

Grip3 的TCP沿直线运动到停止点 p5。当停止点 fine 的50%的位置条件和 50%的速度条件满足的时候，机器人认为它到达了目标点。它等条件满足最多等两秒，参看 stoppointdata数据类型的预定义数据 inpos50。

例3 MoveL \Conc, \*, v2000, z40, grip3;

Grip3 的TCP直线运动到存储在指令中的位置。当机器人移动的时候，后续的逻辑指令开始执行。

例4 MoveL start, v2000, z40, grip3 \WObj:=fixture;

Grip3 的TCP直线运动到位置 start，位置在 fixture 的对象坐标系统中指定。

语法：

```
MoveL _
  [ ' 'Conc ; ]_
  [ ToPoint := ] < robtarget类型的表达式 (IN) > ;'
  [ ' 'ID := ' < identno类型的表达式 (IN) > ] ;'_
  [ Speed := ] < speeddata类型的表达式 (IN) >_
  [ ' 'V := ' < num类型的表达式 (IN) > ]_
  | [ ' 'T := ' < num类型的表达式 (IN) > ] ;'_
  [ Zone := ] < zonedata类型的表达式 (IN) >_
  [ ' 'Z := ' < num类型的表达式 (IN) > ]_
  [ ' 'Inpos := ' < stoppointdata类型的表达式 (IN) > ] ;'_
  [ Tool := ] < tooldata类型的恒量 (PERS) > _
  [ ' 'WObj := ' < wobjdata 类型的恒量 (PERS)> ]_
  [ ' 'Corr ] ;'
```

相关信息：

| 相关信息       | 参看                                       |
|------------|--|
| 其他位置指令     | RAPID 参考手册 - RAPID 概述，RAPID 摘要 - 运动部分    |
| 速度的定义      | 第 1010 页 speeddata—速度数据                  |
| Zone 数据的定义 | 第 1047 页 zonedata—zone 数据                |
| 停止点数据的定义   | 第 1014 页 stoppointdata - 停止点数据           |
| 工具 的定义     | 第 1031 页 tooldata—工具 数据                  |
| 工作 对象的定义   | 第 1039 页 wobjdata—工作 对象数据                |
| 写入一个改正入口   | 第 67 页 CorrWrite 写入一个改正发生 器              |
| 运动 综述      | RAPID 参考手册 —RAPID 概述，运动和 I/O 原理 部分       |
| 坐标系        | RAPID 参考手册 - RAPID 概述，运动和 I/O 原理 - 坐标系部分 |

|         |   |
|---------|---|
| 并发的程序执行 | RAPID 参考手册 — RAPID 概述, 运动和 I/O 原理 — 用逻辑指令 同步 部分 |
|         |   |

## 1.97.MoveLDO - 直线移动机器人并且在转角处设置数字输出

用途：

MoveLDO(直线运动数字输出)用来直线移动 TCP到指定的目标点。在转角路径的中间位置,指定的数字输出信号被置位/复位。

当 TCP仍旧固定的时候,该指令也可以用来给工具重新定向。

该指令只能用在主任务 T\_ROB1 中,或者在多运动系统中的运动任务中。

基本范例：

该指令的基本范例说明如下：

例1 MoveLDO p1, v1000, z30, tool2, do1,1;

工具 tool2 的 TCP直线运动到目标位置 p1,速度数据 v1000和 zone数据 z30。在 p1的转角路径的中间位置,输出信号 do1被置位。

项目：

MoveLDO ToPoint [ID] Speed [T] Zone Tool [WObj] Signal Value

ToPoint：

数据类型：robtarget

机器人和外部轴的目标位置。定义为一个命名的位置或者直接存储在指令中(在指令中用\*标记)。

[ID]：

同步 ID

数据类型：identno

如果并列了同步运动,该项目必须使用在多运动系统中,并且不允许在其他任何情况下使用。

指定的 ID 号在所有协同的程序任务中必须相同。该 ID 号保证在 routine 中运动不会混乱。

Speed：

数据类型：speeddata

应用到运动中的速度数据。速度数据定义 TCP、工具重新定向或者外部轴的速度。

[T]：

时间

数据类型：num

该项目用来指定外部轴运动的总时间，单位秒。它代替相应的速度数据。

Zone :

