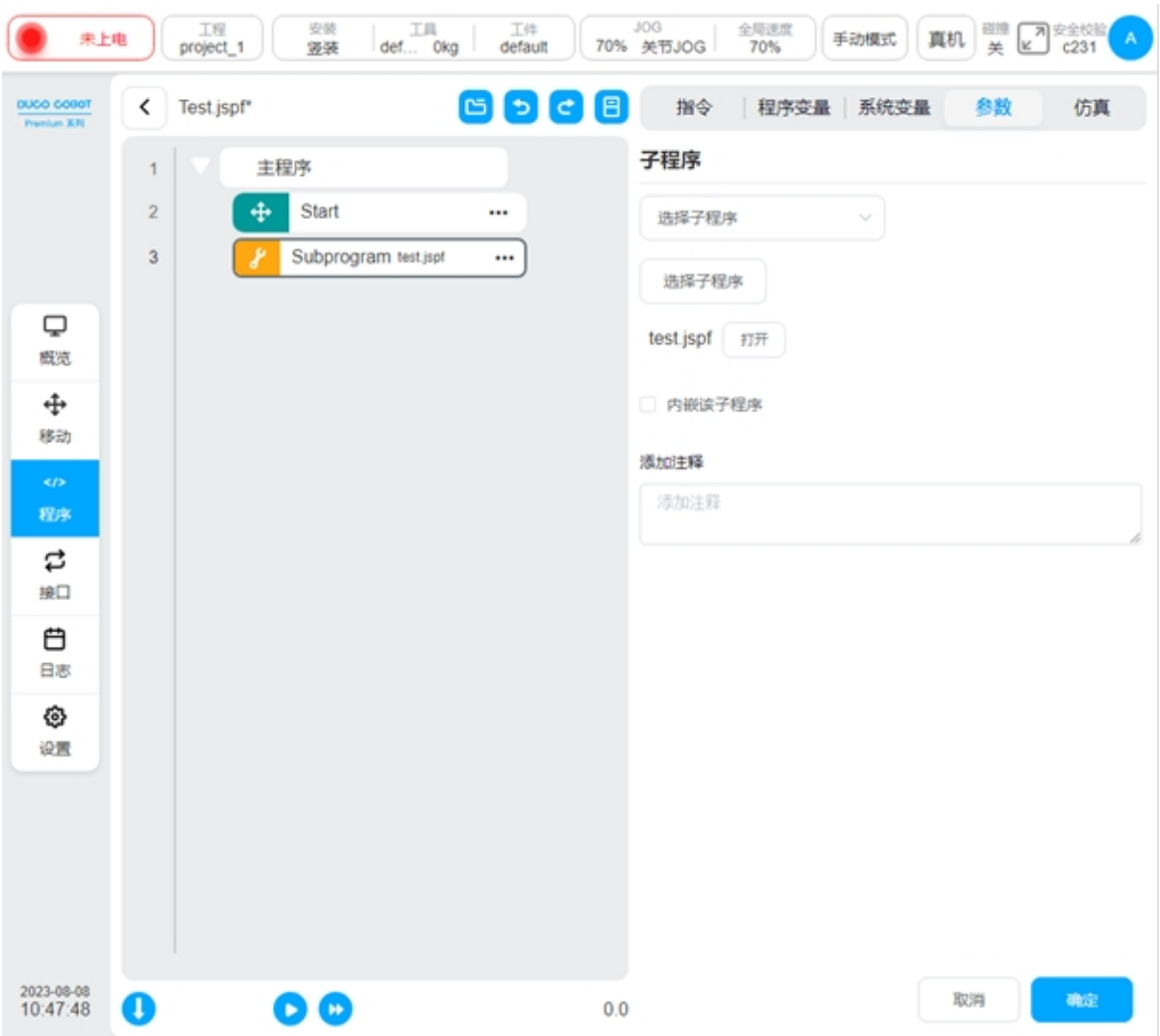
子程序。可以将其他程序嵌入到当前程序中。有以下两种方式：

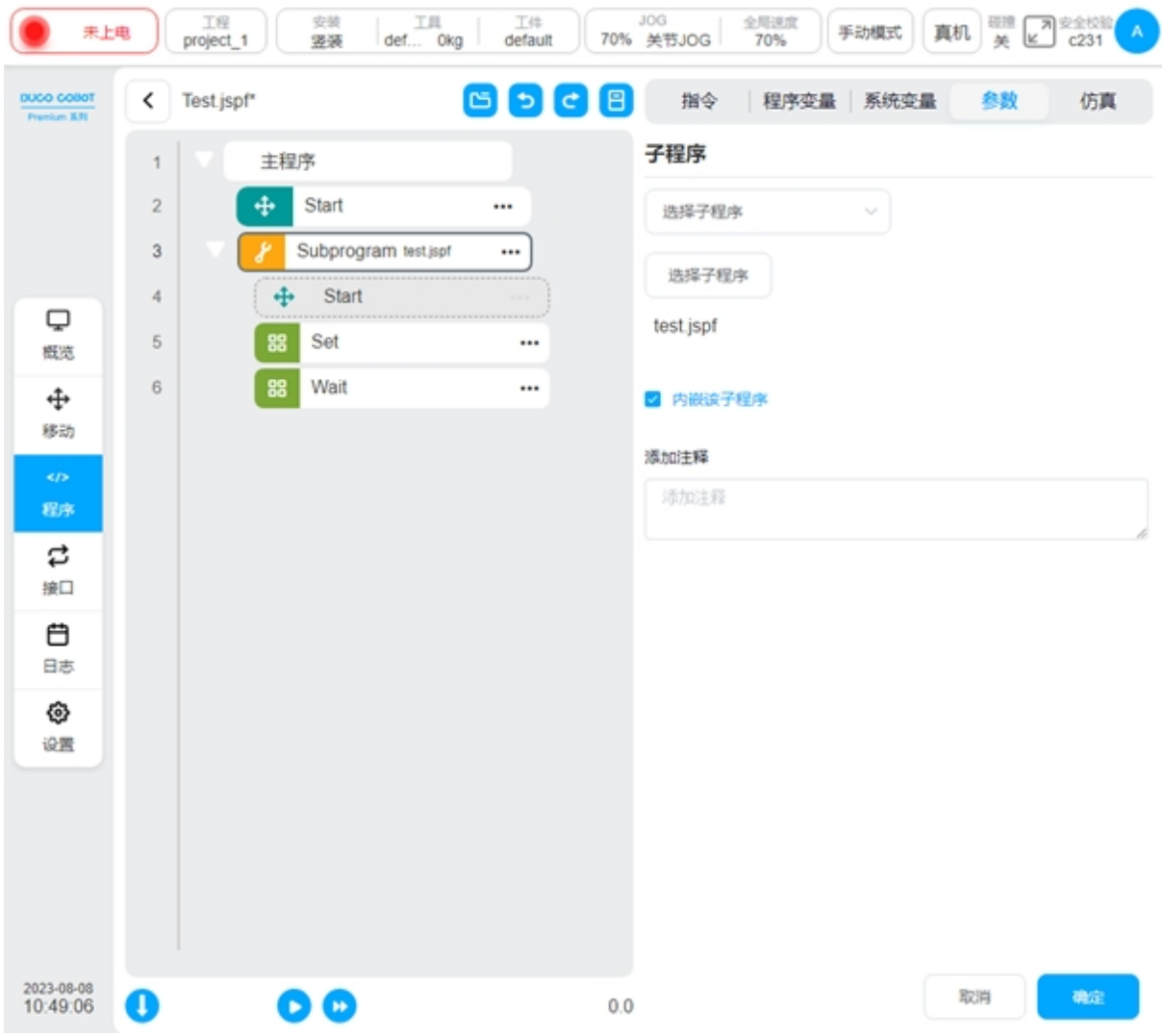
直接从程序列表中选择嵌入的子程序，若不勾选“内嵌该子程序”选项，则每次在程序运行过程中从文件中加载子程序，即子程序文件的改动会影响主程序，且可点击子程序名称后的“打开”按钮，可以直接打开子程序文件进行查看。若勾选，则是将子程序直接复制到了主程序中，此后子程序文件的改动对其无影响，且子程序名称后不显示“打开”按钮。

设定为字符串变量，则程序在运行时，依据变量值作为子程序名来动态加载对应的子程序。

备注： 对于内嵌子程序来说，会将子程序中的程序变量也复制到主程序中，若主程序中存在同名变量时，可能会对主程序有一定的影响。因此使用时，请仔细检查该变量的行为，确保不会影响程序本来的执行逻辑。

使用字符串变量引用子程序时，不得勾选“内嵌该子程序”

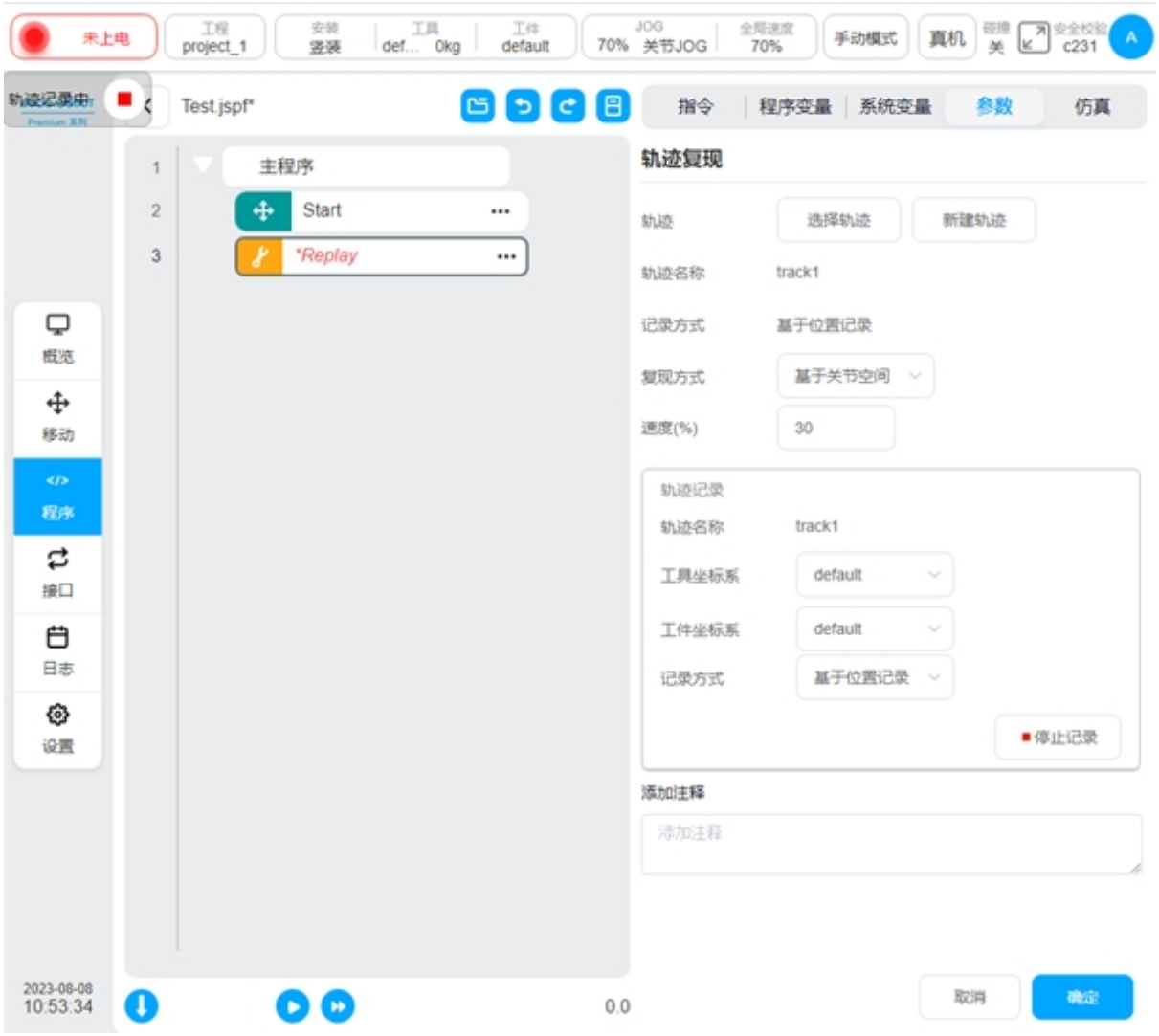

未上电
工程 project_1
安装 竖装
工具 def... 0kg
工件 default
JOG 70% 关节JOG
全局速度 70%
手动模式
真机
碰撞 关
安全校验 c231
A
DUCO COBOT
Test.jspf*
指令
程序变量
系统变量
参数
仿真
1 主程序
2 Start
3 Subprogram test.jspf
子程序
选择子程序
选择子程序
test.jspf 打开
内嵌该子程序
添加注释
添加注释
概览
移动
程序
接口
日志
设置
2023-08-08
10:47:48
0.0
取消
确定



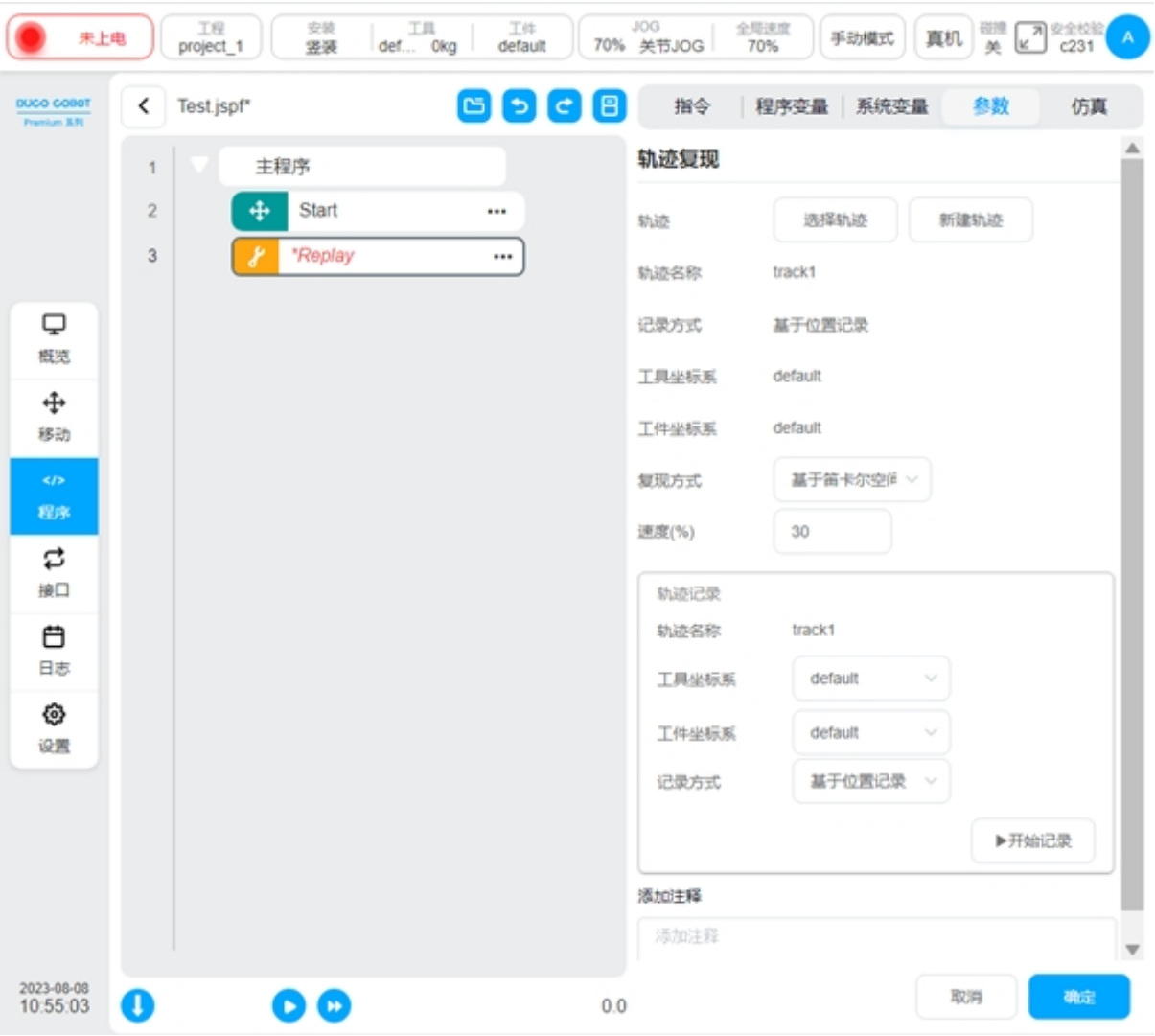
Replay

轨迹复现。可以创建一条记录方式为基于位置记录或基于时间记录的机器人运动轨迹，程序执行到此功能块时机器人按照基于关节空间或基于笛卡尔空间的复现方式来运动该轨迹。可以选择已有的轨迹文件或者创建新的轨迹。

创建新轨迹的方法，点击“新建轨迹”，输入轨迹名后会显示“轨迹记录”框，选择好工具坐标系、工件坐标系、记录方式后点击“开始记录”则开始记录轨迹数据，此时页面会显示一个半透明的悬浮框表明正在记录轨迹，用户可以使用牵引或者点动等方式移动机器人，点击“停止记录”或悬浮框上的停止按钮，则完成该轨迹文件的创建。轨迹记录过程中，用户修改坐标系或者切换当前坐标系，终止当前轨迹记录，“轨迹记录中”悬浮框消失。

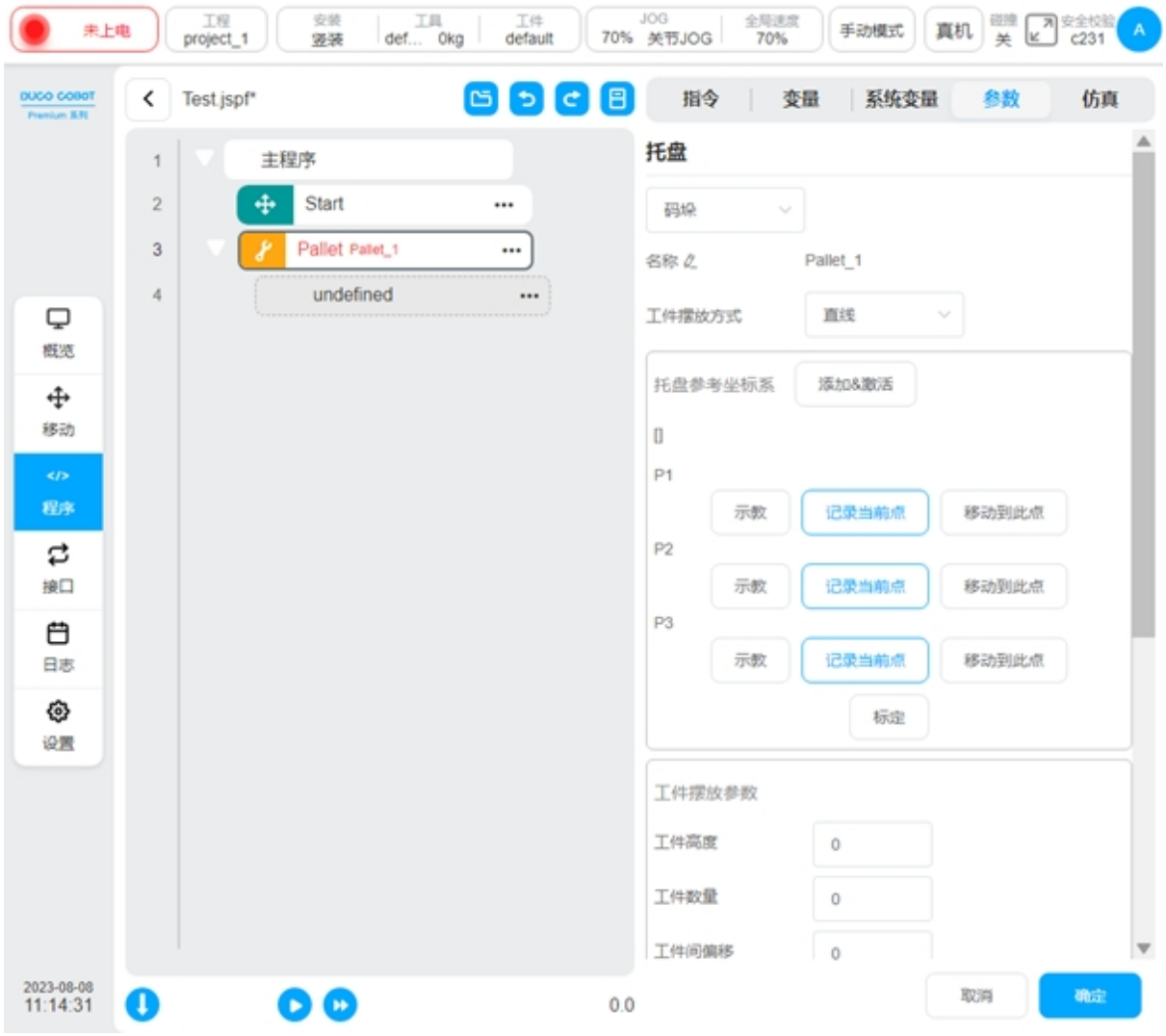


对于选取的轨迹，当复现方式为基于关节空间时，会显示该轨迹的轨迹名称、记录方式；当复现方式为基于笛卡尔空间时，会显示该轨迹的轨迹名称、记录方式以及记录时刻的工具坐标系和工件坐标系。



Pallet

码垛/卸垛。可以通过一些简单的参数设置，自动载入一组码垛/卸垛标准程序模板，并在此基础上进行适配性改动。



按照如下步骤使用码垛/卸垛功能块：

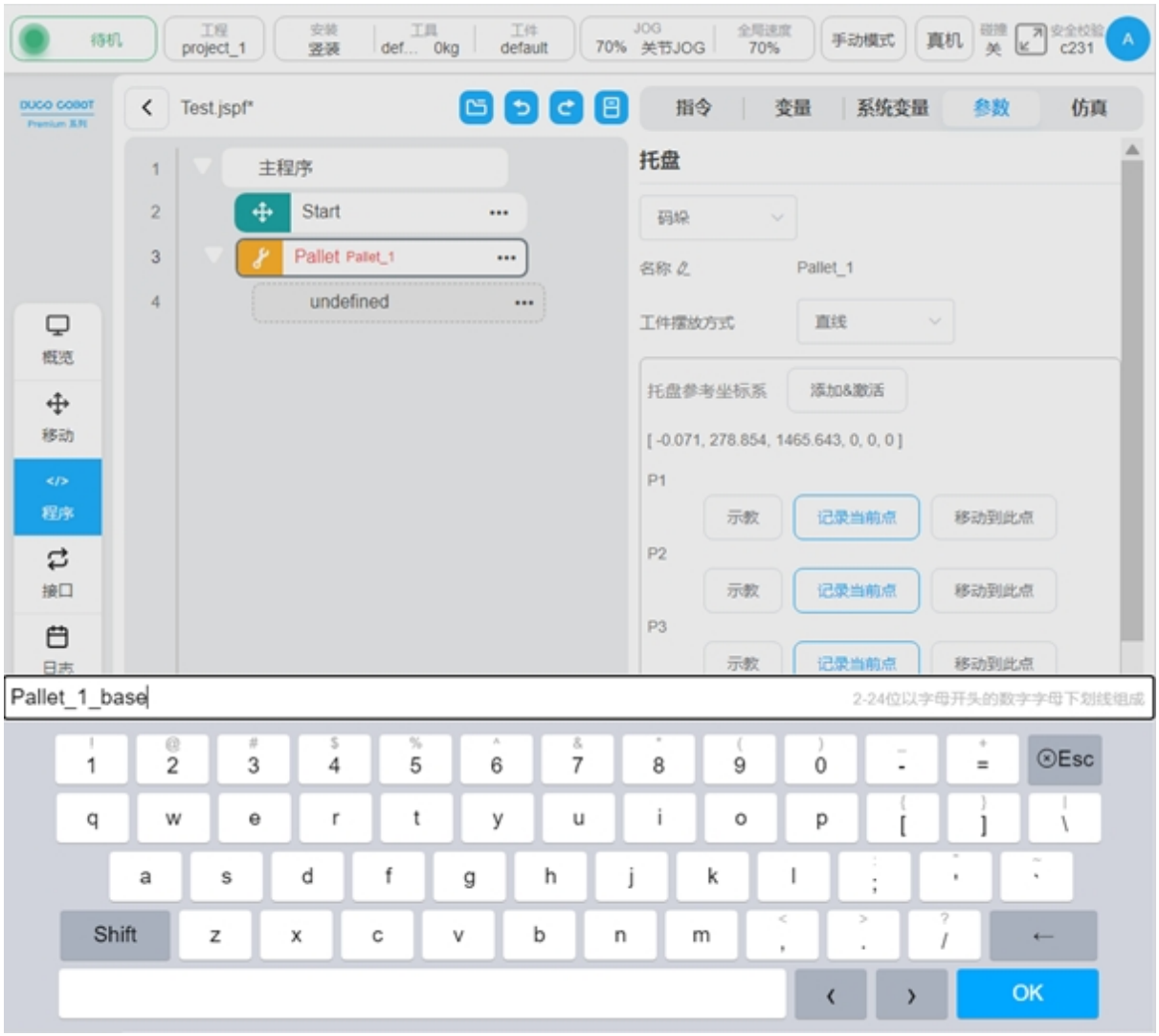
1、确定使用码垛还是卸垛，码垛为将工件移动到托盘上，卸垛为从托盘上移走工件

2、可以指定名称，每次添加 Pallet 时会生成一个默认的名称

3、选择工件在托盘上的摆放模式，直线或者网格模式

4、标定托盘使用的参考坐标系，标定方式：基于托盘上摆放的第一个工件来标定，P1 点为示教到工件上的一点，P2 点示教到行方向（参考坐标系 X 轴方向），P3 点示教到列方向。点击“标定”按钮完成坐标系的标定。上方将显示参考坐标系的值。

点击“添加 & 激活”按钮，将弹出对话框，输入该坐标系的名称后，将添加到系统中，并设为当前坐标系。



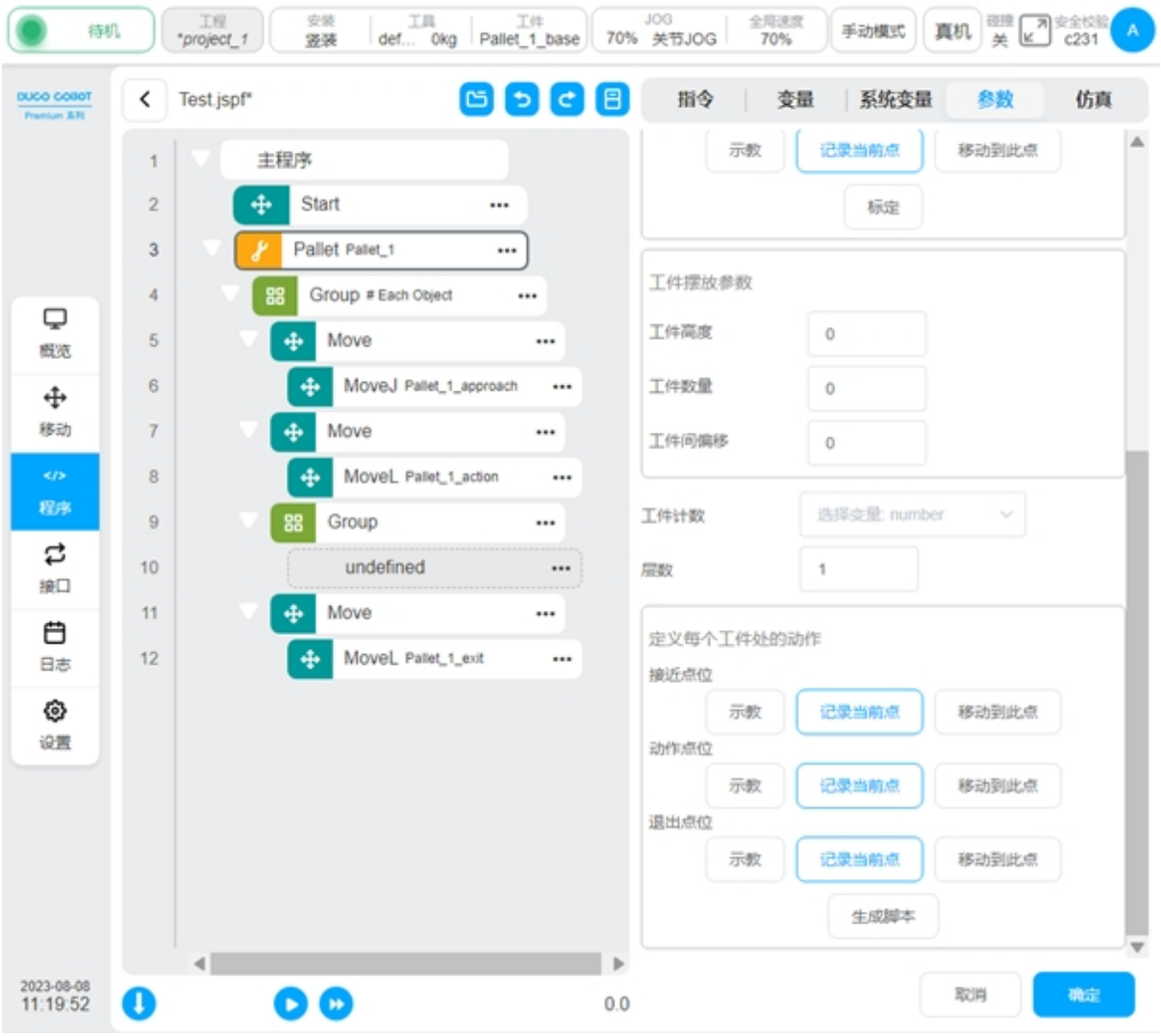
5、设置工件摆放参数，需要设置工件的高度，工件在行及列方向上的摆放数量，工件在行及列方向上的间距

6、设置工件的计数变量，需在程序变量区新建一个类型为 number 的变量

7、设置摆放层数

8、定义在各个工件上执行动作的点位，可分为三个接近点位、动作点位、退出点位。在托盘摆放的第一个工件处示教这三个点位。接近点位为机器人运行到托盘上方计划摆放放/抓取工件的位置；动作点位为机器人摆放/抓取工件的位置；退出点位为机器人在摆放/抓取完成离开托盘时的位置。注意：这三个点位的示教务必在托盘的参考坐标系下进行。

9、示教完成后，点击“生成脚本”按钮，将自动在程序中添加一系列功能块。其中 Group 功能块下可以添加相应的执行动作如夹爪开合等。

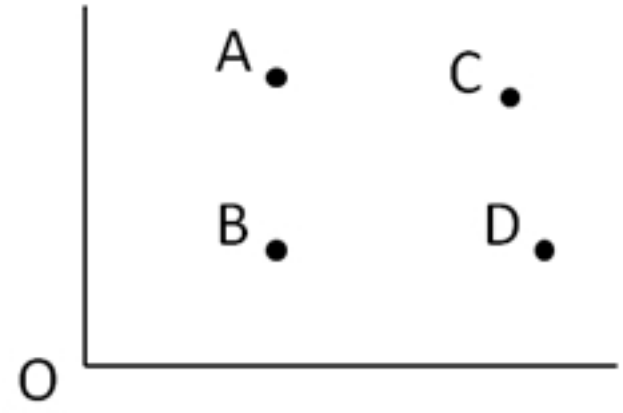


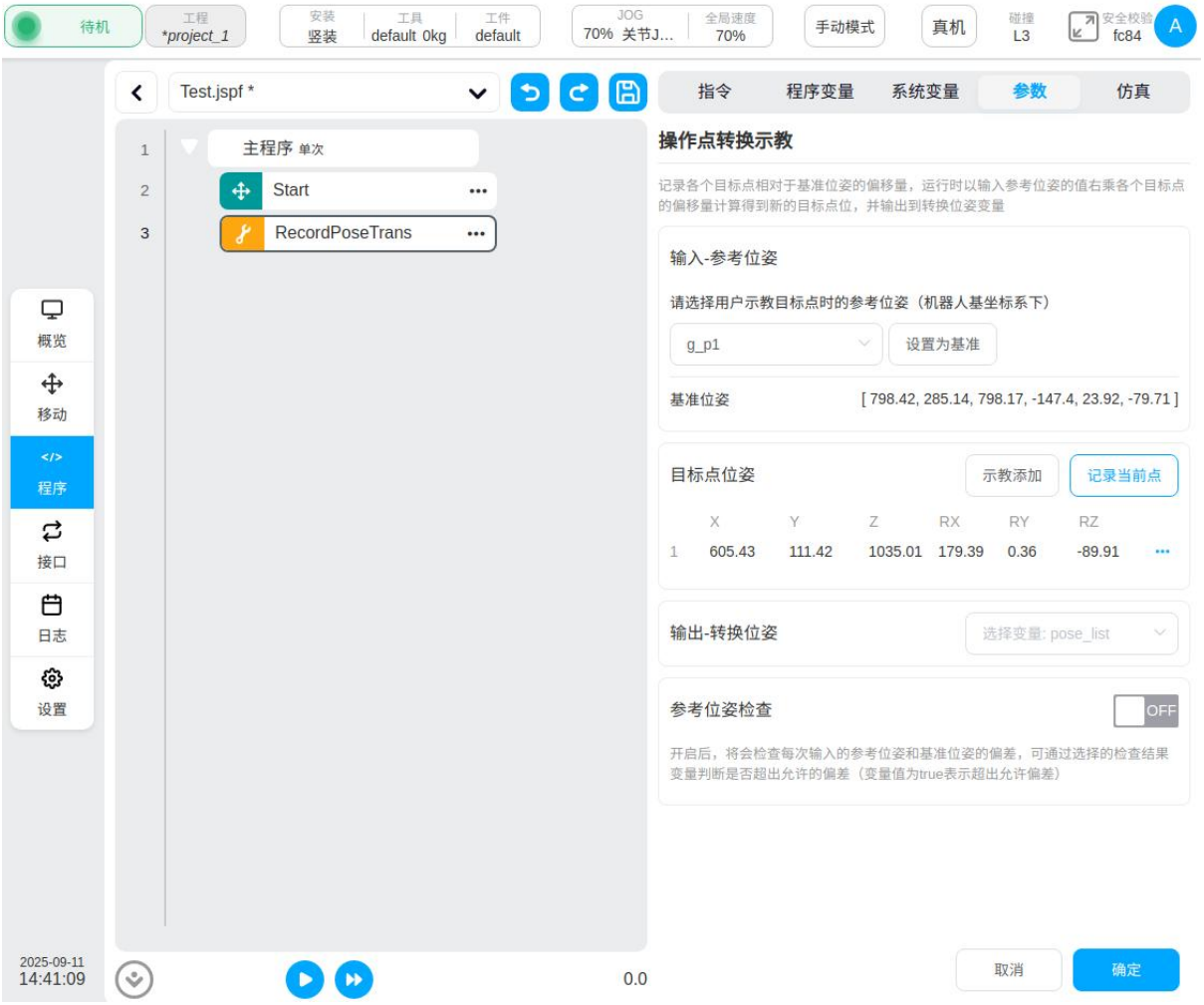
10、程序设置为一直循环方式，码垛程序会在每次执行时，根据设置的参数计算偏移量，确定下一次码垛/卸垛的位置姿态。在变量区，根据所设置的工件计数变量可查看当前码垛/卸垛工件的个数

RecordPoseTrans

操作点转换示教功能块。用来做点位转换。

如下图，点 O 为参考点，A、B、C、D 点相对于参考点 O 的位姿是不变的。操作点转换示教功能块记录这些点位相对于参考点 O 的相对位姿关系，当 O 的位姿发生变化时，输出 A'、B'、C'、D' 点的位姿。常用在视觉识别定位、激光寻位等场合。例如在视觉识别定位场景中，参考点 O 为一个工位的标记块，A、B、C、D 为该工位的四个工件，工件相对于标记块的位姿固定。相机拍照识别得到该标记块在机器人基坐标系下的位姿，使用操作点示教功能块计算得到工件在机器人基坐标系下的位姿。





配置页如图，“输入-参考位姿”中设置参考点位 (O 点)，“目标点位姿”中设置操作点位 (A,B,C,D 点)，“输出-转换位姿”中设置变量存放计算得到的操作点位，即储存 A’,B’,C’,D’ 位姿。“参考位姿检查”按钮，开启后，将会检查每次输入的参考位姿和基准位姿的偏差，可通过选择的检查结果变量判断是否超出允许的偏差（变量值为 true 表示超出允许偏差）。配置阶段，功能块将记录操作点 (A,B,C,D 点) 相对于基准 (O 点) 的偏差；程序运行时，根据当前输入变量的值 (O’点) 加上偏差计算得到操作点 (A’,B’,C’,D’点) 的位姿。

选择一个 pose 类型的全局变量作为参考位姿，点击“设置为基准”，将以该变量当前值作为参考基准 (O 点)。在操作点偏置中点击“示教添加”将跳转到移动页面，可以移动机械臂选取点位 (A,B,C,D)，选取后即新增了一个点位，配置页面会显示添加的点位在基座标系下的

值。输出栏中，选择一个 pose_list 类型变量存放计算结果（A’,B’,C’,D’）。

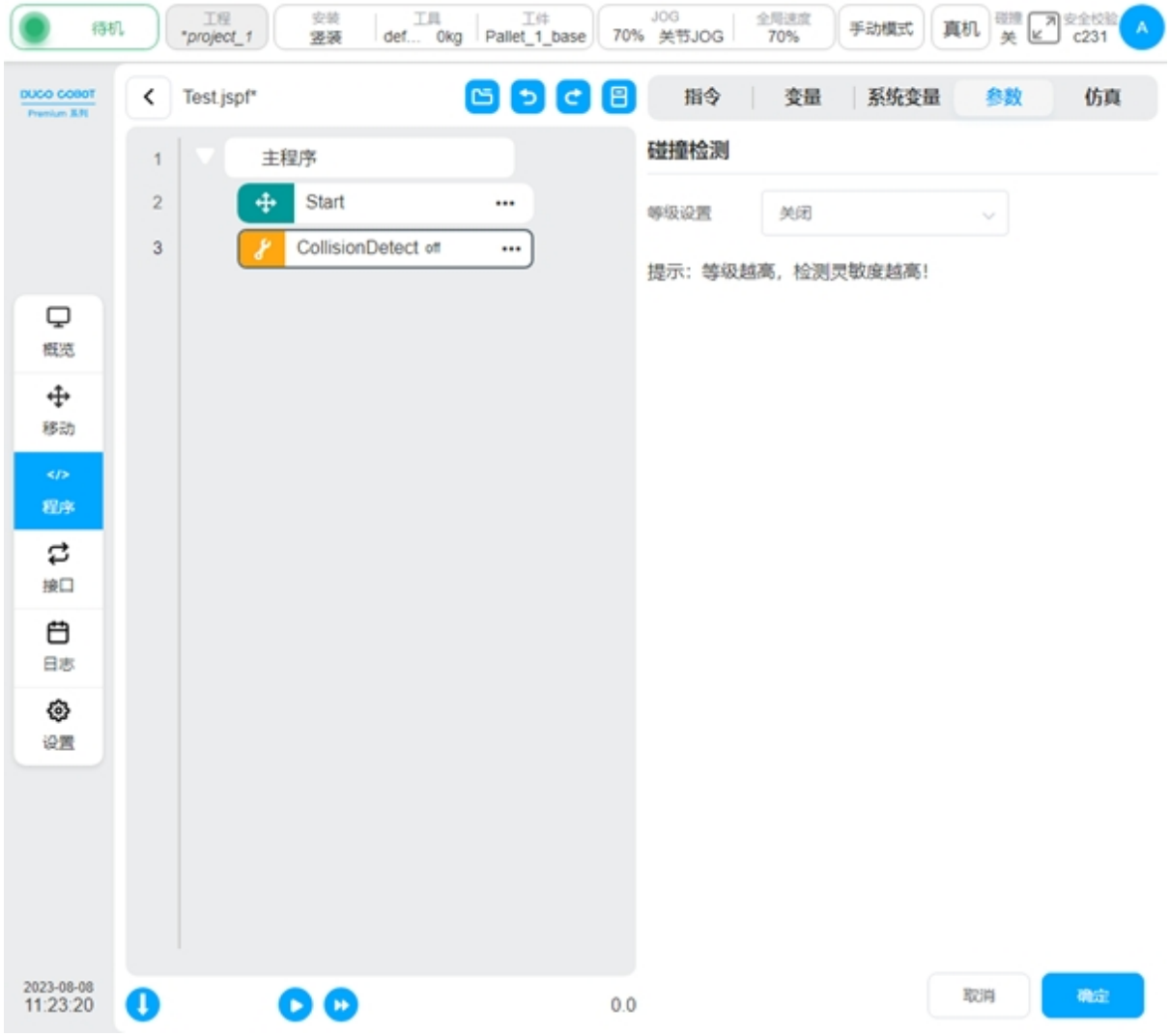
力控相关脚本

当前力控相关脚本包括 Force,ForceGuard,ForceSetZero,ForceWait,ForceMove. 详细的应用介绍请参看《DUCO CORE 末端力控功能操作手册》

CollisionDetect

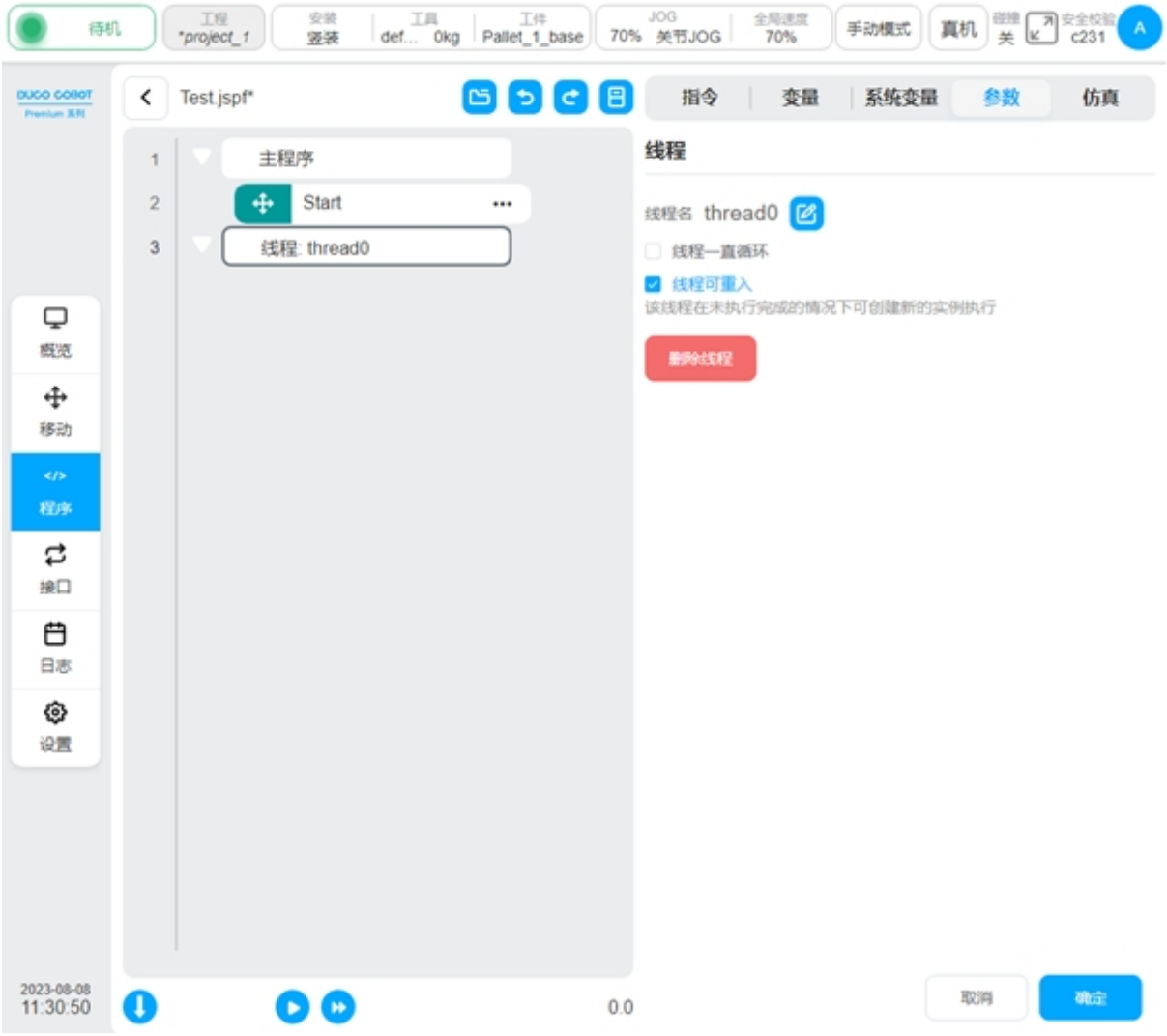
碰撞检测等级。该脚本可以实现在程序运行过程中，对碰撞检测灵敏度的设置。

备注： 该功能块设置的参数仅影响机器人本次开机时的状态，不会断电保存。若需要永久设置碰撞检测灵敏度，请在“设置”——“其他设置"——“碰撞设置”中设置，并保存工程。



线程

线程，与机器人主程序并行执行，通过变量与主程序交互。在参数配置区可设置该线程是否一直循环；还可设置线程是否可重入，即该线程在未执行完成的情况下是否可创建新的实例执行。在一个程序中最多允许创建 10 个线程。



备注： 线程中不允许存在机器人的运动指令。

若在子程序中添加线程，且勾选线程可重入，则当子程序在主程序中被反复调用时，会创建多个同名线程，最终到达允许上限。此时需要将“线程可重入”选项取消勾选。

2.9.4 变量区

该页面可以创建程序变量，监控程序变量值。程序变量与系统变量不同，系统变量是作用于整个工程，而程序变量只存在于本程序内。

添加变量

点击“添加变量”按钮，可弹出对话框，如图，输入变量名、选择类型、给定初始值即可创建。

程序变量有以下类型：

`boolean`：布尔型，只能为 `false` 或 `true`

`number`：数字类型

`string`：字符串类型

`num_list`：数组类型

`pose`：表示机器人笛卡尔位置的数据类型

`joints`：表示机器人关节位置的数据类型

`joints_list`：表示机器人关节位置列表类型

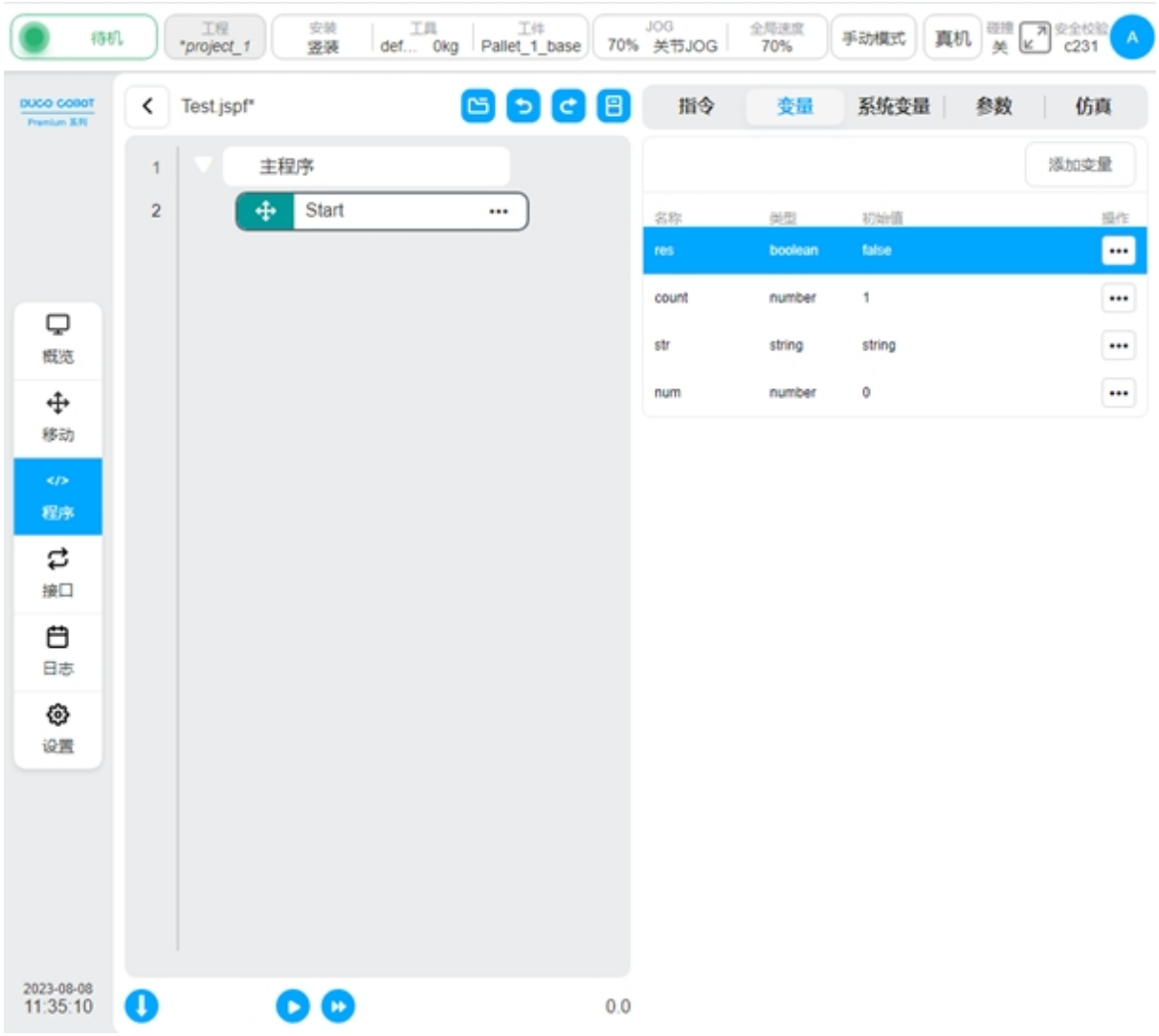
`pose_list`：表示机器人笛卡尔位置列表数据

`pose_speed`：表示机器人末端速度的数据类型

`pose_acc`：表示机器人末端加速度的数据类型

`joint_speed`：表示机器人关节角速度的数据类型

`pose_acc`：表示机器人关节角加速度的数据类型

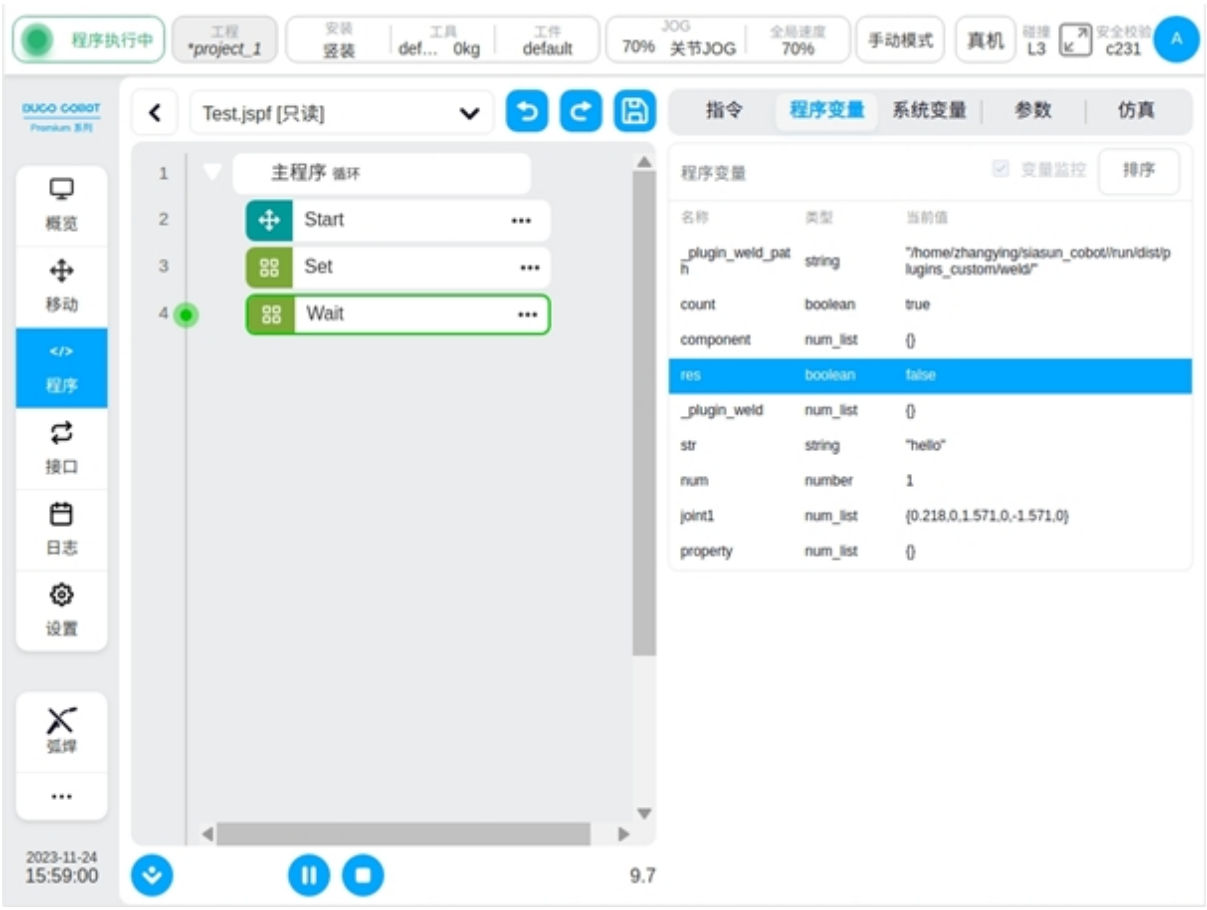


备注： 当存在子程序调用时，主程序及各个子程序之间定义的程序变量无作用域隔离，即主程序、子程序若同时定义了一个名称为 `var` 的变量，该变量实际的类型和值由程序运行过程中最后一次修改决定。

变量监控

当勾选“变量监控”时，程序运行过程中，该页面可以显示程序变量的当前值。点击“排序”按钮还可以对程序变量名称进行按字母排序。

备注： `joints` 和 `pose` 类型的变量在监控期间类型均显示为 `num_list`，并且显示的数据单位为 `m/rad`.

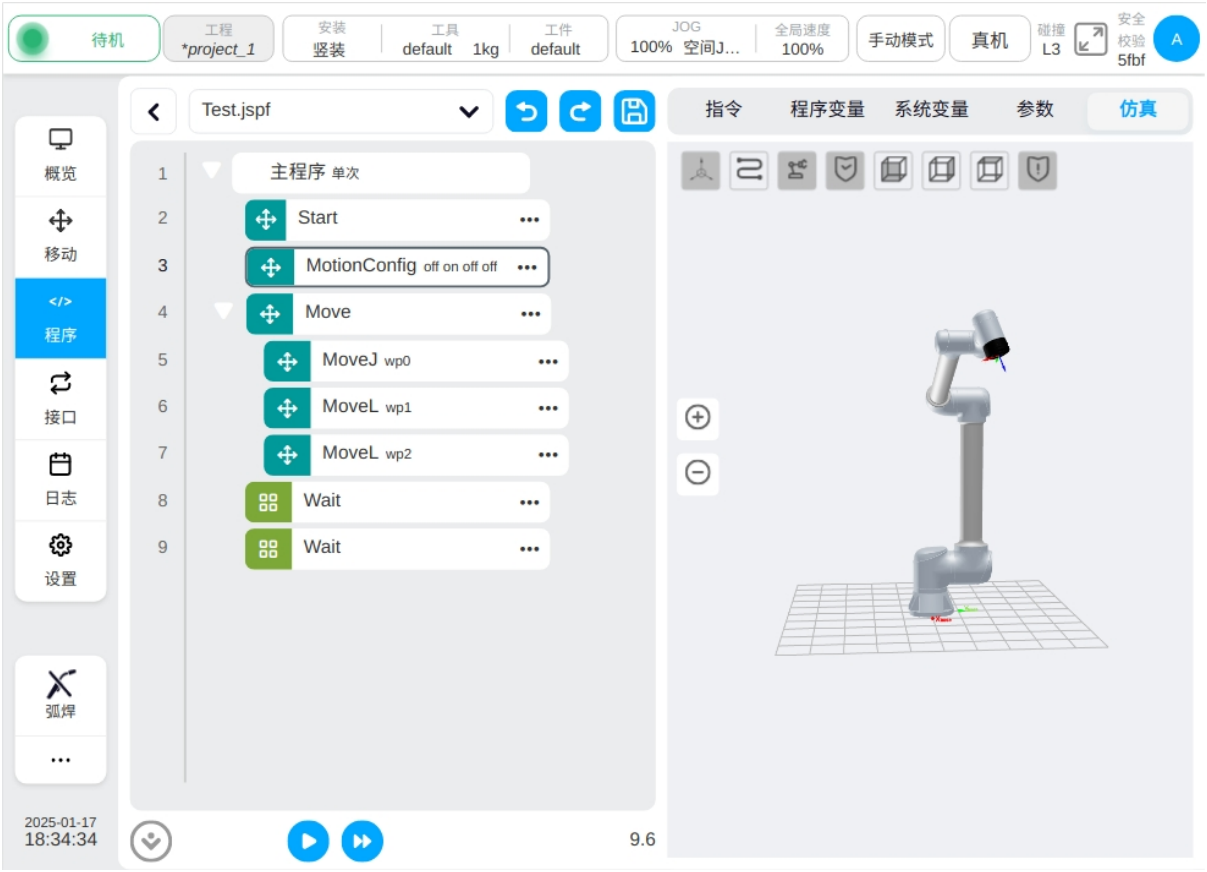


当不勾选“变量监控”时，由于节省了程序变量同步并显示到界面这一过程，将大大提高程序脚本的执行效率。

备注： 由于程序变量不再同步到界面，程序“打断点”、“从此行开始运行“等调试相关功能将全部失效，程序变量页面也不会显示程序变量相关内容。

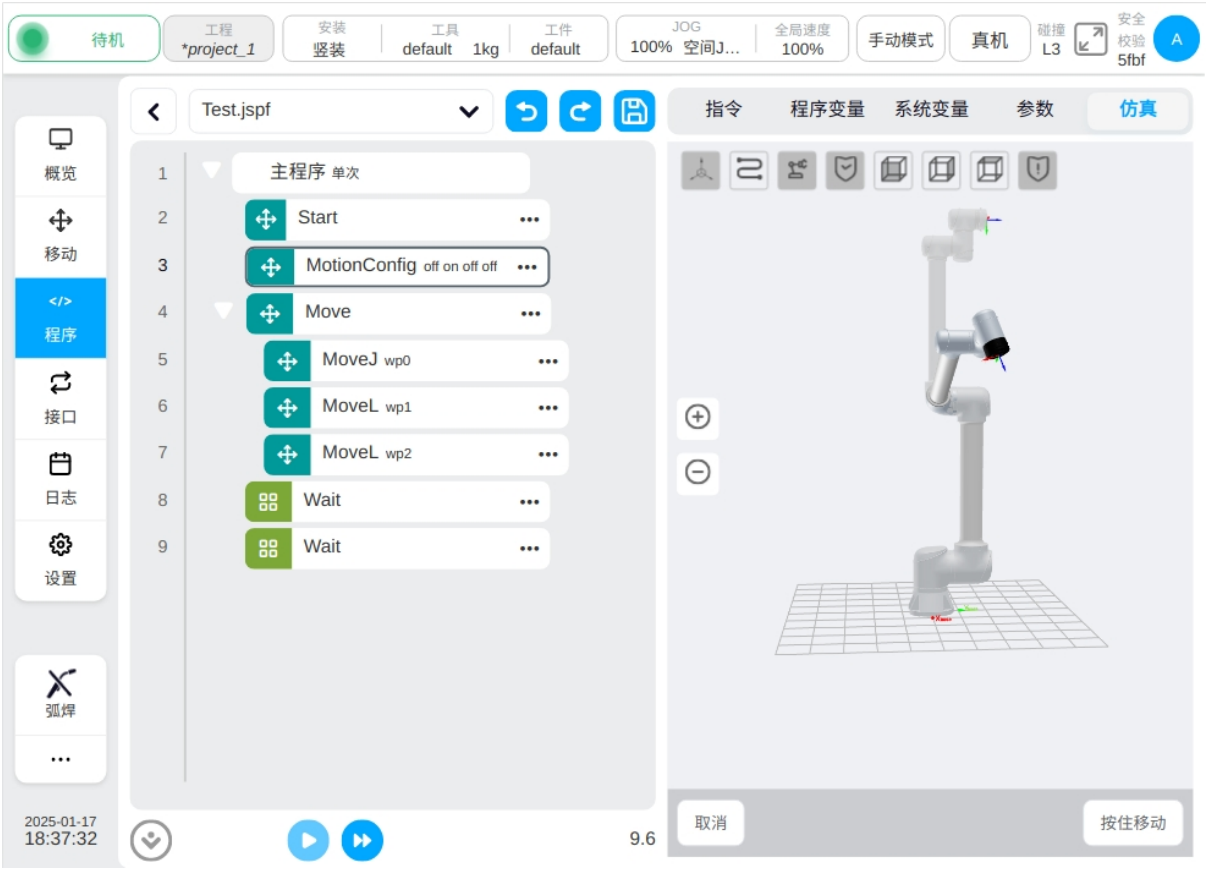
2.9.5 运行

点击“仿真”标签页可以进入 3D 模型页面，该页面显示机器人的 3D 模型。程序树左侧下方的图标控制程序的启停，点击  图标按钮运行程序，点击  图标按钮单步运行程序。在程序运行过程中，可以随时暂停或停止程序。点击  图标按钮设置程序运行过程中光标自动跟随指令块，点击  图标后退出自动跟随，且光标自动跟随功能是默认打开的。图标灰色代表跟随，蓝色代表取消跟随。

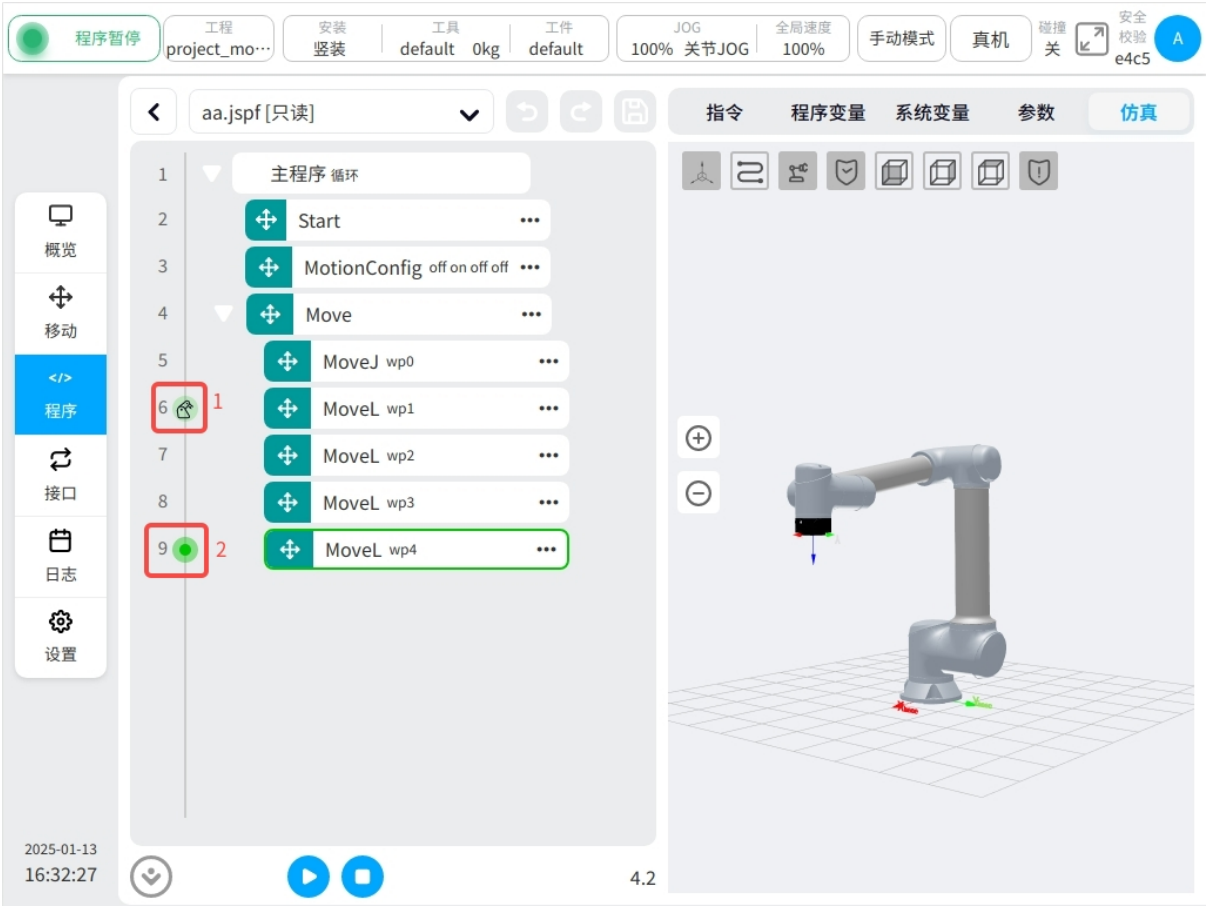


运行程序

运行程序时，会检查程序起始点位和当前机器人所处点位是否一致，若不一致，先要移动到起始点，如图，3D 模型上会显示处于起始点位的机器人（灰色显示），点击“按住移动”按钮，机器人会移动到起始点位。



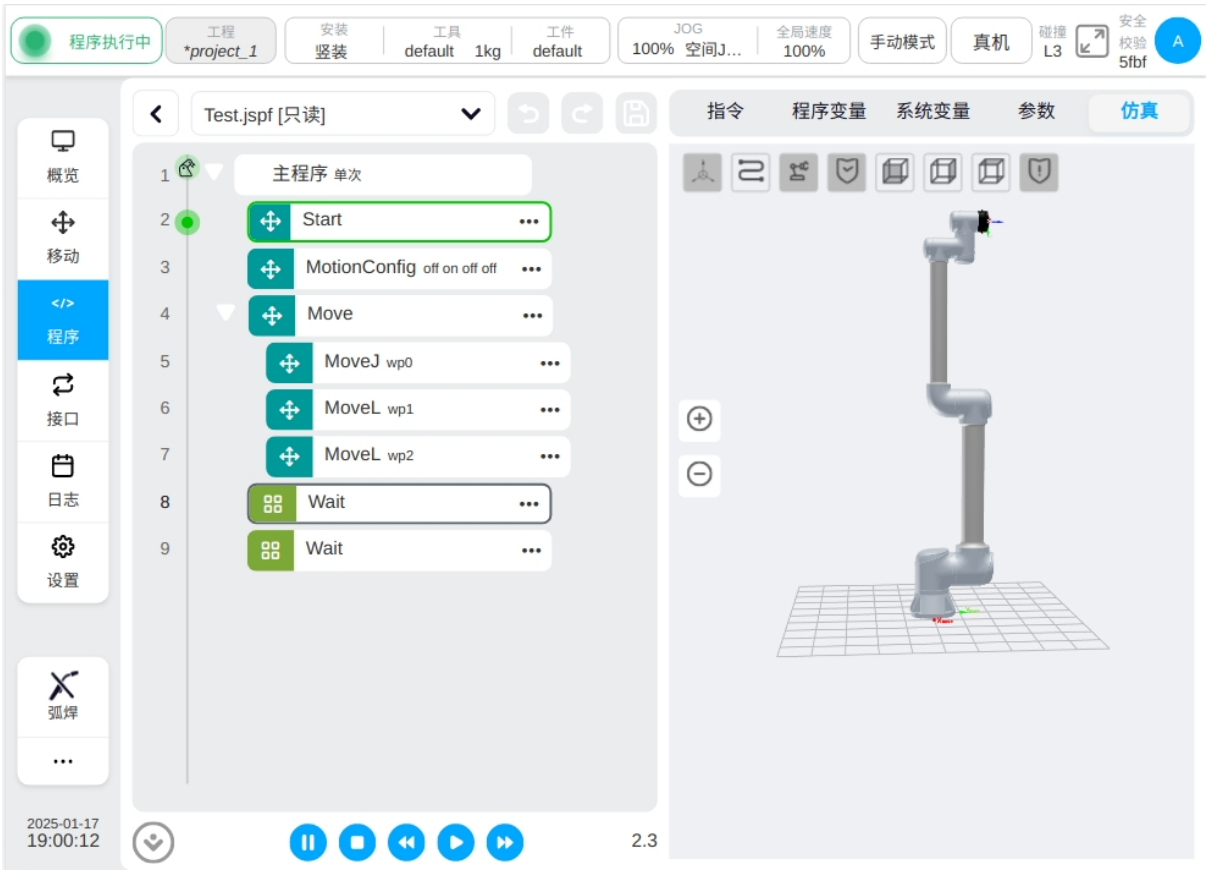
移动到起始点位后，再次点击  图标按钮，程序运行。如下图，程序树左侧下方可以显示程序的运行时间，3D 模型上可以观察机器人的实时姿态，左侧的程序树可以同时显示当前机器人正在执行的运动指令功能块以及正在执行的脚本功能块。即图中 1 处是运动指针所在位置，指示的是机器人当前正在执行的运动指令，图中 2 处是程序指针所在位置，通常运动指针比程序指针落后一个或几个指令，因为控制器执行和计算机器人路径比执行和计算机器人移动指令更快。点击下方的控制按钮可以进行程序的暂停、恢复、停止操作。



单步运行程序

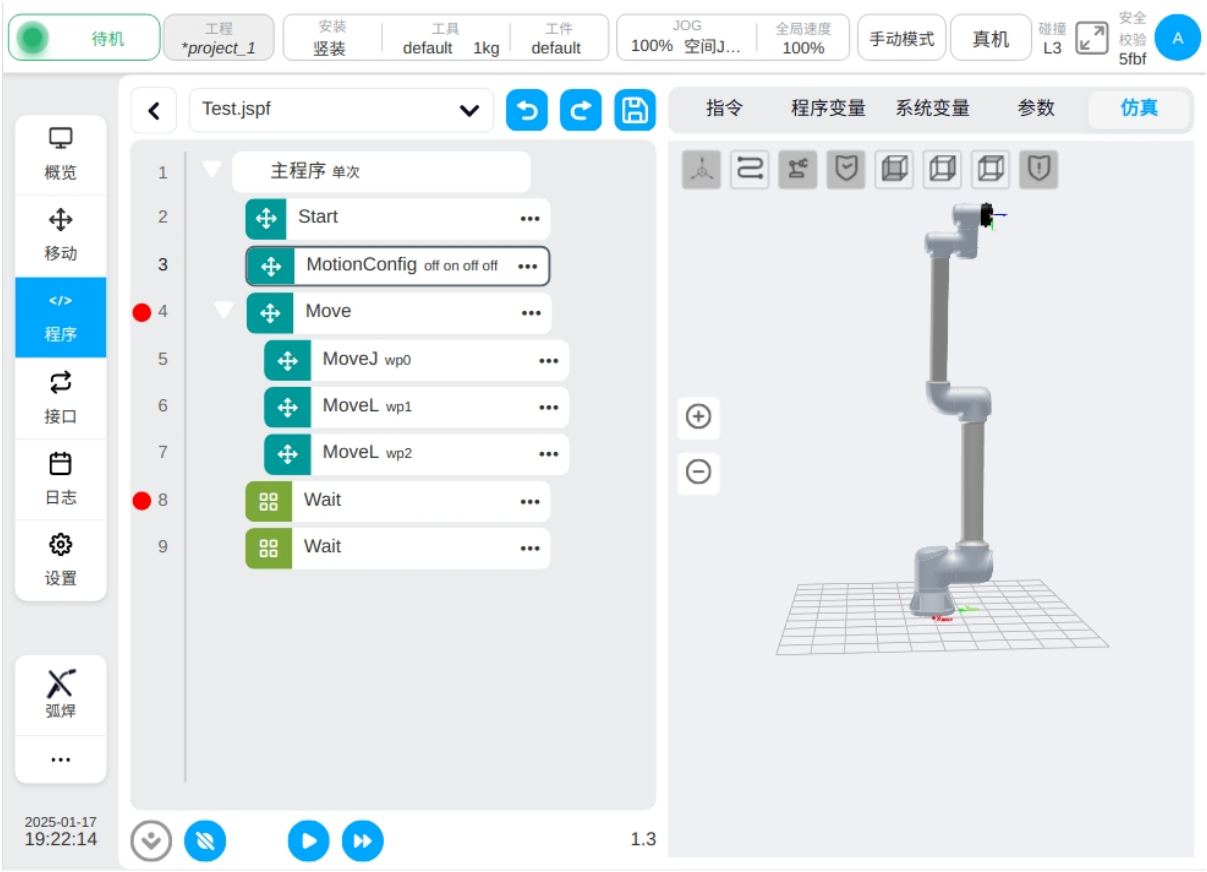
点击  图标按钮单步运行程序。单步时，首先会检查程序的起始点，移动机器人到程序的起始点位置后。每点击一次  图标，程序会向下执行一个功能块。点击  图标按钮，程序将退出单步模式，连续运行。在单步过程中可以暂停、恢复程序。点击  图标按钮，将执行后退操作，此时机器人将后退移动到上一个点位，后退仅针对运动类功能块。

点击  停止按钮，程序停止运行，退出单步状态。



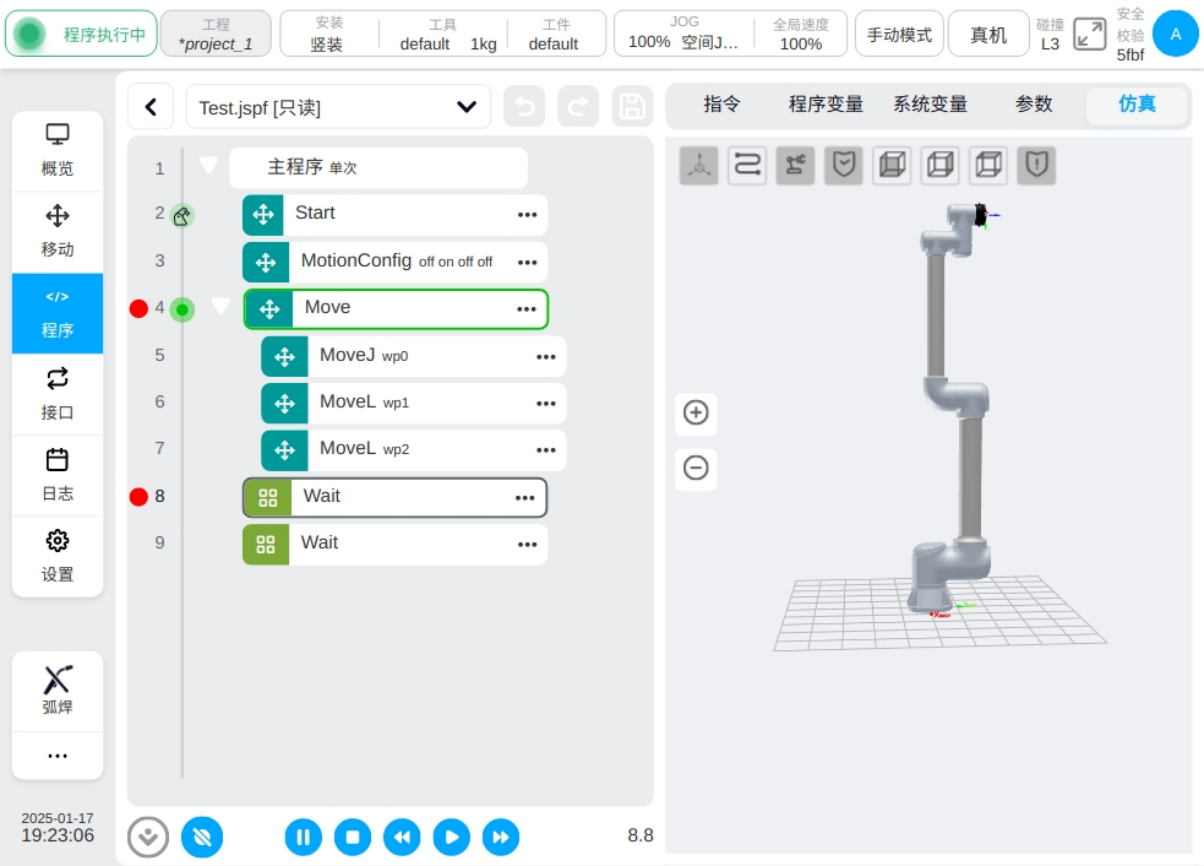
断点

支持断点调试功能，程序在执行到断点位置处，会停止并进入单步模式。可以在程序树和脚本编辑区打断点，点击程序树行号左侧区域、脚本行号左侧区域即可打上断点，再次点击断点则可以取消断点。点击下方的  图标按钮可以取消系统设置的所有断点。



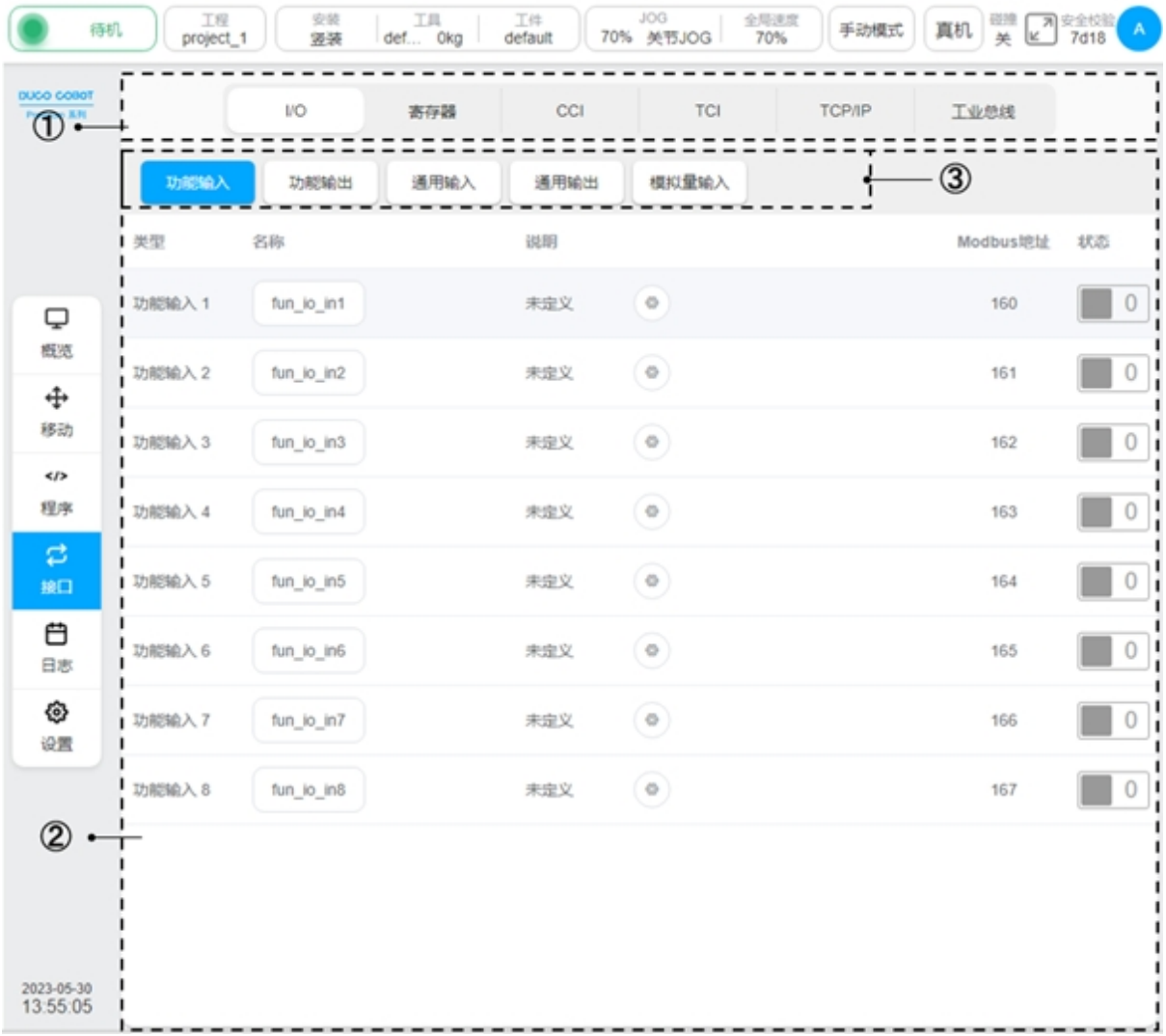
运行程序，当程序运行到断点处，程序将停止并进入单步模式，如图所示。点击  图标按钮，程序将单步执行，（运行到 `Script` 功能块时，将单行执行）；点击  图标按钮，程序将继续运行，直到碰到下一个断点或程序结束。

