

**For a Better Life**  
让生活更美好!



# iER系列 操作手册

iEPM01V30401-01



自动化核心部件  
运动控制系统



工业机器人及  
智能系统工程



数字化产品  
数字化服务



## 申明

### 版权与知识产权

本手册及其包含的所有文本、图像、设计、软件及相关信息（统称“文档材料”）均为南京埃斯顿自动化股份有限公司（简称“埃斯顿公司”）的独家财产，受中华人民共和国及相关国际版权法和条约保护。除非获得埃斯顿公司的明确书面授权，否则严禁对本文档材料进行任何形式的复制、分发、修改、公开展示或反向工程。

### 免责声明

1. 埃斯顿公司致力于确保手册信息的准确性，但不对其完整性、准确性或无误性作任何明示或默示的担保。埃斯顿公司将定期校对本手册的内容，并在之后的版本中作必要的更改，同时保留在不影响功能的情况下进行技术更改的权利，恕不另行通知。
2. 安全操作是用户的首要责任。用户有责任确保其人员已接受充分培训，并严格按照本手册、所有相关安全规程以及国家、地区和行业标准来安装、操作和维护本产品。任何因未遵守操作规程、擅自改装或使用非原装备件所导致的人身伤害、设备损坏或生产损失，埃斯顿公司概不负责。
3. 在任何情况下，埃斯顿公司均不对因使用或无法使用本产品及手册而引发的任何间接损失、附带损失、利润损失、数据丢失或生产中断等后果性损害承担责任。

### 文档申明

1. 本手册为“Original Instruction”，经制造商埃斯顿公司核实确认。
2. 如需本手册的其他语言翻译版本，可向制造商或其授权代表申请获取。

## 安全使用须知

本章说明为安全使用机器人而需要遵守的内容。在使用机器人之前，务必熟读并理解本章中所述内容。

使用埃斯顿机器人的公司、个人应该熟读所在地区、国家的标准和法律，并且安装适当的安全设施保护机器人的使用人员。使用前（安装、运转、保养、检修），请务必熟读并全部掌握本说明书和其他附属资料，在熟知全部设备知识、安全知识及注意事项后再开始使用。但是使用人员即使完全按照手册中给出的所有安全信息进行，埃斯顿公司也无法保证使用人员不会受到任何伤害。

### 使用人员的定义

#### 1. 操作人员：

- 执行机器人电源的开启（ON）与关闭（OFF）操作；
- 通过示教器和控制柜操作面板启动、暂停或停止已预设的机器人运行程序；
- 监控机器人日常运行状态，及时反馈异常情况，不涉及程序修改、示教或维修操作；
- 操作人员不得在安全围栏内进行作业。

#### 2. 编程人员：

- 进行机器人程序的编写、修改、优化及存储管理；
- 在指定安全区域内，通过示教器完成机器人的示教编程、轨迹调试等作业；
- 验证程序运行效果，确保程序符合作业需求，不涉及机器人核心部件维修或结构调整；
- 编程人员必须接受针对机器人的专业培训。

#### 3. 维修人员：

- 执行机器人的定期维护、故障排查、部件修理、参数调整及零件更换作业；
- 为维修作业需要，可在安全区域内进行必要的机器人辅助示教或操作；
- 记录维护维修数据，反馈设备运行损耗情况，确保机器人性能稳定；
- 维修人员必须接受针对机器人的专业培训。

使用人员权限划分：本表格界定安全围栏外作业人员的操作权限范围；符号“0”代表该人员具备对应操作权限，“-”代表无操作权限。

表 0-1 安全围栏外作业权限一览表

作业内容	操作人员	编程人员	维修人员
机器人控制柜电源通断	0	0	0

表 0-1 安全围栏外作业权限一览表 (续)

作业内容	操作人员	编程人员	维修人员
选择自动/远程 (A)、手动 (T1)、手动 100% (T2) 操作模式	-	0	0
示教器选择机器人程序	-	0	0
外部设备选择机器人程序	-	0	0
示教器启动机器人程序	0	0	0
示教器复位报警	-	0	0
示教器参数设置	-	0	-
示教器示教操作	-	0	0
控制柜急停按钮操作	0	0	0
示教器急停按钮操作	0	0	0
控制面板维护	-	0	-
维护期间示教操作	-	-	0

所有涉及机器人操作、编程调试及维护维修的作业场景，操作人员、编程人员、维修人员必须严格遵守安全防护要求，全程规范穿戴符合安全标准的防护装备：



- 需选用与作业内容适配的作业服，避免衣物卷入设备或勾挂部件；
- 必须穿戴具备防砸、防滑功能的安全鞋；
- 作业期间需正确佩戴安全帽；
- 需定期检查防护装备完好性，破损、老化的装备需立即更换，若涉及电气维修、高温作业等特殊场景，还需按对应作业规范额外配备绝缘手套、耐热手套等专项防护装备。



## 📖 说明

关于培训的更多信息，请咨询埃斯顿公司工作人员。

## 安全标示

本手册中出现的以下安全标示，均为核心操作与安全指引依据，用户在阅读手册内容时，必须仔细研读标示对应的说明，严格遵守相关要求，规避安全风险。

标示符号	定义说明
 危险	<ul style="list-style-type: none"> <li>• 本标志对应的文本表示存在高度潜在危险。若无法避免相关风险，将直接导致人员死亡或严重伤害，必须严格遵守相关安全要求。</li> </ul>
 警告	<ul style="list-style-type: none"> <li>• 本标志对应的文本表示存在中度或低度潜在危险。若忽视相关提示，可能导致人员轻微伤害或中等程度伤害，需规范操作规避风险。</li> </ul>

标示符号	定义说明
 注意	<ul style="list-style-type: none"> <li>本标志对应的文本表示存在非人身伤害类潜在风险。若忽视相关说明，可能造成设备损坏、数据丢失、设备性能下降或其他不可预知的不良后果。</li> </ul>
 说明	<ul style="list-style-type: none"> <li>本标志对应的文本为正文的附加补充信息，用于强调核心内容、补充操作细节或解释关键概念，帮助用户更准确理解手册内容并高效完成操作。</li> </ul>

## 使用人员的安全事项

- 搬运与安装机器人时，必须严格遵循埃斯顿公司规定的标准流程操作。采用非标准方法可能导致机器人倾倒、部件损坏，进而引发安全事故。
- 机器人安装前必须划定安全防护区域，需在其工作区域周边设置防护栅栏及安全警示牌。
- 机器人上方严禁放置悬挂物，防止物体掉落砸坏机器人及周边设备。
- 严禁倚靠控制柜或随意触碰按钮，避免机器人产生意外动作，造成人身伤害或设备损坏
- 拆分机器人过程中，需警惕设备上松动或脱落的零件，防止砸伤作业人员。
- 对外部设备进行单独调试时，必须先断开机器人电源，再开展操作。
- 所有外围设备必须连接符合规范的接地线，确保用电安全。
- 首次操作机器人时，必须先以低速运行，再逐步提升速度，全程观察并确认设备无异常。
- 使用示教器时，戴手套可能引发操作失误，必须摘下手套再进行操作。
- 程序、系统变量等关键信息可存储至存储卡等介质，为防止意外导致数据丢失，建议用户定期备份数据。
- 严禁擅自搬动机器人各轴，否则可能引发人身伤害或设备损坏。
- 进行控制柜与机器人、外围设备间的配线及配管时，必须采取防护措施（如将管线、线缆从预埋坑穿过或加装保护盖遮挡），避免被人员踩踏或叉车碾压损坏。
- 机器人工作期间均可能产生意外动作，对工作范围内人员造成严重伤害或损坏设备。机器人启动前，必须测试安全门、抱闸、安全指示灯等所有安全措施的可操作性，且必须确保工作范围内无其他人员。
- 通过软件设定的动作范围及负载条件，严禁设定超出产品规格表中的规定值；设置不当可能引发人员伤害或设备损坏。
- 若必须在机器人工作范围内作业，需遵循以下安全规则：
  - 需将机器人切换至手动模式后再通过示教器上使能，并断开上位机控制等所有自动控制方式；
  - 留意机器人转动关节，防止头发、衣物卷入；同时需警惕机器人及其他附属设备运动可能引发的其他危险（如碰撞、挤压等）；
  - 作业前需测试电机抱闸功能是否正常，防止因机器人异常运动导致人身伤害；


- 需预先规划机器人突然向自身方位运动时的应急避让方案；
- 需预设安全躲避区域，以备紧急情况使用。

### ▲ 危险

在任何工况下，操作人员、编程人员及维修人员均严禁在机器人本体机械臂下方站立、停留或进行作业。该区域存在极高安全风险：机器人可能因系统故障发生异常运动，或被他人误触使能开关启动，导致人员被碰撞、挤压，造成重伤甚至死亡。所有相关人员必须严格遵守此禁令，确保作业安全。

#### 操作人员的安全注意事项：

1. 操作机器人前，必须先按下控制柜前门及示教器右上方的急停按钮，检查“伺服准备”指示灯是否熄灭，确认机器人电源已完全关闭。
2. 操作期间，严禁非工作人员触碰或操作控制柜。
3. 向机器人安装工具时，必须先关闭（OFF）控制柜及待装工具的电源，并悬挂警示牌。安装期间若擅自接通电源，可能造成触电，或导致机器人意外运动引发伤害。
4. 急停功能：
  - 急停功能独立于机器人所有电气控制，可强制停止机器人全部运动。
  - 触发急停后，驱动机器人上伺服电机运行的动力电源将会断开，伺服电机抱闸，机器人停止运动；  
需先释放急停按钮，再重新启动机器人，设备方可恢复运行。

急停按钮示意图	特别说明
	所有急停按钮均需安装在易于触及的位置，确保突发意外时能快速触发，及时停止机器人运行。

### ▲ 危险

操作人员需要注意伺服电机的动力线、连接夹具和其他装置的动力线的高电压危险。

### ▲ 注意

急停功能仅用于紧急场景（如人身安全受威胁、设备突发失控等）下强制停止机器人，严禁用于日常程序暂停、正常关机等非紧急操作。

#### 编程人员的安全注意事项：

机器人示教作业期间，编程人员若需进入机器人动作范围，必须重点关注作业安全。

---

**⚠ 注意**

自动模式下，使能的接通和断开通过操作示教器上的Mot按钮实现切换；对于正在运行的机器人，需要先按下Stop按键停止运行，之后再按下Mot按钮实现使能断开。

---

安全使用示教器，必须遵守以下规则：

- 确保紧急按钮始终处于有效状态，避免应急时无法正常操作。
- 示教者进入机器人工作区域时，必须随身携带示教器，避免他人在不知情时操作机器人。
- 禁止将示教器放置于机器人工作范围内，防止机器人运动时碰撞示教器，引发异常动作。

**维修人员的安全注意事项：**

1. 机器人发热部件安全警示：机器人正常运行时，部分关键部件会发热，其中伺服电机、减速机的发热尤为明显，靠近或直接触碰易导致烫伤。若需在部件发热状态下触碰，必须佩戴耐热手套等防护用具，避免直接接触高温部件。
- 

**⚠ 注意**

触摸这些高温部件前，需先用手靠近部件表面初步感受温度，判断是否安全，防止直接触摸导致烫伤。

机器人停机后，需等待高温部件冷却至安全温度，再开展维修工作，避免高温引发人员灼伤。

---

2. 拆卸部件的安全注意事项：拆卸前必须先确认齿轮、轴承等内部零件完全停止旋转与运动，再打开部件盖子或保护装置；严禁在上述部件旋转时开启保护装置，防止卷入风险。若内部存在活动零件（非固定状态），需使用辅助装置将其固定在原始位置，避免拆卸过程中零件移位、掉落引发安全隐患。

维修、安装或保养后的首次启动测试，必须严格遵循以下步骤：

- a. 彻底清理机器人本体及其工作区域内的所有工具、材料及临时装置，确保无遗留物；
- b. 完整恢复并确认所有安全防护装置已正确安装且功能正常；
- c. 确认所有人员已撤离至机器人安全工作范围之外，禁止任何人员在测试期间进入作业区域；
- d. 启动测试时，应重点监控本次维修或更换部件的运行状态，实时观察是否存在异响、卡顿等异常，及时停机排查。

---

**⚠ 注意**

在维修过程中，禁止站立、攀爬或倚靠机器人任何部位。必须使用符合安全标准的登高设备，以防跌落造成严重人身伤害。

---

3. 气动系统安全注意事项：关闭气源泵后，系统内仍会残留带压气体或液体，此类残留物具有潜在能量。需通过专项措施释放残留能量，避免其对人员或设备造成冲击、喷射等伤害；维修气动元件前，必须确保系统内残留能量已完全释放。
- 

**⚠ 注意**

为防止气动系统能量意外释放引发安全事故，必须安装安全阀，作为系统压力过载时的强制保护装置。

---

4. 故障诊断阶段需开启机器人电源，但维修机器人时必须完全关闭电源，并切断所有相关电源连接，避免带电作业引发安全风险。
5. 抱闸检测：机器人正常运行时，抱闸易因磨损导致锁止性能下降，需定期开展功能检测以保障关节安全锁止。具体检测步骤如下：
- a. 将机器人各关节调整至承受最大负载的姿态，模拟极端受力场景。
  - b. 关闭机器人主电源，使抱闸自动生效。
  - c. 在各关节的固定部件与活动部件间做对位标记，确保标记清晰可辨，便于后续观察位移。
  - d. 静置规定时长（通常为 10-30 分钟）后，观察各关节的对位标记是否偏移，以此判断抱闸是否存在滑闸、锁止失效情况。
6. 向减速机加注润滑油时，存在高温油液烫伤、压力溅射、油液污染等风险，可能伤及人员或损坏设备。开展加油作业前，必须严格遵守以下安全规范：
- 加注或排放润滑油时，需佩戴防化手套、护目镜等防护装备。
  - 打开油腔盖前需确认油腔温度降至安全范围（建议 $\leq 40^{\circ}\text{C}$ ），操作时保持安全距离（建议 $\geq 50\text{cm}$ ）；油腔内可能残留压力，需缓慢开启盖子，避免油液溅射引发伤害。
  - 按油量表刻度加注润滑油，严禁超过最大刻度线；加注完成后，需检查油液指示口，确认油位处于标准范围内。

- 严禁将不同型号、不同牌号的润滑油混入同一减速机；更换润滑油型号前，需用专用清洁剂彻底冲洗油腔，确保残余旧油清理干净。
- 排放旧油时，需将油腔内旧油彻底排空（可借助负压设备辅助）；加油后需再次检查油液指示口，双重确认油位符合要求。

---

### 注意

放空减速机旧油前，可先短时间低速运行机器人（建议 5-10 分钟，使油液温度升至 40-60℃），通过加热降低油液黏度，让放油更顺畅，同时减少油腔内壁的旧油残留。需注意控制运行时长，避免油液温度过高（不超过 60℃），防止后续操作时因高温油液引发烫伤。

---

## 刀具及外围设备安全注意事项

机器人关闭后，其连接的刀具、外围设备可能仍独立运行（未随机器人断电）。若此类设备的电源线、动力线缆出现破损，或人员误触带电部件，易引发触电等人身伤害。因此，操作前必须单独切断刀具及外围设备的电源，并检查线缆完整性，再开展后续作业。

## 机器人手臂紧急救援安全规范

当发生机器人手臂夹伤操作人员的紧急情况时，需立即对夹持手臂进行紧急移除，具体操作及安全要求如下：

### 1. 紧急移除操作：

- 小型机器人手臂：可采用手动移除方式，需 2 人及以上协作（1 人固定手臂、1 人协助受困者脱困），严禁单人操作以防手臂晃动加重伤害。
- 大型机器人手臂：必须使用吊车等专业承重设备，严禁使用小型推车、撬棍等非承重工具，避免设备倾覆或手臂坠落引发二次事故。

### 2. 释放关节抱闸前，必须先通过专用支架、承重绳索等工具稳固固定机械臂，确保机械臂完全脱离“悬空承重”状态，防止释放抱闸后手臂因重力下坠，对受困者造成挤压、砸伤等二次伤害。

---

### 警告

移除过程中若遇机械锁死、部件卡顿等问题，需立即停止操作并联系埃斯顿公司技术人员获取专业指导，严禁擅自拆解关节、抱闸等关键部件，避免破坏设备结构或引发新风险。

---

## 前言

### 修订记录

修订日期	发布版本	变更内容
2026.3	01	新建手册。

### 更多资料

文档编号	文档名称
-	各机器人对应的使用手册。
-	各控制柜对应的安全手册。

### 资料获取

本文档不随产品发货，如需获取电子版PDF文件，可以通过以下方式获取：登录埃斯顿公司官网网站（[www.estun.com](http://www.estun.com)），“**服务与支持 -> 资料下载**”，搜索关键字并下载。

## 目录

申明.....	2
安全使用须知.....	3
前言.....	10
图目录.....	24
表目录.....	25
1 产品介绍.....	26
1.1 产品组成.....	26
1.1.1 装箱内容确认.....	26
1.1.2 订货号确认.....	26
1.2 控制柜.....	26
1.3 示教器.....	27
1.3.1 外观结构.....	27
1.3.2 触摸笔.....	28
1.3.3 USB插槽.....	28
1.3.4 按键.....	28
1.3.5 显示界面.....	30
1.3.6 模式开关说明.....	33
1.3.7 急停按钮.....	34
1.3.8 伺服使能开关.....	35
1.3.9 示教器内部的接线端子信号定义.....	35
1.3.10 示教器线缆.....	36
2 产品使用.....	38
2.1 连接方法.....	38
2.2 操作说明.....	38
2.2.1 机器人坐标系.....	38
2.3 示教操作.....	41
2.3.1 示教前准备.....	41
2.3.2 示教基本步骤.....	42
2.3.2.1 新建示教工程.....	42
2.3.2.2 编制示教程序.....	43
2.3.2.3 修改程序.....	45
2.4 程序再现.....	45
2.4.1 再现前准备.....	45
2.4.2 再现执行步骤.....	46
2.4.3 停止与再启动规则.....	46

2.4.4 再现速度修改方法.....	48
2.5 启停时序.....	48
3 系统功能介绍.....	49
3.1 系统设置.....	49
3.1.1 用户管理.....	49
3.1.2 机器人设置.....	51
3.1.2.1 机器人设置.....	51
3.1.2.2 本体参数.....	52
3.1.2.3 运动学参数.....	53
3.1.2.4 动力学参数.....	53
3.1.2.5 高级运动参数.....	54
3.1.2.6 高级功能参数.....	55
3.1.2.7 安全参数.....	55
3.1.2.8 厂家参数.....	56
3.2 系统维护.....	56
3.2.1 系统管理.....	56
3.2.2 示教器升级.....	57
3.2.3 控制器程序的更新.....	59
3.2.3.1 控制器升级.....	60
3.2.3.2 控制器补丁升级.....	61
3.2.4 备份恢复.....	63
3.2.4.1 手动备份.....	63
3.2.4.2 自动备份.....	66
3.2.4.3 数据恢复.....	68
3.2.5 版本信息.....	69
3.3 应用包安装及升级.....	70
3.3.1 应用软件管理.....	71
3.3.2 应用软件安装.....	71
3.3.3 应用软件卸载.....	72
3.4 日志管理.....	73
3.4.1 报警.....	73
3.4.2 查看日志.....	75
3.4.3 查看帮助文档.....	75
3.5 保养提示.....	76
4 工程管理.....	78
4.1 工程管理.....	78
4.1.1 新建工程/程序.....	79
4.1.2 删除工程/模块/函数.....	80

4.1.3 恢复工程/模块/函数.....	81
4.1.4 重命名工程/模块/函数.....	81
4.1.5 复制粘贴工程/程序.....	82
4.1.6 自启动工程.....	82
4.1.7 查找工程/程序.....	83
4.1.8 工程导入导出.....	84
4.1.9 管理程序信息.....	84
4.2 程序编辑.....	85
4.2.1 新增指令.....	86
4.2.2 撤销操作.....	89
4.2.3 快捷定位程序行位置.....	89
4.2.4 反向粘贴.....	90
4.3 程序数据.....	92
4.3.1 新增/删除变量.....	92
4.3.2 重命名变量.....	95
4.3.3 查看变量别名及注释.....	96
4.3.4 变量监控.....	97
4.4 多任务设置.....	100
4.4.1 新建任务.....	100
4.4.2 编辑任务.....	102
4.4.3 切换任务.....	103
4.5 平移功能.....	104
5 变量介绍.....	110
5.1 基本数据类型.....	110
5.1.1 BOOL.....	110
5.1.2 INT.....	110
5.1.3 FLOAT.....	110
5.1.4 STRING.....	111
5.1.5 BoolOneArray.....	111
5.1.6 IntOneArray.....	111
5.1.7 FloatOneArray.....	111
5.1.8 BoolTwoArray.....	112
5.1.9 IntTwoArray.....	112
5.1.10 FloatTwoArray.....	112
5.1.11 BoolThreeArray.....	112
5.1.12 IntThreeArray.....	113
5.1.13 FloatThreeArray.....	113
5.2 位置数据类型.....	113

5.2.1 APOS-关节位置类型.....	113
5.2.2 POSCFG.....	114
5.2.3 CPOS-坐标位置类型.....	118
5.2.4 DAPOS-关节相对位置类型.....	119
5.2.5 DCPOS 坐标系相对位置类型.....	119
5.3 系统数据类型.....	120
5.3.1 LsScale.....	120
5.3.2 LsThresh.....	120
5.3.3 Zone.....	121
5.3.4 Speed.....	121
5.3.5 ColliParam.....	122
5.3.6 VibrationParam.....	122
5.4 系统预留类型.....	122
5.4.1 AREA.....	122
5.4.2 LoadDyn.....	124
5.4.3 CenterPos.....	124
5.4.4 FRAME.....	126
5.4.5 TOOL.....	126
5.4.6 PAYLOAD.....	127
5.4.7 InertiaTensor.....	127
5.4.8 \$\$SYS_DATA.....	128
5.5 摆动数据类型.....	128
5.5.1 WEAVE.....	128
5.6 时钟数据类型.....	129
5.6.1 CLOCK.....	130
5.7 码垛数据类型.....	130
5.7.1 PALLET.....	130
5.7.2 PLACEOPTIONS.....	131
5.8 SOCKET数据类型.....	131
5.8.1 TcpServer.....	131
5.8.2 TcpSocket.....	131
5.8.3 UdpSocket.....	132
5.9 掉电保持数据类型.....	132
5.9.1 BoolRetain.....	132
5.9.2 IntRetain.....	132
5.9.3 FloatRetain.....	132
5.9.4 StringRetain.....	133
5.10 异常数据类型.....	133

5.10.1 ExceptionInfo.....	133
5.11 返回数据类型.....	133
5.11.1 ReturnValue.....	133
5.12 视觉纠偏类型.....	134
5.12.1 VISIONRET.....	134
6 指令介绍.....	135
6.1 运动指令.....	135
6.1.1 MovJ - 点到点的运动指令.....	135
6.1.2 MovL - 直线运动指令.....	137
6.1.3 MovC - 弧形运动指令.....	139
6.1.4 MovCircle - 圆形运动指令.....	141
6.1.5 MovJSearch - 寻位指令 (MovJ) .....	143
6.1.6 MovLSearch - 寻位指令 (MovL) .....	145
6.1.7 MovJRel - 相对偏移指令.....	148
6.1.8 MovLRel-插补相对偏移指令.....	149
6.1.9 MovJOffset - 关节空间偏移运动指令.....	151
6.1.10 MovJOffsetC - 关节空间偏移运动指令.....	152
6.1.11 MovLOffset - 直线偏移运动指令.....	153
6.1.12 MovLOffsetC - 直线偏移运动指令.....	154
6.1.13 MovLW - 摆动指令.....	155
6.1.14 MovCW - 圆弧摆动指令.....	158
6.1.15 MovCircleW- 整圆摆动指令.....	160
6.1.16 MovJArch - 拱形运动指令.....	162
6.1.17 MovLArch - 直线拱形运动指令.....	164
6.1.18 MovH - 螺旋运动指令.....	166
6.1.19 OnDistance - 距离触发指令.....	168
6.1.20 OnParameter - 距离百分比触发指令.....	170
6.1.21 MovJSyncQuit - 同步跟随退出指令.....	171
6.1.22 MovLSync - 启动传送带同步的直线运动指令.....	172
6.1.23 MovLSyncQuit - 同步跟随退出指令.....	174
6.1.24 MovToIPos - 返回中断点指令.....	175
6.2 控制指令.....	176
6.2.1 LABEL - 定义跳转目标.....	177
6.2.2 GOTO - 跳转指令.....	178
6.2.3 IF-条件判断指令.....	179
6.2.4 ELSIF.....	180
6.2.5 ELSE.....	181
6.2.6 WHILE - 只要满足条件就循环.....	181

6.2.7 FOR - for 循环指令.....	182
6.2.8 RETURN - 返回指令.....	183
6.2.9 SWITCH.....	184
6.2.10 CASE.....	185
6.2.11 BREAK.....	186
6.2.12 DEFAULT.....	186
6.2.13 CONTINUE.....	186
6.2.14 ...=... - 赋值指令.....	186
6.2.15 Try——catch指令.....	189
6.2.16 FINALLY.....	189
6.2.17 #=...=#- 注释指令 .....	189
6.3 等待指令.....	190
6.3.1 Wait - 延时等待指令.....	190
6.3.2 WaitFinish - 等待运动段执行百分比指令.....	191
6.3.3 WaitCondition - 等待指定判断条件指令.....	192
6.4 IO指令.....	193
6.4.1 PulseOut - 设置脉冲输出指令.....	193
6.4.2 WaitIO - 等待IO信号输入指令.....	194
6.4.3 WaitIO8421 - 等待一组IO信号输入指令.....	196
6.5 设置指令.....	197
6.5.1 SetPayload - 设置负载重量指令 .....	197
6.5.2 SetJointDyn - 设置关节运动参数指令.....	198
6.5.3 SetCartDyn - 设置直线或圆弧运动动态参数指令.....	199
6.5.4 SetBlendParam - 设置过渡参数.....	200
6.5.5 SetColliEnable - 设置碰撞检测使能指令.....	201
6.5.6 SetAxisColliParam - 设置碰撞检测灵敏度指令.....	201
6.5.7 GetCurOverride - 获取当前全局速度指令.....	202
6.5.8 CalcTool - 计算工具参数.....	203
6.5.9 SetRtToErr - 设置报警指令.....	204
6.5.10 SetRtInfo - 设置提示信息.....	205
6.5.11 SetRtWarning - 设置报警信息.....	206
6.5.12 Stop - 停止运行指令.....	207
6.5.13 SetMotionMode - 设置运行模式.....	207
6.5.14 AutoTune - 提速自调节指令.....	208
6.5.15 AutoGainEnable - 设置伺服参数增益指令.....	209
6.5.16 AutoGainDisable - 关闭伺服参数增益指令.....	210
6.5.17 SetAxisVibraBLevel - 设置振动抑制等级指令.....	210
6.5.18 SetRestorePC - 设置恢复运行时的 PC 指令.....	211

6.5.19 SetCircMode - 设置圆弧模式指令 .....	212
6.5.20 SetOriMode - 设置姿态模式指令 .....	213
6.5.21 GetRobIsMoving - 获取当前机器人的运动状态指令.....	213
6.5.22 SetVibrationParam - 设置振动抑制自学习参数.....	214
6.5.23 SetWristMoveEnable.....	215
6.5.24 SetWristMoveDisable.....	217
6.6 位置运算指令.....	217
6.6.1 GetCurAPos - 获取当前关节位置指令.....	218
6.6.2 GetCurCPos - 获取当前笛卡尔坐标系位置指令.....	218
6.6.3 APosToCPos - 机器人位置转换指令 (Apos 转换成 Cpos) .....	219
6.6.4 CPosToAPos - 机器人位置转换指令 (Cpos 转换成 Apos) .....	220
6.6.5 CPosToCPos - 机器人位置转换指令 (Cpos 转换成 Cpos) .....	221
6.6.6 CalPosOffset - 计算位置点偏移结果指令.....	222
6.6.7 FrameOffset - 坐标系偏移指令.....	223
6.6.8 ToolOffset-工具坐标系偏移指令.....	224
6.6.9 PosOffset - 轴坐标偏移指令.....	225
6.7 传送带跟随指令.....	226
6.7.1 WaitWObj - 等待视觉信息输入指令.....	226
6.7.2 MovLSync - 启动传送带同步的直线运动指令.....	228
6.7.3 MovJSyncQuit - 同步跟随退出指令.....	229
6.7.4 MovLSyncQuit - 同步跟随退出指令.....	230
6.7.5 SetTargetPos - 设置工件位置信息指令.....	232
6.7.6 SynCToUserC - 示教参考坐标系转换指令.....	233
6.7.7 WaitConvDis - 等待固定增量指令.....	233
6.7.8 SimConveyorOn - 虚拟传送带开启指令.....	234
6.7.9 SimConveyorOff-虚拟传送带关闭指令.....	235
6.7.10 ReceiveWObj-工件数据接收状态指令.....	235
6.7.11 ResetWObjBuf - 复位已工件数据队列指令.....	236
6.8 时钟指令.....	237
6.8.1 CLKStart-启动时钟指令.....	238
6.8.2 CLKStop - 停止时钟指令.....	238
6.8.3 CLKRead - 读取时钟指令.....	238
6.8.4 CLKReset - 复位时钟指令.....	238
6.9 区域指令.....	239
6.9.1 AreaActivate - 激活区域指令.....	239
6.9.2 AreaDeactivate - 冻结区域指令.....	240
6.9.3 AreaConnectIO - 绑定区域 IO 指令.....	240
6.9.4 AreaDisConnectIO - 解除绑定区域 IO 指令.....	241

6.10 码垛指令.....	242
6.10.1 PalletReset - 码垛重置指令.....	243
6.10.2 PalletToPut - 放置工件指令.....	243
6.10.3 PalletFromPut - 离开放置位置指令.....	246
6.10.4 PalletToGet - 抓取工件指令.....	248
6.10.5 PalletFromGet - 离开抓取位置指令.....	249
6.11 Socket指令.....	250
6.11.1 Socket指令样例说明.....	251
6.11.2 SocketCreateServer - 创建 socket服务器指令.....	252
6.11.3 SocketAccept - 监听并接收TCP Socket客户端连接指令.....	253
6.11.4 SocketCreateClient - 创建socket客户端指令.....	254
6.11.5 SocketReceive - 接收Socket数据指令.....	254
6.11.6 SocketSend - 通过 Socket 发送指令.....	255
6.11.7 SocketClose - 关闭 socket指令.....	256
6.11.8 UDPCreate - 创建UDP Socket指令.....	257
6.11.9 UDPBind - 绑定UDP指令.....	258
6.11.10 UDPReceive - 接收UDP指令.....	259
6.11.11 UDPSend - 发送UDP指令.....	260
6.12 ModbusTCP 指令.....	262
6.12.1 GetModConState - 获取与 ModbusTCP 通讯连接连接状态.....	263
6.12.2 ReadModbusReg - 读取指定 Modbus 寄存器值.....	264
6.12.3 WriteModbusReg - 设置指定 Modbus 寄存器值.....	264
6.13 字符串指令.....	265
6.13.1 byte.....	266
6.13.2 char.....	266
6.13.3 find2.....	266
6.13.4 fineEnd.....	266
6.13.5 format.....	266
6.13.6 getAt.....	267
6.13.7 gsub.....	267
6.13.8 len.....	268
6.13.9 left.....	268
6.13.10 lower.....	268
6.13.11 right.....	268
6.13.12 strcmp.....	268
6.13.13 reverse.....	269
6.13.14 trimLeft.....	269
6.13.15 trimRight.....	269

6.13.16 upper.....	269
6.13.17 IToStr.....	269
6.13.18 RToStr.....	270
6.13.19 StrTol.....	270
6.13.20 StrToR.....	270
6.13.21 StrToIntArray.....	270
6.13.22 StrToFloatArray.....	270
6.13.23 IntArrayToStr.....	271
6.13.24 FloatArrayToStr.....	271
6.13.25 APosToStr.....	272
6.13.26 CPosToStr.....	272
6.13.27 TranStrToInt.....	272
6.13.28 TranStrToReal.....	273
6.13.29 TranStrToAPos .....	273
6.13.30 TranStrToCPos.....	273
6.14 视觉指令.....	274
6.14.1 TrigCam - 触发拍照指令.....	274
6.14.2 WaitFinishCam - 等待拍照完成指令.....	275
6.14.3 GetCamPos - 获取相机纠偏数据指令.....	276
6.14.4 SendMessage - 发送消息字符串指令.....	277
6.15 数学运算函数.....	278
6.15.1 sin.....	278
6.15.2 cos.....	279
6.15.3 tan.....	279
6.15.4 asin.....	279
6.15.5 acos.....	279
6.15.6 atan.....	279
6.15.7 atan2.....	280
6.15.8 sinh.....	280
6.15.9 cosh.....	280
6.15.10 tanh.....	280
6.15.11 log.....	280
6.15.12 log10.....	281
6.15.13 sqrt.....	281
6.15.14 exp.....	281
6.15.15 pow.....	281
6.15.16 deg.....	281
6.15.17 rad.....	282

6.15.18 fmod.....	282
6.15.19 floor.....	282
6.15.20 radom.....	282
6.16 阵列指令.....	282
6.16.1 SetMatrix - 设置矩阵组.....	283
6.16.2 GetMatrix - 获取矩阵组.....	284
6.17 中断指令.....	284
6.17.1 IConnect - 中断连接指令.....	285
6.17.2 IDelete - 中断解除指令.....	286
6.17.3 ITimer - 定时器中断指令.....	286
6.17.4 ISignalDI - 输入口中断指令.....	287
6.17.5 ISignalSimDI - 虚拟输入口中断指令.....	288
6.17.6 ExitCycle - 退出循环指令.....	289
6.17.7 MovToIPos - 返回中断点指令.....	290
6.17.8 ISleep - 中断休眠指令.....	291
6.17.9 IWatch - 中断激活指令.....	291
6.18 交互式指令.....	293
6.18.1 DialogPanelInt - int型交互式指令.....	293
6.18.2 DialogPanelFloat - float型交互式指令.....	294
6.18.3 DialogPanelString - string型交互式指令.....	294
6.18.4 DialogPanelList - list型交互式指令.....	295
6.18.5 DialogPanelMessage - message型交互式指令.....	296
6.19 样条曲线指令.....	297
6.19.1 GetPosFromFile.....	298
6.19.2 OnSplineIndex.....	299
6.19.3 SetSplineParam.....	300
6.19.4 MovS.....	301
7 高级功能.....	304
7.1 运动.....	304
7.1.1 碰撞检测.....	304
7.1.1.1 碰撞检测.....	304
7.1.1.2 碰撞回退.....	305
7.1.2 振动抑制.....	306
7.1.2.1 自学习振动抑制.....	307
7.1.3 场景适配调试.....	308
7.1.4 软浮动功能.....	312
7.2 区域监控.....	316
7.2.1 多类型区域监控.....	316

7.3 多工位预约.....	322
7.4 附加轴配置.....	326
7.5 伺服管理.....	331
7.5.1 伺服参数设置.....	331
7.5.2 伺服参数开机自检.....	334
7.5.3 力矩监控功能.....	336
7.6 传送带跟随.....	337
7.6.1 传送带参数配置.....	338
7.6.2 编码器参数配置.....	339
7.6.2.1 直线传送带脉冲当量标定.....	340
7.6.2.2 圆盘传送带脉冲当量标定.....	342
7.6.2.3 脉冲当量验证.....	342
7.6.3 边界参数配置.....	342
7.6.4 跟随参数配置.....	343
7.6.5 触发参数配置.....	344
7.6.6 多机传送带标定.....	346
7.6.7 视觉参数配置.....	347
7.7 负载辨识.....	349
7.8 多机协同功能.....	352
7.9 制动测试功能.....	357
7.10 双机防碰撞功能.....	362
7.11 臂上负载.....	363
7.12 标定.....	364
7.12.1 工件坐标系.....	365
7.12.2 工具坐标系标定.....	369
7.12.3 用户坐标系标定 .....	371
7.12.4 变位机标定.....	373
7.12.5 一点回原位功能.....	376
7.12.6 工具对正.....	378
7.12.7 零点标定.....	379
7.13 I/O.....	383
7.13.1 IO界面.....	384
7.13.2 系统事件与IO信号关联.....	386
7.13.3 Cross-Connect.....	389
7.13.4 外部IO控制.....	392
7.14 通讯.....	394
7.14.1 九点标定.....	395
7.14.2 总线配置与监控.....	396

7.14.2.1 EIP授权激活.....	397
7.14.2.2 EtherNet/IP配置.....	398
7.14.2.3 EIP状态寄存器监控.....	403
7.14.2.4 Modbus网口配置与监控.....	405
7.14.3 二维视觉.....	405
7.14.4 从站动态配置.....	406
7.14.5 ERI激活功能.....	408
7.15 网络与时间.....	409
7.15.1 网络设置.....	410
7.15.2 时间设置.....	411
7.15.3 外部网络设置.....	411
7.16 控制器参数设置.....	415
7.16.1 控制器设置.....	416
7.16.2 远程设置.....	416
7.16.3 程序设置.....	417
7.16.4 语言.....	418
7.16.5 个性化.....	418
7.17 一点到达.....	419
7.18 锁机与解除.....	420
7.18.1 设置锁机期限.....	421
7.18.1.1 设置分期锁机期限.....	422
7.18.1.2 设置定时锁机期限.....	423
7.18.2 解除锁机期限.....	424
7.18.2.1 解除分期锁机期限.....	424
7.18.2.2 解除定时锁机期限.....	425
7.18.3 清除锁机期限.....	426
7.19 CodeSys功能授权.....	427
7.20 示教器分屏功能.....	430
7.21 轴锁功能.....	432
7.22 外部TCP功能.....	433
7.23 断点调试.....	435
7.24 PLC开放功能.....	435
7.25 倒序功能.....	435
7.26 飞拍功能.....	436
7.27 中断功能.....	439
7.28 常用指令组.....	441
7.29 外部接口管理权限.....	442
8 FAQ.....	445

8.1 如何根据铭牌信息查找对应控制柜说明书.....	445
8.2 埃斯顿机器人供电需求.....	445
8.3 如何处理机器人开机后显示急停处于按下状态.....	446
8.4 机器人如何进行周期性保养.....	447
8.5 如何备份机器人示教器程序.....	448
8.6 触发机器人急停按钮后如何恢复.....	448
8.7 使用时发现示教器不能进行程序编辑和修改怎么办.....	449
8.8 埃斯顿机器人目前支持的通讯有哪些.....	450
8.9 如何正常解决使用中机器人运行的点位发生了偏移.....	450
8.10 如何判定机器人开机后无法运动.....	451
8.11 如何解决机器人开机后示教器无法进入系统.....	452
附录 A ModbusTCP 控制接口数据表.....	454
附录 B 系统保留名称.....	458

## 图目录

图 1-1 控制柜外观及外部按键分布图.....	27
图 1-2 线缆与控制柜的连接图.....	37
图 2-1 关节坐标系下各轴的运动方向.....	39
图 2-2 世界坐标系下各轴的运动方向.....	39
图 2-3 工具坐标系方向定义.....	40
图 2-4 工具坐标系下的平行移动操作.....	41
图 2-5 示教程序运行路径示意.....	43
图 2-6 机器人控制器启停时序图.....	48
图 4-1 MovJ 指令插入效果图.....	88
图 5-1 工具和负载质量信息参考坐标系.....	125
图 5-2 工具和负载质量信息坐标系定义实例.....	125
图 6-1 MovArc运动轨迹.....	163
图 6-2 WristMove激光切割小圆示意图.....	216
图 7-1 力矩监控.....	336
图 7-2 电流监控.....	336
图 7-3 零位信息丢失.....	380
图 7-4 标准版.....	409
图 7-5 试用版.....	409
图 7-6 ESTUN视觉+机器人飞拍系统示意图.....	437
图 7-7 目标物体纠偏示意图.....	437

## 表目录

表 1-1 各组件的功能说明表.....	27
表 1-2 各功能按键功能说明表.....	28
表 1-3 各菜单按键功能说明表.....	29
表 1-4 各轴操作按键功能说明表.....	30
表 1-5 快捷按钮的功能说明表.....	31
表 1-6 各模式功能说明表.....	34
表 1-7 I/O端子排引脚信号定义表.....	35
表 1-8 RJ45引脚信号定义表.....	36
表 1-9 航插接口信号定义表.....	36
表 2-1 各类坐标系的定义及作用表.....	38
表 2-2 轴操作键对应的各轴动作说明表.....	39
表 2-3 轴操作键对应的各轴动作说明表.....	40
表 2-4 轴操作键对应的各轴动作说明表.....	40
表 3-1 核心功能组件说明表.....	49
表 4-1 变量的分类/类型说明表.....	94
表 6-1 运算符介绍.....	188
表 7-1 参数说明.....	320
表 7-2 输入类信号功能.....	387
表 7-3 输出类信号功能.....	388
表 8-1 控制柜命名规则.....	445

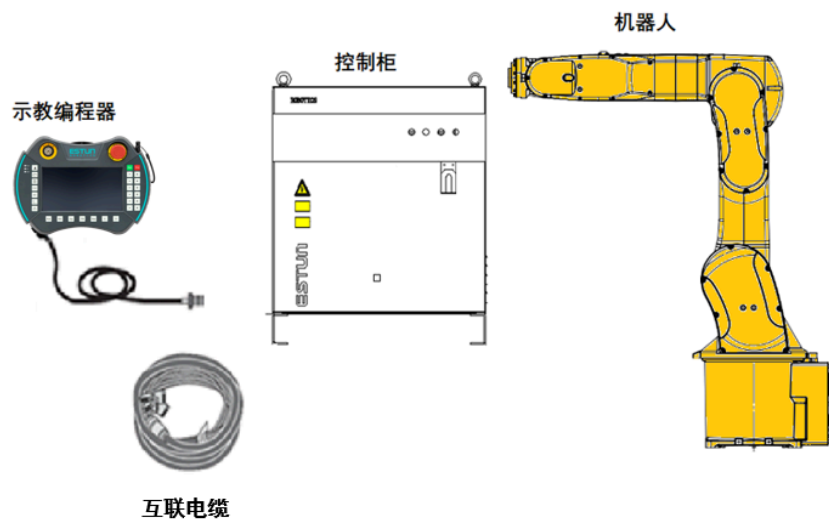
# 1 产品介绍

## 1.1 产品组成

### 1.1.1 装箱内容确认

产品到货后，请对照发货清单清点货物。标准发货清单包含以下 4 项内容（各货物详细信息将单独提供）：

- 机器人。
- 控制柜。
- 示教器。
- 互联线缆（机器人与控制柜间的连接线缆）。



### 1.1.2 订货号确认

确认机器人与控制柜上的**订货号**保持一致。

## 1.2 控制柜

本章节以ERC3-S1A系列控制柜为示例，仅对其外观结构及外部按键功能进行基础说明。

不同型号控制柜的外观按键布局与内部结构存在差异，详细技术参数与操作说明请参照对应型号的控制柜使用手册。

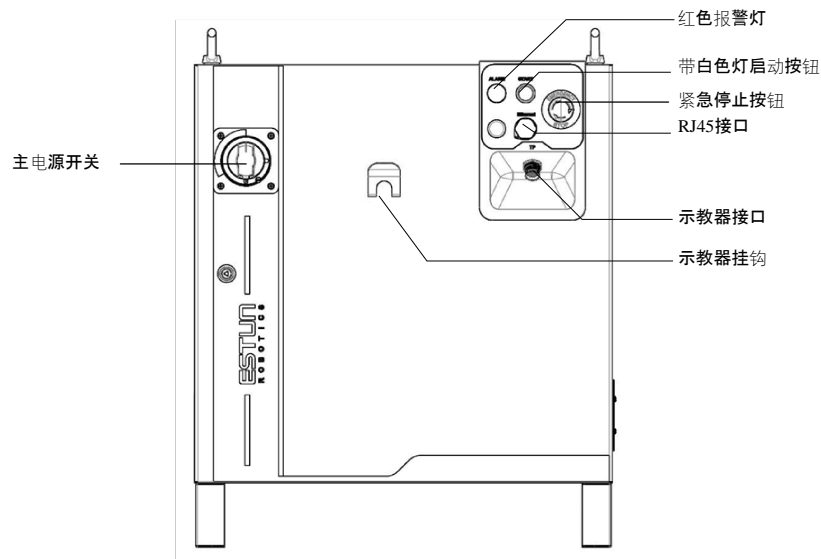


图 1-1 控制柜外观及外部按键分布图

### 说明

控制柜内部结构布局请参见对应型号控制柜手册。（不同型号机器人控制柜的内部布局存在差异）

## 1.3 示教器

### 1.3.1 外观结构

示教器主要由机器人示教、编程操作所需的功能按键及开关组成。

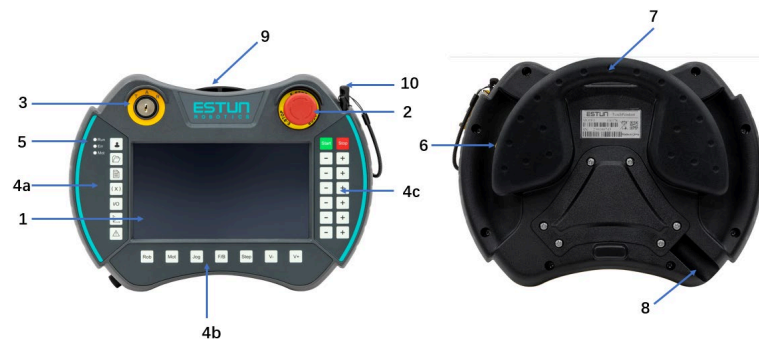


表 1-1 各组件的功能说明表

序号	说明
1	显示屏
2	紧急停止按钮

表 1-1 各组件的功能说明表 (续)

序号	说明
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	线缆接入区
9	USB 插槽
10	触摸笔

### 1.3.2 触摸笔

触摸笔为示教器的标准配件，收纳于示教器背部插槽，使用时直接取出即可。

#### 注意

操作示教器显示屏时，必须使用配套触摸笔，严禁使用螺丝刀等尖锐物品替代，以防划伤或损坏屏幕。

### 1.3.3 USB插槽

将 USB 存储设备接入本插槽，可实现程序文件、参数配置文件的读取与保存操作。

### 1.3.4 按键

#### 功能键

功能键位于示教器下方，各按键功能说明如表 1-2 各功能按键功能说明表所示。

表 1-2 各功能按键功能说明表



按键	功能说明
	快捷菜单键。
	伺服使能功能键。

表 1-2 各功能按键功能说明表 (续)

按键	功能说明
	坐标系选择功能键。 点动模式下切换机器人坐标系，支持关节坐标系、世界坐标系、用户坐标系、工具坐标系。
	倒序执行程序功能键。 手动模式下，按此键切换至程序倒序执行；再次按下，恢复程序正常顺序执行。
	示教再现运行方式选择功能键 示教再现运行时，切换程序运行方式，包含单步运行、连续运行两种模式。
	速度减功能键。 远程模式下，按下或点击此键降低全局运行速度。
	速度加功能键。 远程模式下，按下或点击此键提高全局运行速度。


## 菜单键

菜单键位于示教器左侧，各菜单按键功能说明如表 1-3 各菜单按键功能说明表所示。

表 1-3 各菜单按键功能说明表

按键	功能说明
	用户管理 进入菜单界面，可完成用户登录等操作
	工程管理 进入工程管理菜单界面，可实现工程的新建、删除、重命名、导入及导出等管理功能
	程序编辑 进入程序编辑菜单界面，可完成示教指令的插入、修改等程序编辑操作
	程序数据 进入程序数据菜单界面，可实现变量的新建、赋值及管理等操作
	I/O 检测 进入 I/O 检测菜单界面，可对系统 I/O 信号进行实时监视与故障诊断
	位置管理 进入位置管理菜单界面，可完成点动速度等示教操作参数的配置

表 1-3 各菜单按键功能说明表 (续)

按键	功能说明
	系统日志 进入系统日志菜单界面，可查看系统所有操作反馈信息，包含提示、警告、错误等类型

## 轴操作键

轴操作键位于示教器右侧，各轴操作按键功能说明如表 1-4 各轴操作按键功能说明表所示。

表 1-4 各轴操作按键功能说明表

按键	功能说明
	开始按钮：启动示教再现程序运行
	停止按钮：终止示教再现程序运行
	正方向点动按键 <ul style="list-style-type: none"> <li>关节坐标系下：控制对应关节向正方向运动</li> <li>世界坐标系下：控制机器人沿 X、Y、Z、A、B、C 轴正方向运动</li> <li>工具坐标系下：控制机器人沿 TX、TY、TZ、TA、TB、TC 轴正方向运动（坐标系相关介绍参见<a href="#">机器人坐标系</a>章节）</li> </ul>
	负方向点动按键 <ul style="list-style-type: none"> <li>关节坐标系下：控制对应关节向负方向运动</li> <li>世界坐标系下：控制机器人沿 X、Y、Z、A、B、C 轴负方向运动</li> <li>工具坐标系下：控制机器人沿 TX、TY、TZ、TA、TB、TC 轴负方向运动</li> </ul>

### 1.3.5 显示界面

#### 显示区域概述

示教器配备 7 英寸彩色显示屏，支持数字、字母及符号的清晰显示。

显示屏划分为 4 个功能显示区，分别为通用显示区、状态显示区、快捷按钮显示区及点动显示区。



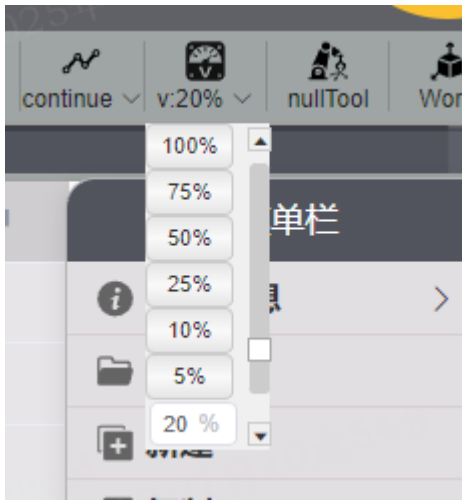

各显示区功能说明如下：

1. 通用显示区：用于显示各类操作菜单界面，支持程序、特性文件的查看、编辑及系统参数的配置操作。
2. 状态显示区：实时展示系统运行状态，包含运行模式、伺服使能状态、当前点动坐标系及点动速度等关键信息。
3. 快捷按钮显示区：显示当前页面下各快捷按钮的功能定义。

表 1-5 快捷按钮的功能说明表

状态	说明
模式指示	<p>示教器运行模式：</p> <p>A：自动模式，可通过“<b>系统 -&gt; 系统维护 -&gt; 系统管理</b>”页面中模式切换按钮来控制。</p> <p>T1：手动慢速，即相当于原来的手动模式。</p> <p>T2：手动高速，不能点动，但支持一点到达功能和回原点功能。</p> <p>AE：远程模式。</p>
伺服状态	<p>机器人伺服状态</p> <p>：伺服使能</p> <p>：伺服未使能</p> <p>：伺服使能，但存在零点丢失</p> <p>：伺服未使能，但存在零点丢失</p>

状态	说明
运行模式	<p>再现程序的运动模式</p> <p> : 单步运行模式</p> <p> : 连续运行模式</p> <p> : 倒序运行模式</p>
运行状态	<p>再现程序运行状态</p> <p> : 运行</p> <p> : 暂停</p> <p> : 停止</p>
急停状态	<p> : 急停按钮未按下</p> <p> : 急停按钮已按下</p>
使能状态	<p>显示示教器上的使能开关当前状态:</p> <p> : 使能开关未接通</p> <p> : 使能开关已接通</p>

状态	说明
当前速度	<p>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。</p>  <p>点击最下方的输入框，可在弹窗键盘中设置自定义全局速度，输入范围为 0-100 (%)。</p>
用户状态	<p>显示当前用户权限：</p> <ul style="list-style-type: none"> <li>：操作人员</li> <li>：编程人员</li> <li>：管理人员</li> <li>：厂家人员</li> </ul>

4. 点动显示区：显示当前点动操作的控制方式。例如显示 A1、A2、A3、A4、A5、A6 时，代表当前为关节点动模式。

### 1.3.6 模式开关说明

通过拨动钥匙或点击模式切换开关，可在系统四种运行模式之间进行切换，具体模式定义及操作规则如下：

1. 转动示教器上方钥匙，并点击系统维护模式切换按钮，可实现机器人自动模式与远程模式的相互切换。
2. 远程模式下，按下或点击示教器 **V+** 按键可提高全局速度，按下或点击 **V-** 按键可降低全局速度。
3. 全局速度最低支持 1%。

表 1-6 各模式功能说明表

模式类型	模式说明
T1 模式（手动慢速模式）	等效于原手动模式，机器人 TCP 末端运行速度不超过 250mm/s
T2 模式（手动全速模式）	解除系统底层速度限制，机器人 TCP 末端速度可达到系统支持的最大值
自动模式（再现模式）	用于复现已编制完成的示教程序
远程模式	支持调取系统存储器中的程序进行编辑与执行

### 说明

- 开启API模式：将系统切换到手动模式，点击示教器左上角菜单“配置 -> 远程设置”，进入远程设置页面，将API模式切换到ON，示教器再次切换到远程模式进入API模式。
- 关闭API模式：将系统切换到手动模式，点击示教器左上角菜单“配置 -> 远程设置”，进入远程设置页面，将API模式切换到OFF，即可退出API模式。

## 程序类型定义与切换规则

1. 编辑程序：指在示教模式下，示教器界面所显示的程序，为程序编辑的操作对象。
2. 执行程序：指在再现模式下，示教器界面所显示的程序，为程序运行的执行对象。

切换规则：在示教器上切换动作模式时，编辑程序与执行程序不会自动转换。若需在再现模式下运行某一编辑程序，需先将模式切换为再现模式，再调取目标程序。

### 1.3.7 急停按钮

急停按钮为安全保护装置，按下后按钮将自动锁定，系统会立即切断伺服电源，机器人随即停止所有动作，无法进行任何操作。

遇紧急情况时，可快速按下此按钮，实现对人员及设备的安全保护。

#### 安装位置：

- 控制柜前面板。
- 示教器面板。

#### 操作说明：

1. 触发急停：直接按下控制柜前面板或示教器上的急停按钮即可。
2. 解除急停：按照按钮上的箭头指示方向旋转，即可解锁按钮并解除急停状态（操作示意图如下）。



### 1.3.8 伺服使能开关

伺服使能开关为示教与点动操作的必备控制装置，用于控制伺服使能信号的接通与断开，仅在按下开关时，系统才能为机器人伺服电机提供励磁，实现机器人的运动控制。

#### 安装位置：

示教器右侧把手处。

#### 操作说明：

1. 接通使能信号：按下开关至第一接触档位，伺服使能信号接通，机器人进入可操作状态。
2. 断开使能信号：将开关按到底，或直接松开开关按钮，伺服使能信号断开，机器人停止运动控制。

### 1.3.9 示教器内部的接线端子信号定义

表 1-7 I/O端子排引脚信号定义表

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)
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 1-8 RJ45引脚信号定义表

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

### 1.3.10 示教器线缆

表 1-9 航插接口信号定义表

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)
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)

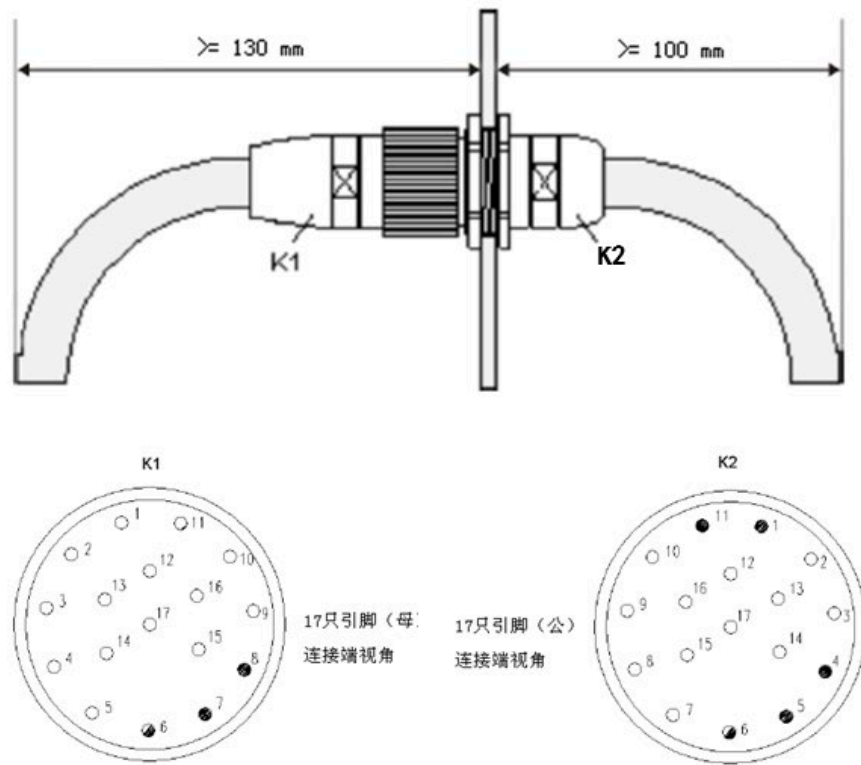


图 1-2 线缆与控制柜的连接图

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。

## 2 产品使用

产品使用前，需完成全部安装与接线工作。

### 2.1 连接方法

使用产品前，应确认机器人、控制柜及示教器的连接状态正常，设备处于可用状态。详细连接规范请参照埃斯顿机器人iER系列本体使用手册。

机器人、控制柜与示教器的连接步骤如下：

1. 拆除设备包装，将电源线缆连接至控制柜电源接口。
2. 使用专用互联线缆，完成机器人与控制柜的连接。
3. 确认线缆接头的规格尺寸与机器人对应插座的匹配性，将线缆插入插座并紧固，确保连接可靠。
4. 将示教器的线缆连接至控制柜的示教器专用接口。

