ABB 机器人 RAPID 常用指令详解 -中文

1.88.MoveAbsJ — 把机器人移动到绝对轴位置

用途：

MoveAbsJ（绝对关节移动）用来把机器人或者外部轴移动到一个绝对位置，该位置在轴定位中定义。

使用实例：

l 终点是一个单一点

l 对于 IR6400C 中的不明确的位置，例如携带超过机器人范围的工具运动。

MoveAbsJ 指令中机器人的最终位置，既不受工具或者工作对象的影响，也不受激活程序更换的影响。但是机器人要用到这些数据来计算负载、TCP 速度和转角点。相同的工具可以被 用在相 邻的运动指令中。

机器人和外部轴 沿着 一个 非直线 的路径 移动到 目标 位置。 所有 轴在同一 时间 运动到 目标 位置。

该指令 只能被 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 207 页 更 多范例。

例 1 MoveAbsJ p50, v1000, z50, tool2;

机器人 将 携带工具 tool2 沿着 一个 非线性路径 到绝对轴位置 p50， 以 速度数据 v1000 和 zone 数据 z50。

例 2 MoveAbsJ *, v1000\T:=5, fine, grip3;

机器人 将 携带工具 grip3 沿着 一个 非线性路径 到一个 停止 点，该 停止 点在指令中作 为一个绝对轴位置 存储（用 * 标示 ）。整 个运动 需 要 5 秒钟 。

项目：

MoveAbsJ [\Conc] ToJointPos [\ID] [\NoEOffs] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool [\Wobj]

[\Conc]:

并发事件

数据 类型 ： switch

当 机器人 正 在移动的 时候执行 的 后续 指令。该 项目通常 不使用，但是 当 和外部 设备通讯 、不 需 要同 步 的 时候 可 以 用来 缩短循环周期 。

当 使用 项目 \Conc 的 时候 ， 连续 运动指令的数 量限制为 5。在 包含 StorePath-RestoPath 的程序 段 中不 允许包含项目 \Conc 的运动指令。

如 果 该项目忽略并且 ToJointPos 不是一个 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 。

by 张建辉， 韩鹏排版

该 项目 不能 用在 多 运动 系统 的坐标 同步 运动中。

ToJointPos：

到 达的关节位置。

数据 类型 ：jointtarget

机器人和外部轴的绝对　　目标 轴位置。 它被 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　　*标示）。

[\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 。

[\NoEOffs] ：

没 有外部 偏移 量

数据 类型 ：switch

如 果项目 \NoEOffs 设为 1，MoveAbsJ 运动 将 不受外部轴的激活 偏移 量 的影响。

Speed：

数据 类型 ：speeddata

运动 所用的速度数据。速度数据定义　　了 TCP、工具 再 定位和外部轴的速度。

[\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，它替代 在速度数据中指定的相　　应 的速度。

[\T] ：

时间

数据 类型 ：num

该 项目 用来指定机器人运动的　　总 时间，单位 秒。 它 替代 相 应 的速度数据。

Zone：

数据 类型 ：zonedata

运动的 zone 数据。 Zone 数据 描述 了产生 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 度用 毫米给出 ，　替代　zone 数据中指定的相 应 数据。

[\Inpos] ：

到位

数据 类型 ： stoppointdata（停止 点数据）

改 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

运动过程中　所 携带的工具。

TCP 的位置和工具的负载在工具数据中定义。　　TCP 位置用来计算运动的速度和转角　　路径 。

[\Wobj] ：

工作对象

数据 类型 ： wobjdata

在运动过程中使用的工作对象。

如 果 机器人 抓 着工具的 时候，该项目 可 以忽略 。但是，如 果 机器人 抓 着 工作对象，　也就 是说 工具是 静 止 的，或者带 有 外部轴，　那么 该 项目必 须 指定。

在 有并 列 工具或者　有并 列 外部轴的　情况 下，　系统 使用该数据计算运动的速度和转角　　路径 ，该数据在工作对象中定义。

程序执行 ：

MoveAbsJ 运动不 会 受激活的程序转移的影响，　　并且 如果 使用 了可 选 项目 \NoEOffs ，　将 没有 外部轴的　偏移。如 果 不使用 \NoEOffs ，外部轴的　目标 位置 将 会 受到激活的外部轴　偏 移的影响。工具　按照 轴角度 插补 移动到绝对轴 目标 位置。这 就 是 说 每一个轴 都按照固　定的速度运动，　并且所有　轴 都 在同一 时间 到达目标 位置，这　样 就形成一个 非线性 的 路径 。

