卡诺普
CROBOTP

控制器占据半壁江山的机器人品牌

让客户用好机器人

# 编程指令说明书 (CRX9)

PROGRAMMING INSTRUCTION MANUAL
OF CRX9 SYSTEM

CRP-JS-YF2-NT-00049-A2

请确保相关说明书到达本产品的最终使用者手中。

**CROBOTP相关说明书：**

卡诺普机器人安全手册

CRX9简易操作手册

CRX9使用说明书

CRX9系统PLC说明书

CRP-G9-CD60 电柜说明书

RA轻负载机器人机械说明书

RA中负载机器人机械说明书

十分感谢您选用本公司产品！

本产品相关手册请妥善保管，以备需要时查阅！

如设备需要转手，请将相关资料一并转交对方！

机器人相关手册未做说明的按键、功能、选项视为不具备，请勿使用！

| 修订说明： | |
|---|---|
| 2022-06-13 | 初稿 |
| 2023-04-06 | 修改封面与部分图片 |
| 2023-10-12 | 修订内容 |

# 用户须知

指令元素中“[ ]”内的指令元素表示可选元素，“\”表示指令中的可选元素默认为不选择状态。

# 前 言

1. 在使用机器人之前，请务必仔细阅读本公司机器人相关说明书，并在理解了该项内容基础上再进行机器人操作。

2. 本公司郑重建议: 所有参与机器人操作、示教、维护、维修、点检的人员，需预先学习本公司系统的操作说明书。

3. 本公司保留未经预先通知而改变、修订或更新本手册的权利。

5. 事先未经本公司书面许可，不可以将本手册全部或其中的一部分再生或复制。

6. 请将本手册小心存放，确保本说明书到达最终使用者手中。机器人如果需要重新安装、或搬运到不同地点、或卖给其他用户时，请务必将本手册附上。一旦出现丢失或严重损坏，请您和本公司代理商或技术人员联络。

7. 所有参数指标和设计可能会随时修改，在不影响使用效果的前提下，恕不另行通告。

8. 我们试图在本说明书中描述可能多的情况。然而对于那些不必做的和不可能发生的情况，由于存在各种可能性，我们没有描述。因此，对于那些在说明书中没有特别进行描述的情况，可以视为“不可能”的情况。

9. 在本书编写的过程中难免会出现遗漏和错误，如在阅读过程中发现有错误或不能理解的地方，欢迎来电咨询并指正。

# 安全

## 简介

本节主要介绍在使用机器人时需要注意的安全原则和流程，在使用机器人之前，请务必熟读并理解本章中所述内容，并按安全操作规程操作机器人。且使用前（安装、运转、保养、检修），请务必熟读并全部掌握本说明书和其他相关资料。

本手册给出的图表、顺序和详细解释可能并不绝对正确。所以，在使用本手册去作业时，有必要投以最大的注意力。一旦出现未说明的问题或麻烦，请与卡诺普联系。

为保证每项工作的安全，请阅读并完全理解本手册和《机器人安全手册》、相关法律、法规、法令及其相关资料中各种有关安全的解释和描述，同时请为各项工作采取合适的安全措施。

除安全章节外，请注意在文档的必要部分有其他的安全提示。

## 安全责任说明

本手册并不对使用非本公司机器人的应用做担保。同时，我司将不会对使用这样的机器人而可能导致的事故、损害和(或)与工业产权相关的任何问题承担责任。

我司尽可能提供出可靠的安全信息，但不对因使用本手册及其中所述产品引起的意外或间接事故承担责任。

除本手册中有明确陈述之外，本手册的内容不应解释为卡诺普对个人损失、财产损坏或具体适用性做出任何担保或保证。

卡诺普对本手册可能出现的错误概不负责。

## 安全标志

| 标志 | 说明 |
| --- | --- |
|  危险 | 表示如果无视该标识并进行错误使用，则可能会导致死亡或重伤等。 |
|  警告 | 误操作时有危险，可能发生中等程度伤害或轻伤事故及设备故障。 |
|  小心 | 不遵守本标志内容可能会引起人身伤害和/或机械损伤。 |
| ★ 注意 | 表示关于机器人规格、操作和维护的注意信息。 |

说明：即使是“小心”所记载的内容，也会因情况不同而产生严重后果，因此任何一条注意事项都极为重要，请务必严格遵守。

甚至在有些地方连“警告”或“危险”等内容都未记载，也是用户必须严格遵守的事项。

## 拟定用途

机器人控制器以及机器人只限于一般工业设备使用，不可用于与预定用途违背的应用，禁止用途包括但不限于以下情况：

- 用于易燃易爆等危险环境中；
- 用于移动或搬运人或其他动物的装置；
- 用于涉及人命的医疗设备等装置；
- 用于对社会性及公共性有重大影响的装置；
- 用于车载、船舶等受到振动环境；
- 用于攀爬工具使用。

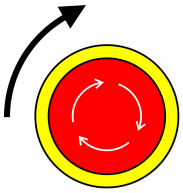
## 急停按钮

紧急停止属于安全停止的一种，是机器人系统中优先级最高的功能。在示教器、电柜、工位盒等均安装有急停按钮。如遇紧急情况，用户可按下急停按钮，立即切断机器人电源。

紧急停止用的急停按钮大多数使用红色的操作主体，最常见的外形是蘑菇头型。如下图所示。

CRP-20220318-2

若需复位，则需按照急停按钮上的箭头方向旋转（如下图所示），急停按钮将弹起复位。



CRP-20220318-1

## 使用前安全须知

1、 搬运和安装机器人时，请务必按照卡诺普公司说明书中所示的方法进行。否则可能导致机器人翻倒，引发事故；

2、 请务必在机器人安装前划分出安全区域。可在机器人工作区域周围安装栅栏及警示牌保证机器人安全工作，防止闲杂人等进入以及防止机器人伤人；

3、机器人上方不能有悬挂物，以防掉落砸坏机器人等设备；

4、严禁倚靠电控柜，或者随意触动按钮，以防机器人产生未预料的动作，引起人身伤害或者设备损坏；

5、拆分机器人时，注意机器人上可能掉落的零件而砸伤人员；

6、在进行外围设备的个别调试时，务必断开机器人电源后执行；

7、外围设备均应连接适当的地线；

8、初次使用机器人操作时，请务必先以低速运行，待运行无误后再逐渐加速。

9、请注意对电控柜与机器人、外围设备间的配线及配管采取防护措施，以免被人踩坏或被叉车辗压而坏；

10、任何工作的机器人都可能因有不可预料的动作，对工作范围内的人员造成严重的伤害或者对设备造成破坏。在准备机器人工作前，需测试各安全措施（栅栏门、抱闸、安全指示灯）的可靠性；

11、 在开启机器人前，确保机器人工作范围内没有其他人员；

12、 通过软件设定的动作范围及负载条件切勿超出产品规格表中的规定值，设置不当可能造成人员伤害或机器损坏；

13、在进入操作区域内工作前，即便机器人没有运行，也要关掉电源或者按下急停按钮；

14、当在机器人工作区内编程时，设置相应看守，保证机器人能在紧急情况，迅速停止。示教和点动机器人时不要带手套操作，点动机器人时要尽量采用低速操作，遇异常情况时可有效控制机器人停止；

15、必须知道机器人控制器和外围控制设备上的紧急停止按钮的位置，以便在紧急情况下能准确的按下这些按钮；

16、永远不要认为机器人处于静止状态时其程序就已经完成。此时机器人很有可能是在等待让它继续运动的输入信号；

# 安全操作规程

## 操作前注意事项

**注意**

★进行机器人示教作业前要检查以下事项，有异常则应及时修理或采取其他必要措施。

- 机器人动作有无异常。
- 原点是否校准正确。
- 与机器人相关联的外部辅助设备是否正常。

★操作机器人必须确认

- 操作人员是否接受过机器人操作的相关培训。
- 对机器人的运动特性有足够的认识。
- 对机器人的危险性有足够的了解。
- 未酒后上岗。
- 未服用影响神经系统、反应迟钝的药物。