### 2.2 操作说明

#### 2.2.1 机器人坐标系

机器人常用坐标系分为关节坐标系、世界坐标系、工具坐标系三类，各类坐标系的定义及作用如表 2-1 各类坐标系的定义及作用表所示：

表 2-1 各类坐标系的定义及作用表

坐标系名称	定义	作用
关节坐标系	机器人各关节轴（A1-A6）独立运动，互不干涉	实现 A1、A2、A3、A4、A5、A6 各轴的单独驱动与动作控制
世界坐标系	以固定的空间基准为参照，机器人末端控制点可沿 X、Y、Z 轴平行移动及旋转	控制机器人末端控制点沿空间 X、Y、Z 轴完成平移与旋转动作
工具坐标系	以机器人腕部法兰盘所持工具的有效方向为 Z 轴，坐标原点定义在工具尖端点	控制机器人末端控制点沿工具尖端点的 X、Y、Z 轴进行平行移动

#### 关节坐标系

采用关节坐标系时，机器人的 A1-A6 各关节轴独立运动，互不影响。

关节坐标系下各轴的运动方向如下图所示。

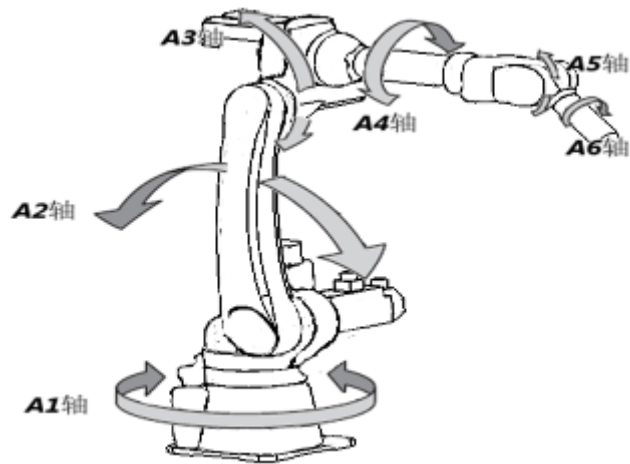


图 2-1 关节坐标系下各轴的运动方向

表 2-2 轴操作键对应的各轴动作说明表

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

## 世界坐标系

采用世界坐标系时，机器人末端控制点以固定空间基准为参照，沿 X、Y、Z 轴进行平行移动。

世界坐标系下各轴的运动方向如下图所示。

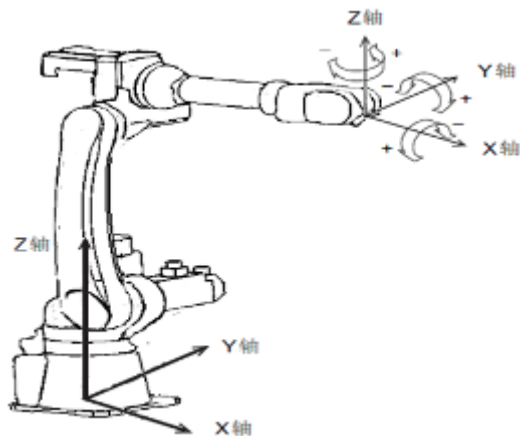


图 2-2 世界坐标系下各轴的运动方向

表 2-3 轴操作键对应的各轴动作说明表

轴名称	动作
X	沿世界坐标系X方向运动
Y	沿世界坐标系Y方向运动
Z	沿世界坐标系Z方向运动
A	沿世界坐标系X方向运动
B	沿世界坐标系Y方向运动
C	沿世界坐标系Z方向运动

## 工具坐标系

工具坐标系的基准以末端工具为核心，将工具有效方向设为 Z 轴，坐标原点定位在工具尖端点。

采用工具坐标系时，机器人末端控制点沿工具尖端点的 X、Y、Z 轴平行移动，其运动姿态与机器人本体的位置、姿势无关。

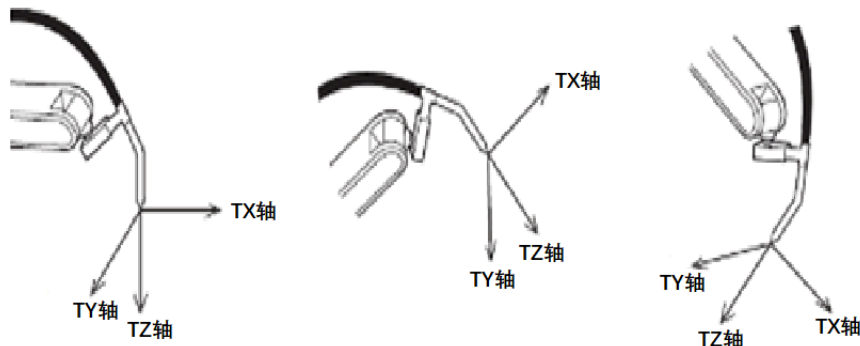


图 2-3 工具坐标系方向定义

表 2-4 轴操作键对应的各轴动作说明表

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	沿工具坐标系X方向旋转
TB	沿工具坐标系Y方向旋转
TC	沿工具坐标系Z方向旋转

该坐标系适用于保持工具姿态不变，相对工件进行平行移动的作业场景。

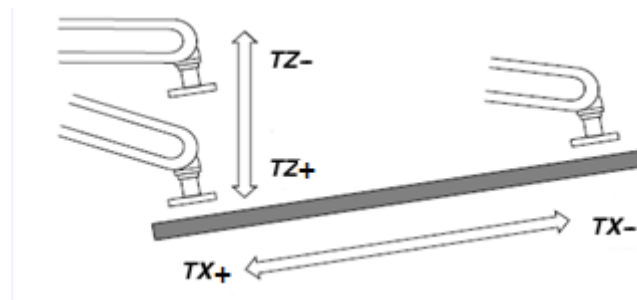


图 2-4 工具坐标系下的平行移动操作

## 2.3 示教操作

### 示教注意事项

- 模式与安全开关说明：  
将示教器模式旋钮切换至手动模式；安全开关为两级触发式设计，若发生误操作时操作人员紧握开关至第二级，系统将立即切断伺服电源。
- 急停功能确认：  
操作机器人前，需按如下步骤确认急停键有效性，详见下表：

序号	项目	说明
1	触发急停	按下控制柜或示教器上的急停键。
2	确认伺服断电	伺服电源接通时，示教器“伺服通电”图标亮起；按下急停键后，图标熄灭则说明断电有效。
3	恢复伺服励磁	确认急停功能正常后，操作按键  使伺服电源励磁，此时示教器 Mot 灯常亮。

### 2.3.1 示教前准备



开始示教前，需完成以下准备工作：

1. 检查控制柜与机器人本体状态：确认电源线、伺服端口、控制器端口连接正确，机械本体无异物；
2. 启动控制柜，确认各模块工作正常（详见[控制柜](#)章节）；
3. 接通伺服主电，测试控制柜及示教器急停按键的有效性；
4. 将示教器模式开关切换至示教模式；
5. 按下手压开关，即可通过“轴操作键”进行轴运动控制（轴运动规则详见[机器人坐标系](#)章节）。

## 2.3.2 示教基本步骤

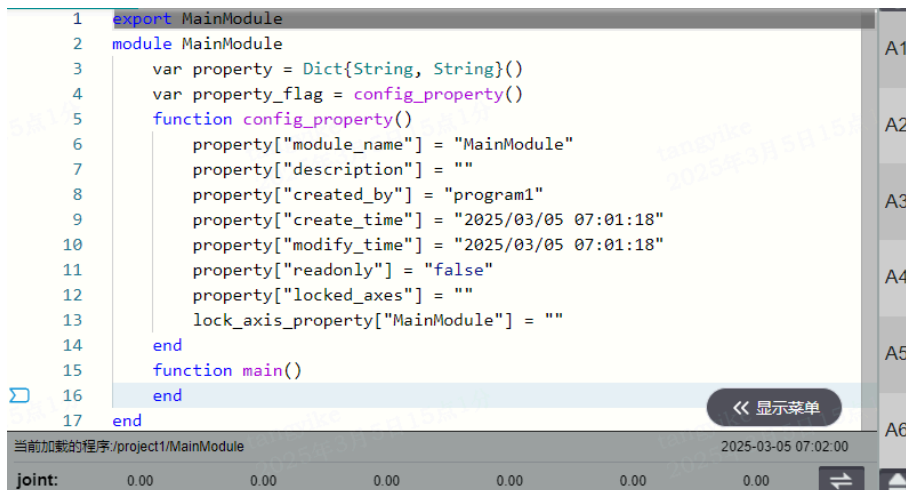
### 2.3.2.1 新建示教工程

示教操作可通过指令记录机器人运动轨迹，再现时按指令复现动作，新建工程步骤如下：

1. 按下  或点击  按键，选择“编程 -> 工程管理”进入工程管理界面。
2. 点击菜单栏“新建”，弹出工程创建对话框。



3. 在“域”菜单栏中选择“工程”。
4. 点击“名称”输入框，通过软键盘输入工程名（示例：project1）。
5. 点击“创建者”输入框，通过软键盘输入程序名（示例：program1）。
6. 点击“确认”完成创建，工程管理界面将显示新建工程“project1”。
7. 点击 project1 左侧下拉键，选择“er\_main”，点击菜单栏“打开”进入程序编辑界面。



```
1 export MainModule
2 module MainModule
3     var property = Dict{String, String}()
4     var property_flag = config_property()
5     function config_property()
6         property["module_name"] = "MainModule"
7         property["description"] = ""
8         property["created_by"] = "program1"
9         property["create_time"] = "2025/03/05 07:01:18"
10        property["modify_time"] = "2025/03/05 07:01:18"
11        property["readonly"] = "false"
12        property["locked_axes"] = ""
13        lock_axis_property["MainModule"] = ""
14    end
15    function main()
16    end
17 end
```

当前加载的程序: /project1/MainModule 2025-03-05 07:02:00

joint: 0.00 0.00 0.00 0.00 0.00 0.00

8. 在程序编辑界面编写运动指令，运动指令包含目标位置、插补方式、再现速度等参数，示例：

```
MovJ (P1,V50,"RELATIVE",C0) // 关节插补（点到点）运动
```

```
MovL (P2, V50, "RELATIVE",C0) // 直线插补运动
```

### 2.3.2.2 编制示教程序

#### 说明

本章节以工件 A→B 点作业程序为例进行说明。

示教前需确认操作员与机器人保持安全距离，该程序包含 6 个程序点，编制步骤如下：

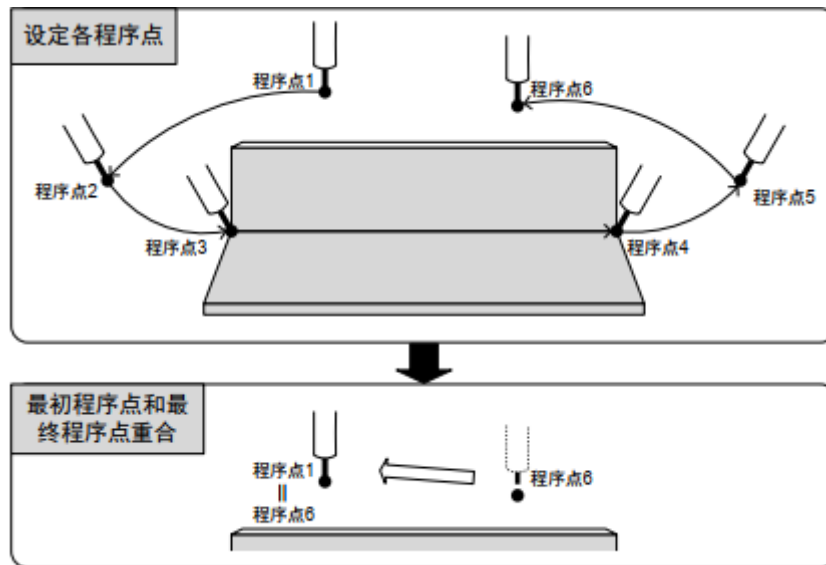





图 2-5 示教程序运行路径示意

程序点	位置说明	示教步骤	生成指令示例
1 (开始位置)	远离周边物体的安全位置	<ol style="list-style-type: none"> <li>1. 按住手压开关，通过轴操作键将机器人移至目标位置；</li> <li>2. 在程序编辑界面选中“end”行，点击“新建 -&gt; 运动指令 -&gt; MovJ”。</li> <li>3. 点击“示教”记录 P0 点位姿，确认生成指令。</li> </ol>	MovJ(p0,v=5.V50,b="RELATIVE")

程序点	位置说明	示教步骤	生成指令示例
2 (作业开始位置附近)	作业姿态对应的位置	保持作业姿态, 按程序点 1 的方法示教。	MovJ(p2,v=S.V50,b="RELATIVE")
3 (作业开始位置)	保持程序点 2 姿态的作业起始位	1. 设定世界坐标系, 移至目标位置。 2. 选中“end”行, 点击“新建 -> 运动指令 -> MovL”。 3. 确认生成指令。	MovL(cp0,v=S.V50,b="RELATIVE")
4 (作业结束位置)	远离工件的作业终止位	移至目标位置, 按程序点 1 方法添加 MovL 指令。	MovL(p4,v=S.V50,b="RELATIVE")
5 (安全位置)	不碰触工件 / 夹具的位置	移至目标位置, 按程序点 1 方法添加 MovL 指令。	MovL(p5,v=S.V50,b="RELATIVE")
6 (开始位置附近)	关节坐标系下的起始位附近	关节坐标系下移至目标位置, 按程序点 1 方法添加 MovJ 指令。	MovJ(p6,v=S.V50,b="RELATIVE")

## 轨迹确认

示教完成后需验证轨迹合理性, 操作步骤如下:

1. 将示教器模式切换到手动, 清除示教器上的所有报警信息, 确认机器人处于安全可运行的位置。
2. 将光标移至程序点 1, 按下“PC”键
3. 按 , 切换至单步运行模式。
4. 调节速度增减键, 将速度设定在 10% 以内。
5. 按下手压开关使伺服使能。
6. 按运行键  单步执行程序 (每按一次运行一条指令)。
7. 单步确认完成后, 将光标移至程序起始处。
8. 按 , 切换至连续模式, 运行全部程序, 验证完整轨迹。


## 说明

速度限制: 示教模式下运行轨迹, 线速度 $\leq 250\text{mm/s}$ , 姿态速度 $\leq 180\text{deg/s}$  (符合国标要求)。

### 2.3.2.3 修改程序

若实际轨迹与预期不符，可通过以下方式调整：

#### 加载程序

1. 按下  或者点击“工程管理”进入工程管理界面。
2. 选择待编辑程序，点击菜单栏“加载”进入编辑界面。

#### 修改程序点位置（以程序点 2 为例）

1. 用轴操作键将机器人移至新位置，记录屏幕下方的位置坐标。
2. 选中程序点 2 所在指令，点击菜单栏“修改”。
3. 在弹窗中选择目标位置 p2，输入记录的坐标值。
4. 点击“确定”完成修改。

#### 插入程序点（以程序点 5、6 间插入为例）

1. 选中程序点 6 所在指令，点击菜单栏“新建”。
2. 参照程序点 1 的添加方法，插入 MovJ 关节插补指令。

#### 删除程序点

进入程序编辑界面，选中需删除的指令，点击右侧菜单栏“删除”。

## 2.4 程序再现

执行已经示教完成的程序称为再现。

### 2.4.1 再现前准备

---



#### 注意

1. 操作前需确认机器人工作区域内无操作人员及其他无关人员。
  2. 确认控制柜及示教器上的急停按钮处于非触发状态。
- 

再现前准备步骤：

1. 确认所有人员处于机器人工作区域外。
2. 检查并确认控制柜、示教器的急停按钮均未被触发。
3. 加载需要执行的目标工程文件。
4. 选中待执行的程序，按下 PC 键，确保程序从首行开始执行。
5. 通过速度增减键，将程序运行速度调整至合适值。

## 2.4.2 再现执行步骤

1. 将示教器的模式旋钮切换至自动模式，并将光标移至程序起始行。
2. 接通伺服主电源，按下 ，使伺服电源励磁。
3. 按下  按键，机器人将执行再现程序，完成一个循环后自动停止。

## 2.4.3 停止与再启动规则

机器人在动作过程中停止或自动停止，主要分为以下几种情况：

- 暂停操作或急停操作。
- 报警触发的停止。
- 其他指令或操作触发的停止。

## 启停安全规范

1. 禁止频繁通断电源，否则易损坏驱动模块；每次电源开启与关闭的时间间隔需不少于 60s。
2. 断开电源前，必须先停止设备运行程序，待机器人完全静止后，再切断电源。

## 暂停操作与解除

### 示教器操作暂停

- 暂停：按下示教器上的暂停按键，机器人立即停止动作。
- 解除：按下示教器上的启动按键，机器人从暂停位置继续执行程序。

### 外部输入信号暂停（专用接口）

- 暂停：触发外部控制装置的停止信号，机器人立即停止动作。
- 解除：触发外部控制装置的启动信号，机器人从暂停位置继续执行程序。

### 暂停转停止的触发条件

程序处于暂停状态时，执行以下任一操作，程序将自动切换为停止状态：

- 程序上下文变更，如工程加载 / 注销、指令新增或删除。
- 执行 PC 键操作。
- 切换系统运行模式。
- 程序执行至 End 指令。
- 执行点动操作或回零操作。
- 执行其他系统设置类操作。

## 急停操作与再启动

### 急停触发位置

可通过以下三处执行急停操作：

- 示教器急停按钮。
- 控制柜前面板急停旋钮。
- 外部专用急停输入信号。

### 急停与解除步骤

- 急停触发：按下示教器或控制柜的急停键，机器人主电源立即切断，动作瞬时停止。
- 急停解除：先按急停键上的箭头指示方向旋转解锁，再按下对应按键重新给伺服上电。

### 急停后再启动注意事项

- 再启动前，需通过点动操作确认机器人当前位置，排查是否存在与工件、夹具的干涉风险。
- 连续程序在高速再现过程中触发急停时，机器人实际停止位置可能与当前显示程序点存在偏差（通常超前 2-3 个程序点），再启动前需重点核查此区域的干涉风险。

## 报警引起的停止与解除

1. 机器人动作过程中触发报警时，将立即停止动作，示教器自动跳转至报警画面。操作人员可进入系统日志界面，查看报警详细信息。
2. 不同类型报警的解除方法如下表所示：

报警种类	报警解除方法
轻故障报警	在示教器报警画面中，选择“清除报警”选项，解除报警状态。
重故障报警	切断伺服主电源，排查并消除报警诱因后，再重新上电

## 其他类型暂停与启动

1. 模式切换触发暂停
  - 触发条件：再现过程中，将模式从自动模式切换为示教模式。
  - 启动方法：将模式切换回自动模式，按下启动按键即可继续运行。
2. PAUSE指令触发暂停
  - 触发条件：程序执行至 PAUSE 指令。
  - 启动方法：按下启动按键，机器人从 PAUSE 指令的下一条指令开始继续运行。

### 2.4.4 再现速度修改方法

操作人员可通过速度增减键调整程序再现速度，具体要求如下：

1. 程序再现过程中可实时修改速度，无需暂停程序。
2. 建议先通过多次试验性调速，确认机器人动作无异常后，再确定最终运行速度。

## 2.5 启停时序

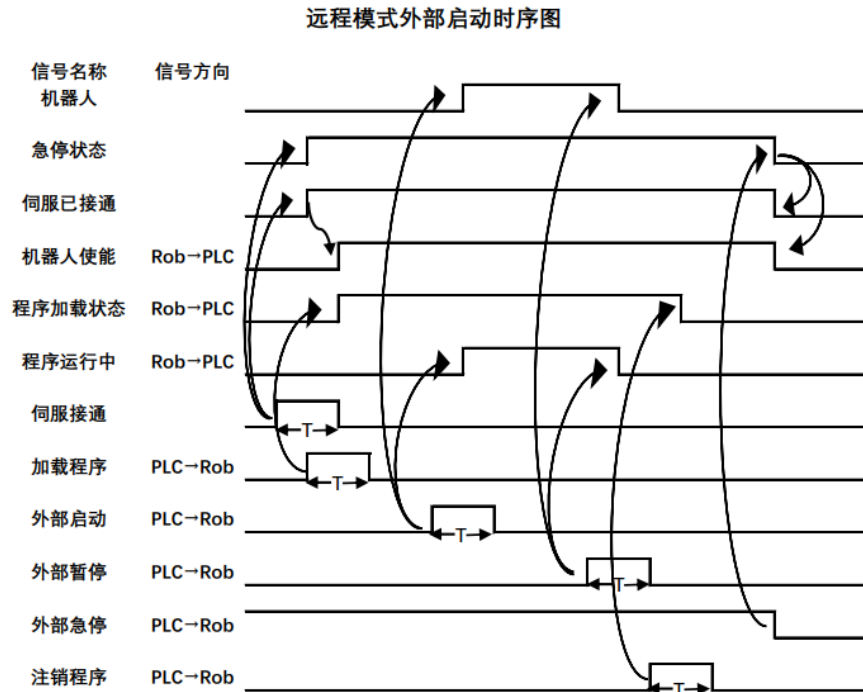


图 2-6 机器人控制器启停时序图

## 3 系统功能介绍

机器人控制器与用户的交互主要通过示教器实现。示教器是机器人控制系统的核心组成部分，操作人员可通过示教器完成手动示教，控制机器人运动至目标位姿，并记录各点位的坐标数据。同时可依托机器人语言开展在线编程，实现程序的存储与回放，驱动机器人按照预设轨迹完成作业任务。

用户可通过示教器的物理按键或触摸操作面板，实现与控制系统的指令交互与信息读取。

ESTUN公司配置的示教器，其核心功能组件包括：

表 3-1 核心功能组件说明表

功能名称	功能说明	快捷按键
用户管理	此菜单界面实现用户登录等功能	
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	

### 3.1 系统设置

#### 3.1.1 用户管理

在左侧菜单栏上点击“用户登录”，进入用户管理界面。



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”。各用户角色权限介绍如下表所示。

### 📖 说明

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。

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。

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”。

3. 切换用户：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。



## 3.1.2 机器人设置

### 3.1.2.1 机器人设置

点击菜单栏“配置 -> 机器人设置”。



参数	参数介绍	操作方式
机器人名称	设置机器人的名称	点击输入框，在弹出的软键盘上输入自定义“机器人名称”，输入完成后，点击“OK” 根据提示“请重启系统”。
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：悬挂、地装和壁装	点击下拉菜单，选择需要的安装方式。 根据提示“请重启系统”。
绕Y旋转	机器人安装方式选择壁装后，可设置机器人的安装角度。	设置后根据提示重启系统。
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	-

### 📖 说明

当机器人配置文件中WallMountParam.ini文件不存在时，机器人安装方式只支持正装和倒装（地装和悬挂）。

### 3.1.2.2 本体参数



根据实际需求设置“机器人结构参数”、“关节耦合”、“实虚轴”。

### 说明

仅当前机型为SCARA时，才需要设置机械人补偿参数。请根据实际需求设置“机器人补偿参数”，修改参数后需重启系统，方可生效。

### 3.1.2.3 运动学参数

根据实际需求设置“点动JOG参数”、“自动运行参数”、“关节参数”、“到位参数”。



### 说明

修改后需要重启示教器，方可生效。

点动跨奇异功能仅适用于六关节机器人点动XYZ方向的腕部跨奇异。

### 3.1.2.4 动力学参数

根据实际需求设置“前馈参数”。



**说明**

修改后需要重启示教器，方可生效。

### 3.1.2.5 高级运动参数

根据实际需求设置“高级运动参数”、“Nominal规划参数”。



**说明**

修改后需要重启示教器，方可生效。

### 3.1.2.6 高级功能参数

根据实际需求设置“碰撞参数”和“软浮动参数”。



#### 说明

修改后需要重启示教器，方可生效。

### 3.1.2.7 安全参数

根据实际需求设置“安全类参数”、“运动检测参数”、“关节力矩过载倍数”、“关节限位”。



### 说明

修改后需要重启示教器，方可生效。

## 3.1.2.8 厂家参数

根据实际需求设置“系统参数”、“控制柜参数”、“关节碰撞参数”、“关节Mdl参数”。



### 说明

修改后需要重启示教器，方可生效。

## 3.2 系统维护

### 3.2.1 系统管理

点击菜单“系统 -> 系统维护”，选择系统管理。该子菜单分类下，用户可对示教器及控制器系统进行关闭和重启操作。




- 关闭示教器：点击“关闭示教器”，此操作会关闭示教器。
- 重启示教器：点击“重启示教器”，此操作会重启示教器。
- 关闭控制系统：点击“关闭控制系统”，此操作会关闭控制器。
- 重启控制系统：点击“重启控制系统”，此操作会重启控制器。
- 恢复出厂设置：点击“恢复出厂设置”，此操作会删除工程文件和清除所有警报，将运动相关系统参数恢复为出厂状态。

### 说明

- 恢复出厂设置功能，删除程序启动后生成的文件，将runtime包恢复到出厂状态。
- 恢复出厂设置后，runtime包中的project工程文件为初始状态，用户新建的工程被删除。
- 恢复出厂设置后，cfg\_user文件夹中默认只保留motion\_cfg.json和ER\_Adapter.eds两个文件，其余文件被删除，如有特殊需要保留的文件，可以配置在runtime/cfg\_default/proj/ recovery\_init文件中，输入文件名即可。
- 恢复出厂设置后，/mnt/sdcard/log 文件夹下面所有的日志文件被清除。

## 3.2.2 示教器升级

1. 点击按钮，选择“系统 -> 系统维护”进入示教器更新界面。
2. 将U盘插入示教器中，通过示教器页面进行示教器的版本更新。

3. 打开示教器主菜单，选中“系统 -> 系统维护”，进入系统维护界面。



4. 在左侧菜单选择“系统升级”，进入示教器升级页面。



5. 点击“选择文件”输入框，系统将自动识别U盘中的升级文件（仅显示.patch后缀文件）  
如未检测到U盘，将弹出U盘不存在提示。
6. 选择示教器版本文件，点击“确定”，可选择示教器更新文件。



7. 点击“更新”，将进行示教器的更新操作，会弹出更新状态提示。



8. 示教器更新完成后，系统弹出“升级完成”提示，点击“确定”重启示教器，示教器更新操作完成。

### 3.2.3 控制器程序的更新

### 3.2.3.1 控制器升级

1. 进入示教器“系统 -> 系统维护”页面，选中左侧“控制器升级”。
2. 点击“选择位置”，可选“本地”或“U盘”升级，U盘升级需要满足升级包命名规则符合xxxx.runtime 后缀。
  - 本地升级：直接点击“更新”，系统自动进入更新流程，同步显示升级进度。
  - U盘升级：将存储有.runtime后缀升级包的U盘插入示教器，点击“选择文件”，系统过滤显示U盘中的.runtime后缀文件，选中目标文件后点击“确定”。



3. 如果未检测到U盘，系统将会弹出异常提示，需检查U盘连接后重新操作。
4. 点击“更新”，系统开始执行升级操作，同步显示文件导入及升级进度。

5. 升级成功后，系统弹出提示“升级成功，请重启控制系统以生效”，点击“确定”，重启控制器后升级生效。



### 3.2.3.2 控制器补丁升级

1. 选择“控制器升级”。
2. 进入示教器“系统 -> 系统维护”页面，选中左侧“控制器升级”。
3. 点击“选择位置”，可选“本地”或“U盘”升级：
  - 本地升级：直接点击“更新”，系统自动进入补丁更新流程，同步显示升级进度。
  - U盘升级：将存储有.patch后缀补丁包的U盘插入示教器，点击“选择文件”，系统过滤显示U盘中的.patch后缀文件，选中目标文件后点击“确定”。



4. 若未检测到U盘，系统将弹出异常提示，需检查U盘连接后重新操作。
5. 点击“更新”，系统开始执行补丁升级操作，同步显示文件导入及升级进度。
6. 升级成功后，系统弹出提示“升级成功，请重启控制系统以生效”，点击“确定”，重启控制器后补丁升级生效。



### 3.2.4 备份恢复

备份恢复功能支持将指定内容备份至不同位置，同时可将备份内容恢复至当前控制器。备份位置包括本地、U盘、FTP服务器、Share文件夹；备份内容包括运行时包（runtime）、用户参数、工程程序、日志。

---

#### 注意

- 涉及U盘的所有备份/恢复操作，需确保SSH状态灯亮起，其他操作需确保Update升级服务状态灯亮起。
  - 备份内容选择“**运行时包**”时，已包含用户参数、工程文件、日志等所有内容，因此选择运行时包后，不可再选择其他备份内容。
- 

#### 3.2.4.1 手动备份

##### 功能简介

手动备份可根据实际需求，随时对指定内容进行备份，支持4种备份位置。

##### 本地备份

1. 进入示教器“**系统 -> 系统维护 -> 备份恢复**”页面，选择“**手动**”备份模式，备份位置选择“**本地**”。
2. 在“**备份内容**”中，勾选需要备份的内容。
3. 点击“**开始备份**”，控制器UpdateServer开始备份，并同步显示当前备份进度。



4. 进度显示100%且无异常提示，即完成本地手动备份。

## U盘备份

1. 进入示教器“系统 -> 系统维护 -> 备份恢复”页面，选择“手动”备份模式，备份位置选择“U盘”。
2. 默认备份当前U盘根路径，支持手动修改。
3. 在“备份内容”中，勾选需要备份的内容，
4. 点击“开始备份”，控制器UpdateServer开始备份，并同步显示当前备份进度。
5. 进度显示100%且无异常提示，即完成U盘手动备份。

## Share备份

1. 进入示教器“系统 -> 系统维护 -> 备份恢复”页面，选择“手动”备份模式，备份位置选择“Share”。
2. 在“备份内容”中，勾选需要备份的内容。
3. 备份路径默认使用配置文件路径和地址，支持手动修改配置路径，点击“修改远程配置”可以修改share IP地址。



4. 点击“**开始备份**”，控制器UpdateServer开始备份，并同步显示当前备份进度。
5. 进度显示100%且无异常提示，即完成Share手动备份。

## FTP备份

1. 进入示教器“**系统 -> 系统维护 -> 备份恢复**”页面，选择“**手动**”备份模式，备份位置选择“**FTP**”。
2. 在“**备份内容**”中，勾选需要备份的内容。
3. 备份路径默认使用配置文件路径和地址，支持手动修改配置路径，点击“**修改远程配置**”可以修改FTP配置信息。



4. 点击“开始备份”，控制器UpdateServer开始备份，并同步显示当前备份进度。
5. 进度显示100%且无异常提示，即完成FTP手动备份。

### 3.2.4.2 自动备份

#### 功能简介

自动备份可设置周期性备份任务，支持本地、Share文件夹、FTP服务器三种备份位置；每种备份位置最多可开启2个周期备份任务，开启后将同步显示上次备份成功时间（未显示则代表当前周期未到）。

#### 📖 说明

自动备份参数修改前，需要关闭“使能开关”，使能开关开启后，无法修改任何备份参数。

#### 本地自动备份

1. 进入示教器“系统 -> 系统维护 -> 备份恢复”页面，选择“自动”备份模式，备份位置设置为“本地”。
2. 设置备份周期，勾选需要备份的内容。
3. 确认参数设置无误后，开启“使能开关”，系统将按照设定周期自动执行备份操作，备份内容保存至本地指定路径。

## Share自动备份

1. 进入示教器“系统 -> 系统维护 -> 备份恢复”页面，选择“自动”备份模式，备份位置设置为“Share”。
2. 设置备份周期，勾选需要备份的内容。
3. 若需要可以修改该周期配置的远程地址，点击“修改远程配置”，修改确认后生效，该IP仅对当前周期配置生效。



4. 开启“使能开关”，系统将按照设定周期自动执行备份操作，备份内容保存至Share文件夹指定路径。

## FTP自动备份

1. 进入示教器“系统 -> 系统维护 -> 备份恢复”页面，选择“自动”备份模式，备份位置设置为“FTP”。
2. 设置备份周期，勾选需要备份的内容。
3. 若需要可以修改该周期配置的远程地址，点击“修改远程配置”，修改确认后生效，该修改仅对当前周期配置生效。



4. 开启“使能开关”，系统将按照设定周期自动执行备份操作，备份内容保存至FTP服务器指定路径。

### 3.2.4.3 数据恢复

#### 功能简介

数据恢复可将已备份的内容（运行时包、用户参数、工程程序）恢复至当前控制器，恢复位置对应备份位置（本地、U盘、远程/Share/FTP），支持用户参数和数据恢复。

#### 📖 说明

- 运行时包和用户参数恢复后，需要断点重启控制器，系统将弹出对应提示，重启后恢复生效。
- 选择用户参数和工程程序时，每种类型最多只支持选择一个恢复。

#### 本地恢复

1. 进入示教器“系统 -> 系统维护 -> 备份恢复”页面，选择“数据恢复”，恢复位置设置为“本地”。
2. 默认显示“运行时包”的备份记录，可切换选择“用户参数”或“工程程序”界面将同步更新对应备份记录。
3. 从备份记录中，选中需要恢复的1条记录（每种内容类型仅可选择一条记录恢复）。

4. 点击“**开始恢复**”，控制器执行恢复操作，界面同步显示恢复进度。
5. 恢复成功后，系统提示本次操作成功；若恢复内容为运行时包或用户参数，提示重新上电后生效，恢复完成。

## U盘内容恢复

1. 进入示教器“**系统 -> 系统维护 -> 备份恢复**”页面，选择“**数据恢复**”，恢复位置设置为“**U盘**”。
2. 默认显示“**运行时包**”的备份记录，可切换选择“**用户参数**”或“**工程程序**”界面将同步更新对应备份记录。
3. 从备份记录中，选中需要恢复的1条记录（每种内容类型仅可选择一条记录恢复）。
4. 点击“**开始恢复**”，示教器将相应文件拷贝到控制器临时路径下，控制器执行恢复操作，界面同步显示恢复进度。
5. 恢复成功后，系统提示本次操作成功；若恢复内容为运行时包或用户参数，提示重新上电后生效，恢复完成。

## 远程内容恢复

1. 进入示教器“**系统 -> 系统维护 -> 备份恢复**”页面，选择“**数据恢复**”，恢复位置设置为“**远程**”。
2. 默认显示“**运行时包**”的备份记录，可切换选择“**用户参数**”或“**工程程序**”界面将同步更新对应备份记录。
3. 从备份记录中，选中需要恢复的1条记录（每种内容类型仅可选择一条记录恢复）。
4. 点击“**开始恢复**”，控制器执行恢复操作，界面同步显示恢复进度。
5. 恢复成功后，系统提示本次操作成功；若恢复内容为运行时包或用户参数，提示重新上电后生效，恢复完成。

### 3.2.5 版本信息

点击菜单“**系统 -> 系统维护**”，选择“**版本信息**”子菜单栏，在该界面右侧可查看当前系统版本的详细信息。点击“**详情**”，对版本信息进行折叠/展开。



- 查看系统信息：  
点击“系统信息”页签，查看当前系统版本号。点击“详情”，可展开/收起版本信息列表，列表内容包括当前hmi/控制器/内核/系统固件版本号等。版本信息列表默认折叠。
- 查看软件包配置：  
点击“应用库”页签，可查看当前系统软件包配置情况。支持名称关键词查询功能，筛选符合软件包。
- 查看库文件版本：  
点击“基础库”页签，可查看当前系统库文件版本情况。支持名称关键词查询功能，筛选符合库文件。

### 📖 说明

- 操作人员和编程人员权限下，仅显示系统版本号。
- 对于管理人员和厂家人员，可查看详细系统信息、软件包配置、组件库信息。

## 3.3 应用包安装及升级

软件的安装升级和卸载，需要对应功能只需要通过该功能安装即可使用。

### 3.3.1 应用软件管理

1. 通过“系统 -> 辅助软件”进入辅助软件安装管理页面。
2. 列表展示当前已经安装的所有软件，依次展示当前安装软件的名称、版本信息、激活状态等基础信息。



### 3.3.2 应用软件安装

1. 所有辅助软件通过U盘安装的方式安装。



2. 选择安装后会弹出U盘文件选择，选择对应文件后点击“确认”开始安装。



3. 示教器同步显示安装流程。
4. 安装成功后需要重启控制器生效。

### 3.3.3 应用软件卸载

1. 选择需要卸载的软件，选中后呈现勾选状态。
2. 点击“卸载”，在弹窗中点击“确定”卸载，并更新软件列表。



3. 重启控制器后卸载成功。

## 3.4 日志管理

按  或在界面菜单点击“日志及帮助 -> 日志”进入“日志管理”界面。

日志管理界面主要由当前报警、历史报警、日志、帮助四个页签组成。

### 3.4.1 报警

#### 当前报警

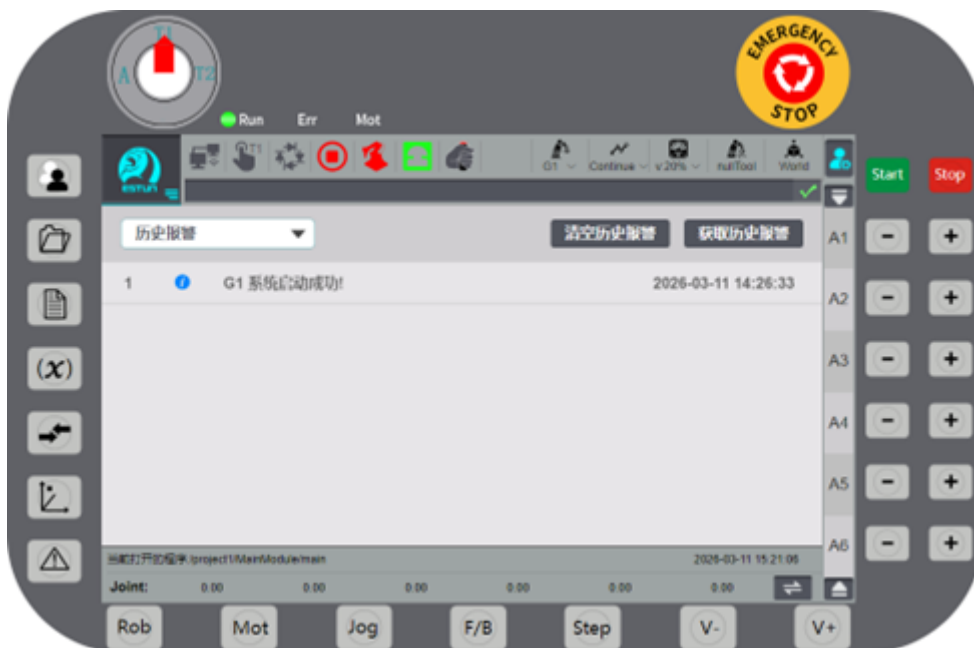
进入“日志管理”界面后，点击“报警”按钮，切换到报警界面，若当前存在需要清除的报警时，优先显示当前报警界面。



- 报警显示页面：支持看到报警产生时间、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。
- 清除当前报警：若存在当前报警，点击“清空当前报警”可清除当前报警，使程序正常运行。

## 历史报警

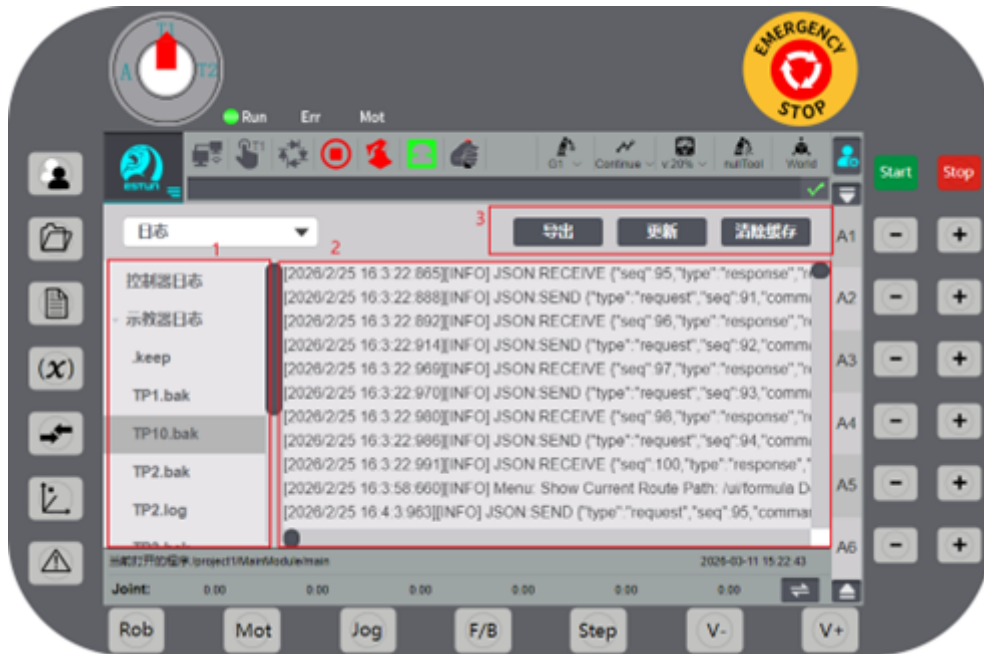
点击左上角下拉栏中的“历史报警”进入“历史报警”页面，可查看历史产生的报警。



- 清空历史报警：可清除所有的报警信息。
- 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。

### 3.4.2 查看日志

在“日志管理”界面点击左侧下拉栏里的“日志”按钮，切换至日志界面。



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. 显示界面：读取并显示对应日志文件的内容。
3. 日志操作：
  - 导出：弹出 U 盘目录浏览窗口如下图并选取需导出的文件夹路径。
  - 更新：更新控制器与内核日志文件并刷新日志文件列表。
  - 清空缓存：清空示教器本地缓存控制器日志文件。

### 3.4.3 查看帮助文档

在“日志管理”界面点击左侧下拉栏里的“帮助”按钮，切换至帮助界面。



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. 翻页按钮：输入页码或点击“首页”、“上页”、“下页”、“尾页”可进行翻页。

## 3.5 保养提示

### 功能简介

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。

在使用中分为本体保养和电池保养，以下为保养功能介绍：

- 累计时长：机器人保养部件周期内累计运行时长，当累计时长超过“周期时长”后，则会提示“已超过保养周期或日期，建议及时保养！”。
- 周期时长：每一款机型都对应有一个保养执行周期（10000h、11520h等），是按运行时间的保养周期。
- 下次保养日期：按自然年的保养日期，每一款机型都对应有一个累计使用年份（2年、3年等）
- 最近保养日期：即最近一次的保养日期。

## 操作步骤

1. 通过系统导航栏选择“系统 -> 维修保养”，进入“保养”界面。



2. 界面介绍：


- 轴号：对应机器人的各个轴号。
- 保养部件：对应机器人各个部件的保养类型，目前支持(减速机、氮气缸、丝杆)。
- 累计时长：周期内累计运行时长。
- 周期时长：按运行时间的保养周期。
- 下次保养日期：按自然年的保养日期。
- 最近保养日期：最近一次的保养日期。
- 复位：单轴的时间重置功能。

3. 操作介绍：

- 刷新：点击“刷新”，可重新获取当前的保养信息。
- 复位：点击“复位”，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。复位会将“累计时长”清零，并将“最近保养日期”以及“下次保养日期”设置为当前系统日期和下一次保养的日期。

## 4 工程管理

### 4.1 工程管理

按  或点击示教器左上角菜单“编程 -> 工程管理”，即可进入工程管理界面。

该界面主要由工程显示界面、底部状态及坐标显示栏、右侧可隐藏菜单栏三部分组成。




各界面区域功能说明：

1. 工程显示界面：用于展示已创建工程的名称，以及各工程的当前基本状态。
2. 右侧可隐藏菜单栏：集成工程与程序管理的核心功能，各功能按钮的操作说明如下：

功能按钮	操作说明
加载	载入已有的工程或程序文件，支持直接加载目标程序
卸载	与“加载”功能配合使用，用于卸载当前已加载的工程
打开	<p>仅支持打开程序文件。 打开程序文件时，需先加载对应工程。</p> <hr/> <p><b>⚠ 警告</b></p> <p>“打开”不等于“加载”，直接运行未加载的程序会触发“请先导入程序”的提示。</p> <hr/>
新建	用于创建新工程，或在已有工程下新建程序文件
复制	复制已有的工程或程序文件。 执行粘贴操作后，需按界面提示为复制的文件重新命名
粘贴	将已复制的工程或程序文件粘贴至当前界面

功能按钮	操作说明
删除	删除不需要的工程或程序文件
恢复	可恢复用户最近一次删除的程序相关文件，包括工程、模块及函数
自启动	配置远程模式下自动加载的程序文件。 若当前运行程序与自启动配置程序不一致，系统将弹出提示弹窗
重命名	修改工程或程序的名称，名称修改时不区分字母大小写
刷新	刷新当前工程管理界面的显示内容
查询	快速检索目标程序文件
写保护	为指定程序设置只读权限，禁止修改操作，具体规则如下： <ul style="list-style-type: none"> <li>• 开启：选中目标程序，点击“写保护”按钮，按提示完成设置。程序开启写保护后，其内部内容及关联的全局变量、工程变量、程序变量均不可修改</li> <li>• 取消：选中已开启保护的程序，点击“取消写保护”按钮，按提示完成权限解除</li> <li>• 权限限制：仅管理人员和厂家人员可执行此项操作</li> </ul>
程序信息	查看选中程序的当前状态信息，同时支持为程序添加备注信息

#### 4.1.1 新建工程/程序

1. 按下  按键或点击示教器左上角菜单“编程 -> 工程管理”，进入工程管理界面。点击右侧可隐藏菜单栏的“刷新”按钮，更新工程列表。



2. 点击可隐藏菜单栏的“新建”按钮，弹出“新建程序”对话框。



### 3. 选择新建类型

- 新建工程：“域”参数请选择“工程”。
- 在已有工程中新建程序：“域”参数请选择“模块”，直接跳转至步骤步骤5。或先在工程列表中选中目标工程，再执行新建操作。

#### 说明

若工程列表为空，系统默认且仅提供“新建工程”。

4. 点击“名称”输入框，通过弹出的软键盘输入工程名称（示例：project0）。
5. 点击“创建者”输入框，通过弹出的软键盘输入程序名称（示例：program0）。
6. 点击“确认”按钮，完成工程 / 程序的新建，系统自动跳转至工程显示界面。

## 4.1.2 删除工程/模块/函数

1. 在工程管理界面选中待删除的工程或程序（示例：project0）。

#### 注意

- a. 处于加载状态的工程或程序无法直接删除，需先点击可隐藏菜单栏的**卸载**按钮，解除加载状态后再执行删除操作。
- b. 删除程序时，其所属工程需先完成卸载，否则无法执行删除操作。

2. 点击可隐藏菜单栏的“删除”按钮，完成工程 / 程序的删除。

### 4.1.3 恢复工程/模块/函数

点击可隐藏菜单栏的“恢复”按钮，可恢复最近一次删除的工程、模块或函数。

#### ⚠ 注意

1. 恢复操作需与删除操作配套使用；若备份文件夹不存在或无对应备份文件，点击恢复按钮后系统会弹出报警提示。
2. 若当前工程管理界面已存在同名的工程、模块或函数，无法执行恢复操作。
3. 恢复模块时，若原所属工程已不存在（如工程被重命名），则无法恢复。
4. 恢复函数时，若原所属模块和工程已不存在，或原模块处于写保护状态，则无法恢复。
5. 若用户主动删除了已删程序关联的全局变量或工程变量，恢复程序后编译器运行时可能会触发报错。

### 4.1.4 重命名工程/模块/函数

#### ⚠ 注意

1. 处于加载状态的工程或程序无法直接重命名，需先点击右侧可隐藏菜单栏的“卸载”按钮，解除加载状态后再执行操作。
2. 重命名程序时，其所属工程需先完成卸载，否则无法执行程序重命名操作。

1. 在工程管理界面中，选中待重命名的工程或程序（示例：project0）。
2. 点击右侧可隐藏菜单栏的“重命名”按钮。
3. 在弹出的软键盘中输入新的名称，点击“确认”按钮，完成重命名操作。



### 4.1.5 复制粘贴工程/程序

#### ⚠ 注意

同一个程序不允许跨工程目录复制粘贴；若将程序粘贴至其他工程目录下，系统会弹出提示信息：“不能将程序粘贴到不同的工程下！”，此时需重新选择目标程序所属的工程，再继续执行操作。

1. 在工程管理界面中，选中待复制的工程或程序（示例：project0）。
2. 点击右侧可隐藏菜单栏的“复制”按钮。
3. 点击右侧可隐藏菜单栏的“粘贴”按钮。
4. 在弹出的软键盘中输入新的工程 / 程序名称，点击“确认”按钮，完成复制粘贴操作。

### 4.1.6 自启动工程

#### ⚠ 注意

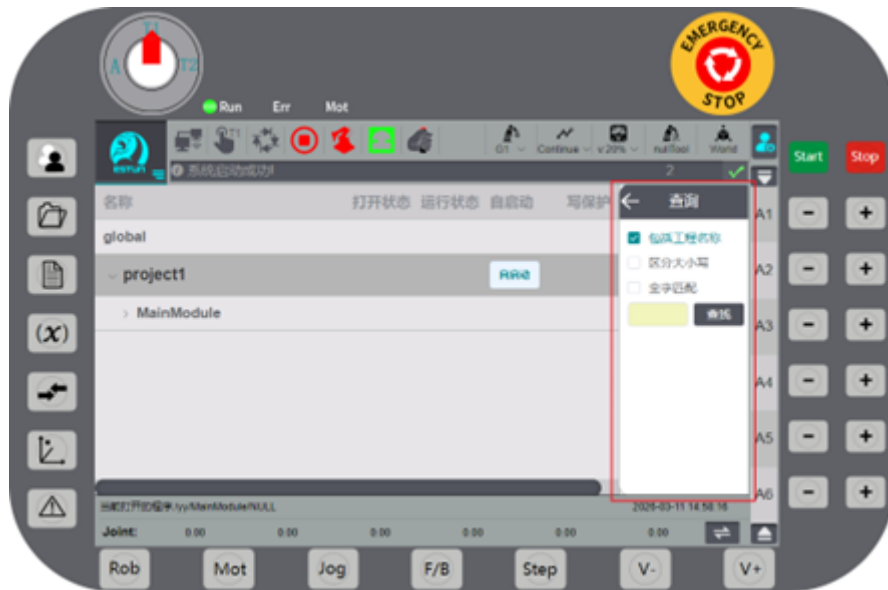
自启动功能仅支持对工程进行配置，不支持单独对程序设置自启动。当某工程被配置为自启动后，系统切换至远程模式时，该工程将被自动加载运行。

1. 在工程管理界面的工程列表中，选中需要设置自启动的目标工程。
2. 点击右侧可隐藏菜单栏的“自启动”按钮。
3. 观察到菜单栏中的“自启动”按钮切换为“取消自启动”，即表示该工程的自启动配置完成。



## 4.1.7 查找工程/程序

1. 点击右侧可隐藏菜单栏的“查询”按钮，弹出“查询”界面。



2. 在“名称”输入框中输入待查找的工程 / 程序名称文本，点击“查找”按钮执行检索操作。

### 说明

查询界面左侧提供 3 类筛选复选框，可按需勾选组合：

- 工程名称：默认选中，检索范围限定为工程名称。
- 区分大小写：勾选后，检索将区分字母大小写。
- 全字匹配：勾选后，仅匹配与输入文本完全一致的名称。

补充规则：支持输入名称中的部分连续字符进行模糊检索。

3. 查看检索结果，系统将按以下 3 种情况反馈：
  - 未找到：无匹配的工程 / 程序。
  - 找到一处：定位并高亮显示匹配结果。

- 找到多处：可点击“上一个”、“下一个”按钮切换结果，工程列表同步定位至对应条目（界面效果如图所示）。



4. 点击界面左上角“后退键”，退出查询功能，返回工程管理主界面。

## 4.1.8 工程导入导出

工程导入、导出操作支持导出至本地或可读写的U盘。工程导入、导出操作请参见[备份恢复](#)。

1. 工程导入：将本地或 U 盘中的工程或程序文件传输至控制器中。
2. 工程导出：将控制器中的工程或程序文件传输至本地或 U 盘中。

## 4.1.9 管理程序信息

通过“程序信息”功能可快速查看当前选中程序模块的基本属性信息，也可对程序添加自定义注释（如程序用途、修改记录等），便于工程管理与后续维护。

1. 在工程管理界面中，选中需要查看/编辑信息的目标程序。
2. 点击右侧可隐藏菜单栏的“程序信息”按钮，系统将弹出程序信息页签。



### 3. 在程序信息页签中：

- 可直接查看程序的基本信息（如创建时间、修改时间、所属工程等）。
- 可在注释输入区域填写自定义备注内容，完成后保存即可。

## 4.2 程序编辑

### 前提条件

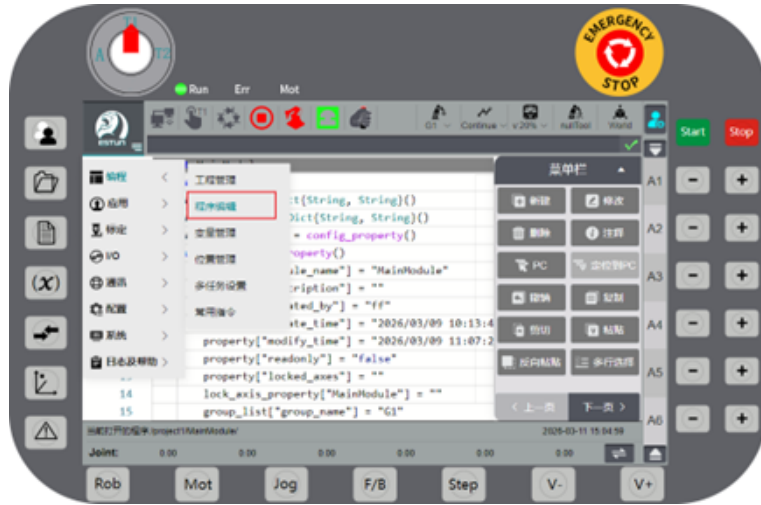
1. 执行任何指令编辑操作前，必须先完成目标程序所属工程的加载，未加载工程的程序无法进行编辑。
2. 若当前系统无已加载的工程/程序，需参照[程序编辑](#)章节的操作指引，先加载待编辑程序所属的工程。

### 进入程序编辑界面

系统提供两种进入程序编辑界面的方式，操作人员可根据实际编辑需求选择，具体操作如下：

**方式一：通过功能键/左侧菜单直接进入**

1. 操作示教器，按下面板上的“程序编辑”功能键；或在示教器左侧菜单栏中，点击“程序编辑”选项。
2. 系统跳转至程序编辑界面（界面布局如图所示）。



#### 方式二：通过工程管理界面进入

1. 在示教器界面，参照[工程管理](#)章节的操作指引，进入工程管理界面。
2. 在工程列表中，选中目标程序（需确保该程序所属工程已处于加载状态；支持选中该工程下尚未单独加载的程序）。
3. 点击工程管理界面右侧可隐藏菜单栏中的“打开”按钮。
4. 系统直接跳转至该目标程序的编辑界面。

## 4.2.1 新增指令

### 前提条件

新增指令前需确保已进入目标程序的编辑界面，进入方式如下：

1. 在示教器界面，参照[工程管理](#)章节操作指引，进入工程管理界面。
2. 在工程列表中选中目标程序（需确保该程序所属工程已加载，支持选中工程下未单独加载的程序）。
3. 点击工程管理界面右侧可隐藏菜单栏的“打开”按钮，系统跳转至该程序的编辑界面。

## 核心规则

1. 新建的指令将插入到光标所在行的上一行，因此新增指令前需将光标移动至目标位置。
2. 指令新建成功后，程序将自动加载并保持在程序编辑界面。
3. 部分指令无参数编辑项，可直接确认创建。

## 操作步骤

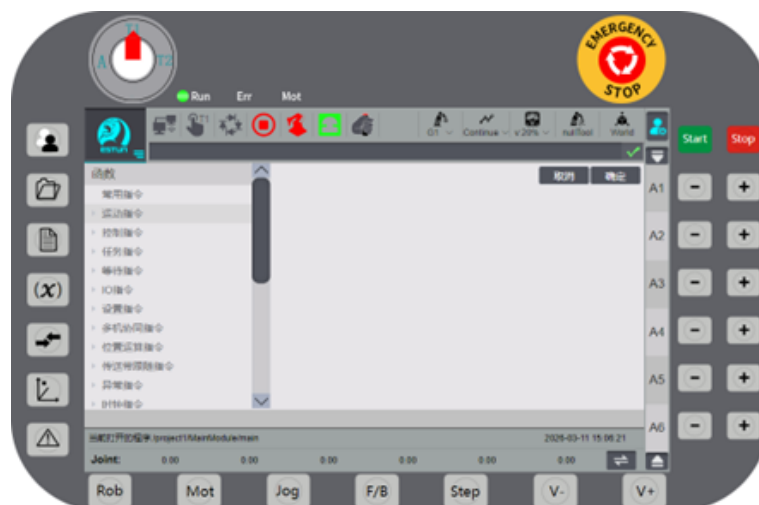
1. 进入程序编辑界面。
  - 若为新程序插入首个指令，需将光标移至“end”所在行（倒数第二行）。
  - 若为已有程序插入指令，将光标移至需插入指令的目标行。

```
16 function user_main()  
17 end  
18 end
```

