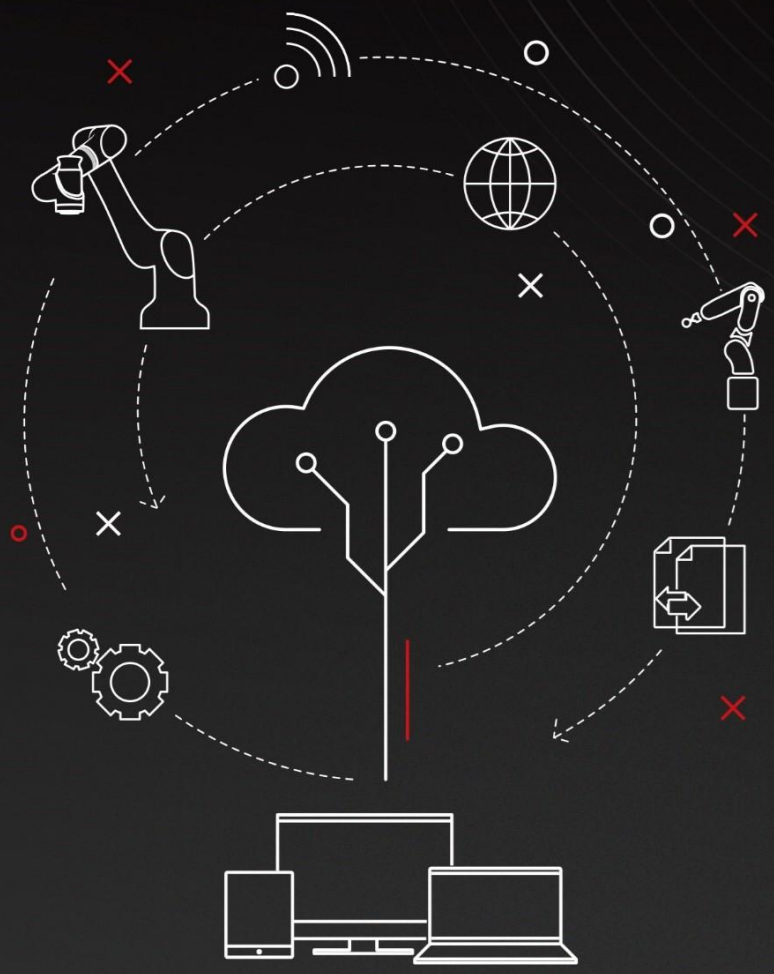
ROKAE 珞石

xCore

控制系统使用手册



人类生产生活的得力伙伴

xCore

控制系统使用手册

控制系统版本：V3.1.1

[备注]

文档版本：[状态]

本手册中记载的内容如有变更，恕不事先通告。本公司对手册中可能出现的错误均不承担任何责任。
本公司对因使用本手册及其中所述产品而引起的意外或间接伤害均不承担任何责任，敬请谅解。
本公司不可能预见所有的危险和后果，因此本手册不能警告用户所有可能的危险。
禁止擅自复印或转载本手册的部分或全部内容。
如您发现本手册的内容有误或需要改进抑或补充之处，请不吝指正。
本手册的原始语言为中文，所有其他语言版本均翻译自中文版本。

珞石（山东）机器人集团有限公司
中国.山东

目录

1 手册概述

1.1 关于本手册

感谢您购买本公司的机器人系统。

本手册记载了 xCore 控制系统的以下说明：

- 控制系统构成与基础操作
- 控制系统的编程与高级参数设置
- 控制系统选项功能介绍
- RL 指令集
- 控制系统错误码清单

安装使用该机器人系统前，请仔细阅读本手册与其他相关手册。

阅读之后，请妥善保管，以便随时取阅。

1.2 手册对象

本手册面向：

- 操作人员
- 产品技术人员
- 技术服务人员
- 机器人程序员

请务必保证以上人员具备控制系统操作所需的知识，并已接受本公司的相关培训。

1.3 如何阅读产品手册

本手册包含单独的安全章节，必须在阅读安全章节后，才能进行安装或维护作业。

1.4 本手册中的插图

由于产品升级或其他原因，产品手册中的一些图片可能会与实际产品存在差异，但操作步骤是正确的。

同时，对于某些通用的信息，可能会使用其他型号机器人的图片进行说明。

1.5 垂询方式

机器人维护、维修等相关事项，请与本公司售后部门或当地经销商联系。

联系时，请准备好如下信息：

- 控制器型号/序列号
- 机器人型号/序列号
- 软件名称/版本
- 系统出现的问题

1.6 手册阅读指南

本操作手册分为以下几个章节。

章节	标题	内容简介
1	手册概述	手册总体情况。
2	安全	安全相关事项。
3	名词术语	手册涉及的名词术语。
4	机器人基础知识	部分必备的机器人基础知识。
5	机器人系统构成与连接	机器人系统的构成，不同机型的物理连接。

6	HMI 简介	HMI 的整体布局和功能介绍。
7	控制系统基础操作	控制系统常用的基础操作；通过示例，演示工业、协作机器人最基础的操作。
8	编程	详细介绍编程模块的使用。
9	设置	详细介绍控制系统各类设置。
10	通信	详细介绍通信类的功能及使用。
11	安全	介绍安全相关的功能。
12	工艺包	概述工艺包情况。
13	日志	介绍日志相关功能及使用。
14	选项	介绍选项模块的功能及使用。
15	RL 指令	详细介绍所有 RL 指令。
16	附录	用户权限明细、协作机型末端把手功能和使用介绍。
17	故障排查	故障码，处理清单。

1.7 手册修订历史

版本号	日期	主要修改内容
V2.1	2023.11	新版手册创建；
V2.2	2024.03	2.2 新增功能说明
V3.0	2024.12	3.0 新增功能说明
V3.1	2025.07	3.1 新增功能说明

1.8 相关手册

xCore 控制系统除完善的基本功能外，还包含丰富的扩展功能。关于扩展功能，我们提供以下文档，如有需要，可以联系官方获取。

名称	简介
《xCore 机器人控制系统使用手册》	介绍 xCore 控制系统基本功能使用；
《xVision 使用手册》	介绍 xVision 机器人视觉基本使用；
《xCore_SDK_Android 使用手册》	介绍 xCore-SDK 的使用；
《xCore_SDK_Python 使用手册》	介绍 xCore-SDK 的使用；
《xCore_SDK_C++使用手册》	介绍 xCore-SDK 的使用；
《xCore_RCI_使用手册》	介绍 RCI 的使用；
《传送带跟踪功能使用手册》	介绍传送带跟踪功能的使用；
《料盘工艺包使用手册》	介绍料盘工艺包使用；
《码垛工艺包使用手册》	介绍码垛工艺包使用；
《RokaeStudio 用户手册》	介绍离线编程软件使用；
《光伏排版工艺包用户手册》	介绍光伏排版工艺包使用；
《光伏插片工艺包用户手册》	介绍光伏插片工艺包使用；
《Profinet 使用手册》	介绍 profinet 总线的使用；
《EthernetIP 使用手册》	介绍 EthernetIP 总线的使用；
《外部轴导轨使用手册》	介绍外部轴导轨的使用；

2 安全

2.1 简介

本章介绍在使用机器人时需要注意的安全原则和流程。
与机器人外部安全防护装置的设计、安装有关的内容不在本章描述范围之内，请与您的系统集成商联系以获得此类信息。

2.2 安全责任说明

珞石机器人致力于提供可靠的安全信息，但不对此承担责任。即使一切操作都按照安全操作说明进行，也不能确保工业机器人不会造成人身和财产方面的损失。

2.3 安全标识

按照本手册内容操作机器人时可能会遇到不同程度的危险状况，因此在可能会造成危险的操作说明附近会有专门的安全标识提示框重点提示用户注意防范，提示框中的内容包括：
一个表示安全级别的图标和对应的名称，例如警告、危险、提示等；
一段简单的描述，用于说明如果操作人员不消除该危险可能会造成的后果；
有关如何消除危险的操作说明。

2.3.1 安全级别

图标	名称	说明
	危险	带有该标识的内容如果没有按照规定操作，将会对人员造成严重甚至致命的伤害，同时将会/可能会对机器人造成严重损坏。
	警告	带有该标识的内容如果没有按照规定操作，可能会导致严重人身伤害，甚至可能致命，对机器人本身也将造成较大损坏。
	触电危险	提示当前操作可能会有人员触电风险，造成严重甚至是致命的伤害。
	警示	带有该标识的内容如果没有按照规定操作，可能会导致人身伤害，对机器人本身可能也会造成损坏。
	防静电（ESD）	提示当前操作涉及的零部件对静电敏感，不按规范操作可能会造成器件损坏。
	提示	用于提示一些重要信息或者前提条件。

图标	名称	说明
	挤压	操作人员、维护人员在调试、维修、检修、工具装夹时进入机器人运动范围，可能会产生伤害。
	夹手	维护人员在进行维护操作时，接近带传动部件时，存在夹手的风险。
	撞击	操作人员、维护人员在调试、维修、检修、工具装夹时进入机器人运动范围，可能会产生严重伤害。
	摩擦	操作人员、维护人员在调试、维修、检修、工具装夹时进入机器人运动范围，可能会产生伤害。

	零件飞出	操作人员、维护人员在调试、维修、检修、工具装夹时进入机器人运动范围，工具或工件可能因夹持松懈飞出，此时可能会产生严重伤害。
	火灾	电路发生短路、导线或器件着火时可能发生火灾，可能会产生严重伤害。
	高温表面	维护人员在进行设备检修、维护时，接触机器人高温表面，可能会导致烫伤危害。

2.3.2 风险说明

2.4 安全停止

机器人的停止方式有三种: STOP 0、STOP 1和 STOP 2。
安全停止是指在由安全控制器触发的停止，安全控制器只触发 STOP 0、STOP 1 两种停止方式，STOP 2 只由控制系统负责。

STOP 0	STOP 0 停止被触发后，立即切断电机的动力电源并闭合各关节抱闸，是安全级别最高的停止方式，但停止过程中机器人处于非受控状态，停止后可能会偏离编程路径。 手动模式的安全停止属于 STOP 0。 STOP0 支持按最大能力减速停止方式。
STOP 1	STOP1 停止被触发后，控制系统立刻沿编程路径执行减速过程，之后不论机器人是否完全停下，安全控制器将切断电机的动力电源并闭合各关节抱闸。在绝大多数情况下，由于是受控停止，机器人最终将停在编程路径上，因此该种急停方式对周边设备的保护性最好。 自动模式下安全门/安全光栅打开、自动模式下急停按钮被按下而发生的安全停止，均属于 STOP 1。 STOP1 支持按最大能力减速停止、正常规划停止两种方式。
STOP 2	STOP 2 停止被触发后， 控制系统立刻沿编程路径执行减速，直到机器人完全停止运动。此时电机的动力电源仍然保持，抱闸仍然打开，机器人保持在当前位置上。 通过 HMI 停止按钮和外部信号控制机器人停止是触发的 STOP2 停止方式

提示

1.紧急停止仅用于在危险情况下立刻停止机器人运行。不能将紧急停止作为正常的程序停止，否则将对机器人的抱闸系统和传动系统造成额外而不必要的磨损，降低机器人的使用寿命。

2.5 安全装置

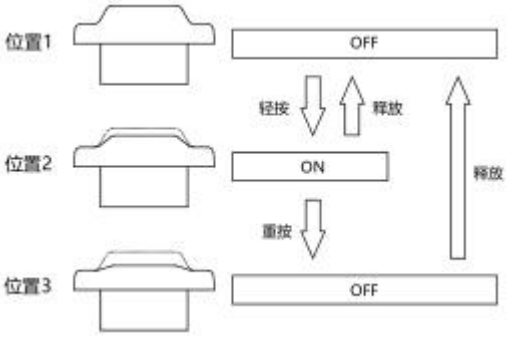
2.5.1 急停

急停按钮是手动触发紧急停止的按钮，多数使用红色的操作主体，最常见的外形是蘑菇头型，通常急停按钮还配合使用黄色的衬底、保护外壳或警示牌。
按下急停时，按钮靠机械锁定，这是急停按钮的安全锁机制，此时必须通过手动释放来复位装

置。大多数急停按钮都采用旋转释放方式，旋转方向会标在按钮的表面，也有一部分按钮支持直接向上拔起的释放方法。

2.5.2 使能装置

使能装置是一个具有 2 段按压 3 个位置的特殊开关，又称三位使能开关（以下简称使能开关），用于在手动模式下控制机器人动力电源的通断，实现机器人的运动使能。只有按下使能开关并保持在中间位置时才会接通电机电源，使得机器人处于允许运动的状态。松手放开或者用力按压到底都会将电机电源切断。



提示

手动使能的黄色按钮为使能开关，当按压到中间位置时电机动力电源接通并自动使能，系统处于上电状态，可以进行点动或运行程序。松开或者按到底时电机动力电切断，系统回到下电状态。

为了安全的使用示教器，必须遵守以下要求：

1. 在任何情况下都必须保证使能开关可以正常工作；
2. 在编程和调试期间，当不需要机器人运动时应尽快松开使能开关；
3. 任何进入机器人工作空间的人员必须随身携带手持使能，以避免其他人在内部人员不知情的情况下启动机器人 。

警告

严禁使用外部装置将使能开关卡住使其停留在中间位置！

2.6 各种情况下的安全注意事项

2.6.1 手动模式的安全事项

在手动模式下，机器人的运动处于手动控制状态。只有在使能开关处于中间位置时，才能对机器人进行 Jog 或者运行程序。手动模式用于编写、调试机器人程序以及参与工作站试运行调试。

2.6.1.1 手动模式下的速度限制

在手动模式下，机器人末端的运动速度被限制在 250mm/s 以下，即无论是 Jog 机器人还是运行程序，不论程序中设置的速度是多少，机器人末端的最大运动速度不会超过 250mm/s。

2.6.1.2 旁路外部安全信号

在手动模式下，外部安全装置如安全门、安全光栅等信号将被旁路，即在手动模式下即使安全门被打开系统也可以进行电机使能的操作， 且不会有安全门打开的信息提示，以方便进行调试。

2.6.2 自动模式的安全事项

自动模式用于在正式生产过程中机器人程序的运行。自动模式下使能开关将被旁路，因此机器人可以在没有人员参与的情况下自动运行。

2.6.2.1 启动外部安全信号

外部安全装置如安全门、安全光栅等在自动模式下会启用， 安全门打开会使电机断开电源且闭
合抱闸。

2.6.3 安装、操作的安全要求

- 在搬运、 安装机器人设备时， 需按照本公司手册说明的方法进行， 否则有可能由于错误操作导致机器人翻倒，进而导致作业人员伤亡或设备损坏。
- 机器人设备安装好后首次使用时，务必先以低速进行，然后逐渐加快速度，不可首次就使用高速运行。
- 程序和系统变量等信息默认保存在控制柜存储设备中，为了预防由于意外引起的数据丢失，建议用户定期进行数据备份。

2.6.4 调试的安全要求

调试时尽可能在安全防护区域外进行， 当必须在安全防护区域内进行调试时，应着重注意下列事项：

- 仔细查看安全防护区域内的情况，确认没有危险再进入安全防护区域。
- 应确认安全防护区域内的所有调试人员的位置。
- 应在确认整个系统的状态后进行作业。
- 要做到随时都可以按下急停按钮。
- 应以低速运行机器人。
- 调试结束后，调试人员务必在安全防护区域外进行操作。

2.6.5 维护的安全要求

- 仔细查看安全防护区域内的情况，确认没有危险再进入安全防护区域应确认安全防护区域内的所有维护人员的位置。
- 当接通电源时， 部分维护作业有触电的危险，应尽可能先断开机器人设备及系统电源，再进行维护作业。
- 维护作业时应避免其他人员无意中接通电源。
- 在进行作业时，不要将身体任何部位搭放在机器人设备的任何部分， 以免造成不必要的人身伤害或对设备造成不良影响。
- 进行维护作业时，应配备适当的照明器具。
- 如需更换部件，务必使用本公司指定部件。若使用指定部件以外的部件，有可能导致机器人设备的损坏。
- 在更换部件时拆下来的零件（如螺钉），应正确装回其原来部位，如果发现零件不够或零件有剩余，则应再次确认并正确安装。

2.6.6 生产线上的安全处理

绝大多数情况下，机器人属于生产线的一部分，因此机器人出现故障往往不只影响机器人本身，而会影响整个生产线，同样，生产线的其他部分出现问题时，也可能会影响到机器人。因此应由对整个生产线非常熟悉的人员来设计故障补救方案，以提高整个系统的安全性。

- 需关注与机器人进行交互的其他设备

例如，当某机器人需要维护时，需将此机器人从生产线上先脱离出来，也必须同时脱离与其交互的其他设备，例如为其上料的机器人。

- 需关注机器人周围仍保持运行的其他设备

例如，生产线上的机器人需要从传送带上抓取工件，当机器人出现故障时，为保证生产过程不中断，在检修机器人的同时，传送带可能仍然保持运行，此时机器人维修人员应额外注意安全，需提前考虑运行中的传送带可能带来的风险， 并制定详细的在此环境中工作的安全措施。

2.6.7 火灾时的安全

在即将发生火灾危险或火灾已经发生但尚未蔓延开来的情况下，不要惊慌，保持镇定，使用现场提供的灭火装置将火焰扑灭。严禁用水扑灭因短路导致的火灾。

> ⚠ 警告
>
> 机器人工作现场使用的灭火装置需由用户提供，用户需根据现场实际情况，选择合适的灭火装置。

当火灾已蔓延开来、处于不可控阶段时，现场工作人员不要再试图灭火，而应立即通知其他工作人员，放弃私人物品，尽快从紧急出口向外撤离，撤离时禁止使用电梯，撤离过程中同时呼叫消防队。若有人员衣物着火，不要让他/她跑动，而应让他/她迅速平躺在地上，用衣服或其它合适物品、方式将火扑灭。

2.6.8 触电时的安全

当发现有人触电，不要惊慌，首先要尽快切断电源。

应根据现场具体条件，果断采取适当的方法和措施，一般有以下几种方法和措施：

1. 如果电源开关或按钮距离触电地点很近，应迅速拉开开关，切断电源。
2. 如果电源开关或按钮距离触电地点很远，可用绝缘手钳或用干燥木柄的斧、刀、铁锹等切断电源侧（即来电侧）的电线，切断的电线不可触及人体。
3. 当导线搭在触电人身上或压在身下时，可用干燥的木棒、木板、竹杆或其它带有绝缘柄（手握绝缘柄）的工具，迅速将电线挑开，不能使用任何金属棒或湿的东西去挑电线，以免救护人触电。

触电伤员脱离电源后的处理：

1. 如果触电伤员神志清醒，应使其就地仰面躺开，严密监视，暂时不要站立或走动。
2. 如果触电伤员神志不清，应使其就地仰面躺开，确保气道通畅，并用 5 秒的时间间隔呼叫伤员或轻拍其肩部，以判断伤员是否意识丧失。禁止摆动伤员头部呼叫伤员。就地抢救的同时尽快联系医院。
3. 如果触电伤员意识丧失，应在 10 秒内判断伤员呼吸、心跳情况。若即无呼吸又无动脉搏动，可判定呼吸心跳已停止，应立即用心肺复苏法对其进行抢救。

3 名词术语

本章简单介绍手册中出现的部分名词术语。

名词	解释
RobotAssist	珞石机器人 xCore 控制系统的上位机软件，具有机器人移动控制、程序编写、参数配置、状态监控等功能，可以运行于 xPad2 示教器、PC 等设备上。
HMI	Human Machine Interface，人机交互界面。
HMID	HMI Device，人机交互设备。
RC	Robot Controller，机器人控制器。
RCI	Rokae Control Interface，珞石机器人外部控制接口，支持底层实时控制。
SDK	Software Development Kit，软件开发工具包，支持通过 C++等语言实现对机器人的底层控制，后续将逐步取代 RCI。
Project	工程，控制机器人运行的程序、任务等对象的管理集合；工程的数据对象可以导出，在其他工程或机器人复用。
Task	在 xCore 中我们称为任务。
Module	在 xCore 中我们称为程序模块。
Elbow	臂角，指的是臂平面与参考平面之间的夹角，臂平面指的是机器人大臂与小臂所组成的平面，参考平面是指，三轴为零时末端达到同样位姿时所形成的臂平面。
RL	Rokae Robot Language，珞石机器人语言，包含丰富的指令，可以基于 RL 搭建机器人任务。
xPad2	示教器。
RSC	Robot Safety Controller，安全控制器。
JOG	点动。
零空间运动	对于冗余自由度机器人，可以通过零空间运动，实现机器人关节运动但末端静止的效果。
PERS 变量	持续性变量，在程序执行过程中如果此类型变量的值发生了变化，将会自动将该变量的初始值修改为当前值，由此达到“持续”存储的效果。

4 机器人基础知识

4.1 本章简介

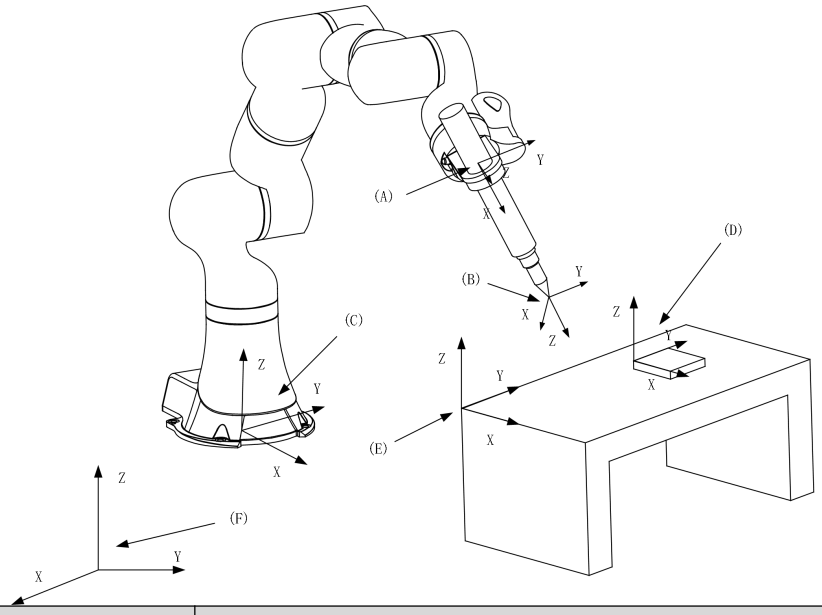
本章介绍机器人学相关的基础知识，熟悉本章内容有助于更好地理解和掌握控制系统及机器人的使用。

4.2 坐标系

空间中的任一物体(工具、工件等)都有六个自由度：三个平动自由度+三个旋转自由度。三个平动自由度构成位置；三个旋转自由度构成姿态；这六个自由度合起来统称位姿。物体的位姿，可以用附着其上的坐标系来描述，一般采用笛卡尔坐标系(以下简称坐标系)。

机器人是一种多自由度机构，典型的作业方式，是在法兰固定一个工具，执行相对外部工件的运动。这种作业方式，可以用坐标系及其相对运动来描述。

xCore 系统中包含的坐标系参见下图：



编号	坐标系	意义
A	法兰坐标系	定义在机器人法兰盘中心位置； 法兰坐标系相对基坐标系定义；
B	工具坐标系	定义在工具末端； 当工具为手持(普通工具)时，工具坐标系相对法兰坐标系定义； 当工具为外部工具时，工具坐标系相对用户坐标系定义；
C	基坐标系	定义在机器人基座的中心位置； 基坐标系相对世界坐标系定义； 注意：如果机器人不是默认安装方式，例如倒装、斜装，需要先进行基坐标系标定；
D	工件坐标系	定义在工件上的坐标系； 当工件为外部工件(普通工件)时，工件坐标系相对用户坐标系定义；当工件为手持工件时，工件坐标系相对法兰坐标系定义；
E	用户坐标系	在定义工件坐标系时充当参考系，不单独使用； 用户坐标系相对世界坐标系定义；
F	世界坐标系	世界坐标系一般作为基准坐标系，该坐标系并没有具体的位置； 单机器人正常安装时，默认与机器人基坐标系重合；当涉及多机

		器人或外部设备协同时，可以通过将他们的世界坐标系统一到同一个坐标系，实现运动基准的统一；

4.3 奇异

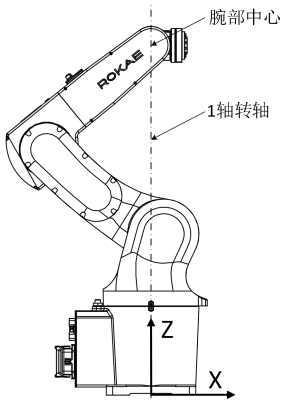
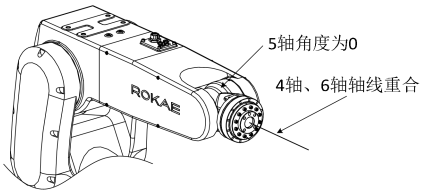
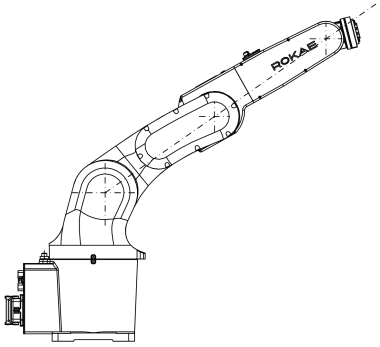
机器人的工作空间中存在若干特殊的位姿，机器人可以使用无数种不同的关节配置到达，这些位姿被称为奇异点。奇异点会导致控制系统在基于笛卡尔空间位姿计算关节角度时出现问题。机器人执行关节运动时，不存在奇异性问题。

当机器人执行靠近奇异点的笛卡尔空间轨迹时，某些关节的速度可能会非常快，可能导致报错，机器人运行中止。

4.3.1 典型机器人奇异位置

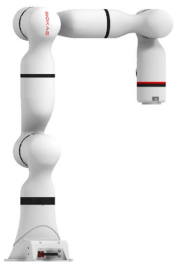
不同构型的机器人，有不同的奇异位置。下面介绍部分机器人构型的典型奇异位置。

4.3.1.1 工业六轴机器人奇异位置

奇异	构型	说明
肩部奇异		机器人腕心位于 1 轴轴线上时。
腕部奇异		当 4 轴和 6 轴轴线重合(5 轴角度为 0)时。
肘部奇异		当腕心、2 轴转轴、3 轴转轴位于一条直线上时。

4.3.1.2 ER PRO 协作机器人奇异位置

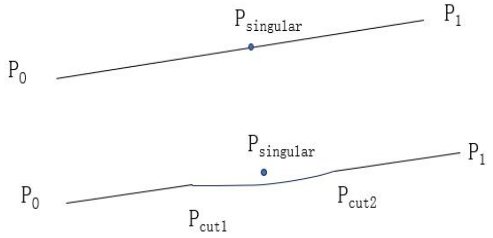
ER RPO 协作机器人的奇异可以分为如下情况：

奇异	构型	说明
2 轴奇异		当 2 轴角度等于 0°时，机器人求解逆解时无法区分 1 轴与 3 轴角度。
4 轴奇异		当 4 轴角度为 0，机器人沿平行于 3 轴或 5 轴方向的运动受到限制。这种奇异状态会使机器人的手腕根部丧失一个运动自由度（手腕根部无法沿手臂轴向的运动）。此时求解逆解时，3 轴和 5 轴的位置无法求解。
6 轴奇异		当机器人 6 轴角度等于 0°时，机器人求解逆解无法区分 5 轴与 7 轴角度。
腕心奇异		当腕心位于 1 轴正上方时，机器人求解逆解无法准确给定 1 轴角度。

4.3.2 奇异规避

奇异问题源自机器人构型，无法完全避免。在实际任务编程中，如遇到机器人必需通过奇异点附近的情况，可以考虑通过降低部分约束(例如姿态或路径精度等)，让机器人顺利穿越奇异点。xCore 控制系统也提供多种奇异规避方法：

方法一	4 轴锁定	开启该奇异规避方法前，需将机器人 4 轴运动到 0 度或者±180 度。开启该形式奇异后，机器人将保持 4 轴不动，在保证 TCP 位置准确的前提下，以一种特殊的形式进行姿态插补。可参考 RL 指令 SingAreaLockAxis4 on/off 和 jog 模式章节。

方法二	笛卡尔牺牲姿态插补	开启该奇异规避方法后，可通过改变机器人形态，以通过腕关节类型奇异。注意：使用此功能，机器人运动时的手腕形态与示教的有可能不同（不仅是通过奇异点的示教点，后续示教点的形态也有可能不同）。
方法三	关节空间插补	开启该奇异规避方法后，控制系统会对后续的笛卡尔轨迹进行奇异性检测，直到关闭该功能。不含奇异点的笛卡尔轨迹，按照普通轨迹进行运动。含有奇异点的笛卡尔轨迹，控制系统检测到奇异点后，围绕奇异点，将轨迹 P_0P_1 切割为 P0Pcut1, Pcut1Pcut2, Pcut2P1 三段，其中 P0Pcut1 和 Pcut2P1 依然沿着原轨迹运动进行笛卡尔插补，Pcut1Pcut2 则采用关节空间插补。三段轨迹间采用转弯区平滑过渡，如下图所示。  奇异规避:关节空间轨迹插补示意图 具体使用，参考 RL 指令 SingAreaJointWay。

注意：

- 机器人在奇异点附近，关节的运动幅度往往很大，请确认是否必须使用奇异规避指令，优先推荐通过改变轨迹点位来规避奇异点。
- 使用奇异规避指令时，在机器人正式作业前，请先确认加上奇异规避指令后机器人的轨迹符合作业需求，再进行正式作业。
- 机器人在奇异点附近，关节运动幅度往往很大，使用前请确认周围环境。
- 鉴于上述原因，如果机器人运行点位或程序运行逻辑受外部信号影响，使用前，请仔细确认程序逻辑和轨迹。

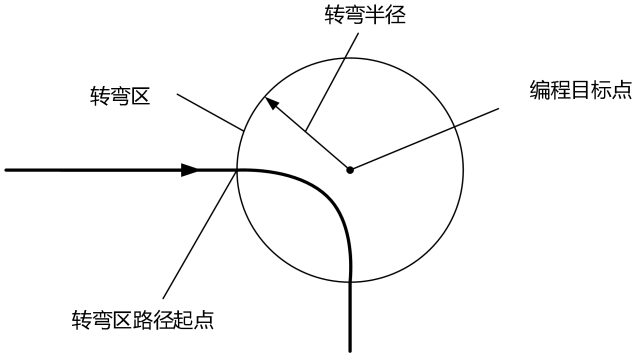
上述三种奇异规避具体特征如下：

方式与特点	4 轴锁定	笛卡尔牺牲姿态插补	关节空间插补
运动特征	1.开启该指令前，需先将机器人 4 轴运动到 0 度或者±180 度。 2.开启该指令后，机器人在接下来的运动指令中保持 4 轴不动。	1.开启该指令后，机器人改变手腕形态，工具姿态发生变化，以便通过有腕关节奇异点的轨迹。 2.运动时的手腕形态与示教的有时不同，不仅通过奇异点的示教点，而且以后的示教点的形态也有可能变更。	1.轨迹中奇异点前后的一部分，采用关节插补(MoveAbsJ)的形式运动，剩余部分采用原笛卡尔轨迹的本身运动方式运动，两种方式间采用转弯区过渡。
轨迹形式是否改变	1.位置轨迹不变。 2.特殊姿态插补方式。	1.位置轨迹不变。 2.改变姿态插补方式。	1.位置轨迹改变。 2.姿态轨迹改变。
目标点可达性	机器人运动空间，部分可达。即 4 轴为 0 度，或者±180 度的目标点位可达。	机器人运动空间，部分可达。	机器人全运动空间可达。
转弯区特征	1.同类型轨迹间支持生成转弯区，即 SingAreaLockAxis4 on 和 SingAreaLockAxis4 off 之间的运动轨迹间支持生成转弯区。	1.同类型轨迹间支持生成转弯区，即 SingAreaWrist on 和 SingAreaWrist4 off 之间的运动轨迹间支持生成转弯区。 2.不同类型间，轴空间轨迹和奇异规避轨	奇异规避轨迹之间，奇异规避轨迹与普通轨迹间均可生成转弯区。

	2.不同类型间不支持生成转弯区，即 SingAreaLockAxis4 on/off 指令前后的运动轨迹间的不支持生成转弯区。	迹之间，可生成转弯区。笛卡尔空间路径和奇异规避轨迹，不可生成转弯区。	
前瞻特征	打断前瞻，即 SingAreaLockAxis4 on/off 为阻塞指令，即机器人运动到 SingAreaLockAxis4 on 前一条轨迹，控制系统才会继续解析奇异规避指令。同理，机器人运动 SingAreaLockAxis4 off 前一条运动指令，控制系统才会继续解析后续指令。	不打断前瞻。	不打断前瞻。
奇异规避强制性	强制，开启锁轴后，所有运动笛卡尔运动指令均采用锁轴所对应的特殊插补形式进行插补，直到关闭锁轴功能。	强制，开启牺牲姿态奇异规避后，所有运动笛卡尔运动指令均采用锁轴对应的特殊插补形式进行插补，直到关闭牺牲姿态奇异规避功能。	不强制，开启该奇异规避后，控制系统自动检测每一条笛卡尔运动轨迹中是否含有奇异点。仅包含奇异点的轨迹会采用特殊的形式插补，不包含奇异点的轨迹依然采用普通路径的运动形式进行插补。 注意：开启后会增加控制系统计算量，非必要不开启。
适用场景	1.法兰平行于基座或者沿基座 z 轴运动的轨迹。 2.机器人运动过程中允许保持 4 轴不动，例如码垛。	不在意路径的姿态精度，也不在意机器人以何种形式到达目标点，仅追求能够到达目标点笛卡尔位置。	不在意路径的位置精度和姿态精度，也不在意机器人以何种形式到达目标点，仅追求能够到达目标点。
支持机型	工业标准六轴系列(XB,NB 型号)，协作 xMate CR/SR	工业标准六轴系列(XB,NB 型号)，协作 xMate CR/SR	工业标准六轴系列(XB,NB 型号)
是否支持 Jog	支持	不支持	不支持

4.4 转弯区

机器人的运动，一般是依次执行用户编程设定的多条轨迹。通常这些轨迹间并非光滑连接，而是存在各种各样的“尖点”。这些“尖点”的存在，使得机器人在执行一条轨迹时，必须先在该轨迹末尾停止运动，才能开始下一条轨迹。为了使机器人能够在不同轨迹间连续运动，必须消除上述“尖点”，可以通过“转弯区”将不同轨迹平滑地连接起来。见下图：



转弯区类型包括笛卡尔空间转弯区和轴空间转弯区。关于转弯区的详细定义和具体参数，可以参考下文“RL 指令”-“zone”章节。

4.5 前瞻机制

前瞻（Lookahead）是指在机器人运动过程中，控制系统提前处理当前机器人正在执行指令之后的程序指令，引入前瞻机制有如下好处：

- 可以获得前方轨迹的速度、加速度要求以及机器人本身的限制条件信息，以便规划性能最优的控制策略；
- 根据编程的转弯区设置规划转弯区轨迹；
- 获取靠近软限位/边界、靠近奇异点等异常状态，以便提前进行处理；

关于前瞻机制更详细的介绍，参考下文“编程”-“关于 RL 程序”-“RL 程序调试”章节。

4.6 力控

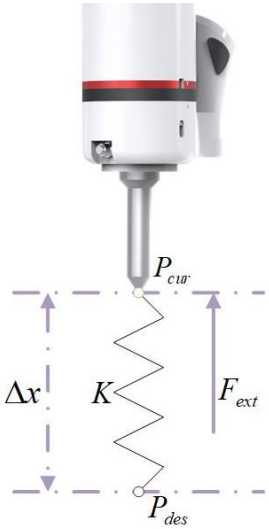
4.6.1 力控简介

机器人力控制是机器人末端与外部环境存在力交互的控制过程。在非接触类机器人运动控制过程中，只关注位置控制过程（速度与精度）；当与环境存在接触时，纯位置控制要求机器人和环境必须具有非常高的精度，避免位置偏差引起的接触力对机器人和环境造成损坏。

与单纯位置控制不同，机器人力控制在与环境交互过程中引入力/力矩反馈回路，并通过力/力矩反馈回路改变机器人的运动特性，从而起到与外部环境动态交互的作用。在机器人与外部环境存在偏差或不确定性时，力控会智能地调整预设的位置轨迹，消除位置偏差引起的内力，保证交互过程的平稳安全。

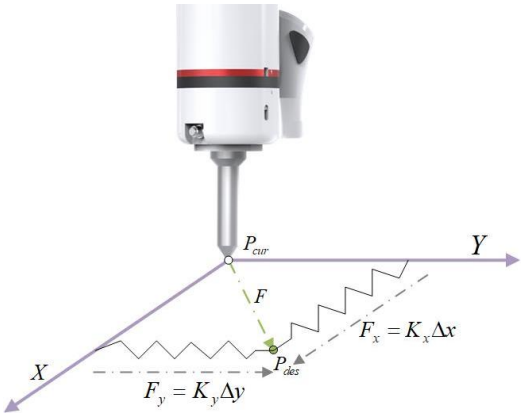
4.6.2 阻抗控制

相比传统工业机器人，xMate 协作机器人关节中配置了扭矩传感器，这使其能够精准的感知关节扭矩。关节的扭矩信息使得 xMate 协作机器人能够通过阻抗来实现力控制，阻抗控制使得机器人具有非常柔顺的交互行为。这意味着机器人与环境的交互相当于一个虚拟的弹簧刚度、阻尼系统。此时，机器人对外力具有敏感性，外力的作用能够使机器人偏离预先设定的轨迹。而外力的作用消失后，机器人能够具有一定的回弹效果。



阻抗运动过程中，在环境中的外力作用下，机器人的实际位置会偏离期望位置一定距离，偏离的距离取决于阻抗刚度和外力的大小，偏差的具体数值可以通过外力和阻抗刚度的比值得到。如上图，阻抗控制模式下，阻抗刚度设置为 K，外力 Fext 的作用下，机器人的当前位置 Pcur 会偏离期望位置 Pdes，位置偏差为Δx。此偏差引起的阻抗力和外力会达到最终的平衡。

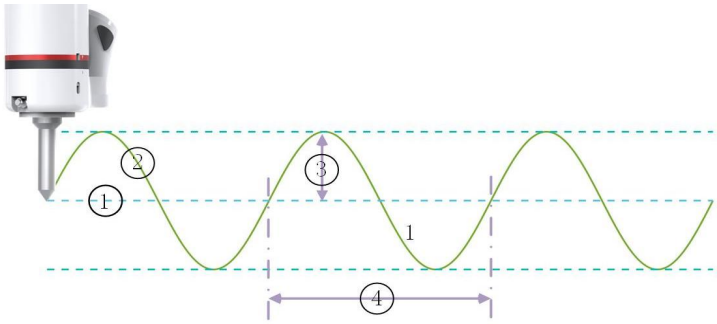
阻抗各方向的刚度可以单独设置，各方向阻抗力为此方向阻抗刚度和位置偏差的乘积，各方向阻抗力最终合成总的阻抗力。如下图，阻抗模式下，在外力的作用下机器人当前位置 Pcur 偏离期望位置 Pdes。X、Y 方向的偏差分别为Δx、Δy，阻抗刚度分别为 Kx、Ky，阻抗力分别为 Fx、Fy。总的阻抗力 F = Fx + Fy。



4.6.3 力控搜索

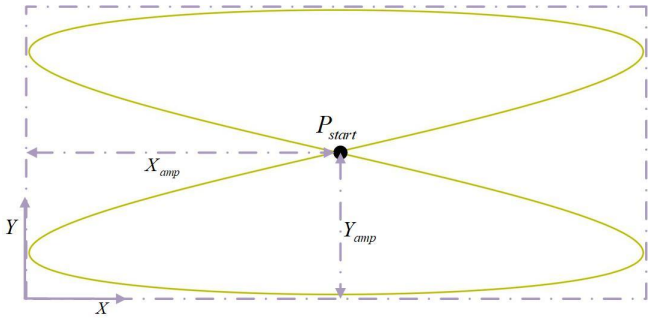
人手在工件装配的过程中，会感受装配过程中力的变化，当感受到阻碍的时候（工件卡住），会尝试通过抖动，来使工件顺利安装。力控制使得机器人也具有这样的手法，也即搜索运动，支持绕轴旋转的正弦搜索运动和平面内的莉萨如搜素运动。搜索运动是在机器人既定运动的基础上叠加的额外运动，搜索运动使得机器人表现出一定的抖动，从而能够在装配过程中更好的克服阻碍。

下图是一个正弦搜索运动过程：



1	期望轨迹	2	实际轨迹（期望轨迹+搜索运动）
3	搜索运动幅值	4	搜索运动周期

所谓莉萨如搜索运动，指的是平面内两个相垂直的方向分别施加一个正弦搜索运动，两方向正弦搜索运动的频率往往成一定的比例。例如下图所示为 XY 平面类的莉萨如搜索运动，其中 X 和 Y 方向搜索运动频率的比值为 2:1，中心点 Pstart 为期望位姿点，Xamp 为 X 方向搜索运动的幅值，Yamp 为 Y 方向搜索运动的幅值。

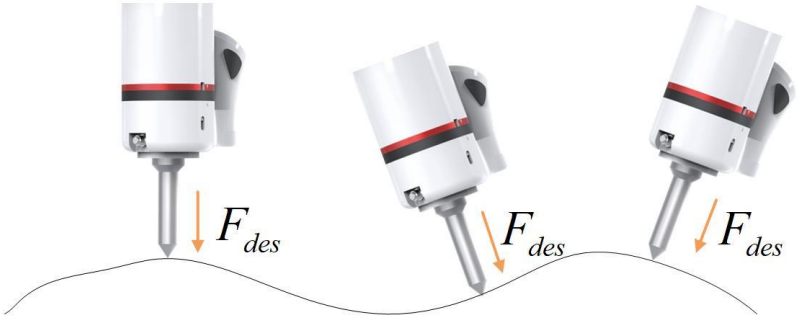


4.6.4 力控应用

工业机器人的力控应用场景大致可以分为两类：恒力跟踪和力控装配。

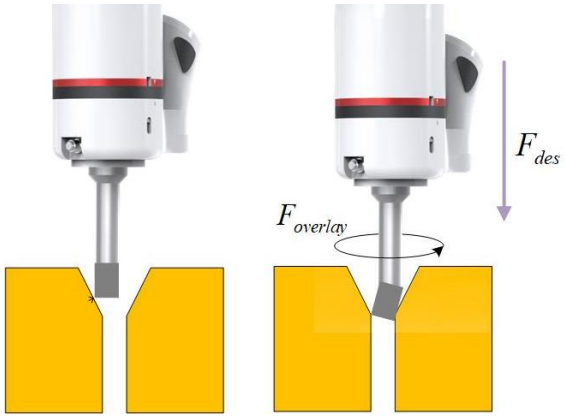
4.6.4.1 恒力跟踪

如下图，在恒力跟踪应用场景中，机器人保证与曲面之间的接触力保持一个恒定值 Fdes，同时机器人能够顺应曲面起伏的变化。恒力跟踪主要应用在打磨、去毛刺等应用场景。



4.6.4.2 力控装配

机器人装配工艺中，如果使用纯位置控制，由于位置、建模误差，机器人很容易就会碰撞到工件上，从而造成工件或机器人的损伤。而在基于力控制的装配工艺中，机器人在感受到外力超过一定范围（工件卡住），会尝试施加额外的搜索运动（抖动）来克服阻碍，从而使工件得以顺利安装。如下图，左边纯位置控制在装配过程中会发生碰撞，右边力控制在装配过程中，通过期望力 Fdes 将机器人推入装配孔，通过搜索运动 Foverlay 克服安装过程工件卡死的情况。



5 机器人系统构成及连接

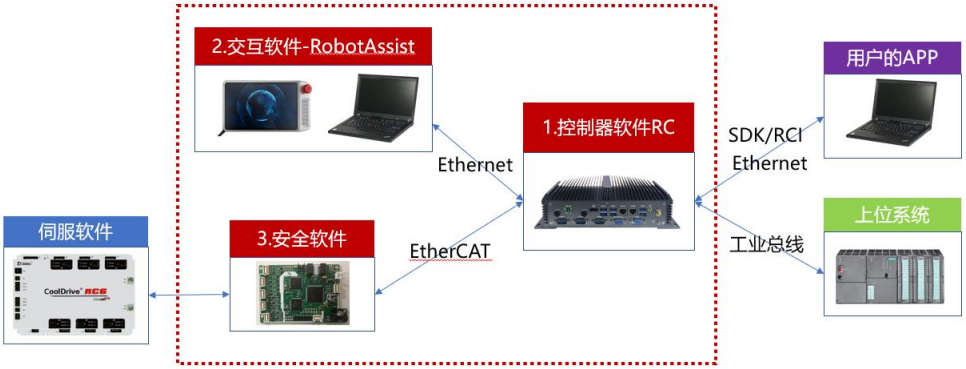
5.1 本章简介

珞石机器人分为多个系列，本章主要介绍不同系列机器人的系统构成与连接方式，加深用户对机器人系统的了解。

用户可以根据自己使用的机型，选择性阅读本章的内容。

5.2 控制系统构成

xCore 控制系统采用的是 CS 架构，主要包括 HMI 软件(RobotAssist)和控制器软件 RC。



控制系统构成

5.2.1 xPad2 示教器简介

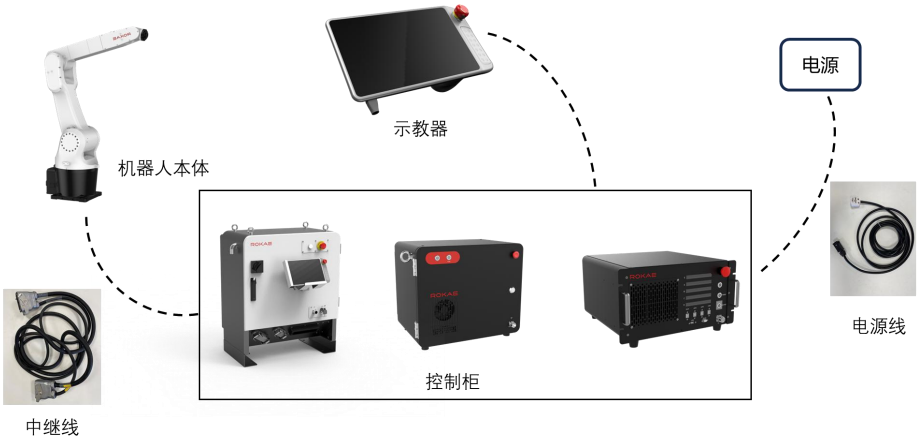
以下介绍 xPad2 示教器的按键及功能。

①	急停按钮；
②	触摸显示屏；
③	实体按键；
④	U 盘插口；
⑤	连接线，用于与控制柜或机器人连接；
⑥	三段使能开关；

5.3 工业机器人系统构成

本章节主要介绍工业机器人结构、接线以及通电开机方法。其中机器人本体和控制柜会因机器人型号的不同而有一定的差异。更多信息，可参考《XBC5 系列控制器（xCore 系统）产品手册》。

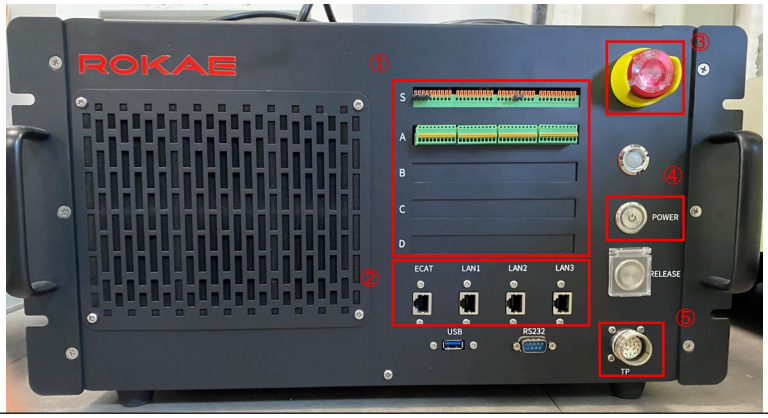
工业机器人系统主要结构及接线关系如下图所示。主要包含：机器人本体、示教器、控制柜、中继线和电源线等。



5.3.1 XBC5 系列控制柜简介

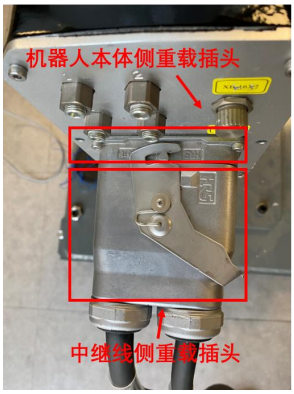
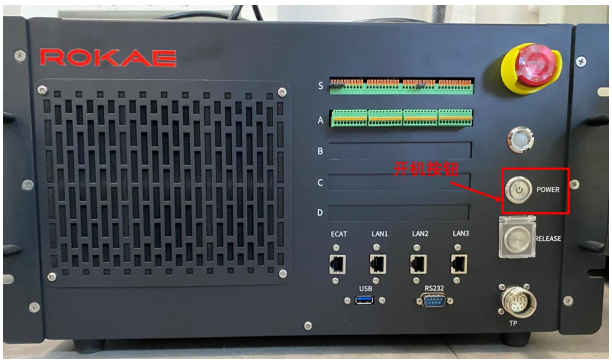
XBC5 系列控制柜包括：XBC5、XBC5-E、XBC5-M 三款。

以 XBC5-M 为例，简单介绍柜体上的主要部件及功能，其他型号类似。

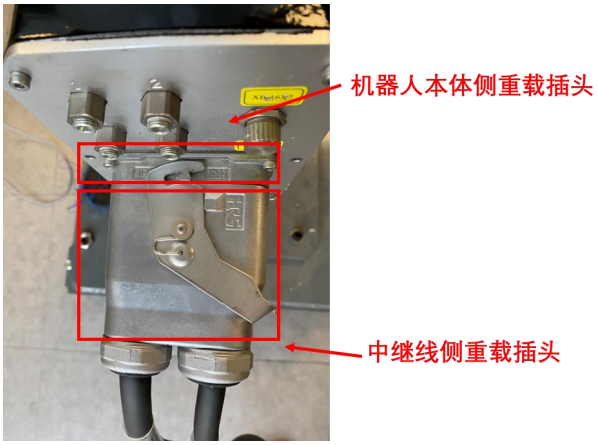


①	安全/通用 IO 接线端子；
②	网口：包括调试接口、EtherCAT 扩展网口以及视觉接口等；
③	急停开关：用于紧急情况控制电机抱闸；
④	电源开关：用于控制机器人开关机；
⑤	示教器接线口：用于连接 xPad2；

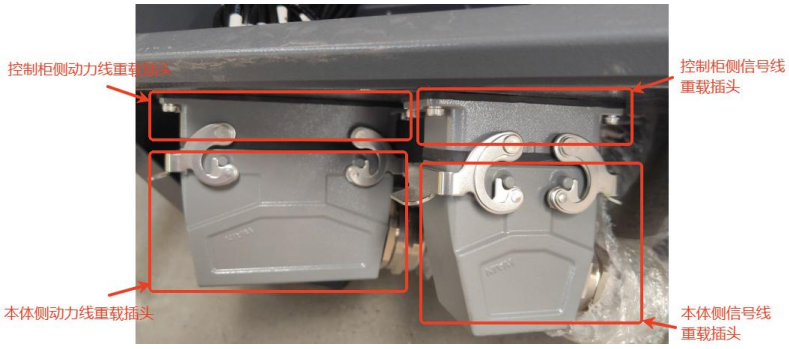
5.3.2 XBC5-M 控制柜接线、通电、开机

步骤	图示	说明
1.通过中继线连接机器人本体与控制器；	 	中继线重载插头两端有区别，连接前请确认。
2.按照图示连接示教器与控制柜；		
3.连接电源线与控制柜；		控制柜侧电源线接口为卡扣设计。
4.通电后开机。		电源线通电后按下开机按钮，xPad2 示教器自动开机。

5.3.3 XBC5 控制柜接线、通电、开机

步骤	图示	说明
1.通过中继线连接机器人本体与控制柜；		控制柜侧电源线和中继线为一体式设计，不需额外安装。
2.按照图示连接示教器与控制柜；		
3.通电后开机。		电源线通电后按下开机按钮，xPad2 示教器自动开机。

5.3.4 XBC5-E 控制柜接线、通电、开机

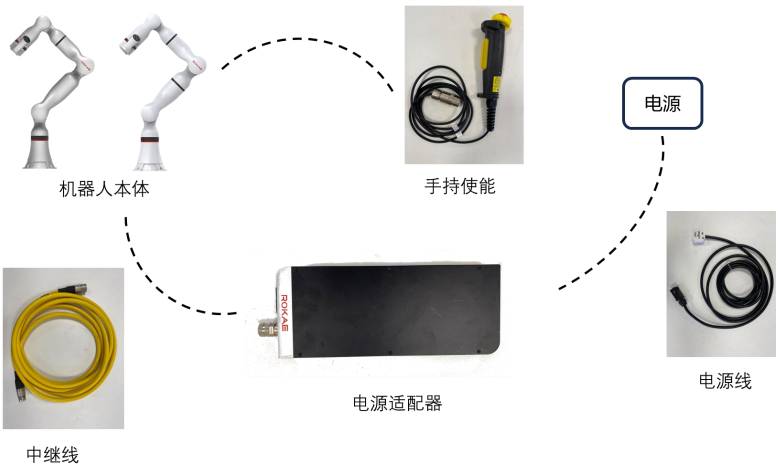
步骤	图示	说明
1.通过中继线连接机器人本体与控制柜;		控制柜侧电源线和中继线为一体式设计，不需额外安装。
2.按照图示连接示教器与控制柜;		
3.通电后开机。		电源线通电后按下开机按钮，xPad2 示教器自动开机。

5.4 协作机器人系统构成

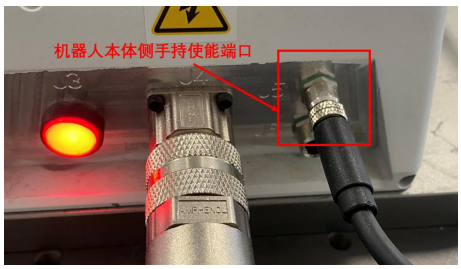
5.4.1 ER 及 ER PRO

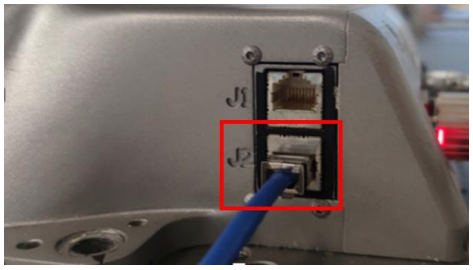
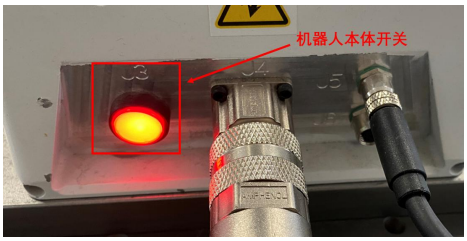
ER 和 ER PRO 系列采用无控制柜设计，系统构成见下图。

注意：ER 系列机器人不支持 xPad2 示教器。



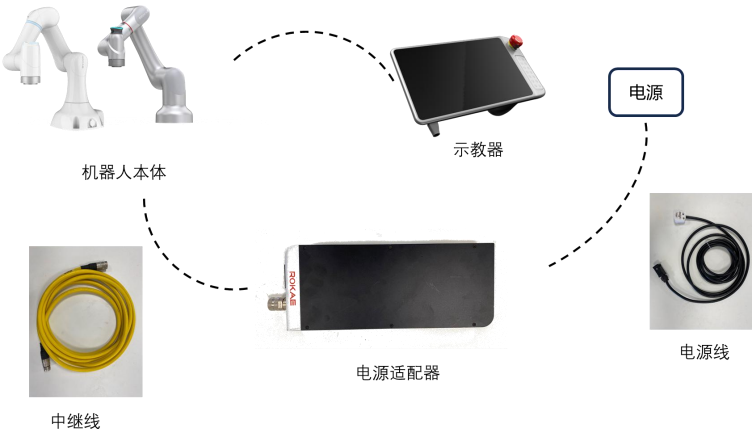
对于 ER 系列机器人，您可以参照以下步骤进行连接和通电。

步骤	图示	说明
1.中继线连接机器人本体和电源适配器；	 	
2.连接手持使能。		

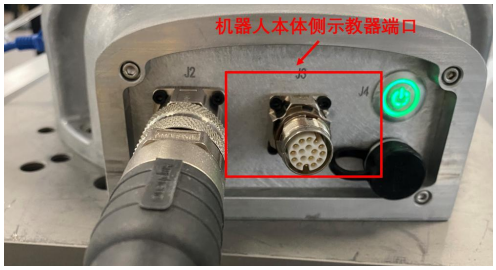
3.连接电源线。		
4.连接 HMI。		ER 系列机器人不支持示教器连接，请通过 PC 连接。详情见下文。
5.连接电源，依次按下电源适配器【开关】和机器人本体电源【开关】。	 	

5.4.2 CR 及 SR

CR 和 SR 系列协作机器人采用无控制柜设计，系统构成见下图。

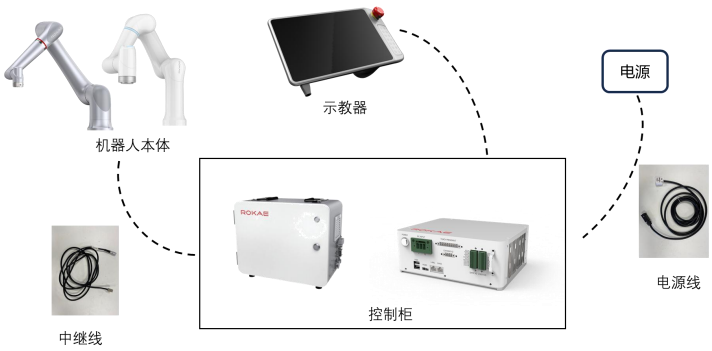


对于 CR 系列机器人，您可以参照以下步骤进行连接和通电。SR 系列与之类似。

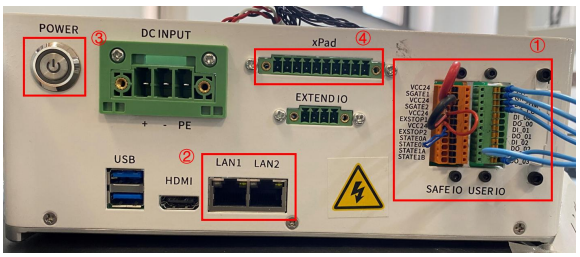
步骤	图示	说明
1.中继线连接机器人本体和电源适配器；	 	
2.连接示教器；		
3.连接电源线；		
4.连接电源，依次按下电源适配器【开关】和机器人本体电源【开关】。	 	机器人开机后，示教器自动开机。

5.4.3 CR-C 及 SR-C

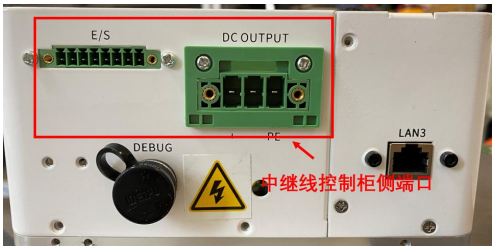
CR 和 SR 系列协作机器人采用有控制柜设计，系统构成见下图。

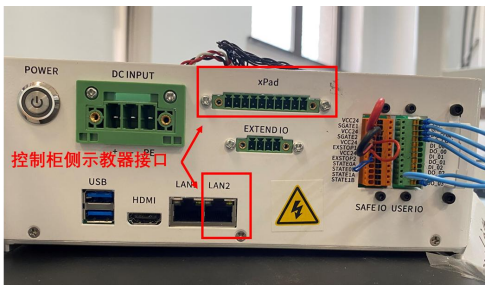
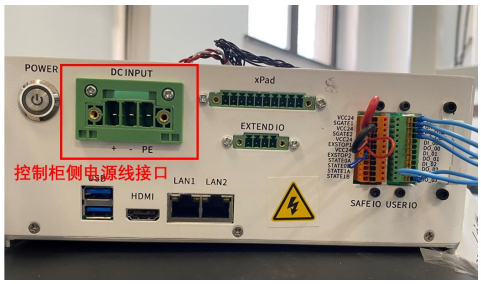
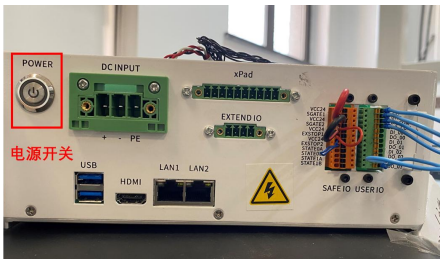


5.4.3.1 SR-C 控制柜及其接线、通电、开机



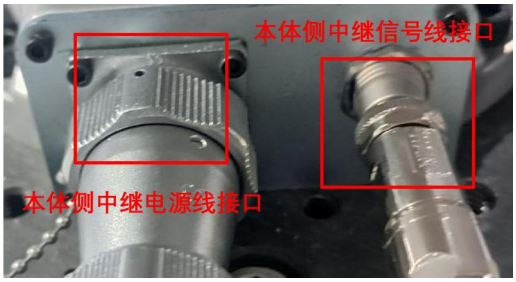
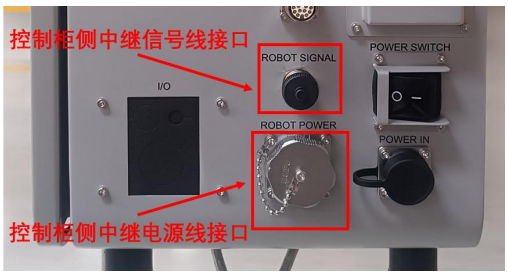
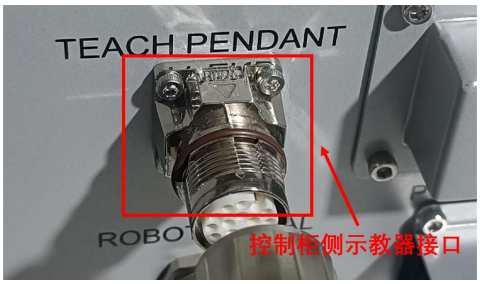
① 安全/通用 IO 接线端子；
② 网口：包括调试接口以及视觉接口等；
③ 电源开关：用于控制机器人上/电状态；
④ 示教器接线口：用于连接 xPad2。

步骤	图示	说明
1.通过中继线连接机器人本体与控制器；	 	中继线重载插头两端有区别，连接前请确认。

2.按照图示连接示教器与控制柜；		
3.连接电源线与控制柜；		控制柜侧电源线接口为卡扣设计。
4.通电后开机。		电源线通电后按下开机按钮，xPad2 示教器自动开机。

5.4.3.2 CR-C 控制柜及其接线、通电、开机

①	安全/通用 IO 接线端子；
②	中继线接口：用于连接机器人和控制柜；
③	示教器接线口：用于连接 xPad2

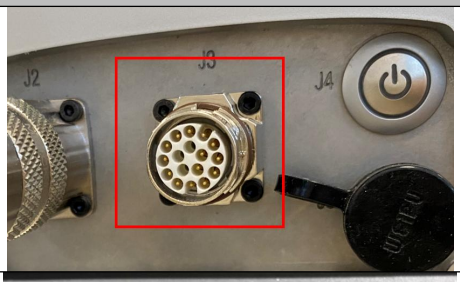
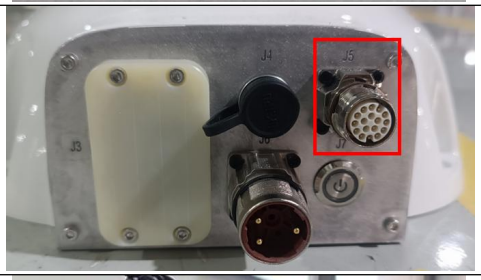
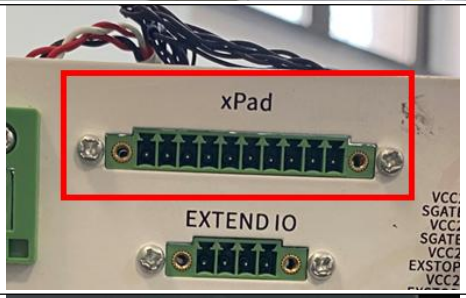
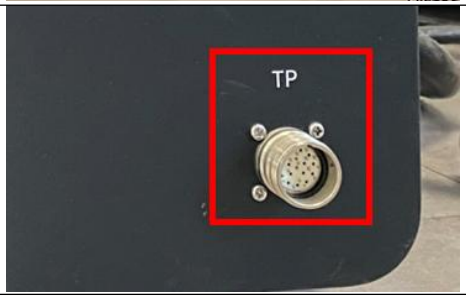
步骤	图示	说明
1.通过中继线连接机器人本体与控制器；	 	注意 CR-C 系列控制柜分为中继电源线和中继信号线两根。
2.按照图示连接示教器与控制柜；		
3.连接电源线与控制柜；		控制柜侧电源线接口为卡扣设计。
4.通电后开机。		电源线通电后按下开机按钮，xPad2 示教器自动开机。

5.5 HMI 与机器人连接

Robot Assist 是机器人的的上位机软件，可运行在 PC、xPad2 等设备上。将 Robot Assist 软件所在设备与机器人接入同一局域网，可选择机器人探测、手动输入控制器服务地址等方式，与连接的机器人建立连接。

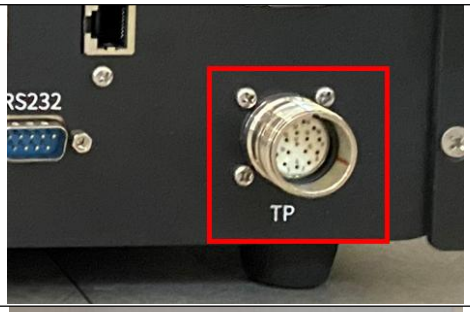
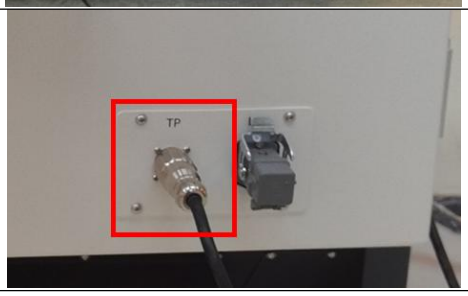
5.5.1 xPad2 与机器人连接

对于使用示教器 xPad2 的情况，示教器默认网段为 192.168.1.X，需先修改示教器的 IP 与机器人本体同一网段，再将 xPad2 连接到机器人对应的插口即可；

机型	简介	配图
CR	xMate CR 系列机器人 xPad2 接线口位于底座处。	
CR-C	xMate CR-C 系列机器人 xPad2 接线口位控制柜上部。	
SR	xMate SR 系列机器人 xPad2 接线口位于底座处。	
SR-C	SR 专用控制柜（需增加转接口）	
XBC5	XBC5 系列控制柜 xPad2 接线口位于控制柜底部。	

5.5.1.1 硬件连接

5.5.1.2 连接配置

XBC5-M	XBC5-M 系列控制柜 xPad2 接线口位于控制柜底部。	
XBC5-E	XBC5-E 系列控制柜 xPad2 接线口位于控制柜底部。	

xPad2 在完成硬件连接后，机器人开机，xPad2 将自动开机并启动内置的 Robot Assist 软件。

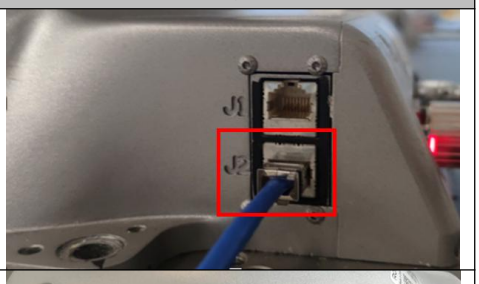
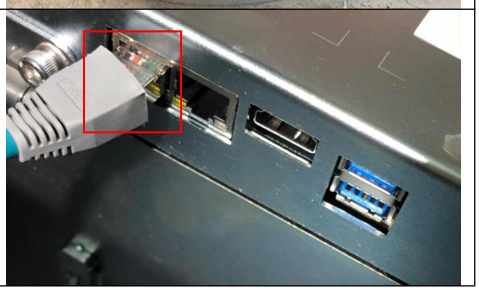
5.5.2 PC 与机器人连接

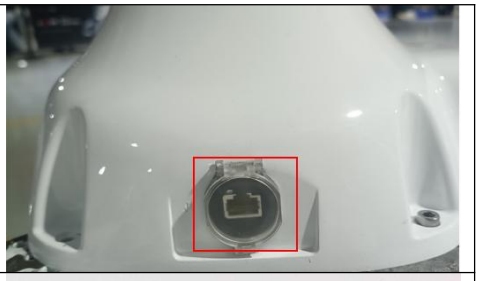
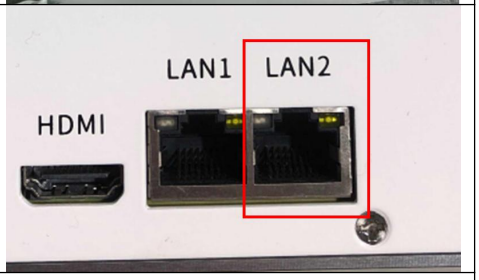
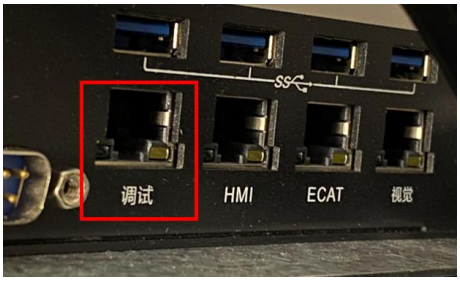
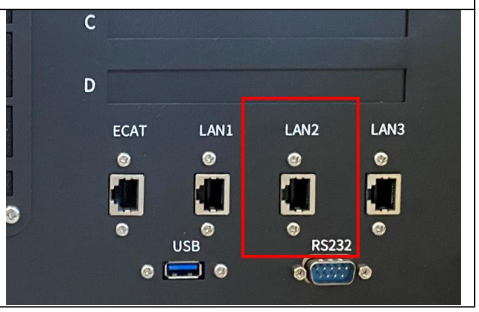
RobotAssist 软件可以运行在 PC 上，再将 PC 与机器人或控制柜建立连接。

5.5.2.1 硬件连接

5.5.2.2 HMI 与机器人一对一

对于使用运行 RobotAssist 的 PC 的调试机器人的情况，PC 可以直接通过网线与机器人系统(表格+配图具体化)连接；

机型	简介	配图
ER/ER PRO	xMate ER 系列协作机器人，底座配有两个以太网口，其中 J2 网口出厂默认配置为固定 IP 192.168.0.160。	
CR	xMate CR 系列协作机器人，底座标配的以太网口只有一个 J1，出厂默认配置为固定 IP 192.168.2.160	
CR-C	xMate CR 有控制柜	

SR	xMate SR(J2 网口)	
SR-C	xMate SR-C 调试网口为 LAN2，默认 IP 为 192.168.0.160；	
XBC5/XBC 5E	控制柜上自左至右有四个以太网口，依次为： ● 调试网口，默认配置为固定 IP 192.168.0.160； ● EtherCAT 设备扩展网口，用于从站拓展； ● 视觉网口，用于连接工业相机，默认配置为固定 IP 192.168.2.160； ● 总线扩展网口，选配。	
XBC5 M	LAN2 为 XBC5 M 控制柜调试网口，默认 IP 为 192.168.0.160；	

5.5.2.3 HMI 与机器人一对多

对于需要切换多台机器人连接的，可以将多台机器人接入同一局域网，运行 RobotAssist 的 PC 通过同网段探测，寻找可连接的机器人；

5.5.2.4 无线连接

对于 AGV 搭载等不方便有线连的场景，可以通过机器人控制柜预留的网口（xMate 协作机器人底座网口，工业机器人控制柜视觉/调试网口）与无线路由连接后，再与 HMID 通过无线连接；

5.5.2.5 连接配置

5.5.2.6 网线直连

机器人底座或控制柜上都有一个网口预设为调试网口，配置固定 IP 地址为 192.168.0.160，该地址在所有机器人上均相同，且不建议轻易修改。您可以将运行 RobotAssist 的 PC 通过网线与此网口直连，控制机器人。

5.5.2.7 外网口连接

外网口连接机器人支持两种方式，自动获取 IP 地址或者设置静态 IP 地址的方式。

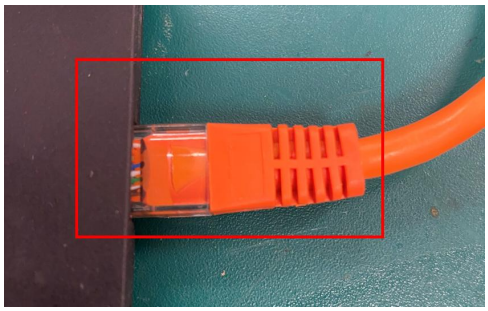
需要自动获取地址时，协作机器人可将 J1 网口设置为 DHCP 模式，工业机器人可将视觉网口设置为 DHCP 模式，机器人通过此网口接入到 DHCP 功能的路由中，用于自动分配 IP 地址给机器人，这时可以通过机器人探测功能探测到机器人并连接。

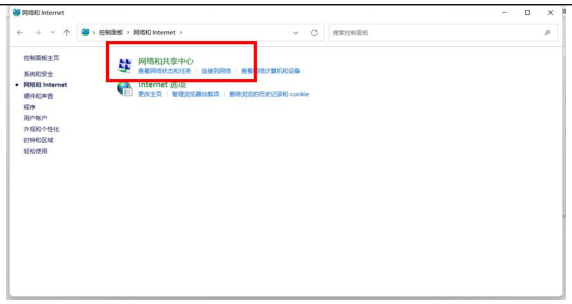
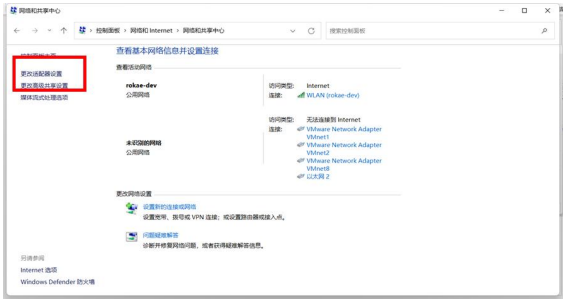
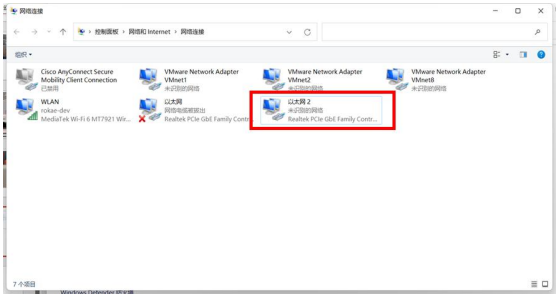
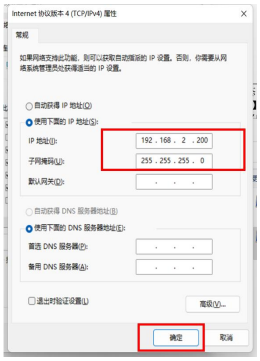
需要使用静态 IP 地址时，协作机器人可将 J1 网口设置为所需网段的 IP 地址，工业机器人可将视觉网口设置为所需网段 IP 地址，机器人通过此网口接入到路由中，这时可以通过机器人的 IP 地址访问和控制机器人。

5.5.2.7.1 PC 等设备网线直连

机器人底座或控制柜上都有一个网口预设为调试网口，配置固定 IP 地址为 192.168.0.160，该地址在所有机器人上均相同，且不建议轻易修改。您可以将运行 RobotAssist 的 PC 通过网线与此网口直连，控制机器人。

使用 PC 等移动设备连接机器人时，需保证移动设备的 LAN 网口地址与机器人处于同一网段。关于 PC 端（win11）静态 IP 的修改方法及机器人（CR 系列）连接，可以参照以下流程步骤：

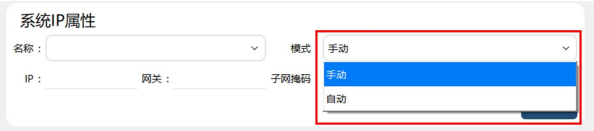
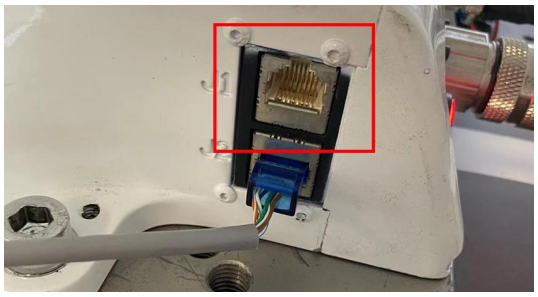
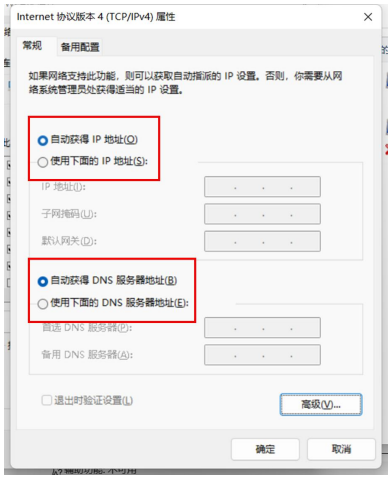
步骤	图示	说明
1.网线连接机器人。网线一端接入笔记本网口，一端接入机器人网口。	 	CR 系列机器人底座侧端网口默认网段为“192.168.2.XX”。
2.修改本地静态 IP。进入 PC 端【控制面板】->【网络和 Internet】->【网络和共享中心】->【更改适配器设置】->右键打开对应网口【属性】->双击【Internet 协议版本 4（TCP/IPv4）】->修改终端设备（PC）的 IP 地址、子网掩码以及默认网关并【确认】。		可将终端设备（PC）的 IP 地址修改为与机器人同一网段未被占用的任意 IP 地址，其子网掩码和默认网关与机器人的一致。

	    	
3.HMI 与机器人连接		

5.5.2.7.2 PC 等设备无线连接

协作机器人可将 J1 网口设置为 DHCP 模式，工业机器人可将视觉网口设置为 DHCP 模式，机

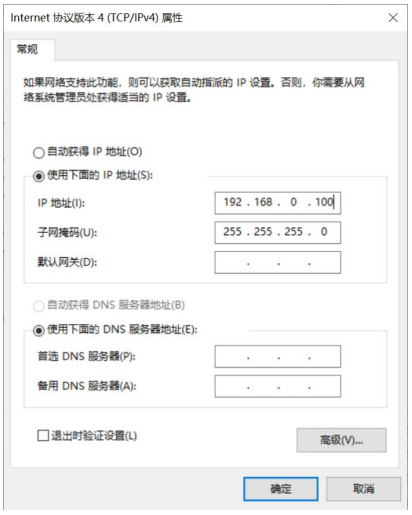
器人通过此网口接入到 DHCP 功能的路由中，用于自动分配 IP 地址给机器人，这时可以通过机器人探测功能探测到机器人并连接。

步骤	图示	说明
1.修改机器人系统 IP 属性为自动模式。		详情参见章节 6；
2.机器人接入路由设备。		协作机器人可将 J1 网口设置为 DHCP 模式，工业机器人可将视觉网口设置为 DHCP 模式，机器人通过此网口接入到 DHCP 功能的路由中，用于自动分配 IP 地址给机器人。
3.PC 端接入相同网段的路由网络，将获取 IP 模式设置为 DHPC。		进入【Internet 协议版本 4（TCP/IPv4）】页面方法参见手动修改 IP 部分步骤 2。
4.HMI 连接机器人		

5.5.2.7.3 关于 IP 修改

以 Windows10 操作系统为例，将网线一端连接至机器人的 J2 接口，另一端连接至终端设备（PC）；在终端设备（PC）上单击“开始 > 控制面板”菜单选择“网络和共享中心”；弹出“网络和共享中心”窗口；在“网络和共享中心”窗口单击“本地连接”；弹出“本地连接状态”界面；在“本地连接状态”界面单击“属性”,弹出“本地连接属性”界面；在“本地连接属性”界面双击“Internet 协议版本 4（TCP/IPv4） ”；弹出“Internet 协议版本 4（TCP/IPv4）属性”界面；在“Internet 协议版本 4（TCP/IPv4）属性”界面选择“使用下面的 IP 地址”，修改终端设备（PC）的 IP 地址、子网掩码

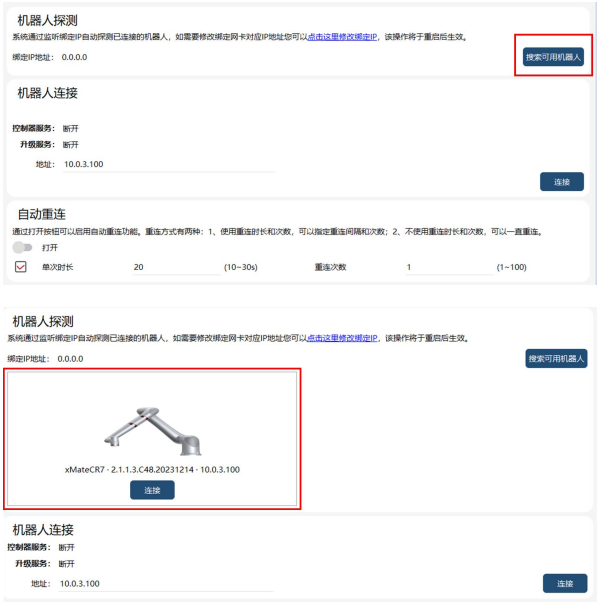
以及默认网关并确认。（可将终端设备（PC）的 IP 地址修改为与机器人同一网段未被占用的任意 IP 地址，其子网掩码和默认网关与机器人的一致）



5.5.3 机器人探测与连接

> **警告**
>
> 手动修改机器人的网口 IP 地址时，切勿将不同的网口设置为同一网段的静态 IP 地址；不要轻易修改调试网口的网络模式和 IP 地址（192.168.0.160）；不要轻易修改示教器 xPad 适配网卡的网络模式和 IP 地址（192.168.1.160）。

HMI 可探测同一网段内所有可以连接的机器人并显示。您可以通过以下步骤探测并连接机器人。

步骤	图示	说明
1.搜索可用机器人。点击底部状态栏网络图标按钮，快捷进入到搜索机器人界面，点击【搜索机器人】可以探测到可用的机器人。		搜索机器人时请确保运行 RobotAssist 的设备和机器人处于同一网络中，网络互通。
2.连接机器人。输入机器人 IP 地址，点击【连接】。		机器人连接成功时，【控制器服务】和【升级服务】会显示“连接到 XXXX”。底部状态栏图标会变为。不支持多 RobotAssist 软件同时连接，当前 RobotAssist 软件断开

		连接确认后或者机器人重启，其他 RobotAssist 软件可连接。
3.断开机器人。在连接界面点击“断开”按钮，可切除 RobotAssist 软件和控制器的连接。		恢复 RobotAssist 软件连接与初始连接过程一致。

6 HMI 简介

6.1 本章简介

本章概述 xCore 的 HMI 软件基本布局、主要功能的分布和作用。
用户在实际操作使用机器人之前，请先阅读本章。

6.2 RobotAssist 软件简介

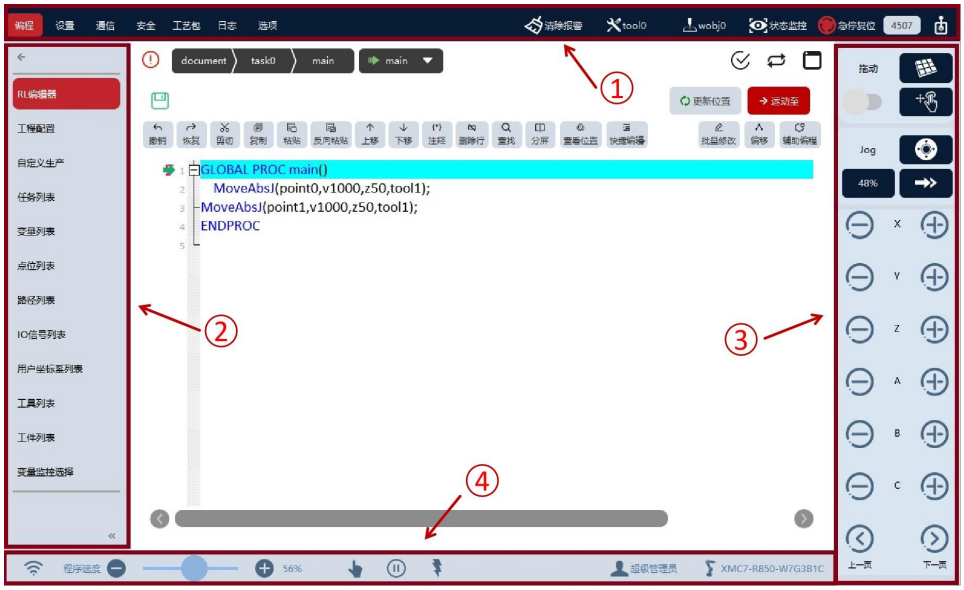
Robot Assist 是 xCore 控制系统的上位机软件，具有机器人移动控制、任务编辑、参数设置、状态监控等功能。该软件可安装于 PC、Surface、xPad2 示教器上，只要与机器人处于同一网段内，连接机器人便可操控机器人。
当您使用 xPad 2 时，如果示教器遇到崩溃或蓝屏等问题，RobotAssist 软件将自动重新启动以恢复操作。（请注意，此自动重新启动功能不适用于 PC 端。）
若选择不使用 xPad2，建议您使用平板电脑或者笔记本电脑作为操作终端，推荐配置如下。

终端类型	平板电脑	终端类型	电脑
存储容量	32GB	存储容量	32GB
系统内存	4GB	系统内存	4GB
屏幕尺寸	8.0 寸及以上	显卡	Intel HD Graphics 4000 或更高
网络通信	Wi-Fi	网络通信	Wi-Fi 或有线网卡
操作系统	Windows7 64bit ，Windows10 64bit，Ubuntu20.04		
处理器	Intel Core I3 及以上		

> 提示
> RobotAssist 软件与控制器实时交互，在 PC 端使用时，如频繁改变窗口大小可能导致监控数据停止更新，此时点击 Alt+Tab 切换窗口即可恢复。

6.3 HMI 整体布局

操作主界面通常由 4 个主要区域构成，包括：顶部状态栏、底部状态栏、左侧边栏、右侧操控面板。



①	顶部状态栏
②	左侧边栏
③	右侧操控面板
④	底部状态栏

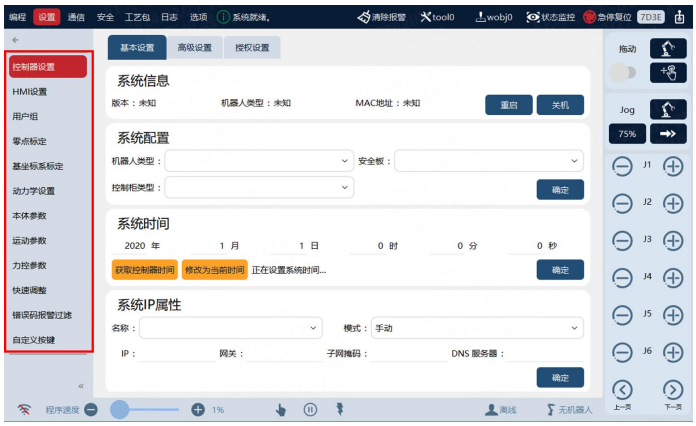
6.3.1 顶部状态栏

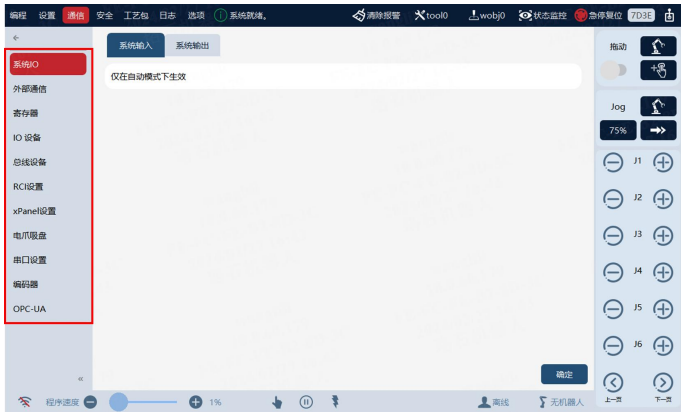
顶部状态栏，包含：若干一级菜单按钮(编程、设置、通信、安全、工艺包、日志、选项)、即时日志、清除报警按钮、工具工件信息按钮、状态监控按钮、RSC 复位按钮、安全校验按钮、操控面板按钮。

①	一级菜单按钮，包含编程、设置、通信、安全、工艺包、日志、选项等若干一级菜单按钮，点击可跳转到功能子界面；
②	即时日志，显示系统最新的控制器日志信息，点击可跳转到控制器日志子界面；
③	清除报警按钮，点击可清除即时日志与控制器的报警状态；
④	工具选择按钮，显示当前使用工具信息，单击后可选择工具；
⑤	工件选择按钮，显示当前使用工件信息，单击后可选择工件；
⑥	状态监控面板按钮，点击可打开、关闭状态监控面板；
⑦	急停复位按钮（仅对配置了安全控制器的机器人开放），当机器人处于急停、安全门或安全停止状态时，在消除导致以上状态的因素后，点击此按钮可恢复机器人状态；
⑧	安全校验和，用于校验涉及安全的五个页面：软限位、虚拟墙、碰撞检测、安全监视器、协作模式。当这五类设置改变时，将弹出本页面进行再次确认；
⑨	操控面板入口按钮，点击可打开、关闭操控面板；

6.3.2 左侧边栏

当通过顶部状态栏切换不同的功能，例如编程、设置、通信等，左侧边栏会显示相应功能的子菜单。

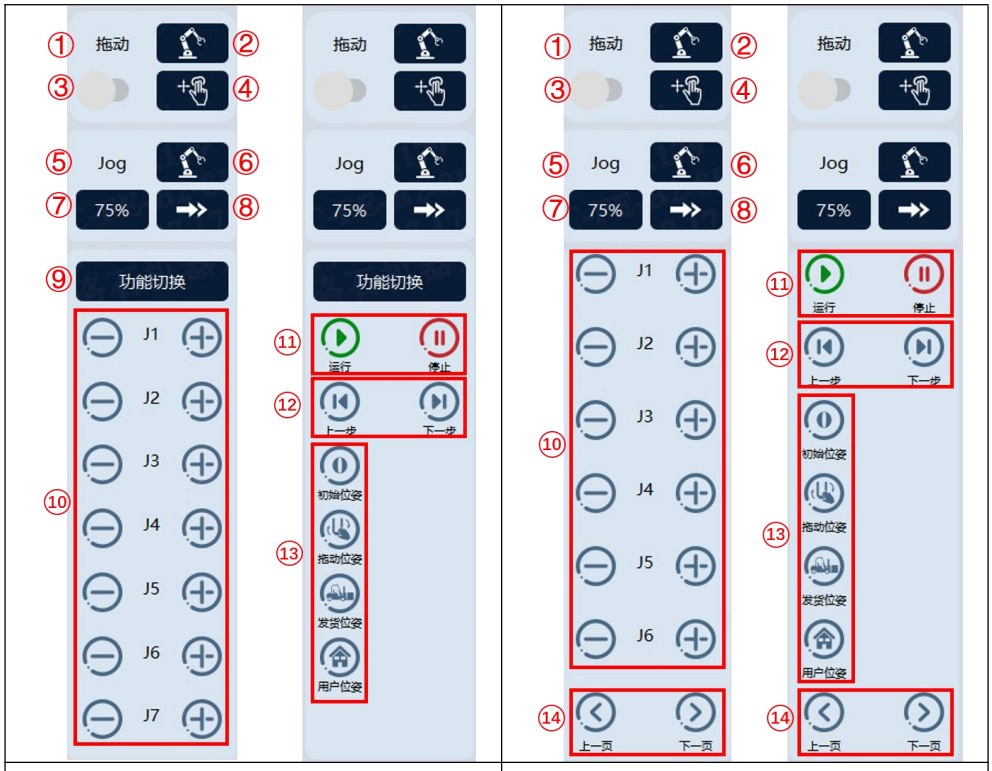
	在顶部状态栏点击“设置”，左侧边栏显示所有“设置”子菜单。
	点击“本体参数”，进入“本体参数”设置页面。

	
	在顶部状态栏点击"通信"，左侧边栏显示所有"通信"子菜单。
	再次在顶部状态栏点击"设置"，左侧边栏显示所有"设置"子菜单，并默认进入上一次进入的"本体参数"页面。

6.3.3 右侧操控面板

点击顶部状态栏的 或 按钮，可打开右侧操控面板，操控面板用于切换机器人控制模式，对机器人进行移动控制、位姿示教。

机器人支持两种方式示教机器人位姿：Jog 模式和拖动模式（仅限协作机器人）。Jog 模式，通过 Jog 按钮控制机器人在相应方向运动；拖动模式，通过末端拖动 Pilot/xPanel 手柄，直接手动引导机器人运动。



7 轴机器人操控面板	6 轴及以下机器人操控面板

①	拖动设置区，表示②、③、④跟拖动相关。
②	拖动空间设置：轴空间拖动、笛卡尔空间拖动。
③	拖动使能开关按钮，用于开启、关闭拖动模式。
④	拖动方式设置；轴空间仅支持：自由拖动；笛卡尔空间支持三种：仅平移拖动、仅旋转拖动以及自由拖动。
⑤	JOG 设置区，表示⑥、⑦、⑧跟 Jog 相关。
⑥	Jog 参考系设置，用来选择 Jog 时的单轴模式和笛卡尔模式以及在笛卡尔模式下的参考坐标系，包括：世界坐标系、基坐标系、法兰坐标系、工具坐标系、工件坐标系。
⑦	Jog 速率设置，用来调节 Jog 时的运动速率，调整范围 1~100%，为相对 Jog 最高速度限制 250mm/s 的百分比。
⑧	Jog 步进模式设置，可选连续 Jog 和步进 Jog，并且可以调整增量步进的大小。
⑨、⑭	功能区切换，可以进行 Jog 按钮与⑪、⑫、⑬按钮的切换。
⑩	Jog 按钮。对于 7 轴机器人，轴空间 Jog 时显示 J1~J7，笛卡尔空间 Jog 时显示 X/Y/Z/A/B/C 及 Elbow；对于 6 轴机器人，轴空间 Jog 时显示 J1~J6，笛卡尔空间 Jog 时显示 X/Y/Z/A/B/C。
⑪	程序启动、停止按钮。
⑫	程序运行上一步(暂不支持)、下一步按钮。
⑬	快捷按键区，可以在自定义按键界面配置该区域按键的快捷功能。默认配置四个按键，初始位姿、拖动位姿、发货位姿、用户位姿。 更详细的介绍，参考"设置"-"快速调整"章节。

提示

请在 JOG 和打开拖动使能开关前，确认当前机器人工作模式为手动模式且下电状态。

当有绑定了暂停功能的寄存器或系统 IO 没有复位时，不允许启动程序。

6.3.4 底部状态栏

底部状态栏显示 RobotAssist 软件与机器人连接状态、程序运行速度、机器人工作模式、机器人状态、机器人电机状态、当前用户登录信息和机器人型号信息。

①	RobotAssist 软件与机器人的连接状态，点击该按钮可打开于机器人的连接设置页面。图标显示为未连接；为已连接；出现动画为正在尝试连接机器人；为升级服务已连接、控制器服务未连接状态，该状态下可以进行控制系统升级操作，无法操作机器人和设置机器人参数。
②	程序运行速度调整控件，用来调节 RL 程序运行时的运动速度，可调范围 1%~100%，该参数分别独立影响手动和自动两种模式的程序运行速度。点击滑条或者-/+按钮，可以微调程序速度（-/+1%）。 注意：程序速度上限，可能受"控制器设置"-"高级设置"中的"手动模式下程序速度上限"、"自动模式程序初始速度上限"影响，可以参考相关内容介绍章节了解更多信息。
③	机器人工作模式，分为手动和自动，在"控制系统基础操作"章节有更详细介绍。 为手动模式，通常用户在此模式进行程序编写及调试。 为自动模式，通常用户在此模式下进行连续自动化生产。
④	机器人状态，包括以下具体状态。
	，空闲状态；程序停止状态，机器人未运动。
	，程序运行；程序运行中，当机器人运动会变成红色。
	，拖动模式；控制器处于拖动模式下，机器人可以进行拖动，当机器人运动会变成红色。
	，Demo 模式；控制器处于 Demo 演示模式下，可以开始 Demo 演示，当机器人运动会变成红色。
	，辨识模式；控制器处于辨识模式下，当机器人运动会变成红色。
	，Jog 模式；控制器处于 Jog 模式下，随 Jog 按钮开始、结束而改变。
	RCI，RCI 模式；控制器处于 RCI 模式下，当机器人运动会变成红色。
	，协作模式；控制器处于协作模式下，会与其他状态进行组合显示再图标的右上角。
	，错误状态；机器人系统处于错误状态。
	，调试模式；机器人系统处于调试模式，当机器人运动会变成红色。
⑤	工程半静态状态；当机器人处于半静态任务运行状态时，显示该按钮；否则不显示该按钮；点击该按钮可以停止半静态任务。
⑥	，上电状态；机器人电机处于上电状态；单击可以下电。
	，下电状态；机器人电机处于下电状态；单击可以上电。
	E，急停状态；机器人处于紧急停止状态，此时无法给机器人电机上电。
	G，安全门状态；机器人处于安全门打开状态，此时无法给机器人电机上电。
	S，安全停止状态(仅限配置了安全控制器的机型)；机器人处于安全停止状态，表示安全控制器检测到工作、通信异常或者某参数超出安全控制器设置安全阈值，此时无法给机器人上电。
⑦	当前登录用户信息：操作员、示教员、编程员、管理员、超级管理员；点击按钮可

	跳转用户登录界面。
⑧	机器人型号信息。

6.4 状态监控

点击顶部状态栏状态的"状态监控"按钮，可打开浮动的状态监控面板。通过状态监控面板可监控：机器人 3D 模型、任务运行状态、IO 信号、网络连接状态、寄存器变量、传送带状态、PERS 变量信息，方便用户快速了解机器人状态。

6.4.1 3D 模型监控

该界面以三维模型的方式直观展示机器人当前状态，三维模型可以通过点击拖动的方式切换观察视角，也可以通过点击"正视"按钮让模型回归默认视角。

①	坐标系切换：可选基坐标系/世界坐标系/工件坐标系。在机器人非正装情况下，或使用机器人组时，基坐标系与世界坐标系可能不重合。
②	正视：点击"正视"按钮可让模型回归默认视角。
③	末端姿态：机器人末端相对某坐标系（工件坐标系、基坐标系、世界坐标系）下的位置和姿态(RPY 或四元数)。
④	关节角度、关节转矩、外部轴信息。

> 提示
> 目前仅有 PC 端的 RobotAssist 软件支持 3D 模型显示，xPad2 上只能展示状态数据。

6.4.2 任务监控

该界面显示当前工程中各个任务的任务名称、任务类型、运行状态。

6.4.3 IO 信号监控

对于 xMate 系列协作机器人，IO 信号监控界面默认显示本体底座上的 4 路 DI、DO 信号以及末端 2 路 DI、DO 信号。

对于工业机器人，IO 信号监控界面默认显示控制柜内已配置的 IO 信号。

①	过滤器，用于筛选显示的 IO，可选择的过滤条件包括：类别、IO 板、信号类型、名称，点击"重置"按钮可重置过滤条件
②	IO 信号列表
③	打开 IO 仿真模式，可对 DI 信号值模拟

IO 仿真模式使用步骤：

步骤	图示	说明
1.进入 IO 信号页面，点击【IO 仿真模式】使能按钮，开启仿真模式。		有权限限制，需要是操作员以上权限才能使用。
2．点击相应 DI 或者 DO 所对应的修改值按钮，可以对其进行仿真。		注意，非仿真模式 DO 也可以强制输出。
3.点击【IO 仿真模式】使能按钮，关闭仿真模式。		实际值和修改值按钮没有强关联，关闭仿真模式后，修改值按钮会统一置为 false。

6.4.4 网络连接监控

该界面显示当前与控制器建立连接的网络信息，包括：名称、类别、IP、端口、状态。

①	名称：MODBUS、RCI、SYS_SOCKET 为固有名称，属于系统默认，用户新建的会显示其自定义名称。
②	类别：可支持显示 MODBUS、RCI、SOCKET 等类型连接。相应连接可以在相关界面进行增加或配置，SYS_SOCKET 特指外部通信对应连接。
③	IP：如果是客户端性质的连接会显示目标服务端的 IP，如果是服务端会显示自身 IP。
④	端口：如果是客户端性质的连接会显示目标服务端的端口，如果是服务端会显示自身端口。
⑤	状态：一般有三种，已连接、已关闭、正在连接；如果是服务端性质连接未被连接时会显示正在监听。

6.4.5 寄存器监控

该界面显示各寄存器的信息。如果觉得默认显示的寄存器过多，可进行过滤，可选择的过滤条件包括：设备、类型、读写、名称、描述，点击"重置"按钮进行更新。

①	过滤器：用于筛选显示的寄存器。
②	寄存器信号列表。

步骤	图示	说明
只写寄存器赋值：选中只写寄存器，在"写入"按钮左侧的输入框中输入期望值，点击"写入"按钮即可更改该寄存器的当前值。		1、操作员权限不可写入； 2、只有只写寄存器、并且未绑定功能码的寄存器才能进行赋值。
只读寄存器赋值：打开"只读寄存器仿真模式"，在"写入"按钮左侧的输入框中输入期望值，点击"写入"按钮即可更改该寄存器的当前值。		

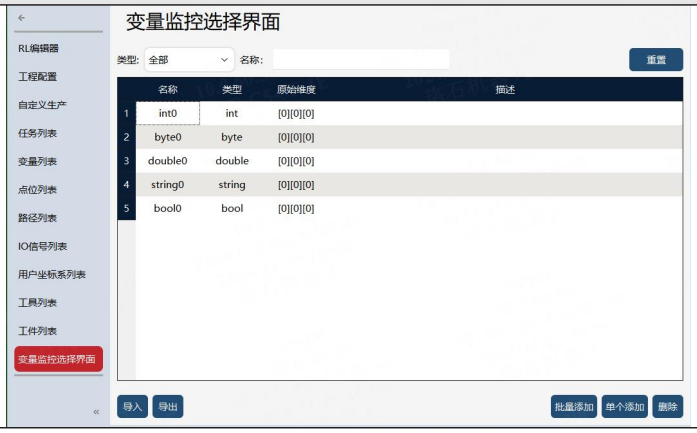
6.4.6 传送带监控

用于配合传送带工艺包，监控传送带状态，详细介绍见传送带跟踪工艺包。

6.4.7 变量监控

该界面显示变量列表中的变量的实时信息。

①	过滤器：可根据类插入型及名称过滤器进行过滤。注意：目前变量监控只支持对 bool、int、double、byte、string 五种类型单个变量或者数组监控。
②	监控变量列表 。
③	翻页按钮，一页只支持 10 个变量的监控。
④	写入按钮：为变量赋值。

步骤	图示	说明
1.在变量监控选择界面，通过"批量添加"按钮或者"单个添加"按钮添加需要监控的变量。		变量监控只监控选择界面存在的变量
2.写入：在"修改值"一列，写入要修改的变量值，点击写入按钮，可将值实时写入变量。		操作员权限不可写入。当修改的变量为非 pers 时，点击 pptomain 会将非 pers 的更改值复原。

注意：新增变量监控后，需要先执行 pptomain 操作，才能对变量数据进行读取及修改。

6.5 编程模块概览

RL 编辑器	主要用于 RL 程序的编写和调试。
工程配置	工程新建、加载、导入导出、推送等操作。
自定义生产	用户可以自定义若干控件，方便地实现对寄存器、DI/DO 信号、工程变量的监控和编辑。
任务列表	用于任务的查看、新建、编辑、导入导出等。
变量列表	变量的查看、新建、编辑、导入导出等。
点位列表	点位的查看、新建、编辑、导入导出等。
路径列表	路径的查看、新建、编辑、导入导出等。
IO 信号列表	IO 信号的查看、新建、编辑、导入导出等。
用户坐标系列表	用户坐标系的查看、新建、编辑、导入导出等。
工具列表	工具坐标系的查看、新建、编辑、导入导出等。
工件列表	工件坐标系的查看、新建、编辑、导入导出等。
变量监控选择界面	状态监控中所监控的变量，可进行监控变量的导入、导出、批量添加、单个添加以及删除等操作。

6.6 设置模块概览

控制器设置	控制器系统相关设置界面，包括机器人类型、系统时间、系统 IP 等。
HMI 设置	HMI 相关的设置，包括语言切换、示教器 IP 设置等。
用户组	用户管理，包括登录和密码管理。
标定	进行机器人零点标定、力传感器标定、软标定、基坐标系标定
坐标系标定	设置全局工具、全局工件、全局用户坐标系
动力学设置	设置机器人动力学相关设置。
本体参数	设置机器人运动学相关设置，例如 RD 参数、减速比、过载合系数等。
运动参数	机器人加减速性能、安全控制等设置。
力控参数	机器人的力控制相关设置。
快速调整	快速调整姿态设置。
电子铭牌	电子铭牌相关设置。（该功能仅部分机型支持）
错误码报警过滤	错误码过滤相关设置。
自定义按键	设置操控面板按键绑定的功能。

6.7 通信模块概览

系统 IO	系统输入、系统输出信号设置。
外部通信	基于 Socket 的外部通信接口设置。
寄存器	寄存器相关设置。
IO 设备	IO 类设备设置。
总线设备	配置各类的总线扩展模块。
RCI 设置	RCI 通信设置。
xPanel 设置	CR 机器人 xPanel 设置。
电爪吸盘	各类电爪、吸盘的设置和测试。
串口设置	串口相关的设置。
编码器	传送带跟踪功能所需的编码器设置。
OPC-UA	OPC-UA 相关设置。

6.8 安全模块概览

关节限制	机器人关节限位设置。
机器人限制	机器人速度，缩减模式速度，功率，动量等设置。
虚拟墙	虚拟墙相关设置。
碰撞检测	碰撞检测相关的设置。
安全区域	安全区域相关的设置。
安全位置	安全位置相关的设置。
安全 DO 设置	安全 DO 相关的设置。

6.9 工艺包模块概览

支持码垛、料盘、传送带跟踪、光伏排版、光伏插片等工艺包，请参考相应章节。

6.10 日志模块概览

HMI 日志	显示当前操作界面日志信息。
控制器日志	显示当前连接机器人控制器运行日志。
运行日志	显示机器人运行相关日志。
日志时间轴	通过时间轴直观显示日志历程。
内部日志	用于显示示教器或 RobotAssist 的底层日志信息。
诊断设置	辅助用于辅助开发人员进行问题诊断。
硬件状态	增加工控机和示教器健康监测信息显示。
诊断数据监控	仅电脑版 HMI 支持，可使用工况校核功能。示教器没有该功能。

6.11 选项模块概览

连接	RobotAssist 软件与控制器连接，相关的操作和设置。
关于珞石	版本信息、公司介绍。
软件升级	控制系统软件升级、备份相关的操作。
导出	控制系统配置导出。
导入	控制系统配置导入。
文件管理器	RobotAssist 软件涉及的若干文件夹。
演示	部分特性功能演示。

7 控制系统基础操作

7.1 本章简介

本章概述机器人最常用的基础必备操作。
用户在实际操作使用机器人之前，请先阅读本章。

7.2 工作模式

机器人工作模式分为手动模式和自动模式。

7.2.1 手动模式

手动模式主要用于机器人程序编写及调试。在手动模式下，机器人的所有运动均由用户手动控制，机器人只有在运动使能时（三位使能开关处于中间位置）才会给电机通电并响应运动指令。

手动模式通常用来执行以下任务：

- 紧急停止后将机器人 Jog 回路径附近以便继续运行程序；
- 创建、编写 RL 程序；
- 调试 RL 程序，包括但不限于启动、停止、单步运行、更新程序位置等；
- 设置控制系统参数、标定坐标系等；
- 查看、修改变量；

7.2.2 自动模式

自动模式用于连续的自动化生产，在该模式下三位使能开关将不再起作用，机器人可以在没有人员参与的情况下正常工作。机器人处于自动模式时，可以通过系统 IO 信号等来控制机器人，以及获取机器人工作状态。

自动模式下通常被用来执行以下任务：

- 加载，启动及停止 RL 程序；
- 急停后返回原编程路径；
- 备份系统；
- 清洗工具（根据工艺要求）；
- 对工件进行加工、处理；

自动模式下的使用限制包括不限于：

- 无法进行 Jog。
- 不允许修改配置文件，配置 IO 板个数，设置机器人安装方式。
- 不允许备份恢复。
- 不允许功能授权。
- 不允许设置软限位。
- 不允许新建，修改，删除 IO。
- 不允许参数辨识。
- 不允许开启/关闭碰撞检测。
- 不允许开启/关闭协作模式。
- 不允许开启/关闭拖动示教。
- 不允许标定。
- 不允许新建变量。
- 不允许升级/恢复出厂设置。

7.2.3 模式确认和切换

通过观察 HMI 软件底部状态栏工作模式图标，即可知道当前控制系统所处的工作模式。

	控制器处于手动模式。
	控制器处于自动模式。

用户可通过点击 HMI 界面上的工作模式图标，进行工作模式切换操作。切换前，会弹框确认，点击"确定"，系统将切换工作模式；点击"取消"，将不进行切换。

注意：安全起见，模式切换时，系统将会切断动力电。假如系统正在执行 RL 程序且机器人正在运动中，系统将触发 STOP 1 停止。

7.2.3.1 手动切至自动

当操作人员需要对程序进行全状态、全速验证时或者程序已经准备好进行正式生产活动时，可切换到自动模式。

可以通过以下方式，从手动模式切至自动模式。

方式一	HMI 按钮：点击 HMI 界面上的工作模式图标，从手动模式切换至自动模式。切换前，会弹框确认，点击"确定"，系统将切换工作模式；点击"取消"，将不进行切换。 
方式二	系统 IO："自动模式"。
方式三	外部通信："switch_mode:auto"+"\r"。
方式四	寄存器功能码：ctrl_switch_operation_auto_manual。
方式五	SDK。

危险

当处于自动模式时，机器人可能在无任何警告的情况下由外部信号触发运动。

在切换到自动模式之前，请确认碰撞检测功能已开启，防止机器人与工作人员发生意外碰撞出现人身伤害！

7.2.3.2 自动切至手动

可以通过以下方式，从自动模式切至手动模式。

方式一	HMI 按钮：点击 HMI 界面上的工作模式图标，从自动模式切换至手动模式。切换前，会弹框确认，点击"确定"，系统将切换工作模式；点击"取消"，将不进行切换。 
方式二	系统 IO："手动模式"。

方式三	外部通信：“switch_mode:manual”+“\r”。
方式四	寄存器功能码：ctrl_switch_operation_auto_manual。
方式五	SDK。

7.3 上下电

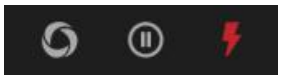
请先阅读第 2 章-安全装置章节，关于使能装置的介绍。

7.3.1 上电

手动模式下，用户通过按住手动使能手柄上的黄色三位使能开关保持在中间位置给电机上电。可以通过听机器人上电声音或者观察 HMI 软件界面上的底部状态栏上电按钮变成红判断是否上电成功。

> 提示
>
> 若上电失败，观察实时日志判断机器人此时状态，调整至支持上电的状态再上电。

自动模式下，点击 HMI 软件底部状态栏的上电按钮给电机上电，判断电机是否正常上电方法与手动模式相同。



7.3.2 下电

手动模式下，用户通过松开或者用力按压黄色三位使能开关使其保持在位置 1 或位置 3 给电机下电。

自动模式下，点击 RobotAssist 软件底部状态栏的上下电按钮给电机下电。

> 警告
>
> 紧急情况下，按下手动使能上的急停按钮给机器人紧急断电，需要再次上电时，请先手动复位急停开关。

7.4 移动控制

7.4.1 Jog 点动

Jog 点动支持多种坐标系/模式，参考下表。

	说明	备注
世界坐标系	笛卡尔空间 Jog，沿给定的坐标系方向进行运动。例如选择“世界坐标系”,Jog X，将沿世界坐标系 X 向平移运动；选择”工具坐标系”，Jog B，将沿工具坐标系的 Y 向旋转。	
法兰坐标系		
基坐标系		
工具坐标系		
工件坐标系		
奇异规避	主要用于笛卡尔 Jog 时规避手腕奇异点，其 XYZ 运动都是相对于基坐标系运动的。 需要先将 4 轴 Jog 到 0 或者±180 后才能进行笛卡尔 XYZ 的 Jog，使用该模式下的 J4 可以快速 Jog4 轴到上述角度。在此基础上，Jog XYZ，4 轴角度将被锁定，不再发生变化，机器人的姿态将会随着臂平面转动。Jog Ry 机器人法兰会绕机器人大小臂构成的平面的法线方向转动(只有当	需要先在”设置”-“运动参数”-“高级设置”页面开启“码垛调试模式”，目前版本适用机型：工业标准六轴系列(XB,NB 型

	4 轴角度为 0 或者±180 时才能 Jog Ry)。Jog J6 与关节空间 Jog6 轴相同，只调整 6 轴。 CR 系列 5 轴机型的基坐标系 jog，对应 6 轴协作机型的奇异规避模式，即 Jog X/Y/Z 时，机器人的姿态将会随着臂平面转动；Jog Ry 机器人法兰会绕臂平面法线转动；Jog J5 与关节空间 Jog 5 轴相同。 注意，开启全 DH 补偿时，无法实现奇异规避，最好在非全 DH 补偿下使用。	号)和协作 xMate CR/SR 系列。
平行基座	主要用于笛卡尔 Jog 时规避手腕奇异点，其 XYZ 运动都是相对于基坐标系运动的。 需要先将法兰 Jog 至基座平行的状态才能进行笛卡尔 XYZ 的 Jog，使用该模式下的 J4 和 Ry 可以快速使法兰到达与基座平行的状态。Jog XYZ 时，机器人的姿态不发生改变，但是可以自动规避机器人腕奇异点。J4，Ry，J6 的使用方式与奇异规避模式一致。 CR 系列 5 轴机型使用该模式下的 Ry 可快速使法兰到达与基座平行的状态。 注意，开启全 DH 补偿时，无法实现 jogZ 方向时的奇异规避。	
轴空间	单独控制各个轴运动。	

提示

xMate 系列的 5 轴机型由于构型限制，在普通坐标系下 Jog xyz 时，机器人的姿态将会随着臂平面转动；Jog A 时机器人法兰会绕臂平面法线转动；Jog B 与关节空间 Jog 5 轴相同；C 按钮为无效。

Jog 点动支持连续和步进两种模式：

- 连续运动模式下，机器人上电后，按住 Jog 按键，机器人在设定 Jog 速度下连续运动，直至松开使能或者 Jog 按钮。
- 增量步进模式下，机器人上电后，每按一次 Jog 按键，机器人运动给定的步长；用户可根据需要选择合适的步长大小，主要用来精确调整机器人位姿。

提示

增量步进模式下 ，需长按 Jog 按钮，等待机器人运行指定步长后方可松开 Jog 按钮，点按操作将可能使机器人提前停下来。

Jog 速度设置：

用来控制 Jog 时机器人的运动速度，可选范围 0.1%~100%，100%时对应机器人最大 TCP 速度 250 mm/s。受安全法规限制，笛卡尔空间 Jog 和轴空间 Jog，TCP 线速度均不会超过 250mm/s。

7.4.1.1 Jog 低速模式

为了方便用户精细调节运动点位，连续 Jog 模式下，如果 Jog 速度被配置为 1%及以下时，机器人会进入 Jog 低速模式。

该模式下，机器人的运行速度将会被锁定为 20%，而单次点击 Jog 的步长将会按照下表进行。

Jog 速度	轴空间 JOG	笛卡尔 JOG
1%	1.0°	1.0 mm
0.5%	0.5°	0.5 mm
0.1%	0.1°	0.1 mm

7.4.2 快速调整

HMI 右侧操控面板提供了快速位姿调整功能，支持的快速位姿调整包括：初始位姿、拖动位姿、发货位姿及 Home 位姿。

快速位姿调整在手动模式下使用，使用方式与 Jog 操作类似，手动模式下通过使能将机器人上电，按下相应目标位姿按钮，机器人将通过轴空间运动至目标位姿。运动过程的速度可通过 Jog 速度调整。

拖动位姿、发货位姿、Home 位姿支持用户自定义，可在"设置"-"快速调整"页面进行设置。如未设置，则使用默认配置。

7.4.3 拖动

在点位示教时，通过拖动定位可以大幅缩短编程时间；机器人拖动，结合轨迹录制、轨迹复现，可以简化部分连续轨迹场景示教的难度。

关于协作机器人各机型末端把手的介绍和使用示例，请参考附录。

注意：拖动及其扩展功能（末端把手、点位示教、连续轨迹示教、轨迹复现）仅适用于 xMate 系列协作机器人。

拖动方式	模式	说明
轴空间		各轴运动独立，直接调整各轴位置到期望位姿；
笛卡尔空间	仅平移	可引导机器人沿笛卡尔空间各方向做平移运动；
	仅旋转	可直接手动引导机器人绕 TCP 做姿态旋转；
	自由拖动	支持笛卡尔空间平移及旋转；

手动模式下电状态下，在操作面板打开拖动使能开关，机器人自动上电并启用拖动模式，用手同时按下末端拖动把手使能按钮，即可拖动机器人进行点位示教、轨迹录制。

提示

请在启用拖动模式前，设置合适的拖动方式、拖动空间。启用拖动模式后，机器人自动上电，此时无法设置拖动方式、拖动空间，请关闭拖动后进行设置。

警告

在启用拖动模式前，必须准确设置机器人的动力学参数以及负载参数，否则拖动模式可能打开失败，拖动过程可能存在飘动感。

可使用系统提供的动力学参数辨识功能以及负载辨识功能来设置参数。

危险

在使用拖动示教之前必须要确保以下参数已经正确设置，否则各轴角度处于错误状态时，控制器无法计算正确的输出力矩，机器人拖动功能不能正常使用。

- 机器人的型号。
- 机器人安装方式，地面安装或者吊装。
- 本体和负载的动力学参数。
- 机械零点标定。

7.5 连续轨迹回放

连续轨迹示教成功后，可在录制界面回放轨迹进行确认，确认后手动保存录制轨迹。

回放模式可勾选循环模式，回放时循环回放。

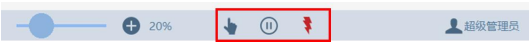
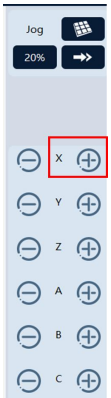
回放速率可设置为1%至300%。建议用户回放速率设置范围1%至100%。当回放速率大于100%，有可能产生驱动器跟随错误。

操作示例，可以参考下文“编程”-“路径列表”章节。

7.6 操作示例一：工业机器人实现 Jog 运动

在进行实机操作前，需确保 HMI 与机器人连接、通电开机正常（详见第五章）。

步骤	图示	说明
1.检查机器人状态。确保机器人处于正常连接且手动操作模式；		
2.切换用户。		

3.选择【基坐标系】运动坐标系。具体坐标系详见章节 4。并选择【连续运动】。		机械臂末端运动方向与基坐标系方向相同。
4.握住示教器的使能开关。		手动上电后成功时，会有明显上电声音，机器人状态栏会显示机器人处于手动上电状态。点动期间，需保持使能开关处于按下状态。
5.机器人点动。点击示教器需要移动的目标方向。此示例演示沿基坐标系 X 正方向移动。		连续点动模式下，需保证运动按钮处于按下状态，松开运动按钮，机器人将停止运动。
6.关闭点动。松开机器人运动按钮，并松开使能按钮。		手动下电后，机器人状态栏将处于手动下电状态。

7.7 操作示例二：CR 协作机器人实现拖动

在进行实机操作前，需确保 HMI 与机器人连接、通电开机正常（详见第五章）。

本节主要基于 xMate CR7 协作机器人，演示协作机器人拖动功能。

步骤	图示	说明
1.检查机器人状态。确保机器人处于正常连接且手动操作模式；		尽量使机器人处于拖动位姿。如机器人处于自动模式，拖动按钮将处于不可选取状态。
2.切换用户。		
3.点击【拖动】按钮。拖动按钮会显示拖动打开状态，机器人状态栏会变为拖动状态，机器人会出现明显上电声音；		
4.按住机器人的拖动使能开关，将机器人拖动至机器人工作空间内的任意位置；		拖动期间，按住拖动按钮进行拖动。

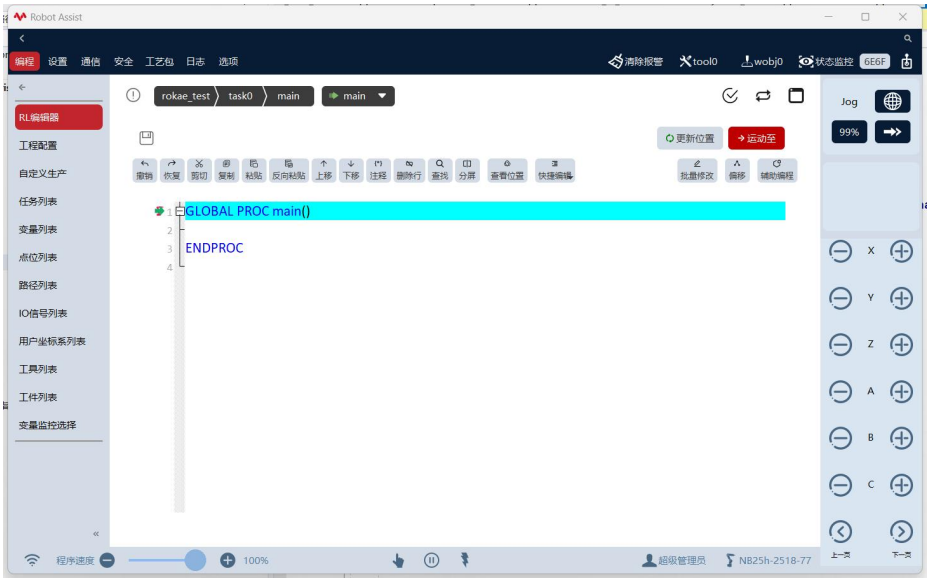
5.松开机器人拖动使能开关，点击【拖动】按钮。拖动按钮会显示拖动关闭状态，机器人状态栏会变为手动下电状态，机器人会出现明显下电声音。		退出拖动状态后方可允许进行其他操作。

