

**For a Better Life**  
让生活更美好!



# iER系列

## 指令系统使用手册

iEPM01V30402-01



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。

## 前言

### 资料简介

本手册适用于控制系统V3.04版本，介绍工业机器人指令系统中结构定义、变量类型、参数含义，有助于使用者深刻了解指令系统。

### 修订记录

修订日期	发布版本	变更内容
2026.02	01	新建手册。

### 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。

## 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“O”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	O	O	O

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，必须仔细阅读标示对应的说明，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；
  - 机器人处于手动模式时，运行速度需限制在 250mm/s 以下；若需调至手动全速度，仅允许充分了解风险的专业人员操作；

- 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。

### 注意

机器人作业现场及控制柜周边，必须配备二氧化碳灭火器（≥2kg），确保灭火器放置在易取用、无遮挡位置。

#### 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。否则可能导致机器人产生意外动作，引发人身伤害或设备损坏。
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
	<p>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。</p>

---

**▲ 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：若当前为使能断开状态，按下按钮后将会切换到使能接通状态；反之，当前为使能接通状态，按下按钮后则会切换到使能断开状态。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：

- 确保使能按钮始终处于有效状态，避免应急时无法正常操作。
- 暂停机器人运行、编程或测试期间，需及时断开使能，防止误触发。
- 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/液压系统安全注意事项：关闭气源或液压泵后，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；维修气动或液压元件前，必须确保系统内残留能量已完全释放。

---

### 注意

为防止气动/液压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  $40-60^{\circ}\text{C}$ ），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  $60^{\circ}\text{C}$ ），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、液压千斤顶等专业承重设备，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。

---

## 目录

申明.....	2
前言.....	3
安全使用须知.....	4
图目录.....	21
1 系统介绍.....	22
1.1 系统简介.....	22
1.2 系统域数据文件.....	22
1.2.1 系统域ZONE预定义常量.....	23
1.2.2 系统域AREA预定义变量.....	23
1.2.3 系统域TOOL预定义变量.....	24
1.2.4 系统域FRAME预定义变量.....	24
1.2.5 系统域PAYLOAD预定义变量.....	24
1.2.6 系统域APOS预定义变量.....	25
1.2.7 系统\$\$SYS_DATA预定义变量.....	25
1.2.8 系统域SPEED预定义常量.....	26
1.3 全局域数据文件.....	26
1.4 工程文件.....	26
1.4.1 工程域数据文件.....	26
1.4.2 模块文件.....	27
1.4.3 函数域数据文件.....	27
1.5 系统关键字（词）.....	27
2 变量数据类型.....	28
2.1 基本数据类型.....	28
2.1.1 Bool.....	28
2.1.2 Int.....	28
2.1.3 Float.....	29
2.1.4 String.....	29
2.1.5 BoolOneArray.....	29
2.1.6 IntOneArray.....	29
2.1.7 FloatOneArray.....	30
2.1.8 BoolTwoArray.....	30
2.1.9 IntTwoArray.....	30
2.1.10 FloatTwoArray.....	31
2.1.11 BoolThreeArray.....	31
2.1.12 IntThreeArray.....	31

2.1.13 FloatThreeArray.....	32
2.2 系统预留类型.....	32
2.2.1 AREA.....	32
2.2.2 TOOL.....	34
2.2.3 LoadDyn.....	34
2.2.4 CenterPos.....	35
2.2.5 InertiaTensor.....	36
2.2.6 FRAME.....	36
2.2.7 PAYLOAD.....	37
2.3 位置数据类型.....	38
2.3.1 Apos.....	38
2.3.2 Cpos.....	38
2.3.3 Poscfg.....	39
2.3.4 DApos.....	42
2.3.5 DCpos.....	43
2.4 系统数据类型.....	44
2.4.1 Speed.....	44
2.4.2 Zone.....	44
2.4.3 LsScale.....	45
2.4.4 LsThresh.....	45
2.4.5 ColliParam.....	45
2.4.6 VibrationParam.....	46
2.5 摆动数据类型.....	46
2.5.1 Weave.....	46
2.6 时钟数据类型.....	47
2.6.1 Clock.....	47
2.7 码垛数据类型.....	48
2.7.1 Pallet.....	48
2.8 Socket数据类型.....	49
2.8.1 TcpServer.....	49
2.8.2 TcpSocket.....	49
2.8.3 UdpSocket.....	49
2.9 掉电保持数据类型.....	50
2.9.1 BoolRetain.....	50
2.9.2 FloatRetain.....	50
2.9.3 StringRetain.....	50
2.10 异常数据类型.....	50
2.10.1 ExceptionInfo.....	50

2.11 返回数据类型.....	51
2.11.1 ReturnValue.....	51
2.12 中断数据类型.....	51
2.12.1 INTERRUPT.....	51
3 指令.....	52
3.1 运动指令.....	52
3.1.1 MovJ.....	52
3.1.2 MovL.....	55
3.1.3 MovC.....	57
3.1.4 MovCircle.....	60
3.1.5 MovJRel.....	62
3.1.6 MovLRel.....	65
3.1.7 MovJSearch.....	67
3.1.8 MovLSearch.....	70
3.1.9 MovJOffset.....	73
3.1.10 MovLOffset.....	75
3.1.11 MovJOffsetC.....	77
3.1.12 MovLW.....	79
3.1.13 MovCW.....	81
3.1.14 MovCircleW.....	83
3.1.15 OnDistance.....	85
3.1.16 OnParameter.....	87
3.1.17 MovJArch.....	88
3.1.18 MovLArch.....	90
3.1.19 MovH.....	93
3.2 控制指令.....	95
3.2.1 LABEL.....	96
3.2.2 GOTO.....	96
3.2.3 if.....	96
3.2.4 elseif.....	97
3.2.5 else.....	97
3.2.6 while.....	97
3.2.7 for.....	98
3.2.8 switch.....	98
3.2.9 case.....	98
3.2.10 break.....	99
3.2.11 default.....	99
3.2.12 continue.....	100

3.2.13 try.....	100
3.2.14 catch.....	100
3.2.15 finally.....	101
3.2.16 return.....	101
3.2.17 ...=.....	101
3.2.18 #=...=#.....	101
3.3 等待指令.....	102
3.3.1 Wait.....	102
3.3.2 WaitFinish.....	102
3.3.3 WaitCondition.....	102
3.4 IO指令.....	103
3.4.1 PulseOut.....	103
3.4.2 WaitIO.....	104
3.4.3 WaitIO8421.....	105
3.5 设置指令.....	106
3.5.1 SetJointDyn.....	107
3.5.2 SetCartDyn.....	107
3.5.3 SetBlendParam.....	108
3.5.4 SetPayload.....	109
3.5.5 SetOverRide.....	109
3.5.6 SetColliEnable.....	110
3.5.7 SetAxisColliParam.....	111
3.5.8 GetCurOverRide.....	111
3.5.9 CalcTool.....	112
3.5.10 CalcFrame.....	113
3.5.11 SetRtToErr.....	113
3.5.12 SetRtInfo.....	114
3.5.13 SetRtWarning.....	114
3.5.14 Stop.....	115
3.5.15 SetMotionMode.....	115
3.5.16 AutoTune.....	116
3.5.17 AutoGainEnable.....	116
3.5.18 AutoGainDisable.....	117
3.5.19 SetRestorePC.....	117
3.5.20 SetOriMode.....	118
3.5.21 SetCircMode.....	119
3.5.22 GetRobotIsMoving.....	119
3.5.23 SetColliAutoTune.....	120

3.5.24 SetColliParam.....	121
3.5.25 SetAxisVibraBLevel.....	122
3.5.26 SetVibrationParam.....	123
3.5.27 RecordUserInfo.....	124
3.5.28 SetWristMoveEnable.....	124
3.5.29 SetWristMoveDisable.....	124
3.6 位置运算指令.....	125
3.6.1 GetCurAPos.....	125
3.6.2 GetCurCPos.....	126
3.6.3 APosToCPos.....	126
3.6.4 CPosToAPos.....	127
3.6.5 CPosToCPos.....	128
3.6.6 ToolOffset.....	128
3.6.7 FrameOffset.....	129
3.6.8 CalPosOffset.....	130
3.6.9 PosOffset.....	130
3.7 时钟指令.....	131
3.7.1 CLKStart.....	132
3.7.2 CLKStop.....	132
3.7.3 CLKRead.....	132
3.7.4 CLKReset.....	133
3.8 区域指令.....	133
3.8.1 AreaActivate.....	133
3.8.2 AreaDeactivate.....	134
3.8.3 AreaConnectIO.....	134
3.8.4 AreaDisConnectIO.....	135
3.9 码垛指令.....	135
3.9.1 PalletReset.....	136
3.9.2 PalletToPut.....	136
3.9.3 PalletFromPut.....	139
3.9.4 PalletToGet.....	141
3.9.5 PalletFromGet.....	142
3.10 Socket指令.....	144
3.10.1 SocketCreateServer.....	144
3.10.2 SocketAccept.....	145
3.10.3 SocketCreateClient.....	146
3.10.4 SocketReceive.....	146
3.10.5 SocketSend.....	147

3.10.6 SocketClose.....	147
3.10.7 UDPCreate.....	148
3.10.8 UDPBind.....	148
3.10.9 UDPReceive.....	149
3.10.10 UDPSend.....	149
3.11 传送带跟随指令.....	151
3.11.1 MovLSync.....	151
3.11.2 MovJSyncQuit.....	153
3.11.3 MovLSyncQuit.....	155
3.11.4 WaitWObj.....	156
3.11.5 SetTargetPos.....	157
3.11.6 SynCToUserC.....	158
3.11.7 WaitConvDis.....	158
3.11.8 SimConveyorOn.....	159
3.11.9 SimConveyorOff.....	159
3.11.10 ReceiveWObj.....	159
3.11.11 ResetWObjBuf.....	160
3.11.12 SyncToNextRob.....	160
3.12 数学运算函数.....	160
3.12.1 sin.....	161
3.12.2 cos.....	161
3.12.3 tan.....	161
3.12.4 asin.....	162
3.12.5 acos.....	162
3.12.6 atan.....	162
3.12.7 atan2.....	162
3.12.8 sinh.....	162
3.12.9 cosh.....	162
3.12.10 tanh.....	163
3.12.11 log.....	163
3.12.12 log10.....	163
3.12.13 sqrt.....	163
3.12.14 exp.....	163
3.12.15 pow.....	164
3.12.16 deg.....	164
3.12.17 rad.....	164
3.12.18 fmod.....	164
3.12.19 floor.....	164

3.12.20 random.....	165
3.13 字符串指令.....	165
3.13.1 byte.....	166
3.13.2 char.....	166
3.13.3 find2.....	166
3.13.4 findEnd.....	166
3.13.5 format.....	166
3.13.6 getAt.....	167
3.13.7 gsub.....	167
3.13.8 len.....	168
3.13.9 left.....	168
3.13.10 lower.....	168
3.13.11 right.....	168
3.13.12 reverse.....	168
3.13.13 strcmp.....	169
3.13.14 trimLeft.....	169
3.13.15 trimRight.....	169
3.13.16 upper.....	169
3.13.17 IToStr.....	169
3.13.18 RToStr.....	169
3.13.19 StrTol.....	170
3.13.20 StrToR.....	170
3.13.21 StrToIntArray.....	170
3.13.22 StrToFloatArray.....	170
3.13.23 IntArrayToStr.....	171
3.13.24 FloatArrayToStr.....	171
3.13.25 APosToStr.....	172
3.13.26 CPosToStr.....	172
3.13.27 TranStrToInt.....	172
3.13.28 TranStrToReal.....	173
3.13.29 TranStrToAPos.....	173
3.13.30 TranStrToCPos.....	173
3.14 ModbusTcp指令.....	174
3.14.1 GetModConState.....	174
3.14.2 ReadModbusReg.....	174
3.14.3 WriteModbusReg.....	175
3.15 视觉指令.....	175
3.15.1 TrigCam.....	175

3.15.2 WaitFinishCam.....	176
3.15.3 GetCamPos.....	176
3.15.4 SendMessage.....	176
3.16 阵列指令.....	177
3.16.1 SetMatrix.....	177
3.16.2 GetMatrix.....	177
3.17 中断指令.....	178
3.17.1 IConnect.....	178
3.17.2 IDelete.....	178
3.17.3 ISignalDI.....	179
3.17.4 ISignalSimDI.....	179
3.17.5 ExitCycle.....	180
3.17.6 MovToIPos.....	180
3.17.7 ISleep.....	180
3.17.8 IWatch.....	180
3.18 交互式指令.....	181
3.18.1 DialogPanelInt.....	182
3.18.2 DialogPanelFloat.....	182
3.18.3 DialogPanelString.....	182
3.18.4 DialogPanelList.....	183
3.18.5 DialogPanelMessage.....	184
3.19 多机协同指令.....	184
3.19.1 Gmove.....	184
3.20 任务指令.....	185
3.20.1 run.....	186
3.20.2 kill.....	186
3.21 异常指令.....	186
3.21.1 throw.....	186
3.22 负载在线辨识指令.....	187
3.22.1 StartMdlLog.....	187
3.22.2 StopMdlLog.....	188
3.22.3 CalcDynMdl.....	188
3.23 飞拍指令.....	189
3.23.1 LatchEnable.....	189
3.23.2 GetPSOStatus .....	189
3.23.3 RecordLatchPos.....	190
3.23.4 SetLatchPos.....	190
3.24 ERI指令.....	191

---

3.24.1 SetERIProgID.....	191
3.24.2 EnableERIControl.....	191
3.24.3 SetERIParam.....	192
3.24.4 SetCorrParam.....	192
3.24.5 EnableERICorrect.....	193
3.25 软浮动指令.....	193
3.25.1 SoftFloatStart.....	193
3.25.2 SoftFloatStop.....	194
3.26 样条曲线指令.....	194
3.26.1 GetPosFromFile.....	194
3.26.2 OnSplineIndex.....	196
3.26.3 SetSplineParam.....	196
3.26.4 MovS.....	197

## 图目录

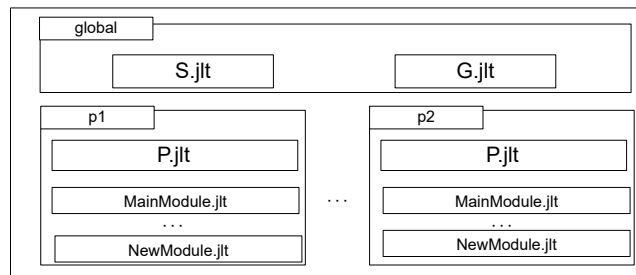
图 2-1 工具&负载质量信息坐标系说明.....	35
图 2-2 工具&负载质量信息坐标系实例.....	35
图 3-1 轨迹示意图.....	55
图 3-2 MovC圆弧关键点示意.....	59
图 3-3 MovCircle整圆关键点示意.....	62
图 3-4 MovH螺旋运动关键点示意.....	95

# 1 系统介绍

## 1.1 系统简介

指令系统文件结构分为工程、模块和数据三部分，其中数据分为四个作用域：系统域、全局域、工程域、函数域。程序代码以.jlt为后缀，统一存储在project文件夹中。

project文件结构：



- 系统域数据文件 S.jlt
- 全局域数据文件 G.jlt
- 工程文件夹P1中
  - 工程域数据文件P.jlt
  - 程序指令文件MainModule.jlt、NewModule.jlt
  - 函数域数据文件MainModule.jlt、NewModule.jlt

### ⚠ 注意

系统域、全局域、工程域及主模块数据文件名称为系统指定，不可修改。其他新建模块名可以自定义。

用户可以通过示教器或其他接口编辑模块指令文件。

## 1.2 系统域数据文件

project/global目录下S.jlt为系统数据文件。变量对应系统域，为预定义，其中Speed和Zone类型变量用户不可编辑，支持所有工程引用。该文件主要包括预定义的八种类型变量：

- Speed：系统预留速度变量，支持用户在其他域自定义该类型变量；
- Zone：系统预留过渡变量，支持用户在其他域自定义该类型变量；
- AREA：系统预留区域变量，共10个，仅支持选择使用，不能在其他域新建该类型变量；

- TOOL：系统预留工具坐标系变量，含nullTool在内共11个，仅支持选择使用，不能在其他域新建该类型变量；
- FRAME：系统预留用户坐标系变量，含World在内共11个，仅支持选择使用，不能在其他域新建该类型变量
- PAYLOAD：系统预留工件负载变量，含FullPayload在内共11个，仅支持选择使用，不能在其他域新建该类型变量；
- APOS：系统预留关节位置变量，共1000个，支持用户在其他域自定义该类型变量；
- CPOS：系统预留坐标系位置变量，共1000个，支持在其他域自行定义该类型变量。

### 1.2.1 系统域ZONE预定义常量

系统预定义的Zone类型变量，具体如下表所示：

名称	转弯区百分比(per):/%	转弯区距离(dis):mm	转弯区速度恒定(vConst)
C0	0.3	0.3	0
C5	5	5	0
C10	10	10	0
C20	20	20	0
C30	30	30	0
C50	50	50	0
C60	60	60	0
C80	80	80	0
C100	100	100	0
C150	100	150	0
C200	100	200	0

### 1.2.2 系统域AREA预定义变量

系统预定义的Area类型变量，具体如下表所示：

名称	id	is_default	act_state	is_in_area	auto_act_state	type	shape	...
AREA[1]	1	0	0	0	0	0	0	...
AREA[2]	2	0	0	0	0	0	0	...
AREA[3]	3	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...
AREA[10]	10	0	0	0	0	0	0	...

**⚠ 注意**

Area类型变量的参数需要在区域监控界面中查看，而不是在变量界面上。

### 1.2.3 系统域TOOL预定义变量

系统预定义的Tool类型变量，具体如下表所示：

名称	id	x	y	z	a	b	c	M	Mx	My	Mz	lxx	lyy	lzz	lxy	lyz	lxz
nullTool	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOOL[1]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOOL[2]	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
TOOL[10]	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

### 1.2.4 系统域FRAME预定义变量

系统预定义的Frame类型变量，具体如下表所示：

名称	type	id	x	y	z	a	b	c
World	0(用户坐标系)	0	0	0	0	0	0	0
FRAME[1]	0(用户坐标系)	1	0	0	0	0	0	0
FRAME[2]	0(用户坐标系)	2	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...
FRAME[10]	0(用户坐标系)	10	0	0	0	0	0	0

### 1.2.5 系统域PAYLOAD预定义变量

系统预定义的Payload类型变量，具体如下表所示：

名称	id	M(kg)	Mx	My	Mz	lxx	lyy	lzz	lxy	lyz	lxz
FullPayload	0	0	0	0	0	0	0	0	0	0	0
PAYLOAD[1]	1	0	0	0	0	0	0	0	0	0	0
PAYLOAD[2]	2	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
PAYLOAD[10]	10	0	0	0	0	0	0	0	0	0	0

## 1.2.6 系统域APOS预定义变量

系统预定义的Apos类型变量，具体如下表所示：

名称	a1	a2	a3	a4	a5	a6
APOS[1]	0	0	0	0	0	0
APOS[2]	0	0	0	0	0	0
APOS[3]	0	0	0	0	0	0
...	...	...	...	...	...	...
APOS[1000]	0	0	0	0	0	0

## 1.2.7 系统\$\$SYS\_DATA预定义变量

系统\$\$SYS\_DATA类型变量集合存放在runtime/cfg\_default/erc/base/lang/System-Variable.json当中，其实现存放在runtime/system/rob\_base/Base\_Dollar.jlt，具体如下表所示：

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

## 1.2.8 系统域SPEED预定义常量

系统预定义的Speed类型变量，具体如下表所示：

名称	关节速度百分比(per): %	TCP线速度(tcp):mm/s	空间旋转速度(ori):deg/s	外部轴线速度(exj_l):mm/s	外部轴角速度(exj_r):deg/s
V5	1	5	5	5	5
V10	2	10	10	10	10
V30	3	30	30	30	30
V50	5	50	50	50	50
V60	6	60	60	60	60
V80	8	80	80	80	80
V100	10	100	100	100	100
V200	20	200	200	200	200
V300	30	300	300	300	300
V500	50	500	500	500	500
V800	80	800	500	800	800
V1000	100	1000	500	1000	1000
V1500	100	1500	500	1500	1000
V2000	100	2000	500	2000	1000
V3000	100	3000	500	2000	1000
V4000	100	4000	500	2000	1000

## 1.3 全局域数据文件

project/global目录下G.jlt为全局数据文件。变量对应全局域，用户可编辑。全局域下的数据可在所有工程中引用。

## 1.4 工程文件

### 1.4.1 工程域数据文件

每次新建工程，都会在project目录下生成一个以工程名命名的文件夹，例如p1。在工程名文件夹下，P.jlt为工程数据文件。变量对应工程域，用户可编辑。工程域下的数据只可在本工程下所有程序指令中引用。

## 1.4.2 模块文件

在工程名文件夹下，默认生成MainModule.jlt文件，代表一个模块，该文件中默认会生成一个main函数，用户可以在函数中编辑指令，如后续章节中介绍的Movj等。

用户还可以在模块中新建自定义的函数，统一模块的所有函数都放在同一文件中，另外，用户也可以新建自定义的模块，每新建一个模块就会新建一个后缀为jlt的文件。

## 1.4.3 函数域数据文件

程序域变量数据和指令一样，都放在函数，所以函数域数据文件与模块文件是同一个。但是，函数域的变量只能在函数中引用。

## 1.5 系统关键字（词）

---

### 注意

以下“名称”为系统保留，主要包含语言关键字、变量类型以及指令名称等。用户不可创建同名的变量。

---

#### 语言关键字：

export、struct、end、Any、function、var、@noview、@label、@foreign、@goto、module、include、const、return、if、else、elseif、while、for、switch、case、default、print、println、false、true、try、catch、finally、continue

## 2 变量数据类型

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。
- 全局域：基本数据类型、位置数据类型、系统数据类型、socket数据类型、时钟数据类型、码垛数据类型、摆动数据类型、异常数据类型、掉电保持数据类型、中断数据类型、返回数据类型。
- 工程域：基本数据类型、位置数据类型、系统数据类型、摆动数据类型、异常数据类型、返回数据类型、socket数据类型。
- 函数域：基本数据类型、位置数据类型、系统数据类型、摆动数据类型、异常数据类型、返回数据类型、socket数据类型。

### 2.1 基本数据类型

#### 2.1.1 Bool

存储Bool型数据。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）。

参数说明：

参数名	数据类型	参数含义
value	Bool	变量值
comment	String	变量注释

#### 2.1.2 Int

存储整数。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）。

参数说明：

参数名	数据类型	参数含义
value	Int	变量值
comment	String	变量注释

### 2.1.3 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）。

参数说明：

参数名	数据类型	参数含义
value	Float	变量值
comment	String	变量注释

### 2.1.4 String

存储字符串。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）。

参数说明：

参数名	数据类型	参数含义
value	String	变量值
comment	String	变量注释

### 2.1.5 BoolOneArray

存储Bool型数组数据。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）。

参数说明：

参数名	数据类型	参数含义
count	Int	数据个数
value	Bool	变量值

### 2.1.6 IntOneArray

存储整数数组数据。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）。

参数说明：

参数名	数据类型	参数含义
count	Int	数据个数

参数名	数据类型	参数含义
value	Int	变量值

### 2.1.7 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）。

参数说明：

参数名	数据类型	参数含义
count	Int	数据个数
value	Float	变量值

### 2.1.8 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）。

参数说明：

参数名	数据类型	参数含义
row	Int	行数
column	Int	列数
value	Bool	变量值

### 2.1.9 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）。

参数说明：

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

## 2.1.10 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）。

参数说明：

参数名	数据类型	参数含义
row	Int	行数
column	Int	列数
value	Float	变量值

## 2.1.11 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）。

参数说明：

参数名	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

## 2.1.12 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）。

参数说明：

参数名	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

### 2.1.13 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）。

参数说明：

参数名	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

## 2.2 系统预留类型

### 2.2.1 AREA

存储标准区域信息的值，系统预定义变量。

参数说明：

参数名	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
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

参数名	数据类型	参数含义
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_enb	Bool	区域形状为轴组时, 依次表示对应轴组的使能信息 (系统最大支持10个轴组)
joint_group3_max ~ joint_group10_max	Float	区域形状为轴组时, 依次表示对应轴组的最大限位 信息 (系统最大支持10个轴组)
joint_group3_min ~ joint_group10_min	Float	区域形状为轴组时, 依次表示对应轴组的最小限位 信息 (系统最大支持10个轴组)
ploy_vertex_num	Int	多边形区域的顶点数量
ploy_vertex_Z	Float	多边形区域的顶点z值
ploy_vertex_H	Float	多边形区域的高度
ploy_vertex_X1	Float	多边形区域的顶点x值
ploy_vertex_Y1	Float	多边形区域的顶点y值
ploy_vertex_X2	Float	多边形区域的顶点x值
ploy_vertex_Y2	Float	多边形区域的顶点y值
ploy_vertex_X3 ~ ploy_vertex_X10	Float	依次表示多边形区域的顶点x值
ploy_vertex_Y3 ~ ploy_vertex_Y10	Float	依次表示多边形区域的顶点y值