### 说明

- 上图中底色为灰，前方有标志，表示程序将（继续）从该行开始执行。
- 底色为蓝，表示光标所在行。

2. 点击程序编辑界面右侧可隐藏菜单栏的“新建”按钮，进入指令选择界面。



3. 选择目标指令：

- a. 在“函数”分类下找到所需指令类型，如运动指令。
  - b. 展开子菜单选择具体指令，如选择：运动指令→MovJ。
  - c. 选择完成后，界面下方将弹出该指令的模板。
4. 点击“确认”按钮，进入“新建指令”参数编辑对话框。  
以 Movj 指令为例，对话框布局如图所示。



5. 按实际需求编辑指令参数。
- 数值输入：点击参数输入框，在弹出的软键盘中输入目标数值。
  - 下拉选择：点击输入框后的下拉箭头，从列表中选择预设参数值。
  - 位置参数：默认以机器人当前位置作为参数值，可按需修改。
6. 参数编辑完成后点击“确定”按钮，系统返回程序编辑界面，指令成功插入至光标指定位置。

```
function user_main()  
    var p0 :: Apos = Apos(0.000000,0.000000,0.000000,0.000000  
    MovJ(p0,v=S.V50,b="RELATIVE")  
end  
end
```

图 4-1 Movj 指令插入效果图

7. 重复步骤 2-6，可继续添加其他指令。
8. 所有指令添加完成后，检查程序编辑内容无误，即可返回主界面执行其他操作。

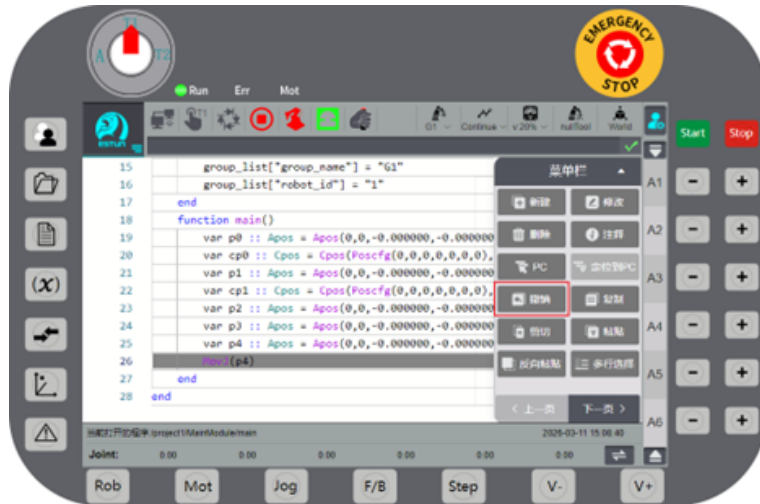
## 4.2.2 撤销操作

在程序编辑过程中，若执行了错误的指令新建或修改操作，可通过撤销功能取消当前操作，返回至上一编辑状态。

### ⚠ 注意

1. 撤销功能最多支持回溯最近 64 步的指令编辑操作。
2. 若对程序执行注销、重新加载或关闭后再次打开操作，此前的编辑操作记录将被清除，无法进行撤销。

1. 在程序编辑界面中，确认当前需要撤销的操作（指令新建或修改操作）。
2. 点击程序编辑界面右侧可隐藏菜单栏的“撤销”按钮（界面布局如图所示）。

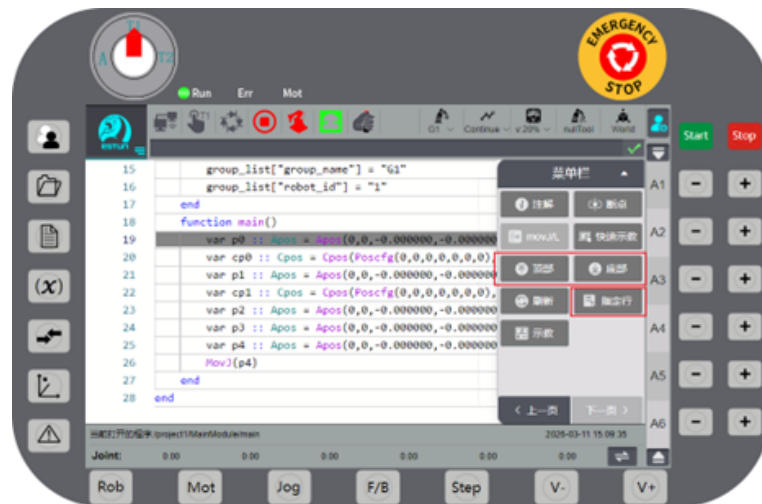


3. 系统自动执行撤销操作，恢复至上一正确的程序编辑状态。

## 4.2.3 快捷定位程序行位置

程序编辑界面提供行位置快捷定位功能，可快速将程序编辑光标/显示区域跳转至指定位置。

该功能入口位于程序编辑界面右侧可隐藏菜单栏，包含“顶部”、“底部”、“指定行”三个选项。



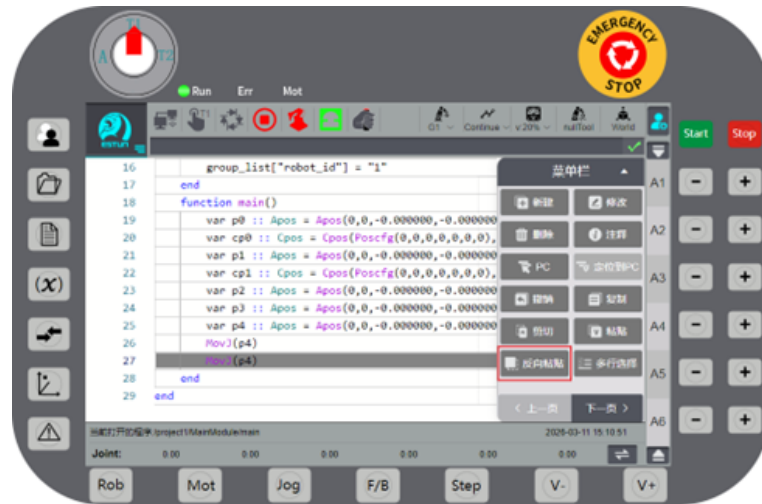
- 定位到顶部
  1. 在程序编辑界面点击右侧可隐藏菜单栏的“顶部”按钮。
  2. 序行显示区域直接定位至“export”行。
- 定位到底部
  1. 在程序编辑界面点击右侧可隐藏菜单栏的“底部”按钮。
  2. 程序行显示区域直接定位至“end”行。
- 定位到指定行
  1. 点击右侧可隐藏菜单栏的“指定行”按钮。
  2. 在弹出的上浮选项中，选择“标签”或“行号”作为定位依据。
  3. 输入 / 选择目标标签 / 行号。
  4. 程序行显示区域定位至所选“标签”或“行号”对应的位置。

## 4.2.4 反向粘贴

### 功能简介

反向粘贴是程序编辑界面的进阶编辑功能，可将已复制的一条/多条指令，按原顺序反向粘贴至指定位置。

该功能支持同一程序内粘贴，也支持同一工程下跨程序粘贴。



## 核心规则

- 操作范围限制：
  - 跨程序：仅支持同一工程下的不同程序间操作；
  - 禁止跨工程：指令的复制、剪切、粘贴、反向粘贴均不支持跨工程执行。
- 操作一致性：跨程序执行复制、剪切、粘贴、反向粘贴的操作步骤，与同一程序内操作完全一致。
- 变量兼容性规则：
  - 程序变量：粘贴的指令若包含程序变量，需满足以下条件否则执行报错：
    1. 当前加载程序有同名程序变量：指令可正常执行，变量值以当前程序定义为准；
    2. 当前加载程序无同名程序变量：需先新建对应变量并赋值，修改后指令可正常执行；
  - 全局/工程变量：不受跨程序粘贴影响，可正常调用。

## 操作步骤

1. 复制目标指令
  - a. 在程序编辑界面选中需要复制的一条/多条指令。
  - b. 点击右侧可隐藏菜单栏的“复制”按钮，完成指令复制。
2. 定位粘贴位置
  - 若为同一程序内粘贴：将光标移至需要粘贴指令的目标行。
  - 若为跨程序粘贴：切换至同一工程下的目标程序编辑界面，将光标移至目标行。
3. 执行反向粘贴

- a. 点击右侧可隐藏菜单栏的“反向粘贴”按钮。
- b. 系统将复制的指令按原顺序反向插入至光标指定位置，完成粘贴。

## 4.3 程序数据

### 4.3.1 新增/删除变量

#### 功能简介

变量管理界面支持创建全局/工程/程序级别的变量，也可删除无需使用的变量。


变量创建时需遵循系统命名规则，避免与保留名称冲突。

#### 核心规则

1. 命名限制：变量名称不可与系统保留名称重复（保留名称包含基础函数、指令、系统变量名称及特定关键字）；
2. 变量范围：支持创建全局变量、工程变量、程序变量三类，需在创建时明确所属“域”（范围）；
3. 操作前提：变量创建 / 删除后，需确保关联程序加载状态正常，避免变量调用异常。

#### 操作步骤

1. 进入变量管理界面。以下两种方式均可以进入“变量管理”界面。

- 按下  功能键。
- 点击示教器左侧菜单栏的“变量管理”选项。



## 2. 发起新建操作。

点击界面中的“新建”按钮，弹出“新建变量”对话框。



## 3. 配置变量基础属性：

- 选择变量所属“域”（范围）。
- 选择变量的分类和类型，分类 / 类型说明请参见表 4-1 变量的分类/类型说明表。
- 点击“名称”输入框，在弹出的软键盘中输入自定义变量名。需规避系统保留名称，系统保留名称请参见系统保留名称。

## 4. 确认创建。

检查变量参数设置无误后，点击“确认”按钮。系统返回变量列表界面，新增变量将显示在对应范围的列表中。

表 4-1 变量的分类/类型说明表

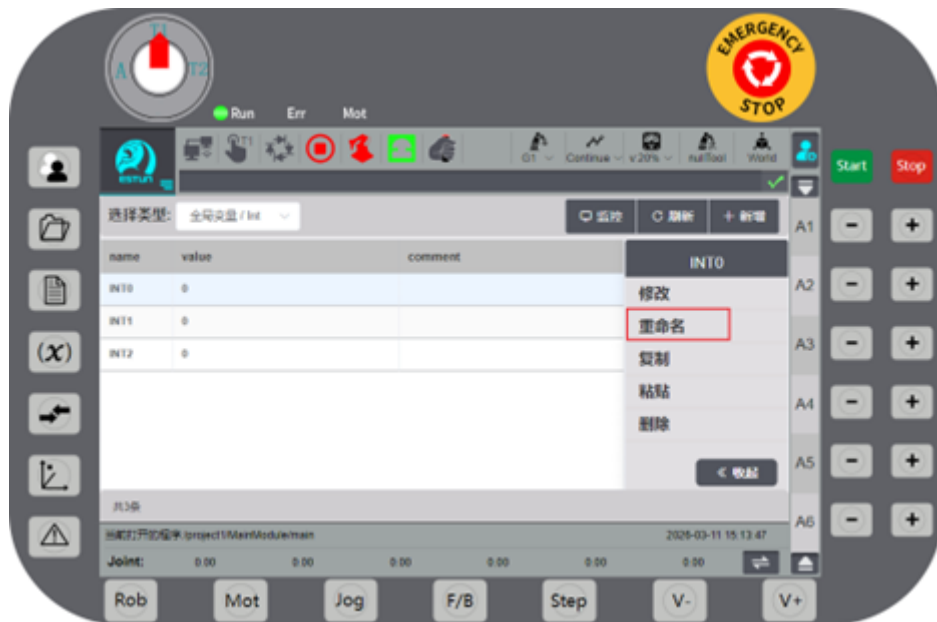
分类	类型	默认名称前缀
Socket数据类型	Socket服务器	TcpServer
	Socket客户端	TcpSocket
	UDP Socket	UdpSocket
位置数据类型	关节位置类型	P
	关节位置数组	APOSARRAY
	关节相对位置类型	DAPOS
	坐标系位置类型	cp
	坐标系位置数组	CPOSARRAY
	坐标系相对位置类型	DCPOS
基本数据类型	一维实数数组	RealOneArray
	一维布尔数组	BoolOneArray
	一维整型数组	IntOneArray
	二维实数数组	FloatTwoArray
	二维布尔数组	BoolTwoArray
	二维整型数组	IntTwoArray
	三维实数数组	FloatThreeArray
	三维布尔数组	BoolThreeArray
	三维整型数组	IntThreeArray
	字符串变量	STRING
	实数变量	REAL
	布尔变量	BOOL
	整型变量	INT
	摆动数据类型	摆弧变量
时钟数据类型	时钟类型	CLOCK
码垛数据变量	码垛变量	PALLET
系统数据类型	速度变量	SPEED
	转弯区变量	ZONE
	伺服低速增益变量	LsScale
	伺服低速阈值变量	LsThresh
	碰撞变量	ColliParam
	振动抑制自学习变量	VibrationParam
掉电保持数据类型	布尔变量	BoolRetain
	整型变量	IntRetain
	实数变量	FloatRetain
	字符串变量	StringRetain

表 4-1 变量的分类/类型说明表 (续)

分类	类型	默认名称前缀
中断数据类型	中断标识变量	INTERRUPT
异常数据类型	错误异常变量	ExceptionInfo
返回数据类型	返回变量	ReturnValue

### 4.3.2 重命名变量

1. 选择需要重新命名的变量，点击右侧菜单栏的“重命名”，将弹出“修改变量名称”的界面。






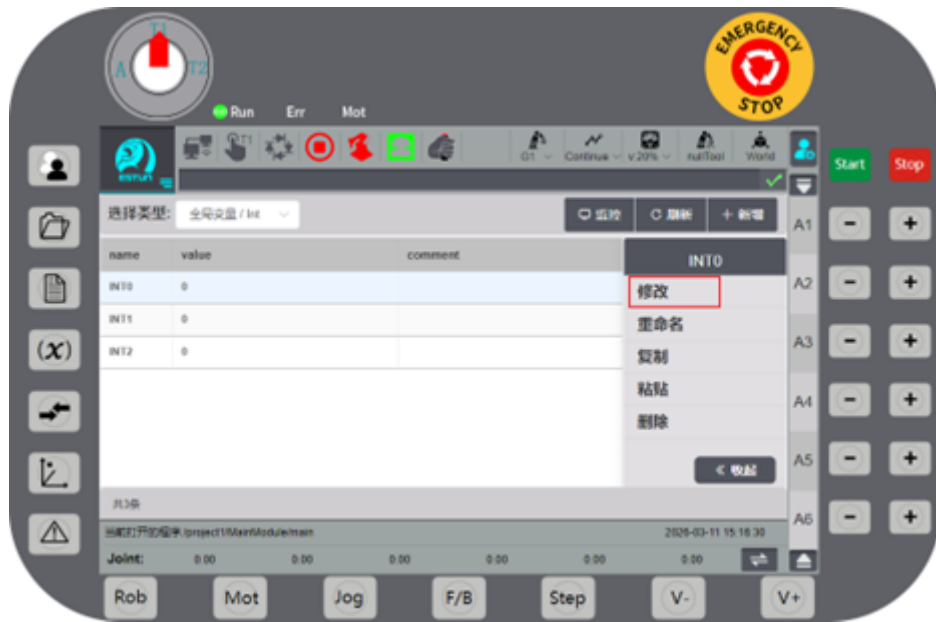
2. 点击输入框，并在弹出的软键盘中键入修改后的名称。
3. 点击“确认”，该变量将被重命名。

### 4.3.3 查看变量别名及注释

数组变量可以查看其别名，其他类型变量可查看其注释。

1. 点击  或如下图所示进入变量管理界面。





2. 单击“修改”，在“修改变量”弹窗中可查看其变量注释（comment）。

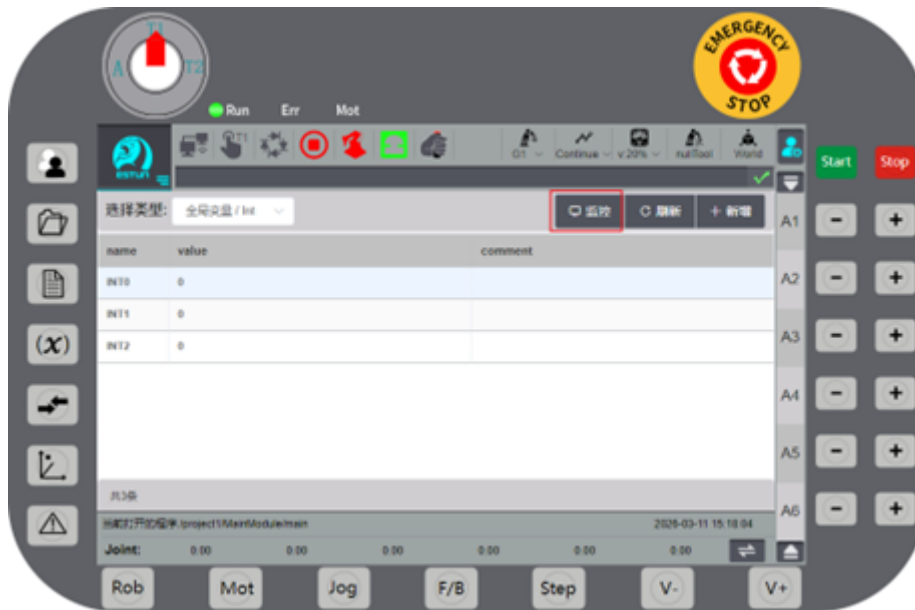


3. 数组别名查看方式相同，本手册以int数组举例。

#### 4.3.4 变量监控

示教器端可在线调试和监控选定的变量值，用户选择需要查看的变量数据添加到变量监视器内，在变量监视器页面即可观察选择的变量。

1. 点击右侧监控按钮，进入变量监控界面。



2. 点击页面右上角“筛选变量”按钮，可按照“类型”的方式筛选变量，将选择的变量添加至监控列表中。



3. 筛选变量后，点击“返回”，可在监控器列表页对所筛选的变量进行查看。



## 4.4 多任务设置

1. 点击“主菜单 -> 编程 -> 多任务设置”进入多任务设置页面。



### 4.4.1 新建任务

1. 点击“新建”，弹出虚拟键盘弹窗，通过键盘进行输入，输入的任务名称符合要求，name栏为蓝色，点击确认即可创建成功。如果不符合要求，name栏会将错误提示其上，并将不符合哪规则提示出来，显示为红色。

#### 📖 说明

任务名称名称规则：

- 不能以数字开头
- 不能全是下划线
- 长度不超过30



## 2. 默认创建的task

- 如果当前具有加载的工程，则默认创建的task工程为当前加载的工程，类型为并行。
- 如果没有加载工程，则默认创建的task工程为global工程，类型为自动。

3. 多任务页面的仅显示工程为global和已加载工程的task。其他工程的task不展示，如果没有加载的工程，仅显示工程为global的task。



#### 4.4.2 编辑任务

1. 选择任务点击“编辑”，进入任务修改界面。



2. 在此页面可以对任务的状态，任务绑定的工程、模块、函数，类型等参数。

- 类型分为自动、普通、并行三种。选择自动类型，工程将自动切换为global,且工程不能切换，自行为自动时，工程只能选择global、普通和并行状态，仅能选择当前加载的工程，如果没有加载工程，则两个状态均为灰色，不可选择，若含有加载工程，则可以选择并行。普通状态只用当机器人个数大于1，并且含有加载工程时才能选择。



3. task激活可通过点击多任务设置页面，或修改任务页面中的激活开关进行开启或关闭。  
(运行过程中，不允许开启或关闭任务)
4. 参数填写完毕后，点击“确定”，修改成功。

### 4.4.3 切换任务

可通过图中红色框选的位置进行切换，点击下面的选项即可修改。



## 4.5 平移功能

### 功能简介

程序平移功能可以将机器人程序中的点位信息进行整体平移，缩短用户重复示教点位的时间，提升用户的操作体验。

#### 说明

- 不支持程序平移空程序。
- 平移工程时，平移后工程中的module、函数命名不变，位置点全局变量和工程变量加上后缀“\_P”（此后缀P为用户指定），函数域位置点命名不变。
- 平移模块时，在原始模块的同工程下生成平移后新模块。平移后模块中的函数命名不变，位置点全局变量和工程变量加上后缀“\_P1”（此后缀“P1”为用户指定），函数域位置点命名不变。
- 所有平移后的位置点的值（包括全局域、工程域、函数域）为平移计算后的值。
- 执行程序转换成功后，自动跳转工程管理页面，列表刷新展示转换结果。

**⚠ 注意**

- 所选择的平移对象需与当前打开的程序一致（见左下角栏），否则不支持进入平移功能界面。
- 写保护模块下的函数不支持平移。
- 示教方式点位配置过程中，不允许切换是否旋转。
- 程序平移时对应的工程、module、函数命名需符合要求，位置变量和命名需符合要求，位置点需运行程序验证。
- 程序运行过程中，不允许执行平移相关操作。

**操作步骤**

程序平移主要包含需要平移程序段中的指令和点位转换。支持用户平移指定工程、指定模块及指定函数。

**1. 选择平移对象。**

打开工程/模块/函数，进入“工程管理”界面。在工程列表中，选中需要执行平移操作的对象（工程/模块/函数），在右侧菜单栏中点击“平移”。

**📖 说明**

所选平移对象需与当前打开的程序一致，否则不支持进入平移功能界面。

**2. 设置平移参数。**

使用平移功能，需首先对平移参数进行配置。根据平移对象不同，分为工程平移、模块平移和函数平移。

- 工程平移。

对于工程，仅支持整体平移。用户需要输入平移后的新工程名（目标工程名）、变量后缀名及选择转换方式。



- 模块平移。

对于模块，仅支持整体平移，且仅支持平移至当前工程下。即目标工程为该模块所属工程。用户需输入平移后的新模块名称（目标模块名）、变量后缀名以及选择转换方式。



- 函数平移。

对于函数，支持整体平移和部分平移，仅支持平移至当前模块，支持新建函数或插入指定函数。即目标工程为该函数所属工程，目标模块为该函数所属模块。用户需输入平移后的目标函数（函数名/插入行）、变量后缀名以及转换方式。



### 3. 点位转换。

平移参数配置完毕后，根据用户所选转换方式，进行相应的点位转换。

- 指定方式转换。

选择指定方式时，用户需在输入框中设置对应的X、Y、Z偏移量。点击“执行”按钮，二次确认后完成程序平移转换。



- 示教方式转换。

选择示教方式时，用户需首先选择是否进行旋转。即除进行位置点的X、Y、Z轴的偏移之外，是否包含位置点的姿态平移。

- 旋转：用户需依次设置三组源点和目标点（P1-Q1）。其中，源位置P1~P3支持选择变量，目标位置Q1~Q3支持选择变量或者示教。



- 非旋转：用户需设置一组源点和目标点（P1-Q1）。其中，源位置P1支持选择变量，目标位置Q1支持选择变量或者示教。源点和目标点设置成功后，点击“**执行**”按钮，二次确认后完成程序转换。



4. 点击“**上一步**”、“**下一步**”，依次配置P1-Q1，P2-Q2，P3-Q3。  
点位配置过程中，不允许切换是否旋转下拉框。三组点位均配置成功后，点击“**执行**”，二次确认后完成程序转换。

## 5 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：socket 数据类型、位置数据类型、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、异常数据类型、返回数据类型、掉电保持数据类型、中端数据类型。
- 工程域：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、异常数据类型、返回数据类型。
- 程序域：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、异常数据类型、返回数据类型。

### 5.1 基本数据类型

#### 5.1.1 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	掉点保存标记

#### 5.1.2 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

#### 5.1.3 FLOAT

存储实数。该变量支持在全局，工程，函数三大变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
value	Float	变量值

参数	数据类型	参数含义
comment	string	变量注释

### 5.1.4 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

### 5.1.5 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。

参数说明：

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

### 5.1.6 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

### 5.1.7 FloatOneArray

存储实数数组数据。该变量支持在全局，工程，函数三大变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

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

### 5.1.8 BoolTwoArray

存储Bool型二维数组数据。该变量只支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
row	int	行数
column	Int	列数
value	bool	变量值

### 5.1.9 IntTwoArray

存储整数二维数组数据。该变量只支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
row	int	行数
column	int	列数
value	int	变量值

### 5.1.10 FloatTwoArray

存储实数二维数组数据。该变量只支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
row	int	行数
column	int	列数
value	float	变量值

### 5.1.11 BoolThreeArray

存储Bool型三维数组数据。该变量只支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
row	int	行数
column	int	列数
depth	int	深度
value	bool	变量值

### 5.1.12 IntThreeArray

存储整数三维数组数据。该变量只支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
row	int	行数
column	int	列数
depth	int	深度
value	int	变量值

### 5.1.13 FloatThreeArray

存储实数三维数组数据。该变量只支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数	数据类型	参数含义
row	int	行数
column	int	列数
depth	int	深度
value	float	变量值

#### 说明

- 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。
- 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

### 5.2.1 APOS-关节位置类型

存储关节空间下各个轴的坐标值。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。

### 说明

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。

## 5.2.2 POSCFG

### 简介

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。

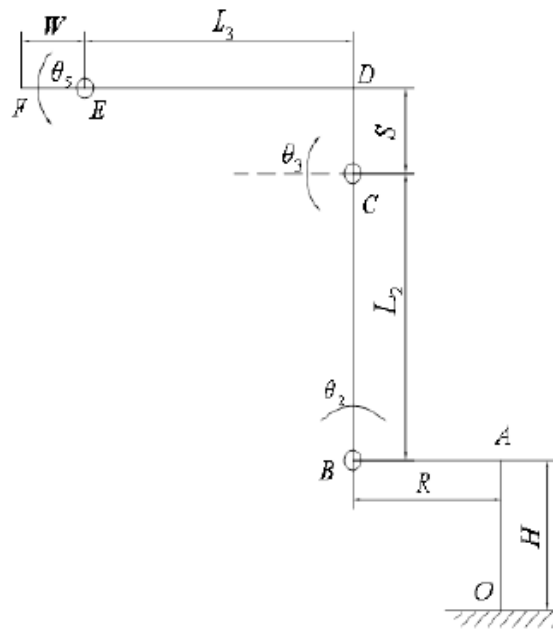
### 说明

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

### 通用六关节运动学mode定义

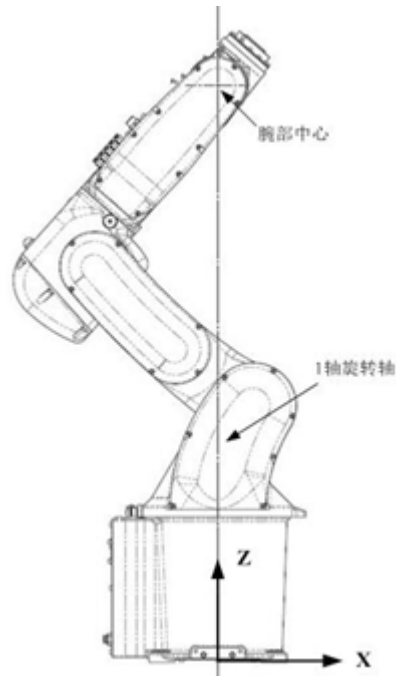
通用六关节的运动学存在八组解，定义 mode 值为 0~7，具体含义如下表所示。



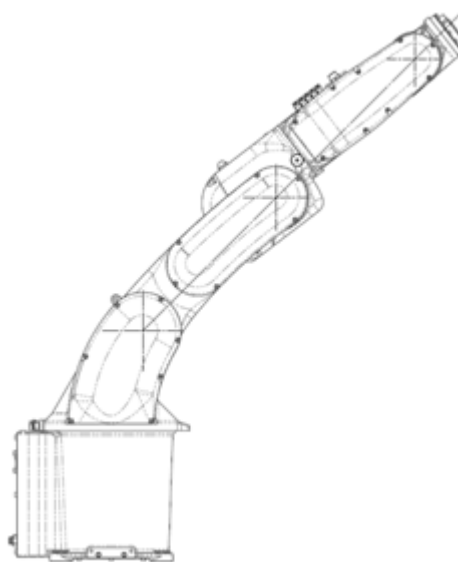
Mode	腕部中心相对于一轴轴心的关系(flag1) 0: 在前; 1: 在后	Axis3(flag3) ( $\theta_3 + 90 - \arctan(S/L_3)$ )	Axis5(flag5)( $\theta_5$ )
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

 说明

- $\text{mode} = \text{flag5} + 2 * \text{flag3} + 4 * \text{flag1}$
- 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。



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



- Axis5(flag5): 即五轴的角度。

## SCARA运动学mode定义

SCARA运动学存在两组解，定义mode值为0和1，具体含义如下表所示。

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

### 说明

mode = flag2

## 三轴平面机器人运动学+“3+1”机器人mode定义

Mode	Axis2(flag2) ( $\theta_2+90$ ) 0:[0,180] 1: (-180,0)
0	0
1	1

### 说明

mode=flag2

## 吊装SCARA运动学mode定义

吊装SCARA运动学存在两组解,定义mode值 0和1，具体含义如下表所示。

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

### 5.2.3 CPOS-坐标位置类型

存储TCP点在笛卡尔坐标系下位置。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。

#### 📖 说明

机器人有几个轴就会显示几个轴的参数。



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。
tool	int	所使用的工具坐标系
frame	int	所使用的用户坐标系或动坐标系
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。

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

#### 5.2.4 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。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	real	关节2号轴角度的偏移量。
~		~
da15		关节15号轴角度的偏移量。
da16	real	关节16号轴角度的偏移量。

##### 说明

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。

#### 5.2.5 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。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。

##### 说明

机器人有几个轴就会显示几个轴的参数。

参数	数据类型	参数含义
dx	real	空间目标点对应的形态配置信息。

参数	数据类型	参数含义
dy	real	TCP点在参考坐标系上x方向的坐标。
dz	real	TCP点在参考坐标系上y方向的坐标。
da	real	TCP点在参考坐标系上z方向的坐标。
db	real	TCP点相对于参考坐标系z轴旋转的欧拉角。
dc	real	TCP点相对于参考坐标系y'轴旋转的欧拉角。
da7	real	TCP点相对于参考坐标系x''轴旋转的欧拉角。
da8 ~ da15	real	关节8号轴的坐标。 ~ 关节15号轴的坐标。
da16	real	关节16号轴角度偏移量。

## 5.3 系统数据类型

### 5.3.1 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	J1轴对应的增益比例阈值，单位：%
J3	int	J1轴对应的增益比例阈值，单位：%
J4	int	J1轴对应的增益比例阈值，单位：%
J5	int	J1轴对应的增益比例阈值，单位：%
J6	int	J1轴对应的增益比例阈值，单位：%

### 5.3.2 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	J1轴对应的速度区间阈值，单位：r/min
J3	int	J1轴对应的速度区间阈值，单位：r/min
J4	int	J1轴对应的速度区间阈值，单位：r/min
J5	int	J1轴对应的速度区间阈值，单位：r/min
J6	int	J1轴对应的速度区间阈值，单位：r/min

### 5.3.3 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。

### 5.3.4 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。

### 5.3.5 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。

### 5.3.6 VibrationParam

VibrationParam型变量通过index来选择对应的振动抑制自学习参数（ID）。

参数	数据类型	参数含义
index	int	取值范围为[0, 6]。其中0表示默认振动抑制参数，其它表示ID为index的自学习振动抑制参数。

## 5.4 系统预留类型

### 5.4.1 AREA

存储标准区域信息的值，系统预定义变量。

参数	数据类型	参数含义
id	int	标准区域变量的编号。
is_default	int	是否为默认值。
act_state	int	是否激活此区域以及此区域是否在激活状态。
is_in_area	int	是否在区域内。
auto_act_state	int	区域自动激活标识。
type	int	区域类型。
shape	int	区域形状。
ref_coord_id	int	参考坐标系id。
input_enb	bool	使能输入。
inhigh_enb	int	输入高有效。
di_type	int	DI类型。
input_port	int	输入端口。
output_enb	bool	使能输出。
outhigh_enb	int	输出高有效。
do_type	int	DO类型。
output_port	int	输出端口。

参数	数据类型	参数含义
is_flange_end	int	监控端参数，在区域监控中用于监控TCP或法兰端。
box_X	float	矩形区域起点的坐标x。
box_Y	float	矩形区域起点的坐标y。
box_Z	float	矩形区域起点的坐标z。
box_lenX	float	矩形区域的长度。
box_lenY	float	矩形区域的宽度。
box_lenZ	float	矩形区域的高度。
cylinder_X	float	圆柱形区域起点的坐标x。
cylinder_Y	float	圆柱形区域起点的坐标y。
cylinder_Z	float	圆柱形区域起点的坐标z。
cylinder_R	float	圆柱形区域的半径。
cylinder_H	float	圆柱形区域的高度。
joint_id	int	区域形状为轴时，存储对应轴 id。
joint_max	float	区域形状为轴时，存储对应轴的最大限位信息。
joint_min	float	区域形状为轴时，存储对应轴的最小限位信息。
joint_group1_enb	bool	区域形状为轴组时，存储对应轴组的使能信息。
joint_group1_max	float	区域形状为轴组时，存储对应轴组的最大限位信息。
joint_group1_min	float	区域形状为轴组时，存储对应轴组的最小限位信息。
joint_group2_enb	bool	区域形状为轴组时，存储对应轴组的使能信息。
joint_group2_max	float	区域形状为轴组时，存储对应轴组的最大限位信息。
joint_group2_min	float	区域形状为轴组时，存储对应轴组的最小限位信息。
joint_group3_enb ~joint_group10_en- b, joint_group3_- max ~joint_- group10_max, join- t_group3_min~join- t_group10_min	float	区域形状为轴组时，存储对应轴组的最小限位信息。
ploy_vertex_num	int	多边体区域的顶点数量。
ploy_vertex_Z	float	多边体区域的顶点z值。
ploy_vertex_H	float	多边体区域的高度。
ploy_vertex_X1	float	多边体区域的顶点x值。
ploy_vertex_Y1	float	多边体区域的顶点y值。
ploy_vertex_X2	float	多边体区域的顶点x值。
ploy_vertex_Y2	float	多边体区域的顶点y值。
ploy_vertex_- X3~ ploy_vertex_- X10, ploy_vertex_- Y3~ ploy_vertex_Y10	float	多边体区域的顶点y值。

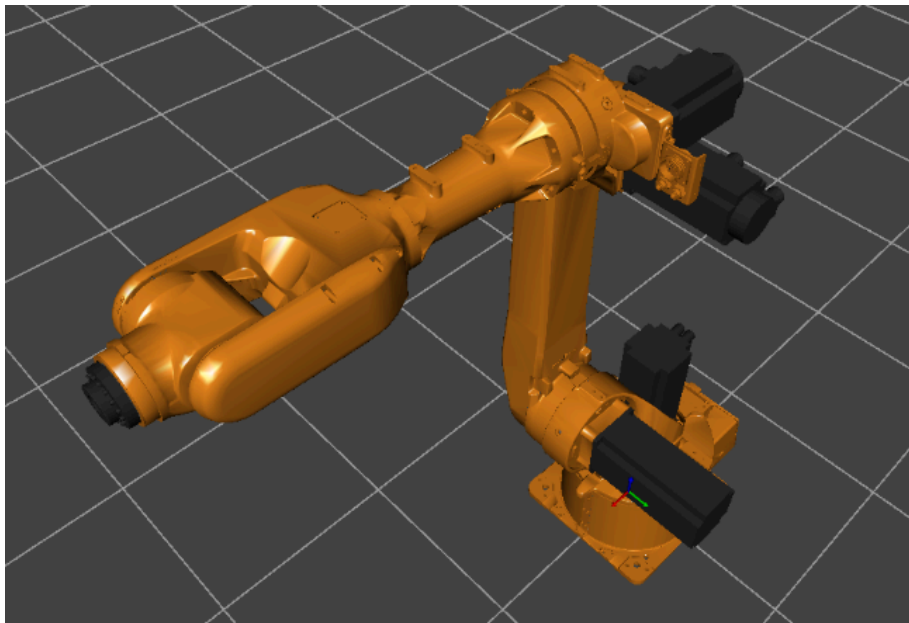
## 5.4.2 LoadDyn

用来存储机器人末端工具&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	float	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。

## 5.4.3 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。



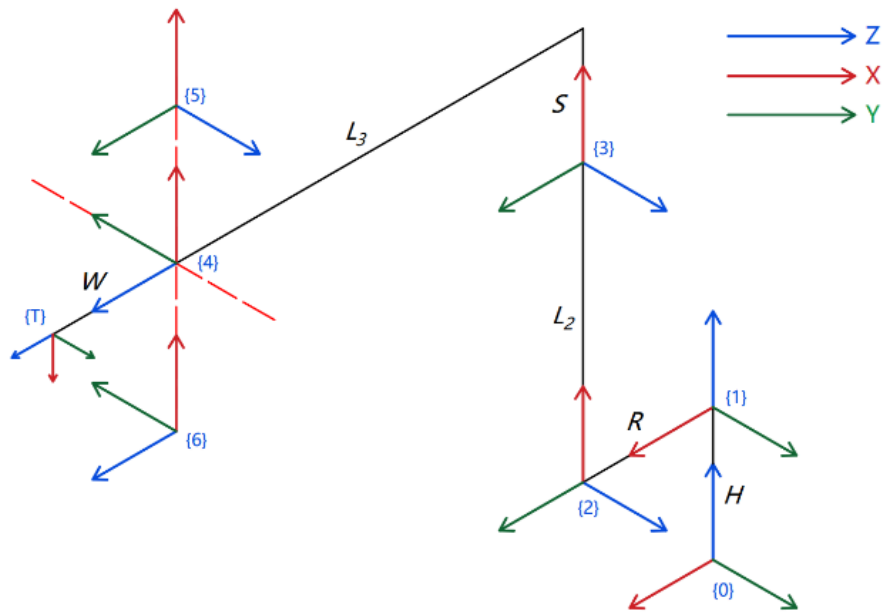


图 5-1 工具和负载质量信息参考坐标系

**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关节机器人为例）轴向同轴。

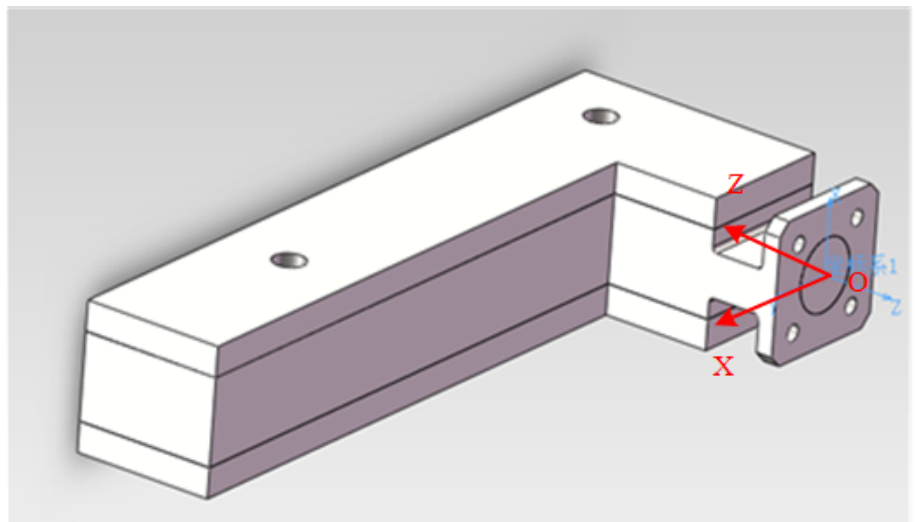


图 5-2 工具和负载质量信息坐标系定义实例

**说明**

工具和负载的质量信息坐标系均参考图1中坐标系 $O_{Tool}-XYZ$ 。

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

#### 5.4.4 FRAME

FRAME型变量用来记录坐标系参数，包括用户坐标系、动坐标系、外部TCP坐标系和变位机坐标系。其中用户坐标系用来记录用户坐标系参数，定义用户坐标系相对于世界（基）坐标系的位移和旋转；动坐标系用来记录动坐标系参数，定义动坐标系相对于世界（基）坐标系的位移和旋转；外部TCP坐标系用来记录外部工具坐标系参数，定义外部工具系相对于某个工具位置不动的位移和旋转；变位机坐标系用来记录变位机坐标系参数，定义变位机坐标系相对于世界（基）坐标系的位移和旋转。系统预定义变量。

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，由创建时系统自动分配。
x	float	TCP相对于世界（基）坐标系在x方向的位移偏移量，单位是mm。
y	float	TCP相对于世界（基）坐标系在y方向的位移偏移量，单位是mm。
z	float	TCP相对于世界（基）坐标系在z方向的位移偏移量，单位是mm。
a	float	TCP相对于世界（基）坐标系z轴旋转的欧拉角，单位是deg。
b	float	TCP相对于世界（基）坐标系y'轴旋转的欧拉角，单位是deg。
c	float	TCP相对于世界（基）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。

#### 5.4.5 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。该变量只存在系统域。

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。

参数	数据类型	参数含义
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。

### 5.4.6 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。

### 5.4.7 InertiaTensor

该参数是以安装的工具或负载的中心位置处所决定的惯性张量。具体的坐标系图可参考上述“CenterPos”变量类型中的说明图。

参数	数据类型	参数含义
lxx	float	安装的工具或装夹的负载在重心位置的X方向回传的惯量矩，单位是kg#mm <sup>2</sup> 。
lyy	float	安装的工具或装夹的负载在重心位置的Y方向回传的惯量矩，单位是kg#mm <sup>2</sup> 。
lzz	float	安装的工具或装夹的负载在重心位置的Z方向回传的惯量矩，单位是kg#mm <sup>2</sup> 。
lxy	float	安装的工具或装夹的负载在重心位置的XY交叉方向的惯量积，单位是kg#mm <sup>2</sup> 。

参数	数据类型	参数含义
lxz	float	安装的工具或装夹的负载在重心位置的XZ交叉方向的惯量积, 单位是kg#mm <sup>2</sup> 。
lyz	float	安装的工具或装夹的负载在重心位置的YZ交叉方向的惯量积, 单位是kg#mm <sup>2</sup> 。

### 5.4.8 \$\$SYS\_DATA

系统预定义了\$\$SYS\_DATA类型的系统变量, 属性表示其读写属性(RO/RW), 若系统变量属性为可写, 可在赋值语句中作为左值使用, 若只读属性可作为右值使用。

名称	属性	类型	描述
\$\$Global_Vel	RW	UINT16	全局速度
\$\$Mode_OP	RO	UINT16	当前操作模式手动-0、自动-1、远程-2、非法-3、T2-4
Servo_Status	RO	UINT16	励磁状态 励磁-1
\$\$Robot_Motion	RO	UINT16	机器人运动状态 运动-1
\$\$Prog_RunStatus	RO	UINT16	程序运行状态 0-程序初始化、1-程序运行、2-程序暂停、3-程序停止、4-程序错误
\$\$Curr_CoordType	RO	UINT16	当前坐标系类型 0:用户坐标系 1:外部TCP坐标系 2:变位机坐标系 3:动坐标系 4:主从坐标系
\$\$Curr_ToolCoordID	RO	UINT16	当前工具坐标系ID
\$\$Curr_UserCoordID	RO	UINT16	当前用户坐标系ID
\$\$Curr_LoadInfo	RO	UINT16	当前负载ID, 暂不支持为左值
\$\$Error_Info	RO	UINT16	错误ID
\$\$Robot_No	RO	UINT16	机器人编号
\$\$TCP_Speed	RO	DOUBLE	TCP速度
\$\$Estop	RO	UINT16	急停状态
\$\$Axis%d_Torque	RO	DOUBLE	轴扭矩
\$\$Axis%d_AlarmState	RO	UINT16	轴报警状态
\$\$Axis%d_AlarmState	RO	UINT16	轴报警状态

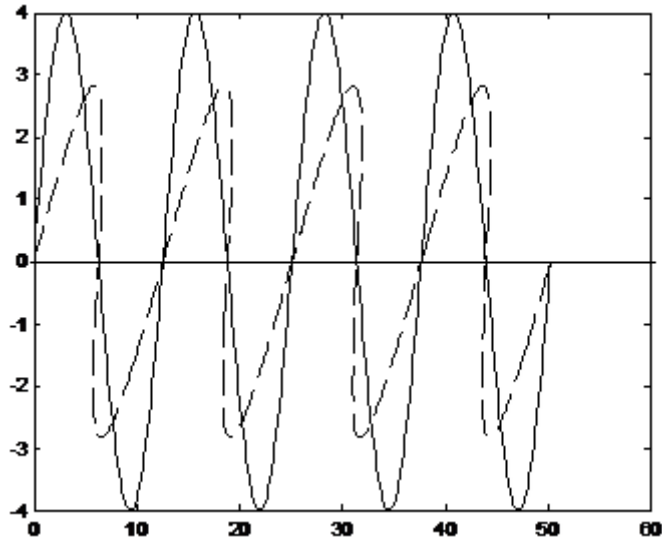
## 5.5 摆动数据类型

### 5.5.1 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。

 说明

- 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。



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。
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

## 5.6 时钟数据类型

## 5.6.1 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	bool	时钟变量的使能状态。
value	real	时钟变量的计数值。

## 5.7 码垛数据类型

### 5.7.1 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。
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。
palletEntryPos	PLACEOPTIONS	前置点信息。
postplaceOptions	PLACEOPTIONS	后置点信息

## 5.7.2 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。

## 5.8 SOCKET数据类型

### 5.8.1 TcpServer

存储TCP Socket服务器信息。该变量支持用户自主创建，但不允许修改，执行相关指令时会设置相应的参数值。该变量支持在全局、工程和程序变量作用域中创建，删除，复制，粘贴等操作。

数据	数据类型	参数含义
handle	int	TCP Socket句柄
status	int	TCP Socket状态
ip	string	IP地址
port	int	端口号

### 5.8.2 TcpSocket

存储TCP Socket客户端信息。该变量支持用户自主创建，但不允许修改，执行相关指令时会设置相应的参数值。该变量支持在全局、工程和程序变量作用域中创建，删除，复制，粘贴等操作。

数据	数据类型	参数含义
handle	int	TCP Socket句柄
status	int	TCP Socket状态
ip	string	IP地址
port	int	端口号

### 5.8.3 UdpSocket

存储UDP Socket信息。该变量支持用户自主创建，但不允许修改，执行相关指令时会设置相应的参数值。该变量支持在全局、工程和程序变量作用域中创建，删除，复制，粘贴等操作。

数据	数据类型	参数含义
handle	int	UDP Socket句柄
status	int	UDP Socket状态
ip	string	IP地址
port	int	端口号

## 5.9 掉电保持数据类型

### 5.9.1 BoolRetain

存储布尔类型掉电保持数据信息，该变量仅支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

数据	数据类型	参数含义
value	bool	掉电保持使能值
comment	string	变量注释

### 5.9.2 IntRetain

存储整型类型掉电保持数据信息，该变量仅支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

数据	数据类型	参数含义
value	int	掉电保持使能值
comment	string	变量注释

### 5.9.3 FloatRetain

存储实数类型掉电保持数据信息，该变量仅支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

数据	数据类型	参数含义
value	float	掉电保持使能值

数据	数据类型	参数含义
comment	string	变量注释

## 5.9.4 StringRetain

存储字符串类型掉电保持数据信息，该变量仅支持在全局变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

数据	数据类型	参数含义
value	string	掉电保持使能值
comment	string	变量注释

## 5.10 异常数据类型

### 5.10.1 ExceptionInfo

存储异常数据信息，该变量支持在全局，工程，函数三大变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

数据	数据类型	参数含义
id	int	异常数据信息索引id
comment	string	变量注释

## 5.11 返回数据类型

### 5.11.1 ReturnValue

存储返回数据，该变量支持在全局，工程，函数三大变量作用域中创建，删除，修改，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

数据	数据类型	参数含义
int_value	int	整型值
bool_value	bool	整型值
str_value	string	字符串值
float_value	float	实数值
comment	string	变量注释

## 5.12 视觉纠偏类型

### 5.12.1 VISIONRET

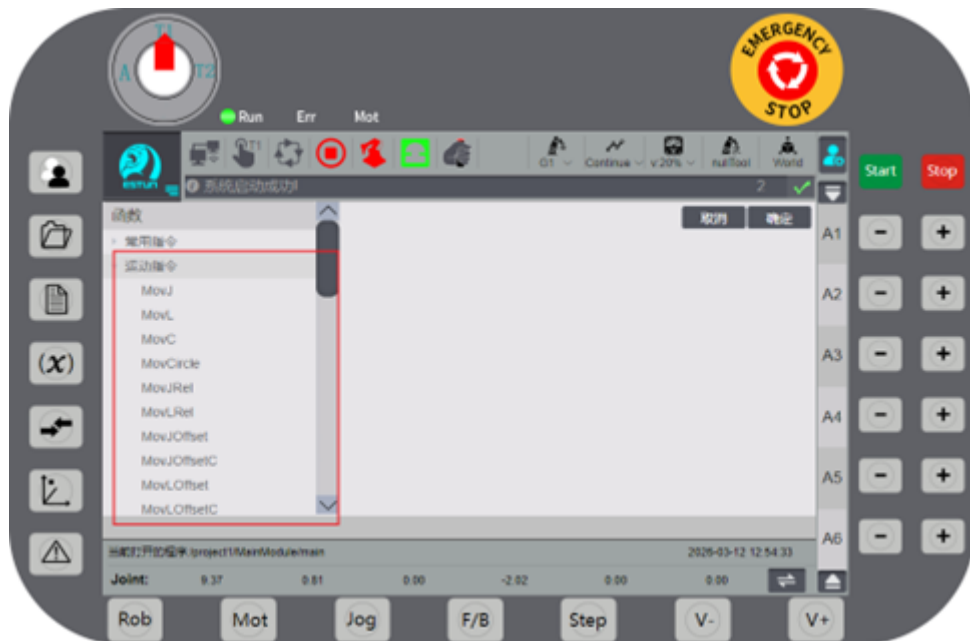
该变量用于存储相机发来纠偏的数据，需配合视觉指令组使用。

参数	数据类型	参数含义
OffsetPos	Cpos	存储视觉发送过来的点位信息，主要是x、y和a。
TagID	int	解析视觉发送过来的TagID的值。
IsValid	int	该标志位用来判断当前视觉数据是否有效。

## 6 指令介绍

### 6.1 运动指令

运动指令列表如下图所示。



#### 6.1.1 MovJ - 点到点的运动指令

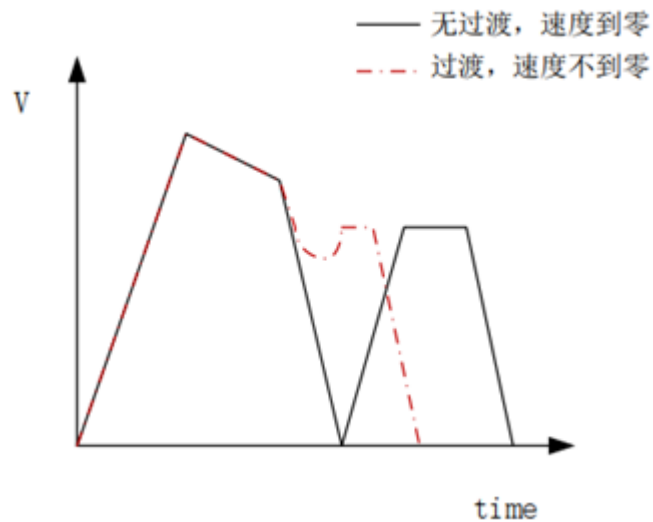
##### 指令简介

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。

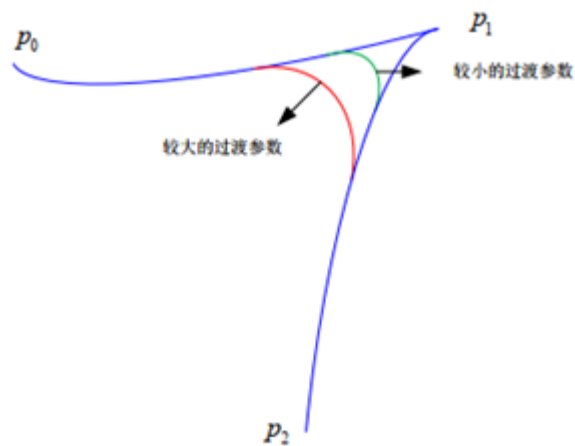


## 参数说明

- 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，则终点位置不受限于confdata的值，而是以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，需要选用系统预定义，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：精确到位。
  - 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。



- AddDo: 机器人执行完该指令后, 可进行 IO 操作。
  - NULL: 无任何操作。
  - IO 操作: 执行 IO 操作, 目前支持的 IO 指令有 Do、Ao、SimDo、SimAo、Sim-Go。

## 6.1.2 MovL - 直线运动指令

### 指令简介

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。



## 参数说明

- 目标位置：指令终点位置，类型为APOS或 CPOS。在弹出的对话框中选择已有的点或者新创建一个点并示教。
- 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，需要选用系统预定义，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：精确到位。
  - 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。
- 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，需要选用系统预定义，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。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- 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影响。
  - DEFAULT：同 GOVRON，即指令速度受示教器全局倍率影响。
  - GOVRON：指令速度受示教器全局倍率影响。
  - GOVROFF：指令速度不受示教器全局倍率影响。
- 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。

### 6.1.3 MovC - 弧形运动指令

#### 指令简介

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。



## 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。
  - 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。
- 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”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。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。

- OPEN: 开启跨奇异。
- CLOSE: 关闭跨奇异。
- DEFAULT: 缺省设置。
- 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影响。
  - DEFAULT: 同 GOVRON, 即指令速度受示教器全局倍率影响。
  - GOVRON: 指令速度受示教器全局倍率影响。
  - GOVROFF: 指令速度不受示教器全局倍率影响。
- 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。

### 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。

## 6.1.4 MovCircle - 圆形运动指令

### 指令简介

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。

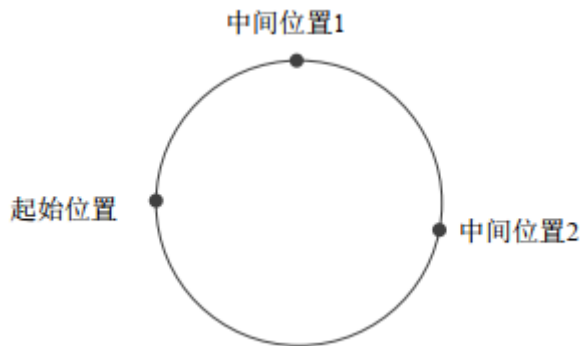


## 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。
  - 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。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
  - DEFAULT：表示沿用最近一次设置的工件负载参数。
  - PAYLOAD：需要选用系统预定义；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器全局倍率影响。

- DEFAULT: 同 GOVRON, 即指令速度受示教器全局倍率影响。
- GOVRON: 指令速度受示教器全局倍率影响。
- GOVROFF: 指令速度不受示教器全局倍率影响。
- 跨奇异: 本运动指令是否开启跨奇异设置, 该选项仅适用于通用六关节机器人。
  - OPEN: 开启跨奇异。
  - CLOSE: 关闭跨奇异。
  - DEFAULT: 缺省设置。
- 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。

### MovCircle整圆关键点示意



## 6.1.5 MovJSearch - 寻位指令 (MovJ)

### 指令简介

寻位指令是指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。



### 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。

### 参数说明

- 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。
- 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），过渡方式仍然按照 FINE 的方式运行。
  - FINE：精确到位。

### 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。

- 工件负载：工件负载信息。
- 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。
- TorqTrig: 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。
- 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。
- 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。
- AddDo: 机器人执行完该指令后, 可进行 IO 操作。

## 配置示例

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; 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 INTO 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 INTO 被设为 0。

```
MovJ(p1,v=S.V50,b="RELATIVE",c=S.C100)
MovJSearch(p2,S.V100,"DITrig",16,1.0,INT0 ,b="RELATIVE")
MovJ(p2,v=S.V50,b="RELATIVE",c=S.C100)
```

## 6.1.6 MovLSearch - 寻位指令 (MovL)

### 指令简介

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，中断返回值置 1。

若指令运行过程中无中断，中断返回值置 0。



## 参数说明

- 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），过渡方式仍然按照 FINE 的方式运行。
  - FINE：精确到位。

### 📖 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。

- 工件负载：工件负载信息。

- 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
- 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。
- 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。
- 跨奇异: 本运动指令是否开启跨奇异设置, 该选项仅适用于通用六关节机器人。
  - OPEN: 开启跨奇异。
  - CLOSE: 关闭跨奇异。
  - DEFAULT: 缺省设置。
- AddDo: 机器人执行完该指令后, 可进行 IO 操作。

