



ER SERIES INDUSTRIAL ROBOT **OPERATION MANUAL**

ER系列工业机器人操作手册

修订记录

序号	日期	版本	描述
1	2023.03	V1.18	1. 调整文档格式，新增手册索引、更新控制柜内容 2. 更新 “2.3.6 模式开关介绍” 3. 更新 “3.3 启停时序图” 4. 新增 “4.1.7 写保护” 5. 新增 “4.1.8 程序信息” 6. 更新 “4.2.3 其他操作” 增加 “MovJ 和 MovL 快速切换功能、反向复制/粘贴功能、运动执行指针” 7. 更新 “4.4 IO 检测” 8. 更新 “4.6 日志管理” 添加 “帮助” 功能 9. 更新 “4.7.1 用户管理” 用户角色名称更新 10. 更新 “4.7.2 通用设置>基本设置” 新增 “远程设置、程序设置、系统时间” 11. 更新 “4.7.3 高级设置>本地设置” 新增 “长时间待机熄屏、示教器重启” 12. 新增 “5.2 位置数据类型” 章节中 “APOSARRAY、CPOSARRAY” 13. 新增 “5.3 系统数据类型” 章节中 “INTERRUPT、COLLIPARAM” 14. 更新 “6.1 运动指令” 章节中过渡方式描述；更新 “MovJOffset、MovLOffset、MovLSearch、MovJSearch” 新增 “MovLArch、OnDistance、OnParameter” 15. 新增 “6.2 控制指令” 章节中 “For 循环指令” 16. 更新 “6.5 设置指令” 中 “SetRtToErr 指令描述支持中英文”；新增 “SetOriMode、SetColliAutoTune、SetColliParam” 指令 17. 新增 “6.8 中断指令” 18. 新增 “6.11 区域指令” 中 “AreaConnectIO、AreaDisConnectIO、PolyhedronAreaConnectIO、PolyhedronAreaDisConnectIO” 19. 新增 “6.18 负载在线识别指令” 20. 新增 “6.21 其他函数” 21. 更新 “7.1 多区域监控” 22. 新增 “7.3 碰撞回退” 23. 更新 “7.27 保养提示” 24. 新增 “7.33 中断功能” 25. 新增 “7.34 自动掉励磁” 26. 新增 “7.35 负载在线识别” 27. 更新 “9.1 示教器权限分配表” “9.3 告警信息一览表”

目录

修订记录.....	i
目录.....	i
前言.....	vi
概述	vi
本书常用词汇	vi
符号约定	vi
手册索引	vii
安全注意事项.....	viii
第 1 章安全.....	1-1
1.1 保障安全	1-1
1.2 专门培训	1-1
1.3 机器人使用说明书清单.....	1-1
1.4 操作人员安全注意事项.....	1-2
1.5 机器人的安全注意事项.....	1-3
1.5.1 安装及配线安全	1-3
1.5.2 作业区安全	1-4
1.5.3 操作安全	1-5
1.6 移动及转让机器人的注意事项.....	1-6
1.7 废弃机器人的注意事项.....	1-6
第 2 章了解产品.....	2-1
2.1 产品确认	2-1
2.1.1 装箱内容确认	2-1
2.1.2 订货号确认	2-1
2.2 控制柜	2-1
2.3 示教编程器	2-3
2.3.1 外观	2-3
2.3.2 触摸笔	2-4
2.3.3 USB 插槽	2-4
2.3.4 按键	2-4
2.3.5 显示	2-5
2.3.6 模式开关介绍	2-8
2.3.7 急停按钮介绍	2-9
2.3.8 伺服使能开关介绍	2-9
2.3.9 示教编程器内部的接线端子信号定义.....	2-9
2.3.10 示教编程器线缆	2-10
第 3 章产品使用.....	3-1
3.1 连接方法	3-1
3.2 操作说明	3-1
3.2.1 机器人的坐标系	3-1
3.3 启停时序	3-5
3.3.2 示教	3-5

3.3.3 再现	3-14
第 4 章系统功能介绍.....	4-1
4.1 工程管理	4-1
4.1.1 新建工程或程序	4-3
4.1.2 删除工程或程序	4-4
4.1.3 重命名工程或程序	4-5
4.1.4 复制/粘贴工程或程序	4-5
4.1.5 加载/注销工程或程序	4-6
4.1.6 自启动程序	4-6
4.1.7 写保护	4-6
4.1.8 管理（工程导入导出）	4-6
4.1.9 查找工程/程序	4-8
4.1.10 程序信息	4-9
4.2 程序编辑	4-10
4.2.1 新增指令	4-11
4.2.2 撤销指令	4-12
4.2.3 其它操作	4-14
4.3 程序数据	4-19
4.3.1 新增变量	4-19
4.3.2 标准区域变量	4-1
4.3.3 多边形区域变量	4-2
4.3.4 工具坐标系变量	4-3
4.3.5 用户坐标系变量	4-4
4.3.6 外部工具坐标系变量	4-5
4.3.7 其它操作	4-6
4.4 IO 检测	4-9
4.5 位置管理	4-15
4.6 日志管理	4-17
4.6.1 报警	4-17
4.6.2 日志	4-18
4.6.3 帮助	4-22
4.7 系统设置	4-23
4.7.1 用户管理	4-23
4.7.2 通用设置	4-26
4.7.3 高级设置	4-30
第 5 章变量介绍.....	5-1
5.1 基本数据类型.....	5-1
5.2 位置数据类型.....	5-3
5.3 系统数据类型.....	5-9
5.4 IO 数据类型	5-16
5.5 摆动数据类型.....	5-17
5.6 时钟数据类型.....	5-18
5.7 区域数据类型.....	5-19
5.8 PLC 数据类型	5-20
5.9 码垛数据类型.....	5-21
5.10 SOCKET 数据类型	5-22
第 6 章指令介绍.....	6-1
6.1 运动指令	6-1
6.2 控制指令	6-22

6.3 等待指令	6-28
6.4 IO 指令	6-29
6.5 设置指令	6-40
6.6 位置运算指令.....	6-51
6.7 传送带跟随指令.....	6-54
6.8 中断指令	6-60
6.9 位运算指令	6-65
6.10 时钟指令	6-68
6.11 区域指令.....	6-68
6.12 视觉指令	6-71
6.13 码垛指令	6-74
6.14 阵列指令	6-79
6.15 Socket 指令.....	6-81
6.16 ModbusTCP 指令	6-86
6.17 软浮动指令	6-1
6.18 负载在线辨识指令.....	6-2
6.19 字符串指令	6-5
6.20 数学运算函数.....	6-10
6.21 其他函数	6-13
第 7 章高级功能.....	7-14
7.1 多类型区域监控.....	7-14
7.2 碰撞检测	7-20
7.2.1 碰撞检测	7-20
7.2.2 碰撞检测自调整	7-21
7.3 碰撞回退	7-22
7.4 振动抑制	7-23
7.5 传送带跟随	7-25
7.6 系统 IP 设置.....	7-31
7.7 示教编程器升级与备份.....	7-34
7.7.1 示教编程器程序升级	7-34
7.7.2 示教编程器程序备份	7-37
7.8 控制器运行程序更新.....	7-38
7.8.1 控制器程序更新	7-38
7.8.2 控制器程序还原	7-39
7.8.3 备份导出	7-40
7.8.4 工程导入	7-42
7.9 一点到达功能介绍.....	7-43
7.10 工具坐标系标定.....	7-44
7.11 用户坐标系/外部工具坐标系标定.....	7-48
7.12 外部 TCP 功能	7-51
7.13 一点对正坐标轴.....	7-53
7.14 伺服管理	7-55
7.15 二维视觉的使用.....	7-58
7.16 三维视觉的使用.....	7-60
7.17 ModbusTCP	7-63

7.18 锁机与解除	7-67
7.19 软浮动功能	7-71
7.20 九点标定功能.....	7-75
7.21 多工位预约	7-78
7.22 变位机功能	7-82
7.23 倒序执行	7-86
7.24 伺服参数自调节功能.....	7-87
7.25 跨腕部奇异功能.....	7-88
7.26 回零功能	7-88
7.26.1 功能概述	7-88
7.26.2 整体回零	7-89
7.26.3 单轴回零	7-89
7.27 保养提示	7-92
7.28 Scara 快速标定.....	7-95
7.29 去限位块丢零保护功能.....	7-104
7.30 一点回原位功能.....	7-105
7.31 全局程序	7-106
7.32 从站配置	7-1
7.33 中断功能	7-2
7.34 自动掉励磁	7-5
7.35 负载在线识别.....	7-7
第 8 章 FAQ.....	8-10
8.1 如何根据铭牌信息查找对应控制柜说明书.....	8-10
解决方法	8-10
8.2 埃斯顿机器人的供电需求.....	8-10
解决方法	8-11
8.3 如何处理机器人开机后显示急停处于按下状态.....	8-11
解决方法:	8-11
8.4 机器人如何进行周期性保养.....	8-12
解决方法	8-12
8.5 如何备份机器人示教器程序.....	8-13
解决方法	8-13
8.6 触发机器人急停按钮后如何恢复.....	8-13
解决方法	8-13
8.7 使用时发现示教器不能进行程序编辑和修改怎么办.....	8-14
解决方法	8-14
8.8 埃斯顿机器人目前支持的通讯有哪些.....	8-15
8.9 如何解决正常使用中机器人运行的点位发生了偏移.....	8-15
解决方法	8-15
8.10 如何判定机器人开机后无法运动.....	8-15
解决方法	8-15
8.11 如何解决机器人开机后示教器无法进入系统.....	8-16
解决方法	8-16
第 9 章附录.....	9-17
9.1 示教编程器权限分配表.....	9-17

9.2 ModbusTCP 控制接口数据表 9-24

9.3 报警信息一览表..... 9-27

前言

概述

本文档适用于控制系统 RCS2 V1.30，描述了 ER 系列机器人的构成及其示教控制器的操作，有助于使用者了解并能掌握 ER 系列机器人的操作和使用。

本书常用词汇

“ESTUN ROBOTICS” 是埃斯顿工业机器人的商品名。ESTUN ROBOTICS 由机器人本体“机器人”、机器人控制柜“控制柜”、“示教编程器”和“供电电缆”构成。在本书中，这些部分如下表表示。

设备	本书表示法
ER 系列机器人电气控制柜	控制柜
ER 系列示教编程器	示教编程器/示教器
机器人与控制柜间的电缆	供电电缆

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
 警告	以本标志开始的文本表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 注意	以本标志开始的文本表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	以本标志开始的文本能帮助您解决某个问题或节省您的时间。
 说明	以本标志开始的文本是正文的附加信息，是对正文的强调和补充。

手册索引

常用手册通常包含：操作手册、控制柜使用说明书、本体使用说明书，相关资料下载请参见：[埃斯顿官网资料下载](#)。

手册类型	下载地址	手册说明
操作手册	常用手册名称： 《ER 系列工业机器人操作手册》 《ER 系列 RCS2 指令系统手册》 您可以登录“ 埃斯顿官网资料下载 ”，搜索相关手册名称，下载您需要的资料。	操作手册主要介绍示教器功能及如何通过示教器对机器人示教再现操作
控制柜说明书	您可以在“资料下载-控制柜”中下载控制柜的资料。	- 控制柜说明书主要介绍控制柜的结构功能及连线方法 - 不同系列控制柜结构不同，请查阅控制柜配套的说明书。 - 可参见 8.1 如何根据铭牌信息查找对应控制柜说明书，查找控制柜对应的说明书。
本体使用说明书	您可以在“资料下载-本体使用说明书”中下载本体说明书的资料。	本体说明书主要介绍本体即“机器人”的详细信息和安装方法

安全注意事项

机器人与其他机械设备的要求通常不同，如它的大运动范围、快速的操作、手臂的快速运动等，这些都会造成安全隐患。阅读和理解使用说明书及相关的文件，并遵循各种规程，以免造成人身伤害或设备事故。用户有责任保证其安全的操作环境符合和遵守地方及国家有关安全性的法令、法规及条例。



注意

在使用本说明书之前，请务必熟读本章节和“第 1 章安全”中的所有安全注意事项。

- 本说明书对 ER 系列机器人的示教、再现、程序及文件编辑操作进行了全面的说明，请务必下认真阅读并充分理解的基础上操作机器人
- 说明书中的图解，有的为了说明细节取下盖子或安全罩进行绘制，运转此类部件时，务必按规定将盖子或安全罩还原后，再按说明书要求运转。
- 说明书中的图及照片为代表性示例，可能与所购买产品不同。
- 说明书有时由于产品改进、规格变更及说明书自身更便于使用等原因而进行适当的修改，将没另行通知，若需最新版本资料，请关注我公司网站或与我公司服务部联系。
- 未经我公司同意，不得随意增加或删除部分或全部内容，不允许将该手册部分或全部内容用于第三方的设计。
- 客户擅自进行产品改造，不在本公司保修范围之内，由此引起的直接或间接损失，本公司概不负责。



危险

- 急停按钮使用

操作机器人前，按下机器人控制柜前门及示教编程器上的急停键，并确认伺服电源被切断。伺服紧急情况下，若不能及时制动机器人，则可能引发人身伤害或设备损坏事故。

解除急停后再接通伺服电源时，要解除造成急停的事故后再接通伺服电源。由于误操作造成的机器人动作，可能引发人身伤害事故。



急停键



解除急停状态

- 在机器人动作范围内示教时，请遵守以下事项：
 - 保持从正面观看机器人。
 - 遵守操作步骤。
 - 考虑机器人突然向自己所处方位运动时的应变方案。
 - 确保设置躲避场所，以防万一。由于误操作造成的机器人动作，可能引发人身伤害事故。

- 进行以下作业时，请确认机器人的动作范围内没人，并且操作者处于安全位置操作：
 - ⇒ 机器人控制柜接通电源时。
 - ⇒ 用示教编程器操作机器人时。
 - ⇒ 试运行。
 - ⇒ 自动再现时。
- 不慎进入机器人动作范围内或与机器人发生接触，都有可能引发人身伤害事故。另外，发生异常时，立即按下急停键。急停键位于电柜及示教编程器的右侧。
-



警告

- 进行机器人示教作业前要检查以下事项，有异常则应及时修理或采取其他必要措施。
 - ⇒ 机器人动作有无异常。
 - ⇒ 外部电线遮盖物及外包装有无破损。
 - 示教编程器用完后须放回原处。
如不慎将示教编程器放在机器人、夹具或地上，当机器人运动时，示教编程器可能与机器人或夹具发生碰撞，从而引发人身伤害或设备损坏事故。
 - 在理解 ER 系列机器人使用说明书的“警告标志”的基础上，使用机器人。
-

第 1 章 安全

机器人与其他机械设备的要求通常不同,如它的大运动范围、快速的操作、手臂的快速运动等,这些都会造成安全隐患。阅读和理解使用说明书及相关的文件,并遵循各种规程,以免造成人身伤害或设备事故。用户有责任保证其安全的操作环境符合和遵守地方及国家有关安全性的法令、法规及条例。

1.1 保障安全

1. 机器人的示教维护必须遵照下列法规

有关工业安全和健康的法律。
有关工业安全和健康法律的强制性命令。
有关工业安全和健康法律的相应条例。

2. 其他有关法律

美国的职业安全与健康法。
德国的工厂法。
英国的工作安全与健康法。
欧盟的 89/392 机械行业指令和欧共体的 91/368。

3. 安全操作规范:

遵守工业机器人的安全操作 (ISO 10218)。机器人的安全操作

4. 其他安全事项

增补安全管理系统。指定授权的操作者及安全管理人员,并给予进一步的安全教育。

示教和维修机器人的工作被列入工业安全和健康法律中的“危险操作”。(仅限中国)

操作人员需参加由南京埃斯顿机器人有限公司提供的专门培训。

1.2 专门培训

示教和维护机器人的人员必须事先经过培训。

关于培训的更多信息请咨询南京埃斯顿机器人工程有限公司。

1.3 机器人使用说明书清单

请确认您拥有如下所有说明书。如缺少上述任何说明书,请与南京埃斯顿机器人工程有限公司联系。

ER 系列机器人使用说明书

ER 系列机器人电气维护手册

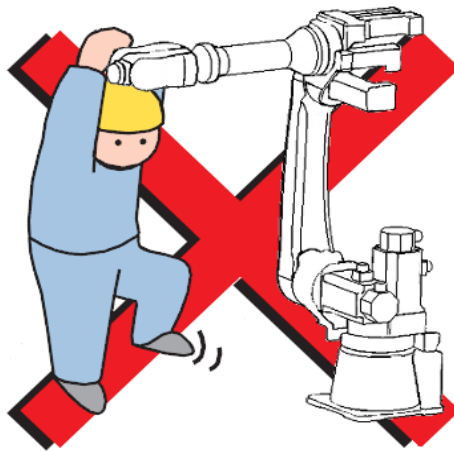
1.4 操作人员安全注意事项

整个机器人的最大动作范围内均具有潜在的危險性。为机器人工作的所有人员（安全管理人员、安装人员、操作人员和维修人员）必须时刻树立安全第一的思想，以确保所有人员的安全。

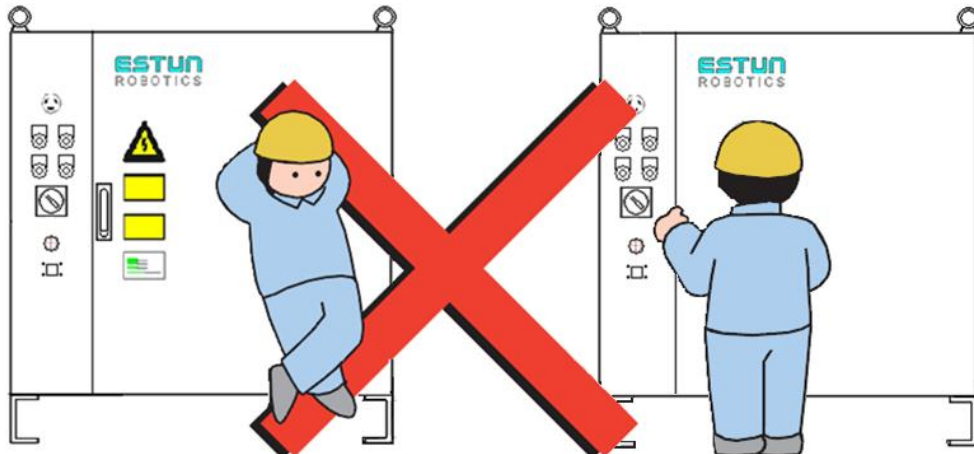


警告

- 机器人的安装区域内禁止进行任何的危險作业。
如任意触动机器人及其外围设备，将会有造成伤害的危險。
- 请采取严格的安全预防措施，在工厂的相关区域内应安放，如“易燃”、“高压”、“止步”或“闲人免进”等相应警示牌。
忽视这些警示可能会引起火警、电击或由于任意触动机器人和其他设备会造成伤害。
- 严格遵守下列条款：
 - ⇒ 穿着工作服（不穿宽松的衣服）。
 - ⇒ 操作机器人时不许戴手套。
 - ⇒ 内衣裤、衬衫和领带不要从工作服内露出。
 - ⇒ 不佩戴大的首饰，如耳环、戒指或垂饰等。
 - ⇒ 必要时穿戴相应的安全防护用品，如安全帽、安全鞋（带防滑底的）、面罩、防护镜和手套。
 - ⇒ 不合适的衣服可能会造成人身伤害。
- 未经许可的人员不得接近机器人和其外围的辅助设备。
不遵守此提示可能会由于触动控制柜、工件、定位装置等而造成伤害。
- 绝不要强制地扳动机器人的轴。
否则可能会造成人身伤害和设备损坏。



- 绝不要倚靠在控制柜上；不要随意地按动操作键。
否则可能会造成机器人产生未预料的动作，从而引起人身伤害和设备损坏。



1.5 机器人的安全注意事项

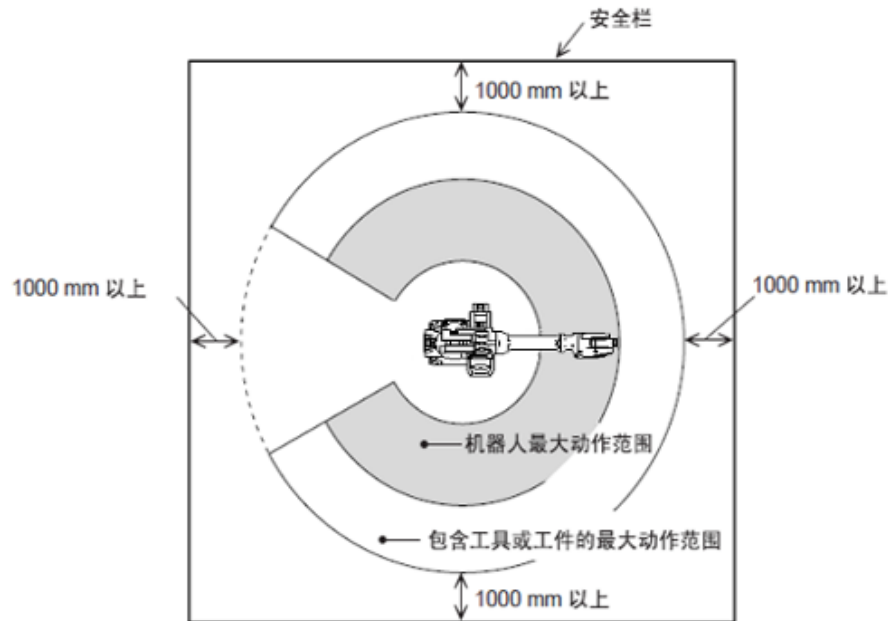
1.5.1 安装及配线安全

安装和配线时,请从 ER 系列机器人使用说明书中查阅详细资料, 在安装准备及安装过程中, 要遵守下列事项, 确保安装过程的安全。



危险

- 选择一个区域安装机器人, 并确认此区域足够大, 以确保装有工具的机器人转动时不会碰着墙、安全围栏或控制柜。
否则可能会因机器人产生未预料的动作而引起人身伤害或设备损坏。



- 接地工程要遵守电气设备标准及内线规章制度。否则会有触电、火灾的危险。



注意

- 确认有足够的空间来维修机器人、控制柜和其它外围设备。否则可能会在维修期间造成伤害事故。
- 机器人是由控制柜或夹具用控制柜进行控制的。
为了确保安全，一定要在能看得见机器人的位置进行操作。
未经授权的人员操作可能会造成人身伤害或设备损坏。
- 控制柜应安装在机器人动作范围的安全围栏之外。否则可能会由于接触机器人而造成人身伤害和设备损坏。
- 针对各种机器人，应按说明书中规定的螺栓大小及类型来安装机器人。否则可能会造成人身伤害和设备损坏。
- 为控制柜配线前须熟悉配线图，配线须按配线图进行。错误的配线或零部件的不正确移位，可能产生设备损坏或人身伤害。
- 在进行控制柜与机器人、外围设备间的配线及配管时须采取防护措施，如将管、线或电缆从坑内穿过或加保护盖予以遮盖，以免被人踩坏或被叉车碾压而坏。
- 操作者和其他人员可能会被明线、电缆或管路绊住而将其损坏，从而会造成机器人的非正常动作，以致引起人身伤害或设备损坏。



1.5.2 作业区安全

在作业区内工作时粗心大意会造成严重的事故，因此强令执行下列防范措施：



危险

在机器人周围设置安全围栏，以防造成与已通电的机器人发生意外的接触。在安全围栏的入口处要张贴一个“远离作业区”的警示牌。安全围栏的门必须加装可靠的安全联锁装置。
忽视此警示会由于接触机器人而可能造成严重的事故。



警告

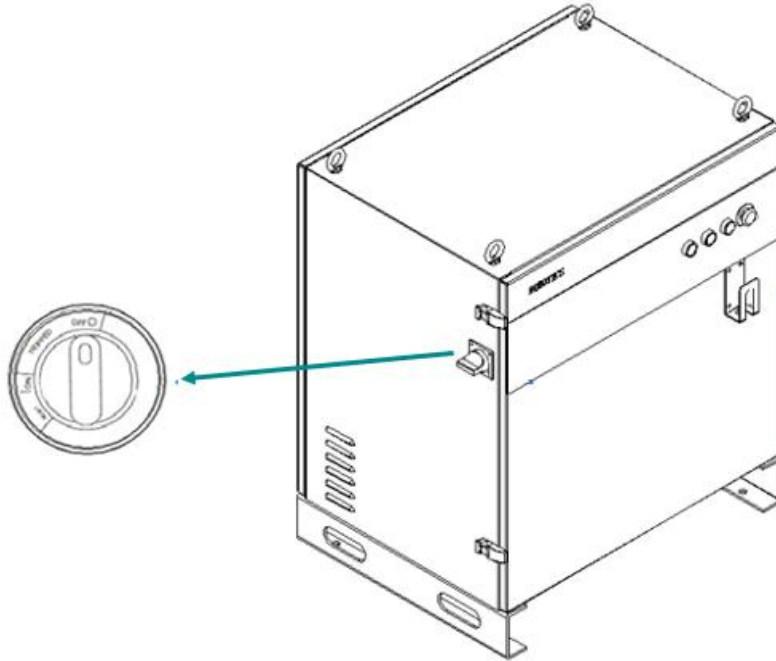
备用工具及类似的器材应放在安全围栏外的合适地区内。
工具和散乱的器材不要遗留在机器人、控制柜或系统（如焊接夹具）等周围，如果机器人撞击到作业区中这些遗留物品，即会发生人身伤害或设备事故。

1.5.3 操作安全



危险

- 当往机器人上安装一个工具时，务必先切断（OFF）控制柜及所装工具上的电源并锁住其电源开关，而且要挂一个警示牌。



安装过程中如接通电源，可能会因此造成电击，或会产生机器人的非正常运动，从而引起伤害。

- 绝不要超过机器人的允许范围（机器人的允许范围请参见说明书中的技术规范部分）。否则可能会造成人身伤害和设备损坏。
- 无论何时如有可能的话，应在作业区外进行示教工作。
- 当在机器人动作范围内进行示教工作时，则应遵守下列警示：
 - 始终从机器人的前方进行观察。
 - 始终按预先制定好的操作程序进行操作。
 - 始终具有一个当机器人万一发生未预料的动作而进行躲避的想法。
 - 确保您自己在紧急的情况下有退路。
 否则可能误操作机器人，造成伤害事故。
- 在操作机器人前，应先按控制柜前门及示教编程器右上方的急停键，以检查“伺服准备”的指示灯是否熄灭，并确认其电源确已关闭。
如果紧急情况下不能使机器人停止，则会造成机械的损害。
- 在执行下列操作前，应确认机器人动作范围内无任何人：
 - 接通控制柜的电源时。
 - 用示教编程器移动机器人时。
 - 试运行。
 - 再现操作时。
 如果人员进入机器人动作范围，可能会因与机器人接触而引起伤害。
如发生问题，则应立即按动急停键。
急停键位于控制柜前门及示教编程器的右上方。



注意

- 示教机器人前先执行下列检查步骤，如发现问题则应立即更正，并确认所有其他必须做的工作均已完成。
 - 检查机器人运动方面的问题。
 - 检查外部电缆的绝缘及护罩是否损害。
- 示教编程器使用完毕后，务必将示教编程器挂回到控制柜的钩子上。
如示教编程器遗留在机器人上、系统夹具上或地面上，则机器人或装载其上的工具将会碰撞它，因此可能引起人身伤害或设备损坏。
- 机器人的操作或检查人员必须接受有关法规及公司策略的培训。

1.6 移动及转让机器人的注意事项

移动及转让机器人时，应遵照下列安全防范事项：

- 移动或转让机器人时，应附带机器人的有关说明书（参见 1.3 机器人使用说明书清单）以便所有用户有权使用这些必须的说明书。
- 如缺少任何说明书，请与埃斯顿机器人工程有限公司联系
- 如果机器人及控制柜上的警示牌模糊不清，请清理此警示牌，以便能被正确辨认。另请注意某些地方法规的规定，如安全警示牌不在适当的位置上，可能会被禁止该设备的使用。
- 移动或转让机器人时，建议请埃斯顿机器人工程有限公司派员进行检查。
错误的安装及配线会造成人身伤害和设备事故。
- 绝不要对机器人或控制柜做任何改动。不遵守此警示会引起火警、电力故障或操作错误，以致造成设备损坏及人身伤亡。

1.7 废弃机器人的注意事项

废弃机器人必须组照国家及地方的法律和相关规定

废弃前即使是临时保管，也将机器人固定牢靠防止倾倒，避免因机器摔倒造成伤害。

第 2 章 了解产品

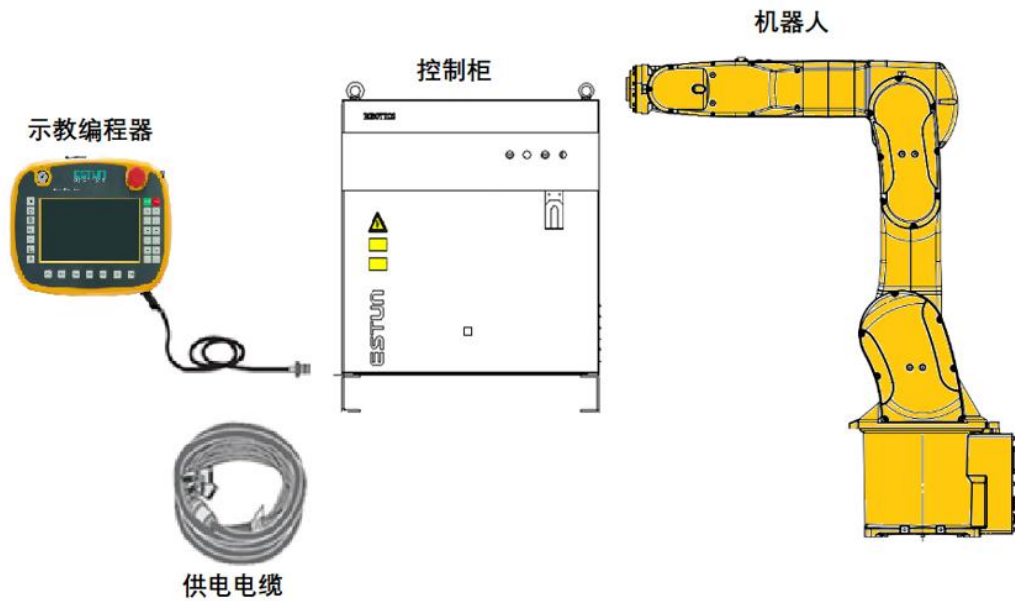
2.1 产品确认

2.1.1 装箱内容确认

产品到达后，请清点发货清单，标准的发货清单中包括下列 4 项内容：（选项货物内容的信息将单独提供）。

- 机器人
- 控制柜
- 示教编程器
- 供电电缆（机器人与控制柜间的电缆）

图2-1 装箱清单示意图



2.1.2 订货号确认

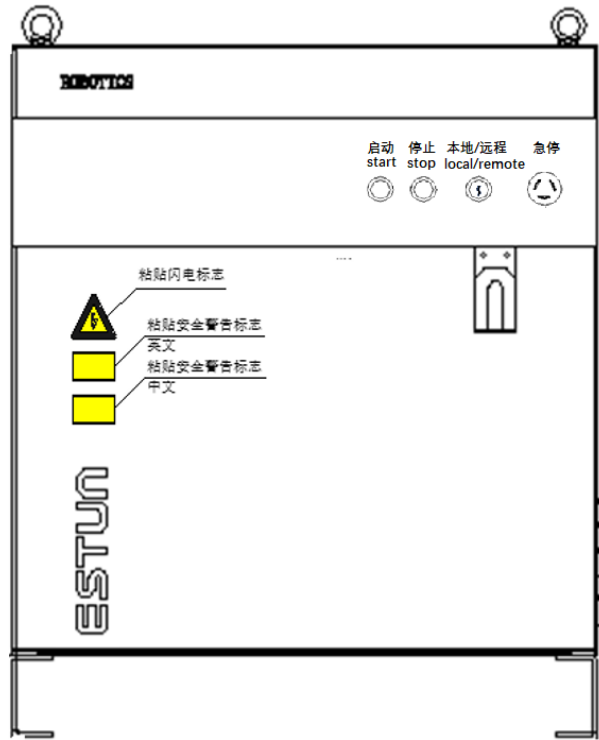
确认机器人与控制柜上的订货号是否一致。

2.2 控制柜

本章节以标准型控制柜为例，只针对控制柜外观和按键做简单介绍，不同控制柜的外观按钮和内部结构存在差异，详细介绍请参考具体型号配套的控制柜使用说明书。

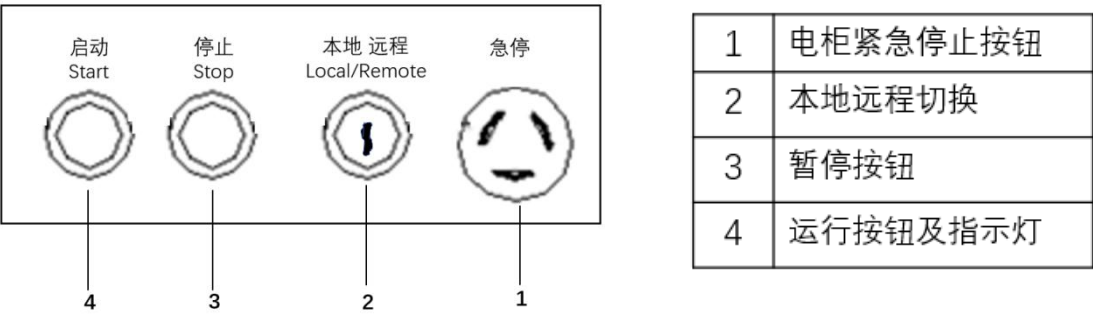
控制柜的外观和电柜外部按钮分布如图 2-2 所示。

图2-2 控制器外观



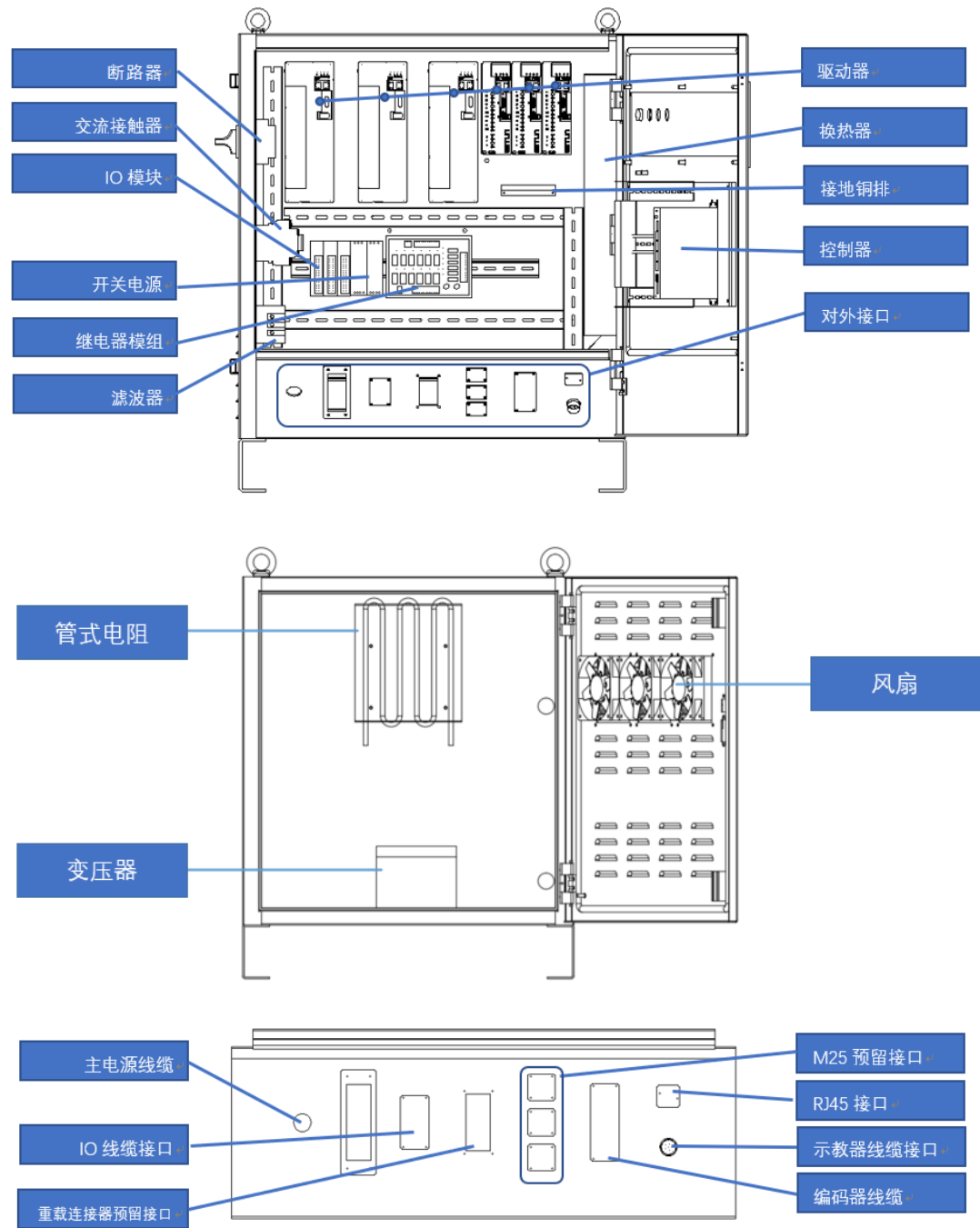
控制柜面板按键分布和说明如图 2-3 所示。

图2-3 控制柜按钮分布



电柜内部布局如所示（不同机器人控制柜布局有所不同）：

图2-4 电柜内部布局图



2.3 示教编程器

2.3.1 外观

示教编程器主要由对机器人进行示教、编程所需的操作键和按钮组成，示教编程器外观如下图所示。

图2-5 示教编程器外观图



表2-1 示教编程器外观结构组件说明

序号	说明
1	显示屏
2	紧急停止按钮
3	模式选择开关
4a, 4b, 4c	全局功能按键
5	状态指示灯
6	伺服使能开关
7	悬挂手柄
8	电缆接入区
9	USB 插槽
10	触摸笔

2.3.2 触摸笔

触摸笔是示教器附带的，放在示教器背后，使用时可拔出触摸笔。在使用触摸笔点击示教器显示屏时，不能使用螺丝刀或其他尖锐物品。

2.3.3 USB 插槽

将 USB 存储器连接到 USB 端口以读取或保存文件。

2.3.4 按键

功能键

功能键在示教编程器的下方，表 2-2 列出各按键的说明。

表2-2 功能键的说明

按键	功能说明
	快捷菜单键
	伺服使能功能键
	坐标系选择功能键 在点动模式下实现机器人坐标系的切换，包括关节坐标系、世界坐标系、用户坐标系和工具坐标系
	倒序执行程序功能键 手动模式下，按此键切换至倒序执行，再次按下，切换回至正常顺序执行。
	示教再现运行方式选择功能键 示教再现运行时，运行方式的选择，包括单步和连续两种方式
	速度减功能键。远程模式下，按下或点击 V-使全局速度减少。
	速度加功能键。远程模式下，按下或点击 V+使全局速度增加。

菜单键

菜单键在示教编程器的左侧，表 2-3 列出各按键的说明。

表2-3 菜单键说明

按键	功能说明
	系统设置 此菜单界面实现用户登录，版本信息等功能
	工程管理 此菜单界面实现对工程的管理功能，如导入导出、新建、删除、重命名等
	程序编辑 此菜单界面实现对程序的编辑功能，如示教指令插入、修改等

按键	功能说明
	程序数据 此菜单界面实现变量的新建、赋值、管理等功能
	I/O 检测 此菜单界面实现系统 I/O 信号的监视，并可进行 I/O 信号的诊断。
	位置管理 此菜单界面实现对示教操作参数的设置功能，如点动速度、坐标系切换等
	系统日志 此菜单界面显示所有系统操作反馈信息，包括提示、警告、错误等

轴操作键

轴操作键在示教编程器的右侧，表 2-4 列出了各按键的说明。

表2-4 轴操作键说明

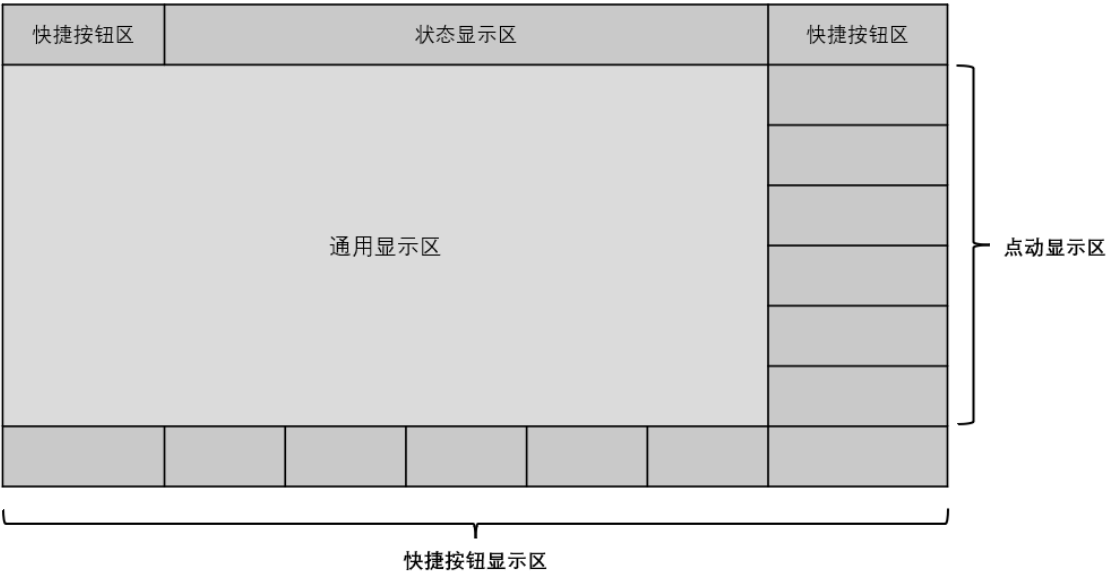
按键	功能说明
	开始按钮：启动示教再现程序
	停止按钮：停止示教再现程序
	正方向点动按键 - 关节坐标系下，对应关节的正方向运动 - 基坐标系下，对应 X、Y、Z、A、B、C 的正方向运动 - 工具坐标系下，对应 TX、TY、TZ、TA、TB、TC 的正方向运动（坐标系相关介绍参见第 3.2.1 章机器人坐标系）
	负方向点动按键 - 关节坐标系下，对应关节的负方向运动 - 基坐标系下，对应 X、Y、Z、A、B、C 的负方向运动 - 工具坐标系下，对应 TX、TY、TZ、TA、TB、TC 的负方向运动。

2.3.5 显示

显示区域说明

示教编程器的显示屏是 7 英寸的彩色显示屏，能够显示数字、字母和符号。显示屏分为 4 个显示区，分别为通用显示区、状态显示区、快捷按钮显示区和点动显示区。示教编程器的显示区界面如图 2-6 所示。

图2-6 示教编程器显示区域说明



通用显示区

在通用显示区，显示各种操作菜单界面，可对程序、特性文件、各种设定进行显示和编辑。

状态显示区

在状态区内显示系统的各运行状态，包括运行模式、伺服状态、点动坐标系和点动速度等，如图 2-7 所示。

图2-7 示教编程器状态显示区

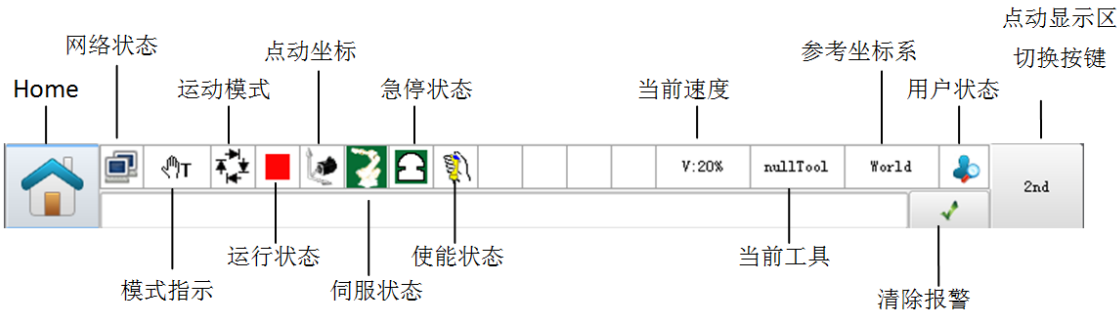
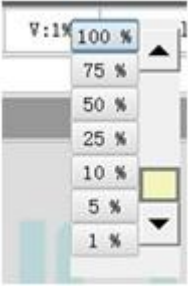


表2-5 状态区显示内容说明

状态	说明
模式指示	示教编程器运行模式 ⌚ ^A ：再现模式 🖐 ^T ：示教模式 ⊗：禁止操作模式 ⌚ ^{AE} ：远程模式

状态	说明
点动坐标系	 : 关节坐标系  : 直角坐标系  : 工具坐标系  : 用户坐标系  : 外部工具坐标系  : 变位机坐标系
伺服状态	机器人伺服状态  : 伺服使能  : 伺服未使能  : 伺服使能，但存在零点丢失  : 伺服未使能，但存在零点丢失  : 伺服自动掉励磁
运行模式	再现程序的运动模式  : 单步运行模式  : 连续运行模式  : 倒序运行模式
运行状态	再现程序运行状态  : 运行  : 暂停  : 停止
急停状态	 : 急停按钮未按下  : 急停按钮已按下
使能状态	显示示教编程器上的使能开关当前状态:  : 使能开关未接通  : 使能开关已接通

状态	说明
当前速度	显示当前全局速度的百分比。最小全局速度为 1%。点击可选择全局速度。 
用户状态	显示当前用户权限:  : 操作人员  : 编程人员  : 管理人员  : 厂家人员

快捷按钮显示区

显示当前页面下，各个快捷按钮的作用。

点动显示区

点动显示区内显示当前对应点动按键的点动方式，如当点动显示区显示 A1、A2、A3、A4、A5、A6 时，对应为关节点动方式。

2.3.6 模式开关介绍

拨动钥匙就可以在三种模式之间切换。系统运行模式由三种：

- 1. 扭动实际示教器上方钥匙，将机器人由自动模式切换为手动模式，速度自动调整为 10%。
- 2. 在远程模式下，按下或点击示教器 V+使全局速度增加，按下或点击 V-使全局速度减少。
- 3. 全局速度最小支持 1%。

手动模式（示教模式）

在示教模式下可以进行：

- 编制、示教程序
- 修改已登录程序
- 程序文件和参数的设定。

自动模式（再现模式）

在再现模式下可以进行：

- 示教程序的再现
- 程序文件的设定、修改或删除。

远程模式

系统随时可调出在存储器中的程序，进行程序的编辑和执行。

作为编辑对象的程序叫做“编辑程序”，是示教模式下显示程序内容时所显示的程序。

作为执行对象的程序叫做“执行程序”，是再现模式下显示程序内容时所显示的程序。

图2-8 运行模式与程序的对应关系



在示教编程器上对动作模式进行切换，编辑程序和执行程序之间不发生转换，在再现模式下运行一个编辑程序，首先把模式转换成再现模式，然后读出程序。

2.3.7 急停按钮介绍

紧急按钮按下后会被锁定，伺服电源被切断，机器人不能再进行操作。发生紧急情况时，可以通过快速按下此按钮来得到保护的措施。

位置：控制柜前面板或示教器上的急停键。

急停：按下控制柜前面板紧急按钮或示教器急停键。

解除急停：按照钮上的指示方向旋转按钮，如下图所示

图2-9 解除急停状态



2.3.8 伺服使能开关介绍

再示教和点运动时，需要按下伺服使能开关才能使伺服使能（励磁）。作用是控制接通和断开使能信号。

开关位置：示教器右把手的位置

使能信号接通：开关按下的第一个接触点。

使能信号断开：开关按到底或松开开关按钮。

2.3.9 示教编程器内部的接线端子信号定义

表2-6 I/O 端子排引脚信号定义

I/O 端子排引脚号	信号定义
1	ENABLE_ED1+(使能开关信号 1 输入)
2	ENABLE_ED1-(使能开关信号 1 输出)

I/O 端子排引脚号	信号定义
3	ENABLE_ED2+(使能开关信号 2 输入)
4	ENABLE_ED2-(使能开关信号 2 输出)
5	NC
6	+24VDC
7	GND
8	E-STOP_ES1+
9	E-STOP_ES1-
10	E-STOP_ES2+
11	E-STOP_ES2-

表2-7 RJ45 引脚信号定义

I/O 端子排引脚号	信号定义
1	TD+
2	TD-
3	RD+
4	—
5	—
6	RD-
7	—
8	—

2.3.10 示教编程器线缆

航插接口信号定义

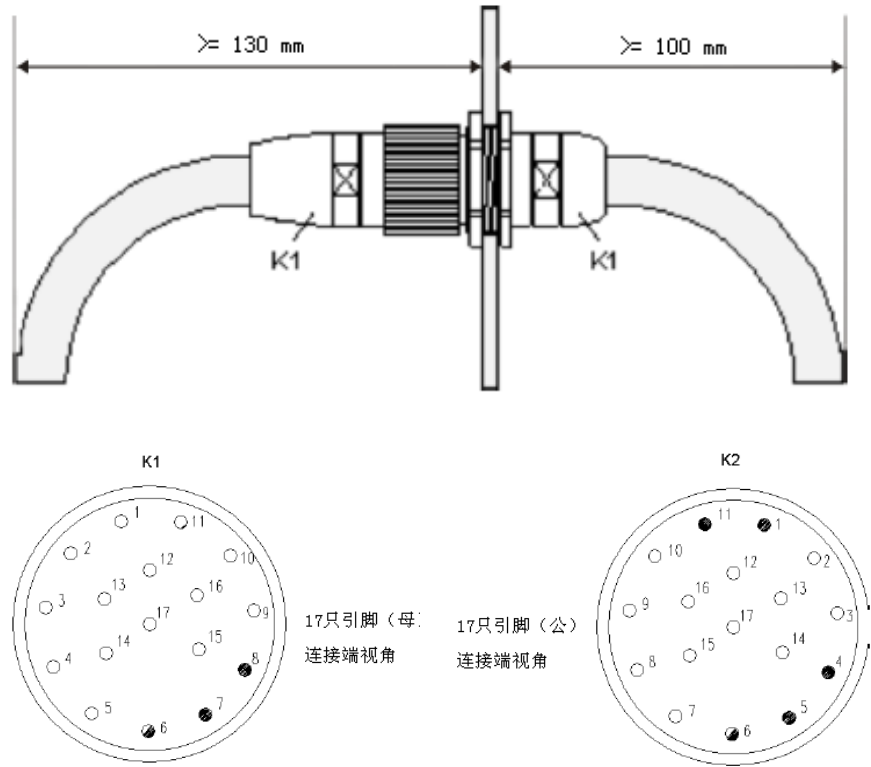
表2-8 航插接口信号定义

信号序号	线的颜色	信号定义
1	粉红	24VDC
2	黑	GND
3	棕绿	E-STOP_ES1+
4	白绿	E-STOP_ES1-
5	灰粉	E-STOP_ES2+
6	红蓝	E-STOP_ES2-
7	棕	ENABLE_ED1+ (使能开关信号 1 输入)
8	黄	ENABLE_ED1- (使能开关信号 1 输出)

信号序号	线的颜色	信号定义
9	—	—
10	—	—
11	紫	—
12	绿	ENABLE_ED2+（使能开关信号 2 输入）
13	蓝	TD+
14	白	TD-
15	橙	RD+
16	红	RD-
17	灰	ENABLE_ED2-（使能开关信号 2 输出）

线缆与控制柜的连接

图2-10 示教编程器线缆与控制柜的连接



K1, K2 的各引脚信号定义同航插接口信号定义中的信号定义说明。

第 3 章 产品使用

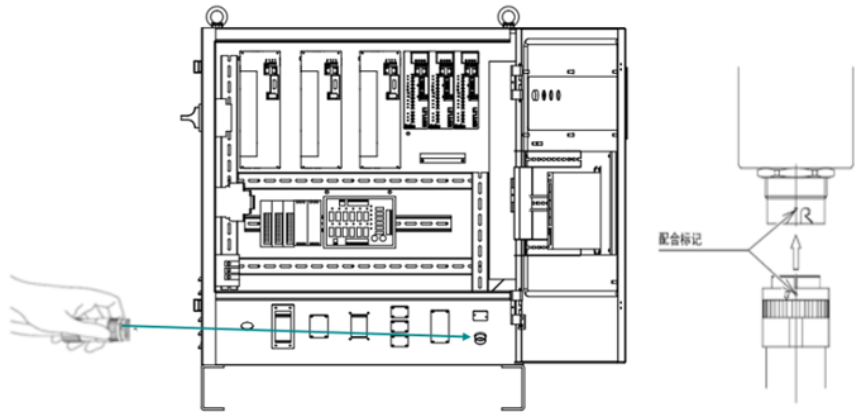
在产品使用前需先完成产品的安装和连接。

3.1 连接方法

产品使用之前，请确认机器人、控制柜和示教器的连接正常，设备是可用状态，连接方法请参考《埃斯顿机器人 XX 系列本体使用说明书》

请参考以下步骤完成机器人、控制柜和示教编程器的连接。

- 步骤 1 拆去包装，取出供电电缆，将电缆连接到控制柜后面的插座上。
- 步骤 2 将机器人与控制柜连接。
- 步骤 3 确认电缆接头的形状与尺寸以及机器人上的插座位置，将电缆插入相应的插座上，并固定牢靠。
- 步骤 4 将示教编程器的电缆连接到控制柜示教器接口



----结束

3.2 操作说明

3.2.1 机器人的坐标系

机器人坐标系主要分为以下几种：

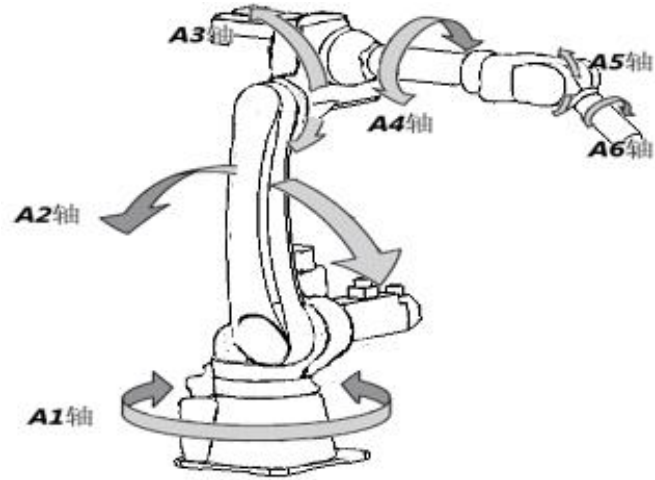
坐标系名称	定义	作用
关节坐标系	机器人各轴进行单独动作。	机器人的 A1、A2、A3、A4、A5、A6 各轴分别运动。

直角坐标系	机器人末端点均可沿直角坐标系下设定的 X 轴、Y 轴、Z 轴平行移动和旋转。	机器人控制点沿 X、Y、Z 轴平行移动和旋转。
工具坐标系	机器人腕部法兰盘所持工具的有效方向作为 Z 轴，并把坐标定义在工具的尖端点。	机器人控制点沿设定在工具尖端点的 X，Y，Z 轴做平行移动。

关节坐标系

设定关节坐标系时，机器人的 A1、A2、A3、A4、A5、A6 各轴分别运动，关节坐标系下各轴运动方向如图 3-1 所示。

图3-1 关节坐标系下各轴的运动方向



按轴操作键时各轴的动作，如表 3-1 所示。

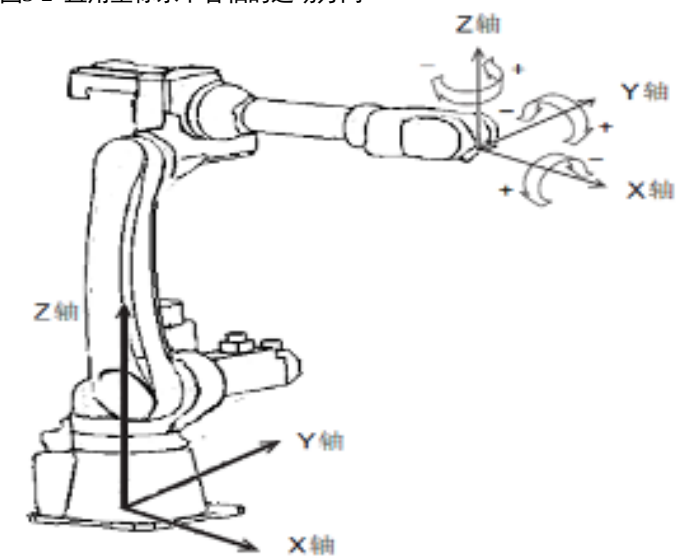
表3-1 按轴操作键的各轴动作

轴名称	动作
A1	本体左右旋转
A2	大臂前后旋转
A3	小臂上下旋转
A4	手腕左右旋转
A5	手腕上下旋转
A6	法兰左右回旋

直角坐标系

设定为直角坐标系时，机器人控制点沿 X、Y、Z 轴平行移动，直角坐标系下各轴运动方向如图 3-2 所示。

图3-2 直角坐标系下各轴的运动方向



按住轴操作键时，各轴的动作如表 3-2 所示。

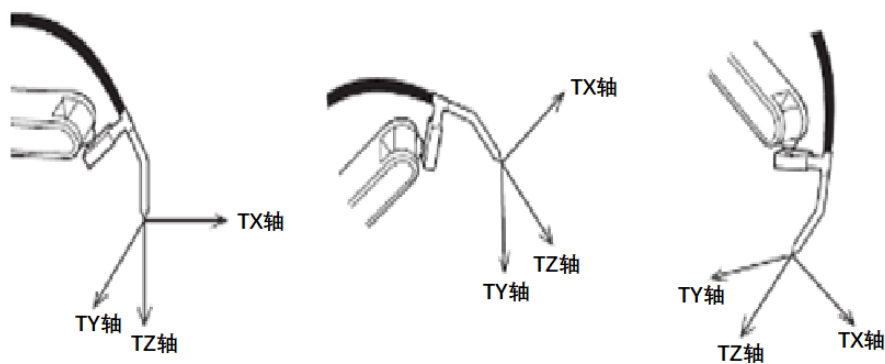
表3-2 直角坐标系下轴操作键对应各轴的运动方向

轴名称	动作
X	沿直角坐标系 X 方向运动
Y	沿直角坐标系 Y 方向运动
Z	沿直角坐标系 Z 方向运动
A	绕直角坐标系 X 方向旋转
B	绕直角坐标系 Y 方向旋转
C	绕直角坐标系 Z 方向旋转

工具坐标系

工具坐标系把机器人腕部法兰盘所持工具的有效方向作为 Z 轴，并把坐标定义在工具的尖端点，设定为工具坐标系时，机器人控制点沿设定在工具尖端点的 X，Y，Z 轴做平行移动。

图3-3 工具坐标系方向定义



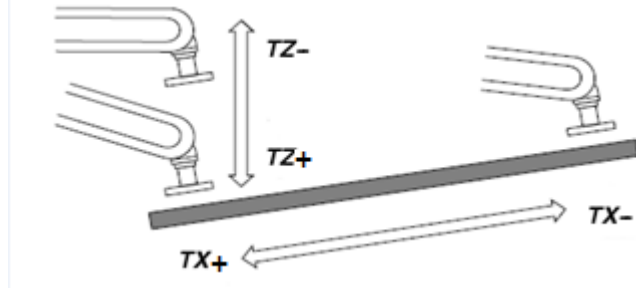
按住轴操作键时，各轴的动作如表 3-3 所示。

表3-3 工具坐标系下轴操作键对应的运动说明

轴名称	动作
TX	沿工具坐标系 X 方向运动
TY	沿工具坐标系 Y 方向运动
TZ	沿工具坐标系 Z 方向运动
TA	绕工具坐标系 X 方向旋转
TB	绕工具坐标系 Y 方向旋转
TC	绕工具坐标系 Z 方向旋转

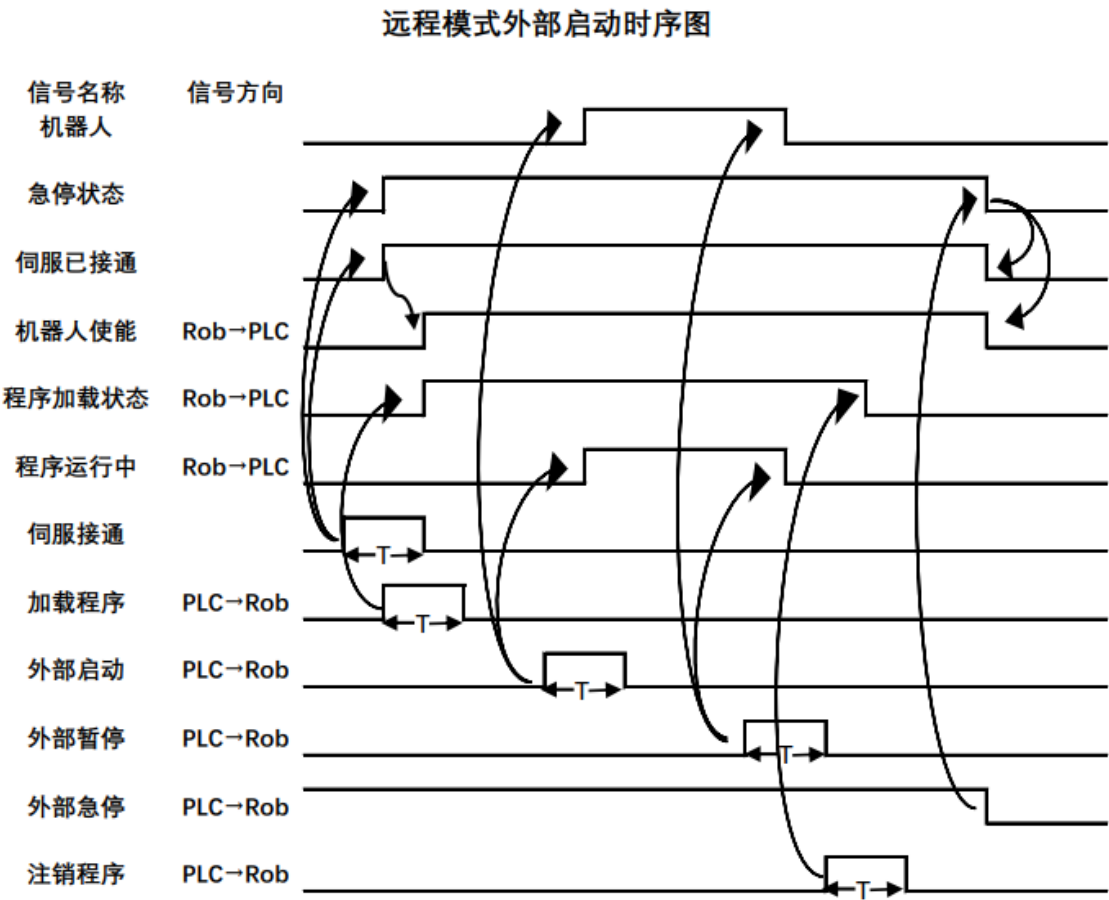
工具坐标的移动，以工具的有效方向为基准，与机器人的位置、姿势无关，所以进行相对于工件不改变工具姿势的平行移动操作时最为适宜。

图3-4 工具坐标系下的平行移动操作



3.3 启停时序

图3-5 机器人控制器启停时序图



3.3.2 示教

注意事项

1. 示教编程器模式旋钮旋至手动模式，若有误操作时，由于人的本能反应，会紧握安全开关，而安全开关是两级开关，当按到第二级时，伺服电源被切断。

急停的确认

操作机器人之前，先分别按下电柜和示教编程器上的急停键，确认伺服电源是否被切断。

表3-4 急停按钮功能确认步骤

序号	项目	说明
1	急停键	按电柜或示教编程器上的急停键。
2	确认伺服电源被切断	当伺服电源接通时，示教编程器上的伺服通电图标是亮的。按下急停键后，伺服电源被切断，伺服通电图标将熄灭。
3	按	确认正常后，按  ，使得伺服电源处于励磁的状态，示教编程器上是 Mot 灯常亮。

示教前的准备

开始示教前，请做以下准备。

- 检查控制柜和机器本体是否有异常，如控制柜电源线、伺服端口和控制器端口是否正确连接，检查机械本体上是否有异物。
- 启动控制柜，并确认各部分处于正常工作状态。（详情请参见 2.2 控制柜）
- 上伺服主电，并测试控制柜和示教编程器上的急停按键是否有效。
- 把示教编程器上的模式开关打到示教模式。
- 此时按下手压开关，即可通过“轴操作键”进行相应的轴操作（轴操作键对应的轴运动参见 3.2.1 机器人的坐标系）。

示教的基本步骤

新建一个示教工程

在示教操作时，可以通过指令的方式记录示教的机器人轨迹，再现时，即可通过记录的指令来再现示教动作，新建一个示教工程的步骤如下。

步骤 1 按  或在主界面上点击“工程管理”进入工程管理界面。

步骤 2 点击“新建”，弹出“新建程序”对话框，如 3-6 所示。

图3-6 新建工程对话框



新建程序对话框的截图。对话框标题为“新建程序”。内部包含三个输入区域：第一个是“请选择工程：”，右侧有一个下拉菜单，当前显示“新建工程”；第二个是“请输入工程名：”，下方是一个空文本输入框；第三个是“请输入程序名：”，下方是一个包含“main”文本的输入框。对话框底部右侧有两个按钮：“确认”和“取消”。

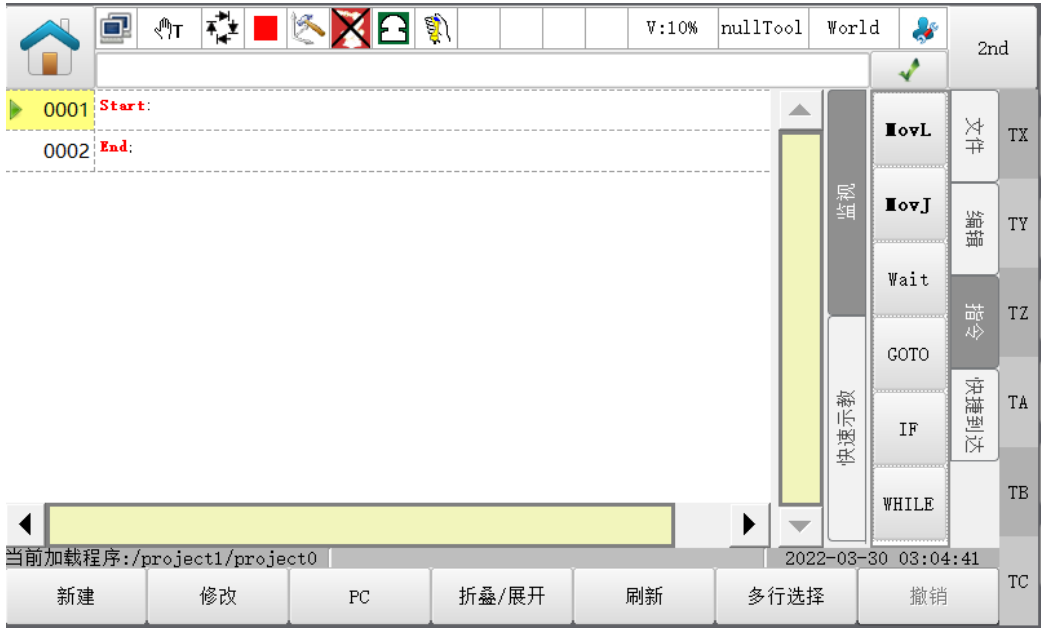
步骤 3 在“请选择工程”右侧的下拉列表中选择“新建工程”。

步骤 4 点击“请输入工程名”的输入框，并在弹出的软键盘中输入需要新建的工程名，如 project0。

步骤 5 点击“请输入程序名”的输入框，并在弹出的软键盘中输入需要新建的程序名，如 program0。

步骤 6 点击“确认”，新建工程或程序成功，界面直接跳至程序编辑界面，如 3-7 所示。

图3-7 程序编辑界面



步骤 7 在程序界面中进行指令编辑，控制机器人运动的指令就是运动指令。在移动指令中，记录有移动的位置、插补方式、再现速度等(其他指令类型详见附录基本指令类型)。

```
MovJ (P1, V50, "RELATIVE", C0) //点到点的运动方式
MovL (P2, V50, "RELATIVE", C0) //直线的运动方式
```

----结束

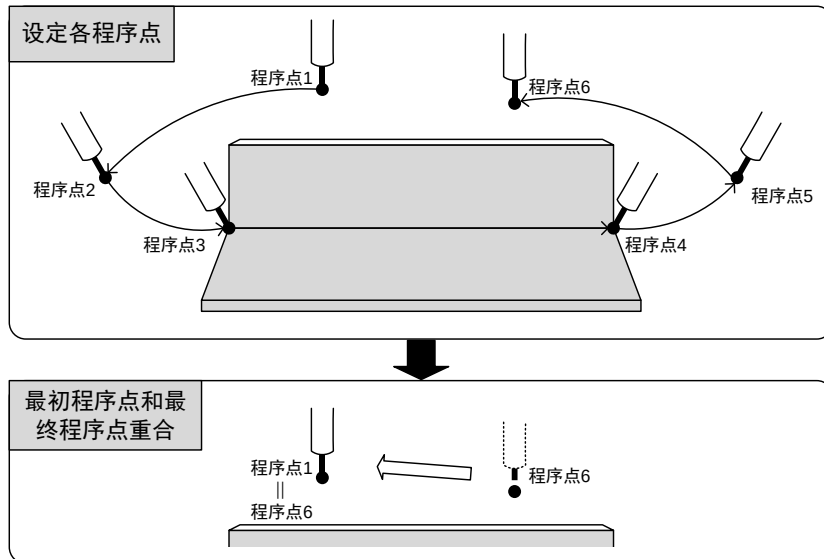
示教概述



执行示教操作时，请确认操作员与机器人之间已经保持了安全距离。

程序是把机器人的作业内容用机器人语言加以描述。现为机器人输入以下从工件 A 点到 B 点的加工程序，此程序有 1 至 6 的 6 个程序点组成。

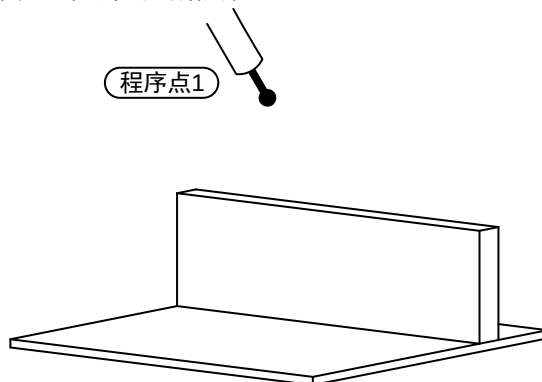
图3-8 示教程序的运行路径示意



程序点 1：开始位置

把机器人移动到完全离开周边物体的位置，输入程序点 1。

图3-9 示教程序开始点示意



步骤 1 按住手压开关，用“轴操作键”移动机器人移动至开始位置，开始位置请设置在安全并适合作业准备的位置。

步骤 2 在程序界面内按“新建”键，并在指令列表中选择“运动指令→MovJ”

步骤 3 单击“确定”，进入 MovJ 的指令编辑界面，如图 3-10 所示。

图3-10 MovJ 的指令编辑界面

指令: MovJ 行号: 3
工程名: project0 程序名: program0

目标位置	A L: P1
目标速度	S: V50
过渡类型	"RELATIVE"
过渡值	S: C0
工具参数	DEFAULT
坐标系参数	DEFAULT
工件负载	DEFAULT

示教

确认

取消

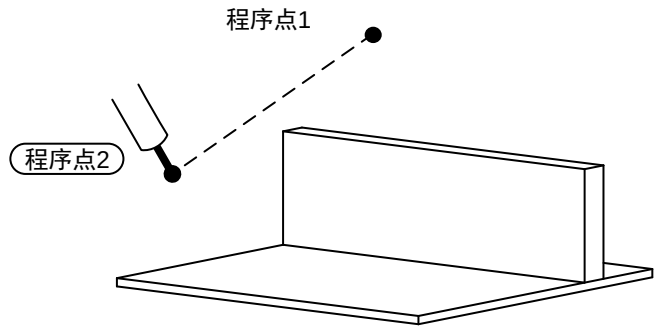
步骤 4 点击“示教”后，P1 点记录此时程序点 1 处的机器人坐姿，点击“插入”后，完成程序点 1 的示教工作，在程序界面内显示的指令如下：

```
MovJ (P1, V50, "RELATIVE", C0)
```

----结束

程序点 2：作业开始位置附近

图3-11 示教程序点 2



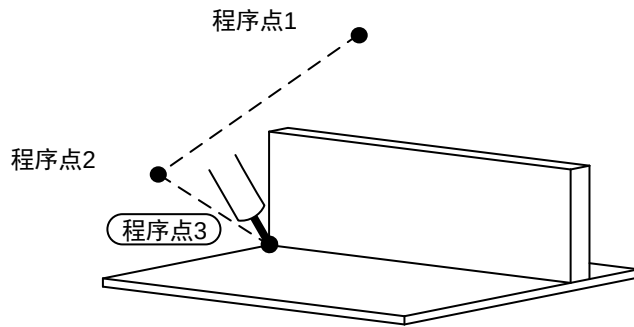
用“轴操作”键，使机器人姿态成为作业姿态。用示教程序点 1 的方法来示教程序点 2，示教完成后程序指令如下：

```
MovJ (P2, V50, "RELATIVE", C0)
```

程序点 3：作业开始的位置

保持程序点 2 的姿态不变，移向作业开始位置，即程序点 3 处。

图3-12 示教程序点 3



步骤 1 保持程序点 2 的姿态不变，按 ，设定机器人坐标系为直角坐标系，用“轴操作键”把机器人移到作业开始位置。

步骤 2 在程序界面内按“新建”，进入指令界面选择直线插补指令。

步骤 3 点击“示教”，完成对程序点 3 的示教工作，程序指令如下。

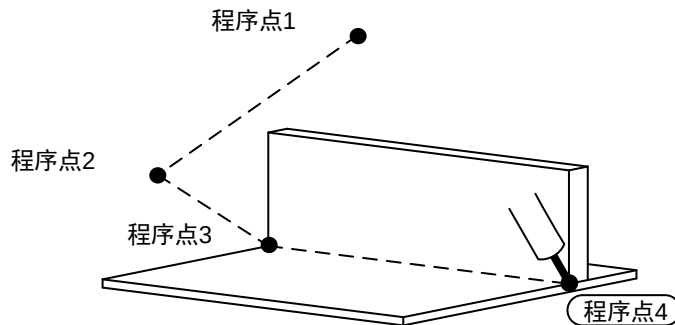
```
MovL (P3, V50, "RELATIVE", C0)
```

----结束

程序点 4：作业结束位置

指定作业结束位置。

图3-13 示教程序点 4



用“轴操作键”把机器人移动到作业结束位置。从作业开始位置到结束位置，不必非常精确，为了不碰撞工件，移动轨迹可远离工件。

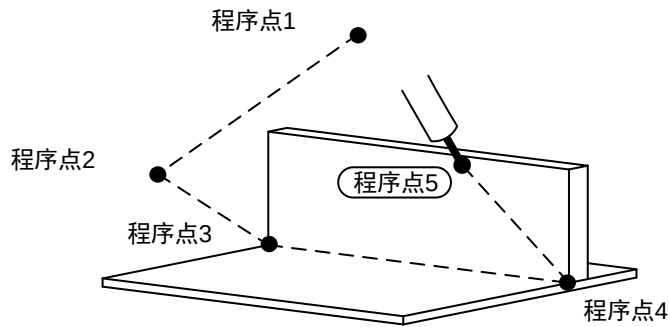
用示教程序点 1 相同的方法，添加一条直线插补指令，其中 P4 为机器人在程序点 4 处的坐姿，程序指令如下。

```
MovL (P4, V50, "RELATIVE", C0);
```

程序点 5：不碰触工件、夹具的位置

把机器人移动到不碰触工件和夹具的位置。

图3-14 示教程序点 5

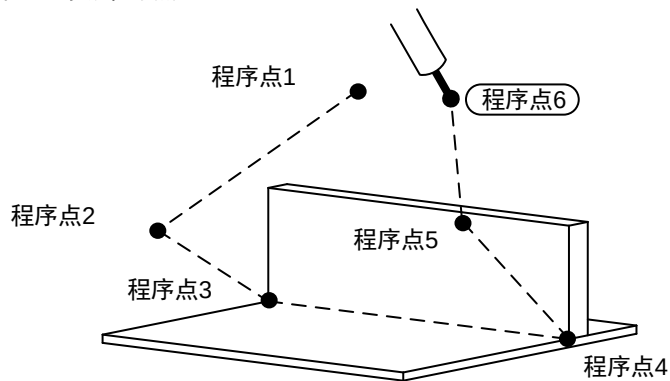


用“轴操作键”把机器人移动到不碰触夹具的位置。用示教程程序点 1 相同的方法，添加一条直线插补指令，其中 P5 为机器人在程序点 5 处的坐姿，程序指令如下：

```
MovL(P5, V50, "RELATIVE", C0);
```

程序点 6：开始位置附近请把机器人移动到开始位置附近

图3-15 示教程程序点 6



在关节坐标下用“轴操作键”把机器人移动到开始位置附近。用示教程程序点 1 相同的方法，添加一条关节插补指令，其中 P6 为机器人在程序点 6 处的坐姿，程序指令如下：

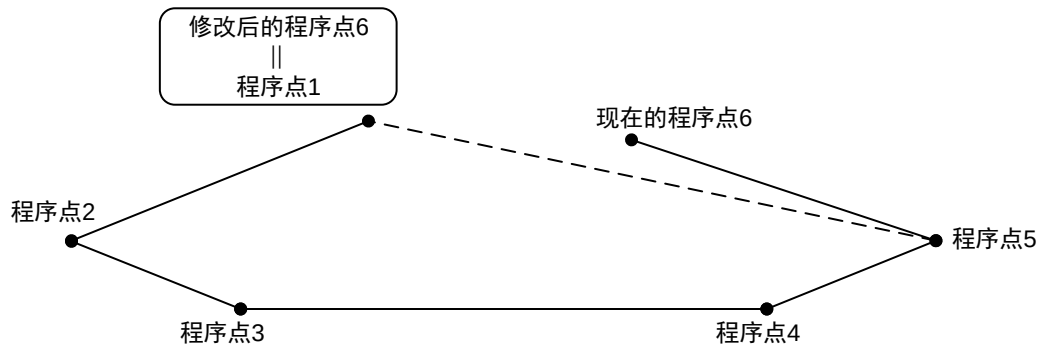
```
MovJ(P6, V50, "RELATIVE", C0);
```

最初的程序点和最后的程序点重合

现在，机器人停在程序点 1 附近的程序点 6 处。如果能从作业结束位置的程序点 5 直接移动到程序点 1 的位置，就可以立刻开始下一个工件的作业，从而提高工作效率。

下面，试着把最终的程序点 6 与最初位置的程序点 1 设在同一个位置。

图3-16 修改程序点 6 到程序点 1



步骤 2 把光标移动到程序点 1 处，点击“设置指针按钮”，按“Step”，选择单步模式，把速度设定 10%以内后，按“Start”，程序开始运行，机器人移动到程序点 1。

步骤 3 把光标移动到程序点 6 处，按“修改”后，进入修改界面。

步骤 4 按“示教”键，程序点 6 的位置被修改到与程序点 1 相同的位置。

----结束

轨迹的确认

在完成了机器人动作示教后，运行一下示教程序，以便检查一下各程序点是否有不妥之处，在轨迹确认时，一定要保证示教编程器在示教模式，且在运动时按下手压开关。

步骤 1 把光标移动到程序点 1，按“PC”。

步骤 2 按 ，选择单步模式。

步骤 3 按速度增减键将速度设定在 10%以内。

步骤 4 按下手压开关，机器人伺服使能。

步骤 5 按 ，程序开始单步运行，每按一次 ，机器人运行一条指令程序。

步骤 6 程序确认完成后，把光标移到程序起始处。

步骤 7 最后试一试所有程序点的连续动作。按 ，选择连续模式。按 后，机器人连续再现所有指令程序，一个循环后停止运行。



说明

示教模式下，运行轨迹时，轨迹速度受限于国标速度的限制，线速度最高不超过 250mm/s，姿态速度最高不超过 180deg/s。

----结束

程序的修改

如果机器人实际运动轨迹和预期的有差别，可以通过修改程序中的程序点位置和速度来调整机器人的运动轨迹。

修改前

确认了在各程序点机器人的动作后，如果需要进行位置修改、程序点插入或删除时，请按以下步骤对程序进行编辑。

步骤 1 按  或在主界面上点击“工程管理”进入工程管理界面。

步骤 2 选择需要编辑的程序。

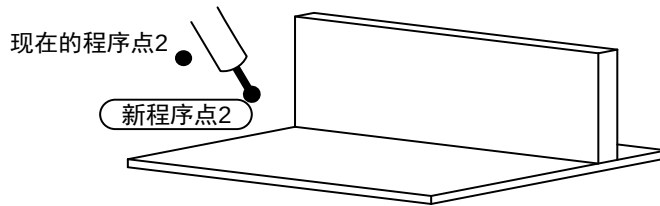
步骤 3 点击“加载”，进入程序编辑界面。

----结束

修改程序点的位置数据

如下图所示，修改程序点 2 的位置。

图3-17 修改程序点 2 的位置



步骤 2 按上面的方法，设为单步模式，速度设为 10%以内，按  将机器人移至待修改的程序点 2 处。每按一次 ，机器人移动一个程序点。

步骤 3 用“轴操作键”移动机器人至新的位置，如图 3-17 所示的新程序点 2 处。

步骤 4 点击“修改”，进入点修改界面。

步骤 5 按“示教”，完成对程序点的位置数据的修改。

----结束

插入程序点

在两个程序点之间插入新的程序点，以在程序点 5、6 之间插入新的程序点为例，步骤如下：

步骤 1 按 ，把机器人移动到程序点 6，在移动之前需确认机器人从当前位置可以移动到位置点 6，不会发生碰撞。

步骤 2 用“轴操作键”把机器人移至欲插入的位置。

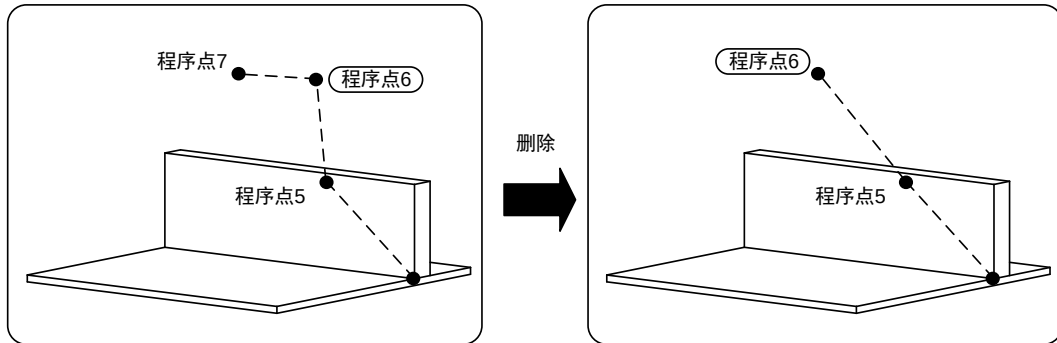
步骤 3 参见添加程序点 1 的方法，添加一条关节插补指令。

----结束

删除程序点

这次试着删除刚刚插入的程序点。从下面的左图状态，返回到原来的右图状态。

图3-18 删除程序点



进入程序编辑界面，点击需要删除的指令，然后点击右侧“删除”，并在弹出的“删除指令”提示框中点击“确定”。

3.3.3 再现

再现前的准备

注意事项：

操作前检查工作区域内无任何操作人员或其他工作人员。

检查示教器和控制柜上的急停按钮处于无效状态。

执行示教完的程序称为再现。在再现之前，请确保如下操作：

- 步骤 1 再操作之前确定操作人员及其他工作人员处在机器人工作区之外。
- 步骤 2 确保控制柜和示教编程器上的急停按钮处于无效状态。
- 步骤 3 加载需要再现的工程文件。
- 步骤 4 选择需要首先执行的程序，然后点击“PC”，确保程序从第一行开始执行。
- 步骤 5 按速度增减键将速度设置较为理想的速度。

----结束

再现步骤

步骤 6 把示教编程器上的模式旋钮设定在自动模式，把光标移到程序开头。

步骤 7 上主电，按 ，接通伺服电源。

步骤 8 按 。机器人将执行再现的程序一个循环后停止。

----结束

停止与再启动

使动作中的机器人停止或让机器人自动停止，有以下原因：

- 暂停操作或急停操作
- 报警引起的停止

- 其他停止
- 由于各项作业引起的停止

启停规范

1. 请不要频繁开关电源，否则会损坏驱动。每次开启和停止电源时间间隔需满足 60s。
2. 断开电源前，请确保机器处于停止状态，应先停止设备程序，再断开电源。

暂停操作

- 用示教编程器执行暂停
 - 暂停
按示教编程器上的 ，机器人将暂停动作。
 - 解除
按示教编程器上的 ，机器人从之前暂停的位置继续开始动作。
- 用外部输入信号（专用）执行暂停
 - 暂停
按外部控制装置上的停止信号后，机器人即暂停动作。
 - 解除
按外部控制装置上的启动信号后，机器人从之前暂停的位置继续开始动作。



注意

当程序处暂停状态下，进行如下操作程序将会自动转到停止状态：

- 程序上下文改变，如工程加载/注销、指令新增、删除等
 - 设置 PC
 - 切换系统模式
 - End 结束
 - 点动操作
 - 回零操作
 - 其它一些系统设置的操作
-

急停操作

急停操作可在以下各处执行。

- 示教编程器上急停按钮
- 控制柜前面板急停旋钮
- 外部输入信号（专用）

【急停】按示教编程器或控制柜的急停键后，机器人主电被切断，机器人立刻停止。

【解除】先按箭头方向旋转急停键，再按 ，给伺服上电。

急停后的再启动



警告

急停后的再启动，按  操作来确认位置，确认无工件、夹具的干涉。
连续程序在高速再现过程中被急停后，机器人有时会在所显示的程序点前 2 至 3 个程序点附近停止。从此处再启动可能会有工件或夹具的干涉。

报警引起的停止

动作过程中发生报警后，机器人会立刻停止动作。示教编程器上显示报警画面，用户可以进入系统日志界面，查看报警的详细信息。

报警种类	报警解除方法
轻故障报警	在示教编程器报警画面上，选择“清除报警”，解除报警状态。
重故障报警	切断主电源，排除报警因素。

其它暂停

- 切换模式引起的暂停
再现过程中，从再现模式切换到示教模式时机器人立即停止。
启动方法：请回到再现模式并执行启动操作。
- 执行 PAUSE 命令引起的暂停
执行 PAUSE 命令，使机器人暂停。
启动方法：需执行启动操作，机器人从下一个命令处开始继续动作。

修改再现速度

用户可通过速度增减键修改执行再现速度，修改再现速度方式如下：

- 再现中可即时修改。
- 可对速度执行几次试验性试修改后，在确认了动作的基础上执行修改。

第 4 章 系统功能介绍

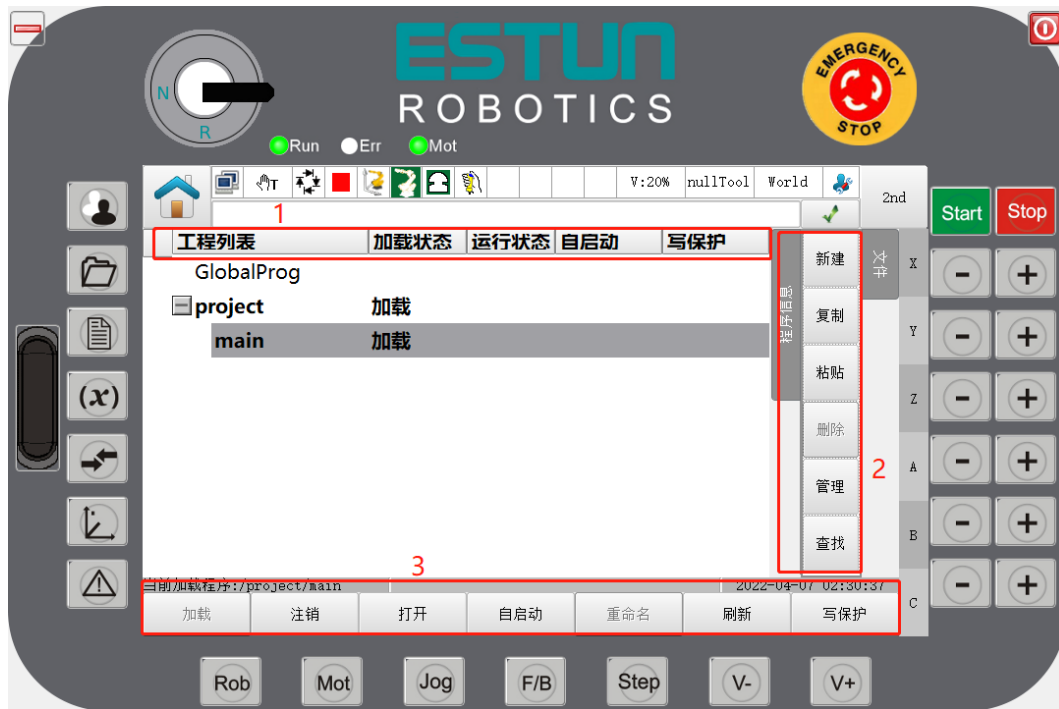
机器人控制器与用户的交互主要通过示教编程器进行的。示教编程器是机器人控制系统的重要组成部分，操作者通过示教编程器进行手动示教，控制机器人达到不同位姿，并记录各个位姿点坐标利用机器人语言进行在线编程，实现程序回放，让机器人执行程序要求的轨迹运动。用户可以通过示教编程器的按键或触摸板来与控制系统交互的目的。

我司使用的示教编程器提供的功能组件包括：

功能名称	功能说明	快捷按键
系统设置	用户管理：此菜单界面实现用户登录，版本信息等功能	
	通用设置：	-
	高级设置：	-
工程管理	实现对工程的管理功能，如导入导出、新建、删除、重命名等	
程序编辑	实现对程序的编辑功能，如示教指令插入、修改等	
程序数据	实现变量的新建、赋值、管理等功能	
I/O 检测	实现系统 I/O 信号的监视，并可进行 I/O 信号的诊断。	
位置管理	对示教操作参数的设置功能，如点动速度、坐标系切换等	
系统日志	显示所有系统操作反馈信息，包括提示、警告、错误等	

4.1 工程管理

按或在主界面上点击“工程管理”进入工程管理界面，主要由状态显示栏、底部功能键和右侧功能键组成，如下图



1. 状态显示栏：

工程列表：显示工程名称，点击 可展开该工程目录下的所有程序文件。

加载状态：显示工程或程序是否“加载”，默认显示空白，加载后显示“加载”。

运行状态：显示工程或程序是否运行中，运行状态有“运行中”、“暂停”、“停止”。

自启动：显示程序是否设置成“自启动”。默认显示空白，开启后显示“自启动”。

写保护：显示程序是否设置成“写保护”。默认显示空白，开启后显示“只读”。

2. 底部功能键

加载：载入已有工程的工程或程序，可以直接加载程序。

注销：配合加载使用，未加载文件时按钮呈灰色不可用状态。

打开：只能打开程序文件，打开前需先加载工程；打开不等同加载，运行时提示先导入程序。

自启动：远程模式下自动加载设置的程序文件；程序与自启动程序不符合时会有弹窗提示。

重命名：重新命名工程或程序名，不区分字母大小写。

刷新：刷新显示当前界面。

写保护：只读程序，无法修改。

- 开启：选中需要的程序，点击“写保护”按钮，根据界面提示开启“写保护”后，用户无法修改该程序中的内容，程序中涉及的全局变量、工程变量和程序变量都不可更改。
- 取消：再次选中该程序，点击“取消写保护”按钮，根据界面提示取消写保护后，用户可对该程序进行修改。
- 若此按钮置灰，表示当前用户没有操作权限。仅支持“管理人员”和“厂家人员”操作此功能。

3. 右侧功能键

新建：用来新建工程或者新建已有工程下的程序文件；

复制：复制已有的工程或程序文件，点击粘贴后提示重新命名复制的工程或程序；

- 粘贴：配合复制功能使用；
- 删除：删除不需要的工程或程序；
- 管理：工程导入导出；
- 查找：查找程序；
- 程序信息：可查看选中程序的状态信息，也可以添加备注信息。

4.1.1 新建工程或程序

按照如下指导步骤新建一个工程或程序。

- 步骤 1 按或在主界面上点击“工程管理”进入工程管理界面，如图 4-1 所示。
在工程管理界面中，点击下方的“刷新”，可更新工程列表。



- 步骤 2 选择右侧“文件→新建”，弹出“新建程序”对话框，如图 4-2 所示。

 **窍门**
在其它界面中选择右侧“文件→新建”可直接进行新建工程的操作。



- 步骤 3 若要新建一个工程，在“请选择工程”右侧的下拉列表中选择“新建工程”。
若要在某个工程中新建一个程序，则在其下拉列表中选择所需要的工程名，并跳至**步骤 5**。
或者事先在工程列表中选择想要的工程，然后点击“新建”。
若工程列表中没有程序，默认且仅有的选项即是“新建工程”。

步骤 4 点击“请输入工程名”的输入框，并在弹出的软键盘中输入需要新建的工程名，如：project0。

步骤 5 点击“请输入程序名”的输入框，并在弹出的软键盘中输入需要新建的程序名，如 program0。

步骤 6 点击“确认”，新建工程或程序成功，界面直接跳至程序编辑界面，如图 4-3 所示。

图4-3 程序编辑界面



----结束

4.1.2 删除工程或程序

按照如下指导步骤删除一个工程或程序。

步骤 1 在工程管理界面中选中要删除的工程或程序，如“project0”。

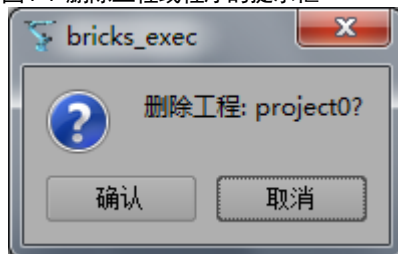


说明

1. “加载”中的工程或程序无法进行删除操作。需先点击下方的“注销”，才能执行删除操作。
2. 若要删除程序，该程序所在的工程状态不能是“加载”，必须先“注销”工程，才能执行程序删除操作。

步骤 2 点击右侧“删除”，将弹出提示框，如图 4-4 所示。

图4-4 删除工程或程序的提示框



步骤 3 点击“确认”，即可删除工程。

----结束

4.1.3 重命名工程或程序

按照如下指导步骤重命名一个工程或程序。



说明

1. “加载”中的工程或程序不能进行“重命名”操作，需先点击下方“注销”，才能执行重命名操作
2. 若需要对程序进行重命名操作，该程序所在的工程状态不能是“加载”，必须先“注销”工程，才能执行程序重命名操作。

步骤 1 在工程管理界面中点击要重命名的工程或程序，如“project0”。

步骤 2 点击下方的“重命名”。



步骤 3 单击“请输入新的工程名称”，并在弹出的软键盘中输入需要新的工程名。



步骤 4 单击“确定”以完成重命名操作。

----结束

4.1.4 复制/粘贴工程或程序

按照如下指导步骤复制并粘贴一个工程或程序。

步骤 1 在工程管理界面中点击要复制的工程或程序，如 project0。

步骤 2 点击“粘贴”。



说明

同一个程序不能被复制粘贴到不同的工程目录下。

对于复制的目标是一个程序时，如果粘贴的路径是另一个工程下，则会弹出提示信息“不能将程序粘贴到不同的工程下！”请点击并选择正确的工程，再继续执行操作。

步骤 3 单击“请输入新的工程名称”，并在弹出的软键盘中输入新的工程名。

步骤 4 单击“确定”以完成复制/粘贴操作。

----结束

4.1.5 加载/注销工程或程序

在工程列表中，“加载状态”一栏中已经显示了工程或程序当前是否正在被加载。

最多只有一个工程或程序被加载，当另一个工程或程序被加载时，之前被加载的工程或程序会自动注销。

若要加载或注销一个工程或程序，只需要点击要进行该操作的工程或程序，然后单击“加载”或“注销”。

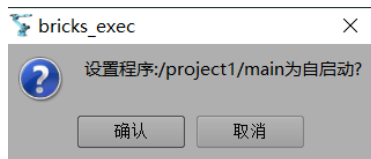
4.1.6 自启动程序

自启动功能只能对程序进行设置，不能对工程设置。若程序设置“自启动”，当用户启用远程模式时，该程序将被自动加载。

步骤 1 在工程列表中，选中需要自启动的程序。

步骤 2 点击下方的“自启动”按钮。

步骤 3 点击弹出对话框中的“确认”，该程序的自启动操作即可完成。



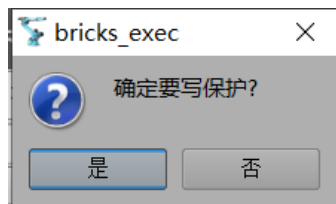
4.1.7 写保护

写保护功能只能对程序进行设置，不能对工程设置。若程序设置“写保护”，用户则无法对程序工程中的程序文件和变量文件进行任何更改。该功能是可以手动取消的，这就充分保证了程序文件的安全性。

步骤 1 在工程列表中选中需要写保护的程序。

步骤 2 点击下方的“写保护”按钮。

步骤 3 点击弹出对话框中的“是”，该程序的写保护操作即可完成。



步骤 4 点击下方“取消写保护”后，根据弹框内容操作，即可去除该程序的写保护功能。

4.1.8 管理（工程导入导出）

在执行工程导入导出操作前，用户需准备一个可读写的 U 盘，然后进行如下指导步骤。

- 工程导入，是指将 U 盘中的工程或程序文件导入至控制器中。

- 工程导出，是指将控制器中的工程或程序文件导出至 U 盘中。

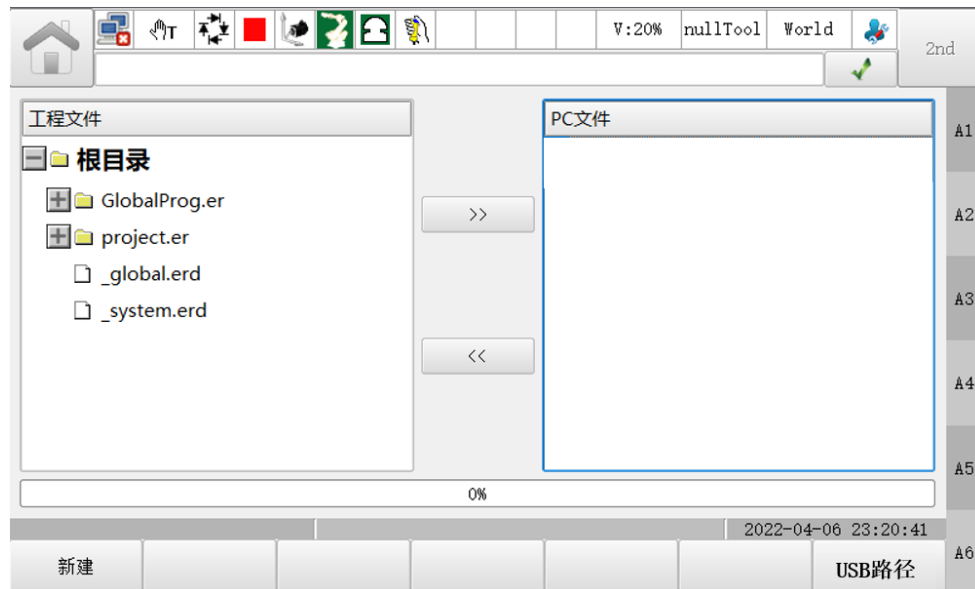
如下操作是工程导入的指导步骤，工程导出与其相似。

步骤 1 将 U 盘插入示教编程器的 USB 接口。

步骤 2 点击，进入工程管理界面。

步骤 3 点击右侧菜单中的“管理”，进入如图 4-5 所示的界面。

图4-5 工程程序导入导出界面说明



点击右下方“USB 路径”。

- 支持导出控制器的工程和数据文件到 U 盘中。
- 支持将 U 盘中的工程和数据文件导入到控制器中。
- 支持在控制器中工程根文件夹下新建工程文件夹。
- 支持删除控制器中工程根文件夹下的文件以及文件夹。
- 支持在 U 盘中进行新建、删除、复制、粘贴、重命名、打开文件夹。