数据类型：zonedata

运动的 zone 数据。它描述产生的转角路径的大小。

Tool :

数据类型：tooldata

机器人运动时所使用的工具。TCP 就是移动到目标点的那个点。

[Wobj] :

工作对象

数据类型：wobjdata

指令中机器人位置相关到的工作对象（坐标系）。该项目可以忽略，如果忽略的话，位置相关到世界坐标系。

另一方面，如果使用了静态 TCP 或者并列了外部轴，该项目必须指定。

Signal :

数据类型：signaldo

要改变的数字输出信号的名称。

Value :

数据类型：dionum

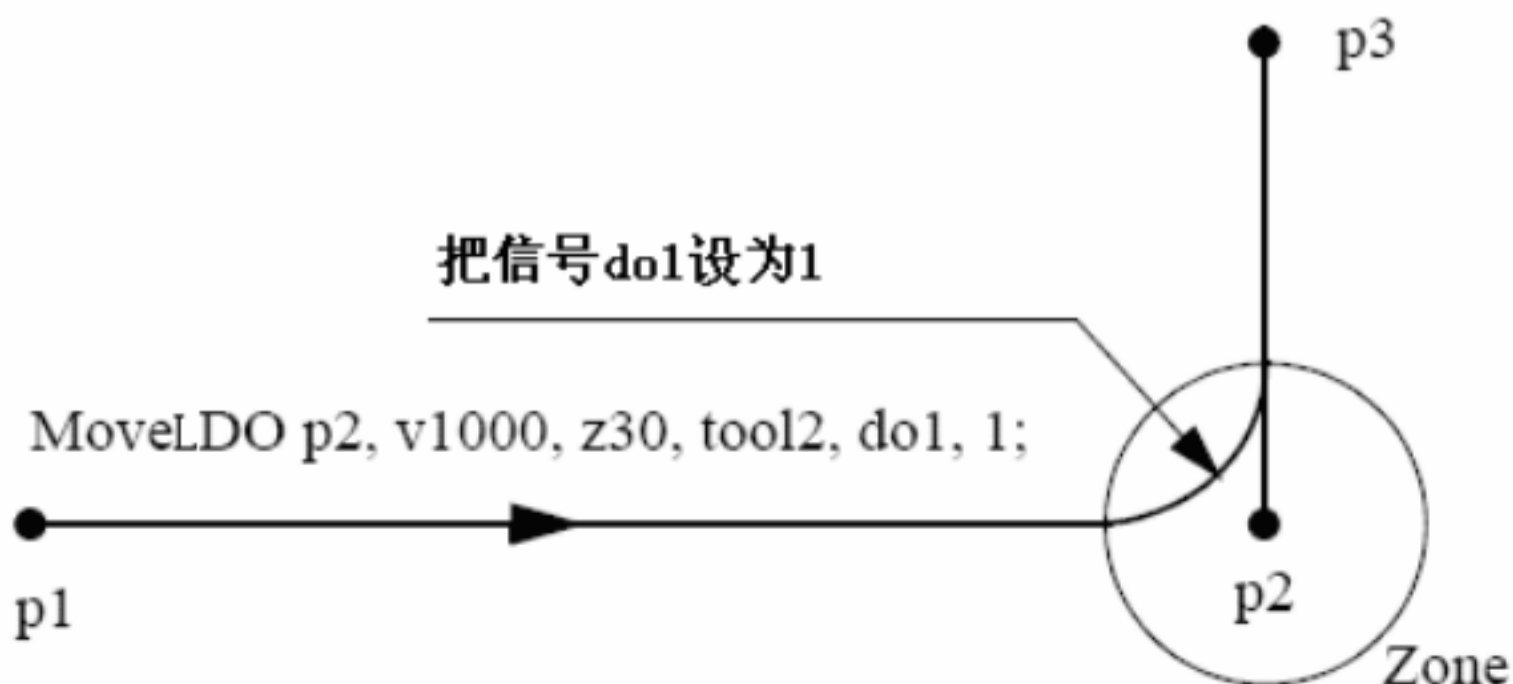
期望的信号数值（0 或者 1）。

程序执行：

参考指令 MoveL，可以得到关节运动的更多信息。

在飞点的转角路径的中间位置，数字输出信号置位/复位，如下图所示。

下图说明在转角路径 MoveLDO 指令的数字输出信号的置位/复位。



对于停止点，我们推荐使用“正常”的编程顺序，即 MoveJ + SetDO。但是当在指令 MoveLDO 中使用停止

点、当机器人到达停止点的时候，数字输出信号置位/复位。

在执行模式继续逐步向前而不是逐步向后时，指定的 I/O 信号被置位/复位。

语法：

MoveLDO \_

[ ToPoint := ] < robtarget类型的表达式 (IN) > ;'\_

[ 'ID := ' < identno类型的表达式 (IN) > ] ;'\_

[ Speed := ] < speeddata类型的表达式 (IN) > \_

[ 'T := ' < num类型的表达式 (IN) > ] ;'\_

[ Zone := ] < zonedata类型的表达式 (IN) > ;'\_

[ Tool := ] < tooldata类型的恒量 (PERS) > \_

[ 'WObj := ' < wobjdata类型的恒量 (PERS) > ] ;'\_

[ Signal := ] < signaldo类型的变量 (VAR) > ] ;'\_

[ Value := ] < dionum类型的表达式 (IN) > ] ;'\_

相关信息：

| 相关信息        | 参看   |
|-------------|--|
| 其他位置指令      | RAPID 参考手册 - RAPID 概述，RAPID 摘要 - 运动部分        |
| 直线移动机器人     | 第 236 页 MoveL - 直线移动机器人                      |
| 速度的定义       | 第 1010 页 speeddata—速度数据                      |
| Zone 数据的定义  | 第 1047 页 zonedata—zone 数据                    |