8 编程

8.1 本章简介

工业/协作机器人是一种非常柔性的生产工具，用户可以通过指令编程，应对不同的需求。本章将介绍 xCore 控制系统编程的方方面面。

从本章开始，用户将逐步深入了解 xCore 编程、设置、通信等高阶使用方式。



8.2 初识工程

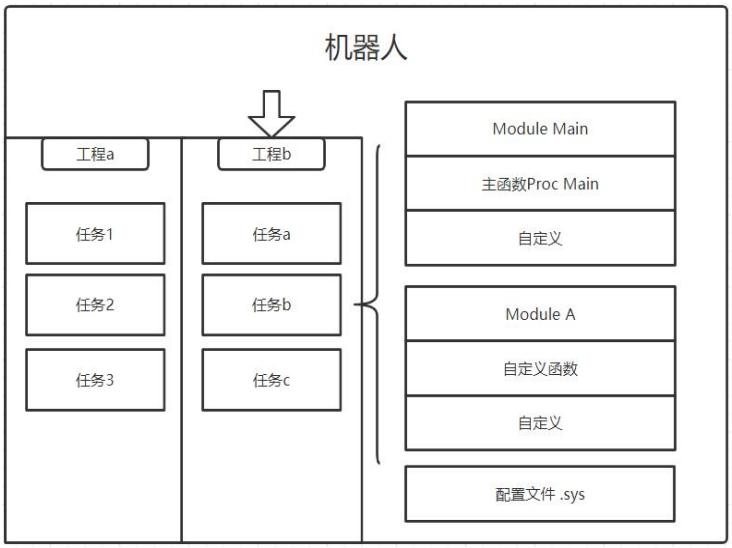
xCore 以工程为单位，管理用户的编程。一个典型的工程，包含了 RL 程序、自定义用户界面、任务、变量、点位、路径、IO、用户坐标系、工具坐标系、工件坐标系等信息。

根据范围大小分为四个层次：

工程	Project	最高级别，包含各类信息，一个工程中可以包含多个任务，原则上每个任务互相独立，仅靠提供的接口进行交互； 机器人一次只能选择一个工程执行；
任务	Task	一个任务中可以包含多个程序模块，但有且仅有 1 个 main.mod；在 main.mod 中包含一个 GLOBAL PROC main，该 GLOBAL PROC main 作为整个工程的入口函数； 加载并执行某个工程，本质上是在执行 main 函数；
程序模块	Module	分为程序模块（.mod）与系统模块（.sys）两种，一个模块就是一个程序文件； 每个程序模块中都包含了若干数据变量以及函数，用于实现具体的机器人功能； 每一个程序文件都可以执行拷贝、删除等常规的文件操作； 模块的定义方式为： PROC main() … ENDPROC PROC test1()

		… ENDPROC PROC test2() … ENDPROC
函数	Routine	用户可在函数内按自己的需求调用机器人功能或者其他模块；支持用户自定义函数，不同的自定义函数可以存储在同一个程序文件中，也可以存储在不同的程序文件中；

工程、任务、程序模块以及函数的相互关系，如下图所示：



提示

每一个模块中，位于文件头部与第一个函数之前的代码区称为声明区，用来存放 GLOBAL 和 LOCAL 作用域的变量声明。该区域中定义的变量在每次 pptomain 时将重新初始化。

例：int0 在变量列表中定义，初始值为 0；int1 在声明区中定义，初始值为 0。

循环运行时，int0 每次运行后递增 1，其打印值为"1，2，3……"；int1 每次运行时将重新初始化为 0，其打印值为"1，1，1……"。

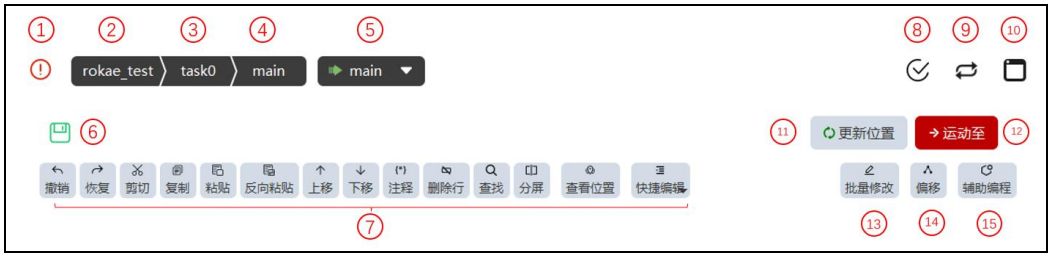
```
//声明区
int int1 = 0;
//
GLOBAL PROC main()
    int0++;
    int1++;
    print("int0=",int0);//int0在变量列表中定义
    print("int1=",int1);//int1在声明区中定义
    pause;
ENDPROC
```

8.3 RL 编辑器

8.3.1 概览

①	工程、任务、程序文件选择、切换和编辑。
②	程序调试快速定位按钮，支持 （1）快速定位至 main 函数;（2）快速定位至光标；（3）快速定位至当前程序模块的某函数。
③	保存当前任务。
④	程序编辑工具栏：撤销、重复、剪切、粘贴、复制、反向粘贴、上移一行、下移一行、批量注释、删除当前行、查找、查看位置、快捷编辑、批量修改、偏移、辅助编程。
⑤	更新点位的位置信息、运动至某点位、程序语法检查、循环模式、输出终端。
⑥	程序编辑区，通过辅助编程等方式，进行 RL 命令编辑。

8.3.2 工具介绍



①	工程同步标识。 红色时：表示控制器内加载程序与 HMI 当前程序不同步。 灰色时：控制器内加载程序与 HMI 当前程序已同步。
②	当前工程：点击可切换至工程配置页面。
③	当前任务：点击可切换任务。
④	当前程序模块：点击可切换程序模块。
⑤	指针移点击下拉夹头可选择指针移动到的目标，可移动程序指针到 Main 函数（相当于程序复位）、程序指针移动到光标所在行、程序指针移动到某函数。
⑥	显示保存状态，点击该按钮可立即保存到本地并推送至控制器。 显示绿色时：程序已经编辑，但是未同步到控制器。 显示灰色时：程序已经同步到控制器。
⑦	对上一步操作进行撤销。
	对上一步撤销的操作进行恢复。
	剪切选中的代码行，支持多行同时剪切。
	复制选中的代码行，支持多行同时复制。
	将剪切或复制到的整行内容，插入光标所在行。
	将复制的代码行，进行反序粘贴；例如：复制第 1、2、3 行，粘贴后显示第 3、2、1 行。
	将选中的代码行上一行，支持多行同时上移。

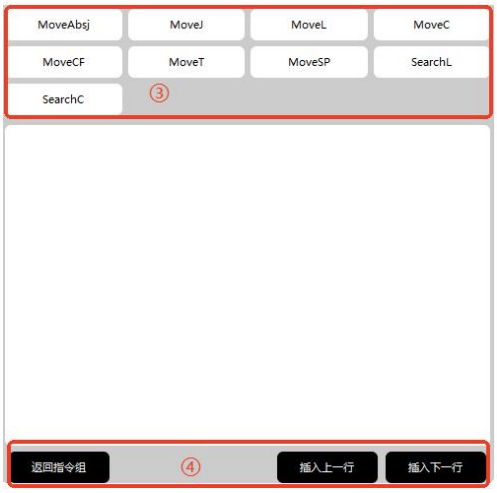
	将选中的代码行下一行，支持多行同时下移。
	注释或取消注释选中的代码。
	删除选中行或光标所在行。
	查找关键字。
	查看位置，详细说明见下文"查看位置功能"
	分屏，详细说明见下文"分屏功能"
	快捷编辑-一键排版：对当前工程所有程序进行排版（如，添加缩进、操作符对齐）。
	快捷编辑-一键展开：将当前 RL 编辑器加载程序展开。
	快捷编辑-一键折叠：将当前 RL 编辑器加载程序折叠。
⑧	检查当前程序中是否存在特定明显错误，例如函数重名、关键标识符缺失。不能检查出所有语法错误。
⑨	切换循环模式：程序循环执行，程序只运行一次。
⑩	显示打 Print 打印信息和语法信息。
⑪	更新光标所在行的点位位姿为当前位姿。注意：光标位于点位所在行即可。
⑫	运动至功能，详细说明见下文"运动至功能"。
⑬	批量修改：对速度、转弯区、工具、工件进行批量修改。
⑭	点位偏移工具，详细说明见下文"点位偏移工具"。
⑮	辅助编程工具。

8.3.3 辅助编程

辅助编程面板可以辅助编程员快速搭建程序框架、插入程序指令、更改指令属性配置。辅助编程面板分为两个部分内容：插入指令和属性配置。

8.3.3.1 插入指令

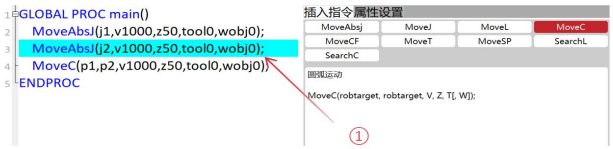
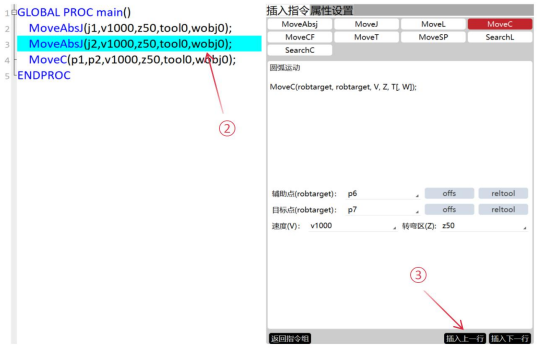
插入指令负责在程序文本中插入期望指令。



序号	说明
①	程序指令组别选择区；
②	程序常用元素选择区；
③	程序指令/元素选择区，选择细分指令；
④	指令/选择插入确认区，用于将选中的指令默认参数或元素插入到工程文本中；

8.3.3.1.1 示例一，插入 MoveL 指令

步骤	图示	说明
1.点击"辅助编程" -> "插入指令"按钮，打开插入指令面板；		打开插入指令面板。
2. 选择"运动指令" -> "MoveL"；		选择"运动指令"后，其他指令会被隐藏，选择"MoveL"指令。
3. 将光标选中期望插入行的上一行；		选择期望插入行。

		
4. 点击"插入下一行"按钮；		指令插入到"3"中选中行的下一行。

8.3.3.1.2 示例二，插入函数

步骤	图示	说明
1.点击"辅助编程" -> "插入指令"按钮，打开插入指令面板。		打开插入指令面板。
2. 选择"函数"。		

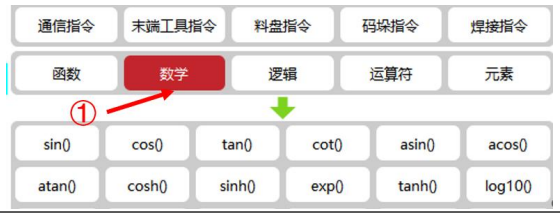
3. 在“名称”下方的输入框中，填入期望函数名称。		
4. 点击“插入”按钮。		点击“插入”按钮后，将在文本末尾创建对应函数的文本。

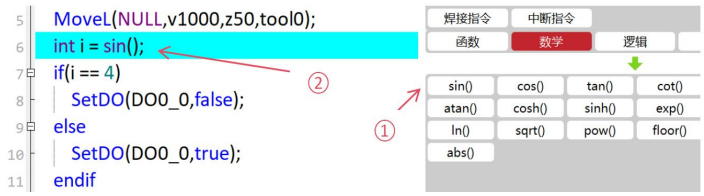
8.3.3.1.3 示例三，插入元素

步骤	图示	说明
1.点击“辅助编程” -> “插入指令”按钮，打开插入指令面板；		打开插入指令面板
2. 选择“元素”；		
3. 在“元素”面板中选择期望插入的变量等信息；		

		
4.将光标移动至期望插入的文本位置；		
5. 点击"插入"按钮；		点击"插入"按钮后，将在光标对应位置生成文本

8.3.3.1.4 示例四，插入数学/逻辑/运算符

步骤	图示	说明
1.点击"辅助编程" -> "插入指令"按钮，打开插入指令面板；		打开插入指令面板
2. 选择"数学"；		

3. 将光标移动至期望插入的文本位置；		
4. 点击"sin()"按钮；		点击对应按钮后，将在光标对应位置生成文本 sin()；

8.3.3.2 属性配置

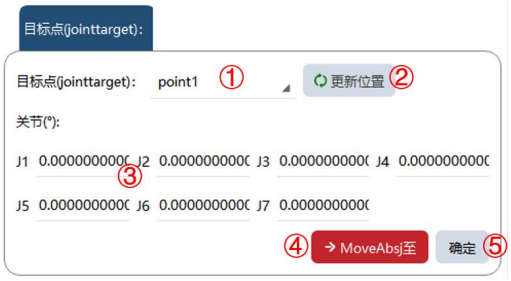
属性配置负责更新选中行的指令参数信息。

序号	说明
①	属性定义区，介绍当前指令的参数名称
②	属性选择区，修改选中指令的各项参数
③	属性确认区，点击"替换"按钮，替换程序文本中当前行得指令参数

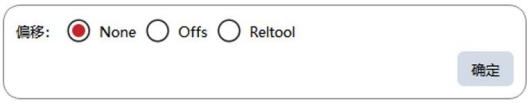
8.3.3.2.1 示例一，配置 SetDO 指令属性

步骤	图示	说明
1 在文本中选中期望修改的指令所在行；		打开属性设置面本，指令详情将在面板中展开
2. 在"属性设置"面板选择期望参数；		
3. 点击"替换"按钮		点击"替换"按钮后，选中行的指令参数将更新为设置后的参数

8.3.3.2.2 示例二，配置 MoveAbsJ 指令属性

序号	图示	说明
1. 运动参数设置；		设置 MoveAbsJ 指令的速度、转弯区、工具、工件参数。
2. 点位设置；		1.选择点位； 2.更新选中点位的位姿信息为当前机器人位姿； 3.可手动修改点位信息； 4.手动 JOG 机器人到选中点位； 5.点击"确定"按钮可将 3 中修改的数据更新到点位数据中；

8.3.3.2.3 示例三，配置 MoveL 指令属性

序号	图示	说明
1. 运动参数设置；		设置 MoveL 指令的速度、转弯区、工具、工件参数；
2. 点位设置；		1.选择点位； 2.更新选中点位的位姿信息为当前机器人位姿； 3.可手动修改点位信息； 4.手动 JOG 机器人到选中点位； 5.点击"确定"按钮可将 3 中修改的数据更新到点位数据中；
3. 偏移设置		
4. 偏移后点位调试面板。		该面板不可编辑，用于 JOG 设置偏移后。

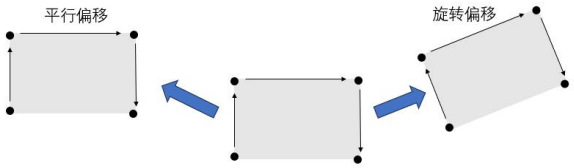
8.3.4 点位偏移工具

8.3.4.1 概述

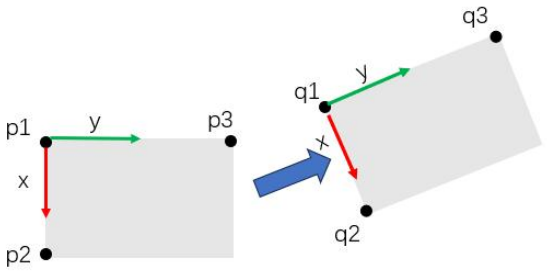
本工具，可以将 RL 程序中用户选定范围内的运动指令，对已示教完成的点位进行整体偏移修改。当前版本控制器支持三种偏移方式，分别是：程序偏移、角度偏移和镜像偏移。

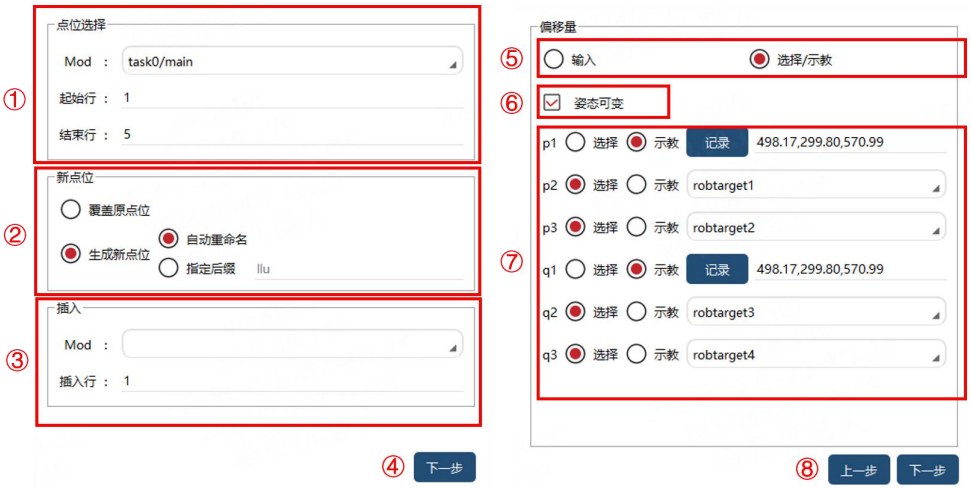
8.3.4.2 程序偏移参数介绍

本工具可以将 RL 程序中用户设定范围内的运动指令，对已示教完成的点位进行整体偏移。需要工件仅平移时，使用平行偏移，只有调整 xyz 数值；当希望工件位姿都调整时，打开"姿态可变"，姿态数值 ABC 也会调整。



原理：通过 p1、p2、p3，q1、q2、q3 共计 6 个点，计算原始坐标系和偏移后坐标系之间的关系。当不打开"位姿可变"时，仅需要设置 p1、q1。





①	点位选择：选择需要偏移的点所在的位置，指定 task 与 mod、起始行、结束行，本工具将搜索从起始行到结束行的所有指令中涉及的点位。 注意：用"//"注释掉的指令不被搜索。
②	新点位生成方式：可选择"覆盖原点位"或"生成新点位"。 如果选择"生成新点位"，可选择新点位命名方式："自动重命名"或"原点位名+指定后缀"。
③	插入位置：指定新生成的指令插入的位置，选择 task 与 mod，生成的指令文将从插入行生成。

④	点击"下一步"按钮，进入偏移量设置页面。
⑤	偏移矩阵生成方式：可选择"输入"或"选择/示教"，仅在后者中支持"位姿可变"。
⑥	选择位姿是否可变，若不可变即仅考虑平移。
⑦	在"选择/示教"条件下，指定用于计算位姿的 2 个点或 6 个点，每个点可以选择"选择"，则在下拉列表中选择已存在的点位；也可选择"示教"，点击"记录"按钮，记录机器人当前位置。
⑧	点击"上一步"按钮，进入点位选择页面；点击"下一步"按钮，开始计算偏移点位。

使用限制：

1、建议每次偏移点位数量 1000 以内，否则可能造成界面卡死、点位丢失。

2、系统在计算偏移点位时，会自动校核点位是否合理，若不合理会报错，同时生成指令失败；但在不合理点位之前的点位仍会生成。

点位不合理情况：①参数错误 ②奇异位置 ③点位不可达 ④其他错误。

3、若指定 mod 的起始行到结束行之间，没有点位，则报错"未找到待偏移点"。

4、使用本工具需要连接机器人，否则报错"机器人通信错误"。

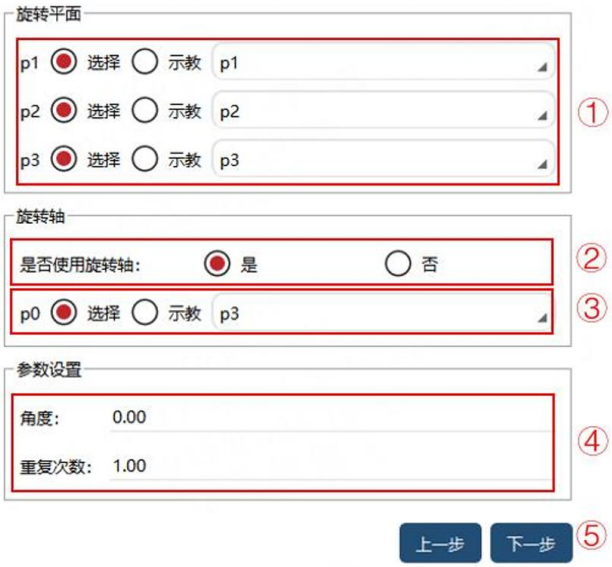
5、p1、p2、p3 两辆组成的向量之间夹角应大于 1°，否则将判定三点共线，报错"参考点错误"；q1、q2、q3 同理。

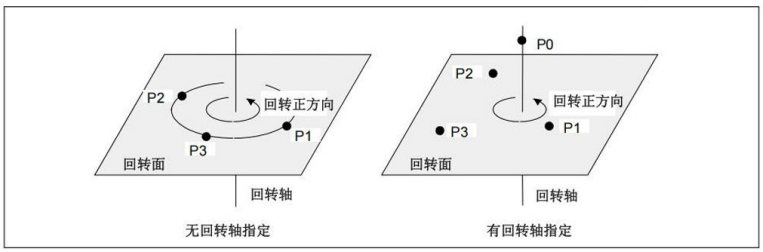
6、指令中的点位名字需要与点位列表中一致（不区分大小写），否则报错"未找到点 xxx"。

7、本工具页面打开情况下，在"任务列表"、"点位列表"、RL 编辑页面进行增减任务、增减点位、编辑 RL、切换工程等操作，本工具不会自动刷新，需要关闭本工具页面再打开。

8.3.4.3 角度偏移参数介绍

该功能提供笛卡尔点位的角度偏移功能，可以将 RL 程序中用户设定范围内的运动指令，对已示教完成的点位按照指定角度进行旋转偏移。





指令点位选择和程序偏移相同，选择生效范围后，按照下面操作可以完成角度偏移。

①	确定旋转平面：用户需要指定 P1，P2，P3 三个点位来确定角度偏移功能的基准旋转平面，这个三个点可以选择点位列表中的点位或者通过示教的方式确认。
②	禁用旋转轴： 将【是否使用旋转轴】设为"否"，系统将以指定的 3 个代表点 P1、P2、P3 计算得到一个圆，三点都在圆上，旋转 轴为通过圆心并垂直于圆面的轴，旋转方向为 p1 指向 p2 的方向。
③	启用旋转轴： 将【是否使用旋转轴】设为"是"，系统将以指定的 3 个代表点 P1、P2、P3 计算得到旋转面，再指定代表点 P0， 通过 P0 并垂直于旋转面的轴即为旋转轴，旋转方向为 p1 指向 p2 的方向。
④	设置偏移的角度以及重复次数，该功能对点位进行多次旋转，每进行一次重复，会生成一组运动指令。

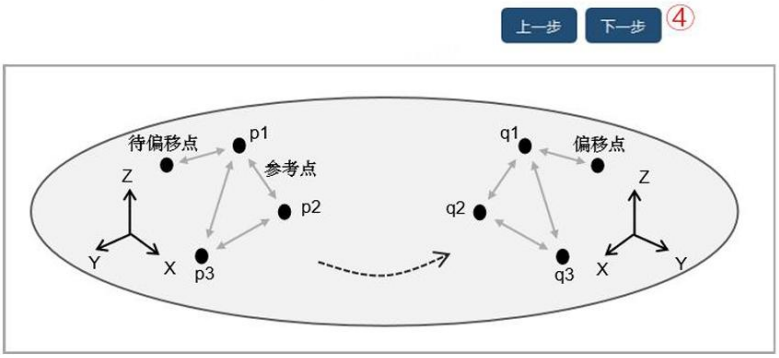
注意：

1、此功能不支持关节点位的角度输入偏移

2、执行角度输入偏移后，新 RL 程序中机器人点位的姿态可能与原程序不一致，请手动慢速单步、单点执行确认后再连续运行程序

8.3.4.4 镜像偏移参数介绍

该功能提供笛卡尔点位的镜像偏移功能，用户可以选择需要进行镜像偏移的点位，按照一定规则对点位进行镜像偏移。

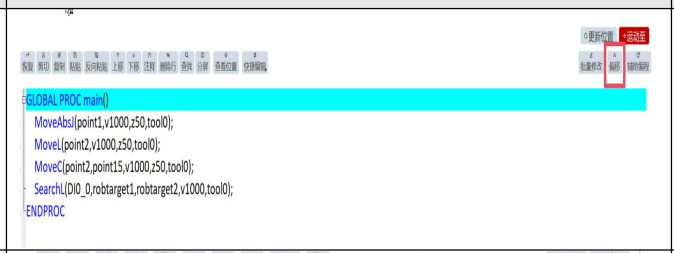
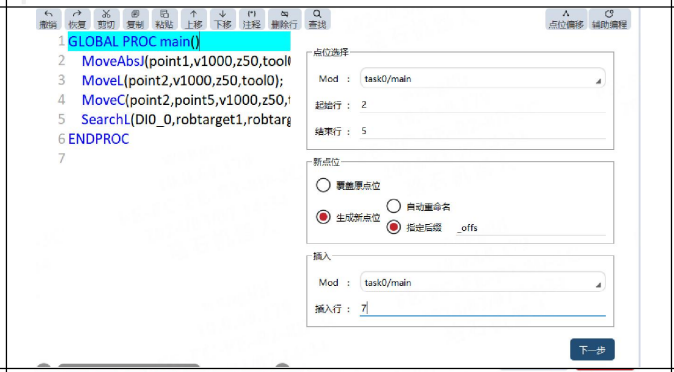
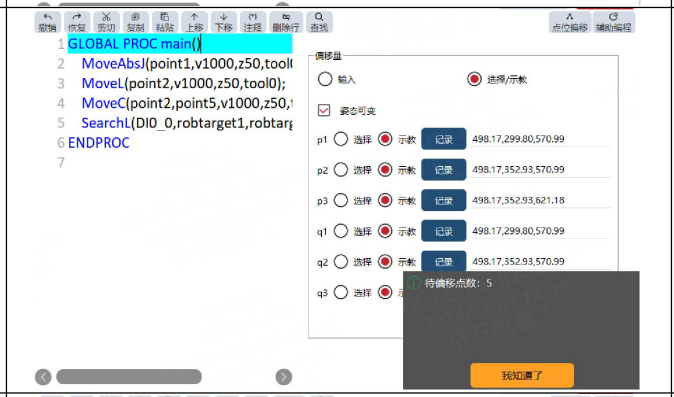
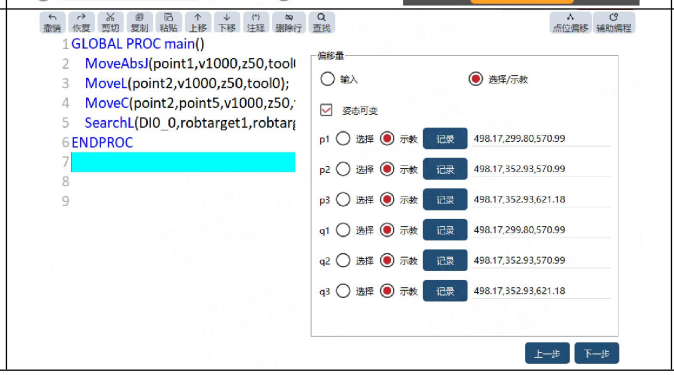


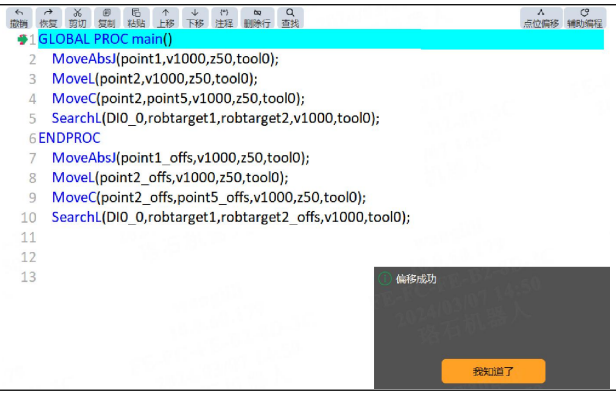
指令点位选择和程序偏移相同，选择生效范围后，按照下面操作可以完成镜像偏移。

①	选择镜像点： 选择镜像点 p1,q1 作为偏移的参考点，可以使用点位列表点位或者示教。
②	不使用旋转：

	将【是否使用旋转】设为“否”，系统将以 p1、q1 的位置计算镜像平面，待偏移点位的位置将以此平面进行镜像偏移，偏移前后点位的姿态不变。
③	使用用旋转： 将【是否使用旋转】设为“是”，需要额外确定参考点 p2、q2、p3、q3，其中（p1、p2、p3）形成的三角形与（q1、q2、q3）形成的三角形需要沿着预想的镜像平面对称，此时待偏移点位的位置和姿态将以此平面进行镜像偏移，偏移前后的姿态也是对称的。

8.3.4.5 操作示例

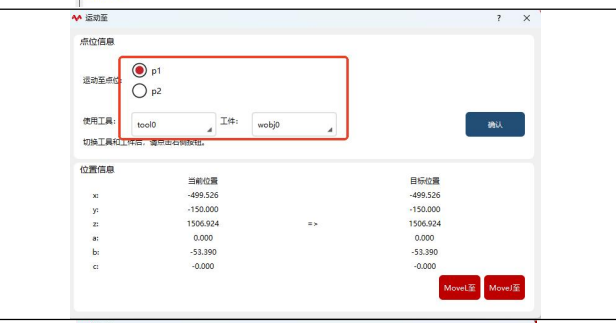
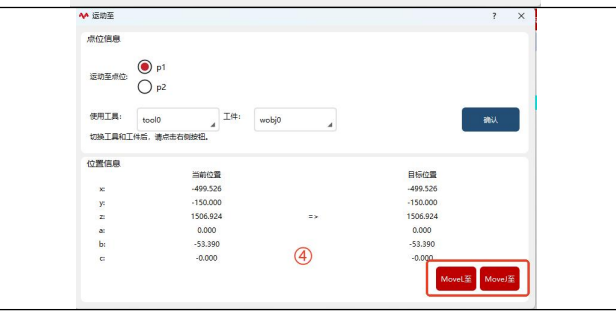
步骤	图示	说明
1.点击“点位偏移”按钮，打开该工具；		仅在“编程员”及以上权限、手动模式下可使用；
2.选择待偏移点位，设置参数后，点击“下一步”按钮；		这里选择 task0 任务中的 main 中的第 2~5 行的点位； 点位生成方式为“生成新的点位”，并指定新点位的名字后缀； 插入的位置选择该 mod 文件的第 7 行；
3.提示“待偏移点位数量”；		这里的待偏移待偏移点位分别是：第 2 行中的 point1、第 3 行的 point2、第 4 行的 point2 和 point5、第 5 行的 robtarget2。 注意： 1.其中“point2”被计算了两次，但实际上只会偏移一次，也只生成一个偏移后的点位。
4.设置偏移量计算方式、参考点位，点击“下一步”按钮；		正在执行偏移过程中弹出进度条，请耐心等待； 

5.提示“偏移成功”，第 7~10 行为新生成的指令；		在点位列表中可以看到生成的点位信息； 

8.3.5 运动至功能

运动至功能，是将机器人通过一系列操作后，运动到选择的点位；

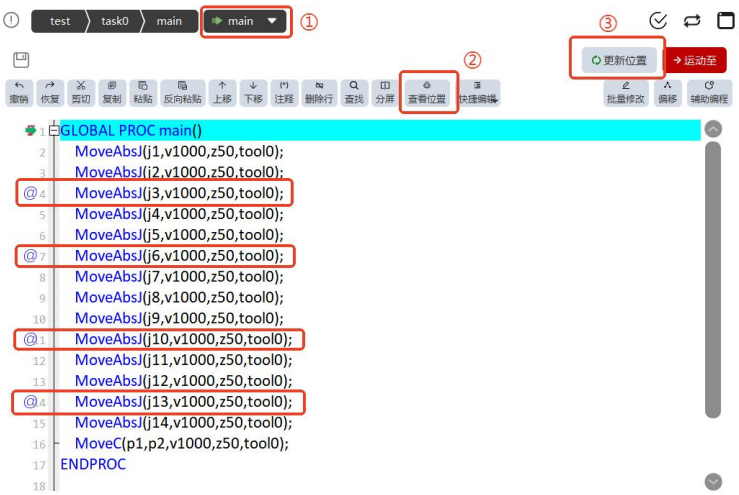
8.3.5.1 操作示例

步骤	图示	说明
1.在 RL 编辑器界面选择对应的运动指令；		仅在“编程员”及以上权限、手动模式下可使用；
2.点击右上角的“运动至”按钮；		
3.在运动至弹窗上，选择对应的工具工件和点位		单点位运动指令，无需选择点位； 多点位运动指令，需要确认选择的点位； 选择点位运动至的工具工件；
4.在运动至弹窗上，选择运动当时，并执行运动至功能.		选择右下角的“MoveJ”或者“MoveL”或者“MoveAbsJ”（仅轴角度点位显示），长按触发运动至功能。

8.3.6 查看位置功能

8.3.6.1 概述

查看位置功能，主要是为了方便用户知道当前机器人停止位置，处于 RL 程序的哪一行。



触发查看位置功能，有以下几种方式：

1. 点击”PPToMain”按钮时；
2. 点击“查看位置”按钮时；
3. 点击“更新位置”按钮时；

当机器人查看位置功能出发后，会在 RL 程序界面的某一行或者多行，显示一个 @ 符号，表示机器人所在的位置和符号所在行所使用的点位位置一致。

8.3.6.2 查看位置参数设置

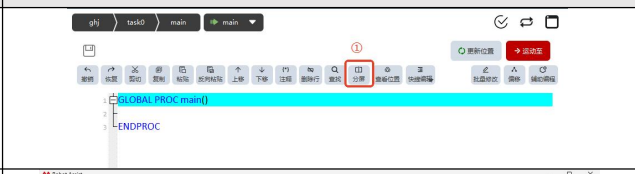
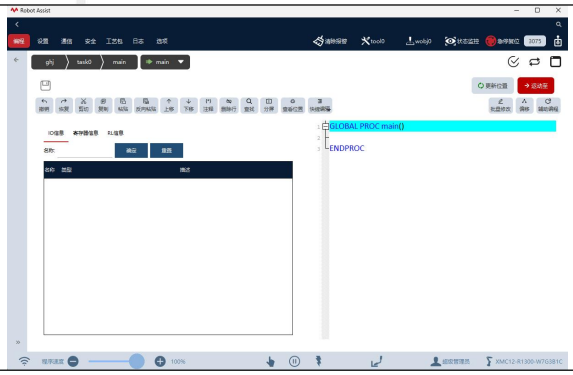
参考“设置”->”HMI 设置”里面的“查看位置参数”设置，用来设置机器人实际位置和 RL 程序点位位置之间的误差范围值；

8.3.7 分屏功能

8.3.7.1 概述

分屏功能，主要是为了方便用户在编辑 RL 程序时，快速查看 IO、寄存器和其他函数界面程序；

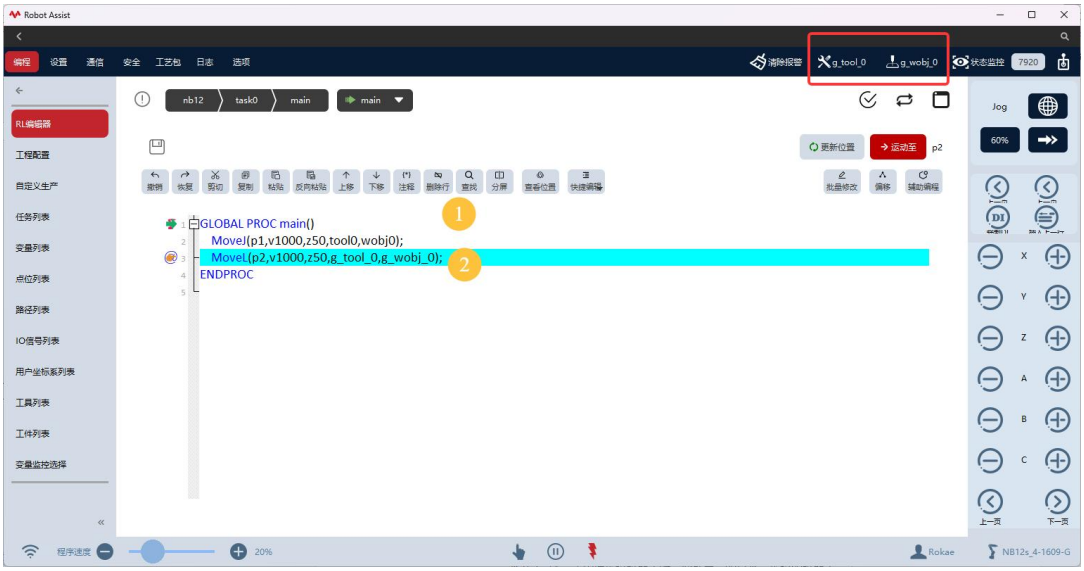
8.3.7.2 操作步骤

步骤	图示	说明
1.点击顶部操作栏“分屏”按钮；		仅在“编程员”及以上权限、手动模式下可使用；
2.显示左侧分屏界面；		分屏时，右侧操作面板和左侧菜单面板自动关闭； 关闭分屏，两侧面板自动展开；

3.切换分屏		点击分屏面板上的 tab 栏，可以切换 IO、寄存器和 RL 程序面板； 其中，IO 列表和寄存器列表支持搜索名称功能；

8.3.8 工具工件跟随指针功能

当机器人处于手动模式时，示教器右上角的工具工件组会识别用户选中的 RL 指令行，将当前启用的工具工件自动切换为当前选中指令的工具工件。注意：RL 中自定义的工具工件不支持此功能。

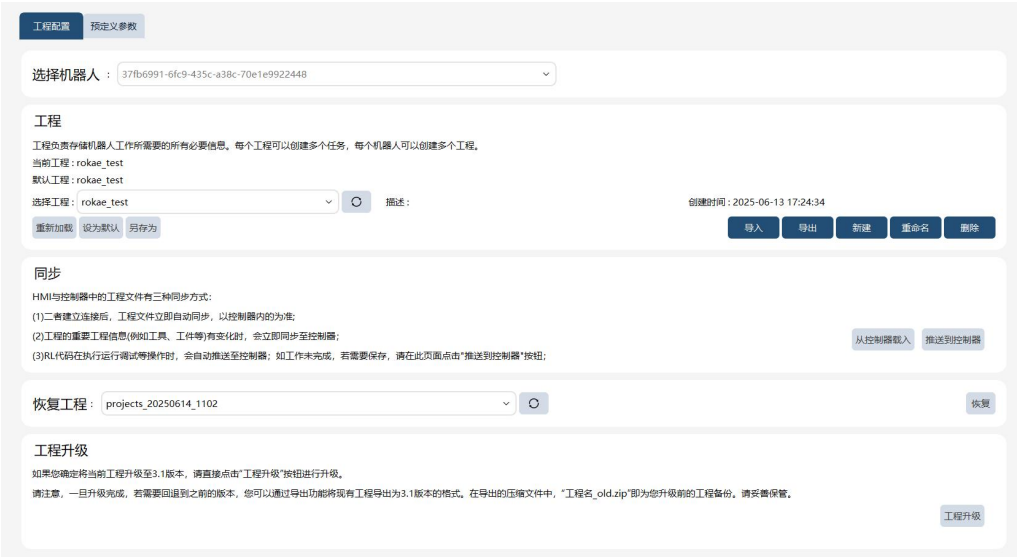


①当选中第二行时，右上角的工具工件会自动变更为对应的 tool0，wobj0

②当选中第三行时，右上角的工具工件会自动变更为对应的 g_tool_0，g_wobj_0

8.4 工程配置

工程配置界面用于对当前工程进行相关配置。



8.4.1 选择机器人

当 PC 端运行 RobotAssist 软件，不连接机器人时，如果想离线编辑 RL 工程，可以在此选择连接过的机器人，系统会切换并显示该机的工程数据。
当 RobotAssist 软件连接机器人时，此处显示的是当前连接的机器人。

8.4.2 工程

重新加载：重新加载选择的工程。
与之前的版本相比，3.1 版本在 RL 编程语言的语法上做了一些调整。主要变化如下：

1. 所有函数和注册指令在调用时，必须加上括号 ()；
2. 每条表达式结束后，需要添加分号 ；

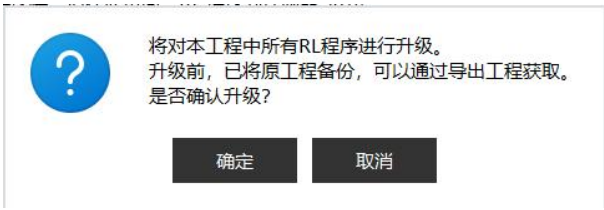
例如，在旧版本中，程序可能像这样书写：

```
MoveL p1,v3000,z50,tool0
int a = 200
ConfL off
```

而在 3.1 版本中，应修改为以下格式：

```
MoveL(p1,v3000,z50,tool0);
int a = 200;
ConfL(off);
```

当您从旧版本升级到 3.1 版本后，在"工程配置"界面中点击重新加载或在"程序编辑界面中点击 PPToMain 按钮时，系统会自动检测并弹出提示，引导您完成工程的升级操作。



请注意，一旦升级完成，若需要回退到之前的版本，您可以通过导出功能将现有工程导出为 3.1 版本的格式，在导出的压缩文件中，"工程名 old1.zip"即为您升级前的工程备份。请务必妥善保存此备份文件，以备后续需要。

设为默认：将选择的工程设为默认工程，当机器人开机后，可自动加载。
另存为：另存为选择的工程，并将其推送到控制器。
导入：打开导入工程页面，选择要导入的工程系统、工程路径等参数。
导出：打开导出工程页面，选择需要导出的工程以及路径，支持导出为 xCorev2.2 及以下的工程格式，xCore3.0 版本的工程格式，xCore3.1 版本的工程格式。
新建：新建工程，工程名称只能是英文字母，数字和下划线的集合。
重命名：重命名当前选择的工程。
删除：删除当前选择的工程。

> ⚠ 警告
> 文件一旦删除，无法恢复！

8.4.3 同步

HMI 与控制器中的工程文件有三种同步方式：

1. 二者建立连接后，工程文件立即自动同步，以控制器内的为准；

2. 工程的重要工程信息(例如工具、工件等)有变化时，会立即同步至控制器；
3. RL 代码在执行运行调试等操作时，会自动推送至控制器；如工作未完成，若需要保存请在此页面点击"推送到控制器"按钮；

“从控制器载入”按钮：示教器将重新从控制器同步工程并加载，此操作将覆盖示教器当前状态下的所有工程。
“推送到控制器”按钮：示教器将当前状态下的工程同步到控制器中。

8.4.4 恢复工程

当工程有修改时，控制器会进行定期备份。在下拉列表中选择一个按时间命名的备份工程，点击”恢复”恢复备份的工程。点击 ，刷新备份工程列表。

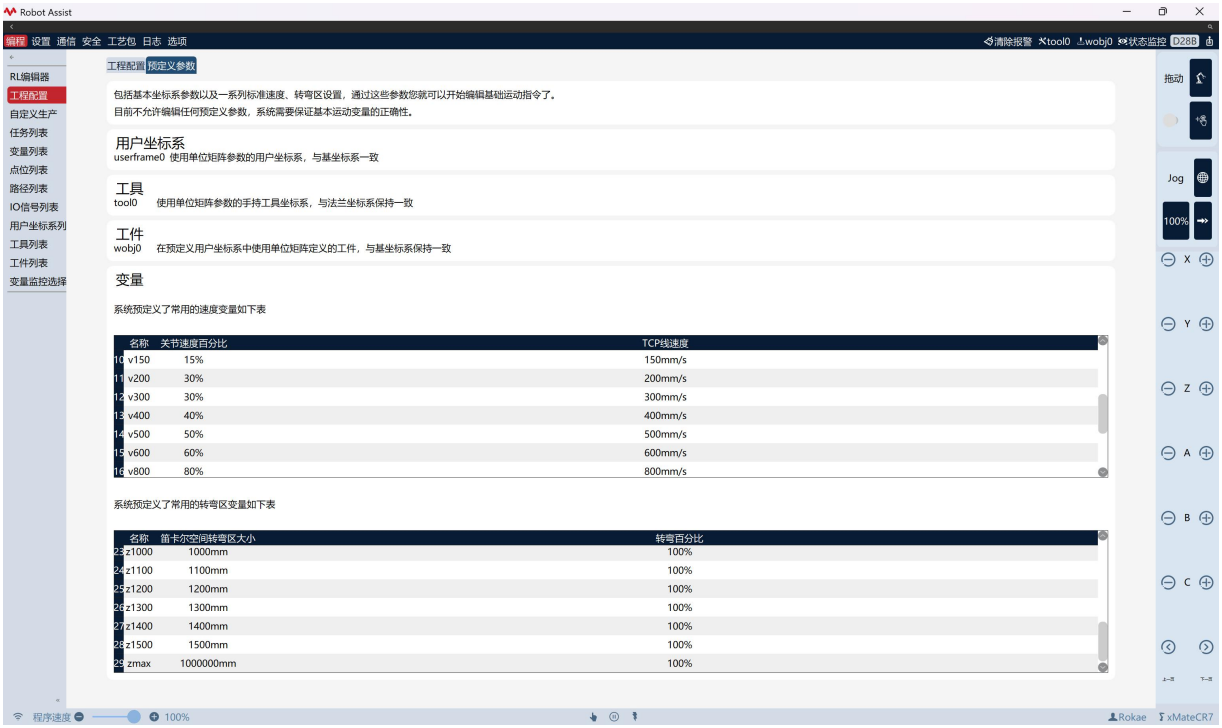
8.4.5 工程升级

通常情况下，您无需手动使用“工程配置”界面中的 工程升级 功能。但在某些特殊场景下，如果无法通过点击重新加载、PPToMain 按钮进行升级，您可以点击该功能按钮手动执行升级操作。

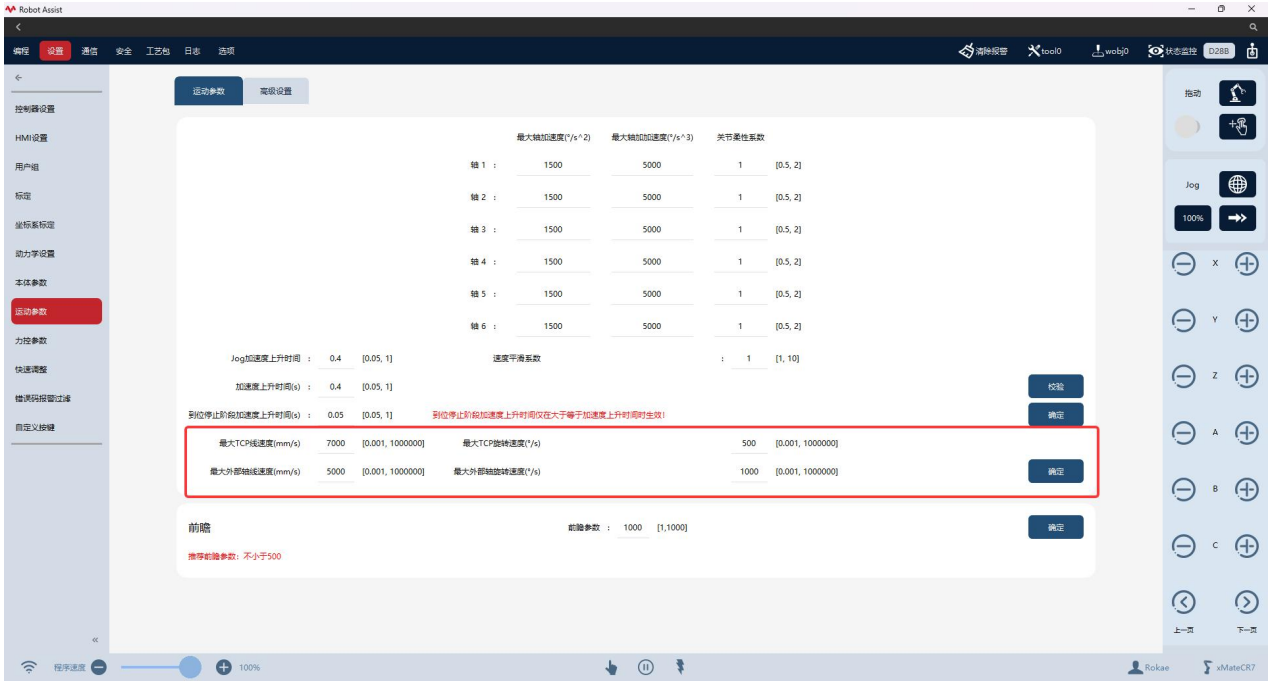
请注意，一旦升级完成，若需要回退到之前的版本，您可以通过导出功能将现有工程导出为 3.1 版本的格式，在导出的压缩文件中，“工程名 old1.zip”即为您升级前的工程备份。请务必妥善保存此备份文件，以备后续需要。

8.4.6 预定义参数

预定义参数包括基本坐标系参数，以及一系列标准速度、转弯区设置，这些变量用于 RL 指令的参数。在本页面可以查看变量的物理含义，暂不允许编辑。
在编程选项卡->工程配置->预定义参数页面，可以看到控制器提供的预定义参数。



在设置选项卡->运动参数->运动参数页面，可以看到预定义参数 vmax 的的配置选项，最大 TCP 速度和最大外部轴速度。vmax 是唯一一个可以由用户配置的预定义参数。

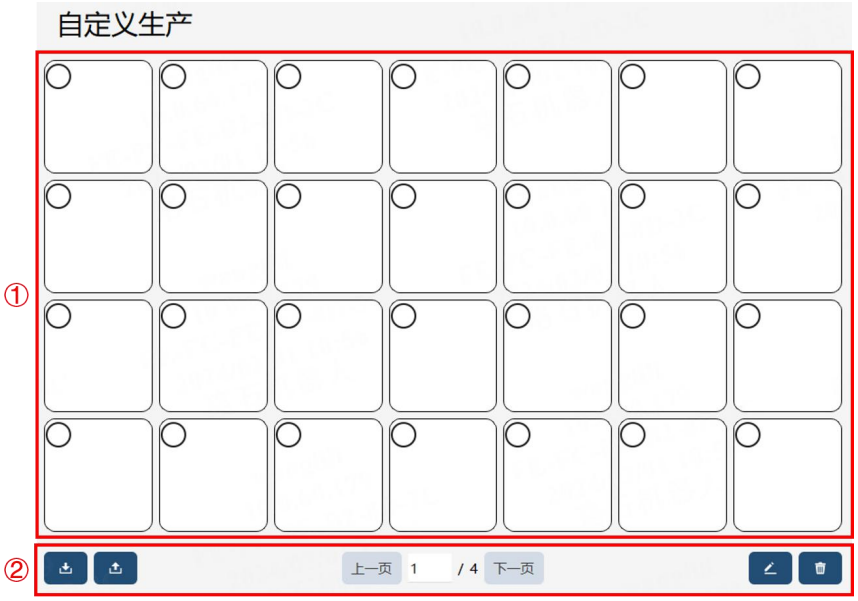


8.5 自定义生产

自定义生产界面为用户提供了一种简易直观的交互式方式。用户可以通过简单地选择感兴趣的寄存器、DI/DO 信号、点位、工程变量，构成极简的监控、交互界面。

该界面支持的功能包括：寄存器的显示及编辑、DI 信号的状态显示、DO 信号的使能操作、点位的更新位置操作、工程变量的显示及编辑。

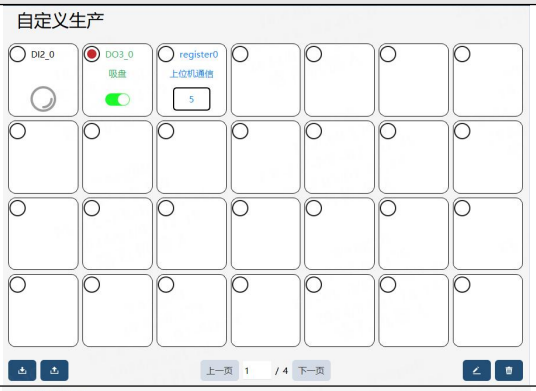
8.5.1 概述

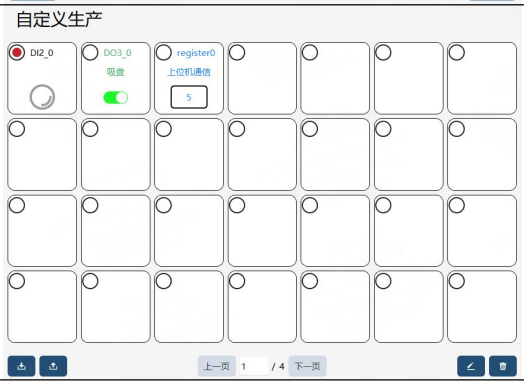


①	操作/显示面板，主要包含不同控件的选择、显示、操作。 当前可选择控件包含单路 DI、单路 DO、寄存器、更新点位、工程变量。
②	面板编辑栏，包括自定义生产界面数据的导入导出、页面的切换、单个控件的编辑和删除。

操作/显示面板中包含多个生产任务控件，可通过下方按钮进行导入、导出、编辑、删除操作。控件包含三个部分，顶部显示该控件的信号名称；中部为该控件的标签文本（用户可自定义），底部可供用户对控件进行的操作/显示。不同的控件提供的交互方式不同。

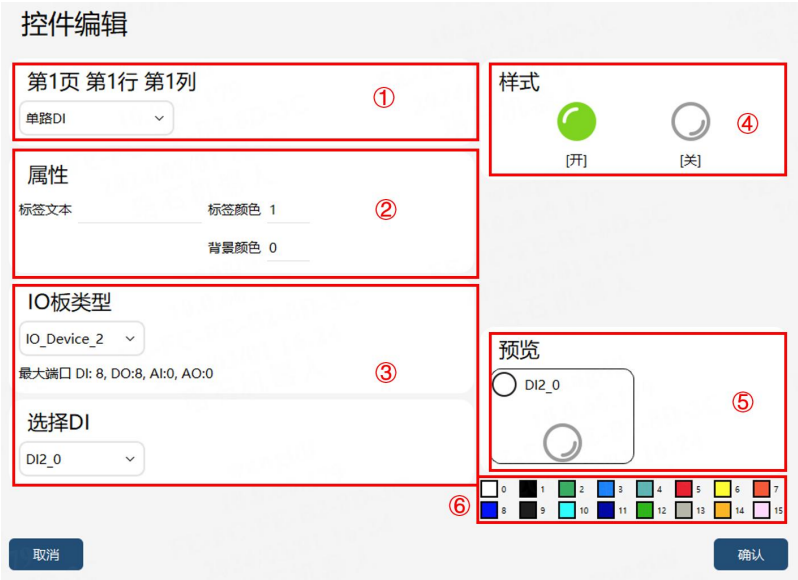
8.5.2 基本操作

操作	图示	说明
控件选择		点击控件左上角的圆圈可以在面板中选中某个控件，被选中的控件左上角圆形框中将显示红色。例如：当前面板被选中的控件为第 1 行第 2 列的“吸盘”控件。
导入		点击按钮进入导入界面，选择需要导入的文件路径、导入条目、是否替换。
导出		点击按钮进入导出界面，选择需要导出的控件、导出路径。

编辑		点击按钮，进入控件编辑页面，详细使用方式见下文。
删除		点击按钮，删除选中的控件。

8.5.3 控件介绍

自定义生产界面目前提供了四种控件，用户可以结合生产环境搭配出适合生产环境的面板风格。这四种控件分别是单路 DI 控件、单路 DO 控件、寄存器控件、PERS 变量控件。控件编辑页面如下。



①	选择控件的类型，包括：单路 DI、单路 DO、寄存器、PERS 变量、空。
②	通用属性，包括：标签文本、标签颜色、背景颜色，标签代表的颜色见⑥。
③	特殊属性，每种控件的特殊属性都不尽相同，详细介绍见下文。
④	样式栏，提供了不同控件所显示的数据表达风格。 个别控件不提供样式，如寄存器控件。
⑤	预览栏，可以预览当前编辑控件最终风格效果。
⑥	16 种颜色可选，在②中输入期望的颜色编号。

8.5.3.1 单路 DI 控件

单路 DI 控件，可用于 DI 显示。可选择的 IO 板类型包含：(1)用户在"IO 设备"配置的 IO 设备；(2)用户在工程的"IO 信号列表"中配置的用户 DI 信号。

编辑界面如下图所示：

当所选 DI 为 True 时，控件将显示样式中的"开"图标，即绿灯；IO 为 False 时，显示样式中的"关"图标，即灰灯。

单路 DI 控件不提供控件交互，仅作为 DI 状态显示。

8.5.3.2 单路 DO 控件

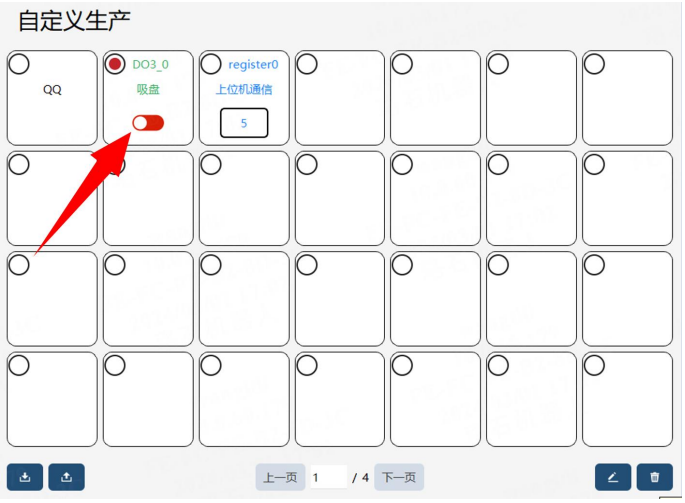
单路 DO 控件，可用于 DO 信号的显示、设置。可选择的 IO 板类型包含：(1)用户在 IO 设备界面配置的 IO 设备；（2）用户在工程的"IO 信号列表"中配置的用户 DO 信号。

编辑界面如下图所示：

当所选 IO 为 True 时，控件将显示样式中的[开]图标，即绿色打开按钮；IO 为 False 时，显示样式中的[关]图标，即红色关闭按钮。

单路 DO 控件提供了按钮开关的交互方式，可以通过"允许操作"属性，设置控件在"操作/显示面板"是否可以操作。

在"操作/显示面板"界面点击下图红框的按钮，可以对 DO 信号进行操作。



8.5.3.3 寄存器控件

寄存器控件可以监控和修改"通信"-"寄存器"配置的寄存器值。可通过"总线设备"来筛选寄存器。

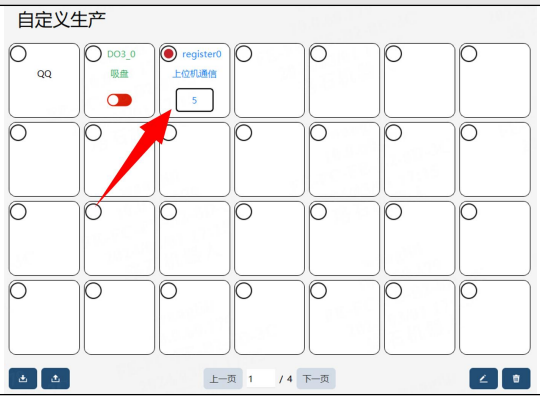
注意：寄存器控件不支持寄存器数组类型。

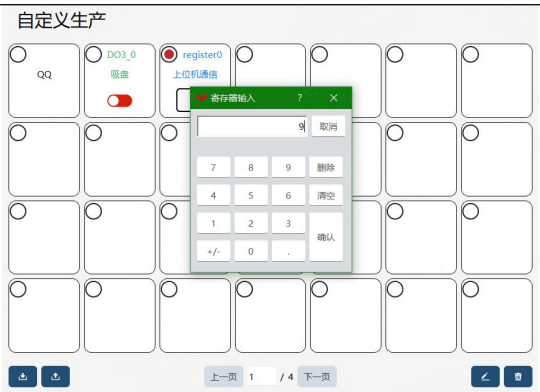
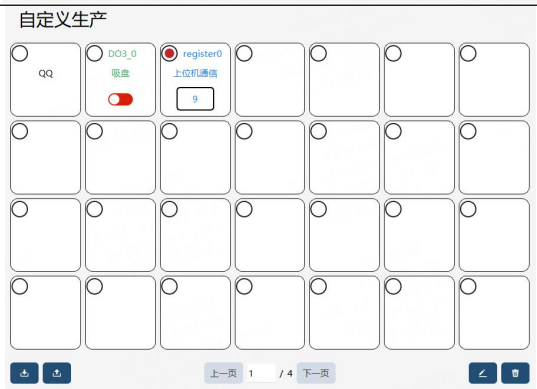
编辑界面如下图所示：

寄存器控件不提供状态样式显示，可以提供寄存器数值显示。

寄存器控件采用数字键盘值写入的方式进行编辑，在"操作/显示面板"界面，点击值"显示/编辑框"，可以写入寄存器值（注意：仅当寄存器为"只写"模式时才允许写入）。

值写入操作步骤：

步骤	图示	说明
1.选择		点击寄存器控件中的值显示/编辑框。

2.输入		输入需要设置的寄存器值。
3.值写入成功		

8.5.3.4 更新点位控件

更新点位控件可以修改点位列表中的点位信息，可以选择的点位类型有笛卡尔与关节两种。编辑界面如下图所示：

更新点位控件不提供状态样式显示，可以提供更新位置按钮。

在"操作/显示面板"界面点击更新位置按钮，可以对点位进行值更新操作。

更新点位操作步骤：

步骤	图示	说明
1.选择		选择空的控件
2.选择更新点位功能		
3.选择需要更新位置的点位		支持笛卡尔与关节两种类型的点位
4.完成		点击完成即可再自定义生产界面中对点位进行更新位置操作

8.5.3.5 工程变量控件

工程变量控件可以监控和修改"工程"-"变量列表"配置的变量，可以选择的变量类型有int/byte/bool/double 四种，不支持变量数组类型。

编辑界面如下图所示：

工程变量控件不提供状态样式显示，可以提供工程变量的数值显示。

工程变量控件采用数字键盘值写入的方式进行编辑，可以通过"允许操作"属性，设置控件在"操作/显示面板"是否可以操作。

在"操作/显示面板"界面点击值显示/编辑框，可以对工程变量进行值写入操作。

值写入操作步骤：

步骤	图示	说明
1.选择		点击工程变量控件中的值显示/编辑框

2.输入		输入需要设置的工程变量值
3.值写入成功		

8.6 任务列表

xCore 控制系统支持多任务。

通过多任务，可实现多个机器人程序的"并行"，典型应用场景：

- 持续监控某个信号，即使 Main 主程序已经停止运行(类似于后台 PLC 功能，但是响应速度比不上真正的 PLC)；
- 在机器人执行运动主程序的同时，跟外设进行数据接收、发送等处理，而不受到主程序执行逻辑的限制；
- 机器人工作的时候，通过示教器获取一些输入；

xCore 系统的"任务列表"，提供了对并行处理任务的管理界面，用户可以在该界面查看已有任务属性，新建、编辑、删除任务。

8.6.1 任务属性

新建或编辑任务时，需要设置任务属性。

任务属性	描述
任务名称	任务的名称在所有任务中必须是唯一的，任务名称仅支持字母数字下划线，首字母不能为数字，最长为 20 个字符。
描述	描述任务的作用，辅助使用者理解。
任务类型	包括常规任务、运动任务、半静态任务、视觉任务（配合 xVision 使用）。其中，运动任务指可以使用 RL 指令来控制机器人的运动，一个工程只能运行一个运动任务。
自动开始	配合生产模式使用，选中自动开始表示系统重启时程序开始重新执行。通常情况下不会被示教器或者急停停止。
优先级	设置任务运行的优先级。
创建文件	勾中生成 main 函数时，创建任务之后会自动生成 main 函数；其他函数同理。

8.6.2 常规任务、运动任务

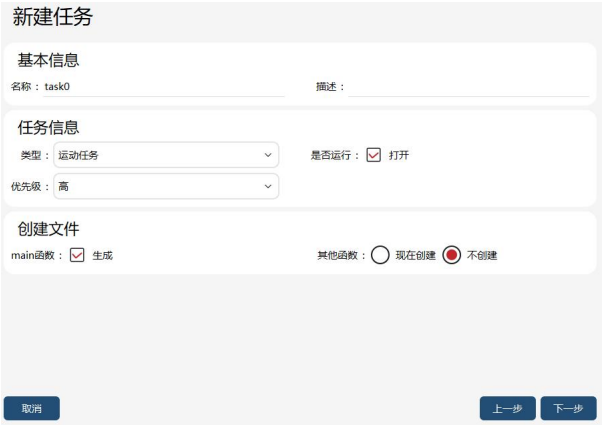
8.6.2.1 新建任务

新建第一个任务前，请先新建工程。如果已有工程，想新增一个任务，则无需重复新建工程。

新建工程

新建任务

新建一个任务后，可以编辑该任务的属性。



任务属性编辑

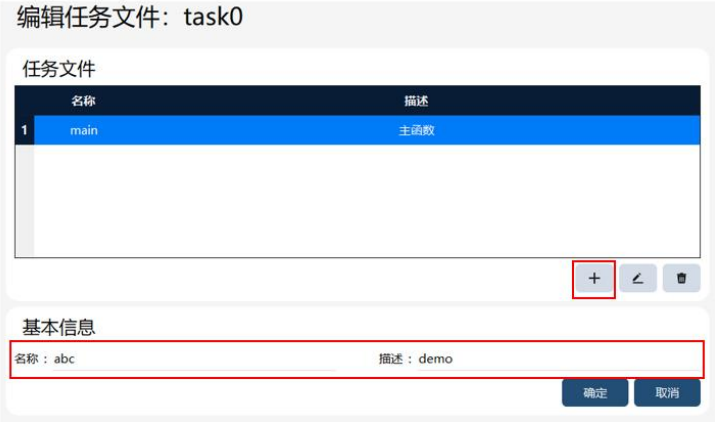
注意：

- 支持 10 个任务。
- 最多只能有一个运动任务运行。
- 更改任务类型、任务入口函数、是否运动任务属性即时生效。

8.6.2.2 新建程序模块

每个任务可以包括若干程序模块(mod 文件)，如下图所示，点击编辑按钮。

在新的页面点击"+"，新建程序模块。设置模块的名称、描述等基本信息后，点击"确定"，完成程序模块的创建。

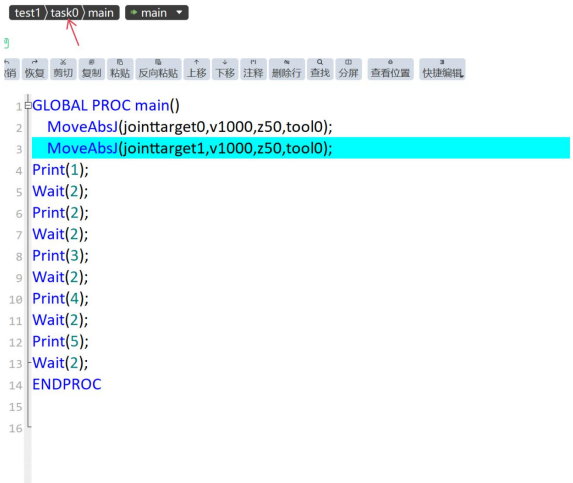


此时，任务列表可以查看新建的程序模块。

RL 编辑器上方，也可以选择新的程序模块，并进行编程。

8.6.2.3 启动和运行

在 RL 编辑器上方点击 task0，可以选择任务。在手动使能或者自动上电情况下，可使用启动/停止按钮或者外部信号控制选中任务的启停。



使用限制：

- 通常情况下，一个后台任务会一直循环运行。如果一个任务中不包含任何等待指令，那么后台任务可能会消耗过多的计算器资源而导致控制器无法处理其他的任务；
- 变量 VARS 和常量 CONST 作用域限制在各自的任务中，但是 GLOBAL 级别的 PERS 变量为全局变量；
- 当执行 PPToMain 时，所有没有运行的任务都执行 PPToMain；
- 有任务运行时，禁止修改任务列表中的内容；

8.6.2.4 任务间通信

任务间通信支持两种方式，PERS 变量。

任务间使用 PERS 变量通信

- 在所有需要进行通信的任务工程中都定义相同名称的 GLOBAL 级别的 PERS 变量，且变量的数据类型、维数均应相同；
- 在需要的地方使用 PERS 变量来控制任务执行、传递数据等；
- 变量列表、点位列表的所有变量所有任务都可以随意使用；

使用限制：

- 只需在其中一个任务中为 PERS 变量指定初始值即可。如果在多个任务中为同一个 PERS 变量指定了初始值，那么将使用第一个运行的任务中定义的初始值。
- 通过 PERS 变量以及 WaitUntil 或 WHILE 指令的方式让一个任务等待另一个任务时，需要注意配合等待指令（大于 0.1s），避免程序快速执行空判断语句导致占用过多系统资源。

8.6.3 半静态任务

从 v2.0.1 版本开始，xCore 控制系统提供半静态任务功能。

半静态任务从属于多任务功能，运行的也是 RL 语言编写的程序。相比常规任务和运动任务，半静态任务有两个特点：一是它在正确配置之后，无需上电、启动等指令，就能开机后自启动；二是暂停按钮不会停止半静态任务。这两个特点使得半静态任务的运行周期覆盖几乎整个控制系统从开机到关机的时间。

半静态的典型应用：

- 周期性判断机器人的位置，并通过寄存器、IO 等通知上位机；
- 输出自定义心跳信号；
- 转发多个设备之间的数据；

8.6.3.1 创建半静态任务

在任务列表，点击"+"，新建任务。任务类型选择半静态任务，完成其他属性设置；点击"下一步"按钮，创建一个半静态任务。

8.6.3.2 启动、停止半静态任务

跟常规任务的启动一样，创建并配置好半静态程序后，手动上电或者自动模式上电后，点击运行，即可启动半静态任务。如果工程中有半静态任务正在运行，则在底部状态栏中间会提示  ，此时点击暂停或者下电，半静态任务都不会停止，此时点击这个"半静态状态"的按钮，将弹窗确认对话框 

，点击"确认"将执行停止半静态任务操作。半静态任务在不触发程序异常时，只有这个按钮可以停止半静态任务（需要先暂停常规任

务，才能暂停半静态任务）。

8.6.3.3 配置半静态开机自启动

在工程配置界面，将半静态任务所属的工程设置为默认工程

并在任务列表中，勾选半静态任务的开机运行和是否运行两个选项。

最后再点击 pptomain，将工程配置从 HMI 同步到控制器，可使半静态任务自启动。

8.6.3.4 半静态任务安全级别

除了常规任务的优先级和是否运行等配置，半静态任务有一个额外属性：安全级别。

它用于定义半静态任务工作出现异常（比如数据接收失败）时，控制器对于半静态任务的异常处理策略。

- 安全无关：适用于上位机或者操作人员，对半静态任务的工作情况不在意的情况，配置该安全级别的半静态任务如果运行出错，则只会将自己停止，其他的任务运行不受影响。
- 系统停止：适用于半静态任务的数据影响程序安全逻辑的情况（比如通过半静态决定运动点位、通过半静态通知上位机如何控制机器人），系统停止级别的半静态任务如果运行出错，会将所有任务暂停。

8.6.3.5 半静态任务调试建议

- 控制器支持在手动上电模式对半静态任务进行单步调试，但是仍然建议先使用常规任务调试完成之后，再将任务类型改为半静态任务，以避免调试过程中重启，未调试完成的半静态任务开机启动引发意外；
- 半静态任务中对可能执行失败的指令尽量增加错误处理，比如 ReadXX 指令就很可能因为网络波动、外部设备意外种种原因导致读取超时或者失败，如果要达成半静态任务尽量不停止的需求，则应该使用 try_catch 语句将可能失败的代码保护起来，并在 catch 语句中进行错误处理（比如 goto 到 Read 之前重新读数据）；
- 由于半静态任务本身长期执行的特殊性，程序调试界面右上角的单次 循环 运行设置对半静态任务不生效，所有半静态任务都是循环运行。

8.6.4 任务监控

可以在状态监控中，监控当前各任务运行状态。

8.7 变量列表

8.7.1 变量命名规则

RL 语言中的变量名称可由字母、下划线、数字组成，必须以字母或者下划线"_"开头，但变量名不能与系统关键字重名，RL 系统关键字详见预定义关键字。
除此之外，还存在以下注意事项：
在同一模块中，不允许出现重名的 GLOBAL、LOCAL 级别变量；
在不同的模块中，不允许出现重名的 GLOBAL 变量；
在不同的模块中，允许出现重名的 LOCAL 级别变量；
在同一模块中，不允许任何变量（GLOBAL，LOCAL 级别，不包括 ROUTINE 级别）与本模块内的函数发生命名冲突；
在不同的模块中，不允许任何 GLOBAL 级别的函数与变量发生命名冲突；

> 提示
> 当变量名只包含两个字符时，需要注意第二个字符不要为"h"、"b"，否则该变量可能会被转换为十六进制或者二进制，更多信息请参考进制转换。

8.7.2 变量作用域

RL 语言系统定义了三种作用域：

全局（GLOBAL）	对当前任务的任意程序模块可见，可在模块声明区内声明； 如果需要跨任务访问变量，需要用 GLOBAL PERS 关键字进行声明。 注意：变量列表、点位列表中的变量均为全局，所有任务可读可写。
局部（LOCAL）	仅对当前程序模块可见，可在模块声明区内声明；
函数（ROUTINE）	仅在当前函数内可见，只能在函数体内部声明，且声明该作用域变量时不允许指定作用域类型（GLOBAL 或者 LOCAL）； 注意：作用域仅适用于变量，不适用于自定义函数；

8.7.3 存储类型

每个变量根据其是否可以在程序执行过程中被修改，分为三种：可变量（VAR）、持续性变量（PERS）、常量（CONST）。

VAR（Variable）	可变量，可以在程序运行过程中进行重新赋值的变量；

CONST（Const Variable）	常量变量，不能在程序运行过程中重新赋值的变量，此类型变量的值必须在其初始化时指定；
PERS（Persisten Variable）	持续性变量，在程序执行过程中如果此类型变量的值发生了变化，将会自动将该变量的初始值修改为当前值，由此达到"持续"存储的效果； 注意： ● 即使某个 PERS 类型变量的值在程序运行过程中被更改，其在程序编辑器声明区内显示的初始值也不会立即刷新，只有当程序重新加载或程序停止时在程序编辑器声明区显示的初始值才会更新到最新值； ● 不论程序运行与否，在"变量管理"界面只能查看 PERS 变量的初始值，可以通过状态监控或者 print 指令查看当前值。

8.7.4 预定义关键字

以下为 RL 语言预定义的保留关键字（不区分大小写）：

Module、EndModule、Proc、EndProc、Func、EndFunc、SetDO、DO_ALL、SetGO、SetAO、WaitDI、Wait、WaitUntil、WaitWObj、WBID、Q、P、J、V、W、T、S、L、CA、DURA、IGNORELEFT、EJ、1J、FCBV、FCCV、FCOL、FCXYZ、FCCART、PE、PER、TCP、ORI、EXJ、CFG、PDIS、JDIS、MoveAbsJ、MoveJ、MoveL、MoveC、MoveT、LOCAL、TASK、GLOBAL、VAR、CONST、PERS、INV、DOT、CROSS、sin、cos、tan、asin、cot、acos、atan、atan2、sinh、cosh、tanh、ln、log10、pow、exp、sqrt、ceil、floor、abs、rand、GetCurPos、Print、PrintToFile、ClkRead、TestAndSet、IF 、Else、Endif、WHILE、ENDWHILE、for、from、to、endfor、Break、Continue、Del、Int、Double、Bool、String、BYTE、Robtarget、Speed、Zone、Tool、Wobj、Jointtarget、TriggData、Load、FCBoxVol、FCSphereVol、FCCylinderVol、FCXyzNum、FCCartNum、Pose、CLOCK、INTNUM、SYNCIDENT、TASKS、Call、Return、EXIT、Pause、StopMove、StartMove、StorePath、RestoPath、True、False、Interrupt、When、Offs、CalcJointT、CalcRobT、CRobT、RelTool、SocketCreate、SocketClose、SocketSendByte、SocketSendInt、SocketSendString、SocketReadString、SocketReadBit、SocketReadInt、SocketReadDouble、AccSet、MotionSup、TriggIO、TriggJ、TriggL、TriggC、On、Off、clock、intnum、userframe、pinf、ninf、FCFRAME_WORLD、FCFRAME_TOOL、FCFRAME_WOBJ、FCFRAME_PATH、FCPLANE_XY、FCPLANE_XZ、FCPLANE_YZ、FC_LINE_X、FC_LINE_Y、FC_LINE_Z、FC_ROT_X、FC_ROT_Y、FC_ROT_Z、Offs、CalcJoinT、CalcRobT、CRobT、RelTool、Start、Time、ClkReset、ClkStart、ClkStop、CONNECT、WITH、IDisable、IEnable、ISignalDI、Single、SingleSafe、WaitWobj、DropWobj、WobjIdentifier、WobjAngle、ActUnit、DeactUnit、INTNO、Exp、DoubleToStr、WaitSyncTask、FCAct、FCDeact、FCLoadID、FCCalib、FCSupvForce、FCSupvTorque、FCSupvPosBox、FCSupvPosSphere、FCSupvPosCylinder、FCSupvOrient、FCSupvOrient、FCSupvReoriSpeed、FCSupvTCPSpeed、FCCondForce、FCCondTorque、FCCondOrient 、 FCCondReoriSpeed 、 FCCondPosBox 、 FCCondPosCylinder 、 FCCondPosSphere、FCCondTCPSpeed、FCCondWaitWhile、FCRefLine、FCRefRot、FCRefSpiral、FCRefCircle、FCRefForce、FCRefTorque、FCRefStart、FCRefStop、FCSetSDPara

8.7.5 进制转换

RL 语言支持在数字或者字母后面增加进制标识符的方式来直接输入十六进制、二进制字符或者科学计数法的数值。

例 1

在 0~9、a~f 以及 A~F 后面增加"h"后缀，RL 编译器会将相应的数字或者字母当做十六进制来处理，并在编译器内部转换成十进制处理：

8h，代表十六进制的 8，十进制的 8；

bh，代表十六进制的 b，十进制中的 11；

25h，代表十六进制的 25，十进制的 37；

例 2

在 0~9、a~f 以及 A~F 后面增加"b"后缀，RL 编译器会将相应的数字或者字母当做二进制来处理：

1b，代表二进制的 1，十进制的 1；

10b，代表二进制的 10，十进制的 2；

1010b，代表二进制的 1010，十进制的 10；

例 3

在数字后面增加"e±x"，表示该数字乘以 10 的 x 次方，例如：

5e+20，代表 5×10^20 ；

26e-15，代表 26×10^(-15)；

112e-10，代表 112×10^(-10)；

8.7.6 变量声明

在使用一个变量之前必须要先进行声明，变量声明语句格式为：

SCOPE STORAGE TYPE varname [= value]

其中：

1. SCOPE 为变量作用域，详见变量作用域；
2. STORAGE 为变量存储类型，详见存储类型；
3. TYPE 为变量类型，可以是基本类型，也可以是专用类型，详见变量类型；
4. varname 为变量名，详见变量命名规则；

中括号[ ]内为可选内容，可以在声明变量的时候进行初始化，也可以不初始化。对于在声明时没有进行显式初始化的变量，系统会自动根据变量的类型赋予不同的初值。默认的初值可能会在某些情况下造成程序执行问题，因此建议对每一个手动新增的变量进行初始化。

示例

以下是几个变量声明示例：

例 1

```
VAR int counter = 8; //声明整型变量 count，并赋初值为 8
VAR double time = 2.5; //声明浮点型变量 time，并赋初值为 2.5
VAR bool ifOpen = true;  //声明 bool 型变量 ifOpen，并赋初值为 true
```

例 2

一般情况下，变量不允许重名：

```
VAR int counter = 8;
VAR double counter = 2.5;
```

此时编译器将报错，提示"添加变量失败"。

例 3

但是全局变量和局部变量变量可以使用相同的变量名：

VAR int counter = 1;

GLOBAL int counter = 555;

虽然不同作用域的变量允许重名，但是为了避免引起混淆和误用，除非使用重名的变量在工艺上有特别的好处，否则不建议使用重名变量。

> 提示
>
> 不可在 while 等循环语句块内部声明变量，否则在该部分代码重复执行时会造成重复声明，导致出现"添加变量失败"错误。请在循环体外部声明需要使用的变量。

使用限制

不支持声明 PERS 存储类型的 ROUTINE 变量；

当不同级别的变量或函数存在重名情况时，编译器会根据作用域的优先级来决定选择使用哪个变量，具有最高优先级顺序的变量将优先被选中，而低优先级顺序的将被遮蔽隐藏，各作用域的优先顺序如下：

- 当出现变量重名时，作用域的优先顺序为：ROUTINE > LOCAL > GLOBAL；
- 当出现函数重名时，作用域的优先顺序为：LOCAL > GLOBAL；

8.7.7 用户变量保持

在 RL 的同一个工程中创建一个带保持的用户变量 a，该用户变量被标记成 pers 变量，那么该变量的值在 RL 停止、机器人重启、机器人关机、机器人断电时保持在非易失存储介质上，当机器人重新上电或者 RL 重新运行时，该变量 a 的值就会恢复成保持值。只有变量第一次创建或者变量重新编辑，变量才会被赋初始值。(注意：只有在变量列表中添加的 pers 变量，才具有保持属性，变量声明区定义的 pers 类型变量，不具备保持属性。)

用户变量目前支持保持的数据类型：Int、byte、double、bool、string、pose、speed、zone、fcboxvol、Fcspherevol、fccylindervol、fcxyznum、fccartnum、torqueinfo、socketserver、socketconn、serials。

用户变量保持配置

在 RL 工程界面，可以创建保持变量的入口如下红框所示：

点击变量、点位、工具、工件任一按钮，进入创建该类型的用户变量。所有可以创建保持属性的变量都有一个"持续"属性项，"是"表示该变量是保持变量，即标记为 pers 变量。比如创建一个 int 类型的 pers 变量，配置如下：（其他类型以此类推）

保持变量的修改

xCore 控制系统的保持（pers）变量采用初值+保持值的形式存储。

初值指的是用户在变量列表输入的数据，保持值是保持变量在程序运行过程中被程序修改后，存储在非易失存储介质上的数据，基础类型（int\bool\double）的保持值可以通过状态监控查看，结构体数据（比如点位、工具、工件）则可以通过 print 指令打印。

RL 程序运行过程中，可以通过运算符"="对保持变量进行修改，修改后的数据将作为保持值存储在控制器内部。下一次运行程序时，会优先读取对应变量的保持值，当保持值不存在的时候，会读取变量的初始值。

如果通过点位更新按钮或者变量列表的编辑功能修改保持变量，则会同时修改变量的初值和保持值。

8.7.8 变量列表操作

变量列表面提供了对机器人系统内几乎所有变量的新建、查看、修改及删除操作，目前支持的变量类型包括：

序号	变量类型	描述
1	系统预定义变量	指用户不可修改的变量，用于存储某些系统参数，例如 tool0/wobj0 等。

2	用户预定义变量	用户可以进行修改，并且可以在多个程序中进行使用的变量，例如用户标定的工具、工件等。
3	程序变量	用户在程序中进行定义的变量，一般仅在当前程序及其子程序中使用，包含了绝大部分系统支持的变量类型。

8.7.8.1 变量查看

对于一些某些有专门定义步骤的变量类型，例如：speed/zone（使用辅助编程界面进行定义和修改），虽然也可在变量查看界面进行查看和修改，但是出于方便性和减少错误的考虑，仍然推荐使用专用界面进行修改，在变量管理界面只进行查看操作。

> 提示
>
> 可以在变量列表界面查看和修改的变量仅限于当前所加载的机器人程序中使用的变量，因此加载其他程序后显示的变量会发生变化。

8.7.8.2 变量编辑

如果需要新增变量或者对某个存在的变量进行修改，可通过点击"新建"或者"修改"按钮进入变量编辑页面进行操作。

类型	在新建变量时用于选择变量的类型，所有支持的类型都列在左侧边栏内。
名称	输入变量的名称。
维数	用于创建或者修改数组，最大支持 3 维数组。
持续	定义为 pers 变量或者非 pers 变量
值	显示变量值
描述	变量描述信息描述

8.8 点位列表

8.8.1 概述

点位列表用于统一管理工程中涉及的机器人点位。RL 程序中用到的点位，都需要先在点位列表进行配置，然后才可以在程序中使用。

①	点位过滤，可以筛选需要显示在下方列表中的点位，支持类型、任务、名称、描述、函数筛选。
②	点位展示，展示各点位的：名称、属性、是否持续、位置、工具、工件坐标系、任务、函数等属性。
③	点位编辑，使用"更新位置"按钮可更新点坐标为机器人当前位姿； 右侧按钮用于移动机器人至该点位：点击"运动至"按钮会弹出运动窗口，对于笛卡尔类型的点位，可点击 MoveL 或 MoveJ 按钮，机器人采用直线或轴空间运动方式运动至该点位；对关节类型的点位，可使用 MoveAbsJ 按钮，机器人采用轴关节运动方式运动至该点位。 ~~注意：点位运动至功能使用的工具工件为点位创建、更新时的工具工件。~~
④	功能按钮区，导入点位，导出点位，新建点位，编辑点位，删除点位，多选点位

点位编辑页面：

笛卡尔空间点位：

①	基本信息，包括点位名称、描述、是否保持，维度、任务和函数。
②	位姿信息，点位类型不可修改。 点击"更新位置"按钮会更新点位至当位姿，也可手动更改；手动修改笛卡尔数据不会改变关节数据，反之亦

	然。 Conf 参数，可以修改点位对应的 conf 参数。
③	“取消”返回点位列表页面，不会保存当前更改；“完成”按钮保存当前更改，并返回列表。

轴空间空间点位：

①	基本信息，包括点位名称、描述、是否保持、维度、相关坐标系、任务、函数等信息；点击维度添加、删除按钮可以修改点位维度。
②	位姿信息中，当数组为多维度时，可以选择不同维度，修改其内容。 点击“更新位置”按钮会更新点位至当前所选择维度的位姿，也可手动更改。对笛卡尔类型的点位，手动更改数据不会改变关节数据，反之亦然。
③	“取消”返回点位列表页面，不会保存当前更改；“完成”按钮保存当前更改。

8.8.2 操作示例

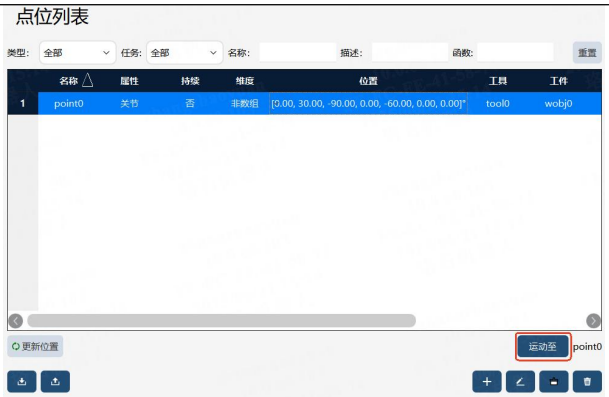
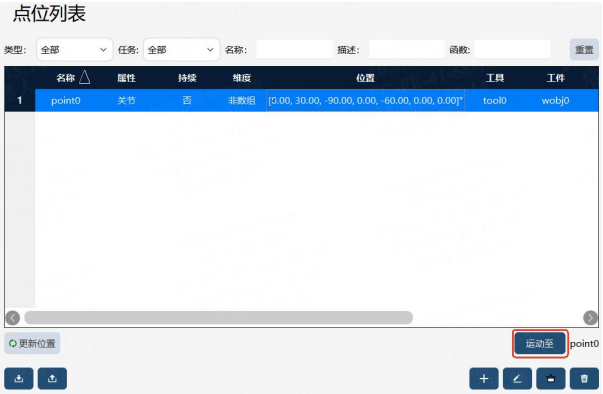
8.8.2.1 点位创建/编辑

操作	图示	说明
1.新建/编辑		点击 + 或 ∠ 按钮，进入点位编辑页面

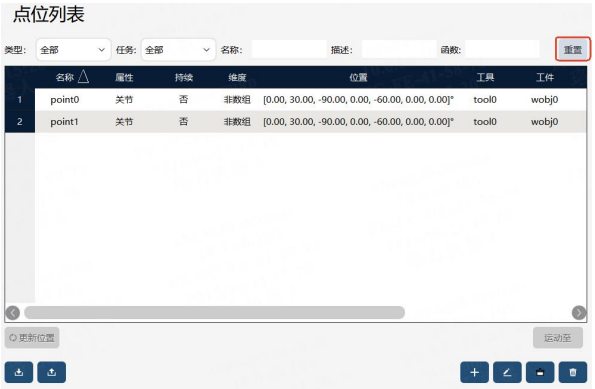
2.基本信息编辑		只有当新建点位时，点位类型才可以选择； 编辑点位时，只能选择名称、点位描述（非必须），是否持续、维度（非必须）、任务、函数等；用户可以通过维度添加与删除按钮来设置点位的维度。
3.位姿信息编辑		选择点位类型，通过“更新位置”获取机器人当前位姿；若点位为新建点位，默认获取当前位姿。
4.完成		点击“完成”按钮，保存当前信息。

8.8.2.2 点位运动至

操作	图示	说明
1.选择对应点位，点击运动至按钮		

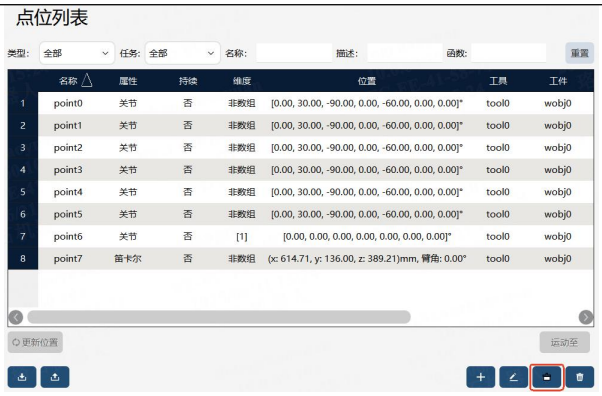
		
2.选择点位运动的工具工件		只有选择的工具工件与点位信息内工具工件一致时，才能不点击确定按钮即可点击运动至按钮；不对应时，运动按钮置灰，不能直接运动，需要点击确定按钮后方可运动。
3.进行运动至操作		关节类型的点位，只能进行 MoveAbsj 运动，笛卡尔类型的点位只能进行 MoveJ 以及 MoveL 运动。

8.8.2.3 点位排序

操作	图示	说明
1.初次进入点位列表界面，排序符号默认显示在名称右侧		箭头朝下为降序，箭头朝上为升序
2.点击排序		根据当前排序符号，进行排序。注意未使用排序功能，初次进入界面的降序符号，不代表点位列表为降序。
3.排序重置		点击重置，会将点位列表点位恢复排序之前的状态。

8.8.2.4 点位批量删除

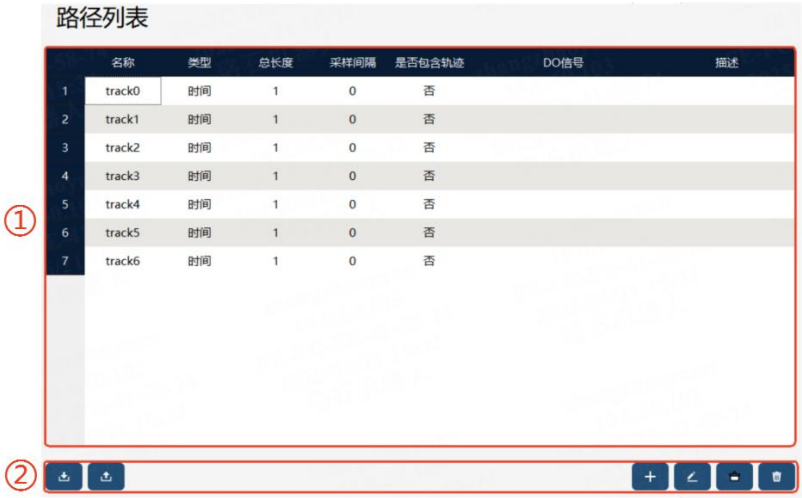
操作	图示	说明
1.点击多选按钮		点击多选按钮，进入多选状态。

		
2.选择要删除的点位		点击点位前的选择框选中点位，也可通过点击全选来选择全部点位。
3.删除点位		点击删除按钮，并在二次确认对话框中点击确定，删除选中的点位。

8.9 路径列表

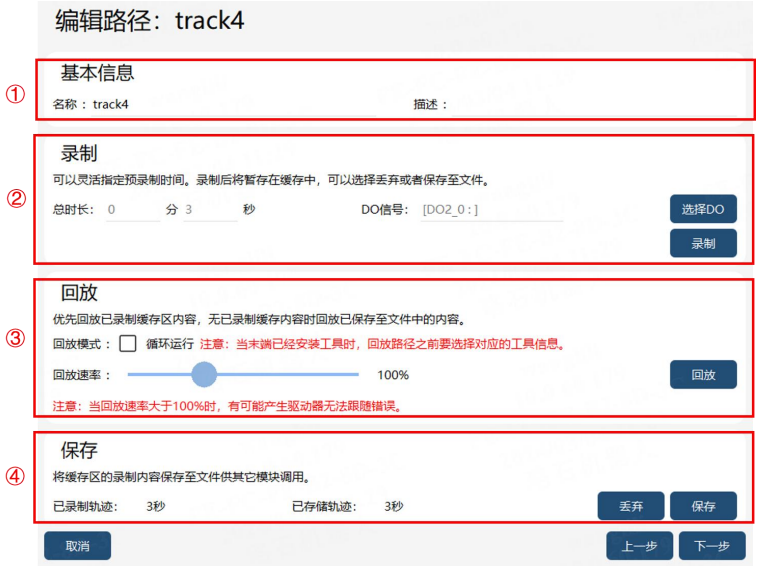
8.9.1 概述

路径列表用于记录拖动示教的轨迹，进行轨迹回放等操作。



①	路径列表，显示相关属性，包括：名称、类型、总长度、采样间隔、是否包含轨迹、DO 信号、描述。
②	功能按钮区，导入路径，导出路径，新建路径，编辑路径，多选路径，删除路径。

路径编辑页面：



①	基本信息，设置路径名称、描述。
②	录制路径，如需录制 DO 信号，点击"选择 DO"按钮，在弹出的对话框中选择 DO 信号，每个 DO 信号还可以设置与一个 DI 信号做映射，这种情况下，录制路径时会记录 DI 信号的变化，后续回放时，DO 信号的输出按记录的 DI 信号进行；若不关联 DI 信号，则会直接记录 DO 信号的变化，此时可以在"状态监控"-"IO 信号"界面手动设置 DO 信号输出。总时长：为录制总时长
③	回放，回放录制的路径，可设置是否循环运行、运行速率等，点击"回放"按钮，进行回放，确认录制的路径是否与期望的一致。
④	可保存缓存区的轨迹，供后续使用。丢弃按钮：丢弃此次录制

8.9.2 操作示例

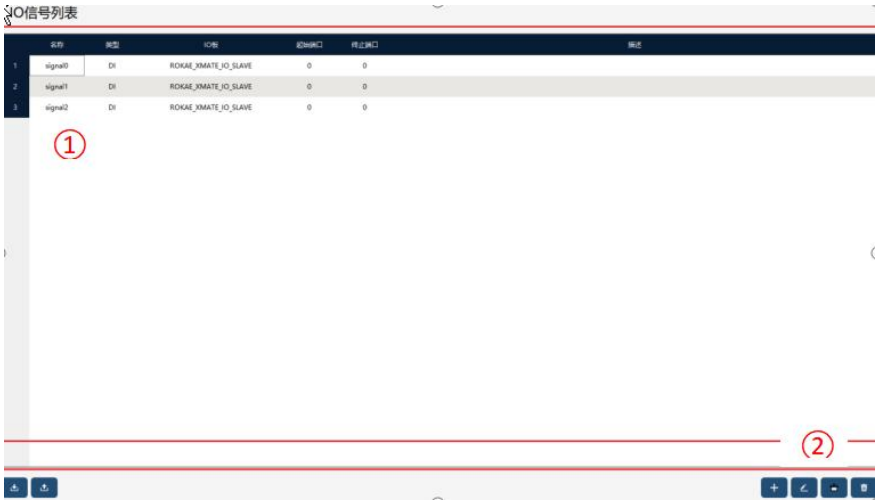
操作	图示	说明
1.打开拖动模式。点击右下角“+”按钮，新建一个路径。		此操作也可在机器人末端按键操作。
2.设置录制时长，按下“录制”按钮后，拖动末端进行轨迹录制，直到计时结束或点击“停止”按钮。		注意：请在录制时间计时结束前停止拖动，否则路径录制失败；
3.关闭拖动使能开关，切换至自动模式，点击上电按钮，机器人上电。设置回放速率、是否循环，点击“回放按钮”，查看轨迹回放。		
4.轨迹回放完成后，完成轨迹确认，点击保存，录制轨迹成功保存到文件。		

8.10 IO 信号列表

8.10.1 概述

除开启状态的"IO 设备"默认创建的通用 IO 信号外，若想要单独使用 IO 设备创建用户 IO 信号，需要在"IO 信号列表"页面进行创建。

在 RL 程序中使用 signalxx 类型的变量来存储和访问 IO 信号，详细信息请参考 RL 指令相关章节。



①	IO 信号列表，可以查看 IO 信号属性，包括： ● 信号名称； ● 类型：包括 DI/DO/GI/GO； ● IO 板：可以是珞石提供的标准 IO 模块，也可以是 Profinet 总线或 Ethernet/IP 总线设备； ● 起始端口、终止端口：IO 信号映射对应的物理地址号。
②	功能按钮，可以对 IO 信号执行导入、导出、新建、编辑，多选和删除操作

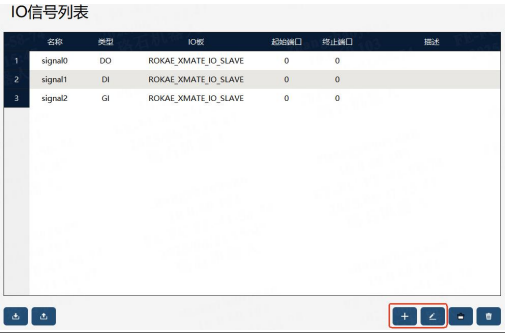
警告

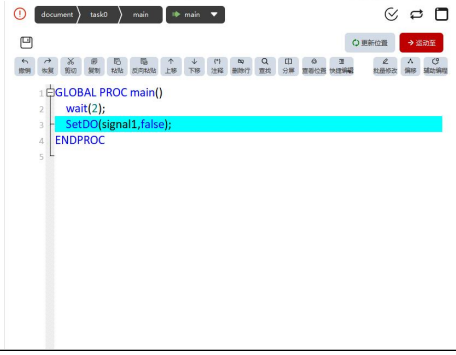
1.如果 IO 配置出现错误，如映射 IO 端口超出物理限制或者出现端口重复分配等情况，控制系统启动时将进入 SYS_ERR 状态，并在 HMI 上给出报警信息，该情况下只允许用户进入系统配置界面修正错误的配置，不允许其他操作。

2.用户自定义 IO 不可映射到系统输出上。

配置完成的 IO 可以在状态监控界面进行查看，该页面支持对 IO 的强制输出或者仿真输入。

8.10.2 操作示例

操作	图示	说明
1.点击右下角“+”按钮，新建一个 IO。		在“编程员”及以上权限时，可进行新建、编辑操作。
2. 设置属性。		设置 IO 信号名称、描述、IO 板、信号类型、端口。
3.在 RL 中使用信号。		对于输入信号来讲(DI/GI)，在 RL 程序中可直接使用输入信号的变量名读取输入节点的状态。 例 ： 使用数字输入的状态作为判断条件 IF （di1 == true） do something... ENDIF

	 	对于输出信号（DO/GO），在 RL 中可以使用专门的指令 SetDO 和 SetGO 来处理，详见各指令的说明章节。

8.11 用户坐标系统列表

8.11.1 概述

用户坐标系，在定义工件坐标系时充当参考系，不单独使用。

建立用户坐标系可以选择"立即标定"、"手动输入"或"不标定"。

选择"立即标定"时，采用 3 点法标定。在标定用户坐标系前，需先标定一个工具，然后使用该工具的 TCP 来进行用户坐标系标定，推荐使用带有尖端的工具。

如果提前知道用户坐标系，可以手动输入。也可选择不标定，此时默认用户坐标系是世界坐标系。

8.11.2 标定用户坐标系

操作	图示	说明

1.点击右下角“+”按钮，新建一个用户坐标系。		在“编程员”及以上权限时，可进行新建、编辑操作。
2.按图示步骤，执行标定。		Jog 机器人，使标定好的工具 TCP 按顺序指到期望的坐标系原点、x 轴上一点、xy 平面上一点或 y 轴上一点，点击“确认第一点、第二点、第三点”按钮。 三个点都完成确认后，点击“确认标定”按钮。

8.12 工具列表

8.12.1 概述

工具是指安装在机器人法兰上，用来完成特定加工工序的器具。常见的工具有气动/电动手爪、焊枪、喷头等。机器人出厂时没有附带任何工具，您需要根据实际情况选择外购或者自行设计合适的工具并完成安装和设置，才可使用机器人进行工作。

使用任何一个工具之前都必须先进行标定，以获取工具中心点的数据。

在 xCore 控制系统中，工具对应的数据类型为 tool。关于 tool 的详细描述，请参考“RL 指令-变量”章节。

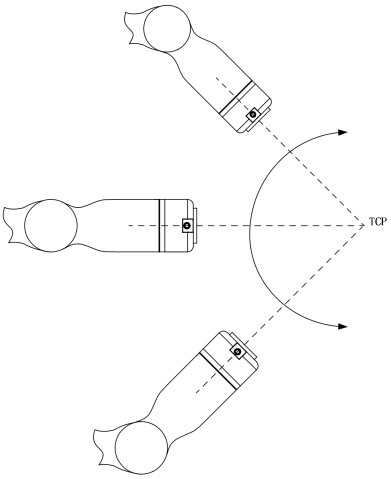
8.12.2 基本概念

工具的属性包括：工具的中心点和姿态，代表了工具的几何参数；工具的重量、质心以及旋转惯量，代表了工具的动力学参数。

> 提示
>
> tool0 是系统预定义的工具变量，其工具坐标系与法兰坐标系重合，动力学参数都是 0。tool0 变量不允许修改。

8.12.2.1 工具中心点

工具中心点（Tool Center Point，TCP）是工具上的一个特定点，通常情况下机器人使用该“点”进行加工作业，例如一个焊枪的焊丝尖端，气动手爪的某一个手指顶端等。机器人可绕 TCP 点旋转变换姿态而保持 TCP 的位置不变。



不同的工具可能有不同的 TCP，根据实际情况定义合适的 TCP 可以大幅提升编程效率。

> 提示
>
> 如无特殊说明，凡是本手册中提到"机器人位置、速度、加速度"的地方，均是指 TCP 相对于工件坐标系的位置、速度、加速度。

8.12.2.2 工具坐标系

工具坐标系标定指的是确定工具坐标系相对于法兰坐标系位姿的过程。

如果已知工具相对法兰的位姿信息，可以在示教器上选择"手动输入"，直接输入，而不必执行标定过程。

如果不了解工具相对法兰的位姿信息，xCore 提供三种标定方法进行工具坐标系标定：

- 4 点法，用来标定工具坐标系的中心点。
- 3 点法，用来标定工具坐标系的姿态。
- 6 点法，同时标定工具坐标系的中心点和姿态，相当于 4 点法+3 点法。

此外，xCore 提供了"TCP 校正功能"以进一步提高工具坐标系标定后的位置精度。

8.12.2.3 负载参数

xCore 系统使用 load 型变量存储工具的负载参数。注意：使用外部工具时，tool 变量中的 load 参数存储的是手持工件的负载。

定义工具的负载参数有两种方法：

如果工具负载数据已知，那么可以在工具坐标系标定页面上选择手动输入方法，并直接输入对应的数据；

如果工具负载数据未知，可以使用 xCore 系统的负载辨识功能获取。

8.12.2.4 负载辨识

负载辨识功能可以方便地计算工具的动力学参数。

负载辨识支持两种方法：一步辨识，两步辨识。工业机器人只支持两步辨识，两步辨识精度通常优于一步辨识。

“两步辨识”操作流程：

1 将机器人切换到自动模式，并执行上电操作；

2 在空载的状态下运行无负载辨识程序，等待辨识程序完成；

3 装上工具负载，运行有负载辨识程序，等待辨识程序完成；

4 辨识完成出现辨识结果弹窗，点击保存。

对协作机器人来说，支持“一步辨识”，操作流程：

1 选择负载辨识方法为一步辨识；

2 装上负载，并确定负载正确安装、且辨识动作无干涉；

3 切换到自动模式并上电；

4 点击带载辨识，运行辨识程序，等待辨识程序完成；

5 确认辨识结果，并保存；

> 提示
>
> 请务必准确定义新工具的动力学参数，否则将影响机器人的运动性能，严重情况下甚至会造成机器人过载而损坏。
>
> 开始辨识之前先将机器人预热半小时以上可提高辨识准确度。
>
> 负载惯量计算基于法兰坐标系。
>
> 只支持机器人正装情况下的负载辨识。

> 提示
>
> 在辨识过程中出现以下情况，将导致辨识过程终止，所有已获得的辨识数据丢失，需要重新辨识：
>
> - 辨识操作进行到中间时，选择了其他工具或切换到其他界面；
> - 辨识程序运行过程中触发急停或者外部安全停止；
> - 辨识程序运行过程中，从自动模式转换到手动模式会导致辨识失败。

> 警告

辨识程序需要在自动模式下执行，因此各项防护措施均应处于有效状态，外部控制信号可能随时启动机器人，因此请在完成安装且人员全部退到安全区域时再切换到自动模式执行。

8.12.2.5 TCP 校正

xCore 提供了 TCP 校正功能来校正工具坐标系 TCP。对于经过位姿标定过的手持工具，使用 TCP 校正功能以提高 TCP 位置精度。对于工具末端发生变形，工具安装位置误差导致的理论 TCP 与实际 TCP 偏差较大的情况，使用 TCP 校正功能可进行工具坐标系 TCP 校准和快速标定。

TCP 校正功能目前支持两种校正类型：XY 校正、Z 校正。分别用于校正工具坐标系 TCP 的 x，y，z 坐标参数。

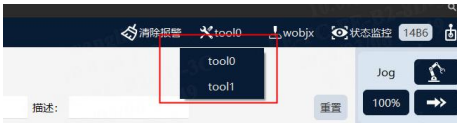
提示

1.TCP 校正功能目前只支持手持工具。
2.机器人正装、侧装、倒装均可以使用 TCP 校正功能。
3.只有经过位姿标定过的工具才可使用 TCP 校正功能。
4.如开启"重力补偿"功能，为保证 TCP 校正的精度，请设置准确的工具负载。
5.TCP 校正功能只开放给 6 轴机器人使用。

8.12.2.6 使用工具坐标系

在 Jog 时使用：

如需要使用特定工具进行 Jog 操作，只需要在示教器界面上方快捷操作栏的"工具"下拉框中选择需要的工具即可。



在 RL 程序中使用：

在程序中使用特定工具非常简单，只需要在运动语句的"工具"参数中使用目标工具即可。使用示教器的"辅助编程"界面编写运动指令时，其默认用的"工具"、"工件"为 tool0、wobj0。

8.12.2.7 外部工具与手持工具

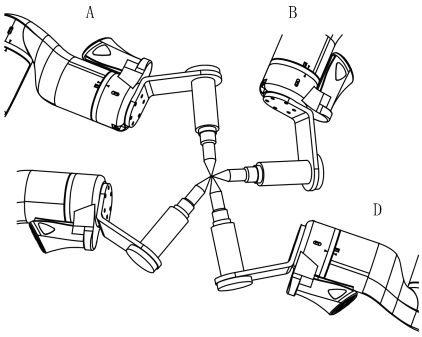
大部分情况下，工具安装在机器人上，工具随机器人运动来完成指定的工作，这样的工具称为手持工具，典型工具包括：抓手、吸盘、焊枪等。

某些特定情况下，把工具安装到机器人上会影响正常使用，例如打磨、涂胶。在这些情况下，把工件安装到机器人上，把工具固定在外部的某个位置会更合适。我们把这些安装在机器人之外，固定在某个位置不动的工具称为外部工具（有些品牌称为 Stationary Tool 或者 Remote TCP），相应的工件称为手持工件。

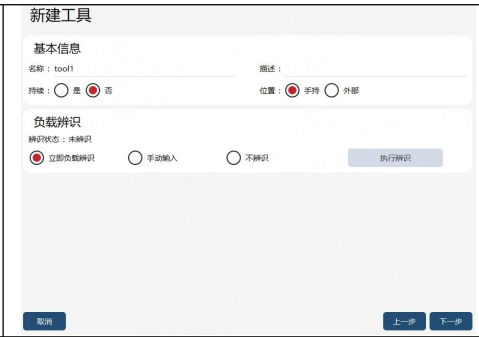
8.12.3 操作示例

8.12.3.1 创建手持工具

标定工具坐标系之前，需要准备一个固定的外部尖端点，要求该点位于机器人工作范围之内，并且使用待标定的工具能以比较灵活的姿态接触到该点。



操作	图示	说明
1.在工具列表界面点击右下角的“+”进入新建工具向导界面。		在“编程员”及以上权限时，可进行新建、编辑操作。

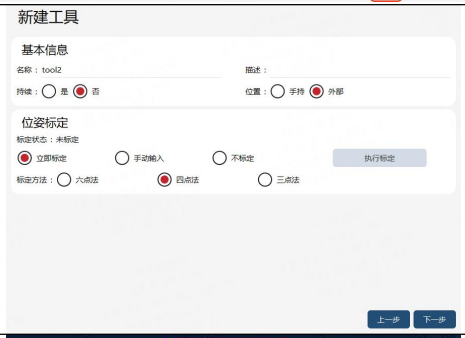
2.设置工具属性：手持。		
3.位姿标定，以三点法为例：选择“立即标定”、标定方法为“三点法”，点击“执行标定”按钮，进入“新建用户坐标系”页面执行标定。		Jog 机器人，使标定好的工具 TCP 按顺序指到期望的坐标系原点/x 轴上一点/xy 平面上一点或 y 轴上一点，点击“确认第一点/第二点、第三点”按钮。 三个点都完成确认后，点击“确认标定”按钮，回到“新建工具”页面，此时“标定状态”为“已标定”。 若使用四点法标定工具原点，应使 4 个点之间的姿态差距应尽可能的大，即机器人应尽量以不同的姿态接触外部尖端点。
4.负载辨识（以手动输入为例）：选择“手动输入”，点击“执行手动输入”按钮，进入“工具负载辨识”页面。		输入工具物理信息，点击“下一步”按钮，回到“新建工具”页面，此时“负载辨识”状态为“已辨识”。
5.点击“下一步”按钮，完成新建工具。		在工具列表中，显示新建的工具。

8.12.3.2 创建外部工具

外部工具的标定方法与手持工具一致，支持 4 点法、6 点法和手动输入三种方式。

注意：标定外部工具坐标系，需要使用已标定好的手持工具来进行。

操作	图示	说明

1.列表中已存在标定好的手持工具，使用它辅助标定外部工具，示例里采用 tool1。		该工具应带尖端。
2.在工具列表界面点击右下角的“+”进入新建工具向导界面。		在“编程员”及以上权限时，可进行新建、编辑操作。
3.设置工具属性：外部		
4.在右上角切换工具，选择已经标定好的手持工具 tool1。		
5.位姿标定，以四点法为例：选择“立即标定”、标定方法为“四点法”，点击“执行标定”按钮，进入“工具标定”页面执行标定。		Jog 机器人，使标定好的工具 TCP 以不同的姿态指到期望的工件坐标系原点上，然后分别确认第一、二、三、四个点。应使 4 个点之间的姿态差距应尽可能的大，即机器人应尽量以不同的姿态接触外部尖端点。 标定完毕后，系统会弹出标定误差，根据误差大小来选择是否需要重新标定（参考确认标定精度）。

注意：

外部工具必须与对应的工件一起使用，即同时选中的工具和工件中的 robhold 参数必须有一个为 False 而另一个为 True，否则系统会给出错误提示，并禁止 Jog 机器人。

使用外部工具时，定义工具坐标系和工件坐标系的参考系与普通工具是不一样的，见下表：

坐标系名称	普通工具时，相对于...定义	外部工具时，相对于...定义
工件坐标系	用户坐标系	用户坐标系
用户坐标系	世界坐标系	法兰坐标系

工具坐标系	法兰坐标系	世界坐标系

8.12.3.3 使用 TCP 校正

使用 TCP 校正功能前，需要准备一个固定的外部尖端点，要求该点位于机器人工作范围之内，并且使用待校正的工具能以比较灵活的姿态接触到该点。

操作	图示	说明
1. 在工具列表界面选择需要校正的工具，点击右下角的"编辑"按钮进入编辑工具向导界面。		在"编程员"及以上权限时，可进行新建、编辑操作。 该工具应带尖端。工具需要经过标定和设置好负载,且是手持工具。
2.以"XY 校正"为例进入位姿标定界面。选择"TCP 校正"按钮，选择校正类型后，点击"执行校正"按钮进入指定类型的 TCP 校正界面。		TCP 校正支持两种校正类型："XY 校正"和"Z 校正"。 "XY 校正"：校正工具 TCP 位置的 x,y 坐标。 "Z 校正"：校正工具 TCP 位置的 z 坐标。
3.调整工具位姿，确认 TCP 校正起始点		Jog 机器人，使工具尖端与外部尖端接触，且机器人法兰与世界坐标系 xy 平面平行，法兰 z 轴指向世界坐标系 z 轴负方向，点击"确认起始点"按钮
4.设置预设旋转角度，运动至第一点		选择预设旋转角度，确保预设旋转角度可到达，之后长按"运动至第一点"按钮至到达目标点。（若预设旋转角度均无法到达，可重新确认起始点）

5.确认 TCP 校正第一点		机器人运动至目标点后，随后点击"确认第一点"按钮。
6.确认 TCP 校正第二点		平移 jog 机器人（仅 jog x，y，z，不可旋转），使工具尖端再次和外部尖端点接触，随后点击"确认第二点"按钮。
7.确认校正，设置校正结果		点击"确认校正"按钮，完成 TCP 校正。界面弹窗显示 TCP 校正结果，点击"确认"按钮，工具 TCP 设置成校正后结果，点击"取消"工具 TCP 保持校正前结果。

8.13 工件列表

8.13.1 概述

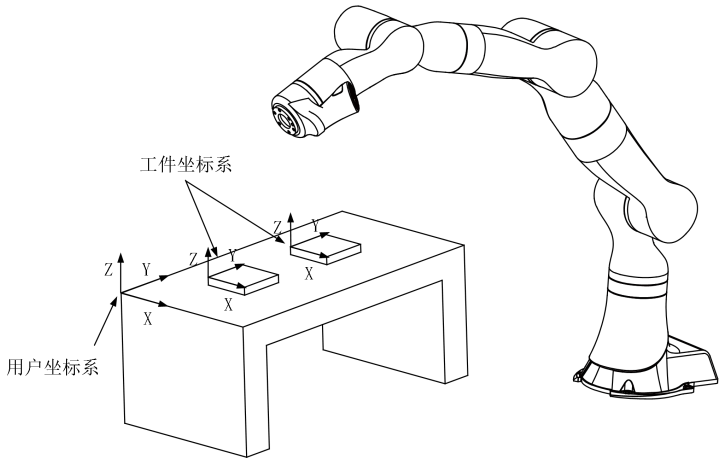
工件是指机器人使用工具进行加工或者处理的物品，在 xCore 系统中使用 wobj（Work Object）类型的变量来描述一个实际的工件。要定义一个工件，就意味着要创建一个 wobj 型的变量。

机器人的运动轨迹都是在工件坐标系下定义的，这样主要有两个好处：

（1）当工件发生移动，或者加工多个相同工件时，只需要重新标定工件坐标系，程序中的所有路径即可随之更新，而不需要重新编写程序；

（2）允许加工被外部轴(如导轨，变位机等)移动的工件；

每一个工件由两个坐标系共同定义：一是用户坐标系，可以理解为摆放工件的工作台/桌子，在处理多个相同工件时非常有用；二是工件坐标系，可以理解为工作台上的工件。机器人的路径点，都是工具相对工件描述的。



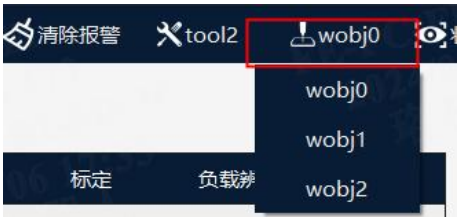
使用外部工具功能，对应的工件将安装在机器人上，称之为手持工件。手持工件同样需要标定工件坐标系，且必须使用已经定义好的外部工具进行标定，更多信息请参考外部工具功能。

> 提示
>
> wobj0 是系统预定义的工件变量，其用户坐标系和工件坐标系都与世界坐标系重合。
> 与 tool0 一样，wobj0 也不允许修改。
> 当机型为 PCB 3、4 轴机器人时，工件坐标系只支持手动输入，姿态 A、C 分量强制为 0，且禁止用户手动修改。

8.13.2 使用工件坐标系

在 Jog 时使用：

如需要在特定的工件坐标系下进行 Jog，只需要在示教器界面上方快捷操作栏的"工件"下拉框中选择需要的工件即可。



在 RL 程序中使用：

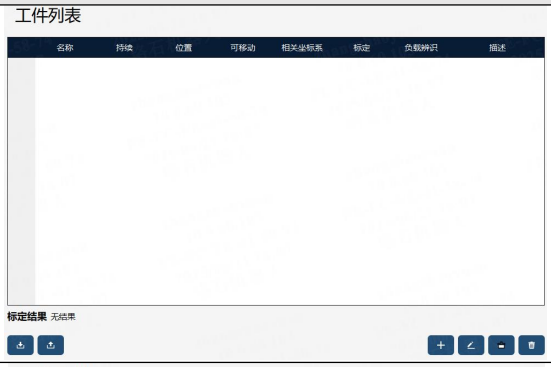
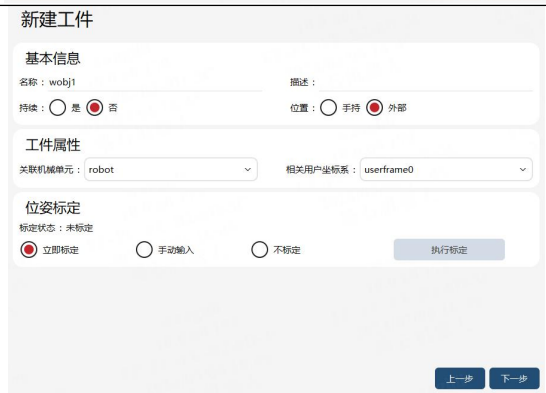
在程序中使用特定工件非常简单，只需要在运动语句的"工件"参数中使用目标工件即可。使用示教器的"辅助编程"界面编写运动指令时，其默认用的"工具"和"工件"与 Jog 时使用的一致，即界面上方快捷操作栏当前选中的"工具"和"工件"，具体操作步骤请参考插入指令。

> 提示
>
> 通常情况下，运动指令的工件参数是可选的，因此如果不特意指定的话，系统会默认使用 wobj0，默认的 wobj0 与世界坐标系重合。
> 如果使用外部工具功能，则必须指定与所用工具配套的工件参数。

8.13.3 操作示例

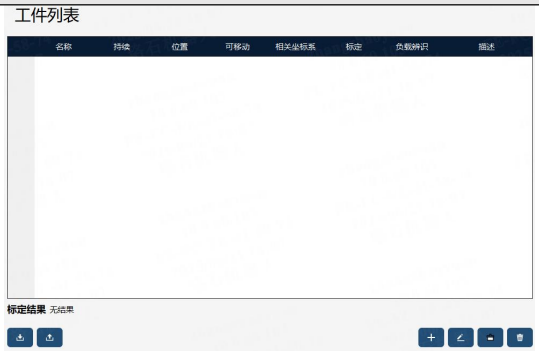
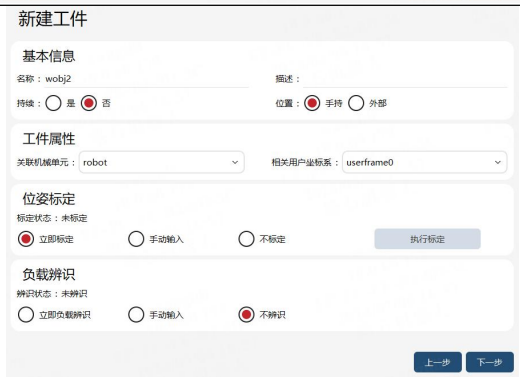
8.13.3.1 创建外部工件

标定外部工件坐标系，需要使用已标定好的手持工具来辅助标定。

操作	图示	说明
1.在工件列表界面点击右下角的“+”进入新建工件向导界面。		在“编程员”及以上权限时，可进行新建、编辑操作
2.设置工件属性：外部。		
3.在右上角切换工具为已经标定好的手持工具tool1		
4.点击“执行标定”按钮，进入工件标定界面。		Jog 机器人，使标定好的工具 TCP 指到期望的工件坐标系原点，点击“确认第一点”按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 X 轴上一点，点击“确认第二点”按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 Y 轴上一点，点击“确认第三点”按钮，点击“确认标定”，弹出标定结果确认对话框，点击“确认。”点击“下一步”按钮，返回“新建工件”页面
5.点击“下一步”按钮，完成新建工件。		在工件列表中，显示新建的工件。

8.13.3.2 创建手持工件

标定手持工件坐标系，需要使用已标定好的外部工具来辅助标定。

操作	图示	说明
1.在工件列表界面点击右下角的"+"进入新建工件向导界面。		在"编程员"及以上权限时，可进行新建、编辑操作。
2.设置工件属性：手持		"外部"为工件与机器人末端不相对固定；"手持"为相对固定。
3.在右上角切换工具为已经标定好的外部工具tool2		
4. 负载辨识同工具负载辨识		
5.点击"执行标定"按钮，进入工件标定界面。		Jog 机器人，使标定好的工具 TCP 指到期望的工件坐标系原点，点击"确认第一点"按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 X 轴上一点，点击"确认第二点"按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 Y 轴上一点或 XOY 平面上一点，点击"确认第三点"按钮，点击"确认标定"，弹出标定结果确认对话框，点击"确认。"点击"下一步"按钮，返回"新建工件"页面

6.点击"下一步"按钮，完成新建工件		在工件列表中，显示新建的工件。

8.14 变量监控选择界面

8.14.1 概述

变量监控选择界面用于添加变量监控需要监控的变量。

①	监控变量过滤，可以筛选需要显示在下方列表中的监控变量。
②	监控变量展示，展示各变量的：名称、类型、原始维度、描述等属性。
③	功能按钮区，导入导入监控变量，导出导出监控变量，批量添加批量添加，单个添加单个添加，删除删除监控变量。

