

川崎机器人控制器  
E 系列

# AS 语言参考手册

# Robot

川崎重工业株式会社

## 前言

本手册介绍了在川崎机器人控制器 E 系列中使用的 AS 语言。为了给 AS 系统的有效使用提供信息支持，本手册详细描述了 AS 系统概述、基本用法、数据类型、轨迹控制和所有的指令/命令。本手册未介绍机器人的操作过程，因此，有关机器人操作，请参阅操作手册。

必须仔细地阅读本手册及其下列相关手册。只有当全部手册完整阅读和理解后，才能使用机器人。

1. 安全手册
2. 手臂的安装和连接手册
3. 控制器的安装和连接手册
4. 外部 I/O 手册 (供连接周边装置之用)
5. 检查与维修手册

本手册内容是以机器人已经按照上述手册安装和连接好为基础进行介绍的。

本手册包含有选件功能的信息，但按照规格的不同，机器人可能并不包含本手册介绍的每个选件功能。在操作中碰到任何未描述的问题或情况时，请联络川崎公司。请参阅本手册背页上的联系信息。

- 
1. 本手册并不构成对使用机器人的整个应用系统的担保。因此，川崎公司将不会对使用这样的系统而可能导致的事故、损害和(或)与工业产权相关的问题承担责任。
  2. 川崎公司郑重建议:所有参与机器人操作、示教、维护、维修、点检的人员，预先参加川崎公司准备的培训课程。
  3. 川崎公司保留未经预先通知而改变、修订或更新本手册的权利。
  4. 事先未经川崎公司书面许可，不可以将本手册全部或其中的一部分再版或复制。
  5. 请把本手册小心存放好，使之保持在随时备用状态。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，请和您的川崎公司代理商联络。

## 符号

在本手册中，带下述符号的内容应特别注意。

为确保机器人的正确安全操作、防止人员伤害和财产损失，请遵守下述方框符号表达的安全信息。

### 危 险

不遵守本标志内容，将会引起迫在眉睫的人身伤害或死亡。

### 警 告

不遵守本标志内容可能会引起人身伤害或死亡。

### 小 心

不遵守本标志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、处理、示教、操作和维护的注意信息。

### 警 告

1. 本手册给出的图表、流程和详细解释可能并不绝对精确和有效。当出现未说明的问题或疑问时，请和川崎公司联系。
2. 本手册中描述的与安全有关的内容，只适用于特定的机器人工作，并不适用于全部机器人的工作。为了确保全部工作的安全，请阅读和完整理解安全手册、全部相关的法律、规章和如同各章中安全说明的相关材料，并且为实际工作准备合适的安全措施。

## 目 录

前言 .....	i
符号 .....	ii
1.0 AS 概述 .....	1-1
1.1 AS 系统概述 .....	1-1
1.2 AS 系统的特点 .....	1-2
1.3 AS 系统的结构 .....	1-3
2.0 AS 系统 .....	2-1
2.1 AS 系统状态 .....	2-1
2.2 AS 系统开关 .....	2-2
2.3 AS 系统设置 .....	2-4
2.4 输入/输出控制 .....	2-5
2.4.1 终端控制 .....	2-5
2.4.2 外部存储装置 .....	2-6
2.5 安装终端软件 .....	2-7
2.5.1 安装 KCWIN32/KCWINTCP .....	2-8
2.5.2 安装 KRTERM .....	2-8
2.6 从个人计算机进行操作 .....	2-9
2.6.1 系统启动 .....	2-9
2.6.1.1 连接到 RS-232C 端口 .....	2-9
2.6.1.2 用以太网联接多台机器人 .....	2-11
2.6.2 上传和下载数据 .....	2-14
2.6.3 系统关闭 .....	2-14
2.6.4 KRTERM 软件的功能 .....	2-15
2.6.4.1 创建日志文件 .....	2-15
2.6.4.2 宏功能 .....	2-18
3.0 AS 语言的信息表达 .....	3-1
3.1 符号和规则 .....	3-1
3.2 位姿信息，数字信息和字符信息 .....	3-3
3.2.1 位姿信息 .....	3-3
3.2.2 数字信息 .....	3-6
3.2.3 字符信息 .....	3-8
3.3 变量 .....	3-9
3.3.1 变量(全局变量) .....	3-9
3.3.2 局部变量 .....	3-9
3.4 变量名 .....	3-11
3.5 定义位姿变量 .....	3-12
3.5.1 用监控指令定义 .....	3-12