工程导出

1. 点击右下方“U 盘路径”，进入 U 盘
2. 从左侧“工程文件”列表中选择需要导出的程序文件。
3. 单击界面中间的“>>”，可将所选的程序文件导出至 U 盘。

工程导入

1. 从右侧“工程文件”列表中选择需要导出的程序文件。
2. 单击界面中间的“<<”，可将所选的程序文件导入至控制器。
3. 如果选中的文件为 _global.erd 文件则会弹出界面让用户勾选不被覆盖的文件。目前可供选择的变量类型如下：APOS、CPOS、DAPOS、DCPOS、AERA、USERCOOR、TOOL（与 EStunEdit 保持一致）。



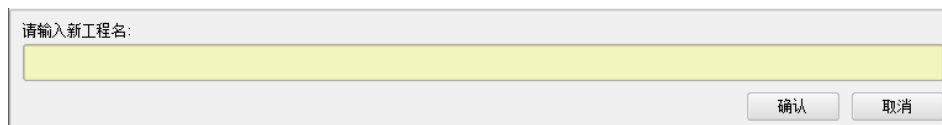
说明

在导入_global.erd 文件时，界面中勾选的数据类型不会被替换到机器人控制器当中，没被勾选或者界面中不存在的数据类型则会被删除，替换为界面右侧选中的_global.erd 数据。

新建

1. 点击下方的“新建”，弹出如图 4-6 所示的新建工程对话框。

图4-6 输入新工程名对话框



2. 点击“输入新工程名”的输入框，并在弹出的软键盘中输入需要新建的工程名，如：project0。
3. 点击“确认”，生成新工程成功，提示“请重载工程生效”，可在左侧示教盒中的工程文件列表中看到新生成的工程。

----结束

4.1.9 查找工程/程序

按照如下指导步骤查找工程或程序。

- 步骤 1 在工程管理界面中点击“查找”按钮，弹出“查找程序”界面，如图 4-7 所示。

图4-7 查找程序界面



- 步骤 2 在“名称”输入框中输入需要查找的文本，点击“查找”按钮进行查找

说明

在查找界面左侧存在包括工程名称(默认选中)、区分大小写、全字匹配三种筛选条件复选框，根据需要自由选中，此外，也可输入需查找名称中部分连续字符进行查找。

步骤 3 查找后将显示查找结果，分为未找到、找到一处、找到多处三种类型，当找到多处时，可点击“上一个”、“下一个”按钮进行查找结果的切换，工程列表中也会随着查找结果定位到该行,如图。

图4-8 查找结果显示图



步骤 4 点击“关闭”，退出查找功能。

----结束

4.1.10 程序信息

可快速查看当前程序的基本信息或者添加注释。

步骤 1 选中需要查看的程序，点击右侧的“程序信息”按钮，切换出程序信息页签。



步骤 2 程序页签中可查看该程序的信息和添加备注。



- 加载状态：加载、未加载、调用、/
- 运行状态：运行中、暂停、停止、并行、中断执行、中断并行、中断调用
- 自启动：是、否
- 写保护：是、否
- 备注信息：无备注信息时显示红色字体[可编辑]，单击备注信息栏，填写备注信息。
- 备注信息可输入最大长度为 35 个字符

4.2 程序编辑

按或在 Home 界面上点击“程序编辑”进入程序编辑界面。

说明

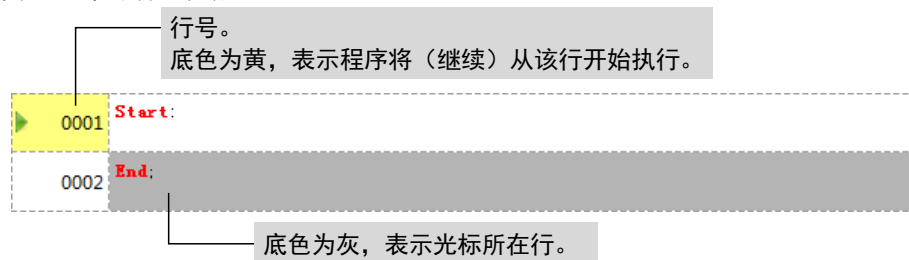
- 在进行指令编辑之前，必须先加载该程序所在的工程。
- 通过菜单键或 Home 界面中所进入的程序编辑，是正在加载的程序。若当前没有加载程序，请参见[加载/注销工程或程序](#)所述的内容加载需要编辑的程序。
- 用户也可以在工程管理界面中点击正被加载的程序，然后点击下方的“打开”，亦可进入程序编辑界面。使用该方法可在正加载的工程下，编辑尚未被加载的程序。

4.2.1 新增指令

按照如下指导步骤新建一个指令。

- 步骤 1 进入程序编辑编辑界面，如图 4-9 所示是一个空程序的界面说明。
在新程序中插入第一个指令时，请将光标移至“End”所在行。

图4-9 空程序界面说明

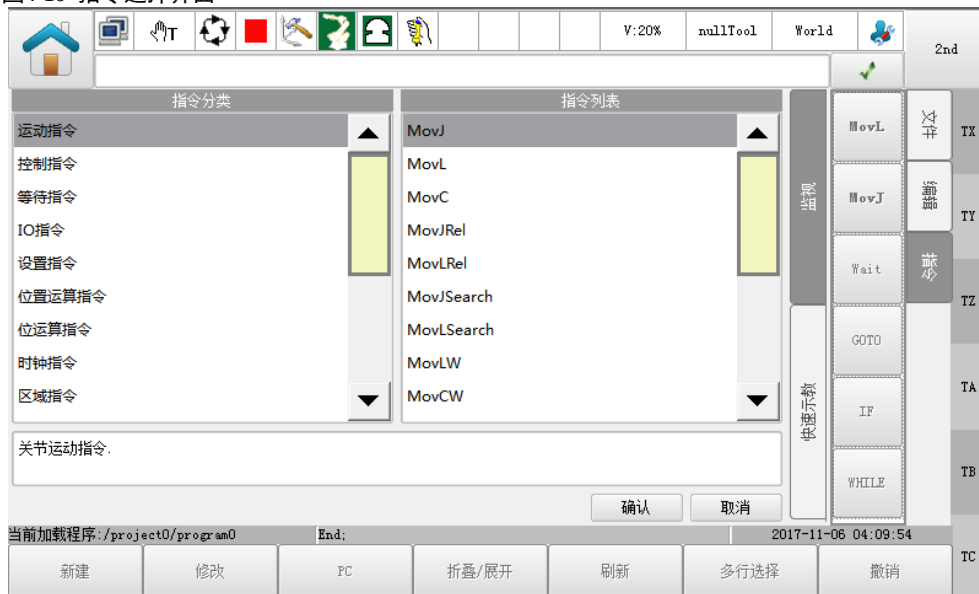


注意

- 新建成功的指令将插入在光标所在行的之前。所以，新建一个指令前，请移动光标至正确的位置。
- 新建成功后的程序将会被自动加载并进入程序编辑界面。

- 步骤 2 点击界面下方的“新建”，进入指令选择界面，如图 4-10 所示。

图4-10 指令选择界面



- 步骤 3 在“指令分类”中找到需要的指令，在右侧“指令列表”中选择具体的指令模板
例如，需要新建一个运动指令 MovJ，则选择“运动指令→MovJ”。

步骤 4 单击“确认”，进入“新建指令”的对话框，在此界面中可编辑所需插入的指令参数。图 4-11 是 MovJ 指令的参数编辑对话框。

图4-11 编辑 MovJ 指令的参数



说明

部分指令可能没有参数。

步骤 5 按照实际要求编辑指令的参数。编辑不同的参数时，可能使用如下方式：

- 点击该参数的输入框，在弹出软键盘中输入正确的数值。
- 点击输入框后的下拉箭头符号并弹出的下拉列表中选择所需要的参数值。
- 对于表示位置的参数，如果要选择当前位置作为参数值，则直接点击“示教”。

步骤 6 参数编辑成功后，点击“插入”。
界面返回至指令编辑界面，表示成功插入一个 MovJ 指令，如图 4-12 所示。

图4-12 成功插入 MovJ 指令



步骤 7 按照上述的方法继续添加其他指令。

步骤 8 检查并确认所有程序的编辑无误后，可返回主页面进行其他操作。

---结束

4.2.2 撤销指令

若在编辑程序时发现了错误操作，想要撤销刚刚新建或修改的指令，可在指令编辑界面中点击下方的“撤销”，可取消并返回上一步的操作，如图 4-13 所示。



 说明

最多可撤销 64 步的指令。

尽管如此，若程序被注销、加载或重新打开，则无法进行撤销指令的操作。

4.2.3 其它操作

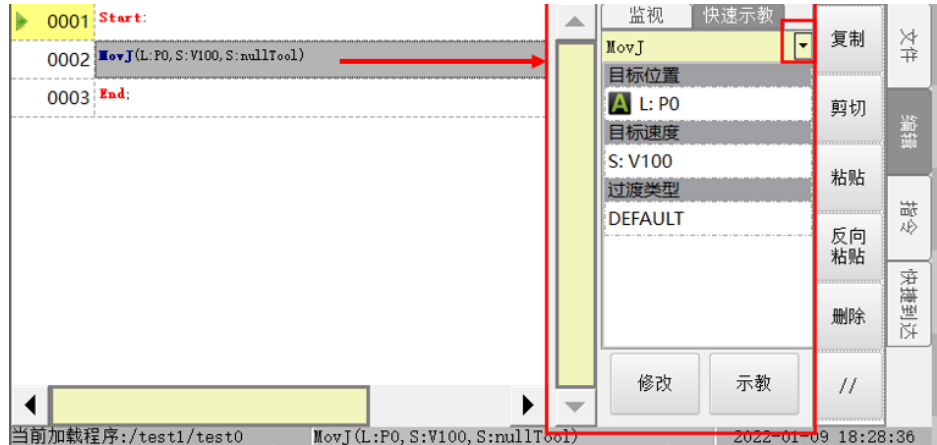
快速修改指令

参考如下步骤，快速修改指令：

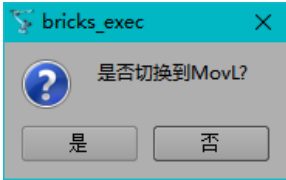
步骤 1 点击需要修改的指令

步骤 2 在右侧“快速示教”弹框中，选择需要修改的参数进行修改，如图 4-14 所示。

图4-14 快速修改某条指令的参数



1. 点击右侧下拉框，可以快速切换 MovJ 和 MovL 指令，以 MovJ 切换至 MovL 为例操作步骤如下，选中需要切换的 MovJ 指令
2. 在右侧“快速示教”界面中点击下拉菜单，选择需要切换成的 MovJ 指令，点击弹框“是否切换 MovJ 至 MovL”中的“是”。



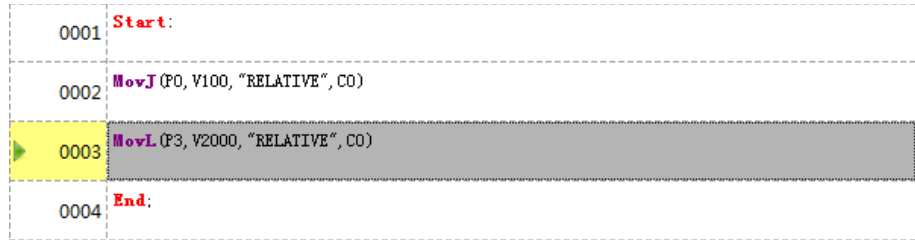
3. 点击“快速示教”界面上的“修改”按钮，完成 MovJ 和 MovL 指令的快速切换。

说明

- MovL 快速切换 MovJ 时，在“速率类型”不为“DEFAULT”时，也能切换成功。
- 若没有点击“快速示教”界面上的“修改”按钮，无法完成快速切换。
- MovJsearch/MovLsearch 不支持快速示教。

步骤 3 若要指定程序从某一指令开始执行，直接点击要开始执行的指令，点击下方的“PC”，指针将变更至该指令，如图 4-15 所示，已经将指令执行位置变更为 0003 行。

图4-15 变更指令执行的开始位置



----结束

 窍门

用户可点击右侧的“指令”快速新建一个常用的指令。

此外，用户可点击右侧的“编辑”，可在该界面执行以下操作：

- 复制：点击选择要复制的指令，然后点击“复制”，支持跨程序“复制”。
- 剪切：点击选择要剪切的指令，然后点击“剪切”，支持跨程序“剪切”。
- 粘贴：点击选择要粘贴的指令的位置，然后点击“粘贴”，支持跨程序“粘贴”。
- 反向粘贴：点击选择要粘贴的指令位置，然后点击“反向粘贴”，即将选择的多行指令按照原来相反的顺序粘贴到该位置，支持跨程序“反向粘贴”。
- 删除：点击选择要删除的指令，然后点击“删除”。
- //：点击选择要注释的指令，然后点击“//”。
- ToPC：点击“ToPC”可使程序编辑界面回到指针所在行。



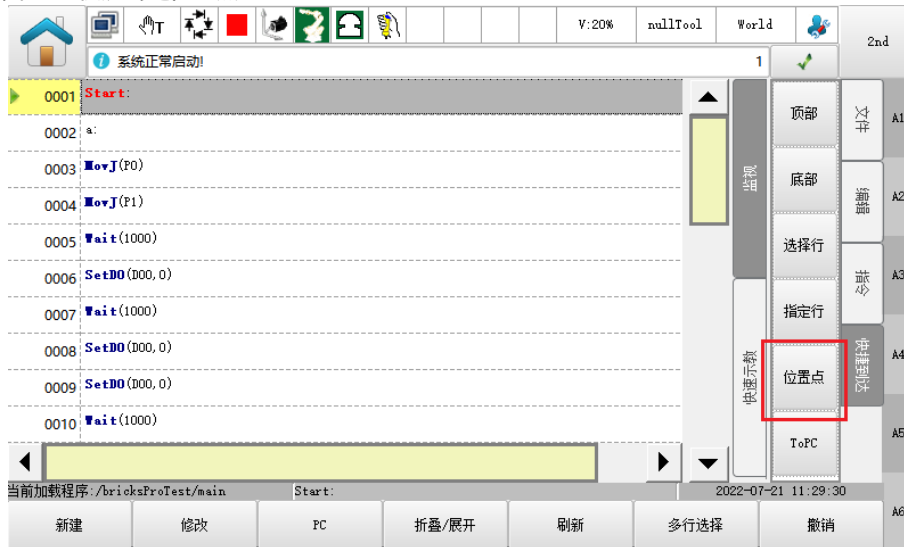
说明

- 跨程序，是指在同一工程下的不同程序。
- 指令支持跨程序复制、剪切、粘贴、反向粘贴等操作，不支持跨工程操作。
- 粘贴后的指令区分变量类型。若指令包含程序变量，且当前加载的程序没有相同程序变量，执行会报错；若当前加载程序中由同名变量，可正常执行，变量值以当前程序下的为准。全局变量和工程变量不会由影响。
- 粘贴包含程序变量的指令，若当前加载程序中没有相应变量，可进行修改，新建变量后，可正常执行指令。
- 关于各指令的详细说明，请参见“第 6 章”。

指令点一点到达

在手动模式下，用户可在程序页面中，点击右侧“快捷到达”→“位置点”，以手动定位至目标点。如图 4-16 所示。

图4-16 快捷到达位置点



在上浮的选项中选择目标“点”以及运行的指令“MovJ”或“MovL”，然后按住“Go”，TCP 将持续运行至目标点；松开“Go”可停止 TCP 的运动，如图 4-17 所示。

图4-17 选择目标点并运行

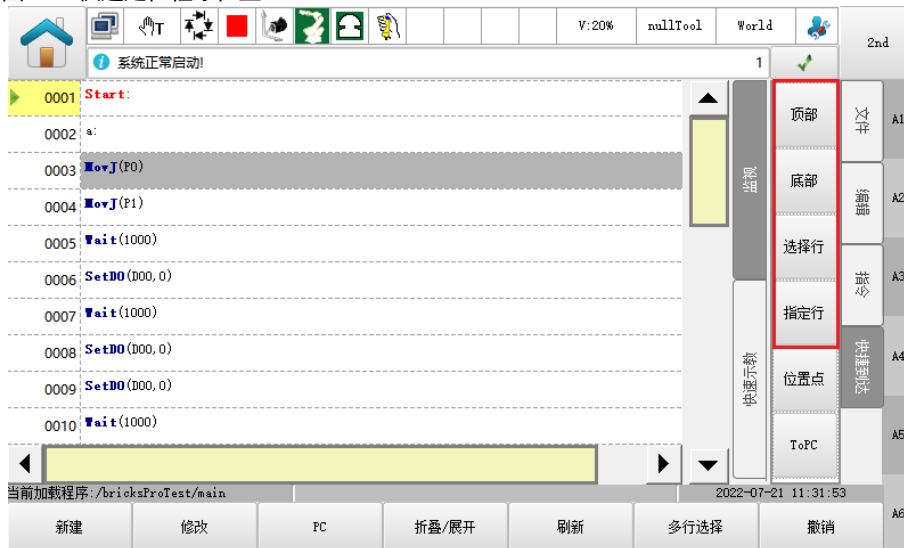


- 请选择指定目标点的指令行，上浮的选项中的“点”会自动出现目标点的变量。
指令行中出现多个点时，可在下拉列表中选择需要的目标“点”。
- 运行至目标点时的指令操作仅限“MovJ”和“MovL”可选。
- 若指令中带了 tool、usercoord 这些坐标系参数，请自行将坐标系切换至这些设定的坐标系再进行点位运动，避免执行出错导致撞机相关问题的产生。

快速定位程序行位置

用户可在程序页面中，点击右侧“快捷到达”，并选择需要进行快速定位的位置，如“顶部”、“底部”、“选择行”或者“指定行”，如图 4-18 所示。

图4-18 快速定位程序位置



- 点击“顶部”，程序行位置的显示将定位至“Start”行。
- 点击“底部”，程序行位置的显示将定位至“End”行。
- 点击“选择行”，程序行位置的显示将定位至当前选择行的位置。
- 点击“指定行”，并在上浮选项中选择需要定位的“标签”或“行号”，程序行位置的显示将定位至所指定位置。

图4-19 快速定位指定的程序行



运动执行指针

运动执行指针，用来指向当前执行的运动段或者当前机器人停留的运动段。

在程序编辑界面新增运动执行指针，其显示位置仍然在选中行的左侧。可以跟执行指针一起指向同一条运动指令。

1. Start:

```
2. MovJ (P0)
3. MovL (P1)
4. SetDO (DO1, 0)
5. INT0 = INT0 + 1
6. Wait (2000)
7. MovL (P2)
8. Wait (2000)
9. End;
```

上述示例程序所示：

1. 执行到第 2 行时，正常的执行指针与运动指针均显示在该行；
2. 当执行指针执行到第 4 行 SetDO 指令时，执行指针会显示到该行，但因为机器人停在第三行 MovL 的 P1 点位置，故运动指针会停在第三行。
3. 直到系统执行到第 7 行 MovL (P2) 处，运动执行指针会跳转到该行显示。

4.3 程序数据

按  或在 Home 界面上点击“程序数据”进入程序数据界面。

在程序数据界面中，点击下方的“刷新”，可更新变量列表。

程序数据（变量）分为系统、全局、工程和程序四个范围。

- 系统（S）：系统初始定义的一些变量。
- 全局（G）：所有工程或程序都可使用的变量。
- 工程（P）：同一个工程下的程序都可使用的变量。
- 程序（L）：仅当前程序可使用的变量。

不同域支持不同变量类型，详细说明请参见“第 5 章变量介绍”。

4.3.1 新增变量

按照如下指导步骤新建一个变量。

步骤 1 进入程序数据界面，如图 4-20 所示。

图4-20 程序数据界面



步骤 2 点击需要新建的所属范围，如“全局变量”。

步骤 3 点击“新建”，在弹出的新建变量对话框中设置相关参数，如图 4-21 所示。

图4-21 新建一个变量



步骤 4 选择所要新建的变量所属的“域”（范围）。

步骤 5 设置要新建的变量的分类和类型，表 4-1 列出了所有分类和类型的说明。

表4-1 变量类别说明

分类	类型	默认名称前缀
IO 数据类型	数字量输入	DI
	数字量输出	DO
	模拟量输入	AI
	模拟量输出	AO
	虚拟数字量输入	SimDI
	虚拟数字量输出	SimDO
	虚拟模拟量输入	SimAI
	虚拟模拟量输出	SimAO
PLC 数据类型	PLC 实数变量	PLCREAL
	PLC 布尔变量	PLCBOOL
	PLC 整型变量	PLCDINT
	PLC 短整型变量	PLCINT
socket 数据类型	socket 名	Socket
位置数据类型	关节位置类型	P
	关节位置数组	APOSARRAY
	关节相对位置类型	DAPOS
	坐标系位置类型	P
	坐标系位置数组	CPOSARRAY
	坐标系相对位置类型	DCPOS
区域数据类型	标准区域	AREA
	多边形区域	POLYHEDRON
基本数据类型	一维实数数组	RealOneArray
	一维布尔数组	BoolOneArray
	一维整型数组	IntOneArray
	字符串变量	STRING
	实数变量	REAL
	布尔变量	BOOL
	整形变量	INT
摆动数据类型	摆弧变量	WEAVE
时钟数据类型	时钟类型	CLOCK

分类	类型	默认名称前缀
码垛数据变量	码垛变量	PALLET
系统数据类型	工具坐标系	TOOL
	中断标识	INTERRUPT
	用户坐标系	USERCOORD
	转弯区变量	ZONE
	速度变量	SPEED
	动坐标系	SYNCOORD
	变位机坐标系	POSITIONER
	负载变量	PAYLOAD
	伺服低速增益变量	LsScale0
	伺服低速阈值变量	LsThresh0
	外部 TCP 坐标系	EXTTCP

步骤 6 点击“名称”的输入框，并在弹出的软键盘中输入想要定义的名称。



说明

以下“名称”为系统保留，主要包含基础函数、指令和变量名称，以及特定名称等。用户不可创建同名的变量。

基础函数：

and、break、do、else、elseif、end、false、for、function、if、in、local、nil、not、or、repeat、return、then、true、until、while、assert、tostring、tonumber、rawget、xpcall、ipairs、print、pcall、rawset、error、loadfile、rawequal、load、pairs、require、collectgarbage、dofile、next、select、type、getmetatable、goto、setmetatable、debug、log、max、acos、huge、pi、cos、deg、tan、random、randomseed、ceil、floor、rad、abs、sqrt、modf、asin、min、fmod、log、atan、exp、sin、coroutine、math、io、os、package、string、table、byte、find、find2、char、gsub、len、sub、lower、trim、upper

变量类型：

BOOL、INT、REAL、STRING、BOOLONEARRAY、INTONEARRAY、REALONEARRAY、APOS、CPOS、TOOL、USERCOORD、AI、AO、DI、DO、SimAI、SimAO、SimDI、SimDO、RET、AREA、POLYHEDRON、CLOCK、PLCREAL、SOCKET、PALLET、WEAVE、SPEED、ZONE、PAYLOAD、EXTTCP、POSITIONER、SYNCOORD、LsScale、LsThresh、PLCREAL、PLCBOOL、PLCINT、PLCDOMNT、APOSARRAY、CPOSARRAY、INTERRUPT

指令名称：

MovJ、MovL、MovC、MovCircle、MovJRel、MovLRel、MovLOffset、MovJSearch、MovLSearch、MovLW、MovCW、MovCircleW、MovH、MovArch、OnDistance、OnParameter、LABEL、GOTO、ENDIF、THEN、IF、THEN、ELSIF、ELSE、ENDWHILE、DO、WHILE、CALL、RUN、KILL、RETURN、/*...*/、...=...、SetDO、SetAO、SetSimDO、SetSimAO、WaitDI、WaitDI8421、WaitAI、WaitSimDI、WaitSimDI8421、WaitSimAI、PulseOut、PulseSimOut、GetDI8421、GetSimDI8421、SetDO8421、SetSimDO8421、GetSimDIToVar、SetSimDOByVar、GetSimAIToVar、SetSimAOByVar、SetDIEdge、SetSimDIEdge、Wait、WaitFinish、WaitCondition、GetCurAPos、GetCurCPos、APosToCPos、CPosToAPos、CPosToCPos、ToolOffset、UserOffset、WaitWObj、MovLSync、MovJSyncQuit、MovLSyncQuit、SetTargetPos、SetSyncoord、SynCToUserC、WaitConvDis、SimConveyorOn、SimConveyorOff、ReceiveWObj、ResetWObjBuf、BitAnd、BitNeg、BitOr、BitXOr、BitLSH、BitRSH、CLKStart、CLKStop、CLKRead、CLKReset、AreaActivate、AreaDeactivate、PolyhedronAreaActivate、PolyhedronAreaDeactivate、TrigCam、WaitFinishCAM、GetCamPos、

SendMessage、Tracking、PalletReset、PalletToPut、PalletFromPut、PalletToGet、PalletFromGet、SetMatrix、GetMatrix、SocketCreate、SocketClose、SocketSendStr、SocketSendReal、SocketSendInt、SocketReadReal、SocketReadInt、SocketReadStr、GetModConState、ReadModbusReg、WriteModbusReg、SoftFloatStart、SoftFloatStop、IToStr、RToStr、StrToI、StrToR、APosToStr、CPosToStr、TranStrToInt、TranStrToReal、TranStrToApos、TranStrToCpos、MA_ELF_START、MA_ELF_STOP、SetTool、SetCoord、SetExternalTCP、SetPositioner、SetPayload、SetJointDyn、SetCartDyn、SetBlendParam、SetOverRide、SetSingularPass、SetColliEnable、SetAxisColliParam、GetCurOverRide、Hand、CalcTool、CalcCoord、SetRtToErr、SetRtInfo、SetRtWarning、Stop、RefRobotAxis、SetMotionMode、AutoGainEnable、AutoGainDisable、SetAxisVibraBLevel、SetRestorePC

特定名称

POSCFG、PLACEOPTIO

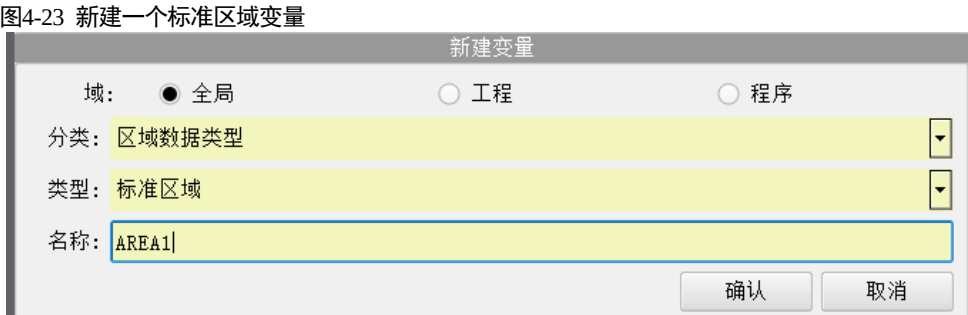
步骤 7 确认所要新增的变量设置无误后，点击“确认”。
界面将返回至变量列表，并显示新增的变量数据。图 4-22 表示成功新增了一个全局的位置变量 P0。



4.3.2 标准区域变量

按照如下操作步骤新建一个标准区域变量。

- 步骤 1 进入程序数据界面，单击“新建”。
- 步骤 2 在弹出的对话框中选择“分类”为“区域数据类型”，“类型”为“标准区域”，并输入想要的“名称”，如：AREA1，如图 4-23 所示。



- 步骤 3 确认所要新建的标准区域变量设置无误后，点击“确认”。
- 步骤 4 界面将返回至变量列表，并显示刚新建的变量数据。图 4-24 表示成功新建了一个全局的标准区域变量 AREA1。

图4-24 成功新建了一个标准区域变量

变量列表	参数列表	值
+ 系统 - 全局变量 - AREA AREA0 AREA1 + PAYLOAD + USERCOOR 工程变量 程序变量	id activate isInArea initActivate areaType areaShape refSysScope refSysName enInPut enInHigh diType	1 false false false WorkArea Box system World false true di

----结束

4.3.3 多边形区域变量

按照如下操作步骤新建一个多边形区域变量。

- 步骤 1 进入程序数据界面，单击“新建”。
- 步骤 2 在弹出的对话框中选择“分类”为“区域数据类型”，“类型”为“多边形区域”，并输入想要的“名称”，如：POLYHEDRON1，如图 4-25 所示。

图4-25 新建一个多边形区域变量

新建变量

域：

☒ 全局

☐ 工程

☐ 程序

分类：

区域数据类型

类型：

多边形区域

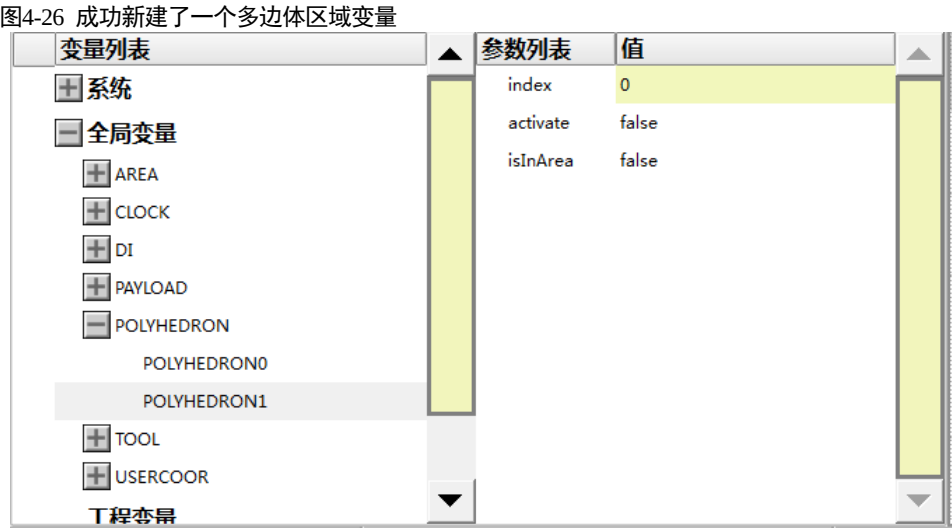
名称：

POLYHEDRON1

确认

取消

- 步骤 3 确认所要新建的多边形区域变量设置无误后，点击“确认”。
- 步骤 4 界面将返回至变量列表，并显示刚新建的变量数据。图 4-26 表示成功新建了一个全局的多边形区域变量 POLYHEDRON1。



-----结束

4.3.4 工具坐标系变量

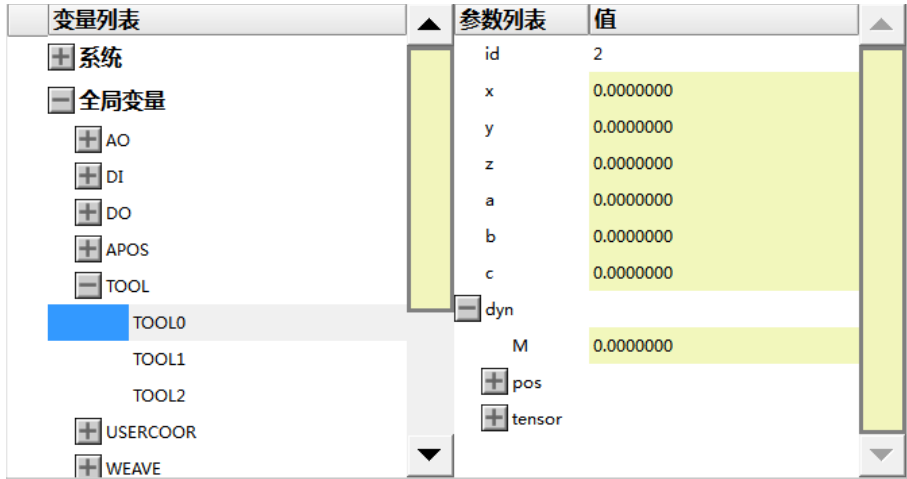
按照如下操作步骤新建一个工具坐标系变量。

- 步骤 1 进入程序数据界面，单击“新建”。
- 步骤 2 在弹出的对话框中选择“分类”为“系统数据类型”，“类型”为“工具坐标系”，并输入想要的“名称”，如 TOOL0，如图 4-27 所示。



- 步骤 3 确认所要新建的工具坐标系变量设置无误后，点击“确认”。
- 步骤 4 界面将返回至变量列表，并显示刚新建的变量数据。图 4-28 表示成功新建了一个全局的工具坐标系变量 TOOL0。

图4-28 成功新建了一个工具坐标系变量



----结束

4.3.5 用户坐标系变量

按照如下操作步骤新建一个用户坐标系变量。

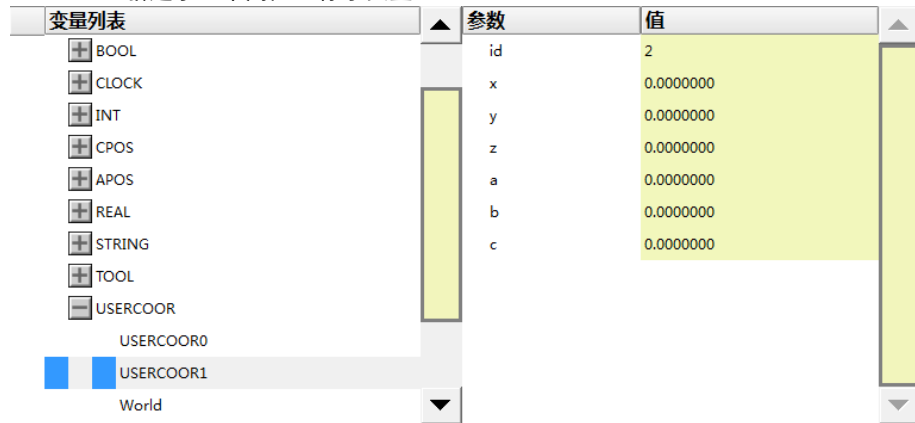
- 步骤 1 进入程序数据界面，单击“新建”。
- 步骤 2 在弹出的对话框中选择“分类”为“系统数据类型”，“类型”为“用户坐标系”，并输入需要的“名称”，如 USERCOOR1，如图 4-29 所示。

图4-29 新建一个用户坐标系变量



- 步骤 3 确认所要新建的用户坐标系变量设置无误后，点击“确认”。
- 步骤 4 界面将返回至变量列表，并显示刚新建的变量数据。图 4-30 表示成功新建了一个全局的用户坐标系变量 USERCOOR1。

图4-30 成功新建了一个用户坐标系变量



----结束

4.3.6 外部工具坐标系变量

按照如下操作步骤新建一个外部工具坐标系变量。

- 步骤 1 进入程序数据界面，单击“新建”。
- 步骤 2 在弹出的对话框中选择“分类”为“系统数据类型”，“类型”为“外部 TCP 坐标系”，并输入需要的“名称”，如 EXTTCP1，如图 4-31 所示。

图4-31 新建一个外部工具坐标系变量

新建变量

域：☒ 全局 ☐ 工程 ☐ 程序

分类：系统数据类型

类型：外部TCP坐标系

名称：EXTTCP1

确认取消

- 步骤 3 确认所要新建的外部工具坐标系变量设置无误后，点击“确认”。
- 步骤 4 界面将返回至变量列表，并显示刚新建的变量数据。图 4-32 表示成功新建了一个全局的外部工具坐标系变量 EXTTCP1。

图4-32 成功新建了一个外部工具坐标系变量

变量列表	参数列表	值
+ 系统 [-] 全局变量 [-] EXTTCP EXTTCP0 EXTTCP1 工程变量 程序变量	id	2
	x	0.0000000
	y	0.0000000
	z	0.0000000
	a	0.0000000
	b	0.0000000
	c	0.0000000

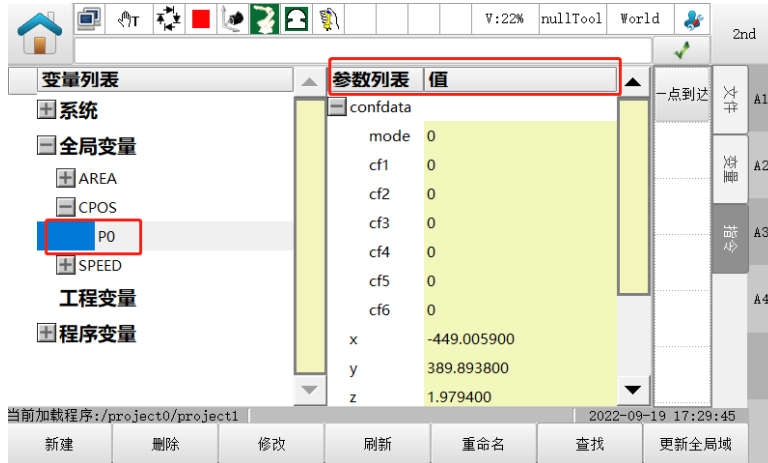
----结束

4.3.7 其它操作

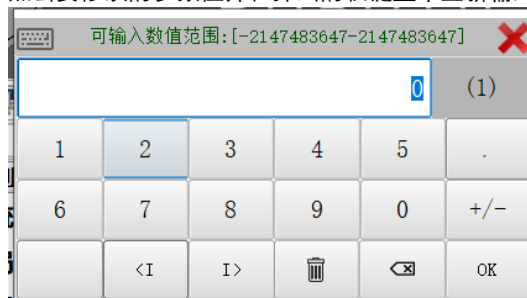
修改变量值

按照如下指导步骤修改一个变量的值：

步骤 1 选择所要修改的变量，在界面右侧将显示该变量所有的“参数列表”及“值”。



步骤 2 点击要修改的参数值并在弹出的软键盘中重新输入正确的数值（注意软键盘上参数值的取值范围）。



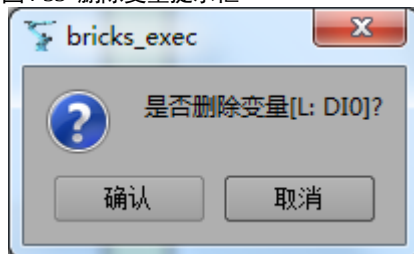
步骤 3 对于某些特殊的变量(点击“值”无法进入编辑模式的变量)，可点击下方的“修改”，界面将显示相应的修改界面来辅助用户的操作。

----结束

删除变量

步骤 1 选择需要删除的变量，点击下方的“删除”，将弹出图 4-33 所示的提示框。

图4-33 删除变量提示框



步骤 2 点击“确认”，该变量将被删除。

重命名变量

步骤 1 选择需要重新命名的变量，点击下方的“重命名”，将弹出“修改变量名称”的界面，如图 4-34 所示。

图4-34 重命名变量



修改变量名称

旧的变量名:

DIO

新的变量名:

确认 取消

步骤 2 点击“新的变量名”的输入框，并在弹出的软键盘中键入修改后的名称。

步骤 3 再点击“确认”，该变量将被重命名。

查找变量

步骤 1 在“变量列表”页面中点击下方菜单“查找”，并在上浮的选项中输入需要查找的变量“名称”如图 4-35 所示。



步骤 2 选择需要查找的“域”或“类型”能够更加精确的找到想要查找的变量。

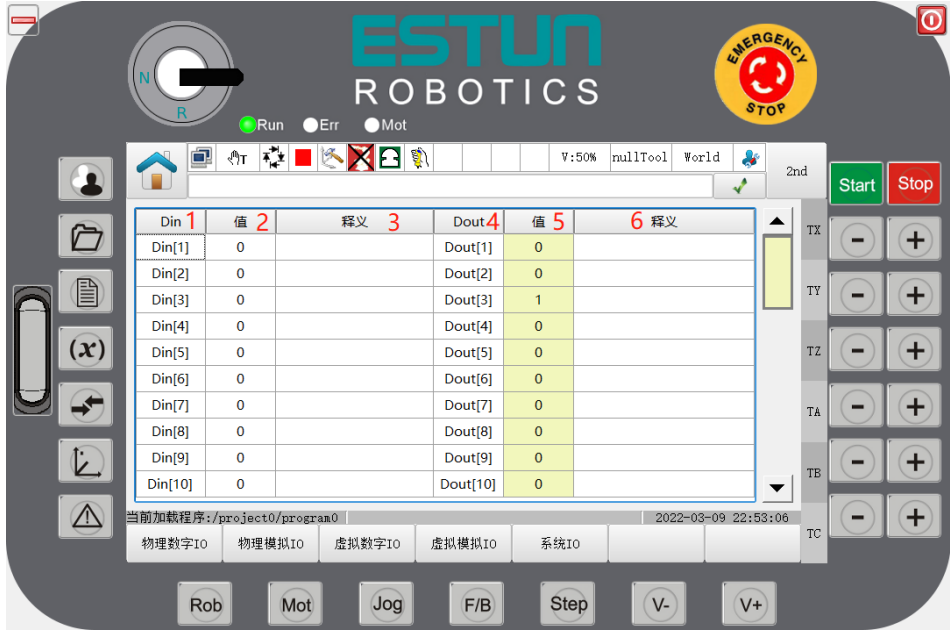
步骤 3 点击“查找”。

步骤 4 查找到匹配的名称，出现多个结果，通过点击“上一个”或“下一个”进行。

4.4 IO 检测

按或在 Home 界面上点击“IO 检测”进入 IO 检测界面，主要由：物理数字 IO、物理模拟 IO、虚拟数字 IO、虚拟模拟 IO、系统 IO 五个页签组成，默认进入的页签是物理数字 IO 界面，如图 4-36 所示。

图4-36 IO 检测界面

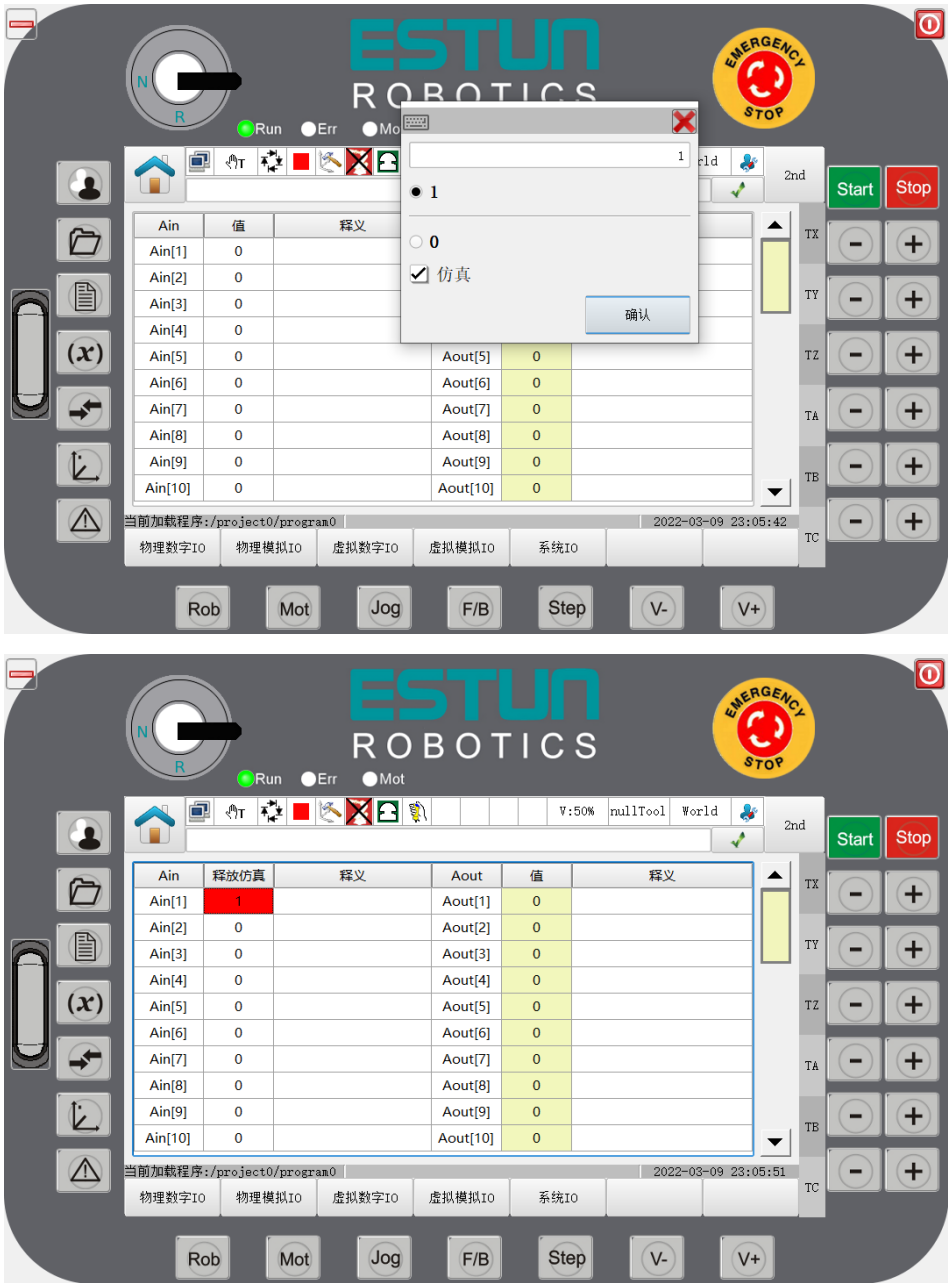


序号	含义	取值
1	输入量	Din、Ain、SDin、SAin
2、5	输入信号	0: 低电平; 1: 高电平
4	输出量	Dout、Aout、SDout、SAout
3、6	释义	输出量和输入量的注释功能

步骤 1 修改值的步骤：点击需要修改数字量 IO 的值，弹出对应的 01 键盘，如下图所示：



步骤 2 选择 0 或 1，同时勾选“仿真”复选框后，点击“确认”按钮，如下图所示对应端口的值变成设置的值且成红色，表示修改成功。



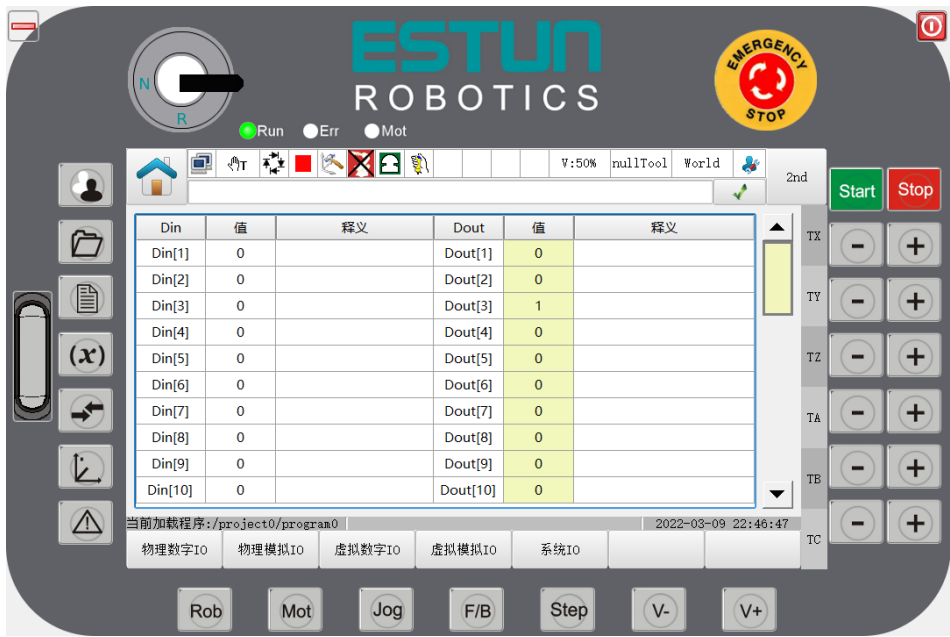
步骤 3 若需要释放其中一个端口的仿真值，点击需要修改的 IO 数字量，在对应的弹框中去勾选“仿真”，点击“确认”按钮即可。

步骤 4 点击上方“释放仿真”，根据界面提示确认释放，即可将物理 IO 的端口值恢复成初始状态，而虚拟 IO 只改变仿真状态，但已修改的端口值不变。

物理数字 IO

点击下方“物理数字 IO”，可切换至“物理数字 IO”界面：

图4-37 物理数字 IO 界面



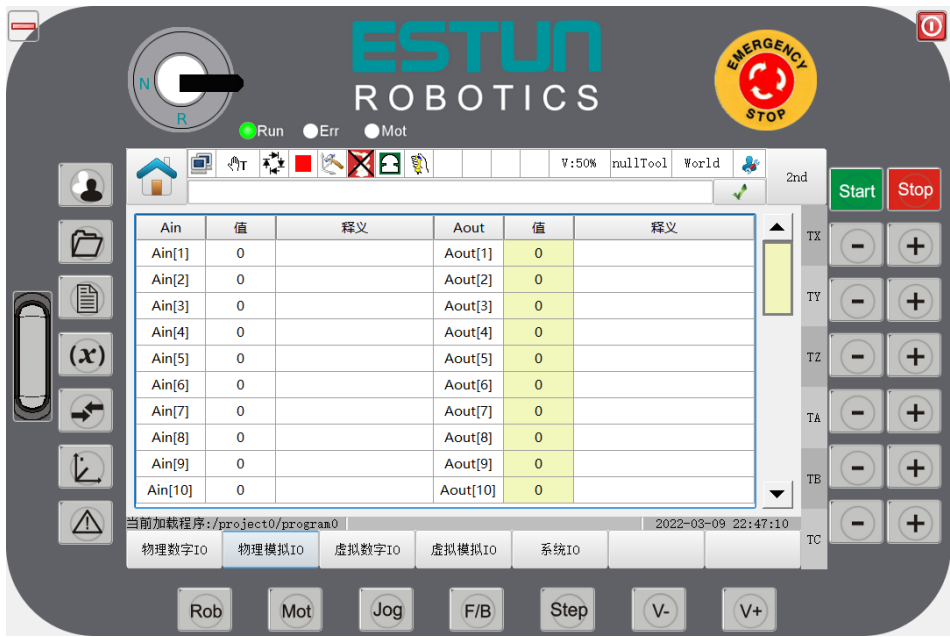
参数介绍如下：

参数	参数介绍
Din	数字量输入端口
Dout	数字量输出端口
值	0：低电平 1：高电平
释义	IO 注释功能

物理模拟 IO

点击下方“物理模拟 IO”按钮，切换至“物理模拟 IO”页签。

图4-38 物理模拟 IO 界面

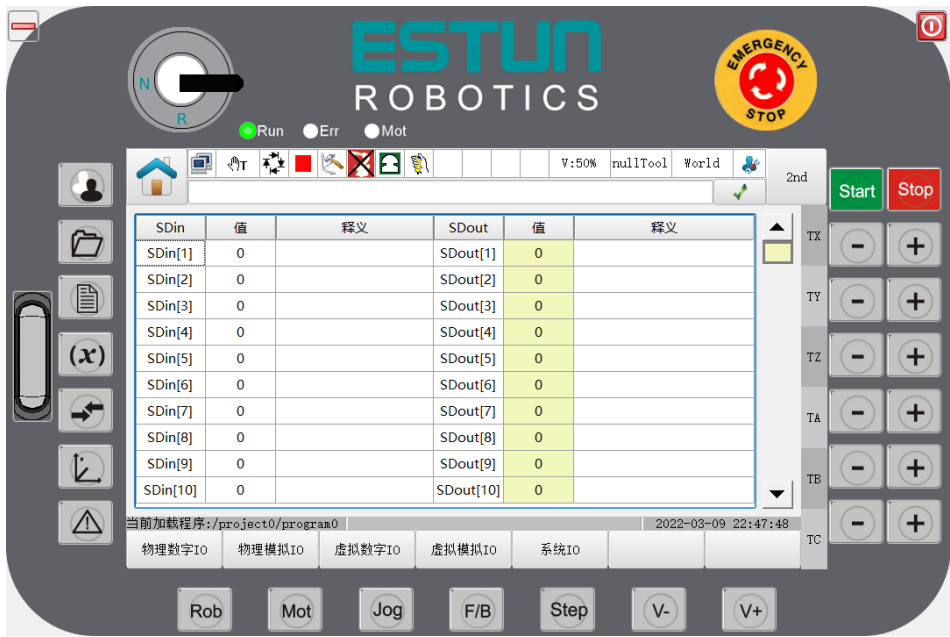


参数介绍如下：

参数	参数介绍
Ain	模拟量输入端口
Aout	模拟量输出端口
值	0：低电平 1：高电平
释义	IO 注释功能

虚拟数字 IO

图4-39 虚拟数字 IO 界面

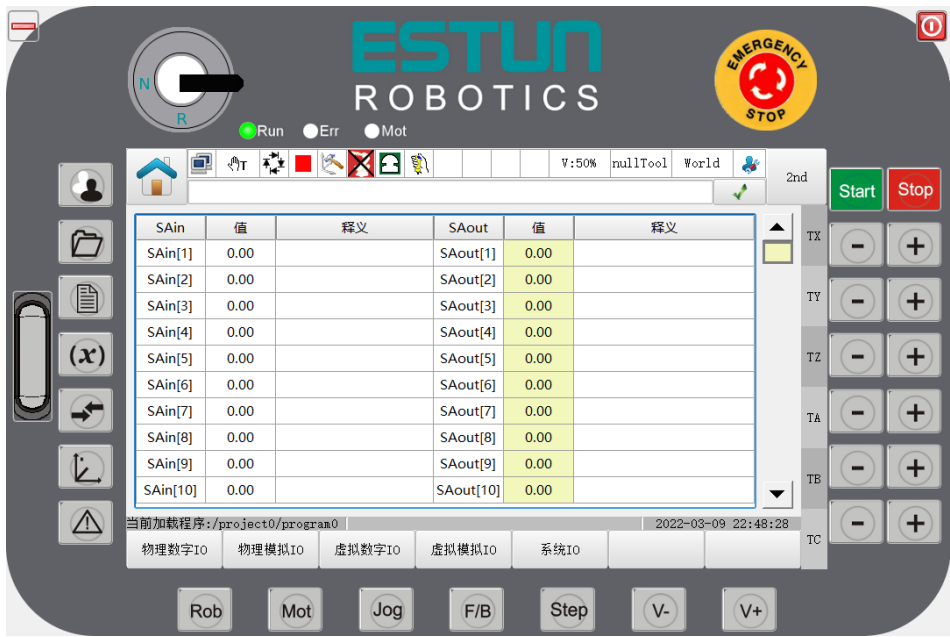


参数介绍如下：

参数	参数介绍
SAin	虚拟数字量输入端口
SAout	虚拟数字量输出端口
值	0：低电平 1：高电平
释义	IO 注释功能

虚拟模拟 IO

图4-40 虚拟模拟 IO 界面

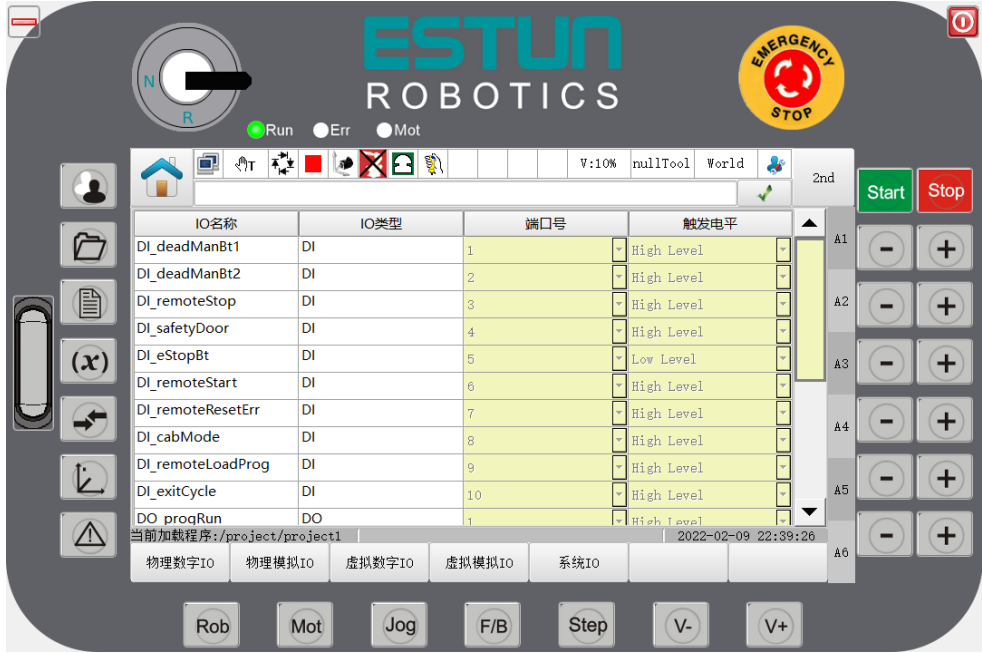


参数	参数介绍
SDin	虚拟模拟量输入端口
SDout	虚拟模拟量输出端口
值	0：低电平 1：高电平
释义	IO 注释功能

系统 IO

点击下方“系统 IO”按钮，在弹出的对话框输入密码后，可进入系统 IO 界面。系统 IO 默认不可修改。系统 IO 界面如图 4-41 所示。

图4-41 系统 IO 界面检测界面



4.5 位置管理

JOG 模式下，可以点动多轴，机器人进行多轴的同时动作，并且速度限制在国标（250mm/s）以内。

按或在 Home 界面上点击“点动管理”插件组的“位置管理”，进入位置管理界面，如图 4-42 所示。

图4-42 位置管理界面



关节坐标

点击“关节坐标”，界面显示机器人当前的关节位置信息（A1、A2、A3、A4、A5、A6）。

世界坐标

点击“世界坐标”，界面显示机器人 TCP 在世界坐标系下的位置信息（X、Y、Z、A、B、C）。

用户坐标

若“参考坐标系”设置为“G: USERCOOR”，点击“用户坐标”，界面显示机器人 TCP 在用户坐标系下的位置信息（RX、RY、RZ、RA、RB、RC）。

外部工具坐标

若“参考坐标系”设置为“G: EXTTCB”，点击“外部工具坐标”，界面显示机器人 TCP 在外部工具坐标系下的位置信息（EX、EY、EZ、EA、EB、EC）。

变位机坐标

若“参考坐标系”设置为“G: POSITIONER”，点击“变位机坐标”，界面显示机器人 TCP 在变位机坐标系下的位置信息（PX、PY、PZ、PA、PB、PC）。

力矩

点击“力矩”，界面显示机器人本体各关节电机的额定转矩值（百分比）。

单圈值

点击“单圈值”，界面左侧显示各个轴的设置单圈值（点击相应位置可进行设置），右侧显示各个轴的实际单圈值。

寸动设置

若“点动模式”设置为“寸动”，点击“寸动设置”，界面显示机器人各轴的寸动参数。若要修改寸动参数，直接点击参数值的输入框并在弹出的软键盘中输入要修改的数值。

用户可修改“点动坐标系”为想要进行点动操作的坐标系，或者按示教编程器上的按钮来切换要想进行点动操作的坐标系。

回零

在机器人伺服励磁状态下，点击“回零”，并在弹出的对话框中输入正确的管理人员密码，界面将显示回零界面，如图 4-43 所示。

图4-43 回零界面

回零

	单圈值	单轴回零
A1	0	回零
A2	0	回零
A3	0	回零
A4	0	回零
A5	0	回零
A6	0	回零

关闭

整体回零

单轴回零：点击对应轴后“回零”按钮，执行单轴回零操作；

整体回零：点击“整体回零”按钮，执行所有轴回零操作。

说明

推荐使用单轴回零或指令回零，整体回零可能导致机器人零位精度丢失。

4.6 日志管理

按或在 Home 界面上点击“日志管理”进入日志管理界面。日志管理界面主要由报警、日志、帮助三个页签组成。

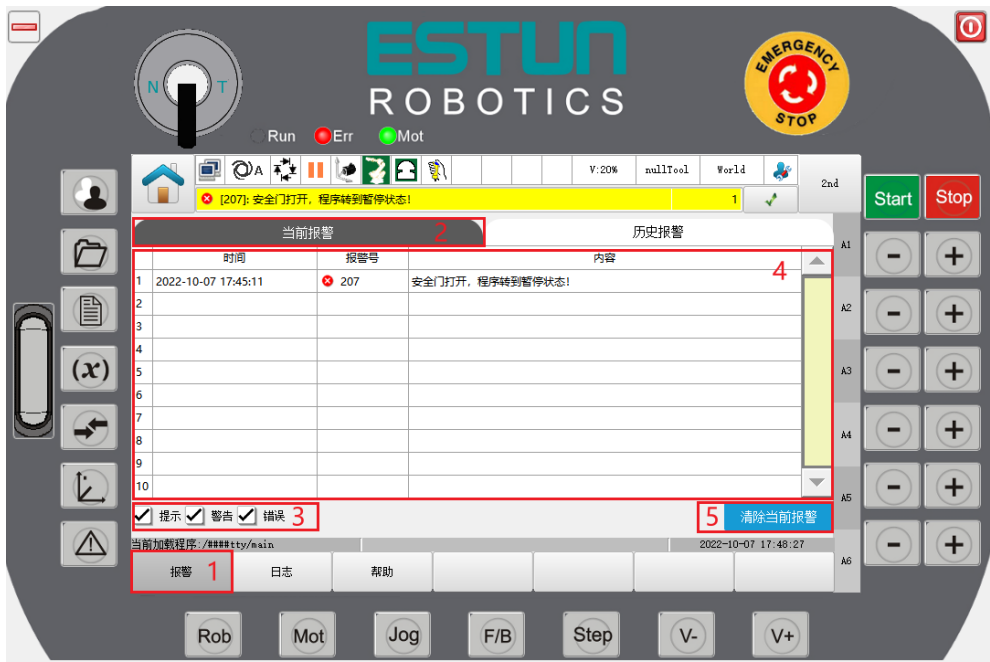
4.6.1 报警

报警页面分为“当前报警”和“历史告警”两个部分

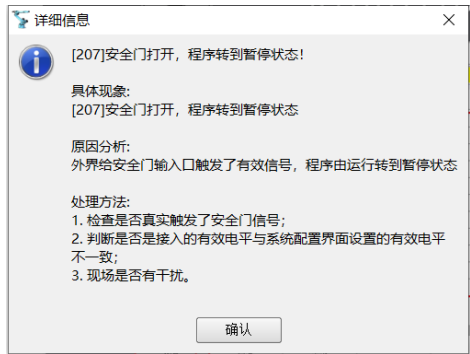
当前报警

进入“日志管理”界面后，点击“报警”按钮，切换到报警界面，若当前存在需要清除的报警时，优先显示当前报警界面。

图4-44 当前报警界面



- 1：报警按钮点击进入报警页面 2：当前报警按钮，点击进入“当前告警”页签，可查看当前产生告警。
- 4：告警显示页面，可看到报警产生时间、报警号、内容。点击告警对应的“内容”，可弹窗显示当前报警的“详细信息”如：具体现象、产生原因、解决方法

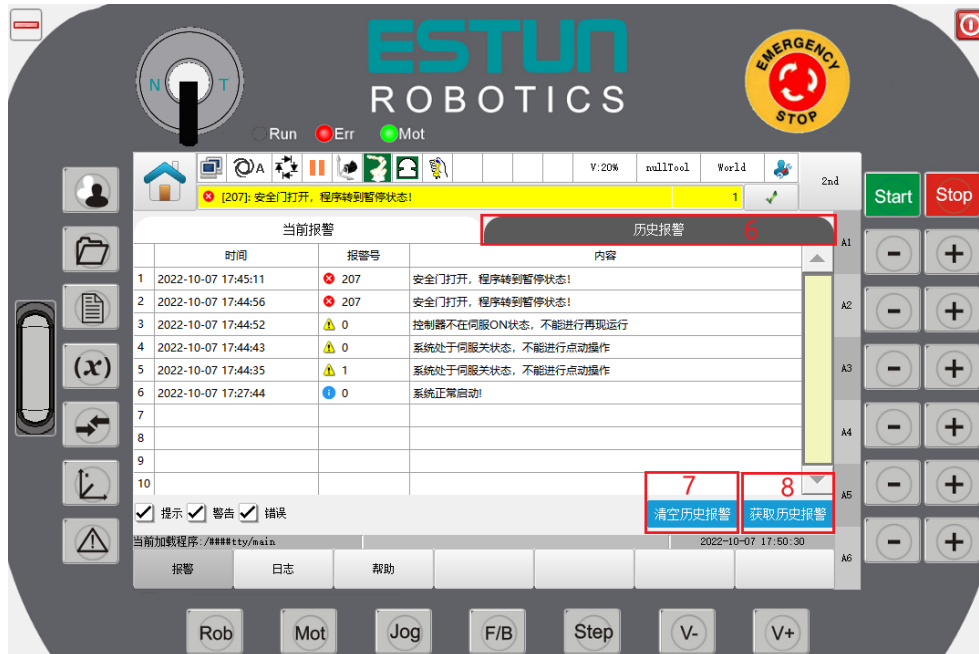


5: 清除当前报警按钮, 若存在当前告警, 点击此按钮可清除当前的告警, 使程序正常运行。作用同告警栏右侧的  按钮。用户可点击  清除告警操作。

历史报警

点击“报警页面”中的“历史报警”进入“历史报警”页面, 可查看历史产生的报警。

图4-45 历史报警界面



6: 历史报警按钮: 点击切换至“历史报警”页面, 界面将显示本次开机后所发生的报警信息。

7: 清空历史报警: 可清除所有的报警信息。

8: 获取历史报警: 同步控制器的所有历史报警信息, 并更新历史报警界面。

4.6.2 日志

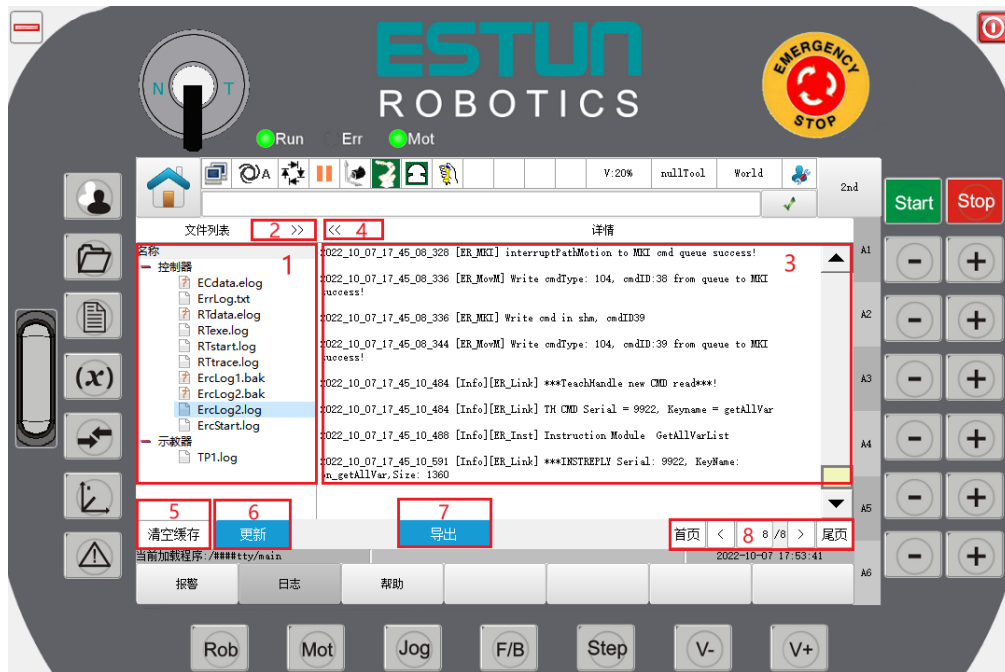
查看和导出系统日志的方式有三种途径:

- 主页>日志管理>日志
- 主页>通用设置>系统诊断
- 主页>高级设置>本地设置>关闭示教器>ESTool

日志

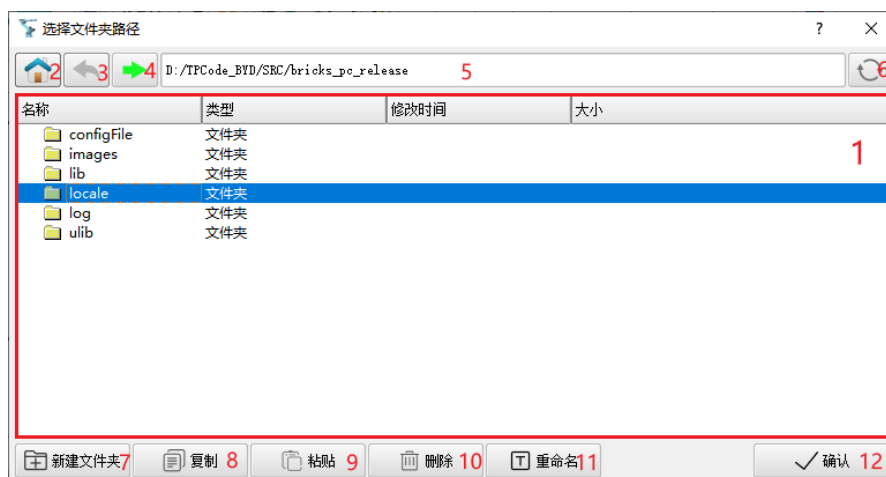
在“日志管理”界面点击“日志”按钮, 切换至日志界面。

图4-46 日志界面



- 1: 日志文件列表, 显示有日志文件名称。分为控制器端和示教器端。
- 2: 可展开文件列表, 显示文件大小。
- 3: 显示界面, 读取并显示对应日志文件的内容。
- 4: 可展开并扩展显示。
- 5: 清空缓存: 清空示教器本地缓存控制器日志文件。
- 6: 更新: 更新控制器与内核日志文件并刷新日志文件列表。
- 7: 导出: 弹出 U 盘目录浏览窗口(如下 4-19 所示)并选取需导出的文件夹路径。
- 8: 翻页按钮: 输入页码可进行翻页。

图4-47 文件管理界面



文件管理界面, 在进行 U 盘导出功能时会弹出显示供用户选择导出的路径:

- 1: 显示界面: 显示文件详细信息, 如: 名称、类型、修改时间、大小
- 2: 快捷按钮: 点击返回根目录。

- 3: 返回按钮: 点击返回上一级目录。
- 4: 显示选中文件夹的内容。
- 5: 显示到当前路径。
- 6: 刷新: 刷新目录显示。
- 7: 新建文件夹: 在当前文件夹下新建文件夹。
- 8: 复制: 复制选中的文件或文件夹。
- 9: 粘贴: 把复制的文件或文件夹粘贴到目标路径下。
- 10: 删除: 删除选中的文件或文件夹。
- 11: 重命名: 对选中的文件或文件夹重新命名。
- 12: 确认: 确认当前选中的文件夹路径。

系统诊断

在通用设置界面中点击“系统诊断”可进入系统诊断界面，如图 4-48 所示。

图4-48 系统诊断界面



- 点击“更新日志”，系统将进入诊断模式，并在片刻后生成日志。
- 点击“查看日志”，界面将显示最近一次系统诊断的日志。日志文件包括 ERC.log（控制器运行日志）、ErrLog.txt（控制器运行的故障日志）、RTstart.elog（控制器启动日志）等。
- 若需要将诊断日志文件导出，用户需自备 U 盘，并将其插入至示教编程器的 USB 端口，然后点击“导出日志”
- 所有日志文件(控制器日志和示教编程器日志)将被导出至 U 盘的根目录下。

启动 ESTool

示教编程器连接且用户类型已切换为“管理人员”的情况下，在“系统主页>高级设置>本地设置”，点击“电源管理”中的“关闭示教器”，如图 4-49 所示，启动 ESTool 进入示教器的维护模式，如图 4-50 所示，按顺序点击按键启动 ESTool。

图4-49 关闭示教编程器



图4-50 启动 ESTool

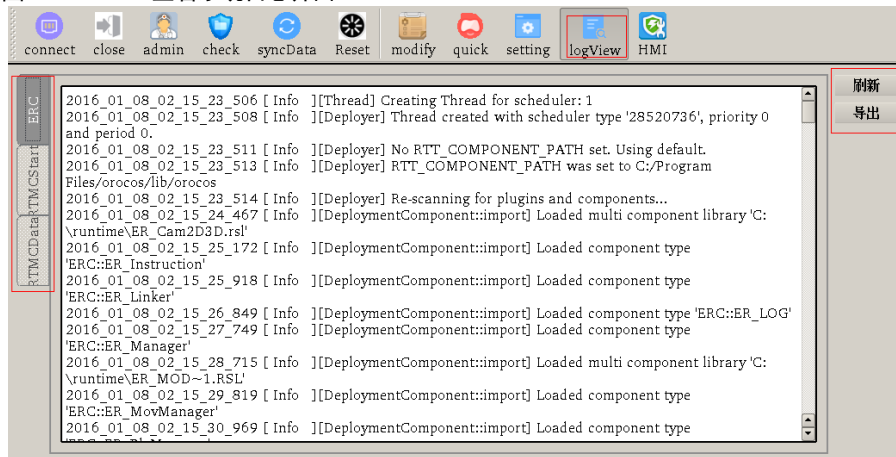


目前支持查看和导出的系统日志文件包括 ERC.log (控制器运行日志)、RTstart.log (控制器启动日志)、RTdata.elog (控制器运行数据日志)。

启动 ESTool 后，点击 logView，在界面左侧选择想要查看的系统日志，点击“刷新”即可查看日志信息，如图 4-51 所示。

若需要导出系统日志文件，将 U 盘插入示教编程器 USB 接口,在界面左侧选择想要导出的日志文件，点击界面右侧“刷新”，再点击“导出”即可。

图4-51 ESTool 查看系统日志界面



4.6.3 帮助

在“日志管理”界面点击“帮助”按钮，切换至帮助界面。

图4-52 帮助界面



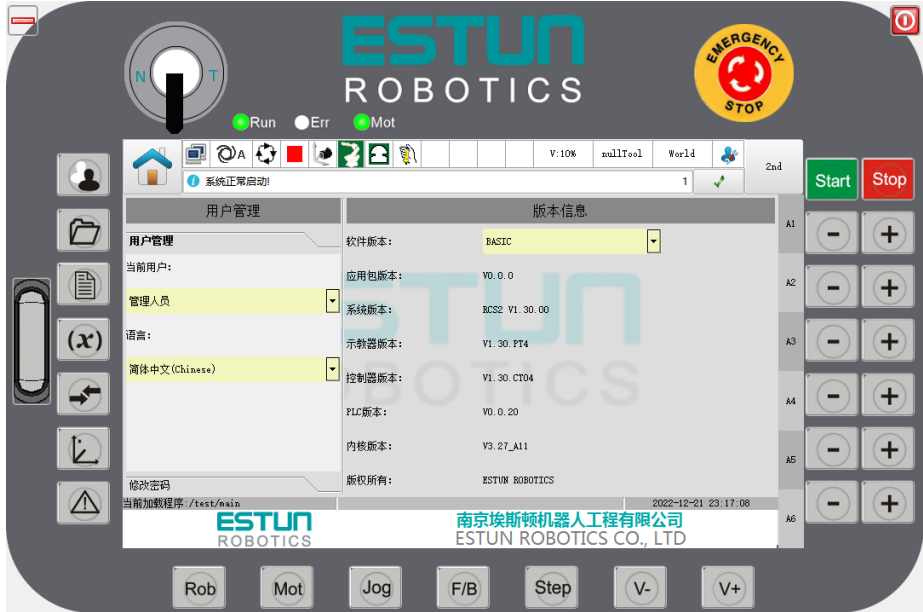
- 1: 显示区：显示解析后帮助文档的案例如：序号、类型、描述
- 2: 详细信息显示：显示选中案例的详细信息。
- 3: 搜索框：输入关键字，统会根据简述与详情中信息过滤检索。
- 4: 复选框：根据帮助类型过滤显示。
- 5: 翻页按钮：输入页码可进行翻页。

4.7 系统设置

4.7.1 用户管理

在 Home 界面上点击“用户登录”，进入用户登录界面，如图 4-53 所示。

图4-53 用户管理界面



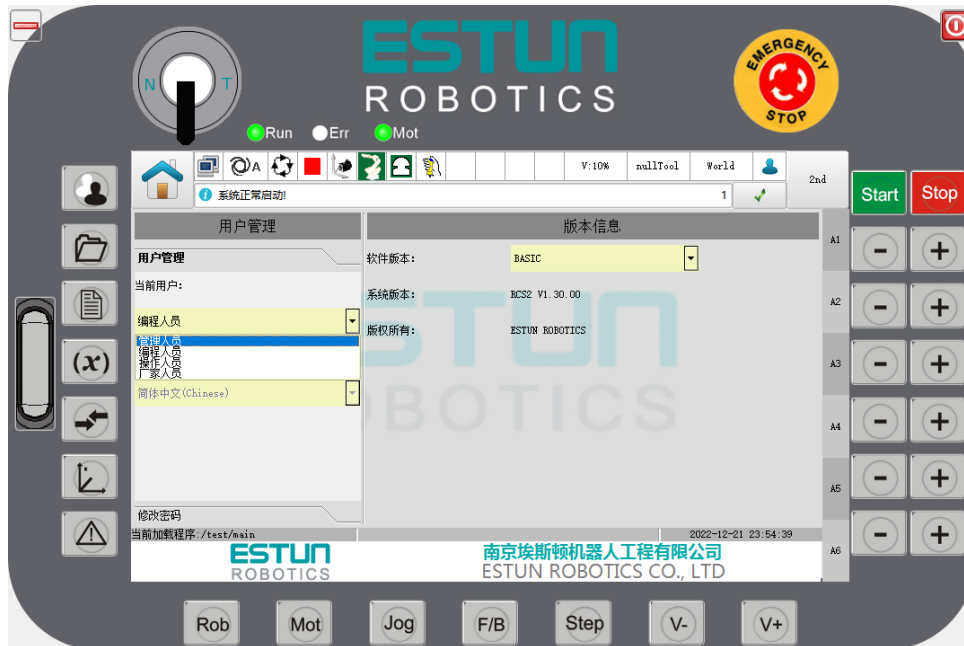
当前用户

1. 用户角色分为四种：“管理人员”、“编程人员”、“操作人员”、“厂家人员”，默认登录用户角色是“操作人员”。

用户角色	操作权限
操作人员	面向一线操作工、生产人员。 仅限于机器人手动移动，示教程序的启动、停止，系统报警状态的复位，查看运行信息。
编程人员	面向现场示教编程人员。 除操作人员权限外，可进行示教和编辑程序，示教程序的运行调试，系统应用参数的设定。
管理人员	面向系统维护人员。 除编程人员权限外，拥有对机器人系统的配置、管理权限：系统参数设定、用户口令管理等。
厂家人员	面向厂家技术人员。 拥有最高权限，可进行机器人的升级、维护。
备注： 手/自动模式用于机器人程序的示教和再现，远程模式指在生产状态下受外部 IO 信号、PLC 设备等控制。 上述操作权限是在未考虑系统模式前提下的通用概述，实际中同一角色对某功能的操作权限会受到系统模式的影响而存在一定区别。	

2. 不同用户角色对应的操作权限不同，查看到的版本信息内容的详细程度也不同，权限级别：“厂家人员>管理人员>编程人员>操作人员”具体权限可参考：[示教编程器权限分配表](#)。

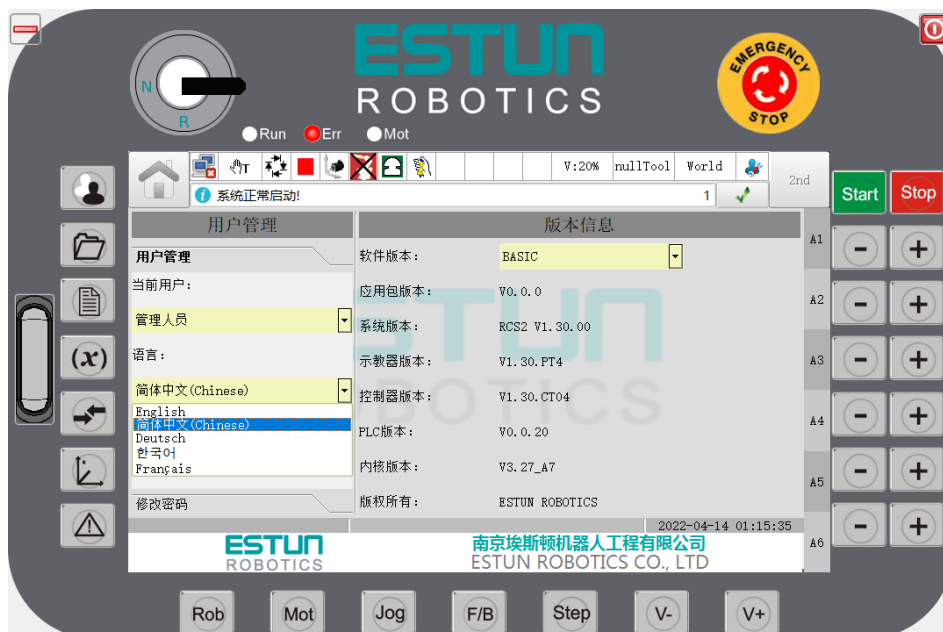
3. 切换用户的方法：点击下拉列表，选择需要的用户角色，部分用户角色之间的切换需要输入密码。



语言

系统默认的语言是“简体中文”，用户可在下拉菜单中选择切换其他系统语言。切换语言，重启系统后生效。

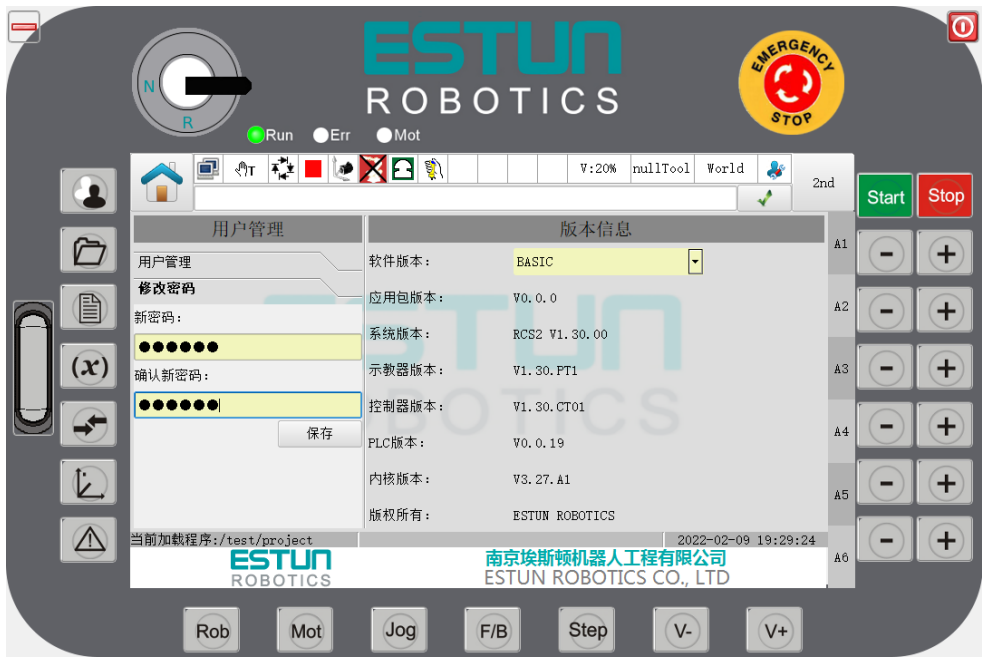
图4-54 切换语言界面



修改密码

用户可点击“修改密码”按钮，切换至“修改密码”页签，输入“新密码”和“确认新密码”（输入 2 次密码需要保持一致），点击“保存”按钮后，即可修改当前登录用户的密码。

图4-55 修改密码



4.7.2 通用设置



说明

1. 本章节主要介绍“通用设置”中的“基本设置”功能，其他功能如：“碰撞检测”、“振动抑制”、“视觉设置”、“伺服管理”、“从站配置”等功能介绍，详见第 7 章。
2. “通用设置”界面里的功能仅对“编程人员”和“管理人员”用户类型开放。若需要使用，请首先在“主页>用户登录”中切换用户。

基本设置

基本设置中可完成对本操作软件的一些基本设置、系统时间、程序设置、远程设置等基本操作。

步骤 1 在 Home 界面上点击“通用设置”，进入通用设置界面，如图 4-56 所示。

图4-56 通用设置界面



步骤 2 点击“基本设置”可进入“基本设置”界面，如图 4-57 所示。根据实际情况设置对应参数。

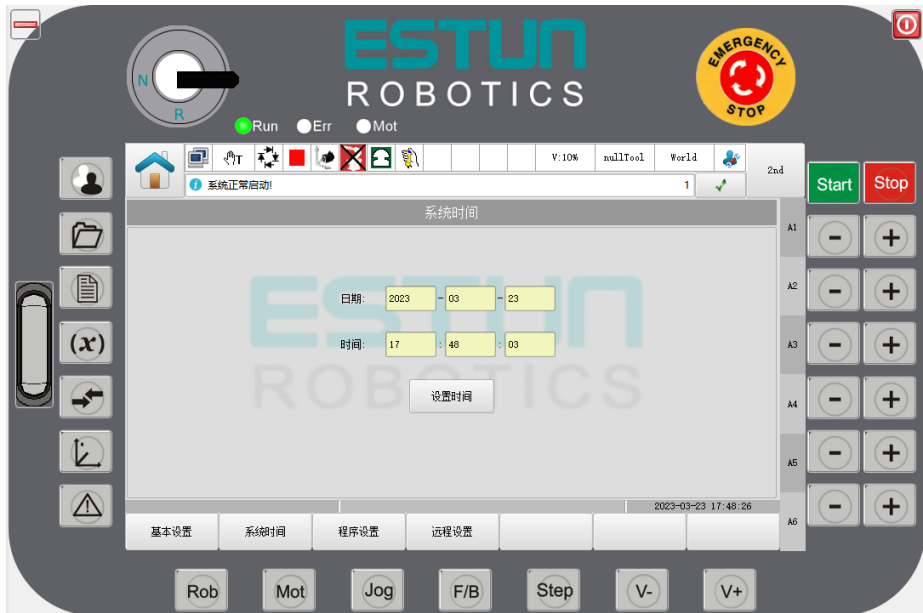
图4-57 基本设置



参数	参数介绍	操作方式
机器人类型	显示当前机器人类型	-
机器人安装类型	设置和显示当前机器人的安装方式。 取值范围：地面或吊装	1. 点击下拉菜单，选择需要的安装方式。 2. 根据提示“请重启系统”。
工件负载重量	显示系统中设置的有效工件重量。单位 Kg	-
工具负载重量	显示系统中设置的有效工具重量。单位 Kg	-

步骤 3 点击“系统时间”，进入“系统时间”设置页签，根据实际需求设置“日期”或“时间”。设置完成后，点击“设置时间”按钮，控制系统及示教器会自动重启，重启完成后，系统时间设置生效。

图4-58 基本设置-系统时间



步骤 4 点击“程序设置”，进入“程序设置”页签，根据实际需求设置相应参数。

图4-59 基本设置-程序设置



- 工具
 - 默认工具：选择开机默认加载的工具类型。默认值：nullTool。
 - 程序加载/注销时的复位：勾选，程序加载/注销后工具类型为默认值，去勾选，程序加载/注销后工具类型与复位前设置值一致。
- 坐标系
 - 默认坐标系：选择开机默认加载的坐标系类型。默认值：World。
 - 程序加载/注销时的复位：勾选，复位后坐标系类型为默认值，去勾选，程序加载/注销后坐标系类型与程序加载/注销前设置值一致。
- 负载惯量
 - 默认负载惯量：选择开机默认加载的负载惯量变量。默认值：FullPayload。
 - 程序加载/注销时的复位：勾选，程序加载/注销后负载惯量类型为默认值，去勾选，程序加载/注销后坐标系类型与复位前设置值一致。
- 指令格式

指令参数作用域显示使能：设置区分程序中系统变量、全局变量、工程变量和程序变量。

S：系统变量；G：全局变量；P：程序变量；L：程序变量。

例：程序中程序变量，勾选显示 **MovJ (L: P1, S: V100, "RELATIVE", S: C100)** 去勾选显示 **MovJ (P1, V100, C100)**。

说明

程序加载/注销时的复位使能，支持掉电保存功能，即修改该参数后断电重启后仍然为断电前的值。

步骤 5 点击“远程设置”，进入“远程设置”页面。

图4-60 基本设置-远程设置

参数	参数介绍	操作方式
远程速度	设置系统全局速度百分比值。	1. 点击输入框，在弹出软键盘中输入切换到远程模式时需要设置的系统全局速度百分比值
刷新	刷新设置“远程速度”的值	1. 设置完“远程速度”后，点击刷新
API 模式	外部控制（API）模式。 取值范围：ON 或 OFF	1. 点击下拉菜单，选择开启（ON）或关闭（OFF）外部控制（API）模式。
远程自动励磁	用来控制系统由自动模式切换至远程模式时是否自动上励磁的标识	1. 将该参数设置为 ON，系统自动模式，未励磁状态，切换至远程模式，自动上励磁。 2. 将该参数设置为 ON，系统自动模式，励磁状态，切换至远程模式，励磁状态保持。 3. 将该参数设置为 OFF，系统自动模式，未励磁状态，切换至远程模式，不会自动上励磁，等待外部励磁命令再上励磁。 4. 将该参数设置为 OFF，系统自动模式，励磁状态，切换至远程模式，掉励磁，等待外部励磁命令再上励磁。（需连接实际轴）
自启动程序	显示程序中是否设置为自启动	1. 若程序设置为自启动状态，则显示该程序目录

4.7.3 高级设置



说明

- 1. 本章节主要介绍“高级设置”中的“本地设置”功能，其他功能如：“网络设置”、“锁机管理”等功能介绍，详见第 7 章。
- 2. 高级设置里的功能仅对“管理人员”用户类型开放。若需要使用，请首先在“主页>用户登录”中切换用户。

本地设置

步骤 1 在 Home 界面上点击“高级设置”，进入高级设置界面，如图 4-61 所示。



步骤 2 在高级设置界面中点击“本地设置”可进入本地设置界面，如图 4-62 所示。



- 通用设置：
 - Rob 按键：勾选“重定义”，通过 Rob 按键不能进入快捷启动项。
 - 屏幕亮度：用户可设置屏幕亮度百分比，调节屏幕亮度（设置完成后，立即生效；重启示教器不改变已设置亮度）。
- 机器人（仅在虚拟示教器中显示）：

标识机器人的设置。仅在使用虚拟示教编程器时显示，以设置标识机器人的 ID。

- 示教器超时未操作（仅在实体示教器中显示）：默认选中熄屏并注销

功能名称	操作现象	唤醒方式
无操作	不会锁屏	-
仅熄屏	直接唤醒，进入当前界面	- 点击示教器屏幕 - 按示教器面板实体按键 - 控制器报警主动唤醒
熄屏并注销	- 唤醒后界面显示未熄屏前界面。 - 角色切换成“操作人员”。	
熄屏并上锁	- 显示当前角色，输入密码后，进入熄屏前界面。 - 可切换至其他用户，输入密码后，关闭锁屏。	
熄屏上锁并注销	- 角色切换成“游客”，点击锁屏界面上的“登录”。 - 可切换至其他用户，输入密码后，关闭锁屏。	

超时时间（分钟）：默认 5 分钟。设置范围 1-360（分钟）。

保存：设置完成后点击“保存”按钮，将参数写入文件中，并提示重启生效。

 说明

- 存在弹窗如软键盘、提示信息框等，不会触发超时事件。
 - 该功能只支持实体示教器，且熄屏功能仅支持 T76 版本示教器。
 - 超时未操作包括：未点击示教器屏幕，未按下示教器任意实体按键、未旋转钥匙开关等。
- 电源管理：

重启控制系统：

- 控制器重启，虚拟示教器则关闭虚拟示教器程序，实际示教器则重启示教器，点击“重启控制器系统”按钮后，需输入密码，密码校验成功，方可重启。

关闭控制系统：

- 控制器关机，虚拟示教器则关闭虚拟示教器程序，实际示教器则进入可选择界面：示教器维护模式等，点击“关闭控制系统”按钮后，需输入密码，密码校验成功，方可关闭。

关闭示教器：

- 点击“关闭示教器”，点击“关闭示教器”按钮后，需输入密码，密码校验成功后可关闭。

虚拟示教器：若当前使用的是虚拟示教，则关闭示教器程序

实际示教器：若当前使用的是实际示教器，则可进入选择界面：示教器维护模式

第 5 章 变量介绍

不同域支持不同变量类型，具体如下描述：

- 系统域：系统预定义变量，不可编辑。
- 全局域：IO 数据类型、PLC 数据类型、socket 数据类型、位置数据类型、区域数据类型、基本数据类型、摆动数据类型、时钟数据类型、码垛数据类型、系统数据类型。
- 工程域：IO 数据类型、socket 数据类型、位置数据类型、基本数据类型、摆动数据类型、码垛数据类型、系统数据类型。
- 程序域：IO 数据类型、socket 数据类型、位置数据类型、基本数据类型、摆动数据类型、系统数据类型。

5.1 基本数据类型

BOOL

存储 bool 型数据。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
value	bool	变量值
saveflag	bool	掉电保存标记

INT

存储整数。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
value	int	变量值
saveflag	bool	掉电保存标记

REAL

存储实数。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
value	real	变量值
saveflag	bool	掉电保存标记

STRING

存储字符串。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
value	string	变量值
saveflag	bool	掉电保存标记

BoolOneArray

存储 bool 型数组数据。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
count	int	数据个数,数值范围为 0~255
value	bool	变量值

IntOneArray

存储整数数组数据。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
count	int	数据个数,数值范围为 0~255
value	int	变量值

RealOneArray

存储实数数组数据。该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
count	int	数据个数,数值范围为 0~255
value	real	变量值



注意

1. 如果程序最后一条指令是对掉电保存变量的赋值指令，执行完后可能不会保存成功，需要在其后加个 wait 一段时间才可，同理单步 pc 到该赋值指令上运行后也有概率不会保存成功。
2. 数组变量中的数据个数过大时（如 count 为 200），请不要把该变量与开启了掉电保存功能的变量建立在同一个变量域中。如将开启了掉电保存属性的 INT 变量建立在全局域中，其后将一维数组变量建在工程域中。

5.2 位置数据类型

APOS – 关节位置类型

存储关节空间下各个轴的坐标值。该变量可支持用户自主创建以及修改，也可在示教程序中进行赋值运算等操作。同时该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
a1	real	关节 1 号轴的坐标。
a2 ... a15	real	关节 2 号轴的坐标。 ... 关节 15 号轴的坐标。
a16	real	关节 16 号轴的坐标。

【备注】机器人有几个轴就会显示几个轴的参数。如通用六轴机器人显示 a1~a6,scara 机器人显示为 a1~a4.

