

# 机器人控制系统指 令集

**Ver21.05**



## 目录

机器人控制系统指令集 .....	8
> 运动控制类 .....	8
MOVJ-点到点 .....	8
MOVL-直线 .....	9
MOVC-圆弧 .....	10
MOVCA-整圆 .....	12
MOVS-曲线插补 .....	14
IMOV-增量 .....	16
MOVJEXT-外部轴点到点 .....	18
MOVLEXT-外部轴直线 .....	19
MOVCEXT-外部轴圆弧 .....	21
SPEED-全局速度 .....	22
SAMOV-定点移动 .....	23
MOVJDOUBLE-双机点到点 .....	24
MOVLDOUBLE-双机直线 .....	25
MOVCDOUBLE-双机圆弧 .....	26
MOVCADOUBLE-双机整圆 .....	27
MOVCOMM-外部点 .....	28
EXTMOV-外部轴随动 .....	29

GEARIN-电子齿轮 .....	30
MRESET-复位外部轴多圈转动量 .....	31
DRAG_TRAJECTORY-拖拽示教 .....	31
➤ 输入输出类 .....	32
DIN-IO 输入 .....	32
DOUT-IO 输出 .....	33
AIN-模拟输入 .....	34
AOUT-模拟输出 .....	35
PULSEOUT-脉冲输出 .....	35
READ_DOUT-读取输出 .....	36
➤ 定时器类 .....	37
TIMER-延时 .....	37
➤ 运算类 .....	37
ADD-加 .....	37
SUB-减 .....	38
MUL-乘 .....	39
DIV-除 .....	40
MOD-模 .....	40
SIN-正弦 .....	41
COS-余弦 .....	42

ATAN-反正切 .....	43
LOGICAL_OP-逻辑运算 .....	44
▶ 条件控制类 .....	45
CALL-调用子程序 .....	45
IF-如果 .....	45
ELSEIF-否则如果 .....	48
ELSE-否则 .....	50
WAIT-等待 .....	51
WHILE-循环 .....	53
LABEL-标签 .....	56
JUMP-跳转 .....	57
UNTIL-直到 .....	59
CRAFTLINE-工艺跳行 .....	62
CMDNOTE-注释指令 .....	62
POS_REACHABLE-到达判断 .....	63
CLKSTART-计时开始 .....	63
CLKSTOP-计时停止 .....	64
CLKRESET-计时复位 .....	64
READLINEAR-读取线速度 .....	65
▶ 变量类 .....	65

INT-整型 .....	65
DOUBLE-浮点型 .....	66
BOOL-布尔型 .....	67
SETINT-赋值整型 .....	68
SETDOUBLE-赋值浮点型 .....	69
SETBOOL-赋值布尔型 .....	70
FORCESET-写入文件 .....	71
➤ 坐标切换类 .....	71
SWITCHTOOL-切换工具手 .....	71
SWITCHUSER-切换用户坐标 .....	72
USERCOORD_TRANS-用户坐标转换 .....	72
SWITCHSYNC-切换外部轴 .....	73
➤ 网络通讯类 .....	73
SENDMSG-发送数据 .....	73
PARSEMSG-解析数据 .....	74
READCOMM-读取 .....	75
OPENMSG-打开数据 .....	75
CLOSEMSG-关闭数据 .....	76
PRINTMSG-输出信息 .....	76
MSG_CONNECTION_STATUS-获取信息连接状态	77

➤ 位置变量类 .....	77
USERFRAME_SET-用户坐标修改 .....	77
TOOLFRAME_SET-工具坐标修改 .....	78
READPOS-读取点位 .....	79
POSADD-点位加 .....	80
POSSUB-点位减 .....	81
POSSET-点位改 .....	82
COPYPOS-复制点位 .....	83
POSADDALL-点位全加 .....	84
POSSUBALL-点位全减 .....	85
POSSETALL-点位全改 .....	86
TOFFSETON-轨迹偏移开始 .....	87
TOFFSETOFF-轨迹偏移结束 .....	88
READPOSMSG-读取点位信息 .....	88
➤ 程序控制类 .....	88
PTHREAD_START-开启线程 .....	89
PTHREAD_END-退出线程 .....	89
PAUSERUN-暂停运行 .....	89
CONTINUERUN-继续运行 .....	90
STOPRUN-停止运行 .....	91

RESTARTRUN-重新运行 .....91

# 机器人控制系统指令集

## > 运动控制类

### MOVJ-点到点

#### 功能

使用关节插补的方式移动到目标点。在机器人向目标点移动中，在不受轨迹约束的区间使用。

#### 参数说明

- P/G: 使用局部位置变量 (P) 或全局位置变量 (G)。当值为“新建”时，插入该指令则新建一个 P 变量，并将机器人的当前位置记录到该 P 变量。

- VJ: 关节插补的速度，范围 1-100，单位为百分比。实际运动速度为机器人关节参数中轴最大速度乘以该百分比。

- PL: 平滑过渡等级，范围 0-5。

- ACC: 加速度比率，范围 0-100，单位为百分比。建议设置为与 VJ 值相同。

- DEC: 减速度比率，范围 0-100，单位为百分比。建议设置为与 VJ 值相同。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

MOVJ P001 VJ = 10 % PL = 1 ACC = 10 DEC = 10 0

MOVJ G002 VJ = 10 % PL = 0 ACC = 10 DEC = 10 0

## MOVL-直线

### 功能

使用直线插补的方式移动到目标点。在机器人向目标点移动的过程中, 机器人末端运动的轨迹为直线。

### 参数说明

- P/G: 使用局部位置变量 (P) 或全局位置变量 (G)。当值为“新建”时, 插入该指令则新建一个 P 变量, 并将机器人的当前位置记录到该 P 变量。

- V: 运动速度, 范围 2-9999, 单位为 mm/s。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

• TIME: 时间, 范围非负整数, 单位 ms。提前时间  
执行下一条指令。

## 使用范例

MOVL P003 V = 200 mm/s PL = 2 ACC = 20 DEC = 20 0

## MOVC-圆弧

### 功能

机器人通过圆弧插补示教的 3 个点画圆移动。

若用圆弧插补示教机器人轴, 移动命令是 MOVC。

单一圆弧和连续圆弧的第一个圆弧的起始点只能为 MOVJ  
或 MOVL。

#### ◦单一圆弧

当圆弧只有一个时, 如下表所示, 用圆弧插补示教  
P1-P3 的 3 个点。

若用关节插补或直线插补示教进入圆弧前的 P0, 则  
P0-P1 的轨迹自动成为直线。

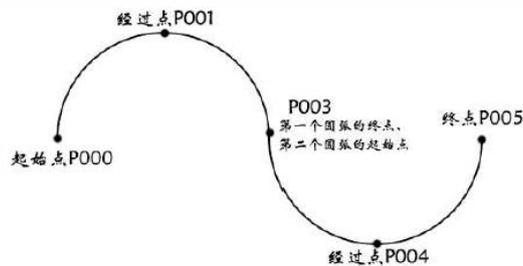


P000-关节/直线

## P001-P002-圆弧

### ◦连续圆弧

如下表所示，当曲率发生改变的圆弧连续有 2 个以上时，圆弧最终将逐个分离。因此，如图 4 所示，请在前一个圆弧与后一个圆弧的连接点加入关节及直线插补的点。



## P000-关节/直线

## P001-P002-圆弧

### 参数说明

- P/G: 使用局部位置变量 (P) 或全局位置变量 (G)。当值为“新建”时，插入该指令则新建一个 P 变量，并将机器人的当前位置记录到该 P 变量。

- V: 运动速度，范围 2-9999，单位为 mm/s。

- PL: 平滑过渡等级，范围 0-5。

- ACC: 加速度比率，范围 0-100，单位为百分比。

建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

MOVJ P001 VJ = 10 % PL = 0 ACC = 10 DEC = 10 0

MOVC P002 V = 100 mm/s PL = 0 ACC = 10 DEC = 10 0

MOVC P003 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0

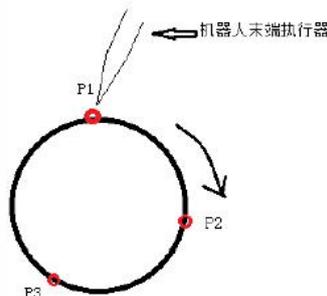
## MOVCA-整圆

### 功能

通过示教圆的起始点和两个经过点, 机器人走一个完整的圆。

### 指令插入前提

点击上方状态栏中的“工具”按钮, 选中之前标定好的工具手;



