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	

1.92． MoveExtJ －移动一个或者多个没有 TCP 的机械单元

用途：

MoveExtJ（移动外部关节）只用来移动线性或者旋转外部轴。该外部轴可以属于一个或者多个没有 TCP 的外部单元。

该指令只能用来：

l 和定义为运动任务的实际程序任务一起使用，并且

l 如果任务控制一个或者多个没有 TCP 的机械单元。

基本范例：

该指令的基本范例说明如下：

也参看第 225 页的 更多范例。

例1 MoveExtJ jpos10, vrot10, z50;

移动旋转外部轴 到关节 位置 jpos10， 速度 10°/秒， zone 数据 z50。

例2 MoveExtJ \Conc, jpos20, vrot10 \T:=5, fine \InPos:=inpos20;

5 秒钟把 外部轴移动 到关节 位置 jpos20。程序 立即向前执行 ， 但是 外部轴 停止在位置 jpos20， 直到 inpos20 的收敛 性标 准满足 。

项目 ：

MoveExtJ [\Conc] To JointPos [\ID] Speed [\T] Zone [\Inpos]

[\Conc] ：

并 发事件

数据类型 ：switch

当 外部轴运动的同 时， 后续 的指令 开始执行 。该 项目通常不 使用， 但是当 使用 飞点 （flyby points）时，可以用来 避免由 CPU 过载引 起的 不想要 的停止 。当 使用 高速度 并且 编程 点相距较近时这是很 有用的。例如， 当不要求与 外部设 备通讯 和外部设 备和机 器人通讯 同步的 时候 ， 这个项目 也很有用。

使用 项目 \Conc 的时候 ， 连续 的运动指令的 数量限 制为 5 个。 在包括 StorePath– RestorePath 的程序 段中不允许 使用带有 \Conc 项目 的运动指令。

如果 不使用该 项目 ， 并且 ToPoint 不是停止点 ， 在机 器人到达 程序 zone 之前 一 段时间后续 指令 就开始执行

了。

在 多运动系 统中 的坐标同步运动　中不 能使用该 项目 。

ToJointPos：

到达 关节 位置

数据类型 ：jointtarget

外部轴的 绝对目 标轴 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

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 定义旋转或者线性外部轴的　　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 定义 停止点 或者 飞点 。如果 是飞点 它描 述线性或者旋转外部轴的　　减 速度 或者 加 速度 。

[\Inpos] ：

到位

数据类型 ：stoppointdata（停止点数据 ）

该 项目 用来指定机　器人　TCP 在停止点位置　的 收敛 性 判别 标 准 。该 停止点数据 代替 在　zone 参 数中 指定的 zone。

程序 执行 ：

线性或者旋转外部轴　　按照 编 程的 速度 移动 到编 程的 点。

更 多范例：

```
CONST    jointtarget    j1 :=[[9E9,9E9.9E9.9E9.9E9.9E9.9E9],[0,9E9,9E9,9E9,9E9,9E9]];
```

by　张建辉，　韩鹏排版

```
CONST    jointtarget    j2 :=[[9E9,9E9.9E9.9E9.9E9.9E9.9E9],[30,9E9,9E9,9E9,9E9,9E9]];
CONST    jointtarget    j3 :=[[9E9,9E9.9E9.9E9.9E9.9E9.9E9],[60,9E9,9E9,9E9,9E9,9E9]];
CONST    jointtarget    j4 :=[[9E9,9E9.9E9.9E9.9E9.9E9.9E9],[90,9E9,9E9,9E9,9E9,9E9]];
CONST    speeddata   rot_ax_speed    :=[0,0,0,45];

MoveeExtJ j1, rot_ax_speed, fine;
        MoveeExtJ j2, rot_ax_speed, z20;
        MoveeExtJ j3, rot_ax_speed, z20;
        MoveeExtJ j4, rot_ax_speed, fine
```

在该例子中，旋转独立轴移动到轴位置 0, 30, 60 和 90 度，移动速度为 45 度/秒。

语法：

MoveExtJ [‘ ’Conc ‘; ’]