## 2.2.2 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。系统预定义变量。

参数说明：

参数名	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，由创建时系统自动分配
x	Float	TCP相对于法兰坐标系在x方向的位移偏移量，单位是mm
y	Float	TCP相对于法兰坐标系在y方向的位移偏移量，单位是mm
z	Float	TCP相对于法兰坐标系在z方向的位移偏移量，单位是mm
a	Float	TCP相对于法兰坐标系z轴旋转的欧拉角，单位是deg
b	Float	TCP相对于法兰坐标系y'轴旋转的欧拉角，单位是deg
c	Float	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，参数说明详见 <a href="#">LoadDyn</a> 含义

### 注意

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。

## 2.2.3 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。

参数说明：

参数名	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 <a href="#">CenterPos</a> 含义
tensor	InertiaTensor	详见 <a href="#">InertiaTensor</a> 含义

## 2.2.4 CenterPos

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

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

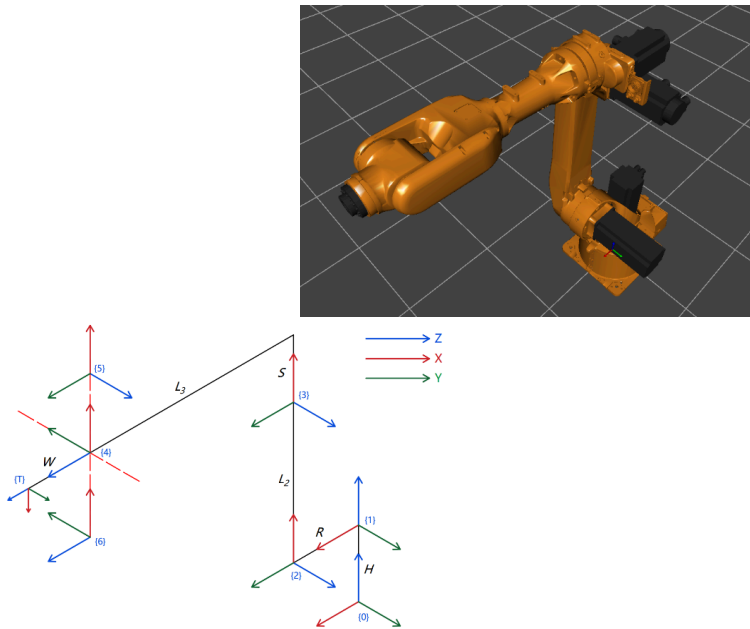


图 2-1 工具&负载质量信息坐标系说明

### ⚠ 注意

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。

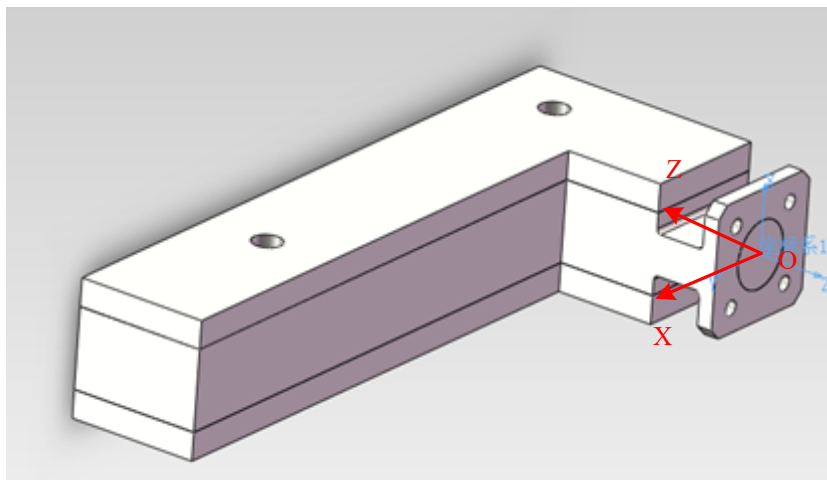


图 2-2 工具&负载质量信息坐标系实例

**⚠ 注意**

工具和负载的质量信息坐标系均参考图 2-1 工具&负载质量信息坐标系说明中坐标系 $O_{Tool}$ -XYZ。

参数说明：

参数名	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

## 2.2.5 InertiaTensor

该参数是已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CenterPos变量类型中的说明图。

参数说明：

参数名称	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方向回转的惯量矩，单位是 $kg \cdot mm^2$
lyy	Float	安装的工具或装夹的负载在重心位置的Y方向回转的惯量矩，单位是 $kg \cdot mm^2$
lzz	Float	安装的工具或装夹的负载在重心位置的Z方向回转的惯量矩，单位是 $kg \cdot mm^2$
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 $kg \cdot mm^2$
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 $kg \cdot mm^2$
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 $kg \cdot mm^2$

## 2.2.6 FRAME

FRAME型变量用来记录坐标系参数，包括用户坐标系、动坐标系和外部TCP坐标系。

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。系统预定义变量。

参数说明：

参数名称	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

### 说明

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。

## 2.2.7 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。系统预定义变量。

参数说明：

参数名称	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，由创建时系统自动分配
dyn	LoadDyn	工件负载的质量信息，用于机器人动力学全模型计算

### 说明

系统预定义了FullPayload这个工件负载，其中各个偏移量的值默认都为0。

## 2.3 位置数据类型

### 2.3.1 Apos

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

参数说明：

参数名称	数据类型	参数含义
a1	Float	关节1号轴的坐标
a2	Float	关节2号轴的坐标
a3	Float	关节3号轴的坐标
a4~a15	Float	关节4~关节15号轴的坐标
a16	Float	关节16号轴的坐标
comment	String	变量注释

#### 注意

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。

### 2.3.2 Cpos

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

参数说明：

参数名称	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	工具坐标系id
frame	Int	用户坐标系id
x	Float	TCP点在参考坐标系上x方向的坐标
y	Float	TCP点在参考坐标系上y方向的坐标
z	Float	TCP点在参考坐标系上z方向的坐标
a	Float	TCP点相对于参考坐标系z轴旋转的欧拉角
b	Float	TCP点相对于参考坐标系y'轴旋转的欧拉角

参数名称	数据类型	参数含义
c	Float	TCP点相对于参考坐标系x''轴旋转的欧拉角
a7	Float	关节7号轴的坐标
a8~a15	Float	关节8~15号轴的坐标
a16	Float	关节16号轴的坐标
comment	String	变量注释

### ⚠ 注意

机器人几个轴就显示几个轴的参数。

## 2.3.3 Poscfg

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

参数说明：

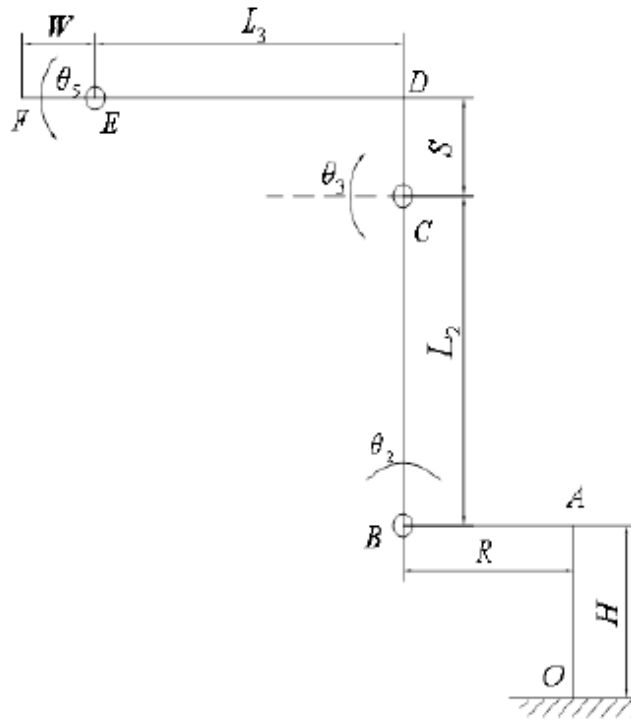
参数名称	数据类型	参数含义
mode	Int	/
cf1	Int	关节1号轴角度所在的象限取值
cf2	Int	关节2号轴角度所在的象限取值
cf3	Int	关节3号轴角度所在的象限取值
cf4	Int	关节4号轴角度所在的象限取值
cf5	Int	关节5号轴角度所在的象限取值
cf6	Int	关节6号轴角度所在的象限取值

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 $-2\pi$

1. 通用六关节运动学mode定义



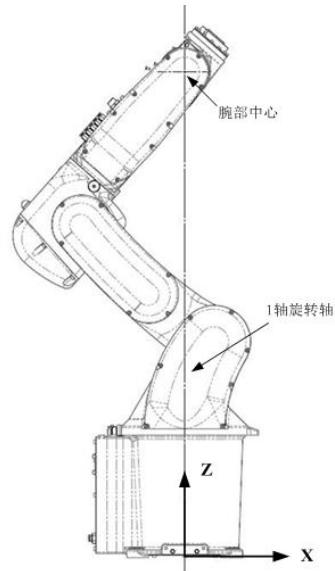
通用六关节的运动学存在八组解，定义mode值为 0~7，含义如下表：

Mode	腕部中心相对于一轴轴心的关系(flag1) 0: 在前; 1: 在后 $R+L_3 \cdot \cos(\theta_2+\theta_3)+L_2 \cdot \sin\theta_2+S \cdot \sin(\theta_2+\theta_3)$	Axis3(flag3) ( $\theta_3+90-\arctan(S/L_3)$ ) 0: [0,180] 1:(-180,0)	Axis5(flag5) ( $\theta_5$ ) 0: [0,180] 1:(-180,0)
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

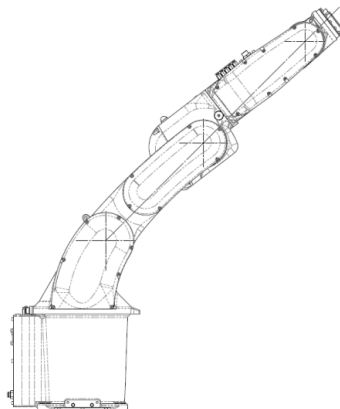
$mode = flag5 + 2*flag3 + 4*flag1。$

**⚠ 注意**

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



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



cc、Axis5(flag5)的含义：即五轴的角度。

**2. SCARA运动学mode定义**

Scara运动学存在两组解，定义mode值为0和1，含义如下表

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

mode = flag2。

### 3. 三轴平面机器人运动学+“3+1”机器人mode定义

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

mode = flag2。

### 4. 吊装SCARA运动学mode定义

吊装Scara运动学存在两组解，定义mode值为0和1，含义如下表

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

## 2.3.4 DApos

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

参数说明：

参数名称	数据类型	参数含义
da1	Float	关节1号轴角度的偏移量
da2	Float	关节2号轴角度的偏移量
da3	Float	关节3号轴角度的偏移量

参数名称	数据类型	参数含义
da4~da15	Float	关节4~关节15号轴角度的偏移量
da16	Float	关节16号轴角度的偏移量
comment	String	变量注释

### ⚠ 注意

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。

## 2.3.5 DCpos

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

参数说明：

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

### ⚠ 注意

机器人几个轴会显示几个轴的参数

## 2.4 系统数据类型

### 2.4.1 Speed

定义机器人和外部轴的运动速度。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，函数三大变量作用域中对该变量进行创建，删除，修改等操作。

参数说明：

参数名称	数据类型	参数含义
per:关节速度百分比	Float	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:TCP线速度	Float	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: 空间旋转速度	Float	定义机器人末端点姿态的旋转速度
exj_l: 外部轴线速度	Float	定义外部直线轴的运动速度
exj_r: 外部轴角速度	Float	定义外部旋转轴的运动速度
comment	String	变量注释

### 2.4.2 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，函数三大变量作用域中对该变量进行创建，删除，修改等操作。

参数说明：

参数名称	数据类型	参数含义
per: 关节轨迹转弯百分比	Float	适用于MovJ,MovJRel,MovJSearch等关节运动指令，表示距离目标角度还有多远时开始转弯
dis: 笛卡尔空间转弯区大小	Float	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: 恒速过渡使能状态	Int	<p>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。</p> <hr/> <p><b>⚠ 注意</b></p> <p>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。</p> <hr/> <p>注：</p>

参数名称	数据类型	参数含义
comment	String	变量注释

### 2.4.3 LsScale

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

参数说明：

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

### 2.4.4 LsThresh

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

参数说明：

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

### 2.4.5 ColliParam

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

参数说明：

参数名称	数据类型	参数含义
index: 碰撞自调整参数序号	Int	index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

## 2.4.6 VibrationParam

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

参数说明:

参数名称	数据类型	参数含义
index: 碰撞自调整参数序号	Int	index取值范围为[0, 6]。其中0表示选择默认振动抑制参数, 其它表示ID为index的自学习振动抑制自学习参数。

## 2.5 摆动数据类型

### 2.5.1 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, 函数三大变量作用域中对该变量进行创建, 删除, 修改等操作。

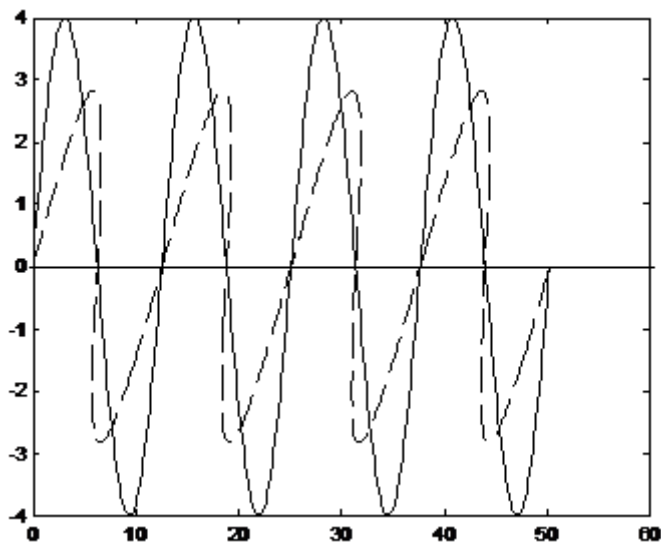
参数说明:

参数名称	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	Float	机器人摆动时的频率, 单位: Hz
amp_l	Float	机器人摆动左幅值, 单位: mm
amp_r	Float	机器人摆动右幅值, 单位: mm
stop_time_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
stop_time_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
stop_time_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
rot_angle_x	Float	摆动基准平面绕摆动行进方向旋转的角度, 该参数用于决定最终的摆动平面, 单位是deg
rot_angle_z	Float	绕摆动平面的法向量旋转的角度, 决定摆弧的形状, 单位: deg
delay_type_l	Int	波峰处停止使能, 指在摆动到左幅值时停止使能, 该值为0表示使能失效, 该值为1表示使能有效

参数名称	数据类型	参数含义
delay_type_c	Int	中间位置处停止使能，指在摆动到中间位置时停止使能，该值为0表示使能失效，该值为1表示使能有效
delay_type_r	Int	波谷处停止使能，指在摆动到右幅值时停止使能，该值为0表示使能失效，该值为1表示使能有效

**⚠ 注意**

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



## 2.6 时钟数据类型

### 2.6.1 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，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数说明：

参数名称	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

参数名称	数据类型	参数含义
value	Float	时钟变量的计数值
comment	String	变量注释

## 2.7 码垛数据类型

### 2.7.1 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，复制，粘贴等操作（其中复制、粘贴操作不能跨域和跨工程）。

参数说明：

参数名称	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	Float	垛体x方向上单个工件长度
ydistOfPart	Float	垛体y方向上单个工件长度
zdistOfPart	Float	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
preplaceOptions	Placeoptions	前置点信息
postplaceOptions	Placeoptions	后置点信息

## 2.8 Socket数据类型

### 2.8.1 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。该变量支持在全局变量作用域中创建，删除，复制，粘贴等操作。

参数说明：

参数名称	数据类型	参数含义
handle	Int	TCP Socket服务器句柄
status	Int	TCP Socket状态
ip	String	IP地址
port	Int	端口号

### 2.8.2 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。该变量支持在全局变量作用域中创建，删除，复制，粘贴等操作。

参数说明：

参数名称	数据类型	参数含义
handle	Int	TCP Socket客户端句柄
status	Int	TCP Socket状态
ip	String	IP地址
port	Int	端口号

### 2.8.3 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。该变量支持在全局变量作用域中创建，删除，复制，粘贴等操作。

参数说明：

参数名称	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

## 2.9 掉电保持数据类型

### 2.9.1 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）。

参数说明：

参数名称	数据类型	参数含义
value	Bool	掉电保持使能值
comment	String	变量注释

### 2.9.2 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）。

参数说明：

参数名称	数据类型	参数含义
value	Float	掉电保持使能值
comment	String	变量注释

### 2.9.3 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）。

参数说明：

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

## 2.10 异常数据类型

### 2.10.1 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）。

参数说明：

参数名称	数据类型	参数含义
id	Int	异常数据信息索引id
message	String	异常数据详细信息
comment	String	变量注释

## 2.11 返回数据类型

### 2.11.1 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）。

参数说明：

参数名称	数据类型	参数含义
int_value	Int	整型值
bool_value	Bool	布尔值
str_value	String	字符串值
float_value	Float	实数值
comment	String	变量注释

## 2.12 中断数据类型

### 2.12.1 INTERRUPT

存储中断标识信息，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）。

参数说明：

参数名称	数据类型	参数含义
id	Int	异常数据信息索引id。
message	String	异常数据详细信息。
comment	String	变量注释。

## 3 指令

### 3.1 运动指令

运动指令列表:

运动指令	MovJ
	MovL
	MovC
	MovCircle
	MovJRel
	MovLRel
	MovJSearch
	MovLSearch
	MovJOffset
	MovJOffsetC
	MovLOffset
	MovLOffsetC
	MovLW
	MovCW
	MovCircleW
	OnDistance
	OnParameter
	MovJArch
	MovLArch
MovH	

#### 3.1.1 MovJ

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。

**MovJ** p [v] [b] [c] [payload] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表:

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置
[v]: 目标速度 (Velo)	Speed变量	<p>用于指定机器人执行MovJ时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的per分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为V4000</p> <hr/>
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位</li> <li>• RELATIVE: 相对过渡</li> <li>• ABSOLUTE: 绝对过渡</li> <li>• DEFAULT: 缺省类型</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省</p> <hr/>
[c]: 过渡值 (BlendValue)	Zone变量	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为C100</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数仅适用于SCARA机型，可缺省，默认值为使用当前系统中左右手设置</p> <hr/>

参数名称	数据类型	参数含义
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异</li> <li>• CLOSE: 关闭跨奇异</li> <li>• DEFAULT: 缺省设置</li> </ul> <hr/> <p><b>⚠ 注意</b></p> <p>该参数仅适用于通用六关节机器人</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b></p> <p>该参数仅在多机器控制时使用, 可缺省, 默认为当前示教器操作的机器人。</p>

## 示例1:

```
MovJ (P1, v=S.V50, b=" RELATIVE" , c=S.C100)
机器人以V50的速度运行到P1点, 并与下一轨迹存在过渡, 过渡值为C100。
```

## 示例2:

```
MovJ (P2, v=S.V50, b=" RELATIVE" , C0) do S.DO[0]=0 end
机器人以V50的速度运行到P2点, 且无过渡, 运行至P2点时, 设置DO0变量对应IO端口为0。
```

## 示例3:

```
MovJ (P1, v=S.V50, b=" RELATIVE" , c=S.C100)
MovJ (P2, v=S.V50, b=" RELATIVE" , c=S.C100)
机器人以50的速度运动到P1点, P2点坐标系与P1点不一致, 故会运动到P2前会先自动将工具参数切换到P2的工具坐标系, 用户坐标系切换到P2的用户坐标系, 其后在运动到P2, 由于自动切换了坐标系, 此时从P1运动到P2的过程中无过渡效果。
```

## 示例4:

```
MovJ (P1, v=S.V50, b=" RELATIVE" )
MovL (P2, v=S.V50, b=" RELATIVE" )
P1点和P2点的坐标系参数一致, 机器人以50的速度运动到P1点, 其后由于P2点坐标系与P1点一致, 此时无需切换坐标系, 机器人会以过渡的方式运行到P2。
```

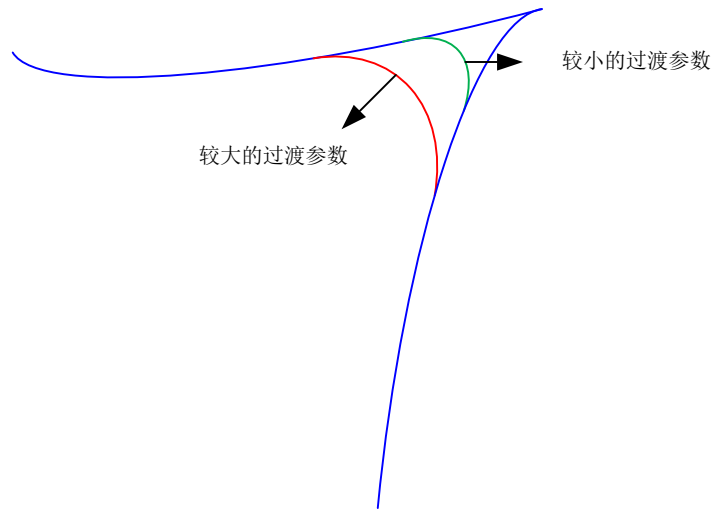


图 3-1 轨迹示意图

### 3.1.2 MovL

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类似。

**MovL** p [v] [b] [v] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 📖 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或Cpos变量	坐标系下的目标点位置
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovL时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。  ⚠️ <b>注意</b> 该参数可缺省，默认值为V4000

参数名称	数据类型	参数含义
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位</li> <li>• RELATIVE: 相对过渡</li> <li>• ABSOLUTE: 绝对过渡</li> <li>• DEFAULT: 缺省类型</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省</p> <hr/>
[c]: 过渡值 (BlendValue)	Zone变量	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为C100</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中设置的负载参数</p> <hr/>
[o]: 速率类型 (Override-Type)	String	<p>设置当前指令轨迹是否受全局速度倍率影响。</p> <ul style="list-style-type: none"> <li>• GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>• GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为GOVRON</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数仅适用于SCARA机型, 可缺省, 默认值为使用当前系统中左右手设置</p> <hr/>

参数名称	数据类型	参数含义
[singular_pass]: 跨奇异 (SingularPass)	String	本运动指令是否开启跨奇异设置 (SingularPass)。 <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <b>⚠ 注意</b> 该参数仅适用于通用六关节机器人
[DO]: 运动完成后执行的IO操作 (Add Do)	String	机器人执行完该指令后, 可触发的IO操作。 <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该参数仅在多机器控制时使用, 可缺省, 默认为当前示教器操作的机器人

### 3.1.3 MovC

圆弧指令指机器人TCP点从起始位置, 经过辅助点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。

**MovC** a p [v] [b] [c] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表:

参数名称	数据类型	参数含义
a: 中间位置 (AuxPos)	Apos或Cpos变量	辅助点位置, 用于确定圆弧大小和运动方向, 该点的姿态不会影响最终轨迹的运行
p: 目标位置 (Target Pos)	Apos或Cpos变量	坐标系下的目标点位置

参数名称	数据类型	参数含义
[v]: 目标速度 (Velo)	Speed变量	<p>用于指定机器人执行MovC时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为V4000</p> <hr/>
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省</p> <hr/>
[c]: 过渡值 (BlendValue)	Zone变量	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为C100</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数</p> <hr/>
[o]: 速率类型 (Override-Type)	String	<p>设置当前指令轨迹是否受全局速度倍率影响。</p> <ul style="list-style-type: none"> <li>• GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>• GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为GOVRON</p> <hr/>

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数仅适用于SCARA机型，可缺省，默认值为使用当前系统中左右手设置</p>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数仅适用于通用六关节机器人</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该参数仅在多机器控制时使用，可缺省，默认为当前示教器操作的机器人。</p>