POSCFG

机器人在相同的笛卡尔空间位置下，可以具备多种关节位置组合（对应机器人逆解的多解）。该属性用于定义空间目标点对应的形态配置数据。

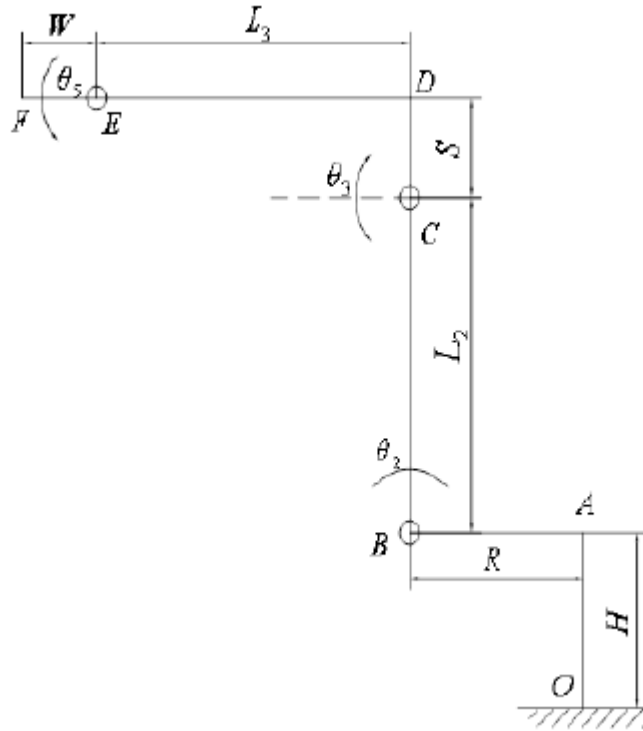
参数	数据类型	参数含义
mode	int	—
cf1	int	关节 1 号轴角度所在的象限取值。
cf2 ... cf5	int	关节 2 号轴角度所在的象限取值。 ... 关节 5 号轴角度所在的象限取值。
cf6	int	关节 6 号轴角度所在的象限取值。

这些象限取值划分如下：

非负数：0: $(-\pi, \pi]$, 1: $(\pi, 3\pi]$, 2: $(3\pi, 5\pi]$, ...; 每象限间相差 2π

负数：-1: $(-3\pi, -\pi]$, -2: $(-5\pi, -3\pi]$, ...; 每象限间相差 -2π

①、通用六关节运动学 mode 定义



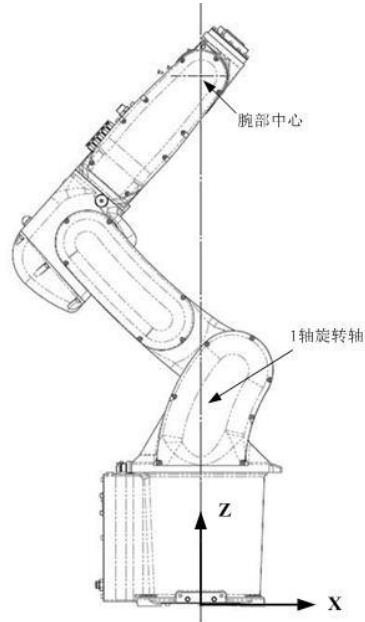
通用六关节的运动学存在八组解，定义 mode 值为 0~7，含义如下表：

Mode	腕部中心相对于一轴轴心的关系(flag1) 0: 在前; 1: 在后 $R + L_3 \cdot \cos(\theta_2 + \theta_3) + L_2 \cdot \sin \theta_2 + S \cdot \sin(\theta_2 + \theta_3)$	Axis3(flag3) $(\theta_3 + 90 - \arctan(S/L_3))$ 0: [0,180] 1:(-180,0)	Axis5(flag5) (θ_5) 0: [0,180] 1:(-180,0)
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

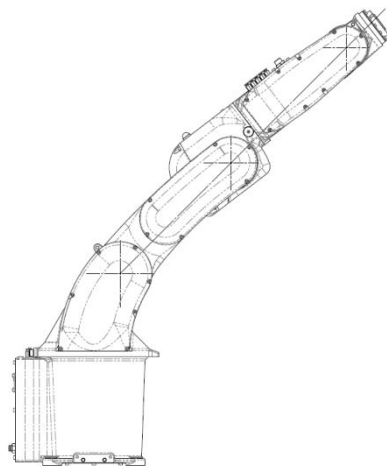
说明

- mode = flag5 + 2*flag3 + 4*flag1

aa、腕部中心相对于一轴轴心的关系如图，在前：指在图示 x 的正方向；在后：指在图示 x 的负方向。



bb、Axis3(flag3)的含义为：二三轴在一条直线上为边界条件，以前倾为 0，后仰为 1。



cc、Axis5(flag5)的含义：即五轴的角度。

②、SCARA 运动学 mode 定义

Scara 运动学存在两组解，定义 mode 值为 0 和 1，含义如下表

Mode	Axis2(flag2) (θ_2) 0:[0,180] 1: (-180,0)
0	0
1	1

说明

mode = flag2

③、三轴平面机器人运动学+ “3+1”机器人 mode 定义

Mode	Aixs2(flag2) ($\theta_2 + 90$) 0: [0,180] 1: (-180,0)
0	0
1	1



说明

mode = flag2

④、吊装 SCARA 运动学 mode 定义

吊装 Scara 运动学存在两组解，定义 mode 值为 0 和 1，含义如下表

Mode	Axis2(flag2) ($\theta_2 - 180$) 0:[0,180] 1: (-180,0)
0	0
1	1

CPOS – 坐标位置类型

存储 TCP 点在笛卡尔坐标系下位置。该变量可支持用户自主创建以及修改，也可在示教程序中进行赋值运算等操作。同时该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数列表	值
confdata	
mode	0
cf1	0
cf2	0
cf3	0
cf4	0
cf5	0
cf6	0
x	0.0000000
y	0.0000000
z	0.0000000

参数	数据类型	参数含义
confdata	POSCFG	空间目标点对应的形态配置信息。
x	real	TCP 点在参考坐标系上 x 方向的坐标。
y	real	TCP 点在参考坐标系上 y 方向的坐标。
z	real	TCP 点在参考坐标系上 z 方向的坐标。
a	real	TCP 点相对于参考坐标系 z 轴旋转的欧拉角。
b	real	TCP 点相对于参考坐标系 y 轴旋转的欧拉角。
c	real	TCP 点相对于参考坐标系 x 轴旋转的欧拉角。
a7	real	关节 7 号轴的坐标。

参数	数据类型	参数含义
a8 ... a15	real	关节 8 号轴的坐标。 ... 关节 15 号轴的坐标。
a16	real	关节 16 号轴的坐标。

【备注】机器人有几个轴就会显示几个轴的参数。

DAPOS – 关节相对位置类型

存储关节空间下各个轴的相对偏移量。该变量可支持用户自主创建以及修改，也可在示教程序中进行赋值运算等操作。同时该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
da1	real	关节 1 号轴角度的偏移量。
da2 ... da15	real	关节 2 号轴角度的偏移量。 ... 关节 15 号轴角度的偏移量。
da16	real	关节 16 号轴角度的偏移量。

【备注】机器人有几个轴就会显示几个轴的参数。如六轴机器人显示 da1~da6，scara 机器人显示为 da1~da4。

DCPOS 坐标系相对位置类型

存储 TCP 点在笛卡尔坐标系下的相对偏移量。该变量可支持用户自主创建以及修改，也可在示教程序中进行赋值运算等操作。同时该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
dx	real	TCP 点在参考坐标系上 x 方向的位移偏移量。
dy	real	TCP 点在参考坐标系上 y 方向的位移偏移量。
dz	real	TCP 点在参考坐标系上 z 方向的位移偏移量。
da	real	TCP 点相对参考坐标系 z 轴旋转的欧拉角偏移量。
db	real	TCP 点相对参考坐标系 y' 轴旋转的欧拉角偏移量。
dc	real	TCP 点相对参考坐标系 x'' 轴旋转的欧拉角偏移量。
da7	real	关节 7 号轴角度偏移量。
da8 ... da15	real	关节 8 号轴角度偏移量。 ... 关节 15 号轴角度偏移量。
da16	real	关节 16 号轴角度偏移量。

【备注】机器人有几个轴就会显示几个轴的参数。

APOSARRAY – 关节位置数组

APOS 的数组形式。存储关节空间下各个轴的坐标值。该变量可支持用户自主创建以及修改，也可在示教程序中进行赋值运算等操作。同时该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作。

参数	参数	数据类型	参数含义
APOSARRAY0[X]	APOSARRAY0[1] ... APOSARRAY0[1000]	APOS	存储的单个 APOS 值

CPOSARRAY – 坐标位置数组

CPOS 的数组形式。存储 TCP 点在笛卡尔坐标系下位置。该变量可支持用户自主创建以及修改，也可在示教程序中进行赋值运算等操作。同时该变量支持在全局，工程，程序三大变量作用域中创建，删除，修改等操作（使用中的数组变量不能删除和修改）。

参数	参数	数据类型	参数含义
CPOSARRAY0[X]	CPOSARRAY0[X] ... CPOSARRAY0[1000]	CPOS	存储的单个 CPOS 值

5.3 系统数据类型

LsScale

LsScale 型变量用来记录各个关节轴的增益比例阈值参数，用来改善机器人在某一速度区间段内的低速抖动现象，与速度区间阈值参数一起搭配使用。设置范围：[100,1000]，单位：%。该变量只允许在全局域创建以及修改。

参数	数据类型	参数含义
J1	int	J1 轴对应的增益比例阈值，单位：%
J2	int	J2 轴对应的增益比例阈值，单位：%
J3	int	J3 轴对应的增益比例阈值，单位：%
J4	int	J4 轴对应的增益比例阈值，单位：%
J5	int	J5 轴对应的增益比例阈值，单位：%
J6	int	J6 轴对应的增益比例阈值，单位：%

LsThresh

LsThresh 型变量用来记录各个关节轴的速度区间阈值参数，用来改善机器人在某一速度区间段内的低速抖动现象，与增益比例阈值参数一起搭配使用。设置范围：[10,1000]，单位：r/min。该变量只允许在全局域创建以及修改。

参数	数据类型	参数含义
J1	int	J1 轴对应的速度区间阈值，单位：r/min
J2	int	J2 轴对应的速度区间阈值，单位：r/min
J3	int	J3 轴对应的速度区间阈值，单位：r/min
J4	int	J4 轴对应的速度区间阈值，单位：r/min
J5	int	J5 轴对应的速度区间阈值，单位：r/min
J6	int	J6 轴对应的速度区间阈值，单位：r/min

SYNCOORD

SYNCOORD 型变量用来记录跟随时运动目标物体的基准参考坐标系参数，定义其相对于世界坐标系的位移和旋转。该参数可由用户根据变量界面标定计算所得，也可由用户自行输入。

参数	数据类型	参数含义
id	int	坐标系的唯一索引编号，该值不可被修改，创建时由系统自动分配。
x	real	坐标系原点相对于世界坐标系在 x 方向的位移偏移量，单位是 mm。
y	real	坐标系原点相对于世界坐标系在 y 方向的位移偏移量，单位是 mm。
z	real	坐标系原点相对于世界坐标系在 z 方向的位移偏移量，单位是 mm。

参数	数据类型	参数含义
a	real	坐标系相对于世界坐标系 z 轴旋转的欧拉角，单位是 deg。
b	real	坐标系相对于世界坐标系 y' 轴旋转的欧拉角，单位是 deg。
c	real	坐标系相对于世界坐标系 x'' 轴旋转的欧拉角，单位是 deg。

POSITIONER

POSITIONER 型变量通过 index 来选择变位机 (ID)。

参数	数据类型	参数含义
index	int	index 取值范围为[0, 6]。其中： • 0：表示该变量无效。 • 1：表示选择变位机 ID=1 的坐标系。 • 2：表示选择变位机 ID=2 的坐标系。 • ... • 6：表示选择变位机 ID=6 的坐标系。



说明

变位机坐标系的标定请参见“7.22 变位机功能”。

EXTTCP

EXTTCP 型变量用来记录外部工具坐标系参数，定义外部工具坐标系相对于世界坐标系下的位移和旋转。该参数可由用户根据变量界面标定计算所得，也可由用户自行输入。该变量只允许在全局域创建以及修改。

参数	数据类型	参数含义
id	int	外部工具坐标系的唯一索引编号，该值不可被修改，创建时由系统自动分配。
x	real	外部工具坐标系原点相对于世界坐标系在 x 方向的位移偏移量，单位是 mm。
y	real	外部工具坐标系原点相对于世界坐标系在 y 方向的位移偏移量，单位是 mm。
z	real	外部工具坐标系原点相对于世界坐标系在 z 方向的位移偏移量，单位是 mm。
a	real	外部工具坐标系相对于世界坐标系 z 轴旋转的欧拉角，单位是 deg。
b	real	外部工具坐标系相对于世界坐标系 y' 轴旋转的欧拉角，单位是 deg。
c	real	外部工具坐标系相对于世界坐标系 x'' 轴旋转的欧拉角，单位是 deg。

TOOL

TOOL 型变量用来记录工具参数，定义工具末端相对机器人法兰盘位移和旋转。该参数可由用户根据变量界面标定计算所得，也可由用户自行输入。该变量只允许在全局域创建以及修改。

参数	数据类型	参数含义
id	int	工具坐标系的唯一索引编号，该值不可被修改，创建时由系统自动分配。
x	real	TCP 相对于法兰坐标系在 x 方向的位移偏移量，单位是 mm。
y	real	TCP 相对于法兰坐标系在 y 方向的位移偏移量，单位是 mm。
z	real	TCP 相对于法兰坐标系在 z 方向的位移偏移量，单位是 mm。
a	real	TCP 相对于法兰坐标系 z 轴旋转的欧拉角，单位是 deg。
b	real	TCP 相对于法兰坐标系 y' 轴旋转的欧拉角，单位是 deg。
c	real	TCP 相对于法兰坐标系 x'' 轴旋转的欧拉角，单位是 deg。
dyn	LoadDyn	工具的质量信息，用于机器人动力学全模型计算。

系统预定义了 nullTool 这个工具变量，其中各个偏移量的值默认都为 0，用户可根据现场需求自行定义需要使用的工具坐标系。

USERCOOR

usercoor 型变量用来记录用户坐标系参数，定义用户坐标系相对于世界坐标系的位移和旋转。该参数可由用户根据变量界面标定计算所得，也可由用户自行输入。该变量只允许在全局域创建以及修改。

参数	数据类型	参数含义
id	int	用户坐标系的唯一索引编号，该值不可被修改，创建时由系统自动分配。
x	real	用户坐标系原点相对于世界坐标系在 x 方向的位移偏移量，单位是 mm。
y	real	用户坐标系原点相对于世界坐标系在 y 方向的位移偏移量，单位是 mm。
z	real	用户坐标系原点相对于世界坐标系在 z 方向的位移偏移量，单位是 mm。
a	real	用户坐标系相对于世界坐标系 z 轴旋转的欧拉角，单位是 deg。
b	real	用户坐标系相对于世界坐标系 y' 轴旋转的欧拉角，单位是 deg。
c	real	用户坐标系相对于世界坐标系 x'' 轴旋转的欧拉角，单位是 deg。

系统预定义了世界坐标系 World，其中各个偏移量的值默认都为 0，用户可根据现场需求自行定义需要使用的用户坐标系。

PAYLOAD

PAYLOAD 型变量用来记录工件负载参数，定义机器人末端负载参数以及质量信息，使用详细的质量信息有助于机器人动力学全模型计算，以便更高的提高运动节拍。该参数目前只支持用户提前算好值后自行输入，无示教编程器标定计算操作。该变量只允许在全局域创建以及修改。

参数	数据类型	参数含义
id	int	工件负载的唯一索引编号，该值不可被修改，创建时由系统自动分配。
dyn	LoadDyn	工件负载的质量信息，用于机器人动力学全模型计算。

系统预定义了 PAYLOAD0 这个工件负载，其中各个偏移量的值默认都为 0，用户可根据现场需求自行定义需要使用的工件负载参数。

ZONE

用于定义某一运动如何结束或者定义两条运动轨迹之间转弯区的大小。为了方便用户使用，系统预设了常用的过渡变量（系统变量，不允许用户修改），同时可支持用户自在全局，工程，程序三大变量作用域中对该变量进行创建，删除，修改等操作。

参数	数据类型	参数含义
per	real	转弯百分比。适用于 MovJ,MovL,MovC 等运动指令，表示距离目标点还有多远时开始转弯。
dis	real	笛卡尔空间转弯区大小。用于 MovL,MovC 等直线圆弧运动指令，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 dis 毫米的地方时，开始转向下一个目标点运动，单位是 mm。
vConst	int	恒速过渡使能状态。过渡段线速度是否恒定不变的使能状态。为 1 表示过渡恒定不变，为 0 表示跟随过渡参数变化。 说明： 该参数一般在线速度不高的情况下使用，具体数值根据机器人类型会发生变化。

SPEED

用来定义机器人和外部轴的运动速度。为了方便用户使用，系统预设了常用的速度变量（系统变量，不允许用户修改），同时可支持用户自在全局，工程，程序三大变量作用域中对该变量进行创建，删除，修改等操作。

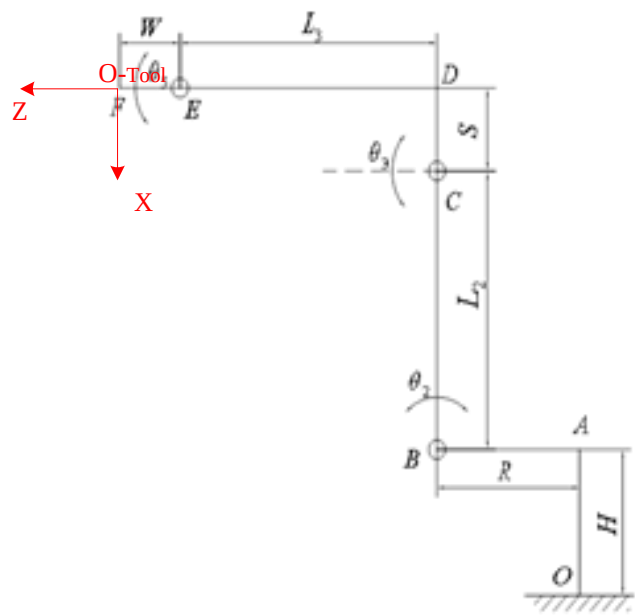
参数	数据类型	参数含义
per	real	关节速度百分比。用于指定关节运动指令时的运动速度，适用于 MovJ, MovJRel, MovJSearch 指令，取值范围 1%~100%。
tcp	real	TCP 线速度。定义机器人末端点的线速度，用于 MovL, MovC 等直线圆弧运动指令。
ori	real	空间旋转速度。定义机器人末端点姿态的旋转速度。
exj_l	real	外部轴线速度。定义外部直线轴的运动速度。
exj_r	real	外部轴角速度。定义外部旋转轴的运动速度。

CenterPos

质心矢量是以安装的工具或负载在坐标系 O_{Tool}-XYZ 上的位置设定，如下所述。

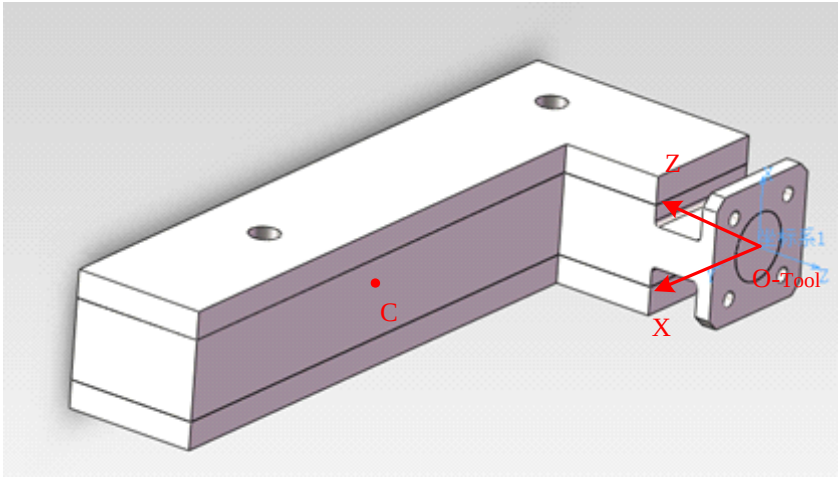
工具&负载质量信息参考坐标系方向如图 5-1 所示，工具&负载质量信息坐标系定义实例如图 5-2 所示。

图5-1 工具&负载质量信息坐标系说明



坐标系 O_{Tool} -XYZ 位于机器人末端法兰平面，该坐标系 Z 轴轴向与 $J6$ 关节（以 6 轴机器人为例）轴向同轴。

图5-2 工具&负载质量信息坐标系实例



【注】工具和负载的质量信息输出坐标系均参考图 5-1 中坐标系 O_{Tool} -XYZ。

参数	数据类型	参数含义
Mx	real	安装的工具或装夹的负载的重心 C 在坐标系 O_{Tool} -XYZ 的 X 方向上的偏移量，单位是 mm。
My	real	安装的工具或装夹的负载的重心 C 在坐标系 O_{Tool} -XYZ 的 Y 方向上的偏移量，单位是 mm。
Mz	real	安装的工具或装夹的负载的重心 C 在坐标系 O_{Tool} -XYZ 的 Z 方向上的偏移量，单位是 mm。

InertiaTensor

该参数由安装的工件或负载的重心决定，并且对齐输出的坐标系 O_{tool} -XYZ。输出坐标系请参考图 5-2 的实例。

$$\text{惯性张量矩阵表示形式: } {}^0I = \begin{bmatrix} L_{xx} & -L_{xy} & -L_{xz} \\ -L_{xy} & L_{yy} & -L_{yz} \\ -L_{xz} & -L_{yz} & L_{zz} \end{bmatrix}$$

参数	数据类型	参数含义
Lxx	real	安装的工具或装夹的负载在重心处 X 方向回转的惯量矩，单位是 $\text{kg}\cdot\text{mm}^2$ 。
Lyy	real	安装的工具或装夹的负载在重心处 Y 方向回转的惯量矩，单位是 $\text{kg}\cdot\text{mm}^2$ 。
Lzz	real	安装的工具或装夹的负载在重心处 Z 方向回转的惯量矩，单位是 $\text{kg}\cdot\text{mm}^2$ 。
Lxy	real	安装的工具或装夹的负载在重心处 XY 交叉方向的惯量积，单位是 $\text{kg}\cdot\text{mm}^2$ 。
Lxz	real	安装的工具或装夹的负载在重心处 XZ 交叉方向的惯量积，单位是 $\text{kg}\cdot\text{mm}^2$ 。
Lyz	real	安装的工具或装夹的负载在重心处 YZ 交叉方向的惯量积，单位是 $\text{kg}\cdot\text{mm}^2$ 。

LoadDyn

用来存储机器人末端工具和负载质量信息参数，用于机器人动力学全模型计算。

参数	数据类型	参数含义
M	real	工具&负载的重量，单位是 kg。
pos	CenterPos	详见 CenterPos 含义。
tensor	InertiaTensor	详见 InertiaTensor 含义。

INTERUPT

存储 INTERRUPT 名称信息，作为中断标识的唯一索引，该变量仅支持全局变量作用域中的创建，删除、修改等操作。

参数	数据类型	参数含义
value	string	存储中断标识的名称信息。

COLLIPARAM

COLLIPARAM 型变量通过 index 来选择对应的碰撞自调整参数（ID）。

参数	数据类型	参数含义
index	int	取值范围为[0, 6]。其中 0 表示该变量无效，其它表示 ID 为 index 的碰撞自调整参数。

5.4 IO 数据类型

DI

数字量输入变量。

参数	数据类型	参数含义
port	int	与该变量绑定的数字量输入端口号。
value	int	取值范围为 0 或 1，代表对应端口输入状态。
riseSts	bool	该端口上升沿信号状态。
downSts	bool	该端口下降沿状态。

DO

数字量输出变量。

参数	数据类型	参数含义
port	int	与该变量绑定的数字量输出端口号。
value	int	取值范围为 0 或 1，代表对应端口输出状态。

AI

模拟量输入变量。

参数	数据类型	参数含义
port	int	与该变量绑定的模拟量输入端口号。
value	real	取值范围为-10~10V，代表对应端口输入电压值。

AO

模拟量输出变量。

参数	数据类型	参数含义
port	int	与该变量绑定的模拟量输出端口号。
value	real	取值范围为-10~10V，代表对应端口输出电压值。

SimDI

虚拟数字量输入变量。

参数	数据类型	参数含义
port	int	与该变量绑定的虚拟数字量输入端口号。

参数	数据类型	参数含义
value	int	取值范围为 0 或 1，代表对应端口输入状态。
riseSts	bool	该端口上升沿信号状态。
downSts	bool	该端口下降沿状态。

SimDO

虚拟数字量输出变量。

参数	数据类型	参数含义
port	int	与该变量绑定的虚拟数字量输出端口号。
value	int	取值范围为 0 或 1，代表对应端口输出状态。

SimAI

虚拟模拟量输入变量。

参数	数据类型	参数含义
port	int	与该变量绑定的虚拟模拟量输入端口号。
value	real	代表对应端口输入值。

SimAO

虚拟模拟量输出变量。

参数	数据类型	参数含义
port	int	与该变量绑定的虚拟模拟量输出端口号。
value	real	代表对应端口输出值。

5.5 摆动数据类型

WEAVE

该变量用于存储机器人走直线/圆弧轨迹摆动时的一些配置参数。同时可支持用户在全局，工程，程序三大变量作用域中对该变量进行创建，删除，修改等操作。

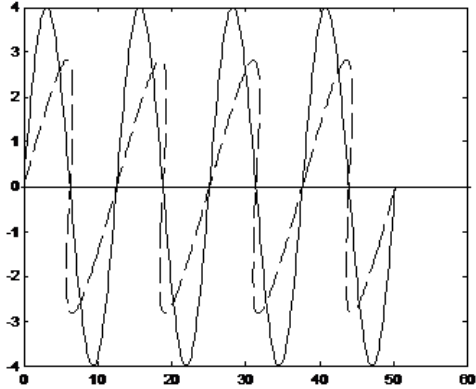
参数	数据类型	参数含义
Type	int	机器人摆动类型，该值为 1 表示正弦摆，该值为 2 表示三角摆。
Freq	real	机器人摆动时的频率，单位：Hz。
Amp_L	real	机器人摆动左幅值，单位：mm。

参数	数据类型	参数含义
Amp_R	real	机器人摆动右幅值，单位：mm。
StopTime_L	int	波峰处停止时间，指在摆动到左幅值时停止的时间，单位是 ms。
StopTime_C	int	中间位置停止时间，指在摆动到中间位置时摆动停止的时间，单位是 ms。
StopTime_R	int	波谷处停止时间，指在摆动到右幅值时停止的时间，单位是 ms。
RotAngle_X	real	摆动基准平面绕摆动行进方向的旋转角度，该参数用于决定最终的摆动平面，单位是 deg。
RotAngle_Z	real	绕摆动平面法向量的旋转角度，决定摆弧的形状，单位：deg



注意

- 由行进路线与工具坐标系的 Tz 方向确定的平面的法向量，与行进路线构成的平面为摆动基准平面。
- RotAngle_Z 的作用，如下图所示：RotAngle_Z=0 为实线，不为 0 为虚线所示。



5.6 时钟数据类型

CLOCK

存储时钟信息的值。该变量仅支持在全局作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
id	int	时钟变量的编号。
state	real	时钟变量的使能状态。
value	real	时钟变量的计数值。

5.7 区域数据类型

AREA

存储标准区域信息的值。该变量仅支持在全局作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
id	int	标准区域变量的编号。
activate	bool	是否激活此标准区域以及此标准区域是否在激活状态。
isInArea	bool	是否在区域内。
initActivate	bool	初始时自动激活标志。
areaType	int	区域类型。
areaShape	int	区域形状。
refSysScope	int	参考坐标系域。
refSysName	string	参考坐标系名字。
enInPut	bool	输入使能。
enInHigh	bool	输入使能高有效。
ditype	int	DI 类型。
InputActStatePort	int	输入实际状态。
enOutPut	bool	输出使能。
enOutHigh	bool	输出使能高有效。
doType	int	DO 类型。
OutputActStatePort	int	输出实际状态。
StartPointX	real	区域起点的坐标 x。
StartPointY	real	区域起点的坐标 y。
StartPointZ	real	区域起点的坐标 z。
lenXR	real	矩形区域的长度或者圆柱形区域的半径。
lenYH	real	矩形区域的宽度或者圆柱形区域的高度。
lenZ	real	矩形区域的高度。
isFlangeEnd	real	区域监控中用于监控 TCP 或法兰端

POLYHEDRON

POLYHEDRON 型变量通过 index 来选择多边形区域（ID）。该变量仅支持在全局作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
index	int	index 取值范围为[0, 4]。其中： • 0：表示该变量无效。 • 1：表示选择区域 ID=1 的多边体区域。 • 2：表示选择区域 ID=2 的多边体区域。 • ...
activate	bool	此多边体区域是否在激活状态。
isInArea	bool	是否在多边体区域内。

5.8 PLC 数据类型

PLCREAL

存储 PLC 实数,根据数据编号与 plc 端的 real 变量相对应。该变量仅支持在全局变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
plcNum	int	数据编号
value	real	变量值
saveflag	bool	掉电保存标记

PLCBOOL

存储 PLC 布尔值，根据数据编号与 plc 端的 bool 变量相对应。该变量仅支持在全局变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
plcNum	int	数据编号
value	bool	变量值
saveflag	bool	掉电保存标记

PLCINT

存储 PLC 短整型数据，根据数据编号与 plc 端的 int 变量相对应。该变量仅支持在全局变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
plcNum	int	数据编号
value	int	变量值
saveflag	bool	掉电保存标记

PLCDINT

存储 PLC 整型数据，根据数据编号与 plc 端的 dint 变量相对应。该变量仅支持在全局变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
plcNum	int	数据编号
value	int	变量值
saveflag	bool	掉电保存标记

5.9 码垛数据类型

PLACEOPTIONS

垛体变量中前置点、后置点包含的信息。该数据类型仅能嵌套在 PALLET 数据类型中一起使用。

参数	数据类型	参数含义
isUse	bool	参数是否有效。
sideOffset	real	水平偏移。
height	real	高度偏移。

PALLET

存储垛体信息。该变量可支持用户自主创建以及修改。该变量支持在全局，工程变量作用域中创建，删除，修改等操作。

参数	数据类型	参数含义
actParts	int	垛体中实际工件数。
maxParts	int	垛体中最大工件数。
isEmpty	bool	垛体是否为空。
isFull	bool	垛体是否为满。
xNum	int	垛体 x 方向上工件数。
yNum	int	垛体 y 方向上工件数。
zNum	int	垛体 z 方向上工件数。
xdistOfPart	real	垛体 x 方向上单个工件长度。
ydistOfPart	real	垛体 y 方向上单个工件长度。
zdistOfPart	real	垛体 z 方向上单个工件长度。
refSysScope	int	垛体参数所参考坐标系域。
refSysName	string	垛体参数所参考坐标系名称。

参数	数据类型	参数含义
palletDir	int	码垛或拆垛方向。
palletOrder	int	码垛或拆垛顺序。
firstPartScope	int	第一个工件位置变量作用域。
firstPartPos	string	第一个工件位置变量名称。
isEntryPosUsed	bool	是否行至码垛入口点。
palletEntryScope	int	码垛入口点位置变量作用域。
palletEntryPos	string	码垛入口点位置点名称。
preplaceOptions	PLACEOPTIONS	前置点信息。
postplaceOptions	PLACEOPTIONS	后置点信息。

5.10 SOCKET 数据类型

Socket

存储 Socket 名称信息。该变量可支持用户自主创建以及修改。该变量支持在全局，工程、程序变量作用域中创建，删除，修改等操作。

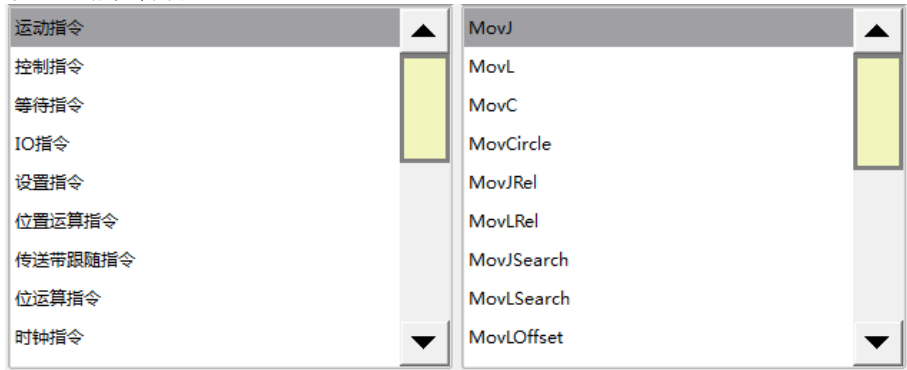
参数	数据类型	参数含义
value	string	socket 名称。

第 6 章 指令介绍

6.1 运动指令

运动指令列表如图 6-1 所示。

图6-1 运动指令列表



MovJ - 点到点的运动指令

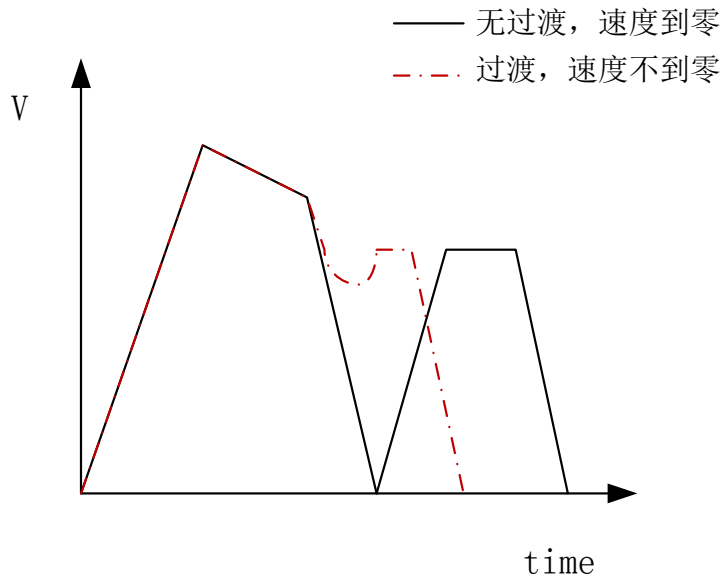
该指令表示机器人各个关节进行点到点的运动（point to point），机器人的末端轨迹为不规则的曲线，并且在
该指令运行结束时可进行 IO 指令的操作，如图 6-2 所示。

图6-2 MovJ 参数编辑界面

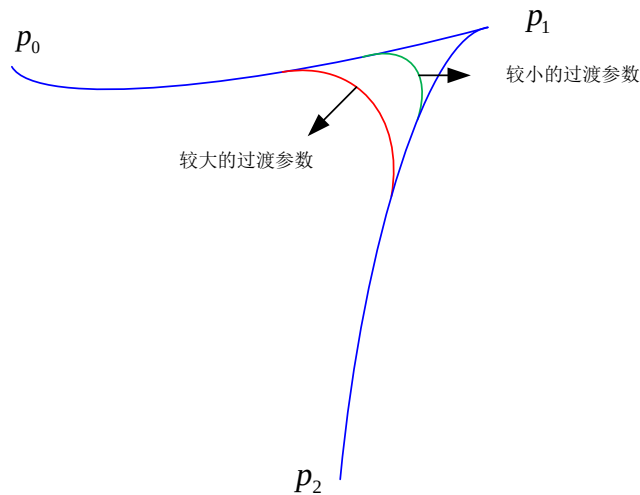
目标位置	A L: P1
目标速度	S: V100
过渡类型	"RELATIVE"
过渡值	S: C0
工具参数	DEFAULT
坐标系参数	DEFAULT
工件负载	DEFAULT

- 目标位置：指令终点位置。
 - 设置为“APOS”时，关节空间下各个轴的坐标值。
 - 设置为“CPOS”时，TCP 点在笛卡尔坐标系下位置。目标位置取决于 confdata 的配置，若 mode 参数设置值为-1，则重点位置不受限于 confdata 的值，而是已最优解的方式到达目的地。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为“SPEED”类型变量时，可以选用系统预定义，也可以自行创建；其中，per 项对 MovJ 指令起调控作用，即为全局速度的百分比。
- 过渡类型：机器人逼近终点时的过渡方式。

- DEFAULT: 缺省, 即指令中不设置过渡参数。若程序中使用了全局过渡参数 (SetBlendParam), 过渡方式按照全局设置的过渡参数运行。
- FINE: 效果等同于 DEFAULT。
- RELATIVE: 相对过渡, 过渡参数为相对值, 表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。



- ABSOLUTE: 绝对过渡, 过渡参数是个绝对值, 表示过渡点到轨迹终点的距离。



MovL - 直线运动指令

MovL 指令为直线运动指令, 通过该指令可以使机器人 TCP 点以设定的速度直线运动到目标位置, 若运动的起止姿态不同, 则运行过程中姿态随位置同步地旋转到终点的姿态。与 MovJ 相同, 在执行完该指令时可以附加进行 IO 操作, MovL 的参数设置与 MovJ 相似, 不同之处为目标速度为绝对速度值, 不是百分比。

图6-3 MovL 参数编辑界面

目标位置	 L: P0	▼ ▲
目标速度	DEFAULT	▼
过渡类型	"RELATIVE"	▼
过渡值	DEFAULT	▼
工具参数	DEFAULT	▼
坐标系参数	DEFAULT	▼
工件负载	DEFAULT	▼
速率类型	DEFAULT	▼
AddDo		▼ ▲

- 目标位置：指令终点位置，类型可为 APOS 或 CPOS。
在下拉列表中选择已有的点或在弹出的对话框中新建并示教一个变量。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建；其中，tcp、ori 分量有效。有外部轴时，SPEED 变量中的 exj_l、exj_r 分量有效。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工具参数。
 - 设置为 TOOL 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工具参数不同的坐标系时，系统会将工具参数切换成设置的工具参数。若本段与前后段轨迹的工具参数变化，则不支持过渡。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的坐标系。
 - 设置为 USERCOORD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换成设置的坐标系。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 速率类型：机器人执行该轨迹时指令速度是否受示教编程器全局倍率影响。
 - DEFAULT：同 GOVRON，即指令速度受示教编程器全局倍率影响。
 - GOVRON：指令速度受示教编程器全局倍率影响。
 - GOVROFF：指令速度不受示教编程器全局倍率影响。

- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。
 - IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetSimDO8421、SetDIEdge、SetSimDIEdge。

MovC - 弧形运动指令

圆弧指令指机器人 TCP 点从起始位置，经过中间位置到目标位置做圆弧运动，若运动的起止姿态不同，则运行过程中姿态随位置同步地旋转到终点的姿态，但不一定经过中间位置的姿态。

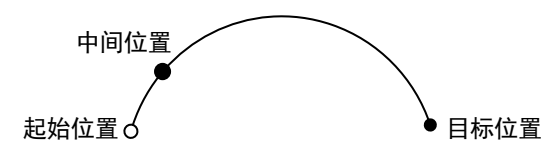
图6-4 MovC 参数编辑界面

中间位置	 L: P0	
目标位置	 L: P1	
目标速度	DEFAULT	
过渡类型	"RELATIVE"	
过渡值	DEFAULT	
工具参数	DEFAULT	
坐标系参数	DEFAULT	
工件负载	DEFAULT	
速率类型	DEFAULT	
AddDo		

- 中间位置：圆弧中间辅助点位置。
- 目标位置：圆弧终点位置。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工具参数。
 - 设置为 TOOL 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工具参数不同的坐标系时，系统会将工具参数切换成设置的工具参数。若本段与前后段轨迹的工具参数变化,则不支持过渡。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的坐标系。
 - 设置为 USERCOOR 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换成设置的坐标系。

- 设置为外部工具坐标系时，机器人的运动相对于外部工具坐标系，指令的目标速度相对于外部工具坐标系。外部 TCP 功能只支持参考坐标系为相同外部工具坐标系时的过渡。
- 设置为变位机坐标系时，机器人的运动相对于变位机坐标系，指令的目标速度相对于变位机法兰。变位机功能只支持同一个变位机之间的过渡。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 速率类型：机器人执行该轨迹时指令速度是否受示教编程器全局倍率影响。
 - DEFAULT：同 GOVRON，即指令速度受示教编程器全局倍率影响。
 - GOVRON：指令速度受示教编程器全局倍率影响。
 - GOVROFF：指令速度不受示教编程器全局倍率影响。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。
 - IO 指令：执行 IO 操作。

图6-5 MovC 圆弧关键点示意



该指令必须遵循以下规定：
机器人 TCP 末端做整圆运动，必须执行两条 MovC 指令或使用 MovCircle 指令。

MovCircle - 圆形运动指令

整圆指令指机器人 TCP 点从起始位置，经过辅助点 1 与辅助点 2 位置到起始位置做整圆运动，整圆运动过程中姿态保持不变。

图6-6 MovCircle 参数编辑界面

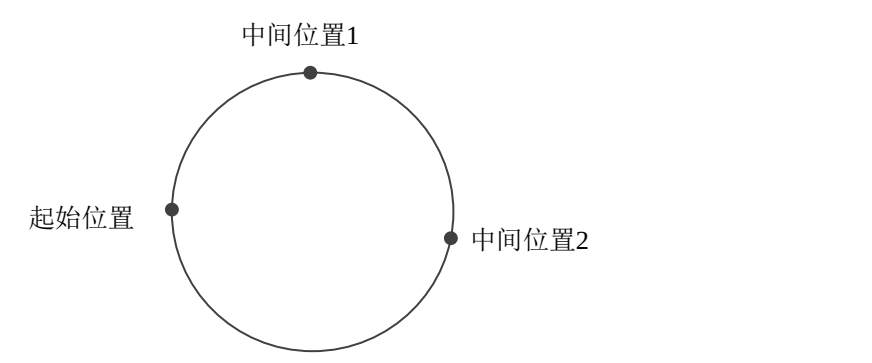
中间位置1	 L: P3	
中间位置2	 L: P4	
目标速度	DEFAULT	
过渡类型	DEFAULT	
工具参数	DEFAULT	
坐标系参数	DEFAULT	
工件负载	DEFAULT	
速率类型	DEFAULT	
AddDo		

- 中间位置 1：整圆中间辅助点位置 1，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运行。
- 中间位置 2：整圆中间辅助点位置 2，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运行。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。

- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工具参数。
 - 设置为“TOOL”类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工具参数不同的坐标系时，系统会将工具参数切换到设置的工具参数。若本段与前后段轨迹的工具参数变化，则不支持过渡。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的坐标系。
 - 设置为“USERCOORD”类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换到设置的坐标系。
 - 设置为“外部工具坐标系”时，机器人的运动相对于外部工具坐标系，指令的目标速度相对于外部工具坐标系。外部 TCP 功能只支持参考坐标系为相同外部工具坐标系时的过渡。
 - 设置为“变位机坐标系”时，机器人的运动相对于变位机坐标系，指令的目标速度相对于变位机法兰。变位机功能只支持同一个变位机之间的过渡。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为“PAYLOAD”类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换到设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 速率类型：机器人执行该轨迹时指令速度是否受示教编程器全局倍率影响。
 - DEFAULT：同 GOVRON，即指令速度受示教编程器全局倍率影响。
 - GOVRON：指令速度受示教编程器全局倍率影响。
 - GOVROFF：指令速度不受示教编程器全局倍率影响。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。

IO 指令：执行 IO 操作。

图6-7 MovCircle 整圆关键点示意



MovJRel - 相对偏移指令

MovJ 插补相对偏移指令。该指令总是以当前机器人位置或者上一步运动指令的目标位置为起点位置，然后机器人相对移动位移偏移。

与 MovJ 等指令不同，其使用的目标相对位置值无法通过示教获取，用户需要事先在变量列表中新建关节相对位置类型的变量，然后在下拉列表中选择。

图6-8 MovJRel 参数编辑界面

指令: MovJRel	行号: 3
工程名: project0	程序名: program0
目标相对位置	L: DAPOS0
目标速度	DEFAULT
过渡类型	DEFAULT
工件负载	DEFAULT
AddDo	

- 目标相对位置：执行该指令时机器人要移动的位置增量。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- AddDo：机器人执行完该指令后，进行 IO 操作。

MovLRel-插补相对偏移指令

MovL 插补相对偏移指令。该指令总是以当前机器人位置或者上一步运动指令的目标位置为起点位置，然后机器人相对于坐标系或者工具进行偏移运动。

与 MovL 等指令不同，其使用的目标相对位置值无法通过示教获取，用户需要事先在变量列表中新建坐标系相对位置类型的变量，然后在下拉列表中选择。

图6-9 MovLRel 参数编辑界面

目标相对位置	L: DCPOS0
参考坐标系	*COORD*
目标速度	DEFAULT
过渡类型	*RELATIVE*
过渡值	DEFAULT
工件负载	DEFAULT
速率类型	DEFAULT
AddDo	

- 目标相对位置：执行该指令时机器人要移动的位置增量。
- 参考坐标系：坐标系偏移或工具偏移选择；
 - Coord：相对于当前用户坐标系偏移；
 - Tool：相对于工具坐标系偏移，即参考 Tx、Ty、Tz 平移或者旋转。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 速率类型：指令速度是否受全局倍率影响。
 - DEFAULT：同 GOVRON，即指令速度受示教编程器全局倍率影响。
 - GOVRON：指令速度受全局倍率影响。
 - GOVROFF：指令速度不受全局倍率影响。
- 速率类型：机器人执行该轨迹时指令速度是否受示教编程器全局倍率影响。
 - DEFAULT：同 GOVRON，即指令速度受示教编程器全局倍率影响。
 - GOVRON：指令速度受示教编程器全局倍率影响。

- GOVROFF: 指令速度不受示教编程器全局倍率影响。
- AddDo: 机器人执行完该指令后, 进行 IO 操作。

MovJOffset - 关节空间偏移运动指令

与 MovLOffset 指令相比，MovJOffset 指令是以 MovJ 运动进行偏移运动，且目标相对位置支持 DAPOS 和 DCPOS 这两种类型。

而与 MovJRel 指令相比, MovJOffset 指令所指定的起点位置不局限于当前机器人位置或者上一步运动指令的目标位置, 且能够相对坐标系或工具进行偏移。

图6-10 MovJOffset 参数编辑界面

参考位置	 L: P0	▶
目标相对位置	 L: DCPOS0	▶
参考坐标系	"COORD"	▶
工具参数	S: nullTool	▶
坐标系参数	 S: World	▶
目标速度	DEFAULT	▶
过渡类型	DEFAULT	▶
工件负载	DEFAULT	▶
AddDo		▶

- 参考位置：偏移运动的起点（APOS 或 CPOS）；
- 目标相对位置：执行该指令时机器人要移动的位置增量（DAPOS 或 DCPOS）；
- 参考坐标系：坐标系偏移或工具偏移选择。其中，“Coord”表示相对于当前用户坐标系偏移；“Tool”表示相对于工具坐标系偏移，即参考 Tx、Ty、Tz 平移或者旋转；
- 工具参数：参考位置所属的工具信息；
- 坐标系参数：参考位置所属的坐标系信息；
- 目标速度：指令运行速度（SPEED 类型变量），其中，仅 per 参数对该指令绝对有效；
- 过渡类型：机器人逼近终点时的过渡方式，RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离；
- 过渡值：机器人逼近终点时的过渡值（ZONE 类型变量），其中，per 参数对应相对过渡类型有效，dis 参数对应绝对过渡类型有效；
- 工件负载：机器人执行该轨迹时使用的工件负载参数；

AddDo: 机器人执行完该指令后, 进行 IO 操作。

MovLOffset - 直线偏移运动指令

MovL 插补相对偏移指令。该指令以指定参考位置为起点位置，相对于坐标系或者工具进行偏移运动。

与 MovLRel 唯一的区别是该指令指定了偏移运动的起点，不一定是当前位置或者上一条指令的终点，其他功能均相同。

图6-11 MovLOffset 参数编辑界面

参考位置	 L: P0
目标相对位置	L: DCPOS0
参考坐标系	"COORD"
工具参数	S: nullTool
坐标系参数	 S: World
目标速度	DEFAULT
过渡类型	"RELATIVE"
过渡值	DEFAULT
工件负载	DEFAULT
速率类型	DEFAULT
AddDo	

相对于 MovLRel 新增指令参数如下：

- 参考位置：偏移运动的起点（APos 或 CPos）。
- 工具参数：参考位置所属的工具信息。
- 坐标系参数：参考位置所属的坐标系信息。

MovJSearch - 寻位指令（MovJ）

寻位指令，指的是在执行这条指令过程中，进行 IO 检测或力矩检测。

当检索条件为 IO 检测时，索引 IO 的值达到触发值时，停止运行该条指令，继续执行下条指令，中断返回值置 1。

当检索条件为力矩检测时，索引轴的力矩值达到触发值时，停止运行该条指令，继续执行下条指令，中断返回值置。

指令执行过程中没有中断，中断返回值置 0。

图6-12 MovJSearch 参数编辑界面

目标位置	 L: P1
目标速度	S: V50
过渡类型	DEFAULT
工具参数	DEFAULT
坐标系参数	DEFAULT
工件负载	DEFAULT
检测类型	"DITrig"

触发索引	0
触发值	0
停止类型	DEFAULT
返回值	L: INTO
寻位成功的位置	DEFAULT
寻位失败时跳转的标签名	
AddDo	

- 目标位置：指令终点位置（Apos 或 CPos 类型）。
- 目标速度：指令位置运行速度（SPEED 类型变量）。仅 per 参数对该指令绝对有效。
- 工具参数：工具信息。

- 坐标系参数：目标位置参考的笛卡尔坐标系。
- 工件负载：工件负载信息。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。
 注意：目前版本中 MovJSearch 不支持与下一条运动指令进行过渡规划。
- 检测类型：
 - DITrig：物理数字量 IO 输入检测。
 - AITrig：物理模拟量 IO 输入检测。
 - SIMDITrig：虚拟数字量 IO 输入检测。
 - SIMAITrig：虚拟模拟量 IO 输入检测。
 - TorqTirg：力矩检测。
- 触发索引：
 - 对于 DITrig/SIMDITrig/ AITrig/ SIMAITrig，该参数表示需要检测的 IO 端口号。
 - 对于 TorqTirg，该参数表示需要检测的轴号。
- 触发值：
 - 对于 DITrig/SIMDITrig/ AITrig/ SIMAITrig，IO 检测的阈值。
 - 对于 InputTrig，IO 检测的阈值。
- 对于 TorqTirg，力矩检测的阈值，单位为额定转矩的千分比。
- 停止类型：检测到信号时停止的方式。
 - DEFAULT：立即停止；
 - HARDSTOP：立即停止；
 - SOFTSTOP：减速停止。
- 减速系数：当“停止类型”选择“SOFTSTOP”时，该值有效。用于调节减速加速度的系数，该值越大减速越快，反之越慢。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
- 返回值：根据返回值可判断运行过程中是否有中断。0：未检测到；1：寻位成功；2：在运动起点就检测信号
- 寻位成功的位置：返回中断发生时刻机器人的坐标位置。（APOS 类型），该参数可缺省。
- 寻位失败时跳转的标签名：寻位失败时程序指针将跳转到的 label 名，这个 label 为当前程序已示教过的 label 名（如“Start:”），该参数可缺省。



说明

目前版本中 MovJSearch 不支持与下一条运动指令进行过渡规划。

示例 1：

```
MovJ(P1, V50, "RELATIVE", C100)
MovJSearch(P2, V50, "RELATIVE", C100, "DITrig", 16, 1, INT0)
MovJ(P3, V50, "RELATIVE", C100)
```

机器人以 50 的速度运动到 P1 点，其后在运动到 P2 点的过程中机器人会开始检测 16 号输入口的状态；当检测到有高电平时，机器人会立即停止往 P2 点的运动，开始往 P3 点运动，同时 INT0 中的值会设为 1；如果在往 P2 点的运动过程中，16 号输入口没有任何的高电平信号，那么等运动到 P2 点后机器人会继续往 P3 点运动，同时 INT0 被设为 0。

MovLSearch - 寻位指令（MovL）

寻位指令，指的是在执行这条 MovL 指令时，进行 IO 检测或力矩检测，与 MovJSearch 类似。

当检索条件为 IO 检测时，索引 IO 的值达到触发值时，停止运行该条指令，继续执行下条指令，中断返回值置 1。

当检索条件为力矩检测时，索引轴的力矩值达到触发值时，停止运行该条指令，继续执行下条指令，中断返回置 1。

若指令运行过程中无中断，中断返回置 0。

图6-13 MovLSearch 参数编辑界面

指令: MovLSearch 行号: 4	
工程名: project 程序名: project1	
目标位置	L: P2
目标速度	S: V50
过渡类型	DEFAULT
工具参数	DEFAULT
坐标系参数	DEFAULT
工件负载	DEFAULT
检测类型	"DITrig"

- 目标位置：指令终点位置（APos 或 CPos 类型）。
- 目标速度：指令位置运行速度（SPEED 类型变量），其中，参数 tcp 对该指令直线速度有效，ori 对姿态速度有效（且两者均不可以为 0）。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。

注意：目前版本中 MovJSearch 不支持与下一条运动指令进行过渡规划。
- 检测类型：
 - DITrig：物理数字量 IO 输入检测。
 - AITrig：物理模拟量 IO 输入检测。
 - SIMDITrig：虚拟数字量 IO 输入检测。
 - SIMAITrig：虚拟模拟量 IO 输入
- 工具参数：工具信息。
- 坐标系参数：目标位置参考的笛卡尔坐标系。
- 工件负载：工件负载信息。
- 触发索引：

对于 DITrig/SIMDITrig/ AITrig/ SIMAITrig，该参数表示需要检测的 IO 端口号。

对于 TorqTirg，该参数表示需要检测的轴号。
- 触发值：

对于 DITrig/SIMDITrig/ AITrig/ SIMAITrig，IO 检测的阈值。

对于 TorqTirg，力矩检测的阈值，单位为额定转矩的千分比。
- 停止类型：检测到信号时停止的方式。
 - DEFAULT：立即停止；
 - HARDSTOP：立即停止；
 - SOFTSTOP：减速停止。
- 减速系数：当“停止类型”选择“SOFTSTOP”时，该值有效。用于调节减速加速度的系数，该值越大减速越快，反之越慢。
- 返回值：根据返回值可判断运行过程中是否有中断。0：未检测到；1：寻位成功；2：在运动起点就检测信号

- 寻位成功的位置：返回中断发生时刻机器人的坐标位置。（CPOS 类型），该参数可缺省。
- 寻位失败时跳转的标签名：寻位失败时程序指针将跳转到的 label 名，这个 label 为当前程序已示教过的 label 名（如 “Start:”），该参数可缺省。
- AddDo：机器人执行完该指令后，可进行 IO 操作。



说明

目前版本中 MovLSearch 不支持与下一条运动指令进行过渡规划。

示例 1:

```
MovL(P1, V50, "RELATIVE", C100)
MovLSearch(P2, V50, "RELATIVE", C100, "DITrig", 16, 1, INT0)
MovL(P3, V50, "RELATIVE", C100)
```

机器人以 50 的速度运动到 P1 点，其后在运动到 P2 点的过程中机器人会开始检测 16 号输入口的状态，当检测到有高电平时，机器人会立即停止往 P2 点的运动，开始往 P3 点运动，同时 INT0 中的值会设为 1；如果在往 P2 点的运动过程中，16 号输入口没有任何的高电平信号，那么等运动到 P2 点后机器人会继续往 P3 点运动，同时 INT0 被设为 0。

MovLW - 摆动指令

摆动指令，指的是根据设定的摆频、摆幅，在前进方向上做摆弧运动。

图6-14 MovLW 参数编辑界面

目标位置	L: P0	
目标速度	DEFAULT	
过渡类型	"RELATIVE"	
过渡值	DEFAULT	
工具参数	DEFAULT	
坐标系参数	DEFAULT	
工件负载	DEFAULT	
摆弧变量	L: WEAVE0	
速率类型	DEFAULT	
AddDo		

- 目标位置：指令终点位置（APOS 或 CPOS 类型位置点）。在下拉列表中选择已有的点或在弹出的对话框中新建并示教一个变量。
- 目标速度：指令运行速度。
 - 设置为 “DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为 “DEFAULT”，表示使用默认值 C100。

- 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为 “DEFAULT”，表示沿用最近一次设置的工具参数。
 - 设置为 TOOL 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工具参数不同的坐标系时，系统会将工具参数切换成设置的工具参数。若本段与前后段轨迹的工具参数变化，则不支持过渡。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 设置为 “DEFAULT”，表示沿用最近一次设置的坐标系。
 - 设置为 USERCOORD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换成设置的坐标系。
 - 设置为变位机坐标系时，机器人的运动相对于变位机坐标系，指令的目标速度相对于变位机法兰。变位机功能只支持同一个变位机之间的过渡。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为 “DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 摆弧变量：摆动参数。

可以设置摆频、摆幅、停止时间、旋转等参数。
- 速率类型：机器人执行该轨迹时指令速度是否受示教编程器全局倍率影响。
 - DEFAULT：同 GOVRON，即指令速度受示教编程器全局倍率影响。
 - GOVRON：指令速度受示教编程器全局倍率影响。
 - GOVROFF：指令速度不受示教编程器全局倍率影响。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。
 - IO 指令：执行 IO 操作。

说明

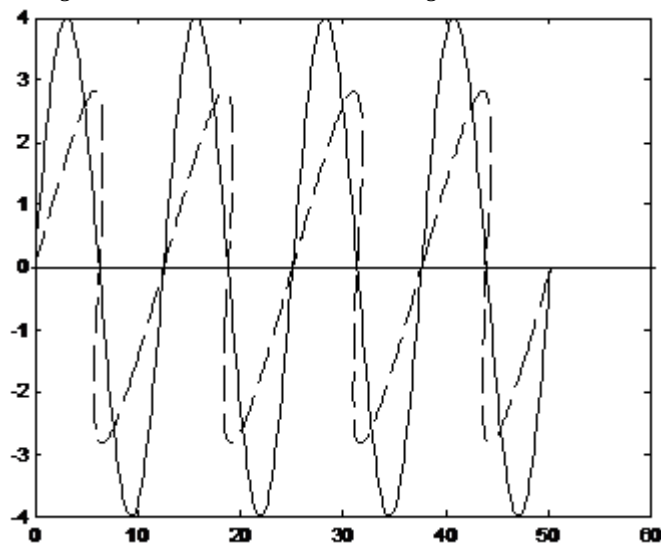
WEAVE 类型变量有如下参数：

- Type: 机器人摆动类型, 该值为 1 表示正弦摆, 该值为 2 表示三角摆
- Freq: 摆频, 单位: Hz
- Amp_L: 机器人摆动左幅值, 单位: mm
- Amp_R: 机器人摆动右幅值, 单位: mm
- StopTime_L: 波峰处停止时间, 指在摆动到左幅值时停止的时间, 单位: ms
- StopTime_C: 中间位置停止时间, 指在摆动到中间位置时摆动停止的时间, 单位: ms
- StopTime_R: 波谷处停止时间, 指在摆动到右幅值时停止的时间, 单位: ms
- RotAngle_X: 摆动基准平面绕摆动行进方向的旋转角度, 该参数用于决定最终的摆动平面, 单位: deg
- RotAngle_Z: 绕摆动平面法向量的旋转角度, 决定正弦摆的形状, 单位: deg



注意

- 由行进路线与工具坐标系的 Tz 方向确定的平面的法向量, 与行进路线构成的平面为摆动基准平面。
- RotAngle_Z 的作用, 如下图所示: RotAngle_Z=0 为实线, 不为 0 为虚线所示。



MovCW

与 MovLW 类似。

MovCircleW

与 MovLW 类似。

MovH - 螺旋运动指令

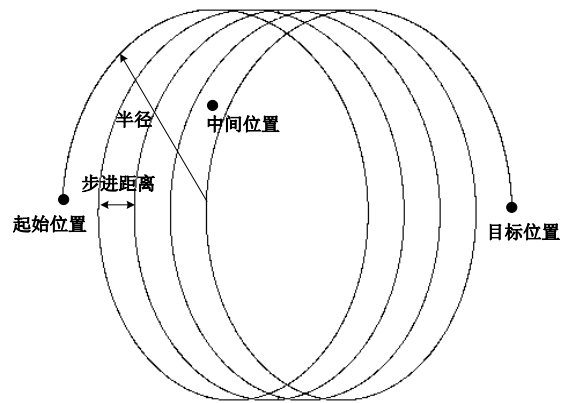
螺旋运动指令, 通过该指令可以使机器人 TCP 点做螺旋运动到目标位置, 若运动的起止姿态不同, 则运行过程中姿态随位置同步地旋转到终点的姿态。

图6-15 MovH 参数编辑界面

中间位置	 L: P1	⌵
目标位置	 L: P2	⌵
目标速度	DEFAULT	⌵
过渡类型	DEFAULT	⌵
工具参数	DEFAULT	⌵
坐标系参数	DEFAULT	⌵
工件负载	DEFAULT	⌵
半径(mm)	10	⌵
步进距离(mm)	5	
AddDo		

- 中间位置：螺旋中间辅助点位置。辅助点是为了确定螺旋平面以及螺旋方向（顺时针旋进、逆时针旋进），螺旋运动并不一定经过辅助点
- 目标位置：螺旋终点位置
- 目标速度：指令位置运行速度，其中，参数 tcp 对该指令直线速度有效，ori 对姿态速度有效。螺旋线最大轨迹线速度为系统最大速度的 0.25 倍
- 过渡类型：不支持过渡
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工具参数。
 - 设置为 TOOL 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工具参数不同的坐标系时，系统会将工具参数切换成设置的工具参数。若本段与前后段轨迹的工具参数变化，则不支持过渡。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的坐标系。
 - 设置为 USERCOOR 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换成设置的坐标系。
 - 设置为变位机坐标系时，机器人的运动相对于变位机坐标系，指令的目标速度相对于变位机法兰。变位机功能只支持同一个变位机之间的过渡。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
- 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡
- 半径：螺旋线半径，单位 mm
- 步进距离：螺旋线旋转一周步进距离，单位 mm
- AddDo：机器人执行完该指令后，可进行 IO 操作

图6-16 MovH 螺旋运动关键点示意



说明

螺旋运动不支持过渡、不支持 WaitFinish

示例：

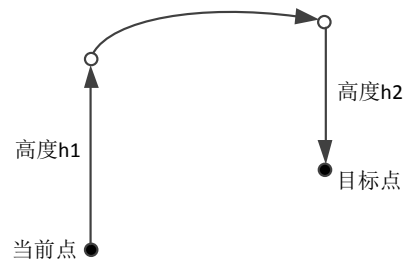
```
MovL (P1, V50, "RELATIVE", C50)
MovH (P2,P3, V100, "FINE", nullTool,World,PayLoad0,10,5)
```

机器人会先用 V50 的速度运动到 P1 点，其后执行螺旋线指令时，以 P1 点为螺旋运动起点，以 P1、P2、P3 三点确定螺旋运动平面，以 P2 点确定螺旋运动的旋进方向，按照螺旋半径为 10mm、步进距离为 5mm 向螺旋运动终点 P3 旋进，从而完成整个的螺旋运动。

MovArch - 拱形运动指令

拱形运动指令。整个运动过程为：首先机器人在当前点抬升 h1 高度，其后再运动到距离目标点高度为 h2 处，最后再下降 h2 高度到达目标点，完成一次拱形运动，拱形运动为点到点的运动方式。该运动是当直接运动到目标点可能存在障碍物时，自动提高到障碍物外的高度再运动到目标点，以提高安全性及操作性。

图6-17 运动轨迹





说明

该指令目前仅适用于 scara 机型。

图6-18 MovArch 参数编辑界面

目标位置	 L: P1	⌵
目标速度	DEFAULT	⌵
过渡类型	DEFAULT	⌵
工具参数	DEFAULT	⌵
坐标系参数	DEFAULT	⌵
工件负载	DEFAULT	⌵
抬升高度(mm)	1.0	⌵
下降高度(mm)	1.0	⌵
转角值	S: C100	⌵
AddDo		⌵

- 目标位置：指令终点位置（APOS 或 CPOS 类型位置点）。在下拉列表中选择已有的点或在弹出的对话框中新建并示教一个变量。
- 目标速度：指令运行速度（SPEED 类型变量）。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果与 DEFAULT 相同。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值（ZONE 类型变量）。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工具参数。
 - 设置为 TOOL 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工具参数不同的坐标系时，系统会将工具参数切换成设置的工具参数。若本段与前后段轨迹的工具参数变化，则不支持过渡。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的坐标系。
 - 设置为 USERCOOR 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换成设置的坐标系。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 抬升高度：机器人末端执行拱形运动时的抬升高度，如图 h1。
- 下降高度：机器人末端执行拱形运动时的下降高度，如图 h2。

- 转角值：机器人执行拱形运动时的过渡参数（ZONE 类型变量）。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。
 - IO 指令：执行 IO 操作。

示例：

```
MovL (P1, V50, "RELATIVE", C50)
MovArch (P2, V100, "FINE",10,10,C100)
机器人会先用 V50 的速度运动到 P1 点，其后执行拱形指令，机器人以 V100 的速度先运动到 P1.z+ 10mm 的位置，其后再运动到 P2.Z+10mm 的位置，最后运动到 P2 的位置，整个运动间均以 C100 过渡。
```

MovLArch – 直线拱形运动指令

直线拱形运动指令。该指令以当前位置作为起点位置，先做抬升运动，然后以 MovL 运动方式运动到目标点上方，最后下降到目标点。

如下图所示：首先，机器人在当前点抬升 h1 高度，再运动到距离目标点高度为 h2 处，最后再下降 h2 高度到达目标点，完成一次拱形运动。

该拱形运动方式是空间直线运动方式。该运动是指当直接运动到目标点可能存在障碍物时，自动提高到障碍物外的高度再运动到目标点，以提高安全性及操作性。

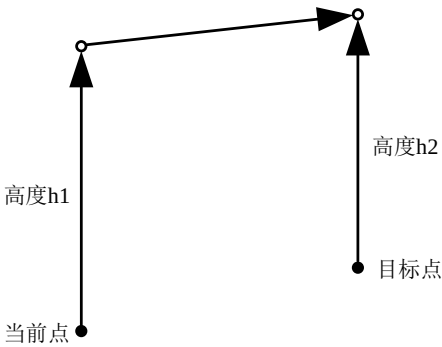


图6-19 MovLArch 参数

指令:	MovLArch	行号:	3
工程名:	project0	程序名:	program0
目标位置	L: P1		
目标速度	DEFAULT		
过渡类型	DEFAULT		
工具参数	DEFAULT		
坐标系参数	DEFAULT		
工件负载	DEFAULT		
抬升高度(mm)	1.0		
下降高度(mm)	1.0		
转角值	S: C100		
速率类型	DEFAULT		
AddDo			

- 目标位置：指令终点位置（APOS 或 CPOS）
- 目标速度：指令运行速度；
- 过渡类型：机器人逼近终点时的过渡方式； DEFAULT:
- 过渡值：机器人逼近终点时的过渡值；

- 工具参数：机器人执行该轨迹时使用的工具参数；
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数；
- 工件负载：机器人执行该轨迹时使用的工件负载参数；
- 抬升高度：机器人末端执行直线拱形运动时的抬升高度，单位 mm，如图 h1；
- 下降高度：机器人末端执行直线拱形运动时的下降高度，单位 mm，如图 h2；
- 转角值：机器人执行直线拱形运动时的过渡参数；
- 速率类型：机器人执行该轨迹时，指令速度是否受示教器全局倍率的影响；
- AddDo：机器人执行完该指令后，进行 IO 操作。



说明

MovLArch 指令目前仅适用于 Scara 机型。

OnDistance - 距离触发指令

距离触发指令，在不打断过渡的前提下，控制器可以从起点运动一段距离或者距离终点一段距离时触发执行一些操作，如赋值，设置 IO 等操作。

图6-20 OnDistance 参数编辑界面

触发类型	"FromBegin"	⌵
触发距离(mm)	0	⌵
触发时间(ms)	0	⌵
触发操作		⌵

当触发类型设定的是从起点判断时，机器人运动到设定的距离后，控制器会进入触发等待，延时触发时间（时间为正值）后，开始执行相关的触发操作。

当触发类型设定的是从末点判断时，机器人运动到设定的距离后，控制器会进入触发等待，延时触发时间（时间为正值）后，开始执行相关的触发操作。

当触发时间为负值时，控制器会提前计算机器人运动到设定的距离前需要多长时间，当延时到触发时间时，系统会开始执行相关的触发操作。

- 触发类型：指定当前触发是距离起点一段距离触发还是距离终点一段距离触发。
- 触发距离：触发的距离参数。该参数为 REALC 参数类型，即可输入常量也可输入 REAL 类型的变量。
- 触发时间：满足触发条件后触发延时触发的时间。该参数为 INTC 参数类型，即可输入常量也可输入 INT 类型的变量。设定为负数，表示提前触发。
- 触发的操作：满足触发条件后，机器人需要执行的触发操作。

示例 1：在 P1 到 P2 轨迹的起点距离为 200mm 处，延时 100ms 执行 Do 操作。

```
MovL (P1, V50, "RELATIVE", C0)
OnDistance("FromBegin",200,100) DO SetDO(DO0,1);
MovL (P2, V50, "RELATIVE", C0)
```

示例 2：在 P1 到 P2 轨迹的终点距离为 200mm 处，延时 100ms 执行 Do 操作。

```
MovL (P1, V50, "RELATIVE", C0)
OnDistance("FromEnd",200,100) DO SetDO(DO0,1);
MovL (P2, V50, "RELATIVE", C0)
```




注意

- 当距离触发满足且时间触发还在延时，如果当前轨迹已经运行完，触发将延后执行，等待时间触发满足后再执行。
- 当设定的触发距离超出轨迹长度时，会有两种情况：触发类型为起点触发时，运动到末点时将执行触发判断。触发类型为末点触发时，在起点处将执行触发判断。满足条件则立刻触发。
- 两段轨迹之间有多条触发指令时，触发各自检测，满足条件后各自触发执行，互不干扰。
- 执行 OnDistance 之前必须执行笛卡尔坐标系下运动指令，否则 OnDistance 将不起作用，同时状态栏提示该操作无效，但不影响系统的运行。
- OnDistance 与下一条笛卡尔坐标系下运动指令之间仅可插入等待指令，否则执行报错。等待指令如 Wait, Waitcondition, WaitFinish 等。
- 离参数不能为负值，时间参数可正可负。
- 触发指令不支持单步模式。
- OnDistance 只支持 MOVL,MOVJ 指令，即该指令两侧不允许存在 MOVJ 指令，否则执行报错。
- 在等待触发的工程中发生停止操作，或者由暂停转为停止，系统会清空所有的监控的触发缓存。

OnParameter - 距离百分比触发指令

距离百分比触发指令，用于在不打断过渡的前提下，在直线轨迹的某一处触发执行一些操作，比如赋值，设置 IO 等快捷操作。

图6-21 OnParameter 参数编辑界面

百分比	0
触发时间(ms)	0
触发操作	

当机器人运动到设定的触发百分比路径时，控制器进入触发等待，延时触发时间（时间为正值）后，开始执行相关的触发操作。

当触发时间为负值时，控制器会提前计算机器人运动到设定的距离前需要多长时间，当延时到触发时间时，系统会开始执行相关的触发操作。

- 路径百分比：需要触发的路径所在占比。
- 触发时间：满足触发条件后触发延时触发的时间。该参数为 INTC 参数类型，即可输入常量也可输入 INT 类型的变量。
设定为负数，表示提前触发。
- 触发的操作：满足触发条件后，机器人需要执行的触发操作。



说明

触发指令不支持单步模式。

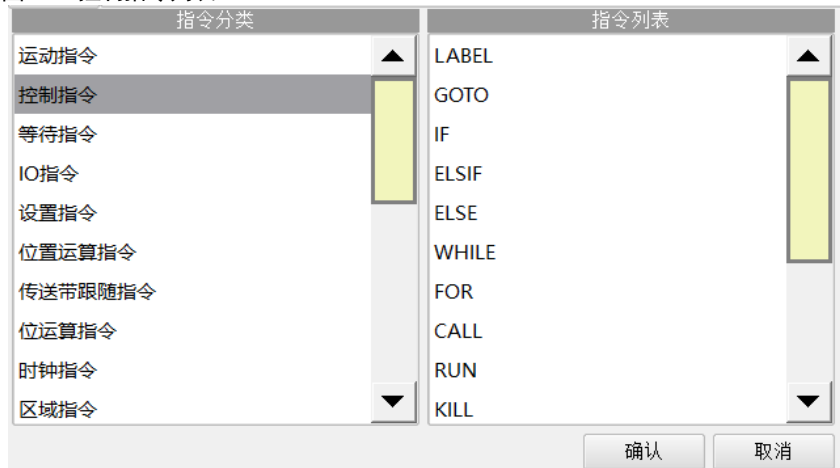
示例：在 P1 到 P2 轨迹运行距离为 60%处，延时 500ms 执行 Do 操作。

```
MovJ (P1, V50, "RELATIVE", C0)
OnParameter(60,500) DO SetDO(DO0,1);
MovL (P2, V50, "RELATIVE", C0)
```

6.2 控制指令

控制指令列表如图 6-22 所示。

图6-22 控制指令列表



LABEL - 定义跳转目标

Label 指令用于定义 GOTO 跳转目标。

GOTO - 跳转指令

GOTO 指令用于跳转到程序不同部分。跳转目标通过 LABEL 指令定义。不允许从外部跳转进入内部程序块。内部程序块可能是 WHILE 循环程序块或者 IF 程序块。

示例：

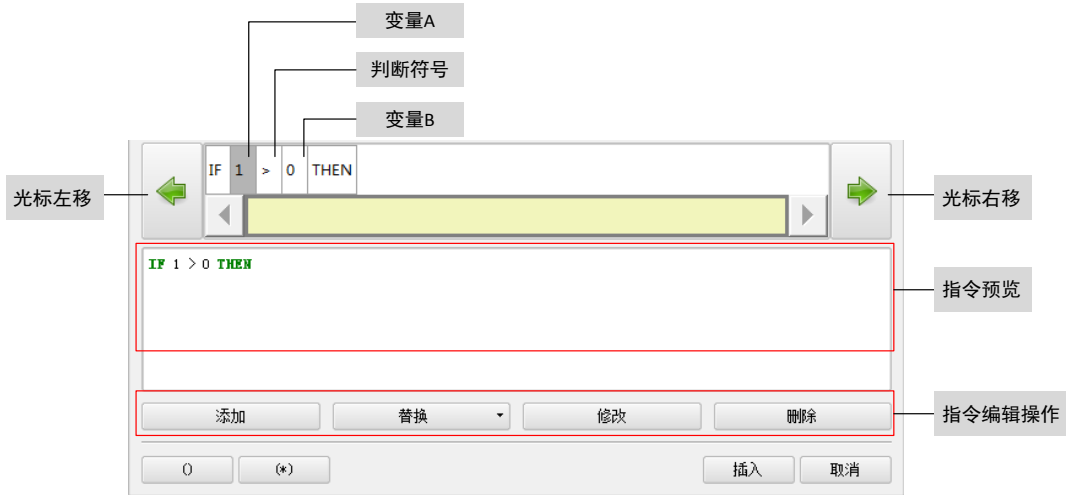
```
label1:  
MovJ(P0)  
GOTO label1  
LABEL 指令设置标签 label1, GOTO 指令执行时, 跳转到行“label1:”
```

IF-条件判断指令

IF 指令用于条件判断表达式跳转控制。当条件判断表达式结果为真时，程序执行 IF 下程序块的内容。类似于 c++ 中的 IF 语句。IF 条件判断表达式结果必须是 BOOL 类型。每一个 IF 指令必须以关键字 ENDIF 做为条件判断程序块的结束。

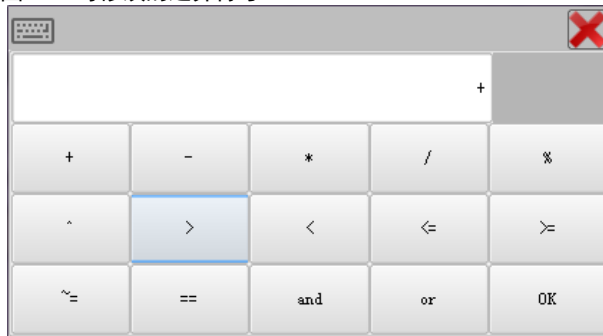
当选择“控制指令→ IF”后，将进入 IF 语句的设置界面，如图 6-23 所示。

图6-23 IF 语句设置界面



- 变量 A 和变量 B 可设定为常数或参数：点击“替换”然后选择想要替换的常量、变量或函数。
若要修改变量的值，可将光标移至需要修改的变量上，再点击“修改”，并在弹出的对话框中设置想要修改的数值。
- 判断符号用来判断变量 A 和变量 B 之间的关系，若为真，则执行 THEN 后的指令，若为假，则跳过 THEN 后的指令。
若要修改判断符号，可将光标移至判断符号上，再点击“修改”，并在弹出的对话框中选择需要的符号。
- 点击“添加”，可在光标的当前位置新增一个“+<Exp>”，其中，“+”表示运算符，“<Exp>”表示操作数。
 - 点击“+”，然后点击“修改”并在弹出的软键盘中选择想要的运算符，如图 6-24 所示。

图6-24 可修改的运算符



- 点击“<Exp>”，然后点击“替换”或“修改”，可变更操作数为想要的常量、变量或函数。
- 点击想要删除的运算符或操作数，然后点击“删除”，可删除该运算符及随后的操作数。
- 点击某个运算符，再点击“()”，可为当前运算符的前后操作数增加一个括号。
点击某个操作数，再点击“()”，可为当前的操作数增加一个括号。
- 若要删除括号，点击运算符或操作数，再点击“(*)”，可删除被选中的运算符或操作数响应的括号。

```
IF    x >1 and y == 1 and z < 1 THEN
  y = 10
ENDIF
```



说明

若要删除 IF 指令，需在程序编辑界面中点击“多行选择”，然后点击 IF 和 END_IF 之间的控制语句，再点击右侧菜单“编辑→删除”。

在“替换”时，可选择“数学函数”或“字符函数”。关于函数的说明，请参见数学运算函数和其他函数。

ELSIF

ELSIF 指令依赖于 IF 指令，紧跟 IF 指令块后，当 IF 逻辑不成立，则进行 ELSIF 逻辑判断。

设置 ELSIF 表达式的方法与设置 IF 表达式的方法相同。

使用该指令时，请注意如下要求：

- ELSIF 可添加在 IF 指令块后，但不可单独添加。
- ELSIF 也可添加在 ELSIF 指令块后，即一个 IF 指令块后面跟多个 ELSIF 指令块。
- ELSIF 指令可嵌套添加，即在 IF 指令块或 ELSIF 指令块内可添加 IF 与 ELSIF 指令。
- IF 逻辑 true 后，执行后，不执行后续的 ELSIF 指令，直接到 ENDIF。
- IF 逻辑 false 后，执行后续的 ELSIF 指令，ELSIF 逻辑 false，则执行后续的 ELSIF 指令。
- IF 逻辑 false 后，执行后续的 ELSIF 指令，ELSIF 逻辑 true，执行后，不执行后续的 ELSIF 指令，直接到 ENDIF。

ELSE

ELSE 指令依赖于 IF 或 ELSIF 指令，紧跟 IF 或 ELSIF 指令块后，当 IF 或 ELSIF 条件判断均不成立时，将执行的语句。

WHILE – 只要满足条件就循环

WHILE 指令在满足条件的时候循环执行子语句。循环控制表达式必须是 BOOL 类型。该指令必须以关键字 ENDWHILE 作为循环控制结束。

设置 WHILE 表达式的方法与设置 IF 表达式的方法相同。

例如：

```
INT0= 1
WHILE INT0<=100DO
MovJ (P1, V50, "RELATIVE", C0)
MovJ (P2, V50, "RELATIVE", C0)
INT0=INT0+1
ENDWHILE
```

该指令执行两点之间的循环运动 100 次。

FOR – for 循环指令

FOR 循环指令，由 3 段字符串组成，以逗号隔开。该指令以关键字 ENDFOR 结束。

FOR 循环指令执行末尾循环体后将再次进行条件判断，若条件还成立，则继续重复上述循环，当条件不成立时则跳出当下 for 循环。



格式：FOR（变量=初始值，条件表达式，末尾循环体）DO //满足条件表达式，执行末尾循环体；不满足直接跳出循环。

.... //需执行的代码，每次 FOR 结束时执行一次。

ENDFOR

例如：

```
FOR INT0.value = 1, INT0.value < 10, INT0.value = INT0.value + 1 DO
  INT1.value = INT1.value+1
ENDFOR
```

CALL - 调用指令

调用指令，当前程序跳转至同一工程下另一个子程序或全局工程下的全局程序，子程序执行完后跳转回当前程序。

假如需要调用的同一工程下的另一子程序为 program1，在程序中生成命令为：

```
CALL program1 或者 CALL L:program1
```

假如需要调用的全局工程下的子程序为 program1，在程序中生成命令为：

```
CALL G:program1
```

程序执行流程如图 6-25 所示。

图6-25 CALL 指令执行流程



RUN - 程序并行指令

程序并行运行指令，使得机器人能够在运行当前程序的同时，运行其它程序，（当前程序继续运行不跳转的同时启动另一个程序）且运行的程序必须是同一个工程下。假如需要运行的程序为 program1，在程序中生成命令为：

```
RUN program1
```



说明

目前，控制器可支持运行 8 个并行程序。

程序执行流程如图 6-26 所示。

图6-26 RUN 指令执行流程



KILL - 停止并行运行指令

停止并行运行的指令，使得机器人能够在运行当前程序的同时，停止运行其它程序，且停止运行的程序必须是同一个工程下且为运行状态。假如需要停止运行的程序为 program1，在程序中生成命令为：

```
KILL program1
```

RETURN - 返回指令

返回指令。一般情况下，执行该指令后，程序会跳转至程序的末尾，如果是在 CALL 指令中使用了 RETURN 指令，则返回至 CALL 指令的上一级程序。

示例：

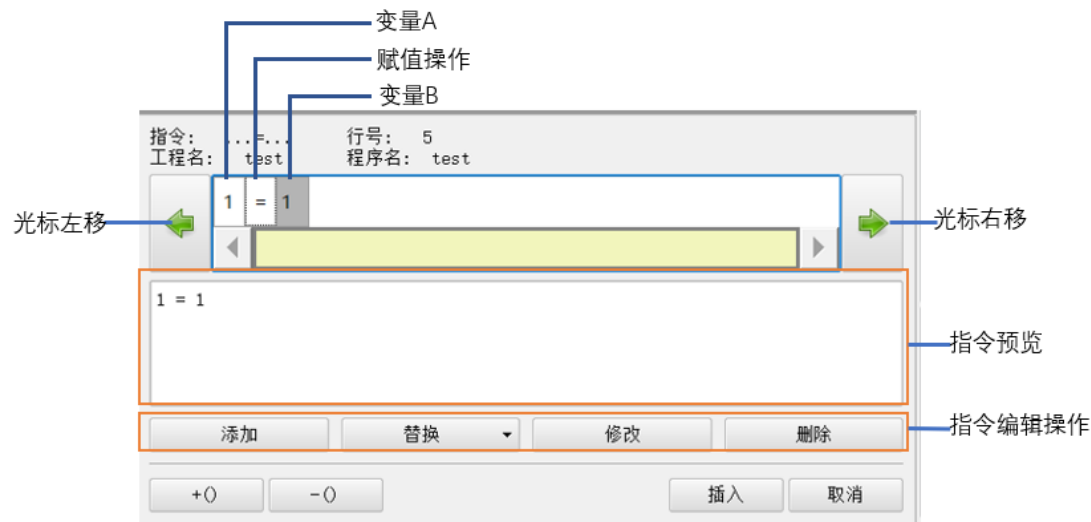
```
IFx >1THEN
```

```
RETURN
ENDIF
x = x + 1
/*other instructions*/
END;
当满足条件，执行到 RETURN 指令，跳转到 END
```

...=... - 建立表达式

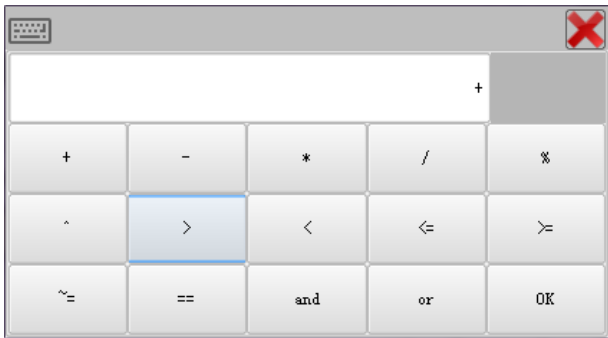
建立一个表达式，用来给某个变量赋值，如图 6-27 所示。

图6-27 建立表达式说明



- 变量 A：只能设置为变量。点击“替换”，选择需要替换的变量。
- 变量 B：可设置为常数或变量，点击“替换”选择需要替换的常量、变量和函数。
- 赋值操作：将变量 B 的值赋给变量 A。
- 指令预览：显示输入的指令。
- 指令编辑操作：
 - 添加：将光标移动至“变量 B”，然后点击“添加”，表达式会自动添加一个新的空数值，默认的运算符号为“+”。
 - 替换：将光标移动至需要替换的变量，点击“替换”选择需要替换的常量、变量或函数。
 - 修改：若要修改变量的值或运算符号，可将光标移至需要修改的变量或运算符号上，再点击“修改”，并在弹出的对话框中设置想要修改的数值或运算符号。如下图所示：

图6-28 可修改的运算符号



运算符介绍：

运算符	作用	运算符	作用
-----	----	-----	----

+	加法运算符，求和	<	小于，如：a<b 判断 a 的值小于 b
-	减法运算符，求差	< =	小于等于，如：a<=b 判断 a 的值小于或等于 b
*	乘法运算符，求积	> =	大于等于,如：a>=b 判断 a 的值大于或等于 b
/	除法运算符	~ =	约等于
^	指数运算符	= =	判断等于，如：a==b，判断 a 和 b 相等
%	百分号	and	与运算，a and b 同时满足条件 a 和 b
>	大于，如：a>b 判断 a 的值大于 b	or	或运算，a or b，满足 a 条件或 b 条件

删除：选择某一新增运算符或随后的变量，然后点击“删除”，可删除该新增的运算符和随后的变量。

- + ()：添加 () 运算符。
- () -：删除添加的 () 运算符。



说明

在“替换”时，可选择“数学函数”或“字符函数”，关于函数的说明，请参见和。

所有操作都需要先移动光标至需要修改的参数，再进行指令编辑操作。

/...*/ - 注释指令

注释指令，为程序添加释义或备注。

该指令不会影响程序的运行，仅仅是便于用户阅读和理解程序。

6.3 等待指令

等待指令列表如图 6-29 所示。

图6-29 等待指令列表

指令分类	指令列表
运动指令	Wait
控制指令	WaitFinish
等待指令	WaitCondition
IO指令	
设置指令	
位置运算指令	
位运算指令	
时钟指令	
区域指令	
视觉指令	
码垛指令	

Wait – 延时等待指令

用于设置机器人等待时间，时间单位为 ms，可为 int 常量或者 INT 变量类型。假如设置等待 1s，生成命令的两种方式为：

```
// 使用常量的方式
Wait (1000)
// 使用变量的方式
INT0.value = 1000
Wait (INT0)
```

WaitFinish – 等待运动段执行百分比指令

用于控制机器人的运动过程中的程序执行，可设置运动执行到一定百分比时，执行下一行指令

指令: WaitFinish 行号: 5
工程名: project0 程序名: project1

执行百分比(%)	20
----------	----

如示例程序所示，SetDO 指令将在第一条 MovL 指令执行到 20%时触发。第一条 MovL 指令执行完后直接过渡执行第二条 MovL 指令。

若删去 WaitFinish (20)，则 SerDO 将在第一条 MovL 指令执行完后执行，然后再执行第二条 MovL 指令。此情况下两条 MovL 指令间无过渡，且有停顿。

```
MovL (P1, V1000,"RELATIVE", C100)
WaitFinish (20)
SetDO (DO1, 1)
MovL (P2, V1000,"RELATIVE", C100)
```

WaitCondition – 等待指定判断条件指令

设置机器人执行等待的条件。满足条件则执行下一条指令。若在设定时间内未满足条件，则返回超时状态，继续执行下一条指令。

指令: WaitCondition 行号: 5	
工程名: project0 程序名: project1	
判断条件	1 > 0
时长(ms)	0
中断使能	0
超时判断值	L: INT0
超时跳转的标签名	

- 当“判别条件”为真时，才执行“下一步”指令，否则程序将继续等待直到表达式为真。
- “时长”表示执行等待时所需的时间，可为 int 常量或者 INT 变量类型，单位为 ms。
 - 如果该参数的值为 0，将强制等待判别条件为真时才继续执行下一条指令。
 - 如果该参数的值非 0，即使判别条件仍未为真，系统将在等待给定时长后跳过该指令，并继续执行下一条指令。
- 中断使能：设定等待过程中是否在程序恢复后继续计时。
 - 0：遇到程序停止时，停止计时，并在恢复时继续程序停止前的计时。
 - 1：遇到程序停止时，停止计时，并在恢复时重新计时。
- 超时判断值：选择一个变量并在如下两种情况下为其赋值。
 - 若输入条件为真且该指令执行完成时，对该变量赋值 0；
 - 当该指令由于超时执行完成时，对该变量赋值 1。
- 超时跳转的标签名：当执行超时时，选择跳转到标签名所在行。

WaitCondition("X> 0", 1000, 0, INT0)
等待时间 1 秒内，若 x> 0 满足，变量 INT0 赋值 0，执行下一条指令；若执行 0.5 秒，程序停止，恢复后，在 0.5 秒的基础上继续计时，1 秒时，条件未满足，变量 INT0 赋值 1，执行下一条指令。

6.4 IO 指令

IO 指令列表如图 6-30 所示。

图6-30 IO 指令列表

指令分类	指令列表
运动指令	SetDO
控制指令	SetAO
等待指令	SetSimDO
IO指令	SetSimAO
设置指令	WaitDI
位置运算指令	WaitAI
位运算指令	WaitSimDI
时钟指令	WaitSimAI
区域指令	PulseOut
视觉指令	PulseSimOut
码垛指令	GetDI8421

这些指令运用到输入输出模块的数字信号和模拟信号，数字与模拟信号经常与触发信息一起配合使用。

SetDO - 设置数字量输出指令

将数字量输出端口设置为 TRUE（1）或者 FALSE（0）状态，

指令: SetDO 行号: 5	
工程名: project0 程序名: project1	
端口变量	L: DO2
端口值	1

端口变量：带设置的数字量输出端口

端口值：要设置的数字量输出端口的值，可输入 0 或 1，分别代表该端口的低电平与高电平。

例如：

```
SetDO (DO1, 1)  
设置 DO1 变量为 1，即设置与 DO1 变量绑定的数字量输出端口为高电平
```

SetAO - 设置模拟量输出指令

将模拟量输出端口设置为一定值，

指令: SetAO 行号: 5	
工程名: project0 程序名: project1	
端口变量	L: AO0
端口值	5.0

例如：

```
SetAO (AO1, 5.0)  
设置 AO1 变量为 5.0，即设置与 AO1 变量绑定的模拟量输出端口电压为 5.0V。
```

SetSimDO - 设置虚拟数字量输出指令

将虚拟数字量输出端口设置为 TRUE（1）或者 FALSE（0）状态。

指令: SetSimDO 行号: 5	
工程名: project0 程序名: project1	
端口变量	L: SimDO0
端口值	1

例如：

```
SetSimDO (SimDO1, 1)  
设置 SimDO1 变量为 1，即设置与 SimDO1 变量绑定的虚拟数字量输出端口为 1。
```

SetSimAO - 设置虚拟模拟输出端口值

将虚拟模拟量输出端口设置为一定值。

指令: SetSimAO 行号: 5
工程名: project0 程序名: project1

端口变量	L: SimAO1	⌵
端口值	5.0	

确认 取消

例如:

```
SetSimAO (SimAO1, 5.0)
```

设置虚拟模拟量输出变量 SimAO1 的值为 5.0，即设置该变量绑定端口的输出值为 5.0。

WaitDI - 等待数字量输入指令

等待直到数字量输入端口被设置或复位。

指令: WaitDI 行号: 5
工程名: project0 程序名: project1

端口变量	L: DIO	⌵
端口值	0	
时长(ms)	0	⌵
中断使能	0	
超时判断值	L: INTO	⌵
超时跳转的标签名		⌵

确认 取消

- 端口变量: 需要等待的输入端口编号。
- 端口值: 等待的数字量输入端口电平。
 - 0: 低电平
 - 1: 高电平
- 时长(ms): 等待信号变换的时间，可为 int 常量或者 INT 变量类型。
- 中断使能: 设定等待过程中是否在程序恢复后继续计时。
 - 0: 遇到程序停止时，停止计时，并在恢复时继续程序停止前的计时。
 - 1: 遇到程序停止时，停止计时，并在恢复时重新计时。
- 超时判断值: 返回该指令执行结束后的状态
 - 0: 成功检测到信号
 - 1: 未检测到信号，指令超时返回。
- 超时跳转的标签名: 当执行超时时，选择跳转的标签名所在行。

示例

```
WaitDI (DI1, 1, 1000, 0, INT1)
```

等待数字量输入变量 DI1 为"1"的状态，等待 1000ms（若期间暂停程序，恢复运行后继续计时），若 1000ms 内未等到该信号，则将 INT1 置为 1，表示等待超时。

WaitDI8421 - 等待一组数字量输入指令

该指令用于在指定时长内等待一组连续数字量输入 DI 端口的状态组合。在设定时长内若等待条件满足，程序继续向下执行；若设定时长内条件未满足，则超时判断值置为 1 后，程序继续向下执行。

指令: WaitDI8421 行号: 5
工程名: project0 程序名: project1

起始DI端口	1	▲ ▼
结束DI端口	1	
起止端口8421值	0	
时长(ms)	0	
中断使能	0	
超时判断值	L: INTO	
超时跳转的标签名		

确认 取消

- 时长：等待该组数字量输入 DI 信号的时长，可为 int 常量或者 INT 变量类型，单位 ms。
- 中断使能：该参数用于设定在等待过程中若暂停程序，再启动后计时是否重新开始。
 - 0：遇到程序暂停时，停止计时，并在恢复运行时继续程序暂停之前的计时。
 - 1：遇到程序暂停时，停止计时，并在恢复运行时重新开始计时。
- 起始 DI 端口：该段连续 DI 端口的起始端口号。
- 结束 DI 端口：该段连续 DI 端口的结束端口号。
- 起始端口 8421 值：将该段连续 DI 端口值转换成十进制数，若与 VALUE 值相等则视为满足条件。
- 超时判断值：返回该指令执行结束后的状态。
 - 0：成功检测到信号
 - 1：未检测到信号，指令超时返回。
- 超时跳转的标签名：当执行超时时，选择跳转的标签名所在行。

```
WaitDI8421(1000, 1, 9, 16, 255,INT1,"Gotoggg")  
/*在 1000ms 时长内等待 9 到 16 号 DI 口状态为 11111111 时，程序继续向下执行，若 1000ms 内未等到条件满足，则将 INT1 置 1。若等待过程中暂停程序，程序恢复运行后计时将重新开始。*/
```

WaitAI - 等待模拟量输入指令

等待直到模拟量输入端口值与给定值相等，参数说明与 WaitDI 相同。

指令: WaitAI 行号: 5
工程名: project0 程序名: project1

端口变量	L: AI0	▼ ▼ ▼ ▼ ▼
端口值	0	
时长(ms)	1000	
中断使能	0	
超时判断值	L: INTO	
超时跳转的标签名		

确认 取消

示例

```
WaitAI (AI1 , 5.0 , 1000 , 1 , INT1)  
等待模拟量输入变量 AI1 为"5.0"的状态，等待 1000ms（若期间暂停程序，恢复运行后重新开始计时），若 1000ms 内未等到该信号，则将 INT1 置为 1，表示等待超时。
```

WaitSimDI - 等待直至已设置虚拟数字输入信号输入信号

等待直到虚拟数字量输入端口被设置或复位，参数说明与 WaitDI 相同。

新建指令

指令: WaitSimDI 行号: 5
工程名: project0 程序名: project1

端口变量	L: SimDI0	⌵
端口值	0	
时长(ms)	1000	⌵
中断使能	0	
超时判断值	L: INT0	⌵
超时跳转的标签名		⌵

确认 取消

WaitSimDI (SimDI1, 1, 1000, 0, INT1)
等待数字量输入变量 SimDI1 为"1"的状态，等待 1000ms（若期间暂停程序，恢复运行后继续计时），若 1000ms 内未等到该信号，则将 INT1 置为 1，表示等待超时。

WaitSimDI8421 - 等待一组虚拟数字量输入指令

用于在指定时长内等待一组连续虚拟数字量输入 SimDI 端口的状态组合，参数说明与 WaitDI8421 相同。

指令: WaitSimDI8421 行号: 5
工程名: project0 程序名: project1

起始SIMDI端口	1	▲
结束SIMDI端口	1	
起止端口8421值	0	
时长(ms)	0	⌵
中断使能	0	
超时判断值	L: INT0	⌵
超时跳转的标签名		⌵

确认 取消

WaitSimAI - 等待虚拟模拟量输入指令

等待直到虚拟模拟量输入端口值与给定值相等，参数说明与 WaitAI 相同。

指令: WaitSimAI 行号: 5
工程名: project0 程序名: project1

端口变量	L: SimAI0	⌵
端口值	0	
时长(ms)	0	⌵
中断使能	0	
超时判断值	L: INT0	⌵
超时跳转的标签名		⌵

确认 取消

WaitSimAI (SimAI1, 5.0 , 1000, 0, INT1)
等待虚拟模拟量输入变量 SimAI1 为"5.0"的状态，等待 1000ms（若期间暂停程序，恢复运行后继续计时），若 1000ms 内未等到该信号，则将 INT1 置为 1，表示等待超时。

PulseOut - 设置脉冲输出指令

设定 IO 变量输出的脉冲时间与条件。

指令: PulseOut 行号: 5
工程名: project0 程序名: project1

端口变量	L: DO2
端口值	0
持续时间(ms)	0
中断使能	0

确认 取消

- 端口变量：需要设定的端口编号。
- 端口值：设定输出的脉冲电平。
 - 0：低电平
 - 1：高电平
- 持续时间：设定脉冲输出的时间。
- 中断使能：设定脉冲输出是否在中断后继续输出。
 - 0：遇到程序停止时，脉冲会持续输出到设定时间后停止输出，即不受程序停止而停止其功能。
 - 1：遇到程序停止时，停止输出，当程序继续运行时才会继续输出，即与程序运行状态同步，程序停则输出暂停，程序运行则输出继续。

PulseOut (DO1, 1, 500, 1)
在数字量输出变量 DO1 绑定端口输出 500ms 的脉冲信号，脉冲幅值为 1，若脉冲输出期间暂停程序，则端口停止输出，在程序恢复启动后恢复脉冲输出。

PulseSimOut - 设置 SimIO 输出信号的脉冲时间和条件

设定 SimIO 变量输出的脉冲时间与条件。

指令: PulseSimOut 行号: 5
工程名: project0 程序名: project1

端口变量	L: SimDO0
端口值	0
持续时间(ms)	0
中断使能	0

确认 取消

- 端口变量：需要设定的端口编号。
- 端口值：设定输出的脉冲电平。
 - 0：低电平

- 1: 高电平
- 持续时间：设定脉冲输出的时间。
- 中断使能：设定脉冲输出是否在中断后继续输出。
 - 0: 遇到程序停止时，脉冲会持续输出到设定时间后停止输出，即不受程序停止而停止其功能。
 - 1: 遇到程序停止时，停止输出，当程序继续运行时才会继续输出，即与程序运行状态同步，程序停则输出暂停，程序运行则输出继续。

```
PulseSimOut (SimDO1, 1, 500, 1)
```

在虚拟数字量输出变量 SimDO1 绑定端口输出 500ms 的脉冲信号，脉冲幅值为 1，若脉冲输出期间暂停程序，则端口停止输出，在程序恢复启动后恢复脉冲输出。

GetDI8421 - 获取一段连续的数字量输入值指令

获取一段连续的 DI 口状态（将其看成一段二进制数据）并以十进制数返回。

指令: GetDI8421 行号: 5
工程名: project0 程序名: project1

起始DI端口	1
结束DI端口	1
返回值	L: INT0

确认 取消

- 起始 DI 端口：想要获取的起始 DI 端口号。
- 结束 DI 端口：想要获取的结束 DI 端口号。
- 返回值：将获取到的端口状态值存储到传入的 int 变量中。

```
GetDI8421 (9, 16, INT1)
```

将第 9 到第 16 号数字量输入端口的状态视作二进制数，将其转化为十进制数后返回给变量 INT1。

GetSimDI8421 - 获取一段连续虚拟数字量输入值指令

获取一段连续的虚拟数字量输入 SimDI 端口状态（将其看成一段二进制数据）并以十进制数返回。

指令: GetSimDI8421 行号: 5
工程名: project0 程序名: project1

起始DI端口	1
结束DI端口	1
返回值	L: INT0

确认 取消

- 起始 DI 端口：想要获取的起始 SimDI 端口号。
- 结束 DI 端口：想要获取的结束 SimDI 端口号。
- 返回值：将获取到的端口状态值存储到传入的 int 变量中。

```
GetSimDI8421 (9, 16, INT1)
```

/*将第 9 到第 16 号虚拟数字量输入端口的状态视作二进制数，将其转化为十进制数后返回给变量 INT1*/

SetDO8421 - 设置一段连续数字量输出端口状态

设置一段连续的 DO 口状态（将其看成一段二进制数据），将传入的十进制数转换成二进制数设置到指定的 DO 口上。

指令：SetDO8421 行号：5

工程名：project0 程序名：project1

起始DO端口	9
结束DO端口	9
设定值	0

确认

取消

- 起始 DO 端口：想要设置的起始 DO 端口号。
- 结束 DO 端口：想要设置的结束 DO 端口号。
- 设定值：想要端口输出的十进制设定值。

以下述指令为例，3 转换成二进制是 0011,那 9 号跟 10 号会分别输出高电平。为 2 代表 9 号低电平，10 号设为高电平。

```
GetSimDI8421 (9, 16, INT1)  
将第 9 到第 16 号虚拟数字量输入端口的状态视作二进制数，将其转化为十进制数后返回给变量 INT1。
```

SetSimDO8421 - 设置一段连续虚拟数字输出量输出指令

设置一段连续的 SimDO 口状态（将其看成一段二进制数据），将传入的十进制数转换成二进制数设置到指定的 SimDO 口上。

新建指令

指令：SetSimDO8421 行号：5

工程名：project0 程序名：project1

起始DO端口	1
结束DO端口	1
设定值	0

确认

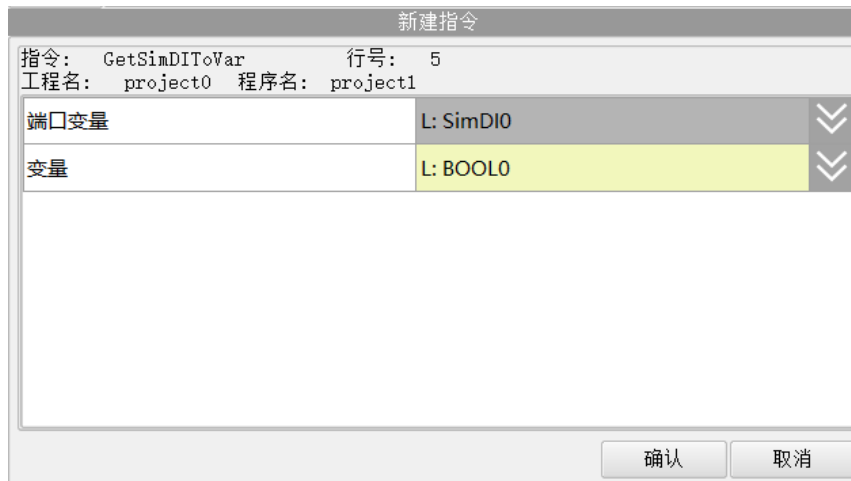
取消

- 起始 DO 端口：想要获取的起始 SimDO 端口号。
- 结束 DO 端口：想要获取的结束 SimDO 端口号。
- 设定值：想要端口输出的十进制设定值。

```
SetSimDO8421 (9, 16, 255)  
/*将第 9 到第 16 号虚拟数字量输出端口置为 1*/
```

GetSimDIToVar - 将虚拟数字量输入值报讯到指定的变量

将虚拟数字量输入信号映射到某个变量上。



- 端口变量：获取虚拟数字量输入信号值的变量，类型为虚拟数字量输入类型变量。
- 变量：被获取的数字量信号值赋值的变量，类型为 BOOL 类型。

例如，将虚拟数字量输入类型变量 SIMDI1 的值赋值给布尔型变量 BOOL0 的指令如下：

```
GetSimDIToVar (SimDI9, BOOL1)
```

获取与变量 SimDI9 绑定的虚拟数字量输入端口的值，并将该值赋予 BOOL1 变量。

SetSimDOByVar - 将布尔类型的值映射到数字输出信号上

将某个布尔型变量的值映射到一个数字量输出变量上。

- 端口变量：数字量输出类型变量，接收变量的值并输出。
- 变量：布尔型变量，将其值输出给端口变量。

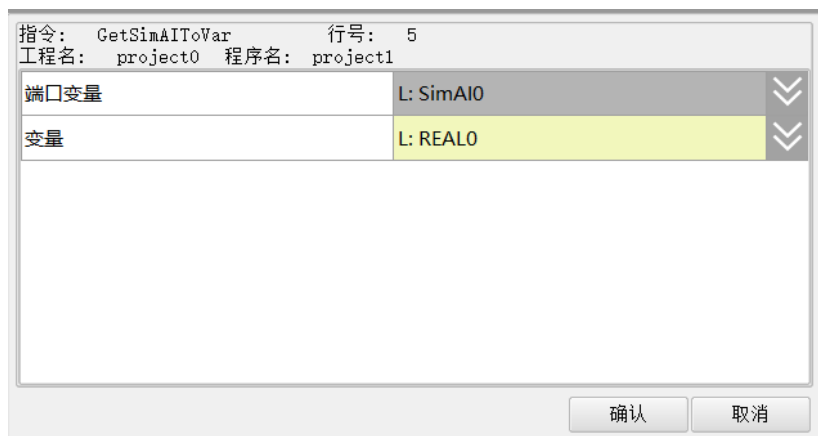
例如，将布尔型变量 BOOL0 的值输出给虚拟数字量输出型变量 SimDO1 的指令如下：

```
SetSimDOByVar (SimDO9, BOOL1)
```

将变量 BOOL1 的状态在虚拟数字量变量 SimDO9 对应端口输出。

GetSimAIToVar - 将虚拟模拟输入信号映射到变量上

将虚拟模拟量输入信号映射到某个变量上。



- 端口变量：获取虚拟模拟量输入信号值的变量，类型为虚拟模拟量输入类型变量。
- 变量：被获取的模拟量信号值赋值的变量，类型为 REAL 类型。

例如，将虚拟模拟量输入类型变量 SIMAI1 的值赋值给实数型变量 REAL0 的指令如下：

```
GetSimAIToVar (SimAI9, REAL1)
```