3.5.2	用程序命令定义 .....	3-14
3.5.3	使用复合变换值 .....	3-14
3.6	定义实型变量 .....	3-18
3.7	定义字符串变量 .....	3-19
3.8	数学表达式 .....	3-20
3.8.1	运算符 .....	3-20
3.8.2	操作顺序 .....	3-21
3.8.3	逻辑表达式 .....	3-22
3.9	字符串表达式 .....	3-23
4.0	AS 程序 .....	4-1
4.1	AS 程序的类型 .....	4-1
4.1.1	机器人控制程序 .....	4-1
4.1.2	PC 程序(过程控制程序) .....	4-1
4.1.3	自动启动 .....	4-2
4.2	创建和编辑程序 .....	4-3
4.2.1	AS 程序格式 .....	4-3
4.2.2	编辑器指令 .....	4-4
4.2.3	编程流程 .....	4-5
4.2.4	创建程序 .....	4-5
4.3	执行程序 .....	4-8
4.3.1	执行机器人控制程序 .....	4-8
4.3.2	停止程序 .....	4-9
4.3.3	继续机器人控制程序 .....	4-10
4.3.4	执行 PC 程序 .....	4-10
4.4	程序的执行过程 .....	4-11
4.4.1	子程序 .....	4-11
4.4.2	带参数的子程序 .....	4-11
4.4.3	异步处理(中断) .....	4-12
4.5	机器人动作 .....	4-13
4.5.1	机器人动作和程序步骤执行的时序 .....	4-13
4.5.2	连续路径(CP)运动 .....	4-15
4.5.3	CP 运动的中断 .....	4-16
4.5.4	CP 开关和 ACCURACY、ACCEL、DECEL 命令之间的关系 .....	4-17
4.5.4.1	CP ON: 运动类型 1 (标准) .....	4-17
4.5.4.2	CP ON: 运动类型 2 .....	4-18
4.5.4.3	CP OFF .....	4-21
4.5.5	沿指定路径的运动 .....	4-21
4.5.5.1	设置负荷数据 .....	4-22
5.0	监控指令 .....	5-1
5.1	编辑器指令 .....	5-2
5.2	程序和数据控制指令 .....	5-15

5.3	程序和数据存储指令 .....	5-27
5.4	程序控制指令 .....	5-34
5.5	位姿信息指令 .....	5-43
5.6	系统控制指令 .....	5-48
5.7	二进制信号指令 .....	5-88
5.8	信息显示指令 .....	5-103
6.0	程序命令 .....	6-1
6.1	运动命令 .....	6-2
6.2	速度和精度控制命令 .....	6-15
6.3	夹紧控制命令 .....	6-32
6.4	形态命令 .....	6-39
6.5	程序控制命令 .....	6-42
6.6	程序结构命令 .....	6-55
6.7	二进制信号命令 .....	6-68
6.8	信息控制命令 .....	6-89
6.9	位姿信息命令 .....	6-98
6.10	程序和数据控制命令 .....	6-113
7.0	AS 系统开关.....	7-1
8.0	运算符 .....	8-1
8.1	算术运算符 .....	8-1
8.2	关系运算符 .....	8-2
8.3	逻辑运算符 .....	8-3
8.4	二进制运算符 .....	8-5
8.5	变换值运算符 .....	8-6
8.6	字符串运算符 .....	8-8
9.0	函数 .....	9-1
9.1	实数函数 .....	9-2
9.2	位姿值函数 .....	9-30
9.3	数学函数 .....	9-46
9.4	字符串函数 .....	9-49
10.0	PC 程序.....	10-1
11.0	示例程序 .....	11-1
11.1	程序的初始化设置 .....	11-1
11.2	码垛 .....	11-2

11.3	外部联锁 .....	11-4
11.4	工具变换 .....	11-7
11.4.1	工具变换值-1(当工具尺寸未知时) .....	11-7
11.4.2	工具变换值-2(当工具尺寸已知时) .....	11-9
11.5	相对位姿 .....	11-10
11.5.1	相对位姿的用法 .....	11-10
11.5.2	使用相对位姿编程的实例 .....	11-11
11.6	用 FRAME 函数的相对位姿 .....	11-13
11.7	设定机器人形态 .....	11-15
附录 1	信号编号的限制 .....	A1-1
附录 2	ASCII 码.....	A2-1
附录 3	欧拉 O, A, T 角 .....	A3-1
附录 4	错误信息表 .....	A4-1
附录 5	AS 语言一览表(按字母为序) .....	A5-1