[ToJointPos ’:= ’]<jointtarget 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）>

[‘ ’Inpos ‘:= ’<stoppointdata 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	
关节目标（jointtarget）的定义	第 959 页 jointtarget －关节位置数据	
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	
运动综述	RAPID 参考手册 －RAPID 概述，运动和 I/O 原理部分	
当前程序执行	RAPID 参考手册－ RAPID 概述，运动和 I/O 原理部分	

1.93． MoveJ －通过关节移动移动机器人

用途：

当运动不必是直线的时候， MoveJ 用来快速将机器人从一个点运动到另一个点。

by 张建辉， 韩鹏排版

机 器人 和外部轴 沿着 一个 非 直 线的 路径 移动 到目 标 点， 所 有轴同 时到达目 标 点。

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：

也可参看第 228 页 更 多指令。

例 1 MoveJ p1, vmax, z30, tool2;

工具 tool2 的 TCP 沿着 一个 非 线性 路径 到位置 p1， 速度数据是 vmax， zone 数据是 z30。

例 2 MoveJ *, vmax \T:=5, fine, grip3;

工具 grip3 的 TCP 沿着 一个 非 线性 路径 运动 到存储在 指令 中 的 停止点 （用 * 标 记）。整 个运动 需 要 5 秒钟 。

项目 ：

MoveJ [\Conc] ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool [\WObj]

[\Conc] ：

并 发事件

数据类型 ：switch

当 机 器人 运动的同 时， 后续 的指令 开始执行 。该 项目通常不 使用， 但是当 使用 飞点 （flyby points）时，可 以用来 避免由 CPU 过载引 起的 不想要 的 停止 。当 使用 高速度 并且 编 程 点相距较近时这是很 有用的。例如， 当不 要求与 外部设 备通讯 或外部设 备和机 器人通讯 同步的 时候 ， 这 个项目 也 很有用。

使用 项目 \Conc 的时候 ， 连续 的运动指令的 数量限 制为 5 个。 在包括 StorePath— RestorePath 的程序 段中不 允许 使用带有 \Conc 项目 的运动指令。

如果 不使用该 项目 ， 并且 ToPoint 不是停止点 ， 在 机 器人到达 程序 zone 之前 一 段时间后续 指令 就开始执行 了。

在 多运动系 统中 的坐标同步运动 中不 能使用该 项目 。

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

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。

by 张建辉， 韩鹏排版

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 。

[\V] ：

速度

数据类型 ：num

该 项目 用来 在 指令 中直接 指定 TCP 的 速度 ， 单 位 mm/s。 它用来 代替 速度数据中相 应 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 。

[\Z] ：

Zone

数据类型 ：num

该 项目 用来 在 指令 中直接 指定机 器人 TCP 的 位置 精 度。转 角路径 的 长度 单 位是 mm，它代替 zone 数据中相 应 的 zone。

[\Inpos] ：

到位

数据类型 ：stoppointdata（停止点数据 ）

该 项目 用来指定机 器人 TCP 在停止点位置 的 收敛 性 判别 标 准 。该 停止点数据 代替 在 zone 参 数中 指定的 zone。

Tool：

数据类型 ：tooldata

当 机 器人 运动的 时候 使用的 工具 。TCP 是 移动 到 指定的 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。

by 张建辉， 韩鹏排版

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。

程序 执行 ：

机 器人TCP用轴 角度 插补 移动 到目 标点。也 就是 说每 一个轴 都 使用一个 固定的轴 速度 并且 所有轴同 时到达 目标点，所走 的是一个 非线性的 路径 。

总 的来说， TCP按照大约 的编 程速度 运动（ 无论 是否 并列了 外部轴）。在TCP运动的同 时， 工具重新 定向，外部轴也 在运行。如果 不能 达到 工具重新 定向或者外部轴的 编程的 速度 ， TCP的速度 将降低 。