获取与变量 SimAI9 绑定的虚拟模拟量输入端口的值，并将该值赋予 REAL1 变量。

SetSimAOByVar - 将指定变量值设置到虚拟量输出指令

将某个 REAL 型变量的值映射到一个虚拟模拟量输出变量上。

指令: SetSimAOByVar 行号: 5
工程名: project0 程序名: project1

端口变量	L: SimAO1	⌵
变量	L: REAL0	⌵

确认 取消

- 端口变量：虚拟模拟量输出类型变量，接收变量的值并输出。
- 变量：REAL 型变量，将其值输出给端口变量。

例如，将实数类型变量 REAL0 的值输出给虚拟模拟量输出型变量 SimAO0 的指令如下：

```
SetSimAOByVar (SimAO9, REAL1)  
将 REAL1 变量的值在与变量 SimAO9 绑定的虚拟模拟量输出端口输出。
```

SetDIEdge - 设置数字量输入变量的边沿信号状态指令

通过指令强制设置某个数字量输入端口的边沿信号状态。

指令: SetDIEdge 行号: 5
工程名: project0 程序名: project1

端口变量	L: DI0	⌵
边沿类型	"riseEdge"	⌵
边沿值	0	

确认 取消

- 端口变量：数字量输入类型变量，设置该端口的上升沿或下降沿状态。
- 边沿类型：选择要设置端口的边沿信号状态。
 - riseEdge：上升沿状态。
 - downEdge：下降沿状态。
- 边沿值：需要设置的边沿状态值。

例如，将 DI1 端口的上升沿状态设置为 0 的指令如下：

```
SetDIEdge (DI9, "riseEdge", 1)  
强制 DI9 端口的上升沿信号为"1"状态。
```

SetSimDIEdge - 设置虚拟数字量输入变量的边沿信号状态

通过指令强制设置某个虚拟数字量输入端口的边沿信号状态。

指令: SetSimDIEdge 行号: 5
工程名: project0 程序名: project1

端口变量	L: SimDI0	⌵
边沿类型	"riseEdge"	⌵
边沿值	0	

确认 取消

- 端口变量: 虚拟数字量输入类型变量, 设置该端口的上升沿或下降沿状态。
- 边沿类型: 选择要设置端口的边沿信号状态。
 - riseEdge: 上升沿状态。
 - downEdge: 下降沿状态。
- 边沿值: 需要设置的边沿状态值。

例如, 将 SimDI1 端口的上升沿状态设置为 0 的指令如下:

```
SetSimDIEdge (SimDI9, "riseEdge", 1)  
强制 SimDI9 端口的上升沿信号为"1"状态。
```

6.5 设置指令

设置指令列表如图 6-31 所示。

图6-31 设置指令列表

指令分类	指令列表
运动指令	SetTool
控制指令	SetCoord
等待指令	SetExternalTCP
IO指令	SetPositioner
设置指令	SetJointDyn
位置运算指令	SetCartDyn
传送带跟随指令	SetBlendParam
位运算指令	SetOverRide
时钟指令	SetPayload

SetTool - 设置工具参数指令

选择工具坐标系指令设置工具坐标系的方法请参见 4.3.4 小节的描述。例如：

```
SetTool (TOOL0)
```

SetCoord - 设置用户坐标系指令

选择用户坐标系指令。设置用户坐标系的方法请参见 4.3.5 小节的描述。例如：

```
SetCoord (USERCOORD0)
```

SetExternalTCP - 设置外部 TCP 坐标系指令

该指令用于设置系统的参考坐标系为外部工具坐标系。执行该指令后，系统将参考坐标系切换至外部工具坐标系，机器人的运动将相对于外部工具坐标系。例如：

```
SetExternalTCP (EXTTCP0)
/*选择 EXTTCP0 为机器人的外部 TCP 坐标系参数*/
```

SetPositioner - 设置变位机坐标系指令

该指令用于设置系统的参考坐标系为变位机坐标系。执行该指令后，系统将参考坐标系切换至变位机坐标系，机器人的运动将相对于变位机坐标系。例如：

```
SetPositioner (POSITIONER0)
/*选择 POSITIONER0 为变位机坐标系参数*/
```



说明

POSITIONER0 是变位机变量，其详细说明请参见 5.3 系统数据类型 POSITIONER

SetPayload - 设置负载重量指令

选择工件负载参数指令。例如：

```
SetPayload (PAYLOAD0)
运行该指令后，系统会将变量 PAYLOAD0 中参数作为当前工件负载参数，后续运动指令均以该质量运行。
```

SetJointDyn - 设置关节运动参数指令

该指令用于设置关节动态参数，包括加速度、减速度、加加速度、力矩。运行该指令后，后续运动指令均以该动态参数运行。

指令: SetJointDyn 行号: 5
工程名: project0 程序名: project1

加速度(%)	100
减速度(%)	100
加加速度(%)	100
力矩(%)	100

确认取消

- 加速度：关节最大加速度的百分比。
- 减速度：关节最大减速度的百分比。
- 加加速度：关节最大加加速度的百分比。
- 力矩：关节最大力矩的百分比。

```
SetJointDyn(80, 80, 80, 60)  
/*运行该指令后，后续关节空间的运动指令均以该动态参数运行。*/
```

SetCartDyn - 设置直线或圆弧运动动态参数指令

该指令用于设置笛卡尔空间的动态参数，运行该指令后，后续运动指令均以该动态参数运行。

指令: SetCartDyn 行号: 5
工程名: project0 程序名: project1

加速度(%)	100
减速度(%)	100
加加速度(%)	100
姿态加速度(%)	100
姿态减速度(%)	100
姿态加加速度(%)	100
力矩(%)	100

确认取消

- 加速度：直线运动最大加速度的百分比。
- 减速度：直线运动最大减速度的百分比。
- 加加速度：直线运动最大加加速度的百分比。
- 姿态加速度：姿态最大旋转加速度的百分比。
- 姿态减速度：姿态最大旋转减速度的百分比。
- 姿态加加速度：姿态最大旋转加加速度的百分比。
- 力矩：关节最大力矩的百分比

```
SetCartDyn(80, 80, 80, 80, 80, 80, 60)  
/*运行该指令后，后续笛卡尔空间的运动指令均以该动态参数运行。*/
```

SetBlendParam - 设置过渡参数

该指令用于设置过渡参数。运行该指令后，后续运动指令均以该过渡参数运行。对指令参数进行说明：

指令: SetBlendParam 行号: 5
工程名: project0 程序名: project1

过渡类型	"RELATIVE"	⌵
过渡值	DEFAULT	⌵

确认 取消

- 过渡类型 (BlendType)

数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - RELATIVE: 相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE: 绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值 (BlendValue)

数据类型: ZONE
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省，默认值为 C100。

示例:

```
SetBlendParam("RELATIVE",C20)
```

运行该指令后，后续运动指令均以该过渡参数运行。

SetOverride - 设置全局速度

全局速度设置功能。设置“全局速度”为一个合适的百分比数值。

指令: SetOverride 行号: 5
工程名: project0 程序名: project1

全局速度(%)	20	⌵
---------	----	---

确认 取消

例如:

```
SetOverride (20)  
SetOverride (INT0) // 将速度倍率设为 INT0 变量中的值
```

SetSingularPass - 设置跨腕部奇异功能指令

设置跨腕部奇异功能开关指令，可选择“OPEN”（开）和“CLOSE”（关）。例如：

```
SetSingularPass("OPEN") /*开启跨腕部奇异功能*/
```

SetColliEnable - 设置碰撞检测使能指令

设置碰撞检测使能指令。

轴号(或轴组)	"GROUP"	⌵
有效性	"ENABLE"	⌵

轴号参数： "GROUP、A1、A2、A3、A4、A5、A6"。当轴号为"GROUP"时，表示所有轴；

使能参数："ENABLE（开）、DISABLE（关）"

```
SetColliEnable("GROUP","ENABLE") /*开启所有轴的碰撞检测使能*/
```

SetAxisColliParam - 设置碰撞检测灵敏度指令

设置碰撞检测灵敏度值指令。

轴号	"A1"	⌵
灵敏度值(%)	100	

轴号参数："A1、A2、A3、A4、A5、A6"；

灵敏度参数：单位为百分比，参数范围为[1,500]。

```
SetAxisColliParam ("A1",200) /*设置轴 1 的碰撞检测灵敏度值为 200*/
```

GetCurOverRide - 获取当前全局速度指令

获取当前全局速度功能。获取“全局速度”为一个合适的百分比数值，输出到 int 变量中。

指令: GetCurOverRide 行号: 5
工程名: project0 程序名: project1

全局速度(%)	L: INT0	⌵
---------	---------	---

确认 取消

例如：

```
GetCurOverRide (INT0)  
运行该指令后，当前的全局速度值将会被读取到 INT0 中，在变量列表可查看到该值。
```

Hand - 设置左右手系

该指令用于设置左右手系，仅适用于 SCARA 机型。设定“左右手设置”的参数值为“Lefty”、“Righty”或“Default”。

指令: Hand 行号: 6
工程名: project0 程序名: project1

左右手设置

"Lefty"

确认

取消

例如:

```
Hand ("Lefty")  
/*执行该指令后, 后续运动指令均以左手设置运行。*/
```

CalcTool - 计算工具参数

该指令根据设定的四个“标定点”来计算工具参数, 并将其保存至“工具参数”所设定的坐标系。

标定 _点 1	 L: P27	
标定 _点 2	 L: P28	
标定 _点 3	 L: P29	
标定 _点 4	 L: P30	
工具参数	G: TOOL1	

例如:

```
CalcTool (P1, P2, P3, P4, TOOL1)  
/*执行该指令后, 可根据所设定的四个标定点来计算工具参数, 并保存至 TOOL1*/
```

CalcCoord - 计算用户坐标参数

该指令根据设定的“原点”、“RX 方向标定_点”和“RXRY 平面标定_点”来计算用户坐标参数, 并将其保存至“用户参数”所设定的坐标系。

原点	 L: P27	
RX _{方向} 标定 _点	 L: P28	
RXRY _{平面} 标定 _点	 L: P29	
用户参数	G: USERCOORD1	

例如:

```
CalcCoord (P1, P2, P3, USERCOORD1)  
/*执行该指令后, 可根据所设定的三个标定点来计算用户坐标参数, 并保存至 USERCOORD1*/
```

SetRtToErr - 设置报警指令

设置机器人停止运行并发出错误信息。

指令: SetRtToErr 行号: 5
工程名: project 程序名: project1

错误信息	
错误编号	90001

确认 取消

错误信息：设置错误信息，支持中英文输入。

错误编号：设置错误编号。

例如：

```
SetRtToErr ("halt", 90001)
```

设置机器人停止运行并发出错误编号为 90001、错误信息为“halt”的错误。

SetRtInfo - 设置提示信息

设置提示信息但不影响机器人的运行。

指令: SetRtInfo 行号: 6
工程名: project0 程序名: project1

提示信息	
提示编号	70001

确认 取消

例如：

```
SetRtInfo ("Info", 70001)
```

设置机器人停止运行并发出提示编号为 70001、提示信息为“Info”的错误。

SetRtWarning - 设置报警信息

设置警告信息但不影响机器人的运行。

指令: SetRtWarning 行号: 6
工程名: project0 程序名: project1

警告信息	
警告编号	80001

确认 取消

例如:

```
SetRtWarning ("Warning", 80001)  
设置机器人停止运行并发出警告编号为 80001、警告信息为"Warning"的错误。
```

Stop - 停止运行指令

该命令用于停止所有激活程序的执行。

```
Stop ()
```

RefRobotAxis - 单轴回零指令

设置机器人单轴回零的指令,该指令仅能在管理人员权限下创建。

指令: RefRobotAxis 行号: 5
工程名: project0 程序名: project1

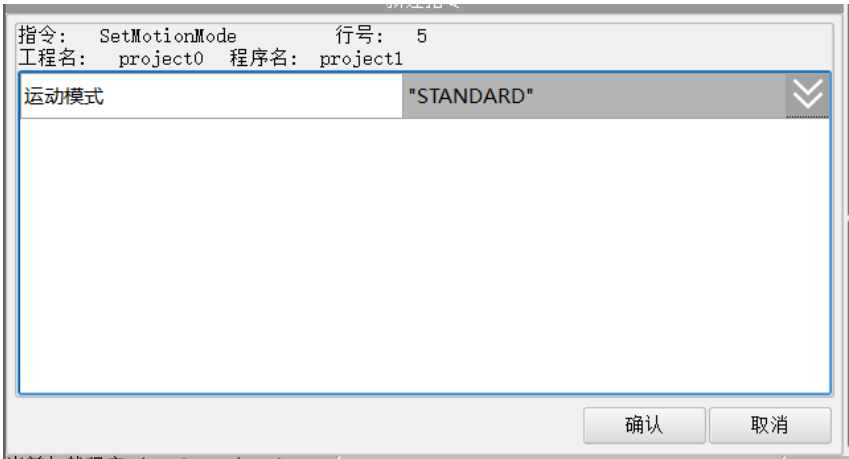
轴号	1
单圈值	0

确认 取消

```
RefRobotAxis (1,500)  
/*执行该指令后,将机器人 1 轴伺服在单圈值为 500 处设置为其零点*/
```

SetMotionMode - 设置运行模式

该指令仅能在管理人员权限下创建。设置运行模式。



运动模式：“STANDARD”（标准）；

“LOW_OSCILLATION”（低振动）

例如：

```
SetMotionMode ("STANDARD")  
将机器人当前的运动模式设置为标准运动模式。
```

AutoGainEnable – 设置伺服参数增益指令

伺服参数自调节开启指令，执行该指令后系统会根据传入的 LsScale 与 LsThresh 参数对后续执行的运动指令进行低速优化处理，详细使用方法可查看“7.24 伺服参数自调节功能”。

```
AutoGainEnable (LsScale0,LsThresh0)  
/*伺服参数自调节功能开启*/
```

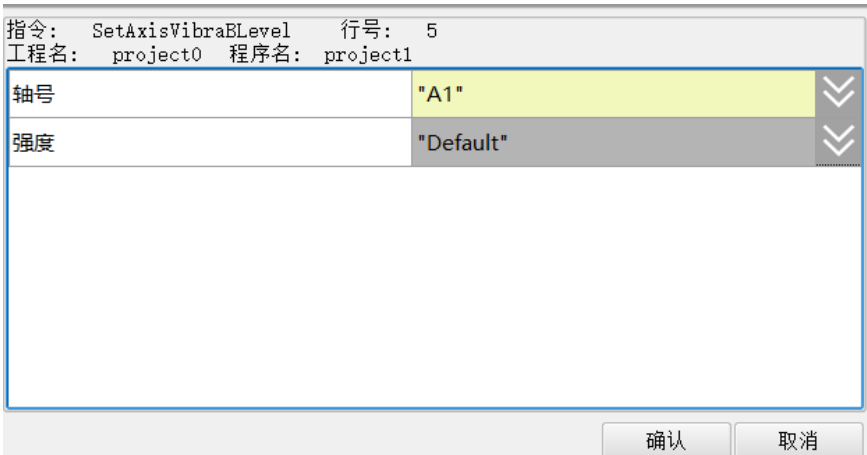
AutoGainDisable – 关闭伺服参数增益指令

伺服参数自调节结束指令，在执行伺服参数自调节开启指令后，执行该指令可关闭伺服参数自调节功能。例如：

```
AutoGainDisable()  
/*伺服参数自调节功能关闭*/
```

SetAxisVibraBLevel - 设置振动抑制等级指令

振动抑制等级设置命令，只对输入整形有效。



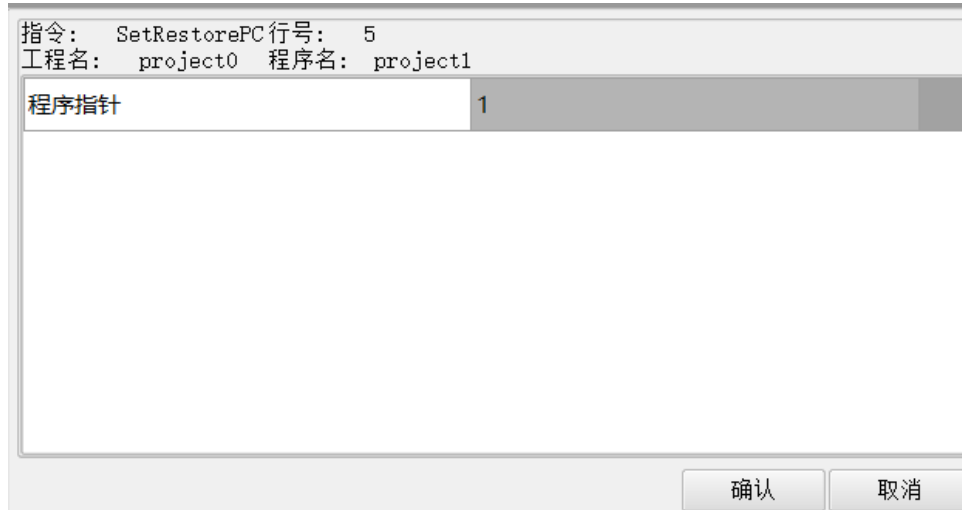
- 轴号：将要设置振动抑制等级的轴号。（A1/A2/A3/A4/A5/A6）
- 等级：将要设置振动抑制的等级。（Highest/High/Default/Low/Lowest）

示例：

```
SetAxisVibraBLevel(A1,High)
/*将 1 轴的振动抑制等级设为中高。*/
```

SetRestorePC - 设置恢复运行时的 PC 指令

该指令可设置 run 程序由暂停/停止转到运行时的 pc 指针。请注意，该指令只在 run 程序中使用有效。对指令参数进行说明：



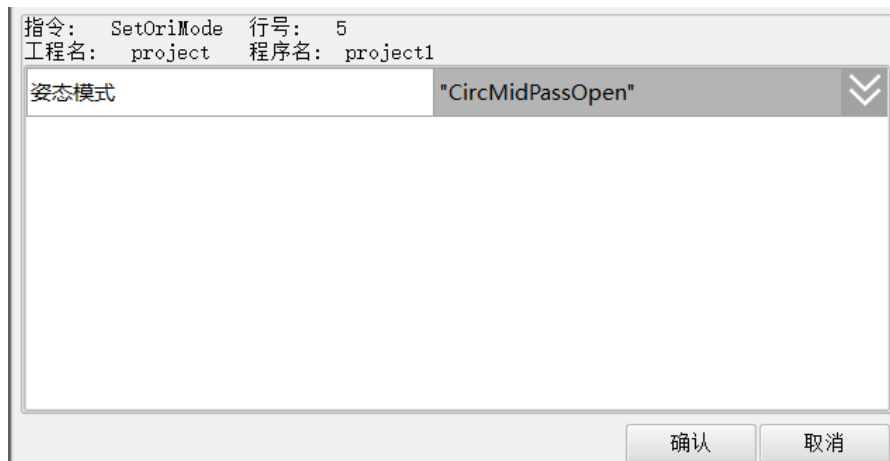
- 程序指针：run 程序恢复运行时的行号。

示例：

```
SetRestorePC(3)
/*待 run 程序由暂停/停止转到运行状态时，程序指针将从第 3 行开始运行。*/
```

SetOriMode - 设置姿态模式指令

图6-32 SetOriMode 配置界面图

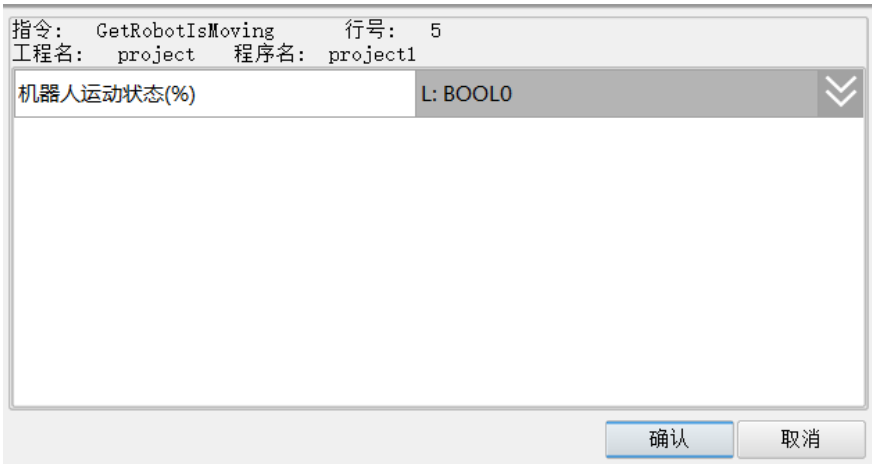


姿态模式：机器人姿态插补模式。目前仅支持圆弧过中间辅助点姿态模式，即“CircMidPassOpen”和“CircMidPassClose”；

- CircMidPassOpen：打开圆弧过中间辅助点姿态插补的效果。
- CircMidPassClose：关闭圆弧过中间辅助点姿态插补的效果。

GetRobIsMoving - 获取当前机器人的运动状态指令

用于获取当前机器人运动状态。



- 机器人运动状态 (%)：设置指令，用户可以获取当前机器人是否处于运动中。

SetColliAutoTune – 碰撞检测参数自调整指令

用于设置碰撞检测参数自调整。



- 碰撞检测参数自调整阶段：
 - Start（重新开始）
 - Pause（暂停）
 - Continue（继续）
 - End（结束）
 - Calculate（计算生成）

 说明

1. 当碰撞检测参数自调整阶段选择 Start、Pause、Continue、End 时，将对自调整功能的数据记录状态进行设置
2. 若碰撞检测自调整阶段参数设置为 “Calculate” 则根据自调整功能记录的数据、检测方式、灵敏度进行计算得到碰撞自调整参数并将结果存储在碰撞参数对应的 ID 中，如下图所示：

碰撞检测参数自调整阶段	"Calculate"	⌵
检测方式	"Default"	⌵
灵敏度	"Default"	⌵
碰撞参数	G: COLLIPARAM0	⌵

- 碰撞检测参数自调整阶段：自调整参数 “Calculate（计算生成）”。
- 检测方式：自调整使用的碰撞检测方法
 - Default（默认）
 - Default+（强）
 - Default++（更强）

【备注】：该参数仅在碰撞检测参数自调整阶段设为 Calculate 时才有效。

- 灵敏度：自调整结果的灵敏度等级
 - Highest（最高）
 - High（高）
 - Default（默认）
 - low（低）
 - lowest（最低）

【备注】：该参数仅在碰撞检测参数自调整阶段设为 Calculate 时才有效。

- 碰撞参数：存储自调整计算结果的碰撞参数。全局变量中的 COLLIPARAM 型变量，一个变量对应一个 ID，自调整结果存储在选择变量对应的 ID 中。

SetColliParam - 碰撞参数设置指令

该指令用于设置碰撞检测参数。

新建指令

指令： SetColliParam 行号： 2

工程名： test 程序名： main

碰撞参数	DEFAULT	⌵
轴号(或轴组)	"DEFAULT"	⌵

确认 取消

- 碰撞参数：碰撞检测功能使用的参数，当需要使用自调整参数时，选择 COLLIPARAM 变量类型中的变量（对应实际需要设置的 ID），否则选择 DEFAULT（出厂默认值）。

- 轴号（或轴组）：当需要设置碰撞检测“有效性”或“灵敏度值”时选择单个轴号或所有轴（GROUP），默认选择 DEFAULT。

DEFAULT：默认值，无意义

GROUP：所有轴

A1：A1 轴

A2：A2 轴

A3：A3 轴

A4：A4 轴

A5：A5 轴

A6：A6 轴
- 有效性：控制对应轴号碰撞检测功能的开启与关闭。

ENABLE（开）

DISABLE（关）

【备注】：该参数仅在轴号设为非 DEFAULT 时才有效。
- 灵敏度值（%）：碰撞检测的灵敏度值，单位为百分比，参数范围为[1,500]。

【备注】：该参数仅在轴号设为非 DEFAULT 时才有效。

 说明

- 仅支持机器人本体轴。
- 设置较强的检测方法和较高的灵敏度等级会提高碰撞检测的灵敏度但同时可能增大误报警的概率。

6.6 位置运算指令

位置运算指令列表如图 6-33 所示。

图6-33 位置运算指令列表

指令分类	指令列表
运动指令	GetCurAPos
控制指令	GetCurCPos
等待指令	APosToCPos
IO指令	CPosToAPos
设置指令	CPosToCPos
位置运算指令	ToolOffset
传送带跟随指令	UserOffset
位运算指令	
时钟指令	

- 姿态模式：机器人姿态插补模式。目前仅支持圆弧过中间辅助点姿态模式，即“CircMidPassOpen”和“CircMidPassClose”。
 - “CircMidPassOpen”：打开圆弧通过中间辅助点姿态插补的效果。
 - “CircMidPassClose”：关闭圆弧通过中间辅助点姿态插补的效果。

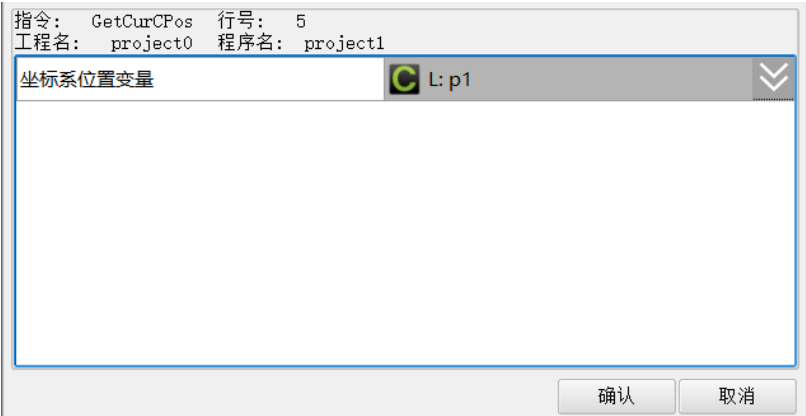
GetCurAPos - 获取当前关节位置指令

该指令用于获取当前关节坐标系下的位置，赋值给 Apos 类型变量。

GetCurAPos (P1)
获取机器人关节坐标系下的当前位置，并保存于 P1 中

GetCurCPos - 获取当前笛卡尔坐标系位置指令

该指令用于获取机器人当前坐标系（世界坐标系或用户坐标系）下的笛卡尔空间位置，赋值给 Cpos 类型变量。

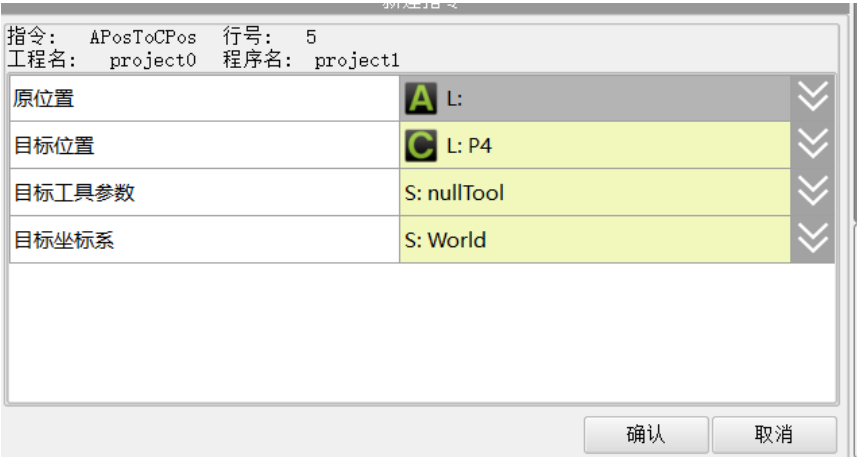


GetCurCPos (P1)
获取机器人当前坐标系（世界坐标系或用户坐标系）下的笛卡尔空间位置，并保存于 P1 中。

APosToCPos - 机器人位置转换指令（Apos 转换成 Cpos）

机器人位置点转换指令，通过该指令可以将 APos 点转换为 CPos 点。

给定 APos 点，以及要转换的目标 CPos 点的参考坐标系及工具参数，可以得到目标 CPos 点的值。



原位置：转换前关节位置。

目标位置：目标笛卡尔空间位置。

目标工具参数：目标工具参数

目标坐标系：参考坐标系

APosToCPos (P1, P2, TOOL1, USERCOORD1)
根据 P1、TOOL1、USERCOORD1，可以得到目标点 P2。

CPosToAPos - 机器人位置转换指令（Cpos 转换成 Apos）

机器人位置点转换指令，通过该指令可以将 CPos 点转换为 APos 点。

给定 CPos 点及其所在的参考坐标系及工具参数，可以得到目标 APos 点的值。

指令: CPosToAPos 行号: 5	
工程名: project0 程序名: project1	
原位置	 L:
原工具参数	G: TOOL0
原坐标系	 S: World
目标位置	 L: P4
确认 取消	

原位置：转换前的笛卡尔空间位置。

原工具参数：转换前的工具参数。

原坐标系：转换前的参考坐标系参数。

目标位置：转换后的关节位置。

APosToCPos (P1, TOOL1, USERCOORD1, P2)
根据 P1、TOOL1、USERCOORD1，可以得到目标点 P2。

CPosToCPos - 机器人位置转换指令（Cpos 转换成 Cpos）

机器人位置点转换指令，通过该指令可以将 CPos 点转换为 CPos 点。

给定 CPos 点及其所在的参考坐标系及工具参数，以及要转换的目标 CPos 点的参考坐标系及工具参数，可以得到目标 CPos 点的值。

CPosToCPos (P1, TOOL1, USERCOORD1, P2, TOOL2, USERCOORD2)
根据 P1、TOOL1、USERCOORD1、TOOL2、USERCOORD2，可以得到目标点 P2。

ToolOffset - 工具坐标系偏移指令

机器人工具坐标系偏移指令，通过该指令可对基准工具坐标系进行旋转或者偏移后，生成一个新的工具坐标系。

给定基准工具坐标系 TOOL，以及需要偏移或旋转的偏移量，可以得到目标工具坐标系的值。

ToolOffset (TOOL1, DCPOS1, TOOL2)
根据工具坐标系 TOOL1、偏移量 DCPOS1，可以得到偏移后的工具坐标系 TOOL2。

UserOffset - 用户坐标系偏移指令

机器人用户坐标系偏移指令，通过该指令可对基准用户坐标系进行旋转或者偏移后，生成一个新的用户坐标系。

给定基准用户坐标系 USERCOORD，以及需要偏移或旋转的偏移量，可以得到目标用户坐标系的值

UserOffset (USERCOORD1, DCPOS1, USERCOORD2)

CalPosOffset - 计算位置点偏移结果指令

计算位置点偏移结果指令，该指令由原位置量和相对偏移量为基础，计算出原位置相对坐标系或工具所偏移后的位置值。

CalPosOffset 指令可以将一个位置点（APOS 或 CPOS）进行相对坐标系或工具偏移，偏移量为 DCPOS，最终获得一个目标位置值，该目标位置值的类型是 CPOS。

图6-34 CalPosOffset 指令的编辑界面

指令: CalPosOffset 行号: 5
工程名: project 程序名: project1

原位置	L:
相对偏移量	L:
参考坐标系	"COORD"
工具参数	S: nullTool
坐标系参数	S: World
目标位置	L: P2

确认 取消

- 原位置：偏移运动的起点（APOS 或 CPOS）；
- 相对偏移量：相对原位置所要移动的位置增量（DCPOS）；
- 参考坐标系：坐标系偏移或工具偏移选择。其中，“Coord”表示相对于当前用户坐标系偏移；“Tool”表示相对于工具坐标系偏移，即参考 Tx、Ty、Tz 平移或者旋转；
- 工具参数：原位置和目标位置所属的工具信息；
- 坐标系参数：原位置和目标位置所属的坐标系信息；
- 目标位置：偏移后的最终位置（CPOS）。

6.7 传送带跟随指令

传送带跟随指令列表如图 6-35 所示。

图6-35 传送带跟随指令列表

指令分类	指令列表
运动指令	WaitWObj
控制指令	MovLSync
等待指令	MovJSyncQuit
IO指令	MovLSyncQuit
设置指令	SetTargetPos
位置运算指令	SetSyncoord
传送带跟随指令	SynCToUserC
位运算指令	WaitConvDis
时钟指令	

WaitWObj - 等待视觉信息输入指令

等待视觉信息指令，等待视觉发来的工件位置信息。

图6-36 WaitWObj 参数编辑界面

动坐标系	G: SYNCOORD0	⌵
工件id	L: INT0	⌵
时长(ms)	1000	⌵
超时判断值	L: INT1	⌵
超时跳转的标签名	Start	⌵

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。
- 工件 id：INT 类型变量，用来返回当前接收到的工件 id 号。
- 时长：表示执行等待时所需的时间，单位为 ms。
 - 如果该参数的值为 0，将强制等待，直至收到视觉发来的工件位置信息数据，才继续执行下一条指令。
 - 如果该参数的值非 0，即使未收到任何数据，系统将在等待给定时长后跳过该指令，并继续执行下一条指令。
- 超时判断值：选择一个变量并在如下两种情况下为其赋值。
 - 当该指令在等待时间内执行完成时，对该变量赋值 0；
 - 当该指令由于超时执行完成时，对该变量赋值 1。
- 超时跳转的标签名：当执行超时时，选择跳转的标签名所在行。

示例：

```
BitAnd (INT1, INT2)  
将 INT1 和 INT2 进行按位与的运算，并将结果赋值给 INT1。
```

MovLSync – 启动传送带同步的直线运动指令

传送带同步跟随启动指令，通过该指令可以使机器人 TCP 点以设定的速度跟随传送带一起进行同步跟随以到达目标位置。

图6-37 MovLSync 参数编辑界面

相对于目标物体的偏移量	 L: P0	⌵
目标速度	S: V2000	⌵
工具参数	DEFAULT	⌵
坐标系参数	G: SYNCOORD0	⌵
工件负载	DEFAULT	⌵
同步过程中放弃时跳转标签	SyncFail	⌵
同步后放弃时抬升高度	15	⌵
同步后放弃时跳转标签	TrackFail	⌵
AddDo		⌵

执行该指令前，机器人控制系统会通过 IO 触发视觉拍照并收到视觉系统发来的目标位置。在执行该指令时，机器人会从当前位置运动到目标位置，并与运动的目标物体保持同步。

- 相对于目标物体的偏移量：相对于跟随目标物体的偏移量，仅可设置为 CPOS 类型的位置点，在下拉列表中可选择已有的点。
- 目标速度：指令运行速度。
 - 设置为 “DEFAULT”，表示使用默认值 V4000。

- 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 工具参数：机器人执行该轨迹时使用的工具参数。
 - 设置为 “DEFAULT”，表示沿用最近一次设置的工具参数。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。仅可设置为 SYNCOORD 类型的变量，并且需要与传送带配置界面设置的动坐标系一致。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为 “DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- 同步过程中放弃时跳转标签：用于跟随丢弃功能，当机器人与目标工件同步过程中无法实现同步，即 MovLSync 指令执行失败时，程序指针会转至跳转标签所在行以执行丢弃操作，在该 Label 标签下用户可根据现场工况加入相应的指令。
 - 该值可缺省，设置为缺省时会跳转到“同步后放弃时跳转标签”所用的 Label 处执行。
- 同步后放弃时抬升高度：同步后放弃时边跟随边抬升的距离，该值为相对于动坐标系 Z 轴正方向的偏移量。
- 同步后放弃时跳转标签：用于跟随丢弃功能，当机器人在跟随运动的过程中无法继续跟随，即 MovLSync 指令后续的指令执行失败时，程序指针会转至跳转标签所在行以执行丢弃操作，在该 Label 标签下用户可根据现场工况加入相应的指令。
 - 该值不可缺省，此处需要特别注意。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。
 - IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetSimDO8421、SetDIEdge、SetSimDIEdge。

MovJSyncQuit - 同步跟随退出指令

传送带同步跟随退出指令，通过该指令可以使机器人 TCP 点以设定的速度退出传送带跟随状态并以速度最快方式到达目标位置。

图6-38 MovJSyncQuit 参数编辑界面

目标位置	 L: P0	
目标速度	DEFAULT	
过渡类型	DEFAULT	
坐标系参数	 S: World	
工件负载	DEFAULT	
AddDo		

执行该指令时，机器人与上一条运动指令平滑过渡并且退出跟随状态，之后恢复常规用户坐标系下的运动。

- 目标位置：指令终点位置（APOS 或 CPOS 类型位置点）。
 - 在下拉列表中选择已有的点或在弹出的对话框中新建并示教一个变量。
- 目标速度：指令运行速度。
 - 设置为 “DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。

- ABSOLUTE: 绝对过渡, 过渡参数是个绝对值, 表示过渡点到轨迹终点的距离。
- 坐标系参数: 机器人执行该轨迹时使用的坐标系参数。
 - 该参数无法设为 “DEFAULT”, 必须强制输入一个坐标系参数。
 - 设置为 USERCOORD 类型变量时, 可以选用系统预定义, 也可以自行创建; 使用与当前坐标系不同的坐标系时, 系统会将坐标系切换到设置的坐标系。
- 工件负载: 机器人执行该轨迹时使用的工件负载参数。
 - 设置为 “DEFAULT”, 表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时, 可以选用系统预定义, 也可以自行创建; 使用与当前工件负载参数不同的负载参数时, 系统会将工件负载参数切换到设置的工件负载参数。若本段与前后段轨迹的工件负载变化, 则不支持过渡。
- AddDo: 机器人执行完该指令后, 可进行 IO 操作。
 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetSimDO8421、SetDIEdge、SetSimDIEdge。

MovLSyncQuit - 同步跟随退出指令

传送带同步跟随退出指令，通过该指令可以使机器人 TCP 点以设定的速度退出传送带跟随状态并以直线运动方式到达目标位置。

图6-39 MovLSyncQuit 参数编辑界面

目标位置	 L: P0	
目标速度	DEFAULT	
过渡类型	DEFAULT	
坐标系参数	 S: World	
工件负载	DEFAULT	
AddDo		

执行该指令时，机器人与上一条运动指令平滑过渡并且退出跟随状态，之后恢复常规用户坐标系下的运动。

- 目标位置：指令终点位置（APOS 或 CPOS 类型位置点）。
 - 在下拉列表中选择已有的点或在弹出的对话框中新建并示教一个变量。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 坐标系参数：机器人执行该轨迹时使用的坐标系参数。
 - 该参数无法设为“DEFAULT”，必须强制输入一个坐标系参数。
 - 设置为 USERCOOR 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前坐标系不同的坐标系时，系统会将坐标系切换成设置的坐标系。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
 - 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。
 - 设置为 PAYLOAD 类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换成设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。
- AddDo：机器人执行完该指令后，可进行 IO 操作。
 - NULL：无任何操作。

IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetSimDO8421、SetDIEdge、SetSimDIEdge。

SetTargetPos - 设置工件位置信息指令

工件位置信息设置指令，用于设置工件进入传送带上被检测到的位置信息。

图6-40 SetTargetPos 参数编辑界面

动坐标系	G: SYNCOORD0	⌵
目标位置	L: P0	⌵

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。
- 目标位置：CPOS 类型变量，表示工件进入传送带的位置相对于动坐标系原点的偏移量信息，使用时需注意，目前系统仅支持 2D 视觉数据，即在该 CPOS 类型变量中仅 X、Y、A 三个数据有效。

SetSyncoord - 设置动坐标系指令

动坐标系设置指令，用于将当前机器人参考的坐标系设置为动坐标系，该指令可以将当前机器人参考的坐标系切换成动坐标系，可以快速示教跟随过程中点位的姿态信息。

图6-41 SetSyncoord 参数编辑界面

坐标系参数	G: SYNCOORD0	⌵
-------	--------------	---

- 坐标系参数：当前机器人所参考的坐标系参数，仅可设置为 SYNCOORD 类型的变量。
- 需要注意，切换至动坐标系后，仅能点动切换前所设定的点动坐标系，例如切换动坐标系前设定的点动坐标系为世界坐标系，则执行该指令后仅能点动世界坐标系且不能切换。

SynCToUserC - 示教参考坐标系转换指令

示教参考坐标系转换指令，用于解算示教跟随同步后轨迹点位时参考的坐标系。

图6-42 SynCToUserC 参数编辑界面

动坐标系	G: SYNCOORD0	⌵
目标位置	L: DCPOS3	⌵
坐标系参数	G: USERCOORD0	⌵

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。
- 目标位置：DCPOS 类型变量，表示工件进入传送带的位置相对于动坐标系原点的偏移量信息。
- 坐标系参数：USERCOORD 类型变量，指示教跟随同步后轨迹点位时参考的坐标系。

WaitConvDis - 等待固定增量指令

该指令用于使机器人沿传送带正方向与传送带同步运动到一定增量后再执行后续的指令，从而让机器人能够达到延时指定增量再运动的需求。与 Wait 指令不同的是，本指令是需要等固定增量而 Wait 是需要等固定时间。

图6-43 WaitConvDis 参数编辑界面

动坐标系	G: SYNCOORD1	⌵
增量	0	⌵

- 动坐标系：SYNCOORD 类型变量，该动坐标系需与传送带配置界面中启用的动坐标系一致。
- 等待增量：机器人待同步运动的增量值。直线传送带时，单位为 mm；圆盘传送带时，单位为 deg。

SimConveyorOn - 虚拟床送带开启指令

虚拟传送带开启指令，用于使能虚拟传送带，并指定其运行的速度，该指令参数编辑界面如图 6-44 所示。

图6-44 SimConveyorOn 参数编辑界面

动坐标系	G: SYNCOORD0	⌵
传送带速度	50	

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。
- 传送带速度：虚拟传送带的运行速度，直线传送带时单位是 mm/s；圆盘传送带时单位是 deg/s。

SimConveyorOff-虚拟传送带关闭指令

虚拟传送带关闭指令，用于禁能虚拟传送带，该指令参数编辑界面如图 6-45 所示。

图6-45 SimConveyorOff 参数编辑界面

动坐标系	G: SYNCOORD0	⌵
------	--------------	---

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。

ReceiveWObj-工件数据接收状态指令

设置工件数据的接收状态指令，通过组合使用该指令可以让机器人系统在某一段轨迹内不接收视觉或 run 程序中发来的工件位置信息，该指令参数编辑界面如图 6-46 所示。

图6-46 ReceiveWObj 参数编辑界面

动坐标系	G: SYNCOORD1	⌵
使能接收状态	"OFF"	⌵

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。
- 使能接收状态：使能机器人系统对工件位置信息的接收状态。

ResetWObjBuf - 复位已工件数据队列指令

复位已接收到的工件数据队列指令，该指令可以将队列中已接收且未被执行的工件信息清空，该指令参数编辑界面如图 6-47 所示。

图6-47 ResetWObjBuf 参数编辑界面

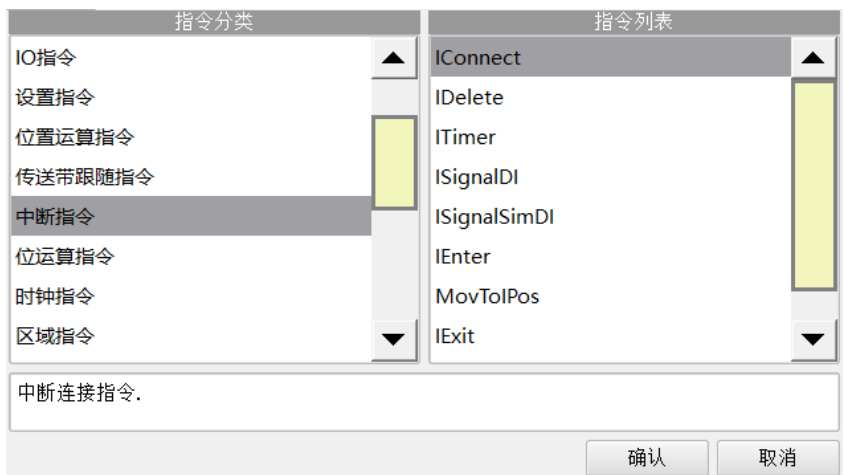
动坐标系	G: SYNCOORD0	⌵
------	--------------	---

- 动坐标系：SYNCOORD 类型变量，设置的动坐标系需与传送带配置界面启用的动坐标系一致。

6.8 中断指令

中断指令列表如图 6-48 所示。

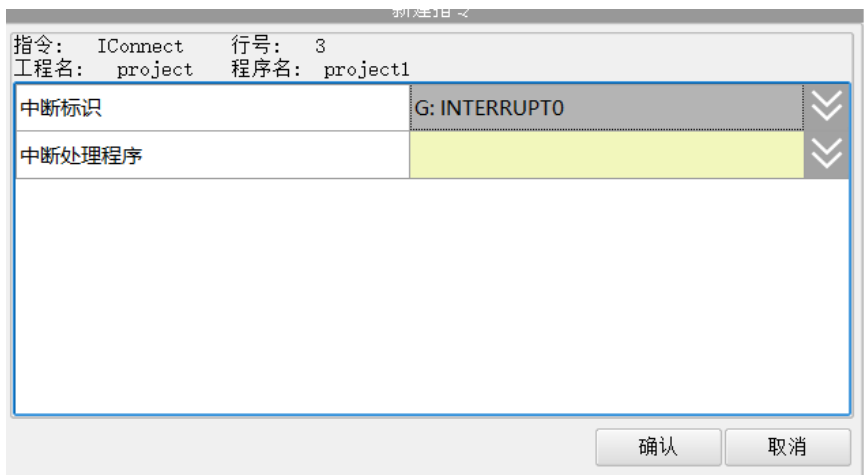
图6-48 中断指令列表



IConnect – 中断连接指令

中断连接指令，将中断标识与中断处理程序进行绑定操作。

将中断索引与中断程序进行绑定操作，与该索引绑定的信号触发后，系统可以执行与该中断索引绑定的中断程序。



- 中断标识：中断索引号（INTERRUPT 类型）。
- 中断处理程序：该中断索引对应的信号满足后需调用的处理程序。

说明

- IConnect 中的中断标识与中断程序绑定后，必须先执行 IDelete 指令删除该标识变量才可重新执行 IConnect，否则会有报错提示。
- 中断标识变量创建时，该变量中的值应与该变量名同步。即新建了 INTERRUPT 类型的 inter1 变量，在变量界面查看其值时应也显示为 inter1。
- 中断处理程序只能为本工程下的程序，不支持全局程序

IDelete – 中断解除指令

中断解除指令，将中断标识与相应的中断处理程序之间的连接去掉。

移除系统中与该中断索引绑定的任何数据，即移除对该中断索引的检测以及执行队列中数据。

指令: IDelete 行号: 3

工程名: project 程序名: project1

中断标识	G: INTERRUPT0	⌵
------	---------------	---

确认 取消

- 中断标识：中断索引号（INTERRUPT 类型）。

ITimer – 定时器中断指令

定时器中断指令，设置中断索引的检测方式为定时触发，即定时处理机器人相应的中断程序。根据设定的延时时间周期性的让系统执行中断程序。

指令: ITimer 行号: 3

工程名: project 程序名: project1

中断标识	G: INTERRUPT0	⌵
触发时间(ms)	2000	

确认 取消

- 中断标识：中断索引号（INTERRUPT 类型）。
- 触发时间：定时检测时间，单位 ms。取值范围：100ms~30000ms

IsignalDI – 输入口中断指令

输入口中断指令，设置中断索引的检测方式为 DI 边沿触发，即实时监控 DI 的上升沿/下降沿信号来处理机器人相应的中断程序。

指令: ISignalDI 行号: 3
工程名: project 程序名: project1

中断标识	G: INTERRUPT0	⌵
端口变量	L: DIO	⌵
触发边沿类型	"RiseEdge"	⌵

确认 取消

监控设定 DI 的边沿信号有效状态来让系统执行中断程序，指令参数如下：

- 中断标识：中断索引号（INTERRUPT 类型）。
- 端口变量：需要监控的 DI 端口号（DI 类型）。
- 触发边沿类型：需要监控的 DI 边沿信号类型：
 - RiseEdge：上升沿信号
 - DownEdge：下降沿信号

说明

- 该指令中的中断索引号必须与中断程序已经绑定，即执行了 IConnect 指令，否则将有错误提示信息。
- 该指令执行后，在程序运行状态下给予对应的 DI 有效边沿信号，与该中断索引绑定的中断程序应能被正确执行。

ISignalSimDI – 虚拟输入口中断指令

虚拟输入口中断指令，设置中断索引的检测方式为 SimDI 边沿触发，即实时监控 SimDI 的上升沿/下降沿信号来处理机器人相应的中断程序。

中断标识	G: INTERRUPT1	⌵
端口变量	L: SimDIO	⌵
触发边沿类型	"RiseEdge"	⌵

- 中断标识：中断索引号（INTERRUPT 类型）。
- 端口变量：需要监控的虚拟 SimDI 端口号（SimDI 类型）。
- 触发边沿类型：需要监控的 SimDI 边沿信号类型：
 - RiseEdge：上升沿信号
 - DownEdge：下降沿信号

IEnter – 中断当前解析与运动指令

中断当前解析与运动的指令，停止主程序的解析与运动并切换到该中断程序中执行。

在中断程序中执行该指令后系统会让主程序停止解析并中断当前的运动，同时将中断程序中的数据与程序显示在程序编辑界面以及变量界面。

说明

该指令只能在中断程序中执行。

MovToIPos – 返回中断点指令

返回中断点指令，使机器人从当前位置移动到主程序运行时被中断时刻的位置点。

指令: MovToIPos	行号: 5
工程名: project	程序名: project1
运动方式	"MOVJ"
目标速度	DEFAULT
确认 取消	

指定的运动形式使机器人自动回到主程序被中断时刻的位置，中断发生时系统会自动缓存这个中断位置点，本指令执行时会自动从内存中取出该中断位置点并运行到该点，无需用户对该点进行额外的指令处理

- 运动方式：运动到中断点的运动方式
 - MOVJ：以关节运动的形式返回到目标点
 - MOVL：以直线运动的形式返回到目标点
- 目标速度：运动到中断点的速度（SPEED 类型）

IExit – 恢复中断前解析与运动指令

恢复解析与运动的指令，将中断程序切到后台执行并恢复主程序的解析与运动。

在中断程序中执行该指令后系统会将中断程序切换到后台执行，并将主程序退出休眠状态，继续上一次的指令解析与运动操作，同时将主程序中的数据与程序显示在程序编辑界面以及变量界面。

ExitCycle – 退出循环指令

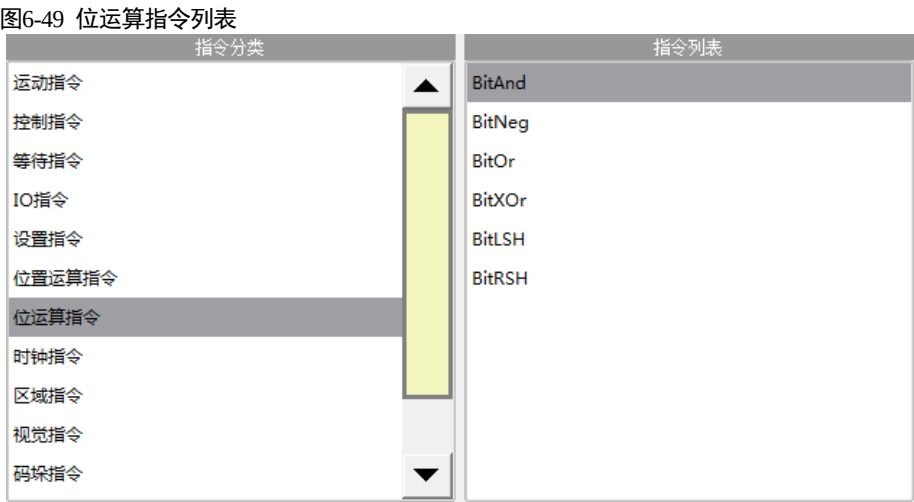
退出循环指令，执行该指令后系统会立即结束当前中断程序的运行，将 PC 指针返回到主程序的起始位置处，并将系统转入停止状态。

说明

1. 在中断程序中，该指令必须与 IEnter 指令搭配使用。
2. 执行该指令后，如果主程序示教了“CycleLable” label 指令，系统会跳转到该位置处执行；如果没示教器该指令，系统会跳转到被加载的程序首行处执行。

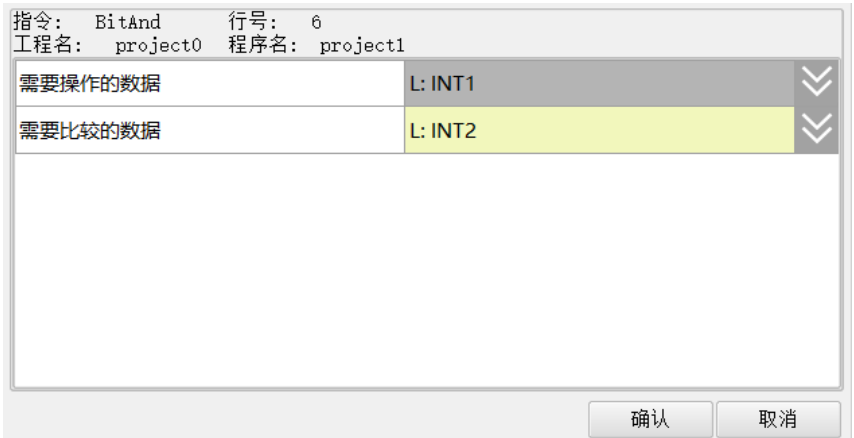
6.9 位运算指令

位运算指令列表如图 6-49 所示。



BitAnd - 与运算指令

实现按位与的运算。该指令将两个操作数进行按位与的运算，并将结果赋值给第一个操作数。



- 需要操作的数据：INT 型变量；运算结果也赋值到该操作数。
- 需要比较的数据：INT 型变量。

BitAnd (INT1, INT2)
将 INT1 和 INT2 进行按位与的运算，并将结果赋值给 INT1。

BitNeg - 取反运算指令

实现按位取反的运算。该指令将操作数进行按位取反的运算，并将结果赋值给该操作数。
此外，取反操作仅操作数后 16 位。

指令：BitNeg 行号： 6

工程名： project0 程序名： project1

需要操作的数据	L: INT1	⌵
---------	---------	---

确认

取消

- 需要操作的数据：INT 型变量；运算结果也赋值到该操作数。

BitNeg (INT1)

将 INT1 进行按位取反的运算，并将结果赋值给 INT1。

BitOr - 或运算指令

实现按位或的运算。该指令将两个操作数进行按位或的运算，并将结果赋值给第一个操作数。

新建指令

指令：BitOr 行号： 6

工程名： project0 程序名： project1

需要操作的数据	L: INT1	⌵
需要比较的数据	L: INT2	⌵

确认

取消

- 需要操作的数据：INT 型变量；运算结果也赋值到该操作数。
- 需要比较的数据：INT 型变量。

BitOr (INT1, INT2)

将 INT1 和 INT2 进行按位或的运算，并将结果赋值给 INT1。

BitXOr - 异或运算指令

实现按位异或的运算。该指令将两个操作数进行按位异或的运算，并将结果赋值给第一个操作数。

指令：BitXOr 行号： 6

工程名： project0 程序名： project1

需要操作的数据	L: INT1	⌵
需要比较的数据	L: INT2	⌵

确认

取消

- 需要操作的数据：INT 型变量；运算结果也赋值到该操作数。
- 需要比较的数据：INT 型变量。

```
BitXOr (INT1, INT2)  
将 INT1 和 INT2 进行按位异或的运算，并将结果赋值给 INT1。
```

BitLSH - 左移位运算指令

实现按位左移的运算。该指令将第一个操作数按第二个操作数指定的左移位数进行按位左移的运算，并将结果赋值给第一个操作数。

指令： BitLSH 行号： 6
工程名： project0 程序名： project1

需要操作的数据	L: INT1	⌵
左移的位数	L: INT2	⌵

确认 取消

- 需要操作的数据：INT 型变量；运算结果也赋值到该操作数。
- 左移的位数：INT 型变量，左移位数。

```
BitLSH (INT1, INT2)  
将 INT1 和 INT2 进行按位左移的运算，并将结果赋值给 INT1。
```

BitRSH - 右移位运算指令

实现按位右移的运算。该指令将第一个操作数按第二个操作数指定的右移位数进行按位右移的运算，并将结果赋值给第一个操作数。

指令： BitRSH 行号： 6
工程名： project0 程序名： project1

需要操作的数据	L: INT1	⌵
右移的位数	L: INT2	⌵

确认 取消

- 需要操作的数据：INT 型变量；运算结果也赋值到该操作数。
- 右移 wei2 的数据：INT 型变量，右移位数。

```
BitRSH (INT1, INT2)  
将 INT1 和 INT2 进行按位右移的运算，并将结果赋值给 INT1。
```

6.10 时钟指令

时钟指令列表如图 6-50 所示。

图6-50 时钟指令列表

指令分类	指令列表
运动指令	CLKStart
控制指令	CLKStop
等待指令	CLKRead
IO指令	CLKReset
设置指令	
位置运算指令	
位运算指令	
时钟指令	
区域指令	
视觉指令	
码垛指令	

CLKStart-启动时钟指令

启动指定时钟（启动后，可以从变量列表中看到指定时钟变量的 state 为 true）。例如：

```
CLKStart (CLOCK0)  
启动时钟 CLOCK0。
```

CLKStop - 停止时钟指令

停止指定时钟（其 state 为 false，但不会复位）。例如：

```
CLKStop (CLOCK0)  
停止时钟 CLOCK0。
```

CLKRead - 读取时钟指令

读取指定时钟的值，在变量列表查看变量对应的 value 即可。例如：

```
CLKRead (CLOCK0)  
读取时钟存于 CLOCK0。
```

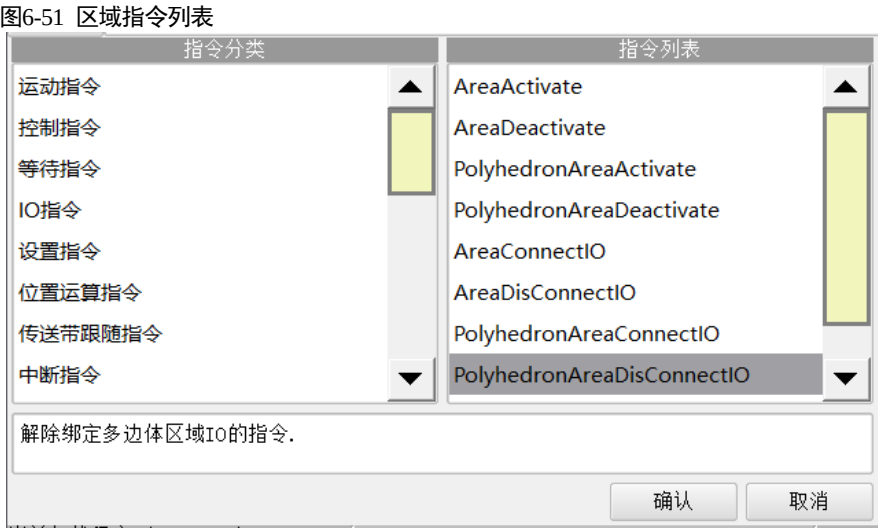
CLKReset - 复位时钟指令

复位指定时钟的状态值。例如：

```
CLKReset (CLOCK0)  
复位时钟 CLOCK0。
```

6.11 区域指令

区域指令列表如图 6-51 所示。



AreaActivate - 激活标准区域指令

激活标准区域指令，使指定的标准区域生效。



AreaActivate (AREA0)
激活标准区域监控，使 AREA0 生效。

AreaDeactivate - 冻结标准区域指令

冻结标准区域指令，使指定的标准区域失效。

AreaDeactivate (AREA0)
冻结标准区域监控，使 AREA0 失效。

PolyhedronAreaActivate - 激活多边形区域指令

激活多边形区域指令，使指定的多边形区域生效。

PolyhedronAreaActivate (POLYHEDRON0)
激活多边形区域监控，使 POLYHEDRON0 对应的多边形区域生效。

PolyhedronAreaDeactivate - 冻结多边形区域指令

冻结多边形区域指令，使指定的多边形区域失效。

PolyhedronAreaDeactivate (POLYHEDRON0)
冻结多边形区域监控，使 POLYHEDRON0 对应的多边形区域失效。

AreaConnectIO – 绑定区域 IO 指令

绑定区域 IO 的指令，通过该指令，可为选定区域绑定一个输入端口，并指定高低有效电平，用于控制机器人的启动与停止。

指令: AreaConnectIO 行号: 3	
工程名: project 程序名: main	
区域变量	G: AREA0
IO类型	"DI"
IO端口号	9
IO极性	"HIGH"
确认 取消	

- 区域变量：AREA，需要绑定的标准区域变量。
- IO 类型：需要绑定的 IO 口类型。
 - DI：实际 DI。
 - SIMD：虚拟 DI。
- IO 端口号：需要绑定的 IO 端口号。
- IO 极性：需要绑定的 IO 极性。
 - HIGH：高位有效。
 - LOW：低位有效。

AreaDisconnectIO – 解除绑定区域 IO 指令

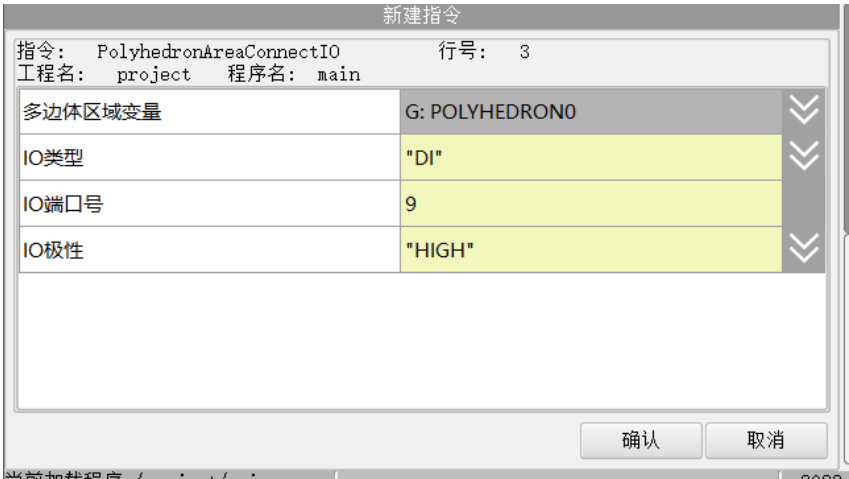
解除绑定区域 IO 的指令，通过该指令，可取消与选定区域输入端口的绑定。

新建指令	
指令: AreaDisconnectIO 行号: 3	
工程名: project 程序名: main	
区域变量	G: AREA0
确认 取消	

- 区域变量：需要解除绑定的标准区域变量。

PolyhedronAreaConnectIO – 绑定多边形区域 IO 指令

绑定区域 IO 的指令，通过该指令，可为选定区域绑定一个输入端口，并指定高低有效电平，用于控制机器人的启动与停止。



- 区域变量：POLYHEDRON，需要绑定的多边形区域变量。
- IO 类型：需要绑定的 IO 口类型。
 - DI：实际 DI。
 - SIMD：虚拟 DI。
- IO 端口号：需要绑定的 IO 端口号。
- IO 极性：需要绑定的 IO 极性。
 - HIGH：高位有效。
 - LOW：低位有效。

PolyhedronAreaDisconnectIO - 解除多边形区域 IO 指令

解除绑定区域 IO 的指令，通过该指令，可取消与选定区域输入端口的绑定。



区域变量：需要解除绑定的多边形区域变量

6.12 视觉指令

视觉指令列表如图 6-52 所示。

图6-52 视觉指令列表

指令分类	指令列表
运动指令	TrigCam
控制指令	WaitFinishCAM
等待指令	GetCamPos
IO指令	SendMessage
设置指令	Tracking
位置运算指令	
位运算指令	
时钟指令	
区域指令	
视觉指令	
码垛指令	

TrigCam - 触发相机指令

触发相机拍摄指令。

指令: TrigCam 行号: 6
工程名: project0 程序名: project1

相机触发参数	OK
相机索引号	1

确认 取消

TrigCam ("OK", 1)
字符串 "OK" 触发相机拍照。

WaitFinishCAM - 等待相机拍摄完成指令

等待相机拍摄完成的指令。

WaitFinishCAM (1000,1)
等待相机拍照完成的时间为 1000ms。

GetCamPos - 获取相机位置指令

获取相机位置的指令。

指令: GetCamPos 行号: 6
工程名: project0 程序名: project1

目标位置	 L: P3	⌵
目标物体ID值	L: INT1	⌵
有效性	L: INT2	⌵
相机索引号	0	

确认 取消

GetCamPos (P6, INT0, INT1,1)
获取相机的位置 P6, 目标物体 ID 为 INT0, 判断是否有效为 INT1。

SendMessage - 发送字符串信息指令

发送字符串消息，可用于 3D 视觉指令，给相机发送指定的字符串，让相机拍照。例如：

SendMessage ("OK", 1)
向相机发送字符串为“OK”。

Tracking - 跟踪传送带指令

跟踪传送带指令，关于该指令的详细说明，请参见 7.5 传送带跟随参数介绍

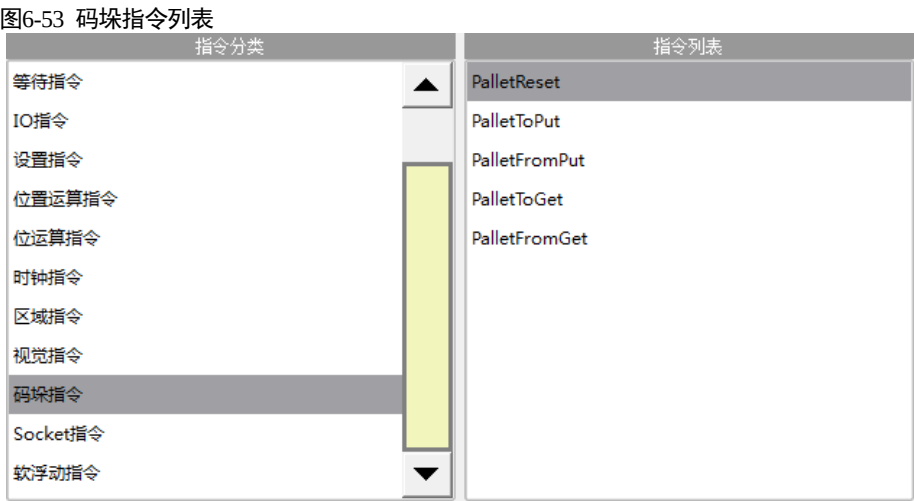
- 使能复选框：控制振动抑制功能的开关
 - 勾选：打开振动抑制功能。
 - 未勾选：关闭振动抑制功能。
- 轴 1~轴 6 的值：设置振动抑制的强度
 - 默认值：标准
 - 取值范围：共计 5 个档位：低、中低、标准、中高、高。振动抑制效果依次逐步增强。

传送带跟随。

Tracking (10.0, 10.0, DO2, DO1, INT0,INT1)

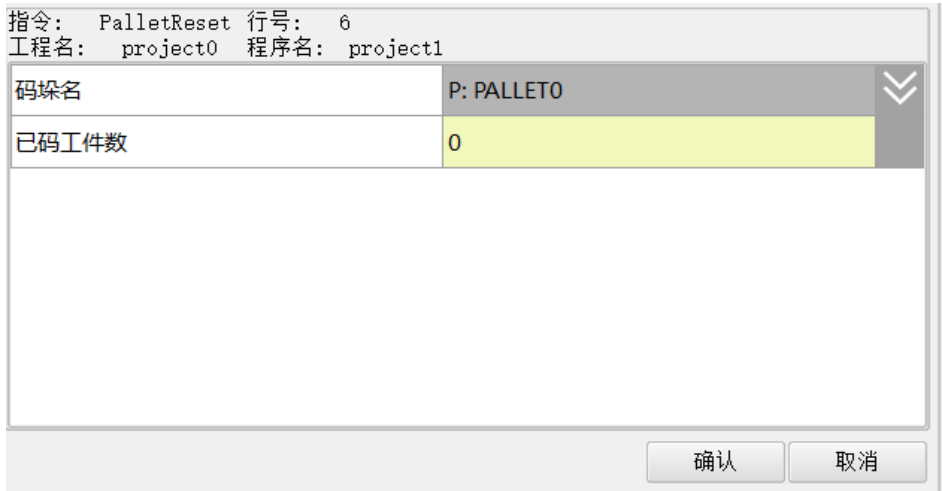
6.13 码垛指令

码垛指令列表如图 6-53 所示。



PalletReset - 码垛重置指令

码垛重置指令。当一个垛体在未码放完成而被迫停止时，可调用该指令来还原。



- 码垛名：当前码垛的 PALLET 变量。
- 已码工件数：当前码垛上已经放置的工件数量。一般必须设置为 0 至该垛体的最大码垛工件数。

```
PalletReset (PALLET0,0)
```

例如，最大工件数为 60 的垛体 Pallet0，在放置完 8 个工件时中止。可调用 “PalletReset(Pallet0, 8);” 重新开始后续工件的码放。

PalletToPut - 放置工件指令

放置工件指令。

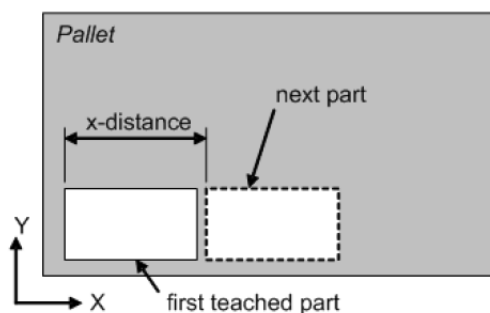
指令: PalletToPut 行号: 6	
工程名: project0 程序名: project1	
码垛名	P: PALLET0
目标速度	DEFAULT
过渡类型	DEFAULT
确认 取消	

- 码垛名: 当前码垛的 PALLET 变量。
- 目标速度: 指令运行速度。
 - 设置为 “DEFAULT”, 表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时, 可以选用系统预定义, 也可以自行创建。
- 过渡类型: 机器人逼近终点时的过渡方式。
 - DEFAULT: 缺省, 即指令中不设置过渡参数。若程序中使用了全局过渡参数 (SetBlendParam), 过渡方式按照全局设置的过渡参数运行。
 - FINE: 效果等同于 DEFAULT。
 - RELATIVE: 相对过渡, 过渡参数为相对值, 表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE: 绝对过渡, 过渡参数是个绝对值, 表示过渡点到轨迹终点的距离。
- 过渡值: 机器人逼近终点时的过渡值。
 - 设置为 “DEFAULT”, 表示使用默认值 C100。
 - 设置为 ZONE 类型变量时, 可以选用系统预定义, 也可以自行创建; 其中, 当变量值为 0 时表示不过渡, 过渡值越大过渡半径越大。

```
PalletToPut (PALLET1,V2000)
```

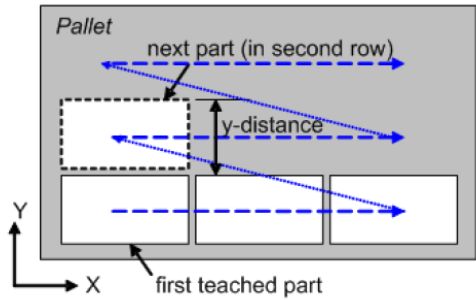
码垛功能的垛位计算取决于示教的第一个工件位置和配置的码垛的距离。在前一个工件的位置基础上加上设定的码垛距离, 从而得到下一个工件的位置, 如图 6-54 所示。

图6-54 计算下一个工件的位置



在图 6-55 中, 默认码垛的顺序从 X 开始。首先放置 x 方向上的所有部件, 然后再开始放置下一行。在 x 方向上放置工件直到达到设置好的 x 方向工件数。

图6-55 默认码垛顺序

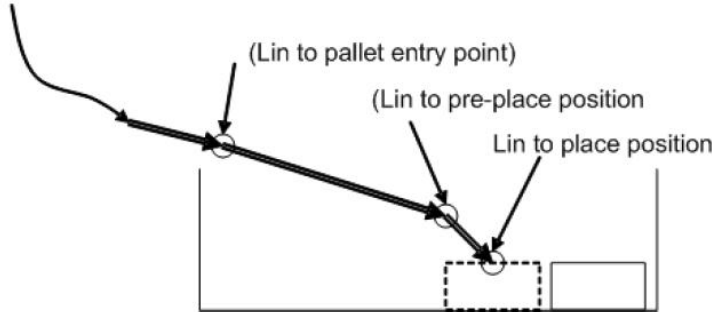


当新的一行开始放置时，在第一行工件位置的基础上加上 y 方向上的偏移即生成了新行的工件的位置数据。

这个例子里默认的码垛顺序是 x y z. 首先在 x 方向放置工件然后再进行第二行。新的一行的第一个部件从图中所示的位置开始。只要行满的，它们就会被放在下一行。在这一层的所有行都被填满后，xy 层在 z 方向重复平移，直到垛满。

进行放置时，建议路径如图 6-56 所示。

图6-56 放置路径



在调用 PalletToPut 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛的进入位置（可选）
- 机器人以直线的运动方式移动至垛点的前点位置（可选）
- 机器人以直线的运动方式移动至工件的放置位置（必选）

PalletFromPut - 离开放置位置指令

放置工件完成后，机器人从放置位离开的指令。

指令: PalletFromPut 行号: 6
工程名: project0 程序名: project1

码垛名	P: PALLETO	⌵
目标速度	DEFAULT	⌵
过渡类型	"RELATIVE"	⌵
过渡值	DEFAULT	⌵

确认 取消

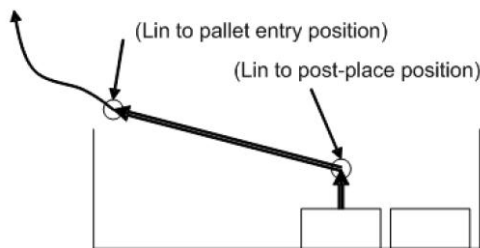
- 码垛名：当前码垛的 PALLET 变量。
- 目标速度：指令运行速度。
 - 设置为 “DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为 “DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。

```
PalletFromPut (PALLET2,V2000)
```

该指令是放置工件的相反动作，从放置好的工件位置离开。

离开时，建议路径如图 6-57 所示。

图6-57 放置后离开的路径



在调用 PalletFromPut 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛点的后点位置（可选）
- 机器人以直线的运动方式移动至垛的进入位置（可选）

如果垛体的后点和进入位置都没有被设置，执行 PalletFromPut 指令时，机器人保持当前的姿态不变。

PalletToGet - 抓取工件指令

该指令是进行抓取工件动作，机器人执行该指令时，会从当前点移动到目标工件的码垛点进行抓取工件。

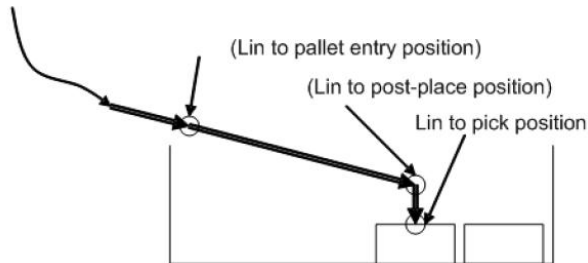
- 码垛名：当前码垛的 PALLET 变量。
- 目标速度：指令运行速度。
 - 设置为 “DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。

- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。

`PalletToGet (PALLET3,V2000)`

抓取过程中，路径的规划建议如图 6-58 所示。

图6-58 抓取工件路径



在调用 PalletToGet 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛的进入位置（可选）
- 机器人以直线的运动方式移动至垛点的后点位置（可选）
- 机器人以直线的运动方式移动至抓取位置（必选）

在这些运动序列之后，机器人到达工件的放置位置，然后关闭夹具，抓取工件。

PalletFromGet - 离开抓取位置指令

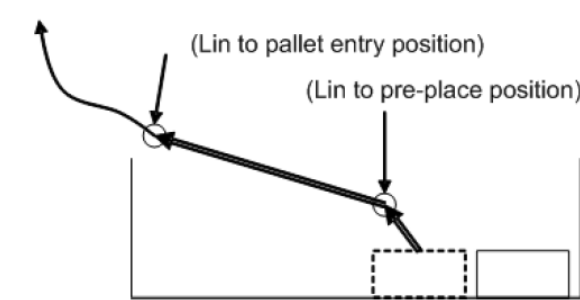
该指令是进行抓取工件动作后离开的过程，机器人执行该指令时，会从抓取工件的码垛点离开。

- 码垛名：当前码垛的 PALLET 变量。
- 目标速度：指令运行速度。
 - 设置为“DEFAULT”，表示使用默认值 V4000。
 - 设置为 SPEED 类型变量时，可以选用系统预定义，也可以自行创建。
- 过渡类型：机器人逼近终点时的过渡方式。
 - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
 - FINE：效果等同于 DEFAULT。
 - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
 - ABSOLUTE：绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。
- 过渡值：机器人逼近终点时的过渡值。
 - 设置为“DEFAULT”，表示使用默认值 C100。
 - 设置为 ZONE 类型变量时，可以选用系统预定义，也可以自行创建；其中，当变量值为 0 时表示不过渡，过渡值越大过渡半径越大。

`PalletFromGet (PALLET4,V2000)`

抓取过程中，路径的规划建议如图 6-59 所示。

图6-59 抓取后离开的路径



在调用 PalletFromGet 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛点的前点位置（可选）
- 机器人以直线的运动方式移动至码垛的进入位置（可选）

如果垛点的后点和进入位置都没有被设置，执行 PalletFromGet 指令时，机器人保持当前的姿态不变。

6.14 阵列指令

阵列指令列表如图 6-60 所示。

图6-60 阵列指令列表

指令分类	指令列表
传送带跟随指令	SetMatrix
位运算指令	GetMatrix
时钟指令	
区域指令	
视觉指令	
码垛指令	
阵列指令	
Socket指令	
软浮动指令	

SetMatrix - 设置矩阵组

以指定三个点形成空间中的一个平行四边形，并将这个平行四边形按照设置的行数和列数进行等分得到一个矩阵点组。

图6-61 SetMatrix 参数编辑界面

新建指令

指令: SetMatrix 行号: 9
工程名: testMatrix 程序名: main

原点	L: P1	⌵ ⌵ ⌵
行方向最后一个点	L: P2	
列方向最后一个点	L: P3	
行数	1	
列数	1	

(Row * Column) < 32767

示教 确认 取消

- 原点：指定平行四边形的第一个点，也称为原点，类型为 CPOS。
- 行方向最后一个点：指定平行四边形的行方向上的最后一个点，类型为 CPOS。
- 列方向最后一个点：指定平行四边形的列方向上的最后一个点，类型为 CPOS。
- 行数：生成矩阵的行数，为常量，范围[1,32767]。
- 列数：生成矩阵的列数，为常量，范围[1,32767]。

说明

根据三点生成的矩阵点仅仅是位移方向上的运算，所以生成矩阵点的姿态及附加轴角度值，均保持与原点（如下 P1_Home）一致：

如果行数为 1，列数>1，则根据原点（如下 P1_Home）和行方向最后一个点（如下 P2_Row）组成直线按照列数进行等分，不考虑列方向最后一个点（如下 P3_Column）；

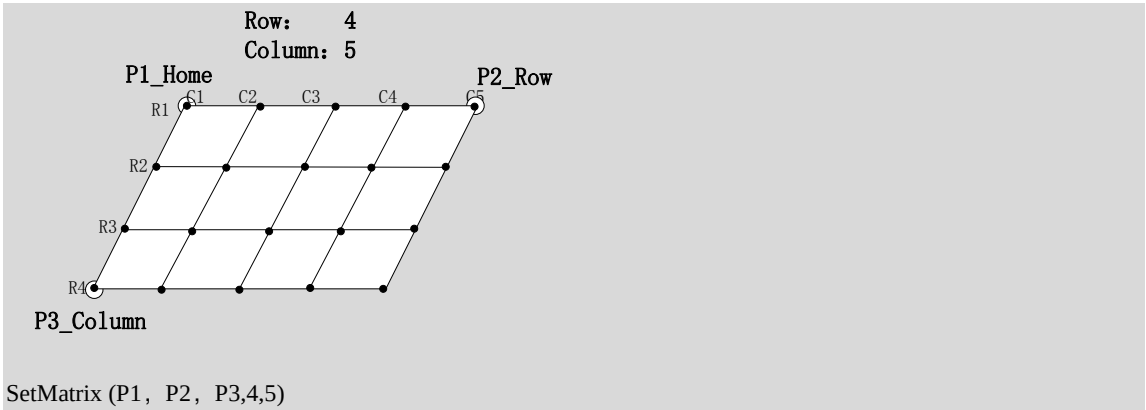
如果列数为 1，行数>1，则根据原点（如下 P1_Home）和列方向最后一个点（如下 P3_Column）组成直线按照行数进行等分，不考虑行方向最后一个点（如下 P2_Row）；

如果行数和列数均为 1，则直接使用原点（如上 P1_Home）；

行数范围[1,32767]，列数范围[1,32767]，且乘积需要小于 32767。

示例 1：

将 P1、P2 和 P3 形成的四边形按照 4 行 5 列均分：



GetMatrix - 获取矩阵组

取 SetMatrix 指令执行后对应行列的该点的值，赋值给目标点。如 SetMatrix 中所述点的姿态和附加轴角度值保持与原点一致。

图6-62 GetMatrix 参数编辑界面

新建指令

指令: GetMatrix 行号: 16
工程名: testMatrix 程序名: main

行数	L: INT0	⌵
列数	L: INT1	⌵
目标点	 L: P4	⌵

确认 取消

- 行数：所要取点在矩阵中的行号，类型 INT。
- 列数：所要取点在矩阵中的列号，类型 INT。
- 目标点：用于保存取到的点值，类型为 CPOS。

 说明

行数范围[1,32767]，列数范围[1,32767]，且行数和列数值需要小于等于 SetMatrix 指令中对应设置值。

6.15 Socket 指令

Socket 指令列表如图 6-63 所示。

图6-63 Socket 指令列表

指令分类	指令列表
传送带跟随指令	SocketCreate
位运算指令	SocketClose
时钟指令	SocketSendStr
区域指令	SocketSendReal
视觉指令	SocketSendInt
码垛指令	SocketReadReal
阵列指令	SocketReadInt
Socket指令	SocketReadStr
软浮动指令	

SocketCreate - 创建 socket 客户端指令

创建 socket 客户端，以便于对服务器进行数据交互。

根据传入的服务器端参数，在本地创建客户端并与服务器端建立连接。

指令: SocketCreate 行号: 6
工程名: project0 程序名: project1

socket名称	L: Socket0	⌵
IP地址		
端口号	1	
操作返回值	L: INT1	⌵

确认取消

- socket 名称：需要创建的 socket 名称，该值传入的是一个 socket 变量。
- IP 地址：需要连接的服务器 ip 地址。
- 端口号：需要连接的服务器端口号。
- 操作返回值：创建操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

```
SocketCreate (Socket0, "192.168.1.1", 8888, INT0)  
/*创建一个 Socket，名称为 Socket0，连接的服务器 IP 为 “192.168.1.1”，端口为 8888*/
```

SocketClose - 关闭 socket 客户端指令

关闭之前已创建的 socket 客户端。

根据传入的 socket 名，关闭创建过的客户端，并将操作的成功与否返回。

指令: SocketClose 行号: 6
工程名: project0 程序名: project1

socket名称	L: Socket0	⌵
操作返回值	L: INT1	⌵

确认取消

- socket 名称：需要关闭的 socket 名称，该值传入的是一个 socket 变量。
- 操作返回值：创建操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

```
SocketClose (Socket0, INT0)  
/*关闭 Socket0*/
```

SocketSendStr - 通过 Socket 发送字符串指令

向服务器端发送字符串，进行命令的交互。

向已经建立连接的服务器端发送字符串，并将成功与否返回。

指令: SocketSendStr 行号: 6	
工程名: project0 程序名: project1	
socket名称	L: Socket0
发送的字符串	
操作返回值	L: INT1
确认 取消	

- socket 名称: 需要执行发送操作的 socket 名称, 该值传入的是一个 socket 变量。
- 发送的字符串: 需要发给服务器端的字符串数据。
- 操作返回值: 创建操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

```
SocketSendStr(Socket0, "test string", INT0)
/*发送字符串 "test string" 到服务器*/
```

SocketSendReal – 通过 Socket 发送 real 数组指令

向服务器端发送 real 数组, 进行命令的交互。

向已经建立连接的服务器端发送 real 数组, 并将成功与否返回。

指令: SocketSendReal 行号: 6	
工程名: project0 程序名: project1	
socket名称	L: Socket0
发送的real数组数据	L: RealOneArray0
操作返回值	L: INT1
确认 取消	

- socket 名称: 需要执行发送操作的 socket 名称, 该值传入的是一个 socket 变量。
- 发送的 real 类型数组: 需要发给服务器端的 real 数组数据。
- 操作返回值: 创建操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

```
SocketSendReal(Socket0, RealOneArray0, INT0)
/*发送 RealOneArray 一维数组信息到服务器*/
```

SocketSendInt – 向 Socket 发送 int 数组指令

向服务器端发送 int 数组, 进行命令的交互。

向已经建立连接的服务器端发送 int 数组, 并将成功与否返回。

指令: SocketSendInt 行号: 6	
工程名: project0 程序名: project1	
socket名称	L: Socket0
发送的int数组数据	L: IntOneArray0
操作返回值	L: INT1
确认 取消	

- socket 名称: 需要执行发送操作的 socket 名称, 该值传入的是一个 socket 变量。
- 发送的 int 类型数组: 需要发给服务器端的 int 数组数据。
- 操作返回值: 创建操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

```
SocketSendInt (Socket0, IntOneReal, INT0)
/*发送 IntOneArray 一维数组信息到服务器*/
```

SocketReadReal - 读取服务器端发送字符串存储成 Real 数组形式

读取服务器端发来的字符串, 并以 real 数组的形式存储。

等待并接收服务器端发送的字符串, 其格式为[1.1,2.2,3.3,4.4](即数据起末为'[', 数字与数字之间用','隔开), 当接收到该字符串后, 机器人系统会将其拆解析并按顺序存放数组中。

指令: SocketReadReal 行号: 6	
工程名: project0 程序名: project1	
socket名称	L: Socket0
读取的数据个数	1
数据返回值	L: RealOneArray0
检测时间(ms)	100
操作返回值	L: INT1
确认 取消	

- socket 名称: 需要执行读取操作的 socket 名称, 该值传入的是一个 socket 变量。
- 读取的数据个数: 需要读取的数据存储到数组的个数。
- 数据返回值: 将读取并转换后的值存入到数组变量中, 并将该数组变量返回。
- 检测时间: 等待服务器端发送数据的等待时间。超时报警。
- 操作返回值: 创建操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

```
SocketReadReal (Socket0, 5, RealOneArray0, 100, INT0)
/*在 100ms 内, 从服务器上读取 5 个实数, 放到变量 RealOneArray0 中*/
```

SocketReadInt - 读取服务器端发送字符串存储成 int 形式

读取服务器端发来的字符串, 并以 int 数组的形式存储。

等待并接收服务器端发送的字符串, 其格式为[1,2,3,4](即数据起末为'[', 数字与数字之间用','隔开), 当接收到该字符串后, 机器人系统会将其拆解析并按顺序存放数组中。

指令: SocketReadInt 行号: 6
工程名: project0 程序名: project1

socket名称	L: Socket0	⌵
读取的数据个数	1	⌵
数据返回值	L: IntOneArray0	⌵
检测时间(ms)	100	⌵
操作返回值	L: INT1	⌵

确认 取消

- socket 名称：需要执行读取操作的 socket 名称，该值传入的是一个 socket 变量。
- 读取的数据个数：需要读取的数据存储到数组的个数。
- 数据返回值：将读取并转换后的值存入到数组变量中，并将该数组变量返回。
- 检测时间：等待服务器端发送数据的等待时间。超时报警。
- 操作返回值：创建操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

```
SocketReadInt (Socket0, 5, IntOneArray0, 100, INT0)  
/*在 100ms 内，从服务器上读取 5 个整数，放到变量 IntOneArray0 中*/
```

SocketReadStr-读取服务器发送的字符串存储成 string 变量形式

读取服务器端发来的字符串，并以 string 变量的形式存储。

等待并接收服务器端发送的字符串，其格式为盘[Hello world!](即数据起末为'[']')，当接收到该字符串后，机器人系统会将其存放到字符串变量中。

指令: SocketReadStr 行号: 6
工程名: project0 程序名: project1

socket名称	L: Socket0	⌵
数据返回值	L: STRING0	⌵
检测时间(ms)	100	⌵
操作返回值	L: INT1	⌵

确认 取消

- socket 名称：需要执行读取操作的 socket 名称，该值传入的是一个 socket 变量。
- 读取的字符串：服务器端发送来的字符串数据，该值以一个 string 变量的形式返回。
- 检测时间：等待服务器端发送数据的等待时间。超时报警。
- 操作返回值：创建操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

```
SocketReadStr (Socket0, STRING0, 100, INT0)  
/*在 100ms 内，从服务器上读取字符串，放到变量 STRING0 中*/
```

SocketResetBuf - 复位本地缓存的字符串

复位本地缓存的字符串，用于在执行 SocketRead 相关读取指令之前复位缓存区中已接收到的数据，避免出现数据读取错位的问题。

- socket 名称：需要执行读取操作的 socket 名称，该值传入的是一个 socket 变量。

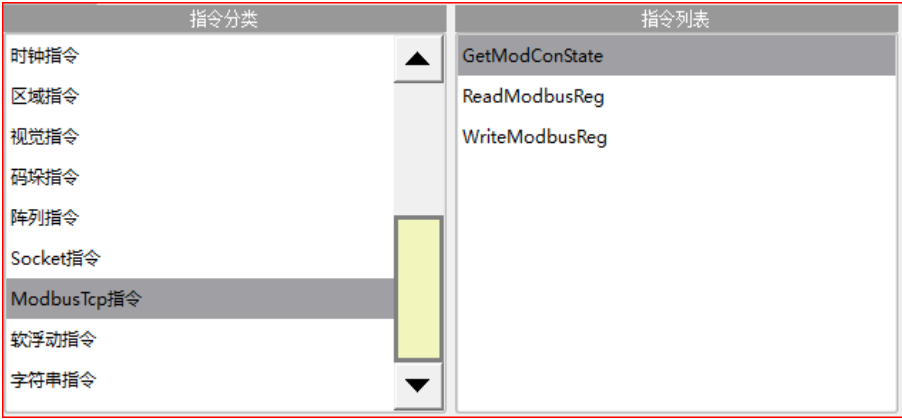
示例：

```
SocketResetBuf (Socket0)
/*复位 Socket0 中缓存的通讯数据*/
```

6.16 ModbusTCP 指令

ModbusTCP 指令列表如图 6-64 所示。

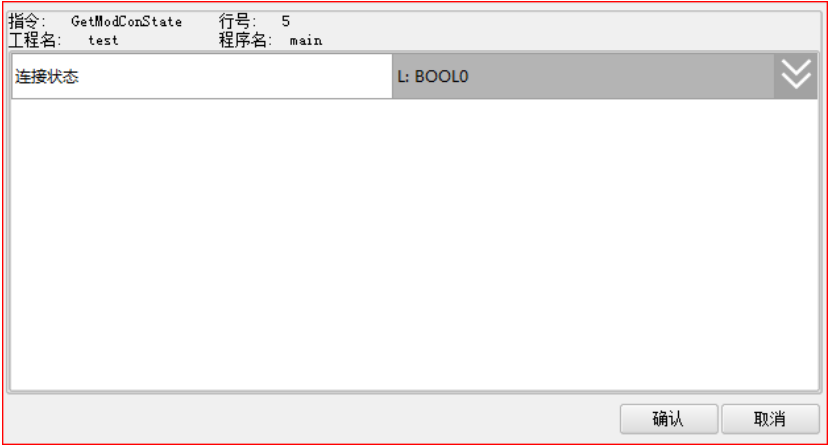
图6-64 Modbus 指令列表



GetModConState - 获取与 ModbusTCP 通讯连接连接状态

该指令用于获取机器人与外界使用 ModbusTCP 通讯的连接状态。

图6-65 GetModConState 参数编辑界面



- 连接状态：返回当前的连接状态，类型为 BOOL。

ReadModbusReg - 读取指定 Modbus 寄存器值

该指令用于读取指定 Modbus 寄存器的值。

图6-66 ReadModbusReg 参数编辑界面

指令: ReadModbusReg

行号: 5

工程名: test

程序名: main

目标寄存器ID号	101
目标寄存器值	L: INT1

确认

取消

- 目标寄存器 ID 号：想要读取的寄存器 ID 号，类型 int。
- 目标寄存器值：返回读取后的寄存器值，类型 INT。

WriteModbusReg - 设置指定 Modbus 寄存器值

该指令用于设置指定 Modbus 寄存器的值。

图6-67 WriteModbusReg 参数编辑界面

指令: WriteModbusReg

行号: 5

工程名: test

程序名: main

目标寄存器ID号	101
目标寄存器值	L: INT1

确认

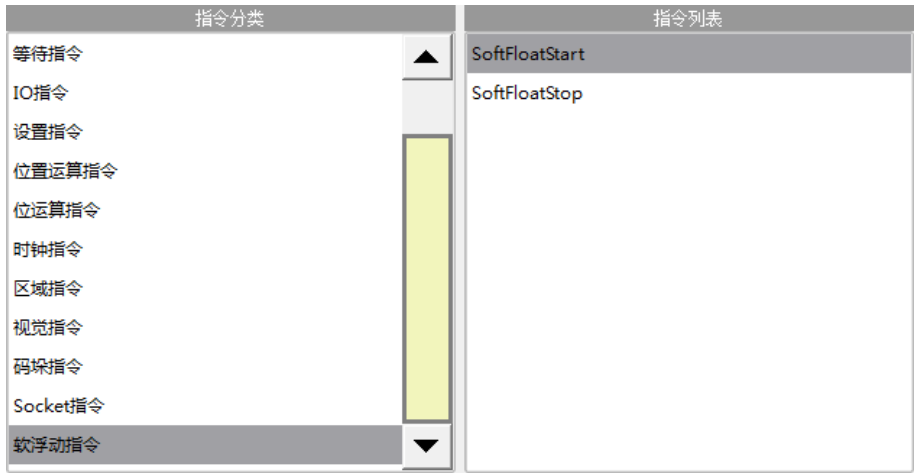
取消

- 目标寄存器 ID 号：想要修改的寄存器 ID 号，类型 int。
- 目标寄存器值：想要修改的寄存器值，类型 INT。

6.17 软浮动指令

软浮动指令列表如图 6-68 所示。

图6-68 软浮动指令列表



SoftFloatStart - 软浮动开始指令

软浮动开始指令。开启软浮动功能并设置软浮动参数。



- 软浮动类型：
 - 直线软浮动。目前只支持直线软浮动。
- 软浮动参考坐标系：
 - WORLD：世界坐标系
 - TOOL：工具坐标系
 - USER：用户坐标系
- 坐标系号：当选择参考坐标系为 WORLD 时，坐标系号默认为空，不显示。当参考坐标系为 TOOL 或者 USER 时，在下拉列表中选择已有的坐标系号或在弹出的对话框中新建并示教一个坐标系号。
- 软浮动方向：
 - X：所选参考坐标系或坐标系号的 X 方向
 - Y：所选参考坐标系或坐标系号的 Y 方向
 - Z：所选参考坐标系或坐标系号的 Z 方向
- 软浮动灵敏度：软浮动柔顺度参数。
 - HIGH：柔顺等级最高

- MidHigh：柔顺等级次高
- MID：中等柔顺
- MidLow：柔顺等级次低
- LOW：柔顺等级最低

其中，设定软浮动方向的说明(详见软浮动功能说明)。

```
SoftFloatStart ("CART","TOOL",tool1,"Y","MID")
```

SoftFloatStop - 软浮动结束指令

软浮动结束指令，关闭软浮动功能。
完成软浮动功能后，调用此指令关闭软浮动功能。

```
SoftFloatStop ()
```

6.18 负载在线辨识指令

机器人负载在线辨识功能，该功能是为在线获取机器人法兰端安装的负载的动力学信息（包括负载质量、质心、惯性张量），辨识所得的负载动力学信息用于机器人动力学模型计算，可改善机器人运行性能，以及改善因负载信息不匹配引起的关节超力矩报警、碰撞检测误报警、机器人运动过程中抖动等现象。

图6-69 负载在线辨识指令



StartMdIlog - 开始采集动力学模型辨识数据指令

数据保存类型

"Robot"

- 数据保存类型：
 - Robot：可选性采集数据，采集机器人本体即未安装负载时的运动数据。使用“Robot”类型数据进行辨识，有助于改善负载辨识的准确度，但该项会延长辨识计算用时。
 - WithLoad：必须采集数据，采集机器人本体有负载安装在法兰端时的运动数据；
 - ValidData：可选性采集数据，需要验证模型准确度时选择此项，采集机器人装有负载的运动数据。使用“ValidData”类型数据，可以在辨识计算的同时验证辨识的模型准确度，但该项会延长辨识计算用时。

说明

- 负载辨识数据采集，选择保存数据类型“Robot”、“WithLoad”时，辨识轨迹所用运动指令需为 MovJ，类型“ValidData”验证轨迹运动指令可使用 MovJ、MovL、MovC 等组合指令。
- 触发指令不支持单步及手动模式。

StopMdlLog - 终止采集动力学模型辨识数据指令

说明

- 必须配合 StartMdllog()指令一起使用。
- StopMdllog()执行结束后，将在“runtime\config”目录下生成和指令 StartMdllog()参数即数据保存类型同名的“.txt”文件，且文档可由 EScope 软件打开查看曲线。
- 触发指令不支持单步或手动模式。

CalDynMdl - 设置模型参数辨识指令

机器人负载在线辨识计算指令，用于辨识机器人负载信息。

负载辨识所需数据采集结束后，执行该指令即开始进行机器人负载信息的辨识计算，辨识过程需要 30~60 分钟的时间，待辨识结束将弹框显示模型验证评分结果。

辨识的模型类型	"TypeTool"	⌵
工具变量	G: TOOL1	⌵
数据间隔长度	0	⌵
是否模型验证	"Disable"	⌵

执行该功能指令需设置指令参数：辨识的模型类型、工件负载变量、数据的间隔长度、是否模型验证，指令参数详细介绍如下：

- 辨识的模型类型：
 - “TypePayload” 类型用于辨识负载参数；
 - “TypeTool” 类型用于辨识工具参数；
- 工件负载变量：用于保存辨识结果的变量（PAYLOAD 负载变量或 TOOL 工具变量）；
- 数据间隔长度：设置从原始数据中抽取有效辨识数据的间隔参数，设置值越大即辨识所用数据量越少，负载辨识结果准确度也将越低，范围：[0, 100]；
- 是否模型验证：负载辨识时是否需要进行模型验证，设置为“Enable 1”时，即需要模型验证，则应设置数据采集指令 StartMdlLog()选择“ValidData”类型采集一组模型验证数据。

说明

- 触发指令不支持单步，及手动模式。
- 负载在线辨识功能目前仅支持通用六轴机型。

- 负载在线辨识功能的运行需保证控制器系统 C 盘剩余空间大于 1GB，Windows 内存大于 1GB，否则该功能无法正常运行；

6.19 字符串指令

字符串指令列表如图 6-70 所示。

图6-70 字符串指令列表

指令分类	指令列表
时钟指令	byte
区域指令	find2
视觉指令	len
码垛指令	char
阵列指令	gsub
Socket指令	sub
ModbusTcp指令	lower
软浮动指令	upper
字符串指令	reverse
冲压流程指令	findEnd
冲压夹具指令	format

byte

取字符串第 n 位的字符的 ASCII 码。

- 参数 1: string 型变量或常量。
- 参数 2: int 型变量或常量。
- 函数返回值: int 型常量。

find2

返回字符串中子字符串的位置。

- 参数 1: string 型变量或常量。
- 参数 2: string 型变量或常量
- 函数返回值: int 型常量。(当找不到对应字符或字符串时, 返回值为-1)

len

计算字符串长度。

- 参数 1: string 型变量或常量。
- 函数返回值: int 型常量。

char

返回 ASCII 码对应的字符。

- 参数 1: int 型变量或常量。
- 函数返回值: string 型常量。

gsub

在字符串 s 内搜索 a 子字符串, 并用字符串 b 替换 a。

- 参数 1: string 型变量或常量。
- 参数 2: string 型变量或常量。
- 参数 3: string 型变量或常量。
- 函数返回值: string 型常量。

lower

返回字符串的小写格式。

- 参数 1: string 型变量或常量。
- 函数返回值: string 型常量。

upper

返回字符串的大写格式。

- 参数 1: string 型变量或常量。
- 函数返回值: string 型常量。

reverse

字符串反转指令，将字符串反转后返回。

- 参数 1: 待反转的字符串，string 型变量或常量。
- 函数返回值: 反转后的字符串，string 型变量。

findEnd

字符串反向查找指令，在字符串中找到指定字符串最后出现的位置，并返回索引号。

- 参数 1: 被查找的源字符串，string 型变量或常量。
- 参数 2: 待查找的指定字符串，string 型变量或常量。
- 函数返回值: 查找后的索引号，int 型变量。

format

格式化字符串指令，通过在参数 1 中传入合理的格式化控制符，其后带上任意多个参数来填充这个格式化控制符，并将填充后的数据返回。

- 参数 1: 字符串格式，string 型变量或常量。
- 参数 2: 待填充格式化控制符的参数，string/real/int 型变量或常量。
- 参数 3: 待填充格式化控制符的参数，string/real/int 型变量或常量。
- ... (参数无限制，总长度只要不超过单个指令字符串长度即可)
- 函数返回值: 成功分割并保存到数组中的个数，int 型变量