总 的来 说，　TCP 大约按照编　程的速度运动。在　 TCP 运动的同　时，　工具 重新 定 向，　并且 外部轴也在运动。如果 重新 定 向的或者外部轴的程序要　 求 的速度不 能达 到，　TCP 的速度 将被 减小 。

当 转换到 路径 的下 一 段 的时候通常　会产生 转角 路径 。如 果停止 点在　Zone 数据中指定，　只有 在机器人和外部轴到 达 合适 的轴位置的　时候 程序 才 能 继续执行 。

更多范例 ：

关于如 何 使用该指令，更　多 范例 说明如 下 ：

by　张建辉，　韩鹏排版

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

Grip3 沿着 一个 非线性路径 运动到一个 存储 在指令中的一个绝对轴位置。　　执行 的运动数据 为 v2000 和 z40。TCP 的速度 大小 是 2200mm/s， zone 的 大小 是 45mm。

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

Grip3 沿着 一个 非线性路径 运动到绝对轴位置 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 位置。 它等待条 件满足 最多等 2 秒。参看 stoppointdata 类型 的 预定义数据 inpos50。

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

Grip3 沿着 一个 非线性路径 运动到一个 存储 在指令中的一个绝对轴位置。　　当 机器人运动的 时候，也执行了并发 的 逻辑 指令。

例4　　MoveAbsJ \Conc, * \NoEOffs, v2000, z40, grip3;

和 以上的指令相同的运动，但是 它不受外部轴的激活的 偏移 量的影响。

例5　　GripLoad obj_mass;

MoveAbsJ start, v2000, z40, grip3 \Wobj:=obj;

机器人把和 固定工具 grip3 相关的工作对象 obj 沿着 一个 非线性路径 移动到绝对轴位置 start。

限制：

为了能 够后台运行 中包括指令 MoveAbsJ，并且 避免 单一点和 模糊区 的问题，并发 指令 满足 以下 的要 求是很必要的（ 参看下 图）

下 图显 示了后 台运 行 MoveAbsJ 指令的一些 限制。

语法：

MoveAbsJ [‘ ’Conc ‘; ’] [ToJointPos ’:= ’] < 关节 目标 表达 式 （ IN ） >

[‘ ’ID ‘:= ’<identno 类型 的 表 达 式(IN)>] [‘ ’NoEOffs] ‘; ’

[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)>] ‘; ’

[‘ ’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(PRS)>] ‘; ’

相关信息：

by 张建辉， 韩鹏排版

相关信息	参看
其它定位指令	RAPID 参考手册—RAPID 概述，RAPID 摘要—运动部分
关节目标的定义	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
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
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

1．89．MoveC —让机器人做圆周运动

用途：

该指令用来 让机器人 TCP 沿圆周运动到一个 给定的目标点。在运动过程中，相对 圆的方向通常保持不变。

该指令 只能在主任务 T_ROB1 中使用，在 多运动系统中的运动 任务中使用。

基本范例：

该指令的 基本范例说明如下：

也可 参看第 212 页更多范例。

例 1　　Move p1, p2, v500, z30, tool2;

Tool2 的 TCP 圆周运动到 p2，速度数据位 v500, zone 数据为 z30.圆由开始点、中间点 p1 和目标点 p2 确定。

例 2　　MoveC *, *, v500 \T:=5, fine, grip3;