## 紧急停止

**危险**

★ 操作机器人前，请按下急停键，并确认伺服主电源被切断，电机处于失电并抱闸状态。伺服电源切断后，伺服电源指示按钮为红色。

紧急情况下，若不能及时制动机器人，则可能引发人身伤害或设备损坏事故。

★ 解除急停后再接通伺服电源时，要解除造成急停的事故后再接通伺服电源。

由于误操作造成的机器人动作，可能引发人身伤害事故。

## 机器人操作注意事项

★在机器人动作范围内示教时，请遵守以下原则：

• 保证机器人在视野范围内

• 严格遵守操作步骤

• 考虑机器人突然向自己所处方位运动时的应变方案

• 确保设置躲避场所，以防万一

由于误操作造成的机器人动作，可能引发人身伤害事故。

★进行以下作业时，请确认机器人的动作范围内操作人员和障碍物：

• 机器人控制电柜接通电源时

• 用示教编程器操作机器人时

• 试运行时

• 自动再现时

不慎进入机器人动作范围内或与机器人发生接触，都有可能引发人身伤害事故。发生异常时，请立即按下急停按钮。

★示教器用完后须放回原处，并确保放置牢固。

• 如不慎将示教编程器放在机器人、夹具或地上，当机器人运动时，示教编程器可能与机器人或夹具发生碰撞，从而引发人身伤害或设备损坏事故。

• 防止示教器意外跌落造成机器人误动作，从而引发人身伤害或设备损坏事故。

• 示教器IP防护等级较低

# 目 录

# 一、运动指令

## 1.1 MoveJ---关节运动

### 1.1.1 指令用法

用于机器人对末端运动路径没有要求的场合，MoveJ将机器人以最快的速度从一个点运行到另一个点。所有轴均同时运动，机器人运动轨迹是一条不确定的曲线。

### 1.1.2 应用举例

例1：

```
MoveJ  VJ=100  PL=9  T=1
```

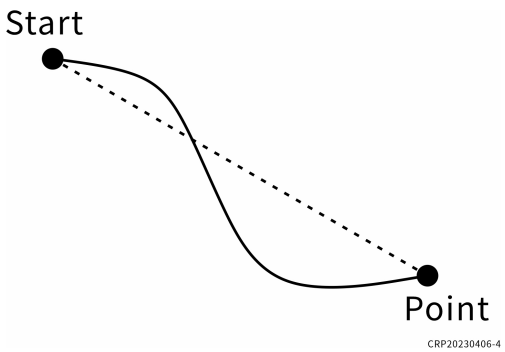



图 1.2.1

将工具1的工具中心点沿非线性路径移动到指令中存储的指定位置，其速度百分比为100%，轨迹平滑度为9。

例2：

```
MoveJ  GP0  VJ=100  PL=2  T=2
```

将工具2的工具中心点沿非线性路径移动到位置GP0，其速度百分比为100%，轨迹平滑度为2。

### 1.1.3 程序运行

运行指令时机器从一点迅速运行到另一点，该指令过程中各轴均以设定轴速率运动，且同时到达目标点，其运动轨迹非线性。当机器人各轴已经到达目标位置（目标位置与指令中PL参数有关系），才会执行下一条指令。一般用于没有轨迹要求的场合。

### 1.1.4 可变元素

MoveJ [\Topoint] [\ID] VJ=[Speed] PL=[zone] [\ACC] [\Dec] T=[Tool] [\U] [\OP] [\signal ]

1、 [\Topoint]

变量类型：GP（全局变量）、LP（局部变量）。

解释：机器人和外部轴的目标点位置。可以选择使用GP、LP变量。当选择变量时，变量中的值为机器人和外部轴的目标点位置。不选择变量时机器人目标位置储存在指令中。

2.、[\ID]

数据类型：Int (范围：1-999)

解释：协同同步时必须使用。在其他任何时候都不允许使用。所有协同同步任务时ID号必须相同，ID号确保协同运行时不混淆。默认不使用。

3、VJ=[Speed]

数据类型： float（范围：0.0-100.0）

变量类型：GR（全局变量）、LR（局部变量）

解释：关节运动以最大速度为基准的百分比。可以选择使用GR、LR变量。当选择变量时，使用变量中的值作为机器人运行速度的百分比。

5、PL=[zone]

数据类型：float（范围：0.0-9.0）

解释：机器人在运动轨迹中到达目标点的逼近等级。描述了所生成拐角的路径大小。逼近等级设置越小，机器人运行轨迹时生成拐角路径越小。

6、[\ACC]

数据类型：Int（范围：1-20）

解释：加速度，该加速度与机器人运动方式有关。加速等级设置越小机器人加速越快，加速等级设置越大加速越慢。默认不使用。

7、[\Dec]

数据类型：Int（范围：1-20）

解释：减速度，该减速度与机器人运动方式有关。减速等级设置越小机器人减速越快，减速等级设置越大减速越慢。默认不使用。

8、T=[Tool]

数据类型:Int (范围：0-49)

解释：机器人移动时使用的工具坐标系。0号工具默认工具坐标点在机器人法兰末端中心处。

9、[\U]

数据类型：Int (范围：0-49)

解释：机器人移动时使用的工件坐标系。0号工件坐标为默认工件坐标系。可选择使用该元素。如果使用协同的外部轴时必须使用该参数。

10、[\OP]

变量类型：OP（局部变量）

解释：机器人在一个指定位置添加一个偏移量。

11、[\signal ]

变量类型:Until（X、SX、M、SM、T、C）

解释：当条件满足，该指令行停止执行，转入下一行执行。否则执行当前行直到结束后，转入下一行。可选择使用该元素。默认不使用。

### 1.1.5 详细举例

例1：

```
MoveJ  VJ=GR0  PL=6  Acc= 1  T=1
```

将工具1的工具中心点沿非线性路径移动到指令中存储的指定位置，其速度百比为储存在变量GR0中的值，轨迹平滑度为6，加速度为1。

例2：

```
MoveJ  GP0  VJ=100  PL=2  T=2  U=1
```

将工具2的工具中心点沿非线性路径移动到GP0，在关于U1的工件坐标系下的指定位置，其速度百分比为100%，轨迹平滑度为2。

例3：

```
MoveJ  GP0  VJ=100  PL=2  Dec=5  T=2  U=1  OP1
```

将工具2的工具中心点沿非线性路径移动到GP0，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其速度百分比为100%，轨迹平滑度为2，减速度为5。

例4：

```
MoveJ  GP0 VJ=100  ID=6  PL=2  Acc=10  Dec=5  T=2  U=1  OP1 Until X0=ON
```

将工具2的工具中心点沿非线性路径移动到GP0，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其速度百分比为100%，轨迹平滑度为2，加速度等级为10、减速度等级为5。运行过程中，若X0的值等于ON，则直接停止运行该行指令，然后直接进入下一行运行；如果X0的值不等于ON，则将该行指令运行完成后再进入下一行运行。

## 1.2 MoveL---直线运动

### 1.2.1 指令用法

将工具中心点以直线的轨迹方式运动到目标点。运动姿态参照TCP坐标方向。

### 1.2.2 应用举例

例1：

```
MoveL  VL=100  PL=9  T=1
```

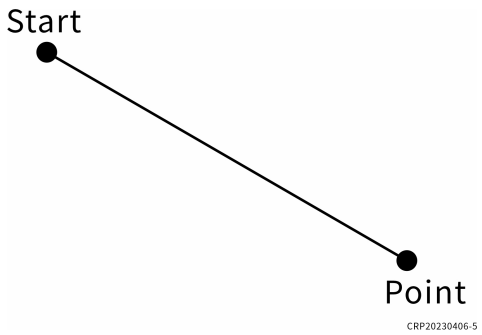



图1.2.1

将工具1的工具中心点以固定的姿态沿直线路径移动到指令中存储的指定位置，其速度为100mm/s，轨迹平滑度为9。

例2：

```
MoveL GP0  VL=100  PL=2  T=2
```

将工具T2的工具中心点以固定的姿态沿直线路径移动到位置GP0，其速度为100mm/s，轨迹平滑为2。

### 1.2.3 程序运行

运行该指令，机器人以恒定的速度、固定的姿态沿直线轨迹移动工具。当机器人工具末端到达目标位置（目标位置与指令中PL参数有关系），才会执行下一条指令。一般用于有轨迹要求的场合。

### 1.2.4 可变元素

MoveL [\Topoint] [\ID] VJ=[Speed] PL=[zone] [\ACC] [\Dec] T=[Tool] [\U] [\OP] [\Signal]

1、 [\Topoint]

变量类型：GP（全局变量）、LP（局部变量）。

解释：机器人和外部轴的目标点位置。可以选择使用GP、LP变量。当选择变量时，机器人目标点位置储存在变量中。不选择变量时机器人目标位置储存在指令中。

2、 [\ID]

数据类型：Int (范围：1-999)

解释：协同同步时必须使用。在其他任何时候都不允许使用。所有协同同步任务时ID号必须相同，ID号确保运行时不混淆。默认不使用

3、 VJ=[Speed]

数据类型： float（范围：0-3000）

变量类型：GR（全局变量）、LR（局部变量）

解释：直线运动最大速度3000mm/s。可以选择使用GR、LR变量。当选择变量时机器人运行速度为变量中的值。

5、PL=[zone]

数据类型： float（范围：0.0-9.0）

解释：机器人在运动轨迹中到达目标点的逼近等级。描述了所生成拐角的路径大小。逼近等级设置越小，机器人运行轨迹时生成拐角路径越小。

6、[\ACC]

数据类型：Int（范围：1-20）

解释：加速度，该加速度与机器人运动方式有关。加速等级设置越小机器人加速越快，加速等级设置越大加速越慢。默认不使用。

7、[\Dec]

数据类型：Int（范围：1-20）

解释：减速度，该减速度与机器人运动方式有关。减速等级设置越小机器人减速越快，减速等级设置越大减速越慢。默认不使用。

8、T=[Tool]

数据类型:Int (范围：0-49)

解释：机器人移动时使用的工具坐标系。0号工具默认工具坐标点在机器人法兰末端中心处。

9、[\U]

数据类型：Int (范围：0-49)

解释：机器人移动时使用的工件坐标系。0号工件坐标为默认工件坐标系。可选择使用该元素。如果使用协同的外部轴时必须使用该参数。

10、[\OP]

变量类型：OP（局部变量）

解释：机器人在一个位置添加一个偏移量。

11、[\signal ]

变量类型:Until（X、SX、M、SM、T、C），Dout（Y、M），Aout（I、V）。

解释：选择Until时，当条件满足，该指令行停止执行，转入下一行执行。否则执行当前行直到结束后，转入下一行。选择Dout、Aout时，运行该指令将会有对应的信号输出。

### 1.2.5 详细举例

例1：

```
MoveL  VL=GR0  PL=6  Acc= 1  T=1
```

将工具T1的工具中心点以固定的姿态沿直线路径移动到指令中存储的指定位置，其速度为GR0变量中的值，轨迹平滑度为6，加速度为1。

例2：

```
MoveL  GP0  VL=100  PL=2  T=2  U=1
```

将工具T2的工具中心点以固定的姿态沿直线路径移动到GP0，在关于U1的工件坐标系下的指定位置，其速度为100mm/s，轨迹平滑度为2。

例3：

```
MoveL  GP0  VL=100  PL=2  Dec=5  T=2  U=1  OP1
```

将工具T2的工具中心点以固定的姿态沿直线路径移动到GP0，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其速度为100mm/s，轨迹平滑度为2，减速度为5。