当 运动 路径 转 到下一 段的时候， 通常 会产生 转 角路径 。如果 在zone数据中 指定 了停止点 ， 只有 当 机 器人 和外部轴 到达 合适 的 位置 的时候，程序 执行 才会继 续。

更 多范例：

如 何 使用 改 指令的 更 多范例说明如下：

例1　　MoveJ *, v2000\V:=2200, z40 \Z:=45, grip3;

工具 grip3的TCP沿着 一个 非 线性 路径 移动 到存储在 指令 中 的 位置 。运动 执行数据 被 设定为 v2000 和z40 ; TCP 的 速度 和zone大小 分 别 是 2200mm/s和 45mm。

例2　　MoveJ p5, v2000, fine \Inpos := inpos50, grip3;

工具 grip3的TCP沿非 线性 路径 移动 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 它已经 到达 该点。它等 待 条 件满足 最多 等2秒 。参考 stoppointdata类型 的 预 定义 数据 inpos50。

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

工具 grip3的TCP沿着 一个 非 线性移动 到指令 中存储 的位置 。当 机 器人 移动的 时候 ， 后续 逻辑指令 开始执行 。

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

工具 grip3的TCP沿着 一个 非线性的 路径 到位置 start。该 位置在 fixture的对象 坐标系 统中 指定。

语法 ：

MoveJ

[’ ’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)>] ’;’_

by　张建辉，　韩鹏排版

[Tool ':= '] < tooldata 类型 的 恒 量 (PERS) >_

[' 'WObj ':= '<wobjdata 类型 的 恒 量 (PERS)>] ';'

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
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

1.94． MoveJDO －通过关节运动移动机器人在转角设定数字输出

用途：

MoveJDO(关节运动 数 字输出 ）在运动 不必是直 线的 时候 用来 快速把 机 器人 从一个 点移动 到另 一个 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。

机 器人 和外部轴 沿着 一个 非 线性的 路径 移动 到目 标 位置 。所 有轴 在同一 时间到达目 标 位置 。

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 MoveJDO p1, vmax, z30, tool2, do1, 1;

工具 tool2 的 TCP 沿着 一个 非 线性 路径 移动 到目 标 位置 p1，速度数据 vmax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 ：