Grip3 的 TCP 沿圆周运动到 存储在指令中的 fine 点（第二个*标记）。中间点也存储在指令中（第一个*标记）。

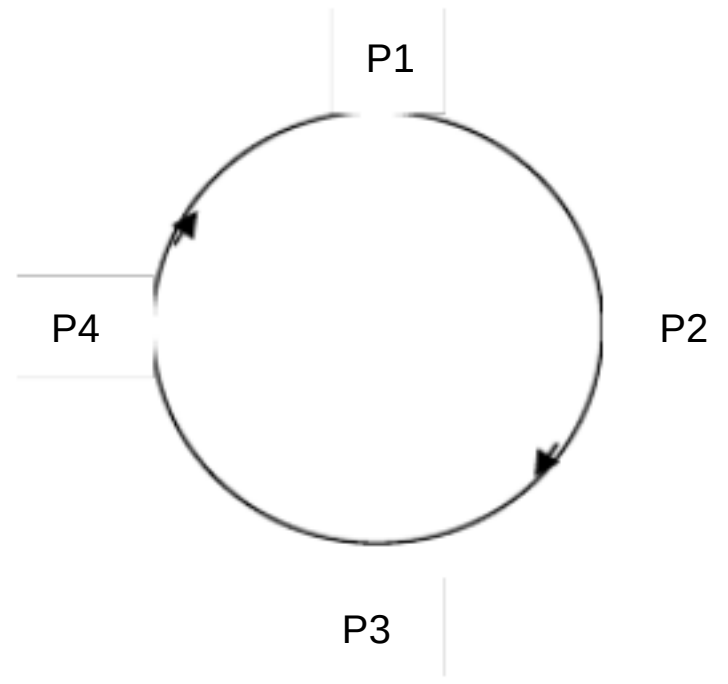
整个运动需要 5 秒钟 。

例 3　　MoveL p1, v500, fine, tool1;

MoveC p2, p3, v500, z20, tool1;

MoveC p4, p1, v500, fine, tool1;

下图说明了怎么用两个 MoveC 指令画一个完整的圆。



项目：

MoveC [\Conc] CirPoint 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 和外部 设备通讯并且　外部 设备 和机器人 通讯 不要 求同 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 。

CirPoint ：

数据 类型 ：robtarget

机器人的 圆 轴 上 的中 间 点。这是 圆 轴上处 于起点和终点 之间 的点。 为了 获得 最 好 的精 度，最 好 选择起 点和终点的中 间位置 附近 的点。如 果太 接 近起 点或者终点，机器人　将 会 报警 。中 间 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）。不使用外部轴的位置。

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

by　张建辉，　韩鹏排版

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 再定位和外部轴的速度。

[\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。它代替 速度数据中指定的相 应的速度。

[\T] ：

时间

数据 类型 ：num

该 项目 用来指定机器人和外部轴运动的 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 度以 毫米 为单位 给出， 它 代替 zone 数据中指定的 zone。

[\Inpos] ：

到位

数据 类型 ：stoppointdata（停止 点数据）

改 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

运动过程中 所 使用的工具。 TCP 是运动到指定 目标 的点。

[\Wobj] ：

工作对象

数据 类型 ：wobjdata

机器人在指令中定位的相关到的工作对象。

该 项目 可 以忽略 ，如 果忽略了 ，定位相关到 世界 坐标系 。在 另 一方面 ，如 果 使用 了 静 态 TCP 或者 并 列 外部轴， 为了执行 相关到工作对象的 圆 周 ，该 项目必 须 被指定。

[\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 以下说 明移动到 目标 位置：

l 工具的 TCP 按照 程序中的定 常速度作 圆 周 运动。

l 工具 按照 定常 速度 重新 定向 ，从 开始 位置的 方 向到 目标 点的 方 向。

l 重新 定 向 相对于 圆 周路径执行 。因此 如 果开始 点和 目标 点的 方 向 相对于 路径 是相同的，在移动过程中相对 方 向保 持 不变（参看下 图）。

下 图 说明 了 圆 周 运动过程中的工具 方 向。

圆 周 点的 方 向没 有 到达， 它只 是用来 区别重新 定 向 中 两个可 能 的方 向。沿着路径 重新 定向 的精 度只 取决 于开始 点和 目标 点的 方 向。

圆 周 运动过程中的工具 方 向 的不同 模式 在指令 CirPathMode 中 有 描述 。

非并 列的外部轴 以 定 常 速度 执行 ，目 的是和机器人轴同 时 到达目标 点。 圆 周 点中的位置 没有 用到。

如 果 重新 定 向 或者外部轴不 能达 到程序中的速度， TCP 得 速度 将被 减小 。

当 运动转换到 路径 中的 下 一段 的 时候通常 会产生 转角 路径 。如 果停止 点在 zone 数据中指定，在机器人和外部轴到 达 合适 位置的 时候 ，程序 才 能 继续执行 。

