

ER 系列 RCS2 指令系统手册

(RCS2V1.12)

修订记录

序号	日期	版本	描述
1	2019.04.17	V1.0	初次发布
2	2019.04.22	V1.1	修正版本
3	2019.07.22	V1.2	1. 新增 SetMotionMode 与 GetCurOverRide 指令说明。 2. 修改码垛指令的参数说明。
4	2019.12.31	V1.3	1. 移除 EMovL, EMovC 指令的说明。 2. 除了 MovJRel, MovLRel 两条指令添加了 PAYLOAD 属性外, 其他所有的运动指令中均添加了 TOOL, USERCOORD, PAYLOAD 属性变量, 对这些添加的属性变量进行说明。 3. 修改 SetPayload 指令说明。 4. 新增 SetRtInfo, SetRtWarning 指令说明。 5. 新增 PAYLOAD 变量说明。 6. 其它修改。
5	2020.7.13	V1.4	1. 新增 SetExternalTCP 指令。 2. 修改了 MovL, MovC 指令中 Coord 的属性说明。 3. 在 WAVE 变量新增 Type 属性。 4. 新增 EXTTCPIP 变量类型。 5. 其它修改。
6	2021.2.8	V1.5	1. 新增 MovLSync、MovJSyncQuit、MovLSyncQuit、WaitWObj、SetTargetPos 指令, 并新建一个传送带跟随指令列表来存储。 2. 新增 SYNCOORD 变量。 3. 在 WaitCondition、WaitDI、WaitDI8421、WaitAI、WaitSimDI、WaitSimDI8421、WaitSimAI 这些时间等待指令中新增了 Goto 属性变量。 4. 在区域变量中新增 initActivate, enInPut, enInHigh 三个属性。 5. 新增 ELSIF 指令
7	2021.5.24	V1.6	1. 在 MovLSync 指令中新增了两个 Goto 属性变量以及太升高度。 2. 新增 LsScale、LsThresh 变量。 3. 在设置指令中新增 AutoGainEnable、AutoGainDisable 指令。
8	2021.7.26	V1.7	1. MovL、MovC、MovLRel、MovLW、MovCW 指令新增参数: 是否受全局速度倍率影响。 2. 新增 MovLOffset、SetSingularPass、SetColliEnable、SetAxisColliParam、SetMatrix、GetMatrix、SocketSendReal、SocketSendInt、SetPositioner、SetBlendParam 指令。 3. 修改指令名 GetOverRide 为 GetCurOverRide。

序号	日期	版本	描述
9	2021.10.15	V1.8	1. 完善系统关键字(词)。 2. 新增区域数据类型的多边形区域变量。 3. 修改指令 MovL、MovC、MovLRel、MovLW、MovCW 指令中的坐标系参数。 4. 新增 PolyhedronAreaActivate、PolyhedronAreaDeactivate 指令。
10	2022.1.6	V1.9	1. 修改 Wait、WaitCondition、WaitDI、WaitDI8421、WaitAI、WaitSimDI、WaitSimDI8421、WaitSimAI 等指令中时间参数的类型说明，修改 SetOverride 指令中速度倍率参数的类型说明。 2. 新建字符串指令列表，将赋值指令与流程控制指令中的字符串函数移至其中，并新增 findEnd、format、getAt、left、right、reverse、strcmp、trimLeft、trimRight、IToStr、RToStr、StrToI、StrToR、APosToStr、CPosToStr、TranStrToInt、TranStrToReal、TranStrToApos、TranStrToCpos 指令。同时移除赋值指令与流程控制指令中的字符串函数。 3. 在传送带跟随指令列表中新增 WaitConvDis 指令。 4. 在设置指令列表中新增 SetRestorePC、SetAxisVibraBLevel 指令。 5. 将 WEAVE 变量中的 Amp 属性变更为 Amp_L、Amp_R，StopTime 属性变更为 StopTime_L、StopTime_C、StopTime_R。 6. 在运动指令列表中增加 MovCircle、MovCircleW 指令。 7. 在 PLC 数据类型中新增 PLCBOOL、PLCINT、PLCDINT 变量。 8. 新增 ModbusTcp 指令列表。
11	2022.4.13	V1.10	1. 在位置运算指令列表中新增 ToolOffset、UserOffset 指令。
12	2022.7.18	V1.11	1. 在 MovArch 指令中新增 H2 参数，并修改了部分指令参数说明。 2. 传送带跟随指令中新增 SimConveyorOn、SimConveyorOff、ReceiveWObj、ResetWObjBuf 指令。
13	2022.10.28	V1.12	1. 修改了 SocketReadReal、SocketReadInt 指令中接收数据的格式说明，以及 SocketSendStr 指令中的参数说明。 2. 在 Socket 指令集中新增了 SocketResetBuf 指令。 3. 在 AREA 变量中新增了 isFlangeEnd 参数。 4. 修正了 RUN 指令的说明。 5. 修正了 WaitDI8421、WaitSimDI8421 指令参数说明顺序。

目录

修订记录.....	i
目录.....	i
前言.....	1-1
概述	1-1
第 1 章 系统介绍.....	1-2
1.1 系统简介	1-2
1.2 系统域数据文件.....	1-2
1.2.1 系统域 TOOL 变量	1-2
1.2.2 系统域 USERCOORD 变量	1-3
1.2.3 系统域 SPEED 预定义常量	1-3
1.2.4 系统域 ZONE 预定义常量	1-3
1.2.5 系统域 PAYLOAD 变量	1-4
1.3 全局域数据文件.....	1-4
1.4 工程文件	1-4
1.4.1 工程域数据文件	1-4
1.4.2 程序指令文件	1-4
1.4.3 程序域数据文件	1-5
1.5 系统关键字（词）	1-5
第 2 章 变量数据类型.....	2-1
2.1 基本数据类型.....	2-1
2.2 位置数据类型.....	2-2
2.3 系统数据类型.....	2-8
2.4 IO 数据类型	2-14
2.5 摆动数据类型.....	2-16
2.6 时钟数据类型.....	2-17
2.7 区域数据类型.....	2-17
2.8 PLC 数据类型	2-19
2.9 码垛数据类型.....	2-20
2.10 SOCKET 数据类型	2-21
第 3 章 指令.....	3-22
3.1 运动指令	3-22
3.2 控制指令	3-45
3.3 等待指令	3-48
3.4 IO 指令	3-49
3.5 设置指令	3-62
3.6 位置运算指令.....	3-71
3.7 位运算指令	3-75
3.8 时钟指令	3-77
3.9 区域指令	3-79

3.10 视觉指令	3-81
3.11 码垛指令.....	3-83
3.12 Socket 指令.....	3-88
3.13 软浮动指令.....	3-93
3.14 传送带跟随指令.....	3-94
3.15 阵列指令	3-100
3.16 数学运算函数.....	3-102
3.17 字符串指令.....	3-105
3.18 ModbusTcp 指令	3-110

前言

概述

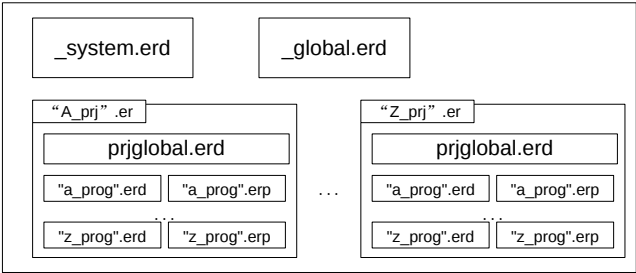
本文档适用于控制系统 RCS2 V1.29 及以上版本，描述了 ER 系列机器人 RCS2 指令系统中结构定义、变量类型、参数含义，有助于使用者深刻了解指令系统。

第 1 章 系统介绍

1.1 系统简介

指令系统文件结构分为工程、程序和数据三部分，其中，数据又分为系统域、全局域、工程域和程序域四个作用域。程序代码保存在文件中，后缀名：.er、.erp 和.erd。一般都在 Project 文件夹中。

整个 Project 文件结构：



- 系统域数据文件 _system.erd
- 全局域数据文件 _global.erd
- 工程文件夹 “A_prj”.er
 - 工程域数据文件 prjglobal.erd
 - 程序指令文件“a_prog”.erp
 - 程序域数据文件“a_prog”.erd

注意：系统和全局、工程域数据文件名称为系统指定，不可修改。其他文件名字可以自定义。

用户可以通过示教编程器或其他接口编辑指令文件。

1.2 系统域数据文件

Project 目录下_system.erd 为系统数据文件。变量对应系统域，为预定义，用户不可编辑，但可被所有工程中引用。该文件主要包括预定义的四种类型变量（变量类型详见下一章节介绍）：

- TOOL
- USERCOORD
- SPEED
- ZONE
- PAYLOAD

1.2.1 系统域 TOOL 变量

系统预定义的 TOOL 类型变量，具体如下表所示：

名称	x	y	z	a	b	c	M	Mx	My	Mz	Ixx	Iyy	Izz	Ixy	Iyz	Ixz
nullTool	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1.2.2 系统域 USERCOORD 变量

系统预定义的 USERCOORD 类型变量，具体如下表所示：

名称	x	y	z	a	b	c
World	0	0	0	0	0	0

1.2.3 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	360	360	180
V10	2	10	360	360	180
V30	3	30	360	360	180
V50	5	50	360	360	180
V60	6	60	360	360	180
V80	8	80	360	360	180
V100	10	100	360	360	180
V200	20	200	360	360	180
V300	30	300	360	360	180
V500	50	500	360	360	180
V800	80	800	360	360	180
V1000	100	1000	360	360	180
V1500	100	1500	360	360	180
V2000	100	2000	360	360	180
V3000	100	3000	360	360	180
V4000	100	4000	360	360	180

1.2.4 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	0	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

1.2.5 系统域 PAYLOAD 变量

系统预定义的 PAYLOAD 类型变量，具体如下表所示：

名称	M(kg)	Mx	My	Mz	Ixx	Iyy	Izz	Ixy	Iyz	Ixz
FullPayload	x	0	0	0	0	0	0	0	0	0

1.3 全局域数据文件

Project 目录下_global.erd 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

每个工程 project.er 目录下 prjglobal.erd 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 程序指令文件

每个工程目录下至少包含一个或多个程序指令文件，其后缀名均为.erp。该文件存放用户编辑的指令，如后续章节中介绍的 MovJ 等。

1.4.3 程序域数据文件

每个程序均有其对应名称的数据文件，其后缀名均为.erd。该文件中存放的变量只能被本程序指令引用。

1.5 系统关键字（词）



说明

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

基础函数：

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

变量类型：

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

指令名称：

...=...、/*...*/、APosToCPos、AreaActivate、AreaDeactivate、PolyhedronAreaActivate、PolyhedronAreaDeactivate、AutoGainDisable、AutoGainEnable、BitAnd、BitLSh、BitNeg、BitOr、BitRSh、BitXOr、CalcCoord、CalcTool、CALL、CLKRead、CLKReset、CLKStart、CLKStop、CompareAI、CompareSimAI、CPosToAPos、CPosToCPos、ELSE、ELSIF、ENDIF、ENDWHILE、GetCamPos、GetCurAPos、GetCurCPos、GetDI8421、GetMatrix、GetSimAIToVar、GetSimDI8421、GetSimDIToVar、GetTrackId、GOTO、Hand、IF、LABEL、MovArch、MovC、MovCW、MovH、MovJ、MovJRel、MovJSearch、MovJSyncQuit、MovL、MovLOffset、MovLRel、MovLSearch、MovLSync、MovLSyncQuit、MovLW、OnDistance、OnParameter、PalletFromGet、PalletFromPut、PalletToGet、PalletToPut、PalletReset、PulseOut、PulseSimOut、RefRobotAxis、RETURN、RUN、SendMessage、SetAO、SetAxisColliParam、SetBlendParam、SetCartDyn、SetColliEnable、SetCoord、SetCurOverRide、SetDIEdge、SetDO、SetDO8421、SetExternalTCP、SetJointDyn、SetMatrix、SetMotionMode、SetPayload、SetPositioner、SetRtInfo、SetRtToErr、SetRtWarning、SetSimAO、SetSimAOByVar、SetSimDIEdge、SetSimDO、SetSimDO8421、SetSimDOByVar、SetSingularPass、SetSyncoord、SetTargetPos、SetTool、SocketClose、SocketCreate、SocketReadInt、SocketReadReal、SocketReadStr、SocketSendInt、SocketSendReal、SocketSendStr、SoftFloatStart、SoftFloatStop、Stop、SyncToUserC、TrigCam、Wait、WaitAI、WaitCondition、WaitDI、WaitDI8421、Waitfinish、WaitFinishCAM、WaitSimAI、WaitSimDI、WaitSimDI8421、WaitWOBJ、WHILE

特定名称

POSCFG、PLACEOPTION

第 2 章 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，不可编辑。
- 全局域：IO 数据类型、PLC 数据类型、socket 数据类型、位置数据类型、区域数据类型、基本数据类型、摆动数据类型、时钟数据类型、码垛数据类型、系统数据类型。
- 工程域：IO 数据类型、socket 数据类型、位置数据类型、基本数据类型、摆动数据类型、码垛数据类型、系统数据类型。
- 程序域：IO 数据类型、socket 数据类型、位置数据类型、基本数据类型、摆动数据类型、系统数据类型。

2.1 基本数据类型

BOOL

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

参数说明：

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

INT

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

参数说明：

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

REAL

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

参数说明：

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

STRING

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

参数说明：

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

BoolOneArray

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

参数说明：

- count：数据类型：int
参数含义：数据个数
- value：数据类型：bool
参数含义：变量值

IntOneArray

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

参数说明：

- count：数据类型：int
参数含义：数据个数
- value：数据类型：int
参数含义：变量值

RealOneArray

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

参数说明：

- count：数据类型：int
参数含义：数据个数
- value：数据类型：real
参数含义：变量值

2.2 位置数据类型

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，程序三大变量作用域中创建，删除，修改等操作。

参数说明：

- a1: 数据类型: real
参数含义: 关节 1 号轴的坐标。
 - a2: 数据类型: real
参数含义: 关节 2 号轴的坐标。
 - a3: 数据类型: real
参数含义: 关节 3 号轴的坐标。
 - ... : 参数 a4~a15 的属性跟上述一样, 都是 real 型变量, 依次表示关节 4~关节 15 号轴的坐标值。。
 - a16: 数据类型: real
参数含义: 关节 16 号轴的坐标。
- 备注: 机器人有几个轴就会显示几个轴的参数。如通用六轴机器人显示 a1~a6,scara 机器人显示为 a1~a4.

POSCFG

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

参数说明:

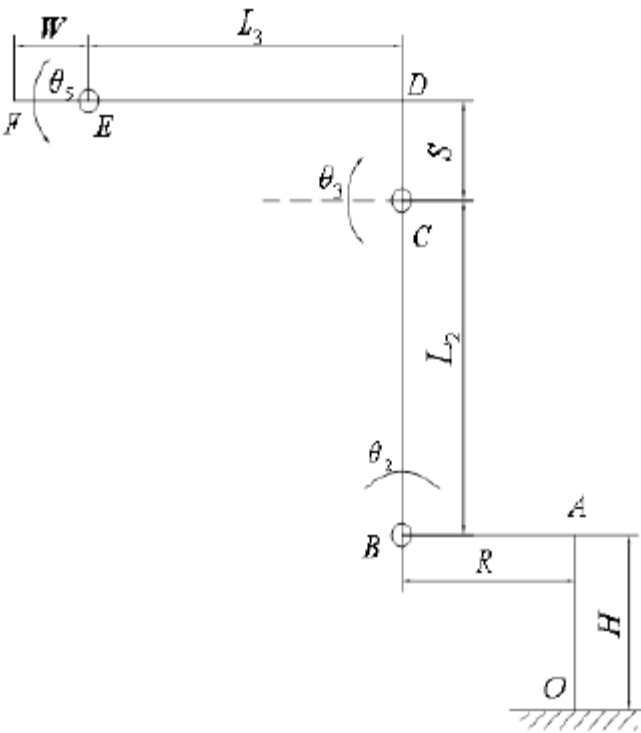
- mode: 数据类型: int
- cf1: 数据类型: int
参数含义: 关节 1 号轴角度所在的象限取值。
- cf2: 数据类型: int
参数含义: 关节 2 号轴角度所在的象限取值。
- cf3: 数据类型: int
参数含义: 关节 3 号轴角度所在的象限取值。
- cf4: 数据类型: int
参数含义: 关节 4 号轴角度所在的象限取值。
- cf5: 数据类型: int
参数含义: 关节 5 号轴角度所在的象限取值。
- cf6: 数据类型: int
参数含义: 关节 6 号轴角度所在的象限取值。

这些象限取值划分如下:

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

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

①、通用六关节运动学 mode 定义



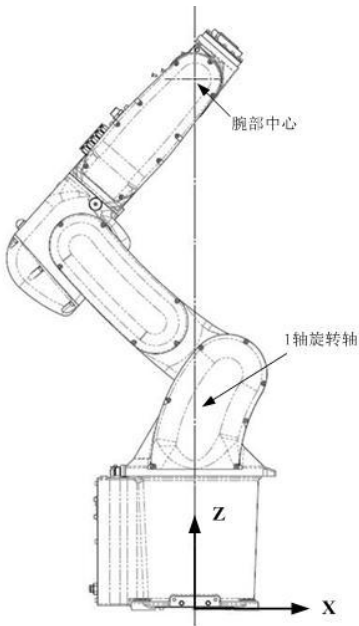
通用六关节的运动学存在八组解，定义 mode 值为 0~7，含义如下表：

Mode	腕部中心相对于一轴轴心的关系(flag1) 0: 在前; 1: 在后 $R + L_3 \bullet \cos(\theta_2 + \theta_3) + L_2 \bullet \sin \theta_2 + S \bullet \sin(\theta_2 + \theta_3)$	Axis3(flag3) $(\theta_3 + 90 - \arctan(S/L3))$ 0: [0,180] 1:(-180,0)	Axis5(flag5) (θ_5) 0: [0,180] 1:(-180,0)
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

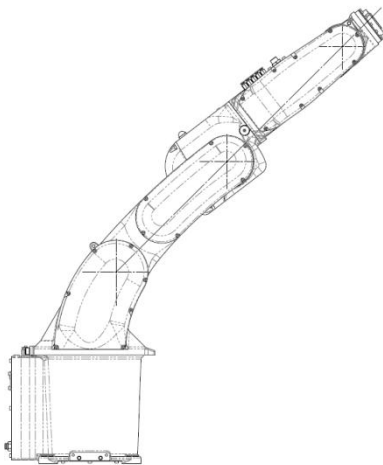
$mode = flag5 + 2*flag3 + 4*flag1。$

注：

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



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



cc、Axis5(flag5)的含义：即五轴的角度。

②、SCARA 运动学 mode 定义

Scara 运动学存在两组解，定义 mode 值为 0 和 1，含义如下表

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

	1: (-180,0)
0	0
1	1

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

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，程序三大变量作用域中创建，删除，修改等操作。

参数说明：

- confdata: 数据类型: POSCFG
参数含义: 空间目标点对应的形态配置信息。
- x: 数据类型: real
参数含义: TCP 点在参考坐标系上 x 方向的坐标。
- y: 数据类型: real
参数含义: TCP 点在参考坐标系上 y 方向的坐标。
- z: 数据类型: real
参数含义: TCP 点在参考坐标系上 z 方向的坐标。
- a: 数据类型: real
参数含义: TCP 点相对于参考坐标系 z 轴旋转的欧拉角。
- b: 数据类型: real
参数含义: TCP 点相对于参考坐标系 y'轴旋转的欧拉角。
- c: 数据类型: real
参数含义: TCP 点相对于参考坐标系 x''轴旋转的欧拉角。
- a7: 数据类型: real
参数含义: 关节 7 号轴的坐标。
- ... : 参数 a8~a15 的属性跟上述一样, 都是 real 型变量, 依次表示关节 8~关节 15 号轴的坐标值。
- a16: 数据类型: real
参数含义: 关节 16 号轴的坐标。
备注: 机器人有几个轴就会显示几个轴的参数。