例4：

```
MoveL  GP0  VL=100  ID=6  PL=2  Acc=10  Dec=5  T=2  U=1  OP1  Aout I=5 V=8
```

将工具T2的工具中心点以固定的姿态沿直线路径移动到GP0，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其速度为100mm/s，轨迹平滑度为2，加速度为10减速度为5，运行过程中输出电流5A、电压为8V。

例5：

```
MoveL  GP0  VL=100  ID=6  PL=2  Acc=10  Dec=5  T=2  U=1  OP1  Dout  Y=ON S=5 E=8
```

将工具T2的工具中心点以固定的姿态沿直线路径移动到GP0，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其速度为100mm/s，轨迹平滑度为2，加速度为10减速度为5，运行过程中轨迹运行到5mm输出Y=ON，运行到8mmY=ON复位。

## 1.3 MoveC---圆弧运动

### 1.3.1 指令用法

将工具中心点以圆弧的轨迹方式运行到目标点。运动姿态参照TCP坐标方向。机器人做圆弧运动时，分别需要一条含Point=2和一条含Point=3的圆弧指令配合使用，指令中除点位数据外所有元素参数必须一致。

### 1.3.2 应用举例

例1：

```
MoveC  VL=100  PL=9  Point=2  T=1

MoveC  VL=100  PL=9  Point=3  T=1
```

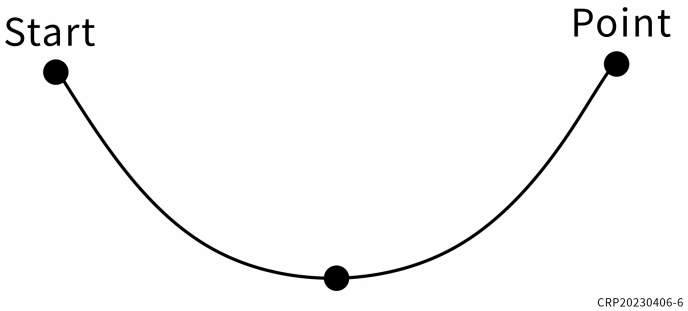



图 1.3.1

将工具T1的工具中心点从当前位置以固定的姿态沿圆弧路径，以机器人当前点为起点，以Point=2为过渡点，运行到Point=3目标点，其速度为100mm/s，轨迹平滑度为9。

例2：

```
MoveC  GP0  VL=100  PL=2  Point=2  T=2

MoveC  GP1  VL=100  PL=2  Point=3  T=2
```

将工具T1的工具中心点从当前位置以固定的姿态沿圆弧路径以机器人当前点为起点，GP0为过渡点，移动到目标点GP1，其速度为100mm/s，轨迹平滑度为2。

### 1.3.3 程序运行

运行该指令机器人以恒定的速度，固定的姿态以当前位置为起点沿圆弧轨迹移动工具。当机器人工具末端到达目标位置（目标位置与指令中PL参数有关系），才会执行下一条指令。一般用于有轨迹要求的场合。

### 1.3.4 可变元素

MoveC [\Topoint] [\ID] VJ=[Speed] PL=[zone] Point=[PT] [\ACC] [\Dec] T=[Tool] [\U] [\OP] [\Signal]

1、[\Topoint]

变量类型：GP（全局变量）、LP（局部变量）。

解释：机器人和外部轴的目标点位置。可以选择使用GP、LP变量。当选择变量时，机器人目标点位置储存在变量中。不选择变量时机器人目标位置储存在指令中。

2、[\ID]

数据类型：Int (范围：1-999)

解释：协同同步时必须使用。在其他任何时候都不允许使用。所有协同同步任务时ID号必须相同，ID号确保运行时不混淆。默认不使用。

3、VJ=[Speed]

数据类型： float（范围：0-100）

变量类型：GR（全局变量）、LR（局部变量）

解释：圆弧运动最大速度3000mm/s。可以选择使用GR、LR变量。当选择变量时机器人运行速度为变量中的值。

4、 PL=[zone]

数据类型：float（范围：0.0-9.0）

解释：机器人在运动轨迹中到达目标点的逼近等级。描述了所生成拐角的路径大小。逼近等级设置越小，机器人运行轨迹时生成拐角路径越小。

5、Point=[PT]

变量类型：Int（范围：2-3）

解释：指定指令中机器人圆弧轨迹的中间点或目标点。

6、[\ACC]

数据类型：Int（范围：1-20）

解释：加速度，该加速度与机器人运动方式有关。加速等级设置越小机器人加速越快，加速等级设置越大加速越慢。默认不使用。

7、[\Dec]

数据类型：Int（范围：1-20）

解释：减速度，该减速度与机器人运动方式有关。减速等级设置越小机器人减速越快，减速等级设置越大减速越慢。默认不使用。

8、T=[Tool]

数据类型:Int (范围：0-49)

解释：机器人移动时使用的工具坐标系。0号工具默认工具坐标点在机器人法兰末端中心处。

9、 [\U]

数据类型：Int (范围：0-49)

解释：机器人移动时使用的工件坐标系。0号工件坐标为默认工件坐标系。可选择使用该元素。如果使用协同的外部轴时必须使用该参数。

10、[\OP]

变量类型：OP（局部变量）

解释：机器人在一个位置添加一个偏移量

11、[\signal ]

变量类型:Until（X、SX、M、SM、T、C），Dout（Y、M），Aout（I、V）。

解释：选择Until时，当条件满足，该指令行停止执行，转入下一行执行。否则执行当前行直到结束后，转入下一行。选择Dout、Aout时，运行该指令将会有对应的模拟量信号输出。

### 1.3.5 详细举例

例1：

```
MoveC  VL=GR0  PL=6  Point=2  Acc= 1  T=1

MoveC  VL=GR0  PL=6  Point=3  Acc= 1  T=1
```

将工具T1的工具中心点从当前位置开始以固定的姿态圆弧线路径移动到指令中含有Point=3元素存储的指定位置，其速度为GR0储存数据，轨迹平滑度为6，圆弧过渡点储存在含有Point=2的指令中，圆弧终点储存在含有Point=3的指令中。

例2：

```
MoveC  GP0  VL=100  PL=2  Point=2  Acc= 1  T=2  U=2

MoveC  GP1  VL=100  PL=2  Point=3  Acc= 1  T=2  U=2
```

将工具T2的工具中心点从当前位置开始以固定的姿态沿圆弧路径移动到GP1，在关于U1的工件坐标系下的指定位置，其中过渡点位置为GP0，目标点位置为GP1，速度为100mm/s，轨迹平滑度为2，加速度为1。

例3：

```
MoveL  GP0  VL=100  PL=2  Point=2  Dec=5  T=2  U=1  OP1

MoveL  GP1  VL=100  PL=2  Point=3  Dec=5  T=2  U=1 OP1
```

将工具T2的工具中心点从当前位置开始以固定的姿态沿圆弧路径移动到GP1，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其过渡点位GP0，目标点位GP1，速度为100mm/s，轨迹平滑度为2，减速度为5。

例4：

```
MoveC  GP0  VL=100  ID=6  PL=2  Point=2  Acc=10  Dec=5  T=2  U=1
OP1 Aout I=5 V=8

MoveC  GP1  VL=100  ID=6  PL=2  Point=3  Acc=10  Dec=5  T=2  U=1
OP1  Aout I=5  V=8
```

将工具T2的工具中心点从当前位置开始以固定的姿态沿圆弧路径移动到GP01，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其过渡点位GP0，目标点位GP1，速度为100mm/s，轨迹平滑度为2，减速度为5，运行过程中输出I=5、V=8。

例5：

```
MoveC  GP0  VL=100  ID=6  PL=2 Point=2  Acc=10  Dec=5  T=2  U=1  OP1
Dout  Y=ON  S=5  E=8
```

MoveC GP0 VL=100 ID=6 PL=2 Point=2 Acc=10 Dec=5 T=2 U=1 OP1 Dout Y=ON S=5 E=8

将工具T2的工具中心点从当前位置开始以固定的姿态沿直线路径移动到GP1，在关于U1的工件坐标系下偏移位置为OP1的指定位置，其过渡点位GP0，目标点位GP1，速度为100mm/s，轨迹平滑度为2，减速度为5，运行过程中轨迹运行到5mm时，输出Y=ON，运行到8mm时，Y=ON复位。

## 1.4 MoveAbsJ ---绝对关节运动

### 1.4.1 指令用法

执行该指令时机器人运动将不会受到指令中的工具坐标和用户坐标影响，机器人和外部轴将运动到一个以轴位置定义的目标位置，机器人运动轨迹是一条不确定的曲线。

### 1.4.2 应用举例

例1：

MoveAbsJ VJ=100 PL=9

将工具T1的工具中心点沿非线性路径移动到指令中存储的指定绝对位置，其速度百分比为100%，轨迹平滑度为9。

例2：

MoveAbsJ GJ0 VJ=100 PL=2

将工具T2的工具中心点沿非线性路径移动到绝对位置GJ0，其速度百分比为100%，轨迹平滑度为2。

### 1.4.3 程序运行

运行该指令、该指令执行的动作不受工具坐标和用户坐标影响，机器人各轴将

同步运动到以轴位置定义的目标位置。当机器人工具末端到达目标位置（目标位置与指令中PL参数有关系），才会执行下一条指令。一般用于没有有轨迹要求的场合。

### 1.4.4 可变元素

MoveAbsJ [\Topoint] VJ=[Speed] PL=[zone] [\ACC] [\Dec] [\signal]