更多范例：

如 何 使用该指令 得 更 多 范例 说 明如 下 ：

例 1 MoveC *, *, v500 \V:=550, z40 \Z:=45, grip3;

Grip3 的 TCP 圆 周 运动到 存储 在指令中的位置。运动中把数据 设 定到 v500 和 z40 执行 ；TCP 的速度是 550mm/s， zone 的大小 是 45mm。

例 2 MoveC p5, p6, v2000, fine \Inpos := inpos50, grip3;

Grip3 的 TCP 圆 周 运动到 停止 点 p6。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

人认为它到达该点。它等待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　　MoveC \Conc, *, *, v500, z40, grip3;

Grip3 的 TCP 圆周运动到指令中存储的位置。圆周点也存储在指令中。当机器人移动的时候，执行后续逻辑指令。

例4　　MoveC cir1, p15, v500, z40, grip3 \Wobj :=fixture;

Grip3 的 TCP 经过圆周点 cir1 圆周运动到位置 p15。这些位置在 fixture 的对象并列系统中指定。

限制：

对于 cirPoint 和 Topoint 如何放置有一些限制，如下图描述：

- 起点和 ToPoint 之间的最小距离是 0.1 毫米。
- 起点和 CirPoint 之间的最小距离是 0.1 毫米。
- 从起点到 CirPoint 和 ToPoint 之间的最小角度是 1 度。

在接近这些限制的时候，精度将会很差，即如果圆的起点和 ToPoint 相距较近，圆倾斜引起的缺陷可能远大于编程点所使用的精度。

确保机器人在程序执行过程中可以到达 Circle Point（圆周点），必要的话把圆再分段。

当机器人停止在圆周路径上，执行模式从向前到向后得改变，或者相反，是不允许的，并且将导致错误信息。

警告！

当 TCP 在圆周点和终点之间的时候，MoveC 指令（或者任何其它包括圆周运动的指令）不允许从开头执行。否则机器人将不能执行编程的路径（从和编程路径方向不同的方向绕圆周路径定位）。

语法：

MoveC [‘ ’Conc ‘; ’] [CirPoint ’:= ’] <robtarget 类型的表达式(IN)> ‘; ’

[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
写到改 正入口	第 67 页 CorrWrite －写 到改正 入口
在圆周 运动中工具 重新 定向	第 32 页 CirPathMode －在圆 周路径 中工具 重新 定向
运动 综 述	RAPID 参考手 册－RAPID 概述，运动和 I/O 原理 部分
并列系统	RAPID 参考手 册－RAPID 概述，运动和 I/O 原理－ 并列系统 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

1.90． MoveCDO －圆周移动机器人并且在转角处设置数字输出

用 途：

MoveCDO （圆周 移动数 字输 出）用来把 TCP 圆 周移动到一个 给 定的 目标 点。指定的数 字输 出 在目标 点的转角 路径 的中 间被 置位 /复 位。在运动过程中，相对于 圆 周的 方 向 通常 保 持 不变。

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 MoveCDO p1, p2, v500, z30, tool2, do1, 1;

Tool2 的 TCP 圆周 移动到位置 p2,速度数据 v500 和 zone 数据 z30。圆 周由开始 点、 圆周点 p1 和 目标 点 p2 确定。在转角 路径 p2 的中 间 位置 设置 输 出 do1。

项目 ：

MoveCDO CirPoint ToPoint [\ID] Speed [\T] Zone Tool [\Wobj] Signal V alue

CirPoint ：

数据 类型 ：robtarget

机器人的 圆周 点。 圆周 点是 圆周 上开始 点和 目标 点之间 的一个位置。 为了 获得 最好的精 度， 它 最好处 于开

始点和 目标 点一 半的位置。如 果它 太靠近 开始 点或者 目标 点，机器人 将给出 一个 警告 。圆周 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）。不使用外部轴的位置。

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 须用在 并 列了同步 运动的 多运动 系统 中，不 允许 在其它任 何条 件下 使用。