## 1.0 AS 概述

川崎机器人由名为 AS 的软件系统控制。本章将对此 AS 系统的概要进行介绍。

### 1.1 AS 系统概述

在 AS 系统中，你可以和机器人进行通讯或用 AS 语言开发程序。AS 系统被编写在机器人控制单元的永久存储器里。在电源开启时，AS 系统启动并等待命令输入。

AS 系统按照给定的指令和程序控制机器人。在程序运行过程中，它也能执行好几种功能。以下功能可以在程序运行中使用：显示系统状态或定义机器人位姿(位置和姿态)变量、保存数据到外部存储设备、编写/编辑程序。



## 1.2 AS 系统的特点

在 AS 系统中，机器人是以事先操作准备的程序来控制 and 运行的，这里对该事先操作的任务进行说明。(示教再生方式)

AS 语言可以分成两种类型：监控指令和程序命令。

**监控指令：**用来写入、编辑和执行程序。它们在画面显示的提示符(>)后面输入，并且被立即执行。有些监控指令也可以作为程序命令在程序中使用。

**程序命令：**用来引导机器人的动作，在程序中监视或控制外部信号等。程序是程序命令的集合。

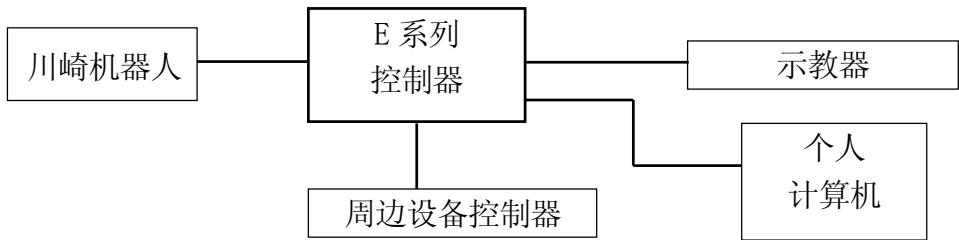
在本手册中，监控指令简称为指令，程序命令简称为命令。

AS 在如下方面有其独特的特点：

1. 可以使机器人沿着连续的路径轨迹运动。(称为：CP 运动：Continuous Path(连续路径)运动)
2. 提供有两种坐标系统，基础坐标系和工件坐标系。可以按两种坐标系移动机器人。
3. 坐标系可以按工作位姿的改变随进行平移或旋转。
4. 在示教位姿时，机器人可以保持工具的定向沿直线路径运动。
5. 程序可以自由命名和保存，而没有程序数量的限制。
6. 可以将每个操作定义为一个程序，并且可以将这些程序组合成一个复杂的程序。(子程序)。
7. 通过监控信号通过，在某个外部信号输入时，程序可以中断，挂起当前动作，并跳转到另一个不同的程序中。(中断)
8. 没有运动指令的过程控制程序(Process Control program, 简称 PC 程序)可以与机器人控制程序同时执行。
9. 程序和位姿数据可以显示在屏幕上，也能存储在 USB 闪存等设备上。
10. 编程工作可以在装有川崎终端软件(KRterm 或 KCwin32/KCwinTCP)的个人计算机上完成。(离线编程)