1. [\Topoint]

变量类型：GJ（全局变量）、LJ（局部变量）。

解释：机器人和外部轴的目标点位置，可以选择使用GJ、LJ变量进行存储。当选择变量时，机器人目标点位置储存在变量中。不选择变量时机器人目标位置储存在指令中。

2. VJ=[Speed]

数据类型： float（范围：0-100）

变量类型：GR（全局变量）、LR（局部变量）

解释：其速度以最大速度为基准的百分比。可以选择使用GR、LR变量。当选择变量时，使用变量中的值作为机器人运行速度的百分比。

4. PL=[zone]

数据类型：float（范围：0.0-9.0）

解释：机器人在运动轨迹中到达目标点的逼近等级。描述了所生成拐角的路径大小。逼近等级设置越小，机器人运行轨迹时生成拐角路径越小。

5. [\ACC]

数据类型：Int（范围：1-20）

解释：加速度，该加速度与机器人运动方式有关。加速等级设置越小机器人加速越快，加速等级设置越大加速越慢。默认不使用。

6. [\Dec]

数据类型：Int（范围：1-20）

解释：减速度，该减速度与机器人运动方式有关。减速等级设置越小机器人减速越快，减速等级设置越大减速越慢。默认不使用。

7. [\signal ]

变量类型:Until（X、SX、M、SM、T、C）

解释：当条件满足，该指令行停止执行，转入下一行执行。否则执行当前行直到结束后，转入下一行。可选择使用该元素。默认不使用。

### 1.4.5 详细举例

例1：

```
MoveAbsJ  VJ=GR0  PL=6  Acc= 1
```

机器人将沿非线性路径移动到指令中存储的绝对位置处，其速度百比用储存在变量GR0，轨迹平滑度为6，加速度为1。

例2：

```
MoveAbsJ  GJ0  VJ=100  PL=2
```

机器人将沿非线性路径移动到绝对位置GJ0，其速度为100mm/s，轨迹平滑度为2。

例3：

```
MoveAbsJ  GJ0  VJ=100  PL=2  Dec=5
```

机器人将沿非线性路径移动到绝对位置GJ0，其速度百分比为100%，轨迹平滑度为2，减速度为5。

例4：

```
MoveAbsJ  GJ0  VJ=100  PL=2  Acc=10  Dec=5  Until X0=ON
```

机器人将沿非线性路径移动到绝对位置GP0，其速度百分比为100%，轨迹平滑度为2，减速度为5。运行过程中，若X0的值等于ON，则直接停止运行该行指令，然后直接进入下一行运行；如果X0的值不等于ON，则将该行指令运行完成后再进入下一行运行。

# 二、I/O指令

## 2.1 Dout ---输出

### 2.1.1 指令用法

用于输出信号状态，将信号状态置位为ON或OFF。

### 2.1.2 应用举例

例1：

```
DOut Y1=OFF
```

将Y1置位为OFF。

例2：

```
DOut Y(UI0)=ON
```

将Y（UI0）置位为ON。Y的信号口数据储存在变量UI0中。

### 2.1.3 程序运行

程序执行到当前行时将输出端口的信号状态量置位成指定状态。在信号获得新的状态时存在短暂的延时。

### 2.1.4 可变元素

```
Dout  [signal]
```

[signal]

变量类型：Y、M。

解释：输出信号的类型和地址。

### 2.1.5 详细举例

例1：

DOut M(UI0)=ON

执行到该行程序时将M（UI0）置位为ON。M的地址号为变量UI0中的值。

例2：

DOut M1=OFF

执行到该行程序时将M1置位为OFF。

## 2.2 TDout ---延时输出

### 2.2.1 指令用法

通过延迟指定时间后改变输出信号状态，不影响程序向下执行。

### 2.2.2 应用举例

例1：

TDOut Y1=OFF T100

延时100ms后，将Y1置位为OFF。

例2：

TDOut Y(UI0)=ON T=200

延时200ms后，将Y（UI0）置位为ON。Y的地址号为变量UI0的值。

### 2.2.3 程序运行

执行到当前程序行时，后台自动开始计时，程序继续向下执行，当后台计时等于指令中设定时间时，将输出指令中设置的信号状态。

### 2.2.4 可变元素

```
Dout  [signal]  [time]
```

[signal]

变量类型：Y、M。

解释：输出信号的类型和地址。

[Time]

数据类型：Int(0-9999)

变量类型：UI

解释：延时输出的延时时间。

### 2.2.5 详细举例

例1：

```
TDOut  M1=ON   T=100
```

执行到该行程序时,系统在后台延时100ms后将M1置位为ON。执行过程中程序不停止继续向下运行。

例2：

```
TDOut  M（UI0）=OFF  T=200
```

执行到该行程序时,系统在后台延时200ms后将M(UI0)置位为OFF。执行过程正程序不停止继续向下运行。M的地址号存储在变量UI0中。

例3：

```
TDOut  M1=OFF  T=（UI0）
```

执行到该行程序时,系统在后台延时UI0后将M1置位为OFF。执行过程正程序不停止继续向下运行。延时时间为变量UI0的值。

## 2.3 Pulse ---脉冲输出

### 2.3.1 指令用法

用于脉冲信号状态输出。

### 2.3.2 应用举例

例1：

```
Pulse  Y1=OFF  T100
```

输出持续100ms的将Y1置位为OFF的脉冲信号。

例2：

```
Pulse  M(UI1)=ON  T100
```

输出持续100ms的将M（UI0）置位为ON的脉冲信号。M信号的地址为变量UI1的值。

### 2.3.3 程序运行

程序执行到当前行时，后台执行输出一个指令中设定时长的脉冲信号，在脉冲启动后程序不停止继续向下执行下一条指令。

### 2.3.4 可变元素

```
Pulse  [signal]  [time]
```

1. [signal]

变量类型：Y、M。

解释：输出信号的类型及地址。

2. [Time]

数据类型：Int(0-9999)

变量类型：UI

解释：脉冲输出持续时间，单位毫秒。

### 2.3.5 详细举例

例1：

```
Pulse  M(UI1)=ON  T100
```

执行到该行程序时,系统在后台执行一个持续时间为100ms的将M（UI1）置位为ON的脉冲信号。脉冲启动后程序不停止、继续向下运行。M的地址号为变量UI1的值。

例2：

```
Pulse  M(UI1)=ON  T=（UI0）
```

执行到该行程序时,系统在后台执行一个持续时间为UI0的将M（UI1）置位为ON的脉冲信号。脉冲启动后程序不停止继续向下运行。M的地址号为变量UI1的值，脉冲持续时间为比变量UI0的值。

## 2.4 Wait ---等待

### 2.4.1 指令用法

用于程序等待，直到预定信号出现，程序继续向下执行。

### 2.4.2 应用举例

例1：

```
Wait  X0==ON  DT=0  CT=0
```

持续等待X0信号输入为ON,信号持续0ms后程序向下执行。

例2：

Wait X0==ON DT=0 CT=100

持续等待X0信号输入为ON,并信号持续100ms后程序向下执行。

### 2.4.3 程序运行

程序执行到当前行时，等待信号满足指令中预定状态后，信号持续时间满足设定时间后，程序继续向下执行。

### 2.4.4 可变元素

Wait [Signal] DT=[time] CT=[time]

1. [Signal]

变量类型：X、SX、Y、SY、M、SM、T、C

解释：输入的信号类型及地址、以及等待的时间。

2. [Time]

数据类型：Int(0-9999)

变量类型：UI

解释：等待时间或信号持续时间

### 2.4.5 详细举例

例1：

Wait X0==ON DT=100 CT=0

执行到该行程序时,等待X0置位为ON时，X0置位为ON的持续时间为0ms后，程序继续向下执行，允许的最长等待时间为100ms。如果超过允许等待最长时间后，X0仍不满足设定信号，程序任继续向下执行。

例2：

Wait X0==ON　DT=100　CT=UI1

执行到该行程序时,等待X0置位为ON时，X0置位为ON的持续时间为UI1后，程序继续向下执行，允许的最长等待时间为100ms。如果超过允许等待最长时间后，X0任不满足设定信号，程序任继续向下执行。持续时间为变量UI1的值。

例3：

Wait M（UI0）==ON　DT=0　CT=100

执行到该行程序时,持续等待M（UI0）置位为ON并持续100ms后，程序继续向下执行。M信号地址为变量UI0的值。

## 2.5 Aout ---模拟量输出

### 2.5.1 指令用法

用于模拟量信号数据输出。

### 2.5.2 应用举例

例1：

AOut  AV1=5.000

执行到该行程序时将输出模拟量AV1为5.000。

例2：

AOut  AV1=GR2

执行到该行程序时将输出模拟量AV1，AV1的值为变量GR2的值。

### 2.5.3 程序运行

程序执行到当前行时将在指定的模拟量输出端口输出一个指定的模拟量信号。

在信号口获得新的状态时存在短暂的延时。

### 2.5.4 可变元素

AOut [Signal]

1. [Signal]