手动模式下，在程序运行过程中可以随时打断点或者取消断点。

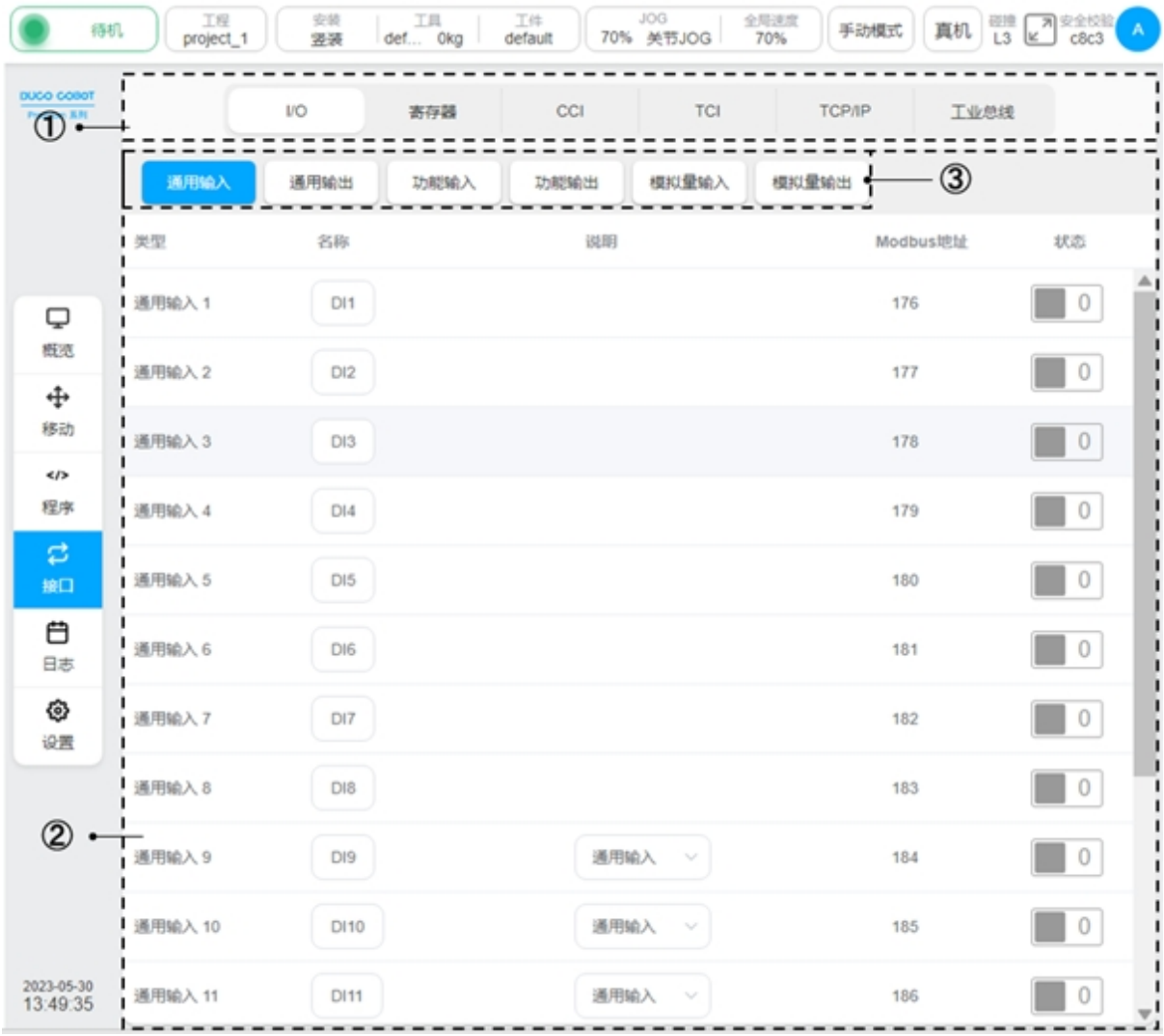


2.10 接口页面

单击导航栏中“接口”会默认进入接口页面下 I/O 子页面。接口页面下有 I/O、寄存器、CCI、TCI、TCP/IP、工业总线共 6 个子页面。根据控制柜柜体型号，分为两种页面，如图所示：


待机
工程 project_1
安装 竖装
工具 def... 0kg
工件 default
JOG 70% 关节JOG
全局速度 70%
手动模式
真机
碰撞 关
安全校验 7d18
A
I/O
寄存器
CCI
TCI
TCP/IP
工业总线
①
③
功能输入
功能输出
通用输入
通用输出
模拟量输入
类型
名称
说明
Modbus地址
状态
功能输入 1 fun_io_in1 未定义 160 0
功能输入 2 fun_io_in2 未定义 161 0
功能输入 3 fun_io_in3 未定义 162 0
功能输入 4 fun_io_in4 未定义 163 0
功能输入 5 fun_io_in5 未定义 164 0
功能输入 6 fun_io_in6 未定义 165 0
功能输入 7 fun_io_in7 未定义 166 0
功能输入 8 fun_io_in8 未定义 167 0
概览
移动
程序
接口
日志
设置
②
2023-05-30
13:55:05

适配 DC30D 控制柜



适配 DC00 / DC15S-J9 / DC30D-J9 控制柜

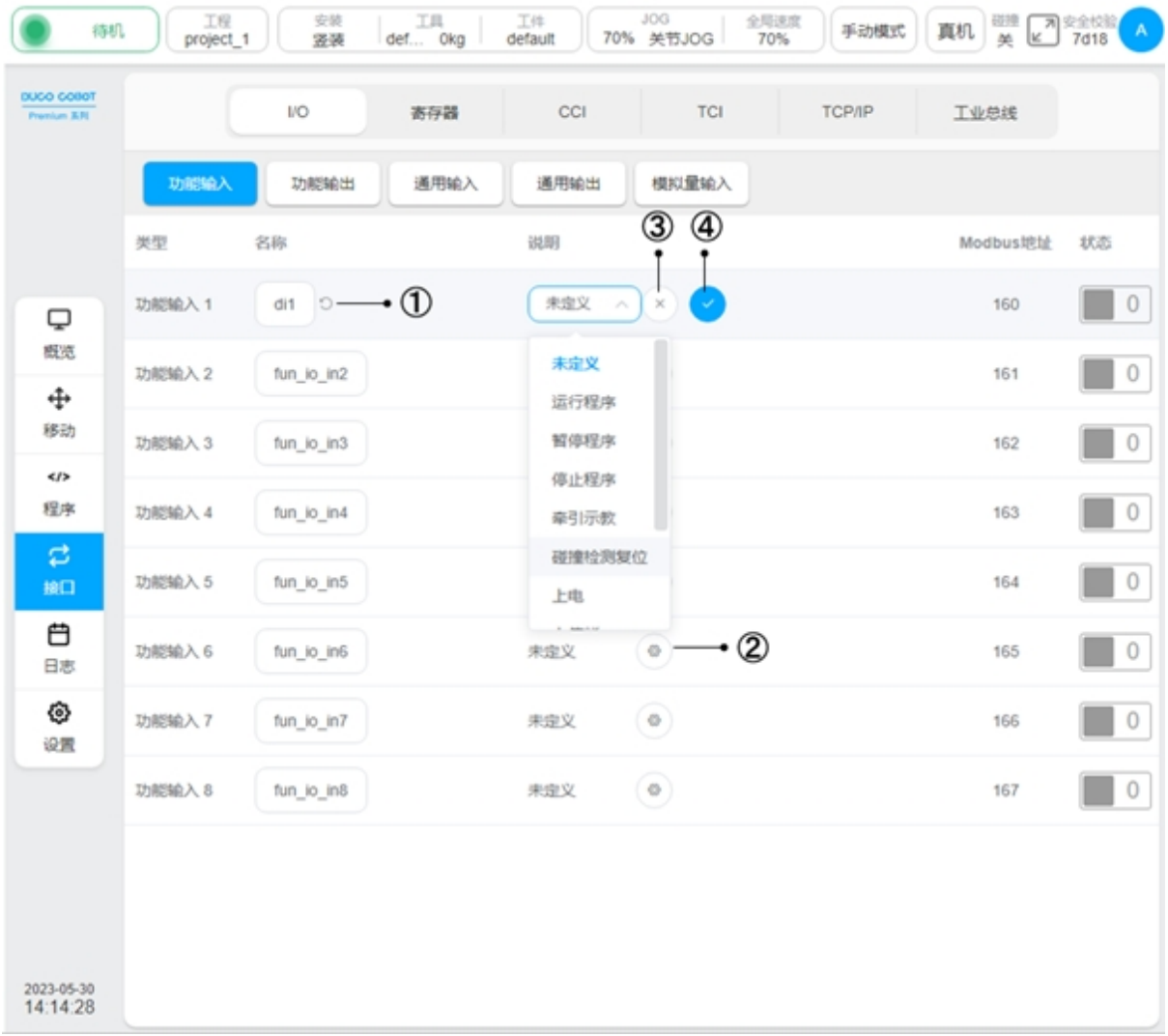
页面上方 ① 是接口页面下子页面选项卡，下方 ② 是对应子页面的显示区。用户选择不同的子页面选项时区域 ② 对应切换成该选项的显示内容。

2.10.1 IO

图中区域 ② 显示的是 I/O 子页面不同选项卡对应内容，I/O 子页面显示区上方 ③ 是 I/O 子页面下针对不同输入输出做的标签。

对于 DC30D 控制柜，I/O 子页面下分别对应有功能输入、功能输出、通用输入、通用输出和模拟输入共 5 种类型，单击不同标签栏切换显示 5 种类型的输入或输出状态信息。

功能输入界面可以监控控制柜中能设置某种特定功能的 8 路数字输入状态和基本信息。单击功能输入名称，会弹出普通虚拟键盘，可对功能输入默认名称进行修改。当修改功能输入默认名称后，名称显示框右侧会显示如图中图标 ①。单击该图标可使功能输入名称恢复成对应的默认名称。



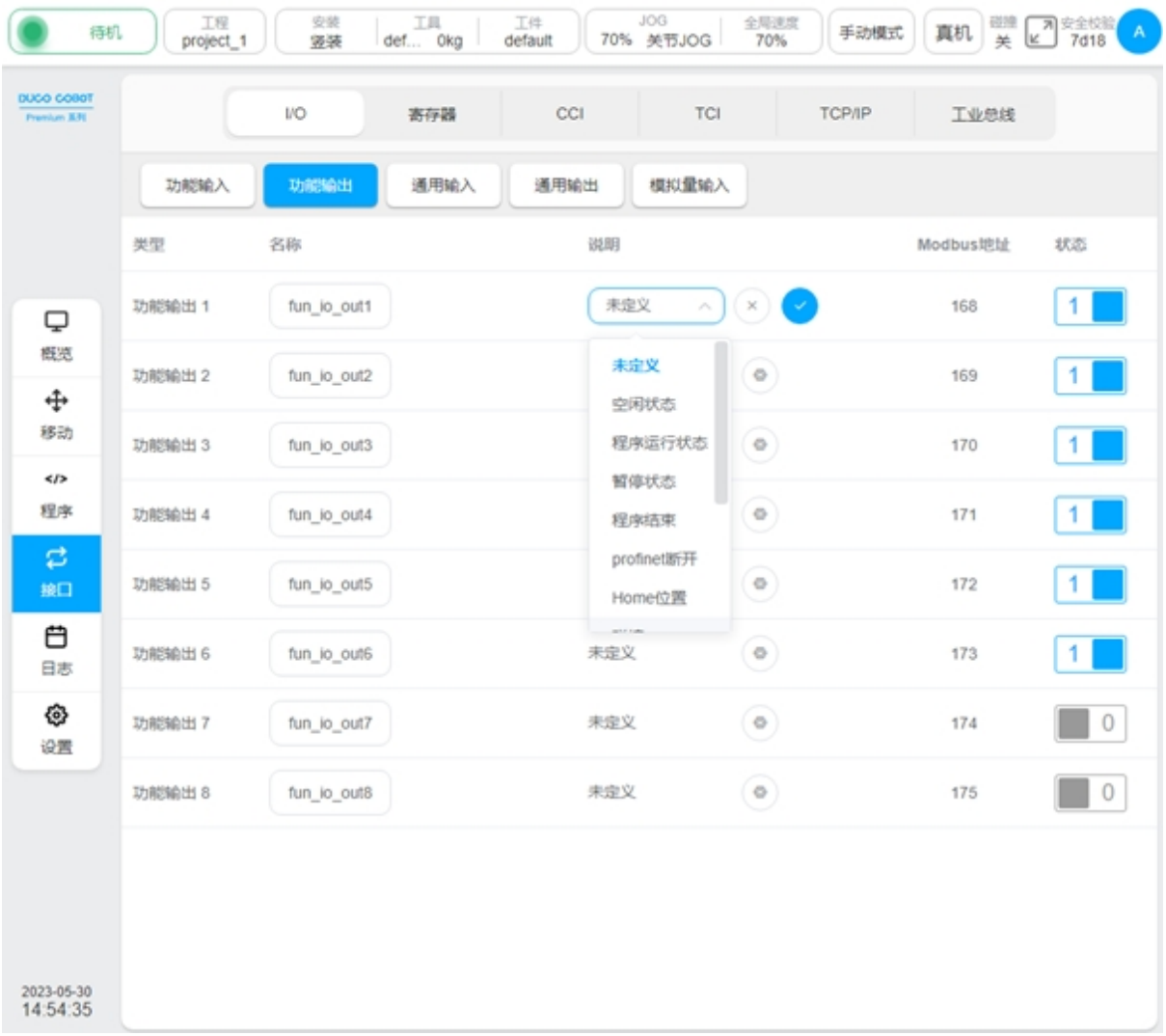
单击图中对应功能输入的图标 ②，会显示功能设置选择器、图标 ③ 和图标 ④，可选择功能输入的功能设置。在显示功能选择器下拉框选择该功能输入所想设置的功能项，单击图标 ④ 确定所选功能项。也可单击图标 ③ 取消所选功能项。当该功能输入信号被触发时，则启动所设置的功能。目前功能输入支持功能有：运行程序、暂停程序、停止程序、上电、上使能、下电、下使能、系统关机、记录点位、按住归位、中止运动任务、低速模式、牵引示教、碰撞检测复位。

备注： 运行程序、暂停程序、停止程序、中止运动任务为上升沿触发信号。

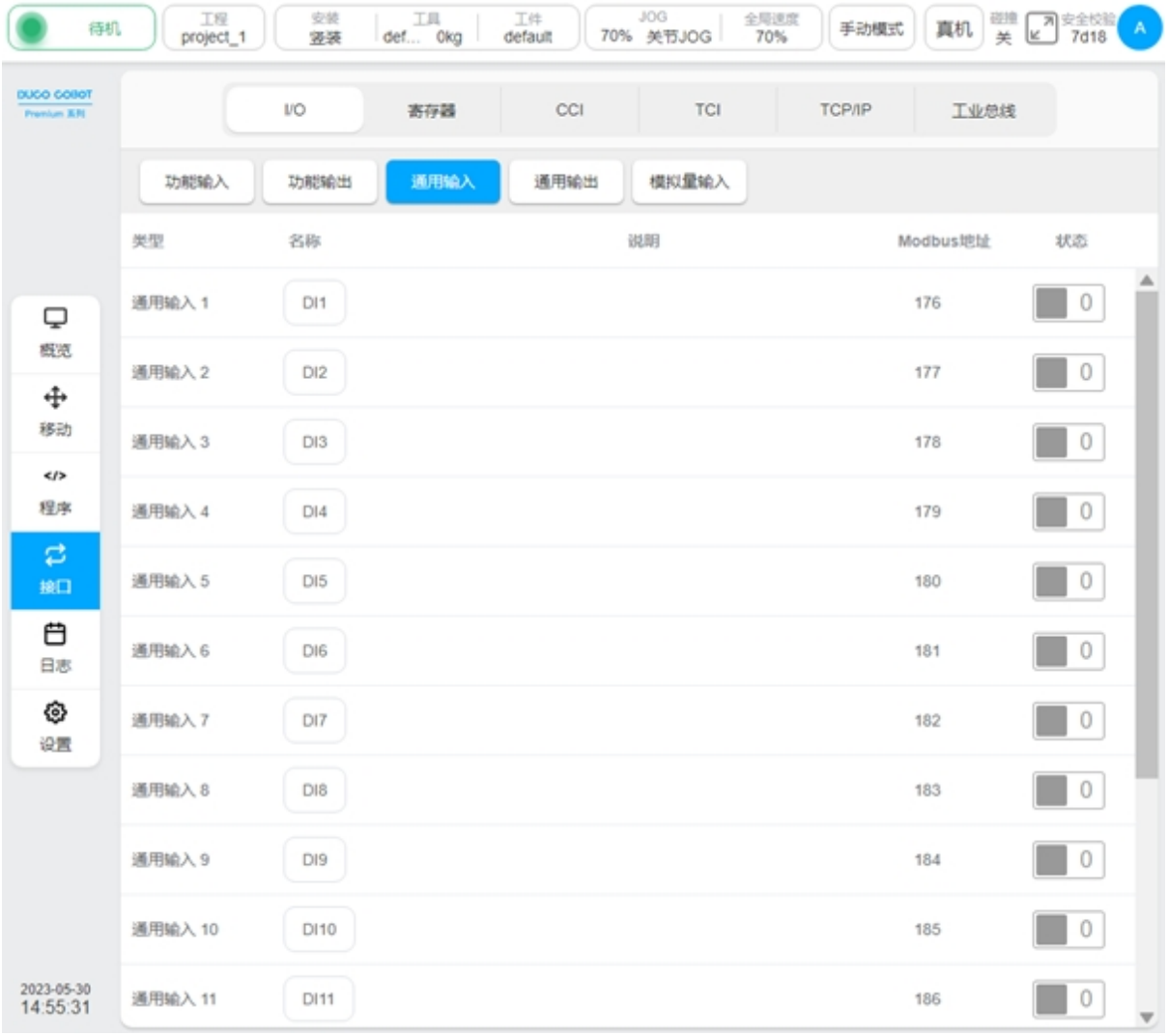
备注： 中止运动任务仅对当前机器人正在执行且已经预读取的轨迹进行中止，不会影响后续其他脚本的执行。

功能输出界面可监控控制柜中能设置某种特定功能的 8 路数字输出状态和基本信息，如图所示。与功能输入界面类似，可对功能输出进行对应的名称修改和选择功能输出的功能设置。在功能设置选择器下拉框选择该功能输出所绑定的系统预定义的状态量，单击确定图标后，该功能输出将反映所设置的系统状态量的状态。目前功能输出支持绑定的状态有：空闲状态、程序运行状态、暂停状态、程序结束、未上电、未使能、机器人供电、机械臂运动中、自动模式、Home 位置、profinet 断开、ethernet 断开、运动任务被中止、低速模式、牵引中、碰撞。

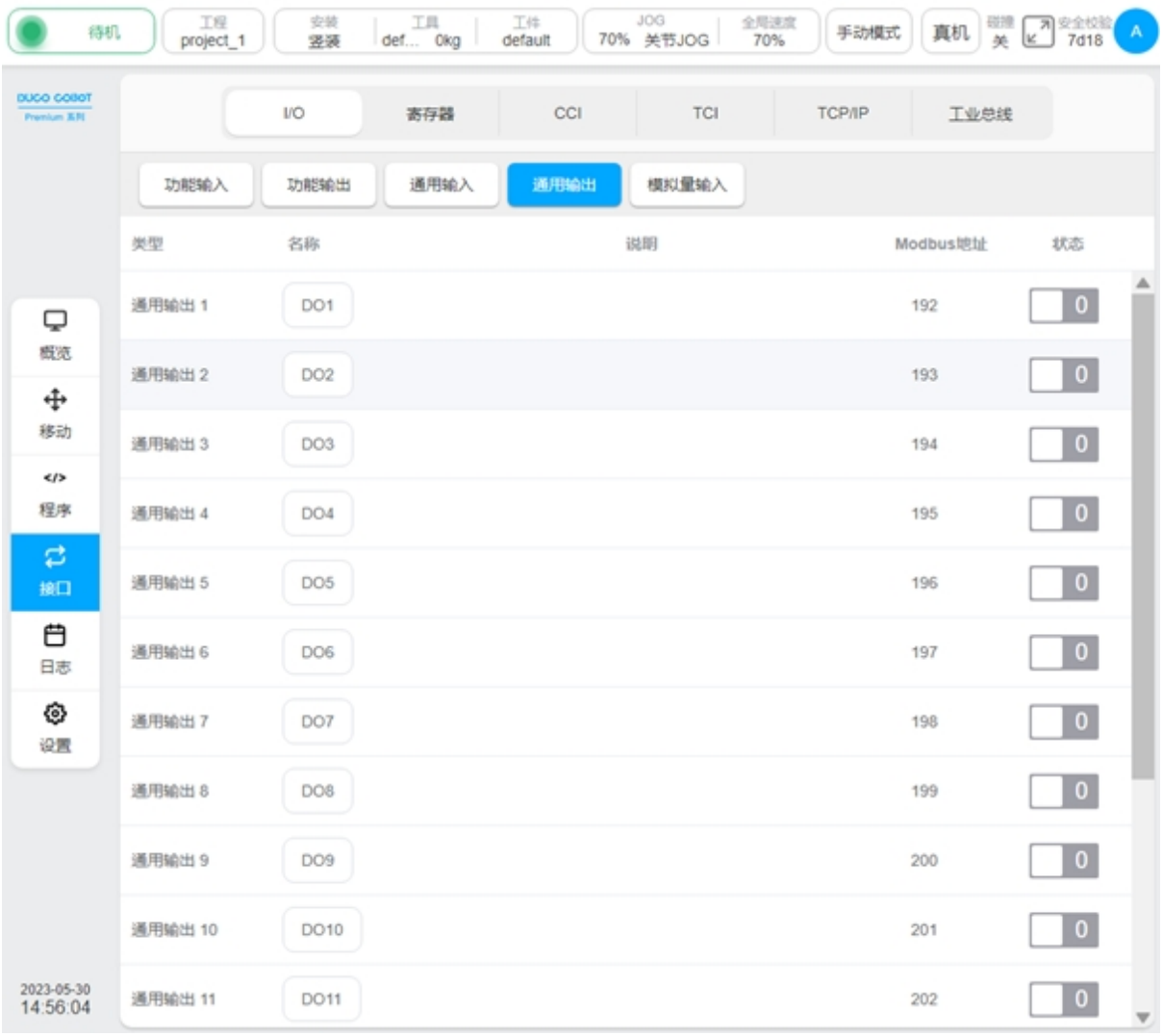
备注： 运动任务被中止信号当且仅当机器人在运动过程中存在通过功能 IO 输入、功能寄存器输入或其他接口形式使机器人中止运动任务时输出高电平。当功能 IO 输出被配置为运动任务被中止时，当且仅当同时配置了功能 IO 输入或功能寄存器输入为中止运动任务，且通过功能 IO 输入或功能寄存器输入触发后，对应的功能 IO 输入或功能寄存器输入被复位为低电平时，功能 IO 输出同时复位为低电平。



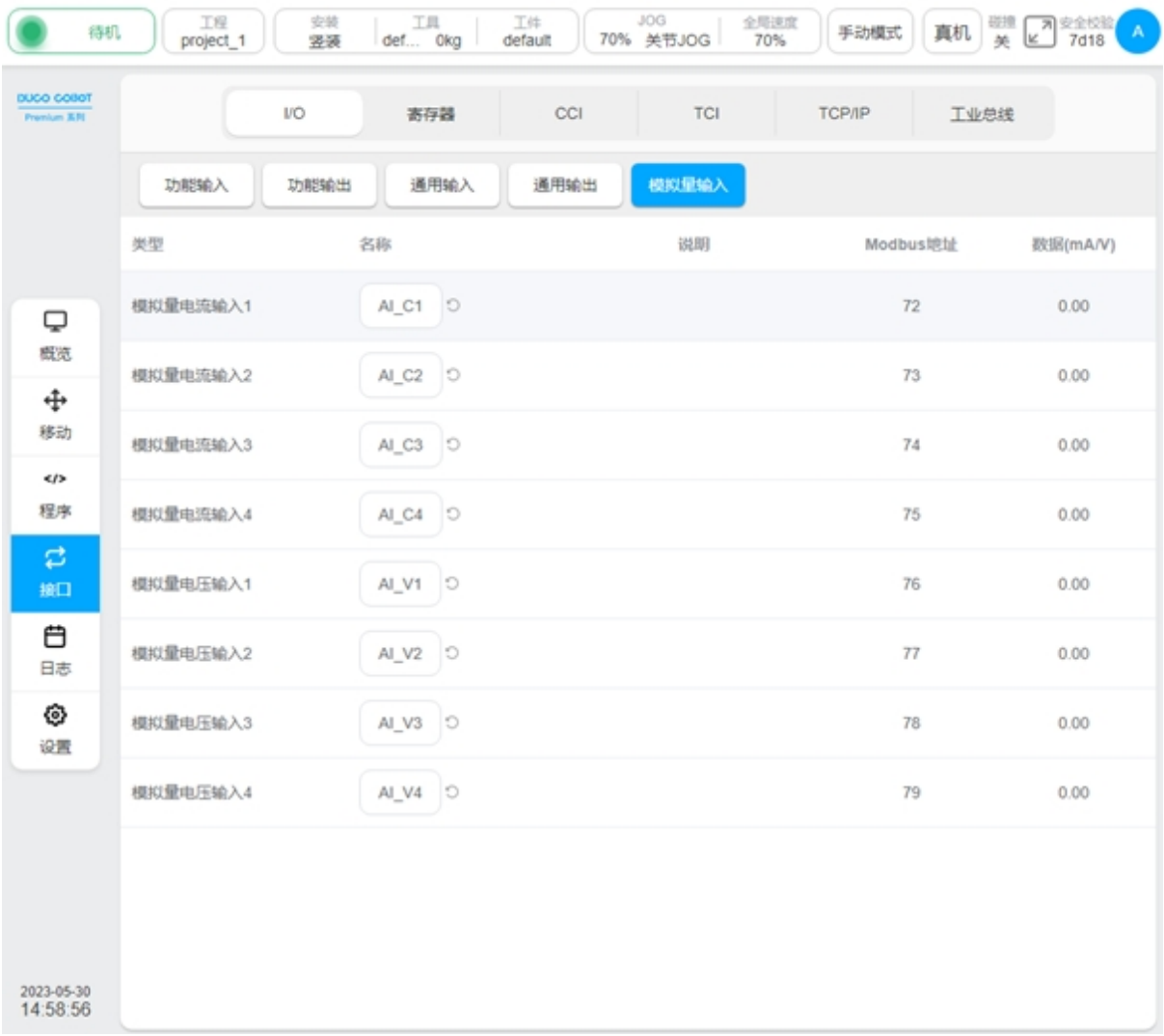
通用输入界面可监控控制柜中普通 16 路数字输入状态和基本信息，如图所示。不同于功能输入和功能输出界面的是只可对输入信号进行对应的名称修改和修改后的恢复，不能设置特定功能。



通用输出界面可监控控制柜中普通 16 路数字输出状态和基本信息，如图所示。该界面也只可对输出信号进行对应的名称修改和修改后的恢复，不能设置特定功能。

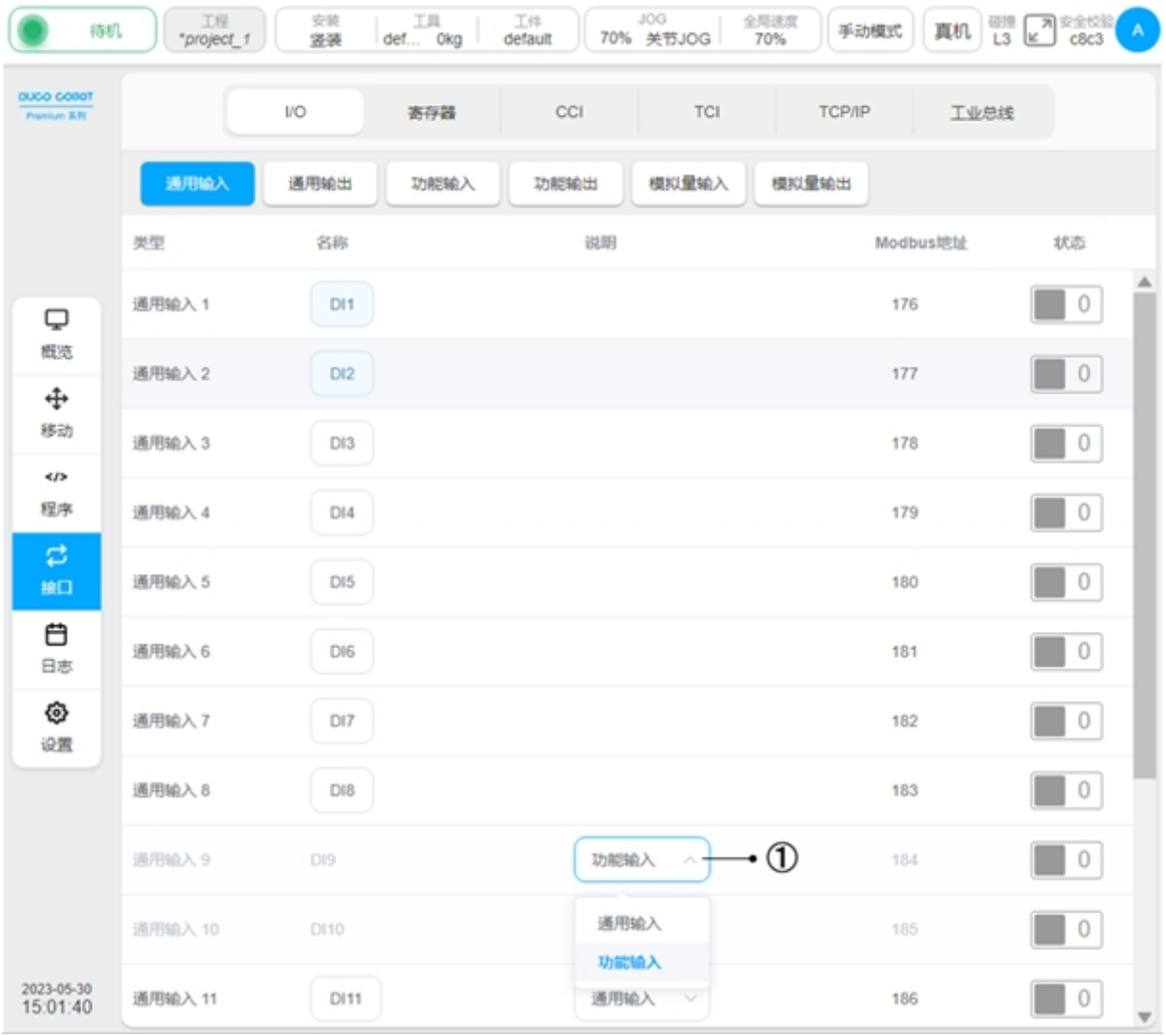


模拟量界面可监控 4 路模拟量电流输入信号和 4 路模拟量电压输入信号，如图所示。同样地，单击模拟量输入信号名称显示框，可对该信号名称进行修改。操作与上述描述的功能输入输出和通用输入输出类似，此处不累述。

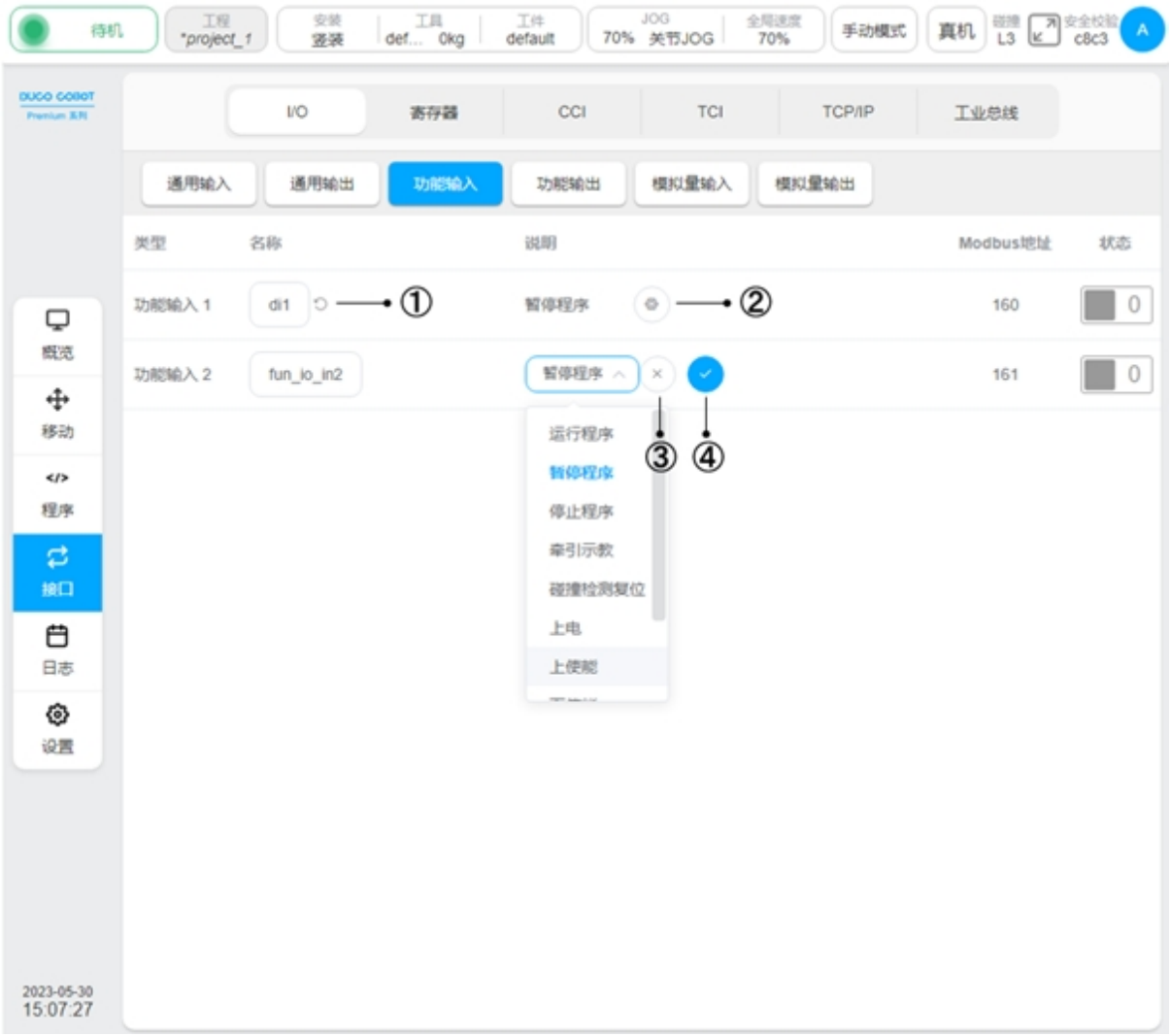


对于 DC00 / DC15S-J9 / DC30D-J9 控制柜，I/O 子页面下分别对应有通用输入、通用输出、功能输入、功能输出、模拟量输入和模拟量输出共 6 种类型，单击不同标签栏切换显示 6 种类型的输入或输出状态信息。

通用输入界面可监控控制柜中普通 16 路数字输入状态和基本信息，如图所示。其中，DI9—DI16 这 8 路数字输入类型可以在页面的说明列 ① 处进行配置，可配置为功能输入或者通用输入。



若选择每路数字输入为功能输入，则在 I/O 通用输入页面该路数字输入显示为禁用状态，且在功能输入子标签页面将显示被配置为功能输入的各路数字状态，如配置 DI9 和 DI10 为功能输入，则 I/O 子页面下功能输入标签页如图所示。

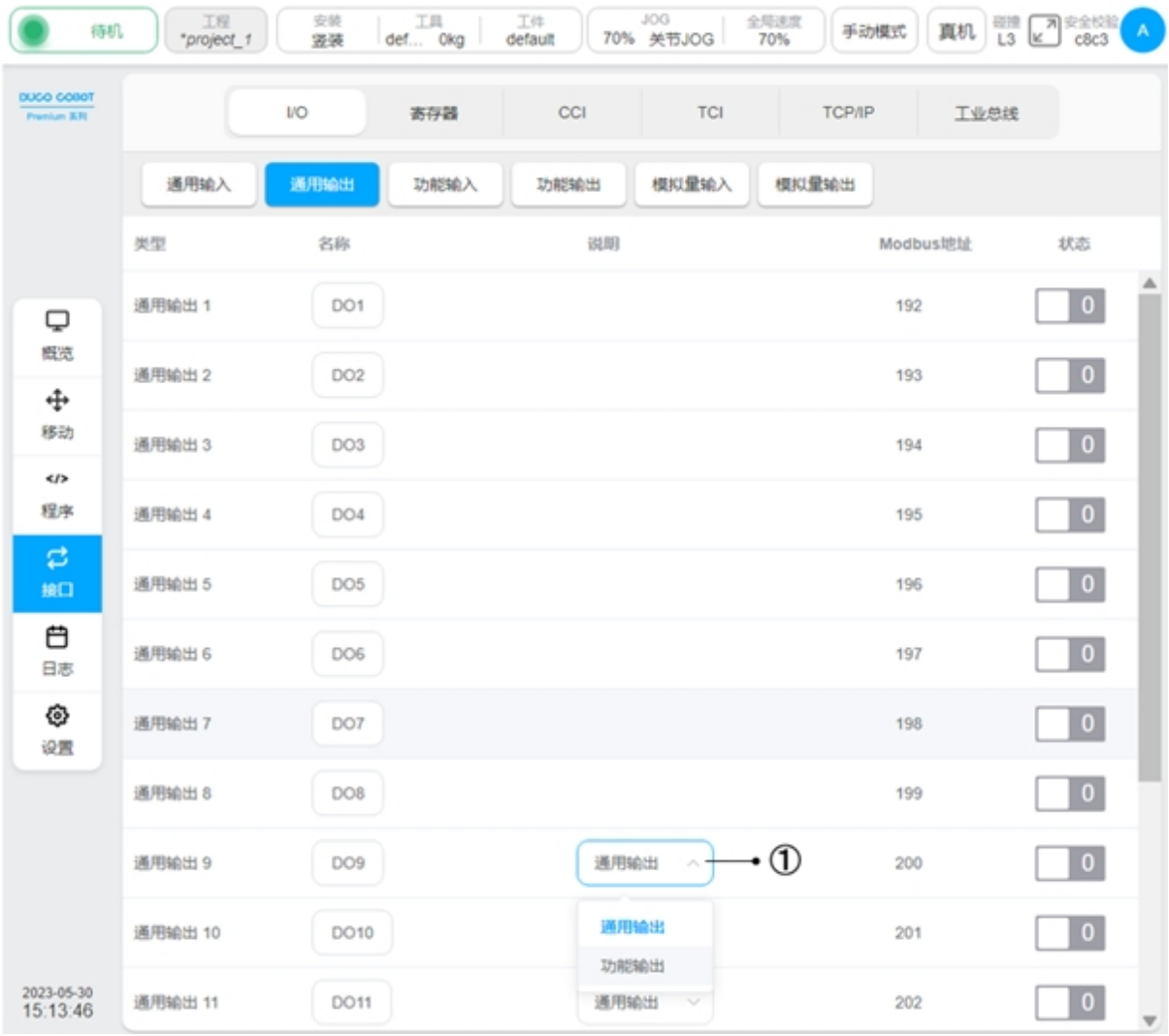


同样地，单击功能输入名称，会弹出普通虚拟键盘，可对功能输入默认名称进行修改。当修改功能输入默认名称后，名称显示框右侧会显示如上图中图标 ①。单击该图标可使功能输入名称恢复成对应的默认名称。

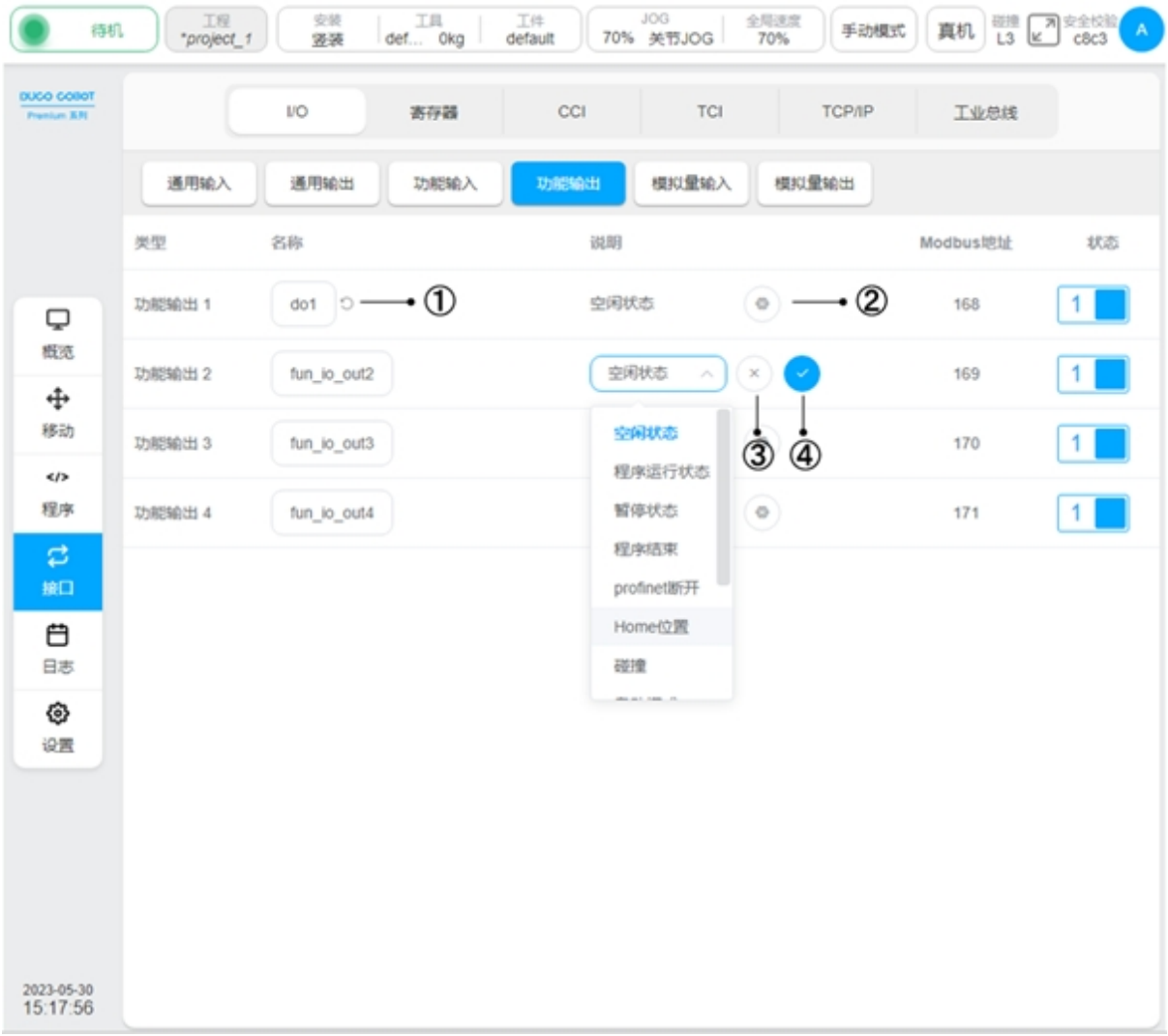
单击图中对应功能输入的图标 ②，会显示功能设置选择器、图标 ③ 和图标 ④，可选择功能输入的功能设置。在显示功能选择器下拉框选择该功能输入所想设置的功能项，单击图标 ④ 确定所选功能项。也可单击图标 ③ 取消所选功能项。当该功能输入信号被触发时，则启动所设置的功能。目前功能输入支持功能有：运行程序、暂停程序、停止程序、上电、上使能、下电、下使能、系统关机、记录点位、按住归位、中止运动任务、低速模式、牵引示教、碰撞检测复位。

备注： 运行程序、暂停程序、停止程序为上升沿触发信号。

通用输出界面可监控控制柜中普通 16 路数字输出状态和基本信息，如图所示。其中，DO9—DO16 这 8 路数字输入类型可以在页面的说明列 ① 处进行配置，可配置为功能输入或者通用输入。



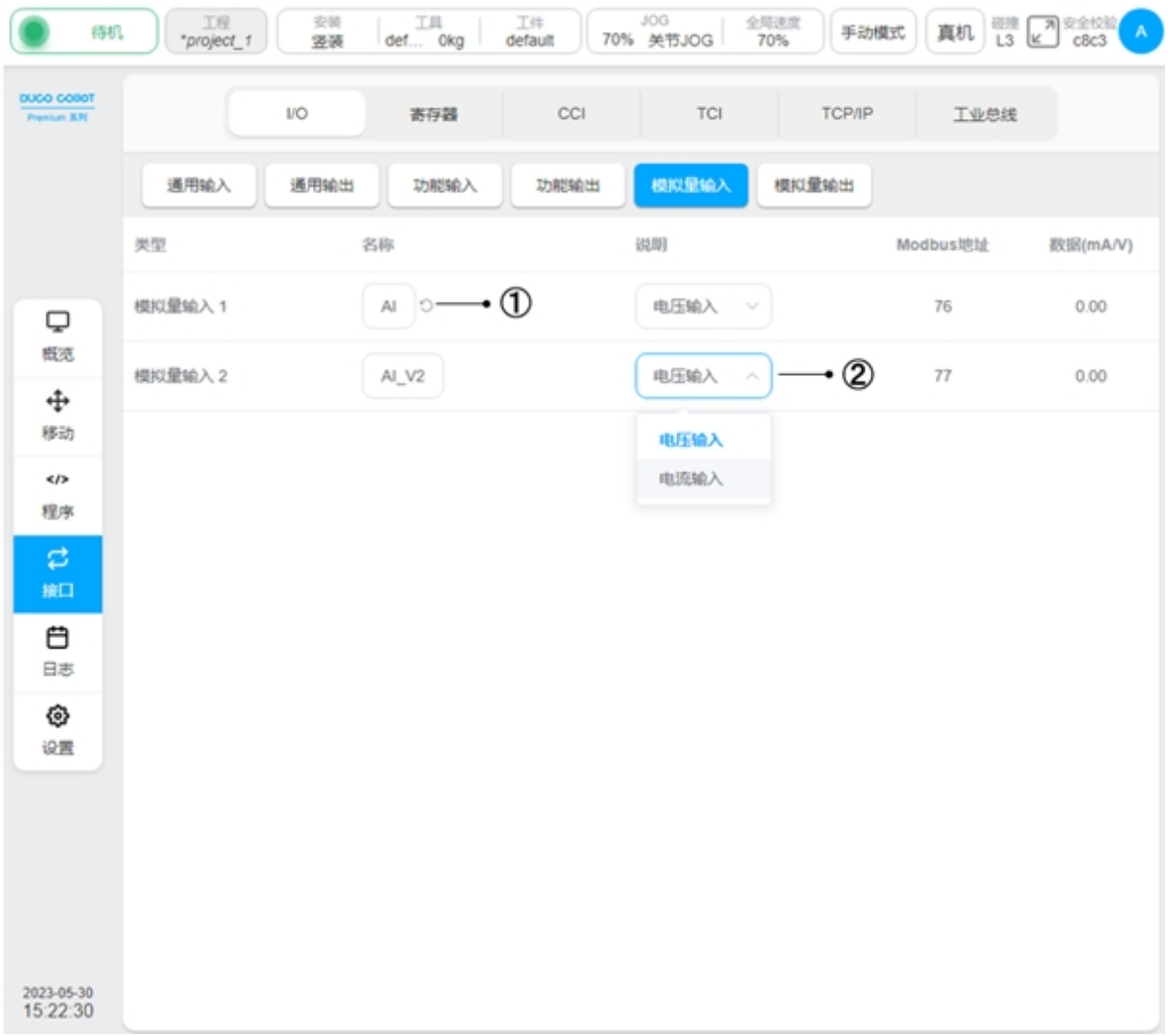
同样地，若选择每路通用数字输出为功能输出，则在 I/O 通用输出页面该路数字输出显示为禁用状态，且在功能输出子标签页面将显示被配置为功能输出的各路数字状态，如配置 DO9、DO10、DO11、DO12 为功能输出，则 I/O 子页面下功能输出标签页如图所示。



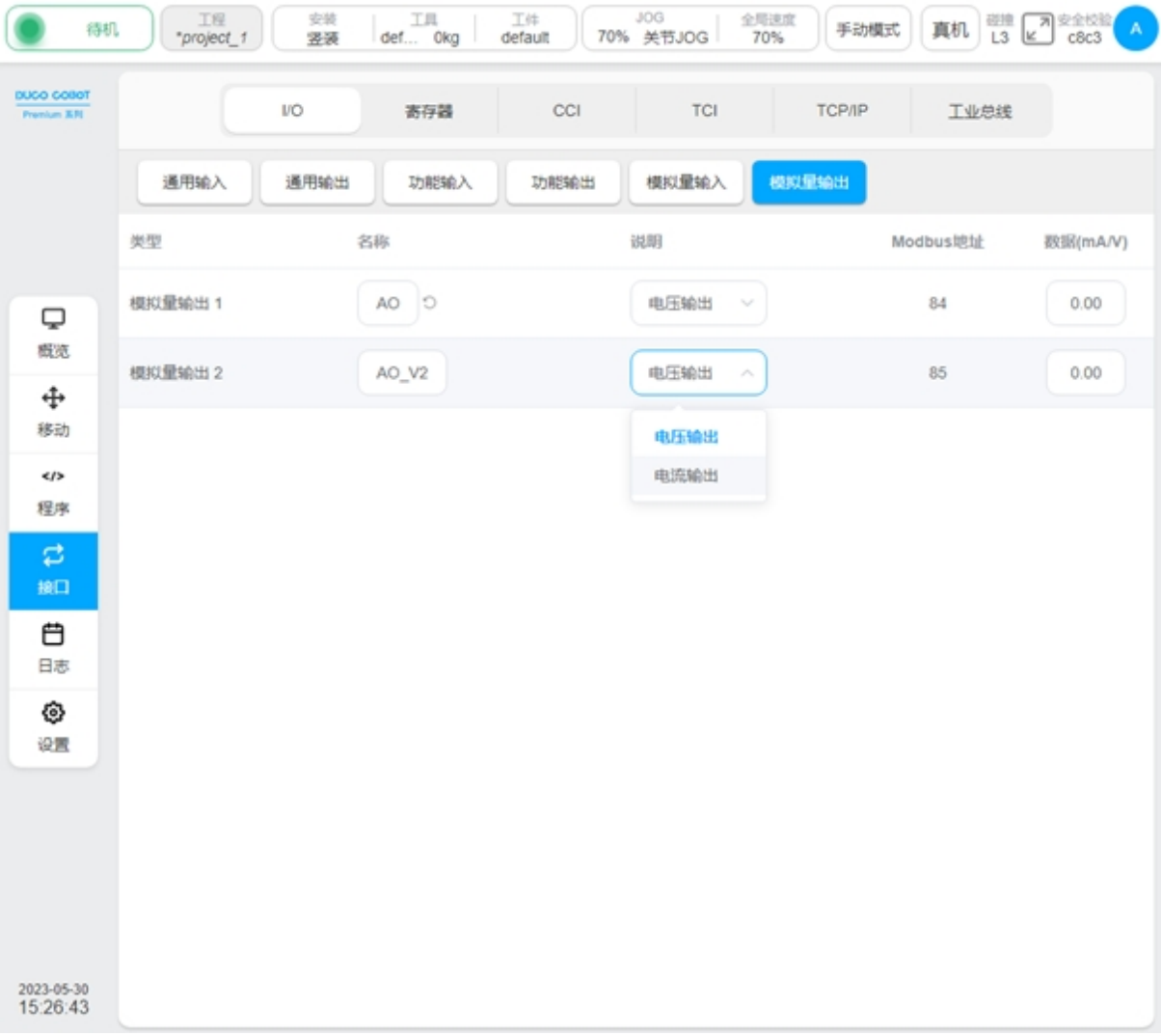
同功能输入标签页一样，单击功能输出名称，会弹出普通虚拟键盘，可对功能输出默认名称进行修改。当修改功能输出默认名称后，名称显示框右侧会显示如上图中图标 ①。单击该图标可使功能输出名称恢复成对应的默认名称。

单击图中对应功能输出的图标 ②，会显示功能设置选择器、图标 ③ 和图标 ④，可选择功能输出的功能设置。在显示功能选择器下拉框选择该功能输出所想设置的功能项，单击图标 ④ 确定所选功能项。也可单击图标 ③ 取消所选功能项。当该功能输出信号被触发时，则启动所设置的功能。目前功能输出支持功能有：空闲状态、程序运行状态、暂停状态、程序结束、未上电、未使能、机器人供电、机械臂运动中、自动模式、Home 位置、profinet 断开、ethernet 断开、运动任务被中止、低速模式、牵引中、碰撞。

模拟量输入界面可监控 2 路模拟量输入信号，如图所示。同样地，单击模拟量输入信号名称显示框，可对该信号名称进行修改。当修改模拟量输入默认名称后，名称显示框右侧会显示如图中图标 ①。单击该图标可使模拟量输入名称恢复成对应的默认名称。且可对模拟量输入信号在说明列图标 ② 进行说明配置选择，即可选择是电压输入还是电流输入。



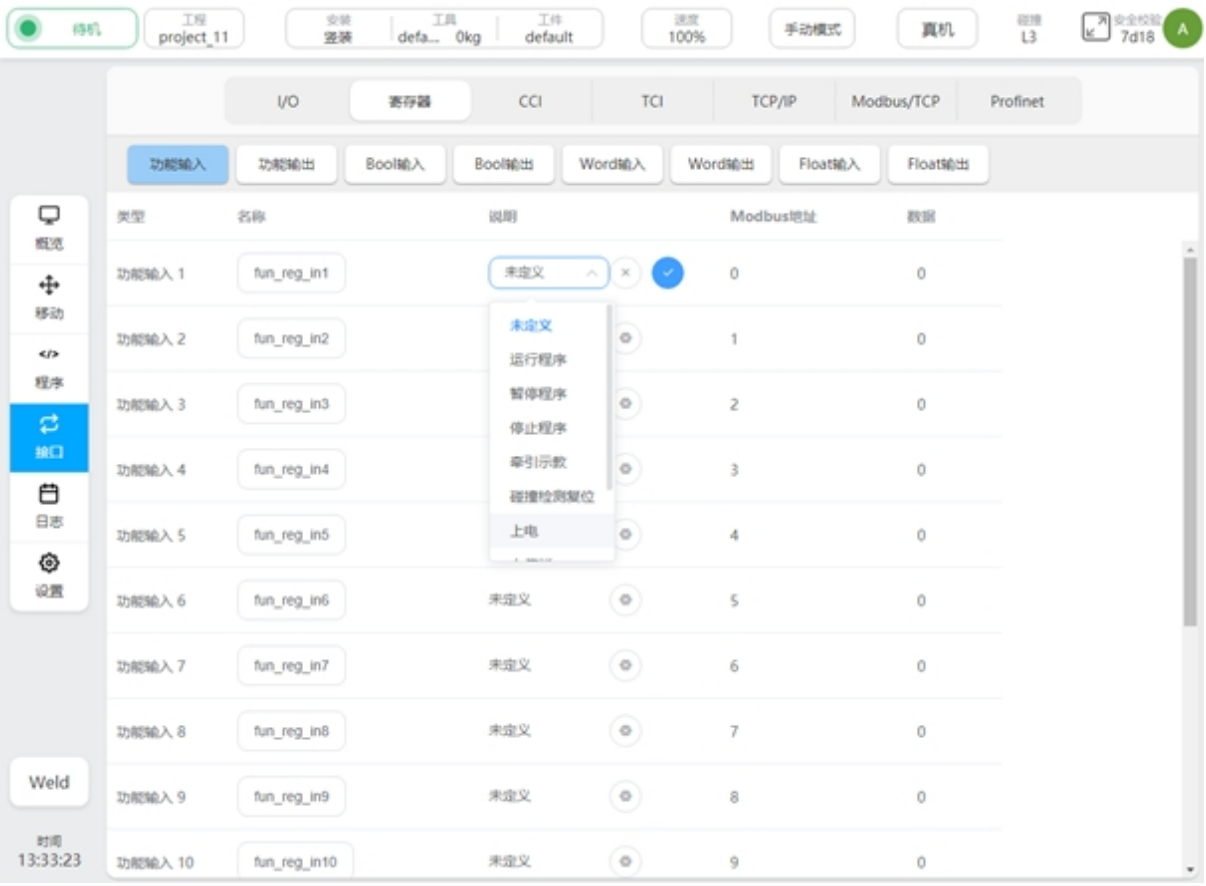
模拟量输出界面可监控 2 路模拟量输出信号，如图所示。同样地，单击模拟量输入信号名称显示框，可对该信号名称进行修改。操作与上述描述的模拟量输入类似，此处不累述。且可对模拟量输出信号在说明列进行说明配置选择，即可选择是电压输出还是电流输出。



2.10.2 寄存器

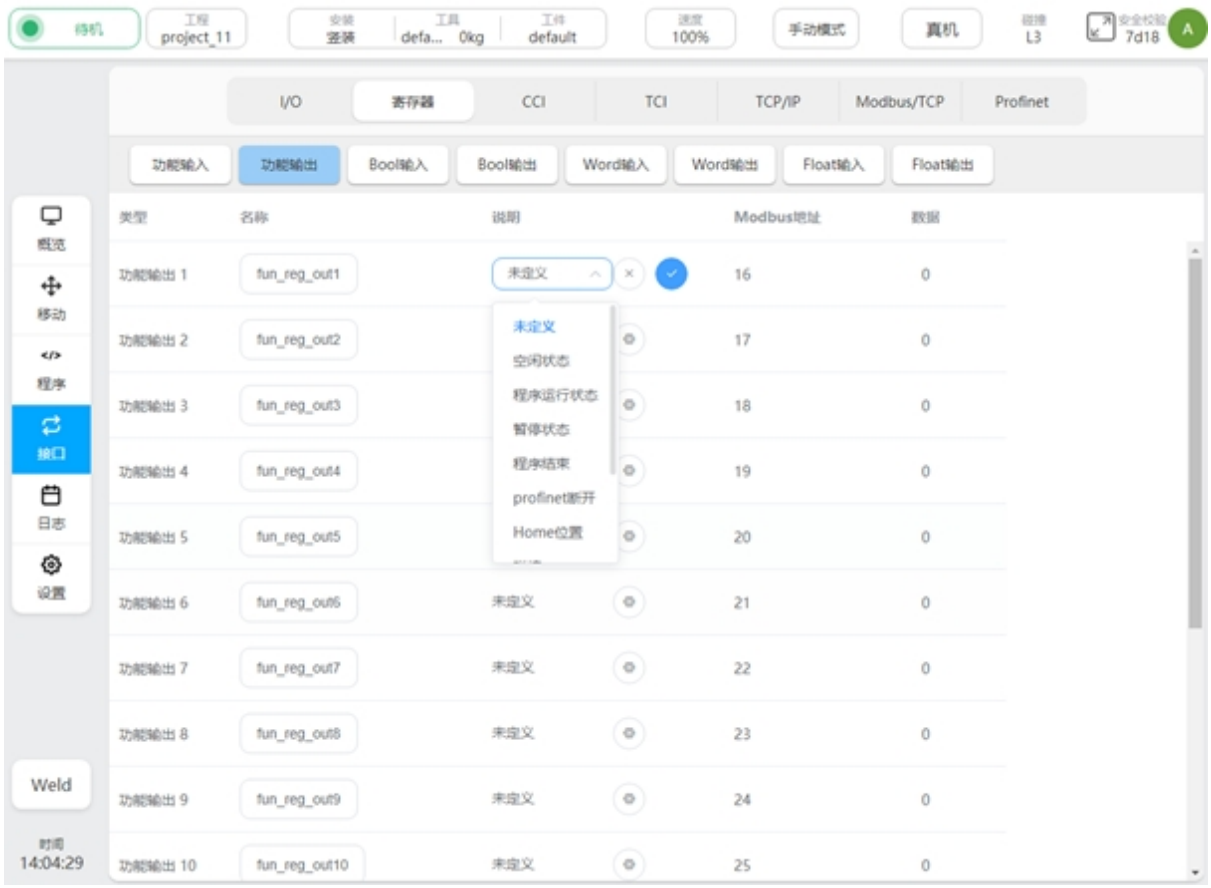
寄存器子页面与 I/O 子页面类似，页面上方是针对不同类型输入输出做的标签，分别有：功能输入、功能输出、Bool 输入、Bool 输出、Word 输入、Word 输出、Float 输入和 Float 输出共 8 种类型。单击不同标签切换显示 8 种类型的输入或输出数据信息。

寄存器功能输入界面可监控机器人内部 16 路功能输入寄存器的状态信息，如图所示。同 I/O 子页面的功能输入一样，可对功能输入名称进行修改和修改后的恢复，且可以对该类寄存器信号做特定功能项设置。目前寄存器功能输入支持的功能项同上述 I/O 子页面中功能输入支持功能。



寄存器功能输出界面可监控机器人内部 16 路功能输出寄存器的状态信息，如图所示。同 I/O 子页面的功能输出一样，可对功能输出名称进行修改和修改后的恢复，且可以对该类寄存器信号做特定系统预定义的状态量绑定。目前寄存器功能输出支持的绑定状态量有：空闲状态、程序运行状态、暂停状态、程序结束、未上电、未使能、机器人供电、机械臂运动中、自动模式、Home 位置、profinet 断开、ethernet 断开、运动任务被中止、低速模式、牵引中、碰撞。

备注： 运动任务被中止信号当且仅当机器人在运动过程中存在通过功能 IO 输入、功能寄存器输入或其他接口形式使机器人中止运动任务时输出高电平。当功能寄存器输出被配置为运动任务被中止时，当且仅当同时配置了功能 IO 输入或功能寄存器输入为中止运动任务，且通过功能 IO 输入或功能寄存器输入触发后，对应的功能 IO 输入或功能寄存器输入被复位为低电平时，功能寄存器输出同时复位为低电平。

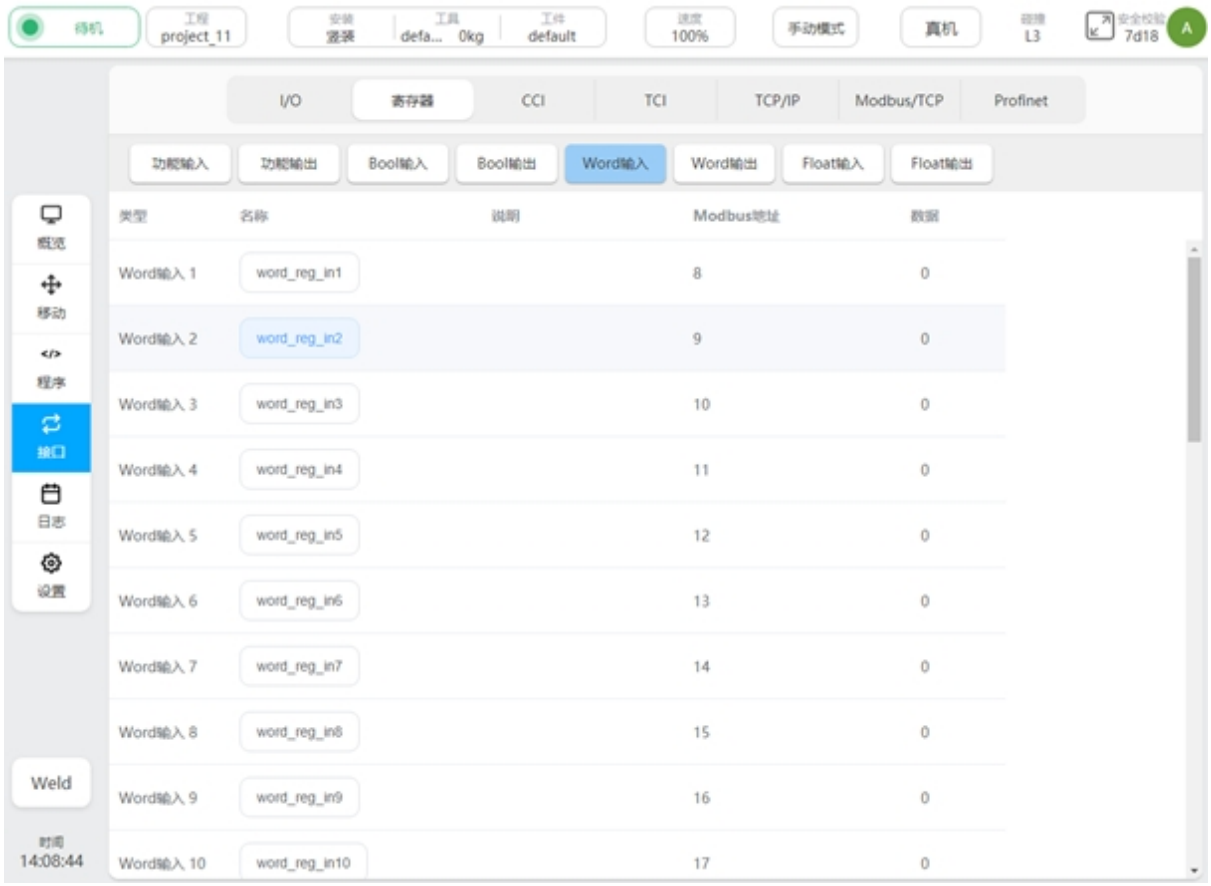


寄存器 Bool 输入界面显示 64 路布尔类型输入寄存器的状态信息。该界面可以对寄存器名称进行修改和修改后的恢复，不可设定特定功能项。

寄存器 Bool 输出界面显示 64 路布尔类型输出寄存器的状态信息。同样地，该界面只可对寄存器名称进行修改和修改后的恢复，不可设定特定功能项。

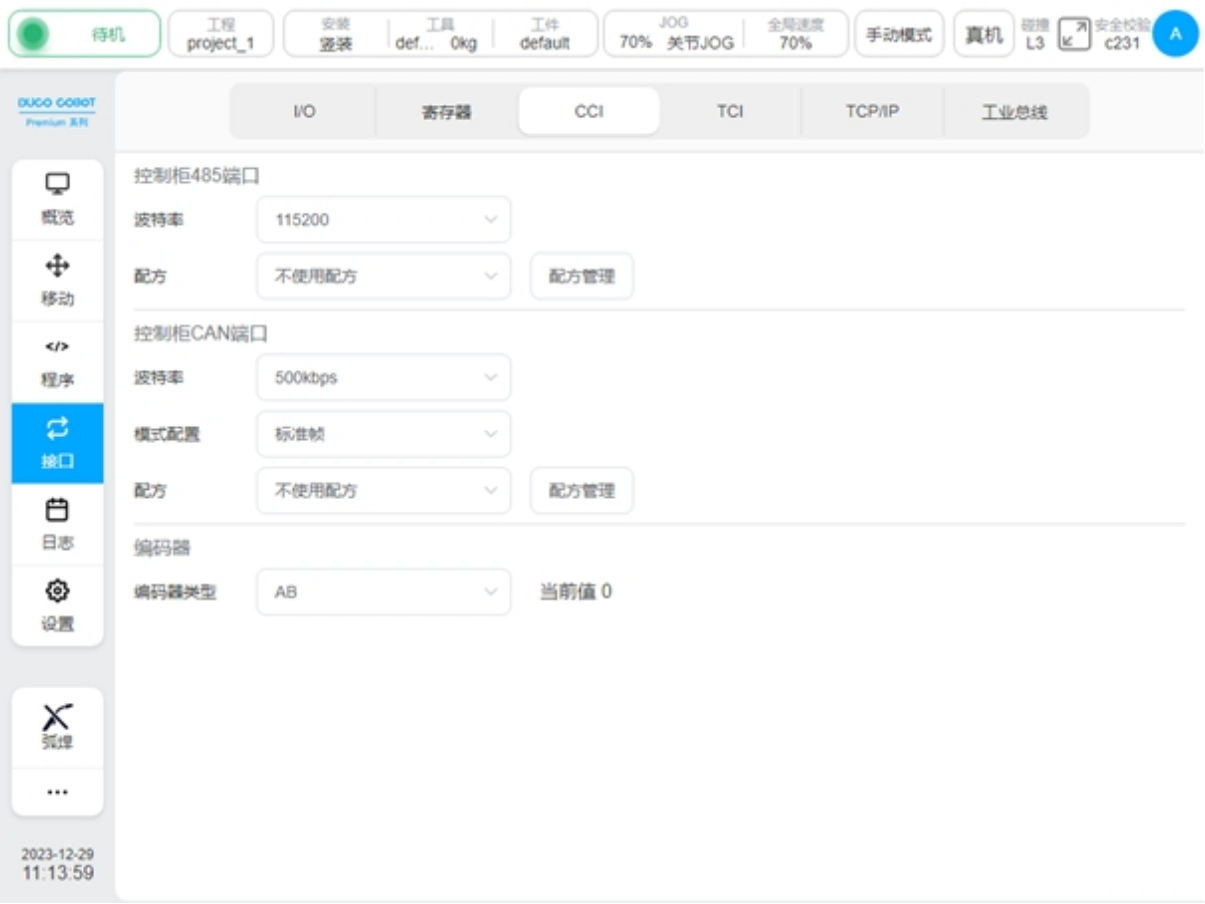
寄存器 word 输入和 Float 输入界面分别显示 32 路字节类型输入寄存器和 32 路浮点类型输入寄存器的状态信息。它们都只能对寄存器名称进行修改和修改后的恢复，不具备设置特定功能，也不能对数据列数据显示框进行操作。

寄存器 word 输出和 Float 输出界面分别显示 32 路字节类型输出寄存器和 32 路浮点类型输出寄存器的状态信息。同寄存器 Bool 输出界面类似，它们都能对寄存器名称进行修改和修改后的恢复，也能对数据列数据显示框进行操作。

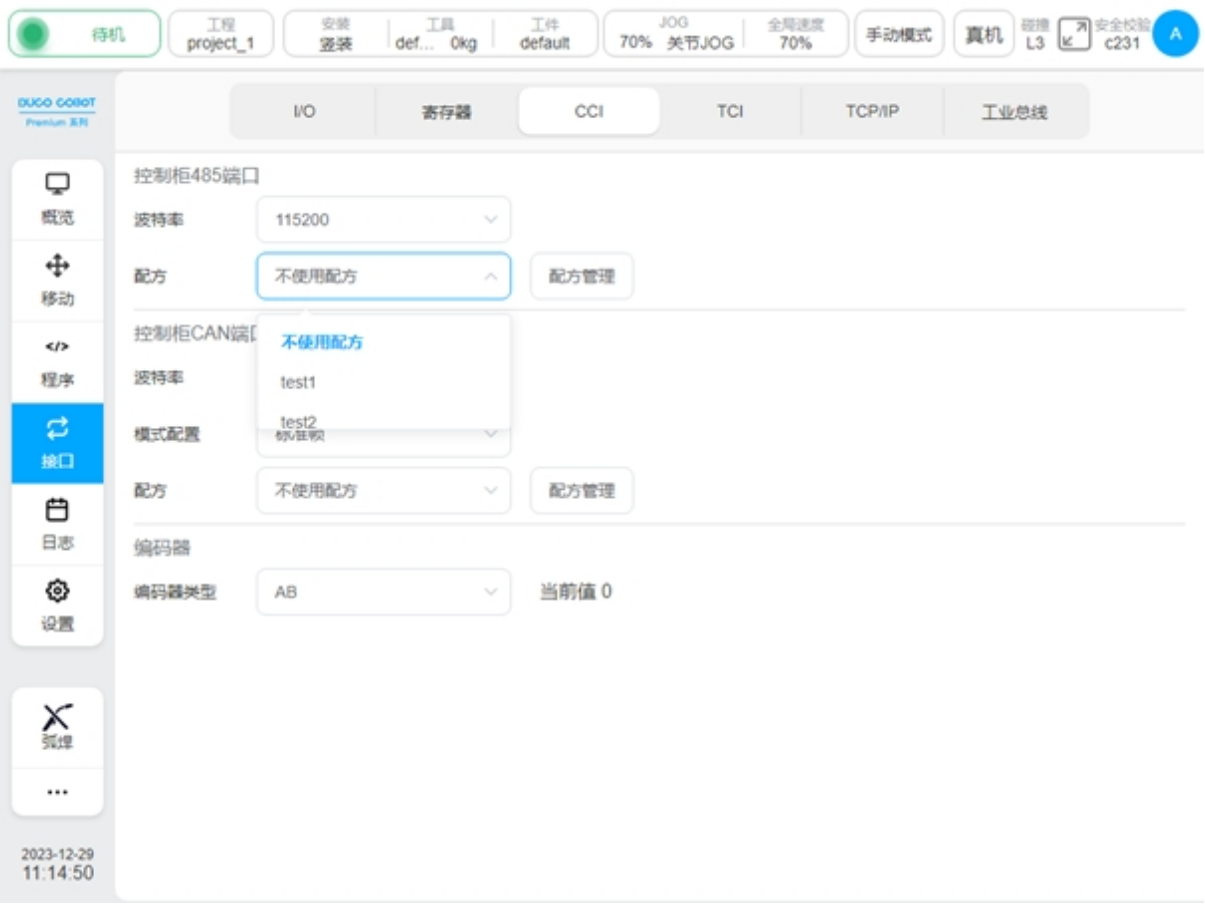


2.10.3 CCI

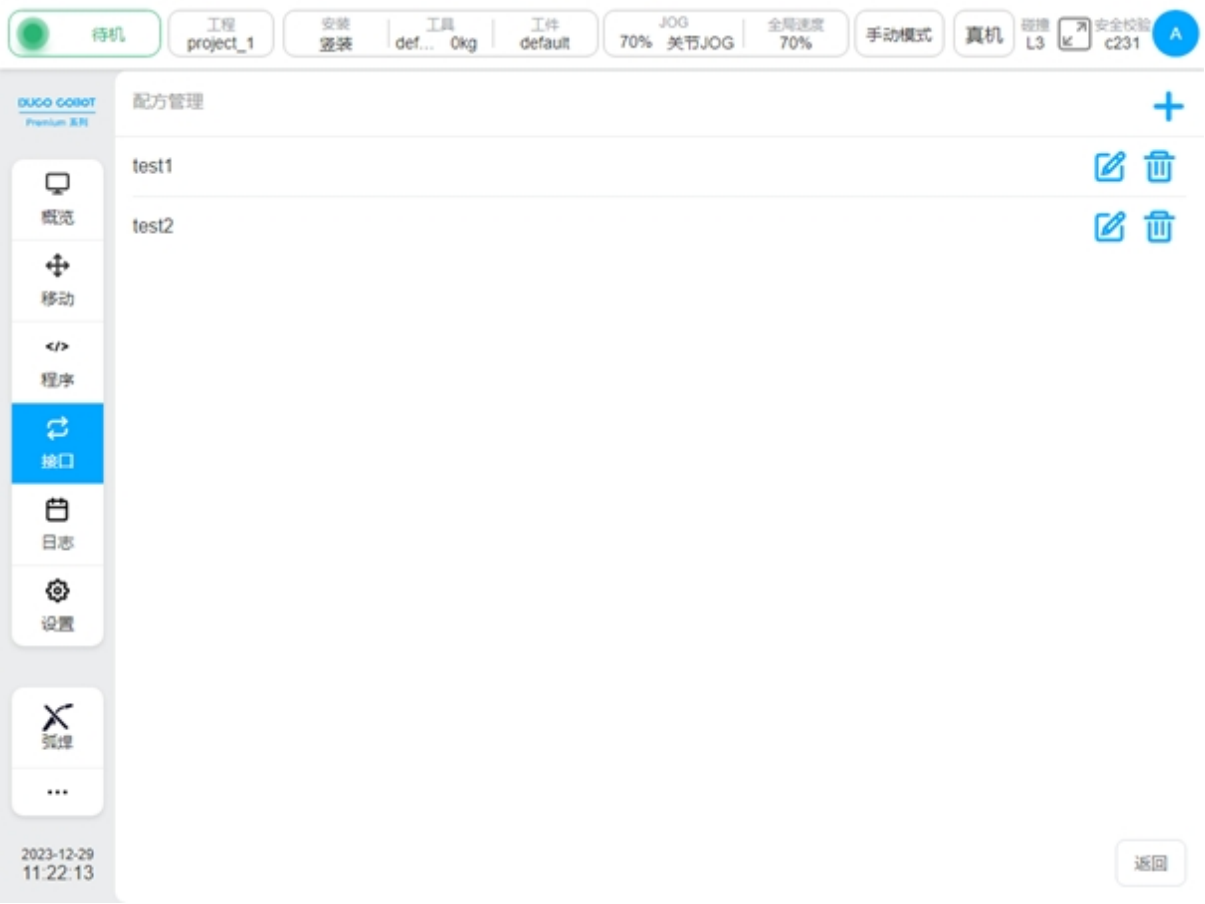
CCI 子页面是对控制柜 485 端口、CAN 端口和编码器接口进行配置，主要是设置控制柜 485 端口、CAN 端口的波特率及模式配置、配方、设置编码器类型。如图所示。



控制柜 485 端口可在波特率选择器下拉框选择该端口的波特率。485 端口支持配置波特率有：9600、19200、38400、57600 和 115200。在配方选择器下拉框选择端口的配方文件。



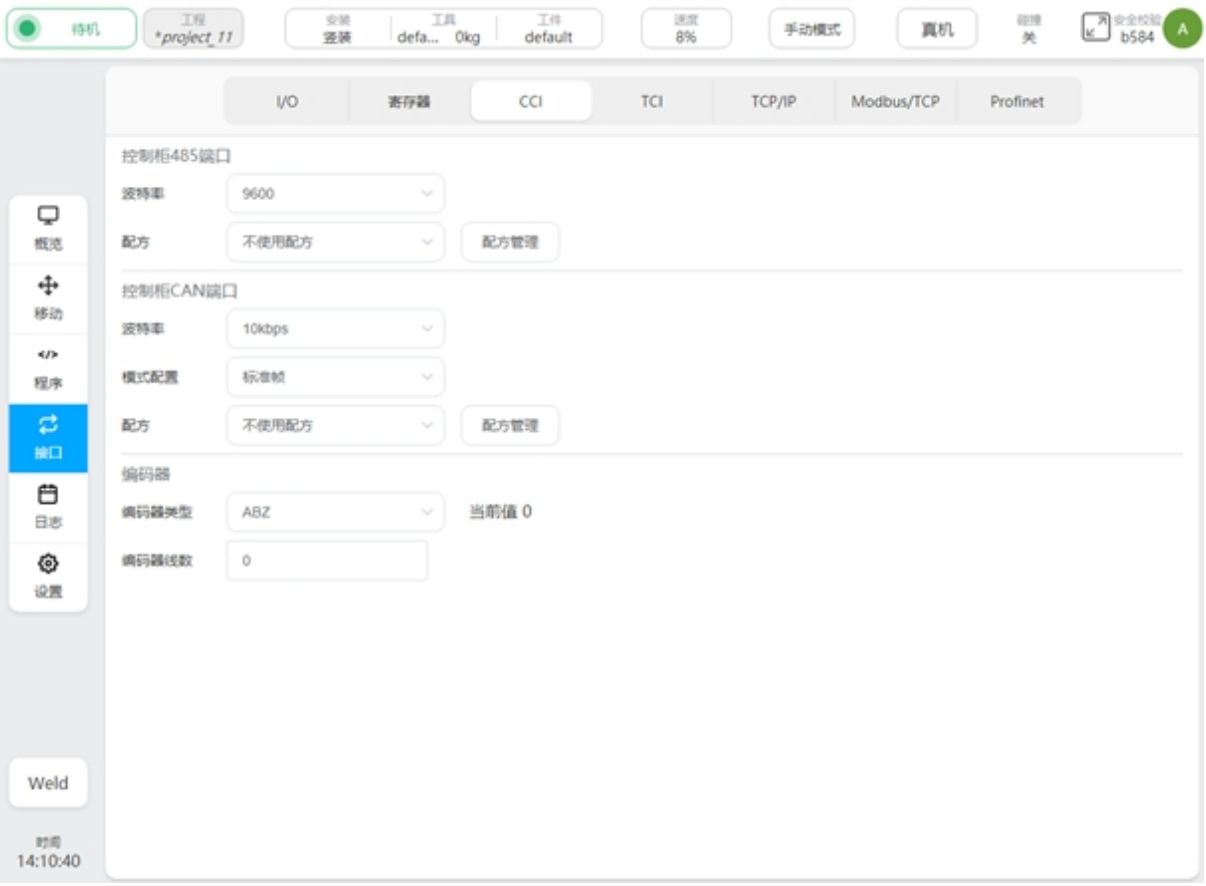
如上图所示，下拉框显示配方文件名称，配方文件是用户自定义的。单击“配方管理”按钮，进入配方管理列表框。界面显示当前工程中已存在的配方文件，用户可进行配方的添加、删除、编辑操作。