插入步骤，共四条指令。

1. 点击插入，点击坐标切换类，选择 SWITCHTOOL，选择之前标定好的工具手号
2. 移动到所要画的圆的任意一个点如图 P1，点击插入，点击运动控制类，选择 movj 或者 movl；
3. 再移动到所要画的圆的任意一个点如图 P2（要不同于第 2 步中的点），点击上方状态栏中的“坐标系”按钮，选中“工具”坐标系，点击插入，点击运动控制类，选择 movca

再移动到所要画的圆的任意一个点如图 P3（要不同于 2,3 步中的点），点击上方状态栏中的“坐标系”按钮，选中“工具”坐标系，点击插入，点击运动控制类，选择 movca

## 参数说明

- P/G：使用局部位置变量 (P) 或全局位置变量 (G)。当值为“新建”时，插入该指令则新建一个 P 变量，并将机器人的当前位置记录到该 P 变量。

- V：运动速度，范围 2-9999，单位为 mm/s。

- PL：平滑过渡等级，范围 0-5。

- ACC：加速度比率，范围 0-100，单位为百分比。

建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V \times 10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间  
执行下一条指令。

## 使用范例

MOVJ P001 VJ = 10 % PL = 0 ACC = 10 DEC = 10SPIN=1 0

MOVCA P002 V = 100 mm/s PL = 0 ACC = 10 DEC =

10SPIN=1 0

MOVCA P003 V = 100mm/s PL = 0 ACC = 10 DEC =

10SPIN=1 0

## MOVS-曲线插补

### 功能

在焊接、切割、熔接、涂底漆等作业时, 若使用自由曲线插补, 对于不规则曲线工件的示教作业可变得容易。

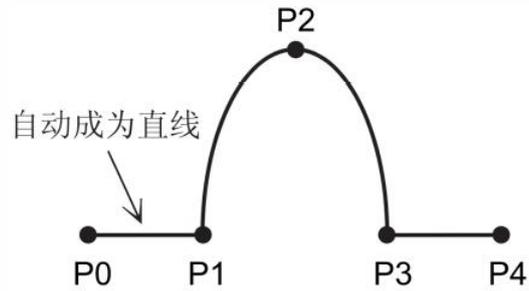
轨迹为通过三个点的抛物线。

若使用自由曲线插补示教机器人轴, 则移动命令为 MOVS。

#### ◦ 单一自由曲线

如下表所示, 用自由曲线插补示教 P1-P3 的 4 个点。

若使用关节插补或直线插补示教进入自由曲线前的 P0 点, 那么 P0-P1 的轨迹自动成为直线。



P0-关节/直线

P2-P5-曲线插补

### ◦连续自由曲线

用重合抛物线合成建立轨迹。

与圆弧插补不同，2个自由曲线的连接处不能是同一点或不能有其它指令。



P0-关节/直线

P1-P5-曲线插补

P6-关节/直线

重合抛物线的情况下建立合成轨迹。

## 参数说明

- P/G: 使用局部位置变量 (P) 或全局位置变量 (G)。当值为“新建”时, 插入该指令则新建一个 P 变量, 并将机器人的当前位置记录到该 P 变量。

- V: 运动速度, 范围 2-9999, 单位为 mm/s。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。建议设置为  $V*10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

MOVJ P001 VJ = 10 % PL = 0 ACC = 10 DEC = 10 0

MOVS P002 V = 100 mm/s PL = 0 ACC = 10 DEC = 10 0

MOVS P003 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0

MOVS P004 V = 100 mm/s PL = 0 ACC = 10 DEC = 10 0

MOVS P005 V = 100mm/s PL = 0 ACC = 10 DEC = 10 0

## IMOV-增量

## 功能

以关节或直线的插补方式从当前位置按照设定的增量距离移动。

## 参数说明

- B: 增量变量, 可选择关节、直角、工具、用户四种坐标系, 对应轴填正数为正方向, 负数为反方向。若不动则填 0。

- V/VJ:

- 当 B 为关节坐标系下的值时, 该处为 VJ, 关节插补的速度, 范围 1-100, 单位为百分比。

- 实际运动速度为机器人关节参数中轴最大速度乘以该百分比。当 B 为直角、工具、用户坐标系下的值时, 该处为 V, 运动速度, 范围 2-9999, 单位为 mm/s。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。  
建议设置为  $V*10\%$ 或 VJ。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。  
建议设置为  $V*10\%$ 或 VJ。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

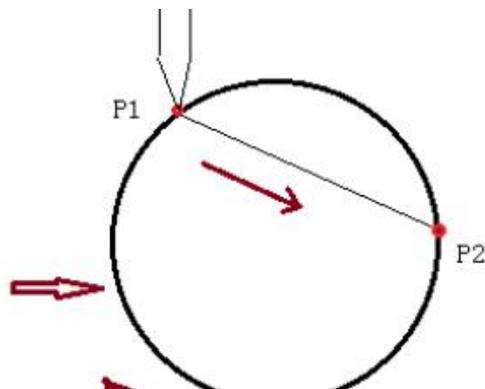
```
IMOV S001 VJ = 10 % RF PL = 0 ACC = 10 DEC = 10 0
```

```
IMOV S002 V = 1000 mm/s BF PL = 1 ACC = 100 DEC = 100  
0
```

## MOVJEXT-外部轴点到点

### 功能

机器人以关节插补方式向示教位置移动, 外部轴用用关节差补运动。



### 参数说明

- E: 同时记录机器人与外部轴位置数据的变量。当值为“新建”时, 插入该指令则新建一个 E 变量, 并将机器人与外部轴的当前位置记录到该 E 变量。

- VJ: 关节插补的速度, 范围 1-100, 单位为百分比。实际运动速度为机器人关节参数中轴最大速度乘以该百分比。外部轴速度随机器人速度改变。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。建议设置为与 VJ 值相同。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。建议设置为与 VJ 值相同。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

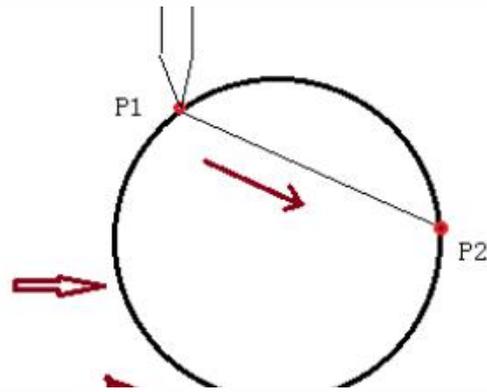
## 使用范例

```
MOVJEXTE001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0
```

## MOVLEXT-外部轴直线

### 功能、

机器人以直线插补的方式向示教位置移动, 外部轴用关节差补的方式运动。



## 参数说明

- E: 同时记录机器人与外部轴位置数据的变量。当值为“新建”时，插入该指令则新建一个 E 变量，并将机器人与外部轴的当前位置记录到该 E 变量。

- V: 机器人运动速度，范围 2-9999，单位为 mm/s。外部轴速度随机器人速度改变。

- PL: 平滑过渡等级，范围 0-5。

- SYNC: 机器人与外部轴是否同步运动，当选是时，机器人与外部轴协作走直线。当选否时，机器人在空间中走直线，外部轴独立运动到目标角度。

- ACC: 加速度比率，范围 0-100，单位为百分比。建议设置为  $V*10\%$ 。

- DEC: 减速度比率，范围 0-100，单位为百分比。建议设置为  $V*10\%$ 。

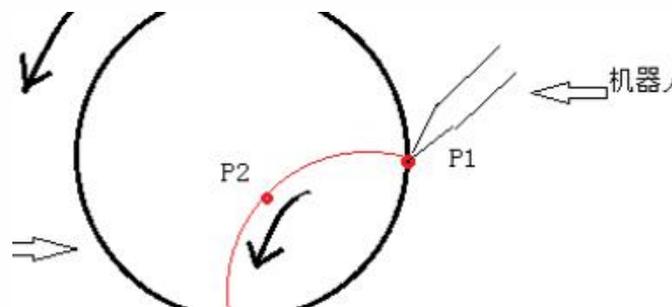
## 使用范例

MOVLEXT E002 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC  
= 0

## MOVCEXT-外部轴圆弧

### 功能

机器人以圆弧插补方式向示教位置移动,外部轴用用关节差补运动。



### 参数说明

- E: 同时记录机器人与外部轴位置数据的变量。当值为“新建”时,插入该指令则新建一个 E 变量,并将机器人与外部轴的当前位置记录到该 E 变量。
- V: 机器人运动速度,范围 2-9999,单位为 mm/s。外部轴速度随机器人速度改变。
- PL: 平滑过渡等级,范围 0-5。
- SYNC: 机器人与外部轴是否同步运动,当选是时,机器人与外部轴协作走圆弧。当选否时,机器人在空间中走圆弧,外部轴独立运动到目标角度。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间  
执行下一条指令。