变量类型：AV

解释：模拟量输出端口。

### 2.5.5 详细举例

例1：

AOut AV(UI1)=5.000

执行到该行程序时,模拟量端口AV（UI1）输出模拟量为5.000。其中AV的地址号为变量UI1的值。

例2：

AOut AV(UI2)=GR3

执行到该行程序时,模拟量端口AV（UI2）输出模拟量为GR3。其中AV的地址号为变量UI2的值。模拟量的值为变量GR3的值。

# 三、控制指令

## 3.1 If ---判断

### 3.1.1 指令用法

判断条件是否满足，执行不同指令。

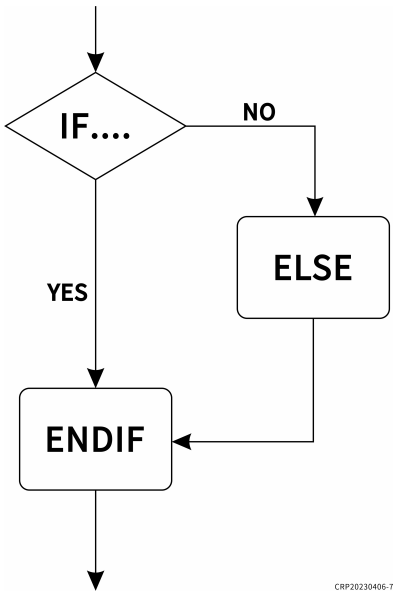



图 3.1.1

### 3.1.2 应用举例

例1：

```
If   X0==ON
  Dout   Y0=ON
EndIf
```

当满足"X0"信号的值为"ON"时，输出"Y0"状态为"ON"，"If"判断语句结束。否则直接结束判断语句。

例2：

```
If  GI0 > 1
```

```
    Dout Y0=ON
Else
    Dout Y1=OFF
EndIf
```

当满足GI0信号的值大于1时，输出Y0状态为ON，If判断语句结束。否则输出Y1状态为OFF，If判断语句结束。

### 3.1.3 程序运行

程序执行时将依次判断是否满足条件，直到满足其中一个条件。如果条件满足将继续执行程序。如果不满足任何条件，将通过符合Else的指令继续执行程序。如果满足多个条件，仅执行与第一个此类条件相关的指令。

### 3.1.4 可变元素

```
IF  [ Conditional ]
    [statement list]
[ElseIf  ......]
[Else......]
EndIf
```

1. [ Conditional ]

变量类型：X、SX、Y、SY、SM、M、T、C、GI、LI、GR、LR、UI

解释：用于判断的条件。

2. [statement list]

解释：执行的语句或指令。

3. [ElseIf ……]

解释：在IF判断语句中，当上一个条件不满足时执行的下一段判断的指令。

4. [Else……]

解释：在IF判断语句中，当判断条件不满足时，执行的下一断语句或指令。

### 3.1.5 详细举例

例1：

```
If  GI0 > 100
    Dout Y0=ON
ElseIf  GI0 < 0
    Dout Y1=OFF
EndIf
```

当满足GI0信号的值大于100时，输出Y0状态为ON，If判断语句结束。如果GI0小于或等于100则判断GI0是否小于0，如果GI0小于0则输出Y1为OFF，ElseIf判断语句结束。如果GI0大于等于0，则语句结束。

例2：

```
If  GI0 > 100
    Dout Y0=ON
ElseIf  GI0 < 0
    Dout Y1=OFF
Else
    Dout Y2=OFF
EndIf
```

当满足GI0信号的值大于100时，输出Y0状态为ON，If判断语句结束。如果GI0小于或等于100则判断GI0是否小于0，如果GI0小于0则输出Y1为OFF，If判断语句结束。如果GI0大于等于0，则输出Y2为OFF。

## 3.2 While---条件循环

### 3.2.1 指令用法

判断条件是否满足，重复执行一些指令。

### 3.2.2 应用举例

例1：

```
While X0==ON
    GI0 = GI0 + 2;
EndWhile
```

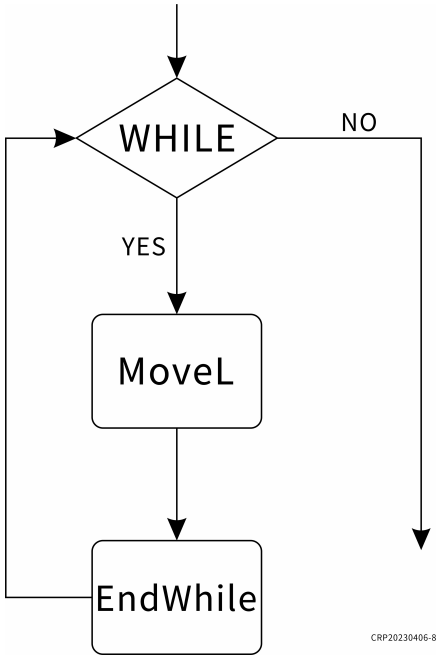



图 3.2.1

当满足X0信号的值为ON时，重复执行While内GI0加2后将值赋值给GI0的语句。当X0信号的值不为ON时循环结束。

例2：

```
While  GI0 < 10
    GI0 = GI0 + 2
```

```
EndWhile
```

当满足GI0小于10时，重复执行While内GI0加2后将值赋值给GI0的语句。当GI0大于等于10时循环结束。

### 3.2.3 程序运行

程序执行时将循环判断条件表达式的值是否满足，满足后就执行While 循环内的指令或表达式，如果不满足则不执行。

### 3.2.4 可变元素

```
While  [ Conditional ]
    [statement list]
EndWhile
```

1. [ Conditional ]

变量类型：X、SX、Y、SY、SM、M、T、C、GI、LI、GR、LR、UI

解释：用于判断的条件。

2. [statement list]

解释：执行的语句或指令。

### 3.2.5 详细举例

例1：

```
While [conditional expression]
   [statement list]
EndWhile
```

在While 指令块中当条件表达式满足时就执行指令块中的指令表或语句表达式，如果指令块中条件表达式不满足，则直接执行指令块后的程序。

## 3.3 Switch---条件选择

### 3.3.1 指令用法

根据表达式或数据的值，选择执行不同指令表或语句表达式。

### 3.3.2 应用举例

例1：

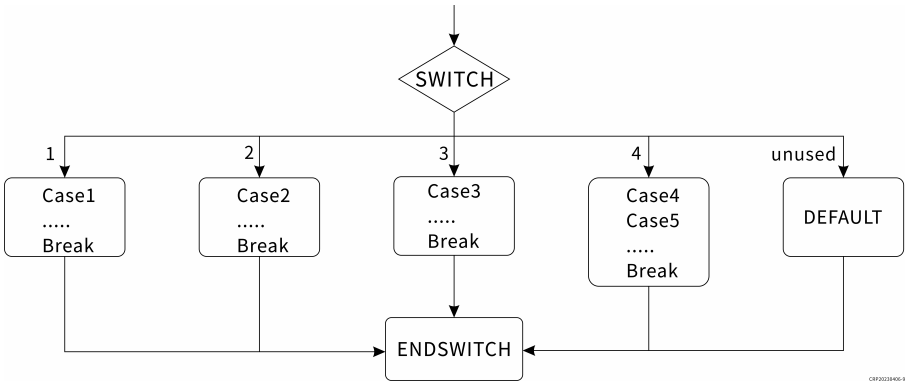



图 3.3.1

```
Switch  GI0
CASE 1
    Dout  Y0=ON
    Break
CASE 2
   Dout  Y1=ON
    Break
EndSwitch
```

当GI0的值等于1时执行Cas1的指令，输出Y0为NO，执行完成后直接执行EndSwitch ，指令块执行结束。当GI0的值等于2时执行Cas2的指令，输出Y1为NO，执行完成后直接执行EndSwitch ，指令块执行结束。

例2：

```
Switch  GI0
CASE 1
CASE 2
    Dout  Y0=ON
Break
Default
    Dout  Y1=ON
EndSwitch
```

当GI0的值等于1或2时都执行，输出Y0为ON，执行完成后直接执行EndSwitch，指令块执行结束。如果GI0不等于1或2，则执行输出Y1为ON，行完成后直接执行EndSwitch，指令块执行结束。

### 3.3.3 程序运行

执行指令块时将对比Switch 中的数据是否与Case条件中的值一致，如果一致则执行相关Case后的指令表或语句表达式，执行完成后通过Break 指令，跳转到程序块中EndSwitch位置，结束指令块执行，继续执下一行程序，如果Case条件后不带Break 指令，指令块将会继续执行下一个Case分支。

如果Switch 中的数据是否与Case条件中的不一致，如果指令块中含有Default语句，则执行Default后的指令表或语句表达式，执行完成后通过EndSwitch 指令，继续执行程序。若没有Default语句，则直接执行EndSwitch 指令，继续执行程序。

### 3.3.4 可变元素

```
Switch   [Test]
   CASE   [Test]
   [\Break]
   [\Default]
```

```
EndSwitch
```

1.[Test]

变量类型：UI、C

解释：用于调用程序块中分支的变量。

2. CASE [Test]

解释：指令块中被调用的程序分支，可以任意添加删减。Switch指令块中至少包含一个Case分支。

3. [\Break]

解释：直接到达EndSwitch指令。Break不是必须存在指令块中。如果指令中无Break，一旦case相应的值成功，那么将会无条件的继续向下执行其它case中的语句，直到遇到break语句或者到达Endswitch指令，才会结束条件选择语句。

4. [\Default]

解释：Default不是必须存在指令块中若所有case条件都不满足条件，则执行default，如果后面有case语句并执行default语句之后的case语句，直到break或Endswitch指令。

### 3.3.5 详细举例

例1：

```
Switch  GI0
CASE 1
CASE 2
    Dout  Y0=ON
Break
Default
    Dout  Y1=ON
CASE 3
```

```
    Dout  Y3=ON
EndSwitch
```