DAPOS

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

参数说明：

- da1: 数据类型: real
参数含义: 关节 1 号轴角度的偏移量。
- da2: 数据类型: real
参数含义: 关节 2 号轴角度的偏移量。
- da3: 数据类型: real
参数含义: 关节 3 号轴角度的偏移量。
- ... : 参数 da4~da15 的属性跟上述一样, 都是 real 型变量, 依次表示关节 4~关节 15 号轴角度的偏移量。
- da16: 数据类型: real
参数含义: 关节 16 号轴角度的偏移量。
备注: 机器人有几个轴就会显示几个轴的参数。如六轴机器人显示 da1~da6,scara 机器人显示为 da1~da4。

DCPOS

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

参数说明：

- dx: 数据类型: real
参数含义: TCP 点在参考坐标系上 x 方向的位移偏移量。

- dy: 数据类型: real
参数含义: TCP 点在参考坐标系上 y 方向的位移偏移量。
 - dz: 数据类型: real
参数含义: TCP 点在参考坐标系上 z 方向的位移偏移量。
 - da: 数据类型: real
参数含义: TCP 点相对参考坐标系 z 轴旋转的欧拉角偏移量。
 - db: 数据类型: real
参数含义: TCP 点相对参考坐标系 y'轴旋转的欧拉角偏移量。
 - dc: 数据类型: real
参数含义: TCP 点相对参考坐标系 x''轴旋转的欧拉角偏移量。
 - da7: 数据类型: real
参数含义: 关节 7 号轴角度偏移量。
 - ... : 参数 da8~da15 的属性跟上述一样, 都是 real 型变量, 依次表示关节 8~关节 15 号轴角度偏移量。
 - da16: 数据类型: real
参数含义: 关节 16 号轴角度偏移量。
- 备注: 机器人有几个轴就会显示几个轴的参数。

2.3 系统数据类型

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, 如下图所示:

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

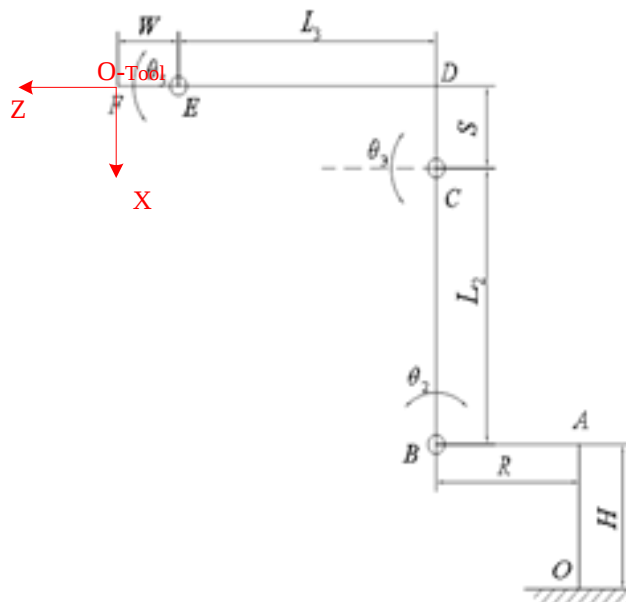


图 1. 工具&负载质量信息坐标系说明

注: 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。

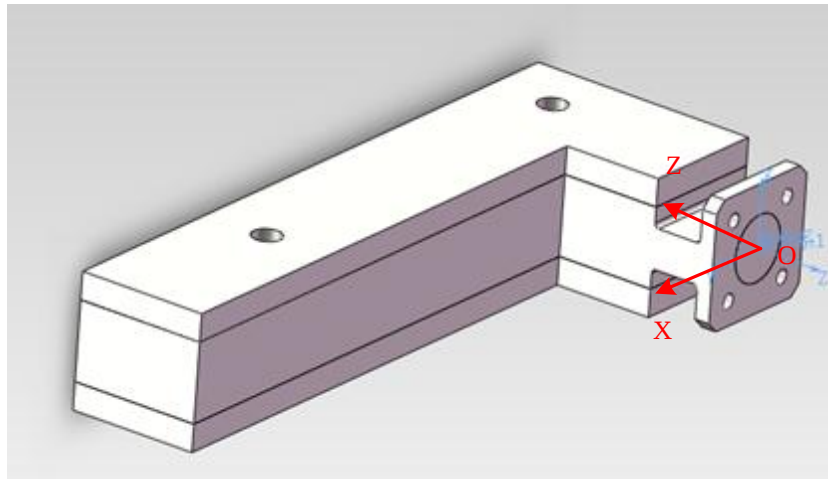


图 2. 工具&负载质量信息坐标系实例

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

参数说明：

- Mx ：数据类型：real
参数含义：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 ：数据类型：real
参数含义：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 ：数据类型：real
参数含义：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。

InertiaTensor

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

参数说明：

- Ixx ：数据类型：real
参数含义：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 $kg \cdot mm^2$ 。
- Iyy ：数据类型：real
参数含义：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 $kg \cdot mm^2$ 。
- Izz ：数据类型：real
参数含义：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 $kg \cdot mm^2$ 。
- Ixy ：数据类型：real
参数含义：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 $kg \cdot mm^2$ 。
- Ixz ：数据类型：real
参数含义：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 $kg \cdot mm^2$ 。
- Iyz ：数据类型：real
参数含义：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 $kg \cdot mm^2$ 。

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：

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

TOOL

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

参数说明:

- id: 数据类型: int
参数含义: 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: 数据类型: real
参数含义: TCP 相对于法兰坐标系在 x 方向的位移偏移量, 单位是 mm。
- y: 数据类型: real
参数含义: TCP 相对于法兰坐标系在 y 方向的位移偏移量, 单位是 mm。
- z: 数据类型: real
参数含义: TCP 相对于法兰坐标系在 z 方向的位移偏移量, 单位是 mm。
- a: 数据类型: real
参数含义: TCP 相对于法兰坐标系 z 轴旋转的欧拉角, 单位是 deg。
- b: 数据类型: real
参数含义: TCP 相对于法兰坐标系 y'轴旋转的欧拉角, 单位是 deg。
- c: 数据类型: real
参数含义: TCP 相对于法兰坐标系 x''轴旋转的欧拉角, 单位是 deg。
- dyn: 数据类型: LoadDyn
参数含义: 工具的质量信息, 用于机器人动力学全模型计算。

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

USERCOOR

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

参数说明:

- id: 数据类型: int
参数含义: 用户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: 数据类型: real
参数含义: 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: 数据类型: real
参数含义: 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: 数据类型: real
参数含义: 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: 数据类型: real

参数含义：TCP 相对于世界（基）坐标系 z 轴旋转的欧拉角，单位是 deg。

- b: 数据类型: real

参数含义：TCP 相对于世界（基）坐标系 y'轴旋转的欧拉角，单位是 deg。

- c: 数据类型: real

参数含义：TCP 相对于世界（基）坐标系 x''轴旋转的欧拉角，单位是 deg。

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

EXTTCP

EXTTCP 型变量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。该参数可由用户根据变量界面标定计算所得，也可由用户自行输入。该变量只允许在全局域创建以及修改。

参数说明：

- id: 数据类型: int

参数含义：外部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: 数据类型: real

参数含义：TCP 相对于工具坐标系在 x 方向的位移偏移量，单位是 mm。

- y: 数据类型: real

参数含义：TCP 相对于工具坐标系在 y 方向的位移偏移量，单位是 mm。

- z: 数据类型: real

参数含义：TCP 相对于工具坐标系在 z 方向的位移偏移量，单位是 mm。

- a: 数据类型: real

参数含义：TCP 相对于工具坐标系 z 轴旋转的欧拉角，单位是 deg。

- b: 数据类型: real

参数含义：TCP 相对于工具坐标系 y'轴旋转的欧拉角，单位是 deg。

- c: 数据类型: real

参数含义：TCP 相对于工具坐标系 x''轴旋转的欧拉角，单位是 deg。

SYNCOORD

SYNCOORD 型变量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。该参数可由用户根据变量界面标定计算所得，也可由用户自行输入。该变量只允许在全局域创建以及修改。

参数说明：

- id: 数据类型: int

参数含义：外部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: 数据类型: real

参数含义：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: 数据类型: real

参数含义：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: 数据类型: real

参数含义：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: 数据类型: real

参数含义：TCP 相对于世界（基）坐标系 z 轴旋转的欧拉角，单位是 deg。

- b: 数据类型: real

参数含义：TCP 相对于世界（基）坐标系 y'轴旋转的欧拉角，单位是 deg。

- c: 数据类型: real

参数含义: TCP 相对于世界 (基) 坐标系 x”轴旋转的欧拉角, 单位是 deg。

POSITIONER

POSITIONER 型变量通过 index 来选择变位机 (ID)。

参数说明:

- index: 变位机序号

数据类型: int

参数含义: index 取值范围为[0, 6]。其中 0 表示该变量无效, 其它表示变位机 ID 为 index 的坐标系。

SPEED

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

参数说明:

- per: 关节速度百分比

数据类型: real

参数含义: 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

数据类型: real

参数含义: 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

数据类型: real

参数含义: 定义机器人末端点姿态的旋转速度。

- exj_l: 外部轴线速度

数据类型: real

参数含义: 定义外部直线轴的运动速度。

- exj_r: 外部轴角速度

数据类型: real

参数含义: 定义外部旋转轴的运动速度。

ZONE

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

参数说明:

- per: 关节轨迹转弯百分比

数据类型: real

参数含义: 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

数据类型: real

参数含义：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

数据类型：int

参数含义：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。

注：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。

PAYLOAD

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

参数说明：

- id：数据类型：int

参数含义：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：数据类型：LoadDyn

参数含义：工件负载的质量信息，用于机器人动力学全模型计算。

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

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。该变量只允许在全局域创建以及修改。

参数说明：

- J1：数据类型：int

参数含义：J1 轴对应的增益比例阈值，单位：%。

- J2：数据类型：int

参数含义：J2 轴对应的增益比例阈值，单位：%。

- J3：数据类型：int

参数含义：J3 轴对应的增益比例阈值，单位：%。

- J4：数据类型：int

参数含义：J4 轴对应的增益比例阈值，单位：%。

- J5：数据类型：int

参数含义：J5 轴对应的增益比例阈值，单位：%。

- J6：数据类型：int

参数含义：J6 轴对应的增益比例阈值，单位：%。

LsThresh

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

参数说明：

- J1：数据类型：int

参数含义：J1 轴对应的速度区间阈值，单位：%。

- J2：数据类型：int

参数含义：J2 轴对应的速度区间阈值，单位：%。

- J3：数据类型：int

参数含义：J3 轴对应的速度区间阈值，单位：%。

- J4：数据类型：int

参数含义：J4 轴对应的速度区间阈值，单位：%。

- J5：数据类型：int

参数含义：J5 轴对应的速度区间阈值，单位：%。

- J6：数据类型：int

参数含义：J6 轴对应的速度区间阈值，单位：%。

2.4 IO 数据类型

DI

数字量输入变量。

参数说明：

- port：数据类型：int

参数含义：与该变量绑定的数字量输入端口号。

- value：数据类型：int

参数含义：取值范围为 0 或 1，代表对应端口输入状态。

- riseSts：数据类型：bool

参数含义：该端口上升沿信号状态。

- downSts：数据类型：bool

参数含义：该端口下降沿状态。

DO

数字量输出变量。

参数说明：

- port：数据类型：int

参数含义：与该变量绑定的数字量输出端口号。

- value：数据类型：int

参数含义：取值范围为 0 或 1，代表对应端口输出状态。

AI

模拟量输入变量。

参数说明：

- port：数据类型：int

参数含义：与该变量绑定的模拟量输入端口号。

- value：数据类型：real

参数含义：取值范围为-10~10V，代表对应端口输入电压值。

AO

模拟量输出变量。

参数说明：

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

SimDI

虚拟数字量输入变量。

参数说明：

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

SimDO

虚拟数字量输出变量。

参数说明：

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

SimAI

虚拟模拟量输入变量。

参数说明：

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

SimAO

虚拟模拟量输出变量。

参数说明：

- port: 数据类型: int
参数含义: 与该变量绑定的虚拟模拟量输出端口号。

- value: 数据类型: real
参数含义: 代表对应端口输出值。

2.5 摆动数据类型

WEAVE

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

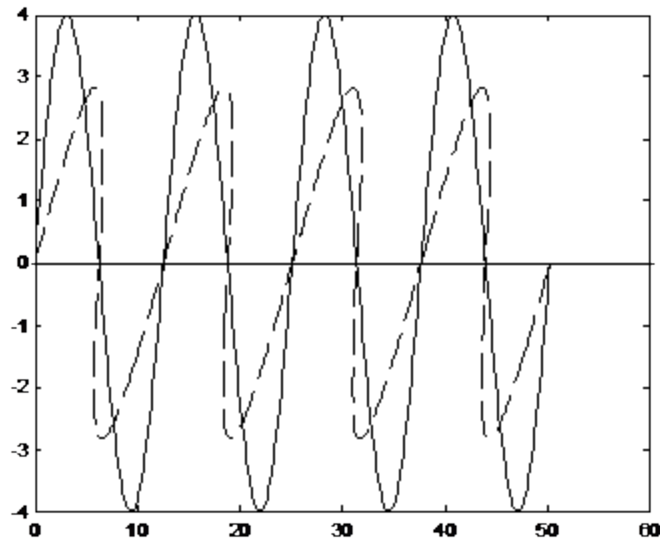
参数说明:

- Type: 数据类型: int
参数含义: 机器人摆动类型, 该为 1 表示正弦摆, 该值为 2 表示三角摆。
- Freq: 数据类型: real
参数含义: 机器人摆动时的频率, 单位: Hz。
- Amp_L: 数据类型: real
参数含义: 机器人摆动左幅值, 单位: mm。
- Amp_R: 数据类型: real
参数含义: 机器人摆动右幅值, 单位: mm。
- StopTime_L: 数据类型: int
参数含义: 波峰处停止时间, 指在摆动到左幅值时停止的时间, 单位是 ms。
- StopTime_R: 数据类型: int
参数含义: 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。
- StopTime_C: 数据类型: int
参数含义: 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。
- RotAngle_X: 数据类型: real
参数含义: 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。
- RotAngle_Z: 数据类型: real
参数含义: 绕摆动平面的法向量旋转的角度, 决定摆弧的形状, 单位: deg



注意

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



2.6 时钟数据类型

CLOCK

存储时钟信息的值。该变量仅支持在全局作用域中创建，删除，修改等操作。

参数说明：

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

2.7 区域数据类型

AREA

存储标准区域信息的值。该变量仅支持在全局用域中创建，删除，修改等操作。

参数说明：

- id: 数据类型: int
参数含义：标准区域变量的编号。
- activate: 数据类型: bool
参数含义：是否激活此区域以及此区域是否在激活状态。
- isInArea: 数据类型: bool

参数含义：是否在区域内。

- initActivate: 数据类型: bool

参数含义：区域自动激活标识。

- areaType: 数据类型: int

参数含义：区域类型。

- areaShape: 数据类型: int

参数含义：区域形状。

- refSysScope: 数据类型: int

参数含义：参考坐标系域。

- refSysName: 数据类型: string

参数含义：参考坐标系名字。

- enInPut: 数据类型: bool

参数含义：使能输入。

- enInHigh: 数据类型: bool

参数含义：输入高有效。

- enOutPut: 数据类型: bool

参数含义：使能输出。

- enOutHigh: 数据类型: bool

参数含义：输出高有效。

- diType: 数据类型: int

参数含义：DI 类型。

- InputActStatePort: 数据类型: int

参数含义：输入实际状态。

- doType: 数据类型: int

参数含义：DO 类型,。

- OutputActStatePort: 数据类型: int

参数含义：输出实际状态。

- StartPointX: 数据类型: real

参数含义：区域起点的坐标 x。

- StartPointY: 数据类型: real

参数含义：区域起点的坐标 y。

- StartPointZ: 数据类型: real

参数含义：区域起点的坐标 z。

- lenXR: 数据类型: real

参数含义：矩形区域的长度或者圆柱形区域的半径。

- lenYH: 数据类型: real

参数含义：矩形区域的宽度或者圆柱形区域的高度。

- lenZ: 数据类型: real

参数含义：矩形区域的高度。

- isFlangeEnd: 数据类型: int

参数含义：监控端参数，在区域监控中用于监控 TCP 或法兰端。

POLYHEDRON

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

- index: 数据类型: int
参数含义: 多边形区域的编号, index 取值范围为[0, 4]。其中 0 表示该变量无效, 其它表示 ID 为 index 的多边形区域。
- activate: 数据类型: bool
参数含义: 多边形区域是否在激活状态。
- isInArea: 数据类型: bool
参数含义: 是否在多边形区域内。

2.8 PLC 数据类型

PLCREAL

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

参数说明:

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

PLCBOOL

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

参数说明:

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

PLCINT

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

参数说明:

- plcNum: 数据类型: int
参数含义: 数据编号
- value: 数据类型: int

参数含义：变量值

- saveflag: 数据类型: bool

参数含义：掉电保存标记

PLCDINT

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

参数说明：

- plcNum: 数据类型: int

参数含义：数据编号

- value: 数据类型: int

参数含义：变量值

- saveflag: 数据类型: bool

参数含义：掉电保存标记

2.9 码垛数据类型

PLACEOPTIONS

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

参数说明：

- isUse: 数据类型: bool

参数含义：参数是否有效。

- sideOffset: 数据类型: real

参数含义：水平偏移。

- height: 数据类型: real

参数含义：高度偏移。

PALLET

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

参数说明：

- actParts: 数据类型: int

参数含义：垛体中实际工件数。

- maxParts: 数据类型: int

参数含义：垛体中最大工件数。

- isEmpty: 数据类型: bool

参数含义：垛体是否为空。

- isFull: 数据类型: bool

参数含义：垛体是否为满。

- xNum: 数据类型: int

参数含义：垛体 x 方向上工件数。

- yNum: 数据类型: int

参数含义：垛体 y 方向上工件数。

- zNum：数据类型：int

参数含义：垛体 z 方向上工件数。

- xdistOfPart：数据类型：real

参数含义：垛体 x 方向上单个工件长度。

- ydistOfPart：数据类型：real

参数含义：垛体 y 方向上单个工件长度。

- zdistOfPart：数据类型：real

参数含义：垛体 z 方向上单个工件长度。

- refSysScope：数据类型：intl

参数含义：垛体参数所参考坐标系域。

- refSysName：数据类型：string

参数含义：垛体参数所参考坐标系名称。

- palletDir：数据类型：int

参数含义：码垛或拆垛方向。

- palletOrder：数据类型：int

参数含义：码垛或拆垛顺序。

- firstPartScope：数据类型：int

参数含义：第一个工件位置变量作用域。

- firstPartPos：数据类型：string

参数含义：第一个工件位置变量名称。

- isEntryPosUsed：数据类型：bool

参数含义：是否行至码垛入口点。

- palletEntryScope：数据类型：int

参数含义：码垛入口点位置变量作用域。

- palletEntryPos：数据类型：string

参数含义：码垛入口点位置点名称。

- preplaceOptions：数据类型：PLACEOPTIONS

参数含义：前置点信息。

- postplaceOptions：数据类型：PLACEOPTIONS

参数含义：后置点信息。

2.10 SOCKET 数据类型

Socket

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

参数说明：

- value：数据类型：string

参数含义：socket 名称。

第 3 章 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
	MovLOffset
	MovLW
	MovCW
	MovCircleW
	MovArch
	MovH
	OnDistance
	OnParameter

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。对指令参数进行说明：

MovJ P [V] [B] [C] [Tool][Coord] [PayLoad][DO]

（备注：带[]参数为可选参数）

- P：目标位置（Target Pos）

数据类型：APOS 或 CPOS 变量

参数含义：坐标系下的目标点位置

- [V]：目标速度 (Velo)

数据类型：SPEED 变量

参数含义：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。

备注：该参数可缺省，默认值为 V4000。

- [B]：过渡类型 (BlendType)

数据类型：enum (FINE/RELATIVE/DEFAULT)

参数含义：机器人逼近终点时的过渡方式

- FINE：无过渡。

- RELATIVE：相对过渡。

- ABSOLUTE：绝对过渡。

- DEFAULT：缺省类型。

备注：该参数可缺省，默认值为 FINE。

- [C]：过渡值 (BlendValue)