| 工具的定义       | 第 1031 页 tooldata—工具 数据                      |
| 工作对象的定义     | 第 1039 页 wobjdata—工作对象数据                     |
| 运动综述        | RAPID 参考手册 —RAPID 概述，运动和 I/O 原理 部分           |
| 坐标系         | RAPID 参考手册 - RAPID 概述，运动和 I/O 原理 - 坐标系部分     |
| 带 I/O 设定的运动 | RAPID 参考手册 —RAPID 概述，运动和 I/O 原理 —用逻辑指令 同步 部分 |

## 1.98.MoveLSync - 直线移动机器人并且执行一个 RAPID 程序

用途：

MoveLSync（同步直线移动）用来直线移动 TCP 到给定的目标位置。在目标点的转角路径的中间位置，指定的 RAPID 程序开始运行。

当 TCP 仍旧固定的时候，该指令也可以用来给工具重新定向。

该指令只能用在主任务 T\_ROB1，或者多运动系统的运动任务中。

基本范 例：

该指令的 基本范 例说明如下。

例1      MoveLSync p1, v1000, z30, tool2, 'proc1 ';

工具 tool2 的 TCP 沿线 性移动到位置 p1，速度数据 v1000，zone 数据 z30。在 p1 的转角路径 的中间位置 程序 proc1 开始执行。

项目：

MoveLSync ToPoint [ID] Speed [T] Zone Tool [WObj] ProcName

ToPoint：

数据类型： rotarget

机器人和 外部轴的目标点。定义为 一个命名 的位置 或者直接存储在指令中 （在指令中 用\*标记）。

[ID]：

同步 ID

数据类型： identno

该 项目必须使用 在多运动系统中， 如果并列 了同步 运动， 则不允许 在其他 任何情况下使用 。

指定的 ID 号在所有协同 的程序任务 中必须 相同。该 ID 号保证 在 routine 中运动 不会混乱 。

如果并列 了同步 运动， 不允许 在其他 任何情况下使用 。

Speed：

数据类型： speeddata

应用 到运动中的速度数据。速度数据定义 TCP、工具重新 定向或者外 部轴的速度。

[T]：

时间

数据类型： num

该 项目用来 指定外 部轴运动的 总时间， 单位秒。它 代替相应的速度数据。

Zone：

数据类型： zonedata

运动的 zone 数据。它 描述产生的转角路径 的大小。

Tool：

数据类型： tooldata

机器人运动时 所使用 的工具。TCP 就是移动到目标点的 那个点。

[Wobj]：

工作 对象

数据类型： wobjdata

指令中机器人位置相关到的 工作对象（坐标系）。该项目可以忽略，如果忽略的话，位置相关到 世界坐标系。另一方面，如果使用了静态 TCP或者并列了外部轴，该项目必须指定。