8.14.2 操作示例

8.14.2.1 批量添加监控变量

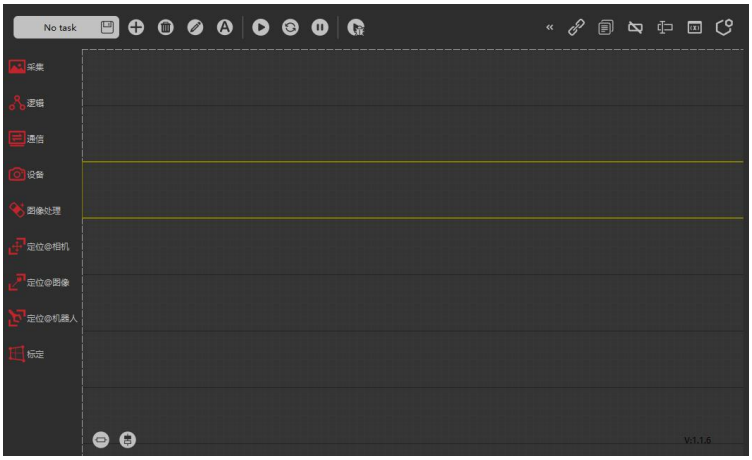
操作	图示	说明
1.点击“批量添加”按钮		
2.选择需要监控的变量，在红框内勾选，然后点击“完成”按钮。		注意：当监控变量为数组时，批量添加后将会监控数组内的全部元素。
3.打开状态监控中的变量监控，查看监控状态。		

8.14.2.2 单个添加监控变量

操作	图示	说明
1.点击"单个添加"按钮		
2.选择需要监控的变量，点击完成，如果所选择的变量为数组，可以通过选择维度来监控指定维度的变量。		只有当变量为数组时，才能操作维度
3.打开状态监控中的变量监控，查看监控状态。		

8.15 视觉

视觉任务编程，可以视为跟 RL 运动任务编程同样级别。通过"编程"-"视觉"，可以打开视觉任务编辑界面。



视觉任务也属于"任务"的一种，单击"任务"，可以查看视觉任务，并进行视觉任务的新建、打开、重命名。

勾选视觉任务"是否运行"属性，该属性在视觉任务中被定义为"开机自启动"功能。在机器人工程中选中的项目在加载后，勾选"是否运行"属性的视觉任务将自动加载并循环运行。

注意：有且只有一个视觉任务可以被勾选。

更多关于视觉功能的使用，请参考手册《xVision 机器人视觉使用手册》。

8.16 关于 RL 程序

8.16.1 RL 程序格式及语法

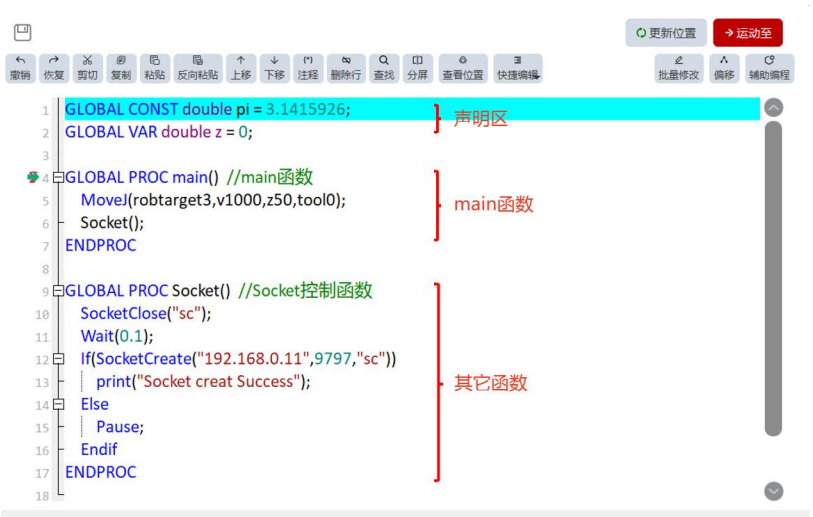
8.16.1.1 概述

RL 语言程序文件后缀名为.mod，mod 是 module（模块）的缩写。例如：MoveObj.mod 或 PickSomething.mod，每一个程序文件构成一个程序模块。

RL 语言的指令不区分大小写，例如对于 MoveAbsJ 、moveabsj、MOVEABSJ 三种写法均可识别，但是为保持统一的语言风格，建议采用单词首字母大写的格式。

8.16.1.2 程序结构

下面为一个简单的 RL 程序：



整个程序分为声明区和实现区两大部分，在每个 Mod 文件中第一个函数之前的区域为声明区，例如 main.mod 中 GLOBAL PROC main 之前的部分为声明区；声明区中可以定义变量或常量。该区域定义的变量将在每次执行程序时被重置为初始值。

VAR 或 CONST 关键字表示存储类型，VAR 表示可变量，CONST 声明为常量；若不显示声明变量的存储类型，默认变量为可变量。

声明区变量、变量列表中的变量存在以下区别：

某个任务运行结束重置时，会重置任务内声明区定义的变量；

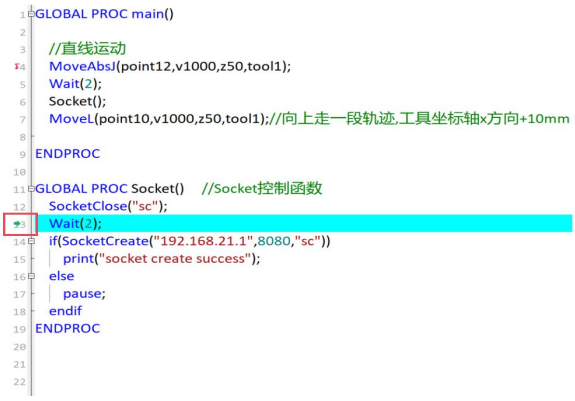
变量列表的变量属于整个工程，是公用变量。变量列表的非保持变量只在执行 pptomain 时重置，保持变量只能通过变量列表编辑功能或者 RL 程序修改。

8.16.2 RL 程序调试

8.16.2.1 程序指针

程序指针指向程序已解析并运行的行号。

在 HMI 界面上，程序指针使用绿色小箭头表示（又称绿指针）。



8.16.2.2 运动指针

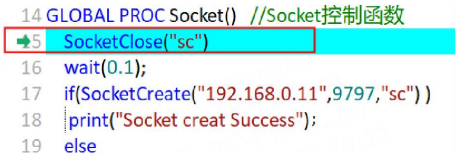
运动指针表示机器人实际正在执行的指令；

在 HMI 界面上，运动指针使用红色箭头表示。

```
GLOBAL PROC main()

  //直线运动
  MoveAbsJ(point12,v1000,z50,tool1);
  Wait(2);
  Socket();
  MoveL(point10,v1000,z50,tool1);//向上走一段轨迹,工具坐标轴x方向+10mm

ENDPROC

GLOBAL PROC Socket()    //Socket控制函数
  SocketClose("sc");
  Wait(2);
  if(SocketCreate("192.168.21.1",8080,"sc"))
    print("socket create success");
  else
    pause;
  endif
ENDPROC
```

8.16.2.3 移动程序指针

如果需要从程序中间的某一行开始执行之后的程序，那么可以使用该功能将程序指针移动到光标所在行，之后程序即可从新的位置开始执行。

操作	图示	说明
1.点击右侧操作面板中的“停止”按钮，使 RL 程序暂停。		
2.点击期望指针移动到的行，使其为选中状态。		选中状态下背景颜色为浅蓝色。
3.点击程序调试快速定位按钮旁的下拉箭头，从列表中选择“光标”。		选择后该按钮显示“光标”字样。
4.点击程序调试快速定位按钮，即可移动指针到选中行。		

使用限制：

1. 使用该功能时，以下指令会被忽略，编译器的编译位置会直接移动到目标行，除此之外，其他所有指令都不执行：
   - 所有的运动指令；
   - SetDO、SetGO、Return、Wait、Print 以及所有的 Socket 指令；
   - 函数调用行；
2. 移动程序指针时，忽略流程控制指令的判断条件。

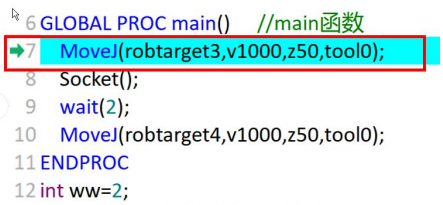
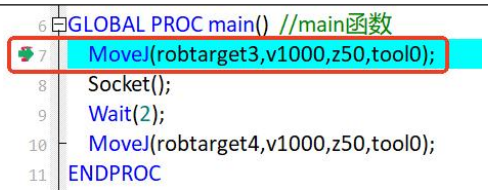
3. 不能跨函数移动程序指针，需要先使用“程序指针到函数”功能将程序指针移动到某个函数的开头，再使用移动程序指针功能。
4. 移动程序指针只能移动到运动指令行。

8.16.2.4 单步调试

单步运行状态又称为单步模式，与之对应的是连续模式，绝大多数情况下，机器人可以在单步模式和连续模式之间进行切换。

单步运行主要用来做程序调试，机器人每次单步运行将尽可能执行完一行指令，并在指令完成后暂停程序，便于确认每一个行的示教点是否符合要求。当调试对象是多任务工程时，单步调试只会执行当前 HMI 调试界面显示的任务，其余任务将不会被调用。

如果单步调试执行的是读取数据指令（ReadDouble，ReadString 等）、时间相关指令（Wait、WaitUntil 等）、逻辑指令（IF、GOTO 等），由于指令特性限制，将需要两到三次点击下一步，才能完成该行指令。

操作	图示	说明
1.移动指针到某行。		
2.点击右侧操作面板中的下一步按钮。		
3.再次点击下一步按钮，可看到机器人正在执行该行。执行完成后再次点击下一步按钮，开始执行下一步。		若遇到函数，将跳转到该函数内部，可通过程序指针（绿色箭头）定位正在执行的行。

使用限制：

1、连续模式程序自动执行，需要处理转弯区，可以有运动前瞻。
2、单步模式直接执行指令，不处理转弯区，无运动前瞻。
3、连续模式下，前瞻点够了才运动，到位后才可以继续解析指令。
4、单步模式下，所有下一步信号由界面触发，不处理转弯区，不前瞻。
5、单步模式下，运动过程中点击下一步，不响应。
6、连续模式下，运动过程中的回调，按照前瞻逻辑响应。
7、下一步可到任意行，执行字面意义。对于 RL 程序而言，只有“程序指令”，不区分运动指令和逻辑指令。
8、连续运行，暂停在转弯区上，下一步均先回到当前转弯区所对应的目标点。

8.16.2.5 步退调试

步退调试又称为上一步调试，主要功能是在调试过程中发现路径错误时，可以直接回退到最后一个正确的点位，而不用通过多次 JOG 才能让机器退出错误轨迹，提高调试效率。

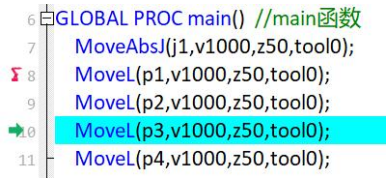
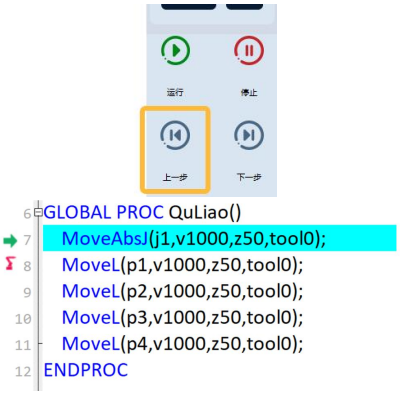
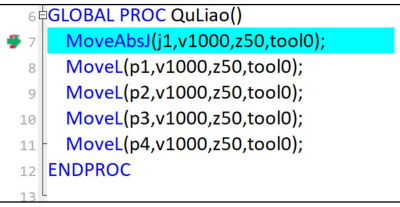
非运动指令在步退过程中一搬不生效，会被直接跳过

力控指令、运动设置指令、逻辑指令会停止步退。

8.16.2.6 步退使用方式

xCore 支持暂停后直接切换步退和 ppto 光标后步退

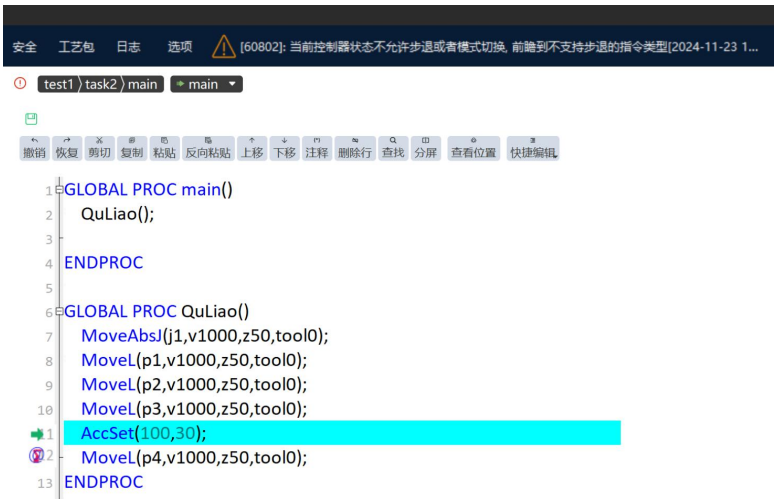
ppto 光标后第一次步退运动时，由于没有任何轨迹信息，笛卡尔点将强转为 MoveJ 运动。

操作	图示	说明
1.运行过程中暂停。		前瞻到第十行，但是机器人在从 J1 前往 P1 的轨迹上
2.点击右侧操作面板中的上一步按钮。		前瞻指针会跳转到运动指针上面，表示进入步退模式
3.再次点击上一步按钮，机器人开始步退运动。		机器人会按暂停时的运动方式（本例中是 MoveL）回退到前瞻指针指向的 J1

8.16.2.7 步退使用限制

步退功能主要用于点位的调试，所以大部份非运动指令在步退模式中是无效的或者被限制使用的，也有少部分运动指令原理上无法步退，也限制使用，如果尝试步退执行受限制的指令，控制器会报错。

例：



步退调试功能支持的指令情况：

类型	具体指令
支持步退的运动指令	MoveAbsJ, MoveJ, MoveL, MoveC, Home
特殊运动指令	SearchL, SearchC, TrigL, TrigJ, TrigC (注 1)
不支持的运动指令	MoveSP, MoveT, MoveCF
终止步退的非运动指令	传送带指令，力控指令,逻辑指令，函数调用，高级指令（注 2）
被跳过的非运动指令	剩余的所有指令

注 1： 探寻、Trig 等功能在步退时不生效，只产生运动效果。

注 2：GetRobotState，BreakLookAhead，GetRobotMaxLoad 指令会被跳过，其它处于该分类的指令都会终止步退。

8.16.2.8 重回路径

在某些特定情况下，机器人的位置会偏离其编程的路径，例如：

在程序运行中被停止的期间（程序复位造成的程序停止不在此列），机器人被 Jog 到其他位置；

程序运行期时触发紧急停止，机器人执行 STOP 0 后；

当程序再次从停止位置启动时，如果系统检测到机器人已经偏离编程路径，那么机器人会首先执行重回路径（Regain Path）运动以返回到原来的编程路径上。

为保证安全，重回路径时机器人的运动速度比较慢，并且可随时按示教器上的"停止"按钮来停止机器人的运动。

使用限制

1、重回路径时机器人执行的是关节轨迹，因此末端路径无法预测，请注意观察是否会与周边环境发生碰撞。

2、只有当机器人从程序中间停止的地方继续执行时，控制系统才会检测是否偏离路径，如果偏离则会执行重回路径操作。

3、如果程序被复位，那么系统将不会检测是否偏离路径，而是直接从第一行开始执行，请注意防范可能发生的碰撞。

8.16.2.9 循环模式

点击"循环模式"按钮切换至 循环模式，或 单次模式。

```
GLOBAL VAR double z = 0;

GLOBAL PROC main()    //main函数
    MoveJ(robtarget3,v1000,z50,tool0);
    Socket();
    Wait(2);
    MoveJ(robtarget4,v1000,z50,tool0);
ENDPROC

int ww=2;

GLOBAL PROC Socket()    //Socket控制函数
    SocketClose("sc");
    Wait(0.1);
    if(SocketCreate("192.168.0.11",9797,"sc"))
        print("socket create success");
    else
        pause;
    endif
ENDPROC
```

循环模式：

所有任务在执行到指定函数（默认为 main 函数）的 endproc 时，会在 0.5 秒之后重置，重置之后从最后一次指定的函数的第一行重新开始执行。

PPToMain 操作将 main 函数作为指定函数；

PPToFunc 将跳转的目标函数作为指定函数；

PPToLine 不会影响下次循环的函数执行目标。

单次模式：

所有任务（半静态任务除外）将在执行到指定函数(默认 Main 函数)的 endproc 后永久停止，直到下一次开始运行，重新加载工程。

8.16.2.10 前瞻机制

8.16.2.10.1基本概念

前瞻机制无法关闭，系统在运行程序时会自动进行前瞻，可以使用程序指针（Program Pointer）来查看前瞻的位置。 从前瞻的角度看，RL 指令分为 4 类：运动指令、不停前瞻指令、转弯区执行指令、停止前瞻指令。

8.16.2.10.2 运动指令

控制机器人产生实际运动效果的指令，详情见 RL 编程指令，其中所有"运动指令"、"Trigger 指令"、"拖动回放指令"以及"Home"指令属于运动指令分类。

8.16.2.10.3 不停前瞻指令

指令在前瞻指针解析后立即执行，随后前瞻指针继续向下运行解析下一条指令，不会影响前后两条运动指令之间的转弯区。

不停前瞻指令：Print 指令，逻辑判断指令，变量赋值操作，用户自定义函数，碰撞检测动态阈

值指令，动态修改运动参数指令；

示例：

```
MoveL (p1);
IF (condition_1)
   Print("满足条件 1");
   MoveL (p2);
ENDIF
MoveL (p3);
```

运行上面程序，在条件 1 满足的情况下，机器人会规划 p1 > p2 > p3 的连续轨迹运动，并在前瞻到 print 指令时打印字符串"满足条件 1"。

在条件 1 不满足的情况下，机器人会规划 p1 > p3 的连续轨迹运动。

8.16.2.10.4 转弯区执行指令

转弯区指令的上一条运动指令进入转弯区或者结束时，则执行该指令。

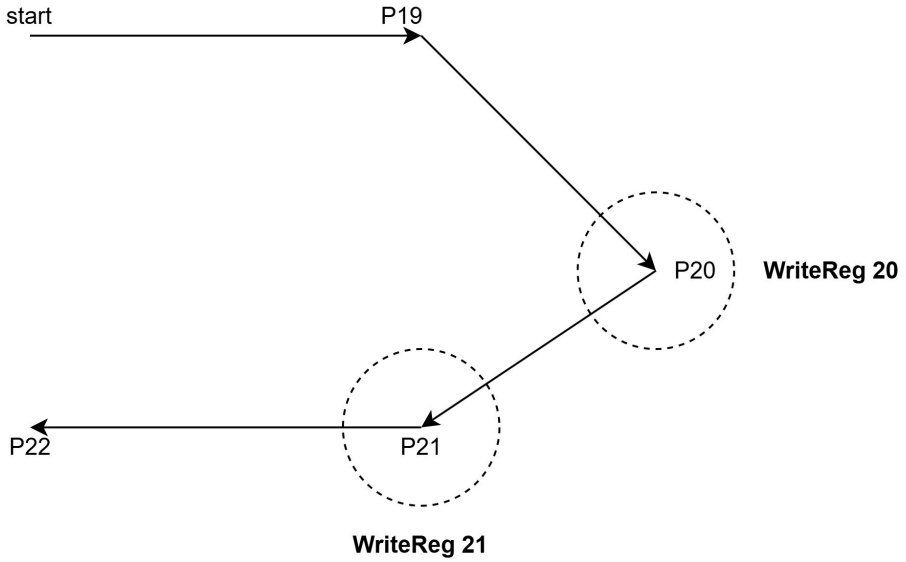
用于在运动的过程中向外部设备发送信号，说明机器人已经运动到哪条运动指令。

转弯区执行指令：WriteRegByName, SetDO, SetGO, SetAO, PulseDO, PulseReg, InZone，SetVarValue, SpeedRefrsh。

示例：

```
MoveL P19
MoveL P20
WriteRegByName (reg_position, 20);
MoveL P21
WriteRegByName (reg_position, 21);
MoveL P22
```

运行上述运动指令，机器人将规划一条 P19 > P20 > P21 > P22 的连续运动轨迹，当机器人即将完成 P20 运动，在 P20 往 P21 转弯区上时，会立即将数 20 写到寄存器 reg_position；当机器人即将完成 P21 运动，在 P21 往 P22 转弯区上时，会立即将数 21 写到寄存器 reg_position；外部设备只要读取这个寄存器，就可以知道机器人的运动过程。



提示

转弯区指令的触发时机受机器人本体性能、转弯区大小、转弯区是否生成、实际运行速度等多方面影响，适用于只关注机器人正准备运动到哪些位置的场景，如果要求精确在轨迹某个位置触发信号，需要使用 trigger 系列指令。
如果没有转弯区或者无法生成转弯区，会在对应运动指令完成后触发相关指令。

8.16.2.10.5 停前瞻指令

除了上述三种指令，其他指令都是停前瞻指令，控制器会在机器人完成该指令前的所有运动之后，再执行指令。
示例：

```
MoveL (P1);
MoveL (P2);
Wait (10);
MoveL (P3);
```

Wait 指令属于停前瞻指令，机器人会运动到 P2 并且减速停止完成后，再开始等待十秒，并在等待完成后，再启动前往 P3。

当运动指令使用转弯区为 Fine 时，RL 程序的前瞻指针会在该行运动指令停止，等待运动结束后，再继续前瞻
示例：

```
MoveAbsJ (j1,v1000,z50,tool0);
MoveL (p1,v1000,z50,tool0);
MoveL (p2,v1000,Fine,tool0);
// 后续所有指令都会等 MoveL P2 结束才运行
MoveL (p3,v1000,z50,tool0);
```

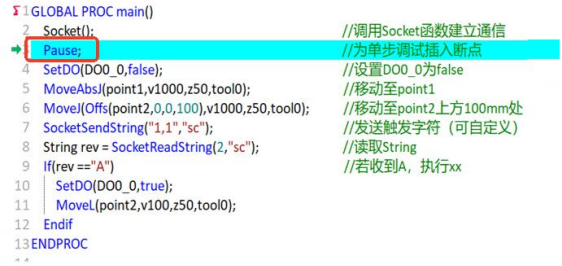
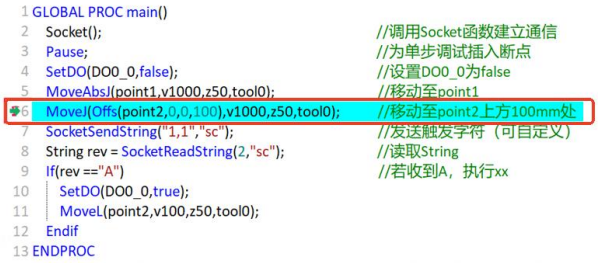
8.16.2.11 中断功能

```
GLOBAL PROC main()
    IRegister(intnum0,DI3_1,\Posflank,"aaa");
While(true)
    PathRecStart();
    MoveL(p1,v1000,z50,tool0,wobj0);
    MoveL(p2,v1000,z50,tool0,wobj0);
    MoveL(p3,v1000,z50,tool0,wobj0);
    MoveL(p4,v1000,z50,tool0,wobj0);
    MoveL(p5,v1000,z50,tool0,wobj0);
    PathRecStop();
Endwhile
ENDPROC

TRAP aaa
    bool0=GetRecStartStatus();
    If(bool0==true)
        PathRecBwd();
    Else
        Print("未在开启录制时触发中断，程序将暂停");
        pause;
    Endif
    //可插入运动指令（moveL,moveC,moveJ,moveabsj）和其他非运动指令
    PathRecFwd();
ENDTRAP
```

中断功能可由 DI 信号或寄存器触发，打断当前程序运行而去执行中断函数。中断功能的一些特性和注意事项如下：

1. 中断功能可中断运动指令，可中断持续性指令（网络/通信 读指令、接收连接指令、SocketCreate、opendev、SocketClose、CloseDev、SendString、SendByte），可中断普通指令（类型转换、字符串操作等立即完成的指令）。
2. 运动指令执行时发生中断，会停止当前运动，转而执行中断函数，中断函数内部指令完成后重回路径到中断前的路径上继续执行。中断函数完成后的重回路径不属于中断。重回路径走轴空间运动。中断函数建议配合记录路径指令以保证机器人可以正确回到安全位置。
3. 除了 waituntil 外，其他持续性指令并不是发生真的中断，而是将持续性指令转入后台运行，该指令的错误处理和唤醒任务执行 都将延后到中断结束。中断时执行 wait 指令，中断结束后已超过 wait 等待时间就继续运行。
4. 中断中寄存器-读指令、外部信号-控制指令效果与连续运动相同，响应寄存器-写、外部信号-监控指令。暂停、急停、碰撞可停止中断运行，中断中可单步运行。
5. 中断相关指令只能在运动任务使用，仅运动任务可执行中断。
6. 进入力控模式不可使用中断，注册中断后不能 FcInit 进入力控模式。
7. 选择单次触发的中断，在被 IDisable 后 收到中断信号，仍视为单次触发结束。
8. 在 IRegister 中断之后，中断可以触发，当任务结束重置时注册的中断会被取消，直到再次运行到 IRegister。
9. 中断执行中，此时触发的中断不会被响应。
10. 任务正在执行中断、由于点击开始运行/中断退出切换正常运行导致调度信息重置时，不响应中断、不停止任务。
11. 中断函数中不能调用除 GetTrapData 外的中断指令、料盘码垛的设置指令;激光焊接功能开启不能中断;传送带启动指令执行后，不能中断;末端运动指令执行不能被中断控制，使用时需要注意。
12. 拖动回放时发生中断，中断后不会继续执行拖动回放。

8.16.3 调试示例

操作	图示	说明
1.RL 程序编辑完成后，使用 pptomain，将指针移动至 main 函数，同时检查程序有无语法错误。		为了方便单步调试，可在程序中插入 pause 指令。
2.切换模式至"手动模式"，机器人上电，调节程序速度到较小值，使机器人低速运行。		
3.点击右侧操作面板中的运行按钮。		
4.程序运行到 pause 将暂停运行，直至点击"下一步"按钮或"运行"按钮。		"下一步"按钮将使程序单步执行
5.点击下一步按钮，开始单步调试。		程序运行到 SetDO 行，打开状态监控，查看该 DO 是否为"关"状态
6.继续单步执行，查看机器人是否到达期望的点位。		

7.也可移动光标至某一行，点击"下一步"按钮，查看指令是否运行成功。		程序运行到 SetDO 行，打开状态监控，查看该 DO 是否为"开"状态。
8.确认程序没有问题后，再次 pptomain，保持"程序速度"较小，完整运行一次之后，可增加程序速度，切换到"自动模式"执行程序。		

9 设置

9.1 本章简介

本章详细介绍 xCore 控制系统各种设置项。

9.2 控制器设置

9.2.1 基本设置

9.2.1.1 系统信息

①	版本：控制系统版本信息；
②	机器人类型：机器人机型信息；
③	MAC 地址：本机 MAC 地址；
④	重启：注意，重启前保存所有配置信息； 关机：需要控制柜断电后重新上电才能重新启动控制器软件。

9.2.1.2 系统配置

①	机器人类型：选择机器人型号；
②	安全板：选择安全板类型；
③	控制柜类型：选择控制柜类型；

注意：系统配置，用户请勿修改；特殊情况下，在厂家指导下进行调整。

9.2.1.3 系统时间

系统时间为日志等功能提供时间参考。

①	时间值：控制器时间；//不会实时刷新
②	获取控制器时间：点击后，刷新并显示控制器时间，用户可以确认该值是否合理；
③	修改为当前时间：将控制器系统时间更新为 RobotAssist 软件所在设备当前的系统时间，无须再点击"确定"。
④	用户可以在①手动修改时间，点击"确定"后，设置控制器时间；

> ⚠ 警告
>
> 1、系统时间是日志信息的绝对时间标准，请勿随意修改，错误的修改系统时间会导致用户

无法根据日志追溯相关事件发生的时刻。 2、该操作不宜频繁执行，“获取控制器时间”、“修改为当前时间”这两个操作或者同操作本身的执行间隔需要大于 5 秒。

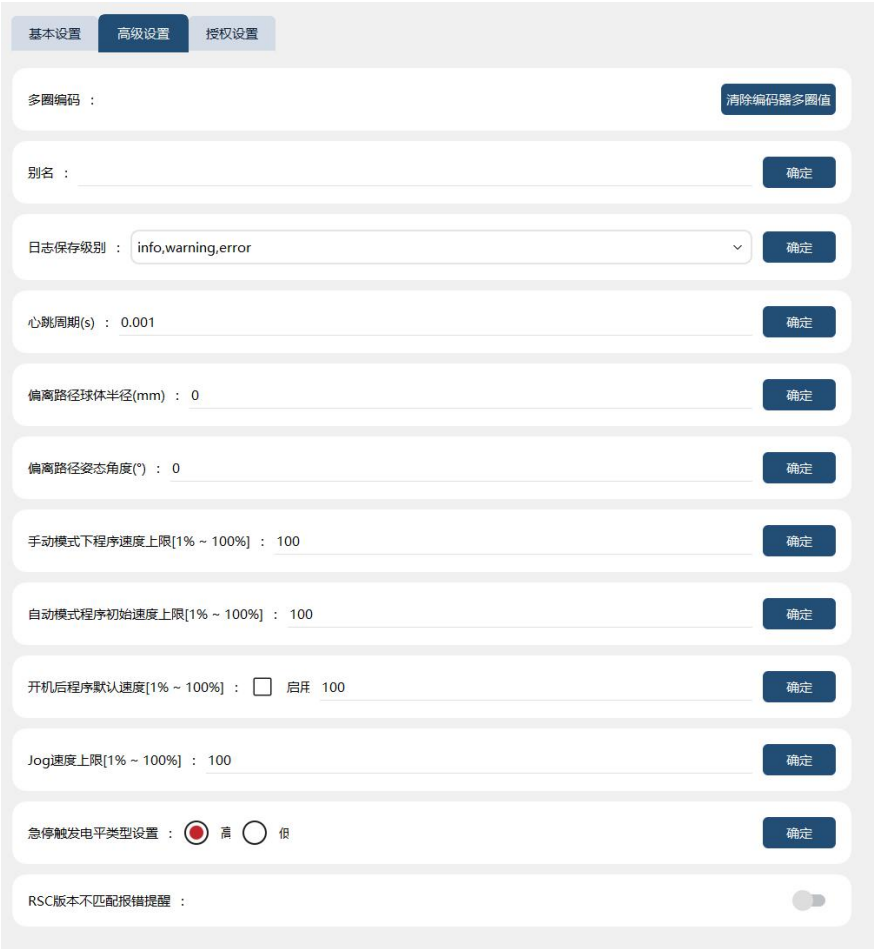
9.2.1.4 系统 IP 属性

通过该页面，对机器人外网口的连接模式、IP 和子网掩码等进行配置。

①	名称：显示自动获取的网卡名称；
②	模式：手动(可以改 IP)或自动(自动分配 IP)；
③	IP、网关、子网掩码、DNS 服务器；

注意：调试网口的 IP 只能修改为 192.168.0.160 或 169.254.160.160。并且当调试网口的 IP 为 192.168.0.160 时，其它网口的 IP 不可以修改为 192.168.0 网段；当调试网口的 IP 为 169.254.160.160 时，其他网口的 IP 可以修改为 192.168.0 网段，但不能修改为 192.168.0.160。

9.2.2 高级设置



多圈编码	清除编码器多圈值。谨慎操作！
别名	对控制器设置别名，方便在同一片局域网内存在多台机器人设备时进行区分。别名会显示在"选项"->"连接"->"机器人探测"界面上。
日志保存级别	日志级别从低到高分为"信息"、"警告"、"错误"三个级别，对应显示为"info"、"warning"和"error"。可以设置从某个级别开始进行日志保留，这样低于设置级别的日志将仅进行在线展示，不会进行历史留存。
心跳周期	"一个高电平+一个低电平"构成一个心跳周期，高、低电平持续时间相等。心跳信号，可以通过系统 IO 与 DO 信号绑定(详情参考系统 IO 章节)，或者通过"sta_heartbeat"功能码与寄存器绑定(详情参考寄存器章节)。
偏离路径球体半径	单位：mm。 范围：[0, 9999] 含义：假设其值 x，当机器人暂停在 A 点处，以 A 点为球心，以 x 为半径，会形成一个球体，当机器人当前位置坐标与 A 点欧氏距离小于偏离路径球体半径 x 时，则认为满足偏离路径球体半径条件。该参数与功能码"sta_near_path"配合使用，详情参考寄存器章节。
偏离路径姿态角度	单位：角度°。 范围：[0°, 180°] 含义：假设其值为 a°，当机器人暂停在 B 点处，以 B 点的姿态为基准，机器人当前位置和 B 点两个坐标系作比较，夹角小于 a 度，则认为满足偏离路径球体角度条件。该参数与功能码"sta_near_path"配合使用，详情参考寄存器章节。
手动模式下程序速度上限	该参数限制了手动模式下，通过程序速度滑块、以及"-/+"微调按钮调节程序速度时，程序速度可以设定的上限值。 举例：将该值设置为 20%后，假如当前处于自动模式、且程序速度为 40%时，工作模式从自动模式切换至手动模式，程序速度将被自动置为 20%。当处于自动模式时，程序速度不受此参数影响。 注意： 1.通过寄存器功能码（ctrl_set_program_speed）或 SDK 设置程序速度时，如果机器人处于手动模式，实际程序速度会受此参数限制；如果机器人处于自动模式，实际程序速度为寄存器功能码或 SDK 的设置值。 2.自动模式切换到手动模式时(切换方法包括：HMI 上的模式切换按钮，寄存器功能码、系统 IO、外部通信、SDK 等)，程序速度会受此参数限制。
自动模式程序初始速度上限	通过该参数，可以让自动模式下程序的运行，以一个比较低的速度（该参数设置的数值）开始，增强运行的安全性。开始运行后，用户可根据实际情况调整程序速度。 举例：该参数设置为 5%；如果当前程序速度为 60%，自动模式下运行程序时，程序速度将自动调节为 5%；如果当前程序速度为 3%，自动模式下运行程序时，程序速度还保持为 3%。
开机后程序默认速度	启用改选项后，控制器会在重启时将程序运行速度设置为该选项配置的默认速度，保证每次重启后控制器的运行速度是一个确定的值。该设置无法突破手动模式速度上限以及自动模式速度上限，如果设置的值超出上限值，会自动降为上限。
Jog 速度上限	机器人 Jog 速度无法调整为大于该上限的值。

急停触发电平类型设置	通过该设置，确定急停触发后，寄存器、系统输出、外部通信与急停状态相关的输出\状态位，是"高"还是"低"。

9.2.3 授权设置

9.2.3.1 EtherCAT 授权

通过授权码对 EtherCAT 通信授权。

> 提示
>
> 当授权码失效或者授权失败，将无法让机器人上电。

9.2.3.2 功能授权

xCore 系统支持部分选配的软件功能，分为两种类型：

1. 语言类：日语、韩语、俄语、英语；
2. 工艺包类：光伏插片工艺包、光伏排版工艺包、激光焊接工艺包、 SDK 二次开发接口；

如果您需要开放选配功能，可以根据要开通的功能列表购买授权码文件。

注意：授权信息与机器人的控制器绑定，一机一文件。

激活方法：在"设置"-"控制器设置"-"授权设置"-"功能授权"中，点击"选择文件"，选择授权文件进行授权。若授权成功，根据界面提示重启 RobotAssist，即可开启相应的可选功能。

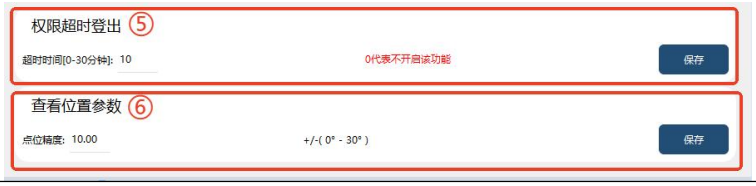
授权有效性：目前功能授权为授权一次后即可一直使用，控制器版本升级、抹除配置、恢复出厂设置等操作均不影响授权状态。

已授权功能：显示当前机器人已授权过的功能。

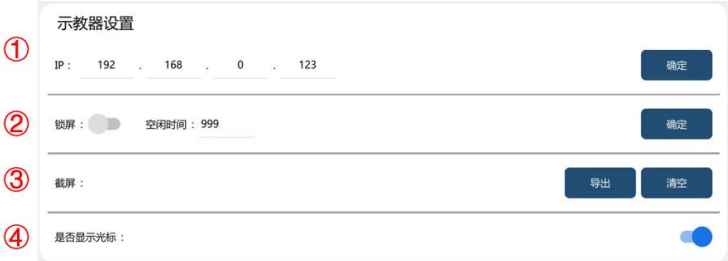
关联内容："HMI 设置"-"基本设置"-"语言"中，仅显示已授权语言。

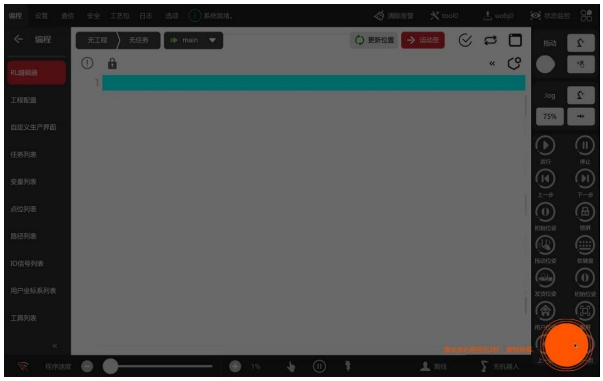
9.3 HMI 设置

9.3.1 基本设置



①	语言：目前支持中、英、日、韩、俄。 注意： 1.中文为默认语言，其他语言需要联系珞石官方授权启用； 2.对于 xCore V1.7 之后的版本，支持多语言日志功能（仅对"控制器日志"生效）。当 HMI 切换语言类型时，控制器也会跟随 HMI 的语言内容进行切换；当 HMI 和控制器语言设置不同时，控制器将在连接建立时切换到 HMI 对应的语言模式；控制器切换语言后不需要重启，该模式属于热切换，但是部分详细日志信息仅在切换后生效，切换语言前生成的日志可能无法切换为对应的语言。
②	绑定 IP 地址："选项-连接"的机器人探测功能，可以在此设置是否通过固定的 PC 端 IP，执行探测。
③	刷新间隔：状态监控的 3D 显示页面刷新频率；上限是 100Hz，默认 18Hz。
④	关闭 3D 显示：关闭状态监控的 3D 显示，重启生效。
⑤	权限超时登出：超过一定时间后，已登录的管理员权限自动登出；默认 10 分钟，设置为 0 表示不开启此功能。
⑥	查看位置参数：RL 编程界面查看位置功能的点位精度参数，允许模糊查询的角度范围，默认 1°。



①	IP（仅适配 xPad2）:设置示教器和机器人连接网卡的静态 IP。
②	锁屏（仅适配 xPad2）：可以通过该功能减少无关人员误触示教器。开启该功能后，一段时间(空闲时间)不操作示教器，或者点击锁屏快捷键后，示教器进入锁屏状态，类似下图：  通过长按锁屏页面圆形按钮 3 秒、或者再次短按锁屏快捷键，可以解锁屏幕。
③	截屏（仅适配 xPad2）：截取当前示教器屏幕，存放在示教器目录中，图片格式 JPG。
④	是否显示光标：示教器光标有无设置。

①	选择主题：选择不同的显示风格；
②	主题大小：调整控件、字体等大小；
③	工作区路径：设置工程文件存储的位置；
④	权限超时登出：当超过设定的超时时间没有操作界面时，用户权限切换到操作员
⑤	点位精度：设置点位精度

9.3.2 示教器模式

机器人开机使用过程中，有可能需要断开 xPad2 示教器与机器人间的物理连接。如果直接断开物理连接，将会导致机器人进入急停状态；如果想不影响机器人正常运行，可按照以下步骤操作。

步骤	图示	说明
1.在该页面查看示教器当前模式，默认为"有示教器模式"，此时单击"切换模式"按钮，将尝试将模式切换为"无示教器模式"，切换成功后，界面显示模式为"无示教器模式"。		● 有示教器模式，示教器图片为彩色，接头为连接状态； ● 无示教器模式，示教器图片为灰色，接头为断开状态；
2.切换为"无示教器模式"后，可断开示教器 xPad2		此时机器人将不会进入急停状态。

与机器人的物理连接。（参考"机器人系统构成及连接"章节。）		

如果需要重新连接示教器 xPad2 至机器人，按照以下步骤操作即可：

步骤	图示	说明
1.示教器 xPad2 与机器人建立物理连接。（参考"机器人系统构成及连接"章节。）		
2.当前页面显示的示教器模式应该为"无示教器模式"，单击"切换模式"按钮，将尝试将模式切换为"有示教器模式"，切换成功后，界面显示模式为"有示教器模式"。		● 有示教器模式，示教器图片为彩色，接头为连接状态； ● 无示教器模式，示教器图片为灰色，接头为断开状态；

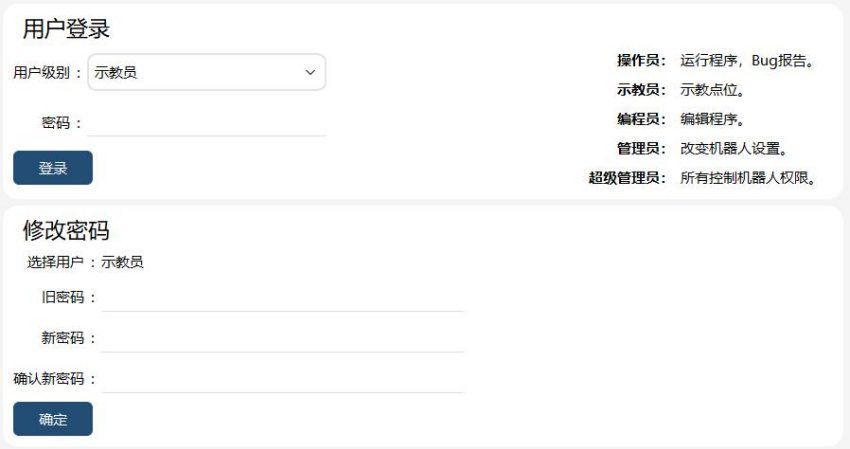
提示

1.热插拔示教器的功能仅支持部分机型，对于不支持此功能的机型将提示示教器模式切换失败，具体机型配置建议咨询技术支持。

2.示教器模式仅建议在有移除示教器需求的时候开启，平时调试及使用过程中，不建议随意开启。

9.4 用户组

xCore 系统内置了五个级别的用户，根据操作权限从低到高分别是操作员（Operator）、示教员（Teacher），编程员（Programmer），管理员（Admin），超级管理员（System）。连接控制器后，默认登录用户为操作员；切换到其他权限用户需要输入密码。



用户	默认密码	权限简述
操作员	无密码	运行程序、bug 报告
示教员	123	示教点位
编程员	1234	编辑程序

管理员	12345	改变机器人设置
超级管理员	123456	所有控制机器人权限

注意：

- 高权限的用户可以修改相同或更低级别用户的密码。
- 操作员级别用户密码不能修改。
- 从高级别用户切换至低级别用户，无需输入密码。

各用户组权限详情参考附录。

9.5 标定

xCore 系统提供了机器人标定功能，包括机械零点标定、软标定(工业机器人)、力传感器零点标定（协作机器人）、基坐标系标定。标定功能可以执行一键标定，也可以执行单轴标定。

9.5.1 零点标定

此处的零点标定，指的是机械零点标定，目的是让机器人理论零点与实际机械零点重合。机器人本体上预设了零点刻线，各关节对准，也就是回到机械零点后，即可执行标定。

为防止用户误操作零点标定导致零点丢失，在点击各轴对应"标定"按钮或"一键标定"按钮后，需输入验证码并点击确定使标定操作生效。

> **警告**
>
> 机械零点请勿随意标定，标定前请使用机械标定块确认机器人各关节都处于零点。
> 机器人出厂经过激光跟踪仪标定之后，机器人不可进行机械零点标定，否则，激光跟踪仪标定后的零点数据会丢失，影响机器人精度。当机器人零点丢失时，请联系机器人厂商尝试零点数据还原。

在某些空间受限制的场景，机器人可能无法回到机械零点，此时可以使用"角度设定"功能。该功能使用的前提是，知道标定时刻机器人的关节角度，将其输入"角度设定"，再标定，可以达到与在零点位置标定相同的效果。

示例：

下面以 xMate7 Pro 机器人为例，假设在 4 轴空间之上有障碍物，机器人无法达到机械零点竖直状态，而又需要进行零点标定，可以先通过 jog，将机器人调整为下图所示直角状态。此时，4 轴 90 度。然后在"角度设定"中，输入当前对应的角度信息(4 轴 90 度，其余是 0 度)后进行。

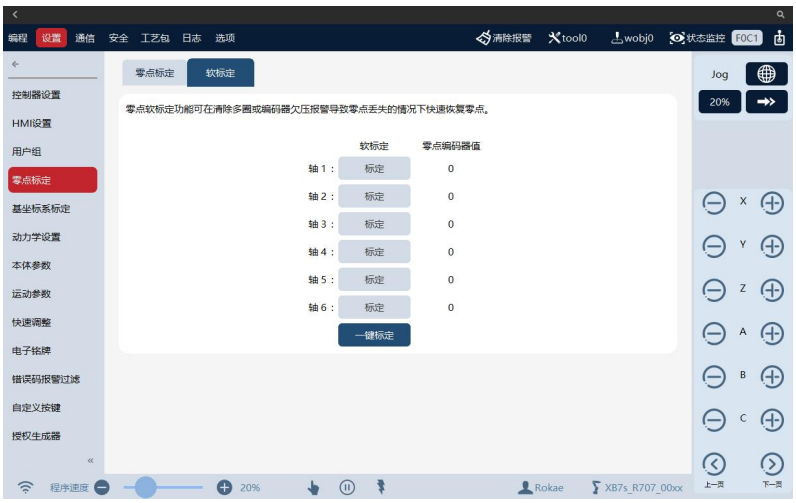
注意：在上面这个例子中，虽然是不同姿态标定，但机器人的零点都是竖直状态，因此在输入 4 轴 90 度标定成功后，如果直接使用 "快速调整至零点"功能，则机器人还是会往机械零点竖直状态运动，与障碍物碰撞！所以一定要注意，该功能标定的是零点，而不是指零点为当前角度。

9.5.2 软标定

软标定功能是指机器人由于编码器电池欠压、拆卸电池或误触清除多圈等非正常操作引发零点丢失后，快速找回零点的功能。

在使用该功能之前，需先手动 jog 机器人至零点（宽窄标定槽对齐，且窄槽完全位于宽槽内），本功能并不能在任意角度恢复零点。

软标定功能对工业机器人，协作机器人都支持



如上图所示，软标定操作步骤如下：

1. 手动将机器人 jog 至零点后（宽窄标定槽对齐，且窄槽完全位于宽槽内）；
2. 进入"主菜单-设置-零点标定-软标定"界面；
3. 点击"一键标定"按钮，并确认弹窗提示，即可找回零点，零点编码器值显示各轴零点的编码器值；
4. 点击各轴对应的"标定"按钮，并确认弹窗提示，可找回单轴零点。

注意：如果是通过角度标定方式标定零点的，需要先 jog 到角度标定值的附近，再执行软标定找回零点。

9.5.3 力传感器标定

该标定针对 xMate 系列协作机器人。机器人长期使用过程中，力矩传感器不可避免可能产生零点漂移，表现为机器人拖动发飘等问题。

出现类似问题后，可以将机器人调整至零点位置，执行单轴标定或一键标定。

如果希望在任意位置进行力传感器标定，可以开启"动态标定"开关，再执行单轴标定或一键标定，这种方式的标定精度可能不如在机器人零点位置执行力传感器标定。此外，开启动态标定功能后，在开启拖动、RL 程序中开启力控时，系统也会自动进行标零，确保力控相关功能正常使用。

警告

动态标定有如下两个使用风险：

1.如果开启拖动的时候，机器人和环境有接触，即处于一个非自由状态下，标定出来的零点误差较大，可能导致计算力矩异常，造成开启力控失败；

2. 开启拖动或者 RL 开启力控时，可能会遇到动态标定报错，提示当前传感器力矩与理论模型偏差较大，可以检查以下几点：1）负载是否正确设置，2）基坐标系是否正确设置；3）机械零点是否偏移较大；4）开启功能时是否和外界环境有直接力接触。

由于以上风险点，如果不是力矩传感器零点漂移现象比较严重，不推荐打开动态标定功能，此功能开关默认为关闭状态。

9.6 基坐标系标定

基坐标系位于机器人底座的中心，相对世界坐标系描述，定义了机器人相对世界坐标系位姿。在机器人倒装、墙装、多机器人协同等场景下，通常需先进行基坐标系标定，否则容易出现机器人过载、抖动、力控失效等异常现象。

基坐标系标定方法有两种：手动输入和六点法标定。

9.6.1 手动输入

手动输入基坐标相对于世界坐标系的位置和姿态，姿态可以选择欧拉角或四元数。

如果选择预设的两种安装方式(正装、倒装)，可以自动设置姿态数值；如果安装方式选择"自定义"，可以手动输入。

9.6.2 六点法标定

步骤	图示	说明
1.标定一个工具；		
2.如果不是正装，先关闭动力学前馈(设置->动力学设置-动力学前馈)；		
3.基坐标系标定的模式，选择"标定"		
4.定义辅助点位置，该步骤可选；		当机器人离世界坐标系距离较远，工具末端不可达时，可通过辅助点来标定基坐标系。 辅助点位置相对于世界坐标系定义。
5.按 HMI 示意步骤，完成六个点的设置，并"确定"。		
6. 根据标定结果，选择是否保存基坐标系数据。		

9.7 坐标系标定

9.7.1 全局工具列表

9.7.1.1 概述

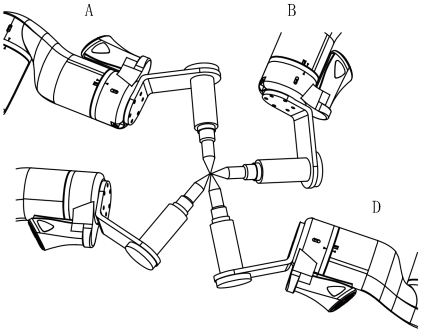
全局工具和工具定义一致，在 xCore 控制系统中，全局工具对应的数据类型为 g_tool_num，num 范围为 0-15，总共 16 个全局工具。全局工具也属于工具类型，关于工具详细描述，请参考"RL 指令-变量"章节。注意，全局工具不同于工程中的工具，全局工具可用于所有的工程中。

9.7.1.2 基本概念

与工具列表基本概念，详细见章节 8.12.2。

9.7.1.3 操作实例-创建全局手持工具

标定工具坐标系之前，需要准备一个固定的外部尖端点，要求该点位于机器人工作范围之内，并且使用待标定的工具能以比较灵活的姿态接触到该点。



操作	图示	说明
1.在坐标系标定-全局工具列表界面，选择需要编辑的全局工具。		在"编程员"及以上权限时，可进行编辑操作，且只有编辑功能，不能新增与删除。
2.设置工具属性：手持。		

3.负载辨识（以手动输入为例）：选择“手动输入”，点击“执行手动输入”按钮，进入“工具负载辨识”页面。		输入工具物理信息，点击“下一步”按钮，回到“新建工具”页面，此时“负载辨识”状态为“已辨识”。
4.位姿标定，以三点法为例：选择“立即标定”、标定方法为“三点法”，点击“执行标定”按钮，进入“新建用户坐标系”页面执行标定。		Jog 机器人，使标定好的工具 TCP 按顺序指到期望的坐标系原点/x 轴上一点/xy 平面上一点或 y 轴上一点，点击“确认第一点/第二点、第三点”按钮。 三个点都完成确认后，点击“确认标定”按钮，回到“新建工具”页面，此时“标定状态”为“已标定”。 若使用四点法标定工具原点，应使 4 个点之间的姿态差距应尽可能的大，即机器人应尽量以不同的姿态接触外部尖端点。
5.包络，选择“手动输入”，点击“执行包络编辑”，进入“包络设置”界面选择包络。		
6.点击“完成”按钮，完成全局工具的编辑。		在全局工具列表中，显示新建的工具。

9.7.1.4 操作实例-创建全局外部工具

全局外部工具的标定方法与全局手持工具一致，支持 4 点法、6 点法和手动输入三种方式。

注意：标定外部工具坐标系，需要使用已标定好的手持工具来进行。

操作	图示	说明
1.列表中已存在标定好的手持工具，使用它辅助标定外部工具，示例里采用 g_tool_0。		该工具应带尖端。
2.在全局工具列表界面点击右下角的“编辑”进入全局工具向导界面。		在“编程员”及以上权限时，可进行编辑操作。
3.设置工具属性：外部		
4.在右上角切换工具，选择已经标定好的手持工具 g_tool_0。		
5.位姿标定，以四点法为例：选择“立即标定”、标定方法为“四点法”，点击“执行标定”按钮，进入“工具标定”页面执行标定。		Jog 机器人，使标定好的工具 TCP 以不同的姿态指到期望的工件坐标系原点上，然后分别确认第一、二、三、四个点。应使 4 个点之间的姿态差距应尽可能的大，即机器人应尽量以不同的姿态接触外部尖端点。 标定完毕后，系统会弹出标定误差，根据误差大小来选择是否需要重新标定（参考确认标定精度）。

注意：

外部工具必须与对应的工件一起使用，即同时选中的工具和工件中的 robhold 参数必须有一个为 False 而另一个为 True，否则系统会给出错误提示，并禁止 Jog 机器人。

使用外部工具时，定义工具坐标系和工件坐标系的参考系与普通工具是不一样的，见下表：

坐标系名称	普通工具时，相对于...定义	外部工具时，相对于...定义
工件坐标系	用户坐标系	用户坐标系
用户坐标系	世界坐标系	法兰坐标系
工具坐标系	法兰坐标系	世界坐标系

9.7.2 全局工件列表

9.7.2.1 概述

全局工具和工具定义一致，详见 8.13.1 章节。在 xCore 控制系统中，全局工件对应的数据类型为 g_wobj_num，num 范围为 0-15，总共 16 个全局工件。全局工件也属于工件类型，关于工件的描述，请参考"RL 指令-变量"章节。注意，全局工件不同于工程种的工件，全局工件可用于所有的工程中。

9.7.2.2 操作实例-创建全局外部工件

标定全局外部工件坐标系，需要使用已标定好的全局手持工具来辅助标定。

操作	图示	说明
1.在工件列表界面点击右下角的"编辑"进入编辑全局工件向导界面。		在"编程员"及以上权限时，可进行编辑操作
2.设置全局工件属性：外部。		
3.在右上角切换工具为已经标定好的全局手持工具 g_tool_0		

4.点击"执行标定"按钮，进入全局工件标定界面。		Jog 机器人，使标定好的工具 TCP 指到期望的工件坐标系原点，点击"确认第一点"按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 X 轴上一点，点击"确认第二点"按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 Y 轴上一点，点击"确认第三点"按钮，点击"确认标定"，弹出标定结果确认对话框，点击"确认。"点击"完成"按钮，返回"全局工件列表编辑"页面
5.点击"完成"按钮，完成全局工具编辑。		在全局工具列表中，显示编辑完成的全局工具。

9.7.2.3 操作实例-创建全局手持工件

标定全局手持工件坐标系，需要使用已标定好的全局外部工具来辅助标定。

操作	图示	说明
1.在全局工件列表界面点击右下角的"编辑"进入新建工件向导界面。		在"编程员"及以上权限时，可进行编辑操作。
2.设置工件属性：手持		"外部"为工件与机器人末端不相对固定；"手持"为相对固定。

3.在右上角切换工具为已经标定好的外部工具 g_tool_1		
4.点击“执行负载辨识”，进入负载辨识界面。		
5.点击“执行标定”按钮，进入工件标定界面。		Jog 机器人，使标定好的工具 TCP 指到期望的工件坐标系原点，点击“确认第一点”按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 X 轴上一点，点击“确认第二点”按钮；Jog 机器人，使 TCP 指到期望的工件坐标系 Y 轴上一点或 XOY 平面上一点，点击“确认第三点”按钮，点击“确认标定”，弹出标定结果确认对话框，点击“确认。”点击“下一步”按钮，返回“新建工件”页面
5.，点击“完成”按钮，完成全局工件编辑		

9.7.3 全局用户坐标系列表

9.7.3.1 概述

全局用户坐标系，在定义工件坐标系或者全局工件坐标系时充当参考系，不单独使用。

选择全局用户坐标系可以选择"立即标定"、"手动输入"或"不标定"。

选择"立即标定"时，采用 3 点法标定。在标定全局用户坐标系前，需先标定一个工具，然后使用该工具的 TCP 来进行用户坐标系标定，推荐使用带有尖端的工具。

如果提前知道全局用户坐标系，可以手动输入。也可选择不标定，此时默认全局用户坐标系是世界坐标系。

9.7.3.2 标定全局用户坐标系

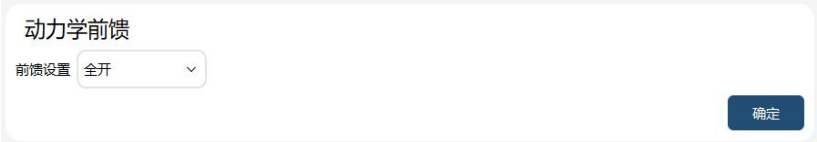
操作	图示	说明
1.点击右下角"编辑"按钮，编辑一个全局用户坐标系。		在"编程员"及以上权限时，可进行编辑操作。
2.按图示步骤，执行标定。		Jog 机器人，使标定好的工具 TCP 按顺序指到期望的坐标系原点、x 轴上一点、xy 平面上一点或 y 轴上一点，点击"确认第一点、第二点、第三点"按钮。 三个点都完成确认后，点击"确认标定"按钮。

9.8 动力学设置

动力学设置页面用来设置机器人的动力学相关参数。动力学设置跟机器人力控、拖动示教、虚拟墙、碰撞检测等功能有关联。请确保机器人动力学设置合理，否则可能出现上述功能无法正常使用，或者机器人异常抖动的现象。

9.8.1 动力学前馈

动力学前馈开关决定着控制器是否开启动力学前馈功能，默认为开启状态。不建议用户自行关闭动力学前馈功能，可能会造成上电点头、轨迹精度下降等负面影响。某些特殊情况下需要关闭动力学前馈，如非正装下进行基坐标系标定、执行摩擦力辨识等。



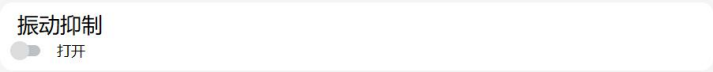
9.8.2 动力学约束

动力学约束开关决定着控制器是否开启动力学约束功能，默认为开启状态。不建议用户自行关闭动力学约束功能，可能会造成电机过载或者异常抖动等负面影响。

动力学约束开关开启后，有"标称动力学参数"和"出厂辨识动力学参数"两个选项。其中"标称动力学参数"，代表动力学控制使用该机型的标称参数，好处是使用"标称动力学参数"的同机型机器，在执行相同的运动程序时，会保证完全相同的运动速度和节拍时间，代价是可能会牺牲一定的运动性能，或者存在电机超载的风险； 使用"出厂辨识动力学参数"会让机器人保持最佳的动力学控制状态，能够达到本台机器允许的最短节拍时间，并极大地保护电机，降低超载风险，但代价是不同机器人在跑同样的运动程序时，运动速度和节拍时间会稍有差异。

9.8.3 振动抑制

振动抑制开关决定着控制器是否开启振动抑制功能，默认为关闭状态。



振动抑制开关开启后，控制器将补偿机器人关节柔性，抑制关节柔性引起的振动以提高位置和路径精度。机器人负载参数的变化会影响关节柔性补偿量，因此在使用振动抑制功能时需要设置准确的负载参数。

> 提示
>
> 1. 更改振动抑制开关，控制器无需重启即可生效。
> 2. 可以在 RL 工程中使用 VibSuppression 指令更改振动抑制功能的状态（见章节 15.4.16.39）。.
> 3. 目前仅部分机型支持振动抑制功能，支持机型参考下表。

振动抑制适配机型表：

产品线	主机型	子型号
协作机器人（6 轴）	SR3	所有型号
	SR4	所有型号
	SR5	所有型号
	CR7	所有型号
	CR12	所有型号
	CR18	所有型号
	CR20	所有型号
	CR35	所有型号

协作机器人（5 轴）	CR17	所有型号
	CR25	所有型号
工业机器人（6 轴）	NB4	NB4-R475-3B
		NB4h-R580-3B
	NB12s	NB12s-1214-5A
		NB12s-1016-5A
		NB12s-1611-5A
	NB12h	NB12h-1214-6A
	NB25s	NB25s-1221-67
		NB25s-2020-67
		NB25s-2518-67
		NB25s-3016-67
		NB25s-3518-67
	NB25h	NB25h-2518-77
		NB25h-1222-77
	NB80s	NB80s-8022-21
		NB80s-5026-21
	XB4h	XB4h-R596-3B
	XB7s	XB7s-R707-3A
		XB7s-R906-0A
		XB7s-R906-3A
	XB10s	XB10s-R1206-3B

9.9 本体参数

本体参数包含 RD 参数、DH 参数、减速比、过载系数、耦合系数等机器人本体相关数据。该页面的参数直接影响运动的准确性，如果没有机器人厂家的许可和协助，请勿修改。

注意：从 3.0 版本开始、除了 PCB 三、四轴机型使用 RD 参数，其他机型都使用 DH 参数。

9.9.1 RD 参数

RD 参数描述了机器人连杆坐标系之间相对位姿关系，是机器人运动学的基础。

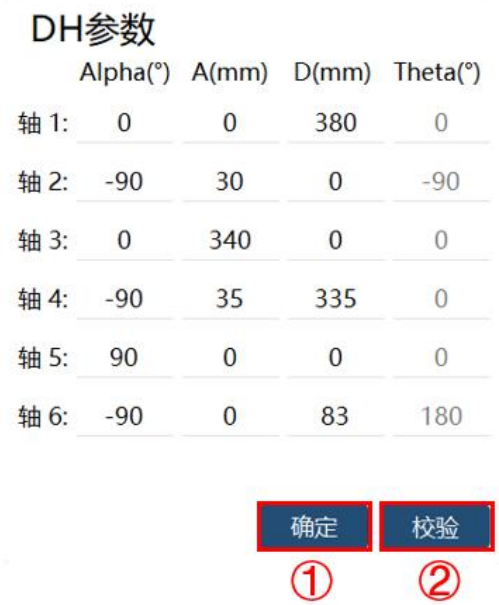
①	导入：点击选择 RD 参数文件，一般无须进行此操作；
②	确定：手动修改 RD 参数后，点击“确定”生效；
③	校验：点击后，会显示当前 RD 参数与系统默认 RD 参数；

注意：

机器人出厂前会存在一组默认参数；用户在修改或导入参数前，需确认 RD 参数的合理性；RD 参数修改后，需要重启控制器生效。

9.9.2 DH 参数

DH 参数描述了机器人连杆坐标系之间相对位姿关系，是机器人运动学的基础。



①	确定：手动修改 DH 参数后，点击“确定”生效；
②	校验：点击后，会显示当前 DH 参数与系统默认 DH 参数；

注意：

机器人出厂前会存在一组默认参数；用户在修改或导入参数前，需确认 DH 参数的合理性；DH 参数修改后，需要重启控制器生效。

DH 参数为改进型 DH 参数，用户只能修改参数 Alpha ，A，D。

9.9.3 减速比

减速比是机器人各轴减速器的参数。如果没有机器人厂家的许可和协助，请勿修改。

①	确定：手动修改减速比参数后，点击"确定"生效；
②	校验：点击后，会显示当前减速比参数与系统默认减速比参数；

9.9.4 过载系数

电机过载系数设置。

①	确定：手动修改过载系数参数后，点击"确定"生效。请在厂家支持下谨慎修改！
②	校验：点击后，会显示当前过载参数与系统默认过载参数；

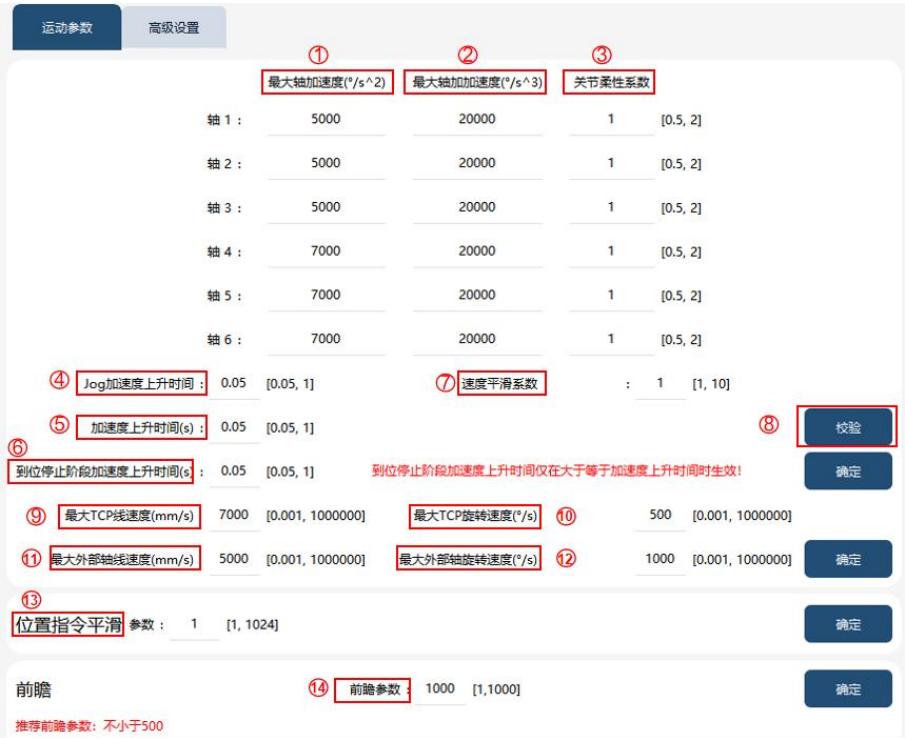
9.9.5 耦合系数

部分机器人，各个轴存在耦合关系，耦合系数描述了这些轴之间的耦合关系。如果没有机器人厂家的许可和协助，请勿修改。

9.10 运动参数

9.10.1 基础运动参数

运动参数包括：机器人各轴最大速度、最大加速度、最大加加速度，以上参数影响机器人运动节拍和平顺性。机器人在出厂前存在一组默认参数，修改以上参数，可能造成机器人异常抖动、报错、影响机器人使用寿命，请谨慎修改。



1	最大轴加速度：机器人各轴加速度上限，受电机力矩能力限制。关闭动力学约束时，该参数生效；开启动力学约束时，该参数失效。该数值一般不小于该轴的最大轴速度的 3-5 倍。
2	最大轴加加速度：机器人各轴加加速度上限。通常情况下，增大加加速度，机器人节拍提升，运动时容易抖动。如果机器人运动中，小转弯区较多，可以通过适当增大加加速度，提升节拍，但是需要注意机器人的抖动情况。该数值不应小于该轴的最大加速度的 3-5 倍。
3	关节柔性系数：用于调整机器人关节刚度模型的系数，设置范围[0.5~1.2]。设置关节柔性系数会影响重力补偿以及振动抑制功能的效果。 注意： 该值一般无需调整，默认值即可；特殊情况，请在厂家指导下修改。
4	Jog 加速度上升时间：Jog 时，机器人加速度从最小值增加到最大值所消耗的时间，仅在 10mm 步长 Jog 和连续模式 Jog 生效。该值越小，机器人 Jog 的启动和停止

	越迅速，但更容易产生抖动。如果感觉 Jog 时机器人反应偏慢，可适当调小该值。
5	加速度上升时间：运动程序时，机器人加速度从最小值增加到最大值所消耗的时间。数值越小，机器人加速越快；数值越大，机器人加速越平缓。
6	到位停止阶段加速度上升时间：针对无转弯区目标点，机器人停止阶段的加速度从最小值增加到 0 的时间。数值越大，机器人停止越平缓。若加速度上升时间大于该到位停止阶段加速度上升时间，按照较大的加速度上升时间停止。
7	速度平滑系数：用于调整机器人通过转弯区时的速度平滑处理。数值越大，机器人经过转弯区时降速越小，越容易产生抖动。 该参数数值为 1.0 时，机器人经过转弯区不额外进行速度平滑处理。 该参数主要用于提升机器人的极限性能。调试时，首先确认数值为 1.0 时，机器人的抖动情况：若机器人抖动较为剧烈，说明机器人已达极限性能，此参数无需增大；若机器人运行平稳、但经过转弯区时降速严重，可逐渐增大该参数提升运动平滑性。该参数调试时，建议以 0.1~0.5 的步长逐步增大。
8	校验：点击后，会显示当前运动参数与系统默认运动参数；
9	最大 tcp 线速度：设置 vmax 参数 tcp 线速度
10	最大旋转线速度：设置 vmax 参数 tcp 旋转速度
11	最大外部轴线速度：设置 vmax 参数外部轴线速度
12	最大外部轴旋转速度：设置 vmax 参数外部轴旋转速度
13	位置指令平滑参数：设置控制器对于机器人关节位置指令的平滑处理程度，参数设置范围[1~1024]。数值为 1 时，控制器不对关节位置指令做平滑处理。数值越大，平滑程度越大，机器人运动越缓。修改位置平滑参数会影响机器人的节拍以及位置精度。 注意： 该值一般无需调整，默认值即可；特殊情况，请在厂家指导下修改。 使用 ARM 平台的机器人，不支持位置指令平滑参数设置。
14	前瞻：数值越大，机器人运动越平滑高速，但是会影响对逻辑、信号处理的及时性。注意，该值一般无需调整，默认值即可；特殊情况，请在厂家指导下修改。

9.10.2 高级设置

9.10.2.1 安全控制

支持一种停止方式，每种的停止参数可设置；

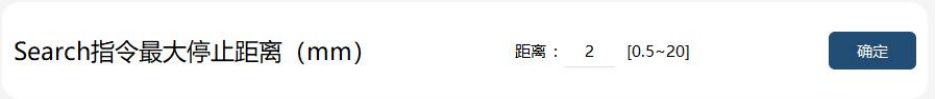
目前采用最大能力减速停止。收到停止信号后，确保机器人路径不偏移前提下，至少一个电机按照最大减速能力进行规划停止。

考虑到不同机器人在不同工况下，实际性能容易受到多种因素影响。该停止方式允许设定缩放系数，设定范围为[0.1,1]。当设定为最大值 1 时，机器人出力迅速到达电机最大能力，进行减速停止；设定为最小值 0.1 时，机器人出力以较平缓的方式到达电机最大能力，进行减速停止。即该系数数值越大，机器人出力值到达电机最大力矩值越迅速，机器人停止越迅猛，该系数数值越小，机器人出力值到达电机力矩值越平缓，机器人停止越平缓。

另外，手动模式停止系数（Stop0）和自动模式停止系数（Stop1）允许分开设定。

9.10.2.2 Search 指令最大停止距离

使用 Search 类指令，且停止方式选择快速停止时，从机器人收到停止运动信号，到机器人完全停止运动，该过程中机器人 tcp 点走过的距离不超过该数值。



9.10.2.3 最小转弯区半径

通过设置最小转弯区半径指定允许生成的最短转弯区大小。可利用该参数避免生成长度过短的转弯区，提升运动的平滑性。当控制系统检测到某条轨迹的长度小于该参数设定值，且该轨迹需要生成转弯区时，控制系统会自动将该轨迹与其相邻的的轨迹合并成一条轨迹，从而生成长度符合要求的转弯区。该参数数值越大，最小转弯区长度越长，机器人经过转弯区时运动越平滑。当该参数数值设为 0 时，控制系统不再对转弯区进而额外处理，严格按照转弯区参数生成转弯区。

9.10.2.4 码垛调试模式

码垛调试模式选项仅在机型类型为 CR、SR、工业六轴机器人（NB、XB）系列机型时，才会显示并可用。开启该模式，会增加两个四轴锁定对应的 Jog 坐标系：奇异规避和平行基座。

注意：CR 系列 5 轴机型默认打开该模式，此时在基坐标系下的 Jog 对应 6 轴机型的奇异规避坐标系。

9.10.2.5 默认 Conf

默认 Conf 的“RL”选项用于设置每次重新加载 RL 工程后，运动指令是否严格按照点位的 Conf 信息进行运动约束。注意，修改该选项后，需要 pptomain 让该选项生效。

默认 Conf 的“点位运动至”选项用于设置“点位运动至”情况下，运动是否按照点位的 Conf 信息进行运动约束。

注意，在该界面没有修改前，默认 Conf 里的 4 个按钮都是默认打开的。

9.11 力控参数

9.11.1 力控参数

力控增益：调节机器人力控响应速度，数值越大，机器人力控制响应越快，但是抗干扰能力越弱。默认值为 1.0。如果机器人底座刚度较低，例如将机器人置于移动底盘上，应适当降低力控增益值。当末端负载接近负载曲线图中的负载上限时，使用拖动以及阻抗功能，可能会出现抖动现象，此时应适当调低力控增益数值；若控制系统为 V2.0 以下版本，应该调整底座刚度选项为低刚度模式。

传感器补偿系数：用来补偿传感器误差，通常不建议修改此参数。若机器人拖动过程中，有朝关节正方向运动的力，可适当调小此参数数值；反之，若有朝关节负方向运动的力，可适当调大此参数数值。默认值为 0.5。

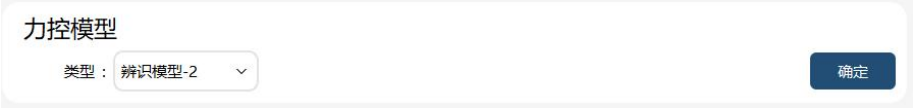
摩擦力补偿系数：补偿拖动和阻抗运动时的摩擦力，补偿数值太大可能出现不稳定现象，请谨慎修改。默认值为 0.5。

注意：

力控参数面向高级开发者，请谨慎修改。

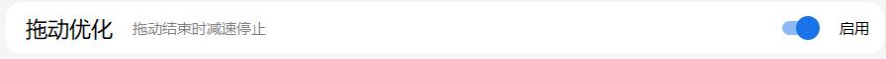
9.11.2 力控模型

力控模型有三种可选，此功能为开发者选项，请谨慎修改。初始化及未修改状态下，使用默认的标称模型。



9.11.3 拖动优化

拖动优化，通过拖动结束时的停止处理，提升拖动体验。初始化及未修改状态下，默认启用该功能。



9.11.4 无末端按键拖动

协作机器人支持无末端按键情况下的拖动；打开自动开启拖动的开关，机器人可打开拖动开关后直接拖动机器人（无需按末端按键）。

注意：

1、自动开启拖动开关开启后，打开拖动时需注意安全，如机器人出现异常，请按急停；

2、自动开启拖动开关开启后，末端拖动按键失效。

9.11.5 力控模型偏差阈值设置

协作机器人在开启拖动时会检查传感器力矩与理论动力学模型力矩的偏差，若偏差较大，会出现开启拖动失败的情况。若在检查负载设置正确后仍然出现开启失败的情况，可以适当调大力控模型偏差阈值。

注意：

1、不启用状态下控制器会采用默认值

9.11.6 传感器双通道偏差阈值设置

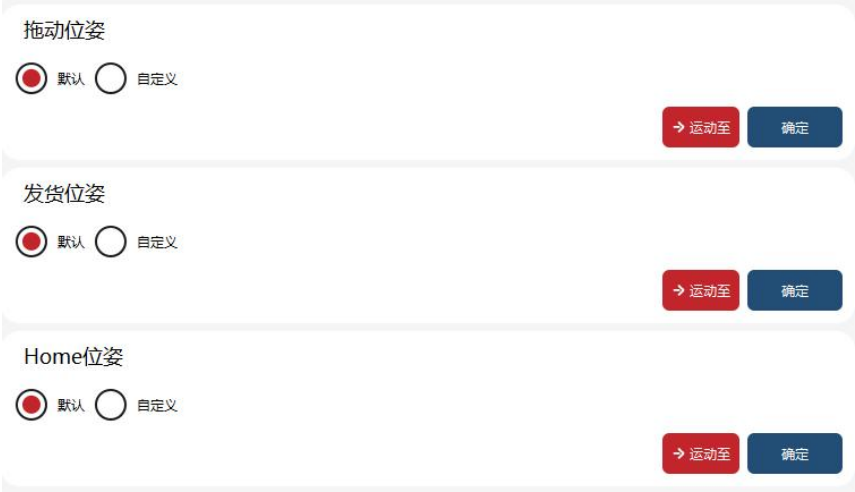
协作机器人在开启拖动时会检查传感器两通道电压值的偏差，若偏差较大，会出现开启拖动失败并提示电压偏差较大，此时可以适当调大传感器双通道偏差阈值。

注意：

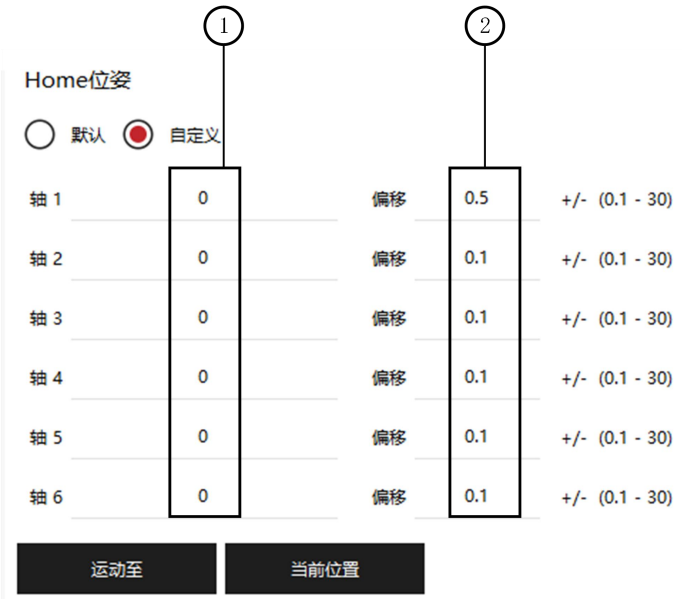
1、不启用状态下控制器会采用默认值

9.12 快速调整

快速调整功能，用户可以自定义若干常用位置，通过按钮或指令，实现机器人位姿快速调整。支持自定义的位姿包括：拖动位姿、发货位姿、Home 位姿。以上位姿，有针对具体机型的默认值，也支持用户自定义。



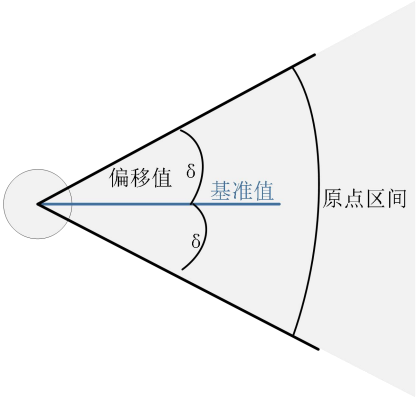
Home 位姿比较特殊，除了定义基准位置，还需设置一组偏移量。当机器人当前位置与基准位置之差小于偏移量，就认为机器人处于 Home 位姿，系统 IO 的"Home 状态"和绑定功能码寄存器"sta_home"的寄存器会有对应输出。

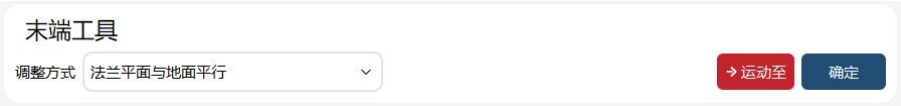


序号	名称	含义
①	基准值	各个关节的原点基准值。
②	偏移值	以基准值为中心，原点区间的上下对称浮动值，支持设定的范围为

		[0.1,30]。如基准值为 1°，偏移值为 3°时，原点的区间范围为[-2°,4°]。

原点区间、基准值和偏移值之间的关系如下图：

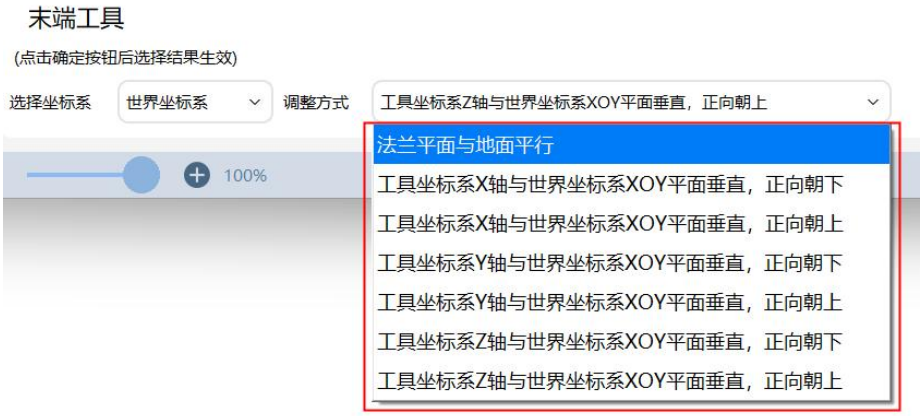




快速调整功能，还可以在保持机器人 TCP 位置和臂角（仅 7 轴机器人有臂角的概念）不变的情况下，快速调整机器人末端工具到一些指定的姿态；末端工具快速调整有两个可选项，分别为"选择坐标系"和"调整方式"。

选择坐标系下拉框包括：

调整方式下拉框包括：（当坐标系选择某坐标系时，调整方式下拉框内容会根据选择的坐标系变化，以"选择坐标系" 选定世界坐标系为例如下图所示）

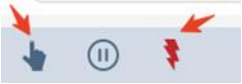


使用流程：

快速调整功能的使用方式与 JOG 操作类似，具体步骤如下：

1. 确定是否为正确的工具、工件；
2. 确定机器人工具末端的目标姿态；
   a) 选择坐标系；（以世界坐标系为例如下图）
   b) 选择调整方式；（以法兰平面与地面平行为例如下图）
   c) 点击"确定"按钮生效；

2. 切换手动模式上电；



3. 长按"运动至"按钮，机器人通过轴空间路径运动至目标位姿；

注意：

1. 使用快速调整功能进行位姿调整时，请开启软限位，以免造成预期之外的碰撞。
2. 运动过程的速度可通过 JOG 速度调整。运动中如果下电或者松开"运动至"按钮，机器人停止运动。
3. 使用匹配的工具、工件类型，即手持工具/外部工件、手持工件/外部工具，否则不允许运动，错误的选择会有对应报错提示；
4. 外部工具情况下，不可以进行基于世界坐标系、基坐标系的 XOY 平面垂直的调整，错误的选择会有对应报错提示；。

9.13 电子铭牌

部分工业机器人本体内置电子铭牌，主要保存和本体相关的数据，避免更换工控机或者更换控制柜导致的基本数据丢失。

电子铭牌的软件功能主要分为控制器和 RobotAssist 软件两部分。控制器负责电子铭牌的数据读取、校验、覆盖等功能，RobotAssist 软件负责电子铭牌相关操作命令的下发及数据展示等功能。

控制器开机后，会先检测电子铭牌是否存在：若存在，则正常读取数据，进行数据校验，并将校验结果进行存储；若不存在，且用户未选择使用电子铭牌，则直接使用控制器数据正常运行；若不存在，且用户选择了使用电子铭牌，则会给出电子铭牌不存在的提示。RobotAssist 软件在连接上控制器后，会首先检测控制器内电子铭牌数据校验的结果，并根据不同的校验结果进行弹窗提示，用户只需根据弹窗提示进行操作即可。

若成功使用电子铭牌中的数据，则会默认使用电子铭牌的数据覆盖控制器中的数据。

存在以下三种出现弹框提示的情况：

（1）检测到存在电子铭牌，且数据与控制器不同，则会弹框提示"是否使用电子铭牌数据"；选择"是"，则使用电子铭牌中的信息；选择"否"，则使用控制器数据；

（2）选择过一次使用电子铭牌数据后，再次开机，会默认使用电子铭牌信息；当控制器信息再次与电子铭牌信息不同时，会重新弹框提示"是否使用电子铭牌数据"；

（3）开机初始就不存在电子铭牌，则会默认使用控制器数据；但使用过电子铭牌信息后，若

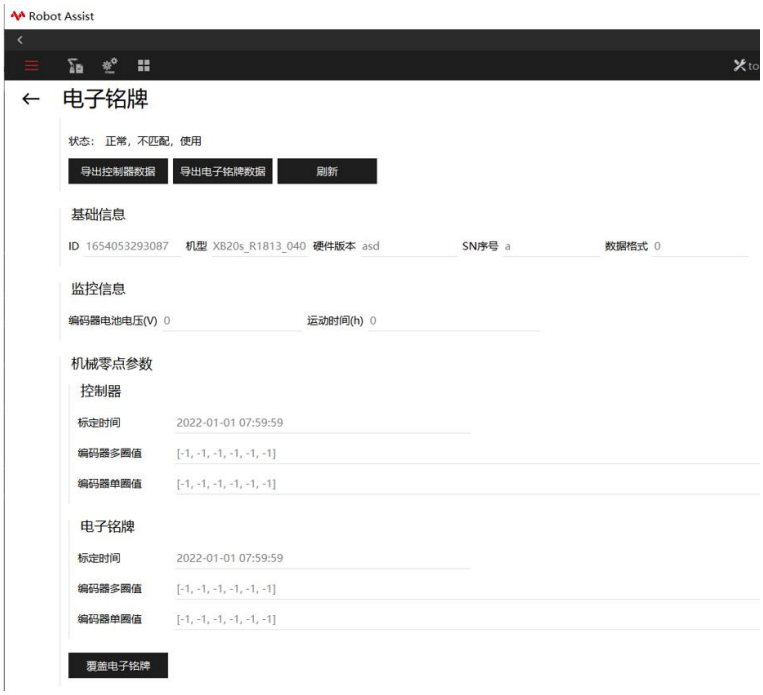
下次开机检测不到电子铭牌，则会提示"是否使用控制器数据"；选择"是"，正常使用控制器数据；选择"否"，控制器整体处于故障状态，无法进行操作，需要重启解决；

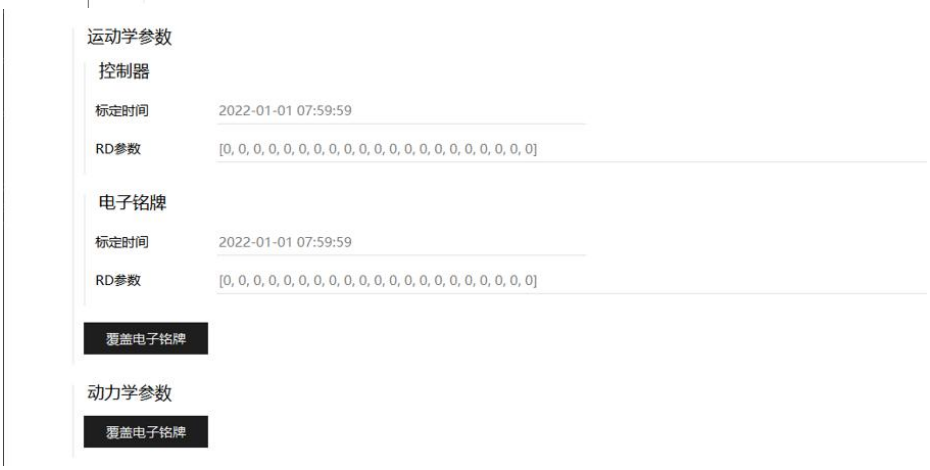
> 提示
>
> 当电子铭牌中的机型与控制器中的机型不一致时，则使用电子铭牌数据失败；若想成功使用电子铭牌中的数据，需注意保持控制器中的机型与电子铭牌中的机型一致。

HMI 软件界面依次点击"设置"->"电子铭牌"，出现电子铭牌中的信息。若控制器检测到电子铭牌，无论是否使用电子铭牌，此界面都会显示电子铭牌中相关参数段的信息。电子铭牌的状态可以通过此界面的状态栏位进行判断，三个参数分别表示：电子铭牌状态、电子铭牌数据与控制器数据的匹配情况、开机是否"点击使用电子铭牌"按钮（只要点击"使用电子铭牌数据"，无论数据使用成功与否，此位置都会显示为使用），如下图：

开机若未检测到电子铭牌，界面显示则如下图：

功能	说明
导出控制器数据	将控制器中相关参数段的数据导出至文件中
导出电子铭牌数据	将电子铭牌中的数据导出至文件中
刷新	同步电子铭牌信息状态
基础信息	显示电子铭牌中基础信息的参数段；不可手动修改
编码器电池电压	显示编码器电池电压；每次开机或开机后的每 24h 会检测一次编码器电池电压，并显示实际的电压值；不可手动修改
运动时间	当电机有动作时，会累加运动时间；界面显示值会每 1h 刷新一次；不可手动修改
机械零点参数、运动学参数	界面会分别显示控制器的目前值及电子铭牌的目前值；不可手动修改
动力学参数	此参数段在界面隐藏显示
覆盖电子铭牌数据	使用控制器中的数据覆盖电子铭牌中的数据





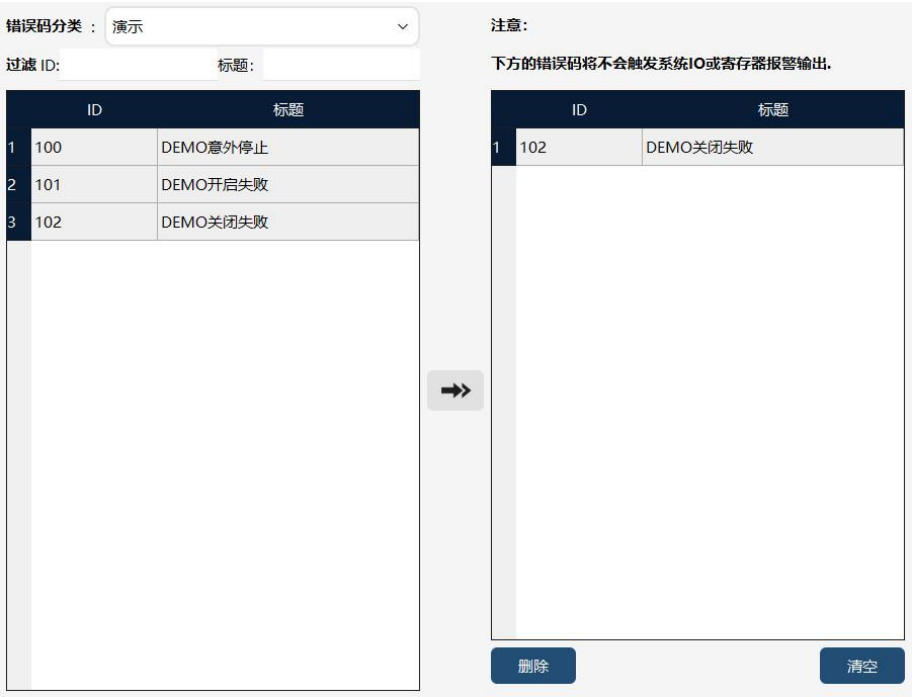
提示

1、导出的所有数据均为加密显示。

2、若选择使用电子铭牌，当机器人在进行零点标定、修改本体参数、动力学辨识操作后，控制器会自动向电子铭牌中同步修改后的数据。

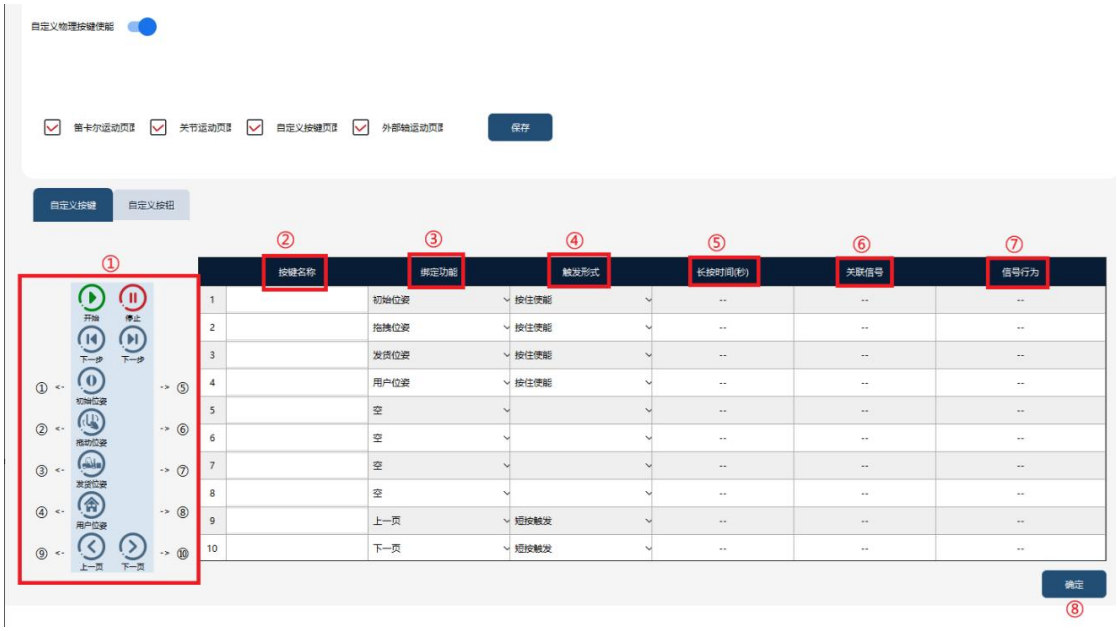
9.14 错误码报警过滤

当 xCore 出现错误级别的报警时，会触发绑定报警状态的系统 IO 或寄存器输出高电平。如果客户不希望某些报警触发报警状态输出(系统 IO 或寄存器)，可以通过该功能进行设置。



9.15 自定义按键

通过自定义按键功能，可以将部分便捷功能与 xPad2 上的若干实物按钮绑定。

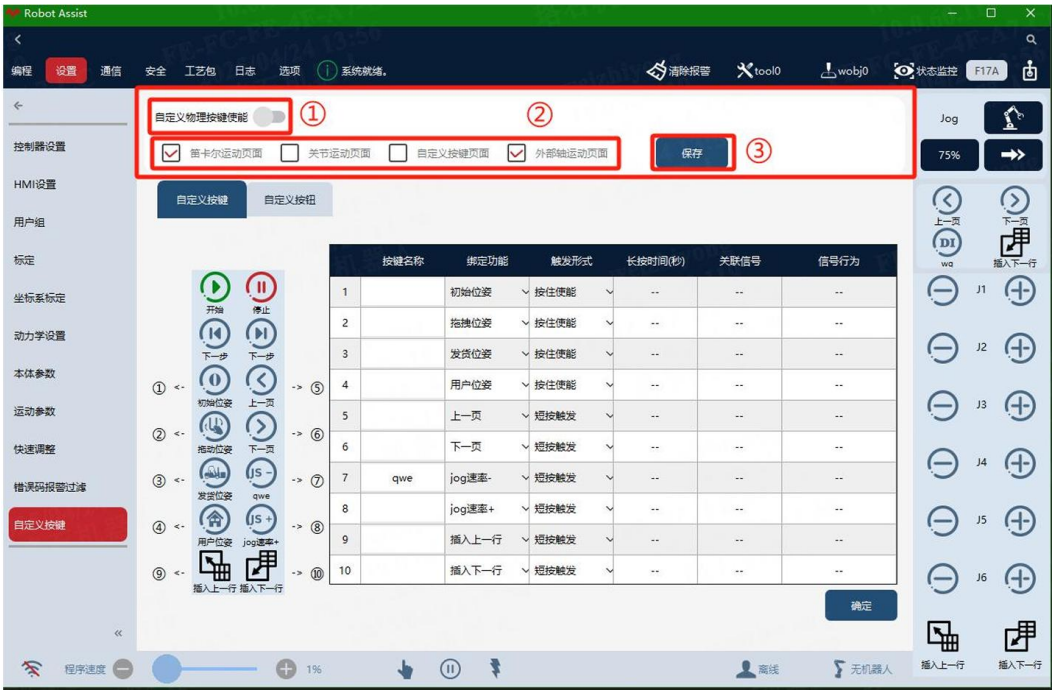


①	这 8 个按键的位置可以自定义；
②	按键名称：用户可以设定按键名称，该名称会显示在对应按键图标下方；建议名称不要过长，以免显示不完整。
③	绑定功能：有"空、强制 DO、强制 DI、截屏、软键盘、初始位姿、拖拽位姿、发货位姿、用户位姿、锁屏(仅支持 xPad2 示教器)"这几项功能可选。
④	触发形式：分为按住使能、短按触发、长按触发三种。 按住使能，就是按住按键，才能使用对应的功能；短按触发，就是按下就立即触发绑定的功能；长按触发，需要长按一段时间才能触发绑定的功能，长按时间可设置。

	每种绑定功能支持的触发形式不尽相同。
⑤	长按时间：在触发形式选择"长按触发"后，可以设置长按时间，单位秒，范围[0.5,5]。
⑥	关联信号：当绑定功能为"强制 DO"、"强制 DI"时，可以在此选择信号。
⑦	信号行为：当绑定功能为"强制 DO"、"强制 DI"时，通过信号行为定位信号如何变化，支持"0"、"1"、"交替"；"0"为触发后对应信号置 0，"1"为触发后对应信号置 1，"交替"代表触发后，如果当前为 0，则置为 1，反之类似。
⑧	确定：自定义按键的修改在点击"确定"后生效。

9.15.1 自定义按键屏蔽功能

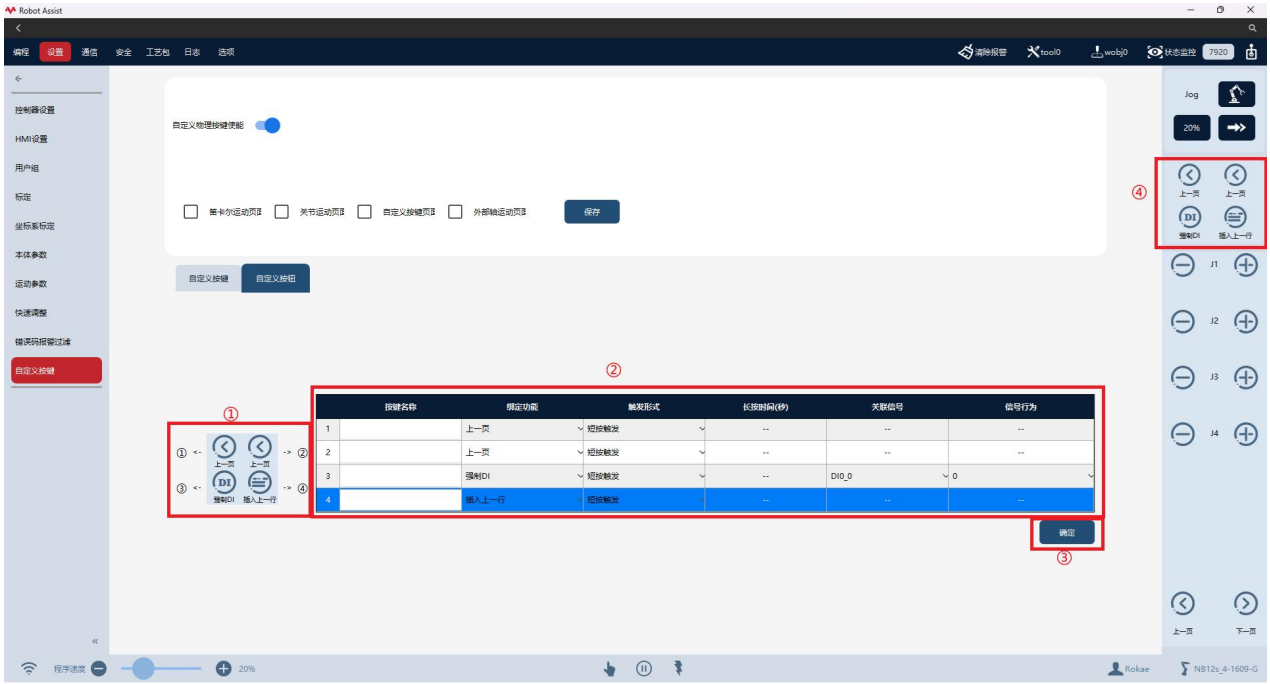
本功能主要用于防止自定义页面按钮的物理按键误触。用户可以自定义页面按钮的物理按键使能作用。



该功能包含以下几个部分：
①自定义物理按键使能按钮：用于控制各个页面的物理按键开关的使能关系。
使能按钮关闭，下方页面的物理 按键开关失效，所有页面的物理按键有效。
使能按钮打开，物理按键开关勾选的页面物理按键功能有效。
②物理按键开关：当使能按钮开启时有效，打开表示页面物理按键功能有效。
③保存按钮：只有点击保存按钮后，修改的状态才会应用。

9.15.2 自定义按键插入下一行

本功能用于工业机器人，提供四个可选的自定义按钮功能。在协作机器人的界面上不显示自定义按钮配置功能。



该功能包含以下几个部分：

①自定义按钮样式展示模块：用于展示配置好的功能按钮名称和图标，并且有数字和右侧的表格函数对应。

②自定义功能设计表：可以通过修改“绑定功能”列切换自定义按钮显示的样式和功能。

③保存按钮：只有点击保存按钮后，操作的表设计可以保存并应用。

④自定义按钮功能模块：软件中可以发挥作用的部分。

10 通信

10.1 本章简介

本章主要介绍 xCore 控制系统的各种通信设置，包括：系统 IO、寄存器、总线设备、末端工具等。

10.2 系统 IO

系统 IO 分为两种：系统输入和系统输出。外部控制器可通过系统输入，给 xCore 控制系统发送各种指令，如电机上电，启动程序，急停复位等；xCore 系统也可使用系统输出，对外发送机器人状态，如上下电状态、运行状态等。

10.2.1 系统输入

在 HMI 主界面，通过"通信"->"系统 IO"，进入系统 IO 设置界面，点击"系统输入"选项卡可进入系统输入配置界面，如下图所示：

xCore 控制系统支持的系统输入如下：

系统输入名称	触发方式	备注
上电	上升沿/下降沿	
下电	上升沿/下降沿	
启动程序	上升沿/下降沿	触发功能码后，若示教器报警提示"程序未同步至控制器，启动失败"，请同步程序后，再次触发该功能码。
暂停程序	上升沿/下降沿	
暂停程序 1	上升沿/下降沿	功能和"暂停程序"相同
自动模式	上升沿/下降沿	可在手动模式生效
手动模式	上升沿/下降沿	
自动且上电	上升沿	可在手动模式生效，依次执行切换自动模式、上电；当机器人处于上电模式不响应该指令
指针到 main	上升沿	
上电且运行	上升沿	依次执行上电、pptomain、运行。 触发功能码后，若示教器报警提示"程序未同步

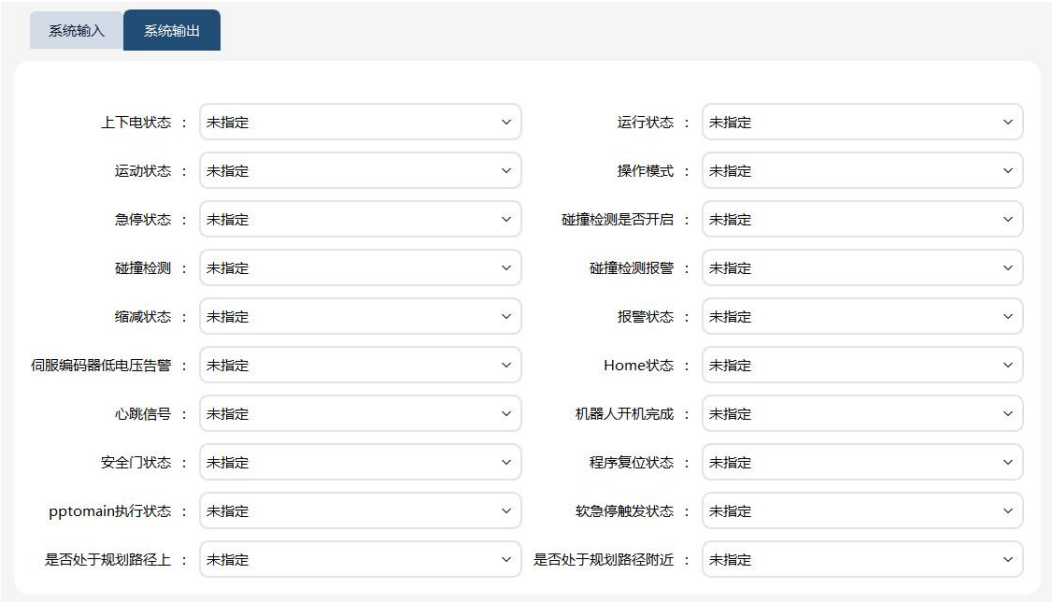
		至控制器，启动失败”，请同步程序后，再次触发该功能码。
上电并继续运行	上升沿	依次执行上电、运行。 触发功能码后，若示教器报警提示“程序未同步至控制器，启动失败”，请同步程序后，再次触发该功能码。
暂停并下电	上升沿	暂停，待机器人停止后下电
进入缩减模式	上升沿/下降沿	仅协作机器人
退出缩减模式	上升沿/下降沿	仅协作机器人
开始拖动	上升沿/下降沿	仅协作机器人，需在操控面板打开拖动
停止拖动	上升沿/下降沿	仅协作机器人，需在操控面板打开拖动
急停复位	上升沿	
清除报警	上升沿	
急停复位并清除报警	上升沿	

注意：

- 所有的系统输入均为脉冲触发，为保证 xCore 系统正确接收外部的指令，请确保外部输入的脉冲宽度不小于 300 毫秒。
- 为保证安全性，同时支持上升沿、下降沿触发的功能(上电、下电、手动模式、自动模式、启动程序、暂停程序、进入协作模式、退出协作模式、开始拖动、停止拖动)存在一一对应关系，例如“上电”和“下电”是对应的；如果“上电”选择了 DI1，那么 DI1 就不能再用于“下电”之外的其他功能。
- 系统输入大部分功能仅在自动模式下有效，手动模式下从系统输入传来的信号将被忽略。
- 当有绑定了暂停功能的寄存器或系统 IO 没复位时，不允许通过任何方式启动程序。
- 暂停程序、暂停程序 1 功能相同，任意一路信号触发都会暂停程序。

10.2.2 系统输出

在 HMI 主界面通过“通信”->“系统 IO”，进入系统 IO 设置界面，点击“系统输出”选项卡可进入系统输出配置界面，如下图所示：



xCore 系统支持的系统输出如下：

系统输出名称	输出有效	输出无效	备注
上下电状态	电机上电	电机下电	
运行状态	程序运行	程序未运行	
运动状态	机器人在运动	机器人静止	只检测 RL 程序中运动指令和 Jog 时，机器人运动状态。（注意：辨识、拖动、力控、拖动回放等情况下，机器人即使在运动，该输出也是机器人静止。）
操作模式	自动模式	手动模式/等待模式	
急停状态	急停状态	非急停状态	此状态的输出受"急停触发电平类型"设置的影响，当设置为高电平时，触发软急停时输出有效，未触发时输出无效；当设置为低电平时，触发软急停输出无效，未触发时输出有效。
碰撞检测是否开启	开启	关闭	PCB3、4 轴机器人不支持
碰撞触发	触发	未触发	PCB3、4 轴机器人不支持
碰撞检测报警	触发碰撞检测报警	报警被复位	PCB3、4 轴机器人不支持
缩减模式状态	缩减模式触发	缩减模式未触发	PCB3、4 轴机器人不支持
报警状态	报警状态	无报警	
伺服编码器低电压告警	编码器低电压报警	无编码器低电压报警	
Home 状态	机器人各关节到 home 点	机器人各关节未处于 home 点	
心跳信号	心跳信号		在"设置-控制器设置"可以设置心跳周期；
机器人开机完成	机器人控制器已完成开机	机器人控制器未完成开机	
安全门状态	安全门打开	安全门关闭	
程序复位状态	程序复位	程序未复位	
外部急停状态	外部急停状态	非外部急停状态	安全板采用 mini 板，且固件版本不低于 1.0.8.7
手持急停状态	手持急停状态	非手持急停状态	安全板采用 mini 板，且固件版本不低于 1.0.8.7
pptomain 执行状态	当前程序正在执行 pptomain	当前程序没有在执行 pptomain 操作	
软急停触发状态	软急停触发状态	非软急停触发状态	
是否处于规划路径上	机器人当前位置在规划路径上	机器人当前位置不在规划路径上	详细使用见寄存器功能码"sta_on_path"
是否处于规划路径附近	机器人当前位置在规划路径附近	机器人当前位置不在规划路径附近	详细使用见寄存器功能码"sta_near_path"

注意：

- 系统输出状态在手动和自动模式下均有效，但出于安全和可用性考虑，建议仅在 xCore 处于自动模式时使用这些信号。
- 某个 IO 点与系统 IO 绑定后，将无法对其进行强制输出或者仿真输入操作。
- 除"工作模式"信号外，其他所有的系统输出均是高电平有效。

- 对于“操作模式”信号，自动模式时输出为高电平，手动模式输出为低电平。

10.3 外部通信

10.3.1 概述

xCore 系统提供了基于 Tcp Socket 的外部通信接口，支持服务端和客户端，上位系统（PLC、MES 等）可以通过该接口向机器人发送控制指令，或者获取机器人状态。

10.3.2 配置

在使用各项交互指令之前，需要先配置 Socket 通信相关参数，并启用该功能。配置界面位于 HMI-“通信”->“外部通信”，如下图所示：

电机上电状态:"motor_on_state"
程序运行状态:"robot_running_state"
急停状态:"estop_state"
工作模式:"operating_mode"
Home 输出状态:"home_state"
故障状态:"fault_state"
碰撞检测状态:"collision_sate"
获取机器人运行任务状态:"task_state"
获取笛卡尔位置:"cart_pos", "cart_pos_name"
获取轴位置:"jnt_pos", "jnt_pos_name"
获取轴速度:"jnt_vel", "jnt_vel_name"
获取轴力矩:"jnt_trq", "jnt_trq_name"
获取缩减模式状态:"reduced_mode_state"
获取IO状态:"io_state"+":"+"IO signal name"
获取报警状态:"alarm_state"
获取碰撞检测报警状态:"collision_alarm_state"
获取碰撞检测开启状态:"collision_open_state"
判断控制器是否开机:"controller_is_running"
编码器低电压报警状态:"encoder_low_battery_state"
获取机器人错误码:"robot_error_code"
获取RL的暂停状态:"program_full"
获取程序复位状态:"program_reset_state"
获取程序运行速度:"program_speed"
获取pptomain执行状态:"robot_is_busy"
获取机器人运动状态:"robot_is_moving"
获取安全门的状态:"safe_door_state"
获取软急停的触发状态:"soft_estop_state"
获取笛卡尔速度:"cart_vel"
获取机器人TCP的位姿:"tcp_pose"
获取机器人TCP的速度:"tcp_vel"
获取机器人TCP的合成线速度:"tcp_vel_mag"
获取外部急停的状态:"ext_estop_state"
获取手持急停的状态:"hand_estop_state"
是否在路径上:"sta_on_path"
是否在路径点附近:"sta_near_path"

电机上电:"motor_on"
电机下电:"motor_off"
程序指针到main:"pp_to_main"
程序启动:"start"
程序停止:"stop"
清除伺服报警:"clear_alarm"
切换到自动模式:"switch_mode:auto"
切换手动模式:"switch_mode:manual"
打开拖动:"open_drag"
关闭拖动:"close_drag"
获取工程列表:"list_prog"
当前工程:"current_prog"
加载工程:"load_prog:prog_name"
急停复位:"estop_reset"
急停复位并清除报警:"estopreset_and_clearalarm"
依次执行上电、程序指针到main、启动程序:"motoron_pptomain_start"
依次执行上电、启动程序:"motoron_start"
暂停程序并下电:"pause_motoroff"
设置程序运行速率:"set_program_speed:speed value"
触发和取消机器人软急停:"set_soft_estop:true/false"
切换自动模式并上电:"switch_auto_motoron"
打开对应的安全区域:"open_safe_region:Index number" (Index number range [1-10])
关闭对应的安全区域:"close_safe_region:Index number"(Index number range [1-10])
开启缩减模式:"open_reduced_mode"
关闭缩减模式:"close_reduced_mode"
设置DO的值:"setdo:IO name, IO valu"
修改控制器和示教器的时间:"set_robot_time:time" (时间格式: YYYY-MM-DD hh:mm:ss)

注意:
Socket 通信接口支持机器人作为客户端或服务端，但同一时刻仅能作为一种状态使用。

当机器人作为客户端使用时，需要配置参数如下:

配置参数	说明
IP	服务端 IP，比如连接的 PLM、MES 系统 IP 地址。
端口	服务端监听端口
后缀	服务端向机器人发送控制指令或者监控指令时，指令结束处需要附加的后缀字符。通常使用简单结束符\r 或者\n 或者\t。注意此处可以使用组合后缀且长度不限制，例如\r\n、\r\t、\r\n\t；也可以使用可见字符，例如字母等。

当机器人作为服务端使用时，支持多连接，如果使用多连接需要客户端注意处理控制时序可冲突。需要配置参数如下:

配置参数	说明
端口	服务端监听端口

后缀	服务端向机器人发送控制指令或者监控指令时，指令结束处需要附加的后缀字符。通常使用简单结束符\r 或者\n 或者\t。注意此处可以使用组合后缀且长度不限制，例如\r\n、\r\t、\r\n\t；也可以使用可见字符，例如字母等。

提示

为了保证机器人运动控制的稳定性，控制系统只将一部分的计算资源分配给了网络通讯功能，当机器人作为 socket 服务器监听链接的时候，如果收到类似"DDOS 攻击"的极度频繁的网络连接请求或者数据流，可能导致机器人和示教器以及别的设备网络连接断开或者操作卡顿。

要求网络连接或者数据交互的频率低于 1 次/ms。

10.3.3 交互指令

交互命令包括控制指令和监控指令。

下表给出了具体的指令内容和格式。（假设用户使用"\r"作为指定的命令结束符，"\r"是转义字符，表示回车，十进制值是 13）。

控制命令：

指令名称	发送的字符串	返回值	备注
关闭 socket 接口	"xCore::SocketInterface::Disable" + "\r"	无返回值	
启动 socket 接口	"xCore::SocketInterface::Enable" +"\r"	无返回值	
启动程序	"start"+"\r"	成功返回"true" 失败返回"false"	当有绑定了暂停功能的寄存器或系统 IO 没有复位时不允许启动；触发功能码后，若示教器报警提示"程序未同步至控制器，启动失败"，请同步程序后，再次触发该功能码；
停止程序	"stop"+"\r"	成功返回"true" 失败返回"false"	
清除伺服报警	"clear_alarm"+"\r"	成功返回"true" 失败返回"false"	
程序指针到 main	"pp_to_main"+"\r"	成功返回"true" 失败返回"false"	
电机上电指令	"motor_on"+"\r"	成功返回"true" 失败返回"false"	
电机下电指令	"motor_off" + "\r"	成功返回"true" 失败返回"false"	
切换至手动模式	"switch_mode:manual"+"\r"	成功返回"true" 失败返回"false"	
切换至自动模式	"switch_mode:auto"+"\r"	成功返回"true" 失败返回"false"	
打开拖动	"open_drag"+"\r"	成功返回"true" 失败返回"false"	仅协作机器人
关闭拖动	"close_drag"+"\r"	成功返回"true"	仅协作机器人

		失败返回“false”	
获取工程列表	list_prog + “\r”	返回工程列表； 无工程返回“null”	
获取当前工程	current_prog + “\r”	返回当前加载工程； 无工程返回“null”	
切换工程	load_prog + (工程名) + “\r”	成功返回“true” 失败返回“false”	
设置 DO 的值	“setdo:” + “IO 名称,IO 的值”+ “\r”	设置成功返回："true\r" 找不到则返回“false\r“	“setdo:DO0-0,true\r” （注意英文标点）
修改控制器和示教器的时间	set_robot_time:time +”\r”（时间格式:YYYY-MM-DD hh:mm:ss）	成功返回“true” 失败返回“false”	
急停复位	estop_reset + “\r”	成功返回“true” 失败返回“false”	(RSC 专用，mini 板无效)
急停复位并清除报警	estopreset_and_clearalarm +”\r”	成功返回“true” 失败返回“false”	
依次执行上电、程序指针到 main、启动程序	motoron_pptomain_start +”\r”	成功返回“true” 失败返回“false”	触发功能码后，若示教器报警提示“程序未同步至控制器，启动失败”，请同步程序后，再次触发该功能码；
依次执行上电、启动程序	motoron_start +”\r”	成功返回“true” 失败返回“false”	触发功能码后，若示教器报警提示“程序未同步至控制器，启动失败”，请同步程序后，再次触发该功能码；
依次执行暂停程序、下电	pause_motoroff +”\r”	成功返回“true” 失败返回“false”	
设置程序运行速率	set_program_speed: +程序速度 +”\r”	成功返回“true” 失败返回“false”	
触发和取消机器人软急停	set_soft_estop:true/false +”\r”	成功返回“true” 失败返回“false”	
依次执行切换自动模式、上电	switch_auto_motoron +”\r”	成功返回“true” 失败返回“false”	
打开对应的安全区域	open_safe_region: +安全区域索引号 +”\r”	成功返回“true” 失败返回“false”	区域索引范围[1-10]
关闭对应的安全区域	close_safe_region: + 安全区域索引号+”\r”	成功返回“true” 失败返回“false”	区域索引范围[1-10]
开启缩减模式	open_reduced_mode +”\r”	成功返回“true” 失败返回“false”	
关闭缩减模式	close_reduced_mode +”\r”	成功返回“true” 失败返回“false”	

监控指令：

指令名称	发送的字符串	返回值	备注
电机上电状态	“motor_on_state” + “\r”	成功返回“true”，电机上电; 失败返回“false”，电机下电；	
程序运行状态	“robot_running_state” + “\r”	成功返回“true”，运行状态; 失败返回“false”，非运行状态；	

急停状态	“estop_state” + “\r”	当急停触发电平类型设置为高时：true,急停状态；false,非急停状态。 当急停触发电平类型设置为低时：true,非急停状态；false,急停状态。	此状态的返回值受急停触发电平类型设置的影响，当设置为高电平时，触发软急停时输出有效，未触发时输出无效；当设置为低电平时，触发软急停输出无效，未触发时输出有效
故障状态	“fault_state” + “\r”	成功返回“true”，故障状态; 失败返回“false”，非故障状态；	
工作模式	“operating_mode” + “\r”	成功返回“true”，自动模式; 失败返回“false”，手动模式/等待模式；	
获取笛卡尔位置	“cart_pos” + “\r”	笛卡尔位置字符串+“\r”；	
获取笛卡尔位置	“cart_pos_name” + “\r”	“cart_pos : ”+笛卡尔位置字符串+“\r”；	
获取轴位置	“jnt_pos” + “\r”	轴位置字符串+“\r”；	
获取轴位置	“jnt_pos_name” + “\r”	“jnt_pos : ”+轴位置字符串+“\r”；	
获取轴速度	“jnt_vel” + “\r”	轴速度字符串+“\r”；	单位:rad/s
获取轴速度	“jnt_vel_name” + “\r”	“Jnt_vel : ”轴速度字符串+“\r”；	单位:rad/s
获取轴力矩	“jnt_trq” + “\r”	轴力矩字符串+“\r”；	单位:N.m
获取轴力矩	“jnt_trq_name” + “\r”	“Jnt_trq : ”轴力矩字符串+“\r”；	单位:N.m
Home 状态输出	“home_state” + “\r”	有输出返回“true”; 无输出返回“false”	
碰撞检测状态	“collision_state” + “\r”	触发碰撞返回“true”; 无碰撞返回“false”	仅协作机器人
获取机器人任务状态	“task_state” + “\r”	机器人当前在进行的任务。包括： ready：就绪； jog ： Jog； load_identify：负载辨识； dynamic_identify：动力学辨识； drag：拖动打开； program：程序运行中； demo ： Demo； rci ： RCI； debug：调试；	详见“HMI 简介章节”中关于机器人当前状态的图标和描述
获取报警状态	“alarm_state”+“\r”	有报警返回 true，无报警返回 false	
获取碰撞检测报警状态	“collision_alarm_state” + “\r”	有报警返回 true，无报警返回 false	
获取碰撞检测开启状态	“collision_open_state” + “\r”	碰撞检测开关开启返回 true 没开启返回 false	
判断控制器是否开机	“controller_is_running”+ “\r”	机器人已开机返回 true 没开机返回 false	
编码器低电压报警状态	“encoder_low_battery_state” + “\r”	有低电压报警返回 true 没低电压报警返回 false	
获取机器人错误码	“robot_error_code” + “\r”	有错误时返回机器的错误码 没有错误码时返回 0	
获取 RL 的暂停状态	“program_full” + “\r”	RL 的暂停状态，包括： 0：初始化状态；	

		1：RL 运行中； 2：HMI 暂停； 3：系统 IO 暂停； 4：寄存器功能码暂停； 5：外部通讯暂停； 6：SDK 暂停； 7：Pause 指令暂停； 10：急停； 11：安全门； 12：其他因素暂停。	
获取程序复位状态	"program_reset_state"+ "\r"	当程序指针在 main 函数第一行，且程序未开始运行时，输出 true；否则输出 false	
获取机器人当前程序运行的实际速度	"program_speed" + "\r"	当前程序运行的实际速度 范围：1~100	
获取程序忙碌状态	"robot_is_busy" + "\r"	如果是正在 pptomain，则返回 true 否则返回 false	获取机器人是否在执行 pptomain 等耗时操作
获取机器人是否在运行（这里不是指程序运行）	"robot_is_moving" + "\r"	如果机器人正在动，则返回 true 否则返回 false	
获取安全门状态	"safe_door_state" + "\r"	如果安全门打开返回 true 否则返回 false	
获取软急停的触发状态	"soft_estop_state" + "\r"	如果触发软急停，则返回 true，否则返回 false	
获取笛卡尔速度	"cart_vel" + "\r"	笛卡尔速度+"\r"	
获取机器人 TCP 的位姿	"tcp_pose" + "\r"	机器人 TCP 的位姿+"\r"	
获取机器人 TCP 的速度	"tcp_vel" + "\r"	机器人 TCP 的速度+"\r"	
获取机器人 TCP 的合成线速度	"tcp_vel_mag" + "\r"	机器人 TCP 的合成线速度+"\r"	
获取外部急停状态	"ext_estop_state" + "\r"	外部急停触发返回"true" 未触发外部急停返回"false"	安全板为 mini 板，且固件版本不低于 1.0.8.7
获取手持急停状态	"hand_estop_state" + "\r"	手持急停触发返回"true" 非手持急停触发返回"false"	安全板为 mini 板，且固件版本不低于 1.0.8.7
获取协作模式状态	"collaboration_state" + "\r"	协作模式触发返回"true" 非协作模式触发返回"false"	
获取缩减开启模式状态	reduced_mode_state + "\r"	开启返回 true，未开启返回 false	
获取 IO 状态	io_state: + IO 名称（多个 IO 用逗号分隔）+ "\r"	返回 IO 的值，用 true 和 fasle 表示，如果没找到对应的 IO，则返回 null	
是否在路径上	"sta_on_path" + "\r"	在路径上返回"true" 不在路径上返回"false"	详细使用见寄存器功能码 "sta_on_path"
是否在路径点附近	"sta_near_path" + "\r"	在路径点附近返回"true" 不在路径点附近返回"false"	详细使用见寄存器功能码 "sta_near_path"