格式化字符串以%开头，支持以下用法:

%c - 接受一个数字,并将其转化为 ASCII 码表中对应的字符

%d, %i - 接受一个数字并将其转化为有符号的整数格式

%o - 接受一个数字并将其转化为八进制数格式

%u - 接受一个数字并将其转化为无符号整数格式

%x - 接受一个数字并将其转化为十六进制数格式,使用小写字母 x

%X - 接受一个数字并将其转化为十六进制数格式,使用大写字母 X

%f - 接受一个数字并将其转化为浮点数格式

%s - 接受一个字符串并按照给定的参数格式化该字符串

样例：

<code>format("%c: %c", 83)</code>	输出 S
<code>format("%+d", 17.0)</code>	输出+17
<code>format("%05d", 17)</code>	输出 00017
<code>format("%o", 17)</code>	输出 21
<code>format("%u", 3.14)</code>	输出 3
<code>format("%x", 13)</code>	输出 d
<code>format("%X", 13)</code>	输出 D
<code>format("%6.3f", 13)</code>	输出 13.000
<code>format("%s", "monkey")</code>	输出 monkey
<code>format("%10s", "monkey")</code>	输出 monkey

getAt

单个字符串获取指令，获取某一位的字符串数据，并将获取后的数据返回。

- 参数 1：待截取的字符串，string 型变量或常量。
- 参数 2：待获取的位置，int 型变量或常量。
- 函数返回值：获取后的字符串，string 型变量。

left

字符串取左指令，从字符串左侧开始截取指定数量的字符串，并将截取后的数据返回。

- 参数 1：待截取的字符串，string 型变量或常量。
- 参数 2：待截取的数量，int 型变量或常量。
- 函数返回值：截取后的字符串，string 型变量。

right

字符串取右指令，从字符串右侧开始截取指定数量的字符串，并将截取后的数据返回。

- 参数 1：待截取的字符串，string 型变量或常量。
- 参数 2：待截取的数量，int 型变量或常量。
- 函数返回值：截取后的字符串，string 型变量。

strcmp

字符串比较指令，返回值为第一个不同字符之间的 ASCII 码差值。

- 参数 1：参与比较的字符串数据，string 型变量或常量。
- 参数 2：被比较的字符串数据，string 型变量或常量。
- 函数返回值：返回的 ASCII 码值，int 型变量。

trimLeft

字符串左修剪指令，去掉字符串左边的空格，并将修改过后的字符串数据返回。

- 参数 1：待修剪的字符串，string 型变量或常量。

- 函数返回值：修剪后的字符串，string 型变量。

trimRight

字符串右修剪指令，去掉字符串右边的空格，并将修改过后的字符串数据返回。

- 参数 1：待修剪的字符串，string 型变量或常量。
- 函数返回值：修剪后的字符串，string 型变量。

IToStr

整型数据转字符串指令，将整型数据转换成字符串类型数据，并返回转换后的字符串。

- 参数 1：待转换的整型数据，int 型变量或常量。
- 函数返回值：转换后的字符串数据，string 型变量。

RToStr

实数型参数转字符串指令，将实数数据转换成字符串类型数据，并返回转换后的字符串。

- 参数 1：待转换的实数型数据，REAL 型变量或常量。
- 函数返回值：转换后的字符串数据，string 型变量。

StrToI

字符串转整型指令，将字符串数据转换成整数类型数据，并返回转换后的整型数据。

- 参数 1：待转换的字符串数据，string 型变量或常量。
- 函数返回值：转换后的整型数据，int 型变量。

StrToR

字符串转实数型数据指令，将字符串数据转换成实数类型数据，并返回转换后的实数。

- 参数 1：待转换的字符串数据，string 型变量或常量。
- 函数返回值：转换后的实数型数据，REAL 型变量。

APosToStr

APOS 点数据转成字符串指令，将 apos 点按照指定的格式转换成字符串，并返回转换后的字符串数据。转换后 apos 点格式为 “0; 1,2,3,4,5,6,0,0,0,0,0,0,0,0;”，其中分号前面分别代表 apos 点类型、坐标值。

- 参数 1：待转换的 APOS 点数据，APOS 型变量。
- 函数返回值：转换后的字符串数据，string 型变量。

CPosToStr

CPOS 点数据转成字符串指令，将 cpos 点按照指定的格式转换成字符串，并返回转换后的字符串数据。转换后 cpos 点格式为 “1;0,0,0,0,0,0;1,2,3,4,5,6,0,0,0,0,0,0,0,0;”，分号前面分别代表 cpos 点类型、cfg 参数、坐标值。

- 参数 1：待转换的 CPOS 点数据，CPOS 型变量。
- 函数返回值：转换后的字符串数据，string 型变量

TranStrToInt

整数型字符串数组获取指令，根据传入的分割符将字符串分割，并将分割后的值转换到整型一维数组中，同时返回有效分割数量。

- 参数 1: 待分割的字符串, string 型变量或常量。
- 参数 2: 分割符, string 型变量或常量。
- 参数 3: 以数组的形式存储分割后的整型变量, IntOneArray 型变量。
- 函数返回值: 成功分割并保存到数组中的个数, int 型变量

样例:

```
P:INT0.value = TranStrToInt("1_2_3er", "_", G:IntOneArray0) // 该指令传入的字符为"1_2_3er", 执行后系统按照传入的分割符'_'来分割提取数据, 分割后的数据分别按照 1、2、0 的顺序存在一维整型数组变量 IntOneArray0 中, 并将分割个数 3 赋给 INT0.value。
```

TranStrToReal

实数型字符串数组获取指令, 根据传入的分割符将字符串分割, 并将分割后的值转换到实数型一维数组中, 同时返回有效分割数量。

- 参数 1: 待分割的字符串, string 型变量或常量。
- 参数 2: 分割符, string 型变量或常量。
- 参数 3: 以数组的形式存储分割后的实数型变量, RealOneArray 型变量。
- 函数返回值: 成功分割并保存到数组中的个数, int 型变量

样例:

```
P:INT0.value = TranStrToReal("1.1_2.2_3er", "_", G:IntOneArray0) // 该指令传入的字符为"1.1_2.2_3er", 执行后系统按照传入的分割符'_'来分割提取数据, 分割后的数据分别按照 1.1、2.2、0 的顺序存在一维整型数组变量 RealOneArray0 中, 并将分割个数 3 赋给 INT0.value。
```

TranStrToApos

apos 点数据字符串数组获取指令, 根据传入的分割符将字符串分割, 并将分割后的值转换到 apos 点变量中, 同时返回转换成功与否的状态。需要转换的字符串格式必须为 “0; 1,2,3,4,5,6,0,0,0,0,0,0,0,0;”, 其中分号前面分别代表 apos 点类型、坐标值, 最后一个分号后面可以带任意字符串。

- 参数 1: 待分割的字符串, string 型变量或常量。
- 参数 2: 分割符, string 型变量或常量。
- 参数 3: 以 apos 变量的形式存储分割后的 apos 点数据, APOS 型变量。
- 函数返回值: 返回提取成功与否的状态, int 型变量

样例:

```
P:INT0.value = TranStrToApos("0; 21,22,23,24,25,26,0,0,0,0,0,0,0,0;$qwqewe", "$", G: P0) // 该指令传入的字符为"0; 21,22,23,24,25,26,0,0,0,0,0,0,0,0;$qwqewe", 执行后系统按照传入的分割符'$'来分割提取数据, '$'后面的数据会被舍弃。分割后的数据分别按照 21、22、23、24、25、26...的顺序存在 P0 的 a1、a2、a3、a4、a5、a6...中, 并将成功返回值 0 赋给 INT0.value。
```

TranStrToCpos

cpos 点数据字符串数组获取指令, 根据传入的分割符将字符串分割, 并将分割后的值转换到 cpos 点变量中, 同时返回转换成功与否的状态。需要转换的字符串格式必须为 “1;0,0,0,0,0;1,2,3,4,5,6,0,0,0,0,0,0,0;”, 分号前面分别代表 cpos 点类型、cfg 参数、坐标值。

- 参数 1: 待分割的字符串, string 型变量或常量。
- 参数 2: 分割符, string 型变量或常量。
- 参数 3: 以 cpos 变量的形式存储分割后的 cpos 点数据, CPOS 型变量。
- 函数返回值: 返回提取成功与否的状态, int 型变量

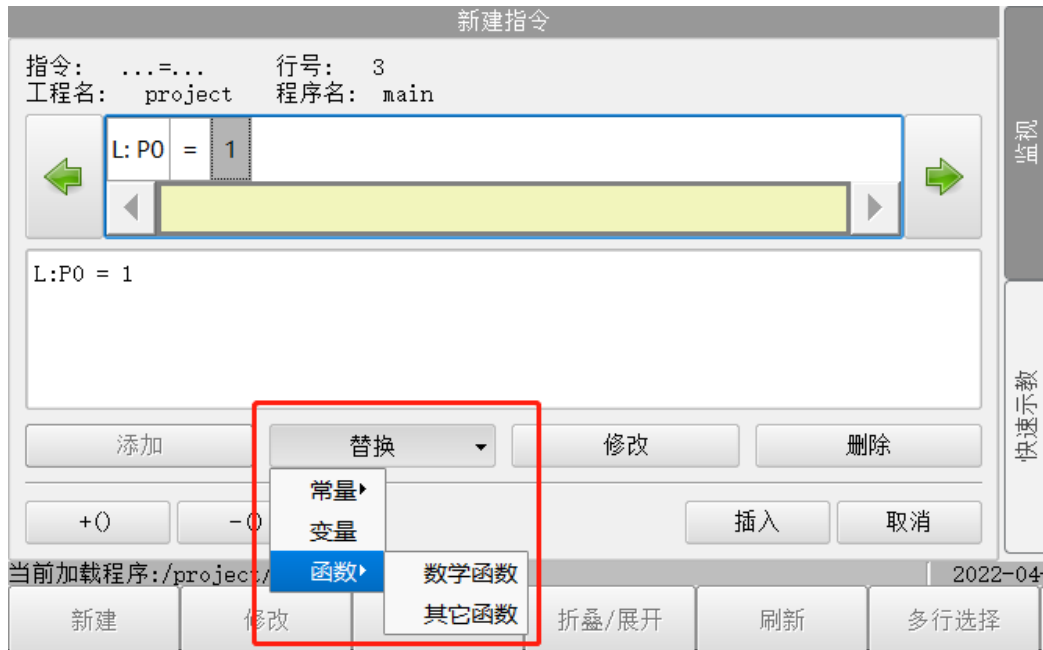
样例:

- ```
P:INT0.value = TranStrToCpos("1;0,0,0,0,0;21,22,23,24,25,26,0,0,0,0,0,0,0;$qwqewe", "$", G: P0) // 该指令传入的字符为"0; 0,0,0,0,0;21,22,23,24,25,26,0,0,0,0,0,0,0;$qwqewe", 执行后系统按照传入的分割符'$'来
```

分割提取数据，'\$'后面的数据会被舍弃。分割后第一个";"后面的"0,0,0,0,0"会按顺序存入 cfg 参数中，第二个";"后面的数据按照 21、22、23、24、25、26...的顺序存在 P0 的 x、y、z、a、b、c...中，并将成功返回值 0 赋给 INTO.value。

## 6.20 数学运算函数

在“IF”指令和“...=...”指令中，可能会使用到数学运算函数。本节是对所能使用到的“数学函数”进行说明。



### sin - 正弦三角函数

正弦三角函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### cos - 余弦三角函数

余弦三角函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### tan - 正切三角函数

正切三角函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### asin - 反正弦三角函数

反正弦三角函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### acos - 反余弦三角函数

反余弦三角函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### atan - 反正切三角函数

反正切三角函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### atan2 - X/Y 反正切函数

X/Y 反正切值函数，返回 X 轴到(x, y)点的弧度值。

- 参数 1: int 或 real 型变量或常量。
- 参数 2: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### sinh - 双曲正弦函数

双曲正弦函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### cosh - 双曲余弦函数

双曲余弦函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### tanh - 双曲正切函数

双曲正切函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### log - 对数函数

自然对数函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### log10 - 以 10 为底的对数函数

以 10 为底的对数函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### sqrt - 开平方根函数

开平方根函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### exp - 以 e 为底的指数函数

以 e 为底的指数函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### pow - 指数函数

指数函数。

- 参数 1: int 或 real 型变量或常量，底数。
- 参数 2: int 或 real 型变量或常量，指数。
- 函数返回值: real 型常量。

### deg - 弧度转角度函数

弧度转角度函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### rad - 角度函数

角度转弧度函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

### fmod - 取余函数

取余函数。

- 参数 1: int 或 real 型变量或常量，被除数。
- 参数 2: int 或 real 型变量或常量，除数。
- 函数返回值: real 型常量。

### floor - 向下取值函数

向下取整函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: int 型常量。

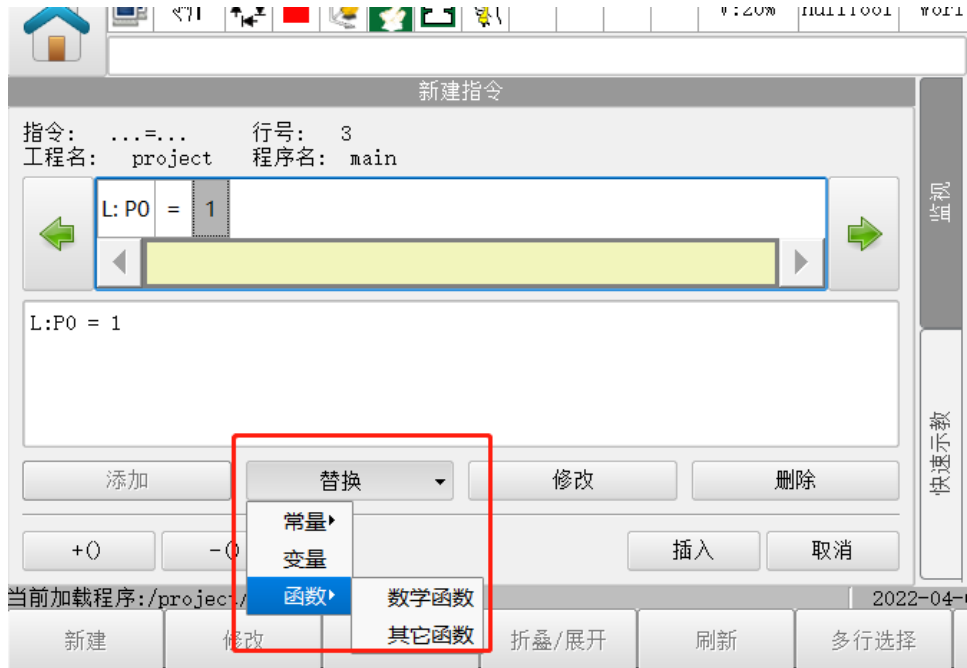
### random - 随机取整

2 个参数之间取随机整数。

- 参数 1: int 型变量或常量。
- 参数 2: int 型变量或常量。
- 函数返回值: int 型常量。

## 6.21 其他函数

在“IF”指令和“...=...”指令中，可能会使用到其他函数。本节是对所能使用到的“其他函数”进行说明。



### SimAiToReal

将 SimAi 变量转换成 Real 类型的函数，转换后可参与运算。

- 参数：SimAi
- 函数返回值：Real 类型常量。

### SimAoToReal

将 SimAo 变量转换成 Real 类型的函数，转换后可参与运算。

- 参数：SimAo
- 函数返回值：Real 类型常量。



## 第 7 章 高级功能

### 7.1 多类型区域监控

#### 功能概述

通过设置安全区域，可以实时监控多种工况下机器人的位置，降低碰撞发生的可能性。

多类型区域监控包含以下三种主要类型：

- 工作区域（WorkArea）：

机器人所有 CheckPoint（TCP）只允许在 WorkArea 内运动。当发生违反工作区域规则情况时有以下两种处理方法：

- 机器人离开工作区域时，程序停止，报警
- 机器人在工作区域内，正常运行，若配置了输出端口使能，输出一个信号

- 禁止区域（BlockedArea）

机器人所有 CheckPoint（TCP）不允许在 BlockedArea 内运动。当发生违反禁止区域规则情况时有以下两种处理方法：

- 机器人进入禁止区域时，程序停止，报警
- 机器人不在禁止区域内，正常运行，若配置了输出端口使能，输出一个信号

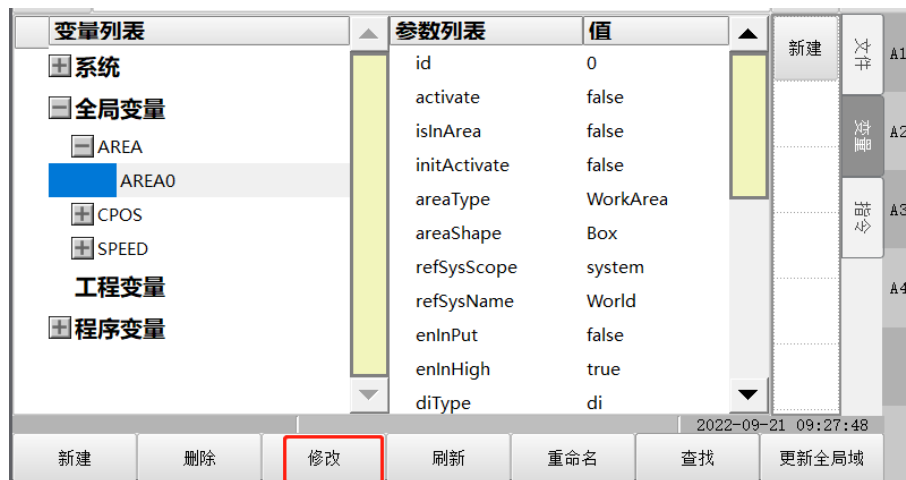
- 信号工作区域（SignalArea）

机器人所有 CheckPoint（TCP）允许在任何区域运动。当发生区域规则情况时有以下两种处理方法：

- 机器人进入信号工作区域时，继续运行程序，若配置了输出端口使能，输出一个信号
- 机器人离开信号工作区域时，继续运行程序，若配置了输出端口使能，输出一个信号

#### 标准区域监控使用方法

步骤 1 进入程序数据界面，点击全局域中要设置的标准区域变量，如 AREA0。点击“修改”，将进入设置区域变量的界面。



步骤 2 默认情况时，标准区域变量为“未激活”状态，用户可点击“激活”，来激活该标准区域变量。

步骤 3 点击“修改”，如图 7-1 所示。

图7-1 设置标准区域变量界面---盒状、圆柱



图7-2 设置标准区域变量界面---轴干涉区



步骤 4 按照如下说明修改该界面的参数。

| 参数名称 | 参数说明                                                                                                                                           |
|------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 区域名  | 显示标准区域变量名称<br><br>默认无法修改                                                                                                                       |
| 状态   | <ul style="list-style-type: none"><li>在该区域激活后，判断监控点是在区域内部还是区域外部</li></ul>                                                                      |
| 类型   | 选择该区域的类型：<br><br>取值范围： <ul style="list-style-type: none"><li>• WorkArea（工作区域）</li><li>• BlockedArea（禁止区域）</li><li>• SignalArea（信号区域）</li></ul> |

|                      |                                                                                                                                        |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 监控端                  | <p>选择区域监控点</p> <p>取值范围：</p> <ul style="list-style-type: none"> <li>• 法兰端（Flange）</li> <li>• 机器人工具末端（Tcp）</li> </ul>                    |
| 参考坐标系                | <p>选择该区域的参考坐标系。</p> <p>默认是世界坐标系</p>                                                                                                    |
| 形状                   | <p>选择区域的形状</p> <p>取值范围：</p> <ul style="list-style-type: none"> <li>• 盒状（BOX）</li> <li>• 圆柱（CYLINDER）</li> <li>• 轴干涉区（JOINT）</li> </ul> |
| 输入端口                 | <ol style="list-style-type: none"> <li>1. 勾选“端口使能”</li> <li>2. 选择需要激活该区域变量的输入端口</li> <li>3. 并设定“低位有效”或“高位有效”。</li> </ol>               |
| 输出端口                 | <ol style="list-style-type: none"> <li>1. 勾选“端口使能”</li> <li>2. 选择需要激活该区域变量的输出端口</li> <li>3. 并设定“低位有效”或“高位有效”。</li> </ol>               |
| X、Y、Z、lenX、lenY、lenZ | <p>设定区域的起点坐标以及区域的大小。</p> <p>起点坐标是相对于参考坐标系。</p> <p>设定盒状或圆柱区域的起点坐标以及区域的大小（对应图 7-1）。起点坐标是相对于参考坐标系</p>                                     |
| 轴、posMax、posMin      | <p>设定轴干涉区参数（对应图 7-2），即检测的轴号、轴运动的最大和最小范围。</p>                                                                                           |

步骤 5 点击“确认”，完成标准区域变量的设定。如图 7-3 所示。

图7-3 标准区域变量界面

| 变量列表                                                                                                                                                                                                           | 参数列表         | 值        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----------|
| <div><div>+</div>系统</div> <div><div>■</div>全局变量</div> <div><div>■</div>AREA</div> <div><div>■</div>AREA0</div> <div><div>+</div>PAYLOAD</div> <div><div>+</div>USERCOORD</div> <div>工程变量</div> <div>程序变量</div> | id           | 0        |
|                                                                                                                                                                                                                | activate     | false    |
|                                                                                                                                                                                                                | isInArea     | false    |
|                                                                                                                                                                                                                | initActivate | false    |
|                                                                                                                                                                                                                | areaType     | WorkArea |
|                                                                                                                                                                                                                | areaShape    | Box      |
|                                                                                                                                                                                                                | refSysScope  | system   |
|                                                                                                                                                                                                                | refSysName   | World    |
|                                                                                                                                                                                                                | enInPut      | false    |
|                                                                                                                                                                                                                | enInHigh     | true     |
|                                                                                                                                                                                                                | diType       | di       |

说明

1.

当准区域激活或冻结功能，可通过变量配置界面、指令或输入端口（配置使能的输入端口号）执行激活或冻结操作。
2.

标准区域输入端口使能设置后，可通过输入端口号的沿触发激活或冻结(与标准区域激活指令、标准区域冻结指令共存，且以后触发的为准) 区域。
3.

当标准区域被激活后，默认为设置此标准区域属性为自动激活。
4.

自动激活属性的说明：系统重启加载标准区域时，当标准区域属性是自动激活，输入端口使能未设置时，该区域自动激活；当标准区域属性是自动激活，输入端口使能设置时，根据设置的输入端口号的电平状态决定是否激活此区域。
5.

当标准区域处于激活状态，实时输出机器人是否在设置区域中的状态；输出端口使能设置后，可根据机器人监控端是否在设置区域中的状态输出信号。
6.

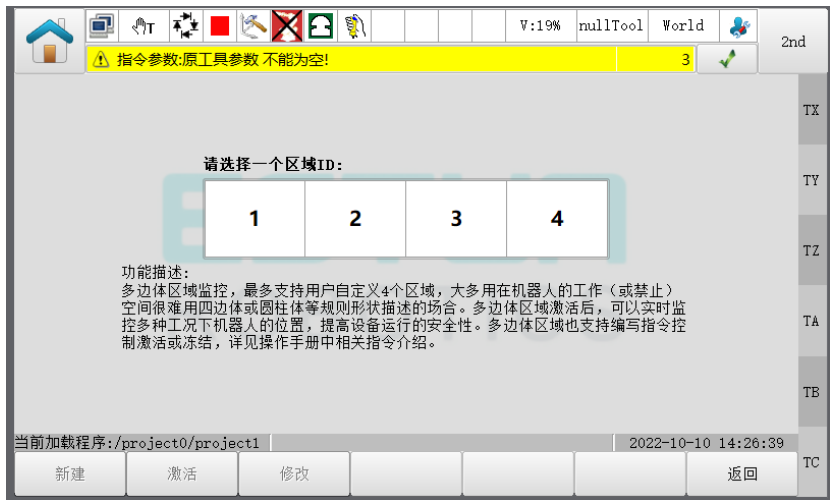
当标准区域处于激活状态，区域类型为工作区域或禁止区域时，机器人运动时将周期性检测机器人监控端位置，若在工作区域外或在禁止区域内，则产生相应的报警。

多边形区域监控使用方法

使用“多边形区域”监控功能时，首先需要对多边形区域进行配置。

步骤 1 点击“用户应用>多边形区域”进入多边形区域插件界面，如图 7-4 所示，目前最大支持 4 个多边形区域；

图7-4 多边形区域插件界面

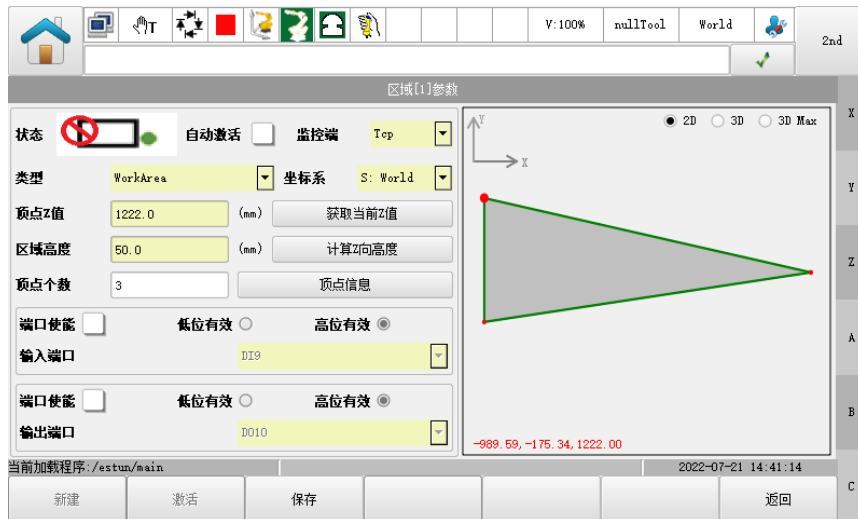


步骤 2 选择需要配置的区域 ID，进入配置界面。

1.

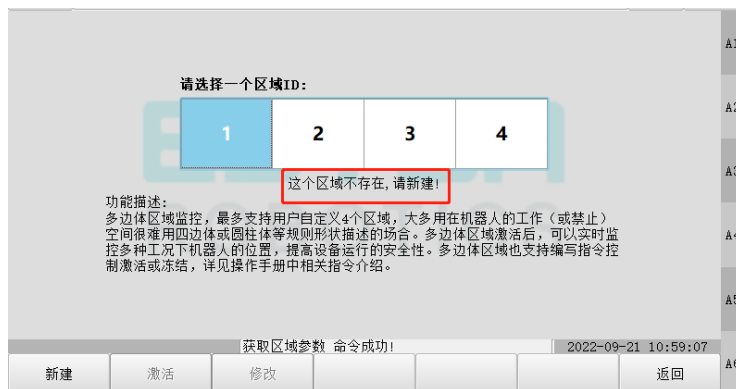
若点击区域 ID 后，进入配置界面，如图 7-5 所示；表示该区域已配置多边区域，请参考步骤 3 对参数进行修改。

图7-5 多边体区域变量配置界面



2. 若点击区域 ID 后，出现如图 7-6 所示，请点击“新建”按钮，进入配置界面后，再点击“修改”按钮，对参考步骤 3 对参数进行修改。

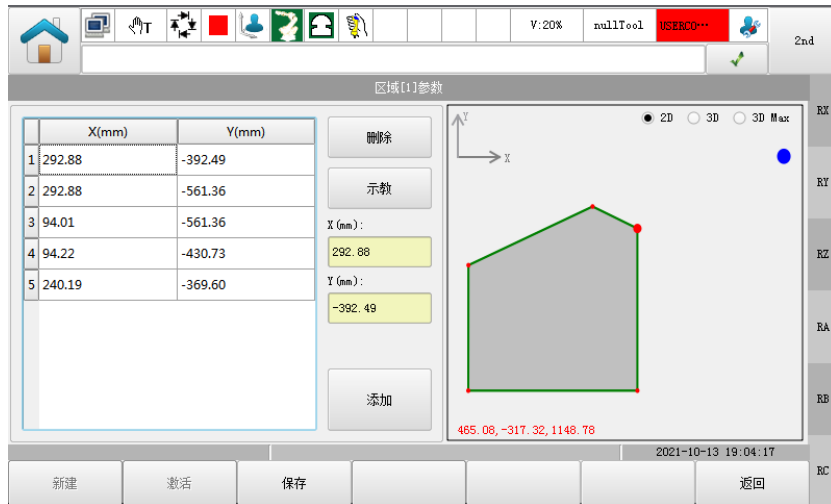
图7-6 多边体区域变量配置界面



步骤 3 按照如下说明修改该界面的参数。

- 状态：在该区域激活后，判断监控点是在区域内部还是区域外部。
- 自动激活：当勾选自动激活后，在控制器启动后将自动激活该区域。
- 监控端：选择区域监控点，目前可选择机器人法兰端（Flange）和机器人工具末端（Tcp）。
- 类型：选择该区域的类型。
- 坐标系：选择该区域的参考坐标系，默认为世界坐标系。
- 顶点 Z 值：表示区域在坐标系 Z 方向上的起始位置，可以手动输入，也可以点击“获取当前 Z 值”按键来获取，结果为当前坐标系下 Z 轴坐标值。
- 区域高度：表示区域在坐标系 Z 方向上的高度，可以手动输入，也可以点击“计算 Z 向高度”来计算当前区域高度，结果为当前坐标系下 Z 轴坐标值减去顶点高度值。
- 顶点个数：表示多边体区域底面顶点的个数，点击“顶点信息”后，进入顶点配置界面，如图 7-7 所示；点击“添加”按键，将自动添加当前用户坐标系下的 TCP 位置；若需要修改多边体的某个顶点，可选择对应顶点，通过点击“示教”或输入 XY 坐标进行修改；若需要删除多边体的某个顶点，可选择对应顶点，点击“删除”即可；在界面右侧支持预览多边体的平面效果（2D）、立体几何效果（3D）、以及仅放大多边体区域的立体几何效果（3D Max）。预览支持同步显示监控点位置（图中蓝色圆点），在 2D 模式下，仅显示为监控点位置在 XY 平面上的位置，切换至 3D 效果可预览实际监控点的相对位置；
- 输入端口：勾选“端口使能”选择需要激活该区域变量的输入端口，并设定“低位有效”或“高位有效”。
- 输出端口：勾选“端口使能”选择需要激活该区域变量的输出端口，并设定“低位有效”或“高位有效”。

图7-7 多边形区域顶点配置界面



步骤 4 完成设置后点击“保存”。

步骤 5 在指令中使用多边形区域时，需要创建 POLYHEDRON 类型变量，并指定其中的参数 index 为正确的多边形区域 ID；创建完成后，可通过 PolyhedronAreaActivate 指令或 PolyhedronAreaDeactivate 指令执行多边形区域的激活或冻结操作；

#### 说明

1. 当多边形区域处于激活状态，且区域类型为工作区域或禁止区域时，在运行状态下系统将周期性检测机器人监控点位置，实时输出机器人是否在设置区域中的状态。
2. 当标准区域处于激活状态，且区域类型为信号区域时，无论是否上励磁，系统都会周期性检测机器人监控点位置，实时输出机器人是否在设置区域中的状态。
3. 设置多边形区域的自动激活参数后，不会立即生效，系统重启时加载多边形参数再进行判断，当 DI 端口未使能时，该区域自动激活；当 DI 端口使能时，该区域根据 DI 的电平状态决定是否激活此区域。
4. 多边形区域激活与冻结的逻辑，输入/输出端口的使用状态与标准区域监控功能相同。
5. 多边形区域激活或冻结功能，可通过变量配置界面、指令或输入端口（如配置）执行激活或冻结操作。

---结束

## 7.2 碰撞检测

### 7.2.1 碰撞检测

#### 功能概述

碰撞检测功能无需外力传感器，实时检测机器人是否发生碰撞，若检测到碰撞，机器人立即报警并停止工作，进而可以提供安全保护，有效降低机器等设备的损失。用户可以根据现场应用需要，打开或关闭碰撞检测功能，出于安全性考虑，建议用户打开碰撞检测功能。

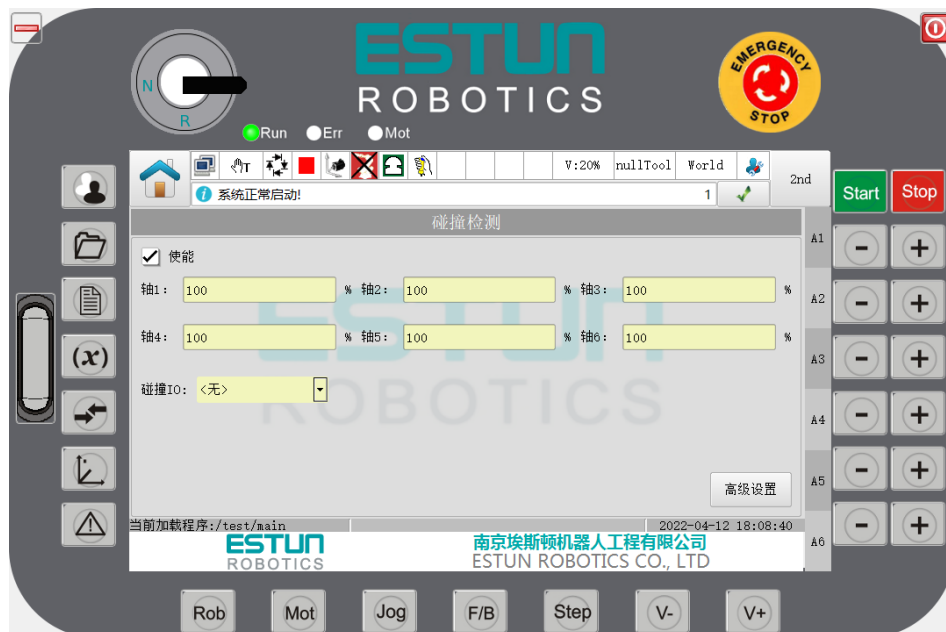
#### 使用方法

##### 说明

- 若当前机型不支持碰撞检测功能，界面将显示“当前机型不支持碰撞检测”。
- 本功能基于负载信息及设备信息来检测碰撞，所以必须正确设定负载信息以避免误报或影响检测灵敏度。关于负载参数的设置，请参看设置负载参数“PAYLOAD0”（6.5）和“SetPayload”指令（6.5）。
- 不同工况，不同运行条件，同样的参数设置可能会带来不同的碰撞灵敏度，因此也可能会出现碰撞误报警的情况，若发生碰撞误报警，根据报警信息，适当降低碰撞轴的灵敏度，将数值调高，直至不再发生碰撞误报警。

步骤 1 在系统主页中选择“通用设置>碰撞检测”，进入设置界面，如图 7-8 所示。

图7-8 碰撞检测配置界面



#### 参数介绍

- 使能复选框：控制碰撞检测功能的开关
  - 勾选：打开碰撞检测功能。
  - 未勾选：关闭碰撞检测功能。
- 轴 1~轴 6 的值：设置碰撞检测灵敏度
  - 默认值：100
  - 值越大：碰撞检测灵敏度越低

值越小：碰撞检测灵敏度越高

碰撞 IO：当碰撞回退完成时，系统会输出到 DO 口。<无>表示不输出，为其他值时则给对应 DO 口输出高电平。

## 7.2.2 碰撞检测自调整

### 功能说明

根据机器人实际运行状态生成匹配的碰撞检测参数，可以提高碰撞检测的灵敏度和降低误报警的概率。

通过功能设置可对目标轨迹的运行数据进行记录，对记录的数据进行计算得到目标轨迹下的碰撞检测自调整参数，使用得到的自调整参数。

### 使用方法

该功能需要碰撞检测功能处于使能状态并且确保进行碰撞检测自调整的目标轨迹能正常运行。

步骤 1 在系统主页中选择“通用设置>碰撞检测>高级设置”，进入设置界面。

步骤 2 勾选左上方“使能”复选框，如图 7-8 所示。

步骤 3 点击右下方“高级设置”进入“碰撞检测自调整”页面。



步骤 4 选择“标定”分页，点击“开始”按钮将重新进行自调整功能的数据记录，运行机器人目标轨迹，数据采集过程中可通过点击“暂停”按钮中断数据采集，并通过点击“继续”按钮恢复数据采集、请至少运行 1 遍目标速度下的轨迹，之后点击“结束”按钮，结束数据采集。

步骤 5 根据实际需求选择“参数 ID”、“检测方法”、“灵敏度等级”后，点击“计算”。

步骤 6 选择“碰撞检测参数设置”分页，设置碰撞检测的默认参数。





## 参数介绍

- 标定页签：
  - 开始/结束：重新进行自调整功能数据记录/结束数据采集。
  - 暂停/继续：中断数据采集/恢复数据采集。
  - 参数 ID：1~6，用来存放计算出的碰撞检测自调整参数。
  - 检测方法：标准、标准+、标准++，对应检测方法依次增强。
  - 灵敏度等级：5 个档位：最低、低、标准、高、最高，对应灵敏度等级依次增高。
  - 计算：根据设置的参数计算出对应的碰撞检测自调整参数。
- 碰撞检测参数设置
  - 默认碰撞检测参数 ID：出厂默认参数为“DEFAULT”，1 到 6 对应“参数 ID”1 到 6 中的自调整参数。

## 7.3 碰撞回退

机器人与物体发生碰撞后会紧急停止运动，机器人停止运动后仍然保持与物体接触状态。机器人与物体之间存在接触应力，造成机器人与物体的损坏。碰撞回退功能可使机器人发生碰撞后沿原路径返回一段距离，达到卸力的目的。

机器人发生碰撞后，触发碰撞报警，机器人快速停止运动后按照原路径返回，使机器人不与障碍物接触，达到卸力的目的。

第一阶段：机器人快速停止运动，机器人检测到碰撞后立即停止运动。

第二阶段：从碰撞停止点到回退结束点，按照原路径返回。

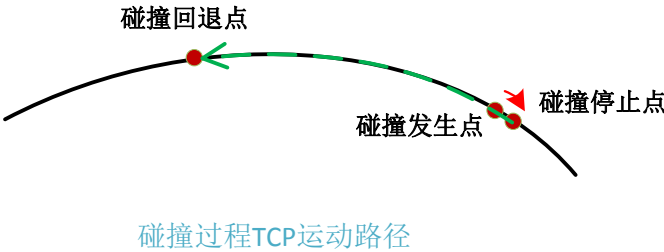


图7-9 碰撞过程 TCP 运动路径

说明

使用该功能需联系厂家。

7.4 振动抑制

功能概述

振动抑制功能用于减弱机器人末端残余振动，使机器人运动平稳、提高定位精度和缩短定位时间。机器人动作轨迹不同，末端残余振动的强弱也存在不同，此外振动还受到运动速度、末端载荷的影响，通常机器人高速运动相对于低速运动，末端载荷重相对于载荷轻，发生振动的可能性更高，振动也会强烈些。

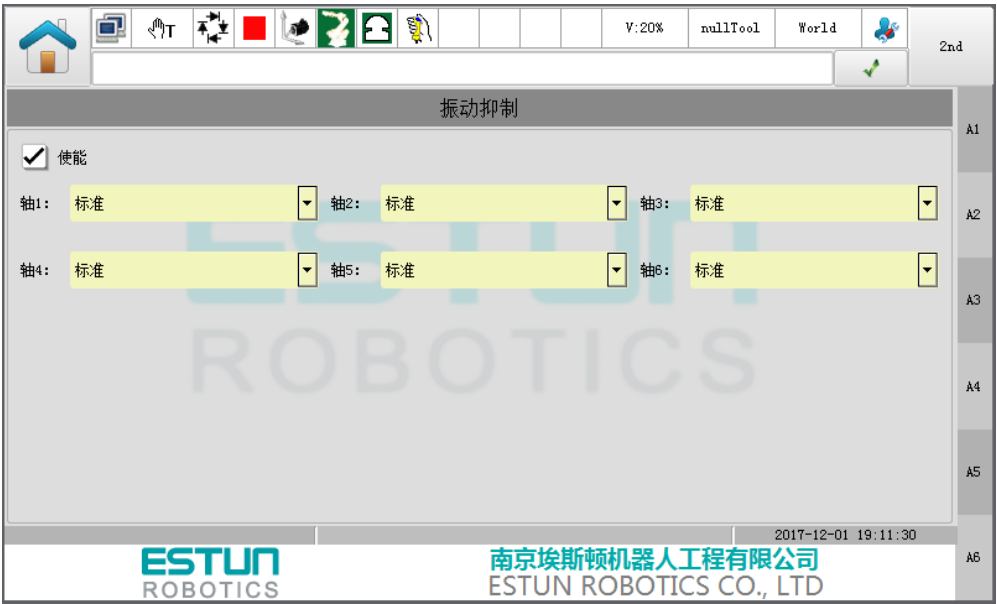
使用方法

说明

在使用振动抑制功能时，必须设置正确的负载参数，负载参数影响控制器对振动抑制的调节。  
关于负载参数的设置，请参看设置负载参数的基本设置界面（4.7.2 ）和 SetPayload 指令（6.5 ）。

步骤 1 在系统主页中选择“通用设置>振动抑制”，进入配置界面如图 7-10 所示。

图7-10 振动抑制配置界面



参数介绍

- 使能复选框：控制振动抑制功能的开关

勾选：打开振动抑制功能。

未勾选：关闭振动抑制功能。

- 轴 1~轴 6 的值：设置振动抑制的强度

默认值：标准

取值范围：共计 5 个档位：低、中低、标准、中高、高。振动抑制效果依次逐步增强。

## 7.5 传送带跟随

### 功能概述

通过传送带跟随功能，可以使机器人跟随传送带上的工件实现同步运动，即工件在传送带上随着传送带的运行而移动时。通过视觉系统或者传感器检测工件的位置信息，获取位置信息后使用指定的交互方式将数据传递给机器人，使机器人能够根据传感器反馈的信息从固定或者移动的传送带上拾取工件，传送带的类型分为直线型和圆盘型。

图7-11 直线传送带跟随系统示意

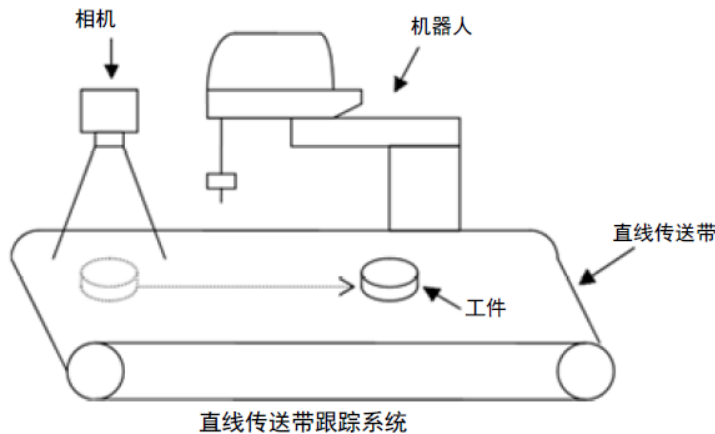
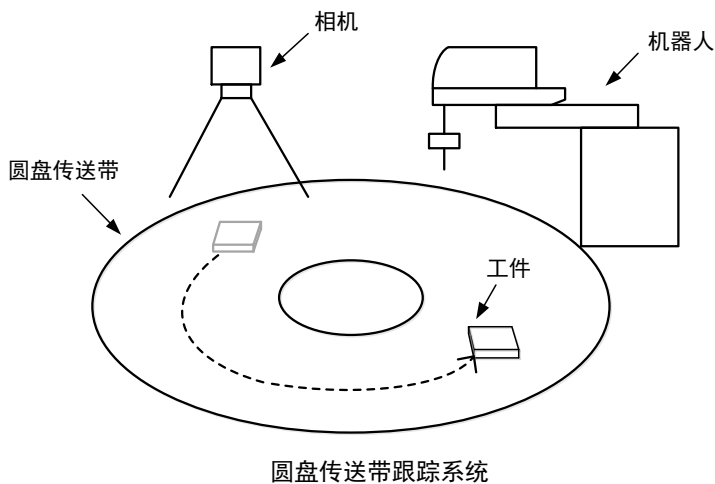


图7-12 圆盘传送带跟随系统示意



### 注意

- 当前系统版本最多支持单机双传送带应用。

通过点击示教编程器“用户应用-传送带跟随”即可进入传送带配置界面，对传送带参数进行配置，目前最多支持 2 个传送带。在该界面中，可以进行传送带使能、脉冲当量查看与标定、传送带位置与速度查看、一键开启/关闭显示未跟随原因，并通过页面切换分别转至“传送带参数”、“跟随参数”、“触发参数”、“视觉参数”、“虚拟传送带”五个配置界面进一步配置。

### 使用方法

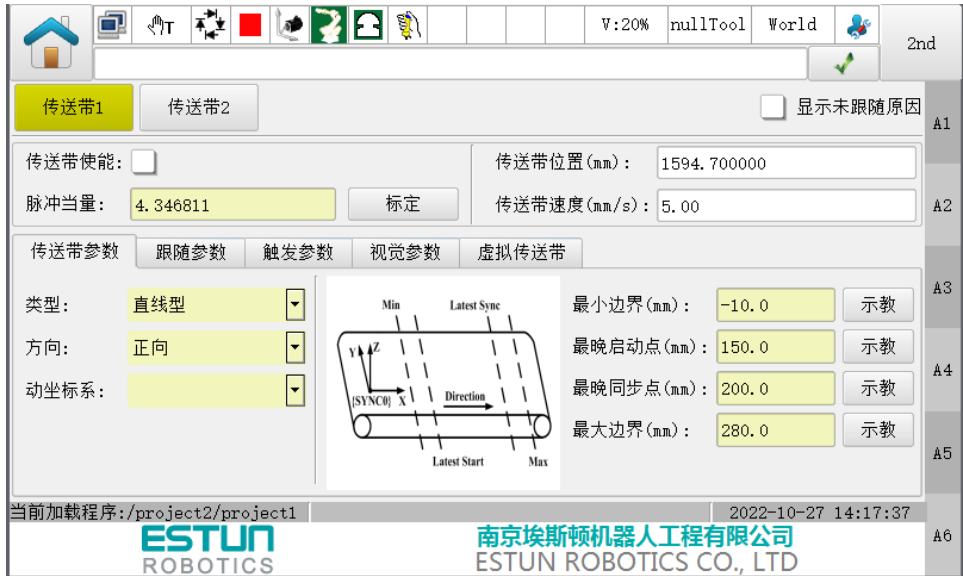


说明

最多支持 2 个传送带。

步骤 1 在系统主页中选择“用户应用>传送带跟随”，进入传送带配置界面。

图7-13 直线传送带参数配置界面



参数介绍

- 显示未跟随原因：
  - 勾选：显示未跟随原因。
  - 未勾选：不显示未跟随原因。
- 传送带使能：控制传送带跟随功能的开关
  - 勾选：打开传送带跟随功能。
  - 未勾选：关闭传送带跟随功能。
- 脉冲当量：
- 传送带位置：
- 传送带速度：

传送带参数配置

选择“传送带参数”，进入传送带参数页签。

- 类型：直线型或圆盘型。

图7-14 直线传送带参数配置界面

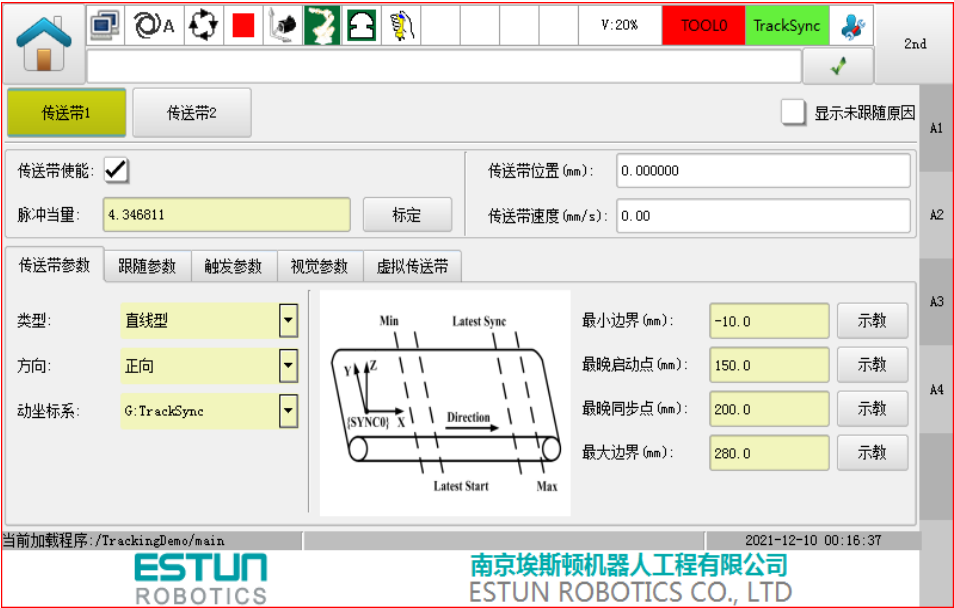


图7-15 圆盘传送带参数配置界面



- 方向：设置传送带的方向。
- 动坐标系：与该传送带绑定的动坐标系。
- 最小边界/最大边界/最晚启动点/最晚同步点：直线型传送带时表示在对应动坐标系下 X 方向的坐标值；圆盘型传送带表示在对应动坐标系 XOY 平面内圆心角的数值。

跟随参数配置

选择“跟随参数”，进入传送带参数配置界面。

图7-16 跟随参数配置界面



- 速度系数/加速度系数/减速度系数：跟随同步过程中的速度/加速度/减速度的比率。
- 跟随允差：跟随同步到位的判断阈值，表示机器人与目标物体的最大位移允许误差。
- X 方向补偿：相对于目标点位 X 方向的工件偏移补偿。
- Y 方向补偿：相对于目标点位 Y 方向的工件偏移补偿。
- A 方向补偿：相对于目标点位 A 方向的工件偏移补偿。

## 触发参数配置

选择“触发参数”，进入触发参数配置界面。

图7-17 触发参数配置界面



- 触发类型：
  - 定时触发：系统会根据配置的触发时间(ms)，周期设置触发输出口输出高电平信号，高电平保持时间为触发时间的 1/2。
  - 外部 IO 触发：当系统检测到触发输入端口有上升沿信号时（检测周期为 4ms），设置触发输出口输出高电平信号，高电平保持时间与触发输入端口保持同步。

- 非相机触发：无视觉操作方式，需要创建一个任务程序：检测输入信号、下发目标信息等。可使用 RUN 程序，编写指令检测触发输入端口有上升沿信号时，调用 SetTargetPos 指令发送工件位置信息。
- 触发时间：“触发类型”为“定时触发”时，该设定生效。表示设置指定触发输出端口，输出高电平信号的间隔时间。
- 触发输入端口：用于指定系统检测的触发输入端口信息。“触发类型”为“外部 IO 触发”或“非相机触发”，该设定生效。
- 触发输出端口：用于指定系统输出的触发输出端口信息，使得相机进行拍照动作。“触发类型”为“定时触发”或“外部 IO 触发”时，该设定生效。
- 触发信号校验类型：
  - 无：该方式不会对触发信号进行任何校验处理，即只要信号触发就认为是有效信号。
  - 时间：时间校验方式，即在上一个有效信号触发后的一段时间内，系统将屏蔽信号接收功能。屏蔽对象不同，屏蔽接收的数据也不同。
  - 距离：距离校验方式，即在上一个有效信号触发后的一段距离内（这个距离指的是传送带运动距离），系统将屏蔽信号接收功能。屏蔽对象不同，屏蔽接收的数据也不同。
- 屏蔽对象：
  - 位置接收：当触发信号校验类型为时间或者距离时，系统对上一次有效信号触发后的一段时间或距离内从视觉或者 SetTargetPos 指令发送过来的工件信息进行丢弃，即该段区间内接收到的任何工件信息均为无效数据。
  - 信号检测：当触发信号校验类型为时间或者距离时，系统对上一次有效信号触发后的一段时间或距离内不再检测触发信号，即在该段区间内接收到的工件信息均以上一次有效信号记录的编码器位置来执行跟随运动。
- 校验时间：用于指定上一次有效信号后屏蔽信号接收的时间，单位 ms。“触发信号校验类型”设为“时间”时，该参数有效。
- 校验距离：用于指定上一次有效信号后屏蔽信号接收的距离，直线传送带时其单位 mm，圆形传送带时其单位为 deg。“触发信号校验类型”设为“距离”时，该参数有效。

## 视觉参数配置

选择“视觉参数”，进入视觉参数配置界面。

图7-18 视觉参数配置界面



- 视觉 IP：设定为相机的 IP 地址。
- 端口号：默认端口号为 5500。
- 连接状态：显示与相机服务器的连接状态。绿色表示已连接。灰色表示已断开。无色表示未连接。



- 清空：点击该按钮，可清空该按钮上方显示的视觉端发来的点位信息。

虚拟传送带配置

选择【虚拟传送带】，进入虚拟传送带参数配置界面。

图7-19 虚拟传送带参数配置界面



- 虚拟传送带使能：控制虚拟传送带开启/关闭。
- 虚拟传送带使能状态：显示当前虚拟传送带的开启/关闭状态。绿色表示已开启。灰色表示关闭。请注意，虚拟传送带开启/关闭指令的执行也会影响该状态的显示。
- 虚拟传送带速度：设置/显示虚拟传送带的速度。

若需进一步了解传送带脉冲当量标定步骤以及跟随现场配置步骤，请详见 《ER 系列工业机器人传送带跟随功能操作手册 RCS2》。

----结束

## 7.6 系统 IP 设置

### 网络设置

系统 IP 设置仅对“管理人员”用户类型开放。

选择系统主页中的“高级设置”，点击“网络设置”进入系统“网络设置”界面，示教编程器会根据控制器系统版本显示匹配的设置界面。

为了保证正常的通讯，该界面中的参数请按照实际情况，设置为正确的数值。

#### 网络设置 A

控制器系统中网卡采用内部桥接方式，当任一网口接入局域网，示教编程器 IP、OS1、OS2 的 IP 地址都不能与局域网其它设备 IP 相同，否则，网络地址冲突导致通讯异常。

图7-20 网络设置界面 A



- 远程网络设置：可远程修改正在与示教编程器端连接的 OS 端 IP 地址。
  - OS1 IP：OS1 的 IP 地址。
  - OS2 IP-TP：OS2 的 IP 地址。
  - OS2 IP-Comm：OS2 通信用的 IP 地址。
- 本地网络设置：只修改示教编程器端本地保存连接的 OS 端 IP 地址。
  - 示教编程器 IP：示教编程器的 IP 地址。
  - OS1 IP：OS1 的 IP 地址。
  - OS2 IP-TP：OS2 的 IP 地址。

#### 网络设置 B

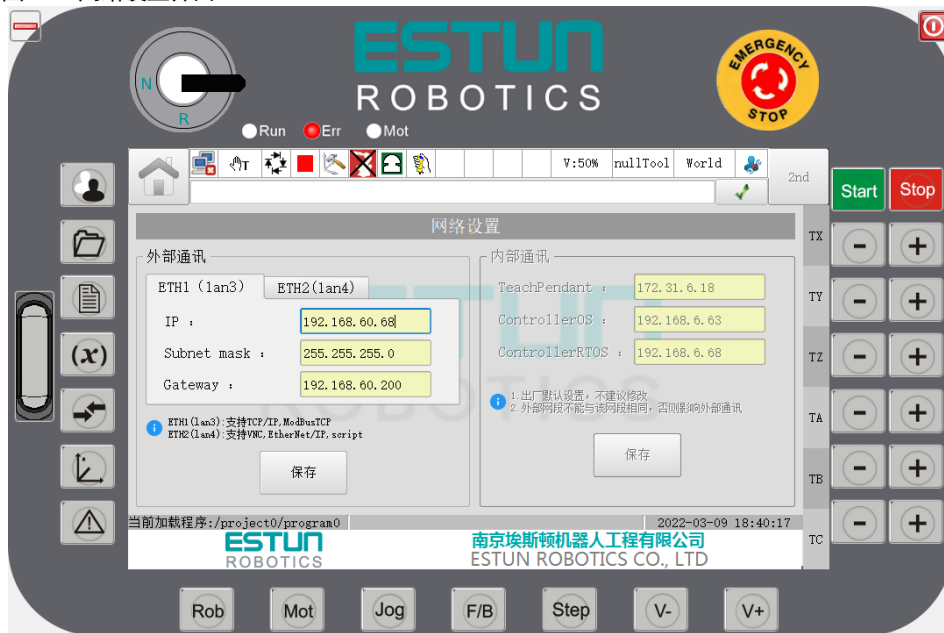
该控制器系统将网络分离为内部网络和外部网络。其中，内部网络主要是示教编程器、控制器有效连接；外部网络是与外部通讯的有效连接，所以当机器人组线时，仅需依据应用修改 ETH1 或 ETH2 的网络地址。



### 注意

- 外部通讯网络地址不可与内部通讯网络地址的标识段相同，否则会影响正常通讯。（比如，内网出厂默认网段为 192.168.6.x，所以外部网络不允许设置为 192.168.6.x）

图7-21 网络设置界面 B

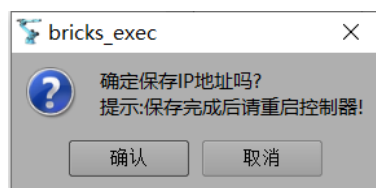


- 外部通讯网络设置：
  - ETH1(lan3): 控制器端 lan3 口，支持 TCP/IP、ModbusTCP 等。
  - ETH2(lan4): 控制器端 lan4 口，支持 VNC、EtherNet/IP、Script 等。
- 内部通讯网络设置：（默认不支持修改，如需修改联系厂家）
  - TeachPendant: 示教编程器的 IP 地址。
  - ControllerOS: 控制器的 IP 地址。
  - ContollerRTOS: 控制器实时系统的 IP 地址。
- 修改子网掩码和网关规则：

示教器启动时自动读取子网掩码及网关，显示在界面中。

子网掩码不能为空，网关输入可以为空。

修改完子网掩码及网关后，点击“保存”按钮后成功，按界面提示重启控制器（如下图所示），重启完成后，修改生效。

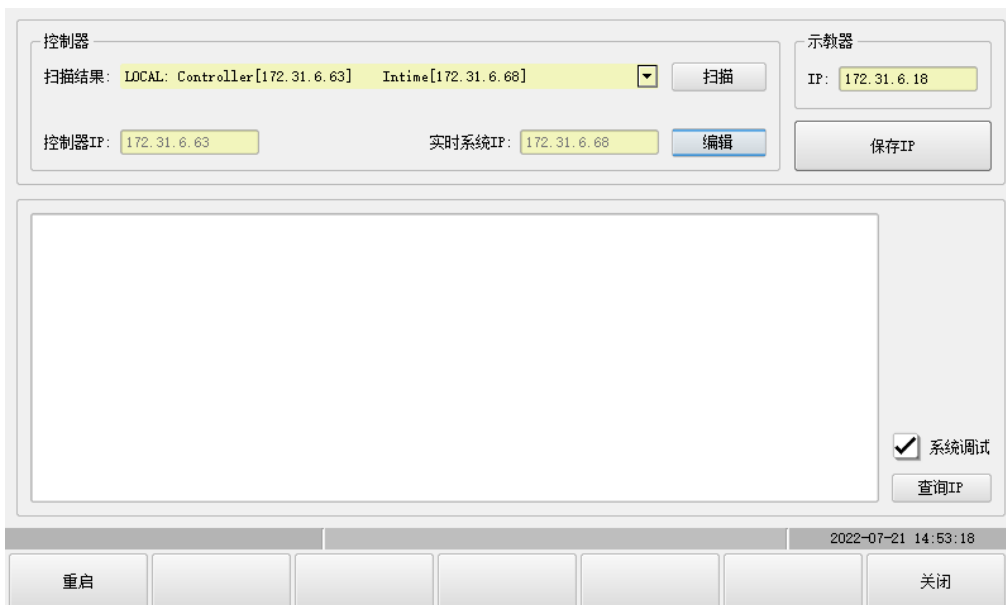


## IP 扫描

- 当控制器系统为网络设置中 B 类型时，该功能暂不可使用。
- 在一些客户现场，可能在同一个局域网中会有多台控制器，在不清除控制器 IP 配置的情况下，会导致示教编程器无法连接到正确的控制器上，所以本功能实现了可通过示教编程器来扫描局域网内所有控制器的 IP，客户可根据扫描到的自主选择一台连接。
- 在示教编程器启动后，如果出现无法连接到控制器的情况，可按照下图顺序按快捷键进入 IP 配置界面。



- 在 IP 配置界面可直接“编辑”IP（需输入密码二次验证），也可点击“扫描”，来自动搜索当前局域网内控制器 IP 以及对应的实时系统 IP，点击下拉列表选中将自动填充到配置栏中。
- 点击“保存 IP”，保存成功后，然后点击“重启”，待重启后示教编程器将连接到指定的控制器。



## 7.7 示教编程器升级与备份

按照如下步骤，可实现对示教编程器程序进行升级和备份。

### 7.7.1 示教编程器程序升级

示教编程器的升级有两种方式：

#### 1、通过 ESTool 操作

直接将示教编程器程序更新到示教编程器系统中，不需要勾选“开机同步更新示教编程器”，当上电启动时直接加载示教编程器系统中的程序运行。

#### 2、通过 TP 更新插件操作

将示教编程器程序更新到控制器系统中，需要勾选“开机同步更新示教编程器”，当上电启动时同步加载控制器系统中的程序运行。选择这种方式时，控制器系统版本升级，示教编程器也会自动同步更新。

### 更新方式设置

步骤 1 切换用户为管理人员，点击高级设置插件组，如图 7-22 所示。

图7-22 高级设置插件组



步骤 2 进入 TP 更新插件，如图 7-23。

图7-23 TP 更新插件



步骤 3 进入如图 7-24TP 更新界面，勾选“开机同步更新示教编程器”，启用示教编程器同步更新方式。

图7-24 HMI 更新界面



步骤 4 点击“选择文件”按钮找到需要更新的文件。

步骤 5 点击“重启”按钮，输入密码，密码校验成功后，重启示教器程序，重启完成后生效。

## ESTool 更新

步骤 1 启动 ESTool，启动方法请参见“启动 ESTool”。

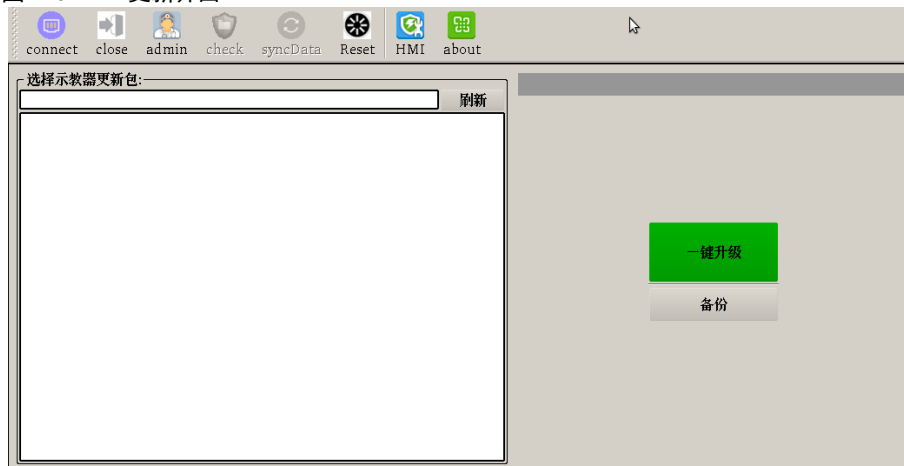
步骤 2 ESTool 登录界面如图 7-25 所示。

图7-25 ESTool 登录界面说明



步骤 3 登录后如图 7-26HMI 更新界面，选择 HMI 功能模块，示教编程器插入 U 盘，在左侧选择要更新的文件，点击一键更新按钮即可实现示教编程器的更新。

图7-26 HMI 更新界面



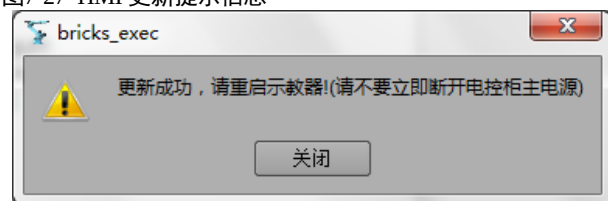
## HMI 更新插件

步骤 1 切换用户为管理人员，进入高级设置插件组中的 TP 更新插件，如图 7-24。

步骤 2 在示教编程器程序列表中选择要 U 盘中需要更新的示教编程器程序文件，点击“选择”。

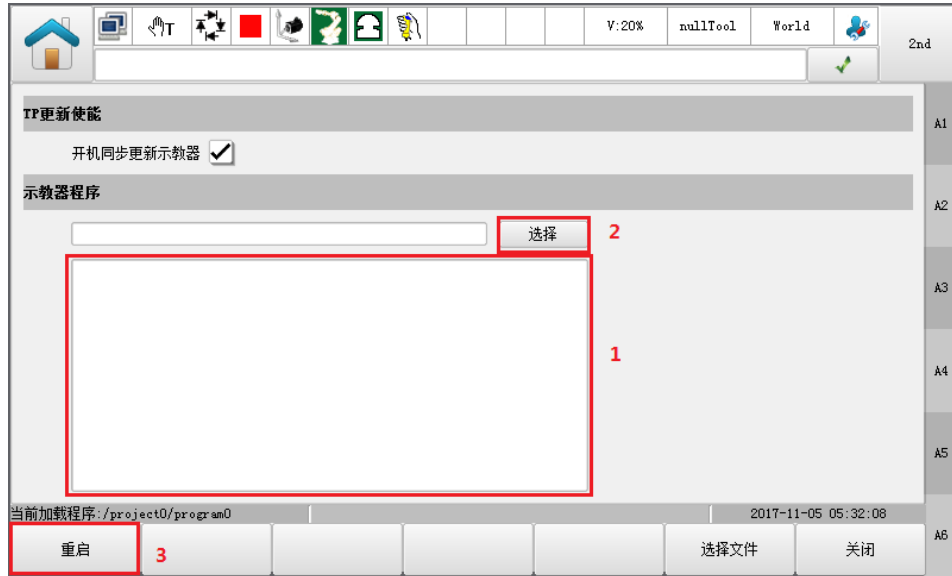
步骤 3 更新完成后，弹出提示信息，如图 7-27。

图7-27 HMI 更新提示信息



步骤 4 点击“重启”，示教编程器重启，并自动升级到更新的 TP 版本。

图7-28 TP 更新步骤



## 7.7.2 示教编程器程序备份

点击图 7-26 中备份按钮可实现将当前的示教编程器程序导出至 U 盘内。

----结束



## 7.8 控制器运行程序更新

### 7.8.1 控制器程序更新

更新控制器运行程序，按照以下步骤操作。

- 步骤 1 将准备更新的控制器运行程序（.runtime 或 .patch 格式文件，该文件通常由研发人员提供）放入 U 盘根目录。
- 步骤 2 将 U 盘插入示教编程器 USB 接口，选择进入系统主页中高级设置，点击进入控制器升级界面如图 7-29 所示。
- 步骤 3 根据需要勾选“后台自动备份”功能，若勾选则在更新前将在控制器端备份现有的系统软件，在进行系统还原时将使用该备份的系统软件进行还原。
- 步骤 4 点击“更新”功能按键，输入管理人员密码进行确认后将提示是否需要导出现有运行时，选择“是”将控制器现有的运行时导出来，选择“否”则跳过该步骤，继续下一步操作。
- 步骤 5 选择更新文件后将进入自动更新状态，然后等待更新完成即可。

#### 说明

当进行控制器程序更新时，当勾选了“后台自动备份”，那么将在更新过程中看到有备份运行时的过程，如图 7-29 图 7-30 所示；

若使用实际示教编程器通过 U 盘更新，那么 U 盘格式必须是 FAT32 格式；

不论是 U 盘更新还是虚拟示教编程器直接选择更新，更新包名称以及更新包所在路径不能包含中文字符；  
在系统更新期间，请勿断开系统电源，以防系统更新尚未完成而导致系统无法正常运行。

图7-29 控制器运行程序更新界面



图7-30 更新状态显示



## 7.8.2 控制器程序还原

控制器程序还原按照以下步骤操作。

- 步骤 1 切换至管理人员用户，进入控制器升级界面，点击“还原”输入管理人员密码进行确认后，弹出确认提示框，如图 7-31 所示，点击“确认”开始向控制器请求还原。
- 步骤 2 待控制器通过还原请求后，开始还原操作，如图 7-32 所示，还原开始后示教编程器端将与控制器断开连接。
- 步骤 3 耐心等待一段时间后，示教编程器将提示还原即将结束的提示信息框，如图 7-33 所示，如提示信息所示，为确保数据同步需要重启示教编程器，对于虚拟示教编程器，无法做到自动重启，所以在点击确认后，虚拟示教编程器将自动关闭，需要人为的重新打开运行，而对于实际示教编程器，点击确认后，实际示教编程器将会自动重启。

### 说明

在等待系统还原时，请勿断开系统电源，以防可能由于系统还原失败导致系统无法正常运行。

图7-31 控制器程序还原确认界面



图7-32 控制器程序还原界面



图7-33 控制器程序还原结束界面



### 7.8.3 备份导出

- 步骤 1 切换至管理人员用户，进入“控制器升级”界面点击“备份导出”，显示备份导出界面，如图 7-34 所示。
- 步骤 2 根据需要选择导出类型（包括：运行时包、工程程序以及系统日志），然后点击“导出”，输入管理人员密码进行确认通过后（若使用实际示教编程器没有插入 U 盘，则会提示插入 U 盘），将向控制器请求导出，如图 7-35 所示。
- 步骤 3 请求获取通过后，将显示输入导出重命名对话框，虚拟示教编程器另外还支持选择保存的目录，设置好导出的名称后确定，将开始导出对应的文件。
- 步骤 4 待导出完成后，如图 7-36 所示。

#### 说明

若使用实际示教编程器，在导出完成前请勿拔出 U 盘，否则导出的文件可能不存在、或被损坏；

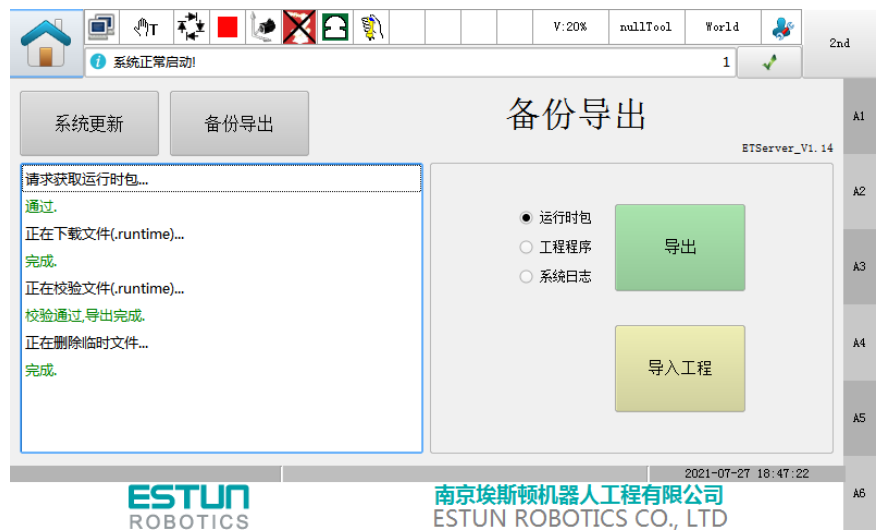
图7-34 备份导出界面



图7-35 请求导出界面



图7-36 导出完成界面



## 7.8.4 工程导入

- 步骤 1 将准备好的工程文件（.project 格式的文件，该文件可由备份导出功能导出工程程序获得，如特殊情况需要制作，请参考“ER 系列工业机器人控制器升级操作手册”）放入 U 盘根目录下，将 U 盘插入示教编程器。
- 步骤 2 切换至管理人员用户，进入“控制器升级”界面点击“备份导出”，点击“导入工程”，输入管理人员密码确认后，显示如图 7-37 所示界面，提示导入工程将覆盖现有工程，是否需要导出现有工程，点击“是”则请求获取现有工程程序，该流程同备份导出工程程序，导出完成后将进入工程程序文件对话框，若点击“否”，则直接进入选择工程程序文件对话框。
- 步骤 3 选择导入的工程程序后，将开始导入，待导入完成后，将提示导入成功，请刷新工程，如图 7-38 所示。
- 步骤 4 点击右侧工程按键，进入工程界面，点击“刷新”即可显示导入的工程。

图7-37 工程导入界面



图7-38 工程导入完成界面



----结束

7.9 一点到达功能介绍

一点到达功能，即机器人从当前点以拱形路径快速运动到目标点。使用界面如图 7-39 所示。

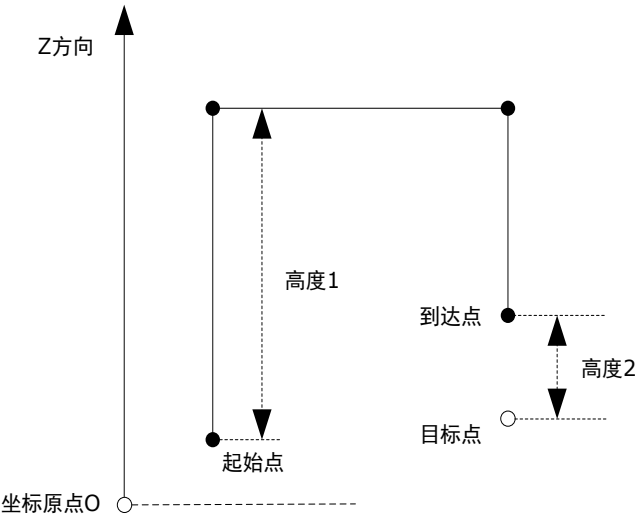
图7-39 一点到达使用界面



说明

- 该功能目前仅可在 SCARA 机型使用。
- 一点到达，目前只适用于世界坐标系下的位置变量。
- 激活表示使一点到达参数配置生效。
- 使用该功能时需在手动模式下按住手压再进行运动操作。
- 长按到达该点按钮，机器人会一直运动到目标点后停止。若中途松开该按钮，机器人则立即停止运动；当再次按下该按钮时，机器人会继续往目标点运动。其中：高度 1 表示从起始点沿 z 轴向上垂直抬升的高度；高度 2 表示最终到达点距离目标点的垂直高度。具体拱形路径如图 7-40 所示。

图7-40 拱形路径

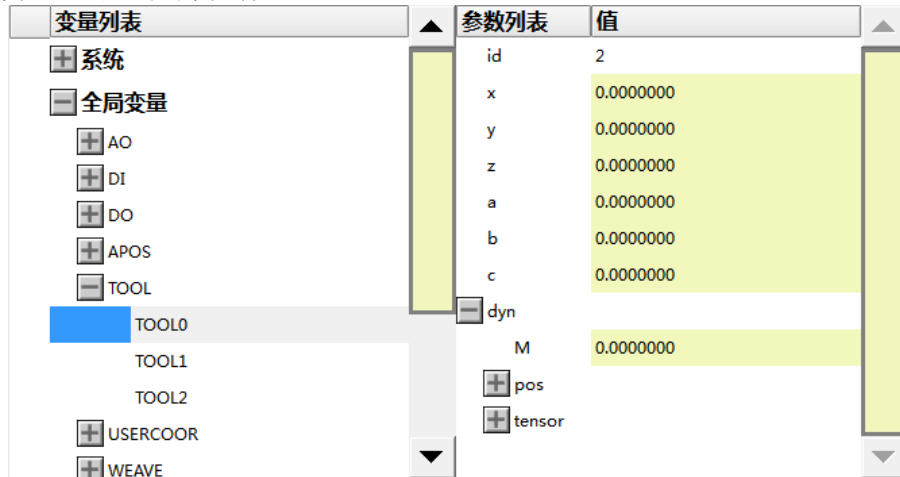


## 7.10 工具坐标系标定

按照如下操作步骤进行一个工具坐标系的标定。

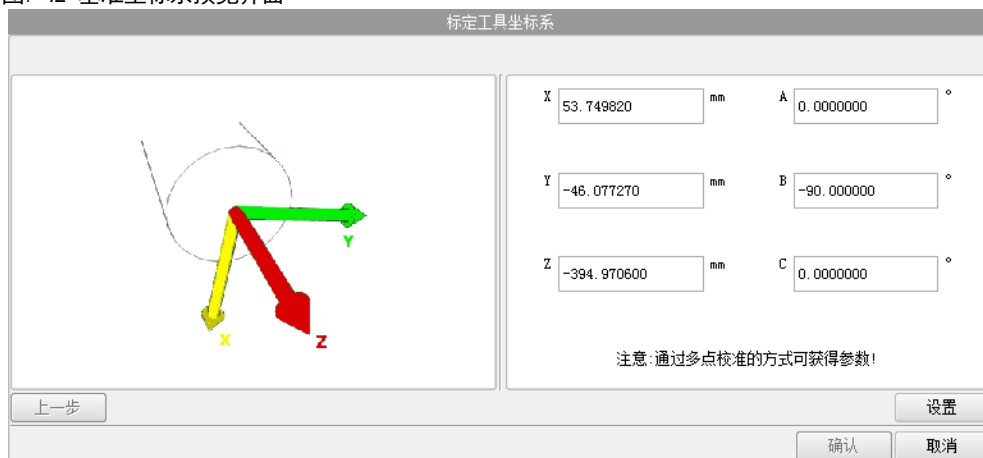
步骤 1 进入程序数据界面，点击全局域中要标定的工具坐标系，如 TOOL0，如图 7-41 所示。

图7-41 工具坐标系变量界面



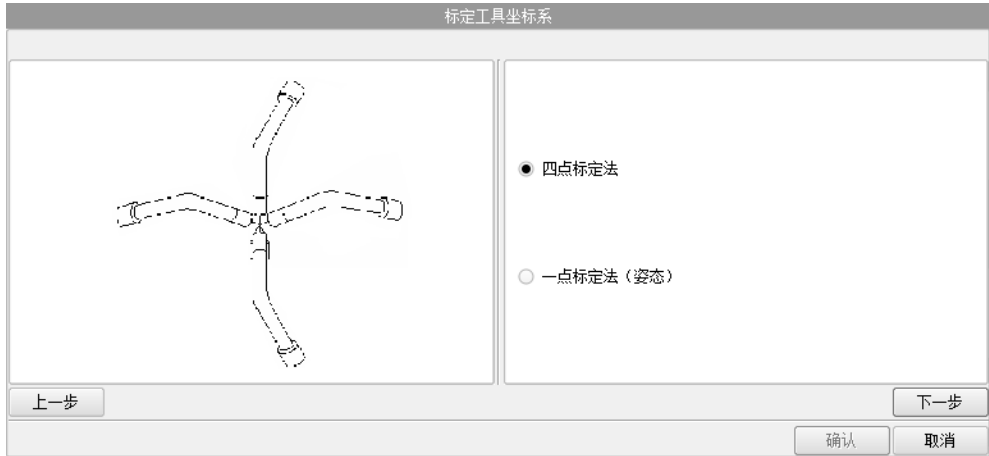
步骤 2 点击“修改”，将进入基准坐标系预览界面，如图 7-42 所示。

图7-42 基准坐标系预览界面



步骤 3 点击“设置”，进入基准坐标系的标定方法选择界面，如图 7-43 所示。

图7-43 选择标定的方法

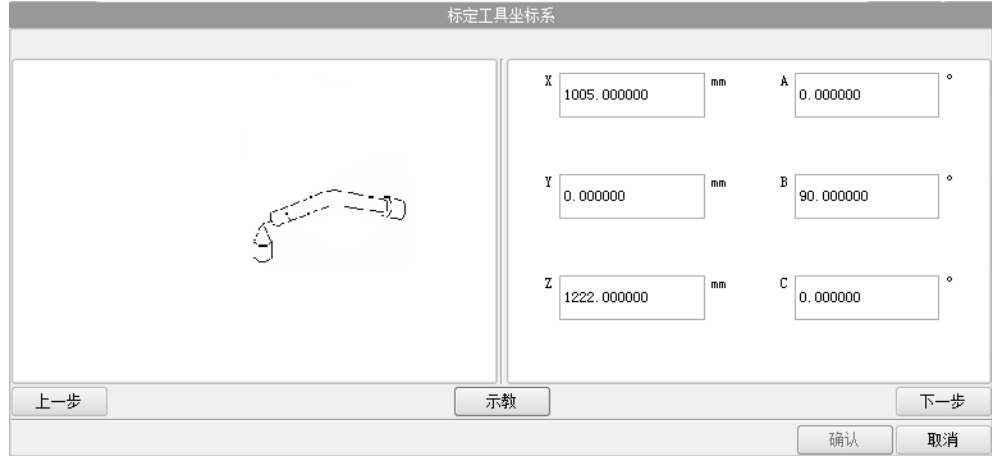


步骤 4 选择标定方法，对偏移或姿态进行标定。

【四点标定法】

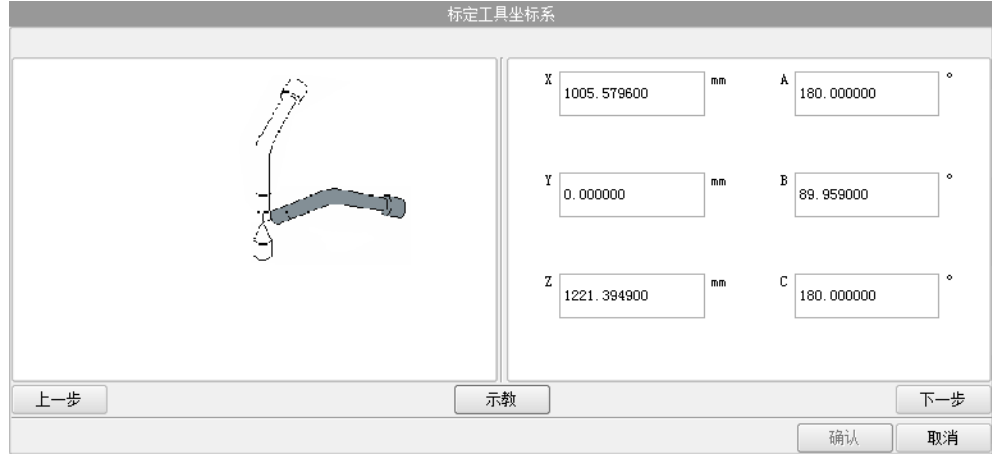
- 1. 选择“四点标定法”，表示接下来需要示教同一个点的 4 种不同姿态的方式来标定工具偏移。点击“下一步”。移动机器人 TCP 至某个方位，点击“示教”，如图 7-44 所示。

图7-44 示教第一个点



- 2. 再点击“下一步”，更改机器人的姿态，点击“示教”，如图 7-45 所示。

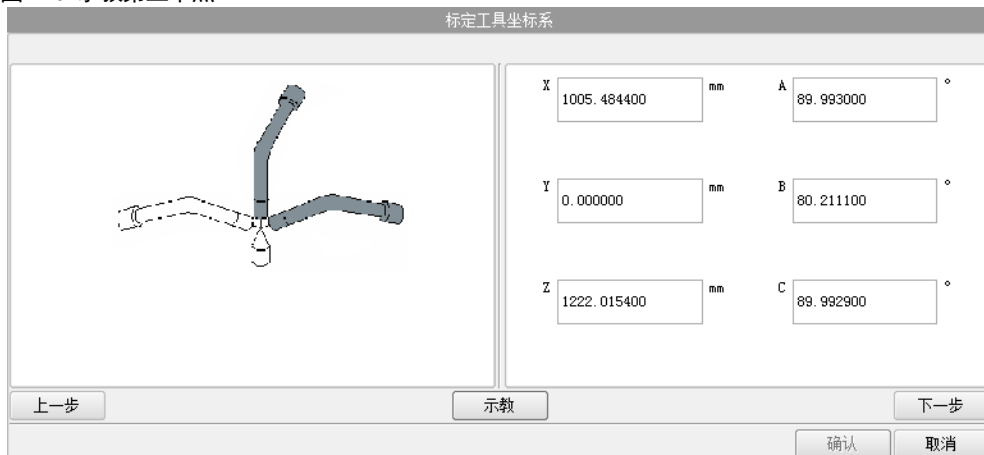
图7-45 示教第二个点





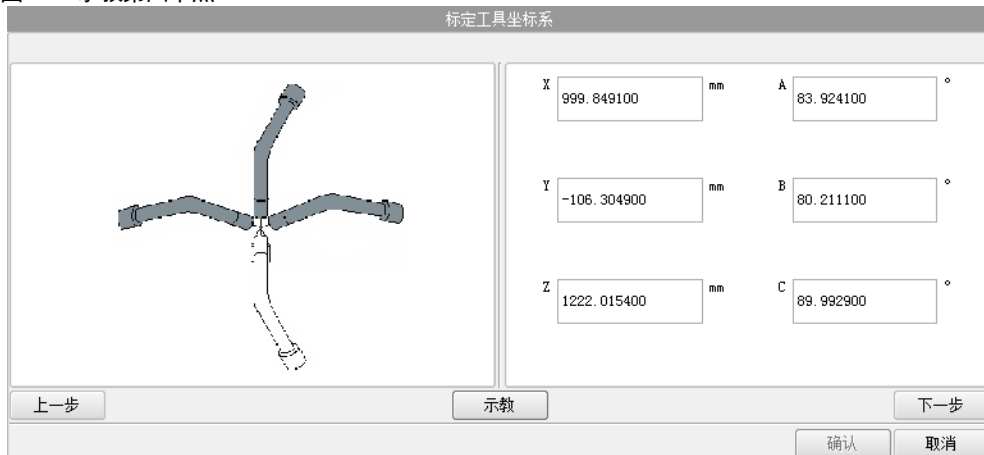
3. 再点击“下一步”，更改机器人的姿态，点击“示教”，如图 7-46 所示。

图7-46 示教第三个点



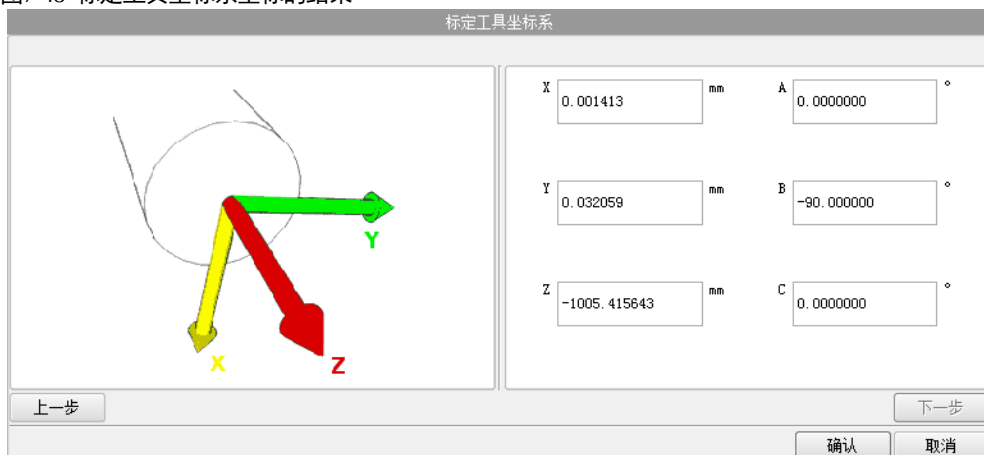
4. 再点击“下一步”，更改机器人的姿态，点击“示教”，如图 7-47 所示。

图7-47 示教第四个点



5. 页面将提示本次示教后的精度误差，用户可根据实际需求确定继续保存结果或重新标定。
6. 再点击“下一步”，得到标定的值，如图 7-48 所示。

图7-48 标定工具坐标系坐标的结果



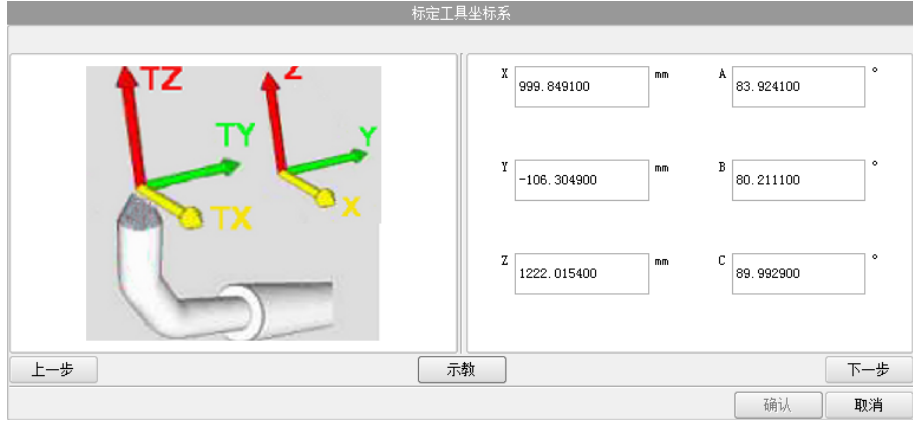
7. 确认无误后，点击“确认”，完成修改。

【一点标定法】

选择“一点标定法”，表示接下来标定工具的姿态。该方法仅能标定工具在工具坐标系下某一个轴方向上姿态变化。

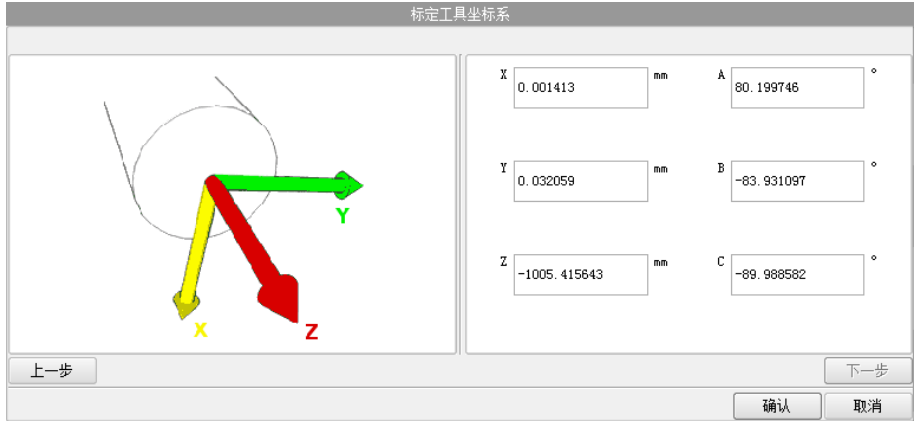
1. 点击“下一步”。将机器人末端方向与 Z 轴方向平行，点击“示教”，如图 7-49 所示。

图7-49 示教 Z 轴上一点



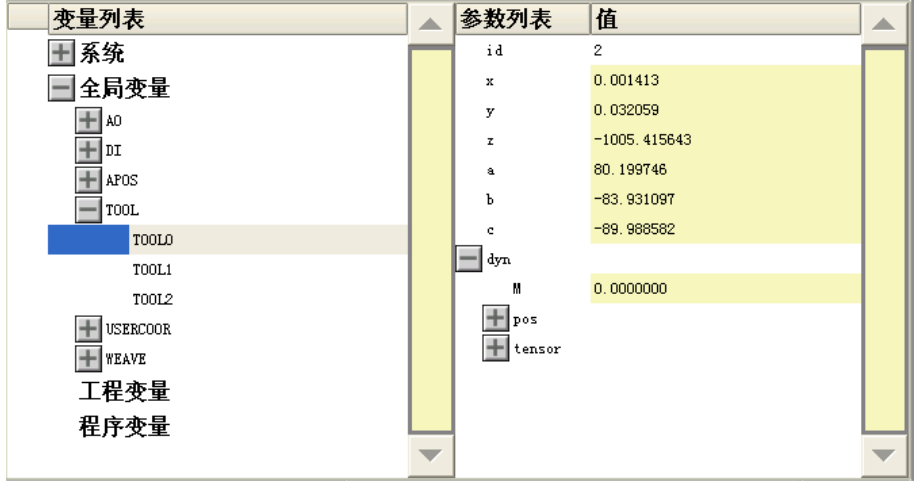
2. 点击“下一步”。显示标定工具坐标系姿态的结果，如图 7-50 所示。

图7-50 标定工具坐标系姿态的结果



步骤 5 确认无误后，点击“确认”，完成修改。如图 7-51 所示。

图7-51 成功标定姿态的工具坐标系变量



----结束

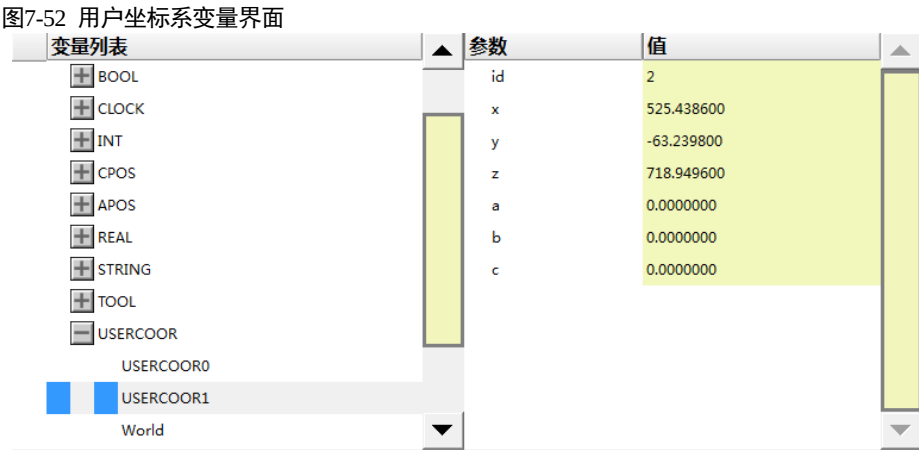
## 7.11 用户坐标系/外部工具坐标系标定

 说明

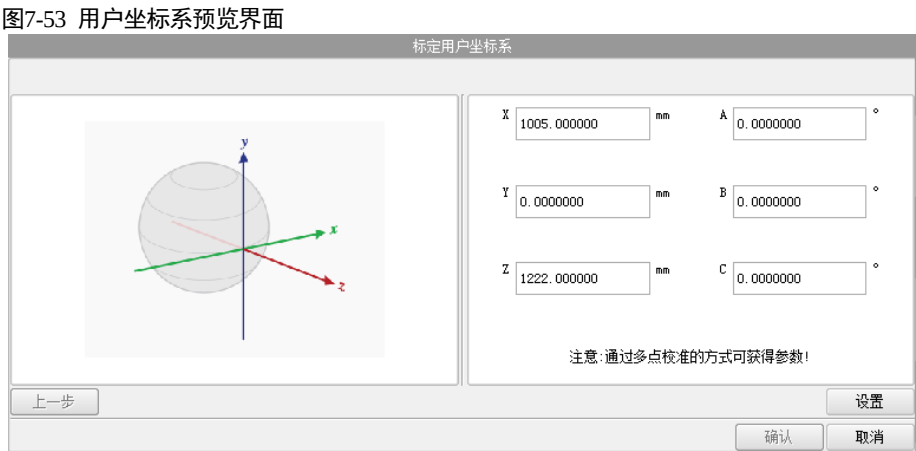
本节以用户坐标系的标定操作为示例。  
外部工具坐标系和用户坐标系标定方法相同，只是修改时，注意变量类型：用户坐标系默认名称为 USERCOOR，而外部工具坐标系默认名称为 EXTTCF。

按照如下操作步骤进行一个用户坐标系的标定。

步骤 1 进入程序数据界面，点击全局域中要标定的用户坐标系，如 USERCOOR1，如图 7-52 所示。

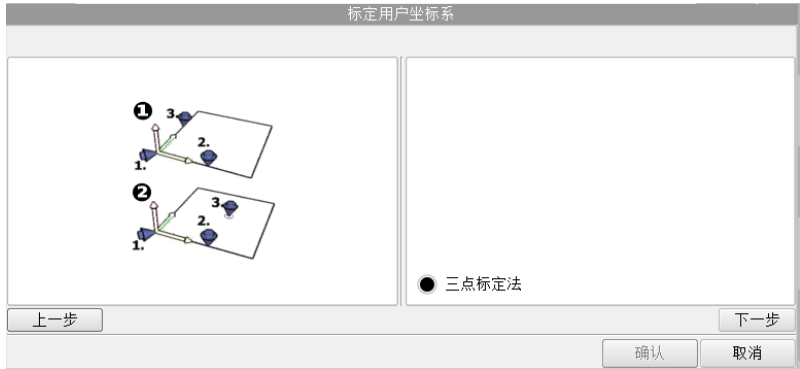


步骤 2 点击“修改”，将进入用户坐标系预览界面，如图 7-53 所示。



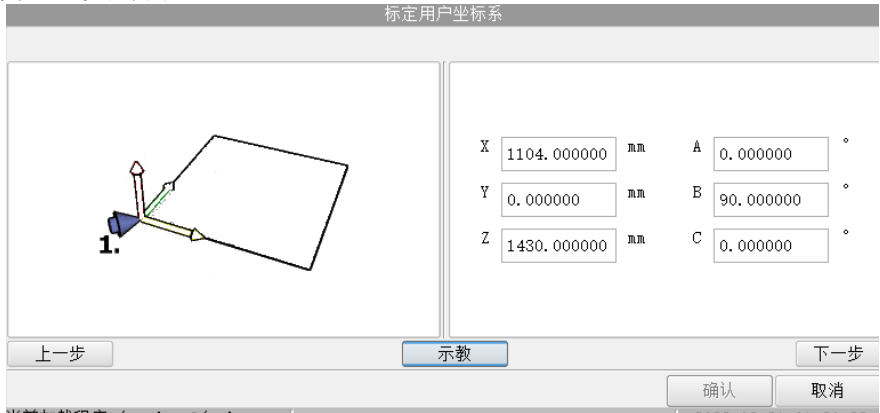
步骤 3 点击“设置”，进入用户坐标系的标定方法选择界面。  
目前仅有“三点标定法”，如图 7-54 所示。

图7-54 选择标定的方法



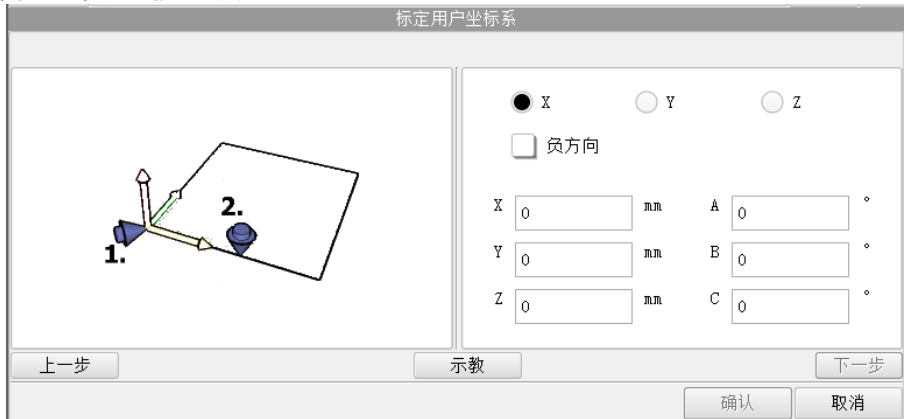
步骤 4 点击“下一步”，开始示教原点的坐标，移动机器人至用户坐标系的原点。点击“示教”，界面将记录原点（第一个点）的坐标，如图 7-55 所示。

图7-55 示教原点



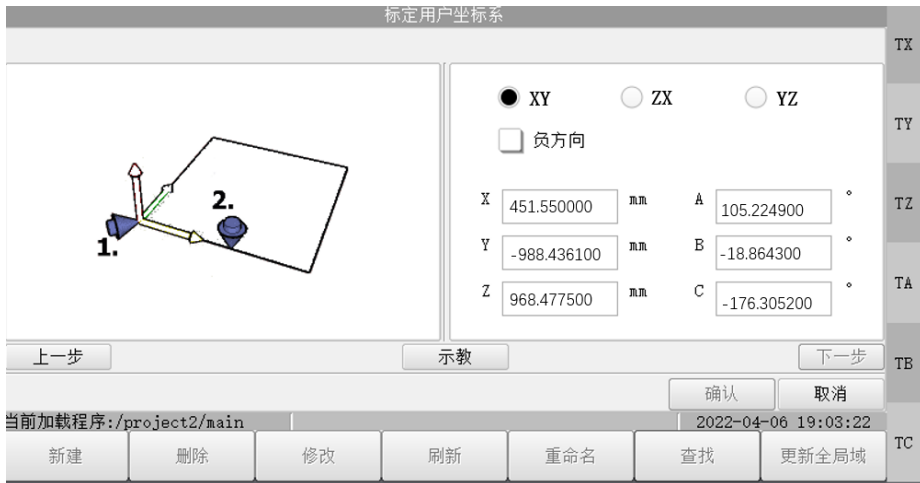
步骤 5 点击“下一步”，移动机器人至用户坐标系的 X 轴上的任意位置处。点击“示教”，界面将记录第二个点的坐标，如图 7-56 所示。