## 使用范例

```
MOVLEXT E002 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC  
= 1 0
```

```
MOVCEXT E003 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC  
= 1 0
```

```
MOVCEXT E004 V = 10 mm/s PL = 0 ACC = 1 DEC = 1 SYNC  
= 1 0
```

## SPEED-全局速度

### 功能

SPEED 指令以下的所有运动类指令的运动类指令的运动速度为: 指令速度\*上方状态栏的速度\*SPEED 的百分比。

### 参数说明

- 全局速度 (%) : 速度百分比: 1-200。

## 使用范例

SPEED= 9 %

## SAMOV-定点移动

### 功能

机器人以关节插补运动到一个设定好的绝对位置

如果你不希望移动某个轴，请在该轴的坐标处留空。（不要填 0）

### 参数说明

- B：绝对位置，可选择关节、直角、工具、用户四种坐标系，若对应轴不填写则对应轴不动。

- V/VJ：

- 当 B 为关节坐标系下的值时，该处为 VJ，关节插补的速度，范围 1-100，单位为百分比。

- 实际运动速度为机器人关节参数中轴最大速度乘以该百分比。当 B 为直角、工具、用户坐标系下的值时，该处为 V，运动速度，范围 2-9999，单位为 mm/s。

- PL：平滑过渡等级，范围 0-5。

- ACC：加速度比率，范围 0-100，单位为百分比。

建议设置为  $V*10\%$ 或 VJ。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 或 VJ。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

```
SAMOVR001 VJ= 10 %RFPL= 2 ACC= 10 DEC= 10 0
```

## MOVJDOUBLE-双机点到点

### 功能

当设置为两台机器人时, 令两台机器人同时走关节插补运动到目标位置。同时启停。

### 参数说明

- E: 同时记录两台机器人位置数据的变量。当值为“新建”时, 插入该指令则新建一个 E 变量, 并将两台机器人的当前位置记录到该 E 变量。

- VJ: 关节插补的速度, 范围 1-100, 单位为百分比。实际运动速度为机器人关节参数中轴最大速度乘以该百分比。两台机器人的速度同步。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。建议设置为与 VJ 值相同。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为与 VJ 值相同。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

```
MOVJDOUBLE E001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0
```

## MOVLDDOUBLE-双机直线

### 功能

当设置为两台机器人时, 令两台机器人同时走直线插补运动到目标位置。同时启停。

### 参数说明

- E: 同时记录两台机器人位置数据的变量。当值为“新建”时, 插入该指令则新建一个 E 变量, 并将两台机器人的当前位置记录到该 E 变量。

- V: 机器人运动速度, 范围 2-9999, 单位为 mm/s。  
两台机器人速度同步。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间  
执行下一条指令。

## 使用范例

```
MOVLDDOUBLE E001 V = 100 mm/s PL = 0 ACC= 10 DEC =  
10 0
```

## MOVCDDOUBLE-双机圆弧

### 功能

当设置为两台机器人时, 令两台机器人同时走圆弧插补运动到目标位置。同时启停。

### 参数说明

- E: 同时记录两台机器人位置数据的变量。当值为“新建”时, 插入该指令则新建一个 E 变量, 并将两台机器人的当前位置记录到该 E 变量。

- V: 机器人运动速度, 范围 2-9999, 单位为 mm/s。  
两台机器人速度同步。

- PL: 平滑过渡等级, 范围 0-5。

- ACC: 加速度比率, 范围 0-100, 单位为百分比。  
建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V \times 10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间执行下一条指令。

## 使用范例

```
MOVLDOUBLE E001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0
```

```
MOVCDouble E002V = 100 mm/s PL = 0 ACC= 10 DEC =  
10 0
```

```
MOVCDouble E003V = 100mm/s PL = 0 ACC= 10 DEC = 10  
0
```

## MOVCADouble-双机整圆

### 功能

当设置为两台机器人时, 令两台机器人同时走整圆插补运动到目标位置。同时启停。

### 参数说明

- E: 同时记录两台机器人位置数据的变量。当值为“新建”时, 插入该指令则新建一个 E 变量, 并将两台机器人的当前位置记录到该 E 变量。

- V: 机器人运动速度, 范围 2-9999, 单位为 mm/s。两台机器人速度同步。

- PL: 平滑过渡等级, 范围 0-5。
- ACC: 加速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- DEC: 减速度比率, 范围 0-100, 单位为百分比。

建议设置为  $V*10\%$ 。

- TIME: 时间, 范围非负整数, 单位 ms。提前时间  
执行下一条指令。

## 使用范例

```
MOVLDDOUBLE E001 VJ = 10 % PL = 0 ACC= 10 DEC = 10 0
```

```
MOVCAODOUBLE E002 V = 100 mm/s PL = 0 ACC= 10 DEC =  
10 0
```

```
MOVCAODOUBLE E003 V = 100 mm/s PL = 0 ACC= 10 DEC =  
10 0
```

## MOVCOMM-外部点

### 功能

以规定的插补方式运动到外部设备通过 Modbus 发给控制器的点位。

### 参数说明

- 插补方式: 运动到目标点所使用的插补方式, 包含  
关节、直线、曲线。

- V/VJ:

- 当 B 为关节坐标系下的值时，该处为 VJ，关节插补的速度，范围 1-100，单位为百分比。

- 实际运动速度为机器人关节参数中轴最大速度乘以该百分比。当 B 为直角、工具、用户坐标系下的值时，该处为 V，运动速度，范围 2-9999，单位为 mm/s。

- PL: 平滑过渡等级，范围 0-5。

- ACC: 加速度比率，范围 0-100，单位为百分比。建议设置为  $V*10\%$  或 VJ。

- DEC: 减速度比率，范围 0-100，单位为百分比。建议设置为  $V*10\%$  或 VJ。

- TIME: 时间，范围非负整数，单位 ms。提前时间执行下一条指令。

## 使用范例

MOVCOMMMovLV= 10 mm/s PL = 0 ACC = 1 DEC = 1 0

## EXTMOV-外部轴随动

### 功能

外部轴按机器人线速度倍数的速度或恒速跟随机器人随动的指令。

### 参数说明

- 外部轴：可选 O1、O2 某个轴进行随动
- 类型：
  - 随动类型：随机器人实时线速度改变速度
  - 恒速类型：为恒定按某个速度运行

- 速度值来源：可选

INT/DOUBLE/GINT/GDOUBLE/手填

- 变量名：速度值来源为

INT/DOUBLE/GINT/GDOUBLE 时，用于选择哪个变量

- 手填值：速度值来源为手填时，用于输入恒定运行的速度值

## 使用范例

```
EXTMOV O1 FOLLOW 22.22
```

## GEARIN-电子齿轮

## 功能

让外部轴某轴随机器人某轴一起运动的指令。

## 参数说明

- 主轴：可选机器人的 J1~J6 轴

- 外部轴：可选 O1、O2 某个轴进行随动（O3 暂不支持）

- 比例关系 K：随动轴速（°/s）=K\*主轴速（°/s）

## 使用范例

GEARIN J1 O1 22.22

## MRESET-复位外部轴多圈转动量

### 功能

让外部轴某轴随机器人某轴一起运动的指令。

### 参数说明

- MRESET：可选全部轴、单个轴

## 使用范例

MRESET 0

## DRAG\_TRAJECTORY-拖拽示教

### 功能

机器人按记录轨迹来运动的指令。

### 参数说明

- 轨迹名：机器人轨迹的名称
- 回放速率：运动速度（0~500%）

## 使用范例

DRAG\_TRAJECTORY Track1 20%

## > 输入输出类

### DIN-IO 输入

#### 功能

读取 IO 板的数字输入值，并存储到一个整型或布尔型变量中。

#### 参数说明

- 变量类型：将输入值存到目标变量的变量类型。
- 变量名：将输入值存到目标变量的变量名。
- 输入 IO 板：若有多个 EtherCAT IO，则可选择是第几个 IO 板。
- 输入组号：输入是按照组来读的，分别为 1 路、4 路、8 路一组。1 路一组则 16 个 DIN 端口为 16 组；4 路一组则 1-4、5-8、9-12、13-16 各为一组；8 路一组则 1-8、9-16 各为一组。读入变量的数据是将输入的端口值由 2 进制转为 10 进制存入变量中。

例：8路一组，1-8号端口的值为10110101，那么从8号端口开始则为10101101。将其转为10进制则为173，则存入变量为173。

## 使用范例

DIN I001 IN#(5)