ProcName：

程序名称

数据类型： string

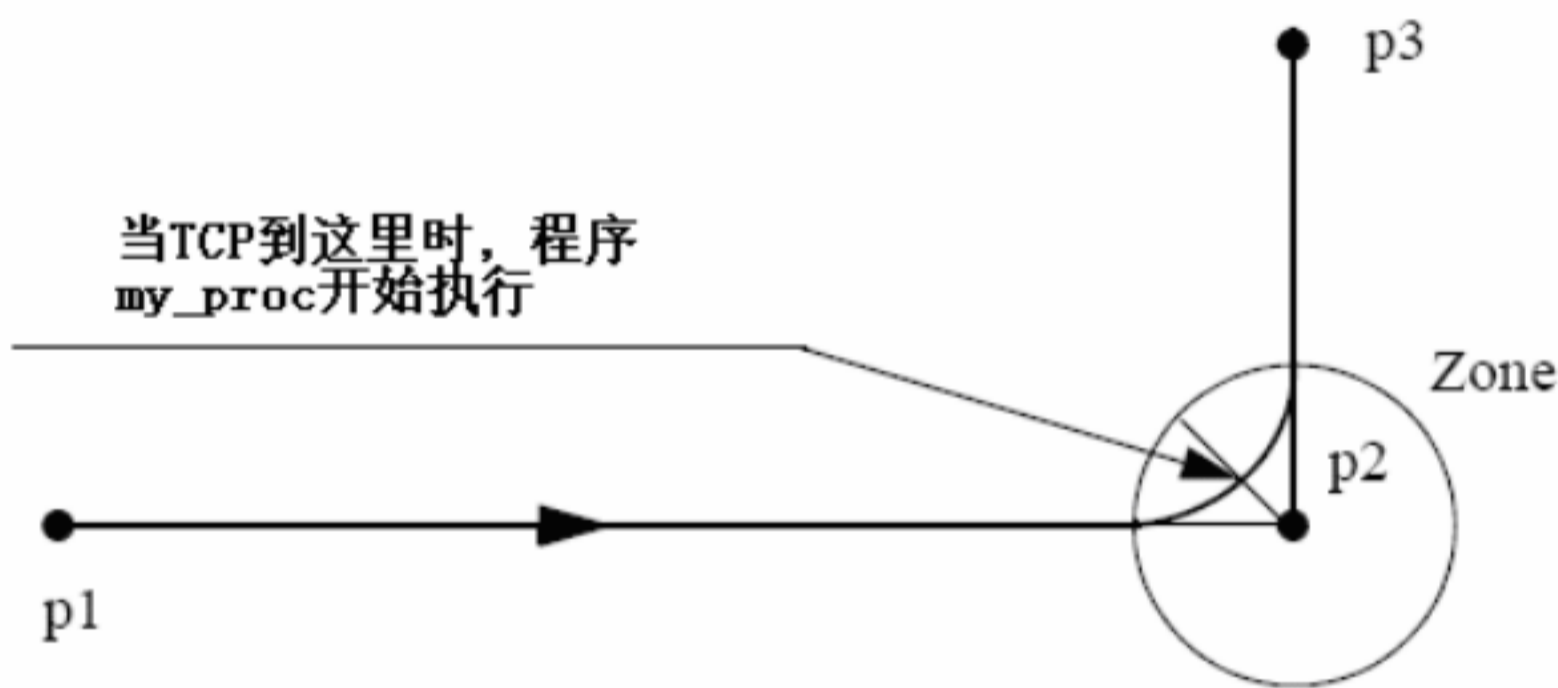
在目标点的 转角路径 的中间位置要执行的 RAPID 程序的名称。

程序执行：

参考指令 MoveJ，可以得到关节运动的更多信息。

当 TCP到达 MoveJSync指令的目标点的 转角路径 的中间位置时，指定的 RAPID 程序开始执行，如下图所示。

```
MoveLSync p2, v1000, z30, tool2, "my_proc";
```



