



## 运动控制

- ◆ 机器人运动控制
- ◆ 搜索功能
- ◆ 指定位置触发信号与中断功能
- ◆ 出错或中断时的运动控制
- ◆ 外轴的控制
- ◆ 独立轴的控制
- ◆ 路径修正功能
- ◆ 路径记录功能
- ◆ 输送链跟踪功能
- ◆ 传感器同步功能
- ◆ 有效载荷和碰撞检测
- ◆ 关于位置的功能



## 1、机器人运动控制

| 指令        | 说明                 |
|-----------|--------------------|
| MoveC     | TCP圆弧运动            |
| MoveJ     | 关节运动               |
| MoveL     | TCP线性运动            |
| MoveAbsJ  | 轴绝对角度位置运动          |
| MoveExtJ  | 外部直线轴和旋转轴运动        |
| MoveCDO   | TCP圆弧运动的同时触发一个输出信号 |
| MoveJDO   | 关节运动的同时触发一个输出信号    |
| MoveLDO   | TCP线性运动的同时触发一个输出信号 |
| MoveCSync | TCP圆弧运动的同时执行一个例行程序 |
| MoveJSync | 关节运动的同时执行一个例行程序    |
| MoveLSync | TCP线性运动的同时执行一个例行程序 |



## (1) MoveCDO

MoveCDO CiPoint,ToPoint,Speed[\T],Zone,Tool [\Wobj], Signal,Value;

**CiPoint :** (robtargt)

中间点, 默认为\*

**ToPoint :** (robtargt)

目标点, 默认为\*

**Speed :** (speeddata)

运行速度

数据 [\T] : (num)

运行时间

控制s **Zone :** (zonedata)

运行转角

数据 **Tool :** (tooldata)

工具中心

点 (TCP) **[\Wobj] :** (wlbjdata)

工件坐标

系 **Signal :** (signaldo)

数字输出

信号名称 **Value :** (dionum)