## DOUT-IO 输出

## 功能

将IO板上对应的IO端口置高或置低。

## 参数说明

- 输出IO板：若有多个EtherCAT IO，则可选择是第几个IO板。

- 输出组号：输出是按照组来输出的，分别为1路、4路、8路一组。1路一组则16个DOUT端口为16组；4路一组则1-4、5-8、9-12、13-16各为一组；8路一组则1-8、9-16各为一组。

- 变量来源：分为手动选择和变量类型。手动选择就是在下面的框中打钩，选中的输出1，未选中的输出0。  
例：当输出组号为4路输出，第2组时，下面的选择框中端口1、端口3选中，其它两个留空，那么运行该指令时，IO板的输出端口中5-8号端口的输出值为1010。

当变量来源选择 INT、GINT、BOOL、GBOOL 时，会将对应变量值转换为 2 进制，输出到 IO 板上。

例：若变量值为 173，则其转换为二进制则为 10101101。若 8 路一组，将二进制值从 8 号端口开始输出，那么 8-1 号端口值为 10101101，1-8 号端口的值为 10110101。

- 变量名：变量来源选择 INT、GINT、BOOL、GBOOL 时，这里选择要输出的变量名。

- 时间：置反输出时间，输出在规定时间内置反。例如 DOUT1=1、时间为 2，则 DOUT1 输出高电平 2 秒后置反为低电平

## 使用范例

DOUTOT#(1) I001 0

## AIN-模拟输入

### 功能

将对应模拟输入口的输入值读入到目标变量中。

### 参数说明

- 模拟输入口：要读取的模拟输入口。
- 变量值来源：目标变量的变量类型。
- 变量名：目标变量的变量名。

## 使用范例

AIN D001 B001

## AOUT-模拟输出

### 功能

将对应模拟输出口的输出值置为定义的值。

### 参数说明

- 模拟输出口：要输出的端口。
- 变量值来源：要输出的值的变量类型。
- 新参数：当变量值选择自定义时，在这里输入手填数据，范围 0-10V，对应端口则会输出该值。
- 变量名：要输出值的变量的变量名。

## 使用范例

AOUTAOUT1 1.1

## PULSEOUT-脉冲输出

### 功能

按照设定的脉冲频率与个数，在 R1 PWMIO 板上的 DB9 端子的引脚 4（PWM+）上进行输出。

### 参数说明

- 个数：脉冲个数。
- 频率：脉冲频率。

## 使用范例

PULSEOUT RATE = 100 SUM = 100

## READ\_DOUT-读取输出

### 功能

读数数字输出端口的输出状态，并存入目标变量中。

### 参数说明

- 变量类型：要存入的目标变量的变量类型。
- 变量名：要存入的目标变量的变量名。
- 输出组号：读取输出端口的值是按照组来读取的。

分别为 1 路、4 路、8 路一组。1 路一组则 16 个 DOUT 端口为 16 组；4 路一组则 1-4、5-8、9-12、13-16 各为一组；8 路一组则 1-8、9-16 各为一组。

例：8 路一组，1-8 号端口的值为 10110101，那么从 8 号端口开始则为 10101101。将其转为 10 进制则为 173，则存入变量为 173。

## 使用范例

READ\_DOUT I001 OT#(1)

## > 定时器类

### TIMER-延时

#### 功能

延时设置的值，然后继续运行。

#### 参数说明

- 变量值来源：若选择自定义，则可以在新参数手填值。若选择其它变量，则会延时变量值对应的时间长度。
- 新参数：变量值选择自定义时在这里手填值，(0~9999]，单位 s。
- 来源参数：当变量值来源选择变量时，在这里选择对应的变量名。

#### 使用范例

TIMER T= 10

## > 运算类

### ADD-加

#### 功能

加法运算 (+) ，  $A=A+B$ 。

#### 参数说明

- 变量类型：被加数 A 的变量类型。
- 变量名：被加数 A 的变量名。
- 变量值来源：加数 B 的变量类型，可以选择自定义填写或其他变量类型。
- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。
- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

ADD GI001 22; 含义：GI001=GI001+22

ADD GI002 I003; 含义：GI002=GI002+I003

## SUB-减

### 功能

减法运算 (-) ,  $A=A-B$ 。

### 参数说明

- 变量类型：被减数 A 的变量类型。
- 变量名：被减数 A 的变量名。
- 变量值来源：减数 B 的变量类型，可以选择自定义填写或其他变量类型。

- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。

- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

SUB GI001 22; 含义：GI001=GI001-22

SUB GI002 I003; 含义：GI002=GI002-I003

## MUL-乘

### 功能

乘法运算 (\*),  $A=A*B$ 。

### 参数说明

- 变量类型：被乘数 A 的变量类型。
- 变量名：被乘数 A 的变量名。
- 变量值来源：乘数 B 的变量类型，可以选择自定义填写或其他变量类型。

- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。

- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

MUL GI001 22; 含义:  $GI001=GI001*22$

MUL GI002 I003; 含义:  $GI002=GI002*I003$

## DIV-除

### 功能

除法运算 ( $\div$ ) ,  $A=A\div B$ 。

### 参数说明

- 变量类型: 被除数 A 的变量类型。
- 变量名: 被除数 A 的变量名。
- 变量值来源: 除数 B 的变量类型, 可以选择自定义填写或其他变量类型。
- 新参数: 当变量值来源选择自定义时, 本输入框有效, 所填值为 B 的值。
- 来源参数: 当变量值来源选择变量时, 这里为 B 的变量名。

## 使用范例

DIV GI001 22; 含义:  $GI001=GI001\div 22$

DIV GI002 I003; 含义:  $GI002=GI002\div I003$

## MOD-模

## 功能

取模运算 (Mod) ,  $A=A \text{ Mod } B$ 。

## 参数说明

- 变量类型：被除数 A 的变量类型。
- 变量名：被除数 A 的变量名。
- 变量值来源：除数 B 的变量类型，可以选择自定义填写或其他变量类型。
- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。
- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

MOD GI001 22; 含义：GI001=GI001 Mod 22

MOD GI002 I003; 含义：GI002=GI002 Mod I003

## SIN-正弦

## 功能

正弦运算 (sin) ,  $A=\sin(B)$  , B 为弧度制 rad。

## 参数说明

- 变量类型：结果值 A 的变量类型。

- 变量名：结果值 A 的变量名。
- 变量值来源：正弦弧度值 B 的变量类型，可以选择自定义填写或其他变量类型。
- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。
- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

SIN GI001 22; 含义：GI001=sin(22)

SIN GI002 I003; 含义：GI002=sin(I003)

## COS-余弦

### 功能

余弦运算 (cos) ,  $A=\cos(B)$ , B 为弧度制 rad。

### 参数说明

- 变量类型：结果值 A 的变量类型。
- 变量名：结果值 A 的变量名。
- 变量值来源：余弦弧度值 B 的变量类型，可以选择自定义填写或其他变量类型。

- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。

- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

COS GI001 22; 含义：GI001=cos(22)

COS GI002 I003; 含义：GI002=cos(I003)

## ATAN-反正切

### 功能

反正切运算 (arctan) ,  $A=\arctan(B)$  , B 为弧度制 rad.

### 参数说明

- 变量类型：结果值 A 的变量类型。
- 变量名：结果值 A 的变量名。
- 变量值来源：反正切弧度值 B 的变量类型，可以选择自定义填写或其他变量类型。

- 新参数：当变量值来源选择自定义时，本输入框有效，所填值为 B 的值。

- 来源参数：当变量值来源选择变量时，这里为 B 的变量名。

## 使用范例

ATAN GI001 22; 含义: GI001=arctan(22)

ATAN GI002 I003; 含义: GI002=arctan(I003)

## LOGICAL\_OP-逻辑运算

### 功能

逻辑运算 (与或非) , B001=I001andI002。

### 参数说明

- 参数 1 类型: 参与运算的参数 1 的变量类型。
- 参数 1 名: 参与运算的参数 1 的变量名。
- 运算类型: 逻辑与(&&), 逻辑或(||), 逻辑非(!)。
- 参数 2 类型: 参与运算的参数 2 的变量类型。
- 参数 2 名: 参与运算的参数 2 的变量名。
- 结果存入变量类型: 运算结果存入的变量类型。
- 结果存入变量名: 运算结果存入的变量名

### 使用范例

LOGICAL\_OPB001 = I001AND 10; 含义: 变量 I001、常数  
10 逻辑与运算结果存入 B001

## > 条件控制类

### CALL-调用子程序

#### 功能

调用另一个程序，被调用程序运行完后则返回原程序 CALL 指令的下一行继续运行。

#### 参数说明

- CALL：被调用程序名称。

#### 使用范例

CALL [Program]：含义：调用程序 Program

### IF-如果

#### 功能

如果 IF 指令的条件满足时，则执行 IF 与 ENDIF 之间的指令，如果 IF 指令的条件不满足，则直接跳转到 ENDIF 指令继续运行 ENDIF 下面的指令，不运行 IF 与 ENDIF 之间的指令。