备注：

	字符串格式	单位
笛卡尔位置	x、y、z、a、b、c、q1、q2、q3、q4	x、y、z 单位 mm； a、b、c 单位为度； q1~q4 是姿态四元数；
轴位置	j1、j2、j3、j4、j5、j6、j7	机器人轴角度单位 rad； 导轨位置单位 m；
轴速度	vj1、vj2、vj3、vj4、vj5、vj6、vj7	机器人轴速度单位 rad/s； 导轨速度单位是 m/s；
轴力矩	tj1、tj2、tj3、tj4、tj5、tj6、tj7	机器人轴与导轨力矩的单位都是电机额定力矩的千分比；

10.4 总线设备

10.4.1 总线设备概述

目前支持 CC-Link、Modbus、EtherCAT、PROFINET 四种总线。

CC-Link 包括了 CC-Link 设备（通过 EtherCAT 转接）和 CC-Link IE Field Basic。

EtherCAT 可用于扩展 IO 模块、PROFINET、EtherNet/IP 等其他总线模块。

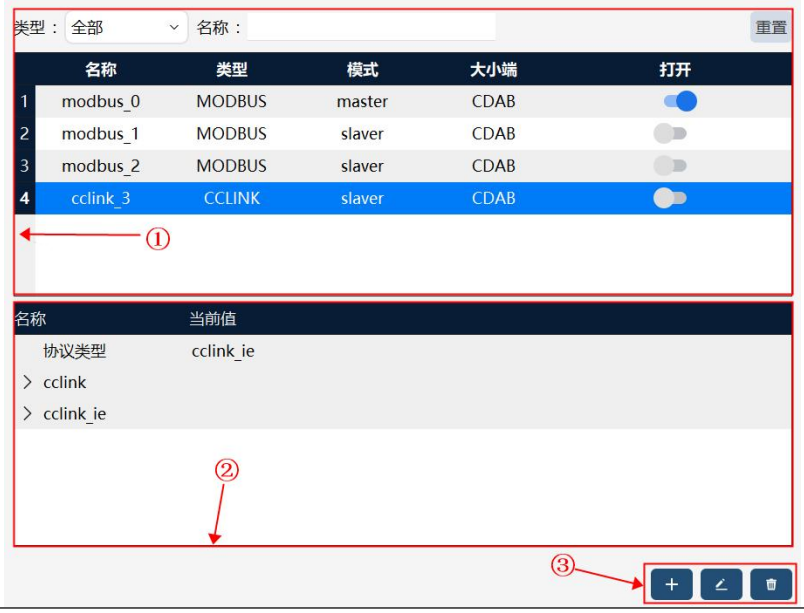
支持总线	协议	支持方式	备注
Modbus	TCP	master 和 slave	
	UDP	不支持	
	RTU	master 和 slave	仅工业机器人
CC-Link	485	远程设备站（slave）	仅工业机器人
	IE Field Basic	远程设备站（slave）	

Modbus 支持功能码如下：

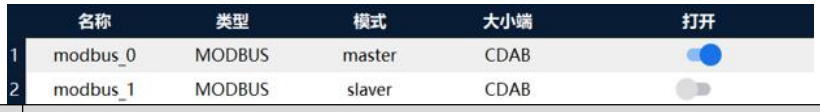
功能码	含义	支持情况
0x01	读线圈	支持
0x05	写单个线圈	支持
0x0F	写多个线圈	支持
0x02	读离散量输入	支持
0x04	读输入寄存器	不支持
0x03	读保持寄存器	支持
0x06	写单个保持寄存器	支持
0x10	写多个保持寄存器	支持

10.4.2 总线设备参数配置

页面位于："通信"->"总线设备"。页面分为上下两部分，上半部分管理所有的总线连接，可以对每条总线连接单独进行打开和关闭操作，当总线连接关闭状态时，"状态监控"->"IO 信号"中将不会显示这个连接配置的 IO；下半部分是当前选中总线设备的属性参数。



①	总线设备列表。
②	当前选中总线设备的参数。
③	总线设备操作按钮，从左至右依次为：新建、编辑、删除。



参数名	参数说明
名称	名称在配置"IO 设备"和"寄存器"时会使用到。 比如下图中，名称分别为 modbus_0、modbus_1、modbus2、 cclink_3。  引用处如下图所示： ● "IO 设备配置"，引用该名称字段，用来表示 IO 设备跟哪个总线相关； ● "寄存器"，引用该字段，表示寄存器与哪条总线相关；

	
类型	在新增/编辑总线设备时可以选择，目前支持 CC-Link、Modbus、EtherCAT、PROFINET 四种。 注意： ● CC-Link 包括了 CC-Link 设备（通过 EtherCAT 转接）和 CC-Link IE Field Basic。 ● EtherCAT 可用于扩展 IO 模块、Profinet、EtherNet/IP 等其他总线模块。
模式	表示当前机器人是作为总线上的主站还是从站。
大小端	主要用于寄存器。由于每个寄存器占用 2 个字节，两个字节的十六进制排列顺序有多种，此属性需要主站和从站对应，否则数据将不符合预期。大小端支持：ABCD、CDAB、BADC、DCBA 四种，默认使用 CDAB 的方式。
打开按钮	可通过该按钮启用或关闭总线功能。支持每个总线设备单独打开或关闭。 注意：关闭某个总线设备后，该总线设备上配置的 IO 将不会显示在"状态监控"->"IO 信号"中。

10.4.2.1 Modbus 通信

在总线设备页面，点击右下角  ，进入新增通信总线设备页面，选择设备类型为"MODBUS"。支持 TCP 和 RTU 两种协议方式，支持主站和从站两种总线配置。

10.4.2.2 Modbus TCP 配置

参数	简介
模式	"master"，机器人作主站；"slave"，机器人作从站；
从站 ID	机器人做从站时，注意总线整体配置，不要与其他从站冲突。 机器人做主站时，此配置表示机器人期望通讯的目标从站 ID。 注意：目前机器人作为主站时，仅支持与外部设备单从站通信。
TCP/IP	机器人作从站时，填入 0.0.0.0 即可，表示监听所有网卡。 机器人做主站时，此项需要填入对应通信的从站的 IP 地址。

TCP 端口	从站使用 TCP 协议时的端口号。
保持寄存器起始地址	功能码 0x03、0x06、0x10 所影响的寄存器的起始地址，1 个寄存器占 2 个字节。 对于只写寄存器，机器人做从站时为保持寄存器功能码 0x03，机器人做主站时为保持寄存器功能码 0x06 或 0x10。 对于只读寄存器，机器人做从站时为保持寄存器功能码 0x06 或 0x10，机器人做主站时为保持寄存器功能码 0x03
保持寄存器个数	从保持寄存器起始地址开始的保持寄存器个数。
线圈起始地址	功能码 0x01、0x05、0x0F 所影响的寄存器的起始地址。
线圈个数	从线圈起始地址开始的线圈寄存器个数。
离散输入起始地址	功能码 0x02 所影响的寄存器的起始地址。
离散输入个数	从离散输入起始地址开始的离散输入寄存器个数。

10.4.2.3 Modbus RTU 配置

部分概念与 Modbus TCP 相同，不重复说明，下面仅介绍不同之处。

RTU 串口名称：表示总线通信应用的串口介质，此项需要在“通信”->“串口设置”中进行配置，包括通信使用的参数。

10.4.2.4 CC-Link 通信

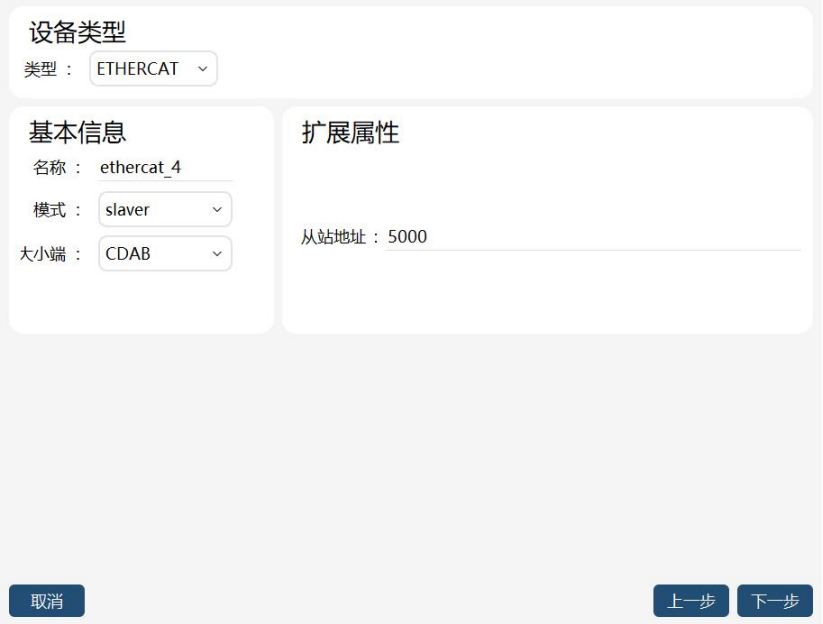
在总线设备页面，点击右下角  ，新增通信总线设备，选择设备类型为”CCLINK”。支持 CC-Link 和 CC-Link IE Field Basic 两种协议方式，仅支持从站总线配置。

10.4.2.5 CC-Link IE 配置

参数	简介
协议类型	cclink_ie 表示直接使用机器人以太网口进行的 CC-Link IE Field Basic 通信协议。
cclink_ie 网卡	配置通信使用哪个以太网卡进行。
cclink_ie 占用站个数	可以配置 1~16 个占用站，默认为 4 个，建议采用默认值。
cclink_ie 协议版本	可选 Ver1 或者 Ver2，此选项需要与主站保持一致。

10.4.2.6 EtherCAT 通信

在总线设备页面，点击右下角 + ，进入新增通信总线设备页面，选择设备类型为"ETHERCAT"。EtherCAT 可以用来转接 PROFINET、EtherNet/IP 等模块网关。



从站地址：此参数为 EtherCAT 拓扑中的从站地址号。

注意：由于机器人内部设备已经占用了 EtherCAT 从站编号 1000~4000，为了避免设备地址冲突，用户自行扩展的 EtherCAT 设备从站编号要求不小于 5000。

10.4.2.7 PROFINET 通信

在总线设备页面，点击右下角 + ，进入新增通信总线设备页面，选择设备类型为"PROFINET"。仅支持从站总线配置。一台机器人支持设置一个 PROFINET 从站，多台机器人可通过修改各自的 profinet 从站名的方式加入同一个 PROFINET 网络，实现多从站的功能。槽 1 至槽 6 的模型选择须和通信伙伴侧配置保持一致。

参数说明：

参数	说明

设备类型	选择 PROFINET。
名称	设备代号。
模式	目前仅支持 slaver。
大小端	一般选择 DCBA，具体选项根据通信双方约定而定。
站名	PROFINET 从站名，需保持和通信伙伴侧一致，不支持中文名设置，不支持大写和下划线。
网卡	选择跟通信伙伴连接的网口；包含了网卡 IP 和网卡名称。
更新周期	默认是 10ms，最小支持 2ms。
槽 1 类型	只能选择 DO_256 模型，表示有 256 个数字量通过槽 1 从机器人输出到通信伙伴。
槽 2 类型	只能选择 DI_256 模型，表示有 256 个数字量通过槽 2 从通信伙伴输入到机器人。
槽 3 类型	选项模型有：AO_Int16_8/AO_Int16_16/AO_Int16_32/AO_Int16_64/ AO_Int16_128/AO_Int16_256。 其中，AO_Int16_8 表示有 8 个 int16 的模拟量通过槽 3 从机器人输出到通信伙伴，其他依次类推。
槽 4 类型	选项模型有：AI_Int16_8/AI_Int16_16/AI_Int16_32/AI_Int16_64/ AI_Int16_128/ AI_Int16_256。 其中，AI_Int16_8 表示有 8 个 int16 的模拟量通过槽 4 从通信伙伴输入到机器人，其他依次类推。
槽 5 类型	选项模型有：AO_Float32_8/AO_Float32_16/ AO_Float32_32/ AO_Float32_64/ AO_Float32_128/AO_Float32_256。 其中，AO_Float32_8 表示有 8 个 float32 的模拟量通过槽 5 从机器人输出到通信伙伴，其他模型依次类推。
槽 6 类型	选项模型有：AI_Float32_8/AI_Float32_16/AI_Float32_32/AI_Float32_64/ AI_Float32_128/AI_Float32_256。 其中，AI_Float32_8 表示有 8 个 float32 的模拟量通过槽 6 从通信伙伴输入到机器人，其他模型依次类推。

10.4.2.8 EtherNet/IP 通信

在总线设备页面，点击右下角  ，进入新增通信总线设备页面，选择设备类型为"EtherNetIP"。仅支持从站总线配置。一台机器人支持设置一个 EtherNet/IP 从站。

参数说明：

参数	说明
设备类型	就是 EtherNetIP。
名称	总线设备名称，不能与其他总线设备名称重复。
模式	目前仅支持 slaver。
大小端	一般选择 CDAB，具体选项根据通信双方约定而定。
网卡	选择与 EtherNet/IP 主站连接的网卡名称，下拉框会显示网卡 IP 及名称。
只读寄存器个数	EtherNet/IP 从站拥有的只读寄存器的个数，一个寄存器代表 2 个字节，可选 32、64、128、248。
只写寄存器个数	EtherNet/IP 从站拥有的只写寄存器的个数，一个寄存器代表 2 个字节，可选 32、64、128、248。

注意事项：

- 仅支持创建一个 EtherNet/IP 总线设备，超过个数或提示错误。
- 寄存器地址均从 0 开始，选择 32，则代表拥有地址 0-31 的寄存器，在配置"寄存器"界面配置寄存器映射时，需要注意地址范围。
- 只读、只写是相对于 xCore 控制系统端来说的，只读寄存器对应 EtherNet/IP 主站的 Output 数据，只写寄存器对应 EtherNet/IP 主站的 Input 数据。
- 只读、只写寄存器个数也代表了通信的数据量，数据越多通信负载越大，在满足需求的情况下，建议尽量选择较小的寄存器个数。
- 主站配置的 Input、Output 数据的字节数应与从站的只写、只读寄存器包含的字节数保持一致，否则可能通信失败。
- EtherNetIP 总线设备只读、只写寄存器是两块单独的数据区，配置的只读、只写的寄存器地址可以重复。
- xCore 控制器系统的 EtherNet/IP 从站默认不提供 EDS 文件，其 Input Assembly Instance ID 为 1， Output Assembly Instance ID 为 2，主站侧在配置时需保持一致，否则可能通信失败。

10.5 寄存器

10.5.1 寄存器概述

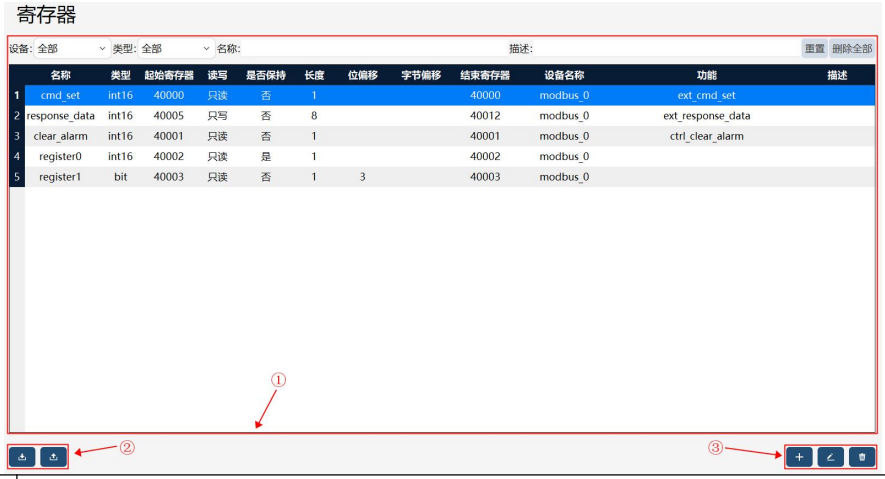
寄存器表示机器人内的可用变量，一般用来与外部设备进行数据通信，从而达到控制机器人、获取机器人状态的目的。寄存器也可以在当前 RL 工程中作为变量使用，支持两种操作寄存器变量的方式：指令式和赋值式。

注意：

- 寄存器是机器人本身的概念，而不是属于总线设备。某个寄存器，可以在其新建或者编辑时，指定绑定到哪个总线设备上，从而进行通信。
- 每个寄存器占 2 个字节。不同类型的变量占用的寄存器数量不同。

10.5.2 寄存器参数配置

在 "通信"->"寄存器"页面，可查看现有寄存器，以及进行添加、编辑、删除操作。



①	寄存器列表。
②	寄存器导入导出按钮，点击后，切换到寄存器导入导出页面，进行导入、导出操作。
③	寄存器操作按钮，从左至右依次是：新建、编辑、删除。

参数	说明
名称	在 RL 中，可以通过该名称，访问寄存器变量。 注意：列表中不可重名，也不可以和 RL 中任何列表中的变量重名，否则 RL 将存在变量冲突，可能产生不可预知的后果。
类型	目前支持六种：bit、byte、bool、int16、float、int32。
起始寄存器	同一总线设备内的同一读写属性的寄存器地址不可交叉占用，另外，Modbus 总线设备内不同读写属性的寄存器地址也不可交叉占用。 例如：定义某寄存器占用 41000~41003，就不能定义另一个寄存器从 41002 开始。
读写	读写属性，表示该寄存器在机器人的视角（而不是主站或从站视角）下是对其进行读操作还是写操作。只写寄存器用于机器人对外输出状态；只读寄存器用于机器人获取外部设备发过来的命令。
是否保持	当寄存器设置为保持时，该寄存器值在机器人重启、关机或者断电、RL 停止时，会保持在非易失的存储介质上。当机器人重新上电或者 RL 重新运行时，此寄存器的值就会恢复成关机、RL 停止运行前的保持值。
长度	长度表示该变量的数量。对于大于 1 的变量，可以使用数组的方式进行变量引用，下标从 1 开始。 注意：此项要区别于寄存器数量。举例：寄存器 40140~40153，该变量类型为 float，每个 float 占用 2 个寄存器，长度为 7，因此占用的寄存器数量为 2*7=14 个。
位偏移	位偏移，代表 bit 类型寄存器映射到寄存器的位置。一个寄存器占两个字节，即 16 位，位偏移是指对应寄存器的位置，偏移值为 1-16。创建 bit 类型寄存器时，如果元素个数为 1 时，可以设置偏移量；反之，则不可以。
字节偏移	字节偏移代表 Byte 类型寄存器映射到寄存器的位置。一个寄存器占两个字节即 16 位，而 byte 变量只需要 8 位。因此，在创建 byte 寄存器变量时，需要选择是映射到寄存器的低地址 8 位(1-8)或者高地址 8 位(9-16)。
结束寄存器	结束寄存器表示寄存器变量所占用的最后一个寄存器地址。在寄存器变量连

	续排布时，通过该值，用户可以快速了解寄存器占用的空间。 举例：可以通过此项的数值加 1，来确定下一个寄存器的起始地址。
设备名称	设备名称是"总线设备"创建时定义的名称，表示寄存器绑定到哪一个总线设备上。支持将寄存器绑定到 CC-Link、CC-Link IE Field Basic、Modbus、EtherCAT 设备上。
功能	这一列中的内容是一些固定的功能码，表示该寄存器对应的机器人功能。功能码分为只读和只写功能码，详见下文寄存器功能码章节。

寄存器列表各列参数说明如下表：

10.5.3 寄存器类型

类型	说明
bit	仅占用一个寄存器某 1 位，bit 数组需要以 16 位的整数倍出现。举例：起始地址 41000 的位 bit 类型寄存器，长度为 64，则占用 41000~41003 这 4 个寄存器。
byte	仅占用一个寄存器的某 8 位，可以选择占据低地址位（寄存器的前 8 位）或者高地址位（寄存器的后 8 位）。创建 byte 寄存器数组时，默认低地址位，不可变更高低地址位。
bool	占用 1 个寄存器。
int16	占用 1 个寄存器。
float	占用 2 个寄存器。
int32	占用 2 个寄存器。注意：设备类型为 PROFINET 时，不支持创建。

bit 类型寄存器的几点说明：

①	元素个数。
②	位偏移。如上图所示，如果元素个数为 1，表示占用一个寄存器某一位，可以设置位偏移数量，可选值为 1-16。

bit 类型的寄存器元素个数大于 1 时，即 bit 变量数组时，不允许设置偏移数量，不允许进行功能绑定。

bit 类型的寄存器输入元素个数大于 1 时，自动隐藏偏移量选项，偏移量设置为 1。

byte 类型寄存器的几点说明：

①	元素个数。
②	字节地址。

如上图所示，如果元素个数为 1，表示占用一个寄存器某 8 位，可以设置字节地址，可选值为：低字节位（1-8）和高字节位（9-16）

byte 类型的寄存器元素个数大于 1 时，即 byte 变量数组，不允许设置字节地址，默认全部是低地址位。

注意：当有绑定了暂停功能的寄存器或系统 IO 没有复位时不允许通过寄存器启动程序。

10.5.4 寄存器功能码

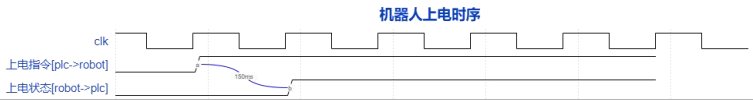
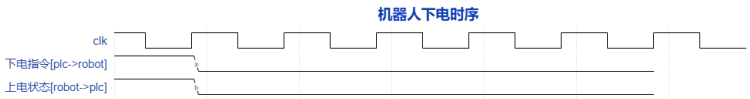
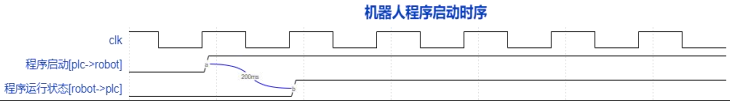
10.5.4.1 只读类型功能码

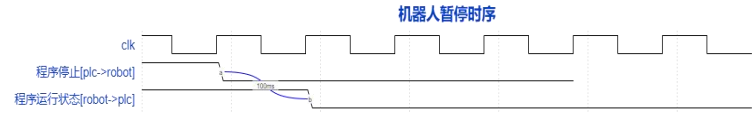
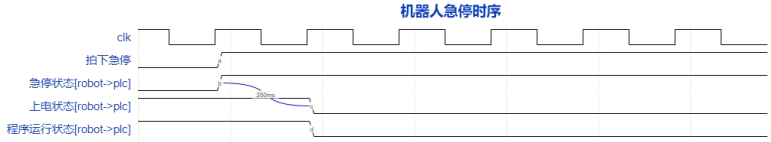
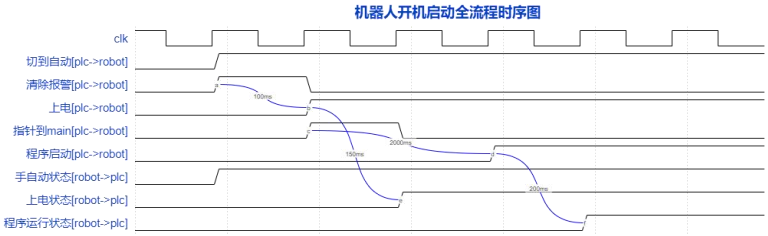
只读类型多用于控制信号，控制信号通常是外部设备发给机器人，指示机器人执行相应的动作。对于机器人来说，这些寄存器是只读的。目前支持的控制信号：

功能码名称	支持绑定类型	对应功能
空白	无	无功能，用户自定义输入。
ctrl_clear_alarm	bit/bool/byte/int16	清除伺服报警。0 到 1 上升沿：清除报警；置 0：复位。
ctrl_estop_reset	bit/bool/byte/int16	急停复位。0 到 1 上升沿：急停复位；置 0：复位。
ctrl_jjwc_A	bit/bool/byte/int16	触发类型为脉冲触发，高电平有效。信号 0->1 时，机器人停止，该信号触发的停止，不能继续运行，只能 pptomain、pptofunc 或 ppto 光标后才能重新运行程序，当机器人停止后，sta_jjwc_B 信号置为 1（高电平）。信号 1->0 时，sta_jjwc_B 信号置为 0（低电平）。
ctrl_motor_off	bit/bool/byte/int16	执行下电操作。0 到 1 上升沿：下电；置 0：复位
ctrl_motor_on	bit/bool/byte/int16	执行上电操作。0 到 1 上升沿：上电；置 0：复位
ctrl_motor_on_off	bit/bool/byte/int16	电机上电或下电：1，上电；0，下电。
ctrl_motoron_pptomain_start	bit/bool/byte/int16	依次执行上电、程序指针到 main、启动程序。 0 到 1 上升沿：下电；置 0：复位。 触发功能码后，若示教器报警提示"程序未同步至控制器，启动失败"，请同步程序后，再次触发该功能码。
ctrl_motoron_start	bit/bool/byte/int16	依次执行上电、启动程序。 0 到 1 上升沿：下电；置 0：复位。 触发功能码后，若示教器报警提示"程序未同步至控制

		器，启动失败”，请同步程序后，再次触发该功能码。
ctrl_pause_motoroff	bit/bool/byte/int16	暂停程序并执行下电操作。0 到 1 上升沿：下电；置 0：复位。
ctrl_pptomain	bit/bool/byte/int16	程序指针到 main。0 到 1 上升沿：下电；置 0，复位。
ctrl_program_start	bool/byte/int16	开始运行 RL 程序。0 到 1 上升沿：下电；置 0，复位。触发功能码后，若示教器报警提示“程序未同步至控制器，启动失败”，请同步程序后，再次触发该功能码。
ctrl_program_start_stop	bit/bool/byte/int16	程序运行/停止。置 1：程序运行；置 0：程序停止。
ctrl_program_stop	bit/bool/byte/int16	停止运行 RL 程序。0 到 1 上升沿：下电；置 0：复位。
ctrl_set_program_speed	bit/bool/byte/int16	设置程序运行速率。输入值为速率，例：输入 10 ，则速率为 10
ctrl_soft_estop	bit/bool/int16	控制机器人软急停，1：不触发软急停；0：触发软急停。
ctrl_switch_auto_motoron	bit/int16/byte/bool	先切换自动模式，然后上电。0 到 1 上升沿：下电；置 0：复位。
ctrl_switch_operation_auto	bool/byte/int16	切换自动模式。0 到 1 上升沿：切换自动；置 0：复位。
ctrl_switch_operation_auto_manu	bit/bool/byte/int16	切换自动模式和手动模式，置 1：自动模式；置 0：手动模式。
ctrl_switch_operation_manu	bit/bool/byte/int16	切手动模式。0 到 1 上升沿：切换手动；置 0：复位。
enable_safe_region01~enable_safe_region10	bit/bool/byte/int16	开启对应的安全区域。0 到 1 上升沿：开启安全区域；置 0：复位
ext_cmd_set	bit/bool/int16	远程控制功能：下发指令见“远程控制”。
ext_request_data	int16 数组	远程控制功能：指令功能码。数组，固定长度 8 寄存器。
ext_reset	bit/bool/int16	远程控制功能：整体功能重置。见“远程控制”。
ext_resp_get	bit/bool/int16	远程控制功能：确认并清除上一条命令的响应。
ctrl_estop_reset_and_clear_alarm	bit/bool/byte/int16	复位急停状态并清除报警。0 到 1 上升沿：复位急停状态并清除报警；置 0：复位。
ctrl_reduced_mode	bit/bool/byte/int16	触发机器人缩减模式。0 到 1 上升沿：触发机器人缩减模式；置 0：复位。

说明：以上系统寄存器，所有的系统输入均为脉冲触发，为了保证 xCore 系统正确接收外部的指令，请保证外部输入的脉冲宽度不小于 60 毫秒。

	前置条件/注意事项	
上电	机器人处于自动模式； 机器人无报警；	
下电	机器人处于自动模式； 机器人无报警；	
程序启动	机器人处于自动模式； 存在机器人程序指针； 机器人处于上电状态； 机器人无报警；	

程序暂停	外部设备通过寄存器发送一个程序暂停操作（不下电）；	
急停处理	拍下急停信号或者触发外部急停时序逻辑。(注意：拍下急停后，程序会停止。同时电机下电，指针丢失，在下一次启动的时候需要先执行清报警，上电、程序到 main 指令。)	
机器人开机启动全流程	操作顺序： (1) 机器人切到手动模式; (2) 清除报警 (清伺服报警或者控制器错误报警）； (3) 电机上电; (4) 指针到 main（工程越大，pptomain 指令耗时越长，建议预留 2s 的执行时间，指令执行完以后再发程序启动信号）； (5)启动程序；	

10.5.4.2 只写类型功能码

只写类型多用于状态信号，状态信号是指机器人对外发出的信号，对外反馈机器人的状态，包括上电状态，程序运行状态等。对于机器人来说，寄存器是只写代表可以绑定该状态信号。目前支持的状态信号有以下几种：

功能码名称	支持绑定类型	对应功能
空白		无功能，用户自定义输出。
ext_error_code	int16	远程控制功能：错误码。
ext_resp_set	bit/bool/ int16	远程控制功能：指令执行完毕的响应。
ext_response_data	int16 数组	远程控制功能：需要反馈的数据。数组，固定长度 8 寄存器。
sta_alarm	bit/bool/byte/int16	伺服报警状态，1：有伺服报警；0：无报警。
sta_board_DI0~sta_board_DI3	bit/bool/byte/int16	实时输出自研 IO 板和实点 IO 每个信号的状态。
sta_board_DO0~sta_board_DO3	bit/bool/byte/int16	实时输出自研 IO 板和实点 IO 每个信号的状态。
sta_collision	bit/bool/byte/int16	碰撞检测状态，1：检测到碰撞；0：无碰撞。
sta_collision_alarm	bool/byte/int16	碰撞检测报警，1：检测到碰撞；0：无碰撞；只有清除报警可清除该报警。
sta_collision_open	bool/byte/int16	碰撞检测开启状态。1：碰撞检测开启；0：碰撞检测未开启。
sta_controller_is_running	bool/byte/int16	控制器开机信号，1：控制器正在运行；0：控制器未运行。
sta_encoder_low_battery	bool/byte/int16	编码器低电压报警状态。
sta_error_code	int16	机器人上报错误码，该错误码与机器人日志中的错误码数值相差

		30000。举例，机器人日志"超出运动范围"的错误码编号 50002，通过 sta_error_code 获取的为 20002。
sta_estop	bit/bool/byte/int16	急停状态 此状态值受急停触发电平类型设置的影响，当设置为高电平时，1：当前触发急停；0：正常；当设置为低电平时，0：当前触发急停；1：正常
sta_heartbeat	bit/bool/byte/int16	心跳信号，只写，在"设置-控制器设置"设置心跳周期。
sta_home	bit/bool/byte/int16	机器人各关节是否在 home 点，1：处于 home 点；0：不在 home 点。
sta_jjwc_B	bit/bool/byte/int16	实时状态输出。触发动作：被动触发。当 ctrl_jjwc_A 信号 0->1 时，sta_jjwc_B 信号置为 1；当 ctrl_jjwc_A 信号 1->0 时，sta_jjwc_B 信号置为 0。
sta_motor	bit/bool/byte/int16	电机上电状态，1：已上电；0：未上电。
sta_operation_mode	bit/bool/byte/int16	当前操作模式，1：自动模式；0：手动模式。
sta_program	bit/bool/byte/int16	当前是否处于程序运行状态，1：程序运行中；0：空闲。
sta_program_full	byte/int16	RL 的暂停状态，0：初始化状态；1：RL 运行中；2：HMI 暂停；3：系统 IO 暂停；4：寄存器功能码暂停；5：外部通讯暂停；6：SDK 暂停；7：Pause 指令暂停；10：急停；11：安全门；12：其他因素暂停。
sta_program_not_run	bool/byte/int16	RL 程序未运行，1：程序未运行；0：程序正在运行。
sta_program_reset	bool/byte/int16	程序复位成功信号，当程序指针在 main 函数第一行，且程序未开始运行时，输出 1；否则输出 0。
sta_program_speed	int16	查询当前程序运行速度，百分比的数值。
sta_robot_is_busy	bit/bool/byte/int16	当前机器人是否在执行 pptomain 等耗时操作，1：正在执行； 0：空闲。
sta_robot_moving	bit/bool/byte/int16	代表机器人是否在运动，1：表示机器人运动中；0：表示机器人静止。只检测 RL 程序中运动指令和 Jog 时，机器人运动状态。（注意：辨识、拖动、力控、拖动回放等情况下，机器人即使在运动，该输出也是机器人静止。）
sta_safe_door	bit/bool/byte/int16	打开安全门时，寄存器绑定状态信号输出有效；关闭安全门时，寄存器绑定状态输出无效。(高电平有效，低电平无效)。
sta_safe_jnt_pos1~sta_safe_jnt_pos8	bool/byte/int16	安全位置触发状态，1：到达安全位置；0：未到达安全位置。
sta_safe_region01~sta_safe_region10	Bool/byte/int16	安全区域触发状态。1 为安全区域触发。
sta_soft_estop	bit/bool/int16	软急停状态的输出。此状态值受急停触发电平类型设置的影响，当设置为高电平时，触发软急停时此状态值为 1，未触发时值为 0；当设置为低电平时，触发软急停时此状态值为 0，未触发时值为 1。
sta_cart_pose	float 数组	查询当前机器人笛卡尔位姿，绑定寄存器的要求：float 数组，长度为 8。
sta_cart_vel	float 数组	机器人笛卡尔速度。
sta_jnt_pose	float 数组	查询当前机器关节角度，绑定寄存器的要求：float 数组，长度为 8。
sta_jnt_trq	float 数组	查询当前机器人关节力矩，绑定寄存器的要求：float 数组，长度为 8,单位 N.m。
sta_jnt_vel	float 数组	查询当前机器人关节速度，绑定寄存器的要求：float 数组，长度为 8,单位 rad/s。
sta_tcp_pose	float 数组	机器人 TCP 的位姿，绑定寄存器的要求：float 数组，长度为 7。

sta_tcp_vel	float 数组	机器人 TCP 速度，绑定寄存器的要求：float 数组，长度为 7。
sta_tcp_vel_mag	float	机器人 TCP 合成线速度。
sta_ext_estop	Bool/byte/int16	外部急停状态 1:外部急停状态；0：非外部急停状态 安全板为 mini 板，且固件版本不低于 1.0.8.7。
sta_hand_estop	bool/byte/int16	手持急停状态 1:外部急停状态；0：非外部急停状态 安全板为 mini 板，且固件版本不低于 1.0.8.7。
sta_sys_stop_di	bit/bool/byte/int16	输出绑定系统暂停信号的值，“暂停程序”和”暂停程序 1”任意一路信号触发 sta_sys_stop_di 输出 1，两路信号都不触发时输出 0。
sta_reduced_mode	bit/bool/byte/int16	机器人是否处于缩减模式。1 为处于缩减模式。
sta_on_path	bit/bool/byte/int16	机器人当前位置是否在预设轨迹上 1：是，如程序运行过程中或点击暂停，机器人在预设路径上。 0：否，如偏离路径（JOG 等）、点击 pptomain、重新加载工程、ppto 光标后。 注意事项： 1、按下急停后，由于存在伺服震荡、下电抖动，控制器认为机器人已偏离预设轨迹，故 sta_on_path 功能码值为 0。 2、打开安全门，触发安全停止后 sta_on_path 功能码值为 0。
sta_near_path	bit/bool/byte/int16	机器人当前位置是否在预设轨迹附近 1：是，满足偏离路径球体半径条件且满足偏离路径球体角度条件。 0：否，其他情况。 在“设置”、“控制器设置”、“高级设置”中设置偏离路径球体半径、偏离路径球体角度参数。 当偏离路径球体半径=0，偏离路径姿态角度=0 时，等同于 sta_on_path

10.5.5 RL 读写寄存器示例

控制系统提供两种方式来对寄存器进行读取、修改操作，分别是：指令式和赋值式。
指令式提供 WriteRegByName 和 ReadRegByName 两条指令。赋值式则更直观简单些，使用操作符“=”。

10.5.5.1 指令式

WriteRegByName(modbus_reg[index], rl_symbol)
modbus_reg 是“通信”->“寄存器”配置的寄存器名称，可以使用[index]在对应寄存器的首地址再进行偏移， index 范围[1 ,寄存器最大长度]， 默认 index = 1。

控制系统中的数据，可以通过寄存器，输出到其绑定的设备中。
举例：控制系统中定义了“int rl_value”，想把它输出到外部设备，可以约定一个寄存器，比如“mtcp_wo_i”的第一个寄存，在 RL 语言中添加一条 WriteRegByName 指令，该值将发送到“mtcp_wo_i”关联的外部设备。

```
GLOBAL PROC main()
    int rl_value = 0;
    while(true)
//生产循环
//控制机器人动作的指令
//...
//...


//计数 + 1
    rl_value+=1;
    WriteRegByName(mtcp_wo_i[1],rl_value);
    endwhile
```

①

ReadRegByName(modbus_reg[index], rl_symbol)

该指令与 WriteRegByName 类似，作用是将寄存器的值，更新到 RL 程序变量中。例如用来控制 RL 程序的执行流程、运动参数等。

10.5.5.2 赋值式

直接使用操作符"="，例如"mtcp_wo_i[1] = 1"是将寄存器 mtcp_wo_i 的第一个元素的值更新为 1。同理，"a = mtcp_wo_i[1]" 是将寄存器 mtcp_wo_i 的第一个元素的值更新到 RL 程序的变量 a 中。

10.5.6 寄存器远程控制

远程控制是一项组合功能，由 7 种不同功能的寄存器配合使用，按照特定的时序可完成复杂业务逻辑交互。外部设备可以通过远程控制功能实现机器人 Jog、更新点位、获取机器人位置、状态等功能。

寄存器功能

外部设备对机器人进行控制时，需要使用 4 种寄存器，对机器人而言属于只读寄存器。

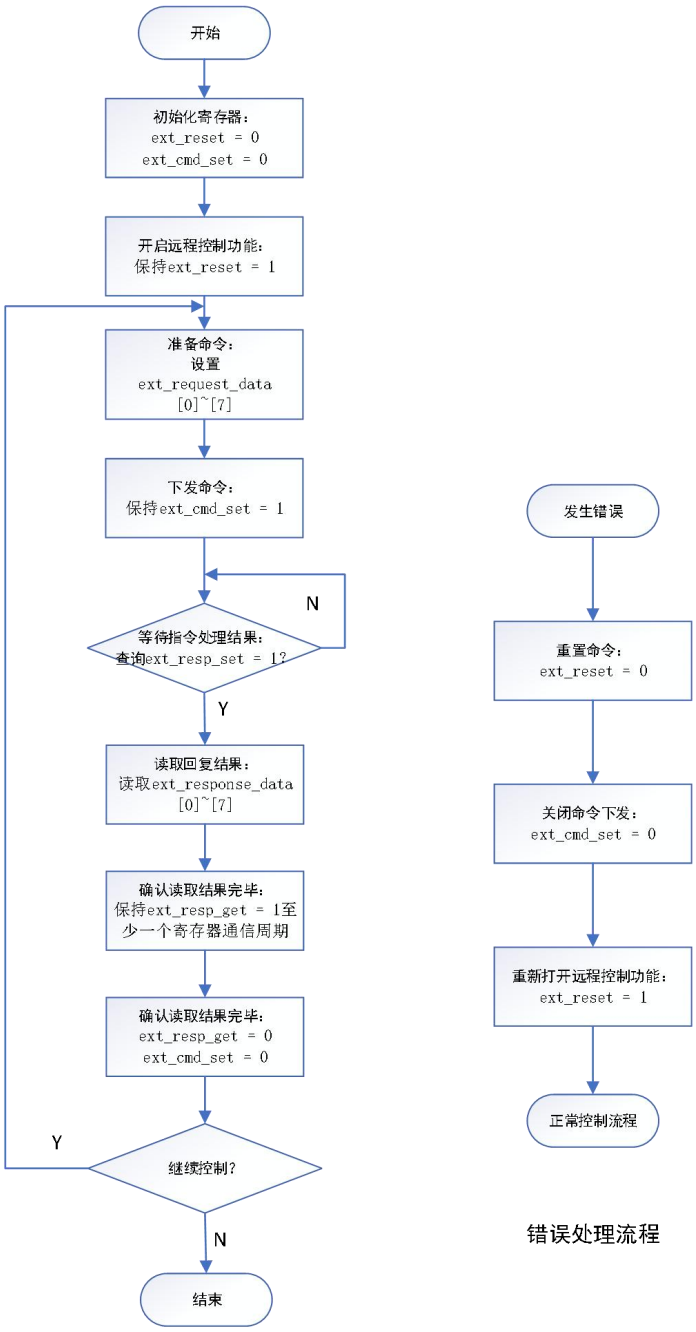
功能码名称	属性	类型	长度	对应功能
ext_cmd_set	只读	int16/bool/bit	1	下发指令： 1、命令执行时通过将请求信号设定为 1 进行请求，当命令为 1 时才响应请求。 2、为了防止误动作，请务必将命令数据设定到数据区域后再执行。（数据暂时放到缓存区，只有该位变为 1 才响应数据）。 3、命令执行完毕后，将该位进行清除(置为 0)。
ext_reset	只读	int16/bool/bit	1	功能重置： 1、用于开启远程控制功能的信号。功能使用过程中，请始终保持该寄存器为 1 状态。 2、0 状态下功能处于停止状态。 3、接口功能发生异常进行重置或中断动作命令时也使用该信号
ext_resp_get	只读	int16/bool/bit	1	确认并清除上一条命令的响应，重置 ext_resp_set 信号为 0。
ext_request_data	只读	int16	8	指令功能码。数组，固定长度 8 寄存器。具体功能详见功能码部分介绍

外部设备希望获取机器人的状态时，需要使用 3 种寄存器，对机器人而言是只写寄存器。

功能码名称	属性	类型	长度	对应功能
ext_error_code	只写	int16	1	远程控制功能：错误码。
ext_resp_set	只写	int16/bool/bit	1	机器人响应控制指令后，会设定该寄存器为 1，表示指令执行完成。
ext_response_data	只写	int16	8	远程控制功能：需要反馈的数据。数组，固定长度 8 寄存器。

10.5.6.1 使用流程

7 种寄存器使用配合及流程控制如下图所示。



错误处理流程

正常控制流程

10.5.6.2 指令格式

指令和响应分别使用 8 个寄存器实现。

命令信号 ext_request_data (占用八个寄存器，reg0 — reg7)，用于指定命令和相关参数的数据信号区域。命令由多字构成：

(1) 字：一个寄存器（16 位） 构成。

(2) 命令形式：命令最多由 8 个字构成。可根据使用的命令变化。最小的命令为 1 个字。

命令编号	命令编号 1	命令编号 2	……	命令编号 7

响应信号 ext_response_data (占用八个寄存器，reg0 — reg7))，用于获取响应的数据信号区域。响应由多字构成：

命令编号	响应 1	响应 2	……	响应 7

(1) 字：一个寄存器（16 位） 构成。

(2) 响应形式：响应最多由 8 个字构成。可根据收到的命令变化。 最小的响应为 1 个字。但是，如果是异常响应则固定占用 3 个字。

具体可用的命令编号如下表：

命令类别	说明	命令码	命令长度	
			命令	响应
JOG	设置 jog 空间	1	2	3
	获取 jog 空间	2	1	4
	设置 jog 速度	3	2	3
	获取 jog 速度	4	1	4
	设置 jog 步长	5	2	3
	获取 jog 步长	6	1	4
	开始 jog	7	4	2
	停止 jog（不带参数）	8	1	2
	更新点位	9	2	2
	运动到点	10	2	2
设置信息	设置工具	11	2	3
	获取当前工具 id	12	1	4
	设置工件	13	2	3
	获取当前工件 id	14	1	4

10.5.6.3 指令详解

（1）设置 jog 空间：

命令/回复	命令码	参数 1	参数 2
设置 jog 空间	1	坐标系: 1: 轴空间 2: 世界坐标系 3: 法兰坐标系 4: 基坐标系 5: 工具坐标系 6: 工件坐标系	N/A
回复	1	设置结果：0 成功；1 失败。	错误码

（2）获取 jog 空间：

命令	命令码	参数 1	参数 2	参数 3
获取 jog 空间	2	N/A	N/A	N/A
回复	2	设置结果: 0 成功；1 失败	错误码	坐标系: 1: 轴空间 2: 世界坐标系 3: 法兰坐标系 4: 基坐标系 5: 工具坐标系 6: 工件坐标系

（3）设置 jog 速度：

命令/回复	命令码	参数 1	参数 2
设置 jog 速度	3	Jog 速度（1~100）	N/A
回复	3	设置结果：0 成功；1 失败	错误码

（4）获取 jog 速度：

命令	命令码	参数 1	参数 2	参数 3
获取 jog 速度	4	N/A	N/A	N/A
回复	4	设置结果： 0 成功；1 失败	错误码	Jog 速度（1~100）

（5）设置 jog 步长：

命令/回复	命令码	参数 1	参数 2
设置 jog 步长	5	1：连续 2：10mm 步长 3：1mm 步长 4：0.1mm 步长 5：0.01mm 步长	N/A
回复	5	设置结果：0 成功；1 失败	错误码

（6）获取 jog 步长：

命令	命令码	参数 1	参数 2	参数 3
获取 jog 步长	6	N/A	N/A	N/A
回复	6	设置结果： 0 成功；1 失败	错误码	1：连续 2：10mm 步长 3：1mm 步长 4：0.1mm 步长 5：0.01mm 步长

（7）开始 jog：

该指令依赖命令码 1：设置 jog 空间。如果是轴空间，参数 1 的值表示的是关节编号（J1~J7，1 代表 J1，7 代表 J7）；如果是笛卡尔空间，参数 1 的值表示的是（x,y,z,a,b,c,elb）的编号（1 表示 x，7 表示 elb）。

命令/回复	命令码	参数 1	参数 2
开始 jog	7	运行方式： 轴空间表示关节编号； 笛卡尔空间（x,y,z,a,b,c,elb）	Jog 方向： 1：负方向 2：正方向
回复	7	设置结果：0 成功；1 失败	错误码

（8）停止 jog：

命令/回复	命令码	参数 1
停止 jog	8	N/A
回复	8	设置结果：0 成功；1 失败。

（9）更新点位：

命令/回复	命令码	参数 1	参数 2

更新点位	9	RL 工程点位列表中的编号	N/A
回复	9	设置结果：0 成功；1 失败	错误码

（10）运动到点位：

命令/回复	命令码	参数 1	参数 2
运动到点位	10	运动方式： 1：MoveAbsj；2：MoveJ；3：MoveL	RL 工程点位列表中的编号
回复	10	设置结果：0 成功；1 失败	错误码

（11）设置当前工具：

命令/回复	命令码	参数 1	参数 2
设置当前工具	11	RL 工程工具列表中的编号	N/A
回复	11	设置结果：0 成功；1 失败	错误码

（12）获取当前工具的编号：

命令/回复	命令码	参数 1	参数 2	参数 3
获取当前工具的编号	12	N/A	N/A	N/A
回复	12	设置结果：0 成功；1 失败	错误码	当前工具编号

（13）设置当前工件：

命令/回复	命令码	参数 1	参数 2
设置当前工件	13	RL 工程工件列表中的编号	N/A
回复	13	设置结果：0 成功；1 失败	错误码

（14）获取当前工件的编号：

命令/回复	命令码	参数 1	参数 2	参数 3
获取当前工件的编号	14	N/A	N/A	N/A
回复	14	设置结果：0 成功；1 失败	错误码	当前工件编号

10.5.6.4 错误码

在进行命令设置时，如参数错误、机器人状态不匹配或者其他情况下，可能发生设置失败。此时可以通过错误码观察机器人出现问题。

控制系统有三种错误码：
指令执行结果的错误码(ext_response_data 中)（命令执行结果）。
ext_error_code 错误码(命令无法执行，比如机器人忙碌，远程控制标志位不正确等)。
sta_error_code，机器人错误码，（jog 运动过程中出现的停止报错，可以读这个寄存器）。

正常情况下用户需要按照流程来使用错误码：
发完执行指令之后（ext_cmd_set=1），先读 ext_error_code，如果没有错误码，那么再读 ext_repose_data 的返回值，如果返回值不为零，那么就读 ext_repose_data 的错误码。
对于运动操作（jog 和运动到点位），如果上面返回值都成功了，还应该读一下 sta_error_code，看运动中是否有错导致的停止（比如奇异点和超限位）。

ext_error_code 含义：

错误码	含义	备注
01	不支持的命令	
02	参数不合法	
03	控制标志不正确	需要确认：ExtRespSet 为 0，ExtRespSet 为 1
04	机器人忙碌	机器人正在执行指令中，禁止响应别的指令
05	找不到对应的编号	工具、点位，工件的 id
06	点位类型和运动类型不匹配	针对运动至到某个点的指令，点位类型和运动类型不匹配，比如关节空间只能用 MoveAbsJ 指令，笛卡尔类型点只能用 MoveJ 或 MoveL
07	输入的轴数和机型不匹配	
11	手自动模式不对	
12	机器人状态不对，请检查机器人是否是 jog 状态	机器人处于 jog 状态时可以被 jog，若处于拖动模式等非 jog 状态，则不允许被 jog。
13	上电状态不对	机器人只有在上电之后才能 Jog。
14	机器人处于非位置模式，不允许 jog	与 12 类似。
15	启动 jog 失败，算法报错	由于诸多原因导致无法 jog 时会报该错误。
20	遇到奇异点	
21	已经运动到目标点	若已经运动至某点再次运动至该点时，会报该错误。

10.5.7 寄存器导入导出

寄存器导入、导出功能可将一台机器人中的寄存器配置快速复制到另一台机器人，无需重新配置寄存器。

10.5.7.1 寄存器导出

在寄存器页面，点击左下角 按钮，进入寄存器导出界面

①	寄存器导出目标文件（包含路径）。
②	寄存器导出选择列表。
③	寄存器导出操作按钮。

导出操作步骤：首先在①中输入或选择寄存器导出目标文件路径，然后在②寄存器列表中，勾

选要导出的寄存器，最后点击③中的下一步按钮执行导出，即可在①中对应路径下生成导出的寄存器文件。

10.5.7.2 寄存器导入

寄存器页面，点击左下角 按钮，进入寄存器导入页面

①	寄存器导入文件。
②	寄存器导入选项，对于冲突项的处理策略。
③	寄存器导入选择条目。
④	寄存器导入操作按钮。

导入操作步骤：首先在①中选择要导入的寄存器文件，其次在②中设置冲突寄存器策略，之后在③中选择要导入的寄存器，最后点击④中的下一步执行导入操作，即可将选择的寄存器导入到本机。

10.5.7.3 寄存器导入时冲突检查

同一设备，同一寄存器属性(读写)，寄存器地址不能一样。如果一样的话，若导入选项设置为不导入，则以原寄存器为准，冲突寄存器不进行导入；若导入选项设置为自动替换，则以新导入的为准，同时把原有冲突的寄存器覆盖，冲突时弹窗提示用户选择是否替换当前寄存器。

以 ext 开头的 7 个寄存器：ext_cmd_set、ext_resp_set、ext_resp_get、ext_reset、ext_response_data、ext_request_data、ext_error_code 在创建时，如果该寄存器被寄存器地址绑定过了，就不能再被别的寄存器绑定了。以上 7 个寄存器在导入时，如果该功能码已经在 HMI 里绑定使用了，且新导入寄存器列表里也有该功能码，以新导入的为准，同时把原有冲突的寄存器覆盖，冲突时弹窗提示用户选择是否替换当前寄存器。

10.6 IO 设备

10.6.1 概述

IO 设备支持四种信号类型：DI、DO、AI、AO。信号来源包括：控制柜自带、EtherCAT 扩展、现场总线扩展等。工业机器人控制柜上自带若干 DI 和 DO。协作机器人底座和末端自带若干 DI 和 DO。工业机器人控制柜上预留了 EtherCAT 扩展接口，可以外接扩展 EtherCAT 模块，产生新的 DI、DO、AI、AO。Modbus 总线扩展也可以配置 IO。

10.6.2 参数配置

可以在“通信”->“IO 设备”页面，查看当前所有 IO 设备，并进行 IO 设备的添加、编辑、删除

操作。

①	IO 设备列表。
②	IO 设备扩展属性。
③	IO 设备操作按钮，从左至右依次是新建、编辑、删除。

在上图中的 IO 设备配置页面，点击③中的新建按钮，进入 IO 设备配置页面，共支持 ETHERCAT、FIELDBUS、ROKAE_IO 三大类 IO 设备，根据设备类型不同，页面参数有所区别。

ETHERCAT-Slave IO 类型设备参数界面

ETHERCAT-SafeBoard Extend IO-DIDO 类型设备参数界面

ETHERCAT-SafeBoard Extend IO-AIAO 类型设备参数界面

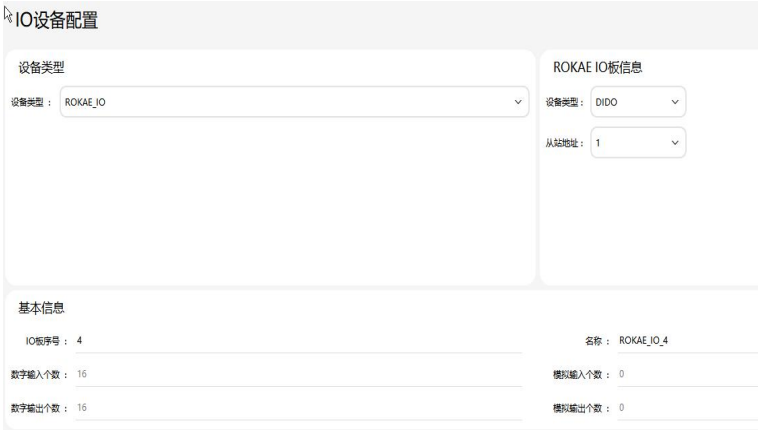
ETHERCAT-ServoBoard Extend IO-DIO 类型设备参数界面

FIELDBUS 类型设备参数界面

IO 设备配置参数界面

参数	说明
设备类型	可选 EtherCAT 和 FIELDBUS。EtherCAT 表示使用 EtherCAT 总线结合扩展模块来实现 IO 扩展。此时仅支持扩展模块作为从站使用，需要配置从站地址。
EtherCAT 从站信息-设备类型	可选 SafeBoard IO、SafeBoard Extend IO、xPanel IO、Slave IO、ServoBoard Extend IO。SafeBoard IO 表示机器人安全板上自带的 DI 和 DO。SafeBoard Extend IO 表示机器人安全板上的扩展 IO，一般是指 XBC_5 控制柜安全板的扩展 IO。xPanel IO 仅适用于协作机器人，表示末端的的 DI 和 DO。Slave IO 表示 EtherCAT 扩展模块。ServoBoard Extend IO 表示机器人伺服扩展 IO，是指 XBC_6 控制柜内伺服的扩展 IO。
EtherCAT 从站信息-IO 板类型	当 EtherCAT 从站信息-设备类型选择为 SafeBoard Extend IO 时，会出现 IO 板类型选项，提供可供选择的安全板扩展 IO 板，可选 DIO16_1、DIO16_4、DIO16_5、DIO16_6、AIAO4_2、AIAO4_3，选项的最后一位编号代表安全板扩展 IO 板的地址。根据实际接入的安全板扩展 IO 选择即可。除了用户手动编辑外，控制器可以自动识别并添加安全板扩展 IO。当 EtherCAT 从站信息-设备类型选择为 ServoBoard Extend IO 时，会出现伺服 IO 板位置选项，可选槽 1-6 对应控制柜上实际扩展槽的位置，IO 板类型暂时仅支持 DIO。除了用户手动编辑外，控制器可以在启动时自动识别并添加伺服扩展 IO。
EtherCAT 从站信息-从站地址	扩展模块在 EtherCAT 总线拓扑中的从站地址。不可以与安全板、关节、协作机器人末端的地址冲突。
FIELDBUS-总线设备名称	“总线设备”页面中，创建总线连接时用户自定义的名称，用来和“总线设备”进行关联。
ROKAE_IO 设备类型	可选 DIDO、AIAO 两种，分别对应珞石自研的数字量、模拟量 IO 板，注意设置时要和实际的 IO 板类型保持一致。
ROKAE_IO 从站地址	接入的自研 IO 板的地址，可选 1 至 15，不能重复，注意设置时要和实际 IO 板的拨码地址相对应，否则可能出现状态异常。
IO 板序号	每一个 IO 设备配置都将生成一块虚拟的 IO 板，控制系统内部对 IO 板进行分类和管理。IO 板序号是进行虚拟 IO 板管理的唯一编号。
名称	虚拟 IO 板的自定义名称。改名称会在“状态监控”->“IO 信号”中用于筛选条件。
数字输入个数	DI 数量。
数字输出个数	DO 数量。
数字量 IO 配置	当 EtherCAT 从站信息-设备类型选择为 ServoBoard Extend IO 时， IO 板类型选择为 DIO，会出现数字量 IO 配置选项，可配置 DIO0-7 和

	DIO8-15 的输出类型，可选择配置为 NPN 或者 PNP。注意此配置需要重启控制器生效。
模拟输入个数	AI 数量。
模拟输出个数	AO 数量。
模拟量 IO 配置	当 EtherCAT 从站信息-设备类型选择为 SafeBoard Extend IO 时，IO 板类型选择为 AIAO4_2 或 AIAO4_3，会出现模拟量 IO 配置选项，可将每个模拟量通道配置为电压型或电流型。

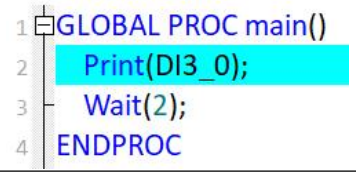
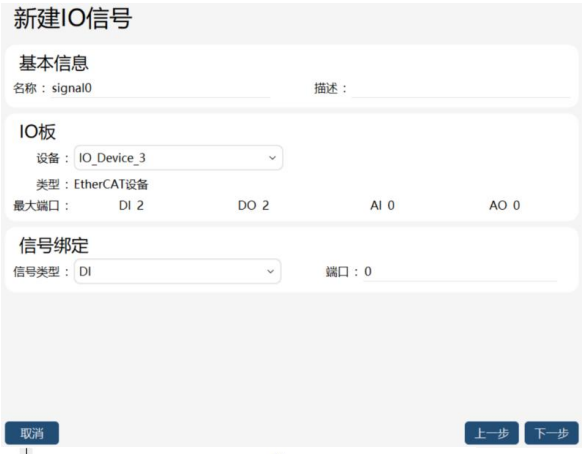
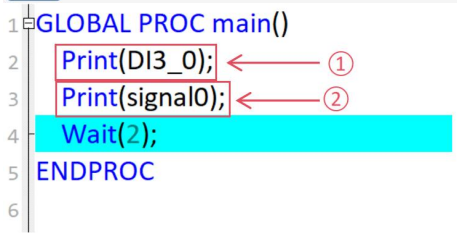


ROKAE_IO 类型设备参数界面

“状态监控”->“IO 信号”中，可以对创建的 DI、DO、AI、AO 进行观察。可以通过“虚拟 IO 板名称”对 IO 信号进行过滤，此时仅显示当前虚拟 IO 板所配置的 IO 信号。可以通过信号类型进行过滤，只显示 DI、DO、AI、AO 中某一类型的信号。

虚拟 IO 板配置好，IO 信号会生成默认的名称。在 RL 有两种使用方式：一是用默认名称；二是起别名，在 RL 中使用新的名称。

用默认值	3 号 IO 板默认生成了 DI3_X 的 IO 信号，如下图，在 RL 程序中可直接使用 DI3_0 进行处理：

	
起别名	在"编程"->"IO 信号列表"中对 IO 信号起别名：   上图中①和②语句效果相同。

10.6.3 Modbus 扩展 IO 示例

当需要使用真实的 IO 信号来与外设进行交互时，推荐使用转接模块，转换模块连接到控制柜上。可以联系珞石官方，获取推荐的 Modbus IO 模块。该模块为 Modbus TCP 从站，以线圈方式进行控制，机器人需要配置为 Modbus Master，并开启线圈功能。

按照现场总线和扩展 IO 的配置方式，使用 Modbus 扩展模块扩展真实 IO 时，需要先配置"总线设备"，然后配置"IO 设备配置"。

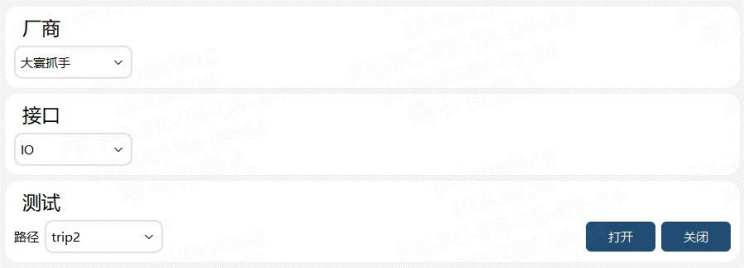
步骤	图示	说明
1.配置"总线设备"。		主从模式选择"主站"，机器人作为主站。从站 ID，表示 Modbus IO 模块从站 ID。机器人作为主站时，IP 填写 Modbus IO 模块的 IP，端口号填写 Modbus IO 模块的端口号。 下一步：参数配置完成后，点击"下一步"，完成参数配置。

2.配置"IO 设备配置"。		设备类型选择"FIELDBUS"。 总线设备名称，选择刚才进行"总线设备"配置时，自定义的名称。 基本信息：数字输入/输出、模拟输入/输出，会在总线设备配置页面配置好，因此采用自动获取，无需配置。
3.开启功能和状态监控。	 	刚配置好"总线设备"和"IO 设备配置"时，状态监控中是不会展示配置好的扩展 IO 的，因为此时还未开启总线连接。如果要使用这些 IO，必须开启总线连接，使控制柜与扩展 IO 模块正确建立连接和保持通信。 如左图，开启总线设备后，控制器与扩展模块建立连接，正常工作，此时 IO 信号的状态监控中将 可以看到 modbus_0 这个总线设备的 IO，并且其配置的 DI、DO 数量与总线设备中配置的一致。

10.7 末端工具

xCore 控制系统支持通过末端工具对大寰爪手进行开合控制，末端工具接口支持 IO 通讯和 RS485 通讯。该功能仅适用于协作机器人 xMate ER 系列。

在 HMI 主界面通过菜单"通信"->"末端工具"，可以进入 RCI 设置界面，如下图所示：



末端工具 IO 接口配置

末端工具 RS485 接口配置

相关设置参数说明如下：

参数设置	值/说明
厂商	选择大寰爪手即可。
接口	通讯协议，可选 IO 或者 RS485 控制。
路径	爪手包括两组行程属性 trip1 和 trip2，包含张开闭合位置和张开闭合力。
最大位置	爪手张开的最大位置，单位是百分比。仅 RS485 支持。
最小位置	爪手张开的最小位置，单位是百分比。仅 RS485 支持。
外撑力	爪手张开时使用的力，单位是百分比。仅 RS485 支持。
夹持力	爪手闭合时使用的力，单位是百分比。仅 RS485 支持。

设置好参数后，可以使用"打开"、"关闭"按钮，打开或关闭抓手。

注意：RS485 支持对夹爪 trip 参数设置，IO 控制时，其 trip 参数只能通过大寰通讯转接盒设置。

10.8 RCI 设置

RCI 是外部控制接口，使用前需要进行 RCI 通信设置。仅协作机型支持。

在 HMI 主界面通过菜单"通信"->"RCI 设置"，进入 RCI 设置界面，如下图所示：

需要设置的参数如下表所示：

配置参数	说明
IP	如果用户 PC 与机器人通过网线直连，那么用户 PC 的 IP 地址应该与机器人的 IP 处于同一网段；如果用户 PC 与机器人通过无线或者交换机的方式连接，那用户 PC 应该与机器人处于同一局域网内。
端口	端口号默认设置为 1337。
包丢失阈值	包丢失阈值为百分比，代表 RCI 通信过程中的丢包率，例如，包丢失阈值设置为 10，代表 RCI 使用过程中的丢包率不得超过 10%。建议丢包阈值设置范围在 10~20 之间。

详细的 RCI 使用方法和例程参考《RCI 用户使用手册》。

10.9 xPanel 设置

xPanel 设置可对机器人的末端工具进行模式设置，该功能仅适用于协作机器人 xMate CR SR 系列。

在 HMI 主界面通过菜单"通信"->"xPanel 设置"，可进入 xPanel 设置界面，如下图所示：

可设置的参数如下表所示：

配置参数	说明
对外供电模式	设置末端的对外供电模式
模拟输入或 RS485 模式选择	选择末端使用模拟输入或者 RS485 串口
数字输出 DO0 模式	将对应的末端 DO 输出设置为 NPN 或 PNP 模式
数字输出 DO1 模式	将对应的末端 DO 输出设置为 NPN 或 PNP 模式
模拟输入 AI0 模式	将对应的模拟量输入设置为电压型或电流型
模拟输入 AI1 模式	将对应的模拟量输入设置为电压型或电流型

设置好需要的参数后，点击"确定"按钮，设置即可生效。

注意：外部接入的模拟量输入信号的电压或电流类型，请与对应的模拟输入模式保持一致，否则可能造成意外错误。

10.10 电爪吸盘

10.10.1 概述

xMate CR、ER、SR 机器人支持末端 RS485 通信，目前已适配多款电爪和吸盘。该界面主要用进行已适配过的电爪、吸盘的配置和测试。

注意：

- 该功能仅适用于 xMate ER 系列、xMate CR 系列、xMate SR 系列；
- 老版本的机器人末端板卡可能不兼容该功能，请联系珞石官方升级板卡固件；
- xMate CR 机型，使用前，请先确认 xPanel 末端参数配置正确；

10.10.2 配置

在 HMI 主界面通过菜单"通信"->"电爪吸盘"，进入电爪吸盘设置界面，如下图所示：

“基本信息”设置参数说明如下：

参数	说明
厂商	目前支持钧舵、增广、Robotiq、大寰；
系列	钧舵，目前支持 EPG 系列电爪、EVS 系列吸盘； 增广，目前支持 RM-RMG、RM-C 两个系列电爪； Robotiq，目前支持 2F85 系列； 大寰，目前支持 PGI 系列；
通信	目前仅支持 RS485 接口；

10.10.2.1 钧舵电爪

在“基本信息”中，厂商选择钧舵， 系列选择 EPG，HMI 会切换至 EPG 系列电爪的测试界面，如下图所示：

初始化：测试某个 ID 的电爪，第一步需要输入电爪设备的 ID，并单击“初始化”按钮，如果电爪执行自动检测，并提示初始化成功，说明电爪硬件连接和通信正常，可以进一步测试和使用。

工具测试：电爪完成初始化后，可以单击“移动至”按钮，控制电爪按照指定速度和指令力向指定位置运动，如果电爪达到指定位置或者遇到物体达到指定力后，会停止运动，电爪的接触检测状态也会显示在测试界面上。

相关参数设置说明如下：

参数	值/说明
工具 ID	输入电爪设备的 ID，此 ID 是使用钧舵调试软件设置的电爪的 ID
工具位置	设置爪手的位置，范围 0-255
工具速度	设置爪手的运动速度，范围 0-255
工具力矩	设置爪手的运动力，范围 0-255

10.10.2.2 钧舵吸盘

在“基本信息”中厂商选择钧舵， 系列选择 EVS，HMI 会切换至 EVS 系列吸盘的测试界面，如下图所示：

初始化：测试某个 ID 的吸盘，第一步需要输入吸盘设备的 ID，并单击“初始化”按钮，如果提示初始化成功，说明吸盘硬件连接和通信正常，可以进一步测试和使用。

工具测试：吸盘完成初始化后，需要按需设置吸盘的参数，输入参数完成后，可以单击“设置”按钮，测试吸盘的工作状态。

设置参数说明如下：

参数	值/说明
通道选择	吸盘支持 2 个通道，可以任意选择 2 个通道的生效状态
最小真空度	设置吸盘目标真空度，当吸盘内真空度达到此值时停止工作
最大真空度	设置吸盘目标真空度，当吸盘内真空度大于此值时开始工作
超时时间	指定时间未能达到指定最小真空度，将会超时

10.10.2.3 增广电爪

在“基本信息”中厂商选择增广， 系列选择 RM-RGM 或 RM-C，HMI 会切换至增广系列电爪测试界面，如下图所示：

初始化：测试某个 ID 的电爪，第一步需要输入电爪设备的 ID，并单击“初始化”按钮，如果电爪配置正确并且通信正常，会初始化成功，可以进一步测试和使用。

当前工具 ID：显示当前已经初始过的当前类型的设备 ID 列表，选择对应 ID 的设备时，可以看到选中设备的状态、测试选中的设备。

工具状态：初始化成功后，增广电爪可以显示工具的状态包括：位置（单位 mm）、速度（单位 mm/s）、力矩（百分比）、到位情况（1 表示已到位）、错误报警（错误码信息）。

工具测试：可以选择在位置模式和力矩模式测试增广电爪的运行，按照电爪厂商的说明书要求，电爪的释放应采用位置模式，电爪的闭合应采用力矩模式。

位置模式：可以设置的参数包括位置（绝对定位位置，单位 mm）、速度（mm/s）、定位范

围（认为到位的误差范围，单位 mm）和加速度（单位 mm/s2）。如上图所示。

力矩模式：可以设置的参数包括距离（相对距离，单位 mm）、速度（mm/s）、力度（百分比）、加速度（单位 mm/s2）、定位范围（认为到位的误差范围，单位 mm）和时间范围；如下图所示：

单击"移动至"按钮，电爪即可按照设置的模式和参数进行运动。

10.10.2.4 Robotiq 2F_85 电爪

在"基本信息"中厂商选择 Robotiq， 系列选择 2F_85，HMI 会切换至 Robotiq2F_85 电爪测试界面，如下图所示：

初始化：测试某个 ID 的电爪，第一步需要输入电爪设备的 ID，并单击"初始化"按钮，如果电爪执行自动检测，并提示初始化成功，说明电爪硬件连接和通信正常，可以进一步测试和使用。

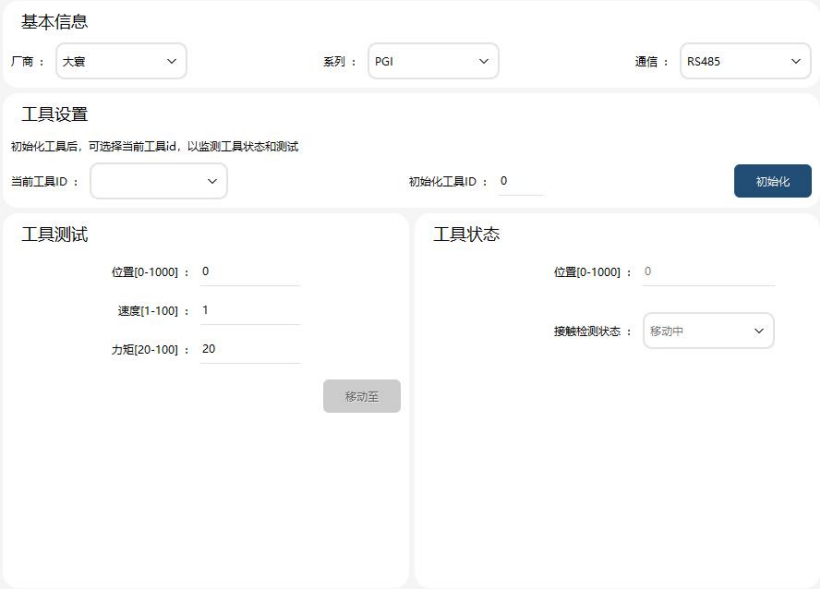
工具测试：电爪完成初始化后，可以单击"移动至"按钮，控制电爪按照指定速度和指令力向指定位置运动，如果电爪达到指定位置或者遇到物体达到指定力后，会停止运动，电爪的接触检测状态也会显示在测试界面上。

相关参数设置说明如下：

参数	值/说明
工具 ID	输入电爪设备的 ID
工具位置	设置爪手的位置，范围 3-255
工具速度	设置爪手的运动速度，范围 0-255
工具力矩	设置爪手的运动力，范围 0-255

10.10.2.5 大寰电爪

在"基本信息"中，厂商选择大寰， 系列选择 PGI，HMI 会切换至 PGI 系列电爪的测试界面，如下图所示：



初始化：测试某个 ID 的电爪，第一步需要输入电爪设备的 ID，并单击"初始化"按钮，如果电爪执行自动检测，进行一次开合动作，并提示初始化成功，说明电爪硬件连接和通信正常，可以进一步测试和使用。注意：初始化的过程中，夹爪进行的开合动作将确认夹爪的运行范围，以后夹爪的运行将以初始化的运行范围为准。

工具测试：电爪完成初始化后，可以单击"移动至"按钮，控制电爪按照指定速度和指令力向指定位置运动，如果电爪达到指定位置或者遇到物体达到指定力后，会停止运动，电爪的接触检测状态也会显示在测试界面上。

注意：移动过程中夹持到物体的话，此时即使将夹爪的物体拿开，夹爪也不会继续移动，必须重新给定一个新的目标位置夹爪才能继续运动。

夹爪状态共有五个，对应状态值为 0（移动中）、1（夹爪未夹持到物体）、2（夹持到物体）、3（发现物体掉落）、4（无显示，即当前 id 未初始化成功）。

相关参数设置说明如下：

参数	值/说明
工具 ID	输入电爪设备的 ID，此 ID 是使用大寰调试软件设置的电爪的 ID
工具位置	设置爪手的位置，范围 0-1000，初始化确定的行程范围的千分比
工具速度	设置爪手的运动速度，范围 0-100，速度百分比
工具力矩	设置爪手的运动力，范围 20-100，出力的百分比

10.11 串口设置

用户可以使用串口的方式和外部设备进行通信。使用串口需要硬件设备支持，工业机器人的 XBC5 控制柜，柜体上预留了一路 RS-232 串口；也可以借用控制柜预留的 USB 接口，使用 USB 转 RS-232 的接口模块来进行串口通信。协作机器人无相关硬件接口，不支持该功能。

在 HMI 主界面通过菜单"通信"->"串口设置"，可以进入串口设置界面，如下图所示：

①	串口显示过滤。
②	串口设备列表。
③	串口编辑按钮，从左至右依次是：新建、编辑、删除。

点击新建按钮，进入新建串口页面，如下图所示：

串口使用前，需要配置的参数说明如下：

参数	说明
名称	用户自定义名称，在 RL 中操作该串口资源使用的唯一标识符。 注意：该名称受 RL 工程中名称冲突限制，不可以和工程中已有的网络标识符同名，不可以和已有的其他串口标识符同名。
端口	系统端口。控制系统会列举扫描到的串口资源（包括 USB 形式转换的串口），供用户选择使用。
波特率	可选：1200/2400/4800/9600/19200/38400/57600/115200。
数据位	可选：5/6/7/8 位。
停止位	可选 ：1/1.5/2 位。
校验位	可选：奇校验（Odd）/偶校验（Even）/全 1 校验（Mark）/全 0 校验（Space）/无校验（None）。

配置完参数后，点击"下一步"按钮即可完成完串口配置，此时在 RL 程序中使用串口。串口功能包含一系列指令，详见 RL 指令中关于串口指令的详细说明。

注意：请尽量保证串口通信两端的参数设置一致，否则可能会导致收发数据异常。

10.12 编码器

此功能属于传送带跟踪功能的一部分，详细使用请参考《传送带跟踪功能使用手册》。

10.13 OPC-UA

10.13.1 概述

xCore 控制系统的 OPC-UA，目前支持 xCore 控制器作为 OPC-UA 通信的服务端，并默认支持 OPC 40010-1 OPC UA for Robotics, Part 1: Vertical Integration 标准里的所有强制性节点及部分可选节点，其详细模型介绍请参考附录 OPC-UA 相关部分。另外,还支持用户自定义变量及 event 事件上送功能。

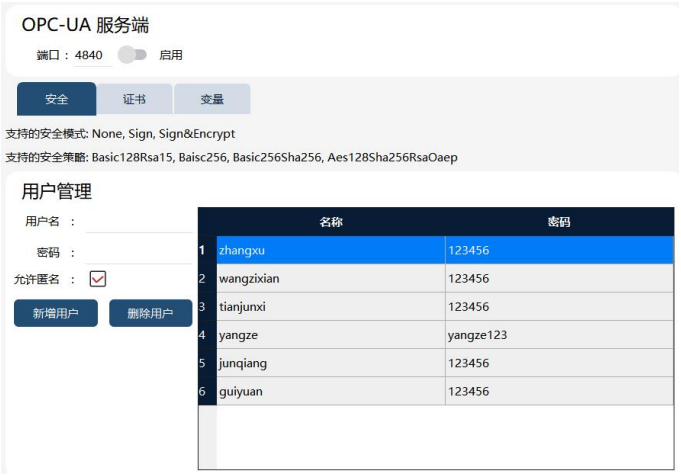
10.13.2 打开关闭

在 HMI 主界面通过菜单"通信"->"OPC-UA"，进入到 OPC-UA 配置界面，如下图，可以在上方的"端口"编辑框输入 OPC-UA 服务的端口，并通过"启用"按钮打开或关闭 OPC-UA 服务。

当 OPC-UA 服务正常打开后，可以在状态监控的网络连接里看到 OPC-UA 服务处于监听状态，如下图所示。

10.13.3 安全

在 OPC-UA 配置界面点击"安全"选项卡，进入安全配置界面，如下图所示：



此界面主要用于说明及配置 OPC-UA 服务端安全的策略。

1、默认支持 None、Sign、Sign&Encrypt 三种安全模式，在客户端连接时选择需要的模式即可。

2、默认支持 Basic128Rsa15、Basic256、Basic256Sha256、Aes128Sha256RsaOaep 四种安全策略，在客户端连接时选择需要的策略即可。

3、勾选“允许匿名”复选框，可以允许客户端进行匿名登录，若没有勾选，则客户端只能使用右边列表中的用户进行登录。

4、输入用户名与密码，点击“新增用户”按钮，即可在右边列表中添加新的用户。

5、在右边列表中选中用户，点击“删除用户”按钮，可以删除不需要的用户。

10.13.4 证书

在 OPC-UA 配置界面点击“证书”选项卡进入证书配置界面，如下图所示：

1、默认情况下，若不导入 OPC-UA 服务端的证书及私钥，控制系统将使用自生成证书和私钥。

2、点击“导入证书”按钮可以导入服务端证书，点击“导入私钥”按钮可以导入服务端的私钥，导入的证书和私钥都需要是 der 格式，并保证证书和私钥是匹配的。另外，需注意导入的服务端证书的 URL 必须是：urn:xcore.opcua.server 。

3、点击右方“导入”按钮，可以导入受信任的客户端证书，点击“删除”按钮可以删除选中的客户端证书。

10.13.5 自定变量配置

在 OPC-UA 配置界面点击“变量”选项卡进入自定义变量配置界面，如下图所示：

xCore 控制系统的 OPC-UA 通信支持自定义变量功能，支持 bool、int、double、string 四种变量类型，并支持配置客户端是否可写属性，其具体的配置功能如下：

1、在页面左边的填写好"名称"、"描述"、"类型"、"可写"、"初始值"等属性，点击"新建"按钮，可以在右侧列表新增一个自定义变量。目前最多支持 128 个自定义变量。

2、点击"删除"按钮可以删除右侧列表中选中的变量。 点击"清除"可以删除右侧列表中的所有变量。

3、勾选右侧变量列表下的"开启监控"复选框，可以开启变量监控，"当前值"会实时显示变量在控制器中的值。点击"修改值"按钮可以修改选中变量的当前值。

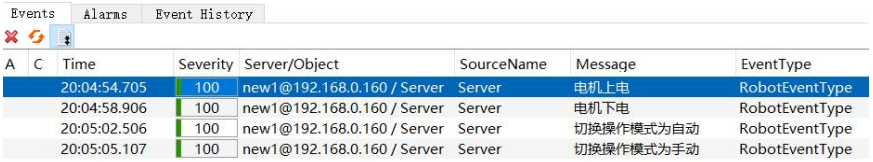
客户端可以在 Robotics 模型中的 CustomVariables 节点下找到自定义变量的节点，并进行读写操作；在机器人侧，用户可以使用 ReadOpcUaVarByName、WriteOpcUaVarByName 两个指令在 RL 程序中读取或修改 OPC-UA 自定义变量，其详细使用方法详见 RL 指令章节中相应的使用说明。

10.13.6 事件(event)

OPC-UA 服务端支持将机器人的一些状态变化通过事件(event)方式通知给 OPCUA 客户端。当前支持的上报事件如下表所示。

事件	等级(Severity)
上、下电	100
手、自动切换	100
程序运行、停止	100
急停触发	600
安全门停止触发	600
保护停止触发	600
碰撞检测报警	600

客户端事件的显示如下图所示：



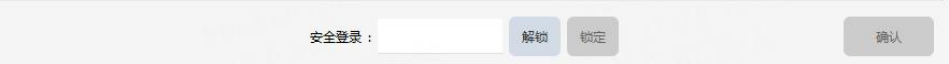
11 安全

11.1 本章简介

本章主要介绍 xCore 安全相关功能的设置。

11.2 安全密码

必须设置一个密码来解锁安全模块的安全设置。默认密码为“safety”。



1、 用户每次进入安全模块，需输入密码并点击解锁才可操作安全界面；
2、 解锁后，进入安全模块的其他界面无需重新输入密码；
3、 切换到其他模块再切回安全模块需重新解锁；
4、 用户修改安全模块的设置，修改完成后需要点击“确认”按钮才可设置安全参数；
5、 安全模块不再受用户权限影响，操作员等低权限用户，安全登录后，也可以对安全模块参数进行修改。

安全密码可在“设置-用户组-安全密码”处进行更改

11.3 关节限制

11.3.1 功能简介

关节限制会对机器人各关节的参数进行监控，当任一关节的状态超出设置阈值时，机器人会立刻停止运行；RSC 机器人会进入安全停止状态。
关节限制主要包含关节位置、关节速度、关节扭矩和关节功率四项限制功能；每一项限制功能都可设置两套参数，根据当前模式（正常模式或缩减模式）会采用不同的参数判断阈值。
用户可以根据需要，自行开启或关闭某个功能。

11.3.2 关节位置

11.3.2.1 关节位置功能简介

关节位置限制是从软件层面设置机器人各轴最大运动范围的功能，避免机器人与周边设备发生干涉或者碰撞。
在拖动过程中，各轴关节角度也受到关节位置保护，拖动到关节限位附近，会给机械臂一个远离关节限制方向的反弹力，产生拖动反弹力的范围是 HMI 界面设置的软限上下限的 10°范围内，假设轴 1 设置的关节位置为-170°~170°，那么产生拖动反弹力的范围是[-170°~-160°]以及[160°~170°]。

①	“启用”开关控制关节位置是否启用；
②	正常模式下的关节位置限制，下限需要小于上限；
③	缩减模式下的关节位置限制，缩减关节位置范围需要小于等于正常模式关节位置；
④	本机可设置关节位置的最大范围。

警告

1.关节位置范围不能超过机器人本体所允许的机械硬限位范围。

2. 协作机器人在伺服固件支持机械最大运动范围的情况下，关节位置可设置范围为各轴机械最大运动范围。

11.3.2.2 超出关节位置限制后的处理

在一些极少见的情况下，机器人可能会运动到关节位置限制之外，例如：机器人在运动到限位边界时触发急停，机器人在执行 STOP 0 时可能会超出关节限制。

在 xCoreV2.1 及之前的版本，当机器人有一个或者多个关节在关节位置限制之外时，将无法进行 Jog 和运行程序。此时需要首先取消关节位置限制，然后将超限的关节 Jog 回关节位置限制范围内，最后再次启用关节位置限制。

从 xCore V2.2 版本起，（非 RSC 机器人）当机器人运动至关节位置限制之外，可以向关节位置限制方向 Jog，从而回到关节位置限制范围内。

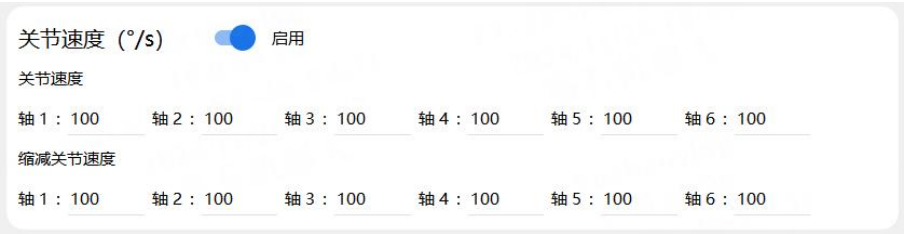
RSC 机器人，机器人运动到关节位置限制之外，机器人会进入“安全停止状态”；此时需要首先取消关节位置限制，然后“急停复位”，再将超限关节 Jog 回关节位置限制范围内，最后再次启用关节位置限制。

警告

取消关节位置限制功能只能用来在机器人关节超出关节位置限制时，将超限关节 Jog 回到正常范围内，关节位置限制取消时将无法运行程序。

11.3.3 关节速度

关节速度限制：可通过使能开关开启/关闭关节速度限制。开启后，会实时监控机器人各关节角速度。根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。任一关节角速度超过阈值，机器人将立即规划停止，下电；RSC 机器人并进入安全停止状态。



11.3.4 关节力矩

关节力矩限制：可通过使能开关开启/关闭关节扭矩限制。开启后，会实时监控机器人各关节力矩。根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。任一关节力矩超过阈值，机器人将立即规划停止，下电；RSC 机器人进入安全停止状态。

11.3.5 关节功率

关节功率限制：可通过使能开关开启/关闭关节功率限制。开启后，会实时监控机器人各关节功率。根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。任一关节功率超过阈值，机器人将立即规划停止，下电；RSC 机器人并进入安全停止状态。

11.4 机器人限制

①	速度限制：速度限制中包含 TCP 线速度、TCP 角速度、肘部线速度和肘部角速度等 4 项速度限制，每项速度限制有单独的使能开关，可通过对应使能开关开启/关闭对机器人的速度进行限制。开启后，会实时监控机器人的速度。

	根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。例如正常模式时采用"TCP 线速度"作为阈值，缩减模式时采用"缩减 TCP 线速度"作为阈值。当监控值超过阈值，机器人将立即规划停止，下电；RSC 机器人并进入安全停止状态。
②	缩减速度：用户若开启缩减速度，当机器人进入缩减模式时，机器人会在设置的 TCP 速度和关节速度运动。
③	整机功率限制：可通过使能开关开启/关闭机器人的整机功率限制。开启后，会实时监控机器人的整机功率，根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。 若整机功率超过阈值，机器人将立即规划停止，下电；RSC 机器人进入安全停止状态。
④	动量限制：可通过使能开关开启/关闭机器人的动量限制。开启后，会实时监控机器人动量，根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。若动量超过阈值，机器人将立即规划停止，下电；RSC 机器人并进入安全停止状态。
⑤	拖动速度限制：为 RSC 机器人专有功能，协作机器人可通过使能开关开启/关闭安全控制器对机器人拖动速度限制。开启后，如果协作机器人的拖动速度超过 250mm/s，将禁止拖动，下电，并进入安全停止状态。

11.5 虚拟墙

11.5.1 功能简介

虚拟墙针对 xMate 协作机器人笛卡尔空间(仅平移)拖动场景，限制法兰末端工作空间。当拖动接近虚拟墙时，用户会感受到虚拟墙的反作用力。

典型使用场景如下：医生将 xMate 协作机器人作为辅助工具，通过拖动进行手术操作。为了提高安全，避免误操作，可以通过设置虚拟墙，限定机器人法兰的操作空间范围。

①	"启用"开关控制虚拟墙功能是否启用；点击"确定"才生效
②	虚拟墙功能使用步骤介绍；
③	虚拟墙类型，包括：球形、立方体； 虚拟墙参数，包括中心点等； 注意："边界"内设置的参数值需大于一定范围，其中立方体大于等于 200mm，球体大于等于 100mm

注意：拖动力过大、速度过快的极限情况下，机器人有可能超出虚拟墙范围，系统会有相应提示。

11.6 碰撞检测

11.6.1 功能简介

碰撞检测功能是一种基于机器人动力学模型估计的被动检测功能。机器人运行过程中，与外界发生意外碰撞时，碰撞检测能够及时检测出碰撞外力，并执行预先设置的处置措施。

11.6.1.1 设置模式

设置模式支持整机设置和单轴设置；两种设置模式需至少勾选一种。
按照设置模式的不同，可以调节整机或单轴的灵敏度；百分比越高，灵敏度越高，机器人越容易检测到碰撞。灵敏度出厂默认设置为 100%，用户可根据需要进行调整。
按照机器人当前模式（正常或缩减模式），会采用不同的灵敏度阈值。

11.6.1.2 力限制

力限制（包括 TCP 力和肘部力）：可通过使能开关开启/关闭机器人的力限制功能（碰撞检测也开启）。开启后，会实时监控机器人 TCP 力和肘部力。根据当前模式(正常模式或缩减模式)，会采用不同的监控参数判断阈值。
当任意监控值超过阈值，机器人将触发碰撞检测触发行为，RSC 机器人会进入安全停止状态。

11.6.1.3 触发行为

触发行为支持"柔顺停止"一种。
柔顺停止：一种针对机器人和高刚度环境的碰撞检测停止方式。柔顺度越大，机器人响应越迅速，机器人关节承载越大；柔顺度一般使用默认 0 即可。安全停止后，机器人会自动下电，该状态下，机器人支持直接上电并沿当前路径继续运行程序。

11.6.1.4 传动力限制

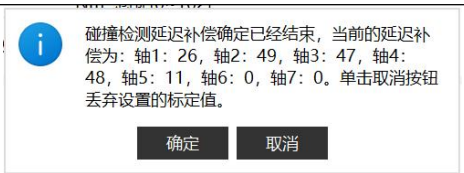
设置传动力限制参数，用于限制减速器的最大传动力矩，保护传动链重要部件和机械零点。
传动力限制参数，作用于碰撞保护*功能。当控制器监测到传动力超过该限制，机器人将触发传动力矩超限的报错。首次启动，机器人将使用默认的传动力限制参数。

11.6.1.5 参数辨识

碰撞检测参数辨识用于辨识和设置碰撞检测算法内部参数，以提升力监测精度，降低误报概率，达到最好的碰撞检测性能。

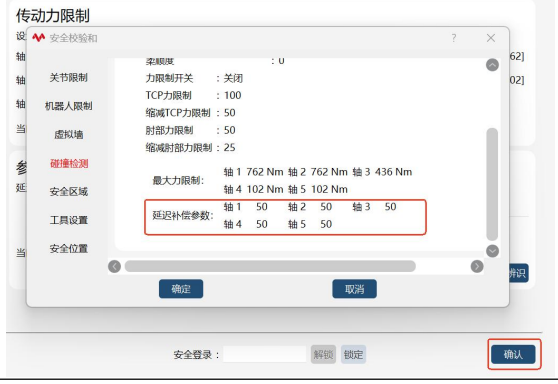
碰撞检测参数辨识支持"延迟补偿参数"辨识和设置。

使用碰撞检测参数辨识具体操作步骤如下:

步骤	图示	说明
1.关闭碰撞检测和碰撞保护		碰撞保护需要进入生产专用界面进行关闭。
1.点击开始辨识按钮;	开启辨识后需使机器人各个轴运动，从而获取准确的辨识值 开始辨识	碰撞检测参数辨识需要提前开启动力学前馈功能
2.切换自动模式并且上电;		
3.运行辨识的 RL 程序。		可使用专用的辨识程序或者是任意使用碰撞检测功能的 RL 程序。
4.等待辨识结果收敛后，停止辨识。	正在辨识中，当前结果已收敛 停止辨识	
5.保存辨识结果，设置延迟补偿参数		点击"确定"后延迟补偿参数立即生效。

手动设置延迟补偿参数步骤如下：

步骤	图示	说明
1.输入延迟补偿参数;		延迟补偿参数手动设置的数值不能超范围

2. 点击确认按钮设置参数；		

提示

1.在特殊情况下，如需要在进行生产作业时进行碰撞检测参数辨识，可以在辨识时开启碰撞检测和碰撞保护功能(不推荐使用）；
2. 为了获得各轴准确的辨识结果，辨识程序中各轴都需要运动，且各轴的运动范围因尽可能大，速度尽可能快；
3.碰撞检测参数辨识每次辨识的结果会有波动，其结果非固定的数值

11.6.1.6 最大出力监测

使用最大出力监测功能实时监测各轴电机在开关打开时间段内最大输出力矩。用户可根据各轴最大出力监测参数调整传动力限制参数。

提示

最大出力监测功能在控制器重启后会默认关闭

11.6.2 注意事项

1.在程序执行过程中机器人高速运动并与外部设备发生了硬碰撞，碰撞力过大直接导致伺服驱动器报警停机，清除碰撞后重启机器人复位伺服报警，才能重新运行机器人。
2.灵敏度模式选择不对，可能会引起机器人碰撞误报，请根据不同的应用场景，选择不同的灵敏度阈值。
3.碰撞检测灵敏度受机器人硬件影响，不同的机器人之间灵敏度阈值存在差异。目前三种灵敏度模式只是提供了一套标称值。用户若对碰撞检测灵敏度有更高需求，可通过单轴设置基于特定应用场景精调各轴灵敏度，或是通过 RL 指令来在线调整检测灵敏度。
4.碰撞检测及安全监控触发后会弹窗，必须点击“确认”以手动清除报警才可以继续运行。
5.碰撞检测功能，出厂时协作机器人默认为启用状态。
6.碰撞保护*功能介绍见生产界面使用手册。

> 警告
>
> 在使用碰撞检测之前必须要确保以下参数已经正确设置，否则控制器无法计算正确的输出力矩，可能会产生误报。
>
> 1. 机器人的型号
> 2. 机器人的安装方式
> 3. 负载信息（工具）
> 4. 机械和传感器零点
> 5. 本体参数信息

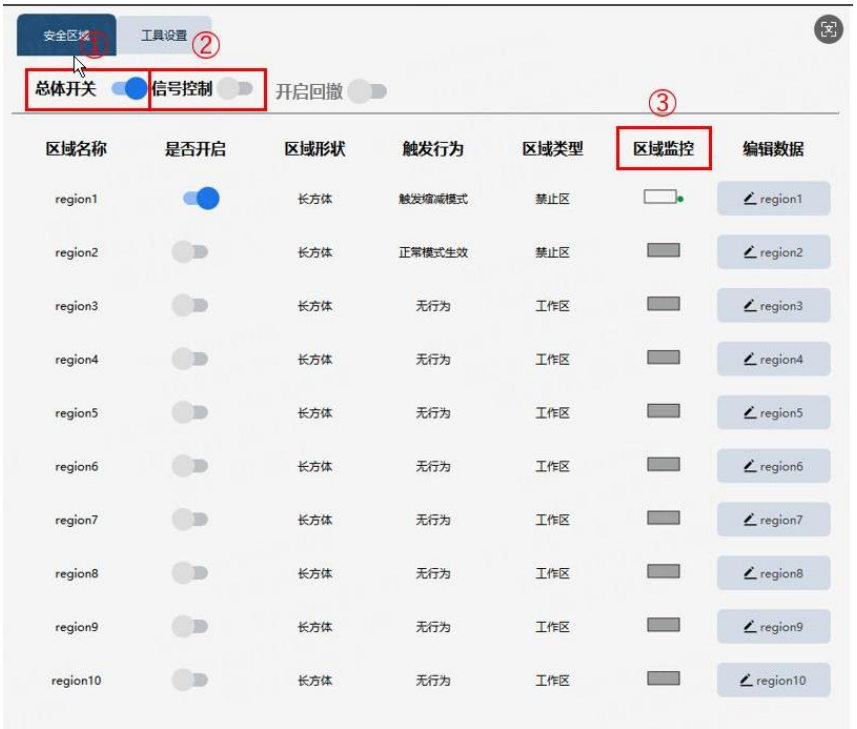
11.7 安全区域

11.7.1 功能简介

安全区域用于设置机器人的末端工具、肘部等在进出某个区域的行为。

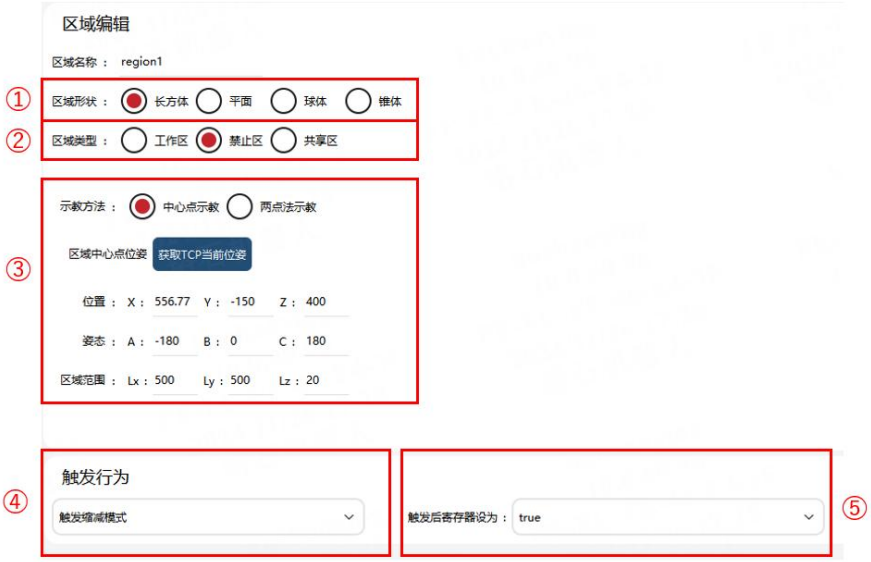
用户可以定义空间中若干安全区域（目前支持最多 10 个），当机器人在进、出安全区域时，选择性触发预设的安全行为，同时自动修改寄存器值(绑定安全区域的寄存器功能码)。

安全回撤功能在 11.7.3 章节介绍。



①	“总体开关”：开启或关闭安全区域功能，当此开关关闭时，所有安全区域无效。
②	“信号控制”：打开后，可以通过寄存器信号控制某个安全区域是否开启。 每个安全区域支持两种方式设置是否开启：一是通过 HMI 设置，二是通过寄存器设置(寄存器功能码”enable_safe_region01~enable_safe_region10”,类型为 bool 或 int16，只读类型)。 注意：当”信号控制”打开时，各安全区域”是否开启”，由绑定的寄存器决定，HMI 上”是否开启”栏下面的按钮不可操作；当”信号控制”关闭时，各安全区域开启情况由 HMI”是否开启”决定；
③	区域监控：显示机器人勾选的工具设置与区域间关系 图标　　说明

	区域未开启 在工作区内，且未触发 在禁止区外，且未触发 在工作区外，且触发 在禁止区内，且触发

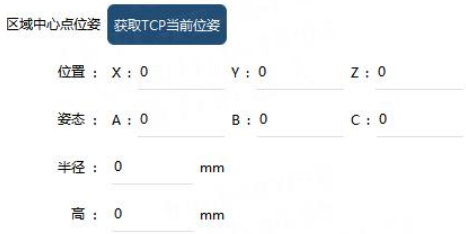


①	区域选择不用的形状支持设置的区域类型也不同，如下表。 区域形状 区域类型 长方体 工作区、禁止区、共享区 平面 工作区、禁止区 球体 工作区、禁止区、共享区 锥体 锥体内部为禁止区、外部为工作区 共享区：当区域选择为共享区时，需绑定 DIDO；当 DI 输入为 true 时，表示区域被占用，机器人即将进入该区域时会在区域外暂停等待；当 DI 输入为 false 时，表示区域解除占用，机器人继续运行进入共享区，并将 DO 输出置为 true。共享区的触发行为无需设置，机器人即将进入被占用共享区，行为为暂停等待；机器人已经进入被占用共享区时，行为为最大能力减速停止。
②	区域示教： 对于长方体区域，可以通过中心点示教和两点法示教两种示教方法。 中心点示教：点击获取 TCP 当前位姿(TCP 相对机器人基坐标系)来确定区域的中心点位姿；手动设置长方体的长、宽、高。 两点法示教：通过示教两个点（长方体对角线的两点）来确定长方体区域；“确定第一点”—>”确定第二点”—>”确认”

<table>
<tr>
<td></td>
<td>


对于平面区域，可通过示教或者手动输入位置（X，Y，Z），矢量暂时只支持手动输入设置。Z 轴所指方向为安全区域。


对于球体区域，可通过示教或手输中心点位姿，输入半径后确定区域。


对于锥体区域，可通过示教或手输中心点位姿，输入半径和高后确定区域。


</td>
</tr>
<tr>
<td>③</td>
<td>
触发行为包括：无行为、正常模式生效、触发缩减模式、缩减模式生效和正常或缩减生效；
无行为：机器人无具体动作；

正常模式生效：当机器人处于正常模式时，工具设置勾选的工具即将进入禁止区会触发规划停止；已经进入禁止区会触发最大能力减速停止；

触发缩减模式：工具设置勾选的工具进入禁止区时，机器人触发缩减模式；

缩减模式生效：当机器人处于缩减模式时，工具设置勾选的工具即将进入禁止区会触发规划停止；已经进入禁止区会触发最大能力减速停止；
</td>
</tr>
</table>

	正常或缩减生效：当机器人处于正常或缩减模式时，工具设置勾选的工具即将进入禁止区会触发规划停止；已经进入禁止区会触发最大能力减速停止；
④	区域绑定的寄存器在区域触发后的状态，包括：True/False； True：表示安全区域触发的安全动作时，寄存器输出信号为 true，反之亦反；（进入禁止区，输出为 true）。 False：表示安全区域触发的安全动作时，寄存器输出信号为 false，反之亦反；（进入禁止区，输出为 false）。

11.7.2 安全区域与寄存器关联

11.7.2.1 安全区域状态输出

步骤	图示	说明
1. 首先新建寄存器，选择类型为只写；		左图仅作为示例；
2. 功能选择“sta_safe_region01 ~ sta_safe_region10”，表示将对应安全区域的触发状态绑定到当前新建寄存器上。		

11.7.2.2 寄存器控制安全区域启用

步骤	图示	说明
1. 首先新建寄存器，选择类型为只读；		左图仅作为示例；
2. 功能码选择“enable_safe_region01 ~ enable_safe_region10”，表示将对应安全区域的控制开关		

绑定到当前新建寄存器上；		

11.7.3 安全区域回撤功能

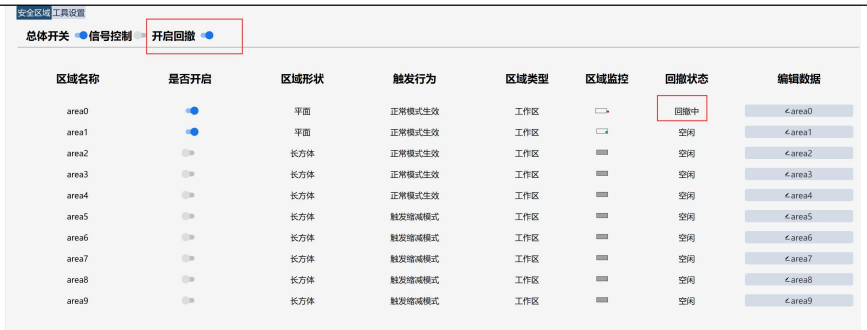
11.7.3.1 回撤功能介绍

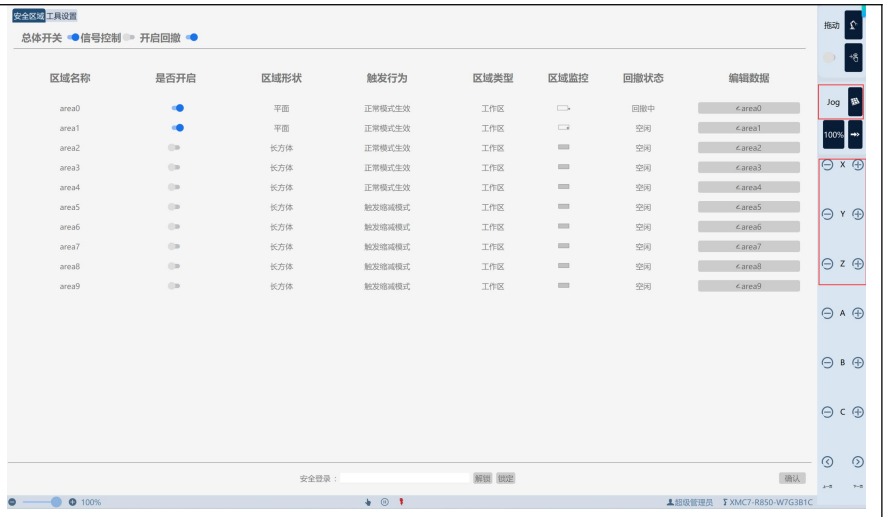
安全区域回撤功能是指，当机器人处于禁止区时，可通过打开安全区域回撤按钮，直接将机器人移出禁止区而无需再关闭所有安全区域。

①	该按钮为回撤开启的按钮，当区域监控显示当前位置已处于平面安全区域的禁止区且该区域为模式生效区域(正常模式生效、缩减模式生效、正常或缩减模式均生效)，则允许开启回撤按钮，其他情况回撤按钮不允许开启。   当所有安全平面区域都完成回撤后，回撤按钮会自动关闭
②	所有区域形状中，只有安全平面支持回撤功能 区域形状　　是否支持安全区域回撤 长方体　　不支持 平面　　支持 球体　　不支持 锥体　　不支持

③	 只有当区域监控处于红色时(处于禁止区)，才允许开启回撤按钮。
④	 回撤状态分为两种，一种是回撤中，表明当前区域正在进行回撤，可同时存在多个区域处于回撤中。另一种状态是空闲，表明当前区域正在等待或无需进行回撤。

11.7.3.2 回撤功能操作流程

①	 当机器人进入平面的禁止区后，会提示已经进入禁止区，此时无法再进行任何移动操作。如果此时想将机器人移动出禁止区，可以打开回撤按钮，开启回撤。
②	 打开回撤按钮，此时回撤状态变为回撤中，这是可以进行上电操作。如果是 RSC 机器人，开启回撤后，需要再点击 HMI 上的急停复位，复位后，也可以进行上电操作。

③	 上电后，可以朝着禁止区的反方向进行 jog，注意：回撤只允许 jog 操作，不允许其他运动操作，且 jog 只允许笛卡尔空间的 X Y Z 移动，不允许其他任何运动方式。当 jog 的方向并非朝禁止区外时，机器人不允许运动，且会上报错误，如下： 
④	当回撤完成后，回撤按钮会自动关闭，同时回撤状态会自动变更未空闲。回撤完成后，各个安全区域继续进行工作。 

11.8 工具设置

11.8.1 工具位置

工具位置限制支持同时对法兰、肘部、实时工具和两个固定工具的位置进行限制。每个位置可以设置一个指定的包络区域，当任何一个位置的包络超过安全区域设置的范围时，将触发安全区域设置的动作（正常模式生效、触发缩减模式等）。

工具包络：工具包络支持无包络、立方体和球体三种形状；

实时工具：在 RL 运行运动指令时，实时工具为指令中的工具；当没有运动指令运行时，实时工具为 HMI 右上角选择的工具；实时工具的包络可在编辑对应工具时设置（坐标系标定中的全局工具列表和工程中的工具列表）。

11.8.2 工具姿态

①	“使能”开关可开启/关闭姿态限制；
②	姿态限制功能同一时刻仅监控一个对象，可以在法兰、工具 1、工具 2 中选择一个；
③	角度：以开启姿态功能时，选中对象的姿态作为参考，按设定的角度形成的一个锥体，作为姿态允许范围；当选择对象姿态超出锥体范围时，将触发安全停止；

11.9 安全位置

11.9.1 功能简介

安全位置功能，指的是当机器人处于预设的安全位置时，绑定的寄存器会输出信号。利用该功能，用户可以获知机器人与安全位置相对位置关系。

xCore 控制系统支持设置最多 8 个以关节角度作为参考的安全位置，每个安全位置对应一个寄存器功能码（类型：bool 或 int16，读写：只写，sta_safe_jnt_pos1~sta_safe_jnt_pos8），当机器人当前关节角度与某个安全位置设定的关节角度在允许误差范围内，绑定该安全位置对应寄存器功能码的寄存器值会自动修改（当处于安全位置允许误差内，如寄存器类型为 bool，寄存器值为 true；如寄存器类型为 int16，寄存器值为 1）。

安全 Home 点为 RSC 机器人支持的功能，可以勾选一个安全位置作为安全 Home 点，当勾选后，若机器人各个关节均达到设定的范围时，可以输出一个安全 DO 信号。若一个都不勾选，则代表安全 Home 点功能未开启；

①	开关：某个安全位置是否启用；
②	安全 Home 点，可勾选一个安全位置作为安全 Home 点；
③	点击后，可以在右侧设置该安全位置的参数
④	安全位置对应的"关节坐标"； 可以手动更新；也可以点击"更新位置"，用机器人当前关节位置数据更新；
⑤	安全位置对应的"允许误差"，当机器人当前关节角度与"关节坐标"小于"允许误差"时，认为机器人处于该安全位置；

11.9.2 安全位置与寄存器关联

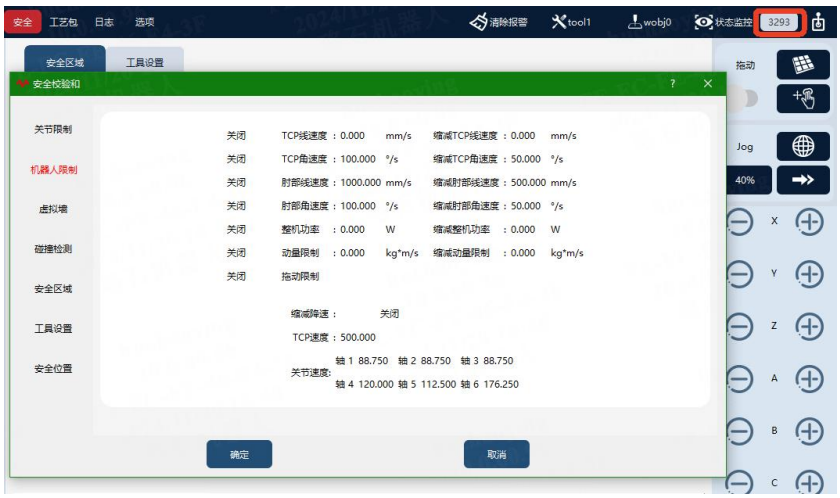
步骤	图示	说明
1. 首先新建寄存器，选择类型为只写；		左图仅作为示例；
2. 功能选择"sta_safe_jnt_pos1 ~ sta_safe_jnt_pos 8"，表示将对应安全位置的反馈状态绑定到当前新建寄存器上。		

11.10 安全校验和

修改安全设置，均需点击界面右下方的"确认"按钮，并经过安全校验确认后才可设置；"安全校验和"图标显示了四位"数字+字母"的组合，用于让用户了解安全相关的设置状态。当安全相关的设置发生变更时，会自动计算，生成新的四位"字母+数字"组合。

点击该图标，可查看当前机器人的安全设置信息，目前包括：关节限制、机器人限制、虚拟墙、碰撞检测、安全区域、工具设置和安全位置。

修改这几项涉及的参数后，点击"确认"按钮，会弹出安全校验和弹窗，点击"确定"按钮，安全参数设置成功，安全校验和也随之发生变化。



11.11 安全控制器

xCore 控制系统可以选配 RSC 安全控制器。安全控制器是一款符合功能安全认证的安全模块，内部进行诸多安全相关的计算和保护，xCore 控制系统自身的安全类功能并行处理，构成双保险。

为了保证安全控制器的数据安全和参数安全，实时数据采用 FSoe 通信机制以确保传输准确无误，非实时数据采用了安全的同步机制，每一次非实时数据的同步大约需要 5-10s 的时间。

采用了安全控制器的机器人，其安全板类型设置为：ROKAE_RSC，如下图所示。

11.11.1 选配安全控制器后的变更

除后续安全控制器配置界面显示的功能以外，选配了安全控制器的机器人，在使用上有几点变化。

11.11.1.1 机器人电机状态的变化

机器人状态额外增加一种"安全停止状态"，用于表示机器人由于安全控制器限制造成的安全状态。

安全停止状态	S	机器人处于安全停止状态，表示安全控制器检测到工作、通信异常或者某参数超出安全控制器设置安全阈值，此时无法给机器人上电

11.11.1.2 新增机器人复位操作

当机器人进入"急停状态"，"安全门状态"或"安全停止状态"中的任何一种时，如果想恢复至"下

电状态”，必须完成以下 2 个步骤：

第一步：消除触发进入以上三种状态的操作或条件，比如将急停按钮旋转归位，消除安全门触发信号，消除机器安全超限因素或关闭对应的安全限制功能；

第二步：单击下图所示界面上的“急停复位”按钮，机器人方可复位至“下电状态”。

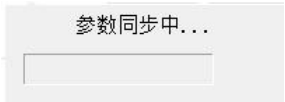
11.11.1.3 安全门逻辑的变化

未配置安全控制器的机器人，当机器人处于自动模式下，如果接收到安全门闭合的信号后，机器人将会立即下电。

配置安全控制器的机器人，当机器人处于自动模式下，如果接收到安全门闭合的信号时 RL 程序处于运行状态，将暂停 RL 程序的运行，机器人并不会下电。此情况下机器人无法运行 RL 程序或单步运行 RL 程序。如果希望恢复机器人状态，可以：执行断开安全门信号，在 HMI 点击“复位”信号。

11.11.1.4 零点标定和摩擦力参数设置时间差异

机器人零点信息和摩擦力参数的信息需要同步给安全控制器，以保证安全控制器的基本安全限制功能正常使用。因此在用户进行零点标定或者设置摩擦力参数的时候，控制器会主动将更新后的参数同步给安全控制器，大约需要等待 5~10 秒。此时界面如下图，用户无法操作和使用机器人。



11.11.2 安全 DO 配置

安全控制器包括四路安全 DO 信号。用户可根据需要，将几种安全状态信号映射到四路安全 DO 上。

①	急停输出：当机器人处于急停状态时，输出为 true，否则为 false；
②	机器人运动中：当机器人的 RL 程序正在运行时，输出为 true；当机器人的 RL 程序未运行时，输出为 false；
③	安全 Home 点：当机器人处在安全控制器设置的安全 Home 点设置的位置范围时，输出为 true，否则为 false；

④	缩减模式：当机器人处于缩减模式时，输出为 true，否则为 false；
⑤	拖动模式：机器人处于拖动状态；
⑥	机器人未停止：当机器人在 RL 程序运行的前提下（也就是"机器人运动中"为 true），当机器人关节确实在运动时，输出为 true，否则为 false；

12 工艺包

xCore 控制系统除完善的基本功能外，还包含丰富的扩展功能。以下工艺包，我们提供专门的文档，如有需要，可以联系官方获取。

12.1 传送带跟踪

主要特性和限制：

- 支持直线类型传送带跟踪；
- 支持光电、2D 视觉两种定位/触发方式；
- 支持工业六轴机器人；
- xMate 协作机器人只支持带控制柜版本；

12.2 导轨

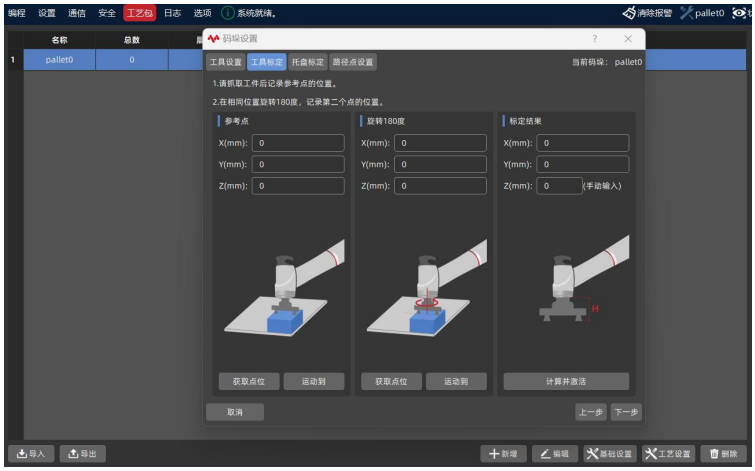
主要特性和限制：

- 只支持直线导轨；

12.3 通用码垛

主要特性和限制：

- 支持创建最多 100 个码垛工艺；
- 支持矩阵重叠式、纵横交错式、旋转交错式等预置垛型；支持用户自定义垛型；
- 1 个码垛工艺最多创建 100 种平面排样，1 个平面排样最多创建 200 个工件，1 个码垛可以创建 50 层。



12.4 料盘

主要特性和限制：

- 支持多达 100 个料盘工艺；
- 可定制机器人码放模式：平行模式、S 形模式等；
- 1 个料盘工艺可提供 16 种平面排样，1 个平面排样最多 999 个工件；

12.5 光伏排版

请联系珞石官方获取更多信息。

12.6 光伏插片

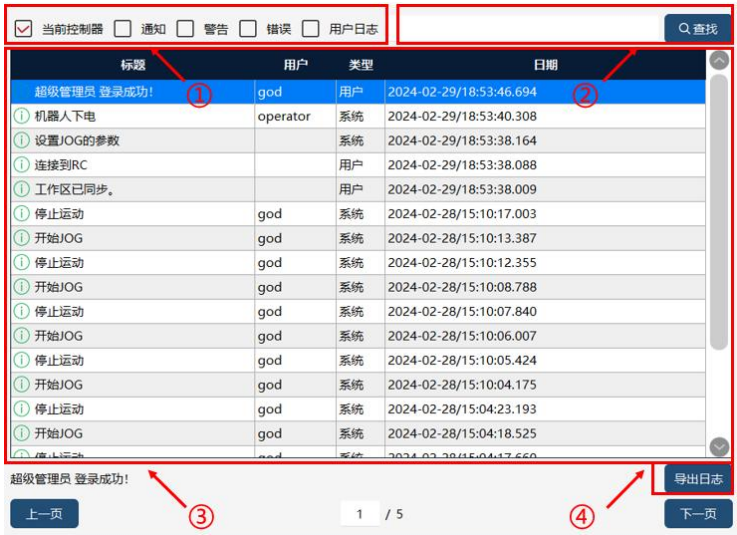
请联系珞石官方获取更多信息。

13 日志

日志模块包含 HMI 日志、控制器日志、运行日志、日志时间轴、内部日志、诊断设置、硬件状态、诊断数据监控八个界面。

13.1 HMI 日志

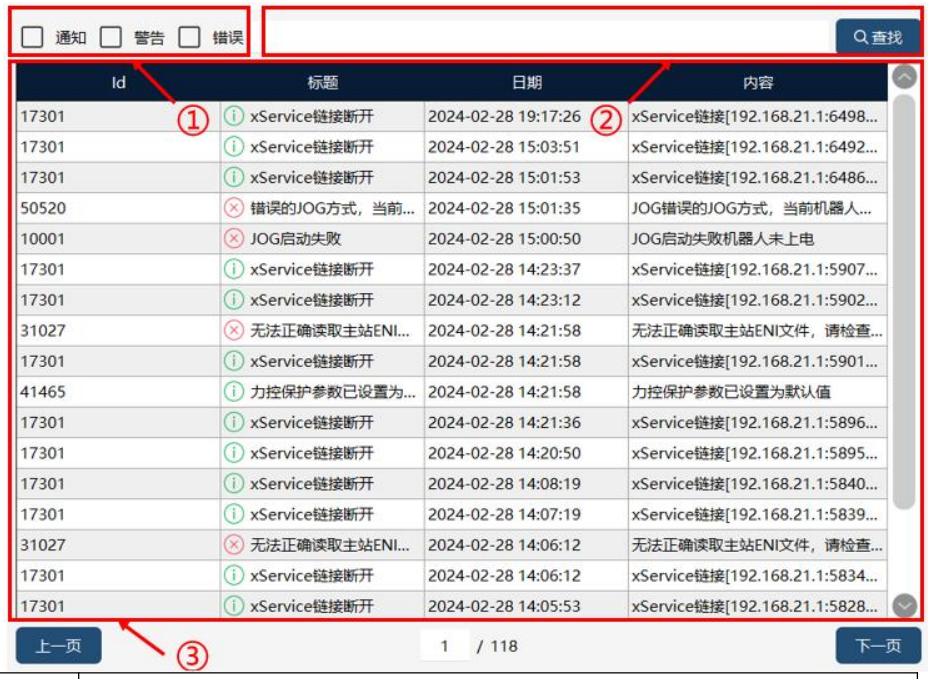
HMI 日志界面显示当前用户 HMI 操作日志信息，该界面有筛选区、查找区、显示页等部分，用户可点击导出日志按钮，导出所需要的 HMI 日志信息。



①	筛选条件区，可选择只查看当前控制器或者与 HMI 连接过的控制器日志；可选择日志级别进行筛选。
②	查找区，可根据关键字对日志进行搜索。
③	日志显示页，显示日志标题、用户、类型及发生时间等，使用翻页按钮切换上下页。
④	点击"导出日志"，可以将日志导出成 csv 格式文件。

13.2 控制器日志

控制器日志界面显示当前机器人控制器日志信息，该界面有筛选区、查找区、显示页三部分，用户可在此处查看控制器日志信息。

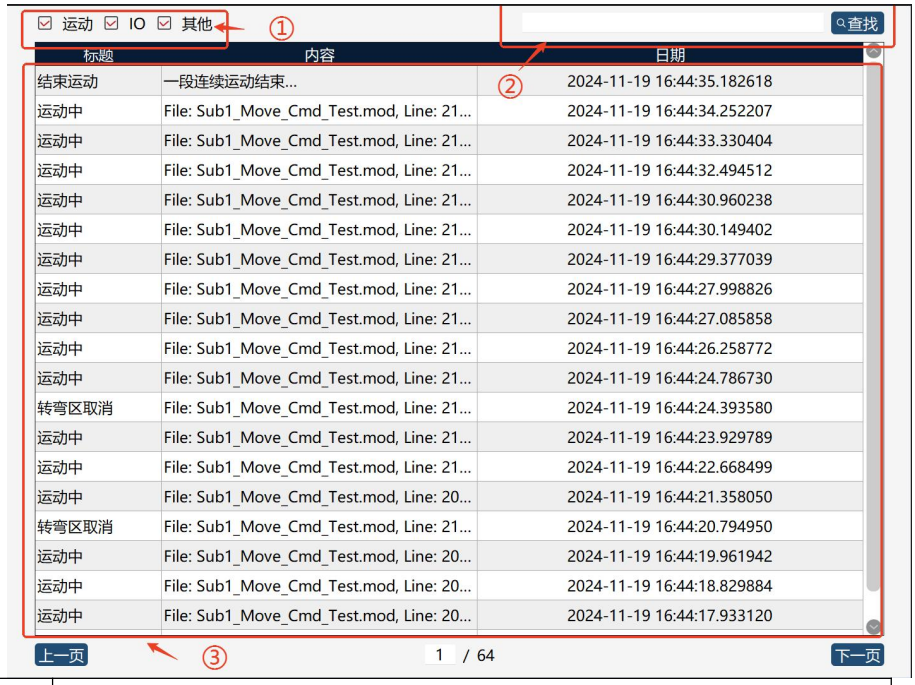


①	筛选条件区，可选择日志级别进行筛选。
②	查找区，可根据关键字对日志进行搜索。
③	日志显示页，显示日志编号、标题、发生时间以及内容等基本信息。使用翻页按钮切换上下页。