对于停止点，我们推荐使用“正常”的编程顺序，即 MoveL + 其他 RAPID 指令。

下表描述了在不同执行模式下指定的 RAPID 程序的执行：

| 执行模式   | RAPID 程序的执行 |
|--------|-------------|
| 继续或者循环 | 按照该描述       |
| 逐步向前   | 在停止点        |
| 逐步向后   | 一点也不执行      |

限制：

当程序停止后，从连续执行或循环执行切换到逐步向前或者向后将导致错误。该错误告诉用户模式切换将导致路径上的执行队列的 RAPID 程序的执行错误。

指令 MoveLSync 不能用在 TRAP 层次上。指定的 RAPID 程序不能用逐步执行测试。

语法：

```
MoveLSync
```

```

[ ToPoint := ] < rotarget类型的 表达式(IN) > ;'
[ ' 'ID := ' < identno类型的 表达式(IN)> ] ;'
[ Speed := ] < speeddata类型的 表达式(IN) >
[ ' 'T := ' < num类型的 表达式(IN) > ] ;'
[ Zone := ] < zonedata类型的 表达式(IN) >
[ Tool := ] < tooldata 类型的 恒量 (PERS) >
[ ' 'WObj := ' < wobjdata类型的 恒量 (PERS)> ] ;'
[ ProcName := ] <string 类型的 表达式(IN) > ] ;'

```

相关信息：

| 相关信息       | 参看                                       |
|------------|--|
| 其他位置指令     | RAPID 参考手册 - RAPID 概述， RAPID 摘要 - 运动部分   |
| 直线移动机器人    | 第 236 页 MoveL - 直线移动机器人                  |
| 速度的定义      | 第 1010 页 speeddata—速度数据                  |
| Zone 数据的定义 | 第 1047 页 zonedata—zone 数据                |
| 工具 的定义     | 第 1031 页 tooldata—工具 数据                  |
| 工作 对象的定义   | 第 1039 页 wobjdata—工作 对象数据                |
| 运动 综述      | RAPID 参考手册 —RAPID 概述，运动和 I/O 原理 部分       |
| 坐标系        | RAPID 参考手册 - RAPID 概述，运动和 I/O 原理 - 坐标系部分 |

## World Zone：

最多可以在机器人的工作区域内定义 10 个不同的体积空间。他们可以用来：

- | 指出机器人的 TCP 是工作区域中的一个明确的部分。
- | 限制机器人的工作区域，阻止和工具的碰撞。
- | 创建一个由两个机器人公用的区域，该区域在同一时间内只能由一个机器人使用。

### 1.227 . WZBoxDef —定义一个箱体形状的 World Zone

用途：

WZBoxDef ( World Zone 箱体定义 ) 用来定义一个直立箱体形状的 World Zone，该箱体的所有边都和 World 坐标系的坐标轴平行。

基本范 例：

该指令的基本范 例说明如下：

```
例1    VAR shapedata volume;

        CONST pos corner1:=[200, 100, 100];

        CONST pos corner2 :=[600, 400, 400];

        ...

        WZBoxDef \Inside, volume, corner1, corner2;
```

定义一个直立的箱体，该箱体的所有边都和 World 坐标系的轴平行，该箱体由两个对角点 corner1 和 corner2 定义。

项目：

```
WZBoxDef [\Inside] | [\Outside] Shape LowPoint HighPoint
```

[\Inside]:

数据类型： switch

定义箱体内部的体积

[\OutSide]:

数据类型： switch

定义箱体外部的体积（反体积）。

必须指定 \Inside 和 \Outside 两个项目中的一个。

Shape:

数据类型： shapedata

定义的体积的存储的变量（系统的私有（private）数据）。

LowPoint：

数据类型： pos

定义箱体的一个较低角点的位置（x, y, z）以毫米为单位。

HighPoint:

数据类型： pos

定义箱体的另一个相对的角点的位置（x, y, z）以毫米为单位。

程序执行：

箱体的定义存储在 shapedata 类型（Shape 项目）的变量中，用于将来在 WZLimSup 和 WZDOSet 指令中使用。

限制：



LowPoint 和 HighPoint 的位置 必须是有效的相对角点 ( x , y 和 z 的坐标 值都不相同 )。如果用 机器人 来指出 LowPoint 和 HighPoint , 工作对象 ( wobj0 ) 必须 激活 ( 在 rotarget 中使用 trans 组件 , 即 p1.trans 作为项目 )。

语法 :

WZBoxDef

```
[[ ' Inside' ] [ ' Outside' ] ;'
[LowPoint := ]<pos 类型的 表达式 ( IN ) > ;'
[Shape := ]<shapedata 类型的 变量 ( VAR ) > ;'
[HighPoint := ]<pos 类型的 表达式 ( IN ) > ;'
```

相关信息 :

| 相关信息                    | 参看  |
|-------------------------|---|
| World Zones             | 《 RAPID 参考手册 —RAPID 概述 》运动和 I/O 原理 —World Zone 部分 |
| World Zone 形状           | 第 1004 页 shapedata—World Zone 形状 数据部分             |
| 定义 球形 World Zone        | 第 636 页 WZSphDef —定义 球形 World Zone。               |
| 定义 圆柱形 World Zone       | 第 613 页 WZCylDef —定义 圆柱形 World Zone。              |
| 定义关节 home 位的 World Zone | 第 625 页 WZHomeJointDef —定义关节 home 位的 World Zone。  |
| 定义关节限位的 World Zone      | 第 629 页 WZLimJointDef —定义关节限位的 World Zone。        |
| 激活 World Zone 限位管理      | 第 633 页 WZLimSup —激活 World Zone 限位管理。             |
| 激活 World Zone 数字输出 设置   | 第 617 页 WZDOSet —激活 World Zone 来设置数字输出 。          |

## 1.228 . WZCylDef —定义一个圆柱形的 World Zone

用途 :

WZCylDef ( World Zone 圆柱 定义 ) 用来定义 一个 圆柱 形状的 World Zone , 该圆柱的轴线平行于 World 坐标系的 z 轴。

基本范 例 :

该指令的 基本范 例说明如下 :

例 1

```
VAR shapedata volume;
CONST pos C2:= [300, 200, 200];
CONST num R2:= 100;
CONST num H2:=200;
```

...

```
WZCylDef \Inside, volume, C2, R2, H2;
```

定义一个圆柱，底面圆心为 C2，半径 R2，高度 H2。

项目：

```
WZCylDef [\Inside] | [\Outside] Shape CenterPoint Radius Height
```

[Inside]:

数据类型： switch

定义圆柱内部的体积。

[Outside]:

数据类型： switch

定义圆柱外部的体积（反体积）。

必须指定两个项目 \Inside 和 \Outside 中的一个。

Shape:

数据类型： shapedata

用来存储定义的体积的变量（系统的私有（private）数据）。

CentrePoint:

数据类型： pos

定义圆柱的一个底面圆的圆心位置（x, y, z），单位是毫米。

Radius : (半径)

数据类型： num

圆柱的半径，单位是毫米。

Height :

数据类型： num

圆柱的高度，单位是毫米。如果是正的（+z 方向），CentrePoint 项目是圆柱较低底面的圆心（如以上例子）。

Height 如果是负的（-z 方向），CentrePoint 项目是圆柱上底面的圆心。

程序执行：

圆柱的定义存储在 shapedata 类型的变量中（项目 Shape），将来在 WZLimSup 或者 WZDOSet 指令中使用。

限制：

如果用机器人指出 CentrePoint，工作对象 wobj0 必须被激活（使用 robotarget 中的 trans 组件，即 p1.trans 作为项目）。

语法：

## WZCylDef

[ ' Inside ] | [ ' Outside ] ; ' '

[Shape := ]<shapedata 类型的 变量 ( VAR ) > ; ' '

[CentrePoint := ]<pos 类型的 表达式 ( IN ) > ; ' '

[Radius := ]<num 类型的 表达式 ( IN ) > ; ' '

[Height := ]<num 类型的 表达式 ( IN ) > ; ' '