MoveJDO 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

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。

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）。

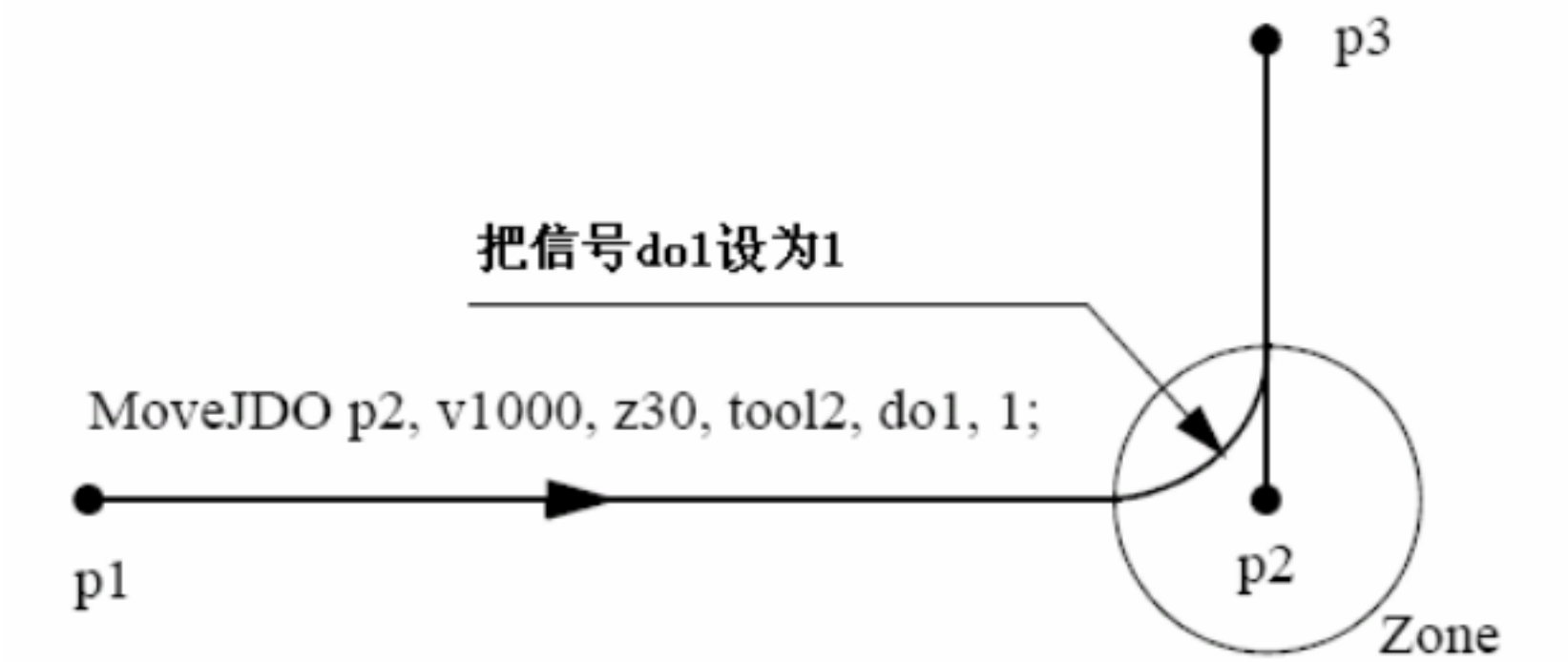
程序 执行 ：

参考指令 MoveJ，可以 得 到 关节运动的 更 多 信息 。

by 张建辉， 韩鹏排版

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 MoveJ 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 MoveJDO 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 ：

MoveJDO

[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
通过 关节运动移动机 器人	第 226 页 MoveJ－通过 关节运动移动机 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

by 张建辉， 韩鹏排版

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

1.95.MoveJSync —通过关节运动移动机器人，并且执行一个 RAPID 程序

用途：

MoveJSync（同步关节移动）用来 在不要求直 线运动的 时候把 机 器人 从一个 点快 速移动 到另一个 点。在目标点的转 角路径 的中间位置 ，指定的 RAPID 程序 开始运行。在运动 过程中， 相对 于圆周 的方 向通常 保持不变。

机 器人 和外部轴 沿着 一个 非线性的 路径 移动 到目 标位置。所有轴 在同一 时间到达目 标位置。

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 MoveJSync p1, vmax, z30, tool2, “proc1 ”;

工具 tool2 的 TCP 沿着 一个 非线性 路径 移动 到位置 p1， 速度数据 vmax， 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 ：