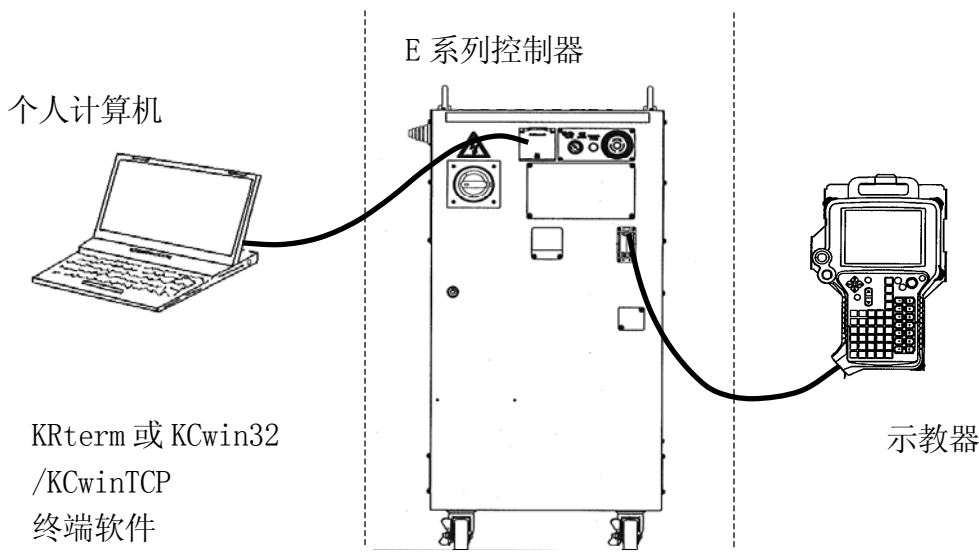
1.3 AS 系统的结构

川崎机器人控制器 E 系列由以下部分组成：



在连接一台装有川崎终端软件(KRterm 或 KCwin32/ KCwinTCP)的个人计算机到一台 E 系列控制器后，可以实现以下操作

- 编写 AS 指令和命令
- 向计算机保存，也可以从计算机装载数据。



个人计算机	控制器	示教器
<ul style="list-style-type: none"><li>· 输入 AS 指令</li><li>· 创造 AS 程序</li><li>· 保存/装载程序</li></ul>	日常操作	<ul style="list-style-type: none"><li>· 选择程序</li><li>· 显示程序名称和步骤</li><li>· 手动控制机器人</li><li>· 监视信号</li><li>· 设置再现条件</li><li>· 示教位姿数据</li><li>· 示教辅助数据（一体化示教）</li></ul>
<div><div>[ 注意 ]</div><div>个人计算机（PC）的监控软件需要 95/98/Me/2000/XP 操作系统（只有 KRterm 软件，需要 Windows 2000/XP/Vista 操作系统）。请准备好合适的操作系统。</div></div>		



## 2.0 AS 系统

本节说明 AS 系统状态、AS 系统开关和系统设置。

### 2.1 AS 系统状态

AS 系统具有下列三种模式：

#### 1. 监控模式

此模式控制并监视 AS 系统的执行，是 AS 系统中的基本模式。监控指令可以在此模式中执行。从此模式可以进入编辑模式(通过执行 EDIT 指令)或再生模式(通过执行 EXECUTE 指令)。

#### 2. 编辑模式

使用此模式可以创建程序或更改一个存在的程序。在此模式，系统仅可执行编辑指令。

#### 3. 再生模式

程序执行过程中系统就处于再生模式。在此模式下，可以处理从终端(如示教器或 PC 机)输入的指令。此时，在一定的周期内进行机器人运动控制计算。在此模式下，某些监控指令不能执行。请参阅 5.0 监控指令，在再生模式下允许输入的监控指令。