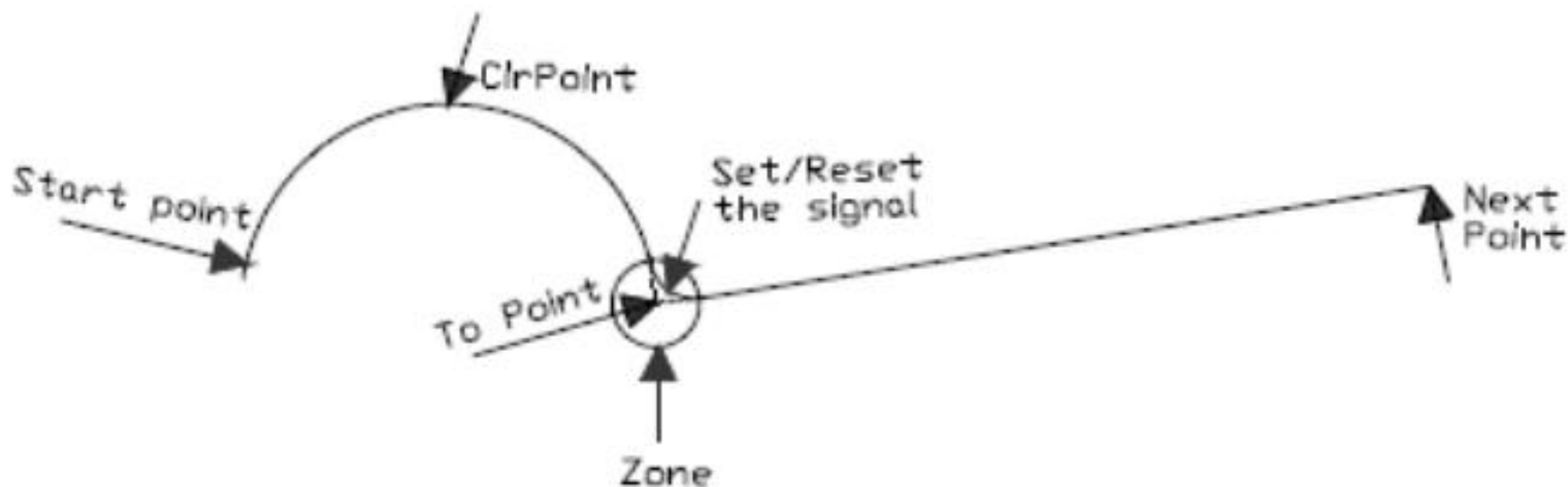
数字输出

信号值



## 应用：

机器人通过中间点以圆弧移动方式运动至目标点，并且在目标点将相应输出信号设置为相应值，在指令 MoveC 基础上增加信号输出功能。





## (2) MoveJDO

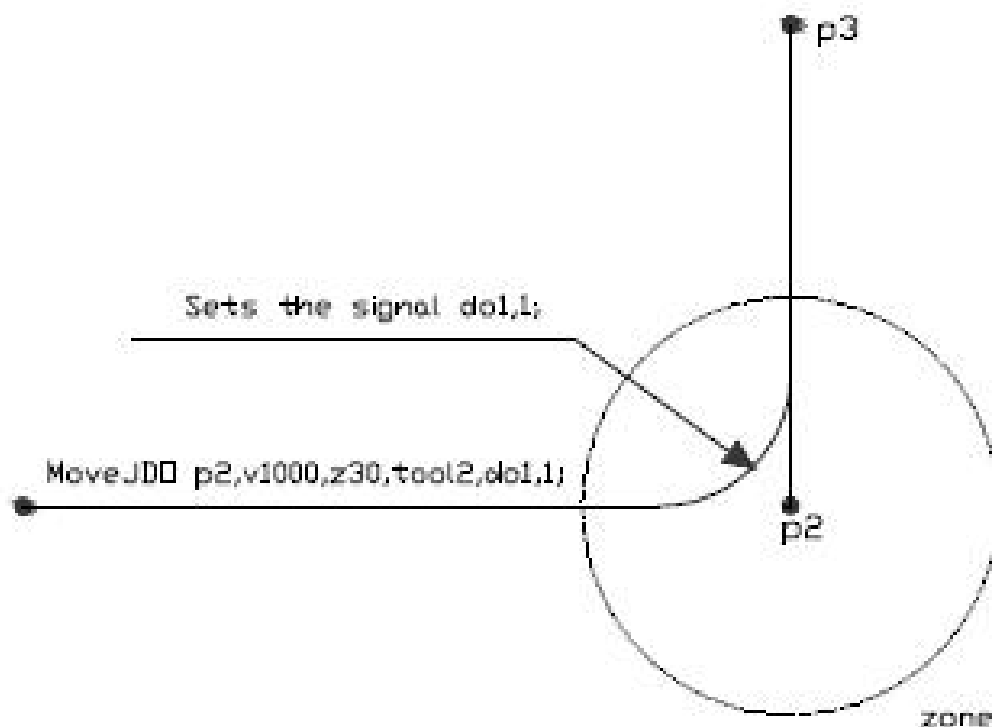
MoveJDO ToPoint,Speed [\T],Zone,Tool[\Wobj], Signal,Value;

|                                 |             |
|---------------------------------|-------------|
| <b>ToPoint :</b><br>(robtarget) | 目标点，默认为*    |
| <b>Speed:</b><br>(speeddata)    | 运行速度数据。     |
| <b>[\T]:</b><br>(num)           | 运行时间控制s     |
| <b>Zone:</b><br>(zonedata)      | 运行转角数据      |
| <b>Tool:</b><br>(tooldata)      | 工具中心点 (TCP) |
| <b>[\Wobj]:</b><br>(wlbjdata)   | 工件坐标系       |
| <b>Signal:</b><br>(signaldo)    | 数字输出信号名称    |
| <b>Value:</b><br>(dionum)       | 数字输出信号值     |



## 应用：

机器人以最快捷的方式运动至目标点，并且在目标点将相应输出设置为相应值，在指令MoveJ基础上增加信号输出功能。







## (3) MoveLDO

MoveLDO ToPoint,Speed [\T],Zone,Tool[\Wobj], Signal,Value;

ToPoint :

目标点，默认为\*

(robtarget)

Speed:

运行速度数据。

(speeddata)

[\T]:

运行时间控制s

(num)

Zone:

运行转角数据

(zonedata)

Tool:

工具中心点 (TCP)

(tooldata)

[\Wobj]:

工件坐标系

(wlbjdata)

Signal:

数字输出信号名称

(signaldo)

Value:

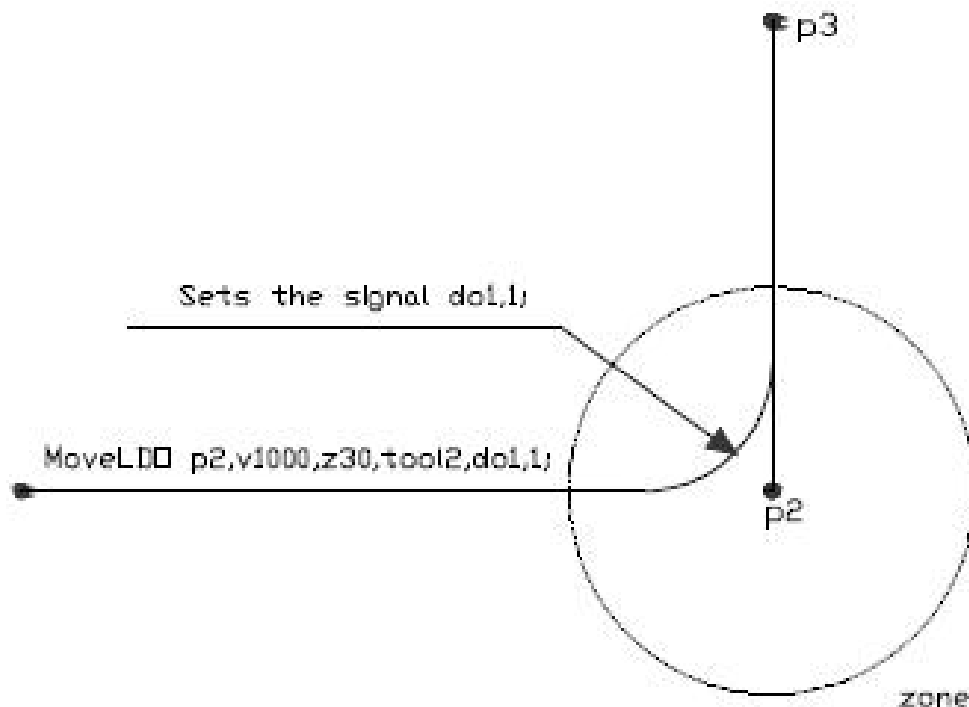
数字输出信号值

(dionum)



## 应用:

机器人以线性运动方式运动至目标点，并且在目标点将相应输出信号设置为相应值，在指令MoveL基础上增加信号输出功能。







## (4) MoveCSync

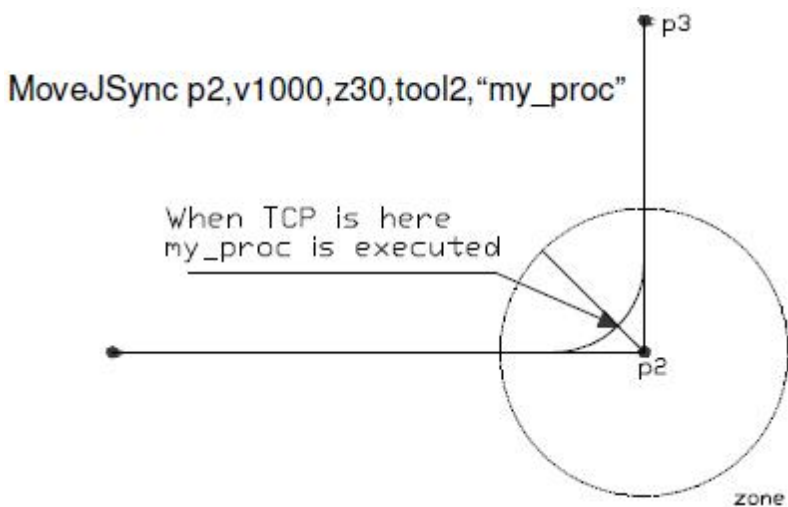
**MoveJSync ToPoint,Speed [\T], Tool Zone,[\Wobj], Proc;**

|                  |             |             |
|------------------|-------------|-------------|
| <b>ToPoint:</b>  | (robtarget) | 目标点, 默认为*   |
| <b>Speed:</b>    | (speeddata) | 运行速度数据      |
| <b>[\T] :</b>    | (num)       | 运行时间控制s     |
| <b>Zone :</b>    | (zonedata)  | 运行转角数据      |
| <b>Tool :</b>    | (tooldata)  | 工具中心点 (TCP) |
| <b>[\Wobj] :</b> | (wlbjdata)  | 工件坐标系       |
| <b>Proc:</b>     | (string)    | 例行程序名称      |



## 应用：

机器人以最快捷的方式运动至目标点，并且在目标点调用相应的例行程序，在指令MoveJ基础上增加例行程序调用功能。



## 限制：

用指令Stop停止当前指令运行，会出现一个错误信息，如需要避免，采用指令StopInstr。  
不能使用指令MoveJSync来采用中断处理程序TRAP。  
不能单步执行指令MoveJSync所调用的例行程序PROC。



## (5) MoveJSync

MoveLSync ToPoint,Speed [\T], Tool Zone,[\Wobj], Proc;

**ToPoint:**

目标点, 默认为\*

(robtarget)

**Speed:**

运行速度数据

(speeddata)

**[\T] :**

运行时间控制s

(num)

**Zone :**

运行转角数据

(zonedata)

**Tool :**

工具中心点 (TCP)

(tooldata)

**[\Wobj] :**

工件坐标系

(wlbjdata)