MoveJSync ToPoint [\ID] Speed [\T] Zone Tool [\WObj] ProcName

ToPoint ：

数据类型 ：robtarget

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：

by 张建辉， 韩鹏排版

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

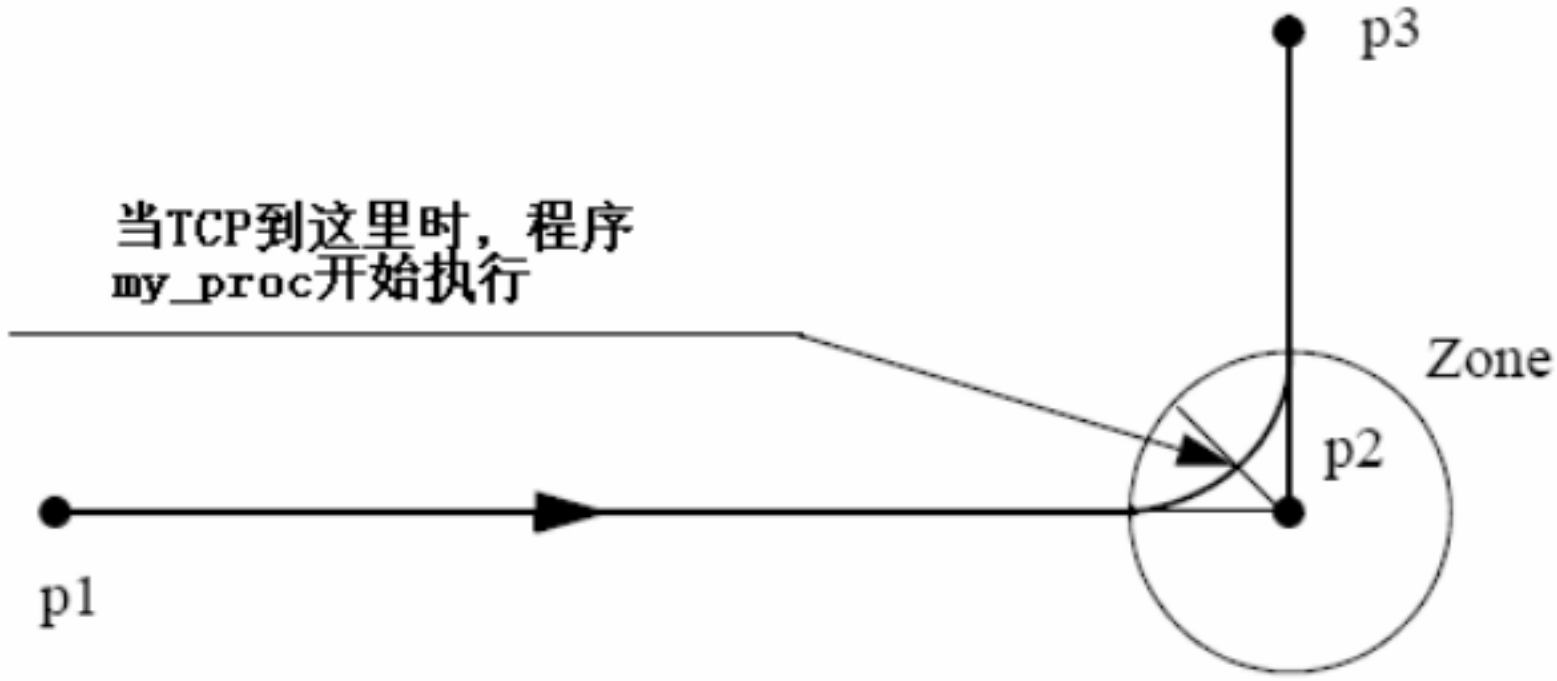
在目 标 点 的转 角路径 的 中间位置要执行 的RAPID 程序的 名 称 。

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 。

MoveJSync 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 MoveJ＋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 MoveJSync 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：

MoveJSync

[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) >

['\' Z ':=' < num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
通过 关节运动移动机 器人	第 226 页 MoveJ－通过 关节运动移动机 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

1.96.MoveL －让机器人作直线运动

用途：

MoveL 用来 让 机 器人 TCP直 线运动 到给 定的 目 标 位置 。 当 TCP仍旧 固 定的 时候 ， 该指令也可以 重新 给 工具 定方 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。

也可以参看第 238 页 更 多范例。

例 1 MoveL p1, v1000, z30, tool2;

Tool2 的 TCP 沿 直 线运动 到位置 p1， 数度数据 为 v1000， zone 数据 为 z30。

例 2 MoveL *, v1000\T:=5, fine, grip3;

Grip3 的 TCP 沿 直线运动 到存储在 指令 中的 停止点 （用 * 标 记）。整 个的运动 过 程 需时 5 秒。

项目 ：

MoveL [\Conc] ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool [\WObj] [\Corr]

[\Conc] ：

并 发事件

数据类型 ：switch

当 机 器人 运动的同 时， 后续 的指令 开始执行 。该 项目通常不 使用， 但是当 使用 飞点 （flyby points）时，可以用来 避免由 CPU 过载引 起的 不想要 的 停止 。当 使用 高速度 并且 编 程 点相距较近时这是很 有用的。例如， 当不要求与 外部设 备通讯 或外部设 备 和机 器人通讯 同步的 时候 ， 这 个项目 也 很有用。