图7-56 示教 X 轴上一点



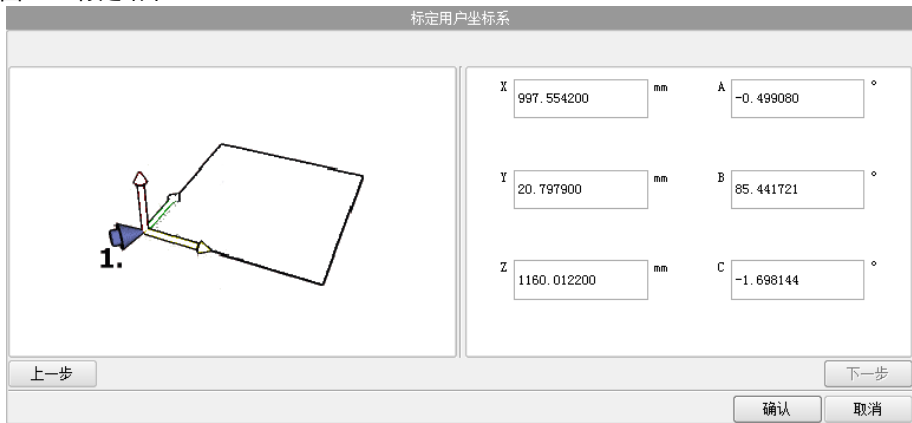
- 若勾选了 X：表示选择了 X 轴，只能选择 xy 平面或 xz 平面。选择 Y/Z，同理。
  - 勾选负方向：表示沿负方向运动。
- 步骤 6 点击“下一步”，移动机器人在用户坐标系 XY 平面上的任意位置处。点击“示教”，界面将记录第三个点的坐标，如图 7-57 所示。

图7-57 示教 XY 平面的任一点。



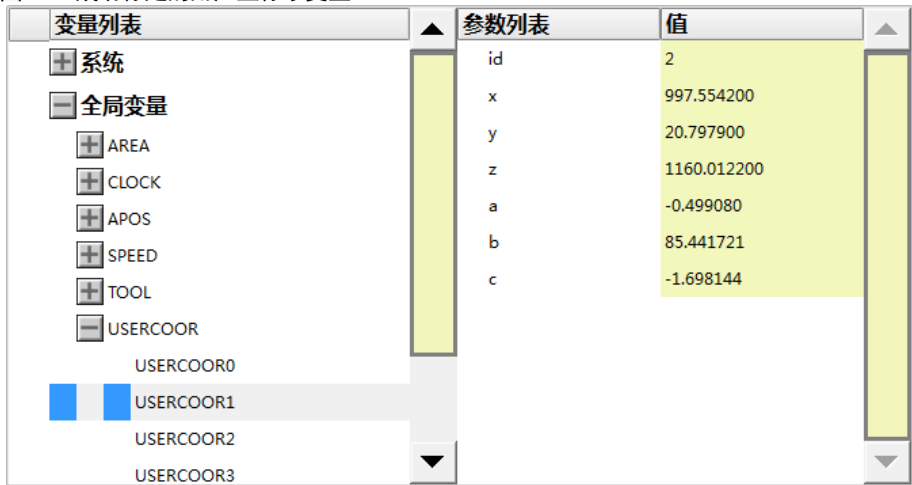
步骤 7 点击“下一步”，回到用户坐标系预览界面。如图 7-58 所示。

图7-58 标定结果



步骤 8 确认无误后，点击“确认”，完成修改。如图 7-59 所示。

图7-59 成功标定的用户坐标系变量

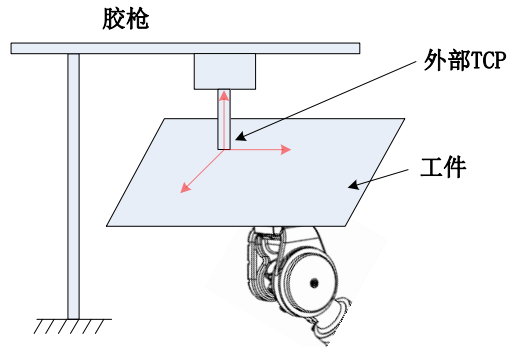


----结束

## 7.12 外部 TCP 功能

### 功能概述

通常，工具安装在机器人上，并跟随机器人的运动来完成指定的工作。在某些特定的应用中，如涂胶应用，需要将胶枪（工具）安装在机器人外部的某个位置，而工件需要固定在机器人上进行涂胶。这种安装在机器人外部，且安装在某个固定位置上的工具叫做外部 TCP。



使用外部 TCP 功能之前需要对外部 TCP 进行标定，详细请参见“7.11 用户坐标系/外部工具坐标系标定”。

点动参考坐标系中可以选择外部工具坐标系。当参考坐标系选择外部工具坐标系时，点动位置和姿态均相对于外部工具坐标系。外部 TCP 下点动位置与用户坐标系下点动位置相似，均沿着参考坐标系的 X、Y、Z 方向运动。外部工具下点动姿态与用户坐标系下点动姿态区别：用户坐标系下点动姿态，TCP 点位置不动，姿态绕着 TCP 点旋转。外部工具下点动姿态，TCP 点绕着外部工具坐标系在旋转。

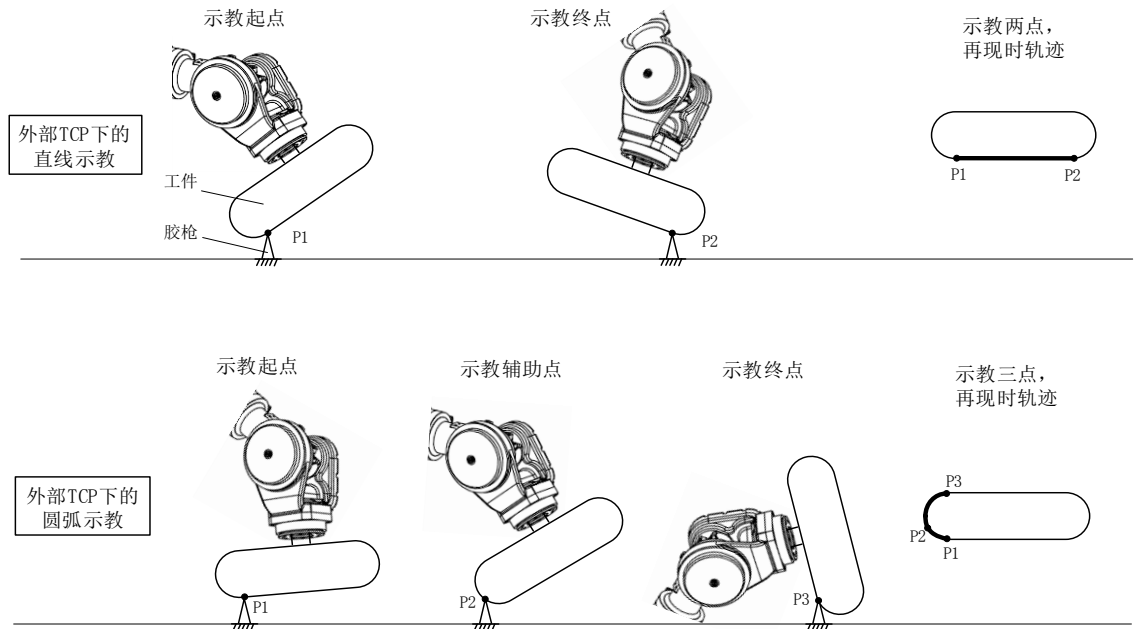
在 MovL、MovC 指令中可以选择参考坐标系为外部工具坐标系，也可通过设置指令 SetExternalTCP 设置当前的参考坐标系为外部工具坐标系。当使用外部 TCP 示教直线圆弧运动时，直线示教 2 点、圆弧示教 3 点。

### 说明

示教存储的位姿点为外部 TCP 相对于 TCP 点的描述，机器人的运动相对于外部 TCP 坐标系。

例如使用外部 TCP 功能时，无论示教起终点的姿态如何变化，若示教起终点工件贴着外部 TCP 原点，工件轮廓为直线，使用直线运动指令时，工件轮廓始终沿着外部 TCP 原点运动，运动速度为工件相对于外部 TCP 坐标系的速度。

### 使用方法



## 7.13 一点对正坐标轴

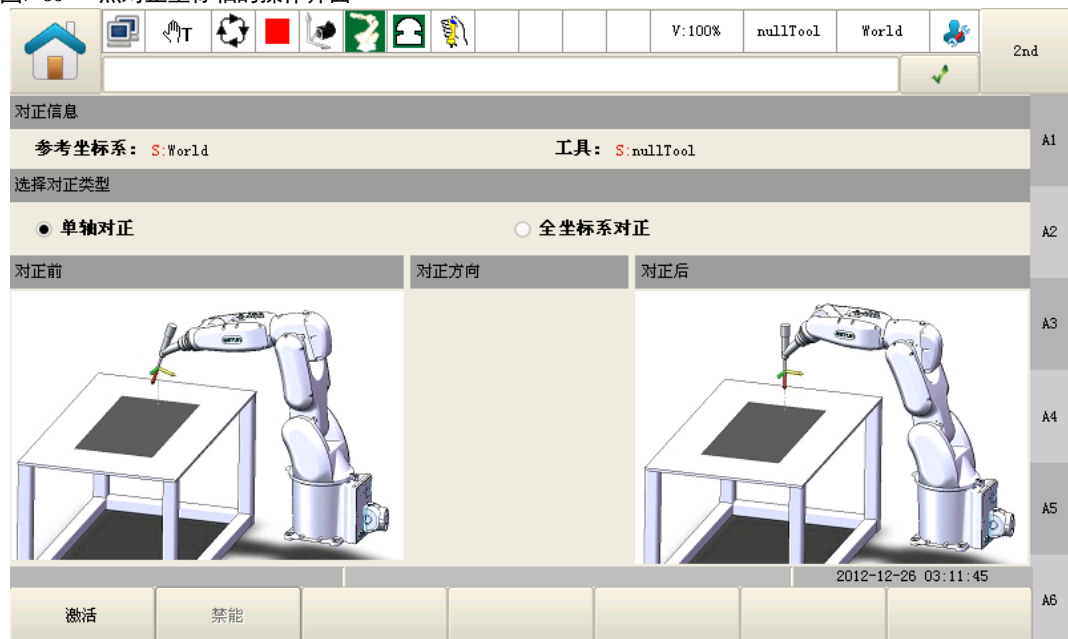
### 功能概述

机器人在某些应用场合，需要旋转姿态，使得工具坐标系的某一轴或全轴与指定平面精确垂直或平行。一点对正坐标轴即可满足此需求，用户可以粗略地示教出机器人要达到的目标姿态，然后应用一点对正坐标轴功能进行精确地姿态调整以达到最终的目标姿态。

### 使用方法

步骤 1 选择进入系统主页中点动管理分组，点击“工具对正”，进入“一点对正坐标轴”的操作界面，如图 7-60 所示。

图7-60 一点对正坐标轴的操作界面



步骤 2 选择“对正类型”。

- 单轴对正：仅对工具坐标系的  $T_z$  方向与指定坐标系的最近坐标轴进行对正。
- 全坐标系对正：对工具坐标系的各个轴 ( $T_x$ 、 $T_y$ 、 $T_z$ ) 与指定的坐标系一一对正。

#### 说明

“对正”是指不改变当前 TCP 的位置，仅通过改变姿态使得 TCP 与指定坐标系上的轴平行。

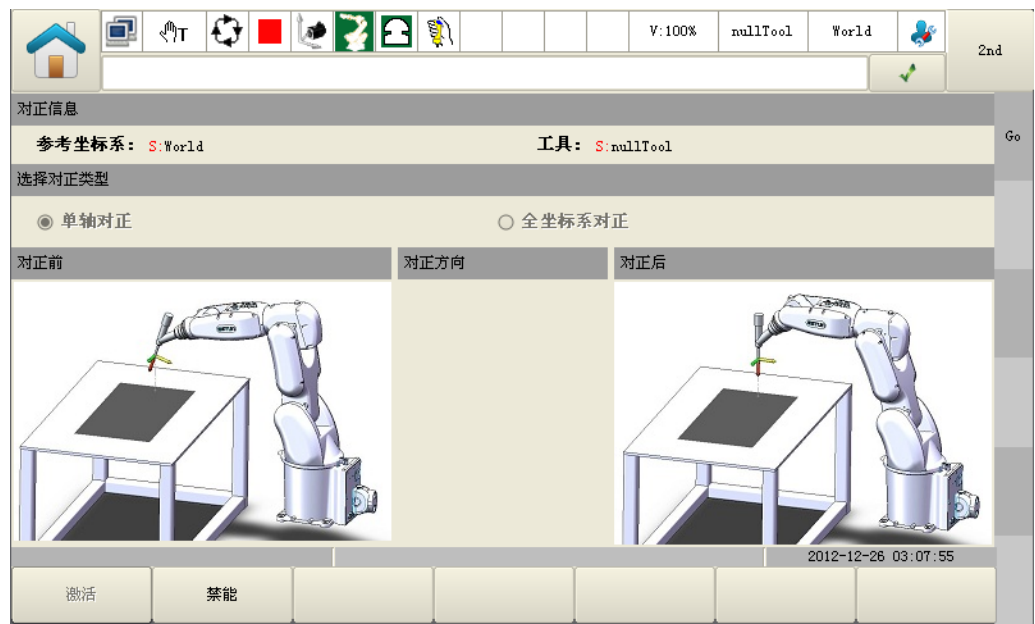
步骤 3 点击“激活”。

若要取消对正操作，可点击“禁能”。

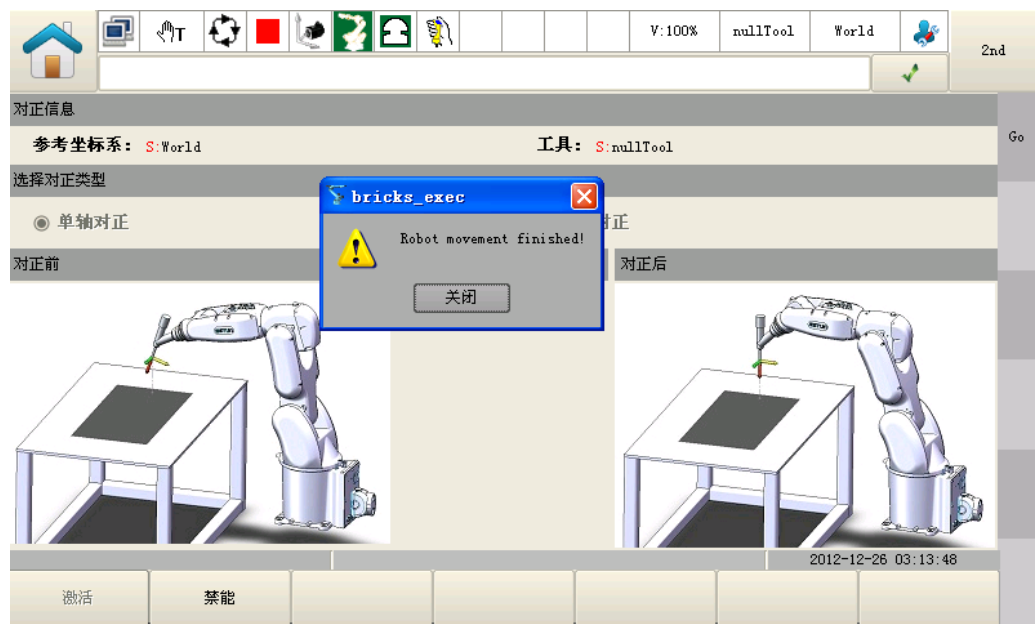
步骤 4 点击右上角“GO”对应的“+/-”按钮进行对正。



图7-61 一点对正坐标轴激活后操作界面



步骤 5 当有弹窗提示代表对正到位，如下图。



----结束

## 7.14 伺服管理

### 功能概述

用户可通过示教编程器的界面直接进行伺服驱动器的报警清除和复位、部分 PN 参数设置、查看伺服版本信息以及批量备份和恢复等操作。



### 注意

- 进行“伺服管理”操作前，请确认如下条件是否满足：
- 已确认轴驱动器标号（1 轴～6 轴）对应了机器人的各关节。

### 使用方法

选择进入系统主页中“通用设置”分组，点击“伺服管理”图标，进入“伺服管理”的操作界面，如图 7-62 所示。

### 清除伺服报警

首次进入伺服管理页面默认显示“清除伺服报警”的操作界面，如图 7-62 所示，也可以在切换操作界面时可以点击底部功能按键，点击“清伺服报警”按键进入操作界面。

图7-62 清除伺服报警



- 点击“复位 Fn010”和“复位 Fn011”可清除与绝对值编码器有关的伺服报警（A.45、A.46、A.47、A.48、A.51）。
- 点击“复位”可清除对应轴的当前伺服报警。
- 点击“清除历史报警”可清除对应轴的历史报警。



## 注意

- 点击“复位 Fn010”，将会清除编码器绝对值多圈信息，请谨慎操作。
- 确认已经排除了伺服故障后再进行“清除伺服报警”操作。
- 进行 Fn010 或 Fn011 操作时，保证伺服不在励磁状态，且进行操作后需要断电重启。
- 只有当电机的编码器为绝对值编码器时，才可进行 Fn010、Fn011 的应用操作。

## 设置 Pn 参数

点击底部“设置 Pn 参数”功能按键，进入“设置 Pn 参数”操作界面，如图 7-63 所示，初始默认显示 1 轴的 Pn 参数。

图7-63 设置 Pn 参数



- 点击底部“设置 Pn 参数”功能按键会获取界面上当前轴的 Pn 参数。
- 点击切换轴号的标签页时会获取对应轴号的 Pn 参数。
- 点击对应 Pn 参数的“设置”按键，弹出输入框，输入数值后点击“OK”按键即将输入的值下发给伺服，设置完成后在对应 Pn 参数后的文本框显示当前 Pn 数值。



## 注意

进行设置 Pn 参数操作之前：

- 断开伺服使能。
- 已掌握伺服参数的设置方法，包括参数的范围、含义等信息。

表7-1 可设置的 Pn 参数

| Pn 参数号 | 名称      | 范围      | 单位     | 说明                  |
|--------|---------|---------|--------|---------------------|
| Pn102  | 速度环增益   | 1~14000 | Hz     | 速度环增益的大小。           |
| Pn103  | 速度环积分时间 | 1~4096  | 0.25ms | 减小此值可以缩短定位时间，提高速度响应 |

| Pn 参数号 | 名称        | 范围      | 单位     | 说明                                      |
|--------|-----------|---------|--------|-----------------------------------------|
| Pn104  | 位置环增益     | 0~1000  | 1/s    | 位置环的增益大小，增大该值可以提高位置控制的伺服刚性，但过大可能引起振荡。   |
| Pn105  | 扭矩指令滤波器常数 | 0~2500  | 0.25ms | 设置扭矩滤波可以消除或减轻机械振动，但设置不合理时有时会引入机械振动。     |
| Pn106  | 负载惯量百分比   | 0~20000 | 1%     | 负载惯量对电机转子惯量之比率。<br>设定值=（负载惯量/转子惯量）× 100 |
| Pn112  | 前馈        | 0~300   | 1%     | 用来设置位置前馈数值，设得越高位置相应越快，位置偏差越小。           |
| Pn113  | 前馈滤波      | 0~640   | 0.1ms  | 平缓位置前馈引起的机械冲击，该值设定太大会使的前馈量滞后较多易引起振荡。    |
| Pn114  | 扭矩前馈      | 0~100   | 1%     | 用来设置扭矩前馈值，加快速度响应。                       |
| Pn115  | 扭矩前馈滤波    | 0~640   | 0.25ms | 减缓扭矩前馈引起的机械冲击。                          |

### 查看伺服软件版本

点击底部“版本信息”功能按键，进入“伺服软件版本”界面，如图 7-64 所示。

图7-64 伺服软件版本



- 示教编程器正常启动后，点击底部“版本信息”按键，会获取伺服软件版本并显示到界面上。



### 注意

伺服软件版本仅在首次进入时获取，对于没有获取到的版本信息显示为“无”。

伺服参数备份与恢复

点击底部“备份与恢复”功能按键，进入“伺服参数备份与恢复”的操作界面，如图 7-65 所示。

图7-65 伺服参数备份与恢复



- 点击“备份”，提示输入密码，使用管理人员密码确认操作，密码验证通过后，控制器将在本地备份当前的部分 Pn 参数。需要注意的是，备份之后的文件将会覆盖原有的备份文件。
- 恢复伺服参数可分为：恢复用户备份的参数和恢复出厂备份的参数两种，通过单选按钮选择类型。

恢复用户备份的参数：即恢复由用户“备份”或“导入”功能的伺服参数。

恢复出厂备份的参数：在出厂时，对于每个机型都会有一个伺服出厂参数备份文件，恢复时即将这个备份文件中的参数恢复到伺服中。

点击单选按钮选择备份类型，然后点击“恢复”，提示输入密码，使用管理人员密码确认操作，密码验证通过后，控制器将执行恢复操作。
- 插入 U 盘，点击“导出”可将用户备份的参数文件下载到 U 盘根目录中。
- 插入 U 盘，点击“导入”，提示输入密码，使用管理人员密码确认操作，密码验证通过后，选择导入的伺服备份参数文件。



注意

- 进行备份与恢复操作之前需要断开伺服使能。
- 进行“备份”和“导入”时会将原备份的文件覆盖，如果需要保留原备份文件可使用“导出”功能。

7.15 二维视觉的使用

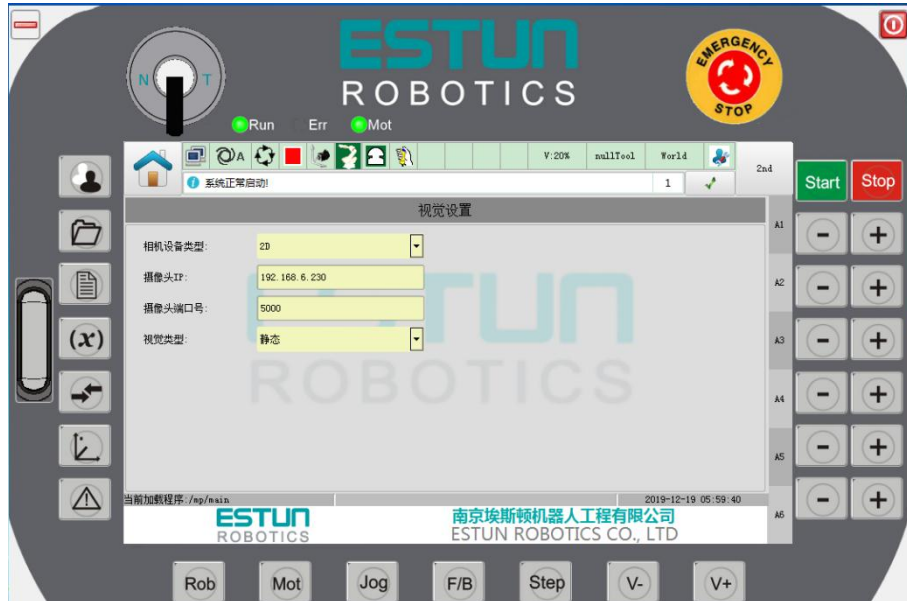
功能概述

机器人视觉是指使机器人具有视觉感知功能的系统，是机器人系统组成的重要部分之一。机器人视觉可以通过视觉传感器获取环境的二维图像，并通过视觉处理器进行分析和解释，进而转换为符号，让机器人能够辨识物体，并确定其位置。

## 使用方法

步骤 1 在系统主页，选择“通用设置>视觉设置”进入视觉设置界面如图 7-66 所示。

图7-66 二维视觉设置界面



步骤 2 “相机设备类型”选择 2D。

步骤 3 “摄像头 IP”和“摄像头端口号”根据实际的参数进行填写，默认为 192.168.6.230:5000。

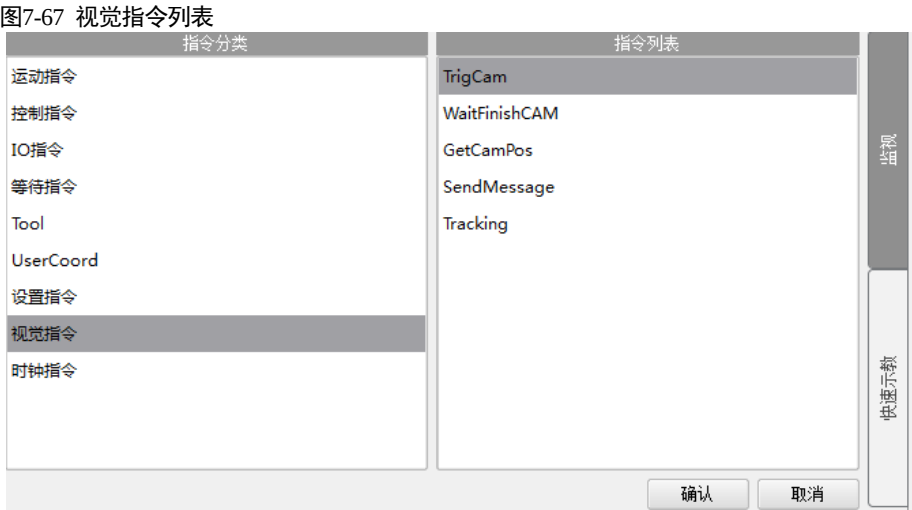
步骤 4 视觉类型目前仅支持静态。

## 参数介绍

- 相机设备类型：设置二维视图或三维视图。  
取值范围：2D、3D、2D 客户端。
- 摄像头 IP：默认 IP 为：192.168.6.230。
- 摄像头端口号：默认值为 5000。
- 视觉类型：分为动态和静态两种，视觉类型目前仅支持静态。

## 视觉命令介绍

视觉指令列表如图 7-67 所示。



TrigCam

触发相机拍摄指令。

```
TrigCam ("OK", 1)
```

字符串“OK”触发相机拍照。

WaitFinishCAM

等待相机拍摄完成的指令。

```
WaitFinishCAM (1000,1)
```

GetCamPos

获取相机位置的指令。

```
GetCamPos (P6, INT0, INI1, 1)
```

7.16 三维视觉的使用

功能概述

机器人视觉是指使机器人具有视觉感知功能的系统，是机器人系统组成的重要部分之一。机器人视觉可以通过视觉传感器获取环境的三维图像，并通过视觉处理器进行分析和解释，进而转换为命令，用来设置机器人的变量、位置点、IO 同时也需要获取机器人的变量、位置点、IO、当前位置等命令，让机器人能够辨识物体，受视觉逻辑控制。

使用方法一

机器人作为客户端，视觉设备作服务端。

步骤 1 在系统主页，选择“通用设置>视觉设置”进入视觉设置界面如图 7-68 所示。

图7-68 三维视觉设置界面



- 步骤 2 “相机设备类型” 选择 3D
- 步骤 3 “摄像头 IP” 和 “摄像头端口号” 根据实际的参数进行填写，默认为 192.168.6.230:6000。

说明

视觉类型在 3D 模式下无作用。

- 步骤 4 通过使用发送数据指令 SendMessage 向视觉设置发送字符串，字符串内容可由用户自定义，例如用户需要发送字符串"ScanReque"到相机，则执行 SendMessage("ScanReque",1)即可。

视觉根据传感器获取环境的三维图像，并进行分析和解释，然后转换成视觉命令来设置或者获取机器人的变量、位置点、IO 等。

视觉对机器人的命令如表 7-2 所示。

表7-2 视觉命令

|                |                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------|
| 修改位置点          | SetPointPos_1                                                                                                |
| 获取位置点          | CamGetPoint_s                                                                                                |
| 修改变量值          | CamSetVar_1_s                                                                                                |
| 获取变量值          | CamReadVar_s                                                                                                 |
| 获取机器人当前关节位置    | GetCurJPos                                                                                                   |
| 获取机器人当前世界坐标系位置 | GetCurWPos                                                                                                   |
| 修改 IO 口值       | SetIOValue                                                                                                   |
| 获取 IO（物理和虚拟）口值 | IOGetDout<br>IOGetDin<br>IOGetAout<br>IOGetAin<br>IOGetSimDout<br>IOGetSimDin<br>IOGetSimAout<br>IOGetSimAin |



## 使用方法二

机器人作服务端，视觉设备作客户端。其机器人服务端的配置如下：

- IP: 192.168.6.63（控制器 IP 地址）
- Port: 5000

步骤 1 建立连接。

1. 控制器内部启动 python 脚本，创建 server 服务端。
2. 配置客户端设备连通机器人。（可 ping 通控制器 IP192.168.6.63）
3. 配置客户端设备，使其使用上述连接参数。
4. 与服务器建立连接。（此时客户端应在每十秒内向服务器发送一次读写请求，以作为心跳保持连接）

步骤 2 视觉客户端向服务器发送命令。

视觉对机器人的命令如表 7-3 所示。

表7-3 视觉命令

|                  |                   |
|------------------|-------------------|
| 获取机器人当前关节坐标      | GetCurJPos        |
| 获取机器人当前世界坐标      | GetCurWPos        |
| 读取变量值            | ReadVar           |
| 修改变量值            | SetVar            |
| 修改变量值，掉电保存       | SetVar_S          |
| 修改位置点的坐标值        | SetPointPos       |
| 修改位置变量点的坐标值，掉电保存 | SetPointPos_S     |
| 读取位置变量点的坐标值      | GetPointPos       |
| 修改多个位置变量点的坐标值    | SetMultiPointPos  |
| 读取多个位置变量点的坐标值    | GetMultiPointPos  |
| 修改 IO 端口值        | SetIOValue        |
| 获取 IO 端口值        | GetIOValue        |
| 修改多个 IO 端口值      | SetMultiIOValue   |
| 获取多个 IO 端口值      | GetMultiIOValue   |
| 获取机器人运行模式        | GetCurSysMode     |
| 获取机器人运行状态        | GetRobotRunStatus |
| 清除报错             | ResetError        |
| 获取报警号            | GetErrorId        |
| 启动示教程序运行         | StartRun          |
| 停止示教程序运行         | StopRun           |
| 加载程序             | LoadUserPrjProg   |
| 判断机器人是否运动        | IsRobotMoving     |
| 判断是否加载程序         | IsProgramLoaded   |

|                    |                     |
|--------------------|---------------------|
| 设置程序运行指针           | SetPc               |
| 切换机器人模式            | ChangeMode          |
| 设置全局速度             | SetGlobalSpeed      |
| 获取当前全局速度           | GetGlobalSpeed      |
| 设置再现程序的运行模式        | SetRunMode          |
| 注销工程               | UnloadUserPrj       |
| 注销用户程序             | UnloadUserProg      |
| 设置伺服使能             | SetMotServoStatus   |
| 获取机器人伺服状态          | GetMotServoStatus   |
| 设置点动坐标系            | SetCoordType        |
| 获取点动坐标系            | GetCoordType        |
| 单轴点动运行             | JogMotion           |
| 单轴点动停止             | JogMotionStop       |
| 单轴寸动               | JogDisMotion        |
| 多轴定点运动(带速度参数无过渡参数) | MoveToSelectPoint   |
| 多轴定点运动(带速度参数和过度参数) | MoveToSelectPointb  |
| 多轴运动停止             | StopDestPosMotion   |
| 示教指定点              | TeachSelectPoint    |
| 设置负载               | SetPayload          |
| 选择工具坐标系            | SetTool             |
| 选择用户坐标系            | SetCoord            |
| 设置机器人报错停止          | SetRTtoErr          |
| 发送运动命令             | SetCurrentMotionCmd |
| 寻位运动               | MoveWithSearch      |

---结束

## 7.17 ModbusTCP

### 功能概述

ModbusTCP 协议接口是指外部逻辑控制器（PLC 等）通过标准工业总线协议（ModbusTCP）与机器人通讯，读写机器人的虚拟 IO 端口的一种通讯方式。机器人可通过 ModbusTCP 的方式与外部设备进行交互，而无需额外扩展 IO 端口。

埃斯顿机器人控制器作为服务器，通过 ModbusTCP 协议提供为外部控制设备提供以下功能：

- 访问与控制虚拟 I/O

- 读取当前机器人运行状态
- 通过控制字启动/停止程序、使能机器人、加载/注销程序、设置全局速度

如表 7-4 所示，ER 系列机器人在 ModBusTCP 保持寄存器中封装了一层数据结构

表7-4 ModBusTCP 保持寄存器数据分区表

| 类型          | 地址          | 定义                |
|-------------|-------------|-------------------|
| Rob Send    | 40003       | 读写标志位应答           |
|             | 40004       | Rob 状态信息          |
|             | 40005~40014 | 当前加载工程名           |
|             | 40015-40018 | 用户用（虚拟数字输出）       |
|             | 40019       | Rob 执行命令状态        |
|             | 40020~40051 | 用户用（虚拟模拟输出）       |
| Rob Receive | 40052       | Rob 操作指令          |
|             | 40053       | 全局速度设置            |
|             | 40054~40063 | 设置加载工程名           |
|             | 40064-40067 | 用户用（虚拟数字输入）       |
|             | 40068~40099 | 用户用（虚拟输入）         |
|             | 40100       | 读写标志位请求           |
| General     | 40101~41500 | 用户用（开放到内部 PLC 读写） |

## 使用方法

ER 系列机器人的 ModBusTCP 接口配置如下：

- IP: 192.168.6.68（实时系统 IP 地址）
- Port: 502
- 采集周期>50ms
- 响应超时>100ms
- 心跳时间<10s

步骤 3 建立连接。

1. 配置客户端设备连通机器人。（可 ping 通实时系统 IP192.168.6.68）
2. 配置客户端设备，使其使用上述连接参数。
3. 与服务器建立连接。（此时客户端应在每十秒内向服务器发送一次读写请求，以作为心跳保持连接）

步骤 4 读写 SimIO。

如下表所示，虚拟输出为 SimOutput，客户端应读取相应寄存器的值，并映射为外部设备的输入，虚拟输入为 SimInput，客户端应向相应寄存器写入值，并映射为外部设备的输出。

表7-5 SimIO 映射表

|             |             |
|-------------|-------------|
| 40015-40018 | 用户用（虚拟数字输出） |
| 40020~40051 | 用户用（虚拟模拟输出） |

|             |             |
|-------------|-------------|
| 40015-40018 | 用户用（虚拟数字输出） |
| 40064-40067 | 用户用（虚拟数字输入） |
| 40068~40099 | 用户用（虚拟输入）   |

步骤 5 读取机器人状态。

服务器的 40004 寄存器为机器人状态字，40002 为机器人当前全局速度。

以下为机器人状态字各 bit 的含义：

表7-6 机器人状态字含义

|      |         |
|------|---------|
| Bit0 | 手动操作模式  |
| Bit1 | 自动操作模式  |
| Bit2 | 远程操作模式  |
| Bit3 | 使能状态    |
| Bit4 | 运行状态    |
| Bit5 | 错误状态    |
| Bit6 | 程序运行状态  |
| Bit7 | 机器人正在动作 |

客户端可通过 mask 将相应的 Bit 读取，以获取机器人状态。

步骤 6 下发控制命令。

按照以下步骤，可向机器人下发启动/停止程序、加载/注销程序、设置全局速度等命令。

- 1. 将命令控制字 40052 写为 0，将 40100 写为 0 以复位命令执行接口
- 2. 将 40100 写为十六进制的 0x11
- 3. 当 40003 的值为 0x11 时，根据下表，将控制字相应的 Bit 置位。
- 4. 当命令执行状态字 40019 相应的 Bit 为 1 时，表示相应指令执行成功。
- 5. 当 40100 为 0 时，40019 被复位。

表7-7 机器人命令执行系统数据表

|       |         |
|-------|---------|
| 40003 | 读写标志位应答 |
| 40019 | 命令执行状态字 |
| 40052 | 控制字     |
| 40100 | 读写标志位请求 |

表7-8 机器人控制字含义

|      |         |
|------|---------|
| Bit0 | 保留      |
| Bit1 | 保留      |
| Bit2 | 机器人程序启动 |

|       |          |
|-------|----------|
| Bit3  | 机器人程序停止  |
| Bit4  | 机器人错误复位  |
| Bit5  | 保留       |
| Bit6  | 保留       |
| Bit7  | 加载工程文件   |
| Bit8  | 注销当前工程文件 |
| Bit9  | 设置全局速度   |
| Bit10 | 指令状态机重置  |

表7-9 机器人命令执行状态字含义

|       |           |
|-------|-----------|
| Bit0  | 命令为 0     |
| Bit1  | 急停命令执行成功  |
| Bit2  | 启动命令执行成功  |
| Bit3  | 停止命令执行成功  |
| Bit4  | 复位命令执行成功  |
| Bit5  | 上使能命令成功状态 |
| Bit6  | 下使能命令成功   |
| Bit7  | 加载工程命令成功  |
| Bit8  | 注销工程命令成功  |
| Bit9  | 设置全局速度成功  |
| Bit10 | 等待控制权     |
| Bit11 | 等待命令      |
| Bit12 | 等待命令执行完成  |
| Bit13 | 命令执行错误    |
| Bit14 | 保留        |
| Bit15 | 保留        |

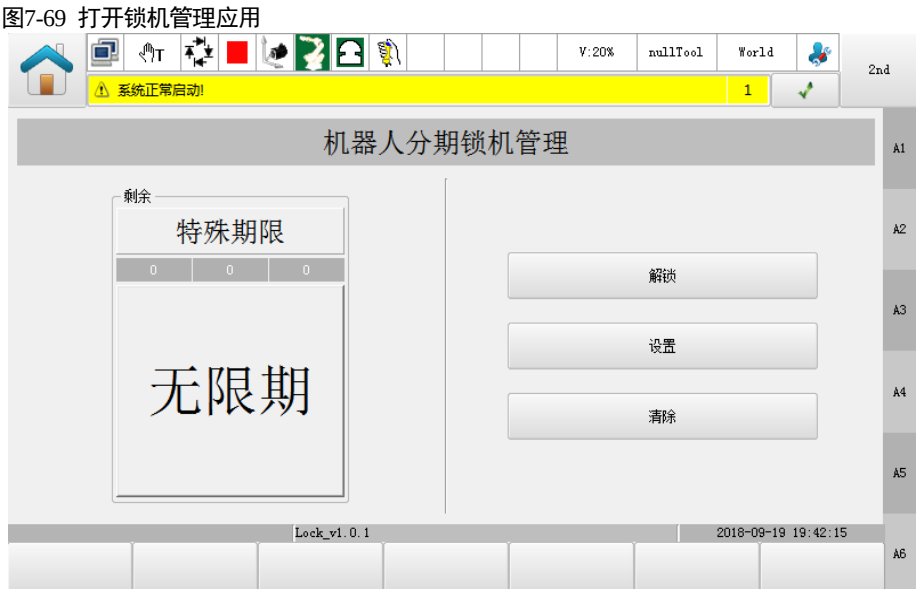
----结束

## 7.18 锁机与解除

- 1. 机器人锁机功能，是指通过机器人锁机应用程序，对机器人的使用权限设定时间范围，满足设定时间范围机器人可正常操作运行，超出时间范围，机器人停止工作并锁定运动功能。
- 2. 机器人可处于四种锁机状态：机器人锁定状态，1 期解锁状态，2 期解锁状态，3 期解锁状态，无限期解锁状态。
- 3. 机器人出厂未做锁机设定，默认为无限期解锁状态，设置锁定机器人后，无法执行机器人运动等相关操作。
- 4. 锁机天数的设定和清除由厂商通过动态密码进行登录操作，期限解锁通过获取厂商解锁密钥，可由用户自行操作完成。
- 5. 需使用“用户”及以上的权限角色。

### 打开锁机管理应用

步骤 1 在系统主页选择“高级设置>锁机管理”，进入机器人分期锁机管理应用界面，如图 7-69 所示。



- 剩余：显示当前期限类型，当前期限剩余天数。若永久解除锁机功能显示期限为“特殊期限”，剩余天数为：“无限期”；
- 解锁：打开解除期限锁定界面，该界面用户可操作。
- 设置：打开设置期限天数界面，该界面需厂商人员权限，使用动态密码登录后操作
- 清除：打开永久解锁机器人设置界面，该界面需厂商人员权限，使用动态密码登录后操作。

----结束

### 设置锁机期限

步骤 1 打开机器人分期锁机管理应用界面，点击“设置”按钮，示教编程器显示如图 7-70 所示，根据提示的动态码，提供给厂商人员获取登录密钥，并登录。

图7-70 登录



步骤 2 登录后如图 7-71 所示，输入第 1、2、3 期的限制天数，并点击“设置”按钮使设置的天数生效。

图7-71 设置期限



说明

- 机器人软件出厂安装后，在没有对锁机进行配置的情况下，默认不做锁机限制，可永久使用全部功能。
- 三期的天数需全部填写，并一起设置生效。
- 设置后即开始生效第 1 期锁机功能，起始时间为当前解锁时间。

----结束

解除锁机期限

步骤 1 打开机器人分期锁机管理应用界面，点击“解锁”按钮，示教编程器显示如图 7-72 所示，提供**机器码**和需解锁的**期数**给厂商人员，以获取解锁密钥，机器码包括两部分，格式为：固定码 - 动态码。

图7-72 解除锁机期限



步骤 2 解锁对应期限后，该期限天数开始生效，在剩余天数界面可刷新查看当前的剩余天数。

说明

- 解锁后当天天数立即生效，生效起始时间为当前解锁时间。
- 若提前解锁，只生效当前期限内的天数，默认前面期限天数失效。

----结束

清除锁机期限

步骤 1 打开机器人分期锁机管理应用界面，点击“清除”按钮，示教编程器显示如图 7-73 所示，根据提示的动态码，提供给厂商人员获取登录秘钥，并登录。

图7-73 清除锁定登录



步骤 2 登录清除锁定设置界面如图 7-74 所示，根据提示的机器码，提供给厂商人员获取清除解锁秘钥。



图7-74 清除锁定设置



步骤 3 清除锁定功能后，机器人进入无限期解锁状态，可在剩余天数界面，刷新查看当前期限无“特殊期限”，剩余天数为“无限期”。

----结束

## 7.19 软浮动功能

### 功能概述

软浮动功能主要应用于机器人从压铸机内取出工件的场合。机器人的软浮动功能，指将机器人变为一个柔性体，机器人在受到外界力的作用下能够顺应外力的变化而运动，以防刚性接触造成的机器人本体及工件损坏。本功能可以沿着指定方向实现直线软浮动功能，机器人可以跟随压铸机顶杆直线运动。

- 机器人工件抓取、机床上下料时，毛坯工件的抓取位置与定位位置存在差异。直线软浮动功能可以在不使用外部传感器或者增加机械结构的情况下使机器人由刚性状态变为柔性状态。机器人末端可以顺应外力沿指定方向直线运动，与工件完全贴合，避免碰撞与摩擦。主要应用于压铸等工件上下料领域。
- 软浮动开启时，机器人的运动速度由外力决定，外力越大，机器人运动越快。
- 软浮动功能支持以下模式下开启。
  - 自动运行（单步、连续）
  - 手动运行（单步、连续）



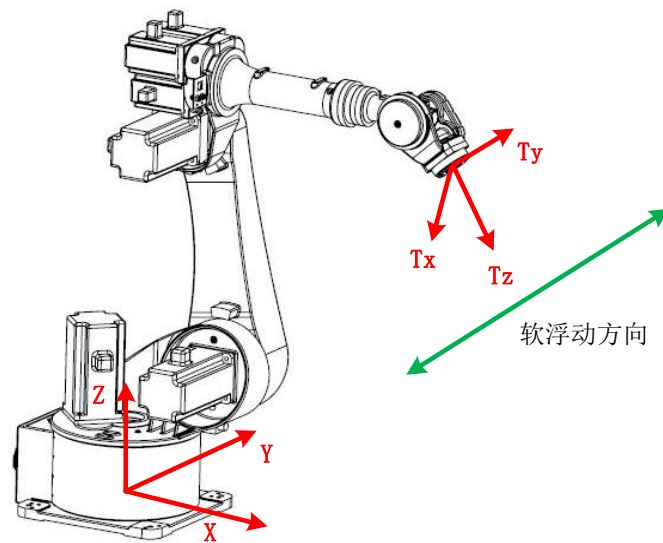
### 注意

- 在使用软浮动功能时请指定正确的软浮动方向确保与工件实际运动的方向一致。
- 软浮动开启后，出现下列情况，软浮动功能关闭。
  - 控制器报错
  - 停止按钮或者急停按钮被按下
  - 软浮动开启超过系统设置的等待时间
- 软浮动开始指令与结束指令需要配合使用，且先设置软浮动开始指令再设置软浮动结束指令。若想在同一个示教程序中执行多组软浮动功能，应设置多组开始与结束指令。以下指令设置时，控制系统会报错。
  - 只执行软浮动开始指令
  - 只执行软浮动关闭指令
  - 先执行软浮动结束指令，再执行软浮动开启指令
  - 连续执行软浮动开始指令
- 软浮动开始和结束指令中插入动作指令，控制系统会报错。
- 请不要在奇异点附近使用软浮动功能。
- 请在外力撤销后再停止软浮动功能，以防止机器人停止时振动。
- 若软浮动开始指令的上条运动指令中含有过渡信息，不进行过渡处理。
- 软浮动开启时，碰撞检测以及最大力矩检测功能关闭，软浮动结束或者软浮动功能开启后发生报错时，碰撞检测和最大力矩检测功能恢复。

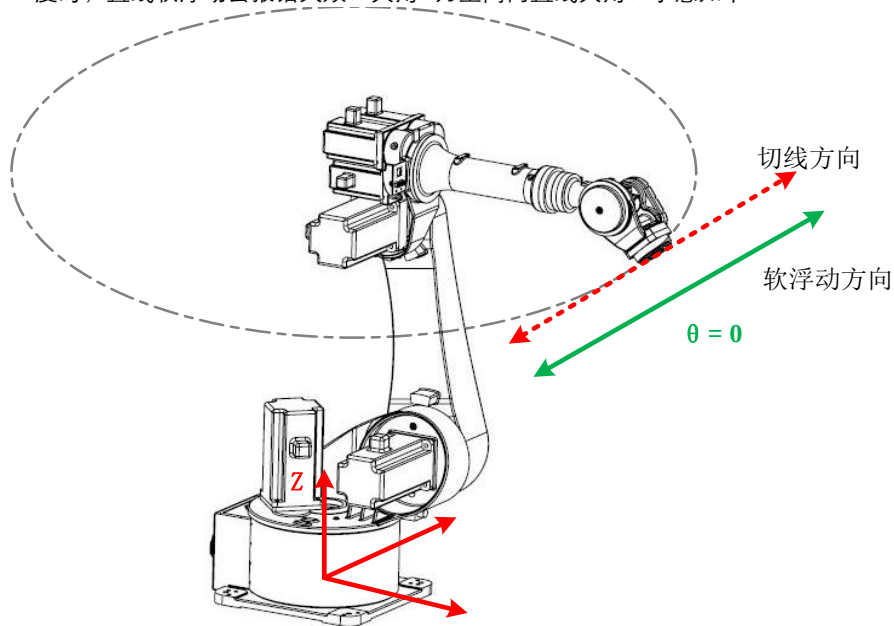
其中，设定软浮动方向的说明如下：

使用软浮动功能取件前，首先确认软浮动方向是否设置正确。当参考坐标系选择工具坐标系时，直线运动方向为软浮动开始点时的所选工具坐标系方向，软浮动工具坐标系的方向与点动操作机器人时工具坐标系方向

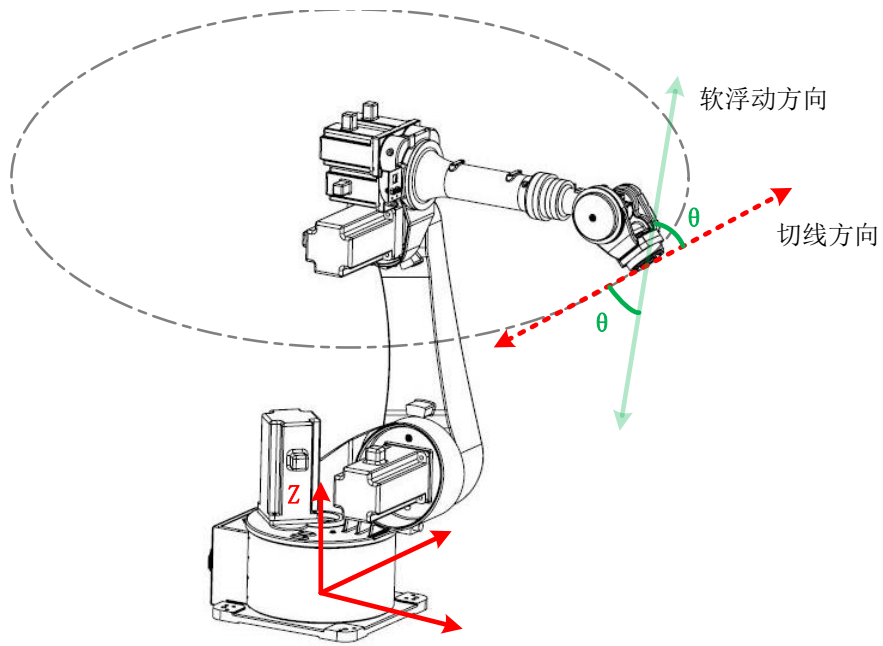
一致。当参考坐标系选择 TOOL、坐标系号选择 nullTool、软浮动方向选择 Y 时，直线软浮动方向示意图如下：



软浮动设置方向与机器人受力点绕世界坐标系 Z 轴水平旋转的切线方向平行时，柔顺效果最佳；夹角 $\theta$ 大于 60 度时，直线软浮动会报错失效。夹角 $\theta$ 为空间两直线夹角。示意图如下：



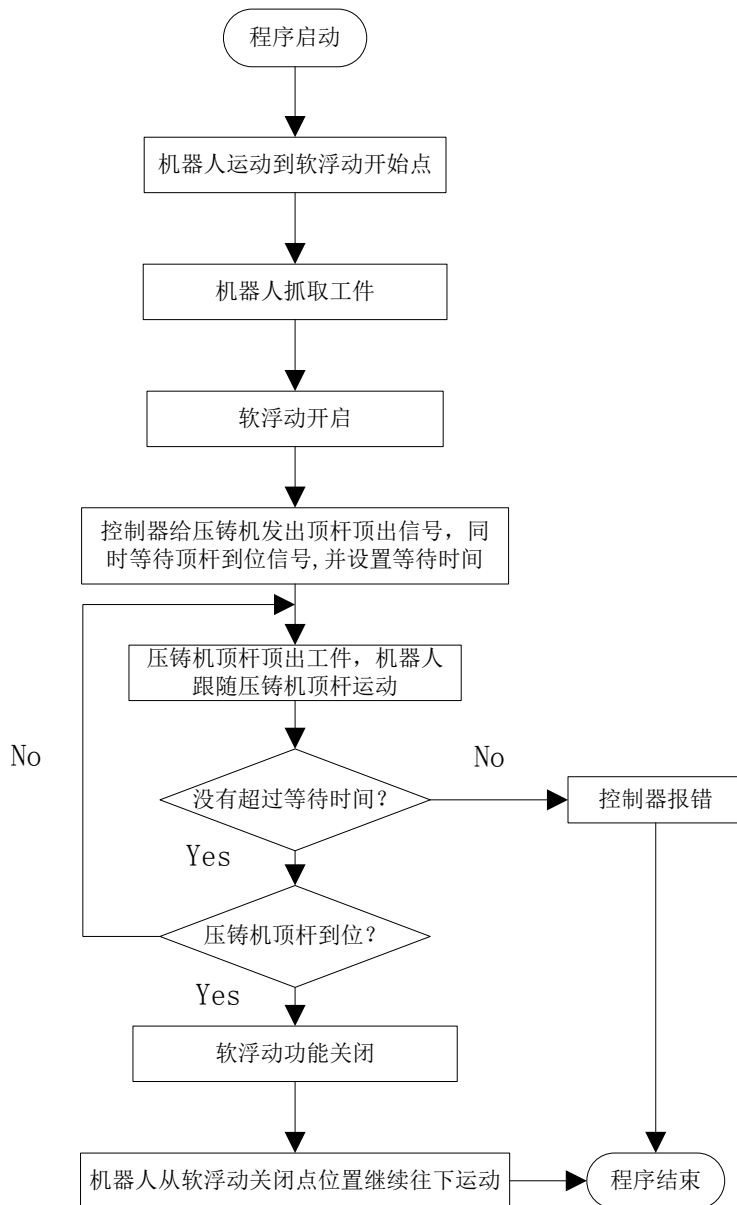
切线方向与软浮动方向平行，柔顺度佳。



切线方向与软浮动方向夹角较大，柔顺效果较差。

### 应用示例

以软浮动功能应用于从压铸机内取件为例，软浮动现场应用示例可参考如下流程：



1. 程序运行到软浮动功能开始点时，先抓取工件。
2. 软浮动功能开启后，控制器发送 DO 信号给压铸机，同时等待压铸机顶杆到位信号的 DI 信号输入并设置等待时间；
3. 当压铸机接收到 DO 信号后，压铸机顶杆顶出，机器人可以顺应压铸机顶杆运动；当超过设置的等待时间时，控制器报警；
4. 当控制器接收到顶杆到位信号后，软浮动结束。
5. 机器人从软浮动结束的位置继续往下运动。

示教程序示例：

```

MovJ(P0,V100,"RELATIVE",C0) /*机器人运动到软浮动开始点 P0*/
SetDO(DO0,0) /*控制抓手抓住工件*/
Wait(500) /*等待 0.5s*/
SoftFloatStart(CART,TOOL,nullTool,Y,LOW) /*直线软浮动功能开启，机器人可以沿着无工具参数的工具坐标系 Y 方向直线运动，软浮动柔度等级为 LOW*/
SetDO(DO0,1) /*发出压铸机顶杆前进信号*/
WaitDI(DIO,0,8000,0,INT0) /*控制器等待压铸机顶杆到位信号并设置等待时间 8s，返回到位信号状态 INT0*/
IF INT0.value > 0 THEN /*判断是否到位*/
SetRtToErr("WaitError",10000) /*超过设置的等待时间 8s 控制器报错*/
ENDIF;
Wait(500) /*等待 0.5s 等待机器人停止运动，等待时间示现场情况而定*/

```

```
SoftFloatStop() /*软浮动功能关闭*/
MovJ(P1,V100,"RELATIVE",C0) /*机器人从软浮动结束点运动到 P1 点*/
```

## 7.20 九点标定功能

### 功能概述

九点标定功能适用于 ESTUN 机器人搭配相机视觉设备的现场应用，通过该功能相机可以直接发送被拍摄物体的像素点，机器人控制器接收到相机像素点后可以在内部转换为机器人世界坐标系下的位置点，机器人根据转化后的位置点实现精确抓取目标物体的功能。

### 使用方法

步骤 1 在系统主页选择“用户应用>九点标定”，进入“九点标定功能”的操作界面，如图 7-75 所示。

图7-75 九点标定功能的操作界面



相机：默认相机  
点位：1  
标定方向：负

误差  
RX: 0  
RY: 0

视觉  
X(pix): 0  
Y(pix): 0

机器人  
X(mm): 0  
Y(mm): 0  
Z(mm): 0

☒ 九点标定使能

九点标定功能使用说明：  
1、“相机”- 选择要设置的相机索引号  
2、“点位”- 选择九点中一点  
3、“标定方向”- 修改视觉坐标方向  
4、“九点标定使能”- 启动或关闭九点标定  
5、“示教”- 在机器人类型下获取机器人当前坐标  
6、“保存”- 保存当前类型下当前点的坐标  
7、“计算”- 计算转换矩阵并将绝对误差显示在RX和RY中  
8、“视觉”- 视觉像素点  
9、“机器人”- 机器人坐标点

示教  
保存  
计算

2018-09-11 00:36:43

标定 相机配置 帮助

步骤 2 设置视觉坐标和机器人坐标。

将标定纸放在相机的下方，保证 9 个标定纸点皆在相机的拍摄范围内，相机拍摄完后生成 9 个像素位置点，记录下九个像素位置点。然后手动操作机器人，将机器人 TCP 点依次走到标定纸上 1~9 的位置，走到一个标定纸点，先将对应像素点坐标输入视觉框中，再点击示教显示机器人坐标点，最后点击保存按钮。按照标定纸

上 1~9 顺序依次进行。

1. 选择对应相机，每个相机都有对应的标定参数。
2. 初始界面不可操作，勾选右上角打开九点标定使能。
3. 手动操作示教编程器移动机器人至标定点 1，选择点位 1，手动输入标定点 1 对应的像素位置点，保持机器人 TCP 点与标定纸上的 1 点重合，然后点击示教当前机器人坐标点，点击保存。
4. 重复上述操作，依次选择标定点 2~9，手动输入标定点对应的像素位置点，每次输入后都需点击保存。

#### 说明

标定方向选择：当相机工作坐标系的 Z 方向与机器人世界坐标系的 Z 方向相同时，这里选择正方向；当相机工作坐标系的 Z 方向与机器人世界坐标系的 Z 方向相反时，这里选择负方向。

每个点示教完后都需点击保存按钮

#### 步骤 3 计算矩阵。

设置完视觉像素位置点和机器人坐标点后，点击计算按钮，计算误差显示在 RX 和 RY 处，正常误差在  $\pm 0.1$  范围内；计算结束后保持九点标定使能一直处于使能状态，接着可以通过相机发送拍照的目标物体点至机器人端，观察机器人能否正确运行至目标物体点。

#### “帮助”菜单

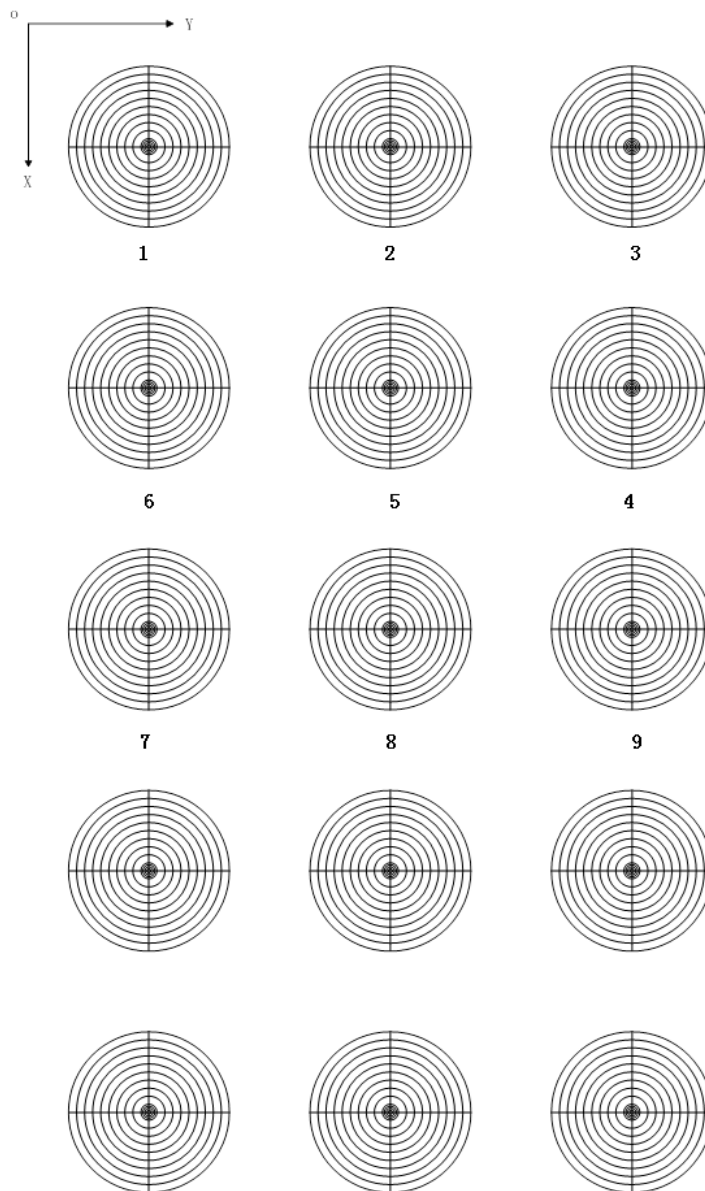
点击右下角帮助按钮，标定界面切换到带有使用说明的操作页面，左侧区域也可以使用，使用方法同上面章节所

述，点击左下角功能按钮，再切回到原画面。



- 标定图纸

视觉与机器人标定点的序号必须要与下图标定纸中的顺序相同。



----结束



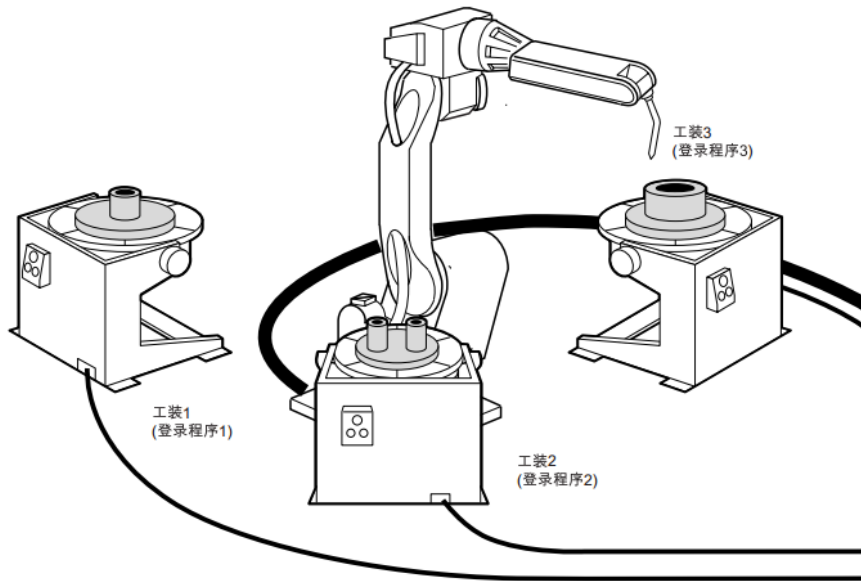
## 7.21 多工位预约

### 功能概述

使用多工位预约功能，可以使机器人无间歇地完成至多 4 个工位上的作业。

机器人多工位预约功能，是指将机器人工作区域内的每一个工位与不同的输入/输出进行绑定，同时赋予这些输入/输出启动与停止的功能，其后用户可以通过给该输入/输出有效信号，让机器人自动的执行与之绑定的工位程序。并且可以根据给信号的顺序，依次启动对应工装上的程序。

下图给出一个机器人的作业示例：其工作顺序依次为工装 1、工装 2 和工装 3；各工装上的作业依次为程序 1、程序 2 和程序 3。



3 个工装各自处理不同工件的加工，对应不同的加工程序，假设依次为程序 1、程序 2、程序 3，在再现模式下启动有下面两种效果：

- 当前没有任何程序执行时，用户给工装 1 启动信号，机器人会自动的加载并运行程序 1；用户给工装 2 启动信号，机器人会自动的加载并运行程序 2；工装 3 的启动方式与前两者一样。
- 当前机器人正在执行程序的情况：假设当前正在执行程序 1，且工装 2、3 都已准备好工件时，用户先给工装 3 启动信号，再给工装 2 启动信号，此时由于机器人正在执行工装 1，那么系统会依次记录工装 3 跟工装 2 的预约等待状态。当程序 1 执行完时机器人会先执行程序 3，待程序 3 执行完后再执行程序 2，从而完成整个预约流程。

机器人作业时，用户可通过示教编程器查看各工位上工作状态和 IO 信号。

## 使用方法

正确使用该功能，用户需要完成如下工作：

- 已经正确编辑了各工位的程序，程序的编辑方法，请参见 4.2 程序编辑。
- 每个工位的工装上的启动按钮能够正常使用。

步骤 1 在 Home 界面上“用户应用”插件组中点击“多工位预约”，如图 7-76 和图 7-77 所示。

图7-76 用户应用插件组



图7-77 点击多工位预约



步骤 2 进入“多工位预约设置”后，勾选“多工位预约使能”，并勾选需要进行作业的“工位”。

图7-78 勾选工位

多工位预约设置

多工位预约状态

☒ 多工位预约使能

|      | 使能                                  | 程序名         | 启动输入 | 停止输入 | 状态输出 | 停止输出 |
|------|-------------------------------------|-------------|------|------|------|------|
| 工位01 | <input checked="" type="checkbox"/> | yuyue1.main | Di9  | Di10 | Do17 | Do18 |
| 工位02 | <input checked="" type="checkbox"/> | yuyue2.main | Di11 | Di12 | Do19 | Do20 |
| 工位03 | <input checked="" type="checkbox"/> | yuyue3.main | Di13 | Di14 | Do21 | Do22 |
| 工位04 | <input checked="" type="checkbox"/> | yuyue4.main | Di15 | Di16 | Do23 | Do24 |

请点击表头获取帮助信息

步骤 3 点击各“工位”所在行的“程序名”。

图7-79 点击程序名

多工位预约设置

多工位预约状态

☒ 多工位预约使能

|      | 使能                                  | 程序名         | 启动输入 | 停止输入 | 状态输出 | 停止输出 |
|------|-------------------------------------|-------------|------|------|------|------|
| 工位01 | <input checked="" type="checkbox"/> | yuyue1.main | Di9  | Di10 | Do17 | Do18 |
| 工位02 | <input checked="" type="checkbox"/> | yuyue2.main | Di11 | Di12 | Do19 | Do20 |
| 工位03 | <input checked="" type="checkbox"/> | yuyue3.main | Di13 | Di14 | Do21 | Do22 |
| 工位04 | <input checked="" type="checkbox"/> | yuyue4.main | Di15 | Di16 | Do23 | Do24 |

请点击表头获取帮助信息

步骤 4 在弹出的“工程列表”中选择想要进行的程序。

图7-80 选择程序

工程列表

加载状态

运行状态

自启动

☐ estun

☐ asd

☐ ert

☐ main

☐ str

☒ yuyue1

☐ main

☒ yuyue2

☐ main

☒ yuyue3

☐ main

2017-07-11 01:52:13

后退

清除

选择

- 点击“后退”，可返回至“多工位预约设置”界面。
- 点击“清除”，可重置该“工位”的预设“程序”。
- 点击“选择”，可将当前所选的“程序”预设至该“工位”，并返回至“多工位预约设置”界面。

步骤 5 为各“工位”正确地分配“启动输入”、“停止输入”、“状态输出”和“停止输出”的信号。

注意：统一 IO 端口，不要重复使用。

图7-81 选择 IO 信号

多工位预约设置

多工位预约状态

☒ 多工位预约使能

|      | 使能                                  | 程序名         | 启动输入 | 停止输入 | 状态输出 | 停止输出 |
|------|-------------------------------------|-------------|------|------|------|------|
| 工位01 | <input checked="" type="checkbox"/> | yuyue1.main | Di9  | Di10 | Do17 | Do18 |
| 工位02 | <input checked="" type="checkbox"/> | yuyue2.main | Di11 | Di12 | Do19 | Do20 |
| 工位03 | <input checked="" type="checkbox"/> | yuyue3.main | Di13 | Di14 | Do21 | Do22 |
| 工位04 | <input checked="" type="checkbox"/> | yuyue4.main | Di15 | Di16 | Do23 | Do24 |

请点击表头获取帮助信息

用户可点击界面所列出的表格表头的各个单元格，来查看帮助信息。

图7-82 查看帮助

多工位预约设置

多工位预约状态

☒ 多工位预约使能

|      | 使能                                  | 程序名         | 启动输入 | 停止输入 | 状态输出 | 停止输出 |
|------|-------------------------------------|-------------|------|------|------|------|
| 工位01 | <input checked="" type="checkbox"/> | yuyue1.main | Di9  | Di10 | Do17 | Do18 |
| 工位02 | <input checked="" type="checkbox"/> | yuyue2.main | Di11 | Di12 | Do19 | Do20 |
| 工位03 | <input checked="" type="checkbox"/> | yuyue3.main | Di13 | Di14 | Do21 | Do22 |
| 工位04 | <input checked="" type="checkbox"/> | yuyue4.main | Di15 | Di16 | Do23 | Do24 |

单个工位的启动输入口，该输入口的上升沿信号为有效信号。

请点击表头获取帮助信息

步骤 6 点击“多工位预约状态”，可查看各“工位”的状态及其 IO 信号状态。

图7-83 查看状态

多工位预约设置

多工位预约状态

|      | 程序名         | 状态  | 启动输入 | 停止输入 | 状态输出 | 停止输出 |
|------|-------------|-----|------|------|------|------|
| 工位01 | yuyue1.main | 未预约 |      |      |      |      |
| 工位02 | yuyue2.main | 未预约 |      |      |      |      |
| 工位03 | yuyue3.main | 未预约 |      |      |      |      |
| 工位04 | yuyue4.main | 未预约 |      |      |      |      |

与单个工位绑定的示教程序名。以yuyue1.main为例说明，“yuyue1”为工程名，“.”分隔符，“main”为程序名。

请点击表头获取帮助信息

- 说明
- 设置完多工位预约参数后，需将系统转到再现模式，系统才会进行多工位预约操作。
  - 当某一程序未执行且预约成功时，如果想取消当前预约操作，再次给启动输入信号即可取消预约。
- 结束

7.22 变位机功能

功能概述

使用变位机功能，可以通过工作台的回转和翻转来移动工件至焊接的合适位置，并与机器人协同运动，以此提高生产效率和焊接质量。

用户最多可配置 6 个变位机，变位机类型支持 1 轴旋转变位机及 2 轴旋转变位机。

注：本章节截图中附加轴名称 Ex 等同于 PEx。

标定变位机坐标系

步骤 1 在系统主页上选择“用户应用>变位机”，如图 7-84 所示。

图7-84 选择变位机



步骤 2 选择需要设定的“变位机 ID”，如选择变位机 ID= “1”。

图7-85 选择需要设定变位机 ID



步骤 3 点击“修改”，对变位机 ID=1 的坐标系进行设定。  
变位机坐标系可以直接输入或进行标定获取。

图7-86 变位机 ID=1 的设定或示教



步骤 4 按照需要，设定“轴数”。

- 默认为“1”，表示 1 个附加轴。用户只需要对 1 个附加轴进行标定。
- 设定为“2”，表示 2 个附加轴。用户需要对 2 个附加轴进行标定。

设定轴数为“2”时，通过点击“切换轴”来选择需要标定的附加轴。



注意

设定为 2 个附加轴时，在标定“1 轴坐标系”时，需要确保变位机 2 轴对应的附加轴处于零位；反之亦然。  
变位机 1 轴对应的附加轴应选择变位机的底座轴。

步骤 5 选择“附加轴”，然后点击“标定”。

图7-87 选择附加轴并准备标定

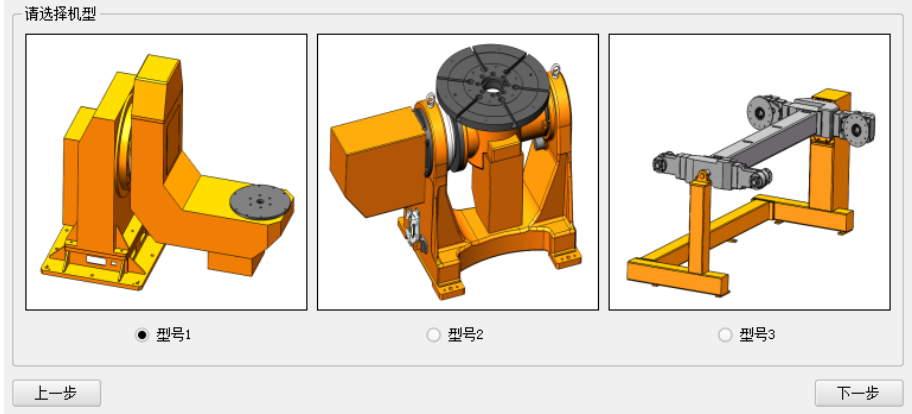


注意

标定过程中，请确保该变位机轴沿着正方向旋转示教，并保持变位机的其他轴处于零位。

步骤 6 选择变位机的机型，如选择“型号 1”。

图7-88 选择变位机的机型



说明

对于型号 3 对应的“三轴变位机”，可配置成两个两轴变位机。

步骤 7 点击“下一步”，将对变位机进行三点示教。

【标定方法】

1. 请在标定前安装一个尖端装置或寻找一个特征点，作为标志位置。
2. 移动机器人 TCP 至该标志位置，然后点击“示教”。
3. 点击“下一步”，并继续旋转变位机轴至一定角度（建议大于 30°）后，移动机器人 TCP 至该标志位置，然后点击“示教”。
4. 点击“下一步”，并第二次旋转变位机轴至一定角度（建议大于 30°）后，移动机器人 TCP 至该标志位置，然后点击“示教”。
5. 点击“下一步”，将显示标定“结果”。

步骤 8 确认无误后，点击“确定”。



#### 说明

标定“2 轴坐标系”时，请先确保变位机 1 轴对应的附加轴处于零位。然后点击“切换轴”，并使用同样的方法对 2 轴坐标系进行标定。

步骤 9 返回变位机坐标系界面后，点击“确定”，以成功标定变位机 ID=1 的坐标系。

----结束

## 使用变位机坐标系

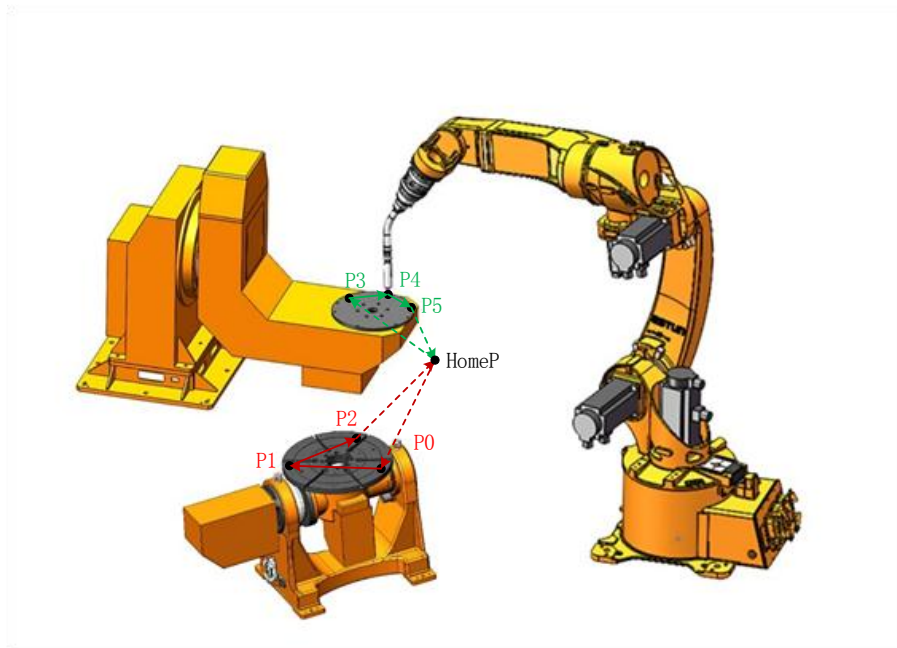
使用变位机坐标系前，需要创建 POSITIONER 变量，并指定其中的参数 index 为正确的变位机 ID。然后，可通过两种方式进行：

- 直接使用指令“SetPositioner”，表示切换系统坐标系至变位机坐标系。
- 在使用如下运动指令时，设定“坐标系参数”为指定的 POSITIONER 变量。  
MovL、MovC、MovLRel、MovLSearch、MovH、MovLW、MovCW。

## 使用示例

机器人左右两边各有一个变位机，机器人先与左边 U 型变位机（已设变量 POSITIONER1）进行同步运动，其后再与右边 L 型变位机（已设变量 POSITIONER0）进行同步运动。指令结构如图 7-89 所示。

图7-89 变位机功能使用例图



其执行指令如下所示。

```

Start:
MovJ{P=t_l.HomeP} /*机器人移动到 HomeP 点*/
MovL{P=t_l.P0} /*机器人运动到 1 号变位机起始点*/
MovL{P=t_l.P1,Coord=t_g.POSITIONER1}
/*机器人先将当前的用户坐标系切换成变位机坐标系 POSITIONER1，其后以默认的速度以及过渡参数移动到 P1 点*/
MovL{P=t_l.P2} /*变位机与机器人进行同步运动*/
MovL{P=t_l.HomeP} /*机器人返回到 HomeP 点*/
MovL{P=t_l.P3} /*机器人运动到 0 号变位机起始点*/
SetPositioner{Positioner=t_g.POSITIONER0}
/*将当前变位机坐标系 POSITIONER1 切换成变位机坐标系 POSITIONER0*/
MovL{P=t_l.P4} /*在变位机坐标系 POSITIONER0 下进行同步运动*/
MovL{P=t_l.P5} /*在变位机坐标系 POSITIONER0 下进行同步运动*/
MovL{P=t_l.HomeP,Coord=t_s.World}
/*机器人将变位机坐标系切换成世界坐标系，并以默认的速度以及过渡参数移动到 HomeP 点*/

```



End;

7.23 倒序执行

功能概述

程序执行正常的复杂运动指令时，有时需要按原轨迹返回，避免发生碰撞等，引入此功能。在上述情况时，进入倒序模式执行，即可原轨迹返回。

在倒序模式下，允许倒序执行的运动指令将会倒序执行，不允许倒序执行的运动或其它指令，将会报警提醒，其它不影响原轨迹返回的指令，将被跳过，不执行。

使用方法

手动模式下，按“F/B”功能键，进入倒序执行模式，再次按下“F/B”功能键，则返回正序模式。



注意

- 使用倒序功能时请注意如下事项。
- 程序注销或加载后，自动还原为正序执行。
  - 倒序功能仅限手动单步执行。
  - 使用倒序功能时，遇到可跳过的指令，直接跳过，不会执行。
  - 使用倒序功能时，遇到不可执行的指令，系统将报警提示。
  - 倒序执行时，SetPC 等导致运行路径变化，为人工强制干预，风险由操作人承担。
  - 倒序功能不可跨程序，不支持由 CALL 程序回主程序。

使用倒序功能时，可执行和可跳过的指令如表 7-10 所示，未列出的指令均为不可执行的指令。



说明

END 行或被注释的指令行，以及注释行或 Label 指令行，均默认为可跳过。

表7-10 倒序执行指令说明

| 类型  | 指令名称           |               |                |                |       |
|-----|----------------|---------------|----------------|----------------|-------|
| 可执行 | MovJ           | MovL          | MovC           | MovLW          | MovCW |
|     | MovLOffset     |               |                |                |       |
| 可跳过 | SetDO          | SetAO         | SetSimDO       | SetSimAO       |       |
|     | WaitDI         | WaitDI8421    | WaitAI         | WaitSimDI      |       |
|     | WaitSimDI8421  | WaitSimAI     | PulseOut       | PulseSimOut    |       |
|     | GetDI8421      | GetSimDI8421  | SetDO8421      | SetSimDO8421   |       |
|     | GetSimDIToVar  | SetSimDOByVar | GetSimAIToVar  | SetSimAOByVar  |       |
|     | SetDIEdge      | SetSimDIEdge  | SetJointDyn    | SetCartDyn     |       |
|     | SetOverRide    | SetPayload    | GetCurOverRide | Hand           |       |
|     | CalcTool       | CalcCoord     | SetRtToErr     | SetRtInfo      |       |
|     | SetRtWarning   | Stop          | RefRobotAxis   | SetMotionMode  |       |
|     | Wait           | WaitFinish    | WaitCondition  | BitAnd         |       |
|     | BitNeg         | BitOr         | BitXOr         | BitLSH         |       |
|     | BitRSH         | CLKStart      | CLKStop        | CLKRead        |       |
|     | CLKReset       | TrigCam       | WaitFinishCAM  | GetCamPos      |       |
|     | SendMessage    | SocketCreate  | SocketClose    | SocketSendStr  |       |
|     | SocketReadReal | SocketReadInt | SocketReadStr  | SocketSendReal |       |
|     | SocketSendInt  |               |                |                |       |

## 7.24 伺服参数自调节功能

### 功能概述

伺服参数自调节功能，该功能是为改善机器人各种应用中在低速运行阶段的抖动，同时也将改善因抖动造成的点位定位误差较大的现象，可通过设置相应的参数在各个转速区间达到最佳效果。

- 伺服参数自调节功能主要用于改善机器人在某一速度区间段内的低速抖动现象。执行该功能开启指令 `AutoGainEnable()`，并设置指令参数：增益比例阈值、速度区间阈值；  
比例阈值（`LsScale`）：对应各个关节轴，设置范围：[100, 1000]，单位：%；  
速度阈值（`LsThresh`）：对应各个电机轴，设置范围：[10, 1000]，单位：r/min；
- 伺服参数自调节过程依赖于伺服参数 `Pn104`，如修改 `Pn104`，在修改前和后使用同一参数（比例阈值），效果稍有差异。
- 执行伺服参数自调节功能关闭指令 `AutoGainDisable()`后，该功能即关闭。



### 注意

- 伺服参数自调节功能允许开启，需满足以下条件：  
→ 配置文件 `EAxisParam.ini` 中需配置使能项  
→ 配置文件夹中需配置相应伺服参数文件 `AutoGainServoParam.json`
- 伺服参数自调节功能默认不开启，执行开启指令后，所在程序注销/执行关闭指令后关闭该功能。

### 应用示例

程序执行伺服参数自调节功能开始指令时，即开启该功能。伺服参数自调节功能开启后，控制器将自动调节伺服相应参数；当设置的指令参数值超过最大限值时，控制器报警。

示教程序示例：

```
/*程序执行段，伺服参数自调节功能默认关闭*/
.
.
AutoGainEnable(LsScaleP=t_g.LsScale0,LsThreshP=t_g.LsThresh0)/*伺服参数自调节功能开启*/
.
.MovL(P1) //执行该运动指令时控制器将自动调节伺服相应参数
.
AutoGainDisable()/*伺服参数自调节功能关闭*/
.
```

## 7.25 跨腕部奇异功能

### 功能概述

跨腕部奇异功能可以使机器人沿着示教的路径通过腕部奇异点位置（J5=0情况。当示教起终点J5轴位置反向时，腕部奇异发生），从而提升机器人的运动空间。跨腕部奇异不能保证轨迹运动过程中的姿态。

跨腕部奇异功能可以通过指令形式开启关闭。

- 通过设置指令 SetSingularPass(“字符串”)设置跨腕部奇异功能的开关。  
其中，字符串可选“OPEN”或“CLOSE”。当选择“OPEN”时为打开跨腕部奇异功能，当选择“CLOSE”时为关闭跨腕部奇异功能。



### 注意

- 开启腕部奇异功能时，不能保证示教轨迹过程中的姿态，请先慢速执行运动防止机器人发生碰撞；
- 当前段若不跨腕部奇异，用户即使选了跨奇异方式，还是按照不跨奇异方式运动；
- 跨奇异功能只支持通用六关节机器人腕部发生奇异的情况，不支持其他机器人类型或其他奇异位形；
- 跨腕部奇异功能支持运动指令间的过渡，但 SetSingularPass 指令打断过渡；
- 点动不支持跨奇异；
- 摆弧运动、跟随运动、外部 TCP 功能不支持跨奇异；
- 支持起点在奇异位置，不支持终点在奇异位置；
- 注销加载工程时，是否跨奇异复位为默认值。

### 应用示例

程序执行跨奇异功能开始指令时，即开启该功能。开启后，若不执行跨奇异关闭指令，该功能一直开启。

示教程序示例：

```
Start:
SetSingularPass("OPEN")/*跨腕部奇异功能开启 */
MovJ(P0);
MovL(P1); /*若 P0 至 P1 的运动不跨奇异或者只跨了腕部奇异，则机器人运动至 P1 */
MovC(P2,P3); /*若 P1 至 P3 的运动不跨奇异或者只跨了腕部奇异，则机器人运动至 P3 */
SetSingularPass("CLOSE")/*跨腕部奇异功能关闭 */
End;
```

## 7.26 回零功能

### 7.26.1 功能概述

零位校准是将机器人位置与绝对编码器位置进行对照的操作。回零操作一般是在出厂前进行的，但在下列情况下会出现零位丢失，必须再次进行零位校准。

- 更换了电机、绝对编码器时，导致零位信息丢失；
- 编码器电池故障等导致的零位位置信息丢失；
- 其它情况。

如图 7-90 所示，为零位信息丢失的状态（unRef：零位信息丢失）：

图7-90 机器人零位丢失状态图

|    | 值    | 单位  | 状态    |
|----|------|-----|-------|
| A1 | 0.00 | deg | unRef |
| A2 | 0.00 | deg | unRef |
| A3 | 0.00 | deg | unRef |
| A4 | 0.00 | deg | unRef |
| A5 | 0.00 | deg | unRef |

零位丢失



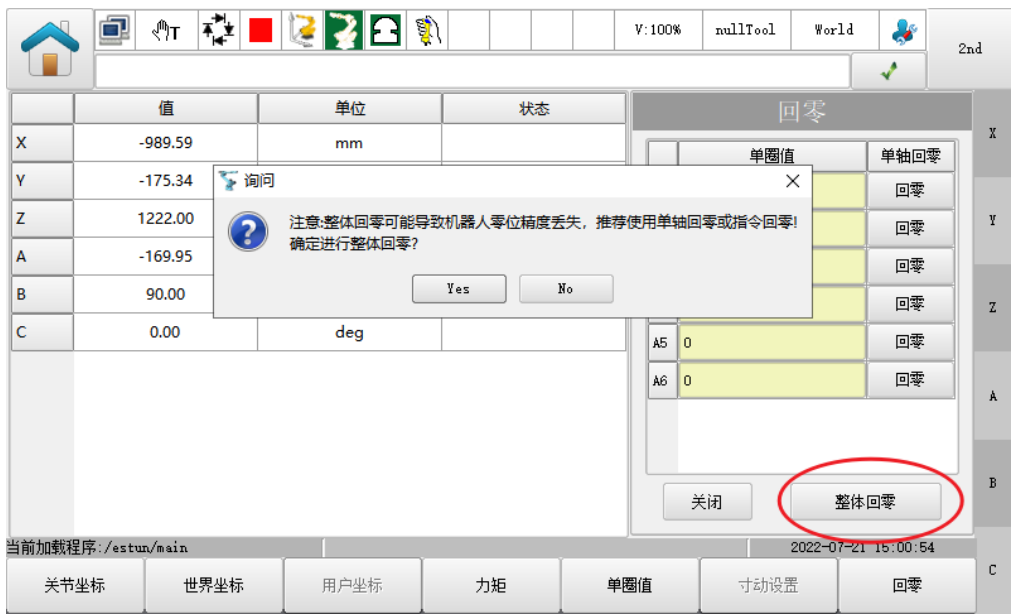
注意

- 本章节中的操作仅适用于上述第二种情况不更换器件时的零位找回（即：编码器电池故障等导致的零位位置信息丢失），其它更换了部件后，需要利用仪器或其它校准方法找回零位；
- 系统提供了两种回零方式：整体回零和单轴回零。

7.26.2 整体回零

整体回零会让机器人轴组内所有轴同时进行回零操作，一般会造成一定的精度丢失，只有在对精度要求不高时采用，如图 7-91 所示。

图7-91 整体回零图



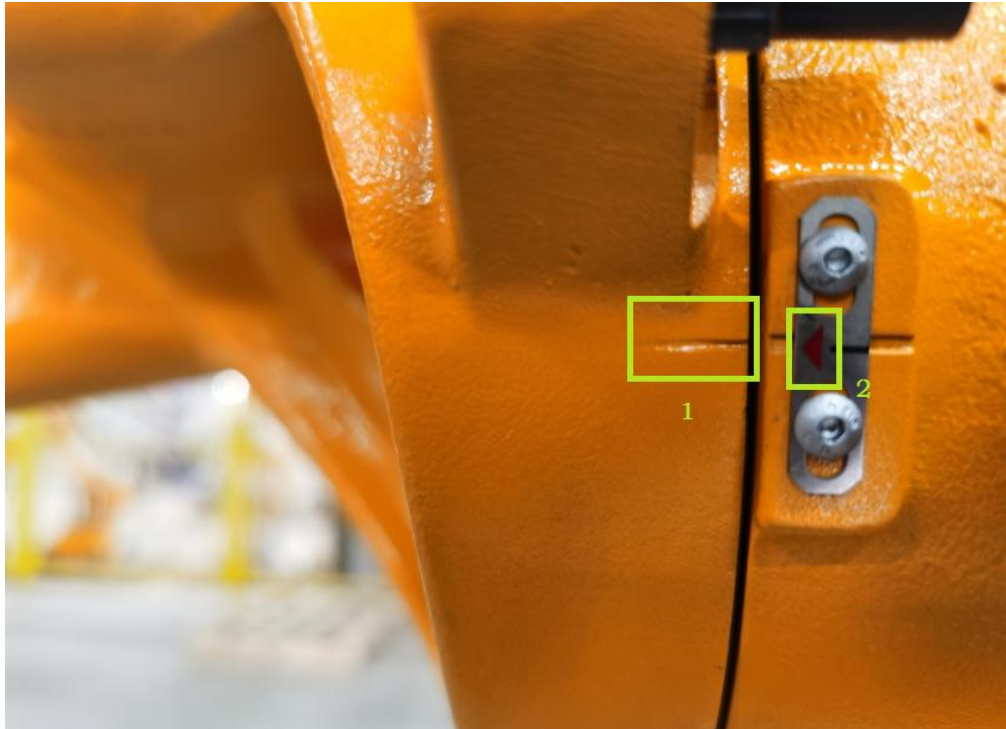
7.26.3 单轴回零

单轴回零是通过对应轴一键回零或 RefRobotAxis 指令进行零位校准的方式。这种形式可以保证机器人原有的精度，在应用中推荐使用。

步骤 1 将机器人轴点动到本体上对应轴的零标刻度线附近，如图 7-92 所示。

一般机器人标定后，会调整零标贴 2 的腰型孔位置，使得与 1 处的刻线对齐。并将标定后，各轴的单圈绝对值数据信息存档。该值可在点动界面的“单圈值”界面的“设置单圈值”中看到。

图7-92 本体上轴的零标刻度线图



### 注意

- 2 处的箭头要通过点动使得与 1 处的刻线尽可能精确对齐。如果 1 刻线宽度太大，则在点动到刻线附近时要注意观察，是否箭头走过 1 处的刻线区域时会导致对应轴的单圈值跨过一圈（比如从第 20000 个脉冲达到最高值后又跨过 20000 脉冲值）。假如 1 的刻线区域使得单圈值跨越一圈，则该轴以单轴回零方式进行回零后仍可能存在精度丢失的问题。

步骤 2 示教编程器“点动管理->位置管理”界面中，选择“单圈值”项查看出厂校准时记录的单圈值信息，如图 7-93 所示。

图7-93 单圈值信息图



步骤 3 执行一键回零或指令回零

- 一键回零

将出厂设置的单圈值数据填充到指令参数中，点击需要回零的轴号后“回零”按钮，一键回零，如图 7-94 所示。

图7-94 一键回零界面



- 指令回零
- 编辑 RefRobotAxis 指令，指定需要回零的轴号，并将出厂设置的单圈值数据填充到指令参数中，如图 7-95 所示。运行 RefRobotAxis 指令（机器人指定轴会走到单圈值位置，然后进行回零的操作）。
- ```
RefRobotAxis (1,2419239)
/*执行该指令，机器人 1 轴将在单圈值为 2419239 处进行零点校准*/
```

图7-95 指令释义图



注意

- 当关节存在耦合时，需要注意回零顺序，即从耦合的源端轴开始顺次回零。比如 456 存在耦合（4 对 5、5 对 6 有耦合）时，回零顺序应为 4->5->6。

----结束

7.27 保养提示

功能概述

该功能主要用于提示机器人及时维修保养，避免机器人因使用时间过长未保养，导致机器人的使用寿命减少。

在使用中分为整机保养和氮气弹簧保养，整机保养又分两种方式进行提示，以下为保养功能介绍：

- 累计运行时长：每一款机型都对应有一个保养执行周期（10000h、11520h 等），当保养执行时长超过这个周期后，则会提示“已超过保养周期或日期，建议及时保养！”
- 使用日期：每一款机型都对应有一个累计使用年份（2 年、3 年等）。从出厂/执行日期开始，到使用年份后，也就是下次保养日期。则提示“已超过保养周期或日期，建议及时保养！”
- 氮气弹簧保养：适配带氮气弹簧的机型，氮气弹簧的检查周期为半年（180 天），从出厂/执行日期开始，到半年（180 天）后，也就是检查日期时，则提示“氮气弹簧使用时间已超过半年，请及时维护检查！”

使用方法

步骤 1 在 Home 界面上选择“用户应用>保养提示”，如图 7-96 和图 7-97。

图7-96 用户应用插件组



图7-97 点击保养提示



步骤 2 进入到“保养提示”界面，如图 7-98 所示：

图7-98 保养提示界面



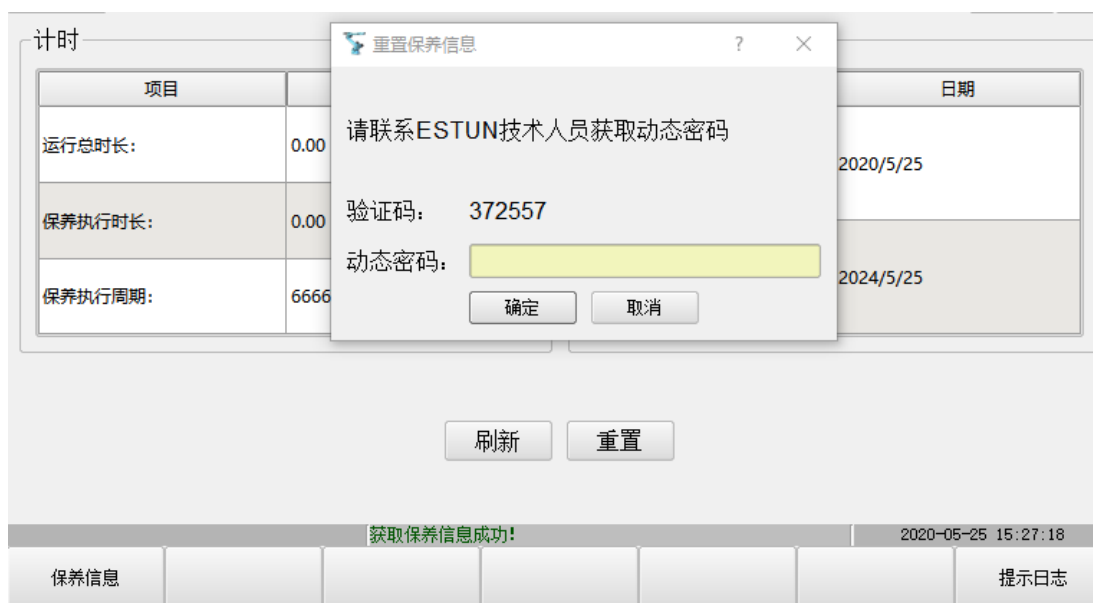
步骤3 界面介绍:

- 运行总时长：从出厂以来机器人累计的运行时间，此数值一直累加，不可清除。
- 保养执行时长：从出厂或重置之后开始累计的运行时间。
- 保养执行周期：每一种机型对应有一个保养的周期（如 10000h、11520h 等），当**保养执行时长**大于**保养执行周期**，则会出现保养提示。
- 出厂/执行日期：机器人的控制系统首次运行的日期，或重置时的日期。
- 下次保养日期：到下次保养的日期，当**系统日期**大于**下次保养日期**，则会出现保养提示。

步骤 4 操作介绍:

- 刷新：点击刷新，可重新获取当前的保养信息。
- 重置：点击重置，需输入动态密码才能执行，通过验证码获取动态密码有专门的软件，请联系相关技术人员获取动态密码。重置会将“保养执行时长”清零，并会将“出厂/执行日期”以及“下次保养日期”设置为当前系统日期。

图7-99 重置保养信息



步骤 5 氮气弹簧保养界面介绍

图7-100 氮气弹簧保养界面



- 出厂/执行日期：机器人的控制系统首次运行的日期，或重置时记录的日期。
- 检查日期：是指从出厂/执行日期开始半年（180 天）后的日期，当**系统日期**大于**检查日期**，则会出现氮气弹簧的保养提示。

步骤 6 氮气弹簧重置

- 重置：将氮气弹簧保养信息重置，**出厂/执行日期**会记录为**系统日期**，重置方式参考步骤 4。

步骤 7 提示信息：

- 提示日志：近期保养提示的信息会保存下来，如下图 7-101 所示，点击“提示日志“，可获取历史提示记录。
- 提示内容介绍：
 - 整机保养：过了保养周期或日期，则会出现提示，提示号为 53006。
 - 氮气弹簧保养：系统日期到检查日期或超过了检查日期，则会出现提示，提示号为 53010。

图7-101 提示日志界面

序号	时间	提示信息
1	2025-01-27 10:51:26	已超过保养周期或日期，建议及时保养！
2	2025-01-27 10:51:19	已超过保养周期或日期，建议及时保养！
3	2025-01-27 10:51:12	已超过保养周期或日期，建议及时保养！
4	2025-01-27 10:51:05	已超过保养周期或日期，建议及时保养！
5	2025-01-27 10:50:58	已超过保养周期或日期，建议及时保养！
6	2025-01-27 10:50:51	已超过保养周期或日期，建议及时保养！
7	2025-01-27 10:50:44	已超过保养周期或日期，建议及时保养！
8	2025-01-27 10:50:37	已超过保养周期或日期，建议及时保养！
9	2025-01-27 10:50:30	已超过保养周期或日期，建议及时保养！

刷新日志

首页

上一页

1/100

下一页

尾页

转至

99

页

获取保养日志成功!