**Proc:**

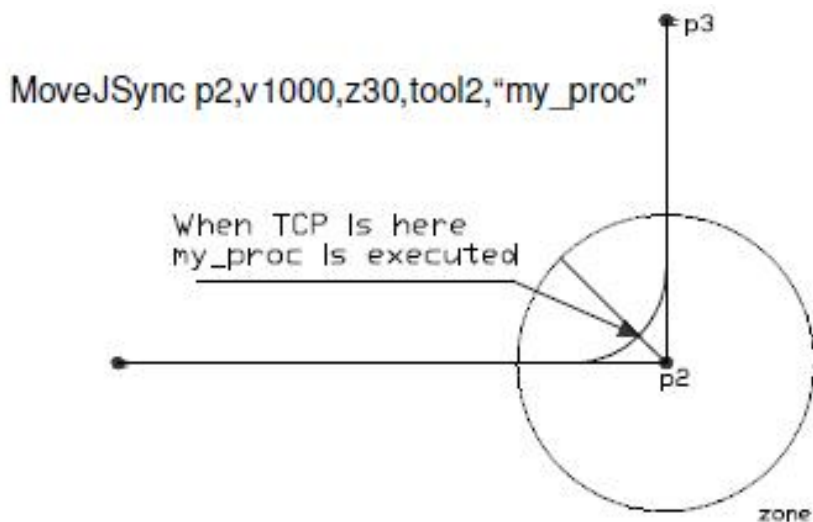
例行程序名称

(string)



## 应用：

机器人以线性的方式运动至目标点，并且在目标点调用相应的例行程序，在指令MoveL基础上增加例行程序调用功能。



## 限制：

用指令Stop停止当前指令运行，会出现一个错误信息，如需要避免，采用指令StopInstr。

不能使用指令MoveLSync来调用中断处理程序TRAP。

不能单步执行指令MoveLSync所调用的例行程序PROC。



## (6) MoveLSync

MoveLSync ToPoint,Speed [\T], Tool Zone,[\Wobj], Proc;

**ToPoint:**

目标点, 默认为\*

(robtarget)

**Speed:**

运行速度数据

(speeddata)

**[\T] :**

运行时间控制s

(num)

**Zone :**

运行转角数据

(zonedata)

**Tool :**

工具中心点 (TCP)

(tooldata)

**[\Wobj] :**

工件坐标系

(wlbjdata)

**Proc:**

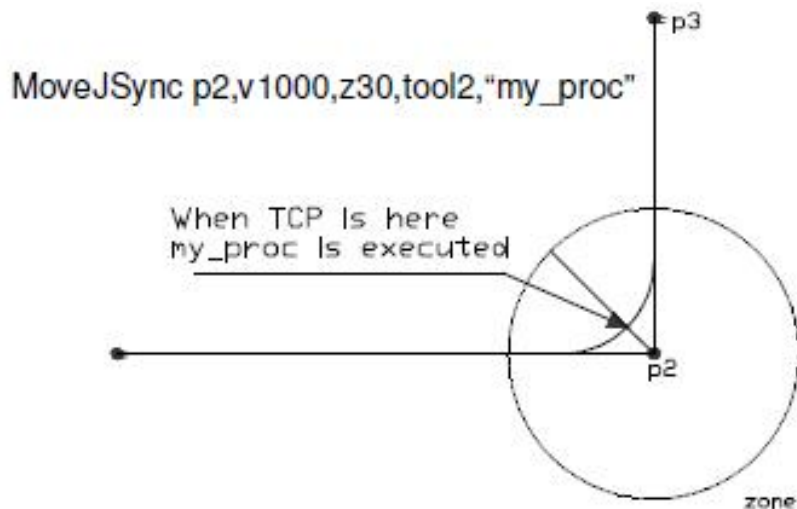
例行程序名称

(string)



## 应用：

机器人以线性的方式运动至目标点，并且在目标点调用相应的例行程序，在指令MoveL基础上增加例行程序调用功能。



## 限制：

用指令Stop停止当前指令运行，会出现一个错误信息，如需要避免，采用指令StopInstr。

不能使用指令MoveLSync来调用中断处理程序TRAP。

不能单步执行指令MoveLSync所调用的例行程序PROC。





## 2、搜索功能

| 指令         | 说明        |
|------------|-----------|
| SearchC    | TCP圆弧搜索功能 |
| SearchL    | TCP线性搜索功能 |
| SearchExtJ | 外轴搜索运动    |



### 3、指定位置触发信号与中断功能

| 指令            | 说明  |
|---------------|---|
| TriggIO       | 定义触发条件在一个指定的位置触发输出信号                        |
| TriggInt      | 定义触发条件在一个指定的位置触发中断程序                        |
| TriggCheckIO  | 定义一个指定的位置进行I/O状态的检查                         |
| TriggEquip    | 定义触发条件在一个指定的位置触发输出信号，并对信号响应的延迟进行补偿设定        |
| TriggRamoAO   | 定义触发条件在一个指定的位置触发模拟输出信号，并对信号响应的延迟进行补偿设定      |
| TriggC        | 带触发时间的圆弧运动                                  |
| TriggJ        | 带触发事件的关节运动                                  |
| TriggL        | 带触发事件的线性运动                                  |
| TriggLIOs     | 在一个指定的位置触发输出信号的线性运动                         |
| StepBwdPath   | 在RESTART的事件程序中进行路径的返回                       |
| TriggStopProc | 在系统中创建一个监控处理，用于在STOP和QSTOP中需要信号复位和程序数据复位的操作 |
| TriggSpeed    | 定义模拟输出信号与实际TCP速度之间的配合                       |