使用 项目 \Conc 的 时候 ， 连续 的运动指令的 数量限 制为 5 个。 在包括 StorePath— RestorePath 的程序 段中不

允许 使用带有 \Conc 项目 的运动指令。

如果 不使用该 项目，并且 ToPoint 不是停止点 ，在机 器人到达 程序 zone 之前 一 段时间后续 指令 就开始执行了。

在 多运动系 统中 的坐标同步运动 中不 能使用该 项目 。

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

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 。

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 。

[\V] ：

速度

数据类型 ：num

该 项目 用来 在 指令 中直接 指定 TCP 的 速度 ，单 位 mm/s。 它用来 代替 速度数据中相 应 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 。

[\Z] ：

Zone

数据类型 ：num

该 项目 用来 在 指令 中直接 指定机 器人 TCP 的 位置 精 度。转 角路径 的 长度 单 位是 mm，它代替 zone 数据中相应 的 zone。

by 张建辉， 韩鹏排版

[\Inpos]：

到位

数据类型 ：stoppointdata（停止点数据 ）

该 项目 用来指定机 器人 TCP 在停止点位置 的 收敛 性 判别 标 准 。该 停止点数据 代替 在 zone 参 数中 指定的 zone。

Tool：

数据类型 ：tooldata

当 机 器人 运动的 时候 使用的 工具 。TCP 是 移动 到 指定的 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。

[\Corr] ：

改 正

数据类型 ：switch

如果使用该 项目 的 话， 通过 CorrWrite 指令 写到改 正入口 的 改 正数据 将被 添加 到路径 和 目 标位置 。

程序 执行 ：

机 器人 和外部单元 按照 下列 步 骤 运动：

l 工具 的TCP按照 程序 中的 速度 匀 速直 线运动。

l 工具沿着路径 以相 等的 间 隔 重新 定 向。

l 为 了 和机 器人 轴 在 同一 时间到达目 标位置， 非 并 列 的外部轴 按照 匀 速度执行 。

重新 定向或者外部轴如果 不能 达到 程序 中的 速度 ， TCP的 速度 将减小 。

当 运动 路径 转 到下一 段的时候， 通常 会产生 转 角路径 。如果 在zone数据中 指定 了停止点 ，只有 当 机 器人 和外部轴 到达 合适 的 位置 的 时候，程序 执行 才会继 续。

更 多范例：

如 何 使用 改 指令的 更 多范例说明如下：

例1 MoveL *, v2000 \V:=2200, z40 \Z:=45, grip3;

Grip3 的TCP线性移动 到存储在 指令 中的 位置 。该运动 执行时 的 数据 为v2000和 z40。 TCP的 速度 和 zone大小分别 是 2200mm/s和45mm。

例2 MoveL p5, v2000, fine \Inpos := inpos50, grip3;