当GI0的值等于1或2时都执行，输出Y0为ON，执行完成后直接执行EndSwitch；当GI0值等于3时则执行输出Y3为ON，执行完成后直接执行EndSwitch 指令块执行结束。如果GI0不等于1或2或3时，则执行输出Y1为ON，执行完成后继续执行CASE 3指令分支输出Y3为ON，执行完成后直接执行EndSwitch 指令块执行结束。

例2：

```
Switch  GI0
CASE 1
CASE 2
    Dout  Y0=ON
Break
Default
    Dout  Y1=ON
Break
CASE 3
    Dout  Y3=ON
EndSwitch
```

当GI0的值等于1或2时都执行，输出Y0为ON，执行完成后直接执行EndSwitch，当GI值等于3时则执行输出Y3为ON，执行完成后直接执行EndSwitch 指令块执行结束。如果GI0不等于1或2或3时，则执行输出Y1为ON，执行完成后通过Break指令，直接执行EndSwitch 指令块执行结束。

## 3.4 Time ---延时

### 3.4.1 指令用法

程序延时等待。

### 3.4.2 应用举例

例1：

```
Time 100
```

执行到该行程序时程序延时等待100ms后继续向下执行。

### 3.4.3 程序运行

程序运行时，当程序行执行到该行程序后，程序开始延时等待，直到到达指令设定的时间后，程序继续向下执行。

### 3.4.4 可变元素

```
Time  [Time]
```

1. [time]

变量类型：C、UI

解释：延时等待设定的时间

### 3.4.5 详细举例

例1：

```
Time UI0
```

执行到该行程序时程序延时等待UI0后继续向下执行。延时等待时间储存在变量UI0中。

## 3.5 Pause---暂停

### 3.5.1 指令用法

用于暂停正在运行的程序，分条件暂停和直接暂停。

### 3.5.2 应用举例

例1：

```
.....
Pause
```

执行到该行程序时程序暂停运行。

### 3.5.3 程序运行

程序运行时，当程序行执行到该行程序后，直接暂停程序运行。如果指令后跟随判断条件，若条件满足则程序暂停运行，若条件不满足则继续运行程序，不执行暂停指令。

### 3.5.4 可变元素

```
Pause  [\verdict]
```

1. [\verdict]

可选类型：If

解释：用于判断是否执行暂停指令

### 3.5.5 详细举例

例1：

Pause If X0==ON

执行到该行程序时如果X0的值为ON则程序暂停运行、否则程序继续向下执行。

例2：

Pause If GI0 > 5

执行到该行程序时如果GI0的值大于5则程序暂停运行、否则程序继续向下执行。

## 3.6 Jump---跳转

### 3.6.1 指令用法

用于程序中的跳转，分条件跳转和直接跳转。使用跳转指令时必须和“*”跳转标识符指令配套使用

### 3.6.2 应用举例

例1：

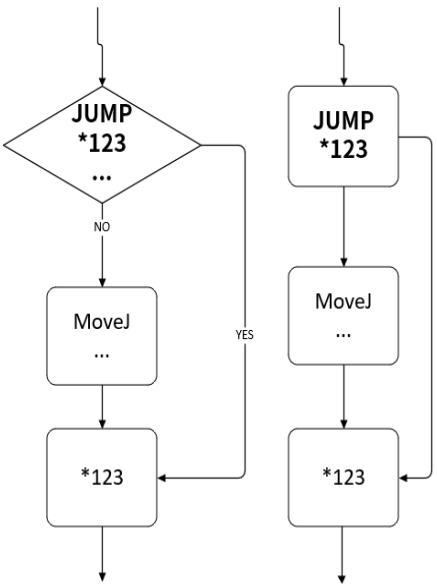



图 3.6.1

.....

Jump * ABC

.....

* ABC

程序执行到该指令行时，直接跳转到带有*ABC的跳转目标标识符处继续向下执行程序。

### 3.6.3 程序运行

程序运行时，当程序行执行到该行程序后，直接跳转到跳转目标标识符处继续执行程序。如果指令后跟随判断条件，若条件满足则程序跳转，若条件不满足则继续运行程序，不执行跳转指令。

### 3.6.4 可变元素

Jump [target] [\verdict]

[target]

变量类型：float（数字，字母，汉字，最大15个汉字）

解释：跳转目标标识符。必须和“*”的类型一致

[\verdict]

可选类型：If

解释：用于判断是否执行暂停指令。

### 3.6.5 详细举例

例1：

```
.....
Jump  *ABC   If X0==ON
.....
*ABC
```

执行到该行程序时如果X0的值为ON则程序跳转到带有*ABC 跳转目标标识符处继续向下执行程序。

例2：

```
.....
Jump  *ABC   If GI0 >5
.....
*ABC
```

执行到该行程序时如果GI0的值大于5则程序跳转到带有*ABC 跳转目标标识符处继续向下执行程序。

## 3.7 *---跳转目标

### 3.7.1 指令用法

用于程序跳转时的跳转目标，与Jump指令配合使用，详细信息参考Jump指令使用方法。

## 3.8 Call---子程序调用

### 3.8.1 指令用法

程序执行过程中，将程序执行转移到另一个程序中执行。分条件调用和直接调用。

### 3.8.2 应用举例

例1：

```
Call  ABC
```

执行到该行程序时直接调用程序名称为ABC程序继续执行。

### 3.8.3 程序运行

程序运行时，当程序行执行到该行程序后，直接调用该程序行指定的程序并继续执行。如果指令后跟随判断条件，若条件满足则执行程序调用，若条件不满足则继续运行程序，不执行程序调用。

### 3.8.4 可变元素

```
Call   [Subroutine]  [\verdict]
```

1. [Subroutine]

解释：被调用的程序。

2. [\verdict]

可选类型：If

解释：用于判断是否执行暂停指令

### 3.8.5 详细举例

例1：

```
Call  ABC  If  X0==ON
```

执行到该行程序时，如果X0的值为ON则调用程序名称为ABC程序。

例2：

```
Call ABC  If GI0  >  5
```

执行到该行程序时，如果GI0的值大于5则调用程序名称为ABC程序。

## 3.9 Return---程序返回

### 3.9.1 指令用法

结束子程序运行，返回主程序。分为条件程序返回和直接返回。

### 3.9.2 应用举例

例1：

```
Return
```

执行到该行程序时，直接退出当前程序回到主程序继续运行。

### 3.9.3 程序运行

当程序行执行到该行程序后，直接退出子程序并回到主程序继续执行。如果指令后跟随判断条件，若条件满足则执行子程序返回，若条件不满足则继续运行程序，不执行子程序返回。

### 3.9.4 可变元素

```
Return  [\verdict]
```

1. [\verdict]

可选类型：If

解释：用于判断是否执行程序返回。

### 3.9.5 详细举例

例1：

```
Return  If  X0==ON
```

执行到该行程序时，如果X0的值为ON则结束当前程序并返回到主程序继续运行。

例2：

```
Return  If  GI0 > 5
```

执行到该行程序时，如果GI0的值大于5则结束当前程序并返回到主程序继续运行。

# 四、运算指令

## 4.1 ADD---加运算

### 4.1.1 指令用法

用于数值、变量或者永久数据对象的加法运算。

### 4.1.2 应用举例

例1：

Add OP1=OP2+OP3

将OP3的值增加OP2计算完成后并赋值给OP1

例2：

Add GI1=GI2+5

将5增加到GI2中计算完成并赋值给GI1

例3：

Add OP0.X=OP1.X+5

将OP1的X方向坐标值增加5计算完成后，并赋值给OP0的X方向坐标

### 4.1.3 程序运行

执行此程序行，对应的变量进行加运算，运算完成后并赋值到指定变量中储存。当数据类型不一致时，不能进行运算。

### 4.1.4 可变元素

```
Add  [Name}=[AddValue]+[AddDvalue]
```

1. [Name]

数据类型：num

解释：数值变量或者永久数据

2. [AddValue]

数据类型：num

解释：数值变量或者永久数据

3. [AddDvalue]

数据类型：num

解释：数值变量或者永久数据

### 4.1.5 详细举例

例1：

```
Add   GP(UI1).J6=LP3.J6+UI2
```

将局部位置变量LP3的J6轴方向坐标值增加变量UI2中的值。运算完成后将值赋值给全局位置变量GP（UI1）的J6轴方向坐标。GP号为变量UI1的值。

例2：

```
Add   LP(UI1).Z=LP(UI2).Z+GI(UI3)
```

将局部变量LP（UI2）的Z方向坐标值增加GI（UI3）运算完成后赋值给变量LP（UI1）的Z方向坐标。局部变量LP的变量号分别为变量UI1、UI2的值，全局变量GI的变量号为变量UI3的值。

例3：

```
Add  GJ5.J6=GJ5.J6+UI3
```

将机器人在GJ5位置J6轴的关节坐标值增加UI3，运算完成后将值赋值给GJ5的J6轴的关节坐标。增加的数据为变量UI3的值。

## 4.2 Sub---减运算

### 4.2.1 指令用法

用于数值变量或者永久数据对象减一个数值。

### 4.2.2 应用举例

例1：

```
Sub OP1=OP2-OP3
```

将OP2的值减去OP3计算完成后并赋值给OP1

例2：

```
Sub  GI1=GI2-5
```

将GI2的值减去5，计算完成后并赋值给GI1

例3：

```
Sub  OP0.X=OP1.X-5
```

将OP1的X方向坐标值减去5，计算完成后并赋值给OP0的X方向坐标。

### 4.2.3 程序运行

执行此程序行，对应的变量进行减运算，运算完成后并赋值到指定变量中储存。当数据类型不一致时，不能进行运算。

### 4.2.4 可变元素

Sub [Name}=[SubValue]-[SubDvalue]