数据类型：ZONE 变量

参数含义：机器人逼近终点时的过渡值。

备注：该参数可缺省，默认值为 C100。

- [Tool]：工具参数 (Tool)

数据类型：TOOL 变量

参数含义：机器人执行该轨迹时使用的工具参数。

备注：该参数可缺省，默认值为使用当前系统中设置的工具参数。

- [Coord]：坐标系参数 (Coord)

数据类型：USERCOOR 变量

参数含义：机器人执行该轨迹时使用的用户坐标系。

备注：该参数可缺省，默认值为使用当前系统中设置的坐标系参数。

- [PayLoad]：工件负载 (PayLoad)

数据类型：PAYLOAD 变量

参数含义：机器人执行该轨迹时使用的工件负载参数。

备注：该参数可缺省，默认值为使用当前系统中设置的负载参数。

- [DO]：运动完成后执行的 IO 操作 (Add Do)

数据类型：string

参数含义：机器人执行完该指令后，可触发的 IO 操作：

- NULL：无任何操作。

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

示例 1：

```
MovJ (P1, V50, "RELATIVE", 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, V50, "RELATIVE", C0)Do SetDo(DO0,0)
```

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, V50, "RELATIVE", C100,nullTool,World)
MovJ (P2, V50, "RELATIVE", C100,Tool1,UserCoord1)
```

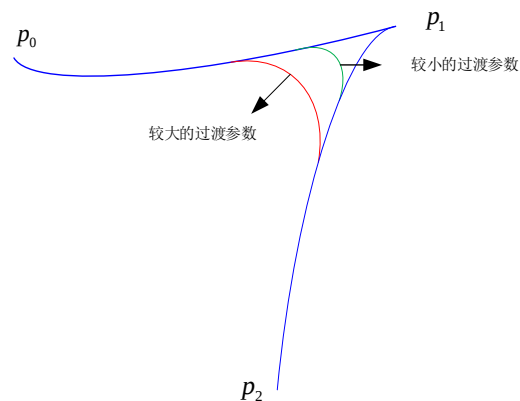
机器人以 50 的速度运动到 P1 点，其后由于 P2 点坐标系与 P1 点不一致，故会运动到 P2 前会先自动将工具参数切换成 Tool1，用户坐标系切换成 UserCoord1，其后在运动到 P2，由于自动切换了坐标系，此时从 P1 运动到 P2 的过程中无过渡效果。

示例 4:

```
MovJ (P1, V50, "RELATIVE", C100,Tool1,UserCoord1)
MovL (P2, V50, "RELATIVE", C100,Tool1)
```

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



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。对指令参数进行说明：

MovLP [V] [B] [C][Tool][Coord] [PayLoad][O] [DO]

（备注：带[]参数为可选参数）

- P：目标位置（Target Pos）
数据类型：APOS 或 CPOS 变量
参数含义：坐标系下的目标点位置
- [V]：目标速度（Velo）
数据类型：SPEED 变量
参数含义：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。
备注：该参数可缺省，默认值为 V4000。
- [B]：过渡类型（BlendType）
数据类型：enum（FINE/RELATIVE/DEFAULT）
参数含义：机器人逼近终点时的过渡方式
 - FINE：无过渡。
 - RELATIVE：相对过渡。
 - ABSOLUTE：绝对过渡。
 - DEFAULT：缺省类型。

备注：该参数可缺省，默认值为 FINE。

- [C]: 过渡值 (BlendValue)

数据类型: ZONE 变量

参数含义: 机器人逼近终点时的过渡值。

备注: 该参数可缺省，默认值为 C100。

- [Tool]: 工具参数 (Tool)

数据类型: TOOL 变量

参数含义: 机器人执行该轨迹时使用的工具参数。

备注: 该参数可缺省，默认值为使用当前系统中设置的工具参数。

- [Coord]: 坐标系参数 (Coord)

数据类型: USERCOOR 变量或 EXTTCP 变量或变位机变量

参数含义: 机器人执行该轨迹时使用的用户坐标系/外部 TCP 坐标系/变位机坐标系。

备注: 该参数可缺省，默认值为使用当前系统中设置的坐标系参数。

- [PayLoad]: 工件负载 (PayLoad)

数据类型: PAYLOAD 变量

参数含义: 机器人执行该轨迹时使用的工件负载参数。

备注: 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 (OverrideType)

数据类型: enum (GOVRON/GOVROFF)

- GOVRON: 本段轨迹受全局速度倍率影响。
- GOVROFF: 本段轨迹不受全局速度倍率影响。

参数含义: 设置当前指令轨迹是否受全局速度倍率影响。

备注: 该参数可缺省，默认值为 GOVRON。

- [DO]: 运动完成后执行的 IO 操作 (Add Do)

数据类型: string

参数含义: 机器人执行完该指令后，可触发的 IO 操作:

- NULL: 无任何操作。
- IO 指令: 执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

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。

MovCA P[V] [B] [C][Tool][Coord] [PayLoad][O] [DO]

(备注: 带[]参数为可选参数)

- A: 中间位置 (AuxPos)

数据类型: APOS 或 CPOS 变量

参数含义: 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)

数据类型: APOS 或 CPOS 变量

参数含义: 坐标系下的目标点位置

- [V]: 目标速度 (Velo)

数据类型: SPEED 变量

参数含义：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。

备注：该参数可缺省，默认值为 V4000。

- [B]：过渡类型 (BlendType)

数据类型：enum (FINE/RELATIVE/DEFAULT)

参数含义：机器人逼近终点时的过渡方式

- FINE：无过渡。
- RELATIVE：相对过渡。
- ABSOLUTE：绝对过渡。
- DEFAULT：缺省类型。

备注：该参数可缺省，默认值为 FINE。

- [C]：过渡值 (BlendValue)

数据类型：ZONE 变量

参数含义：机器人逼近终点时的过渡值。

备注：该参数可缺省，默认值为 C100。

- [Tool]：工具参数 (Tool)

数据类型：TOOL 变量

参数含义：机器人执行该轨迹时使用的工具参数。

备注：该参数可缺省，默认值为使用当前系统中设置的工具参数。

- [Coord]：坐标系参数 (Coord)

数据类型：USERCOORD 变量或 EXTTCP 变量或变位机变量

参数含义：机器人执行该轨迹时使用的用户坐标系/外部 TCP 坐标系/变位机坐标系。

备注：该参数可缺省，默认值为使用当前系统中设置的坐标系参数。

- [PayLoad]：工件负载 (PayLoad)

数据类型：PAYLOAD 变量

参数含义：机器人执行该轨迹时使用的工件负载参数。

备注：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 (OverrideType)

数据类型：enum (GOVRON/GOVROFF)

- GOVRON：本段轨迹受全局速度倍率影响。
- GOVROFF：本段轨迹不受全局速度倍率影响。

参数含义：设置当前指令轨迹是否受全局速度倍率影响。

备注：该参数可缺省，默认值为 GOVRON。

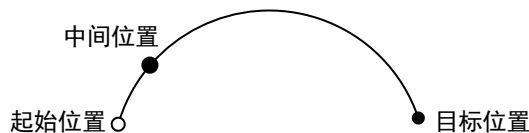
- [DO]：运动完成后执行的 IO 操作 (Add Do)

数据类型：string

参数含义：机器人执行完该指令后，可触发的 IO 操作：

- NULL：无任何操作。
- IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

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。

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][Tool] [Coord] [PayLoad] [O] [DO]

（备注：带[]参数为可选参数）

- A: 中间位置 1 (AuxPos1)
数据类型: APOS 或 CPOS 变量
参数含义: 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)
数据类型: APOS 或 CPOS 变量
参数含义: 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)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
– FINE: 无过渡。
– RELATIVE: 相对过渡。
– ABSOLUTE: 绝对过渡。
– DEFAULT: 缺省类型。
备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE 变量
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省, 默认值为 C100。
- [Tool]: 工具参数 (Tool)
数据类型: TOOL 变量
参数含义: 机器人执行该轨迹时使用的工具参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)
数据类型: USERCOORD 变量或 EXTTCP 变量或变位机变量

参数含义：机器人执行该轨迹时使用的用户坐标系/外部 TCP 坐标系/变位机坐标系。

备注：该参数可缺省，默认值为使用当前系统中设置的坐标系参数。

- [PayLoad]：工件负载 (PayLoad)

数据类型：PAYLOAD 变量

参数含义：机器人执行该轨迹时使用的工件负载参数。

备注：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 (OverrideType)

数据类型：enum (GOVRON/GOVROFF)

- GOVRON：本段轨迹受全局速度倍率影响。
- GOVROFF：本段轨迹不受全局速度倍率影响。

参数含义：设置当前指令轨迹是否受全局速度倍率影响。

备注：该参数可缺省，默认值为 GOVRON。

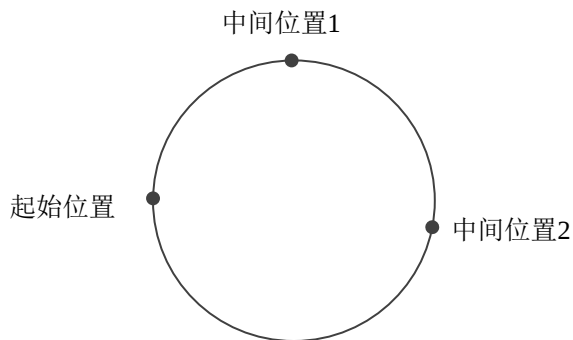
- [DO]：运动完成后执行的 IO 操作 (Add Do)

数据类型：string

参数含义：机器人执行完该指令后，可触发的 IO 操作：

- NULL：无任何操作。
- IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

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；

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。其参数定义如下：

MovJRelP[V] [B] [C] [PayLoad][DO]

(备注：带[]参数为可选参数)

- P: 目标相对位置 (RelPos)
数据类型: DAPOS 变量
参数含义: 执行该指令时机器人要移动的位置增量
- [V]: 目标速度 (Velo)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - RELATIVE: 相对过渡。
 - ABSOLUTE: 绝对过渡。
 - DEFAULT: 缺省类型。
 备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE 变量
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省, 默认值为 C100。
- [PayLoad]: 工件负载 (PayLoad)
数据类型: PAYLOAD 变量
参数含义: 机器人执行该轨迹时使用的工件负载参数。
备注: 该参数可缺省, 默认值为当前系统中设置的负载参数。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)
数据类型: string
参数含义: 机器人执行完该指令后, 可触发的 IO 操作:
 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

示例 1:

```
MovJ(P1, V50, "RELATIVE", C100)
MovJRel (DAPOS0, V50, "RELATIVE", 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, V50, "RELATIVE", 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。
```

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, 其参数定义如下图:

MovLRelPT [V] [B] [C][PayLoad][O] [DO]

(备注：带[]参数为可选参数)

- P: 目标相对位置 (RelPos)

数据类型: DCPOS 变量

参数含义: 执行该指令时机器人要移动的位置增量
- T: 参考坐标系 (RefSys)

数据类型: enum (COORD/TOOL)

 - COORD: 笛卡尔坐标系。
 - TOOL: 工具坐标系。

参数含义: 参考坐标系: 笛卡尔或工具
- [V]: 目标速度 (Velo)

数据类型: SPEED 变量

参数含义: 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。

备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)

数据类型: enum (FINE/RELATIVE/DEFAULT)

参数含义: 机器人逼近终点时的过渡方式

 - FINE: 无过渡。
 - RELATIVE: 相对过渡。
 - ABSOLUTE: 绝对过渡。
 - DEFAULT: 缺省类型。

备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)

数据类型: ZONE 变量

参数含义: 机器人逼近终点时的过渡值。

备注: 该参数可缺省, 默认值为 C100。
- [PayLoad]: 工件负载 (PayLoad)

数据类型: PAYLOAD 变量

参数含义: 机器人执行该轨迹时使用的工件负载参数。

备注: 该参数可缺省, 默认值为当前系统中设置的负载参数。
- O: 速率类型 (OverrideType)

数据类型: enum (GOVRON/GOVROFF)

 - GOVRON: 本段轨迹受全局速度倍率影响。
 - GOVROFF: 本段轨迹不受全局速度倍率影响。

参数含义: 设置当前指令轨迹是否受全局速度倍率影响。

备注: 该参数可缺省, 默认值为 GOVRON。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)

数据类型: string

参数含义: 机器人执行完该指令后, 可触发的 IO 操作:

 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

示例 1:

```
MovL(P1, V50, "RELATIVE", C100)
```

MovLRel (DCPOS0, V50, "RELATIVE", 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, V50, "RELATIVE", 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。

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。

请注意，该指令会打断过渡。

其指令参数定义如下：

MovJSearchP [V] [B] [C][Tool][Coord] [PayLoad]Type Number Value [StopType]
[StopDecScale] R [DO]

(备注：带[]参数为可选参数)

- P: 目标位置 (Target Pos)
数据类型: APOS 或 CPOS 变量
参数含义: 坐标系下的目标点位置
- [V]: 目标速度 (Velo)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省，默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - DEFAULT: 缺省类型。
 备注: 该参数可缺省，默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省，默认值为 C100。
- [Tool]: 工具参数 (Tool)
数据类型: TOOL 变量
参数含义: 机器人执行该轨迹时使用的工具参数。
备注: 该参数可缺省，默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)
数据类型: USERCOOR 变量
参数含义: 机器人执行该轨迹时使用的用户坐标系。
备注: 该参数可缺省，默认值为使用当前系统中设置的坐标系参数。

- [PayLoad]: 工件负载 (PayLoad)

数据类型: PAYLOAD 变量

参数含义: 机器人执行该轨迹时使用的工件负载参数。

备注: 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
- Type: 检测类型 (SearchType)

数据类型: enum (DITrig/AITrig/SIMDITrig/SIMAITrig/TorqTrig)

 - DITrig: 物理数字输入口触发。
 - AITrig: 物理模拟量输入口触发。
 - SIMDITrig: 虚拟数字输入口触发。
 - SIMAITrig: 虚拟模拟量输入口触发。
 - TorqTrig: 力矩触发。

参数含义: 机器人运行该轨迹时需要监控的操作类型。
- Number: 触发索引 (SearchIndex)

数据类型: int

参数含义: 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。
- Value: 触发值 (SearchValue)

数据类型: real

参数含义: 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。
- StopType: 停止类型 (StopType)

数据类型: enum (HARDSTOP/SOFTSTOP/DEFAULT)

 - HARDSTOP: 立即停止。
 - SOFTSTOP: 减速停止。
 - DEFAULT: 缺省类型。

参数含义: 检测到信号时停止的方式。

备注: 该参数可缺省, 默认值为 HARDSTOP。
- StopDecScale: 减速系数 (StopDecScale)

数据类型: real。

参数含义: 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。
- R: 返回值 (ResultV)

数据类型: INT 变量

参数含义: 返回该条指令执行结束的状态。返回 0 表示未满足条件触而结束, 返回 1 表示满足条件触而结束。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)

数据类型: string

参数含义: 机器人执行完该指令后, 可触发的 IO 操作:

 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

示例 1:

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

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

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。

请注意，该指令会打断过渡。

具体指令参数定义如下：

MovLSearchP [V] [B] [C][Tool][Coord] [PayLoad]Type Number Value [StopType]
[StopDecScale] R [DO]

(备注：带[]参数为可选参数)

- P: 目标位置 (Target Pos)
数据类型: APOS 或 CPOS 变量
参数含义: 坐标系下的目标点位置
- [V]: 目标速度 (Velo)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省，默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - DEFAULT: 缺省类型。
 备注: 该参数可缺省，默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省，默认值为 C100。
- [Tool]: 工具参数 (Tool)
数据类型: TOOL 变量
参数含义: 机器人执行该轨迹时使用的工具参数。
备注: 该参数可缺省，默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)
数据类型: USERCOORD 变量或 EXTTCOORD 变量或变位机变量
参数含义: 机器人执行该轨迹时使用的用户坐标系/外部 TCP 坐标系/变位机坐标系。
备注: 该参数可缺省，默认值为使用当前系统中设置的坐标系参数。
- [PayLoad]: 工件负载 (PayLoad)
数据类型: PAYLOAD 变量
参数含义: 机器人执行该轨迹时使用的工件负载参数。

备注：该参数可缺省，默认值为使用当前系统中设置的负载参数。

- Type: 检测类型 (SearchType)

数据类型: enum (DITrig/AITrig/SIMDITrig/SIMAITrig/TorqTrig)

- DITrig: 物理数字输入口触发。
- AITrig: 物理模拟量输入口触发。
- SIMDITrig: 虚拟数字输入口触发。
- SIMAITrig: 虚拟模拟量输入口触发。
- TorqTrig: 力矩触发。

参数含义: 机器人运行该轨迹时需要监控的操作类型。

- Number: 触发索引 (SearchIndex)

数据类型: int

参数含义: 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。

- Value: 触发值 (SearchValue)

数据类型: real

参数含义: 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。

- StopType: 停止类型 (StopType)

数据类型: enum (HARDSTOP/SOFTSTOP/DEFAULT)

- HARDSTOP: 立即停止。
- SOFTSTOP: 减速停止。
- DEFAULT: 缺省类型。

参数含义: 检测到信号时停止的方式。

备注: 该参数可缺省, 默认值为 HARDSTOP。

- StopDecScale: 减速系数 (StopDecScale)

数据类型: real。

参数含义: 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。

- R: 返回值 (ResultV)

数据类型: INT 变量

参数含义: 返回该条指令执行结束的状态。返回 0 表示未满足条件触而结束, 返回 1 表示满足条件触而结束。

- [DO]: 运动完成后执行的 IO 操作 (Add Do)

数据类型: string

参数含义: 机器人执行完该指令后, 可触发的 IO 操作:

- NULL: 无任何操作。
- IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

示例 1:

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

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

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。

具体指令参数定义如下：

MovLOffset RPTP T Tool Coord [V] [B] [C][PayLoad][O] [DO]

(备注：带[]参数为可选参数)

- RP: 参考位置 (RefPos)
 - 数据类型: APOS 或 CPOS 变量
 - 参数含义: 坐标系下的参考点位置
- TP: 目标相对位置 (RelPos)
 - 数据类型: DCPOS 变量
 - 参数含义: 相对于参考位置的位置增量
- T: 参考坐标系 (RefSys)
 - 数据类型: enum (COORD/TOOL)
 - COORD: 笛卡尔坐标系。
 - TOOL: 工具坐标系。
 - 参数含义: 参考坐标系: 笛卡尔或工具
- Tool: 工具参数 (Tool)
 - 数据类型: TOOL 变量
 - 参数含义: 机器人执行该轨迹时使用的工具参数。
- Coord: 坐标系参数 (Coord)
 - 数据类型: USERCOOR 变量
 - 参数含义: 机器人执行该轨迹时使用的用户坐标系。
- [V]: 目标速度 (Velo)
 - 数据类型: SPEED 变量
 - 参数含义: 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。
 - 备注: 该参数可缺省，默认值为 V4000。
- [B]: 过渡类型 (BlendType)
 - 数据类型: enum (FINE/RELATIVE/DEFAULT)
 - 参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - RELATIVE: 相对过渡。
 - ABSOLUTE: 绝对过渡。
 - DEFAULT: 缺省类型。
 - 备注: 该参数可缺省，默认值为 FINE。
- [C]: 过渡值 (BlendValue)
 - 数据类型: ZONE
 - 参数含义: 机器人逼近终点时的过渡值。
 - 备注: 该参数可缺省，默认值为 C100。
- [PayLoad]: 工件负载 (PayLoad)
 - 数据类型: PAYLOAD 变量
 - 参数含义: 机器人执行该轨迹时使用的工件负载参数。

备注：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 (OverrideType)

数据类型: enum (GOVRON/GOVROFF)

- GOVRON: 本段轨迹受全局速度倍率影响。
- GOVROFF: 本段轨迹不受全局速度倍率影响。

参数含义: 设置当前指令轨迹是否受全局速度倍率影响。

备注: 该参数可缺省，默认值为 GOVRON。

- [DO]: 运动完成后执行的 IO 操作 (Add Do)

数据类型: string

参数含义: 机器人执行完该指令后，可触发的 IO 操作:

- NULL: 无任何操作。
- IO 指令: 执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

示例 1:

```
MovLOffset(P1, DCPOS0, "COORD", nullTool, World, V50, "RELATIVE", 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。

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。

其指令参数定义如下图所示:

MovLWP [V] [B] [C][Tool][Coord] [PayLoad]WEAVE [O] [DO]

(备注: 带[]参数为可选参数)

- P: 目标位置 (Target Pos)

数据类型: APOS 或 CPOS 变量

参数含义: 坐标系下的目标点位置

- [V]: 目标速度 (Velo)

数据类型: SPEED 变量

参数含义: 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。

备注: 该参数可缺省，默认值为 V4000。

- [B]: 过渡类型 (BlendType)

数据类型: enum (FINE/RELATIVE/DEFAULT)

参数含义: 机器人逼近终点时的过渡方式

- FINE: 无过渡。
- RELATIVE: 相对过渡。
- ABSOLUTE: 绝对过渡。
- DEFAULT: 缺省类型。

备注: 该参数可缺省，默认值为 FINE。

- [C]: 过渡值 (BlendValue)

数据类型: ZONE

参数含义: 机器人逼近终点时的过渡值。

备注: 该参数可缺省，默认值为 C100。

- [Tool]: 工具参数 (Tool)
 - 数据类型: TOOL 变量
 - 参数含义: 机器人执行该轨迹时使用的工具参数。
 - 备注: 该参数可缺省, 默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)
 - 数据类型: USERCOORD 变量或变位机变量
 - 参数含义: 机器人执行该轨迹时使用的用户坐标系/变位机坐标系。
 - 备注: 该参数可缺省, 默认值为使用当前系统中设置的坐标系参数。
- [PayLoad]: 工件负载 (PayLoad)
 - 数据类型: PAYLOAD 变量
 - 参数含义: 机器人执行该轨迹时使用的工件负载参数。
 - 备注: 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
- WEAVE: 摆弧变量 (Weave Value)
 - 数据类型: WEAVE 变量
 - 参数含义: 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。
- O: 速率类型 (OverrideType)
 - 数据类型: enum (GOVRON/GOVROFF)
 - GOVRON: 本段轨迹受全局速度倍率影响。
 - GOVROFF: 本段轨迹不受全局速度倍率影响。
 - 参数含义: 设置当前指令轨迹是否受全局速度倍率影响。
 - 备注: 该参数可缺省, 默认值为 GOVRON。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)
 - 数据类型: string
 - 参数含义: 机器人执行完该指令后, 可触发的 IO 操作:
 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

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。

其指令参数定义如下图所示:

MovCW AP [V] [B] [C][Tool][Coord] [PayLoad]WEAVE [O] [DO]

(备注: 带[]参数为可选参数)

- A: 中间位置 (AuxPos)
 - 数据类型: APOS 或 CPOS 变量
 - 参数含义: 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)
 - 数据类型: APOS 或 CPOS 变量
 - 参数含义: 坐标系下的目标点位置
- [V]: 目标速度 (Velo)
 - 数据类型: SPEED 变量

参数含义：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。

备注：该参数可缺省，默认值为 V4000。

- [B]: 过渡类型 (BlendType)

数据类型: enum (FINE/RELATIVE/DEFAULT)

参数含义: 机器人逼近终点时的过渡方式

 - FINE: 无过渡。
 - RELATIVE: 相对过渡。
 - ABSOLUTE: 绝对过渡。
 - DEFAULT: 缺省类型。

备注: 该参数可缺省，默认值为 FINE。
- [C]: 过渡值 (BlendValue)

数据类型: ZONE

参数含义: 机器人逼近终点时的过渡值。

备注: 该参数可缺省，默认值为 C100。
- [Tool]: 工具参数 (Tool)

数据类型: TOOL 变量

参数含义: 机器人执行该轨迹时使用的工具参数。

备注: 该参数可缺省，默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)

数据类型: USERCOOR 变量或变位机变量

参数含义: 机器人执行该轨迹时使用的用户坐标系/变位机坐标系。

备注: 该参数可缺省，默认值为使用当前系统中设置的坐标系参数。
- [PayLoad]: 工件负载 (PayLoad)

数据类型: PAYLOAD 变量

参数含义: 机器人执行该轨迹时使用的工件负载参数。

备注: 该参数可缺省，默认值为使用当前系统中设置的负载参数。
- WEAVE: 摆弧变量 (Weave Value)

数据类型: WEAVE 变量

参数含义: 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。
- O: 速率类型 (OverrideType)

数据类型: enum (GOVRON/GOVROFF)

 - GOVRON: 本段轨迹受全局速度倍率影响。
 - GOVROFF: 本段轨迹不受全局速度倍率影响。

参数含义: 设置当前指令轨迹是否受全局速度倍率影响。

备注: 该参数可缺省，默认值为 GOVRON。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)

数据类型: string

参数含义: 机器人执行完该指令后，可触发的 IO 操作:

 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

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。

其指令参数定义如下图所示：

MovCircleW A P [V] [B] [C] [Tool] [Coord] [PayLoad] WEAVE [O] [DO]

（备注：带[]参数为可选参数）

- A: 中间位置 1 (AuxPos1)
数据类型: APOS 或 CPOS 变量
参数含义: 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)
数据类型: APOS 或 CPOS 变量
参数含义: 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)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
– FINE: 无过渡。
– RELATIVE: 相对过渡。
– ABSOLUTE: 绝对过渡。
– DEFAULT: 缺省类型。
备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省, 默认值为 C100。
- [Tool]: 工具参数 (Tool)
数据类型: TOOL 变量
参数含义: 机器人执行该轨迹时使用的工具参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)
数据类型: USERCOOR 变量或变位机变量
参数含义: 机器人执行该轨迹时使用的用户坐标系/变位机坐标系。
备注: 该参数可缺省, 默认值为使用当前系统中设置的坐标系参数。
- [PayLoad]: 工件负载 (PayLoad)
数据类型: PAYLOAD 变量
参数含义: 机器人执行该轨迹时使用的工件负载参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的负载参数。

- **WEAVE**: 摆弧变量 (Weave Value)
数据类型: WEAVE 变量
参数含义: 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。
- **O**: 速率类型 (OverrideType)
数据类型: enum (GOVRON/GOVROFF)
 - GOVRON: 本段轨迹受全局速度倍率影响。
 - GOVROFF: 本段轨迹不受全局速度倍率影响。
 参数含义: 设置当前指令轨迹是否受全局速度倍率影响。
备注: 该参数可缺省, 默认值为 GOVRON。
- **[DO]**: 运动完成后执行的 IO 操作 (Add Do)
数据类型: string
参数含义: 机器人执行完该指令后, 可触发的 IO 操作:
 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

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。

其指令参数定义如下:

OnDistance TypeDistTimeMs DO

- **Type**: 触发类型 (Type)
数据类型: enum (FromBegin/FromEnd)
 - FromBegin: 从起点处触发。
 - FromEnd: 从终点处触发。
 参数含义: 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
- **Dist**: 触发距离 (TrigDis)
数据类型: real
参数含义: 相对于起点或者终点的触发距离
- **TimeMs**: 触发时间 (TrigTime)
数据类型: int
参数含义: 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)
数据类型: string
参数含义: 机器人满足触发条件后, 可触发的指令操作, 这些操作包括赋值指令, IO 指令

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

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

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



注意

- 当距离触发满足且时间触发还在延时时，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。

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。

其指令参数定义如下：

OnParameter Percence TimeMsDO

- Percence：触发类型（Type）
数据类型：int
参数含义：需要触发的路径所在占比
- TimeMs：触发时间（TrigTime）
数据类型：int
参数含义：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）
数据类型：string
参数含义：机器人满足触发条件后，可触发的指令操作，这些操作包括赋值指令，IO 指令

示例：

```
MovJ (P1, V50, "RELATIVE", C0)
OnParameter(60, 500) DO SetDO(DO0, 1);
MovL (P2, V50, "RELATIVE", C0)
在 P1 到 P2 轨迹运行到 60%处后，延时 500ms 执行 Do 操作。
```

MovArch

SCARA 机型的拱形运动指令。整个运动过程为：首先机器人在当前点抬升 h1 高度，其后再运动到距离目标点高度为 h2 处，最后再下降 h2 高度到达目标点，完成一次拱形运动。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：该指令目前只用于 scara 机型，不支持六轴机器人。

对指令参数进行说明：

MovArchP [V] [B] [C][Tool][Coord] [PayLoad]H H2Z [DO]

（备注：带[]参数为可选参数）

- P: 目标位置 (Target Pos)
数据类型: APOS 或 CPOS 变量
参数含义: 坐标系下的目标点位置
- [V]: 目标速度 (Velo)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - RELATIVE: 相对过渡。
 - ABSOLUTE: 绝对过渡。
 - DEFAULT: 缺省类型。备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE
参数含义: 机器人逼近终点时的过渡值 (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。
备注: 该参数可缺省, 默认值为 C100。
- [Tool]: 工具参数 (Tool)
数据类型: TOOL 变量
参数含义: 机器人执行该轨迹时使用的工具参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的工具参数。
- [Coord]: 坐标系参数 (Coord)
数据类型: USERCOOR 变量
参数含义: 机器人执行该轨迹时使用的用户坐标系。
备注: 该参数可缺省, 默认值为使用当前系统中设置的坐标系参数。
- [PayLoad]: 工件负载 (PayLoad)
数据类型: PAYLOAD 变量
参数含义: 机器人执行该轨迹时使用的工件负载参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
- H: 抬升高度 (Lift High)
数据类型: real
参数含义: 机器人走拱形运动时需要抬高的高度。
- H2: 下降高度 (Drop High)
数据类型: real
参数含义: 机器人走拱形运动时需要下降的高度。
- Z: 转角值 (CornerValue)
数据类型: ZONE
参数含义: 机器人走拱形运动时的转角值。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)
数据类型: string

参数含义：机器人执行完该指令后，可触发的 IO 操作：

- NULL：无任何操作。
- IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

示例：

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

```
MovArch (P2, V100, "FINE", 10, 10, C100)
```

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

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。

对指令参数进行说明：

MovH A P [V] [B] [C] [Tool] [Coord] [PayLoad] R S [DO]

（备注：带[]参数为可选参数）

- A：中间位置（AuxPos）

数据类型：APOS 或 CPOS 变量

参数含义：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）

数据类型：APOS 或 CPOS 变量

参数含义：坐标系下的目标点位置

- [V]：目标速度（Velo）

数据类型：SPEED 变量

参数含义：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）

备注：MovH 不支持过渡。

- [C]：过渡值（BlendValue）

备注：该参数不起作用。

- [Tool]：工具参数（Tool）

数据类型：TOOL 变量

参数含义：机器人执行该轨迹时使用的工具参数。

备注：该参数可缺省，默认值为使用当前系统中设置的工具参数。

- [Coord]：坐标系参数（Coord）

数据类型：USERCOORD 变量

参数含义：机器人执行该轨迹时使用的用户坐标系。

备注：该参数可缺省，默认值为使用当前系统中设置的坐标系参数。

- [PayLoad]：工件负载（PayLoad）

数据类型：PAYLOAD 变量

参数含义：机器人执行该轨迹时使用的工件负载参数。

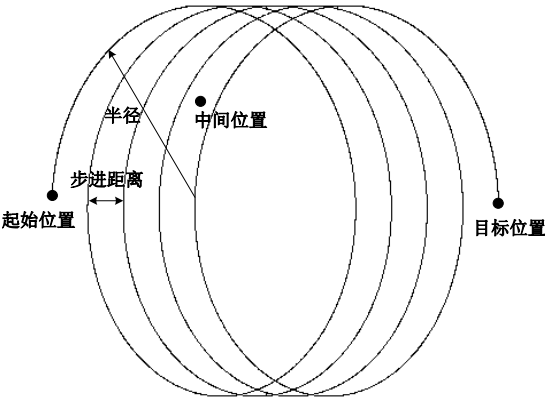
备注：该参数可缺省，默认值为使用当前系统中设置的负载参数。

- R：半径（Radius）

数据类型：real

- 参数含义：螺旋线运动时螺旋线的半径，单位 mm。
- S：步进距离（Stepping）
数据类型：real
参数含义：螺旋旋转一周后沿起终点方向移动的距离，单位 mm。
 - [DO]：运动完成后执行的 IO 操作（Add Do）
数据类型：string
参数含义：机器人执行完该指令后，可触发的 IO 操作：
 - NULL：无任何操作。
 - IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetSimDO8421、SetDIEdge、SetSimDIEdge。

图3-4 MovH 螺旋运动关键点示意



螺旋运动不支持过渡、不支持 WaitFinish

示例：

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

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

3.2 控制指令

控制指令列表：

控制指令	LABEL
	GOTO
	IF
	ELSIF
	ELSE
	WHILE
	CALL
	RUN
	KILL
	RETURN
	...=...
	/* ... */

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
```

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:”
```

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 ENDIF 做为条件判断程序块的结束。

示例：

```
IFx >1 and y == 1 or z < 1 THEN
y = 10
ENDIF
满足 IF 条件表达式，触发执行“y = 10”，否则 ENDIF 退出
```

ELSIF

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

设置 ELSIF 表达式的方法与设置 IF 表达式的方法相同。

```
IFx >1 and y == 1 or z < 1 THEN
y = 10
ELSIF x <1 THEN
y = 20
ENDIF
满足 IF 条件表达式，触发执行“y = 10”，否则继续检查，满足 ELSIF 条件表达式，触发执行“y = 20”，ENDIF 退出
```

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：

```
IFx >1 and y == 1 or z < 1 THEN
y = 10
ELSE
y = 20
ENDIF
满足 IF 条件表达式，触发执行“y = 10”，否则执行“y = 20”，ENDIF 退出
```

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 ENDWHILE 作为循环控制结束。

示例：

```
WHILE X < Y DO
MovJ(P1)
MovJ(P2)
X=X+1
ENDWHILE
满足 WHILE 条件表达式，执行内部指令行，直到 ENDWHILE 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 ENDWHILE 指令下一行执行
```

CALL

调用程序指令，当前程序跳转至同一工程下另一个子程序，子程序执行完后跳转回当前程序。

如：程序 A 中 CALL B，B 中 CALL C，C 中 CALL D，D 中 CALL E

示例：

```
CALL program1
跳转到程序 program1 内执行，执行完成后，跳转回当前程序，继续执行
```

RUN

程序并行运行指令，使得机器人能够在运行当前程序的同时，运行其它程序（程序必须在同一个工程下），即当前程序继续正常运行，启动一个线程执行另一个程序。

注：系统目前支持同时 RUN 八个并行程序（除主运行程序外），且 RUN 程序中不可以执行运动相关的指令。

示例：

```
RUN program1
并行运行程序 program1
```

KILL

停止并行运行的指令，用于结束 RUN 指令启动的程序。

示例：

```
KILL program1  
停止并行程序 program1
```

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：

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

...=...

建立一个表达式，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
```

/*...*/

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：

```
/*goto program1*/  
CALL program1
```

3.3 等待指令

等待指令列表：

等待指令	Wait
	WaitFinish
	WaitCondition

Wait

延时等待，参数可设为 int 或者 INT 变量类型，时间单位：ms。

示例：

```
Wait(1000)  
等待 1 秒
```

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", C100)  
WaitFinish (20)  
SetDO(DO1, 1)  
当 MovL 指令执行到 20%时，执行 SetDO 指令，指令参数为百分比，范围 0~100
```

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。

WaitConditionCTIETV [Goto]

- C：判别条件（Condition）
参数含义：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。
- T：时长（Time）
数据类型：int 或者 INT 变量类型，其值为正整数与 0
参数含义：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。
- IE：中断使能（Interrupt enable）
数据类型：0 或 1
参数含义：设定等待过程中是否在程序恢复后继续计时。
0：遇到程序停止时，停止计时，并在恢复时继续程序停止前的计时。
1：遇到程序停止时，停止计时，并在恢复时重新计时。
- TV：超时判断值（Timeout Value）
数据类型：INT 变量。

参数含义：若输入条件为真且该指令执行完成时，对该变量赋值 0；当该指令由于超时执行完成时，对该变量赋值 1。

- [Goto]：超时跳转的标签名（Label name）

数据类型：label

参数含义：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。

备注：该参数可缺省，默认值为不跳转。

示例：

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

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

3.4 IO 指令

IO 指令列表：

IO 指令	SetDO
	SetAO
	SetSimDO
	SetSimAO
	WaitDI
	WaitDI8421
	WaitAI
	WaitSimDI
	WaitSimDI8421
	WaitSimAI
	PulseOut
	PulseSimOut
	GetDI8421
	GetSimDI8421
	SetDO8421
	SetSimDO8421
	GetSimDIToVar
	SetSimDOByVar

	GetSimAIToVar
	SetSimAOByVar
	SetDIEdge
	SetSimDIEdge

SetDO

该指令用于将数字量输出端口设置为 TRUE（1）或者 FALSE（0）状态。

对指令参数进行说明：

SetDO DOUT VALUE

- DOUT：数字量输出变量（Target DO Port）
数据类型：DO 变量类型
参数含义：待设置的数字量输出端口变量。
- VALUE：目标值（Value）
数据类型：int
参数含义：要设置的数字量输出端口的值，可输入 0 或 1，分别代表该端口的低电平与高电平。

示例 1：

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

SetAO

该指令用于将模拟量输出端口设置为一定值。

对指令参数进行说明：

SetAOAOUT VALUE

- AOUT：模拟量输出变量（Target AO Port）
数据类型：AO 变量类型
参数含义：待设置的模拟量输出端口变量。
- VALUE：目标值（Value）
数据类型：real
参数含义：要设置的模拟量输出端口的值，可输入范围为-10.0~10.0，代表该端口的输出电平。

示例 1：

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

SetSimDO

该指令用于将虚拟数字量输出端口设置为 TRUE（1）或者 FALSE（0）状态。

SetSimDOSIMDOUT VALUE

- SIMDOUT：虚拟数字量输出变量（Target SimDO Port）
数据类型：SimDO 变量类型
参数含义：待设置的虚拟数字量输出端口变量。

- **VALUE**: 目标值 (Value)
数据类型: int
参数含义: 要设置的虚拟数字量输出端口的值, 可输入 0 或 1, 代表该端口状态。

示例 1:

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

SetSimAO

该指令用于将虚拟模拟量输出端口设置为一定值。

对指令参数进行说明:

SetSimAOSIMAOUT VALUE

- **SIMAOUT**: 虚拟模拟量输出变量 (Target SimAO Port)
数据类型: SimAO 变量类型
参数含义: 待设置的虚拟模拟量输出端口变量。
- **VALUE**: 目标值 (Value)
数据类型: real
参数含义: 要设置的虚拟模拟量输出端口的值, 可输入范围为-10.0~10.0, 代表该端口的输出值。

示例 1:

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

WaitDI

该指令用于在指定时长内等待数字量输入端口被设置 (1) 或复位 (0)。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。

对指令参数进行说明:

WaitDIDIN VALUE T INTERENABLE RET [Goto]

- **DIN**: 数字量输入变量 (Target DI Port)
数据类型: DI 变量类型
参数含义: 用于等待判断的数字量输入变量。
- **VALUE**: 目标值 (Value)
数据类型: int
参数含义: 当数字量输入变量的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。
- **T**: 时长 (Time)
数据类型: int 或者 INT 变量类型
参数含义: 等待时长, 该参数用于指定等待某 DI 信号的时长, 单位为 ms。当值为 0 时, 将强制等待直到 DI 信号满足时方认为执行成功。
- **INTERENABLE**: 中断使能 (Interrupt Enable)
数据类型: int
参数含义: 该参数用于设定在等待过程中若暂停程序, 再启动后计时是否重新开始。
 - 0: 遇到程序暂停时, 停止计时, 并在恢复运行时继续程序暂停之前的计时。
 - 1: 遇到程序暂停时, 停止计时, 并在恢复运行时重新开始计时。
- **RET**: 超时判断值 (Timeout Flag)
数据类型: INT

参数含义：返回该指令执行结束后的状态。

- 0：成功检测到信号
- 1：未检测到信号，指令超时返回。

- [Goto]：超时跳转的标签名 (Label name)

数据类型：label

参数含义：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。

备注：该参数可缺省，默认值为不跳转。

示例 1：

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

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

WaitAI

该指令用于在指定时长内等待模拟量输入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。

对指令参数进行说明：

WaitAI AIN VALUE T INTERENABLE RET [Goto]

- AIN：模拟量输入变量 (Target AI Port)