在 所有 协 作的程序 任务 中，指定的 ID 号 码必 须 相同。 ID 号保证 了 在 routine 中运动不 会混 淆。

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 和外部轴的速度。

[\T] ：

时间

数据 类型 ：num

该 项目 用来指定机器人和外部轴运动的 总 时间，单位 秒。 它 用来 替 换相 应 的速度数据。

Zone：

数据 类型 ：zonedata

运动的 zone 数据。 Zone 数据 描述产生 的转角 路径 的大小 。

Tool ：

数据 类型 ：tooldata

当 机器人运动 时时 用的工具。工具中 心 点 就是运动到 目标 点的 那 个点。

[\Wobj] ：

工作对象

数据 类型 ：wobjdata

工作对象（对象 坐标系统 ），就 是在指令中机器人相关到的对象。

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 止 TCP 或者 并 列 的外部轴， 为了执行 相关到工作对象的 圆 周，该 项目必 须指定。

Signal：

数据 类型 ：signaldo

by 张建辉， 韩鹏排版

要改变的数字输出信号的名称。

Value：

数据类型：dionum

期望的信号的数值（0 或者 1）。

程序执行：

关于圆周运动得更多信息参看指令 MoveC。

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 MoveC 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 MoveC＋SetDO。但是当在指令 MoveCDO 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。

限制：

按照指令 MoveC 的常规限制。

语法：

MoveCDO [CirPoint ‘:= ’] <robtarget 类型的表达式（IN）> ‘; ’

[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
圆周运动机器人	第 209 页 MoveC －圆周移动机器人
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

1.91． MoveCSync —圆周移动机器人，并且执行一个 RAPID 程序

用 途 ：

MoveCSync（同 步 圆周 移动）用来 圆周 移动 TCP 到一个 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。在运动过程中，相对于 圆 周 的 方向 通常 保持 不 变。

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2 MoveCSync p1, p2, v500, z30, tool2, “ proc1 ”;

Tool2 的 TCP 圆周 移动到位置 p2,速度数据 v500 和 zone 数据 z30。圆 周 由 开始 点、 圆 周 点 p1 和 目标 点 p2 确定。在转角 路径 p2 的中 间 位置程序 proc1 开始执行 。

项目 ：

MoveCSync CirPoint ToPoint [\ID] Speed [\T] Zone Tool [\Wobj] ProcName

CirPoint ：

数据 类型 ：robtarget

机器人的 圆 周 点。 圆周 点是 圆 周 上 开始 点和 目标 点 之间 的一个位置。 为了 获得 最好的精 度， 它 最好处 于 开始 点和 目标 点一 半 的位置。如 果它 太靠近 开始 点或者 目标 点， 机器人 将给出 一个 警告 。圆 周 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）。不使用外部轴的位置。

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 须用在 并 列 了同步 运动的 多运动 系统 中，不 允许 在 其它任 何条 件下 使用。

在 所有 协 作的程序 任务 中，指定的 ID 号 码必 须 相同。 ID 号保证 了 在 routine 中运动不 会混 淆。

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 和外部轴的速度。

[\T] ：

时间

数据 类型 ：num

该 项目 用来指定机器人和外部轴运动的 总 时间，单位 秒。 它 用来 替 换相 应 的速度数据。

Zone：

数据 类型 ：zonedata

运动的 zone 数据。 Zone 数据 描述产生 的转角 路径 的大小 。

Tool：

数据 类型 ：tooldata

当 机器人运动 时时 用的工具。工具中 心 点 就 是运动到 目标 点的 那 个点。

[\Wobj] ：

工作对象

数据 类型 ：wobjdata

工作对象（对象 坐标系统 ），就 是在指令中机器人相关到的对象。

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 止 TCP 或者 并 列 的外部轴， 为了执行 相关到工作对象的 圆 周，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 ：

关于 圆周 运动 得更 多 信息 参看 指令 MoveC。

当 TCP 到达 MoveCSync 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。

下 图 说明在转角 路径 的中 间 位置用 户 定义的 RAPID 程序的 执行 。

by 张建辉， 韩鹏排版

对于 停止 点， 我们推荐 使用“ 正常 ”的编程顺序， 即 MoveC ＋其他 RAPID 程序。

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 ：

按照 指令 MoveC 的常 规限制 。

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 MoveCSync 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 ：

MoveCSync [CirPoint ‘:= ’] <robtarget 类型 的 表 达 式（IN） > ‘; ’

[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 ‘:= ’] <sting 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	
圆 周 运动机器人	第 209 页 MoveC — 圆 周 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	

by 张建辉， 韩鹏排版