## 2.2 AS 系统开关

下述系统开关可以在 AS 系统中使用监控指令 SWITCH 来设置。为各开关设置的状态和条件可以从终端上进行检查和更改。

### 1. CHECK.HOLD(保持检查)

仅在 HOLD 状态下，决定是否接受由键盘输入的 EXECUTE, DO, STEP, MSTEP, CONTINUE 指令。

### 2. CP(连续路径)

启用或关闭连续路径运动。当此开关为 ON 时，机器人在两个动作段之间作平滑过渡。当该开关为 OFF 时，机器人在每一段动作路径终点处减速并停止。

### 3. CYCLE.STOP(循环停止)

决定当有外部保持信号输入停止机器人动作时，是否保持 CYCLE START(循环启动)为 ON 状态或关闭为 OFF。

### 4. MESSAGES(信息)

允许或禁止 PRINT 或 TYPE 指令向终端输出信息。

### 5. OX.PREOUT(输出信号预输出)

在一体化命令中设定为 OX 信号输出的时序。

### 6. PREFETCH.SIGINS(预取输入信号)

决定是否允许通过 AS 命令/指令来做信号输入和输出的早期处理。

### 7. QTOOL(工具变换)

决定是否只有当在再现模式下工具指令/命令或一体化命令执行时，更改工具数据，或是否允许即使在示教模式下按照一体化命令所示教的工具编号自动改变工具数据。

### 8. REP\_ONCE(Repeat Once, 再现单次)

当此开关为 ON 时，程序运行一次。当它为 OFF 时，程序连续不断运行。

### 9. RPS(Random Program Selection, 随机程序选择)

允许或禁止按照外部信号进行程序选择。

10. SCREEN(屏幕)

设置允许或禁止当要显示的信息多于一屏时的屏幕画面翻滚。

11. STP\_ONCE(单步)

设置程序是单步执行还是连续执行。

关于如何设置系统开关的更多详情，请参阅 5.6 系统控制指令 SWITCH, ON, OFF。

## 2.3 AS系统设置

以下系统设置可以根据需要用监控指令改变。

### 1. 调零 (ZZERO 指令)

ZZERO 用来设定机器人已知机械位置对应的编码器值。当更换伺服电机或维护编码器时，需要用该指令来调整编码器值(该指令仅用于维护)。

### 2. 夹紧设定 (HSETCLAMP 指令)

本设定在出厂前已经设定好。单头/双头和输出规格(关时为 ON/关时为 OFF)等的设定，可以用 HSETCLAMP 指令来改变。但是设定的改变仅仅影响软件，所以请确认其是否与硬件一致。

### 3. 输入输出信号的最大编号 (ZSIGSPEC 指令)

ZSIGSPEC 指令设置可用的输入和输出信号的最大编号。该设定在出厂前已设定好(这是个默认设定，具有软件的错误检查的功能，请确认其是否与硬件一致)。

### 4. 软件专用信号 (DEFSIG指令)

除了硬件专用信号之外，在软件中 I/O 信号也可用作专用信号(软件专用信号)。下面表格中的信号可用作专用信号。请注意，由于软件中 I/O 信号的数量为软件专用信号和通用信号数量总和，所以软件专用信号数量增多时，通用信号数量就会减少。

软件专用输入信号	软件专用输出信号
外部马达电源 ON	马达电源 ON
外部错误复位	错误发生
外部循环启动	自动
外部程序复位	循环启动
外部程序选择有效 ( JUMP-ON, JUMP-OFF, RPS-ON, 外部程序编号)	示教模式
外部暂停	原点位置 1, 原点位置 2
外部低速再现模式	电源 ON
	RGS0
	外部程序选择有效 (JMP_ST, RPS_ST)

## 2.4 输入/输出控制

### 2.4.1 终端控制

从终端输入的数据和指令首先被系统缓冲区接收。然后他们被监视器或程序读取，再反馈或显示在终端屏幕上。终端上可以输入最多 128 个字符，多出的输入字符将被忽略。

用 PRINT (打印) 和 TYPE (打字) 命令可以控制数据输出到终端。终端屏幕以 8 位为单位显示。除非 PRINT 或 TYPE 命令用 “/S” 格式代码设定输出格式，否则数据将在每个命令以后，以一个新行开始显示。(详细信息，请参阅 6.8 信息控制命令)

终端的输入和输出可以用下表所示的指令控制。这些指令叫做终端控制指令。**Ctrl** (控制键) 与不同的字母 (大小写均可) 同时按下。不同于其它 AS 指令，这些指令在输入后无需按 ENTER 键。

指令	功能
<b>Ctrl</b> + <b>S</b>	停止终端显示屏的翻滚。
<b>Ctrl</b> + <b>Q</b>	恢复由 <b>Ctrl</b> + <b>S</b> 停止的数据输出。
<b>Ctrl</b> + <b>C</b>	取消最后输入的行。
<b>Ctrl</b> + <b>H</b>	删除最后输入的字符 (退格)。
<b>Ctrl</b> + <b>M</b>	结束当前行的输入。
<b>Ctrl</b> + <b>L</b>	在当前输入行上显示以前输入的内容。最多可回显 7 次以前的内容 (前一行)。
<b>Ctrl</b> + <b>N</b>	显示用 <b>Ctrl</b> + <b>L</b> 命令显示的下一输入行的内容。该操作仅能在 <b>Ctrl</b> + <b>L</b> 执行了多于一次后使用 (下一行)。
<b>Backspace</b>	删除最后输入的字符。

TAB (**Ctrl** + **I** 或 **TAB**) (制表键) 输入将作为空格处理。



### 2.4.2 外部存储装置

如下指令可以将机器人存储器中的程序、变量和位姿信息数据保存到 PC 卡、软盘或计算机硬盘中。

1. 显示 USB 闪存的内容。(USB\_FDIR)
2. 保存机器人存储器的数据到个人计算机或USB闪存文件。(SAVE\*, USB\_SAVE)
3. 装载计算机或 USB 闪存驱动器的数据到机器人存储器。(LOAD, USB\_LOAD)
4. 删除个人计算机或 USB 闪存里的文件。(USB\_FDEL)