单击配方管理界面右上角 图标，可添加新配方，配方类型默认为非实时配方，也可选择实时配方，具体操作使用请参考配方使用说明。单击配方列表中任意配方文件右侧的 图标或 图标，用户可对该配方文件进行修改或删除操作。

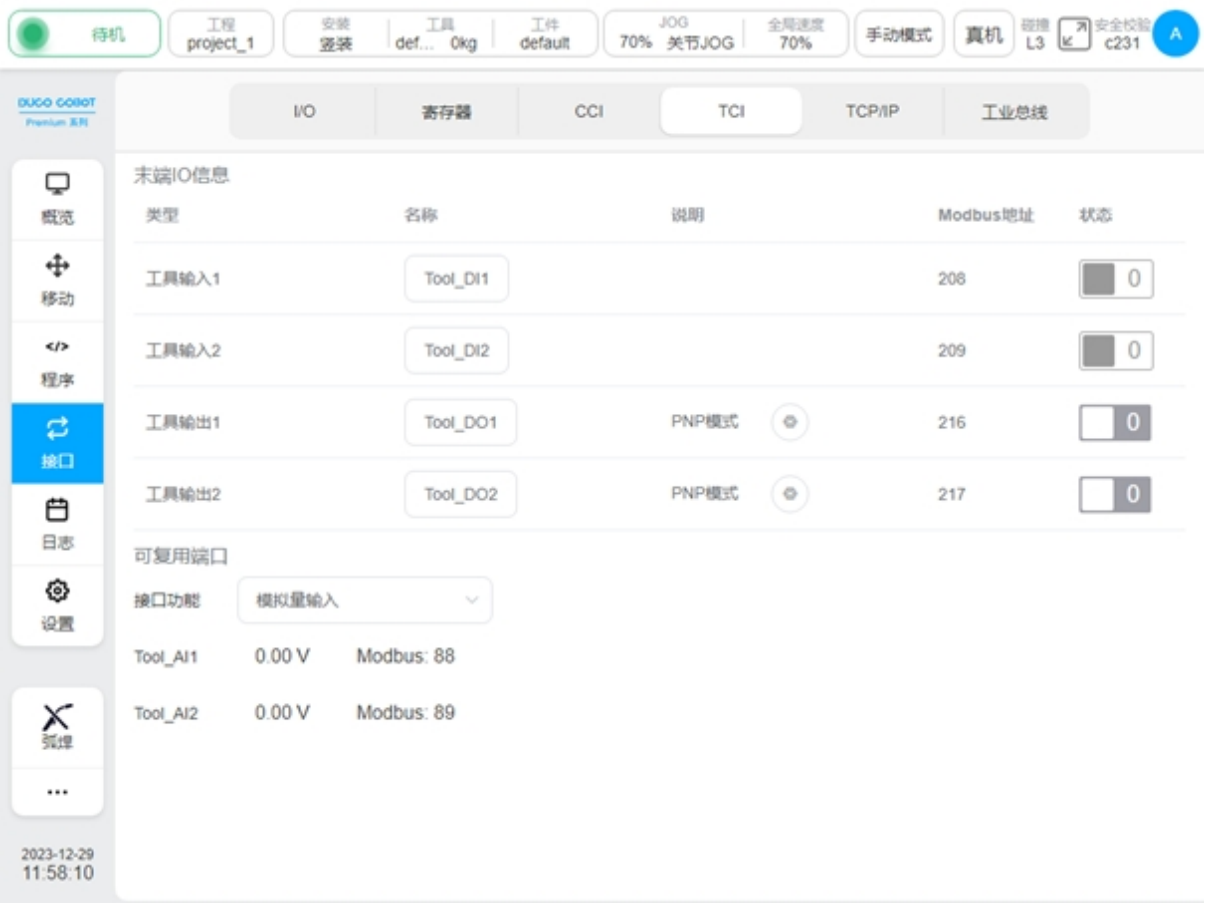
控制柜 CAN 端口的波特率和配方设置与 485 端口类似，此处不累述。CAN 端口支持配置波特率有：10kbps、20kbps、50kbps、100kbps、125kbps、250kbps、500kbps 和 1000kbps。CAN 的模式配置有两种：标准帧和扩展帧。

控制柜编码器类型配置有两种，即 AB 和 ABZ。在编码器右侧会显示当前系统所记录到的编码器脉冲数。当选择编码器类型为 ABZ 时，还需设置编码器线数，默认为 0。

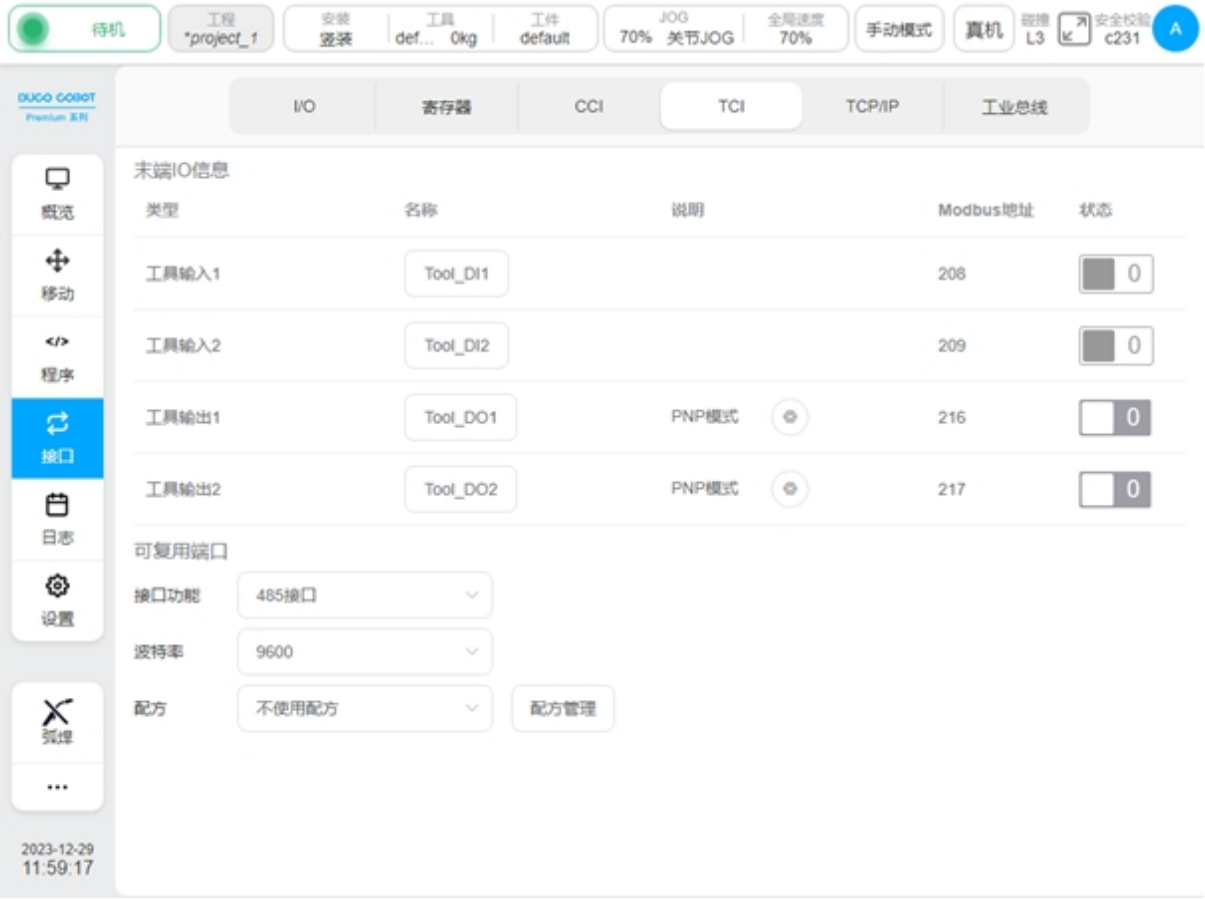


2.10.4 TCI

TCI 子页面显示末端板 IO 信息、末端可复用接口相关信息。末端板有 2 路工具输入和 2 路工具输出，与 I/O 子页面里通用输入/输出信号一样，该界面可对工具输入和工具输出名称进行修改和修改后恢复默认名称的操作。用户可配置工具输出模式，且 2 路工具输出配置模式会强制统一，即配置任意一路工具输出模式为 PNP 模式/NPN 模式，另外一路工具输出模式会跟随改变。还可以设置工具输出状态，0 表示低电平，1 表示高电平，如图所示。



用户在使用前可配置末端接口功能是模拟量输入或 485 接口，并根据配置获取对应信号。如上图所示，当选择末端接口功能为模拟量输入，界面会实时显示 2 路模拟量电压输入的数据和对应的 Modbus 地址。当选择末端接口为 485 接口，界面会显示端口的波特率和配方相关配置信息，如下图所示。



末端 485 接口可设置波特率和配方，具体操作与 CCI 子页面里控制柜 485 端口的设置类似，此处不累述。

2.10.5 TCP/IP

TCP/IP 子页面主要是显示机器人外部 TCP/IP 连接信息。主要包含默认网络、自定义服务器、自定义客户端三个子标签页。

默认网络子标签页面显示机器人作为服务器监听 2000、2001、2011 端口相关信息。即显示系统 IP、2000 和 2001 端口的连接状态和连接数量。

备注： 服务器所显示的 IP 地址，对应的是 LAN1 网口的 IP 地址。LAN1 的地址设置请参考“网络设置”部分

2000 端口可以接收机器人的控制命令。页面下方区域会显示 2000 端口收到的信息。

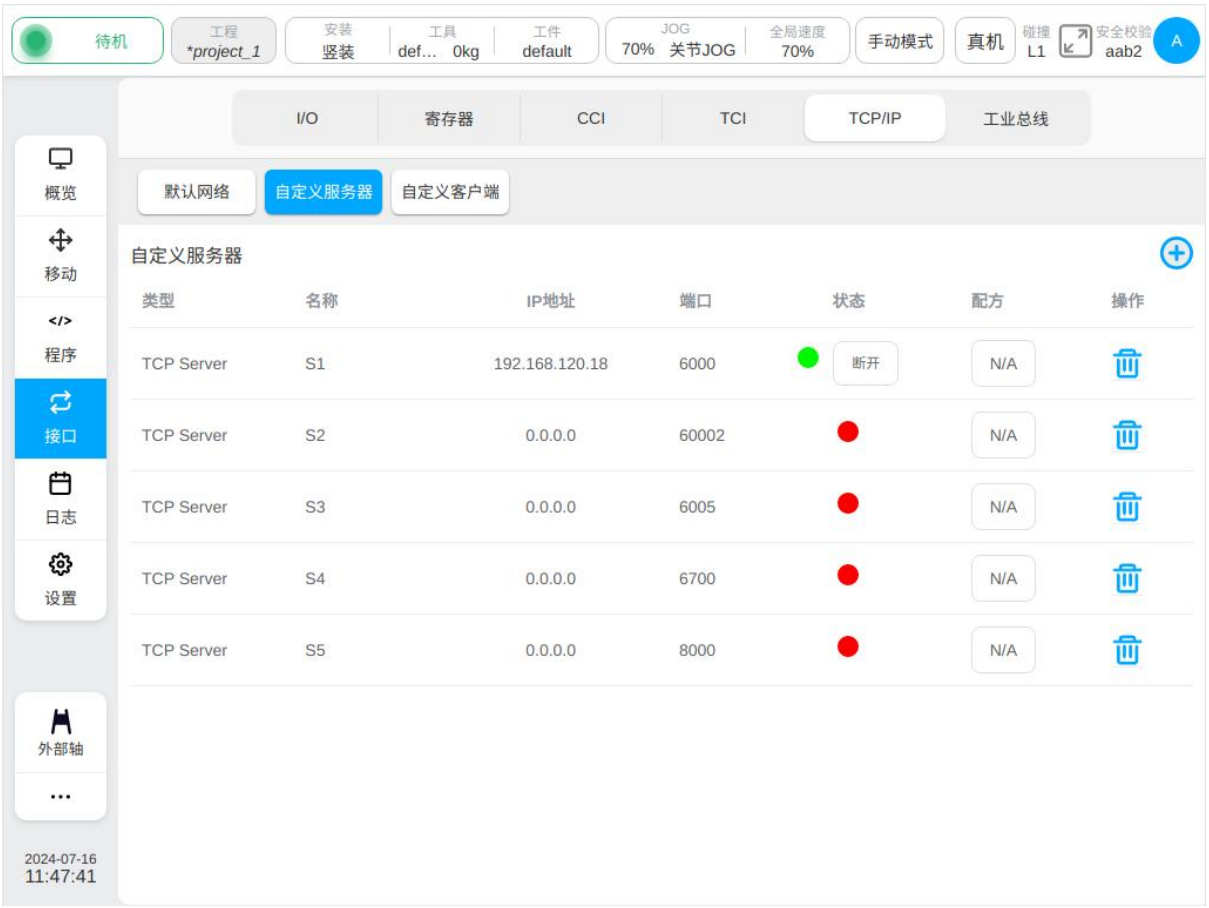
2001 端口默认会以 10Hz 频率向外发送机器人当前状态信息，频率可设置，范围 1hz-100hz

2011 端口是 UDP 端口，可以通过配方配置的数据项控制机器人。

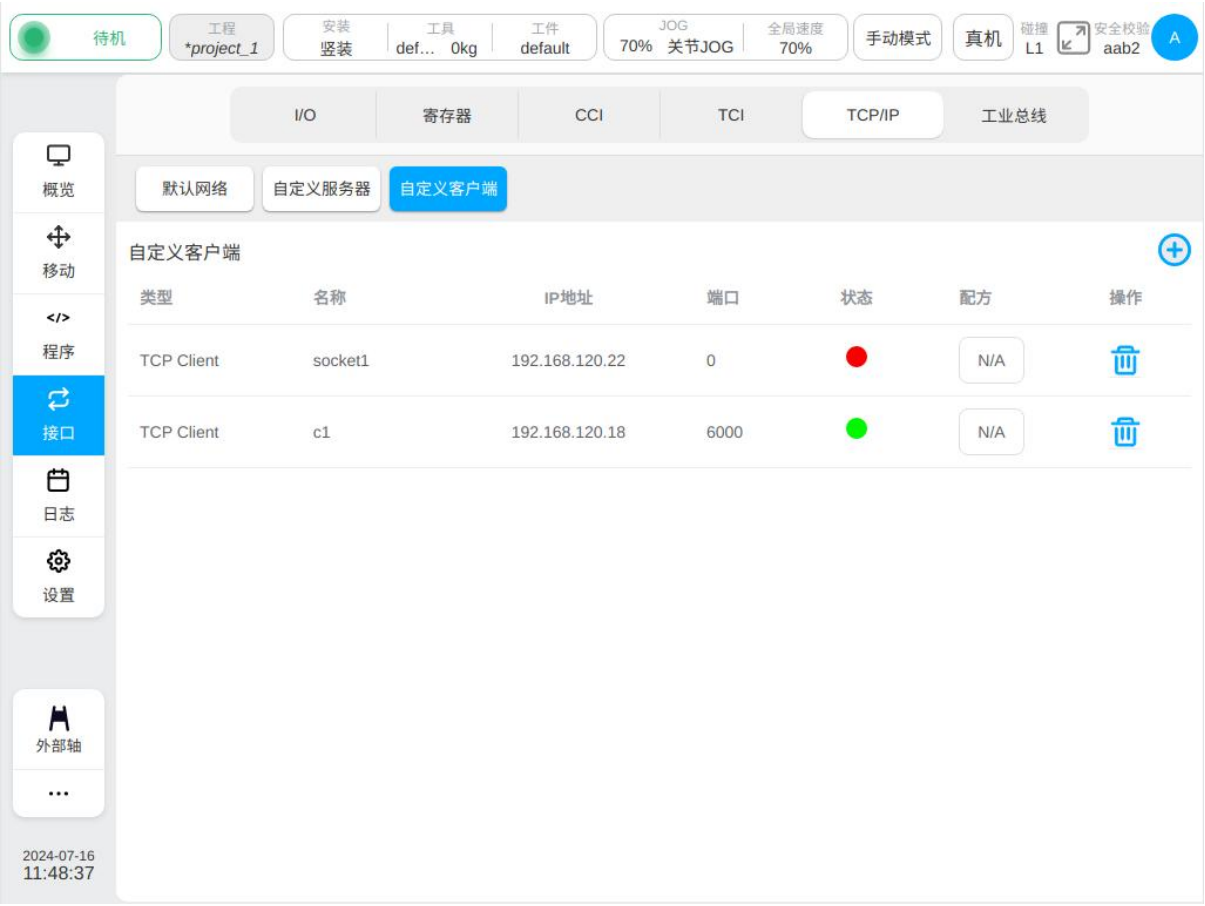
2000 和 2001 端口的详细使用请参见协作机器人对外通讯接口说明。

2011 端口的使用请参见配方使用说明。

自定义服务器子标签页显示用户创建的 socket 服务器列表信息，即自定义服务器的类型、名称、连接的客户端 IP 地址、端口、连接状态、配方以及操作图标。当连接状态灯显示绿色时，可以点击状态灯右侧“断开”按钮断开连接。点击配方按钮可以设置配方，点击操作列  图标可以删除添加的自定义服务器。点击页面右上方  图标会弹出添加连接的弹窗，输入名称和端口号，点击“确定”按钮即可添加自定义服务器。



自定义客户端子标签页显示当前机器人作为客户端的所有连接信息，即自定义客户端的类型（TCP/UDP）、名称、IP 地址、端口、配方以及操作图标。点击配方按钮可以设置配方，点击操作列  图标可以删除添加的自定义客户端。可以在程序中通过脚本来建立连接或者点击页面右上方  图标来建立。

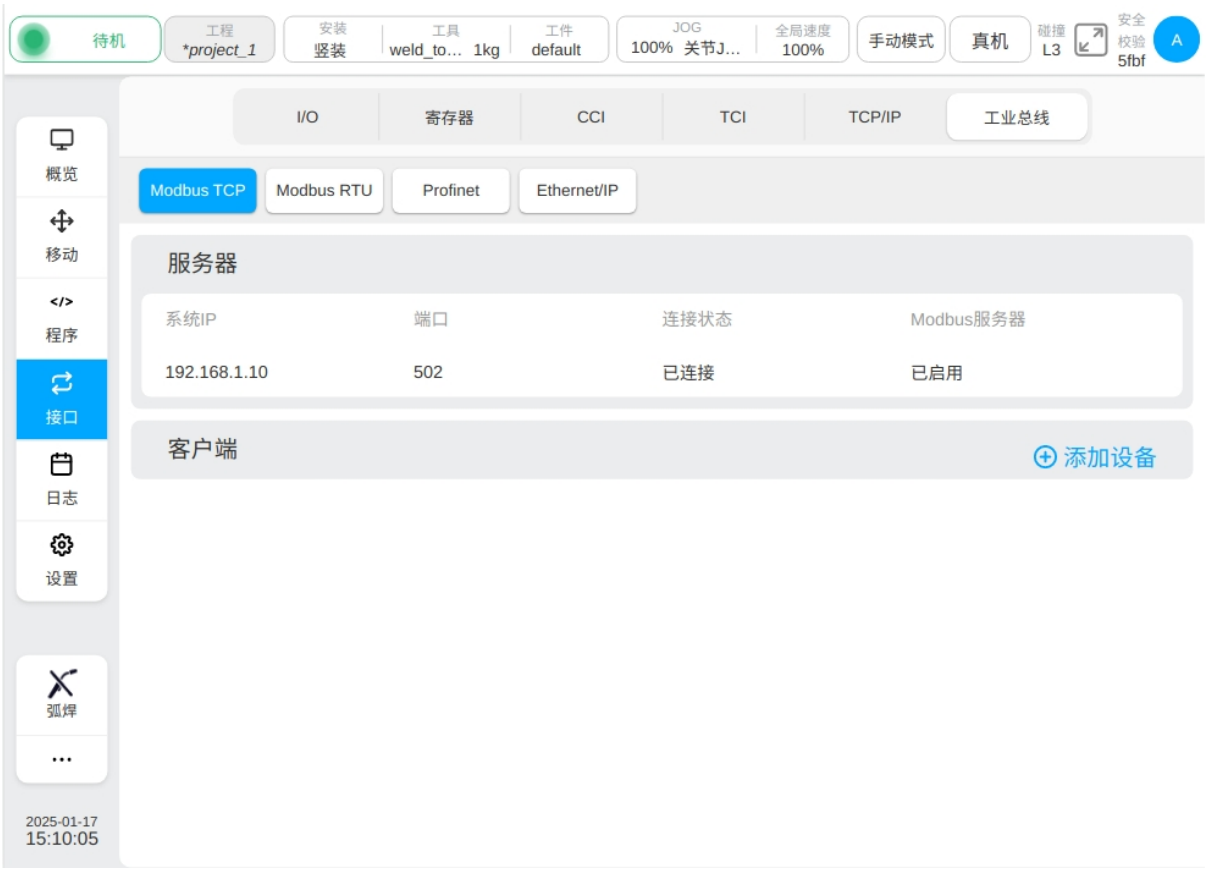


点击  图标弹出如下对话框，输入连接信息后点击“确定”即创建了一个连接。

2.10.6 工业总线

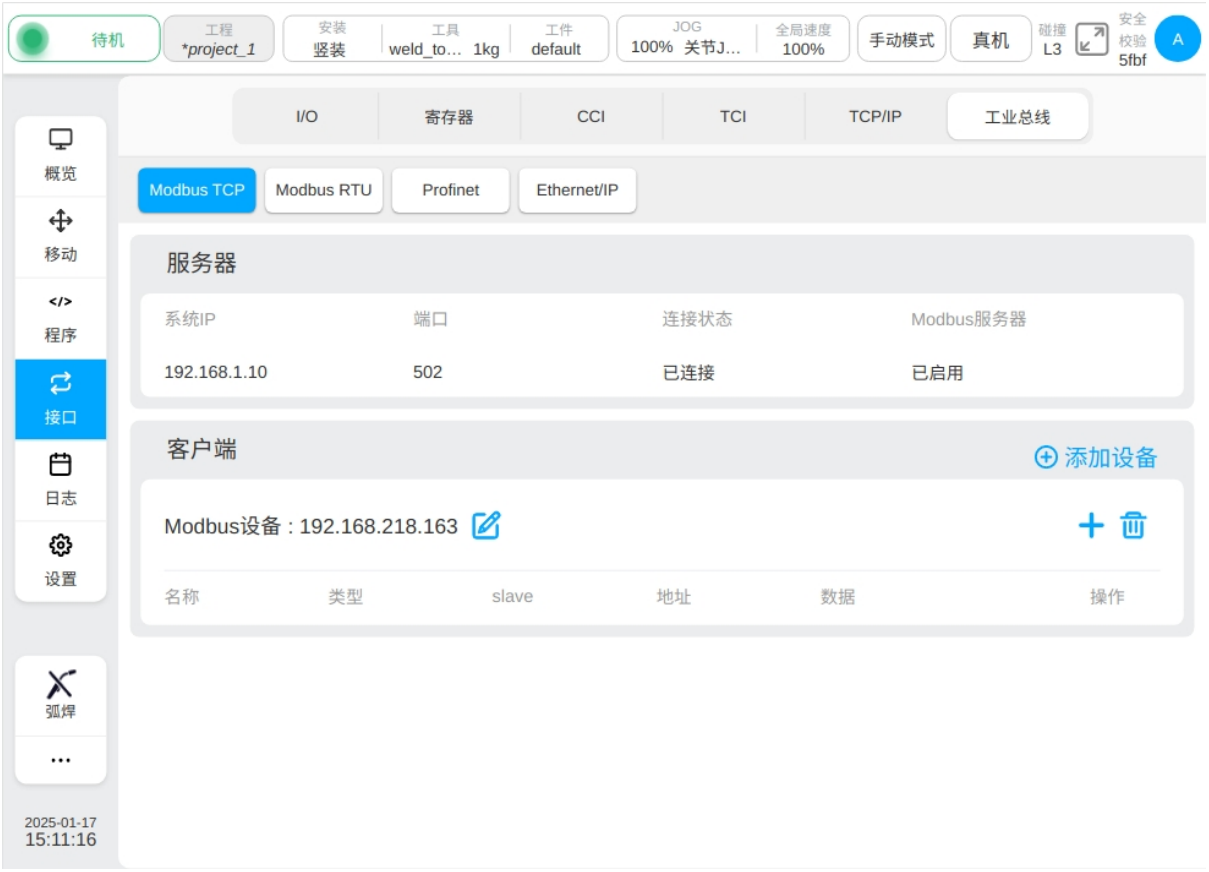
工业总线子页面主要包含 Modbus TCP、Modbus RTU、Profinet、Ethernet/IP 四个子标签页。

使用 Modbus TCP 功能，机器人在上电状态下可作为 Modbus 服务器监听外部 Modbus 客户端的连接状态，同时响应外部 Modbus 客户端设备指令读写对应 Modbus 地址内容。Modbus TCP 标签页显示机器人外部 Modbus TCP 通讯接口相关信息及操作，如图所示。

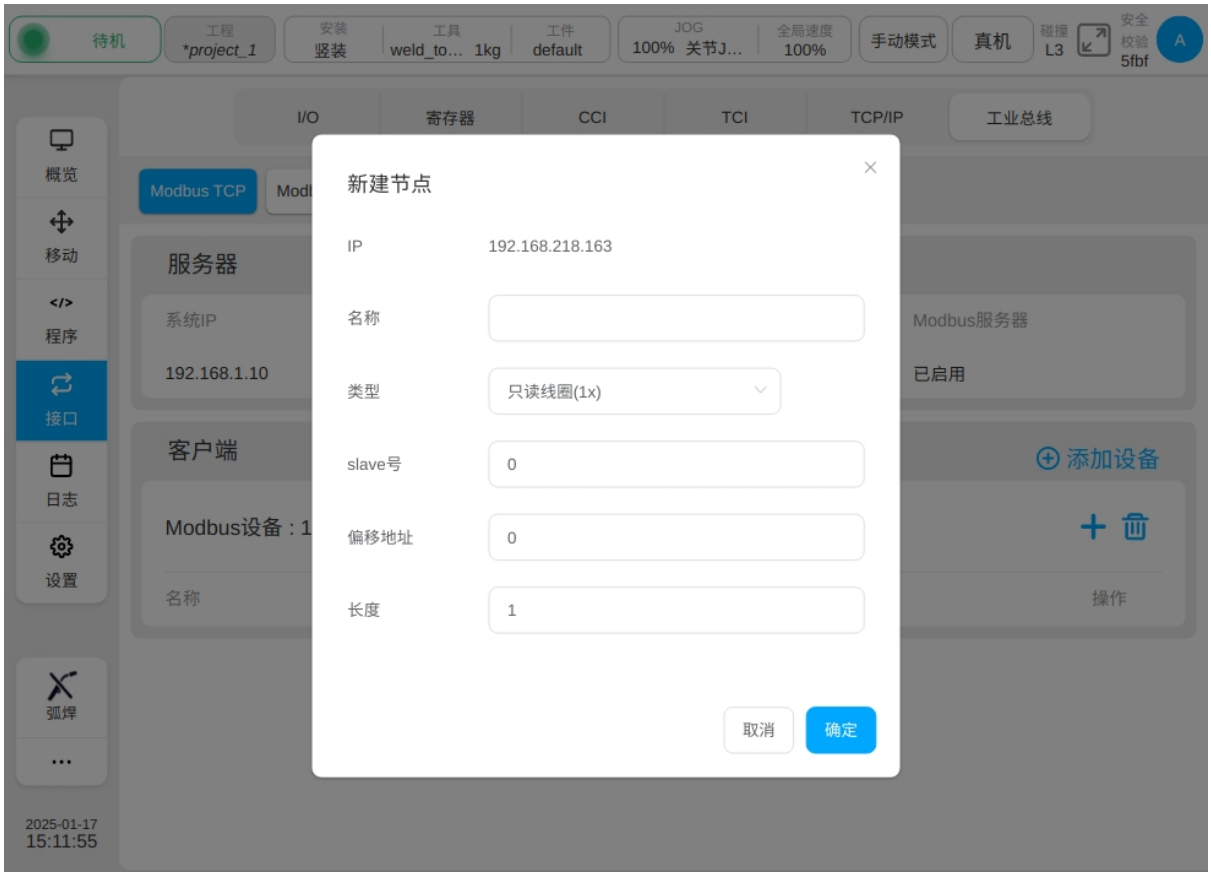


使用 Modbus TCP 功能，机器人可以作为 Modbus 客户端连接外部 Modbus 服务器设备，并读写 Modbus 服务器设备所提供的 Modbus 寄存器地址中所对应的数据内容。

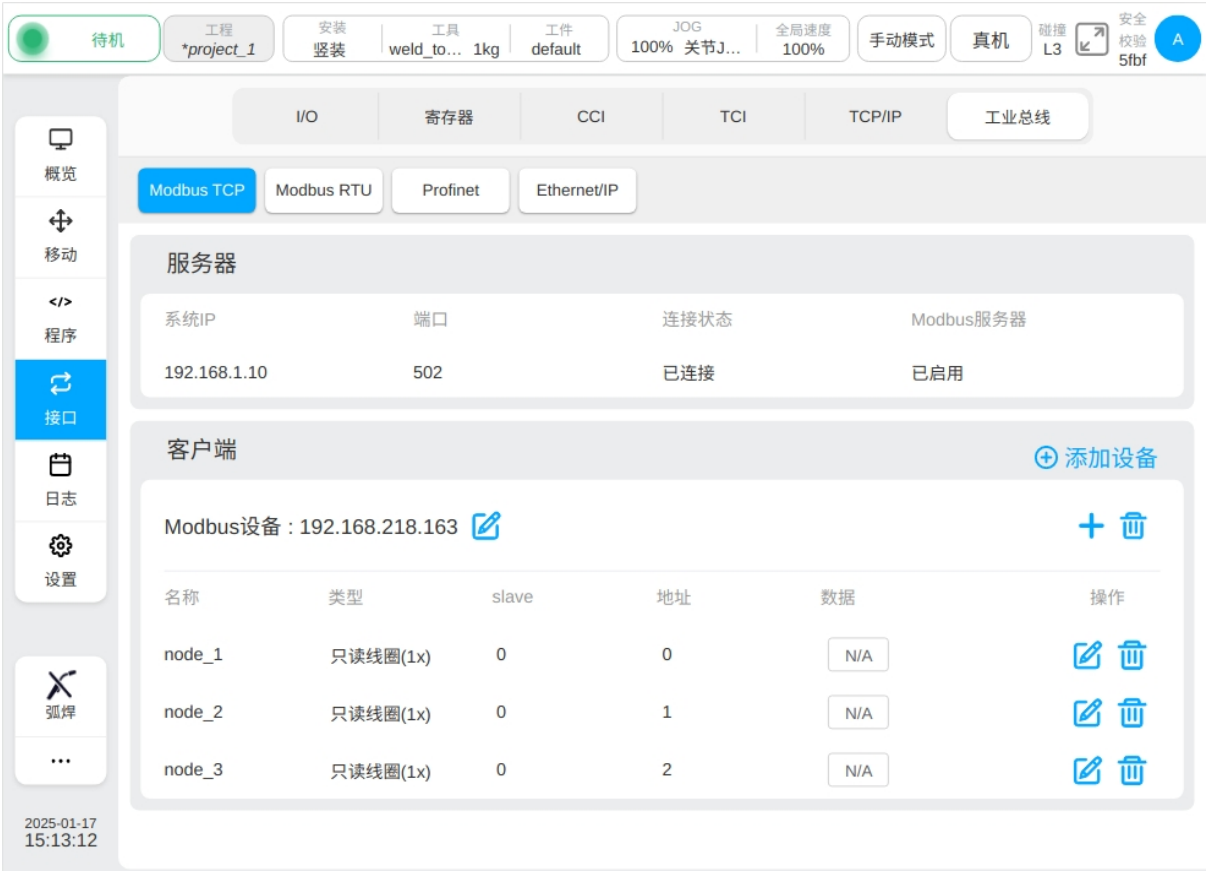
界面上方显示 Modbus 服务器系统 IP、端口、连接状态等信息。单击“添加设备”，会弹出虚拟普通键盘，输入外部 Modbus 客户端设备的 IP 地址，如输入：192.168.218.163，界面会显示该 Modbus 设备 IP 地址及相关操作图标。



单击  图标，弹出新建 Modbus 节点窗口，如图所示。输入节点名称、节点类型、slave 号、偏移地址、长度，单击“确定”按钮即可新建成功。其中，节点类型共 4 种：只读线圈 (1x)、可读写线圈 (0x)、只读寄存器 (3x)、可读写寄存器 (4x)。

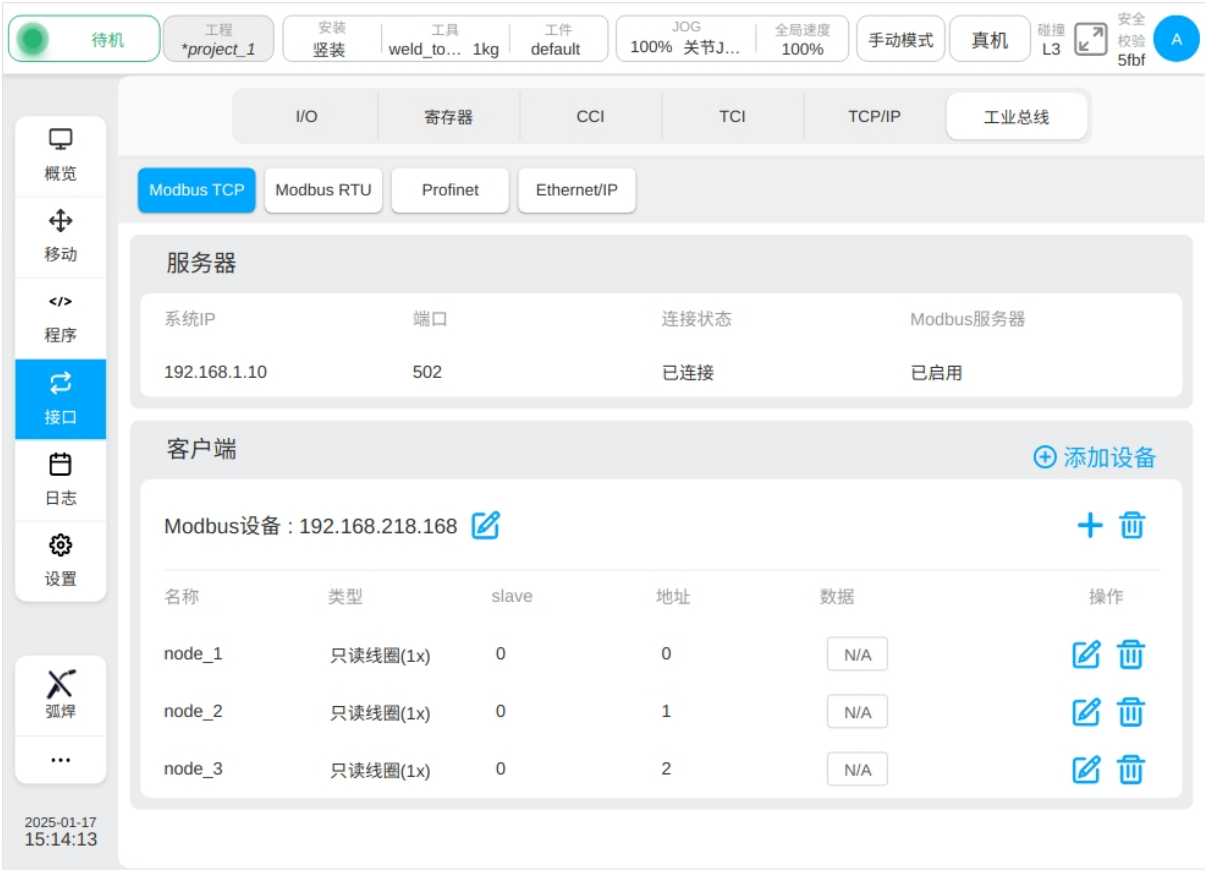


新建节点时，长度值默认为 1。当长度值为 1 时，创建单个节点；当长度值大于 1 时，会批量创建出多个节点。假如长度设置为 3，根据设置长度节点名称后依次加下划线后缀，即：node_1、node _2、node _3，地址依次为 0、1、2，如图所示。

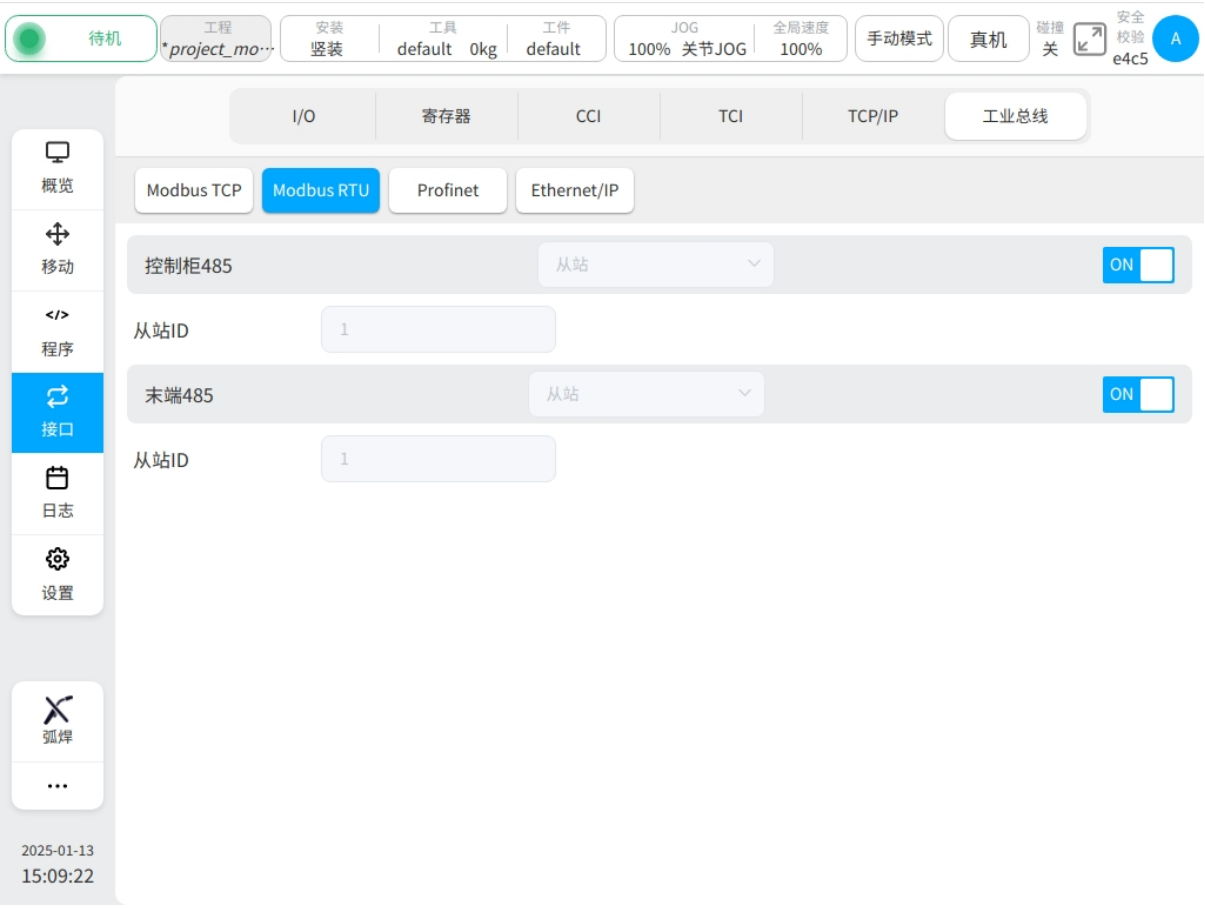


新建节点后，界面会显示新定义的节点相应信息。数据列无相应数据时显示“N/A”，用户可对每个节点进行修改和删除操作。单击对应节点行的  按钮，弹出窗口与新建节点时类似，只是节点名称不可更改，长度不显示，节点类型、slave 号、偏移地址均可更改。

用户可修改新创建的 Modbus 设备 IP 地址，单击 Modbus 设备 IP 地址后的  图标，会弹出虚拟键盘，输入目标 IP 地址，如输入：192.168.218.168，点击“OK”键后，界面显示如下。Modbus 设备 IP 地址修改完成后，新 IP 地址的节点信息保留与 IP 地址修改前一致。

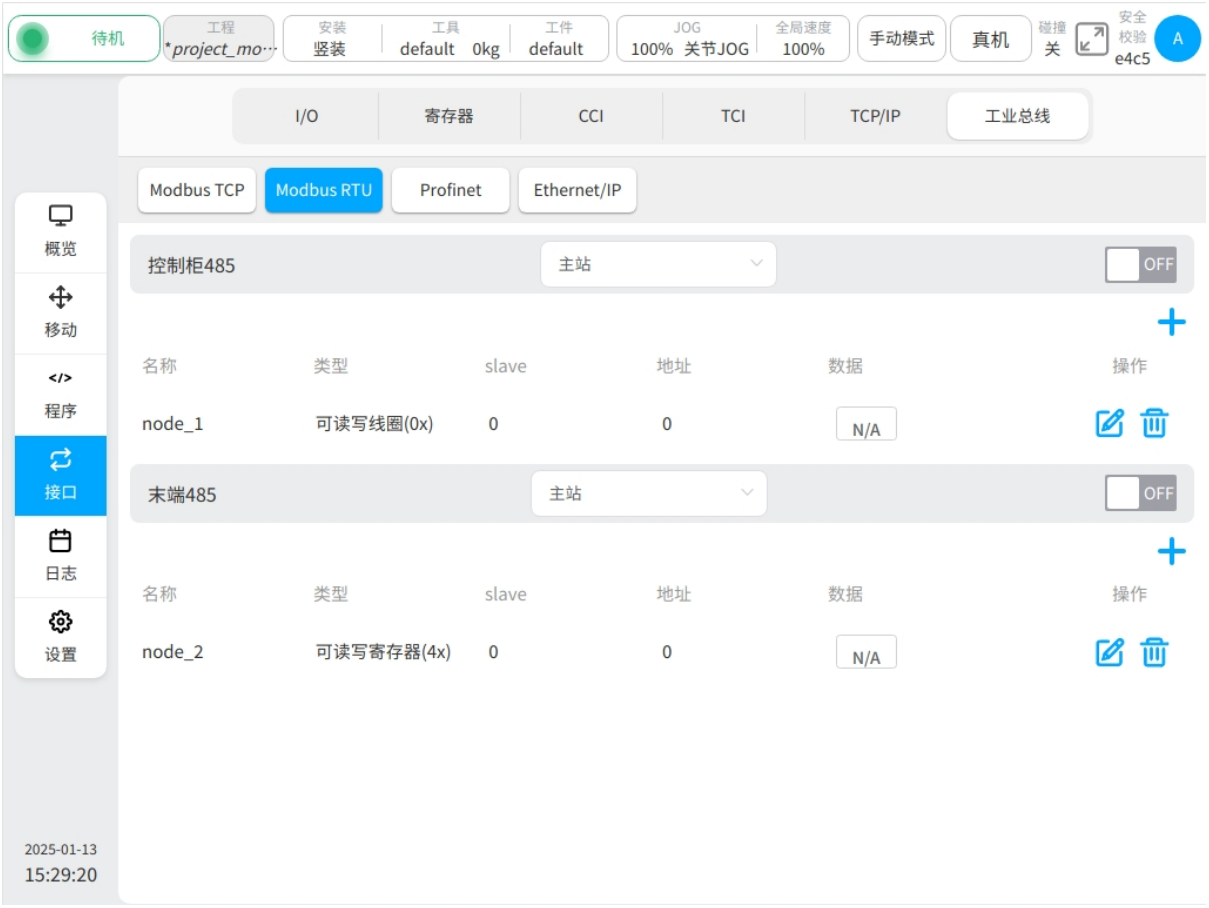


使用 Modbus RTU 功能，机器人在上电状态下可通过控制柜 485 接口与机器人末端 485 接口作为 Modbus 从站响应外部连接的 Modbus 主站设备指令读写对应 Modubs 地址内容。

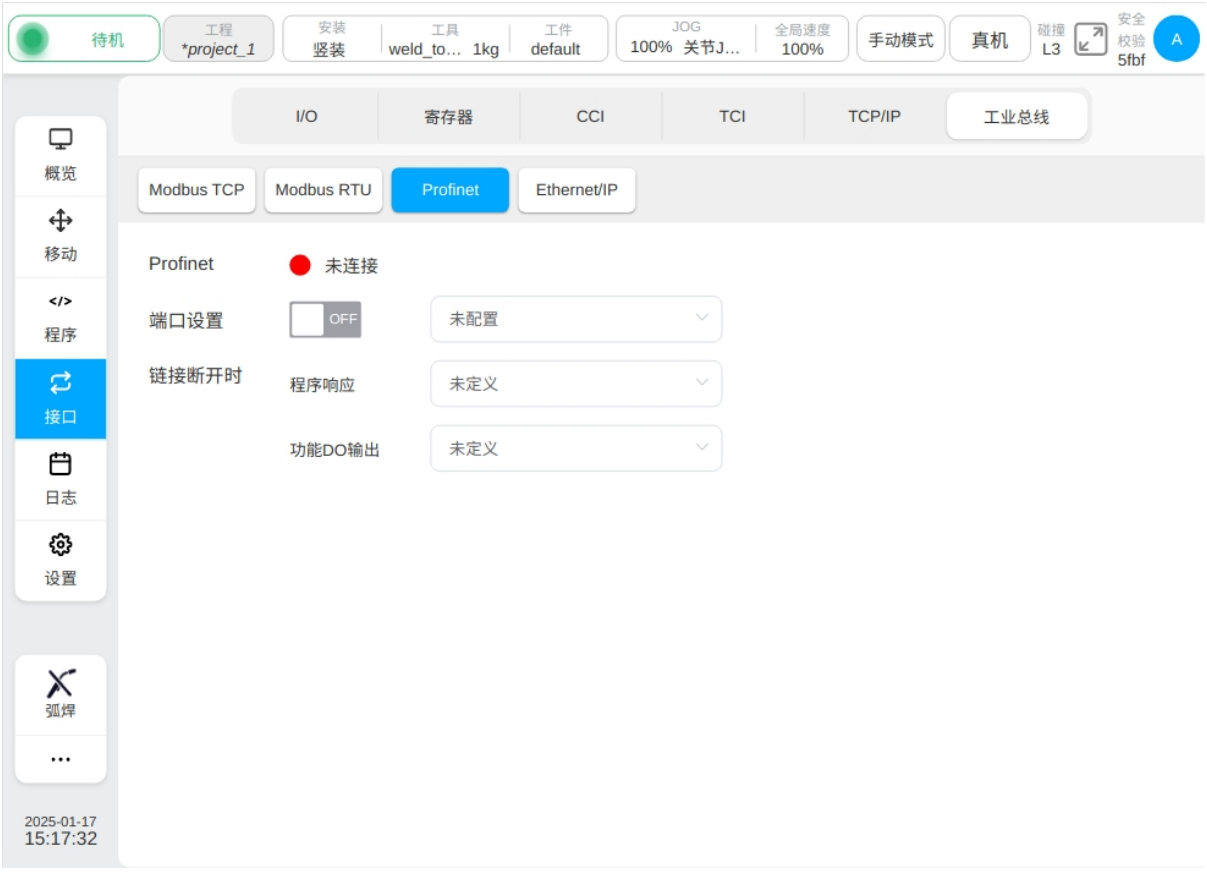


使用 Modbus RTU 功能，机器人可以在上电状态下通过控制柜 485 接口与机器人末端 485 接口作为 Modbus 主站配置并连接外部连接的 Modbus 从站设备，控制并读写 Modbus 从站设备对应的 Modbus 地址内容。关于目标 Modbus 从站设备节点的新增、删除、编辑部分内容，与 Modbus TCP 功能中机器人作为 Modbus 客户端对目标 Modbus 服务器设备节点的操作相同。

备注： 当使用 Modbus RTU 功能令控制柜 485 与机器人末端 485 接口工作于 Modbus RTU 模式并启用时，对应的 485 接口无法通过 485 操作相关脚本或接口进行操作，仅可通过 Modbus 操作脚本或接口对目标地址的 Modbus 数据内容进行读写操作。

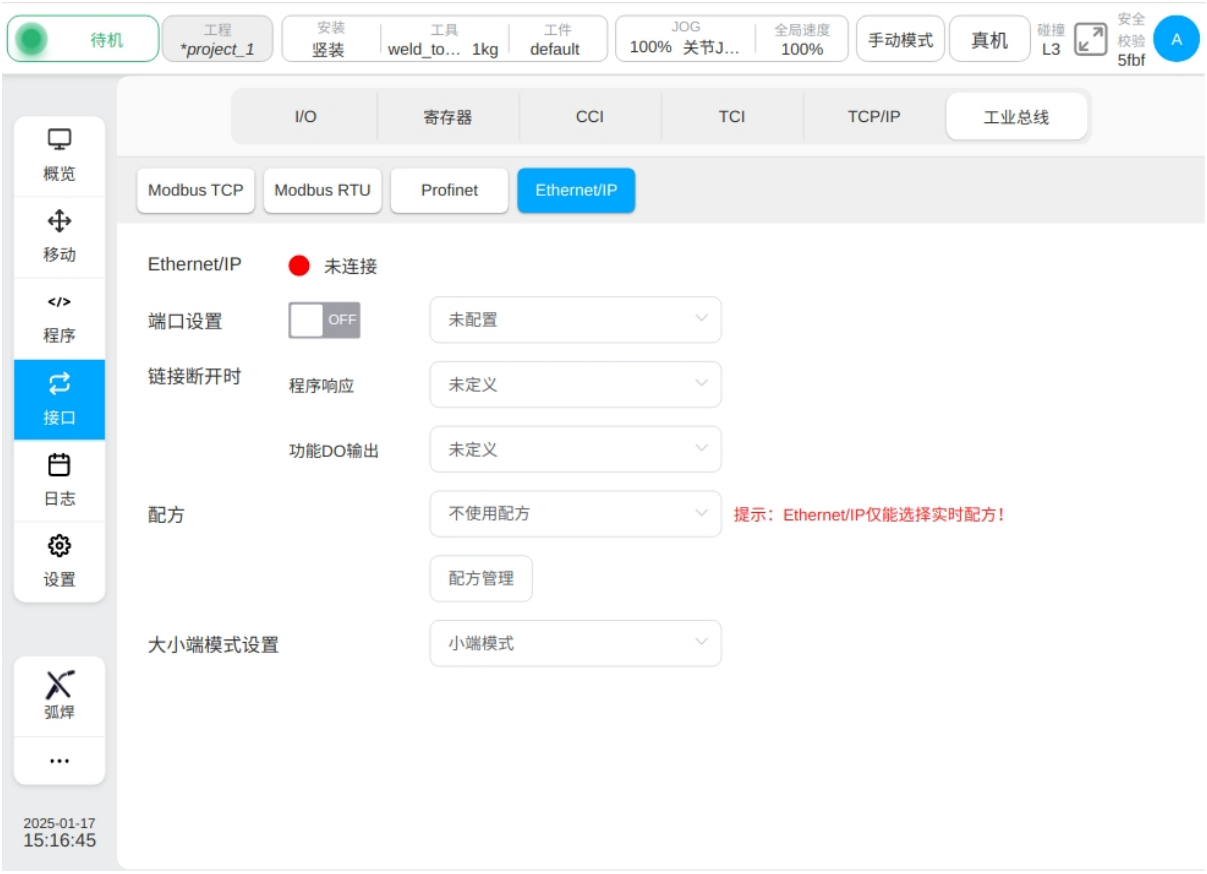


Profinet、Ethernet/IP 子标签页分别主要是显示连接状态、端口设置和断开连接时处理配置，包括程序响应动作及对应配置的功能输出 IO。端口设置主要包括对端口是否启用、配置对应端口名称的功能；程序响应选择器可选项有：未定义、暂停程序、停止程序；功能 DO 输出选择器下拉选项包含：未定义和 I/O 子页面中功能输出信号中未定义的输出信号名称。各子标签页面下配置的 Profinet、Ethernet/IP 断开对应的功能 DO 与 I/O 子页面中功能输出配置保持同步。



另外，Ethernet/IP 子标签页还可进行配方配置、配方管理以及大小端模式设置。

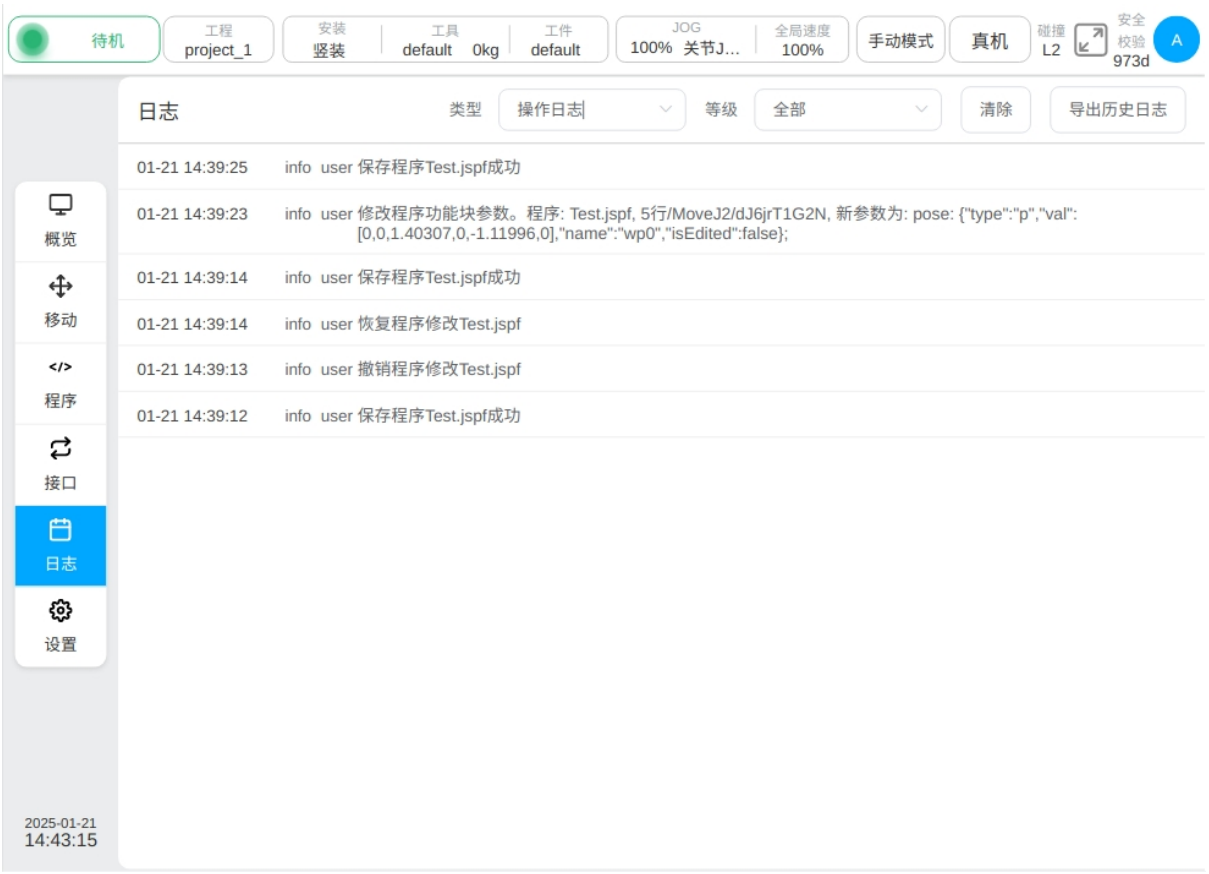
备注： 注意：Ethernet/IP 仅能选择实时配方！



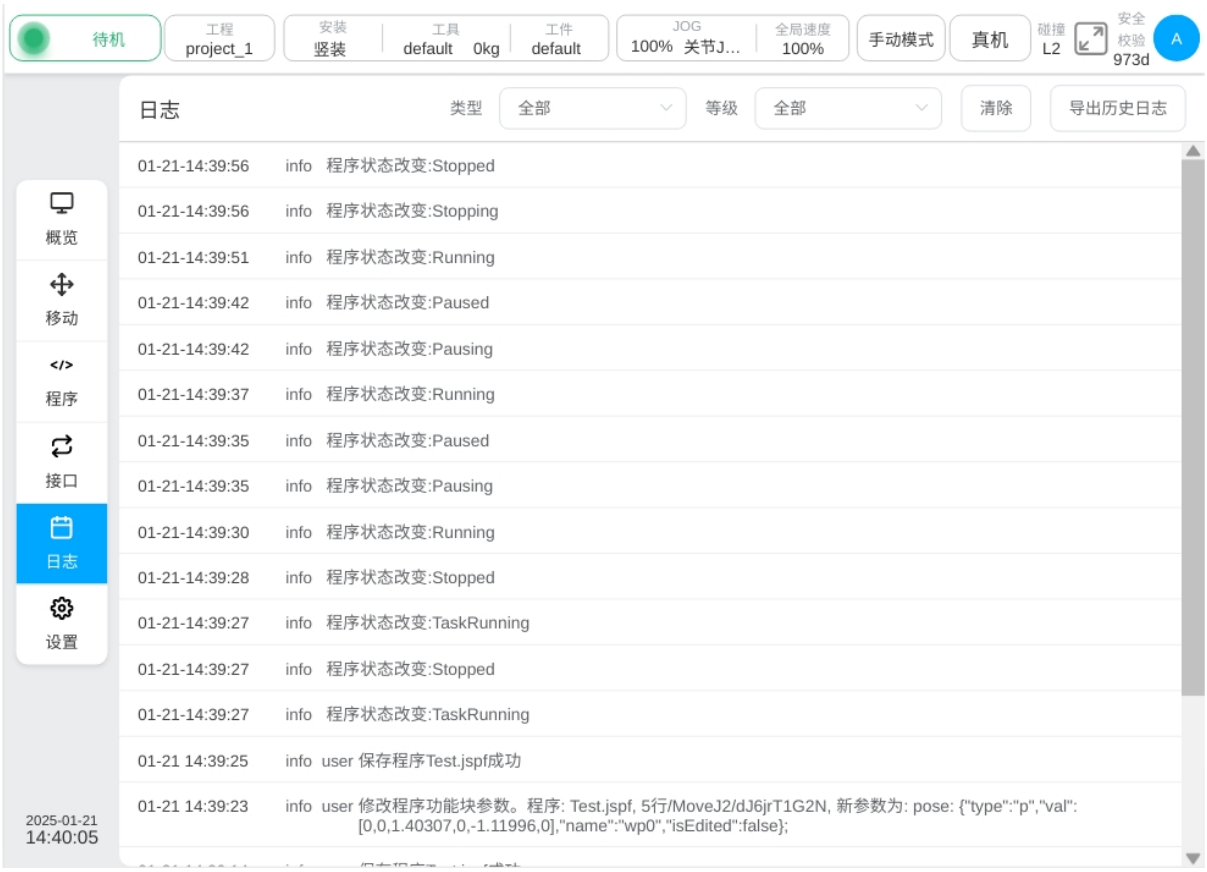
2.11 日志页面

单击导航栏中“日志”会进入日志页面。日志页面是记录机器人使用及运行过程中用户操作与机器人触发的相关信息记录，分为致命、错误、警告和信息。该页面日志信息与概览页面中日志信息保持一致。

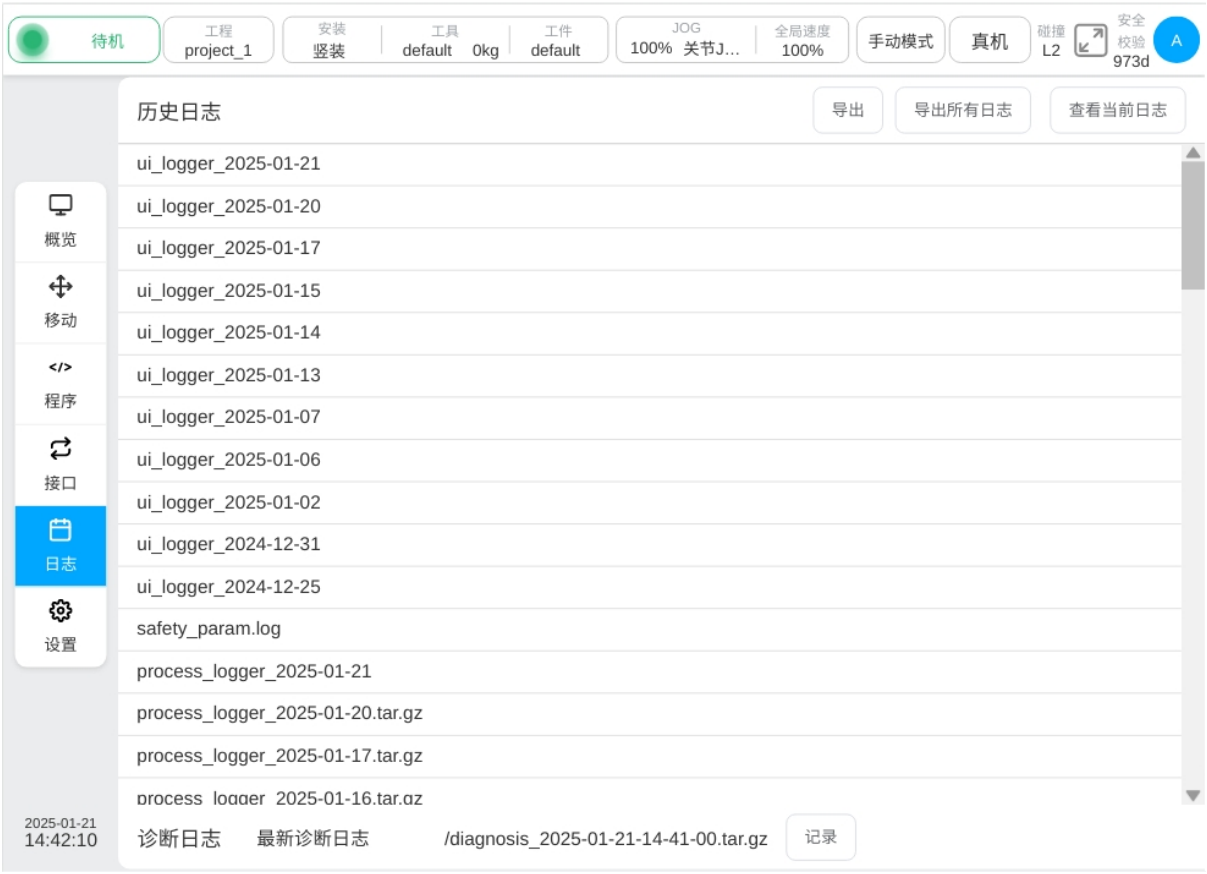
选择日志类型为“操作日志”类型，可以单独显示所有用户操作对机器人工程配置内容产生变动的操作内容历史。



选择日志类型为“全部”类型，可以显示包括操作日志在内的所有机器人运行过程中的重要信息。日志信息可作为诊断问题、处理历史数据的追踪重要来源。当在操作机器人时出现相关错误报警信息时，可到日志页面查询错误报警原因进行检查。当问题无法解决时，请及时联系相关技术支持人员，并告知日志信息以便快速解决问题。

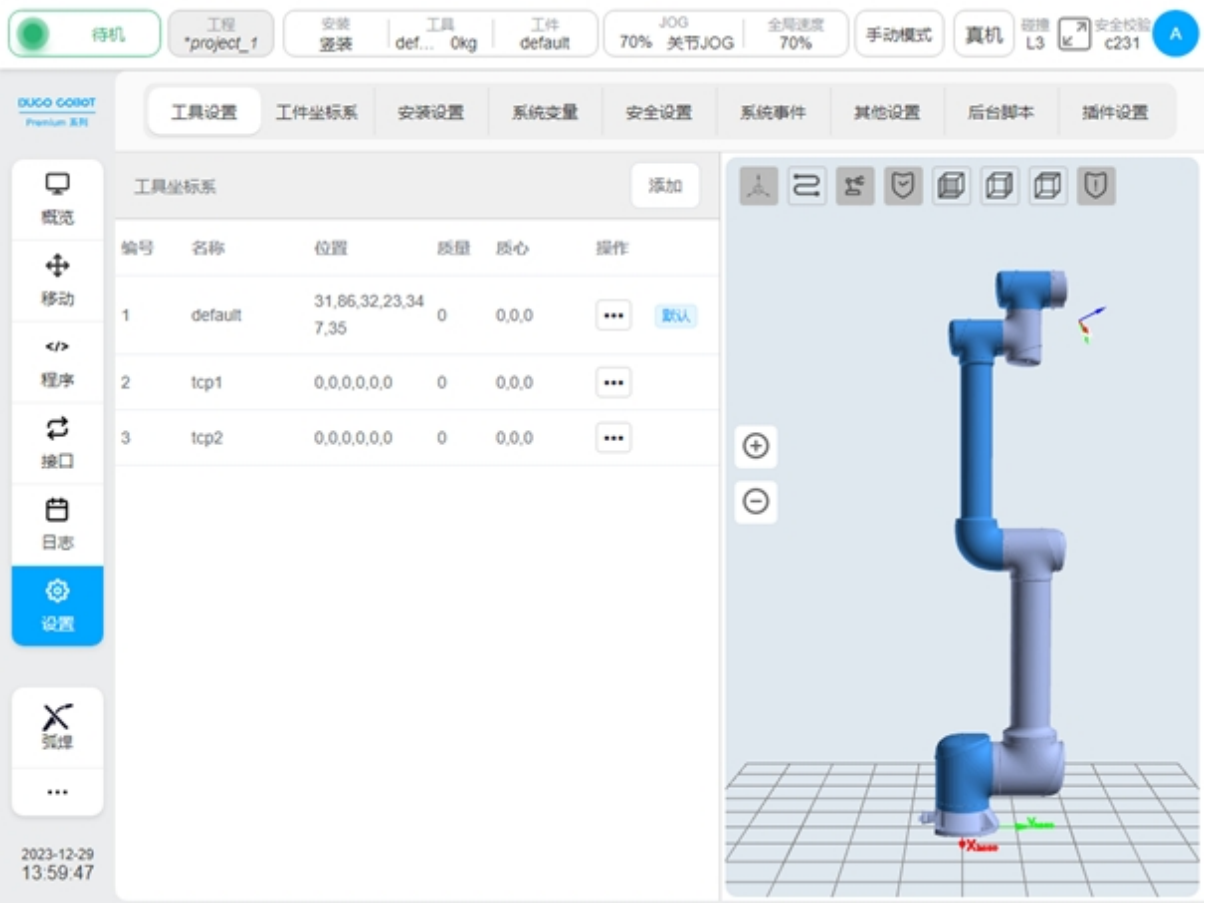


单击“导出历史日志”按钮，页面上方区域会显示所有历史日志文件列表信息，底部区域显示最新诊断日志信息以及主动触发记录诊断日志的“记录”按钮。单击页面右上角“导出”按钮可以导出其中某个日志文件，单击“导出所有日志”按钮可以导出所有日志。单击“查看当前日志”按钮，又会返回到当前日志页面。单击底部诊断日志区域“记录”按钮，可以主动触发记录诊断日志，有利于用户现场进行特性时刻的诊断日志记录，并且最近一次记录诊断日志超过 1 分钟后才可再次主动触发记录。



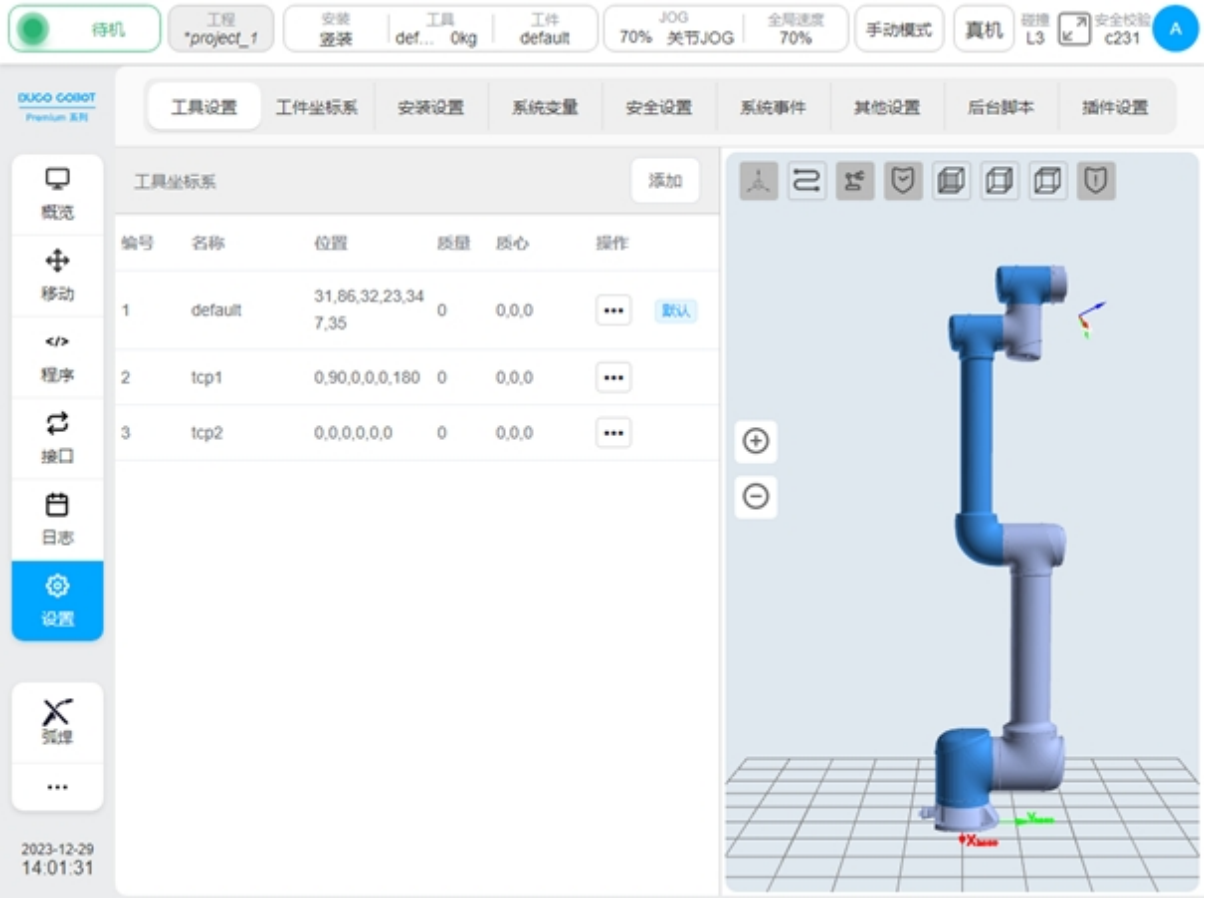
2.12 设置页面

单击导航栏中“设置”会默认进入设置页面下工具设置子页面。设置页面下有工具设置、工件坐标系、安装设置、系统变量、安全设置（详见第 5 章）、系统事件、其他设置、后台脚本、插件设置共 9 个子页面。

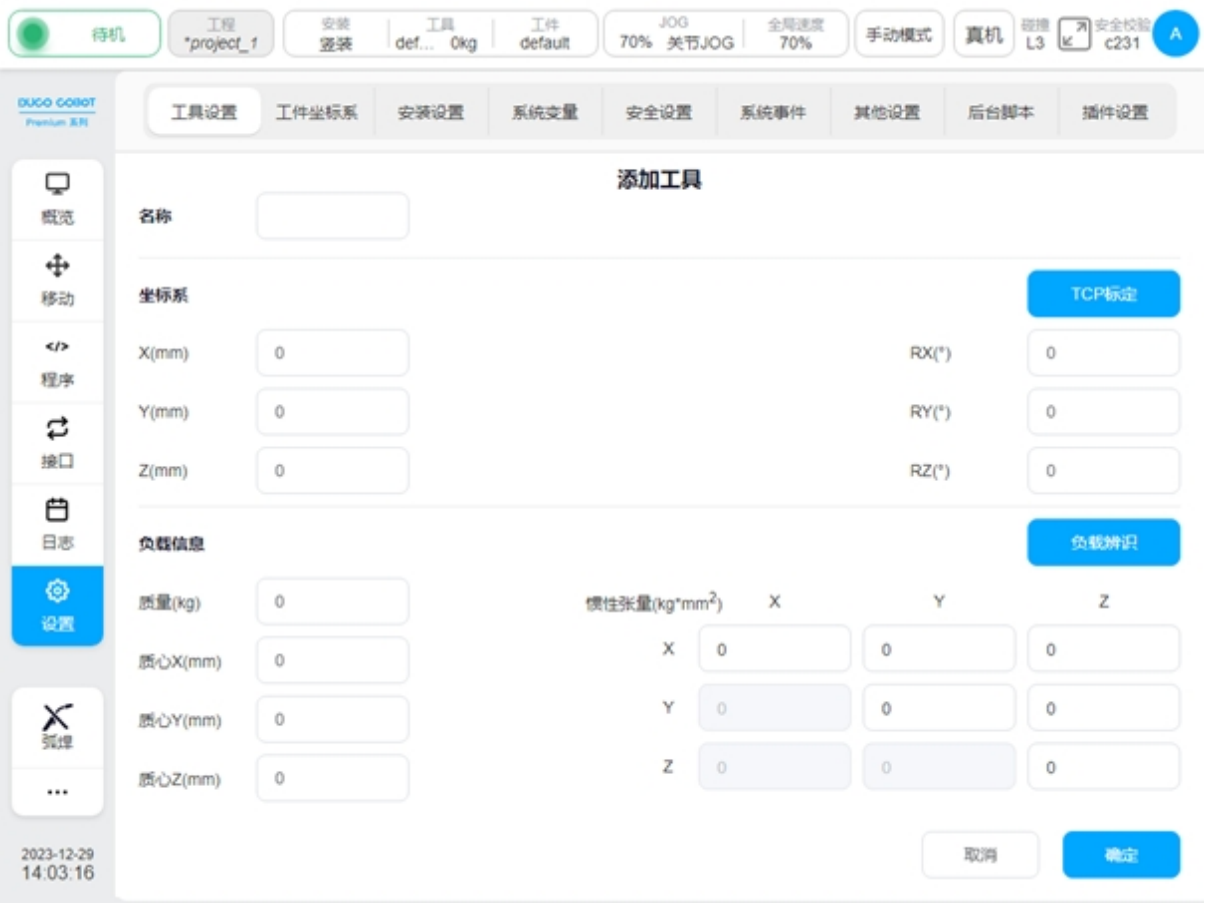


2.12.1 工具设置

工具设置子页面左侧是设置的工具坐标系相关信息，其中名称为 `default` 的工具坐标系是系统默认工具坐标系；右侧是机器人 3D 模型。

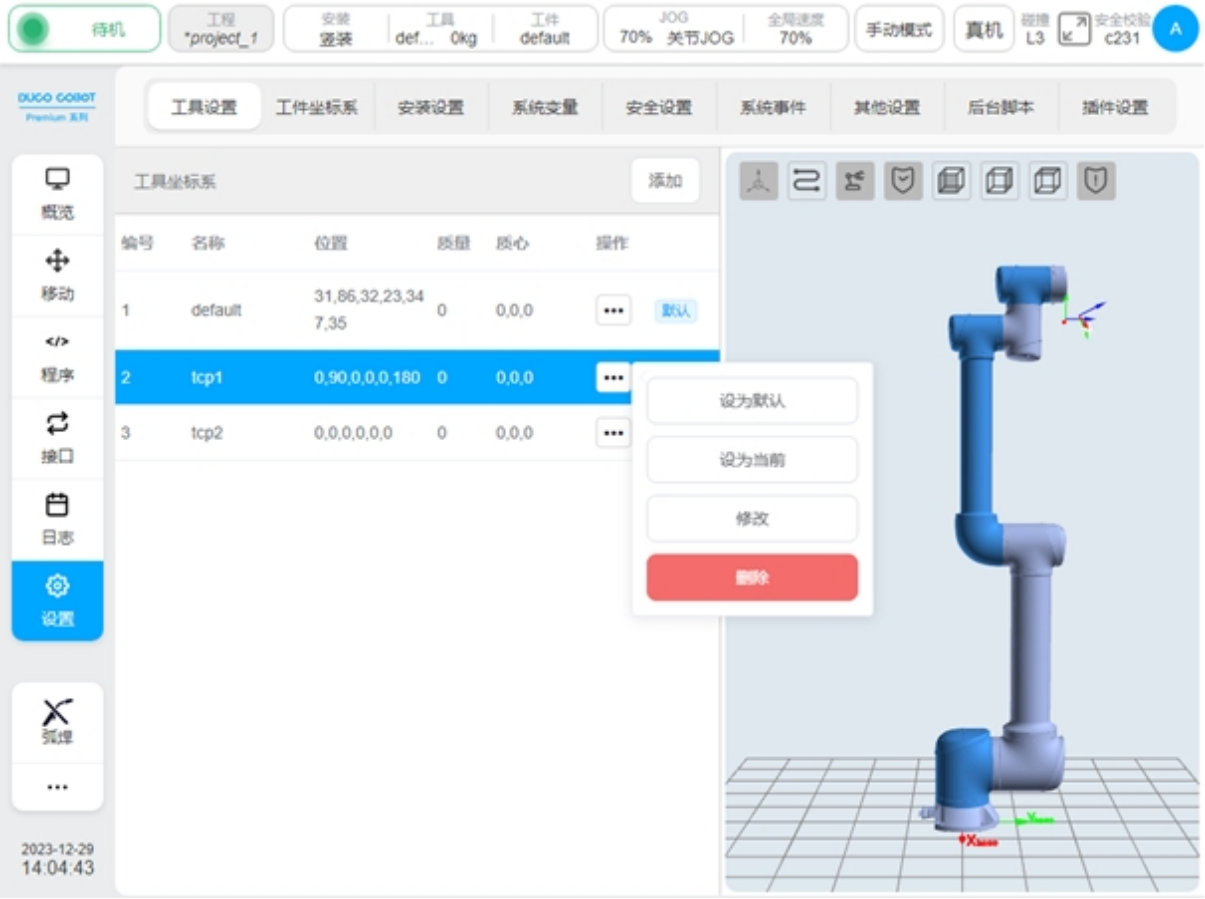


单击“添加”按钮，会弹出新建工具坐标系窗口。可设置工具坐标系名称，工具坐标系在机器人末端的位置 X、Y、Z 和对应的 X、Y、Z 轴正方向（RX、RY、RZ），同时可以设置工具的质量和质心位置，或者通过负载辨识功能识别质量和质心。关于负载辨识的功能介绍，详情请参看《DUCO CORE 末端力控功能操作手册》。可以设置工具的惯量参数，设置惯量矩阵对应的 xx、xy、xz、yy、yz、zz 元素。