2025-01-27 10:57:21

保养信息						提示日志
------	--	--	--	--	--	------

说明

- 1、 到日期或到保养周期后，请及时保养，如不及时保养，提示信息则会在机器人正常使用时出现提示。
- 2、 氮气弹簧保养：出现氮气保养提示时，请尽快根据机器人维护检查手册对氮气弹簧做检查，检查合格后，请重置氮气弹簧保养信息，以便下次保养提醒。

----结束

参数介绍

- 运行总时长：从出厂以来机器人累计的运行时间，此数值一直累加，不可清除。
- 保养执行时长：从出厂或重置之后开始累计的运行时间。
- 保养执行周期：每一种机型对应有一个保养的周期（如 10000h、11520h 等），当保养执行时长大于保养执行周期，则会出现保养提示。
- 出厂/执行日期：机器人的控制系统首次运行的日期，或重置时的日期。
- 下次保养日期：到下次保养的日期，当系统日期大于下次保养日期，则会出现保养提示。
- 检查日期：是指从出厂/执行日期开始半年（180 天）后的日期，当**系统日期**大于**检查日期**，则会出现氮气弹簧的保养提示。

7.28 Scara 快速标定

功能描述

该功能是 scara 机器人快速标定的一种方法，分为单独定点的 2 轴零点标定，和需要标定杆的杆长标定，该功能只用于 scara 机型。

使用方法

步骤 1 在 Home 界面上选择“用户应用> scara 快速标定”，如图 7-102 和图 7-103。

图7-102 用户应用插件组



图7-103 点击 scara 快速标定



步骤 2 进入到 scara 快速标定界面。

- 2 轴零点标定：该功能只标定 2 轴的零点位置，通过左右手系取同一个定点，点击计算即可得到 2 轴偏差结果，按照偏差结果将 2 轴回零即可。

图7-104 2 轴零点标定



- 杆长标定：该功能是对 scara 大臂和小臂的杆长标定，以及对 2 轴的零点标定，需要一个固定长度的标定杆，标定杆的左右两端各对应两个点，且知道两点之间的长度。该功能目前只对埃斯顿生产人员开放，有密码进入。

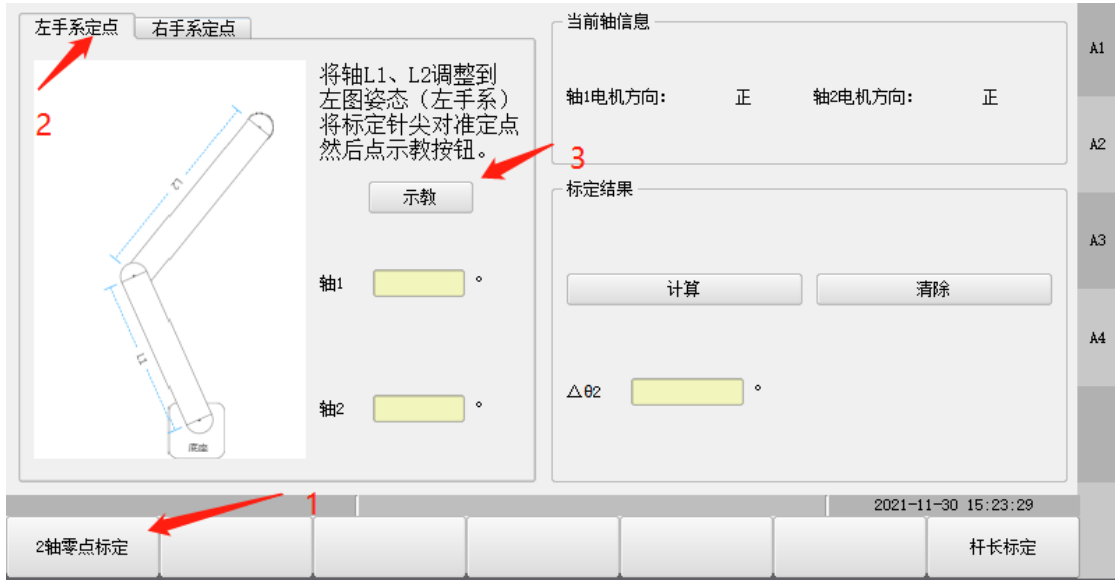
图7-105 杆长标定



步骤 3 操作介绍：

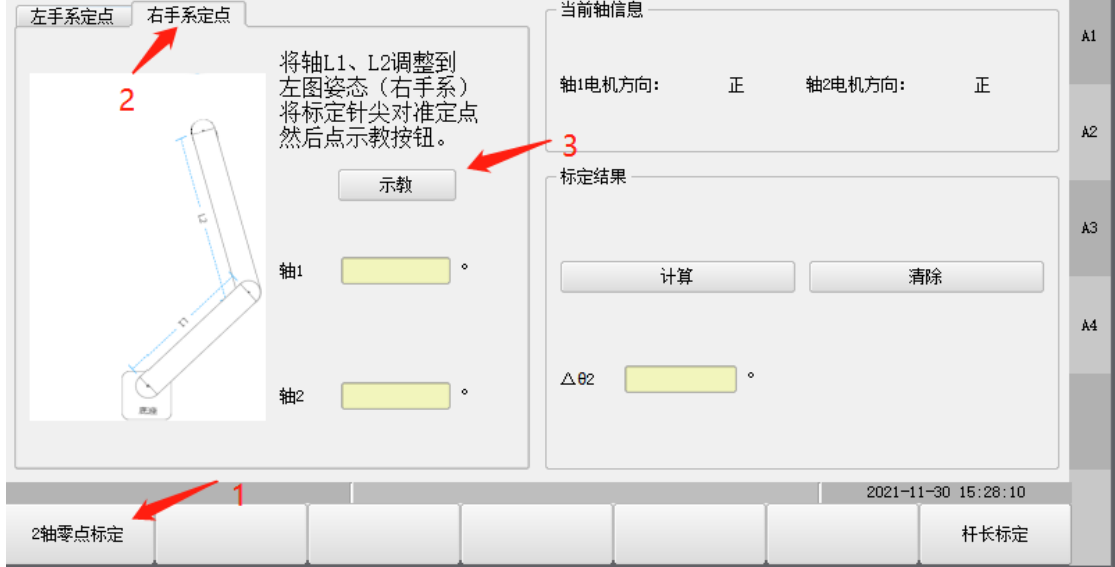
- 2 轴零点标定
 - 左侧 Tab 页切换到左手系定点，将标定针尖对准定点，然后点击“示教”。

图7-106 左手系标定

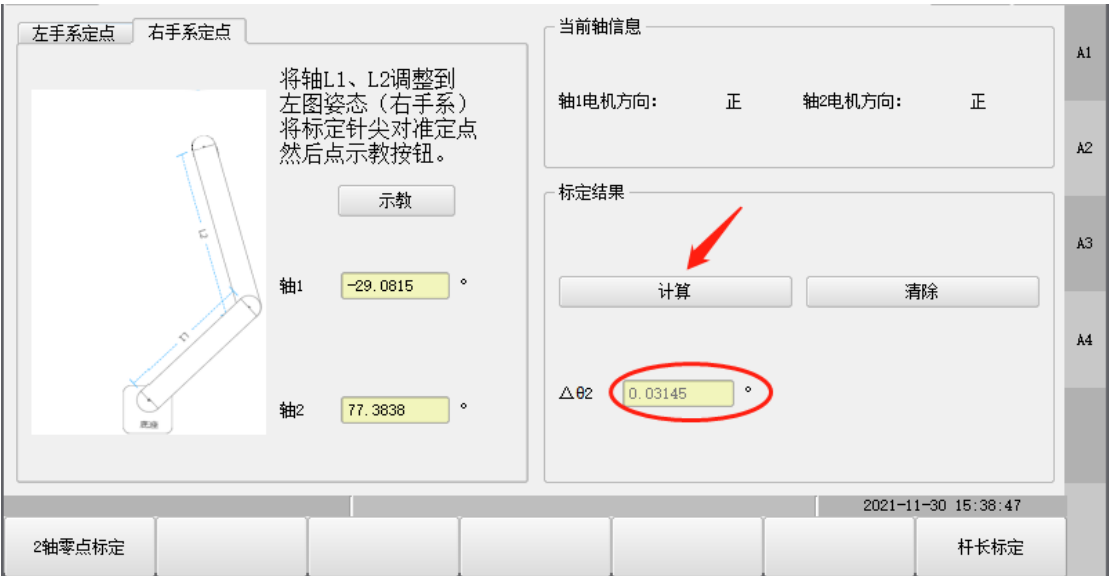


(2) 左侧 Tab 页切换到右手系定点，将标定针尖对准准定点，然后点击“示教”。

图7-107 右手系标定



(3) 点击“计算”，获取到 2 轴零点偏差值。



(4) 切换到回零界面，将 2 轴位置移到偏差值位置，点击“回零”，标定完成。



• 杆长标定

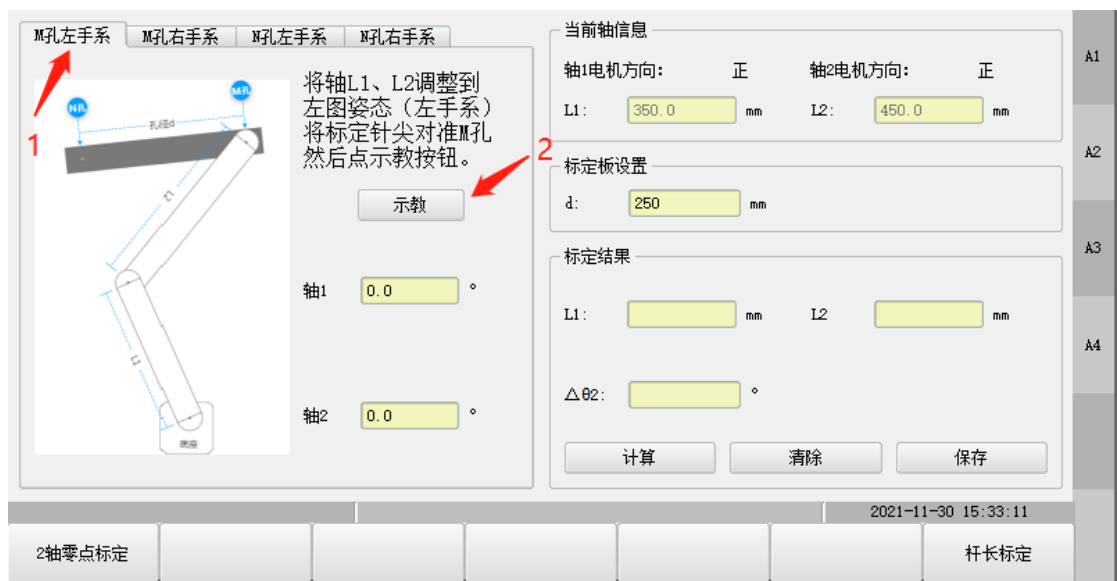
(1) 点击杆长标定，先输入密码，密码只对埃斯顿生产人员开放。

图7-108 密码界面



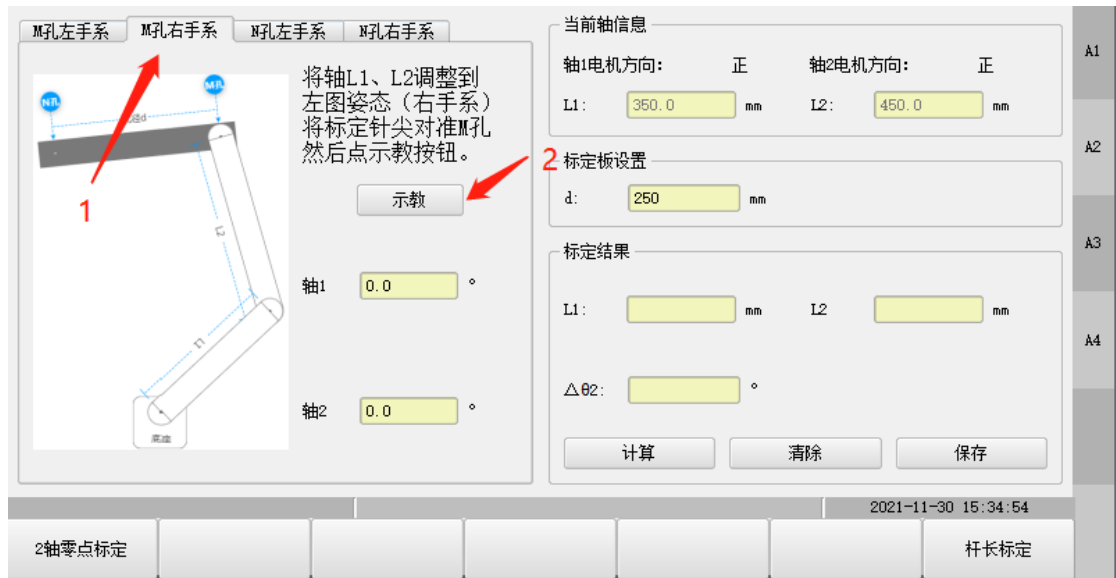
- (2) 左侧 Tab 页切换到 M 孔左手系，将轴 L1 和轴 L2 调整到图中所示姿势，将标定针尖对准 M 孔，然后点“示教”按钮；

图7-109 M 孔左手系



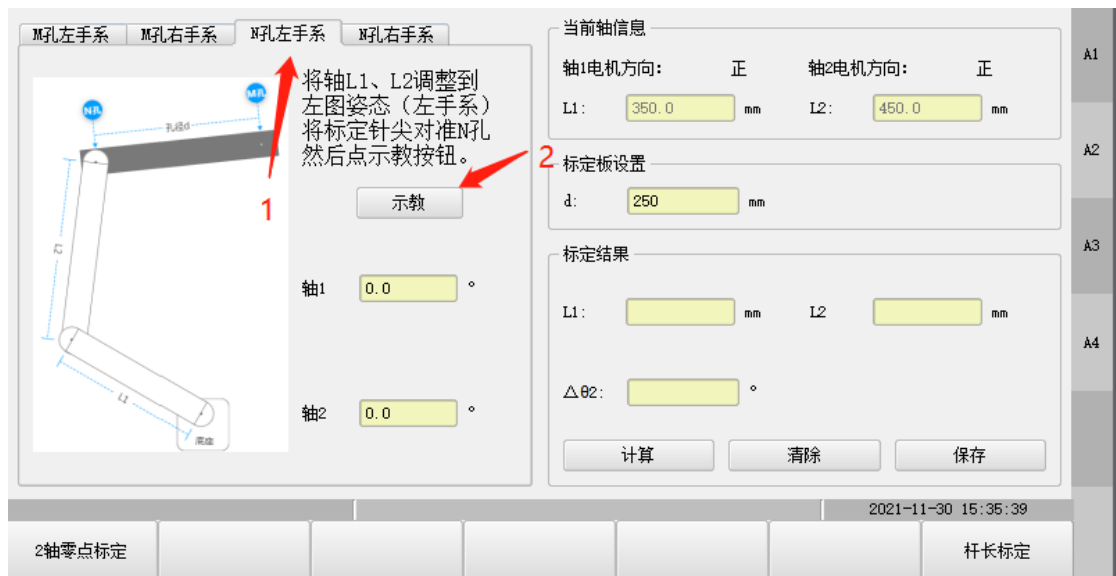
- (3) 左侧 Tab 页切换到 M 孔右手系，将轴 L1 和轴 L2 调整到图中所示姿势，将标定针尖对准 M 孔，然后点击“示教”按钮；

图7-110 M 孔右手系



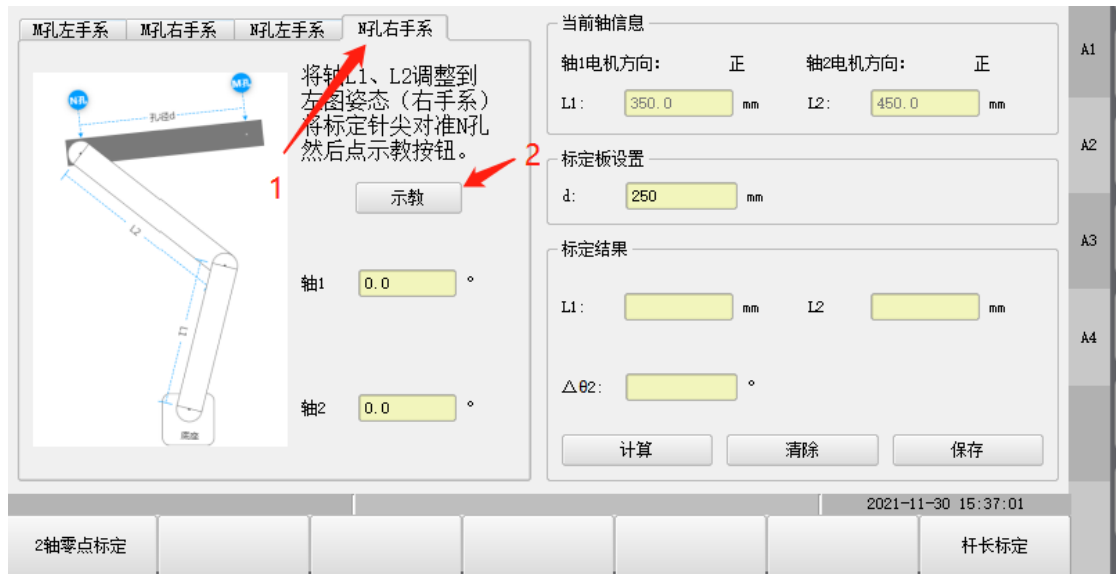
- (4) 左侧 Tab 页切换到 N 孔左手系，将轴 L1 和轴 L2 调整到图中所示姿势，将标定针尖对准 N 孔，然后点击“示教”按钮；

图7-111 N 孔左手系



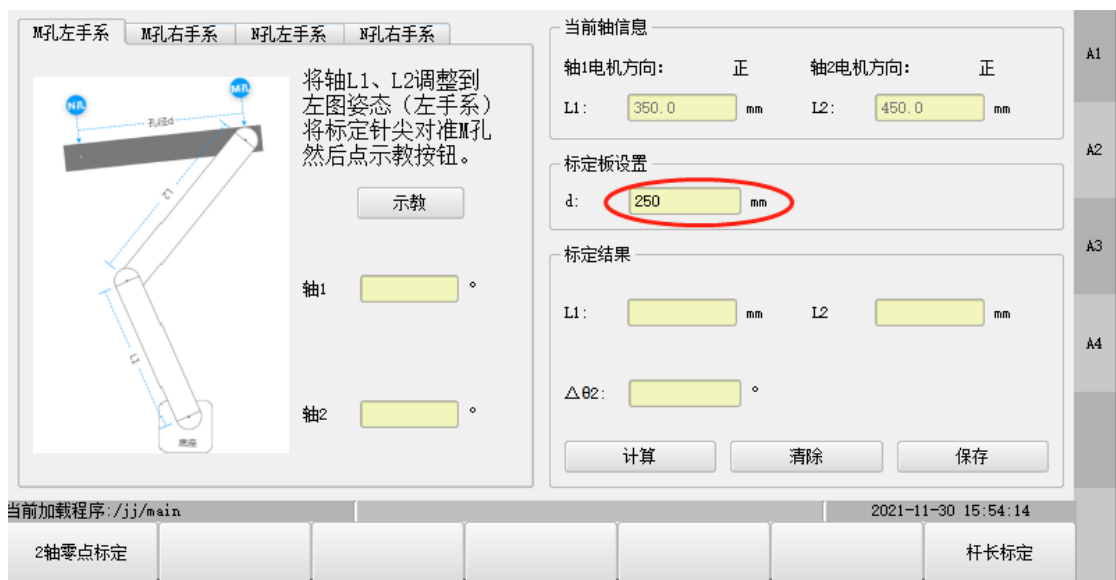
- (5) 左侧 Tab 页切换到 N 孔右手系，将轴 L1 和轴 L2 调整到图中所示姿势，将标定针尖对准 N 孔，然后点击“示教”按钮；

图7-112 N 孔右手系



(6) 输入标定板设置中的标定板长度 d，然后点击“计算”按钮；

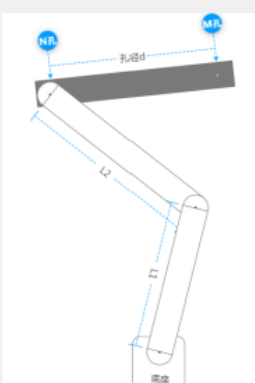
图7-113 标定板设置



(7) 点击“保存”按钮，会将标定结果中的 L1 和 L2 发送给控制器进行存储；

图7-114 获取结果

M孔左手系M孔右手系N孔左手系N孔右手系



将轴L1、L2调整到左图姿态（右手系）
将标定针尖对准N孔
然后点示教按钮。

示教

轴1 -56.4158 °

轴2 83.9461 °

当前轴信息

轴1电机方向：正 轴2电机方向：正

L1: 350.0 mm L2: 450.0 mm

标定板设置

d: 250 mm

标定结果

L1: 350.04 mm L2: 450.46 mm

Δθ2: 0.0239 °

计算清除保存

当前加载程序: /jj/main

2021-11-30 15:56:47

2轴零点标定杆长标定

(8) 切换到回零界面，将 2 轴位置移到偏差值位置，点击“回零”，标定完成。

图7-115 回零界面

	值	单位	状态
A1	0.00	deg	
A2	0.02	deg	
A3	0.00	mm	
A4	0.00	deg	

工具: S:nullTool

参考坐标系: S:World

点动模式: 连续

点动坐标系: 关节坐标系

TCP速度: 0.00 mm/s

当前加载程序: /jj/main

2021-11-30 15:59:02

关节坐标世界坐标用户坐标力矩单圈值寸动设置回零

----结束

7.29 去限位块丢零保护功能

功能概述

- 本功能在机器人不安装机械限位块且零点丢失的情况下，在寻零操作过程中能够一定程度的保护本体内部线缆不被绕坏。
- 当去限位块的机器人轴零点丢失后，示教编程器位置界面会显示机器人丢零前的位置，如下图。以此位置为参考值进行点动，进而执行回零操作。



.

说明

- 默认通用六关节机器人第 4 轴去限位块丢零保护功能开启；
- 当去限位块丢零保护功能开启后，若去限位块的轴丢零，则示教编程器位置界面新增显示各轴丢零前的位置；若去限位块轴未丢零而其他轴丢零，则示教编程器界面不新增显示各轴丢零前的位置。

 注意

- 示教编程器上显示的丢零前位置在以下几种情况下可能不准确:
 - 1) 更换了控制器软件;
 - 2) 机器人在高速运动过程中丢零;
 - 3) 机器人在运动过程中异常断电重启后丢零;
 - 4) 机器人丢零后人为的将关节进行移动。

7.30 一点回原位功能

功能概述

- 在一些现场应用中，需要经常使机器人回到某个特定的原位，来查看机器人点位是否发生偏移。所以本功能实现了通过界面来选择回到的原位，并运行至该位置。
- 功能界面支持显示当前位置以及目标原位位置，支持两个可选的原位位置，并且支持对位置进行示教或者直接修改。
- 选择目标位置后，点击“Go”并保持按下状态，机器人开始以 MovJ 的方式运行至目标位置，在运行过程中松开“Go”，则机器人停止运行，再次点击“Go”可继续运行，直到到达目标位置后停止运行。

图7-117 一点回原位



说明

- 回原位仅显示本体轴。
- 回原位动作需要伺服使能，并且只能在示教模式下执行。
- 可通过按示教编程器 Rob 键快捷到达一键回原位界面。

7.31 全局程序

功能概述

- 该功能提供一种可以被跨工程调用的程序，即建立一种全局工程，该工程中新建的任意程序均可被所有工程中的程序进行调用。例如现场每个工程中都有同样的换抓手程序，在全局工程中新建抓手程序，让其他工程中来调用这个程序。
- 如下图所示，工程界面中新增了"GlobalProg"工程来管理全局程序，加载这个工程后可以进行全局程序的创建、删除、加载、复制、重命名等操作，并可在变量界面中的程序域变量中进行该程序变量相关的操作。在非 GlobalProg 工程的其他工程中，可以使用 call 指令来选择本地程序与 GlobalProg 工程中的全局程序。
- 举例说明，gProg 为"GlobalProg"中的全局程序，工程"prog0"中的 l_1、l_2、l_3、l_4、main 程序新建 CALL 指令时均可选"GlobalProg"中的 gProg 程序来进行调用操作。同理工程"prog1"中的 main 程序新建 CALL 指令时也可选"GlobalProg"中的 gProg 程序来进行调用操作。

图7-118 全局程序示例



说明

- 全局工程不支持删除、重命名以及复制操作。
- 全局工程中无工程域变量，故当系统由其他工程中的程序跳转到全局程序时，变量界面中的工程变量会处于不可见状态，待跳出全局程序后可恢复显示原工程中的工程变量。
- 当前版本全局程序没有做删除校验，使用时需注意，即当其他工程的程序中使用 CALL 指令来调用全局程序 A 时，用户可直接在工程界面选择全局程序 A 进行删除操作。

7.32 从站配置

功能概述

- 从站配置功能用于配置 EtherCAT 通讯中的 IO 模块和伺服从站，使用户可以根据现场拓扑实现灵活配置，方便现场调试。用户可在示教器添加、删除从站，控制器断电重启，EtherCAT 网络通信正常。

使用方法

选择进入系统主页中的通用设置分组，点击从站配置进入配置界面如图 7-119 所示。

- 首先，用户可以根据不同的应用场景和需求，通过从站配置界面点击勾选“从站类型”下拉列表选择伺服类型；
- 其次用户点击勾选“从站型号”下拉列表选取对应的设备型号；
- 最后，用户点击添加或者删除按钮，等待操作完成即可。

图7-119 从站配置界面



说明

- 在使用该功能时，必须了解设备当前的拓扑结构和设备类型。
- 此功能目前支持 EC4-1616 类型的 IO 添加、删除功能；支持 EC4-P20DE 类型编码器添加、删除功能；支持 ESTUN ED3L 类型的伺服添加、删除功能；
- 添加的设备拓扑仅支持 IO 设备-编码器设备-伺服设备这样的拓扑顺序，不支持 IO 或编码器在伺服设备之后；
- 目前暂不支持集成式电柜进行添加、删除 IO 功能；
- 网络拓扑图 7-1 中设备名称可显示的范围为：

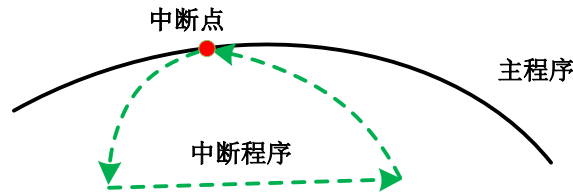
IO 设备识别范围：coupling、EC4-1616、XB6-EC0002E、EC4-A04VE、TunderIO、EC-P20DE；

伺服设备识别范围：Pronet、EDSD、ED3L、ED3S；

其余设备类型统一识别为 Slave。

7.33 中断功能

- 该功能提供一种机制，可以在执行主程序的过程中，被中断信号打断转去执行中断程序，当中断程序执行结束后可以回到主程序的中断处继续执行。



- 通过指令的形式来配置中断机制相关的参数，包括中断后需执行的程序、中断的触发方式、中断需要让主程序做的操作等。
- 目前系统支持的触发方式有三种：定时触发、DI 触发以及 SIMDI 触发。
- 对应的指令存放在中断指令集中，具体指令定义见 6.8 中断指令，这些指令包括 IEnter、IExit、MovToIPos、IConnect、IDelete、ITimer、ISignalDI、ExitCycle。

说明

- 如果当前程序正在执行 MovH、MovArch 指令时，系统不响应中断触发信号。
- 在中断程序中只允许部分指令执行，依据指令列表说明如下：
 - 运动指令：除 MovH、MovArch 以外的运动指令（执行 IEnter 指令后才允许执行运动指令）
 - 控制指令：全部支持
 - 等待指令：全部支持
 - IO 指令：全部支持
 - 设置指令：全部支持
 - 时钟指令：全部支持
 - 区域指令：全部支持
 - 中断指令：IEnter、IExit、MovToIPos、ExitCycle

中断具有如下几种特性：

- 目前的中断信号检测有两种方式：检测 DI 的边沿信号、根据设定的时间定时触发。
- 同时仅允许一个中断程序在执行，其他的中断程序将按照信号的触发顺序存储在队列中，待本次中断程序执行结束后依次执行队列中的中断程序。
- IEnter 指令表示打断主程序的解析与运动，如果中断程序中没有示教该指令时，中断程序的执行效果将类似于后台 run 的执行效果。
- 中断程序与主程序中均支持 call 指令，但中断程序仅支持 call 本工程下的其他程序，不支持全局程序。

应用场景

中断功能的三种应用场景如下所示：

- ◆ 场景 1：中断触发后立即停止，待中断程序执行完并走到中断点后继续执行。
- ◆ 场景 2：中断触发后立即停止，待中断程序执行完退出中断至程序首行。
- ◆ 场景 3：中断触发后进入后台并行执行操作（即不下发中断命令以及让主程序进入休眠的命令），待中断程序执行完后自动退出。

场景 1 示例：

Prog1 为中断触发后立即停止，待中断程序执行完并走到中断点后继续执行的程序示例。如下所示：

1. 在 main 程序的运行过程中，当 DI15 被设置成有效上升沿信号时，系统会中断 main 程序的运行并切换到程序 Prog1 中运行，同时工程界面中 Prog1 程序的状态显示为中断执行状态，程序编辑界面以及变量界面均会显示为 Prog1 中的对应数据。
2. 等到 Prog1 程序执行结束后，系统再恢复到 main 程序的执行以及对 main 程序相关数据的显示。

```

main()
Start:

IDelete(INTERRUPT1) //复位中断标识（以变量的形式呈现）相关的所有数据

IConnect(INTERRUPT1, Prog1) //将中断标识与程序进行绑定

ISignalDI(INTERRUPT1, DI15, "riseEdge",1) //设置中断标识对应的检测方式为 DI 上升沿信号有效的方式。

GetCurCPOS (P0)

IF P0 ~= PHome Then //判断机器人是否在零位，不在零位则走到零位
MovL(PHome)
ENDIF

tt:
MovL(P1)
MovC(P2,P3)
...
MovL(P10)
Goto tt
End;

Prog1()
Start:

IEnter() //暂停当前的运动，并让主解析进入休眠状态，并存储停下来的点位信息，包括当前工具、用户坐标系、负载等信息。

MovL(P1)
...

MovToIPos ("MovJ",V200) //运动到之前缓存的暂停点位位置处，并恢复成之前的工具、用户坐标系、负载惯量

IExit() //退出中断，恢复主程序的运行，从暂停的运动指令上重新解析指令。

End;

```


场景 2 示例：

Prog2 为中断触发后立即停止，待中断程序执行完退出中断至程序首行示例。如下所示：

1. 在 main 程序的运行过程中，当 DI16 被设置成有效上升沿信号时，系统会中断 main 程序的运行并切换到程序 Prog2 中运行，同时工程界面中 Prog2 程序的状态显示为中断执行状态，程序编辑界面以及变量界面均会显示为 Prog2 中的对应数据。
2. 等到 Prog2 程序执行到 ExitCycle 后，系统会自动转入停止状态。
3. 系统判断 main 程序中是否有 CycleLabel 指令，来复位 main 的 PC 指针到指定行以及恢复对 main 程序相关数据的显示。(若有 CycleLabel 指令，跳转至 CycleLabel 处执行；若不存在则跳转到程序首行)

```
main()
Start:
IDelete(INTERRUPT2)//复位中断标识（以变量的形式呈现）相关的所有数据
IConnect(INTERRUPT2, Prog2) //将中断标识与程序进行绑定
ISignalDI(INTERRUPT2, DI16, "riseEdge",1) //设置中断标识对应的检测方式为 DI 上升沿信号有效的方式。
GetCurCPOS (P0)
IF P0 ~= PHome Then //判断机器人是否在零位，不在零位则走到零位
MovL(PHome)
ENDIF
tt:
MovL(P1)
MovC(P2,P3)
...
MovL(P10)
Goto tt
End;

Prog2()
Start:
IEnter () //暂停当前的运动，并让主解析进入休眠状态，并存储停下来的点位信息，包括当前工具、用户坐标、负载等信息。
MovL(P1)
...
ExitCycle() //复位主程序的 PC 指针，让其从首行处开始执行。
End
```

场景 3 示例：

Prog3 为中断触发后进入后台并行执行操作（即不下发中断命令以及让主程序进入休眠的命令），待中断程序执行完后自动退出的程序示例。如下所示：

1. 在 main 程序的运行过程中，每间隔 2000ms 系统会调用一次 Prog3，但 Prog3 程序只会在后台并行执行，不会打断 main 程序的运行与相关数据显示，同时工程界面中 Prog3 程序的状态显示为中断并行状态。
2. 等到 Prog3 执行结束后并行执行结束。

```
main()
Start:
IDelete(INTERRUPT3)//复位中断标识（以变量的形式呈现）相关的所有数据
IConnect(INTERRUPT3, Prog3) //将中断标识与程序进行绑定
ITimer(INTERRUPT3, 2000) //设置中断标识 INTERRUPT3 检测方式为定时器中断，每 2000ms 触发一次中断
GetCurCPOS (P0)
IF P0 ~= PHome Then //判断机器人是否在零位，不在零位则走到零位
MovL(PHome)
ENDIF
tt:
MovL(P1)
MovC(P2,P3)
...
MovL(P10)
Goto tt
End;

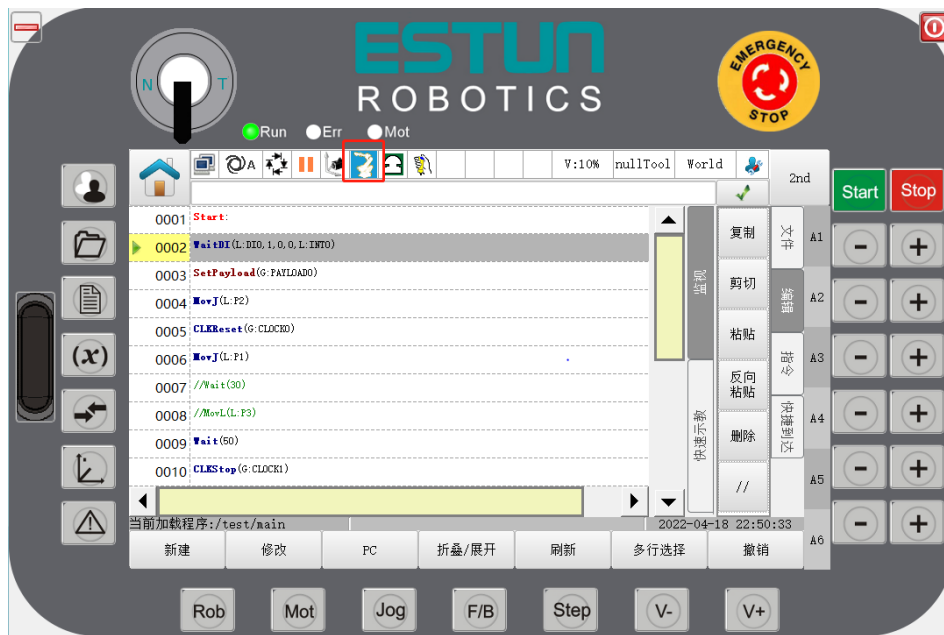
Prog3()
Start:
SetDO(DO1,0) //无 IEnter 指令时，中断程序会像 run 程序一样在后台执行，并不会发送中断命令。
INT0 = INT0 + 1
...
SetDO(DO1,1)
INT0 = 0
End;
```

7.34 自动掉励磁

机器人长时间待机时，控制器能自动掉励磁，帮助客户节约工厂能源，同时还能降低对电控柜里部分元器件寿命的影响。

因机器人“点头”现象，即掉励磁后机器人会下坠一点，重新上励磁后，机器人会采用暂停再启动的逻辑，先回到下坠前的位置，再继续运行。

- 机器人停止/暂停后，或者 WaitDI/WaitTime 等，控制器开始计时，当超过设定值（该值通过参数配置）时，控制器自发掉励磁命令，示教器界面使用不同颜色来区分此情况下的掉励磁与常规操作的掉励磁状态；



- 当再次收到启动命令，或接收到 DI 信号，或等待时间到时，控制器先发自动上励磁命令，紧接着走指令运行。

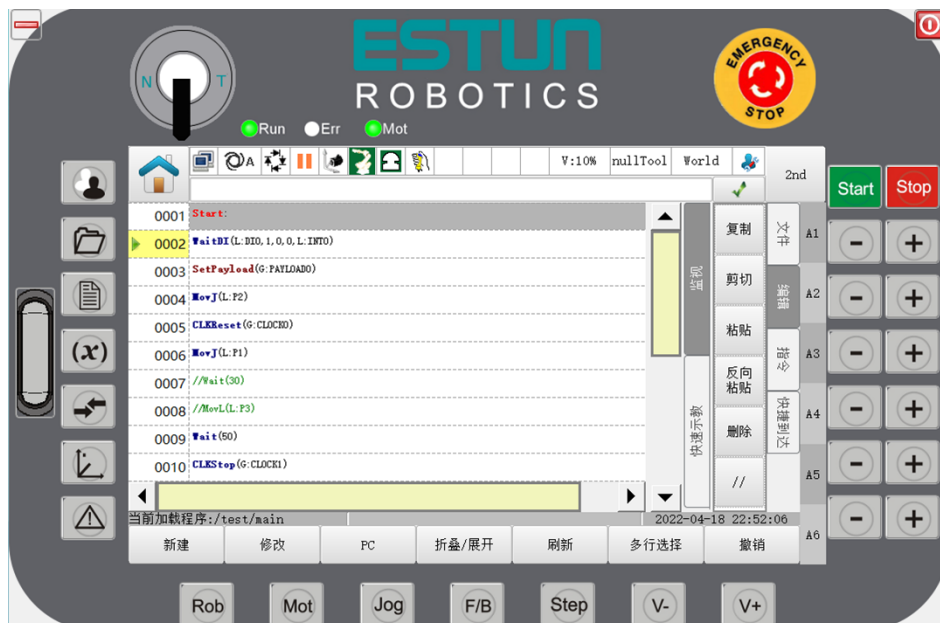
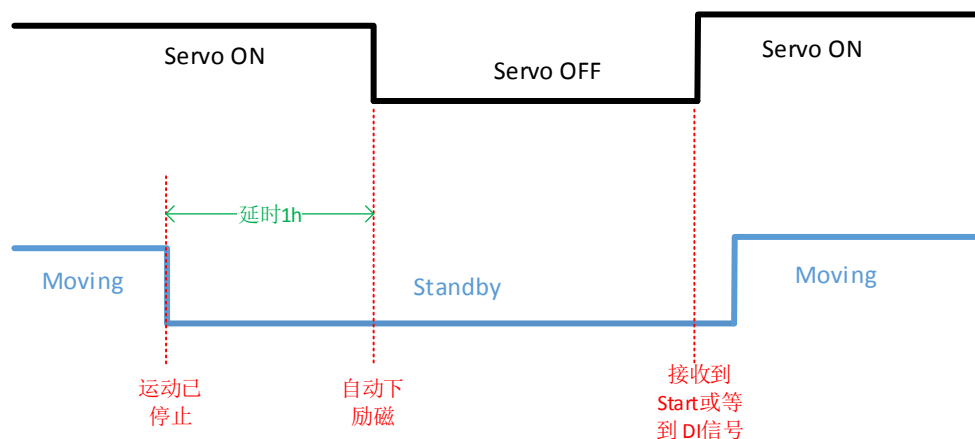


图7-120 自动掉励磁时序图



7.35 负载在线识别

机器人负载在线辨识功能，该功能是为在线获取机器人法兰端安装的负载的动力学信息（包括负载质量、质心、惯性张量），辨识所得的负载动力学信息用于机器人动力学模型计算，可改善机器人运行性能，以及改善因负载信息不匹配引起的关节超力矩报警、碰撞检测误报警、机器人运动过程中抖动等现象。

功能描述

- 数据采集开始指令 StartMdlLog();
- 数据采集结束指令 StopMdlLog();
- 负载辨识计算指令即 CalcDynMdl()。

使用机器人负载在线辨识功能时，请根据示例步骤进行操作：

应用场景示例

数据采集

步骤 1 数据采集。点击程序下方“新建”找到“负载在线识别指令> StartMdLog”选择对应数据保存类型。设置好对应的负载动力学信息，设置完成后选择“StopMdlLog”指令。

(1) 采集“Robot”类型数据用例：

```
/****未安装负载时采集，轨迹点位可调整，选选项****/
StartMdlLog{dataType="Robot"}
/*运行速度，建议包括慢速和高速*/
SetOverride{V=10}
CALL PayloadPath
.
.
.
SetOverride{V=100}
CALL PayloadPath
StopMdlLog{}
```

(2) 采集“WithLoad”类型数据用例：

```
/****安装有负载时采集，轨迹点位可调整，必须执行****/
StartMdlLog{dataType="WithLoad"}
/*运行速度，建议包括慢速和高速*/
SetOverride{V=10}
CALL PayloadPath
.
.
.
SetOverride{V=100}
CALL PayloadPath
StopMdlLog{}
```

(3) 采集“ValidData”类型数据用例：

```
/****用于模型验证时采集，轨迹可自定义，选选项****/
StartMdlLog{dataType="ValidData"}
/*轨迹自定义，MovJ\MovL\MovC 等运动指令*/
.
MovL(P=t_1.P1)
MovJ(P=t_1.P2)
MovC(A=t_1.P3,P=t_1.P4)
.
StopMdlLog{}
```

步骤 2 获取到负载辨识所用轨迹程序如下所示。

```
/* PayloadPath 程序执行段，负载辨识轨迹段*/
MovJ(P=t_1.P0)
```

```

MovJ{P=t_1.P1}
MovJ{P=t_1.P2}
MovJ{P=t_1.P3}
MovJ{P=t_1.P4}
MovJ{P=t_1.P5}
MovJ{P=t_1.P6}
MovJ{P=t_1.P7}
MovJ{P=t_1.P8}
MovJ{P=t_1.P9}
MovJ{P=t_1.P8}
MovJ{P=t_1.P7}
MovJ{P=t_1.P6}
MovJ{P=t_1.P5}
MovJ{P=t_1.P4}
MovJ{P=t_1.P3}
MovJ{P=t_1.P2}
MovJ{P=t_1.P1}
MovJ{P=t_1.P0}

```

步骤 3 负载辨识轨迹的点位数据如下，其中 J1 和 J2 关节在数据采集过程中不需要运动，可保持某一固定角度位置即可，在运动过程中负载与机器人本体或工作空间物体有干涉时，可根据实际情况修改点位。

关节 序号	J3（角度）	J4	J5	J6
P0	0	0	0	0
P1	-5.0	10.0	10.0	180.0
P2	5.0	95.0	95.0	170.0
P3	-5.0	85.0	85.0	-10.0
P4	5.0	-5.0	-95.0	10.0
P5	-5.0	5.0	-85.0	-180.0
P6	5.0	-5.0	95.0	-170.0
P7	-5.0	-95.0	105.0	-180.0
P8	5.0	-85.0	-95.0	-170.0
P9	-5.0	165.0	-85.0	-180.0

负载识别计算

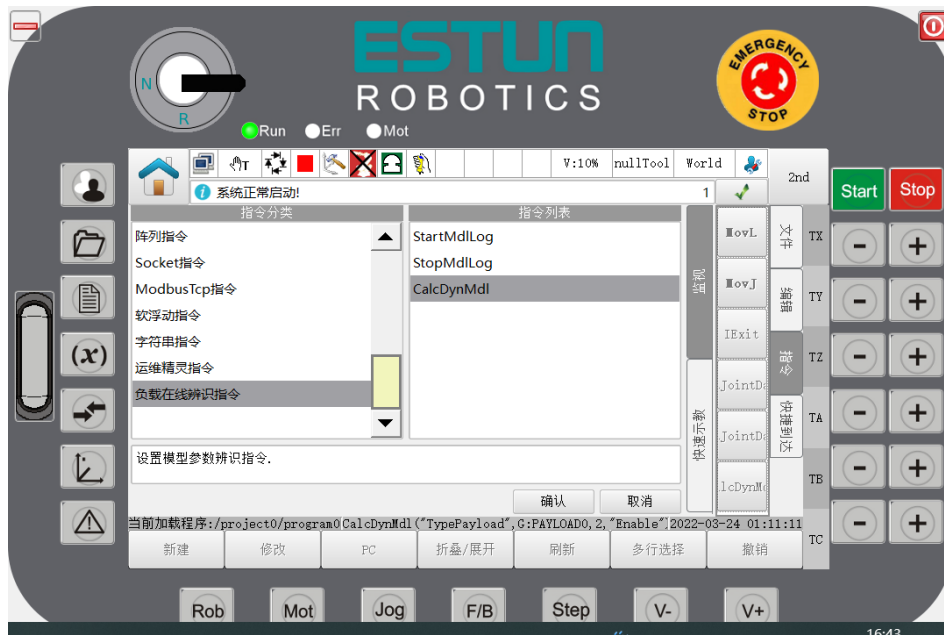
编辑负载辨识计算指令 CalcDynMdl()，指定辨识的模型类型、工件负载变量、数据间隔长度、是否模型验证等参数。

```

/****负载辨识计算****/
CalcDynMdl{mdlType="TypePayload",PayLoad=t g.idenPayload,dataStepPer=1,validEnable="Disable"}
/*负载辨识结束，结果将存于用户指定的负载/工具变量中*/。

```

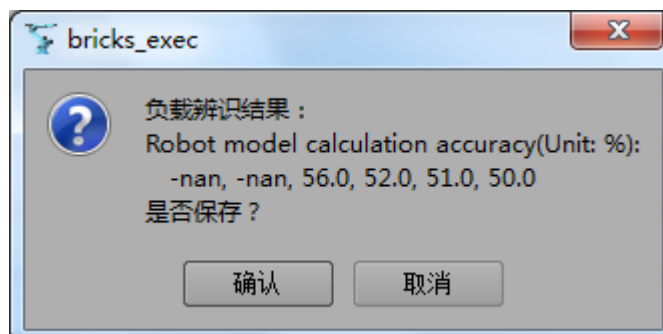
步骤 1 在点击程序下方“新建”按钮，选择“负载在线辨识指令>CalDynMdl”指令，点击“确认”按钮。



步骤 2 参数设置，“辨识的模型类型”设置为“TypePayload/TypeTool”，“工件负载变量”设置为“G:PAYLOAD1”，“数据间隔长度”设置为“1”，“是否模型验证”设置为“Enable”后，点击“确认”按钮。



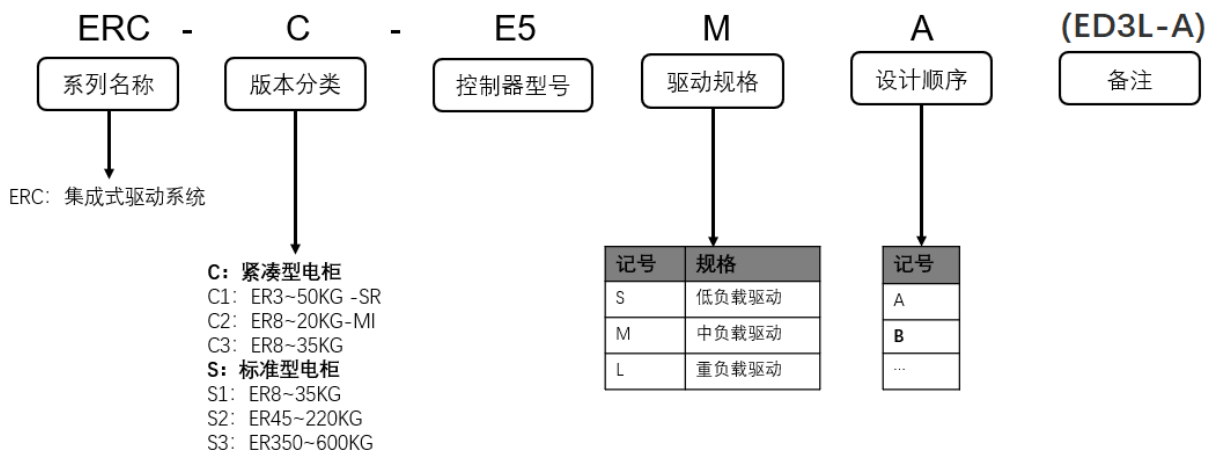
步骤 3 在线辨识指令执行过程时间较长（一般用时为 30~60min），待辨识计算结束，将弹窗显示各轴模型验证评分（由于 J1 和 J2 轴没有指定运动角度，因此评分一般为 nan），如下图所示。



第 8 章 FAQ

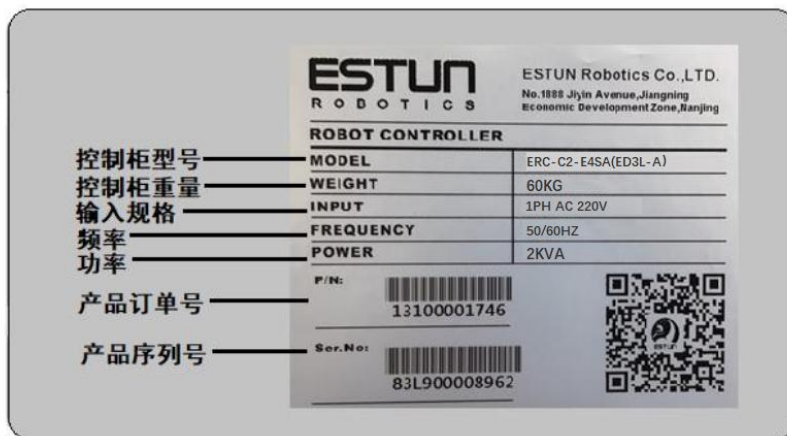
8.1 如何根据铭牌信息查找对应控制柜说明书

控制柜型号说明如下图所示：



解决方法

步骤 1 找到控制柜的铭牌，以 C2E 控制柜手册为例。



步骤 2 根据铭牌信息可知是 C2 系列控制柜，若想了解控制柜详细信息，可查找《ESTUN 机器人 C2E 系列控制柜使用说明书》。

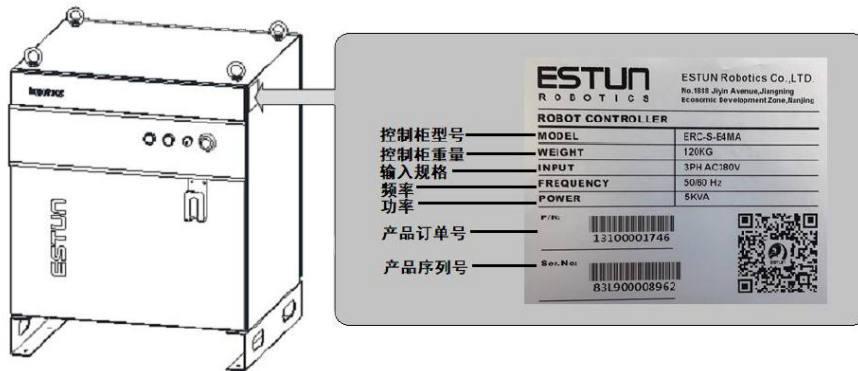
步骤 3 若有疑问或无法判准确找到控制柜对应的说明书，请联系埃斯顿工程师。

8.2 埃斯顿机器人的供电需求

埃斯顿机器人的接入电源需要满足铭牌上的输入电压和功率要求。如果手册中有要求接入地线或零线，则不可省略。

解决方法

步骤 1 找到控制柜上的铭牌，接入的主电源需要满足铭牌标定的输入电压和功率。如下图所示：

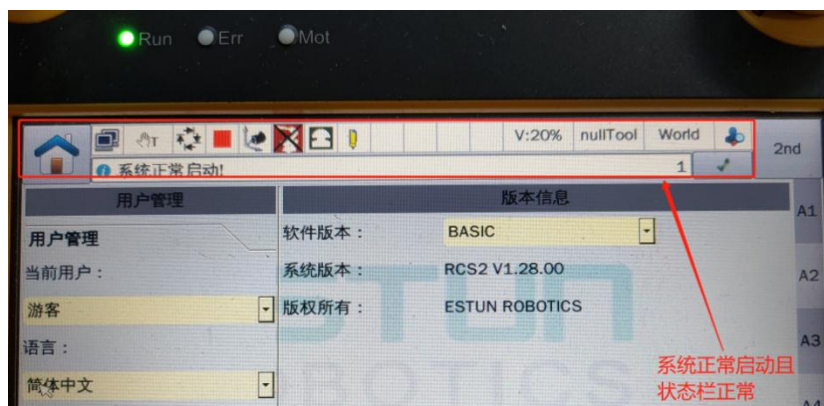


步骤 2 在对应型号的控制柜说明书中，若要求有地线或零线，则不可省略。

8.3 如何处理机器人开机后显示急停处于按下状态

解决方法：

步骤 1 检查示教器是否正常启动。



步骤 2 检查示教器、电柜以及外部急停按钮是否被按下。



步骤 3 检查驱动器是否有报警。

步骤 4 若电柜面板有“启动”按钮，开机时需按亮。



8.4 机器人如何进行周期性保养

不同型号的机器人保养要求和周期不同。具体保养周期和要求可查阅对应机型的本体说明书。

解决方法

以 ER20-1780 机器为例。

步骤 1 登录[埃斯顿官网](#)，选择“服务与支持 > 资料下载”。



步骤 2 输入搜索“本体使用说明”，找到《ESTUN 机器人 通用小负载系列本体使用说明书》文档，查看“维护与维修”章节，获取保养相关说明。

搜索（本体使用说明书）结果

 ESTUN机器人 通用大负载（高防护）系列本体使用说明书 M-1103CN-01

 ESTUN机器人 通用小负载系列本体使用说明书M-0106CN-01

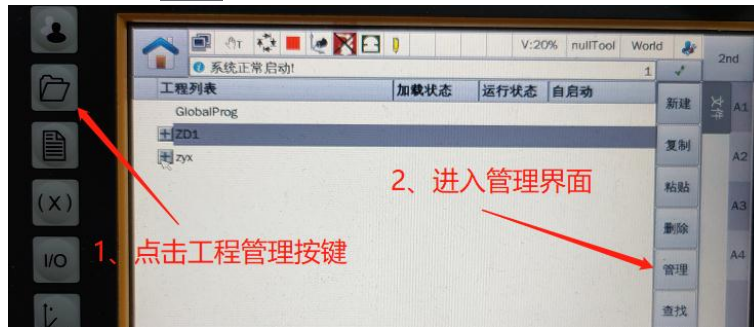
8.5 如何备份机器人示教器程序

可以通过示教器 USB 接口，完成示教器程序的备份。

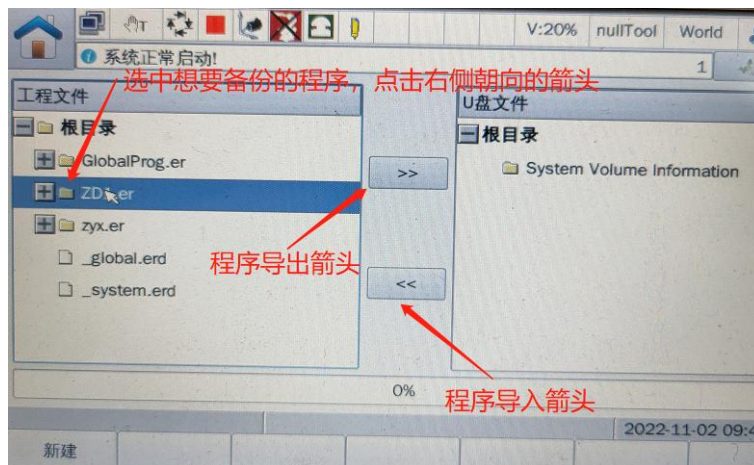
解决方法

步骤 1 准备一个 FAT32 格式的 U 盘，插到示教器的 USB 接口。

步骤 2 按示教器上  按键或点击主页中“工程管理”，进入“工程管理”界面。



步骤 3 选择需要导入或导出的程序，使用对应的箭头指示进行导入导出。



步骤 4 导入导出程序，提示完成后，等待 30 秒再拔出 U 盘。

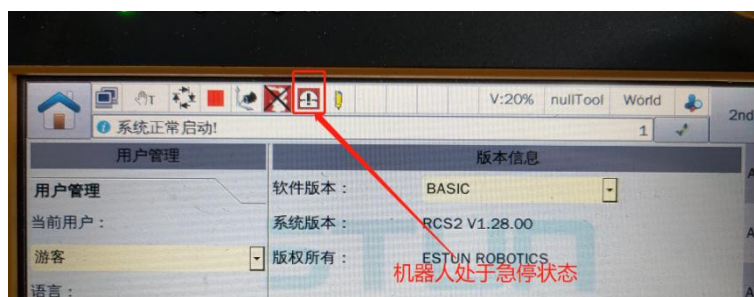
8.6 触发机器人急停按钮后如何恢复

在机器人操作过程中，紧急按钮被触发后，可以通过旋转触发的急停按钮解除。

急停按钮的位置：示教器右上方或控制柜上。

解决方法

如下图所示，机器人处于急停状态。



步骤 1 先找到触发的急停按钮，并解除。



步骤 2 若电柜面板有“启动”按钮，需按下。



步骤 3 清除示教器的报警提示。

步骤 4 检查示教器状态正常，根据实际情况运行即可。



8.7 使用时发现示教器不能进行程序编辑和修改怎么办

示教器默认登录的角色是“操作人员”，此用户权限下无法编辑程序，需要切换至“管理人员”。

解决方法

步骤 1 按  或在示教器主页面点击“用户管理”，进入“用户管理”界面。



步骤 2 在“当前用户”点击下拉箭头，选择“管理人员”角色，输入正确的密码。验证成功后重新加载需要编辑的程序，即可对程序进行编辑修改操作。



8.8 埃斯顿机器人目前支持的通讯有哪些

目前埃斯顿机器人支持的通讯协议主要有：ModBusTcp、TCPIP、EtherNetIP、Profinet、MC、Fins 等

8.9 如何解决正常使用中机器人运行的点位发生了偏移

解决方法

- 步骤 1 检查机器人零点是否正常。
- 步骤 2 检查夹具是否形变。
- 步骤 3 检查线体是否移位。
- 步骤 4 可单独建立示教基准点，复现点位偏后进行对比。

8.10 如何判定机器人开机后无法运动

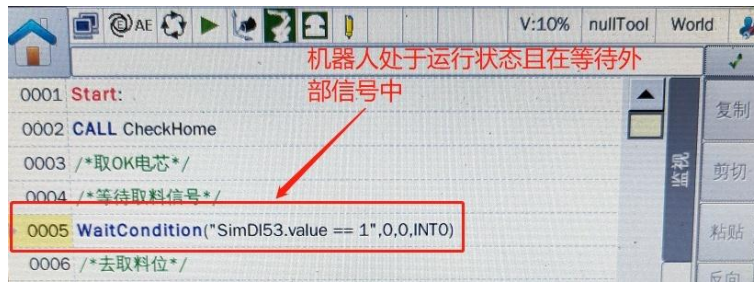
机器人开机之后不运动，可能是程序不在运行状态或程序中的判断条件不满足所致。

解决方法

- 步骤 1 通过示教器状态栏，检查机器人是否处于运行状态。



步骤 2 点到程序界面，依据程序指针，检查是否某句指令条件不满足导致。



8.11 如何解决机器人开机后示教器无法进入系统

系统异常和示教器设置的 IP 与控制器不匹配，都有可能造成示教器加载进度条卡在某个状态不动，无法进入系统。

解决方法

场景 1：若示教器进度条停留在 10%或 13%，此时示教器系统异常，联系埃斯顿技术人员处理。

场景 2：若示教器进度条停留 20%，此时示教器设置的 IP 与控制器不匹配。请按如下操作。

步骤 1 通过示教器左侧实体按键组合进入示教器 IP 设置界面。如下图所示，依次按下 1/7/6/5/4 按键。



步骤 2 设置正确 IP 后，保存并重启。

场景 3：若示教器进度条停留 80%、85%、90%，此时控制器启动异常，联系埃斯顿技术人员处理。

第 9 章 附录

9.1 示教编程器权限分配表

功能		√ 有权限 × 无权限 ★ 二次密码确认											
		厂家人员			管理人员			编程人员			操作人员		
		手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
面板按键	界面切换	√	√	×	√	√	×	√	√	×	√	√	×
	点动按键	√	×	×	√	×	×	√	×	×	√	×	×
	start/stop	√	√	×	√	√	×	√	√	×	√	√	×
	模式开关	√	√	√	√	√	√	√	√	√	√	√	√
	Rob 快捷启动项	√	√	×	√	√	×	√	√	×	√	√	×
	Mot 上电	√	√	×	√	√	×	√	√	×	√	√	×
	Jog 切换坐标	√	×	×	√	×	×	√	×	×	√	×	×
	F/B	√	×	×	√	×	×	√	×	×	√	×	×
	Step 单步/循环	√	√	×	√	√	×	√	√	×	√	√	×
	全局速度按键	√	√	√	√	√	√	√	√	√	√	√	√
主界面	清除警报	√	√	√	√	√	√	√	√	√	√	√	√
	切换轴(2nd)	√	√	√	√	√	√	√	√	√	√	√	√
	设置全局速度	√	√	√	√	√	√	√	√	√	√	√	√
	界面切换	用户登录	√	√	√	√	√	√	√	√	√	√	√
		工程管理	√	√	√	√	√	√	√	√	√	√	√
		程序编辑	√	√	√	√	√	√	√	√	√	√	√
		程序数据	√	√	√	√	√	√	√	√	√	√	√
		IO 检测	√	√	√	√	√	√	√	√	√	√	√
		点动管理	√	√	√	√	√	√	√	√	√	√	√
		日志管理	√	√	√	√	√	√	√	√	√	√	√
		通用设置	√	√	√	√	√	√	√	√	√	√	√
		高级设置	√	√	√	√	√	√	√	√	√	√	√
		用户应用	√	√	√	√	√	√	√	√	√	√	√
用户登录	用户切换	★	★	★	★	★	★	★	★	★	★	★	★
	密码管理	√	√	×	√	√	×	×	×	×	×	×	×

功能			√ 有权限 × 无权限 ★ 二次密码确认											
			厂家人员			管理人员			编程人员			操作人员		
			手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
	语言切换		√	√	×	√	√	×	×	×	×	×	×	×
工程管理	查找		√	√	√	√	√	√	√	√	√	√	√	√
	刷新		√	√	√	√	√	√	√	√	√	√	√	√
	新建		√	√	×	√	√	×	√	√	×	×	×	×
	复制		√	√	×	√	√	×	√	√	×	×	×	×
	粘贴		√	√	×	√	√	×	√	√	×	×	×	×
	删除		√	√	×	√	√	×	√	√	×	×	×	×
	管理		√	√	×	√	√	×	√	√	×	×	×	×
	加载		√	√	×	√	√	×	√	√	×	√	√	×
	注销		√	√	×	√	√	×	√	√	×	√	√	×
	打开		√	√	×	√	√	×	√	√	×	√	√	×
	自启动		√	√	×	√	√	×	√	√	×	×	×	×
	重命名		√	√	×	√	√	×	√	√	×	×	×	×
	写保护		√	√	×	√	√	×	×	×	×	×	×	×
程序编辑	修改		√	√	×	√	√	×	√	√	×	√	√	×
	折叠/展开		√	√	×	√	√	×	√	√	×	√	√	×
	刷新		√	√	√	√	√	√	√	√	√	√	√	√
	多行选择		√	√	×	√	√	×	√	√	×	√	√	×
	新建指令		√	√	×	√	√	×	√	√	×	×	×	×
	设置 PC		√	√	×	√	√	×	√	√	×	√	√	×
	撤销		√	√	×	√	√	×	√	√	×	×	×	×
	文件-新建		√	√	×	√	√	×	√	√	×	×	×	×
	编辑	复制	√	√	×	√	√	×	√	√	×	×	×	×
		剪切	√	√	×	√	√	×	√	√	×	×	×	×
		粘贴	√	√	×	√	√	×	√	√	×	×	×	×
		反向粘贴	√	√	×	√	√	×	√	√	×	×	×	×
		删除	√	√	×	√	√	×	√	√	×	×	×	×
		注释	√	√	×	√	√	×	√	√	×	×	×	×
	快捷新建指令		√	√	×	√	√	×	√	√	×	×	×	×
	快捷到达	顶部/底部	√	√	×	√	√	×	√	√	×	√	√	×
		选择行/指定行	√	√	×	√	√	×	√	√	×	√	√	×

功能			√ 有权限 × 无权限 ★ 二次密码确认											
			厂家人员			管理人员			编程人员			操作人员		
			手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
		位置点	√	×	×	√	×	×	√	×	×	√	×	×
		ToPC	√	√	×	√	√	×	√	√	×	√	√	×
	快速示教	插入	√	√	×	√	√	×	√	√	×	×	×	×
		示教	√	×	×	√	×	×	√	×	×	×	×	×
		J/L 快捷切换	√	√	×	√	√	×	√	√	×	×	×	×
程序数据	刷新		√	√	√	√	√	√	√	√	√	√	√	√
	查找		√	√	√	√	√	√	√	√	√	√	√	√
	新建		√	√	×	√	√	×	√	√	×	×	×	×
	删除		√	√	×	√	√	×	√	√	×	×	×	×
	修改		√	√	×	√	√	×	√	√	×	×	×	×
	重命名		√	√	×	√	√	×	√	√	×	×	×	×
	更新全局域		√	√	×	√	√	×	√	√	×	×	×	×
	快捷指令	新建程序	√	√	×	√	√	×	√	√	×	×	×	×
		新建变量	√	√	×	√	√	×	√	√	×	×	×	×
IO 检测	物理 IO	查看	√	√	√	√	√	√	√	√	√	√	√	√
		物理数字 IO	√	√	√	√	√	√	√	√	×	×	×	×
		物理模拟 IO	√	√	√	√	√	√	√	√	×	×	×	×
	虚拟 IO	查看	√	√	√	√	√	√	√	√	√	√	√	√
		虚拟数字 IO	√	√	√	√	√	√	√	√	×	×	×	×
		虚拟模拟 IO	√	√	√	√	√	√	√	√	×	×	×	×
	系统 IO	查看	★	★	★	★	★	★	×	×	×	×	×	×
		修改端口号	√	√	×	×	×	×	×	×	×	×	×	×
		修改端口值	√	√	×	×	×	×	×	×	×	×	×	×
点动管理	位置管理	回零	★	×	×	★	×	×	★	×	×	×	×	×
		查看	√	√	√	√	√	√	√	√	√	√	√	√
		关节/世界/用户坐标切换	√	√	√	√	√	√	√	√	√	√	√	√
		力矩	√	√	√	√	√	√	√	√	√	√	√	√
		单圈值	√	√	√	√	√	√	√	√	√	√	√	√
		设置单圈值	√	×	×	√	×	×	×	×	×	×	×	×
		寸动设置	√	×	×	√	×	×	√	×	×	×	×	×
		切换工具	√	×	×	√	×	×	√	×	×	√	×	×

功能			√ 有权限 × 无权限 ★ 二次密码确认											
			厂家人员			管理人员			编程人员			操作人员		
			手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
		切换参考坐标系	✓	×	×	✓	×	×	✓	×	×	✓	×	×
		切换点动模式	✓	×	×	✓	×	×	✓	×	×	✓	×	×
		切换点动坐标系	✓	×	×	✓	×	×	✓	×	×	✓	×	×
	工具对正	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		选择对正类型	✓	×	×	✓	×	×	✓	×	×	×	×	×
		激活	✓	×	×	✓	×	×	✓	×	×	×	×	×
	回原位	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		修改目标位置	✓	×	×	✓	×	×	✓	×	×	×	×	×
		修改点数	✓	×	×	✓	×	×	✓	×	×	×	×	×
		修改序号	✓	×	×	✓	×	×	✓	×	×	×	×	×
		示教	✓	×	×	✓	×	×	✓	×	×	×	×	×
		GO	✓	×	×	✓	×	×	✓	×	×	×	×	×
日志管理	报警	当前报警查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		历史报警查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		清空历史报警	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		获取历史报警	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		清除当前报警	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	日志	清空缓存	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		更新	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		导出	✓	✓	×	✓	✓	×	×	×	×	×	×	×
通用设置	基本设置	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		查看机器人类型	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		查看工件负载重量	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		查看工具负载重量	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		查看远程速度	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		机器人安装类型	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×
		API 模式	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×
		系统时间设置	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		程序设置 (all)	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×
	碰撞检测	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		使能开关	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×

功能			√ 有权限 × 无权限 ★ 二次密码确认											
			厂家人员			管理人员			编程人员			操作人员		
			手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
	振动抑制	轴 1-6	√	√	×	√	√	×	√	√	×	×	×	×
		查看	√	√	√	√	√	√	√	√	√	√	√	√
		使能开关	√	√	×	√	√	×	√	√	×	×	×	×
		抑制范围调节(轴 1-6)	√	√	×	√	√	×	√	√	×	×	×	×
	视觉设置	查看	√	√	√	√	√	√	√	√	√	√	√	√
		相机设备类型	√	√	×	√	√	×	√	√	×	×	×	×
		摄像头 ip	√	√	×	√	√	×	√	√	×	×	×	×
		摄像头端口号	√	√	×	√	√	×	√	√	×	×	×	×
		视觉类型	√	√	×	√	√	×	√	√	×	×	×	×
	伺服管理	查看	√	√	√	√	√	√	√	√	√	√	√	√
		复位(清除单个伺服报警)	√	√	√	√	√	√	√	√	√	×	×	×
		清除伺服报警(轴 1-6)	√	√	×	√	√	×	√	√	×	×	×	×
		设置 Pn 参数(轴 1-6)	√	√	×	√	√	×	×	×	×	×	×	×
		版本信息查看	√	√	×	√	√	×	√	√	×	×	×	×
		复位 FN	★	★	×	★	★	×	★	★	×	×	×	×
		导入	★	★	×	★	★	×	×	×	×	×	×	×
		备份参数设置	★	★	×	★	★	×	×	×	×	×	×	×
		恢复参数设置	★	★	×	★	★	×	×	×	×	×	×	×
	内核诊断	查看	√	√	√	√	√	√	√	√	√	√	√	√
		状态监控	√	√	√	√	√	√	√	√	√	√	√	√
		功能输入	√	√	√	√	√	√	×	×	×	×	×	×
		参数输入	√	√	√	√	√	√	×	×	×	×	×	×
	系统诊断	查看	√	√	√	√	√	√	√	√	√	√	√	√
		更新日志	√	√	√	√	√	√	√	√	√	×	×	×
		查看日志	√	√	√	√	√	√	√	√	√	×	×	×
		导出日志	√	√	√	√	√	√	√	√	√	×	×	×
	从站配置	查看	√	√	√	√	√	√	√	√	√	√	√	√
		从站类型	√	√	√	√	√	√	√	√	√	×	×	×
		从站型号	√	√	√	√	√	√	√	√	√	×	×	×
高级设置	本地设置	查看	√	√	√	√	√	√	√	√	√	√	√	√
		示教器开关	★	★	×	★	★	×	★	★	×	×	×	×

功能			√ 有权限 × 无权限 ★ 二次密码确认											
			厂家人员			管理人员			编程人员			操作人员		
			手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
		控制系统开关	★	★	×	★	★	×	★	★	×	×	×	×
		Rob 按键重定义	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		机器人命名	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		设置屏幕亮度	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		示教器超时未操作	✓	✓	×	✓	✓	×	×	×	×	×	×	×
	网络设置	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		外部通讯(lan3/lan4)	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×
		内部通讯	✓	✓	×	×	×	×	×	×	×	×	×	×
	锁机设置	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		解锁	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		设置	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		清除	✓	✓	×	✓	✓	×	×	×	×	×	×	×
	控制器升级	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		系统升级	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		备份导出	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		更新	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		还原	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		后台自动备份	✓	✓	×	✓	✓	×	×	×	×	×	×	×
	硬件测试		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TP 更新	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		TP 更新使能	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		选择示教器程序	✓	✓	×	✓	✓	×	×	×	×	×	×	×
		重启	★	★	×	★	★	×	×	×	×	×	×	×
		选择文件	✓	✓	×	✓	✓	×	×	×	×	×	×	×
	软件授权		✓	✓	×	✓	✓	×	×	×	×	×	×	×
用户应用	多工位预约	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		多工位预约状态查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		多工位预约使能	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×
	九点标定	查看	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		相机选择	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×
		点位选择	✓	✓	×	✓	✓	×	✓	✓	×	×	×	×

功能			√ 有限权限 × 无限权限 ★ 二次密码确认											
			厂家人员			管理人员			编程人员			操作人员		
			手动	自动	远程	手动	自动	远程	手动	自动	远程	手动	自动	远程
		九点标定使能	√	√	×	√	√	×	√	√	×	×	×	×
		帮助	√	√	×	√	√	×	√	√	×	×	×	×
	变位机		√	√	√	√	√	√	√	√	√	√	√	√
	多边形区域		√	√	√	√	√	√	√	√	√	√	√	√
	传送带跟随	查看	√	√	√	√	√	√	√	√	√	√	√	√
		传送带使能	√	√	√	√	√	√	√	√	√	×	×	×
		脉冲当量标定	√	√	√	√	√	√	√	√	√	×	×	×
		传送带参数	√	√	√	√	√	√	√	√	√	×	×	×
		跟随参数	√	√	√	√	√	√	√	√	√	×	×	×
		触发参数	√	√	√	√	√	√	√	√	√	×	×	×
		视觉参数	√	√	√	√	√	√	√	√	√	×	×	×
		虚拟传送带	√	√	√	√	√	√	√	√	√	×	×	×
	保养信息	查看	√	√	√	√	√	√	√	√	√	√	√	√
		刷新	√	√	×	√	√	×	√	√	×	×	×	×
		提示日志	√	√	×	√	√	×	√	√	×	×	×	×
		重置	√	√	×	√	√	×	√	√	×	×	×	×
	跑合测试	查看	√	√	√	√	√	√	√	√	√	√	√	√
		修改测试时长	√	√	×	√	√	×	√	√	×	×	×	×
		重置	√	√	×	√	√	×	√	√	×	×	×	×
		保存	√	√	×	√	√	×	√	√	×	×	×	×
	scara 快速标定		√	×	×	√	×	×	×	×	×	×	×	×

9.2 ModbusTCP 控制接口数据表

	本地地址	寄存器地址	定义	说明	注释
Rob Send	MBDataBuffer[0]	40001	预留		
	MBDataBuffer[1]	40002		全局速度	
	MBDataBuffer[2]	40003		读写标志位应答	
	MBDataBuffer[3]	40004	Rob 状态信息	• bit0: 手动操作模式 • bit1: 自动操作模式 • bit2: 远程操作模式 • bit3: 使能状态 • bit4: 运行状态 • bit5: 错误状态 • bit6: 程序运行状态 • bit7: 机器人正在动作	
	MBDataBuffer[4]	40005	当前加载工程名	20 个字节	例如加载的工程文件名为:estun.test,则各寄存器的值如下: [4]0x6573, [5]0x7475, [6]0x6E2E, [7]0x6D61, [8]0x696E,
	MBDataBuffer[5]	40006			
	MBDataBuffer[6]	40007			
	MBDataBuffer[7]	40008			
	MBDataBuffer[8]	40009			
	MBDataBuffer[9]	40010			
	MBDataBuffer[10]	40011			
	MBDataBuffer[11]	40012			
	MBDataBuffer[12]	40013			
	MBDataBuffer[13]	40014			
	MBDataBuffer[14]	40015	SimDout[1-16]	DO 1-16	
	MBDataBuffer[15]	40016	SimDout[17-32]	DO 17-32	
	MBDataBuffer[16]	40017	SimDout[33-48]	DO 33-48	
	MBDataBuffer[17]	40018	SimDout[49-64]	DO 48-64	

	本地地址	寄存器地址	定义	说明	注释
	MBDataBuffer[18]	40019	Rob 执行命令状态	• bit0: 命令为 0 • bit1: 急停命令执行成功 • bit2: 启动命令执行成功 • bit3: 停止命令执行成功 • bit4: 复位命令执行成功 • bit5: 上使能命令成功 • bit6: 下使能命令成功 • bit7: 加载工程命令成功 • bit8: 注销工程命令成功 • bit9: 设置全局速度成功 • bit10: 等待控制权 • bit11: 等待命令 • bit12: 等待命令执行完成 • bit13: 命令执行错误 • bit14: 保留 • bit15: 保留	命令寄存器为 0 时, bit[0]为 1, 命令寄存器由命令时, bit[0]为 0 命令执行成功后, 相应成功位置 1 当重新获取控制权且命令为 0 时, 之前的成功位会被清零 因此, 可下发命令的状态码为 0x801
	MBDataBuffer[19]	40020	用户用	AO 1-16	每两个寄存器对应一个模拟量点 (浮点型)
					
					
					
	MBDataBuffer[50]	40051			
Rob Receive	MBDataBuffer[51]	40052	Rob 操作指令	• bit2 (0→0x4) : 机器人程序启动 • bit3 (0→0x8) : 机器人程序停止 • bit4 (0→0x10) : 机器人错误复位 • bit7 (0→0x80) : 加载工程文件 • bit8 (0→0x100) : 注销当前工程文件 • bit9 (0→0x200) : 设置全局速度 • bit10 (0→0x400) : 指令状态机重置	所有指令上升沿触发, 配合读写标志位 0x11 (命令状态位 0x801 时可下发指令) 注意: 当遇到命令响应失败时, 必须利用 bit10 复位状态机, 方可以下发新命令
	MBDataBuffer[52]	40053	全局速度值		
	MBDataBuffer[53]	40054	设置工程名	20 个字节	
	MBDataBuffer[54]	40055			
	MBDataBuffer[55]	40056			
	MBDataBuffer[56]	40057			
	MBDataBuffer[57]	40058			
	MBDataBuffer[58]	40059			
	MBDataBuffer[59]	40060			
	MBDataBuffer[60]	40061			

	本地地址	寄存器地址	定义	说明	注释
	MBDataBuffer[61]	40062			
	MBDataBuffer[62]	40063			
	MBDataBuffer[63]	40064	SimDI[1-16]	DI 1-16	
	MBDataBuffer[64]	40065	SimDI[17-32]	DI 17-32	
	MBDataBuffer[65]	40066	SimDI[33-48]	DI 33-48	
	MBDataBuffer[66]	40067	SimDI[49-64]	DI 48-64	
	MBDataBuffer[67]	40068	用户用	AI 1-16	每两个寄存器对应一个模拟量点（浮点型）
					
					
					
	MBDataBuffer[98]	40099			
	MBDataBuffer[99]	40100	读写标志位		0x11 打开 rob 命令下发权限
General	MBDataBuffer[100]	40101	用户用（已开放到内部 PLC）		读写权限
					
					
					
	MBDataBuffer[1499]	41500			

9.3 报警信息一览表

报警代码	报警内容	报警原因	解决措施
201	设置系统模式失败，不支持该模式	-	-
202	系统模式切换导致程序报警停止	系统处于运行状态下，切换系统模式会导致系统转入停止状态并报警提示	清除报警可继续工作
203	系统切换到远程模式，自动加载程序失败	可能的原因： 1. 未设置自启动加载的程序。 2. 自启动加载的工程或程序不存在。 3. 自启动加载的工程或者程序中有变量不合法。	可按照左侧三个原因，依次排查
204	系统切换到远程模式，启动设置全局速度失败	-	-
205	远程模式下，启动程序失败	系统内部问题	联系厂家
206	远程模式下，停止程序失败		
207	安全门打开，程序转到暂停状态	外界给安全门输入口触发了有效信号，程序由运行转到暂停状态	1. 检查是否真实触发了安全门信号。 2. 判断是否是接入的有效电平与系统配置界面设置的有效电平不一致。 3. 现场是否有干扰。
208	多工位加载程序失败！请检查预约程序的正确性	系统内部问题	联系厂家
209	与示教器断开连接，暂停运动	控制器与示教器网络断开，系统由运行转到暂停	检查网线连接
210	远程模式下，ExitCycle 只在非运行状态有效，暂停运动	ExitCycle 外部输入功能，仅在远程模式并且机器人处于非运行状态下才有效。	在远程模式并且机器人处于非运行状态下，再给予外部有效信号。
251	主管理模块 1 运行异常	系统内部问题	联系厂家
252	运动管理模块 2 运行异常		
253	指令管理模块 3 运行异常		
254	通讯模块 4 运行异常		
255	PLC 管理模块 5 运行异常		
256	视觉管理模块 6 运行异常		
257	现场总线 Modbus 模块 7 运行异常		
601	configTable.cfg 文件加载失败		
602	userTable.cfg 文件加载失败		
603	执行逻辑控制指令失败	可能的原因： 1. 逻辑判断指令格式不合法。 2. 指令中的变量不存在或者变量的某个属性不存在	1. 检查逻辑判断指令格式是否正确 2. 检查指令中的变量在变量列表中是否存在
604	命令执行失败	-	可根据错误信息做进一步的排查
605	执行失败！run 程序中存在禁止运行的指令在运行	启动的 run 程序中有禁止运行的指令存在	加载需要 run 的程序，对照说明书中 run 禁止执行的指令列表来删除该程序中示教的禁止运行的指令。
606	判断指令中有变量不存在	判断指令中的变量不存在，特指 AREA 变量	检查指令中的变量在变量列表中是否存在

报警代码	报警内容	报警原因	解决措施
607	加载指令字符串失败	可能的原因： 1. 指令名不对 2. 指令格式不正确 3. 指令中的变量名不存在 4. 指令中的变量属性不存在 5. 指令中有不合法的字符存在	在程序编辑界面对报错的指令进行编辑，按照左侧可能的原因挨个进行排查
608	运动内核缓存已满	-	-
609	加载程序数据失败	可能的原因： 1. 需加载的程序变量文件不存在 2. 程序变量文件中的数据有不合法的数据存在	在工程界面直接加载这个程序，观察加载时提示的报警信息，根据信息再进行下一步排查。
610	加载 run 程序数据失败		
611	加载触发程序数据失败		
612	解析失败！程序未加载	当前未加载程序，运行失败	加载需要运行的程序后再启动
613	解析逻辑控制指令失败	可能的原因： 1. 标签名重复 2. if、elseif、else 三者逻辑关系不匹配，或者有 endif 缺失 3. while、endwhile 不匹配	根据左侧的情况检查示教的程序流程信息
614	call 指令行变更，请重新载入工程	当前由运行切到暂停后，对 call 指令进行新建、修改、删除操作时再启动程序均会报此错误。	变更完成后，请重新加载程序再运行即可
615	执行 call 时获取 call 程序文件失败	可能的原因： 1. 待 call 的程序文件不存在 2. 待 call 的程序文件内容长度超限或者单行指令最大长度超限 3. 程序文件的版本号与当前指令系统版本不匹配	可在工程界面加载 call 失败的程序，观察提示的失败信息，根据这些信息做下一步的排查。
616	需要 call 的程序文件不存在	待 call 的程序文件不存在	从工程界面观察待 call 的程序存不存在
617	圆弧中间停止，不允许直接再启动	机器人在走圆弧过程中被暂停，此时禁止再启动。	请切换到停止状态再启动，如 pc 到圆弧的上一行。
618	系统最多同时支持八个程序 run	当前程序已经同时 run 了八个程序，再执行 run 会报此错误	请先 kill 掉不用的 run 程序再 run 新的程序，或者自行优化程序结构。
619	kill 与 run 程序不匹配	kill 指令只能 kill 已经 run 的程序，否则执行会报此错误。	检查程序结构
620	表达式指令执行失败	-	-
621	机器人零点已丢失！当前仅允许执行 RefRobotAxis 指令	在机器人已丢零点的情况下，系统仅允许执行 RefRobotAxis 指令。	请先回零再执行其他指令。
622	触发指令缓冲已满，请确保程序结构正确	-	-
623	变量类型不存在	执行变量的整体赋值指令时，操作的变量类型未在变量列表中定义	确保该变量类型在变量类型列表中已定义
624	变量类型不匹配	执行变量的整体赋值指令时，等号两侧的变量类型不匹配，赋值失败	确认等号两侧的变量类型为同一类型变量
625	单步模式下，不允许执行 run 指令	单步模式下不支持 run 指令	pc 到 run 指令的下一行再执行
626	程序处于 run 时，不允许再次 run	当程序已经处于 run 状态，此时再 run 该程序则会报错	可先 kill 该程序后再 run，或者重新注销加载程序再运行。
627	单步模式下，不允许执行触发指令	单步模式下不支持 OnDistance、OnParameter 触发指令	pc 到非 OnDistance、OnParameter 指令的下一行再执行

报警代码	报警内容	报警原因	解决措施
628	触发指令执行失败！请检查指令是否合法	-	-
629	OnParameter 触发指令不支持该运动指令	-	-
630	触发指令与运动指令之间不允许使用等待指令	OnDistance、OnParameter 与运动指令之间不允许使用 wait 等时间等待指令，否则触发指令将无意义。	检查程序中的指令结构，确保触发指令之间无 wait 等时间等待指令
631	软浮动指令之间，不允许运动指令	软浮动启动指令与结束指令之间禁止插入任何运动指令	移除掉两者之间的运动指令
632	软浮动停止指令之前缺少软浮动启动指令	软浮动结束指令必须与软浮动启动指令搭配使用	检查程序结构，是否缺少软浮动启动指令
633	多个软浮动指令嵌套使用	软浮动指令不允许嵌套使用	检查是否存在嵌套使用
634	缺少软浮动结束指令	软浮动结束指令必须与软浮动启动指令搭配使用	检查程序结构，是否缺少软浮动结束指令
635	该区域变量未被激活，不允许条件判断	区域变量未激活时，不允许进行条件判断操作	在变量界面对区域变量进行激活操作，或者在合适的位置添加区域激活指令。
636	单条轨迹的触发数量超过最大限制数量，执行失败	单条轨迹可支持的最大触发指令为 20 条，这个是包含 OnDistance、OnParameter 两类触发指令之和。	缩减单条轨迹的触发指令
637	获取程序变量文件失败	可能的原因： 1. 跳转到下一个程序时，程序文件加载失败。 2. 跳转到下一个程序时，程序数据获取失败。	在工程界面直接加载这个程序，观察加载时提示的报警信息，根据信息再进行下一步排查。
638	BendTrack 过程中停止，不允许再启动	折弯同步过程中停止，禁止直接再启动	请设置程序指针到其他行再启动
639	OnDistance 触发指令不支持该运动指令	该运动指令不在 OnDistance 支持的运动指令列表中，请按照其可支持的指令表中使用该触发指令。	联系厂家或者使用正确的运动指令。
640	触发指令执行失败	等待执行的 AddDo 指令或者触发指令在缓存中堆积数量超过了最大数量限制（1000 条），报错	检查程序结构，判断是否有阻塞执行的触发或 AddDo 指令
641	当前指令不支持倒退执行	倒序执行时，对指令有限制，仅支持部分指令	pc 到该指令的上一行，再进行倒序操作
642	待跳转的 label 不存在	待跳转的 label 名在当前程序中未定义	检查该 label 是否真实存在
643	跟随状态下有禁止的指令在运行	跟随状态下，仅支持部分指令执行，详细支持的指令可查看说明书中跟随功能相关章节。	请根据说明书的跟随状态下支持的指令列表来优化程序。
644	跟随过程中停止，请 pc 到带有用户坐标系的运动指令上再启动	跟随过程中发生停止，系统还处于动坐标系下，出于安全考虑，要求 pc 到带有用户坐标系的运动指令上再启动。	1. 可以 pc 到 MovLSyncQuit 指令上退出当前坐标系后再 pc 到指定行运动。 2. 可以在位置界面上将用户坐标系切换成想要 pc 到指定行所使用的用户坐标系，然后再 pc 到指定行运动。
645	跟随失败后跳转的 label 不能为空	MovLSync 指令没有指定跳转的 label	检查相关指令，添加跳转 label
646	全局程序中不支持 RUN、KILL 指令	-	删除 RUN、KILL 指令

报警代码	报警内容	报警原因	解决措施
647	全局程序中 CALL 指令不支持 call 到非全局程序以外的程序	-	检查修改 call 指令
648	AddDo 指令执行失败	AddDo 指令在 2s 内未执行成功	检查指令不能成功的具体原因，如 IO 被其它地方控制
649	单步模式下，不允许执行负载在线辨识指令	负载辨识相关指令不支持单步运行。	在连续自动模式下再运行负载辨识相关指令，或者 PC 到该指令的下一行继续单步运行。
650	CALL 指令不允许调用当前程序本身	当前程序名为 prog1，程序中有 CALL prog1 指令，那该指令执行将报 650 错误，因为 CALL 不能 CALL 自身所在的程序。	修改 CALL 指令或程序结构。
651	指令中参数数量异常	系统内部问题	联系厂家
652	指令中位置点数据异常	指令中传入的位置点参数异常，传入的是一个非表结构参数	在程序界面对指令进行编辑，查看传入的该位置参数是否是一个坐标点变量
653	指令参数中位置点类型不匹配	指令中传入的位置点参数类型与定义的参数类型不一致	在程序界面对指令进行编辑，查看传入的该位置参数是否是一个坐标点变量
654	指令参数中位置点类型无法找到	指令中传入的位置点参数在变量类型类表中未定义	联系厂家
655	指令参数中位置点中某项的值找不到	指令中传入的位置点参数某项的属性值在该变量类型中的属性值定义中未找到	
656	指令参数中速度项参数异常	-	-
657	指令参数中过渡类型参数异常	传入的过类型参数必须为 "RELATIVE"、"ABSOLUTE"、"FINE"三者中的一个	检查指令的过渡类型参数，看是否为 "RELATIVE"、"ABSOLUTE"、"FINE"三者中的一个
658	指令参数中触发参数异常	传入的触发类型参数必须为 "DITrig"、"TorqTrig"、"SIMDITrig"、"AITrig"、"SIMAITrig"中的一个	检查指令的触发类型参数，看是否为 "DITrig"、"TorqTrig"、"SIMDITrig"、"AITrig"、"SIMAITrig"中的一个
659	指令参数中 AddDO 类型参数异常	传入的 AddDo 类型参数必须是指令字符串格式的的参数	在程序界面对指令进行编辑，检查 AddDo 类型参数格式是否正确，也可重新修改该参数
660	指令参数中速度变量数据异常	指令中传入的速度变量参数异常，传入的是一个非表结构参数	在程序界面对指令进行编辑，查看传入的该速度参数是否是一个速度变量
661	指令参数中速度变量类型不匹配	指令中传入的速度变量参数类型与定义的参数类型不一致	在程序界面对指令进行编辑，查看传入的该速度参数是否是一个速度变量
662	指令参数中速度变量类型无法找到	指令中传入的速度变量参数在变量类型类表中未定义	联系厂家
663	指令参数中速度变量中某项的值找不到	指令中传入的速度变量参数某项的属性值在该变量类型中的属性值定义中未找到	
664	指令参数中过渡变量数据异常	指令中传入的过渡变量参数异常，传入的是一个非表结构参数	在程序界面对指令进行编辑，查看传入的该过渡参数是否是一个过渡变量
665	指令参数中过渡变量类型不匹配	指令中传入的过渡变量参数类型与定义的参数类型不一致	
666	指令参数中过渡变量类型无法找到	指令中传入的过渡变量参数在变量类型类表中未定义	联系厂家
667	指令参数中过渡变量中某项的值找不到	指令中传入的过渡变量参数某项的属性值在该变量类型中的属性值定义中未找到	

报警代码	报警内容	报警原因	解决措施
668	指令参数 POS 点中 MODE 类型找不到	指令中传入的位置点参数的 MODE 属性在变量类型类表中未定义	
669	指令参数 POS 点中 MODE 中某项的值找不到	指令中传入的位置点参数的 MODE 属性中某项属性值在该变量类型中的属性值定义中未找到	
670	指令参数中 socket 名字异常	指令中传入的 socket 名必须为字符串数据	在程序界面对指令进行编辑, 对 socket 名重新输入
671	指令参数中 IP 地址不合法	指令中传入的 IP 地址必须为字符串数据	在程序界面对指令进行编辑, 对 IP 地址重新输入
672	指令参数中 tool 坐标系参数异常	指令中传入的 tool 坐标系参数异常, 传入的是一个非表结构参数	在程序界面对指令进行编辑, 查看传入的该 tool 坐标系参数是否是一个 tool 坐标系变量
673	指令参数中 tool 坐标系类型不匹配	指令中传入的 tool 坐标系参数类型与定义的参数类型不一致	
674	指令参数中 tool 坐标系类型无法找到	指令中传入的 tool 坐标系参数在变量类型类表中未定义	联系厂家
675	指令参数中 tool 坐标系中某项的值找不到	指令中传入的 tool 坐标系参数某项的属性值在该变量类型中的属性值定义中未找到	
676	指令参数中用户坐标系参数异常	指令中传入的用户坐标系参数异常, 传入的是一个非表结构参数	在程序界面对指令进行编辑, 查看传入的该用户坐标系参数是否是一个用户坐标系变量
677	指令参数中用户坐标系类型不匹配	指令中传入的用户坐标系参数类型与定义的参数类型不一致	
678	指令参数中用户坐标系类型无法找到	指令中传入的用户坐标系参数在变量类型类表中未定义	联系厂家
679	指令参数中用户坐标系中某项的值找不到	指令中传入的用户坐标系参数某项的属性值在该变量类型中的属性值定义中未找到	
680	指令参数中负载惯量类型不匹配	指令中传入的负载惯量参数类型与定义的参数类型不一致	在程序界面对指令进行编辑, 查看传入的该负载惯量参数是否是一个负载惯量变量
681	指令参数中负载惯量类型无法找到	指令中传入的负载惯量参数在变量类型类表中未定义	联系厂家
682	指令参数中负载惯量中某项找不到	指令中传入的负载惯量参数某项的属性值在该变量类型中的属性值定义中未找到	
683	指令参数中摆弧类型不匹配	指令中传入的摆弧类型参数类型与定义的参数类型不一致	在程序界面对指令进行编辑, 查看传入的该摆弧类型系参数是否是一个摆弧变量
684	指令参数中摆弧类型无法找到	指令中传入的摆弧类型参数在变量类型类表中未定义	联系厂家
685	指令参数中摆弧中某项找不到	指令中传入的摆弧类型参数某项的属性值在该变量类型中的属性值定义中未找到	
686	指令参数中外部 TCP 坐标系参数异常	指令中传入的外部 TCP 坐标系参数异常, 传入的是一个非表结构参数	在程序界面对指令进行编辑, 查看传入的该外部 TCP 坐标系参数是否是一个外部 TCP 坐标系变量
687	指令参数中外部 TCP 坐标系类型不匹配	指令中传入的外部 TCP 坐标系参数类型与定义的参数类型不一致	
688	指令参数中外部 TCP 坐标系类型无法找到	指令中传入的外部 TCP 坐标系参数在变量类型类表中未定义	联系厂家
689	指令参数中外部 TCP 坐标系中某项的值找不到	指令中传入的外部 TCP 坐标系参数某项的属性值在该变量类型中的属性值定义中未找到	

报警代码	报警内容	报警原因	解决措施
690	指令中坐标系参数数据异常	指令中传入的坐标系参数异常，传入的是一个非表结构参数	在程序界面对指令进行编辑，查看传入的该坐标系参数是否是一个坐标系变量
691	指令参数中坐标系类型不匹配	指令中传入的坐标系参数类型与定义的参数类型不一致	
692	指令参数中坐标系类型无法找到	指令中传入的坐标系参数在变量类型类表中未定义	联系厂家
693	指令参数中坐标系中某项的值找不到	指令中传入的坐标系参数某项的属性值在该变量类型中的属性值定义中未找到	
694	指令参数中变位机坐标系参数异常	指令中传入的变位机坐标系参数异常，传入的是一个非表结构参数	在程序界面对指令进行编辑，查看传入的该变位机坐标系参数是否是一个变位机坐标系变量
695	指令参数中变位机坐标系类型不匹配	指令中传入的变位机坐标系参数类型与定义的参数类型不一致	
696	指令参数中变位机坐标系类型无法找到	指令中传入的变位机坐标系参数在变量类型类表中未定义	联系厂家
697	指令参数中变位机坐标系中某项的值找不到	指令中传入的变位机坐标系参数某项的属性值在该变量类型中的属性值定义中未找到	
698	指令中待跳转的 label 格式不正确	传入的待跳转 label 参数必须为字符串式的参数	在程序界面对指令进行编辑，对 label 进行重新选择
699	指令参数中动坐标系参数异常	指令中传入的动坐标系参数异常，传入的是一个非表结构参数	在程序界面对指令进行编辑，查看传入的该动坐标系参数是否是一个动坐标系变量
700	指令参数中动坐标系类型不匹配	指令中传入的动坐标系参数类型与定义的参数类型不一致	
701	指令参数中动坐标系类型无法找到	指令中传入的动坐标系参数在变量类型类表中未定义	联系厂家
702	指令参数中动坐标系中某项的值找不到	指令中传入的动坐标系参数某项的属性值在该变量类型中的属性值定义中未找到	
703	指令参数中速率类型参数异常	-	
704	指令参数中扫频参数异常	-	
705	指令参数中扫频类型不匹配	-	
706	指令参数中扫频类型无法找到	-	
707	指令参数中工具坐标系下的 DAPOS 变量无法找到	对于 MovJOffset 指令，其工具坐标系不能选择 DAPOS 变量	-
800	中断指令只能在中断程序调用时执行	IEnter、IExit、MovToIPos、ExitCycle 这些中断指令只能在中断程序中执行。	检查主程序或者 run 程序中是否示教了 IEnter、IExit、ExitCycle 这些中断指令。
801	执行失败！中断程序中存在禁止运行的指令在运行	中断程序中并不支持系统所有的指令。	请根据中断功能章节中的中断可支持指令列表来排查是否使用了其他指令。
802	中断程序中的运动指令只能在执行 IEnter 指令后执行	中断程序分为中断执行模式与中断并行模式，只有中断执行模式才能执行运动指令，即中断程序需先执行 IEnter 进入中断执行模式才可执行运动指令。	合理的在中断程序中示教 IEnter 指令
803	中断程序中的 IExit 指令前必须为 MovToIPos 指令	IEnter+运动指令+MovToIPos+IExit 三条指令必须按顺序示教，否则执行报错。	检查中断程序的指令结构，按照 IEnter+运动指令+MovToIPos+IExit 顺序来正确示教指令。
804	中断程序中的 IEnter 必须与 IExit 或 ExitCycle 指令搭配执行	中断程序有如下三种组合方式： IEnter +IExit IEnter+ExitCycle	-

报警代码	报警内容	报警原因	解决措施
		IEnter+ExitCycle+ IExit 如果不成对出现，系统将报错。	
805	中断程序中不支持 CALL 全局程序	中断程序只支持 CALL 同工程下的其他程序。	-
851	示教程序 call 嵌套次数达到系统限制，请重新加载程序运行	-	联系厂家
901	3D 视觉信息发送失败，请确认相机已连接	SendMessage 指令向相机发送命令失败，一般都是系统与相机端断连了	1. 检查系统与相机的网线是否连接。
902	2D 视觉拍照命令触发失败	trigCam 指令向相机发送命令失败，一般都是系统与相机端断连了	2. 相机与控制器系统的网络配置是否正确。
903	计算失败，请确认点位数据正确	九点标定计算失败，传入的点位数据不合法	检验需标定的点是否合理，重新标定
951	APOS 转 CPOS 失败	可能的原因： 1. 内核处于错误状态时，转换将失败 2. 等待转换结果超时	1. 确认系统当前的状态 2. 联系厂家
952	CPOS 转 APOS 失败	可能的原因： 1. 内核处于错误状态时，转换将失败 2. 等待转换结果超时	1. 确认系统当前的状态 2. 联系厂家
953	CPOS 转 CPOS 失败	可能的原因： 1. 内核处于错误状态时，转换将失败 2. 等待转换结果超时	1. 确认系统当前的状态 2. 联系厂家
954	位运算失败！不能对负数进行位操作	位运算指令中所带的参数必须为正数	检查当前位运算指令中的值是否为正数
955	当前码垛计数已经超过最大码垛数量	码垛变量中当前码垛计数不能超过最大可码垛的数量	-
956	放置失败,当前放置已满	码垛变量中当前码垛计数已经达到最大可码垛的数量，无法再进行放置操作。	1. 根据现场情况执行 PalletReset 指令来复位当前码垛计数。 2. 使用 PalletGet 等指令将该处放置的工件放到别处。 3. 根据现场情况来判断是否需要纠错处理。
957	抓取失败,当前可抓取工件为空	码垛变量中当前码垛计数已经达到最小可抓取的数量(0)，无法再进行抓取操作。	1. 根据现场情况执行 PalletReset 指令来复位当前码垛计数。 2. 根据现场情况来判断是否需要纠错处理。
958	码垛参考位置点不存在	传入的码垛参考点在变量列表中找到	在变量列表中查找码垛变量中设置的起始位置点与入口点在变量列表中是否存在。 不存在可新建同名点并重新示教，也可直接修改该码垛变量中的起始位置点与入口点
959	批量 IO 操作失败，起始端口号大于结束端口号	IO 相关的 8421 指令，读写时是从起始端口号到结束端口号的顺序（从小到大）来读写，两者不可反向。	合理的输入起始端口号与结束端口号
960	Hand 指令答复失败	Hand 指令执行超时	联系厂家
961	1~8 号输出口为系统输出口，禁止用户操作	前 8 号输出口为系统输出口，程序后台会自动控制这些口。	使用 8 号口以后的输出口来进行相关操作
962	读取的数据个数不能超过数组的大	socket 指令中的“读取数据个数”	根据现场情况将“读取数据个数”调

报警代码	报警内容	报警原因	解决措施
	小	参数值必须小于等于“数组返回值”中选择的数组变量中的 count 值。	小，或者在“数组返回值”中选择 count 大一点的变量。
963	IO 类型的触发值只能为 0 或者 1	数字输入口的值只有高（1）低（0）两个属性	在程序编辑界面对报警的指令进行修改，将其中的触发值参数设为 0 或 1
964	AreaActivate 或 AreaDeactivate 指令失败	AreaActivate 或 AreaDeactivate 指令失败！详细错误见与其一起报出来的错误信息	根据报警信息做下一步排查
965	参数异常，请检查参数是否合法	Hand 指令的“左右手设置”参数只能为"Lefty","Righty","Default"三者中的一种	在程序编辑界面编辑 Hand 指令，选取合理的左右手设置参数
966	无法获取明确的回退位置点，不能继续	倒序执行中获取回退点信息失败，无法回退。	检查当前是否满倒序条件
967	SetTargetPos 指令仅能在传送带为无视视觉触发模式下运行	-	-
968	动坐标系对应的传送带未使能，SetTargetPos 指令执行失败	-	-
969	根据动坐标系的值计算用户坐标系失败	-	-
970	伺服增益参数设置失败	-	-
971	阵列指令:行数及列数必须为大于 0 的整数	-	-
972	阵列指令:行号或列号大于设置值	-	-
973	阵列指令:行数和列数乘积需要小于限定值(32767)	-	-
974	当前运动指令选择不受全局速度影响模式时需设置运行速度	-	-
975	Damping 加速度限制系数设置失败	-	-
976	Damping 滤波器参数设置失败	-	-
977	同一时刻执行的脉冲指令数量超过了最大限制	-	请检查程序结构，确保同时执行的脉冲指令数量不要超过 20 个
978	传入的变量类型不合法	-	请检查该指令的参数类型是否合法
979	模型辨识数据采集失败	-	-
980	传送带使能状态设置失败	-	-
981	MovArch 指令中下降高度未设置	MovArch 指令中下降高度未设置	设置下降高度
982	参数不合法，传入的值只能为 1~100	-	修改参数
983	中断标识已与其他程序绑定，请先解绑再执行绑定操作	中断标识 INTERRUPT 有且只能与一个程序进行绑定	先执行 IDelete，再执行 IConnect 指令来将中断标识与程序绑定。
984	需要操作的中断标识未绑定	ITimer、ISignalDI、ISignalSimDI 等这些中断标识相关的指令执行之前，必须确保已经执行 IConnect 指令来将对应的中断标识与程序进行绑定操作，避免触发中断后无法找到对应的中断程序来执行。	先执行 IConnect 指令将对应的中断标识与程序进行绑定操作。
985	队列中已有待执行的中断操作命令		联系厂家
986	返回中断发生位置指令只能在 IEnter 触发后执行	IEnter+运动指令+MovToIPos +IExit 三条指令必须按顺序示教，否则执行报错。	按照报警原因合理示教对应的指令。