by 张建辉， 韩鹏排版

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

by 张建辉， 韩鹏排版

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

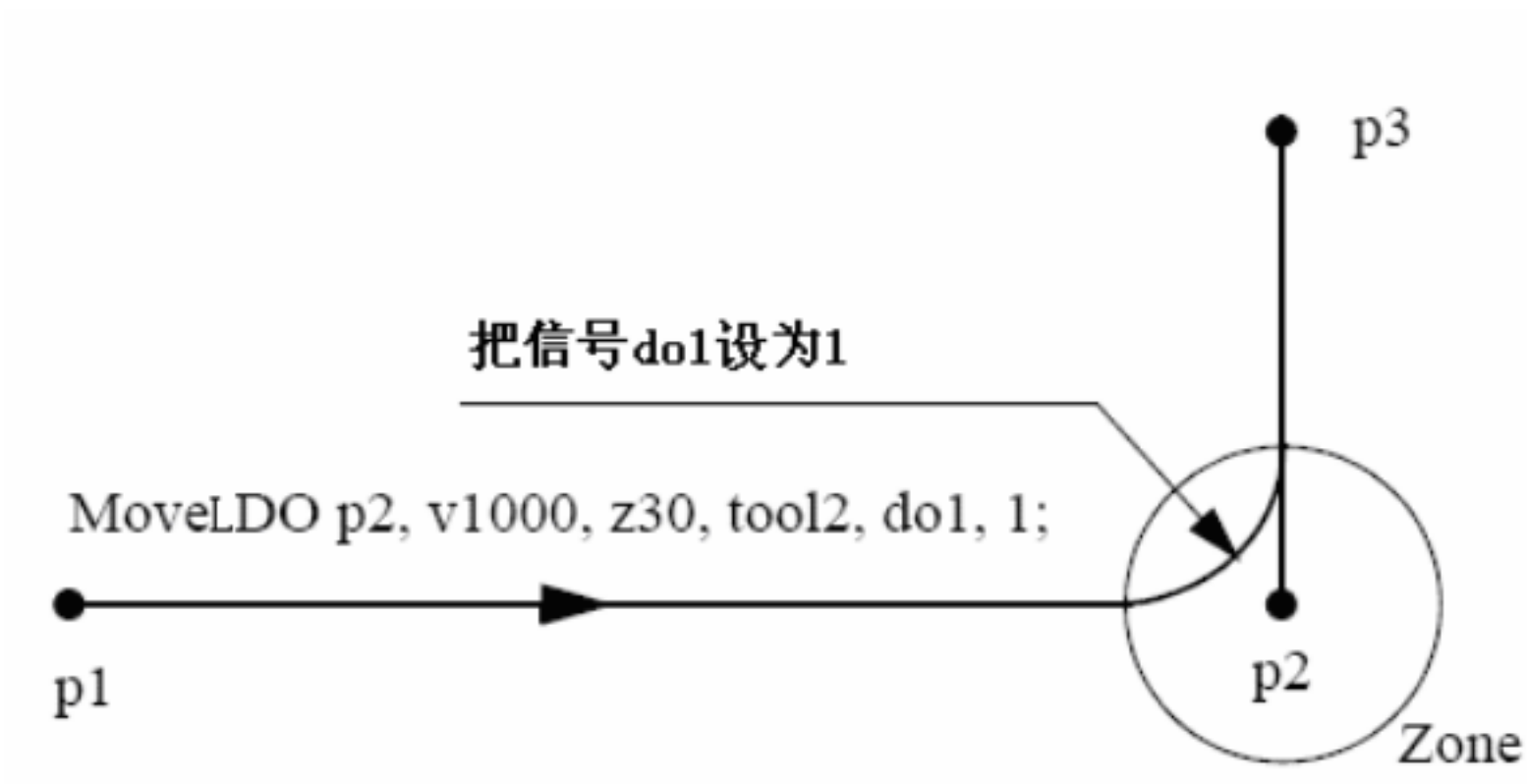
期望 的信号 数 值（ 0或者 1)。

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。

by　张建辉，　韩鹏排版

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 ：robtarget

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

by 张建辉， 韩鹏排版

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

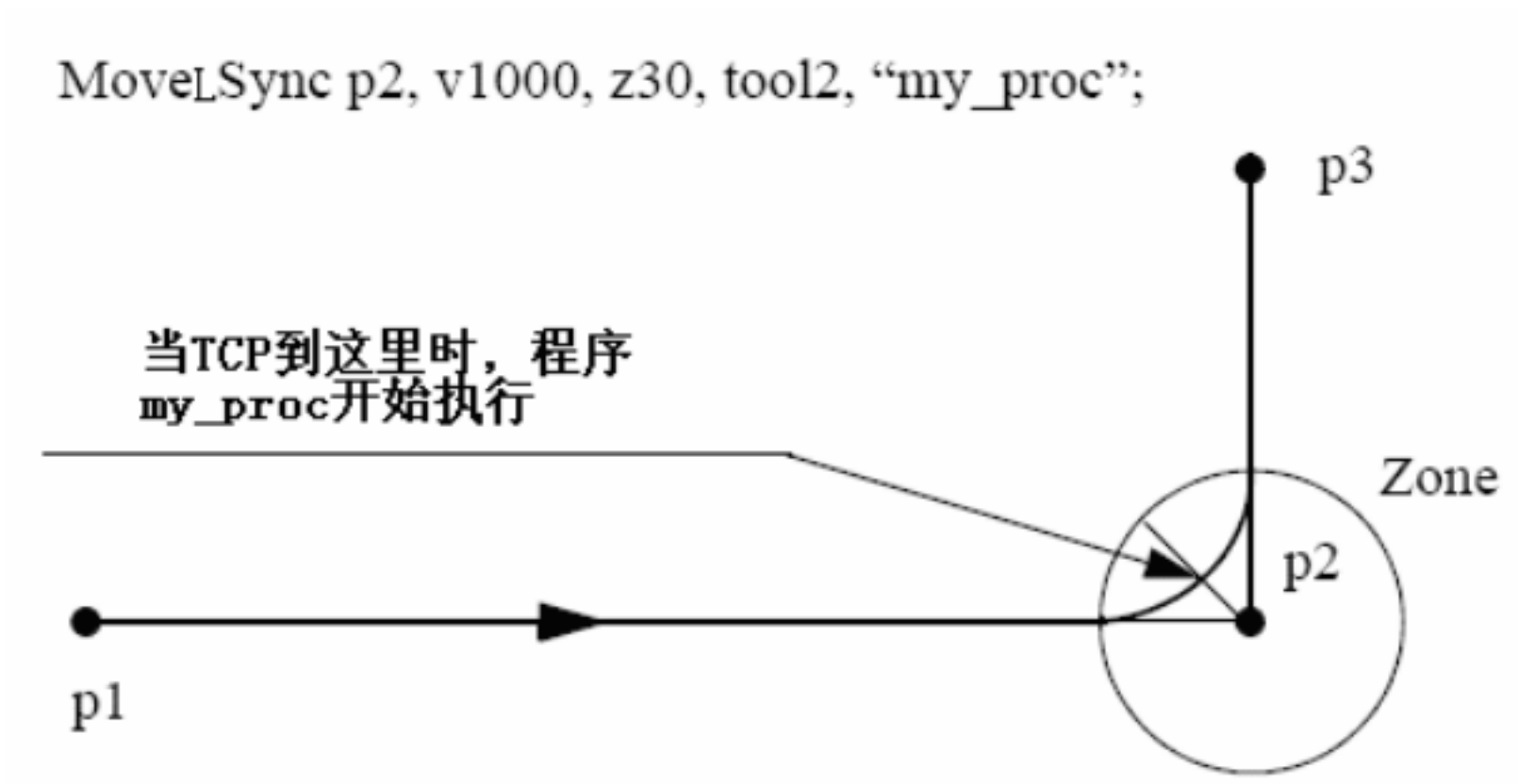
数据类型 ：string

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。



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

by 张建辉， 韩鹏排版

[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)>] ';'

[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：

l 指 出 机 器人 的 TCP 是 工作 区域 中 的一个明 确 的部分。

l 限 制机 器人 的 工作 区域， 阻 止和 工具 的碰撞 。

l 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。

by 张建辉， 韩鹏排版

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 低的角点的位置 （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：

by 张建辉， 韩鹏排版

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 robtarget 中使用 trans 组件，即 p1.trans 作为项目）。

语法 ：

WZBoxDef

[[‘]nside] | [‘ Ɔutside] ’,’

[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 robtarget 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 wzone;
…
PROC …
    WZLimSup \Temp, wzone, volume;
    MoveL p_pick, v500, z40, tool1;
    WZDisable wzone;
    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 wzone 内部。

by 张建辉， 韩鹏排版

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 恒 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 ':= ']<wztemporaty 类型 的变量或者 恒 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。

如果使用可选项目 \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。

by 张建辉， 韩鹏排版

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 quip1，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。

by 张建辉， 韩鹏排版