## 配置示例

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 INTO 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 INTO 被设为 0。

```
MovL(p1,v=S.V50,b="RELATIVE",c=S.C100)  
MovLSearch(p2,S.V100,"DITrig",16,1.0,INT0 ,b="FINE")  
MovL(p2,v=S.V50,b="RELATIVE",c=S.C100)
```

## 6.1.7 MovJRel - 相对偏移指令

### 指令简介

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。



### 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。

- 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。
- 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, 需要选用系统预定义; 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。

## 6.1.8 MovLRel-插补相对偏移指令

### 指令简介

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。



## 参数说明

- 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。
  - Frame：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，需要选用系统预定义。
- 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：精确到位。
  - 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。
- 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，需要选用系统预定义；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器全局倍率影响。
  - DEFAULT：同 GOVRON，即指令速度受示教器全局倍率影响。
  - 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。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- AddDo：机器人执行完该指令后，进行 IO 操作。

## 6.1.9 MovJOffset - 关节空间偏移运动指令

### 指令简介

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。



### 参数说明

- 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，“Frame”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。
- 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。
- 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。

## 6.1.10 MovJOffsetC - 关节空间偏移运动指令

### 指令简介

与 MovJOffset 指令功能一致，但 MovJOffsetC 指令目标相对位置由 DCPOS 类型的变量更改为 X 增量、Y 增量、Z 增量、A 增量四个 real 型常量。



### 参数说明

- 参考位置：偏移运动的起点（APOS 或 CPOS）。
- X 增量：执行该指令时机器人在 X 方向上要移动的位置增量。
- Y 增量：执行该指令时机器人在 Y 方向上要移动的位置增量。
- Z 增量：执行该指令时机器人在 Z 方向上要移动的位置增量。
- A 增量：执行该指令时机器人在 A 方向上要移动的旋转增量。
- 参考坐标系：坐标系偏移或工具偏移选择。
  - “Frame” 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。
- 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。
- 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。

### 6.1.11 MovLOffset - 直线偏移运动指令

#### 指令简介

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, 其他功能均与 MovLRel 指令相同。



#### 参数说明

相对于 MovLRel 新增指令参数介绍如下, 其他指令请参见 [MovLRel 指令](#) 中参数介绍。

- 参考位置: 偏移运动的起点 (APos 或 CPos)。

## 6.1.12 MovLOffsetC - 直线偏移运动指令

### 指令简介

与 MovLOffset 指令功能一致，但 MovLOffsetC 指令目标相对位置由 DCPOS 类型的变量更改为 X 增量、Y 增量、Z 增量、A 增量四个 real 型常量。



### 参数说明

- 参考位置：偏移运动的起点（APOS 或 CPOS）；
- X 增量：执行该指令时机器人在 X 方向上要移动的位置增量；
- Y 增量：执行该指令时机器人在 Y 方向上要移动的位置增量；
- Z 增量：执行该指令时机器人在 Z 方向上要移动的位置增量；
- A 增量：执行该指令时机器人在 A 方向上要移动的旋转增量；
- 参考坐标系：坐标系偏移或工具偏移选择。
  - “Frame” 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；
- 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。
- 工件负载: 机器人执行该轨迹时使用的工件负载参数;
- 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影响。
  - DEFAULT: 同 GOVRON, 即指令速度受示教器全局倍率影响。
  - GOVRON: 指令速度受示教器全局倍率影响。
  - GOVROFF: 指令速度不受示教器全局倍率影响。
- 跨奇异: 本运动指令是否开启跨奇异设置, 该选项仅适用于通用六关节机器人。
  - OPEN: 开启跨奇异。
  - CLOSE: 关闭跨奇异。
- DEFAULT: 缺省设置。AddDo: 机器人执行完该指令后, 进行 IO 操作。

### 6.1.13 MovLW - 摆动指令

#### 指令简介

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。



---

 说明

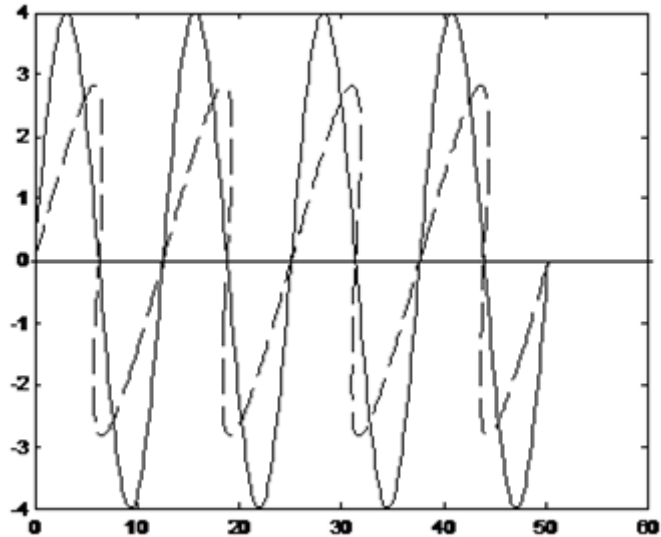
WEAVE 类型变量有如下参数：

- 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
-

### 说明

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。



### 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。
  - 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，需要选用系统预定义；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器全局倍率影响。
  - DEFAULT：同 GOVRON，即指令速度受示教器全局倍率影响。
  - GOVRON：指令速度受示教器全局倍率影响。
  - GOVROFF：指令速度不受示教器全局倍率影响。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- 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。

## 6.1.14 MovCW - 圆弧摆动指令

### 指令简介

圆弧摆动指令，指的是根据设定的摆频、摆幅，在圆弧前进方向上做摆弧运动。



## 参数说明

- 中间位置：辅助点位置（APOS 或 CPOS 类型位置点），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。  
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。
- 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。
- 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，需要选用系统预定义。
- 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：精确到位。
  - 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，需要选用系统预定义；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器全局倍率影响。
  - DEFAULT：同 GOVRON，即指令速度受示教器全局倍率影响。
  - GOVRON：指令速度受示教器全局倍率影响。
  - GOVROFF：指令速度不受示教器全局倍率影响。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- 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。

## 6.1.15 MovCircleW- 整圆摆动指令

### 指令简介

整圆摆动指令，通过该指令可以使机器人在整圆运动的过程中，保持持续的摆弧运动。目前支持的摆弧运动包括正弦摆和三角摆两种运动。



## 参数说明

- 中间位置1：整圆中间辅助点位置1（APOS 或 CPOS 类型位置点），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。  
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。
- 中间位置2：整圆中间辅助点位置1（APOS 或 CPOS 类型位置点），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。  
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。
- 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。
- 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，需要选用系统预定义。
- 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：精确到位。
  - 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，需要选用系统预定义；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器全局倍率影响。
  - DEFAULT：同 GOVRON，即指令速度受示教器全局倍率影响。
  - GOVRON：指令速度受示教器全局倍率影响。
  - GOVROFF：指令速度不受示教器全局倍率影响。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- 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。

## 6.1.16 MovJArch - 拱形运动指令

### 指令简介

拱形运动指令。整个运动过程为：首先机器人在当前点抬升  $h_1$  高度，其后再运动到距离目标点高度为  $h_2$  处，最后再下降  $h_2$  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。

---

#### 说明

该指令目前仅适用于 SCARA 机型。

---

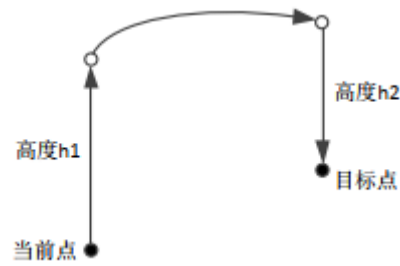


图 6-1 MovArc运动轨迹



## 参数说明

- 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：精确到位。
  - 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。
- 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，需要选用系统预定义；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。
- 高度类型：机器人执行轨迹时使用绝对高度（ABSOLUTE）或相对高度（RELATIVE）。
  - 高度类型为相对高度（RELATIVE）时。
    - 抬升高度：机器人末端执行关节拱形运动时的抬升高度，如图1中h1。
    - 下降高度：机器人末端执行关节拱形运动时的下降高度，如图1中h2。
  - 高度类型为绝对高度时。  
绝对高度（mm）：机器人末端执行关节拱形运动时抬升到的绝对高度，即设置的值为机器人实际运行时抬升的高度。
- 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。

## 配置示例

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。

```
MovL(p1,v=S.V50,b="RELATIVE",c=S.C50)  
MovJArch(p2,S.C100,v=S.V100,b="FINE",h1=10.0,h2=10.0)
```

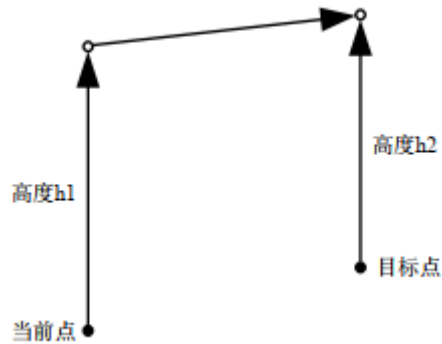
## 6.1.17 MovLArch - 直线拱形运动指令

### 指令简介

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。



## 参数说明

- 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。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
- 高度类型：机器人执行轨迹时使用绝对高度（ABSOLUTE）或相对高度（RELATIVE）。
  - 高度类型为相对高度（RELATIVE）时。
    - 抬升高度：机器人末端执行关节拱形运动时的抬升高度，如上图h1。
    - 下降高度：机器人末端执行关节拱形运动时的下降高度，如上图h2。
  - 高度类型为绝对高度时。
 

绝对高度（mm）：机器人末端执行关节拱形运动时抬升到的绝对高度，即设置的值为机器人实际运行时抬升的高度。

- 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。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- 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。

## 6.1.18 MovH - 螺旋运动指令

### 指令简介

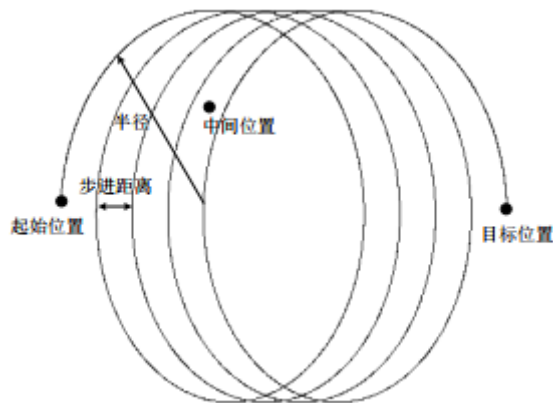
螺旋运动指令，通过该指令可以使机器人 TCP 点做螺旋运动到目标位置，若运动的起止姿态不同，则运行过程中姿态随位置同步地旋转到终点的姿态。



## 参数说明

- 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，需要选用系统预定义；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。
- 跨奇异：本运动指令是否开启跨奇异设置，该选项仅适用于通用六关节机器人。
  - OPEN：开启跨奇异。
  - CLOSE：关闭跨奇异。
  - DEFAULT：缺省设置。
- AddDo：机器人执行完该指令后，可进行 IO 操作。

## 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。

### 配置示例

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。

```
MovL(p1,v=S.V50,b="RELATIVE",c=S.C50)  
MovH(p2,p3,10.0,5.0,v=S.V100,b="FINE",payload=S.PAYLOAD[1])
```

## 6.1.19 OnDistance - 距离触发指令

### 指令简介

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。



## 参数说明

- 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。
  - 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未点判断时，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。
- 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。  
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。
- 触发的操作：满足触发条件后，机器人需要执行的触发操作。

---

## 说明

- 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。
- 触发指令默认为不支持单步模式，可在主界面按照“通用设置 > 程序设置”修改“单步使能”参数，开启/关闭触发指令的单步模式。
- OnDistance 只支持 MOVL,MOVC 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。

## 配置示例

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

```
MovL(p1,v=S.V50,b="RELATIVE",c=S.CO)
OnDistance("FromBegin",200.0,100) do S.DO[1]=1 end
MovL(p2,v=S.V50,b="RELATIVE",c=S.CO)
```

## 6.1.20 OnParameter - 距离百分比触发指令

### 指令简介

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。



### 参数说明

- 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。

- 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。

### • 说明

触发指令默认为不支持单步模式，可在主界面选择“通用设置 -> 程序设置”修改“单步使能”参数，开启/关闭触发指令的单步模式。

## 配置示例

```
MovJ(p1,v=S.V50,b="RELATIVE",c=S.C0)
OnParameter(60,500) do S.DO[1]=1 end
MovL(p2,v=S.V50,b="RELATIVE",c=S.C0)
```

## 6.1.21 MovJSyncQuit - 同步跟随退出指令

### 指令简介

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。

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。



## 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。
  - 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。
- 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，需要选用系统预定义；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。
- 退出点使能：执行完本次跟随轨迹，是否退出的使能状态。
  - DEFAULT：默认值，表示沿用最近一次设置的使能状态。
  - ENABLE：执行完本次跟随轨迹后退出跟随。
  - DISABLE：执行完本次跟随轨迹后不退出跟随，直接参与下一次的跟随运动。
- 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 PulseOut、WaitIO、WaitIO8421。

### 6.1.22 MovLSync - 启动传送带同步的直线运动指令

#### 指令简介

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。

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。



## 参数说明

- 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。
- 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 MovL.Sync 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。
- 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，需要选用系统预定义。
- 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。
- 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 PulseOut、WaitIO、WaitIO8421。

## 6.1.23 MovLSyncQuit - 同步跟随退出指令

### 指令简介

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。

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。



### 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。
  - 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。
- 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，需要选用系统预定义；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。
- 退出点使能：执行完本次跟随轨迹，是否退出的使能状态。
  - DEFAULT：默认值，表示沿用最近一次设置的使能状态。
  - ENABLE：执行完本次跟随轨迹后退出跟随。
  - DISABLE：执行完本次跟随轨迹后不退出跟随，直接参与下一次的跟随运动。
- 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PulseOut、WaitIO、WaitIO8421。

## 6.1.24 MovToIPos - 返回中断点指令

### 指令简介

该指令用于返回中断点，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。以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到该点，无需用户对该点进行额外的指令处理。

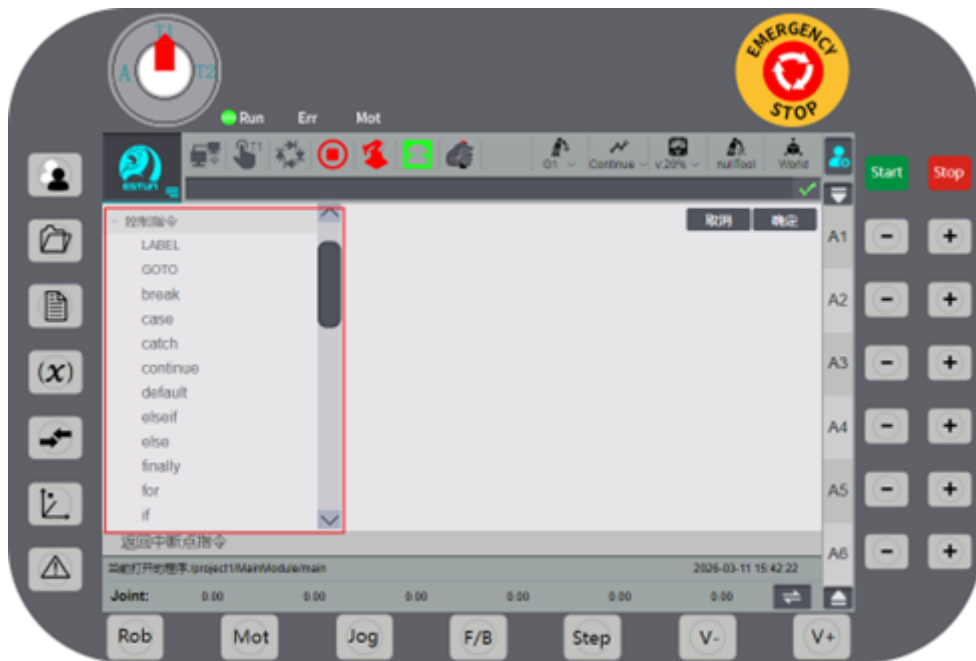


## 参数说明

- 运动方式：指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。

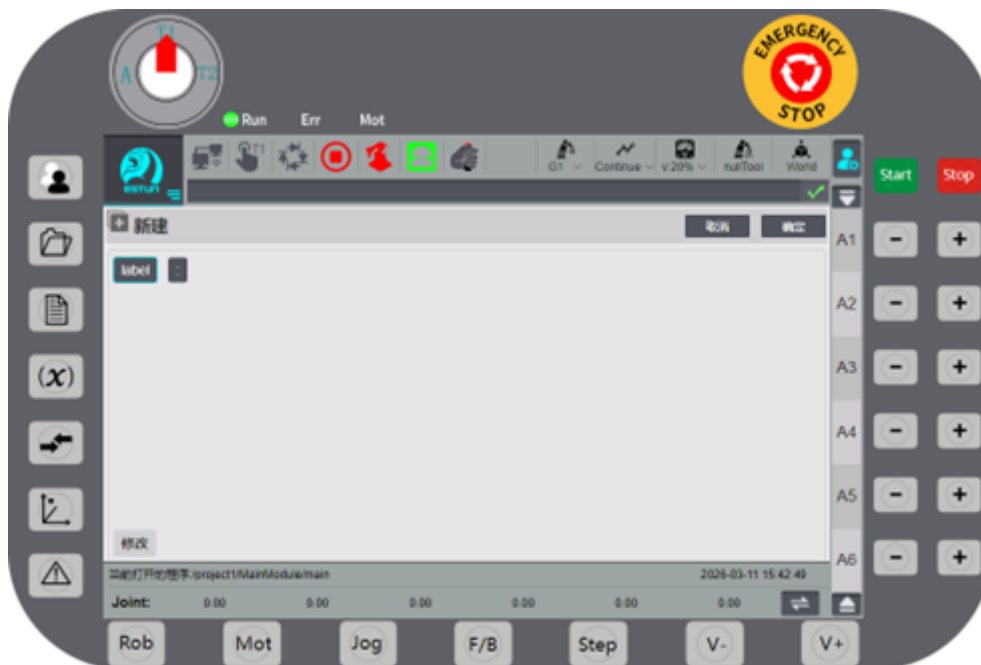
## 6.2 控制指令

控制指令列表如下图所示。



## 6.2.1 LABEL - 定义跳转目标

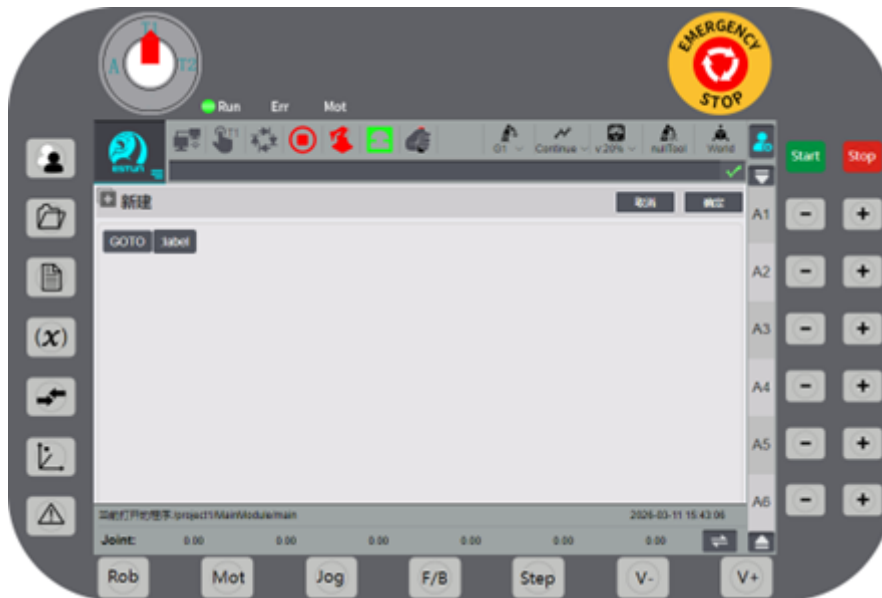
Label 指令用于定义 GOTO 跳转目标。



## 6.2.2 GOTO - 跳转指令

### 指令简介

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。



### 配置示例

LABEL 指令设置标签 label1，GOTO 指令执行时，跳转到行“label1:”

```
label1: MovJ(P0) GOTO label1
```

```
function config_property()
6   property["module_name"] = "MainModule"
7   property["description"] = ""
8   property["created_by"] = "program1"
9   property["create_time"] = "2025/03/05 07:01:18"
10  property["modify_time"] = "2025/03/05 08:50:44"
11  property["readonly"] = "false"
12  property["locked_axes"] = ""
13  lock_axis_property["MainModule"] = ""
14  end
15  function main()
16    var p0 :: Apos = Apos(-0.000000,-0.000000,0.000000,0.000000,0.000000)
17    label1 :
18    MovJ(p0)
19    GOTO label1
20  end
21  end
```

### 6.2.3 IF-条件判断指令

#### 指令简介

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。



- 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，则执行后续指令，若为假，则跳过后续指令。  
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。

#### 操作步骤

1. 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。
2. 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。



3. 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。

## 后续操作

- 删除运算符或操作数： 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。
- 增加括号：
  - 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。
- 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。

## 6.2.4 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。



### 📖 说明

- 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。

## 6.2.5 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。

## 6.2.6 WHILE - 只要满足条件就循环

### 指令简介

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。



## 配置举例

执行两点之间的循环运动 100 次。

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

### 6.2.7 FOR - for 循环指令

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，条件表达式，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

## 6.2.8 RETURN - 返回指令

### 指令简介

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。



## 配置示例

当满足条件，执行到 RETURN 指令，跳转到 END。

```
IFx >1THEN  
RETURN  
ENDIF  
X = X + 1  
END
```

## 6.2.9 SWITCH

### 指令简介

SWITCH-CASE 逻辑指令，以 SWITCH 关键字开头 ENDSWITCH 结束，其中可以添加 CASE、DEFAULT 语句等。

执行到 SWITCH 语句后，找到匹配的 CASE 直接跳转到对应 CASE 语句块里面执行。

添加的变量可以是 int、string 类型。



## 配置示例

```
SWITCH P.INTO
CASE 1
  Movj(P0)
  break
CASE 2
  Movj{P1}
  Movj{P2}
  break
end
```

### 说明

- SWITCH 语句不允许嵌套执行。
- CASE 语句后面变量类型必须与 SWITCH 类型匹配。
- SWITCH 和 ENDSWITCH 之间必须包含 CASE 或 DEFAULT 至少一个，且 DEFAULT 必须放在最后。
- SWITCH 后面如果是 int 类型 CASE 支持此种形式 CASE 1,2,3 匹配任何一个将执行到该 CASE 语句块。
- SWITCH 后面如果是 string 类型 CASE 不支持此种形式 CASE "hel-lo" , "h1" , "h3" 。

## 6.2.10 CASE

只能在 SWITCH 指令中使用，详情请参见 [SWITCH 指令](#)。



### 6.2.11 BREAK

只能在 SWITCH 指令中使用，详情请参见 [SWITCH 指令](#)。

### 6.2.12 DEFAULT

只能在 SWITCH 指令中使用，详情请参见 [SWITCH 指令](#)。

### 6.2.13 CONTINUE

continue用于在循环中跳过当前循环，进入下一次循环。通常与for，while循环一起使用。

### 6.2.14 ...=... - 赋值指令

#### 指令简介

建立一个表达式，用来给某个变量赋值。

变量A = 变量B。



## 参数说明

- 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。运算符说明请参见表 6-1 运算符介绍。



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

### 说明

- 在“替换”时，可选择“数学函数”或“字符函数”。
- 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。
- + ( )：添加 ( ) 运算符。
- - ( )：删除添加的 ( ) 运算符。

表 6-1 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，求商	~=	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

### 6.2.15 Try——catch指令

try—catch 逻辑指令。try中代码块运行时若有exception异常，由catch捕获，可在catch中进行对该exception的处理，finally中为进行异常处理后的退出程序。

```
14   end
15   function main()
16       var ExceptionInfo0 :: ExceptionInfo = ExceptionInfo(0, "0")
17       var ExceptionInfo1 :: ExceptionInfo = ExceptionInfo(0, "0")
18       try
19           throw(ExceptionInfo0)
20       catch test
21           ExceptionInfo1 = test::ExceptionInfo
22           if ExceptionInfo1.id == 1
23               Wait(1000)
24           else
25               Wait(100)
26           end
27       finally
28       end
29   end
30 end
```

其中，exception是异常数据类型中的数据异常变量。通过throw抛出异常，通过catch捕捉异常，通过赋值指令赋值该exception1，通过exception1的类型执行不同的动作。通过finally退出。

#### 📖 说明

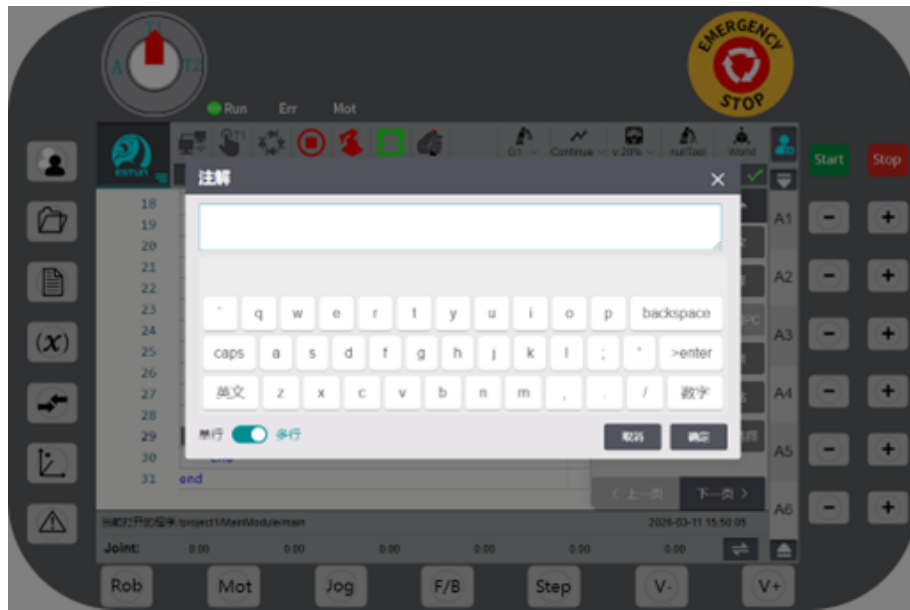
end以try关键字开头end结束。

### 6.2.16 FINALLY

finally用于指定无论是否发生异常都会执行的代码块，finally必须与try搭配使用。

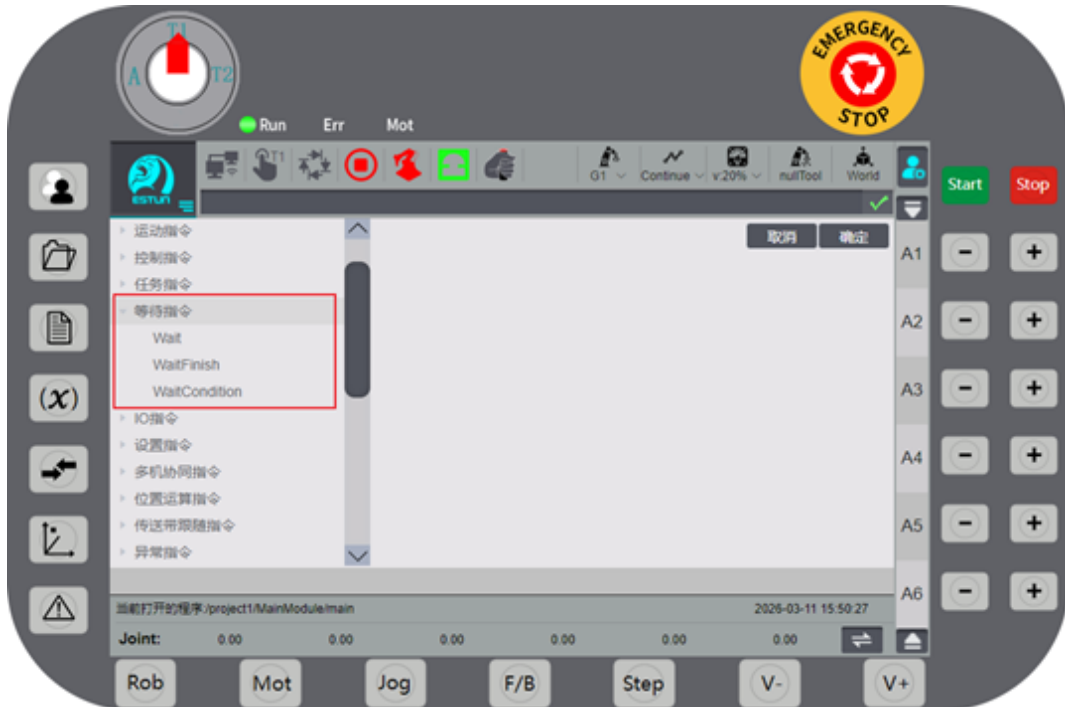
### 6.2.17 #=...=#- 注释指令

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

等待指令列表如图所示。



### 6.3.1 Wait - 延时等待指令

#### 指令简介

用于设置机器人等待时间，时间单位为 ms，可为常量或者 INT 变量类型。



## 配置示例

假如设置等待 1s，生成命令的两种方式为：使用常量的方式，和使用变量的方式。其中变量 INTO 赋值为 1000。

```
Wait(1000)
Wait(INT0)
```

## 6.3.2 WaitFinish - 等待运动段执行百分比指令

### 指令简介

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。



## 配置示例

如下示例程序所示，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 SetDO 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)
```

### 6.3.3 WaitCondition - 等待指定判断条件指令

#### 指令简介

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。



#### 参数说明

- 判断条件：当“判断条件”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。
- 超时跳转的标签名：当执行超时，选择跳转到标签名所在行。

## 配置示例

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。

```
WaitCondition("X > 0", 1000, 0, INT0)
```

## 6.4 IO指令

### 6.4.1 PulseOut - 设置脉冲输出指令

#### 指令简介

设定 IO 变量输出的脉冲时间与条件。



## 参数说明

- 端口号：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。

## 配置示例

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。

```
PulseOut (DO1, 1, 500, 1)
```

## 6.4.2 WaitIO - 等待IO信号输入指令

### 指令简介

该指令用于在指定时长内等待DI/SimDI端口被设置（1）或复位（0），或者AI/SimAI端口值与给定值相等。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。



## 参数说明

- IO类型：DI, SimDI, AI或SimAI, 表示需要等待的IO类型。
- 端口：需要等待的IO端口号。
- 端口值：当IO输入值和目标值相等时视为满足等待条件，程序可继续向下执行。
- 时长：等待时长，该参数用于指定等待某IO信号的时长，单位为ms。当值为0时，将强制等待直到信号满足时方认为执行成功。
- 操作返回值：ReturnValue变量，返回该指令执行结束后的状态，其中的int\_value参数表示：
  - 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：等待超时后跳转的label名称。

## 配置示例

等待数字量输入端口1的值为0的状态，等待1000ms，若1000ms内未等到该信号，则将ReturnValue1的int\_value参数置为1，表示等待超时。

```
WaitIO("DI",1,0.0,1000,ReturnValue0)
```

### 6.4.3 WaitIO8421 - 等待一组IO信号输入指令

#### 指令简介

该指令用于在指定时长内等待一组连续IO信号输入端口的状态组合。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。



#### 参数说明

- IO类型：DI或SimDI，表示等待的IO类型。
- 起始端口：该段连续端口的起始端口号。
- 结束端口：该段连续端口的结束端口号。
- 端口值：将该段连续端口值转换成的十进制数。
- 时长：等待该组信号的时长，单位ms。当值为0时，将强制等待直到信号满足时方认为执行成功。
- 操作返回值：ReturnValue变量，返回该指令执行结束后的状态，其中的int\_value参数表示：
  - 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：等待超时后跳转的label名称。

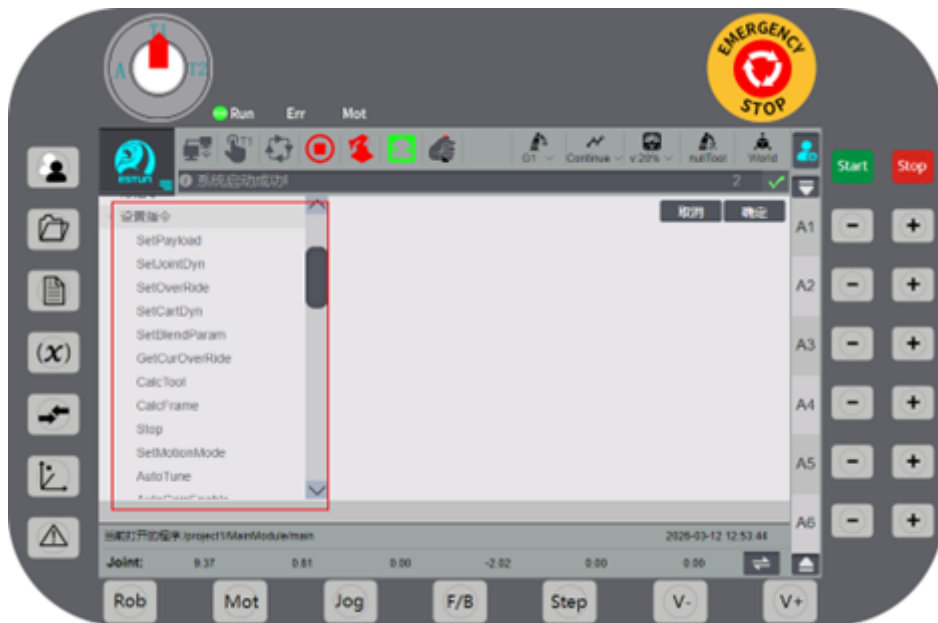
## 配置示例

在1000ms时长内等待1到3号DI口状态为111（10进制7）时，程序继续向下执行，若1000ms内未等到该信号，则将ReturnValue0的int\_value参数置为1，表示等待超时。

```
WaitIO8421("DI",1,3,7,1000,ReturnValue0)
```

## 6.5 设置指令

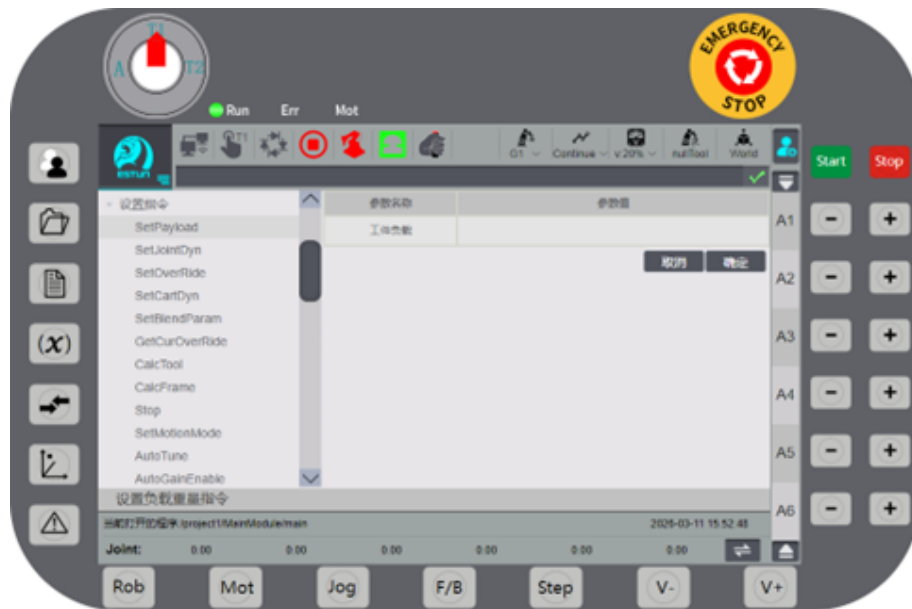
设置指令列表如下图所示。



### 6.5.1 SetPayload - 设置负载重量指令

#### 指令简介

选择工件负载参数指令。



## 配置示例

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。

```
SetPayload (PAYLOAD0)
```

## 6.5.2 SetJointDyn - 设置关节运动参数指令

### 指令简介

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。



## 参数说明

- 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。

## 配置示例

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

### 6.5.3 SetCartDyn - 设置直线或圆弧运动动态参数指令

#### 指令简介

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。



#### 参数说明

- 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。

## 配置示例

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

## 6.5.4 SetBlendParam - 设置过渡参数

### 指令简介

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。



### 参数说明

- 过渡类型 (BlendType)：机器人逼近终点时的渡方式。
  - FINE：无过渡，精确到位。
  - 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。

## 配置示例

如执行SetBlendParam("RELATIVE",C20)指令后，后续运动指令均以该过渡参数运行。

## 6.5.5 SetColliEnable - 设置碰撞检测使能指令

### 指令简介

设置碰撞检测使能指令。



### 参数说明

- 轴号： "GROUP、 A1、 A2、 A3、 A4、 A5、 A6"。当轴号为"GROUP"时，表示所有轴。
- 有效性： ENABLE（开）、 DISABLE（关）

如执行SetColliEnable("GROUP","ENABLE")开启所有轴的碰撞检测使能。

## 6.5.6 SetAxisColliParam - 设置碰撞检测灵敏度指令

### 指令简介

设置碰撞检测灵敏度值指令。



## 参数说明

- 轴号: "A1、A2、A3、A4、A5、A6"。
- 灵敏度值: 单位为百分比, 参数范围为[1,500]。

## 配置示例

如执行SetAxisColliParam ("A1",200)指令设置轴 1 的碰撞检测灵敏度值为 200。

## 6.5.7 GetCurOverRide - 获取当前全局速度指令

### 指令简介

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。



## 配置示例

如执行GetCurOverride (INT0)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。

## 6.5.8 CalcTool - 计算工具参数

### 指令简介

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。

#### 📖 说明

四个标定点和工具参数均需提前在变量管理里设置好相应变量。



## 配置示例

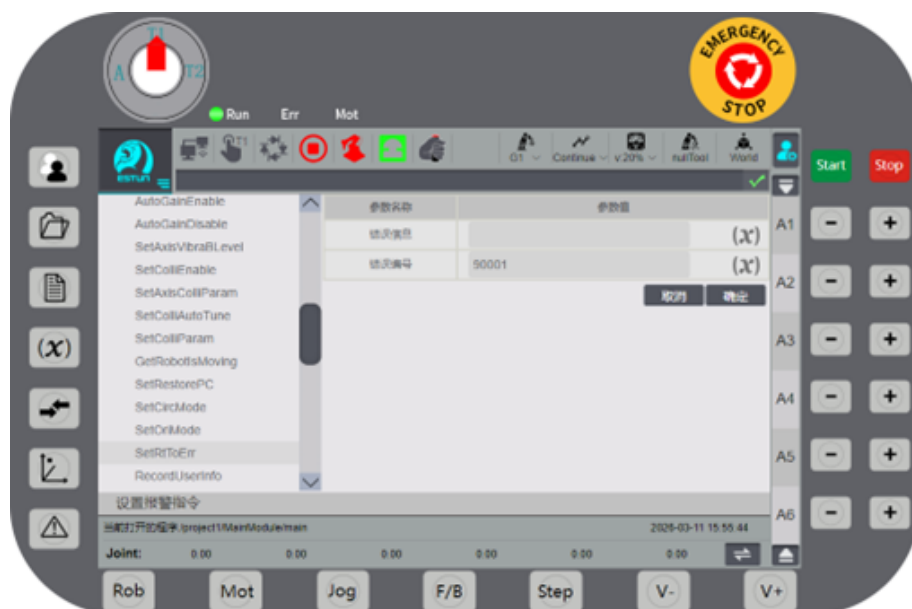
根据所设定的四个标定点来计算工具参数，并保存至 TOOL1。

```
CalcTool (P1, P2, P3, P4, TOOL1)
```

## 6.5.9 SetRtToErr - 设置报警指令

### 指令简介

设置机器人停止运行并发出错误信息。



## 参数说明

- 错误信息：设置错误信息，支持中英文输入。也可设置为系统中已定义的变量，只支持参数为 String 类型的变量。
- 错误编号：设置错误编号。可手动输入“错误编号”，也可设置为系统中已定义的变量，只支持参数为 INT类型变量，且参数范围为【90001-100000】。

## 配置示例

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。

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

### 6.5.10 SetRtInfo - 设置提示信息

#### 指令简介

设置提示信息但不影响机器人的运行。



#### 参数说明

- 提示信息：设置提示信息，支持中英文输入。也可设置为系统中已定义的变量，只支持参数为 String 类型的变量。
- 提示编号：设置提示编号。可手动输入“提示编号”，也可设置为系统中已定义的变量，只支持参数为 INT类型变量，且参数范围为【70001-80000】。

## 配置示例

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。

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

## 6.5.11 SetRtWarning - 设置报警信息

### 指令简介

设置警告信息但不影响机器人的运行。



### 参数说明

- 告警信息：设置告警信息，支持中英文输入。也可设置为系统中已定义的变量，只支持参数为 String 类型的变量。
- 告警编号：设置告警编号。可手动输入“告警编号”，也可设置为系统中已定义的变量，只支持参数为 INT 类型变量，且参数范围为【80001-90000】。

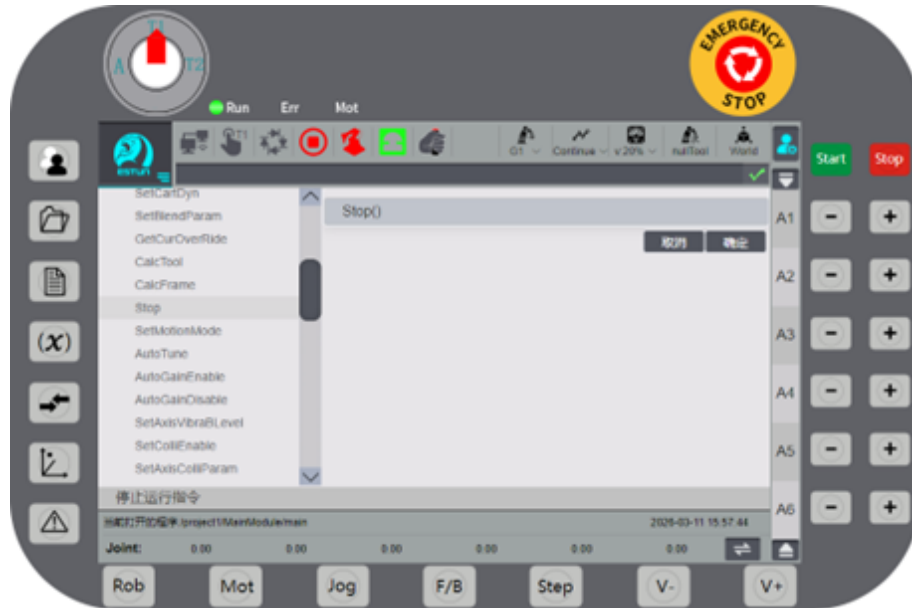
### 配置示例

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。

```
SetRtWarning ( "Warning" , 80001)
```

### 6.5.12 Stop - 停止运行指令

该命令用于停止所有激活程序的执行。



### 6.5.13 SetMotionMode - 设置运行模式

#### 指令简介

该指令仅能在管理人员权限下创建。设置运行模式。



## 参数说明

运动模式：

- STANDARD：标准
- LOW\_SPEED\_ACCURACY：低速高精
- DENSE\_POINT：小线段
- "USER\_MODE1：用户模式1
- "USER\_MODE2：用户模式2
- "USER\_MODE3：用户模式3
- "USER\_MODE4：用户模式4

## 配置示例

将机器人当前的运动模式设置为标准运动模式。

```
SetMotionMode ( "STANDARD" )
```

## 6.5.14 AutoTune - 提速自调节指令

### 指令简介

设置自调节功能类型。



## 参数说明

自调节功能类型：“ADP”。

### 说明

- 该指令仅能在管理人员权限下创建。
- 当前版本仅支持ADP类型。

## 6.5.15 AutoGainEnable - 设置伺服参数增益指令

### 指令简介

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。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。



### 参数说明

- 低速段增益比例 (LsScale)：对应各个关节轴，设置范围：[100, 1000]，单位：%。
- 低速段速度阈值 (LsThresh)：对应各个电机轴，设置范围：[10, 1000]，单位：r/min。
- 伺服参数增益指令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。

## 配置示例

伺服参数自调节功能开启。

```
AutoGainEnable(LsScaleP=t_g.LsScale0,LsThreshP=t_g.LsThresh0)
```

### 6.5.16 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。

如执行AutoGainDisable()伺服参数自调节功能关闭。

### 6.5.17 SetAxisVibraBLevel - 设置振动抑制等级指令

#### 指令简介

振动抑制等级设置命令，只对输入整形有效。



#### 参数说明

- 轴号：将要设置振动抑制等级的轴号。
- 强度：将要设置振动抑制的等级。包括Highest、High、Default、Low、Lowest五个等级。

## 配置示例

将 1 轴的振动抑制等级设为中高。

```
SetAxisVibraBLevel(A1,High)
```

## 6.5.18 SetRestorePC - 设置恢复运行时的 PC 指令

### 指令简介

该指令可设置多任务配置中task绑定的程序由暂停/停止转到运行时的 pc 指针。

#### 📖 说明

该指令只在 task绑定的程序中使用有效。



### 参数说明

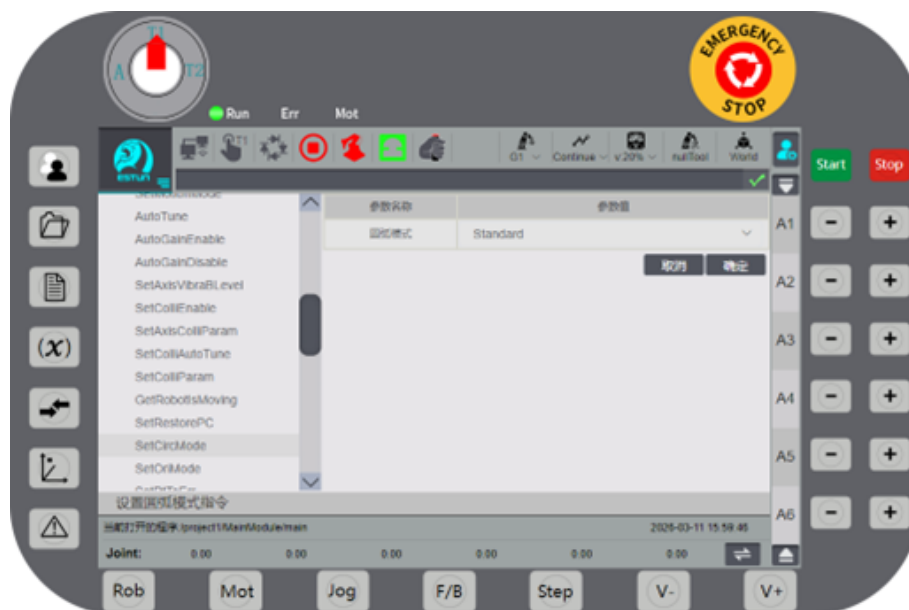
- 程序指针： task程序恢复运行时的行号。

### 配置示例

待task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。

```
SetRestorePC(3)
```

### 6.5.19 SetCircMode - 设置圆弧模式指令



圆弧模式：打开或关闭圆弧过中间辅助点功能。

- Standard：关闭圆弧过中间辅助点姿态功能。
- MidPosePass：打开圆弧过中间辅助点姿态功能。

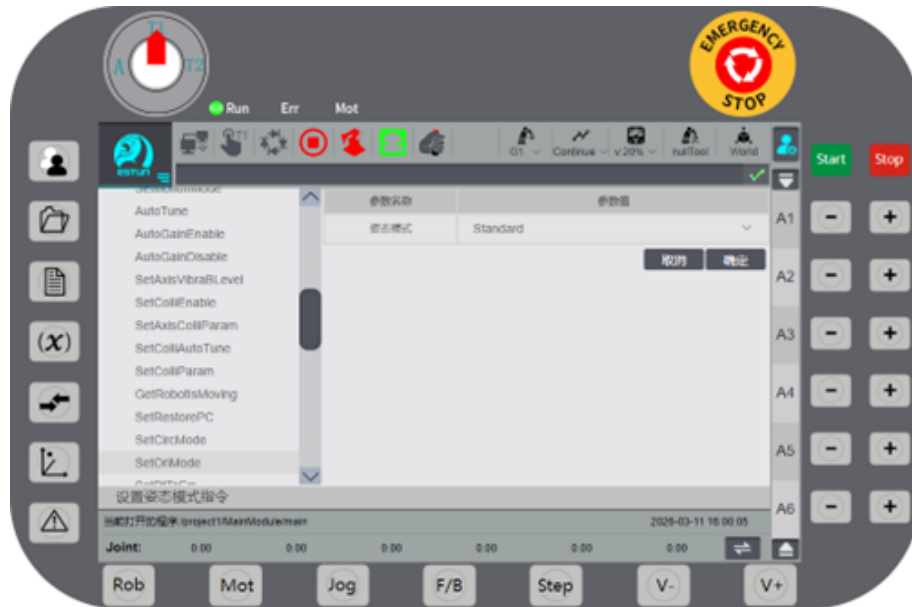
---

#### 📖 说明

该功能只对 SCARA 机型生效。

---

### 6.5.20 SetOriMode - 设置姿态模式指令



姿态模式：机器人姿态插补模式。

- Standard：按照标准模式运动到目标点。
- Teach：按照实际示教点位运动到目标点。
- Shortest：J4 以最少旋转量到达目标点，若角度差小于等于  $180^\circ$ ，则 J4 完全运动到目标点；若 J4 旋转量大于  $180^\circ$ ，机器人反向运动到目标点；但是当反向运动超出机器人极限范围时，采取原方向继续运动（例如：起始点 J4 为  $280^\circ$ ，终点 J4 为  $40^\circ$ ，此时 J4 由  $280^\circ$  经由  $0^\circ$  旋转到  $40^\circ$ ）。

#### 📖 说明

该功能只对 SCARA 机型生效。

### 6.5.21 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。

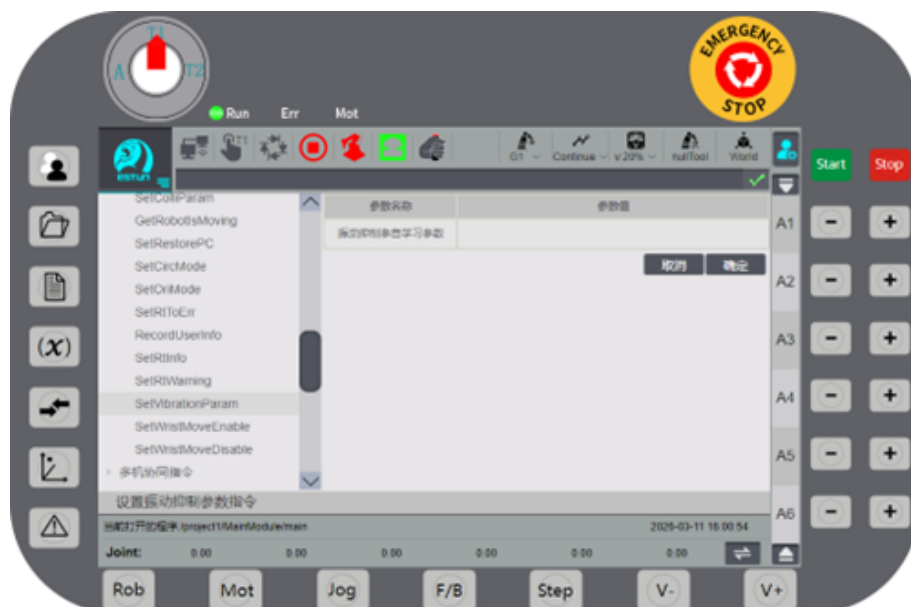


- 返回值：用户可以获取当前机器人是否处于运动中。

## 6.5.22 SetVibrationParam - 设置振动抑制自学习参数

### 指令简介

该指令可以设置机器人运动的自学习振动抑制参数，包括默认振动抑制参数和自学习振动抑制参数。使用该指令需要新建振动抑制自学习变量，其中index为振动抑制自学习id编号，范围为0-6。



## 参数说明

- 振动抑制参自学习参数 (VibraParam)：指定自学习振动抑制参数，0表示默认振动抑制参数，而1-6表示自学习振动抑制参数。

## 配置示例

设置默认振动抑制参数。