带有 USB\_的命令是对 USB 闪存的命令。

**注释\*** SAVE命令仅可在连接了计算机时使用。

请参阅 5.2 程序和数据控制指令、5.3 程序和数据存储指令。

2.5 安装终端软件

机器人是通过 AS 语言被个人计算机控制。为此，装载 KRterm, KCwin32 或 KCwinTCP 终端软件到个人计算机上，并把个人计算机连接到 E 系列控制器。各种软件的操作环境如下：

硬件	安装有80486以上的CPU，运行微软Windows操作系统
OS操作系统	<b>KCwin32/KCwinTCP</b> Microsoft Windows 95/98/NT4.0/2000/XP <b>KRterm</b> Microsoft Windows 2000/XP/Vista
已检测的计算机型号	<b>KCwin32/KCwinTCP</b> 东芝个人计算机Dynabook Satellite 2520 (Windows98) Compaq Armada 1500C (Windows98) IBM ThinkPad 365X (Windows95) <b>KRterm</b> DELL Inspiron 8000 (Windows2000) Lenovo ThinkCentre (WindowXP) Gateway GT5228J (Windows Vista)

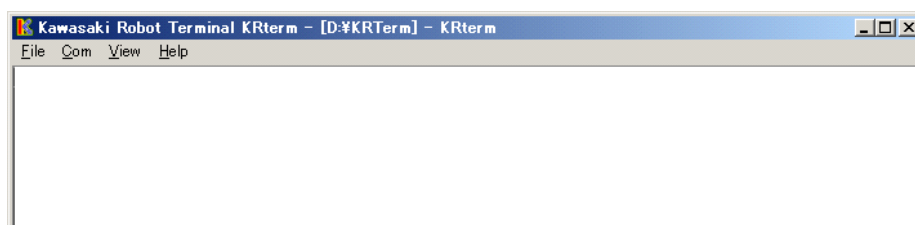
注\* 在没被检测的计算机型号软件可能会不正确地运转。

用 RS-232C 电缆来连接计算机和控制器，使一台单独的计算机能够控制一台单独的机器人。  
以太网连接使多台计算机能够控制多台机器人。

按照以下的步骤安装终端软件到个人计算机上。

### 2.5.1 安装 KCWIN32/KCWINTCP

安装由川崎公司提供的 KCwin32/KCwinTCP 软件的复制品到 Windows 个人计算机的文件中。在软件安装完成后，KCwin32/KCwinTCP 的图标被创建，请双击此图标。KCwin32/KCwinTCP 开始，以下的画面将出现。