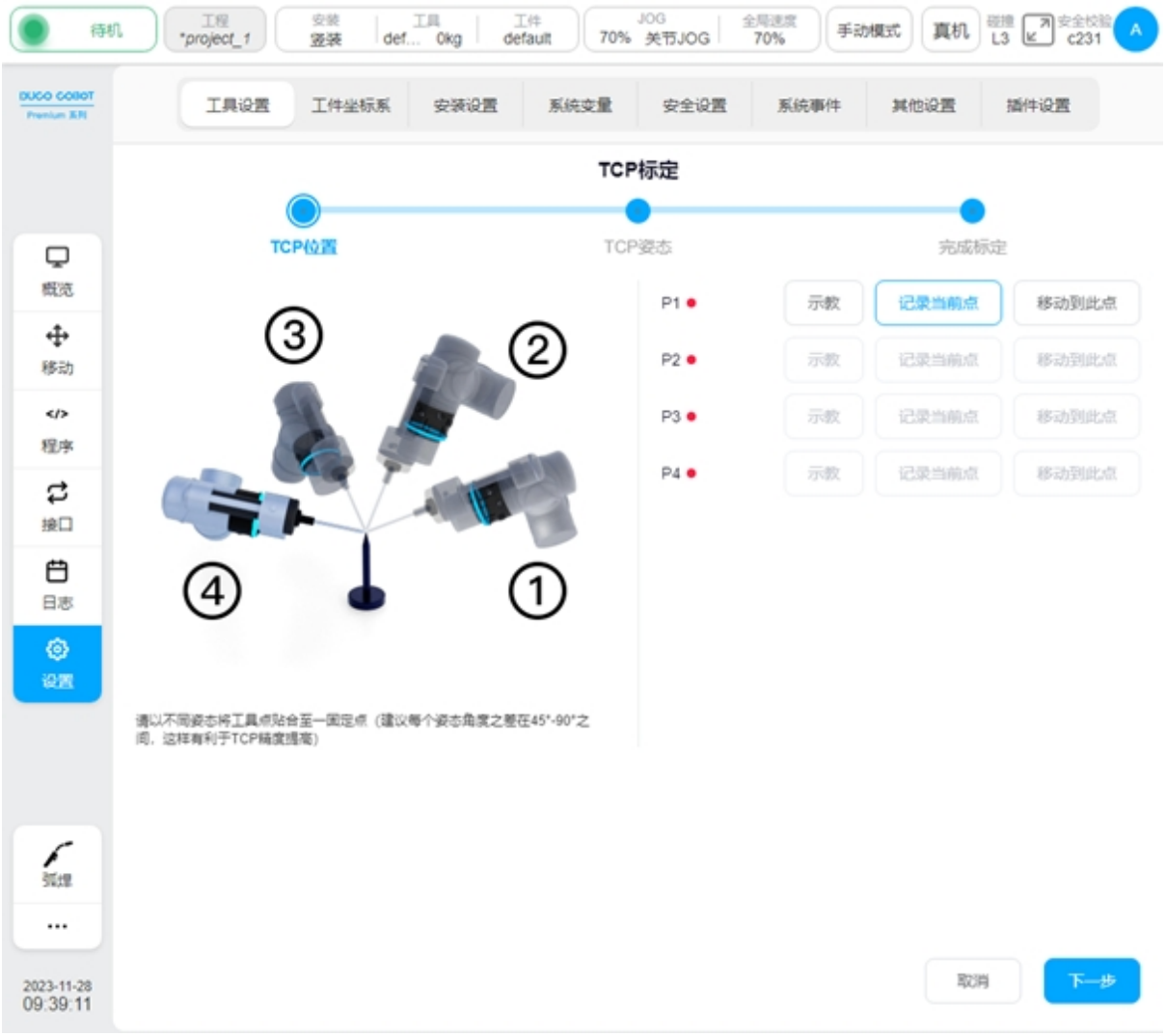
设定工具坐标系后，工具设置页面会显示出对应的工具坐标系，如图中工具坐标系 `tcp1`。可以保存多个工具坐标系的设置。

单击每行工具坐标系的操作列图标 ••• 会显示坐标系弹框。当设置对应工具坐标系为默认坐标系，图标 ••• 后会显示“默认”字样。当设置对应工具坐标系为当前坐标系，则状态栏中工具框里显示该坐标系名称和工具质量。单击“修改”按钮可对工具坐标系相关参数进行修改，也可单击“删除”按钮删除工具坐标系。注意：被设为默认或当前的工具坐标系是无法被删除的！

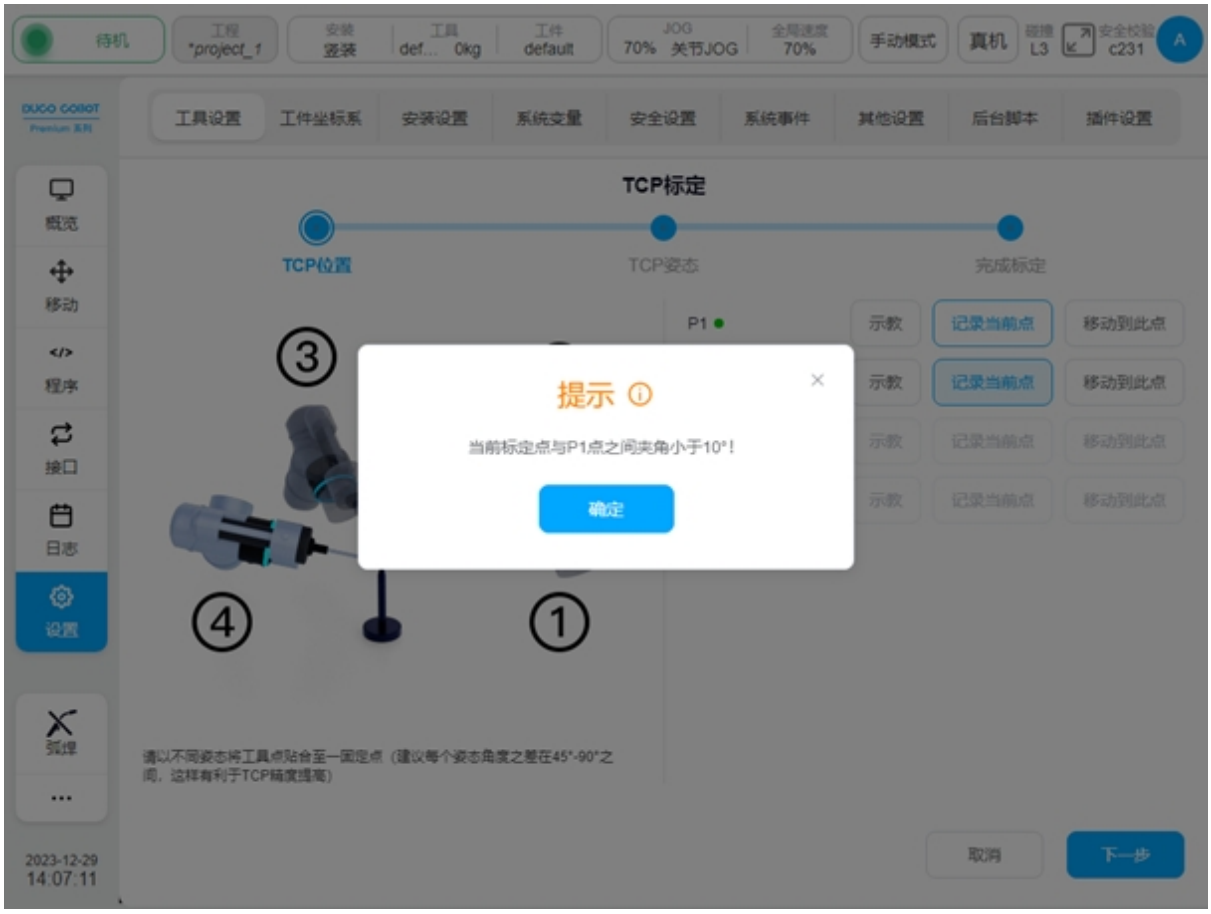


选择不同工具坐标系后，3D 模型区会相应显示该坐标系的位置。其中，红色表示 X 轴，绿色表示 Y 轴，蓝色表示 Z 轴。

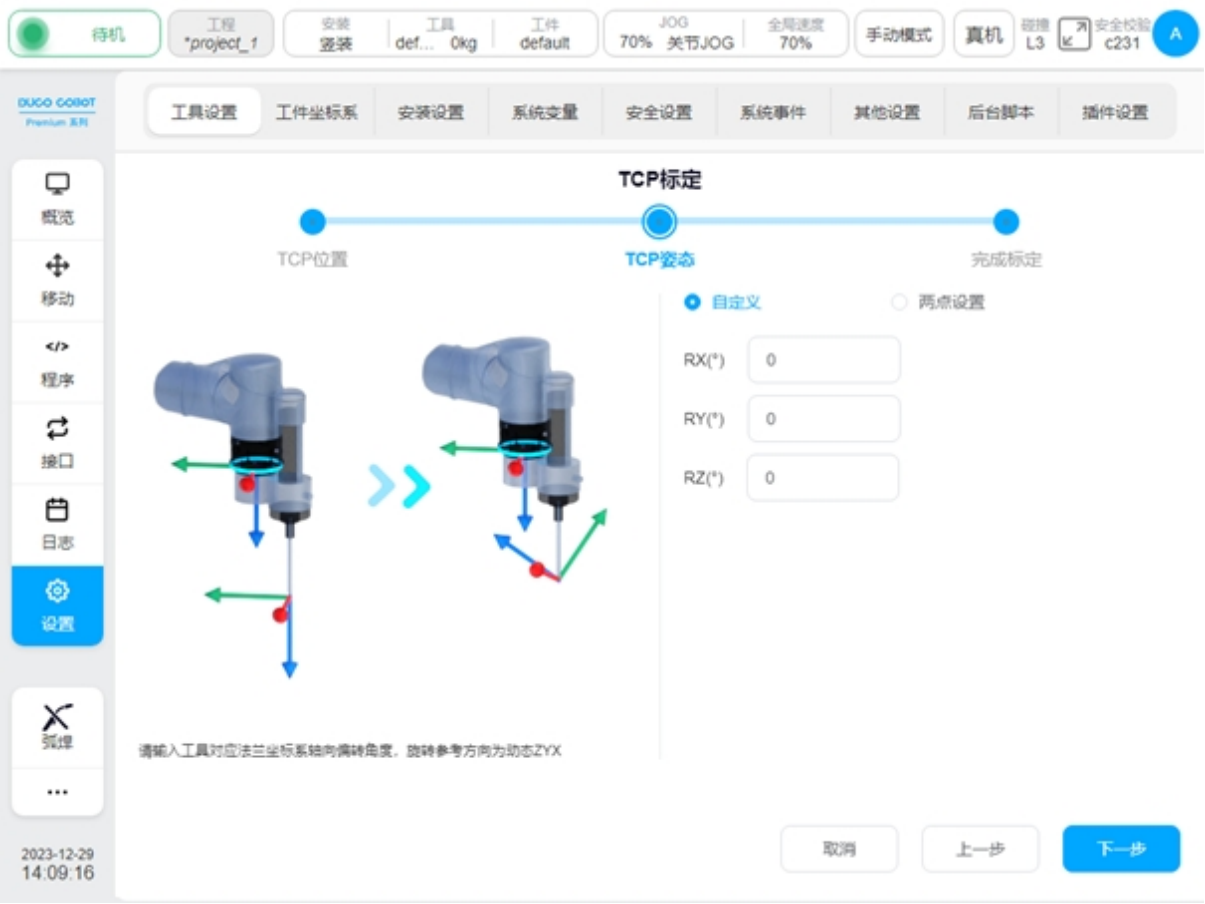
除了可手动设置工具坐标系以外，还可进行工具坐标的标定。工具坐标系的标定有两种方式：4 点标定（自定义）和 6 点标定（两点设置）。在添加工具窗口，单击“TCP 标定”按钮，进入 TCP 标定流程的 TCP 位置标定阶段显示如下：



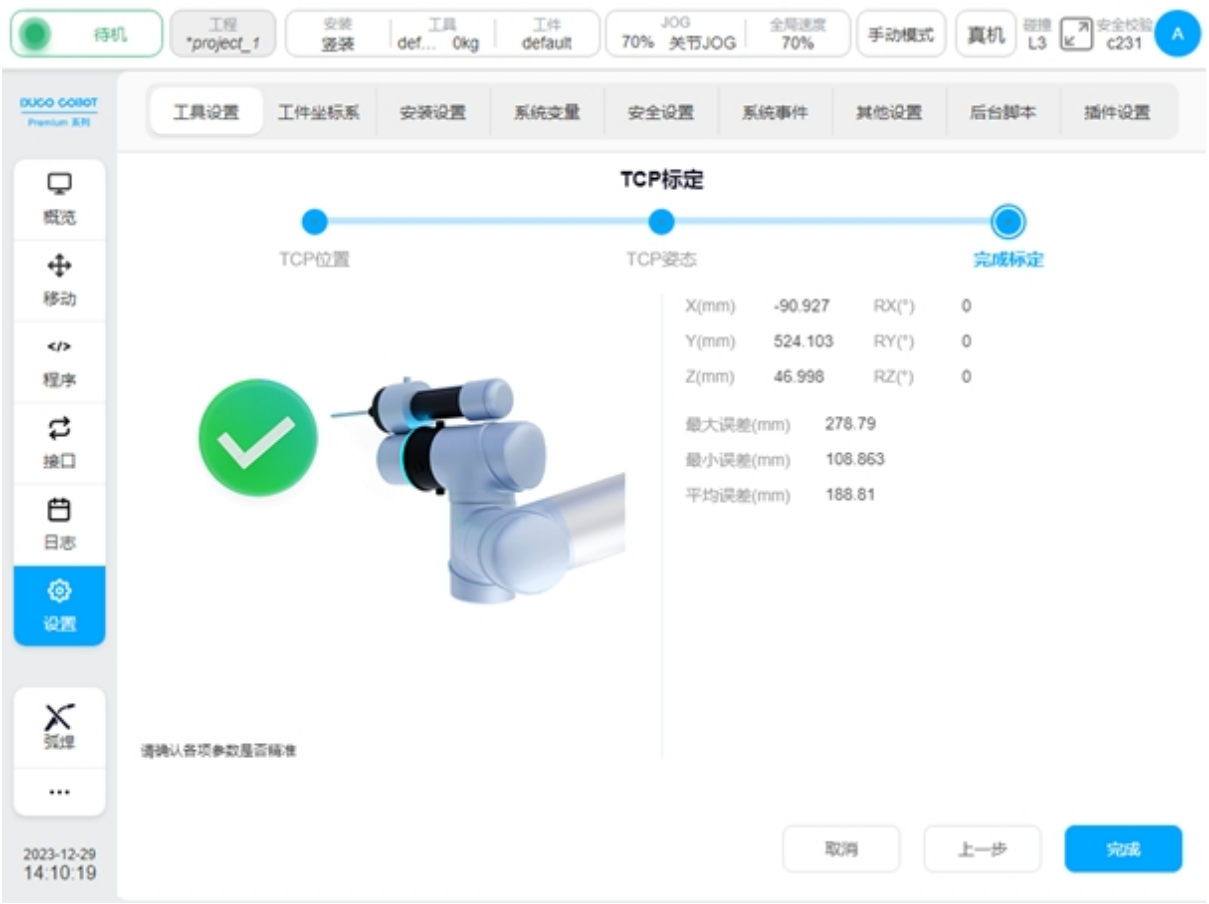
首先，需要以不同姿态将工具点贴合至标定针固定点，即示教好 P1-P4 点，且建议点与点之间的姿态角度变化在 45° 至 90° 之间，这样有利于提高 TCP 标定精度。以 P1 点为例，单击 P1 点处的“示教”按钮，界面会跳转到移动页面，移动机器人到目标点后，单击“记录当前位置”，界面会跳回当前示教标定点界面，且示教好一个标定点后，对应标定点后的红色圆点会变成绿色。也可通过示教器物理按钮直接移动机器人到目标点位后，直接单击“记录当前点”。示教过程中，如果当前示教点位不满足限制条件会弹框提示如下：



依次示教好 P1-P4 全部点位后，单击页面右下角“下一步”按钮，进入 TCP 姿态标定阶段。TCP 姿态标定有两种形式：自定义和两点设置，默认为自定义，如下图：



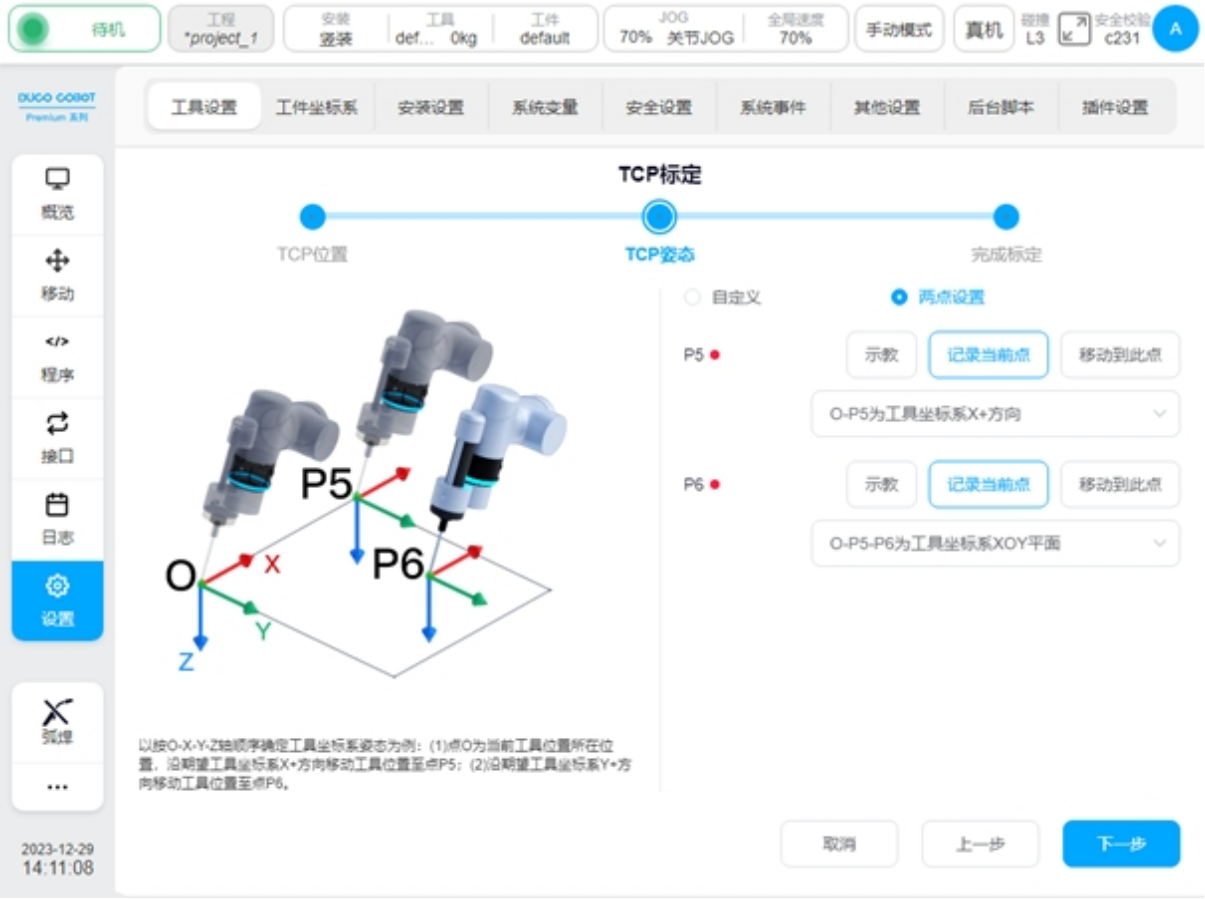
设置好工具对应法兰坐标系轴向偏转角度后，单击“下一步”按钮，进入完成标定阶段如下图，会显示标定结果和误差信息。用户可通过误差信息来判断该次标定质量，从而决定是否需要加载此坐标系。如果需要加载此坐标系，单击“确认”按钮即可，否则单击“取消”按钮，或者单击“上一步”返回进行重新示教点位。



备注： 标定结果的误差值，真实含义是计算得到的 TCP 位置与 4 个标定点位之间距离的标准差。该数据可作为标定结果的参考，但不能作为标定精度的绝对依据，实际精度以应用效果为准。

TCP 的标定结果，通常以平均误差小于 2mm 为准。

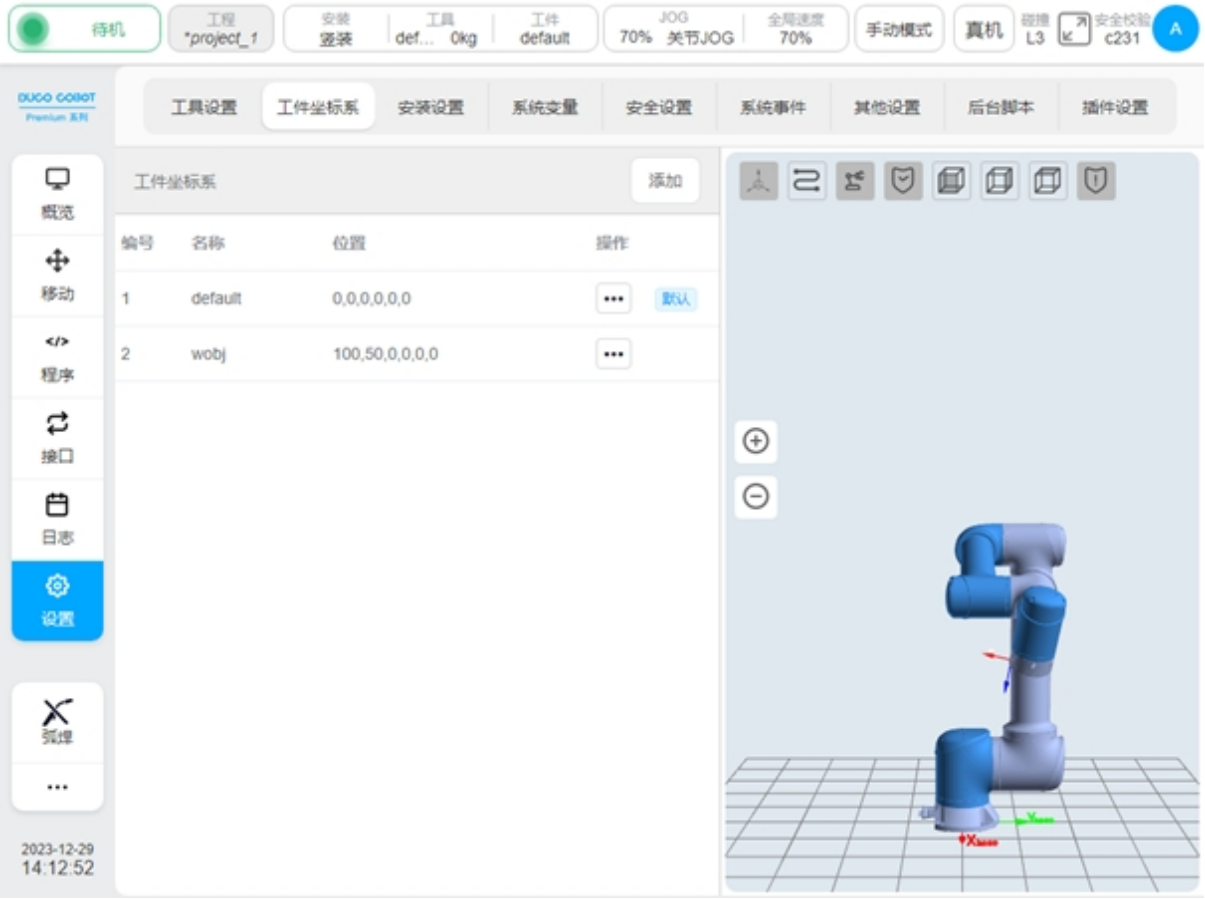
当 TCP 姿态标定阶段选择“两点设置”时，显示如下图：P5 点可选项有 O-P5 为工具坐标系 X+ 方向 / Y+ 方向 / Z+ 方向，P6 点对应可选项有 O-P5-P6 为工具坐标系 XOY 平面/XOZ 平面、O-P5-P6 为工具坐标系 XOY 平面/YOZ 平面、O-P5-P6 为工具坐标系 XOZ 平面/YOZ 平面。以按 O-X-Y-Z 轴顺序确定工具坐标系姿态为例：点 O 为当前工具位置所在位置，首先沿期望工具坐标系 X+ 方向移动工具位置至点 P5，接着沿期望工具坐标系 Y+ 方向移动工具位置至点 P6。



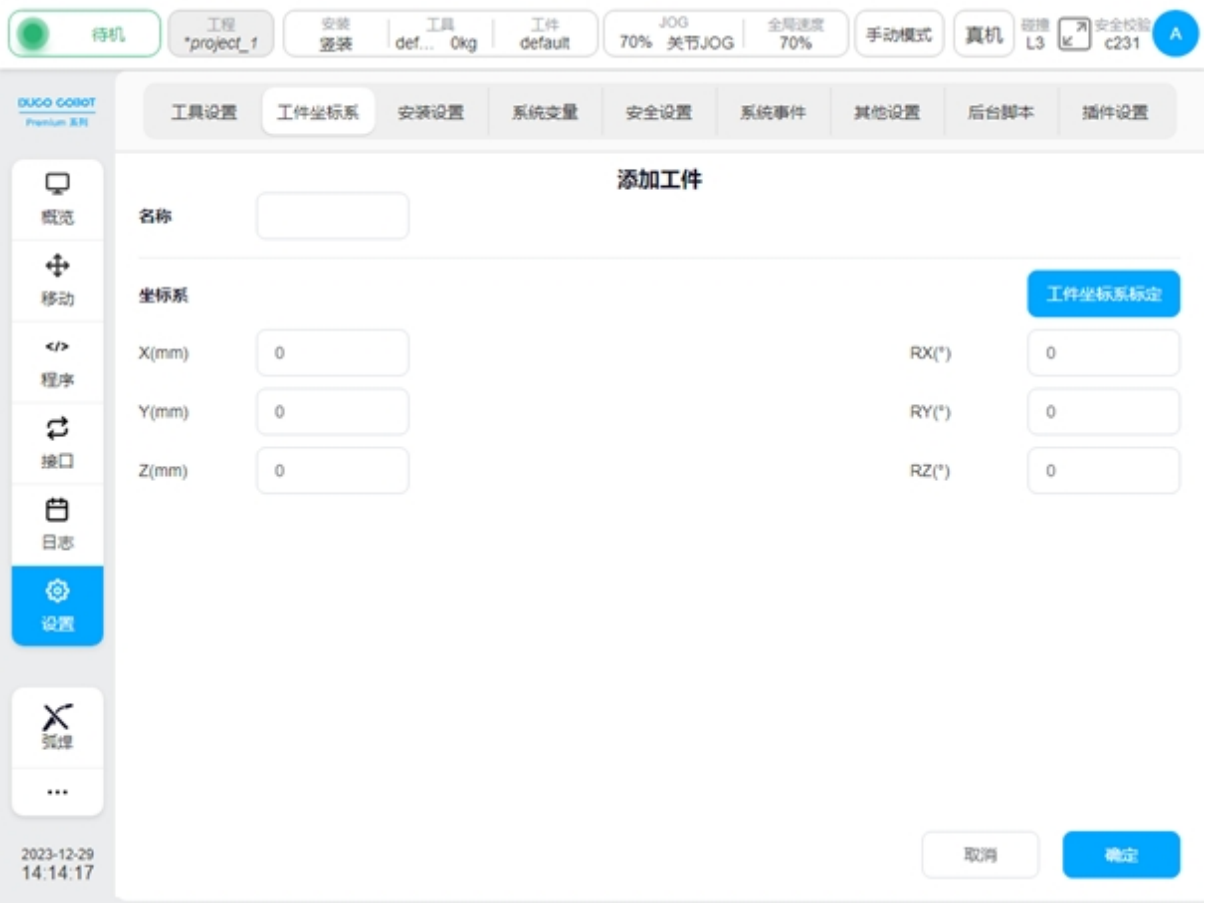
设置示教好 P5、P6 后，单击“下一步”按钮，同样地进入完成标定阶段，会显示标定结果和误差信息。用户可通过误差信息来判断该次标定质量，从而决定是否需要加载此坐标系。如果需要加载此坐标系，单击“确认”按钮即可，否则单击“取消”按钮，或者单击“上一步”返回进行重新示教点位。

2.12.2 工件坐标系

工件坐标系子页面中可设置工件坐标相对于世界坐标系的偏移量，与工具设置页面类似，页面左侧是设置的工件坐标系相关信息，其中名称为 default 的工件坐标系是系统默认工件坐标系；右侧是机器人 3D 模型。



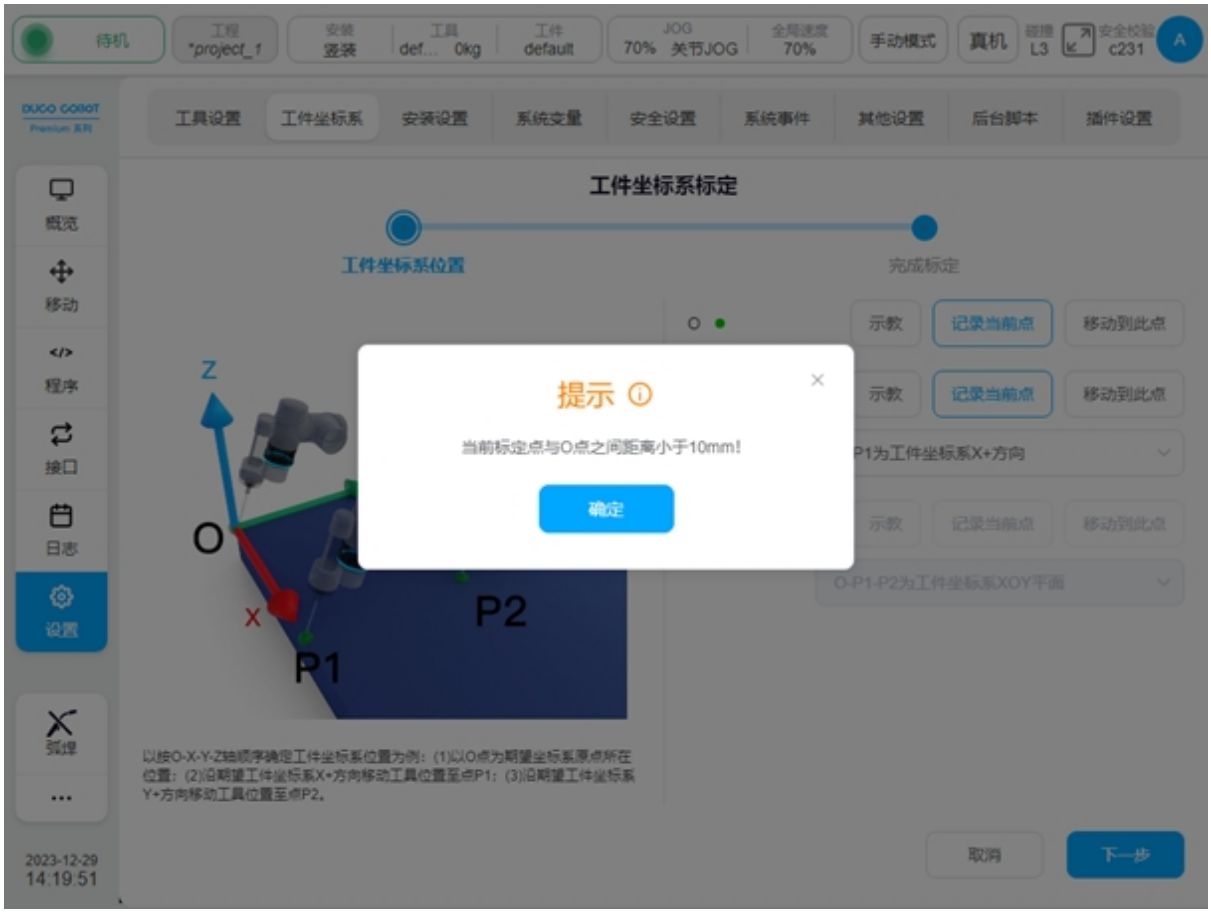
单击“添加”按钮，会弹出新建工件坐标系窗口。可设置工件坐标系名称，工件坐标系相对于世界坐标系的位置 X、Y、Z 和对应的 X、Y、Z 轴正方向（RX、RY、RZ）。设定工件坐标系后，工件设置页面会显示出对应的工件坐标系。可以保存多个工件坐标系的设置。



同样地，除了可手动设置工件坐标系以外，还可进行工件坐标的标定。单击“工件坐标系标定”按钮，进入工件坐标系标定流程的工件坐标系位置标定阶段显示如下：

设置 O 点为工件坐标系的原点，P1 点可选项有 O-P1 为工具坐标系 X+ 方向 / Y+ 方向 / Z+ 方向，P2 点对应可选项有 O-P1-P2 为工具坐标系 XOY 平面/XOZ 平面、O-P1-P2 为工具坐标系 XOY 平面/YOZ 平面、O-P1-P2 为工具坐标系 XOZ 平面/YOZ 平面。以按 O-X-Y-Z 轴顺序确定工具坐标系姿态为例：点 O 为当前工具位置所在位置，首先沿期望工件坐标系 X+ 方向移动工具位置至点 P1，接着沿期望工件坐标系 Y+ 方向移动工具位置至点 P2。

以 P1 点为例，单击 P1 点后的“示教”按钮，界面会跳转到移动页面，移动机器人到目标点后，单击“记录当前位置”，界面会跳回当前示教标定点界面，且示教好一个标定点后，对应标定点后的红色圆点会变成绿色。也可通过示教器物理按键直接移动机器人到目标点位后，直接单击“记录当前点”。示教过程中，如果当前示教点位不满足限制条件会弹框提示如下：



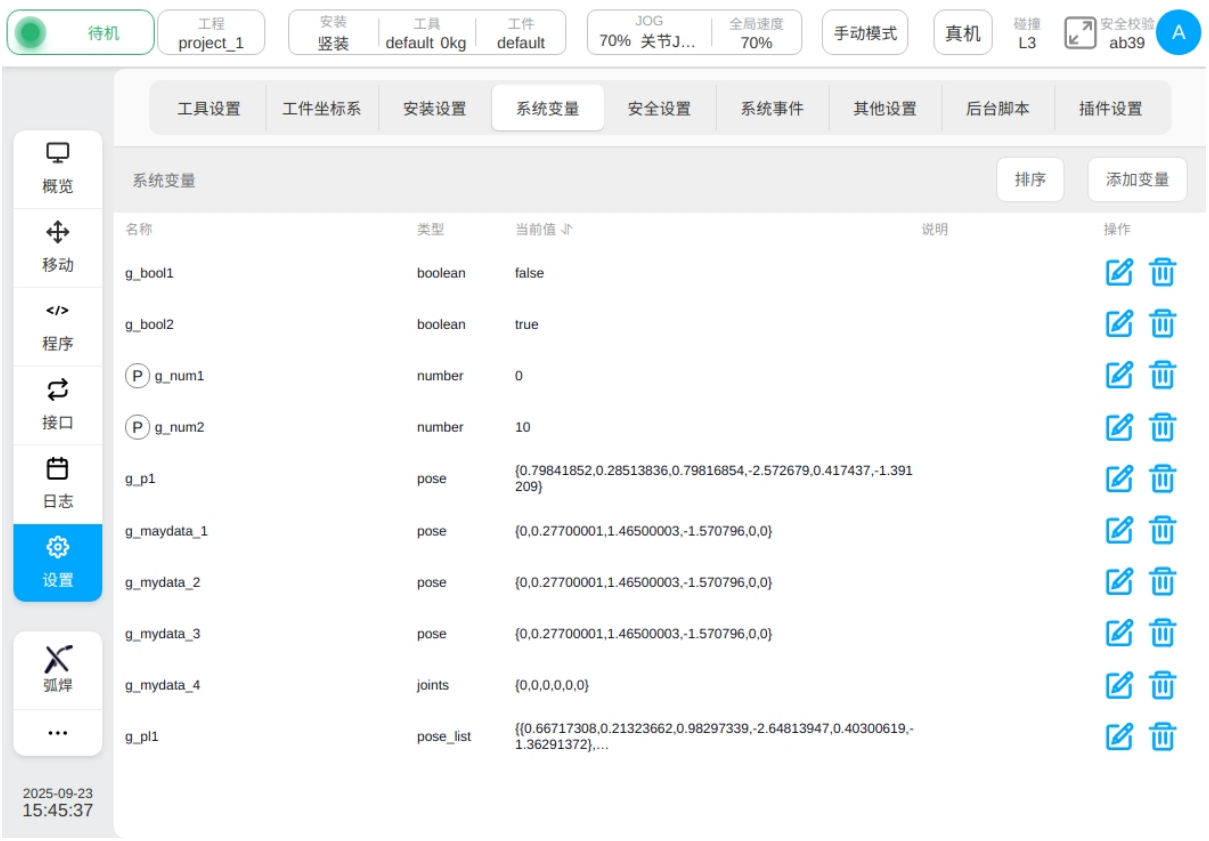
设置示教好 O、P1、P2 后，单击“下一步”按钮，同样地进入完成标定阶段如下图，会显示标定结果。用户可通过误差信息来判断该次标定质量，从而决定是否需要加载此坐标系。如果需要加载此坐标系，单击“确认”按钮即可，否则单击“取消”按钮，或者单击“上一步”返回进行重新示教点位。

2.12.3 安装设置

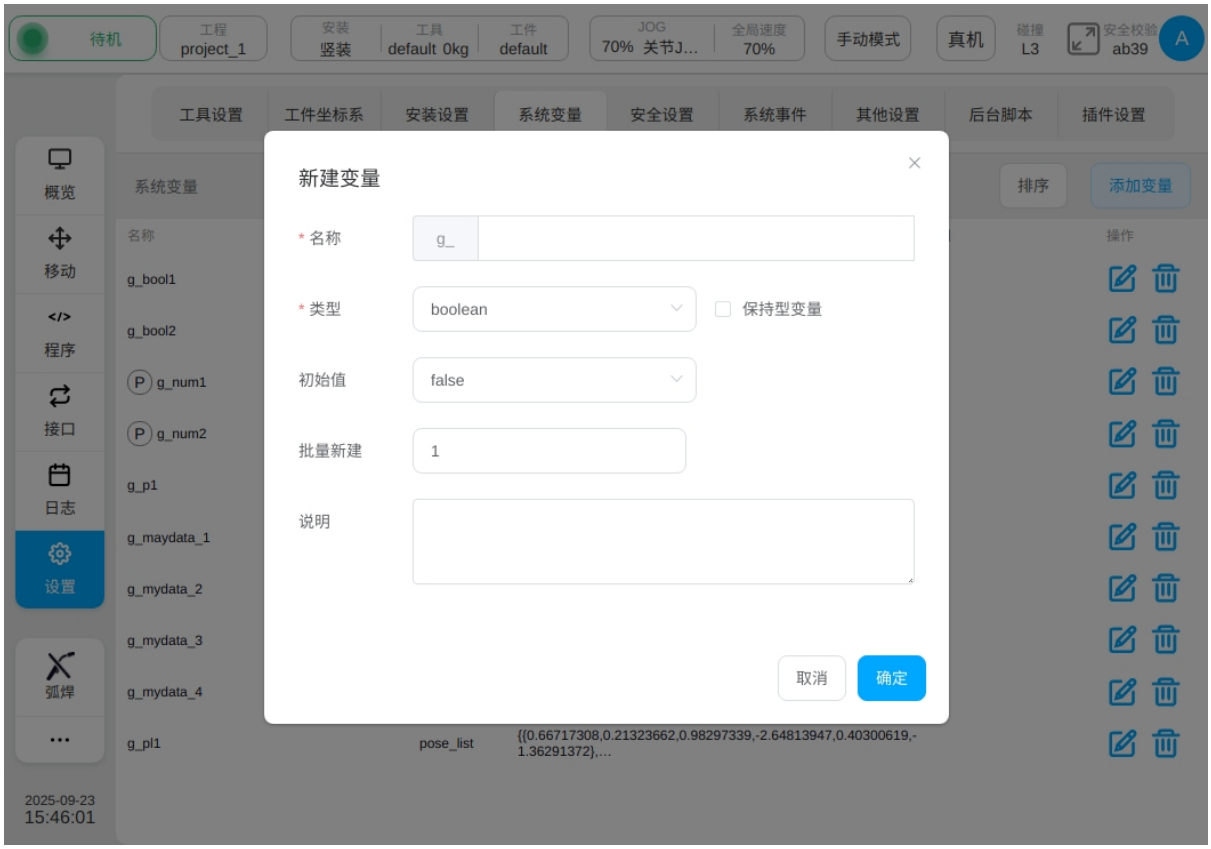
安装设置子页面主要是设置机器人的安装方向，可设置成竖装、横装、倒装和自定义中任意一种。其中，自定义方式是用户自己设置机器人绕基座倾斜角度和机器人绕基座旋转角度，设置范围是-180°~180°。切换安装方式同时 3D 模型也会跟着改变安装方向。设置好机器人安装方向后，单击界面右上方“设置”按钮即可，且状态栏中安装框里对应显示该安装方式。

2.12.4 系统变量

系统变量子页面主要是创建和显示系统变量相关信息。显示系统变量的信息有: 名称、类型、初始值、当前值、变量的说明和对每个变量进行修改或删除的按钮。其中, 变量类型有 15 种, 分别是 `boolean`、`number`、`string`、`num_list`、`pose`、`joints`、`pose_list`、`joints_list`、`timer`、`pose_speed`、`pose_acc`、`joint_speed`、`joint_acc`、`boolean_list`、`string_list`。初始值和当前值分别表示变量定义时的初值和程序应用过程中的过程值。



单击界面左上方“添加变量”按钮，会弹出新建变量的窗口。输入变量名称、类型、初始值、批量新建和说明，单击“确定”按钮即可。每个变量名称前都会有 g_ 的标记，代表是全局变量，且每个变量类型都可选择是否为“保持型变量”。当新建的变量勾选了变量类型后的“保持型变量”单选框，则页面上变量名称前会显示 P 字样。设置为保持性变量后，系统每次均会保存上次的值；否则会按照初始值初始化变量。



布尔类型 boolean 初始值有 false 和 true 两种可选值;

数字类型 number 初始值为 0，变量值支持整数和小数;

字符串类型 string 初始值为空;

数组类型 num_list 是以花括号包裹的整型数组或浮点型数组;

位姿 pose 类型是一个长度为 6 以花括号包裹的浮点型数组;

关节 joints 类型是一个长度为 7 的浮点数组;

位姿列表 pose_list 类型是以花括号包裹着若干以花括号包裹的长度为 6 的浮点型数组的二维数组;

关节列表 joints_list 类型是以花括号包裹着若干以花括号包裹的长度为 7 的浮点数组的二维数组;

计时类型 timer 是以花括号包裹的整型数组或浮点型数组;

末端速度 pose_speed 类型、末端加速度 pose_acc 类型、关节角速度 joint_speed 类型、关节角加速度 joint_acc 类型变量本质上也是 number 类型，支持整数和小数，且每种变量类型同时存在两种单位显示，可以各自输入一种单位变量数值后自动填充另一种单位换算数值。

布尔列表 boolean_list 类型是以花括号包裹的布尔类型数组;

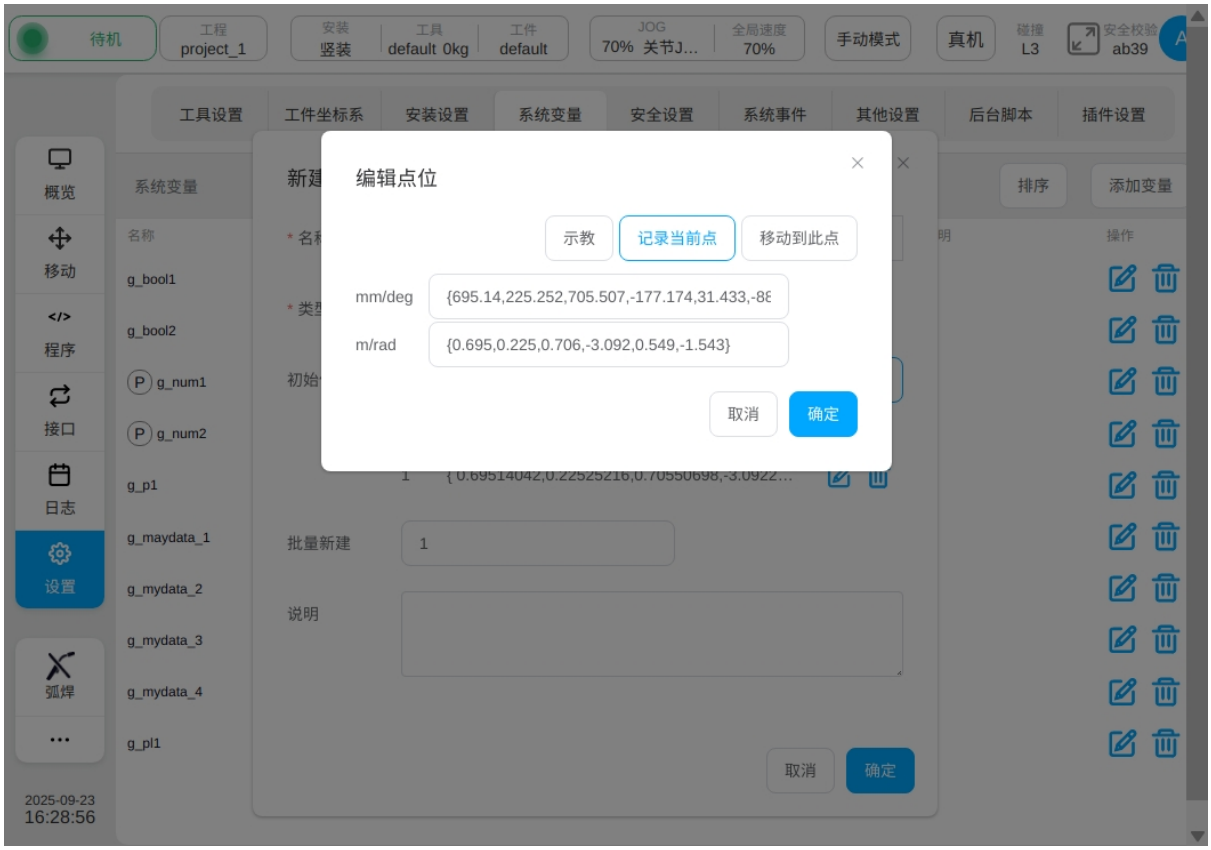
字符串列表 string_list 类型是以花括号包裹的字符串类型数组;

所有变量类型都可手动赋值，其中 pose 和 joints 两种类型变量可通过示教方式进行初始化和编辑，pose_list 和 joints_list 两种类型变量可通过示教添加和记录当前点两种方式进行添加若干个位姿或关节点位。

单击“示教”按钮，界面会跳转到移动界面，示教好所需位姿后，单击“记录当前位置”，界面会回到新建变量窗口，且将示教位姿值记录到对应初始值位置，对应有 mm/deg 和 m/rad 两种单位下的显示值。

除了示教初始化外，用户也可单击初始值输入框，进行手动初始化赋值。joints 类型变量创建过程与 pose 类型类似，此处不累述。

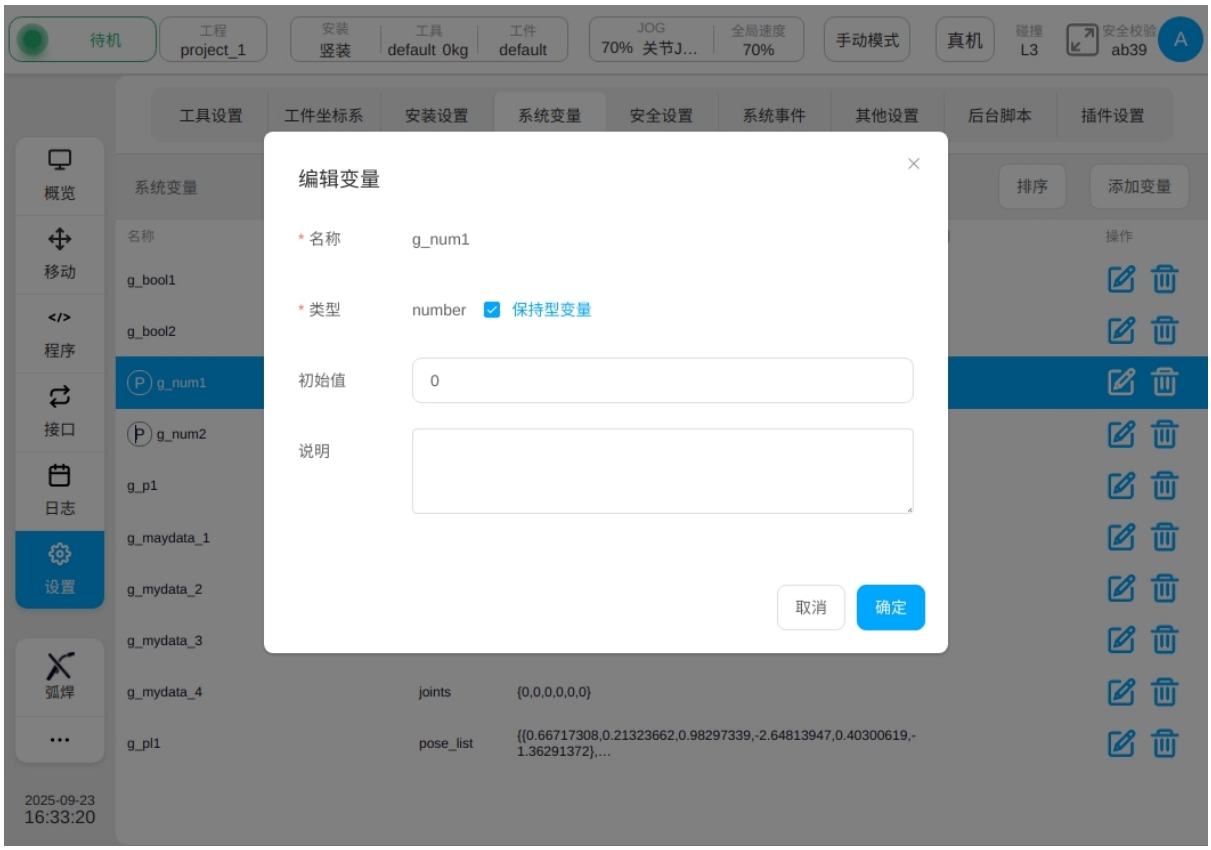
新建 pose_list 类型变量时，初始值可以通过“示教添加”或“记录当前点”任意方式进行添加，点击“记录当前点”按钮，会在界面下方显示对应变量序号、值、操作，点击操作列的  图标，会弹出编辑点位的窗口，操作过程与 pose 类型变量示教过程类似，此处不作累述。



选取好位姿点位后，界面如下：

`joints_list` 类型变量创建过程与 `pose_list` 类型类似，此处不累述。

当修改变量时，单击变量行对应的 图标，会弹出编辑变量的窗口。变量类型是不可修改的，只可修改初始值、是否是保持型变量以及变量说明。删除变量，单击变量行对应的 图标即可。

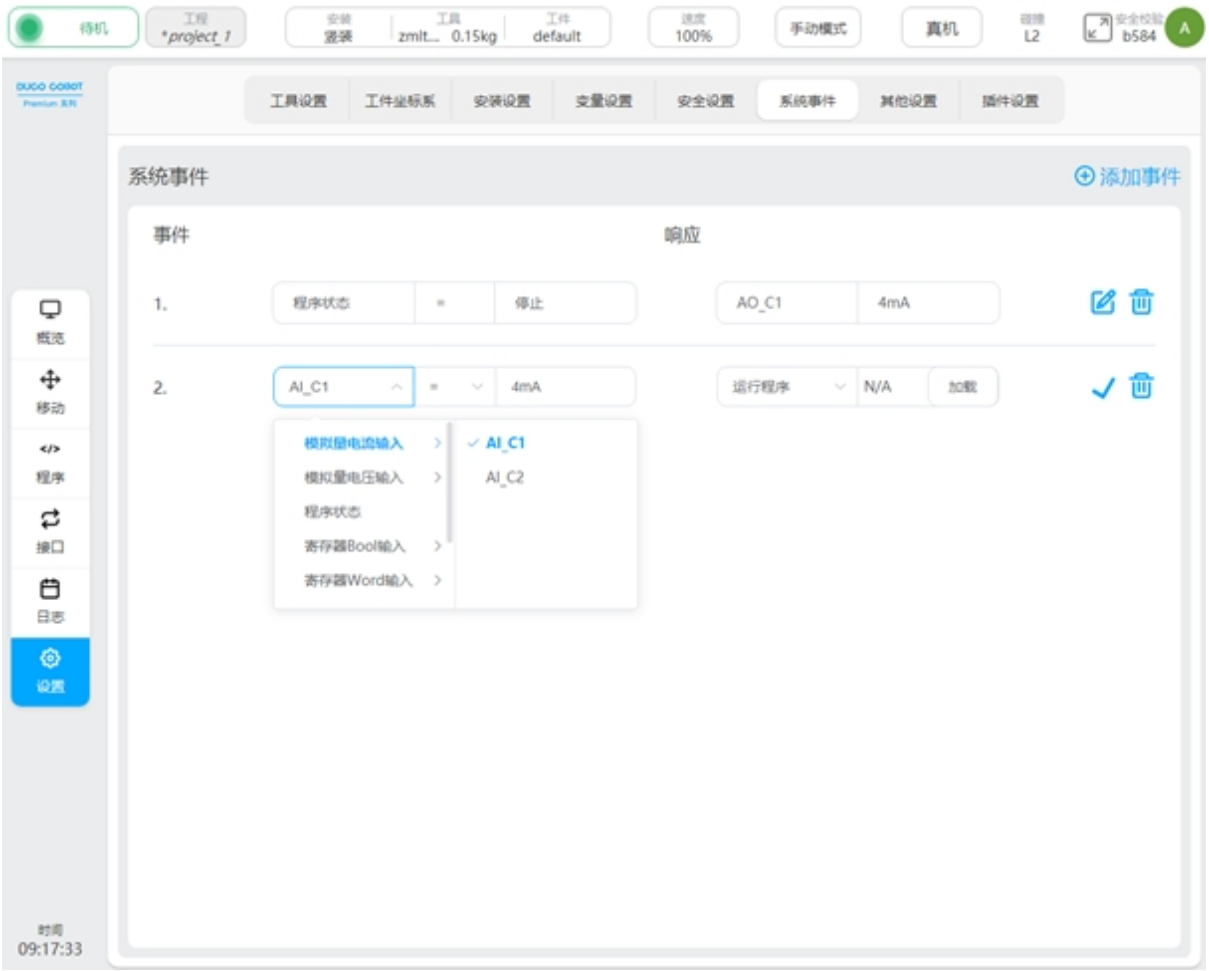


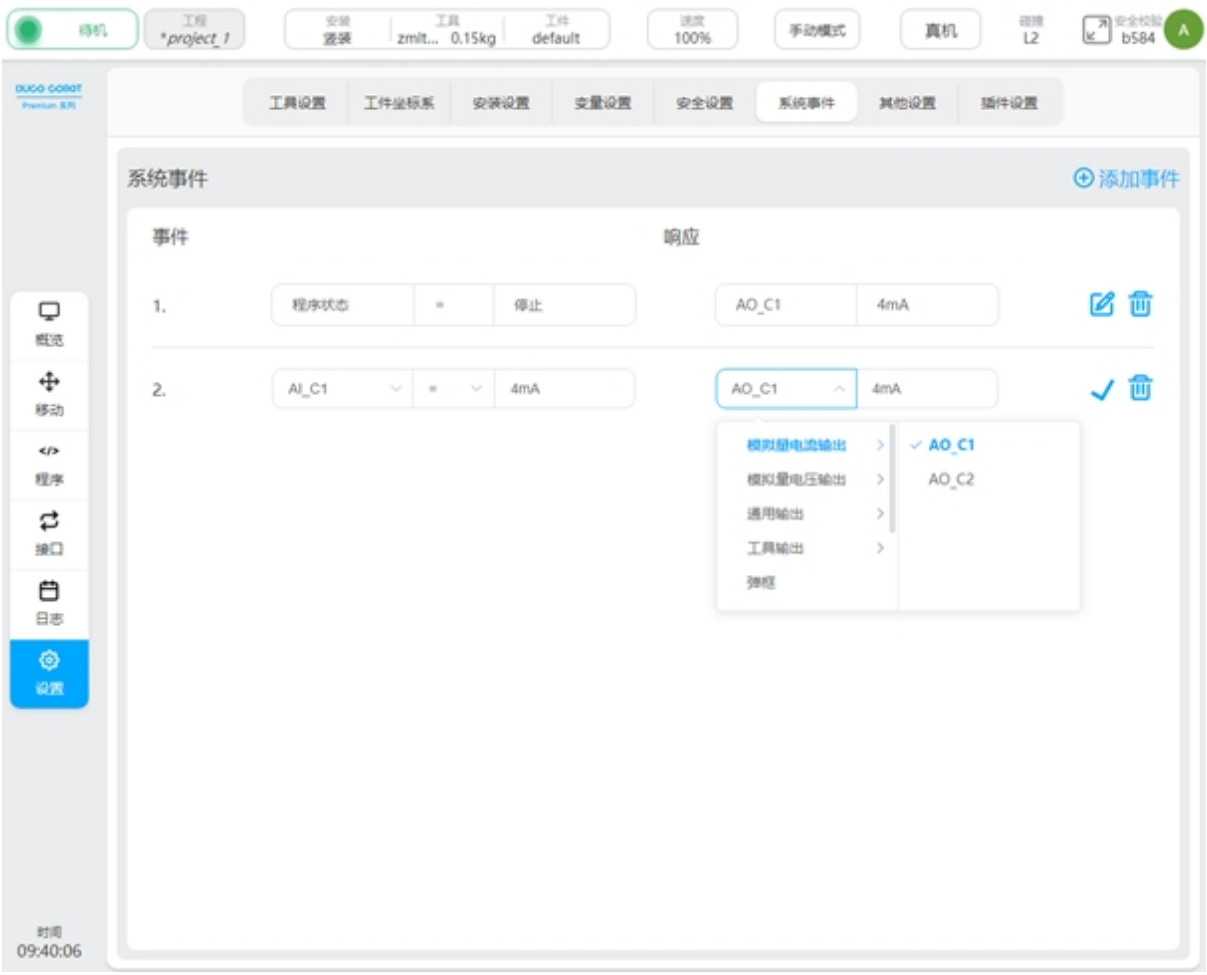
2.12.5 系统事件

系统事件子页面主要是为系统添加额外的通用事件功能，在发生某些事件时，用户可以自定义事件的响应。目前，事件类型有：模拟量电流输入、模拟量电压输入、程序状态、寄存器 Bool 输入、寄存器 Word 输入、寄存器 Float 输入、末端速度、关节扭矩和自定义事件共 9 种事件；响应类型有：模拟量电流输出、模拟量电压输出、通用输出、工具输出、弹框、运行程序、寄存器 Bool 输出、寄存器 Word 输出和寄存器 Float 输出共 9 种。

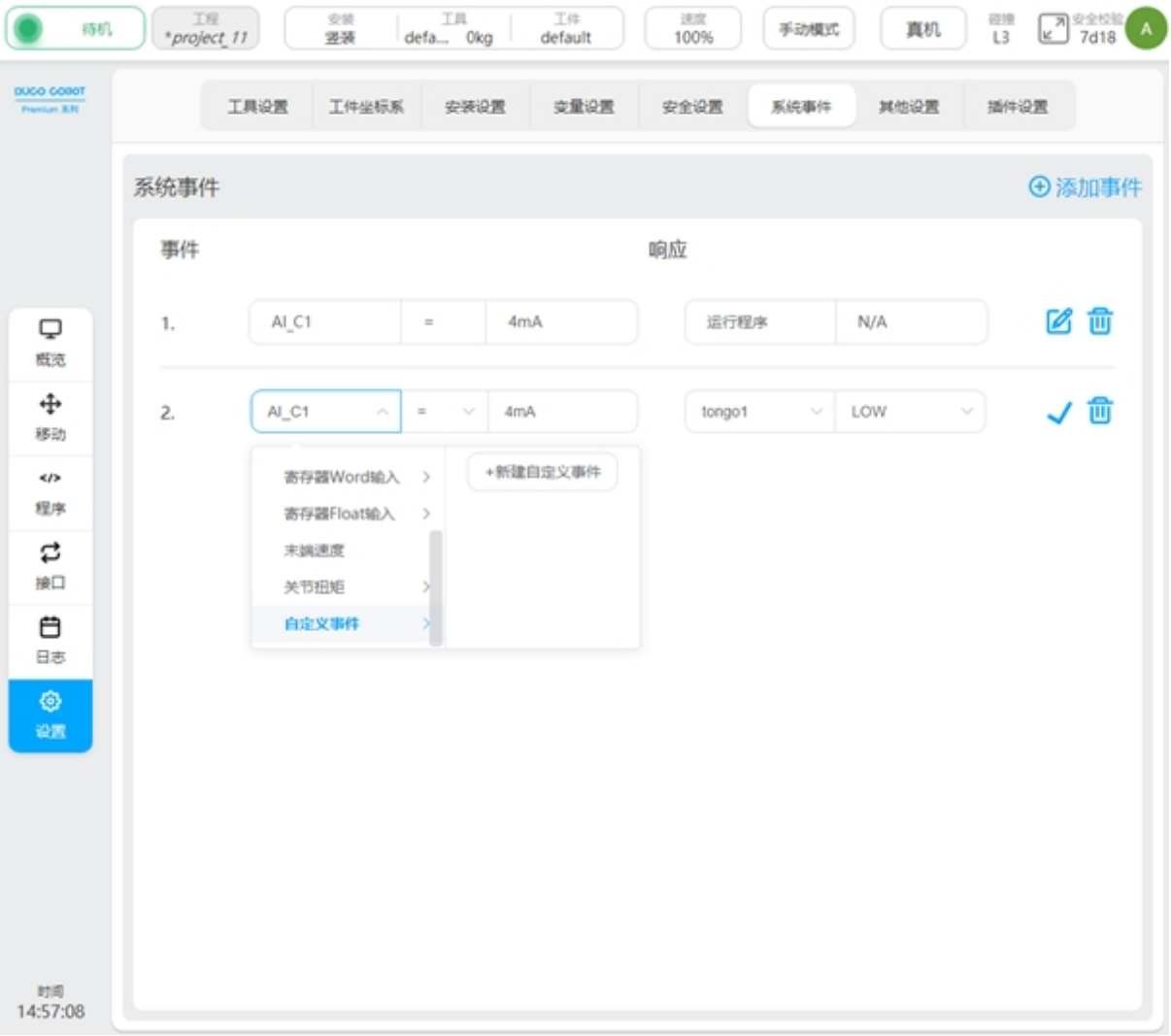
备注： 事件触发条件均为沿触发。

若事件触发条件设置时已经满足，在设置后不会立刻触发，需等待下次沿触发。


待机
工程
*project_1
安装
竖装
工具
zmlt... 0.15kg
工件
default
速度
100%
手动模式
真机
L2
b584
工具设置
工件坐标系
安装设置
变量设置
安全设置
系统事件
其他设置
插件设置
系统事件
添加事件
事件
响应
1.
程序状态
=
停止
AO_C1
4mA
2.
AI_C1
=
4mA
运行程序
N/A
加载
模拟量电流输入
模拟量电压输入
程序状态
寄存器Bool输入
寄存器Word输入
AI_C1
AI_C2
概览
移动
程序
接口
日志
设置
时间
09:17:33

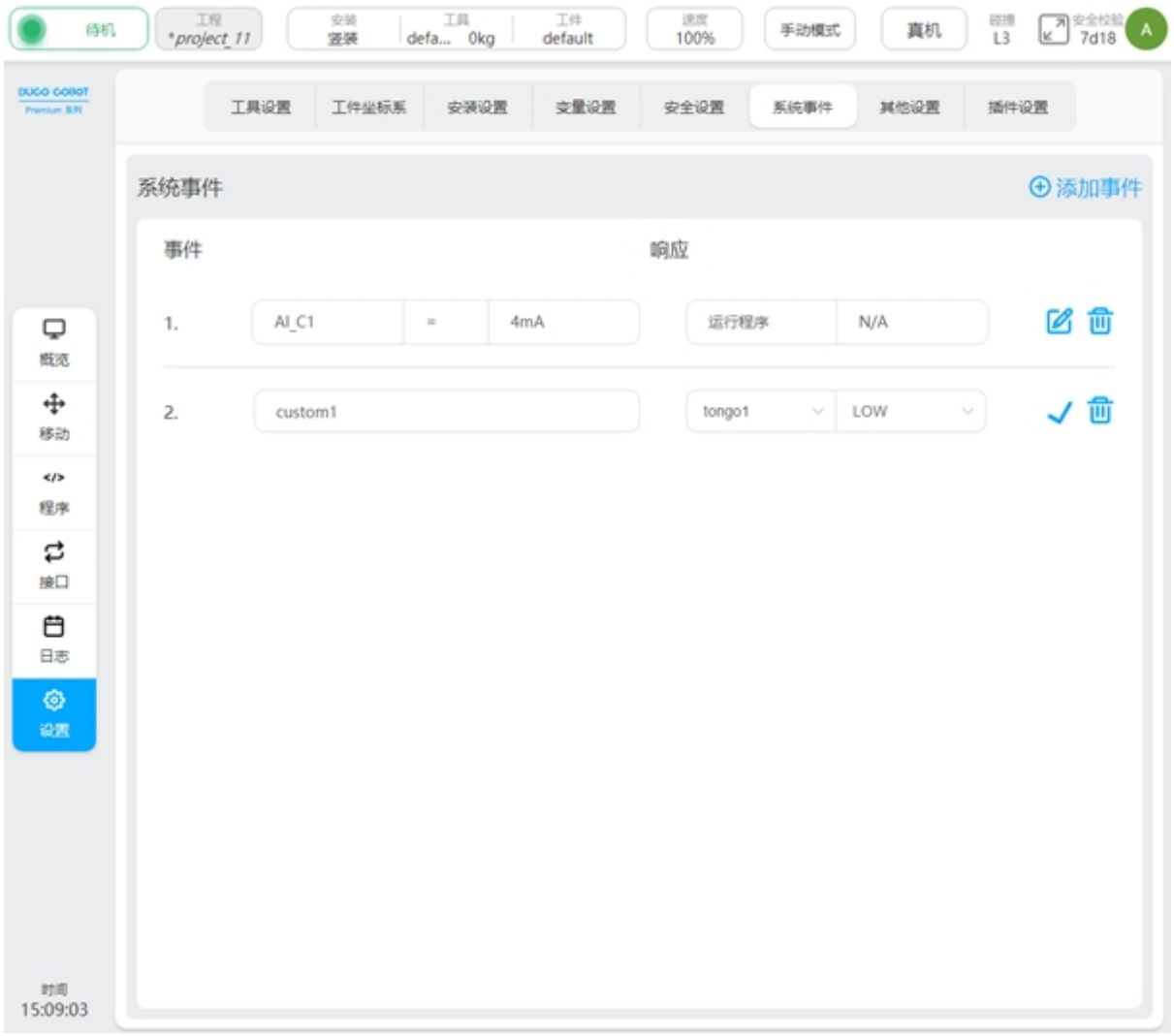


单击页面右上角“添加事件”，新增一条事件响应，新增事件类型默认为模拟量电流输入，事件条件是“=”，值为 4mA，响应类型默认为模拟量电流输出，值为 4mA。单击每条事件响应后的 图标，进入编辑状态，即可对每条事件响应进行编辑，编辑好后单击 图标即可。单击 图标可以删除添加的事件响应。



对于自定义事件，当自定义事件选项右侧列表没有可选项时，可以通过单击自定义事件右侧列表中“新建自定义事件”按钮创建自定义事件。单击“新建自定义事件”按钮，弹出虚拟键盘，提示用户输入自定义事件名。

用户输入自定义事件名（如：custom1）后，界面新增一条事件响应，如图所示：



新建的自定义事件会在自定义事件选项列表显示，同一自定义事件可以对应多个响应。且，当所有包含同一自定义事件的事件响应被删除后，该自定义事件也会被删除，即自定义事件选项列表无该自定义事件可选。

2.12.6 其他设置

其他设置子页面主要是启动设置、编辑 HOME 点、碰撞设置（对于 Safety2 该设置移入安全设置中)、力传感器设置、末端设置、模型导入设置、干涉区设置、手自动模式设置、奇异区间设置。界面左侧是导航选项卡，右侧是对应选项卡显示内容区。页面默认显示启动设置内容。

启动设置

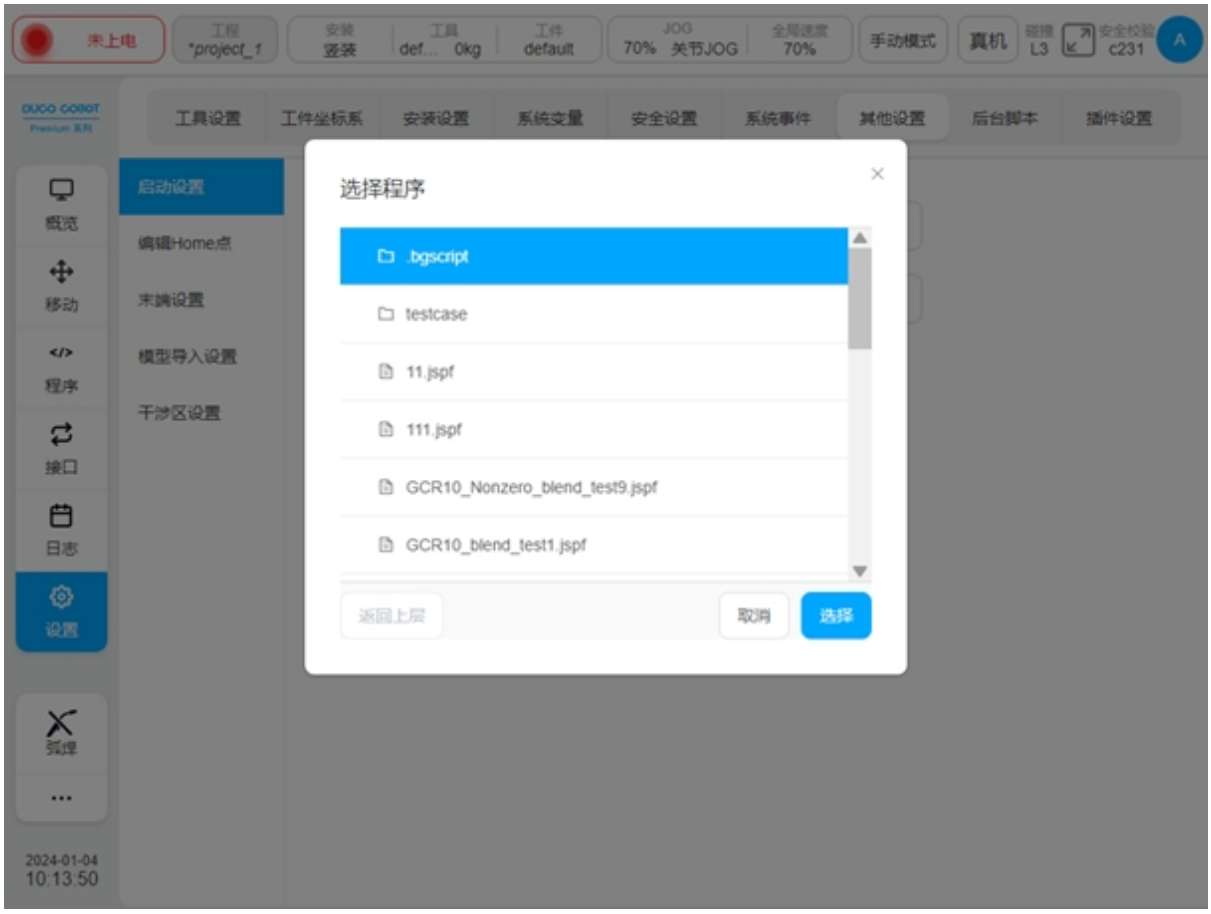
单击“启动设置”选项卡，可设置默认全局速度、默认 jog 速度、是否自动登录和设定默认程序。

调节默认全局速度右侧滑块，可以设置机器人默认的全局速度；

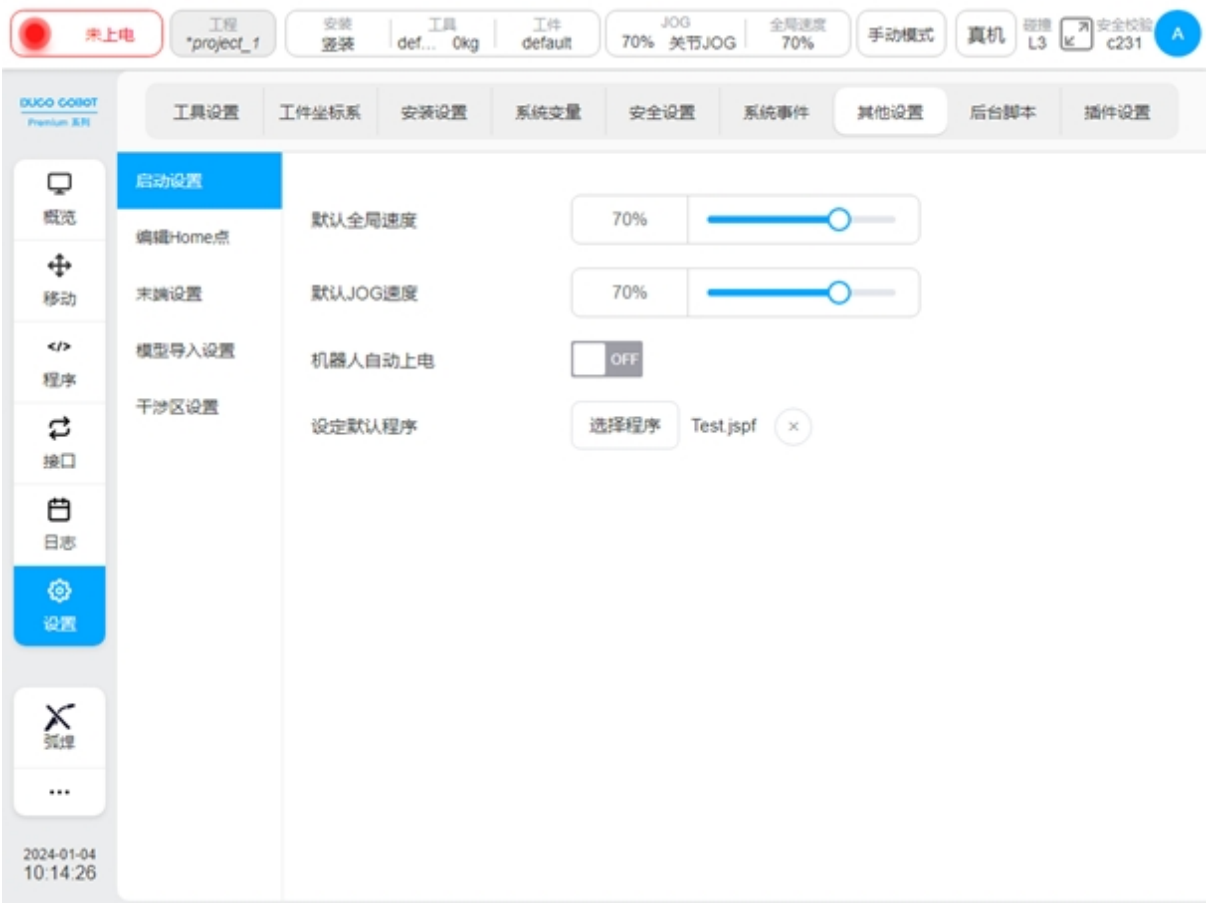
调节默认 jog 速度右侧滑块，可以设置机器人默认的 jog 速度；

“机器人自动上电”右侧开关可进行是否机器人自动上电的切换。

单击“选择程序”按钮，会弹出程序列表窗口。



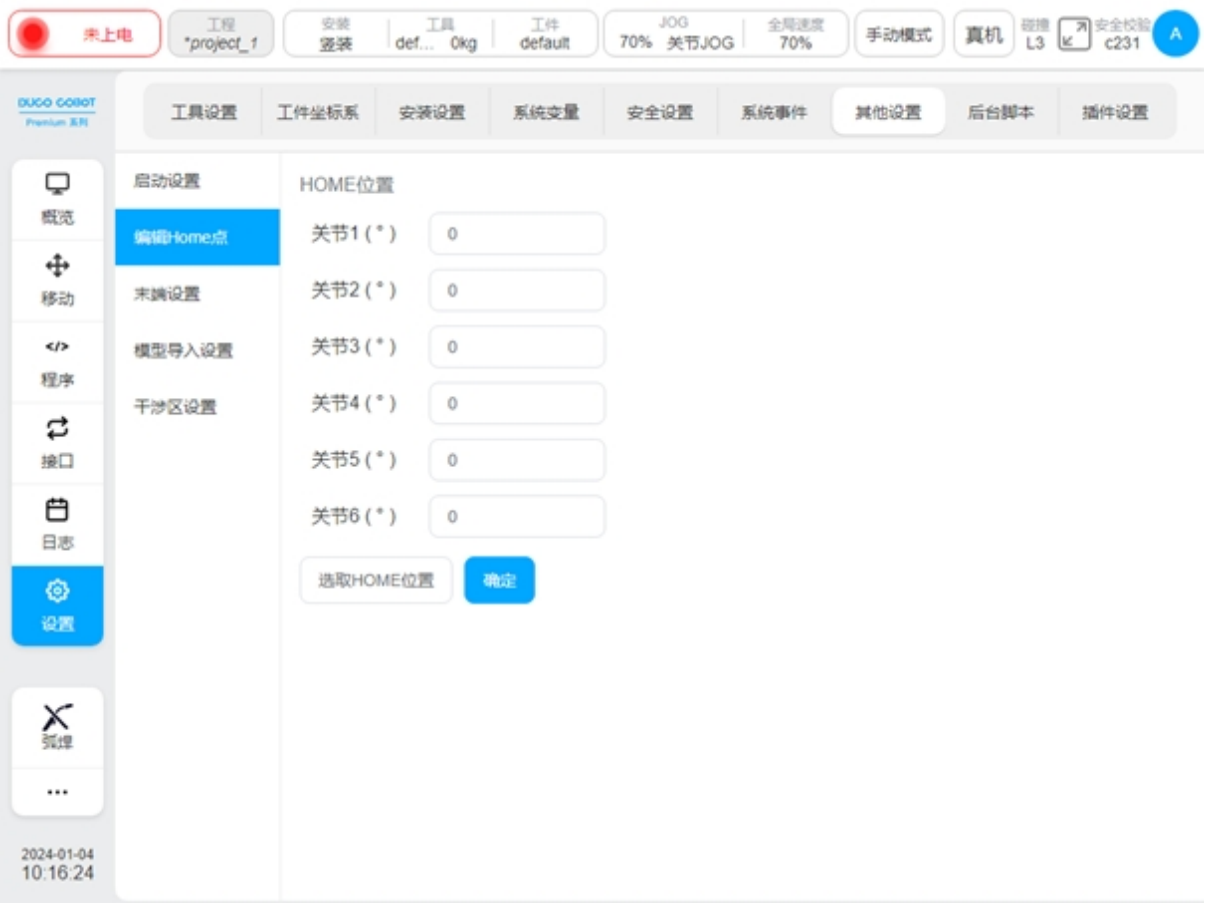
选择好程序后，单击“选择”按钮即可。设置好默认程序后，“选择程序”按钮后会显示出默认程序名称及叉号图标用于取消默认程序。



Home 设置

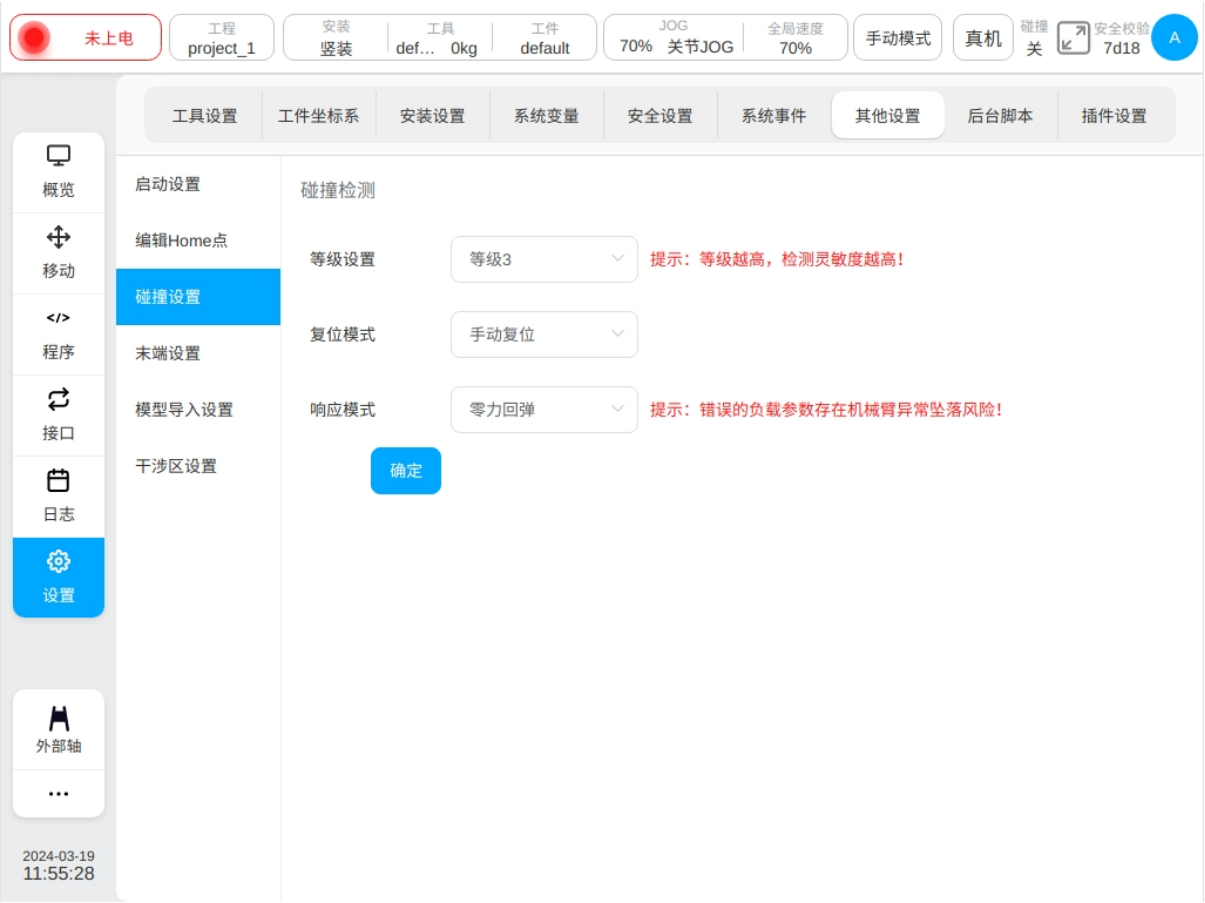
界面上方 HOME 位置处显示当前 HOME 位置时各个关节角度值。用户可通过单击各个关节输入框，手动输入关节角度，也可通过单击“选取 HOME 位置”按钮跳转到移动界面进行示教设置，同上述新建 pose 类型变量过程类似，示教好位置后跳回到当前界面单击“确定”按钮，即可设置生效。