## (1) TriggIO

TriggIO  
TriggData,Distance[\Start][\Time][\Dop][\Gop][\Aop][\ProcID],SetValue[\DODelay

|                     |         |             |
|---------------------|---------|-------------|
| <b>[TriggData]:</b> | 触发变量名称。 | (triggdata) |
| <b>Distance:</b>    | 触发距离mm。 | (num)       |
| <b>[\Start]:</b>    | 触发起始开关。 | (switch)    |
| <b>[\Time]:</b>     | 时间触发开关。 | (switch)    |
| <b>[\Dop]:</b>      | 时间数字输出。 | (signaldo)  |
| <b>[\Gop] :</b>     | 触发组合输出。 | (signalgo)  |
| <b>[\Aop] :</b>     | 触发模拟输出。 | (signalao)  |
| <b>[\ProcID] :</b>  | 过程处理触发。 | (num)       |
| <b>SetValue :</b>   | 相应信号值。  | (num)       |
| <b>[\DODelay] :</b> | 数字输出延迟。 | (num)       |



## 应用:

机器人可以在运动时通过触发指令精确的输出相应信号，当前指令用于定义触发性质，此指令必须与其他触发指令TriggJ、TriggL或TriggC同时使用才有意义，同机器人指令TriggEquip比较，多了时间控制功能，少了外部设备触发延迟功能，通常用于喷涂、涂胶等行业。

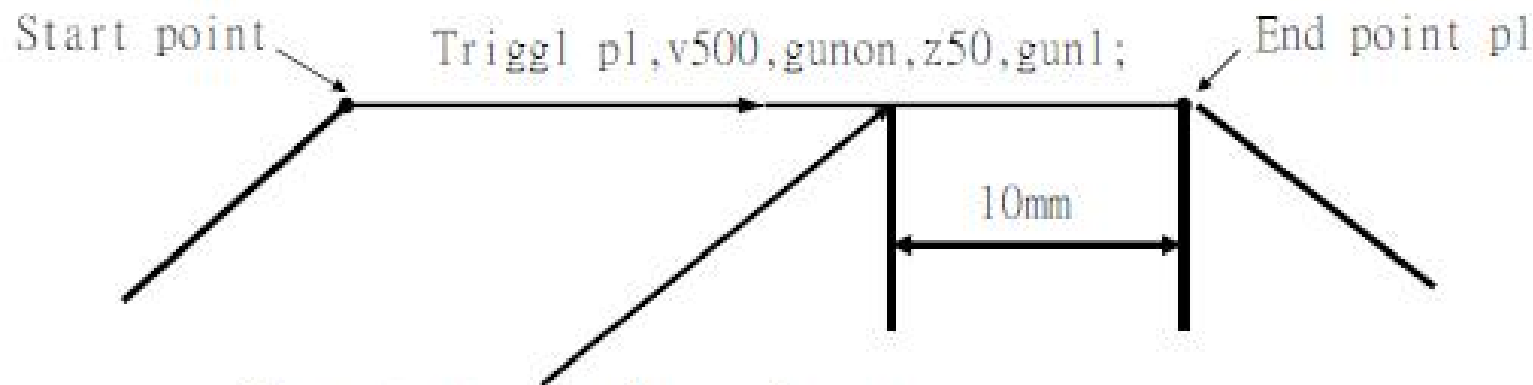
使用参变量[\Start]，表示以运动起始点触发基准点，默认为运动终止点；

使用参变量[\Time]，以时间来控制触发，允许最大时间为0.5s，详见限制，参变量[\ProcID]，正常情况下用户无法自行使用，此参变量用于IPM过程处理。



## 实例:

```
VAR triggdata gunon;  
TriggIO gunon,10\Dop:=gun,l;  
TriggL pl,v500,gunon,z50,gunl;
```



The output signal gun is set  
when the TCP is here



## 限制:

当前指令使有参变量[\Time]可以提高信号输出精度，此参变量以目标点为基准，使用固定的目标点fine此转角zone精度高，一般情况下，此参变量采用固定目标点。

参变量[\Time]设置的时间小于机器人开始减速时间（最大0.5s），例如：运行速度500mm/s，IRB2400为150ms，IRB6400为250ms，机器人在设置时间超过减速时间的的情况下，实际控制时间会缩短，但不会对正常运行造成影响。





## (2) TriggInt

```
TriggInt TriggData,Distance[\Start][\Time],Interrupt;
```

|                     |         |             |
|---------------------|---------|-------------|
| <b>[TriggData]:</b> | 触发变量名称。 | (triggdata) |
| <b>Distance:</b>    | 触发距离mm。 | (num)       |
| <b>[\Start]:</b>    | 触发起始开关。 | (switch)    |
| <b>[\Time]:</b>     | 时间触发开关。 | (switch)    |
| <b>Interrupt :</b>  | 触发中断名称。 | (signaldo)  |