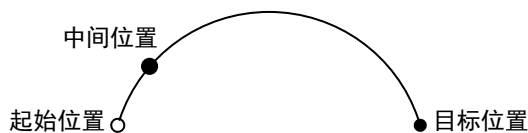


图 3-2 MovC圆弧关键点示意

该指令必须遵循以下规定：

机器人TCP末端做整圆运动，必须执行两个圆弧运动指令。

### 3.1.4 MovCircle

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。

**MovCircle** a p [v] [b] [c] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
a: 中间位置1 (AuxPos1)	Apos或 Cpos变量	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。
p: 中间位置2 (AuxPos2)	Apos或 Cpos变量	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。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovCircle时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效；有外部轴时，Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省，默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[c]: 过渡值 (BlendValue)	Zone变量	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省，默认值为C100。

参数名称	数据类型	参数含义
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数。</p> <hr/>
o: 速率类型 (Override-Type)	String	<p>设置当前指令轨迹是否受全局速度倍率影响。</p> <ul style="list-style-type: none"> <li>GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为GOVRON。</p> <hr/>
[hand]: 左右手 (hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>LEFT: 左手设置。</li> <li>RIGHT: 右手设置。</li> <li>CHANGE: 与点位mode手系相反。</li> <li>DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中左右手设置。</p> <hr/>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置</p> <ul style="list-style-type: none"> <li>OPEN: 开启跨奇异。</li> <li>CLOSE: 关闭跨奇异。</li> <li>DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。该参数可缺省，默认为关闭跨奇异。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作：</p> <ul style="list-style-type: none"> <li>NULL: 无任何操作。</li> <li>IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  ⚠ <b>注意</b> 该选项仅在多机器控制时使用。 该参数可缺省，默认为当前示教器操作的机器人。

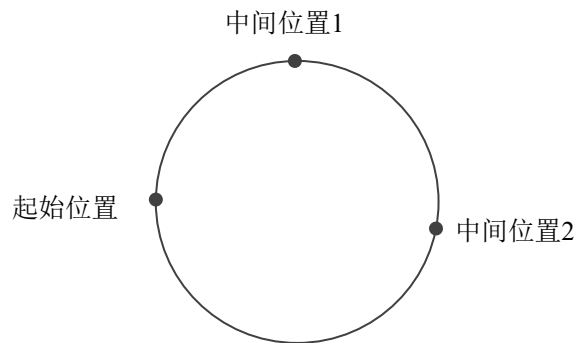


图 3-3 MovCircle整圆关键点示意

该指令必须遵循以下规定：

- 示教整圆轨迹的点位时，“中间位置1”或“中间位置2”与起点距离不能太近；“中间位置1”与“中间位置2”之间的距离不能太近；否则会报警“计算圆弧辅助信息错误”；
- 整圆运动过程中机器人姿态保持不变；
- 整圆运动，支持过渡，支持指令速度不受全局倍率影响；
- 起点位于奇异位置时，需开启跨奇异才能执行整圆运动；
- 外部TCP功能支持整圆运动；
- 变位机、导轨支持整圆运动，需特别注意整圆运动时附加轴位置不变；
- 传送带跟随功能不支持整圆运动。

### 3.1.5 MovJRel

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。

**MovJRel** p [v] [b] [c] [payload] [hand] [singular\_pass] [DO] [r]

 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标相对位置 (RelPos)	DApos变量	执行该指令时机器人要移动的位置增量。
[v]: 目标速度 (Velo)	Speed变量	<p>用于指定机器人执行MovJRel时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的per分量有效；有外部轴时，Speed变量中的exj_l、exj_r分量有效。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为V4000。</p> <hr/>
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE：精确到位。</li> <li>• RELATIVE：相对过渡。</li> <li>• ABSOLUTE：绝对过渡。</li> <li>• DEFAULT：缺省类型。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省。</p> <hr/>
[c]: 过渡值 (BlendValue)	Zone变量	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为C100。</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为当前系统中设置的负载参数。</p> <hr/>

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型。该参数可缺省，默认值为使用当前系统中左右手设置。</p>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认为关闭跨奇异。</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称；</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用。该参数可缺省，默认为当前示教器操作的机器人。</p>

示例1:

```
MovJ(P1, v=S.V50, b=" RELATIVE" , c=S.C100)
MovJRel (DAPOS0, v=S.V50, b=" RELATIVE" , c=S.C100)
机器人以50的速度运动到P1点，其后在P1点的基础上，使用C100中的过渡参数相对运动DAPOS0中的位移，运动到P1`点。
```

示例2:

```
MovJRel (DAPOS0, v=S.V50, b=" RELATIVE" , c=S.C100)
机器人以50的速度，从当前点相对运动DAPOS0中的位移，运动到P1`点。
```

### 3.1.6 MovLRel

该指令为直线相对运动指令。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或者姿态偏移。运动还可设置过渡参数，与MovJRel类似。

**MovLRel** p t [v] [b] [c] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标相对位置 (RelPos)	DCpos变量	执行该指令时机器人要移动的位置增量。
t: 参考坐标系 (RefSys)	enum (COORD/TOOL)	参考坐标系：笛卡尔或工具； <ul style="list-style-type: none"> <li>• COORD: 笛卡尔坐标系。</li> <li>• TOOL: 工具坐标系。</li> </ul>
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovLRel时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效；有外部轴时，Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省，默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[c]: 过渡值 (BlendValue)	Zone变量	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省，默认值为C100。

参数名称	数据类型	参数含义
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为当前系统中设置的负载参数。</p> <hr/>
[o]: 速率类型 (Override-Type)	String	<p>设置当前指令轨迹是否受全局速度倍率影响。</p> <ul style="list-style-type: none"> <li>GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为GOVRON。</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式</p> <ul style="list-style-type: none"> <li>LEFT: 左手设置。</li> <li>RIGHT: 右手设置。</li> <li>CHANGE: 与点位mode手系相反。</li> <li>DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中左右手设置。</p> <hr/>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>OPEN: 开启跨奇异。</li> <li>CLOSE: 关闭跨奇异。</li> <li>DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认为关闭跨奇异。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>NULL: 无任何操作。</li> <li>IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul> <hr/>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称；</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用。该参数可缺省，默认为当前示教器操作的机器人。</p> <hr/>

示例1：

```
MovL(P1, v=S.V50, b=" RELATIVE" , c=S.C100)
MovLRel (DCPOS0, v=S.V50, b=" RELATIVE" , c=S.C100)
机器人以50的速度运动到P1点，其后在P1点的基础上，使用C100中的过渡参数相对运动DCPOS0中的距离，运动到P1`点。
```

示例2：

```
MovLRel (DCPOS0, v=S.V50, b=" RELATIVE" , c=S.C100)
机器人以50的速度，从当前点相对运动DCPOS0中的距离，运动到P1`点。
```

### 3.1.7 MovJSearch

寻位指令，指的是在执行这条MovJ指令时，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。

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。

#### 注意

该指令会打断过渡。

**MovJSearch** p v SearchType SearchIndex SearchValue ResultV StopType [b] [c] [payload] [StopDecScale] [SearchP] [goto] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	APos或 CPos变量	坐标系下的目标点位置。
v: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovJSearch时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的per分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。

参数名称	数据类型	参数含义
SearchType: 检测类型 (SearchType)	String	<p>机器人运行该轨迹时需要监控的操作类型。</p> <ul style="list-style-type: none"> <li>• DITrig: 物理数字输入口触发。</li> <li>• AITrig: 物理模拟量输入口触发。</li> <li>• SIMDITrig: 虚拟数字输入口触发。</li> <li>• SIMAITrig: 虚拟模拟量输入口触发。</li> <li>• TorqTrig: 力矩触发。</li> </ul>
SearchIndex: 触发索引 (SearchIndex)	Int	<p>机器人运行该轨迹时需要检测的触发索引编号。以DITrig触发为例, 该索引编号对应的就是需要监控的输入口号。以TorqTrig触发为例, 该索引编号对应的就是需要监控的轴编号。</p>
SearchValue: 触发值 (SearchValue)	Float	<p>机器人运行该轨迹时需要检测的阈值。以DITrig触发为例, 该值就是某一输入口需要检测的高低电平状态。以TorqTrig触发为例, 该值为某一轴的额定转矩千分比。</p>
ResultV: 返回值 (ResultV)	ReturnValue变量	<p>返回该条指令执行结束的状态。返回0表示未满足条件触发而结束, 返回1表示满足条件触发而结束, 返回2表示在运动起点就满足条件触发而结束。</p>
StopType: 停止类型 (StopType)	String	<p>检测到信号时停止的方式。</p> <ul style="list-style-type: none"> <li>• HARDSTOP: 立即停止。</li> <li>• SOFTSTOP: 减速停止。</li> </ul>
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• DEFAULT: 缺省类型。</li> </ul>
[c]: 过渡值 (BlendValue)	Zone	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为C100。</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中设置的负载参数。</p> <hr/>
[StopDecScale]: 减速系数 (StopDecScale)	Float	<p>当“停止类型”选择“SOFTSTOP”时, 该值有效。用于调节减速加速度的系数, 该值越大减速越快, 反之越慢。</p>
[SearchP]: 寻位成功的位置 (SearchPos)	Apos	<p>返回满足条件发生时刻机器人的坐标位置。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省。</p> <hr/>

参数名称	数据类型	参数含义
[goto]: 寻位失败时跳转的标签名 (LabelName)	label	<p>寻位失败时程序指针将跳转到的label名。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，设为缺省后，当寻位失败时系统将不会跳转到label处执行。</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型； 该参数可缺省，默认值为使用当前系统中左右手设置。</p> <hr/>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置 ( )。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人 该参数可缺省，默认为关闭跨奇异。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。</p> <hr/>

示例1:

```
MovJ(P1, v=S.V50, " RELATIVE" , c=S.C100)
MovJSearch(P2, S.V50, "DITrig", 16, 1.0, ReturnValue0, "SOFTSTOP" , StopDecScale=0.5)
MovJ(P3, v=S.V50, " RELATIVE" , c=S.C100)
机器人以50的速度运动到P1点，其后在运动到P2点的过程中机器人会开始检测16号输入口的状态，
```

当检测到有高电平时，机器人会立即停止往P2点的运动，开始往P3点运动，同时ReturnValue0中的int\_value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ReturnValue0中的int\_value值被设为0。

### 3.1.8 MovLSearch

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。

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。

#### 注意

该指令会打断过渡。

**MovLSearch** p v SearchType SearchIndex SearchValue ResultV StopType [b] [c] [payload] [StopDecScale] [SearchP] [goto] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或Cpos变量	坐标系下的目标点位置。
v: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovLSearch时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效；有外部轴时，Speed变量中的exj_l、exj_r分量有效。
SearchType: 检测类型 (SearchType)	String	机器人运行该轨迹时需要监控的操作类型。 <ul style="list-style-type: none"> <li>• DITrig: 物理数字输入口触发。</li> <li>• AITrig: 物理模拟量输入口触发。</li> <li>• SIMDITrig: 虚拟数字输入口触发。</li> <li>• SIMAITrig: 虚拟模拟量输入口触发。</li> <li>• TorqTrig: 力矩触发。</li> </ul>
SearchIndex: 触发索引 (SearchIndex)	Int	机器人运行该轨迹时需要检测的触发索引编号。以DITrig触发为例，该索引编号对应的就是需要监控的输入口号；以TorqTrig触发为例，该索引编号对应的就是需要监控的轴编号。

参数名称	数据类型	参数含义
SearchValue: 触发值 (SearchValue)	Float	机器人运行该轨迹时需要检测的阈值。以DITrig触发为例, 该值就是某一输入口需要检测的高低电平状态; 以TorqTrig触发为例, 该值为某一轴的额定转矩千分比。
ResultV: 返回值 (ResultV)	ReturnVal-ue变量	返回该条指令执行结束的状态。返回0表示未满足条件触发而结束, 返回1表示满足条件触发而结束, 返回2表示在运动起点就满足条件触发而结束。
StopType: 停止类型 (Stop-Type)	enum (HARDSTOP/SOFTSTOP)	检测到信号时停止的方式。 <ul style="list-style-type: none"> <li>• HARDSTOP: 立即停止。</li> <li>• SOFTSTOP: 减速停止。</li> <li>• DEFAULT: 缺省类型。</li> </ul>
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• DEFAULT: 缺省类型。</li> </ul>
[c]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。 <b>⚠ 注意</b> 该参数可缺省, 默认值为C100。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数; <b>⚠ 注意</b> 该参数可缺省, 默认值为当前系统中设置的负载参数。
[StopDecScale]: 减速系数 (StopDecScale)	Float	当“停止类型”选择“SOFTSTOP”时, 该值有效。用于调节减速加速度的系数, 该值越大减速越快, 反之越慢。
[SearchP]: 寻位成功的位置 (SearchPos)	Cpos	返回满足条件发生时刻机器人的坐标位置; <b>⚠ 注意</b> 该参数可缺省。
[goto]: 寻位失败时跳转的标签名 (LabelName)	label	寻位失败时程序指针将跳转到的label名; <b>⚠ 注意</b> 该参数可缺省, 设为缺省后, 当寻位失败时系统将不会跳转到label处执行。

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中左右手设置。</p>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称；</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用；该参数可缺省，默认为当前示教器操作的机器人。</p>

示例1：

```
MovL(P1, v=S.V50, b=" RELATIVE" , c=S.C100)
MovLSearch(P2, S.V50, "DITrig", 16, 1.0, ReturnValue0, "SOFTSTOP" , StopDecScale=0.5)
MovL(P3, v=S.V50, b=" RELATIVE" , c=S.C100)
```

机器人以50的速度运动到P1点，其后在运动到P2点的过程中机器人会开始检测16号输入口的状态，当检测到有高电平时，机器人会立即停止往P2点的运动，开始往P3点运动，同时ReturnValue0中的int\_value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ReturnValue0中的int\_value值被设为0。

### 3.1.9 MovJOffset



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

**MovJOffset** rp tp t [tool] [coord] [v] [b] [c] [payload] [hand] [singular\_pass] [DO] [r]

#### 注意

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
rp: 参考位置 (RefPos)	Apos或 CPos变量	坐标系下的参考点位置。
tp: 目标相对位置 (RelPos)	DCpos或 DApos变量	相对于参考位置的位置增量。
t: 参考坐标系 (RefSys)	String	参考坐标系：笛卡尔或工具。 <ul style="list-style-type: none"> <li>COORD: 用户坐标系。</li> <li>TOOL: 工具坐标系。</li> </ul>
tool: 工具参数 (Tool)	Tool变量	机器人执行该轨迹时使用的工具参数；只在RP为Apos类型时使用。
coord: 坐标系参数 (Co-ord)	Frame变量	机器人执行该轨迹时使用的用户坐标系；只在RP为Apos类型时使用。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovJOffset时的速度，其中Speed变量中仅per分量有效；  <b>注意</b> 该参数可缺省，默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>FINE: 精确到位。</li> <li>RELATIVE: 相对过渡。</li> <li>ABSOLUTE: 绝对过渡。</li> <li>DEFAULT: 缺省类型。</li> </ul>  <b>注意</b> 该参数可缺省。

参数名称	数据类型	参数含义
[c]: 过渡值 (BlendValue)	Zone	<p>机器人逼近终点时的过渡值;</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为C100。</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数;</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为当前系统中设置的负载参数。</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中左右手设置。</p> <hr/>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称;</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p> <hr/>

示例1:

MovJOffset(P1, DAPOS0, "COORD", S.nullTool, S.World, S.V50, " RELATIVE" , S.C100)  
机器人在笛卡尔坐标系下，工具参数为nullTool，坐标系为World，使用C100中的过渡参数直线运动到相对P1点DAPOS0距离的新点。

### 3.1.10 MovLOffset

直线偏移指令，指的是以指定参考位置为起点位置，相对于坐标系或者工具进行偏移运动。与MovLRel唯一的区别是该指令指定了偏移运动的起点，不一定是当前位置或者上一条指令的终点，其他功能均相同。

**MovLOffset** rp tp t [tool] [coord] [v] [b] [c] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
rp: 参考位置 (RefPos)	Apos或Cpos变量	坐标系下的参考点位置。
tp: 目标相对位置 (RelPos)	DCpos变量	相对于参考位置的位置增量。
t: 参考坐标系 (RefSys)	String	参考坐标系：笛卡尔或工具。 • COORD：笛卡尔坐标系。 • TOOL：工具坐标系。
tool: 工具参数 (Tool)	Tool变量	机器人执行该轨迹时使用的工具参数。
coord: 坐标系参数 (Coord)	Frame变量	机器人执行该轨迹时使用的用户坐标系。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovLOffset时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省，默认值为V4000。

参数名称	数据类型	参数含义
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省。</p> <hr/>
[c]: 过渡值 (BlendValue)	Zone	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为C100。</p> <hr/>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数。</p> <hr/>
[o]: 速率类型 (Override-Type)	String	<p>设置当前指令轨迹是否受全局速度倍率影响。</p> <ul style="list-style-type: none"> <li>• GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>• GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为GOVRON。</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型； 该参数可缺省，默认值为使用当前系统中左右手设置。</p> <hr/>

参数名称	数据类型	参数含义
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p>

示例1:

```
MovLOffset(P1, DCPOS0, "COORD", S.nullTool, S.World, S.V50, " RELATIVE" , S.C100)
```

机器人在笛卡尔坐标系下, 工具参数为nullTool, 坐标系为World, 使用C100中的过渡参数直线运动到相对P1点DCPOS0距离的新点。

### 3.1.11 MovJOffsetC

关节空间偏移运动指令, 与MovJOffset指令功能一致, 区别在于目标相对位置由DCpos类型的变量更改为X增量、Y增量、Z增量、A增量四个Float型常量。

**MovJOffsetC** rp tpx tpy tpz tpa t [tool] [coord] [v] [b] [c] [payload] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表:

参数名称	数据类型	参数含义
rp: 参考位置 (RefPos)	Apos或 Cpos变量	坐标系下的参考点位置

参数名称	数据类型	参数含义
tpx: X增量 (OffsetX)	Float常量	执行该指令时机器人在X方向上要移动的位置增量
tpy: Y增量 (OffsetY)	Float常量	执行该指令时机器人在Y方向上要移动的位置增量
tpz: Z增量 (OffsetZ)	Float常量	执行该指令时机器人在Z方向上要移动的位置增量
tpa: A增量 (OffsetA)	Float常量	执行该指令时机器人在A方向上要移动的位置增量
t: 参考坐标系 (RefSys)	String	参考坐标系: 笛卡尔或工具 <ul style="list-style-type: none"> <li>COORD: 用户坐标系。</li> <li>TOOL: 工具坐标系。</li> </ul>
tool: 工具参数 (Tool)	Tool变量	机器人执行该轨迹时使用的工具参数
coord: 坐标系参数 (Co-ord)	Frame变量	机器人执行该轨迹时使用的用户坐标系
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovJOffset时的速度, 其中Speed变量中仅per分量有效 <b>⚠ 注意</b> 该参数可缺省, 默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式 <ul style="list-style-type: none"> <li>FINE: 精确到位。</li> <li>RELATIVE: 相对过渡。</li> <li>ABSOLUTE: 绝对过渡。</li> <li>DEFAULT: 缺省类型</li> </ul>
[c]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值 <b>⚠ 注意</b> 该参数可缺省, 默认值为C100。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数 <b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中设置的负载参数。

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	终点位置的左右手类型方式 <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <b>⚠ 注意</b> 该选项仅适用于SCARA机型; 该参数可缺省, 默认值为使用当前系统中左右手设置。
[singular_pass]: 跨奇异 (SingularPass)	String	本运动指令是否开启跨奇异设置 <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <b>⚠ 注意</b> 该选项仅适用于通用六关节机器人
[DO]: 运动完成后执行的IO操作 (Add Do)	String	机器人执行完该指令后, 可触发的IO操作 <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称 (仅在多机器控制时使用) <hr/> <b>⚠ 注意</b> 该参数可缺省, 默认为当前示教器操作的机器人。

### 3.1.12 MovLW

摆动指令, 指的是根据设定的摆频、摆幅, 在直线前进方向上做摆弧运动。

**MovLW** p weave [v] [b] [c] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表:

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置。
weave: 摆弧变量 (Weave Value)	Weave变量	机器人摆动时的参数摆动参数。其中存储着摆频、摆幅、停止时间、旋转等参数。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovLW时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省，默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[c]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省，默认值为C100。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数。  <b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数。
[o]: 速率类型 (Override-Type)	String	设置当前指令轨迹是否受全局速度倍率影响。 <ul style="list-style-type: none"> <li>• GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>• GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul>

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式 ( )。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型; 该参数可缺省, 默认值为使用当前系统中左右手设置。</p>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p>

### 3.1.13 MovCW

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。

**MovCW** a p weave [v] [b] [c] [payload] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表:

参数名称	数据类型	参数含义
a: 中间位置 (AuxPos)	Apos或 Cpos变量	辅助点位置, 用于确定圆弧大小和运动方向, 该点的姿态不会影响最终轨迹的运行。
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置。
weave: 摆弧变量 (Weave Value)	Weave变量	机器人摆动时的参数摆动参数。其中存储着摆频、摆幅、停止时间、旋转等参数。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovCW时的速度, 包括机器人末端的平移速度, 旋转速度以及外部轴的移动速度等。此时, Speed变量中的tcp、ori分量有效。有外部轴时, Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省, 默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[c]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省, 默认值为C100。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数。  <b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
[o]: 速率类型 (Override-Type)	String	设置当前指令轨迹是否受全局速度倍率影响。 <ul style="list-style-type: none"> <li>• GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>• GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <b>⚠ 注意</b> 该参数可缺省, 默认值为GOVRON。

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式（该选项仅适用于SCARA机型）。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型； 该参数可缺省，默认值为使用当前系统中左右手设置。</p>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。</p>

### 3.1.14 MovCircleW

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。

**MovCircleW** a p weave [v] [b] [c] [o] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
a: 中间位置1 (AuxPos1)	Apos或 Cpos变量	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。
p: 中间位置2 (AuxPos2)	Apos或 Cpos变量	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。
weave: 摆弧变量 (Weave Value)	Weave变量	机器人摆动时的参数摆动参数。其中存储着摆频、摆幅、停止时间、旋转等参数。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovCircleW时的速度, 包括机器人末端的平移速度, 旋转速度以及外部轴的移动速度等。此时, Speed变量中的tcp、ori分量有效。有外部轴时, Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省, 默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[c]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省, 默认值为C100。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数。  <b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
[o]: 速率类型 (Override-Type)	String	设置当前指令轨迹是否受全局速度倍率影响。 <ul style="list-style-type: none"> <li>• GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>• GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <b>⚠ 注意</b> 该参数可缺省, 默认值为GOVRON。

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型; 该参数可缺省, 默认值为使用当前系统中左右手设置。</p> <hr/>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称 (。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p> <hr/>

### 3.1.15 OnDistance

距离触发指令, 在不打断过渡的前提下, 控制器可以从直线或圆弧轨迹的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。

**OnDistance** type dist delay\_time DO [r]

指令参数说明表:

参数名称	数据类型	参数含义
type: 触发类型 (Type)	String	距离触发检测的方式, 即从起点处计算距离还是从终点处计算距离。 <ul style="list-style-type: none"> <li>• FromBegin: 从起点处触发。</li> <li>• FromEnd: 从终点处触发。</li> </ul>
dist: 触发距离 (TrigDis)	Float	相对于起点或者终点的触发距离。
delay_time: 触发时间 (TrigTime)	Int	满足触发条件后延时触发的时间。该值可为负数, 负数表示向前延时触发。
DO: 运动完成后执行的IO操作 (Do)	String	机器人满足触发条件后, 可触发的IO指令。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

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, 延时触发时间后, 开始执行相关的触发操作。

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

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

```
MovL (P1, S.V50, " RELATIVE" , S.CO)
OnDistance( "FromBegin" ,200.0,100) do S.DO[1]=1 end
MovL (P2, S.V50, " RELATIVE" , S.CO)
```

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

```
MovL (P1, S.V50, " RELATIVE" , S.CO)
OnDistance( "FromEnd" ,200.0,100) do S.DO[0]=1 end
MovL (P2, S.V50, " RELATIVE" , S.CO)
```

**⚠ 注意**

- 当距离触发满足且时间触发还在延时时，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。触发类型为终点触发时，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可插入除等待以外指令，否则执行报错。等待指令如wait, waitcondition, waitfinish等。

### 3.1.16 OnParameter

距离百分比触发指令，用于在不打断过渡的前提下，在所有运动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。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，延时触发时间后，开始执行相关的触发操作。

**OnParameter** percence delay\_time DO [r]

指令参数说明表：

参数名称	数据类型	参数含义
percence: 触发类型 (Type)	Int	需要触发的路径所在占比。
delay_time: 触发时间 (TrigTime)	Int	满足触发条件后触发延时触发的时间。该值可为负数，负数表示向前延时触发。
DO: 运动完成后执行的IO操作 (Do)	String	机器人满足触发条件后，可触发IO指令。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例：

```
MovJ (P1, S.V50, " RELATIVE" , S.C0)
OnParameter(60,500) do S.DO[0]=1 end
```