备注： 对于 J9/J9-ZII 控制柜，在选取完”HOME”位置后，需要同步至安全控制器。



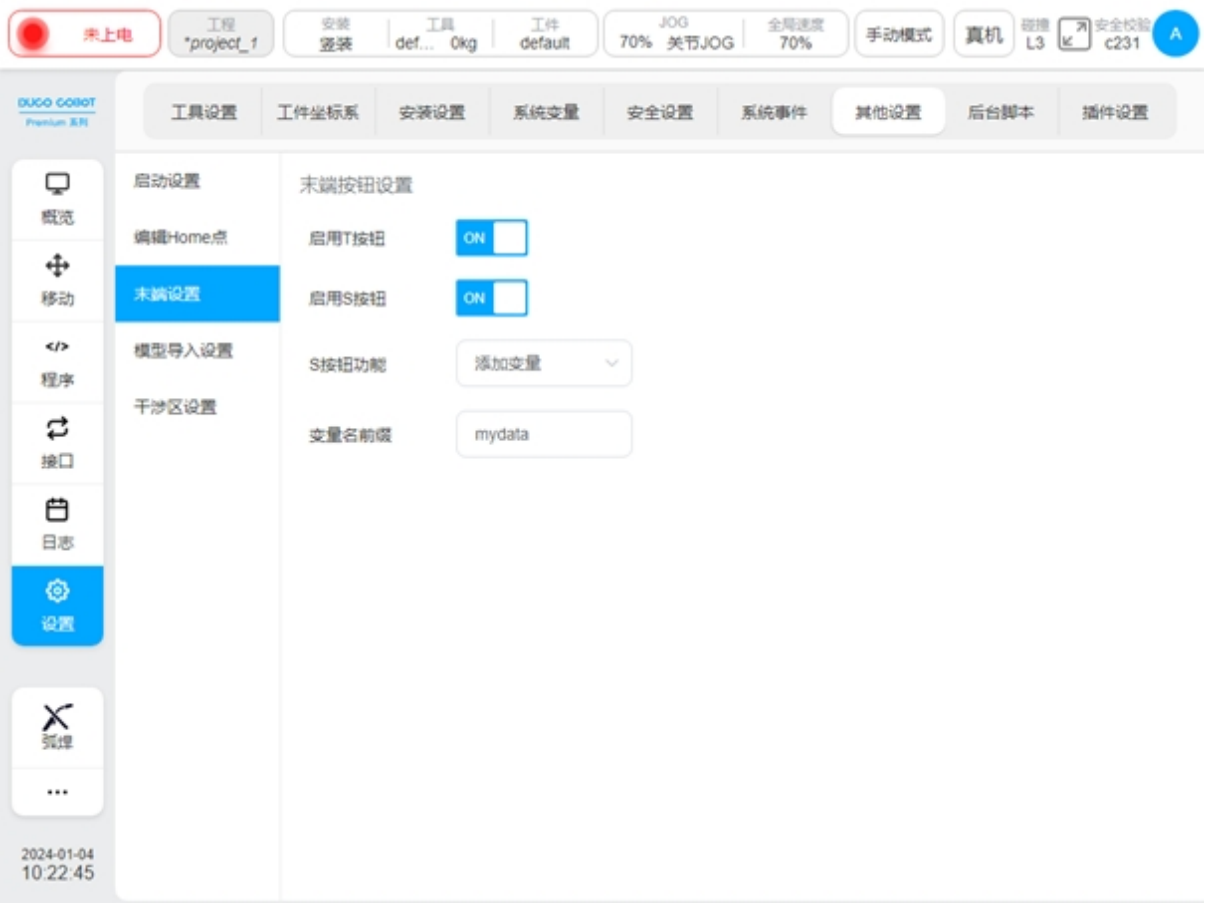
碰撞设置

碰撞设置对于 DC30 控制柜，在其他设置中进行配置；对于 J9/J9-ZII 控制柜，在安全设置中进行配置（详见第 6.2 章节）。碰撞检测区域可进行碰撞检测等级设置、复位模式和响应模式设置。碰撞检测等级分为：关闭、等级 1、等级 2、等级 3、等级 4、等级 5，且等级越高，碰撞检测的灵敏度越高。碰撞检测复位模式分为手动复位和自动复位两种。碰撞检测响应模式可选项有保护性停止、安全急停及零力回弹三种，其中，当选“零力回弹”选项时，会显示提示语“提示：错误的负载参数存在机械臂异常坠落风险！”。用户配置好碰撞检测等级、复位模式以及响应模式后，单击“确定”按钮即可。当碰撞检测响应模式配置为零力回弹模式时，机器人会在检测到碰撞的 1s 内适应碰撞外力从而产生回弹运动，直到与发生碰撞的外部物体不再接触或接触力极小时停止。

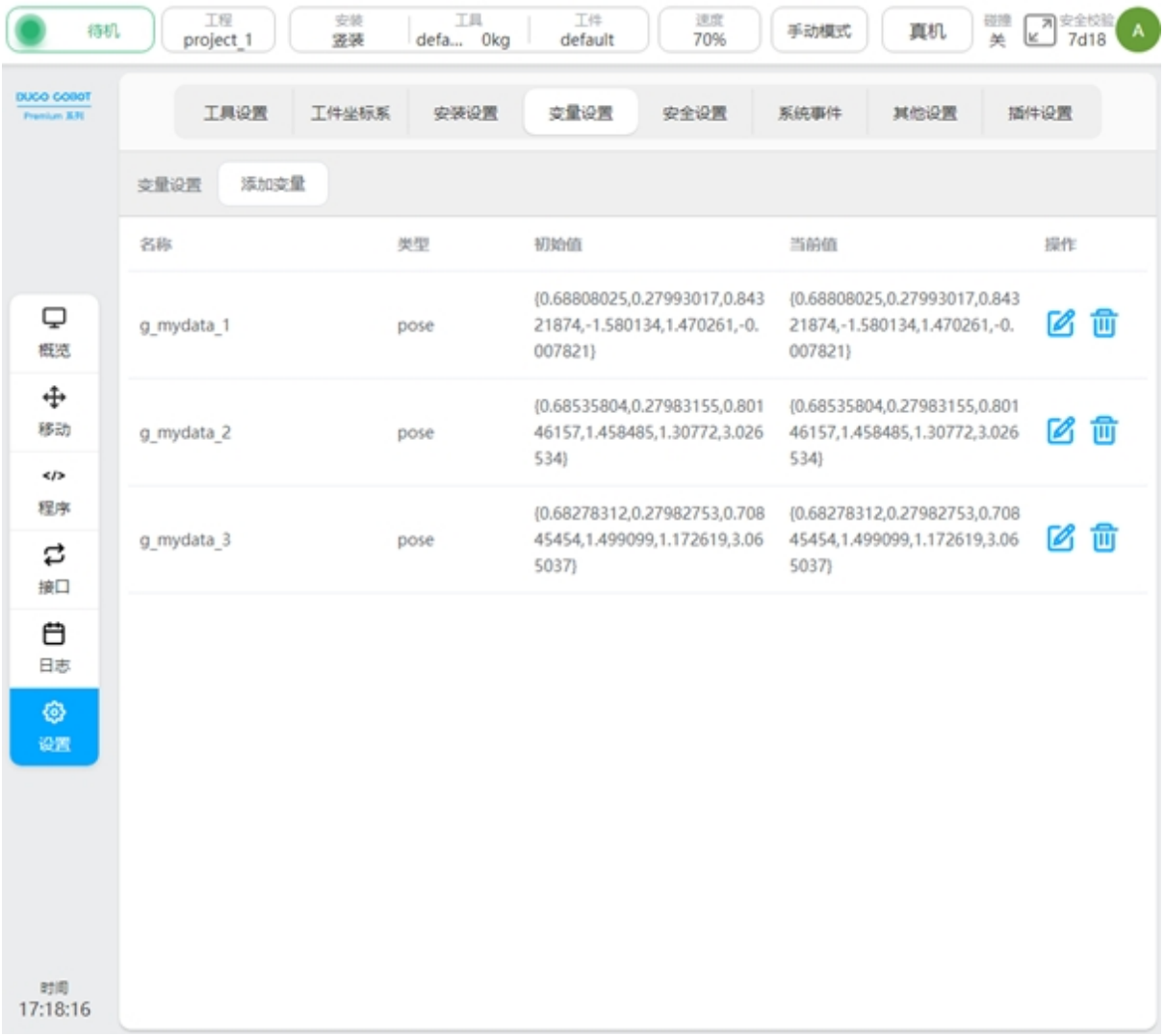


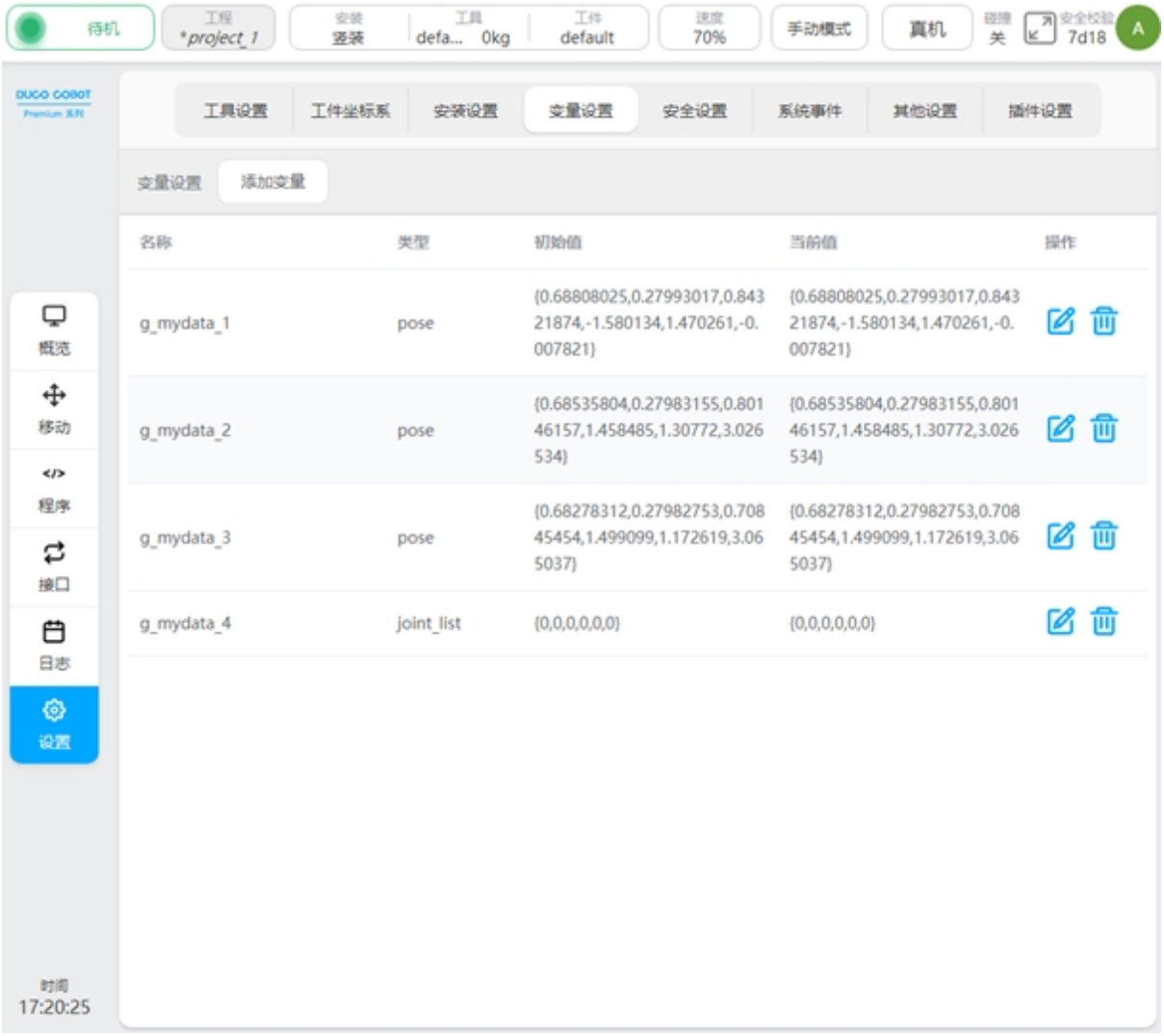
末端设置

对末端 T 按钮和 S 按钮进行相关设置。可以选择是否启用末端的 T 按钮和 S 按钮。选择启用 T 按钮，该按钮具有按住牵引机械臂的功能。启用 S 按钮，可以对 S 按钮功能进行设置。S 按钮功能有三种：添加变量、添加运动指令、添加变量及运动指令。当选择 S 按钮功能为添加变量或者添加变量及运动指令时，还需要配置变量名前缀，默认为 mydata，且记录全局变量后，正式名称为 g_mydata_1、g_mydata_2……，每次添加变量都会检查变量名后缀的数值最大值，并在此基础上做计数累加。



在启用 S 按钮前提下，当配置 S 按钮功能为添加变量时，变量名前缀使用默认值 mydata。单击触发 S 按钮，在变量设置页面会新添加一个 pose 类型的全局变量，如单次触发 3 次 S 按钮，变量设置页面会新增 3 个 pose 类型变量，变量名依次为 g_mydata_1、g_mydata_2、g_mydata_3，如图所示：


待机
工程
project_1
安装
盛装
工具
defa... 0kg
工件
default
速度
70%
手动模式
真机
碰撞
关
安全校验
7d18
A
DUCO COBOT
工具设置
工件坐标系
安装设置
变量设置
安全设置
系统事件
其他设置
插件设置
变量设置
添加变量
名称
类型
初始值
当前值
操作
g_mydata_1
pose
{0.68808025,0.27993017,0.84321874,-1.580134,1.470261,-0.007821}
{0.68808025,0.27993017,0.84321874,-1.580134,1.470261,-0.007821}
g_mydata_2
pose
{0.68535804,0.27983155,0.80146157,1.458485,1.30772,3.026534}
{0.68535804,0.27983155,0.80146157,1.458485,1.30772,3.026534}
g_mydata_3
pose
{0.68278312,0.27982753,0.70845454,1.499099,1.172619,3.065037}
{0.68278312,0.27982753,0.70845454,1.499099,1.172619,3.065037}
概览
移动
程序
接口
日志
设置
时间
17:18:16



长按触发 S 按钮，变量设置页面会新添加一个 joints 类型的全局变量，如长按触发 1 次 S 按钮，变量设置页面新增 1 个 joints 类型变量，如图所示：

当配置 S 按钮功能为添加运动指令时，在满足添加指令条件下（比如：须在程序编辑状态下），单击触发 S 按钮，程序页面会新添加一条 MoveL 指令，触发 S 按钮时机器人的笛卡尔位姿对应记录在 MoveL 指令参数页面的 X/Y/Z/RX/RY/RZ 输入框；当长按触发 S 按钮（超过 2 秒），程序页面新添加一条 MoveJ 指令，触发 S 按钮时刻的机器人每个关节角度对应记录在 MoveJ 指令参数页面的 Joint1/Joint2/Joint3/Joint4/Joint5/Joint6 输入框。

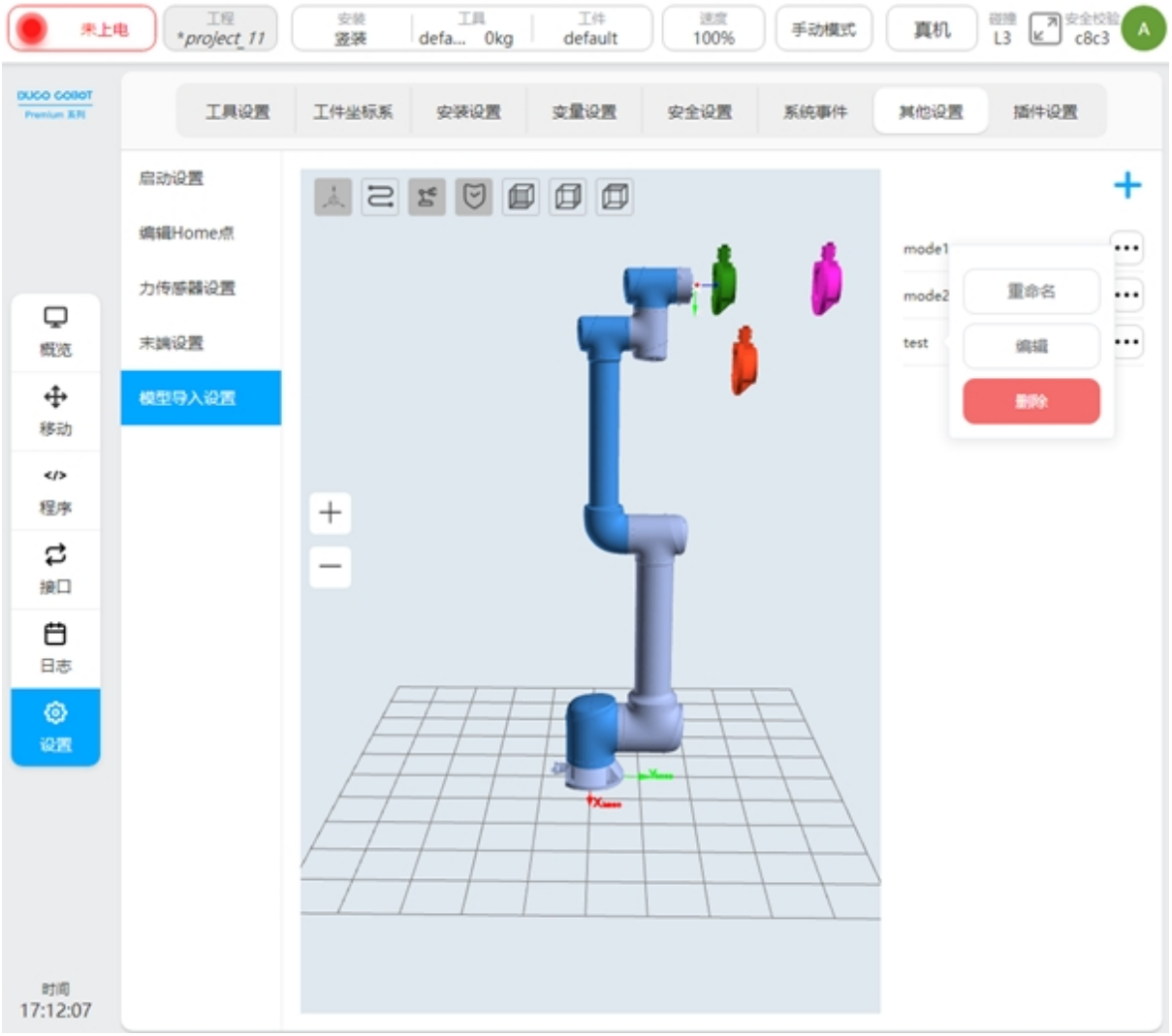
当配置 S 按钮功能为添加变量及运动指令时，单击触发 S 按钮，除了在变量设置页面新添加一个 pose 类型的全局变量之外，该变量记录的是触发 S 按钮时机器人的笛卡尔位姿，还会在满足添加指令条件下，在程序页面新添加一条 MoveL 指令，新添加的 MoveL 指令为设为变量形式，且变量已被默认选为新添加的 pose 类型全局变量；长按触发 S 按钮，除了在变量设置页面新添加一个 joints 类型的全局变量之外，该变量记录的是触发 S 按钮时机器人的每个关节角度，还会在满足添加指令条件下，在程序页面新添加一条 MoveJ 指令，新添加的 MoveJ 指令为设为变量形式，且变量已被默认选为新添加的 joints 类型全局变量。

以上关于运动指令的添加均当程序页面指令节点在 Move 节点下才有效，且新添加的 MoveL 指令坐标系调用触发 S 按钮时正在使用的坐标系，父节点坐标系选项默认不勾选。

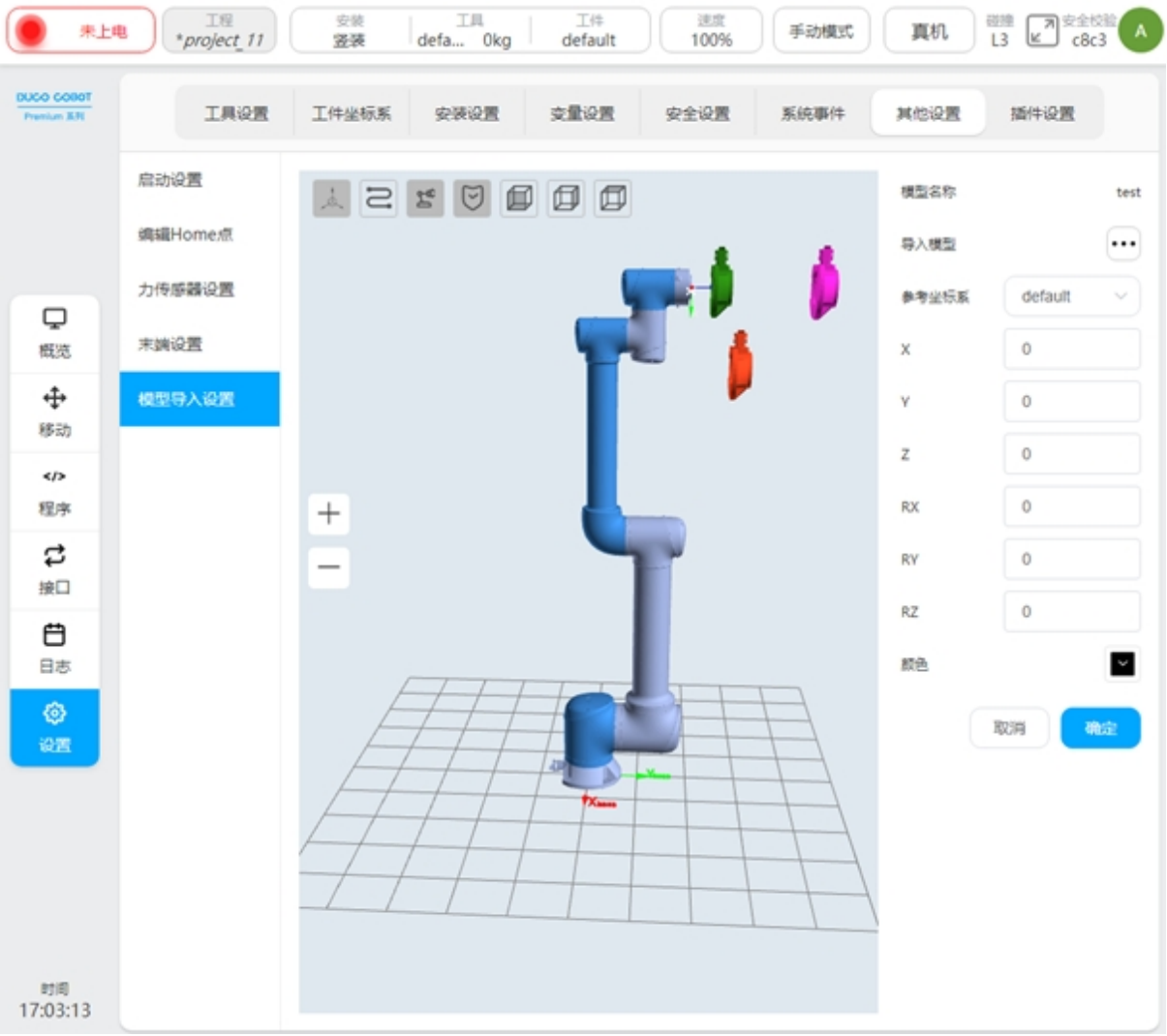
当程序页面指令节点在 Spline 时，S 按钮添加运动指令的方式可以在 Spline 中添加路点。

模型导入设置

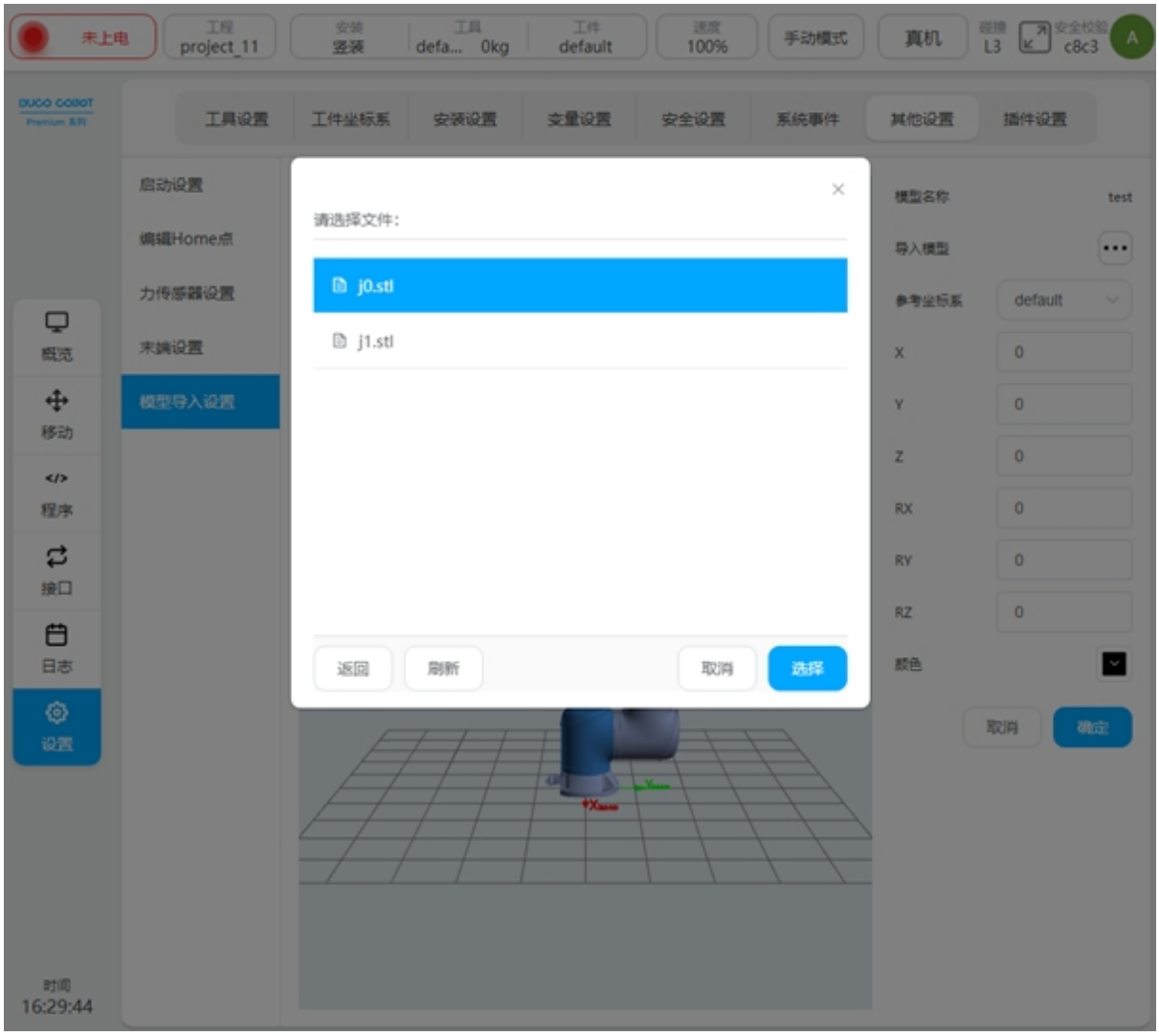
该页面允许导入工具或工件模型并显示，页面左侧显示 3D 仿真模型，右侧是导入的工具/工件模型列表。单击模型列表每行右侧的 ··· 图标，会显示对模型进行重命名、编辑、删除的操作弹框。



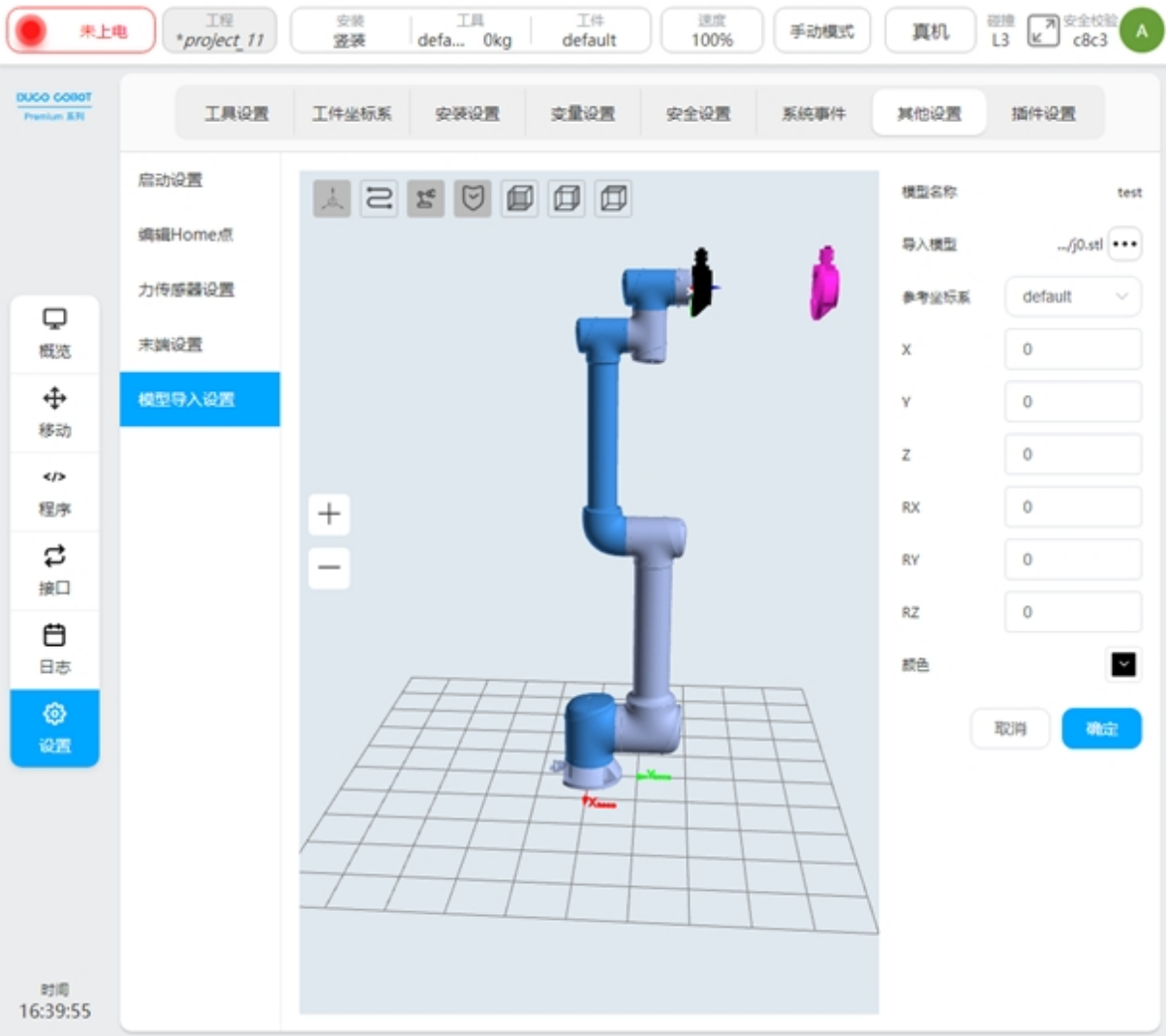
单击右上方 + 图标，弹出虚拟键盘，输入模型名称（如：test）后，进入新添加模型 test 的参数编辑页面，如图所示。



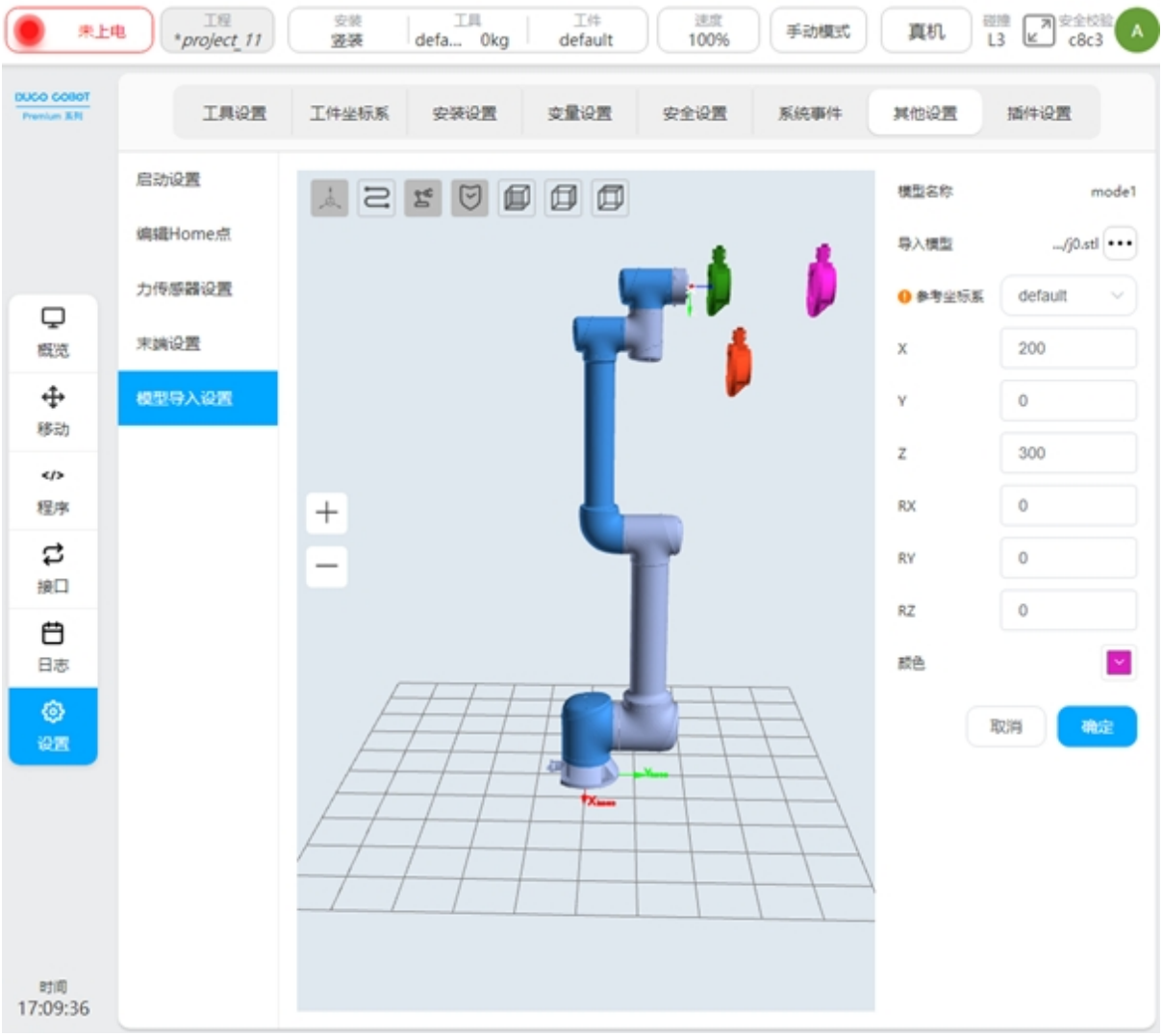
单击“导入模型”后 ••• 图标，会弹出模型导入方式选择弹框，有两种导入方式：外部导入、内部导入。当选择外部导入时，即从外部 U 盘导入所需模型文件，且模型文件大小不得超过 3M；当选择内部导入时，即从系统内部模型文件夹导入所需模型文件。模型文件仅支持后缀为.stl 或.STL 格式。



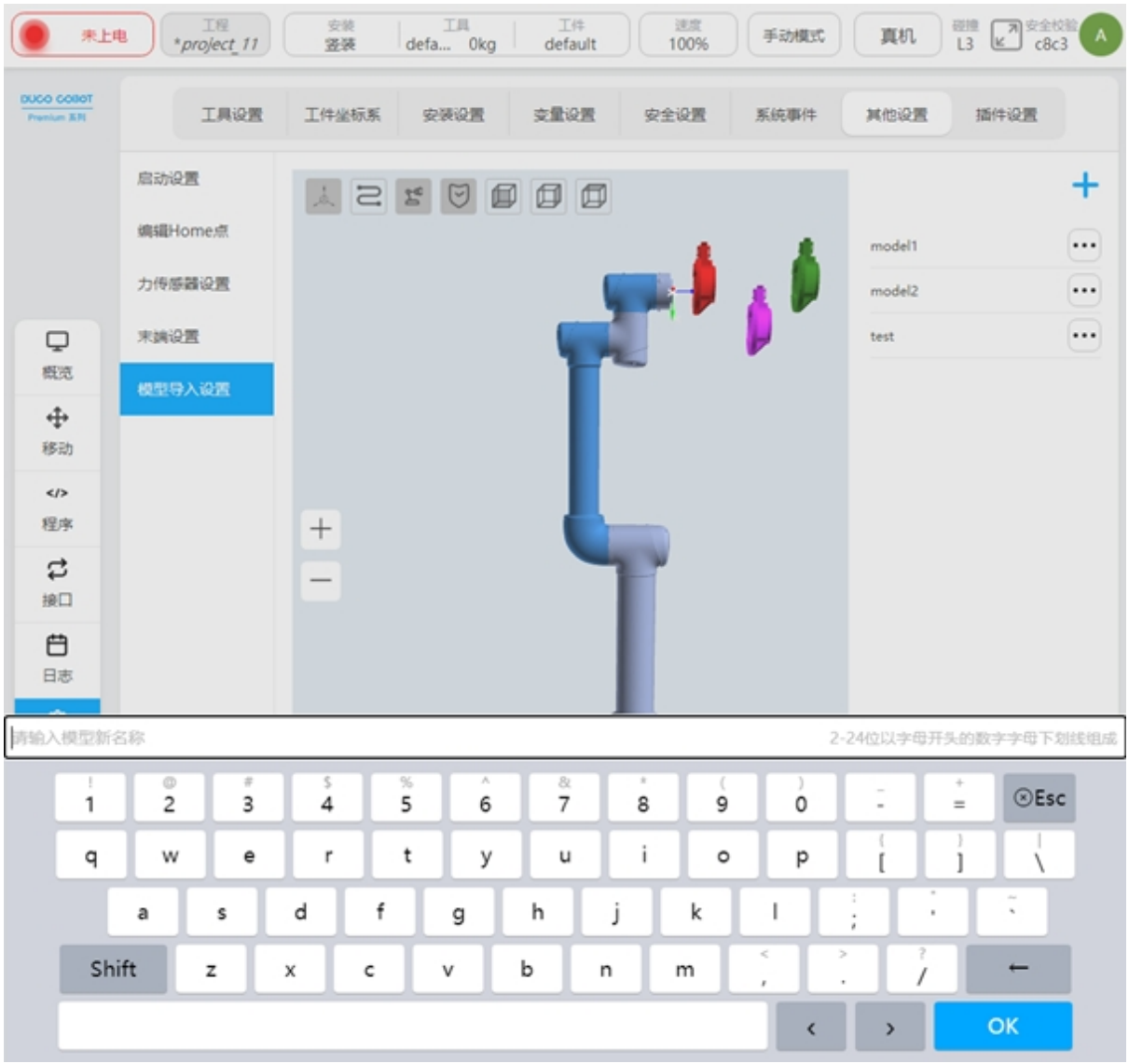
选择好模型文件后，模型显示的参考坐标系默认为工具坐标系 default，X、Y、Z、Rx、Ry、Rz 的值默认均为 0，模型颜色默认为黑色。用户可以根据自身需求对参考坐标系、X、Y、Z、Rx、Ry、Rz 的值以及模型颜色进行设置。



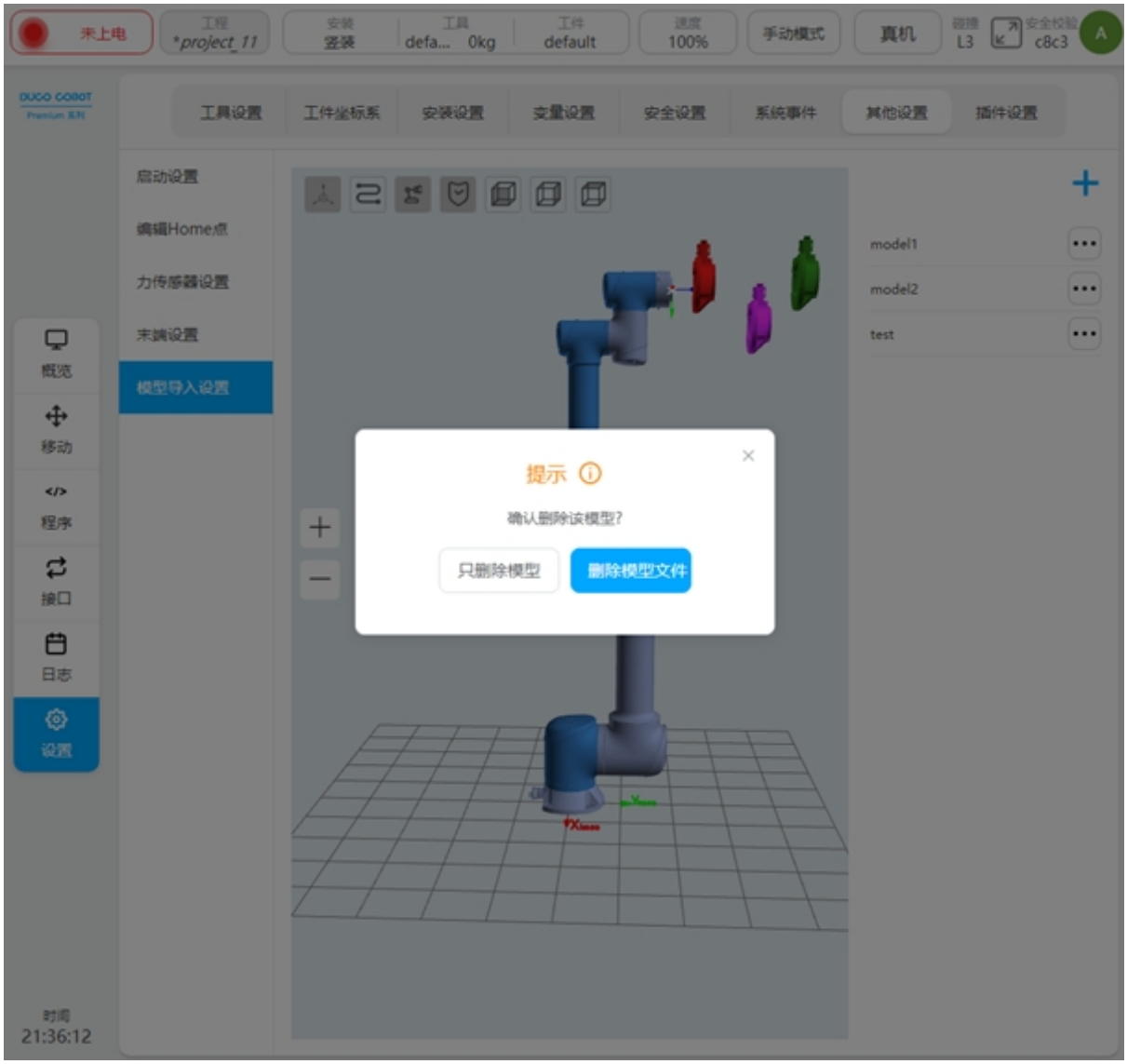
参考坐标系可选择世界坐标系、基座、工具坐标系及工件坐标系。其中，工具坐标系和工件坐标系分别为工具设置和工件坐标系页面中已建立的坐标系。调节模型的参考坐标系及其偏移量、颜色参数会实时变换模型位姿和颜色。当编辑已有模型时（如：mode1），进入模型参数编辑页面，如果选择的工具/工件坐标系参数值被修改，和模型导入设置时用到的坐标系的参数值不一致，会在参考坐标系处显示  提醒图标。



当对模型重命名时，单击该模型右侧的 图标操作弹框中“重命名”按钮，会弹出虚拟键盘，提示请输入模型新名称，如图所示。



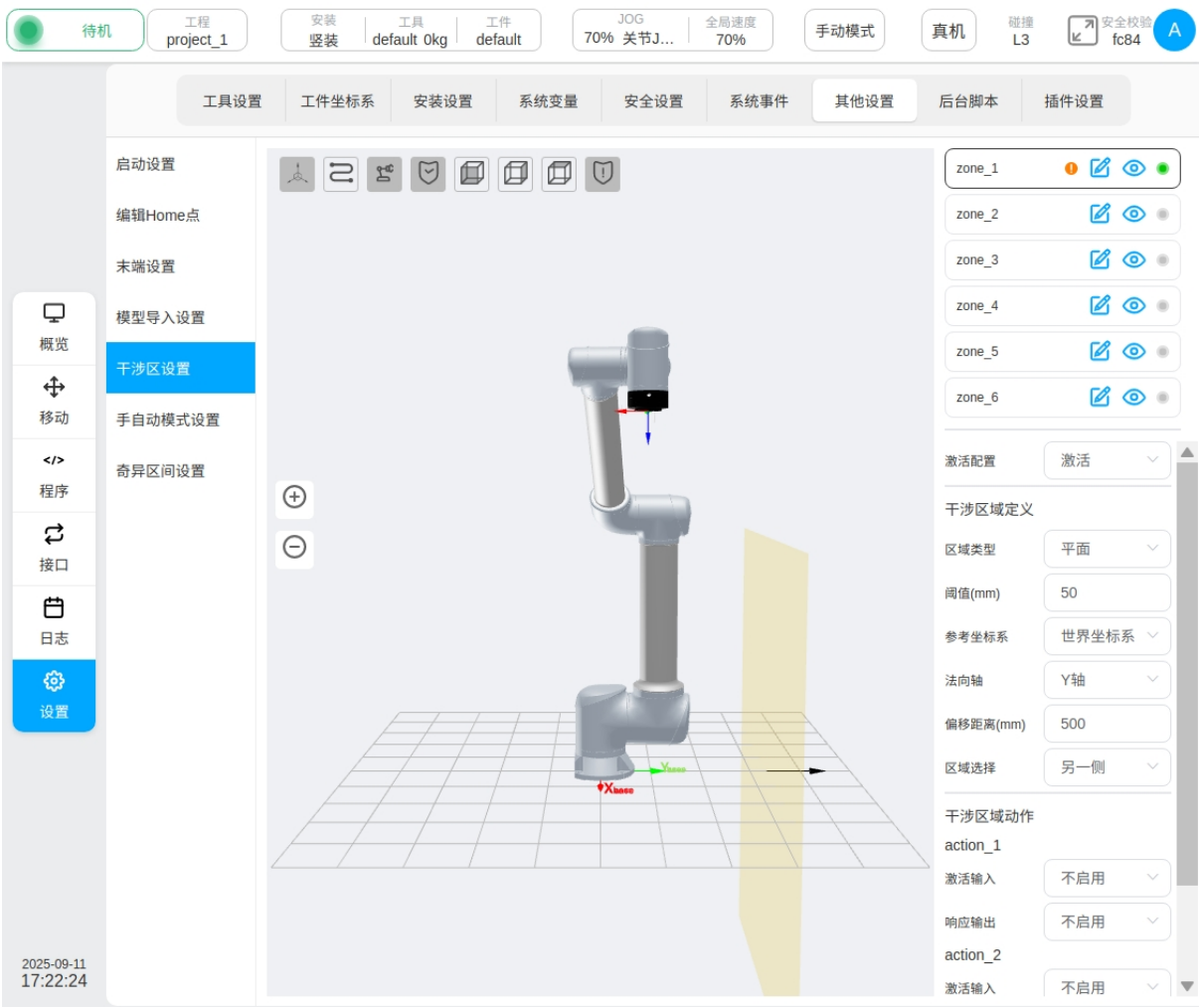
当对已有模型编辑时，单击模型右侧的 ▪▪▪ 图标操作弹框中“编辑”按钮，会进入该模型的参数编辑页面，如上述新建模型时图所示，此处不累赘。当对已有模型进行删除时，单击模型右侧的 ▪▪▪ 图标操作弹框中“删除”按钮，会弹提示框如图所示。点击弹框中右侧“只删除模型”按钮，将只是删除显示的模型；点击“删除模型文件”按钮，不仅会删除显示的模型还会删除对应模型选择的模型文件。点击提示框右上角的叉号即可取消删除模型操作。



干涉区设置

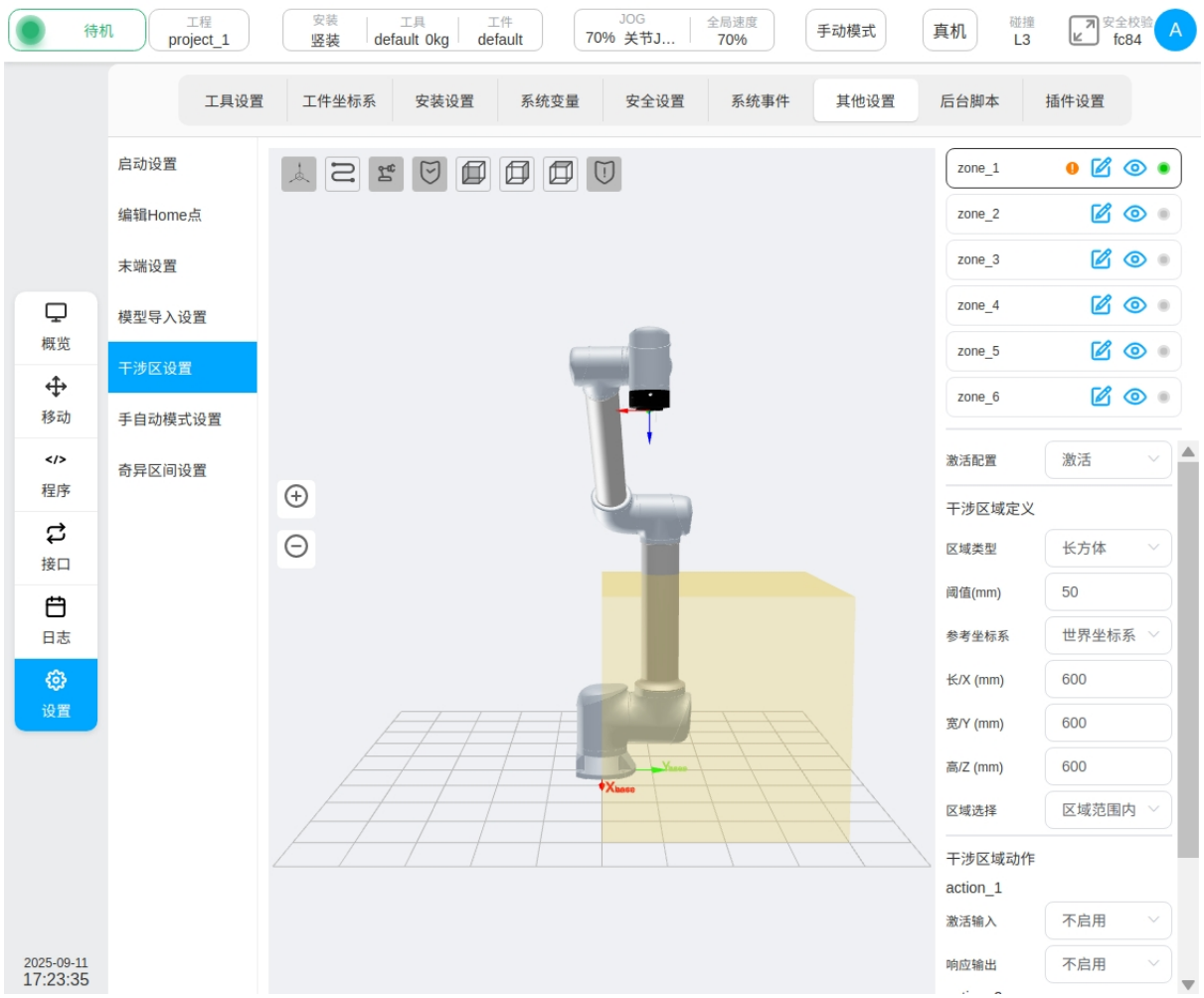
干涉区域定义类型包括平面、长方体、圆柱体、关节区域、外部轴区域五种形式。用户可以最多设置 6 个相互独立的干涉区域。通过点击干涉区域名称后 图标可以对配置的干涉区域类型为平面、长方体、圆柱体的干涉区域在 3D 显示区进行显示和隐藏的切换，且可以点击区域名称（如：默认名称 zone_1）后的 图标修改安全区域名称。未被禁用的干涉区域会显示 ，否则会显示 。对于每个干涉区域，修改任意一项参数后，在没有点击“确认”按钮的情况下，干涉区名称后会显示 图标进行提示。

干涉区域的激活配置有禁用和激活两种可选。干涉区域动作配置激活输入和响应输出。当激活输入信号为高电平时，且机器人末端 TCP 进入干涉区内，则机器人将暂停程序运行。当此激活输入信号消失时，程序将自动复位，机器人继续运行程序。激活输入可配置通用输入、寄存器 BOOL 输入；响应输出可配置通用输出、寄存器 BOOL 输出。机器人进入干涉区域后，触发干涉区域输出，机器人在干涉区内作业时，同样监控干涉区激活输入信号，当激活输入信号为高电平时，立即暂停作业。用户可最多配置 2 个干涉区动作。

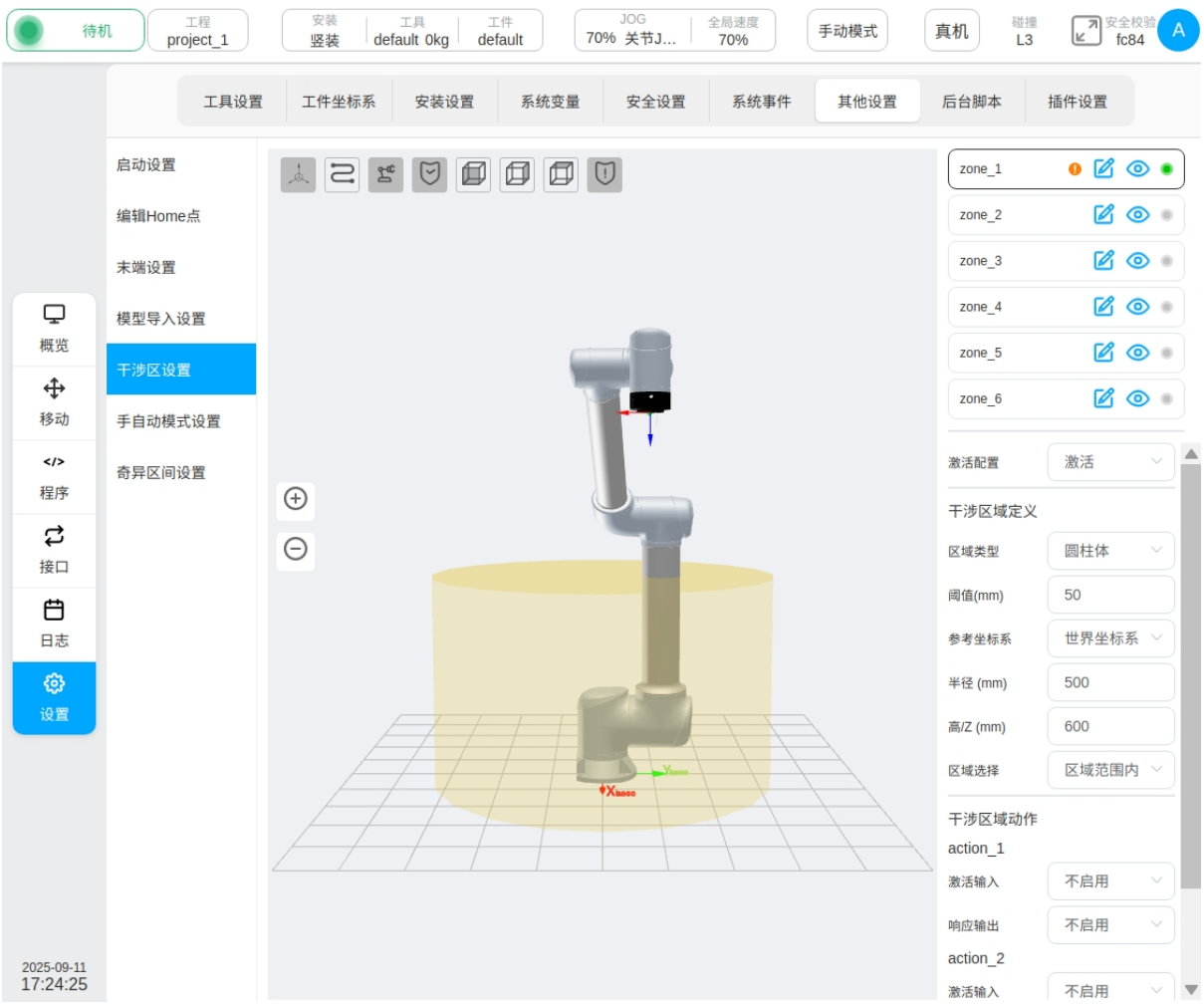


当配置区域类型为平面时，平面区域的参考基准是基于世界坐标系/基座/自定义的工件坐标系，通过设定平面法向轴、法向偏移距离从而确定平面。平面的设置可以通过“区域选择”配置生效区域，且对应方向在 3D 区域中通过黑色箭头显示。

当配置区域类型为长方体时，长方体区域的设置基准时基于世界坐标系/基座/自定义的工件坐标系。以工件坐标系作为长方体的一个角点，三个坐标轴方向分别对应长（X）、宽（Y）、高（Z）。长宽高的设置范围均为-3000mm—3000mm。

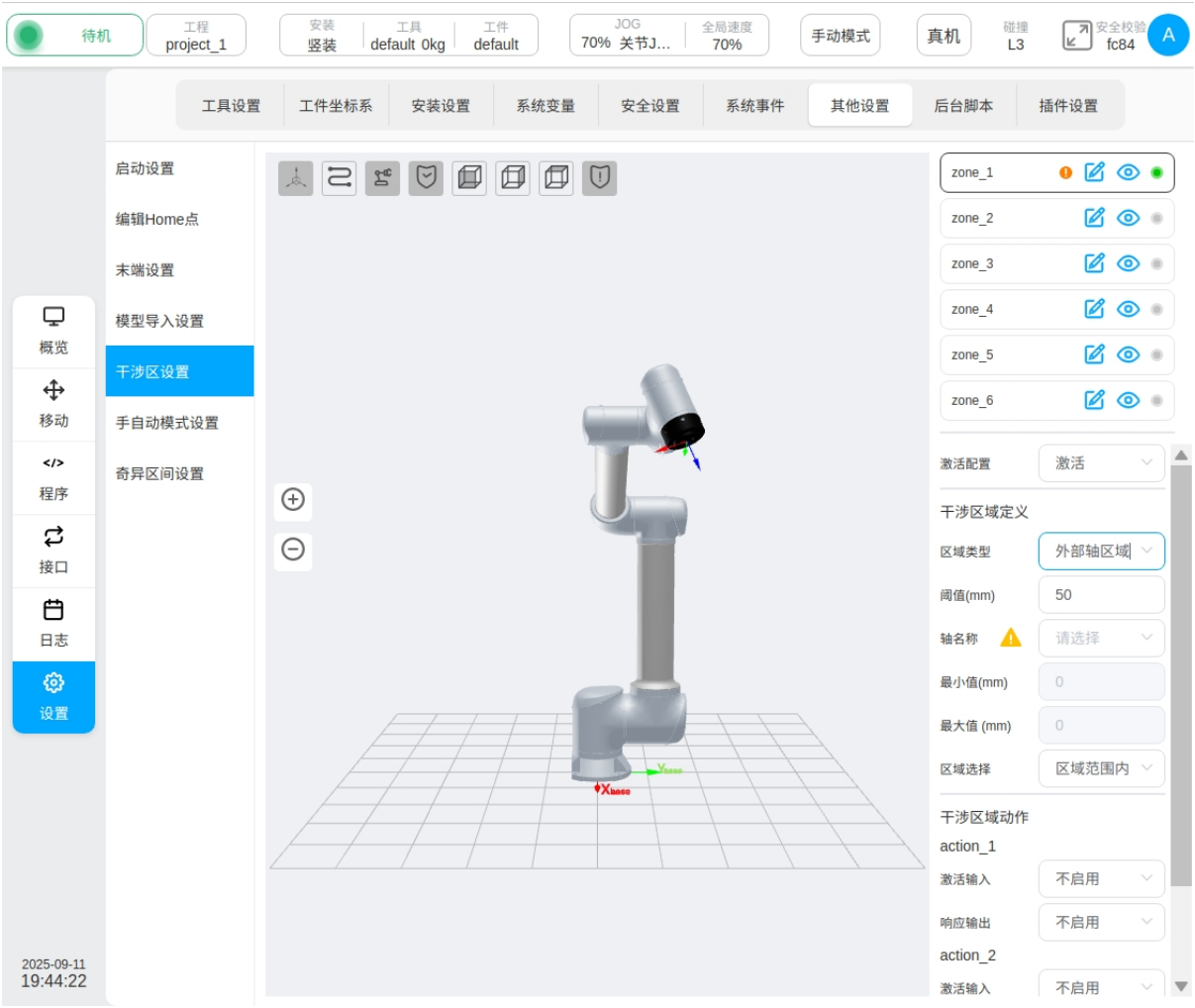


当配置区域类型为圆柱体时，其设置基准时基于世界坐标系/基座/自定义的工件坐标系。以工件坐标系作为圆柱体底面的圆心，Z 方向指向高度方向，可以设置半径、高度。半径取值范围为 0-3000mm，高度取值范围为-3000mm—3000mm。



当配置区域类型为关节区域时，可选择关节 1—关节 6 任意一个，进行最大值和最小值的设置。最大值和最小值的取值范围均为-360°-360°。

当配置区域类型为外部轴区域时，可配置某一外部轴的轴名称，以及外部轴的最大值（上限值）与最小值（下限值）。当系统没有添加外部轴时，“轴名称”后会显示  图标进行提示，且最大值和最小值输入框均被禁用，区域配置底部的“确定”按钮也被禁用不可点击。

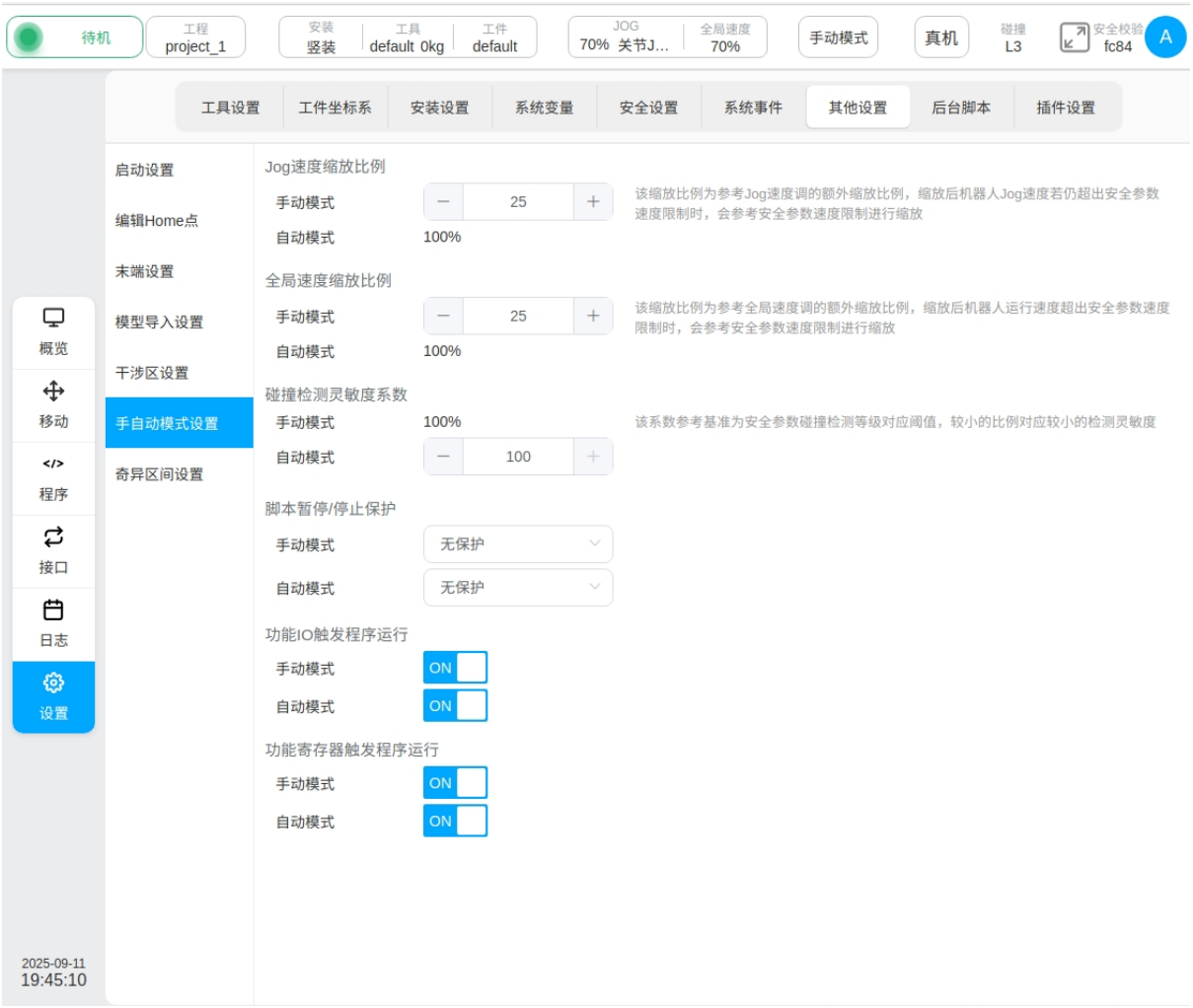


除平面干涉区外，所有其他干涉区类型的区域选择参数可选项为“区域范围内”和“区域范围外”，参数默认为“区域范围内”。当用户选择“区域范围内”作为区域选择类型时，用户所定义空间几何体内部或关节外部轴角度限制范围内为干涉区；相反地，用户选择“区域范围外”作为区域选择类型时，用户所定义空间几何体或关节外部轴角度限制范围外为干涉区。

手自动模式设置

部分机器人功能在手动模式与自动模式下，存在功能表现及性能上的差异。在手自动模式设置界面中，可以对该类功能针对手动模式与自动模式下存在差异化的参数分别进行自定义配置，从而根据实际需求对功能表现进行调整，以提升最终人机交互以及机器人性能表现。

当前支持配置手自动模式参数的功能参数有：Jog 速度缩放比例、全局速度缩放比例、碰撞检测灵敏度系数、脚本暂停/停止保护、功能 IO 触发程序运行、功能寄存器触发程序运行。



Jog 速度缩放比例：默认情况下，在使用机器人 Jog 功能时，机器人所产生的最终速度控制指令大小在手动模式下相比自动模式下会以 25% 的速度进行运动。可以通过修改 Jog 速度缩放比例系数从而调整该比例关系。

全局速度缩放比例：默认情况下，在使用除机器人 Jog 功能以外的机器人运动功能时，如移动到、运行程序中的机器人运动程序等，机器人所产生的最终速度控制指令大小在手动模式下相比自动模式下会以 25% 的速度进行运动。可以通过修改全局速度缩放比例系数从而调整该比例关系。

备注： 机器人最终所产生的运动速度控制指令，始终优先保证机器人不会违反安全参数中所设定的速度相关安全限制，如机器人末端速度限制（TSL）、机器人关节速度限制（SLS）等。因此若实际使用过程中，发生调整 Jog 速度缩放比例或全局速度缩放比例，最终机器人实际运动速度并未对应发生改变的情况，应优先确认机器人实际运动速度是否接近安全限制所设定的参数限制。

碰撞检测灵敏度系数：由于机器人在自动模式下会以较高的速度运行，此时对于机器人所受外力的观测精度会降低。因此在默认情况下，碰撞检测功能会以相较于手动模式碰撞检测灵敏度 60% 的检测灵敏度进行工作。机器人实际使用过程中，若希望机器人在自动模式下同样保证与手动模式下相同或接近的碰撞检测灵敏度，可以通过增大碰撞检测灵敏度系数实现。反之，若机器人外围存在更高等级的安全设施以保证机器人自动模式下的碰撞产生风险，可以适当降低碰撞检测灵敏度系数已降低机器人运行过程中的碰撞误报概率。

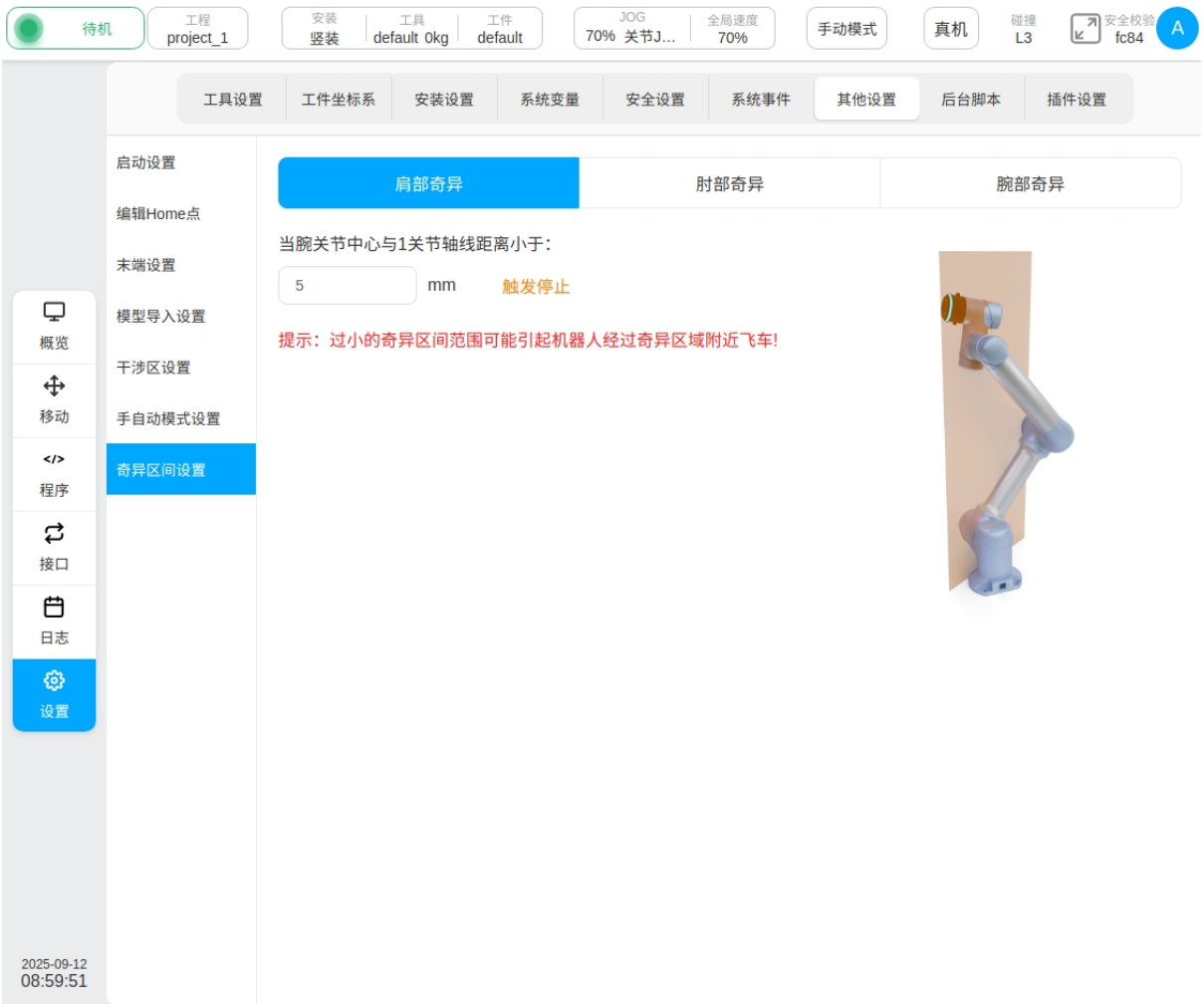
脚本暂停/停止保护：默认情况下，无论手动模式还是自动模式，在机器人程序运行过程中，按下暂停或停止按钮都会直接执行暂停或停止操作。在实际使用过程中，为了防止机器人在不同模式下可能存在的使用过程中的误触情况，可以通过修改脚本暂停/停止保护参数以启用对机器人程序运行过程中暂停或停止程序操作的二次确认保护。

功能 IO 触发程序运行：默认情况下，无论手动模式还是自动模式，机器人在配置了触发程序运行的功能 IO 输入且对应 IO 接收到对应的上升沿指令时都会直接执行运行程序操作。在实际使用过程中，为了防止现场在调试机器人过程中接收到来自于外部连接设备所产生的错误的功能 IO 指令而错误执行运行机器人程序操作，可以通过修改功能 IO 触发程序运行从而长期或临时屏蔽此类行为，降低安全风险。

功能寄存器触发程序运行：与功能 IO 触发程序运行类似，默认情况下，无论手动模式还是自动模式，机器人在配置了触发程序运行的功能寄存器输入且对应寄存器接收到对应的上升沿指令时都会直接执行运行程序操作。在实际使用过程中，为了防止现场在调试机器人过程中接收到来自于外部连接设备所产生的错误的功能寄存器指令而错误执行运行机器人程序操作，可以通过修改功能寄存器触发程序运行从而长期或临时屏蔽此类行为，降低安全风险。

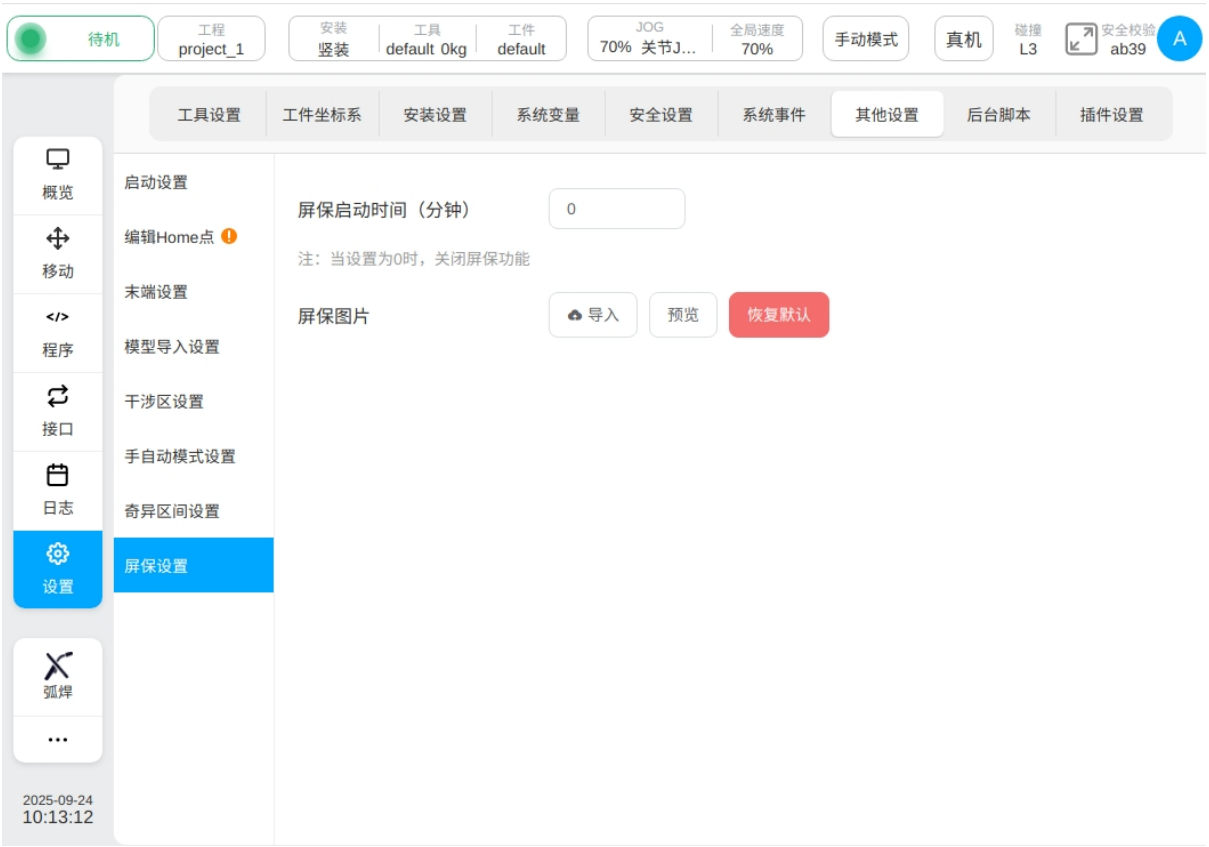
奇异区间设置

该页面可设置肩部奇异、肘部奇异、腕部奇异三个奇异区间。肩部奇异：当腕关节重心与 1 关节轴线距离小于某一数值时，会触发程序停止；肘部奇异：当 3 关节角度值与奇异关节值的差值小于某一数值时，会触发程序停止；腕部奇异：当 5 关节角度值与奇异关节值的差值小于某一数值时，会触发程序停止。需要注意的是无论是哪种奇异区间设置，过小的奇异区间范围可能引起机器人经过奇异区间附近飞车！



屏保设置

屏保设置子页面可以设置屏保启动时间，范围为 0-30 分钟，默认为 0，即关闭屏保功能。如图所示：

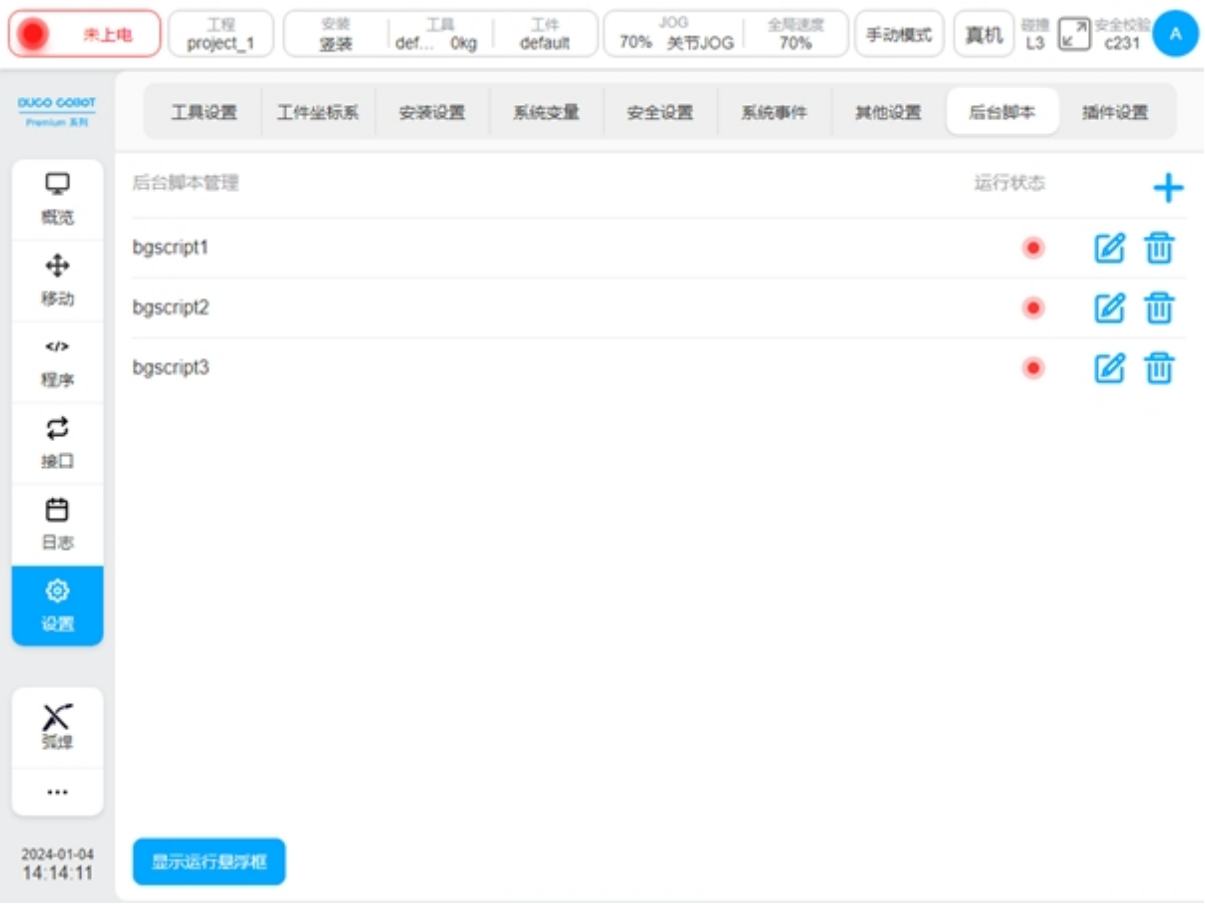


屏保默认为开机画面，用户点击“导入”按钮，可选择导入图片，注意可择图片的格式限制和文件大小限制。点击“预览”按钮，可预览设置的屏保图片；点击“恢复默认”按钮，可恢复开机默认屏保。