数据类型：AI 变量类型

参数含义：用于等待判断的模拟量输入变量。

- VALUE：目标值 (Value)

数据类型：real。

参数含义：当模拟量输入变量的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。

- T：时长 (Time)

数据类型：int 或者 INT 变量类型

参数含义：等待时长，该参数用于指定等待某 AI 信号的时长，单位为 ms。当值为 0 时，将强制等待直到 DI 信号满足时方认为执行成功。

- INTERENABLE：中断使能 (Interrupt Enable)

数据类型：int

参数含义：该参数用于设定在等待过程中若暂停程序，再启动后计时是否重新开始。

- 0：遇到程序暂停时，停止计时，并在恢复运行时继续程序暂停之前的计时。
- 1：遇到程序暂停时，停止计时，并在恢复运行时重新开始计时。

- RET：超时判断值 (Timeout Flag)

数据类型：INT

参数含义：返回该指令执行结束后的状态。

- 0：成功检测到信号
- 1：未检测到信号，指令超时返回。

- [Goto]：超时跳转的标签名 (Label name)

数据类型：label

参数含义：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。

备注：该参数可缺省，默认值为不跳转。

示例 1：

```
WaitAI (AI1 , 5.0 , 1000 , 1 , INT1)
```

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

WaitSimDI

该指令用于在指定时长内等待虚拟数字量输入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。

对指令参数进行说明：

WaitSimDISIMDIN VALUE T INTERENABLE RET [Goto]

- **SIMDIN**：虚拟数字量输入变量（Target SimDI Port）
数据类型：SimDI 变量类型
参数含义：用于等待判断的虚拟数字量输入变量。
- **VALUE**：目标值（Value）
数据类型：int
参数含义：当虚拟数字量输入变量的值和目标值相等时视为满足等待条件，程序继续向下执行。
- **T**：时长（Time）
数据类型：int 或者 INT 变量类型
参数含义：等待时长，该参数用于指定等待某 SimDI 信号的时长，单位为 ms。当值为 0 时，将强制等待直到 DI 信号满足时方认为执行成功。
- **INTERENABLE**：中断使能（Interrupt Enable）
数据类型：int
参数含义：该参数用于设定在等待过程中若暂停程序，再启动后计时是否重新开始。
 - 0：遇到程序暂停时，停止计时，并在恢复运行时继续程序暂停之前的计时。
 - 1：遇到程序暂停时，停止计时，并在恢复运行时重新开始计时。
- **RET**：超时判断值（Timeout Flag）
数据类型：INT
参数含义：返回该指令执行结束后的状态。
 - 0：成功检测到信号
 - 1：未检测到信号，指令超时返回。
- **[Goto]**：超时跳转的标签名（Label name）
数据类型：label
参数含义：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。
备注：该参数可缺省，默认值为不跳转。

示例 1：

```
WaitSimDI (SimDI1, 1, 1000, 0, INT1)
```

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

WaitSimAI

该指令用于在指定时长内等待虚拟模拟量输入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。

对指令参数进行说明：

WaitSimAISIMAIN VALUE T INTERENABLE RET [Goto]

- **SIMAIN**：虚拟模拟量输入变量（Target SimAI Port）

数据类型：SimAI 变量类型

参数含义：用于等待判断的虚拟模拟量输入变量。

- VALUE：目标值（Value）

数据类型：real

参数含义：当虚拟模拟量输入变量的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。

- T：时长（Time）

数据类型：int 或者 INT 变量类型

参数含义：等待时长，该参数用于指定等待某 SimDI 信号的时长，单位为 ms。当值为 0 时，将强制等待直到 DI 信号满足时方认为执行成功。

- INTERENABLE：中断使能（Interrupt Enable）

数据类型：int

参数含义：该参数用于设定在等待过程中若暂停程序，再启动后计时是否重新开始。

- 0：遇到程序暂停时，停止计时，并在恢复运行时继续程序暂停之前的计时。
- 1：遇到程序暂停时，停止计时，并在恢复运行时重新开始计时。

- RET：超时判断值（Timeout Flag）

数据类型：INT

参数含义：返回该指令执行结束后的状态。

- 0：成功检测到信号
- 1：未检测到信号，指令超时返回。

- [Goto]：超时跳转的标签名（Label name）

数据类型：label

参数含义：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。

备注：该参数可缺省，默认值为不跳转。

示例 1：

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

WaitDI8421

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

对指令参数进行说明：

WaitDI8421 BEGINPORT ENDPORT VALUE T INTERENABLE RET [Goto]

- BEGINPORT：起始 DI 端口（Begin Port）

数据类型：int

参数含义：该段连续 DI 端口的起始端口号。

- ENDPORT：结束 DI 端口（End Port）

数据类型：int

参数含义：该段连续 DI 端口的结束端口号。

- VALUE：目标值（Value）

数据类型：int

参数含义：将该段连续 DI 端口值转换成十进制数，若与 VALUE 值相等则视为满足条件。

- T：时长（Time）

数据类型：int 或者 INT 变量类型

参数含义：等待该组数字量输入 DI 信号的时长，单位 ms。当值为 0 时，将强制等待直到 DI 信号满足时方认为执行成功。

- INTERENABLE：中断使能（Interrupt Enable）

数据类型：int

参数含义：该参数用于设定在等待过程中若暂停程序，再启动后计时是否重新开始。

- 0：遇到程序暂停时，停止计时，并在恢复运行时继续程序暂停之前的计时。
- 1：遇到程序暂停时，停止计时，并在恢复运行时重新开始计时。

- RET：超时判断值（Timeout Flag）

数据类型：INT

参数含义：返回该指令执行结束后的状态。

- 0：成功检测到信号
- 1：未检测到信号，指令超时返回。

- [Goto]：超时跳转的标签名（Label name）

数据类型：label

参数含义：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。

备注：该参数可缺省，默认值为不跳转。

示例 1：

```
WaitDI8421(1000, 1, 9, 16, 255, INT1)
```

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

WaitSimDI8421

该指令用于在指定时长内等待一组连续虚拟数字量输入 SimDI 端口的状态组合。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。

对指令参数进行说明：

WaitSimDI8421 BEGINPORT ENDPORT VALUE T INTERENABLE RET [Goto]

- BEGINPORT：起始 SimDI 端口（Begin Port）

数据类型：int

参数含义：该段连续 SimDI 端口的起始端口号。

- ENDPORT：结束 SimDI 端口（End Port）

数据类型：int

参数含义：该段连续 SimDI 端口的结束端口号。

- VALUE：目标值（Value）

数据类型：int

参数含义：将该段连续 SimDI 端口值转换成十进制数，若与 VALUE 值相等则视为满足条件。

- T：时长（Time）

数据类型：int 或者 INT 变量类型

参数含义：等待该组虚拟数字量输入 SimDI 信号的时长，单位 ms。当值为 0 时，将强制等待直到 DI 信号满足时方认为执行成功。

- INTERENABLE：中断使能（Interrupt Enable）

数据类型：int

参数含义：该参数用于设定在等待过程中若暂停程序，再启动后计时是否重新开始。

- 0: 遇到程序暂停时, 停止计时, 并在恢复运行时继续程序暂停之前的计时。
- 1: 遇到程序暂停时, 停止计时, 并在恢复运行时重新开始计时。
- RET: 超时判断值 (Timeout Flag)
 - 数据类型: INT
 - 参数含义: 返回该指令执行结束后的状态。
 - 0: 成功检测到信号
 - 1: 未检测到信号, 指令超时返回。
- [Goto]: 超时跳转的标签名 (Label name)
 - 数据类型: label
 - 参数含义: 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。
 - 备注: 该参数可缺省, 默认值为不跳转。

示例 1:

```
WaitSimDI8421(1000, 0, 9, 16, 255, INT1)
```

在 1000ms 时长内等待 9 到 16 号 SimDI 口状态为 11111111 时, 若满足条件则程序继续向下执行, 若 1000ms 内未等到条件满足, 则将 INT1 置 1。若等待过程中暂停程序, 程序恢复运行后计时将继续。

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。

对指令参数进行说明:

PulseOutDOUT VALUE T INTERENABLE

- DOUT: 数字量输出变量 (Target DO Port)
 - 数据类型: DO 变量类型
 - 参数含义: 用于指定输出脉冲的数字量输出端口。
- VALUE: 目标值 (Value)
 - 数据类型: int
 - 参数含义: 在指定数字量输出端口输出脉冲的状态。
- T: 时长 (Time)
 - 数据类型: int
 - 参数含义: 脉冲信号持续时间
- INTERENABLE: 中断使能 (Interrupt Enable)
 - 数据类型: int
 - 参数含义: 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。
 - 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。
 - 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。

示例 1:

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

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

PulseSimOut

该指令用于在某一虚拟数字量输出端口输出指定时长与状态的脉冲。

对指令参数进行说明:

PulseOutSIMDOUT VALUE T INTERENABLE

- **SIMDOUT**: 虚拟数字量输出变量 (TargetSim DO Port)
数据类型: SimDO 变量类型
参数含义: 用于指定输出脉冲的虚拟数字量输出端口。
- **VALUE**: 目标值 (Value)
数据类型: int
参数含义: 在指定虚拟数字量输出端口输出脉冲的状态。
- **T**: 时长 (Time)
数据类型: int
参数含义: 脉冲信号持续时间
- **INTERENABLE**: 中断使能 (Interrupt Enable)
数据类型: int
参数含义: 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。
 - 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。
 - 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。

示例 1:

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

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

GetDI8421

该指令用于获取一段连续的数字量输入 DI 端口状态, (将其看成一段二进制数据) 并以十进制数返回。

对指令参数进行说明:

GetDI8421BEGINPORT ENDPORT RETURNVALUE

- **BEGINPORT**: 起始 DI 端口 (Begin Port)
数据类型: int
参数含义: 该段连续 DI 端口的起始端口号。
- **ENDPORT**: 结束 DI 端口 (End Port)
数据类型: int
参数含义: 该段连续 DI 端口的结束端口号。
- **RETURNVALUE**: 返回值 (Return Value)
数据类型: INT
参数含义: 将该段连续 DI 端口视作二进制数, 转化成十进制后, 返回的十进制数值。

示例 1:

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

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

GetSimDI8421

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

对指令参数进行说明:

GetSimDI8421BEGINPORT ENDPORT RETURNVALUE

- **BEGINPORT**: 起始 SimDI 端口 (Begin Port)
数据类型: int
参数含义: 该段连续 SimDI 端口的起始端口号。
- **ENDPORT**: 结束 SimDI 端口 (End Port)
数据类型: int
参数含义: 该段连续 SimDI 端口的结束端口号。
- **RETURNVALUE**: 返回值 (Return Value)
数据类型: INT
参数含义: 将该段连续 SimDI 端口视作二进制数, 转化成十进制后, 返回的十进制数值。

示例 1:

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

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

SetDO8421

该指令用于设置一段连续的数字量输出 DO 口状态 (将其看成一段二进制数据), 将传入的十进制数转换成二进制数设置到指定的连续 DO 口上。

对指令参数进行说明:

SetDO8421BEGINPORT ENDPOR T SETVALUE

- **BEGINPORT**: 起始 DO 端口 (Begin Port)
数据类型: int
参数含义: 该段连续 DO 端口的起始端口号。
- **ENDPORT**: 结束 DO 端口 (End Port)
数据类型: int
参数含义: 该段连续 DO 端口的结束端口号。
- **SETVALUE**: 设置值 (Set Value)
数据类型: int
参数含义: 将该十进制数转化为二进制数并设置到指定段连续 DO 端口上。

示例 1:

```
SetDO8421 (9, 16, 255)
```

将第 9 到第 16 号数字量输出端口置为 1。

SetSimDO8421

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

对指令参数进行说明:

SetSimDO8421BEGINPORT ENDPOR T SETVALUE

- **BEGINPORT**: 起始 SimDO 端口 (Begin Port)
数据类型: int
参数含义: 该段连续 SimDO 端口的起始端口号。
- **ENDPORT**: 结束 SimDO 端口 (End Port)
数据类型: int
参数含义: 该段连续 SimDO 端口的结束端口号。
- **SETVALUE**: 设置值 (Set Value)

数据类型：int

参数含义：将该十进制数转化为二进制数并设置到指定段连续 SimDO 端口上。

示例 1:

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

GetSimDIToVar

该指令用于将虚拟数字量输入信号映射到某个变量上。

对指令参数进行说明：

GetSimDIToVarSIMDIN VAR

- SIMDIN：虚拟数字量输入端口变量（SimDI Port）
数据类型：SimDI 变量类型
参数含义：获取该端虚拟数字量输入端口的状态，作为赋予目标变量的值。
- VAR：目标变量（Target Variable）
数据类型：BOOL
参数含义：接受获取的虚拟数字量输入端口的值。

示例 1:

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

SetSimDOByVar

该指令用于将某个布尔型变量的值映射到一个虚拟数字量输出变量上。

对指令参数进行说明：

SetSimDOByVarSIMDOUT VAR

- SIMDOUT：虚拟数字量输出口变量（SimDO Port）
数据类型：SimDO 变量类型
参数含义：将获取到的变量值在该变量绑定的 SimDO 端口输出。
- VAR：变量（Variable）
数据类型：BOOL
参数含义：获取该变量的值输出给指定虚拟数字量输出口。

示例 1:

```
SetSimDOToVar (SimDO9, BOOL1)  
将变量 BOOL1 的状态在虚拟数字量变量 SimDO9 对应端口输出。
```

GetSimAIToVar

该指令用于将虚拟模拟量输入信号映射到某个变量上。

对指令参数进行说明：

GetSimAIToVarSIMAIN VAR

- SIMAIN：虚拟模拟量输入端口变量（Sim AI Port）
数据类型：SimAI 变量类型
参数含义：获取该端虚拟模拟量输入端口的状态，作为赋予目标变量的值。
- VAR：目标变量（Target Variable）

数据类型：REAL

参数含义：接受获取的端虚拟模拟量输入端口的值。

示例 1:

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

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

SetSimAOByVar

该指令用于将某个 REAL 型变量的值映射到一个虚拟模拟量输出变量上。

对指令参数进行说明：

SetSimAOByVarSIMAOUT VAR

- SIMAOUT: 虚拟模拟量输出端口变量 (Sim AO Port)
数据类型：SimAI 变量类型
参数含义：待设置输出的虚拟模拟量输出端口。
- VAR: 变量 (Variable)
数据类型：REAL
参数含义：获取该变量的值输出给指定虚拟模拟量输出端口。

示例 1:

```
SetSimAOByVar (SimAO9, REAL1)
```

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

SetDIEdge

该指令用于强制设置某个数字量输入端口的边沿信号状态

对指令参数进行说明：

SetDIEdgeDIN EDGETYPE VALUE

- DIN: 数字量输入端口变量 (Target DI Port)
数据类型：DI 变量类型
参数含义：待强制边沿信号的数字量输入变量。
- EDGETYPE: 边沿类型 (Edge Type)
数据类型：enum
参数含义：用于选择指定端口待强制的边沿类型。
- VALUE: 强制值 (Force Value)
数据类型：int
参数含义：要在指定端口强制的边沿信号的值。

示例 1:

```
SetDIEdge (DI9, "riseEdge", 1)
```

强制 DI9 端口的上升沿信号为"1"状态。

SetSimDIEdge

通过指令强制设置某个虚拟数字量输入端口的边沿信号状态。

对指令参数进行说明：

SetSimDIEdgeSIMDIN EDGETYPE VALUE

- SIMDIN: 虚拟数字量输入端口变量 (Target SimDI Port)

数据类型：SimDI 变量类型

参数含义：待强制边沿信号的虚拟数字量输入变量。

- EDGETYPE：边沿类型（Edge Type）

数据类型：enum

参数含义：用于选择指定端口待强制的边沿类型。

- VALUE：强制值（Force Value）

数据类型：int

参数含义：要在指定端口强制的边沿信号的值。

示例 1：

```
SetSimDIEdge (SimDI9, "riseEdge", 1)
```

强制 SimDI9 端口的上升沿信号为"1"状态。

3.5 设置指令

设置指令列表:

设置指令	SetTool
	SetCoord
	SetExternalTCP
	SetPositioner
	SetJointDyn
	SetCartDyn
	SetBlendParam
	SetOverRide
	SetPayload
	SetSingularPass
	SetColliEnable
	SetAxisColliParam
	GetCurOverRide
	Hand
	CalcTool
	CalcCoord
	SetRtToErr
	SetRtInfo
	SetRtWarning
	Stop
	RefRobotAxis
	SetMotionMode
	AutoGainEnable
	AutoGainDisable

	SetRestorePC
	SetAxisVibraBLevel

SetTool

该指令用于选择工具参数。对指令参数进行说明：

SetTool Tool

- Tool：工具参数（Tool Param）
数据类型：TOOL 变量
参数含义：被选择的工具参数

示例：

```
SetTool (TOOL0)  
选择 TOOL0 为机器人的工具参数。
```

SetCoord

该指令用于选择用户坐标系。对指令参数进行说明：

SetCoordCoord

- Coord：用户坐标系参数（UserCoord Param）
数据类型：USERCOORD 变量
参数含义：被选择的用户坐标系参数

示例：

```
SetCoord (USERCOORD0)  
选择 USERCOORD0 为机器人的用户坐标系参数。
```

SetExternalTCP

该指令用于选择外部工具坐标系。对指令参数进行说明：

SetExternalTCPExtTCP

- ExtTCP：外部 TCP 参数（ExternalTCP Param）
数据类型：EXTTCP 变量
参数含义：被选择的外部 TCP 参数

示例：

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

SetPositioner

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

SetPositioner Positioner

- Positioner：变位机坐标系参数（Positioner Param）
数据类型：POSITIONER 变量
参数含义：被选择的变位机坐标系参数

示例：

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

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。对指令参数进行说明：

SetJointDynAcc Dec Jerk Torq

- Acc：加速度(%) (Acc(%))
数据类型：int
参数含义：关节最大加速度的百分比，取值范围为 1-100。
- Dec：减速度(%) (Dec(%))
数据类型：int
参数含义：关节最大减速度的百分比，取值范围为 1-100。
- Jerk：加加速度 (Jerk(%))
数据类型：int
参数含义：关节最大加加速度的百分比，取值范围为 1-100。
- Torq：力矩 (Torq(%))
数据类型：int
参数含义：关节最大力矩的百分比，取值范围为 1-100。

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。
```

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。对指令参数进行说明：

SetCartDynAcc Dec Jerk OriAcc OriDec OriJerk Torq

- Acc：加速度(%) (Acc(%))
数据类型：int
参数含义：直线及圆弧运动最大加速度的百分比，取值范围为 1-100。
- Dec：减速度(%) (Dec(%))
数据类型：int
参数含义：直线及圆弧运动最大减速度的百分比，取值范围为 1-100。
- Jerk：加加速度 (Jerk(%))
数据类型：int
参数含义：直线及圆弧运动最大加加速度的百分比，取值范围为 1-100。
- OriAcc：姿态加速度(%) (OriAcc(%))
数据类型：int
参数含义：姿态最大旋转加速度的百分比，取值范围为 1-100。

- OriDec: 姿态减速度(%) (OriDec(%))

数据类型: int

参数含义: 姿态最大旋转减速度的百分比, 取值范围为 1-100。

- OriJerk: 姿态加加速度 (OriJerk(%))

数据类型: int

参数含义: 姿态最大旋转加速度的百分比, 取值范围为 1-100。

- Torq: 力矩 (Torq(%))

数据类型: int

参数含义: 关节最大力矩的百分比, 取值范围为 1-100。

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。
```

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。对指令参数进行说明:

SetBlendParamB C

- B: 过渡类型 (BlendType)

数据类型: enum (FINE/RELATIVE/DEFAULT)

参数含义: 机器人逼近终点时的过渡方式

- FINE: 无过渡。
- RELATIVE: 相对过渡。
- ABSOLUTE: 绝对过渡。

- C: 过渡值 (BlendValue)

数据类型: ZONE

参数含义: 机器人逼近终点时的过渡值。

备注: 该参数可缺省, 默认值为 C100。

示例:

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

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。对指令参数进行说明:

SetOverridevf

- vf: 全局速度(%) (velocity factor(%))