1. [Name]

数据类型：num

数值变量或者永久数据

2. [SubValue]

数据类型：num

解释：数值变量或者永久数据

3. [SubDvalue]

数据类型：num

解释：数值变量或者永久数据

### 4.2.5 详细举例

例1：

Sub GP(UI1).J6=LP3.J6-UI2

将局部变量LP3的J6轴的坐标值减变量UI2的值。运算完成后将值赋值给全局位置变量GP（UI1）的J6轴的坐标。GP号为变量UI1的值。

例2：

Sub LP(UI1).Z=LP(UI2).Z-GI(UI3)

将局部变量LP（UI2）的Z方向的坐标值减少GI（UI3）运算完成后赋值给LP（UI1）的Z方向坐标。局部变量LP的变量号分别为变量UI1、UI2的值，全局变量GI的变量号为变量UI3的值。

例3：

Sub GJ5.J6=GJ5.J6-UI3

将机器人在GJ5位置J6轴的关节坐标值减少UI3，运算完成后将值赋值给GJ5位置J6轴的关节坐标。减少的数据存储UI3中。

## 4.3 Mul---乘运算

### 4.3.1 指令用法

用于数值变量或者永久数据对象乘以一个数值。

### 4.3.2 应用举例

例1：

Mul GI1=GI2*5

将GI2的值乘以5计算完成后并赋值给GI1

例2：

Mul LI(UI1)=GI3*GI(UI0)

将GI3的值乘以GI（UI0）的值，计算完成后并赋值给LI（UI1）。局部变量LI的变量号为变量UI1的值，全局变量GI的变量号为变量UI0的值。

### 4.3.3 程序运行

执行此程序行，对应的变量进行乘法运算，运算完成后并赋值到指定变量中储存。当数据类型不一致时，不能进行运算。

### 4.3.4 可变元素

Mul [Name}=[MulValue]*[MulDvalue]

1. [Name]

数据类型：num

解释：数值变量或者永久数据

2. [MulValue]

数据类型：num

解释：数值变量或者永久数据

3. [MulDvalue]

数据类型：num

解释：数值变量或者永久数据

#### 4.3.5 详细举例

例1：

```
Mul GR(UI2)=LR(UI5)*GI(UI4)
```

将变量LR（UI5）的值乘以GI（UI4）的值，计算完成后将值赋值给GR（UI2）。变量GR的变量号为变量UI2的值，变量LR的变量号为变量UI5的值，变量GI的变量号为UI4的值。

### 4.4 Div---除运算

#### 4.4.1 指令用法

用于数值变量或者永久数据对象除以一个数值。

#### 4.4.2 应用举例

例1：

```
Div GI1=GI2/5
```

将GI2的值除以5计算完成后并赋值给GI1。

例2：

Div LI(UI1)=GI3/GI(UI0)

将GI3的值除以GI（UI0）的值，计算完成后并赋值给LI（UI1）。局部变量LI的变量号为变量UI1的值，全局变量GI的变量号为变量UI0的值。

### 4.4.3 程序运行

执行此程序行，对应的变量进行除法运算，运算完成后并赋值到指定变量中储存。当数据类型不一致时，不能进行运算。

### 4.4.4 可变元素

Div [Name}=[DivValue]/[DivDvalue]

1. [Name]

数据类型：num

解释：数值变量或者永久数据

2. [DivValue]

数据类型：num

解释：数值变量或者永久数据

[DivDvalue]

数据类型：num

解释：数值变量或者永久数据

### 4.4.5 详细举例

例1：

```
Div GR(UI2)=LR(UI5)/GI(UI4)
```

将变量LR（UI5）的值除以GI（UI4）的值，计算完成后将值赋值给GR（UI2）。变量GR的变量号为变量UI2的值，变量LR的变量号为变量UI5的值，变量GI的变量号为变量UI4的值。

## 4.5 Inc---加一运算

### 4.5.1 指令用法

用于数值变量对象加一运算。

### 4.5.2 应用举例

例1：

```
Inc GI1
```

将GI1的值进行加一计算完成后并赋值给GI1，与Add GI1=GI1+1用法一致。

### 4.5.3 程序运行

执行此程序行，对指定变量进行加一运算，运算完成后并赋值到该变量中储存。

### 4.5.4 可变元素

IN[Name]

[Name]

数据类型：num

解释：数值变量

### 4.5.5 详细举例

例1：

Inc LI(UI0)

将LI(UI0）的值进行加一计算完成后并赋值给LI(UI0），与Add LI(UI0）=LI(UI0）+1用法一致。变量LI的变量号为变量UI0的值。

## 4.6 Dec---减一运算

### 4.6.1 指令用法

用于数值变量对象减一运算。

### 4.6.2 应用举例

例1：

Dec GI1

将GI1的值进行减一计算完成后并赋值给GI1，与Sub GI1=GI1-1用法一致。

### 4.6.3 程序运行

执行此程序行，对指定变量进行减一运算，运算完成后并赋值到该变量中储存。

### 4.6.4 可变元素

Dec [Name]

1. [Name]

数据类型：num

解释：数值变量

### 4.6.5 详细举例

例1：

Dec LI(UI0)

将LI(UI0）的值进行减一计算完成后并赋值给LI(UI0），与Sub LI(UI0）=LI(UI0）+1用法一致。变量LI的变量号为变量UI0的值。

## 4.7 Set---赋值指令

### 4.7.1 指令用法

用于数值变量或者永久数据对象赋值运算。

### 4.7.2 应用举例

例1：

Set OP1=OP2

将OP2的值保存到OP1中并将OP1原来的数值覆盖。

例1：

Set LP2.X=100

将100保存到局部位置变量LP的X方向数值中，并将原来LP中X方向的值覆盖。

### 4.7.3 程序运行

执行此程序行，将“=”符号右侧的数据保存到“=”左侧并将原来的数据覆盖。

### 4.7.4 可变元素

Set [Name]=[SetValue]

1. [Name]

变量类型：num

解释：数值变量

2. [SetValue]

变量类型：num

数值变量或者永久数据

#### 4.7.5 详细举例

例1：

Set OP1.X=50

将50保存到全局位置变量OP的X方向的坐标数值中，并将原来OP中X方向的坐标值覆盖。

例2：

Set GP(UI2).Y=GP(UI3).Y

将全局位置变量GP（UI3）中的Y值，保存到全局位置变量GP（UI2）的Y值中并将原来的值覆盖。

## 五、焊接指令

### 5.1 ArcOn---起弧

#### 5.1.1 指令用法

用于弧焊起弧。需要与ArcOff指令配合使用。

#### 5.1.2 应用举例

例1：

ArcOn （0）

调用弧焊工艺0号参数文件。

例2：

ArcOn （0）Vl=10

调用弧焊工艺0号参数文件，弧焊速度为10mm/s。

### 5.1.3 程序运行

执行此程序行，机器人系统将会调用既定的焊接工艺号，并使用指令中设置的相关参数。

### 5.1.4 可变元素

ArcOn [Process] [\Speed] [\Current 、Voltage] [\Pattern] [\Cataed] [\La][\MArc] [\RArc]

1. [process]

数据类型：num

解释：数值变量或者永久数据。

2. [\Speed]

数据类型：num

解释: 焊接速度。

3. [\Current 、Voltage]

数据类型：num（I：0-160 V：0-20）

解释: 焊接电流电压。

4. [\Pattern]

变量类型：Int（0-99）

解释: 焊接模式包含焊接模式Mode和调用的工作JOB。

5. [\Cataed]

数据类型：Int（0-9999）

解释: 鱼鳞焊，包含时间T，间距L1和空行L2数据。

6. [\La]

数据类型：Int（-1，0，1）

解释: 搭接，-1为关闭、0为保持、1为开启。

7. [\MArc]

数据类型：Int（-1，0，1）

解释: 多次起弧，-1为关闭、0为保持、1为开启。

8. [\RArc]

数据类型：Int（-1，0，1）

解释: 焊接状态，-1为关闭、0为保持、1为开启。

### 5.1.5 详细举例

例1：

```
MoveL  VL=100  PL=9  T=1

ArcOn(1)  VL=10  I=160  V=20
```

机器运行到指定点后，机器人弧焊起弧并调用焊接工艺号1，机器人按设定的焊接速度为10mm/s、电流为160A、电压为20V开始焊接。

例2：

```
MoveL  VL=100  PL=9  T=1

ArcOn(UI1)  VL=10  I=160  V=20  Mode=2  Job=3  La=1
```

机器运行到指定点后，机器人弧焊起弧并调用UI1，变量UI1的值为焊接工艺号，机器人按设定的焊接速度为10mm/s、电流为160A、电压为20V、焊机控制模式为2、调用焊机工作号为3、搭接模式为开启，开始焊接。

例3：

```
MoveL  VL=100  PL=9  T=1

ArcOn(UI1)  VL=10  I=160  V=20  T=6  L1=10  L2=8  MArc=1  RArc=1
```

机器运行到指定点后，机器人弧焊起弧并调用UI1，变量UI1的值为焊接工艺号，机器人按设定的焊接速度为10mm/s、电流为160A、电压为20V、鱼鳞焊开启、焊接时间为6000ms、间距为10mm、空行距离为8mm、多次起弧开启、显示焊接状态开启，开始焊接。