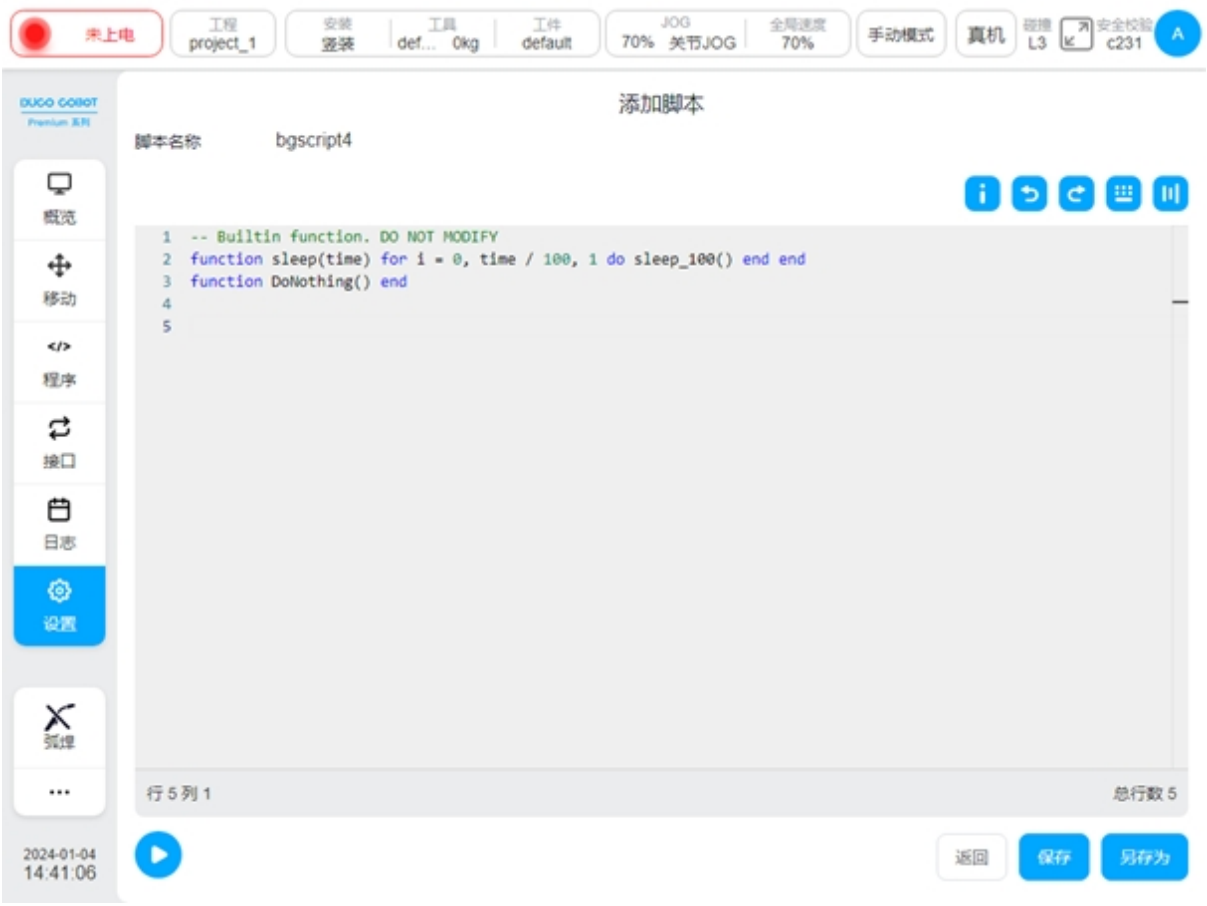
备注： 可选图片文件格式为 jpg/jpeg、png、gif，文件大小限制最大为 10MB。

2.12.7 后台脚本

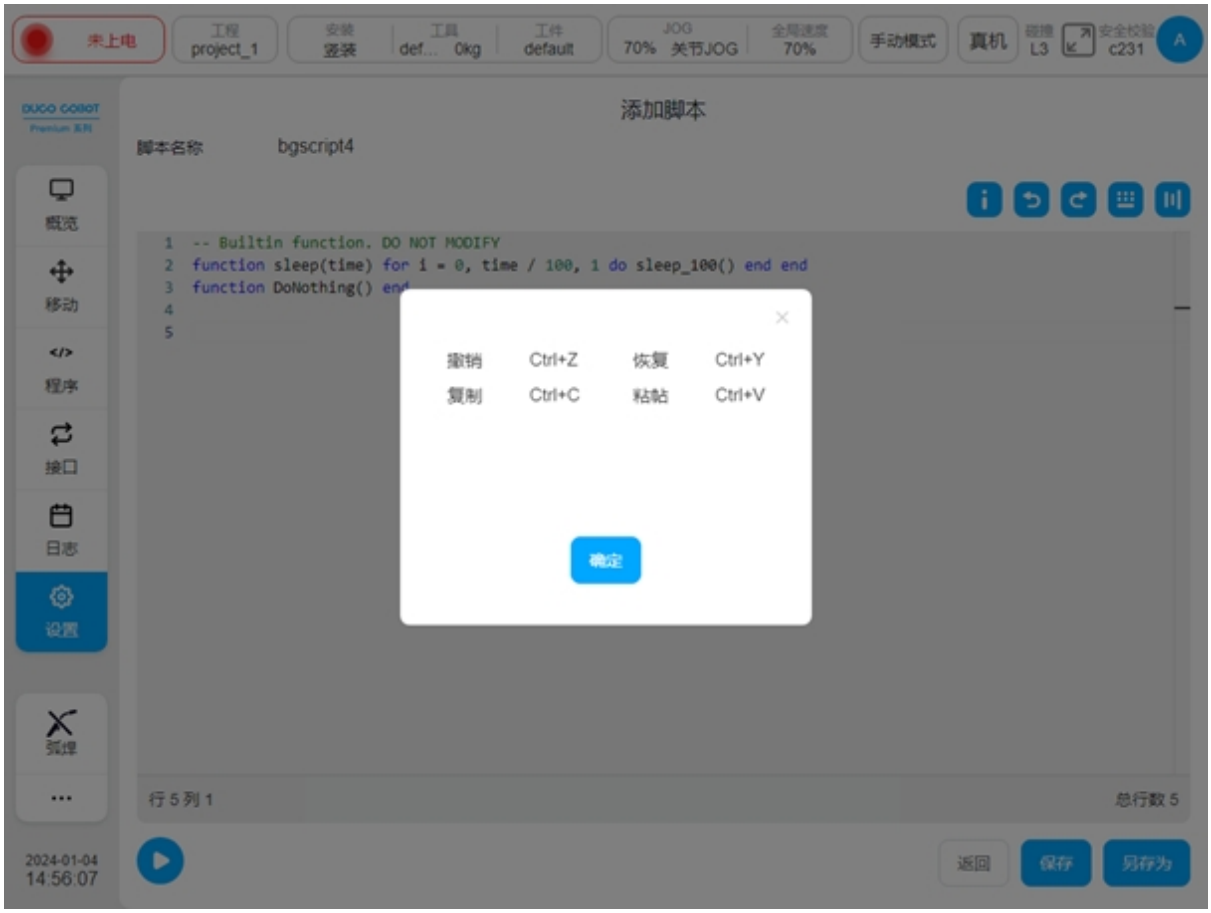
后台脚本子页面主要是管理独立于机器人程序的后台程序，这些脚本是在机器人未上电或未上使能的情况下就可以独立运行的。后台脚本用来处理周期性通讯或者逻辑较复杂的系统事件响应。后台脚本管理页面显示已创建的后台脚本列表，脚本运行状态，以及相关操作按钮，用户可进行后台脚本的添加、编辑、删除操作。如下图所示。



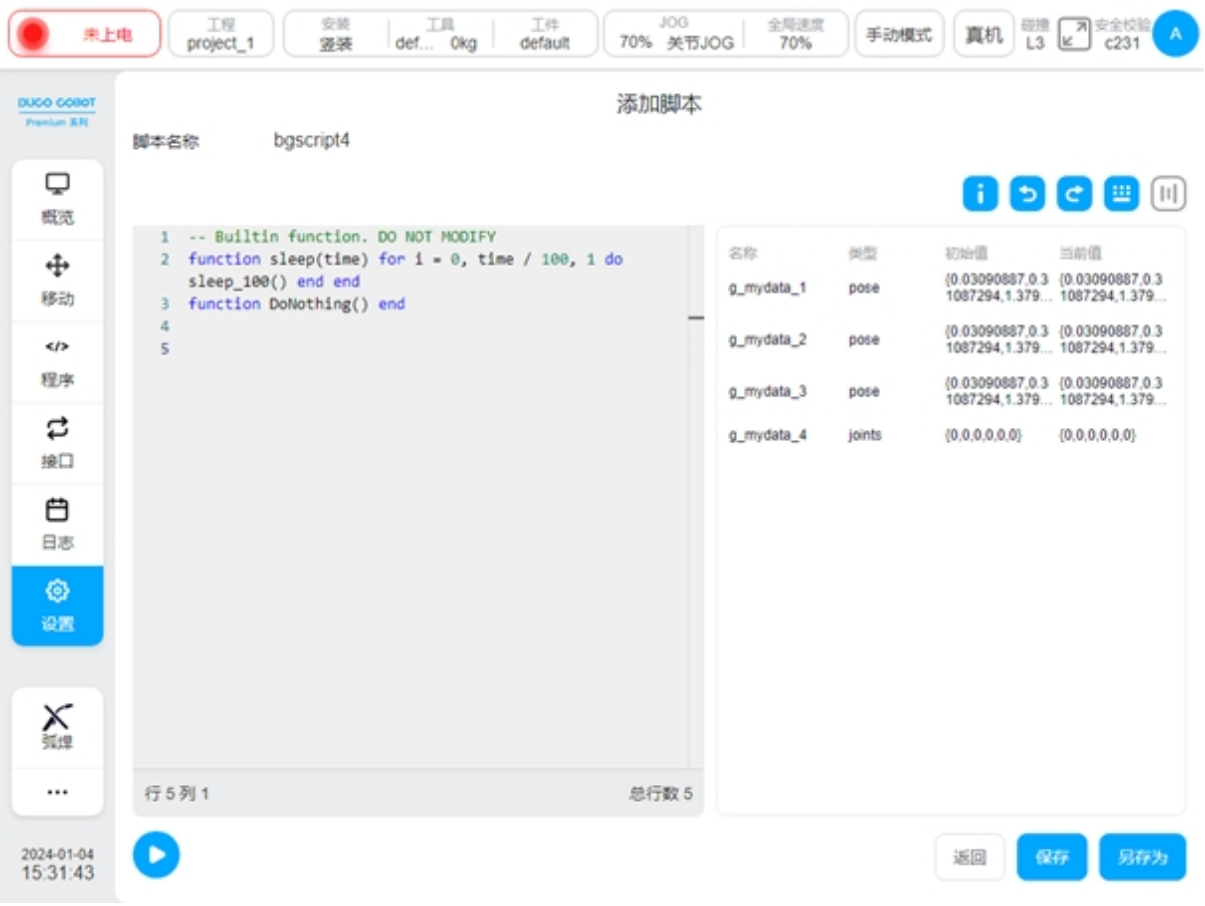
单击页面右上角  图标，弹出虚拟键盘，默认后台脚本名称为 `bgscript+` 序号（序号为在已有的后台脚本数量基础上加 `1`），确认好后台脚本名称后，进入该添加脚本的编辑页面，如下图所示：



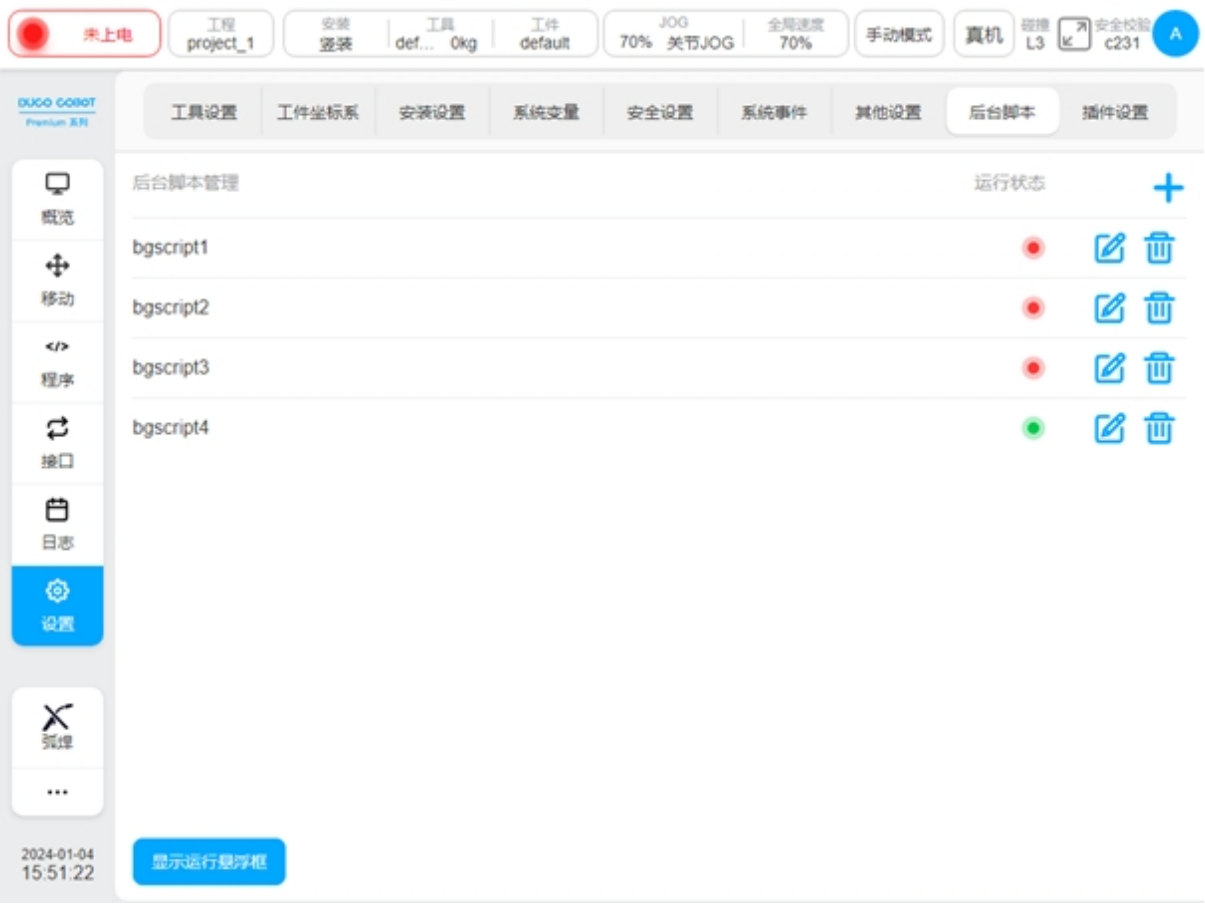
页面中间区域为脚本编辑区域，单击右上方  提示图标会弹出编辑快捷键提示框，如下图所示，  为撤销图标，  为恢复图标。单击  会弹出系统快捷输入框，悬浮在编辑器上方且该弹框可被按住拖动。



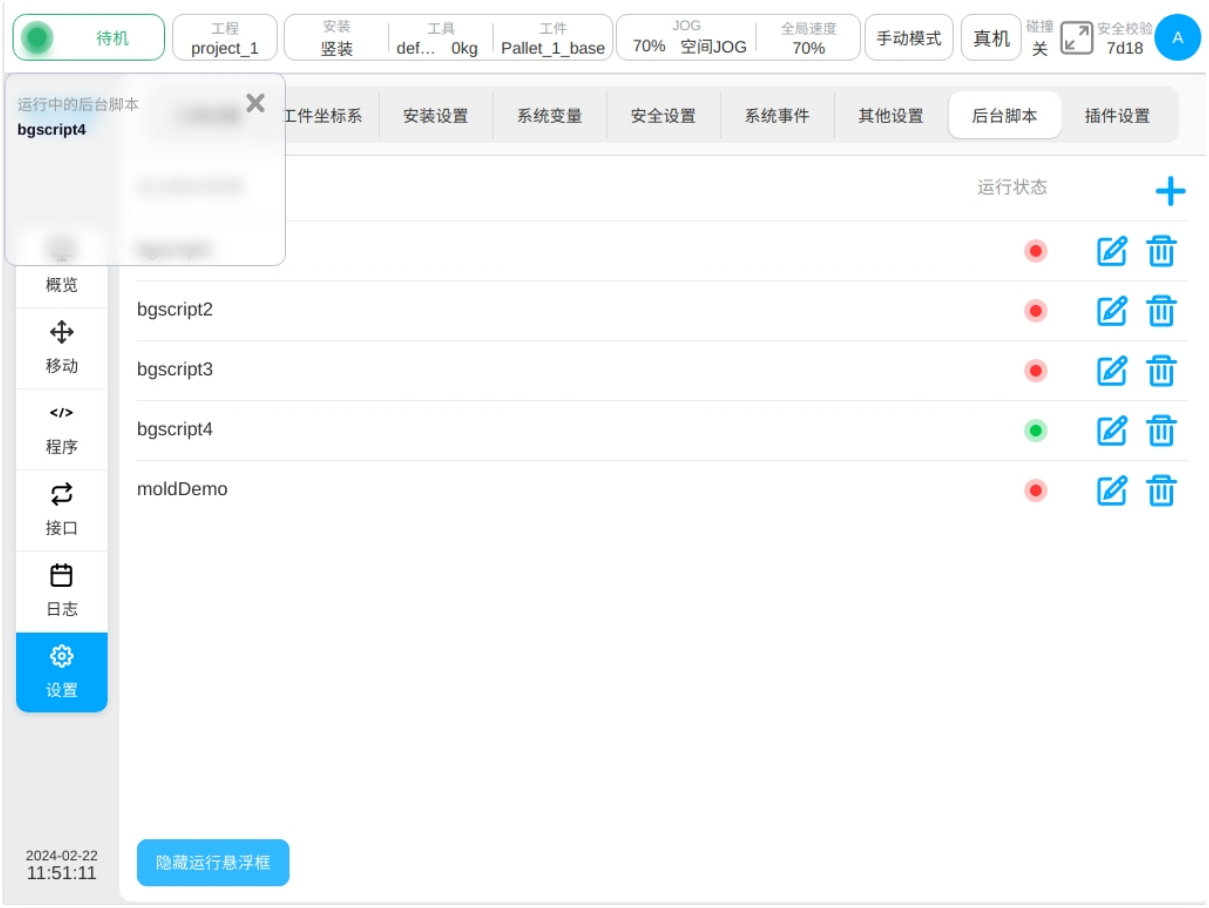
单击  会显示系统变量，如下图所示，隐藏系统变量单击图标即可。



编辑好脚本后，单击页面右下方“保存”按钮，即可保存新添加胡后台脚本，然后单击“返回”按钮到管理页面，后台脚本列表页面会显示刚新添加的后台脚本。也可单击“另存为”按钮给脚本重新命名后保存。单击页面左下方  图标，可运行后台脚本，此时后台脚本管理页面状态运行栏会显示 如下：



单击后台脚本管理页面左下角“显示运行悬浮框”按钮，会在页面左上角显示正在运行的后台脚本悬浮框，如下图所示：



若需要对已创建的后台脚本进行编辑，单击该后台脚本列表行右侧  图标，会进入编辑脚本页面；单击后台脚本列表右侧  图标可以删除添加的后台脚本。

后台脚本编辑与 `Script` 功能块脚本编辑类似，此处不累述。

警告： 在后台脚本最后，及后台脚本的 `while` 循环中，需要加 `sleep` 延迟。否则后台脚本持续运行将会占用过多系统资源，最终导致机器人算法报错。

备注： 后台脚本中禁止添加 `Move` 类移动功能块!

2.13 系统设置

系统设置主要包括对网络和语言的设置，以及对系统、安全控制器固件的版本更新、时间设置、插件管理、机器人参数（配置文件的导入/导出）、云平台设置以及恢复出厂设置。单击状态栏最右边的用户头像，选择“系统设置”按钮，会显示系统设置页面。页面左侧是导航选项卡，右侧是对应选项卡显示内容区。

2.13.1 语言设置

在语言设置中可以进行语言切换，选择需要设置的语言，单击“确定”按钮，语言设置成功。

2.13.2 网络设置

单击“网络设置”选项卡，会显示网络设置相关内容，可以进行 IP 设置和 WLAN 设置。其中，网络 IP 设置根据不同的硬件平台，显示不同数量的网络端口配置信息，以方便用户修改不同端口的 IP 地址。显示的 IP 详细信息包括是否启动 DHCP、IP 地址和子网掩码；IP 设置可选择 DHCP 或静态 IP 设置。

当选择 DHCP 时，IP 地址、子网掩码和默认网关都是由 DHCP 服务器自动分配的，用户不能进行手动输入，选择后单击“设置”按钮即可。当选择静态 IP 设置时，用户需要手动输入 IP 地址、子网掩码和默认网关，再单击“设置”按钮即可。

备注： 当控制柜存在多个物理网口时，接口设置中的 TCP/IP 服务器的 IP 地址，对应 LAN1 的 IP 地址。

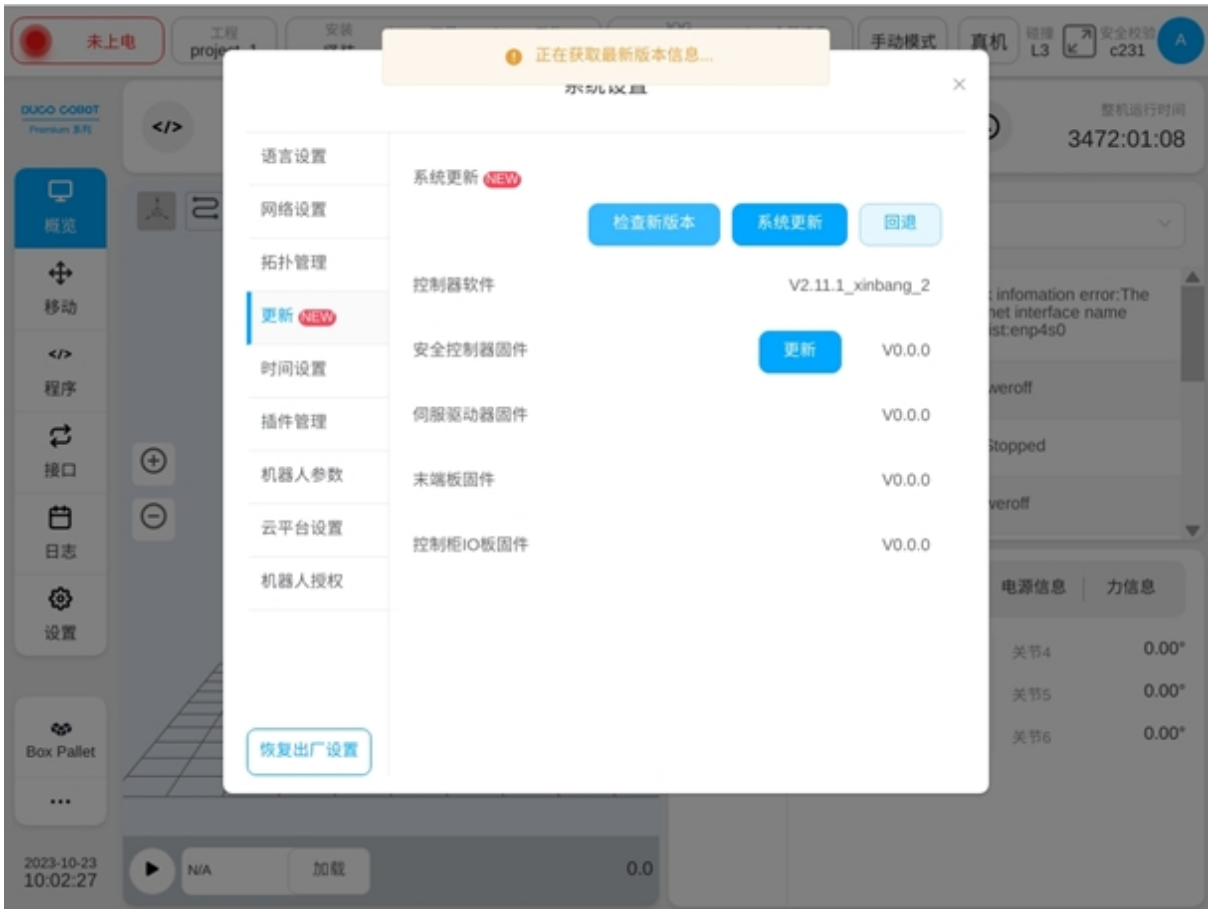
设置 WLAN 时，输入网络名称和密码，单击“设置”按钮后重启即可生效。

2.13.3 更新

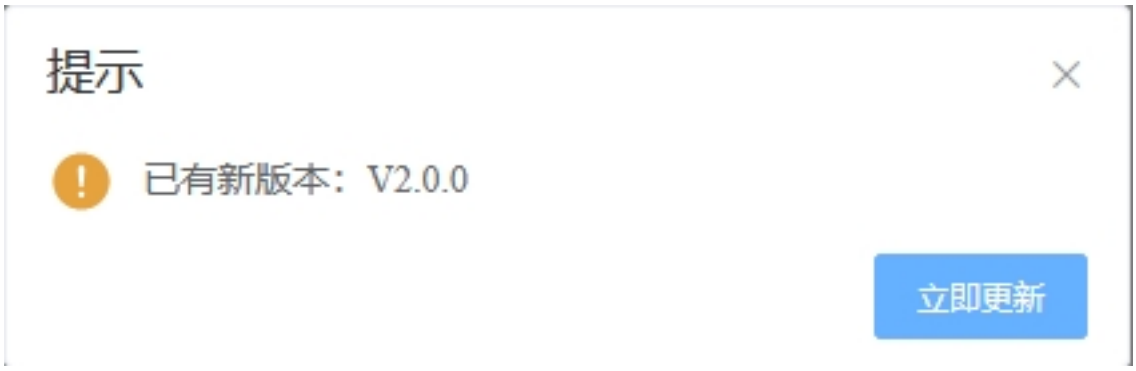
单击“更新”选项卡，会显示系统更新相关操作按钮，即“检查新版本”、“系统更新”、“回退”按钮,、及控制器软件版本、安全控制器固件版本、伺服控制器固件版本、末端板固件版本、控制柜 IO 板固件版本。其中，安全控制器固件版本有单独“更新”按钮进行更新，系统更新可主动检查系统版本或者本地系统一键更新到某个版本，也可回退到之前的某个版本。

机器人进行系统更新和安全控制器固件版本时，需确保机器人处在断电状态。

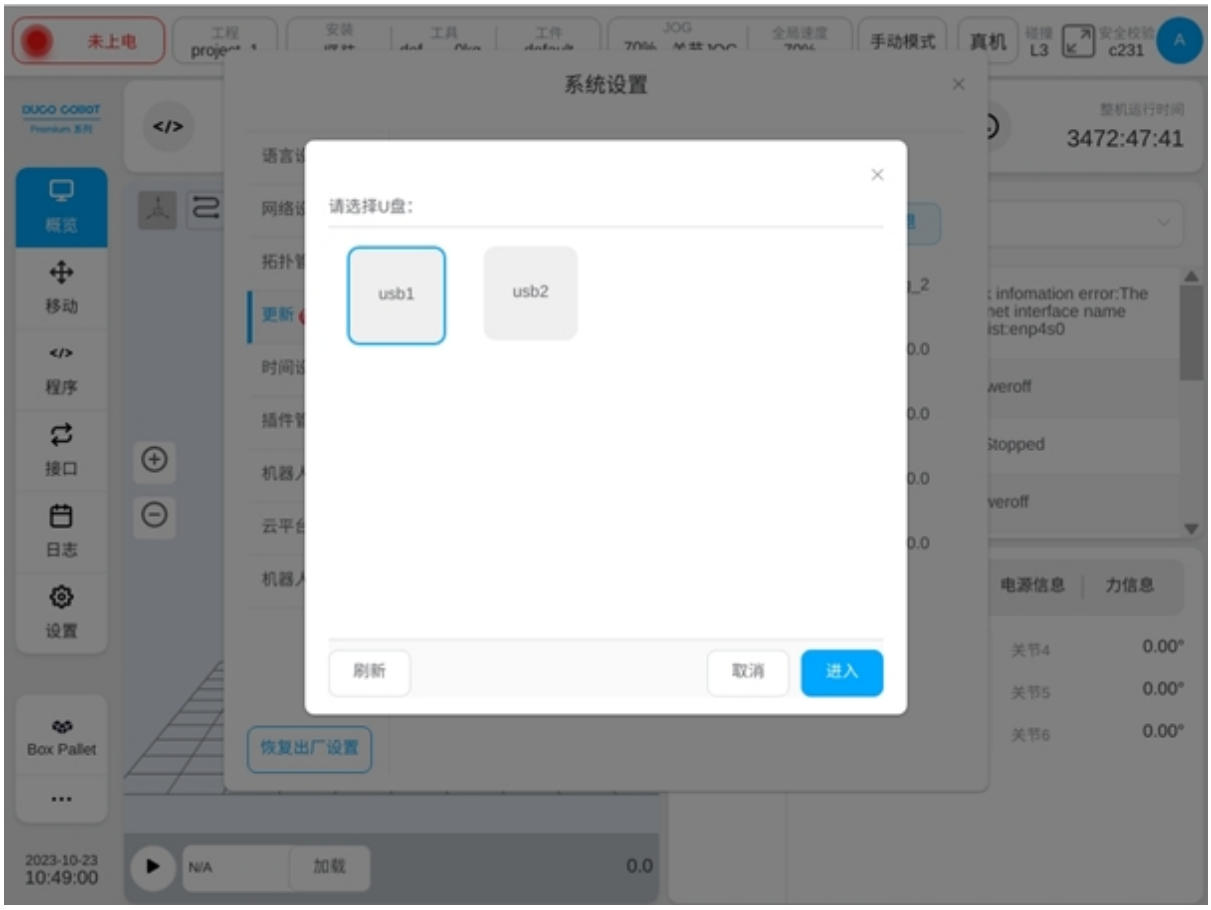
单击“检查新版本”按钮后，界面上方会显示“正在获取最新版本信息……”提示语的弹框。



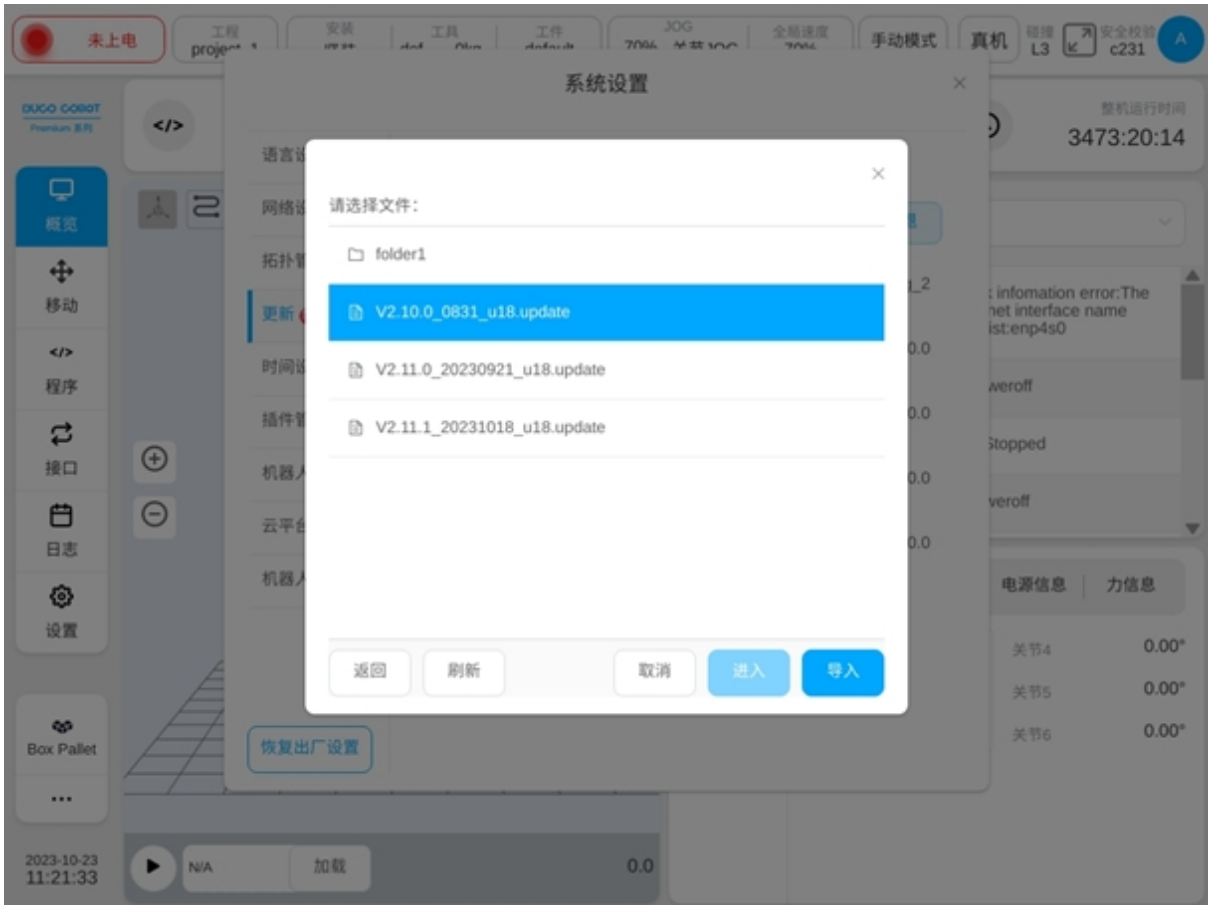
当检查无软件新版本时，界面会显示提示当前软件版本已是最新版本的弹框；当检查有新版本时，界面会显示最新版本的提示框。若要更新到新版本，单击“立即更新”即可。



单击“系统更新”按钮，会弹出“更新过程中机器人会自动上下电，请勿操作机器人!”的提示框，用户单击提示框“确定”按钮后，会显示 U 盘选择的弹窗。



选择好 U 盘后，单击“选择”按钮，会显示出该 U 盘下所有符合对应类型更新的更新包，如系统更新包是以.updatePro 后缀结尾，其中包括了软件更新包（以.update 后缀结尾），伺服固件包，末端固件更新包，IO 板通讯包（.firmware)。软件更新包是以.update 后缀结尾。安全控制器更新包是以.firmware 后缀结尾。

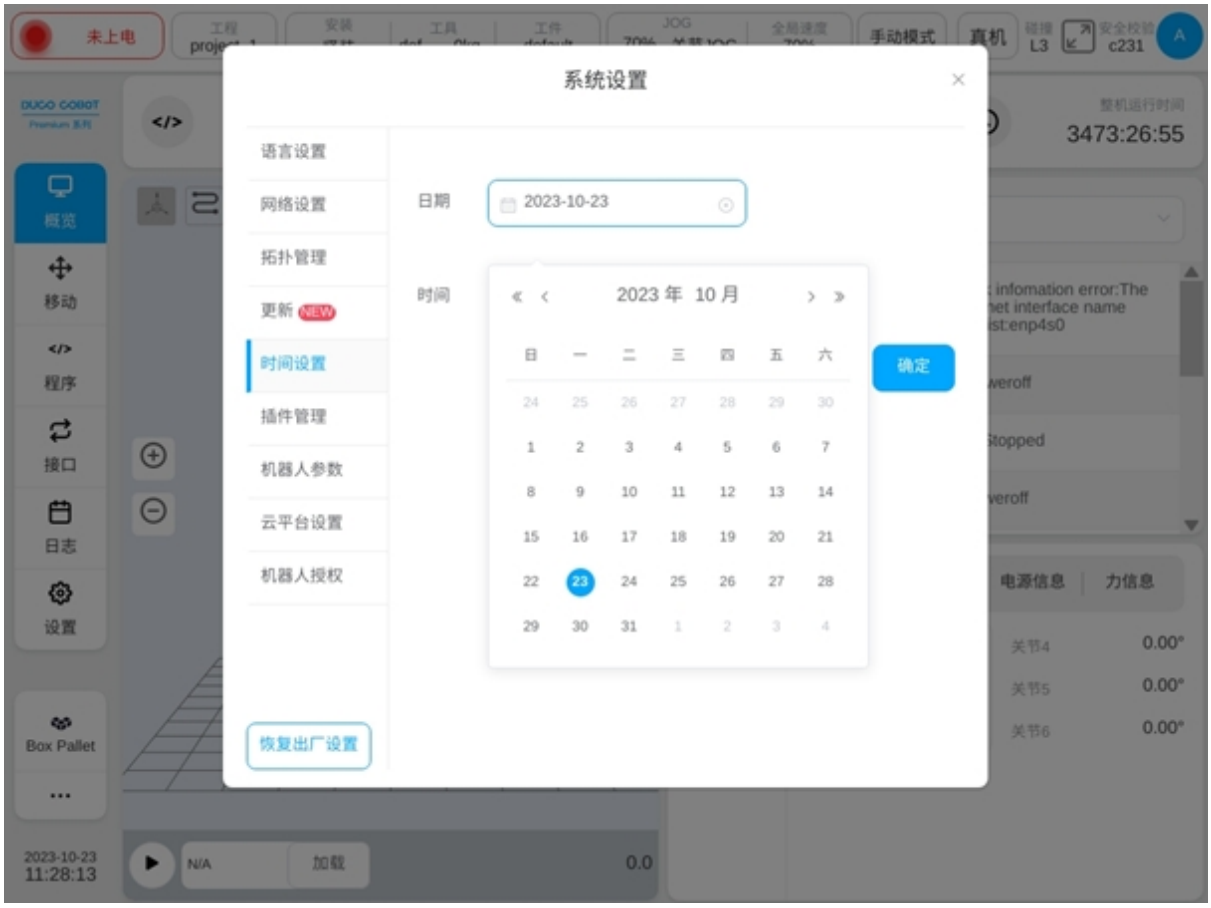


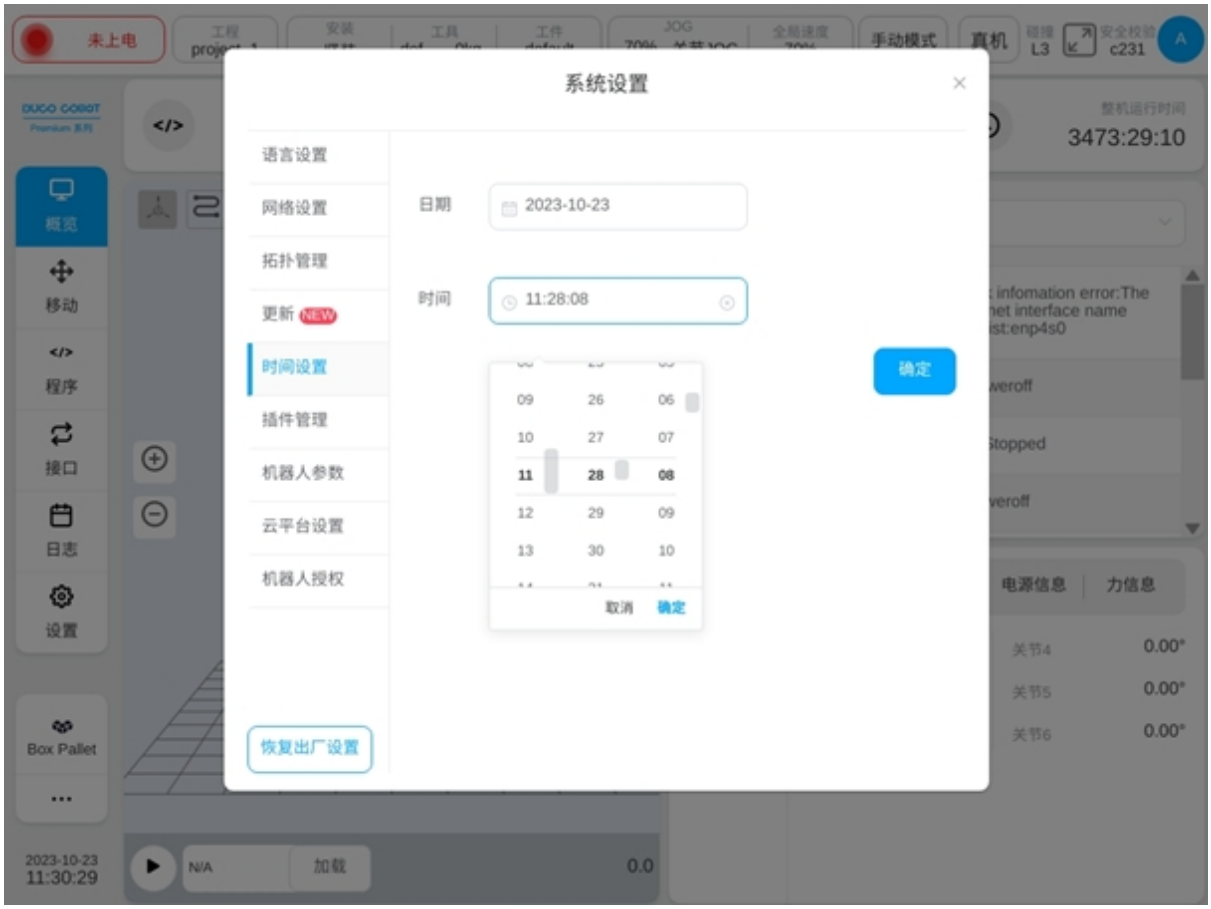
单击“回退”按钮，可以进行机器人系统版本回退操作，会显示系统回退文件列表界面。系统版本回退文件是以.updatePro 后缀结尾的。若只需要回退软件版本，则选择软件回退文件，以.update 后缀结尾。选择所需文件单击“回退”按钮会提示重启后生效。机器人进行系统版本回退时，需确保机器人处在断电状态。

除了用户可主动查询软件版本，云端可推送新版本到设备。若云端推送新版本，单击用户头像后弹出框里“系统设置”按钮旁会显示字样“NEW”的红色标记。

2.13.4 时间设置

单击“时间设置”选项卡，会显示当前系统的日期和时间。单击日期选择器或时间选择器的选择框，分别可以进行日期和时间的选择，如下图所示：

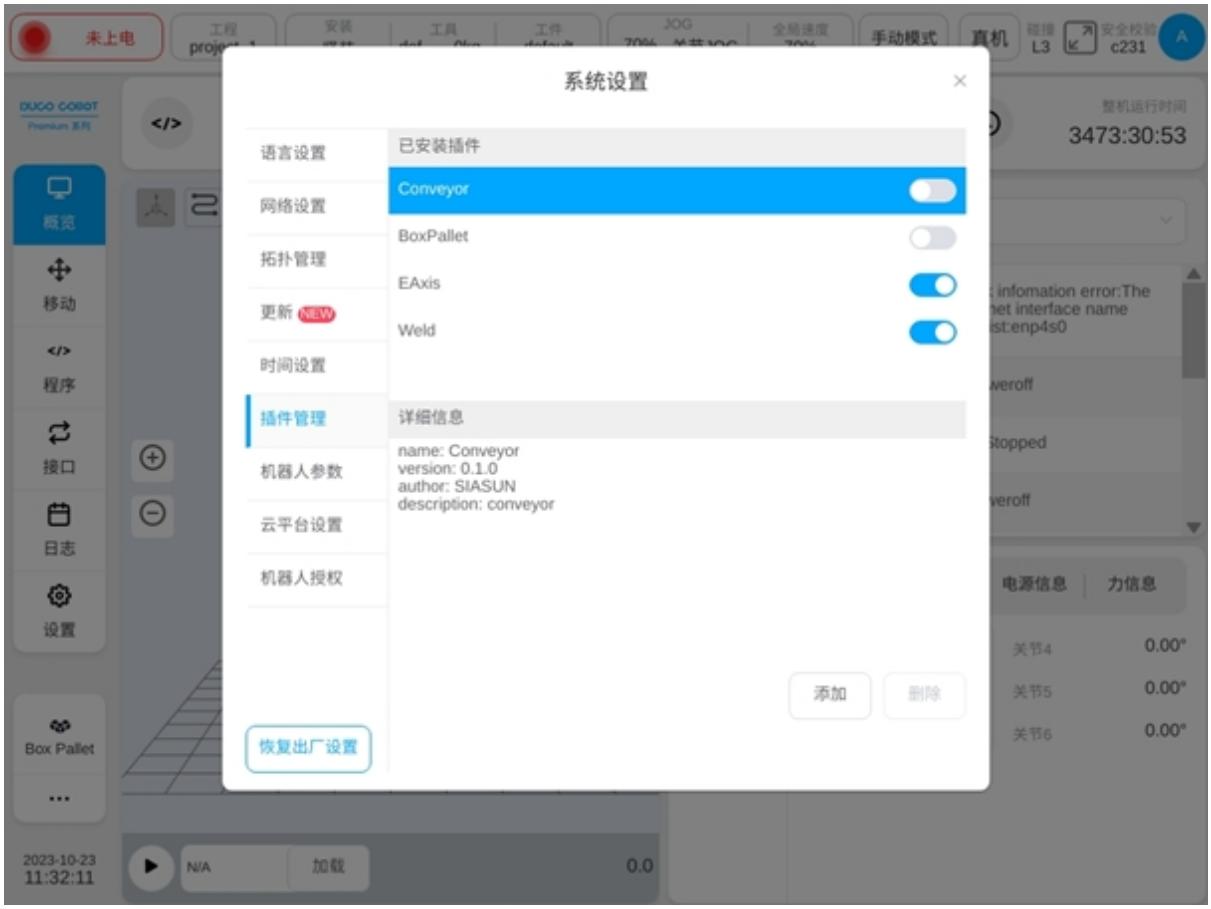

未上电
手动模式
真机
系统设置
语言设置
网络设置
拓扑管理
更新 NEW
时间设置
插件管理
机器人参数
云平台设置
机器人授权
日期
2023-10-23
时间
2023 年 10 月
确定
恢复出厂设置
概览
移动
程序
接口
日志
设置
Box Pallet
3473:26:55
电源信息
力信息
2023-10-23
11:28:13
N/A
加载
0.0


未上电
手动模式
真机
系统设置
语言设置
网络设置
拓扑管理
更新 NEW
时间设置
插件管理
机器人参数
云平台设置
机器人授权
日期
2023-10-23
时间
11:28:08
确定
取消
确定
恢复出厂设置
概览
移动
程序
接口
日志
设置
Box Pallet
3473:29:10
电源信息
力信息
2023-10-23
11:30:29
N/A
加载
0.0

设置好日期和时间后，单击“确定”按钮，即可设置成功。

2.13.5 插件管理

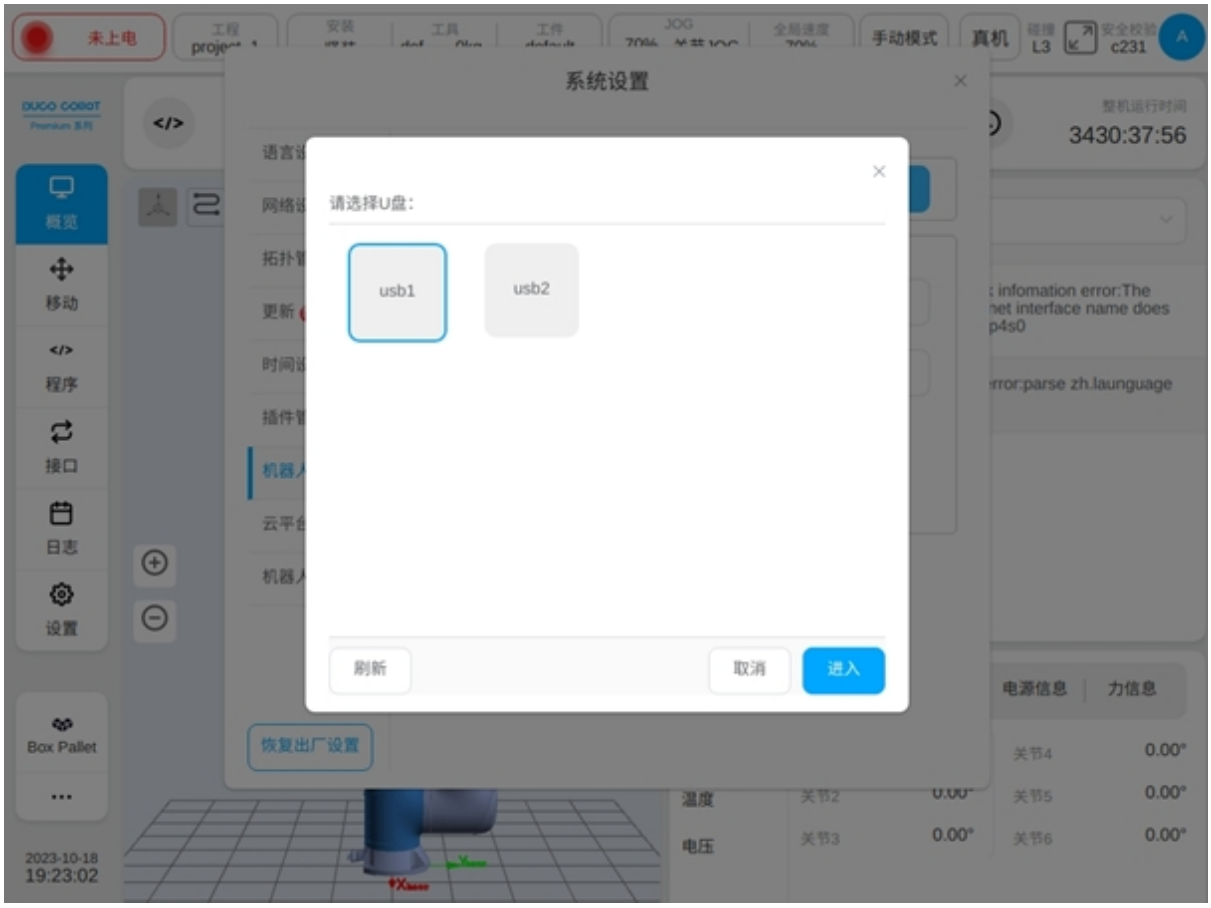
. 单击“插件管理”选项卡，会显示已安装插件和对应插件的详细信息。如下图:



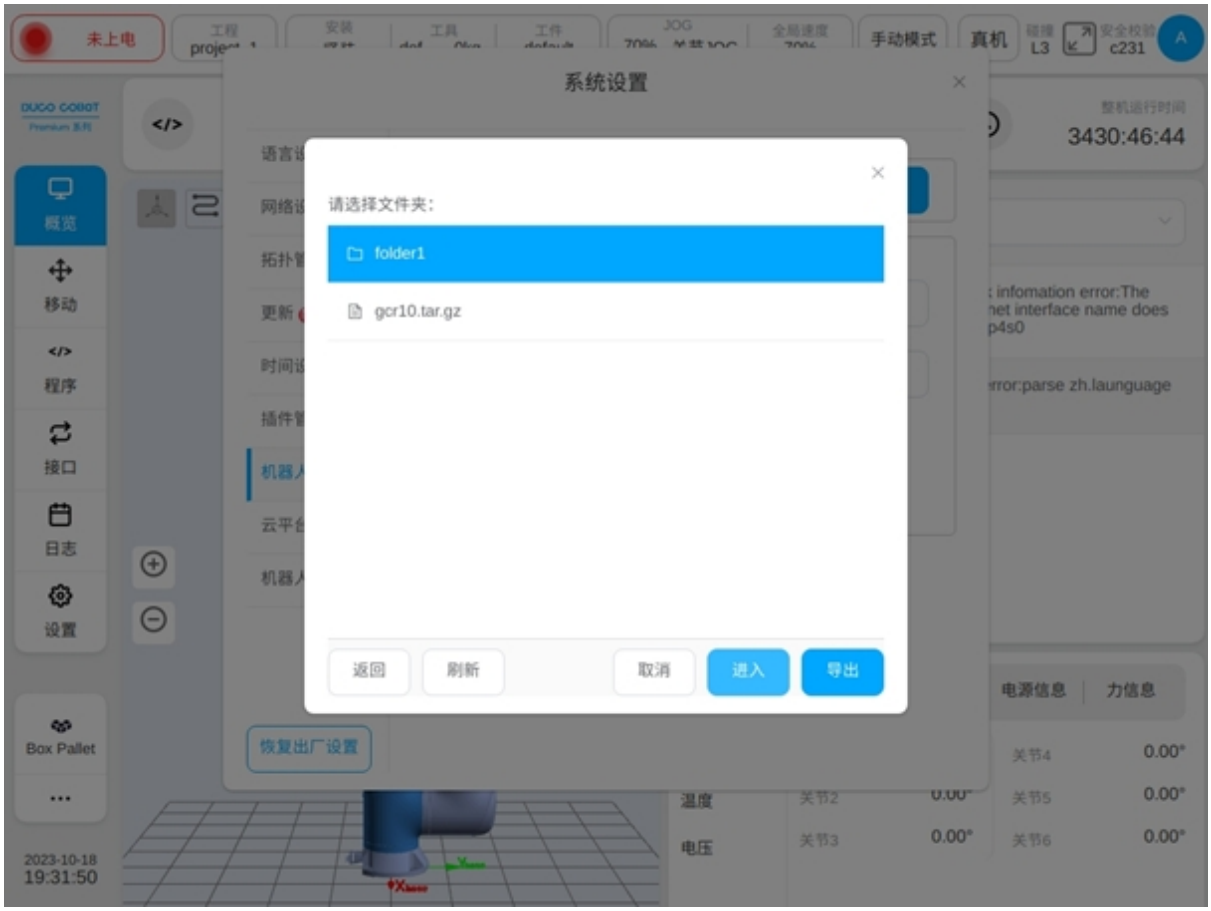
2.13.6 机器人参数

单击“机器人参数”选项卡，可进行机械臂配置文件的导入和导出，以及控制柜参数同步到末端板的操作，如下图:

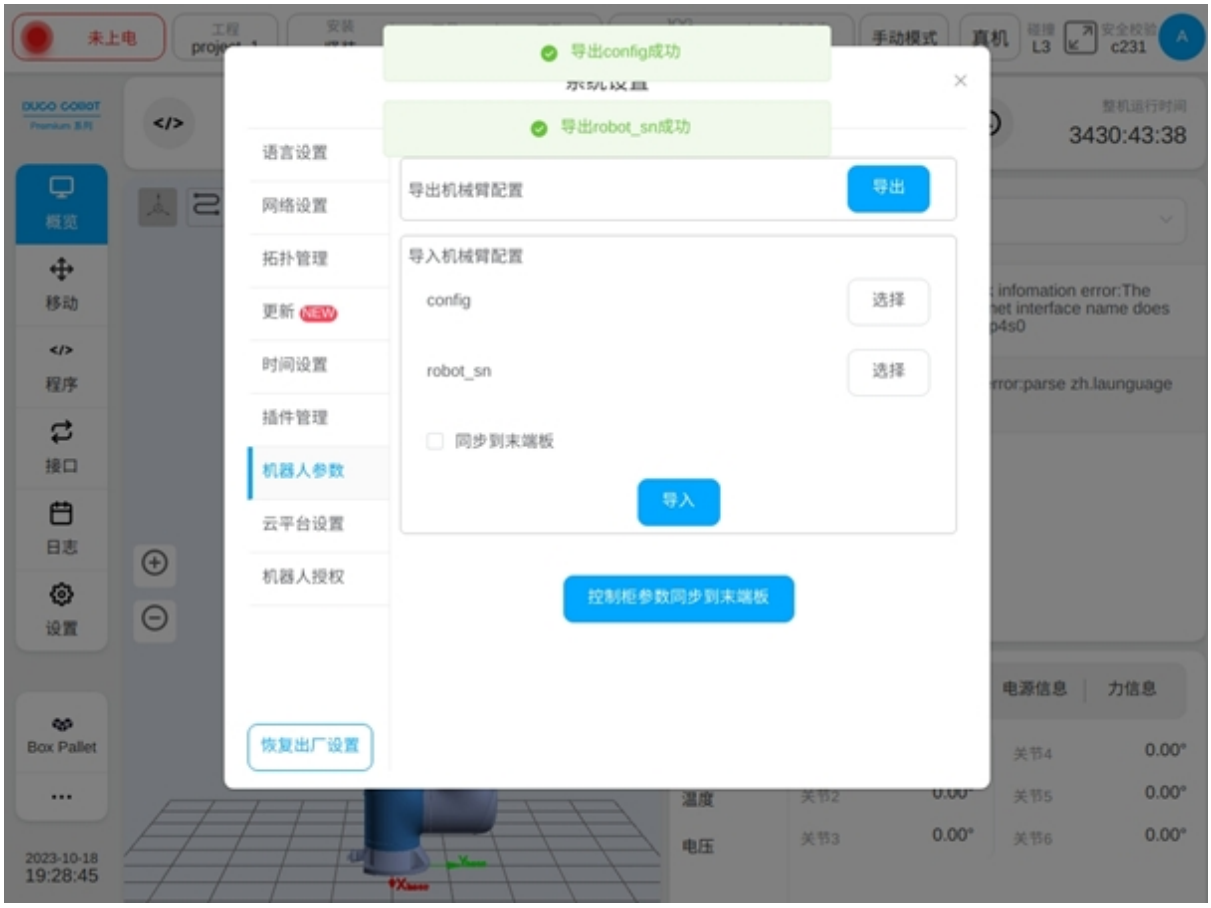
单击“导出”按钮，弹出外部 U 盘导出文件对话框，如下图:



选择好 U 盘（如：选择 U 盘 disk1）及对应 U 盘下文件存放位置（如：选择 U 盘下 disk11 文件夹），单击对话框右下角“选择”按钮即可。



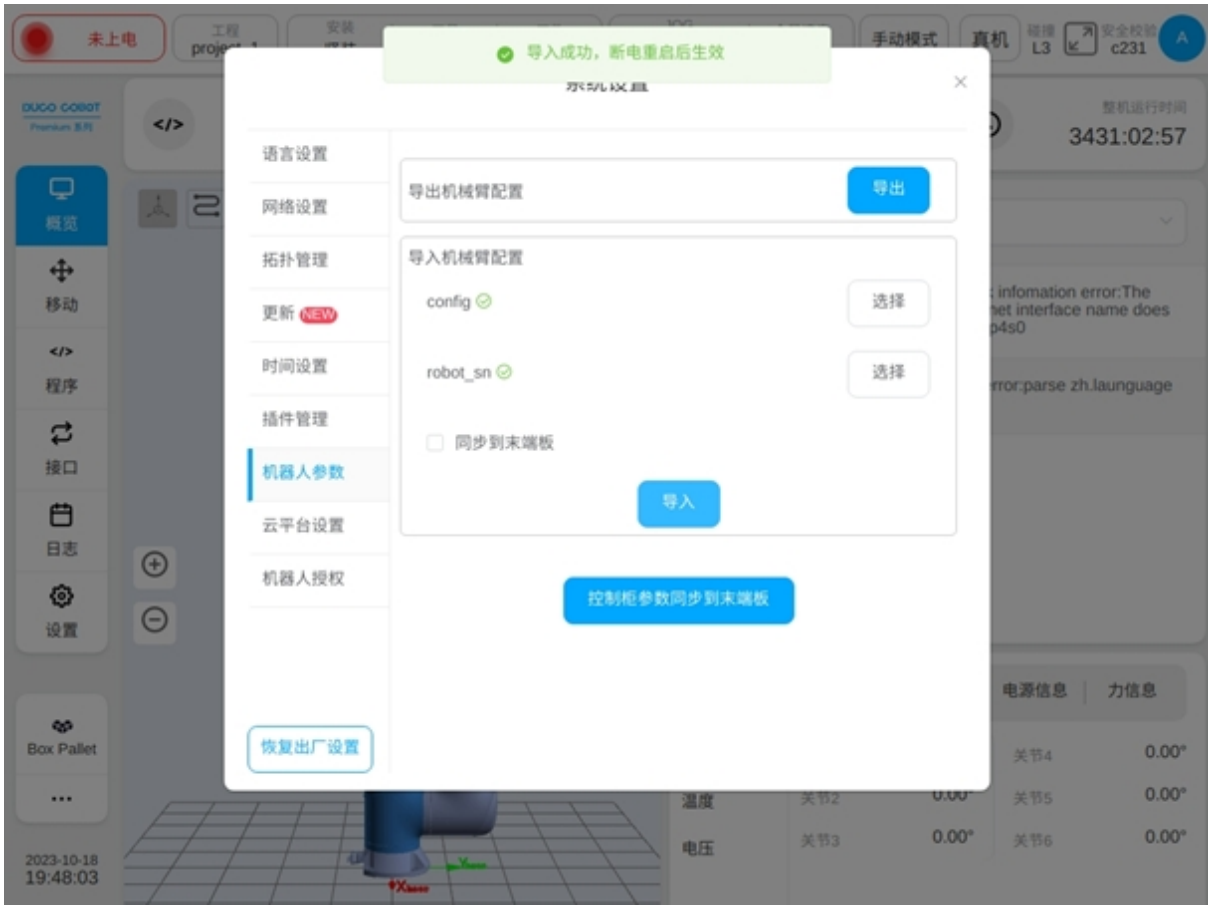
导出成功后，界面上方会依次弹出“导出 `config` 成功”和“导出 `robot_sn` 成功”提示语。



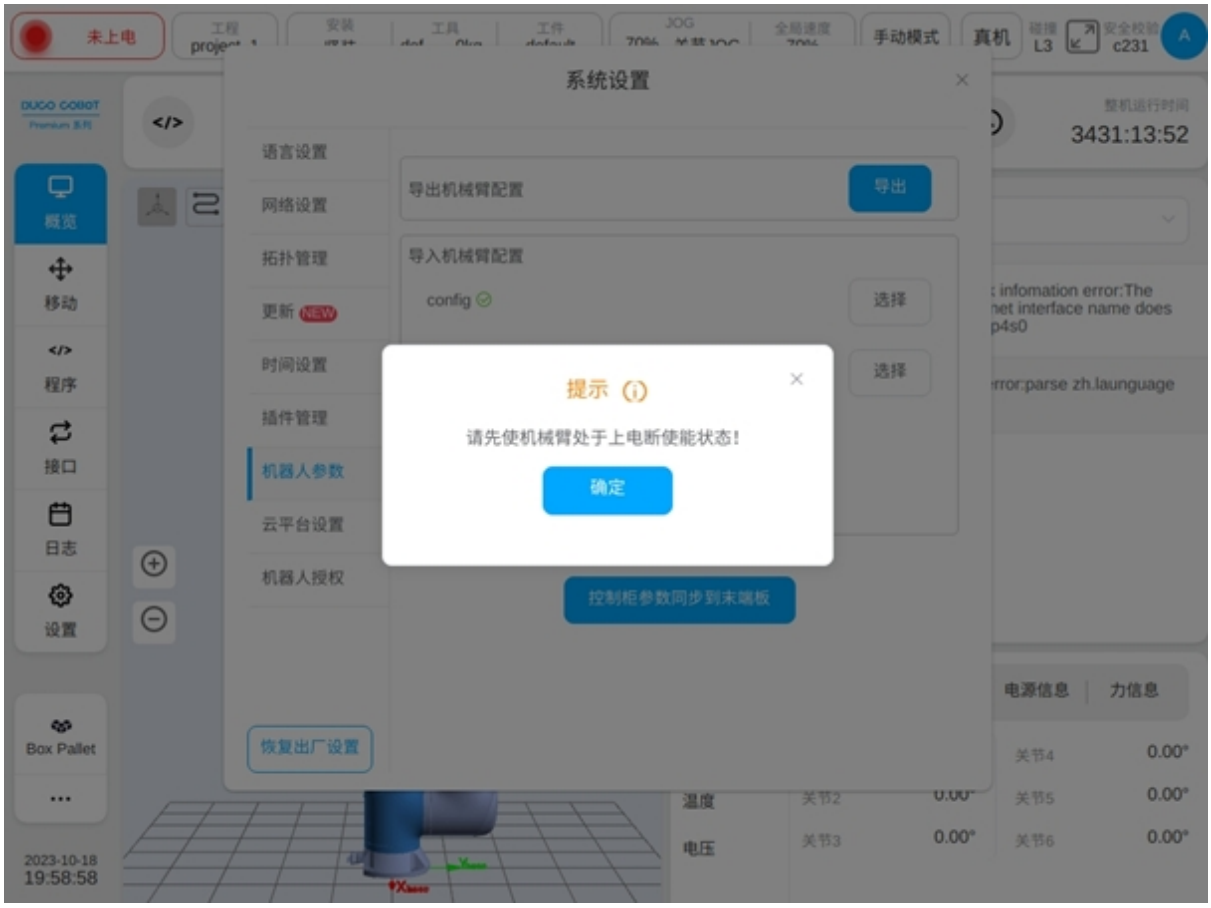
此时，查看刚才所选择 U 盘对应文件夹路径下，会有以机械臂型号命名的.tar.gz 后缀文件（如：gcr5.tar.gz）和 robot_sn.json 文件。

分别单击导入机械臂配置部分里 config 和 robot_sn 后“选择”按钮，从弹出的外部 U 盘导入文件对话框中分别选择好所需 config 文件和 robot_sn 文件，对应 config 和 robot_sn 后会显示绿色对号图标，如下图：

选择好导入的机械臂配置文件后，单击界面“导入”按钮即可。导入成功后，界面上方会显示“导入成功，断电重启后生效”提示语，如下图:



点击“控制柜参数同步到末端板”按钮，可以将控制柜内的机器人参数，同步到机器人的末端板中。



备注： 整个机器人系统中有两处存放机械臂的参数文件，一处在机器人控制柜内，是当前生效的机械臂意数；一处在机械臂末端板内，作为机械臂参数备份。

机器人系统上电时，将校验这两份参数的一致性，只有当两份参数文件一致，才允许正常使能机械臂。若不一致，则需要同步参数。

通常认为机械臂末端板内的参数是本机械臂绝对正确的参数。若由于某些原因需要修改该参数文件，则需要按上述步骤，先将改动后的机械臂参数文件”config" 与"robot_sn" 导入到控制柜内，再通过“控制柜参数同步到末端板”功能，将参数同步到机械臂末端板。

2.13.7 云平台设置

单击“云平台设置”选项卡，会显示机器人与云平台连接状态，可进行云平台 IP 地址输入，以及是否与云平台连接设置。如下图:

当用户手动输入云平台服务器 IP 地址后，切换“是否连接”开关，即可主动连接或断开与云端服务器的连接，且连接状态会对应改变。

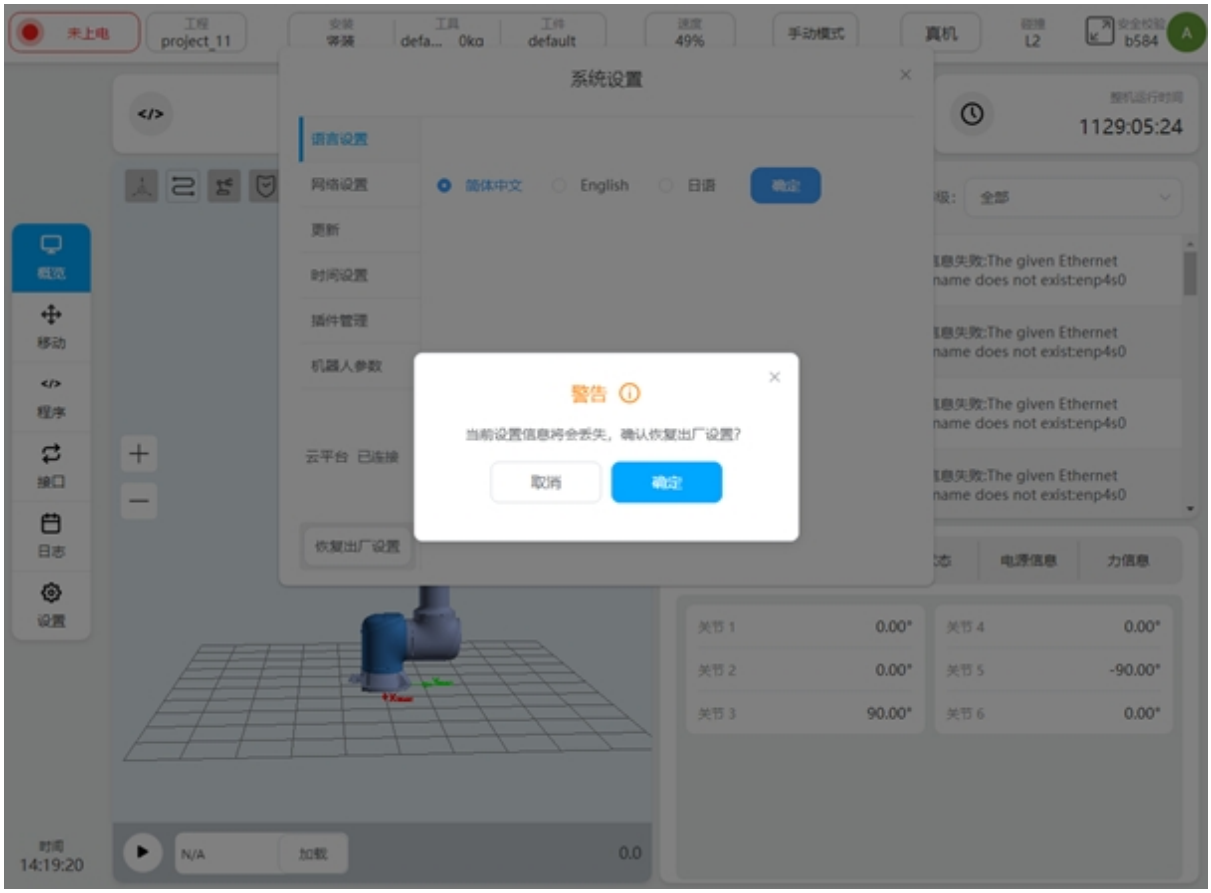
2.13.8 机器人授权

单击“机器人授权”选项卡，会显示机器人授权有关信息和操作按钮，点击“生成动态码”按钮，会生成一个动态码，并提示用户“请将该动态码发送给经销商，获取授权文件，并在有效时间内激活”显示如下图：

通过动态码从经销商那里获取到授权文件后，点击“导入授权文件”按钮，将其导入激活即可。

2.13.9 恢复出厂设置

单击系统设置页面左侧导航选项卡下方的“恢复出厂设置”按钮（仅限 admin 权限用户），弹出警告框如下图：



单击“取消”按钮，即取消恢复出厂设置；单击“确定”按钮后，界面上方会显示“恢复出厂设置成功，需重启生效!”弹框。恢复出厂设置后，会清空如下内容：`project` 中所有非 `template` 工程、所有 `logs` 日志、`robot_sn` 文件以及系统文件下的 `project` 和默认的 `IP` 地址。

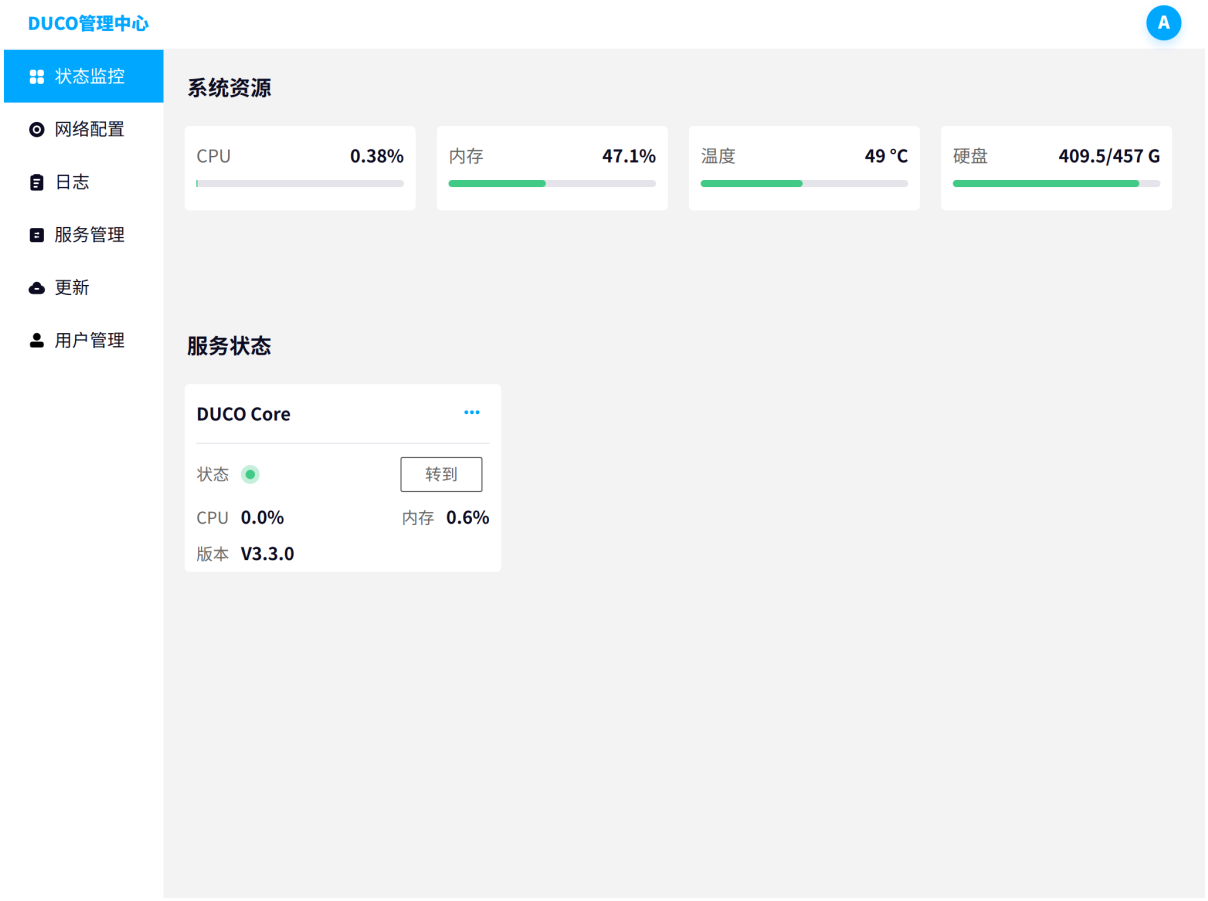
警告： 在使用该功能时，需谨慎操作，以免造成重要数据的丢失!