报警代码	报警内容	报警原因	解决措施
987	已执行过 IEnter 指令, 请执行 IExit 指令后再执行 IEnter 指令	中断程序有如下三种组合方式: IEnter + IExit IEnter+ExitCycle IEnter+ExitCycle+ IExit 如果不成对出现, 系统将报错。	
988	IExit 指令只能在 IEnter 触发后执行		
989	ExitCycle 指令只能在 IEnter 触发后执行		
990	设置转矩限制值失败		1、确认当前轴是否可设置; 2、确认参数是否合理; 3、确认相关配置文件是否正常
991	MovLArch 指令中下降高度未设置	MovLArch 指令中下降高度未设置	设置下降高度
992	计算位置点偏移结果失败	CalPosOffset 指令计算点位相对偏移的位置量失败	请修改 CalPosOffset 指令的参数
993	获取目标相对位置失败	-	-
994	获取参考坐标系失败	-	-
995	获取工具参数失败	-	-
996	获取坐标系参数失败	-	-
5103	不能朝远离正软限位方向运动	当前机器人关节位置处于正软限位外	请往限位内方向点动关节
5104	不能朝远离负软限位方向运动	当前机器人关节位置处于负软限位外	
5105	手动已达正软限位	当前机器人关节位置已到达正软限位	
5106	手动已达负软限位	当前机器人关节位置已到达负软限位	
5108	奇异点附近不能进行轴点动	当前机器人处于奇异区域内	点动关节至奇异区域外, 再进行轴点动
5109	点动逼近奇异点减速停	点动轴, 不能跨过奇异点。在逼近奇异位置一定范围时减速停止	请离开奇异外再进行轴点动
5110	参数配置错误无法点动	系统安全校验	联系厂家
5111	寸动参数设置超范围	寸动长度过大	寸动长度角度最大为 1 度, 长度最大为 10mm
5112	当前不允许进行多轴点动	一点对正、外部 TCP/变位机、姿态进行多轴 JOG	一点对正、外部 TCP/变位机、姿态不支持多轴 JOG
5115	厂家预留报警	软件内部错误	联系厂家
5116			
5121	系统未回零不接受启动命令	系统未回零不接受启动命令	请检查各轴回零状态, 系统需要在全部轴回零状态下才能启动程序运行
5122	系统最大速度配置为零	系统最大速度配置错误	联系厂家
5123	系统加速度配置为零	系统加速度配置错误	
5125	当前为非自动运行模式, 不允许执行运动指令	系统安全校验	
5133	跟随目标位置超最晚启动位置, 本次不再跟随	跟随边界校验	检查跟随边界参数
5134	跟随目标位置超最晚同步位置, 本次不再跟随		
5135	跟随目标位置超跟随边界, 本次不再跟随		
5136	跟随过渡段超机器人可达空间, 本次不再跟随	跟随轨迹超机器人可达空间	检查跟随动坐标系、跟随指令点位

报警代码	报警内容	报警原因	解决措施
5137	跟随编程段超机器人可达空间，本次不再跟随		
5138	跟随过渡段速度规划太小，请调整过渡参数	系统安全校验	修改过渡参数
5139	跟随运动段位移或时间规划太小，请调整过渡参数或重新示教		
5140	跟随同步上抬段轨迹规划错误，请修改上抬高度或联系厂家		联系厂家
5141	跟随建立过渡样条曲线错误，请修改过渡参数		修改过渡参数
5161	负载质量超范围	负载质量设置超过机型允许范围	联系厂家
5162	负载质心超范围	负载质心设置超过机型允许范围	
5163	负载惯量超范围	负载惯量设置超过机型允许范围	
5171	伺服参数文件无当前参数对象地址，需升级参数文件	加载伺服参数配置文件失败	
5172	伺服参数配置错误，需升级参数文件	伺服参数配置错误	
5173	因 EtherCAT sdo 命令正在运行中故不响应该命令	当前正在执行 SDO 操作	等待当前命令执行完成后再进行操作
5181	打开日志文件错误	系统安全校验	联系厂家
5182	写日志内存太大		
5183	准备日志数据溢出		
5184	打开运动控制数据文件错误		
5185	打开平移镜像日志文件错误		
5186	打开伺服参数备份文件错误	伺服参数备份文件加载失败	
5195	当前运动状态不响应该命令	当前为非空闲状态，无法响应命令	执行命令前请将系统切换至空闲状态
5196	中断恢复时的点位与发生中断时的点位不一致	系统安全校验	联系厂家
5197	中断恢复后本段剩余小圆弧将转为直线执行		此为中断过程状态提示，无需处理
5198	系统处于非正常运行状态不响应中断命令		
5199	插补处于非正常运行状态不响应中断命令		
5200	轴号超范围不响应该命令	当前设置的轴号超出机器人本体轴范围	正确设置轴号
5201	当前系统状态不响应该命令	可能的原因： 1.系统非空闲时执行运行指令/手动命令/一点对正命令/回零命令/坐标系选择/单圈值设置 2.系统非暂停时执行继续命令/伺服 Pn 参数设置 3.系统处于非运行状态执行跟随命令 4.系统处于运行状态执行跟随参数/传送带参数设置命令/碰撞检测参数设置/振动抑制参数设置	执行命令前请将系统切换至合适的状态
5202	当前系统模式不响应该命令	非手动模式不支持点动	切换到手动模式再进行点动

报警代码	报警内容	报警原因	解决措施
5203	虚拟轴不响应该命令	当前轴配置的是虚拟轴	虚轴时不能进行此操作
5204	伺服未励磁不响应该命令	可能的原因： 1.伺服未励磁时执行回零命令 2.伺服未励磁时执行运行命令	伺服励磁后再执行命令
5205	急停按下，不响应该命令	急停按下	清除急停信号后再操作，请确认外部环境安全下，再释放急停信号
5206	当前报警级别不响应该命令	可能的原因： 1.系统处于严重报警时，此时报警信息不能清除 2.系统处于报警状态，不能励磁	1.若处于严重报警且报警信息不能清除时，请重启控制器 2.若系统处于报警状态但报警信息可以清除，请清除报警后再执行励磁操作
5207	视觉目标物体队列空，不响应该命令	可能的原因： 1.视觉通讯未连接 2.视觉目标点位未下发	检查视觉通讯及视觉目标点是否及时发送至控制器
5208	机器人位置点个数超范围，不响应该命令	系统安全校验	联系厂家
5209	机器人位置点超运动范围	机器人位置点错误	请检查点位信息
5210	厂家预留报警	软件内部错误	联系厂家
5212	正在系统诊断中无法再次诊断	正在系统诊断中	请稍后操作
5213	仅 SCARA 支持左右手切换，当前机型不支持	当前机型不是 SCARA 机型	检查机型
5214	CANOpen 通讯类型不支持该命令	CANOpen 通讯类型不支持该命令	检查通讯类型
5215	当前 PLC 轴数超范围不响应该命令，请检查配置	PLC 轴数超所配置个数	联系厂家
5218	伺服大电未使能，请重新励磁	伺服大电未使能	确认伺服主电是否已使能
5219	回零中不响应该命令	当前系统处于回零中	请稍后操作
5220	点动坐标系与参考坐标系类型不匹配，无法点动	系统安全校验	联系厂家
5221	条件触发不及时，运动中请不要提高速度倍率，如倍率未提高请调整触发时间	条件触发未能及时检测，可能的原因：当前段执行时间少于触发设置的时间	1.正确设置触发时间 2.若不影响使用，忽略提示信息
5222	力矩设置值偏小，请改大 SetCartDyn 或 SetJointDyn 中的力矩百分比	SetCartDyn 或 SetJointDyn 指令中的力矩百分比设置	适当改大 SetCartDyn 或 SetJointDyn 指令里的力矩比例
5230	螺旋圈数过大，建议分段示教	螺旋运动圈数过大可能带来螺旋运动速度慢（超过 100 圈）	1.分两段或多段示教螺旋轨迹； 2.若不影响运动，可忽略报警信息
5235	Search 索引值超范围，不执行 Search 功能	轴号超范围	轴号超范围，请检查参数
5236	Search 触发值为负数，不执行 Search 功能	指令参数力矩值设置不合理	指令参数力矩值应为正数
5237	安全区域存在暂停信号，不响应该命令	在区域暂停信号生效时收到了启动命令	需要通过命令启动时，应先将区域暂停信号复位
5241	DMC 未使能加载模型参数文件	系统安全校验	联系厂家
5242	DMC 加载动力学模型参数失败		
5243	DMC 加载 VB 振动抑制模型参数失败		
5244	DMC 加载机器人连杆名义模型参数失败		
5245	DMC 传动参数的减速比为零		
5246	DMC 传动参数的关节最大转速为		

报警代码	报警内容	报警原因	解决措施
	零		
5247	DMC 传动参数的电机额定转矩为零		
5248	DMC 传动参数的电机过载率为零		
5249	DMC 动力学轴数超范围		
5250	DMC 动力学计算时模型参数文件未初始化		
5251	DMC 振动抑制时间参数为零		
5252	DMC 振动抑制频率计算异常		
5253	DMC 机器人轴数设置为零		
5254	DMC 加载 IS 振动抑制模型参数失败		
5255	DMC 加载振动抑制模型参数失败		
5256	DMC 工具或负载质量不能为负值	DMC 内部检测到工具或负载质量设置为负值	检查负载参数是否填写有误
5257	DMC 满载设置不能为负值	DMC 内部检测到满载质量设置为负值	检查满载参数是否有误
5258	[RC]细插补数据缓存空, 请联系厂家		
5259	[Sg]细插补数据缓存空, 请联系厂家		
5260	DMC 控制轴数不匹配, 需升级库文件		
5261	DMC 函数个数不匹配, 需升级库文件		
5262	DMC 函数的形参个数不匹配, 需升级库文件		
5263	DMC 结构体变量类型个数不匹配, 需升级库文件		
5264	DMC 结构体的大小不匹配, 需升级库文件		
5265	DMC 枚举变量类型个数不匹配, 需升级库文件		
5266	DMC 枚举的元素个数不匹配, 需升级库文件		
5267	DMC 未知错误, 需升级库文件		
5268	[RC]细插补缓存参数配置失败		
5269	[Sg]细插补缓存参数配置失败		
5270	[Sg]无限旋转轴范围配置错误		
5271	DMC 不支持该机器人类型		
5272	DMC 机器人类型与轴数不匹配		
5273	DMC 伺服参数自调节功能, 电机转子惯量参数设置超范围		
5274	DMC 伺服参数自调节功能, 关节最大负惯比参数设置超范围		
5275	DMC 伺服参数自调节功能, 低速段增益比例参数设置超范围	AutoGainEnable 指令中增益比例参数设置超范围	适当调整增益比例参数在合理范围
5276	DMC 伺服参数自调节功能, 低速段速度阈值参数设置超范围	AutoGainEnable 指令中速度阈值参数设置超范围	适当调整速度阈值参数在合理范围

报警代码	报警内容	报警原因	解决措施
5277	DMC 伺服参数自调节功能，类型配置项超范围	系统安全校验	联系厂家
5278	DMC 伺服参数自调节功能，获取伺服参数失败		
5279	DMC 伺服参数自调节功能，伺服参数 Kv 计算超范围		
5282	DMC 伺服参数自调节功能，伺服参数 Ki 计算超范围	系统安全校验	
5283	DMC 伺服自调节功能，伺服参数 Kp 计算超范围		
5284	DMC 伺服参数自调节功能，负惯比参数计算超范围		
5285	DMC 伺服参数自调节功能，初始化未完成不允许使用该功能的指令		
5286	DMC 伺服参数自调节功能，伺服参数文件获取参数失败	DMC 伺服参数自调节功能获取伺服参数文件失败	
5287	DMC 伺服参数自调节功能，伺服参数写入文件失败	系统安全校验	
5288	DMC 伺服参数自调节功能，伺服 Pn 恢复用时参数设置超范围		
5289	DMC-Json 文件读失败		
5290	DMC-Json 文件写失败		
5291	DMC-ini 文件不存在		
5292	DMC-ADP 参数计算初始化失败		
5401	碰撞检测功能打开	碰撞功能打开提示	提示信息，无需处理
5402	碰撞检测功能关闭	碰撞功能关闭提示	
5600	PLC 心跳异常	控制器与 PLC 断连	检查与 PLC 通讯的心跳是否正常
5601	编码器通道错误	编码器通道号未配置或者超过该版本支持的最大号数	请检查配置文件参数
5602	编码器减速比分母为零	编码器减速比分母为零	请检查编码器减速比参数
5603	配置参数缺失或设置错误	配置文件中参数缺失或设置超范围	请检查配置文件参数
5604	试运行时间到，请重启或联系厂家	系统安全校验	联系厂家
5605	长时间待机后自动上励磁时，启动命令与状态不一致		联系厂家
5606	长时间待机自动掉励磁后位置偏差过大		检查掉励磁后机器人的位置是否发生较大变化
5621	齿轮比分母为零	参数不能为零	请检查配置文件参数
5622	电子齿轮输出溢出	编码器分辨率设置错误	请检查编码器分辨率是否正确，或联系厂家
5623	丝杠导程为零	参数不能为零	请检查配置文件参数
5624	齿轮比超范围	齿轮比不能超过 500	请检查配置文件
5649	回零轴数超出范围不响应该命令	单轴回零指令的轴号超过配置范围	请重新输入轴号
5650	给定单圈值超范围回零失败	单轴回零的单圈值超过编码器分辨率	请重新输入单圈值
5651	伺服反馈单圈值超范围回零失败	系统安全校验	1.尝试多次回零 2.调节伺服 PID 参数 3.联系厂家

报警代码	报警内容	报警原因	解决措施
5652	打开丢零前位置文件错误, 请谨慎回零	去限位块的轴丢零并且读取丢零前位置数据失败	根据丢零关节实际位置谨慎回零
5671	摆动幅值设置超范围:(0~100)mm	摆动幅值参数设置不合理	摆动幅值参数应在 0~100mm 之间设置
5672	摆动频率设置超范围:(0~10)Hz	摆动频率参数设置不合理	摆动频率参数应在 0~10Hz 之间设置
5673	摆动绕 X 旋转角度超范围:(-180~180)deg	摆弧 Rot_X 参数设置不合理	摆动 Rot_X 参数应在-180°~180°之间设置
5674	摆动绕 Z 旋转角度超范围:(-180~180)deg	摆弧 Rot_Z 参数设置不合理	摆动 Rot_Z 参数应在-180°~180°之间设置
5675	摆动停止时间超范围:(0~600000)ms	摆弧 StopTime 参数设置不合理	StopTime 参数应在 0~600000ms 之间设置
5676	当前摆弧类型不支持:仅支持正弦摆(Type=1)和三角摆(Type=2)	摆弧类型设置错误	目前摆弧类型仅支持正弦摆(Type=1)和三角摆(Type=2)
5682	圆弧起点、中间点和目标点太近或共线, 请重新示教	可能的原因: 1.圆弧起点、终点、辅助点三点共线 2.圆弧起点、终点、辅助点三点中存在相同的点	请正确示教圆弧示教点
5683	圆弧弧长太短, 请重新示教	可能的原因: 圆弧起点、终点、辅助点三点相同	请正确示教圆弧示教点
5684	计算圆弧曲线错误, 请重新示教中间点或目标点	系统安全校验	联系厂家
5685	计算圆弧曲线错误, 请重新示教中间点或目标点		
5686	指令起点在奇异位置	所示教的指令起点处于奇异位置	请修改指令起点位置在奇异区域外
5687	指令终点在奇异位置	所示教的指令终点处于奇异位置	请修改指令终点位置在奇异区域外
5688	指令起终点跨奇异位置	不支持起终点跨奇异位置	请检查起点和终点位置
5689	指令过渡类型不支持	系统安全校验	联系厂家
5690	指令过渡值为零, 请检查过渡变量	过渡值小于 0.01	修改过渡值
5691	点位姿态超限, 请保证姿态在正负 180° 内	姿态 ABC 超过 $\pm 180^\circ$	获取正确的点位姿态
5692	指令速度值为零, 请检查速度变量	指令速度值为 0	修改指令速度
5693	指令解析关节超正软限位, 请检查参考坐标系及指令起终点	轨迹超关节正软限位	检查坐标系, 或修改指令起点或终点位置
5694	指令解析关节超负软限位, 请检查参考坐标系及指令起终点	轨迹超关节负软限位	检查坐标系, 或修改指令起点或终点位置
5695	建立过渡样条曲线错误, 请修改过渡参数	建立过渡三次样条曲线时错误	修改过渡参数, 或联系厂家
5696	指令段 GJobID 不连续	系统安全校验	联系厂家
5697	指令段 PJobID 不连续		
5698	指令段 AJobID 不连续		
5699	指令段 CJobID 不连续		
5700	仅 SCARA 支持拱形运动, 当前机型不支持	当前机型不支持拱形运动	仅 SCARA 机型才支持拱形运动
5701	指令段运动类型不支持	系统安全校验	联系厂家
5702	指令段解析未知错误		
5703	螺旋线步进距离不能大于螺旋半径	螺旋运动步进距离应小于螺旋线	请合理设置螺旋线参数

报警代码	报警内容	报警原因	解决措施
		半径	
5704	螺旋线起终点距离太近不能画出半圆，请重新示教	螺旋线起终点距离小于两倍螺旋半径	请合理设置螺旋线参数或重新示教螺旋点
5705	螺旋线示教点不能共线	螺旋线示教起点、辅助点、终点共线不能画出螺旋轨迹	请重新示教螺旋点
5706	连续旋转关节不支持绝对位移指令	连续旋转关节只支持 MC_MoveRelative 及 MC_MoveVelocity 功能块，此报警仅 PLC 轴有效	连续旋转关节只支持 MC_MoveRelative 及 MC_MoveVelocity 功能块，此报警仅 PLC 轴有效
5707	过渡段速度规划太小，请调整过渡参数	系统安全校验	调整过渡参数或 WaitFinish 比例
5708	运动段位移或时间规划太小，请调整过渡参数或重新示教		
5709	轨迹运行过程中存在奇异位置，请重新示教		重新示教或修改点位
5710	运动段规划不够减速，请调整过渡参数或 WaitFinish 比例		调整过渡参数或 WaitFinish 比例
5711	运动段规划不够减速，请调整过渡参数或 WaitFinish 比例		
5712	运动段规划获取段不及时，请调整过渡参数或 WaitFinish 比例		报警段修改为不过渡，或者联系厂家
5713	运动段的过渡速度倍率规划到零，请联系厂家修改运动规划方式		联系厂家
5720	指令解析关节超耦合软限位，请检查参考坐标系及指令起终点	关节超耦合软限位	请往限位内方向点动关节，重新示教点位
5721	关节逼近正软限位	机器人某一轴当前位置在正软限位附近	1.请往限位内方向点动关节 2.修改示教位置点在限位内
5722	关节逼近负软限位	机器人某一轴当前位置在负软限位附近	
5723	关节逼近耦合软限位	机器人某一轴当前位置在耦合软限位附近	
5731	接近工作或禁止区域边缘	机器人当前位置在工作区域或者禁止区域边缘附近	1.修改示教位置点，使其远离区域边界一段距离（默认 15mm） 2.如空间受限，距离需要缩短，联系厂家
5742	平均力矩过载	实际力矩超过允许范围	1.检查反馈力矩有无明显异常波动、毛刺、静态力矩值，如有异常，联系厂家 2.如反馈力矩无明显异常，可降低程序运行的加速度、加加速度
5743	平均功率过载	平均功率超过允许范围	1.检查反馈力矩有无明显异常波动、毛刺、静态力矩值，如有异常，联系厂家 2.如反馈力矩无明显异常，可降低程序运行的加速度、加加速度
5751	负载惯量为负值	惯量值不能为负	设置真实惯量
5761	关节超过最大加速度	系统安全校验	联系厂家
5781	加工段的段地址错误，请联系厂家	系统安全校验	联系厂家
5782	加工段的插补周期为零，请检查配置参数		
5783	加工段为非运动指令段，请升级版本		

报警代码	报警内容	报警原因	解决措施
5784	加工段的速度为零, 请检查速度参数		
5785	加工段的加速度为零, 请检查加速度参数		
5786	加工段的减速度为零, 请检查减速度参数		
5787	加工段的加速比例为零		
5788	加工段的减速比例为零		
5789	加工段的终点速度被修改, 请调整过渡参数或速度参数		修改过渡参数、速度参数, 或联系厂家
5790	加工段的起点速度被修改, 请调整过渡参数或速度参数		
5792	加工段的加速段 Jerk 为零		联系厂家
5793	加工段的减速段 Jerk 为零		
5794	加工段的运算参数错误		
5801	过渡段的段地址错误, 请联系厂家		
5802	过渡段的速度为零, 请调整过渡参数或速度参数	过渡段的速度为零	调整过渡参数或速度参数
5803	过渡不打断失败, 请增大过渡参数或减小 WaitFinish 比例	系统安全校验	修改过渡参数、速度参数, 或联系厂家
5804	加工段插补超关节速度, 请调整速度参数或联系厂家	加工段插补超关节速度	联系厂家
5821	触发段已插补完, 请调整触发时间或降低运动段速度	触发段已插补完	请调整触发时间或降低运动段速度
5822	触发段地址异常为空, 请联系厂家	系统安全校验	联系厂家
5823	触发段时间计算失败, 请调整触发时间或降低运动段速度	触发时间或运动速度参数不合理, 无法触发	调整触发时间或降低运动段速度
5824	同步触发的缓存不够, 请升级版本	同一时刻触发的数量大于 10	请减少同一时刻触发的个数
5825	MovJ 运动不支持 OnDistance 触发	MovJ 运动不支持 OnDistance	使用 OnParameter 替换 OnDistance 触发指令
5826	触发段已解析, 未及时读取触发, 请调整触发时间或降低运动段速度	系统安全校验	联系厂家
5831	多边体区域类型设置错误	区域类型应为工作区域或禁止区域或信号工作区域	按照提示检查参数
5832	多边体区域顶点或边数设置错误	多边体区域的顶点距离太近或顶点个数大于 10 或小于 3 或边存在交叉	
5833	多边体区域参数文件写入失败	写入多边体区域参数文件失败	重启系统或关闭已打开的参数文件
5834	多边体区域不存在, 请重新设置	当前多边体区域不存在	按照提示检查参数
5841	区域范围太小, 需 5mm 以上	1.盒形区域长度 LenX/LenY/LenZ 太短 (应大于 5mm), 请检查 2.圆柱区域半径/高度太短 (应大于 5mm) 3.轴干涉区最大值与最小值的差值应大于 5mm/5°	按照提示检查参数
5842	区域号超出限制	系统安全校验	联系厂家
5843	圆柱区域半径需为正数	圆柱区域半径不能为负数或零	按照提示检查参数
5844	区域未激活, 无法连接启停信号	在区域冻结时连接了启停信号	先激活区域, 再连接启停信号

报警代码	报警内容	报警原因	解决措施
5845	区域激活 DI 与启停 DI 端口号冲突, 请重新设置	区域启停与激活使用了相同的输入端口	重新设置不同的端口号
5850	焊枪与行进路线平行, 请调整焊枪姿态	焊枪与行进路线平行会导致计算错误	调整焊枪的姿态
5860	PN 参数初始化失败, 碰撞后快速停止功能关闭	伺服连线网络异常	1.请检查伺服网络连线是否正常。 2 检测 ServoParam.ini 文件配置。 3.联系厂家
5865	当前机型不支持杆件参数修改	当前机型不支持杆件参数修改	当前机型不支持杆件参数修改
5866	杆件参数设置超范围	臂长小于 0, 或偏角不在 -180° ~180°	请配置合理的杆件参数
5870	伺服未励磁无法启动程序	伺服未励磁	检查伺服励磁状态
5871	伺服未励磁无法回零		
5872	参数辨识数据采集过程中无需重复采集	当前正在采集参数辨识数据	无需重复采集
5873	参数辨识数据采集未开始无需结束采集	当前未开始采集参数辨识数据	无需结束采集
5874	当前状态全局速度为零, 请设置非零全局速度参数	全局速度设置为零进行运动	设置全局速度为非零值后, 运动
5880	变位机参数写文件创建失败, 请重启系统	写变位机文件失败	重启系统或联系厂家
5881	变位机 ID 不在可用范围, 请选择已存在的变位机 ID	系统安全校验	确认 ID 号设置是否正确, 如无误请联系厂家
5882	变位机附加轴未配置	变位机对应的附加轴未配置	1.若使用变位机, 请按步骤正确配置附加轴参数 2.若不使用变位机, 请联系厂家
5883	变位机附加轴轴号超范围, 请修改配置轴号	变位机对应的附加轴号超过系统总配置轴数	1.正确配置附加轴号 2.若为标定时报警, 请联系厂家
5884	变位机附加轴轴号相同, 请修改配置轴号	同一个变位机对应的附加轴有相同的轴号	1.若使用变位机, 请检查变位机附加轴轴号配置 2.若不使用变位机, 请联系厂家
5885	变位机附加轴轴号与本体轴相同, 请修改配置轴号	变位机对应的附加轴号与本体轴号相同	
5886	变位机附加轴类型错误, 请修改轴参数 LinkType	变位机对应的附加轴类型配置错误	1.若使用变位机, 请检查变位机附加轴轴类型配置 2.若不使用变位机, 请联系厂家
5890	跟随运动不支持跨奇异	在跟随运动中开启了跨奇异功能	关闭跨奇异开启指令
5891	摆弧运动不支持跨奇异	摆弧起终点跨奇异, 同时开启了跨奇异功能	
5892	外部 TCP 运动不支持跨奇异	外部 TCP 运动起终点跨奇异, 同时开启了跨奇异功能	
5893	当前机型不支持跨奇异	当前机型不是通用六关节机器人	
5894	示教终点超限或在奇异点不支持跨奇异	1.示教终点在奇异点 2.示教终点位置没有解	请重新示教
5895	不能跨非腕部奇异	跨奇异功能只支持腕部奇异, 非腕部奇异不支持	请重新示教或者关闭跨奇异功能
5900	示教终点超臂展空间, 请检查参考坐标系及示教点	示教终点超臂展空间	检查参考坐标系及示教点
5901	过渡轨迹超臂展空间, 请检查参考坐标系及示教点	过渡轨迹超臂展空间	
5902	运动轨迹超臂展空间, 请检查参考坐标系及示教点	运动轨迹超臂展空间	
5903	当前运动不支持速度不受倍率影响	当前运动的速率类型不支持	修改当前指令的速率类型

报警代码	报警内容	报警原因	解决措施
5904	当前运动坐标系下不支持工具偏移运动	当前不是用户坐标系或 World 坐标系，不支持工具偏移运动	1、检查坐标系； 2、若无法改变坐标系，请删除该指令
5911	中断过程中不响应该命令	当前处于中断过程，不响应命令	退出中断后再执行相应的命令
5912	中断停止在过渡段或 MovArch 指令上，中断响应失败	系统安全校验	联系厂家
5913	中断停止在跨奇异段上，中断响应失败		
5914	中断停止在相对运动段上，中断响应失败		
5920	碰撞检测参数写文件失败	写碰撞检测参数文件失败	重新生成碰撞检测参数
5921	碰撞检测参数的 ID 超范围，请确认设置的 ID 在 1 到 6 范围内	碰撞检测参数对应的 ID 不在 1 到 6 的范围内	修改碰撞检测参数对应的 ID 值
5922	碰撞检测参数 ID 未被标定，请标定后使用	碰撞检测参数对应的 ID 没有被标定过	选择已经标定过的 ID
5930	跟随过程中不响应该命令，可通过切换参考坐标系退出跟随后再操作	当前处于跟随过程，命令不响应	切换参考坐标系为世界坐标系或用户坐标系
5931	传送带未连接，请检查通讯配置及 ECAT 相关文件	系统安全校验	检查传送带通讯配置文件，或联系厂家
5932	跟随触发时间配置太小		根据实际视野范围正确设置跟随触发时间
5933	跟随触发 DI 端口号配置不在可用范围，请重新配置		正确设置 DI 端口号，使用通用 DI
5934	跟随触发 DO 端口号配置不在可用范围，请重新配置		正确设置 DO 端口号，使用通用 DO
5935	传送带脉冲当量设置错误	所设置的传送带脉冲当量过小或过大	依据操作手册的步骤正确标定传送带脉冲当量
5936	跟随速度或加减速系数设置错误	跟随速度/加速度/减速度系数设置错误	跟随速度/加速度/减速度系数应在 0.001~1 之间设置
5937	跟随边界参数设置错误	跟随边界参数设置错误	修改并正确设置边界参数
5938	折弯模具参数设置错误	折弯模具参数设置错误	修改并正确设置折弯模具参数
5940	MovLSync 解析打断标志有误	系统安全校验	联系厂家
5941	跟随未启动无需退出		
5942	跟随启动重复		
5943	跟随过程中不支持该类型的运动指令	该指令不支持	删除该指令
5944	跟随过程中不支持 JPos 类型的目标点	跟随过程中使用了 JPos 类型的点位	使用 CPos 类型的点位
5945	跟随过程中不支持摆弧指令	该指令不支持	按照提示修改或者删除对应指令
5946	跟随过程中不支持寻位指令		
5947	跟随过程中不支持切换工具参数		
5948	跟随过程中不支持切换参考坐标系		
5949	跟随规划坐标系切换错误	系统安全校验	联系厂家
5950	动坐标系对应的传送带 ID 号错误		检查传送带通讯配置文件，或联系厂家
5951	动坐标系对应的传送带未使能	传送带未使能	使能与动坐标系绑定的传送带
5952	跟随目标物体队列空	未获取到视觉点位信息	检查视觉通讯及视觉目标点是否及时发送至控制器

报警代码	报警内容	报警原因	解决措施
5953	跟随规划坐标系错误跟随无法退出	系统安全校验	联系厂家
5960	跟随过程中不支持指令速度不受倍率影响	跟随过程中运动指令的速率类型设置错误	修改当前指令的速率类型
5971	跟随退出时上一段不能带过渡	跟随退出前的运动指令中有过渡	检查过渡参数
5972	跟随退出时上一段不能是 WaitFinish 指令	跟随退出前是 WaitFinish 指令	检查程序中的指令
5973	出现非笛卡尔空间运动指令导致跟随无法退出, 请联系厂家	系统安全校验	联系厂家
5974	参考坐标系类型错误导致跟随无法退出		
5975	出现异常跟随状态导致跟随无法退出, 请联系厂家		
5978	衔接段未准备完成导致跟随无法退出, 请增大 WaitFTimeFTk 参数		
5979	传送带 ID 错误不跟随		检查传送带通讯配置文件, 或联系厂家
5980	跟随位移允差不能为零	跟随允差设置错误	检查跟随允差值
5981	诊断期间设置虚拟传送带无效	在诊断界面勾选诊断参数的同时设置了虚拟传送带	取消诊断界面勾选的参数
5982	传送带已使能, 设置折弯跟随参数失败	传送带使能, 占用了光栅尺对应的编码器通道	关闭当前已使能的传送带
5983	光栅尺已使能, 设置传送带跟随参数失败	光栅尺使能, 占用了传送带对应的编码器通道	关闭当前已使能的光栅尺
5984	折弯跟随运动超机器人臂展空间	系统安全校验	调整跟随动坐标系或指令点位
6101	关节超过最大软限位	关节超过最大软限位	请往限位内方向点动关节, 或重新示教点位
6102	关节超过最小软限位	关节超过最小软限位	
6103	关节号超范围	系统安全校验	联系厂家
6104	轴号超范围		
6105	机器人类型错误		
6106	机器人个数错误		
6107	目标位置超机器人运动范围或在奇异位置	目标位置超机器人运动范围或在奇异位置	请检查示教点位置
6108	加载用户动态库失败, 请重启, 多次失败请联系厂家升级库文件	系统安全校验	联系厂家
6109	获取用户动态库函数指针异常, 请重启, 多次失败请联系厂家升级库文件		
6110	加载模型动态库失败, 请重启, 多次失败请联系厂家升级库文件	加载模型动态库失败	请重启, 多次失败请联系厂家升级库文件
6112	加载碰撞检测动态库失败, 请重启, 多次失败请联系厂家升级库文件	加载碰撞检测动态库失败	
6114	获取插值库函数指针异常, 请重启, 多次失败请联系厂家升级库文件	系统安全校验	联系厂家
6115	加载插值动态库失败, 请重启, 多次失败请联系厂家升级库文件	加载插值动态库失败	请重启, 多次失败请联系厂家升级库文件
6116	获取动力学模型库函数指针异常, 请重启, 多次失败请联系厂家升级库文件	系统安全校验	联系厂家

报警代码	报警内容	报警原因	解决措施
6117	加载动力学模型动态库失败，请重启，多次失败请联系厂家升级库文件	加载动力学模型动态库失败	请重启，多次失败请联系厂家升级库文件
6118	加载软浮动动态库失败，请重启，多次失败请联系厂家升级库文件	加载软浮动动态库失败	
6122	手动参数配置内存异常	系统安全校验	联系厂家
6123	齿轮比分子或者分母为零	齿轮比分子或者分母为零	参数不能配置为零，请检查配置文件
6124	电机脉冲当量或者丝杠导程为零	电机脉冲当量或者丝杠导程为零	参数不能配置为零，请检查配置文件
6125	厂家预留报警	软件内部错误	联系厂家
6131	系统未回零不能进行轴点动，可进行关节点动	系统未回零不能进行轴点动	请先回零，再点动轴
6132	关节未激活不能点动	配置文件中未配该轴参数信息	请检查配置文件
6133	点动目标位置不可达	目标位置不可达	请在机器人可达空间范围内进行点动
6134	关节超耦合软限位	关节超耦合软限位	请往限位内方向点动关节，重新示教点位
6141	系统未回零不能走指令	系统未回零	请先回零
6142	加工段解析慢，请调整过渡参数或 WaitFinish 比例	系统安全校验	修改过渡参数或速度参数，或者联系厂家
6143	无法从当前暂停状态再启动运行，请联系厂家		联系厂家
6145	滤波器规划失败，请联系厂家	启动位置偏差校验过大	请从安全位置重新运行，或者联系厂家
6149	折弯坐标系计算错误，跟随失败	系统安全校验	联系厂家
6156	折弯跟随参考坐标系数据为空	系统安全校验	联系厂家
6157	折弯跟随插补时间与系统插补周期不一致，请检查折弯跟随数据		
6158	折弯跟随导轨速度不能为负值或零	导轨速度为负或 0	请设置合理速度参数
6159	折弯跟随导轨运动链类型设置错误，请修改轴参数 LinkType	导轨运动链类型设置错误	导轨轴 LinkType 配置错误，请检查配置文件
6160	折弯跟随导轨轨迹规划错误，请修改速度参数	系统安全校验	联系厂家
6161	脉冲输出超关节最大加速度，请联系厂家	关节超加速度	1.请移动机器人到安全位置，重新运行 2.降低程序运行的加速度、加加速度，如仍然报警请联系厂家
6162	发给伺服脉冲频率变化率过大，请联系厂家	关节超加速度	
6163	脉冲输出超关节最大速度，请联系厂家	关节超速	
6164	发给伺服脉冲频率过大，请联系厂家	关节超速	
6181	机器人末端在工作区域外	机器人末端在工作区域外	确认外部环境安全下，关闭区域，点动机器人使其回到工作区域内，再激活区域
6182	机器人末端在禁止区域内	机器人末端在禁止区域内	确认外部环境安全下，关闭区域，点动机器人使其回到禁止区域外，再激活区域
6191	伺服控制模式错误不能运行指令	模式未切换成功就下发了运行命令	检查逻辑或检查轴实虚轴配置
6192	软浮动方向与机器人受力点绕世界	软浮动指定方向角错误	请重新设置软浮动方向，确保夹角

报警代码	报警内容	报警原因	解决措施
	坐标系 Z 轴旋转圆切线方向夹角过大		小于 60 度
6193	软浮动超速，请联系厂家	直线软浮动运行速度大于系统设置的默认最大速度（500mm/s），或者当前位置在奇异点附近	检查机器人当前位置或联系厂家
6194	此机型不支持软浮动	当前机型不支持直线软浮动功能	请确认当前机型是否支持软浮动功能，若支持，请检查软浮动功能是否开启
6195	伺服模式切换错误，请联系厂家	系统安全校验	联系厂家
6196	软浮动执行时间超时，可修改超时参数	软浮动执行时间超过系统配置的最大默认时间 60 秒	
6201	[RC]细插补模块的插补周期为零	系统安全校验	检查参数配置
6202	[RC]细插补模块的插补周期超范围		
6203	[RC]细插补模块的插补类型超范围		
6204	[RC]细插补步数小于 1		联系厂家
6205	[RC]细插补数据缓存满		
6206	[RC]细插补数据缓存空		升级 CUB 库
6207	[RC]细插补未知错误，需升级库文件		
6211	[Sg]细插补模块的插补周期为零	附加轴控制周期配置参数不合理	检查参数配置
6212	[Sg]细插补模块的插补周期超范围		
6213	[Sg]细插补模块的插补类型超范围	附加轴细插补类型配置参数不合理	
6214	[Sg]细插补步数异常	系统安全校验	联系厂家
6215	[Sg]细插补数据缓存满		
6216	[Sg]细插补数据缓存空		
6217	[Sg]细插补未知错误，需升级库文件		升级 CUB 库
6501	运行中手压松开	点动运行或程序运行中，手压松开掉励磁	如需再次运行，按下手压即可
6502	运行中异常掉励磁	运行中未知原因掉励磁	检查伺服励磁状态
6503	安全门打开	系统安全校验	关闭安全门
6601	关节超差，请检查最大跟踪误差参数，如参数合理请联系厂家	给定位置与反馈位置偏差超过范围	请检查配置文件的最大跟踪误差参数是否合理。如改大范围仍无效请联系厂家
6621	力矩波动检测超范围	伺服力矩波动过大，可能的原因： 1.与障碍物发生接触碰撞 2.机器人设置的负载质量等信息与实际不符 3.机器人工作轨迹恶劣，机器人抖动厉害	1.检查是否轨迹异常导致碰撞，或障碍物出现在了正常运行的机器人轨迹上 2.按用户手册中负载方向及单位定义，正确设置负载信息 3.优化机器人轨迹，降低动态参数，减缓抖动情况 4.在不影响正常使用的前提下，可适当降低碰撞检测的灵敏度 5.调整适合的 PID 参数 6.先慢速热机，热机完成后再全速运行
6622	力矩波动检测超范围	4.电机或机械性能造成的实际力矩输出有毛刺尖峰 5.伺服的 PID 设置不好导致给定与实际的跟踪性不好 6.由于天气寒冷造成的冷机状态	

报警代码	报警内容	报警原因	解决措施
		影响 7.机器人设置的安装方式，比如正装或者倒装与实际不符	7.设置正确的安装方式 8.上述处理后仍有报警请联系厂家
6623	焊枪碰撞	检测到焊枪对应有效 IO 信号输入	1.如误报警，请检查设置的焊枪 IO 信号是否正常 2.如机器人发生碰撞，将机器人移动至安全位置后再运行机器人
6641	最大力矩过载	同 6621/6622	同 6621/6622
6661	常规停止已打开	当前已触发 AS 信号	AS 信号触发，请确认外部环境安全下，才能复位信号
6662	常规停止已打开不能励磁	当前已触发 AS 信号情况下上励磁	请确认外部环境安全下，才能复位信号再上励磁
6663	自动停止已打开	当前已触发 GS 信号	GS 信号触发，请确认外部环境安全下，才能复位信号
6664	自动停止已打开不能励磁	当前已触发 GS 信号情况下上励磁	请确认外部环境安全下，才能复位信号再上励磁
6900	急停按下	当前已拍下急停	急停按下，请确认外部环境安全下，才能释放急停开关
6901	急停按下不能励磁	急停未释放情况下上励磁	请确认外部环境安全下，释放急停开关再上励磁
6902	外部急停按下	外部急停未释放情况下上励磁	请确认外部环境安全下，释放外部急停开关再上励磁
6903	电柜异常掉电	C1 电柜发生了异常掉电	请检查电柜上电状态
6910	碰撞停止距离超范围	系统安全校验	请联系厂家
6911	PN 参数写错误，请重启系统	伺服网络异常	请断电重启
7000	EtherCAT 报警	EtherCAT 网络未成功连接	检查网络硬件连接，使用正确 bootfile 文件和 eclr 文件，若依然存在问题，联系厂家
7001-7099	EtherCAT 伺服报警	对应伺服报警 A.xx	检查伺服
7400	EtherCAT 通讯报警：从站未到 OP 状态	EtherCAT 网络未成功连接	检查网络硬件连接，使用正确的通讯配置文件，若依然存在问题，联系厂家
7401	EtherCAT 通讯报警：参数配置错误，请检查相关配置	参数配置错误	检查相应参数配置
7402	EtherCAT 通讯报警：网口配置错误，请检查相关配置	网口配置错误	检查相应网口配置
7403	EtherCAT 通讯报警：初始化句柄错误，请重启系统	系统安全校验	尝试重启，若仍未解决联系厂家
7404	EtherCAT 通讯报警：加载 eni 文件失败，请检查 eni 文件是否丢失	加载文件失败	检查通讯配置文件文件是否存在或正确
7405	EtherCAT 通讯报警：主站通信配置失败，请检查从站是否匹配	配置错误	检查相应配置是否与实际相符
7411	EtherCAT 通讯报警：未连接网线，请检查网络连接	EtherCAT 网络未成功连接	检查控制器与从站之间的网络连接，检查从站之间的网络连接，或从站连接是否正确
7412	EtherCAT 通讯报警：网络连接错误，请检查网络连接		检查控制器与从站之间的网络连接，检查从站之间的网络连接，或从站连接是否正确
7413	EtherCAT 通讯报警：从站配置错误，请检查耦合器模块拨码开关	实际连接与配置的耦合器模块拨码开关不符合	检查耦合器模块的拨码开关是否正确
7600	EtherCAT 伺服回零失败	伺服回零不成功或者伺服回零时间太长	尝试再次回零，若仍未解决联系厂家

报警代码	报警内容	报警原因	解决措施
7601	EtherCAT 伺服回零过程中掉励磁	正在回零过程中掉励磁	尝试再次回零，注意回零过程中不要掉励磁
7800	EtherCAT 伺服未上主电	当前伺服未上主电	检查伺服主电连接状态
7801	EtherCAT FOE 命令超时	EtherCAT 从站未响应 FOE 命令	检查从站的 FOE 功能
7802	EtherCAT FOE 命令失败，从站状态切换 BOOTSTRAP 失败	EtherCAT 从站未切换到 BOOTSTRAP 状态	检查从站是否支持 FOE 功能
7803	EtherCAT FOE 命令失败，从站不支持	EtherCAT 从站不支持 FOE 功能	检查从站是否支持 FOE 功能
7804	EtherCAT FOE 命令失败，文件打开失败	文件打开失败	检查用于 FOE DOWNLOAD 的文件是否存在或正确
7805	EtherCAT FOE 命令失败，从站反馈错误	EtherCAT 从站反馈 FOE 命令错误	根据从站 FOE 命令反馈的错误进行检查
7806	EtherCAT FOE 命令失败，从站上传的值为空	EtherCAT 从站 FOE 上传的值为空	检查从站的 FOE 功能
7990	EtherCAT IO 安全逻辑板报警	-	-
8600	CANOPEN 伺服回零失败	伺服回零不成功或者伺服回零时间太长	联系厂家
29001-29081	厂家预留报警	软件内部错误	
29101	加载内核报警文件失败，需升级报警文件	文件版本不匹配	升级文件
29102	加载机器人参数配置文件失败，需升级参数文件		
29103	机器人参数配置错误，缺少参数或者参数范围错误	缺少参数或者参数范围错误	按照提示检查参数是否缺少、参数名称、参数范围是否正确
29104	系统不支持此机型	系统不支持此机型	检查机型名称
29106	[RC]关节通道号配置错误与其它轴重复或超范围	关节通道号配置错误与其它轴重复或超范围	检查关节通道号
29107	关节频率文件组数超限或负载顺序配置错误	组数范围应在 2-10，须由满载到空载顺序填写各轴频率	检查频率参数配置文件
29108	插补周期错误或者细插补方式错误	插补周期、CUB 库开关、CUB 类型不匹配	检查周期参数、CUB 库开关、CUB 类型
29110	[RC&Sg]关节通道号配置个数超范围	关节通道号配置个数超范围	减少关节个数
29111	[RC&Sg]关节通道号配置不连续	关节通道号配置不连续	检查关节通道号
29112	[Sg]关节通道号配置错误:与其它轴重复或超范围	关节通道号配置错误与其它轴重复或超范围	
29113	[RC&Sg]关节通道号配置错误:与其它轴重复	关节通道号配置错误与其它轴重复	
29114	监控数据个数超出限制	系统安全校验	联系厂家
29115	加载机器人轴参数配置文件失败，需升级参数文件	文件版本不匹配	检查系统参数文件是否存在以及版本是否匹配
29116	加载系统 IO 参数配置文件失败，需升级参数文件	文件版本不匹配	检查 IO 参数文件是否存在以及版本是否匹配
29118	加载频率参数配置文件失败，需升级参数文件	文件打开失败	频率参数配置文件打开失败，请检查文件是否存在
29120	通讯总轴数超范围	通讯总轴数超范围	检查通讯方式与通讯轴数是否匹配
29121	[RC&Sg]关节正限位脉冲溢出，请设置伺服电子齿轮比或缩短正限位	关节正限位脉冲溢出	请设置伺服电子齿轮比或缩短正限位
29122	[RC&Sg]关节负限位脉冲溢出，请设置伺服电子齿轮比或缩短负限位	关节负限位脉冲溢出	请设置伺服电子齿轮比或缩短负限位

报警代码	报警内容	报警原因	解决措施
29123	外部轴伺服版本无法适配机器人控制系统, 请联系厂家更换伺服驱动器	配置的外部轴驱动器型号错误	更换为厂家支持的驱动器
29124	ServoParam 文件缺少 TorqP 或 TorqN 对象, 请增加地址后重启系统	缺少 TorqP 或 TorqN 对象地址	在 ServoParam.ini 文件中增加 TorqP 或 TorqN 的对象地址
29131	ADP 库文件缓存定义与系统不匹配, 需升级库文件	系统安全校验	升级 ADP 库至相匹配版本
29135	OVR 库文件缓存定义与系统不匹配, 需升级库文件		升级 OVR 库至相匹配版本
29141	CUB 控制轴数不匹配, 需升级库文件		升级 CUB 库至相匹配版本
29142	CUB 函数个数不匹配, 需升级库文件		
29143	CUB 函数的形参个数不匹配, 需升级库文件		
29144	CUB 结构体变量类型个数不匹配, 需升级库文件		
29145	CUB 结构体的大小不匹配, 需升级库文件		
29146	CUB 枚举变量类型个数不匹配, 需升级库文件		
29147	CUB 枚举的元素个数不匹配, 需升级库文件		
29200	EtherCAT 主站配置错误或 bootfile 缺少或 bootfile 不匹配	EtherCAT 主站配置错误或 bootfile 缺少或 bootfile 不匹配	检查主站配置及 bootfile 文件
50001	设置系统模式的参数异常	-	-
50002	设置系统模式时, 程序由执行转到停止失败	系统内部错误	联系厂家
50003	获取系统中运动相关配置参数失败	-	-
50004	程序执行中, 不允许伺服励磁操作	程序运行中, 系统不响应伺服的上下励磁操作	请先让系统退出运行状态后, 再进行伺服的励磁操作
50005	系统不在自动模式, 不允许伺服励磁操作	系统不处于自动模式时, 系统不响应伺服的上下励磁操作	请将系统打到自动模式后, 再进行伺服的励磁操作
50006	系统下发运动控制复位命令失败	-	-
50007	系统下发指令处理复位命令失败	-	-
50008	系统清理错误列表失败	-	-
50009	系统错误清空复位命令执行失败	-	-
50010	获取到系统远程速度参数异常, 超过[1,100]	从文件中读取的自启动远程速度的取值范围时 1~100,超过该范围均会报错。	在配置界面重新将该速度设为 1~100 区间的值
50011	设置系统远程速度参数异常, 超过 [1,100]	可能的原因: 1. 设置的自启动远程速度的取值范围时 1~100 2. 远程模式下设置自启动远程速度也会报该警	1. 根据 1~100 区间去设置该速度值。 2. 切换到非远程模式后再进行设置
50012	设置自启动中工程或程序名存在空值	-	-
50013	取消自启动中工程或程序名与原本不对应	这个报警一般在示教器未重启, 控制器被替换了新的 runtime 包后再启动, 当与示教器重新连接后直接在工程界面进行取消自启	示教器与控制器同步重启时不会报该警, 报警后可点击工程界面下方的刷新按钮再重试该操作

报警代码	报警内容	报警原因	解决措施
		动操作会报该警，这个时候刷新界面再进行取消自启动操作即可。	
50014	程序执行中，不允许设置系统 IP	系统处于运行状态下，不响应设置系统 IP 命令	请先让系统退出运行状态后，再进行设置系统 IP 操作
50015	设置系统 IP 参数中端口号值异常	控制器端口号或者 RTOS 的端口号必须为正数	检查输入的控制器端口号和 RTOS 的端口号是否为大于等于 0 的正数
50016	加密信息不合法，加密失败	进行加密操作时，系统会对下发的每一级天数进行校验，输入的天数必须为数字并且只能为小于 1000 的数	确认输入的天数是合法的
50017	解密密码错误，解密失败	可能的原因： 1. 密码输错了 2. 厂家给的密码不对	确认有误输错密码，确认无错后可联系厂家确认密码给的是否正确
50018	加密数据被其他线程占用，请稍后重试	系统内部错误	联系厂家
50019	系统当前没有加密，无需解密	系统只有处于加密状态下，才需要解密，默认出厂是无加密状态的。该报警属于提示信息，而非报警信息	-
50020	系统执行期限已过，请联系厂家进行解密	加密后允许使用的时间到期会提示这个报警，此时需要联系厂家进行解密操作	联系厂家
50021	系统时间被篡改，运行不合法	处于加密状态时，系统会实时记录时间的计时状态，一旦发现时间往前改则会报错。	请改成原来的时间或者在当前时间重新进行加密操作。
50022	系统时间获取失败，操作频繁，请重试	未获取到系统时间，加密参数获取失败。	联系厂家
50023	程序由暂停转到停止状态出错	-	-
50024	多工位参数设置失败，参数不合法	四个使能开启的工位之间的启动输入口、停止输入口、启动输出口、停止输出口只能唯一且不能重复，并且同一工位的启动/停止输入口（输出口）之间也不能重复	确认输入的输入口/输出口端口的有效性
50025	设置远程速度失败，请切换到其它模式下设置	远程模式下，系统不响应设置远程启动速度命令	请切换到手动或者自动模式再进行速度设置操作。
50026	被使能的多工位输入口/输出口/程序名不能为空或无效值	每次由手动/自动切换到多工位模式时，为确保能够正常运行，系统会挨个检测一遍被使能的工位参数，确保这些参数的值为有效值，否则将报错。	在多工位参数配置界面确认输入的输入口/输出口端口的以及程序名的有效性
50027	更新加密数据到文件中失败	文件写入、删除、复制失败三种都有可能报该警	联系厂家
50028	设置系统模式时，程序倒序切换到单步失败	切换系统模式时，将程序切换到单步模式失败，可能的原因如下 1. 当前处于运行模式，切换模式失败。 2. 当前处于暂停模式，停止程序失败。 3. 倒序模式下系统并非转入手动模式，切换失败。	
50029	该系统不支持设置外部通讯网络信息	系统底层配置不支持该通讯方式	联系 ESTUN 技术人员

报警代码	报警内容	报警原因	解决措施
50030	非静止状态下禁止设置 API 模式	API 模式只能在系统处于 stop 状态下才可设置。	自行确认系统状态
51001	程序执行中，不允许进行回零操作	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作。	请先让系统退出运行状态后再执行操作
51002	伺服不在励磁状态，不允许回零操作	伺服未上励磁时，机器人不响应任何运动命令。	请上励磁后再进行回零操作
51003	回零命令执行失败	回零命令下发失败，内核未接收回零命令	联系厂家
51004	全局速度命令设置失败	全局速度命令下发失败，内核未接收该命令	
51005	程序运行中，不允许设置点动参考坐标系	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作	请先让系统退出运行状态后再执行操作
51006	点动坐标系命令设置失败	点动坐标系命令设置失败，内核未接收该命令	联系厂家
51007	程序运行中，不允许机器人点动	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作	请先让系统退出运行状态后再执行操作
51008	伺服不在励磁状态，不允许机器人点动	伺服未上励磁时，机器人不响应任何运动命令	请上励磁后再进行点动操作
51009	自动模式下，不允许机器人点动	只有手动模式下，才可对机器人进行点动操作	切换到手动模式，再对机器人进行点动操作
51010	当前系统模式下，不允许机器人点动操作		
51011	程序运行中，不允许切换点动模式	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作。	请先让系统退出运行状态后再执行操作
51012	点动命令执行失败	1. 点动方式只能为点动(0)或者寸动(1)，下发其他的值则会报该警 2. 传入点动的轴号只能为 0~15 之间，不符合范围内则报警。 3. 点动命令设置失败，内核未接收该命令	联系厂家
51013	设置 DOUT 命令执行失败	待设置的 io 口端口号超过了设定的范围，要么为负值，要么超过了最大限制值	确认设置的端口号在有效范围内
51014	设置 AOUT 命令执行失败		
51015	设置 SIMDOUT 命令执行失败		
51016	设置 SIMAOUT 命令执行失败		
51017	系统接口异常，重启系统后再尝试	未能成功与内核建立共享缓存，可能的原因是内核组件库没有启动起来或者启动太慢了。	断电重启
51018	命令参数超限定范围，命令执行失败	待设置的伺服轴编号必须为 0~63 以内的值	请检查输入的伺服轴号是否超限
51019	程序运行中，不允许标定工具参数	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作。	请先让系统退出运行状态后再执行操作
51020	程序运行中，不允许标定用户坐标系	-	-
51021	设置传送带参数命令执行失败	设置传送带参数命令下发失败，内核未接收该命令	联系厂家
51022	设置跟随参数命令执行失败	设置跟随参数命令下发失败，内核未接收该命令	
51023	刷新单圈值失败	刷新单圈值命令下发失败，内核	

报警代码	报警内容	报警原因	解决措施
		未接收该命令	
51024	映射运动内核接口数据失败，检查运动内核组件是否正常启动	指令系统未能在规定时间内与内核数据建立链接，可确认内核组件是否正常启动	1. 断电重启 2. 复现多次时，请联系厂家
51025	起终点跨奇异，不支持一点到达运动	scara 当前位置的 mode 属性与想要拱形运动到的位置的 mode 属性不一致，用 movl 运动不过去	请先将机器人移动到与目标位置处于同一个 mode 位置状态。
51026	设置的高度值不合法	设置的高度不能为负值，并且抬升以后的高度必须要高于目标点的高度	请确认抬升高度值的有效性
51027	当前机型不支持一点到达操作(拱形运动)	拱形运动只支持 SCARA 机型	
51028	当前模式不支持一点到达操作(拱形运动)	拱形运动只支持手动模式，其他模式下不允许运行	请切换到手动模式后再进行拱形运动
51029	运动启动失败	可能的原因： 1. 系统当前有轴处于未回零状态。 2. 系统处于错误状态。	1. 可通过坐标系界面观察轴的坐标，看是否有轴处于未回零状态，如果有则先回零后再执行拱形运动 2. 确认系统有无报警，有报警清除报警后再进行拱形运动 如果上述两种方式都确认无误还是不能报该错，请联系厂家
51030	运动缓存已满		
51031	程序运行中，不允许激活一点对正	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作。	请先让系统退出运行状态后再执行操作
51032	程序运行中，不允许进行一点对正操作		
51033	一点对正操作失败	一点对正命令下发失败，内核未接收该命令	联系厂家
51034	未激活一点对正，不允许操作	未激活一点对正功能时，一点对正操作被禁止	使用前请先激活一点对正功能
51035	程序暂停状态中，不允许进行回零操作	暂停状态下，禁止操作。	请将系统切换到停止状态后再执行操作
51036	程序暂停状态中，不允许设置点动参考坐标系		
51037	程序暂停状态中，不允许机器人点动		
51038	程序暂停状态中，不允许切换点动模式		
51039	程序暂停状态中，不允许标定工具参数		
51040	程序暂停状态中，不允许标定用户坐标系		
51041	程序暂停状态中，不允许激活一点对正		
51042	程序暂停状态中，不允许进行一点对正操作		
51043	设置 DI 命令执行失败	待设置的 io 口端口号超过了设定的范围，要么为负值，要么超过了最大限制值	确认设置的端口号在有效范围内
51044	设置 AI 命令执行失败		
51045	设置 SIMD I 命令执行失败	当前版本不支持设置 SIMD I 端口的值	-
51046	设置 SIMAI 命令执行失败	当前版本不支持设置 SIMAI 端口的值	-
51047	伺服励磁状态，不允许清除伺服报	伺服处于励磁状态下，系统不响	请先让伺服下励磁后再执行该操作

报警代码	报警内容	报警原因	解决措施
	警	应清除伺服报警命令	
51048	清除伺服报警或伺服历史报警失败	清除伺服报警命令下发失败，内核未接收该命令	联系厂家
51049	获取伺服 Pn 参数失败	获取伺服 Pn 参数命令下发失败，内核未接收该命令	
51050	设置伺服 Pn 参数失败	设置伺服 Pn 参数命令下发失败，内核未接收该命令	
51051	软浮动期间，不允许机器人点动	软浮动状态下，系统不响应点动命令。	请先退出软浮动状态后再进行点动操作
51052	内核由暂停转空闲状态失败	系统下发暂停命令后未在规定时间内转到空闲状态，那内核有可能是没有接收到命令、也可能是超时后转到了空闲状态	联系厂家
51053	设置诊断功能项失败	设置诊断功能项命令下发失败，内核未接收该命令	
51054	设置诊断参数失败	设置诊断参数命令下发失败，内核未接收该命令	
51055	程序运行中，不允许标定外部坐标系	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作	请先让系统退出运行状态后再执行操作
51056	程序暂停状态中，不允许标定外部坐标系	暂停状态下，禁止标定、切换点动模式、回零、设置点动坐标系、机器人点动等操作	请将系统切换到停止状态后再执行操作
51057	映射运动内核接口数据错误，版本不匹配	指令系统库与内核库接口不一致时，会报该警，此时应确认系统版本号	1. 确认版本号，判断替换的版本是否有问题 2. 无法确认时，请联系厂家
51058	变位机参数设置失败,请检查输入参数的合法性	可能的原因： 1. 传入的参数不合法，参数的详细报警可查看同时报出来的其他报警 2. 设置变位机参数命令下发失败，内核未接收该命令	根据同时提示的其他报警来合理输入参数
51059	变位机参数读取失败，请检查输入参数的合法性	可能的原因： 1. 传入的参数不合法，参数的详细报警可查看同时报出来的其他报警 2. 读取变位机参数命令下发失败，内核未接收该命令	根据同时提示的其他报警来合理输入参数
51060	程序运行中，不允许标定变位机坐标系	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作	请先让系统退出运行状态后再执行操作
51061	程序暂停状态中，不允许标定变位机坐标系	暂停状态下，禁止标定、切换点动模式、回零、设置点动坐标系、机器人点动等操作。	请将系统切换到停止状态后再执行操作
51062	程序运行中，不允许设置变位机坐标系参数	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作	请先让系统退出运行状态后再执行操作
51063	第一个变位机轴不在零点位置！标定失败	标定第二个变位机坐标系时，第一个变位机轴的坐标系必须在零点位置，否则会导致标定结果有较大误差	将第一个变位机轴移动到零点位置后再进行标定操作。
51064	第二个变位机轴不在零点位置！标定失败	标定第一个变位机坐标系时，第二个变位机轴的坐标系必须在零点位置，否则会导致标定结果有	将第二个变位机轴移动到零点位置后再进行标定操作。

报警代码	报警内容	报警原因	解决措施
		较大误差	
51065	激活或冻结区域命令失败	激活或冻结区域命令下发失败，内核未接收该命令	联系厂家
51066	传入的传送带 ID 错误	系统目前只支持两个传送带，传入的传送带 id 只能为 1 或者 2，其他值则报错	确认输入的传送带 id 值的有效性
51067	程序运行中，不允许标定传送带脉冲当量	当系统处于运行状态时，系统只支持暂停、急停、设置速度倍率、修改 IO 值这些设置操作	请先让系统退出运行状态后再执行操作
51068	程序暂停状态中，不允许标定传送带脉冲当量	暂停状态下，禁止标定、切换点动模式、回零、设置点动坐标系、机器人点动等操作	请将系统切换到停止状态后再执行操作
51069	当前点与上一次记录的点相同	传送带脉冲当量标定时，第一个点与第二个点的坐标值不能一致，否则算出来的值无意义	请正确的按照说明书中脉冲当量的标定操作来使用
51070	计算失败！Z 轴坐标必须相等	两个标定点的脉冲值不能一样，标定时请确保传送带可以运动	请正确的按照说明书中脉冲当量的标定操作来使用
51071	传入的动坐标系未与传送带绑定，运行失败	指令中使用的动坐标系必须与传送带配置界面中设置的动坐标系一致，否则运行时系统不知道当前使用的是哪一个传送带	1. 在指令中选择在传送带配置界面设置的动坐标系 2. 将传送带配置界面中配置的动坐标系更改为指令中使用的动坐标系，但要确保该动坐标系是针对该传送带标定的坐标系
51072	设置的传送带输出口超过设定范围	设置触发输出口有效值域为 9~31，超过该值域则会报错	检验设置的输出端口的正确性
51073	设置的传送带输入口超过设定范围	设置触发输入口有效值域为 9~31，超过该值域则会报错	检验设置的输入端口的正确性
51074	设置的传送带输出口跟其他传送带的输出口重复	每个启用的传送带输出口号只能唯一，不可共用	选择其他未被占用的输出口
51075	设置的传送带输入口跟其他传送带的输入口重复	每个启用的传送带输入端口号只能唯一，不可共用	选择其他未被占用的输入口
51076	未获取到视觉点信息，运行失败	机器人系统开始传送带同步跟随运动之前，需要确保视觉已经给机器人发送了工件位置信息，否则将无法跟随	1. 通过传送带跟随配置界面的网络监控，观察系统能否接收到视觉的位置信息数据。如果不能接收则对网口、端口号、触发输出口做进一步排查 2. 如果上一步正常，只是接收后程序还报错，那么可以加大 WaitWObj 指令中的等待时间
51077	设置的传送带坐标系跟其他传送带的动坐标系重复	每个启用的传送带动坐标系只能唯一，不可共用	选择其他未被占用的动坐标系
51078	动坐标系下，不允许机器人点动	当系统的用户坐标系被设为动坐标系时，系统不响应点动、切换手动坐标系、标定脉冲当量、设置碰撞参数、设置振动抑制参数等命令	请先退出动坐标系后再进行相关操作，可通过位置界面中选择其他类型的用户坐标系来退出动坐标系。
51079	动坐标系下，不允许切换坐标系		
51080	动坐标系下，禁止标定传送带的脉冲当量		
51081	动坐标系下，禁止设置碰撞参数		
51082	动坐标系下，禁止设置振动抑制参数		
51083	计算失败！两个点之间的距离太近	标定脉冲当量时，两个标定点之间的距离需要在合理范围内，太大太小都会导致算出来的值不准确	扩大两个标定之间的距离

报警代码	报警内容	报警原因	解决措施
51084	恢复参数完成！但是，以下参数无法恢复：	有一些伺服参数不能恢复；	确认要恢复的伺服参数文件，查看伺服参数值是否正确合理；确认伺服是否开放伺服参数操作；
51085	设置机械臂参数失败	命令下发失败，内核未接收该命令	联系厂家
51086	更改轨迹纠偏使能状态失败		
51087	设置轨迹纠偏参数失败		
51088	设置区域参数失败		
51089	备份参数失败		
51090	恢复参数失败		
51091	当前动坐标系与待设置参数的传送带动坐标系不一致	-	-
51092	当前操作只能在动坐标系下进行	-	-
51093	设置多边体区域参数失败	命令下发失败，内核未接收该命令	联系厂家
51094	获取多边体区域参数失败		
51095	删除多边体区域参数失败		
51096	激活或冻结多边体区域命令失败		
51097	一键清除 IO 输入量仿真状态失败	命令下发失败，示教器未接收到该命令	联系厂家
51098	获取 IO 注释失败	ioNote.ini 文件缺失	
51099	设置 IO 注释失败	ioNote.ini 文件缺失	
51100	虚拟示教器急停命令失败	-	
51101	设置更新伺服监测数据命令失败	-	
51102	获取伺服历史报警命令失败	-	
51103	文件上传命令失败	-	
51104	文件下载命令失败	-	
52001	系统处于运行状态，操作被禁止	系统在运行状态下，一些操作被禁止执行	操作是否为运行状态下，禁止执行的操作
52002	请先加载工程或程序	在未加载工程或程序时，一些操作被禁止	操作是否为未加载工程或程序状态下，禁止执行的操作
52003	工程创建失败	-	-
52004	工程加载失败	-	-
52005	工程注销失败	程序运行时无法注销、转停止状态失败、转单步模式失败	1. 程序是否正在运行； 2. 是否为停止状态； 3. 是否在单步模式；
52006	工程复制失败	-	根据报警信息检查操作是否合法
52007	工程粘贴失败	-	
52008	工程删除失败	-	
52009	工程重命名失败	-	
52010	工程刷新失败	-	
52011	程序创建失败	-	
52012	程序加载失败	重置（内核状态、工具、坐标系、负载，其中某项）失败，SCARA 机型也可能是设置左右	联系厂家

报警代码	报警内容	报警原因	解决措施
		手失败	
52013	程序注销失败	重置（内核状态、工具、坐标系、负载，其中某项）失败，SCARA 机型也可能是设置左右手失败、转单步模式失败	1. 是否在单步模式 2. 联系厂家
52014	程序打开失败	-	根据报警信息检查操作是否合法
52015	程序关闭失败	-	
52016	程序复制失败	-	
52017	程序粘贴失败	-	
52018	程序被调用，删除失败	此程序被别的程序使用，如 CALL、RUN、KILL	检查其他程序，是否使用此程序
52019	程序被调用，重命名失败		
52020	程序刷新失败	-	根据报警信息检查操作是否合法
52021	程序列表获取失败	-	
52022	工程名过长或为空	工程或程序的操作（新建、打开、关闭、拷贝、粘贴、删除、重命名、刷新等）、获取程序的 MD5 值、程序指令操作（新建、修改、复制、剪切、删除、粘贴、注释、取消注释、撤销、获取内容、获取参数等）、变量操作（获取内容、删除、重命名等）等命令携带的工程名过长或为空	检查所做操作传递的工程名是否过长或为空
52023	工程名已存在	工程操作（新建、重命名、粘贴）时，使用的名称为已存在的工程名	检查工程名是否已存在
52024	工程数量超出限制	工程操作（新建、粘贴）时，超出了最大允许数量	检查工程个数是否超限制
52025	创建工程变量数据文件失败	-	根据报警信息检查操作是否合法
52026	创建工程后获取工程列表失败	-	
52027	工程文件异常	创建工程文件夹失败，因为是调用系统函数创建，可能是存在权限问题，如不是管理人员，可能无法创建此文件夹	联系厂家
52028	该工程不存在	工程或程序操作（加载、拷贝、粘贴、删除、重命名）时，传入的工程名为不存在工程	检查工程是否存在
52029	获取全局数据失败	工程下工程变量文件不存在或者太大，无法读取	检查工程变量文件是否存在或太大
52030	加载工程数据失败	工程下变量加载到 lua 内存失败	检查被加载的工程下的变量文件是否正常
52031	工程没有被加载	1、打开程序时，发现工程没加载 2、注销程序时，发现工程没加载 3、注销工程时，发现工程没加载	检查相关工程是否被加载
52032	非法的程序名	程序名不能为“prjglobal”	是否给程序起名为“prjglobal”
52033	程序名过长或为空	程序的操作（新建、打开、关闭、拷贝、粘贴、删除、重命名、刷新等）、获取程序的 MD5 值、程序指令操作（新建、修	检查所做操作传递的程序名是否过长或为空

报警代码	报警内容	报警原因	解决措施
		改、复制、剪切、删除、粘贴、注释、取消注释、撤销、获取内容、获取参数等）、变量操作（获取内容、删除、重命名等）等命令携带的程序名过长或为空	
52034	程序名已存在	程序操作（新建、重命名、粘贴）时，使用的名称为已存在的程序名	检查程序是否存在
52035	程序数量超出限制	程序操作（新建、粘贴）时，超出了最大允许数量	检查程序个数是否超限制
52036	创建程序变量文件失败	创建程序变量文件失败，因为是调用系统函数创建，可能是存在权限问题，如不是管理人员，可能无法创建文件	联系厂家
52037	创建程序文件失败	正常流程是新建一个文本，写入默认字符，读取这个文本；将写入前的字符串，MD5 处理，再将读取文本得到的字符串，MD5 处理，比较是否相同；因为都是调用固定的函数，所以只可能是写文件错误，或读文件错误，即可能有权问题	
52038	获取选定工具坐标系参数失败	工具坐标系域不正确、不存在此变量或变量字符串格式异常	检查工具坐标系变量是否正常（去文本中查看）
52039	获取选定用户坐标系参数失败	用户坐标系域不正确、不存在此变量或变量字符串格式异常	检查用户坐标系变量是否正常（去文本中查看）
52040	工程名或程序名为空	添加 AddDo 指令、操作（加载、调用、打开）程序时，发现工程名或程序名为空	检查相关工程和程序名是否为空
52041	程序文件无法打开	程序 erd 文件不存在或者存在但无法打开（调用 open 失败，可能存在系统权限问题，正常不会出现）	检查相关 erd 文件是否存在
52042	该程序没有在工程中找到或没有被加载	注销程序时，发现当前加载程序与要注销的程序不同，或者根本没有加载程序	检查当前加载程序与要注销的程序是否一致
52043	程序文件不存在	程序操作（删除、拷贝、读取（CALL）、打开、重命名）时，在工程下的程序列表中找到此程序	检查此程序是否存在（检查相关文件）
52044	删除数据文件失败	删除程序 erd 文件失败	1. 检查 erd 文件是否存在 2. 若存在，尝试手动删除（只读文件无法删除）
52045	删除程序文件失败	删除程序 erp 文件失败	1. 检查 erp 文件是否存在； 2. 若存在，尝试手动删除（只读文件无法删除）
52046	获取程序列表失败	虽然此错误码被使用，但是不会触发	
52047	无法删除当前加载的工程	工程在加载状态，不能删除	检查工程是否为加载状态
52048	从程序文件中获取全部指令失败	-	根据报警信息检查操作是否合法
52049	程序文件格式转换失败	-	
52050	当前打开的程序无法关闭	-	
52051	程序数据文件大小超出限制	程序 erd 文件大小超限制	检查 erd 文件是否太大（500K 以

报警代码	报警内容	报警原因	解决措施
			上)
52052	恢复运行的数据失败	系统原因	联系厂家
52053	当前加载程序的程序变量数据获取失败	系统原因	联系厂家
52054	当前加载程序的程序变量数据加载失败	程序 erd 文件格式错误 (加载到缓存 t_x 域)	检查 erd 文件, 是否正常
52055	当前加载程序的程序变量数据保存失败	程序 erd 文件格式错误 (加载到缓存 t_xb 域)	
52056	没有被复制的程序, 程序粘贴失败	粘贴程序前, 要先执行复制操作	先复制, 再粘贴
52057	没有被复制的工程, 工程粘贴失败	粘贴工程前, 要先执行复制操作	
52058	要复制的工程不存在, 工程粘贴失败	要复制的工程不存在, 不能粘贴	检查工程是否存在
52059	工程文件目录获取失败	系统原因	联系厂家
52060	工程被加载、程序被加载或打开时, 不允许重命名操作	加载的工程, 不允许重命名	检查工程是否是加载状态
52061	程序数据文件重命名失败	调用系统函数 rename 失败, 可能有权限问题, 正常不会出现	手动重命名 erd 文件验证原因
52062	程序指令文件重命名失败	调用系统函数 rename 失败, 可能有权限问题, 正常不会出现	手动重命名 erp 文件验证原因
52063	工程文件目录拷贝失败	拷贝工程文件夹失败, 调用系统函数, 可能有权限问题, 正常不会失败	手动拷贝文件夹验证
52064	工程文件目录删除失败	删除工程文件夹, 调用系统函数, 可能有权限问题, 正常不会失败	手动删除文件夹验证
52065	拷贝工程目录失败	同 52063	手动拷贝文件夹验证
52066	工程加载状态下不允许删除工程中的程序	工程加载状态下不允许删除工程中的程序	检查工程是否是加载状态
52067	工程加载状态下不允许重命名工程中的程序	工程加载状态下不允许重命名工程中的程序	检查工程是否是加载状态
52068	变量名长度超限	暂时不会触发 (平移镜像中使用)	-
52069	IF 命令携带设置坐标系或工具参数	暂时不会触发 (平移镜像中使用)	-
52070	程序行数超限	暂时不会触发 (平移镜像中使用)	-
52071	打开程序文本失败	暂时不会触发 (平移镜像中使用)	-
52072	平移命令下发错误	暂时不会触发 (平移镜像中使用)	-
52073	平移命令获取结果超时	暂时不会触发 (平移镜像中使用)	-
52074	镜像命令下发错误	暂时不会触发 (平移镜像中使用)	-
52075	镜像命令获取结果超时	暂时不会触发 (平移镜像中使用)	-
52076	更新变量管理表失败	暂时不会触发, 控制器接收更新变量 (某工程或全部) 命令后, 重新生成变量管理文件, 以应对手动拷贝的工程文件, 但是现在没有使用此命令	-

报警代码	报警内容	报警原因	解决措施
52077	操作失败！系统初始化未完成，请稍后重试	程序启动失败	联系厂家
52078	获取选定外部 TCP 坐标系参数失败	外部 TCP 坐标系域不正确、不存在此变量或变量字符串格式异常	检查外部 TCP 坐标系变量是否正常（去文本中查看）
52079	获取选定变位机坐标系参数失败	变位机坐标系域不正确、不存在此变量或变量字符串格式异常	检查变位机坐标系变量是否正常（去文本中查看）
52080	获取选定负载惯量参数失败	负载惯量域不正确、不存在此变量或变量字符串格式异常	检查负载惯量变量是否正常（去文本中查看）
52081	传入的 label 名不存在，获取 label 所在行失败	程序中找不到相关 label	手动检查程序中是否存在此 label
52082	程序删除成功	程序删除成功提示	仅在删除完成时，起到提示作用
52083	工程删除成功	工程删除成功提示	仅在删除完成时，起到提示作用
52084	API 模式下不支持该操作	API 模式下，系统的运动由外部控制，部分命令操作将被禁止。	退出 API 模式后再进行相关操作。
52085	全局工程不支持该操作	全局工程不支持删除、重命名、复制以及新建工程变量等操作	-
52086	工程文件处于写保护状态	程序工程中存在程序文件是写保护状态，无法重命名和删除工程	只有对程序文件取消写保护状态，方可对程序文件进行更改
52087	程序文件处于写保护状态	程序文件处于写保护状态，无法对程序文件做出任何更改操作	
52088	变量文件处于写保护状态	当程序文件处于写保护状态，该程序文件所涉及的全局变量、工程变量和程序变量都不能进行任何更改操作	
52089	设置程序文件为只读权限失败	程序文件的权限并没有被成功设置为只读	联系厂家
52090	设置程序文件为读写权限失败	程序文件的权限并没有被成功设置为读写	
52091	程序注释超过了最大长度	-	-
52301	枚举字符串未找到	获取指令中 enum 类型字符串失败	检查指令文件 configTable.cfg 中相关指令中"_enum"参数
52302	指令校验不合法	-	-
52303	获取完整的指令内容失败	-	-
52304	获取新建的指令列表失败	新建指令时，获取其参数失败	检查指令文件 configTable.cfg 中相关指令参数
52305	获取修改的指令列表失败	系统原因	联系厂家
52306	获取新建的小指令列表失败		
52307	获取修改的小指令列表失败		
52308	小功能指令创建失败	新建指令时，获取其 AddDo 参数失败（参数名，参数类型或参数内容空或长度太长）	检查指令文件 configTable.cfg 中相关指令参数
52309	指令创建失败	系统原因	联系厂家
52310	指令修改失败		
52311	指令复制失败		
52312	指令剪切失败		
52313	指令粘贴失败		
52314	指令删除失败		
52315	指令注释失败	可能的原因：	-

报警代码	报警内容	报警原因	解决措施
		1. 指令的工程名、程序名空或者与当前打开（加载）不同 2. 当前没有打开（加载）的程序 3. 操作成功后写文件时出错	
52316	指令恢复失败	系统原因	联系厂家
52317	当前行号超过了最大数量限制	当前命令给出的行号，超过了正常的行号，例如当前程序 20 行，但是新建一行指令给出的行号（理论上 ≤ 20 ）是 100，就报错了	联系厂家
52318	变量数据超过了最大限制	新建或修改指令时，给出的某个参数内容过大，如 IF 指令给出的判断字符串长度大于 400	检查指令参数内容是否过长
52319	不支持的指令类型	传入的指令名称（MovJ、MovL 等）空或者长度太大	检查指令名称长度
52320	指令参数缺失，添加 AddDo 指令失败	传入的 AddDo 参数项不存在，如 SetDO 命令正常 2 个参数（端口变量、端口值），但是给了第 3 个变量（端口类型），此时报错	检查 AddDo 参数是否正确
52321	指令转译失败	指令新建或修改等操作时，指令内容为空，或者转换后（P:转 t.p.）指令长度超最大值（400）	检查指令长度
52322	指令校验时获取全部指令失败	错误描述笼统，覆盖了更详细的错误描述	-
52323	指令 MD5 校验失败	指令内容长度超过最大值（410），正常不会触发	检查指令长度
52324	工程名或程序名错误	指令操作（插入、修改、删除、拷贝、剪切、粘贴、取消注释、获取内容）时，给出的工程名或程序名为空；或者与当前打开（加载）的名称不同；或者当前就没有打开（加载）程序	检查操作的工程名与程序名
52325	程序为空，或结束行号大于起始行号	指令操作（拷贝、删除、剪切、粘贴、注释、取消注释）时，当前程序指令为空；或者给出的结束行大于起始行；或者起始行为 0	1. 检查当前程序是否正常 2. 联系厂家
52326	存在特殊字符或不闭合的语句，指令批量操作失败	IF 与 ENDIF、WHILE 与 ENDWHILE 不匹配，或者存在多个 START:、END；特殊字符行	检查程序是否有问题
52327	更新程序文件失败	程序内容变更后，更新到文件失败	联系厂家
52328	无法删除当前加载的程序	-	-
52329	程序指令数量超出最大限制	指令操作（新建，粘贴）时，使程序行超过最大行数（1000）限制	检查行数是否超限
52330	内存插入指令操作失败	插入的指令为空，逻辑上不会触发	-
52331	指令文件打开错误	程序 erp 文件不存在	检查 erp 文件是否存在
52332	指令行数超出最大限制	程序 erp 文件内容超过最大行（1000）限制	检查 erp 文件行数是否超限
52333	指令文件为空	程序 erp 文件为空	检查 erp 文件是否为空
52334	指令文件损坏，系统已从备份文件	系统原因	联系厂家

报警代码	报警内容	报警原因	解决措施
	恢复该文件		
52335	单行指令长度超出限制	指令新建或修改等操作时，转换前或转换后（P:转 t.p.），指令长度超最大值（400）	检查指令长度是否超限
52336	获取工程下的程序文件列表失败	系统原因	联系厂家
52337	指令粘贴失败，没有被复制或剪切的指令	没有执行复制或剪切，无法粘贴	先执行复制或剪切，再粘贴
52338	没有待粘贴的指令，指令粘贴失败	系统原因	联系厂家
52339	撤销失败	撤销上一步失败	联系厂家
52340	没有可撤销的操作	没有执行可撤销的操作（删除、修改、剪切、粘贴、注释、取消注释）	检查是否有可撤销操作
52341	撤销文件名错误	当前打开（加载）的程序，跟撤销命令传过来的不一致	联系厂家
52342	触发操作不能为空	OnDistance 与 Onparameter 指令触发操作不能为空	检查指令触发操作是否为空
52343	操作失败！程序文件与指令系统版本不匹配，两者分别为：	加载程序时，检测到程序文件版本与当前不一致	检查程序文件，查看其版本是否正确
52344	操作的行号超过当前的程序总数，请刷新界面后重试	程序操作（拷贝、删除、剪切、注释、取消注释）时，给出的行号大于程序现在的总行数	联系厂家
52345	反向粘贴不能包含 IF 或 WHILE 指令	-	-
52346	IF 逻辑内 ELSIF 或 ELSE 不匹配	IF 指令中 ELSIF 或 ELSE 指令逻辑错乱	联系厂家
52347	FOR 指令内没有与之匹配的 IF 指令	在 IF 指令中，FOR 指令存在逻辑错乱问题	
52348	WHILE 指令内没有与之匹配的 IF 指令	在 IF 指令中，WHILE 指令存在逻辑错乱问题	
52501	获取所有的变量值域失败	系统原因	联系厂家
52502	变量列表获取失败	系统原因	联系厂家
52503	变量名为空或超过最大长度	变量名不合法，为空或纯数字或超过最大长度	检查变量是否合法
52504	变量作用域错误	变量操作（新建、修改、删除、重命名）时，给出的作用域不正确（不是 1,2,3） 新建变量给出的作用域里找不到变量类型，也会报此错误。如：给出的作用域为程序域：3，但是类型是用户坐标系，这时报错	联系厂家
52505	变量名已存在	变量操作（新建、重命名）时，给出的名字已存在	查找是否有同名变量
52506	变量创建失败	新建变量，工程名异常（空、超过最大长度、与当前加载不同）时，报此错误 区域变量个数多于 32 个，也会报此错误	1.检查区域变量是否过多 2. 联系厂家
52507	变量删除失败	删除未激活的区域变量失败，下发命令未成功，可能是队列满，也可能是队列被锁住	联系厂家
52508	变量修改失败	区域变量：不存在或者字符串转换结构体失败；	

报警代码	报警内容	报警原因	解决措施
		DI 与 SimDI: 字符串内容不正确; 其他: 修改时工程名异常 (空、超过最大长度、与当前加载不同) 时, 报此错误	
52509	变量值获取失败	变量域错误, 非 0-12 的值 (系统、全局、工程、程序、AddDo、RUN1-RUN8)	
52510	变量名获取失败	-	-
52511	变量未找到	获取变量类型失败	联系厂家
52512	变量类型不合法	根据给定的变量名找到的变量的类型, 与给定的类型不一致	
52513	示教点坐标获取失败	系统原因	联系厂家
52514	程序域变量掉电保存属性不允许修改	程序域变量不能修改掉电保存属性, 但是在判断过程中, 获取变量失败, 也会报此错误	检查是否更新了程序变量的掉电保存属性
52515	安全区域不是全局变量或者变量不存在	系统原因	联系厂家
52516	变量新建写入文件失败		
52517	变量修改写入文件失败		
52518	变量示教写入文件失败		
52519	变量值不合法或超过最大长度	变量值 (字符串形式) 超过最大长度 400	检查变量值内容是否过长
52520	PLC 变量索引号不能相同	PLC 变量索引号不能相同	检查是否存在相同索引的 PLC 变量
52521	PLC 变量已达最大数量	PLC 变量最多支持 150 个	检查 PLC 变量个数
52522	PLC 变量正在被其他应用占用	PLC 变量正在被其他应用占用	可稍后尝试
52523	变量重命名失败	-	-
52524	系统变量不允许修改	系统变量不允许修改	查看变量是否为系统变量
52525	系统变量不允许删除	系统变量不允许删除	
52526	系统变量不允许重命名	系统变量不允许重命名	
52527	全局变量内部处理错误	缓存中的全局变量结构 (表) 转换为字符串时, 找不到类型	查看变量文件, 检查相关变量是否正常
52528	工程变量内部处理错误	缓存中的工程变量结构 (表) 转换为字符串时, 找不到类型	
52529	程序变量内部处理错误	缓存中的程序变量结构 (表) 转换为字符串时, 找不到类型	
52530	Run/Kill 数据内部处理错误	缓存中的 RUN 程序变量结构 (表) 转换为字符串时, 找不到类型	
52531	删除全局变量执行失败	-	-
52532	删除工程变量执行失败	-	-
52533	删除程序变量执行失败	-	-
52534	安全区域变量数据获取失败	-	-
52535	安全区域变量不存在或者不是全局变量	安全区域变量不存在或者不是全局变量	查看全局变量文件, 检查相关变量是否存在
52536	系统变量不允许删除	-	-

报警代码	报警内容	报警原因	解决措施
52537	当前加载的工程或程序与变量所在工程或程序不匹配	当前加载的工程与变量所在工程不匹配	联系厂家
52538	没有找到该用户坐标系变量		
52539	没有找到该工具坐标系变量		
52540	该变量在程序指令中已经被引用	正在使用的变量不允许删除或重命名	检查变量是否在程序中使用
52541	系统变量不允许创建	系统变量不允许创建	是否在新建系统变量
52542	变量所在程序与当前打开的程序不一致	变量操作（新建、修改、获取、删除、重命名）时，当前打开的程序与变量所在程序不一致，或者当前加载的工程与变量所在的工程不一致，或者给定的工程名或程序名为空或大于最大长度	联系厂家
52543	程序变量文本打开失败	程序 erd 文本不存在或者太大（大于 500K）	检查相关文本
52544	禁止对已被设为系统工具坐标系的 tool 参数进行修改/删除/重命名操作	已被设为系统工具坐标系的 tool 参数，不能进行修改/删除/重命名操作	检查是否已被设为系统工具坐标系
52545	禁止对已被设为系统用户坐标系的 user 参数进行修改/删除/重命名操作	已被设为系统用户坐标系的 user 参数，不能进行修改/删除/重命名操作	检查是否已被设为系统用户坐标系
52546	禁止对已被设为系统工件负载的 payload 参数进行修改/删除/重命名操作	已被设为系统工件负载的 payload 参数，不能进行修改/删除/重命名操作	检查是否已被设为系统工件负载
52547	当前域变量总数达到上限	当前域变量个数多于 1500	检查当前域变量个数
52548	变量创建失败,个数超出	-	-
52549	禁止对使用中的 extTcp 参数进行修改/删除/重命名操作	使用中的 extTcp 参数，不能进行修改/删除/重命名操作	检查是否在使用中
52550	禁止对使用中的变位机参数进行修改/删除/重命名操作	使用中的变位机参数，不能进行修改/删除/重命名操作	
52551	修改 t_x 域变量失败	修改缓存中变量失败	联系厂家
52552	修改 t_xb 域变量失败	修改缓存中变量副本失败	
52553	修改临时域变量失败	修改变量时，为提高效率，会新建临时域（只有一个变量），生成字符串，在临时域缓存中新建变量失败	
52554	临时域变量转字符串失败	临时域变量转字符串失败	
52555	更新变量文本失败	更新变量文件（erd）失败	可能存在权限问题，尝试手动修改相关文件是否成功
52556	禁止对已设为默认工具坐标系的 tool 参数进行删除/重命名操作	已设为默认工具坐标系的 tool 参数，不能进行删除/重命名操作	检查是否设为默认工具坐标系
52557	禁止对已设为默认用户坐标系的 user 参数进行删除/重命名操作	已设为默认用户坐标系的 user 参数，不能进行删除/重命名操作	检查是否设为默认用户坐标系
52558	禁止对已设为默认工件负载的 payload 参数进行删除/重命名操作	已设为默认工件负载的 payload 参数，不能进行删除/重命名操作	检查是否设为默认工件负载
52559	此区域变量处于激活状态，请冻结区域后再删除此变量	区域变量被激活时，无法删除	检查区域变量是否为激活状态
52560	禁止对已被设为动坐标系的 syncoord 参数进行修改/删除/重命名操作	已被设为动坐标系的 syncoord 参数，不能进行修改/删除/重命名操作	检查是否设为动坐标系
52561	多边形区域变量数据获取失败	-	-

报警代码	报警内容	报警原因	解决措施
52562	多边体区域变量不存在或者不是全局变量	多边体区域变量不存在或者不是全局变量	查看全局变量文件，检查相关变量是否存在
52563	全局工程中不可创建/使用/查看工程变量	全局工程中无工程变量，故无法支持工程变量的相关操作。	-
52564	获取 IO 值失败	-	-
52565	保存的点位数组元素太多，可删除一些数组后尝试	系统对存储点位数组变量的文件长度有限制，超过限制时系统将报错。	根据现场情况适当优化点位数组元素
52566	点位数组个数已达最大值	系统中点位数组变量的总个数不能超过 32 个	根据现场情况适当优化点位数组变量的个数
52567	获取显示变量域失败	变量域显示标志未被成功从文件中读取到	联系厂家
52568	设置显示变量域失败	变量域显示标志未被成功写入到文件中	
52701	“停止”命令写入到解析器或者内核中失败	命令队列正在被操作，或者系统处于锁定状态	联系厂家
52702	启动解析器或者内核失败	-	-
52703	程序指针设置失败	设置指针，则暂停状态转到停止状态，失败即报此错误； 程序未加载，或行号不合理也报错	1. 检查是否在暂停状态 2. 联系厂家
52704	程序模式设置失败	模式分为单步、连续、倒序，若给定值不在此三种，则报错	联系厂家
52705	工具坐标系变量获取失败	-	-
52706	工具坐标系设置失败	当前状态不允许设置，或者队列被锁定，或者队列满	联系厂家
52707	用户坐标系获取失败	域给定错误，或者不存在此变量，或者变量格式错误	查看变量文件相关变量是否正常
52708	用户坐标系设置失败	-	联系厂家
52709	程序没有加载	程序没有加载	查看程序是否加载
52710	PLC 变量属性有变更，更改将在重启后生效	PLC 变量操作（新建、修改、删除）时，提示	
52711	当前程序仅为打开状态，不允许启动	加载或者调用（CALL）的程序，才允许启动	检查程序状态是否仅为打开
52712	程序不处于停止或暂停状态，启动程序失败	程序运行状态在停止或暂停时，才允许启动	检查程序状态是否为停止或暂停
52713	启动运动内核失败，无法启动程序	启动运动内核失败，无法启动程序	原因较复杂，需根据操作与日志分析
52714	停止指令下发失败	命令队列正在被操作，或者系统处于锁定状态	联系厂家
52715	当前无工程加载，或打开的程序与加载程序不匹配	-	-
52716	暂停程序失败	-	-
52717	恢复启动内核失败，无法启动程序	不在暂停状态，或者命令队列正在被操作，或者系统处于锁定状态	1. 检查是否在暂停状态 2. 联系厂家
52718	恢复指令解析失败，无法启动程序	不在暂停状态，或者没有加载或调用的程序	检查是否在暂停状态
52719	启动指令解析失败，无法启动程序	不在停止状态，或者没有加载或调用的程序	检查是否在停止状态
52720	“停止”命令写入解析器失败	-	-

报警代码	报警内容	报警原因	解决措施
52721	“暂停”命令写入解析器失败	-	-
52722	“暂停”命令写入内核失败	命令队列正在被操作，或者系统处于锁定状态，或者队列满	联系厂家
52723	激活区域命令写入解析器失败	-	-
52724	冻结区域命令写入内核失败	-	-
52725	此次工具参数标定错误，请重新标定	此次工具参数标定错误，请重新标定	重新标定
52726	此次用户坐标系标定错误，请重新标定	存在重合点	检查标定点是否存在相同点，重新标定
52727	获取暂停点与当前点是否一致的状态失败	写入命令队列失败，或者未得到命令回复，或者命令执行失败	联系厂家
52728	请切换到手动模式，再设置为倒序执行	-	-
52729	设置的变量类型不合法	变量类型：1、工具；2、坐标系；3、负载。其他值均不合法	查看类型是否合法
52730	一点到达只支持手动模式	一点到达只支持手动模式	切换到手动模式
52731	机器人未完全停止状态，不响应 GO 操作，请增大抬起与按下的时间间隔	机器人完全停止，需要一点时间，若操作频繁且速度快，则可能在没完全停止时，接到再次开启命令，以致报错	增大 GO 按钮释放与按下的时间间隔
52732	从文件中刷新全局变量失败	-	-
52733	因系统切到停止状态而导致本次中断被忽略	在中断程序执行 IEnter 指令的过程中，系统触发了暂停操作后本次中断无效，系统会退出中断执行状态。	待系统运行后，再次触发中断即可
52734	暂停复位命令写入内核失败	-	-
53001	modbus tcp 通讯中断	modbus 发送失败，或者接收失败，或者超时	联系厂家
53002	网络连接断开超过时限，系统加密	-	-
53003	公网下发命令，系统加密	-	-
53004	公网下发命令，功能使能	-	-
53005	公网下发命令，永久解锁	-	-
53006	已超过保养周期或日期，建议及时保养！	使用超出保养周期或日期	联系厂家
53007	当前机型与保养机型不匹配，请重置保养信息！	实际机型与记录保养机型不匹配	进入“保养提示”界面，点击“重置”按钮，联系厂家提供动态密码。
53008	示教器设置 Modbus 寄存器失败	-	-
53009	示教器读取 Modbus 寄存器失败	-	-
53010	氮气弹簧使用时间已超过半年，请及时维护检查！	氮气弹簧使用时长超出保养周期或日期	联系厂家
53502	加载 eni.xml 文件失败	eni.xml 文件不存在或者格式错误	检查 eni.xml 文件是否存在或检查 eni.xml 文件格式是否正确
53503	加载 slaveinfo.xml 文件失败	slaveinfo.xml 文件不存在或者格式错误	检查 slaveinfo.xml 文件是否存在或检查 slaveinfo.xml 文件格式是否正确
53504	保存文件失败	执行文件保存操作时，写入内容失败	检查当前文件是否被其他程序占用，导致无法写入
53505	备份文件失败	执行备份文件操作时失败	检查当前文件是否被其他程序占

报警代码	报警内容	报警原因	解决措施
			用，导致无法写入
53506	设备类型输入错误	暂不支持该设备类型的从站配置功能	-
53507	操作文件错误	执行文件修改操作时出现错误	联系厂家
53508	打开文件错误	执行打开文件时失败	检查文件是否存在或者文件路径是否正确
53509	该电柜不支持操作 IO	不支持该电柜类型添加 IO 设备	-
53510	添加伺服数量超限	伺服添加数量超出限制，不支持继续添加	-
53511	添加 IO 数量超限	IO 添加数量超出限制，不支持继续添加	-
53512	添加编码器数量超限	编码器添加数量超出限制，不支持继续添加	-
53513	删除伺服失败，拓扑最后一个伺服不可删除	当前拓扑中最后一个伺服从站类型不可以被删除	-
53514	删除 IO 失败，拓扑最后一个 IO 不可删除	当前拓扑中最后一个 IO 不可以被删除	-
53515	删除编码器失败，拓扑最后一个编码器不可删除或不存在	当前拓扑中不存在编码器或者最后一个编码器不可以被删除	-
54001	模型辨识模块，与内核通讯失败	辨识功能模块映射内核的通讯模块失败	联系厂家
54002	模型辨识模块，创建与子模块的通讯失败	辨识功能模块的子模块运行失败	
54003	模型辨识模块，与子模块的通讯失败	辨识功能模块与子模块通讯失败	
54004	模型辨识模块，校验与子模块的参数配置，当前不支持该机型类型	辨识功能不支持当前的机型	-
54005	模型辨识模块，校验与子模块的参数配置，本体轴数设置有误	机器人本体轴数设置有误	检查机型配置参数
54006	模型辨识模块，校验与子模块的参数配置，轴速比参数设置有误	机器人本体关节的减速比参数设置有误	检查机型配置参数
54099	模型辨识过程中，请等待数分钟...	模型辨识进行中	-
54101	模型辨识模块，通讯连接失败	辨识功能模块映射内核的通讯模块失败	联系厂家
54102	模型辨识模块，不支持当前机型	辨识功能不支持当前的机型	-
54103	模型辨识模块，关节参数校验有误	机器人本体的关节参数设置有误	检查机型配置参数
54104	模型辨识模块，腕部关节参数校验有误	机器人本体的腕部关节参数设置有误	检查机型配置参数
54105	模型辨识模块，校验初始化状态失败	机器人本体的参数设置有误	检查机型配置参数
54106	模型辨识模块，校验机型配置参数失败	机器人配置参数设置有误	检查机型配置参数
54107	模型辨识模块，校验机型对象失败	机型配置参数有误	联系厂家
54108	模型辨识模块，子模块通讯连接失败	辨识功能模块与子模块通讯失败	
54109	模型辨识模块，校验子模块初始化失败	辨识功能模块的子模块初始化失败	
54110	模型辨识模块，模型类型配置有误	模型辨识指令参数设置有误	检查指令参数设置
54111	模型辨识模块，负载/工具辨识时，缺少本体模型	缺少辨识依赖的模型文件	联系厂家

报警代码	报警内容	报警原因	解决措施
54112	模型辨识模块, 子模块通讯建立失败	辨识功能模块的子模块通讯建立失败	
54113	模型辨识模块, 子模块通讯数据为空	辨识功能模块的子模块通讯失败	
54114	模型辨识模块, 更新模型文件失败	辨识的模型更新文件时出错	
54115	模型辨识模块, 清理缓存失败	清理缓存失败	
54116	模型辨识模块, 清理临时文件	清理临时文件失败	
54117	模型辨识模块, 子模块建立失败	辨识功能模块的子模块建立失败	
54118	模型辨识模块异常,请重启后再使用该功能	模型辨识模块运行中出错	
54119	模型辨识模块, 负载/工具结果数据误差较大, 不建议采用	辨识的模型准确度太低	重新采集辨识数据, 以及重新执行模型辨识
54120	模型辨识模块, 未找到 Robot 辨识数据	未找到辨识数据	重新采集辨识数据
54121	模型辨识模块, Robot 辨识数据量太少	辨识数据量太少	修改辨识轨迹, 并重新采集辨识数据
54122	模型辨识模块, Robot 辨识数据量过多	辨识数据量太多	修改辨识轨迹, 并重新采集辨识数据
54123	模型辨识模块, 未找到 WithLoad 辨识数据	未找到辨识数据	重新采集辨识数据
54124	模型辨识模块, WithLoad 辨识数据量太少	辨识数据量太少	修改辨识轨迹, 并重新采集辨识数据
54125	模型辨识模块, WithLoad 辨识数据量过多	辨识数据量太多	修改辨识轨迹, 并重新采集辨识数据
54126	模型辨识模块, 未找到 ValidData 辨识数据	未找到辨识数据	重新采集辨识数据
54127	模型辨识模块, ValidData 辨识数据量太少	辨识数据量太少	修改辨识轨迹, 并重新采集辨识数据
54128	模型辨识模块, ValidData 辨识数据量过多	辨识数据量太多	修改辨识轨迹, 并重新采集辨识数据
54130	模型辨识模块, 当前剩余存储空间过小	当前系统剩余存储空间过小	清理系统空间
54170	模型辨识模块, 机型对象中参数信息有误	机器人机型及参数设置有误	检查机型及关节参数
54171	模型辨识模块, 杆件参数信息有误	机器人的杆件参数设置有误	检查机器人配置参数
54172	模型辨识模块, 轴电机额定转矩参数信息有误	机器人轴的参数设置有误	检查机器人配置参数
54173	模型辨识模块, 关节运动当量参数信息有误		
54174	模型辨识模块, 关节速比或耦合参数信息有误		
54175	模型辨识模块, 指令配置参数信息有误	辨识指令参数设置有误	检查辨识指令的配置参数
54176	模型辨识模块, 本体安装类型信息有误	机器人本体安装类型参数设置有误	检查机器人配置参数
54177	模型辨识模块, 附加机构配置信息有误	机器人附加机构参数设置有误	
54178	模型辨识模块, 附加信息有误	机器人附件参数设置有误	
54300	模型辨识模块, 配置参数异常导致模块运行中断	机器人配置参数设置有误	联系厂家
54301	模型辨识模块, 计算过程中发生暂停, 系统转入停止	模型辨识过程中出错	重新执行辨识指令



南京埃斯顿机器人工程有限公司

NANJING ESTUN ROBOT ENGINEERING CO., LTD.

南京市江宁经济开发区吉印大道1888号

025-52785866

www.estun.com

025-52785576

info@estun.com