IF 的判断条件为（比较数 1 比较方式 比较数 2），例如比较数 1 为 2，比较数 2 为 1，比较方式为">"，则  $2 > 1$ ，判断条件成立；若比较方式为"<"或"=="，则判断条件不成立。

IF 指令可以单独使用，也可搭配 ELSEIF、ELSE 两条指令使用。注意，ELSEIF、ELSE 指令不可脱离 IF 指令单独使用！

注意，当程序的开头为 IF 且最后一行为 ENDIF 指令时，请在 IF 指令上方或 ENDIF 下方插入一条 0.1 秒的 TIMER（延时）指令，否则当 IF 指令的条件不满足时会导致程序陷入死机状态。

插入 IF 指令时会同时插入 ENDIF 指令，当删除 IF 指令时请注意将对应的 ENDIF 指令也删掉，否则会导致程序无法执行。

IF 指令中可以嵌套另一个 IF 指令或 WHILE、JUMP 等其它条件判断类指令。

## 参数说明

- 参数类型：比较数 1 的类型，变量或数字、模拟量的输入值。

- 参数名：

若参数类型选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若参数类型选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

- 比较方式：

== 等于

<小于

>大于

<= 小于或等于

>= 大于或等于

!= 不等于

- 变量值来源：比较数 2 的类型，自定义或变量或数字、模拟量的输入值。

- 新参数：

若变量值来源选择的类型为自定义，则此处不可选

若变量值来源选择的类型为变量 (INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL)，则此处为比较数 1 的变量名

若变量值来源选择的类型为输入值 (DIN、AIN)，则此处为数字输入或模拟输入的端口号

- 来源参数：若变量值来源处选择的为自定义，则在此处直接填写比较数 2 的值。

## 使用范例

```
IF(GI001>=D001)
```

其它指令，如 MOVJ 等

```
ENDIF
```

## ELSEIF-否则如果

### 功能

ELSEIF 指令必须插入在 IF 和 ENDIF 之间。ELSEIF 与 ENDIF 之间还可以插入一条 ELSE 指令或多条 ELSEIF 指令。

当 IF 的条件满足时，会忽略掉 ELSEIF 和 ELSEIF 与 ENDIF 之间的指令，仅运行 IF 与 ELSEIF 之间的指令，然后跳转到 ENDIF 下面的一行指令继续运行。

当 IF 的条件不满足时，会跳转到 ELSEIF 指令，判断 ELSEIF 的判断条件，若满足，则运行 ELSEIF 和 ENDIF 之间的指令，然后继续运行 ENDIF 下面的指令；若不满足，则直接跳转到 ENDIF 下面的一行指令继续运行。

若在 IF 与 ENDIF 中嵌套了多条 ELSEIF，当 IF 的判断条件不成立时首先判断第一条 ELSEIF 的判断条件，若成立则运行第一条 ELSEIF 与第二条 ELSEIF 之间的指令；若不成立则判断第二条 ELSEIF 的判断条件，以此类推。

注意，当删除 IF 指令时，需删除与其对应的 ELSEIF 和 ENDIF 指令，否则会导致程序无法运行。

### 参数说明

- 参数类型：比较数 1 的类型，变量或数字、模拟量的输入值。

- 参数名：

若参数类型选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若参数类型选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

- 比较方式：

== 等于

< 小于

> 大于

<= 小于或等于

>= 大于或等于

!= 不等于

- 变量值来源：比较数 2 的类型，自定义或变量或数字、模拟量的输入值。

- 新参数：

若变量值来源选择的类型为自定义，则此处不可选

若变量值来源选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若变量值来源选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

• 来源参数：若变量值来源处选择的为自定义，则在此处直接填写比较数 2 的值。

## 使用范例

```
IF(GI001>=D001)
```

```
    其它指令 1, 如 MOVJ 等
```

```
ELSEIF(D001<9)
```

```
    其它指令 2, 如 MOVJ 等
```

```
ENDIF
```

## ELSE-否则

### 功能

ELSE 指令必须插入在 IF 和 ENDIF 之间,但是一个 IF 指令只能嵌入一条 ELSE 指令。

当 IF 的判断条件成立时, 会运行 IF 与 ELSE 之间的指令后跳转到 ENDIF 的下一行指令继续运行, 而不运行 ELSE 和 ENDIF 之间的指令。

当 IF 的判断条件不成立时, 会跳转到 ELSE 与 ENDIF 之间的指令运行, 而不运行 IF 与 ELSE 之间的指令。

注意，当删除 IF 指令时，需删除与其对应的 ELSE 和 ENDIF 指令，否则会导致程序无法运行。

### 参数说明

无

### 使用范例

IF(GI001<9)

其它指令 1, 如 MOVJ 等

ELSE

其它指令 2, 如 MOVJ 等

ENDIF

### WAIT-等待

#### 功能

WAIT 即等待,可以选择是否有等待时间。当没有勾选“TIME”选项,则在判断条件不成立时一直停留在该 WAIT 指令等待,直到判断条件成立。若勾选了“TIME”选项,则会在等待该参数的时长后不再等,继续运行下一条指令。若在等待时条件变为成立,则立刻运行下一条指令。

### 参数说明

- 参数类型：比较数 1 的类型，变量或数字、模拟量的输入值。

- 参数名：

若参数类型选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若参数类型选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

- 比较方式：

== 等于

< 小于

> 大于

<= 小于或等于

>= 大于或等于

!= 不等于

- 变量值来源：比较数 2 的类型，自定义或变量或数字、模拟量的输入值。

- 新参数：

若变量值来源选择的类型为自定义，则此处不可选

若变量值来源选择的类型为变量 (INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL) ，则此处为比较数 1 的变量名

若变量值来源选择的类型为输入值 (DIN、AIN) ，则此处为数字输入或模拟输入的端口号

- 来源参数：若变量值来源处选择的为自定义，则在此处直接填写比较数 2 的值

- TIME：

可选项，不选则永远等待直到条件成立

选择则可填写等待时间（秒），等待到该时长后，即使条件依然不成立，依然会跳转到下一行继续运行。

## 使用范例

```
WAIT (GI001==2) T = 2
```

## WHILE-循环

### 功能

当 WHILE 指令的条件满足时，会循环运行 WHILE 与 ENDWHILE 两条指令之间的指令。在运行到 WHILE 指令之前若判断条件不满足，在运行到 WHILE 指令时会直接跳转到 ENDWHILE 指令而不运行 WHILE 与 ENDWHILE 之间的指令；若在运行 WHILE 与 ENDWHILE 之间的指令过程中，判

断条件变成不满足，会继续运行，直到运行到 ENDWHILE 行，不再循环而是继续运行 ENDWHILE 下面的指令。

WHILE 的判断条件为（比较数 1 比较方式 比较数 2），例如比较数 1 为 2，比较数 2 为 1，比较方式为">"，则  $2 > 1$ ，判断条件成立；若比较方式为"<"或"=="，则判断条件不成立。

注意，插入 WHILE 指令的同时会同时插入 ENDWHILE 指令。若要删除 WHILE 指令请同时删掉其对应的 ENDWHILE 指令，否则会导致程序无法运行。

当程序的开头为 WHILE 且最后一行指令为 ENDWHILE 时，请在程序的开头或结尾插入一条 0.3 秒的 TIMER（延时）指令。否则当 WHILE 指令的条件不满足时会导致程序陷入死机。

当 WHILE 内部的指令没有运动类指令或在某种情况下可能会陷入死循环时，请在 WHILE 与 ENDWHILE 间插入一条 0.3 秒的 TIMER（延时）指令，否则当 WHILE 指令的条件满足时可能会导致程序陷入死机。

WHILE 指令可以同时嵌套多个 WHILE、IF 或 JUMP 等其它判断类指令使用。

## 参数说明

- 参数类型：比较数 1 的类型，变量或数字、模拟量的输入值。

- 参数名：

若参数类型选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若参数类型选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

- 比较方式：

== 等于

< 小于

> 大于

<= 小于或等于

>= 大于或等于

!= 不等于

- 变量值来源：比较数 2 的类型，自定义或变量或数字、模拟量的输入值。

- 新参数：

若变量值来源选择的类型为自定义，则此处不可选

若变量值来源选择的类型为变量 (INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL) ，则此处为比较数 1 的变量名