CHAPTER 3

管理中心

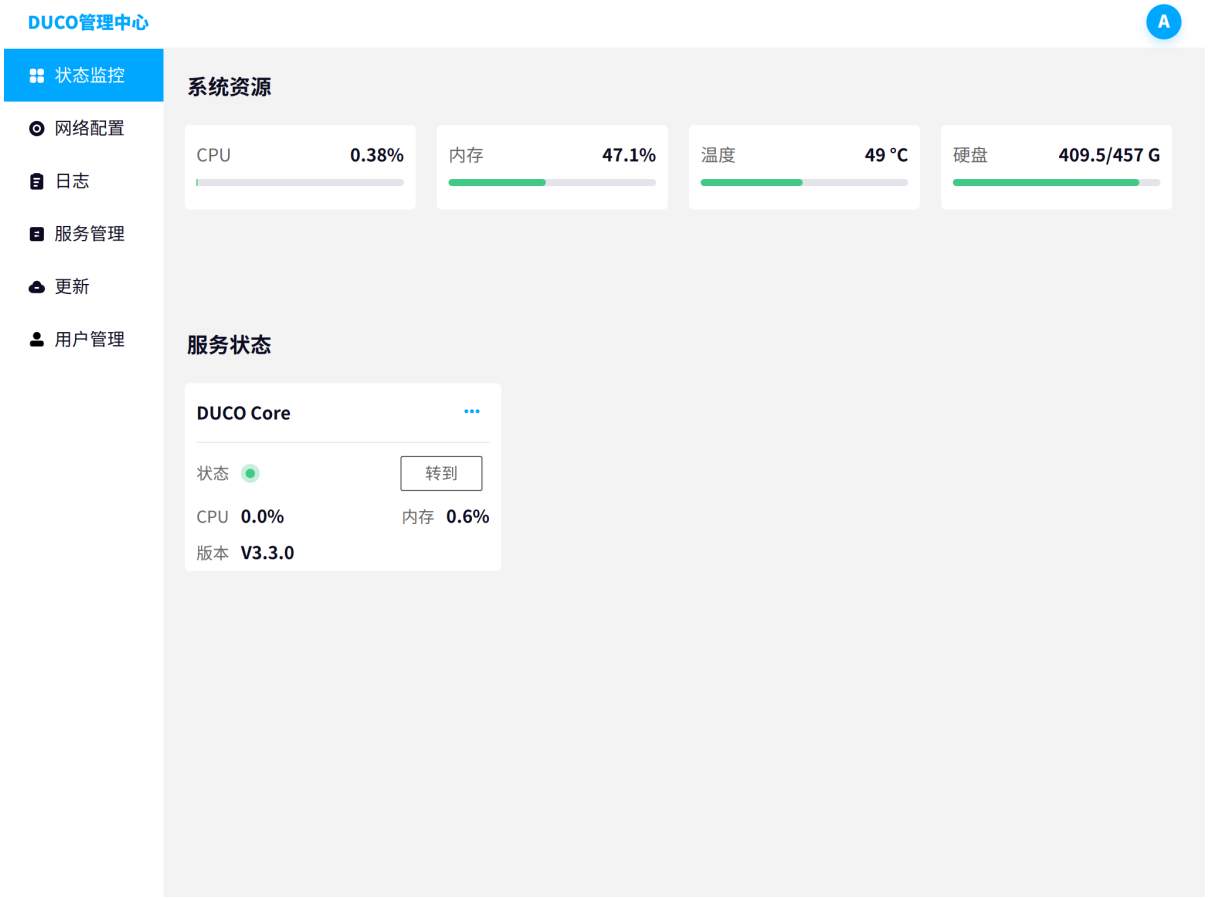
3.1 管理中心

点击管理中心，进入控制器管理中心。



3.1.1 管理中心导航菜单介绍

由状态监控、网络配置、日志、服务管理、更新、用户管理组成。



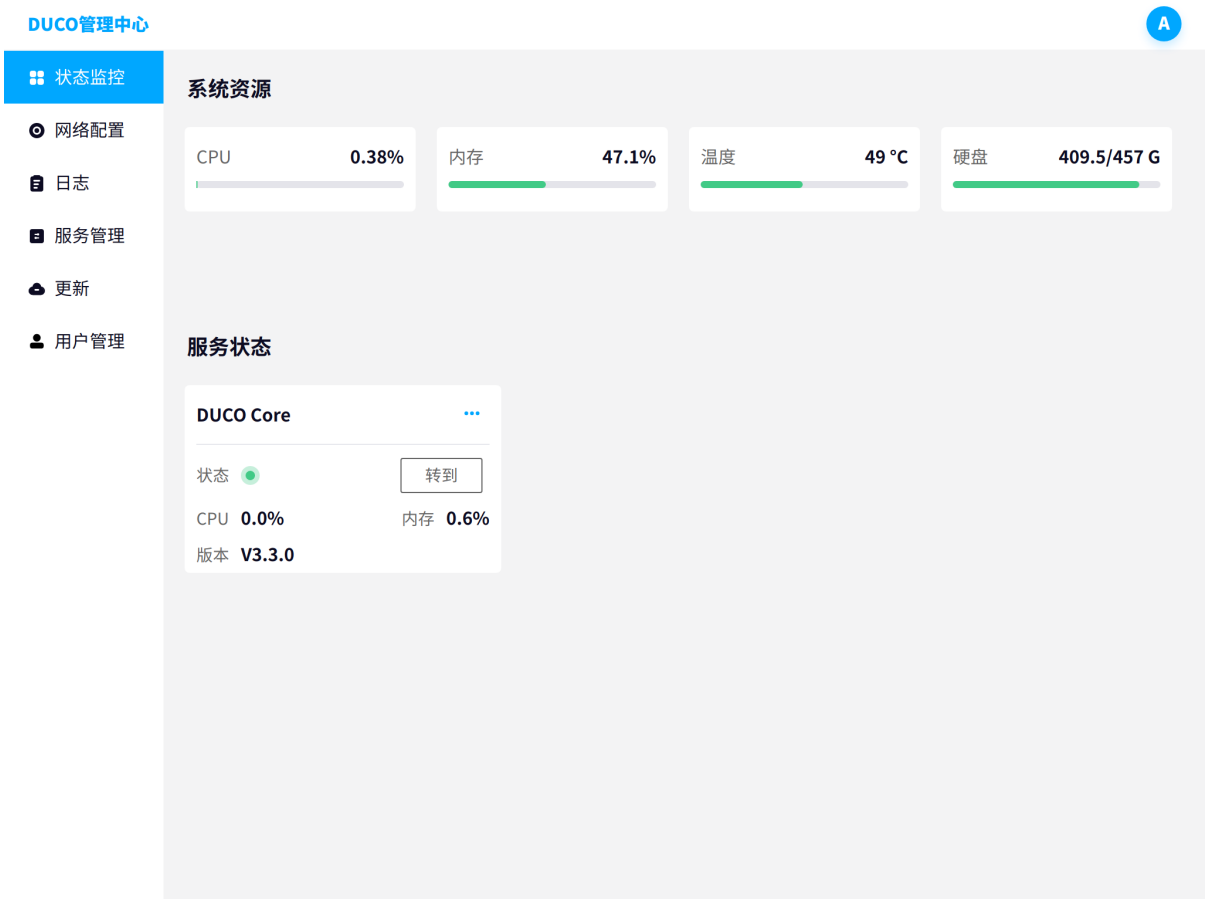
- **状态监控**：系统资源及开通的服务的运行状态的展示。
- **网络配置**：有线网卡及无线网络配置页。
- **日志**：查看管理中心日志及下载历史运行日志。
- **服务管理**：查看控制柜安装的服务。
- **更新**：更新各个服务软件版本。
- **用户管理**：管理用户。

3.2 状态监控

查看系统硬件资源、对 DUCO Core、DUCO Mind 等产品进行服务监控、启停控制等。

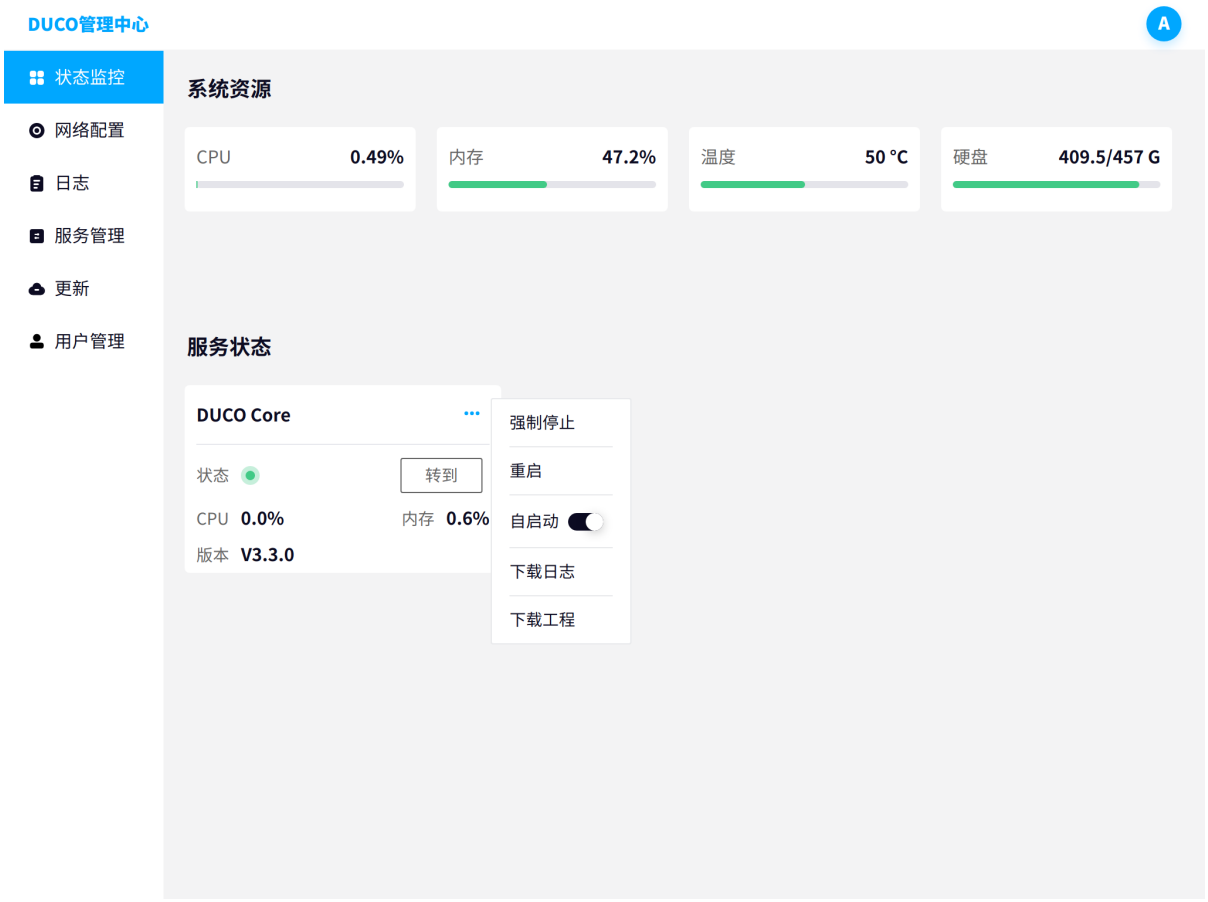
系统资源栏显示当前控制器 CPU 使用率、内存使用率、温度以及硬盘使用率。

服务状态栏展示了控制柜中各个服务的运行状态，所占 CPU、内存信息，版本信息。点击“转到”按钮可以跳转到该服务的界面。



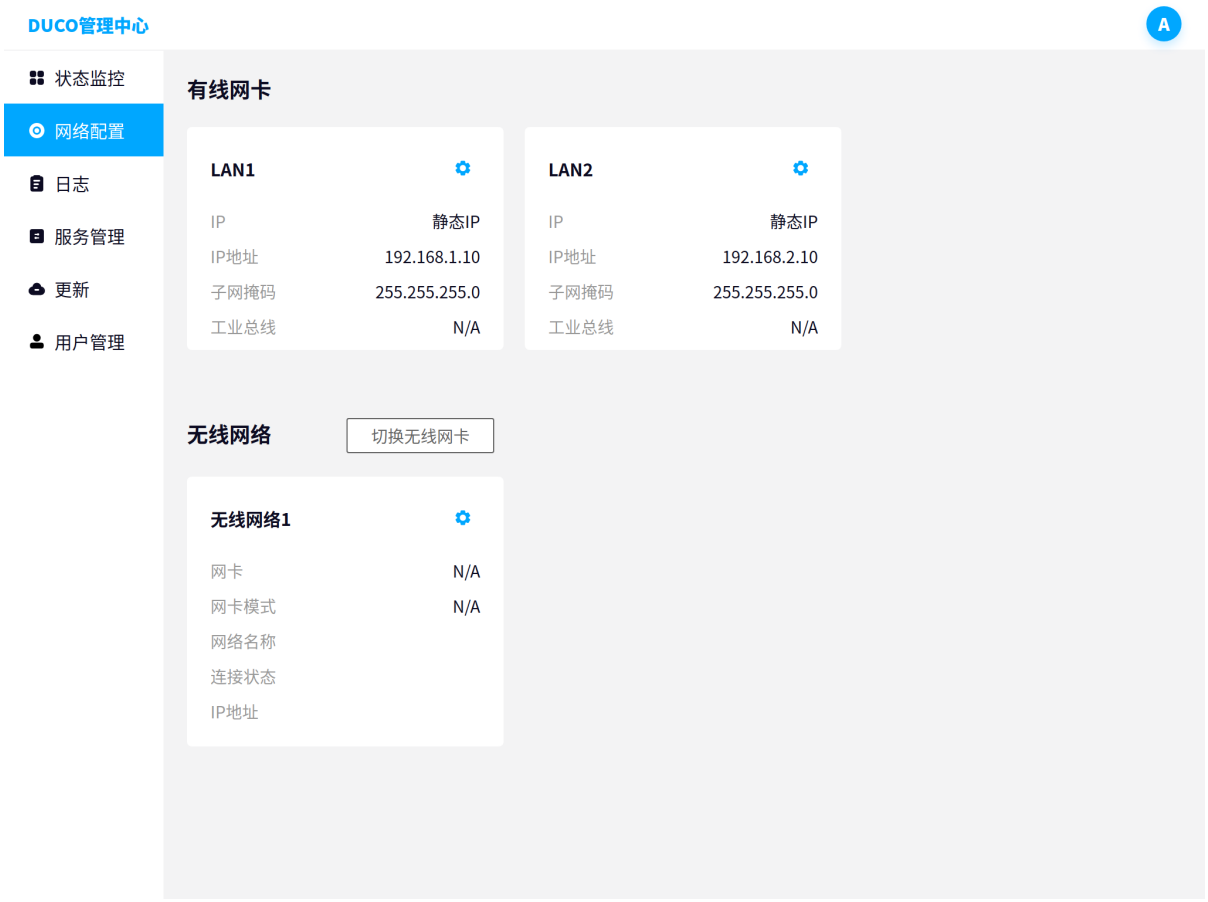
点击服务卡片有上方的按钮，弹出该服务相关的操作，可执行如下操作

- 强制停止：停止该服务
- 重启：重启该服务
- 自启动：配置该服务是否开机自启动
- 下载日志：弹出日志对话框，可以下载或者导出该服务的日志
- 下载工程：弹出工程对话框，可以下载或者导出该服务的工程



3.3 网络配置

配置控制柜的有线和无线网络



3.3.1 有线网卡配置

网卡配置弹框中可修改 IP 模式：静态 IP、DHCP 或本地链路连接状态，设置 IP 地址、子网掩码。

3.3.2 无线网络配置

将官方配置的 usb 无线模块插入控制柜中任意 usb 口。

点击切换无线网卡，选中扫描到的无线模块，点击确认。

AP 热点模式配置

点击无线网络右边的配置按钮，模式选择 AP 热点，设置网络名称和密码，点击确认。

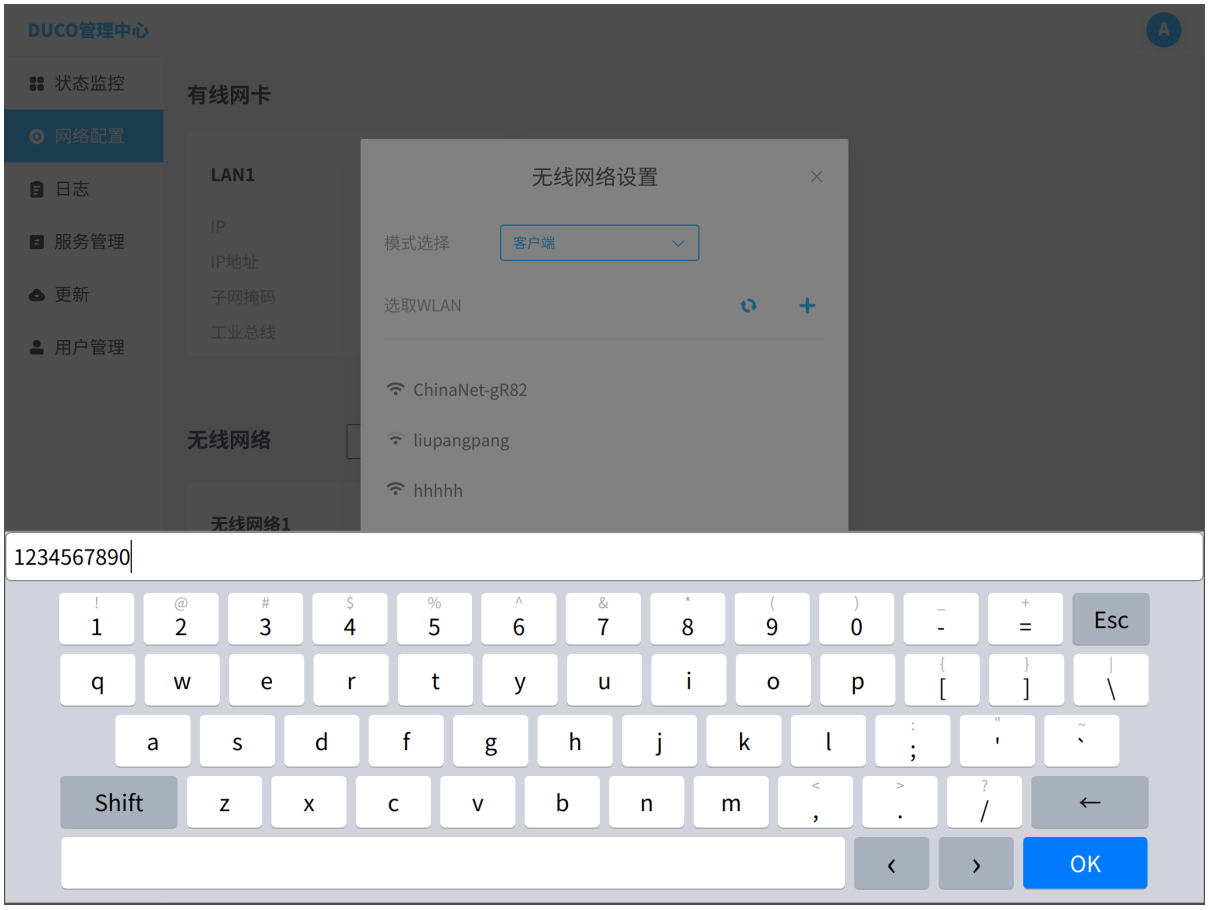
外部设备搜索设置的网络名称，输入密码。

外部设备连接无线后，通过 192.168.117.66:7000 登录机器人，通过 192.168.117.66:7200 登录管理中心。

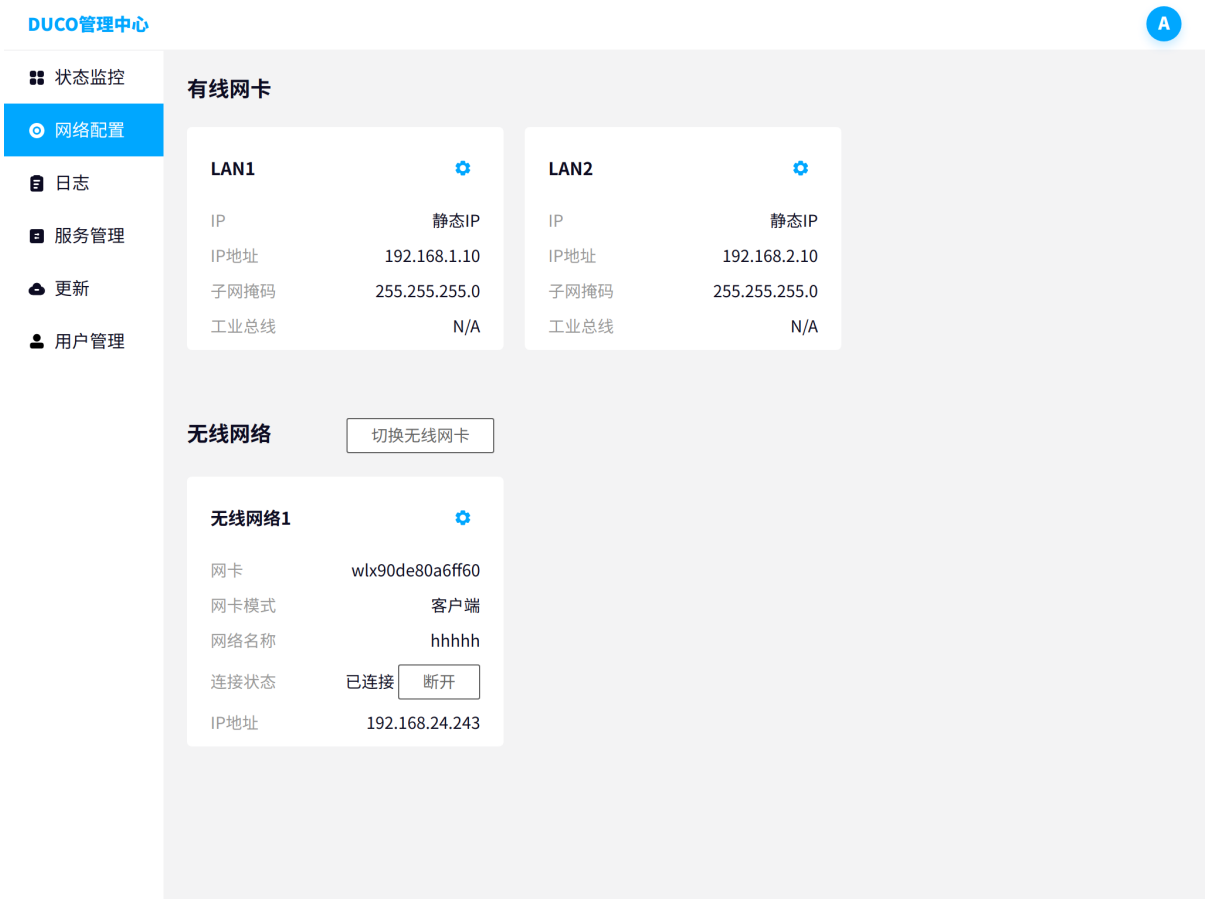
客户端模式配置

点击无线网络右边的配置按钮，模式选择客户端，点击扫描按钮，扫描无线网络。

选中连接的无线网络，输入密码点击确定，连接外部设备。

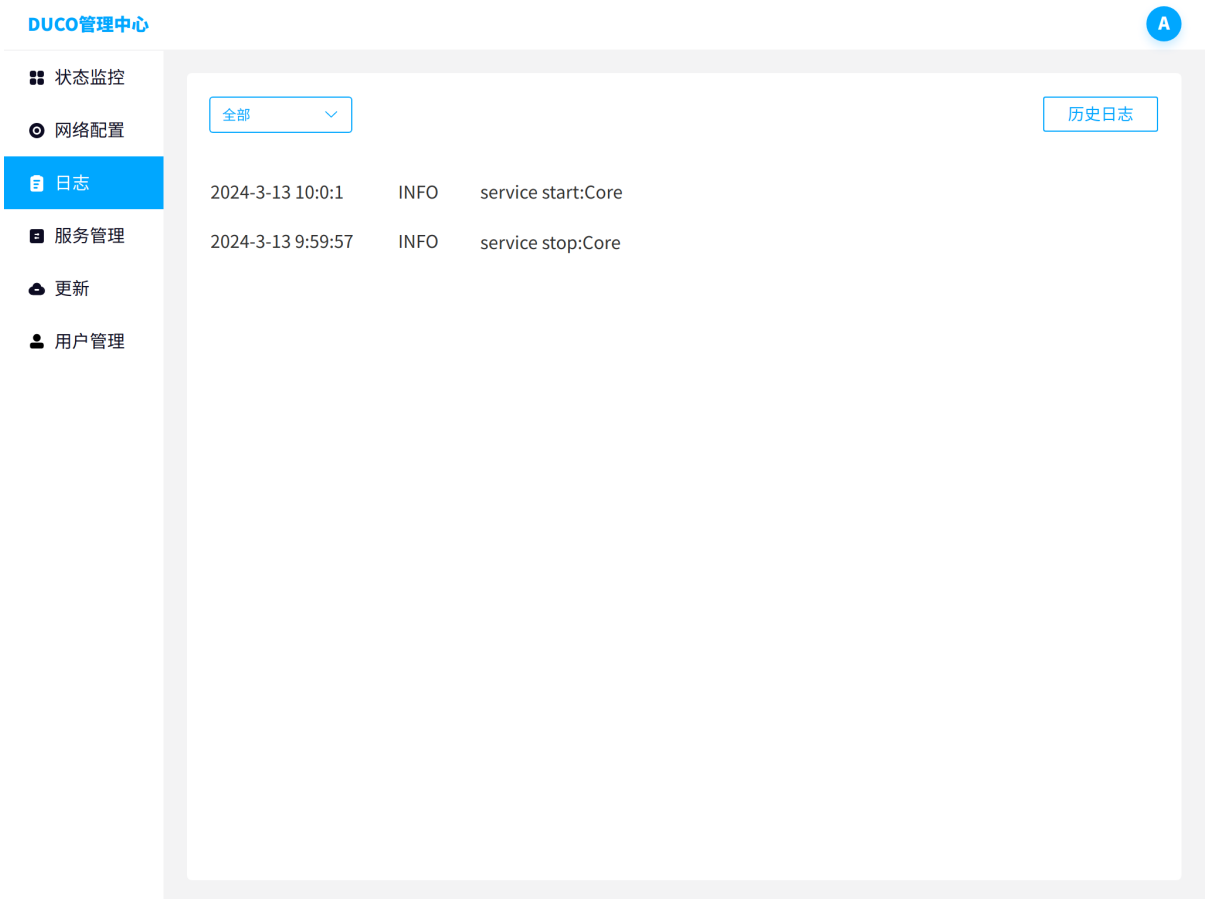


外部设备通过显示的 IP 地址（以实际显示为准）192.168.24.243:7000 登录机器人，通过 192.168.24.243:7200 登录管理中心



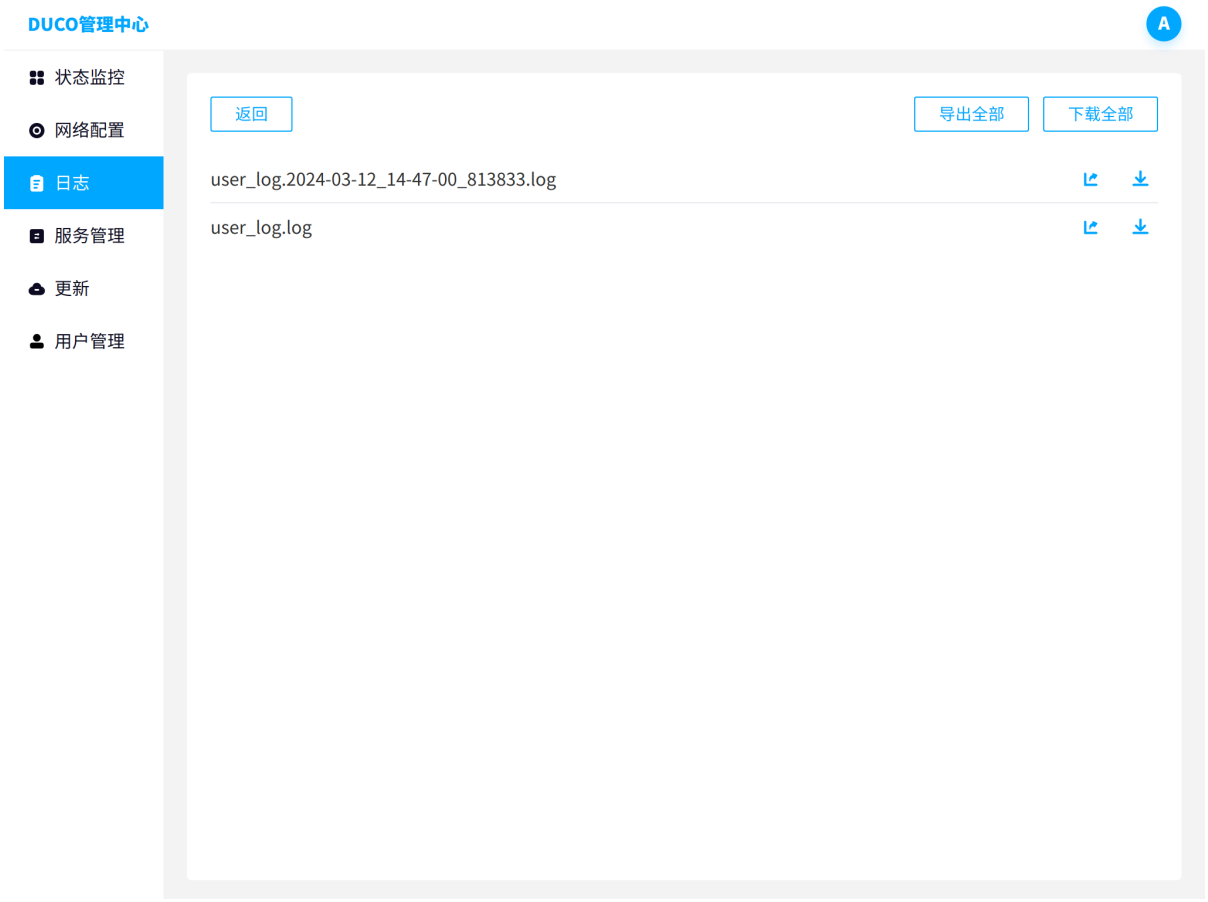
3.4 日志

查看 Manager 实时日志。可对报错进行分类查看，如致命，错误，警告，信息等。右上角历史日志按钮，点击可进入查看历史日志。



3.4.1 历史日志

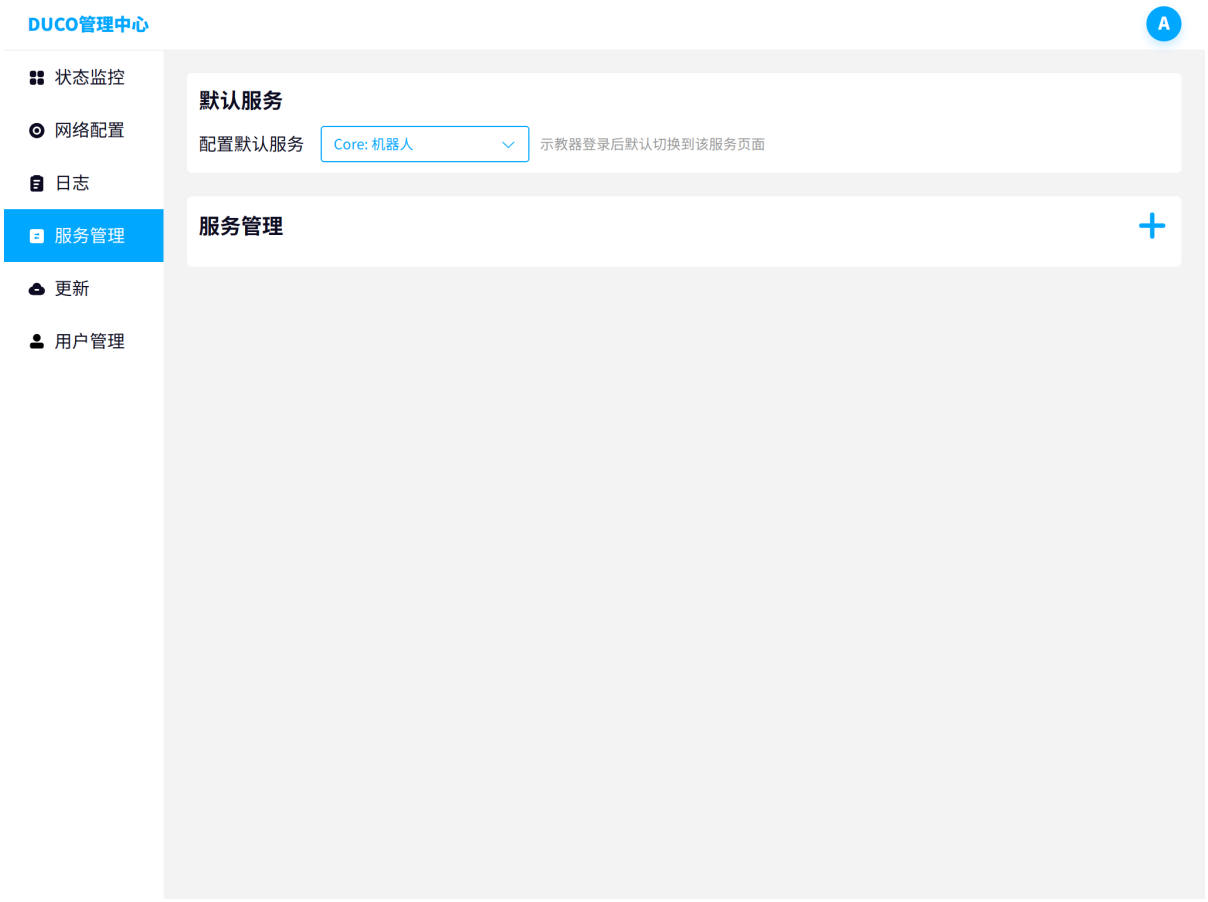
在历史日志页。用户可以批量导出日志到 U 盘或下载日志到本地电脑查看，方便用户追溯错误源。点击返回按钮返回日志首页。

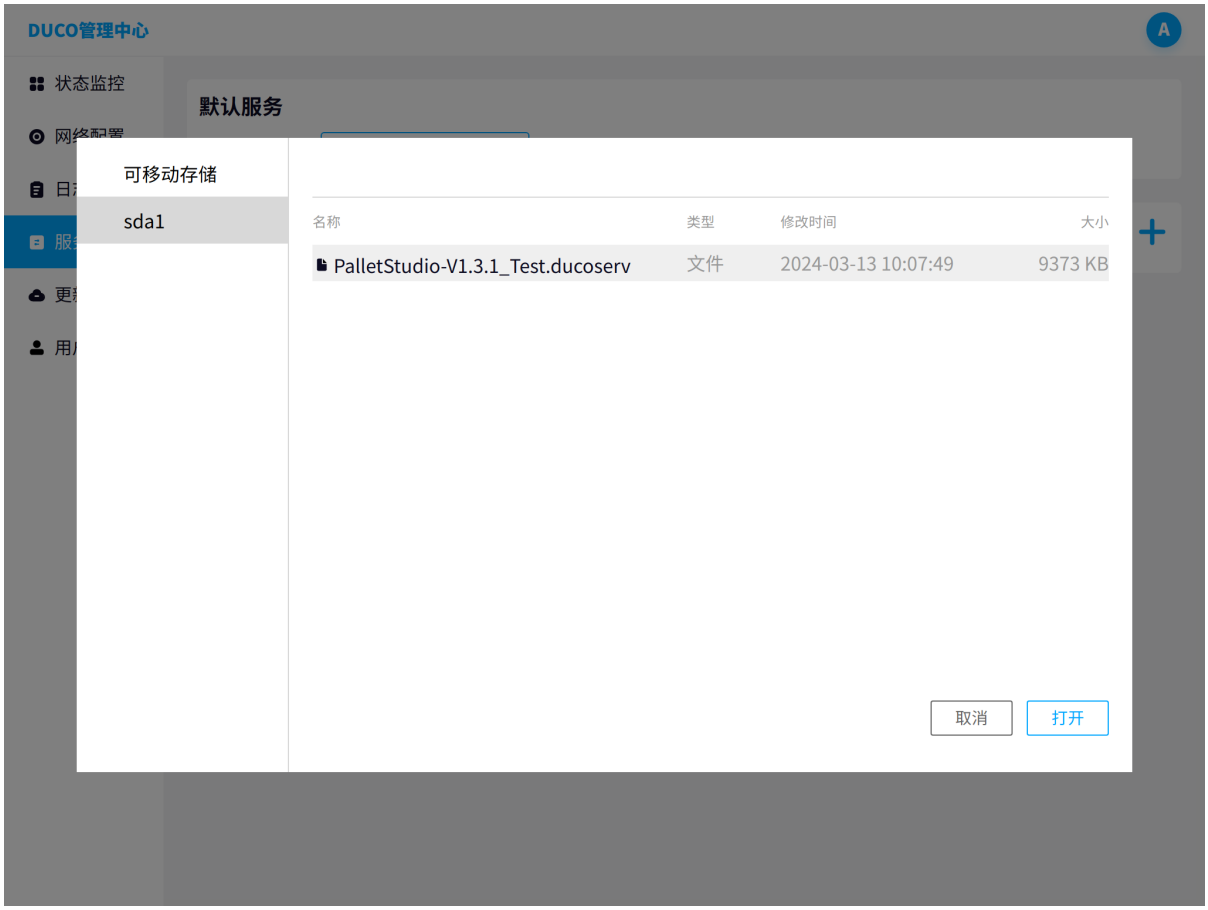


3.5 服务管理

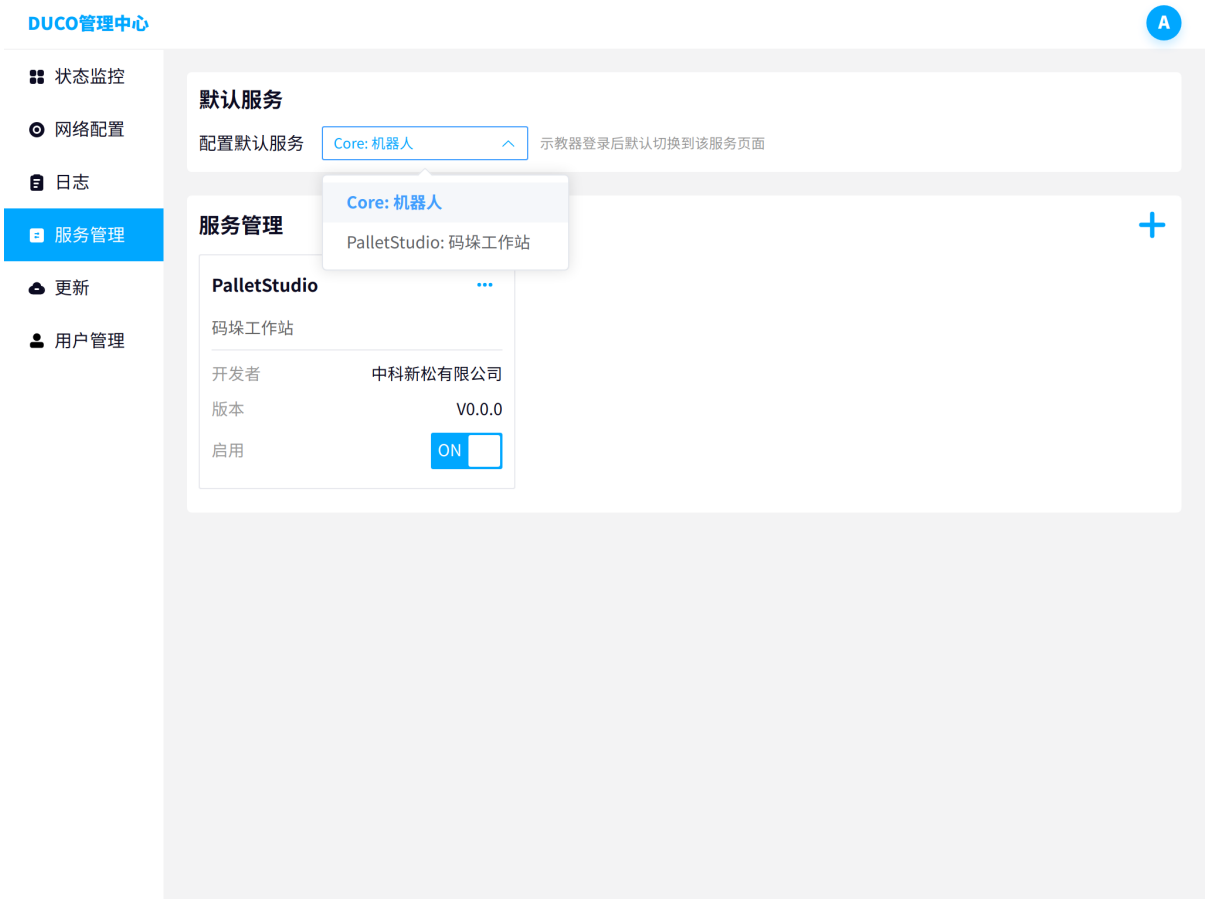
对系统服务进行管理，包括安装、卸载、配置默认服务。

点击加号安装新增的服务

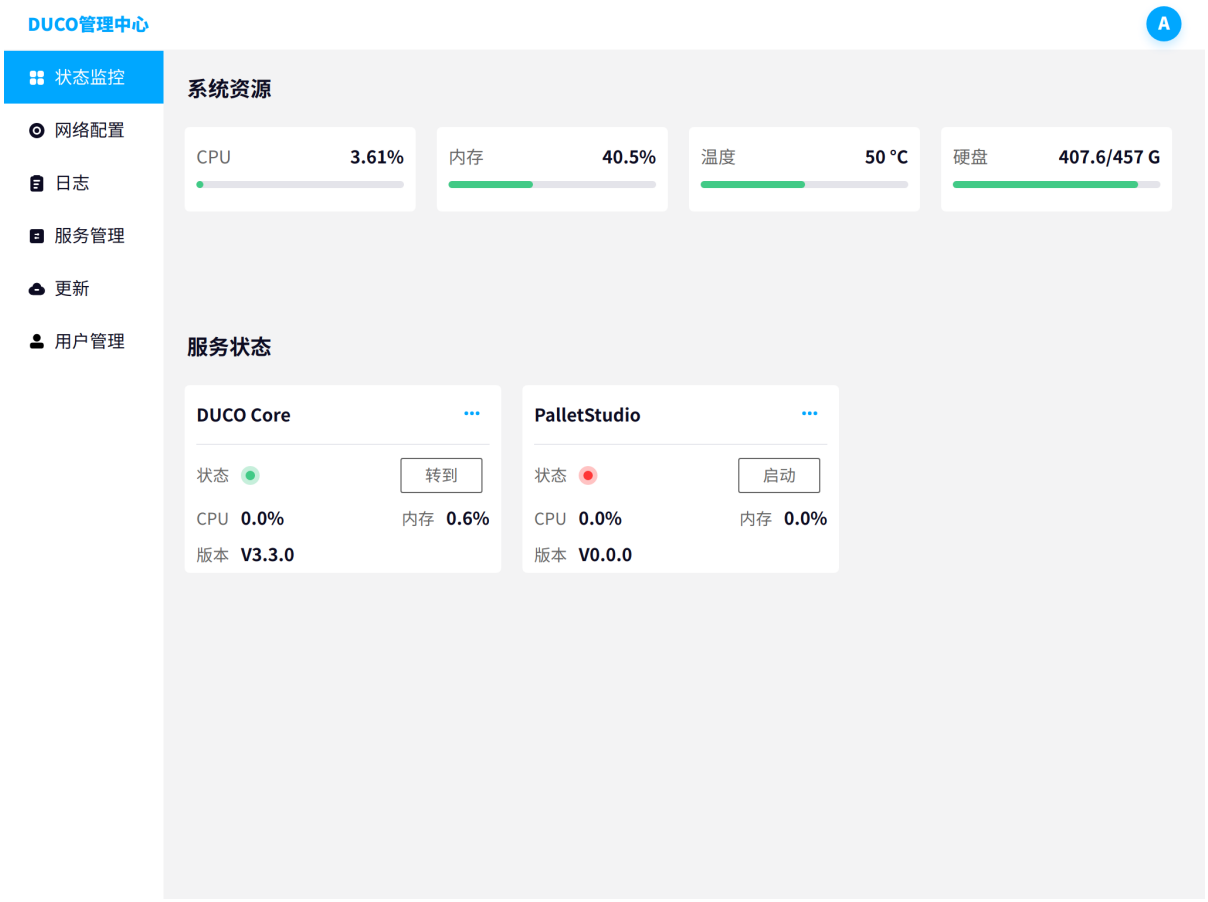

DUCO管理中心
A
状态监控
网络配置
日志
服务管理
更新
用户管理
默认服务
配置默认服务
Core: 机器人
示教器登录后默认切换到该服务页面
服务管理


DUCO管理中心
A
状态监控
默认服务
可移动存储
sda1
名称
类型
修改时间
大小
PalletStudio-V1.3.1_Test.ducoserv
文件
2024-03-13 10:07:49
9373 KB
取消
打开

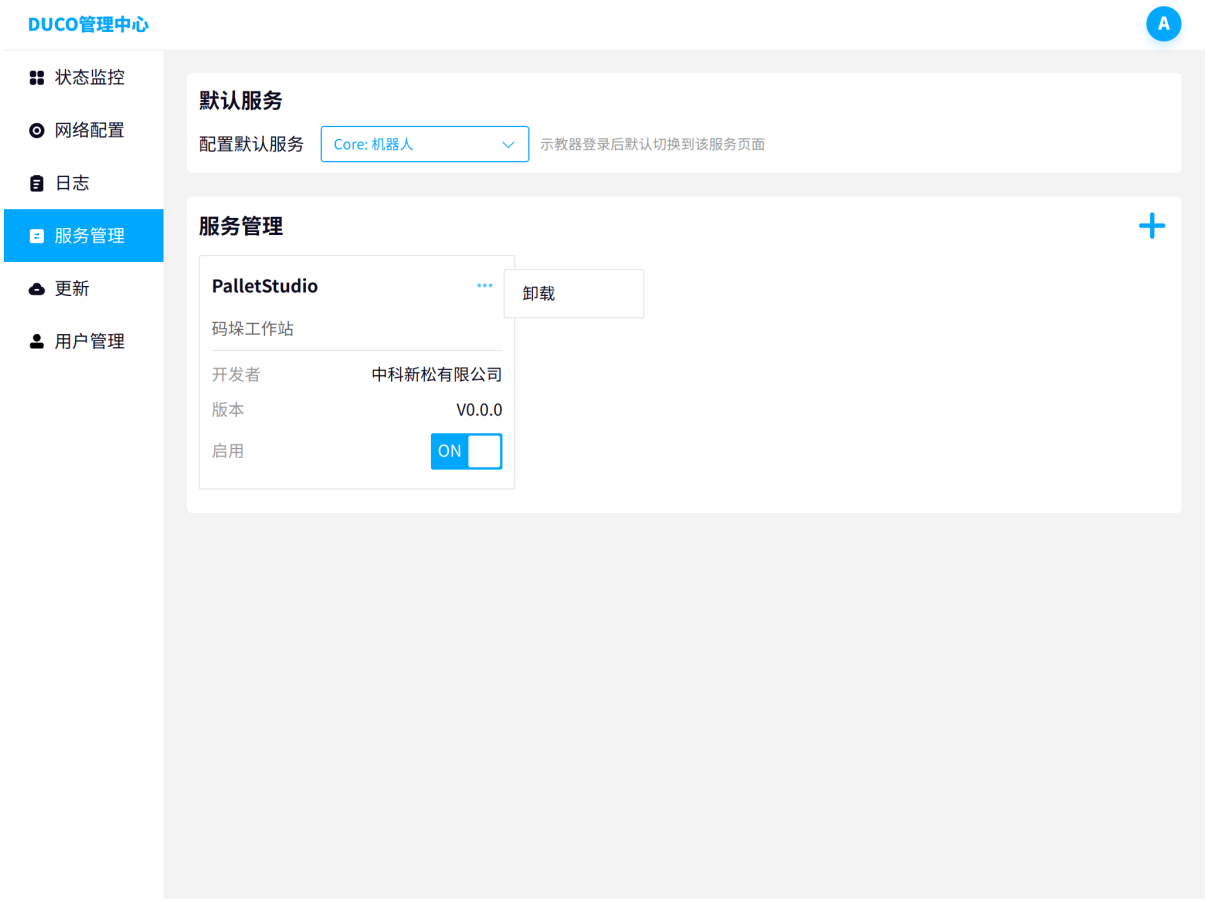
通过配置服务的启用按钮可以配置服务开机是否加载，通过配置默认服务可以选择示教器开启默认的登录界面



安装服务后可以在状态监控界面进行查看

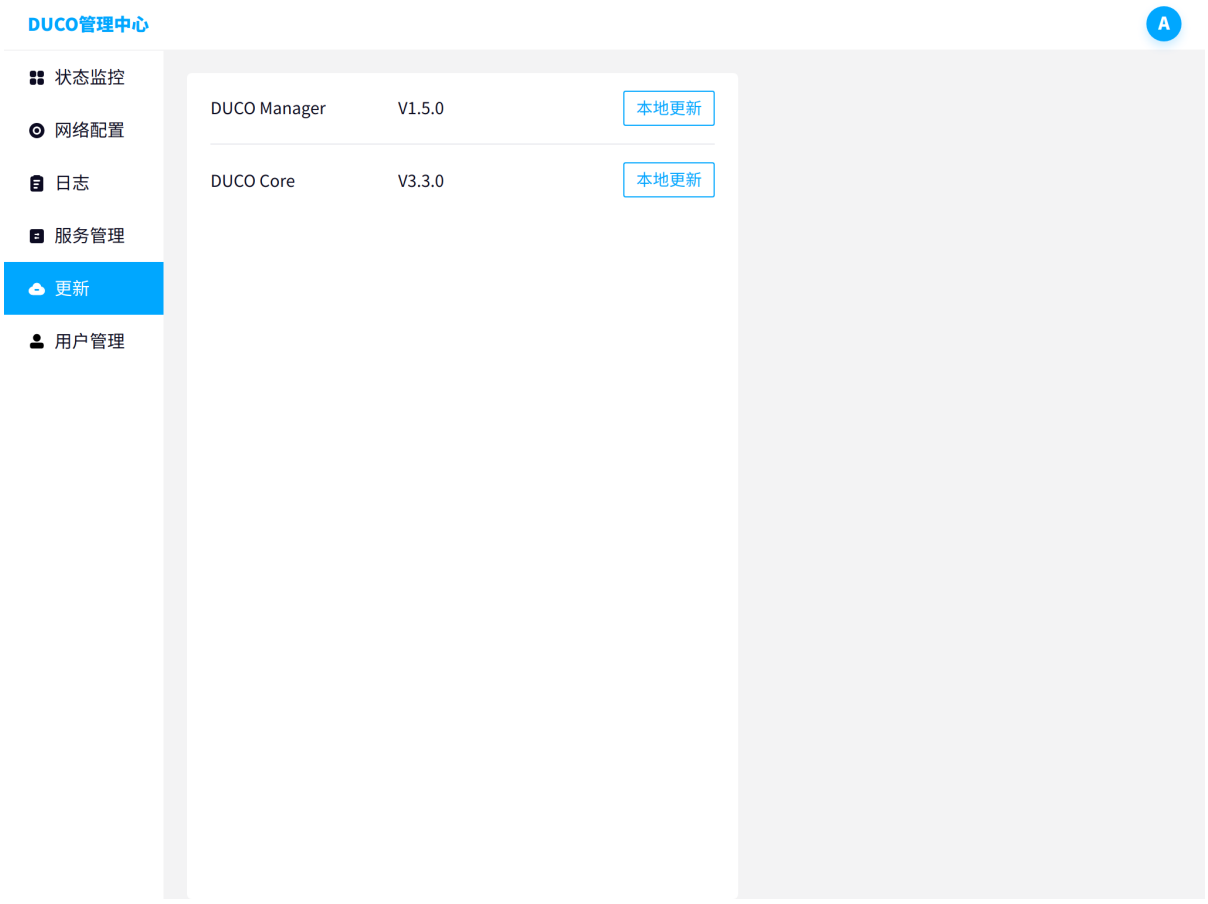


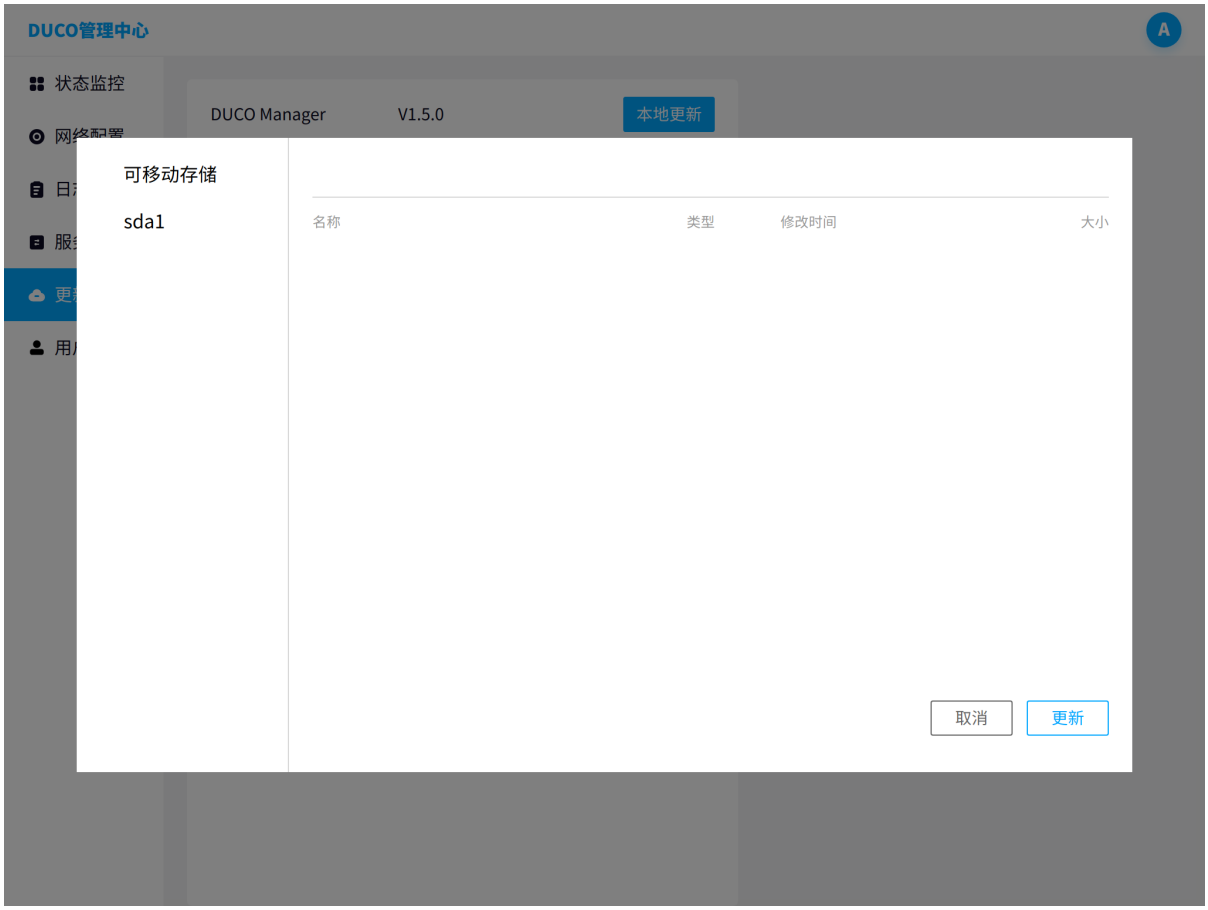
通过点击服务的卸载按钮卸载服务



3.6 更新

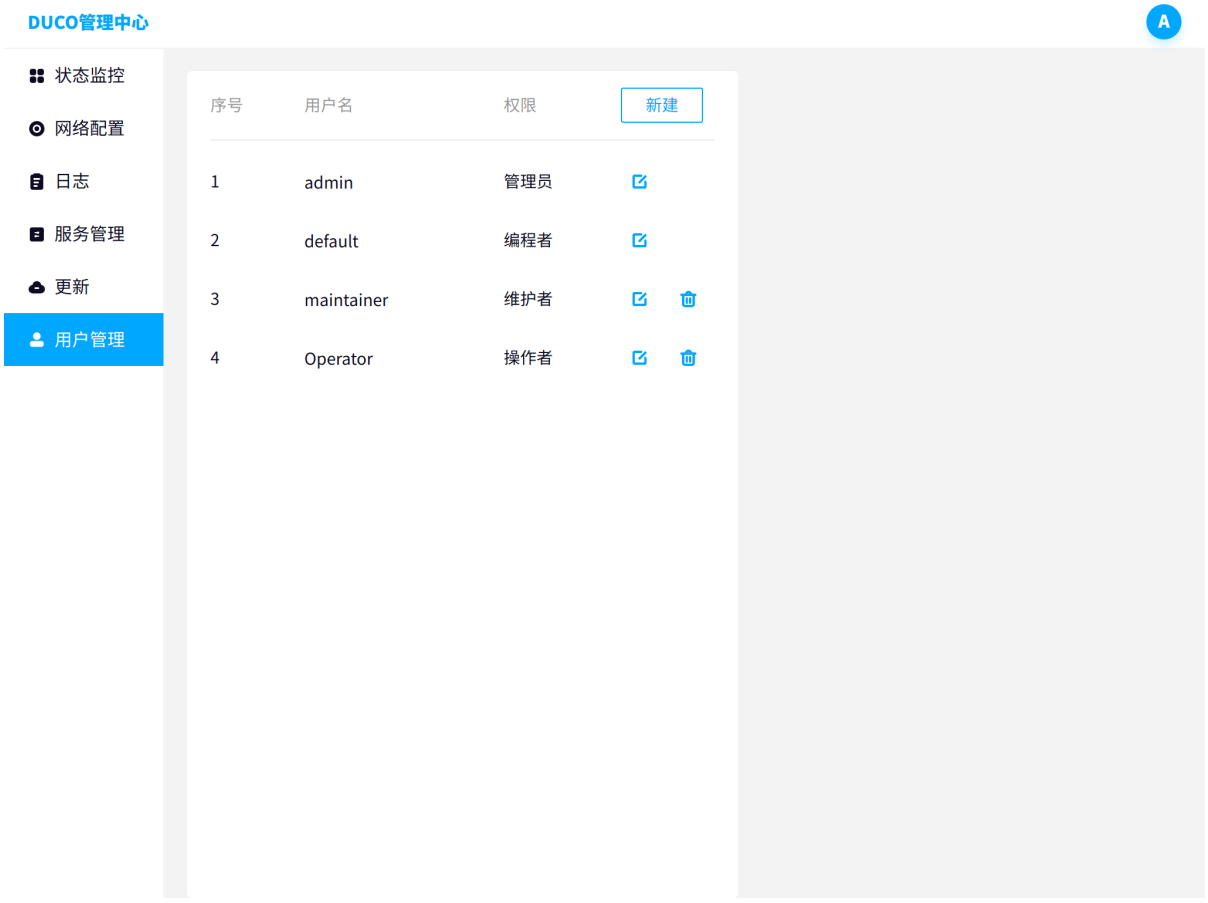
选择本地更新，可对管理中心和管理中心的服务进行更新。


DUCO管理中心
A
状态监控
网络配置
日志
服务管理
更新
用户管理
DUCO Manager V1.5.0 本地更新
DUCO Core V3.3.0 本地更新


DUCO管理中心
A
状态监控
DUCO Manager V1.5.0 本地更新
可移动存储
sda1
名称
类型
修改时间
大小
取消
更新

3.7 用户管理

可添加、删除用户及权限，重新编辑用户名，重新分配用户密码等。



CHAPTER 4

脚本手册

4.1 简介

此脚本函数手册适用于新松协作机器人编程使用。

备注： 脚本函数名不能使用如下划线、标点符号等特殊字符作为起始字符。

4.2 数据类型和变量

4.2.1 内置数据类型

空类型

nil，表示没有任何有效值

算术类型 number

系统内置算数类型为双精度实浮点数 (number)

字符串类型 string

例如“siasun robot”

布尔类型 boolean

例如 true、false

列表类型 list

可容纳其他数据类型的表 (数组结构)

例如 {1,2,3,4,5}、{“a”,”b”,”c”}

注：为了方便后续函数说明，定义以下类型

joints

特指长度为机器人关节自由度数量的描述各关节独立的相关数据信息（如关节位置）的双精度实浮点数列表，单位 rad，例如 {1,2,3,4,5,6}

pose

特指长度为 6 的描述笛卡尔空间位姿数据的双精度实浮点数列表，单位 m、rad，例如 {1,2,3,4,5,6}

num_list

特指双精度实浮点数列表，例如 {1.1,2.2,3,4}

pose_list

特指描述一组位姿数据的 pose 列表，例如 {{1,2,3,4,5,6},{2,3,4,5,6,7}}

joints_list

特指描述一组关节相关数据的 joints 列表，例如 {{1,2,3,4,5,6},{2,3,4,5,6,7}}

pose_speed

特指描述末端速度的双精度浮点数，单位 mm/s、m/s，例如 2.5

pose_acc

特指描述末端加速度的双精度浮点数，单位 mm/s^2、m/s^2，例如 2.5

joint_speed

特指描述关节最大运行速度的双精度浮点数，单位 deg/s、rad/s，例如 2.5

joint_acc

特指描述关节最大加速度的双精度浮点数，单位 deg/s^2、rad/s^2，例如 2.5

timer

特指描述时间间隔的双精度浮点数列表，单位 s，例如 1

4.2.2 系统常量

一个形如数字 42 的值被称作字面值常量。每个字面值常量都对应一种数据类型，字面值常量的形式和值决定了它的数据类型和在脚本中的使用方式。系统支持的常量如下。

基本类型常量

算数类型常量、true、false。

算数类型常量通常指的双精度实浮点数常量，该类型常量可以参与关系运算、算数运算、逻辑运算，并可以为基本类型变量进行赋值和初始化，亦可作为函数的实参进行参数传递。

true 在系统中代指真。

false 在系统中代指假。

字符串常量

字符串常量是字符的集合，字符串常量可用于字符串变量的定义，函数参数传递。

4.2.3 变量

变量在系统中通常用一个唯一的标识符来表示（id)。并且变量需要在定义之后才可以使用。

基本类型变量

a =1.23 --变量赋值上面定义了 a 的类型为双精度实浮点数，初始化 a 的值为 1.23

b = true -- b 布尔变量定义，初始化为真

c = false --c 布尔变量定义，初始化为假

字符串类型变量

s1=“siasun” --s1 变量定义字符串类型初始化为“siasun”

s2=“SiasunCobot” --s2 变量定义字符串类型初始化为“SiasunCobot”

s2=s1 --s2 变量赋值，值修改为”siasun”

列表类型变量

a1={} --a1 变量定义，空列表，长度为 0

a2={1.1,2.2,3.3,4.4} --a2 变量定义，双精度实浮点数列表，长度为 4

a3={“a”,”b”,”c”} --a3 变量定义，字符串列表，长度为 3

a4=a3 #a4 变量定义，a4 和 a3 指向同一个列表，修改 a4 的值，a3 同时被修改

可以通过索引操作具体的元素，索引下标从 1 开始，比如

a2[1]=12

a3[2]=”siasun”

4.3 表达式

表达式在系统中指的是有返回值的运算集合，是一个递归的定义。在此用 exp 表示表达式，用 op 表示运算符。

4.3.1 算术运算表达式

系统支持加（+）减（-）乘（*）除（/）四类算数运算，算数运算表达式返回算数类型常量。

4.3.2 关系运算表达式

系统支持 > < >= <= == ~= 类型的关系运算，关系运算表达式返回常量 True 或 False。

4.3.3 逻辑运算表达式

系统支持 and 、or 、not 类型的逻辑运算。

exp1 and exp2，若 exp1 为真而且 exp2 为真，那么表达式返回 true，否则返回 false。

exp1 or exp2，若 exp1 为真或者 exp2 为真，那么表达式返回 true，否则返回 false。

not exp1，若 exp1 为真，那么表达式返回 false，否则返回 true。

4.3.4 赋值表达式

变量名（id）= exp

赋值表达式的左值是一个变量的表示符，只有在变量已经定义过的情况下该表达式才是一个赋值表达式，若变量没有定义，那么就是一个变量定义语句。

4.3.5 函数调用表达式

函数名（实参列表）

4.4 语句

4.4.1 循环语句 while

```
while (exp) do
exp
...
end
```

4.4.2 函数定义

```
function fun( 参数列表)
语句列表
...
return exp
end
```

例:

```
function fun1(a,b,c)
...
return a+b+c
end
```

参数类型：以实际传入参数为准

返回值类型：以实际返回参数为准

函数可以多参数返回，以”,”分割列如：

```
function fun1(a,b,c)
...
return a,b,c
end
```

4.4.3 return 语句

```
return exp
```

return 语句只能用在函数内部，用来返回函数的运算结果。

return 语句返回的数据类型决定了函数的返回值。若函数中没有包含 return 语句，那么函数不返回任何值。

4.4.4 条件控制语句

形式一：

```
if (exp) then
...
elseif (exp) then
...
else
...
end
```

形式二：

```
if (exp)
then
...
end
```

形式三：

```
if (exp) then
...
else
...
end
```

同大多数语言一样，系统支持 if 条件控制语句，当 if / elseif 中的判定条件表达式为真时，就执行语句块中的内容。

注意：控制结构的条件表达式结果可以是任何值，false 和 nil 为假，true 和非 nil 为真。特别注意，0 为 true：

```
if (0) then
print(“0 is true”)
end
>> 0 is true
```

4.4.5 goto 语句

```
goto Label
::Label:: exp
```

例：

```
a=1
::label:: print(“----goto-----”)
if a<3 then
goto label
end
>> ----goto-----
```

4.4.6 break 语句

使用 break 语句终止循环。

例：

```
a=10
while(a<20) do
print("a=",a)
a=a+1
if(a>15) then
break
end
```

```
end
```

4.5 系统函数说明

4.5.1 移动控制

Op={number:time_or_dist_1,number:trig_io_1,boolean:trig_value_1, number:trig_time_1,number:trig_dist_1, string:trig_event_1, number:time_or_dist_2,number:trig_io_2,boolean:trig_value_2, number:trig_time_2,number:trig_dist_2, string:trig_event_2}

特殊类型说明：

该参数类型用于控制机械臂在移动过程中控制机器人系统 IO 与寄存器输出。

参数说明：

time_or_dist_1：轨迹起始点触发类型，0：不启用，1：时间触发，2：距离触发。

trig_io_1：：轨迹起始点目标触发 IO 或寄存器索引号，定义如下：1-16 为控制柜通用 IO 输出 1-16，101-164 为 bool 寄存器输出 1-64，201-232 为 word 寄存器输出 1-32，301-332 为 float 寄存器输出 1-32，401-402 为机器人末端 IO 输出 1-2。

trig_value_1：：轨迹起始点目标触发 IO 或寄存器目标值。当目标触发类型为 IO 时，0 为低电平，非零为搞定平。当目标触发类型为寄存器时，trig_value_1 的值会根据目标寄存器数据类型进行强制转换后输出。

trig_time_1：轨迹起始点触发时间参数。当且仅当 time_or_dist_1 为时间触发时有效，代表轨迹运行多少时间长度触发 IO，单位 ms。

trig_dist_1：轨迹起始点触发距离参数。当 time_or_dist_1 为距离触发时，代表轨迹运行多少距离长度触发 IO，单位 m。

trig_event_1 ：轨迹起始点触发目标用户自定义事件名称。目标触发触发事件需要预先在机器人系统事件功能中定义且名称匹配。事件触发优先级高于 IO 或寄存器触发优先级，即若存在目标触发自定义事件，则不会触发目标 IO 或寄存器。若目标触发 IO 或寄存器，该参数写空。

time_or_dist_2：轨迹结束点触发类型，0：不启用，1：时间触发，2：距离触发。

trig_io_2：轨迹结束点目标触发 IO 或寄存器索引号，定义如下：1-16 为控制柜通用 IO 输出 1-16，101-164 为 bool 寄存器输出 1-64，201-232 为 word 寄存器输出 1-32，301-332 为 float 寄存器输出 1-32，401-402 为机器人末端 IO 输出 1-2。

trig_value_2：轨迹结束点目标触发 IO 或寄存器目标值。当目标触发类型为 IO 时，0 为低电平，非零为搞定平。当目标触发类型为寄存器时，trig_value_2 的值会根据目标寄存器数据类型进行强制转换后输出。。trig_time_2：轨迹结束点触发时间参数。对于 time_or_dist_2 为时间触发时有效。当 trig_time_2 > =0 时，代表轨迹运行剩余多少时间长度触发目标 IO 或寄存器，单位 ms；当 trig_time_2 < 0 时，代表代表轨迹运行结束后多少时间长度后触发目标 IO 与寄存器。

trig_dist_2: 轨迹结束点触发距离参数。对于 time_or_dist_2 为距离触发时有效，当 trig_dist_2 > =0 时，代表轨迹运行剩余多少距离长度触发 IO，单位 m；当 trig_dist _2 < 0 时，代表代表轨迹运行结束后多少距离长度后触发目标 IO 与寄存器。

trig_event_2: 轨迹结束点触发目标用户自定义事件名称。目标触发触发事件需要预先在机器人系统事件功能中定义且名称匹配。事件触发优先级高于 IO 或寄存器触发优先级，即若存在目标触发自定义事件，则不会触发目标 IO 或寄存器。若目标触发 IO 或寄存器，该参数写空。

！ 注意

Op 参数在所有移动脚本中均为可缺省参数，即当没有 Op 指令时，不会再运动过程中触发 IO、寄存器或事件

---

movej2(joints : axis, number : v, number : a, number : rad = 0, boolean : block = true,Op, boolean : def_acc=false)

函数说明:

该指令控制机械臂从当前状态，按照各关节相位同步运动的方式移动到目标关节角状态。

参数说明:

axis : 目标机器人关节角度位置，范围 [-2*PI, 2*PI]，单位 (rad)。

v : 最大关节角速度指令，范围 [0.01*PI/180, 1.25*PI]，单位 (rad/s)。

a : 最大关节角加速度指令，范围 [0.01*PI/180, ∞]，单位 (rad/s2)。

rad : 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

block : 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op : 详见上方 Op 特殊类型说明，可缺省参数。

def_acc : 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例:

```
movej2({0,0,1.57,0,-1.57,0},0.523,5.23,0)
```

---

movej_pose2(pose : p, number : v, number : a, number : rad, joints : qnear, string : tool, string : wobj, boolean : block=true,Op, boolean : def_acc=false)

函数说明:

该指令控制机械臂从当前状态，按照各关节相位同步运动的方式移动到末端目标位置。

参数说明：

p ： 目标机器人工具在参考机器人工件坐标系中的位姿，位置单位 m，姿态以 Rx、Ry、Rz 表示，范围 [-2*PI, 2*PI]，单位 (rad)。

v ： 最大关节角速度，范围 [0.01*PI/180, 1.25*PI]，单位 (rad/s)。

a ： 最大关节加速度，范围 [0.01*PI/180, ∞]，单位 (rad/s2)。

rad ： 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

qnear ： 目标点附近位置对应的关节角度，用于确定逆运动学选解。

tool ： 设置使用的工具的名称，默认为当前使用的工具。

wobj ： 设置使用的工件坐标系的名称，默认为当前使用的工件坐标系。

block ： 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op ： 详见上方 Op 特殊类型说明，可缺省参数。

def_acc ： 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值：

无

示例：

```
movej_pose2({0.49,0.14,0.44,-3.14,0,-1.57},
0.5,5,0,{0,0,1.57,0,-1.57,0} ,” default” ,” default” )
```

---

movel(pose : p, number : v, number : a, number : rad, joints : qnear, string : tool, string : wobj, boolean : block=true,Op, boolean : def_acc=false)

函数说明：

该指令控制机械臂末端从当前状态按照直线路径移动到目标位姿。

参数说明：

p ： 目标机器人工具在参考机器人工件坐标系中的位姿，位置单位 m，姿态以 Rx、Ry、Rz 表示，范围 [-2*PI, 2*PI]，单位 (rad)。

v ： 最大末端线速度，范围 [0.01, 5]，单位 (m/s)。

a ： 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad ： 轨迹融合半径，单位 (m)，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

qnear ： 目标点附近位置对应的关节角度，用于校验机器人运动过程中逆运动学选解空间。

tool ： 设置使用的工具的名称，默认为当前使用的工具。

wobj ： 设置使用的工件坐标系的名称，默认为当前使用的工件坐标系。

block ： 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op ： 详见上方 Op 特殊类型说明，可缺省参数。

def_acc ： 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例:

```
movel({0.49,0.14,0.35,-3.14,0,-1.57},0.5,5,0,{0,0.02,1.77,-0.22,-1.57,
0},” default” ,” default” )
```

---

movec(pose : p1, pose : p2, number : v, number : a, number : rad, number: mode, joints : qnear, string : tool, string : wobj, boolean : block=true,Op, boolean : def_acc=false)

函数说明:

该指令控制机械臂做圆弧运动，起始点为当前位姿点，途径 p1 点，终点为 p2 点。

参数说明:

p1 ： 圆弧运动过程中任意机器人工具在参考机器人工件坐标系中的位姿中间点，位置单位 m，姿态以 Rx、Ry、Rz 表示范围 [-2*PI, 2*PI]，单位 (rad) 。

p2 ： 目标机器人工具在参考机器人工件坐标系中的位姿，位置单位 m，姿态以 Rx、Ry、Rz 表示范围 [-2*PI, 2*PI]，单位 (rad) 。

v ： 最大末端线速度，范围 [0.01, 5]，单位 (m/s)。

a ： 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad ： 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

mode：姿态控制模式。

0：姿态与终点保持一致，即机器人会以 p2 点的姿态为目标姿态，平滑运动到目标姿态。

1：姿态与起点保持一致，即机器人会以开始执行 movec 函数时机器人末端工具坐标系在工件坐标系中的姿态为准，始终保持该姿态值。

2：姿态受圆心约束，即机器人会以开始执行 movec 函数时机器人末端工具坐标系与目标圆弧路径起点处切线方向间关系为参考，在圆弧运动过程中始终保持末端工具与圆弧实时运动所处位置切线方向参考关系。

qnear ： 目标点附近位置对应的关节角度，用于校验机器人运动过程中逆运动学选解空间。

tool ： 设置使用的工具的名称，默认为当前使用的工具。

wobj ： 设置使用的工件坐标系的名称，默认为当前使用的工件坐标系。

block ： 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op ： 详见上方 Op 特殊类型说明，可缺省参数。

def_acc ： 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例:

```
movec({0.397,0.14,0.35,-3.14,0,-1.57},{0.397,0.04,0.35,-3.14,0,-1.57},
0.5,5,0,{-0.47,-0.26,2.02,-0.19,-1.57,-0.47},” default” ,”
default” )
```

---

move_circle (pose : p1, pose : p2, number : v, number : a, number : rad, number: mode, joints : qnear, string : tool, string : wobj, boolean : block=true,Op, boolean : def_acc=false)

函数说明:

该指令控制机械臂做圆周运动，起始点为当前位姿点，途径 p1 点和 p2 点

参数说明:

p1 圆周运动过程中任意机器人工具在参考机器人工件坐标系中的位姿中间点 1，位置单位 m，姿态以 Rx、Ry、Rz 表示范围 [-2*PI, 2*PI]，单位 (rad) 。

p2 ： 圆周运动过程中任意机器人工具在参考机器人工件坐标系中的位姿中间点 2，最终以机器人初始运动位置-p1-p2 的顺序决定最终整圆轨迹，位置单位 m，姿态以 Rx、Ry、Rz 表示范围 [-2*PI, 2*PI]，单位 (rad) 。

v ： 最大末端线速度，范围 [0.01, 5]，单位 (m/s)。

a ： 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad ： 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

mode: 姿态控制模式。

1: 姿态与起点保持一致，即机器人会以开始执行 movec 函数时机器人末端工具坐标系在工件坐标系中的姿态为准，始终保持该姿态值。

2: 姿态受圆心约束，即机器人会以开始执行 movec 函数时机器人末端工具坐标系与目标圆弧路径起点处切线方向间关系为参考，在圆弧运动过程中始终保持末端工具与圆弧实时运动所处位置切线方向参考关系。

qnear ： 目标点附近位置对应的关节角度，用于校验机器人运动过程中逆运动学选解空间。

tool ： 设置使用的工具的名称，默认为当前使用的工具。

wobj ： 设置使用的工件坐标系的名称，默认为当前使用的工件坐标系。

block ： 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op: 详见上方 Op 特殊类型说明，可缺省参数。

def_acc ： 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例:

```
move_circle({0.5,0.0114,0.35,-3.14,0,-1.57},{0.358,0.144,0.35,-3.14,0,
-1.57},        0.5,5,0,{-0.01,-0.3,2.05,-0.179,-1.57,-0.012},”
default”,” default”)
```

---

tcp_move(pose:pose_offset, number : v, number : a, number : rad, string : tool, boolean : block=true,Op, boolean : def_acc=false)

函数说明:

该指令控制机械臂沿工具坐标系直线移动一个增量。

参数说明:

pose_offset:pose 数据类型，或者长度为 6 的 number 型数组，表示工具坐标系下的位姿偏移量。偏移量将会转换为齐次变换矩阵右乘于当前机器人末端位姿之上。

v: 最大末端线速度，范围 [0.01, 5]，单位 m/s，当 x、y、z 均为 0 时，线速度按比例换算成角速度。

a ： 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad ： 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

tool ： 设置使用的工具的名称，默认为当前使用的工具。

block ： 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op: 详见上方 Op 特殊类型说明，可缺省参数。

def_acc ： 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例：

```
tcp_move({0,0,0.1,0,0,0},0.5,5,0 ,” default” )
```

---

tcp_move_2p (pose:p1, pose:p2 , number : v, number : a, number : rad, string : tool, string : wobj,boolean : block=true,Op, boolean : def_acc=false)

函数说明：

该指令控制机器人沿工具坐标系直线移动一个增量，增量为 p1 与 p2 点之间的差，运动的目标点为：当前点 $*p1^{-1}*p2$。

参数说明：

p1 ：表示工具坐标系下的位姿偏移量计算点 1。

p2 ：表示工具坐标系下的位姿偏移量计算点 2。

v：最大末端线速度，范围 [0.01, 5]，单位 (m/s)，当 x、y、z 均为 0 时，线速度按比例换算成角速度。

a ：最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad ：轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

tool ：设置使用的工具的名称，默认为当前使用的工具。

wobj ：设置使用的工件坐标系的名称，默认为当前使用的工件坐标系。

block ：指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op：详见上方 Op 特殊类型说明，可缺省参数。

def_acc ：是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值：

无

示例：

```
tcp_move_2p({0.397,0.04,0.25,-3.14,0,-1.57},{0.397,-0.045,0.43,-3.14,0,
-1.57},0.5,5,0,” default” ,” default” )
```

---

wobj_move(pose:pose_offset, number : v, number : a, number : rad, string : wobj, boolean : block=true,Op, boolean : def_acc=false)

函数说明：

该指令控制机械臂沿工件坐标系直线移动一个增量。

参数说明：

pose_offset:pose 数据类型，或者长度为 6 的 number 型数组，表示工件坐标系下的位姿偏移量。

v: 最大末端线速度，范围 [0.01, 5]，单位 m/s，当 x、y、z 均为 0 时，线速度按比例换算成角速度。

a : 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad : 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

wobj : 设置使用的工件的名称，默认为当前使用的工件。

block : 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op: 详见上方 Op 特殊类型说明，可缺省参数。

def_acc : 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例:

```
wobj_move({0,0,0.1,0,0,0},0.5,5,0 ,”default”)
```

---

wobj_move_2p (pose:p1, pose:p2 , number : v, number : a, number : rad, string : tool, string : wobj,boolean : block=true,Op, boolean : def_acc=false)

函数说明:

该指令控制机器人沿工件坐标系直线移动一个增量，增量为 p1 与 p2 点之间的位姿偏移在工件坐标系下的描述 $offset_{wobj}$，运动的目标点为: 当前点 * $offset_{wobj}$。

参数说明:

p1 : 表示工件坐标系下的位姿偏移量计算点 1。

p2 : 表示工件坐标系下的位姿偏移量计算点 2。

v: 最大末端线速度，范围 [0.01, 5]，单位 (m/s)，当 x、y、z 均为 0 时，线速度按比例换算成角速度。

a : 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

rad : 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。产生融合运动时，机器人程序会根据融合预读取设定参数在运动指令执行过程中尝试提前获取后续运动函数。

tool : 参考点使用的工具的名称，默认为当前使用的工具。

wobj : 参考点使用的工件坐标系的名称，默认为当前使用的工件坐标系。

block : 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op：详见上方 Op 特殊类型说明，可缺省参数。

def_acc ：是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值：

无

示例：

```
wobj_move_2p({0.397,0.04,0.25,-3.14,0,-1.57},{0.397,-0.045,0.43,-3.14,0,
-1.57},0.5,5,0,”default”,”default”)
```

---

speedj (joints : axis_speed, number: a, number: t =-1)

函数说明：

该指令控制机械臂每个关节按照给定的速度一直运动，函数执行后会直接运行后续指令。运行 speedj 函数后，机械臂会持续运动并忽略后续运动指令，直到接收到 speed_stop() 函数后停止。

参数说明：

axis_speed：每个关节的速度，范围 [0.01*PI/180, 1.25*PI]，单位 (rad/s)。

a：主导轴的关节加速度，范围 [0.01*PI/180, ∞]，单位 (rad/ s2)。

t：运行时间，到达时间会停止运动，单位 ms。默认-1 表示一直运行。

返回值：

无

示例：

```
speedj({0.2,0.2,0,0,0,0},5,3000)
```

---

speedl (pose : position_speed, number: a, number: t=-1)

函数说明：

该指令控制机械臂末端按照给定的速度一直运动，函数执行后会直接运行后续指令。运行 speedl 函数后，机械臂会持续运动并忽略后续运动指令，直到接收到 speed_stop() 函数后停止。

参数说明：

position_speed：末端速度向量，线速度范围 [0.01, 5]，单位 (m/s)，角速度范围 (0, 1. 25*PI]，单位 (rad/s)。

a：末端的线性加速度，范围 [0.01, ∞]，单位 (rad/ s2)。

t：运行时间，到达时间会停止运动，单位 ms。默认-1 表示一直运行。

返回值：

无

示例：

```
speedl({0.2,0.2,0,0,0,0},5,3000)
```

---

speed_stop ( )

函数说明:

停止 speedj 及 speedl 函数的运动。

参数说明:

无

返回值:

无

示例:

```
speed_stop()
```

---

spline(pose_list : p_list, number : v, number : a, string : tool, string : wobj, boolean : block=true,Op,number : rad, boolean : def_acc=false)

函数说明:

样条运动函数，该指令控制机器人按照空间样条进行运动。

参数说明:

p_list: 在设置工件坐标系下的末端位姿列表，最多不超过 50 个点，格式如下:

{p1,p2,p3,…,pi,…}

其中 pi 为空间位姿，如 {0.4,0.5,0.5,1.2,0.717,1.414}。

v : 最大末端线速度，范围 [0.01, 5]，单位 (m/s)。

a : 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

tool : 设置使用的工具的名称，默认为当前使用的工具。

wobj : 设置使用的工件坐标系的名称，默认为当前使用的工件坐标系

block : 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op: 详见上方 Op 特殊类型说明，可缺省参数。

rad : 轨迹融合半径，单位 m，默认值为 0，表示无融合。当数值大于 0 时表示与下一条运动融合。

def_acc : 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值:

无

示例:

```
spline({{0.397,-0.11,0.5,-3.14,0,-1.57},{0.397,-0.11,0.43,-3.14,0,-1.
57},{0.316,-0.11,0.43,-3.14,0,-1.57}},     0.5,5,” default” ,”
default” )
```

警告： 连续两条 spline 一起使用时，需要注意前一条的 spline 结束点与后一条的 spline 起始点，不能是同一个点。

move_spiral(pose_list : p1, pose_list : p1, number : rev, number : len, number : rad, number : mode, number : v, number : a, joints : qnear, string : tool, string : wobj, boolean : block=true,Op, boolean : def_acc=false)

函数说明：

该指令通过参数或者结束点两种设置方式，在笛卡尔空间做螺旋轨迹运动。

参数说明：

p1 ： 螺旋线中心点位姿。

p2 ： 螺旋线的目标点位姿，参数设置模式时不参考此参数。

rev ： 总旋转圈数，rev < 0，表示顺时针旋转；rev > 0，表示逆时针旋转。

len ： 轴向移动距离，正负号遵循右手定则，结束点设置模式时不参考

此参数，单位 (m)。

red ： 目标点半径，结束点设置模式时不参考此参数，单位 (m)。

mode ： 螺旋线示教模式，0：参数设置，1：结束点设置。

v ： 最大末端线速度，范围 [0.01, 5]，单位 (m/s)。

a ： 最大末端线加速度，范围 [0.01, ∞]，单位 (m/s2)。

qnear ： 目标点位置对应的关节角度，用于确定逆运动学选解，单位 (rad)

tool ： 设置使用的工具的名称，默认为当前使用的工具。

wobj ： 设置使用的工件坐标系的名称，默认为当前使用的工件坐标系

block ： 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回，默认为阻塞。

Op：详见上方 Op 特殊类型说明，可缺省参数。

def_acc ： 是否使用默认加速度，默认为 false，可缺省参数。当开启默认加速度时，执行运动时最大加速度参数不在生效，系统将会根据实际工况计算机器人可以产生的最大加速度指令进行运动，从而提升节拍。