## 5.2 ArcOff---灭弧

### 5.2.1 指令用法

用于弧焊灭弧。需要与ArcOn指令配合使用。

### 5.2.2 应用举例

例1：

ArcOff

机器人灭弧。

例2：

ArcOff I=160 V=20

机器人执行灭弧，灭弧电流电压渐变为160A， 20V。

### 5.2.3 程序运行

执行此程序行，机器人系统将会灭弧，灭弧参数将会从工艺号中的参数渐变为指令中的设定参数。

### 5.2.4 可变元素

ArcOff [\Current 、Voltage] [\ArcSA] [\ArcST] [\ArcSD] [\WSD]

[\Current 、Voltage]

数据类型：num（I：0-160 V：0-20）

解释:灭弧时从焊接工艺中的电流电压渐变至灭弧时的电流电压。

[\ArcSA]

数据类型：float（0-10）

解释:收弧衰减，ArcSA1为正常衰减、ArcSA2为提前衰减。

[\ArcST]

数据类型：Int（1-5）

解释:收弧次数。

[\ArcSD]

数据类型：Int（-1，0，1）

解释:灭弧检测，-1为关闭、0为保持、1为开启。

[\WSD]

数据类型：Int（-1，0，1）

解释:防粘丝，-1为关闭、0为保持、1为开启。

### 5.2.5 详细举例

例1：

MoveL VL=100 PL=9 T=1

ArcOff I=160 V=20

机器运行到指定点后，机器人执行灭弧，灭弧电压电流从工艺号中的电流电压渐变为指令中设置的160A 、20V。

例2：

MoveL VL=100 PL=9 T=1

ArcOff I=160 V=20 ArcSA1=0.5 ArcST=2

机器运行到指定点后，机器人执行灭弧，灭弧电压电流从工艺号中的电流电压渐变为指令中设置的160A、20V，正常收弧衰减参数为0.5、灭弧次数为2次。

例3：

MoveL VL=100 PL=9 T=1

ArcOff(UI0) I=160 V=20 ArcST=2 ArcSD=1 WSD=1

机器运行到指定点后，机器人灭弧并关闭调用UI0，变量UI0的值为焊接工艺号，灭弧电压电流从工艺号中的电流电压渐变为指令中设置的160A、20V，灭弧检测开启，防粘丝功能开启。

## 5.3 WeaveOn--摆弧开启

### 5.3.1 指令用法

用于弧焊时开启并调用摆弧工艺。需要与WeaveOff指令配合使用。

### 5.3.2 应用举例

例1：

WeaveOn（0）

开启并调用弧焊摆弧工艺0。

例2：

WeaveOn（0）F=2 R=3.5

开启并调用摆弧工艺0，摆动频率为2，摆动幅度为3.5。

### 5.3.3 程序运行

执行此程序行，机器人系统将会开启并调用指定的摆弧工艺，并按照指令中设置的参数进行摆弧。

### 5.3.4 可变元素

WeaveOn [Process] [\F] [\R] [Mode] [refer]

1. [process]

数据类型：num

解释：数值变量或者永久数据。

2. [\F]

数据类型：float（1-20）

解释:摆动频率。

3. [\R]

数据类型：float（0-100）

解释:摆动幅度。

4. [Mode]

数据类型：Int（0，1）

解释:变化模式（0：渐变，1：突变）。

5. [refer]

变量类型：Tra、Tool、Use

解释:参考坐标的坐标方向。

### 5.3.5 详细举例

例1：

ArcOn （0）.....

WeaveOn（0）F=2 R=3.5

机器开启弧焊后，运行到指令行系统开启并调用摆弧工艺0，摆动频率为2，摆动幅度为3.5。

例2：

ArcOn （0）.....

WeaveOn（UI0）F=LR0 R=3.5 Tool=X

机器开启弧焊后，运行到指令行系统开启并调UI0，变量UI0的值为焊接摆弧工艺号，摆动频率为储存在LR0中的数据，摆动幅度为3.5，参考坐标为工具坐标的X方向。

## 5.4 WeaveOff--摆弧关闭

### 5.4.1 指令用法

用于弧焊时关闭摆弧工艺。需要与WeaveOn指令配合使用。

### 5.4.2 应用举例

例1：

WeaveOff

关闭弧焊摆弧工艺。

例2：

WeaveOff F=2 R=3.5

关闭弧焊摆弧工艺，摆动频率从工艺相应中渐变为2，摆动幅度从相应工艺中渐变为3.5。

### 5.4.3 程序运行

执行此程序行，机器人系统将会关闭摆弧工艺，并将工艺号中的参数渐变成指令中设置的参数。

### 5.4.4 可变元素

WeaveOff [\F] [\R]

[\F]

变量类型：float（0-10）

解释:摆动频率，单位Hz。

[\R]

变量类型：float（0-100）

解释:摆动幅度，单位mm。

### 5.4.5 详细举例

例1：

WeaveOff F=LR2 R=3.5

关闭弧焊摆弧工艺，摆动频率从工艺中的相应的摆动参数，渐变为变量LR2的值，摆动幅度从相应工艺中的参数渐变为3.5。

例2：

WeaveOff F=LR2 R=GR1

关闭弧焊摆弧工艺，摆动频率从工艺相应中的参数渐变为储存在变量LR2中的值，摆动幅度从相应工艺中的参数渐变为储存在变量GR1中的值。

# 六、特殊指令

## 6.1 AxisAct---轴禁止开启

### 6.1.1 指令用法

用于禁止机器人或机械单元组的指定轴。轴禁止后需要轴禁止结束指令解除轴禁止。

### 6.1.2 应用举例

例1：

AxisAct B1 S1

运行该行程序时，机器人系统将禁止机械单元B1的S1轴移动。

### 6.1.3 程序运行

执行此程序行，机器人系统将会禁止指令中设置的轴，需要使用轴禁止解除指令解除相应被禁止的轴。

### 6.1.4 可变元素

AxisAct [B] [E]

1. [B]

变量类型：B(机械单元)

解释：机器人系统在添加机械单元时配置的机械单元。

2. [E]

变量类型：E(轴)

解释：机器人系统在添加机械单元时，机械单元内的轴，指令中可以禁止多个轴。

### 6.1.5 详细举例

例1：

AxisAct B1 E1

运行该行程序时，机器人系统将禁止机械单元B1中的E1轴移动。

## 6.2 AxisDeact---轴禁止解除

### 6.2.1 指令用法

用于解除机器人或机械单元组被轴禁止指令禁止的轴组。

### 6.2.2 应用举例

例1：

AxisDeact B1 E1

远行完成指令后将解除机械单元B1中的E1轴轴禁止。

### 6.2.3 程序运行

执行此程序行，机器人系统将会解除指令中设置相应的轴组。

### 6.2.4 可变元素

AxisDeact [B] [E]

1. [B]

变量类型：B(机械单元)

解释：机器人系统在添加机械单元时配置的机械单元。

2. [E]

变量类型：E(轴)

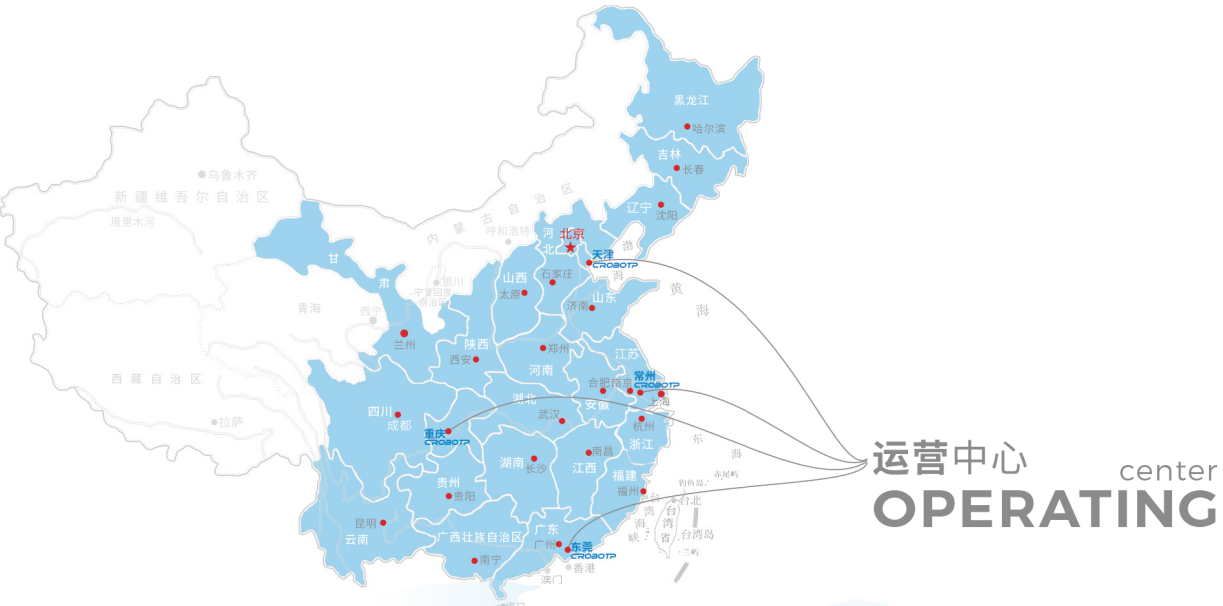
解释：机器人系统在添加机械单元时，机械单元内的轴，指令中可以解除被禁止多个轴。

### 6.2.5 详细举例

例1：

AxisDeact B1 E1 E2

运行该行程序时，机器人系统将解除被禁止的机械单元B1中的E1和E2轴。




微信公众号　抖音号　资料下载

**成都卡诺普机器人技术股份有限公司**
**CHENGDU CRP ROBOT TECHNOLOGY CO.,LTD**

400-668-8633
crobotp@crprobot.com
www.crprobot.com
**四川成都市成华区华月路188号**