```
SetVibrationParam(VibrationParam0)
```

## 6.5.23 SetWristMoveEnable

### 指令简介

SetWristMoveEnable，设置腕部运动功能指令使能。

通过该指令开启WristMove运动模式后，机器人将由指定J4、J5、J6中的任意两个轴实现笛卡尔空间运动。共三种运动模式可供设定：“Wrist45”、“Wrist46”、“Wrist56”。



### 参数说明

- 运动模式
  - Wrist45: 仅机器人4轴、5轴运动。
  - Wrist46: 仅机器人4轴、6轴运动。（不推荐使用该模式）
  - Wrist56: 仅机器人5轴、6轴运动。

## 应用场景

对小圆等几何形状切割加工而言，机器人1~3轴的摩擦效应以及伺服跟踪精度等因素会显著削弱其加工轨迹的圆度与精度。WristMove指令通过指定腕部两个轴的运动来控制切割工具，用于薄板激光切割小圆（推荐直径小于10mm）等加工应用场景。

### 说明

建议在小圆的圆心处开启该功能，调整机器人腕部姿态，确保开启点处的TCP Z向与切割平面垂直，以提高切割截面的垂直度。

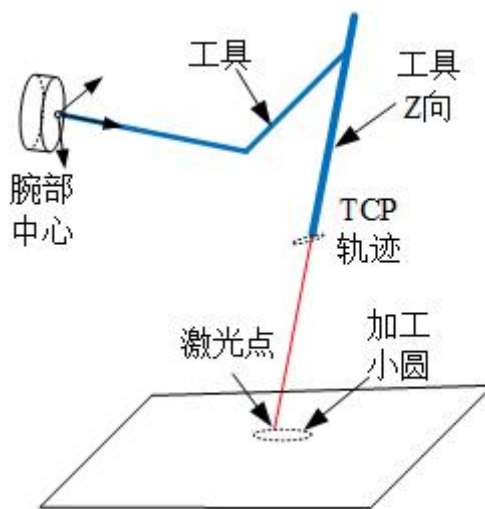


图 6-2 WristMove激光切割小圆示意图

## 限制及约束

- 仅通用六关节机器人支持WristMove功能。
- 该功能支持除MovLSearch的笛卡尔空间运动指令，如：MovL、MovC、MovCircle等，不支持点动。
- 该功能不支持与其他运动模式之间的运动过渡。
- 该功能仅保证实际工作点（如激光点）的位置，无法保证其姿态。
- 该功能需设置实际的工具尺寸参数，且尺寸参数的准确性影响实际运动效果。
- 该功能适用于切割小圆、矩形、三角形等，不适用于长路径运动。
- TCP标定时要求其Z方向必须与切割工具作用方向(比如激光、水刀)重合。
- 功能开启后需低速执行该运动指令以检查并确定不会发生干涉或者不可及等情况。

- 在SetWristMoveEnable与SetWristMoveDisEnable之间设置Pc不改变运动学方式，在两条开关指令之外设置Pc，运动学切换为正常运动学。
- 在开启SetWristMoveEnable后，需注意在退出腕部移动模式后需要使用SetWrist Move-Disable以关闭该功能。

### 6.5.24 SetWristMoveDisable

SetWristMoveDisable，腕部运动功能关闭指令。

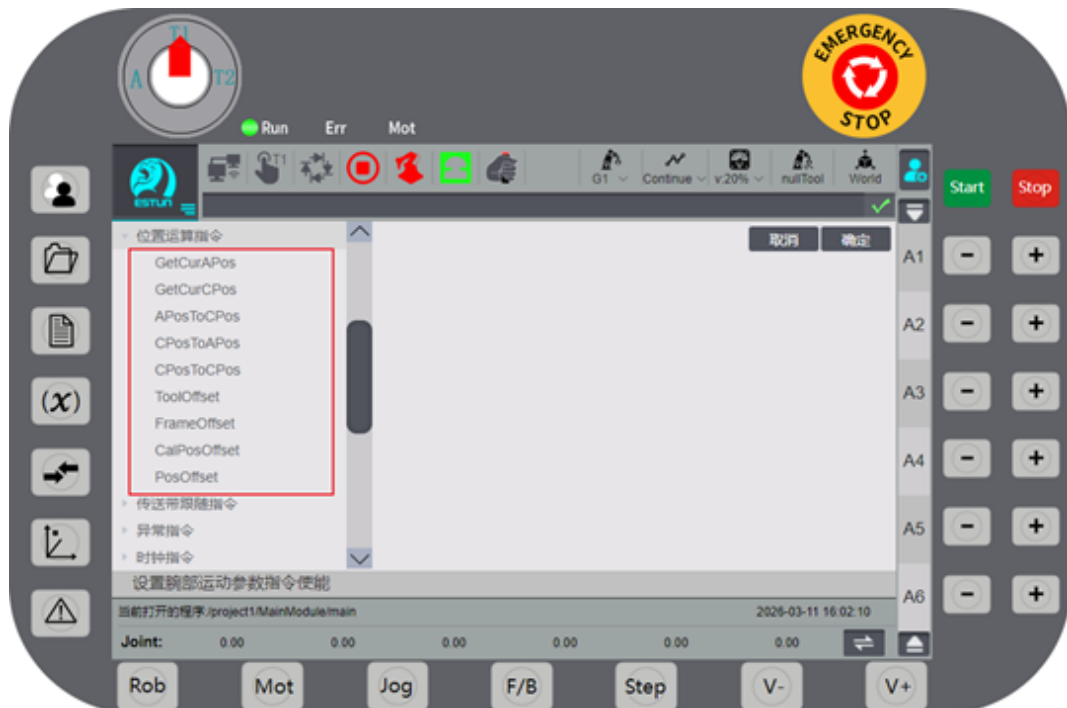
通过该指令关闭机器人WristMove运动，功能关闭后，机器人切换为正常运动学。该指令可以设置三个参数：“Wrist45”、“Wrist46”、“Wrist56”。

#### 说明

应根据所开启的腕部运动功能指令，关闭相应的腕部运动功能。

## 6.6 位置运算指令

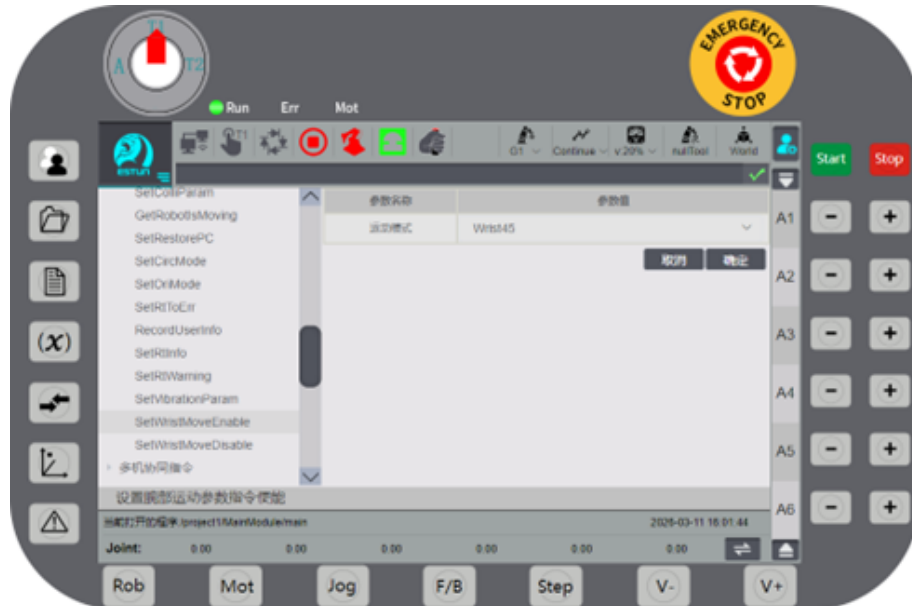
位置运算指令列表如下图所示。



## 6.6.1 GetCurAPos - 获取当前关节位置指令

### 指令简介

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。



### 配置示例

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

## 6.6.2 GetCurCPos - 获取当前笛卡尔坐标系位置指令

### 指令简介

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。



## 配置示例

执行GetCurCPos (cp0)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cp0中。

### 6.6.3 APosToCPos - 机器人位置转换指令（Apos 转换成 Cpos）

#### 指令简介

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。



## 参数说明

- 原位置：转换前关节位置。
- 目标位置：目标笛卡尔空间位置。

## 配置示例

如执行APosToCPos (cp0, P0)根据cp0可以得到目标点P0。

### 6.6.4 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。



- 原位置：转换前的笛卡尔空间位置。
- 目标位置：转换后的关节位置。

如执行APosToCPos (cp0, P0)根据 cp0可以得到目标点 P0。

## 6.6.5 CPosToCPos - 机器人位置转换指令 (Cpos 转换成 Cpos)

### 指令简介

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。



## 参数说明

- 原位置：转换前的笛卡尔空间位置。
- 目标位置：转换后的关节位置。

## 配置示例

执行CPosToCPos (cp0, cp1)根据 cp0可以得到目标点cp1。

## 6.6.6 CalPosOffset - 计算位置点偏移结果指令

### 指令简介

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。



## 参数说明

- 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。
- 目标位置：偏移后的最终位置（CPOS）。

### 6.6.7 FrameOffset - 坐标系偏移指令

#### 指令简介

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



## 参数说明

- 基准坐标系：转换前的参考坐标系参数。
- 偏移量：相对原位置所要移动的位置增量（DCPOS）。
- 目标坐标参数：目标坐标系的值。

### 6.6.8 ToolOffset-工具坐标系偏移指令

#### 指令简介

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



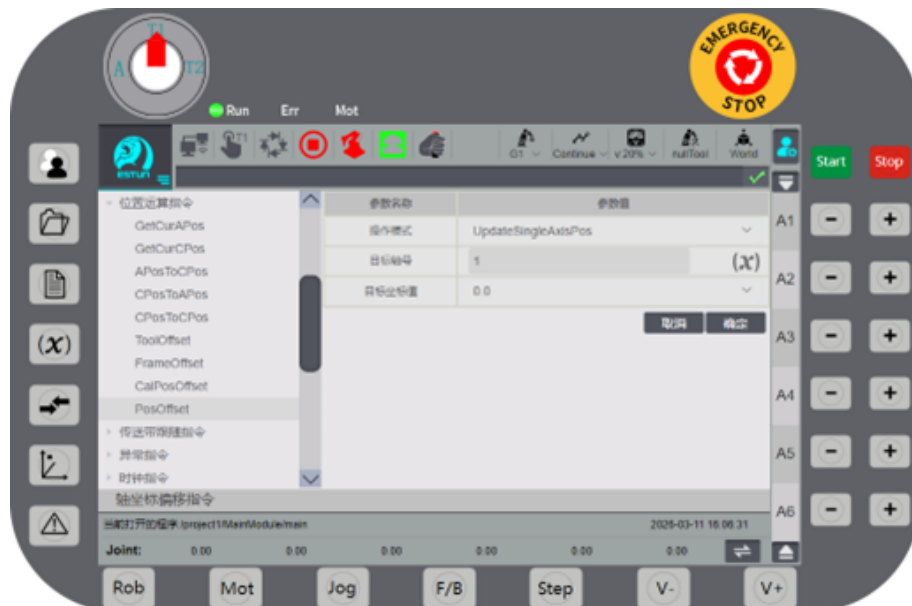
### 参数说明

- 基准工具参数：转换前的工具坐标系参数。
- 偏移量：相对原位置所要移动的位置增量（DCPOS）
- 目标工具参数：目标工具坐标系的值。

## 6.6.9 PosOffset - 轴坐标偏移指令

### 指令简介

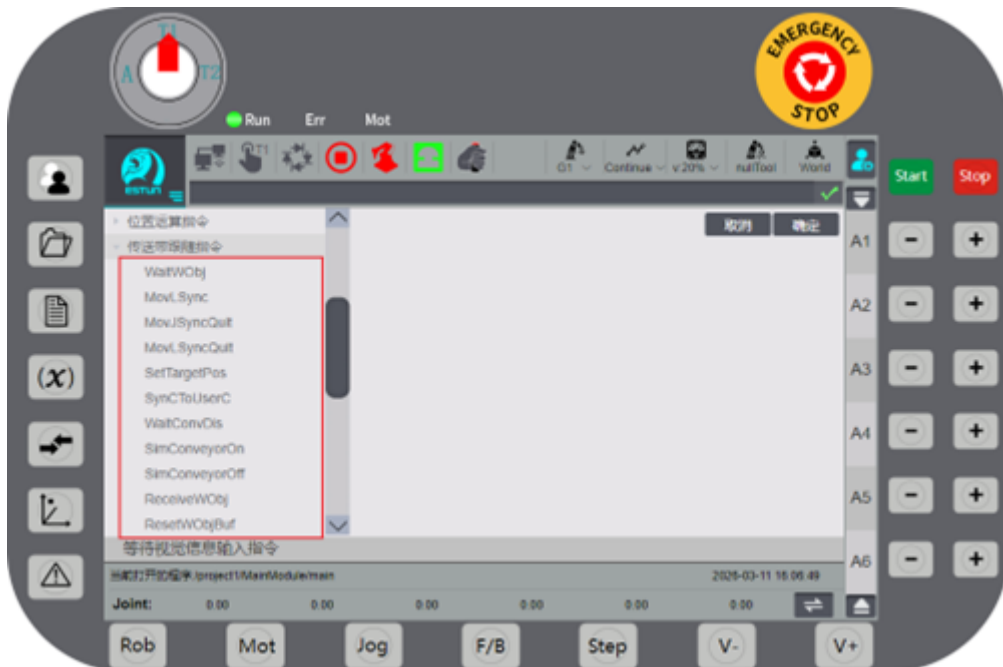
对程序域中点位变量的附加轴坐标进行偏移运算，并将偏移后的坐标更新到内存与文件中。



## 参数说明

- 操作模式：需要对指定轴坐标进行的偏移方式，有"UpdateSingleAxisPos"与"ExchangeAxisPos"两种模式可选。当该指令在等待时间内执行完成时，对该变量赋值 0。
  - UpdateSingleAxisPos：单轴坐标更新模式，将当前程序域点位变量的目标轴坐标强制更新为目标坐标值，并更新到内存与文件中。
  - ExchangeAxisPos：双轴坐标互换模式，将当前程序域点位变量的目标轴坐标与基准轴坐标值互换，并更新到内存与文件中。
- 目标轴号：需要进行坐标偏移的轴号。
- 目标坐标值：需要将目标轴设置的目标值，该参数仅在操作模式为“UpdateSingleAxis-Pos”才有效。

## 6.7 传送带跟随指令



### 6.7.1 WaitWObj - 等待视觉信息输入指令

#### 指令简介

等待视觉信息指令，等待视觉发来的工件位置信息。



## 参数说明

- 动坐标系：Frame 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。
  - 若该参数的值为 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。
  - 若该参数的值非 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。
- 相机反馈Attr：相机反馈技术可以使工件高速且无停顿的对位，且精准贴合目标。
- 相机反馈的点位：通过图像传感器测量工件的位置和角度，基于这些信息实时反馈并移动平台。
- 超时跳转的标签名：当执行超时，选择跳转的标签名所在行。

## 6.7.2 MovLSync - 启动传送带同步的直线运动指令

### 指令简介

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。

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。



### 参数说明

- 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。
- 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。
- 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，需要选用系统预定义。
- 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。
- 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 PulseOut、WaitIO、WaitIO8421。

### 6.7.3 MovJSyncQuit - 同步跟随退出指令

#### 指令简介

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。

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。



## 参数说明

- 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，需要选用系统预定义。
- 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：精确到位。
  - 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。
- 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，需要选用系统预定义；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。
- 退出点使能：执行完本次跟随轨迹，是否退出的使能状态。
  - DEFAULT：默认值，表示沿用最近一次设置的使能状态。
  - ENABLE：执行完本次跟随轨迹后退出跟随。
  - DISABLE：执行完本次跟随轨迹后不退出跟随，直接参与下一次的跟随运动。
- 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 PulseOut、WaitIO、WaitIO8421。

### 6.7.4 MovLSyncQuit - 同步跟随退出指令

#### 指令简介

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。

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。



## 参数说明

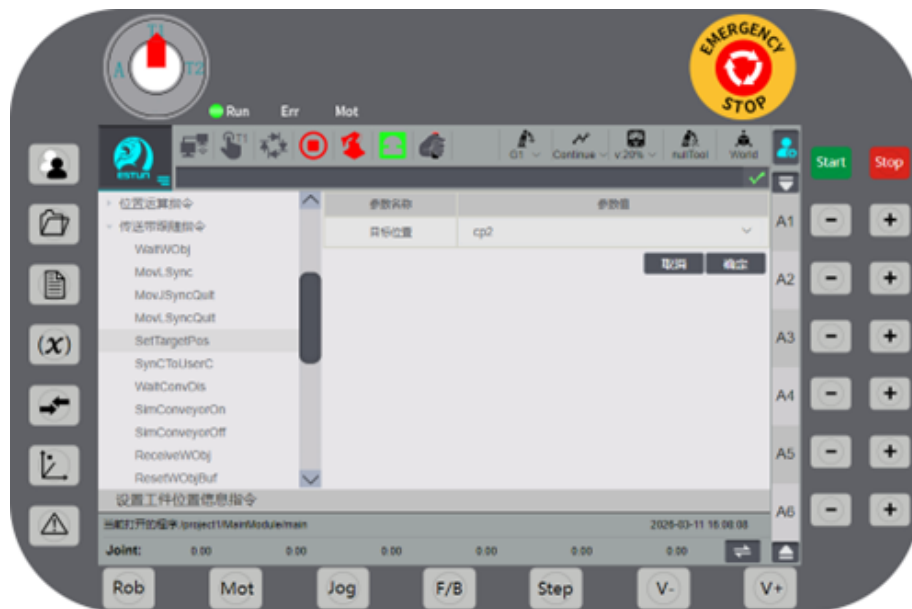
- 目标位置：指令终点位置（APOS 或 CPOS 类型位置点）。在下拉列表中选择已有的点或在弹出的对话框中新建并示教一个变量。
- 目标速度：指令运行速度。
  - 设置为“DEFAULT”，表示使用默认值 V4000。
  - 设置为 SPEED 类型变量时，需要选用系统预定义。
- 过渡类型：机器人逼近终点时的过渡方式。
  - DEFAULT：缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
  - FINE：精确到位。
  - RELATIVE：相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段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

- 退出点使能：执行完本次跟随轨迹，是否退出的使能状态。
  - DEFAULT：默认值，表示沿用最近一次设置的使能状态。
  - ENABLE：执行完本次跟随轨迹后退出跟随。
  - DISABLE：执行完本次跟随轨迹后不退出跟随，直接参与下一次的跟随运动。
- 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PulseOut、WaitIO、WaitIO8421。

## 6.7.5 SetTargetPos - 设置工件位置信息指令

### 指令简介

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。



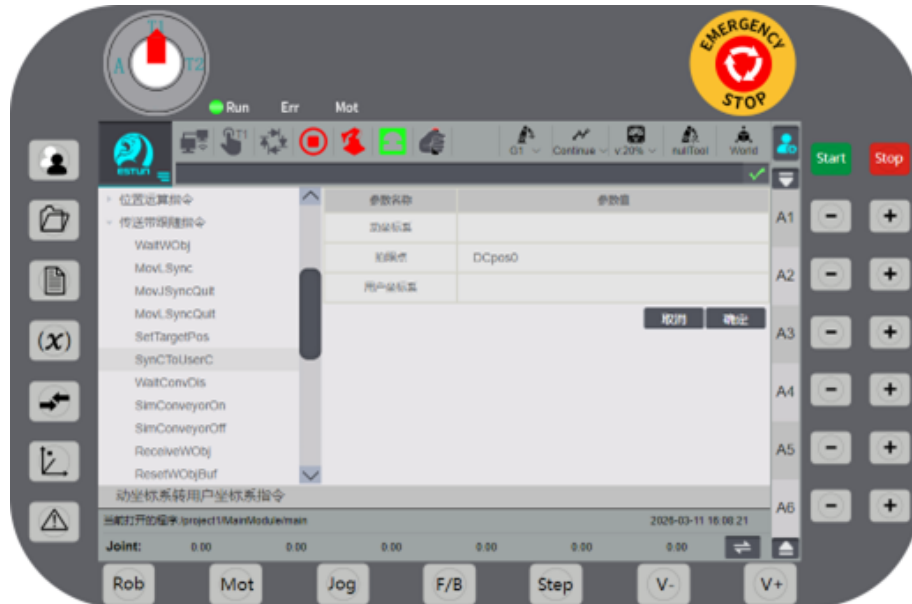
### 参数说明

- 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。

## 6.7.6 SynCToUserC - 示教参考坐标系转换指令

### 指令简介

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。



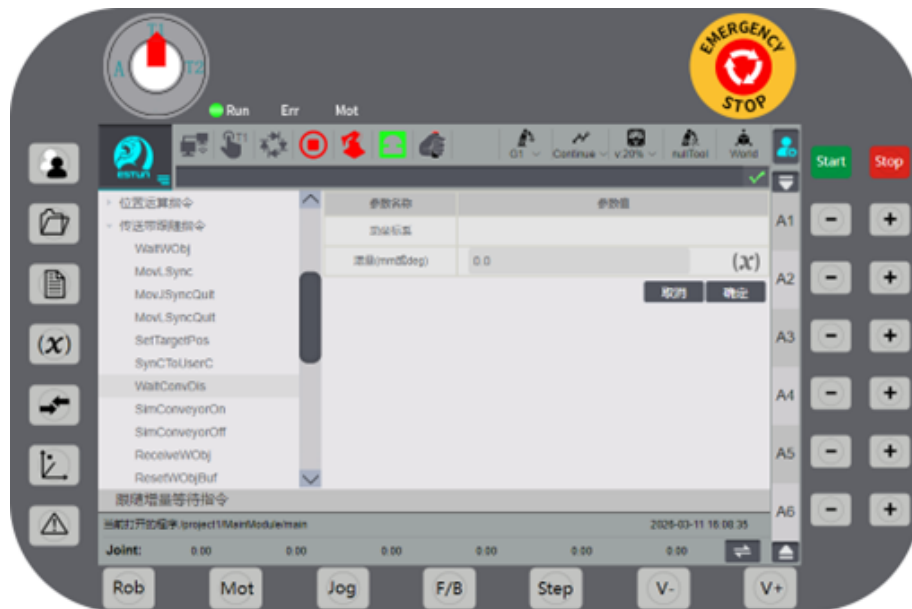
### 参数说明

- 动坐标系：Frame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。
- 拍照点：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。
- 用户坐标系：Frame 类型变量，指示教跟随同步后轨迹点位时参考的坐标系。

## 6.7.7 WaitConvDis - 等待固定增量指令

### 指令简介

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。



## 参数说明

- 动坐标系：Frame 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。
- 增量距离（ms）：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。

## 6.7.8 SimConveyorOn - 虚拟传送带开启指令

### 指令简介

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。



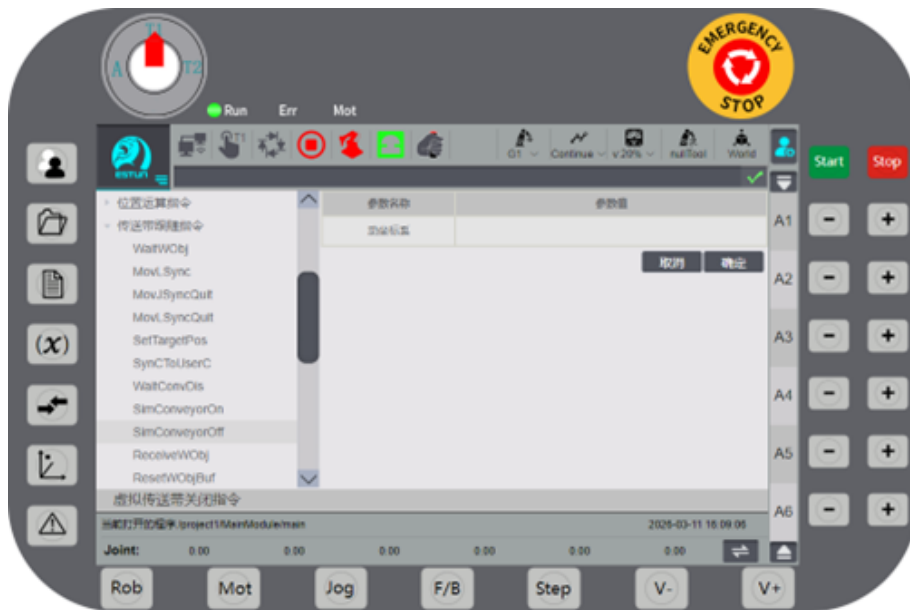
## 参数说明

- 动坐标系：Frame 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。

## 6.7.9 SimConveyorOff-虚拟传送带关闭指令

### 指令简介

虚拟传送带关闭指令，用于禁能虚拟传送带。



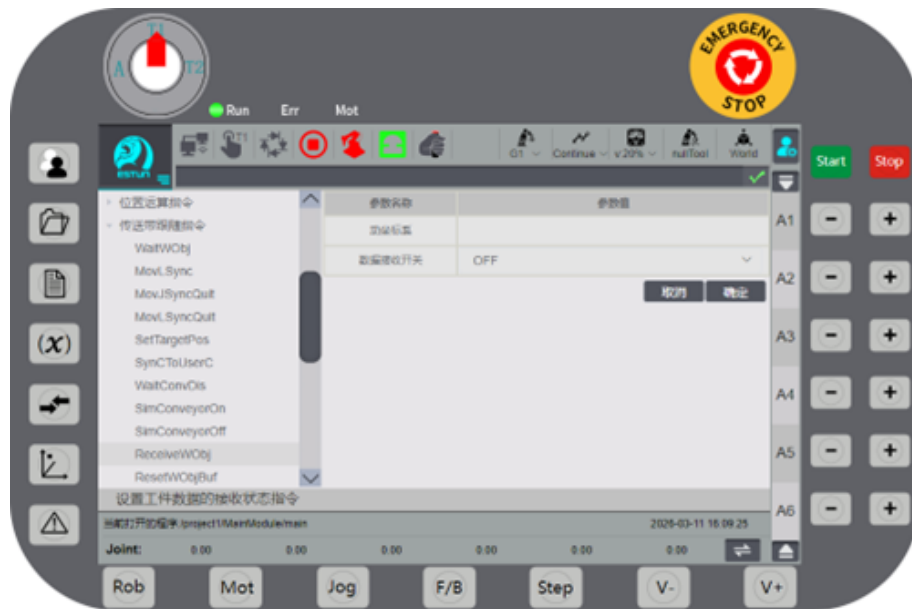
## 参数说明

- 动坐标系：Frame 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。

## 6.7.10 ReceiveWObj-工件数据接收状态指令

### 指令简介

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task 程序中发来的工件位置信息。



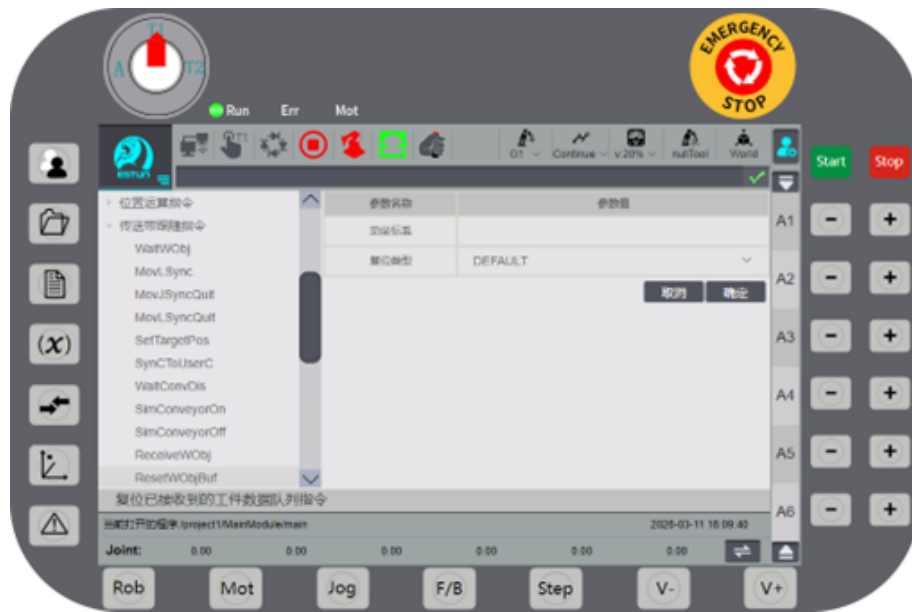
## 参数说明

- 动坐标系：Frame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。
- 数据接收开关：使能机器人系统对工件位置信息的接收状态。

## 6.7.11 ResetWObjBuf - 复位已工件数据队列指令

### 指令简介

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。

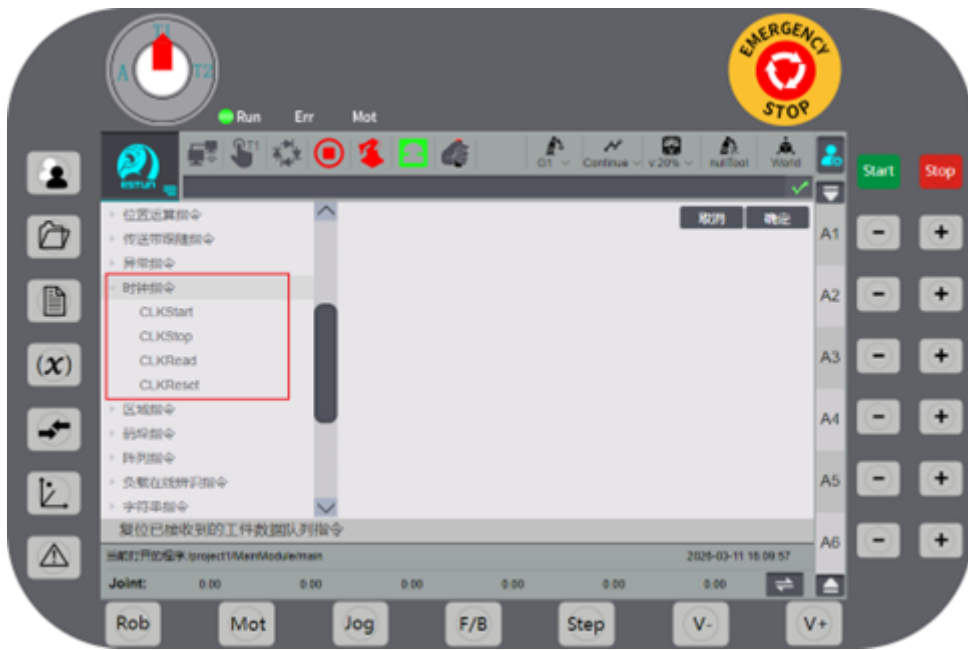


## 参数说明

- 动坐标系：Frame 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。
- 复位类型：选择适合的复位方式。“ALL”、“SELECT”、“CURRENT”三种类型。
  - ALL：默认值，表示将已接收的队列数据全部清除。
  - SELECT：将队列中与传入的视觉 id 号一致的数据全部删除。
  - CURRENT：将队列中的第一个数据清除。

## 6.8 时钟指令

时钟指令列表如下图所示。



### 6.8.1 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）。

如执行CLKStart (CLOCK0)启动时钟 CLOCK0。

### 6.8.2 CLKStop - 停止时钟指令

停止指定时钟（其 state 为 false，但不会复位）。

如执行CLKStop (CLOCK0)停止时钟 CLOCK0。

### 6.8.3 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。

如执行CLKRead (CLOCK0)读取时钟存于 CLOCK0。

### 6.8.4 CLKReset - 复位时钟指令

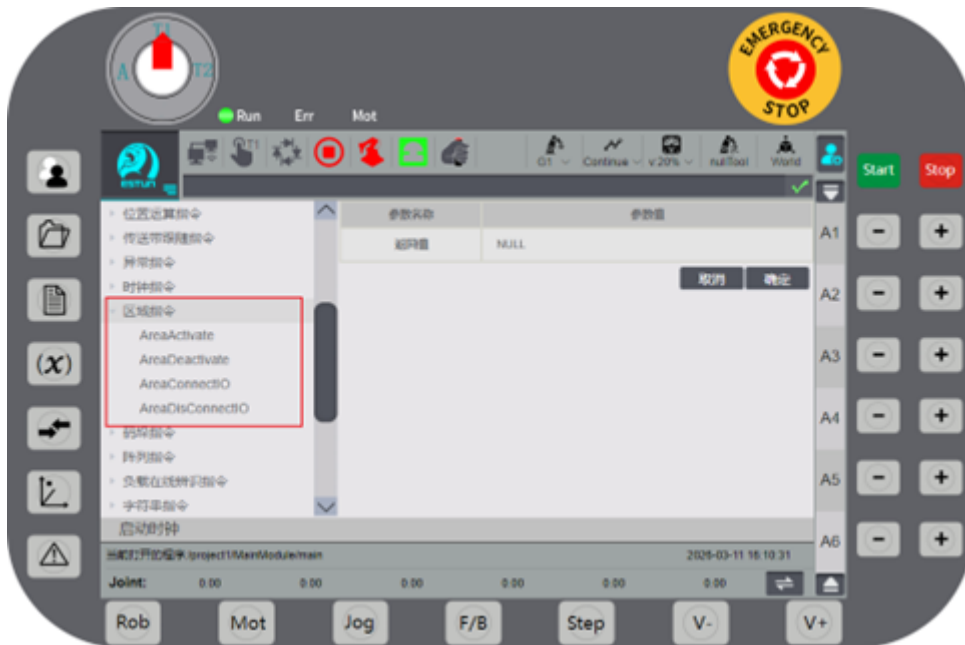
复位指定时钟的状态值。

如执行CLKReset (CLOCK0)复位时钟 CLOCK0。



## 6.9 区域指令

区域指令列表如图所示。



### 6.9.1 AreaActivate - 激活区域指令

激活区域指令，使指定的区域生效。



如执行AreaActivate (AREA0)激活区域监控，使 AREA0 生效。

## 6.9.2 AreaDeactivate - 冻结区域指令

冻结区域指令，使指定的区域失效。

如执行AreaDeactivate (AREA0)冻结区域监控，使 AREA0 失效。

## 6.9.3 AreaConnectIO - 绑定区域 IO 指令

### 指令简介

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。



## 参数说明

- 区域变量：AREA，需要绑定的区域变量。
- IO 类型：需要绑定的 IO 口类型。
  - DI：实际 DI。
  - SIMD I：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。

## 6.9.4 AreaDisConnectIO - 解除绑定区域 IO 指令

### 指令简介

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。

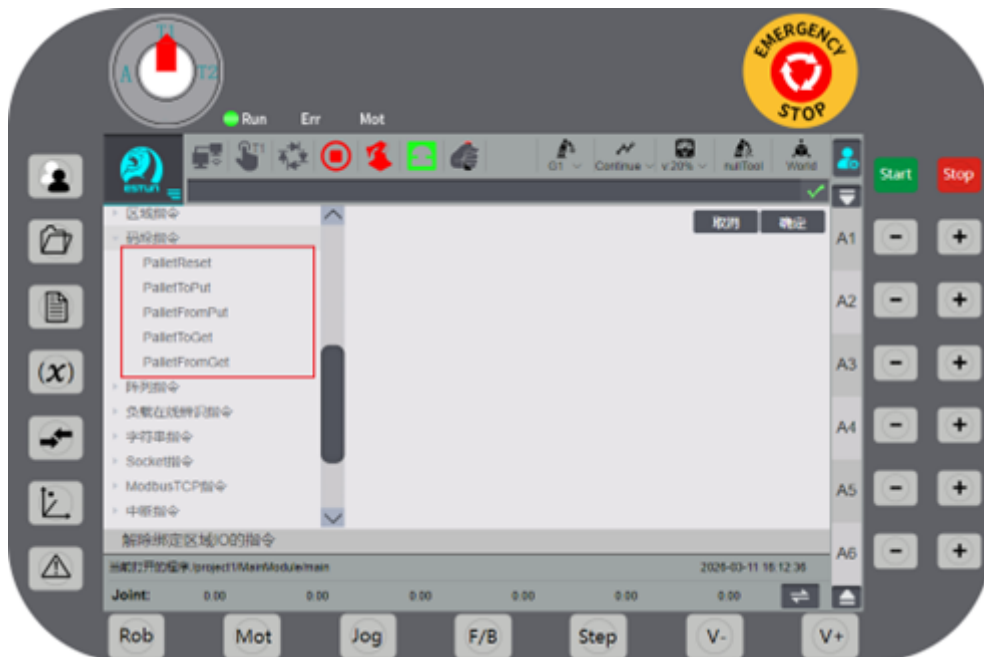


### 参数说明

- 区域变量：需要解除绑定的区域变量。

## 6.10 码垛指令

码垛指令列表如图所示。

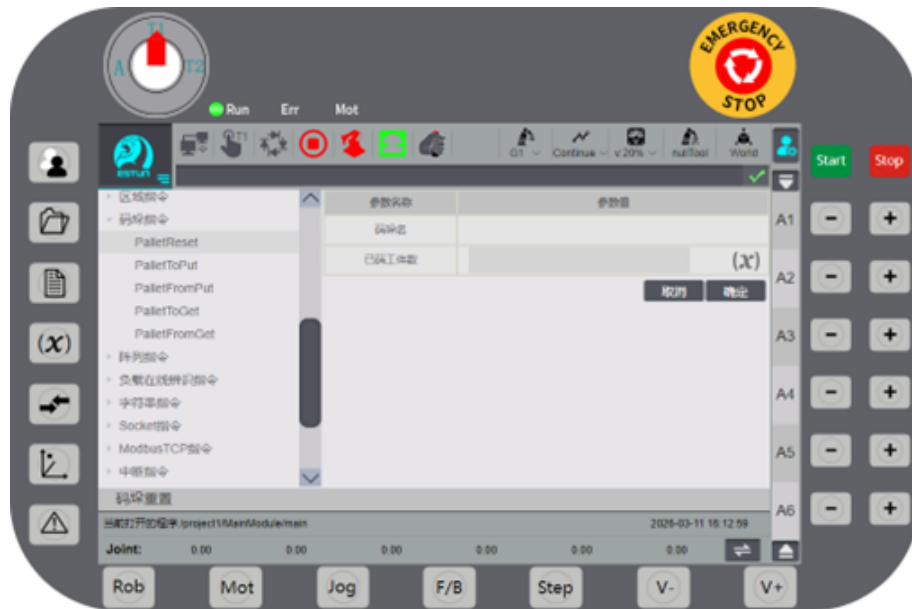


## 6.10.1 PalletReset - 码垛重置指令

### 指令简介

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。

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。



### 参数说明

- 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。

## 6.10.2 PalletToPut - 放置工件指令

### 指令简介

放置工件指令。

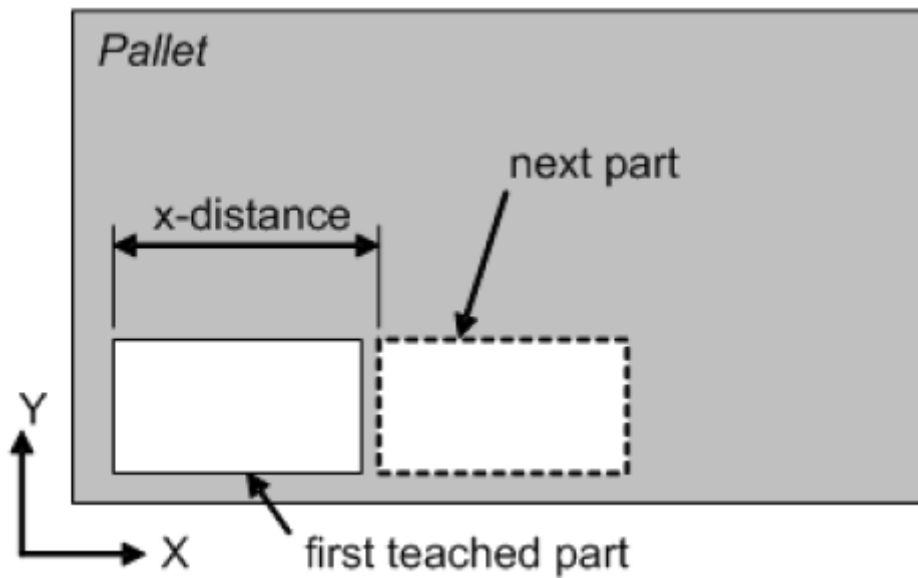


## 参数说明

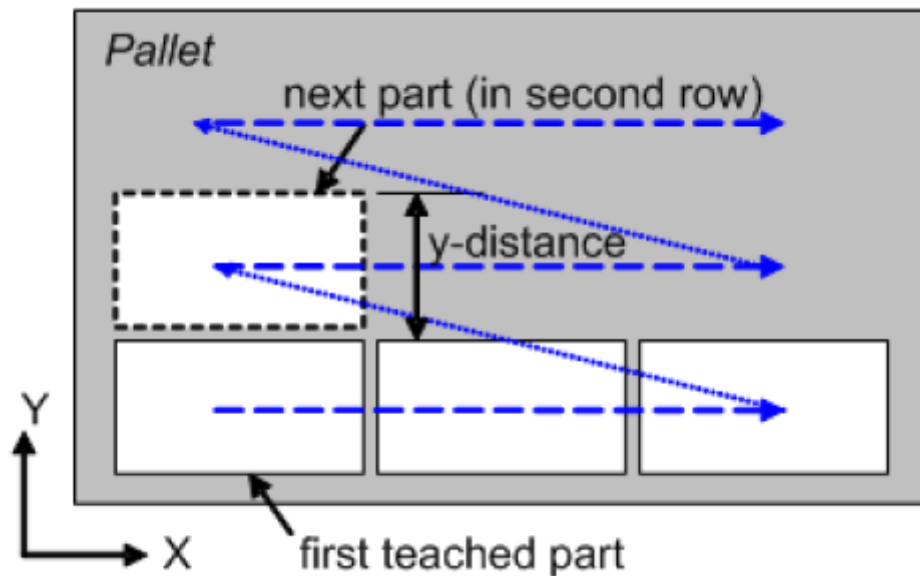
- 码垛名：当前码垛的PALLET
- 目标速度：指令运行速度。
  - 设置为“DEFAULT”，表示使用默认值V4000。
  - 设置为SPEED类型变量时，需要选用系统预定义。
- 过渡类型：机器人逼近终点时的过渡方式。
  - DEFAULT:缺省，即指令中不设置过渡参数。若程序中使用了全局过渡参数（SetBlendParam），过渡方式按照全局设置的过渡参数运行。
  - FINE：效果等同于DEFAULT。
  - RELATIVE:相对过渡，过渡参数为相对值，表示过渡点到轨迹终点的距离相对于整段轨迹长度的百分比。
  - ABSOLUTE:绝对过渡，过渡参数是个绝对值，表示过渡点到轨迹终点的距离。

## 配置示例

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。



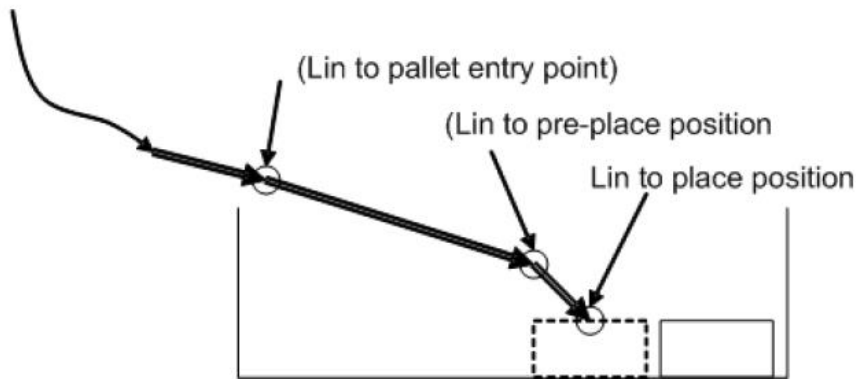
如下图所示，默认码垛的顺序从 X 开始。首先放置 x 方向上的所有部件，然后再开始放置下一行。在 x 方向上放置工件直达到达到设置好的 x 方向工件数。



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。

上述举例中默认的码垛顺序是 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，直到踩满。

进行放置时，建议路径如下图所示。



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）

### 6.10.3 PalletFromPut - 离开放置位置指令

#### 指令简介

放置工件完成后，机器人从放置位离开的指令。



#### 参数说明

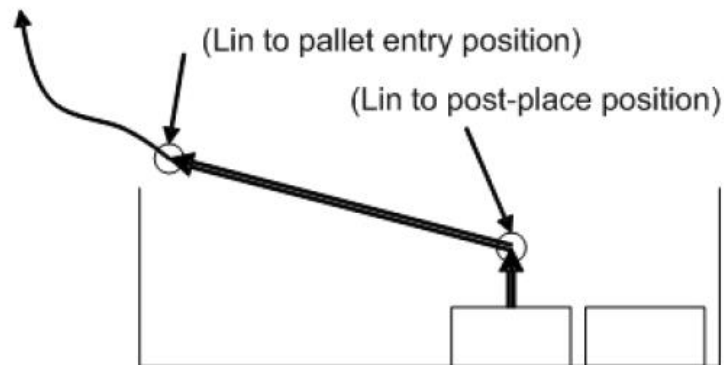
- 码垛名：当前码垛的 PALLET 变量。
- 目标速度：指令运行速度。

- 设置为“DEFAULT”，表示使用默认值 V4000。
- 设置为 SPEED 类型变量时，需要选用系统预定义。
- 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。

## 配置示例

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，建议路径如下图所示。



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。

## 6.10.4 PalletToGet - 抓取工件指令

### 指令简介

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。

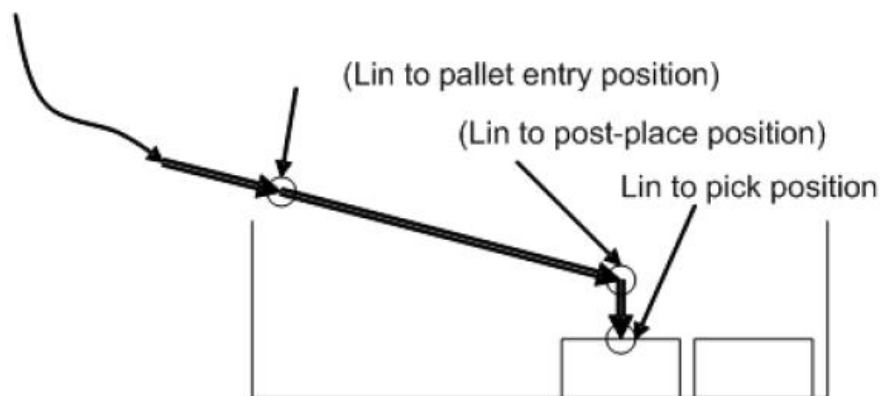
### 参数说明

- 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，需要选用系统预定义。
- 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。

### 配置示例

PalletToGet (PALLET3,V2000)

抓取过程中，路径的规划建议如下图所示。



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踩的进入位置（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。

## 6.10.5 PalletFromGet -离开抓取位置指令

### 指令简介

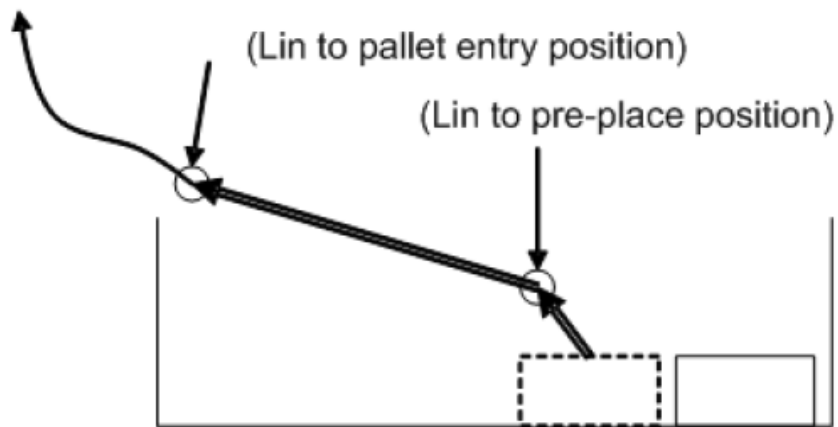
该指令是进行抓取工件动作后离开的过程，机器人执行该指令时，会从抓取工件的码垛点离开。

### 参数说明

- 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，需要选用系统预定义。
- 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。

### 配置示例

抓取过程中，路径的规划建议如下图所示。



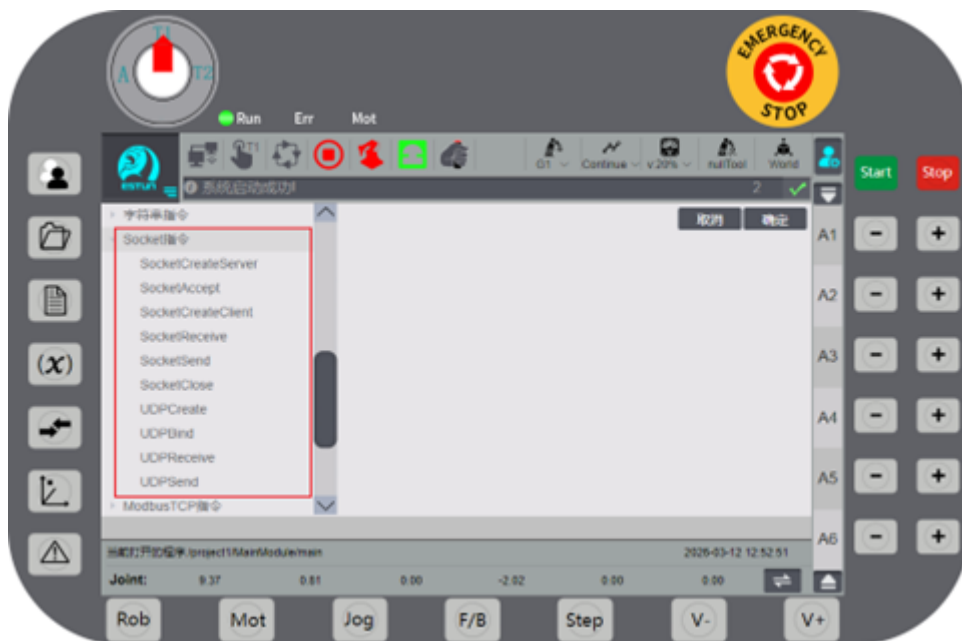
在调用 PalletFromGet 指令时，执行了以下运动指令：

- 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。

## 6.11 Socket指令

Socket 指令列表如图所示。



## 6.11.1 Socket指令样例说明

### 功能简介

Socket指令提供了一组TCP Socket和UDP Socket相关指令，可以将机器人控制器作为TCP服务器或者客户端进行TCP通信。

### 配置示例

- TCP服务器端示例（机器人做服务器）

```
function TCP_Server_Test()  
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)  
SocketCreateServer(G.TcpServer0,"172.31.16.100",9000,ReturnValue0)  
SocketAccept(G.TcpServer0,G.TcpSocket0,ReturnValue0)  
G.STRING0 = SocketReceive(G.TcpSocket0,10000,ReturnValue0)  
SocketSend(G.TcpSocket0,"123",ReturnValue0)  
SocketClose(G.TcpSocket0,ReturnValue0)  
SocketClose(G.TcpServer0,ReturnValue0)  
end
```

- TCP客户端示例（机器人作为客户端）

```
function TCP_Client_Test()  
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)  
SocketCreateClient(G.TcpSocket0,"172.31.16.200",9001,ReturnValue0)  
G.STRING0 = SocketReceive(G.TcpSocket0,10000,ReturnValue0)  
SocketSend(G.TcpSocket0,"123",ReturnValue0)  
SocketClose(G.TcpSocket0,ReturnValue0)  
end
```

- UDP示例（UDP不区分客户端和服务端）

```
function UDP_Test()  
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)  
UDPCreate(G.UdpSocket0,ReturnValue0)  
UDPBind(G.UdpSocket0,"172.31.16.100",9002,ReturnValue0)  
G.STRING0 = UDPReceive(G.UdpSocket0,10000,ReturnValue0)  
UDPSend(G.UdpSocket0,"172.31.16.200",9004,"123",ReturnValue0)  
SocketClose(G.UdpSocket0,ReturnValue0)  
end
```

## 6.11.2 SocketCreateServer - 创建 socket服务器指令

### 指令简介

创建TCP socket服务器，可以接收客户端连接并进行数据交互。该指令将机器人作为服务器使用。根据传入的参数，创建服务器。



- TcpServer: TcpServer变量。该指令会设置TcpServer变量的参数值，存放服务器信息。
- IP地址: 创建的服务器IP地址。
- 端口号: 创建的服务器端口号。
- 操作返回值: ReturnValue变量，返回值的int\_value参数为0表示成功，为1则为失败。

### 配置示例

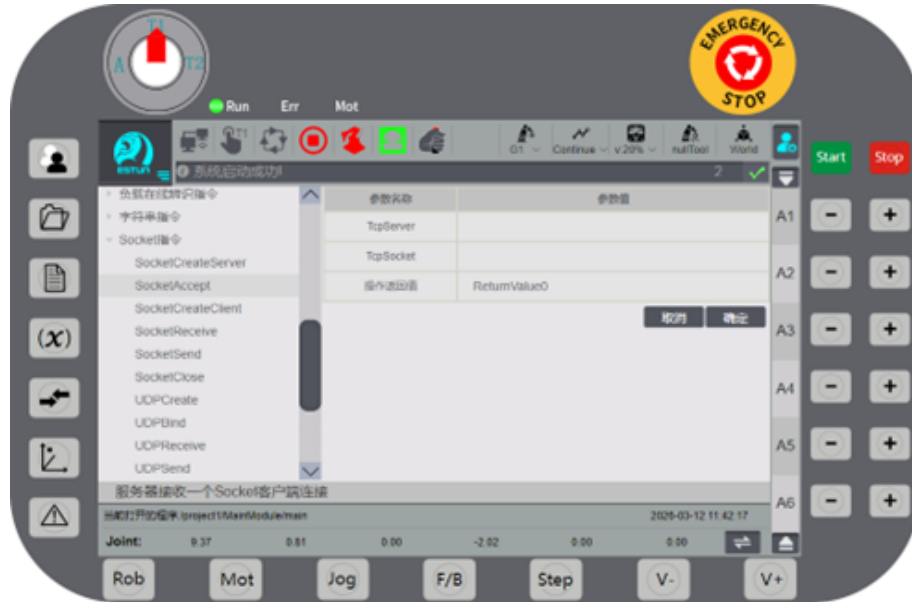
创建一个TCP Server，IP为“192.168.1.2”，端口为9000，操作返回值存放在ReturnValue0中。

```
SocketCreateServer(G.TcpServer0,"192.168.1.2",9000,ReturnValue0)
```

### 6.11.3 SocketAccept - 监听并接收TCP Socket客户端连接指令

#### 指令简介

TCP服务器监听并接收一个客户端连接，以便与客户端进行数据交互。该指令将机器人作为服务器。



- TcpServer: TcpServer变量。SocketCreateServer创建的变量，用来监听及接收客户端连接。
- TcpSocket: TcpSocket变量。用来存放接收的客户端Socket信息，后面可以通过该参数与客户端进行数据交互。
- 操作返回值: ReturnValue变量，返回值的int\_value为0表示成功，为1则为失败。

#### 配置示例

通过服务器G.TcpServer0监听并接收一个客户端连接，存放在G.TcpSocket0中，后续可以通过该标识接收客户端发来的数据，或者向客户端发送数据，操作返回值存放在ReturnValue0中。

```
SocketAccept(G.TcpServer0,G.TcpSocket0,ReturnValue0)
```

## 6.11.4 SocketCreateClient - 创建socket客户端指令

### 指令简介

创建TCP Socket客户端，可以与服务器进行数据交互。根据传入的服务器端参数，创建客户端并与服务器端建立连接。该指令将机器人作为客户端。



- TcpSocket: TcpSocket变量。存放TCP客户端连接信息
- 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：ReturnValue变量，返回值的int\_value为0表示成功，为1则为失败。

### 配置示例

创建一个TCP客户端并与服务器进行连接，Socket连接信息存放在G.TcpSocket0中，连接的服务器IP为“192.168.1.2”，端口为9000，操作返回值存放在ReturnValue0中。

```
SocketCreateClient(G.TcpSocket0,"192.168.1.2",9000,ReturnValue0)
```

## 6.11.5 SocketReceive - 接收Socket数据指令

### 指令简介

接收TCP服务器或客户端发来的字符串，并以String变量的形式存储。该指令可用于客户端接收服务器发送的数据，或服务器接收客户端发送的数据。



- TcpSocket: TcpSocket变量。SocketAccept或SocketCreateClient创建的TcpSocket变量。
- 超时时间: 等待服务器端或客户端发送数据的超时时间 (ms) , 超时则设置操作返回值的int\_value为1。
- 操作返回值: ReturnValue变量, 返回值的int\_value为0表示成功, 为1则为失败。
- 返回值: String变量, 用来存放接收的字符串数据。

## 配置示例

通过G.TcpSocket0接收TCP服务器或客户端发来的字符串数据, 并存入G.String0, 超时检查时间为1000ms, 操作返回值存放在ReturnValue0中。

```
G.String0 = SocketReceive(G.TcpSocket0,1000,ReturnValue0)
```

## 6.11.6 SocketSend - 通过 Socket 发送指令

### 指令简介

向服务器端或客户端发送字符串。该指令可用于客户端发送字符串数据给服务器, 或者服务器发送数据给客户端。



- TcpSocket: TcpSocket变量。SocketCreateClient或SocketAccept创建的TcpSocket变量。
- 字符串: String变量或常量。需要发送的字符串数据。
- 操作返回值: ReturnValue变量，返回值的int\_value为0表示成功，为1则为失败。

## 配置示例

通过G.TcpSocket0发送字符串“test string”，操作结果存放在ReturnValue0中。

```
SocketSend(G.TcpSocket0,"test string",ReturnValue0)
```

## 6.11.7 SocketClose - 关闭 socket指令

### 指令简介

关闭之前创建过的TCP Socket服务器，TCP Socket客户端或者UDP socket。



- Socket标识: TcpSocket变量, TcpServer变量或者UdpSocket变量。SocketAccept/SocketCreateClient创建的TcpSocket变量, 或者UDPCreate创建的UdpSocket变量。
- 操作返回值: ReturnValue变量, 返回值的int\_value为0表示成功, 为1则为失败

## 配置示例

关闭G.TcpServer0, 返回值存放在ReturnValue0中。

```
SocketClose(G.TcpServer0,ReturnValue0)
```

## 6.11.8 UDPCreate - 创建UDP Socket指令

### 指令简介

创建UDP指令, 用于UDP数据交互。



## 参数说明

- UdpSocket: UdpSocket变量, 该指令会设置UdpSocket变量, 存放UDP Socket信息。
- 操作返回值: ReturnValue变量, 返回值的int\_value为0表示成功, 为1则为失败。

## 6.11.9 UDPBind - 绑定UDP指令

### 指令简介

为UDP接收端绑定IP地址和端口号。如果机器人需要接收UDP对端发来的数据, 需要绑定IP地址和端口号, 当UDP对端向此IP地址和端口号发送数据, 机器人即可接收到数据。如果不需要接收UDP数据, 只是发送数据, 则不需要执行该指令。



- UdpSocket: UdpCreate创建的UdpSocket，用于绑定UDP接收端的IP地址和端口号。
- IP地址: 需要绑定的UDP接收端IP地址。
- 端口号: 需要绑定的UDP接收端端口号。
- 操作返回值: ReturnValue变量，返回值的int\_value为0表示成功，为1则为失败。

## 配置示例

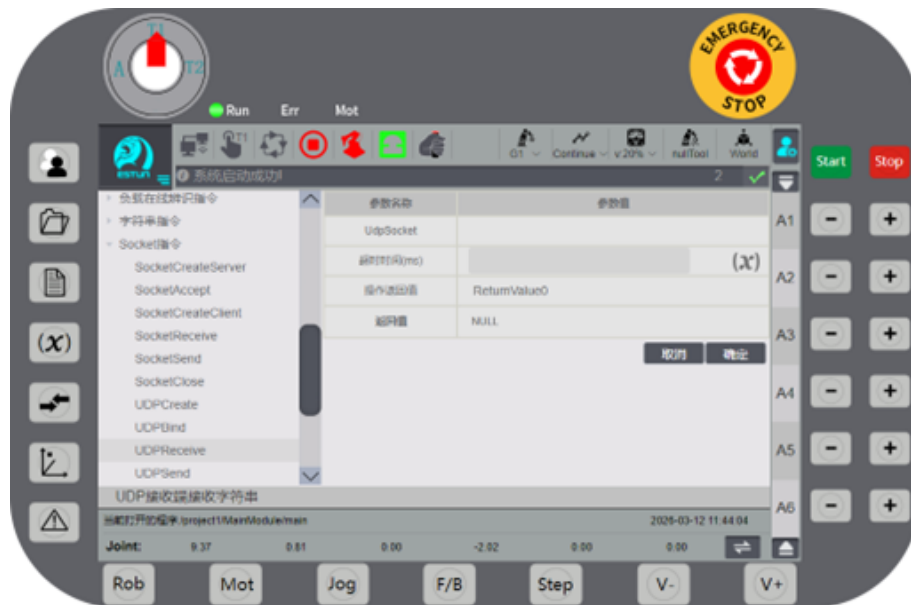
为G.UdpSocket0绑定IP地址和端口号，以便接收UDP对端发来的数据，操作返回值存放在ReturnValue0中。

```
UDPBind(G.UdpSocket0,"192.168.1.2",9001,ReturnValue0)
```

### 6.11.10 UDPReceive - 接收UDP指令

#### 指令简介

接收UDP对端发来的字符串数据。该指令执行前需要先执行UDPBind指令。



- UdpSocket: UDPBind指令执行之后的UdpSocket变量，用于接收对端发来的UDP数据。
- 超时时间: 等待UDP对端发送数据的超时时间（ms），超时则设置操作返回值的int\_value为1。
- 操作返回值: ReturnValue变量，返回值的int\_value为0表示成功，为1则为失败。
- 返回值: String变量, 用来存放接收的字符串数据。

## 配置示例

通过G.UdpSocket0接收UDP对端发来的字符串并存放在G.String0中，超时等待时间为1000，操作返回值存放在ReturnValue0中。