返回值：

无

示例：

```
move_spiral({{0.41,0.008,0.43,-3.14,0,-1.57},{},
2,0.2,0,0.1,nil,” default” ,” default” )
```

hand_teach ( )

函数说明:

当运行脚本程序时，该函数调用并且触发末端牵引按钮时，可以启用手动牵引功能。

参数说明:

无

返回值:

无

示例:

```
hand_teach()
```

---

replay(string : name, number : v, number : mode)

函数说明:

该函数用于对记录的轨迹基于关节空间（或基于笛卡尔空间）复现。

参数说明:

name : 轨迹名称

v : 轨迹速度，(系统设定速度的百分比%)，取值范围 (0,100]。

mode : 复现方式，0：基于关节空间，1：基于笛卡尔空间。

返回值:

无

示例:

```
replay(“tmp”, 100,0)
```

---

is_moving()

函数说明:

该函数判断机器人是否处于运动状态。

参数说明:

无。

返回值:

True : 机器人正在运动;

False : 机器人没有运动。

示例:

```
is_moving()
```

---

set_blend_ahead (number :per, number:pre_num)

函数说明:

该函数用于设置融合预读取百分比以及预读取运动指令数量。

参数说明:

per : 融合预读取的百分比，单位: %，通常设为 0 或者 50。

pre_num: 当前仅当融合预读取百分比参数设置为 0 时生效，可缺省，范围[1,3]，默认为 1，即仅额外预读取一行运动指令。在机器人目标工作在高速连续短轨迹融合运动时，若不存在过程逻辑问题，应尽可能设置较大的预读取轨迹数量以保证融合稳定性。

返回值:

无

示例:

```
set_blend_ahead (0,2)
```

4.5.2 网络通信

socket_open(string : ip, number : port, string : name ="socket1" , number : timeout=3000)

函数说明:

创建一个以太网客户端并连接目标服务端，如果超时时间内没有创建成功，那么链接创建失败并返回 false。

参数说明:

ip: 服务器端的 IP 地址，需要用“”括起来。

port: 服务器端的端口号。

name: 连接名称，可省略，默认为“socket1”。

tiomeout : 设置超时时间，单位 (ms)，默认 3000ms。

返回值:

true: 操作成功。

false: 操作失败。

示例:

```
socket_open("192.168.67.60",2003,"socket2",3000)
```

---

socket_write (msg_send,string : name="socket1", number : timeout=3000)

函数说明:

发送数据给已连接的服务器，函数根据不同数据类型自动判断。

参数说明:

msg_send ：需要发送的数据，接受以下类型：

<1> 字符串，需要以“作为字符串头并且已”作为字符串结尾。

<2> number 列表，列表中每个数据为 32 位 float。

name：连接名称，默认为“socket1”。

tiomeout ：设置超时时间，单位 (ms)，默认 3000ms。

返回值：

true：操作成功。

false：操作失败。

示例：

```
socket_write ("send message", "socket1",1000)
socket_write ({1,2,3.14},"socket1",1000)
```

---

socket_read_string(number:length,string:prefix="",string:subfix="", string: name="socket1", number:timeout = 3000)

函数说明：

从已连接的服务器接收一定长度的字符串。

参数说明：

length：需要接收的字符串长度，当设置后缀时，长度参数无效。

prefix：接收的字符串前缀，默认为空，当设置前缀后，从设置前缀开始读取，直至达到设置长度或者设置后缀。

subfix：接收的字符串的后缀，默认为空，当设置后缀后，从字符串开始位置读取，直至读取到设置后缀，当达到最大限制长度 255 时，返回 255 字节长度的字符串。

name：socket 连接的名字，字符串格式。

timeout：超时时间，单位 ms。不填写时默认为 3000ms。

返回值：

收到字符串信息，如果读取失败，返回空 string。

示例：

```
socket_read_string (10, " ", " ","socket1",3000)
```

---

socket_read_str_num (string:name="socket1",number: timeout = 3000)

函数说明：

从已连接的服务器接收一组字符串，解析成 num 数值，所有数据应当在括号内，数据之间通过“,”隔开，数据格式为：(num1,num2,num3)

参数说明：

name：socket 连接的名字，字符串格式。

timeout：超时时间，单位 (ms)。不填写时默认为 3000ms。

返回值：

收到的 number 型列表，如果读取失败返回空列表。

示例：

```
list=socket_read_str_num("socket",1000)
```

---

socket_read_num(number:count, name="socket1",timeout = 3000)

函数说明：

从已连接的服务器接收一组浮点数。

参数说明：

count：需要接收的数据个数，每个数据为单精度浮点数。

name：socket 连接的名字，字符串格式。

timeout：超时时间，单位 (ms)。不填写时默认为 3000ms。

返回值：

收到的 number 列表，如果读取失败返回空列表。

示例：

```
list= socket_read_num (10, "socket",1000)
```

---

socket_close(string : name = "socket1")

函数说明：

关闭和服务器端的以太网通信链接。

参数说明：

name：连接名称，默认为“socket1”。

返回值：

无

示例：

```
list= socket_close ("socket")
```

---

set_data_frequence(number : freq)

函数说明：

设置 2001 端口数据的发送频率。

参数说明：

freq：发送频率，范围 [1,100]，单位 hz。

返回值：

无

示例：

```
list= set_data_frequence (10)
```

---

socket_write_byte_list (table:list,table:head,table:tail,name="socket1",timeout=3000)

函数说明：

发送 byte 类型数据给已连接的服务器。

参数说明：

list：发送的 byte 类型数据列表

head：数据头。该数据可缺省。

tail：数据尾。该数据可缺省。

name：socket 连接的名字，字符串格式，默认为 socket1。

timeout：超时时间，单位 (ms)。不填写时默认为 3000ms。

返回值：

true：操作成功。

false：操作失败。

示例：

```
socket_write_byte_list ({255,254,1,2,3,253,252}," socket1" ,
3000)
socket_write_byte_list ({1,2,3},{255,254},{253,252}"socket1",
1000)
```

---

socket_read_byte_list (table:head, int:len(table:tail),name="socket1",timeout=3000)

函数说明：

从已连接的服务器接收一组 byte 类型数据。

参数说明：

head：数据头。该数据可缺省，如果缺省，则数据长度是必要参数；如果使用数据头，则数据长度或数据尾，可任选一种。

len：数据长度。可以用数据头与数据尾代替。

tail：数据尾。该数据可缺省。

name：socket 连接的名字，字符串格式，默认为 socket1。

timeout：超时时间，单位 (ms)。不填写时默认为 3000ms。

返回值：

收到的 byte_list 列表，如果读取失败返回空列表。

示例：

```
list=socket_read_byte_list (10,” socket1” ,3000);
list=socket_read_byte_list ({255,254},10,” socket1” ,3000);
list=socket_read_byte_list ({255,254},{253,252},” socket1” ,
3000);
```

4.5.3 can 及 485 总线

read_raw_data_485 (number:len, number:time=3000)

函数说明:

485 端口在设置的时间内读取长度为 len 的字节数据。

参数说明:

len: 需要读取的长度。

time: 等待时间，默认 3000ms，单位 (ms)。

返回值:

number_list 读取到的数据，在规定的时间内，接收到指定长度的数据会立即返回，未接收到指定长度的数据时，等待结束后，返回收到的数据。

示例:

```
list= read_raw_data_485 (10,3000)
```

---

read_raw_data_485 (number_list:head, number_list:tail, number:time=3000)

函数说明:

匹配头 head 和尾 tail 读取到一帧匹配的数据。

参数说明:

head: 需要匹配的头数据。

tail: 需要匹配的尾数据。

time: 等待时间，默认 3000ms，单位 (ms)。

返回值:

number_list 读取到的数据，在规定的时间内，接收到指定长度的数据会立即返回，未接收到指定长度的数据时，等待结束后，返回收到的数据。

示例:

```
list= read_raw_data_485 ({255,1},{1,255})
```

---

read_raw_data_485 (number_list:head, number:len, number:time=3000)

函数说明:

匹配头 head 后读取到长度为 len 的一帧数据。

参数说明:

head: 需要匹配的头数据。

len: 需要读取的长度。

time: 等待时间，默认 3000ms，单位 (ms)。

返回值:

number_list 读取到的数据，在规定的时间内，接收到指定长度的数据会立即返回，未接收到指定长度的数据时，等待结束后，返回收到的数据。

示例:

```
list= read_raw_data_485 ({255,1},10)
```

---

read_data_485 ()

函数说明:

配方 485 读数据，按照配方将输入的数据转换成配方后的数据（自定义配方情况下使用)。

参数说明:

无

返回值:

number_list 读取到的数据，未读到数据返回空列表。

示例:

```
list= read_data_485 ()
```

---

write_raw_data_485 (number_list:value)

函数说明:

485 写原生数据，将列表 value 中的数据写入 485 端口。

参数说明:

value: 需要写入的数据列表。

返回值:

true: 成功。

false: 失败。

示例:

```
write_raw_data_485 ({1,2,3})
```

---

write_raw_data_485 (number_list:value,number_list:head)

函数说明:

485 写原生数据，将列表 value 中的数据加上 head 写入 485 端口。

参数说明:

value: 需要写入的数据列表。

head: 需要添加的头。

返回值:

true: 成功。

false: 失败

示例:

```
write_raw_data_485 ({1,2,3},{255,255})
```

---

write_raw_data_485 (number_list:value,number_list:head, number_list:tail)

函数说明:

485 写原生数据，将列表 value 中的数据加上头 head 和尾 tail 写入 485 端口。

参数说明:

value: 需要写入的数据列表。

head: 需要添加的头。

tail: 需要添加的尾

返回值:

true: 成功。

false: 失败。

示例:

```
write_raw_data_485 ({1,2,3},{255,255},{255,255})
```

---

write_data_485 (number_list:value)

函数说明:

配方 485 写数据，按照配方的输出配置将转换后的流数据写入 485 端口（自定义配方情况）。

参数说明:

value: 需要写入的数据列表。

返回值:

true: 成功。

false: 失败。

示例:

```
write_data_485 ({255,255,1,1,1,238,238})
```

---

tool_read_raw_data_485 (number:len)

函数说明:

末端 485 端口读取长度为 len 的字节数据。

参数说明:

len: 需要读取的长度。

返回值:

number_list 读取到的数据，未读到数据返回空列表。

示例:

```
list= tool_read_raw_data_485 (10)
```

---

tool_read_raw_data_485 (number_list:head, number_list:tail)

函数说明:

末端 485 匹配头 head 和尾 tail 读取到一帧匹配的数据。

参数说明:

head: 需要匹配的头数据。

tail: 需要匹配的尾数据。

返回值:

number_list 读取到的数据，未读到数据返回空列表。

示例:

```
list= tool_read_raw_data_48 ({255,1},{1,255})
```

---

tool_read_raw_data_485 (number_list:head, number:len)

函数说明:

末端 485 匹配头 head 后读取到长度为 len 的一帧数据。

参数说明:

head: 需要匹配的头数据。

len: 需要读取的长度。

返回值:

number_list 读取到的数据，未读到数据返回空列表。

示例:

```
list= tool_read_raw_data_485 ({255,1},10)
```

---

tool_read_data_485 ()

函数说明:

末端配方 485 读数据，按照配方将输入的数据转换成配方后的数据（自定义配方情况下使用)。

参数说明:

无

返回值:

number_list 读取到的数据，未读到数据返回空列表。

示例:

```
list= tool_read _data_485 ()
```

---

tool_write_raw_data_485 (number_list:value)

函数说明:

末端 485 写原生数据，将表 value 中的数据写入 485 端口。

参数说明:

value: 需要写入的数据列表。

返回值:

true: 成功。

false: 失败。

示例:

```
tool_write_raw_data_485 ({1,2,3})
```

---

tool_write_raw_data_485 (number_list:value,number_list:head)

函数说明:

末端 485 写原生数据，将表 value 中的数据加上 head 写入 485 端口。

参数说明:

value: 需要写入的数据列表。

head: 需要添加的头。

返回值:

true: 成功。

false: 失败

示例:

```
tool_write_raw_data_485 ({1,2,3},{255,255})
```

---

tool_write_raw_data_485 (number_list:value,number_list:head, number_list:tail)

函数说明:

末端 485 写原生数据，将表 value 中的数据加上头 head 和尾 tail 写入 485 端口。

参数说明:

value: 需要写入的数据列表。

head: 需要添加的头。

tail: 需要添加的尾

返回值:

true: 成功。

false: 失败。

示例:

```
tool_write_raw_data_485 ({1,2,3},{255,255},{255,255})
```

---

tool_write_data_485 (number_list:value)

函数说明:

末端配方 485 写数据，按照配方的输出配置将转换后的流数据写入 485 端口(自定义配方情况)。

参数说明:

value: 需要写入的数据列表。

返回值:

true: 成功。

false: 失败。

示例:

```
tool_write_data_485 ({255,255,0,0,0,255,255})
```

---

read_raw_data_can ()

函数说明:

读取一帧 can 的字节数据。

参数说明:

无

返回值:

number_list 读取到的数据，未读到数据返回空列表，读到数据时，列表的第一个数据为发送端的 can 帧 id。

示例:

```
list = read_raw_data_can ()
```

---

read_raw_data_can (number:id)

函数说明:

根据 id 读取一帧 can 的字节数据。

参数说明:

id: 需要读取的帧 id。

返回值:

number_list 读取到的数据，未读到数据返回空列表，读到数据时，列表的第一个数据为发送端的 can id。

示例:

```
list = read_raw_data_can (1)
```

---

read_data_can ()

函数说明:

配方 can 读数据，按照配方将输入的数据转换成配方后的数据（自定义配方情况下）。

参数说明:

无

返回值:

number_list 读取到的数据，未读到数据返回空列表，读到数据时，列表的第一个数据为发送端的 can id。

示例:

```
list = read_data_can ()
```

---

write_raw_data_can (number:id, number_list:value)

函数说明:

can 写帧为 id，数据为 value 的原生数据。

参数说明:

id: 数据帧的 id。

value: 要发送的数据列表。

返回值:

true: 成功。

false: 失败。

示例:

```
write_raw_data_can (1,{1,2,3})
```

write_data_can (number: id, number_list: value)

函数说明:

can 写帧为 id，数据为 value 的配方数据。

参数说明:

id: 数据帧的 id。

value: 要发送的数据列表。

返回值:

true: 成功。

false: 失败。

示例:

```
write_data_can (1,{1,2,3})
```

4.5.4 系统函数及外设

! 注意

若范围参数越界，函数将返回 false 或 0

sleep(number : time)

函数说明:

该函数会让程序暂停一定时间。

参数说明:

time : 需要暂停的时间，单位 (ms)。

返回值:

无

示例:

```
sleep (1000)
```

---

set_digital_output_mode (number : portnum, number: type, number: freq, number: duty_cycle)

函数说明:

该函数可设置控制柜上的通用 IO 输出的信号类型。

参数说明:

portnum : 控制柜上的 IO 输出口序号，范围从 1-16。

type : 0 为电平模式，1 为周期脉冲模式。

freq : 频率，单位: Hz，范围从 1-100。

duty_cycle ： 占空比，单位：%，1-100。

参数错误时函数不改变 IO 输出信号类型。

类型设置完成后，若设置未电平迷失，则需要主动发送输出信号才能生效。若设置成周期脉冲模式后，需要发送一个高电平的启动信号。

返回值：

无

示例：

```
set _digital_out_mode (1,1,100,50)
set_standard_digital_out (1,true)
```

备注： 若要将通用 output 修改为脉冲输出，在对通用 IO 输出类型完成配置后，仍然需要使用 set_standard_digital_out 函数来控制脉冲的有无。当前仅允许对脉冲类型统一设置，无法对不同通用 output 口，设置不同的脉冲类型。脉冲脉宽的最小识别率为 1ms。即当频率为 100hz 时，占空比最小值为 10。若超出范围将关闭脉冲输出。

---

set_standard_digital_out (number : portnum, boolean: val)

函数说明：

该函数可控制控制柜上的通用 IO 输出口的高低电平。

参数说明：

portnum ： 控制柜上的 IO 输出口序号，范围从 1-16。

val ： true 为高电平，false 为低电平。

参数错误时函数不改变 IO 输出。

返回值：

无

示例：

```
set_standard_digital_out (1,true)
```

---

get_standard_digital_out (number : portnum)

函数说明：

该函数可读取控制柜上的通用 IO 输出口的高低电平，返回 true 为高电平，false 为低电平。若对应的 IO 输出口被配置为周期性脉冲模式，则该接口返回值不具备参考意义。

参数说明：

portnum ： 控制柜上的 IO 输出口序号，范围从 1-16。

返回值：

boolean，返回 true 为高电平，false 为低电平。

示例:

```
value = get_standard_digital_out (1)
```

---

get_standard_digital_in (number : portnum)

函数说明:

该函数可读取控制柜上的通用 IO 输入口的高低电平，返回 true 为高电平，false 为低电平。

参数说明:

portnum : 控制柜上的 IO 输入口序号，范围从 1-16。

返回值:

boolean，返回 true 为高电平，false 为低电平。

示例:

```
value = get_standard_digital_in (1)
```

---

set_tool_digital_out (number : portnum, boolean : val)

函数说明:

该函数可控制机械臂末端的 IO 输出口的高低电平。

参数说明:

portnum : 机械臂末端的 IO 输出口序号，范围从 1-2。

val : true 为高电平，false 为低电平。

参数错误时函数不改变 IO 输出。

返回值:

无

示例:

```
set_tool _digital_out (1, true)
```

---

get_tool_digital_in (number : portnum)

函数说明:

该函数可读取机械臂末端的 IO 输入口的高低电平，返回 true 为高电平，false 为低电平。

参数说明:

portnum : 机械臂末端的 IO 输入口序号，范围从 1-2。

返回值:

boolean，返回 true 为高电平，false 为低电平。

示例:

```
value = get_tool _digital_in (1)
```

get_tool_digital_out (number : portnum)

函数说明:

该函数可读取机械臂末端的 IO 输出口的高低电平,返回 true 为高电平,false 为低电平。

参数说明:

portnum : 机械臂末端的 IO 输出口序号,范围从 1-2。

返回值:

boolean,返回 true 为高电平,false 为低电平。

示例:

```
value = get_tool _digital_out (1)
```

---

get_function_digital_in (number : portnum)

函数说明:

该函数可读取控制柜功能输入 IO 高低电平,返回 true 为高电平,false 为低电平。

参数说明:

portnum : 控制柜功能 IO 输入口序号,范围从 1-8。

返回值:

boolean,返回 true 为高电平,false 为低电平。

示例:

```
value = get_function _digital_in (1)
```

---

get_function_digital_out (number : portnum)

函数说明:

该函数可读取控制柜功能输出 IO 高低电平,返回 true 为高电平,false 为低电平。

参数说明:

portnum : 控制柜功能 IO 输出口序号,范围从 1-8。

返回值:

boolean,返回 true 为高电平,false 为低电平。

示例:

```
value = get_function _digital_out (1)
```

---

get_standard_analog_voltage_in (int : num)

函数说明:

该函数可读取控制柜上的模拟电压输入。

参数说明:

num : 控制柜上的模拟电压通道序号，范围从 1-4。

返回值:

对应通道的模拟电压值。

示例:

```
value = get_standard_analog_voltage_in (1)
```

---

set_standard_analog_voltage_out (int : num, double : value, bool : block)

函数说明:

该函数可设置控制柜上的模拟电压输出。

参数说明:

num : 控制柜上的模拟电压通道序号，范围从 1-4;

value : 设置的模拟电压值，范围 [0,10]，单位 V;

block : 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回。

返回值:

当配置为阻塞执行，返回值代表当前任务结束时的状态，当配置为非阻塞执行，返回值代表当前任务的 id 信息，用户可以调用 get_noneblock_taskstate (id) 函数查询当前任务的执行状态。

示例:

```
value = set_standard_analog_voltage_out (1, 1.5, true)
```

---

get_tool_analog_voltage_in (int : num)

函数说明:

该函数可读取机械臂末端的模拟电压输入，单位 V。

参数说明:

num : 机械臂末端的模拟电压通道序号，范围从 1-2。

返回值:

对应通道的模拟电压值。

示例:

```
value = get_tool_analog_voltage_in (1)
```

---

get_standard_analog_current_in (int : num)

函数说明:

该函数可读取控制柜上的模拟电流输入，单位 mA。

参数说明:

num : 控制柜上的模拟电流通道序号，范围从 1-4。

返回值:

对应通道的模拟电流值

示例:

```
value = get_standard_analog_current_in (1)
```

---

set_standard_analog_current_out (int : num, double : value, bool : block)

函数说明:

该函数可设置控制柜上的模拟电流输出。

参数说明:

num : 控制柜上的模拟电流通道序号，范围从 1-4;

value : 设置的模拟电流值，范围 [4,20]，单位 mA;

block : 指令是否阻塞型指令，如果为 false 表示非阻塞指令，指令会立即返回。

返回值:

当配置为阻塞执行，返回值代表当前任务结束时的状态，当配置为非阻塞执行，返回值代表当前任务的 id 信息，用户可以调用 get_noneblock_taskstate (id) 函数查询当前任务的执行状态。

示例:

```
value = set_standard_analog_current_out (1, 1.5,true)
```

---

get_standard_analog_current_out (int : num)

函数说明:

该函数可获取控制柜上的模拟电流输出。

参数说明:

num : 控制柜上的模拟电流通道序号，范围从 1-4;

返回值:

对应通道的模拟电流值。

示例:

```
value = get_standard_analog_current_out (1)
```

---

get_standard_analog_voltage_out (int : num)

函数说明:

该函数可获取控制柜上的模拟电压输出。

参数说明:

num : 控制柜上的模拟电压通道序号，范围从 1-4;

返回值:

对应通道的模拟电压值。

示例:

```
value = get_standard_analog_voltage_out (1)
```

---

write_reg(number: num, number: type, val)

函数说明:

该函数可修改内部寄存器的值。

参数说明:

num: 内部寄存器序号。

type: 修改的寄存器类型 1 为 bool 寄存器，2 为 word 寄存器，3 为 float 寄存器。

val: 根据 type 类型确定 val 类型。

当 type 为 1 时，val 类型为 boolean，true 表示真，false 表示假，num 范围为 1-64

当 type 为 2 时，val 类型为 number，并且为整数，范围 0-65535，num 范围为 1-32

当 type 为 3 时，val 类型为 number，num 范围为 1-32

参数错误时函数不改变内部寄存器数值。

返回值:

无

示例:

```
write_reg( 5, 1,true)
```

---

read_reg (number: num, number: type, number: in_out)

函数说明:

该函数可读取内部寄存器的值。

参数说明:

num: 内部寄存器序号。

type: 寄存器类型 1 为 bool 寄存器，2 为 word 寄存器，3 为 float 寄存器。

当 type 为 1 时，num 范围为 1-64。

当 type 为 2 时，num 范围为 1-32。

当 type 为 3 时，num 范围为 1-32。

参数错误时函数不改变内部寄存器数值。

in_out：为 0 时代表读取输入寄存器，1 代表读取输出寄存器。

返回值：

当 type 为 1 时，返回值类型为 boolean，true 表示真，false 表示假。

当 type 为 2 时，返回值类型为 number，并且为整数，范围为 0-65535。

当 type 为 3 时，返回值类型为 number。

示例：

```
ret=read_reg(10,1,1)
```

---

get_function_reg_in (number : portnum)

函数说明：

该函数可读取功能输入寄存器值。

参数说明：

portnum ：功能输入寄存器序号，范围从 1-16。

返回值：

boolean，返回 true 为真，false 为假。

示例：

```
ret= get_function_reg_in (1)
```

---

get_function_reg_out (number : portnum)

函数说明：

该函数可读取功能输出寄存器值。

参数说明：

portnum ：功能输出寄存器序号，范围从 1-16。

返回值：

boolean，返回 true 为真，false 为假。

示例：

```
ret= get_function_reg_out (1)
```

---

set_wobj_offset (boolean : val , number_list: wobj_offset)

函数说明:

基于当前的工件坐标系设置一个偏移量，后续的 move 类脚本的参考工件坐标系上都将添加这个偏移量。该偏移量以右乘的形式叠加在当前工件坐标系的初始值上。

参数说明:

val : true 为启用偏移量，false 为取消偏移量。

wobj_offset:{x, y, z, rx, ry, rz} 工件坐标系偏移量（单位: m, rad)。

返回值:

无

示例:

```
ret= set_wobj_offset (true, {0.1,0,0,0,0,0})
```

---

set_tool_data(string: name, number_list: tool_offset, number_list: load, number_list inertia_tensor)

函数说明:

该函数会设置当前生效工具信息，设置成功后，后续运动函数中的 TCP 设置为空（即使用当前坐标系）时，使用该工具参数。

参数说明:

name: 工具的名字，类型 string，最长不超过 32 个字节长度。

tool_offset: 工具 TCP 偏移量 {x_off, y_off,z_off,rx,ry,rz}，单位 (m, rad)。

load: 末端工具质量，质心，{mass,x_cog,y_cog,z_cog}，相对于法兰坐标系，单位 (kg, m)。

inertia_tensor: 末端工具惯量矩阵参数，参数 1-6 分别对应矩阵 xx、xy、xz、yy、yz、zz 元素，单位 kg*m^2，可缺省，缺省时默认为 0。

返回值:

无

示例:

```
ret= set_tool_data (“default”, {0.1,0,0,0,0,0},
{5,0,0.05,0},{0,0,0,0,0,0})
```

警告: 该函数只会临时修改当前控制器软件中生效的工具的信息。即使目标工具名称与当前机器人系统中已存在工具名称相同，也不会修改该工具的原始参数信息。当其他函数以名称的方式在此访问并使用该工具时，其相关信息仍为原始信息。若想要永久性修改目标工具信息，请使用 set_tool_data_workcell 代替该函数。

---

set_wobj(string: name, number_list: wobj_offset)

函数说明:

该函数会设置当前生效工件坐标系信息，设置成功后，后续运动函数中的 wobj 设置为空（即使用当前坐标系）时，使用该工件坐标系参数。

参数说明:

name：工件坐标系的名字，类型 string，最长不超过 32 个字节长度。

wobj_offset：工件坐标系偏移量 {x_off，y_off,z_off,rx,ry,rz}，单位(m，rad)。

返回值:

无

示例:

```
ret= set_wobj （“default”，{0.1,0,0,0,0,0})
```

警告： 该函数只会临时修改当前控制器软件中生效的工件坐标系信息。即使目标工件坐标系名称与当前机器人系统中已存在工件坐标系名称相同，也不会修改该工件坐标系的原始参数信息。当其他函数以名称的方式在此访问并使用该工件坐标系时，其相关信息仍为原始信息。若想要永久性修改目标工件坐标系信息，请使用 set_wobj_workcell 代替该函数。

---

set_load_data (number_list: load)

函数说明:

设置抓取负载。可以在程序运行过程中设置机器人当前除工具以外的负载信息（质量、质心)。

参数说明:

load：末端工具抓取负载质量，质心，{mass,x_cog,y_cog,z_cog}，相对于工具坐标系，质量范围 [0，35]，单位（kg，m)。

返回值:

无

示例:

```
ret= set_load_data ({5,0,0.05,0})
```

警告： 该函数会永久性改变当前除工具外的负载信息，即程序结束运行后仍然会保持。因此若出现运行该函数且程序停止运行后力控相关功能异常的情况，优先确认是否负载信息未被正确复位。

---

cal_ikine(pose : p, joints : q_near,number_list:tcp,number_list:wobj)

函数说明:

基于目标工具在工件坐标系中的笛卡尔空间位姿，通过机器人逆运动学计算机器人对应的关节角位置，在求解过程中，会选取靠近参考关节角位置或当前机械臂关节位置的解。

参数说明:

p：需要计算的末端位姿在设置工件坐标系的值，包含当前有效的工具偏移量，位置单位 m，姿态单位 rad。

q_near ：用于计算逆运动学的参考关节位置，为空时使用当前关节值。

tcp ：工具坐标系信息,tcp 偏移量 {x_off,y_off,z_off,rx,ry,rz},(单位：m，rad)，为空使用当前工具。

wobj：工件坐标系相对于世界坐标系的位移 {x，y，z，rx，ry，rz}，(单位：m，rad)，为空使用当前工件坐标系。

返回值:

返回的关节位置列表 {q1,q2,q3,q4,q5,q6}

示例:

```
ret= cal_ikine({0.492,0.140,0.442,-3.14,0,-1.57},{},{},{})
```

---

cal_fkine(joints:axis,number_list:tcp,number_list:wobj)

函数说明:

基于目标机器人关节角位置，通过机器人正运动学计算目标工具在目标工件坐标系中的笛卡尔空间位姿。

参数说明:

axis ：目标机器人关节角位置，单位 (rad)。

tcp ：目标工具坐标系信息,TCP 偏移量 {x_off,y_off,z_off,rx,ry,rz},(单位：m，rad)，为空使用当前工具。

wobj：目标工件坐标系相对于世界坐标系的偏移量 {x，y，z，rx，ry，rz}，(单位：m，rad)，为空使用当前工件。

返回值:

返回末端姿态列表 {x,y,z,rx,ry,rz}

示例:

```
ret= cal_fkine({0,0,1.57,0,-1.57,0},{},{})
```

---

get_tcp_pose()

函数说明:

该函数可获取当前状态下机械臂当前生效工具在基坐标系下的位姿。

参数说明:

无

返回值:

返回末端位置 pose。

示例:

```
ret= get_tcp_pose()
```

---

get_tcp_pose_coord(string : tool, string : wobj)

函数说明:

该函数可获取末端法兰在目标工具坐标系参考目标工件坐标系下的位姿。

参数说明:

tool: 工具坐标系名称，默认为当前使用的坐标系。

wobj: 工件坐标系名称，默认为当前使用的坐标系。

返回值:

末端法兰的位姿，单位 (m, rad)。

示例:

```
ret= get_tcp_pose_coord ("tool_1", "wobj _1")
```

---

get_tcp_speed()

函数说明:

该函数可获取当前状态下机械臂当前生效工具末端原点的空间速度。

参数说明:

无

返回值:

末端速度列表，单位 m/s,rad/s。

示例:

```
ret= get_tcp_speed()
```

---

get_tcp_acceleration()

函数说明:

该函数可获取当前状态下机械臂工具当前生效工具末端原点的空间加速度。

参数说明:

无

返回值:

末端加速度列表，单位 (m/s2, rad/s2)。

示例:

```
ret= get_tcp_acceleration()
```

---

get_tcp_force()

函数说明:

该函数可获取当前机械臂工具当前生效工具末端原点的空间力信息。该函数当且仅当安装了末端力传感器并正确配置启用时有效。

参数说明:

无

返回值:

返回末端力矩信息，{Fx,Fy,Fz,Mx,My,Mz}，单位 N、N.m。

示例:

```
ret= get_tcp_force()
```

---

get_tcp_force_tool(string : tool)

函数说明:

该函数可获取机械臂工具末端在目标工具坐标系下的力矩信息。该函数当且仅当安装了末端力传感器并正确配置启用时有效。

参数说明:

tool：工具坐标系名称，默认为当前使用的坐标系。

返回值:

返回工具坐标系下的力矩信息，{Fx,Fy,Fz,Mx,My,Mz}，单位 (N、N.m)。

示例:

```
ret= get_tcp_force_tool ("tool_1")
```

---

get_tcp_offset()

函数说明:

该函数可获取当前状态下机械臂有效的末端工具的偏移量。

参数说明:

无

返回值:

{x_off, y_off,z_off,rx,ry,rz} 返回 TCP 偏移量信息，单位 (m, rad)。

示例:

```
ret= get_tcp_offset()
```

---

get_tool_load()

函数说明:

该函数可获取当前设置工具的有效负载质量、质心位置和惯性张量。

参数说明:

无

返回值:

质量单位 kg，质心位置单位 (m)，{mass,x_cog,y_cog,z_cog,t_1,t_2,t_3,t_4,t_5,t_6,}。

示例:

```
ret= get_tcp_load()
```

---

get_wobj()

函数说明:

该函数可获取当前设生效的工件坐标系的值。

参数说明:

无

返回值:

{x, y, z, rx, ry, rz} 工件坐标系相对于世界坐标系的偏移量，单位 (m, rad)。

示例:

```
ret= get_wobj()
```

---

get_actual_joints_position()

函数说明:

该函数可获取当前状态下机械臂各关节位置角度。

参数说明:

无

返回值:

返回 1-6 轴关节角度列表，单位 (rad)。

示例:

```
ret= get_actual_joints_position()
```

---

get_target_joints_position()

函数说明:

该函数可获取当前状态下机械臂各关节位置指令角度。

参数说明:

无

返回值:

返回 1-6 轴关节角度列表，单位 (rad)。

示例：

```
ret= get_target_joints_position()
```

---

get_actual_joints_speed()

函数说明：

该函数可获取当前状态下机械臂各关节角速度。

参数说明：

无

返回值：

返回 1-6 轴关节速度列表，单位 (rad/s)。

示例：

```
ret= get_actual_joints_speed()
```

---

get_target_joints_speed()

函数说明：

该函数可获取当前状态下机械臂各关节角速度指令。

参数说明：

无

返回值：

返回 1-6 轴关节速度列表，单位 (rad/s)。

示例：

```
ret= get_target_joints_speed()
```

---

get_actual_joints_acceleration()

函数说明：

该函数可获取当前状态下机械臂各关节角加速度。

参数说明：

无

返回值：

返回 1-6 轴关节加速度列表，单位 (rad/ s2)。

示例：

```
ret= get_actual_joints_acceleration()
```

---

get_target_joints_acceleration()

函数说明:

该函数可获取当前状态下机械臂各关节角加速度指令。

参数说明:

无

返回值:

返回 1-6 轴关节加速度列表，单位 (rad/ s2)。

示例:

```
ret= get_target_joints_acceleration()
```

---

get_actual_joints_torque()

函数说明:

该函数可获取当前状态下机械臂各关节力矩。

参数说明:

无

返回值:

返回 1-6 轴关节力矩列表，单位 (N.m)。

示例:

```
ret= get_actual_joints_torque()
```

---

get_target_joints_torque()

函数说明:

该函数可获取当前状态下机械臂各关节力矩指令。

参数说明:

无

返回值:

返回 1-6 轴关节力矩列表，单位 (N.m)。

示例:

```
ret= get_target_joints_torque()
```

---

get_flange_pose()

函数说明:

该函数可获取当前状态下机械臂末端法兰在基坐标系下的位姿。

参数说明:

无

返回值:

末端法兰位置 pose。

示例:

```
ret= get_flange_pose()
```

---

get_flange_speed()

函数说明:

该函数可获取当前状态下机械臂末端法兰在基坐标系下的速度。

参数说明:

无

返回值:

末端法兰速度列表，单位 (m/s, rad/s)。

示例:

```
ret= get_flange_speed()
```

---

get_flange_acceleration()

函数说明:

该函数可获取当前状态下机械臂末端法兰在基坐标系下的加速度。

参数说明:

无

返回值:

末端法兰加速度列表，单位 (m/ s2,rad/ s2)。

示例:

```
ret= get_flange_acceleration()
```

---

sub_program(string : program_name, string: tree_name)

函数说明:

调用子程序。

参数说明:

program_name : 子程序名。

tree_name: 子程序树节点名称，显示使用。

返回值:

无

示例:

```
sub_program("program1.jspf", " ")
```

---

start_record_track (string : name, number : mode, string : tool, string : wobj)

函数说明:

该函数开启轨迹记录，当超过允许记录的轨迹长度（针对基于位置记录）或允许记录的时长时（针对基于时间记录），会自动停止文件记录，并且暂停当前运行的程序。文件会记录机器人的各关节弧度值和选定工具、工件坐标系下的笛卡尔空间位姿。

参数说明:

name ：轨迹名称。

mode ：轨迹类型，mode=0 基于位置记录（与上一记录点所有关节偏移总量到达 5° 时记录新点）；mode=1 基于时间记录（与上一记录点间隔 250ms 记录新点）。

tool ：工具坐标系名称。

wobj ：工件坐标系名称。

返回值:

无

示例:

```
start_record_track (“track”,0,”default”,”default”)
```

---

stop_record_track()

函数说明:

该函数停止轨迹记录。

参数说明:

无。

返回值:

无

示例:

```
stop_record_track()
```

---

collision_detect (number:level)

函数说明:

设置碰撞检测等级。

参数说明:

level:0：关闭碰撞检测，1-5：对应设置碰撞检测等级 1 到等级 5。

返回值:

无

示例:

```
collision_detect (1)
```

---

set_value (string:name,var:data)

函数说明:

设置系统变量值。

参数说明:

name: 系统变量名称

var: 设置的系统变量值，根据对应的系统变量类型确定其类型

当系统变量类型为:

boolean,data 类型为 boolean;

number, data 类型为 number;

string, data 类型为 string;

number_list, data 类型为 number_list;

pose, data 类型为 number_list;

joints, data 类型为 number_list;

返回值:

无

示例:

```
set_value (“g_data”,true)
```

---

get_value (string:name)

函数说明:

获取系统变量值。

参数说明:

name: 系统变量名称

返回值:

ret: 根据对应的系统变量类型确定其返回类型，当系统变量类型为:

boolean, ret 类型为 boolean;

number, ret 类型为 number;

string, ret 类型为 string;

number_list, ret 类型为 number_list;

pose, ret 类型为 number_list;

joints, ret 类型为 number_list;

示例:

```
ret = get_value (“g_data”)
```

---

timer_start()

函数说明:

开始/重置计时器。

参数说明:

无

返回值:

无

示例:

```
timer_start()
```

---

timer_end (string : timer_list)

函数说明:

计算时间间隔。

参数说明:

timer_list: 存放时间间隔的 timer 类型系统变量的名称。

返回值:

无

示例:

```
timer_end ("g_timer_1")
```

---

get_robot_state()

函数说明:

获取机器人状态信息。

参数说明:

无。

返回值:

number_list: 机器人状态信息列表，number_list[1] 表示机器人状态，number_list[2] 表示程序状态，number_list[3] 表示安全控制器状态，number_list[4] 表示操作模式。

示例:

```
get_robot_state()
```

---

get_program_state()

函数说明:

获取程序状态信息。

参数说明:

无。

返回值:

程序状态信息。0 ： 程序停止，1 ： 程序正在停止中，2 ： 程序正在运行，3 ： 程序已经暂停，4 ： 程序暂停中，5 ： 手动示教任务执行中。

示例:

```
get_program_state()
```

---

get_safety_state()

函数说明:

获取机器人安全状态信息。

参数说明:

无。

返回值:

机器人安全状态信息。0 ： 初始化，2 ： 等待，3 ： 配置模式，4 ： 下电状态，5 ： 正常运行状态，6 ： 恢复模式，7 ： Stop2，8 ： Stop1，9 ： Stop0，10 ： 模型配置状态，12 ： 缩减模式状态，13 ： 引导，14 ： 致命错误状态，15 ： 更新状态。

示例:

```
get_safety_state()
```

---

get_operation_mode()

函数说明:

获取手自动模式。

参数说明:

无。

返回值:

手自动模式。0 ： 手动，1 ： 自动。

示例:

```
get_operation_mode()
```

---

get_collision_state()

函数说明:

获取机器人碰撞触发状态。

参数说明:

无。

返回值:

机器人碰撞触发状态。0 ： 碰撞未触发，1 ： 碰撞触发。

示例:

```
get_collision_state()
```

---

get_robot_protective_stop_state()

函数说明:

获取机器人防护性停止状态。

参数说明:

无。

返回值:

机器人防护性停止状态。0： 防护性停止未触发，1： 防护性停止触发。

示例:

```
get_robot_protective_stop_state()
```

---

get_robot_automode_protective_stop_state()

函数说明:

获取机器人自动模式防护性停止状态。

参数说明:

无。

返回值:

机器人自动模式防护性停止状态。0： 自动模式防护性停止未触发，1： 自动模式防护性停止触发。

示例:

```
get_robot_automode_protective_stop_state()
```

---

get_robot_in_reduce_mode_state()

函数说明:

机器人是否处于缩减模式状态。

参数说明:

无。

返回值:

机器人缩减模式状态。0: 正常模式，1: 缩减模式。

示例:

```
get_robot_in_reduce_mode_state()
```

---

get_robot_at_home_position_state()

函数说明:

机器人是否处于安全 home 位置。

参数说明:

无。

返回值:

是否处于安全 home 位置。0: 未处于安全 home 位置，1: 处于安全 home 位置。

示例:

```
get_robot_at_home_position_state()
```

---

get_robot_safety_in(number: portnum)

函数说明:

获取指定通道的安全输入信号。

参数说明:

portnum: 安全信号输入通道。范围: [1, 2]。

返回值:

安全输入信号。0: 低电平，1: 高电平。

示例:

```
get_robot_safety_in(1)
```

---

get_robot_safety_out(number: portnum)

函数说明:

获取指定通道的安全输出信号。

参数说明:

portnum: 安全信号输出通道。范围: [1, 2]。

返回值:

安全输出信号。0: 低电平，1: 高电平。

示例:

```
get_robot_safety_out(1)
```

is_moving()

函数说明:

判断当前机器臂是否在运动。

参数说明:

无。

返回值:

机械臂运动时返回 `true`; 机械臂不运动时返回 `false`。

示例:

```
is_moving()
```

4.5.5 调试相关

log(string: describe, val)

函数说明:

该函数可在程序中插入 `log` 日志，记录程序运行中的相关信息。

参数说明:

`describe` : 需要在日志中记录的描述信息。

`val`: 需要在 `describe` 信息后记录的变量值信息，会自动转换为 `string` 类型并被记录，可以是 `number,string,list` 等任意类型。

返回值:

无

示例:

```
log("this is log",{1,2,3,4})
```

---

message(string: describe)

函数说明:

该函数可在程序运行过程中产生弹窗，显示目标描述信息，并暂停当前程序。

参数说明:

`describe` : 弹窗所显示的目标描述信息。

返回值:

无

示例:

```
message ("this is message")
```

4.5.6 Modbus

modbus_read(string : signal_name)

函数说明:

该函数可读取 modbus 节点的数据，返回值为 number 类型。

参数说明:

signal_name: modbus 节点名。

返回值:

number

示例:

```
val=modbus_read(" mbus1")
```

---

modbus_write(string : signal_name, number : val)

函数说明:

该函数可对 modbus 节点进行写操作。

参数说明:

signal_name: modbus 节点名。

val: 要写入的数值，寄存器节点取值为 0-65535 内的整数，线圈节点取值为 0 或 1。

返回值:

无

示例:

```
modbus_write(" mbus1", 1)
```

---

modbus_set_frequency(string : signal_name, number: frequence)

函数说明:

该函数可修改 modbus 节点的刷新频率，默认频率为 10Hz。

参数说明:

signal_name: modbus 节点名。

frequence: 频率值，取值范围: 1~100Hz。

返回值:

无

示例:

```
modbus_set_frequency (" mbus1", 20)
```

---

modbus_write_multiple_regs(number : slave_num, string : name, number : len, number_list : word_list)

函数说明:

该函数可对多寄存器进行写操作。

参数说明:

slave_num : modbus 节点号。

name : modbus 节点名。

len : 需要写入数据的寄存器长度。

word_list: 需要写入的数据。

返回值:

无

示例:

```
modbus_write_multiple_regs (1, " mbus1", 5, {1,2,3,4,5})
```

---

modbus_write_multiple_coils(number : slave_num, string : name, number : len, number_list :byte_list)

函数说明:

该函数可对多线圈进行写操作。

参数说明:

slave_num : modbus 节点号。

name : modbus 节点名。

len : 需要写入数据的线圈长度。

byte_list: 需要写入的数据。

返回值:

无

示例:

```
modbus_write_multiple_coils (2, " mbus1", 5, {1,1,1,1,1})
```

4.5.7 数学运算函数

abs (number: number)

函数说明:

计算绝对值。

参数说明:

number: number 数据类型。

返回值:

number 类型。

示例:

```
ret = abs (2)
```

---

cos(number: number)

函数说明:

计算余弦值。

参数说明:

number: number 数据类型，单位 (rad)。

返回值:

number 类型。

示例:

```
ret = cos (1.57)
```

---

acos(number: number)

函数说明:

计算反余弦值。

参数说明:

number: number 数据类型，余弦值，取值范围 [-1,1]。

返回值:

number 类型，单位 (rad)。

示例:

```
ret = acos (0.5)
```

---

sin(number: number)

函数说明:

计算正弦值。

参数说明:

number: number 数据类型，单位 (rad)。

返回值:

number 类型。

示例:

```
ret = sin (1.57)
```

---

asin(number: number)

函数说明:

计算反正弦值。

参数说明:

number: number 数据类型，正弦值，取值范围 [-1,1]。

返回值:

number 类型，单位 (rad)。

示例:

```
ret = asin (0.5)
```

---

tan(number: number)

函数说明:

计算正切值。

参数说明:

number: number 数据类型，单位 (rad)。

返回值:

number 类型。

示例:

```
ret = tan (1.57)
```

---

atan(number: number)

函数说明:

计算反正切值。

参数说明:

number: number 数据类型，正切值。

返回值:

number 类型，范围 (-pi/2,+pi/2)，单位 (rad)。

示例:

```
ret = atan (0.5)
```

---

atan2(number: number1, number: number2)

函数说明:

计算 number1/number2 的反正切值，根据 number1、number2 的符号，确定返回值所在的象限。

参数说明:

number1: number 数据类型。

number2: number 数据类型。

返回值:

number 类型，范围（-pi,pi），单位（rad）。

示例:

```
ret = atan2 (1,2)
```

---

pow(number: base,number: exponent)

函数说明:

幂运算。

参数说明:

base: 底数。

exponent: 指数。

注: 如果 base 为负值并且 exponet 不是一个整数值，则发生定义域错误。当 base 为 0 且 exponent 小于 0 时，会发生定义域错误。

返回值:

number 类型。

示例:

```
ret = pow (10,2)
```

---

sqrt(number: number)

函数说明:

计算平方根。

参数说明:

number: number 数据类型，取值大于等于 0。

返回值:

number 类型。

示例:

```
ret = sqrt (10)
```

---

deg2rad(number: number)

函数说明:

角度值转换成弧度值。

参数说明:

number: number 数据类型，单位度。

返回值:

number 类型，单位 (rad)。

示例:

```
ret = deg2rad (90)
```

---

rad2deg(number: number)

函数说明:

弧度值转换成角度值。

参数说明:

number: number 数据类型，单位 (rad)。

返回值:

number 类型，单位度。

示例:

```
ret = rad2deg (1.57)
```

---

pose_trans (pose: p1,pose: p2)

函数说明:

坐标变换函数，位姿 p1，经过 p2 变换得到一个新的位姿，返回的位姿 ret=p1*p2。

参数说明:

p1:pose 类型，姿态信息按照 Rz*Ry*Rx 方式计算旋转矩阵，位置单位 m，姿态单位 (rad)。

p2:pose 类型，姿态信息按照 Rz*Ry*Rx 方式计算旋转矩阵，位置单位 m，姿态单位 (rad)。

返回值:

pose 类型，经过坐标变换后的位姿，位置单位 m，姿态单位 (rad)。

示例:

```
ret = pose_trans ({492,140.5,442,-3.14,0,-1.57},
{231.4,140.5,582.3,-3.14, 0,-1.57} )
```

---

pose_inv (pose: p1)

函数说明:

坐标逆变换函数，求解一个变换的逆变换。

参数说明:

p1:pose 类型，姿态信息按照 Rz*Ry*Rx 方式计算旋转矩阵，位置单位 m，姿态单位 (rad)。

返回值:

pose 类型，逆变换的解。姿态信息按照 Rz*Ry*Rx 方式计算旋转矩阵，位置单位 m，姿态单位 (rad)。

示例:

```
ret = pose_inv ({492,140.5,442,-3.14,0,-1.57} )
```

---

pose_distance(pose: p1, pose: p2)

函数说明:

计算两点之间的空间距离，姿态不参与计算。

参数说明:

p1：要计算的点 1。

p2：要计算的点 2。

返回值:

number：两点间的距离。

示例:

```
ret = pose_distance ({492,140.5,442,-3.14,0,-1.57},
{231.4,140.5,582.3, -3.14,0,-1.57} )
```

---

pose_offset(pose: p1,pose: p2)

函数说明:

计算两个位姿之间的偏移量，计算方式为 ret=$p1^{-1}$*p2。

参数说明:

p1：要计算的点 1。

p2：要计算的点 2。

返回值:

两个位姿之间的偏移量。

示例:

```
ret = pose_offset ({492,140.5,442,-3.14,0,-1.57},
{231.4,140.5,582.3, -3.14,0,-1.57} )
```

---

num2str (number: num, number: precision)

函数说明:

将数值按照指定的精度转换为字符串。比如 num2str(1.23456, 3) 转换成“1.235”，num2str(1.23) 转换成“1.230000”。

参数说明:

num：要转换的数值。

precision：保留的精度，可缺省，默认为 6。

返回值：

string

示例：

```
ret = num2str (1.234, 2)
```

---

str2num (string:s)

函数说明：

将字符串转换成数值，比如“1.23”转换成 1.23。

参数说明：

s：需要转译的字符串。

返回值：

number 类型，转译后的数。

示例：

```
ret = str2num (“1.234”)
```

---

list2str (number_list:a)

函数说明：

将 number 类型列表转换为固定格式字符串，字符串以“(”开头，“)”结尾，中间的数据之间使用“,”分隔。比如：列表 {1.23,1.57,2.3} 会转换成字符串“(1.23,1.57,2.3)”。

参数说明：

a ： number 类型的列表。

返回值：

string

示例：

```
ret = list2str ({1.2,3.4,5.6})
```

---

str2list(string:s)

函数说明：

将字符串转换为列表，字符串以“(”开头，“)”结尾，中间的数据之间使用“,”分隔。比如：字符串“(1.23,1.57,2.3)”会转换成列表 {1.23,1.57,2.3}。

参数说明：

s ： 需要转换的格式化字符串。

返回值：

number_list，如果转换错误，则返回空列表。

示例:

```
ret = str2list ("(1.2,3.4,5.6)")
```

---

get_pose_trans(pose:p)

函数说明:

获取机器人位姿中关于位置的偏移量。

参数说明:

p : pose 类型，机器人位姿数据，单位：m、rad。

返回值:

number_list，返回 pose 的 3 维位置偏移量信息，单位：m。

示例:

```
ret = get_pose_trans ({1,2,3,4,5,6})
```

期望结果:ret={1,2,3}

---

get_pose_rpy(pose:p)

函数说明:

获取机器人位姿中关于姿态的偏移量。

参数说明:

p: pose 类型，机器人位姿数据，单位：m、rad。

返回值:

number_list，返回 pose 的 3 维姿态偏移量信息，单位：rad。

示例:

```
ret = get_pose_rpy({1,2,3,4,5,6})
```

期望结果:ret={4,5,6}

---

get_number_list_norm(number_list : nl)

函数说明:

获取 nl 中所有数据的模长，方式为取 nl 所有元素的平方和。

参数说明:

nl : number_list 类型，输入参数。

返回值:

number，输入参数的模长。

示例:

```
ret = get_number_list_norm ({1,2,3,4,5,6})
```

---

normalize_number_list(number_list:nl)

函数说明:

将输入的 nl 归一化并返回输出，方式为 nl 的所有元素除以其模长。

参数说明:

nl : number_list 类型，输入参数。

返回值:

number_list，归一化后的数据。

示例:

```
ret = normalize_number_list ({1,2,3,4,5,6})
```

---

number_list_cross(number_list : nl1, number_list : nl2)

函数说明:

计算两个维度为 3 的 number_list 的叉乘，计算方式为 nl1×nl2。

参数说明: nl1 : number_list 类型，输入参数，维度为 3。

nl2 : number_list 类型，输入参数，维度为 3。

返回值:

number_list，叉乘结果。

示例:

```
ret = number_list_cross ({1,2,3},{4,5,6})
```

---

number_list_cross(list : nl)

函数说明:

计算的 nl 中所有 number_list 的叉乘。

参数说明:

nl : list 类型，{number_list: nl1, number_list: nl2, ...}，所有 number_list 的维度必须大于等于 3 且相等，且 number_list 的参数个数必须为 number_list 的维度-1。

返回值:

number_list，叉乘结果，维度与 nl 的元素的维度相同。

示例:

```
ret = number_list_cross({{1,2,3},{4,5,6})
```

---

number_list_dot(number_list : nl1, number_list : nl2)

函数说明:

计算 nl1 和 nl2 的点积。

参数说明:

nl1: number_list 类型，输入参数。

nl2: number_list 类型，维度与 nl1 相同。

返回值:

number，nl1 · nl2 的值。

示例:

```
ret = number_list_dot ({1,2,3,4,5,6},{1,2,3,4,5,6})
```

---

rpy_to_axial_angle(number_list : rpy)

函数说明:

计算输入 rpy 所对应的轴角，rpy 默认为动态 ZYX。

参数说明:

rpy : number_list 类型，默认参考动态 ZYX，单位 rad，必须为 3 维。

返回值:

number_list，rpy 所对应的轴角，维度为 4 维，第 1 个值对应参考旋转轴矢量的旋转角度，单位 rad，后三个值为所对应的旋转轴矢量。

示例:

```
ret = rpy_to_axial_angle({1,2,3})
```

---

rpy_to_rot(number_list : rpy)

函数说明:

计算输入 rpy 所对应的旋转矩阵，rpy 默认为动态 ZYX。

参数说明:

rpy : number_list 类型，默认参考动态 ZYX，单位 rad，必须为 3 维。

返回值:

list，{number_list, number_list, number_list}，按 RX、RY、RZ 的顺序返回 rpy 所对应的 3 组正交基，组成对应的旋转矩阵。

示例:

```
ret = rpy_to_rot ({1,2,3})
```

---

rot_to_rpy(list : rot)

函数说明:

计算输入旋转矩阵 rot 所对应 rpy，rpy 默认为动态 ZYX。

参数说明:

rpy : list 类型,{number_list, number_list, number_list},按 RX、RY、RZ 的顺序输入三组正交基，numberl_list 必须为 3 维。

返回值:

number_list，RX、RY、RZ 值。

示例:

```
ret = rot_ro_rpy ({{1,2,3},{4,5,6},{7,8,9}})
```

4.5.8 字符串相关函数

str_cat(string : str1, string : str2)

函数说明:

拼接两个字符串。

参数说明:

str1 ： 要拼接的左侧字符串。

str2 ： 要拼接的右侧字符串。

返回值:

拼接后的字符串。

示例:

```
ret = str_cat （“a”，”b”）
```

---

str_cmp(string : str1, string : str2)

函数说明:

比较两个字符串是否相等，系统支持的字符串最大长度为 255。

参数说明:

str1 ： 要比较的字符串 1。

str2 ： 要比较的字符串 2。

返回值:

false: 不相等。

true: 相等。

示例:

```
ret = str_cmp （“a”，”b”）
```

---

str_del(string : str, number: index, number: num)

函数说明:

删除字符串中的一段。

参数说明:

str ： 要进行操作的字符串。

index：表示从第几位开始删除，从 0 开始计数。

num：表示删除几位。

返回值：

删除后的字符串。

示例：

```
ret = str_del (“abcde”, 2, 2)
```

---

str_insert(string : str1, number: index, number:num, string : str2)

函数说明：

在一个字符串中插入另一个字符串。

参数说明：

str1 ：进行操作的基准字符串。

index：从基准字符串哪一位开始插入，从 0 开始计数。

num：插入多少位。

str2 ：插入的字符串。

返回值：

操作后的字符串。

示例：

```
ret = str_insert (“abcde”, 2, “fff”)
```

---

str_substr(string: str, number: index, number: num)

函数说明：

取字符串的子集。

参数说明：

str ：要操作的字符串。

index：从哪一位开始取，从 0 开始计数。

num：取多少位。

返回值：

操作后的字符串。

示例：

```
ret = str_substr (“abcde”, 2, 2)
```

---

str_len(string : str)

函数说明:

字符串长度。

参数说明:

str1 : 要计算长度的字符串。

返回值:

number，字符串长度。

示例:

```
ret = str_len (“abcde”)
```

---

str_find(string : str1, string : str2)

函数说明:

字符串 str1 中首次出现 str2 的位置。

参数说明:

str1 : 要操作的字符串。

str2 : 要操作的字符串。

返回值:

number: 返回位置，从 0 开始计数，未找到返回-1。

示例:

```
ret = str_find (“abcde”, “ab”)
```

---

str_split(string : str1, string : separator)

函数说明:

按照字符 separator 分割 str1 字符串，返回列表。

参数说明:

str1 : 要操作的字符串。

separator: 分割字符。

返回值:

num_list，字符串列表。

示例:

```
ret = str_split (“abcde”,” c”)
```

---

str_at(string : str1, number : index)

函数说明:

获取字符串 str1 中 index 处的字符，index 从 1 开始。

参数说明:

str1 : 要操作的字符串。

index : 索引，从 1 开始计数。

返回值:

string，index 超出范围返回空字符串。

示例:

```
ret = str_at (“abcde”, 2)
```

4.5.9 辅助函数

list_len (list:value)

函数说明:

该函数返回输入列表的长度。

参数说明:

value: 输入的列表。

返回值:

number: 列表的长度。

示例:

```
ret = list_len ({1,1,1,1,1,1})
```

---

is_valid (value)

函数说明:

该函数判断输入的条件是否合法。

参数说明:

value: 输入的数据，会根据数据类型自动推断，判断逻辑如下

当 value 为 boolean 类型时，true 为合法，false 为非法

当 value 为 list 类型时，列表非空为合法，列表为空为非法

当 value 为 string 类型时，字符非空为合法，字符为空字符串为非法

当 value 为 number 类型时，数字大小非 0 为合法，打字大小为 0 为非法

当 value 为 nil 类型时为非法。

返回值:

false: 非法。

true: 合法。

示例:

```
ret = is_valid ("")
```

---

int16_to_byte_list (number:value)

函数说明:

将 int16 数据类型按照内存结构转换成 byte list。

参数说明:

value: 待转换的 int16 类型数据。

返回值:

转换得到的 byte 数据列表。

示例:

```
ret = int16_to_byte_list (5)
```

---

uint16_to_byte_list (number:value)

函数说明:

将 uint16 数据类型按照内存结构转换成 byte list。

参数说明:

value: 待转换的 uint16 类型数据。

返回值:

转换得到的 byte 数据列表。

示例:

```
ret = uint16_to_byte_list (5)
```

---

int32_to_byte_list (number:value)

函数说明:

将 int32 数据类型按照内存结构转换成 byte list。

参数说明:

value: 待转换的 int32 类型数据。

返回值:

转换得到的 byte 数据列表。

示例:

```
ret = int32_to_byte_list (5)
```

---

uint32_to_byte_list (number:value)

函数说明:

将 uint32 数据类型按照内存结构转换成 byte list。

参数说明:

value: 待转换的 uint32 类型数据。

返回值:

转换得到的 byte 数据列表。

示例:

```
ret = uint32_to_byte_list (5)
```

---

float_to_byte_list (number:value)

函数说明:

将 float 数据类型按照内存结构转换成 byte list。

参数说明:

value: 待转换的 float 类型数据。

返回值:

转换得到的 byte 数据列表。

示例:

```
ret = float_to_byte_list (5.0)
```

---

double_to_byte_list (number:value)

函数说明:

将 double 数据类型按照内存结构转换成 byte list。

参数说明:

value: 待转换的 double 类型数据。

返回值:

转换得到的 byte 数据列表。

示例:

```
ret = int16_to_byte_list (5.0005)
```

---

byte_list_to_int16 (byte_list:value)

函数说明:

将 byte list 数据类型按照内存结构转换成 int16 类型数据。

参数说明:

value: 待转换的 byte list(number list) 类型数据。

返回值:

转换得到的 int16 类型数据。

示例:

```
ret = byte_list_to_int16 ({0x01,0x05})
```

---

byte_list_to_uint16 (byte_list:value)

函数说明:

将 byte list 数据类型按照内存结构转换成 uint16 类型数据。

参数说明:

value: 待转换的 byte list(number list) 类型数据。

返回值:

转换得到的 uint16 类型数据。

示例:

```
ret = byte_list_to_uint16 ({0x01,0x05})
```

---

byte_list_to_int32 (byte_list:value)

函数说明:

将 byte list 数据类型按照内存结构转换成 int32 类型数据。

参数说明:

value: 待转换的 byte list(number list) 类型数据。

返回值:

转换得到的 int32 类型数据。

示例:

```
ret = byte_list_to_int32 ({0x01,0x05})
```

---

byte_list_to_uint32 (byte_list:value)

函数说明:

将 byte list 数据类型按照内存结构转换成 uint32 类型数据。

参数说明:

value: 待转换的 byte list(number list) 类型数据。

返回值:

转换得到的 uint32 类型数据。

示例:

```
ret = byte_list_to_uint32 ({0x01,0x05})
```

---

byte_list_to_float (byte_list:value)

函数说明:

将 byte list 数据类型按照内存结构转换成 float 类型数据。

参数说明:

value: 待转换的 byte list(number list) 类型数据。

返回值:

转换得到的 float 类型数据。

示例:

```
ret = byte_list_to_float ({0x01,0x05})
```

---

byte_list_to_double (byte_list:value)

函数说明:

将 byte list 数据类型按照内存结构转换成 double 类型数据。

参数说明:

value: 待转换的 byte list(number list) 类型数据。

返回值:

转换得到的 double 类型数据。

示例:

```
ret = byte_list_to_double ({0x01,0x05})
```

---

float2word (number:value)

函数说明:

将 float 数据类型按照内存结构转换成长度为 2 的 word 列表类型数据。

参数说明:

value: 待转换的 float 类型数据。

返回值:

转换得到的 word 列表 {word_H, word_L}。

示例:

```
ret = float2word (5.5)
```

---

word2float(number:word_H, number:word_L)

函数说明:

将两个 word 类型数据按照内存结构转换成一个 float 类型数据。

参数说明:

word_H, word_L: 待合并转换的 word 类型数据。

返回值:

转换得到的 float 类型数据。

示例:

```
ret = word2float (5)
```

4.5.10 力控函数

fc_start()

函数说明:

该指令控制机械臂开启机器人末端力控。开启末端力控后所有运动函数除正常运动外，会额外基于已配置的末端力控参数叠加末端力控运动实现力位混合运动。

参数说明:

无

返回值:

无

示例:

```
fc_start ()
```

警告: 在使用 fc_start 函数启用末端力控功能前，需要严格确保机器人末端已正常安装力传感器，并在正确完成其通讯配置及安装位置设置启用。错误的通讯配置或未正确启用传感器会导致使用 fc_start 函数后机器人报错。错误的传感器安装位置参数会导致异常的末端力控运动，从而造成安全风险。

---

fc_stop()

函数说明:

该指令控制机械臂退出末端力控，结束力位混合控制。

参数说明:

无

返回值:

无

示例:

```
fc_stop ()
```

警告: 当前在调用 fc_stop 函数退出末端力控前，需要严格保证所有机器人绝对位置运动已结束，即所有来自脚本以及外部接口所触发的机器人运动已结束。机器人运动过程中能够使用 fc_stop 函数结束末端力控，会引起机器人产生异常运动并报错。

fc_move()

函数说明:

该指令控制机械臂没有绝对位置运动的基础上产生仅基于末端力控产生的补偿运动。

参数说明:

无

返回值:

无

示例:

```
fc_move ()
```

fc_config(number_list: direction, number_list: ref_ft, number_list: damp, number_list: max_vel, string: tool, string: wobj, number: type, number_list:ft_deadzone)

函数说明:

该指令修改并配置机器人末端力控参数。

参数说明:

direction:: 6 个笛卡尔空间方向末端力控开关，开为 true，关为 false。开启力控的笛卡尔空间方向，除了机器人末端原有绝对运动外，同时还会根据末端传感器感知到的外部力，基于其他力控参数做出对应力控运动调整。未开启力控的笛卡尔空间方向则保持原始机器人末端绝对运动。

ref_ft: 6 个笛卡尔空间方向末端力控目标维持力，范围 [-1000, 1000]，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm，方向符号参考末端力控参考坐标系方向。该目标维持力会使机器人末端力控产生额外运动，直到机器人末端传感器感知到与外界接触力且达到目标维持力的大小。

damp: 6 个笛卡尔空间方向末端力控阻尼，X/Y/Z 方向单位 (N/(m/s)，RX/RY/RZ 方向单位 (Nm/(rad/s))。末端力控所产生的叠加运动最大速度，会根据传感器在笛卡尔空间启用力控方向上感知到的外力与目标维持力之间的偏差值，除以对应方向的阻尼值计算得到。

max_vel: 6 个笛卡尔空间方向末端力控最大调整速度，范围 [-5, 5]，X/Y/Z 方向单位 (m/s)，RX/RY/RZ 方向单位 (rad/s)。若基于末端传感器感知到的外力与目标维持力的偏差除以阻尼后计算得到的速度超过力控最大调整速度，则会以最大调整速度进行末端力控运动。该参数可以使末端力控运动在使用较小阻尼提升响应的基础上降低调整最大速度，从而保证末端力控运动的安全及稳定性。

tool ： 设置使用的末端力控工具的名称，默认为当前使用的工具。所有末端力控运动所产生的叠加运动，都会以 tool 坐标系的原点产生。若同时末端力控参考坐标系配置为工具坐标系，则力控运动笛卡尔方向的定义参考 tool 坐标系的笛卡尔方向。

wobj ： 设置使用的末端力控工件坐标系的名称，默认为当前使用的工件坐标系。若末端力控参考坐标系配置为工件坐标系，则力控运动笛卡尔方向的定义参考 wobj 坐标系的笛卡尔方向。

type：末端力控参考坐标系选择标志位，0 为参考工具坐标系，1 为参考工件坐标系。

number_list:6 个笛卡尔空间方向末端接触力死区，范围 [-1000, 1000]，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm。若末端力传感器感知到外力与目标维持力的偏差绝对值小于接触力死区时，则会停止对于力控运动，直到由于外力变化再次使偏差值大于死区。

返回值：

无

示例：

```
fc_config ({true,false,false,false,false,false},{0.1,0,0,0,0,0},{1000,1000,
1000,57.29,57.29,57.29},{0.15,0.15,0.15,1.04,1.04,1.04},"","",
0, {0,0,0,0,0,0})
```

---

fc_guard_deact()

函数说明：

该指令控制机械臂在末端力控过程中禁用力安全力监控。

参数说明：

无

返回值：

无

示例：

```
fc_guard_deact ()
```

---

fc_guard_act(number_list: direction, number_list: ref_ft, string: tool, string: wobj, number: type, number: force_property)

函数说明：

该指令控制机械臂在末端力控过程中进行力安全力监控。当监控力拆过所设定范围时，机器人会触发急停并报警。

参数说明：

direction:: 6 个笛卡尔空间方向末端力安全监控开关，开为 true，关为 false。

ref_ft: 6 个笛卡尔空间方向末端力安全监控最大力，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm。该值未监控最大力范围绝对值，不区分所在自由度上的 ± 方向。

tool ： 设置使用的末端力安全监控工具的名称，默认为当前使用的工具。若末端力控参考坐标系配置为工具坐标系，则安全力监控的笛卡尔方向的定义参考 tool 坐标系的笛卡尔方向。

wobj ：设置使用的末端力安全监控工件坐标系的名称，默认为当前使用的工件坐标系。若末端力控参考坐标系配置为工件坐标系，则安全力监控的笛卡尔方向的定义参考 wobj 坐标系的笛卡尔方向。

type：末端力安全监控参考坐标系选择标志位，0 为参考工具坐标系，1 位参考工件坐标系。

force_property ：监控力属性，0 为末端负载力及外力，1 为末端外力（不含负载），可缺省，默认为 0。

返回值：

无

示例：

```
fc_guard_act({true,false,false,false,false,false},{1,0,0,0,0,0},"","",
0, 0)
```

---

fc_force_set_value(number_list: direction, number_list: ref_ft)

函数说明：

该指令控制机械臂末端力传感器读数设置为指定值。

参数说明：

direction：6 个末端力传感器输出力设置标志位，需要设置为 true，不需要设置为 false。需要注意的是，该方向定义未传感器自身力坐标系方向。

ref_ft：6 个末端力传感器输出力设置目标值，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm。

返回值：

无

示例：

```
fc_force_set_value({true,false,false,false,false,false},{1,0,0,0,0,0})
```

---

fc_wait_pos(number_list: middle_value, number_list: range, boolean: absolute, number: duration, number: timeout)

函数说明：

该指令控制机械臂在执行 fc_start() 函数后的末端力控过程中满足指定位置判断条件时自动停止当前运动函数并跳过后续运动函数，直到 fc_stop() 函数被执行停止末端力控或再次调用 fc_wait_logic 函数复位力控条件判断。

参数说明：

middle_value：位置判断条件绝对值，X/Y/Z 方向单位 m，RX/RY/RZ 方向单位 (rad)。

range：位置判断条件范围大小，X/Y/Z 方向单位 m，RX/RY/RZ 方向单位 (rad)。由于位置信息存在波动与偏差，适当设定范围大小可以有效保证逻辑触发成功率。

absolute: 绝对/增量条件判断标志位，true 为绝对位置判断，false 为增量位置判断。增量判断会以最后一次调用 fc_wait_logic 函数复位力控条件判断时机器人所在位置为参考点。

duration: 条件满足触发保持时间，单位 ms。

timeout: 条件满足触发超时时间，单位 ms。起始计算时间未最后一次调用 fc_wait_logic 函数复位力控条件判断时刻。

返回值:

无

示例:

```
fc_wait_pos ({0.005,0.005,0,0,0,0},{0.001,0.001,0,0,0,0},false,
1000,5000)
```

---

fc_wait_vel(number_list: middle_value, number_list: range, boolean: absolute, number: duration, number: timeout)

函数说明:

该指令控制机械臂在执行 fc_start() 函数后的末端力控过程中满足指定速度判断条件时自动停止当前运动函数并跳过后续运动函数，直到 fc_stop() 函数被执行停止末端力控或再次调用 fc_wait_logic 函数复位力控条件判断。

参数说明:

middle_value: 速度判断条件绝对值，X/Y/Z 方向速度范围 [-5, 5]，单位 (m/s)，RX/RY/RZ 方向速度范围 [-2*PI, 2*PI]，单位 (rad/s)。

range: 速度判断条件偏移范围大小，X/Y/Z 方向单位 (m/s)，RX/RY/RZ 方向单位 (rad/s)。由于速度信息存在波动与偏差，适当设定范围大小可以有效保证逻辑触发成功率。

absolute: 绝对/增量条件判断标志位，true 为绝对速度判断，false 为增量速度判断。增量判断会以最后一次调用 fc_wait_logic 函数复位力控条件判断时机器人末端速度为参考基准。

duration: 条件满足触发保持时间，单位 ms。

timeout: 条件满足触发超时时间，单位 ms。起始计算时间未最后一次调用 fc_wait_logic 函数复位力控条件判断时刻。

返回值:

无

示例:

```
fc_wait_vel ({0.01,0.01,0,0,0,0},{0.001,0.001,0,0,0,0},false,
0,5000)
```

---

fc_wait_ft(number_list: middle_value, number_list: range, boolean: absolute, number: duration, number: timeout)

函数说明:

该指令控制机械臂在执行 fc_start() 函数后的末端力控过程中满足指定力判断条件时自动停止当前运动函数并跳过后续运动函数，直到 fc_stop() 函数被执行停止末端力控或再次调用 fc_wait_logic 函数复位力控条件判断。

参数说明：

middle_value：力判断条件绝对值，范围 [-1000, 1000]，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm。

range：力判断条件偏移范围大小，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm。。由于力信息存在波动与偏差，适当设定范围大小可以有效保证逻辑触发成功率。

absolute：绝对/增量条件判断标志位，true 为绝对力判断，false 为增量力判断。增量判断会以最后一次调用 fc_wait_logic 函数复位力控条件判断时机器人末端感知力为参考基准。

duration：条件满足触发保持时间，单位 ms。

timeout：条件满足触发超时时间，单位 ms。起始计算时间未最后一次调用 fc_wait_logic 函数复位力控条件判断时刻。

返回值：

无

示例：

```
fc_wait_ft ({1,1,0,0,0,0},{0.1,0.1,0,0,0,0},false,0,5000)
```

---

fc_wait_logic(number : pos, number : vel, number : force)

函数说明：

该指令控制机械臂在执行 fc_start() 函数后的末端力控过程中位置条件判断、速度条件判断与力条件判断间的逻辑关系。不配置时默认三个条件判断都禁用。当已通过

参数说明：

pos ： 位置条件判断， 0 代表不启用，1 代表与逻辑，2 代表或逻辑；

vel ： 速度条件判断， 0 代表不启用，1 代表与逻辑，2 代表或逻辑；

force ： 力条件判断， 0 代表不启用，1 代表与逻辑，2 代表或逻辑；

例如开启位置条件判断，禁用速度条件判断，开启力条件判断，并且位置与力的关系为或，则输入 {1,0,2}。

返回值：

无

示例：

```
fc_wait_logic (1,0,2)
```

---

fc_get_ft()

函数说明:

该指令用以获取当前机器人末端传感器的原始反馈读数。该读数经过去皮操作，当机器人上下使能或调用 fc_force_set_value 时，会根据机器人系统中设置的负载参数（默认负载安装在传感器力检测端）以及目标参考传感器输出度数的大小进行去皮。

参数说明:

无

返回值:

6 自由度末端力读数，X/Y/Z 方向单位 N，RX/RY/RZ 方向单位 Nm

示例:

```
ret = fc_get_ft ()
```

---

fc_mode_is_active()

函数说明:

该指令用以获取当前机器人末端力控功能启用状态。需要注意的是，即使通过该函数获取到当前机器人末端处于力控状态下，也需要机器人当前存在正在执行的绝对运动函数，例如 movel、movec、fc_move 等，才会产生对应的末端力控运动。

参数说明:

无

返回值:

机器人末端力控启用返回 true，未启用返回 false。

示例:

```
ret = fc_mode_is_active ()
```

4.5.11 运动优化函数

enable_speed_optimization()

函数说明:

该指令控制机械臂开启速度优化功能。开启该功能后，机器人会参考当前模式下的安全限制参数，以不会违反安全限制参数为基础实时优化机器人速度控制指令，以达到最优速度节拍。速度优化功能仅在机器人自动模式下有效。当机器人开启速度优化功能并且处于自动模式下，原始机器人程序中所设定的关节最大速度与末端最大速度参数不再生效。

参数说明:

无

返回值:

无

示例：

```
enable_speed_optimization ()
```

---

disable_speed_optimization()

函数说明：

该指令用以控制机械臂退出速度优化。

参数说明：

无

返回值：

无

示例：

```
disable_speed_optimization ()
```

---

enable_acc_optimization()

函数说明：

该指令控制机械臂开启加速度优化功能。开启该功能后，系统会根据机器人动力学模型、机械功率模型及电功率模型计算机器人起停最优加速度，在满足速度约束前提下，机械臂以尽可能高的加速度进行规划。当速度优化同时打开后，系统默认同时开启加速度优化功能，该函数不在生效。

参数说明：

无

返回值：

无

示例：

```
enable_acc_optimization ()
```

---

disable_acc_optimization()

函数说明：

该指令用以控制机械臂退出加速度优化。

参数说明：

无

返回值：

无

示例：

```
disable_acc_optimization ()
```

---

enable_vibration_control()

函数说明:

该指令用以开启对机械臂起停振动控制功能进行优化。该功能在机器人系统启动时默认开启，以最大程度的保证机器人运动起停稳定性。需要注意的是，该功能开启后，机器人每一次独立运动都会有短暂的节拍降低（约 200-400ms）。

参数说明:

无

返回值:

无

示例:

```
enable_vibration_control ()
```

---

disable_vibration_control()

函数说明:

该指令用以退出对机械臂末端振动的优化。若在机器人实际调试过程中受限于连续独立短轨迹运行节拍，且对机器人末端起停稳定性要求较低，可以通过使用该函数临时禁用振动控制功能。需要注意的是，调用该函数后，机器人在程序运行结束后并不会自动复位并开启振动控制功能，因此若需要依旧保持手动试教过程中机器人运动起停稳定性，应再次调用 enable_vibration_control 函数开启振动控制功能。

参数说明:

无

返回值:

无

示例:

```
disable_vibration_control ()
```

---

enable_singularity_control()

函数说明:

该指令用以开启机械臂奇异点规避功能。开启奇异规避功能后，若机器人后续运动过程为笛卡尔空间运动且运动过程中会穿过机器人预先定义好的奇异空间，则会自动切换到关节空间过度运动，从而避免机器人经过奇异空间过程中引发的失速问题。在奇异规避过程中，机器人末端仅能尽可能保证末端与原始路径的一致性，实际会产生一定程度的精度损失。

参数说明:

无

返回值:

无

示例：

```
enable_singularity_control ()
```

---

disable_singularity_control()

函数说明：

该指令用以关闭机械臂奇异点规避功能。关闭该功能时，若机器人在笛卡尔空间运动过程中经过奇异区域，则会有失速风险，且机器人在接近奇异区域后会触发停机报警。

参数说明：

无

返回值：

无

示例：

```
disable_singularity_control ()
```

---

enable_posture_control()

函数说明：

该指令用以开启融合段姿态约束功能。开启该功能后，机器人在笛卡尔运动过程中，会以起点机器人末端工具坐标系与目标轨迹路径起点处切线方向之间的关系为参考，在运动过程中始终保持机器人末端工具坐标系与运动过程中轨迹路径切线方向的关系。

参数说明：

无

返回值：

无

示例：

```
enable_posture_control ()
```

---

disable_posture_control()

函数说明：

该指令用以关闭融合段姿态约束功能。

参数说明：

无

返回值：

无

示例：

```
disable_posture_control ()
```

enable_qnear_conf()

函数说明:

该指令用以启用机器人运动选解空间校验功能。启用机器人运动选解空间校验功能，若运动函数存在机器人参考关节角指令，如 movej_pose、movej_pose2、movel、movec 等，在轨迹开始运行时，运动规划会优先对当前机器人所处运动学选解空间与参考关节角指令所表征运动学选解空间做校验。若校验结果不一致，机器人系统则会判定其存在机器人产生实际运动与试教参考点位存在运动学选解不一致的风险，即虽然最终机器人到达目标机器人末端位置空间一致，但是机器人关节角不一致，从而触发安全报警并停机。

参数说明:

无

返回值:

无

示例:

```
enable_qnear_conf ()
```

disable_qnear_conf()

函数说明:

该指令用以禁用机器人运动选解空间校验功能。禁用机器人运动选解空间校验功能后，机器人最终笛卡尔运动所到达的目标机器人关节位置以机器人其实运动所在关节位置为参考，可能存在最终实际运动到位后与期望目标关节位置选解不一致的风险。

参数说明:

无

返回值:

无

示例:

```
enable_qnear_conf ()
```

4.5.12 复合运动函数

combine_motion_config(number:type, number:ref_plane, number:fq, number:amp, number:el_offset, number:az_offset, number:up_height, numble_list:time, boolean: path_dwell)

函数说明:

该指令控制机器人在原始运动轨迹上额外叠加一个复合运动，被叠加的符合运动为一周期性运动轨迹，且可以被几何描述。

参数说明:

type ： 复合运动轨迹几何类型。1：平面三角形轨迹，2：平面正弦轨迹，3：平面圆形轨迹，4：平面梯形轨迹，5：平面 8 字形轨迹

ref_plane ： 参考平面，0：工具 XOY 平面，1：工具 XOZ 平面，2：工具 YOZ 平面，3：工件 XOY 平面，4：工件 XOZ 平面，5：工件 YOZ 平面。

fq ： 频率，单位（Hz）。

amp ： 振幅，单位（m）。

el_offset ： 仰角偏移，单位（m）。（参数预留）

az_offset：方向角偏移，单位（m）。（参数预留）

up_height：中心隆起高度，单位（m）。（参数预留）

time ： 左右停留时间，仅针对平面梯形轨迹以及平面正弦轨迹有效，单位 ms。

path_dwell ： 主路经同步停留。为 true 时，机器人摆动到两端时，将完全停止。默认为 false

返回值：

无

示例：

```
combine_motion_config (1,0,1,0.1,0,0,0,{0,0})
```

---

enable_combined_motion ()

函数说明：

该指令会以最近一次使用 combine_motion_config 脚本设置的复合运动参数开启复合运动。

参数说明：

无

返回值：

无

示例：

```
enable_combined_motion ()
```

---

disable_combined_motion ()

函数说明：

该指令用以关闭复合运动。

参数说明：

无

返回值：

无

示例：

```
disable_combined_motion ()
```

备注： 当且仅当机器人停止时，可以对复合运动是否启用进行开关，无法在轨迹运动过程中动态开启或关闭。