13.3 运行日志

相比于控制器日志，运行日志主要记录机器人曾经执行过的关键指令，以下信息会被运行日志记录：

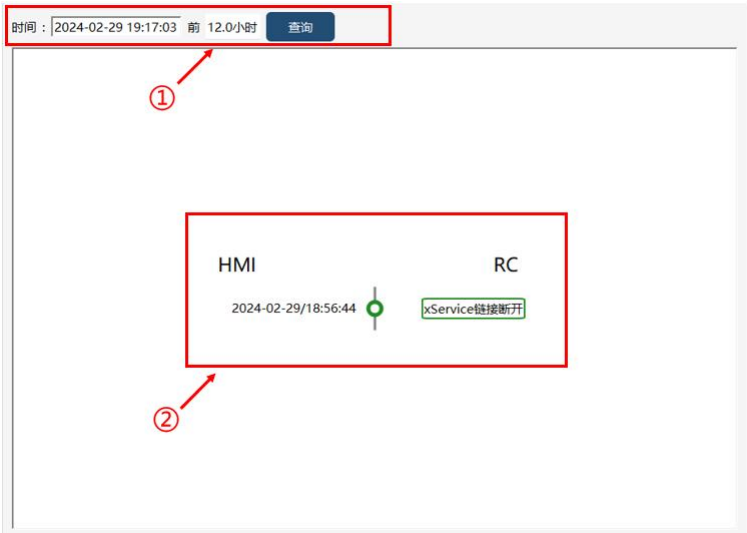
1. 机器人的启动停止以及运动指令的执行情况
2. IF-ELSE WAITUNTIL 这两条逻辑指令的执行情况
3. SetDO\PulseReg\PulseDO 指令的执行情况
4. JOG 操作、拖动、重回路径指令
5. 急停、暂停、启动等操作
6. 对外输出的大部分寄存器状态功能码，收到的外部控制机器人的控制功能码
7. 安全区域的变化情况
8. 碰撞检测、诊断数据保存等功能触发
9. 通过通信指令读取到的外部数据



①	筛选条件区，可选择日志级别进行筛选。
②	查找区，可根据关键字对日志进行搜索。
③	日志显示页，显示日志标题、发生时间以及内容等基本信息。使用翻页按钮切换上下页。

13.4 日志时间轴

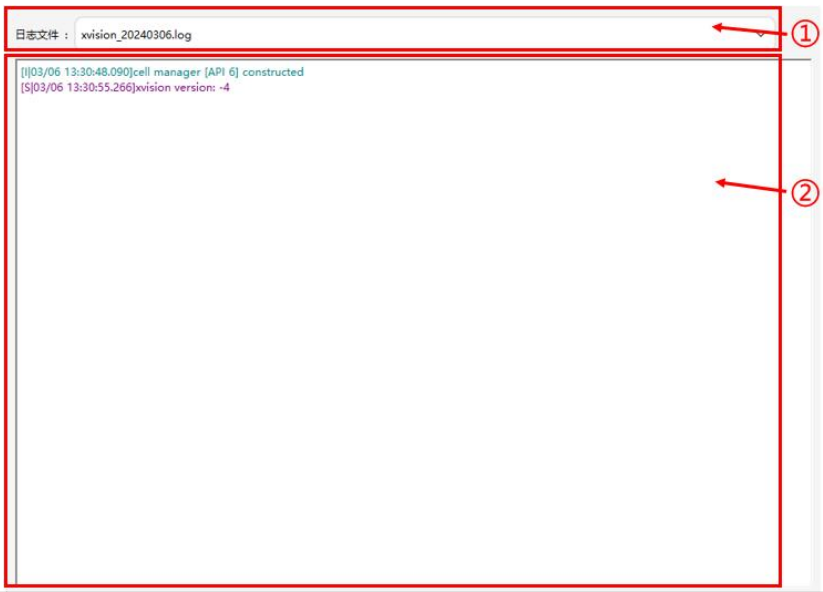
日志时间轴界面按照时间顺序同时显示 HMI 与控制器的日志信息，该界面有查找区与显示页，用户可通过查找区，查找不同时间段的 HMI 日志与控制器日志。



①	查找区，可设置时间区间对日志进行搜索。
②	日志显示页，左侧为 HMI 日志，右侧为 RC 日志。

13.5 内部日志

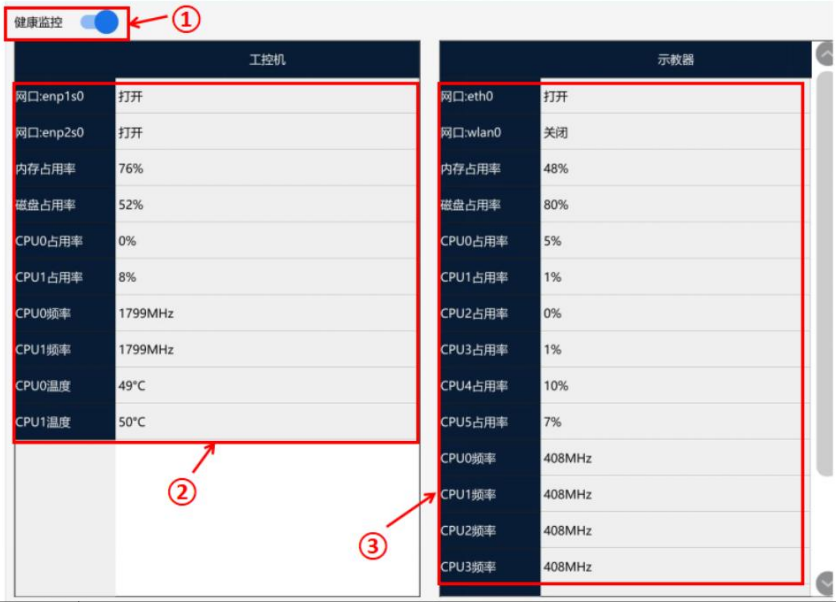
内部日志具有为技术问题、机器人故障等状况提供排查依据、定位问题原因等功能，该界面主要有查找区与显示页两部分。



①	内部日志选择区。
②	内部日志显示页。

13.6 硬件状态

用于监测工控机及示教器的硬件状态，包括内存占用率、磁盘使用率、网卡开启状态、cpu频率、温度、占用率。该界面有开启开关、工控机硬件状态显示页和示教器硬件状态显示页部分。



①	硬件状态监测开关

②	工控机硬件状态显示区
③	示教器硬件状态显示区

13.7 诊断设置

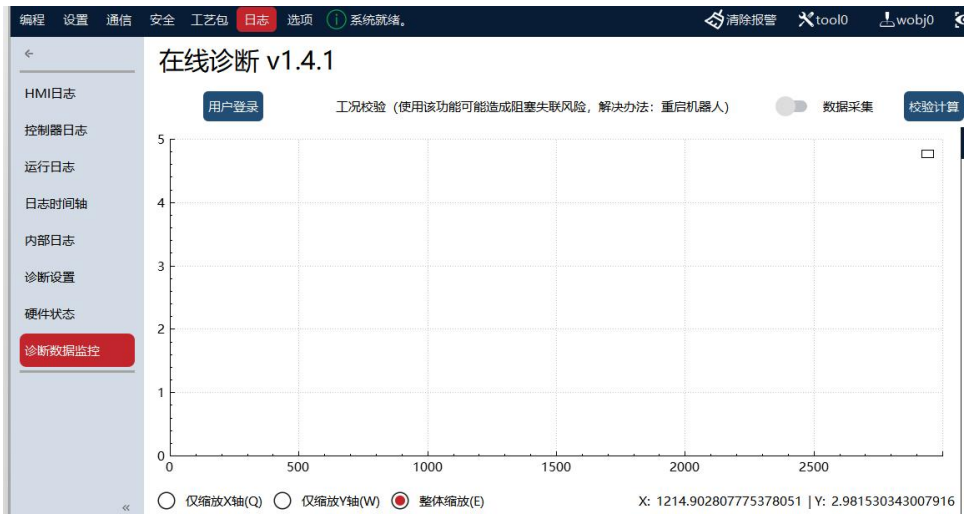
此界面功能用于辅助开发人员进行诊断伺服、ECAT 等设备问题，开启实时线程告警监测等功能。开启诊断功能后会加重控制器运行负荷，在实际生产环境中非必要不要打开。

伺服诊断	伺服诊断模块用来保存伺服出现错误的数据。点击保存按钮，5s 后可以导出诊断数据。
EC 诊断	EC 软件诊断功能可以辅助排查 ECAT 设备问题。
超时警告	开启后可以上送实时线程超时警告。
转弯区	开启后上报转弯区警告。
延迟运动	开启后运动延迟会延长。
前瞻转弯区	开启后上报前瞻转弯区警告。

网络测试用来检测机器人工控机与其他设备的通讯状态，网络是否能通。示教器连接控制器时，为控制器网络测试功能，用于检测控制器工控机与其他设备网络连接情况。示教器未连接控制器时，为示教器网络测试功能，用于检测示教器与工控机的网络连接情况。

13.8 工况校验

工况校验功能用于辅助开发人员和现场人员通过 RL 工程运行采集机器状况数据，用于分析机器问题。此功能在 PC 版本的 HMI 中支持，示教器版 HMI 不支持。在 xCore 3.1 版本中不再支持采集数据的导出和曲线绘制



该功能位于日志模块中的诊断数据监控页面，在运行 RL 时点击数据采集按钮，完成任务之后再点击数据采集按钮关闭，点击校验计算得出工况数据计算结果。

1	0.010497	正常	1.839600	正常	0.000000	正常
2	0.508205	正常	56.309925	正常	0.000000	正常
3	0.187171	正常	21.128640	正常	0.000000	正常
4	0.069840	正常	5.497049	正常	0.000000	正常
5	0.005934	正常	0.495526	正常	0.000000	正常
6	0.001251	正常	0.721062	正常	0.000000	正常

14 选项

选项页面，主要涉及机器人连接、软件升级、导入导出、特性演示相关内容。注意：U 盘容量太大会导致拷贝缓慢，会影响备份、导出的速度。

14.1 连接

①	搜索可用机器人，搜索在同一局域网中全部的机器人（直连情况除外），当连接机器人后，会显示控制器服务及升级服务均连接。 如果 RobotAssist 软件与机器人处于同一网络中仍搜不到机器人，或连接机器人后 3D 界面不显示机器人的实时位置，可以通过在"设置"-"HMI 设置"界面，绑定 IP 地址下拉框中选择局域网分配的 IP，尝试解决上述两个问题。
②	显示连接情况，用户可以主动断开连接。
③	启用自动重连功能。 当机器人控制器与 RobotAssist 软件间的网络断开后，RobotAssist 软件会尝试自动重连，超过设定的重连时间后将不再尝试连接。
④	重连方式有两种： (1)复选框勾选，重连总时长=单次时长*重连次数，用户可以指定单次时长和重连次数，在重连总时长范围内，会尝试自动重连。 (2)复选框未勾选，RobotAssist 软件会一直尝试重连控制器。

14.2 关于珞石

介绍了软件版本、组件信息，以及公司简介。

14.3 软件升级

14.3.1 控制器升级

通过控制器升级功能，可实现：（1）控制器软件版本升级，（2）数据恢复。

控制器版本升级：选择升级包，点击"上传"，界面会提示"上传成功"，根据弹窗提示重启即可。

数据恢复：选择要恢复的数据包，勾选要恢复的数据后点击上传，根据弹窗提示重启。

14.3.2 控制器备份

该功能可实现控制器中数据的备份。勾选希望备份的数据，点击"打开"，选择备份路径，点击"导出"。导出的文件为 rpa 格式的加密文件。

机器人配置，包括：控制器设置-系统配置、标定-力矩传感器零点电压值、本体参数-过载系数、运动参数-参数、安全控制等；

本机参数，包括：标定-编码器值、角度标定设定、基坐标系、动力学设置-(三阶)摩擦力系数；本体参数-DH 参数, 减速比等；

自定义配置，包括：动力学设置-前馈、约束开关、运动参数-Search 停止距离、最小转弯区半径、Conf、力控参数、xPanel 配置、快速调整、示教器模式、通信、安全。

控制器日志，包括：版本信息文件、用户设置文件、控制器日志、工程日志。只导出近 3 天的日志，更多日志在导出页面日志模块进行导出。

HMI 日志，包括：示教器的 syslog、示教器的镜像版本描述文件、版本信息文件。

系统日志，包括：工控机的系统日志 syslog、interfaces 文件、rc.local 文件。

14.3.3 HMI 升级

仅适用 xPad2，用于示教器 HMI 软件的升级。点击"打开"：选择 U 盘目录中的 HMI 软件升级压缩包；点击"升级"按钮，开始执行 HMI 升级流程。升级完成后，HMI 软件将自启动，升级完

成。

14.3.4 重启机器人

用于重启工控机系统，执行该操作需要建立升级服务连接。

14.3.5 抹除配置

用于抹除自定义配置、机器人配置、本体参数、工程数据等，关于以上数据的说明，参考"控制器备份"章节。执行抹除配置，仍会保留控制系统相关运行日志，务必谨慎操作。

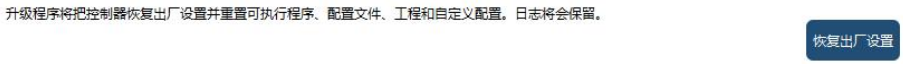
勾选要抹除的内容，点击"抹除配置"按钮，手动重启机器人生效。

注意：执行该操作，需要建立升级服务连接。

14.3.6 恢复出厂配置

用于将控制系统恢复到出厂默认状态。控制系统内的配置文件、工程和用户自定义配置将被重置，日志会保留，务必谨慎操作。

注意：执行该操作需要建立升级服务连接。



14.3.7 控制系统升级操作示例

当用户想进行控制系统升级时，可联系珞石获取新版控制系统安装包，参考操作步骤如下：

步骤	配图	说明
1.准备好安装包。		
2.打开 HMI，连接控制器，连接升级服务。		
3.升级前，建议先进行控制系统备份，以免重要数据丢失。可在"备份选项"中勾选备份内容，选择导出文件夹，导出备份。		此步骤非必须。

4.选择安装包，升级选项会根据安装包配置升级选项，点击上传。		请勿选中控制器升级选项中的自定义配置、机器人配置、控制器日志、工程数据演示案例、伺服。
5.上传成功后，HMI 会提示重启机器人，重启完成控制系统升级。		

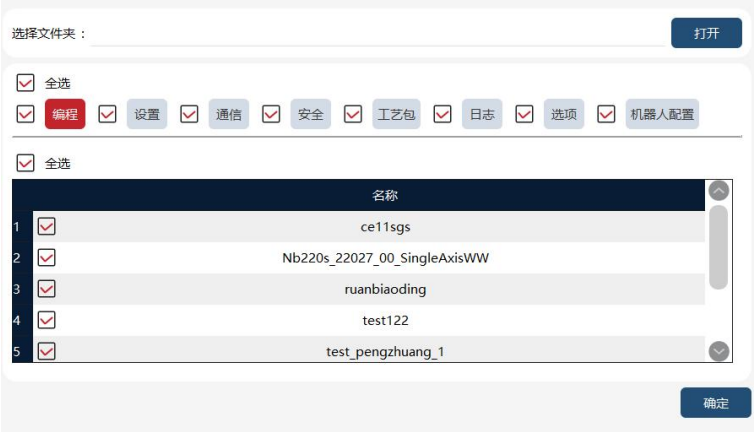
提示

1.当控制器版本与 HMI 版本不匹配时，HMI 将在顶部状态栏显示即时日志信息，内容为"版本不匹配。推荐使用 HMI 版本：[xxx]"。

2.当控制器版本与配置文件版本不匹配时，开机后 HMI 将会在中间弹出如下图所示的警告，此时将禁止任何机器人运动的操作，需要按照控制器日志信息升级正确版本的配置文件。

14.4 导出

可通过导出功能，来备份控制器相关设置。



如图所示，选择导出目标文件夹，选中要导出的内容，点击"确定"即可。

支持的导出内容如下：

功能模块	导出内容
编程	可导出具体的单个工程
设置	控制器设置—其他设置
	HMI 设置—基础设置、主题、工作区路径
	用户组-用户登录
	动力学设置—动力学前馈、动力学约束
	全局坐标系-全局工具、全局工件、全局坐标系
	力控参数—力控参数、力控模型、拖动优化
	运动参数—运动参数、AccSet、安全控制、Search 指令、最小转弯区、默认 Conf
	错误码过滤

	自定义按键
通信	系统 IO
	外部通信
	IO 设备
	总线设备
	末端工具
	RCI 设置
	串口设置
	编码器
安全	关节限制
	机器人限制
	碰撞检测
	安全区域
	安全位置
	虚拟墙
工艺包	传送带
日志	控制器日志
	HMI 日志
	工程日志
	系统日志
	备份工程
	控制器日志全选
选项	连接
机器人配置	机型文件

14.5 导入

可通过导入功能，导入控制器设置。

选择要导入的 zip 包，打开后会自动勾选 zip 包内包含的配置项，不包含的配置项置灰；用户可根据需求自行选择具体导入内容。注意：导入后，需重启控制系统才能生效。

14.6 文件管理器

文件管理器，主要用于快捷打开 RobotAssist 软件涉及的若干文件夹。

注意：该功能仅 PC 版 HMI 软件具备。

缓存文件夹：存储 RobotAssist 软件的缓存。

日志文件夹：存储 RobotAssist 软件日志，此处的日志与诊断界面的内部日志内容一致，可以点击此处进入文件夹进行日志拷贝。

工作区文件夹：存储机器人工程文件。

14.7 演示

14.7.1 七轴冗余运动

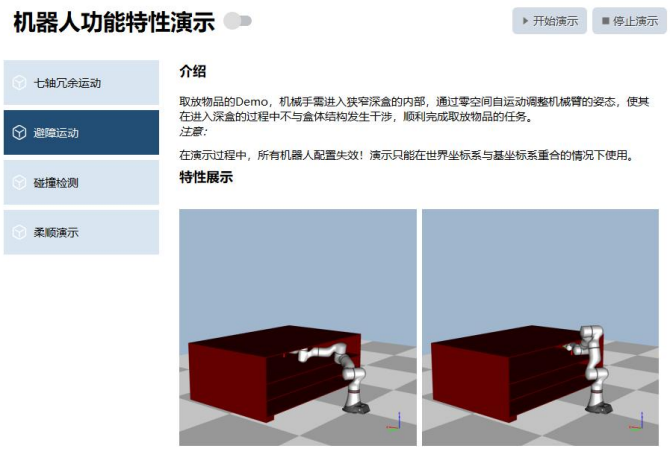
七轴冗余运动演示，支持 xMate ER PRO，演示运动包括：圆弧、直线、转弯区、零空间运动等。

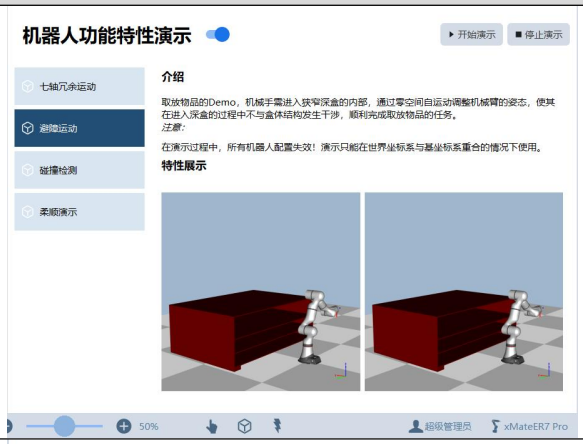
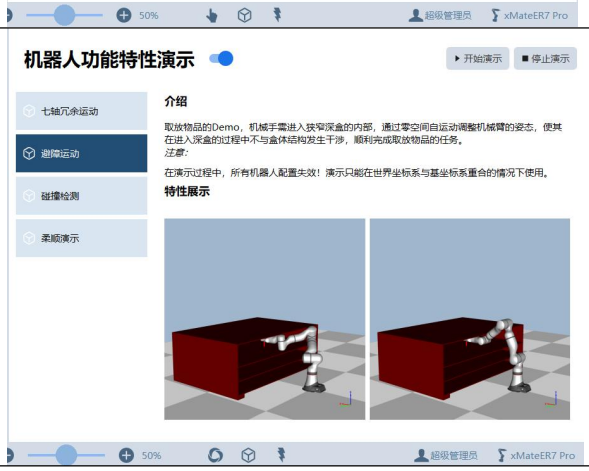
步骤	图示	说明
1.选择选项-演示-七轴冗余运动，打开演示开关，进入演示模式。		

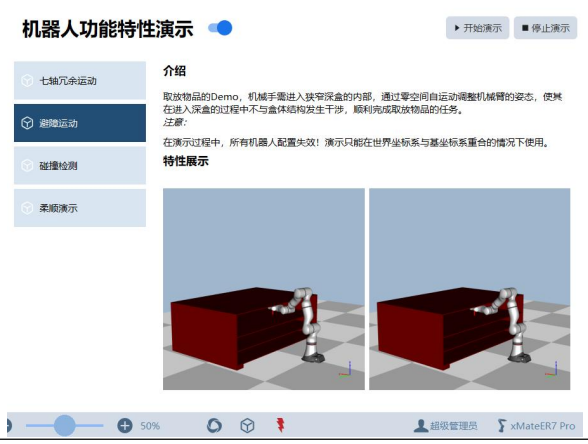
2. 点击底部状态栏操作模式切换按钮，将机器人切换至自动模式。		
3.点击底部状态栏上电按钮，进行上电。		
4.点击右上角的开始演示。		
5.Demo 演示完毕后，点击右上角的停止演示。		

14.7.2 避障运动

演示机械手进入窄深盒的内部，通过零空间自运动调整机械臂的姿态，使其在进入深盒的过程中不会与盒体结构发生干涉，顺利完成取放物品的任务。



步骤	图示	说明
1.选择选项-演示-避障运动，打开演示开关，进入演示模式。		
2．点击底部状态栏操作模式切换按钮，将机器人切换至自动模式。		

3.点击底部状态栏上电按钮，进行上电。		
4.点击右上角的开始演示。		
5.Demo 演示完毕后，点击右上角的停止演示。		

14.7.3 碰撞检测

演示碰撞检测功能。

碰撞检测灵敏度设置，支持单轴设置，也支持高、中、低三档设置。

当机器人检测到碰撞后，会柔顺停止。3.0 版本碰撞检测不再支持按压继续功能。

使用演示碰撞检测功能时，如碰撞保护*功能已开启则碰撞保护*会被暂时关闭。

步骤	图示	说明
1.选择选项-演示-碰撞检测，打开演示开关，进入演示模式。		
2．点击底部状态栏操作模式切换按钮，将机器人切换至自动模式。		

3.点击底部状态栏上电按钮，进行上电。		
4.点击右上角的开始演示。		
5.Demo 演示完毕后，点击右上角的停止演示。		演示过程中，需要调整碰撞检测灵敏度阈值时，先点击停止演示按钮，调整完阈值之后，再开始演示。

14.7.4 柔顺演示

柔顺演示，展示了 xMate 不同刚度、不同空间下的柔顺控制效果。

步骤	图示	说明
1.选择选项-演示-柔顺演示，打开演示开关，进入演示模式。		
2. 点击底部状态栏操作模式切换按钮，将机器人切换至自动模式。		
3.点击底部状态栏上电按钮，进行上电。		

4.点击右上角的开始演示。		
5.Demo 演示完毕后，点击右上角的停止演示。		演示过程中，需要调整柔顺演示刚度阈值时，先点击停止演示按钮，调整完阈值之后，再开始演示。

警告

演示功能期间，机器人所有配置失效，注意：

1.机器人基坐标系默认与世界坐标系重合。

2.机器人默认末端无负载，机器人末端安装负载时，会影响碰撞检测、柔顺演示等演示。

15 RL 指令

15.1 变量类型

15.1.1 Int

说明	整型 int 变量的取值范围是-2147483647~2147483647，建议赋值在规定范围内，如果超过范围则会随机赋值，使用时请不要超过最大取值范围。
示例	在变量列表中定义：  表示定义了一个整型的变量类型的数据 counter，且初始值等于 4。

15.1.2 Double

说明	浮点数，使用 8 个字节存储，使用时请不要超出取值范围。
示例	在变量列表中定义：  表示定义了一个浮点型的变量 time，且初始值等于 1.5。

15.1.3 Bool

说明	bool 型变量主要用于状态或逻辑判断，取值为 true 或者 false。 当被赋值 int 或 double 值时，非 0 取值为 true，0 取值为 false。
示例	在变量列表中定义：  表示定义了一个 bool 型的变量 ifClose，且初始值为 true

15.1.4 String

说明	字符串类型变量由多个字母或者数字组成。 注意：在 RL 文本中定义时必须放在双引号""内。
示例	在变量列表中定义：  表示定义了一个字符串变量 name，并初始化为"rokae"。 字符串类型变量支持"+"操作，可实现字符串拼接。 例如： name = "Rok" + "ae" 表示变量 name 被赋值为"Rokae"。

15.1.5 Array

说明	数组是由相同类型的变量组成的集合，可以是一维也可以是多维。数组中的元素使用下标来访问，每一维的下标都从 1 开始。数组总长度不允许超过 1000。
示例	在变量列表中定义：

	 表示定义了一个二维数组，包含 16 个整型变量，将数组的第一行第 6 个元素赋值为 8。

15.1.6 byte

说明	byte 表示 RL 语言中的无符号字节，相当于 C++中的 unsigned char。 取值范围 0~255，不允许为负值。一般用于 SocketSendByte 指令。 当 byte 变量值超过 255 时，会自动进行截断，只保留低 8 位的值，比如 var byte data2=288，截断后 data2 的值为 32。
示例	在变量列表中定义：  定义了一个 byte 类型变量 data，它的值为 177。

15.1.7 clock

说明	clock 用于计时，clock 指令就像用于计时的秒表。 clock 类型存储的时间精度是 0.001s，最大时间间隔 45 天（即 45×24×3600 秒）。
示例	在变量列表中定义：

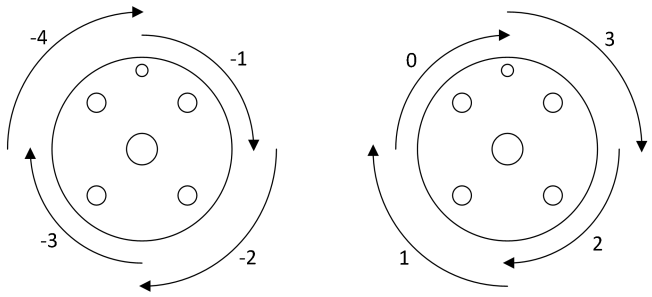
	 下面的例子显示了如何使用变量类型 clock: 例 1: ClkStart (clock1); ClkStop (clock1); interval=ClkRead(clock1); ClkReset (clock1); Interval(预先声明的 double 变量)读到的是 ClkStart 和 ClkStop 之间的时间间隔，单位是 s。

15.1.8 隐式类型转换

说明	目前，在变量列表中设置数据时，限制了数据类型，即不符合变量类型的值不能成功输入，从而避免类型隐式转化。
示例	在定义整型变量 counter 时，只能输入整数，而不能输入小数。

15.1.9 Confdata

说明	confdata（Robot Configuration Data）用于定义空间目标点对应的形态配置数据。 由于机器人多采用旋转关节，任何一个关节在 1°和 361°时表现出来的状态是一样的，因此在选定机器人的形态后，还需要采取其他措施来处理关节的多圈性问题。 对于工业机器人和 xMate ER 协作机器人，我们以象限数的方法来标记关节角度的大致范围，例如当关节角度位于 0~90 度时其象限数为 0，关节角度为 90~180 度时记为 1，每隔 90 度增减 1。角度为负数时，相应的象限数也为负数，如下图所示：左图为关节角度为负时的情形，右图为关节角度为正时的情形。对于机器人关节而言，逆时针转动角度增加，顺时针转动角度减小。在下图中，某关节顺时针转动，关节角度减小，则关节角度对应的 confdata 变化分别为-1->-2->-3->-4 和 3->2->1->0。 对于 xMate CR 和 SR 协作机器人，我们在 confdata 中直接记录各关节角度向下取整的数值。 此外，同一个笛卡尔空间目标点对应多组不同的运动学逆解，因此需要使用 confdata 来明确指定选择哪一个形态。



定义

需要 7 个参数来构成完整的 confdata，包括：

cf1，数据类型：int，1 轴角度对应的象限或向下取整的关节角度。

cf2，数据类型：int，2 轴角度对应的象限或向下取整的关节角度。

cf3，数据类型：int，3 轴角度对应的象限或向下取整的关节角度。

cf4，数据类型：int，4 轴角度对应的象限或向下取整的关节角度。

cf5，数据类型：int，5 轴角度对应的象限或向下取整的关节角度。

cf6，数据类型：int，6 轴角度对应的象限或向下取整的关节角度。

cf7，数据类型：int，7 轴角度对应的象限或向下取整的关节角度。

cfx，数据类型：int，机器人使用的形态配置编号，取整范围 0~8。

示例

同一个末端笛卡尔位姿有多组不同的逆解，cfx 使用 0~8 这 8 个整数代表各组逆解，详细解释如下。

对于 6 轴工业机器人、xMateER 协作机器人和 7 轴机器人：

cfx	腕心位于 1 轴.....	腕心位于大臂.....	手腕角度为.....
0	前面	前面	正
1	前面	前面	负
2	前面	后面	正
3	前面	后面	负
4	后面	前面	正
5	后面	前面	负
6	后面	后面	正
7	后面	后面	负

对于 xMateCR 协作机器人（Cfx=0 表示离 cf1~6 所代表的关节角度较近的解）：

cfx	腕心位于 1 轴.....	腕心位于大臂.....	5 轴角度为.....
0			
1	前面	前面	负
2	前面	前面	正
3	前面	后面	负
4	前面	后面	正
5	后面	前面	负
6	后面	前面	正
7	后面	后面	负
8	后面	后面	正

对于 xMateSR 协作机器人，cfx 恒为 0，表示离 cf1~6 所代表的关节角度较近的解。

对于 3 轴工业机器人和 4 轴工业机器人：

cfx	腕心位于大臂.....
0	前面
1	后面

15.1.10 Jointtarget

说明	存储机器人关节角度和外部轴位置。
定义	robax，机器人关节角度（Robot Axis），数据类型：double，包含 7 个 double 型的成员，分别存储机器人 7 个关节的角度 Degree。 extax，外部轴位置（External Axis），数据类型：double，包含 6 个 double 型的成员，最多可存储 6 个外部轴的位置信息。如果外部轴为旋转轴，则单位是角度 Degree；如果外部轴为直线轴，则单位是毫米 mm。
示例	在变量列表中定义：  上述语句定义了一个名为"jointtarget0"的关节空间的点，除六轴为 90 度外，机器人其他轴角度均为 0 度，第一个外部轴位置为 10 度或 10 毫米，取决于外部轴类型。
结构体赋值	jointtarget j1 = J:{-268.649031, 321.536626, 259.344893, 55.143011, 66.111070, 169.543340, 29.916387}{EJ 0,0,0,0,0,0}; 说明： 1、 J:{...} 表示机器人 7 个关节的角度，类型为 double。 2、 {EJ ...} 表示 6 个外部轴的位置信息，类型为 double，本例中全部设置为 0。

15.1.11 load

说明	load 变量类型用于存储机器人负载的动力学参数。 机器人的负载主要有两种： ● 安装在机器人末端的工具或工件本身； ● 工具所抓起/吸起来的物体。 load 型变量不支持单独创建，只能作为 tool 型变量的成员在标定工具界面手动修改，或者使用负载辨识功能由控制系统自动修改。 正确定义负载的动力学参数可以使得机器人获得最佳性能。错误的定义可能会导致如下后果： ● 机器人无法最大化利用伺服系统的能力，导致性能下降。 ● 路径精度降低，定位误差变大。 ● 机械部件过载导致寿命降低或损坏。
定义	在 xCore 系统中负载被当做刚体来处理，用于描述负载的参数如下：

mass，质量（Mass），数据类型：double，用于描述负载的质量，单位是千克 kg。

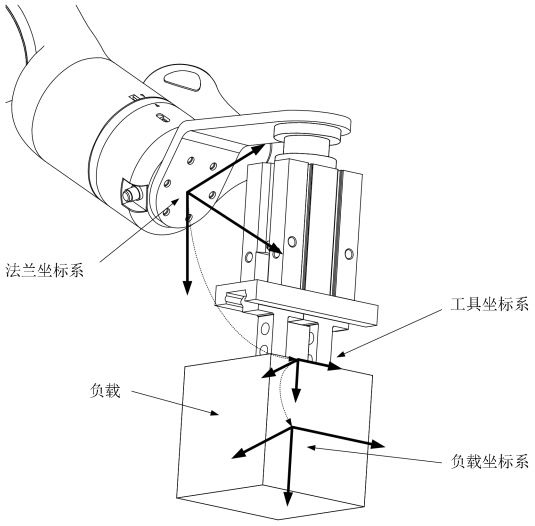
cogx，质心（Center of Gravity）在 x 方向的偏移量，数据类型：double，如果工具安装在机器人上，则 cogx 记录负载质心在工具坐标系下 X 方向的偏移量；如果使用外部工具功能，则 cogx 记录被抓手夹持的负载质心在工件坐标系下 X 方向的偏移量。

Cogy，质心（Center of Gravity）在 Y 方向的偏移量，数据类型：double，如果工具安装在机器人上，则 cogy 记录负载质心在工具坐标系下 Y 方向的偏移量；如果使用外部工具功能，则 cogy 记录被抓手夹持的负载质心在工件坐标系下 Y 方向的偏移量。

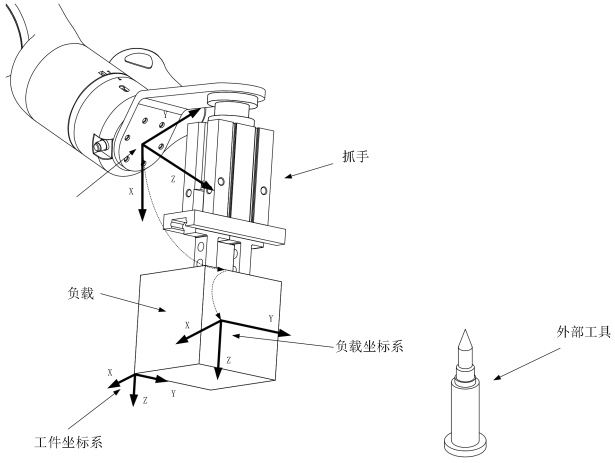
Cogz，质心（Center of Gravity）在 Z 方向的偏移量，数据类型：double，如果工具安装在机器人上，则 cogz 记录负载质心在工具坐标系下 Z 方向的偏移量；如果使用外部工具功能，则 cogz 记录被抓手夹持的负载质心在工件坐标系下 Z 方向的偏移量。

q1~q4，四元数，记录负载惯量主轴的方向，数据类型：double；

当工具安装在机器人上时，惯量主轴的方向是在工具坐标系下描述的，见下图：



当使用外部工具时，惯量主轴的方向是在工件坐标系下描述的，见下图：

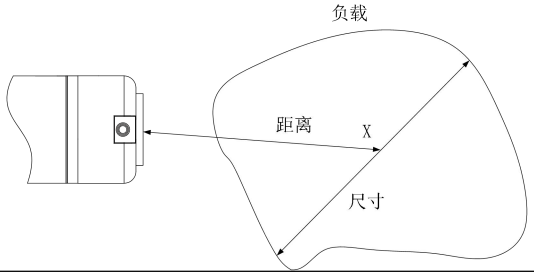


Ix，惯量 x，数据类型：double，负载沿 x 主轴的惯量，单位 kgm2。

Iy，惯量 y，数据类型：double，负载沿 y 主轴的惯量，单位 kgm2。

Iz，惯量 z，数据类型：double，负载沿 z 主轴的惯量，单位 kgm2。

注意：正确定义负载惯量有助于提升机器人运动精度，尤其是搬运较大物体的情况下。如果 ix、iy、iz 都被设置为零，则负载将会被当做一个质点来处理。通常情况下，如果负载质心到法兰中心点的距离小于负载本身的最大尺寸时，就应该对负载惯量进行定义，如下图：

15.1.12 orient

说明	存储坐标系或空间刚体的姿态信息。 orient 类型的变量不支持单独创建或者修改，仅作为某些变量的成员变量。
定义	RL 语言系统使用四元数来表示姿态，因此共有 4 个分量，表示形式为： q1，数据类型：double，四元数的第一个分量。 q2，数据类型：double，四元数的第二个分量。 q3，数据类型：double，四元数的第三个分量。 q4，数据类型：double，四元数的第四个分量。
备注	通常情况下我们使用旋转矩阵来描述刚体的姿态，四元数是另一种更为简洁的姿态描述方式。 四元数的四个分量满足如下关系：$q_1^2+q_2^2+q_3^2+q_4^2=1$ 旋转矩阵和四元数之间可以相互转换，假设有一个旋转矩阵 R： $\mathrm{R}=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ 则： $q_1=\frac{\sqrt{r_{11}+r_{22}+r_{33}+1}}{2}$ \ $q_2=\frac{\sqrt{r_{11}-r_{22}-r_{33}+1}}{2}$ $sign\ q_2=sign(r_{32}-r_{23})$ $q_3=\frac{\sqrt{r_{22}-r_{11}-r_{33}+1}}{2}$ $sign\ q_3=sign(r_{13}-r_{31})$ $q_4=\frac{\sqrt{r_{33}-r_{11}-r_{22}+1}}{2}$ $sign\ q_4=sign(r_{21}-r_{12})$

15.1.13 pos

说明	用来存储三维空间的位置信息。 pos 类型的变量不支持单独创建或者修改，仅作为某些变量的成员变量。
定义	RL 语言系统使用笛卡尔坐标系来描述三维空间，因此 pos 变量有 x、y、z 三个分量。 X，数据类型：double，位置的 x 坐标。 Y，数据类型：double，位置的 y 坐标。 Z，数据类型：double，位置的 z 坐标。

15.1.14 pose

说明	存储笛卡尔空间的位置和姿态。
定义	X，数据类型：double，位置的 X 坐标。

	Y，数据类型：double，位置的 Y 坐标。 Z，数据类型：double，位置的 Z 坐标。 Q1，数据类型：double，四元数的第一个分量。 Q2，数据类型：double，四元数的第二个分量。 Q3，数据类型：double，四元数的第三个分量。 Q4，数据类型：double，四元数的第四个分量。
结构体赋值	pose pose_obj = PE:{{X,Y,Z},{Q1,Q2,Q3,Q4}}; 参数含义见上方定义。 具体示例：pose pose_obj = PE:{{100,100,100},{1,0,0,0}};

15.1.15 robtarget

说明	存储三维空间的笛卡尔位置和姿态，用于 MoveJ、MoveL、MoveC 以及 MoveT 指令。 由于机器人运动学逆解存在多解性，对于同一个目标位姿，机器人往往可以采用多种不同的形态到达，为了明确的指定采用哪一种配置形态，robtarget 变量中还包含了机器人配置（Robot Configuration）数据。 robtarget 类型的变量在通过辅助编程插入运动指令时自动创建，手动更改变量内部的值可能会导致 Pose 和 ConfData 不对应，机器人无法正常执行运动指令。 注意：机器人程序中使用笛卡尔位置和姿态均是在工件坐标系下定义的。如果最终使用的工件与最初编程时使用的工件不一样，会导致机器人的运动偏离期望路径。因此对于以下两种情况要确认工件的变更不会造成危险： ● 使用"修改指令"功能更改指令的 wobj 参数时； ● 实际使用的工件与程序指令中使用的工件不一样时。 不当使用会导致人员受伤或设备损坏！
定义	Trans，空间位置，数据类型：pos，保存在参考坐标系下的位置偏移量。 Rot，姿态，数据类型：orient，保存在参考坐标系下的姿态。 Conf，机器人配置数据（Robot Configuration），数据类型：confdata，保存机器人的位形配置数据，详见 confdata 介绍。 Extax，外部轴信息（External Axes），数据类型：double，extax 包含 6 个 double 型的成员，最多可存储 6 个外部轴的位置信息。如果外部轴为旋转轴，则单位是角度 Degree；如果外部轴为直线轴，则单位是毫米 mm。

<table>
<tr><td>示例</td><td> 

定义了一个名为 p1 的笛卡尔位姿，其位置与姿态（四元数表示）如上所示，臂角角度为 10°，1,3,5,7 轴的角度均处于 0~90°之间，机器人对应第一组位型构型（详见 confdata），所有外部轴处于零位。</td></tr>
<tr><td>结构体赋值</td><td>robtarget rob1 = p:{{849.572593, -347.654631, 35.341636},{-0.361604, 0.078279, 0.640346, -0.673106}, -1.058584}{cfg 1,2,3,4,5,6,7,8}{EJ 1,2,3,4,5,6};
说明：p:{{空间位置 Trans},{姿态 Rot}，臂角}{cfg 机器人配置数据 Conf}{EJ 外部轴信息 Extax};</td></tr>
</table>

15.1.16 signalxx

<table>
<tr><td>说明</td><td>signalxx 类型的变量用于描述输入与输出信号。
所有 signalxx 类型的变量都需要在"IO 信号列表"中进行定义，然后在程序中使用，不支持在程序中直接声明。
signalxx 目前仅支持数字输入输出，包括如下变量类型：
<table>
<tr><th>变量类型</th><th>用于描述...</th><th>说明</th></tr>
<tr><td>signaldi</td><td>数字输入信号</td><td>值为 True 或者 False，仅表示状态</td></tr>
<tr><td>signaldo</td><td>数字输出信号</td><td>值为 True 或者 False，对输出赋值</td></tr>
<tr><td>signalgi</td><td>数字组输入信号</td><td>将一段连续的物理输入端口定义为一个二进制数，在 RL 中可转换为十进制使用，最多支持 16 个 DI 构成组输入，因此 signalgi 的取值范围是 0 ~ （2^n -1），n 是组输入包含的 DI 点个数</td></tr>
<tr><td>signalgo</td><td>数字组输出信号</td><td>将一段连续的物理输出端口定义为一个二进制数，在 RL 中可转换为十进制使用，最多支持 16 个 DO 构成组输出，因此 signalgo 的取值范围是 0 ~ （2^n -1），n 是组输出包含的 DO 点个数</td></tr>
</table>
signaldo 和 signalgo 类型仅包含信号的引用，可使用单独的指令（例如 SetDO、SetGO 等）来进行赋值。</td></tr>
</table>

	signaldi 和 signalgi 可用于在程序中直接获取所对应输入信号的值。 注意： ● 不支持在程序中定义/声明 signalxx 类型的变量，如果出现这种用法，程序将会报错。在使用 signalxx 类型的变量之前，请首先在 IO 信号列表中进行配置。 ● signalxx 型变量的作用域为 System，与其他作用域类型的关系为 System > GLOBAL > LOCAL。 ● 如果 IO 配置界面的 Signal 与 RL 程序中声明的变量重名，那么较低作用域级别的变量将被选中。

示例	例 1
	//使用数字输入的状态作为判断条件
	IF（di1 == true）
	do something...
	ENDIF
	
	例 2
	//使用数字组输入的状态作为判断条件，例如定义组输入 gi2 映射了 Profinet IO 的第一个 byte 的前三个 bit，那么当 bit0~bit2 的值分别为 0,1,1 时，gi2 的值为 110（换算成 int 型为 6），组输出（signalgo）同理。
	IF（gi2 == 6）
	do something...
	endif

15.1.17 speed

说明	用来定义机器人和外部轴的运动速度。 为方便用户使用，系统预设了常用的速度变量，可通过辅助编程直接选择使用，详见插入指令。
定义	speed 型变量包含 5 个成员变量，分别是：关节速度百分比，TCP 线速度，空间旋转速度，外轴线速度，外轴角速度。 关节速度百分比（Joint Velocity Percentage），数据类型：double，用于指定关节运动指令时的运动速度，适用于 MoveAbsJ 和 MoveJ 指令，取值范围 1%~100%。 TCP 线速度（Linear Velocity），数据类型：double，定义工具中心点的线速度，取值范围 0.001 毫米/秒 ~ 1000000 毫米/秒。 空间旋转速度（Orientation Velocity），数据类型：double，定义工具的旋转速度，取值范围 0.001 度/秒 ~ 1000000 度/秒。 外部轴线速度，数据类型：double，定义外部直线轴的运动速度，取值范围 0 毫米/秒 ~1000000 毫米/秒。 外部轴角速度，数据类型：double，定义外部旋转轴的运动速度，取值范围 0 度/秒 ~1000000 度/秒。
示例	在变量列表中定义：

定义了一个名为 speed0 的 speed 变量，其中关节旋转速度为最大允许速度的 40%，TCP 线速度为 300 mm/s，空间旋转速度为 100°/s，外轴角速度为 1000°/s，外轴线速度为 200 mm/s。

结构体赋值

speed speed0 = v:{40,300,100,1000,200};

说明：定义了一个名为 speed0 的 speed 变量，其中关节旋转速度为最大允许速度的 40%，TCP 线速度为 300 mm/s，空间旋转速度为 100°/s，外轴角速度为 1000°/s，外轴线速度为 200 mm/s。

名称	关节速度百分比	TCP 线速度	空间旋转速度	外部轴角速度	外部轴线速度
v5	1%	5 mm/s	200°/s	1000°/s	5000 mm/s
v10	3%	10 mm/s	200°/s	1000°/s	5000 mm/s
v25	5%	25 mm/s	200°/s	1000°/s	5000 mm/s
v30	5%	30 mm/s	200°/s	1000°/s	5000 mm/s
v40	5%	40 mm/s	200°/s	1000°/s	5000 mm/s
v50	8%	50 mm/s	200°/s	1000°/s	5000 mm/s
v60	8%	60 mm/s	200°/s	1000°/s	5000 mm/s
v80	8%	80 mm/s	200°/s	1000°/s	5000 mm/s
v100	10%	100 mm/s	200°/s	1000°/s	5000 mm/s
v150	15%	150 mm/s	200°/s	1000°/s	5000 mm/s
v200	20%	200 mm/s	200°/s	1000°/s	5000 mm/s
v300	30%	300 mm/s	200°/s	1000°/s	5000 mm/s
v400	40%	400 mm/s	200°/s	1000°/s	5000 mm/s
v500	50%	500 mm/s	200°/s	1000°/s	5000 mm/s
v600	60%	600 mm/s	200°/s	1000°/s	5000 mm/s
v800	70%	800 mm/s	200°/s	1000°/s	5000 mm/s
v1000	100%	1000 mm/s	200°/s	1000°/s	5000 mm/s
v1500	100%	1500 mm/s	200°/s	1000°/s	5000 mm/s
v2000	100%	2000 mm/s	200°/s	1000°/s	5000 mm/s

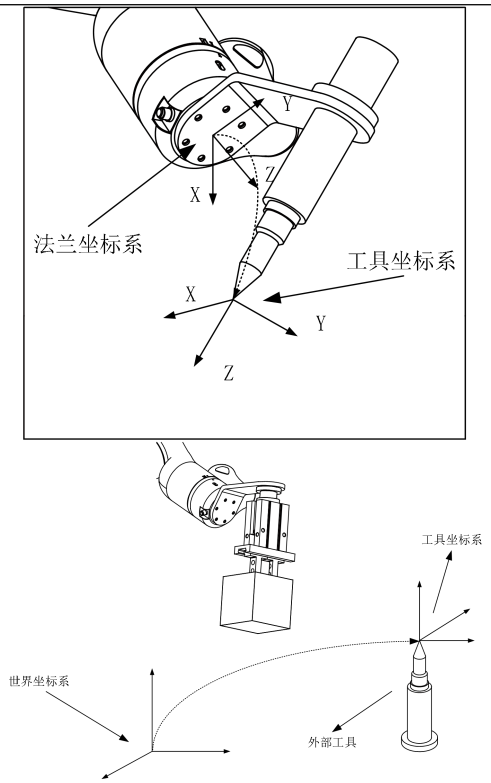
V3000	100%	3000 mm/s	200°/s	1000°/s	5000 mm/s
v4000	100%	4000 mm/s	200°/s	1000°/s	5000 mm/s
v5000	100%	5000 mm/s	200°/s	1000°/s	5000 mm/s
v6000	100%	6000 mm/s	200°/s	1000°/s	5000 mm/s
v7000	100%	7000 mm/s	200°/s	1000°/s	5000 mm/s
vmax	100%	infinite	200°/s	1000°/s	5000 mm/s

系统预定义了常用的速度变量，具体如下表所示。

> 提示
>
> 系统预定义的 speed 变量中所有的空间旋转速度都是 200°/s，如果对机器人末端的旋转速度有特殊要求，可根据工艺要求自行定义新的 speed 变量以便使用。

15.1.18 tool

说明	tool 型变量用来记录工具参数，包括机器人所用工具的 TCP、姿态以及动力学参数。 机器人使用工具与外部环境交互，因此 tool 变量会从以下几个方面影响机器人的运动： 只有工具中心点 TCP 会按照编程的路径和速度运动，当机器人执行一个空间纯旋转运动时，只有 TCP 会保持不动； 编程时指定的运动路径和速度均是指工具坐标系相对于工件坐标系运动的路径和速度，因此更换良好标定后的工具或工件不影响路径的形状和速度大小； 当使用外部工具时，编程的速度指的是工件的速度（相对于外部工具）。 注意，当使用外部工具时，tool 变量中的 tframe 记录外部工具的零点位置和姿态偏移量，而 tload 则用来记录机器人末端所安装的用于抓取工件的手抓的动力学参数。 tool 型变量的数据都存储在数据库中，加载程序时由程序编辑器从数据库中读出，因此请不要尝试在程序编辑器中直接对 tool 型变量直接修改，以免造成不可预知的错误。如果需要修改 tool 型变量，请通过标定界面进行修改，详见标定工具坐标系。 请务必正确定义机器人末端负载的动力学参数，包括工具本身以及由工具抓取的物体两部分。错误的定义可能会导致如下后果： ● 机器人无法最大化利用伺服系统的能力，导致性能下降； ● 路径精度降低，定位误差变大； 机械部件过载导致寿命降低或损坏。
示例	Robhold，数据类型：bool，定义工具是否安装在机器人上，True 表示工具安装在机器人上，False 表示工具没有安装在机器人上，当前正在使用外部工具。在进行 Jog 或者执行程序时，同时使用的工具/工件组合中，robhold 参数只能有一个为 True，即如果工具的 robhold 为 True，则对应的工件 robhold 就必须为 false，反之亦然，否则机器人会给出错误提示，并无法进行 Jog 或执行相应的程序语句。 Tframe，工具坐标系（Tool Frame），数据类型：pose，记录所用工具的工具坐标系，包括： TCP 表示相对于机器人末端法兰坐标系在 x、y、z 三个方向的偏移量，单位是毫米。工具坐标系相对于法兰坐标系的姿态偏移量，用四元数表示，见下图：



Tload，工具的动力学参数，数据类型：load，记录工具的动力学参数，对于普通工具来讲，tload 描述整个工具的动力学参数，对于外部工具来讲，tload 描述机器人用来夹持工件的手抓的动力学参数。对于安装在机器人上的普通工具来讲，其 load 参数包括：

工具的质量（重量），单位 kg；

工具的重心，在法兰坐标系下描述，单位毫米；惯量主轴的方向，在法兰坐标系下描述；

以及工具沿惯量主轴的惯量大小，单位 kg.m2。如果所有的惯量分量都定义为 0 kgm2，则该工具会被当做一个质点（Point Mass）来处理。

注意：

当使用外部工具功能时，工具的 TCP 和姿态是相对于世界坐标系定义的。

如果机器人使用的是外部工具，那 tload 成员则用来记录安装在机器人上的手抓的动力学参数，具体的参数含义保持不变。

请注意 tload 成员仅仅定义机器人用来（夹持工件的）手抓的动力学参数，被夹持的工件的动力学参数并不包含在内，为了保证机器人在任何情况下都获得最佳性能，需要定义两个 tool 变量来处理这种情况：

- 一个 tool 保存手抓本身的所有参数；
- 另一个 tool 保存手抓+被夹持工件的所有参数；

在使用时，通过在运动语句中使用不同的工具来实现有无负载的切换功能。

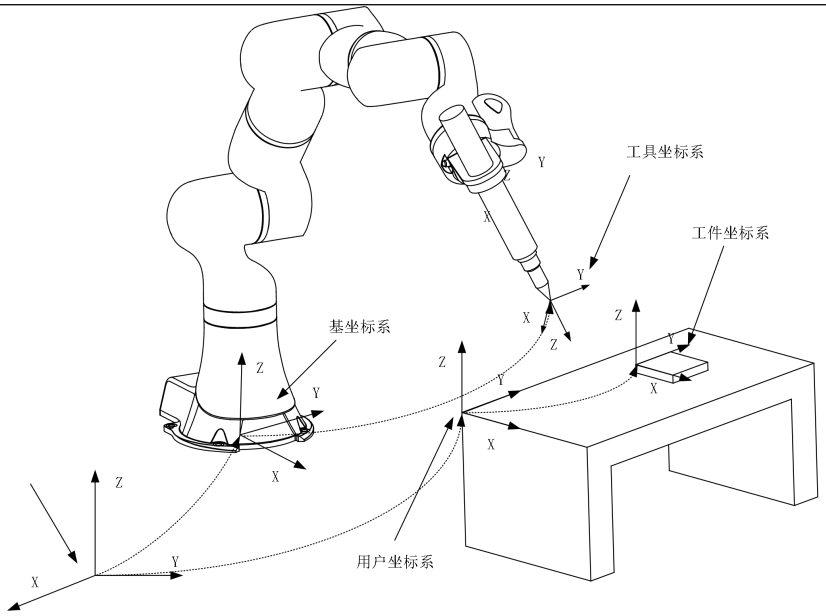
结构体赋值	tool tool0 = {工具是否手持,{{工具位置},{工具姿态四元数}},{质量,{重心 X,重心 Y,重心 Z},{负载姿态四元数},惯量 ix,惯量 iy,惯量 iz}}; 工具是否手持：true，手持；false，外部。 示例：tool tool0 = {true,{{0,0,0},{1,0,0,0}},{0,{0,0,0},{1,0,0,0},0,0,0}};

15.1.19 Trigdata

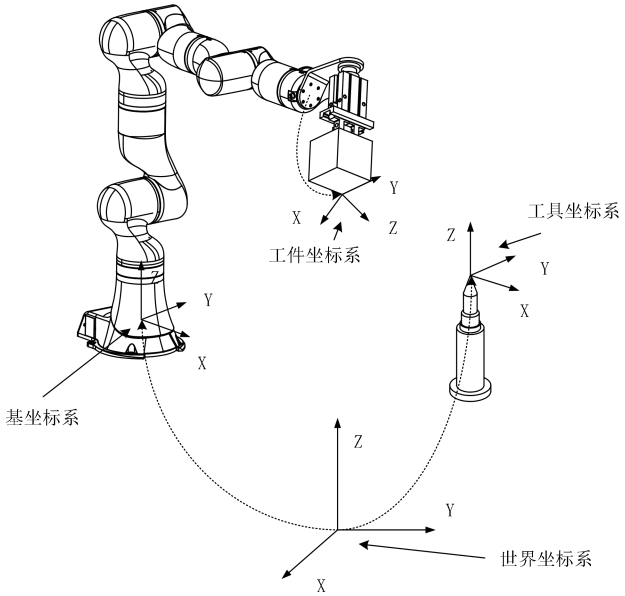
说明	trigdata 用于存储有关机器人移动期间触发事件的信息数据，包括触发条件和触发动作。 触发条件，通常是机器人到达路径上的指定位置点；触发动作，具体形式可以是设置 IO、设置变量等。 trigdata 类型的变量不能通过赋值操作符进行定义，只能通过特定的 RL 指令来定义，因此每一个 trigdata 类型的变量内存储什么样的信息，取决于所使用的 Trig 指令，例如 TrigIO 等； 最终在运动指令 TrigL、TrigC、TrigJ 中，以该数据作为参数使用。
示例	下面的例子显示了如何使用 trigdata： 例 1 VAR trigdata gripopen; TrigIO(gripopen,0.5,do1,true); TrigL(p1,v500,gripopen,fine,tool1);

15.1.20 wobj

说明	wobj 是工件（Work Object）的缩写，工件指被机器人加工、处理、搬运的物体。 所有运动指令中使用的位置都是在工件坐标系下定义的（如果没有指定工件坐标系，则默认在世界坐标系下定义，世界坐标系可以被看作是 wobj0），这样做有如下几个好处： ● 很多加工点的位置可以从工件的设计图纸中获得并直接使用； ● 当机器人被重新安装或者工件被移动后，只需要重新标定工件坐标系就可以直接复用之前的程序，避免了重新编程； ● 在配备合适传感器的情况下，可以自动补偿工件的震动或者轻微移动。 正常情况下，如果不定义专门的工件坐标系，那么控制系统将把世界坐标系当作默认的工件坐标系 wobj0。但是当使用外部工具时，必须要定义工件坐标系，因为此时编程的路径和速度是指工件的路径和速度，而不是工具的。 通常工件坐标系是相对于用户坐标系定义的，但是如果用户未指定用户坐标系，则工件坐标系默认相对于世界坐标系定义，详见机器人坐标系统。 工件实际上由两个坐标系组成，分别是用户坐标系和工件坐标系，在工件坐标系的上层插入一个用户坐标系，是为了支持有多个相同工件需要加工的情况，有关坐标系定义关系的解释详见"定义"部分有关 oframe 的解释。
定义	Robhold，定义该工件是否安装在机器人上。True 表示工件安装在机器人上，当前正在使用外部工具，False 表示工件没有安装在机器人上，当前正在使用普通工具。 Ufprog，用户坐标系是否移动（User Frame Programmed），变量类型：bool，定义用户坐标系是固定的还是移动的，True 表示用户坐标系是固定的，False 表示用户坐标系是移动的，e.g 定义在外部变位机或者其他机器人上。 该值多用于当需要机器人与变位机或其他机器人协调运动时。 Ufmec，用户坐标系关联的机械单元（User Frame Mechanical Unit），数据类型：string，用机械单元名称的方式来指定用户坐标系与哪个机械单元绑定，只有当 ufprog 为 false 时才有用。 Oframe，工件坐标系（Work Object Frame），数据类型：pose，存储工件坐标系的原点和姿态。 uframe_id，用户坐标系（User Frame）id，数据类型：int，存储用户坐标系的 id。可通过 id 找到对应的用户坐标系。 在使用普通工具时（非外部工具），坐标系的定义链为：工件坐标系相对于用户坐标系定义；用户坐标系相对于世界坐标系定义。



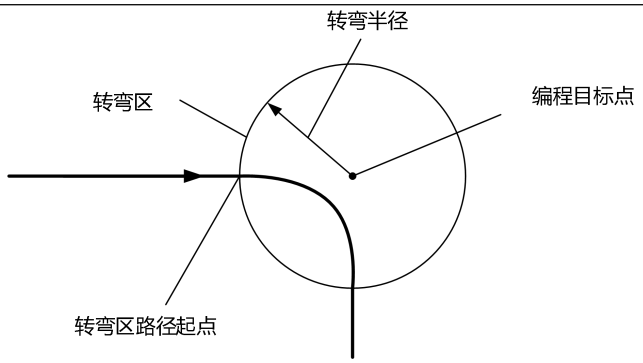
当使用外部工具时，坐标系的定义链为：工件坐标系相对于用户坐标系定义；用户坐标系相对于法兰坐标系定义。



结构体赋值	wobj wobj0 = {Robhold,Ufprog,Ufmec,Oframe,uframe_id,{ 质量,{重心 X,重心 Y,重心 Z},{负载姿态四元数},惯量 ix,惯量 iy,惯量 iz}}; 参数含义见本表格"定义"部分。 示例：wobj wobj0 = {false,true,"robot",{0,0,0},{1,0,0,0},0,{0,{0,0,0},{1,0,0,0},0,0,0}};

15.1.21 zone

说明	zone 变量用于定义某一个运动如何结束或者说定义两条运动轨迹之间转弯区的大小。 对于同一个机器人指令目标点，在运动指令中有两种处理方式： ● 当做停止点处理，机器人将运动到目标点且到达目标点时的速度为 0，之后才会继续执行下一条指令； ● 当做过渡点处理，机器人不会运动到目标点，而是从距离该目标点若干毫米的地方开始转向往下一个目标点运动，转弯路径会偏离编程路径。两条轨迹之间的过渡区域我们称之为转弯区，见下图：



定义的转弯区的大小不能超过路径长度的一半，如果超过，系统会自动将转弯区缩小到总路径长度一半大小。使用转弯区可以避免机器人频繁启停，显著减少节拍时间。

注意：
在某些特定情况下转弯区会被取消，系统会上报"转弯区被取消 Corner Path Failed"日志，可能的原因包括

- 转弯区长度过小（0.01 mm/0.00001 rad）；
- 前后两条轨迹中的至少一条长度过小（1 mm/0.001 rad）；
- 前后两条轨迹接近平行且运动方向相反；
- 前后两条轨迹纯旋转运动转轴反向，如前一条只有末端轴正转，后一条只有末端轴反转。
- 产生"转弯区被取消"警告时，程序自动将受到影响的语句目标点当做停止点处理。
- 除了以上几种特殊情况外，所有的逻辑指令也会导致其上一条运动指令的转弯区被取消。

定义

关节轨迹和笛卡尔空间轨迹使用不同的参数来定义转弯区，所以该变量包括 distance 和 percent 两部分。

Distance，笛卡尔空间转弯区大小，数据类型：double，用于 MoveL、MoveC 和 MoveT 指令，定义笛卡尔空间轨迹的转弯区大小，即当机器人运动到距离目标点还有 distance 毫米的地方时，开始转向往下一个目标点运动，单位是毫米，取值范围 0 ~ 200 mm。

Percent，转弯百分比，数据类型：double，用于 MoveJ 和 MoveAbsJ，表示距离目标角度还有多远时开始转弯，100%表示整个转动角度值的一半；对于只有空间纯旋转运动的 MoveL 指令，也会使用 Percent 参数，而不使用 Distance。

示例

在变量列表中定义：

定义了一个 zone 变量，其中笛卡尔空间转弯区大小为 100 mm，关节空间转弯区大小为 50%。

结构体赋值	zone z1 = s:{Distance,Percent}; 参数含义见本表格"定义"部分。 示例：zone z1 = s:{1,1};

系统预定义了常用的转弯区变量，具体如下表所示。

名称	笛卡尔空间转弯区大小	转弯百分比
fine	0 mm	0%
z0	0.3 mm	0.15%
z1	1 mm	1%
z5	5 mm	3%
z10	10 mm	5%
z15	15 mm	8%
z20	20 mm	10%
z30	30 mm	15%
z40	40 mm	20%
z50	50 mm	25%
z60	60 mm	30%
z80	80 mm	40%
z100	100 mm	50%
z150	150 mm	75%
z200	200 mm	100%

15.1.22 torqueinfo

说明	用来描述机器人受到力和力矩信息； 内部包括轴空间力矩信息以及笛卡尔空间力矩信息两个部分；
定义	joint_torque，数据类型：轴空间力矩信息； cart_torque，数据类型：笛卡尔空间力矩信息； joint_torque.measure_torque，数据类型：double 数组，轴空间测量力信息，力传感器测量到的各轴所受力矩； joint_torque.external_torque，数据类型：double 数组，轴空间外部力信息，控制器根据机器人模型和测量力计算出的各轴所受力矩信息； cart_torque.m_force，数据类型：double 数组，笛卡尔空间(xyz)各个方向受到的力的大小； cart_torque.m_torque，数据类型：double 数组，笛卡尔空间(xyz)各个方向受到的力矩的大小；

示例	下面的例子显示了如何使用变量类型 torqueinfo：
	例 1
	TorqueInfo tmp_info = GetEndtoolTorque(tool1, wobj1);
	//获得在 tool1 wobj1 条件下机器人末端工具的力矩信息结构体
	…
	print(tmp_info.joint_torque.measure_torque);
	print(tmp_info.joint_torque.external_torque);
	//打印各个轴的测量力和外部力
	…
	print(tmp_info.cart_torque.m_torque);
	//打印笛卡尔空间力矩
	…
	print(tmp_info.cart_torque.m_force[1]);
	print(tmp_info.cart_torque.m_torque[1]);
	//打印 X 方向的力和力矩信息

15.1.23 SocketServer

说明	在控制器上建立 Socket TCP 服务器，用来监听外部设备以客户端身份发起的连接。该服务器仅用来监听连接请求，支持多连接。当连接建立之后，会生成一个新的 SocketConn 对象来进行通信。 注意： ● 系统资源申请和释放需要一定时间来处理，不要频繁创建（OpenDev）和销毁（CloseDev）服务器资源，创建和销毁资源期间推荐保留至少 500ms 时间间隔，否则系统资源会超负荷导致出现问题。 ● 该语句仅仅创建了一个服务器资源对象，但是并没有完成服务器创建。服务器需要通过 OpenDev 和 SocketAccept 来进入监听状态。 ● 支持服务器一对多连接。
定义	Ip，数据类型：string，控制系统使用 ip 参数来匹配网卡，并使用对应的网卡进行网络监听。该参数如果设置为"0.0.0.0"则表示监听所有网卡的连接。通常设置为"0.0.0.0"即可。 Port，数据类型：int，监听端口。外部客户端发起连接时，需要指定服务器端口为此设置的数值。 Name，数据类型：string ，RL 程序中使用，此服务器的唯一标识。工程内唯一，多任务间可共享使用，不可命名冲突。

示例	例 1
	SocketServer ss = {"192.168.0.160", 8090, "svr"};
	//仅监听 ip 配置为 192.168.0.160 的网卡
	SocketConn conn = SocketAccept("svr");
	
	例 2
	SocketServer ss = {"0.0.0.0", 8090, "svr"}; //监听机器人所有网卡
	SocketConn conn = SocketAccept("svr");

15.1.24 SocketConn

说明	Socket TCP 连接对象，可用于对外部设备的通信。分为两种类型： ● 机器人作为客户端，通过该对象发起向外部设备 TCP 服务器的连接和通信。 ● 机器人作为服务器，当外部设备的 TCP 客户端发起连接之后，产生的与对端设备的通信连接。当外部不同设备发起多个 TCP 客户端连接，则对每个连接产生一个连接。
定义	Ip，数据类型：string，机器人作为客户端使用时，该参数表示外部设备服务器的 ip。机器人作为服务器使用时，当外部设备建立连接后，该参数表示外部客户端的 ip。 Port，数据类型：int，监听端口。机器人发起连接时，需要指定外部设备服务器端口。 Name，数据类型：string ，RL 程序中使用，此连接的唯一标识。工程内唯一，多任务间可共享使用，连接之间以及连接与服务器的命名不能有冲突。 Cache，数据类型：int ，缓存大小，表示接收数据最多可以缓存的数量。可缺省，默认值为 1。 Suffix，数据类型：string ，结束符，表示一条消息的结束。可缺省，默认值为"\r"。 Attr，数据类型：string ，连接属性。 ● "incoming" ： 本地为服务器，由对端客户端连入。ip、port 标识客户端信息。 ● "outgoing"： 本地为客户端，连接外部服务器。ip、port 为连接对端服务器的信息。 ● ""及其他 ： 不可用连接，表示还没有打开连接或者查找到未建立过的连接。 State，数据类型：string ，当前通信连接状态。Closed：连接已关闭；establish：连接已建立正常使用中。 注意：

	● 做客户端使用时，ip、port 信息应该由用户编程设置。做服务端使用时，ip、port 应该从 accept 指令自动获取。连接建立以后不要轻易修改这两个值，除非您非常清楚这两项的使用，避免程序逻辑使用出错。 ● suffix 随时可以重新设置，在下一次 read 之前可以生效。谨慎使用该功能，否则会造成通信数据错误。suffix 应该在进行通信之前设置好，就不要再更改了。

示例	例 1
	//服务器 ip 为"192.168.0.202"，端口号 8090，此连接名称为"clt"，缓存缺省为 1，后缀缺省为"\r"
	SocketConn scnn1 = {"192.168.0.202", 8090, "clt"};
	
	例 2
	//服务器 ip 为"192.168.0.203"，端口号 8091，此连接名称为"clt1"，缓存为 2，后缀缺省为"\ r"
	SocketConn scnn2 = {"192.168.0.203", 8091, "clt1", 2};
	例 3
	//服务器 ip 为"192.168.0.204"，端口号 8092，此连接名称为"clt2"，缓存为 2，后缀为"\n"
	SocketConn scnn3 = {"192.168.0.204", 8092, "clt2", 2, "\n"};
	
	例 4
	//作为服务器使用，外部设备建立连接以后
	//服务器 ip 为"192.168.0.204"，端口号 8092，此连接名称为"clt2"，缓存为 2，后缀为"\n"
	SocketConn conn = SocketAccept("svr1");
	Print(conn.ip); //外部设备的 ip
	Print(conn.port); //外部设备产生该连接所使用的端口
	Print(conn.cache); //接收消息缓冲区队列
	Print(conn.suffix); //收发后缀

15.1.25 FCBoxVol

说明	定义一个空间立方体，用于力控开启之后的位置监控或终止条件。
定义	Xmax 表示立方体边界在 x 正方向的坐标值； Xmin表示立方体边界在 x 负方向的坐标值； Ymax 表示立方体边界在 y 正方向的坐标值； Ymin 表示立方体边界在 y 负方向的坐标值； Zmax 表示立方体边界在 z 正方向的坐标值； Zmin 表示立方体边界在 z 负方向的坐标值；
示例	
结构体赋值	FcBoxVol fcboxvol0 = FCBV:{Xmin,Xmax,Ymin,Ymax,Zmin,Zmax}; 参数含义见本表格"定义"部分。 示例：FcBoxVol fcboxvol0 = FCBV:{0,1,2,3,4,5};

15.1.26 FCSphereVol

说明	定义一个空间球体，用于力控开启之后的位置监控或终止条件。
定义	Xc空间球体球心坐标在 x 方向的坐标值； Yc 空间球体球心坐标在 y 方向的坐标值； Zc空间球体球心坐标在 z 方向的坐标值； Radius:空间球半径；
示例	
结构体赋值	FcSphereVol fc = FCSV:{xc,yc,zc,radius}; 参数含义见本表格"定义"部分。 示例：FcSphereVol fc = FCSV:{1,2,3,4};

15.1.27 intnum

说明	用作中断的标识。一个 intnum 变量只能标识一个中断。
定义	
示例	

15.2 基础变量和结构体

RL 指令支持的所有变量类型，其中不可再次分割的 int, double, bool, string 类型属于基础变量（也称为基本变量），它们是所有变量类型的基础。把它们按照一定的规则组合在一起的变量类型，被称为结构体。

15.2.1 结构体的构成

结构体的组合规则通常是对具有实际物理意义的数据进行抽象。

示例：

- 结构体 pos 是将三个 double 组合成了一个三维空间的位置（xyz）。
- 结构体 orient 是将四个 double 组合成了一个描述旋转角度的四元数。
- 结构体 pose 是将位置（pos）和旋转角度（orient）组合成了一个描述机器人位置的姿态参数。

15.2.2 结构体的使用方法

结构体除了作为指令的参数，还可根据使用场景，进行更精细的操作。通过特定的 RL 指令，

能够直接修改结构体的数据。

示例 1:

Robtarget 结构体由: 空间位置 (pos) , 姿态 (orient) , 配置数据 (confdata) , 外部轴 (double 数组) 构成。它们的名字分别是 trans(pos), rot(orient), conf(confdata), extax(double)，通过名字可以直接在 RL 函数中直接访问结构体的成员。

```
robtarget rob1 = ... // 变量列表或者用户自定义了一个笛卡尔变量
rob1.trans.x += 20   // 将点位的 x 数据加 20
// trans(pos)的结构体定义中，它由 xyz 三个变量组成
// 所以最后访问 rob 的 x 坐标是 rob1.trans.x
print(rob1.trans) // 只打印位置数据
```

示例 2:

wobj 工件坐标系，可以进行下面操作:

```
// 以默认工件 wobj0 为例
wobj0.robhold // 是否手持工件，bool
wobj0.ufprog // 工件坐标系的用户坐标系是否移动 bool 不常用
wobj0.ufmec   // 工件坐标系的用户坐标系关联的机械单元 string 常用于电镀线、传送到跟踪
wobj0.oframe // 工件坐标系的位姿 Pose
wobj0.oframe.x // 工件坐标系的位姿 Pose 的 x 坐标
wobj0.oframe.y // 工件坐标系的位姿 Pose 的 y 坐标
wobj0.oframe.z // 工件坐标系的位姿 Pose 的 z 坐标
wobj0.q1      // 工件坐标系的位姿 Pose 的四元数
wobj0.uframe_id// 工件相关的用户坐标系 id
```

其他复杂结构体也可参照该方法进行结构体访问。

15.3 函数

使用函数功能可以简化代码结构，提高代码可读性和复用率。用户可以将需要经常执行的程序段定义成新的函数，这样在主程序中就可以方便的随时调用。

15.3.1 函数定义

15.3.1.1 PROC

PROC 表示无返回值的函数，其定义方式为:

```
SCOPE PROC RoutineName()
...
...
//do something
...
...
ENDPROC
```

其中，

1. SCOPE 为函数作用域，支持 GLOBAL 和 LOCAL 两种;
2. PROC 是无返回值函数的定义关键字;
3. RoutineName 为函数名称，命名规则与变量命名规则相同，详见变量命名规则。

辅助编程，可以通过以下方式插入 PROC:

15.3.1.2 FUNC

FUNC 为有返回值的函数，其定义方式为：

SCOPE FUNC RET RoutineName()

…

…

//do something

…

…

ENDFUNC

其中，

1. SCOPE 为函数作用域，支持 GLOBAL 和 LOCAL 两种；
2. FUNC 是有返回值函数的定义关键字；
3. RET 是返回值类型；
4. RoutineName 为函数名称，命名规则与变量命名规则相同，详见变量命名规则。

辅助编程，可以通过以下方式插入 FUNC：

15.3.1.3 TRAP

TRAP 为无返回值的中断函数，其定义方式为：

TRAP TrapName()

…

…

//do something

…

…

ENDTRAP

其中，

1. TRAP 是无返回值中断函数的定义关键字；

4. TrapName 为函数名称，命名规则与变量命名规则相同，详见变量命名规则。

辅助编程，可以通过以下方式插入 TRAP ：

15.3.2 函数调用

调用函数时，直接在程序编辑器输入函数名即可，例如：

RoutineName()

注意：

- 仅能调用本工程内的其它 GLOBAL 级别的函数或本模块文件中 LOCAL 级别的函数，不支持递归调用，不支持两个子函数之间相互交叉调用。
- 对于函数的调用在编译器中被视为一条独立的程序指令。
- 不允许在函数中定义函数。

15.4 指令

15.4.1 变量类型转换

15.4.1.1 ByteToStr

说明	用于将一个 byte 数据按照指定的格式转换成 string 数据
定义	返回值，数据类型：string，转换得到的 string 数据。 ByteToStr (BitData [\Hex] | [\Okt] | [\Bin] | [\Char]); BitData，数据类型：byte，要转换的 byte 数据，默认按照十进制转换。 \Hex，标识符，按 16 进制转换。 \Okt，标识符，按 8 进制转换。 \Bin，标识符，按 2 进制转换。 \Char，标识符，按 Ascii 码字符格式转换。
示例	例 1 VAR byte data1 = 122 VAR string str1

	str1 = ByteToStr(data1); //"122"
	str1 = ByteToStr(data1 \Hex); //"7A"
	str1 = ByteToStr(data1 \Okt); //"172"
	str1 = ByteToStr(data1 \Bin); //"01111010"
	str1 = ByteToStr(data1 \Char); //"z"
	定义 byte 类型变量 data1，赋值 122，将 data1 按不同格式转换得到不同的 string，按十进制转换得到"122"，按十六进制转换得到"7A"，按八进制转换得到"172"，按二进制转换得到"01111010"，按字符转换得到"z"。

15.4.1.2 DecToHex

说明	将十进制的数转换成十六进制数
定义	返回值，数据类型：string，表示转换得到的 16 进制数据，用 0-9，a-f，A-F 表示。 DecToHex(str); str，数据类型：string，表示要被转换的十进制数据，用 0-9 表示。
注意	数据范围从 0~2147483647 或 0~7ffffffff。

15.4.1.3 DoubleToByte

说明	用于将 double 或 double 数组转换成为 byte 数组
定义	返回值，数据类型：byte 数组，double 或 double 数组转换得到的 byte 数组，每 1 个 double 转换得到 8 个 byte。 DoubleToByte(dou1); dou1，数据类型：double，要转换的 double 变量。

15.4.1.4 DoubleToStr

说明	将 double 类型的变量转换为 string
定义	DoubleToStr(Val, Dec); Val，数据类型：double，要转换的 double 变量。 Dec，数据类型：string，要保留的小数点位数。
注意	小数点位数最多 15 位。

15.4.1.5 HexToDec

说明	用于将 16 进制的数转换成 10 进制数
定义	返回值，返回转换得到的 10 进制整型数据，用 0-9 表示。 HexToDec(str); str，数据类型：string，要转换的 16 进制数据，用 0-9，a-f，A-F 表示
注意	数据范围从 0~2147483647 或 0~7ffffffff

15.4.1.6 IntToByte

说明	用于将 int 或 int 数组转换成为 byte 数组
定义	返回值，转换得到的 byte 数组，每 1 个 int 转换得到 4 个 byte。数据类型为 byte 数组。 IntToByte(int1); int1，数据类型：int 或 int 数组，要被转换的整数变量或数组。

注意	数据范围从-2147483647~2147483647

15.4.1.7 IntToStr

说明	用于将整数转换成字符串
定义	返回值，转换得到的字符串。 IntToStr(int1); int1，数据类型：int，要被转换的整数变量。
注意	数据范围从-2147483647~2147483647

15.4.1.8 EulerToQuaternion

说明	用于将欧拉角转成四元数
定义	返回值，表示转换结果， 0 表示正常转换，其他-异常情况。 EulerToQuaternion (type,A,B,C,q1,q2,q3,q4); Type，欧拉角顺规类型，包括 EULER_XYZ 和 EULER_ZYX。 A,B,C，要转换的欧拉角。数据类型： double q1~q4，转换得到的四元数。数据类型： double

15.4.1.9 QuaternionToEuler

说明	用于将四元数转成欧拉角
定义	返回值，表示转换结果， 0 表示正常转换，其他-异常情况。 QuaternionToEuler (type,q1,q2,q3,q4,A,B,C); Type，欧拉角顺规类型，包括 EULER_XYZ 和 EULER_ZYX。 q1~q4，要转换的四元数。数据类型： double A,B,C，转换得到的欧拉角。数据类型： double

15.4.2 运动指令

15.4.2.1 MoveAbsJ

说明	MoveAbsJ （Move Absolute Joint） 用于把机器人和外部轴运动到一个以轴角度定义的位置上，用于快速定位或者移动机器人到某个精确的轴角度。所有的轴同步运动，机器人末端沿一条不规则的曲线移动，请注意是否有发生碰撞的危险。 MoveAbsJ 指令中使用的 tool 参数不会影响机器人的终点位置，但控制器仍然需要使用 tool 参数进行动力学的计算。
定义	MoveAbsJ (ToJointPos， Speed， Zone， Tool， [Wobj]); 其中，带[ ]的参数为可选参数，可不写。 TojointPos，目标关节角度（To Joint Position），数据类型： jointtarget，机器人和外部轴的目标角度和位置值。 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 MoveAbsJ 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 Zone，转弯区（Turning Zone），数据类型： zone，用来定义当前轨迹的转弯区大小。 Tool，数据类型： tool，执行该轨迹时使用的工具，MoveAbsJ 指令使用工具的 TCP 数据来计算运动速度和转弯区大小。 [Wobj]，工件（Work Object），数据类型： wobj，执行该轨迹时使用的工件，当工具安装在机

	器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 内存储的数据来计算运动速度和转弯区大小。
示例	例 1 MoveAbsJ (j10, v500, fine, tool1); 机器人使用工具 tool1，以 v500 的速度沿不规则的路径运动到 j10 定义的绝对关节角度，转弯区大小为 0。 例 2 MoveAbsJ (startpoint, v1000, z100, gripper, phone); 机器人使用工具 gripper，在工件坐标系 phone 下，以 v1000 的速度沿不规则的路径运动到 startpoint 定义的绝对关节角度，转弯区大小为 100 mm。

15.4.2.2 MoveJ

说明	MoveJ （Move The Robot By Joint Movement） 用于对机器人末端运动轨迹没有要求的场合，使机器人快速的从一个点运动到另一个点。所有的轴同步运动，机器人末端沿一条不规则的曲线移动，请注意是否有发生碰撞的危险。 MoveJ 指令与 MoveAbsJ 最大的区别在于给定的目标点格式不同。 MoveJ 的目标点是工具（TCP）的空间位姿而不是关节轴角度。
示例	**MoveJ (ToPoint, Speed, Zone, Tool, [Wobj]);** 其中，带[ ]的参数为可选参数，可不写。 **ToPoint**，目标位姿（To Point），数据类型： robtarget，在笛卡尔空间描述的目标位置。 **Speed**，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 MoveJ 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 **Zone**，转弯区（Turning Zone），数据类型： zone，用来定义当前轨迹的转弯区大小。 **Tool**，数据类型： tool，执行该轨迹时使用的工具，MoveJ 指令使用工具的 TCP 数据来计算运动速度和转弯区大小。 **[Wobj]**，工件（Work Object），数据类型： wobj，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 MoveJ (p30, v100, z50, tool1); 机器人使用工具 tool1， TCP 以 v100 的速度沿不规则的路径运动到 p30 定义的目标点，转弯区大小为 50 mm。 例 2 MoveJ (endpoint, v500, z50, gripper, wobj2); 机器人使用工具 gripper，在工件坐标系 wobj2 下， TCP 以 v500 的速度沿不规则的路径运动到 endpoint 定义的目标位置，转弯区大小为 50 mm。

15.4.2.3 MoveL

说明	用于将工具中心点 TCP 沿直线移动到给定的目标位置。 当起点和终点姿态不同时， 姿态将与位置同步旋转到终点的姿态。由于平移和旋转速度是分开指定的，为了保证不超出指定速度的限制， MoveL 指令最终的运动时间取决于姿态、位置和臂角变化时间较长的那一个。因此在执行某些特定轨迹（例如位移很小而姿态变化很大）时，如果机器人运动速度明显过慢或过快，请检查转动速度设置是否合理。

	当需要保持工具中心点 TCP 静止，而只调整工具姿态时，可以通过为 MoveL 指定位置相同但姿态不同的起点和终点来实现。
定义	MoveL （ToPoint， Speed， Zone， Tool， [Wobj]）； 其中，带[ ]的参数为可选参数，可不写。 ToPoint，目标位姿（To Point），数据类型： robtarget，在笛卡尔空间描述的目标位置。 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 MoveL 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 Zone，转弯区（Turning Zone），数据类型： zone，用来定义当前轨迹的转弯区大小。 Tool，数据类型： tool，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度及旋转速度。 [Wobj]，工件（Work Object），数据类型： wobj。执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 MoveL (p10， v1000， z50， tool0); 机器人使用工具 tool0， TCP 以 v1000 的速度沿直线路径运动到 p10 定义的目标点，转弯区大小为 50 mm。 例 2 MoveL(endpoint， v500， z50， gripper， wobj2); 机器人使用工具 gripper，在工件坐标系 wobj2 下， TCP 以 v500 的速度沿直线路径运动到 endpoint 定义的目标位置，转弯区大小为 50 mm。

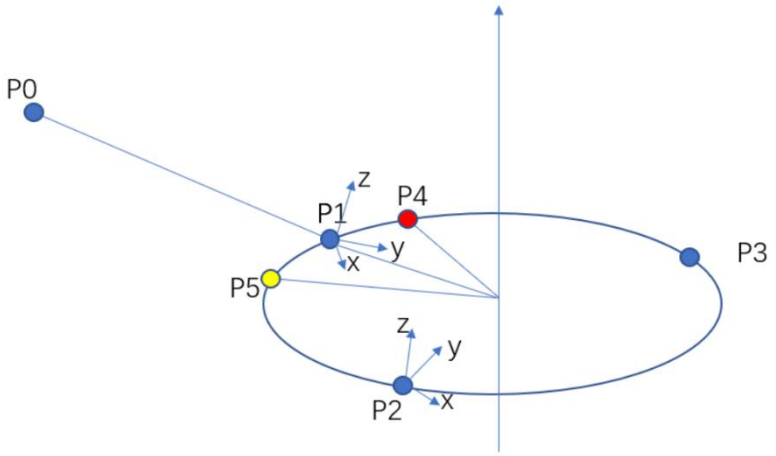
15.4.2.4 MoveC

说明	MoveC （ Move Circle） 用于将工具中心点 TCP 沿圆弧经过中间辅助点移动到给定的目标位置。当起点和终点姿态不同时，姿态将与位置同步旋转到终点的姿态，辅助点的姿态不影响圆弧运动过程。由于平移和旋转速度是分开指定的，为了保证不超出指定速度的限制，MoveC 指令最终的运动时间取决于姿态、位置和臂角变化时间较长的那一个。 因此在某些特定轨迹（例如位移很小而姿态变化很大）下，如果机器人运动速度明显过慢或过快，请检查转动速度设置是否合理。
定义	MoveC（AuxPoint， ToPoint， Speed， Zone， Tool， [Wobj]）； 其中，带[ ]的参数为可选参数，可不写。 AuxPoint，辅助点（Auxiliary Point），数据类型： robtarget，在笛卡尔空间描述的辅助点位置，用于确定圆弧大小和运动方向，该点的姿态不会影响最终轨迹的执行。 ToPoint，目标位姿（To Point），数据类型： robtarget，在笛卡尔空间描述的目标位置。 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 MoveC 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 Zone，转弯区（ Turning Zone），数据类型： zone，用来定义当前轨迹的转弯区大小。 Tool，数据类型： tool，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度及旋转速度。 [Wobj]，工件（ Work Object），数据类型： wobj，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 MoveC(p10， p20， v1000， z50， tool0); 机器人使用工具 tool0， TCP 以 v1000 的速度沿圆弧经过 p10 点运动到 p20 定义的目标位置，

	转弯区大小为 50 mm。 例 2 MoveC (auxpoint， endpoint， v500， z50， gripper， wobj2); 机器人使用工具 gripper，在工件坐标系 wobj2 下， TCP 以 v500 的速度沿圆弧经过 auxpoint 点运动到 endpoint 定义的目标位置，转弯区大小为 50 mm。

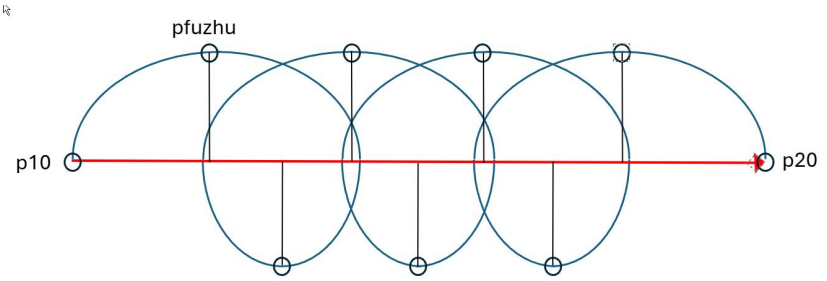
15.4.2.5 MoveCF

说明	MoveCF（Move Circle Full）用于将工具中心点 TCP 沿着由起点和两个辅助点确定的圆形路径，以给定的全圆执行角度进行移动和旋转。MoveCF 全圆运动的终点由全圆执行角度确定，两个辅助点只用来确定圆形路径的空间位置。全圆运动过程中姿态将以参数指定的方式变化，且辅助点的姿态不影响全圆运动过程中的姿态。 MoveCF 指令最终的运动时间只取决于位置变化的时间。因此在圆半径较小的情况下，请检查运动速度设置是否合理，以免发生姿态旋转过快的情况。
定义	MoveCF (AuxPoint1, AuxPoint2, RunDeg, RotType, Speed, Zone, Tool, Wobj); 其中，带[ ]的参数为可选参数，可不写。 AuxPoint，辅助点（Auxiliary Point），数据类型：robtarget，在笛卡尔空间描述的两个辅助点位置。用于确定圆弧的位置和大小，以及运动方向，两个辅助点的姿态不会影响运动的执行。 RunDeg，全圆运动执行角度，数据类型：double，数值范围：-359-3600（请注意机器人末端轴软限位限制），全圆运动的角度，用于确定绕圆运动的圆心角度数。可以是负角度，即按相反方向画圆。 RotType，姿态变化方式，数据类型：char。参数可选范围："ConstPose"、"RotAxis" 、"FixedAxis"，全圆运动的姿态变化方式，支持三种不同的姿态变化方式，请根据需要进行选择： ● "ConstPose"：不变姿态全圆运动，运动过程中，姿态将保持起点的姿态不变。 ● "RotAxis":动轴旋转全圆运动，姿态由起始点姿态和圆的位置决定，运动过程中姿态绕着圆中心轴线变化一周。该方式运动时末端姿态变化较大，请合理设置起点末端轴的角度，以免运动过程中超过软限位(尤其是最后一个关节)。 ● "FixedAxis"：定轴旋转全圆运动，姿态由起始点姿态和圆的位置决定，运动过程中姿态绕着圆中心轴线变化，但不会绕着自身 Z 轴进行旋转。 Speed，运动速度（Move Speed），数据类型：speed，用于指定机器人执行 MoveCF 时的 TCP 运动速度。姿态旋转速度和全圆路径长度相关，不由 speed 参数决定，因此使用时请务必注意该参数的设置，以免末端轴旋转过快。 Zone ，转弯区（Turning Zone），数据类型： zone ，用来定义当前轨迹的转弯区大小。 Tool ，数据类型：tool ，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度。 [Wobj] ，工件（Work Object），数据类型：wobj ，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。

示例	 例 1 MoveJ (P1, v1000, z100, tool1); MoveCF (P2, P3, 360,"RotAxis", v100, z50, tool1); 起点 P1，辅助点 P2、P3 共同确定空间圆形轨迹。 以 P1->P2->P3 为正方向，沿着圆形轨迹运动 360°，回到 P1 点。 运动过程中姿态以起点 P1 姿态为准，绕圆形轨迹中轴线进行相应的旋转。 例 2 MoveJ (P1, v1000, z100, tool1); MoveCF (P2, P3, -330,"ConstPose", v100, z50, tool1); 以 P1->P2->P3 为正方向，沿着圆形轨迹运动-330°，到 P5 点。 运动过程中姿态保持和 P1 姿态相同。 例 3 MoveJ (P1, v1000, z100, tool1); MoveCF (P3, P2, 30,"FixedAxis", v100, z50, tool1); 以 P1->P3->P2 为正方向，沿着圆形轨迹运动 30°，到 P4 点。 运动过程中姿态以起点 P1 姿态为准，绕圆形轨迹中轴线进行相应的旋转，但不会绕着自身 Z 轴进行旋转。

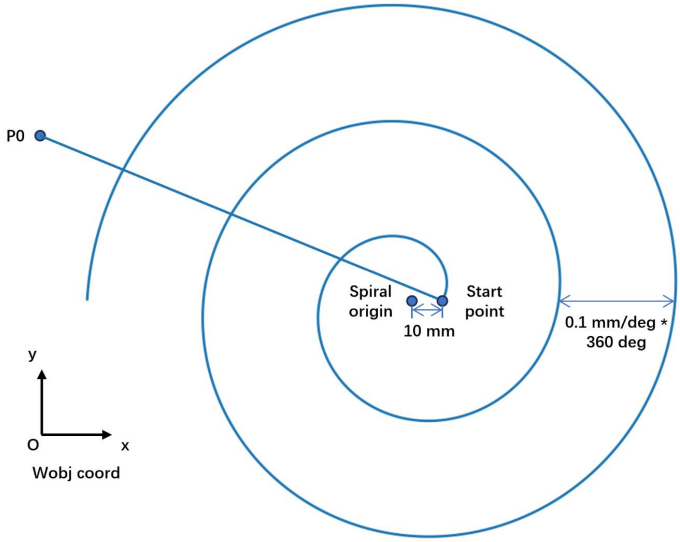
15.4.2.6 MoveT

说明	MoveT (Move trochoid)用于将工具中心点 TCP 按照指定步进长度，经过辅助点画次摆线旋转步进移动到给定的目标位置。 当起点和终点姿态不同时，姿态将与位置同步旋转到终点的姿态，辅助点的姿态不影响螺旋运动过程。 由于平移和旋转速度是分开指定的，为了保证不超出指定速度的限制，MoveT 指令最终的运动时间取决于姿态、位置和臂角变化时间较长的那一个。 因此在某些特定轨迹（例如位移很小而姿态变化很大）下，如果机器人运动速度明显过慢或过快，请检查转动速度设置是否合理。
定义	MoveT (AuxPoint, ToPoint, Radius, Step, Speed, Zone, Tool, [Wobj]); 其中，带[ ]的参数为可选参数，可不写。

	AuxPoint，辅助点（Auxiliary Point），数据类型： robtarget，在笛卡尔空间描述的辅助点位置，用于确定圆弧大小和运动方向，该点的姿态不会影响最终轨迹的执行。 ToPoint，目标位姿（To Point），数据类型： robtarget，在笛卡尔空间描述的目标位置。 Radius，摆线半径，数据类型： double，次摆线前进的半径长度，单位 mm。 Step，步进长度，数据类型： double，次摆线前进的步进长度，单位 mm。 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 MoveT 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 Zone，转弯区（Turning Zone），数据类型： zone，用来定义当前轨迹的转弯区大小。 Tool，数据类型： tool，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度及旋转速度。 [Wobj]，工件（Work Object），数据类型： wobj，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 MoveL (p10, v1000,fine, tool0); MoveT (pfuzhu, p20, 150, 50, v1000, z50, tool0); 机器人使用工具 tool0， TCP 以 v1000 的速度沿圆弧经过 p10 点画次摆线，每次摆线半径为 150mm，步进 50mm，最后运动到 p20 定义的目标位置，转弯区大小为 50 mm。 

15.4.2.7 MoveSP

说明	MoveSP (Move Spiral) 用于将工具中心点 TCP 按照指定初始半径、旋转增率、合计旋转角度和旋转方向，在与工件坐标系 xy 平行的平面内画阿基米德螺旋线。运动过程中姿态线性变化到目标点中的指定姿态。 注意：当 MoveSP 指令运动中途暂停，继续运行，将以当前点为起点重新生成路径，不再继续之前的路径。
定义	MoveSP (Point, Radius, Radius_step, Angle, Direction, Speed, Zone, Tool, [Wobj]); 其中，带[ ]的参数为可选参数，可不写。 Point，目标点，数据类型： robtarget，笛卡尔点位，仅使用其中的姿态作为螺旋线终点的姿态。 Radius，初始半径，数据类型： double，螺旋线的初始半径，单位 mm，要求不小于 0 mm。螺旋线的中心点位置为：TCP 当前位置沿工件坐标系 x 轴负方向移动 Radius 距离。 Radius_step，旋转增率，数据类型：double，螺旋线的旋转增率，单位 mm/deg，要求不小于 0.0001 mm/deg。 Angle，合计旋转角度，数据类型：double，螺旋线的合计旋转角度，单位 deg，要求不小于 0.1 deg，且不大于 3600deg。 Direction，旋转方向，数据类型： int，0：顺时针；1：逆时针。 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 MoveSP 时的运动速度，包括机器人末端的平移速度、旋转速度。 Zone，转弯区（Turning Zone），数据类型： zone，用来定义当前轨迹的转弯区大小。螺旋线

	暂不支持转弯区，系统会自动取消螺旋线前后的转弯区。 Tool，数据类型： tool，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度及旋转速度。 [Wobj]，工件（Work Object），数据类型： wobj，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；该指令仅支持外部工件。TCP 运动平面与工件坐标系 xy 平面平行。
示例	例 1 MoveL (Start_point, v1000, fine, tool1); MoveSP (Point, 10, 0.1, 900, 1, v100, fine, tool1); 首先机器人使用工具 tool1， TCP 以 v1000 的速度从 P0 沿直线运动 Start_point。 然后 TCP 以 v100 的速度进行阿基米德螺旋线运动，螺旋线中心点在 Start_point 沿工件坐标系 x 轴负方向 10 mm 处，从工件坐标系 z 轴向下看旋转方向为逆时针，每旋转 1°半径增大 0.1 mm，合计旋转 900°后停下。 

15.4.2.8 SearchL

说明	SearchL（Search Liner） 用于沿直线移动工具中心点时搜索位置。在移动时，机器人会监控一个数字输入信号（DI）或只读寄存器。当监测的信号状态符合触发模式时，机器人立即读取当前位置。当由机械臂固定的工具为用于表面探测的探针时，通常可使用该指令。使用 SearchL 指令，可获得工件的概略坐标。本指令仅可用于运动任务。
定义	SearchL ([action,] [signal_type], [trigger_mode,] save_rob, target_rob, Speed, Tool ,Wobj); 其中，带[ ]的参数为可选参数，可不写。 Action，触发 DI 后的行为，数据类型：关键字，无参数：不停止 ● \Stop：快速停止，可能导致机器人偏离路径，不限制速度\PStop：规划停止，机器人会在预定轨迹上停止，不限制速度 signal_type，信号类型，数据类型：关键字，无参数：DI 信号 ● \DI：DI 信号 ● \Reg：寄存器信号 signal_name，数据类型：DI 信号或寄存器，SearchL 指令触发特定行为的信号，使用用户自定义 DI 信号或在通信-寄存器中创建的寄存器 trigger_mode，DI 信号触发模式，数据类型：关键字，无参数：默认为上升沿触发 ● \Flanks：边缘触发（上升沿或下降沿） ● \Posflank：上升沿触发

	● \Negflank：下降沿触发 ● \Highlevel：高电平触发 ● \Lowlevel：低电平触发 对于数值型寄存器来说，0 代表低电平，非 0 代表高电平 save_rob，数据类型： robtarget，存储机器人触发信号时位置数据的点位 target_rob，数据类型： robtarget，直线运动的目标点位 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 Search 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 Tool，数据类型： tool，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度及旋转速度。 Wobj，工件（Work Object），数据类型： wobj，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 SearchL(di0, save_rob, target_rob, v500, tool0); 机器人使用工具 tool0， TCP 以 v500 的速度朝 target_rob 直线运动，如果运动过程中 di0 跳变为高电平，将信号跳变时机器人的坐标信息记录在 save_rob。 例 2 SearchL(\PStop, di0, \Lowlevel, save_rob, target_rob, v500, tool0); 机器人使用工具 tool0， TCP 以 v500 的速度朝 target_rob 直线运动，如果运动过程中 di0 为低电平，机器人将立即规划停止，并在检测到信号为低电平时将机器人的坐标信息记录 save_rob。 例 3 SearchL(\PStop, \Reg, register0, save_rob, target_rob, v500, tool0, wobj1); 机器人使用工具 tool0 工件 wobj1， TCP 以 v500 的速度朝 target_rob 直线运动，如果运动过程中 register0 由 0 变为非 0，机器人将立即规划停止，并将机器人的坐标信息记录 save_rob。

15.4.2.9 SearchC

说明	SearchC（Search Circle） 用于沿圆周移动工具中心点时搜索位置。 在移动时，机器人会监控一个数字输入信号（DI）或只读寄存器。当监测的信号状态符合触发模式时，机器人立即读取当前位置。 当由机械臂固定的工具为用于表面探测的探针时，通常可使用该指令。使用 SearchC 指令，可获得工件的概略坐标。本指令仅可用于运动任务。
定义	SearchC ([action,] di, [trigger_mode,] save_rob, aux_rob, target_rob, Speed, Tool [,Wobj]); 其中，带[ ]的参数为可选参数，可不写。 Action，触发 DI 后的行为，数据类型：关键字无参数：不停止 ● \Stop：快速停止，可能导致机器人偏离路径，但停止速度更快，仅当速度小于 v100 时可用 ● \PStop：规划停止，机器人会在预定轨迹上停止，不限制速度 signal_type，信号类型，数据类型：关键字，无参数：DI 信号 ● \DI：DI 信号 ● \Reg：寄存器信号 signal_name，数据类型：DI 信号或寄存器，SearchC 指令触发特定行为的信号，使用用户自定义 DI 信号或在通信-寄存器中创建的寄存器 trigger_mode，DI 信号触发模式，数据类型：关键字，无参数：默认为上升沿触发

	● \Flanks：边缘触发（上升沿或下降沿） ● \Posflank：上升沿触发 ● \Negflank：下降沿触发 ● \Highlevel：高电平触发 ● \Lowlevel：低电平触发 对于数值型寄存器来说，0 代表低电平，非 0 代表高电平 save_rob，数据类型： robtarget，存储机器人触发信号时位置数据的点位 aux_rob，数据类型： robtarget，圆周运动的辅助点 target_rob，数据类型： robtarget，圆周运动的目标点位 Speed，运动速度（Move Speed），数据类型： speed，用于指定机器人执行 Search 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度。 Tool，数据类型： tool，执行该轨迹时使用的工具，指令中的速度指的是该工具的 TCP 速度及旋转速度。 [Wobj]，工件（Work Object），数据类型： wobj，执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 SearchC (di0, save_rob, aux_rob, target_rob, v500, tool0); 机器人使用工具 tool0， TCP 以 v500 的速度经过辅助点 aux_rob 朝 target_rob 圆周运动，如果运动过程中 di0 跳变为高电平，将信号跳变时机器人的坐标信息记录在 save_rob。 例 2 SearchC (\PStop, di0, \Flanks, save_rob, target_rob, v500, tool0); 机器人使用工具 tool0， TCP 以 v500 的速度经过辅助点 aux_rob 朝 target_rob 直线运动，如果运动过程中 di0 由低电平变为高电平，或高电平变为低电平，机器人将立即规划停止，并在信号跳变时将机器人的坐标信息记录在 save_rob。 例 3 SearchC (\Reg, register0, \Flanks, save_rob, target_rob, v5, tool0); 机器人使用工具 tool0， TCP 以 v5 的速度经过辅助点 aux_rob 朝 target_rob 直线运动，如果运动过程中寄存器 register0 由 0 变为非 0，或非 0 变为 0，机器人将立即规划停止，并在信号跳变时将机器人的坐标信息记录在 save_rob。

15.4.3 Trigger 指令

15.4.3.1 TrigIO

说明	TriggIO 用来将一个 trigdata 设置为在运动过程中输出 I/O 的触发器，支持数字输出 DO 和数字组输出 GO。
定义	TrigIO (TrigData,Distance,RefStart,SignalName,Value); TrigData 数据类型 trigdata，用于存储本条 TrigIO 设置的触发数据的变量； Distance 数据类型 double，非负数(负数当成 0 处理)，定义触发事件在路径上的位置偏移，至于该位置偏移是相对路径的起点 or 终点，由 RefStart 定义； RefStart 数据类型 bool，定义触发位置相对起点(true)还是终点(false)； SignalName 数据类型 signaldo 或 signalgo，与本次定义的 IO 事件关联的数字输出或者数字组输出的信号名称，该信号必须是已经正确设置的输出信号；//add Value 数据类型：bool 或 int，定义触发 IO 事件时，指定输出信号的目标值，给定值的数据类型

	应该与 SignalName 类型相匹配。
示例	参考 TrigL 的示例

15.4.3.2 TrigReg

说明	TrigReg 用来将一个 trigdata 设置为在运动过程中修改寄存器值；寄存器类型支持 int16,bool,float,bit 四种。
定义	TrigReg (TrigData,Distance,RefStart,RegName,Value); TrigData 数据类型 trigdata，用于存储本条 TrigIReg 设置的触发数据的变量； Distance 数据类型 double，非负数(负数当成 0 处理)，定义触发事件在路径上的位置偏移，至于该位置偏移是相对路径的起点 or 终点，由 RefStart 定义； RefStart 数据类型 bool，定义触发位置相对起点(true)还是终点(false)； RegName 寄存器名字，无数据类型。注意：寄存器无法在 RL 中新建，需要用户在"机器人-通信-寄存器"页面新建； Value 数据类型：需支持 int16、bool、float、bit，定义了触发寄存器修改事件时，指定寄存器的目标值，给定值的数据类型应该与 RegName 类型相匹配；如果用户给的值与寄存器类型不符，进行强制类型转换；
示例	参考 TrigL 的示例

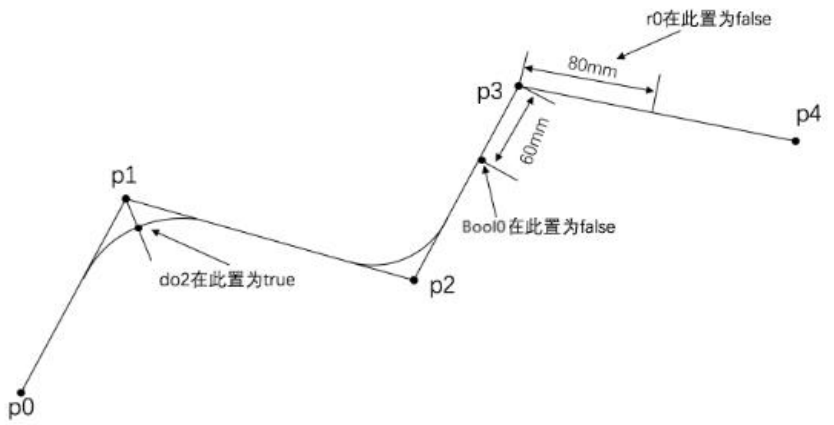
15.4.3.3 TrigVar

说明	TrigVar 用来将一个 trigdata 设置为在运动过程中修改 RL 变量值；RL 变量的类型支持 bool, byte, int, double 四种。
定义	TrigVar (TrigData,Distance,RefStart,VarName,Value); TrigData 数据类型 trigdata，用于存储本条 TrigVar 设置的触发数据的变量； Distance 数据类型 double，非负数(负数当成 0 处理)，定义触发事件在路径上的位置偏移，至于该位置偏移是相对路径的起点 or 终点，由 RefStart 定义； RefStart 数据类型 bool，定义触发位置相对起点(true)还是终点(false)； VarName 希望修改的 RL 变量名，无数据类型； Value 数据类型：需支持 bool、byte、int、double，定义了触发 RL 变量修改事件时，指定 RL 变量的目标值，给定值的数据类型应该与 VarName 类型相匹配；如果用户给的值与 RL 变量类型不符，进行强制类型转换；
示例	参考 TrigL 的示例

15.4.3.4 TrigL

说明	TrigL 与 MoveL 类似，都是执行空间直线运动的指令，区别是 TrigL 可以在运动过程中的若干指定位置执行预定义的操作，其他参数的数量与含义二者之间并无差异。
定义	TrigL (ToPoint,Speed,Trigger,Zone,Tool,[Wobj]); ToPoint 目标位姿(To Point),类型:robtarget，笛卡尔空间描述的目标位姿； Speed 运动速度，类型：speed，用于指定机器人执行 MoveL 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度； Trigger 触发条件和动作，类型：trigdata；trigdata 必须是使用 TrigX 指令处理过之后的 trigdata，否则运动到该行时编译器会给出错误提示； Zone 转弯区，类型：zone，用来定义当前轨迹的转弯区大小； Tool 工具，类型：tool； [Wobj] 工件，类型：wobj；执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽

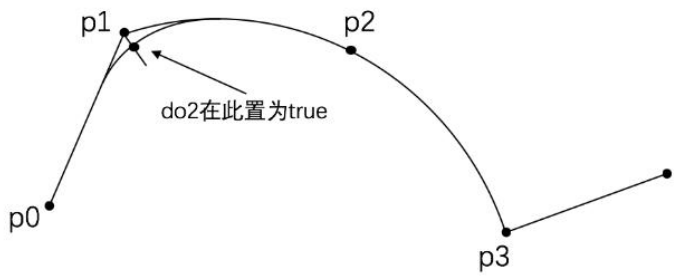
	略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。

	例 1
	VAR trigdata tc1
	VAR trigdata tc2
	VAR trigdata tc3
	…
	//设置 tc1、tc2、tc3
	TrigIO (tc1,0,true,do2,true);
	TrigIO (tc2,60,false,do2,false);
	TrigVar (tc2,60,false,Bool0,false); //Bool0 为 bool 类型的 RL 变量
	TrigReg (tc3,80,true,r0,false); //r0 为 bool 类型的寄存器
	…
	//运动
	MoveL (p1,v500,z50,tool1);
	TrigL (p2,v500,tc1,z50,tool1);
	TrigL (p3,v500,tc2,fine,tool1);
	TrigL (p4,v500,tc3,fine,tool1);
	 
	
	Bool0

15.4.3.5 TrigC

说明	TriggC 与 MoveC 类似，都是执行圆弧运动的指令，区别是 TriggC 可以在运动过程中的若干指定位置执行预定义的操作，其他参数的数量与含义二者之间并无差异。
定义	TrigC (AuxPoint,ToPoint,Speed,Trigger,Zone,Tool,[Wobj]); AuxPoint 辅助点(Auxiliary Point)，类型:robtarget，笛卡尔空间描述的目标位姿； ToPoint 目标位姿(To Point),类型:robtarget，笛卡尔空间描述的目标位姿； Speed 运动速度，类型：speed，用于指定机器人执行 MoveL 时的运动速度，包括机器人末端的平移速度、旋转速度以及外部轴的运动速度；

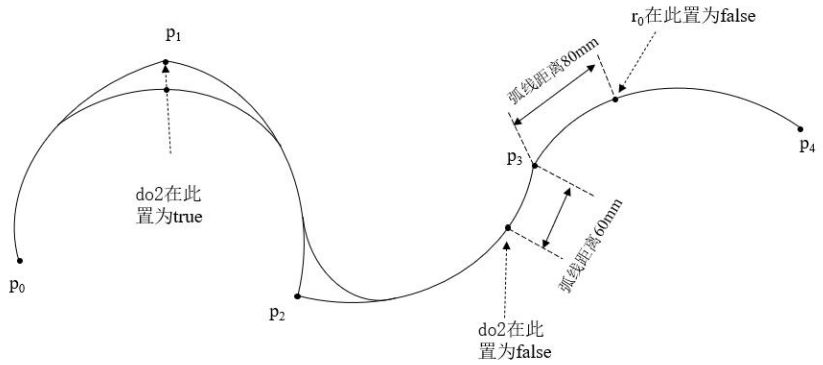
	Trigger 触发条件和动作，类型：trigdata；trigdata 必须是使用 TrigX 指令处理过之后的 trigdata，否则运动到该行时编译器会给出错误提示； Zone 转弯区，类型：zone，用来定义当前轨迹的转弯区大小； Tool 工具，类型：tool； [Wobj] 工件，类型：wobj；执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。

示例	例 1
	VAR trigdata tc1
	...
	//设置 tc1
	TrigIO (tc1,0,true,do2,true);
	...
	//运动
	MoveL (p1,v500,z50,tool1);
	TrigC (p2,p3,v500,tc1,fine,tool1);
	 

15.4.3.6 TrigJ

说明	TrigJ 与 MoveJ 的运动形式完全相同，都是执行关节空间运动的指令，区别是 TrigJ 可以在运动过程中的若干指定位置执行预定义的操作，其他参数的数量与含义二者之间并无差异。 机器人在执行 MoveJ 指令时，tcp 的运动轨迹通常是一条弧线。触发 Trigger 信号时，TrigJ 也是按照 tcp 所走的弧线计算距离的，详情可参看示例。
定义	TrigJ (ToPoint,Speed,Trigger,Zone,Tool,[Wobj]); ToPoint 目标位姿(To Point),类型:robtarget，笛卡尔空间描述的目标位姿； Speed 运动速度，类型：speed，用于指定机器人执行 MoveJ 时的运动速度，包括机器人关节速度，末端的平移速度、旋转速度以及外部轴的运动速度； Trigger 触发条件和动作，类型：trigdata；trigdata 必须是使用 TrigX 指令处理过之后的 trigdata，否则运动到该行时编译器会给出错误提示； Zone 转弯区，类型：zone，用来定义当前轨迹的转弯区大小； Tool 工具，类型：tool； [Wobj] 工件，类型：wobj；执行该轨迹时使用的工件。当工具安装在机器人上时，该参数可以忽略；当使用外部工具时，必须要指定该参数，且机器人将使用 wobj 中存储的数据来计算运动速度和转弯区大小。
示例	例 1 VAR trigdata tc1; VAR trigdata tc2;

```
VAR trigdata tc3;
...
//设置 tc1、tc2、tc3
TrigIO (tc1,0,true,do2,true);
TrigIO (tc2,60,false,do2,false);
TrigReg (tc3,80,true,r0,false);      //r0 为 bool 类型的寄存器
...
//运动
MoveJ (p1,v500,z50,tool1);
TrigJ (p2,v500,tc1,z50,tool1);
TrigJ (p3,v500,tc2,fine,tool1);
TrigJ (p4,v500,tc3,fine,tool1);
```



15.4.4 力控指令

15.4.4.1 CalibSensorError

说明	标定力矩传感器零点
定义	没有参数，直接使用。
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); CalibSensorError();
注意	● 只适用于协作机型的力矩传感器零点标定，工业机型不生效。 ● 机械臂需处于静止状态，任意位姿均可调用此接口标定，使用时请注意设置正确负载，否则可能会出现报错。

15.4.4.2 FcInit

说明	用于力控开启前的一些初始化工作，如设置工件、工具和力控坐标系。
定义	FcInit (Tool, Wobj, ForceFrameRef); Tool，数据类型：pose，力控所使用的工具，力控坐标系的原点是该工具的 TCP（姿态与第三个参数中所选择的坐标系姿态相同）。需要注意的是，所有使用的转接法兰都需要包含在工具的定义中。 Wobj，数据类型：pose，力控所使用的工件，很多力控功能的定义是相对于工件坐标系来的，例

	如力控坐标系的姿态、搜索模式和终止条件等。该参数默认为 Wobj0。 ForceFrameRef，数据类型：int，定义力控坐标系相对于哪个坐标系定义，支持： ● 0：世界坐标系 ● 1：工件坐标系 ● 2：工具坐标系 ● 3：基坐标系 默认值为世界坐标系（0）。
示例	例 1 FcInit (Tool1, Wobj0, 0); 力控初始化，并且定义力控所使用的工具 tool1 和工件 wobj0，以及力控坐标系相对于世界坐标系定义。
注意	FcInit 和 FcStop 之间不允许再次调用 FcInit。

15.4.4.3 SetControlType

说明	设置阻抗控制类型。
定义	SetControlType (ctrl_type); ctrl_type，数据类型：int，阻抗控制类型，支持： ● 0：关节阻抗 ● 1：笛卡尔阻抗
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType(0); FcInit 之后设置阻抗控制模式为关节阻抗。
注意	FcInit 之后，FcStart 之前，不满足此限制，阻抗类型将无法设置成功。 设置阻抗控制类型后，需配合设置对应阻抗刚度来运行。如设置控制类型为关节阻抗，应该继续调用 SetJntCtrlStiffVec 接口设置关节阻抗刚度；若只设置笛卡尔阻抗刚度，那么当机器人运动时，笛卡尔刚度设置并不生效；若同时关节阻抗和笛卡尔阻抗刚度，那么当机器人运动时，只有关节阻抗刚度设置生效；默认关节阻抗和笛卡尔阻抗刚度都为 0。

15.4.4.4 SetCartNsStiff

说明	设置零空间阻抗刚度
定义	SetCartNsStiff(cart_ns_stiff); cart_ns_stiff，数据类型：double，笛卡尔零空间阻抗刚度，取值范围（0~4），单位：Nm/rad。
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType(1); SetCartNSStiff(2); 设置阻抗控制模式为笛卡尔阻抗，零空间刚度设置为 2。
注意	SetControlType 1 之后，也即设置阻抗控制类型为笛卡尔阻抗之后，方可调用此接口，不满足此限制，零空间阻抗参数将无法设置成功。

15.4.4.5 SetJntCtrlStiffVec

说明	设置关节阻抗刚度
定义	SetJntCtrlStiffVec(jnt1_stiff，jnt2_stiff，jnt3_stiff，jnt4_stiff，jnt5_stiff，jnt6_stiff，jnt7_stiff);

	Jnt1_stiff，数据类型：double，关节 1 阻抗刚度，单位：Nm/rad。 Jnt2_stiff，数据类型：double，关节 2 阻抗刚度，单位：Nm/rad。 Jnt3_stiff，数据类型：double，关节 3 阻抗刚度，单位：Nm/rad。 Jnt4_stiff，数据类型：double，关节 4 阻抗刚度，单位：Nm/rad。 Jnt5_stiff，数据类型：double，关节 5 阻抗刚度，单位：Nm/rad。 Jnt6_stiff，数据类型：double，关节 6 阻抗刚度，单位：Nm/rad。 Jnt7_stiff，数据类型：double，关节 7 阻抗刚度，单位：Nm/rad。//设置 6 轴机器人时此参数默认为 0。
示例	例 1 FcInit(Tool1, Wobj0, 0); SetControlType(0); SetJntCtrlStiffVec(1500,1500, 1500,1500,100,100,100); 设置阻抗控制模式为关节阻抗，1~7 关节的阻抗刚度分别设置为 1500、1500、1500、1500、100、100、100。
注意	SetControlType 0 之后，也即设置阻抗控制类型为关节阻抗之后，方可调用此接口，不满足此限制，关节阻抗参数将无法设置成功。

协作机型各轴刚度上限，单位：Nm/rad

	J1	J2	J3	J4	J5	J6	J7
ER3P	6000	6000	6000	1000	1000	1000	1000
ER7P	6000	6000	6000	1000	1000	1000	1000
ER3	3000	3000	3000	300	300	300	
ER7	3000	3000	3000	300	300	300	
SR3	3000	3000	3000	300	300	300	
SR4	3000	3000	3000	300	300	300	
SR5	3000	3000	3000	300	300	300	
CR7	6000	6000	6000	1000	1000	1000	
CR12	20000	20000	20000	3000	2500	2500	
CR17	20000	20000	20000	2500	2500		
CR17/25	20000	20000	20000	2500	2500		
CR18	20000	20000	20000	3000	2500	2500	
CR20	20000	20000	20000	3000	2500	2500	
CR25	20000	20000	20000	2500	2500		

15.4.4.6 SetCartCtrlStiffVec

说明	设置笛卡尔阻抗刚度
定义	SetCartCtrlStiffVec trans（stiff_x，trans_stiff_y，trans_stiff_z，rot_stiff_x，rot_stiff_y，rot_stiff_z）； trans_stiff_x，数据类型：double，X 方向笛卡尔阻抗力刚度，单位：N/m。 trans_stiff_y，数据类型：double，Y 方向笛卡尔阻抗力刚度，单位：N/m。 trans_stiff_z，数据类型：double，Z 方向笛卡尔阻抗力刚度，单位：N/m。 rot_stiff_x，数据类型：double，X 方向笛卡尔阻抗力矩刚度，单位：N.m/rad。 rot_stiff_y，数据类型：double，Y 方向笛卡尔阻抗力矩刚度，单位：N.m/rad。 rot_stiff_z，数据类型：double，Z 方向笛卡尔阻抗力矩刚度，单位：N.m/rad。
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType (1);

	SetCartCtrlStiffVec(1000，1000，1000，100，100，100); 设置阻抗控制模式为笛卡尔阻抗，X/Y/Z 方向的阻抗力刚度设置为 1000，阻抗力矩刚度设置为 100。
注意	SetControlType 1 之后，也即设置阻抗控制类型为笛卡尔阻抗之后，方可调用此接口，不满足此限制，笛卡尔阻抗参数将无法设置成功。

协作机型各轴刚度上限，单位：N/m 及 Nm/rad

	trans_x	trans_y	Trans_z	rot_x	rot_y	rot_z
ER3P	6000	6000	6000	1000	1000	1000
ER7P	6000	6000	6000	1000	1000	1000
ER3	3000	3000	3000	300	300	300
ER7	3000	3000	3000	300	300	300
SR3	3000	3000	3000	300	300	300
SR4	3000	3000	3000	300	300	300
SR5	3000	3000	3000	300	300	300
CR7	6000	6000	6000	1000	1000	1000
CR12	18000	18000	18000	2500	2500	2500
CR17/25	18000	18000	18000	2500	2500	2500
CR18	18000	18000	18000	2500	2500	2500
CR20	18000	18000	18000	2500	2500	2500

15.4.4.7 SetJntTrqDes

说明	设置关节期望力矩
定义	SetJntTrqDes (tau_d1,tau_d2,tau_d3,tau_d4,tau_d5,tau_d6,tau_d7); tau_d1，数据类型：double，关节 1 期望力矩，取值范围（-30~30），单位 N.m。 tau_d2，数据类型：double，关节 2 期望力矩，取值范围（-30~30），单位 N.m。 tau_d3，数据类型：double，关节 3 期望力矩，取值范围（-30~30），单位 N.m。 tau_d4，数据类型：double，关节 4 期望力矩，取值范围（-30~30），单位 N.m。 tau_d5，数据类型：double，关节 5 期望力矩，取值范围（-30~30），单位 N.m。 tau_d6，数据类型：double，关节 6 期望力矩，取值范围（-30~30），单位 N.m。 tau_d7，数据类型：double，关节 7 期望力矩，取值范围（-30~30），单位 N.m。//设置 6 轴机器人时此参数默认为 0。
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType (0); FcStart(); SetJntTrqDes (5,5,5,5,5,5,5); FcStop(); 设置关节期望力矩，所有关节的期望力矩设置为 5N.m。
注意	FcStart 之后，FcStop 之前，方可调用此接口，不满足此限制，关节期望力矩设置不成功。

15.4.4.8 SetCartForceDes

说明	设置笛卡尔期望力/力矩
定义	SetCartForceDes (force_x，force_y，force_z，torque_x，torque_y，torque_z); force_x，数据类型：double，X 方向笛卡尔期望力，取值范围（-60~60），单位 N。 force_y，数据类型：double，Y 方向笛卡尔期望力，取值范围（-60~60），单位 N。

	force_z，数据类型：double，Z 方向笛卡尔期望力，取值范围（-60~60），单位 N。 torque_x，数据类型：double，X 方向笛卡尔期望力矩，取值范围（-10~10），单位 N.m。 torque_y，数据类型：double，Y 方向笛卡尔期望力矩，取值范围（-10~10），单位 N.m。 torque_z，数据类型：double，Z 方向笛卡尔期望力矩，取值范围（-10~10），单位 N.m。
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType(1); FcStart(); SetCartForceDes(0,0,5,0,0,0); FcStop(); 设置笛卡尔期望力/力矩，将 z 方向的期望力设置为 5N。
注意	FcStart 之后，FcStop 之前，方可调用此接口，不满足此限制，笛卡尔期望力/力矩设置不成功。

15.4.4.9 SetSineOverlay

说明	设置绕单轴旋转的正弦搜索运动
定义	SetSineOverlay(line_dir, amplify, frequncy, phase, bias); line_dir，数据类型：int，搜索运动参考轴，支持 ● 0：参考方向为 X 轴 ● 1：参考方向为 Y 轴 ● 2：参考方向为 Z 轴 Amplify，数据类型：double，搜索运动幅值，单位 N.m。 Frequncy，数据类型：double，搜索运动频率，单位 Hz。 Phase，数据类型：double，搜索运动相位，取值范围（-3.14~3.14），单位 rad。 Bias，数据类型：double，搜索运动偏置，取值范围（-10~10），单位 N.m。
示例	例 1 FcInit(Tool1, Wobj0, 0); SetControlType(1); SetSineOverlay(0，10，5，3.14，2); 设置绕 x 轴（0）的旋转搜索运动，幅值为 10N.m，频率为 5Hz，相位为 3.14rad，偏置为 2N.m。
注意	SetControlType 1 之后，也即设置阻抗控制类型为笛卡尔阻抗之后，StartOverlay 之前。方可调用此接口，不满足此限制，正弦搜索运动将无法设置成功。