```
G.String0 = UDPReceive(G.UdpSocket0,1000,ReturnValue0)
```

### 6.11.11 UDPSend - 发送UDP指令

#### 指令简介

向UDP对端发送字符串数据。



- UdpSocket: UDPCreate创建的变量，用于向UDP对端发送数据。
- IP地址: UDP对端的IP地址。
- 端口号: UDP对端的端口号。
- 字符串: 需要发送的字符串数据, String变量或常量。
- 操作返回值: ReturnValue变量, 返回值的int\_value为0表示成功，为1则为失败。

## 配置示例

通过G.UdpSocket0向UDP对端（IP地址为192.16.1.3，端口号为9002）发送一个字符串“test string”，操作返回值存放在ReturnValue0中。

```
UDPSend(G.UdpSocket0,"192.168.1.3",9002,"test string",ReturnValue0)
```

## Socket指令使用示例

- TCP Socket服务器示例（机器人作为服务器）

```
function TCP_Server_Test()
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)
SocketCreateServer(G.TcpServer0,"172.31.16.100",9000,ReturnValue0)
SocketAccept(G.TcpServer0,G.TcpSocket0,ReturnValue0)
G.STRING0 = SocketReceive(G.TcpSocket0,10000,ReturnValue0)
SocketSend(G.TcpSocket0,"123",ReturnValue0)
SocketClose(G.TcpSocket0,ReturnValue0)
```

```
SocketClose(G.TcpServer0,ReturnValue0)
end\
```

- TCP Socket客户端示例（机器人作为客户端）

```
function TCP_Client_Test()
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)
SocketCreateClient(G.TcpSocket0,"172.31.16.200",9001,ReturnValue0)
G.STRING0 = SocketReceive(G.TcpSocket0,10000,ReturnValue0)
SocketSend(G.TcpSocket0,"123",ReturnValue0)
SocketClose(G.TcpSocket0,ReturnValue0)
end
```

- UDP Socket示例(UDP不区分客户端和服务端)

```
function UDP_Test()
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)
UDPCreate(G.UdpSocket0,ReturnValue0)
UDPBind(G.UdpSocket0,"172.31.16.100",9002,ReturnValue0)
G.STRING0 = UDPReceive(G.UdpSocket0,10000,ReturnValue0)
UDPSend(G.UdpSocket0,"172.31.16.200",9004,"123",ReturnValue0)
SocketClose(G.UdpSocket0,ReturnValue0)
end
```

## 6.12 ModbusTCP 指令

ModbusTCP 指令列表如图所示。



## 6.12.1 GetModConState - 获取与 ModbusTCP 通讯连接连接状态

### 指令简介

该指令用于获取机器人与外界使用 ModbusTCP 通讯的连接状态。



### 参数说明

- 返回值：返回当前的连接状态，类型为 BOOL，返回值为 0 表示成功，返回值为 1 则为失败。

## 6.12.2 ReadModbusReg - 读取指定 Modbus 寄存器值

### 指令简介

该指令用于读取指定 Modbus 寄存器的值。



### 参数说明

- 寄存器 ID：想要读取的寄存器 ID 号，类型 int。
- 返回值：返回值为 0 表示成功，返回值为 1 则为失败。

## 6.12.3 WriteModbusReg - 设置指定 Modbus 寄存器值

### 指令简介

该指令用于设置指定 Modbus 寄存器的值。

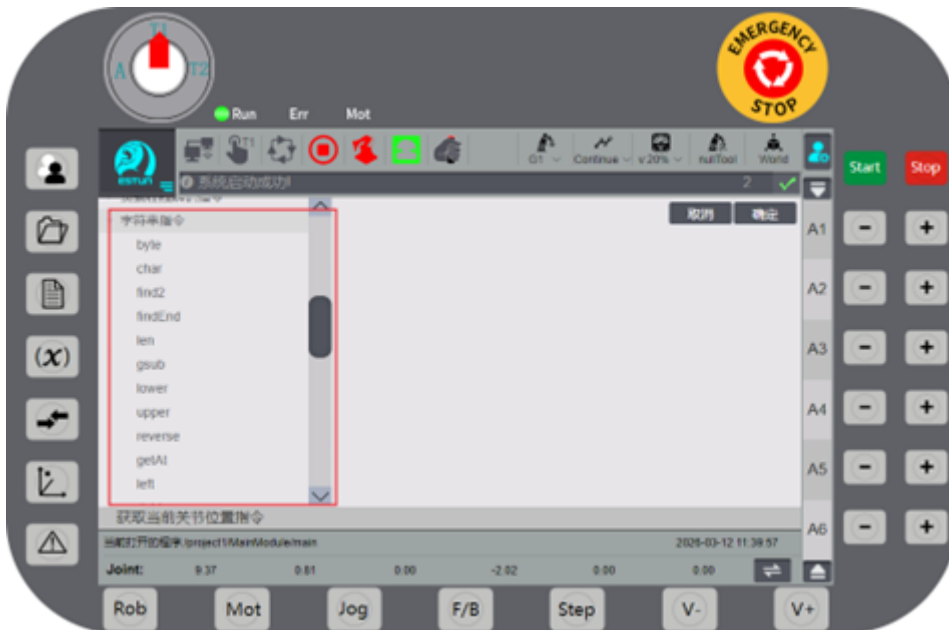


### 参数说明

- 寄存器 ID： 想要修改的寄存器 ID 号，类型 int。
- 寄存器值： 想要修改的寄存器值，类型 INT。

## 6.13 字符串指令

字符串指令列表如图所示。



### 6.13.1 byte

取字符串第n位的字符的ASCCII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。

### 6.13.2 char

返回ASCCII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。

### 6.13.3 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)

### 6.13.4 fineEnd

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。

### 6.13.5 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。

- 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。如执行format("%c", 83)，输出S。
  - %d, %i - 接受一个数字并将其转化为有符号的整数格式。如执行format("%+d", 17.0)，输出+17。执行format("%05d", 17)，输出00017。
  - %o - 接受一个数字并将其转化为八进制数格式。如执行format("%o", 17)，输出21。
  - %u - 接受一个数字并将其转化为无符号整数格式。如format("%u", 3.14)，输出3。
  - %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。如执行format("%x", 13)，输出d。
  - %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。如执行format("%X", 13)，输出D。
  - %f - 接受一个数字并将其转化为浮点数格式。如执行format("%6.3f", 13)，输出13.000。
  - %s - 接受一个字符串并按照给定的参数格式化该字符串。如执行format("%s", "monkey")，输出monkey。

---

### 说明

参数无限制，总长度只要不超过单个指令字符串长度即可。

---

## 6.13.6 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。

## 6.13.7 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。

### 6.13.8 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。

### 6.13.9 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。

### 6.13.10 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。

### 6.13.11 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。

### 6.13.12 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。

### 6.13.13 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。

### 6.13.14 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。

### 6.13.15 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。

### 6.13.16 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。

### 6.13.17 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。

### 6.13.18 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，Float型变量或常量。
- 函数返回值：转换后的字符串数据，String型变量。

### 6.13.19 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。

### 6.13.20 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，Float型变量。

### 6.13.21 StrToIntArray

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，Array{Int, 1}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。

如 `Int0 = StrToIntArray("1_2_3er", "_", 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。

### 6.13.22 StrToFloatArray

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, FloatOneArray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。

如`Int0 = StrToFloatArray("1.1_2.2_3er", "_", 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`。

### 6.13.23 IntArrayToStr

整型数组转字符串指令, 根据传入的整型数组、数组长度以及分割符, 返回以分隔符分隔的整型数组字符串。

- 参数1: 整型数组, `Array{Int, 1}`型变量。
- 参数2: 整型数组长度, Int变量或常量。
- 参数3: 分割符, String型变量或常量。
- 函数返回值: 成功把指定长度的整型数组元素, 以分隔符分隔保存在字符串中, String型变量。

如`String0 = IntArrayToStr(IntOneArray0, 3, "_")`, 该指令执行后系统按照传入的分割符'\_'来分割整型数组中的3个元素, 并用`String0`进行存储。

### 6.13.24 FloatArrayToStr

实数型数组转字符串指令, 根据传入的实数数组、数组长度以及分割符, 返回以分隔符分隔的实数数组字符串。

- 参数1: 实数数组, `Array{Float, 1}`型变量。
- 参数2: 实数数组长度, Int变量或常量。
- 参数3: 分割符, String型变量或常量。
- 函数返回值: 成功把指定长度的实数数组元素, 以分隔符分隔保存在字符串中, String型变量。

如`String0 = IntArrayToStr(FloatOneArray0, 3, "_")`, 该指令执行后系统按照传入的分割符'\_'来分割实数数组中的3个元素, 并用`String0`进行存储。

### 6.13.25 APosToStr

APos转字符串指令，根据传入的APos点位数据，返回转换后的字符串类型的点位数据。

- 参数1：待转换的APos点位数据，必须为APos类型变量。
- 函数返回值：返回转换后的点位数据，必须为String类型变量。

如String0 = APosToStr(APos0)，转换后的固定格式

为“0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;”第一个';'前面的0表示为APos点，其后跟的是16个轴的坐标值。最后一个';'表示分隔符。

### 6.13.26 CPosToStr

CPos转字符串指令，根据传入的CPos点位数据，返回转换后的字符串类型的点位数据。

- 参数1：待转换的CPos点位数据，必须为CPos类型变量。
- 函数返回值：返回转换后的点位数据，必须为String类型变量。

如String0 = CPosToStr(CPos0)，转换后的固定格式为“1;

0,0,0,0,0,0;0,0;0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;”第一个';'前面的1表示为Cpos点，第二个';'前面表示的是cfg参数。第三个';'前面表示的是工具坐标系id和用户坐标系id，其后跟的是16个轴的坐标值。最后一个';'表示分隔符。

### 6.13.27 TranStrToInt

字符串转一维整型数组指令，根据传入的字符串数据、分隔符以及一维整型数组，返回成功转换成整型数据的数量。

- 参数1：待分割的字符串数据，可为String类型的常量或者变量。
- 参数2：分隔符，可为String类型的常量或者变量。
- 参数3：存储字符串分割与转换后的整型数据，以一维数组的形式存储，必须为IntOne-Array类型变量。
- 函数返回值：返回成功转换成整型数据的数量。

如INT0 = TranStrToInt("1\_2\_3\_4","\_", IntOneArray0)，该指令执行后INT0的值为4，IntOneArray0中的前四位存储的数据分别为1,2,3,4。

### 6.13.28 TranStrToReal

字符串转一维实数数组指令，根据传入的字符串数据、分隔符以及一维实数数组，返回成功转换成实数数据的数量。

- 参数1：待分割的字符串数据，可为String类型的常量或者变量。
- 参数2：分隔符，可为String类型的常量或者变量。
- 参数3：存储字符串分割与转换后的实数数据，以一维数组的形式存储，必须为FloatOneArray类型变量。
- 函数返回值：返回成功转换成实数数据的数量。

如INT0=TranStrToReal("1.1\_2.2\_3.3\_4.4","\_",FloatOneArray0)，该指令执行后INT0的值为4，FloatOneArray0中的前四位存储的数据分别为1.1,2.2,3.3,4.4。

### 6.13.29 TranStrToAPos

字符串转APos指令，根据传入的字符串数据、分隔符以及APos点位数据，返回转换成功与否的状态。

- 参数1：待分割的字符串数据，可为String类型的常量或者变量。
- 参数2：分隔符，可为String类型的常量或者变量。
- 参数3：存储字符串分割与转换后的点位数据，以Apos点的形式存储，必须为Apos类型变量。
- 函数返回值：返回转换成功与否的状态。

如INT0=TranStrToApos("0;21,22,23,24,25,26&1231","&",P0)，该指令执行后INT0的值为1，P0中的前六位存储的数据分别为21,22,23,24,25,26。

---

#### 说明

传入的字符串格式必须与APosToStr格式一致。

---

### 6.13.30 TranStrToCPos

字符串转CPos指令，根据传入的字符串数据、分隔符以及CPos点位数据，返回转换成功与否的状态。

- 参数1：待分割的字符串数据，可为String类型的常量或者变量。
- 参数2：分隔符，可为String类型的常量或者变量。

- 参数3: 存储字符串分割与转换后的点位数据, 以Cpos点的形式存储, 必须为Cpos类型变量。
- 函数返回值: 返回转换成功与否的状态。

如INT0=TranStrToCpos("1;1,2,3,4,5,6,7;1,2;11,22,33,44,55,66",&,P1), 执行后INT0的值为1, P1中confdata的值分别为1,2,3,4,5,6,7。工具坐标系id为1, 用户坐标系id为2。x、y、z、a、b、c的值分别为11,22,33,44,55,66。

### 说明

传入的字符串格式必须与CPosToStr格式一致。

## 6.14 视觉指令

视觉指令列表如图所示。



### 6.14.1 TrigCam - 触发拍照指令

#### 指令简介

该指令用于发送触发字符串给相机, 用来触发相机拍照。



## 参数说明

- 相机索引：支持多台相机，0则表示默认的视觉界面设置连接参数匹配的相机，1-10则表示需要配合九点标定中连接参数使用的相机。
- 触发字符串：由用户约定的视觉检测触发字符串。

## 6.14.2 WaitFinishCam - 等待拍照完成指令

### 指令简介

该指令用于实现等待延迟，避免视觉回复数据机器人还未接收并解析完成。



## 参数说明

- 相机索引：支持多台相机，0则表示默认的视觉界面设置连接参数匹配的相机，1-10则表示需要配合九点标定中连接参数使用的相机。
- 等待时长：用户设置等待时长，该参数单位为ms。

### 6.14.3 GetCamPos - 获取相机纠偏数据指令

#### 指令简介

该指令用于获取并解析相机回复的数据，将协议数据解析后存储在变量字段里。



## 参数说明

- 相机索引：支持多台相机，0则表示默认的视觉界面设置连接参数匹配的相机，1-10则表示需要配合九点标定中连接参数使用的相机。
- 返回值：需要用户新建VisionRet类型变量，并在此处选择该参数。

## 6.14.4 SendMessage - 发送消息字符串指令

### 指令简介

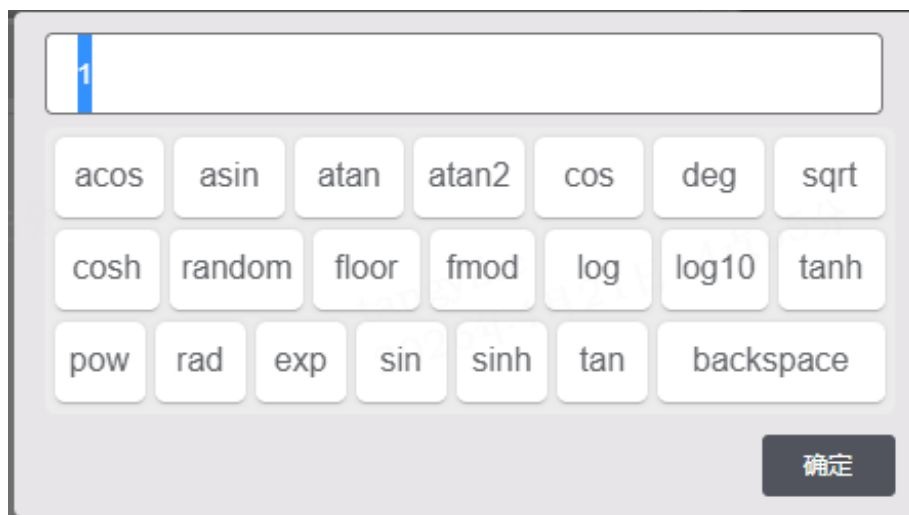
该指令用于按照指定格式发送消息字符串给相机设备。



## 参数说明

- 相机索引：支持多台相机，0则表示默认的视觉界面设置连接参数匹配的相机，1-10则表示需要配合九点标定中连接参数使用的相机。
- 消息字符串：用户设置字符串，指令执行时会根据指定格式将该字符串发出。

## 6.15 数学运算函数



### 6.15.1 sin

正弦三角函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.2 cos

余弦三角函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.3 tan

正切三角函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.4 asin

反正弦三角函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.5 acos

反余弦三角函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.6 atan

反正切三角函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.7 atan2

反正切三角函数，返回X轴到(x, y)点的弧度值。

参数1：Int或Float型变量或常量。

参数2：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.8 sinh

双曲正弦函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.9 cosh

双曲余弦函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.10 tanh

双曲正切函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.11 log

自然对数函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.12 log10

以10为底的对数函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.13 sqrt

开平方根函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.14 exp

以e为底的指数函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.15 pow

指数函数。

参数1：Int或Float型变量或常量，底数。

参数1：Int或Float型变量或常量，指数。

函数返回值：Float型结果。

### 6.15.16 deg

弧度转角度函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.17 rad

角度转弧度函数。

参数1：Int或Float型变量或常量。

函数返回值：Float型结果。

### 6.15.18 fmod

取余函数。

参数1：Float型变量或常量，被除数。

参数2：Float型变量或常量，除数。

函数返回值：Float型结果。

### 6.15.19 floor

向下取整函数。

参数1：Float型变量或常量。

函数返回值：Int型结果。

### 6.15.20 radom

取随机数函数。

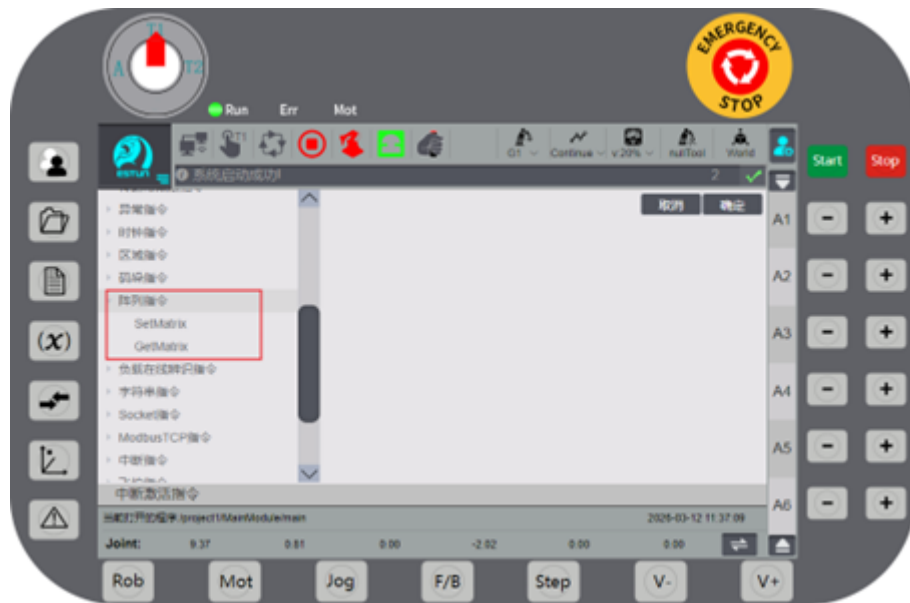
参数1：Int型变量或常量。

参数2：Int型变量或常量。

函数返回值：Int型结果。

## 6.16 阵列指令

阵列指令列表如图所示。



## 6.16.1 SetMatrix - 设置矩阵组

### 指令简介

该指令用于设置矩阵组，即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。



### 参数说明

- 原点：指定平行四边形的第一个点，也称为原点，CPOS类型。
- 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，CPOS类型。

- 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，CPOS类型。
- 行数：指生成矩阵的行数，范围为[1,32767]，Int常量或Int变量。
- 列数：指生成矩阵的列数，范围为[1,32767]，Int常量或Int变量。

## 6.16.2 GetMatrix - 获取矩阵组

### 指令简介

该指令用于获取矩阵组，取SetMatrix指令执行后对应行列的该点的值，赋值给目标点。SetMatrix中所述点的姿态和附加轴角度值保持与原点一致。

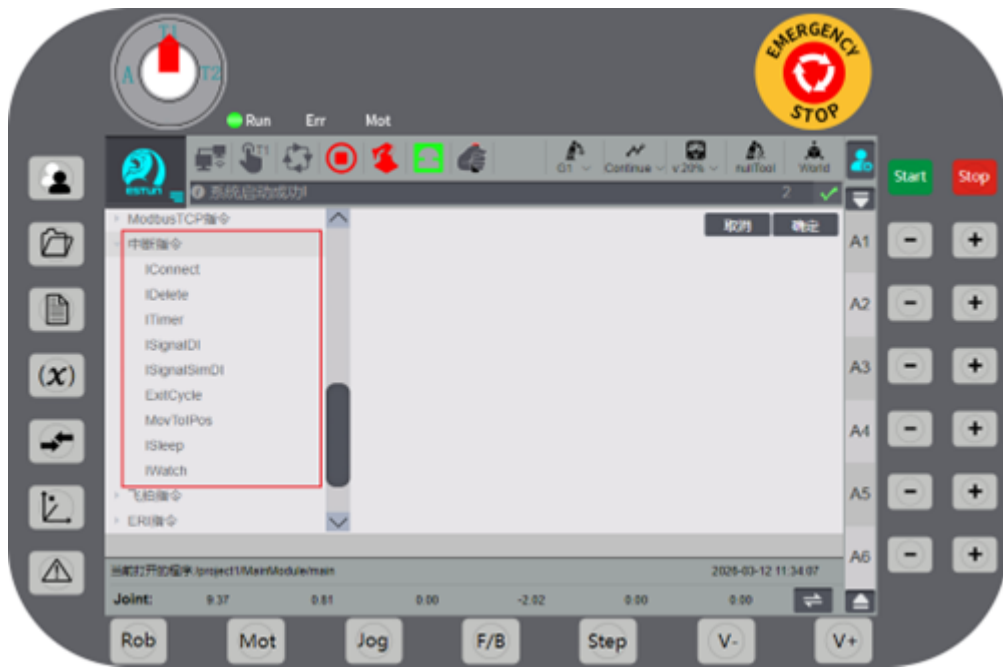


### 参数说明

- 行数：所要取点在矩阵中的行号，需要小于等于SetMatrix指令中对应设置行值，Int常量或Int变量。
- 列数：所要取点在矩阵中的列号，需要小于等于SetMatrix指令中对应设置列值，Int常量或Int变量。
- 目标点：用于保存取到的点值，CPOS类型。

## 6.17 中断指令

中断指令列表如图所示。



## 6.17.1 IConnect - 中断连接指令

### 指令简介

该指令用于将中断标识与中断处理程序进行绑定操作，以便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。



## 参数说明

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

## 6.17.2 IDelete - 中断解除指令

### 指令简介

该指令用于中断解除，将中断标识与相应的中断处理程序之间的连接去掉，移除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。



## 参数说明

- 中断标识：指中断索引号，INTERRUPT类型。

## 6.17.3 ITimer - 定时器中断指令

### 指令简介

该指令用于定时器中断，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。



## 参数说明

- 中断标识：指中断索引号，INTERRUPT类型。
- 触发时间：指定时检测时间，Int类型。

### 6.17.4 ISignalDI - 输入口中断指令

#### 指令简介

该指令用于监控设定DI的边沿信号有效状态来让系统执行中断程序，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。



## 参数说明

- 中断标识：指中断索引号，INTERRUPT类型。
- 端口号：指需要监控的DI端口号，Int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。
- 去重时间：当触发一次中断后，在指定时间内不再响应中断触发信号，Int类型。

### 6.17.5 ISignalSimDI - 虚拟输入口中断指令

#### 指令简介

该指令用于监控设定SimDI的边沿信号有效状态来让系统执行中断程序，虚拟输入中断指令，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。

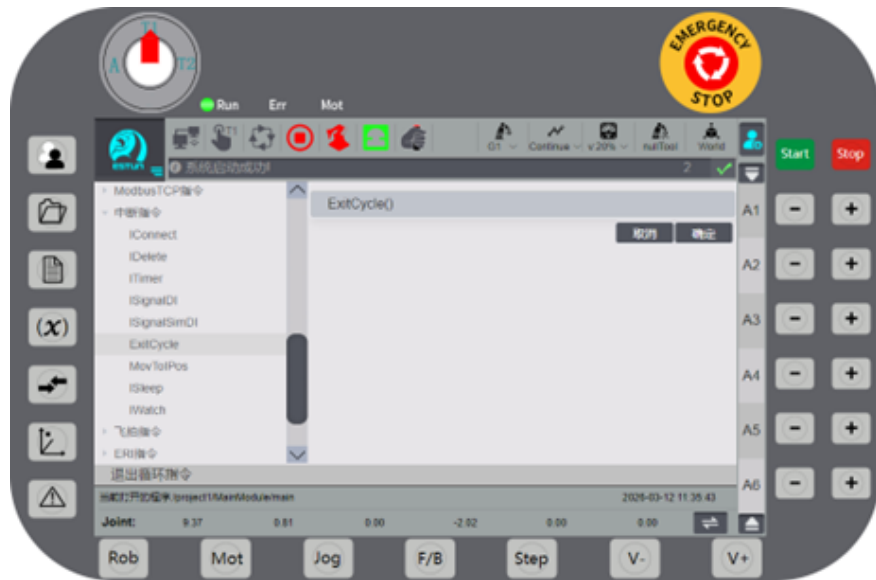


## 参数说明

- 中断标识：指中断索引号，INTERRUPT类型。
- 端口号：指需要监控的SimDI端口号，Int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。
- 去重时间：当触发一次中断后，在指定时间内不再响应中断触发信号，Int类型。

### 6.17.6 ExitCycle - 退出循环指令

该指令用于退出循环，执行该指令后系统会立即结束当前程序的自动运行，并返回到主程序的起始位置处，系统转为停止状态。



## 6.17.7 MovToIPos - 返回中断点指令

### 指令简介

该指令用于返回中断点，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。以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到该点，无需用户对该点进行额外的指令处理。



## 参数说明

- 运动方式：指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。

## 6.17.8 ISleep - 中断休眠指令

### 指令简介

该指令用于中断休眠，绑定中断标识变量，执行后可将指定中断状态置为休眠，此时不再响应中断触发信号。



### 参数说明

- 中断标识：指中断索引号，INTERRUPT类型。

## 6.17.9 IWatch - 中断激活指令

### 指令简介

该指令用于中断激活，绑定中断标识变量，执行后可将指定中断状态置为激活，此时正常响应中断触发信号。



## 参数说明

- 中断标识：指中断索引号，INTERRUPT类型

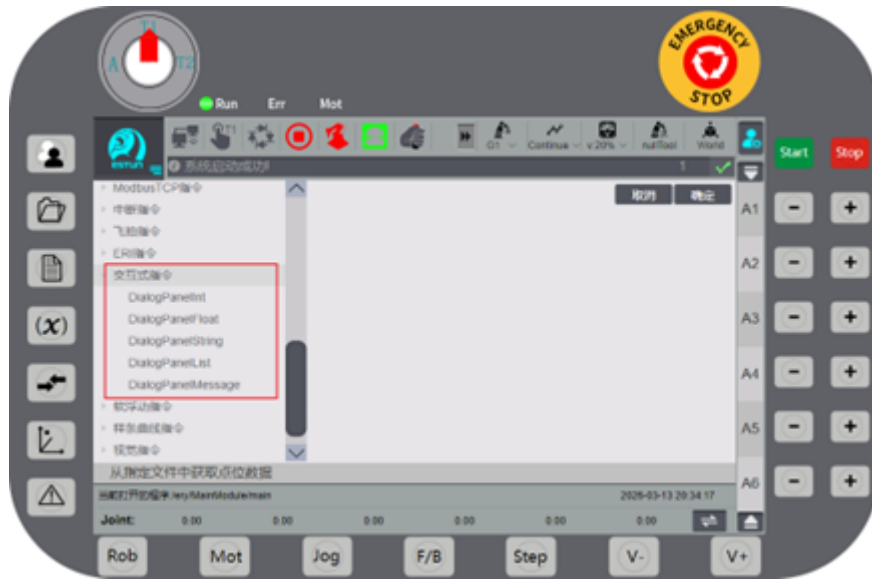
## 配置示例

prog1和prog2为2个中断程序，分别绑定了DI 10上升沿触发和DI 15下降沿触发，当DI信号满足时则会执行对应的中断程序。prog1执行完会返回到主程序被中断时刻的位置，然后继续运行主程序。prog2执行完中断程序后会跳转到主程序的开头，此时系统将转为停止状态。

```
function prog1()
  MovJ(G.p3)
  ...
  MovToIPos("MOVJ")
end
function prog2()
  MovJ(G.p4)
  ...
  ExitCycle()
end
function main()
  IDelete(G.INTERRUPT1)
  IDelete(G.INTERRUPT2)
  IConnect(G.INTERRUPT1, prog1)
  IConnect(G.INTERRUPT2, prog2)
  ISignalDI(G.INTERRUPT1,10,"RiseEdge",5000)
  ISignalDI(G.INTERRUPT2,15,"DownEdge",5000)
  label :
  MovJ(G.p0)
  MovJ(G.p1)
  MovJ(G.p2)
  GOTO label
End
```

## 6.18 交互式指令

交互指令列表如图所示。



### 6.18.1 DialogPanelInt - int型交互式指令

#### 指令简介

该指令用于在程序运行过程中弹出交互式对话框，用户可输入int型数据，所输入的数据将会赋值给预先设置的返回值类型变量的int\_value参数中。



## 参数说明

- 标题：指对话框的标题文字。
- 文本：指对话框内的提示信息文字。
- 返回值：指用于接收用户输入的返回值类型变量。

## 6.18.2 DialogPanelFloat - float型交互式指令

### 指令简介

该指令用于在程序运行过程中弹出交互式对话框，用户可输入float型数据，所输入的数据将会赋值给预先设置的返回值类型变量的float\_value参数中。



## 参数说明

- 标题：指对话框的标题文字。
- 文本：指对话框内的提示信息文字。
- 返回值：指用于接收用户输入的返回值类型变量。

## 6.18.3 DialogPanelString - string型交互式指令

### 指令简介

该指令用于在程序运行过程中弹出交互式对话框，用户可输入string型数据，所输入的数据将会赋值给预先设置的返回值类型变量的str\_value参数中。



## 参数说明

- 标题：指对话框的标题文字。
- 文本：指对话框内的提示信息文字。
- 返回值：指用于接收用户输入的返回值类型变量。

### 6.18.4 DialogPanelList - list型交互式指令

#### 指令简介

该指令用于在程序运行过程中弹出交互式对话框，用户可选择预先设置好的按钮，所选按钮的编号将会赋值给预先设置的返回值类型变量的int\_value参数中。



## 参数说明

- 标题：指对话框的标题文字。
- 文本：指对话框内的提示信息文字。
- 自定义选项1-10：指对话框中按钮的文字，10个选项中至少要设置一个。
- 返回值：指用于接收用户输入的返回值类型变量。

### 6.18.5 DialogPanelMessage - message型交互式指令

#### 指令简介

该指令用于在程序运行过程中弹出交互式对话框，用户可选择是或者否按钮，所选按钮的值将会赋值给预先设置的返回值类型变量的int\_value参数中，其中是返回1，否返回0。



## 参数说明

- 标题：指对话框的标题文字。
- 文本：指对话框内的提示信息文字。
- 返回值：指用于接收用户输入的返回值类型变量。

## 6.19 样条曲线指令

样条曲线指令列表如图所示。



## 6.19.1 GetPosFromFile

### 指令简介

GetPosFromFile，从指定文件中获取点位数据。

#### 说明

该指令执行后仅将文件中的数据，读取到点位变量内存，不会修改变量文件，断电重启后数据恢复。

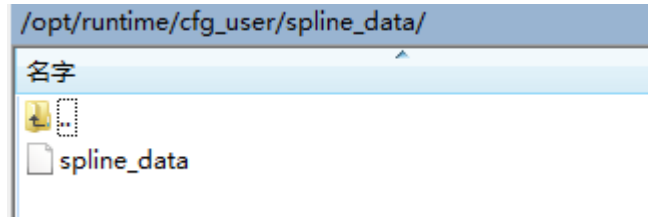


### 参数说明

- 文件名：要读取的文件名。
- 点位变量名称：支持APOS和CPOS，对应系统预留变量。
- 点位起始序号：从文件中读取的第一个数据存放的起始位置，如输入1，那么第一个点位数据会存在CPOS[1]中。
- 点位数量：要读取点位的数量。

## 限制及约束

- 点位文件需放在 /opt/runtime/cfg\_user/spline\_data 路径下。



- 文件内容格式有指定要求，读取Cpos时，文件内容如下图所示。  
依次表示：confdata数据，tool\_id，frame\_id，xyzabc数据，a7-a16的数据，其中a7-a16的数据可以缺省。

```
Cpos(Poscfg(0,0,0,0,0,0),0,0,1293.660000,-13.041000,810.899000,89.811000,-3.483000,177.509000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1201.404000,24.676000,712.600000,89.840000,0.134000,-176.567000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1179.982000,26.090000,613.518000,89.840000,0.134000,-176.567000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1174.071000,24.613000,614.017000,89.834000,0.117000,-176.773000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1168.170000,24.688000,614.202000,89.832000,0.113000,-176.841000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
```

如果有附加轴则增加对应数据即可，如下图所示。

```
Cpos(Poscfg(0,0,0,0,0,0),0,0,1293.660000,-13.041000,810.899000,89.811000,-3.483000,177.509000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1201.404000,24.676000,712.600000,89.840000,0.134000,-176.567000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1195.397000,24.458000,612.596000,89.840000,0.134000,-176.567000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1187.797000,26.291000,613.048000,89.840000,0.134000,-176.567000)
```

- 文件内容格式有指定要求，读取Apos时，文件内容如下图所示。  
依次表示：tool\_id，frame\_id，a1-a16数据，a7-a16的数据，其中附加轴的数据可以缺省，6轴机器人从第7位开始缺省，4轴机器人从第5位开始缺省。

```
Apos(0,0,1179.982000,26.090000,613.518000,89.840000,0.134000,-176.567000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Apos(0,0,1177.021000,24.614000,613.694000,89.840000,0.134000,-176.567000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Apos(0,0,1174.071000,24.613000,613.867000,89.836000,0.123000,-176.704000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Apos(0,0,1171.119000,24.613000,614.017000,89.834000,0.117000,-176.773000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
```

## 6.19.2 OnSplineIndex

### 指令简介

OnSplineIndex，样条曲线点位触发指令。

在不打断过渡的前提下，控制器可以在样条运动到中间指定某个控制点附近时触发执行一些操作，如赋值，设置IO等操作。机器人运动到设定的样条控制点附近，控制器会进入触发等待，延时触发时间后，开始执行相关的触发操作。



## 说明

触发指令默认为不支持单步模式。

## 参数说明

- 触发点序号：执行触发操作时对应点位在原点位数组中的序号。
- 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。

## 配置示例

在样条轨迹运行经过原数组中序号100点位，延时 500ms 执行 Do 操作。

```
OnSplineIndex(100,500) do S.DO[1]=1 end
MovS("CPOS",80,120,v=S.V50,b="RELATIVE",c=S.C0)
```

## 6.19.3 SetSplineParam

### 指令简介

SetSplineParam，设置样条曲线运动参数指令。

用于配置机器人样条运动规划相关参数。



## 参数说明

- 样条曲线运动模式：指定样条运动模式，当前仅有SMOOTH与ACCURATE两种模式，ACCURATE模式可保证经过中间控制点，但路径易发生扭曲变形。
  - SMOOTH：平滑模式，不强制经过样条中间控制点。
  - ACCURATE：精确模式，经过样条中间控制点。

## 6.19.4 MovS

### 指令简介

MovS，样条曲线运动指令。

用于控制机器人进行样条运动。



## 参数说明

- 点位变量名称：所选的点位变量所在点位数组名称。
- 点位起始序号：样条运动控制起始点位在所选点位数组中的序号。
- 点位终止序号：样条运动控制终止点位在所选点位数组中的序号
- 目标速度：指令运行速度。
  - DEFAULT：示使用默认值 V4000。
  - SPEED：需要选用系统预定义。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：精确到位。
  - 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。
- 工件负载：机器人执行该轨迹时使用的工件负载参数。
  - DEFAULT：表示沿用最近一次设置的工件负载参数。
  - PAYLOAD：需要选用系统预定义，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器全局倍率影响。
  - DEFAULT：同 GOVRON，即指令速度受示教器全局倍率影响。
  - GOVRON：指令速度受示教器全局倍率影响。
  - GOVROFF：指令速度不受示教器全局倍率影响。
- 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。

## 7 高级功能

### 7.1 运动


#### 7.1.1 碰撞检测

##### 7.1.1.1 碰撞检测

#### 功能简介

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。

#### 操作步骤

1. 点击左上角，选择“应用 -> 运动 -> 碰撞检测”，进入“碰撞检测”界面。



2. 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”。



### 说明

- 碰撞检测功能基于负载信息及设备信息来检测碰撞，必须正确设定负载信息以避免误报或影响检测灵敏度。
- 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。

### 7.1.1.2 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。

### 说明

使用该功能需联系厂家。

## 7.1.2 振动抑制

### 功能简介

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，振动也会更强烈。

### 操作步骤

#### 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见机器人设置界面和“[SetPayload](#)”。

1. 在界面菜单选择“应用 -> 运动 -> 振动抑制”，进入配置界面。



- 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。默认为标准强度。  
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。
- 自学习振动抑制：通过采集机器人振动轨迹或者振动点位的振动数据进行自学习，生成匹配的振动抑制功能参数，改善机器人振动。

### 7.1.2.1 自学习振动抑制

#### 功能简介

当机器人振动且无法通过“振动抑制”功能界面的“振动抑制等级”修改来改善振动时。可以通过“自学习振动抑制”功能，采集机器人振动轨迹或者振动点位的振动数据进行自学习，生成匹配的振动抑制功能参数，改善机器人振动，功能效果包括本体轴和附加轴。

#### 说明

“自学习振动抑制”界面相关操作需要机器人处于停止状态（当机器人处于运行状态时，各按键灰显，无法操作）。

#### 操作步骤

1. 在菜单选择“应用 -> 运动 -> 振动抑制 -> 自学习振动抑制”进入配置界面。



2. 振动数据采集。点击“开始采集”，进行振动数据的采集。当机器人处于振动数据记录状态时，机器人会进入振动模式（自动运行和示教模式下的振动会加剧）。

3. 数据处理。选择需要存储的自学习结果的参数 ID，点击“**学习**”，自学习结果将会存储在选择的 ID 中。
4. 结果应用。选择“步骤二”中的自学习ID进行使用，“NONE”标识表示不使用自学习结果。

## 参数说明

- 开始采集：将重新进行振动数据的采集。当机器人处于振动数据记录状态时，机器人会进入振动模式（自动运行和示教模式下的振动会加剧）。
- 暂停采集：开始采集之后，用户可通过“**暂停采集**”按键中断数据采集。
- 继续采集：暂停采集之后，用户可通过“**继续采集**”按键恢复数据采集。
- 结束采集：数据采集完成后，用户可通过“**结束采集**”按钮，结束数据采集。
- 参数ID：取值范围 1~6；用于存储自学习结果的 ID。
- 应用参数 ID：取值范围“NONE、1~6”。表示需要使用自学习的“ID”，“NONE”表示不使用自学习。

## 7.1.3 场景适配调试

### 功能简介

为避免用户实际使用中无法充分利用机器人性能，通过设置模式记录场景适配的参数，在对应场景调用来增加性能的利用率。

---

#### 说明

- 调用模式仅能通过开机时默认调用模式，以及SetMotionMode指令切换模式，程序注销、加载时模式不会复位。
  - 自动调节功能执行结束后会自动恢复为自动调节功能开启前的模式，调用调节得到的模式需要通过指令设置。
  - 低速高精模式用于提升路径精度，减少路径变形量，高速运行可能出现抖动，设置有200mm/s的路径速度上限。
  - 指令中可调用小线段模式DENSE\_POINT，用于点位密集的情况，可降低运动速度波动，若用于高速且运动幅度大的程序可能引发抖动。
  - 若自定义模式开启高增益开关，SetMotionMode指令调用时会修改系统伺服，请确认伺服参数是否适用于当前轨迹，当前速度，若想恢复初始PN参数，需通过SetMotion-Mode指令切换到标准模式。
  - 若未开启高增益开关，用户模式不显示PN参数选项。
-

## 操作步骤

在菜单选择“应用 -> 运动 -> 场景适配”，进入配置界面。



- 默认开机启动：开机时默认调用的模式。
- 高增益模式开关：自定义模式是否使用自定义的PN参数。
- 关节加速度系数：影响关节运动规划的加速度。
- 关节加加速度系数：影响关节运动规划的加加速度。
- 关节加速度限制系数：影响所有运动规划中对关节加速度的限制。
- 关节加加速度限制系数：影响所有运动规划中对关节加加速度的限制。
- 关节力矩系数：影响所有运动规划中对关节力矩的限制。
- TCP路径加速度系数：影响笛卡尔空间运动规划的路径加速度。
- TCP路径加加速度系数：影响笛卡尔空间运动规划的路径加加速度。
- TCP姿态加速度系数：影响笛卡尔空间运动规划的姿态加速度。
- TCP姿态加加速度系数：影响笛卡尔空间运动规划的姿态加加速度。
- PN102：速度环增益，仅在高增益模式开关使能后显示。
- PN103：位置环积分，仅在高增益模式开关使能后显示。
- PN103：位置环增益，仅在高增益模式开关使能后显示。
- 备注：为自定义模式设置的备注信息。

## 自定义模式

自定义模式可手动设置模式参数，也可使用自动功能生成程序对应的模式参数。

- 手动模式

1. 点击“手动”，进入手动设置界面。



2. 点击“确定”提交数据。

- 自动设置

1. 点击“自动”，进入自动调节界面。



2. 选择是否“从低速开始”，点击“开始调参”。

从低速开始选项可使机器人自动从较低的参数开始调节，否则从当前参数开始调节。



3. 加载目标程序，并在程序中添加标记指令AutuTune(ADP)。该指令会打断过渡，建议添加在非过渡位置。
4. 执行程序。等待进度条达到100%，也可中途手动点击stop停止机器人运动以提前结束。



5. 点击“保存”，将数据保存至对应的自定义模式。
6. 在目标程序中移除标记指令AutuTune。

## 7.1.4 软浮动功能

### 功能简介

目前软浮动支持三种模式：关节软浮动、笛卡尔软浮动、推出软浮动。

在使用软浮动功能之前要对软浮动相关参数进行配置。

### 参数说明

在示教器界面点击“应用 -> 运动 -> 软浮动设置”，进入软浮动设置界面。



- 关节软浮动参数配置说明
  - 激活状态：关节号前的小正方形可以通过选中控制机器人的关节处于激活状态，激活的关节在受到外力时会产生规避。
  - 阻尼和刚度：控制机器人运动的参数，可能会导致飞车（用户不要自行调整，如需调整请联系我司技术服务人员）。
  - MAX：控制机器人关节在正向浮动的最大角度。
  - MIN：控制机器人关节在负方向浮动的最小角度。

- 编号：指参数的序号，可以设置8组：JointCfg1-JointCfg8。
- 灵敏度：推动机器人外力的大小，设置越高，越容易推动机器人（同时也可能导致撤去外力机器人停不下来）。
- 笛卡尔软浮动参数配置
  - 激活状态：轴号前的小正方形可以通过选中控制机器人的轴处于激活状态，激活的轴在受到外力时会产生规避。
  - 阻尼和刚度：控制机器人运动的参数，可能会导致飞车（用户不要自行调整，如需调整请联系我司技术服务人员）。
  - MAX：控制机器人轴在正向浮动的最大范围/角度。
  - MIN：控制机器人轴在负方向浮动的最小范围/角度。
  - 编号：参数的序号，可以设置8组：CartCfg1-CartCfg8。
  - 灵敏度：推动机器人外力的大小，设置越高，越容易推动机器人（同时也可能导致撤去外力机器人停不下来）。
- 推出软浮动参数配置
  - 编号：参数的序号，可以设置8组：PushCfg1-PushCfg8。
  - 灵敏度：指推动机器人外力的大小，设置越高，越容易推动机器人。
  - 坐标轴：指推出浮动的方向（目前只支持X/Y方向）。

## 相关指令介绍

- SoftFloatStart指令  
在程序界面选择“新建 -> 软浮动指令 -> SoftFloatStart”。

- 软浮动类型：可以根据需求选择，包括AXIS(关节)、CART(笛卡尔)、PUSH(推出)三种模式。



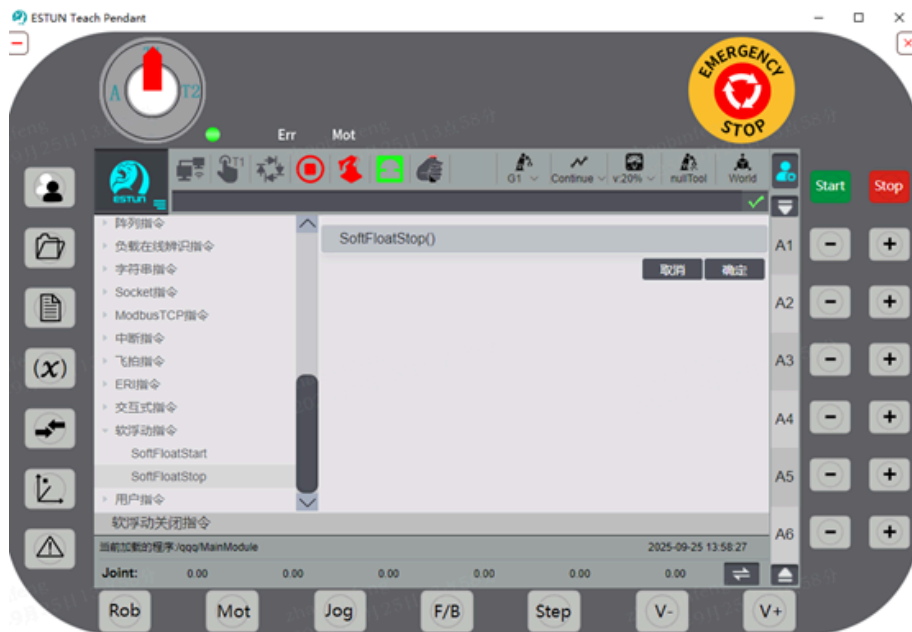
- 关节软浮动：只要设置软浮动ID（配置的参数ID）即可。



- 笛卡尔软浮动：需设置软浮动ID，以及参考坐标系（目前支持世界坐标系、工具坐标系和用户坐标系）。



- 推出软浮动：需设置软浮动ID，以及参考坐标系（目前支持世界坐标系、工具坐标系和用户坐标系）。
- SoftFloatStop指令  
软浮动结束指令没有参数，用于结束软浮动，推出软浮动模式。且软浮动开始和结束指令需成对出现，有软浮动开始指令时必须添加软浮动结束指令，并且执行开始指令后要执行软浮动结束指令，否则会报错。



## 配置示例

机器人运动到P1点后，开启关节软浮动功能，机器可推动机器人运动，30s后结束软浮动。



## 7.2 区域监控

### 7.2.1 多类型区域监控

#### 功能简介


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。

## 操作步骤

1. 点击左上角，选择“应用 -> 区域监控”，进入区域监控的变量修改界面。默认情况下，标准区域变量为“未激活”状态。



2. 点击“编辑”进入参数编辑界面。



3. 配置基本设置参数。配置完成后点击“下一步”。



4. 配置点位标定参数。配置完成后点击“下一步”。



5. 配置IO绑定参数。配置完成后点击“完成”。



6. 当监控端选择为Box或者Sphere时，可设置监控端参数。



7. 修改包络数据。请依据表 7-1 参数说明修改该界面的参数。



8. 点击“完成”，然后点击“激活”，完成标准区域变量的设定。


表 7-1 参数说明

参数名称	参数说明
区域名	显示标准区域变量名称。 默认无法修改。
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。

表 7-1 参数说明 (续)

参数名称	参数说明
类型	选择该区域的类型： <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	选择区域监控点： <ul style="list-style-type: none"> <li>• 法兰端 (Flange)</li> <li>• 机器人工具末端 (Tcp)</li> <li>• (Box)</li> <li>• (Sphere)</li> </ul>
参考坐标系	选择该区域的参考坐标系。 默认是世界坐标系
形状	选择区域的形状： <ul style="list-style-type: none"> <li>• 盒状 (BOX)</li> <li>• 圆柱 (CYLINDER)</li> <li>• 轴干涉区 (JOINT)</li> <li>• 轴组干涉区域: (JointGroup)</li> <li>• 多边形干涉区: (Polyhedron)</li> </ul>
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”。
X、Y、Z、lenX、lenY、lenZ	设定区域的起点坐标以及区域的大小。 起点坐标是相对于参考坐标系。 设定盒状或圆柱区域的起点坐标以及区域的大小。起点坐标是相对于参考坐标系
轴、posMax、posMin	设定轴干涉区参数，即检测的轴号、轴运动的最大和最小范围。   <b>说明</b> <ul style="list-style-type: none"> <li>• 轴为当前机型中的所有轴，包含本体轴和附加轴。</li> <li>• 可监控的本体轴和附加轴的个数不超过 10 个。</li> </ul>

表 7-1 参数说明 (续)

参数名称	参数说明
添加、移除	<p>点击“添加”按钮，该轴设置的范围会同步到右侧轴组范围内。点击“移除”按钮，该轴设置的范围会清零。</p> <hr/> <p> <b>说明</b></p> <p>此参数在形状为 JointGroup 时可操作。</p> <hr/>
轴组范围	<p>显示各轴的限制范围[0,0]表示该轴未设置范围。</p> <hr/> <p> <b>说明</b></p> <p>此参数在形状为 JointGroup 时可显示。</p> <hr/>

### 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。
7. 对于配置为焊钳的附加轴，焊钳在耦合状态时参与检测；焊钳处于解耦状态时不参与检测。
8. 若某一个轴在 JointGroup 与 Joint 类型的区域中均有设置，则监控同时生效。

## 7.3 多工位预约

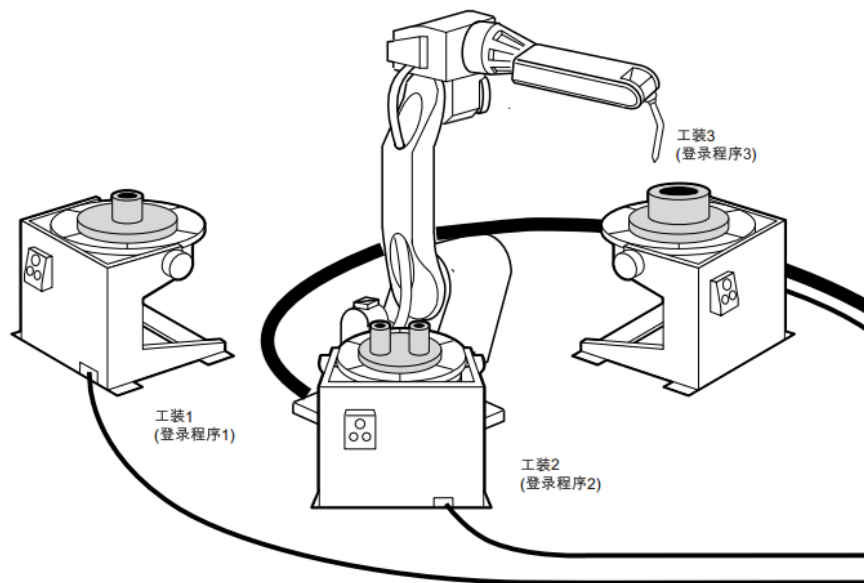
### 功能简介

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口进行绑定，同时赋予这些输入口启动与停止的功能，其后用户可以通过给该输入口有效信号，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。

下图为机器人的作业示例：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。

机器人作业时，用户可通过示教器查看各工位上工作状态和 IO 信号。



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。

## 操作步骤

### 说明

为了正确使用该功能，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[程序编辑](#)。
- 每个工位的工装上的启动按钮能够正常使用。

1. 在主界面菜单上点击“应用 -> 多工位预约”。
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”。



3. 点击各“工位”所在行的“程序名”。
4. 在弹出的“工程列表”中选择想要进行的程序。
  - 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。
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。

---

 **说明**

统一 IO 端口，不要重复使用。

---

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。



---

 **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。

---

## 7.4 附加轴配置


1. 选择“应用 -> 附加轴配置”可对“附加轴”和“PLC轴”进行配置，并在该插件中对导轨轴进行标定。



### 2. 附加轴设置方式

- a. 根据实际情况设置附加轴“轴数”，如“1”。



- b. 点击  图标，进入设置界面，根据实际情况修改对应参数，点击“保存”后，根据界面提示重启控制器系统，重启完成后设置生效。



若需配置成双驱轴，进入“双驱参数”中设置。



### 📖 说明

- 进入附加轴参数配置页面，选择焊钳轴为OFF，显示所有附加轴参数。
- 选择焊钳轴为ON，只显示实虚轴和焊钳参数，其余参数在工艺包页面配置。
- 双驱参数中单双控模式为即时生效，仅修改此参数时无需重启。

3. PLC轴-设置方法：与附加轴同理。

#### 4. 导轨轴标定方法

导轨轴标定功能，在示教器上导轨标定界面，通过该界面可对导轨轴进行标定操作，标定的值即时生效，无需重启，提高调试效率。



a. 当前所选附加轴为 Robot 轴时，点击“**导轨轴标定**”，进入标定页面。

机器人坐标转换：标定机器人和导轨的转换关系。

机器人位置转换微调：微调导轨的精度。

### 📖 说明

当前附加轴不是第一个导轨附加轴时，不可标定。



b. 点击“开始标定”，示教第一个点位，点击“下一步”。



c. 示教第二个点位，点击“下一步”。



d. 计算标定结果。



- e. 点击“确认保存”，弹出提示框询问是否保存结果，选“确定”则保存本次标定结果，选“取消”则放弃标定结果。



### 说明

- 按步骤依次示教点位，能够正确获取导轨点位信息，正确计算出标定结果。
- 通过“确认保存”或“取消标定”可对标定结果进行保存或放弃，标定结果保存数据正常，重启控制器标定精度不丢失。
- 详细参数介绍和设置方式，请参见《附加轴使用说明书》。

## 7.5 伺服管理

### 7.5.1 伺服参数设置

#### 功能简介

用户可通过示教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，请确认轴驱动器标号（1 轴 ~ 6 轴）对应了机器人的各关节。

## 操作步骤

1. 在菜单选择“应用 -> 伺服管理”。



2. 设置 Pn 参数。

进入“设置 Pn 参数”操作界面，初始默认显示 1 轴的 Pn 参数。在数字框选择需要设置的参数，点击旁边的设置按键即可设置 Pn 参数。



### 说明

进行设置 Pn 参数操作之前请确保：

- 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。

Pn参数	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
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。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。

Pn参数	名称	范围	单位	说明
Pn115	扭矩前馈滤波	0 ~ 640	0.25ms	减缓扭矩前馈引起的机械冲击。

## 7.5.2 伺服参数开机自检

### 功能简介

用户更换驱动器或控制器时，要保证控制器与驱动器参数与上一次是一致的，当两者参数未能恢复到上次状态时要始终报警，不允许动机器人，因为会产生危险动作。为了预防此情况，机器人应该开机时自动检测伺服和电机ID的情况，来提醒用户进行比较和替换参数文件等操作。用户断开控制柜电源，更换机器人本体轴的伺服或者电机后，重启控制柜。示教器上会有报警提示，电机或伺服ID不一致报警。开机自检支持的功能如下：

- 用户能够进行参数文件比较。
- 用户能够进行参数选择下载更新。

### 说明

开机自检功能仅支持C1控制柜和S2控制柜。

### 操作步骤

1. 选择“应用 -> 伺服管理”，在右侧菜单栏中选择“比较与更新”，进入对应的功能页面。



2. 通过右上角开关，可控制开机自检功能的开启与关闭。

- 勾选轴后点击“参数比较”按钮，控制器利用FOE上传伺服的参数文件，与需要对比的参数文件进行比较，界面上如有黄色边框轴表示ID不一致。  
比较完成后，绿色边框说明此轴的伺服参数文件与需要比对的文件一致，无需修改。若为红色边框，则文件不一致。



- 用户想要更新参数文件，需要勾选对应轴，然后点击“参数更新”，进行参数文件的更新。



### 7.5.3 力矩监控功能

该功能包含对力矩和电流的监控。用户可以对力矩、平均负载率、电流、电流历史最大值进行实时监控。

用户在左上角选择“应用 -> 伺服监控”，在右下角选择“力矩监控”进入该功能，可以通过左侧标签选择查看“力矩”或者“电流”。



图 7-1 力矩监控



图 7-2 电流监控

## 7.6 传送带跟随

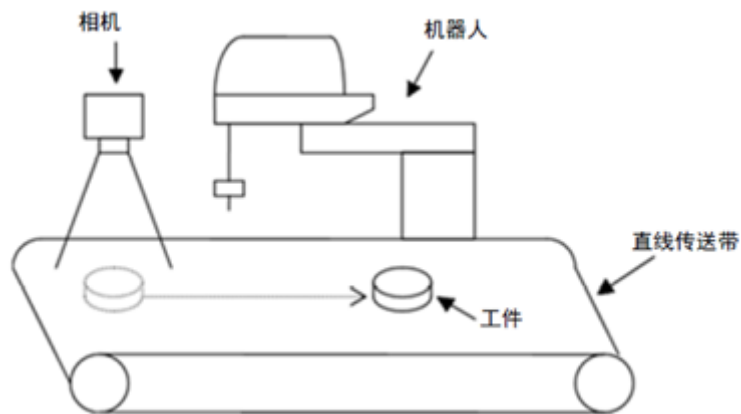
### 功能简介

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。

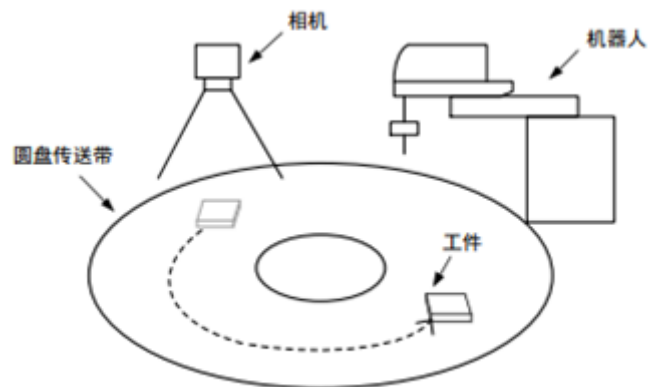
#### 说明

当前系统版本最多支持单机双传送带应用。

目前最多支持 4 个传送带。



直线传送带跟踪系统



圆盘传送带跟踪系统

## 操作步骤

1. 在菜单栏中，点击“应用 -> 传送带跟随”进入传送带配置界面。  
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](#)、[编码器参数](#)、[边界参数](#)、[跟随参数](#)、[触发参数](#)、[多机传送带标定](#)、[视觉参数](#)七个配置界面进一步配置。



- 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。
- 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。

### 7.6.1 传送带参数配置

选择“编辑参数”进入传送带参数页签。



- 类型：直线型或圆盘型。
- 动坐标系：与该传送带绑定的动坐标系。
- 编码器通道号：配置传送带对应的编码器模块通道号。
- 多机使能：多机传送带功能开启/关闭。
- 主从选择：选择当前机器人作为多机传送带功能的主机或者从机。

### 📖 说明

只有开启多机使能后才有该选项。

- 虚拟传送带使能：控制虚拟传送带开启/关闭。
- 虚拟传送带状态：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。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，范围0-2000。

## 7.6.2 编码器参数配置

选择“编码器参数”，该界面用于标定脉冲当量，若使用虚拟传送带，则无需标定脉冲当量。设置完所有参数后，需点击“保存”。



- 动坐标系：控制当前参考坐标系是否为动坐标系。
- 方向：设置传送带的方向。
- 脉冲当量：表示编码器脉冲值与传送带位置之间的转换关系。由标定起始点、终止点的位置和编码器脉冲值共同计算得出。

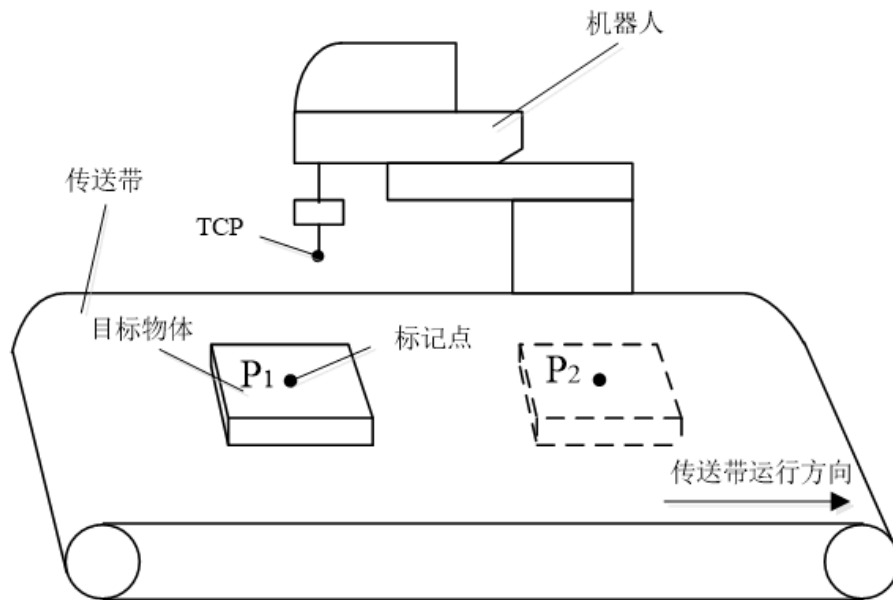
### 📖 说明

直线传送带设置方向时，动坐标系X正方向与传送带运动方向相同，传送带运动时位置越变越大。

圆盘传送带设置方向时，动坐标系Z正方向与传送带旋转轴正方向（使用右手法则判定）相同，传送带运动时位置越变越大。

#### 7.6.2.1 直线传送带脉冲当量标定

直线传送带时脉冲当量的单位为pulse/mm。



1. 根据当前的真实工具设置工具坐标系（建议使用针尖标定工具）。
2. 点击“开启”动坐标系，将机器人的参考坐标系切换为与当前传送带绑定的动坐标系。
3. 静止传送带，在机器人臂展范围内选择一个标记点 $P_1$ ，点动机器人使TCP与 $P_1$ 重合，点击“标定”记录当前位置。
4. 点动机器人至安全位置，开启传送带，使其向前运动一段距离后停止，此时标记点位于 $P_2$ 处，点动机器人使TCP与点 $P_2$ 重合，点击“标定”记录当前位置。



5. 点击“计算”，系统将自动计算并显示脉冲当量值。点击“保存”完成脉冲当量的标定。

### 7.6.2.2 圆盘传送带脉冲当量标定

圆盘传送带时脉冲当量的单位为pulse/deg。圆盘传送带脉冲当量的标定方法与直线传送带类似，区别在于直线传送带时标记点 $P_1$ 和 $P_2$ 由直线运动生成，而圆盘传送带时标记点 $P_1$ 和 $P_2$ 由旋转运动生成。



### 7.6.2.3 脉冲当量验证

标定完成后，验证标定值是否正确，可在传送带停止时记录当前跟随配置界面上显示的传送带位置Pos1，然后开启传送带使其向前移动一段距离（圆盘传送带时为旋转角度）后停止，再次记录传送带位置Pos2，如果计算值  $(Pos2 - Pos1) \approx$  测量值（可使用尺子测量），即表示脉冲当量标定正确。

### 7.6.3 边界参数配置

选择“边界参数”，设置所有参数后，点击“保存”。界面中各参数说明如下。



- 最小边界：机器人开始跟随的最小边界位置，若物料在最小边界外，机器人将一直处于等待跟随状态，直到物料进入最小边界位置内，机器人才会启动跟随（执行MovLSync/MovJSync指令）。
- 最大边界：机器人最晚启动跟随位置，物料在最小边界和最晚启动位置之间才会启动跟随（执行MovLSync指令），若物料超过最晚启动位置，机器人将放弃跟随。
- 最晚同步位置：机器人最晚跟随物料的位置，机器人最晚与传送带上的物料保持同步运动的位置，即最晚执行完MovLSync指令位置。
- 最晚同步点：机器人与物料保持同步的最大边界位置，机器人超过最大边界放弃目标不再跟随的位置，即最晚开始执行MovLSyncQuit指令的位置。

#### 📖 说明

边界参数基于所标定的动坐标系，必须结合机器人、工件和动坐标系的相对位置关系设置合理的数值。

### 7.6.4 跟随参数配置

选择“跟随参数”，进入传送带参数配置界面。



- 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。

### 7.6.5 触发参数配置

选择“触发参数”，进入触发参数配置界面。



- 触发类型：
  - 定时触发：系统会根据配置的触发时间(ms)，周期设置触发输出端口输出高电平信号，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），设置触发输出端口输出高电平信号，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，需要创建一个并行任务程序：检测输入信号、下发目标信息等模拟视觉下发的位置点。可使用并行任务，编写指令检测触发输入端口有高电平信号时，调用SetTargetPos指令发送工件位置信息。
  - 主机触发：选择主机触发后，从机的视觉位置点由主机下发，无需检测外部信号。

---

### 说明

当传送带参数配置页面中选择开启多机使能，并且主从选择为从机时，该选项才可选。

---

- 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。

## 7.6.6 多机传送带标定

### 功能简介

此功能是标定传送带上的从机机器人与主机机器人之间的位置关系。标定前需保证所有的机器人网络已正常连接，且标定过程必须先完成主机的标定才可标定其他的从机。

### 说明

若机器人不重新安装，一般不需要重新标定这个参数。

### 标定步骤

1. 选择“多机传送带标定”，进入多机传送带标定界面。



2. 在主机器人的可达空间内固定一个标定纸，传送带的运动方向作为标定纸坐标系的X轴正方向。
3. 点动机器人末端移动到标定纸的第一个点，点击示教获取可以得到第一个点机器人的位置以及传送带的位置。此点作为标定纸上坐标系的原点。

4. 沿着传送带运动的方向点动机器人，然后点击示教获取第二个点。此点是标定纸上坐标系X轴正方向上的一个点。
5. 点动机器人在XOY平面上运动到任意位置，点击示教获取第三个点（主机和从机一定要确保运动的方向是一致的）。此点是标定纸上坐标系XOY平面上的任一点。主机只需要示教点位，示教的点位会自动发送给从机，无需计算标定结果。
6. 开启传送带，使标定纸运动到从机器人的可达空间内。
7. 从机器人重复步骤2~4，获取标定纸相对于从机器人的位置关系，然后点击计算得到当前从机器人相对于主机的位置关系。每个从机器人都重复这个操作即可完成标定。

## 7.6.7 视觉参数配置

### 功能简介

选择“视觉参数”，进入视觉参数配置界面。



- 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。
- 相机数据类型：2D 或 3D。
- 去重误差：当视觉拍照频率过快，检测到的目标物体可能重复，系统会根据误差范围自动剔除重复的目标。
- 视觉点类型：可设置相机发送的点位是基于动坐标系或世界坐标系。

- 分配方式：该下拉项只有开启多机使能并且主从选择为主机时才可见。  
当前版本只支持视觉指定的分配方式，由视觉设备指定目标物体该由哪个ID号的机器人进行抓取。此参数对应视觉点位中的ATTR。
- 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。
- 视觉标定：实现外部视觉设备与机器人之间的坐标转换，将视觉设备发送的坐标转换为机器人工具坐标系下的坐标。



## 操作步骤

1. 标定板放在传送带上，相机进行拍照，并在界面上点击传送带拍照位置后的取当前位置按钮，记录传送带的位置。
2. 启动传送带，将标定板移动到机器人的工作空间内，停止传送带，在界面上的传送带示教位置后点击“取当前位置”，记录传送带的位置。
3. 依次移动机器人到标定板的九个点上进行标定，需要切换到与动坐标系相同的用户坐标系进行示教，按照右上角图示示教对应点位，点击标定计算可得到步骤1中视觉相机拍照时的对应机器人动坐标系下的点位。
4. 将标定计算后的点位输入到视觉端。
5. 查看示教点可返回查看步骤2中的示教点位。

## 说明

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《iER系列传送带功能使用手册 iEPM01V30403-01》。

## 7.7 负载辨识

### 功能简介

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。

- 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()

### 说明

在负载辨识过程中，应禁止其他操作。

机器人上下限位置、辨识点数在限制范围内。

### 开启负载辨识功能

1. 点击菜单，选择“配置 -> 控制器设置”进入控制器设置界面，可开启负载辨识使能开关。



## 数据采集

1. 数据采集。点击程序下方“新建”，选择“负载在线识别指令 -> StartMdlLog”，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。

- 采集“Robot”类型数据用例