MovL (P2, S.V50, " RELATIVE" , S.C0)  
在P1到P2轨迹运行到60%处后，延时500ms执行Do操作。

### 3.1.17 MovJArch

SCARA机型的关节拱形运动指令，即以MovJ的形式走完轨迹。整个运动过程为：首先机器人在当前点抬升h1高度，其后再运动到距离目标点高度为h2处，最后再下降h2高度到达目标点，完成一次拱形运动。与正常的运动指令相同，在执行完该指令时可以附加进行IO操作，该指令的目标速度为百分比。

#### 注意


该指令目前只用于scara机型，不支持六轴机器人。

**MovJArch** p height\_type z [v] [b] [c] [payload] h1 h2 abs\_height [hand] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置。
height_type: 高度类型 (HeightType)	String	机器人执行该轨迹时使用的高度类型。 <ul style="list-style-type: none"> <li>RELATIVE: 相对高度类型。</li> <li>ABSOLUTE: 绝对高度类型。</li> </ul>
z: 转角值 (CornerValue)	Zone	机器人走拱形运动时的转角值。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovL时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的per分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。 <p> <b>注意</b> 该参数可缺省，默认值为V4000。</p>

参数名称	数据类型	参数含义
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省。</p>
[c]: 过渡值 (BlendValue)	Zone	<p>机器人逼近终点时的过渡值 (Zone类型变量), 其中, per参数对应相对过渡类型有效, dis参数对应绝对过渡类型有效。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为C100。</p>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中设置的负载参数。</p>
h1: 抬升高度 (Lift High)	Float	<p>使用相对高度类型时, 机器人执行关节拱形运动时需要抬高的高度。</p>
h2: 下降高度 (Drop High)	Float	<p>使用相对高度类型时, 机器人执行关节拱形运动时需要下降的高度。</p>
abs_height: 绝对高度 (AbsHeight)	Float	<p>使用绝对高度类型时, 机器人执行关节拱形运动时抬升的绝对高度。</p>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为使用当前系统中左右手设置。</p>

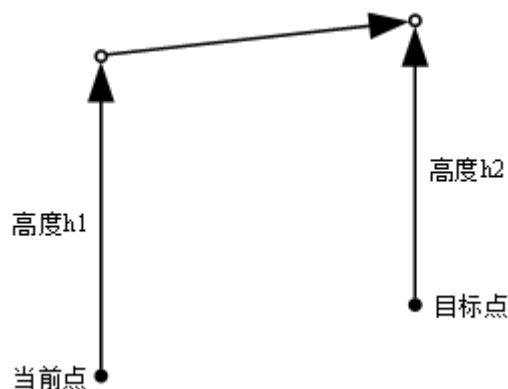
参数名称	数据类型	参数含义
[DO]: 运动完成后执行的IO操作 (Add Do)	String	机器人执行完该指令后, 可触发的IO操作。 <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

示例:

```
MovL (P1, S.V50, " RELATIVE" , S.C50)
MovJArch (P2, " RELATIVE" , S.C100, v=S.V100, h1=10, h2=10,)
机器人会先用V50的速度运动到P1点, 其后执行关节拱形运动指令, 机器人以V100的速度先运动到
P1.z+10mm的
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。
```

### 3.1.18 MovLArch

SCARA机型的直线拱形运动指令, 即以MovL的形式走完轨迹。整个运动过程为: 首先机器人在当前点抬升 $h_1$ 高度, 其后再以MovL的方式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。与正常的运动指令相同, 在执行完该指令时可以附加进行IO操作, 该指令的目标速度为百分比。



#### ⚠ 注意

该指令目前只用于scara机型, 不支持六轴机器人。

**MovLArch** p height\_type z [v] [b] [c] [payload] h1 h2 abs\_height [o] [hand] [DO] [r]

 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置。
height_type: 高度类型 (HeightType)	String	机器人执行该轨迹时使用的高度类型。 <ul style="list-style-type: none"> <li>• RELATIVE: 相对高度类型。</li> <li>• ABSOLUTE: 绝对高度类型。</li> </ul>
z: 转角值 (CornerValue)	Zone	机器人走拱形运动时的转角值。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovL时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的per分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。  <b>⚠ 注意</b> 该参数可缺省，默认值为V4000。
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[c]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值 (Zone类型变量)，其中，per参数对应相对过渡类型有效，dis参数对应绝对过渡类型有效。  <b>⚠ 注意</b> 该参数可缺省，默认值为C100。

参数名称	数据类型	参数含义
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数。</p> <hr/>
h1: 抬升高度 (Lift High)	Float	使用相对高度类型时，机器人执行直线拱形运动时需要抬高的高度。
h2: 下降高度 (Drop High)	Float	使用相对高度类型时，机器人执行直线拱形运动时需要下降的高度。
abs_height: 绝对高度 (AbsHeight)	Float	使用绝对高度类型时，机器人执行直线拱形运动时抬升的绝对高度。
[o]: 速率类型 (Override-Type)	String	<p>设置当前指令轨迹是否受全局速度倍率影响。</p> <ul style="list-style-type: none"> <li>GOVRON: 本段轨迹受全局速度倍率影响。</li> <li>GOVROFF: 本段轨迹不受全局速度倍率影响。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为GOVRON。</p> <hr/>
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>LEFT: 左手设置。</li> <li>RIGHT: 右手设置。</li> <li>CHANGE: 与点位mode手系相反。</li> <li>DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为使用当前系统中左右手设置。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>NULL: 无任何操作。</li> <li>IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。</p> <hr/>

示例：

```
MovL (P1, S.V50, " RELATIVE" , S.C50)
MovLArch (P2, " RELATIVE" , S.C100, v=S.V100, h1=10, h2=20)

```

机器人会先用V50的速度运动到P1点，其后执行直线拱形指令，机器人以V100的速度先运动到P1.z+10mm的位置，之后再运动到P2.Z+20mm的位置，最后运动到P2的位置，整个运动间均以C100过渡。

### 3.1.19 MovH




该指令为螺旋线运动指令，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。

**MovH** a p r s [v] [b] [c] [payload] [hand] [singular\_pass] [DO] [r]

#### 说明

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
a: 中间位置 (AuxPos)	Apos或 Cpos变量	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。
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置。
r: 半径 (Radius)	Float	螺旋线运动时螺旋线的半径，单位mm。
s: 步进距离 (Stepping)	Float	螺旋旋转一周后沿起终点方向移动的距离，单位mm。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovH时的速度，螺旋线最大轨迹线速度为系统最大速度的0.25倍。
[b]: 过渡类型 (BlendType)	/	 <b>注意</b> MovH不支持过渡
[c]: 过渡值 (BlendValue)	/	 <b>注意</b> 该参数不起作用。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数。  <b>注意</b> 该参数可缺省，默认值为使用当前系统中设置的负载参数。

参数名称	数据类型	参数含义
[hand]: 左右手 (Hand)	String	<p>终点位置的左右手类型方式。</p> <ul style="list-style-type: none"> <li>• LEFT: 左手设置。</li> <li>• RIGHT: 右手设置。</li> <li>• CHANGE: 与点位mode手系相反。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于SCARA机型; 该参数可缺省, 默认值为使用当前系统中左右手设置。</p> <hr/>
[singular_pass]: 跨奇异 (SingularPass)	String	<p>本运动指令是否开启跨奇异设置。</p> <ul style="list-style-type: none"> <li>• OPEN: 开启跨奇异。</li> <li>• CLOSE: 关闭跨奇异。</li> <li>• DEFAULT: 缺省设置。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该选项仅适用于通用六关节机器人。</p> <hr/>
[DO]: 运动完成后执行的IO操作 (Add Do)	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p> <hr/>

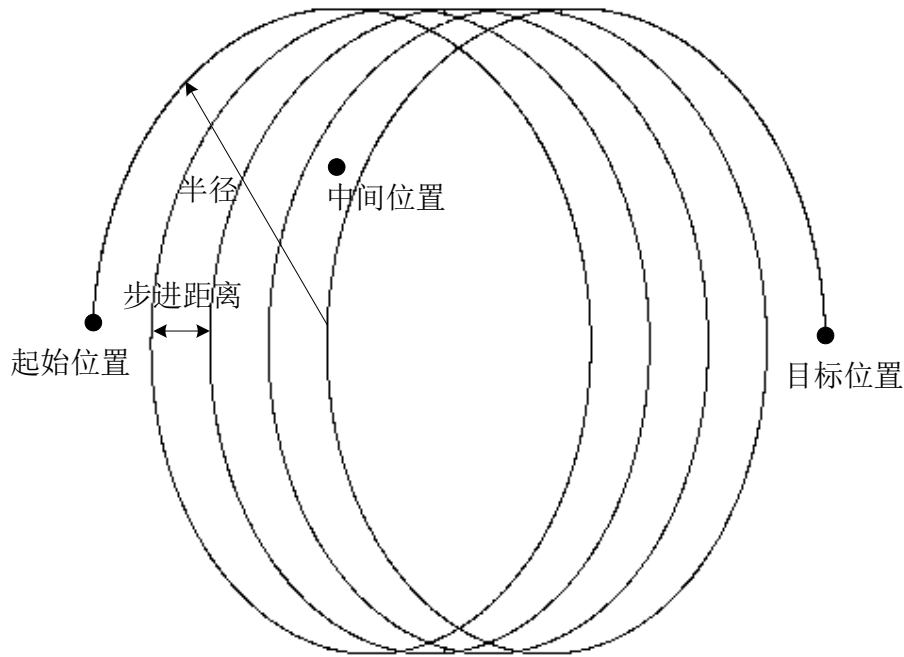


图 3-4 MovH螺旋运动关键点示意

**⚠ 注意**

螺旋运动不支持过渡、不支持WaitFinish

示例：

```
MovL (P1, S.V50, " RELATIVE" , S.C50)
```

```
MovH (P2, P3, 10, 5, v=S.V100, payload=S.Payload[0])
```

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。

## 3.2 控制指令

控制指令列表：

控制指令	LABEL
	GOTO
	if
	elseif
	else

	while
	for
	switch
	case
	break
	default
	continue
	try
	catch
	finally
	return
	...=...
	#=...=#

### 3.2.1 LABEL

LABEL指令用于定义GOTO跳转目标。

示例：

```
label1:
Label指令设置标签行label1
```

### 3.2.2 GOTO

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while循环程序块或者if程序块内部。

示例：

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

### 3.2.3 if

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。每一个if指令必须以关键字end做为条件判断程序块的结束。

示例：

```
if x > 1 && y == 1 || z < 1
y = 10
end
满足if条件表达式，触发执行“y = 10”，否则end退出
```

### 3.2.4 elseif

elseif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elseif逻辑判断。设置elseif表达式的方法与设置if表达式的方法相同。

示例：

```
if x > 1 && y == 1 || z < 1
y = 10
elseif x < 1
y = 20
end
满足if条件表达式，触发执行“y = 10”，否则继续检查，满足elseif条件表达式，触发执行“y = 20”，end退出
```

### 3.2.5 else

else指令需与if指令联合使用，表示if条件判断的不成立时将执行的语句。

示例：

```
if x > 1 && y == 1 || z < 1
y = 10
else
y = 20
end
满足if条件表达式，触发执行“y = 10”，否则执行“y = 20”，end退出
```

### 3.2.6 while

while指令在满足条件的时候循环执行子语句。该指令必须以关键字end作为循环控制结束。

示例：

```
while X < Y
MovJ(P1)
MovJ(P2)
X=X+1
end
满足while条件表达式，执行内部指令行，直到end指令行，此指令行触发跳回到while指令行，再次判断while条件表达式，如此循环，直到while条件表达式不满足，跳转到end指令下一行执行
```

### 3.2.7 for

for指令在满足条件的时候循环执行子语句，该指令必须以关键字end作为循环控制结束。for指令用于迭代范围中的元素，并对每个元素执行相同操作，结构如下：

```
for <variable> in <range>
...
end
```

其中<variable>是循环变量，用于在每次迭代中代表循环中的一个元素。<range>是要迭代的范围，若满足范围条件，则执行for中间内容，不满足，则退出for循环。

示例：

```
for i in 1:5
Movj(P1)
end
Movj(P2)
```

执行for指令时，当范围条件满足时，执行Movj (P1) 语句，然后再次进行范围条件的判断，重复上述的操作。待不满足范围条件时，结束循环，执行Movj(P2)语句。

### 3.2.8 switch

switch语句必须以end为结尾，之间与case、break、default等语句配合使用，switch后面必须为变量，该变量与case值进行比较，执行与之匹配case的语句块，如果没有匹配的case，将会执行default语句块。

示例：

```
switch Var
case 1
Movj{P1}
break
case 2
Movj{P2}
break
case 3
Movj{P3}
break
default
Movj{P4}
break
end
```

switch执行匹配的case语句遇到break结束执行，没有匹配的case，将会执行default语句

### 3.2.9 case

case语句必须放在switch里面使用，且case后面必须是常量，可以为int、string类型的常量，且case后面常量类型必须与switch后面的变量类型一致。

示例：

```
switch INTVar
case 1
MovJ{P1}
break
case 2
MovJ{P2}
break
case 3
MovJ{P3}
break
default
MovJ{P4}
break
end
```

如果满足条件的case后面没有break，将会执行后续的case语句且不会判断case的值，直到遇到break或者end结束运行，如下示例：

```
switch INTVar
case 1
MovJ{P1}
case 2
MovJ{P2}
break
default
break
end
```

如果满足case 1， case 2的语句块也将被执行，不管case 2值为何值

### 3.2.10 break

break语句必须和case或者default成对使用，用于跳出switch语句。

示例：

```
switch Var
case 1
MovJ{P1}
break
case 2
MovJ{P2}
break
case 3
MovJ{P3}
break
default
MovJ{P4}
break
end
```

当执行到break语句时，会跳出switch语句的执行，会执行到end下一行指令。

### 3.2.11 default

default指令，用于在switch里面，必须在case语句下面即处于整个switch语句块的最下面，其作用的是如果所用的case语句都不匹配，将会执行default语句块。

示例：

```
switch StrVar
case "helloone"
  MovJ{P1}
  break
case "hellotwo"
  MovJ{P2}
  break
case "hellothree"
  MovJ{P3}
  break
default
  MovJ{P4}
  break
end
```

如果switch后面的变量和所用case后面常量不匹配，会执行default语句块。

### 3.2.12 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。

示例：

```
for i in 1:5
  if i == 3
    continue
  end
  MovJ{P0}
end
```

当i=3时，执行continue语句，结束本次循环，不执行MovJ语句，直接进入下一次循环中。

### 3.2.13 try

try用于包裹可能会抛出异常的代码块，catch用于捕获异常并处理异常，finally用于指定无论是否发生异常都会执行的代码块。

示例：

```
try
  a = 10/0
finally
  MovJ(P0)
end
```

try包裹可能会抛出异常的代码块a = 10/0。

### 3.2.14 catch

catch用于捕获异常并处理异常，catch必须与try搭配使用。

示例：

```
try
  a = 10/0
catch
  SetRtInfo(" 出现异常" ,70001)
finally
  Movj(P0)
end
```

try包裹可能会抛出异常的代码块，若有异常，则catch捕获异常并处理。

### 3.2.15 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。

示例：

```
try
  a = 10/0
finally
  Movj(P0)
end
```

无论是否抛出异常，都会执行Movj语句。

### 3.2.16 return

返回指令。执行该指令后，程序会跳转至程序的末尾，达到结束当前程序的作用。

示例：

```
if x>1
  return 1
end
return 0
```

当满足if条件，执行到return 1指令，跳转至程序的末尾，达到结束当前程序的作用

### 3.2.17 ...=...

建立一个表达式，用来给某个变量赋值。

示例：

```
X = Y + Z + 1
```

此指令可以执行变量、常量、函数等混合运算赋值

### 3.2.18 #=...=#

注释指令，为程序添加释义或备注。

该指令不会影响程序的运行，仅仅是便于用户阅读和理解程序。

示例：

```
#=给变量赋值=#  
X = Y + Z + 1
```

### 3.3 等待指令

等待指令列表：

等待指令	Wait
	WaitFinish
	WaitCondition

#### 3.3.1 Wait

延时等待，参数可设为Int变量类型，时间单位：ms。

示例：

```
Wait(1000)  
等待1秒
```

#### 3.3.2 WaitFinish

运动等待，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。

示例：

```
MovL(P1, 1000, " RELATIVE" , S.C100)  
WaitFinish (20)  
S.DO[1] = 1  
当MovL指令执行到20%时，执行设置DO指令，指令参数为百分比，范围0~100
```

#### 3.3.3 WaitCondition

条件等待，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。

**WaitCondition** c timeout [goto]

**WaitCondition**指令参数说明表

参数名称	数据类型	参数含义
c: 判别条件 (Condition)	/	条件表达式, 为真时, 才执行下一条指令, 否则程序将继续等待直到表达式为真。
timeout: 时长 (Time)	Int变量类型, 其值为正整数与0	执行等待时所需的时间, 单位为ms。如果该参数的值为0, 将强制等待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。
[goto]: 超时跳转的标签名 (Label Name)	label	<p>超时后跳转的label名, 该label为在当前加载的程序中示教过的label指令名, 如Start。当等待超时后, 程序指针将跳转到该标签名对应的指令行。</p> <hr/> <p><b>⚠ 注意</b></p> <p>该参数可缺省, 默认值为不跳转。</p> <hr/>

示例:

```
WaitCondition(1000, lbl=:label1) do X > 0 end
等待时间1秒内, 若X > 0满足, 执行下一条指令; 否则跳转到lable1进行执行
```

## 3.4 IO指令

IO指令列表:

IO指令	PulseOut
	WaitIO
	WaitIO8421

### 3.4.1 PulseOut

该指令用于在某一数字量输出端口输出指定时长与状态的脉冲。

**PulseOut** io\_type port value keep\_time interrupt [r]

指令参数说明表:

参数名称	数据类型	参数含义
io_type: DO/SimDO	String	用于指定输出脉冲的数字量输出端口类型。
port: 数字量输出变量 (Target DO Port)	Int	用于指定输出脉冲的数字量输出端口号。
value: 目标值 (value)	Int	在指定数字量输出端口输出脉冲的状态。
keep_time: 时长 (Keep Time)	Int	脉冲信号持续时间。

参数名称	数据类型	参数含义
interrupt: 中断使能 (Interrupt Enable)	Int	<p>该参数用于设定在脉冲输出过程中若暂停程序，端口是否继续输出脉冲信号。</p> <ul style="list-style-type: none"> <li>0: 遇到程序暂停时，脉冲会持续输出到设定时间后停止输出，即不受程序暂停而停止其功能。</li> <li>1: 遇到程序暂停时，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。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b></p> <p>该选项仅在多机器控制时使用。</p> <p>该参数可缺省，默认为当前示教器操作的机器人。</p> <hr/>

示例1:

```
PulseOut( "DO", 1, 1, 500, 1)
```

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

### 3.4.2 WaitIO

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。

**WaitIO** io\_type port value timeout ret [Goto]

指令参数说明表:

参数名称	数据类型	参数含义
io_type: DI, SimDI, AI或SimAI	String	表示需要等待的IO类型。
port: 端口号 (Port)	Int变量或常量	需要等待的IO端口号。
value: 目标值 (Value)	Float变量或常量	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。
timeout: 时长 (Timeout)	Int变量或常量	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。

参数名称	数据类型	参数含义
ret: 操作返回值 (ReturnValue)	ReturnValue	返回该指令执行结束后的状态, 其中的int_value参数表示对应执行结果。 <ul style="list-style-type: none"> <li>• 0: 成功检测到信号</li> <li>• 1: 未检测到信号, 指令超时返回。</li> </ul>
[Goto]: 超时跳转的标签名 (Label name)	label	超时后跳转的label名, 该label为当前函数中示教过的label指令名, 如Start。当等待超时后, 程序指针将跳转到该标签名对应的指令行。 <hr/> <b>⚠ 注意</b> 该参数可缺省, 默认值为不跳转。 <hr/>

示例1:

```
WaitIO("DI",15,1,0,1000,ReturnValue1)
等待数字量输入端口15的值为"1"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。
```

### 3.4.3 WaitIO8421

该指令用于在指定时长内等待一组连续DI/SimDI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。

**WaitIO8421** io\_type begin\_port end\_port value timeout ret [Goto]

指令参数说明表:

参数名称	数据类型	参数含义
io_type: IO类型, 可以选择DI或SimDI	String	表示等待的IO类型。
begin_port: 起始端口 (Begin Port)	Int变量或常量	该段连续DI/SimDI端口的起始端口号。
end_port: 结束端口 (End Port)	Int变量或常量	该段连续DI/SimDI端口的结束端口号。
value: 目标值 (Value)	Int变量或常量	将该段连续DI或SimDI端口值转换成十进制数, 若与value值相等则视为满足条件。
timeout: 时长 (Timeout)	Int变量或常量	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。
ret: 操作返回值 (ReturnValue)	ReturnValue	返回该指令执行结束后的状态, 其中的int_value参数表示对应执行结果。 <ul style="list-style-type: none"> <li>• 0: 成功检测到信号</li> <li>• 1: 未检测到信号, 指令超时返回。</li> </ul>

参数名称	数据类型	参数含义
[Goto]: 超时跳转的标签名 (Label name)	label	<p>超时后跳转的label名, 该label为当前函数中示教过的label指令名, 如Start。当等待超时后, 程序指针将跳转到该标签名对应的指令行。</p> <hr/> <p><b>⚠ 注意</b></p> <p>该参数可缺省, 默认值为不跳转。</p> <hr/>

示例1:

```
WaitIO8421("DI",11,13,7,1000,ReturnValue0)
在1000ms时长内等待11到13号DI口状态为111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。
```

### 3.5 设置指令

设置指令列表:

设置指令	SetJointDyn
	SetCartDyn
	SetBlendParam
	SetPayload
	SetOverRide
	SetColliEnable
	SetAxisColliParam
	GetCurOverRide
	CalcTool
	CalcFrame
	SetRtToErr
	SetRtInfo
	SetRtWarning
	Stop
	SetMotionMode
	AutoGainEnable
	AutoGainDisable
	SetRestorePC
	SetOriMode
	SetCircMode
GetRobotIsMoving	
SetColliAutoTune	

	SetColliParam
	SetAxisVibraBLevel
	RecordUserInfo
	SetWristMoveEnable
	SetWristMoveDisable

### 3.5.1 SetJointDyn

该指令用于设置关节动态参数。运行该指令后，后续运动指令均以该动态参数运行。

**SetJointDyn** acc dec jerk torq [r]

指令参数说明表：

参数名称	数据类型	参数含义
acc: 加速度(%) (Acc(%))	Int或Int变量	关节最大加速度的百分比，取值范围：1-100。
dec: 减速度(%) (Dec(%))	Int或Int变量	关节最大减速度的百分比，取值范围：1-100。
jerk: 加加速度 (Jerk(%))	Int或Int变量	关节最大加加速度的百分比，取值范围：1-100。
torq: 力矩 (Torq(%))	Int或Int变量	关节最大力矩的百分比，取值范围：1-100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称（该选项仅在多机器控制时使用）。  <b>⚠ 注意</b> 该参数可缺省，默认为当前示教器操作的机器人。

示例：

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

### 3.5.2 SetCartDyn

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。

**SetCartDyn** acc dec jerk oriAcc oriDec oriJerk torq [r]

指令参数说明表：

参数名称	数据类型	参数含义
acc: 加速度(%) (Acc(%))	Int或Int变量	直线及圆弧运动最大加速度的百分比，取值范围：1-100。

参数名称	数据类型	参数含义
dec: 减速度(%) (Dec(%))	Int或Int变量	直线及圆弧运动最大减速度的百分比, 取值范围: 1-100。
jerk: 加加速度 (Jerk(%))	Int或Int变量	直线及圆弧运动最大加加速度的百分比, 取值范围: 1-100。
oriAcc: 姿态加速度(%) (OriAcc(%))	Int或Int变量	姿态最大旋转加速度的百分比, 取值范围: 1-100。
oriDec: 姿态减速度(%) (OriDec(%))	Int或Int变量	姿态最大旋转减速度的百分比, 取值范围: 1-100。
orijerk: 姿态加加速度 (OriJerk(%))	Int或Int变量	姿态最大旋转加加速度的百分比, 取值范围: 1-100。
torq: 力矩 (Torq(%))	Int或Int变量	关节最大力矩的百分比, 取值范围: 1-100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称 (该选项仅在多机器控制时使用)。  <b>⚠ 注意</b> 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

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

### 3.5.3 SetBlendParam

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。

**SetBlendParam** b c [r]

指令参数说明表:

参数名称	数据类型	参数含义
b: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> </ul>
c: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省, 默认值为C100。