## 应用:

机器人可以在运动时通过触发指令精确的输出相应信号，当前指令用于定义触发性质，此指令必须与其他触发指令TriggJ、TriggL或TriggC同时使用才有意义，通常用于喷涂、涂胶等行业，使用参变量[\Start]，表示以运动起始点触发基准点，默认为运动终止点；使用参变量[\Time]，以时间来控制触发，允许最大时间为0.5s，详见限制。



## 限制:

正常情况下, 当前指令从触发中断到得到响应, 有5-120ms延迟, 用指令TriggIO或TriggEquit控制信号输出效果最佳。

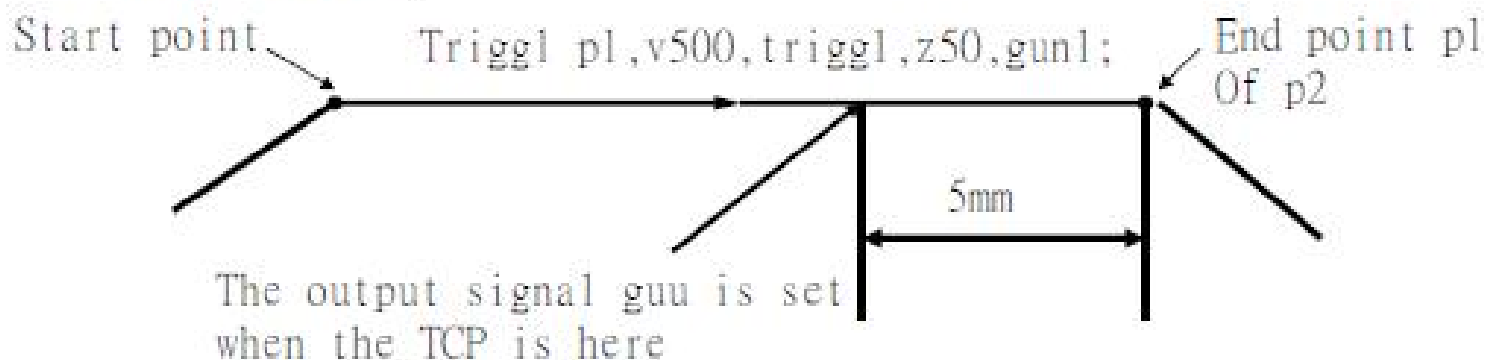
当前指令使用参变量, [\Time]可以提高信号输出精度, 此参变量以目标点为基准, 使用固定的目标点fine, 此转角zone精度高, 一般情况下, 此参变量采用固定目标点。

参变量[\Time]设置的时间小于机器人开始减速时间(最大0.5s), 例如: 运行速度500mm/s, IRB2400为150ms, IRB6400为250ms, 机器人在设置时间超过减速时间的情况下, 实际控制时间会缩短, 但不会对正常运行造成影响。



## 实例:

```
VAR intnum intno1;  
VAR triggdata triggl;  
CONNECT intno1 WITH trap1;  
Triggint triggl,5,intno1;  
TriggL p1,v500,triggl,z50,gun1;  
TriggL p2,v500,triggl,z50,gun1;  
Idelete intno1;
```





## (3) TriggEquip

TriggEquip TriggData,Distance [\Start],EquipLag[\DOp][GOp][AOp][\ProcID],SetValue  
[\Inhib];

|                     |          |             |
|---------------------|----------|-------------|
| <b>[TriggData]:</b> | 触发变量名称。  | (triggdata) |
| <b>Distance:</b>    | 触发距离mm。  | (num)       |
| <b>[\Start]:</b>    | 触发起始开关。  | (switch)    |
| <b>EquipLag:</b>    | 触发延迟补偿s。 | (switch)    |
| <b>[\Dop]:</b>      | 触发数字输出。  | (signaldo)  |
| <b>[\Gop]:</b>      | 触发组合输出。  | (signalgo)  |
| <b>[\Aop]:</b>      | 触发模拟输出。  | (signalao)  |
| <b>[\ProcID]:</b>   | 过程处理触发。  | (num)       |
| <b>SetValue:</b>    | 相应信号值。   | (num)       |
| <b>[\Inhib]:</b>    | 信号抑止数据。  | (bool)      |



## 应用：

机器人可以在运动时通过触发指令精确的输出相应信号，当前指令用于定义触发性质，此指令必须与其他触发指令TriggJ、TriggL或TriggC同时使用才有意义，同机器人指令TriggIO比较，多了外部设备触发延迟功能，少了时间控制功能，通常用于喷涂、涂胶等行业。使用参变量[\Start]，表示以运动起始点触发基准点，默认为运动终止点；参变量[\ProcID]，正常情况下用户无法自行使用，此参变量用于IPM过程处理。当参变量[\Inhib]值为TRUE，在触发点所有输出信号（AO GO DO）将被置为0。