协作机型参数上限：

	搜索运动幅值上限	搜索运动频率上限
ER3P	10	5
ER7P	10	5
ER3	10	5
ER7	10	5
SR3	5	5
SR3-C	5	5
SR3-A	5	5
SR4	5	5
SR4-C	5	5
CR7	10	5
CR12	10	5

CR17/25	10	5
CR18	10	5
CR20	10	5

15.4.4.10 SetLissajousOverlay

说明	设置平面内的莉萨如搜索运动
定义	SetLissajousOverlay(plane, amplify_one, frequncy_one, amplify_two, frequncy_two, phase_diff); Plane，数据类型：int，搜索运动参考平面，支持 ● 0：参考平面为 XY 平面 ● 1：参考平面为 XZ 平面 ● 2：参考平面为 YZ 平面 amplify_one，数据类型：double，搜索运动一方向幅值，取值范围（-20~20），单位 N.m。 frequncy_one，数据类型：double，搜索运动一方向频率，取值范围（0~5），单位 Hz。 amplify_two，数据类型：double，搜索运动二方向幅值，取值范围（-20~20），单位 N.m。 frequncy_two，数据类型：double，搜索运动二方向频率，取值范围（0~5），单位 Hz。 phase_diff，数据类型：double，搜索运动两个方向相位偏差，取值范围（-3.14~3.14），单位 rad。
示例	例 1 FcInit (Tool1，Wobj0，0); SetControlType (1); SetLissajousOverlay （0，5，2.5，10，5，3.14); 设置 xy 平面(0)内的莉萨如搜索运动，x 方向的幅值为 5N.m，频率为 2.5Hz；x 方向的幅值为 10N.m，频率为 5Hz，y 方向相对于 x 方向的相位偏差为 3.14 rad。
注意	SetControlType 1 之后，也即设置阻抗控制类型为笛卡尔阻抗之后，StartOverlay 之前。不满足此限制，搜索运动将无法设置成功。

15.4.4.11 SetLoad

说明	设置力控模块使用的负载信息
定义	SetLoad(m,rx,ry,rz,Ixx,Iyy,Izz); M，数据类型：double，负载质量，单位：kg，范围（0~25）； Rx，数据类型：double，负载质心在法兰坐标系 x 方向的距离，单位：mm，范围（-300，300）； Ry，数据类型：double，负载质心在法兰坐标系 y 方向的距离，单位：mm，范围（-300，300）； Rz，数据类型：double，负载质心在法兰坐标系 z 方向的距离，单位：mm，范围（-300，300）； Ixx，数据类型：double，负载质心 x 轴的主轴惯量，单位：kg*mm^2，范围（0，100000）； Iyy，数据类型：double，负载质心 y 轴的主轴惯量，单位：kg*mm^2，范围（0，100000）； Izz，数据类型：double，负载质心 z 轴的主轴惯量，单位：kg*mm^2，范围（0，100000）；
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); SetLoad (1,0,0,10,0.001,0.001,0.0001); 设置末端负载，此负载具有这样的信息：质量 1kg，质心在法兰坐标系下的分量分别为 0，0，10mm，负载相对于负载质心坐标系的主轴转动惯量分别为 0.001kg*mm^2，0.001kg*mm^2，0.0001kg*mm^2。
注意	FcStart 之后，方可调用此接口，不满足此限制，负载参数将无法设置成功。

15.4.4.12 FcStart

说明	用于开启力控操作，此操作将机器人从单纯的位置控制切换到力控制
定义	没有参数，直接使用。
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); FcInit 之后通过 FcStart 开启力控，此时机器人处于力控模式。
注意	FcInit 之后调用此接口，且调用之前应确保机器人机械零点、力传感器零点、负载信息已经正确设置，本体参数已经正确辨识。否则，将影响力控功能的使用效果，甚至无法开启力控功能。

15.4.4.13 FcStop

说明	用于停止力控，此操作将机器人从力控制切换到位置控制。执行该指令内部会自动停止所有搜索运动。
定义	没有参数，直接使用。
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); FcStop(); FcStop 停止力控，将机器人从力控制切换到位置控制。执行该指令会清空所有的力控状态。
注意	FcStart 之后调用此接口，调用此接口时会清空力控状态，例如：力控负载信息、阻抗参数、搜索运动、期望力等。如需再次开启力控，则需重新 FcInit。

15.4.4.14 StartOverlay

说明	开启前面设置的搜索运动
定义	没有参数，直接使用。
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType (1); SetSineOverlay (0，10，5，3.14，2); SetLissajousOverlay (0，5，2.5，10，5，3.14); FcStart(); StartOverlay(); 开启搜索运动，搜索运动为之前设置的各种搜索运动的叠加。示例中的搜索运动为绕 x 轴的正弦搜索运动和 xy 平面内的莉萨如搜索运动的叠加。
注意	FcStart 之后，方可调用此接口，不满足此限制，搜索运动将无法开启成功。

15.4.4.15 PauseOverlay

说明	暂停搜索运动
定义	没有参数，直接使用
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType (1); SetSineOverlay (0，10，5，3.14，2);

	FcStart(); StartOverlay(); PauseOverlay(); 暂停搜索运动。
注意	StartOverlay 之后，方可调用此接口。

15.4.4.16 RestartOverlay

说明	重新开启暂停的搜索运动
定义	没有参数，直接使用。
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType(1); SetSineOverlay(0，10，5，3.14，2); FcStart(); StartOverlay(); PauseOverlay(); RestartOverlay(); 重新开启搜索运动。
注意	PauseOverlay 之后，方可调用此接口。此接口和 PauseOverlay 搭配使用，用于重新启动暂停的搜索运动。

15.4.4.17 StopOverlay

说明	停止搜索运动
定义	没有参数，直接使用
示例	例 1 FcInit (Tool1, Wobj0, 0); SetControlType(1); SetSineOverlay(0，10，5，3.14，2); FcStart(); StartOverlay(); StopOverlay(); 停止搜索运动。
注意	StartOverlay 之后，调用此接口才具有实际意义。

15.4.4.18 FcCondForce

说明	用于定义与接触力有关的终止条件
定义	FcCondForce(xmin, xmax, ymin, ymax, zmin, zmax, IsInside, TimeOut); Xmin，定义 X 方向上的力限制下限，为负值时表示 X 负方向上的最大值。单位为 N，默认值为负无穷。数据类型： double Xmax，定义 X 方向上的力限制上限，为负值时表示 X 负方向上的最小值。单位为 N，默认值为正无穷。数据类型： double Ymin，定义 Y 方向上的力限制下限，为负值时表示 Y 负方向上的最大值。单位为 N，默认值为负无穷。数据类型： double Ymax，定义 Y 方向上的力限制上限，为负值时表示 Y 负方向上的最小值。单位为 N，默认值为

	正无穷。数据类型： double Zmin，定义 Z 方向上的力限制下限，为负值时表示 Z 负方向上的最大值。单位为 N，默认值为负无穷。数据类型： double Zmax，定义 Z 方向上的力限制上限，为负值时表示 Z 负方向上的最小值。单位为 N，默认值为正无穷。数据类型： double IsInside，用于定义限制条件内部为 true 还是外部为 true。数据类型： bool TimeOut，定义超时时间，单位为秒，取值范围 1~600。数据类型： double
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); FcCondForce (-100，100，-100，100，-100，100，true，60); 定义一个终止条件，当接触力在力控坐标系 X/Y/Z 轴方向的大小在正负 100N 的范围之内时，条件为 true，当超出 100N 范围时终止。超时时间为 60 秒。
注意	FcStart 之后，FcStop 之前，方可调用此接口，不满足此限制，接触力终止条件无法设置成功

15.4.4.19 FcCondPosBox

说明	用于定义与接触位置有关的终止条件
定义	FcCondPosBox(SupvFrame, Box, IsInside, Timeout); SupvFrame，用于选择监控的空间体相对哪个坐标系定义。该坐标系是从工件坐标系上叠加一个坐标系转换得来的，转换坐标系由 pose 定义，默认使用 pose0，即不使用任何转换，直接使用工件坐标系。数据类型： pose 。 Box，定义一个长方体。数据类型： fcboxvol IsInside，用于定义限制条件内部为 true 还是外部为 true。数据类型： bool TimeOut，定义超时时间，单位为秒，取值范围 1~600。数据类型： double
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); VAR fcboxvol box1 = fcbv:{-100.0, 100.0, -200.0, 200.0, -300.0, 300.0}; VAR pose pose1 = pe:{{0, 0, 0},{1, 0, 0, 0}}; FCCondPosBox (pose1, box1, false, 60); 定义一个终止条件，当机器人 TCP 进入定义的长方体内部或等待超过 60 秒 ，触发终止条件。
注意	FcStart 之后，FcStop 之前，方可调用此接口，不满足此限制，立方体位置终止条件无法设置成功。

15.4.4.20 FcCondTorque

说明	用于定义与接触力矩有关的终止条件
定义	FcCondTorque(xmin, xmax, ymin, ymax, zmin, zmax, IsInside, TimeOut); Xmin，定义 X 方向上的力矩限制下限，为负值时表示 X 负方向上的最大值。单位为 N.m，默认值为负无穷。数据类型： double Xmax，定义 X 方向上的力矩限制上限，为负值时表示 X 负方向上的最小值。单位为 N.m，默认值为正无穷。数据类型： double Ymin，定义 Y 方向上的力矩限制下限，为负值时表示 Y 负方向上的最大值。单位为 N.m，默认值为负无穷。数据类型： double Ymax，定义 Y 方向上的力矩限制上限，为负值时表示 Y 负方向上的最小值。单位为 N.m，默认值为正无穷。数据类型： double Zmin，定义 Z 方向上的力矩限制下限，为负值时表示 Z 负方向上的最大值。单位为 N.m，默认

	值为负无穷。数据类型： double Zmax，定义 Z 方向上的力矩限制上限，为负值时表示 Z 负方向上的最小值。单位为 N.m，默认值为正无穷。数据类型： double IsInside，用于定义限制条件内部为 true 还是外部为 true。数据类型： bool TimeOut，定义超时时间，单位为秒，取值范围 1~600。数据类型： double
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); FcCondTorque (-10，10，-10，10，-10，10，true，60); 定义一个终止条件，当接触力矩在力控坐标系各轴方向的大小大于 10Nm，或时间超时 60 秒，则触发终止条件。
注意	FcStart 之后，FcStop 之前，方可调用此接口，不满足此限制，接触力矩终止条件无法设置成功。

15.4.4.21 FcCondWaitWhile

说明	用于激活在之前定义的终止条件，并在当前行等待直到这些条件变为 False 或者超时
定义	没有参数，直接使用
示例	例 1 FcInit (Tool1, Wobj0, 0); FcStart(); FcCondTorque (-10，10，-10，10，-10，10，true，60); FcCondForce (-100，100，-100，100，-100，100，true，60); FcCondWaitWhile(); 激活终止条件，程序阻塞在当前位置，等待触发终止条件。
注意	定义力控终止条件之后

15.4.4.22 FcMonitor

说明	用于开启或关闭力控模块保护监控。 力控保护监控是指控制器在阻抗模式下,使用用户设置的保护参数,来限制机器人的速度、动量、功率等状态，以实现阻抗模式下的保护。
定义	FcMonitor (On);开启力控模块保护监控，用户设置的保护参数在阻抗运动中生效。 FcMonitor (Off);关闭力控模块保护监控，用户设置的保护参数在阻抗运动中不生效，控制器将采用默认保护参数限制机器人的运动状态。

示例	例 1
	FcInit (tool0,wobj0,0);
	SetControlType (0);
	SetFcJointVelMax (1.0, 1.0, 1.0, 0.5, 0.5, 0.5, 0);
	SetFcJointEnergyMax (100, 100, 100, 100, 100, 100, 0);
	FcMonitor (On);//开启力控模块保护监控
	FcStart();
	…
	FcMonitor (Off);//关闭力控模块保护监控
	…
	FcStop();

注意	● 建议在使用阻抗模式的时候，先设置好保护参数，再使用 FcMonitor On 开启力控模块保护

	监控。 ● 如果未设置保护参数或者使用 FcMonitor Off 关闭力控模块保护监控，控制器将生效默认的保护参数。 ● 默认的保护参数限制较大，在使用阻抗模式下容易触发力控保护报错。

15.4.4.23 GetEndToolTorque

说明	用于获取机器人当前的力矩信息
定义	GetEndToolTorque(Tool, Wobj [, RefType]); []表示该参数可以省略 返回值，力矩信息，数据类型：TorqueInfo Tool，当前使用的工具信息。数据类型：Tool Wobj，当前使用的工件信息。数据类型：Wobj RefType，力矩相对的参考系，数据类型：Int ● 0：默认值，末端相对世界坐标系的力矩信息 ● 1：末端相对于法兰盘的力矩信息 ● 2：末端相对于工具坐标系的力矩信息

示例	TorqueInfo tmp_info = GetEndtoolTorque(tool1, wobj1);
	//获得在 tool1 wobj1 条件下机器人末端工具的力矩信息结构体
	…
	Print(tmp_info.joint_torque.measure_torque);
	Print(tmp_info.joint_torque.external_torque);
	//打印各个轴的测量力和外部力
	…
	Print(tmp_info.cart_torque.m_torque);
	//打印笛卡尔空间力矩
	…
	Print(tmp_info.cart_torque.m_force[1]);
	Print(tmp_info.cart_torque.m_torque[1]);
	//打印 X 方向的力和力矩信息

15.4.4.24 SetFcJointVelMax

说明	用于设置阻抗运动时的最大轴速度。
定义	SetFcJointVelMax jnt1(vel，jnt2_vel，jnt3_vel，jnt4_vel，jnt5_vel，jnt6_vel，jnt7_vel);_ Jnt1_ vel,数据类型：double,关节 1 最大速度，单位：rad/s。 Jnt2_ vel,数据类型：double,关节 2 最大速度，单位：rad/s。 Jnt3_ vel,数据类型：double,关节 3 最大速度，单位：rad/s。 Jnt4_ vel,数据类型：double,关节 4 最大速度，单位：rad/s。 Jnt5_ vel,数据类型：double,关节 5 最大速度，单位：rad/s。 Jnt6_ vel,数据类型：double,关节 6 最大速度，单位：rad/s。 Jnt7_ vel,数据类型：double,关节 7 最大速度，单位：rad/s。
示例	SetFcJointVelMax(1.0,1.0,1.0,1.0,0.5,0.5,0.5);

协作机型可设置各轴速度上限，单位：rad/s

	J1	J2	J3	J4	J5	J6	J7
ER3P	4.7	4.7	4.7	4.7	6.2	6.2	6.2

ER7P	4.7	4.7	4.7	4.7	6.2	6.2	6.2
ER3	4.7	4.7	4.7	6.2	6.2	6.2	
ER7	4.7	4.7	4.7	6.2	6.2	6.2	
SR3	4.7	4.7	4.7	6.2	6.2	6.2	
SR4	4.7	4.7	4.7	6.2	6.2	6.2	
SR5	4.7	4.7	4.7	6.2	6.2	6.2	
CR7	4.7	4.7	4.7	6.2	6.2	6.2	
CR12	4.7	4.7	4.7	6.2	6.2	6.2	
CR17	4.7	4.7	4.7	6.2	6.2		
CR18	4.7	4.7	4.7	6.2	6.2	6.2	
CR20	4.7	4.7	4.7	6.2	6.2	6.2	
CR25	4.7	4.7	4.7	6.2	6.2		

15.4.4.25 SetFcCartVelMax

说明	用于设置阻抗运动时的最大笛卡尔速度。
定义	SetFcCartVelMax(vel_x，vel_y，vel_z，vel_a，vel_b， vel_c); Vel_x，数据类型：double，X 方向上最大线速度，单位：m/s。 Vel_y，数据类型：double，Y 方向上最大线速度，单位：m/s。 Vel_z，数据类型：double，Z 方向上最大线速度，单位：m/s。 Vel_a，数据类型：double，绕 X 轴最大角速度，单位：rad/s。 Vel_b，数据类型：double，绕 Y 轴最大角速度，单位：rad/s。 Vel_c，数据类型：double，绕 Z 轴最大角速度，单位：rad/s。
示例	SetFcCartVelMax (1.0,1.0,1.0,0.5,0.5,0.5);

协作机型可设置笛卡尔速度上限，单位：m/s ，rad/s

	Vel_x	Vel_y	Vel_z	Vel_a	Vel_b	Vel_c
ER3P	2.0	2.0	2.0	4.7	4.7	4.7
ER7P	2.0	2.0	2.0	4.7	4.7	4.7
ER3	2.0	2.0	2.0	4.7	4.7	4.7
ER7	2.0	2.0	2.0	4.7	4.7	4.7
SR3	2.0	2.0	2.0	4.7	4.7	4.7
SR4	2.0	2.0	2.0	4.7	4.7	4.7
SR5	2.0	2.0	2.0	4.7	4.7	4.7
CR7	2.0	2.0	2.0	4.7	4.7	4.7
CR12	2.0	2.0	2.0	4.7	4.7	4.7
CR17	2.0	2.0	2.0	4.7	4.7	4.7
CR18	2.0	2.0	2.0	4.7	4.7	4.7
CR20	2.0	2.0	2.0	4.7	4.7	4.7
CR25	2.0	2.0	2.0	4.7	4.7	4.7

15.4.4.26 SetFcJointMomentumMax

说明	用于设置阻抗运动时的关节最大角动量。

定义	SetFcJointMomentumMax(jnt1_moment， jnt2_moment， jnt3_moment， jnt4_moment， jnt5_moment，jnt6_moment，jnt7_moment); Jnt1_moment，数据类型：double，关节 1 最大角动量，单位：kg*m/s。 Jnt2_moment，数据类型：double，关节 2 最大角动量，单位：kg*m/s。 Jnt3_moment，数据类型：double，关节 3 最大角动量，单位：kg*m/s。 Jnt4_moment，数据类型：double，关节 4 最大角动量，单位：kg*m/s。 Jnt5_moment，数据类型：double，关节 5 最大角动量，单位：kg*m/s。 Jnt6_moment，数据类型：double，关节 6 最大角动量，单位：kg*m/s。 Jnt7_moment，数据类型：double，关节 7 最大角动量，单位：kg*m/s。
示例	SetFcJointMomentumMax (0.1, 0.1, 0.1, 0.1, 0.055, 0.055, 0.055);

协作机型可设置各轴角动量上限，单位：kg*m/s

	J1	J2	J3	J4	J5	J6	J7
ER3P	1.0	1.0	1.0	1.0	0.55	0.55	0.55
ER7P	2.0	2.0	1.0	1.0	0.55	0.55	0.55
ER3	1.0	1.0	1.0	0.55	0.55	0.55	
ER7	2.0	2.0	1.0	0.55	0.55	0.55	
SR3	0.55	0.55	0.55	0.2	0.2	0.2	
SR4	1.0	1.0	0.55	0.2	0.2	0.2	
SR5	1.0	1.0	0.55	0.2	0.2	0.2	
CR7	2.0	2.0	1.0	0.55	0.55	0.55	
CR12	3.5	3.5	2.0	1.0	0.55	0.55	
CR17	7.0	7.0	5.5	3.5	2.0		
CR18	3.5	3.5	2.0	1.0	0.55	0.55	
CR20	7.0	7.0	3.5	2.0	1.0	1.0	
CR25	7.0	7.0	3.5	2.0	1.0		

15.4.4.27 SetFcJointEnergyMax

说明	用于设置阻抗运动时关节的最大功率。
定义	SetFcJointEnergyMax (jnt1_ energy，jnt2_ energy，jnt3_ energy，jnt4_ energy，jnt5_ energy，jnt6_ energy，jnt7_ energy); Jnt1_ energy，数据类型：double，关节 1 最大功率，单位：kg.m2/s3。 Jnt2_ energy，数据类型：double，关节 2 最大功率，单位：kg.m2/s3。 Jnt3_ energy，数据类型：double，关节 3 最大功率，单位：kg.m2/s3。 Jnt4_ energy，数据类型：double，关节 4 最大功率，单位：kg.m2/s3。 Jnt5_ energy，数据类型：double，关节 5 最大功率，单位：kg.m2/s3。 Jnt6_ energy，数据类型：double，关节 6 最大功率，单位：kg.m2/s3。 Jnt7_ energy，数据类型：double，关节 7 最大功率，单位：kg.m2/s3。
示例	SetFcJointEnergyMax (100, 100, 100, 100, 100, 100, 100);

协作机型可设置各关节功率上限，单位：kg*m2/s3

	J1	J2	J3	J4	J5	J6	J7
ER3P	2500.0	2500.0	2500.0	2500.0	1500.0	1500.0	1500.0
ER7P	4000.0	4000.0	3000.0	3000.0	2000.0	1000.0	1000.0
ER3	2500.0	2500.0	2500.0	1500.0	1500.0	1000.0	
ER7	4000.0	4000.0	3000.0	2000.0	1000.0	1000.0	

SR3	1500.0	1500.0	1500.0	600.0	600.0	600.0	
SR4	2500.0	2500.0	1500.0	600.0	600.0	600.0	
SR5	2500.0	2500.0	1500.0	600.0	600.0	600.0	
CR7	4000.0	4000.0	3000.0	2000.0	1000.0	1000.0	
CR12	8000.0	8000.0	5000.0	3500.0	1500.0	1500.0	
CR17	8000.0	8000.0	5000.0	1500.0	1500.0		
CR18	4000.0	4000.0	3000.0	2000.0	1000.0	1000.0	
CR20	16000.0	16000.0	8000.0	4500.0	2500.0	2500.0	
CR25	16000.0	16000.0	8000.0	4500.0	2500.0		

15.4.5 拖动回放

15.4.5.1 ReplayPath

说明	对使用拖动示教进行录制的轨迹进行回放复现，可以控制回放时的运行速率。 注意：如果录制拖动轨迹速度较大，回放速率设置太大，容易触发伺服报警损伤机器，建议从低速率慢慢增加。
定义	ReplayPath(path [, rate] [, wobj/tool]); Path，数据类型：path，拖动回放路径类型，在路径列表中定义、拖动示教录制。 Rate，数据类型：double，回放百分比，范围 0.01~3.00。0.01 表示按照拖动时 1%运行速率回放；1.00 表示 100%运行速率原速回放；3.00 表示 300%运行速率回放。 wobj/tool，数据类型：工具|工件。指定多动回放指令的末端设备是某个工具或者工件，回放时机器人会按照对应设备的工具更改回放的控制参数，提高运行的稳定性
示例	例 1 ReplayPath(path , 1, tool1); 使用原始速率进行录制回放。

15.4.6 IO 指令

15.4.6.1 SetDO

说明	设置某个数字输出信号的值，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考例 2
定义	SetDO(DoName，Value); DoName，数据类型：signaldo ，指定需要改变状态的 DO 信号名称，必须是已经在输入输出界面定义过的变量。 Value，数据类型：bool ，DO 信号的目标状态，仅支持 true 和 false。
示例	例 1 SetDO(do2，true); 将 do2 对应的数字输出点置为高电平。 例 2 情况 1：两条带转弯区的运动指令之间使用 IO 指令 MoveL(p1，v1000，z50，tool0); SetDO(do2，true); MoveL(p2，v1000，z50，tool0);

	此时 SetDO 指令在 p1 到 p2 的转弯区起点处触发。 情况 2：带转弯区运动指令之后使用 IO 指令，但后面不再有运动指令 MoveL (p1， v1000， z50， tool0); SetDO (do2， true); ……(后续没有运动指令，或者存在打断转弯区指令) 此时 SetDO 指令在运动指令到达 p1 点后，触发执行。 情况 3：运动指令不带转弯区 MoveL (p1， v1000， fine， tool0); SetDO (do2， true); ……(不管后续指令是什么) 此时 SetDO 指令在运动指令到达 p1 点后，触发执行。

15.4.6.2 SetAllDO

说明	设置所有数字输出信号的值，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	SetAllDO(Value); Value，数据类型：bool ，DO 信号的目标状态，仅支持 true 和 false。
示例	例 1 SetAllDO (true); 将所有数字输出电压置为高电平，绑定了系统功能的 DO 除外。

15.4.6.3 SetGO

说明	设置某个组输出的值，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	SetGO(GoName，Value); GoName ，数据类型：signalgo ，指定需要改变值的 go 信号名称，必须是已经在输入输出界面定义过的变量。 Value ，数据类型：int ，go 信号的目标值.注意：支持的最大值为 2147483648(2 的 31 次方)。
示例	例 1 SetGO (go3，8); 将 go3 对应的一组物理端口的值设置为 8。

15.4.6.4 SetAO

说明	设置某个模拟输出信号的值，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	SetAO(AoName，Value); AoName ，数据类型：signalao ，指定需要改变值的 ao 信号名称，必须是已经在输入输出界面定义过的变量。 Value ，数据类型：double ，ao 信号的目标值.
示例	例 1 SetAO (ao3，5.123); 将 ao3 对应的一组物理端口的值设置为 5.123。

15.4.6.5 PulseDO

说明	用于产生一个脉冲的 DO 信号，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	PulseDO ([\High,] [length,] signal); [\High] ，当指令执行时，不论当前状态，始终将 signal 状态置为高（1）。 [length] ，指定脉冲长度（0.001-2000s）。缺失时默认 0.2s。 数据类型：double 或 int signal ，要产生脉冲的信号。 数据类型：signaldo
注意	PulseDO 过程中如果执行 SetDO/SetGO，PulseDO 失效，按 SetDO/SetGO 执行。

15.4.6.6 PulseReg

说明	指定一个寄存器产生指定时间的一个脉冲信号，并在时间结束后恢复寄存器初值，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	PulseReg (Register, Value, Time); Register，要产生脉冲信号的寄存器名称，数据类型：Bit|Bool 寄存器 Value，指定脉冲信号的值，数据类型：Bool。 Time ，脉冲信号的持续时间，单位秒，限制范围[0.001, 10.0]。 数据类型：double
注意	PulseReg 过程中如果执行 WriteRegByName 或者寄存器等号赋值操作，寄存器的有效值会按最后执行的指令生效，但是在 PulseReg 指定的时间结束后，会恢复执行 PulseReg 之前的初值。

15.4.7 通信指令

RL 程序中机器人与外部设备通信，支持以太网和串口两种方式。在指令集上，设计了一套统一的指令进行资源管理和数据收发，以提供一致的使用体验。

指令集	TCP 客户端	TCP 服务器	串口
OpenDev	Y	Y	Y
SocketAccept	N/A	Y	N/A
CloseDev	Y	Y	Y
SendString	Y	Y	Y
SendByte	Y	Y	Y
ReadBit	Y	Y	Y
ReadByte	Y	Y	Y
ReadDouble	Y	Y	N/A
ReadInt	Y	Y	N/A
ReadString	Y	Y	Y
GetSocketConn	Y	N/A	N/A
GetSocketServer	N/A	Y	N/A
GetBufSize	N/A	N/A	Y
ClearBuffer	Y	Y	Y

15.4.7.1 OpenDev

说明	用于打开监听服务器、作为客户端发起连接、打开串口资源，取决于参数所表示的对象。 打开 SocketServer 对象时，将初始化资源，并且完成端口绑定和端口监听。 打开 SocketConn 对象时，机器人将作为 TCP 客户端，按照预设的 ip 和端口尝试连接外部服务器。

	打开串口资源，将按照窗口参数初始化串口，准备好通信条件。
定义	OpenDev(name); name ，数据类型：string ，客户端对象或者服务端对象或者串口资源的名称

示例	例 1
	SocketConn scnn3 = {"192.168.0.200", 8090, "clt1", 2, "\n"};
	Try
	OpenDev("clt1") // 尝试连接远程服务器，连接成功则 clt1 的 attr 自动修改为 outgoing。
	string readstr = ReadString(30, "clt1");
	..... //对 readstr 的逻辑处理
	string sendstr = "hello server！";
	SendString(sendstr ， "clt1"); //使用 clt1 的客户端连接来发送数据
	... // 一系列代码
	catch(ERROR e); // ERROR 错误类型，包括产生错误的文件、行号、错误码、错误内容等
	... //一系列异常处理
	Endtry
	
	例 2
	SocketServer listener1 = {"192.168.0.200", 8090, "svr1"};
	global pers bool exit = false;
	try
	OpenDev("svr1"); //绑定端口、监听端口
	while(exit != true)
	SocketConn conn = SocketAccept("svr1"); //阻塞式接收连入的客户端
	Endwhile
	catch(ERROR e);
	... //一系列异常处理
	Endtry
	若发生报错，控制系统将抛出异常，并上报错误原因。如果异常没有被 try 代码块捕获，则控制系统将停止程序运行。

15.4.7.2 SocketAccept

说明	阻塞等待客户端连接到来，并完成客户端连接处理。该指令仅在机器人作为 TCP 服务器时使用
定义	返回值，数据类型：SocketConn，外部设备以 TCP 客户端形式连接上机器人后，控制系统产生一个通信对象，用于 RL 程序来控制通信读写。 SocketConn conn = SocketAccept(name); name，数据类型：string，已经准备好的 SocketServer 对象的名称，并且已经使用 OpenDev 打开成功。

示例	例 1
	SocketServer listener1 = {"192.168.0.200", 8090, "svr1"};
	global pers bool exit = false;
	try
	OpenDev("svr1") //绑定端口、监听端口
	while(exit != true)
	SocketConn conn = SocketAccept("svr1"); //阻塞式接收连入的客户端
	conn.name = "client1"; // 重要！ 给通信连接起个名，否则后续难以按名称进行数据读写

	conn.suffix = "\n"; //可选，设置数据包结束符
	Endwhile
	catch(ERROR e)
	... //一系列异常处理
	Endtry
	若发生报错，控制系统将抛出异常，并上报错误原因。如果异常没有被 try 代码块捕获，则控制系统将停止程序运行。

注意	● 指令将阻塞当前任务，因此正确的使用方法是在多任务中使用，单独作为一个低优先级任务持续不断接收并产生通信连接对象 SocketConn。 ● 指令返回一个连接操作对象，并且拥有客户端连接 ip 和 port 信息，可供程序其他部分使用。返回的连接对象是一个 SocketConn 结构体，名称由系统随机分配。拿到连接对象以后，请更改连接对象的名称，避免连接丢失。 ● 支持服务器一对多连接。

提示

为了保证机器人运动控制的稳定性，控制系统只将一部分的计算资源分配给了网络通讯功能，当机器人作为 socket 服务器监听链接的时候，如果收到类似"DDOS 攻击"的极度频繁的网络连接请求或者数据流，可能导致机器人和示教器以及别的设备网络连接断开或者操作卡顿。

要求网络连接或者数据交互的频率低于 1 次/ms。

15.4.7.3 CloseDev

说明	关闭资源，可用于关闭 TCP 通信链接、TCP 监听服务器或者串口资源
定义	CloseDev(name); name，数据类型：string，用于通信的 SocketConn 连接、监听服务器 SocketServer 对象或者串口资源。

示例	例 1
	SocketConn scnn3 = {"192.168.0.200", 8090, "clt1", 2, "\n"};
	Try
	OpenDev("clt1");
	string readstr = ReadString(30, "clt1");
	..... //对 readstr 的逻辑处理
	string sendstr = "hello server! ";
	SendString（sendstr ， "clt1"）； //使用 clt1 的客户端连接来发送数据
	... // 一系列代码
	catch(ERROR e);
	... //一系列异常处理
	endtry
	CloseDev("clt"); //无论是否发生错误，最后关闭 socket 客户端。
	
	例 2
	SocketServer listener1 = {"192.168.0.200", 8090, "svr1"};
	global pers bool exit = false;

	try
	OpenDev("svr1"); //绑定端口、监听端口
	while(exit != true)
	SocketConn conn = SocketAccept("svr1"); //阻塞式接收连入的客户端
	conn.name = "client1"; // 重要！ 给通信连接起个名，否则后续难以按名称进行数据读写
	conn.suffix = "\n"; //可选，设置数据包结束符
	Endwhile
	catch(ERROR e)
	… //一系列异常处理
	Endtry;
	CloseDev("client1"); //关闭与外部 TCP 客户端通信。重要！
	CloseDev("svr1"); //关闭监听服务器
	
	示例 2 中，有两个网络对象存在，必须按照顺序，先关闭通信连接，再关闭服务器对象，否则将产生资源释放不彻底的状态（TCP TIME_WAIT 状态）。
	如果机器人做服务器时，已经与外部建立了多路通信连接，则需要在关闭服务器之前，先将这些通信连接依次关闭掉。
	如果发生了资源释放不彻底，需要重启控制系统才能恢复。不过不必过于担心，控制系统中允许的资源数量有冗余，可以保证少量的资源被占用后，依然可以正常工作。但还是需要避免大量资源因为使用方式不正确而处于占用状态。

15.4.7.4 SendString

说明	对外发送一个字符串。可以是通过网络或串口，取决于参数中的标识符所表示的硬件资源
定义	SendString(StringData，name); StringData，数据类型：string ，待发送的字符串数据。 name ，数据类型：string ，用于发送数据的硬件资源名称，可以是建立好的 TCP 通信连接 SocketConn 对象，也可以是已经成功打开的串口资源。
示例	例 1 SendString("Hello World"，"Socket0"); 通过 Socket0 对外发送 Hello World 字符串。 其中 Socket0 是已经定义并连接成功的 SocketConn 类型。 例 2 VAR String str1 ="Hello World"; SocketSendString(str1，"Serial1"); 通过 Serial1 对外发送 str1 存储的字符串 Hello World。 其中 Serial1 是已经定义并成功打开的串口。

15.4.7.5 SendByte

说明	对外发送一个字节 byte，在需要发送 ASCII 字符时非常有用
定义	SendByte(ByteData, name); ByteData ，数据类型：int 或 byte 或 byte 数组 ，发送一个 0~255 的无符号字节或数组，主要用于发送 ASCII 码。 Name，数据类型：string ，用于发送数据的 Socket 或者串口名称。
示例	例 1

	SendByte(13, "socket0"); 通过 socket0 对外发送一个回车符。 例 2 VAR byte data1 = 13; SendByte(data1, "serial0"); 首先定义一个 byte 类型变量 data1，它其实是一个回车符。然后通过 serial0 对外发送。 例 3 VAR byte data2[2] = {13,17}; SendByte(data2, "socket0"); 通过 socket0 发送一个 byte 类型数组变量 data2。会将数组中的内容全部发出去。 例 4 VAR byte data2[2] = {13,17,20}; SendByte(data2[2], "socket0"); 通过 socket0 发送一个 byte 类型变量，该变量为 data2[2]，仅表示数组中的第 2 个元素。此时会将 data2[2]的值 17 发出去，而不会多发任何其他元素。

15.4.7.6 ReadBit

说明	控制系统按 bit 接收数据。 1）通过网络通信以 TCP 形式接收，外部发送的数据需以 SocketConn 配置好的结束符结尾。 2）通过串口通信方式接收，外部设备只需要发送数据部分即可，无需考虑结束符。
定义	返回值，数据类型：bool 数组，使用 bool 型数组存储接收到的 bit 数据，每一个 bit 对应一个 bool 成员。 Ret = ReadBit(BitNum, TimeOut, name); BitNum ，数据类型：int ，需要读取的 bit 数量，大小应该为 8 的整数倍。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 name ，数据类型：string，通信连接 SocketConn 的名称或者串口的名称。 Ret ，数据类型：bool 数组，接收到的数据，数组第一个元素表示最低位。
示例	例 1 bool groupio[16]; groupio = ReadBit(16, 60, "Socket0"); 通过 ReadBit 指令读取 16 个 bit 的数据存储到名为 groupio 的布尔型数组中，超时时间 60s。假设外部设备发送 ASCII 字符：95+结束符，则机器人接收到"95"。其中"9"的十六进制为 0x39，"5"的十六进制为 0x35，因此用户接收到的数据为 0x3935。此时 groupio 数组从[1]~[16]表示为：1001 1100 1010 1100。[1]是数据低位，与 0x3935 符合。

15.4.7.7 ReadByte

说明	接收一定字节的数据。注意数据之间需要使用逗号隔开
定义	返回值，数据类型：byte 数组，使用 byte 型数组存储接收到的数据。 Ret = ReadByte(ByteNum, TimeOut, name); ByteNum ，数据类型：int ，需要读取的 bit 数量，大小应该为 8 的整数倍。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 name ，数据类型：string，通信连接 SocketConn 的名称或者串口的名称。

	Ret ，数据类型：byte 数组，接收到的数据。
示例	例 1 byte rets[6] = {0,0,0,0,0,0}; rets = ReadByte(6,60,"clt1"); 读取 6 个 byte 的数据存储到名为 rets 的 byte 类型数组中，超时时间 60s。
注意	注意外部设备的 byte 数据之间需要使用逗号隔开，比如发送"1,2,3,4,5,6"。 通过 TCP 发送数据，需要在数据后面加上预设好的结束符。 通过串口发送数据，不需要结束符。

15.4.7.8 ReadDouble

说明	通过 Socket 接收 double 型数据，外部发送的数据需以配置好的结束符结尾。 注意，该指令仅对 TCP 网络通信生效，适用于机器人做客户端和服务端，但不适用于串口。
定义	返回值，数据类型：double 数组，使用 double 型数组存储接收到的数据。 Ret = ReadDouble(DoubleNum, TimeOut, name); DoubleNum ，数据类型：double ，需要读取的 double 数个数，最大为 4096 个。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 name ，数据类型：string，用于接收数据的 Socket 名称。
示例	例 1 double dd[10]; dd =ReadDouble(10, 60, "Socket0") ; 读取 10 个 double 型的数据存储到名为 dd 的 double 型数组中，超时时间 60s。

15.4.7.9 ReadInt

说明	通过 Socket 接收 int 型数据，外部发送的数据需以配置好的结束符结尾。 注意，该指令仅对 TCP 网络通信生效，适用于机器人做客户端和服务端，但不适用于串口。
定义	返回值，数据类型：int ，使用 int 型数组存储接收到的数据。 Ret = ReadInt(IntNum, TimeOut, name); IntNum ，数据类型：int ，需要读取的 int 数个数，最大为 4096 个。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 name ，数据类型：string，用于接收数据的 Socket 名称。
示例	例 1 int ii[10]; ii = ReadInt(10, 60, "Socket0") ; 读取 10 个 int 型的数据存储到名为 ii 的 int 型数组中，超时时间 60s。

15.4.7.10 ReadString

说明	读取字符串并返回，外部发送的数据需以配置好的结束符结尾
定义	返回值，数据类型： string，存储接收到的字符串。 Ret = ReadString(TimeOut, name, [len]); TimeOut ，数据类型： int ，超时时间。 单位 s， 范围 0~86400，默认 60s。 name ，数据类型：string，用于接收数据的 Socket 或者串口名称。 len，数据类型：int，可选参数，使用串口读取时才使用。由于串口中没有加入结束符的概念，因此需要指定长度后才能正确读取和解析。
示例	例 1

	VAR String str1 str1 = ReadString(60, “Socket1”); 从 Socket1 中接收一个字符串，并存储到 str1 中，超时时间 60s。网络通信方式。 例 2 VAR String str1 str1 = ReadString(60, “serial0”,5); 从 serial0 中接收一个长度为 5 的字符串，并存储到 str1 中，超时时间 60s。串口通信方式。

15.4.7.11 GetSocketConn

说明	从 socket 连接名称查找对应的 socket 属性集对象。该指令获取的结果可以用来进行判断和处理逻辑。仅应该将其当作只读对象使用。该指令仅适用于通信连接（包括机器人作为客户端，或者作为服务端已经连接上的用于通信的通道），不适用于监听服务器和串口。
定义	返回值，数据类型：SocketConn，通过字面名称查找到的 socket 属性对象。 Ret = GetSocketConn(name); name ，数据类型：string，通信连接 SocketConn 的名称。 Ret，数据类型：SocketConn ，通过字面名称查找到的 socket 属性对象。

可查询属性	查询方法	含义和示例
ip 地址	ret.ip	字符串，如”192.168.0.161”
端口号	ret.port	整型，如 8090
属性	ret.attr	机器人做服务器：“incoming”； 机器人做客户端：“outgoing”； 如果连接未建立：””或者其他值，通常为空
缓存大小	ret.cache	1~100
名称	ret.name	示例为”client0”
连接状态	ret.state	closed、establish

示例	例 1 SocketConn ret= GetSocketConn(“client0”); 查找名称为”client0”的 SocketConn 对象。可以使用 ret 来获得这个连接的属性，包括 ip 地址、端口号、通信结束符、连接状态等信息。

15.4.7.12 GetSocketServer

说明	从用户定名称查找对应的服务器属性集对象。该指令获取的结果可以用来进行判断和处理逻辑。仅应该将其当作只读对象使用。该指令仅适用于监听服务器（SocketServer 对象），不适用于通信连接（包括机器人作为客户端，或者作为服务端已经连接上的用于通信的通道）和串口。
定义	返回值，数据类型：SocketServer，通过字面名称查找到的服务器属性对象。 Ret = GetSocketServer(name); Name，数据类型：string，通信连接 SocketServer 的名称。 Ret ，数据类型：SocketServer，通过字面名称查找到的 socket 属性对象。

可查询属性	查询方法	含义和示例
ip 地址	ret.ip	字符串，如”192.168.0.161”
端口号	ret.port	整型，如 8090
名称	ret.name	示例为”svr1”
连接状态	ret.state	closed、listening、error

示例	例 1
	SocketServer listener1 = {"192.168.0.200", 8090, "svr1"};
	OpenDev("svr1"); //绑定端口、监听端口
	//根据连接标识 "svr1" 获取 SocketServer 对象，此时 ret 将复制任务 1 中 listener1 的全部状态
	SocketServer ret= GetSocketServer("svr1");
	if(ret.state == "listening") //使用 SocketServer 的 attr 属性，判断是否监听中
	//逻辑处理
	endif

15.4.7.13 GetBufSize

说明	获取串口缓冲区中还剩余多少数据未读，单位为字节。指令仅适用于串口，不适用于 TCP 服务器和客户端。
定义	返回值，数据类型：int，缓冲区中未处理数据量，以字节为单位。 Ret = GetBufSize(name); name ，数据类型：string，串口资源的字面名称。 Ret，数据类型：int，缓冲区中未处理数据量，以字节为单位。
示例	例 1 OpenDev("serial0"); int a = GetBufSize("serial0") ; Print(a);

15.4.7.14 ClearBuffer

说明	清除对应连接的缓冲区，未读取完毕的数据将丢失。支持串口和 socket 数据。 已经被结束符分割完成的数据和还未被分割数据都将被清除。
定义	ClearBuffer(name); name ，数据类型：string，链接的字面名称。
例 1	假设结束符为\r，已经接收 2 份数据和一份正在接收的数据，执行该指令后，缓冲区内的所有数据都将被清空，RL 程序只能读取到清空之后重新发送的数据 123456789\r Abcdefg\r mmmmmm

15.4.7.15 ReadOpcUaVarByName

说明	通过名称读取 OPC-UA 自定义变量的值。
定义	ReadOpcUaVarByName(name, value); name，数据类型：string，OPC-UA 自定义变量名称。 value ，数据类型：bool/byte/int/double/string，用于存放读取到的 OPC-UA 自定义变量的值，如果 value 的类型与 opcua 变量的类型不一致，将自动转换，注意：string 转换为数值类型时，都为 0。 无返回值。
示例	例 1 int value = 0;

	ReadOpcUaVarByName("int_var", value); Print(value);

15.4.7.16 WriteOpcUaVarByName

说明	通过名称修改 OPC-UA 自定义变量的值。
定义	**WriteOpcUaVarByName(name, value);** **name** ,数据类型：string,OPC-UA 自定义变量名称。 **value** ,数据类型：bool/byte/int/double/string,OPC-UA 变量的修改值，如果 value 的类型与 opcua 变量的类型不一致，将自动转换，注意：string 转换为数值类型时，都为 0。
示例	例 1 int value = 0; WriteOpcUaVarByName("int_var", value); WriteOpcUaVarByName("int_var", 123);

15.4.8 网络指令

15.4.8.1 SocketCreate（过期的）

说明	建立一个 Socket 连接，通过使用 Socket 指令，RL 程序可以从外部设备获取数据或者向外发送程序数据。 RL 语言支持同时建立多个不同的 Socket 以便于连接多个外部设备，不同 Socket 之间采用不同的名称来进行区分。Socket 指令是基于 TCP/IP 协议的，因此理论上任何支持 TCP/IP 的外部设备都可以和 RL 程序通信以交换数据。所有发送给 RL Socket 指令的数据（即使用 SocketRead 系列指令接收的数据），都应该以"回车"结尾，在接收到"回车"之前的所有数据都将合并做为同一条数据处理。使用 Socket 功能时，机器人控制器仅支持作为 client 连接外部 server。 最多支持创建 10 个 Socket 连接。 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	返回值，数据类型：bool，如果创建成功返回 true，创建失败返回 false **SocketCreate（"ip_Address", Port, "Name", [Cache] [, "Terminator"]);** **ip_Address**，数据类型：string 定义需要连接 server 的 ipv4 地址，需使用双引号包含。 **Port** ，数据类型：int ，定义 server 端口号。 **Name** ，数据类型：string ，定义新建 Socket 的名称，不同 Socket 之间需指定不同的名称。 **Cache** ，数据类型：int，定义 Socket 缓存大小，通信数据存在缓存队列中，可省略。 **Terminator** ，数据类型：string 定义 Socket 通信的结束符类型，可省略，默认是"\r"。

示例	例子：
	if (SocketCreate("10.0.6.11",8080,"S1",10,"\r"))
	　　// 创建成功
	else
	　　// 错误处理
	endif

注意	由于 TCP/IP 协议资源释放机制限制，请不要频繁调用 SocketCreate 和 SocketClose 指令，否则可能会造成程序运行出错。 为避免循环模式下，频繁调用 SocketCreate 和 SocketClose 指令，两条指令之间最好增加时间延时，例如： SocketClose("S1");

	wait (0.1); SocketCreate("10.0.6.11",8080,"S1",10,"\r");

15.4.8.2 SocketClose（过期的）

说明	关闭 Socket 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	SocketClose（"SocketName"）; SocketName ，数据类型：string ，需要关闭的 Socket 名称。
示例	例 1 SocketClose("Socket0");
注意	不要在 SocketSend 系列指令后直接使用 SocketClose 指令，否则可能造成数据发送失败，等待收到确认消息 后再使用 SocketClose 指令。

15.4.8.3 SocketSendString（过期的）

说明	通过 Socket 对外发送一个字符串。 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	SocketSendString（StringData，"SocketName"）; StringData ，数据类型：string ，待发送的 string 数据。 SocketName ，数据类型：string ，用于发送数据的 Socket 名称。
示例	例 1 SocketSendString（"Hello World"，"Socket0"）; 通过 Socket0 对外发送 Hello World 字符串。 例 2 VAR String str1 ="Hello World" ; SocketSendString（str1，"Socket0"）; 通过 Socket0 发送 str1 存储的字符串。

15.4.8.4 SocketSendByte（过期的）

说明	通过 Socket 对外发送一个字节 byte，在需要发送 ASCII 字符时非常有用。 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	SocketSendByte(ByteData，"SocketName") ; ByteData ，数据类型：int 或 byte 或 byte 数组 ，发送一个 0~255 的无符号字节或数组，主要用于发送 ASCII 码。 SocketName ，数据类型：string ，用于发送数据的 Socket 名称。
示例	例 1 SocketSendByte(13，"socket0"); 通过 socket0 对外发送一个回车符。 例 2

	VAR byte data1 = 13; SocketSendByte(data1, "socket0"); 首先定义一个 byte 类型变量 data1，它其实是一个回车符。然后通过 socket0 对外发送。 例 3 VAR byte data2[2] = {13,17} ; SocketSendByte(data2, "socket0") ; 通过 socket0 发送一个 byte 类型数组变量 data2。

15.4.8.5 SocketReadBit（过期的）

说明	通过 Socket 按 bit 接收数据，外部发送的数据需以回车结尾。 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	返回值，数据类型：bool ，使用 bool 型数组存储接收到的 bit 数据，每一个 bit 对应一个 bool 成员。 SocketReadBit(BitNum, TimeOut, "SocketName")； BitNum ，数据类型：int ，需要读取的 bit 数量，大小应该为 8 的整数倍。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 SocketName ，数据类型：string，用于接收数据的 Socket 名称。
示例	例 1 bool groupio[16]; groupio = SocketReadBit(16, 60, "Socket0"); 通过 SocketReadBit 指令读取 16 个 bit 的数据存储到名为 groupio 的布尔型数组中，超时时间 60s。

15.4.8.6 SocketReadDouble（过期的）

说明	通过 Socket 接收 double 型数据，外部发送的数据需以回车结尾。 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	返回值，数据类型：double ，使用 double 型数组存储接收到的数据。 SocketReadDouble(DoubleNum, TimeOut, "SocketName")； DoubleNum ，数据类型：double ，需要读取的 double 数个数，最大为 30 个。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 SocketName ，数据类型：string，用于接收数据的 Socket 名称。
示例	例 1 double dd[10]; dd = SocketReadDouble(10, 60, "Socket0"); 通过 SocketReadDouble 指令读取 10 个 double 型的数据存储到名为 dd 的 double 型数组中，超时时间 60s。

15.4.8.7 SocketReadInt（过期的）

说明	通过 Socket 接收 int 型数据，外部发送的数据需以回车结尾。 注意，该指令被标注为"过期的"，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。

定义	返回值，数据类型：int ，使用 int 型数组存储接收到的数据。 SocketReadInt(IntNum, TimeOut, “SocketName”); IntNum ，数据类型：int ，需要读取的 int 数个数，最大为 30 个。 TimeOut ，数据类型：int ，超时时间。单位 s，范围 0~86400，默认 60s。 SocketName ，数据类型：string，用于接收数据的 Socket 名称。
示例	例 1 int ii[10]; ii = SocketReadInt(10, 60, “Socket0”); 通过 SocketReadInt 指令读取 10 个 int 型的数据存储到名为 ii 的 int 型数组中，超时时间 60s。

15.4.8.8 SocketReadString（过期的）

说明	从 Socket 读取一个字符串并返回，外部发送的数据应以回车结尾。 注意，该指令被标注为“过期的”，为 xCore 控制系统 1.3 版本使用指令，在更高版本中依然有效，但不再继续维护，也不推荐继续使用。
定义	返回值，数据类型： string，存储接收到的字符串。 SocketReadString(TimeOut, “SocketName”); TimeOut ，数据类型： int ，超时时间。 单位 s， 范围 0~86400，默认 60s。 SocketName ，数据类型：string，用于接收数据的 Socket 名称。
示例	例 1 VAR String str1 str1 = SocketReadString(60, “Socket1”); 从 Socket1 中接收一个字符串，并存储到 str1 中，超时时间 60s。

15.4.9 逻辑指令

15.4.9.1 Return

说明	函数返回。 程序遇到 RETURN 指令时，如果程序当前处于子函数中，则程序将返回到上一级函数中。如果程序当前处于主函数中，则程序直接结束。

15.4.9.2 Wait

说明	程序等待一段时间，范围是 0~2147484 秒
示例	例 1 Wait (2); 表示等待 2s 的时间

15.4.9.3 WaitUntil

说明	程序等待某个条件成立，若超时，则将超时标志置 true，结束等待继续向下执行。
定义	WaitUntil(cond,\MaxTime,\TimeFlag) Cond，bool 类型逻辑表达式 MaxTime，超时等待时间，可选参数；单位 s，使用 int 或 double 类型 TimeFlag，超时标志位，若超时则置 true，可选参数；使用 bool 类型变量
示例	例 1 WaitUntil (di2 == true);

	... 表示等待 di 2 信号值为 true，然后才开始执行后面的语句。 例 2 WaitUntil (di2 == true,5); ... 表示等待 di 2 信号值为 true，若等待超过 5s，di2 信号依然为 false，则执行后面的语句。 例 3 Bool flag = false; WaitUntil (di2 == true, 5,flag); ... 表示等待 di 2 信号值为 true；若等待超过 5s，di2 信号依然为 false，则将 flag 置为 true，然后才开始执行后面的语句；若在 5s 内 di2 变为 true，则 flag 置为 false。可以将 flag 用于后续的逻辑判断

15.4.9.4 Break

说明	跳出当前循环，在 RL 语言中在 WHILE 循环中使用，当 WHILE 循环执行到 Break 时，不管 WHILE 的 CONDITION 如何，都会直接跳出 WHILE 循环。

示例	例 1
	VAR int counter = 0;
	WHILE(1)
	  IF(counter == 5)
	  break;
	  Endif
	counter++;
	ENDWHILE
	该程序在执行到 counter 等于 5 时会跳出 WHILE 循环。

15.4.9.5 IF...Else if...Else

说明	条件判断语句

示例	例 1
	IF(condition1)
	//a
	Else if (condition2)
	//b
	Else if (condition3)
	//c
	Else
	//d
	Endif
	condition1 成立时执行逻辑 a，condition2 成立时执行逻辑 b，以此类推。

15.4.9.6 Goto

说明	Goto 语句允许把指针跳转到被标记的语句

示例	例 1
	int a = 0;
	int b = 9;
	Goto end;
	print(a);
	end:;
	print(b);
	先定义两个变量 a 和 b，然后用 print 函数打印两句话，直接用 Goto 语句强制跳转到打印 b 语句的 end 标记位置，此时 a 的打印就不会执行了

15.4.9.7 For

说明	For 循环允许您编写一个执行指定次数的循环控制结构

示例	例 1
	For(int i from 1 to 10)
	  Print("i = %d\n", i);
	endfor
	该程序把 i 从 1 到 9 每次加 1 依次打印 9 次。
	
	例 2
	For(int i from 1 to 10 step 3)
	  Print("i = %d\n", i);
	Endfor
	该程序把 i 从 1 到 10 每次加 3 依次打印 3 次。
	
	补充说明:
	Continue 和 Break 可以用来控制 For 的流程,详细操作见 Continue 和 Break 指令说明。

15.4.9.8 Continue

说明	跳出本次循环。继续从循环起始处执行下条语句，但不退出循环体，仅仅结束本次循环

示例	例 1
	VAR int count = 0
	WHILE(1)
	count++
	IF(count == 1)
	Continue
	Else
	Break;
	MoveAbsJ(j10, v500, fine, tool1);
	Endif
	ENDWHILE
	MoveAbsJ 的代码将不会被执行到。

15.4.9.9 Inzone

说明	该指令和 SetDO 或者 modbus、cclink 等 IO 操作或指令配合使用，可保证信号在确定的点位触发，不会被前瞻指针提前触发。
示例	MoveL p1 MoveL p2 Inzone SetDO(dox, true); print(123); EndInzone MoveL p3; 补充说明： 在示例中，使用了一个 Inzone 指令，解释器前瞻到 Inzone 之后，并不会立即执行，而是生成了一个附加函数，函数内容是 SetDO 以及 print 指令，这个附加函数会在运动指令 move p2 完成之后生效。 1、如果 p2 p3 两条运动指令之间存在转弯区，则附加函数会在机器人进入两段运动的转弯区的时刻开始执行 2、如果没有转弯区，则附加函数会在机器人到达 p2 的时刻开始执行

15.4.9.10 While

说明	While 循环允许您编写一个在条件满足前不断执行的循环控制结构

示例	例 1
	int count = 0;
	while(count < 10)
	count++;
	print(count);
	endwhile
	该程序实现一个 count 从 0 到 10 每次加 1 并打印的循环。
	补充说明：Continue 和 Break 可以用来控制 While 的流程,详细操作见 Continue 和 Break 指令说明。

15.4.9.11 Pause

说明	暂停程序运行。 程序会在 pause 语句的前一句执行完毕后进入暂停状态，必须使用示教器点击运行或者通过外部程序启动信号才可恢复程序运行。
注意	该指令暂时不支持辅助编程

15.4.9.12 try/catch

说明	try-catch 指令是一种 RL 语言的错误处理机制，try 到 catch 指令中间的指令，如果出错后，程序会将执行错误转换为错误信息集合"e" 并从 catch-endtry 的代码块继续运行

定义	try
	// do something
	catch(error e)
	print(e);
	endtry

	举例，从网络链接读数据是很有可能失败的指令，但是此时不希望机器人停机，可以用 try-catch 将错误捕获并通过 RL 编程处理
	详细说明
	error 类型说明，error 是一个结构体，一共有四个参数组成，分别是
	● file : string (错误发生文件名)
	● line : int (错误行)
	● num : int (错误码)
	● reason : string(错误原因)
	error 结构体可以通过 print 指令直接打印。
	// error data
	...
	catch(error e)
	print(e.line);
	print(e.num);
	print(e.reason);
	print(e);
	endtry

示例	例 1
	ReadOnce:;
	Try
	Double xyz[3] = ReadDouble(3, timeout, socketname);
	Robtarget_0.trans.x = xyz[1];
	Robtarget_0.trans.y = xyz[2];
	Robtarget_0.trans.z = xyz[3];
	
	MoveL (Robtarget_0, v2000, fine, tool0);
	Catch(error e)
	SendString("Recv rob xyz error", socketname);
	Goto ReadOnce;
	endtry
	
	该程序实现了一个简单的应用场景，使用通信指令 ReadDouble 从 TcpSocket 读取一个三维数组作为运动点位的 xyz 参数，然后使用 MoveL 指令运动到对应笛卡尔点。
	如果没有使用 try/catch 指令并且从 TcpSocket 收到点位是错误数据，则机器人会报错"超出运动范围"或者"规划错误"，并且停止程序的运行。
	如果使用了 try/catch 指令，虽然依然会报告运动指令错误，但是程序不会停止，而是跳转到 catch 到 endtry 的代码段，执行用户想要的错误处理。本样例中就是通过 SendString 告诉 Socket 上位机收到的点位错误，再由上位机决定如何处理，并执行 goto 指令重新执行 ReadDouble 等待下一次的位置。
	
	例 2
	re_read:;
	try
	opendev("conn_name");
	string_res = readstring("conn_name");
	catch (error e)

	if (e.num == xxx)
	// 某种可处理的错误不暂停
	goto re_read;
	else
	print(e);
	Pause;
	endif
	endtry

注意	注意：力控指令不能触发 try-catch

try/catch 能够处理的错误类型及标准错误码：

分类	出错指令	说明	error.num	error.reason
默认错误		未分配专属错误码的指令	-1	未知错误
串口相关指令		串口不存在时	-1	未知错误
运动相关指令	MoveXX, Search, TrigL 等 AccRamp, HomeSet 等运动参数设置	运动坐标的工具、工件错误 运动速度错误 运动负载错误 超出运动范围 规划错误 遇到奇异点等	-1	未知错误
网络指令	OpenDev	网络链接的端口错误	-1	未知错误
	所有网络指令	通过 RL 操作外部通讯的连接	-1	未知错误
计算、逻辑指令	CalcJoinT CalcRobt CRobT CJointT CLKSTOP GOTO	控制器内部错误	-1	未知错误
外设控制(Jodell 系列)(RM 系列)	JodellGripInit JodellSuckInit JodellSuckStatus RMRGMGripPosMove RMRGMGripTrqMove RMRGMGripStatus RMRGMResetErr RMCGripPosMove RMCGripTrqMove RMCGripStatus RMCResetErr RMRGMGripInit RMCGripInit	外设通讯异常	-1	未知错误
激光控制	Laser 所有指令	激光焊接已关闭	-1	未知错误
码垛控制	TrayUpdate TrayCount PalletUpdate PalletLayerCount PalletWobjCount SolarVisionExec	与上位机数据收发错误	-1	未知错误
寄存器控	ReadRegByteByName	读取数据失败	-1	未知错误

制				
四轴锁定	SingAreaLockAxis4	位姿错误，无法开启四轴锁定功能	-1	未知错误
解释器内部错误	绝大部分指令的参数类型、数量错误		-1	XXX 参数错误
解释器内部错误			0	
网络指令串口指令	OpenDev	连接到服务器失败	101	OpenDevConn 失败
	OpenDev	机器人作为服务端开启失败	102	OpenDevServer 失败
	GetSocketConn	SocketConn 对应的连接未建立	103	GetSocketConn 失败 连接不存在
	GetSocketConn	获取 SocketConn 结构体的名字是一个服务器	104	GetSocketConn 失败 对象是 SocketServer
	GetSocketServer	GetSocketServer 失败 服务器不存在	105	GetSocketServer 失败 服务器不存在
	OpenDev	开启连接输入参数错误，变量列表无对应连接	106	OpenDev 失败 使用不存在的对象
	SocketAccept	输入参数不是服务器名称	107	SocketAccept(server) 指令需要服务器名称
	GetSocketConn	获取 SocketConn 结构体的名字错误	108	GetSocketConn(conn) 不存在的 SocketConn
	GetSocketServer	获取 SocketConn 结构体的名字错误	109	GetSocketConn(server) 不存在的 SocketServer
	ReadBit	指令输入参数错误	110	ReadBit 必须读取 8 的整数倍数
	ReadDouble	指令输入参数错误	111	ReadDouble 指令超出预设范围(0, 4096]
	ReadInt	指令输入参数错误	112	ReadInt 指令超出预设范围(0, 4096]
	ReadByte	指令输入参数错误	113	ReadByte 指令超出预设范围(0, 4096]
	ReadBit ReadDouble ReadInt ReadByte ReadString	输入的时间太长	114	ReadXX 指令时间超出预设范围(0. 86400]
	ReadBit ReadDouble ReadInt ReadByte ReadString	连接断开 或者 读取数据错误	115	Read 指令失败
	ReadDouble	超出限定时间	116	ReadDouble 指令超时
	ReadInt	超出限定时间	117	ReadInt 指令超时
	ReadString	超出限定时间	118	ReadString 指令超时
	ReadBit	超出限定时间	119	ReadBit 指令超时
	ReadByte	超出限定时间	120	ReadByte 指令超时
	SendString	指令超时或者连接断开	121	SendString 指令超时或者连接断开
	SendByte	指令超时或者连接断开	122	SendByte 指令超时或者连接断开
传送带跟踪指令	WaitObj	执行指令时，工件已经越过启动窗口，无法跟踪	123	Out StartWindow
	WaitObj	等待跟踪工件超时	124	Out WaitTime
	WaitObj	重复跟踪工件	125	Connected Twice
	开启跟踪后可能发生	跟踪过程超出工作区域抛出异常	126	Out MaxDistance

15.4.9.13 SwitchCase

说明	SwitchCase 指令跟 IF 指令类似，是根据输入的变量条件进行流程控制的指令。

	RL 解释器将根据输入的变量（condition）依次与 Case 字段的变量进行比较。 如果两个变量相等，解释器将进入对应 Case 的代码分支，并且不再进行后续比较，不会进入其他代码分支； 如果所有条件都不满足，则会进入 Default 分支； 如果没有 Case 条件匹配且没有 Default 分支，则不会进入任何分支，Switch 指令结束； Case 指令可以输入多个条件（见指令结构 Case C1,C12,C13 和示例 1）。
定义	Switch(condition) Case C1,C12,C13: Functions1() Case C2: Functions2() Default: DefaultFunction(); EndSwitch

示例	例 1
	reg_int 是一个寄存器变量，上位机（PLC）会通过相关寄存器协议（如 modbus、cclink）更新变量的数值，生产工程希望机器人根据寄存器的数值，执行对应的函数分支（比如走不通的运动轨迹），如果寄存器输入 1,2,3 则执行 A 函数，如果输入 4,5,6 则执行 B 函数，如果不符合上述条件，则进入 Default 分支执行 C 函数。
	Switch(reg_int)
	Case 1,2,3:
	FunctionsA();// 机器人走功能 A 相关点位
	Case 4,5,6:
	FunctionsB();// 机器人走功能 B 相关点位
	Default:
	FunctionC(); // 无指定输入执行 C 功能
	EndSwitch

15.4.10 起始点指令

15.4.10.1 Home

说明	以轴空间运动让机器人回到设定好的初始点
定义	指令无参数
示例	例 1 HomeSet(0,30,0,60,0,90,0); Home; 通过 HomeSet 指令设置初始点，再通过 Home 指令让机器人运动到轴空间的拖拽位姿
注意	必须在机器人设置>快速调整 界面开启 Home 位姿设置或者通过 HomeSet 指令开启 Home 位姿设置，才可以使用 Home 指令，否则 Home 指令会报错。

15.4.10.2 HomeSet

说明	设定机器人的轴空间初始点位置
定义	HomeSet (axis1,axis2,axis3,axis4,axis5,axis6,axis7); Axisx，数据类型：Double，设定在初始点各个轴的角度
示例	例 1

	HomeSet(0,30,0,60,0,90,0); Home; 通过 HomeSet 指令设置初始点，再通过 Home 指令让机器人运动到轴空间的拖拽位姿

15.4.10.3 HomeSetAt

说明	获得机器人的初始点的设定数据
定义	HomeSetAt(index); 返回值，数据类型：double，关节角，单位° Index，数据类型：int，获得初始点指定轴的关节角，当 index 为 0 时，返回是否开启了 Home 设置，1 表示已开启，0 表示未开启。
示例	例 1 HomeSet (0,30,0,60,0,90,0); double angle2 = HomeSetAt(2) angle2 获得关节 2 的关节角 30°。

15.4.10.4 HomeDef

说明	判断是否设置了初始点
定义	HomeDef() 返回值，数据类型：bool，true 已设置初始点，false 未设置初始点

15.4.10.5 HomeSpeed

说明	设定 Home 指令的运行速度
定义	HomeSpeed Speed
示例	例 1 HomeSpeed(v1000); Home(); 设定初始点运动的速度为 V1000，随后 Home 指令让机器人按照 V1000 的速度往初始点运动。

15.4.10.6 HomeClr

说明	清除初始点设置
示例	例 1 HomeClr(); 清除程序中设定的起始点。清除后 Home 指令将无法执行。

15.4.11 数学指令

15.4.11.1 Sin

说明	sin()用来计算参数 x 的正弦值，然后将结果返回。
定义	double sin(double x) x 单位为弧度； 返回值：返回-1 至 1 之间的计算结果。

15.4.11.2 Cos

说明	cos()用来计算参数 x 的余弦值，然后将结果返回
定义	double cos(double x) x 单位为弧度； 返回值：返回-1 至 1 之间的计算结果。

15.4.11.3 Tan

说明	tan()用来计算参数 x 的正切值，然后将结果返回。
定义	double tan(double x) x 单位为弧度； 返回值：返回参数 x 的正切值。

15.4.11.4 Cot

说明	cot()用来计算参数 x 的余切值，然后将结果返回。
定义	double cot(double x) x 单位为弧度； 返回值：返回参数 x 的余切值。

15.4.11.5 Asin

说明	asin()用来计算参数 x 的反正弦值，然后将结果返回。
定义	double asin(double x) x 范围为 - 1 至 1 之间，超过此范围则会报错； 返回值：返回 - PI/2 之 PI/2 之间的计算结果，单位为弧度

15.4.11.6 Acos

说明	acos()用来计算参数 x 的反余弦值，然后将结果返回。
定义	double acos(double x) x 范围为 - 1 至 1 之间，超过此范围则会报错； 返回值：返回 0 至 PI 之间的计算结果，单位为弧度。

15.4.11.7 Atan

说明	atan()用来计算参数 x 的反正切值，然后将结果返回；
定义	double atan(double x) 返回值：返回-PI/2 至 PI/2 之间的计算结果

15.4.11.8 Sinh

说明	sinh()用来计算参数 x 的双曲线正弦值，然后将结果返回，
定义	double sinh(double x) 数学定义式为：(exp(x)-exp(-x))/2； 返回值：返回参数 x 的双曲线正弦值。

15.4.11.9 Cosh

说明	cosh()用来计算参数 x 的双曲线余弦值，然后将结果返回
定义	double cosh(double x) 数学定义式为：(exp(x)+exp(x))/2； 返回值：返回参数 x 的双曲线余弦值。

15.4.11.10 Tanh

说明	tanh()用来计算参数 x 的双曲线正切值，然后将结果返回。
定义	double tanh(double x) 数学定义式为：sinh(x)/cosh(x)； 返回值：返回参数 x 的双曲线正切值。

15.4.11.11 Exp

说明	exp()用来计算以 e 为底的 x 次方值，即 e^x 值，然后将结果返回；
定义	double exp(double x) 返回值：返回 e 的 x 次方计算结果。

15.4.11.12 Ln

说明	ln()用来计算以 e 为底的 x 对数值，然后将结果返回。
定义	double ln(double x) 函数说明：也就是求 x 的自然对数 ln(x)，x > 0； 返回值：返回参数 x 的自然对数值。

15.4.11.13 log10

说明	log10()用来计算以 10 为底的 x 对数值，然后将结果返回。
定义	double log10(double x) 要求 x>0； 返回值：返回参数 x 以 10 为底的对数值。

15.4.11.14 pow

说明	pow()用来计算以 x 为底的 y 次方值，即 xy 值，然后将结果返回；
定义	double pow(double x, double y) 返回值：返回 x 的 y 次方计算结果。

15.4.11.15 sqrt

说明	sqrt()用来计算参数 x 的平方根，然后将结果返回。
定义	double sqrt(double x) 参数 x 必须为正数； 返回值：返回参数 x 的平方根值。

15.4.11.16 ceil

说明	ceil()会返回不小于参数 x 的最小整数值，结果以 double 形态返回；

定义	double ceil(double x) 返回值：返回不小于参数 x 的最小整数值。

15.4.11.17 floor

说明	floor()会返回不大于参数 x 的最大整数值，结果以 double 形态返回；
定义	double floor(double x) 返回值：返回不大于参数 x 的最大整数值。

15.4.11.18 abs

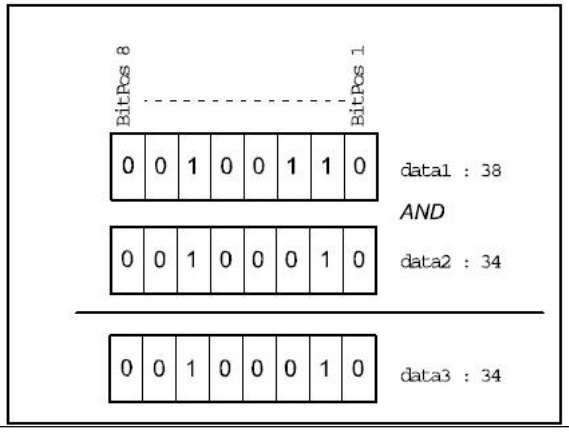
说明	求 x 的绝对值 |x|；
定义	int abs(int x)/double abs(double x) 返回值：当输入参数为 int 型时，输出也为 int 型。当输入参数为 double 型时，输出也为 double 型。

15.4.11.19 rand

说明	产生一个整型随机数；
定义	rand() 返回值：一个整型随机数，范围为 0~2147483647。

15.4.12 位操作

15.4.12.1 BitAnd

说明	BitAnd 用于生成一个逻辑位与的操作，针对 byte 类型数据。如下表： 
定义	返回值，数据类型：byte，表示 2 个 byte 类型数据执行逻辑与返回的结果。 BitAnd (BitData1, BitData2)； BitData1，数据类型：byte，要操作的字节数据 1。 BitData2，数据类型：byte，要操作的字节数据 2。
示例	例 1 VAR byte data1 = 34; VAR byte data2 = 38; VAR byte byte3 = BitAnd(data1, data2); //34 定义 byte 类型变量 data1，赋值 34，定义 byte 类型变量 data2，赋值 38，对 data1 和 data2 执行逻辑与操作，得到 34，赋值给 byte3。

15.4.12.2 BitCheck

说明	BitCheck 用于检查定义的 byte 类型数据的某一位是否为 1，若为 1 则返回 true，否则返回 false。
定义	返回值，数据类型：bool，true 表示指定位为 1，false 表示指定位为 0。 BitCheck (BitData, BitPos)； BitData，数据类型：byte，要操作的字节数据。 BitPos，数据类型：int，要操作位的位置，范围 1~8。
示例	例 1 VAR byte data1 = 130; VAR bool b1 = BitCheck(data1, 8) //true; 定义 byte 类型变量 data1，赋值 130，检测 data1 第 8 位是否为 1，返回 true。

15.4.12.3 BitClear

说明	通过 BitClear 可将 byte 或 int 类型的数据某一位置为 0。位数从 1 开始。
定义	BitClear(BitData | IntData, BitPos); BitData，数据类型：byte，要操作的字节数据。 IntData，数据类型：int，要操作的整型数据。 BitPos，数据类型：int，要操作位的位置，对于 byte 数据来说是（1-8），对于 int 数据来说是 1-32。
示例	例 1 VAR byte data1 = 255; BitClear data1 1 //254; BitClear data1 2 //252; 定义 byte 类型变量 data1，赋值 255，对 data1 进行 BitClear 操作，将第 1 位置为 0，得到 254，将第 2 位置为 0，得到 252。

15.4.12.4 BitLSh

说明	BitLSh 用于对 byte 数据执行逻辑左移操作。
定义	返回值，数据类型：byte，表示执行左移操作得到的 byte 数据。 BitLSh (BitData, ShiftSteps); BitData，数据类型：byte，要操作的字节数据。 ShiftSteps，数据类型：int，要左移的位数，范围 1~8。
示例	例 1 VAR int left_shift = 3; VAR byte data1 = 38; VAR byte data2; data2 = BiLSh(data1, left_shift) //48; 定义 byte 类型变量 data1，赋值 38，对 data1 进行左移 3 位操作，得到 48。

15.4.12.5 BitNeg

说明	BitNeg 用于对 byte 数据执行逻辑非操作。
定义	返回值，数据类型：byte，表示执行逻辑非操作得到的 byte 数据。 BitNeg (BitData)； BitData，数据类型：byte，要操作的字节数据。

示例	例 1 VAR byte data1 = 38; VAR byte data2; data2 = BitNeg(data1) //217; 定义 byte 类型变量 data1，赋值 38，对 data1 执行逻辑非操作，得到 217。

15.4.12.6 BitOr

说明	BitOr 用于对 byte 数据执行逻辑或操作。
定义	返回值，数据类型：byte，表示执行逻辑或操作得到的 byte 数据。 **BitOr (BitData1, BitData2);** **BitData1**，数据类型：byte，要操作的字节数据 1。 **BitData2**，数据类型：byte，要操作的字节数据 2。
示例	例 1 VAR byte data1 = 39; VAR byte data2 = 162; VAR byte data3; data3 = BitOr(data1, data2); //167 定义 byte 类型变量 data1，赋值 39，定义 byte 类型变量 data2，赋值 162，对 data1 和 data2 执行逻辑或操作，得到 167。

15.4.12.7 BitRSh

说明	BitRSh 用于对 byte 数据执行逻辑右移操作。
定义	返回值，数据类型：byte，表示执行右移操作得到的 byte 数据。 **BitLSh (BitData, ShiftSteps);** **BitData**，数据类型：byte，要操作的字节数据。 **ShiftSteps**，数据类型：int，要右移的位数，范围 1~8。
示例	例 1 VAR int right_shift = 3; VAR byte data1 = 38; VAR byte data2; data2 = BiRSh(data1, right_shift); //4 定义 byte 类型变量 data1，赋值 38，对 data1 进行右移 3 位操作，得到 4。

15.4.12.8 BitSet

说明	通过 BitSet 可将 byte 或 int 类型的数据某一位置为 1，位数从 1 开始。
定义	**BitSet (BitData | IntData, BitPos);** **BitData**，数据类型：byte，要操作的字节数据。 **IntData**，数据类型：int，要操作的整型数据。 **BitPos**，数据类型：int，要操作位的位置，对于 byte 数据来说是（1-8），对于 int 数据来说是 1-32。

示例	例 1
	VAR byte data1 = 0;
	BitSet (data1,1); //1
	BitSet (data1,2); //3
	定义 byte 类型变量 data1，赋值 255，对 data1 进行 BitSet 操作，将第 1 位置为 1，得到 1，将第 2

	位置为 1，得到 3。

15.4.12.9 BitXOr

说明	BitXOr 用于对 byte 数据执行逻辑异或操作。
定义	返回值，数据类型：byte，表示执行逻辑或操作得到的 byte 数据。 BitXOr (BitData1, BitData2); BitData1，数据类型：byte，要操作的字节数据 1。 BitData2，数据类型：byte，要操作的字节数据 2。
示例	例 1 VAR byte data1 = 39; VAR byte data2 = 162; VAR byte data3; data3 = BitOr(data1, data2) ;//133 定义 byte 类型变量 data1，赋值 39，定义 byte 类型变量 data2，赋值 162，对 data1 和 data2 执行逻辑异或操作，得到 133。

15.4.13 字符串操作

15.4.13.1 StrFind

说明	StrFind 用于在字符串中查找，从一个特定位置开始查找属于另一个特定字符集合的位置。
定义	返回值，数据类型：int，表示执行查找得到的第一个字符匹配的位置。如果没有找到，返回字符串长度+1。 StrFind (Str ChPos Set [\NotInSet]); Str，数据类型：string，表示要查找的字符串。 ChPos，数据类型：int，表示开始查找的位置，从 1 开始，如果位置越界，报错提示。 Set，数据类型：string，表示要匹配的字符集合。 [\NotInSet]，标识符，标识搜索不在匹配的字符集合内的字符。

示例	例 1
	VAR int found;
	found = StrFind("Robotics", 1, "aeiou"); //2
	从第 1 个字符"R"开始匹配，发现第 2 个字符"o"在字符集合"aeiou"，返回匹配位置 2。
	found = StrFind("Robotics", 1, "aeiou" \NotInSet); //1
	从第 1 个字符"R"开始匹配，发现第 1 个字符"R"不在字符集合"aeiou"，返回匹配位置 1。

15.4.13.2 StrLen

说明	StrLen 用于获取字符串长度
定义	返回值，数据类型：int，表示当前字符串长度，>=0。 StrLen (Str); Str，数据类型：string，表示要计算字符个数的字符串。
示例	例 1 VAR int num; num = StrLen("Robotics"); //8 字符串"Robotics"长度为 8。

15.4.13.3 StrMap

说明	StrMap 用于创造一份 string 备份，它其中所有字符都将按照指定映射关系进行替换。映射字符根据位置一一对应，没有映射的字符保持不变。
定义	返回值，数据类型：string，表示替换得到的字符串。 StrMap (Str, FromMap, ToMap); Str，数据类型：string，表示原字符串。 FromMap，数据类型：string，表示映射的索引部分。 ToMap，数据类型：string，表示映射的值部分。

示例	例 1
	VAR string str;
	str = StrMap("Robotics", "aeiou", "AEIOU") //RObOtIcs;
	对字符串"Robotics"进行字符映射，"aeiou"分别映射成"AEIOU"。
	
	使用限制：FromMap 和 ToMap 必须匹配，长度一致。

15.4.13.4 StrMatch

说明	StrMatch 用于在字符串中搜索，从指定位置开始，搜索特定格式或字符串，返回匹配的位置。
定义	返回值，数据类型： int，表示匹配字符串首字符所在的位置，如果没有匹配，则返回字符串长度+1。 StrMatch (Str, ChPos, Pattern); Str，数据类型： string，表示要搜索的字符串。 ChPos，数据类型： int，表示起始位置，如果超出字符串长度范围报错。 Pattern，数据类型： string，表示要匹配的格式字符串。
示例	例 1 VAR int found; Found = StrMatch("Robotics", 1, "bo") //3; 从第 1 个字符开始搜索"bo"，发现第 3 个位置匹配，返回匹配的首字符所在位置 3。

15.4.13.5 StrMemb

说明	StrMemb 用于检查字符串中某个字符是否属于指定的字符集合。
定义	返回值，数据类型：bool，true 表示字符串中指定位置的字符属于指定的字符集合，否则 false。 StrMemb (Str, ChPos, Set); Str，数据类型：string，表示要检查的字符串。 ChPos，数据类型：int，表示要检查的字符位置，超出字符串范围报错。 Set，数据类型：string，表示要匹配的字符集合。

示例	例 1
	VAR bool memb;
	memb = StrMemb("Robotics", 2, "aeiou") //true;
	第 2 个字符 o 属于字符集合"aeiou"中一员，返回 true。

15.4.13.6 StrOrder

说明	StrOrder 用于比较两个字符串，并且返回布尔值。
定义	返回值，数据类型：bool，当 str1<=str2 时返回 true，否则 false。

	StrOrder (Str1, Str2); Str1，数据类型：string，表示第一个字符串值。 Str2，数据类型：string，表示第二个字符串值。

示例	例 1
	VAR bool le;
	le = StrOrder("FIRST", "SECOND"); //true;
	le = StrOrder("FIRSTB", "FIRST"); //false

15.4.13.7 StrPart

说明	StrPart 用于截取字符串一部分生成一个新的字符串。
定义	返回值，数据类型：string，表示截取得到的字符串，从指定位置开始截取指定长度的字符串。 StrPart (Str, ChPos, Len); Str，数据类型：string，表示要被截取的原字符串。 ChPos，数据类型：int，表示起始截取位置，当超出字符串范围时报错。 Len，数据类型：int，表示要截取的长度。

示例	例 1
	VAR string part;
	part = StrPart("Robotics", 1, 5); //Robot
	从第 1 个位置开始截取长度为 5 的字符串，得到"Robot"。

15.4.13.8 StrSplit

说明	StrSplit 可以对字符串进行分割，通过指定分隔符，将字符串分割成字符串数组
定义	返回值，数据类型：string 数组，表示分割得到的字符串数组 StrSplit (Str [, separator]); Str，数据类型：string，表示要被分割的原字符串。 Separator，数据类型：string，分隔符，该字符串中的所有字符都被看做分隔符，可以缺省，当缺省时，空格作为默认分隔符
示例	string str_arr[4] = StrSplit("test1,test2;test3\test4", "\,;"); 字符串分割结果为四个子串(test1 test2 test3 test4)。 使用限制： ● 输入字符串为空时会报错。 ● 分割结果和定义字符串的长度不匹配时会报错。

15.4.13.9 StrToByte

说明	StrToByte 可以将字符串转换为 byte 类型数据
定义	返回值，数据类型：byte，字符串的转换结果。 StrToByte (Str, [trans]); Str，数据类型：string，将要进行数据转换的字符串。 Trans，数据类型：枚举，表示字符串的数学进制格式，可用参数为 \Bin（二进制），\Okt（八进制），\Hex（十六进制），\Char（字符）以及默认参数（无参数，十进制）
示例	例 1 Byte NumBin = StrToByte("10", \Bin); Byte NumOkt = StrToByte("10", \Okt);

	Byte NumBin = StrToByte("10"); Byte NumHex = StrToByte("10", \Hex); 依次将字符串 "10" 按照二进制、八进制、十进制、16 进制转换为 byte 数字，结果依次为 2, 8, 10, 16。 例 2 Byte NumChar = StrToByte("0", \Char); 将字符"0"按照字符与 ASCII 的转换关系，转为 48。 使用限制：输入字符串不符合指定的数据格式时会报错。

15.4.13.10 StrToDouble

说明	StrToDouble 可以将字符串转换为 double 类型数据
定义	返回值，数据类型：double，字符串的转换结果。 **StrToDouble (Str);** **Str**，数据类型：string，将要进行数据转换的字符串。
示例	例 1: Double NumDouble = StrToDouble("3.1415926"); 将"3.1415926"字符串转换为 double 类型数据。 使用限制： 输入字符串不符合指定的数据格式时会报错。

15.4.13.11 StrToInt

说明	StrToInt 可以将字符串转换为 Int 类型数据
定义	返回值，数据类型：Int，字符串的转换结果。 **StrToDouble (Str);** **Str**，数据类型：string，将要进行数据转换的字符串。
示例	例 1 Int NumInt = StrToInt("99"); 将"99"字符串转换为 Int 类型数据。 使用限制：输入字符串不符合指定的数据格式时会报错。

15.4.13.12 StrToDoubleArray

说明	可以对大量 double 字符串进行类型转换，转换成 double 数组数据
定义	返回值，数据类型：Int，转换是否遇到异常，-1 错误，0 正常。 **StrToDoubleArray(output, input, spilit);** **Output**，数据类型：double 数组，转换结果输出 **Input**，数据类型：str 字符串，输入的字符串 **Spilit**，数据类型: str 字符串，字符串的分隔符
示例	例 1 string tmp_ss = "1,2,3,4,5,6,7"; double db_arr[10]; StrToDoubleArray (db_arr, tmp_ss, ",") ;

	// 结果 db_arr = {1,2,3,4,5,6,7,0,0,0} 使用限制： 字符串允许结尾出现一个多余的结束符 转换失败或含非法字符报错"输入的字符串分割后不是 Double 形式" double 数组应大于等于字符串中数据量的大小，否则报错 "输入的数组大小不足 或者 数组不是一维数组"

15.4.14 运算符

15.4.14.1 基础运算符

15.4.14.1.1 算数运算符

算数运算符包括：

运算符	作用
+	加
-	减/负号
*	乘
/	除
%	取余
--	自减
++	自加

算数运算操作符支持 bool,byte,int,double 类型数据的操作，
不同类型的变量如果放在一起进行加减乘除运算，会触发隐式转换
各种算术运算符用法示例如下：

例 1

```
VAR int a = 1;
VAR int b = 2;
VAR int c = -b;//取负
VAR int ac = a * c; //乘法
```

例 2

++，--两个运算符又称单目运算符，是指对一个操作数进行操作的运算符，RL 不区分前后自加自减：

```
x = n++;          //表示 n 先加 1，再将 n 的值赋给 x
x = --n;          //表示 n 先减 1 后，再将新值赋给 x
```

例 3

不同类型变量的进行加减乘除运算的隐式转化结果

类型 1	类型 2	结果
bool	bool	bool
bool	byte	byte
bool	int	int
bool	double	double

byte	byte	byte
byte	int	int
byte	double	double
int	int	int
int	double	double
double	double	double

15.4.14.1.2 逻辑运算符

逻辑运算符支持基本数据类型的运算，包括：

运算符	作用
&&	逻辑与
||	逻辑或
<	小于
>	大于
<=	小于等于
>=	大于等于
==	等于
!=	不等于
!	取逻辑非

逻辑与&&表达式为真的条件是两边的结果都为真，而逻辑或||为真的条件是两边只要有一个条件为真即可。

例 1

其他逻辑运算符用法示例如下：

```
VAR int res = 1;
while(res < 3)        //比较 res 是否小于 3
res++;
endwhile
di5 = !di6;              //取逻辑非
VAR int counter = 4;
while(di7 && di8)   //求逻辑与
if(counter == 5)       //是否相等
        break;
   endif
endwhile
```

15.4.14.1.3 赋值运算符

赋值运算符包括：

运算符	作用
=	赋值

+=	加等
-=	减等
*=	乘等
/=	除等
%=	取模等

各种赋值运算符用法示例如下

```
VAR int num1 = 3;
VAR int num2 = 4;
num1 += num2;          //等同于 num1 = num1 + num2 则 num1 = 7。
num1 -= num2;          //等同于 num1 = num1 – num2，则 num1 = -1。
num1 *= num2;          //等同于 num1 = num1 * num2 则 num1 = 12。
num1 /= num2;          //等同于 num1 = num1 / num2 则 num1 = 0。
num1 %= num2;          //等同于 num1 = num1 % num2 则 num1 = 3。
```

变量所有赋值操作，都支持隐式转换。当赋值运算左右两边的数据类型不一致时，解释器会尝试触发隐式转换以使得程序能继续运行，当转换失败时，程序会报错停止。
Bool, Byte, Int, Double 四种数据类型之间能进行转换。IO、寄存器变量属于上面四种变量的特殊形式，如果它们用于赋值操作，也可触发隐式转换。
函数的返回值如果属于上面四种类型，也可作为赋值操作的右值，进行赋值计算。

例 1

```
int tmp_num = 10.5;     // 10
bool tmp_bool = 1;   // true
tmp_bool = 0;          // false
double tmp_d = 999;   // 999.0
```

例 2

```
// 直接用寄存器变量修改普通变量
double tmp_num = register0;
// 寄存器变量可以直接用于条件判断
WaitUntil(register0 == 10);
```

例 3

```
int mem_ret = StrMemb("Robotics", 2, "aeiou");
// StrMemb 的返回值是 bool 类型，如果一定要用 int 类型接收返回值，
// 控制器不会报错，而是会进行隐式转换
// true -> 1 ,   false-> 0
```

15.4.14.1.4 其他运算符

运算符	作用
( )	圆括号
.	点操作符

各运算符用法示例如下：

例 1

```
VAR int num = arr[1];        //取数组第一个元素赋给 num
VAR int num2 = (1+2)*3;      //使用括号可以改变运算顺序，这里 num2 的值为 9
```

例 2

定义一个 robtarget 变量 pt1

```
pt1.trans.x = 200;                //使用“.”操作符将 pt1 点的 x 坐标改为 200
```

使用限制：

“.”操作符不支持对 robtarget 变量 A，B，C 成员的修改。

15.4.14.2 运算优先级

优先级	运算符	使用形式	结合方向
1	( )	（表达式）/函数名(形参表)	
	.	变量名.	
2	-	-表达式	右到左
	++	++变量名/变量名++	
	--	--变量名/变量名--	
	!	!表达式	
3	/	表达式/表达式	左到右
	*	表达式*表达式	
	%	整型表达式/整型表达式	
4	+	表达式+表达式	左到右
	-	表达式-表达式	
5	>	表达式>表达式	左到右
	>=	表达式>=表达式	
	<	表达式<表达式	
	<=	表达式<=表达式	
6	==	表达式==表达式	左到右
	!=	表达式!= 表达式	
7	&&	表达式&&表达式	左到右
8	||	表达式||表达式	左到右
9	=	变量=表达式	右到左
	/=	变量/=表达式	
	*=	变量*=表达式	
	%=	变量%=表达式	
	+=	变量+=表达式	
	-=	变量-=表达式	

15.4.15 时钟指令

15.4.15.1 ClkRead

说明	ClkRead 用于读取计时器的值。

定义	返回值，数据类型：double，返回计时器停止时刻或当前时刻距启动 clock 的时间间隔，精度 0.001s。 ClkRead (Clock); Clock，数据类型：clock，计时器名称。
示例	例 1 VAR clock clock1; ClkStart(clock1); ClkStop(clock1); VAR double interval=ClkRead(clock1); interval 存储 clock1 在启动和停止之间的时间间隔。

15.4.15.2 ClkReset

说明	ClkReset 用于重置一个计时器。使用一个计时器前可通过 ClkReset 保证计数为 0。
定义	ClkReset (Clock); Clock，数据类型：clock，计时器名称。
示例	例 1 VAR clock clock1; ClkReset (clock1); 重置 clock1。

15.4.15.3 ClkStart

说明	ClkStart 用于启动一个计时器。 当一个计时器启动后，它会不断的运行计数，直到计时器停止或程序重置。 即使程序停止或机器人下电，计时器仍会继续运行。
定义	ClkStart (Clock); Clock，数据类型：clock，计时器名称。
示例	例 1 VAR clock clock1; ClkStart (clock1); 声明 clock1，启动计时器 clock1。

15.4.15.4 ClkStop

说明	ClkStop 用于停止一个计时器。 当计时器停止后，它会停止计数。计时器停止后，可以被读取间隔，重新启动或重置。
定义	ClkStop (Clock); Clock，数据类型：clock，计时器名称。
示例	例 1 VAR clock clock1; ClkStart (clock1); … ClkStop (clock1); 停止计时器 clock1。

15.4.16 高级指令

15.4.16.1 RelTool

说明	在当前指令指定的工具坐标系下对空间位置进行平移或者旋转。 与 Offs 主要区别：Offs 是相对于工件坐标系偏移，RelTool 是相对于工具坐标系偏移。
定义	返回值，数据类型： robtarget，返回偏移后的新位姿。 RelTool(Point, XOffset, YOffset, ZOffset, Rx, Ry, Rz [, Tool, Wobj]); Point，数据类型： robtarget，待偏移的位置点，或者说偏移指令的初始点。 XOffset，数据类型： double，沿工具坐标系 x 方向上的偏移量。 YOffset，数据类型： double，沿工具坐标系 y 方向上的偏移量。 ZOffset，数据类型： double，沿工具坐标系 z 方向上的偏移量。 Rx，数据类型： double，绕工具坐标系 x 轴的转动角度。 Ry，数据类型： double，绕工具坐标系 y 轴的转动角度。 Rz，数据类型： double，绕工具坐标系 z 轴的转动角度。 Tool，数据类型： tool 工具，包含描述 Point 点位的工具坐标系信息。 Wobj，数据类型： wobj 工件，包含描述 Point 点位的工件坐标系信息。

示例	例 1
	p2=RelTool(p1,100,0,30,20,0,0);
	由于未指定工具工件，默认使用 tool0,wobj0 作为工具工件，将 p1 点沿工具坐标系的 x 方向偏移 100 mm， y 方向偏移 0 mm， z 方向偏移 30 mm，绕 x 轴旋转 20 度后，将新的目标点位置赋给 p2 点。
	
	例 2
	p2=(RelTool(p1,100,0,30,20,0,0), tool5, wobj6);
	将 p1 点沿 tool5 工具坐标系的 x 方向偏移 100 mm，y 方向偏移 0 mm，z 方向偏移 30 mm，绕 x 轴旋转 20 度后，将新的目标点位置赋给 p2 点。
	
	例 3
	MoveL (RelTool(p1, 100,0,30,20,0,0), v4000, fine, tool2, wobj4);
	RelTool 和 Move 指令配合使用，未指定特定的工具工件坐标系，将使用运动指令的 tool 和 wobj，将 p1 点沿 tool2 工具坐标系的 x 方向偏移 100 mm，y 方向偏移 0 mm，z 方向偏移 30 mm，绕 x 轴旋转 20 度。

注意	该指令的可选参数 Tool 和 Wobj 暂时不支持辅助编程。 xMate CR 系列的 5 轴机型，由于一个姿态自由度的缺失，在使用 RelTool 命令时会有较多点位不可达情况，需要结合可达姿态特点确定偏移（如在法兰平行基座的情况下点位的平移通常都可达）。

15.4.16.2 Offs

说明	位置偏移函数，用于把某个点在当前指令中指定的工件坐标系下偏移一段距离并返回新点的位置值，平移偏移量由 x、 y、 z 三个量来表示，姿态旋转偏移量由 Rx、 Ry、 Rz 表示。
定义	返回值，数据类型： robtarget，偏移后的新位姿。 Offs（Point， XOffset， YOffset， ZOffset [, Rx, Ry, Rz] ） Point，数据类型： robtarget，待偏移的位置点，或者说偏移指令的初始点。 XOffset，数据类型： double，沿工件坐标系 x 方向上的偏移量。 YOffset，数据类型： double，沿工件坐标系 y 方向上的偏移量。 ZOffset，数据类型： double，沿工件坐标系 z 方向上的偏移量。 Rx，数据类型： double，绕工件坐标系 x 轴的转动角度。 Ry，数据类型： double，绕工件坐标系 y 轴的转动角度。

	Rz，数据类型： double，绕工件坐标系 z 轴的转动角度
示例	例 1 p11=Offs(p10,100,200,300); 将 p10 点沿工件坐标系的 x 方向偏移 100 mm， y 方向偏移 200 mm， z 方向偏移 300 mm，并将新的目标点位置赋给 p11 点。
注意	该指令暂时不支持辅助编程。

15.4.16.3 ConfL

说明	该指令开启后进行两个方面的检查： 1. 笛卡尔路径实际到达的轴角度（即 cf1~7）是否与目标点示教的相差过大，如果相差过大，日志会显示对应的提示，用户可考虑是否按照这样的运动方式到达目标点。 2. 笛卡尔路径实际到达的轴关节配置（即 cfx 值）是否与目标点示教的一致，如果不一致会报错并停止运动；

示例	例 1
	ConfL(on);;
	MoveL (p1, v1000 ....);
	//如果 P1 示教位置的 J6 角度为-5°，则 MoveL 指令运行后 J6 角度可能为 355°（两者呈现+-360°的差距，即对应的是同一个位置），此时会日志提示"回转角实际到达点与设定点差别过大"，但不停止运动；
	...
	ConfL(off);
	MoveL (p1, v1000 ....);
	//关闭笛卡尔路径的 conf 检查，机器人能够运动到 p1 点，但不针对点位实际运动到达与点位示教时的 confdata 的 cf1~7 值和 cfx 值进行检查。

注意	1. ConfL 指令对应 MoveL、MoveC 等笛卡尔运动指令，不对 MoveJ、MoveAbsj 指令产生影响，也不会对"点位运动至"的 conf 设定产生影响 2. 在执行直线、圆弧等笛卡尔运动指令的情况下，机器人在选取离开始点的姿势最接近姿势的同时向目标点方向移动。此时，目标点的角度将被自动选定，所以在目标点位置的机器人实际回转角度，在某些情况下会与所示教的位置的角度不同。该指令开启后，对这种情况会进行提示。

15.4.16.4 ConfJ

说明	笛卡尔坐标对应一组 conf 参数(cf1~7, cfx)，用户手动更改或写入的笛卡尔坐标点对应的 conf 数据可能是错误的，导致控制器无法解析出目标点的路径，但是存在部分场景，用户只关心机器人 TCP 点的位置而不关心姿态，此时可以使用 ConfJ Off 解除点位求解的 conf 限制，让控制器尝试计算出离路径起点最（较）近的逆解（可能算不出来，导致运动指令失败）。 可以阅读 confdata 的介绍，了解更多 conf 相关的知识。

示例	例 1
	p1.trans.x = ...;
	MoveJ (p1, v1000 ....);
	//只修改了坐标，未修改 confdata 参数，该指令很可能导致执行失败。
	...
	ConfJ(Off);
	MoveJ (p1, v1000 ....);
	//关闭 conf 检查，机器人能够运动到 p1 点，但是轴的姿态不确定。

注意	ConfJ 指令对应 MoveJ 指令，不对其他运动指令产生影响，也不会对"点位运动至"的 conf 设定产生影响。

15.4.16.5 Conf

说明	ConfL 和 ConfJ 的统一开启、关闭指令，具体效果为： Conf on 等效：ConfL on 和 ConfJ on Conf off 等效：ConfJ off 和 ConfJ off
示例	例 1 Conf(on); MoveL (p0, v1000 ...); MoveJ (p1, v1000 ....); //等效为下面的指令： ConfL (on); ConfJ (on); MoveL (p0, v1000 ...); MoveJ (p1, v1000 ....);

15.4.16.6 VelSet

说明	当机器人搬运易碎物品时，可使用 VelSet 指令调节最大运动速度以达到更加平顺的运动效果。机器人在实际运动路径下，每个关节的最大速度能力往往并非恒定值，而是会随着负载状态、本体姿态等因素始终处于变化中，VelSet 指令是对特定任务路径下的最大速度能力曲线进行的缩放，缩放后的最大速度能力曲线同样也是一条变化的曲线。
定义	VelSet (gain); gain，数据类型：int，按照百分比来指定最大速度能力的大小，取值范围 1%~100%，其中 100%对应最大加速度，超过限制范围机器人会报错。
示例	例 1 VelSet (50); 将最大速度能力设置为机器人默认最大速度的一半。
注意	1. VelSet 指令只影响所在的 RL 工程的运动指令，对非工程的 JOG 功能、运动至功能、快速移动功能等均不生效。 2. VelSet 功能会打断转弯区，请勿在需要转弯区的运动指令之间插入 VelSet 指令。 3. VelSet 指令和程序运行速率调整滑块的功能区别：程序运行速率调整滑块修改的是用户期望速度，如 V4000 的运动指令，在 50%的滑块控制下，实际相当于 V2000 的用户期望速度，但假如此时机器人处于极限工况，此运动指令能够达到的实际最大速度只有 V1000，那么不管速度滑块处于 50%还是 100%的时候，机器人实际的运动速度是不会改变的，因为不论是 V2000 还是 V4000，都超过了 V1000，所以在这个区间范围内改变期望速度，对实际执行速度不会造成影响；相反，VelSet 50 并不改变用户期望速度，而是在运动规划的过程中，将该运动指令能达到的实际最大速度缩小了 50%，同样以此运动指令为例，实际机器人的运动速度将会缩小一半，从原本的 V1000 变为 V500。这两个功能的区别需要用户在使用的过程中能够加以区分。 4. 发生以下操作时，速度自动恢复为默认大小（100%）： ● RL 程序被手动重置时（PP to Main） ● 加载新的 RL 程序时

15.4.16.7 AccSet

说明	当机器人搬运易碎物品时，可使用 AccSet 指令调节加速度以达到更加平顺的运动效果。
定义	AccSet (acc, ramp); Acc，数据类型：int，按照系统预设值的百分比大小指定加速度的大小，取值范围 30%~100%，其中 100%对应最大加速度，超过限制范围机器人会报错。 Ramp，数据类型：int，按照系统预设值的百分比大小指定加加速度（Jerk），取值范围 30%~100%，其中 100%对应最大加加速度，超过限制范围机器人会报错。
示例	例 1 AccSet (50,50); 加速度、加加速度设置为默认的一半。
注意	发生以下操作时，加速度自动回复为默认大小（100%）： ● RL 程序被手动重置时（PP to Main） ● 加载新的 RL 程序时

15.4.16.8 MotionSup

说明	用于打开、关闭碰撞检测功能
定义	MotionSup(type [, level, event]); Type，数据类型：关键字，On 打开，Off 关闭 Level，数据类型：int，MotionSup On 的额外参数，修改碰撞检测灵敏度百分比，范围：[1,200] Event，数据类型：string，MotionSup On 的额外参数，设置碰撞后的行为 ● "softstop" 柔顺停止
示例	例 1 MotionSup(On); //... 其他指令 MotionSup(Off); 开启碰撞检测后执行其他指令，执行完成后使用 MotionSup Off 关闭碰撞检测 例 2 MotionSup(On, 200, "softstop"); 开启碰撞检测，设置检测灵敏度百分比为 200%，并设置检测到碰撞后触发的行为为柔顺停止。

15.4.16.9 MotionSupPlus

说明	MotionSupPlus（Motion Supervision Plus）运动监控用于在 RL 程序中随时调整机器人各关节碰撞检测灵敏度参数。
定义	MotionSupPlus(x1,x2,x3,x4,x5,x6,x7); x1 至 x7，分别表示关节 1 到关节 7 碰撞检测灵敏度，单位%。
示例	例 1 MotionSupPlus(5,20,7,20,6,20,5); 表示 7 个关节灵敏度分别为 5，20，7，20，6，20，5。 注意： 当用于六轴机器人，也需要设置 7 个参数，其中前 6 个参数起作用，对应 1~6 关节； 该指令支持协作机器人和六轴工业机器人，不支持三、四轴工业机器人。

15.4.16.10 MotionSupJointTrq

说明	MotionSupJointTrq（Motion Supervision Joint Torque）运动监控用于在 RL 程序中随时调整机器

	人各关节传动力限制参数。
定义	MotionSupJointTrq(x1,x2,x3,x4,x5,x6,x7); x1 至 x7，分别表示关节 1 到关节 7 传动力矩限制参数，单位 Nm。
示例	例 1 MotionSupJointTrq (100,91,59,14,14,14,5); 表示 7 个关节传动力矩限制参数分别为 100Nm, 91Nm, 59Nm, 14Nm, 14Nm, 14Nm, 5Nm。 注意： 当用于六轴机器人，也需要设置 7 个参数，其中前 6 个参数起作用，对应 1~6 关节； 该指令支持协作机器人和六轴工业机器人，不支持三、四轴工业机器人。

15.4.16.11 BreakLookAhead

说明	该指令通知控制系统取消前瞻，强制取消上一条运动指令和下一条运动指令之间的转弯区。机器人 TCP 将运行至上一条运动指令的目标点后，再向下一个点位运动，不衔接转弯区。程序指针也将等待 TCP 运行至上一条运动指令的目标点后，再继续往下前瞻扫描。
定义	指令无参数和返回值
示例	例 1 MoveL(P1,v1000,z50,tool0); BreakLookAhead MoveL(P2,v1000,z50,tool0); MoveL(P3,v1000,z50,tool0); 1）点位 P1 的转弯区设置为 z50 ，由于 BreakLookAhead 指令存在，此处前瞻将取消，转弯区也将取消，机器人 TCP 会精确运动至 P1 点，再向 P2 运动。而 P2 和 P3 之间没有 BreakLookAhead 指令，因此在 P2 点会前瞻并且保留 z50 转弯区向 P3 运动。 2）BreakLookAhead 指令与 wait 0 指令效果完全相同。

15.4.16.12 GetRobotMaxLoad

说明	获取当前型号机器人的最大负载值
定义	Ret = GetRobotMaxLoad(); Ret，返回值，数据类型：int，最大负载

示例	int maxload = GetRobotMaxLoad();
	print(maxload);
	以 xMate 7 为例，返回值为 7。

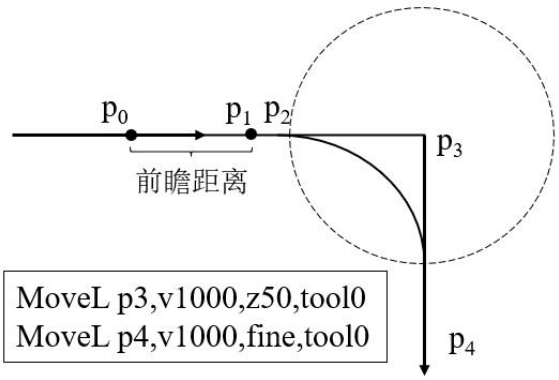
15.4.16.13 GetRobotState

说明	获取控制系统当前的运行状态。用 4 个字节的 bit 信息来表示控制系统的状态，包括故障、紧急停止、安全门、操作模式、伺服模式、运动状态等信息。具体如下表
定义	Ret = GetRobotState(); Ret，返回值，数据类型：byte 数组，用四个 byte 类型来表示机器人状态。

示例	例 1
	byte st[4] = GetRobotState();
	print(st);
	返回{0,5,0,0}，查表可知当前状态为：无故障，电机已上电，自动模式，机器人运动状态，伺服处于位置模式

序号	状态位	含义
1	Byte[1].bit[1]	1：控制系统未授权
2	Byte[1].bit[2]	1：控制系统可恢复故障
3	Byte[1].bit[3]	1：控制系统致命错误
4	Byte[1].bit[4]	1：伺服系统故障
5	Byte[1].bit[5]	1：伺服系统致命故障
6	Byte[1].bit[6]	1：紧急停止
7	Byte[1].bit[7]	1：安全门停止
8	Byte[1].bit[8]	保留
9	Byte[2].bit[1]	上电状态 ， 0 ：电机未上电；1 ：电机已上电
10	Byte[2].bit[2]	机器人运动状态 ， 0：空闲；1：运动中
11	Byte[2].bit[3]	操作模式， 0：手动模式；1：自动模式
12	Byte[2].bit[4]	伺服模式， 0：位置模式；1：力矩模式
13	Byte[2].bit[5]	保留
14	Byte[2].bit[6]	保留
15	Byte[2].bit[7]	保留
16	Byte[2].bit[8]	保留
17	Byte[3]	保留
18	Byte[4]	保留

15.4.16.14 AutoIgnoreZone

说明	用于指定是否允许控制系统自动忽略转弯区
定义	AutoIgnoreZone (true/false); true：允许控制系统自动忽略转弯区（这也是控制系统地默认状态）； false：不允许控制系统自动忽略转弯区；  如上图所示：机器人运行两条 MoveL 指令，中间含有 z50 的转弯区。机器人在运动的过程中，为了运动的柔顺性和安全性，需要从当前位置，向前前瞻一段距离。例如，机器人运动到 P0 点时，前瞻到了 P1 点。在此过程中，控制系统会对 p0 到 p1 点间的信息进行预处理。 机器人向前运动的过程中，前瞻终点也在不断向前移动，在某一时刻，前瞻终点 p1 和转弯区起点 p2 重合。此时，如果控制系统已经收到了第二条运动指令，便可正常生成转弯区，控制机器人沿着预定的轨迹运动；如果控制系统未能收到第二条运动指令，则无法生成转弯区，此时控制系统会依据 AutoIgnoreZone 指令状态对转弯区进行处理。具体处理逻辑如下： AutoIgnoreZone true：控制系统不会等待第二条运动指令，而是将转弯区取消，直接向着 P3 点运动。

	AutoIgnoreZone false：控制系统会等待第二条运动指令，在此过程中，机器人会降低运动速度，直至生成转弯区轨迹。若机器人运动到 p2 点时，仍未收到第二条运动指令，则机器人会停止运动，并通过 HMI 报错。
	机器人未能及时收到第二条运动指令，往往在于两条运动指令间加入了过多的非运动指令，例如：
	MoveL p3,v1000,z50,tool0
	For(int i from 1 to 10000)
	  printf("i = %d\n", i)
	endfor
	MoveL p4,v1000,fine,tool0
	两条运动指令间加入了非常多的打印指令，控制系统处理完第一条运动指令后，经过很长时间才能收到第二条运动指令。

示例	例 1 AutoIgnoreZone(true); MoveL(p3,v1000,z50,tool0); MoveL(p4,v1000,fine,tool0); 允许控制系统自动忽略转弯区 例 2 AutoIgnoreZone(false); MoveL(p3,v1000,z50,tool0); MoveL(p4,v1000,fine,tool0); 不允许控制系统自动忽略转弯区

15.4.16.15 MotionWaitAtFinePoint true/false

说明	机器人静止状态时，点击启动，控制系统会依据前瞻参数，前瞻一段距离后再启动机器人。该指令用于设定，当前瞻遇到 fine 点时，机器人是否立即启动运动。 Fine 点:指的是没有衔接转弯区的目标点，即转弯区参数写作 fine 的目标点。
定义	MotionWaitAtFinePoint(true/false); MotionWaitAtFinePoint true：控制系统严格按照前瞻参数，控制机器人的启动，当前瞻距离达到前瞻参数设定值或前瞻完所有运动指令时，机器人才启动运动。此状态下，控制系统能够保证设定的前瞻距离长度。 MotionWaitAtFinePoint false：控制系统不严格遵守前瞻参数，当前瞻遇到 fine 点时，机器人立即启动运动。此状态下，遇到非常复杂的程序逻辑，机器人依然能够顺利启动，但是无法保证前瞻距离的长度。 控制系统默认状态：MotionWaitAtFinePoint false
示例	例 1 MotionWaitAtFinePoint(true); MoveL(p1,v1000,fine,tool0); MoveL(p2,v1000,fine,tool0); MoveL(p13v1000,fine,tool0); MoveL(p1,41000,fine,tool0); MoveL(p1,51000,fine,tool0); 控制系统前瞻到 p1 点时，不会立即启动机器人，而是检查当前前瞻距离是否达到了前瞻参数设定长度，从而决定是否启动机器人。

	例 2 MotionWaitAtFinePoint(false); MoveL(p1,v1000,fine,tool0); MoveL(p2,v1000,fine,tool0); MoveL (p3,v1000,fine,tool0); MoveL (p4,v1000,fine,tool0); MoveL (p5,v1000,fine,tool0); 控制系统前瞻到 p1 点时，不会检查当前前瞻距离是否达到了前瞻参数设定长度，而是立即启动机器人。

15.4.16.16 IgnoreOverride

说明	焊接、涂胶等对沿路径的运动速度有严格要求的场景，希望工艺段的运动速度不受全局速率影响。可以通过该指令，暂时屏蔽速率滑块对运动指令的影响，使指定某段运动指令、轨迹不受全局速率影响。 该指令支持在自动模式和手动模式下使用。手动模式下，运动速度受 v250 限制，速度超过 v250，按照 v250 执行。自动模式下则按期望速度运动。
定义	IgnoreOverride(On/Off); IgnoreOverride On 表示其后运动指令不受速率滑块影响； IgnoreOverride Off 效果与其相反。

示例	例 1
	MoveJ (p1, v1000 ....);//受滑块速率影响
	IgnoreOverride(On);
	MoveJ (p2, v1000 ....);//不受滑块速率影响
	MoveJ (p3, v1000 ....);//不受滑块速率影响
	IgnoreOverride(Off);
	MoveJ (p4, v1000 ....);//受滑块速率影响
	
	例 2（手动模式）
	IgnoreOverride(On);
	MoveJ (p2, v1000 ....);//不受滑块速率影响，速度按 v250 执行
	MoveJ (p3, v100 ....);//不受滑块速率影响，速度按 v100 执行
	IgnoreOverride(Off);
	
	例 3（自动模式）
	IgnoreOverride(On);
	MoveJ (p2, v1000 ....);//不受滑块速率影响，速度按 v1000 执行
	MoveJ (p3, v100 ....);//不受滑块速率影响，速度按 v100 执行
	IgnoreOverride(Off);

注意	受影响的运动指令：MoveAbsJ、MoveJ、MoveL、MoveC、MoveCF、MoveT、SearchL、SearchC、TrigL、TrigC、TrigJ。 该指令未立即执行指令，不打断转弯区。 只能在运动任务中使用且不能在 Inzone 中使用，否则报错。

15.4.16.17 SingAreaLockAxis4

说明	该指令表示采用锁定 4 轴的方式，规避机器人手腕奇异点

定义	SingAreaLockAxis4(on/off); On,表示开启锁定 4 轴规避腕部奇异功能。需要注意的是只有当机器人已经 4 轴处于 0 度或者±180 度时才能开启该功能。通常 SingAreaLockAxis4 on 前一句运动指令的目标点位，需要保证 4 轴处于上述点位，或者机器人四轴已经处于上述角度才能保证程序正常运行，否则将会引发报错提示。 Off,表示关闭该功能。 注意：SingAreaLockAxis4 on 和 SingAreaLockAxis4 off 之间的笛卡尔运动指令，其姿态将采用特殊的插补方式，不会改变 4 轴的运动角度。任何试图改变 4 轴角度的运动指令都会引起报错。同时该指令被设计为阻塞指令，会打断 SingAreaLockAxis4 前后运动指令间的转弯区。 目前版本适用机型为：工业标准六轴系列(XB,NB 型号)和协作 xMateCR（5 轴机型除外），xMateSR 系列。 开启全 DH 补偿后，不支持该指令。
示例	例 1 MoveAbsJ(p1,v1000,z50,tool0); SingAreaLockAxis4(on); MoveL(p2,v1000, z50,tool0); MoveL(p3,v1000, z50,tool0); SingAreaLockAxis4(off); MoveL(p5,v1000, z50,tool0); 点位 p1 需要保证 4 轴角度为 0 度或者±180 度，运行到 SingAreaLockAxis4 on 锁定四轴进行腕奇异规避功能开启，MoveL p2，MoveL p3 姿态将进行特殊的插补，保持 4 轴角度不变。SingAreaLockAxis4 off 关闭该功能。

15.4.16.18 SpeedRefresh

说明	用于覆盖当前运动程序任务中的速度值。
定义	SpeedRefresh(override); override，数据类型：int，取值范围：[1%, 100%]，速度覆盖值超过限值范围机器人会报错。
示例	例 1 SpeedRefresh(70); 将当前的速度覆盖值设置为 70%
注意	1、 执行 pptomain、pptofunc、to 光标、重新加载工程、进入或退出 demo 时，通过 SpeedRefresh 指令设置的速度值会被清除，并恢复为示教器界面上设定的程序运行速度。 2、 优先级说明：如果通过 SpeedRefresh 设置了速度值，则以该指令设置的速度为准；如果在 SpeedRefresh 指令执行后，手动调整了机器人运行速度（例如拖动示教器上的速度滑条），则之后将以手动调整的速度为准。 3、 SpeedRefres 指令在转弯区生效。 4、 SpeedRefresh 指令可以设置超过手动模式下的程序速度上限、自动模式下的初始程序速度上限速度值，请谨慎使用。（在自动模式下运行 RL 程序时，如果点击暂停并切换到手动模式，再点击"运行"按钮继续执行，此时程序运行速度将取"手动模式下的程序速度上限"和"SpeedRefresh 设置的速度值"两者中的较小值） 5、 通过 SpeedRefresh 设置的速度覆盖值不会立即完成，下达命令与物理机械臂产生速度影响之间会存在一定时间滞后。

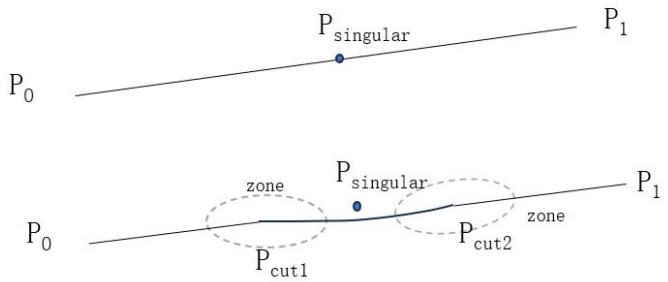
15.4.16.19 CSpeedOverride

说明	用户读取当前速度覆盖值

定义	Ret = CSpeedOverride(); Ret，数据类型： int，取值范围 1~100%速度覆盖值

示例	例 1
	int override = CSpeedOverride();
	print(override);
	若当前速度覆盖值为 70%，则返回 70

15.4.16.20 SingAreaJointWay

说明	该指令表示采用关节空间轨迹插补的方式，规避笛卡尔指令中的奇异点。 SingAreaJointWay on 和 SingAreaJointWay off 之间的笛卡尔运动指令，控制系统会自动检测其上是否有奇异点。该轨迹如果不包含奇异点，将按照普通轨迹的方式进行运动，如果包含奇异点，将按照该模式特有的模式进行运动。具体如下所示：  如上图所示，对于含有奇异点的笛卡尔轨迹 P0P1，控制系统检测到奇异点 Psingular 后，围绕奇异点 Psingular，在原轨迹上加入 Pcut1 和 Pcut2 两点，将原轨迹划分为 P0Pcut1, Pcut1Pcut2, Pcut2P1 三部分，其中 P0Pcut1 和 Pcut2P1,依然为原笛卡尔轨迹，Pcut1Pcut2 采用关节空间轨迹(MoveAbsJ)代替原轨迹部分，从而能够通过奇异点。三段轨迹间采用转弯区平滑过渡，该转弯区的转弯半径可以进行设置，即为指令中的 zone 参数。 机器人在奇异点附近的运动，关节角度通常运动幅度很大，所以使用该指令时，需要确认机器人的运动轨迹是否符合要求。 注意：目前版本适用机型为：工业标准六轴系列(XB,NB 型号)。
定义	SingAreaJointWay(on/off,zone); on/off ,分别代表关节插补奇异功能的开启和关闭 Zone,表示切割后 P0Pcut1, Pcut1Pcut2, Pcut2P1 三段轨迹间的转弯区半径，如上图，参考转弯区定义。
示例	例 1 MoveAbsJ(p1,v1000,z50,tool0); SingAreaJointWay(on,50); MoveL(p2,v1000, z50,tool0); MoveL(p3,v1000, z50,tool0); SingAreaJointWay(off); MoveL(p5,v1000, z50,tool0); 上述指令中 SingAreaJointWay on,50 开启奇异规避，并指定奇异规避内部转弯区半径为 50mm，SingAreaJointWay off 关闭奇异规避，两者之间的运动指令将采用关节插补奇异规避的方式进行运动。

15.4.16.21 SingAreaWrist

说明	该指令表示采用牺牲姿态的方式，规避笛卡尔指令中的奇异点。

	SingAreaWrist on 和 SingAreaWrist off 之间的笛卡尔运动指令，都使用牺牲姿态的方式来运动。在这种情况下，机器人工具遵循正确的，精准的轨迹运动，但是机器人腕关节形态将被改变。未通过奇异点时，亦将出现上述情况。 机器人使用牺牲姿态的方式来运动，机器人腕关节姿态可能运动幅度很大，所以使用该指令时，需要确认机器人的运动轨迹是否符合要求。 注意： 目前版本适用机型包括工业标准六轴系列(XB,NB 型号)和协作 xMateCR, xMateSR 系列。 开启全 DH 补偿后，不支持该指令。
定义	SingAreaWrist(on/off, limit); on/off，分别代表牺牲姿态规避奇异点功能开启和关闭。 Limit，表示设置的允许最大的可牺牲姿态的值，单位角度。
示例	例 1 MoveAbsJ (p1,v1000,z50,tool0); SingAreawrist (on,30); MoveL (p2,v1000, z50,tool0); MoveL (p3,v1000, z50,tool0); SingAreaWrist off; MoveL (p4,v1000, z50,tool0); 上述指令中 SingAreaWrist on,30 开启奇异规避，并指定奇异规避最大可牺牲姿态为 30 度，SingAreaWrist off 关闭奇异规避，两者之间的运动指令将采用奇异规避的方式进行运动。需注意，p1 与 p2, p2 与 p3 之间的转弯区可正常生成，p3 和 p4 之间的转弯区不可生成。
注意	只可以对直线运动使用,不可对弧线运动使用。 使用牺牲姿态奇异规避功能，示教点在奇异点范围内时，示教点将被变更。运动时的手腕姿态与示教的可能有所不同，不仅通过奇异点的示教点，而且以后的示教点的姿态也可能被改变。 单步运动时和连续运动时的奇异点附近姿态的运动可能不同。

15.4.16.22 SetRobotJointsMaxAcc

说明	用于动态修改机器人轴的最大加速度值，当需要机器人放大轴的最大加速度值，提升节拍，增加机器人运行速度时，可使用该指令对每个轴设置最大加速度。指令的效果与通过 HMI 设置运动参数下的"最大轴加速度"是一致的，该指令可以针对特定的运动指令设置该效果。
定义	SetRobotJointsMaxAcc(jointval1, jointval2, jointval3, jointval4, jointval5, jointval6, jointval7); jointval1~7，数据类型： double/int，对应轴的最大加速度值，单位（°/s^2）；
示例	例 1 SetRobotJointsMaxAcc (80000.0, 70000.0, 70000.0, 150000.0, 150000.0, 20000.0, 20000.0);
注意	当两段估计存在转弯区时，转弯区按照最大轴加速度小的约束生成；

15.4.16.23 SetRobotJointsMaxJerk

说明	用于动态修改机器人轴的最大加加速度值，当需要机器人放大轴的最大加加速度值，提升节拍，增加机器人运行速度时，可使用该指令对每个轴设置最大加加速度。指令的效果与通过 HMI 设置运动参数下的"最大轴加加速度"是一致的，该指令可以针对特定的运动指令设置该效果。
定义	SetRobotJointsMaxJerk(jointval1, jointval2, jointval3, jointval4, jointval5, jointval6, jointval7); jointval1~7，数据类型： double/int，对应轴的最大加加速度值，单位（°/s^3）；
示例	例 1 SetRobotJointsMaxJerk (30000.0, 27000.0, 27000.0, 50000.0, 40000.0, 60000.0, 60000.0);

注意	当两段轨迹存在转弯区时，转弯区按照最大轴加加速度小的约束生成；

15.4.16.24 ResetJointKineLimit

说明	用于恢复轴最大速度、最大加速度为原来的值，配合 SetRobotJointsMaxAcc、SetRobotJointsMaxJerk 指令使用；生效后以上指令的效果被清除；
定义	ResetJointKineLimit(); 无参数。
示例	例 1 ResetJointKineLimit();

15.4.16.25 SetTransmissionOverloadParams

说明	用于动态修改机器人的传动过载系数，当需要机器人适当放大轴的传动过载系数，提升节拍，增加机器人运行速度时，可使用该指令对每个轴设置传动过载系数。指令的效果与通过 HMI 设置本体参数下的过载系数中的"传动过载系数"是一致的，但可以针对特定的运动指令设置该效果。
定义	SetTransmissionOverloadParams (jointval1, jointval2, jointval3, jointval4, jointval5, jointval6, jointval7); jointval1~7，数据类型： double/int，对应轴的传动过载系数。
示例	例 1 SetTransmissionOverloadParams (0.95, 0.95, 0.95, 0.95, 1.5, 0.95, 1.0);
注意	当两段轨迹存在转弯区时，转弯区按照传动过载系数小的约束生成；

15.4.16.26 ResetTransmissionOverloadParams

说明	用于恢复机器人的传动过载系数为原来的值，配合 SetTransmissionOverloadParams 指令使用；生效后该指令的效果被清除；
定义	ReSetTransmissionOverloadParams(); 无参数。
示例	例 1 ReSetTransmissionOverloadParams();

15.4.16.27 SetAccRampTime

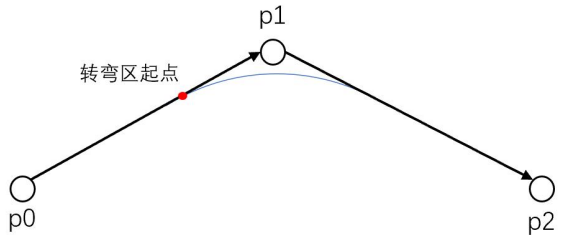
说明	用于设置加速度上升时间，机器人加速度从最小值增加到最大值所消耗的时间。数值越小，机器人加速越快；数值越大，机器人加速越平缓。
定义	SetAccRampTime (ramptime); ramptime, 数据类型： double/int，加速度上升时间值，范围为[0.02, 0.5]，单位 s；
示例	例 1 SetAccRampTime (0.15);

15.4.16.28 ResetAccRampTime

说明	用于恢复默认的加速度上升时间，配合 SetAccRampTime 指令使用；
定义	ResetAccRampTime (); 无参数；
示例	例 1 ResetAccRampTime();

15.4.16.29 SetVarValue

说明	实现在转弯区进行变量赋值，并且前瞻时不触发赋值操作。
定义	SetVarValue(var1, var2); 将数据源 var2 赋值给被赋值变量 var1。 var1，可选数据类型：byte、int、double、bool、未绑定功能码的可写寄存器变量。 var2，可选数据类型：byte、int、double、bool、IO 变量、寄存器变量、函数返回值、表达式。

示例	例 1
	 
	
	// 机器人当前位于 p0 位置
	MoveL(p1);
	SetVarValue(var1, var2); // 在转弯区起点将变量 var2 赋值给变量 var1
	MoveL(p2);

注意	赋值过程中会发生隐式转换，如：
	int a=1;
	double b=20.22;
	SetVarValue(a, b);
	执行赋值操作后，变量 a 的值为 20。

15.4.16.30 SetStopAccRampTime

说明	用于设置到位停止阶段加速度上升时间，即针对无转弯区目标点，机器人停止阶段的加速度从最小值增加到 0 的时间。数值越小，机器人停止越快；数值越大，机器人停止越平缓。若加速度上升时间大于到位停止阶段加速度上升时间，按照较大的加速度上升时间停止。
定义	SetStopAccRampTime(ramptime); ramptime, 数据类型： double/int，到位停止阶段加速度上升时间值，范围为[0.01, 1]，单位 s；
示例	例 1 SetAccRampTime(0.15);

15.4.16.31 ResetStopAccRampTime

说明	用于恢复默认的到位停止阶段加速度上升时间，配合 SetStopAccRampTime 指令使用；
定义	ResetStopAccRampTime(); 无参数；
示例	例 1 ResetStopAccRampTime();

15.4.16.32 MotionSupJointTrq

说明	运动监控用于在 RL 程序中随时调整机器人各关节最大出力阈值;
定义	MotionSupJointTrq (J1,J2,J3,J4,J5,J6,J7);

示例	例 1 MotionSupJointTrq(50,50,70,50,60,60,50);

15.4.16.33 PathRecStart

说明	开始记录机器人的路径。是一个停前瞻指令。 执行 PathRecStart 开启路径记录后，支持记录的运动类型有 MoveL、MoveJ、MoveAbsJ、MoveC、TrigL、TrigC、TrigJ。路径记录时 MoveC、TrigC 指令执行时不能被中断打断，否则将无法记录目标点。MoveC、TrigC 指令被中断打断时将报错停止。
定义	PathRecStart();
示例	MoveL(p1,v1000,z50,tool0,wobj0); PathRecStart(); MoveL(p2,v1000,z50,tool0,wobj0); 前瞻到 PathRecStart，停止前瞻，待 PathRecStart 前的运动结束后再执行 PathRecStart，在此指令之后的运动指令将被记录。

15.4.16.34 PathRecStop

说明	停止记录机器人的路径，并将记录的路径信息清空。是一个停前瞻指令。
定义	PathRecStop();
示例	MoveL(p1,v1000,z50,tool0,wobj0); PathRecStop(); MoveL(p2,v1000,z50,tool0,wobj0); 前瞻到 PathRecStop，停止前瞻，待 PathRecStop 前的运动结束后再执行 PathRecStop，在此指令之后 将不会再记录路径并且记录的路径将被清空。

15.4.16.35 PathRecBwd

说明	使机器人沿所记录的路径后退。是一个停前瞻的运动指令，PathRecBwd 的运动不会被记录。 如果运动指令之间有打断前瞻的指令，则执行记录的路径时，打断前瞻指令前的轨迹的转弯区会被设置为 z0。 执行 PathRecStart 开启路径记录后，触发中断，中断中 PathRecBwd 指令必须在其他运动指令之前，否则将报错停止。 PathRecBwd 无法被中断，当执行 PathRecBwd 时触发中断将报错停止。
定义	PathRecBwd();
示例	MoveAbsJ(j1,v1000,z50,tool_weld); PathRecStart(); MoveL(p1,v1000,z50,tool0,wobj0); MoveAbsJ(j2,v1000,z50,tool_weld); PathRecBwd(); MoveL(p2,v1000,z50,tool0,wobj0); PathRecStart 之后记录了两条运动指令。当执行到 PathRecBwd 时，机器人将进行轨迹回退，从 j2->p1->PathRecBwd 执行时机器人的位置(j1)

15.4.16.36 PathRecFwd

说明	使机器人后退至执行 PathRecBwd 处。停前瞻的运动指令，PathRecFwd 的运动不会被记录。 如果运动指令之间有打断前瞻的指令，则执行记录的路径时，打断前瞻指令前的轨迹的转弯区会被设置为 z0。 执行 PathRecStart 开启路径记录后，触发中断，中断中 PathRecFwd 后不可以有运动指令，否则

	将报错停止。 PathRecFwd 无法被中断，当执行 PathRecFwd 时触发中断将报错停止。
定义	PathRecFwd();
示例	MoveAbsJ(j1,v1000,z50,tool_weld); PathRecStart(); MoveL(p1,v1000,z50,tool0,wobj0); MoveAbsJ(j2,v1000,z50,tool_weld); PathRecBwd(); PathRecFwd(); 执行到 PathRecBwd 时，机器人进行路径回退【j2->p1->PathRecBwd 执行时机器人的位置(j1)】，此时机器人在 j1 的位置，当执行 PathRecFwd，机器人将从 j1->p1->j2 回到 j2 位置。

15.4.16.37 GetRecStartStatus

说明	获取记录路径是否开启。返回值是 bool 类型
定义	GetRecStartStatus();
示例	bool b1 = GetRecStartStatus(); 获取是否开启路径记录。返回 true 表示开启，false 则表示未开启。

15.4.16.38 SetMaxMotionJerk

说明	设置机器人底层 otgc jerk 的数值，默认值为 300，可以通过增大此值以实现快速启停的效果，pptomain 后恢复默认值;
定义	SetMaxMotionJerk(Num); Num：数据类型 int/double 大于 100
示例	例 1 SetMaxMotionJerk(10000);

15.4.16.39 VibSuppression

说明	设置振动抑制功能开关的指令。Pptomain 时，振动抑制功能恢复 HMI 界面设置的状态（见 9.8 章中动力学设置模块的振动抑制功能）。
定义	VibSuppression(Type); Type，数据类型：关键字，on 打开，off 关闭
示例	VibSuppression(on); MoveAbsJ(p1,v500,fine,tool1); MoveAbsJ(p2,v500,fine,tool1); MoveAbsJ(p3,v500,fine,tool1); VibSuppression(off); MoveAbsJ(p4,v500,fine,tool1); 执行到 VibSuppression(on)指令时，控制器开启振动抑制功能。后续 p1~p3 点的运动收到振动抑制功能的影响，启停时机器人的振动得到抑制，机器人的轨迹精度得到提高。执行到 VibSuppression(off)指令时，控制器关闭振动抑制功能。

15.4.17 功能指令

15.4.17.1 CRobT

说明	用于获取机器人位姿。使用该函数时，需要给定工具名称和工件名称，返回指定工具坐标系的 pose，当前的轴配置信息以及外部轴位置。

	注意：使用 CRobT 时，机器人应处于停止状态，即 CRobT 之前的运动语句转弯区设置应为 fine。
定义	返回值，数据类型： robtarget，返回当前机器人的位置、姿态、轴配置数据以及外部轴信息。 **CRobT(Tool， Wobj);** **Tool**，数据类型： tool，计算位置时使用的工具。 **Wobj**，数据类型： wobj，计算位置时使用的工件。
示例	例 1 p2 = CRobT(tool1， wobj2);

15.4.17.2 CJointT

说明	CJointT 用于读取机器人轴和外部轴当前角度。 注意：使用 CJointT 时，机器人应处于停止状态，即 CRobT 之前的运动语句转弯区设置应为 fine。
定义	返回值，数据类型：jointtarget，旋转轴单位：度，线性轴单位：mm，返回机器人轴和外部轴的当前角度值。 **CJointT();**
示例	例 1 VAR jointtarget j2; j2 = CJointT();

15.4.17.3 CalcJointT

说明	根据指定的 robtarget 变量计算对应的关节角度
定义	返回值，数据类型： jointtarget，返回输入位置对应的关节角度和外部轴位置。关节角度的单位是度（°），直线外部轴的单位是毫米（mm），旋转外部轴的单位是度（Degree）。 **CalcJointT(Rob_Target，Tool，Wobj);** **Rob_Target**，数据类型： robtarget，指定的笛卡尔空间目标点，请注意该点定义时使用的工具和工件应和 CalcJointT 指令中使用的工具/工件保持一致，否则可能会导致错误的结果。 **Tool**，数据类型： tool，计算关节角度时使用的工具，注意需要与定义所用的 robtarget 时使用的工具一致。 **Wobj**，数据类型： wobj，计算关节角度时使用的工件，注意需要与定义所用的 robtarget 时使用的工件一致。
示例	例 1 jpos2 = CalcJointT(pt1, tool1,wobj2); 计算 tool1 到达 pt1 点时对应的关节角度，并赋值给 jpos2， pt1 点是在工件 wobj2 下定义的。

15.4.17.4 CalcRobt

说明	根据指定的关节角度计算对应的笛卡尔空间位姿。
定义	返回值，数据类型： robtarget，返回给定关节角对应的笛卡尔空间位姿。 **CalcRobt(Joint_Target，Tool，Wobj);** **Joint_Target**，数据类型： jointtarget，给定的用来计算笛卡尔空间位姿的关节角度。 **Tool**，数据类型： tool，计算笛卡尔空间位姿时使用的工具。 **Wobj**，数据类型： wobj，计算笛卡尔空间位姿时使用的工件。
示例	例 1 pt1 = CalcRobT(jpos1, tool2,wobj); 根据关节角度 jpos1 来计算笛卡尔位姿，并赋值给 pt1。 pt1 是工具坐标系 tool2 在工件坐标系 wobj1 下描述的位姿。

15.4.17.5 Print

说明	将用户定义的内容打印输出到示教器，用户可以使用该函数对程序进行调试。 Print 函数的输入参数比较特殊，输入参数的个数不限，但至少要有一个，且每个参数必须为一个定义过的变量或者常量。 系统将这些变量转换为字符串并串接在一起，最后输出到程序编辑器的调试窗口。（调试窗口关闭时，print 的信息仍然被记录，记录数量上限为 500 条）
定义	Print(var1， var2，.....);

示例	例 1
	counter = 0;
	while(true)
	　　counter++;
	　　Print("counter = ",counter);
	endwhile
	该程序段执行后， HMI 的程序调试窗口将打印出如下信息：
	counter = 1
	counter = 2
	counter = 3
	counter = 4
	.....
	