相关信息：

| 相关信息                    | 参看  |
|-------------------------|---|
| World Zones             | 《 RAPID 参考手册 —RAPID 概述 》运动和 I/O 原理 —World Zone 部分 |
| World Zone 形状           | 第 1004 页 shapedata—World Zone 形状 数据部分             |
| 定义球形 World Zone         | 第 636 页 WZSphDef —定义球形 World Zone。                |
| 定义箱体形状的 World Zone      | 第 611 页 WZBoxDef —定义箱体形状的 World Zone。             |
| 定义关节 home 位的 World Zone | 第 625 页 WZHomeJointDef —定义关节 home 位的 World Zone。  |
| 定义关节限位的 World Zone      | 第 629 页 WZLimJointDef —定义关节限位的 World Zone。        |
| 激活 World Zone 限位管理      | 第 633 页 WZLimSup —激活 World Zone 限位管理。             |
| 激活 World Zone 数字输出 设置   | 第 617 页 WZDOSet —激活 World Zone 来设置数字输出 。          |

### 1.229 . WZDisable —解除临时 World Zone 监视

用途：

WZDisable ( 解除 World Zone ) 用来解除对临时 World Zone 的监视，该监视原先用来停止运动 或者设置一个输出。

基本范例：

该指令的基本范例说明如下：

例 1     VAR wztemporary wzzone;

...

PROC ...

    WZLimSup \Temp, wzzone, volume;

    MoveL p\_pick, v500, z40, tool1;

    WZDisable wzzone;

    MoveL p\_place, v200, z30, tool1;

ENDPROC

当移动到 p\_pick 的时候，机器人 TCP 的位置 被检测到，这样机器人 将不能够进入指定的 体积 wzzone 内部。

当移动到 p\_place 的时候，该监视没有执行。

项目：

WZDisable WorldZone

WorldZone：

数据类型：wztemporary

Wztemporary 类型的变量或者常量，包含要解除的 WorldZone 的标识符。

程序执行：

临时 WorldZone 被解除。也就是说对机器人 TCP 在相应体积空间内的监视被临时停止。它可以通过 WZEnable 指令被再次激活。

限制：

只有临时 WorldZone 可以被解除。一个静态的 WorldZone 总是激活的。

语法：

WZDisable

[WorldZone != ]<wztemporary 类型的变量或者常量 ( INOUT ) > ;'

相关信息：

| 相关信息                 | 参看  |
|----------------------|---|
| World Zones          | 《RAPID 参考手册 —RAPID 概述》运动和 I/O 原理 —World Zone 部分 |
| 定于圆柱形状 World Zone    | 第 613 页 WZCylDef —定义圆柱形状的 World Zone。           |
| 临时 World Zone 数据     | 第 1045 页 wztemporary—临时 WorldZone 数据            |
| 激活 WorldZone 限位监视    | 第 633 页 WZLimSup —激活 WorldZone 限位监视             |
| 激活 World Zone 数字输出设置 | 第 617 页 WZDOSet—激活 World Zone 来设置数字输出。          |
| 激活 WorldZone         | 第 621 页 WZEnable—激活临时监视                         |
| 擦除 WorldZone         | 第 623 页 WZFree—擦除临时 WorldZone 监视                |

1.230 . WZDOSet —激活 WorldZone 来设置数字输出

用途：

WZDOSet ( WorldZone 数字输出设置 ) 用来定义动作并且激活一个 WorldZone 来监视机器人运动。

在该指令执行以后，当机器人的 TCP 或机器人 /外部轴 ( 关节中的区域 ) 在定义的 WorldZone 内部或者接近 WorldZone 时，一个数字输出信号被设为一个特定的数值。

基本范例：

该指令的基本范例说明如下：

例 1 VAR wztemporary service;

```

PROC zone_output( )

    VAR shapedata volume;

    CONST pos p_service:= [500, 500, 700];

    ...

    WZSphDef \Inside, volume, p_service, 50;

    WZDOSet \Temp, service \Inside, volume, do_service, 1;

ENDPROC

```

在应用程序中定义临时 WorldZone service，当机器人 TCP 在程序执行过程中或者点动过程中进入定义的球体时，设定信号 do\_service。

项目：

```
WZDOSet [\Temp] | [\Stat] WorldZone [\Inside] | [\Before] Shape Signal SetV alue
```