数据类型: int 或者 INT 变量类型

参数含义: 全局速度的百分比, 取值范围为 1-100。

示例:

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

SetPayload

该指令用于设置工件负载质量。对指令参数进行说明：

SetPayloadpl

- pl: 负载质量 (Payload)

数据类型: PAYLOAD 变量

参数含义: 工件负载参数, 其中存储负载指令等一系列相关参数。

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。

SetSingularPass

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

SetSingularPassSingularPass

- SingularPass: 跨腕部奇异 (SingularPass)

数据类型: enum (OPEN/CLOSE)

参数含义: 是否过腕部奇异。

– OPEN: 开启。

– CLOSE: 关闭。

示例:

```
SingularPass("OPEN")
```

运行该指令后, 开启过腕部奇异功能。

SetColliEnable

设置碰撞检测使能指令。

SetColliEnableAxisIDValid

- AxisID: 轴号 (AxisID)

数据类型: enum (GROUP/A1/A2/A3/A4/A5/A6)

参数含义: 将要设置碰撞使能的轴号。

– GROUP: 全部轴。

– A1: 1 轴。

– A2: 2 轴。

– A3: 3 轴。

– A4: 4 轴。

– A5: 5 轴。

– A6: 6 轴。

- Valid: 有效性 (Valid)

数据类型: enum (ENABLE/DISABLE)

参数含义: 是否有效。

– ENABLE: 有效。

– DISABLE: 无效。

示例：

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

SetAxisColliParam

设置碰撞检测灵敏度值指令。

SetAxisColliParamAxisIDValue

- AxisID: 轴号 (AxisID)

数据类型: enum (A1/A2/A3/A4/A5/A6)

参数含义: 将要设置碰撞参数的轴号。

- A1: 1 轴。
- A2: 2 轴。
- A3: 3 轴。
- A4: 4 轴。
- A5: 5 轴。
- A6: 6 轴。

- Value: 灵敏度值 (Value)

数据类型: 常量

参数含义: 碰撞灵敏度。

示例：

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

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。对指令参数进行说明：

GetCurOverrideV

- V: 全局速度(%) (velocity factor(%))

数据类型: INT 变量

参数含义: 存储从系统中读取的全局速度百分比，以便其他使用。

示例：

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

Hand

该指令用于设置左右手系，仅适用于 SCARA 机型。对指令参数进行说明：

Handst

- st: 左右手设置类型 (set type)

数据类型: enum (Lefty/Righty/Default)

参数含义: 设置机器人的左右手类型方式

- Lefty: 左手设置。
- Righty: 右手设置。

– Default: 默认设置。

示例:

```
Hand ("Lefty")
```

运行该指令后，后续运动指令均以左手设置运行。

CalcTool

该指令根据提供的四个点计算标定工具参数。对指令参数进行说明：

CalcToolP1 P2 P3 P4 Tool

- P1: 标定点 1 (the first position)
数据类型: CPOS
参数含义: 第一个标定点。
- P2: 标定点 2 (the second position)
数据类型: CPOS
参数含义: 第二个标定点。
- P3: 标定点 3 (the third position)
数据类型: CPOS
参数含义: 第三个标定点。
- P4: 标定点 4 (the forth position)
数据类型: CPOS
参数含义: 第四个标定点。
- Tool: 工具参数 (Tool Param)
数据类型: TOOL
参数含义: 计算标定出的工具参数。

示例:

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

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 TOOL1 中。

CalcCoord

该指令根据提供的三个点计算标定用户坐标系参数。对指令参数进行说明：

CalcCoordP1 P2 P3 Coord

- P1: 原点 (the original position)
数据类型: CPOS
参数含义: 第一个标定点，用户坐标系原点。
- P2: RX 方向标定点 (the second position)
数据类型: CPOS
参数含义: 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)
数据类型: CPOS
参数含义: 第三个标定点，用户坐标系 RXRY 平面标定点。
- Coord: 用户参数 (UserCoord Param)
数据类型: USERCOORD
参数含义: 计算标定出的用户坐标系参数。

示例:

```
CalcCoord (P1, P2, P3, USERCOORD1)
```

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 USERCOORD1 中。

SetRtToErr

该指令用于设置机器人停止运行并发出错误信息。对指令参数进行说明：

SetRtToErr msg errId

- msg: 错误信息 (errInfo)

数据类型: str

参数含义: 设置的错误信息。

- errId: 错误编号 (errId)

数据类型: int

参数含义: 设置的错误编号，取值范围为 90001-100000。

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。
```

SetRtInfo

该指令用于设置提示信息但不影响程序运行。对指令参数进行说明：

SetRtInfo msg errId

- msg: 提示信息 (errInfo)

数据类型: str

参数含义: 设置的提示信息。

- errId: 提示编号 (errId)

数据类型: int

参数含义: 设置的提示编号，取值范围为 70001-80000。

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。
```

SetRtWarning

该指令用于设置警告信息但不影响程序的运行。对指令参数进行说明：

SetRtWarning msg errId

- msg: 警告信息 (errInfo)

数据类型: str

参数含义: 设置的警告信息。

- errId: 警告编号 (errId)

数据类型: int

参数含义: 设置的警告编号，取值范围为 80001-90000。

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。
```

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。

示例：

```
Stop ()
停止所有执行中的程序。
```

RefRobotAxis

该指令用于机器人单轴回零。对指令参数进行说明：

RefRobotAxisnum

- num：轴号（axis number）

数据类型：int

参数含义：关节轴号，取值范围为 1-16。

示例：

```
RefRobotAxis (1)
对机器人的关节 1 轴进行回零操作。
```

SetMotionMode

该指令用于设置机器人运动模式。对指令参数进行说明：

SetMotionMode MotionMode

- MotionMode：运动模式（Mode）

数据类型：enum（STANDARD/LOW_OSCILLATION）

参数含义：设置机器人的运动模式

– STANDARD：标准模式。

– LOW_OSCILLATION：低振动模式。

示例：

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

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。对指令参数进行说明：

AutoGainEnableLsScaleP LsThreshP

- LsScaleP：低速段增益比例（LsScale）

数据类型：LsScale 类型变量

参数含义：设置机器人各个关节轴的低速段增益比例阈值。

- LsThreshP：低速段速度阈值（LsThresh）

数据类型：LsThresh 类型变量

参数含义：设置机器人各个关节轴的低速段速度阈值。

示例：

```
AutoGainEnable (LsScale0, LsThresh0)
```

```
/*伺服参数自调节功能开启*/
```

AutoGainDisable

关闭伺服参数增益指令，该指令执行后系统将关闭对运动指令低速段的优化处理。

示例：

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

SetRestorePC

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

SetRestorePCindex

- index: 程序指针 (PC index)
数据类型: int
参数含义: run 程序恢复运行时的行号。

示例：

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

SetAxisVibraBLevel

振动抑制等级设置命令，只对输入整形有效。

SetAxisVibraBLevel AxisID Level

- AxisID: 轴号 (AxisID)
数据类型: enum (A1/A2/A3/A4/A5/A6)
参数含义: 将要设置振动抑制等级的轴号。
 - A1: 1 轴。
 - A2: 2 轴。
 - A3: 3 轴。
 - A4: 4 轴。
 - A5: 5 轴。
 - A6: 6 轴。
- level: 等级 (Calibration Level)
数据类型: enum (Highest/High/Default/Low/Lowest)
参数含义: 将要设置振动抑制的等级。
 - Highest: 高。
 - High: 中高。
 - Default: 标准。
 - Low: 中低。
 - Lowest: 低。

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
	UserOffset

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。对指令参数进行说明：

GetCurAPosP

- P：关节位置（joint position）
数据类型：APOS
参数含义：机器人关节坐标系下的当前位置。

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。
```

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。对指令参数进行说明：

GetCurCPosP

- P：笛卡尔空间位置（cartisian position）
数据类型：CPOS
参数含义：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。

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。
```

APosToCPos

该指令用于机器人位置点转换，根据给定的 APos 点及要转换的目标 CPos 点的用户坐标系和工具参数，可以将 APos 点转换为 CPos 点。对指令参数进行说明：

APosToCPosOP TP Tool Coord

- OP：原位置（originalposition）
数据类型：APOS
参数含义：转换前关节位置。

- TP: 目标位置 (target position)
数据类型: CPOS
参数含义: 目标笛卡尔空间位置。
- Tool: 目标工具参数 (target tool param)
数据类型: TOOL
参数含义: 目标工具参数。
- Coord: 目标坐标系 (target usercoord)
数据类型: USERCOORD、EXTTCP、POSITIONER
参数含义: 参考坐标系。

示例:

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

CPosToAPos

该指令用于机器人位置点转换指令, 给定 CPos 点及其所在的用户坐标系及工具参数, 可以将 CPos 点转换为 APos 点。对指令参数进行说明:

CPosToAPosOP Tool Coord TP

- OP: 原位置 (originalposition)
数据类型: CPOS
参数含义: 转换前的笛卡尔空间位置。
- Tool: 原工具参数 (originaltool param)
数据类型: TOOL
参数含义: 转换前的工具参数。
- Coord: 原坐标系 (originalusercoord)
数据类型: USERCOORD、EXTTCP、POSITIONER
参数含义: 转换前的参考坐标系参数。
- TP: 目标位置 (target position)
数据类型: APOS
参数含义: 转换后的关节位置。

示例:

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

CPosToCPos

该指令用于机器人位置点转换指令, 给定 CPos 点及其所在的用户坐标系及工具参数, 以及要转换的目标 CPos 点的用户坐标系及工具参数, 可以将 CPos 点转换为 CPos 点。对指令参数进行说明:

CPosToCPosOP OTool OCoord TP TTool TCoord

- OP: 原位置 (originalposition)

数据类型：CPOS

参数含义：转换前的笛卡尔空间位置。

- Tool: 原工具参数 (originaltool param)

数据类型：TOOL

参数含义：转换前的工具参数。

- OCoord: 原坐标系 (originalusercoord)

数据类型：USERCOORD、EXTTCP、POSITIONER

参数含义：转换前的参考坐标系参数。

- TP: 目标位置 (target position)

数据类型：CPOS

参数含义：目标笛卡尔空间位置。

- TTool: 目标工具参数 (targettool param)

数据类型：TOOL

参数含义：目标工具参数。

- TCoord: 目标坐标系 (targetusercoord)

数据类型：USERCOORD、EXTTCP、POSITIONER

参数含义：目标参考坐标系参数。

示例：

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

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。对指令参数进行说明：

ToolOffset F1 P F2

- F1: 基准工具参数 (Tool1)

数据类型：TOOL

参数含义：转换前的基准工具坐标系参数。

- P: 偏移量 (offset)

数据类型：DCPOS

参数含义：新工具坐标系相对于原始工具参数的偏移量。

- F2: 目标工具参数 (Tool2)

数据类型：TOOL

参数含义：基于基准工具参数偏移后获得的新工具参数。

示例：

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

UserOffset

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。对指令参数进行说明：

UserOffsetF1PF2

- F1: 用户坐标系 (Usercoord1)
数据类型: USERCOORD
参数含义: 转换前的用户坐标系参数。
- P: 偏移量 (offset)
数据类型: DCPOS
参数含义: 新用户坐标系相对于原始用户坐标系参数的偏移量。
- F2: 目标用户坐标系参数 (Usercoord2)
数据类型: USERCOORD
参数含义: 基于基准用户坐标系偏移后获得的新用户坐标系参数。

示例:

```
UserOffset (USERCOORD1, DCPOS1, USERCOORD2)  
根据工具坐标系 USERCOORD1、偏移量 DCPOS1，可以得到偏移后的用户坐标系 USERCOORD2。
```

3.7 位运算指令

位运算指令列表:

设置指令	BitAnd
	BitNeg
	BitOr
	BitXor
	BitLSH
	BitRSH

BitAnd

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

BitAndI1 I2

- I1: 操作数 1 (data1)
数据类型: INT
参数含义: 需要操作的数据，运算结果也赋值到该操作数。
- I2: 操作数 2 (data2)

数据类型：INT

参数含义：需要比较的数据。

示例：

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

BitNeg

该指令将操作数进行按位取反的运算，并将结果赋值给这个操作数，实现按位取反的运算。此外，取反操作仅操作数后 16 位。对指令参数进行说明：

BitNegI1

- I1：操作数（data）

数据类型：INT

参数含义：需要操作的数据，运算结果也赋值到该操作数。

示例：

```
BitNeg (INT1)  
将 INT1 进行按位取反的运算，并将结果赋值给 INT1。
```

BitOr

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

BitOrI1 I2

- I1：操作数 1（data1）

数据类型：INT

参数含义：需要操作的数据，运算结果也赋值到该操作数。

- I2：操作数 2（data2）

数据类型：INT

参数含义：需要比较的数据。

示例：

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

BitXOr

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

BitXOrI1 I2

- I1：操作数 1（data1）

数据类型：INT

参数含义：需要操作的数据，运算结果也赋值到该操作数。

- I2：操作数 2（data2）

数据类型：INT

参数含义：需要比较的数据。

示例：

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

BitLSH

该指令将两个操作数进行按位左移的运算，并将结果赋值给第一个操作数，实现按位左移的运算。对指令参数进行说明：

BitLSHI1 I2

- I1：操作数 1（data1）
数据类型：INT
参数含义：需要操作的数据，运算结果也赋值到该操作数。
- I2：操作数 2（data2）
数据类型：INT
参数含义：需要比较的数据。

示例：

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

BitRSH

该指令将两个操作数进行按位右移的运算，并将结果赋值给第一个操作数，实现按位右移的运算。对指令参数进行说明：

BitRSHI1 I2

- I1：操作数 1（data1）
数据类型：INT
参数含义：需要操作的数据，运算结果也赋值到该操作数。
- I2：操作数 2（data2）
数据类型：INT
参数含义：需要比较的数据。

示例：

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

3.8 时钟指令

时钟指令列表：

时钟指令	CLKStart
	CLKStop

	CLKRead
	CLKReset

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）。对指令参数进行说明：

CLKStartclk

- clk: 时钟（clock）
数据类型：CLOCK
参数含义：需要启动的时钟变量。

示例：

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

CLKStop

该指令停止指定时钟（其 state 为 false，但不会复位）。对指令参数进行说明：

CLKStopclk

- clk: 时钟（clock）
数据类型：CLOCK
参数含义：需要停止的时钟变量。

示例：

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

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。对指令参数进行说明：

CLKReadclk

- clk: 时钟（clock）
数据类型：CLOCK
参数含义：需要读取的时钟变量。

示例：

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

CLKReset

该指令用于复位指定时钟的状态、值。对指令参数进行说明：

CLKResetclk

- clk: 时钟（clock）

数据类型：CLOCK

参数含义：需要复位的时钟变量。

示例：

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

3.9 区域指令

区域指令列表：

区域指令	AreaActivate
	AreaDeactivate
	PolyhedronAreaActivate
	PolyhedronAreaDeactivate

AreaActivate

该指令用于激活标准区域，使指定的标准区域生效。对指令参数进行说明：

AreaActivateArea

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

示例：

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

AreaDeactivate

该指令用于冻结标准区域，使指定的标准区域失效。对指令参数进行说明：

AreaDeactivateArea

- Area：标准区域变量（Area）
- 数据类型：AREA
- 参数含义：需要冻结的标准区域变量。

示例：

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

PolyhedronAreaActivate

该指令用于激活多边形区域，使指定的多边形区域生效。对指令参数进行说明：

PolyhedronAreaActivate □ Polyhedron

- Polyhedron：多边形区域变量（Polyhedron）

数据类型：POLYHEDRON

参数含义：需要激活的多边体区域变量。

示例：

```
PolyhedronAreaActivate (POLYHEDRON0)  
激活多边体区域监控，使 POLYHEDRON0 对应的多边体区域生效。
```

PolyhedronAreaDeactivate

该指令用于冻结多边体区域，使指定的多边体区域失效。对指令参数进行说明：

PolyhedronAreaDeactivate □ Polyhedron

- Polyhedron: 多边体区域变量 (Polyhedron)

数据类型：POLYHEDRON

参数含义：需要冻结的多边体区域变量。

示例：

```
PolyhedronAreaDeactivate (POLYHEDRON0)  
冻结多边体区域监控，使 POLYHEDRON0 对应的多边体区域失效。
```

3.10 视觉指令

视觉指令列表:

视觉指令	TrigCam
	WaitFinishCAM
	GetCamPos
	SendMessage

TrigCam

该指令用于触发相机拍摄指令。对指令参数进行说明：

TrigCam_{tiid}

- ti: 相机触发参数 (trigger info)
数据类型: string
参数含义: 相机触发参数。
- id: 相机索引参数 (camera id)
数据类型: int
参数含义: 相机索引参数。

示例:

```
TrigCam ("OK", 1)  
字符串 "OK" 触发相机拍照。
```

WaitFinishCAM

该指令用于等待相机拍摄完成的指令。对指令参数进行说明：

WaitFinishCAM_{wtid}

- wt: 等待时间 (wait time)
数据类型: int
参数含义: 等待相机拍照完成的时间, 取值范围为 0-20000ms。
- id: 相机索引参数 (camera id)
数据类型: int
参数含义: 相机索引参数。

示例:

```
WaitFinishCAM (1000,1)  
等待相机拍照完成的时间为 1000ms。
```

GetCamPos

该指令用于获取相机位置的指令。对指令参数进行说明：

GetCamPos P tarID ret id

- P: 目标位置 (target position)
数据类型: CPOS
参数含义: 相机要获取目标位置。
- tarID: 目标位置 (targetID)
数据类型: INT
参数含义: 目标物体的 ID 值。
- ret: 有效性 (valid)
数据类型: INT
参数含义: 获取的位置点是否有效。
- id: 相机索引参数 (camera id)
数据类型: int
参数含义: 相机索引参数。

示例:

```
GetCamPos (P6, INT0, INT1, 1)  
获取相机的位置 P6, 目标物体 ID 为 INT0, 判断是否有效为 INT1。
```

SendMessage

该指令用于发送字符串消息, 可用于 3D 视觉指令, 给相机发送指定的字符串, 让相机拍照。对指令参数进行说明:

SendMessage msg id

- msg: 信息 (message info)
数据类型: string
参数含义: 发送的信息内容。
- id: 相机索引参数 (camera id)
数据类型: int
参数含义: 相机索引参数

示例:

```
SendMessage ("OK", 1)  
向相机发送字符串为 "OK"。
```

3.11 码垛指令

码垛指令列表：

码垛指令	PalletReset
	PalletToPut
	PalletFromPut
	PalletToGet
	PalletFromGet

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。对指令参数进行说明：

- PalletReset**PalletName actPart
- PalletName：码垛名（Pallet Name）
数据类型：PALLET 变量
参数含义：当前垛体信息变量
 - actPart：垛体已有工件数（ActPart）
数据类型：int
参数含义：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]。

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。

PalletToPut

该指令用于放置工件。对指令参数进行说明：

- PalletToPut**PalletName[V] [B] [C]
- （备注：带[]参数为可选参数）**
- PalletName：码垛名（Pallet Name）
数据类型：PALLET 变量
参数含义：当前垛体信息变量
 - [V]：目标速度（Velo）
数据类型：SPEED 变量
参数含义：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。
备注：该参数可缺省，默认值为 V4000。
 - [B]：过渡类型（BlendType）
数据类型：enum（FINE/RELATIVE/DEFAULT）

参数含义：机器人逼近终点时的过渡方式

- FINE：无过渡。
- RELATIVE：相对过渡。
- ABSOLUTE：绝对过渡。
- DEFAULT：缺省类型。

备注：该参数可缺省，默认值为 FINE。

- [C]：过渡值 (BlendValue)

数据类型：ZONE

参数含义：机器人逼近终点时的过渡值。

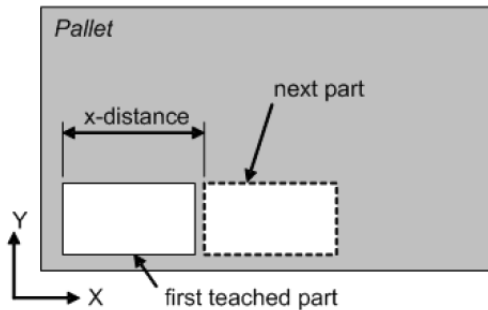
备注：该参数可缺省，默认值为 C100。

示例 1：