	注意：当需要输出字符串时可以使用双引号""来包含想要显示的字符，但不支持在双引号中嵌套双引号。

15.4.17.6 Print_f

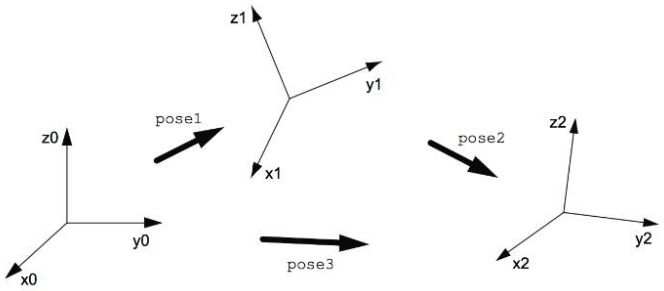
说明	Print_f 函数的用法类似 Print 指令，但是将参数内容输出到特定的日志文件，用户可以使用该函数记录程序的关键数据、关键事件。通过备份功能可以导出查看。 Print_f 函数的输入参数的个数不限，但至少要有一个，且每个参数必须为一个定义过的变量或者常量。 系统将这些变量转换为字符串并串接在一起，最后输出到日志文件中。
定义	Print_f(var1， var2，.....);

示例	例 1
	counter = 0;
	while(true)
	　　counter++;
	　　Print_f("counter = ",counter);
	　　// 其他功能
	endwhile
	该程序段执行后，用户日志文件会记录 counter 的变化情况，可以在导出的 rl_log 中看到

```
g3log: created log file at: Sat Nov 23 12:54:56 2024
    Running on machine: ubuntu
    LOG format : [level | mmdd hh:mm:ss.uuuuuu] msg

[RLINFO|11/23 12:55:02.632768 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 1
[RLINFO|11/23 12:55:03.637821 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 2
[RLINFO|11/23 12:55:04.645517 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 3
[RLINFO|11/23 12:55:05.649153 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 4
[RLINFO|11/23 12:55:06.661848 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 5
[RLINFO|11/23 12:55:07.675333 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 6
[RLINFO|11/23 12:55:08.677965 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 7
[RLINFO|11/23 12:55:09.682166 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 8
[RLINFO|11/23 12:55:10.686953 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 9
[RLINFO|11/23 12:55:11.693370 11058][xMateCR7][demo_step_back]task0:main.mod:MAIN[5]->counter = 10
```

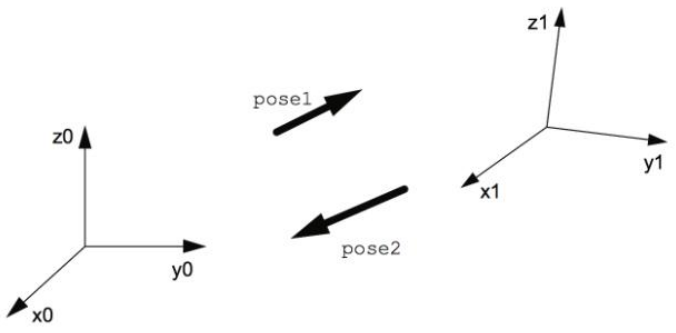
……

15.4.17.7 PoseMult

说明	PoseMult 用于计算两个位姿变换的乘积
定义	pose3 = PoseMult(pose1，pose2)； **pose1**，数据类型，pose，位姿 1。 **pose2**，数据类型，pose，位姿 2。 **pose3**，返回值，数据类型：pose，表示位姿乘积结果。
示例	 pose1 代表坐标系 1 相对坐标系 0 的位姿，pose2 代表坐标系 2 相对坐标系 1 的位姿，坐标系 2 相对坐标系 0 的位姿 pose3 可以通过以下方式计算： VAR pose pose1; VAR pose pose2; VAR pose pose3; … pose3 = PoseMult(pose1, pose2);

15.4.17.8 PoseInv

说明	PoseInv 用于计算一个位姿变换的逆。
定义	pose2 = PoseInv(pose1); **pose1**，数据类型：pose，输入位姿； **pose2**，数据类型：pose，返回值。
示例	pose1 代表坐标系 1 相对坐标系 0 的位姿，pose2 代表坐标系 0 相对坐标系 1 的位姿。 如果已知 pose1，pose2 可以通过以下方式计算： VAR pose pose1; VAR pose pose2; … pose2 = PoseInv(pose1);

15.4.17.9 GetRobABC

说明	获取笛卡尔空间点 P 的欧拉角姿态 ABC；旋转顺序定义为，初始坐标系（运动指令中选定的工件坐标系）首先绕自身 X 轴旋转后，再绕初始坐标系的 Y 轴旋转，最后绕初始坐标系的 Z 轴旋转
定义	double db_arr[3] = GetRobABC(Point [, A, B, C]); Point，数据类型： robtarget，计算位置时使用的笛卡尔点位。 A, B, C，数据类型： double，笛卡尔点位欧拉角姿态的返回值。 **返回值**，数据类型： double 三维数组，笛卡尔点位欧拉角姿态的返回值。

示例	示例 1
	point0 是一个笛卡尔点位变量，将变量的欧拉角姿态转换成 RL 的 Double 变量，可以使用下面的 RL 指令
	VAR double Rob_A;
	VAR double Rob_B;
	VAR double Rob_C;
	// 将 point0 的欧拉角赋值到 Rob_A|B|C
	GetRobABC(point0, Rob_A, Rob_B, Rob_C);
	
	示例 2
	point0 是一个笛卡尔点位变量，生成一个临时变量数组来存储笛卡尔点位的欧拉角信息，可以使用下面的 RL 指令
	double db_arr[3] = GetRobAbc(point0);

15.4.17.10 SetRobABC

说明	根据输入的欧拉角 ABC，设置笛卡尔空间点 P 的姿态；旋转顺序定义为，初始坐标系（运动指令中选定的工件坐标系）首先绕自身 X 轴旋转后，再绕初始坐标系的 Y 轴旋转，最后绕初始坐标系的 Z 轴旋转。
定义	SetRobABC(Point , A, B, C); Point，数据类型： 笛卡尔点位，将要被修改姿态的笛卡尔点位。 A, B, C，数据类型： double，设置笛卡尔点位的欧拉角姿态，单位°（度）。
示例	示例 1 point0 是一个笛卡尔点位变量，点位绕定轴 X,Y,Z 轴依次旋转 30°，60°，90°。 SetRobABC(point0, 30, 60, 90);

15.4.17.11 RotRobABC

说明	根据输入的欧拉角 ABC，在笛卡尔空间 P 的已有的姿态前提下，再进行欧拉角的旋转；旋转顺序

	定义为，初始坐标系（P 自身的姿态）首先绕自身 X 轴旋转后，再绕初始坐标系的 Y 轴旋转，最后绕初始坐标系的 Z 轴旋转。表现为已有的欧拉角跟输入的 ABC 相加。
定义	RotRobABC(Point , A, B, C); Point，数据类型： 笛卡尔点位，将要被修改姿态的笛卡尔点位。 A, B, C，数据类型： double，设置笛卡尔点位旋转的欧拉角姿态，单位°（度）。
示例	示例 1 point0 是一个笛卡尔点位变量，将点位绕 X,Y,Z 再旋为 30°, 60°, 90° RotRobABC(point0, 30, 60, 90);

15.4.17.12 OpMode

说明	用于获取机器人当前的运行模式
定义	ret = OpMode(); ret，返回值，数据类型： int，0 为未定义模式，1 为自动模式，2 为手动模式。

示例	示例 1
	int mode = OpMode();
	print(mode);
	若当前为自动模式则返回 1，手动模式则返回 2。

15.4.18 寄存器指令

15.4.18.1 ReadRegByName

说明	根据寄存器的名字，读取对应寄存器的变量值
定义	ReadRegByName(RegData，Value); RegData，数据类型：可读的寄存器变量 ，设置>通信>寄存器界面功能，寄存器变量。 Value ，数据类型：bool/int/double ，寄存器数据将写入 Value，如果寄存器的变量类型和解释器变量不一致，将自动进行格式转换
示例	例 1 int tmp_int; ReadRegByName(modbus_int_read[6], tmp_int); 将名为 modbus_int_read 的下标为 6 的数据，读取到 tmp_int 变量中

15.4.18.2 WriteRegByName

说明	根据寄存器的名字，读取对应寄存器的变量值 ，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	WriteRegByName(RegData，Value); RegData，数据类型：可写的寄存器变量 ，设置>通信>寄存器界面功能，寄存器变量。 Value，数据类型：bool/int/double ，将 Value 数据写入寄存器中，如果寄存器的变量类型和解释器变量不一致，将自动进行格式转换。
示例	例 1 WriteRegByName(modbus_int_write[6], 200); 将 INT 200 的数据，写入 modbus_int_write[6] 对应的寄存器中。

15.4.18.3 ReadRegByteByName

说明	根据寄存器的名字，读取寄存器对应字节的值

定义	ReadRegByteByName(RegData，Value，byteFlag)； RegData，数据类型：可读或可写的 int16/int32 类型寄存器，通信>寄存器界面功能，寄存器变量。 Value ，数据类型：byte 类型变量，读取寄存器对应字节值到 Value 中，Value 必须为 byte 类型变量。 byteFlag ，字节标志位。取值范围 1-4，1 为低字节，4 为高字节。
示例	例 1 byte tmp_value; ReadRegByteByName(modbus_reg, tmp_value,1); 将名为 modbus_reg 寄存器的第一个字节，读取到 tmp_value 变量中

15.4.18.4 WriteRegByteByName

说明	根据寄存器的名字，写对应寄存器字节的值 ，该指令如果放在运动指令之后使用，不会打断转弯区，在运动指令轨迹终点触发，或在转弯区起点触发，具体使用可参考 SetDO 例 2。
定义	WriteRegByteByName(RegData，Value，byteFlag); RegData，数据类型：可写的 int16/int32 类型寄存器变量 ，通信>寄存器界面功能，寄存器变量。 Value ，数据类型：byte，将 Value 数据写入寄存器对应字节中 byteFlag ，字节标志位。取值范围 1-4，1 为低字节，4 为高字节。
示例	例 1 WriteRegByteByName(modbus_reg, 200,2); 将 200 的数据，写入 modbus_reg 的第 2 个字节中。

15.4.19 末端工具指令

15.4.19.1 JodellGripInit

说明	钧舵电爪的初始化指令
定义	JodellGripInit (ID,wait_time); ID，数据类型：Int 变量 ，建立通信，初始化钧舵电爪，参数 ID。 Wait_time，数据类型：Int 变量 ，等待初始化完成，等待时间阈值，超时报错；单位：s。
示例	

15.4.19.2 JodellGripMove

说明	钧舵电爪的运动指令
定义	JodellGripMove (ID,Pos,Vel,Trq); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Int 变量 ，目标位置，无单位，设置范围 0-255。 Vel，数据类型：Int 变量 ，电爪运行的速度，无单位，设置范围 0-255。 Trq，数据类型：Int 变量 ，电爪运行检测的力，无单位，设置范围 0-255。
示例	

15.4.19.3 JodellGripStatus

说明	获取钧舵电爪的状态
定义	JodellGripStatus (ID,Pos,Vel,Trq,Contact); ID，数据类型：Int 变量 ，获取电爪运动状态的电爪 ID。 Pos，数据类型：Int 变量 ，获取电爪当前位置，无单位，数据范围 0-255。

	Vel，数据类型：Int 变量 ，获取电爪运行的速度，无单位，数据范围 0-255。 Trq，数据类型：Int 变量 ，获取电爪运行的力矩，无单位，数据范围 0-255。 Contact，数据类型：Int 变量 ，获取电爪运行的状态字，无单位，数据范围 0-255，其中 bit6-7 表示电爪是否检测到物体的状态。

Bit	名称	值/说明
0	gAct	0 表示电动夹爪复位中，1 表示电动夹爪处于使能状态
2	gMode	0 表示处于参数控制模式，1 表示处于无参数控制模式
3	gGTO	0 表示停止，1 表示正在向目标位置运动
4-5	gSTA	0 电动夹爪处于复位或巡检状态，1 表示正在激活，2 表示未使用，3 表示激活完成
6-7	gOBJ	0 表示手指正在指定位置移动；1 表示手指在张开达到指定位置时，由于接触到物体停止；2 表示手指在闭合达到指定位置时，由于接触到物体停止；3 手指到达指定的位置，但没有检测到对象。

示例	

15.4.19.4 JodellSuckInit

说明	钧舵吸盘的初始化指令
定义	JodellSuckInit(ID); ID，数据类型：Int 变量 ，初始化此 ID 的吸盘，检测是否正确连接此 ID 的吸盘。
示例	

15.4.19.5 JodellSuckSet

说明	钧舵吸盘工作的指令 ，下发此指令后，吸盘立刻按照设定参数开始工作
定义	JodellSuckSet(ID,CH1_enable,CH1_VacMin, CH1_VacMax, CH1_Waittime, CH2_enable, CH2_VacMin, CH2_VacMax, CH2_Waittime); ID，数据类型：Int 变量 ，控制吸盘工作的吸盘 ID。 CH1_enable，数据类型：Int 变量 ，吸盘第一个通道是否工作，1 表示工作，0 表示不工作。 CH1_VacMin，数据类型：Int 变量 ，吸盘第一个通道最小真空度设置值，范围 0-255，0 表示纯真空，大于 100 表示吸盘释放；当实际真空度低于此阈值时停止抽气； CH1_VacMax，数据类型：Int 变量 ，吸盘第一个通道最大真空度设置值，范围 0-255，0 表示纯真空，大于 100 表示吸盘释放；当实际真空度高于此阈值时开始抽气； CH1_Waittime，数据类型：Double 变量 ，吸盘第一个通道超时设置值； CH2_enable，数据类型：Int 变量 ，吸盘第二个通道是否工作，1 表示工作，0 表示不工作。 CH2_VacMin，数据类型：Int 变量 ，吸盘第二个通道最小真空度设置值，范围 0-255，0 表示纯真空，大于 100 表示吸盘释放；当实际真空度低于此阈值时停止抽气； CH2_VacMax，数据类型：Int 变量 ，吸盘第二个通道最大真空度设置值，范围 0-255，0 表示纯真空，大于 100 表示吸盘释放；当实际真空度高于此阈值时开始抽气； CH2_Waittime，数据类型：Double 变量 ，吸盘第二个通道超时设置值；
示例	

15.4.19.6 JodellSuckStatus

说明	获取钧舵吸盘的状态
定义	JodellSuckStatus(ID,Vac1,Contact1,Time_Err1,Vac2,Contact2,Time_Err2);

	ID，数据类型：Int 变量 ，获取吸盘状态的吸盘 ID。 Vac1，数据类型：Int 变量 ，获取的吸盘通道一当前的实际真空度，数据范围 0-100。 Contact1，数据类型：Int 变量 ，获取的吸盘通道一当前的状态信息数据，数据范围 0-255，其中 bit6-7 表示是否检测到物体，具体的状态信息含义见下表。

Bit	名称	值/说明
0	gAct	0 表示电动吸盘未使能，1 表示电动吸盘已使能
2	gMode	0 表示处于自动控制模式，1 表示处于高级控制模式
3	gGTO	0 表示停止调节，1 表示正在调节压力或真空度
4-5	gSTA	0 电动吸盘处于未激活状态，1 和 2 未使用，3 表示吸盘已激活
6-7	gOBJ	0 表示气压已低于最低气压；1 表示检测到工件，最低压力值已经达到；2 表示检测到工件，最高压力值已达到；3 表示没有检测到对象，物体已丢失或脱落。

	Time_Err1，数据类型：Int 变量，获取的吸盘通道一是否超时报警。 Vac2，数据类型：Int 变量，获取的吸盘通道二当前的实际真空度，数据范围 0-100。 Contact2，数据类型：Int 变量，获取的吸盘通道二当前的状态信息数据，数据范围 0-255，其中 bit6-7 表示是否检测到物体，具体的状态信息含义与上表相同。 Time_Err2，数据类型：Int 变量，获取的吸盘通道二是否超时报警。
示例	

15.4.19.7 RMRGMGripInit

说明	增广电爪 RM-RGM 系列的初始化指令
定义	RMRGMGripInit(ID); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。
示例	

15.4.19.8 RMCGripInit

说明	增广电爪 RM-C 系列的初始化指令
定义	RMCGripInit (ID); ID，数据类型：Int 变量，控制电爪运动的电爪 ID。
示例	

15.4.19.9 RMRGMGripPosMove

说明	增广电爪 RMRGM 系列的位置模式的运动指令
定义	RMRGMGripPosMove(ID,Pos,Vel,Acc,PCheck); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Double 变量 ，目标位置，单位 mm，设置范围-2000.0-2000.0。 Vel，数据类型：Double 变量 ，电爪运行的速度，单位 mm/s，设置范围 0.01-1000.0。 Acc，数据类型：Double 变量 ，电爪运行加速度，单位 mm/s^2，设置范围 0.01-2000.0。 PCheck，数据类型：Double 变量 ，定位范围，单位 mm，设置范围 0.01-10.0。
示例	

15.4.19.10 RMCGripPosMove

说明	增广电爪 RMC 系列的位置模式的运动指令

定义	RMCGripPosMove(ID,Pos,Vel,Acc,PCheck); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Double 变量 ，目标位置，单位 mm，设置范围-2000.0-2000.0。 Vel，数据类型：Double 变量 ，电爪运行的速度，单位 mm/s，设置范围 0.01-1000.0。 Acc，数据类型：Double 变量 ，电爪运行加速度，单位 mm/s2，设置范围 0.01-2000.0。 PCheck，数据类型：Double 变量 ，定位范围，单位 mm，设置范围 0.01-10.0。
示例	

15.4.19.11 RMRGMGripTrqMove

说明	增广电爪 RMRGM 系列的力矩模式的运动指令
定义	RMRGMGripTrqMove(ID,Pos,Vel,Acc,Trq,PCheck,TCheck); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Double 变量 ，目标距离，单位 mm，设置范围-2000.0-2000.0。 Vel，数据类型：Double 变量 ，电爪运行的速度，单位 mm/s，设置范围 0.01-1000.0。 Acc，数据类型：Double 变量 ，电爪运行加速度，单位 mm/s2，设置范围 0.01-2000.0。 Trq，数据类型：Double 变量 ，定位范围，单位 Nm，设置范围 0.01-100.0。 PCheck，数据类型：Double 变量 ，定位范围，单位 mm，设置范围 0.01-10.0。 TCheck，数据类型：Double 变量 ，时间范围，单位 mm，设置范围 0.01-1000.0。
示例	

15.4.19.12 RMCGripTrqMove

说明	增广电爪 RMC 系列的力矩模式的运动指令
定义	RMCGripTrqMove (ID,Pos,Vel,Acc,Trq,PCheck,TCheck); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Double 变量 ，目标距离，单位 mm，设置范围-2000.0-2000.0。 Vel，数据类型：Double 变量 ，电爪运行的速度，单位 mm/s，设置范围 0.01-1000.0。 Acc，数据类型：Double 变量 ，电爪运行加速度，单位 mm/s2，设置范围 0.01-2000.0。 Trq，数据类型：Double 变量 ，定位范围，单位 Nm，设置范围 0.01-100.0。 PCheck，数据类型：Double 变量 ，定位范围，单位 mm，设置范围 0.01-10.0。 TCheck，数据类型：Double 变量 ，时间范围，单位 mm，设置范围 0.01-1000.0。
示例	

15.4.19.13 RMRGMGripStatus

说明	增广电爪 RMRGM 系列的获取状态指令
定义	RMRGMGripStatus (ID,Pos,Vel,Trq,Reach,Err); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Double 变量 ，电爪位置，单位 mm。 Vel，数据类型：Double 变量 ，电爪运行的速度，单位 mm/s。 Trq，数据类型：Double 变量 ，电爪出力力矩，单位%。 Reach，数据类型：Bool 变量 ，电爪是否到位。 Err，数据类型：Int 变量 ，电爪的错误代码。
示例	

15.4.19.14 RMCGripStatus

说明	增广电爪 RMC 系列的获取状态指令
定义	RMCGripStatus (ID,Pos,Vel,Trq,Reach,Err); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Double 变量 ，电爪位置，单位 mm。 Vel，数据类型：Double 变量 ，电爪运行的速度，单位 mm/s。 Trq，数据类型：Double 变量 ，电爪出力力矩，单位%。 Reach，数据类型：Bool 变量 ，电爪是否到位。 Err，数据类型：Int 变量 ，电爪的错误代码。
示例	

15.4.19.15 RMRGMResetErr

说明	增广电爪 RMRGM 系列的复位电爪错误指令
定义	RMRGMResetErr (ID); ID，数据类型：Int 变量 ，需要复位电爪的 ID。
示例	

15.4.19.16 RMCResetErr

说明	增广电爪 RMC 系列的复位电爪错误指令
定义	RMCResetErr(ID); ID，数据类型：Int 变量 ，需要复位电爪的 ID。
示例	

15.4.19.17 RobotiqGripInit

说明	Robotiq 电爪 2F_85 系列的初始化指令
定义	RobotiqGripInit(ID); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。
示例	

15.4.19.18 RobotiqGripGetStatus

说明	Robotiq 电爪 2F_85 系列的获取状态指令
定义	RobotiqGripGetStatus(ID，Pos，gOBJ，Err); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Int 变量 ，电爪的位置，范围 0-255。 gOBJ，数据类型：Int 变量 ，电爪的接触状态。0 表示电爪运动中，未接触物体；1 表示在电爪打开的过程中接触物体；2 表示在电爪闭合的过程中接触物体；3 表示电爪到达指定位置，但未接触物体。 Err，数据类型：Int 变量 ，电爪的错误码。
示例	

15.4.19.19 RobotiqGripMove

说明	Robotiq 电爪 2F_85 系列的运动指令
定义	RobotiqGripMove(ID，Pos，Vel，Trq); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Int 变量 ，电爪位置，设置范围 3-227。 Vel，数据类型：Int 变量 ，电爪运行的速度，设置范围 0-255 。 Trq，数据类型：Int 变量 ，电爪出力的力矩，设置范围 0-255。
示例	

15.4.19.20 DhGripInit

说明	大寰宇电爪 PGI-140-80 系列的初始化指令
定义	DhGripInit(ID Time_wait); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Time_wait，数据类型：double 变量，单位 s，设置范围 0-10。
示例	

15.4.19.21 DhGripGetStatus

说明	大寰宇电爪 PGI-140-80 系列的获取状态指令
定义	DhGripGetStatus(ID，Pos，gOBJ); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Int 变量 ，电爪的位置，范围 0-1000。 gOBJ，数据类型：Int 变量 ，电爪的接触状态，：0：运动中；1：到达位置；2：夹住物体；3：物体掉落；4、当 前 id 未初始化成功。
示例	

15.4.19.22 DhGripMove

说明	大寰宇电爪 PGI-140-80 系列的运动指令
定义	DhGripMove(ID，Pos，Vel，Trq); ID，数据类型：Int 变量 ，控制电爪运动的电爪 ID。 Pos，数据类型：Int 变量 ，电爪位置，设置范围 0-1000。 Vel，数据类型：Int 变量 ，电爪运行的速度，设置范围 1-100。 Trq，数据类型：Int 变量 ，电爪出力的力矩，设置范围 20-100。
示例	

15.4.20 中断指令

15.4.20.1 IRegister

说明	注册中断，确定中断是否可以单次触发、是否可调试(单步、单步暂停下触发中断)，以及中断号、触发 信号、触发类型。。一个触发源只能绑定一个中断函数。
定义	IRegister([\SINGLE,][\DEBUG,]int_num, signal, trigger_type, trap_function); []内为可选项 int_num，数据类型：中断号变量， 用作中断的标识 signal，寄存器信号/DI 信号 trigger_type，触发方式

	DI 信号触发方式: ● \Posflank：上升沿触发 ● \Negflank：下降沿触发 ● \Highlevel：高电平触发 ● \Lowlevel：低电平触发 注意：高低电平触发会持续触发中断 寄存器(int\bool\bit\byte)触发方式: 0 代表低电平，非 0 代表高电平 trap_function：中断函数名 \SINGLE，单次触发(可选项) \DEBUG，可调试(可选项)，添加此参数后可在单步、单步暂停状态触发中断
示例	IRegister \SINGLE,\DEBUG,intnum0,DI1_0,\Posflank,"trapfun1" 注册中断，当 DI 信号 DI1_0 发生上升沿变化时，执行 trapfun1 函数，触发一次后不再响应此中断；此中 断可在单步和单步暂停状态触发；intnum0 为此中断的标识符，用于 IEnable、IDisable、GetTrapData 指令

15.4.20.2 IEnable

说明	激活被 IDisable 禁用的中断，前瞻触发（IRegister 后默认是激活的）
定义	IEnable (int_num); int_num，数据类型：中断号变量， 用作中断的标识
示例	IEnable(intnum0); 激活 intnum0 对应的中断，激活后可被绑定的信源触发

15.4.20.3 IDisable

说明	禁用某个中断，前瞻触发。需要注意的是选择单次触发的中断，在被 IDisable 后 收到中断信号，仍视为单次触发结束
定义	IDisable(int_num); int_num，数据类型：中断号变量， 用作中断的标识
示例	IDisable(intnum0); 禁用 intnum0 对应的中断，禁用后不可被绑定的信源触发

15.4.20.4 GetTrapData

说明	获取某中断的信息，前瞻触发。
定义	GetTrapData (int_num,str); int_num，数据类型：中断号变量， 用作中断的标识 Str，数据类型：接收中断信息的字符串
示例	GetTrapData (intnum0,string0); 获取 intnum0 对应的中断信息，将信息放入 string0

16 附录

16.1 用户权限明细

类别	功能	操作员	示教员	编程员	管理员	超级管理员
工程相关与程序编辑	工程配置（新建、导入、导出、加载、恢复、另存为、设置默认工程）	N	N	Y	Y	Y
	查看工程（包括程序和 IO、变量、预定义参数等对象数据）	Y	Y	Y	Y	Y
	编辑工程（包括程序编辑和工具等对象设置）	N	N	Y	Y	Y
	示教点位（包括 RL 编辑器界面、变量列表的点位类型、点位列表）	N	Y	Y	Y	Y
机器人运动与程序运行	切换手/自动模式	Y	Y	Y	Y	Y
	上/下电	Y	Y	Y	Y	Y
	启动/停止程序	Y	Y	Y	Y	Y
	切换程序循环模式	Y	Y	Y	Y	Y
	调整程序运行速度	N	Y	Y	Y	Y
	单步调试程序	N	Y	Y	Y	Y
	Jog/拖动机器人	N	Y	Y	Y	Y
状态监控	查看运行数据	Y	Y	Y	Y	Y
	设置 IO 信号	N	Y	Y	Y	Y
	设置寄存器值	N	Y	Y	Y	Y
	变量监控	N	Y	Y	Y	Y
设置	控制器设置-基本设置	N	N	Y	Y	Y
	控制器设置-高级设置	N	N	Y	Y	Y
	控制器设置-授权设置	N	N	N	Y	Y
	HMI 设置-基本设置	N	N	N	Y	Y
	HMI 设置-示教器模式	N	N	Y	Y	Y
	用户组	Y	Y	Y	Y	Y
	零点标定	N	N	Y	Y	Y
	基坐标系标定	N	N	Y	Y	Y
	动力学设置	N	N	Y	Y	Y
	本体参数	N	N	Y	Y	Y
	运动参数	N	N	Y	Y	Y
	力控参数	N	N	Y	Y	Y
	快速调整	N	Y	Y	Y	Y
	电子铭牌	N	N	Y	Y	Y
	错误码报警过滤	N	N	Y	Y	Y
	自定义按键	N	N	Y	Y	Y
通信	系统 IO	N	N	Y	Y	Y
	外部通信	N	N	Y	Y	Y

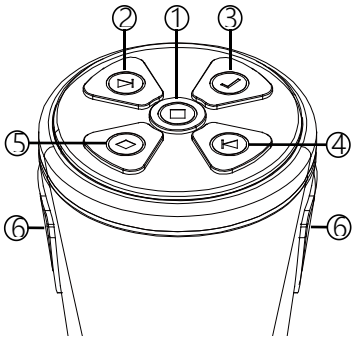
	IO 设备	N	N	Y	Y	Y
	总线设备	N	N	Y	Y	Y
	寄存器	N	N	Y	Y	Y
	末端工具	N	N	Y	Y	Y
	RCI 设置	N	N	Y	Y	Y
	xPanel 配置	N	N	Y	Y	Y
	电爪吸盘	N	N	Y	Y	Y
	串口设置	N	N	Y	Y	Y
	编码器	N	N	Y	Y	Y
	OPC-UA	N	N	Y	Y	Y
安全	软限位	N	N	Y	Y	Y
	虚拟墙	N	N	Y	Y	Y
	碰撞检测	N	N	Y	Y	Y
	安全区域	N	Y	Y	Y	Y
	安全监视器	N	N	Y	Y	Y
	协作模式	N	N	Y	Y	Y
	安全位置	N	Y	Y	Y	Y
工艺包	传送带	N	N	Y	Y	Y
	导轨	N	N	Y	Y	Y
日志	日志查询	Y	Y	Y	Y	Y
	诊断设置	N	N	N	N	Y
选项	连接/关于/演示	Y	Y	Y	Y	Y
	软件升级-控制器升级/其他设置	N	N	N	Y	Y
	软件升级-HMI 升级/控制器备份	Y	Y	Y	Y	Y
	导出	Y	Y	Y	Y	Y
	导入	N	N	N	Y	Y
	文件管理器	N	N	N	Y	Y

16.2 协作机器人末端把手介绍

16.2.1 ER 系列

xMate ER 系列机器人末端集成智能交互面板 Pilot 把手，在拖动模式下，通过手柄上按键可快速进行点位示教、连续轨迹示教，实现更好的人机交互。

末端 Pilot 把手按键定义：

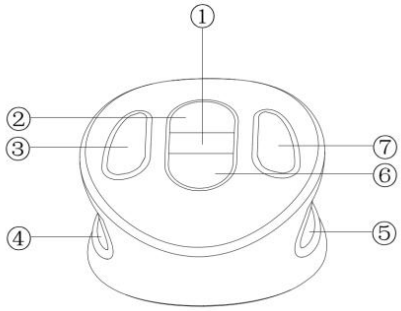


标号	定义
①	以当前位姿更新示教点，开始/停止轨迹录制
②	下一个
③	增加列表中点位/轨迹，确认弹窗提示，弹出提示取消
④	上一个
⑤	删除列表中点位/轨迹，弹窗提示取消
⑥	拖动模式下，两个使能按键同时按下开启拖动功能

16.2.2 CR 系列

xMate CR 系列机器人末端集成智能交互面板 xPanel 把手，在拖动模式下，通过手柄上按键可快速进行点位示教、连续轨迹示教，实现更好的人机交互。

末端 xPanel 把手按键定义：



标号	定义
①	增加列表中点位/轨迹，确认弹窗提示
②	列表中光标向前方向
③	删除列表中点位/轨迹，取消弹窗提示
④	拖动模式下，两个使能按键同时按下开启拖动功能
⑤	
⑥	列表中光标向后方向
⑦	以当前位姿更新示教点，确认弹窗提示，开始/停止轨迹录制

16.3 点位、路径示教(基于协作机器人末端把手)

16.3.1 点位示教

在操作面板打开拖动使能开关，机器人自动上电并启用拖动模式，通过 RobotAssist 软件和末端拖动把手进行以下操作：

步骤	说明
1.创建/加载一个工程，进入点位列表界面；	末端按键只在 RobotAssist 软件当前打开页面为点位列表/

	路径列表时响应；
2.同时按下末端把手两个使能按键，拖动机器人至任意一个位置后松开拖动使能，按下末端把手按键"点位增加"按钮；	点位列表末尾新增一个当前位姿信息的示教点，光标定位到新增示教点；
3.按下末端把手按键"上一个/下一个"按钮	点位列表中光标向上一个/下一个点位移动并选中；
4.在点位列表中选择一个点位更新位置，拖动机器人至另一个位置后松开拖动使能按键，按下末端把手按键"点位更新"按钮；	更新点位时有弹窗提示，按下末端按键"确认"按钮，当前选择点位以当前位姿更新；按下末端按键"取消"按钮，弹窗提示取消，当前选择点位保持不变；
5.在点位列表中选择一个点位进行删除，按下末端把手按键"点位删除"按钮并确认；	删除点位时有弹窗提示，按下末端按键"确认"按钮，当前选择点位从点位列表中删除；按下末端按键"取消"按钮，弹窗提示取消，当前选择点位保留；

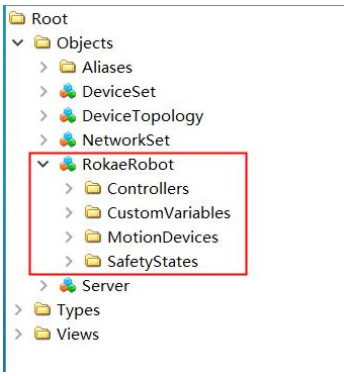
16.3.2 路径示教

在操作面板打开拖动使能开关，机器人自动上电并启用拖动模式，通过 RobotAssist 软件和末端拖动把手进行以下操作：

步骤	说明
1.创建/加载一个工程，调整机器人至任意初始位置，进入路径列表界面；	末端按键只在 RobotAssist 软件当前打开页面为路径列表、路径列表时响应；
2.按下末端把手按键"路径增加"按钮；	路径列表末尾新增一个路径，光标定位到新增路径；
3.按下末端把手按键"上一个/下一个"按钮；	路径列表中光标向上一个/下一个路径移动并选中；
4.在路径列表中选择一个路径开始录制，按下末端把手按键"开始轨迹录制"按钮，同时按下末端把手两个使能按键拖动机器人进行轨迹录制；	按下"开始轨迹录制"按钮后，轨迹录制开始；直到按下"停止轨迹录制"按钮，录制结束并自动保存；
5.在路径列表中选择一个路径进行删除，按下末端把手按键"路径删除"按钮并确认；	删除路径时有弹窗提示，按下末端按键"确认"按钮，当前选择路径从路径列表中删除；按下末端按键"取消"按钮，弹窗提示取消，当前选择路径保留；

16.4 OPC-UA Robotics 模型

xCore 控制器系统的 OPCUA 服务端默认支持 OPC 40010-1 OPC-UA for Robotics, Part 1: Vertical Integration 标准规定的所有强制性选项，顶级目录如下所示：

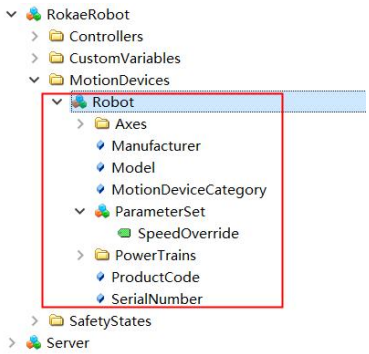


类型	名称(Browse Name)	描述
MotionDeviceSystem	RokaeRobot	每个服务器有一个 MotionDeviceSystem 类型的实例，名称为"RokaeRobot"，放在 Objects 节点下面；

MotionDeviceSystem 子节点如下：

类型	名称(Browse Name)	描述
MotionDevices	MotionDevices	可容纳 MotionDeviceType 实例的容器
Controllers	Controllers	可容纳 MotionDeviceType 实例的容器
SafetyStates	SafetyStates	可容纳 ControllerType 实例的容器
CustomVariables	CustomVariables	容纳自定义变量的容器，此节点是 Robotics 标准模型的一个扩展，所有用户自定义的变量可以在此节点下找到

16.4.1 MotionDevices 模型

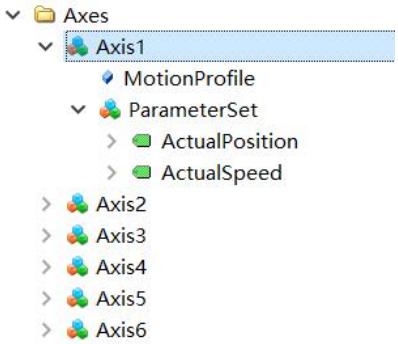


类型	名称(Browse Name)	描述
MotionDevice	Robot	MotionDeviceType 类型的实例，名称为“Robot”

MotionDevice 子节点如下：

类型	名称(Browse Name)	描述
MotionDeviceCategory	MotionDeviceCategory	ISO 8373 规定的运动设备类型，设置为：ARTICULATED_ROBOT， 即关节机器人
Manufacturer	Manufacturer	制造商
Model	Model	机器人型号
ProductCode	ProductCode	产品号，暂不支持
SerialNumber	SerialNumber	序列号，暂不支持
ParameterSet	ParameterSet/SpeedOverride	程序速度，RL 程序速度：1-100%

16.4.1.1 Axes 子节点

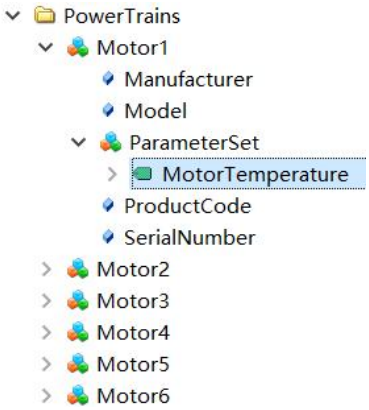


类型	名称(Browse Name)	描述
Axis	Axis%1	1、每个实例对应一个轴，6 轴机器人对应 6 个实例 2、根据轴编号进行命名，如：Axis1、Axis2

Axis 子节点如下：

类型	名称(Browse Name)	描述
MotionProfile	MotionProfile	轴类型，一般为：1 (ROTARY)
ParameterSet	ParameterSet/ActualPosition	当前轴位置，单位：度
	ParameterSet/ActualSpeed	当前轴速度，度/s

16.4.1.2 PowerTrains 子节点

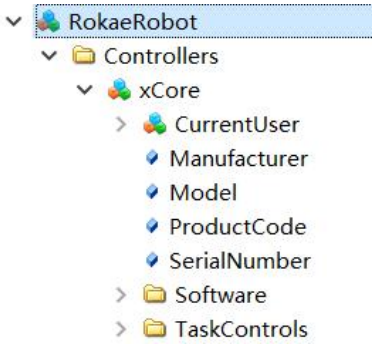


类型	名称(Browse Name)	描述
PowerTrain	Motor%1	1、每个实例对应一个轴动力单元，6 轴机器人对应 6 个实例，包含 6 个 MotorType 实例 2、根据轴编号进行命名，如：Motor1、Motor2

每个 Motor 子节点如下：

类型	名称(Browse Name)	描述
Manufacturer	Manufacturer	制造商，暂不支持
Model	Model	机器人型号，暂不支持
ProductCode	ProductCode	产品号，暂不支持
SerialNumber	SerialNumber	序列号，暂不支持
ParameterSet	ParameterSet/MotorTemperatur	电机温度，暂不支持

16.4.2 Controllers 模型

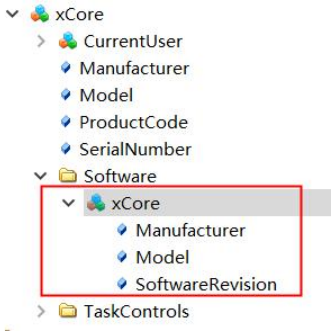


类型	名称(Browse Name)	描述
ContorllerType	xCore	1、每个机器人包含一个 ContorllerType 实例，名称为 xCore

每个 ContorllerType 子节点如下：

类型	名称(Browse Name)	描述
Manufacturer	Manufacturer	制造商
Model	Model	机器人型号
ProductCode	ProductCode	产品号，暂不支持
SerialNumber	SerialNumber	序列号，暂不支持
CurrentUser	CurrentUser	暂不支持
Software	Software	包含 SoftwareType 实例的容器
TaskControls	TaskControls	包含 TaskControlType 实例的容器

16.4.2.1 Software 子节点

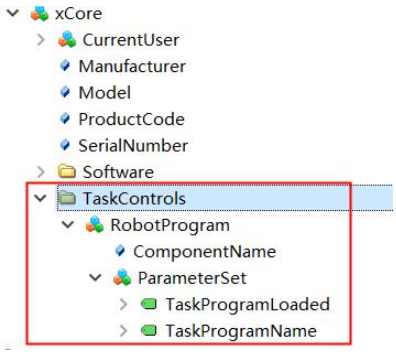


类型	名称(Browse Name)	描述
Software	xCore	xCore 控制系统

每个 ContorllerType 子节点如下：

类型	名称(Browse Name)	描述
Manufacturer	Manufacturer	制造商
Model	Model	机器人型号
SoftwareRevision	SoftwareRevision	软件版本号，如：2.1.2

16.4.2.2 TaskControls 子节点

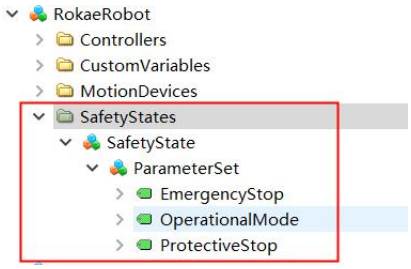


类型	名称(Browse Name)	描述
TaskControl	RobotProgram	名称为 RobotProgram 的 TaskControlType 实例

每个 ContorllerType 子节点如下：

类型	名称(Browse Name)	描述
ComponentName	ComponentName	为空
ParameterSet	ParameterSet/TaskProgramName	显示当前 RL 工程名称
	ParameterSet/TaskProgramLoaded	若已加载工程，则显示为：true，无工程显示为：false

16.4.3 SafetyStates

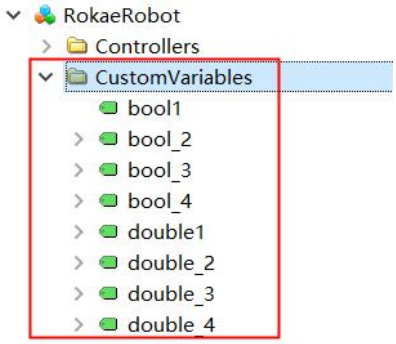


类型	名称(Browse Name)	描述
SafetyState	SafetyState	一个名称为 SafetyState 的 SafetyStateType 实例

SafetyState 子节点如下：

类型	名称(Browse Name)	描述
ParameterSet	ParameterSet/EmergencyStop	为空
	ParameterSet/OperationalMode	操作模式，枚举值：0-Other、1-Manual Reduced Speed、2-Manual High Speed、3-Automatic、 4-Automatic External 目前控制系统中的手动和自动模式，分别对应上诉枚举中的 1 和 3
	ParameterSet/ProtectiveStop	控制器内部安全门停止和保护停止状态取或

16.4.4 CustomVariables



在 11.13 节中配置的自定义变量都可以在此节点下找到，客户端可以通过读写此节点下变量实现与控制器数据的交互。

17 故障排查

17.1 控制系统错误码

17.1.1 1XXXX

错误码	错误描述	故障可能原因	修复方法
10000	HMI 请求包解析错误	HMI 请求数据包协议不正确	请检查 HMI 和控制系统版本是否匹配
10001	JOG 启动失败	1.未处于手动模式; 2.机器人未上电; 3.机器人运动中; 4.控制模式非位置模式	请确保机器人处于手动模式并上电
10002	快速调整启动失败	1.未处于手动模式; 2.机器人未上电; 3.机器人运动中; 4.控制模式非位置模式	请确保机器人处于手动模式并上电
10003	机械零点标定失败	1.未处于手动模式; 2.机器人运动中; 3.控制模式非位置模式	请确保机器人处于手动模式并停止运动
10004	机械零点标定成功	无	无
10005	传感器零点标定失败	1.未处于手动模式; 2.机器人运动中; 3.传感器力矩偏差超阈值; 4.控制模式非位置模式	请将机器人运动到机械零点后，确保机器人处于手动模式并停止运动
10006	传感器零点标定成功	无	无
10007	重启控制器失败		请停止机器人运动或者程序运动
10008	清除编码器报警失败	伺服编码器故障	重启机器人,如无法解决请联系技术支持
10009	清除编码器报警成功	无	无
10010	切换至手动模式失败	机器人运行中不允许切换模式；或已处于手动模式	停止运行再试切换
10011	切换至自动模式失败	以下原因会导致切换失败: 1.机器人处于运动中; 2.机器人处于急停状态; 3.拖动已开启; 4.已经为自动模式	1.停止运动; 2.恢复急停状态; 3.关闭拖动
10012	上电条件检查失败	1.未处于正确模式; 2.机器人在急停状态; 3.机器人处于力矩控制模式;4.收到关机信号;5.伺服处于严重故障状态;6.控制器未启动完成	1.切换到正确模式; 2.恢复急停状态;3.切换机器人至位置模式;4.检查伺服故障;5.等待控制器启动完成
10013	下电失败	机器人在运行中，或未处于自动模式	停止机器人运行
10014	开启拖动功能失败！	以下状态无法打开拖动: 1.自动模式; 2.上电状态; 3.非位置模式; 4.运动中; 5.非常规任务状态下; 6.触发了安全监控或者安全监视器！	切换到手动模式，保持电机下电状态，位置模式，重新尝试；或者重启后尝试。
10015	开启拖动功能成功	无	无
10016	关闭拖动功能失败！	工作模式不正确	1.如果是通过 RCI 打开拖动，需要通过 RCI 关闭; 2.重启机器人尝试恢复。

10017	关闭拖动成功	无	无
10018	虚拟墙更新失败	1.设置的区域过小; 2.机器人当前位置位于设置的区域外; 3.未打开拖动模式	1.扩大虚拟墙边界; 2.将机器人法兰末端运动到设置的区域内; 3.打开拖动模式
10019	虚拟墙更新成功	无	无
10020	拒绝新示教器连接	已有示教器连接	断开已有连接
10021	socket 连接服务端超时	连接超时	1.请检查设备连接情况; 2.请检查服务端是否工作正常
10022	socketread 指令接收数据格式错误	网络故障	
10023	socket 创建 ip 和端口被占用	RL 程序，或者外部通讯 socket 中已经使用了该 ip 和端口号，不允许两个 socket 使用相同的 ip 和端口	设置不同的 IP 地址或端口号
10024	socket 创建名称重复	socket 创建名称重复	更换别的 socket 名
10025	socket 接收数据失败	1.网络连接异常; 2.服务端未及时发送数据; 3.结束符错误; 4.数据类型错误	1.检查网络连接; 2.检查服务端是否发送数据; 3.检查结束符是否匹配; 4.检查发送的数据类型
10026	socket 断开	无	无
10027	socket 连接	无	无
10028	SocketRead 数据长度与设定长度不匹配	接收到的数据个数与设定不符	发送正确个数的数据
10029	设置输入模拟模式失败		
10030	仿真输入信号失败	1.未打开仿真模式; 2.该信号不存在	1.打开仿真模式; 2.检查输入信号设置
10031	仿真 GI 信号失败	1.未打开仿真模式; 2.该信号不存在	1.打开仿真模式; 2.检查 GI 信号设置
10032	设置输出信号失败	1.该信号不存在; 2.DO 信号已被设置为系统输出	1.检查输出信号设置; 2.仿真没有被绑定为系统输出的信号
10033	设置 GO 信号失败	1.该信号不存在; 2.设定的值超出允许的范围	检查 GO 信号设置
10034	RCI 包解析错误	消息长度错误导致无法解析	请检查 RCI 指令格式
10035	RCI 参数保存成功	无	无
10036	打开 RCI 成功	无	无
10037	关闭 RCI 成功	无	无
10038	RCI 打开失败	1.IP 地址不能设置为机器人地址(192.168.0.160) 和 本 地 主 机(127.0.0.1); 2.IP 地址格式错误; 3.端口号已被占用; 4.机器人运动中; 5.控制模式非位置模式	1.请设置允许使用的 IP 地址和端口号; 2.停止机器人运动
10039	RCI 关闭失败	1.机器人运动中; 2.控制模式非位置模式	停止机器人运动并再次尝试关闭
10040	拖动失败	实际反馈与模型力矩偏差较大，不允许开启拖动。	1.检查当前工具(Tool)设置是否与实际一致，设定的工具质量质心是

			否合理; 2.查看监控窗口，检查机器人坐标系、位姿是否与实际一致; 3.确认当前机型、RD 参数与实际一致; 4.可尝试回到机械零位，对传感器清零后再拖动; 5.其他详见《拖动故障排查手册》
10041	客户端连接 RCI	无	无
10042	客户端断开 RCI	检测到客户端断开连接	请检查客户端
10043	RCI 暂不响应，请检查自动和上电状态		
10044	本体参数辨识完成，请重启机器人	无	无
10045	负载参数辨识成功	无	无
10046	更新碰撞检测参数失败		
10047	开启碰撞检测	无	无
10048	关闭碰撞检测	无	无
10049	负载辨识失败，负载超过了机器人额定负载	安装的工具或工件超过了机器人的额定负载	请使用机器人额定负载内的工具或工件
10050	负载辨识失败，负载辨识结果不正确	传感器力矩反馈异常	检查传感器力矩值
10051	关节位置限制设置失败	1.设定的角度超出机器人机械限位; 2.非手动模式; 3.机器人运动中; 4.控制模式非位置模式	1.切换到手动模式，并停止机器人运动; 2.请确认机器人的机械限位并设定该范围内的角度
10052	关节位置限制设置成功	无	无
10053	修正机器人硬限位	无	无
10054	修正机器人关节位置限制	无	无
10055	设置伺服滤波参数失败	伺服滤波器读取失败	重启机器人,如无法解决请联系技术支持
10059	此机型控制器不支持升级 3.1 及以上版本!	大负载机型指令滤波未开启	保持控制器在升级前版本，如有问题，请联系技术支持
10060	设置的最大关节速度超过限制	1.协作模式下最大关节速度可设定的范围为 0-15 度/秒; 2.非协作模式下不能超过各轴的最大轴速度	请检查并设定范围内的值
10061	设置的最大 TCP 速度超过限制	1.协作模式下最大 TCP 速度可设定的范围为 0-250; 2.非协作模式下不能超过额定最大 TCP 速度	请检查并设定范围内的值
10062	设置的关节力矩超过限制	各轴的最大关节力矩限制为 3 倍额定力矩	请检查并设定范围内的值
10063	拖出虚拟墙范围，虚拟墙失效	拖动施加外力矩过大，导致拖出虚拟墙的范围	1.扩大虚拟墙边界; 2.减小施加的拖动力矩; 3.增加虚拟墙刚度
10064	离开虚拟墙后，再次拖入虚拟墙范围，虚拟墙重新生效	离开虚拟墙后，再次拖入虚拟墙范围，虚拟墙重新生效	无
10065	同步 NTP 失败	未安装 NTP 服务；或无法和服务端	升级安装 NTP 服务；确认服务端

		同步	IP 地址正确且能同步时间
30060	关节速度超过限制		
30061	TCP 线速度超过限制		
30062	关节力矩超过限制		
30063	动量超过限制		
30064	总功率超过限制	总功率超过了机器人限制设置的值	1.降低运行速度; 2.减少负载和惯量; 3.关闭机器人限制的整机功率功能
30065	双编码器位置偏差超过限制		
30066	力矩传感器双通道数据偏差超过限制		
10067	总功率超过限制	各轴的最大功率限制为最大关节力矩(Nm)×最大关节速度(弧度/秒),其中: 1.协作模式下最大关节速度为 15 度/秒; 2.非协作模式下不能超过各轴的最大轴速度	请检查各项参数及其单位
10068	工业机型碰撞检测的触发行为不支持当前“触发暂停”模式，请改为“安全停止”		
10070	进入安全区域，停止运动	无	无
10071	进入安全区域，进入协作模式	无	无
10072	触发缩减模式并减速运动	无	无
10073	触发二级协作模式，减速运动	无	无
10074	退出缩减模式	无	无
10075	触发缩减模式	无	无
10078	RCI 运行过程中触发限制	1.关节或笛卡尔空间位置超限; 2.关节速度超限; 3.关节或电机力矩超限; 4.力矩指令和实际力矩偏差大; 5.机器人处于奇异点	请检查力矩指令和机器人初始状态
10079	触发力控模块保护,退出力控模式	见内容	1.检查力控模式下机器人状态是否正常; 2.合理设置力控保护参数
10080	开启拖动失败:关节位置超限位	机器人当前位置超出关节位置限制	将机器人运动到关节位置限制内
10081	socket 发送数据失败	网络故障原因	检查网络问题
10082	socket 接收数据超时	网络故障或者代码逻辑错误	1.检查代码逻辑; 2.检查网络另一端是否正常收发数据
10083	外部通讯作为客户端连接失败，尝试重连中	1.IP 和端口号不正确; 2.服务端设备没有正常启动; 3.设备连接异常	1.检查 ip 和端口设置是否正确; 2.重新打开服务端
10084	外部通讯作为服务端启动失败	1.ip 和端口号不正确; 2.服务端设备没有正常启动; 3.设备连接异常	1.将 ip 设为空或者 0.0.0.0; 2.端口不能设置为 0
10085	socket 创建失败，和外部通	RL 程序中的 socket 名称不可以和	创建 socket 时使用和外部通讯不

	讯名称重名	外部通讯 socket 名字同名	同的名字
10086	socket 创建失败，socket 名称超过 30 个字节	socket 名称过长	减少 socket 名称长度
10087	打开外部通讯失败：$arg	$arg	根据原因提示进行处理
10088	socket 连接	无	无
10090	设置安全区域成功	无	无
10091	设置安全区域失败	1.非手动模式; 2.机器人运动中; 3.设置安全区域行为为"触发协作模式"但未打开协作模式	1.切换到手动模式; 2.停止机器人运动; 3.如需触发协作模式，请先打开
10092	设置协作模式成功	无	无
10093	设置协作模式失败	1.非手动模式; 2.机器人已上电; 3.机器人运动中; 4.设定的参数超出限制	1.确保机器人处于手动模式，停止运动并下电; 2.检查协作模式各项参数设置
10094	设置安全监视器成功	无	无
10095	设置安全监视器失败	1.非手动模式; 2.机器人已上电; 3.机器人运动中	确保机器人处于手动模式，停止运动并下电
10096	设置碰撞检测触发动作失败	该触发行为不支持	请修改触发行为
10097	DH 参数保存成功，请重启机器人生效	无	无
10098	此版本只支持修改部分连杆参数		
10099	RD 参数保存失败，超出 RD 参数修改的限制范围	超出 RD 参数修改的限制范围	RD 参数修改的限制范围为 +/-50
10100	socket 创建 ip 和端口被占用	RL 程序，或者外部通讯 socket 中已经使用了该 ip 和端口号，不允许两个 socket 使用相同的 ip 和端口	请使用不同的 IP 地址或端口号
10101	socket ip:port 错误	RL 程序、外部通讯 socket 中已经使用了重复或错误的 IP	修改 Socket 为本机空余的 IP:PORT，或者查询网络设置
10102	socket 端口错误	RL 程序，或者外部通讯 socket 中已经使用了错误的端口	修改 socket 端口为 0~65535 的空余端口
10103	SocketSendString 失败	连接未建立或者网络错误	请检查网络连接，或使用已连接的 socket
10104	SocketSendByte 失败	连接未建立或者网络错误	请检查网络连接，或使用已连接的 socket
10105	Socket 解析数据失败	发送数据不符合规则	发送符合规则的数据
10106	快速移动失败，不处于手动模式	机器人状态错误，不处于手动模式	切换手动模式
10107	快速移动失败，程序正在运行	机器人状态错误，程序正在运行	暂停机器程序
10108	快速移动失败，机器人未上电	机器人状态错误，机器人未上电	机器人使能上电
10109	未设置起始点，Home 指令错误	未设置起始点，Home 指令错误	在设置界面>基本设置>快速调整界面设置起始点

10110	设置标定角度成功	无	无
10111	设置标定角度失败	标定角度超过限位数据	设置限位数据以内的角度值
10112	末端快速调整失败，请选择合适的调整类型	末端调整用于末端姿态微调，大范围末端调整可能计算失败	1.检查当前机器人的姿态以及工具工件参数，选择合适的末端调整类型; 2.机器人已经处于对应姿态; 3.指定姿态超出运动范围; 4.指定位姿会使机器人位于奇异点
10114	串口命名冲突，请重新输入	新建或者修改串口名时，不能和系统io，socket 名以及已有串口名称重复	检查系统 IO,socket 名和已有串口名称，重新输入新的串口名
10115	串口打开成功	无	无
10116	串口打开失败	1.串口硬件连线断开; 2.串口套接字不存在	1.请检查串口硬件连线; 2.变量列表中删除串口变量并重新创建
10117	串口关闭失败	1.串口硬件连线断开; 2.串口套接字不存在	1.请检查串口硬件连线; 2.点击pptomain 重新运行
10118	串口关闭成功	无	无
10119	串口不存在	该串口不存在	请在通信-串口设置界面创建串口
10121	串口发送 string 失败	1.串口硬件连线断开; 2.串口套接字不存在	1.请检查串口硬件连线; 2.变量列表中删除串口变量并重新创建以确定串口通信正常可用
10122	串口读取 byte 失败	读取的数据无法转换为字节类型	请检查发送的内容
10123	串口发送 byte 失败	1.串口硬件连线断开; 2.串口套接字不存在	1.请检查串口硬件连线; 2.变量列表中删除串口变量并重新创建以确定串口通信正常可用
10124	串口清空缓冲区失败	1.串口硬件连线断开; 2.串口套接字不存在	1.请检查串口硬件连线; 2.点击pptomain 重新运行
10125	串口获取缓冲区长度失败	1.串口硬件连线断开; 2.串口套接字不存在	1.请检查串口硬件连线; 2.点击pptomain 重新运行
10126	serial 接收数据超时	网络故障或者代码逻辑错误	1.检查代码逻辑; 2.检查网络另一端是否正常收发数据
10127	serial 接收数据失败	1.串口连接异常; 2.对端未及时发送数据; 3.串口两端参数配置不匹配	1.检查串口硬件连接; 2.检查对端是否发送数据; 3.检查串口两端参数配置
10128	当前 xDiagnose 较低，无法看数据	当前 xDiagnose 较低	可使用0.3.8及以上的xDiagnose版本
10129	传送带等待工件超过规定时限	1.传送带停止运动但跟踪功能依旧开启; 2.光电开关异常，无法捕捉工件触发信号	1.检测传送带是否正常运动; 2.检测光电开关是否正常捕捉到触发信号
10130	有工件超出启动窗口		
10131	工件已经超出工作区	工件超出工作区，无法继续跟踪	请调整工件位置
10132	传送带速度低于阈值	传送带停止运动或者编码器接线断开	请检查传送带是否正常运动，或者编码器接线是否断开
10133	切换热插拔模式失败	机型、控制柜或安全板固件不支持热插拔功能	1.检查当前机器人的机型为工业机型配备 XBC5 控制柜，或者为协

			作 CR 机型; 2.检查当前机器人的安全板是否为 mini 板，且固件升级为 2.0 版本; 3.检查是否正确配置 ENI 文件
10134	切换热插拔模式失败	1.安全板未能正确切换状态; 2.通信错误	请联系供应商解决
10135	切换热插拔模式成功	无	无
10136	快速移动失败,角度超出机械限位	关节角度超出机械限位	修改角度数值在机械限位内
10137	传送带工件关联两次	未 DropWObj 时，再次 WaitWobj	请检查指令，同一工件只需关联一次
10138	Waitwobj 指令运行失败	手动模式下连续运行导致	请在手动模式下单步运行，或自动模式下连续运行
10139	当前没有被关联的工件	1.没有正在被跟踪的工件; 2.没有该指定名称的传送带	请放置工件在传送带并触发光电，可在状态监控中查看工件位置是否正常
10140	摩擦力辨识结果异常,使用无摩擦模型		使用无摩擦模型，摩擦力系数置为 0.1
10141	输入的质量超过范围		输入的质量取值范围应在 0 到最大负载之间
10142	未进行空载辨识，带载辨识停止		请先进行空载辨识，后进行带载辨识
10143	运动至点位或力控辨识前设置速度失败		
10144	此机型做动力学辨识后只修改摩擦力，建议直接做摩擦力辨识		
10145	PCB3/4 轴机器人暂不支持缩减模式，触发动作实际为无动作	目前 PCB3/4 轴机器人暂不支持协作模式	
10146	当前机型不支持动力学辨识	当前机型不支持动力学辨识	若需要辨识模型，请直接做摩擦力辨识
10147	功能授权失败	见内容	请联系授权管理员
10148	用户设置的关节最大轴速度超出限制，保存不生效	见内容	请修改 hmi 上设置的最大轴速度
10149	外部通信断开，依照外部通信设置暂停程序	外部通信断开	检查外部通信连接状态
10150	外部通信断开，依照外部通信设置暂停程序并下电	外部通信断开	检查外部通信连接状态
10151	运动至点位失败	工具、工件不能同时手持或外部	选择正确的工具、工件
10152	DH 参数保存失败，超出 DH 参数修改的限制范围	超出 DH 参数修改的限制范围	DH 参数修改的限制范围为： 长度 +/-50mm,角度:+/-10°
10153	标定失败，用户坐标系错误	标定时选择的用户坐标系不存在	选择正确的用户坐标系，重新标定
10154	机器人处于安全回撤状态，	机器人处于安全回撤状态时，不允许	手动 jog 机器人安全回撤完成后，

	无法进行当前操作	进行当前操作	再重试
10200	电子铭牌不存在	1.本体没有电子铭牌; 2.电子铭牌硬件损坏; 3.电子铭牌串口配置错误	1.检查电子铭牌串口硬件连接; 2.检查电子铭牌硬件; 3.检查电子铭牌串口配置
10201	电子铭牌读取数据错误	1.电子铭牌协议地址错误; 2.电子铭牌硬件损坏	1.检查协议是否匹配; 2.检查电子铭牌硬件
10202	电子铭牌写入数据错误	1.电子铭牌协议地址错误; 2.电子铭牌硬件损坏	1.检查协议是否匹配; 2.检查电子铭牌硬件
10203	电子铭牌数据不匹配	1.更换了机器人本体; 2.更换了控制柜	确认后进行数据覆盖
10204	编码器电池电压过低	1.运行时间过长; 2.编码器电池损坏	更换编码器电池
10205	机器人运动时间超过保养期限	未保及时养	定期保养
10206	覆盖数据成功	无	无
10207	不允许覆盖电子铭牌数据	1.电子铭牌不存在; 2.机型, ID 或硬件版本不匹配	请根据日志内容检查电子铭牌
10208	不允许覆盖 RC 数据	1.控制器机型与本体不匹配; 2.电子铭牌未烧录数据或数据错误	1.选择正确的机型后，重启控制器; 2.电子铭牌重新烧录正确数据
10209	socket 通信失败	1.IP 和 port 已被外部通信使用 2.socket 名称与外部通信同名，默认 SYS_SOCKET 名称已被外部通信使用	1.更改 socket 的 port 或者 IP 2.更改 socket 命名
10210	切换精度补偿状态失败，$arg	编码器零点数据错误，切换前后零点角度相差过大	请检查编码器零点
13013	急停被触发	无	手动恢复急停
13014	安全门被打开	无	手动关闭安全门
13015	本体参数辨识开始计算	无	无
13016	程序词法或者语法检查错误	RL 程序语法错误	请检查 RL 程序
13017	程序 PP_to_main 失败	1.任务已销毁; 2.未加载工程文件; 3.缺少 main 函数	1.重新初始化任务; 2.重新加载工程; 3.检查文件中是否有 main 函数
13018	程序 PP_to_func 失败	1.任务已销毁; 2.任务未建立符号表; 3.要跳转的函数不存在	1.重新初始化任务; 2.检查函数中是否有语法错误，重新建立符号表; 3.检查函数是否存在
13019	程序 PP_to_line 失败	1.任务已销毁; 2.任务未建立符号表; 3.程序 PP_to_line 跳转只允许在同一个 PROC/FUNC 内部	1.重新初始化任务; 2.检查函数中是否有语法错误，重新建立符号表；3. 请检查是否在同一个 PROC/FUNC 内部进行跳转
13020	所有 RL 任务都已经停止，请检查错误信息或者点击 PPToMain	1.单循环模式任务结束; 2.任务出错停止	1.检查程序是否存在逻辑错误; 2.点击 pptomain 重新运行
13021	基坐标系错误，设置失败	控制器无法解析设置基坐标系的命令,可能是由于控制器版本与 HMI 不兼容	请检查控制器的版本与 HMI 是否适配

13022	下一步选择未加载的任务		在工程的任务列表勾选对应任务并且执行 pptomain
13030	RSC 检测到机器人功率超限	机器人功率超限，或 RSC 配置机器人功率过低，或 RSC 故障	检查 RSC 的机器人功率限制范围是否合理，或关闭 RSC 机器人功率限制
13031	RSC 检测到机器人动量超限	机器人动量超限，或 RSC 配置机器人动量限制过低，或 RSC 故障	检查 RSC 的机器人动量限制范围是否合理，或关闭 RSC 机器人动量限制
13032	RSC 检测到机器人肘部力或 TCP 力超限	机器人 TCP 力超限，或 RSC 配置机器人肘部力或 TCP 力过低，或 RSC 故障	检查 RSC 的机器人肘部力或 TCP 力限制范围是否合理，或关闭 RSC 机器人肘部力&TCP 力限制
13033	RSC 检测到机器人 TCP 速度超限	机器人 TCP 速度超限，或 RSC 配置机器人 TCP 速度过低，或 RSC 故障	检查 RSC 的机器人 TCP 速度限制范围是否合理，或关闭 RSC 机器人 TCP 速度限制
13034	RSC 检测到机器人位置超限	机器人位置超限，或 RSC 故障	检查 RSC 的机器人位置限制范围是否合理，或关闭 RSC 机器人位置限制
13035	RSC 检测到机器人姿态超限	机器人姿态超限，或 RSC 故障	检查 RSC 的机器人姿态限制范围是否合理，或关闭 RSC 机器人姿态限制
13036	RSC 检测到机器人关节碰撞力超限	机器人关节碰撞力超限，或 RSC 故障	检查 RSC 的机器人关节碰撞力限制范围是否合理，或关闭 RSC 机器人关节碰撞力限制
13037	RSC 检测到机器人关节功率超限	机器人关节功率超限，或 RSC 故障	检查 RSC 的机器人关节功率限制范围是否合理，或关闭 RSC 机器人关节功率限制
13038	RSC 检测到机器人关节位置超限	机器人关节位置超限，或 RSC 故障	检查 RSC 的机器人关节位置限制范围是否合理，或关闭 RSC 机器人关节位置限制
13039	RSC 检测到机器人关节速度超限	机器人关节速度超限，或 RSC 故障	检查 RSC 的机器人关节速度限制范围是否合理，或关闭 RSC 机器人关节速度限制
13040	RSC 检测到机器人关节力矩超限	机器人关节力矩超限，或 RSC 故障	检查 RSC 的机器人关节力矩限制范围是否合理，或关闭 RSC 机器人关节力矩限制
13041	RSC 检测到机器人关节通信异常	机器人关节通信异常，或 RSC 故障	重启机器人尝试、或联系厂商检查硬件故障
13042	RSC 检测到机器人关节工作异常	机器人关节工作异常，或 RSC 故障	重启机器人尝试、或联系厂商检查硬件故障
13043	RSC 与主控之间出现通信错误	RSC 与主控之间出现 FSoe 通信错误	重启机器人尝试、或联系厂商检查硬件故障
13044	控制器与 RSC 同步 SDO 数据失败	控制器与 RSC 同步 SDO 数据失败	可尝试重新配置，或者联系厂商检查硬件
13045	控制器与 RSC 同步 SDO 数		

	据成功		
13046	RSC 拖动时超出安全限制	拖动时速度过快，超过了 250mm/s，已触发安全停止	
13047	RSC 初始化参数检查不一致	RSC 的基本安全参数与主控制器的参数不一致	硬重启机器人或者控制柜
13048	RSC 与主控的通信进入安全数据状态(0x08)	RSC 与主控的通信进入安全数据状态(0x08)	
13049	RSC 关节位置限制参数非法	设定的角度超出机器人机械限位	请确认机器人的机械限位并设定该范围内的角度
13050	RSC 安全 Home 位置参数非法	设定的角度超出机器人机械限位	请确认机器人的机械限位并设定该范围内的角度
13051	安全门关闭		
13052	安全门开启，无法继续运行 RL 程序	安全门开启	请复位安全门后重试
13053	RSC 触发过速保护	机器人运动过程触发过速保护	
13057	RSC 检测到 RSC 检测到通道短路故障		请检查接线是否有误，故障恢复后需断电重启
13058	RSC 检测到安全 DI 信号双通道信号不一致	RSC 安全 DI 信号双通道信号不一致	请检查和修复对应 DI 信号的接线，然后重置为 0 后，再同时置为 1 来恢复故障
13059	RSC 检测到双机通讯故障		若一直出现，建议更换 RSC
13060	RSC 检测到 MCU 地址故障		若一直出现，建议更换 RSC
13061	RSC 检测到输出数据不一致故障	主机与从机计算输出的数据信息不一致	若一直出现，建议更换 RSC
13062	RSC 检测到电压超限	供电电源电压不在 20~30V 之间	检查 RSC 电源电压
13063	RSC 检测到算法库故障		
13064	RSC 检测到 Bamboo 未正常运行		断电重启，若重启后仍然异常，建议更换 RSC
13065	RSC 检测到 FSOE 内部通讯故障		重新上电测试，若仍然异常，建议更换 RSC
13066	RSC 检测到 FSOE 外部通讯故障		重新上电测试，若仍然异常，建议更换 RSC
13067	RSC 检测到从站通讯异常		
13068	RSC 检测到从站未处于 FSoE 数据状态		
14001	网络连接已断开	无	无
14002	网络连接已关闭	无	无
14003	网络连接已建立	无	无
14004	网络连接已开启监听	无	无
14005	网络连接正在重连中	无	无
14006	网络连接正在重连中	无	无
14010	绑定寄存器不允许写操作	寄存器为系统绑定寄存器，不允许在 RL 中执行写操作	请使用其它寄存器，或解除该寄存器的绑定

14011	总线设备打开失败，未扫描到对应的 Ethercat 从站	没有配置 Ethercat 从站，或者设备链接异常	导入并配置 Ethercat 从站
14012	只读寄存器不允许写操作	寄存器为只读寄存器，不允许执行写操作	请使用只写属性寄存器，或修改寄存器读写属性
14020	打开总线设备成功	无	无
14021	打开总线设备失败	见内容	请检查总线设备配置，并确保对应设备已正确连接
14022	关闭总线设备成功	无	无
14023	串口已被总线设备占用	该串口已经在 modbus RTU 总线设备中使用	关闭总线设备，或使用别的串口
14030	profinet 总线模型配置错误	该槽位数据模型选择和 PLC 不一致	重新配置该槽位的数据模型
14031	寄存器保持文件错误	寄存器保持文件 crc 校验失败，文件损坏	删除寄存器保持文件
14032	总线设备打开失败，SDO 设置失败	设置 SDO 值失败	检查 SDO 初始化设置是否符合设备要求
14501	无法设置系统 DO	DO 信号为系统 IO 信号，无法被设置	请使用其它输出信号，或在系统 IO 中取消该信号的绑定
14502	GO 信号设置失败	设置值超出信号的有效值范围	请检查数值
14503	AO 信号设置失败	设置值超出信号的有效值范围	请检查数值
14504	PulseDO 设置失败，时间超出范围	脉冲时间设置错误，范围 [0.001, 2000] S	重新输入脉冲时间
14505	PulseReg 设置失败，时间超出范围	脉冲时间设置错误，范围 [0.001, 10] S	重新输入脉冲时间
14510	输出信号加载失败	映射的物理端口与其它信号或系统输出信号发生冲突	1.检查输出端口是否被占用; 2.重新配置系统 IO
14511	系统输入信号设置失败，存在冲突	违反系统输入配置规则，可能原因: 1.配置了重复的系统 IO; 2.对应 IO 已经被 RL 工程的 IO 占用; 3.IO 信号不存在	1.配置互相独立的系统 IO; 2.修改 RL 工程占用的系统 IO; 3.打开或创建 IO 设备
14512	系统输出信号设置失败，存在冲突	违反系统输出信号配置规则，可能原因: 1.配置了重复的系统 IO; 2.对应 IO 已经被 RL 工程的 IO 占用; 3.IO 信号不存在	1.配置互相独立的系统 IO; 2.修改 RL 工程占用的系统 IO; 3.打开或创建 IO 设备
14521	Ethercat IO 设备初始化失败，未扫描到对应的 Ethercat 从站	没有配置 Ethercat 从站，或者设备链接异常	导入并配置 Ethercat 从站
14530	安全板扩展 IO 设备状态发生变化	扩展 IO 设备硬件损坏或链接异常	请检查安全板扩展 IO 设备硬件状态及链接状态
14531	安全板扩展 IO 设备状态发生变化	新接入了扩展 IO 设备	无
14532	IO 设备状态发生变化	IO 设备配置变化	无
14533	IO 设备状态发生变化	IO 设备配置变化	无
14534	IO 设备状态发生变化	IO 设备通信异常	无

14535	导入的寄存器变量使用了当前机器中不存在的总线设备	总线设备未在当前机器上配置	手动编辑寄存器变量，更改设备名称为当前机器中存在的总线设备
14536	使用了空信号	使用了一个空的信号	使用正常名称的信号
14537	使用的信号是无效类型	使用了一个无效类型的信号	使用正确的信号类型
14538	信号映射到不存在的设备	信号绑定到了不存在的设备	将信号绑定到正确的设备
14539	信号映射到被禁用的设备	信号映射的设备是被禁用的	1.绑定其他正常的设备 2.禁用的设备重新启用
14540	组信号端口设置错误	组信号端口设置错误	1.组信号结束端口号应大于起始端口号; 2.信号使用的端口号不要超过设备端口范围
14541	组信号端口超限	组信号端口数过多	更改组信号端口数不超过 32
14542	信号端口设置错误	信号端口设置错误	1.信号起始端口和结束端口应保持一致; 2.信号使用的端口号不要超过设备端口范围
14543	信号映射的端口被占用	信号使用的端口已被占用	1.配置互相独立的系统 IO; 2.修改 RL 工程占用的系统 IO; 3.打开或创建 IO 设备
15000	执行拖动路径回放失败	执行拖动路径回放，缓冲区无可用的拖动数据	重新录制拖动轨迹
15001	拖动数据保存失败	缓冲区无可用的拖动数据	请重新录制拖动轨迹
15002	执行拖动路径回放失败	拖动路径记录被意外删除	重新录制拖动轨迹或者重新导入路径
15003	执行拖动路径回放失败，拖动文件数据错误	回放路径来自其他类型机器人	导入正确的拖动回放路径或者重新录制轨迹
15004	拖动数据保存失败	拖动回放序列化数据存储硬盘文件失败	请重新录制拖动轨迹，或重启机器人
15005	拖动数据保存成功	无	无
15006	执行拖动路径回放	无	无
15007	开始记录拖动回放数据	无	无
15008	停止记录拖动回放数据	无	无
15009	停止执行拖动路径回放	无	无
15010	记录路径数据失败：路径点过少	拖动路径点过少	请重新录制拖动轨迹
15011	记录路径数据失败：关节角超限	拖动轨迹中存在关节超过关节位置限制	请重新录制拖动轨迹
15012	记录路径数据失败：关节速度超限	拖动轨迹中存在关节速度超限	请重新录制拖动轨迹
15013	记录路径数据失败：停止录制时速度不为零	结束路径录制时,速度不为零	待机器人停止后再点击结束录制
15014	记录路径数据失败：开始录制时速度不为零	开始录制时速度不为零	请确保机器人从停止状态开始录制
15015	记录路径数据失败：电机未上电	电机未上电	机器人上电

15016	记录路径数据成功	无	无
15017	执行拖动路径回放，读取路径失败	拖动路径数据解析失败，路径数据可能被篡改	重新导入路径或者重新录制轨迹
15018	路径回放，设置速率超限	设置回放速率过大	调低路径回放速率
15019	路径回放，回放路径点中有点位超关节位置限制	回放路径点位超关节位置限制	调整当前关节位置值
15020	路径回放失败，导轨开启时不允许路径回放	导轨开启时不允许路径回放	关闭导轨后进行路径回放
17001	寄存器读数据失败	1.该寄存器不存在; 2.寄存器与变量类型不符; 3.数组下标超出范围	请检查寄存器设置
17002	寄存器写数据失败	1.该寄存器不存在; 2.寄存器与变量类型不符; 3.数组下标超出范围	请检查寄存器设置
17003	modbus 输入寄存器读取失败		
17004	加载寄存器配置失败	寄存器配置文件(registers.json)解析失败，原因见内容	请尝试重新配置寄存器，或抹除配置
17005	modbus 通讯失败，modbus 链接未建立		
17006	modbus 配置保存成功	无	无
17007	外部通讯配置保存成功	无	无
17008	modbus 配置解析失败	modbus 寄存器配置有错误	检查 modbus 寄存器配置
17009	加载总线设备配置文件失败	总线设备配置文件(fieldbus_device.json)解析失败，原因见内容	请尝试重新配置寄总线设备，或抹除配置
17100	cclink 开启失败，未检索 cclink 网关模块	没有配置 cclink 网关模块，或者设备链接异常	导入并配置 cclink 网关模块
17101	cclink 已开启	无	无
17102	cclink 开启失败		
17103	cclink 已关闭	无	无
17104	与 CC-LINK IEF Basic 主站通讯成功	无	无
17105	与 CC-LINK IEF Basic 主站通讯断开	无	无
17200	控制器中不存在任意一组符合规则的伺服参数文件	控制器中至少有一组伺服参数才能使用伺服参数切换的各项功能	无
17201	上电失败，伺服参数切换过程中禁止上电	上电失败，伺服参数切换过程中禁止上电	无
17202	上电状态下，禁止切换伺服参数	上电状态下，禁止切换伺服参数	无
17203	伺服参数切换成功	无	无
17204	伺服参数切换不匹配	无	无
17205	传送带视觉 socket，ip 不能为空或“0.0.0.0”，端口不	无	无

	能为空		
17206	快速调整自定义位姿超过限位限制	无	无
17207	有一个被拒绝的外部输入	无	无
17300	RL 程序定制停止，不能继续运行	无	可 pptomain 、 pptoline 或 pptofunc 后运行程序
17301	xService 链接断开	1、HMI 和控制器的连接意外断开；2、SDK 客户端主动断开；	1、在 HMI 操作断开并重新连接控制器；
17310	伺服参数获取失败		
17311	伺服参数获取成功		
17312	伺服参数下载失败		
17313	伺服参数下载成功		
17314	请检查配置文件版本,当前配置文件缺少安全停止参数字段	当前配置文件缺少安全停止参数字段	请前往 sw 网站下载对应的最新版本配置文件,并将该配置文件升级至控制器
17315	修改 IP 失败	见内容	输入合适的 IP
17316	导轨零点标定失败	1.未处于手动模式; 2.机器人运动中; 3.控制模式非位置模式	请确保机器人处于手动模式并停止运动
17317	导轨零点标定成功	无	无
17318	软急停状态不允许上电	当前有只读寄存器绑定了 ctrl_soft_estop 功能码且为低电平，不允许上电	将绑定了 ctrl_soft_estop 功能码的寄存器复位
17319	软急停被触发	无	通过寄存器恢复软急停
17320	运动指令缓存状态不匹配	见内容	按照内容 PPtoMain 或重置缓存
17500	驱动器处于严重错误状态，不能清除报警、上电	驱动器处于严重错误状态，可能存在硬件相关故障。	1、尝试控制柜断电重启清除异常；2、联系技术支持进行排查处理
17400	OpcUA 变量读数据失败	1.该变量不存在; 2.OpcUA 变量与参数类型不符	请检查 RL 指令及 OpcUA 变量设置
17401	OpcUA 变量写入数据失败	1.该变量不存在; 2.OpcUA 变量与参数类型不符	请检查 RL 指令及 OpcUA 变量设置
17402	软标定失败	1.未处于手动模式; 2.机器人运动中; 3.控制模式非位置模式	请确保机器人处于手动模式并停止运动
17403	单轴零点标定成功		
17404	单轴软标定成功		
17405	软标定成功		
17406	单轴软标定失败	1.未处于手动模式; 2.机器人运动中; 3.控制模式非位置模式	请确保机器人处于手动模式并停止运动
17407	开启导轨失败，请先关闭安全区域		
17408	安全区域打开失败	导轨开启时，不能开启安全区域	
17507	传送带开始跟踪失败	无	无
17508	传送带停止跟踪失败	无	无
17509	传送带标定失败，X 方向点位	请检查标定流程	无

	个数不为 3		
17510	传送带标定失败，传动比为 0	1、传送带未开启，2、编码器值异常	无
17511	安全停止参数设置不合理	安全停止参数数值异常,数值为 0	请修改安全停止控制参数为合理值
17512	传送带跟踪过程中，禁止执行普通运动指令	无	无
17513	传送带跟踪过程中停止，无法继续运行	无	需 pptomain 后，重新运行程序
17520	OPC UA 服务启动失败	1.端口设置错误或已被占用; 2.配置错误	1.使用空闲端口且端口不能为 0; 2.修改错误配置
17521	HMI 链接	无	无
17522	HMI 上下电	无	无
17523	HMI 上下电	无	无
17524	PP_To_Main	无	无
17525	PP_To_Line	无	无
17526	PP_To_Func	无	无
17527	加载工程	无	无
17528	启动程序	无	无
17529	程序下一步	无	无
17530	程序上一步	无	无
17531	手动模式切换成功	无	无
17532	自动模式切换成功	无	无
17600	关闭最大力矩监控		
17700	无	无	
30067	关节功率超过限制	无	无
30068	TCP 角速度超过限制	无	无
30069	肘部线速度超过限制	无	无
30070	肘部角速度超过限制	无	无
30071	动量超过限制	无	无
30072	工具位姿超出限制	无	无
18000	机器人出共享区，解除共享区占用	无	无
18001	机器人进入共享区，占用共享区	无	无
18002	机器人已经进入被占用共享区，最大能力减速停止	无	无
18003	机器人即将进入被占用共享区，减速暂停，等待共享区释放	无	无
18004	机器人已经进入禁止区，最大能力减速停止	无	无
18005	机器人即将进入禁止区，规	无	无

	划停止		
18006	共享区失效，机器人继续运行	无	无
18007	切换缩减模式时，当前关节轴角度已超出缩减关节位置	无	请重新设置合理的缩减关节位置
18008	不可改变数组维度	无	请勿修改数组维度
18009	控制器日志损坏,使用备份日志	可能是断电/异常/重启导致	无需修复

17.1.2 3XXXX

错误码	错误描述	故障可能原因	修复方法
31001	EtherCAT 主站配置参数错误	EtherCAT 配置文件不匹配	重新导入配置文件，并重新启动控制系统
31002	读取 EtherCAT 授权文件失败	授权文件不存在或读写权限错误	重新导入授权文件
31003	EtherCAT 授权失败	EtherCAT 未授权	请检查 EtherCAT 授权码，尝试在 EtherCAT 授权界面重新授权，授权成功后重启机器人生效
31004	配置 EtherCAT 主站失败	EtherCAT 配置文件不匹配，导致主站配置失败	重新导入配置文件，并重新启动控制系统
31005	DC 配置失败	EtherCAT 配置文件不匹配，导致配置失败	重新导入配置文件，并重新启动控制系统
31006	DCM 配置失败	EtherCAT 配置文件不匹配，导致配置失败	重新导入配置文件，并重新启动控制系统
31007	扫描总线失败	EtherCAT 配置文件不匹配，导致配置失败	重新导入配置文件，并重新启动控制系统
31008	配置从站个数与扫描到的从站个数不匹配	1. EtherCAT 网络配置与实际网络拓扑不一致，导致配置失败;2.EtherCAT 网络从站设备硬件故障	1. 重新导入配置文件，并重新启动控制系统 2 联系厂家支持
31009	启动 EtherCAT 总线失败	EtherCAT 总线启动过程出错，有部分从站无法正确切换至 OP 模式，导致总线失败	尝试重启控制系统，联系厂家支持
31010	内部轴伺服初始化失败	驱动器内部发生故障	伺服驱动器之间的连线中断
31011	初始化 IO 从站失败	1. 工控机与通用 IO 模块之间的硬件连线中断 2. 通用 IO 模块与安全 IO 模块之间的硬件连线中断	根据失败原因，重新配置 IO 信号后重启控制系统
31012	初始化末端从站失败	工控机与通用末端从站模块之间的硬件连线中断，或末端从站硬件故障	检查硬件连接，或者更换末端从站硬件尝试
31013	安全板从站初始化失败	工控机与安全板从站模块之间的硬件连线中断	检查硬件连接
31014	与从站设备通讯出现异常	1、工控机与从站模块之间的通讯中断，2、从站设备出现故障	1、确认 ethercat 调试线是否从机器人端拔出 2、检查硬件连接，并

			重启控制系统 3、请联系技术支持
31015	伺服出现告警	伺服驱动器出现告警，告警具体原因需要根据伺服上报的错误码查找对应的驱动器手册	1.根据错误码查找对应的驱动器手册，根据手册指导处理; 2.联系技术支持
31016	伺服告警已清除	无	无
31017	Ethercat 未授权，试用时间结束	Ethercat 没有授权，只能使用一个小时	请联系技术支持索要 Ethercat 授权码，并在 EtherCAT 授权界面授权，授权成功后，重启机器人生效
31018	从站设备与控制器连接异常	1、从站设备与控制器连接线出现故障 2、从站出现故障	请检查从站设备之间的连接线出现故障 2、请联系技术支持
31019	CClink 网关从站初始化初始化失败	硬件连接异常或硬件故障	检查硬件连接
31020	EthercatIO 从站模块未适配	使用了未曾适配过的 EthercatIO 从站模块，控制器对当前 IO 从站的信息未知，用户需自行确保正确配置和使用此 IO 模块	由于未曾适配的 IO 模块的 PDO 设置是未知的，直接使用存在一定风险，请谨慎使用。一般情况下，对于 64 路以内的数字 IO 模块，可以直接使用；对于模拟 IO 模块，由于控制器对于其模拟量类型、量程、精度参数都未知，各设备厂商的 PDO 设置也可能不同，不建议直接使用，请联系技术支持适配后再使用
31021	初始化安全版从站失败	eni 文件配置的安全板固件版本与实际固件版本不一致	检查 eni 文件的安全板版本是否与安全板固件版本是否一致
31022	末端从站输入电流信号过载	末端从站通道 AI 设置为电流模式时，请确保实际输入为电流且不能超过限制参数，否则将可能损坏硬件	请断开对应通道的输入信号，并且进入 xpanel 配置界面将模式设置为电压模式
31023	末端从站输入电流信号过载	已经完成 xpanel 设置，过载已恢复	如需继续使用电流模式，请确保输入电流不过载
31024	xPanel 末端从站硬件测试失败	xPanel 末端从站硬件测试流程错误、硬件连接错误、或者硬件故障	请确认 xPanel 测试流程是否正确，如无法解决，请联系研发排查问题
31025	安全板类型配置错误	控制器配置的安全板类型与实际使用的安全板类型不一致	检查控制器配置的安全板类型与实际使用的安全板类型是否一致
31026	内部轴伺服初始化失败	主站配置的 ENI 文件中的伺服驱动器版本与扫描到的伺服驱动器配置不一致	更换主站 ENI 文件，确认使用正确的 ENI 文件
31027	无法正确读取主站 ENI 文件	系统中没有找到主站 ENI 文件	重新升级当前机型对应的 ENI 文件
31028	EtherCAT 扫描总线失败，扫描从站信息与配置 ENI 文件中的从站信息不一致	主站的 ENI 文件中配置的从站信息或拓扑结构与扫描的 EtherCAT 从站不一致	更换并升级正确的 ENI 文件
31029	EtherCAT 扫描总线失败，有从站接线交叉	EtherCat 从站接线错误	请检查 EtherCat 从站的接线是否正确
31030	外部轴伺服初始化失败	1.错误开启了未连接的外部轴，如	根据错误原因，关闭错误开启的外

		导轨 2.驱动器内部发生故障	部轴或检查与外部轴的连线是否中断
31031	外部轴伺服初始化失败	主站配置的 ENI 文件中的伺服驱动器版本与扫描到的伺服驱动器配置不一致	更换主站 ENI 文件，确认使用正确的 ENI 文件
31032	安全板 RSC 的 ESI 文件和控制器版本不匹配	控制器版本为 3.0，安全板 RSC 的 ESI 和控制器版本不匹配，RSC 的部分安全功能将无法使用	需要升级安全板 RSC 的 ESI 文件和主控的 ENI 文件
32001	上电失败	伺服有故障、STO 未闭合、上电过程中中止等原因	确认操作方式、检查伺服、清除伺服报警后再进行上电
32002	上电失败,STO 未接通	无	无
32003	控制器主站与设备通讯线程阻塞	由于软件原因导致控制器与从站设备通讯出现异常	1、控制器软重启 2、联系技术支持
32004	EtherCAT 线程被阻塞,调度超时	系统内部错误	重启控制系统
32005	EtherCAT 线程被阻塞，连续超时超过 5000 次	系统内部错误	重启控制系统
32006	位置指令被拒绝	即将发给伺服的位置指令发生大的跳变，产生的速度超过电机最大速度，可能会产生危险，为保险起见，执行下电操作	重新上电运行
32007	Ethercat 线程事件执行超时	内部告警	内部状态记录，可以忽略
32008	写入伺服零点出错	仅支持 CR 关节，确认是否为 CR 机器人，确认固件是否正确	无
32009	写入伺服零点成功		
32010	编码器电池电压低警告	编码器电池电压不足	请及时更换编码器电池
32011	编码器电池电压低警告清除		
35001	驱动器短路	输出到输出，输出到地，内部 PWM 桥错误	排查线路问题，消除短路现象
35002	驱动器过温	驱动器内部温度达到设定值	使驱动器温度降到设定值以下
35003	驱动器过压	母线电压超出指定的电压限制	母线电压恢复到指定的电压范围
35004	驱动器欠压	母线电压在指定的电压以下	母线电压恢复到指定的电压范围
35005	编码器电机过温	电机过温开关状态显示过温错误	将温度开关恢复到正常状态
35006	编码器反馈错误	1.驱动器内部 5V 输出过流; 2.旋转变压器或者模拟编码器没有接线; 3.电平超出误差范围; 4.增量编码器差分信号没有接线	1.编码器电源恢复到指定的电压范围; 2.反馈信号恢复到指定的电平范围内; 3.差分信号连接完好
35007	驱动器相位错误	基于编码器的相位角与 HALL 的开关状态不相符。该错误仅发生在无刷电机被配置为正弦给定时。当旋转变压器反馈或者是 HALL 纠正功能被关闭时，该错误不会发生。	基于编码器的相位角与 HALL 的开关状态一致
35008	驱动器达到电流限	电机过载或者线路异常	1.排查电机是否过载; 2.检查线路; 3.联系技术支持

35009	驱动器达到电压限	1.速度给定过高; 2.电机异常	1.检查速度给定是否异常; 2.排查电机是否异常; 3.联系技术支持
35010	驱动器正限位接通	驱动器超正限位	恢复驱动器限位
35011	驱动器负限位接通	驱动器超负限位	恢复驱动器限位
35012	驱动器跟随超差	超出用户设置的跟随误差	排查上层指令是否有问题，排查机器人状态
35013	位置计数器到达	驱动器内部错误	尝试重启，如恢复不了请联系技术支持
35014	适用于故障无其他紧急情况	驱动器内部错误	尝试重启，如恢复不了请联系技术支持
35015	节点错误	驱动器内部错误	尝试重启，如恢复不了请联系技术支持
35016	驱动器命令错误	1.此项错误报错不太准，可忽略; 2.PWM 或其他命令信号不存在	恢复输入信号
35100	驱动器直流母线欠压	伺服实时检测到母线电压小于设定的欠压保护阈值,可能由于母线供电电压不足	1.检查机器人供电电压是否正常; 2.检查驱动器电容是否正常; 3.更换伺服驱动器
35101	驱动器直流母线过压	伺服实时检测到母线电压高于设定的过压保护阈值,可能由于：1.母线供电电压过高; 2.机器人减速过快; 3.底座电源处理异常	1.检查机器人供电电压; 2.检查机器人减速度; 3.检查机器人电源管理回路
35102	驱动器电机过流	伺服实时检测电流矢量的幅值大于设定的安全保护阈值,可能由于电机在运行过程中失步急停会造成电流飙升	1.检查电机三相线及 48v 动力线是否正确连接; 2.检查电机偏角是否正确
35103	驱动器电机过载	关节电机连续超过额定转矩保护阈值范围,可能由于伺服闭环控制,如果电机实际位置不能跟踪上给定位置,造成转矩电流偏大,持续时间达到一定程度会报错	1.检查电机三相线及抱闸线是否正确连接; 2.看检查电机偏角是否正确
35104	驱动器电机磁通过限	判断励磁电流幅值是否小于设定值，超出设定值报错。弱磁控制需要改变励磁电流达到更高的转速，目前低压伺服适配的电机母线电压足够，不需要进行弱磁，因此已屏蔽此报错	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本
35105	驱动器失速报警	关节速度跟踪误差大于设定阈值且持续 1s 以上会报错。启动阶段报失速，电机为正常运转起来，需要重新找偏角。电机运行过程中报失速，需要检查硬件配置及软件版本及参数	1.检查电机三相线及抱闸线是否正确连接; 2.检查电机偏角是否正确; 3.检查硬件配置; 4.检查伺服软件版本
35106	驱动器位置超差	关节位置跟踪误差超过设定阈值,需要检查硬件及软件是否为正确的	1.检查线束是否正确连接; 2.检查关节编码器是否正常; 3.更换关节硬

		配置及版本	件,电机,减速器,驱动器; 4.确认主站发指令是否合理
35107	驱动器零电流报警	零电流值需要超过设定阈值,未使能时电流为采集异常	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35108	驱动器 A 相过流	1.电机的 A 相电流超过大于驱动器电流采样范围; 2.电机堵转; 3.负载过大或者转速过快; 4.关节机构阻力大	1.先检查线束是否链接正确; 2.检查电机偏角是否正确; 3.检查上使能时抱闸是否正确打开; 4.负载大的时候，降低转速查看故障是否消失; 5.观察运行时的故障关节是否有阻力过大的异常噪音
35109	驱动器 B 相过流	1.电机的 B 相电流超过大于驱动器电流采样范围; 2.电机堵转; 3.负载过大或者转速过快; 4.关节机构阻力大	1.先检查线束是否链接正确; 2.检查电机偏角是否正确; 3.检查上使能时抱闸是否正确打开; 4.负载大的时候，降低转速查看故障是否消失; 5.观察运行时的故障关节是否有阻力过大的异常噪音
35110	驱动器 C 相过流	1.电机的 C 相电流超过大于驱动器电流采样范围; 2.电机堵转; 3.负载过大或者转速过快; 4.关节机构阻力大	1.先检查线束是否链接正确; 2.检查电机偏角是否正确; 3.检查上使能时抱闸是否正确打开; 4.负载大的时候，降低转速查看故障是否消失; 5.观察运行时的故障关节是否有阻力过大的异常噪音
35111	驱动器 IGBT 过流	电机电流超过设定好的阈值; 2.电机堵转; 3.负载过大或者转速过快; 4.关节机构阻力大	1.先检查线束是否链接正确; 2.检查电机偏角是否正确; 3.检查上使能时抱闸是否正确打开; 4.负载大的时候，降低转速查看故障是否消失; 5.观察运行时的故障关节是否有阻力过大的异常噪音
35112	驱动器对象溢出		
35113	驱动器映像溢出		
35114	驱动器 H 线程超时		
35115	驱动器 M 线程超时	伺服软件调度异常,规定时间内未执行完程序,可能由于固件版本或参数版本不是最新版	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35116	驱动器 L 线程超时		
35117	驱动器 C 线程超时		
35118	驱动器中断死机	伺服软件调度异常,规定时间内未执行完程序,可能由于固件版本或参数版本不是最新版	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35119	驱动器主任务死机		
35120	驱动器硬件非法		
35121	驱动器调试版超时		
35122	驱动器对象 ID 超时		

35123	驱动器线程 ID 出错		
35124	驱动器 PV 变量 ID 错		
35125	驱动器 VV 变量 ID 错		
35126	驱动器 VU 变量 ID 错		
35127	驱动器数据类型错		
35128	驱动器数据对象已存在		
35129	驱动器数据队列不存在		
35130	驱动器数据队列满		
35131	驱动器数据队列空		
35132	驱动器应用版本错		
35133	驱动器参数校验错		
35134	驱动器参数号错		
35135	驱动器 PN 参数 ID 错		
35136	驱动器 UN 参数 ID 错		
35137	驱动器 FN 参数 ID 错		
35138	驱动器 SPI 通讯错	DSP 与参数存储器通信异常。可能由于固件版本或参数版本不是最新版；或伺服驱动器硬件异常	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35139	驱动器 E2P 溢出	参数存储器参数读取过多。可能由于固件版本或参数版本不是最新版；或伺服驱动器硬件异常	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35140	驱动器动态内存空	伺服软件运行逻辑出错。可能由于固件版本或参数版本不是最新版	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35141	驱动器指令码错	伺服软件运行逻辑出错。可能由于固件版本或参数版本不是最新版	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35142	驱动器指令长度错	伺服软件运行逻辑出错。可能由于固件版本或参数版本不是最新版	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器
35143	驱动器 Meter 通道号错		
35144	驱动器 OSCI 通道号错		
35145	驱动器静态模式错	伺服运行模式错误。可能由于伺服固件和参数不是最新版；或伺服参数未烧写成功	更新伺服驱动器至最新的固件版本，更换伺服驱动器
35146	驱动器动态模式错	伺服运行模式错误。可能由于伺服固件和参数不是最新版；或伺服参数未烧写成功	更新伺服驱动器至最新的固件版本，更换伺服驱动器
35147	驱动器模式改变错	伺服运行模式错误。可能由于伺服固件和参数不是最新版；或伺服参数未烧写成功	更新伺服驱动器至最新的固件版本，更换伺服驱动器
35148	驱动器 IPM 报警	可能由于伺服固件和参数不是最新版	更新伺服驱动器至最新的固件版本，更换伺服驱动器

35149	驱动器模块过热报警	关节温度超过安全可靠工作的温度范围	进行相应的冷机处理
35150	驱动器 STO 开关没接通	伺服一直在检测 STO 信号电平，与设置逻辑不符，故报错。可能由于机器人使能把手未生效	1.按下手动使能把手; 2.重新插拔 STO 线; 3.更换 STO 线，更换伺服驱动器
35151	编码器错误（0x7371）	QEP 类型编码器报错，ABZ 线断开。可能因为伺服未升级至最新的固件和参数	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本
35152	编码器错误（0x7372）		
35153	编码器错误（0x7373）		
35154	编码器错误（0x7374）		
35155	编码器错误（0x7375）		
35156	编码器错误（0x7376）		
35157	编码器错误（0x7377）		
35158	编码器错误（0x7378）	编码器芯片报错: Biss2 通信异常	1.检查关节编码器线; 2.更换编码器; 3.更换编码器线; 4.更换伺服驱动器
35159	编码器错误（0x7379）	编码器芯片报错: Biss1 通信异常，找不到编码器解码芯片	1.检查关节电机侧编码器线; 2.更换编码器; 3.更换编码器线; 4.更换伺服驱动器
35160	编码器错误（0x737A）	1.编码器通信 CRC 错误; 2.编码器通信异常	1.重新插拔编码器线; 2.更换编码器; 3.更换编码器线; 4.更换伺服驱动器
35161	编码器错误(0x737B)	编码器通信异常：编码器 CRC 校验通过，但编码器芯片报错	1.重新插拔编码器线; 2.更换编码器; 3.更换编码器线; 4.更换伺服驱动器
35162	编码器错误（0x737C）	1.单个编码器硬件故障; 2.编码器电池耗尽。	1.如果单个编码器报错，且重启无法恢复，则需要联系售后维修。2.所有轴同时报错误，联系售后更换编码器电池。
35163	TZ 触发		
35164	内部错误(0xFF91)		
35165	上使能电流环给定为 0 (0xFFA2)		
35166	不能跟踪主站位置指令 (0x8612)		
35201	二级编码器 5v 过电流		CDHD 能提供最大为 250mA 的电流给二级编码器。核对编码器是否短路，检查编码器绘制是否超过当前限制
35202	过电流	1.驱动器或电机故障; 2.机器人负载超过限值; 3.发送碰撞等	检查电机连接是否短路，检查在电流回路中是否过冲
35203	电机折返	1.驱动器或电机故障; 2.机器人负载超过限值; 3.发送碰撞等	检查驱动器-电动机的规格。如果该电动机的容量过小（功率过小）可

			能发生此故障。
35204	驱动折返		检查驱动机-电动机规格，如果该驱动器的容量过小（功率过小）可能发生此报警，检查换向角是否正确（即，整流是否平衡）
35205	电流传感器补偿无效	驱动器故障	重启故障，如果故障依然存在，或许需要修理驱动，请联系技术支持
35206	电机相位断开	驱动器故障	检查电机相位的接线
35207	输出过流检测	驱动器故障	验证数字输出的正确接线，确保输出电路没有短路。
35208	过电压		检查是否需要应用再生电阻
35209	欠压故障		检查主交流电压电源是否连接驱动器并打开。欠压限制可以由 UVTHRESH 命令读取
35210	再生过电流		增加再生电阻的值
35211	STO 故障		检查 STO 连接器(P1)是否正确连接
35212	Vbus 测量电路失败		重启故障，如果故障依然存在，或许需要修理驱动，请联系技术支持
35213	总线交流电源断开		请检查接线,或联系技术支持
35214	再生电阻过载		检查再生电阻器特性是否适合该应用
35215	集成电源模块过热		检查环境温度是否超过驱动器规范。否则联系技术支持
35216	控制板过热		检查环境温度是否超过驱动器规范。否则联系技术支持
35217	温度传感器失败		重启电源。如果问题仍然存在，请联系技术支持
35218	功率级过热		检查环境温度是否超过驱动器规范。否则联系技术支持
35219	电动机过热故障		检查该驱动器是否正确配置（用 THERMODE,THERMTYPE,THERMTHRESH 和 THERMTIME）,如果必要，检查电动机温度传感器是否正确连接到驱动上。如果驱动正确配置并连线，核对是否该电动机规格过小。
35220	内部电源供应超出范围		或许需要修复驱动。请联系技术支持
35221	5v 超出范围		或许发生在停电时。如果发生在其他情况下，请联系技术支持。
35222	电源 EEPROM 故障		联系技术支持
35223	控制 EEPROM 故障		联系技术支持
35224	CAN 供应故障 CAN Supply Fault		或许需要修复驱动，请联系技术支持。

35225	自检失败		联系技术支持
35226	参数内存校验失败		重新配置驱动器，或下载参数集，并保存参数。如果问题依然存在，请联系技术支持。
35227	闪存写入失败		联系技术支持
35228	现场总线速度超过限制		启用驱动器并发送有效的位置命令
35229	未配置		设置驱动参数后执行 CONFIG
35230	FPGA 配置失败		联系技术支持
35231	电机安装失败		检查相位和电机接线，确保选中正确反馈类型，检查 MOTORSETUPST 提示
35232	相位查找失败		检查是否设置好针对此应用的电机反馈类型和相位查找参数
35233	FPGA 版本不匹配		更新 FPGA 版本或驱动版本
35234	紧急停止发布		关闭输入
35235	现场总线版本不匹配		确保正确的版本已经下载到驱动上
35236	ESI 版本不匹配		确保正确的版本已经下载到驱动上
35237	BiSS-C 编码器内部错误		参考 BiSS-C 编码器使用手册
35238	HIPERFACE 编码器数据错误		使用驱动器软件输入命令 HSAVE 1。
35239	ESI 厂商不匹配		确保正确的厂商数据已经下载到驱动上
35240	索引行中断		核对驱动器是否配置为和索引信号(使用 MENCTYPE)一同工作，并检查索引信号是否连接
35241	电源制动器打开负载		
35242	电源制动器短路		
35243	停转问题		移除失速状态，并注意防止失速条件。
35244	二次反馈指数中断		核对驱动器是否配置为使用二级编码器的索引信号，并检查索引信号是否连接
35245	二次反馈 A/B 断线		核对是否所有来自二级编码器的信号都正确连接到了驱动
35246	脉冲和方向输入断线		检查所有的 P&D（脉冲和方向）输入信号是否正确连接到驱动器
35247	电源制动故障		更换电机制动器
35248	电动机失控状态检测		核对 MPHASE 设置，启用并改善相位发现过程。
35249	反馈通信错误，编码器通信断开	1.本体到控制柜线路连接不良 2.编码器或者驱动器是否故障	检查反馈装置是否正确接线。检查是否选择正确编码器种类（MENCTYPE）
35250	尼康编码器运行故障		核对反馈装置是否正确连线。检查

			是否选择正确编码器种类（MENCTYPE）
35251	Tamagawa 初始失败		检查编码器是否正确连线
35252	A/B 断线		请参阅正弦编码器和解析器诊断一节。检查是否来自主反馈设备的所有信号都正确连接到驱动器
35253	无效 Halls		检查霍尔信号是否都正确连接。但转动电机时，请阅读霍尔状态(用 HALLS)来查看哪个信号没有连接。若果反馈类型是 Tamagawa，检查反馈接线是否正确
35254	绝对编码器电池低电压	电池属于易耗品，正常更换，请联系厂家购买	更换电池，然后重设驱动器。如果当驱动器打开时更换电池，位置信息被保留
35255	锁相环同步失败		核对控制器同步信号，核对电缆接线或连接
35256	编码器模拟频率太高		检查用于设置等效编码器输出的参数。如果使用正弦编码器，请检查 ENCOUTRES 参数设置
35257	TamagawaAbs 操作故障		检查电池电量及反馈接线。确保电机在编码器初始化时不会高速移动
35258	自定义绝对编码器操作故障		核对电池电量和反馈接线。确保电机在编码器初始化时没有高速运动
35259	差分 Halls 线断开		确保 HALLSTYPE 与使用的霍尔传感器匹配（单端或差分）。检查是否所有来自差分霍尔传感器的信号都正确连接到了传感器
35260	编码器相位错误		将 MENCAQBFILT 设置为 0 以删除 A 和 B 信号的过滤器。如果问题仍然存在，可能是因为编码器故障。
35261	AqB 换相故障		如果在运动开始不久发生故障，请检查 MENCRES 设置。如果在一段时间后发生故障，可能是因为 EMI 噪音。改善安装，确保接地。确保屏蔽层连接在反馈和电机电缆上。
35262	SensAR 编码器故障		使用 SRVSNSINFO 命令来识别错误
35263	正弦反馈通信失败		检查传递给 EnDat 的数据和时钟信号连接正确。电缆必须被屏蔽
35264	A/B 超出范围		请参阅正弦编码器和解析器诊断一节。检查正弦和余弦信号的幅度
35265	Sankyo 绝对编码器故障		检查电池电量及反馈接线。确保电机在编码器初始化时不会高速移动
35266	正弦编码器正交故障		检查反馈设备的连线。确保正确的

			编码器类型（MENCTYPE）被选中
35267	正弦/余弦校准无效		重新执行正弦/余弦校准过程
35268	反馈 5v 过电流		CDHD 可以提供 250mA 的电流给主编码器。检查编码器是否短路，检查编码器绘制是否超过当前限制
35269	解析器初始化失败		检查分解器接线盒增益值
35270	Endat2X 反馈错误		重设编码器，包括关闭编码器电源
35271	现场总线电缆断开连接		重新建立控制器与驱动器之间的连接
35272	现场总线控制命令丢失		清除故障并允许控制器发送新指令。
35273	CAN 心跳缺失		重新连接主从机，然后重启驱动器
35274	驱动锁定		联系技术支持
35275	EtherCAT 丢包		确保 EtherCAT 主站（控制器）在定义的时间内发送包（由主设备）
35276	力矩反馈超限		
35277	不稳定的电流回路		检查并修改当前的控制器设置
35278	超速过度		核对 VLIM 设置是否匹配需求。使用速度环调整，检查过度超调。
35279	超过最大速度误差		改变驱动调谐以改善速度跟踪，或增加 VEMAX 来允许更大的位置误差。
35280	超过最大定位误差		改变驱动调谐以改善位置跟踪，或增加 PEMAX 来允许更大的位置误差。
35281	二次反馈位置不匹配		增加 SFBPETHRESH,STBPETIME.SFBPEMAX,或提高位置调优。
35282	过度 PE 值		检查调整
35283	CAN/EtherCAT 状态不可操作		确保当驱动被禁用时，控制器不切换到较低的通讯状态。
35284	内部错误		联系技术支持
35285	电机读板失败		重新连接反馈装置。确保电机类型铭牌数据存在。
35286	保存和重启电源要求		保存，然后重启驱动电源
35287	实时过载故障		联系技术支持
35288	PFB 关闭校验无效		如果应用需要，将机器返回原位
35289	PFB 关闭数据不匹配		如果应用需要，将机器返回原位
35290	没有 PFB 关闭数据		如果应用需要，将机器返回原位
35291	脉冲频率过高		从控制器降低齿轮传动脉冲的频率
35301	驱动器短路	i. 驱动器 U、V、W 输出存在短接现象; ii. 驱动器受干扰导致 DI 信号异常，此为误报现象; 1. 地线	i. 排查驱动器 U、V、W 接线(比如断开电机动力线缆后再观察驱动器是否仍报短路故障，须在电机抱

		未接好; 2. 电流环调节器参数设置不合适,导致电流振荡引发干扰; iii. 驱动器损坏(比如 IGBT 短路，电流检测电路异常).	闸断开的前提下进行以保证机械安全）；ii. 用万用表检查驱动器 IGBT，确认是否短路; iii. 规范布线，尤其是地线; iv. 调节电流环参数; v. 更换驱动器.
35302	驱动器输出对地短路	i. 驱动器 U、 V、 W 输出对地存在短路现象; ii. 驱动器损坏（比如电流检测电路异常）。	i. 排查驱动器 U、 V、 W 接线; ii. 更换驱动器
35303	编码器数据异常	i. Hiperface 编码器故障; ii. 编码器接线错误; iii. 驱动器内部 AD 校准参数异常。	i. 更换编码器; ii. 检查编码器接线，并确保接线正确; iii. 更换驱动器。
35304	转子定位错误	i. 位置环、速度环、电流环调节器参数设置不合理; ii. 电机参数设置错误; iii. 参数 0x20D2 设置过小; iv. 参数 0x2003 设置错误; v. 驱动器外围接线错误（比如电机动力线，电机编码器线）；vi. 驱动器内部电路异常; vii. 参数 0x2207 静态平衡力矩补偿值设置不合适。	i. 检查驱动器外围接线，并确保接线正确。ii. 调整驱动器位置环、速度环、电流环调节器参数，并确保电机参数设置正确; iii. 重新进行转子位置补偿角检测; iv. 增大参数 0x20D2 设定值; v. 更换驱动器; vi. 减小参数 0x2207 静态平衡力矩补偿值的设定值。
35305	电机抱闸异常	i. 电机抱闸本身出现异常; ii. 电机在高速运行时突然伺服 OFF; iii. 伺服参数 0x2233， 0x20D2 设置过小; iv. 电机抱闸线路存在短路现象。	i. 更换电机抱闸; ii. 增大参数 0x2233 和 0x20D2; iii. 检查电机抱闸线路。
35306	EtherCAT PDO 配置错误	i. PDO 配置错误。	i. 更正 EtherCAT PDO 配置。
35307	编码器内部通信异常	a. 编码器发生故障; b. 电机编码器接线异常（比如断线，未采用屏蔽双绞线，与电机动力线耦合在一起）；c. 驱动器地线未可靠连接; d. 驱动器周围存在强干扰源。	a. 检查电机编码器接线并确保接线规范正确; b. 编码器线缆，电机动力线缆增加磁环; c. 可靠的连接驱动器地线; d. 更换电机编码器; e. 移除驱动器周围强干扰源，或者驱动器与周围强干扰源独自供电; f. 驱动器动力输入电源增加进线滤波器。
35308	编码器类型变更	i. 更换了编码器类型。	i. 重上电或者软复位驱动器。
35309	驱动器过流 U	i. 电流环调节器参数设置不合理，导致电流控制振荡; ii. 电机参数设置错误; iii. 驱动器内部电流采样电路异常。	i. 调整电流环调节器参数; ii. 正确设置电机参数; iii. 更换驱动器。
35310	驱动器过流 V	i. 电流环调节器参数设置不合理，导致电流控制振荡; ii. 电机参数设置错误; iii. 驱动器内部电流采样电路异常。	i. 调整电流环调节器参数; ii. 正确设置电机参数; iii. 更换驱动器。
35311	驱动器过流 W	i. 电流环调节器参数设置不合理，导致电流控制振荡; ii. 电机参数设	i. 调整电流环调节器参数; ii. 正确设置电机参数; iii. 更换驱动器。