若变量值来源选择的类型为输入值 (DIN、AIN) ，则此处为数字输入或模拟输入的端口号

• 来源参数：若变量值来源处选择的为自定义，则在此处直接填写比较数 2 的值。

## 使用范例

```
WHILE(GI001<2)
```

```
    其它指令 1, MOVJ 等
```

```
    WHILE(D001<10)
```

```
    其它指令 2, MOVJ 等
```

```
    ADD D001 1
```

```
    ENDWHILE
```

```
    其它指令 3
```

```
    ADD GI001 1
```

```
ENDWHILE
```

LABEL-标签

功能

JUMP 指令跳转的目标标签。

## 参数说明

- 标签名：标签的名字，需使用字母开头的字符串。

## 使用范例

```
LABEL *A1
```

## JUMP-跳转

### 功能

JUMP 用于跳转，必须与 LABEL（标签）指令配合使用。

JUMP 可以设置有无判断条件。当设置为没有判断条件时，运行到该指令会直接跳转到对应的 LABEL 指令后继续运行 LABEL 下一行指令。

当设置为有判断条件时，若条件满足则跳转到 LABEL 指令行；若条件不满足则忽略 JUMP 指令，继续运行 JUMP 指令的下一行指令。

LABEL 标签可以插在 JUMP 的上方或者下方，但不可跨程序跳转。

LABEL 标签名必须为字母开头的两位以上字符。

插入 LABEL 标签对程序的运行没有影响，但是要符合程序运行规则，例如不能插在 MOVC 指令的上面或插在局部变量定义指令的上面。

## 参数说明

- 标签名：已插入 LABEL 指令的标签名，选项。

- 判断条件：

选项，若选中则可以设置判断条件。

若不选中则运行到 JUMP 后直接跳转。

- 参数类型：比较数 1 的类型，变量或数字、模拟量的输入值。

- 参数名：

若参数类型选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若参数类型选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

- 比较方式：

== 等于

< 小于

> 大于

<= 小于或等于

>= 大于或等于

!= 不等于

- 变量值来源：比较数 2 的类型，自定义或变量或数字、模拟量的输入值。

- 新参数：

若变量值来源选择的类型为自定义，则此处不可选

若变量值来源选择的类型为变量 (INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL)，则此处为比较数 1 的变量名

若变量值来源选择的类型为输入值 (DIN、AIN)，则此处为数字输入或模拟输入的端口号

- 来源参数：若变量值来源处选择的为自定义，则在此处直接填写比较数 2 的值。

## 使用范例

MOVJ

LABEL \*C1

其它指令 1, MOVJ 等

JUMP \*C1 WHEN (I001==0)

其它指令 2

UNTIL-直到

功能

UNTIL 指令用于在一个运动过程中跳出。即在机器人的一个运动过程中暂停并开始下一个过程。当条件满足时，不论当前机器人是否运行，立即暂停并开始 ENDUNTIL 指令下面的一条指令。

UNTIL 的判断条件为（比较数 1 比较方式 比较数 2），例如比较数 1 为 2，比较数 2 为 1，比较方式为">"，则  $2 > 1$ ，判断条件成立；若比较方式为"<"或"=="，则判断条件不成立。

注意，插入 UNTIL 指令的同时会同时插入 ENDUNTIL 指令。若要删除 UNTIL 指令请同时删掉其对应的 ENDUNTIL 指令，否则会导致程序无法运行。

### 参数说明

- 参数类型：比较数 1 的类型，变量或数字、模拟量的输入值。

- 参数名：

若参数类型选择的类型为变量（INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL），则此处为比较数 1 的变量名

若参数类型选择的类型为输入值（DIN、AIN），则此处为数字输入或模拟输入的端口号

- 比较方式：

== 等于

<小于

>大于

<= 小于或等于

>= 大于或等于

!= 不等于

- 变量值来源：比较数 2 的类型，自定义或变量或数字、模拟量的输入值。

- 新参数：

若变量值来源选择的类型为自定义，则此处不可选

若变量值来源选择的类型为变量 (INT、DOUBLE、BOOL、GINT、GDOUBLE、GBOOL)，则此处为比较数 1 的变量名

若变量值来源选择的类型为输入值 (DIN、AIN)，则此处为数字输入或模拟输入的端口号

- 来源参数：若变量值来源处选择的为自定义，则在此处直接填写比较数 2 的值。

## 使用范例

UNTIL(GI001<2)

其它指令

ENDUNTIL

MOVJ P003

## CRAFTLINE-工艺跳行

### 功能

专用工艺指令，程序中运行该指令后，在专用工艺界面会跳转到对应的行数。

### 参数说明

- 新参数：专用工艺界面中对应的行数。

### 使用范例

CRAFTLINE 22

## CMDNOTE-注释指令

### 功能

指令注释，可以使用该指令在程序适当位置添加注释，便于调试。

### 参数说明

- 注释内容：支持中英文。

### 使用范例

##inexbot\$\$; 含义：注释内容为“INEXBOT”。

## POS\_REACHABLE-到达判断

### 功能

到达判断指令，用于判断目标点是否能到达，点位能够到达变量置 1，不能到达置 0。

### 参数说明

- 位置变量名：可选择 P 点、G 点。
- 运动类型：可选择 MOVJ、MOVL。
- 状态存入变量类型：可存入 BOOL、GBOOL。
- 状态存入变量名：BOOL、GBOOL 变量名称。

### 使用范例

POS\_REACHABLE MOVJ P001 B001; 含义：计算能否使用 MOVJ 插补运行到 P001 位置，可以到达 B001 值为 1，不可以到达 B001 值为 0。

## CLKSTART-计时开始

### 功能

CLKSTART 指令用于计时。运行该指令开始计时，并将时间记录到一个局部或者全局 DOUBLE 变量中。

计时指令的精度为小数点后两位（即 10ms，误差 $\pm 2$ ms）

### 参数说明

- 序号：计时器的序号，可以同时使用 32 个计时器分别计时。
- 存入变量类型：将计时的时间存入到局部 DOUBLE 变量或者全局的 GDOUBLE 变量。
- 存入变量名：将时间存入的变量的变量名。

## 使用范例

CLKSTARTID= 1 D001; 含义：工艺号 1 开始计时，计时结果存入 D001。

## CLKSTOP-计时停止

### 功能

CLKSTOP 指令用于停止对应序号的计时器计时。停止后已存入变量的值不会归零。

### 参数说明

- 序号：要停止计时的计时器的序号。

## 使用范例

CLKSTOPID=1; 含义：工艺号 1 计时停止

## CLKRESET-计时复位

### 功能

CLKRESET 指令用于将对应序号的计时器归零。若没有使用该指令，下次运行 CLKSTART 指令会累积计时。

### 参数说明

- 序号：要归零计时的计时器的序号。

### 使用范例

CLKPESETID=1; 含义：重置工艺号 1 计时结果。

## READLINEAR-读取线速度

### 功能

将机器人线速度实时读取到变量中

### 参数说明

- 变量类型：存入变量的类型，可选 GINT/GDOUBLE。
- 变量名：存入变量的名

### 使用范例

READLINEAR GDOO1

## > 变量类

### INT-整型

**(定义该变量直接到局部变量界面操作即可，以取消该指令)**

## 功能

定义局部整型变量，并同时赋值。该指令必须插在程序头部。

## 参数说明

- 变量类型：此处已固定为 INT。
- 变量名：所要建立的 INT 变量的变量名，范围 1-999。
- 变量值来源：给新变量赋值，可以选择自定义手填或其他变量。
- 新参数：当变量值来源选择自定义时，在这里直接填写新变量的初始值。
- 来源参数：当变量值来源选择变量类型时，在这里选择变量名，将另外一个变量的值赋值给该变量。

## 使用范例

INTI001 = 11

INT I002 = GI003

## DOUBLE-浮点型

**(定义该变量直接到局部变量界面操作即可，以取消该指令)**

## 功能

定义局部浮点型变量，并同时赋值。该指令必须插在程序头部。

## 参数说明

- 变量类型：此处已固定为 DOUBLE。
- 变量名：所要建立的 DOUBLE 变量的变量名，范围 1-999。
- 变量值来源：给新变量赋值，可以选择自定义手填或其他变量。
- 新参数：当变量值来源选择自定义时，在这里直接填写新变量的初始值。
- 来源参数：当变量值来源选择变量类型时，在这里选择变量名，将另外一个变量的值赋值给该变量。