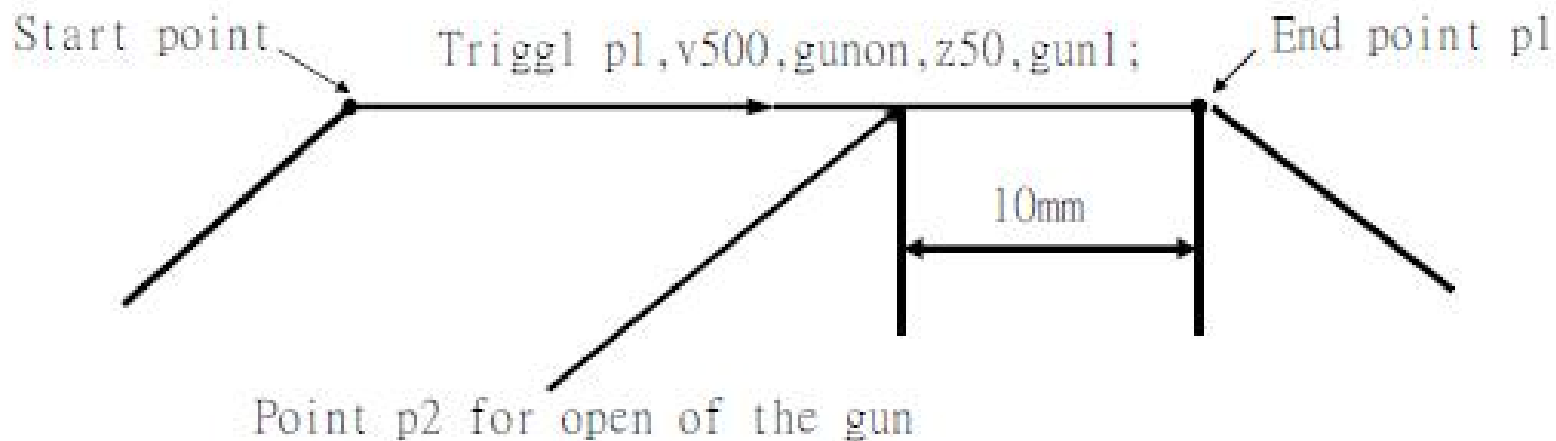






## 实例:

```
VAR triggdata gunon;  
TriggEquip gunon,10,0.1\Dop:=gun,1;  
TriggL p1,v500,gunon,z50,gun1;
```





## 限制:

当前指令通过触发延迟可以提高信号输出精度，设置的时间小于机器人开始减速时间（最大0.5s），例如：运行速度500mm/s，IRB2400为150ms，IRB6400为250ms，机器人在设置时间超过减速时间的情况下，实际控制时间会缩短，但不会对正常运行造成影响。

触发延迟EquipLag值应小于系统参数内Event preset Time配置值，默认为60ms。

如果触发延迟EquipLag值应大于系统参数内Event preset Time配置值，需要使用指令SingArea \Wrist。



## (4) TriggJ

WaitTime [\InPos,] Time;

**[\InPos]** : 程序运行提前量开关。 (switch)

**Time** : 相应等待时间。

(num)

### 应用:

当前指令只用于机器人等待相应时间后,才执行以后指令,使用参变量[\InPOS],机器人及外轴必须在完全停止的情况下,才进行等待进间计时,此指令会延长循环时间。



## (5) TriggL

**[\InPos] :** 程序运行提前量开关。 (switch)  
**Time :** 相应等待时间。  
(num)

### 应用:

当前指令只用于机器人等待相应时间后,才执行以后指令,使用参变量[\InPOS],机器人及外轴必须在完全停止的情况下,才进行等待进间计时,此指令会延长循环时间。



## 4、出错或中断时的运动控制

| 指令              | 说明                        |
|-----------------|---------------------------|
| StopMove        | 停止机器人运动                   |
| StartMove       | 重新启动机器人运动                 |
| StartMoveRetry  | 重新启动机器人运动及相关的参数设定         |
| StartMoveReset  | 对停止运动状态复位，但不重新启动机器人运动     |
| StorePath       | 储存已生成的最近路径                |
| RestoPath       | 重新生成之前储存的路径               |
| ClearPath       | 在当前的运动路径级别中，清空整个运动路径      |
| PathLevel       | 获取当前路径级别                  |
| SyncMoveSuspend | 在StorePath的路径级别中暂停同步坐标的运动 |
| SyncMoveResume  | 在StorePath的路径级别中重返同步坐标的运动 |
| IsStopMoveAct   | 获取当前停止运动标志符               |



## 5、外轴的控制

| 指令               | 说明               |
|------------------|------------------|
| DeactUnit        | 关闭一个外轴单元         |
| ActUnit          | 激活一个外轴单元         |
| MechUnitLoad     | 定义外轴单元的有效载荷      |
| GetNextMechUnit  | 检索外轴单元在机器人系统中的名字 |
| IsMechUnitActive | 检查一个外轴单元状态是关闭/激活 |