```
StartMdlLog("Robot")
SetOverRide(10)
PayloadPath()
-
SetOverRide(100)
PayloadPath()
StopMdlLog()
```

- 采集“WithLoad”类型数据用例

```
StartMdlLog("WithLoad")
SetOverRide(10)
PayloadPath()
-
SetOverRide(100)
PayloadPath()
StopMdlLog()
```

- 采集“ValidData”类型数据用例

```
StartMdlLog("ValidData")
MovL(P1)
MovJ(P2)
MovC(P3,P4)
StopMdlLog{}
```

2. 获取负载辨识所用轨迹程序

```
MovJ(P0)
MovJ(P1)
MovJ(P2)
MovJ(P3)
MovJ(P4)
MovJ(P5)
```

MovJ(P6)  
 MovJ(P7)  
 MovJ(P8)  
 MovJ(P9)  
 MovJ(P8)  
 MovJ(P7)  
 MovJ(P6)  
 MovJ(P5)  
 MovJ(P4)  
 MovJ(P3)  
 MovJ(P2)  
 MovJ(P1)  
 MovJ(P0)

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。

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

## 负载辨识计算

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。

```
CalcDynMdl("TypePayload",S.PAYLOAD[1],1,"Disable")
```

1. 点击程序右侧“新建”，选择“负载再现辨识指令 -> CalDynMdl”，点击“确认”。
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“S.PAYLOAD1”，“数据间隔长度”设置为“1”，“是否模型验证”设置

为“Disable”（设置为“Enable”时，即需要模型验证，StartMdlLog() 应选择Valid-Data 类型采集一组模型验证数据）后，点击“确认”。



3. 在线辨识指令执行过程时间较长（一般用时30min以内），待辨识计算结束，将弹窗显示辨识结果。

## 7.8 多机协同功能

### 功能概述

该功能提供一种可以支持多个机器人进行协同运动的机制。

## 机构设置

1. 选择“配置 -> 机构设置”，进入“机构设置”界面。



- 机构组：机构组的名称。
  - 机构：机构组的本体，包含robot或者robot带变位机。
  - 附加轴：机构组中所包含的附加轴
2. 点击“编辑”，进入“编辑”界面。可选本体机构，变位机ID。若本体机构带附加轴，可勾选附加轴是否启用。



## 多机设置

1. 选择“应用 -> 多机协同”，进入多机设置界面，包含基本设置和协同设置。
  - 基本设置：包含多机协同使能参数和机器人个数。



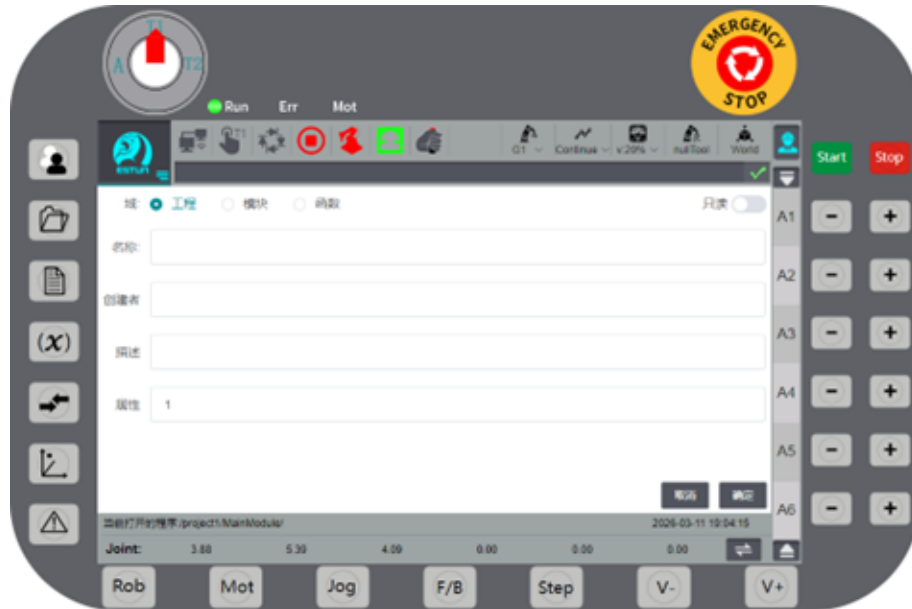
- 使能：是否开启多机功能。
  - 机器人个数：当前协同运动的机器人个数。
- 协同设置：包含协同对、主动组、从动组。



- 协同对：总的存在6个，用于配置哪些机构组进行协同运动。
- 主动组：协同运动的主机器人。
- 从动组：协同运动的从机器人。

## 工程创建

创建一个多机的工程和模块，属性设置为多个架构组。



## 多机标定

1. 选择“**标定 -> 多机标定**”进入标定页面。
2. 基本设置可以设置主机器人和从机器人，点击“**协同关系**”表格，可用于手动编辑协同关系。



3. 依次示教4个不同的位置点。



4. 点击“计算”得到计算结果。



5. 点击“保存”存储数据。

## 7.9 制动测试功能

### 功能简介

制动测试总体分为主动测试和被动测试两种。主动测试可以通过用户在示教器上手动执行，也可由用户通过外部PLC调用。被动测试在机器人进行正常生产任务时实时检测（手动和自动运行均可触发）。

## 主动测试

1. 点击示教器“应用 -> 制动测试”，进行参数配置和开始主动测试。



2. 若用户第一次使用该功能，未进行参数配置，则无法进入测试界面，需要先点击“设置”进入配置界面完成参数配置。
3. 配置界面分为“参数设置”和“轨迹设置”两个部分。



- 参数设置

---

 **说明**

“被动检测使能”参数修改即生效，其余参数修改后用户请点击“保存”按钮以生效。

- 
- 被动检测使能：只用于被动测试，设置被动检测的开启与关闭。
  - 灵敏度：只用于被动测试，灵敏度越低，机器人在急停过程中越容易触发失效报警。
  - 测试周期：用于设置单次测试之间的间隔时间，单位为小时(h)，若周期设置为0，则表示主动测试功能关闭。
  - 制动测试状态输出IO：用于绑定测试周期到达时输出的IO信号端口，支持DO和SimDO。
  - 选择轴：表示执行主动测试时需要测试的轴。
- 轨迹设置。主要包括三个点的选取或示教，分别定义为home点、safe点、park点。

---

 **说明**

三个点的配置界面均有“移动到点”按钮，用于将机器人移动至该点（需要调到手动模式）。此外，用户可根据左侧图标了解机器人当前位置是否处于选取或示教的点位附近。

- 
- home点：通常为用户自己创建的点，可通过选择栏进行选取，并且限制变量域只能为全局变量。
  - safe点：为执行主动测试时所在的位置点，创建或修改时需要用户手动将机器人移动至目标安全位置进行示教。
  - park点：为执行主动测试且存在制动失效时所移动到的位置点，创建或修改时同样需要用户手动将机器人移动至目标安全位置进行示教。

4. 配置完参数和轨迹点后，点击“测试”可进入测试界面。

---

 **说明**

该界面中的“移动至Home点”按钮与轨迹设置中home点的“移动到点”按钮功能一致，同样需要在手动模式下执行。

---

点击“移动至Home点”，移动之前，会有弹窗提示，关闭之后，继续长按“移动至Home点”，直到提示成功，点击“下一步”。



5. 点击“移动到Safe点”，移动之前，会有弹窗提示，关闭之后，继续长按“移动到Safe点”，直到提示成功，点击“下一步”。



6. 若当前不是手动模式或者当前机器人不处于safe点位置时，点击“开始测试”会进行报警提示。

若满足上述条件后点击“开始测试”，界面会弹框让用户再次确认从当前点运动到park点的过程中为安全的。用户点击确认后，机器人会移动到safe点并进入到最终开始执行测试的界面。



7. 开始测试后，机器人会在safe点位置对所有勾选轴逐个进行测试并输出单个轴的测试结果，当所有轴都测试完成后则会进行弹窗显示最终结果。若没有轴存在制动失效，机器人会自动回到home点，若有轴存在制动失效，机器人则会自动移动到park点。
8. 如果存在轴制动失效的情况，会锁定所有程序的自动运行，用户只能进行点动操作。只有用户更换完抱闸并再次执行制动测试且无制动失效后，锁定才会解除。

## 被动测试

- 机器人在高速运行过程中拍急停、或者较高速度下进行手动jog时突然松开励磁均可能触发被动测试失效报警，触发难易程度与灵敏度有关。
- 当被动测试触发抱闸状态异常后，该提示会一直显示，且必须在执行主动测试且所有轴均无异常情况后才能消除。

### ⚠ 注意

- 机器人在执行制动测试时，若非紧急情况，尽量避免按STOP按钮、切换模式、拍急停等异常操作。
- 机器人在执行制动测试时，尽量不要中途修改机器人运行速度，可能会影响测试结果。
- 被动测试灵敏度参数在未与厂家确认情况下请勿轻易修改。
- 只有ED3L的伺服支持制动测试功能。

## 7.10 双机防撞功能

### 功能简介

针对双机或多机现场机器人的安全性能，需要实时检测机器人之间的距离，不仅为机器人提供安全保护，同时提高双机或者多机工作效率。

### 说明

双机防撞功能目前仅支持六轴机器人。

### 操作步骤

1. 点击示教器“应用 -> 双机防撞”。



防撞使能开关：控制防撞功能的开启与关闭。当使能为关闭状态时，禁止用户界面相关操作。


碰撞组使能设置/查看页：对碰撞组信息进行查看以及使能设置。

机器人包络数据查看/编辑设置页：对机器人包络数据进行查看及编辑。


工具包络数据查看/编辑设置页：对工具包络数据进行查看及编辑。

2. 编辑机器人包络数据



用户点击机器人包络数据列表中的编辑按钮 ，可进入对应行序号的机器人包络数据编辑界面。在编辑页面对机器人包络进行相关参数设置，包括包络使能状态、形状、关节、半径及预警距离。

### 3. 编辑工具包络数据

用户点击机器人包络数据列表中的编辑按钮 ，可进入对应行序号的机器人工具包络数据编辑界面。

#### 说明

工具包络数据中关节与包络形状已默认指定（关节为A6，形状为球体），不可编辑。另外，工具包络仅支持单组数据使能，不支持多组数据同时使能。机器人包络数据无此限制。

#### 注意

- 设置机器人本体或工具包络并使能后，实时检测碰撞组内的两台机器人包络体之间的距离，当大于安全距离时，机器人之间无碰撞风险，可正常运行。
- 当小于安全距离时，有提示信息，此时机器人仍可运动，或由PLC根据此信号确定是否运动机器人。
- 当小于包络体之间的半径时，认为机器人发生碰撞，立即停止，有相应报警信息。

## 7.11 臂上负载

### 功能简介

当机器人轴1~轴3上安装负载时，可设置相应的负载参数，保证前馈输出准确，控制稳定。

#### 说明

该功能仅在通用六轴机器人上支持。

### 操作步骤

点击“应用 -> 臂上负载”，设置对应轴上的臂上负载参数。

#### 说明

请合理设置参数，如果设置值超过了允许范围，会报错提示，本次设置无效。



参数	单位	含义
m	kg	负载质量
$M_x (R_x)$	mm	负载质心相对于参考坐标系X方向上的位置
$M_y (R_y)$	mm	负载质心相对于参考坐标系Y方向上的位置
$M_z (R_z)$	mm	负载质心相对于参考坐标系Z方向上的位置
$I_{xx}$	$kg \cdot mm^2$	负载在重心位置的参考坐标系X方向回转的惯量矩
$I_{yy}$	$kg \cdot mm^2$	负载在重心位置的参考坐标系Y方向回转的惯量矩
$I_{zz}$	$kg \cdot mm^2$	负载在重心位置的参考坐标系Z方向回转的惯量矩

**⚠ 注意**

- 设置的臂上负载参数一定要和实际值相匹配，如果两者误差过大，会对机器人的节拍、精度产生严重影响，并触发碰撞报警。
- 使用该功能后，会影响（末端）负载辨识的精，请在未使用该功能且未安装臂上负载前进行末端负载辨识。

## 7.12 标定

## 7.12.1 工件坐标系

### 功能简介

工件坐标系(Workpiece Coordinate System)是相对于工件建立的坐标系，用于定义加工或操作对象的位置和姿态。

### 操作步骤

1. 选择“标定 -> 工件标定”。
2. 当变量未标定，工件坐标系会包含（new），表示未标定，此时系统中也无法选择到该变量，该页面中的“关联机构”和“关联坐标系”表示工件坐标系变量中存储的关联机构和关联坐标系，不可选择。



3. 点击“设置”，可配置用于标定的关联坐标系和关联机构，支持通过状态栏的机构和坐标系栏切换。



4. 选择相应的坐标系名称与标定方法，点击“下一步”。




5. 记录原点的坐标：移动机器人至用户坐标系的原点，点击“记录点”，界面将记录原点（第一个点）的坐标。
6. 点击X，移动机器人至用户坐标系的 X 轴上的任意位置处，点击“记录点”记录第二个点的坐标。
- 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。
7. 重复上述步骤记录第三个点，点击“计算”查看计算结果。

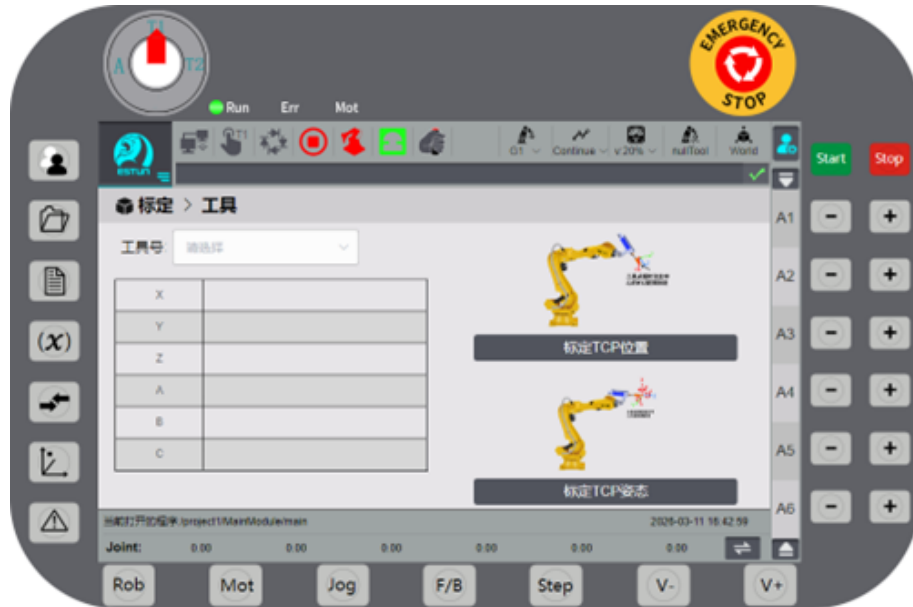


8. 确认结果无误后点击“保存”。此时工件坐标系不带 (new)，可在系统中找到该变量。



## 7.12.2 工具坐标系标定

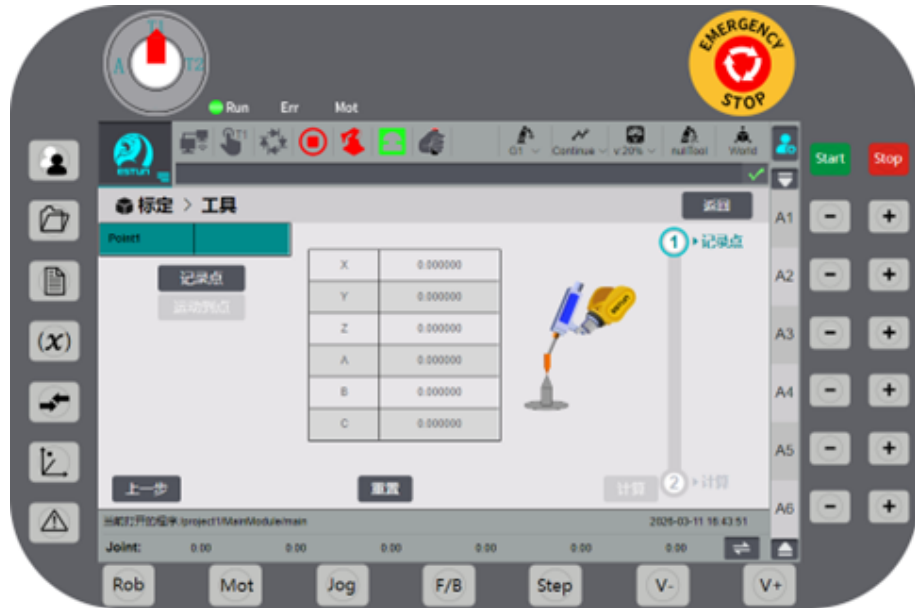
1. 在用户界面点击左上角进入菜单界面，选择“标定 -> 工具标定”进入工具坐标系标定界面。



2. 选择标定的对象，标定TCP位置和标定TCP姿态。
3. 标定TCP姿态。包括三点标定和一点标定。
  - “三点标定法”（姿态），通过示教三个点位完成工具坐标系的标定。当标定方法选择三点标定法时，点击右下角“设置”。



- 当选择标定方法为One Point（一点标定法）时，点击右下角“设置”。



将机器人末端方向与 Z 轴方向平行，点击“记录点”，再点击右下角“计算”，显示标定工具坐标系姿态的结果。确认无误后，点击“确认”，完成修改。

4. 标定TCP位置。示教同一个点的多种不同姿态的方式来标定工具偏移。
  - a. 移动机器人 TCP 至某个方位，点击“记录点”，可增加点及删除点。

---

#### 📖 说明

最多不超过20个点，最低不少于4个。

---



- b. 用同样的方法完成剩余点的记录，此时再点击右下角计算得到标定的值。确认无误后点击“确认”完成修改。

### 7.12.3 用户坐标系标定

#### 📖 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。

1. 点击左上角菜单栏，选择“标定 -> 坐标系标定”。



2. 选择坐标系名称并点击右下角“设置”，进入设置界面。



3. 选择相应的坐标系名称与标定方法，然后点击“下一步”。



4. 开始记录原点的坐标，移动机器人至用户坐标系的原点。点击“记录点”，界面将记录原点（第一个点）的坐标。
5. 然后点击X，移动机器人至用户坐标系的 X 轴上的任意位置处，点击“记录点”界面将记录第二个点的坐标。
  - 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。

- 用同样的方法记录第三个点，然后点击“计算”查看计算结果。



- 确认结果无误后点击“保存”即可。

## 7.12.4 变位机标定

- 在系统菜单栏上选择“标定 -> 变位机标定”。



- 选择需要设定的“变位机 ID”，如选择变位机 ID=“1”。
- 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。

### 📖 说明

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。

4. 选择“附加轴”，然后点击“下一步”。



### 📖 说明

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。

5. 选择变位机的机型，如选择“型号1”。

### 📖 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。

6. 点击“下一步”，将对变位机进行三点示教。
  - a. 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。
  - b. 移动机器人TCP至该标志位置，然后点击“示教”。

- c. 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**”。
  - d. 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**”。
  - e. 点击“**下一步**”，将显示标定“**结果**”。
7. 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。

---

8. 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。

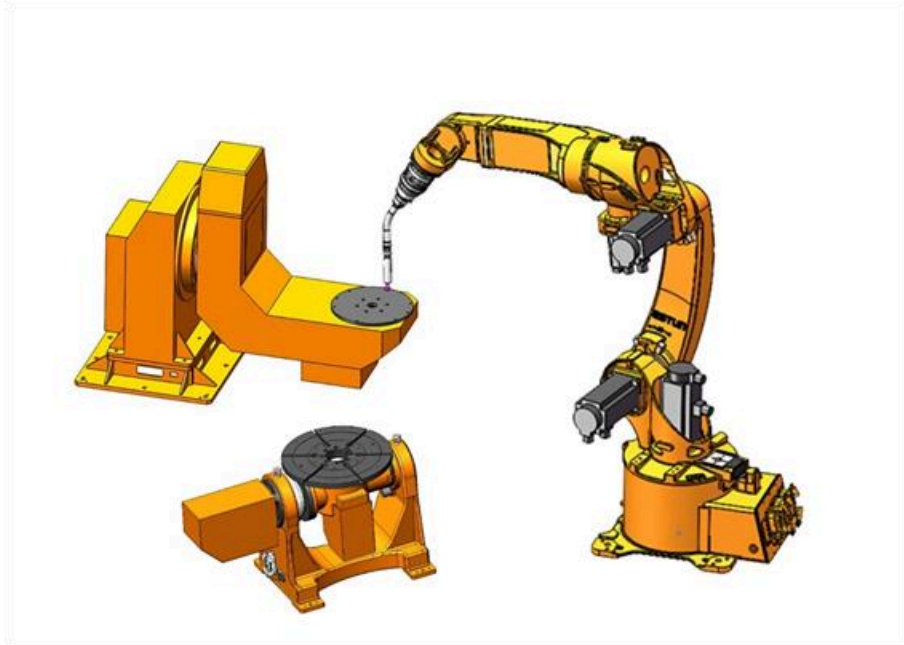
## 使用变机位坐标系

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。



其执行指令如下所示。