## 使用范例

DOUBLED001 = 11

DOUBLED002 = GD003

## BOOL-布尔型

**(定义该变量直接到局部变量界面操作即可，以取消该指令)**

## 功能

定义局部布尔型变量，并同时赋值。该指令必须插在程序头部。

注：原变量为 A000，更名为 B000。

### 参数说明

- 变量类型：此处已固定为 BOOL。
- 变量名：所要建立的 BOOL 变量的变量名，范围 1-999。
- 变量值来源：给新变量赋值，可以选择自定义手填或其他变量。
- 新参数：当变量值来源选择自定义时，在这里直接填写新变量的初始值。
- 来源参数：当变量值来源选择变量类型时，在这里选择变量名，将另外一个变量的值赋值给该变量。

### 使用范例

```
BOOLEAN B001 = 1
```

```
BOOLEANB002 = GB002
```

### SETINT-赋值整型

### 功能

给已有整型变量赋值。

## 参数说明

- 变量类型：可以选择要赋值变量为局部或全局整型变量。
- 变量名：要赋值的变量名。
- 变量值来源：可以选择自定义手填或其他变量。
- 新参数：当变量值来源选择自定义时，在这里直接填写值。
- 来源参数：当变量值来源选择变量类型时，在这里选择变量名，将另外一个变量的值赋值给该变量。

## 使用范例

SETINTI001= 2

## SETDOUBLE-赋值浮点型

## 功能

给已有浮点型变量赋值。

## 参数说明

- 变量类型：可以选择要赋值变量为局部或全局浮点型变量。
- 变量名：要赋值的变量名。
- 变量值来源：可以选择自定义手填或其他变量。

- 新参数：当变量值来源选择自定义时，在这里直接填写值。

- 来源参数：当变量值来源选择变量类型时，在这里选择变量名，将另外一个变量的值赋值给该变量。

## 使用范例

SETDOUBLED001= 2.11

## SETBOOL-赋值布尔型

## 功能

给已有布尔型变量赋值。

## 参数说明

- 变量类型：可以选择要赋值变量为局部或全局布尔型变量。

- 变量名：要赋值的变量名。

- 变量值来源：可以选择自定义手填或其他变量。

- 新参数：当变量值来源选择自定义时，在这里直接填写值。

- 来源参数：当变量值来源选择变量类型时，在这里选择变量名，将另外一个变量的值赋值给该变量。

## 使用范例

SETBOOLB001= 1

## FORCESET-写入文件

### 功能

在程序运行过程中，所有的计算、赋值操作均是对缓存中的数值进行更改的，并不会存入系统文件中，即当程序运行停止后所有全局变量的值都会还原。

若要强制将内容中的全局数值变量写入文件中，则可以使用 FORCESET 指令。

### 参数说明

- 变量类型：选择要强制写入文件的变量类型。
- 变量名：要强制写入文件的变量名。

### 使用范例

FORCESETGI001

## > 坐标切换类

### SWITCHTOOL-切换工具手

#### 功能

在程序运行中切换当前使用的工具手坐标系。

#### 参数说明

- 工具坐标：要切换到的工具手坐标系的工具号。

## 使用范例

SWITCHTOOL (3)

SWITCHUSER-切换用户坐标

## 功能

在程序运行中切换当前使用的用户坐标系。

## 参数说明

- 用户坐标：要切换到的用户坐标系的序号。

## 使用范例

SWITCHUSER (3)

USERCOORD\_TRANS-用户坐标转换

## 功能

将 B、C 用户坐标系叠加 (×)，结果置入 A 用户坐标系。

## 参数说明

- 用户坐标 A：结果存入该用户坐标系，这里是用户坐标系序号。
- 用户坐标 B：用户坐标系序号。
- 用户坐标 C：用户坐标系序号。

## 使用范例

USERCOORD\_TRANS (1) (2) (3)

## SWITCHSYNC-切换外部轴

### 功能

在程序运行中切换当前使用的外部轴。

### 参数说明

- 外部轴组号：要切换到的外部轴的组号。

## 使用范例

SWITCHSYNC 1

## > 网络通讯类

## SENDMSG-发送数据

### 功能

向另外一个网络设备发送字符串信息。

### 参数说明

- ID：设置-网络设置界面中的工艺号。

- 发送字符：要发送的字符串。若要发送变量，则在变量前加入\$。若要发送\$则要字符，则需要两个\$。支持转义符与格式化输出。

## 使用范例

```
SENDMSG ID = 1 #this is $D001#
```

## PARSEMSG-解析数据

### 功能

解析另外一台网络设备通过 TCP 发送的数据，并将数据参数多个变量中。

当有 TCP 接收到多位数值时，会将数值分别存入多个变量中，所使用的变量分别为第一位变量、第一位变量往下顺延。即，若发来 3 位数值，A、B、C，设置的第一位变量名为 GI006，则将 A 存入 GI006，B 存入 GI007，C 存入 GI008.

### 参数说明

- ID：设置-网络设置界面中的工艺号。
- 第一位变量类型：存入第一位变量的类型。
- 第一位变量名：存入第一位变量的变量名。
- 解析后清除缓存区：解析数据后清空缓存的数据

## 使用范例

PARSEMSG ID = 1 GI006CLEARCAHE= 0; 含义：把接收到的数据存到变量 GI001 中，解析完成后清除缓存的数据

## READCOMM-读取

### 功能

读取以太网或 Modbus 发送的点位存到位置变量中、个数存到数值变量中。

### 参数说明

- 工艺号：要打开通讯的网络通讯的工艺号。
- 通讯方式：使用以太网通讯或者 Modbus 通讯
- 位置变量类型：可选全局位置变量、局部位置变量
- 位置变量名：位置变量名；存接收到的点位，多个点位位置变量顺延，例如指令位置变量填 G003，接收 3 个点位，则分别存到 G003、G004、G005
- 变量类型：可选全局整型、局部整型
- 变量名：变量名；存接收到点位的数量

注：目前仅 Modbus 可用

### 使用范例

```
READCOOMID=1 EHTERNETTOG001I001
```

## OPENMSG-打开数据

## 功能

打开网络通讯。

## 参数说明

- ID：设置 - 网络设置界面中的工艺号。

## 使用范例

OPENMSGID = 1

## CLOSEMSG-关闭数据

## 功能

关闭网络通讯。

## 参数说明

- ID：设置 - 网络设置界面中的工艺号。

## 使用范例

CLOSEMSG ID = 2

## PRINTMSG-输出信息

## 功能

通过提示条的方式打印字符串。

## 参数说明

- 输出字符：要打印的字符串。若要打印变量，则在变量前加入\$。若要打印\$则要字符，则需要两个\$。支持转义符与格式化输出。

## 使用范例

```
PRINTMSG #this is $D001#
```

MSG\_CONNECTION\_STATUS-获取信息连接状态

## 功能

获取网络设置里某个工艺号的连接状态

## 参数说明

- 工艺号：需要知道的网络设置的工艺号
- 状态存入变量类型：可选 BOOL/GBOOL
- 状态存入变量名：变量名

## 使用范例

```
MSG_CONN_ST 1 B001
```

## > 位置变量类

USERFRAME\_SET-用户坐标修改

## 功能

改变用户坐标系某一轴的值。

## 参数说明

- 用户坐标编号：要改变值的用户坐标编号。
- 用户坐标参数：要改变值的用户坐标轴。
- 变量类型：可以选择手填值或其他变量。
- 变量名：当选择其他变量时，在这里选择变量名，会将该变量的值赋给用户坐标对应的坐标轴。
- 手填值：当变量类型选择手填值时，在这里直接填入要改变的目标值。

## 使用范例

USERFRAME\_SET ID = 1 UX GI001

USERFRAME\_SET ID = 2 UY 99

TOOLFRAME\_SET-工具坐标修改

## 功能

改变工具坐标系某一轴的值。

## 参数说明

- 工具坐标编号：要改变值的工具坐标编号。
- 工具坐标参数：要改变值的工具坐标轴。
- 变量类型：可以选择手填值或其他变量。

- 变量名：当选择其他变量时，在这里选择变量名，会将该变量的值赋给用户坐标对应的坐标轴。

- 手填值：当变量类型选择手填值时，在这里直接填入要改变的目标值。

## 使用范例

TOOLFRAME\_SET ID = 1 TX GI001; 含义：把工具手 1 的 X 轴偏移参数改为 GI001 的变量值

TOOLFRAME\_SET ID = 2 TY 99; 含义：把工具手 2 的 X 轴偏移参数改为 99

## READPOS-读取点位

### 功能

将一个位置变量的某个轴的值读入到一个浮点型变量中。0

### 参数说明

- 变量类型：要读入的浮点型变量类型，局部或全局。
- 变量名：要读入的浮点型变量的变量名。
- 位置变量类型：要读取的位置变量类型，当前位置、局部位置变量或全局位置变量。
- 位置变量名：当位置变量类型选择局部位置变量或全局位置变量，这里选择对应的变量名。若选择 P\$INT、P\$GINT、G\$INT、G\$GINT，在这里选择对应的整型变量