```
PalletToPut (PALLET1,V2000)
```

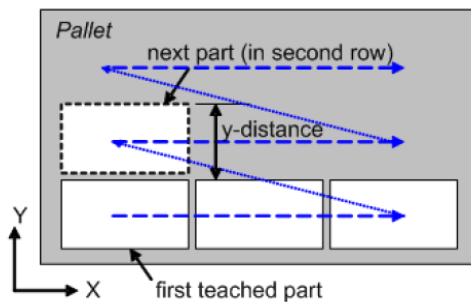
码垛功能的垛位计算取决于示教的第一个工件位置和配置的码垛的距离。在前一个工件的位置基础上添加设定的码垛距离，从而得到下一个工件的位置，如图 3-5 所示。

图3-5 计算下一个工件的位置



在图 3-6 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。

图3-6 默认码垛顺序

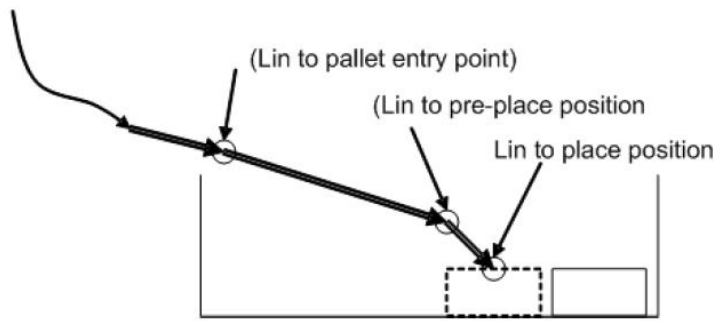


当一个新行开始放置时，在第一行工件位置的基础上添加 y 方向上的偏移即生成了新行的工件的位置数据。

这个例子里默认的码垛顺序是 x y z. 首先在 x 方向放置工件然后再进行第二行。新的一行的第一个部件从图中所示的位置开始。只要行满的，它们就会被放在下一行。在这一层的所有行都被填满后，xy 层在 z 方向重复平移，直到垛满。

进行放置时，建议路径如图 3-7 所示。

图3-7 放置路径



在调用 PalletToPut 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛的进入位置（可选）
- 机器人以直线的运动方式移动至垛点的前点位置（可选）
- 机器人以直线的运动方式移动至工件的放置位置（必选）

PalletFromPut

该指令用于放置工件完成后，机器人从放置位置离开。对指令参数进行说明：

PalletFromPutPalletName [V] [B] [C]

（备注：带[]参数为可选参数）

- PalletName: 码垛名 (Pallet Name)
数据类型: PALLET 变量
参数含义: 当前垛体信息变量
- [V]: 目标速度 (Velo)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省，默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
– FINE: 无过渡。
– RELATIVE: 相对过渡。
– ABSOLUTE: 绝对过渡。
– DEFAULT: 缺省类型。
备注: 该参数可缺省，默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省，默认值为 C100。

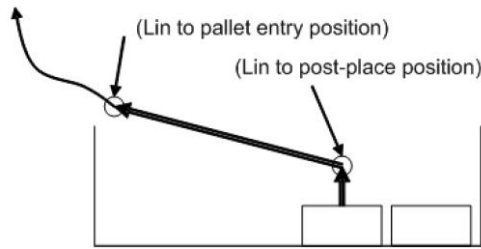
示例 1:

```
PalletFromPut (PALLET1,V2000)
```

该指令是放置工件的相反动作，从放置好的工件位置离开。

离开时，建议路径如图 3-8 所示。

图3-8 放置后离开的路径



在调用 PalletFromPut 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛点的后点位置（可选）
- 机器人以直线的运动方式移动至垛的进入位置（可选）

如果垛体的后点和进入位置都没有被设置，执行 PalletFromPut 指令时，机器人保持当前的姿态不变。

PalletToGet

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。对指令参数进行说明：

PalletToGetPalletName[V] [B] [C]

（备注：带[]参数为可选参数）

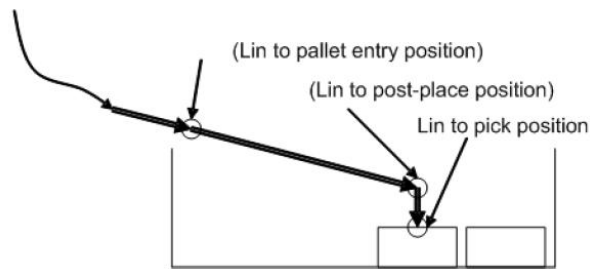
- PalletName：码垛名（Pallet Name）
数据类型：PALLET 变量
参数含义：当前垛体信息变量
- [V]：目标速度（Velo）
数据类型：SPEED 变量
参数含义：用于指定机器人执行 PalletToGet 时速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。
备注：该参数可缺省，默认值为 V4000。
- [B]：过渡类型（BlendType）
数据类型：enum（FINE/RELATIVE/DEFAULT）
参数含义：机器人逼近终点时的过渡方式
– FINE：无过渡。
– RELATIVE：相对过渡。
– ABSOLUTE：绝对过渡。
– DEFAULT：缺省类型。
备注：该参数可缺省，默认值为 FINE。
- [C]：过渡值（BlendValue）
数据类型：ZONE
参数含义：机器人逼近终点时的过渡值。
备注：该参数可缺省，默认值为 C100。

示例 1：

```
PalletToGet (PALLET1,V2000)
```

抓取过程中，路径的规划建议如图 3-9 所示。

图3-9 抓取工件路径



在调用 PalletToGet 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至栈的进入位置（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。

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。对指令参数进行说明：

PalletFromGetPalletName [V] [B] [C]

（备注：带[]参数为可选参数）

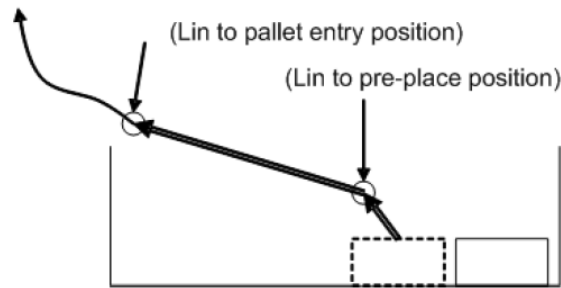
- PalletName：码垛名（Pallet Name）
数据类型：PALLET 变量
参数含义：当前垛体信息变量
- [V]：目标速度（Velo）
数据类型：SPEED 变量
参数含义：用于指定机器人执行 PalletFromGet 时速度，包括机器人末端的平移速度，旋转速度以及外部轴的移动速度等。
备注：该参数可缺省，默认值为 V4000。
- [B]：过渡类型（BlendType）
数据类型：enum（FINE/RELATIVE/DEFAULT）
参数含义：机器人逼近终点时的过渡方式
– FINE：无过渡。
– RELATIVE：相对过渡。
– ABSOLUTE：绝对过渡。
– DEFAULT：缺省类型。
备注：该参数可缺省，默认值为 FINE。
- [C]：过渡值（BlendValue）
数据类型：ZONE
参数含义：机器人逼近终点时的过渡值。
备注：该参数可缺省，默认值为 C100。

示例 1：

```
PalletFromGet (PALLET1,V2000)
```

抓取过程中，路径的规划建议如图 3-10 所示。

图3-10 抓取后离开的路径



在调用 PalletFromGet 指令时，执行了以下运动指令：

- 机器人以直线的运动方式移动至垛点的后点位置（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。

3.12 Socket 指令

Socket 指令列表：

Socket 指令	SocketCreate
	SocketClose
	SocketSendStr
	SocketSendReal
	SocketSendInt
	SocketReadReal
	SocketReadInt
	SocketReadStr
	SocketResetBuf

SocketCreate

创建 socket 客户端，以便于对服务器进行数据的交互。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。

SocketCreateSN IP P RV

- SN: socket 名称 (Socket name)
参数类型： Socket 变量
- IP: IP 地址 (IP address)
参数类型： string
参数含义： 需要连接的服务器 IP 地址。

- P: 端口号 (Port)
参数类型: int
参数含义: 需要连接的服务器端口号。
- RV: 操作返回值 (Return value)
参数类型: INT 变量
参数含义: 创建操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

示例:

```
SocketCreate (Socket0, "192.168.1.1", 8888, INT0)  
创建一个 Socket, 名称为 Socket0, 连接的服务器 IP 为 "192.168.1.1", 端口为 8888
```

SocketClose

关闭之前创建过的 socket 客户端。

SocketClose SN RV

- SN: socket 名称 (Socket name)
参数类型: Socket 变量
参数含义: SocketCreate 创建的 Socket
- RV: 操作返回值 (Return value)
参数类型: INT 变量
参数含义: 操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

示例:

```
SocketClose (Socket0, INT0)  
关闭 Socket0
```

SocketSendStr

向已经建立连接的服务器端发送字符串。

SocketSendStr SN SD RV

- SN: socket 名称 (Socket name)
参数类型: Socket 变量
参数含义: SocketCreate 创建的 Socket
- SD: 发送的数据 (Send data)
参数类型: string 常量或者 STRING 变量
- RV: 操作返回值 (Return value)
参数类型: INT 变量
参数含义: 操作的返回值变量, 返回值为 0 表示成功, 返回值为 1 则为失败。

示例:

```
SocketSendStr (Socket0, "test string", INT0)  
发送字符串 "test string" 到服务器
```

SocketSendReal

向已经建立连接的服务器端发送 real 数组。

SocketSendReal SN RA RV

- SN: socket 名称 (Socket name)

参数类型：Socket 变量

参数含义：SocketCreate 创建的 Socket

- RA：发送的 real 数组数据（Real array）

参数类型：real 数组

- RV：操作返回值（Return value）

参数类型：INT 变量

参数含义：操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

示例：

```
SocketSendReal (Socket0, RealOneArray0, INT0)
发送 real 数组 RealOneArray0 到服务器
```

SocketSendInt

向已经建立连接的服务器端发送 int 数组。

SocketSendInt SN IA RV

- SN：socket 名称（Socket name）

参数类型：Socket 变量

参数含义：SocketCreate 创建的 Socket

- IA：发送的 int 数组数据（Int array）

参数类型：int 数组

- RV：操作返回值（Return value）

参数类型：INT 变量

参数含义：操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

示例：

```
SocketSendInt (Socket0, IntOneArray0, INT0)
发送 int 数组 IntOneArray0 到服务器
```

SocketReadReal

读取服务器端发来的字符串，并以 real 数组的形式存储。

等待并接收服务器端发送的字符串，其格式为[1.1,2.2,3.3,4.4](即数据起末为'[]'，数字与数字之间用','隔开)或 1.1,2.2,3.3,4.4(数字与数字之间用','隔开)，当接收到该字符串后，机器人系统会将其拆解析并按顺序存放到数组中。

SocketReadReal SN DN RD DT RV

- SN：socket 名称（Socket name）

参数类型：Socket 变量

参数含义：SocketCreate 创建的 Socket

- DN：数据个数（Data num）

参数类型：int

- RD：实际读取的数据（Real data）

参数类型：RealOneArray 变量

- DT：检测时间（Detection time）

参数类型：int

参数含义：等待服务器端发送数据的等待时间（ms），超时报警

- RV：操作返回值（Return value）

参数类型：INT 变量

参数含义：操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

示例：

```
SocketReadReal (Socket0, 5, RealOneArray0, 100, INT0)
在 100ms 内，从服务器上读取 5 个实数，放到变量 RealOneArray0 中
```

SocketReadInt

读取服务器端发来的字符串，并以 int 数组的形式存储。

等待并接收服务器端发送的字符串，其格式为[1,2,3,4](即数据起末为'[]'，数字与数字之间用','隔开) 或 1,2,3,4(数字与数字之间用','隔开)，当接收到该字符串后，机器人系统会将其拆分解析并按顺序存放到数组中。

SocketReadInt SN DN ID DT RV

- SN: socket 名称 (Socket name)

参数类型：Socket 变量

参数含义：SocketCreate 创建的 Socket
- DN: 数据个数 (Data num)

参数类型：int
- ID: 实际读取的数据 (Int data)

参数类型：IntOneArray 变量
- DT: 检测时间 (Detection time)

参数类型：int

参数含义：等待服务器端发送数据的等待时间 (ms)，超时报警
- RV: 操作返回值 (Return value)

参数类型：INT 变量

参数含义：操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

示例：

```
SocketReadInt (Socket0, 5, IntOneArray0, 100, INT0)
在 100ms 内，从服务器上读取 5 个整数，放到变量 IntOneArray0 中
```

SocketReadStr

读取服务器端发来的字符串，并以 string 变量的形式存储。

等待并接收服务器端发送的字符串，其格式为[Hello world!](即数据起末为'[]')，当接收到该字符串后，机器人系统会将其存放到字符串变量中。

SocketReadStr SN SD DT RV

- SN: socket 名称 (Socket name)

参数类型：Socket 变量

参数含义：SocketCreate 创建的 Socket
- SD: 实际读取的数据 (String data)

参数类型：STRING 变量
- DT: 检测时间 (Detection time)

参数类型：int

参数含义：等待服务器端发送数据的等待时间 (ms)，超时报警
- RV: 操作返回值 (Return value)

参数类型：INT 变量

参数含义：操作的返回值变量，返回值为 0 表示成功，返回值为 1 则为失败。

示例：

```
SocketReadStr (Socket0, STRING0, 100, INT0)  
在 100ms 内，从服务器上读取字符串，放到变量 STRING0 中
```

SocketResetBuf

复位本地缓存的字符串，用于在执行 SocketRead 相关读取指令之前复位内存中已接收到的数据，避免出现数据读取错位的问题。

SocketReadStr SN

- SN：socket 名称 (Socket name)

参数类型：Socket 变量

参数含义：SocketCreate 创建的 Socket

示例：

```
SocketResetBuf (Socket0)  
复位 Socket0 中缓存的通讯数据。
```

3.13 软浮动指令

软浮动指令列表：

软浮动指令	SoftFloatStart
	SoftFloatStop

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。对指令参数进行说明：

SoftFloatStartType Coordinate[Tool] [User]DirectionSensitivity

（备注：带[]参数为可选参数）

- Type：软浮动类型（SoftType）
数据类型：enum （"CART"）
参数含义：软浮动类型。目前仅支持直线软浮动。
---"CART": 直线类型。
- Coordinate：参考坐标系（Coordinate）
数据类型：enum （"WORLD","TOOL","USER"）
参数含义：软浮动参考坐标系。
---"WORLD": 世界坐标系
---"TOOL": 工具坐标系
---"USER": 用户坐标系
- Tool：坐标系号（SoftTool）
数据类型：TOOL 变量
参数含义：工具参数变量。只有参考坐标系为"TOOL"时，该参数有效。
备注：当参考坐标系为"WORLD"或"USER"时，该参数无效。
- User：坐标系号（SoftUser）
数据类型：USERCOOR 变量
参数含义：用户坐标系变量。只有参考坐标系为"USER"时，该参数有效。
备注：当参考坐标系为"WORLD"或"TOOL"时，该参数无效。
- Direction：软浮动方向（SoftDirection）
数据类型：enum （"X","Y","Z"）
参数含义：软浮动方向。
---"X": 所选参考坐标系或坐标系号的 X 方向
---"Y": 所选参考坐标系或坐标系号的 Y 方向
---"Z": 所选参考坐标系或坐标系号的 Z 方向
- Sensitivity：软浮动灵敏度（SoftSensitivity）
数据类型：enum （"HIGH","MidHigh","MID","MidLow","LOW"）
参数含义：软浮动柔顺度。
---"HIGH": 柔顺等级最高
---"MidHigh": 柔顺等级次高

- "MID": 中等柔顺
- "MidLow": 柔顺等级次低
- "LOW": 柔顺等级最低

示例 1:

```
SoftFloatStart ("CART","TOOL",tool1,"Y","MID")
```

SoftFloatStop

该指令用于关闭软浮动功能。

示例 1:

```
SoftFloatStop ()
```

3.14 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
	SetSyncoord
	SyncCToUserC
	WaitConvDis
	SimConveyorOn
	SimConveyorOff
	ReceiveWObj
	ResetWObjBuf

MovLSync

MovLSync 指令为传送带同步运动指令，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，并保持与传送带的同步运动。与 MovJ 相同，在执行完该指令时可以附加进行 IO 操作。对指令参数进行说明：

MovLSyncP [V][Tool] Coord [PayLoad][SyncGoto]HeightConvyGoto[DO]

（备注：带[]参数为可选参数）

- P：目标位置（Target Pos）

数据类型：CPOS 变量

参数含义：跟随目标点，其数值是相对于目标物体识别点的偏移量，非示教所得。

- [V]：目标速度 (Velo)

数据类型：SPEED 变量

参数含义：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。

备注：该参数可缺省，默认值为 V4000。

- [Tool]：工具参数 (Tool)

数据类型：TOOL 变量

参数含义：机器人执行该轨迹时使用的工具参数。

备注：该参数可缺省，默认值为使用当前系统中设置的工具参数。

- Coord：坐标系参数 (Coord)

数据类型：SYNCOORD 变量

参数含义：机器人执行该轨迹时使用的动坐标系，该动坐标系需与传送带配置界面中启用的动坐标系一致。

- [PayLoad]：工件负载 (PayLoad)

数据类型：PAYLOAD 变量

参数含义：机器人执行该轨迹时使用的工件负载参数。

备注：该参数可缺省，默认值为使用当前系统中设置的负载参数。

- [SyncGoto]：同步过程中放弃时跳转标签 (SyncGoto)

数据类型：label

参数含义：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。

备注：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。

- Height：同步后放弃时抬升高度 (Height)

数据类型：real

参数含义：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。

- ConvyGoto：同步后放弃时跳转标签 (ConvyGoto)

数据类型：label

参数含义：用于跟随丢弃功能，当机器人在跟随运动的过程中无法继续跟随，即 MovLSync 指令后续的指令执行失败时，程序指针会转至跳转标签所在行以执行丢弃操作，在该 Label 标签下用户可根据现场工况加入相应的指令。

- [DO]：运动完成后执行的 IO 操作 (Add Do)

数据类型：string

参数含义：机器人执行完该指令后，可触发的 IO 操作：

- NULL：无任何操作。
- IO 指令：执行 IO 操作，目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

MovJSyncQuit

MovJSyncQuit 指令为退出传送带同步关节运动指令，通过该指令可以使机器人以设定的速度退出与传送带的同步运动，并使用关节运动的方式运动到目标位置，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。对指令参数进行说明：

MovJSyncQuitP [V] [B] [C][Tool] Coord [PayLoad][DO]

(备注：带[]参数为可选参数)

- P: 目标位置 (Target Pos)
 - 数据类型: APOS 或 CPOS 变量
 - 参数含义: 坐标系下的目标点位置
- [V]: 目标速度 (Velo)
 - 数据类型: SPEED 变量
 - 参数含义: 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。
 - 备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)
 - 数据类型: enum (FINE/RELATIVE/DEFAULT)
 - 参数含义: 机器人逼近终点时的过渡方式
 - FINE: 无过渡。
 - RELATIVE: 相对过渡。
 - ABSOLUTE: 绝对过渡。
 - DEFAULT: 缺省类型。
 - 备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)
 - 数据类型: ZONE 变量
 - 参数含义: 机器人逼近终点时的过渡值。
 - 备注: 该参数可缺省, 默认值为 C100。
- [Tool]: 工具参数 (Tool)
 - 数据类型: TOOL 变量
 - 参数含义: 机器人执行该轨迹时使用的工具参数。
 - 备注: 该参数可缺省, 默认值为使用当前系统中设置的工具参数。
- Coord: 坐标系参数 (Coord)
 - 数据类型: USERCOOR 变量
 - 参数含义: 机器人执行该轨迹时使用的用户坐标系。
- [PayLoad]: 工件负载 (PayLoad)
 - 数据类型: PAYLOAD 变量
 - 参数含义: 机器人执行该轨迹时使用的工件负载参数。
 - 备注: 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)
 - 数据类型: string
 - 参数含义: 机器人执行完该指令后, 可触发的 IO 操作:
 - NULL: 无任何操作。
 - IO 指令: 执行 IO 操作, 目前支持的 IO 指令有 SetDo、SetAo、SetSimDo、SetSimAo、PulseOut、PulseSimOut、SetDO8421、SetDIEdge、SetSimDIEdge。

MovLSyncQuit

MovLSyncQuit 指令为退出传送带同步直线运动指令, 通过该指令可以使机器人 TCP 点以设定的速度退出与传送带的同步运动, 并使用直线运动的方式运动到目标位置, 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。对指令参数进行说明:

MovLSyncQuit P [V] [B] [C][Tool] Coord [PayLoad][DO]**(备注：带[]参数为可选参数)**

- P: 目标位置 (Target Pos)
数据类型: APOS 或 CPOS 变量
参数含义: 坐标系下的目标点位置
- [V]: 目标速度 (Velo)
数据类型: SPEED 变量
参数含义: 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。
备注: 该参数可缺省, 默认值为 V4000。
- [B]: 过渡类型 (BlendType)
数据类型: enum (FINE/RELATIVE/DEFAULT)
参数含义: 机器人逼近终点时的过渡方式
– FINE: 无过渡。
– RELATIVE: 相对过渡。
– ABSOLUTE: 绝对过渡。
– DEFAULT: 缺省类型。
备注: 该参数可缺省, 默认值为 FINE。
- [C]: 过渡值 (BlendValue)
数据类型: ZONE 变量
参数含义: 机器人逼近终点时的过渡值。
备注: 该参数可缺省, 默认值为 C100。
- [Tool]: 工具参数 (Tool)
数据类型: TOOL 变量
参数含义: 机器人执行该轨迹时使用的工具参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的工具参数。
- Coord: 坐标系参数 (Coord)
数据类型: USERCOOR 变量
参数含义: 机器人执行该轨迹时使用的用户坐标系。
- [PayLoad]: 工件负载 (PayLoad)
数据类型: PAYLOAD 变量
参数含义: 机器人执行该轨迹时使用的工件负载参数。
备注: 该参数可缺省, 默认值为使用当前系统中设置的负载参数。
- [DO]: 运动完成后执行的 IO 操作 (Add Do)
数据类型: string
参数含义: 机器人执行完该指令后, 可触发的 IO 操作:
 - NULL: 无任何操作。

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

WaitWObj

该指令用于在指定时长内等待外部设备发来的工件在指定传送带上的位置信息。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。

对指令参数进行说明：

WaitWObj Coord T Ret [Goto]

- Coord: 坐标系参数 (Coord)

数据类型: SYNCOORD 变量

参数含义: 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。

- T: 时长 (Time)

数据类型: int

参数含义: 等待外部设备发来的工件位置信息的时长, 单位 ms。当值为 0 时, 将强制等待直到 DI 信号满足时方认为执行成功。

- Ret: 超时判断值 (Timeout Flag)

数据类型: INT

参数含义: 返回该指令执行结束后的状态。

- 0: 成功检测到信号
- 1: 未检测到信号, 指令超时返回。

- [Goto]: 超时跳转的标签名 (Value)

数据类型: label

参数含义: 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。

备注: 该参数可缺省, 默认值为不跳转。

示例 1:

```
WaitWObj (SYNCOORD0, 1000, INT1, "Goto Start")
在 1000ms 时长内等待与动坐标系 SYNCOORD0 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 INT1 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。
```

SetTargetPos

该指令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。

对指令参数进行说明:

SetTarget Coord P

- Coord: 坐标系参数 (Coord)

数据类型: SYNCOORD 变量

参数含义: 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)

数据类型: CPOS 变量

参数含义: 工件相对于 SYNCOORD 坐标系原点的偏移量。该参数中仅 xya 有效。

SetSyncoord

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

对指令参数进行说明:

SetSyncoord Coord

- Coord: 坐标系参数 (Coord)

数据类型: SYNCOORD 变量

参数含义: 前机器人所参考的坐标系参数, 仅可设置为 SYNCOORD 类型的变量。

SynCToUserC

示教参考坐标系转换指令, 用于解算示教跟随同步后轨迹点位时参考的坐标系。

对指令参数进行说明:

SynCToUserC Sync P Coord

- Sync: 动坐标系参数 (SynCoord)

数据类型: SYNCOORD 变量

参数含义: 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)

数据类型: DCPOS 变量

参数含义: 工件进入传送带的位置相对于动坐标系原点的偏移量信息。

- Coord: 坐标系参数 (Coord)

数据类型: USERCOORD 变量

参数含义: 示教跟随同步后轨迹点位时参考的用户坐标系。

WaitConvDis

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

对指令参数进行说明:

WaitConvDis Coord Dis

- Coord: 坐标系参数 (Coord)

数据类型: SYNCOORD 变量

参数含义: 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: real

参数含义: 机器人待同步运动的增量值。直线传送带时, 单位为 mm; 圆盘传送带时, 单位为 deg。

SimConveyorOn

虚拟传送带开启指令, 可用于开启虚拟传送带, 并对传送带的速度进行指定。

对指令参数进行说明:

SimConveyorOn Coord Speed

- Coord: 坐标系参数 (Coord)

数据类型: SYNCOORD 变量

参数含义: 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: real
参数含义: 虚拟传送带的运行速度。

SimConveyorOff

虚拟传送带关闭指令，可用于关闭虚拟传送带。

对指令参数进行说明：

SimConveyorOffCoord

- Coord: 坐标系参数 (Coord)
数据类型: SYNCOORD 变量
参数含义: 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。

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 run 程序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。

对指令参数进行说明：

ReceiveWObjCoordenableSts

- Coord: 坐标系参数 (Coord)
数据类型: SYNCOORD 变量
参数含义: 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。
- enableSts: 使能接收状态 (EnableSts)
数据类型: enum (OFF/ON)
参数含义: 使能机器人系统对工件位置信息的接收状态

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。

对指令参数进行说明：

ReceiveWObjCoord

- Coord: 坐标系参数 (Coord)
数据类型: SYNCOORD 变量
参数含义: 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。

3.15 阵列指令

阵列指令列表:

阵列指令	SetMatrix
------	-----------

	GetMatrix
--	-----------

SetMatrix

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。

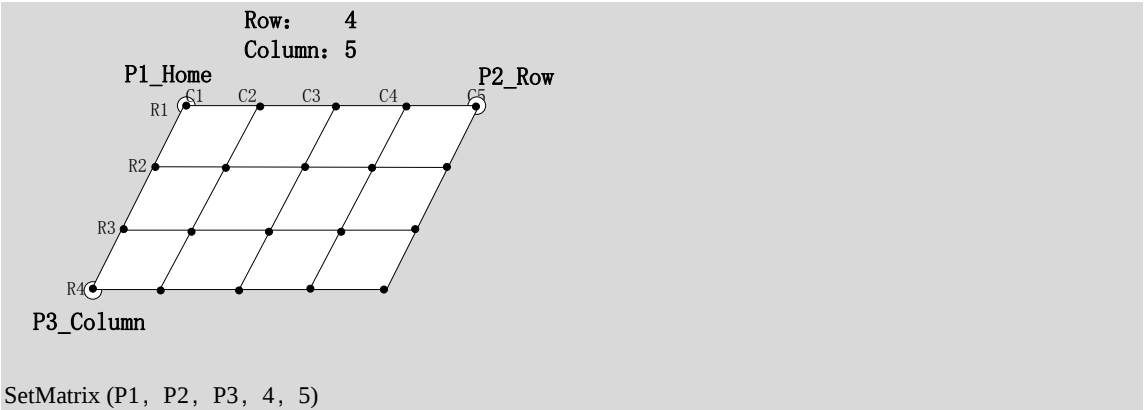
对指令参数进行说明：

SetMatrixP1 P2 P3 Row Column

- P1: 原点 (HomePos)
数据类型：CPOS 变量
参数含义：指定平行四边形的第一个点，也称为原点。
- P2: 行方向最后一个点 (RowPos)
数据类型：CPOS 变量
参数含义：指定平行四边形的行方向上的最后一个点。
- P3: 列方向最后一个点 (ColumnPos)
数据类型：CPOS 变量
参数含义：指定平行四边形的列方向上的最后一个点。
- Row: 行数 (Row)
数据类型：常量
参数含义：生成矩阵的行数。
- Column: 列数 (Column)
数据类型：常量
参数含义：生成矩阵的列数。

示例：

将 P1、P2 和 P3 形成的四边形按照 4 行 5 列均分：



GetMatrix

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

对指令参数进行说明：

GetMatrixRow Column TarPos

- Row: 行数 (Row)
数据类型：INT 型变量

参数含义：所要取点在矩阵中的行号。

- Column：列数（Column）

数据类型：INT 型变量

参数含义：所要取点在矩阵中的列号。

- TarPos：目标点（TarPos）

数据类型：CPOS 变量

参数含义：用于保存取到的点值。

3.16 数学运算函数

本节是对所能使用到的“数学函数”进行说明。

sin

正弦三角函数。

- 参数 1：int 或 real 型变量或常量。
- 函数返回值：real 型常量。

cos

余弦三角函数。

- 参数 1：int 或 real 型变量或常量。
- 函数返回值：real 型常量。

tan

正切三角函数。

- 参数 1：int 或 real 型变量或常量。
- 函数返回值：real 型常量。

asin

反正弦三角函数。

- 参数 1：int 或 real 型变量或常量。
- 函数返回值：real 型常量。

acos

反余弦三角函数。

- 参数 1：int 或 real 型变量或常量。
- 函数返回值：real 型常量。

atan

反正切三角函数。

- 参数 1：int 或 real 型变量或常量。
- 函数返回值：real 型常量。

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 real 型变量或常量。
- 参数 2: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

sinh

双曲正弦函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

cosh

双曲余弦函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

tanh

双曲正切函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

log

自然对数函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

log10

以 10 为底的对数函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

sqrt

开平方根函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

exp

以 e 为底的指数函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

pow

指数函数。

- 参数 1: int 或 real 型变量或常量, 底数。
- 参数 2: int 或 real 型变量或常量, 指数。
- 函数返回值: real 型常量。

deg

弧度转角度函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

rad

角度转弧度函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: real 型常量。

fmod

取余函数。

- 参数 1: int 或 real 型变量或常量, 被除数。
- 参数 2: int 或 real 型变量或常量, 除数。
- 函数返回值: real 型常量。

floor

向下取整函数。

- 参数 1: int 或 real 型变量或常量。
- 函数返回值: int 型常量。

random

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 型常量。

3.17 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
	sub
	trimLeft
	trimRight
	IToStr
	upper
	RToStr
	StrToI
	StrToR
	APosToStr
	CPosToStr
	TranStrToInt

	TranStrToReal
	TranStrToApos
	TranStrToCpos

。

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。

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。

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)

findEnd

字符串反向查找指令，在字符串中找到指定字符串最后出现的位置，并返回索引号。

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

format

格式化字符串指令，通过在参数 1 中传入合理的格式化控制符，其后带上任意多个参数来填充这个格式化控制符，并将填充后的数据返回。

- 参数 1: 字符串格式，string 型变量或常量。
- 参数 2: 待填充格式化控制符的参数，string/real/int 型变量或常量。
- 参数 3: 待填充格式化控制符的参数，string/real/int 型变量或常量。
- ... (参数无限制，总长度只要不超过单个指令字符串长度即可)
- 函数返回值: 成功分割并保存到数组中的个数，int 型变量

格式化字符串以%开头，支持以下用法:

- %c - 接受一个数字,并将其转化为 ASCII 码表中对应的字符
- %d,%i - 接受一个数字并将其转化为有符号的整数格式
- %o - 接受一个数字并将其转化为八进制数格式

%u - 接受一个数字并将其转化为无符号整数格式
 %x - 接受一个数字并将其转化为十六进制数格式,使用小写字母 x
 %X - 接受一个数字并将其转化为十六进制数格式,使用大写字母 X
 %f - 接受一个数字并将其转化为浮点数格式
 %s - 接受一个字符串并按照给定的参数格式化该字符串

样例:

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

getAt

单个字符串获取指令，获取某一位的字符串数据，并将获取后的数据返回。

- 参数 1: 待截取的字符串，string 型变量或常量。
- 参数 2: 待获取的位置，int 型变量或常量。
- 函数返回值: 获取后的字符串，string 型变量。

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。

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。

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。

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。

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。

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。

strcmp

字符串比较指令，返回值为第一个不同字符之间的 ASCII 码差值。

- 参数 1：参与比较的字符串数据，string 型变量或常量。
- 参数 2：被比较的字符串数据，string 型变量或常量。
- 函数返回值：返回的 ASCII 码值，int 型变量。

trimLeft

字符串左修剪指令，去掉字符串左边的空格，并将修改过后的字符串数据返回。

- 参数 1：待修剪的字符串，string 型变量或常量。
- 函数返回值：修剪后的字符串，string 型变量。

trimRight

字符串右修剪指令，去掉字符串右边的空格，并将修改过后的字符串数据返回。

- 参数 1：待修剪的字符串，string 型变量或常量。
- 函数返回值：修剪后的字符串，string 型变量。

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。

IToStr

整型数据转字符串指令，将整型数据转换成字符串类型数据，并返回转换后的字符串。

- 参数 1：待转换的整型数据，int 型变量或常量。

- 函数返回值：转换后的字符串数据，string 型变量。

RToStr

实数型参数转字符串指令，将实数数据转换成字符串类型数据，并返回转换后的字符串。

- 参数 1：待转换的实数型数据，REAL 型变量或常量。
- 函数返回值：转换后的字符串数据，string 型变量。

StrToI

字符串转整型指令，将字符串数据转换成整数类型数据，并返回转换后的整型数据。

- 参数 1：待转换的字符串数据，string 型变量或常量。
- 函数返回值：转换后的整型数据，int 型变量。

StrToR

字符串转实数型数据指令，将字符串数据转换成实数类型数据，并返回转换后的实数。

- 参数 1：待转换的字符串数据，string 型变量或常量。
- 函数返回值：转换后的实数型数据，REAL 型变量。

APosToStr

APOS 点数据转成字符串指令，将 apos 点按照指定的格式转换成字符串，并返回转换后的字符串数据。转换后 apos 点格式为 “0; 1,2,3,4,5,6,0,0,0,0,0,0,0;”，其中分号前面分别代表 apos 点类型、坐标值。

- 参数 1：待转换的 APOS 点数据，APOS 型变量。
- 函数返回值：转换后的字符串数据，string 型变量。

CPosToStr

CPOS 点数据转成字符串指令，将 cpos 点按照指定的格式转换成字符串，并返回转换后的字符串数据。转换后 cpos 点格式为 “1;0,0,0,0,0;1,2,3,4,5,6,0,0,0,0,0,0,0;”，分号前面分别代表 cpos 点类型、cfg 参数、坐标值。

- 参数 1：待转换的 CPOS 点数据，CPOS 型变量。
- 函数返回值：转换后的字符串数据，string 型变量

TranStrToInt

整数型字符串数组获取指令，根据传入的分割符将字符串分割，并将分割后的值转换到整型一维数组中，同时返回有效分割数量。

- 参数 1：待分割的字符串，string 型变量或常量。
- 参数 2：分割符，string 型变量或常量。
- 参数 3：以数组的形式存储分割后的整型变量，IntOneArray 型变量。
- 函数返回值：成功分割并保存到数组中的个数，int 型变量

样例：

```
P:INT0.value = TranStrToInt("1_2_3er", "_", G:IntOneArray0) // 该指令传入的字符为"1_2_3er"，执行后系统按照传入的分割符'_'来分割提取数据，分割后的数据分别按照 1、2、0 的顺序存在一维整型数组变量 IntOneArray0 中，并将分割个数 3 赋给 INT0.value。
```

TranStrToReal

实数型字符串数组获取指令，根据传入的分割符将字符串分割，并将分割后的值转换到实数型一维数组中，同时返回有效分割数量。

- 参数 1：待分割的字符串，string 型变量或常量。
- 参数 2：分割符，string 型变量或常量。
- 参数 3：以数组的形式存储分割后的实数型变量，RealOneArray 型变量。
- 函数返回值：成功分割并保存到数组中的个数，int 型变量

样例：

```
P:INT0.value = TranStrToReal("1.1_2.2_3er", "_", G:INTOneArray0) // 该指令传入的字符为"1.1_2.2_3er"，
执行后系统按照传入的分割符'_'来分割提取数据，分割后的数据分别按照 1.1、2.2、0 的顺序存在一维整
型数组变量 RealOneArray0 中，并将分割个数 3 赋给 INT0.value。
```

TranStrToApos

apos 点数据字符串数组获取指令，根据传入的分割符将字符串分割，并将分割后的值转换到 apos 点变量中，同时返回转换成功与否的状态。需要转换的字符串格式必须为 “0; 1,2,3,4,5,6,0,0,0,0,0,0,0,0,0;”，其中分号前面分别代表 apos 点类型、坐标值，最后一个分号后面可以带任意字符串。

- 参数 1：待分割的字符串，string 型变量或常量。
- 参数 2：分割符，string 型变量或常量。
- 参数 3：以 apos 变量的形式存储分割后的 apos 点数据，APOS 型变量。
- 函数返回值：返回提取成功与否的状态，int 型变量

样例：

```
P:INT0.value = TranStrToApos("0; 21,22,23,24,25,26,0,0,0,0,0,0,0,0,0;$qwqewe", "$", G: P0) // 该指令传入
的字符为"0; 21,22,23,24,25,26,0,0,0,0,0,0,0,0,0;$qwqewe"，执行后系统按照传入的分割符'$'来分割提取数
据，'$'后面的数据会被舍弃。分割后的数据分别按照 21、22、23、24、25、26...的顺序存在 P0 的 a1、a2、
a3、a4、a5、a6...中，并将成功返回值 0 赋给 INT0.value。
```

TranStrToCpos

cpos 点数据字符串数组获取指令，根据传入的分割符将字符串分割，并将分割后的值转换到 apos 点变量中，同时返回转换成功与否的状态。需要转换的字符串格式必须为 “1;0,0,0,0,0,0;1,2,3,4,5,6,0,0,0,0,0,0,0,0;”，分号前面分别代表 cpos 点类型、cfg 参数、坐标值。

- 参数 1：待分割的字符串，string 型变量或常量。
- 参数 2：分割符，string 型变量或常量。
- 参数 3：以 cpos 变量的形式存储分割后的 cpos 点数据，CPOS 型变量。
- 函数返回值：返回提取成功与否的状态，int 型变量

样例：

```
P:INT0.value = TranStrToCpos("1;0,0,0,0,0,0;21,22,23,24,25,26,0,0,0,0,0,0,0,0,0;$qwqewe", "$", G: P0) // 该
指令传入的字符为"0; 0,0,0,0,0,0;21,22,23,24,25,26,0,0,0,0,0,0,0,0,0;$qwqewe"，执行后系统按照传入的分
割符'$'来分割提取数据，'$'后面的数据会被舍弃。分割后第一个";"后面的"0,0,0,0,0,0"会按顺序存入 cfg 参
数中，第二个";"后面的数据按照 21、22、23、24、25、26...的顺序存在 P0 的 x、y、z、a、b、c...中，并
将成功返回值 0 赋给 INT0.value。
```

3.18 ModbusTcp 指令

ModbusTcp 指令列表：

ModbusTcp 指令	GetModConState
--------------	----------------

	ReadModbusReg
	WriteModbusReg

GetModConState

该指令用于获取机器人与外界使用 modbus 通讯的连接状态。对指令参数进行说明：

GetModConState IsConnected

- IsConnected：连接状态（Is Connected）
数据类型：BOOL 变量
参数含义：返回当前的连接状态。

示例 1：

```
GetModConState (BOOL0)
```

ReadModbusReg

该指令用于读取指定 Modbus 寄存器的值。对指令参数进行说明：

ReadModbusRegRegisterID RegisterValue

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

示例 1：

```
ReadModbusReg (101, INT0)    // 读取 101 寄存器的值
```

WriteModbusReg

该指令用于设置指定 Modbus 寄存器的值。对指令参数进行说明：

WriteModbusReg RegisterIDRegisterValue

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

示例 1：

```
INT0.value = 100  
WriteModbusReg (101, INT0)    // 将 101 寄存器的值改为 100
```



南京埃斯顿机器人工程有限公司

地址：南京江宁经济技术开发区水阁路16号 邮编：211106

电话：025-52785866

传真：025-52785576

公司主页：www.estun.com

电子信箱：info@estun.com



www.estun.com

ESTUN