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
SetBlendParam("RELATIVE")
运行该指令后, 后续运动指令均以该过渡参数运行。
```

### 3.5.4 SetPayload

该指令用于设置工件负载质量。

**SetPayload** payload [r]

指令参数说明表:

参数名称	数据类型	参数含义
payload: 负载质量 (Payload)	Payload变量	工件负载参数, 其中存储负载指令等一系列相关参数。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
SetPayload (Payload0)
运行该指令后, 系统会将变量Payload0中参数作为当前工件负载参数, 后续运动指令均以该质量运行。
```

### 3.5.5 SetOverride

该指令用于设置全局速度。运行该指令后, 后续运动指令均以该速度运行。

**SetOverride** ovri [r]

指令参数说明表:

参数名称	数据类型	参数含义
ovri: 全局速度(%) (velocity factor(%))	Int或者Int变量类型	全局速度的百分比, 取值范围为1-100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

示例:

```
SetOverride(80)
运行该指令后, 后续运动指令均以该速度运行。
```

### 3.5.6 SetColliEnable

设置碰撞检测使能指令。

**SetColliEnable** axisid valid [r]

指令参数说明表:

参数名称	数据类型	参数含义
axisid: 轴号 (AxisID)	String	将要设置碰撞使能的轴号。 <ul style="list-style-type: none"> <li>• GROUP: 全部轴。</li> <li>• A1: 1轴。</li> <li>• A2: 2轴。</li> <li>• A3: 3轴。</li> <li>• A4: 4轴。</li> <li>• A5: 5轴。</li> <li>• A6: 6轴。</li> </ul>
valid: 有效性 (Valid)	String	是否有效。 <ul style="list-style-type: none"> <li>• ENABLE: 有效。</li> <li>• DISABLE: 无效。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

示例：

```
SetColliEnable("GROUP","ENABLE")
所有轴碰撞使能有效。
```

### 3.5.7 SetAxisColliParam

设置碰撞检测灵敏度值指令。

**SetAxisColliParam** axisid value

指令参数说明表：

参数名称	数据类型	参数含义
axisid: 轴号 (AxisID)	String	将要设置碰撞参数的轴号。 <ul style="list-style-type: none"> <li>• A1: 1轴。</li> <li>• A2: 2轴。</li> <li>• A3: 3轴。</li> <li>• A4: 4轴。</li> <li>• A5: 5轴。</li> <li>• A6: 6轴。</li> </ul>
value: 灵敏度值 (Value)	常量	碰撞灵敏度。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

示例：

```
SetAxisColliParam("A1",100)
设置1轴碰撞灵敏度值为100。
```

### 3.5.8 GetCurOverRide

该指令用于获取当前的全局速度。运行该指令后，获取的全局速度值将存储在传入的变量中。

v = **GetCurOverRide** [r]

指令参数说明表：

参数名称	数据类型	参数含义
v: 全局速度(%) (velocity factor(%))	Int变量	存储从系统中读取的全局速度百分比, 以便其他使用。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
Int0 = GetCurOverRide()
运行该指令后, 当前的全局速度值将会被读取到Int0中, 在变量列表可查看到该值。
```

### 3.5.9 CalcTool

该指令根据提供的四个点计算标定工具参数。

**CalcTool** p1 p2 p3 p4 t [r]

指令参数说明表:

参数名称	数据类型	参数含义
p1: 标定点1 (the first position)	Cpos	第一个标定点。
p2: 标定点2 (the second position)	Cpos	第二个标定点。
p3: 标定点3 (the third position)	Cpos	第三个标定点。
p4: 标定点4 (the forth position)	Cpos	第四个标定点。
t: 工具参数 (Tool Param)	Tool	计算标定出的工具参数。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:


```
CalcTool (p1, p2, p3, p4, S.TOOL[1])
运行该指令后, 可根据提供的四个点信息计算标定出工具参数, 并保存于S.TOOL[1]中。
```

### 3.5.10 CalcFrame

该指令根据提供的三个点计算标定用户坐标系参数。

**CalcFrame** p1 p2 p3 coord [r]

指令参数说明表：

参数名称	数据类型	参数含义
p1: 原点 (the original position)	Cpos	第一个标定点, 用户坐标系原点。
p2: RX方向标定点 (the second position)	Cpos	第二个标定点, 用户坐标系RX方向标定点。
p3: RXRY平面标定点 (the third position)	Cpos	第三个标定点, 用户坐标系RXRY平面标定点。
coord: 用户参数 (UserCoord Param)	Frame	计算标定出的用户坐标系参数。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。   <b>注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例：

```
CalcFrame(p1, p2, p3, S.FRAME[1])
```

运行该指令后, 可根据提供的三个点信息计算标定出用户坐标系参数, 并保存于S.FRAME[1]中。

### 3.5.11 SetRtToErr

该指令用于设置机器人停止运行并发出错误信息。

**SetRtToErr** msg error\_id [r]

指令参数说明表：

参数名称	数据类型	参数含义
msg: 错误信息 (errInfo)	String或者String型变量	设置的错误信息。
error_id: 错误编号 (errId)	Int或者Int型变量	设置的错误编号, 取值范围为90001-100000。

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

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

### 3.5.12 SetRtInfo

该指令用于设置提示信息但不影响程序运行。

**SetRtInfo** msg info\_id

指令参数说明表:

参数名称	数据类型	参数含义
msg: 提示信息 (Info)	String或者String型变量	设置的错误信息。
info_id: 提示编号 (info_id)	Int或者Int型变量	设置的提示编号, 取值范围为70001-80000。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

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

### 3.5.13 SetRtWarning

该指令用于设置警告信息但不影响程序的运行。

**SetRtWarning** msg warn\_id [r]

指令参数说明表：

参数名称	数据类型	参数含义
msg: 警告信息 (errInfo)	String或者String型变量	设置的错误信息。
warn_id: 警告编号 (warnId)	Int或者Int型变量	设置的警告编号, 取值范围为80001-90000。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例：

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

### 3.5.14 Stop

该指令用于停止所有已运行的程序的执行, 同时停止机器人的运动。

**Stop [r]**

指令参数说明表：

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例：

```
Stop()
停止所有执行中的程序。
```

### 3.5.15 SetMotionMode

该指令用于设置机器人运动模式。

**SetMotionMode mode [r]**

指令参数说明表：

参数名称	数据类型	参数含义
mode: 运动模式 (Mode)	String	设置机器人的运动模式 <ul style="list-style-type: none"> <li>• STANDARD: 标准模式。</li> <li>• LOW_SPEED_ACCURACY: 低速高精模式。</li> <li>• DENSE_POINT: 小线段模式:</li> <li>• USER_MODE1: 用户模式1。</li> <li>• USER_MODE2: 用户模式2。</li> <li>• USER_MODE3: 用户模式3。</li> <li>• USER_MODE4: 用户模式4。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例：

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

### 3.5.16 AutoTune

该指令用于用户自定义模式参数的自动调节。

**AutoTune tuneType [r]**

指令参数说明表：

参数名称	数据类型	参数含义
tuneType: 自调节功能类型	String	提速自调节 <ul style="list-style-type: none"> <li>• ADP: ADP模式。</li> </ul>

### 3.5.17 AutoGainEnable

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。

**AutoGainEnable scale thresh [r]**

指令参数说明表：

参数名称	数据类型	参数含义
scale: 低速段增益比例 (Ls-Scale)	LsScale类型变量	设置机器人各个关节轴的低速段增益比例阈值。
thresh: 低速段速度阈值 (LsThresh)	LsThresh类型变量	设置机器人各个关节轴的低速段速度阈值。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例：

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

### 3.5.18 AutoGainDisable

关闭伺服参数增益指令，该指令执行后系统将关闭对运动指令低速段的优化处理。

#### AutoGainDisable [r]

指令参数说明表：

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例：

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

### 3.5.19 SetRestorePC

该指令可设置非主task程序由暂停/停止转到运行时的pc指针。请注意，该指令只在非主task程序中使用有效。

### SetRestorePC index [r]

指令参数说明表：

参数名称	数据类型	参数含义
index: 程序指针 (PC index)	int	非主task程序恢复运行时的行号。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例：

```
SetRestorePC(3)
/*待非主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。*/
```

## 3.5.20 SetOriMode

姿态模式设置指令，该指令能够在机器人执行运动轨迹时，以某种姿态模式进行插补运动。

### SetOriMode oriMode [r]

指令参数说明表：

参数名称	数据类型	参数含义
oriMode: 姿态模式 (orientation Mode)	String	设置当前轨迹的姿态模式。 <ul style="list-style-type: none"> <li>Standard: 按照标准模式运动到目标点。</li> <li>Teach: 按照实际示教点位运动到目标点。</li> <li>Shortest: 按照J4最少旋转量运动到目标点，若角度差小于等于180°，则J4完全运动到目标点；若J4旋转量大于180，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。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例：

```

SetOriMode("Standard")
MovL(P1)/*姿态优化为标准模式。*/

SetOriMode("Teach")
MovL(P2)/*姿态优化为Teach模式，按实际示教点位运动。*/
SetOriMode("Shortest")/*姿态优化为Shortest模式，J4以最少旋转量到达目标点。*/

MovL(P3)

```

### 3.5.21 SetCircMode

圆弧模式设置指令，该指令能够在机器人执行圆弧运动轨迹时，以某种姿态模式进行插补运动。

**SetCircMode** circMode [r]

指令参数说明表：

参数名称	数据类型	参数含义
circMode: 圆弧模式 (orientation Mode)	String	设置当前圆弧轨迹的姿态模式。 <ul style="list-style-type: none"> <li>• MidPosePass: 开启圆弧过中间辅助点姿态插补的功能，优化必须将整圆拆分成3-4段圆弧才能保证焊接姿态的问题。</li> <li>• Standard: 关闭圆弧过中间辅助点姿态插补的功能。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。 <hr/>

示例：

```

SetCircMode("MidPosePass")
MovC(P1,P2)
/*开启圆弧过中间辅助点姿态插补的功能，使得机器人走MovC(P1,P2)指令时能够按照优化的效果运动。*/

```

### 3.5.22 GetRobotIsMoving

该指令可获取机器人当前的运动状态。

mov\_status = **GetRobotIsMoving** [r]

指令参数说明表：

参数名称	数据类型	参数含义
mov_status: 机器人运动状态 (Robot Moving Status)	Bool变量	存储机器人当前的运动状态。true为运动状态, false为非运动状态。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
Bool0 = GetRobotIsMoving()
/*获取系统当前的运动状态, 并将返回的值存储在Bool0中。*/
```

### 3.5.23 SetColliAutoTune

该指令用于设置碰撞检测自调整功能, 当碰撞检测参数自调整阶段选择Start、Pause、Continue、End时, 将对自调整功能的数据记录状态进行设置; 当碰撞检测参数自调整阶段选择Calculate时, 将根据自调整功能记录的数据进行计算得到碰撞自调整参数。

**SetColliAutoTune** cmdType method level colliid [r]

指令参数说明表:

参数名称	数据类型	参数含义
cmdType: 碰撞检测参数自调整阶段 (Collicmd Type)	String	自调整功能的数据记录状态。 <ul style="list-style-type: none"> <li>• Start: 重新开始。</li> <li>• Pause: 暂停。</li> <li>• Continue: 继续。</li> <li>• End: 结束。</li> <li>• Calculate: 计算生成。</li> </ul>
method: 检测方式 (Method)	String	自调整使用的碰撞检测方式。 <ul style="list-style-type: none"> <li>• Default: 默认。</li> <li>• Default+: 强。</li> <li>• Default++: 更强。</li> </ul> <b>⚠ 注意</b> 该参数仅在碰撞检测参数自调整阶段设为Calculate时才有效。

参数名称	数据类型	参数含义
level: 灵敏度 (level)	String	<p>自调整结果的灵敏度等级。</p> <ul style="list-style-type: none"> <li>• Highest: 最高。</li> <li>• High: 高。</li> <li>• Default: 默认。</li> <li>• Low: 低。</li> <li>• Lowest: 最低。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数仅在碰撞检测参数自调整阶段设为Calculate时才有效。</p>
colliid: 碰撞参数 (Colliid)	Colli-param变量	<p>存储自调整计算结果的碰撞参数。</p> <hr/> <p><b>⚠ 注意</b> 该参数仅在碰撞检测参数自调整阶段设为Calculate时才有效。</p>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p>

### 3.5.24 SetColliParam

该指令用于设置碰撞检测参数。

**SetColliParam** colliid axisid valid value [r]

指令参数说明表:

参数名称	数据类型	参数含义
colliid: 碰撞参数 (Colli id)	Colli-param变量	碰撞检测功能使用的参数。该参数可缺省, 默认为DEFAULT (出厂默认值)。

参数名称	数据类型	参数含义
axisid: 轴号 (AxisID)	String	需要设置碰撞检测有效性与灵敏度值的轴号。 <ul style="list-style-type: none"> <li>• DEFAULT: 无意义。</li> <li>• GROUP: 所有轴</li> <li>• A1: 1轴</li> <li>• A2: 2轴。</li> <li>• A3: 3轴。</li> <li>• A4: 4轴。</li> <li>• A5: 5轴。</li> <li>• A6: 6轴。</li> </ul>
valid: 有效性 (Valid)	String	控制对应轴号碰撞检测功能的开启与关闭。 <ul style="list-style-type: none"> <li>• ENABLE: 开启碰撞检测。</li> <li>• DISABLE: 关闭碰撞检测。</li> </ul> <hr/> <b>⚠ 注意</b> 该参数仅在轴号设非为DEFAULT时才有效。 <hr/>
value: 灵敏度值 (Value)	Int	碰撞检测的灵敏度值, 单位(%), 取值范围1~500。 <hr/> <b>⚠ 注意</b> 该参数仅在轴号设非为DEFAULT时才有效。 <hr/>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

### 3.5.25 SetAxisVibraBLevel

振动抑制等级设置命令, 只对自学习振动抑制有效。

**SetAxisVibraBLevel** axisid vibraLevel [r]

指令参数说明表:

参数名称	数据类型	参数含义
axisid: 轴号 (AxisID)	String	将要设置振动抑制等级的轴号。 <ul style="list-style-type: none"> <li>• A1: 1轴。</li> <li>• A2: 2轴。</li> <li>• A3: 3轴。</li> <li>• A4: 4轴。</li> <li>• A5: 5轴。</li> <li>• A6: 6轴。</li> </ul>
vibraLevel: 等级 (Calibration Level)	enum (Highest/High/Default/Low/Lowest)	将要设置振动抑制的等级。 <ul style="list-style-type: none"> <li>• Highest: 高。</li> <li>• High: 中高。</li> <li>• Default: 标准。</li> <li>• Low: 中低。</li> <li>• Lowest: 低。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

### 3.5.26 SetVibrationParam

设置振动抑制自学习参数。通过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。

#### SetVibrationParam VibrationParam [r]

指令参数说明表:

参数名称	数据类型	参数含义
VibraParam: 振动抑制自学习参数	Vibration-Param	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。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

### 3.5.27 RecordUserInfo

该指令用于在用户的日志文件中记录用户输入的信息，执行该指令不影响程序运行。

**RecordUserInfo** msg [r]

指令参数说明表：

参数名称	数据类型	参数含义
msg: 提示信息 (Info)	String或String型变量	用户想要写进用户日志文件的记录信息。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例：

```
RecordUserInfo ("Info")
将用户输入的“Info”字符串写入用户日志文件。
```

### 3.5.28 SetWristMoveEnable

该指令用于设置WristMove功能及腕部运动模式生效。

**SetWristMoveEnable** type

指令参数说明表：

参数名称	数据类型	参数含义
type: 运动模式 (move type)	enum ("Wrist45 / Wrist46 / Wrist56")	不同的运动模式。

示例：

```
SetWristMoveEnable("Wrist45")
```

### 3.5.29 SetWristMoveDisable

该指令用于设置WristMove功能及腕部运动模式生效。

**SetWristMoveDisable** type

指令参数说明表：

参数名称	数据类型	参数含义
type: 运动模式 (move type)	enum ("Wrist45 / Wrist46 / Wrist56")	不同的运动模式。

示例：

```
SetWristMoveDisable("Wrist45")
```

### 3.6 位置运算指令

位置运算指令列表：

位置运算指令	GetCurAPos
	GetCurCPos
	APosToCPos
	CPosToAPos
	CPosToCPos
	ToolOffset
	FrameOffset
	CalPosOffset
	PosOffset

#### 3.6.1 GetCurAPos

该指令用于获取机器人关节坐标系下的当前位置，保存于Apos类型变量。

**GetCurAPos** point [r]

指令参数说明表：

参数名称	数据类型	参数含义
• point: 关节位置 (Joint Position)	Apos	机器人关节坐标系下的当前位置。

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
GetCurAPos(p1)
获取机器人关节坐标系下的当前位置, 并保存于p1中。
```

### 3.6.2 GetCurCPos

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，保存于Cpos类型变量。

**GetCurCPos** point [r]

指令参数说明表:

参数名称	数据类型	参数含义
point: 笛卡尔空间位置 (Cartisian Position)	Cpos	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。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
GetCurCPos(p1)
获取机器人当前坐标系（世界坐标系或用户坐标系）下的笛卡尔空间位置, 并保存于p1中。
```

### 3.6.3 APosToCPos

该指令用于机器人位置点转换，根据给定的Apos点，可以将Apos点转换为Cpos点。

**APosToCPos** original\_point target\_point [r]

指令参数说明表:

参数名称	数据类型	参数含义
original_point: 原位置 (Original Position)	Apos	转换前关节位置。
target_point: 目标位置 (Target Position)	Cpos	目标笛卡尔空间位置。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

APosToCPos (p1, p2)  
根据p1, 可以得到目标点p2。

### 3.6.4 CPosToAPos

该指令用于机器人位置点转换指令, 给定Cpos点, 可以将Cpos点转换为Apos点。

**CPosToAPos** original\_point target\_point [r]

指令参数说明表:

参数名称	数据类型	参数含义
original_point: 原位置 (Original Position)	Cpos	转换前的笛卡尔空间位置。
target_point: 目标位置 (Target Position)	Apos	转换后的关节位置。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:


APosToCPos(p1, p2)  
根据p1, 可以得到目标点p2。

### 3.6.5 CPosToCPos

该指令用于机器人位置点转换指令，可以将给定一个Cpos点转换为另一个Cpos点，两个Cpos点中含有不同的用户坐标系或工具坐标系信息。

**CPosToCPos** original\_point target\_point [r]

指令参数说明表：

参数名称	数据类型	参数含义
original_point: 原位置 (Original Position)	: Cpos	转换前的笛卡尔空间位置。
target_point: 目标位置 (Target Position)	Cpos	目标笛卡尔空间位置。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。   <b>注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例：

```
CPosToCPos(p1, p2)
根据p1, 可以得到目标点P2。
```

### 3.6.6 ToolOffset

通过该指令可对基准工具坐标系进行旋转或者偏移后，生成一个新的工具坐标系，其后使机器人TCP在修正后的坐标系下进行位置移动，无需重复对点，可以有效提高现场调试效率，降低调试难度。

**ToolOffset** base\_tool offset target\_tool [r]

指令参数说明表：

参数名称	数据类型	参数含义
base_tool: 基准工具参数 (Base Tool)	Tool	转换前的基准工具坐标系参数。
offset: 偏移量 (Offset)	DCpos	新工具坐标系相对于原始工具参数的偏移量。
target_tool: 目标工具参数 (Target Tool)	Tool	基于基准工具参数偏移后获得的新工具参数。

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
ToolOffset(S.TOOL[1], DCpos1, S.TOOL[2])
根据工具坐标系S.TOOL[1]、偏移量DCpos1, 可以得到偏移后的工具坐标系S.TOOL[2]。
```

### 3.6.7 FrameOffset

通过该指令可对基准用户坐标系进行旋转或者偏移后, 生成一个新的用户坐标系, 其后使机器人TCP在修正后的坐标系下进行位置移动, 无需重复对点, 可以有效提高现场调试效率, 降低调试难度。

**UserOffset** base\_frame offset target\_frame [r]

指令参数说明表:

参数名称	数据类型	参数含义
base_frame: 用户坐标系 (Base Frame)	Frame	转换前的用户坐标系参数。
offset: 偏移量 (Offset)	DCpos	新用户坐标系相对于原始用户坐标系参数的偏移量。
target_frame: 目标用户坐标系参数 (Target Frame)	Frame	基于基准用户坐标系偏移后获得的新用户坐标系参数。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
UserOffset(S.Frame[1], DCpos1, S.Frame[2])
根据工具坐标系S.Frame[1]、偏移量DCPOS1, 可以得到偏移后的用户坐标系S.Frame[2]。
```

### 3.6.8 CalPosOffset

计算位置点偏移结果指令，该指令由原位置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** op rp t [tool] [frame] tp [r]

指令参数说明表：

参数名称	数据类型	参数含义
op: 原位置 (Original Position)	Apos/Cpos	偏移运动的起点。
rp: 相对偏移量 (Rel Position)	DCpos	相对原位置所要移动的位置增量。
t: 参考坐标系 (RefSys)	String	参考坐标系：笛卡尔或工具。 <ul style="list-style-type: none"> <li>• FRAME: 用户坐标系。</li> <li>• TOOL: 工具坐标系。</li> </ul>
tool: 目标工具参数 (Target Tool Param)	Tool	目标工具参数，当op类型为Apos时才有这个参数。
frame: 目标坐标系 (Target Frame)	Frame	参考坐标系，当op类型为Apos时才有这个参数。
tp: 目标位置 (Target Position)	Cpos	目标笛卡尔空间位置。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。 <hr/>

示例：

```
CalPosOffset(p0, DCpos0, "FRAME", p1)
根据p0、DCpos0，可以得到目标点p1。
```

### 3.6.9 PosOffset

通过该指令可对程序域中所有点位变量的指定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。

**PosOffset** mode destaxis pos srcaxis [r]

指令参数说明表：

参数名称	数据类型	参数含义
mode: 操作模式 (Operate Mode)	String	<p>需要对指定轴坐标进行的偏移方式。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>
destaxis: 目标轴号 (Target Axis number)	Int	需要进行坐标偏移的附加轴号。
pos: 目标坐标值 (Target position)	Float	需要将目标轴坐标设置的目标值该参数仅在操作模式为"UpdateSingleAxisPos"才有效。
srcaxis: 基准轴号 (Src Axis number)	Int	需要将目标轴坐标互换的基准附加轴号, 该参数仅在操作模式为"ExchangeAxisPos"才有效。
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。</p> <hr/>

**⚠ 注意**

该指令会更新变量文件, 同时会重新将内存中程序域的变量数据复位成程序变量文件中的数据, 在程序中使用时需注意。

示例:

```
PosOffset ("UpdateSingleAxisPos", 7, 10)
将当前执行的程序文件中所有的点位变量的第7轴坐标设置成10, 并重新从变量文件中更新数据到内存中。
```

### 3.7 时钟指令

时钟指令列表:

时钟指令	CLKStart
	CLKStop
	CLKRead
	CLKReset

### 3.7.1 CLKStart

该指令用于启动指定时钟（启动后，可以从变量列表中看到指定时钟变量的state为true）。

#### CLKStart clock

指令参数说明表：

参数名称	数据类型	参数含义
clock: 时钟 (Clock)	Clock	需要启动的时钟变量。

示例：

```
CLKStart(Clock0)  
启动时钟Clock0。
```

### 3.7.2 CLKStop

该指令停止指定时钟（其state为false，但不会复位）。

#### CLKStop clock

指令参数说明表：

参数名称	数据类型	参数含义
clock: 时钟 (Clock)	Clock	需要停止的时钟变量。

示例：

```
CLKStop(Clock0)  
停止时钟Clock0。
```

### 3.7.3 CLKRead

该指令用于读取指定时钟的值，在变量列表查看变量对应的value即可。

#### CLKRead clock

指令参数说明表：

参数名称	数据类型	参数含义
clock: 时钟 (Clock)	Clock	需要读取的时钟变量。

示例：

CLKRead(Clock0)  
读取时钟存于Clock0。

### 3.7.4 CLKReset

该指令用于复位指定时钟的状态值。

**CLKReset** clock

指令参数说明表：

参数名称	数据类型	参数含义
clock: 时钟 (Clock)	Clock	需要复位的时钟变量。

示例：

CLKReset(Clock0)  
复位时钟Clock0。

## 3.8 区域指令

区域指令列表：

区域指令	AreaActivate
	AreaDeactivate
	AreaConnectIO
	AreaDisConnectIO

### 3.8.1 AreaActivate

该指令用于激活标准区域，使指定的标准区域生效。

**AreaActivate** area\_var [r]

指令参数说明表：

参数名称	数据类型	参数含义
area_var: 标准区域变量 (Area)	AREA类型变量	需要激活的标准区域变量。

参数名称	数据类型	参数含义
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
AreaActivate(S.AREA[0])
激活标准区域监控, 使S.AREA[0]生效。
```

### 3.8.2 AreaDeactivate

该指令用于冻结标准区域, 使指定的标准区域失效。

**AreaDeactivate** area\_var [r]

指令参数说明表:

参数名称	数据类型	参数含义
area_var: 标准区域变量 (Area)	AREA类型变量	需要冻结的标准区域变量。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例:

```
AreaDeactivate(S.AREA[0])
冻结标准区域监控, 使S.AREA[0]失效。
```

### 3.8.3 AreaConnectIO

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。

**AreaConnectIO** area\_var io\_type io\_port io\_polarity

指令参数说明表:

参数名称	数据类型	参数含义
area_var: 标准区域变量 (Area)	AREA类型变量	需要绑定的标准区域变量。
io_type: IO类型 (IoType)	String	需要绑定的IO口类型。 <ul style="list-style-type: none"> <li>DI: 实际DI。</li> <li>SIMDI: 虚拟DI。</li> </ul>
io_port: IO端口号 (IoPort)	Int	需要绑定的IO端口号。
io_polarity: IO极性 (IoPolarity)	String	需要绑定的IO极性。 <ul style="list-style-type: none"> <li>HIGH: 绝对过渡。</li> <li>LOW: 缺省类型。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

### 3.8.4 AreaDisconnectIO

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

**AreaDisconnectIO** area\_var [r]

指令参数说明表:

参数名称	数据类型	参数含义
area_var: 标准区域变量 (Area)	AREA类型变量	需要解除绑定的标准区域变量。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。 <hr/>

## 3.9 码垛指令

码垛指令列表:

码垛指令	PalletReset
	PalletToPut
	PalletFromPut

	PalletToGet
	PalletFromGet

### 3.9.1 PalletReset

该指令用于重置垛体信息。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。

**PalletReset** PalletName actPart [r]

指令参数说明表：

参数名称	数据类型	参数含义
PalletName: 码垛名 (Pallet Name)	Pallet变量	当前垛体信息变量。
actPart: 垛体已有工件数 (ActPart)	Int	当前码垛上已经放置的工件数量。设置区间[0,垛体的最大工件数]。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

示例1：

```
PalletReset (Pallet0,8)
```

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。

### 3.9.2 PalletToPut

该指令用于放置工件。

**PalletToPut** PalletName [V] [B] [C] [r]

#### 说明

带[]参数为可选参数)

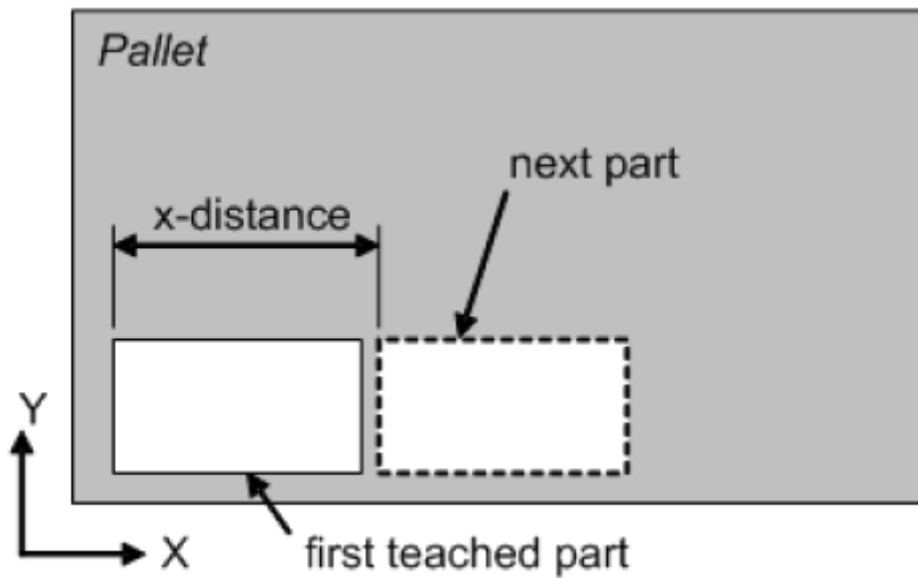
指令参数说明表：

参数名称	数据类型	参数含义
PalletName: 码垛名 (Pallet Name)	Pallet变量	当前垛体信息变量。
[V]: 目标速度 (Velo)	Speed变量	用于指定机器人执行PalletToPut时速度, 包括机器人末端的平移速度, 旋转速度以及外部轴的移动速度等。 <b>⚠ 注意</b> 该参数可缺省, 默认值为V4000。
[B]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省。
[C]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。 <b>⚠ 注意</b> 该参数可缺省, 默认值为C100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <b>⚠ 注意</b> 该选项仅在多机器控制时使用; 该参数可缺省, 默认为当前示教器操作的机器人。

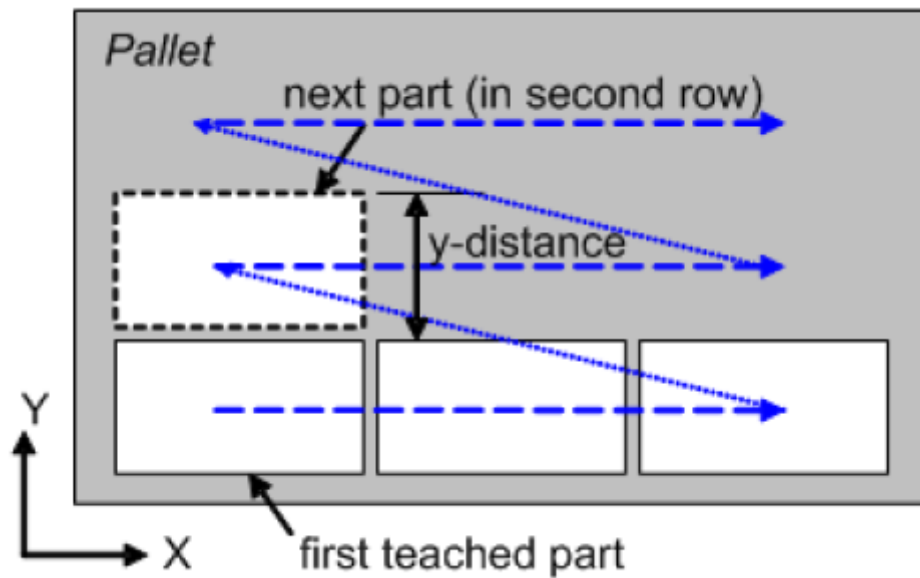
示例1:

```
PalletToPut(Pallet1, S.V2000)
```

码垛功能的垛位计算取决于示教的第一个工件位置和配置的码垛的距离。在前一个工件的位置基础上添加设定的码垛距离, 从而得到下一个工件的位置, 如下图所示:



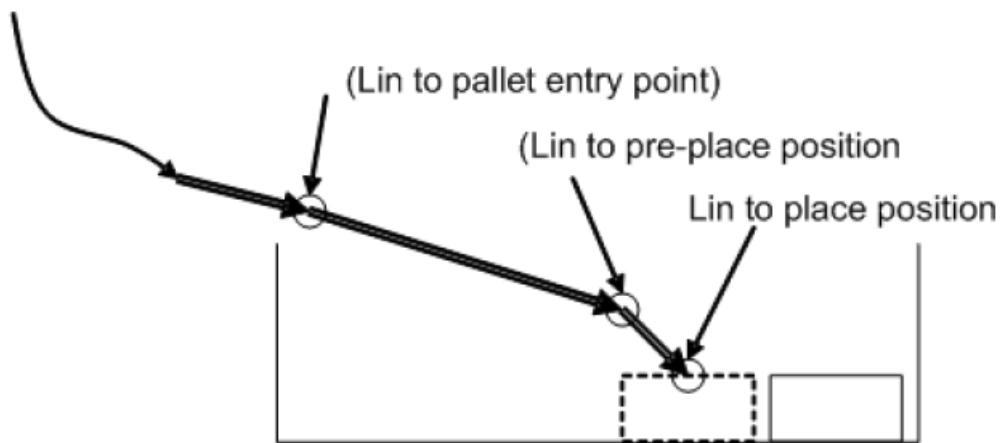
在下图中，默认码垛的顺序从x开始。首先放置x方向里的所有部件，然后再开始放置下一行。在x方向上放置工件直到达到设置好的x方向工件数。



当一个新行开始放置时，在第一行工件位置的基础上添加y方向上的偏移即生成了新行的工件的位置数据。

这个例子默认的码垛顺序是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）

### 3.9.3 PalletFromPut

该指令用于放置工件完成后，机器人从放置位置离开。

**PalletFromPut** PalletName [V] [B] [C] [r]

#### ⚠ 注意

(备注：带[]参数为可选参数)

指令参数说明表：

参数名称	数据类型	参数含义
PalletName: 码垛名 (Pallet Name)	Pallet变量	当前垛体信息变量。
[V]: 目标速度 (Velo)	Speed变量	用于指定机器人执行PalletFromPut时速度，包括机器人末端的平移速度，旋转速度以及外轴的移动速度等。  ⚠ 注意 该参数可缺省，默认值为V4000。

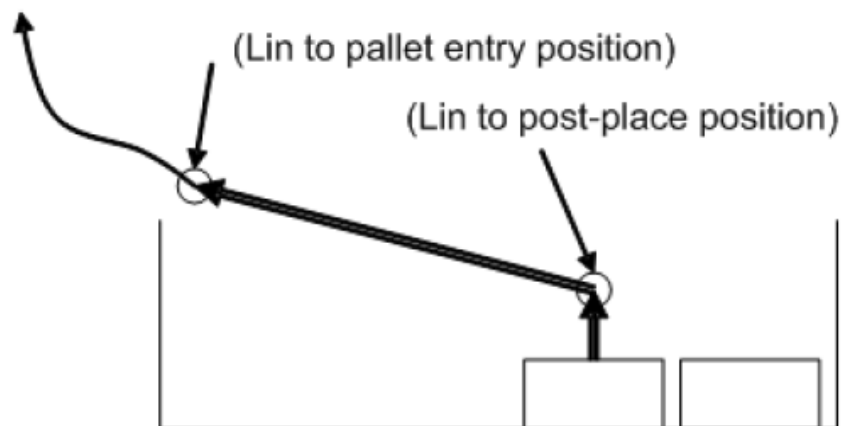
参数名称	数据类型	参数含义
[B]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <hr/> <b>⚠ 注意</b> 该参数可缺省。
[C]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。 <hr/> <b>⚠ 注意</b> 该参数可缺省，默认值为C100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例1:

```
PalletFromPut(Pallet1, S.V2000)
```

该指令是放置工件的相反动作，从放置好的工件位置离开。

离开时，建议路径如下图所示：



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。

### 3.9.4 PalletToGet

该指令用于抓取工件。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。

**PalletToGet** PalletName [V] [B] [C] [r]

#### 注意

(备注：带[]参数为可选参数)

指令参数说明表：

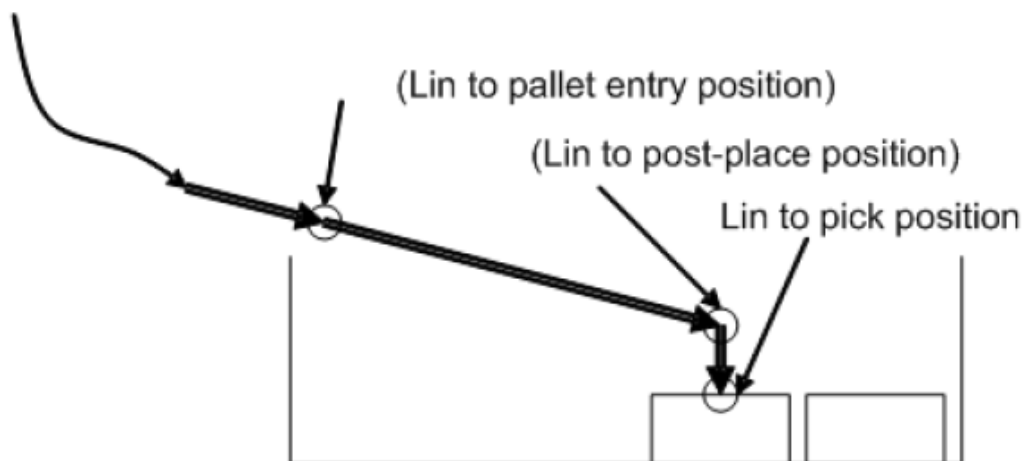
参数名称	数据类型	参数含义
PalletName: 码垛名 (Pallet Name)	Pallet变量	当前垛体信息变量。
[V]: 目标速度 (Velo)	Speed变量	用于指定机器人执行PalletToGet时速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。   <b>注意</b> 该参数可缺省，默认值为V4000。
[B]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul>  <b>注意</b> 该参数可缺省

参数名称	数据类型	参数含义
[C]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。 <hr/> <b>⚠ 注意</b> 该参数可缺省，默认值为C100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。 <hr/> <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例1：

```
PalletToGet(Pallet1,S.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前点位置（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。

### 3.9.5 PalletFromGet

该指令用于抓取工件动作后离开。机器人执行该指令时，会从抓取工件的码垛点离开。

**PalletFromGet** PalletName[V] [B] [C] [r]**⚠ 注意**

(备注：带[]参数为可选参数)

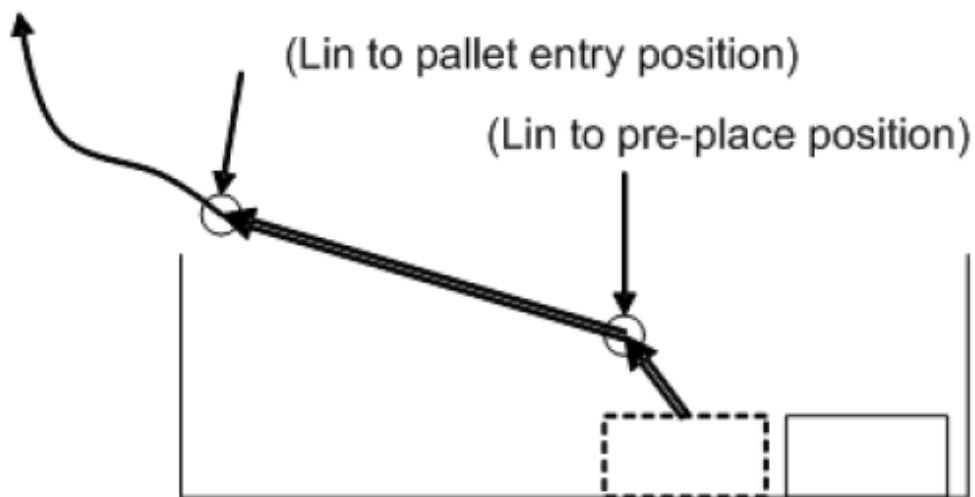
指令参数说明表：

参数名称	数据类型	参数含义
PalletName: 码垛名 (Pallet Name)	Pallet变量	当前垛体信息变量。
[V]: 目标速度 (Velo)	Speed变量	用于指定机器人执行PalletFromGet时速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。  <b>⚠ 注意</b> 该参数可缺省，默认值为V4000。
[B]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <b>⚠ 注意</b> 该参数可缺省
[C]: 过渡值 (BlendValue)	Zone	机器人逼近终点时的过渡值。  <b>⚠ 注意</b> 该参数可缺省，默认值为C100。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例1：

PalletFromGet(Pallet1,S.V2000)

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后点位置（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。

### 3.10 Socket指令

Socket指令列表：

Socket指令	SocketCreateServer
	SocketAccept
	SocketCreateClient
	SocketReceive
	SocketSend
	SocketClose
	UDPCreate
	UDPBind
	UDPReceive
	UDPSend

#### 3.10.1 SocketCreateServer

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，在本地创建服务器。

**SocketCreateServer** server ip port ret

指令参数说明表：

参数名称	数据类型	参数含义
server: TcpServer标识	TcpServer变量	该指令会设置TcpServer变量的参数值，存放服务器信息。
ip: IP地址 (IP address)	String变量或常量	创建的服务器IP地址。
port: 端口号 (Port)	Int变量或常量	创建的服务器端口号。
ret: 操作返回值 (Return value)	ReturnValue变量	返回值变量，返回值的int_value参数为对应执行结果。 <ul style="list-style-type: none"> <li>0: 成功;</li> <li>1: 失败。</li> </ul>

示例：

```
SocketCreateServer(G.TcpServer0,"192.168.1.2",9000,ReturnValue0)
创建一个TCP
Server，标识为全局变量TcpServer0，IP为“192.168.1.2”，端口为9000，操作返回值存放在
ReturnValue0中。
```

**3.10.2 SocketAccept**

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。

**SocketAccept** server client ret

指令参数说明表：

参数名称	数据类型	参数含义
server: TcpServer标识	TcpServer变量	SocketCreateServer创建的变量，用来监听及接收客户端连接。
client: TcpSocket标识	TcpSocket变量	用来存放接收的客户端Socket连接信息，后面可以通过该参数与客户端进行数据交互。
ret: 操作返回值 (Return value)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

```
ocketAccept(G.TcpServer0,G.TcpSocket0,ReturnValue0)
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。
```

### 3.10.3 SocketCreateClient

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，在本地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。

**SocketCreateClient** client ip port ret

指令参数说明表：

参数名称	数据类型	参数含义
client: TcpSocket标识	TcpSocket变量	该指令会修改client参数值，存放TCP客户端连接信息。
IP地址 (IP address)	String变量或常量	需要连接的服务器IP地址。
port: 端口号 (Port)	Int变量或常量	需要连接的服务器端口号。
ret: 操作返回值 (Return value)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

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

### 3.10.4 SocketReceive

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。

**SocketReceive** socket timeout ret result

指令参数说明表：

参数名称	数据类型	参数含义
socket: TcpSocket标识	TcpSocket变量	SocketAccept或SocketCreateClient创建的TcpSocket变量。
timeout: 超时检测时间 (Timeout)	Int变量或常量	等待服务器端或客户端发送数据的超时时间 (ms)，超时则设置返回值为1。
ret: 操作返回值 (Return value)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。
result: 结果返回值 (Result)	String变量	用来存放接收的字符串数据。

示例：

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

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。

### 3.10.5 SocketSend

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。

**SocketSend** socket msg ret

指令参数说明表：

参数名称	数据类型	参数含义
socket: TcpSocket标识	TcpSocket变量	SocketCreateClient或SocketAccept创建的TcpSocket变量。
msg: 发送的数据 (Message)	String变量或常量	需要发送的字符串数据
ret: 操作返回值 (ReturnValue)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

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

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

### 3.10.6 SocketClose

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

**SocketClose** socket ret

指令参数说明表：

参数名称	数据类型	参数含义
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。
ret: 操作返回值 (ReturnValue)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

SocketClose(G.TcpServer0,ReturnValue0)  
关闭G.TcpServer0，返回值存放在ReturnValue0中。

### 3.10.7 UDPCreate

创建UDP Socket，用于UDP数据交互。

**UDPCreate** socket ret

指令参数说明表：

参数名称	数据类型	参数含义
socket: UdpSocket标识	UdpSocket变量。	该指令会设置UdpSocket变量，存放UDP Socket信息。
ret: 操作返回值 (ReturnValue)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

UDPCreate(G.UdpSocket0,ReturnValue0)  
创建一个UdpSocket，并把信息存放在G.UdpSocket0中，操作返回值存放在ReturnValue0中。

### 3.10.8 UDPBind

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，如果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。

**UDPBind** socket ip port ret

指令参数说明表：

参数名称	数据类型	参数含义
socket: UdpSocket标识	UdpSocket变量。	UDPCreate创建的UdpSocket，用于绑定UDP接收端的IP地址和端口号。
ip: IP地址 (IP address)	String变量或常量	需要绑定的UDP接收端IP地址。
port: 端口号 (Port)	Int变量或常量	需要绑定的UDP接收端端口号。
ret: 操作返回值 (ReturnValue)	ReturnValue变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

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

### 3.10.9 UDPReceive

接收UDP对端发来的字符串数据。

**UDPReceive** socket timeout ret result

指令参数说明表：

参数名称	数据类型	参数含义
socket: UdpSocket标识	UdpSocket变量。	UDPBind指令执行之后的UdpSocket变量，用于接收对端发来的UDP数据。
timeout: 超时检测时间 (Timeout)	Int变量或常量	等待UDP对端发送数据的超时时间 (ms)，超时则设置返回值为1。
ret: 操作返回值 (Return value)	Return Value变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。
result: 结果返回值 (Result)	String变量	用来存放接收的字符串数据。

示例：

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

### 3.10.10 UDPSend

向UDP对端发送字符串数据。

**UDPSend** socket ip port msg ret

指令参数说明表：

参数名称	数据类型	参数含义
socket: UdpSocket标识	UdpSocket变量。	UDPCreate创建的变量，用于向UDP对端发送数据。
ip: IP地址 (IP address)	String变量或常量	UDP对端的IP地址。
port: 端口号 (Port)	Int变量或常量	UDP对端的端口号。
msg: 发送的数据 (Message)	String变量或常量	需要发送的字符串数据
ret: 操作返回值 (Return value)	Return Value变量	返回值变量，返回值的int_value为0表示成功，为1则为失败。

示例：

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

### Socket指令使用示例：

#### 1. TCP Socket服务器示例（机器人作为服务器）

```
function TCP_Server_Test()
# 返回值ReturnValue0.int_value存放指令执行的结果，0表示成功
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)
# 创建TCP服务器G.TcpServer0，指定IP地址和端口号
SocketCreateServer(G.TcpServer0,"172.31.16.100",9000,ReturnValue0)
# 服务器G.TcpServer0接收一个客户端连接并存入G.TcpSocket0
SocketAccept(G.TcpServer0,G.TcpSocket0,ReturnValue0)
# 接收客户端G.TcpSocket0发来的字符串并存入G.STRING0，超时时间10s
G.STRING0 = SocketReceive(G.TcpSocket0,10000,ReturnValue0)
# 向客户端G.TcpSocket0发送字符串“123”
SocketSend(G.TcpSocket0,"123",ReturnValue0)
# 关闭客户端连接G.TcpSocket0
SocketClose(G.TcpSocket0,ReturnValue0)
# 关闭服务器G.TcpServer0
SocketClose(G.TcpServer0,ReturnValue0)
end
```

#### 2. TCP Socket客户端示例（机器人作为客户端）

```
function TCP_Client_Test()
# 返回值ReturnValue0.int_value存放指令执行的结果，0表示成功
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)
# 创建TCP客户端G.TcpSocket0并连接到指定IP和端口号的服务器
SocketCreateClient(G.TcpSocket0,"172.31.16.200",9001,ReturnValue0)
# 接收服务器发来的字符串并存入G.STRING0，超时时间10s
G.STRING0 = SocketReceive(G.TcpSocket0,10000,ReturnValue0)
# 向服务器发送字符串“123”
SocketSend(G.TcpSocket0,"123",ReturnValue0)
# 关闭该客户端连接
SocketClose(G.TcpSocket0,ReturnValue0)
end
```

#### 3. UDP Socket示例(UDP不区分客户端和服务端)

```

function UDP_Test()
# 返回值ReturnValue0.int_value存放指令执行的结果，0表示成功
var ReturnValue0 :: ReturnValue = ReturnValue(0, false, "0", 0.0)
# 创建UDP Socket
UDPCreate(G.UdpSocket0,ReturnValue0)
# 为UDP绑定IP和端口号
UDPBind(G.UdpSocket0,"172.31.16.100",9002,ReturnValue0)
# 接收UDP对端发来的字符串并存入G.STRING0，超时时间10s
G.STRING0 = UDPReceive(G.UdpSocket0,10000,ReturnValue0)
# 向指定IP和端口号的UDP对端发送字符串" 123"
UDPSend(G.UdpSocket0,"172.31.16.200",9004,"123",ReturnValue0)
# 关闭该UDP Socket
SocketClose(G.UdpSocket0,ReturnValue0)
end

```

### 3.11 传送带跟随指令

传送带跟随指令列表:

传送带跟随指令	MovLSync
	MovJSyncQuit
	MovLSyncQuit
	WaitWObj
	SetTargetPos
	SyncCToUserC
	WaitConvDis
	SimConveyorOn
	SimConveyorOff
	ReceiveWObj
	ResetWObjBuf
	SyncToNextRob

#### 3.11.1 MovLSync

MovLSync指令为传送带同步跟随启动指令，通过该指令可以使机器人TCP点以设定的速度直线运动到目标位置。执行该指令前，机器人控制系统会通过IO触发视觉拍照并接收视觉系统发送的目标位置。通过执行该指令，机器人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。