名。例选择 P\$INT, 变量名 I001, I001=33, 则得到的位置变量为 P033。

- 位置变量坐标系：要读取的位置变量值所在的坐标系。
- 位置变量轴：要读取的位置值在对应坐标系下的轴。

## 使用范例

```
READPOS GD004 P$GI003 RF 1
```

## POSADD-点位加

### 功能

位置变量加法运算 (+) , 该指令能够对位置变量 (全局、局部) 单一轴的值进行加法运算, 然后再赋值给该轴。

### 参数说明

- 位置变量类型：要改变的位置变量的类型, 局部或全局。
- 位置变量名：要改变的位置变量的变量名。
- 位置变量坐标系：要改变位置变量轴所对应的的坐标系。
- 位置变量轴：要改变位置变量在对应坐标系下的轴。

- 变量类型：可以选择手填值或其他变量。
- 数值变量名：当选择其他变量时，在这里选择变量名，会将该变量的值加上位置变量对应轴的值，再赋值给该位置变量。
- 手填值：当变量类型选择手填值时，在这里直接填入目标值，会将该值加上位置变量对应轴的值，再赋值给该位置变量。

## 使用范例

POSADD P001 RF 1 788

## POSSUB-点位减

### 功能

位置变量减法运算（-），该指令能够对位置变量（全局、局部）单一轴的值进行减法运算，然后再赋值给该轴。

### 参数说明

- 位置变量类型：要改变的位置变量的类型，局部或全局。
- 位置变量名：要改变的位置变量的变量名。
- 位置变量坐标系：要改变位置变量轴所对应的的坐标系。

- 位置变量轴：要改变位置变量在对应坐标系下的轴。
- 变量类型：可以选择手填值或其他变量。
- 数值变量名：当选择其他变量时，在这里选择变量名，会将位置变量对应轴的值减去该变量的值，再赋值给该位置变量。
- 手填值：当变量类型选择手填值时，在这里直接填入目标值，会将位置变量对应轴的值减去该值，再赋值给该位置变量。

## 使用范例

POSSUB P001 RF 1 88

## POSSET-点位改

### 功能

该指令能够对位置变量（全局、局部）单一轴的值进行修改。

### 参数说明

- 位置变量类型：要改变的位置变量的类型，局部或全局。
- 位置变量名：要改变的位置变量的变量名。

- 位置变量坐标系：要改变位置变量轴所对应的的坐标系。
- 位置变量轴：要改变位置变量在对应坐标系下的轴。
- 变量类型：可以选择手填值或其他变量。
- 数值变量名：当选择其他变量时，在这里选择变量名，会将位置变量对应轴的值减去该变量的值，再赋值给该位置变量。
- 手填值：当变量类型选择手填值时，在这里直接填入目标值，会将位置变量对应轴的值减去该值，再赋值给该位置变量。

## 使用范例

POSSUB P001 RF 1 88

## COPYPOS-复制点位

## 功能

将一个位置变量所有轴的值复制到另一个位置变量中。

## 参数说明

- 位置变量类型：要取出值的位置变量的类型。可以选择当前位置，既将当前机器人位置赋值给另一个位置变量。

- 位置变量名：要取出值的位置变量的变量名。
- 位置变量类型：被赋值的位置变量的变量类型。
- 位置变量名：被赋值的位置变量的变量名。

## 使用范例

COPYPOSG003 TOP001

COPYPOSCURPOSTOP002

POSADDALL-点位全加

## 功能

位置变量加法运算（+），该指令能够对位置变量（全局、局部）若干轴的值进行加法运算，然后再赋值给该轴。

## 参数说明

- 位置变量类型：要改变的位置变量的类型，局部或全局。
- 位置变量名：要改变的位置变量的变量名。
- 位置变量坐标系：要改变位置变量轴所对应的的坐标系。
- 变量类型：可以选择手填值或其他变量。

- 数值变量名：当选择其他变量时，在这里选择变量名，会将位置变量对应轴的值减去该变量的值，再赋值给该位置变量。

- 手填值：当变量类型选择手填值时，在这里直接填入目标值，会将位置变量对应轴的值减去该值，再赋值给该位置变量。

## 使用范例

POSADDALL G001 RF I001 GI001 D001 GD001 10.1 10

## POSSUBALL-点位全减

## 功能

位置变量减法运算（-），该指令能够对位置变量（全局、局部）若干轴的值进行减法运算，然后再赋值给该轴。

## 参数说明

- 位置变量类型：要改变的位置变量的类型，局部或全局。

- 位置变量名：要改变的位置变量的变量名。

- 位置变量坐标系：要改变位置变量轴所对应的的坐标系。

- 变量类型：可以选择手填值或其他变量。

- 数值变量名：当选择其他变量时，在这里选择变量名，会将位置变量对应轴的值减去该变量的值，再赋值给该位置变量。

- 手填值：当变量类型选择手填值时，在这里直接填入目标值，会将位置变量对应轴的值减去该值，再赋值给该位置变量。

## 使用范例

POSSUBALL G001 RF I001 GI001 D001 GD001 10.1 10

## POSSETALL-点位全改

### 功能

该指令能够对位置变量（全局、局部）若干轴的值进行修改。

### 参数说明

- 位置变量类型：要改变的位置变量的类型，局部或全局。

- 位置变量名：要改变的位置变量的变量名。

- 位置变量坐标系：要改变位置变量轴所对应的的坐标系。

- 变量类型：可以选择手填值或其他变量。

- 数值变量名：当选择其他变量时，在这里选择变量名，会将位置变量对应轴的值减去该变量的值，再赋值给该位置变量。

- 手填值：当变量类型选择手填值时，在这里直接填入目标值，会将位置变量对应轴的值减去该值，再赋值给该位置变量。

## 使用范例

POSSETALL G001 RF I001 GI001 D001 GD001 10.1 10

## TOFFSETON-轨迹偏移开始

### 功能

该指令能够对机器人的运行轨迹进行实时偏移。

### 参数说明

- 偏移坐标系：要改变运行轨迹所对应的的坐标系。
- 偏移量类型：可以选择手填值或其他变量。
- 偏移量：当变量类型选择手填值时，在这里直接填入目标值，会将机器人的轨迹坐标加上这个手填值。
- 数值变量名：当选择其他变量时，在这里选择变量名，会将机器人的轨迹坐标加上这个变量里的值。

## 使用范例

TOFFSETON RF GI001 I002 2 3 4 5

## TOFFSETOFF-轨迹偏移结束

### 功能

轨迹偏移结束，此后的运动轨迹不再偏移。

### 使用范例

TOFFSETOFF

## READPOSMSG-读取点位信息

### 功能

轨迹偏移结束，此后的运动轨迹不再偏移。

### 参数说明

- 变量类型：可选择全局位置变量和局部位置变量。
- 变量名：位置变量的名称
- 信息：读取值
- 目标变量类型：被读取的位置变量的变量类型
- 目标变量名：被读取的位置变量的变量名称

### 使用范例

READPOSMSG P001 TOOL I001

## > 程序控制类

## PTHREAD\_START-开启线程

### 功能

开启后台任务。后台任务执行一次即结束。若要编辑后台任务，请到设置-后台任务界面进行编程。

### 参数说明

- 后台任务：后台任务名。

### 使用范例

PTHREAD\_START [TTT]

## PTHREAD\_END-退出线程

### 功能

关闭已开启的后台任务。

### 参数说明

- 后台任务：后台任务名。

### 使用范例

PTHREAD\_END [TTT]

## PAUSERUN-暂停运行

### 功能

暂停程序运行。

## 参数说明

- 类型：要暂停的程序类型，包括全部、主程序、后台程序。
- 程序：要暂停的程序名。

## 使用范例

PAUSERUN [TTT]

PAUSERUN MAIN

PAUSERUN ALL

## CONTINUERUN-继续运行

## 功能

继续运行已暂停的程序（已停止的程序不能继续）。

## 参数说明

- 类型：要继续运行的程序类型，包括全部、主程序、后台程序。
- 程序：要继续运行的程序名。

## 使用范例

CONTINUERUN [TTT]

CONTINUERUN MAIN

CONTINUERUN ALL

## STOPRUN-停止运行

### 功能

停止运行所有程序。

### 参数说明

无

### 使用范例

STOPRUN

## RESTARTRUN-重新运行

### 功能

重新运行已停止的程序。

### 参数说明

无

### 使用范例

RESTARTRUN