		置错误; iii. 驱动器内部电流采样电路异常。	
35312	直流母线过压	i. 驱动器动力输入电源电压过大; ii. 电机快速停止时的能耗制动能量过大; 1. 电机停止减速度过大; 2. 能耗制动电阻接线错误; 3. 能耗制动电阻阻值过大; iii. 驱动器内部电压采样电路异常; iv. 驱动器内部能耗制动电路异常。	i. 调整驱动器动力输入电源至允许范围; ii. 减小电机停止减速度; iii. 检查能耗制动电阻接线并确保接线正确; iv. 适当减小能耗制动电阻阻值（阻值不能低于允许的最小值），增大能耗制动电阻功率。
35313	控制电源欠压	i. 24V 控制电源异常; ii. 24V 控制电源接线错误，比如接线不良; iii. 24V 控制电源负载过大; iv. 驱动器内部电路异常。	i. 检查 24V 控制电源接线并确保接线可靠; ii. 检查 24V 控制电源负载，确保 24V 控制电源的容量能满足所有工况下的负载消耗。iii. 更换 24V 控制电源; iv. 更换驱动器。
35314	驱动器持续过载	i. 电机负载过大; 1. 实际机械负载过大; 2. 机械负载存在卡死现象; 3. 电机抱闸未释放; ii. 电机加减速时间设置过小; iii. 驱动器内部电流采样电路异常; iv. 驱动器抱闸电路异常。	i. 减小电机实际机械负载; ii. 增大电机加减速时间; iii. 检查机械负载传动方式，确保无卡死等现象; iv. 检查电机抱闸接线确保接线可靠; v. 更换电机; vi. 更换驱动器。
35315	编码器接线错误	i. 对于 CDA8 V1 版产品， 编码器线端连接器引脚 8 和引脚 15 未短接; ii. 对于 CDA8 V2 版产品和 CDR 系列产品， 编码器线端连接器引脚 6 和引脚 8 未短接; iii. 编码器线缆接线不良。	i. 对于 CDA8 V1 版产品， 短接编码器线端连接器引脚 8 和引脚 15；对于 CDA8 V2 版产品和 CDR 系列产品，短接编码器线端连接器引脚 6 和引脚 8，并确保编码器线缆接线可靠。
35316	CPU 过载	i. 驱动器运行受到干扰; ii. 驱动器内部电路异常; iii. DriveStarter 数据采集过大。	i. 规范驱动器外围布线，增加抗干扰措施; ii. 更换驱动器; iii.关闭部分 DriveStarter 数据采集通道。
35317	驱动器输出缺相	i. 驱动器 U、 V、 W 输出存在断线或接线不良等现象; ii. 电机阻抗过大; iii. 驱动器内部电流采样电路异常。	i. 检查电机 U、 V、 W 接线并确保接线可靠; ii. 更换电机（或关闭驱动器输出缺相检测功能）；iii. 更换驱动器。
35318	驱动器瞬时过载	i. 电机负载过大; ii. 驱动器内部温度采样电路异常; iii. 驱动器运行环境温度超出了允许工作范围; iv. 驱动器运行受到干扰（比如不同步）；v. 电机加速度及减速度设置过大，加减速时间设置过小。	i. 减小电机实际机械负载; ii. 规范驱动器外围布线，增加抗干扰措施; iii. 降低环境温度，比如改善机柜散热条件; iv. 更换驱动器。
35319	编码器外部通信发送异常	a. 电机编码器接线异常（比如断线，未采用屏蔽双绞线，与电机动力线耦合在一起）；b. 驱动器地线未可靠连接; c. 驱动器周围存在强	a. 检查电机编码器接线并确保接线规范正确; b. 编码器线缆，电机动力线缆增加磁环; c. 可靠的连接驱动器地线; d. 移除驱动器周围强

		干扰源。	干扰源，或者驱动器与周围强干扰源独自供电; e. 驱动器动力输入电源增加进线滤波器。
35320	编码器外部通信接收异常	a. 电机编码器接线异常（比如断线，未采用屏蔽双绞线，与电机动力线耦合在一起）; b. 驱动器地线未可靠连接; c. 驱动器周围存在强干扰源。	a. 检查电机编码器接线并确保接线规范正确; b. 编码器线缆，电机动力线缆增加磁环; c. 可靠的连接驱动器地线; d. 移除驱动器周围强干扰源，或者驱动器与周围强干扰源分开供电; e. 驱动器动力输入电源增加进线滤波器。
35321	驱动器硬件过流	i. 机械负载存在卡死现象或堵转现象; ii. 转子补偿角设置不正确; iii. 编码器接线错误; iv. 电流环调节器参数设置不合理，导致电流控制振荡; v. 电机参数设置错误（线电阻、线电感、反电动势等）; vi. 驱动器内部电流检测电路异常; vii. 驱动器抱闸电路损坏，无 24V 输出; viii. 电机抱闸损坏; ix. 电机负载过大或电机加速度、减速度设置过大，加减速时间设置过小; x. 0x60B2 转矩补偿值或 0x2207 静态平衡力矩补偿值给定不合理。	i. 检查机械负载传动方式，确保无卡死等现象; ii. 重新检测转子补偿角; iii. 检查电机编码器接线并确保接线规范正确; iv. 调整电流环调节器参数; v. 正确设置电机参数; vi. 更换驱动器; vii.更换电机; vii. 检测机械负载传动方式，确保无卡死等现象；确认机械设计是否合理；优化电机加减速，延长加减速时间; ix. 优化上位控制器动力学模型，优化给定值 0x60B2 转矩补偿值；或重新设定合理的 0x2207 静态平衡力矩补偿值。
35322	驱动器抱闸电路异常	i. 驱动器抱闸输出短路; ii. 驱动器抱闸输出电流过大导致过热; iii. 驱动器抱闸输出断路; iv. 驱动器内部检测电路异常。	i. 检查驱动器抱闸输出接线并确保接线正确可靠; ii. 更换驱动器。
35323	驱动器旋变电路异常	i. 驱动器内部旋变电路异常; ii. 驱动器旋变参数设置与实际旋变不匹配。	i. 正确设置驱动器旋变参数; ii. 更换驱动器。
35324	控制模式设定错误	i. 伺服使能时，控制器设定了驱动器不支持的控制模式（各产品支持的控制模式，详见对象字典 0x6502）。	i. 伺服使能前，控制器先设定正确的控制模式。
35325	输入缺相故障	i. 驱动器动力输入电源接线不良; ii. 驱动器伺服参数“功率回路设定”选择为三相输入，但实际动力电源输入为单相; iii. 前端使用电子变压器，电子变压器谐波异常，伺服无法识别。	i. 检查驱动器动力输入电源接线并确保接线可靠; ii. 正确设置伺服参数“功率回路设定”; iii. 在伺服驱动器前端加装滤波器。
35326	直流母线欠压	i. 驱动器动力输入电源电压过低; ii. 驱动器内部电压采样电路异常; iii.伺服参数 0x202C 设置错误，220V 供电设置为 380V 供电; iv.	i. 调整驱动器动力输入电源至允许范围; ii. 更换驱动器; iii.正确设置 0x202C 驱动参数; iv. 检查驱动器动力输入电源线接线。

		驱动器动力输入电源线断开。	
35327	逆变功率模块过热	i. 电机负载过大; ii. 驱动器内部温度采样电路异常; iii. 驱动器运行环境温度超出了允许工作范围。	i. 减小电机实际机械负载; ii. 降低环境温度，比如改善机柜散热条件; iii. 更换驱动器。
35328	能耗制动过载	i. 电机频繁进行快速停止操作导致能耗制动能量过大; ii. 伺服参数“能耗制动电阻阻值”和“能耗制动电阻功率”设置错误。	i. 正确设置伺服参数“能耗制动电阻阻值”和“能耗制动电阻功率”；ii. 改变电机运行工况，避免电机频繁进行快速停止操作。比如延长电机停止时间。
35329	电机持续过载	i. 电机负载过大; 1. 实际机械负载过大; 2. 机械负载存在卡死现象; 3. 电机抱闸未释放; ii. 电机加减速时间设置过小; iii. 电机参数设置错误; iv. 驱动器内部电流采样电路异常; v. 驱动器抱闸电路异常; vi. 电机选型错误，功率过小（比如大功率驱动器，带载小功率电机长; 时间满载高速运行）。	i. 减小电机实际机械负载; ii. 增大电机运行时的加减速时间; iii. 检查机械负载传动方式，确保无卡死等现象; iv. 检查电机抱闸接线确保接线可靠; v. 检查电机参数，确保电机参数设置正确（比如电机额定电流，电机热时间常数）。vi. 更换大容量电机; vii. 更换驱动器。
35330	整流功率模块过热	i. 电机负载过大; ii. 驱动器内部温度采样电路异常; iii. 驱动器运行环境温度超出了允许工作范围	i. 减小电机实际机械负载; ii. 降低环境温度，比如改善机柜散热条件。iii. 更换驱动器。
35331	电机 U 相瞬时过载	i. 电机负载过大; 1. 实际机械负载过大; 2. 机械负载存在卡死现象或堵转现象; 3. 电机抱闸未释放; ii. 电机加减速时间设置过小; iii. 转子补偿角设置不正确; iv. 电机参数设置错误。v. 驱动器内部电流采样电路异常; vi. 驱动器抱闸电路异常; vii. 电机选型偏小; viii. 电机动力线某相接触不良或脱。	i. 减小电机实际机械负载; ii. 增大电机运行时的加减速时间; iii. 检查机械负载传动方式，确保无卡死等现象; iv. 重新检测转子补偿角; v. 检查电机抱闸接线确保接线可靠; vi. 检查电机参数，确保电机参数设置正确（比如电机额定电流，电机快速过载保护阈值，电机快速过载保护时间）；vii. 更换大容量电机; viii. 更换驱动器; ix. 检查电机动力线接线是否可靠。
35332	电机 V 相瞬时过载	i. 电机负载过大; 1. 实际机械负载过大; 2. 机械负载存在卡死现象或堵转现象; 3. 电机抱闸未释放; ii. 电机加减速时间设置过小; iii. 转子补偿角设置不正确; iv. 电机参数设置错误。v. 驱动器内部电流采样电路异常; vi. 驱动器抱闸电路异常; vii. 电机选型偏小; viii. 电机动力线某相接触不良或脱。	i. 减小电机实际机械负载; ii. 增大电机运行时的加减速时间; iii. 检查机械负载传动方式，确保无卡死等现象; iv. 重新检测转子补偿角; v. 检查电机抱闸接线确保接线可靠; vi. 检查电机参数，确保电机参数设置正确（比如电机额定电流，电机快速过载保护阈值，电机快速过载保护时间）；vii. 更换大容量电机; viii. 更换驱动器; ix. 检查电机动力线接线是否可靠。
35333	电机 W 相瞬时过载	i. 电机负载过大; 1. 实际机械负载	i. 减小电机实际机械负载; ii. 增大

		过大; 2. 机械负载存在卡死现象或堵转现象; 3. 电机抱闸未释放; ii. 电机加减速时间设置过小; iii. 转子补偿角设置不正确; iv. 电机参数设置错误。v. 驱动器内部电流采样电路异常; vi. 驱动器抱闸电路异常; vii. 电机选型偏小; viii. 电机动力线某相接触不良或脱。	电机运行时的加减速时间; iii. 检查机械负载传动方式，确保无卡死等现象; iv. 重新检测转子补偿角; v. 检查电机抱闸接线确保接线可靠; vi. 检查电机参数，确保电机参数设置正确（比如电机额定电流，电机快速过载保护阈值，电机快速过载保护时间）；vii. 更换大容量电机; viii. 更换驱动器; ix. 检查电机动力线接线是否可靠。
35334	电源单元模块通讯异常	i. 电源单元模块与电机模块之间的控制线缆连接不良; ii.电机抱闸线或外部 24V（STO）对地短路造成电源模块与电机模块之间控制端口烧坏。	i. 检查电源单元模块与电机模块之间的控制线缆连接并确保接线可靠; ii. 检查电机抱闸线或外部 24V（STO）对地是否有短路; iii.更换驱动器。
35335	硬件 STO1 触发	i. STO1 触发或接线不良。	i. 检查 STO1 接线并确保接线可靠; ii. 确认 STO1 回路（如急停开关）未处于触发状态。
35336	硬件 STO2 触发	i. STO2 触发或接线不良。	i. 检查 STO2 接线并确保接线可靠; ii. 确认 STO2 回路（如急停开关）未处于触发状态。
35337	STO 配线异常	i. STO1/STO2 接线不良。	i. 检查 STO1/STO2 接线并确保接线可靠。
35338	驱动器外部故障	i. 其它轴发生了故障。ii. 驱动器内部电路异常。	i. 检查其它轴，并确保其它轴无故障; ii. 此故障可通过修改伺服参数“故障操作开关”来禁止检测; iii. 更换驱动器。
35339	位置跟随误差过大	i. 电机机械负载存在卡死现象导致电机无法运动; ii. 上位机位置目标值规划加速度过高; iii. 伺服参数 0x6065 和伺服参数 0x6066 设置过小; iv. 驱动器调节器参数设置不合理导致位置跟踪性能不理想; v. 驱动器内部电路异常。	i. 检查机械负载传动方式，确保无卡死等现象; ii. 适当降低上位机位置目标值规划加速度; iii. 适当增大伺服参数 0x6065 和伺服参数 0x6066 设定值; iv. 优化调节器参数以提高位置跟踪性能; v. 更换驱动器。
35340	位置控制溢出	i. 位置实际值或位置目标值超出了允许的最大范围。	i. 执行“编码器多圈清零”命令，并确保电机运行范围不超出允许的最大范围; ii. 若需要在大范围内运行电机，可通过伺服参数“位置控制开关”使能无限位置控制模式 。
35341	速度跟随误差过大	i. 电机机械负载存在卡死现象导致电机无法运动; ii. 伺服参数 0x20A3 和伺服参数 0x20A4 设置过小; iii. 驱动器调节器参数设置不合理导致速度跟踪性能不理想;	i. 检查机械负载传动方式，确保无卡死等现象; ii. 适当增大伺服参数 0x20A3 和伺服参数 0x20A4 设定值; iii. 优化调节器参数以提高速度跟踪性能; iv. 更换驱动器。

		iv. 驱动器内部电路异常。	
35342	控制周期参数设置错误	i. EtherCAT 通讯周期，位置控制周期，速度控制周期设置不合理。	i. 正确设置 EtherCAT 通讯周期，位置控制周期，速度控制周期。
35343	写 EEPROM 失败	i. 驱动器内部电路异常; ii. 驱动器受到干扰	i. 更换驱动器; ii. 驱动器重新上电，加强驱动器抗干扰措施。
35344	寻原点失败	i. 寻原点参数设置不合理（对象 0x6098， 0x6099,0x609A）；ii. 寻原点启动时，电机已处于限位开关触发状态; iii. 寻原点过程中切换至非 HM 模式。	i. 正确设置寻原点参数（对象 0x6098， 0x6099,0x609A）；ii. 寻原点启动时，确保电机未处于限位开关触发状态。
35345	EtherCAT 总线指令非法	i. EtherCAT 通讯状态机与控制字时序配合错误。	i. 上位机正确处理 EtherCAT 通讯状态机与控制字时序。
35346	DriveStarter 通讯异常	i. 驱动器调试线缆断开或接触不良。ii. 驱动器调试串口通讯受到干扰。	i. 检查驱动器调试线缆接线并确保连接可靠; ii. 更换隔离型串口调试线缆; iii. 加强调试线缆的抗干扰措施，比如增加磁环，调试电脑可靠接地，调试电脑与驱动器分开供电。
35347	EtherCAT 总线通讯异常	i. EtherCAT 通讯受到干扰; ii. EtherCAT 网线断开或接触不良; iii. 上位机实时性不够; iv. 上位机 EtherCAT 主站底层 DC 同步机制与驱动器需求不匹配; v. 驱动器内部电路异常。	i. 优化 EtherCAT 通讯布线，加强抗干扰措施，比如使用超 5 类屏蔽网线，控制器保证可靠接地等; ii. 检查 EtherCAT 网线连接确保连接可靠; iii. 更换实时性更强的上位机，或延长 EtherCAT 通讯周期; iv. 适当增大伺服参数 0x20D3 设定值; v. 修改上位机 EtherCAT 主站底层 DC 同步机制，确保上位机下发 RxPDO 数据比 DC 同步信号至少提前 130us; vi. 更换驱动器。
35348	位置硬超限	i. 限位开关输入触发。	i. 检查限位开关状态并确保限位开关未触发。
35349	正向软限位	i. 位置实际值超出了伺服参数 0x2004 和伺服参数 0x2005 设定阈值。	i. 适当增大伺服参数 0x2004 和伺服参数 0x2005 设定值; ii. 将电机运行到伺服参数 0x2004 和伺服参数 0x2005 规定的范围内; iii. 若不想使用该功能，可通过伺服参数“位置控制开关”，禁止软限位检测功能。
35350	负向软限位	i. 位置实际值超出了伺服参数 0x2004 和伺服参数 0x2005 设定阈值。	i. 适当增大伺服参数 0x2004 和伺服参数 0x2005 设定值; ii. 将电机运行到伺服参数 0x2004 和伺服参数 0x2005 规定的范围内; iii. 若不想使用该功能，可通过伺服参数“位置控制开关”，禁止软限位

			检测功能。
35351	上电位置偏差过大	i. 驱动器掉电后，电机位置发生了偏移; ii. 对于带电池的电机编码器，未外接电池或电池欠电压。	i. 对于带电池的电机编码器，确保已接入电池且电池电压正常; ii. 若不想使用该功能，可将伺服参数 0x200E 设定为 0，禁止上电位置偏差过大检测功能。
35352	上电位置控制溢出	i. 对于带电池的电机编码器，未外接电池或电池欠电压; ii. 驱动器断电之前的控制模式为无限位置控制模式，或速度模式，或转矩模式，且位置已超出允许的范围。	i. 驱动器执行编码器多圈清零命令后重新上电; ii. 无限位置控制模式下，若不想使用该功能，可通过修改伺服参数“位置控制开关”，禁止上电位置控制溢出检测功能。
35353	编码器电池欠电压故障	i. 编码器未外接电池或电池接线不良; ii. 编码器电池欠电压。	i. 检查编码器电池接线并确保接线可靠; ii. 更换电池; iii. 若连接为松下和多摩川编码器， 个别版本需执行编码器多圈清零命令; iv. 若不想使用该功能，可通过修改伺服参数 0x2009 禁止编码器电池欠电压检测功能。
35354	电机超速	i. 驱动器调节器参数设置不合理导致速度跟踪超调较大; ii. 编码器接线不良; iii. 编码器数据传输受到干扰; iv. 编码器损坏; v. 驱动器内部电路异常。	i. 优化调节器参数以提高速度跟踪性能; ii. 检查编码器线缆连接情况并确保接线可靠; iii. 加强编码器线缆的抗干扰措施，比如增加磁环，使用屏蔽双绞线缆，可靠接地等; iv. 更换编码器; v. 更换驱动器。
35355	电压限幅位置跟随误差过大	i. 电机机械负载存在卡死现象导致电机无法运动; ii. 上位机位置目标值规划加速度过高; iii. 伺服参数 0x6065 和伺服参数 0x6066 设置过小。iv. 驱动器调节器参数设置不合理导致位置跟踪性能不理想; v. 驱动器动力输入电源电压过低; vi. 驱动器内部电路异常。	i. 检查机械负载传动方式，确保无卡死等现象; ii. 适当降低上位机位置目标值规划加速度; iii. 适当增大伺服参数 0x6065 和伺服参数 0x6066 设定值 ; iv. 优化调节器参数以提高位置跟踪性能; v. 确保驱动器动力输入电源电压处于规定范围内; vi. 更换驱动器。
35356	编码器超速故障	i. 驱动器调节器参数设置不合理导致速度跟踪超调较大; ii. 编码器接线不良; iii. 编码器数据传输受到干扰; iv. 编码器损坏; v. 驱动器内部电路异常; vi.伺服上使能时，有外力旋转电机轴。	i. 优化调节器参数以提高速度跟踪性能; ii. 检查编码器线缆连接情况并确保接线可靠; iii. 加强编码器线缆的抗干扰措施，比如增加磁环，使用屏蔽双绞线缆，可靠接地等; iv. 更换编码器; v. 更换驱动器; vi. 检查电机轴端机械，并确保电机轴不受重力或机械外力。
35357	位置规划运行错误	i. 位置规划参数设置不合理，比如位置目标值，规划目标减速度（0x6084） 。	i. 正确设置位置规划参数。
35358	多轴同步异常	i. 驱动器内部电路异常。	i. 更换驱动器。

35359	EtherCAT 总线同步异常	i. 伺服参数 0x20D3 设置不合理; ii.EtherCAT 主站同步模式配置错误。	i. 正确设置伺服参数 0x20D3; ii. EtherCAT 主站正确配置同步模式。
35360	EEPROM 版本变更	i. 升级了新的驱动器固件。	i. 驱动器重新上电。
35361	电机过载告警	i. 电机负载过大; 1. 实际机械负载过大; 2. 机械负载存在卡死现象; 3. 电机抱闸未释放; ii. 电机加减速时间设置过小; iii. 电机参数设置错误; iv. 驱动器内部电流采样电路异常; v. 驱动器抱闸电路异常。	i. 减小电机实际机械负载; ii. 增大电机运行时的加减速时间; iii. 检查机械负载传动方式，确保无卡死等现象; iv. 检查电机抱闸接线确保接线可靠; v. 检查电机参数，确保电机参数设置正确（比如电机额定电流，电机热时间常数）；vi. 更换大容量电机。vii. 更换驱动器。
35362	转速限制告警	i. 速度模式下，速度目标值超出了电机最高转速; ii. 位置模式下，规划速度超出了电机最高转速; iii. 伺服参数设置不合理。	i. 减小速度目标值或速度规划值; ii. 更换最高转速更大的电机; iii. 根据实际情况重新设置参数 0x607F 最大规划速度。
35363	直流母线欠压告警	i. 驱动器动力输入电源电压过低; ii. 驱动器内部电压采样电路异常。	i. 调整驱动器动力输入电源至允许范围; ii. 更换驱动器。
35364	不支持设定控制模式	i. (伺服使能时，删掉)控制器设定了驱动器不支持的控制模式（各产品支持的控制模式，详见对象字典 0x6060,0x6502，）；ii. 控制器与伺服通讯建立后，未指定控制模式。	i. 伺服使能前，控制器先设定正确的控制模式。
35365	更改了重上电有效参数	i. 修改了重上电有效的伺服参数	i. 驱动器重新上电
35366	编码器电池欠电压告警	i. 编码器未外接电池或电池接线不良; ii. 编码器电池欠电压; iii. 编码器电池线正负接反或者破皮对地短路。	i. 检查编码器电池接线并确保接线可靠; ii. 更换电池; iii. 若连接为松下或多摩川编码器， 个别版本需执行编码器多圈清零命令; iv. 若不想使用该功能，可通过修改伺服参数 0x2009 禁止编码器电池欠电压检测功能。
35367	驱动器内部告警	i. 驱动器内部电路异常	i. 更换驱动器; ii. 联系售后技术支持。
35368	机械原点未标定	i. 发生了编码器电池欠电压故障，且伺服参数 0x2009.Byte3 设定为"检出编码器电池低电压故障并提示原点未标定"; ii. 发生了上电位置偏差过大故障，且用户判定机械原点已丢失; iii. 电机带单圈绝对式编码器，且驱动器未成功执行寻原点命令。	i. 驱动器执行回原点操作。
35369	编码器外部通信接收告警	a. 电机编码器接线异常（比如断线，未采用屏蔽双绞线，与电机动力线耦合在一起）；b. 驱动器地线	编码器外部通信接收告警

		未可靠连接; c. 驱动器周围存在强干扰源。	
35370	编码器外部通信发送告警	a. 电机编码器接线异常（比如断线，未采用屏蔽双绞线，与电机动力线耦合在一起）; b. 驱动器地线未可靠连接; c. 驱动器周围存在强干扰源。	a. 检查电机编码器接线并确保接线规范正确; b. 编码器线缆，电机动力线缆增加磁环; c. 可靠的连接驱动器地线; d. 移除驱动器周围强干扰源，或者驱动器与周围强干扰源独自供电; e. 驱动器动力输入电源增加进线滤波器。
35371	编码器内部通信告警	编码器内部通信告警	a. 检查电机编码器接线并确保接线规范正确; b. 编码器线缆，电机动力线缆增加磁环; c. 可靠的连接驱动器地线; d. 更换电机编码器; e. 移除驱动器周围强干扰源，或者驱动器与周围强干扰源独自供电; f. 驱动器动力输入电源增加进线滤波器。
35372	软限位告警	i. 位置实际值或者位置目标值超出了伺服参数 0x2004 和伺服参数 0x2005 设定阈值。	i. 适当增大伺服参数 0x2004 和伺服参数 0x2005 设定值; ii. 将电机运行到伺服参数 0x2004 和伺服参数 0x2005 规定的范围内; iii. 减小位置目标设定值，确保其位于伺服参数 0x2004 和伺服参数 0x2005 规定的范围内; iv. 若不想使用该功能，可通过伺服参数“位置控制开关”，禁止软限位检测功能。
35373	AD 校正系数无效告警	i. 驱动器尚未进行 AD 校正。	i. 重置驱动器 AD 校正系数
35374	位置规划参数异常告警	i. 位置规划参数设置不合理。	i. 正确设置位置规划参数。
35375	上电位置偏差过大告警	i. 驱动器掉电后，电机位置发生了偏移。	i. 驱动器执行故障复位命令; ii. 驱动器重新上电或软复位。
35401	驱动器短路	1.驱动器 UVW 输出线缆发生短路或对地短路; 2.电机 UVW 发生短路或对地短路; 3.驱动器内部发生短路或对地短路; 4.驱动器受干扰导致误报此故障	1.电缆的 UVW 相间、UVW 与接地之间发生短路时，处理或更换线缆; 2.电机的 UVW 相间、UVW 与接地之间发生短路时，更换电机; 3.断开驱动器输出 UVW 接线，仍发生故障时，更换驱动器; 4.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35402	U 相输出电流过大	1.电流环调节器参数设置不合理，导致电流控制振荡; 2.电机参数设置错误; 3.驱动器内部电流采样电路异常。	1.调整电流环调节器参数; 2.正确设置电机参数; 3.更换驱动器
35403	V 相输出电流过大	1.电流环调节器参数设置不合理，	1.调整电流环调节器参数; 2.正确设

		导致电流控制振荡; 2.电机参数设置错误; 3.驱动器内部电流采样电路异常。	置电机参数; 3.更换驱动器
35404	W 相输出电流过大	1.电流环调节器参数设置不合理,导致电流控制振荡; 2.电机参数设置错误; 3.驱动器内部电流采样电路异常。	1.调整电流环调节器参数; 2.正确设置电机参数; 3.更换驱动器
35405	驱动器硬件过流	1.电机负载过大或电机加速度、减速度设置过大,加减速时间设置过小; 2.转子补偿角设置值有偏差，同时不满足转子定位错误检出条件; 3.编码器反馈有异常跳变; 4.电流环调节器参数设置不合理，导致电流控制振荡; 5.电机参数设置错误(线电阻、线电感、反电动势和转子惯量等); 6.驱动器内部电流检测电路异常，或驱动器抱闸电路损坏,无 24V 输出; 7.电机抱闸损坏; 8.转矩偏移值或静态平衡补偿值设置不合理。	1.检测并处理机械负载传动,确保无卡死等现象，增大加减速时间; 2.重新检测转子补偿角; 3.检查电机编码器接线并确保接线规范正确; 4.调整电流环调节器参数; 5.正确设置电机参数; 6.更换驱动器; 7.更换电机; 8.优化上位控制器动力学模型,优化给定值或根据实际负载设定。
35406	驱动器输出对地短路	1.驱动器 UVW 输出线缆发生对地短路; 2.电机 UVW 发生对地短路; 3.驱动器内部发生短路或对地短路。	1.电缆的 UVW 与接地之间发生短路时，处理或更换线缆; 2.电机的 UVW 与接地之间发生短路时，更换电机; 3.断开驱动器输出 UVW 接线，仍发生故障时，更换驱动器。
35407	直流母线过压	1.驱动器动力输入电源电压过大; 2.电机快速停止时的能耗制动能量过大; 3.能耗制动电阻未接或接线错误; 4.能耗制动电阻阻值过大; 5.驱动器内部异常。	1.调整驱动器动力输入电源至允许范围内; 2.减小电机停止减速度或减小负载; 3.改正能耗制动电阻接线，确保接线正确; 4.适当减小能耗制动电阻阻值(阻值不能低于允许的最小值),增大能耗制动电阻功率; 5.更换驱动器。
35408	直流母线欠压	1.驱动器动力输入电源电压过低; 2.驱动器内部电压采样电路异常; 3.驱动器功率回路设置错误,220V 供电设置为 380V 供电; 4.驱动器动力输入电源线断开。	1.调整驱动器动力输入电源至正常工作允许范围内; 2.更换驱动器; 3.驱动器功率回路设置与实际供电电源一致; 4.检查并处理驱动器动力输入电源线接线，确保接线正确牢固。
35409	功率模块过热	1.电机负载过大; 2.驱动器内部温度采样电路异常; 3.驱动器运行环境温度超出了允许工作范围。	1.减小电机实际机械负载; 2.更换驱动器; 3.降低环境温度,比如改善机柜散热条件。
35410	CPU1 看门狗溢出	驱动器内部异常。	更换驱动器。
35411	CPU2 看门狗溢出	驱动器内部异常。	更换驱动器。
35412	能耗制动电阻过载	1.电机频繁进行快速停止操作导致	1.改变电机运行工况,避免电机频繁

		能耗制动能量过大; 2.能耗制动电阻功率设置与实际电阻不一致。	进行快速停止操作,比如延长电机停止时间。或者更换更大功率的能耗制动电阻; 2.正确设置能耗制动电阻功率，设置值与能耗制动电阻实际功率一致。
35413	电机持续过载	1.电机负载过大; 2.电机加减速时间设置过小; 3.电机参数设置错误; 4.抱闸释放动作异常; 5.电机选型错误,功率过小(比如大功率驱动器,带载小功率电机长时间满载高速运行); 6.驱动器内部电流采样电路异常。	1.减小电机实际机械负载，确保机械无卡死现象; 2.增大电机运行时的加减速时间; 3.检查电机参数,确保电机参数设置正确(比如电机额定电流,电机热时间常数等); 4.检查电机抱闸线路，确保电机抱闸动作正常; 5.更换大容量电机; 6.更换驱动器
35414	位置跟随误差过大	1.电机负载过大; 2.控制参数不合适; 3.抱闸释放动作异常; 4.位置跟随误差过大判定阈值或判定时间设施过小。	1.减小电机实际机械负载，确保机械无卡死现象; 2.优化控制参数，增强伺服相应性能; 3.检查电机抱闸线路，确保电机抱闸动作正常; 4.适当增大位置跟随误差过大判定阈值或判定时间。
35415	正向软限位	位置反馈值超过（正向软限位值+定位完成阈值）。	运动范围不要超过正向软限位设定值，如果不需要正向软限位功能，可通过参数位置控制开关禁止。
35416	负向软限位	位置反馈值超过（负向软限位值-定位完成阈值）。	运动范围不要超过负向软限位设定值，如果不需要正向软限位功能，可通过参数位置控制开关禁止。
35417	编码器数据溢出	位置模式下，未使能无限位置控制时，编码器多圈值超出了实际编码器的多圈位数。	执行编码器多圈清零操作，或使能无限位置控制模式，或工作在非位置模式下（转矩模式或速度模式）。
35418	CPU1 工作异常	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35419	CPU2 工作异常	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35420	CPU1 内存异常	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35421	CPU2 内存异常	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35422	CPU 内存冲突	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35423	磁极定位错误	1.电机转子位置补偿角设定值与检测值不一致; 2.转子定位故障检测灵敏度设置过小; 3.静态平衡力矩补偿值设置与实际负载不一致; 4.电机接线错误导致转子相位角发生改变; 5.电机异常导致转子相位角	1.重新检测电机转子位置补偿角，并正确设置; 2.适当增大转子定位故障检测灵敏度设置值; 3.根据实际负载正确设置静态平衡力矩补偿值; 4.正确接线，并重新检测电机转子位置补偿角; 5.更换电机; 6.根据

		发生改变; 6.重力负载导致伺服使能瞬间电机发生转动，速度超过转子定位故障检测灵敏度设置阈值。	实际负载正确设置静态平衡力矩补偿值。
35424	编码器数据异常	1.编码器本身数据发生异常; 2.编码器线缆线序错误或接触不良; 3.由于噪音干扰导致编码器数据异常。	1.更换电机或编码器; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35425	编码器通信异常	1.编码器本身数据发生异常; 2.编码器线缆线序错误或接触不良; 3.由于噪音干扰导致编码器数据异常。	1.更换电机或编码器; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35426	编码器通信超时	1.编码器本身数据发生异常; 2.编码器线缆线序错误或接触不良; 3.由于噪音干扰导致编码器数据异常。	1.更换电机或编码器; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35427	编码器内部异常 1	编码器内部状态异常	编码器清零后软复位或重启驱动器
35428	驱动器其它轴异常	1.其它轴发生了故障; 2.驱动器内部电路异常	1.检查其它轴,并复位报故障的轴，确保其它轴无故障，此故障可通过参数 0x2094 来禁止检测; 2.更换驱动器
35429	控制编码器超速	1 编码器位置反馈值在一个位置采样周期内的变化量过大，超过电机最高转速的 1.3 倍; 2.编码器异常; 3.由于噪音干扰导致编码器数据异常。	1.优化电机参数和控制参数，电机最高转速设定值通常不小于实际电机最高转速; 2.检查编码器设置和编码器接线; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35430	驱动器持续过载	1.电机负载过大或加减速时间设置过小; 2.实际机械负载过大或存在卡死现象; 3.电机抱闸未释放; 4.电机或电机抱闸异常; 5.驱动器内部异常。	1.减小电机实际机械负载或增大电机加减速时间; 2.检查机械负载传动方式,确保无卡死等现象; 3.检查电机抱闸接线确保接线可靠; 4.更换电机; 5.更换驱动器。
35431	驱动器输出缺相	1.驱动器 U,V,W 输出存在断线或接线不良等现象; 2.电机阻抗过大; 3.驱动器内部电流采样电路异常。	1.检查电机 U,V,W 接线并确保接线可靠; 2.更换电机或禁止驱动器输出缺相检测功能; 3.更换驱动器。
35432	电机失速	1.驱动器调节器参数设置不合理导致速度跟踪超调过大; 2.由于电磁噪音干扰导致编码器数据异常变化; 3.由于编码器损坏导致编码器数据异常变化; 4.驱动器内部电路异常。	1.优化调节器参数; 2.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 3.更换电机或编码器; 4.更换驱动器。
35433	电流跟随误差过大	1.驱动器调节器参数设置不合理导	1.优化调节器参数; 2.通过规范接线

		致速度跟踪超调过大; 2.由于电磁噪音干扰导致编码器数据异常变化; 3.由于电机损坏导致编码器数据异常变化; 4.驱动器内部电路异常。	和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 3.更换电机; 4.更换驱动器。
35434	位置目标值异常	1.CSP 模式下，伺服使能瞬间，位置目标值和位置实际值之差超过位置跟随误差过大设定阈值; 2.CSP 模式下，电机运行过程中，目标轨迹加速度超过参数 最大加速度设定阈值，且位置目标值和位置实际值之差超过位置跟随误差过大设定阈值。	1.检查并确认位置目标值和位置实际值无异常，使差值不超过位置跟随误差过大设定阈值; 2.检查并确认位置目标值无异常突变，或者适当增大最大加速度设定阈值或位置跟随误差过大设定阈值。
35435	编码器上电数据溢出	驱动器上电时反馈位置值超出了编码器允许的最大范围。	编码器清零后软复位或重启驱动器。
35436	位置目标值溢出	位置模式下，当禁止无限位置控制时，位置目标值超出了允许的最大范围。	执行编码器多圈清零操作，或使能无限位置控制模式，或工作在非位置模式下（转矩模式或速度模式）。
35437	电机抱闸异常	1.电机抱闸本身发生异常，不能正常制动; 2.电机在高速运行时突然伺服 OFF，制动时间过长; 3.电机抱闸制动时间设置值小于抱闸实际制动动作时间; 4.转子定位故障检测灵敏度设置过小。	1.更换电机; 2.优化工艺逻辑控制，避免高速运行时突然伺服 OFF; 3.电机抱闸制动时间设置值要不小于抱闸实际制动动作时间; 4.适当增大转子定位故障检测灵敏度设置值。
35438	控制电源欠压	1.24V 控制电源异常; 2.24V 控制电源接线错误,或接触不良; 3.24V 控制电源负载过大; 4.驱动器内部电路异常。	1.更换 24V 控制电源; 2.检查 24V 控制电源接线并确保接线可靠; 3.检查 24V 控制电源负载,确保 24V 控制电源的容量能满足所有工况下的负载消耗 4.更换驱动器。
35439	STO1 触发	STO1 触发或接线不良。	检查 STO 接线，确保接线可靠并且未处于触发状态。
35440	STO2 触发	STO2 触发或接线不良。	检查 STO 接线，确保接线可靠并且未处于触发状态。
35441	正向硬限位开关触发	单方向运行至机械限位，导致硬件限位触发。	可直接清除故障，反方向运行至机械限位恢复。注意位置规划不要超出硬件限位的范围。
35442	负向硬限位开关触发	单方向运行至机械限位，导致硬件限位触发。	可直接清除故障，反方向运行至机械限位恢复。注意位置规划不要超出硬件限位的范围。
35443	电机超速	1.电机实际速度反馈值超出电机最高转速的 1.1 倍; 2.编码器异常。	1.优化电机参数和控制参数，电机最高转速设定值通常不小于实际电机最高转速; 2.检查编码器设置和编码器接线。
35444	急停输入开关触发	急停输入开关触发或接线不良。	检查急停输入开关接线，确保接线

			可靠并且未处于触发状态。
35445	转矩监测饱和故障	1.电机负载过大，超过了转矩监测告警阈值; 2.转矩饱和监测阈值设置过小。	1.减小电机实际机械负载或增大电机加减速时间; 2.增大转矩饱和监测阈值设置值，当该阈值设置为 0 时，不检测该故障。
35446	速度跟随误差过大	1.电机负载过大; 2.控制参数不合适; 3.抱闸释放动作异常; 4.速度跟随误差过大判定阈值或判定时间设置过小。	1.减小电机实际机械负载，确保机械无卡死现象; 2.优化控制参数，增强伺服相应性能; 3.检查电机抱闸线路，确保电机抱闸动作正常; 4.适当增大速度跟随误差过大判定阈值或判定时间。
35447	驱动器短路 2	1.驱动器 UVW 输出线缆发生短路或对地短路; 2.电机 UVW 发生短路或对地短路; 3.驱动器内部发生短路或对地短路; 4.驱动器受干扰导致误报此故障	1.电缆的 UVW 相间、UVW 与接地之间发生短路时，处理或更换线缆; 2.电机的 UVW 相间、UVW 与接地之间发生短路时，更换电机; 3.断开驱动器输出 UVW 接线，仍发生故障时，更换驱动器; 4.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35448	寻原点失败	1.寻原点参数设置不合理; 2.寻原点启动时,电机已处于限位开关触发状态; 3.寻原点过程中切换至非 HM 模式。	1. 正确设置寻原点参数; 2.寻原点启动时,确保电机未处于限位开关触发状态; 3.寻原点过程中，避免切换控制模式。
35449	EtherCAT 过程数据错误	PDO 设定值超出了对象允许范围。	PDO 设定值在对象允许范围内。
35450	EtherCAT 总线指令非法	EtherCAT 通讯状态机与控制字时序配合错误。	上位机正确处理 EtherCAT 通讯状态机与控制字时序。
35451	EtherCAT 通讯周期错误	1.EtherCAT 通讯周期小于伺服控制周期; 2.EtherCAT 通讯周期设置不是 250us 的 2 的整数次幂。	1.调整 EtherCAT 通讯周期或伺服控制周期，使通讯周期大于伺服控制周期; 2.EtherCAT 通讯周期设置为 250us 的 2 的整数次幂。
35452	位置规划运行错误	1.EtherCAT 控制权下，运行 PP 模式时，位置目标值缓存超限; 2.驱动器内部异常。	1.优化 EtherCAT 主站控制工艺，减少位置目标值缓存数量，位置目标值缓存一般不超过 4 个; 2.更换驱动器。
35453	EtherCAT 非法同步模式	1.EtherCAT 通信 DC 模式配置错误; 2.EtherCAT 通信未激活 DC 模式。	1.正确配置 EtherCAT 通信 DC 模式; 2.激活 EtherCAT 通信 DC 模式。
35454	位置目标值超出设定范围	模数无限位置控制模式或者禁止无限位置控制模式时，位置目标值超出了允许设定范围。	位置目标值设置在位置范围下限值和位置范围上限值之间，或者使用普通无限位置模式。
35455	电机 U 相瞬时过载	1.电机负载过大; 2.电机加减速时间设置过小; 3.电机参数设置错误; 4.抱闸释放动作异常; 5.电机选型错误,功率过小(比如大功率驱动器,	1.减小电机实际机械负载，确保机械无卡死现象; 2.增大电机运行时的加减速时间; 3.检查电机参数,确保电机参数设置正确(比如电机额

		带载小功率电机长时间满载高速运行); 6.驱动器内部电流采样电路异常; 7.电机快速过载保护阈值和保护时间设置偏小。	定电流,电机热时间常数等); 4.检查电机抱闸线路，确保电机抱闸动作正常; 5.更换大容量电机; 6.更换驱动器; 7.适当增大电机快速过载保护阈值和保护时间。
35456	电机 V 相瞬时过载	1.电机负载过大; 2.电机加减速时间设置过小; 3.电机参数设置错误; 4.抱闸释放动作异常; 5.电机选型错误,功率过小(比如大功率驱动器,带载小功率电机长时间满载高速运行); 6.驱动器内部电流采样电路异常; 7.电机快速过载保护阈值和保护时间设置偏小。	1.减小电机实际机械负载，确保机械无卡死现象; 2.增大电机运行时的加减速时间; 3.检查电机参数,确保电机参数设置正确(比如电机额定电流,电机热时间常数等); 4.检查电机抱闸线路，确保电机抱闸动作正常; 5.更换大容量电机; 6.更换驱动器; 7.适当增大电机快速过载保护阈值和保护时间。
35457	电机 W 相瞬时过载	1.电机负载过大; 2.电机加减速时间设置过小; 3.电机参数设置错误; 4.抱闸释放动作异常; 5.电机选型错误,功率过小(比如大功率驱动器,带载小功率电机长时间满载高速运行); 6.驱动器内部电流采样电路异常; 7.电机快速过载保护阈值和保护时间设置偏小。	1.减小电机实际机械负载，确保机械无卡死现象; 2.增大电机运行时的加减速时间; 3.检查电机参数,确保电机参数设置正确(比如电机额定电流,电机热时间常数等); 4.检查电机抱闸线路，确保电机抱闸动作正常; 5.更换大容量电机; 6.更换驱动器; 7.适当增大电机快速过载保护阈值和保护时间。
35458	动态制动过载	电机运行时，相邻 2 次动态制动停机间隔时间过短。	如电机运行时动态制动停机，请间隔至少 360*实际速度^2/额定速度^2（s）。
35459	驱动器内部异常	驱动器内部异常。	更换驱动器。
35460	限位开关异常	限位开关触发或接线不良。	检查限位开关接线，确保接线可靠并且未处于触发状态。
35461	EtherCAT 总线通讯异常	1.EtherCAT 通讯受到干扰; 2.EtherCAT 网线断开或接触不良; 3.上位机实时性不够; 4.上位机 EtherCAT 主站底层 DC 同步机制与驱动器需求不匹配; 5.驱动器内部异常。	1.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 2.检查并处理 EtherCAT 网线连接确保连接可靠; 3.更换实时性更强的上位机，或延长 EtherCAT 通讯周期，或适当增大超时检测灵敏度设定值; 4.修改上位机 EtherCAT 主站底层 DC 同步机制，确保上位机 SM2 事件比 DC 同步信号至少提前 125us; 5.更换驱动器。
35462	接口编码器分辨率变更	更改了接口编码器分辨率。	重上电或者软复位驱动器。
35463	编码器过热	1.编码器实际温度过高; 2.编码器异常。	1.降低编码器实际工作环境温度至允许范围内; 2.更换电机或编码器。
35464	编码器电池欠电压故障	1.编码器电池电压过低; 2.编码器	1.更换编码器电池; 2.检查并处理电

		电池接线接触不良。	池接线，确保电池接线正确牢固。
35465	控制模式设定错误	伺服 ON 时，控制模式设定为驱动器不支持的模式，如 NM、VL 或 IP，或者 EtherCAT 控制权时控制模式设置为 PV 或 PT。	伺服 ON 时，设置驱动器支持的控制模式。
35466	上电位置偏差过大	驱动器上电时，与上一次掉电保存位置不一致，超过设定阈值。	检查机械位置是否改变，确认机械零点无异常后可清除。
35467	编码器加速度异常故障	1.编码器本身数据发生异常; 2.编码器线缆线序错误或接触不良; 3.由于噪音干扰导致编码器数据异常。	1.更换电机或编码器; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35468	电机堵转	1.机械负载存在卡死现象或堵转现象; 2.电机抱闸未释放。	1.检测并处理机械负载传动,确保无卡死等现象; 2.检测并处理抱闸回路，确保电机抱闸正常释放。
35469	写 EEPROM 数据异常	驱动器内部异常。	更换驱动器。
35470	读 EEPROM 数据异常	驱动器内部异常。	更换驱动器。
35471	抱闸控制电路异常	1.电机抱闸接线短路或接触不良; 2.电机抱闸内部发生短路或接触不良; 3.驱动器内部异常。	1.检查驱动器抱闸输出接线并确保接线正确可靠; 2.更换电机; 3.更换驱动器。
35472	CPU1 过载	1.驱动器运行受到噪音干扰; 2.调试软件数据采集过大; 3.驱动器内部电路异常。	1.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 2.关闭部分调试软件数据采集通道; 3.更换驱动器。
35473	CPU2 过载	1.驱动器运行受到噪音干扰; 2.调试软件数据采集过大; 3.驱动器内部电路异常。	1.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 2.关闭部分调试软件数据采集通道; 3.更换驱动器。
35474	CPU1 握手失败	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35475	DriveMaster 通讯超时	1.驱动器调试线缆断开或接触不良; 2.驱动器调试串口通讯受到干扰。	1.检查驱动器调试线缆接线并确保连接可靠; 2.使用隔离型串口调试线缆，或者通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35476	ESC 配置 EEPROM 异常	驱动器内部异常。	更换驱动器。
35477	ESC 内部访问错误	驱动器内部异常。	更换驱动器。
35478	伺服使能未准备好	1.伺服 ON 时，驱动器处于实际电机且虚拟编码器模式; 2.伺服 ON 时，编码器通信处于断开状态; 3.伺服 ON 时，电机转速高于 30rpm; 4.伺服 ON 时，STO 状态未解除; 5.伺服 ON 时，直流母线电压过低，充电继电器未吸合; 6.伺服 ON 时，	1.检查驱动器电机模式，确保伺服 ON 时驱动器处于实际电机和实际编码器模式; 2.检查编码器通信状态，确保伺服 ON 时编码器通信状态正常; 3.检查电机运行状态，确保伺服 ON 时电机处于静止状态; .检查 STO 状态，确保伺服 ON 时 STO

		动态制动状态未解除; 7.驱动器内部异常。	状态已经解除; 5.检查直流母线电压状态，确保伺服 ON 时直流母线电压满足使能阈值且充电继电器吸合; 6.检查动态制动状态，确保伺服 ON 时动态制动状态已经解除; 7.更换驱动器。
35479	CPU2 握手失败	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35480	CPU1 主任务超时	1.驱动器运行受到噪音干扰; 2.调试软件数据采集过大; 3.驱动器内部电路异常。	1.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 2.关闭部分调试软件数据采集通道。
35481	直流母线充电继电器异常	驱动器内部充电继电器发生异常。	更换驱动器。
35482	CPU 内部错误	1.驱动器固件运行异常; 2.驱动器内部异常。	1.升级驱动器固件; 2.更换驱动器。
35483	位置实际值溢出	位置模式下，当禁止无限位置控制时，位置实际值超出了允许的最大范围。	执行编码器多圈清零操作，或使能无限位置控制模式，或工作在非位置模式下（转矩模式或速度模式）。
35484	编码器内部异常 2	编码器内部状态异常。	编码器清零后软复位或重启驱动器。
35485	编码器内部异常 3	编码器内部状态异常。	编码器清零后软复位或重启驱动器。
35486	第 2 位置跟随误差过大	1.电机负载过大; 2.控制参数不合适; 3.抱闸释放动作异常; 4.第 2 位置跟随误差过大判定阈值或判定时间设施过小。	1.减小电机实际机械负载，确保机械无卡死现象; 2.优化控制参数，增强伺服相应性能; 3.检查电机抱闸线路，确保电机抱闸动作正常; 4.适当增大第 2 位置跟随误差过大判定阈值或判定时间。
35487	STO 接线异常	STO1 或 STO2 触发或接线不良。	检查 STO 接线，确保接线可靠并且未处于触发状态。
35488	第 2 速度跟随误差过大	1.电机负载过大; 2.控制参数不合适; 3.抱闸释放动作异常; 4.第 2 速度跟随误差过大判定阈值或判定时间设施过小。	1.减小电机实际机械负载，确保机械无卡死现象; 2.优化控制参数，增强伺服相应性能; 3.检查电机抱闸线路，确保电机抱闸动作正常; 4.适当增大第 2 速度跟随误差过大判定阈值或判定时间。
35489	主电源输入异常	1.驱动器动力输入电源接线不良; 2.驱动器功率回路设定选择为三相输入,但实际动力电源输入为单相;3.前端使用电子变压器,电子变压器谐波异常。	1.检查驱动器动力输入电源接线并确保接线可靠; 2.正确设置驱动器功率回路，设置值与实际动力电源输入一致;3.按照变压器手册接线，必要时在伺服驱动器前端加装滤波器。
35490	电机抱闸断线	1.电机抱闸未连接或接触不良; 2.电机抱闸异常; 3.驱动器内部异常。	1.检查并处理电机抱闸接线，确保接线正确牢固; 2.更换电机; 3.更换

			驱动器。
35491	协处理器通讯异常	1.编码器线缆线序错误或接触不良; 2.由于噪音干扰导致编码器数据异常。	1.改正接线线序或对接线做加固处理; 2.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35492	编码器 AB 信号变化异常	1.编码器线缆线序错误或接触不良; 2.由于噪音干扰导致编码器数据异常。	1.改正接线线序或对接线做加固处理; 2.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35493	整流模块过热	1.驱动器内部温度采样电路异常; 2.驱动器运行环境温度超出了允许工作范围。	1.更换驱动器; 2.降低环境温度,比如改善机柜散热条件。
35494	散热器过热	1.驱动器内部温度采样电路异常; 2.驱动器运行环境温度超出了允许工作范围。	1.更换驱动器; 2.降低环境温度,比如改善机柜散热条件。
35495	电机过热	1.电机负载过大; 2.电机运行环境温度过高，超过允许范围; 3.电机过热保护热偶电阻值设置错误; 4.电机温度传感器异常;5.驱动器内部异常。	1.减小电机实际机械负载，确保机械无卡死现象; 2.增强电机散热，确保运行环境温度在允许范围内; 3.正确设置电机过热保护热电偶电阻设置值;4.更换电机;5.更换驱动器。
35496	增量式编码器 Z 信号异常	1.编码器本身数据发生异常; 2.编码器线缆线序错误或接触不良; 3.由于噪音干扰导致编码器数据异常。	1.更换电机或编码器; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35497	能耗制动电路异常	伺服参数设置的能耗制动选择，与能耗制动电阻实际接线不一致。	正确设置能耗制动选择的伺服参数，使之与能耗制动电阻实际接线一致。
35498	CPU 过热	1.驱动器内部温度采样电路异常; 2.驱动器运行环境温度超出了允许工作范围。	1.更换驱动器; 2.降低环境温度,比如改善机柜散热条件。
35499	主电源掉电	1.驱动器伺服 ON 状态下，发生了动力电源掉电; 2.由于噪音干扰导致主电源掉电检测异常; 3.伺服参数主电源掉电检测时间设置过小。	1.检查驱动器动力电源及接线，确保电源正常，连接可靠; 2.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境; 3.适当增大伺服参数主电源掉电检测时间设置值。
35500	STO1 电路诊断异常	1.STO1 触发或接线不良; 2.驱动器内部异常。	1.检查 STO 接线，确保接线可靠并且未处于触发状态; 2.更换驱动器。
35501	STO2 电路诊断异常	1.STO2 触发或接线不良; 2.驱动器内部异常。	1.检查 STO 接线，确保接线可靠并且未处于触发状态; 2.更换驱动器。
35502	霍尔信号异常	1.霍尔传感器本身信号发生异常;	1.更换电机或霍尔传感器，或者禁

		2.霍尔传感器接线线序错误或接触不良; 3.由于噪音干扰导致霍尔信号异常。	止霍尔信号检测; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35503	编码器 AB 信号欠相异常	1.霍尔传感器或 AB 编码器本身信号发生异常; 2.霍尔传感器或 AB 编码器接线线序错误或接触不良; 3.由于噪音干扰导致霍尔或 AB 编码器信号异常。	1.更换电机或霍尔传感器和编码器; 2.改正接线线序或对接线做加固处理; 3.通过规范接线和布线，增加接地线截面积，加套磁环等措施改善设备电磁环境。
35504	驱动器内部异常 2	驱动器内部异常。	驱动器内部异常。
35505	机型文件错误，机器人不允许使用	机型文件错误。	使用正确的机型文件。
35506	软限位超出硬限位	软限位超出硬限位。	软限位超出硬限位。
35507	配置文件有 STOP_STO_TIME 字段且 stop1 时间大于 760ms，但安全板固件版本过低	配置文件有 STOP_STO_TIME 字段且 stop1 时间大于 760ms，但安全板固件版本过低	升级安全板固件
35508	设置安全板 STO 断开时间失败	硬件故障	检查硬件或者固件是否正常
35509	给定时间停止参数设置失败！	给定时间停止参数设置过大	STOP0 的给定时间停止参数应小于手动模式 STO 断开时间减去 30，STOP1 的给定时间停止参数应小于自动模式 STO 断开时间减去 60
35600	STO 开关没接通	机器人使能把手未生效；	按下手动使能把手；重新插拔 STO 线，更换 STO 线，更换伺服驱动器
35601	抱闸电压过高	固件版本不对；参数设置抱闸电压不对；驱动器硬件损坏；	检查固件及参数版本；更换驱动板；
35602	驱动模块过热报警	进行相应的冷机处理；	更新伺服驱动器至最新的固件版本，更换伺服驱动器；
35603	编码器 1 报错	关节电机端编码器异常；	确认固件是否为正确版本，重新进行关节编码器标定；更换机器人关节；
35604	编码器 2 报错	关节电机端编码器异常；	确认固件是否为正确版本，重新进行关节编码器标定；更换机器人关节；
35605	编码器存在扰动	线路连接正确，编码器本身有问题，周围存在干扰。驱动板有问题；	编码器线重新插拔；更换驱动板；检查是否是正确的固件；
35606	驱动器 A 相过流	检查电机是否堵转；检查负载过大或者转速过快；关节机构阻力大；	先检查线束是否链接正确；检查电机偏角是否正确；检查上使能时抱闸是否正确打开；负载大的时候，降低转速查看故障是否消失；观察运行时的故障关节是否有阻力过大的异常噪音；
35607	驱动器 B 相过流	检查电机是否堵转；检查负载过大	先检查线束是否链接正确；检查电

		或者转速过快；关节机构阻力大；	机偏角是否正确；检查上使能时抱闸是否正确打开；负载大的时候，降低转速查看故障是否消失；观察运行时的故障关节是否有阻力过大的异常噪音；
35608	驱动器 C 相过流	检查电机是否堵转；检查负载过大或者转速过快；关节机构阻力大；	先检查线束是否链接正确；检查电机偏角是否正确；检查上使能时抱闸是否正确打开；负载大的时候，降低转速查看故障是否消失；观察运行时的故障关节是否有阻力过大的异常噪音；
35609	驱动器 IGBT 过流	检查电机是否堵转；检查负载过大或者转速过快；关节机构阻力大；	先检查线束是否链接正确；检查电机偏角是否正确；检查上使能时抱闸是否正确打开；负载大的时候，降低转速查看故障是否消失；观察运行时的故障关节是否有阻力过大的异常噪音；
35610	电机过载	电机堵转时间过长，准据输出过大；加减速设置过大，造成转矩输出过大；电机烧毁，上使能状态电流输出过大造成过载；抱闸损坏，电机无法运转为卡死状态；	检查电机三相线及抱闸线是否正确连接；看检查电机偏角是否正确；检查加减速设置；测量电机两端电阻判断电机是否烧毁；
35611	零电流报警	未使能时电流为采集异常；	更新伺服驱动器至最新的固件版本，更新伺服驱动器至最新的参数版本，更换伺服驱动器；
35612	直流母线欠压	母线供电电压不足；	检查机器人供电电压是否正常；检查驱动器电容是否正常；更换伺服驱动器；检查电源板是否正常；
35613	直流母线过压	母线供电电压过高；机器人减速过快；底座电源处理异常；	检查机器人供电电压；检查机器人减速度；检查电源板是否正常；；
35614	位置超差	需要检查硬件及软件是否为正确的配置及版本；	检查线束是否正确连接；检查关节编码器是否正常；更换关节硬件，电机，减速器，驱动器；确认主站发指令是否合理；
35615	主从站失联	需要检查线缆连接情况，检查固件版本；	检查线束是否正确连接；检查端子是否松动；检查固件版本是否正确；
35616	电角度寻找失败	线缆连接不正常，需要检查编码器线电机三相线抱闸线缆连接情况，检查固件版本及参数版本，检查设置的找偏角参数是否正确；	检查编码器线电机三相线抱闸线缆连接情况，检查固件版本及参数版本，检查设置的找偏角参数是否正确；
35617	高速侧多圈值错误	固件版本不对，驱动板上 EE 损坏；	检查固件版本；更换驱动板；
35618	低速侧多圈值错误	固件版本不对，驱动板上 EE 损坏；	检查固件版本；更换驱动板；
35619	RSC 固件版本与控制器版本不匹配	RSC 固件版本与控制器版本不匹配	检查 RSC 固件版本；升级 RSC 固件版本；

35620	电机短路(电流过大)保护	1、电流保护机制触发，电流过大；	1、检查是不是发生了剧烈的撞击；2、检查是否存在电机相间短路；
35621	电源板母线电压过低	1、电源线断开或者接入电源电压不足；2、电源板硬件故障；	1、检查电源接入是否正确；2、检查驱动器硬件是否有异常；
35622	电源板母线电压过高	1、电源线接错，导致输入电压过高；2、电源板硬件故障；	1、检查电源接入是否正确；2、检查电源板硬件是否有异常；
35623	电源板主继电器反馈异常	1、电源板硬件故障；	1、检查电源板硬件是否有异常；
35624	电源板漏电流过大	1、本体线束或者电源线与控制柜短路；2、电源板硬件故障；	1、检查本体线束和电源线束是否有破损；2、检查电源板硬件是否有异常；
35625	电源板抱闸过载	1、抱闸线之间存在断路；2、抱闸硬件故障；3、电源板硬件故障或者伺服驱动器模块故障；	1、检查本体线束是否有破损短路；2、检查抱闸硬件是否有异常；3、检查电源板硬件和驱动器硬件是否有异常；
35626	电源板用户电源过載	1、用户外接设备，接线错误或者接线短路；2、用户外接了超过额定使用范围的设备；	1、检查用户设备接线是否异常；2、检查用户设备是否超过额定范围
35627	电源板 EEPROM 写错误	1、电源板硬件 EEPROM 器件硬件故障；	1、检查电源板硬件是否有异常；
35628	电源板 EEPROM 读错误	1、电源板硬件 EEPROM 器件硬件故障；	1、检查电源板硬件是否有异常；
35629	电源板功率继电器异常	1、电源板硬件故障；	1、检查电源板硬件是否有异常；
35630	电源板热交换器故障（CR35）	1、热交换器接线发生短路；2、电源板硬件故障；	1、检查热交换器线束连接是否异常；2、检查电源板硬件是否有异常；
35631	电源板 48V 电源电压过低	1、关节运行阻力大、本体所接负载过重或者加速度过大；2、电源接线可能存在异常；	1、检查关节运行是否平顺；2、检查电源接入是否正确；
35632	电源板 48V 电源电流过大	1、关节运行阻力大、本体所接负载过重或者加速度过大；	1、检查关节运行是否平顺；
35633	电源板泄放 MOS 异常（CR35）	1、电源板硬件故障；	1、检查电源板硬件是否有异常；
35634	电源板泄放电流过大（CR35）	1、泄放电阻硬件故障；	1、检查泄放电阻硬件是否异常；
35635	电源板泄放 MOS 短路（CR35）	1、电源板硬件故障；	1、检查电源板硬件是否有异常；
35636	电源板泄放电流 I2T 故障	1、电源板硬件故障；	1、检查电源板硬件是否有异常；
35637	电源板电源板通信故障	1、电源板硬件故障；	1、检查电源板硬件是否有异常；
35638	FPGA 看门狗故障	1、伺服驱动器控制板硬件故障；	1、检查驱动器硬件是否异常；
35639	漏电流过大（FPGA 报）	1、本体线束或者电源线与控制柜短路；	1、检查本体线束和电源线束是否有破损；
35640	编码器插补错误	1、检查编码器硬件是否异常；	
35641	编码器多圈值异常	1、编码器电池断开或者接触不良；2、编码器电池电压不足；	1、检查编码器电池线束；2、检查编码器电池是否有电；
35642	编码器数据异常	1、检查电机端编码器线缆是否正确；2、检查关节电机端编码器磁环与读数板间距是否符合；	

35643	与编码器通信异常	1、编码器线束断开或者接触不良；	1、检查编码器线束连接是否可靠；
35644	编码器超速故障	1、编码器线束断开或者接触不良；2、编码器磁环与读数板之间间隙发生变化；	1、检查电机端编码器线缆是否正确；2、检查关节电机端编码器磁环与读数板间距是否符合；
35645	传感器数据 CRC 错误	1、传感器数据帧 CRC 校验错误；	1、检查传感器线束和硬件是否异常；
35646	传感器数据帧全零错误	1、传感器收到的数据帧都为 0；	1、检查传感器线束和硬件是否异常；
35647	传感器数据速度超限	1、传感器计算的关节速度与关节实际速度偏差过大；	1、检查传感器线束和硬件是否异常；2、检查传感器板安装是否可靠；
35648	传感器零点读写异常	1、传感器零点标定时，写入和读取数据不一致；	1、检查传感器线束和硬件是否异常；

17.1.3 4XXXX

错误码	错误描述	故障可能原因	修复方法
41410	力超限	笛卡尔空间观测力 Fx、Fy、Fz 超限	检查力控前置设置和传感器硬件状态,编辑设定的程序，减少总合外力
41411	力矩超限	笛卡尔空间观测力矩 Mx、My、Mz 超限	检查力控前置设置和传感器硬件状态,编辑设定的程序，减少总合外力
41412	位置超限	轴空间位置超限	移动当前位置到限位内
41413	位置超限	笛卡尔空间位置超限	移动当前位置到限位内
41414	线速度超限	错误修复前不能继续运动	降低速度，选取更好的经过点，增加加速度，避免奇异点，改变动态属性等
41415	角速度超限	错误修复前不能继续运动	降低速度，选取更好的经过点，增加加速度，避免奇异点，改变动态属性等
41416	奇异点问题	到达或即将到达奇异位置	关节空间少许运动离开奇异位置，手动模式单轴 jog，自动模式使用 MoveAbsJ
41417	命令执行错误	不允许在超限时继续执行命令	將机器人关节位置移动到限位内
41418	力矩超限	触发力控监视，设定或测量的外力超过当前机器人的安全限制	检查力控参数设置并重新编辑程序，减少发给机器人的力矩指令
41419	力控初始化失败	TCP 长度超限，或是输入了不支持的力控参考坐标系	请检查实际工具长度，关于力控参考坐标系的定义相见产品手册
41420	力控开启失败	见内容	1.检查当前工具(Tool)设置是否与实际一致，设定的工具质量质心是否合理; 2.查看监控窗口，检查机器人坐标系、位姿是否与实际一致; 3.确认当前机型、RD 参数与实际一致; 4.可尝试回到机械零位，

			对传感器清零后再拖动; 6.确认开启拖动时机器人没有收到外力; 7.其他详见《拖动故障排查手册》
41421	力控暂停失败	机器人当前不处于力控运行状态	力控状态开启时，此命令才有效
41422	力控重新启动失败	机器人当前不处于力控暂停状态	首先暂停或停止力控，再重启开启
41423	传感器使用方式设置失败	力控处于未初始化状态	检查前置设置并重新初始化力控
41424	传感器校准失败	当前传感器使用方式为动态补偿，不支持清除传感器误差	如想使用清除传感器误差功能，请将传感器使用方式变为清零方式
41425	设置正弦搜索运动失败	非阻抗控制模式下、搜索运动暂停或是运行状态下不能够设置搜索运动,或参数设置超过阈值	保证当前处于阻抗模式下，且搜索运动处于停止状态,请检查参数设置
41426	设置利萨如搜索运动失败	非阻抗控制模式下、搜索运动暂停或是运行状态下不能够设置搜索运动,或参数设置超过阈值	保证当前处于阻抗模式下，且搜索运动处于停止状态,请检查参数设置
41427	开启搜索运动失败	未设置搜索运动或是当前不处于笛卡尔阻抗运行状态	检查控制模式指令，设置搜索运动参数后再尝试
41428	暂停搜索运动失败	搜索运动未开启或是当前不处于笛卡尔阻抗运行状态	请保证搜索运动已开启且处于笛卡尔阻抗运行状态
41429	重新启动搜索运动失败	搜索运动未暂停或是当前不处于笛卡尔阻抗运行状态	暂停当前的力控任务并检查控制模式，切换到迪卡尔阻抗控制模式后再尝试
41430	设置阻抗控制类型失败	阻抗处于非初始化状态	停止当前的力控任务并重新参数初始化
41431	关节阻抗初始化失败	当前不处于关节阻抗控制模式，初始化指令执行失败	运行力控前设置轴空间阻抗控制模式
41432	笛卡尔阻抗初始化失败	当前不处于笛卡尔阻抗控制模式，初始化指令执行失败	运行力控前设置笛卡尔阻抗控制模式
41433	设置零空间阻抗失败	当前不处于笛卡尔阻抗控制模式或是阻抗处于未初始化状态	检查当前的阻抗控制模式后再进行尝试。确保力值输入正确，且设定迪卡尔阻抗控制模式
41434	设置关节空间期望力失败	当前不处于关节阻抗控制模式或是阻抗处于未初始化状态,或参数设置超过阈值	检查当前的阻抗控制模式后再进行尝试。确保力值输入正确，且设定轴空间阻抗控制模式
41435	设置笛卡尔空间期望力失败	当前不处于笛卡尔阻抗控制模式或是阻抗处于未初始化状态,或参数设置超过阈值	检查当前的阻抗控制模式后再进行尝试。确保力值输入正确，且设定迪卡尔阻抗控制模式
41436	不允许切模式	力控运行中不允许切换阻抗模式	停止当前的力控任务并重新初始化
41437	开启虚拟墙失败	机器人法兰处于虚拟墙范围外，不满足虚拟墙约束	请将机器人法兰移动至虚拟墙范围内再开启虚拟墙
41438	开启虚拟墙失败	虚拟墙功能只能在拖动模式下开启	开启拖动模式后，重新打开虚拟墙
41439	开启虚拟墙失败	机器人当前状态不支持虚拟墙功能	
41440	碰撞检测关闭	阻抗模式下不需要碰撞检测	无
41441	碰撞检测重新开启	阻抗模式开启时自动关闭了碰撞检测，	无

		现在阻抗模式关闭，则自动打开碰撞检测	
41442	阻抗开启失败	阻抗开启时机器人不满足关节位置限制要求，无法开启阻抗	请确认机器人开启阻抗时机器人在关节位置限制内
41443	阻抗刚度数值异常	设置阻抗刚度超过理论最大值，已被默认最大值取代	请重新设置阻抗刚度数值在合理范围内
41444	阻抗刚度数值异常	设置阻抗刚度数值异常，不具有物理意义	请重新设置阻抗刚度数值在合理范围内
41445	力控辨识结果计算完成	力控辨识计算完成，结果满足技术要求	两遍辨识均完成后，请重启机器人
41446	力控辨识出现异常	力控辨识中，单轴标定结果超过许用范围	辨识结果将被边界值取代，请持续关注其可能的硬件问题
41447	拖动失败	实际反馈与模型力矩偏差较大，不允许开启拖动。	1.检查当前工具(Tool)设置是否与实际一致，设定的工具质量质心是否合理; 2.查看监控窗口，检查机器人坐标系、位姿是否与实际一致; 3.确认当前机型、RD 参数与实际一致; 4.可尝试回到机械零位，对传感器清零后再拖动; 6.确认开启拖动时机器人没有受到外力; 7.其他详见《拖动故障排查手册》
41448	力控初始化失败	力控过程中下电，不允许再直接 Fcinit	点击 pptomain 重新运行
41449	力控初始化失败	力控指令尚未执行完毕，不允许再开启力控	力控指令发送太频繁，请增加退出力控到重新开启之间的时间
41450	拖动失败	传感器电压值异常，不允许开启拖动。	请检查传感器状态是否正常
41451	力控模式触发保护,控制器下电	见内容	无
41452	碰撞检测关闭		
41453	碰撞检测重新开启		
41454	下电力控关闭		
41455	不检查路径偏离		
41456	将以位置控制模式继续执行		
41457	请重新开启力控执行		
41458	时间设置不能为负数或大于600		请检查设置的时间格式
41459	处于力控模式，不允许执行的操作		
41460	虚拟墙区域设置过小		
41462	开启拖动时动态标定传感器失败	理论模型与传感器偏差大	请检查负载设置和机械零点,如果负载检查正确仍然标定失败,请执行CalibSensorError RL指令来标定传感器
41461	力控模式工具 TCP 长度超过限制(norm<=0.15m),力控参考点设置为法兰中心		

41463	手持工件模式下禁止开启拖动		
41464	开始拖动功能执行失败	关节位置限制范围不满足拖动开启条件	请检查关节位置限制设置,关节位置限制范围应大于等于 25°,或调整关节角度
41465	传感器力矩与理论力矩相差过大，标定失败		请检查负载设置
41470	力控保护参数已设置为默认值		
41471	力控保护参数已设置为用户设定值		
41472	SetFcJointVelMax 执行失败	指令参数不合法	请检查参数设置，修正 SetFcJointVelMax 参数
41473	SetFcCartVelMax 执行失败	指令参数不合法	请检查指令参数设置，修正 SetFcCartVelMax 参数
41474	SetFcJointMomentumMax 执行失败	指令参数不合法	请检查指令参数设置，修正 SetFcJointMomentumMax 参数
41475	SetFcJointEnergyMax 执行失败	指令参数不合法	请检查指令参数设置，修正 SetFcJointEnergyMax 参数
41476	由于触发了力控保护，拒绝继续执行程序	触发了力控保护	请重新 pptomain 执行程序
41477	指令参数中，存在下限超过上限	指令参数不合法	请检查指令参数设置，修正参数
41478	力控辨识开启失败	开启力控辨识前机器人不处于机械零点	请 jog 到机械零点再开启辨识
42001	导纳运动超过安全范围	超过安全范围	1 请检查安全范围参数 2 请检查导纳控制参数是否合理

17.1.4 5XXXX

错误码	错误描述	故障可能原因	修复方法
50000	转弯区取消	超过转弯区取消阈值	1.增大前后两条轨迹夹角; 2.增大前后两条轨迹长度; 3.增大 zone
50001	控制器当前状态异常	控制器当前状态异常	执行 PPToMain 重置工程或重新加载程序
50002	超出运动范围	1. 目标点在机器人运动范围之外 2. 目标点为笛卡尔奇异位置	1.检查目标点位置 2.用关节方式移动机器人 3.检查 CONFDATA 配置
50003	相邻两个目标点过近	相邻两个目标点过近	检查相邻两条 Move 指令的目标点是否是同一个点
50004	圆弧起点与终点过近，生成圆弧失败,轨迹将被忽略	圆弧起点终点过近，生成圆弧失败,轨迹将被忽略	1、调整圆弧起点或终点点位之间的距离
50005	圆弧起点与辅助点过近，生成圆弧失败	圆弧起点与辅助点过近，生成圆弧失败	1、调整圆弧起点或辅助点点位之间的距离
50006	圆弧终点与辅助点过近，生	圆弧终点与辅助点过近，生成圆	1、调整圆弧终点或辅助点点位之间的距离

	成圆弧失败	弧失败	
50007	圆弧终点与辅助点过近，生成圆弧失败	起始点，辅助点，终点之间夹角过小，一般是由于某两个点之间距离过近导致	1、调整圆弧起点、辅助点、终点点位间之间的距离
50008	圆弧起点、辅助点、终点在一条直线上，生成圆弧失败	圆弧起点、辅助点、终点在一条直线上，生成圆弧失败	1、调整圆弧起点、辅助点、终点点位之间的距离
50009	圆弧半径过小，生成圆弧失败	圆弧起点、辅助点、终点距离过近	1、调整圆弧起点、辅助点、终点点位之间的距离
50010	无法生成合理的圆弧，生成圆弧失败	圆弧起点、辅助点、终点距离过近或圆弧起点、辅助点、终点在一条直线上	1、调整圆弧起点、辅助点、终点点位之间的距离或方位
50011	摆线起点与终点过近，生成摆线失败	摆线起点终点过近，生成摆线弧败	1、调整摆线起点或终点点位之间的距离;2、生成摆线需要满足:所指定半径大于1mm,所指定进给量大于1mm,摆线长度(由起点、辅助点、终点距离决定)大于2倍的半径加上1.5倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于5度
50012	摆线起点与辅助点过近，生成摆线失败	摆线起点与辅助点过近，生成摆线失败	1、调整摆线起点或辅助点点位之间的距离;2、生成摆线需要满足:所指定半径大于1mm,所指定进给量大于1mm,摆线长度(由起点、辅助点、终点距离决定)大于2倍的半径加上1.5倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于5度
50013	摆线终点与辅助点过近，生成摆线失败	摆线终点与辅助点过近，生成失败	1、调整摆线终点或辅助点点位之间的距离;2、生成摆线需要满足:所指定半径大于1mm,所指定进给量大于1mm,摆线长度(由起点、辅助点、终点距离决定)大于2倍的半径加上1.5倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于5度
50014	摆线起点、辅助点、终点在一条直线上，生成摆线失败	起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角小于5度，生成摆线失败	1、调整摆线起点、辅助点、终点点位之间的距离;2、生成摆线需要满足:所指定半径大于1mm,所指定进给量大于1mm,摆线长度(由起点、辅助点、终点距离决定)大于2倍的半径加上1.5倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于5度
50015	输入半径过小，生成摆线失败	输入半径过小，生成摆线失败	1、增大输入半径;2、生成摆线需要满足:所指定半径大于1mm,所指定进给量大于1mm,摆线长度(由起点、辅助点、终点距离决定)大于2倍的半径加上1.5倍的进给量;起点与辅助点所在直线，起点与终点所

			在直线，两条直线之间的夹角大于 5 度
50016	指定的进给量过小，生成摆线失败	指定的进给量过小，生成摆线失败	1、增大输入进给量;2、生成摆线需要满足:所指定半径大于 1mm,所指定进给量大于 1mm,摆线长度(由起点、辅助点、终点距离决定)大于 2 倍的半径加上 1.5 倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于 5 度
50017	指定的小于进给量，生成摆线失败	指定的摆线小于摆线进给量，生成摆线失败	1、指定的半径大小不应小于指定的进给量;2、生成摆线需要满足:所指定半径大于 1mm,所指定进给量大于 1mm,摆线长度(由起点、辅助点、终点距离决定)大于 2 倍的半径加上 1.5 倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于 5 度
50018	摆线长度过短，生成摆线失败	相对于指定的半径和进给量，起点和终点距离较短，不能生成合理的摆线	1、减小指定的半径或进给量；2、增大起点终点间的距离;3、生成摆线需要满足:所指定半径大于 1mm,所指定进给量大于 1mm,摆线长度(由起点、辅助点、终点距离决定大于 2 倍的半径加上 1.5 倍的进给量;起点与辅助点所在直线，起点与终点所在直线，两条直线之间的夹角大于 5 度
50019	生成路径内部错误	生成路径内部错误	1、重新调整目标点的位置、姿态和臂角(仅 7 轴机器人需要考虑臂角)
50020	设定的工具坐标系不符合当前机型特征,生成路径失败	设定的工具坐标系不符合当前机型特征	1、当前机型的工具坐标系必须满足，持有方式:手持、位置:x=0，y=0
50021	超出运动范围	1. 指定 confdata 下目标点无解	1.采用指令 ConfJ off 取消 confdata；2.重新示教点位
50022	不合法的指令	1. 不合法的指令	1.重新示教点位；2.重启机器人
50023	奇异点错误	1. 该目标点没有示教，或目标点 conf 信息出现错误	修改目标点 conf
50024	错误的轨迹类型,当前机器人无法沿着指定的方向平移,请重新示教点位	当前机器人,只能在基座坐标下的 xz 平面内平移,不能偏离基坐标系下的 xz 平面	重新示教点位
50025	错误的轨迹类型,当前机器人无法沿着指定的方向旋转,请重新示教点位	当前机器人,旋转运动的旋转轴必须与基坐标系的 y 轴平行	重新示教点位
50026	等待解析下一条运动指令时间过长，自动取消转弯区	两条运动指令间，插入过多的非运动指令，自动取消转弯区	1.使用 RL 指令 AotoIgnoreZone Off,系统将不会自动取消转弯区;2.简化两条运动指令间的非运动指令
50027	轨迹长度小于最小转弯区半径，自动拼接路径，整合转弯区	1.该轨迹需要前后都衔接转弯区，但是轨迹长度小于最小转弯区半径的两倍;2.该轨迹设定衔接一条转弯区，但是轨迹长度小	该功能可以使得运动更平滑，如想关闭该功能，可将最小转弯区半径设为 0

		于最小转弯区半径	
50028	螺旋线初始半径小于 0，生成螺旋线失败	螺旋线初始半径小于 0，生成螺旋线失败	1、螺旋线初始半径不小于 0;2、生成螺旋线需要满足:螺旋线初始半径不小于 0;旋转增率大于 0.0001mm/deg;合计旋转角度在 0.1~3600deg 之间;3、使用 MoveC 或 MoveCF 指令
50029	螺旋线旋转增率过低，生成螺旋线失败	螺旋线半径变化率过低，生成螺旋线失败	1、增大螺旋线旋转增率;2、生成螺旋线需要满足::螺旋线初始半径不小于 0;旋转增率大于 0.0001mm/deg;合计旋转角度在 0.1~3600deg 之间;3、使用 MoveC 或 MoveCF 指令
50030	螺旋线合计旋转角度非法，生成螺旋线失败	螺旋线合计旋转角度非法，生成螺旋线失败	1、增大螺旋线合计旋转角度;2、生成螺旋线需要满足:螺旋线初始半径不小于 0;半径变化率大于 0.0001mm/deg;合计旋转角度在 0.1~3600deg 之间;
50031	螺旋线轨迹不支持手持工件，生成螺旋线失败	螺旋线轨迹不支持手持工件，生成螺旋线失败	改用手持工具;
50033	错误的轨迹,锁轴状态下,目标点锁轴角度发生了偏离,请调整目标点或关闭锁轴状态	进入锁轴状态后，四轴角度不允许改变，当前轨迹目标点四轴角度不符合锁轴要求	调整轨迹目标点或者关闭锁轴指令，锁轴状态下四轴角度要求为 0 度或 180 度或 -180 度
50034	6 轴机型在锁轴状态下或 5 轴机型无法到达目标点,请更改点位或关闭锁轴	锁轴状态下的 6 轴机型或 5 轴机型,部分轨迹无法到达,当手腕穿越基座上方附近时较为常见	6 轴机型在锁轴状态下或 5 轴机型无法到达目标点,请更改点位或关闭锁轴
50035	锁轴状态下无法生成辨识轨迹,请关闭锁轴	锁轴状态下,部分轨迹无法正确生成	请关闭锁轴
50036	5 轴机器人无法进行负载辨识	5 轴机器人无法进行负载辨识	无
50040	未开启关节位置限制，不允许进行辨识	未开启关节位置限制	请开启关节位置限制
50041	关节位置限制设置不合理，不允许进行辨识	关节位置限制未设置在上/下限 5 度范围内	请将关节位置限制设置在上/下限 5 度范围内
50042	起始点姿态错误，未保证法兰平行基座	起始点姿态错误，未保证法兰平行基座	重新示教点位
50043	目标点姿态错误，未保证法兰平行基座	目标点姿态错误，未保证法兰平行基座	重新示教点位
50044	回撤方向错误，请向工作区方向移动	回撤方向错误，请向工作区方向移动	请向工作区方向移动
50101	关节角度超限	轴角度运动超出运动范围	1.取消关节位置限制 2.手动将机器人各轴移动到正常的工作范围内
50102	轨迹前瞻过程中，遇到奇异点	存在穿越奇异点的轨迹	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.或者将笛卡尔空间运动指令改为关节空间运动指令
50103	笛卡尔路径终点不符合给	无法通过笛卡尔空间规划到达	1.使用 ConfL Off; 2.改为 MoveJ 或

	定的 ConfData	给定 ConfData 下的目标点	MoveAbsJ; 3.更改目标点 ConfData
50104	前瞻中过程中发现关节力矩超过限制	1.负载数值设置过大，超过机器人承受能力; 2.机器人摩擦力系数过大; 3.机器人电机过载系数或传动过载系数过小	1.检查负载数值是否符合实际负载情况; 2.检查机器人摩擦力系数、电机过载系数、传动过载系数等参数; 3.尝试更换指令形式，例如将笛卡尔空间指令更换为关节空间指令;
50105	前瞻点不连续	1.程序逻辑问题; 2.动力学参数错误，存在不在合理范围内的数	1.修改程序逻辑; 2.检查动力学参数; 3.更改点位或转弯区
50106	轨迹起点大于终点，轨迹不合理	程序逻辑问题	程序逻辑问题,修改程序点位，或者修改转弯区
50107	生成错误转弯区轨迹	内部错误	取消转弯区,改变转弯区大小或改变目标点
50108	未及时获取到新的轨迹，运动停止	系统故障	
50109	轨迹队列为空，无法规划	设置轨迹错误	两条反向轨迹间避免生成转弯区或减少转弯区的长度
50110	无法获取程序初始位置	无法获取程序初始位置	1、修改程序逻辑；2、修改第一条轨迹的转弯区大小
50111	前后轨迹位置无法衔接	前后轨迹位置无法衔接	1、重新示教点位；2、修改转弯区大小
50112	轨迹起点或终点位姿有误，无法进行对应的插补运动	轨迹起点或终点位姿有误，无法进行对应的插补运动	1、重新示教位姿;2、如果当前机型为三轴或者四轴机器人,请检查输入位姿与当前机型特征是否相符;3、如果使用了 SingAreaLockAxis4 On 请检查目标是是否符合锁轴要求;4、如果使用了 CompletePostureLerp ON,请检查起点和终点姿态是否符合要求，或者关闭该指令
50113	[奇异规避]路径点相对于非奇异规避路径改变的最大姿态超过用户设定的阈值	[奇异规避]路径点相对于非奇异规避路径改变的最大姿态超过用户设定的阈值	请规避奇异点(相关信息查阅使用手册):1.请重新设置奇异规避姿态改变的阈值;2.重新示教点位,更换目标点;3.使用其他方式的奇异规避
50114	关节角度超限	轴角度运动超出运动范围	1.取消关节位置限制 2.手动将机器人各轴移动到正常的工作范围内
50115	轨迹前瞻过程中，遇到奇异点，请规避奇异点	前瞻中轨迹穿过奇异点	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.或者更换 jog 方式，尝试轴空间 jog; 3.或者将笛卡尔空间运动指令改为关节空间运动指令
50116	轨迹前瞻过程中，遇到奇异点	轨迹衔接处为奇异点	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.或者在报错轨迹后面加入 wait0
50117	[奇异规避]轨迹前瞻过程中，遇到肩关节/肘关节奇异点	轨迹前瞻过程中，发现轨迹中存在肩关节/肘关节奇异点	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.更换奇异规避方式
50118	[奇异规避轨迹前瞻过程中，改变姿态后求解的终点	轨迹前瞻过程中，使用奇异规避方法解析出的终点与设置的终	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.更换奇异规

	与所需要终点不一致	点位置不一致，路径生成失败	避方式
50119	该奇异规避方式与开启导轨不兼容,开启奇异规避失败	该奇异规避方式与开启导轨不兼容,开启奇异规避失败	关闭导轨
50120	[奇异规避]使用奇异规避路径方式，沿路经无法到达终点	[奇异规避]经过轨迹预插补模拟后，沿路经无法到达终点，可能的原因:1.轨迹前瞻中发现轴超限 2.奇异规避算法遇到不可达点失效 3.遇到其他类型奇异点 4.路径点超出运动范围 5.如果 Conf 开启，使用当前配置的终点无法抵达	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.更换奇异规避方式 3.如果 Conf 开启，可以关闭 Conf
50121	[奇异规避]使用奇异规避方法解析路径点由于机器人形态变化过大或靠近边界，未能获取路径点角度，路径生成失败	[奇异规避]使用奇异规避方法解析路径点由于机器人形态变化过大或靠近边界，未能获取路径点角度，路径生成失败	请规避奇异点(相关信息查阅使用手册): 1.重新示教点位,更换目标点; 2.更换奇异规避方式
50122	该奇异规避方式与力控模式/辨识功能不兼容，开启奇异规避失败	该奇异规避方式与力控模式/辨识功能不兼容，开启奇异规避失败	关闭力控模式/辨识功能
50123	该奇异规避方式不支持当前种类路径类型，开启奇异规避失败	该奇异规避方式不支持当前种类路径类型，开启奇异规避失败	相关信息请查阅使用手册: 更换路径类型
50124	大臂和小臂夹角超出安全范围，提前停止运动	该机型依然会对大臂和小臂夹角进行安全检查，发现大臂和小臂夹角将超出安全范围	重新示教点位
50125	大臂和小臂夹角超出安全范围，提前停止运动	该机型依然会对大臂和小臂夹角进行安全检查，发现大臂和小臂夹角将超出安全范围	更改运动方式
50126	前瞻中过程中发现关节速度超过限制	1.关节最大速度设定数值过小	1.恢复默认关节最大速度设定数值;
50127	前瞻中过程中发现关节加速度超过限制	1.关节最大加速度设定数值过小	1.增大关节最大加速度设定数值;
50128	前瞻中过程中发现关节加加速度超过限制	1.关节最大加加速度设定数值过小	1.增大关节最大加加速度设定数值;
50201	笛卡尔路径中遇到不可达点	笛卡尔路径中遇到奇异点	更换轨迹，或通过轴空间运动前往目标位置
50202	笛卡尔路径中遇到非法的位姿	笛卡尔路径中目标点不符合当前机器人构型	重新示教目标点，如果目标点来自手动输入可改为示教点
50203	路径规划错误	内部错误	
50204	规划过程中，发现采样点不足	1、在两条运动指令之间，插入了过多的逻辑判断或者点位计算；2、工控机状态不稳定，请检查是否存在非 xCore 程序在	1、修改 RL 工程，重新开始运行工程； 2、关闭非 xCore 控制系统的 Linux 后台程序

		机器人上运行	
50205	运动规划中相邻位置指令点相差过大	控制器错误，运动规划中相邻位置指令相差过大，超过限制	1.重新示教轨迹目标点位; 2.尝试更改指令形式，例如 MoveL 改为 MoveJ; 3.报错轨迹后面加入 Wait 0 来刷新状态
50206	路径规划内部计算出现错误	控制器规划过程中出现计算错误	1.重新示教轨迹目标点位; 2.尝试更改指令形式，例如 MoveL 改为 MoveJ; 3.尝试更改轨迹转弯区大小、期望速度大小; 4.报错轨迹后面加入 Wait 0 来刷新状态
50207	未停止在路径终点	规划错误,在机器人路径中间某处停止	改变运动指定参数(转弯区大小,目标点位置等),或改变目标点位姿,或改变运动指令(如 MoveL 改为 MoveJ)
50208	轨迹内部错误	轨迹内部错误	1.增大或减小转弯区；2.重新示教点位;
50209	无法按指定的停止距离停止	设定的停止距离过短	增加停止距离
50210	路径回转角度与设定值差异过大	笛卡尔路径实际到达回转角度与点位设定值差别过大	使用 ConfL off 指令避免该警告或按照与起始点相近的回转角度重新示教目标点
50211	规划过程中，发现静止时超过约束	1.负载数值设置过大，超过机器人承受能力; 2.机器人摩擦力系数过大; 3.机器人电机过载系数或传动过载系数过小; 4.规划内部计算错误	1.检查负载数值是否符合实际负载情况; 2.检查机器人摩擦力系数、电机过载系数、传动过载系数等参数; 3.尝试更换指令形式，例如将笛卡尔空间指令更换为关节空间指令; 4.请联系技术支持;
50301	运动规划线程被阻塞,执行超时	运动规划线程被阻塞,执行超时	重新 pptomain 运行工程或者重启控制器
50302	运动规划线程被阻塞,调度超时	系统故障	重新 pptomain 运行工程或者重启控制器
50303	运动规划线程被阻塞，连续超时超过 5000 次	系统故障	重新 pptomain 运行工程或者重启控制器
50304	EtherCAT 线程未收到足够的规划器指令	系统故障	
30400	机器人发生碰撞停止运行，请检查机器人运行环境，确认人员、设备安全后再恢复运行	机器人发生碰撞	1.请检查机器人运行环境，确认人员、设备安全后，重新上电，再恢复运行
30401	机器人发生碰撞停止运行，请检查机器人运行环境，确认人员、设备安全后再恢复运行	机器人发生碰撞	1.请检查机器人运行环境，确认人员、设备安全后，重新上电，再恢复运行；2.检查当前工具(Tool)设置是否与实际一致，设定的工具质量质心是否合理
30402	机器人发生碰撞停止运行，请检查机器人运行环境，确认人员、设备安全后再恢复运行	机器人发生碰撞	1.请检查机器人运行环境，确认人员、设备安全后，重新上电，再恢复运行；2.检查当前工具(Tool)设置是否与实际一致，设定的工具质量质心是否合理；3.关闭肘部力碰撞检测
30403	机器人发生碰撞停止运行，请检查机器人运行环境，确	机器人发生碰撞	1.请检查机器人运行环境，确认人员、设备安全后，重新上电，再恢复运行；2.检查当

	认人员、设备安全后再恢复运行		前工具(Tool)设置是否与实际一致，设定的工具质量质心是否合理；3.关闭 TCP 力碰撞检测
50401	最大出力限制参数设置错误	最大出力限制参数超过减速器最大启停力矩，参数设置失败	重新设置参数
50501	工具工件坐标系设置冲突	设置同时手持工件、手持工具	选择正确的 tool、wobj
50502	工具工件坐标系设置冲突	设置同时外部工件、外部工具	选择正确的 tool、wobj
50503	工具工件坐标系设置冲突	设置同时手持工件、手持工具	选择正确的 tool、wobj
50504	工具工件坐标系设置冲突	设置同时外部工件、外部工具	选择正确的 tool、wobj
50506	运动模块第一类内部模式错误	内部问题	重启机器人可解决，并请联系技术支持
50507	运动模块第二类内部模式错误	内部问题	重启机器人可解决，并请联系技术支持
50508	重置模块状态失败	内部问题	重启机器人可解决，并请联系技术支持
50509	设置运动起点失败	内部问题	重启机器人可解决，并请联系技术支持
50510	生成轨迹失败	内部问题	重启机器人可解决，并请联系技术支持
50511	执行运动指令失败	机器人当前状态，不允许开始运动	1.停止程序，并重新开始运行工程;2.或重启机器人;3.如果上述方法不能解决请尽快联系客户支持
50512	轴数量不匹配	软件问题	请联系技术支持
50513	速度设置无效	软件问题	请联系技术支持
50514	步长设置无效	软件问题	请联系技术支持
50515	坐标系设置无效	软件问题	请联系技术支持
50516	运动轴设置无效	软件问题	请联系技术支持
50517	末端快速调整失败，请重新示教点位	1.处于特殊或者极限位置，无法使用末端快速调整	在机器人正常工作范围内重新示教点位
50518	目标点位可能是当前点	目标点位可能是当前点	无
50519	生成轨迹失败，目标点不符合要求	目标点位可能超出机器人工作范围	请在机器人正常工作范围内，重新示教点位
50520	错误的 JOG 方式，当前机器人无法沿指定的方向平移，请修改 JOG 方式	当前机器人,只能在基座坐标下的 xz 平面内平移,不能偏离基坐标系下的 xz 平面	建议优先选择基坐标系或关节坐标系进行 JOG
50521	错误的 JOG 方式，当前机器人无法沿指定的方向旋转，请修改 JOG 方式	当前机器人,旋转运动的旋转轴必须与基坐标系的 y 轴平行,或沿法兰坐标系 Z 轴旋转	建议优先选择基坐标系或关节坐标系进行 JOG
50522	错误的轨迹，当前机器人不支持手持工件,外部工具,请选用手持工具	1 当前机器人为 4 轴机器人不支持手持工件,外部工具，2.开启了锁轴状态，不支持手持工件	改用手持工具
50523	机器人已经处于当前轨迹终点	点击下一步时,机器人已经处于当前轨迹终点，机器人不需要运动	请确认是否想让机器人处于当前轨迹终点，如果需要机器人移动至下一条轨迹终点，请继续点击下一步
50524	JOG 开启失败，缺少关键运动参数	当前机器人配置文件缺少关键运动参数，不允许进行 JOG 操作	请确认机器人配置文件是否更新

50525	机器人当前四轴角度不为0或者正负180度,无法沿当前方式jog，请先调整四轴角度	当前机器人四轴角度为0或者正负180度,才能开启奇异规避模式的jog	调整四轴角度至0或者正负180度
50526	当前机器人法兰与基座不平行,请先jog J4,再jog Ry使法兰与基座平行	当前机器人法兰与基座不平行，不允许jog X Y Z	请先jog J4,再jog Ry使法兰与基座平行
50527	错误的轨迹,开启锁轴后,机器人笛卡尔运动不支持手持工件,外部工具,请选用手持工具	.开启了锁轴状态，不支持手持工件	改用手持工具
50528	当前机器人法兰与基座不平行,请jog Ry使法兰与基座平行	当前机器人法兰与基座不平行，不允许jog X Y Z	请jog Ry使法兰与基座平行
50529	负载设置无效	当前负载超过最大负载限制	请检查负载设置
50530	按钮点击过快	上次运动还未完全停止，无法开始本次运动	延长两次运动之间的时间间隔，避免过快点击
50531	末端快速调整失败，请选择正确的工具或正确的坐标系	1.工具为外部工具，调整基于坐标系为机器人基坐标系	选择正确的工具或正确的坐标系
50532	JOG无效	精度补偿状态开启时，无法开启奇异规避与平行基座jog	关闭精度补偿状态
50601	动力学未打开，力矩前馈置零，可能会带来上电点头	关闭力矩前馈，力矩前馈被置零	开启力矩前馈
50602	摩擦力辨识完成	摩擦力辨识完成	摩擦力辨识完成
50603	振动抑制开启失败	1. 当前机型不支持振动抑制；2. 关节柔性系数过小	修改关节柔性系数到合理范围内
50604	重力补偿开启失败	当前机型不支持重力补偿	当前机型不支持重力补偿
50605	碰撞检测延迟补偿参数辨识开启失败，机器人动力学前馈未开启	机器人动力学前馈未开启	请开启机器人动力学前馈
50701	指令传输时丢失	指令传输时丢失	重新开始运动
50702	配置文件与控制器版本不匹配	配置文件与控制器版本不匹配	请前往sw网站下载对应的最新版本配置文件，并将该配置文件升级至控制器
50801	摆动模式下轨迹类型错误	摆动不支持轴空间轨迹、纯旋转轨迹和摆线轨迹	不使用轴空间轨迹、纯旋转轨迹和摆线轨迹
50802	JOG时网络断联或网络不稳定，停止机器人运动。	在进行JOG、运动至或快速调整时，由于网络断联或网络不稳定，为防止机器发生碰撞，停止机器人运动	请检查网络连接是否正常、确保网络环境稳定、重新尝试JOG、运动至或快速调整操作

17.1.5 6XXXX

错误码	错误描述	故障可能原因	修复方法

60000	RL 指令参数错误	RL 指令参数错误	修改指令 输入正确的参数
60001	SocketReadBit 的参数错误	RL 运行时遇到错误	SocketReadBit 的参数必须是 8 的倍数
60002	SocketReadBit 读取数据失败	RL 运行时遇到错误	SocketReadBit 读取数据失败(数据长度不匹配)
60003	SocketReadDouble 读取数据非法	RL 运行时遇到错误	SocketReadDouble 读取数据非法
60004	SocketReadDouble 读取数据失败	RL 运行时遇到错误	SocketReadDouble 读取数据失败(数据长度不匹配)
60005	加载 RL 工程失败	1.rsync 配置文件错误或者丢失 2.hmi 版本不匹配 3.RL 工程文件被意外修改 4.网络故障	1.配置正确的运行环境 2.检查对应的 HMI 版本 3.联系技术支持
60006	机器人运行中，pptomain 失败	机器人正在运行，不允许 pptomain	使用暂停或者急停使机器人停止运行
60007	关节位置限制未开启，不允许单步调试	关节位置限制未开启，不允许单步调试	在设置界面打开关节位置限制设置
60008	运行速度同步失败，取消下一步	运行速度同步失败，取消下一步	暂停并重启控制器
60009	RL 正在运行或执行器错误状态，拒绝下一步	RL 正在运行或执行器错误状态，拒绝下一步	若 RL 正在运行，等待 RL 运行停止 或者点击暂停；若执行器错误状态，需点击 pp_to_main 重新执行
60010	启动解释器失败，无法执行下一步	启动解释器失败，无法执行下一步	重启控制器或者联系技术支持
60011	RL 任务执行过程中出错，该任务停止	RL 任务执行过程中出错，该任务停止	切换到对应任务检查程序指针附近的逻辑错误，排除错误后 pptomain 重新运行
60012	GetSocketConn 失败	连接不存在或者已经断开	在 SocketConnect 连接后再执行该指令
60013	GetSocketServer 失败	连接不存在或者已经断开	在 SocketServer 进入监听状态后再执行该指令
60014	无法执行开始 RL 程序操作	以下原因会导致开始运行程序失败: 1)机器人没有上电; 2)机器人已经处于运行中	确保机器人上电并处于空闲中
60015	Reload 工程失败，工程数据超过阈值	1.全局变量累计超过 14000 个	删除过多的 RL 变量
60100	HexToDec 的参数错误	RL 运行时遇到错误	HexToDec 输入的字符串必须是 16 进制整数
60101	StrToByte 的参数错误	RL 运行时遇到错误	StrToByte(string, \Hex) 输入的字符串必须是 16 进制整数
60102	StrToByte 的参数错误	RL 运行时遇到错误	StrToByte(string) 输入的字符串必须是 10 进制整数
60103	StrToByte 的参数错误	RL 运行时遇到错误	StrToByte(string, \Okt) 输入的字符串必须是 8 进制整数
60104	StrToByte 的参数错误	RL 运行时遇到错误	StrToByte(string, \Bin) 输入的字符

			串必须是 2 进制整数
60105	StrToByte 的参数错误	RL 运行时遇到错误	StrToByte(string, \Char) 输入的字符串必须是 ASCII 字符
60106	StrToByte 的结果溢出	RL 运行时遇到错误	StrToByte(string HEX) 输入的字符串数据溢出
60107	StrToByte 的结果溢出	RL 运行时遇到错误	StrToByte(string DEC) 输入的字符串数据溢出
60108	StrToByte 的结果溢出	RL 运行时遇到错误	StrToByte(string OKT) 输入的字符串数据溢出
60109	StrToByte 的结果溢出	RL 运行时遇到错误	StrToByte(string BIN) 输入的字符串数据溢出
60110	StrToByte 的结果溢出	RL 运行时遇到错误	StrToByte(string CHAR) 输入的字符串数据溢出
60111	StrPart 截取字符串的范围超限	RL 运行时遇到错误	StrPart 截取字符串的范围超限
60112	StrMatch 匹配参数过大，无法查找	RL 运行时遇到错误	StrMatch 匹配参数过大，无法查找
60113	BitCheck 指令的 BitPos 参数值范围超限	RL 运行时遇到错误	BitCheck 指令的 BitPos 参数值范围超限
60114	BitClear 指令的 BitPos 参数值范围超限	RL 运行时遇到错误	BitClear 指令的 BitPos 参数值范围超限
60115	BitLsh 指令的 ShiftSteps 参数值范围超限	RL 运行时遇到错误	BitLsh 指令的 ShiftSteps 参数值范围超限
60116	BitRsh 指令的 ShiftSteps 参数值范围超限	RL 运行时遇到错误	BitRsh 指令的 ShiftSteps 参数值范围超限
60117	BitSet 指令的 BitPos 参数值范围超限	RL 运行时遇到错误	BitSet 指令的 BitPos 参数值范围超限
60198	指令执行失败		
60199	指令执行成功		
60200	FcInit 执行时遇到错误	FcInit 执行时遇到错误	修改 FcInit 指令为正确的指令参数
60201	FcStart 执行时遇到错误	FcStart 执行时遇到错误	1.请先执行 FcInit 指令;2.请设置正确的负载参数;3.请标定力传感器零点;4.请将机器人移动至远离关节位置限制边界的位置或适当增大软限位范围
60202	FcPause 执行时遇到错误	FcPause 执行时遇到错误	请先执行 FcStart 指令
60203	FcRestart 执行时遇到错误	FcRestart 执行时遇到错误	请先执行 FcPause 指令
60204	FcStop 执行时遇到错误	FcStop 执行时遇到错误	修改 FcStop 指令为正确的指令参数
60205	ClearFcError 执行时遇到错误	ClearFcError 执行时遇到错误	修改 ClearFcError 指令为正确的指令参数
60206	SetControlType 执行时遇到错误	SetControlType 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStart 指令前执行该指令;3.请修改 SetControlType 指令为正确的指令参数

60207	SetJntCtrlStiffVec 执行时遇到错误	SetJntCtrlStiffVec 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetJntCtrlStiffVec 指令为正确的指令参数
60208	SetCartCtrlStiffVec 执行时遇到错误	SetCartCtrlStiffVec 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetCartCtrlStiffVec 指令为正确的指令参数
60209	SetCartNSStiff 执行时遇到错误	SetCartNSStiff 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetCartNSStiff 指令为正确的指令参数
60210	SetLoad 执行时遇到错误	SetLoad 执行时遇到错误	1.请检查负载参数是否超过机器人最大负载;2.请检查负载质心设置是否超过 0.3m;3.请修改 SetLoad 指令为正确的指令参数
60211	StartOverlay 执行时遇到错误	StartOverlay 执行时遇到错误	请先执行 FcStart 指令
60212	StopOverlay 执行时遇到错误	StopOverlay 执行时遇到错误	请先执行 StartOverlay 指令
60213	PauseOverlay 执行时遇到错误	PauseOverlay 执行时遇到错误	请先执行 StartOverlay 指令
60214	SetSineOverlay 执行时遇到错误	SetSineOverlay 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetSineOverlay 指令为正确的指令参数
60215	SetLissajousOverlay 执行时遇到错误	SetLissajousOverlay 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetLissajousOverlay 指令为正确的指令参数
60216	SetJntTrqDes 执行时遇到错误	SetJntTrqDes 执行时遇到错误	1.请先执行 FcStart 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetJntTrqDes 指令为正确的指令参数
60217	SetCartForceDes 执行时遇到错误	SetCartForceDes 执行时遇到错误	1.请先执行 FcStart 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetCartForceDes 指令为正确的指令参数
60218	RestartOverlay 执行时遇到错误	RestartOverlay 执行时遇到错误	请先执行 PauseOverlay 指令
60219	SetSensorUseType 执行时遇到错误	SetSensorUseType 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令;3.请修改 SetSensorUseType 为正确参数
60220	CalibSensorError 执行时遇到错误	CalibSensorError 执行时遇到错误	1.请先执行 FcInit 指令;2.请在 FcStop 指令前执行该指令
60221	FcCondForce 执行时遇到错误	FcCondForce 执行时遇到错误	1.请先执行 FcStart 指令;2.请在 FcStop 指令前执行该指令;3.请修改

			FcCondForce 指令为正确的指令参数
60222	FcCondPosBox 执行时遇到错误	FcCondPosBox 执行时遇到错误	1.请先执行 FcStart 指令;2.请在 FcStop 指令前执行该指令;3.请修改 FcCondPosBox 指令为正确的指令参数
60223	FcCondTorque 执行时遇到错误	FcCondTorque 执行时遇到错误	1.请先执行 FcStart 指令;2.请在 FcStop 指令前执行该指令;3.请修改 FcCondTorque 指令为正确的指令参数
60224	FCCondWaitWhile 执行时遇到错误	FCCondWaitWhile 执行时遇到错误	修改 FCCondWaitWhile 指令为正确的指令参数
60225	FcCondOrient 执行时遇到错误	FcCondOrient 执行时遇到错误	修改 FcCondOrient 指令为正确的指令参数
60226	FcCondPosSphere 执行时遇到错误	FcCondPosSphere 执行时遇到错误	修改 FcCondPosSphere 指令为正确的指令参数
60227	FcCondPosCylinder 执行时遇到错误	FcCondPosCylinder 执行时遇到错误	修改 FcCondPosCylinder 指令为正确的指令参数
60228	FcCondTcpSpeed 执行时遇到错误	FcCondTcpSpeed 执行时遇到错误	修改 FcCondTcpSpeed 指令为正确的指令参数
60229	CalcJointT 逆解失败	笛卡尔坐标的 conf 错误	更改输入的笛卡尔坐标
60300	ReplayPath 回放速率超限	RL 运行时遇到错误	ReplayPath 回放速率超限
60400	RL 调用层级超过限制	RL 运行时遇到错误	RL 调用层级超过限制
60500	不能加载多个运动任务	不能加载多个运动任务	只指定一项任务为运动任务
60501	前置任务错误	前置任务不能为自己	修改前置任务为其它任务
60502	运动任务不允许设置前置任务	运动任务不允许设置前置任务	运动任务不允许设置前置任务
60503	运动指令不能同时手持工件和工具	运动指令不能同时手持工件和工具	修改指令 tool 和 wobj 参数
60504	Search 指令无效输入信号	该信号类型不是 DI 信号或不是只读寄存器	使用正确的 DI 信号/只读寄存器
60505	Search 指令记录点位失败	工具和工件同时为手持或外部	修改参数，使用位置不一样的工具和工件
60506	Search 指令 DI 信号/寄存器错误	该 DI 信号不存在，或已被设置为系统输入；或寄存器不存在/绑定为功能码	修改参数，使用未被设置为系统输入的 DI 信号，或为绑定功能的寄存器
60507	SearchL \Stop 指令，速度大于 v100	仅当速度小于等于 v100 时可让机器人快速停止	使用更小的速度参数
60508	SearchC \STOP 指令，速度大于 v100	仅当速度小于等于 v100 时可让机器人快速停止	使用更小的速度参数
60509	程序状态错误，不能执行下一步	1.运动指令点位无法到达; 2.运动经过奇异点;	1.检查点位列表; 2.优化运动轨迹; 3.点击 PPtoMain
60510	RL 指令运动速度过慢	tcp 小于 1μm/s 或 ori 小于 1e-6°/s 或 jnt 小于 1%;	运动指令更换合适的速度参数
60511	拖动回放路径丢失/不存在	拖动回放路径丢失/不存在	录制/使用有效回放路径

60512	拖动回放功能错误	拖动回放功能错误	检查拖动回放功能是否有效
60513	HomeSet 指令范围错误	设置 home 点位角度超出限制	修改 Home 点位的角度到机器人硬限位范围内
60514	Home 参数保存失败	机器人配置模块错误	尝试重启机器人或者联系厂家
60515	HomeSetAt 获取失败	HomeSetAt 获取的轴不存在	更改 HomeSetAt 的参数
60516	HomeClr 指令失败	机器人配置模块错误	尝试重启机器人或者联系厂家
60517	HomeSet 参数数量错误	HomeSet 参数数量错误	修改参数个数，应与机器人轴数一致
60518	Hordr 参数数量错误	Hordr 参数数量错误	修改 RL 文件 传输正确的 Hordr 参数
60519	Hordr 参数值错误	Hordr 参数值错误	修改 RL 文件 传输正确的 Hordr 参数
60520	HordrAt 参数值错误	HordrAt 参数值错误	修改 RL 文件 传输正确的 Hordr 参数
60521	GetEndtoolTorque 工具工件不兼容	工具工件同时为外部或手持	修改 GetEndtoolTorque 指令，传输正确的 tool,wobj 参数
60522	GetEndtoolTorque 末端坐标系类型错误	未定义的末端坐标系类型	修改 GetEndtoolTorque 指令，传输正确的坐标系类型参数：0~2
60523	AccSet 参数超出限制范围 30% ~ 100%	AccSet 加速度，加加速度都限制 30% ~ 100%	修改 AccSet 指令参数
60524	四元数参数错误	变量的四元数的平方和应该等于 1	修改对应的四元数为正确值
60525	OpenDev 端口错误	端口可用范围[0, 65535]	修改 OpenDev 端口为正确值
60526	控制器处于错误状态，启动程序失败，请检查最近生成的错误记录	控制器处于错误状态，启动程序失败，请检最近生成的错误记录	1.根据错误记录修复 RL 程序中的语法、逻辑错误；2.pp_to_main 后重新执行
60527	Trigger 指令参数错误	设置的 TrigData 变量不存在	创建好要修改的目标变量，并用 TrigVar 指令设置
60528	Trigger 指令参数错误	trigdata 设置的 DO/GO/寄存器变量不存在；或未设置输出信号；或寄存器为只读寄存器	创建好输出信号/可写寄存器，并用 TrigIO/TrigReg 指令设置
60529	启动 RL 程序无效，机器人正忙	1、机器人正在运动，不能重复启动；2、正在解析 RL 文件，不能启动	等待寄存器功能码 "sta_robot_is_busy"为 0 后再发启动指令
60530	用户坐标系命名冲突	导入的用户坐标系和已有坐标系信息命名重复冲突，重复的坐标系信息可能无法正确使用	检查用户坐标系列表，保留需要的数据，删除重复的坐标系信息
60531	料盘数据更新失败	1、料盘名称或工件号错误；2、从料盘模块获取数据失败；3、料盘变量值更新失败	1、检查"TrayUpdate"函数传入参数是否有误；2、联系售后处理；
60532	料盘工件数量获取失败	1、料盘名称错误；2、从料盘模块获取数据失败；	1、检查料盘名称是否有误；2、联系售后处理；
60533	码垛数据更新失败	1、码垛名称或工件号错误或层号错误；2、从码垛模块获取数据失败；3、码垛变量值更新失败	1、检查"PalletUpdate"函数传入参数是否有误；2、联系售后处理；
60534	获取码垛层数失败，名称 "$arg" 不存在	码垛名称错误	请检查"PalletLayerCount"函数传入参数

60535	获取工件数量失败	码垛名称或层号错误	请检查"PalletWobjCount"函数传入参数
60536	数据库中没有 Pers 数据，使用变量初值	数据库中没有 Pers 数据，使用变量初值	数据库中没有 Pers 数据，使用变量初值
60537	无效的工具、工件或用户坐标系	工具或工件没有成功加载，可能的原因：1、工具或其负载四元数参数错误；2、工件或其关联用户坐标系四元数参数错误；3、工件关联用户坐标系失败，需要重新编辑工件信息	编辑并更新相关工具、工件或用户坐标系
60538	运动规划参数错误	运动规划参数错误	重新设置正确的运动参数
60600	激光焊接获取参数失败	激光焊接配置参数获取失败	1.检查激光焊接工艺参数表是否创建。2.检查激光焊接工艺参数表对应参数是否有误。3.激光焊接内部错误，请联系厂家处理
60601	激光焊接设置参数失败	激光焊接设置参数出错	1.检查激光焊接端口配置，确认出错信号的寄存器或 IO 配置是否有误
60602	激光焊接单步后，请先执行 PPToMain 操作，再继续运行	激光焊接单步后，不允许切换继续运行	激光焊接单步后，不允许切换继续运行，请先执行 PPToMain 操作
60603	激光焊接不支持从光标处运行	激光焊接不支持从光标处运行	激光焊接不支持从光标处运行
60604	激光焊接工艺文件不存在	激光焊接工艺文件不存在	请检查激光焊接指令中的工艺文件是否已经创建
60605	当前运动目标点与激光焊接起点或终点重合	当前运动目标点与激光焊接起点或终点重合，激光焊接时序无法正常触发	1.检查出错运动指令目标点，使目标点与当前机器人位置不重合
60606	激光焊接功能已关闭，请勿使用激光焊接指令	激光焊接功能开关已关闭	1.打开激光焊接功能开关 2.删除激光焊接指令
60607	无法响应外部程序启动信号，当前控制器存在错误报警，需要手动清除报警后再启动	控制器存在错误报警	手动清除报警后再启动
60608	通过 socket 指令切换工程失败	1.切换不存在的工程 2.有非半静态任务正在运行	1.切换存在的工程 2.停止正在运行的工程再进行工程切换
60609	通过 socket 指令切换工程成功	切换工程成功	切换工程成功
60610	当前有绑定了程序暂停功能的寄存器或系统 IO 没有复位，不允许启动程序	当前有绑定了程序暂停功能的寄存器或系统 IO 没有复位	将绑定了程序暂停功能的寄存器和系统 IO 都复位
60611	正在生成诊断数据，启动程序失败	正在生成诊断数据，计算资源紧张，此时不能运行程序	等待 10 秒后再次尝试启动
60612	配置工具工件冲突，使用最后一次正确配置	示教器(HMI)右上角配置的工具工件，同时手持或同时不手持，机器人无法计算	正确配置手持工具或者手持工件，勿产生冲突
60613	配置工具工件冲突，无法运动	示教器(HMI)右上角配置的工具工件，同时手持或同时不手持，	正确配置手持工具或者手持工件，勿产生冲突

		机器人无法计算	
60614	未开启关节位置限制，启动程序失败	未开启关节位置限制	打开关节位置限制
60615	按下了软关机按钮，启动程序失败	按下了软关机按钮	不建议此状态下启动程序
60700	Jodell 设备的指令执行失败	末端与 Jodell 设备的通信异常导致指令执行失败	检查硬件连接，重新尝试执行
60701	RM 设备的指令执行失败	末端与 RM 设备的通信异常导致指令执行失败	检查硬件连接，重新尝试执行
60702	四轴锁定开启失败	不满足四轴锁定开启条件导致开启失败	请检查第四轴当前角度是否为 0°或 180°
60703	四轴锁定关闭失败	不满足四轴锁定关闭条件导致关闭失败	
60704	SingAreaWrist 开启失败	不满足 SingAreaWrist 开启条件导致开启失败	请检查当前状态是否满足 SingAreaWrist 开启状态
60705	SingAreaWrist 关闭失败	不满足 SingAreaWrist 关闭条件导致关闭失败	
60706	SingAreaJointWay 开启失败	不满足 SingAreaJointWay 开启条件导致关闭失败	请检查当前状态是否满足 SingAreaJointWay 开启状态
60707	SingAreaJointWay 关闭失败	不满足 SingAreaJointWay 关闭条件导致关闭失败	
60708	非标准六轴构型或 CR 六轴构型的机器人不允许使用 SingAreaLockAxis4 指令	非标准六轴构型或 CR 六轴构型的机器人不允许使用 SingAreaLockAxis4 指令	不使用 SingAreaLockAxis4 指令或者更换机器人
60709	非标准六轴构型或 CR、ER 六轴构型的机器人不允许使用 SingAreaWrist 指令	非标准六轴构型或 CR、ER 六轴构型的机器人不允许使用 SingAreaWrist 指令	不使用 SingAreaWrist 指令或者更换机器人
60710	非标准六轴构型的机器人不允许使用 SingAreaJointWay 指令	非标准六轴构型的机器人不允许使用 SingAreaJointWay 指令	不使用 SingAreaJointWay 指令或者更换机器人
60711	非运动任务中不允许使用 SingAreaLockAxis4 指令	非运动任务中不允许使用 SingAreaLockAxis4 指令	将 SingAreaLockAxis4 指令移动到运动任务中
60712	非运动任务中不允许使用 SingAreaWrist 指令	非运动任务中不允许使用 SingAreaWrist 指令	将 SingAreaWrist 指令移动到运动任务中
60713	非运动任务中不允许使用 SingAreaJointWay 指令	非运动任务中不允许使用 SingAreaJointWay 指令	将 SingAreaJointWay 指令移动到运动任务中
60714	无法恢复连续运行，发生过特殊状态下的急停或者碰撞检测	1、急停规划超时，控制器异常； 2、重回路径过程中发生急停或者碰撞	PPToMain 重新运行
60715	设备初始化执行成功	检查初始化成功	初始化成功，可以控制末端工具运动
60716	设备初始化执行失败	硬件型号不匹配或连接有误	检查连线或查看工具型号是否匹配
60717	Dh 设备的指令执行失败	末端与 Dh 设备的通信异常导致指令执行失败	检查硬件连接，重新尝试执行
60800	不支持的运动类型	传送带工件仅支持 MoveL 和	更改为 MoveL 或 MoveC 运动类型

		MoveC 运动类型	
60801	传送带、导轨、变位机等功能，不支持步退调试	传送带、导轨、变位机等功能，不支持步退调试	使用 ppto 光标、JOG 等方式调试程序
60802	当前控制器状态不允许步退或者运行模式切换。	当前控制器状态不允许步退或者模式切换。	执行 pptoline 操作重置指针位置。
60803	当前调试任务不是运动任务，不允许步退	当前调试任务不是运动任务，不允许步退	在 RL 编辑页面选择运动任务后重试
60804	任务已经步退完成，无法继续步退	任务已经步退完成，无法继续步退	使用 pptoline、pptomain 等方式调试程序
60805	任务已经运行完成，或者程序指针处于不能切换模式的状态	任务已经运行完成，或者程序指针处于不能切换模式的状态	使用 pptoline、pptomain 等方式调试程序
60806	步退模式第一条指令，规划失败	步退模式第一条指令，规划失败	重新示教点位
60807	步退模式运动指令规划失败	运动指令规划参数错误	1.重新示教点位 2.检查 RL 程序输入是否正确
60900	不支持的工具工件	工具工件不能同时为手持或外部	
60901	Robotiq 2F_85 初始化失败	1、末端与 Robotiq 设备的通信异常导致指令执行失败；2、Robotiq 设备 ID 错误	1、检查硬件连接，重新尝试执行；2、输入正确 ID，重新尝试执行
60902	Robotiq 2F_85 获取设备状态失败		
61000	碰撞检测力矩阈值设置失败	不满足条件，碰撞检测力矩阈值设置失败	
61001	碰撞检测未开启，不可使用 MotionSup on 指令	碰撞检测未开启，不可使用 MotionSup on 指令	开启碰撞检测
61002	读取包络信息错误	未正确读取到包络信息	
61003	获取码垛名称失败	码垛编号错误	请检查"GetPalletName"函数传入参数
61004	触发中断失败，任务不存在	任务不存在	
61005	触发中断失败，非运动任务不触发中断	非运动任务不触发中断	请在运动任务中设置中断
61006	触发中断失败，存在已触发但未执行的中断	存在已触发但未执行的中断	请在一个中断完成后触发另一个中断
61007	触发中断失败，单步模式 未选择\DEBUG 选项，此次中断不触发	单步模式 未选择\DEBUG 选项，此次中断不触发	如果要在单步模式触发中断，请选择\DEBUG 选项
61008	触发中断失败，连续模式-暂停状态下，中断不触发	连续模式-暂停状态下，中断不触发	如果要在暂停状态下触发中断，请选择\DEBUG 选项，并切换单步模式
61009	触发中断失败，任务正在重置，此次中断不触发	任务正在重置，此次中断不触发	
61010	触发中断失败，任务正在执行中断，此次中断不触发	任务正在执行中断，此次中断不触发	请在一个中断完成后触发另一个中断
61011	触发中断失败，中断函数不存	中断函数不存在	请检查中断函数是否存在

	在		
61012	触发中断失败，解释器协程已满	解释器协程已满	
61013	触发中断失败，任务运行完 endproc	任务运行完 endproc	请在任务运行时触发中断
61014	触发中断失败,中断被关闭，注意：若是单次触发，则此次之后不再响应中断	中断被 IDisable	请 IEnable 后再触发中断
61015	执行了中断函数内不允许执行的指令	执行了中断函数内不允许执行的指令	请勿在中断函数内执行不允许执行的指令
61016	中断中满足 WaitUntil	中断中满足 WaitUntil	
61017	触发中断失败		
61018	程序未同步至控制器导致启动程序失败	HMI 中程序更改，未同步到控制器，禁止通过系统 IO、寄存器功能码、外部通信启动程序	请通过示教器点击 pptomain 或 reload 操作后，再执行启动程序操作
61019	使用工具不一致导致配方运行失败	工程使用工具配置被更改，但未同步更新配方文件	更新配方以匹配工程使用工具，然后重新运行程序
61020	振动抑制指令执行失败	该机型暂不支持振动抑制功能	
61030	运动指令执行失败	PathRecStart 已开启路径记录，指令不能被记录	PathRecStart 和 PathRecStop 之间不能执行无法记录的运动指令
61031	记录路径相关指令执行失败		
61032	码垛数据更新失败	1、码垛名称或工件号错误或层号错误；2、从码垛模块获取数据失败；3、码垛变量值更新失败	1、检查"PalletUpdateByUniversal"函数传入参数是否有误；2、联系售后处理；
61033	获取码垛层数失败，名称"$arg" 不存在	码垛名称错误	请检查"PalletLayerCountByUniversal"函数传入参数
61034	获取工件数量失败	码垛名称或层号错误	请检查"PalletWobjCountByUniversal"函数传入参数

ROKAE

珞石机器人

400-010-8700
www.rokae.com
sales@rokae.com