**MovLSync** p h convey\_goto [v] [payload] [sync\_goto] [DO] [r]

**⚠ 注意**

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Cpos变量	相对于跟随目标物体的偏移量，只能为Cpos类型变量，在下拉列表中可选择已有的点。
h: 同步后放弃时抬升高度 (Height)	Float	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。
convey_goto: 同步后放弃时跳转标签 (ConvyGoto)	label	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。
[v]: 指令运行速度 (Velo)	Speed变量	<p>用于指定机器人执行MovLSync时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。</p> <ul style="list-style-type: none"> <li>DEFAULT: 使用默认值V4000。</li> <li>Speed类型变量: 系统预定义或自行创建。</li> </ul> <hr/> <p><b>⚠ 注意</b></p> <p>该参数可缺省，默认值为V4000。</p>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <ul style="list-style-type: none"> <li>设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。</li> <li>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。</li> </ul> <hr/> <p><b>⚠ 注意</b></p> <p>该参数可缺省，默认值为使用当前系统中设置的负载参数。</p>

参数名称	数据类型	参数含义
[sync_goto]: 同步过程中放弃时跳转标签 (SyncGoto)	label	<p>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。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，设置为缺省时会跳转到“同步后放弃时跳转标签”所用的Label处执行。</p> <hr/>
[DO]: 机器人执行完该指令后，可进行IO操作 (Add Do)	string	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。</p> <hr/>

### 3.11.2 MovJSyncQuit

MovJSyncQuit指令为传送带同步跟随退出指令，通过该指令可以使机器人退出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。

**MovJSyncQuit** p [v] [b] [payload] [quit\_enable] [DO] [r]

#### **⚠ 注意**

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或 Cpos变量	坐标系下的目标点位置。(可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。)

参数名称	数据类型	参数含义
[v]: 目标速度 (Velo)	Speed变量	<p>用于指定机器人执行MovJSyncQuit时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效。有外部轴时，Speed变量中的exj_l、exj_r分量有效。</p> <ul style="list-style-type: none"> <li>• 设置为“DEFAULT”，表示使用默认值V4000。</li> <li>• 设置为Speed类型变量时，可以选用系统预定义，也可以自行创建。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省，默认值为V4000。</p> <hr/>
[b]: 过渡类型 (BlendType)	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE：精确到位。</li> <li>• RELATIVE：相对过渡。</li> <li>• ABSOLUTE：绝对过渡。</li> <li>• DEFAULT：使用默认的过渡方式，即无过渡。</li> </ul>
[payload]: 工件负载 (Payload)	Payload变量	<p>机器人执行该轨迹时使用的工件负载参数。</p> <ul style="list-style-type: none"> <li>• 设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。</li> <li>• 设置为Payload类型变量时，可以选用系统预定义，也可以自行创建；使用与当前工件负载参数不同的负载参数时，系统会将工件负载参数切换到设置的工件负载参数。若本段与前后段轨迹的工件负载变化，则不支持过渡。</li> </ul>
[quit_enable]: 退出点使能 (QuitPEnable)	String	<p>执行完本次跟随轨迹，是否退出的使能状态。</p> <ul style="list-style-type: none"> <li>• DEFAULT：使用默认的使能方式，即ENABLE。</li> <li>• ENABLE：跟随结束后退出至指令参数中所设的目标位置。</li> <li>• DISENABLE：跟随结束后退出至机器人TCP当前所在位置，此时指令参数中所设的目标位置无效。</li> </ul>
[DO]: 运动完成后执行的IO操作 (Add Do)	string	<p>机器人执行完该指令后，可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL：无任何操作。</li> <li>• IO指令：执行IO操作，目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。</p> <hr/>

### 3.11.3 MovLsyncQuit


MovLsyncQuit指令为传送带同步跟随退出指令，通过该指令可以使机器人退出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。

**MovLsyncQuit** p [v] [b] [payload] [quit\_enable] [DO] [r]

#### 注意

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Apos或Cpos变量	坐标系下的目标点位置；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。
[v]: 目标速度 (Velo)	Speed变量	用于指定机器人执行MovLsyncQuit时的速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。此时，Speed变量中的tcp、ori分量有效；有外部轴时，Speed变量中的exj_l、exj_r分量有效。 <ul style="list-style-type: none"> <li>设置为“DEFAULT”，表示使用默认值V4000。</li> <li>设置为Speed类型变量时，可以选用系统预定义，也可以自行创建。</li> </ul>
[b]: 过渡类型 (BlendType)	String	机器人逼近终点时的过渡方式。 <ul style="list-style-type: none"> <li>DEFAULT: 使用默认的过渡方式，即无过渡。</li> <li>FINE: 精确到位。</li> <li>RELATIVE: 相对过渡。</li> <li>ABSOLUTE: 绝对过渡。</li> </ul> <hr/>  <b>注意</b> 该参数可缺省。
[payload]: 工件负载 (Payload)	Payload变量	机器人执行该轨迹时使用的工件负载参数。 <ul style="list-style-type: none"> <li>设置为“DEFAULT”，表示沿用最近一次设置的工件负载参数。</li> <li>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。</li> </ul>

参数名称	数据类型	参数含义
[quit_enable]: 退出点使能 (QuitPEnable)	String	<p>执行完本次跟随轨迹, 是否退出的使能状态。</p> <ul style="list-style-type: none"> <li>• DEFAULT: 使用默认的使能方式, 即ENABLE。</li> <li>• ENABLE: 跟随结束后退出至指令参数中所设的目标位置。</li> <li>• DISENABLE: 跟随结束后退出至机器人TCP当前所在位置, 此时指令参数中所设的目标位置无效。</li> </ul>
[DO]: 运动完成后执行的IO操作 (Add Do)	string	<p>机器人执行完该指令后。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>
[r]: 机器人名称 (Robot-Name)	String	<p>与本指令关联的机器人名称。</p> <hr/> <p><b>⚠ 注意</b></p> <p>该选项仅在多机器控制时使用;</p> <p>该参数可缺省, 默认为当前示教器操作的机器人。</p> <hr/>

### 3.11.4 WaitWObj

该指令为等待视觉信息指令, 用于获取视觉发来的目标工件位置, 该指令一般编写在 MovLSync指令的上一行。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。

**WaitWObj** coord id t ret attr [goto] [cam\_pos] [r]

指令参数说明表:

参数名称	数据类型	参数含义
coord: 坐标系参数 (SynCoord)	Frame变量	当前机器人所等待的工件位置信息所参考的动坐标系, 该动坐标系需与传送带配置界面中启用的动坐标系一致。
id: 工件id (WorkPieceID)	RETURN-VALUE类型变量	用来返回当前接收到的工件id号。
t: 时长 (Time)	Int	<p>表示执行等待时所需的时间, 单位为ms。</p> <ul style="list-style-type: none"> <li>• 如果该参数的值为0, 将强制等待, 直至收到工件位置信息数据, 才继续执行下一条指令。</li> <li>• 如果该参数的值非0, 如果未收到数据, 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。</li> </ul>

参数名称	数据类型	参数含义
ret: 超时判断值 (IsOver-time)	RETURN-VALUE类型变量	返回该指令执行结束后的状态，并在如下两种情况下为其赋值： <ul style="list-style-type: none"> <li>当该指令在等待时间内执行完成时，对该变量赋值0；</li> <li>当该指令由于超时执行完成时，对该变量赋值1。</li> </ul>
attr: 视觉反馈的Attr (Attr)	RETURN-VALUE类型变量	用来返回当前接收到的物体属性值。
[goto]: 超时跳转的标签名 (LabelName)	label	当执行超时时，选择跳转的标签名所在行；该参数可缺省，默认值为不跳转。
[cam_pos]: 视觉反馈的点位 (Camera Pos)	Cpos变量	用来返回当前接收到的坐标值；该参数可缺省。
[r]: 机器人名称 (Robot-Name)	String	与本指令关联的机器人名称。  <b>⚠ 注意</b> 该选项仅在多机器控制时使用； 该参数可缺省，默认为当前示教器操作的机器人。

示例1：

```
WaitWObj (S.FRAME[1], ReturnValue0, 1000, ReturnValue1, ReturnValue2, "Goto Start")
```

在1000ms时长内等待与动坐标系FRAME[1]绑定的传送带上外部设备发来的工件位置信息成功时，程序继续向下执行，若1000ms内未等到条件满足，则将ReturnValue1的int\_value值置1，同时程序指针将跳转到“Start”标签所在行(第一行)开始执行。

### 3.11.5 SetTargetPos

该指令为工件目标位置设置指令，用于设置工件进入传送带上被检测到的位置信息。用于给指定的传送带发送工件的位置信息，从而让机器人完成传送带跟随运动。在传送带界面中的触发类型选择为非视觉触发类型才可执行该指令。

#### SetTargetPos p

指令参数说明表：

参数名称	数据类型	参数含义
p: 目标位置 (Target Pos)	Cpos变量	表示工件进入传送带位置相对于动坐标系原点的偏移量，且该目标位置的坐标系参数必须为传送带配置界面所启用的动坐标系的ID值。  ⚠ <b>注意</b> 目前系统支持2D、3D视觉数据，当选择2D数据类型，仅X、Y、A三个数据有效。

### 3.11.6 SynCToUserC

参考坐标系转换指令，用于计算出用户坐标系的值以便在示教跟随同步后的轨迹点位时使用。

**SynCToUserC** sync p coord

指令参数说明表：

参数名称	数据类型	参数含义
sync: 动坐标系参数 (Syn-Coord)	FRAME变量	当前工件位置信息所参考的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。
p: 目标位置 (Target Pos)	DCpos变量	表示工件进入传送带位置而相对于动坐标系原点的偏移量。
coord: 用户坐标系参数 (Coord)	Frame变量	示教跟随同步后轨迹点位时参考的用户坐标系。

### 3.11.7 WaitConvDis

该指令为跟随位移等待指令，使机器人与目标物体同步运动指定位移后再执行后续的指令。该指令与Wait指令的区别是，该指令等待固定距离，而Wait指令等待固定时间。

**WaitConvDis** coord dis

指令参数说明表：

参数名称	数据类型	参数含义
coord: 动坐标系参数 (Coord)	Frame变量	当前工件位置信息所参考的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。
dis: 等待增量 (Increment)	数据类型: Float	机器人与目标物体保持同步的位移，直线传送带时单位是mm；圆盘传送带时单位是deg。

### 3.11.8 SimConveyorOn

虚拟传送带开启指令，可用于开启虚拟传送带，并对传送带的速度进行指定。

**SimConveyorOn** coord speed

指令参数说明表：

参数名称	数据类型	参数含义
coord: 动坐标系参数 (Coord)	Frame变量	当前工件位置信息所参考的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。
speed: 传送带速度 (ConveyorSpeed)	数据类型: Float	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。

### 3.11.9 SimConveyorOff

虚拟传送带关闭指令，可用于关闭虚拟传送带。

**SimConveyorOff** coord

指令参数说明表：

参数名称	数据类型	参数含义
coord: 动坐标系参数 (Coord)	Frame变量	当前工件位置信息所参考的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。

### 3.11.10 ReceiveWObj

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并行任务中发来的工件位置信息，避免出现误跟随的情况，满足不同的现场工艺。

**ReceiveWObj** coord enable\_sts

指令参数说明表：

参数名称	数据类型	参数含义
coord: 动坐标系参数 (Coord)	Frame变量	当前工件位置信息所参考的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。
enable_sts: 使能接收状态 (EnableSts)	String	使能机器人系统对工件位置信息的接收状态 <ul style="list-style-type: none"> <li>OFF: 使能机器人系统不接收工件位置信息。</li> <li>ON: 使能机器人系统接收工件位置信息。</li> </ul>

### 3.11.11 ResetWObjBuf

复位已接收到的工件数据队列指令，该指令可以将队列中从外部接收到且未被执行的工件信息清空。

**ReceiveWObj** coord [reset\_type] [id]

指令参数说明表：

参数名称	数据类型	参数含义
coord: 动坐标系参数 (Coord)	Frame变量	当前工件位置信息所参考的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。
[reset_type]: 复位类型 (Reset Type)	String	指定复位目标队列数据时的类型。 <ul style="list-style-type: none"> <li>• DEFAULT: 使用默认的复位类型，即ALL。</li> <li>• ALL: 清除队列中已接收到的全部数据。</li> <li>• SELECT: 清除队列中指定ID号的数据。</li> <li>• CURRENT: 清除队列中的第一个数据。</li> </ul> <hr/> <b>⚠ 注意</b> 该参数可缺省，默认值为ALL。
[id]: 需复位的视觉ID号 (ResetId)	Int	需要被复位的ID号。只有当复位类型设为“SELECT”时，该参数才有效。

### 3.11.12 SyncToNextRob

设置下一个继续跟随的机器人ID传送带关闭指令。

**SyncToNextRob** ID

指令参数说明表：

参数名称	数据类型	参数含义
ID: 下一个继续跟随的机器人ID	Int	下一个继续跟随的机器人ID。

## 3.12 数学运算函数

本节是对所能使用到的“数学函数”进行说明。

数学运算函数	sin
	cos
	tan

	asin
	acos
	atan
	atan2
	sinh
	cosh
	tanh
	log
	log10
	sqrt
	exp
	pow
	deg
	rad
	fmod
	floor
	random

### 3.12.1 sin

正弦三角函数。

- 参数1: Int或Float型变量或常量（单位为弧度）。
- 函数返回值: Float型结果。

### 3.12.2 cos

余弦三角函数。

- 参数1: Int或Float型变量或常量（单位为弧度）。
- 函数返回值: Float型结果。

### 3.12.3 tan

正切三角函数。

- 参数1: Int或Float型变量或常量（单位为弧度）。
- 函数返回值: Float型结果。

### 3.12.4 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。

### 3.12.5 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。

### 3.12.6 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。

### 3.12.7 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。

### 3.12.8 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。

### 3.12.9 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。

### 3.12.10 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。

### 3.12.11 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。

### 3.12.12 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。

### 3.12.13 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。

### 3.12.14 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。

### 3.12.15 pow

指数函数。

- 参数1: Int或Float型变量或常量, 底数。
- 参数2: Int或Float型变量或常量, 指数。
- 函数返回值: Float型结果。

### 3.12.16 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。

### 3.12.17 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。

### 3.12.18 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。

### 3.12.19 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。

### 3.12.20 random

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。

### 3.13 字符串指令

字符串指令列表如下:

字符串指令	byte
	char
	find2
	findEnd
	format
	getAt
	gsub
	len
	left
	lower
	right
	reverse
	strcmp
	trimLeft
	trimRight
	IToStr
	upper
	RToStr
	StrToI
	StrToR
StrToIntArray	
StrToFloatArray	
IntArrayToStr	
FloatArrayToStr	

### 3.13.1 byte

取字符串第n位的字符的ASCII码。

- 参数1: String型变量或常量。
- 参数2: Int型变量或常量。
- 函数返回值: Int型常量。

### 3.13.2 char

返回ASCII码对应的字符。

- 参数1: Int型变量或常量。
- 函数返回值: String型常量。

### 3.13.3 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)

### 3.13.4 findEnd

字符串反向查找指令, 在字符串中找到指定字符串最后出现的位置, 并返回索引号。

- 参数1: 被查找的源字符串, String型变量或常量。
- 参数2: 待查找的指定字符串, String型变量或常量。
- 函数返回值: 查找后的索引号, Int型变量。

### 3.13.5 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/Float/Int型变量或常量。
- 参数3: 待填充格式化控制符的参数, String/Float/Int型变量或常量。

- ... (参数无限制, 总长度只要不超过单个指令字符串长度即可)。
- 函数返回值: 成功分割并保存到数组中的个数, Int型变量。

格式化字符串以%开头, 支持以下用法:

```
%c - 接受一个数字,并将其转化为ASCII码表中对应的字符
%d, %i - 接受一个数字并将其转化为有符号的整数格式
%o - 接受一个数字并将其转化为八进制数格式
%u - 接受一个数字并将其转化为无符号整数格式
%x - 接受一个数字并将其转化为十六进制数格式,使用小写字母x
%X - 接受一个数字并将其转化为十六进制数格式,使用大写字母X
%f - 接受一个数字并将其转化为浮点数格式
%s - 接受一个字符串并按照给定的参数格式化该字符串
```

样例:

```
format("%c: %c", 83)      输出S
format("%d", 17.0)       输出+17
format("%05d", 17)       输出00017
format("%o", 17)         输出21
format("%u", 3.14)       输出3
format("%x", 13)         输出d
format("%X", 13)         输出D
format("%6.3f", 13)      输出13.000
format("%s", "monkey")   输出monkey
format("%10s", "monkey") 输出  monkey
```

### 3.13.6 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。

### 3.13.7 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。

### 3.13.8 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。

### 3.13.9 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。

### 3.13.10 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。

### 3.13.11 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。

### 3.13.12 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。

### 3.13.13 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。

### 3.13.14 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。

### 3.13.15 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。

### 3.13.16 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。

### 3.13.17 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。

### 3.13.18 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。

### 3.13.19 StrTol

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。

### 3.13.20 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。

### 3.13.21 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

样例:

```
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。
```

### 3.13.22 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

样例：

```
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。
```

### 3.13.23 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

样例：

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

### 3.13.24 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

样例：

```
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。
```

### 3.13.25 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。

样例：

```
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。
```

### 3.13.26 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。

样例：

```
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。
```

### 3.13.27 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。

样例：

```
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。
```

### 3.13.28 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。

样例：

```
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。
```

### 3.13.29 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。

样例：

```
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。请注意：传入的字符串格式必须与APosToStr格式一致。
```

### 3.13.30 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。

样例：

```
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。请注
意：传入的
字符串格式必须与CPosToStr格式一致。
```

## 3.14 ModbusTcp指令

ModbusTcp指令列表：

ModbusTcp指令	GetModConState
	ReadModbusReg
	WriteModbusReg

### 3.14.1 GetModConState

该指令用于获取机器人与外界使用modbus通讯的连接状态。

is\_connected = **GetModConState**

指令参数说明表：

参数名称	数据类型	参数含义
is_connected：连接状态 (Is Connected)	Bool变量。	返回当前的连接状态。

示例1：

```
Bool0 = GetModConState()
```

### 3.14.2 ReadModbusReg

该指令用于读取指定Modbus寄存器的值。

RegisterValue = **ReadModbusReg** RegisterID

指令参数说明表：

参数名称	数据类型	参数含义
RegisterID: 目标寄存器ID号 (Target Modbus Register ID)	Int, 取值范围为101~1500。	想要读取的寄存器ID号。
RegisterValue: 目标寄存器值 (Target Modbus Register Value)	Int变量。	返回读取后的寄存器值。

示例1:

```
Int0 = ReadModbusReg(101) // 读取101寄存器的值
```

### 3.14.3 WriteModbusReg

该指令用于设置指定Modbus寄存器的值。

**WriteModbusReg** RegisterID RegisterValue

指令参数说明表:

参数名称	数据类型	参数含义
RegisterID: 目标寄存器ID号 (Target Modbus Register ID)	Int, 取值范围为101~1500。	想要修改的寄存器ID号。
RegisterValue: 目标寄存器值 (Target Modbus Register Value)	Int变量。	想要修改的寄存器值。

示例1:

```
Int0 = 100  
WriteModbusReg(101, Int0) // 将101寄存器的值改为100
```

## 3.15 视觉指令

视觉指令列表:

视觉指令	TrigCam
	WaitFinishCam
	GetCamPos
	SendMessage

### 3.15.1 TrigCam

该指令用于发送触发字符串给相机。

**TrigCamcam\_id trig\_string**

指令参数说明表：

参数名称	数据类型	参数含义
cam_id: 相机索引	常量	支持多台相机。0表示默认的视觉界面设置连接参数匹配的相机，1-10表示需要配合九点标定中连接参数使用的相机。
trig_string: 触发字符串	String常量或变量。	由用户约定的视觉检测触发字符串。

**3.15.2 WaitFinishCam**

该指令用于实现等待延迟，避免视觉数据还未答复给机器人。

**WaitFinishCamcam\_id wait\_time**

指令参数说明表：

参数名称	数据类型	参数含义
cam_id: 相机索引	常量	支持多台相机。0表示默认的视觉界面设置连接参数匹配的相机，1-10表示需要配合九点标定中连接参数使用的相机。
wait_time: 等待时长	常量	用户设置等待时长，该参数的单位为ms。

**3.15.3 GetCamPos**

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。

**GetCamPoscam\_id ret**

指令参数说明表：

参数名称	数据类型	参数含义
cam_id: 相机索引	常量	支持多台相机。0表示默认的视觉界面设置连接参数匹配的相机，1-10表示需要配合九点标定中连接参数使用的相机。
ret: 返回值	VisionRet类型变量	相机回复的数据。

**3.15.4 SendMessage**

该指令用于按照指定格式发送消息字符串给相机设备。

**SendMessagecam\_id msg\_string**

指令参数说明表：

参数名称	数据类型	参数含义
cam_id: 相机索引	常量	支持多台相机。0表示默认的视觉界面设置连接参数匹配的相机，1-10表示需要配合九点标定中连接参数使用的相机。
msg_string: 消息字符串	String常量或变量	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。

## 3.16 阵列指令

阵列指令列表：

阵列指令	SetMatrix
	GetMatrix

### 3.16.1 SetMatrix

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。

**SetMatrix**HomePos RowPos ColumnPos Row Column

指令参数说明表：

参数名称	数据类型	参数含义
HomePos: 原点	CPOS	指定平行四边形的第一个点，也称为原点。
RowPos: 行方向最后一个点	CPOS	指定平行四边形的行方向上的最后一个点。
ColumnPos: 列方向最后一个点	CPOS	指定平行四边形的列方向上的最后一个点。
Row: 行数	Int常量或Int变量	指生成矩阵的行数，范围为[1,32767]。
Column: 列数	Int常量或Int变量	指生成矩阵的列数，范围为[1,32767]。

### 3.16.2 GetMatrix

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。

**GetMatrix**Row Column TarPos

指令参数说明表：

参数名称	数据类型	参数含义
Row: 行数	Int常量或Int变量	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。
Column: 列数	Int常量或Int变量	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。
TarPos: 目标点	CPOS	用于保存取到的点值。

### 3.17 中断指令

中断指令列表:

中断指令	IConnect
	IDelete
	ITimer
	ISignalDI
	ISignalSimDI
	ExitCycle
	MovTolPos
	ISleep
	IWatch

#### 3.17.1 IConnect

该指令用于中断绑定, 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。

**IConnect** Interrupt Interrupt function

指令参数说明表:

参数名称	数据类型	参数含义
Interrupt: 中断标识	INTERRUPT	指中断索引号。
Interrupt function: 中断处理函数	function	指该中断索引对应的信号满足后需调用的处理程序。

#### 3.17.2 IDelete

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。

**IDelete**Interrupt

指令参数说明表：

参数名称	数据类型	参数含义
Interrupt: 中断标识	INTERRUPT	指中断索引号。

### 3.17.3 ISignalDI

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。

**ISignalDI**Interrupt Port Trigger edge Time

指令参数说明表：

参数名称	数据类型	参数含义
Interrupt: 中断标识	INTERRUPT	指中断索引号。
Port: 端口号	Int	需要监控的DI端口号。
Trigger edge: 触发边沿类型	String	需要监控的DI边沿信号类型，RiseEdge: 上升沿信号，DownEdge: 下降沿信号。
Time: 去重时间	Int	当触发一次中断后，在指定时间内不再响应中断触发信号。

### 3.17.4 ISignalSimDI

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。

**ISignalSimDI**Interrupt Port Trigger edge time

指令参数说明表：

参数名称	数据类型	参数含义
Interrupt: 中断标识	INTERRUPT	指中断索引号。
Port: 端口号	Int	需要监控的SimDI端口号。
Trigger edge: 触发边沿类型	String	需要监控的SimDI边沿信号类型，RiseEdge: 上升沿信号，DownEdge: 下降沿信号。
Time: 去重时间	Int	当触发一次中断后，在指定时间内不再响应中断触发信号。

### 3.17.5 ExitCycle

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。

### 3.17.6 MovToIPos

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。

**MovToIPos** move type Speed

指令参数说明表：

参数名称	数据类型	参数含义
move type: 运动方式	String	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。
Speed: 目标速度	Speed	运动到中断点的速度。