## (1) DeactUnit

**MecUnit:** 外轴名 (mecunit)

**应用:**

使机器人外轴失效，例如，当多个外轴公用一个驱动板时，通过外轴激活指令DeactUnit使当前所使用的外轴失效。



## 实例:

MoveL p10,v100,fine,tool1      --p10,外轴不动

ActUnit track\_motion;

MoveL p20,v100,z10,tool1;      --p20,外轴联动

DeactUnit track\_motion;

ActUnit orbit\_a;

MoveL p30,v100,z10,tool1;      --P30,外轴联动 orbit\_a

## 限制:

不能在指令StorePath...Restopath内使用。

不能在预置程序RESTART内使用。



## (2) ActUnit

AccUnit MechUnit

MechUnit: 外轴名

数据类型: mecunit (机械单元) 要激活的机械单元的名称

**应用:**

将机器人一个外轴激活, 例如: 当多个外轴公用一个驱动板时, 通常外轴激活指令ActUnit选择当前所使用的外轴。



## 实例：

```
MoveL p10,v100,fine,tool1          --p10,外轴不动
ActUnit track_motion;
MoveL p20,v100,z10,tool1;          --p20, 外轴联动
DeactUnit track_motion;            track_motion
ActUnit orbit_a;
MoveL p30,v100,z10,tool1;          --P30,外轴联动
                                   orbit_a
```

## 限制：

不能在指令StorePath...Restopath内使用。

不能在预置程序RESTART内使用。

不能在机器人转轴处于独立状态使用。



## 6、独立轴的控制

| 指令       | 说明                      |
|----------|-------------------------|
| IndAMove | 将一个轴设定为独立轴模式并进行绝对位置方式运动 |
| IndCMove | 将一个轴设定为独立轴模式并进行连续方式运动   |
| IndDMove | 将一个轴设定为独立轴模式并进行角度方式运动   |
| IndRMove | 将一个轴设定为独立轴模式并进行相对位置方式运动 |
| IndReset | 取消独立轴模式                 |
| IndInpos | 检查独立轴是否已达到指定位置          |
| IndSpeed | 检查独立轴是否已达到指定的速度         |



## 7、路径修正功能

| 指令         | 说明                   |
|------------|----------------------|
| CorrCon    | 连接一个路径修正生成器          |
| CorrWrite  | 将路径坐标系统中的修正值写到修正生成器  |
| CorrDiscon | 断开一个已连接的路径修正生成器      |
| CorrClear  | 取消所有已连接的路径修正生成器      |
| CorrRead   | 读取所有已连接的路径修正生成器的总修正值 |





## 8、路径记录功能

| 指令              | 说明                             |
|-----------------|--------------------------------|
| PathRecStart    | 开始记录机器人的路径                     |
| PathRecStop     | 停止记录机器人的路径                     |
| PathRecMoveBwd  | 机器人根据记录的路径作后退运动                |
| PathRecMoveFwd  | 机器人运动到执行PathRecMoveBwd这个指令的位置上 |
| PathRecValidBwd | 检查是否已激活路径记录和是否有可后退的路径          |
| PathRecValidFwd | 检查是否有可向前的记录路径                  |



## 9、输送链跟踪功能

| 指令       | 说明          |
|----------|-------------|
| WaitWObj | 等待输送链上的工件坐标 |
| DropWObj | 放弃输送链上的工件坐标 |



## 10、传感器同步功能

| 指令           | 说明                    |
|--------------|-----------------------|
| WaitSensor   | 将一个在开始窗口的对象与传感器设备关联起来 |
| SyncToSensor | 开始/停止机器人与传感器设备的运动同步   |
| DropSensor   | 断开当前对象的连接             |



## 11、有效载荷和碰撞检测

| 指令        | 说明         |
|-----------|------------|
| MotionSup | 激活/关闭运动监控  |
| LoadId    | 工具或有效载荷的识别 |
| ManLoadId | 外轴有效载荷的识别  |



## 12、关于位置的功能

| 指令                 | 说明                      |
|--------------------|-------------------------|
| Offs               | 对机器人位置进行偏移              |
| RelTool            | 对工具的位置和姿态进行偏移           |
| CaleRobT           | 从jointtarget计算出robtargt |
| CPos               | 读取机器人当前的X、Y、Z           |
| CRobT              | 读取机器人当前的jointtarget     |
| CJointT            | 读取机器人当前的关节轴角度           |
| ReadMotor          | 读取轴电动机当前的角度             |
| CTool              | 读取工具坐标当前的数据             |
| CWObj              | 读取工作坐标当前的数据             |
| MirPos             | 镜像一个位置                  |
| CaleJointT         | 从robtargt计算出jointtarget |
| Distance           | 计算两个位置的距离               |
| PFRestart          | 检查当路径因电源关闭而中断的时候        |
| CSpeedOverrid<br>e | 读取当前使用的速率倍率             |