```
Start
MovJ{P=t_l.HomeP}
MovL{P=t_l.P0}
MovL{P=t_l.P1,Coord=t_g.POSITIONER1}
MovL{P=t_l.P2}
MovL{P=t_l.HomeP}
MovL{P=t_l.P3}
SetPositioner{Positioner=t_g.POSITIONER0}
MovL{P=t_l.P4}
MovL{P=t_l.P5}
MovL{P=t_l.HomeP,Coord=t_s.World}
End
```

## 7.12.5 一点回原位功能

### 功能简介

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，通过功能界面右侧进行原位位置的序号选择，并且支持对位置进行示教或者直接修改。

## 操作步骤

1. 在菜单选择“标定 -> 回原位”进入功能界面。



2. 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。

设置回原点输出使能及输出信号：

- 通过输出使能下拉框，打开或关闭输出使能功能，选择当前位置点变量是否开启检测 (On/Off)。
- 在输出IO下拉框，选择需要绑定信号输出的类型，DO或SimDO，并通过点击右侧输入框，输入具体的数字，以关联DO或SimDO序号，输入范围为0至32。

回原点输出信号：

- 当输出使能打开并且选择输出IO后，回原位到达预定点时会触发选择的IO信号。



### 📖 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。

可通过按示教器 Rob 键快捷到达一键回原位界面。

能够选择不同的IO进行输出，当输出使能关闭时，不会有IO信号输出。

## 7.12.6 工具对正

### 功能简介

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。

## 操作步骤

1. 选择进入系统主页中点击左上角，点击“标定 -> 工具对正”，进入“一点对正坐标轴”界面。



2. 选择“对正类型”。
  - 单轴对正：仅对工具坐标系的 Tz 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（Tx、Ty、Tz）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。

---

3. 点击“激活”。  
若要取消对正操作，可点击“禁能”。
4. 点击右侧对应的“+/-”按钮进行对正。当有弹窗提示代表对正到位。

## 7.12.7 零点标定

### 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，但以下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。



图 7-3 零位信息丢失

## 说明

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。

系统支持两种回零方式：整体回零和单轴回零。

## 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。

1. 点击主菜单，选择“**标定 -> 零点标定**”，进入回零功能页面。
2. 点击界面右上角开关，在弹窗中输入密码后，可解锁回零功能。



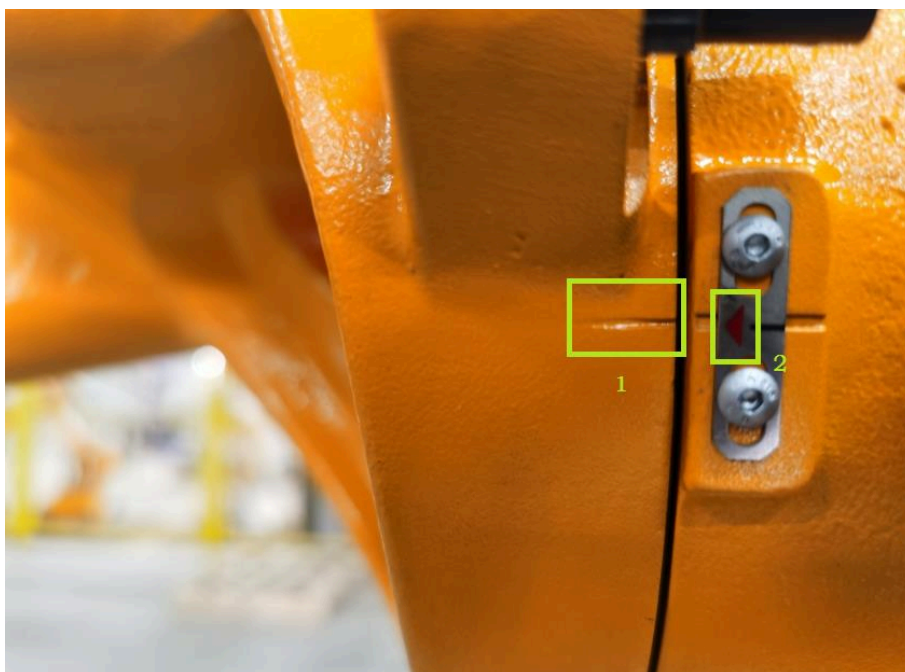
3. 完成功能解锁后，点击“整体回零”按钮，二次确认后，执行整体回零。



## 单轴回零

单轴回零是通过对应轴一键回零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。

1. 将机器人轴点动到本体上对应轴的零标刻度线附近。  
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。



### 📖 说明

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。

2. 在示教器“零点标定”界面中，“设置单圈值”列默认显示出厂校准时记录的单圈值信息，“实际单圈值”列显示当前实际单圈值。



### 3. 执行单轴回零。


点击需要回零的轴号对应的“回零”按钮，弹窗二次确认后，进行单轴回零。

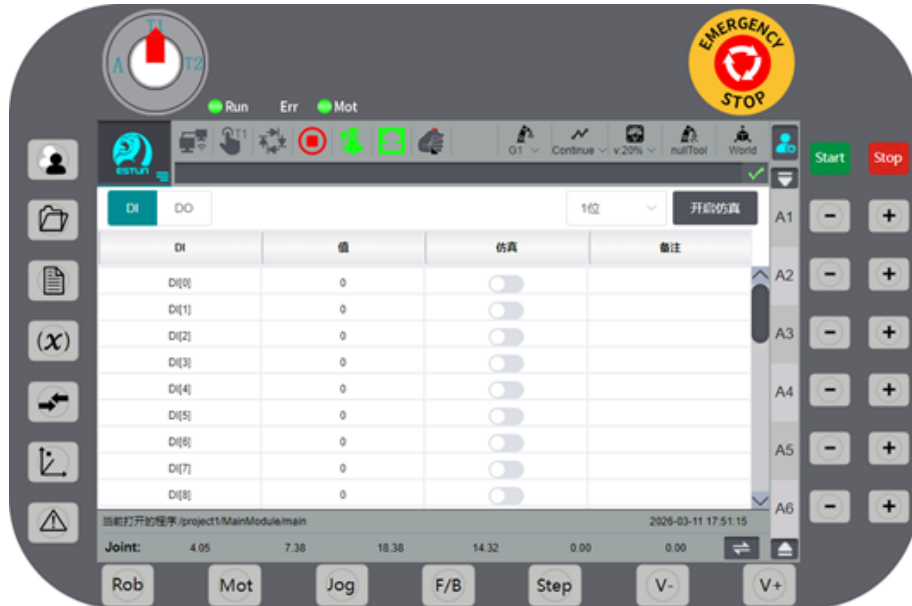
#### 📖 说明


- 一键导入：是指可以将当前“实际单圈值”输入到“设置单圈值”中。
- 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、6 有耦合，5 对 6 有耦合）时，回零顺序应为 4->5->6。

## 7.13 I/O

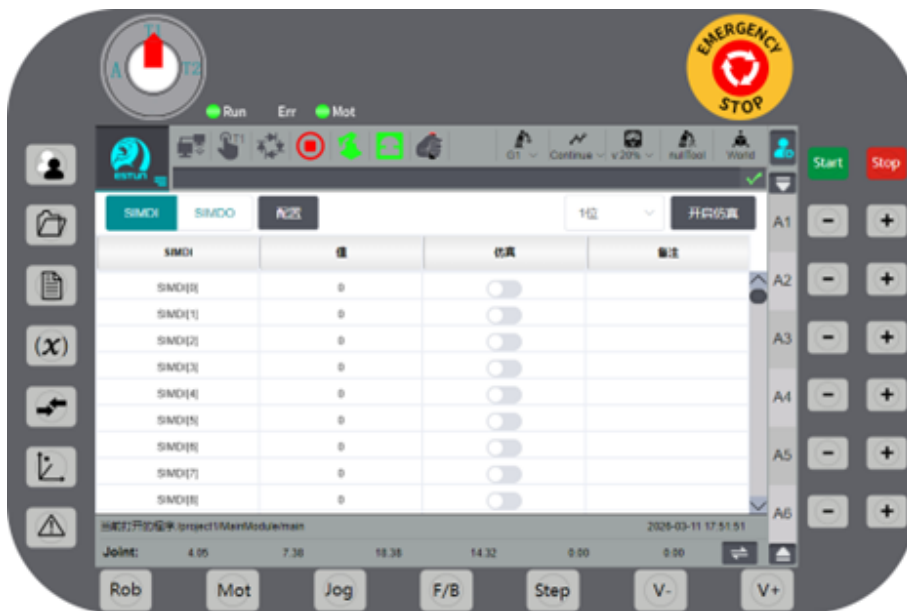
### 7.13.1 IO界面

1. 点击 ，选择“**I/O**”即可进入IO监控页面，默认进入的页签是DI/DO界面。该页面可以监控相关IO的状态以及设置其值。



2. 点击 选择SIMDI/SIMDO，进入SIMDI/SIMDO对应监控界面。





3. 点击“配置”按钮，即可进入映射配置界面。



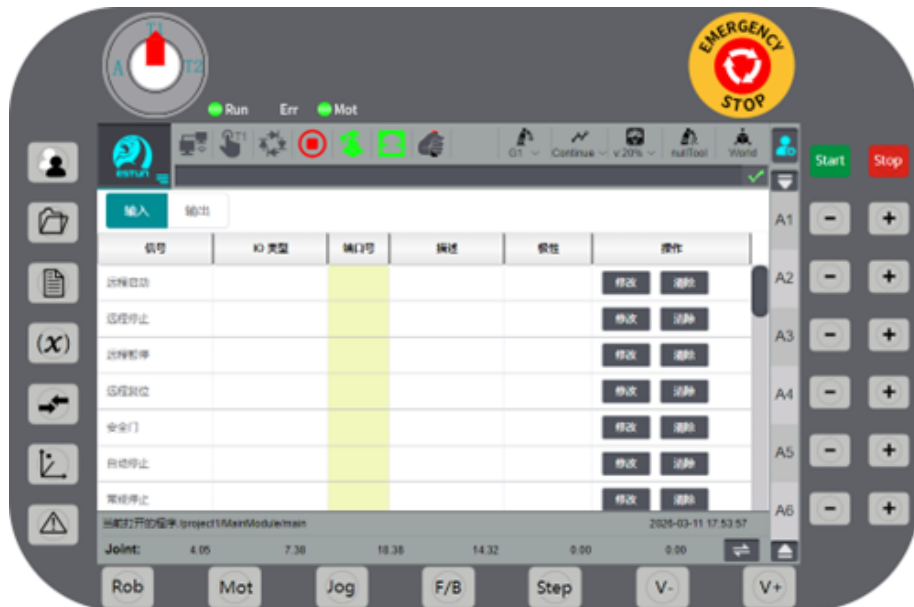
4. 在该界面中可完成EIP Adapter、ModbusServer和ProfiNet的映射配置。第一列的IO范围中的IO数据将与通讯数据对应的起始位作映射。

示例：可将通讯数据的对应比特位开始的数据映射到SIMDI其中一段或一个上。



## 7.13.2 系统事件与IO信号关联

1. 点击 ，选择“**I/O -> 系统信号**”进入系统信号配置界面。



2. 点击对应信号进行修改，选择IO类型并绑定端口号。



后续也可在IO关联主界面统一对创建的IO关联进行管理。

表 7-2 输入类信号功能

参数	描述	模式	生效方式
远程启动	运行程序	远程	边沿触发
远程暂停	暂停程序	远程	电平触发
远程停止	停止程序	远程	边沿触发
远程复位	清除错误	远程	边沿触发

表 7-2 输入类信号功能 (续)

参数	描述	模式	生效方式
安全门	/	任意	电平触发
重新加载自启动程序	加载自启动工程	远程	边沿触发
急停	/	任意	电平触发
卸载工程	卸载当前工程	远程	边沿触发
心跳检测	2秒中未检测到信号变化则报警	任意	边沿触发
控制权限	控制运动相关	任意	电平触发
上励磁	需控制权限	远程	边沿触发
下励磁	任意模式且程序非运行	任意	边沿触发
设置全局速度	需控制权限, 值取决于预设速度值	手动	边沿触发
加载自定义项目	程序处于非运行, 工程名需为ES-TUN%d	远程	边沿触发
点动信号	需控制权限	手动	电平触发
预设速度值	需使用设置全局速度信号生效	远程	-
设置坐标系	设置点动坐标系, 需控制权限, 改变生效。 <ul style="list-style-type: none"> <li>• 1: 关节</li> <li>• 2: 世界</li> <li>• 3: 工具</li> <li>• 4: 用户</li> </ul>	手动	-
预加载工程号	工程名需为ES-TUN%d	远程	-

表 7-3 输出类信号功能

参数	描述	参数	描述
运行状态	程序状态	是否运动	机器人是否在运动
暂停状态		系统锁机状态	-
停止状态		PC首行	PC是否在首行
伺服状态	励磁状态	手动模式	当前操作模式
错误状态	是否存在错误	自动模式	
重新加载工程成功	加载成功后一直为1, 仅加载失败为0	远程模式	

表 7-3 输出类信号功能 (续)

参数	描述	参数	描述
示教器连接状态	/	轨迹在途	/
心跳输出	500毫秒的方形波	安全门状态	/
当前速度	全局速度值	急停状态	/
当前坐标系	当前点动坐标系类型 <ul style="list-style-type: none"> <li>• 1: 关节</li> <li>• 2: 世界</li> <li>• 3: 工具</li> <li>• 4: 用户</li> </ul>	轴报警状态	/
TCP速度	/	轴报警代码	以十进制显示
报警代码	以十进制显示	当前坐标系ID	/
当前加载的工程号	工程名需为ESTUN%d	当前负载ID	/
当前位置	根据当前点动坐标系返回对应的值	/	/

### 说明

目前输入类型的信号均为上升沿或高电平触发，输出信号均为高电平有效。

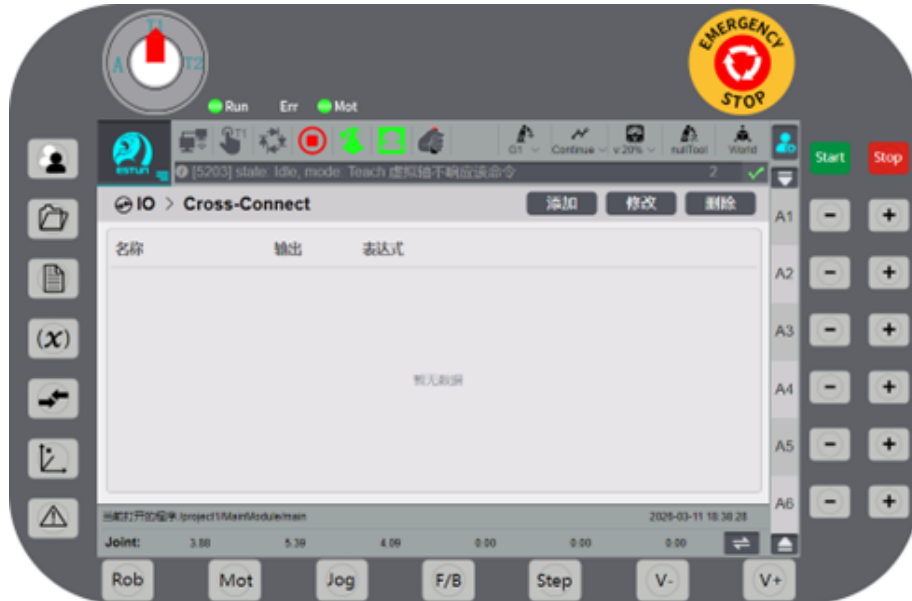
## 7.13.3 Cross-Connect

### 功能简介

Cross-Connect功能，通过逻辑运算来控制其他IO信号。此功能相当于一个简单的PLC，机器人可以通过此功能来监视与控制外部设备的运行。用户可在显示界面了解目前所有已配置的Cross-Connect详情信息，并在配置界面对已有的IO信号进行配置。

## 操作步骤

1. 选择系统菜单中的“**I/O -> Cross-Connect**”，进入功能界面。



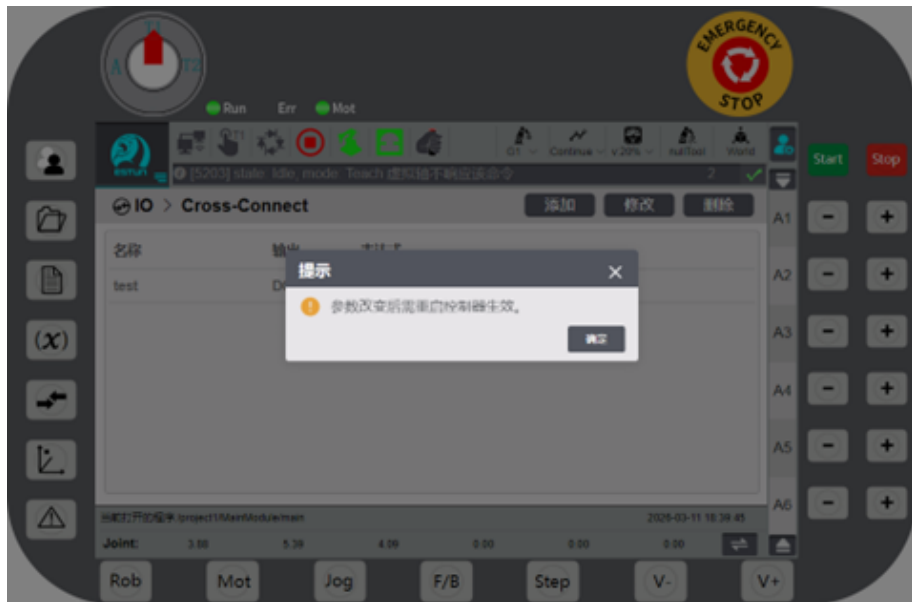
2. 添加：点击“**添加按钮**”，进入新建连接界面。



3. 填入需要设置的参数项，点击“**确认**”。



4. 重启后生效，重启后默认后台生效。



5. 修改：选中想要修改的表达式，点击“修改”，即可进入参数界面进行修改设置。

### 📖 说明

表达式名称不支持修改。

6. 删除：选中想要删除的表达式，点击“删除”，即可完成删除操作。

 说明


配置时IO信号不存在环路。

单个信号耦合最多不超过5层，多个信号耦合最多不超过20层。

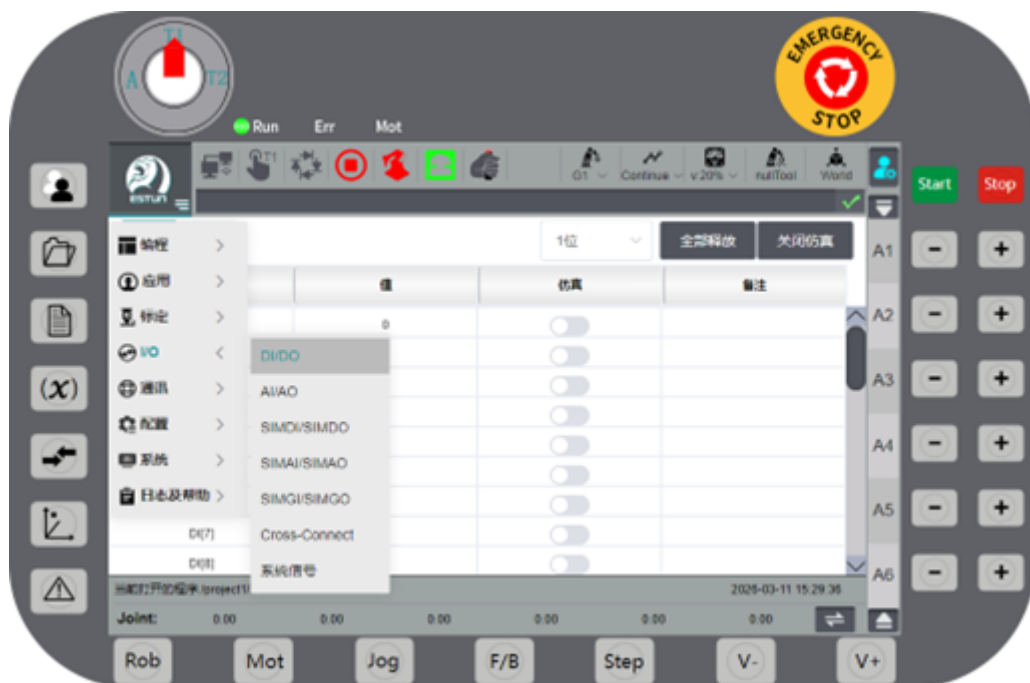
按照耦合信号先后顺序进行运算。

## 7.13.4 外部IO控制

### 功能简介

按  或在菜单栏上点击“**I/O**”进入IO监控界面。

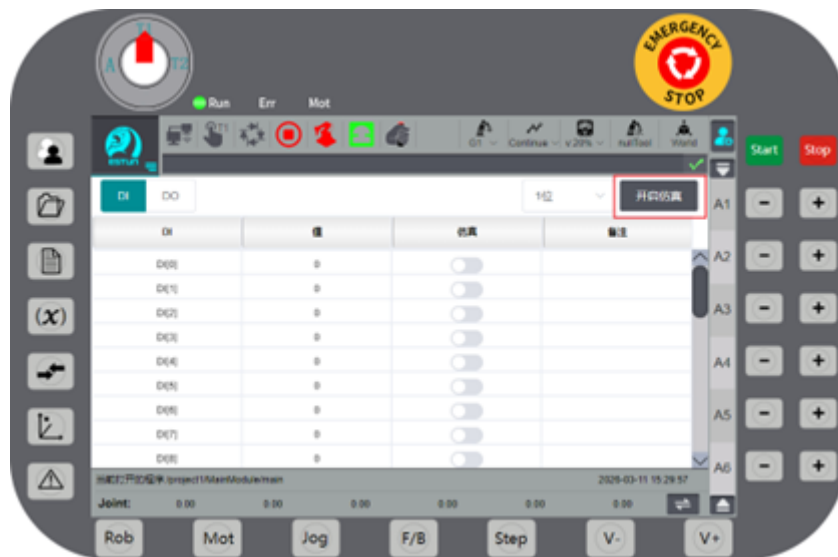
主要由DI/DO、AI/AO、SIMDI/SIMDO、SIMAI/SIMAO、SIMGI/SIMGO、系统信号六个页签组成，默认进入DI/DO界面。



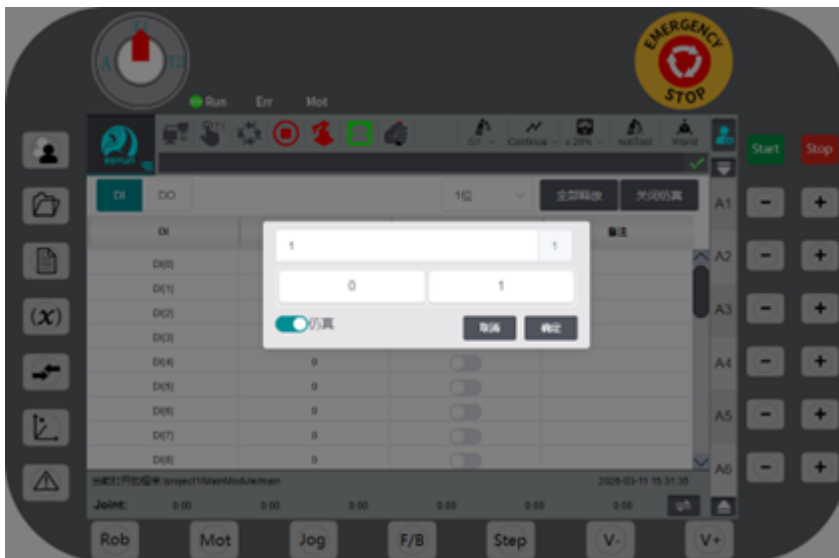
序号	含义	取值
1	输入量	DI、AI、SIMDI、SIMAI、SIMGI
2	输入信号	0: 低电平; 1: 高电平
3	输出量	DO、AO、SIMDO、SIMAO、SIMGO
4	输出信号	0: 低电平; 1: 高电平

## 修改物理IO仿真值

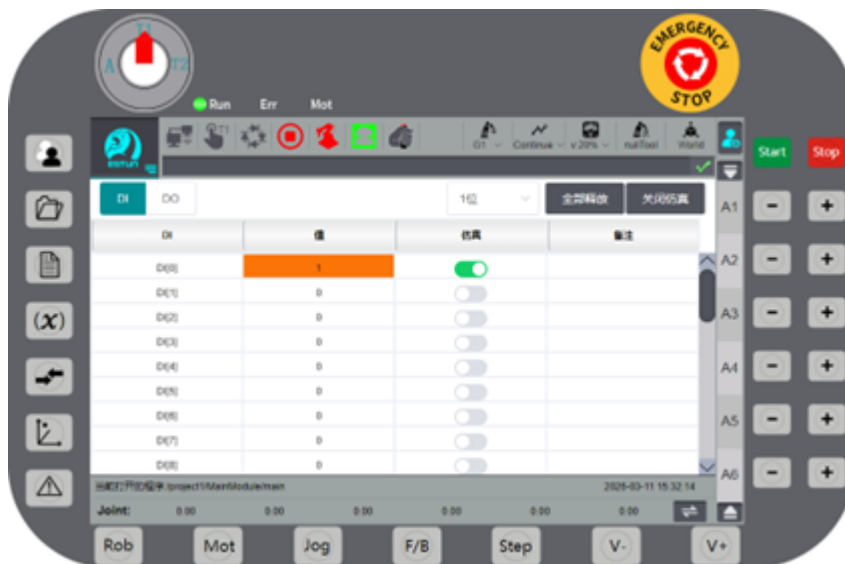
1. 点击右上角“开启仿真”，打开仿真使能。



2. 点击需要修改数字量IO的值，在弹框中设置值，并勾选左下角“仿真”复选框。



3. 点击“确认”，对应端口的值变成设置的值且成橙色时，表示修改成功。



4. 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左下角复选框，点击“确认”。或直接点击需要修改的 IO 数字量对应的仿真状态（即修改“仿真”列状态），释放仿真状态。
- 全部释放：释放所有输入量仿真状态及仿真值。
  - 关闭仿真：关闭输入量仿真功能。
  - 开启仿真：开启输入量仿真功能。

### 📖 说明

- 虚拟IO设置与物理IO方法相同。
- 用户可点击“全部释放”，对当前所有的输入量仿真状态进行释放，输入量包括DI、AI、SIMDI、SIMAI、SIMGI。
- 若需关闭仿真开关，可点击“关闭仿真”按钮，关闭后所有的输入量将重置为初始状态，此状态下不可对输入量进行仿真及设置仿真值操作。
- 打开仿真开关后，方可重新设置仿真，同时输入量仿真状态及仿真值将恢复至仿真开关关闭前的最后状态。

## 7.14 通讯

## 7.14.1 九点标定

### 功能简介

九点标定功能，可将视觉坐标系下的点位，直接转为为机器人坐标系下的点位。同时可以启用多个相机，此时外部相机为通信服务端，机器人主动与外部相机来建立连接，即可满足机器人与多个视觉进行连接的效果。

### 操作步骤

1. 通过系统导航栏选择“**通讯 -> 九点标定**”，进入“**九点标定**”界面。



2. 先选择目标相机索引。  
如果是默认相机则无需在此处设置IP和Port参数，只需启动九点标定，设置坐标系方向，点击“**下一页**”进入标定操作。  
如果是非默认相机，则需要依次先点击启用，设置相机IP、相机Port、启用九点标定、设置坐标系方向，再设置连接参数，设置坐标系方向，点击“**下一页**”按钮进入标定操作。
3. 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。
4. 手动操作示教器移动机器人至标定点点1，下拉选择点位1，手动输入标定点点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，然后点击下方“**示教**”按钮记录当前机器人坐标点，并点击“**保存**”。



5. 根据步骤4依次完成9个点的标定。当第9个点示教完成后可以点击“**标定**”，标定结果会显示到左侧位置。
6. 误差结果正常在 $\pm 0.1$ 范围内，如果误差过大建议重新标定。标定完成后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。

## 7.14.2 总线配置与监控

总线配置与监控可分为如下几块功能：

1. EIP激活。
2. EIP配置。
3. EIP状态、寄存器监控。
4. Modbus网口配置与监控。

用户通过示教器界面可以激活EIP功能、进行EIP配置并实现EIP正常通讯以及可以监控EIP和Modbus的寄存器值。

### 7.14.2.1 EIP授权激活

1. 选择菜单中“系统 -> EIP软件授权”，点击进入EIP软件安装及激活界面。



2. 点击“前往授权”。



3. 选择激活“标准版”。

获取产品码后，通过输入正确的激活码进行激活。



### 7.14.2.2 EtherNet/IP配置

在Scanner或Adapter软件成功安装并激活后，用户可通过示教器进行“EIP配置”，建立EIP主站与从站之间的连接，进行数据交互。

1. 点击菜单，选择“**通讯 -> EtherNet/IP**”，根据实际使用的软件进行详细配置。  
若当前未安装或授权，则跳转EIP软件授权界面，请参照[EIP授权激活](#)依次完成EIP软件的安装及激活后，方可正常使用EIP功能。
2. 配置界面左侧进行Adapter/Scanner的切换，点击右侧可显示当前网口、IP地址以及具体的设备列表和连接状态等信息。




 说明

EtherNet/IP主界面可配置EIP的网口，通过网口下拉框进行选择；目前支持Scanner配置lan3和lan4，Adapter仅支持lan4。

默认情况下，Scanner为lan3网口即Eth2，Adapter默认lan4网口即Eth3。

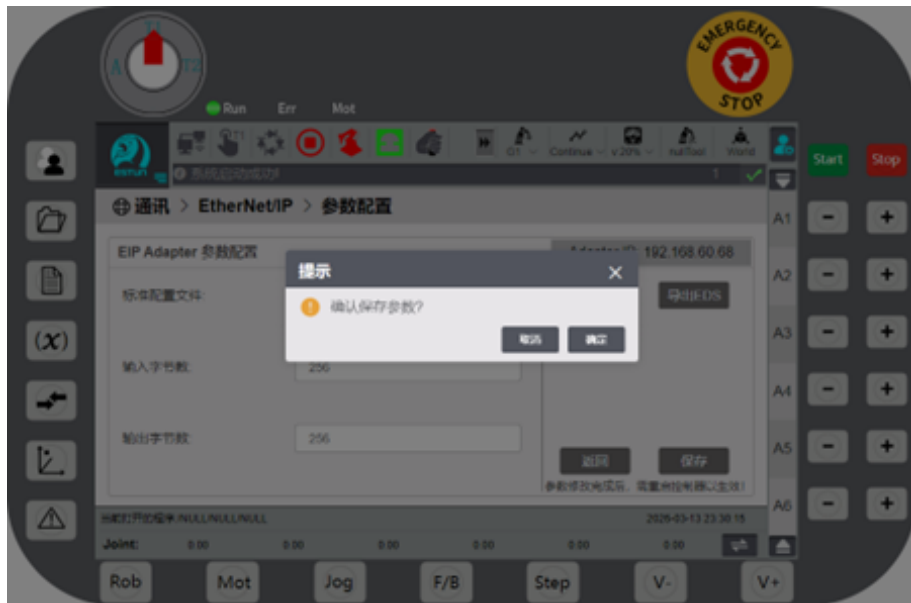
点击“EDS管理”，可前往导入/导出EDS文件。或通过菜单选择“系统 -> 系统维护”，进入EDS文件管理页面入口。EDS文件的导入/导出操作步骤同工程文件导入导出。

## 3. Adapter - 设置方法




- a. 进入Adapter界面，可查看当前已有的EIP连接信息。用户可点击  按钮，进入参数配置页面进行修改。



- b. 根据实际需求设置参数，修改完成后，点击“保存”，进行弹窗二次确认。参数设置完成后，需要重启控制器以生效。



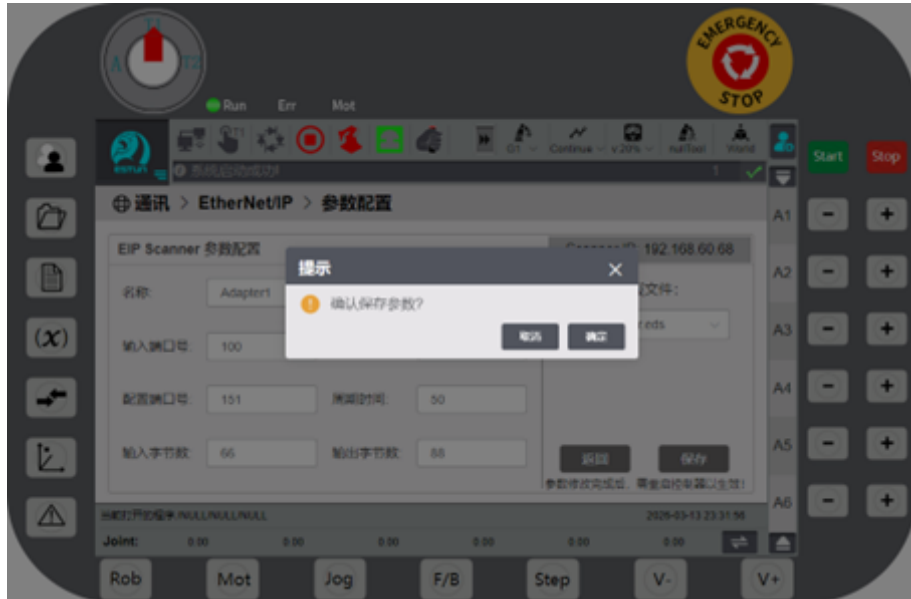
#### 4. Scanner - 设置方法

- a. 进入Scanner界面，可查看当前已有的EIP连接信息。用户可点击  或  操作按钮，进入参数配置页面进行连接设备的添加或修改，或点击  操作按钮，来删除选中的设备。





- b. 根据实际需求设置参数，或于界面右侧选择EDS模板文件，可自动填充文件内参数。修改完成后，点击“保存”，进行弹窗二次确认。参数设置完成后，需要重启控制器以生效。



### 📖 说明

不支持相同 IP 的连接。

连接参数信息的范围建议以实际使用为主。

增加连接时注意最大连接数不能超过 16 个。

周期设置最短不能低于 20ms，低于 20ms 以后准确性和稳定性无法保证。

### 7.14.2.3 EIP状态寄存器监控

1. 通过点击EIP主页左下角“状态值监控”，进入EIP寄存器监控功能界面，可查看当前设置的各寄存器的实时值。



2. 用户可新增配置寄存器监控项进行状态值监控，或对现有寄存器监控项进行编辑或删除操作。



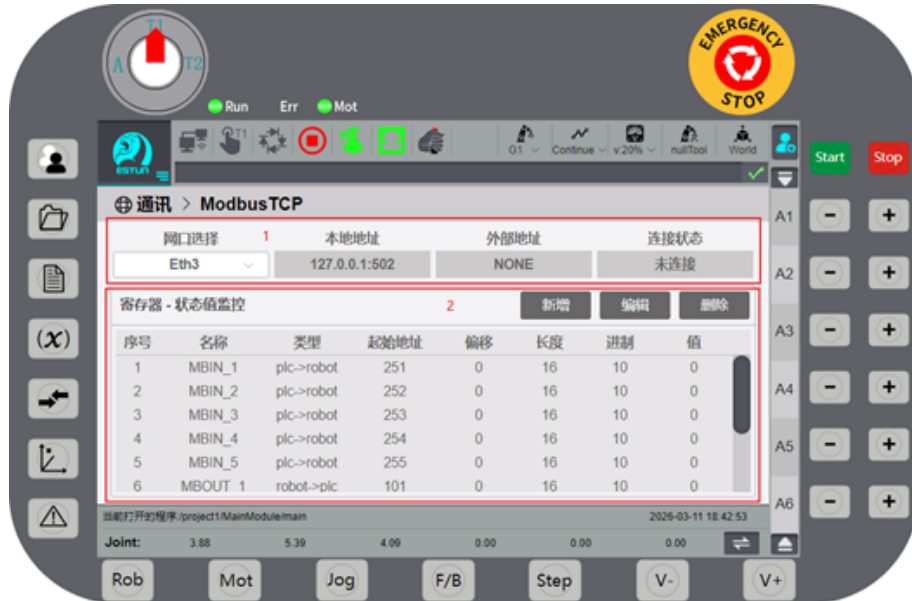
## 说明

EIP状态值监控列表会实时显示各寄存器的值，与Multiprog中值保持一致。

通过在Ethernet/IP主页左侧选择Scanner或Adapter页签栏，进入相应的Scanner或Adapter状态值监控入口。

### 7.14.2.4 Modbus网口配置与监控

1. 选择菜单“通讯 -> Modbus TCP”，进入Modbus网口配置与监控页面。



- 功能界面上方显示当前Modbus网口、本地地址、外部地址以及实时连接状态等信息（Modbus控制器本地作为Server端）。
- Modbus网口配置：用户可通过网口选择下拉框，对Modbus网口进行切换。
- Modbus寄存器状态值监控：功能界面下方展示Modbus配置寄存器的参数配置值及对应实时值。
- 配置寄存器监控项：用户可对寄存器进行新增、编辑（配置）及删除操作。具体操作步骤请参见EIP状态寄存器监控中“EIP状态值监控项配置”步骤。

### 7.14.3 二维视觉

#### 功能简介

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。

## 操作步骤

1. 通过系统导航栏选择“通讯 -> 视觉”，进入“视觉设置”界面。



2. “视觉类型”选择为服务端时，表示外部相机作为通信服务端，此时机器人主动与外部相机来建立连接，下方参数为“视觉IP”和“视觉端口”，根据视觉实际的参数进行填写。
3. “视觉类型”选择为客户端时，表示外部相机作为通信客户端，此时外部相机与机器人主动建立连接，下方参数为“机器人端口号”，根据视觉实际的参数进行填写。
4. 该功能需配合视觉指令使用，请参见[视觉指令](#)章节。

---

### 📖 说明

- “视觉类型”目前仅支持静态类型。
  - 该功能需配合视觉指令使用，请参考视觉指令介绍章节。
- 

## 7.14.4 从站动态配置

用户可通过示教器的界面直接进行从站的添加、删除和查询等功能等操作。

1. 选择进入系统菜单中的“通讯”分组，点击“EtherCAT”。



2. 点击“+”可增加从站，通过下拉框选择从站类型。



3. 添加从站可查看对应从站的信息，并且可通过“上移”、“下移”来修改拓扑顺序。
4. 配置后点击“保存”，重启后即可生效。



5. 若控制器已连接好从站，可点击“扫描”并“保存”。若EtherCAT正常，可将ERC3-S3系列控制柜的安全IO数字扩展卡扫描出来。

#### 📖 说明

- Copling耦合器不支持AI/AO模块配置。
- ERC3-S3系列控制柜仅支持C1-CE/S1/C2控制柜。

### 7.14.5 ERI激活功能

在左上角选择“通讯 -> ERI软件授权”，该功能分为“标准版”和“试用版”两种，其中标准版需要输入激活码进行激活，而试用版会提供两个月的试用时间。



图 7-4 标准版

图 7-5 试用版



## 7.15 网络与时间

用户点击菜单“配置 -> 网络&时间”，进入系统网络与时间设置界面，示教器会根据控制器系统版本显示匹配的设置界面。

## 7.15.1 网络设置

- 在系统网络与时间设置界面，选择“网络设置”。
- 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，示教器会根据控制器系统版本显示匹配的设置界面。



### 说明

- 系统 IP 设置仅对“管理人员”用户类型开放。
  - 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。
  - 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，示教器 IP、OS1、OS2 的 IP 地址都不可与局域网其它设备 IP 相同，否则，网络地址冲突导致通讯异常。
- 
- 本地网络设置：可远程修改正在与示教器端连接的 OS 端 IP 地址。
    - Connect IP：当前连接IP。
    - TeachPendant IP：本地示教器IP。
    - Controller TP：修改控制器IP。
  - 外部网络设置：

- ETH0: 连接外部设备的 IP（如运维精灵）。
- Subnet mast: 子网掩码。
- Gateway: 网关。

### 📖 说明

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）

## 7.15.2 时间设置

在功能界面选择“时间设置”，可在该界面对控制器时间进行设置。仅厂家可对此参数进行设置，当前用户为非厂家人员时，设置时间按钮置灰，无法操作。



## 7.15.3 外部网络设置

### 📖 说明

- 当前版本该功能仅服务于多机传送带功能的通讯连接。
- 添加通讯对象时不能服务器对象与客户端对象同时存在，并且服务器对象只能创建单个，客户端对象最多只能创建5个。
- 创建多个客户端对象时，ID号为各个对象的唯一标识，不可重复。

1. 在功能界面选择“外部网络设置”，进入外部网络设置页面。



2. 添加网络通讯对象。

点击“添加”，弹出配置框，输入对应网络配置信息。



3. 点击“确认”。

- 成功添加客户端对象后页面如下。



- 成功添加服务器对象后页面如下。



#### 4. 删除网络通讯对象。

选中需要删除的网络通讯对象，点击“删除”即可删除。



5. 修改网络通讯对象。

选中需要修改的网络通讯对象，点击“修改”即可进入对象信息编辑界面。其中类型一栏在添加完对象后不可修改。



6. 使能网络通讯对象。

选中需要使能的网络通讯对象，点击右侧使能开关即可使能通讯对象。

- 若网络通讯对象为服务器，则当前机器人作为客户端会周期性地尝试连接目标服务器。若通讯建立成功，则状态一栏显示为“已连接”。若通讯未建立成功或使能被关闭，则状态一栏显示为“未连接”。



- 若网络通讯对象为客户端，则当前机器人作为服务器会将所有使能的客户端对象信息加入监听列表，若通讯建立成功，则状态一栏显示为“已连接”。若通讯未建立成功或使能被关闭，则状态一栏显示为“未连接”。



## 7.16 控制器参数设置

## 7.16.1 控制器设置

点击菜单“配置 -> 控制器设置”，进入控制器设置界面。在该界面用户可对控制器系统进行相关设置。

- 模式切换：系统处于自动/远程模式时，点击“自动/远程”按钮，进行控制系统在远程模式和自动模式之间的切换。若当前处于非自动/远程模式，此按钮将置灰，无法操作。
- 断电恢复使能：可选为ON/OFF，控制断电恢复使能的开启与关闭。
- 负载辨识使能：可选为ON/OFF，设置为ON时，开启负载辨识，为OFF，关闭负载辨识功能。（此参数修改后需重启控制系统生效）
- 模式切换&jog不转停止使能：可选为ON/OFF，代表开启或关闭。系统切换系统模式和点动时，系统会根据该参数确认系统是否转为停止状态。（此参数修改后立即生效）



## 7.16.2 远程设置

点击菜单“配置 -> 远程设置”，进入远程设置界面。在该界面用户可对远程模式相关参数进行设置。

- 远程速度：设置远程模式下系统全局速度百分比值，设置范围为0-100。
- API模式：设置外部（API）模式，可选为ON/OFF，表示打开/关闭外部控制模式。
- 远程自动励磁：可选为ON/OFF，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。
- 自启动程序：显示系统切到远程模式后会自动加载的程序。若程序设置为自启动状态，则显示该程序目录。



### 7.16.3 程序设置

点击菜单“配置 -> 程序设置”，进入程序设置界面。在该界面用户可设置开机默认加载的相关参数。

- 默认工具：选择开机默认加载的工具类型。默认值：nullTool。
- 默认坐标系：选择开机默认加载的坐标系类型。默认值：World。
- 默认负载：选择开机默认加载的负载变量。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。



**⚠ 注意**

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。

## 7.16.4 语言

点击菜单“配置 -> 语言&个性化”，选择左侧菜单中“语言”，进入设置界面。在该界面用户可更改示教器语言，目前支持中文，英语。

- 当前语言：表示当前示教器系统语言。
- 修改语言：选中对应语言后，点击“设置”按钮，弹窗确认后完成示教器系统语言的切换。




## 7.16.5 个性化

点击菜单“配置 -> 语言&个性化”，选择左侧菜单中“个性化”，进入设置界面。个性化里可以改变示教器屏幕亮度，超时动作，超时时间和字体大小。



## 7.17 一点到达

1. 点击示教器左侧  按钮，进入变量管理界面，选择类型里选择Cpos或Apos。



2. 点击变量行，在右侧弹窗中选择“一点到达”。



3. 选择运动方式，点击“移动至”即可。



## 7.18 锁机与解除

### 📖 说明

由于锁机管理功能对用户权限的限制，故只能“管理人员”或“厂家人员”权限才能进行锁机配置操作。

1. 点击主菜单按键，选择“系统 -> 锁机管理”，进入锁机管理界面。
2. 当进入锁机管理界面后，界面上会显示当前锁机状态、锁机期限剩余天数、“解锁”按钮、“设置”按钮和“清除”按钮。



### 7.18.1 设置锁机期限

在锁机管理界面中，点击“设置”按钮来对系统锁机期限进行配置。而设置锁机期限方式一共分为两种，分别为“分期锁机”和“定时锁机”。

### 7.18.1.1 设置分期锁机期限

1. 点击“设置”按钮，提供两种锁机期限方式可供选择。



2. 选择“分期锁机”，输入第一期、第二期、第三期的期限天数，并点击“设置”按钮使设置的锁机期限生效。



### 7.18.1.2 设置定时锁机期限

1. 点击“设置”按钮。



2. 选择“定时锁机”，输入锁机期限天数，并点击“设置”使设置的锁机期限生效。



### 说明

1. 机器人出厂时，在没有锁机配置的情况下，默认是无限期解锁状态，即用户可永久使用机器人全部功能。
2. 设置“分期锁机”的期限时，第一、二、三期的期限天数需全部填写，并一起设置生效。
3. 设置“分期锁机”的期限后，立即生效第一期锁机功能，起始时间为当前锁机时间。
4. 设置“定时锁机”的期限后，锁机功能立即生效，起始时间为当前锁机时间。

## 7.18.2 解除锁机期限

在锁机管理界面中，点击“解锁”来对系统锁机状态进行解锁。而解除锁机期限方式一共分为两种，分别为“分期解锁”和“定时解锁”。

### 7.18.2.1 解除分期锁机期限

1. 在点击“解锁”，用户需根据当前锁机状态来选择对应的解锁期限方式。



2. 点击“分期解锁”，用户需提供机器码和需解锁的期数给厂商人员，以获取解锁密钥。机器码主要包括固定码和动态码两部分，格式为：固定码 - 动态码。当系统处于分期锁机时，默认第一期是解锁状态。



3. 在解锁对应的期限后，之前所设置的期限天数开始生效，并在锁机管理界面上可查看当前的剩余天数。

### 7.18.2.2 解除定时锁机期限

1. 点击“解锁”，用户需根据当前锁机状态来选择对应的解锁期限方式。

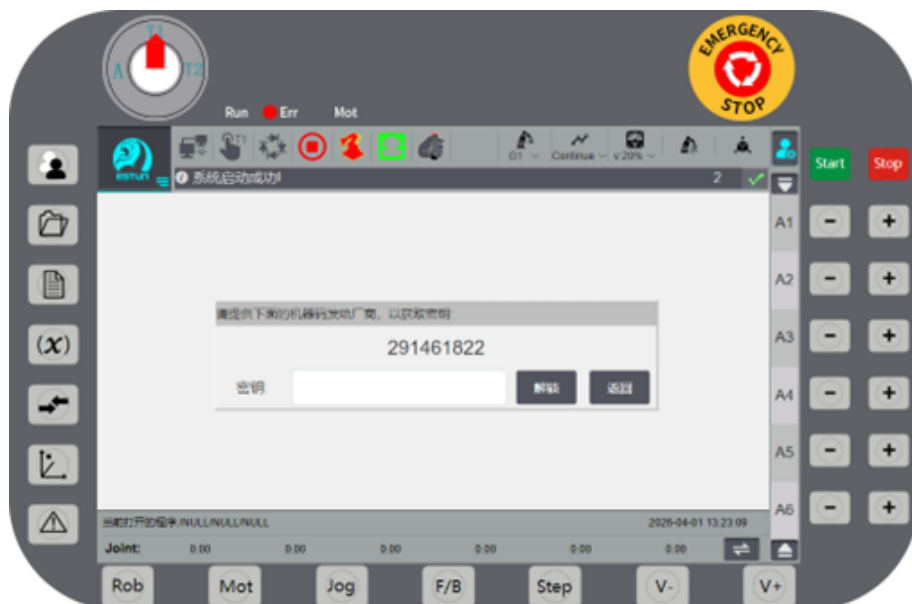


2. 点击“定时解锁”，用户需提供机器码和需解锁时间给厂商人员，以获取解锁密钥。机器码主要包括固定码和动态码两部分，格式为：固定码 - 动态码。



### 7.18.3 清除锁机期限

1. 点击“清除”，进入清除锁机期限的界面。根据界面显示的机器码，将其提供给厂商人员，以便获取清除系统锁机状态的解锁密钥。



2. 当解锁成功后，当前机器人则进入无限期解锁状态，可在锁机管理界面查看当前锁机状态为“无限期解锁”，剩余天数为“100000天”。



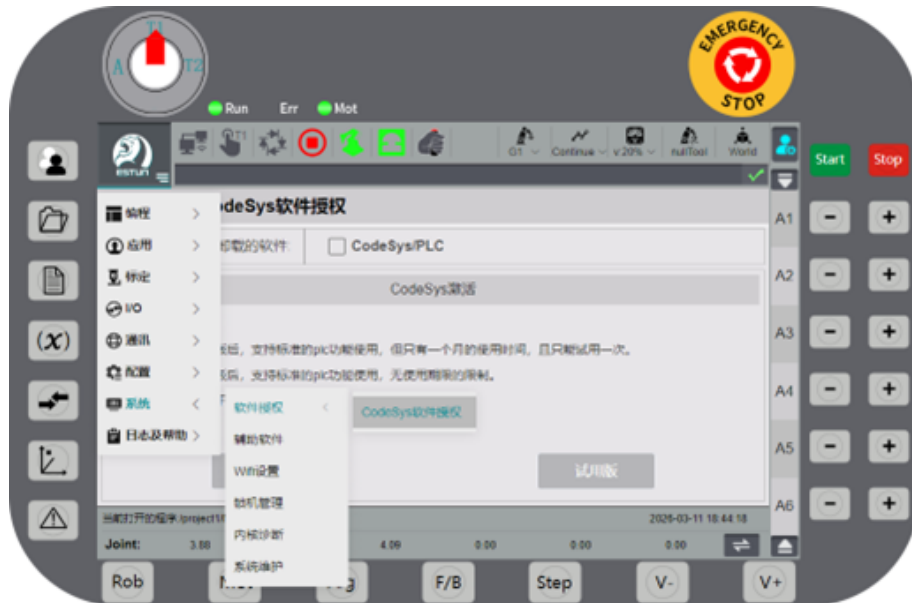
## 7.19 CodeSys功能授权

### 功能简介

通过示教器界面，对CodeSys软PLC功能开放使用授权，控制软PLC的使用。用户可在CodeSys授权界面，获取对应控制器的软PLC功能产品码，并输入对应激活码来完成CodeSys软PLC功能授权。成功激活后，CodeSYS软PLC功能和自定义库能够正常使用。

## 操作步骤

1. 选择菜单中“系统”，点击“CodeSys激活”，进入功能界面。



2. 首先勾选需要安装的软件，若界面显示为勾选状态，则说明已安装软件；如需卸载软件，取消勾选框即可完成软件卸载操作。
3. 成功安装后，选择激活“标准版”或“使用版”。

---

### 说明

当前软plc为CodeSys时，用户可通过点击页面的激活按钮，选择标准版或试用版进行激活。否则页面按钮置灰，用户无法进行软件安装及激活相关操作。

---



4. 点击激活“标准版”，进入授权界面，用户获取产品码后，通过输入正确的激活码进行标准版本激活。



5. 或点击激活试用版，弹窗确认激活后，进入授权页面，激活试用版成功，首次激活试用版试用天数为30天，如下图所示。（注意：试用版仅可试用激活一次。）



### 📖 说明

此激活仅适用于软plc为CodeSys，不适用于Multiprog功能。

试用版仅可激活试用一次，有效期为30天。试用版到期后，如需继续使用软件请激活标准版。

## 7.20 示教器分屏功能

通过打开分屏功能，使得示教器上能够呈现左右两块页面，分别展示不同的内容。

1. 击菜单中“配置 -> 语言&个性化”进入分屏设置。




2. 分屏模式下示教器页面可以展示两个不同的插件。

- 页面切换：

仅支持示教器左侧物理按键切换插件页面。点击选中左侧/右侧屏幕后，点击示教器左侧按钮，页面即可跳转至按钮对应插件页。



- 退出分屏：

如需退出分屏模式，可点击左上角  菜单图标，则关闭分屏，恢复原示教器界面。

## 7.21 轴锁功能

### 功能简介

机器人配置附加轴时，默认附加轴是不启用的。需在“机构配置”界面，激活需要使用的附加轴，可以通过配置不同的机构来绑定不同的附加轴。当指定机构没有激活相应附加轴时，该附加轴不会参与运动。

### 操作步骤

选择“配置 -> 机构设置”，设置对应机器人附加轴激活情况。勾选附加轴时，表示该轴为激活状态，可以参与运动及点动。



如下图所示，G3，G4，G5，G6分别表示rob3附加轴的四中不同的锁定情况。

机器人rob3共配置了3个附加轴，G3机构中，E1、E2、E3都处于激活状态。G4机构中，E1激活状态，其他处于锁定状态。G5机构中，E2激活状态，其他处于锁定状态。G6机构中，E3激活状态，其他处于锁定状态。

当新建工程，绑定对应机构时，附加轴信息也一同绑定。



示教器界面，右侧点动窗口，未激活的附加轴不会显示。



## 7.22 外部TCP功能

### 功能简介

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，标定操作请参见[工具坐标系标定](#)、[用户坐标系标定](#)。



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。

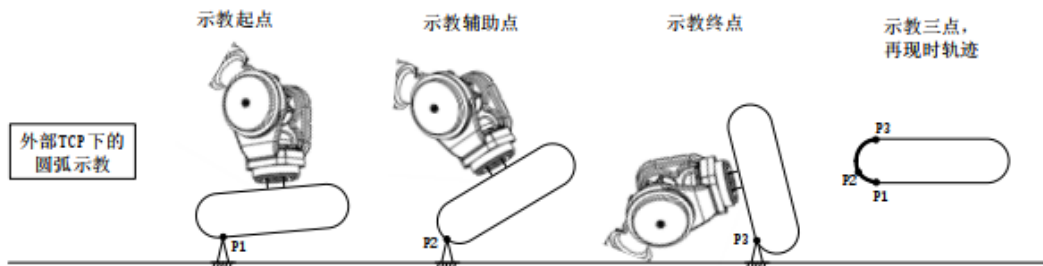
在 MovL、MovC 指令中可以选择参考坐标系为外部工具坐标系，也可通过设置指令 SetExternalTCP 设置当前的参考坐标系为外部工具坐标系。当使用外部 TCP 示教直线圆弧运动时，直线示教 2 点、圆弧示教 3 点。

## 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。

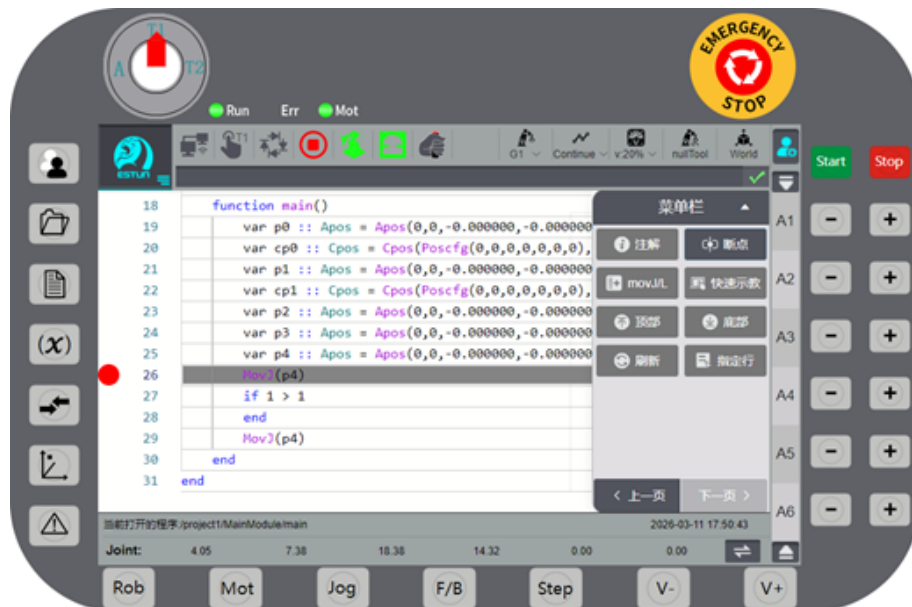
## 使用示例





## 7.23 断点调试

在程序编辑界面选择需要断点调试的点，点击右下角显示菜单，在菜单里找到断点按键，点击即可开启断点调试。可以在程序的某一行或多行添加断点，每当程序运行到断点行时会停下，方便用户进行调试。可以取消已经设置的断点。



## 7.24 PLC开放功能

机器人控制系统正常运行，PLC编程环境开放给用户使用，如控制mc轴。开放PLC编辑环境给用户，可方便实现复杂应用和系统的集成，可生成定制化的bootfile文件。

## 7.25 倒序功能

### 功能简介

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。

## 操作步骤

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。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。

---

### 说明

使用倒序功能时，可执行的指令如下所示，其余指令均不支持。

MovJ, MovL, MovC, MovLW, MovCW, MovLOffset.

---

## 7.26 飞拍功能

### 功能简介

SCARA机器人常见的应用场景之一为抓取和放置物料。为了提高放置精度，一般采用视觉系统进行纠偏，即：在机器人从抓取点抓取物料到放置点放置物料的过程中，经过一个由下而上的视觉系统，视觉经拍照处理后给出位置偏差值（即抓取偏差），系统根据偏差值进行放置点的位置修正，以保证放置点的精度。

传统纠偏技术为定拍，即机器人在经过视觉拍照时，先停下来再进行位置纠正，虽然精度高，但是节拍较慢。而飞拍技术，在运动的过程中进行视觉拍照，并进行位置纠偏，无需停下来，可大幅提高效率。

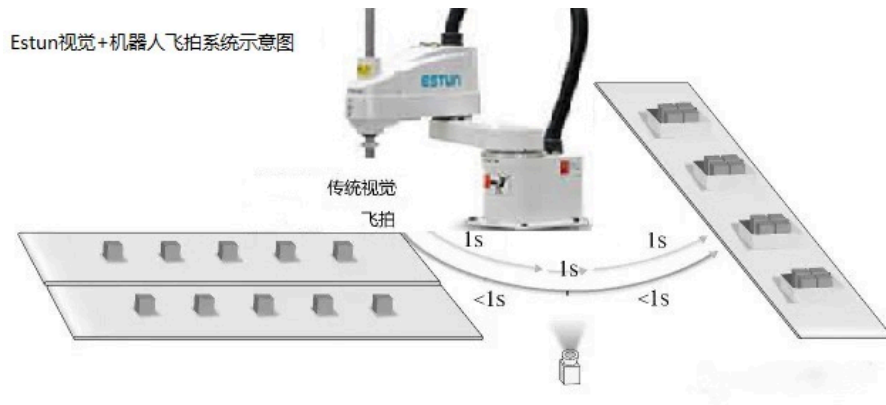


图 7-6 ESTUN视觉+机器人飞拍系统示意图

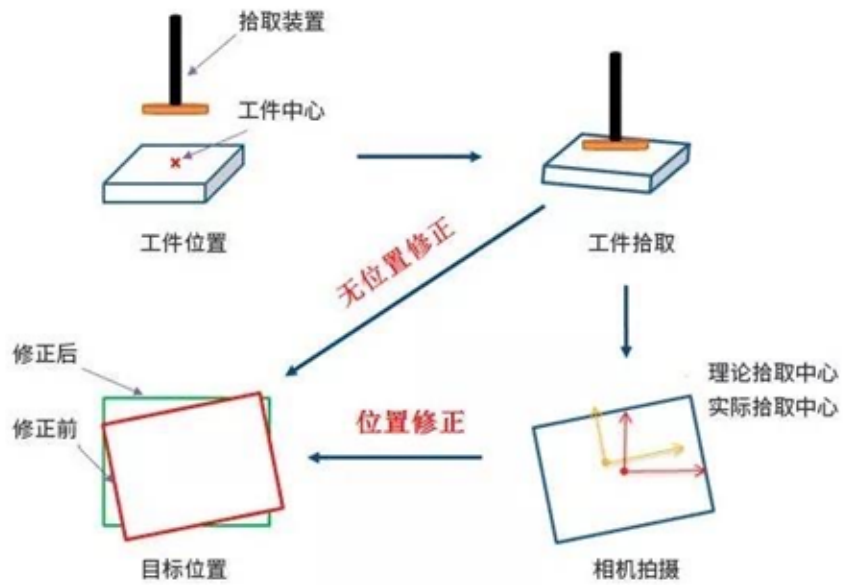


图 7-7 目标物体纠偏示意图

📖 说明

使用飞拍功能，需要搭配使用视觉相机与伺服驱动器的PSO（Position Synchronized Output位置同步输出）功能。

## 操作步骤

- 与视觉相机建立通讯连接  
相机设备类型选择“2D”，并配置IP与端口号（默认为192.168.6.230:5000），视觉类型选择“静态”。所有配置重启后生效，控制器作为客户端，与相机服务器建立通讯。

- 飞拍功能指令使用

分为两部分，调试与执行：

- 飞拍调试

调试目的是让伺服驱动器记录下指定轴的当前位置作为比较点，在后续机器人运动到记录的位置时，输出信号触发相机拍照。调试示例程序如下所示：

调试代码执行结束后，此时伺服驱动器会记录下1轴的当前位置作为第1个比较点。

```
Start
MovJ{P=t_l.P_Home}
MovL{P=t_l.P_StartUp}
Wait{T=1000}
LatchEnable{AxisID=1,EnableSts="ON"}
MovL{P=t_l.P_Camera}
Wait{T=1000}
LatchEnable{AxisID=1,EnableSts="OFF"}
SetLatchPos{AxisID=1,ComparePointID=1,Ret=t_l.INT0,P=t_l.P0}
End
```

- 飞拍执行

调试完成后，执行示例程序如下所示：

```
Start
MovJ{P=t_l.P_Home}
MovL{P=t_l.P_StartUp}
MovL{P=t_l.P_Start}
SetDO{IO=t_l.DO1,Val=1}
LatchEnable{AxisID=1,EnableSts="ON"}
MovL{P=t_l.P_StartUp,B="ABSOLUTE",C=t_s.C100}
MovL{P=t_l.P_TargetUp}
WaitFinish{Percent=10}
WaitCameraDCPOS{WTime=10000,Vision=0,TP=t_l.CameraDCPOS,tagID=t_l.INT0,Ret=t_l.INT1}
IF t_l.INT1.value ~= 1 THEN
SetRtToErr{msg="获取相机偏移点失败",eid=90001}
```

```

ENDIF
t_l.INT1.value = 0
MovLOffset{RP=t_l.P_Target,TP=t_l.CameraDCPOS,T="COORD",Tool=t_s.nullTool,
Coord=t_s.World}
SetDO{IO=t_l.DO1,Val=0}
LatchEnable{AxisID=1,EnableSts="OFF"}
MovJ{P=t_l.P_Home}
End

```

## 7.27 中断功能

### 功能简介

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。

- 目前的中断信号检测有三种方式：检测DI或SimDI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。
- 中断程序不支持全局函数，只允许使用本工程下的函数。

### 操作步骤

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，指令包括IConnect、IDelete、ITimer、ISignalDI、ISignalSimDI, MovToI-Pos, ExitCycle, ISleep, IWatch。

### 说明

ISignalDI或ISignalSimDI指令支持“去重时间”，当触发一次中断后，在指定时间内不再响应中断触发信号。

在中断程序中只允许部分指令执行，具体指令列表如下：

指令类型	具体指令
运动指令	除MovH、MovArch以外的运动指令
控制指令	全部支持

指令类型	具体指令
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	MovToIPos、ExitCycle、ISleep、IWatch

## 场景说明

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重新执行。

## 配置示例

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。

Prog2为中断触发后立即停止，待中断程序执行完退出中断并跳转至主程序首行。

Prog1绑定了DI 10上升沿触发，Prog2绑定了DI 11上升沿触发，当DI信号满足时会执行对应的中断程序。

```
function main()
  IDelete(INTERRUPT1)
  IDelete(INTERRUPT2)
  IConnect(INTERRUPT1, Prog1)
  IConnect(INTERRUPT2, Prog2)
  ISignalDI(INTERRUPT1, 10, "RiseEdge", 1000)
  ISignalDI(INTERRUPT2, 11, "RiseEdge", 1000)
  GetCurCPOS ( P0)
  if P0 != PHome
  MovL(PHome)
  end
  tt:
  MovL(P1)
  MovC(P2,P3)
  ...
  MovL(P10)
  Goto tt
end

function Prog1()
  MovL(P1)
  MovToIPos("MovJ",V200)
end

function Prog2()
  MovL(P1)
  ...
  ExitCycle()
end
```

## 7.28 常用指令组

### 功能简介

支持用户将常用指令添加至常用指令组中，提高编程时新建指令的效率。

支持用户自定义常用指令集，包括在该分类中新增及删除指令。

### 操作步骤

- 查看常用指令。  
指令列表顶部“常用指令”分组，该分组为常用指令的集合。用户可正常查看及使用该分组下的指令。



- 自定义常用指令。  
选择“编程 -> 常用指令”，进入常用指令组编辑界面。用户在该界面可选择指令添加至常用组，或者移除常用组。

- 添加至常用组。选中左侧指令，点击“>>”添加至“常用指令”，二次确认后，完成添加操作。
- 移除常用组。选中右侧指令，点击“<<”，二次确认后，完成移除操作。



## 说明

函数调用指令不属于系统内置指令，不支持将函数调用指令添加至常用指令组。

## 7.29 外部接口管理权限

### 功能简介

远程模式耦合了Modbus、API、PLC、ERI等外部控制方式，每个方式都有不同的操作权限。在示教器界面提供权限配置界面，统一配置机制，细分各个外部接口的控制权限。

### 权限查看

在示教器菜单选择“配置 -> 外部控制权限配置”，可查看外部控制权限表。

- 单元格显示“#”，表示该外部控制方式配置此权限。
- 单元格显示“—”，表示不具备对应权限。



## 权限修改

仅厂家人员可对权限进行配置，通过点击具体单元格，可修改对应的权限值。



---

 **说明**

- 控制方式为TP时的权限不支持修改。
  - 仅支持手动、自动模式下修改权限，程序运行过程中不允许修改权限。
  - 非厂家人员仅可查看当前权限的配置情况，无法操作权限表。
  - 当外部控制时，功能按照相应权限执行。例如，外部控制为API时，设置卸载工程无权限，则不可以执行卸载工程，否则出现报警。
  - 当前系统仅支持设置API权限。
-

## 8 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表所示：

表 8-1 控制柜命名规则

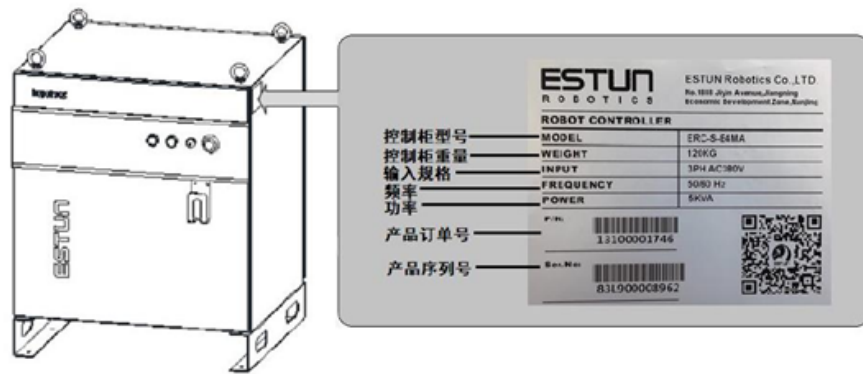
控制柜系列代号	规格	CPU型号	驱动规格	设计顺序	控制柜版本	定制类型
ERC3: 第3代控制柜	C1: 紧凑型C1	A2: ARM 控制器版本, NGC软件	A4: 小功率4轴版本	A	缺省: 标准版本	缺省: 标准版本
	C2: 紧凑型C2	-	B4: 中功率4轴版本	B	INT: CE、UL版本	C: 定制CLOOS版本
	S1: 标准型S1		C4: 大功率4轴版本	C	ESS: 功能安全版本	E: 定制EWAS版本
	S1A: 预留附加轴安装空间S1		A6: 小功率6轴版本	D	CE: CE版本	N: 无LOGO版本
	S2A: 预留附加轴安装空间S2		B6: 中功率6轴版本	E	UL: UL版本	DM: 打磨专用版本
	S3A: 预留附加轴安装空间S3		T9: 1.2T 机器人版本	-	FCE: 功能安全CE版本	HA: 高精版本
	-		-		FNA: 功能安全ETL	PV: PV机型

1. 找到控制柜的铭牌。
2. 根据铭牌信息，参考上表中内容，确定控制柜型号。若想了解控制柜详细信息，可查找对应系列控制柜使用说明书。
3. 若有疑问或无法判断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。

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。



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

1. 检查示教器是否正常启动。



2. 检查示教器、控制柜以及外部急停按钮是否被按下。



3. 检查驱动器是否有报警。
4. 若控制柜面板有“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。

本章节以 ER20-1780 机器为例。


1. 登录埃斯顿官网，选择“服务支持 -> 资料下载”。



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》，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。

1. 准备一个 FAT32 格式的 U 盘，插到示教器的 USB 接口。
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。
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。
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。

1. 先找到触发的急停按钮，并解除。



2. 若控制柜面板有“启动”按钮，需按下。




3. 清除示教器的报警提示。
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

### 问题原因

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

1. 按  进入“用户管理”界面。



2. 在“用户选择”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 等。

## 8.9 如何正常解决使用中机器人运行的点位发生了偏移

### 解决方法

1. 检查机器人零点是否正常。
2. 检查夹具是否形变。
3. 检查线体是否移位。
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

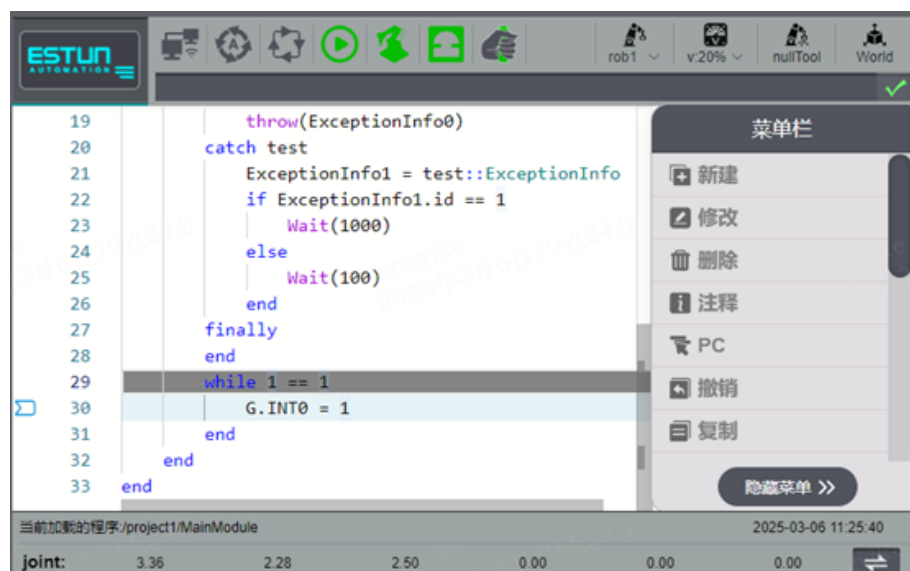
### 问题原因

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。

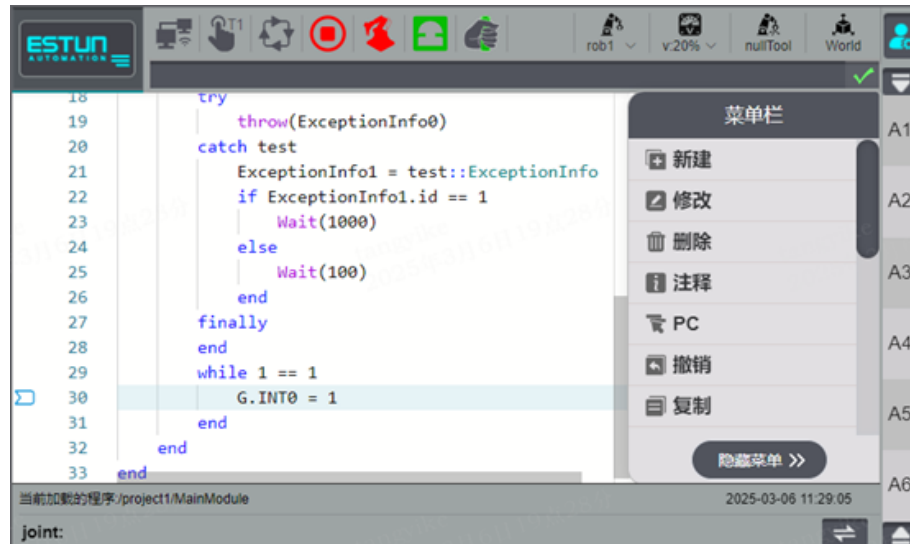
### 解决方法

1. 通过示教器状态栏，检查机器人是否处于运行状态。

- 机器人处于运行状态



- 机器人处于停止状态



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

### 问题原因

系统异常或示教器设置的 IP 与控制器不匹配，都可能造成示教器加载进度条卡在某个状态，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。

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



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。

## 附录 A ModbusTCP 控制接口数据表

Rob作为ModBusTcp通信的Server，定义了大小为3000个字节的数据寄存器区域（即MB-DataBuffer : ARRAY [0..1499] OF WORD;）。

其中，MBDataBuffer[0]~MBDataBuffer[50]为Server的发送区域（即外部PLC需要从Rob获取相关状态信息，只读）；

其中，MBDataBuffer[51]~MBDataBuffer[99]为Server的接收区域（即外部PLC需要对Rob写入相关操作指令和信息）。

其中，MBDataBuffer[100]~MBDataBuffer[1499]为寄存器通用读写区域（可在Multi-prog和指令中读写）。

类型	本地地址	寄存器地址	定义	说明	注释
-	MBData-Buffer[0]	40001	心跳检测位	用于通讯连接检测	该值由1到256循环
-	MBData-Buffer[1]	40002	当前全局速比	-	值为百分比，即[1,100]
-	MBData-Buffer[2]	40003	写权限应答	返回服务器收到的40100寄存器的值	
Send	MBData-Buffer[3]	40004	Rob状态信息	<b>bit0</b> : 手动操作模式 <b>bit1</b> : 自动操作模式 <b>bit2</b> : 远程操作模式 <b>bit3</b> : 使能状态 <b>bit4</b> : 运行状态 <b>bit5</b> : 错误状态 <b>bit6</b> : 程序运行状态 <b>bit7</b> : 机器人正在动作	-
	MBData-Buffer[4]	40005	当前加载工程名	20个字节	例如加载的工程文件名为es-tun，则各寄存器的值如下： [4]0x7365 [5]0x7574 [6]0x006E
	MBData-Buffer[5]	40006			
	MBData-Buffer[6]	40007			
	MBData-Buffer[7]	40008			
	MBData-Buffer[8]	40009			
	MBData-Buffer[9]	40010			

类型	本地地址	寄存器地址	定义	说明	注释
	MBData-Buffer[10]	40011			
	MBData-Buffer[11]	40012			
	MBData-Buffer[12]	40013			
	MBData-Buffer[13]	40014			
	MBData-Buffer[14]	40015	SimDO[0-15]	DO 0-15	-
	MBData-Buffer[15]	40016	Sim-DO[16-31]	DO 16-31	-
	MBData-Buffer[16]	40017	Sim-DO[32-47]	DO 32-47	-
	MBData-Buffer[17]	40018	Sim-DO[48-63]	DO 48-63	-
	MBData-Buffer[18]	40019	Rob执行命令状态	<b>bit0:</b> 命令为0 <b>bit2:</b> 启动命令执行成功 <b>bit3:</b> 停止命令执行成功 <b>bit4:</b> 复位命令执行成功 <b>bit7:</b> 加载工程命令成功 <b>bit8:</b> 注销工程命令成功 <b>bit9:</b> 设置全局速度成功 <b>bit10:</b> 等待控制权 <b>bit11:</b> 等待命令 <b>bit12:</b> 等待命令执行完成 <b>bit13:</b> 命令执行错误 <b>bit15:</b> 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
	MBData-Buffer[19]	40020			
	.....	-	SimAO[0-15]	AO 0-15	每两个寄存器对应一个模拟量点(浮点型)
	MBData-Buffer[50]	40051			

类型	本地地址	寄存器地址	定义	说明	注释	
Re-ceive	MBData-Buffer[51]	40052	Rob操作指令	<p><b>bit2</b> (0→0x4) : 机器人程序启动</p> <p><b>bit3</b> (0→0x8) : 机器人程序停止</p> <p><b>bit4</b> (0→0x10) : 机器人错误复位</p> <p><b>bit7</b> (0→0x80) : 加载工程文件</p> <p><b>bit8</b> (0→0x100) : 注销当前工程文件</p> <p><b>bit9</b> (0→0x200) : 设置全局速度</p> <p><b>bit10</b> (0→0x400) : 指令状态机重置</p>	<p>机器人必须处于远程励磁状态, 所有命令均上升沿触发, 配合读写标志位40100为0x11 (命令状态位40019为0x801时可下发指令)</p> <p><b>特别: 当指令执行失败后或状态机出错时, 需利用bit10来重置指令执行器, 才可以再下发新的指令</b></p>	
	MBData-Buffer[52]	40053	全局速度值	-	-	
	MBData-Buffer[53]	40054	设置工程名	20个字节		
	MBData-Buffer[54]	40055				
	MBData-Buffer[55]	40056				
	MBData-Buffer[56]	40057				
	MBData-Buffer[57]	40058				
	MBData-Buffer[58]	40059				
	MBData-Buffer[59]	40060				
	MBData-Buffer[60]	40061				
	MBData-Buffer[61]	40062	SimDI[0-15]	DI 0-15		
	MBData-Buffer[62]	40063				
	MBData-Buffer[63]	40064	SimDI[16-31]	DI 16-31		
	MBData-Buffer[64]	40065				

类型	本地地址	寄存器地址	定义	说明	注释
	MBData-Buffer[65]	40066	SimDI[32-47]	DI 32-47	-
	MBData-Buffer[66]	40067	SimDI[48-63]	DI 48-63	-
	MBData-Buffer[67]	40068	SimAI[0-15]	AI 0-15	每两个寄存器对应一个模拟量点（浮点型）
	.....				
	MBData-Buffer[98]	40099			
-	MBData-Buffer[99]	40100	读写标志位		0x11打开rob命令下发权限
General	MBData-Buffer[100]	40101	用户用（寄存器值已开放到内部PLC）		读写权限
	...	...			
	...	...			
	...	...			
	MBData-Buffer[1499]	41500			

## 附录 B 系统保留名称

类别	名称合集
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、BOOLTWOARRAY、INTTWOARRAY、REALTWOARRAY、BOOLTHREEARRAY、INTTHREEARRAY、REALTHREEARRAY、APOS、CPOS、TOOL、USERCOOR、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、BOOLRETAIN、INTRETAIN、FLOATRETAIN、STRINGRETAIN、EXCEPTIONINFO、RETURNVALUE

类别	名称合集
指令名称	<p>MovJ、MovL、MovC、MovCircle、MovJRel、MovLRel、Mov- LOffset、MovJSearch、MovLSearch、MovLW、MovCW、Mov- CircleW、MovH、MovArch、OnDistance、OnParameter、LA- BEL、GOTO、ENDIF、THEN、IF、THEN、ELSIF、ELSE、END- WHILE、DO、WHILE、CALL、RUN、KILL、RETURN、/*...*/、...=...、Set- DO、SetAO、SetSimDO、SetSimAO、WaitDI、WaitDI8421、Wait- AI、WaitSimDI、WaitSimDI8421、WaitSimAI、PulseOut、PulseSim- Out、GetDI8421、GetSimDI8421、SetDO8421、SetSimDO8421、Get- SimDItoVar、SetSimDOByVar、GetSimAItoVar、SetSimAOByVar、Set- DIEdge、SetSimDIEdge、Wait、WaitFinish、WaitCondition、GetCu- rAPos、GetCurCPos、APosToCPos、CPosToAPos、CPosToCPos、Tool- Offset、UserOffset、WaitWObj、MovLSync、MovJSyncQuit、Mov- LSyncQuit、SetTargetPos、SetSyncoord、SynCToUserC、WaitConv- Dis、SimConveyorOn、SimConveyorOff、ReceiveWObj、ResetWObj- Buf、BitAnd、BitNeg、BitOr、BitXOr、BitLSH、BitRSH、CLKStart、CLKS- top、CLKRead、CLKReset、AreaActivate、AreaDeactivate、Polyhe- dronAreaActivate、PolyhedronAreaDeactivate、TrigCam、WaitFin- ishCAM、GetCamPos、SendMessage、Tracking、PalletReset、Pallet- ToPut、PalletFromPut、PalletToGet、PalletFromGet、SetMatrix、Get- Matrix、SocketCreate、SocketClose、SocketSendStr、SocketSend- Real、SocketSendInt、SocketReadReal、SocketReadInt、SocketRead- Str、GetModConState、ReadModbusReg、WriteModbusReg、Soft- FloatStart、SoftFloatStop、ItoStr、RtoStr、StrToI、StrToR、APosTo- Str、CPosToStr、TranStrToInt、TranStrToReal、TranStrToApos、Tran- StrToCpos、MA_ELF_START、MA_ELF_STOP、SetTool、SetCoord、Se- tExternalTCP、SetPositioner、SetPayload、SetJointDyn、SetCart- Dyn、SetBlendParam、SetOverride、SetSingularPass、SetColliEn- able、SetAxisColliParam、GetCurOverride、Hand、CalcTool、CalcCo- ord、SetRtToErr、SetRtInfo、SetRtWarning、Stop、RefRobotAxis、Set- MotionMode、AutoGainEnable、AutoGainDisable、SetAxisVibraBLev- el、SetRestorePC</p>
特定名称	POSCFG、PLACEOPTIONS

For a Better Life  
让生活更美好！



埃斯顿自动化股份有限公司  
ESTUN AUTOMATION CO., LTD.

📍 南京市江宁经济开发区吉印大道1888号

☎ 025-52785866

🏠 [www.estun.com](http://www.estun.com)

☎ 025-52785576

✉ [info@estun.com](mailto:info@estun.com)