### 3.17.7 ISleep

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。

**ISleep**Interrupt

指令参数说明表：

参数名称	数据类型	参数含义
Interrupt: 中断标识	INTERRUPT	指中断索引号。

### 3.17.8 IWatch

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。

**IWatch**Interrupt

指令参数说明表：

参数名称	数据类型	参数含义
Interrupt: 中断标识	INTERRUPT	指中断索引号。

### 中断指令使用样例说明

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)      # 执行MovJ
...            # 中间执行其他指令
MovToIPos("MOVJ") # 运动到主程序被中断时刻位置，然后继续运行主程序
end

function prog2()
MovJ(G.p4)      # 执行MovJ
...            # 中间执行其他指令
ExitCycle()     # 跳转到主程序的开头，然后系统转为停止状态
end

function main()
IDelete(G.INTERRUPT1)      # 解除中断INTERRUPT1
IDelete(G.INTERRUPT2)     # 解除中断INTERRUPT2
IConnect(G.INTERRUPT1, prog1) # 将INTERRUPT1与prog1连接
IConnect(G.INTERRUPT2, prog2) # 将INTERRUPT2与prog2连接
ISignalDI(G.INTERRUPT1,10,"RiseEdge",5000) # 设置DI 10上升沿触发，去重时间5s
ISignalDI(G.INTERRUPT2,15,"DownEdge",5000) # 设置DI 15下降沿触发，去重时间5s
label :
MovJ(G.p0)
MovJ(G.p1)
MovJ(G.p2)
GOTO label
end
```

## 3.18 交互式指令

交互式指令列表：

交互式指令	DialogPanelInt
	DialogPanelFloat
	DialogPanelString
	DialogPanelList
	DialogPanelMessage

### 3.18.1 DialogPanelInt

该指令用于在程序运行过程中通过弹窗输入int型数据动态地修改变量值。

**DialogPanelInt** Title Context Result

指令参数说明表：

参数名称	数据类型	参数含义
Title: 标题 (Title)	String	弹窗的标题。
Context: 文本 (Context)	String	弹窗的内容。
Result: 返回值 (Result)	ReturnVal-ue	用于接收弹窗中输入的int数据的返回值类型变量。

示例1：

```
DialogPanelInt("title","context" ,ReturnValue0)
将弹窗标题设为title，标题设为context，执行时输入的int数据返回给返回值类型变量ReturnValue0
```

### 3.18.2 DialogPanelFloat

该指令用于在程序运行过程中通过弹窗输入float型数据动态地修改变量值。

**DialogPanelFloat** Title Context Result

指令参数说明表：

参数名称	数据类型	参数含义
Title: 标题 (Title)	String	弹窗的标题。
Context: 文本 (Context)	String	弹窗的内容。
Result: 返回值 (Result)	ReturnVal-ue	用于接收弹窗中输入的float数据的返回值类型变量。

示例1：

```
DialogPanelFloat("title","context" ,ReturnValue0)
将弹窗标题设为title，标题设为context，执行时输入的float数据返回给返回值类型变量ReturnValue0
```

### 3.18.3 DialogPanelString

该指令用于在程序运行过程中通过弹窗输入String型数据动态地修改变量值。

**DialogPanelString** Title Context Result

指令参数说明表：

参数名称	数据类型	参数含义
Title: 标题 (Title)	String	弹窗的标题。
Context: 文本 (Context)	String	弹窗的内容。
Result: 返回值 (Result)	ReturnVal-ue	用于接收弹窗中输入的String数据的返回值类型变量。

示例1:

```
DialogPanelString("title","context",ReturnValue0)
将弹窗标题设为title，标题设为context，执行时输入的String数据返回给返回值类型变量ReturnValue0
```

### 3.18.4 DialogPanelList

该指令用于在程序运行过程中通过弹窗按钮动态地修改变量值。

**DialogPanelList** Title Context [Opt1] [Opt2] [Opt3] [Opt4] [Opt5] [Opt6] [Opt7] [Opt8] [Opt9] [Opt10] Result

指令参数说明表:

参数名称	数据类型	参数含义
Title: 标题 (Title)	String	弹窗的标题。
Context: 文本 (Context)	String	弹窗的内容。
Opt1: 自定义选项1 (Opt1)	String	弹窗中按钮1的名称。
Opt2: 自定义选项2 (Opt2)	String	弹窗中按钮2的名称。
Opt3: 自定义选项3 (Opt3)	String	弹窗中按钮3的名称。
Opt4: 自定义选项4 (Opt4)	String	弹窗中按钮4的名称。
Opt5: 自定义选项5 (Opt5)	String	弹窗中按钮5的名称。
Opt6: 自定义选项6 (Opt6)	String	弹窗中按钮6的名称。
Opt7: 自定义选项7 (Opt7)	String	弹窗中按钮7的名称。
Opt8: 自定义选项8 (Opt8)	String	弹窗中按钮8的名称。
Opt9: 自定义选项9 (Opt9)	String	弹窗中按钮9的名称。
Opt10: 自定义选项10 (Opt10)	String	弹窗中按钮10的名称。
Result: 返回值 (Result)	ReturnVal-ue	用于接收消息提示弹窗中被按下的按钮编号的返回值类型变量。

#### 注意

该指令Opt1-Opt10中至少有一个按钮不为空（为空的按钮不会在弹窗中显示），否则执行时会有报警提示。

示例1：

```
DialogPanelList("title","context","btn1","btn2",ReturnValue0)
将弹窗标题设为title，标题设为context，Opt 1和Opt 2分别命名为btn1和btn2，其余按钮默认，执行时选择的按钮编号返回给返回值类型变量ReturnValue0
```

### 3.18.5 DialogPanelMessage

该指令用于在程序运行过程中通过弹窗选择是或否按钮来动态地修改变量值。

**DialogPanelMessage** Title Context Result

指令参数说明表：

参数名称	数据类型	参数含义
Title: 标题 (Title)	String	弹窗的标题。
Context: 文本 (Context)	String	弹窗的内容。
Result: 返回值 (Result)	ReturnVal-ue	用于接收弹窗中所选是(1)或否(0)按钮的返回值类型变量。

示例1：

```
DialogPanelMessage("title","context",ReturnValue0)
将弹窗标题设为title，标题设为context，执行时所选是(1)或否(0)按钮对应的int值返回给返回值类型变量ReturnValue0
```

## 3.19 多机协同指令

多机协同指令列表：

多机协同指令	Gmove
--------	-------

### 3.19.1 Gmove

该指令用于多机协同运动。

**GMove** [SyncMotion] [AuxPos] [targetPos] [Coord] [Gweave] [Speed] [BlendType] [Zone] [AddDo]

指令参数说明表：

参数名称	数据类型	参数含义
SyncMotion: 多机协同运动方式	String	配置每台机器人的运动方法。
AuxPos: 多机中间位置	Gpos	当运动方式包含圆弧运动时，需要写入中间位置点。

参数名称	数据类型	参数含义
targetPos: 多机目标位置	Gpos	坐标系下的多机运动目标位置。
Coord: 多机协同参数	Coord	多机协同所使用的协同对。
Gweave: 多机弧摆参数	Gweave	多机运动方式包含弧摆运动时, 需要写入弧摆参数。
Speed: 目标速度	Speed	<p>执行时的速度, 包括机器人末端的平移速度, 旋转速度以及外部轴的移动速度等。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为V4000。</p> <hr/>
BlendType: 过渡类型	String	<p>机器人逼近终点时的过渡方式。</p> <ul style="list-style-type: none"> <li>• FINE: 精确到位。</li> <li>• RELATIVE: 相对过渡。</li> <li>• ABSOLUTE: 绝对过渡。</li> <li>• DEFAULT: 缺省类型。</li> </ul> <hr/> <p><b>⚠ 注意</b> 该参数可缺省。</p> <hr/>
Zone: 过渡值	Zone	<p>机器人逼近终点时的过渡值。</p> <hr/> <p><b>⚠ 注意</b> 该参数可缺省, 默认值为C100。</p> <hr/>
AddDo: 运动完成后执行的IO操作	String	<p>机器人执行完该指令后, 可触发的IO操作。</p> <ul style="list-style-type: none"> <li>• NULL: 无任何操作。</li> <li>• IO指令: 执行IO操作, 目前支持设置DO、AO、SimGO、SimDO、SimAO。</li> </ul>

示例1:

```
Gmove("L+L+L",Gpos0,coord=S.Coord1,v=S.V4000,b="ABSOLUTE",c=S.C200)
```

多机配置了3台机器人, 都使用直线运动的方式, 运动到Gpos0位置, 协同对采用Coord1, 速度值V4000, 采用绝对过渡的方式, 过渡值C200。

## 3.20 任务指令

任务指令列表:

任务指令	run
	kill

### 3.20.1 run

该指令用于启动任务。

**run** func name type

指令参数说明表：

参数名称	数据类型	参数含义
func: 函数地址 (function address)	Any	函数的地址。
name: 任务名称 (task name)	String	用于启动的任务名称。
type: 任务类型 (task type)	String	用于启动的任务类型。

### 3.20.2 kill

该指令用于终止任务。

**kill** func

指令参数说明表：

参数名称	数据类型	参数含义
func: 函数地址 (function address)	Any	函数的地址。

## 3.21 异常指令

异常指令列表：

异常指令	throw
------	-------

### 3.21.1 throw

该指令用于抛出异常。

**throw** e

指令参数说明表：

参数名称	数据类型	参数含义
e: 错误异常 (ExceptionInfo)	Exception-Info	错误异常。

## 3.22 负载在线辨识指令

负载在线辨识列表：

负载在线辨识指令	StartMdlLog
	StopMdlLog
	CalcDynLog

### 3.22.1 StartMdlLog

负载在线辨识数据采集开始指令，用于采集机器人负载辨识所需的机器人运动数据。

**StartMdlLog** data\_type

指令参数说明表：

参数名称	数据类型	参数含义
data_type: 数据保存类型 (data type)	enum (Robot/WithLoad/ValidData)	数据保存类型。 <ul style="list-style-type: none"> <li>Robot: 采集机器人本体即未安装负载时的运动数据。</li> <li>WithLoad: 采集机器人法兰端安装有负载时的运动数据。</li> <li>ValidData: 用于需要验证模型准确度，采集机器人装有负载的运动数据。</li> </ul>

#### 注意

- “WithLoad”类型数据为必须采集，“Robot”、“ValidData”类型数据可以选择性采集。
- 使用“Robot”类型数据进行辨识，有助于改善负载辨识的准确度，但该项会延长辨识计算用时。
- 使用“ValidData”类型数据，可以在辨识计算的同时验证辨识的模型准确度，但该项会延长辨识计算用时。
- 负载辨识数据采集，选择保存数据类型“Robot”、“WithLoad”时，辨识轨迹所用运动指令需为MovJ，而类型“ValidData”验证轨迹运动指令可使用MovJ、MovL、MovC等组合指令。

### 3.22.2 StopMdlLog

负载在线辨识数据采集结束指令，用于结束机器人负载辨识运动数据的采集，并自动保存数据。

#### ⚠ 注意

该指令必须与StartMdlLog搭配使用。

示例1：

```
StartMdlLog(WithLoad)
MovJ(P0)
MovJ(P1)
MovJ(P2)
StopMdlLog()
```

### 3.22.3 CalcDynMdl

该指令用于对之前采集的机器人负载信息进行辨识计算，并将计算后的结果存储到对应的变量中。

**CalcDynMdl** mdl\_type Tool Payload data\_step valid\_enable

指令参数说明表：

参数名称	数据类型	参数含义
mdl_type: 辨识的模型类型 (Identified model type)	enum (Type-Tool/Type-Payload)	辨识的模型类型。 <ul style="list-style-type: none"> <li>TypeTool: 用于辨识工具参数。</li> <li>TypePayload: 用户辨识负载参数。</li> </ul>
Tool: 工具变量 (TOOL)	TOOL变量	用于保存辨识结果。当辨识的模型类型选择Type-Tool时，本参数才有效。
Payload: 工件负载变量 (PAYLOAD)	PAYLOAD变量	用于保存辨识结果。当辨识的模型类型选择TypePayload时，本参数才有效。
data_step: 数据间隔长度 (Identify the percentage of data extraction length)	int	从原始数据中抽取有效辨识数据的间隔参数，该值越大则辨识所用数据量越少，负载辨识结果准确度也就越低，取值范围0~100。
valid_enable: 是否模型验证 (Model validation)	enum (Enable/Disable)	设置负载辨识时进行模型验证的使能。 <ul style="list-style-type: none"> <li>Enable: 开启模型验证功能。</li> <li>Disable: 关闭模型验证功能。</li> </ul>

示例1：

```
StartMdlLog(WithLoad)
MovJ(P0)
```

```
MovJ(P1)
MovJ(P2)
StopMdlLog()
/*负载辨识结束，结果将存于用户指定的负载/工具变量中*/
CalcDynMdl{mdlType="TypePayload",Payload=t_g.idenPayload,dataStepPer=1,validEnable="Disable"}
```

### 3.23 飞拍指令

飞拍指令列表：

飞拍指令	LatchEnable
	GetPSOStatus
	RecordLatchPos
	SetLatchPos
	SetLatchPosAttr

#### 3.23.1 LatchEnable

该指令用于设置机器人指定伺服轴PSO控制字的使能与禁能；当设置某一轴使能时，伺服才会检测PSO比较点。

**LatchEnable** AxisID EnableSts

指令参数说明表：

参数名称	数据类型	参数含义
AxisID: 轴号 (AxisID)	Int	需要操作的机器人轴号。
EnableSts: 使能状态 (Enable Status)	enum (ON/OFF)	需要设置的PSO控制字的使能状态。

示例1：

```
LatchEnable(1, "ON")
将机器人1轴的PSO控制字设置为使能状态
```

#### 3.23.2 GetPSOStatus

该指令用于获取机器人指定伺服轴PSO状态字的值。

**GetPSOStatus** AxisID PSOValue

指令参数说明表：

参数名称	数据类型	参数含义
AxisID: 轴号 (AxisID)	Int	需要操作的机器人轴号。

参数名称	数据类型	参数含义
PSOValue: PSO状态值反馈 (PSO Status Feedback)	ReturnValue	PSO状态字的值。

示例1:

```
GetPSOStatus(1, ReturnValue0)
获取机器人1轴的PSO状态字的值
```

### 3.23.3 RecordLatchPos

该指令用于设置下一段运动在过渡段上的记录点序号，调试时配合SetLatchPos指令一起使用，可以设置运动过渡段上的点作为触发点。

**RecordLatchPos** RecordLatchID

指令参数说明表:

参数名称	数据类型	参数含义
RecordLatchID: 记录点编号 (ID of the record point)	String	需要设置的下一段运动在过渡段上的记录点序号。

示例1:

```
RecordLatchPos("LATCH_1")
设置下一段运动在过渡段上的记录点序号为LATCH_1
```

### 3.23.4 SetLatchPos

该指令用于设置机器人指定轴的指定PSO比较点，成功时返回机器人当前关节坐标。当机器人停在某一位置时执行该条指令，那么伺服会记录下当前指定轴的位置作为一个比较点，后续机器人运动到该位置时，伺服则会触发拍照信号。

**SetLatchPos** AxisID ComparePointID Ret LatchPos RecordLatchID

指令参数说明表:

参数名称	数据类型	参数含义
AxisID: 轴号 (AxisID)	Int	需要操作的机器人轴号。
ComparePointID: 比较点编号 (ID of the comparison point)	Int	比较点的编号。
Ret: 操作返回值 (Return Value)	ReturnValue	操作返回值。
LatchPos: 当前关节位置反馈 (APOS)	APOS	返回的机器人当前关节坐标。

参数名称	数据类型	参数含义
RecordLatchID: 过渡段记录点的索引 (The ID of the transition recording point)	enum (LATCH_1、LATCH_2、LATCH_3.....LATCH_20)	过渡段记录点索引, 对应RecordLatchPos指令所设置的ID号。

示例1:

```
SetLatchPos(1,1,ReturnValue0,p0,RecordLatchID="LATCH 1")
设置机器人1轴的第1个PSO比较点, 使用RecordLatchPos所设置的过渡段上的比较点作为触发点, LATCH_1对应
RecordLatchPos(1), 伺服记录当前机器人1轴的位置, 成功时返回机器人当前关节坐标
```

## 3.24 ERI指令

ERI指令列表:

ERI指令	SetERIProgID
	EnableERIControl
	SetERIParam
	SetCorrParam
	EnableERICorrect

### 3.24.1 SetERIProgID

该指令用于执行ERI对应的程序编号。

**SetERIProgID** ProgId

指令参数说明表:

参数名称	数据类型	参数含义
ProgId: 程序ID (Program Id)	Int	程序编号。

示例1:

```
SetERIProgID(1)
```

### 3.24.2 EnableERIControl

该指令用于开启对ERI的控制。

**EnableERIControlP**

示例1：

```
EnableERIControl()
```

### 3.24.3 SetERIParam

该指令用于设置ERI参数。

**SetERIParam** Period fType

指令参数说明表：

参数名称	数据类型	参数含义
Period: 采样周期 (Sampling Period)	Int	采样周期。
fType: 滤波器类型 (Filter Type)	enum ("DisEnable"/"LowPass"/"Ellipse")	滤波器类型。

示例1：

```
SetERIParam(4,"DisEnable")
```

### 3.24.4 SetCorrParam

该指令用于设置纠偏参数。

**SetCorrParam** X Y Z A B C CoordType

指令参数说明表：

参数名称	数据类型	参数含义
X: X报警阈值 (X Corr Threshold)	Float	X报警阈值。
Y: Y报警阈值 (Y Corr Threshold)	Float	Y报警阈值。
Z: Z报警阈值 (Z Corr Threshold)	Float	Z报警阈值。
A: A报警阈值 (A Corr Threshold)	Float	A报警阈值。
B: B报警阈值 (B Corr Threshold)	Float	B报警阈值。
C: C报警阈值 (C Corr Threshold)	Float	C报警阈值。
CoordType: 纠偏坐标系类型 (Coord Type)	enum ("World"/"Tool"/"TTS")	纠偏坐标系类型。

示例1：

```
SetCorrParam(6.0,6.0,6.0,6.0,6.0,6.0,"World")
```

### 3.24.5 EnableERICorrect

该指令用于开启ERI的纠偏控制。

**EnableERICorrect EnableSts**

指令参数说明表：

参数名称	数据类型	参数含义
EnableSts: 使能状态 (Enable Status)	enum ("ON"/"OFF")	使能状态。

示例1：

```
EnableERICorrect("ON")
```

## 3.25 软浮动指令

软浮动指令列表：

软浮动指令	SoftFloatStart
	SoftFloatStop

### 3.25.1 SoftFloatStart

该指令可以在不使用外部传感器或者增加机械结构的情况下使机器人由刚性状态变为柔性状态。指令开启后，机器人末端可以顺应外力沿指定方向直线运动。

**SoftFloatStart**Type ID [Tool] [Frame]

#### 注意

带[]参数为可选参数

指令参数说明表：

参数名称	数据类型	参数含义
Type: 软浮动类型 (SoftType)	enum ("AXIS"/ "CART"/"PUSH")	软浮动类型。
ID: 软浮动ID (SoftTypeID)	Int	软浮动ID。

参数名称	数据类型	参数含义
ref_coord: 参考坐标系 (refer coord)	enum ("Tool"/"Frame")	软浮动参考坐标系。 <ul style="list-style-type: none"> <li>• Tool: 工具坐标系。</li> <li>• Frame: 用户坐标系。</li> </ul>
ref_coord_cfg: 参考坐标系参数 (refer coord cfg)	Union{Frame, Tool}	参考坐标系参数。

示例1:

```
SoftFloatStart("CART",1,ref_coord="TOOL",frame_or_tool=S.TOOL[2])
```

### 3.25.2 SoftFloatStop

该指令用于关闭软浮动功能。

示例1:

```
SoftFloatStop ()
```

## 3.26 样条曲线指令

样条曲线指令列表:

样条曲线指令	GetPosFromFile
	OnSplineIndex
	SetSplineParam
	MovS

### 3.26.1 GetPosFromFile

该指令用于从指定文件中获取点位数据。

**GetPosFromFile** FileName PosName StartIndex Number

指令参数说明表:

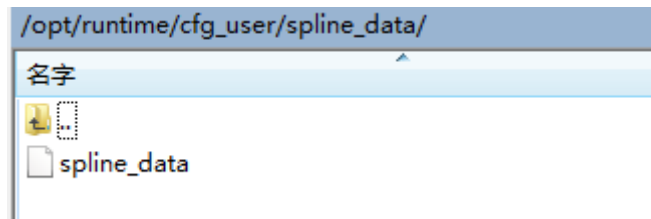
参数名称	数据类型	参数含义
FileName: 文件名 (file name)	String	文件名。
PosName: 点位变量名称 (point variable name)	enum ("APOS" / "CPOS")	点位变量名称。

参数名称	数据类型	参数含义
StartIndex: 点位起始序号 (point start index)	Int	点位起始序号。
Number: 点位数量 (point number)	Int	点位数量。

示例1:

```
GetPosFromFile ("spline_data","CPOs",1,180)
```

点位文件需要存放在指定路径/opt/runtime/cfg\_user/spline\_data，如下图所示：



文件内容格式遵循如下规范：

1. Cpos 模式读取时，文件内容如下：

```
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),0,0,1195.397000,24.458000,612.596000,89.840000,0.134000,-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),0,0,1187.797000,26.291000,613.048000,89.840000,0.134000,-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),0,0,1177.025000,25.614000,613.694000,89.840000,0.134000,-176.567000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1174.071000,25.122000,613.867000,89.836000,0.123000,-176.704000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Cpos(Poscfg(0,0,0,0,0,0),0,0,1171.119000,24.613000,614.037000,89.834000,0.117000,-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),0,0,1168.070000,24.089000,614.207000,89.832000,0.111000,-176.841000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
```

文件内容字段依次表示为：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，如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)
```

2. Apos模式读取时，文件内容如下：

```
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(0,0,1177.025000,25.614000,613.694000,89.840000,0.134000,-176.567000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Apos(0,0,1174.071000,25.122000,613.867000,89.836000,0.123000,-176.704000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
Apos(0,0,1171.119000,24.613000,614.037000,89.834000,0.117000,-176.773000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000)
```

文件内容字段依次表示：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

例：指令样式和执行效果如下所示：

```
Wait(10)
GetPosFromFile("spline_data","CPOS",1,180)
end
```

选择类型: 系统变量 / CPOS										
name	alias	tool	frame	x	y	z	a	b	c	a7
CPOS[1]		0	0	1293.66	-13.041	810.899	89.811	-3.483	177.509	0.0
CPOS[2]		0	0	1201.404	24.676	712.6	89.84	0.134	-176.567	0.0
CPOS[3]		0	0	1195.397	24.458	612.596	89.84	0.134	-176.567	0.0
CPOS[4]		0	0	1187.797	26.291	613.048	89.84	0.134	-176.567	0.0
CPOS[5]		0	0	1179.982	26.09	613.518	89.84	0.134	-176.567	0.0
CPOS[6]		0	0	1177.025	25.614	613.694	89.84	0.134	-176.567	0.0

### ⚠ 注意

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。

## 3.26.2 OnSplineIndex

该指令用于触发样条曲线点位。

**OnSplineIndex** PointIndex delay\_time Trig\_opt

指令参数说明表：

参数名称	数据类型	参数含义
PointIndex: 触发点序号 (Trigger point index)	Int	触发点序号。
delay_time: 触发时间 (delay time)	Int	触发时间。
Trig_opt: 触发操作 (Trigger operation)	enum ("时钟指令" / "IO指令" / "控制指令")	触发操作。

示例1：

```
SetSplineParam("SMOOTH")
```

## 3.26.3 SetSplineParam

该指令用于设置样条曲线运动参数。对指令参数进行说明：

**SetSplineParam** SplineMode

指令参数说明表：

参数名称	数据类型	参数含义
SplineMode: 样条曲线运动模式 (spline motion mode)	enum ("SMOOTH" / "ACCURATE")	样条曲线运动模式。

示例1：

```
SetSplineParam("SMOOTH")
```

### 3.26.4 MovS

该指令用于执行样条曲线运动。

**MovS** PosName StartIndex EndIndex v b c payload singular\_pass o

指令参数说明表：

参数名称	数据类型	参数含义
PosName: 点位变量名称 (point variable name)	enum ("APOS" / "CPOS")	点位变量名称。
StartIndex: 点位起始序号 (point start index)	Int	点位起始序号。
EndIndex: 点位终止序号 (point end index)	Int	点位终止序号。
v: 目标速度 (speed)	Speed	目标速度。
b: 过渡类型 (BlendType)	enum ("DEFAULT" / "RELATIVE" / "ABSOLUTE" / "FINE")	过渡类型。
c: 过渡值 (BlendValue)	Zone	过渡值。
payload: 工件负载 (PayLoad)	Payload	工件负载。
o: 速率类型 (OverrideType)	enum ("GOVRON" / "GOVROFF")	速率类型。

示例1：

```
MovS("CPOS",1,1,v=S.V10,b="RELATIVE",c=S.C20,payload=S.PAYLOAD[1],o="GOVRON")
```

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)