[Temp]：

临时的

数据类型： switch

要定义的 WorldZone 是一个临时的 WorldZone。

[\Stat]：

静态的

数据类型： switch

要定义的 WorldZone 是一个静态的 WorldZone。

必须指定 [\Temp] 和 [\Stat] 两个项目中的一个。

WorldZone：

数据类型： wztemporary 或者 wzstationary

可以根据 WorldZone 的特性（数字数值）进行更新的变量或者恒量。

如果使用可选项 \Temp，数据类型必须是 wztemporary。如果使用了 \Stat，数据类型必须是 wzstationary。

[\Inside]：

数据类型： switch

当机器人的 TCP 或者某一个轴进入定义的体积空间内的时候，将设定数字输出信号。

[\Before]：

数据类型： switch

当机器人的 TCP 或者某一个轴进入定义的体积空间之前（马上就要进入空间），将设定数字输出信号。

两个项目 [\Inside] 和 [\Before] 必须选定一个。

Shape :

数据类型 : shapedata

定义 WorldZone 空间的变量。

Signal :

数据类型 : signaldo

将要改变的数字输出信号的名称。

如果使用了静态 WorldZone , 信号必须写保护 , 防止用户进入 ( RAPID , FP 示教器 )。在系统参数 或者 指定的轴上设定用户进入等级。

SetValue:

数据类型 : dionum

当机器人 TCP 进入体积空间或者恰好在进入之前 , 期望的信号输出的数值 ( 1 或者 0 )。

在机器人 TCP 在外面或者正好在空间外面 , 信号输出为相反的数值。

程序执行 :

定义的 WorldZone 被激活。从这时开始 , 机器人 TCP 位置 ( 或者 机器人 /外部轴位置 ) 将被监视 , 当机器人 TCP 位置 ( 或者 机器人 /外部轴位置 ) 在空间内 ( \Inside ) 或者接近空间的边界 ( \Before ) , 将被设置输出。

如果和 WZDOSet 同时使用了 WZHomeJointDef 或者 WZLimJointDef 指令 , 只有在带空间监视的所有激活的轴即将进入或者已经进入关节空间时 , 才能够设置数字输出信号。

更多范例 :

有关该指令如何使用的更多范例说明如下 :

```
例 1    VAR wztemporary home;

        VAR wztemporary service;

        PERS wztemporary equip1:=0];

        PROC main( )

            ...

            ! 定义所有临时的 WorldZone

            Zone_output;

            ...

            ! equip1 在机器人 工作 区域

            WZEnable equip1;
```

```

...
! equip1 在机器人 工作 区域 之外
WZDisable equip1;
...
! 不再使用 equip1
WZFree equip1;
...
ENDPROC

PROC zone_output( )
    VAR shapedata volume;
    CONST pos p_home:=[800, 0, 800];
    CONST pos p_service:=[800, 800, 800];
    CONST pos p_equip1:=[-800,-800, 0];
    ...
    WZSphDef \Inside, volume, p_home, 50;
    WZDOSet \Temp, home \Inside, volume, do_home, 1;
    WZSphDef |Inside, volume, p_service, 50;
    WZDOSet \Temp, service \Inside, volume, do_service, 1;
    WZCylDef \Inside, volume, p_equip1, 300, 1000;
    WZLimSup \Temp, equip1, volume;
    ! equip1 不在机器人 工作 区域。
    WZDisable equip1;
ENDPROC

```

在应用程序中定义临时 WorldZone home 和 service，当机器人在程序执行或者点动过程中分别进入球体 home 或者 service 时，这两个 WorldZone 用来设定信号 do\_home 和 do\_service。

同时，定义一个临时 WorldZone equip1，equip1 只有在机器人程序中、当 equip1 在机器人工作区域以内的时候才会被激活。这时候，无论在程序执行或者手动的时候，机器人在进入 equip1 区域之前停止。通过使用恒量 equip1 的数值，equip1 可以从其它程序任务中使能或者解除。

限制：

WorldZone 不能通过使用项目 WorldZone 中的相同的变量重复定义。