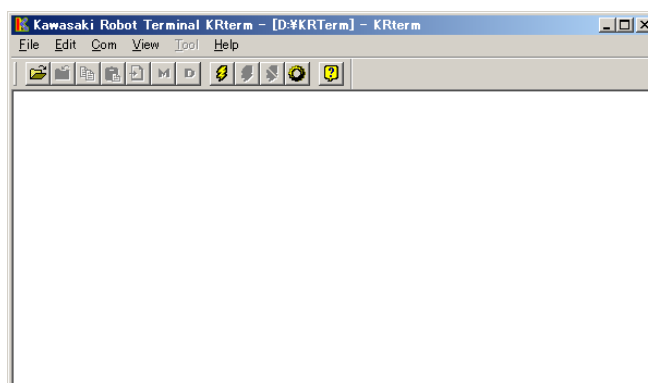


### 2.5.2 安装 KRTERM

复制由川崎公司提供的设置软件 KRterm (SetupE.exe) 到 Windows 个人计算机的文件中，并运行其软件。按照以下的安装指南来完成安装。



在软件安装完成后，KRterm 的图标被创建，请双击此图标。KRterm 开始，以下的画面出现。

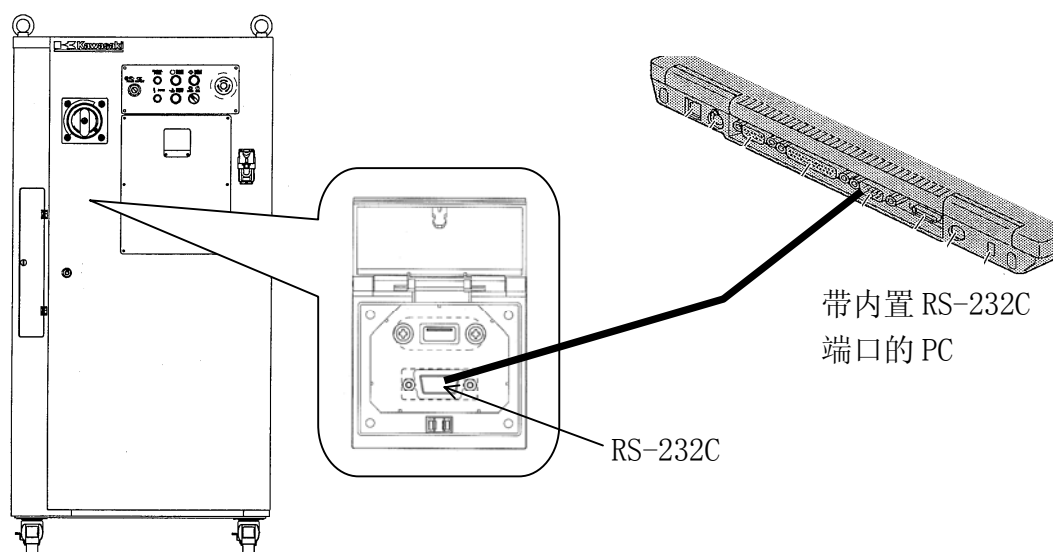


## 2.6 从个人计算机进行操作

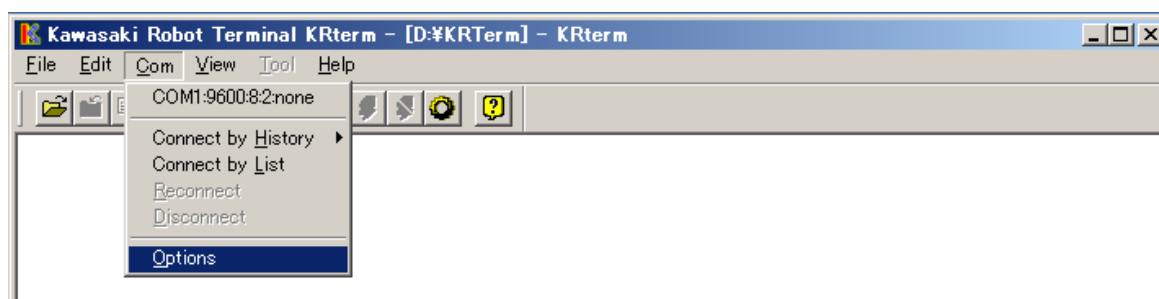
### 2.6.1 系统启动

#### 2.6.1.1 连接到 RS-232C 端口

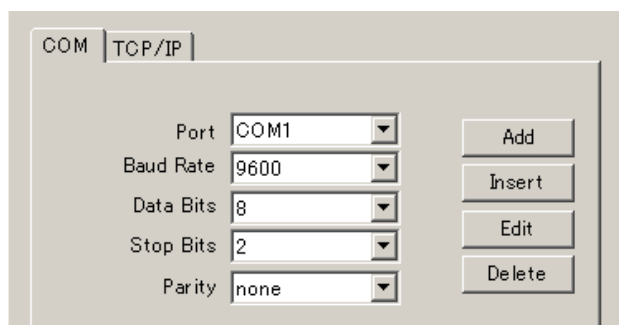
1. 在确认机器人控制器电源和微机电源都关闭的情况下，用 RS-232C 电缆将 PC 机和控制器连接起来。



2. 打开 PC 电源，启动终端软件 (KRterm 或 KCwin32/KCwinTCP, 下图为 KRterm 软件)。
3. 软件运行后，选择所使用的连接类型。从菜单条上选择 [Com(munication)] → [Options]。



4. 向〈Baud Rate〉，〈Data Bits〉，〈Stop Bits〉，〈Parity〉里分别输入 9600，8，2，“none”。对于 KRterm 软件，请选择〈COM〉并同样地设定参数，然后单击〈OK〉。



其次，选择[Com(munication)]→[Connect by List]，以下的画面出现，选择以上设定好的连接设定。点击〈OK〉。



对于 KCwinTCP 软件，设定参数如上，并点击〈OK〉。

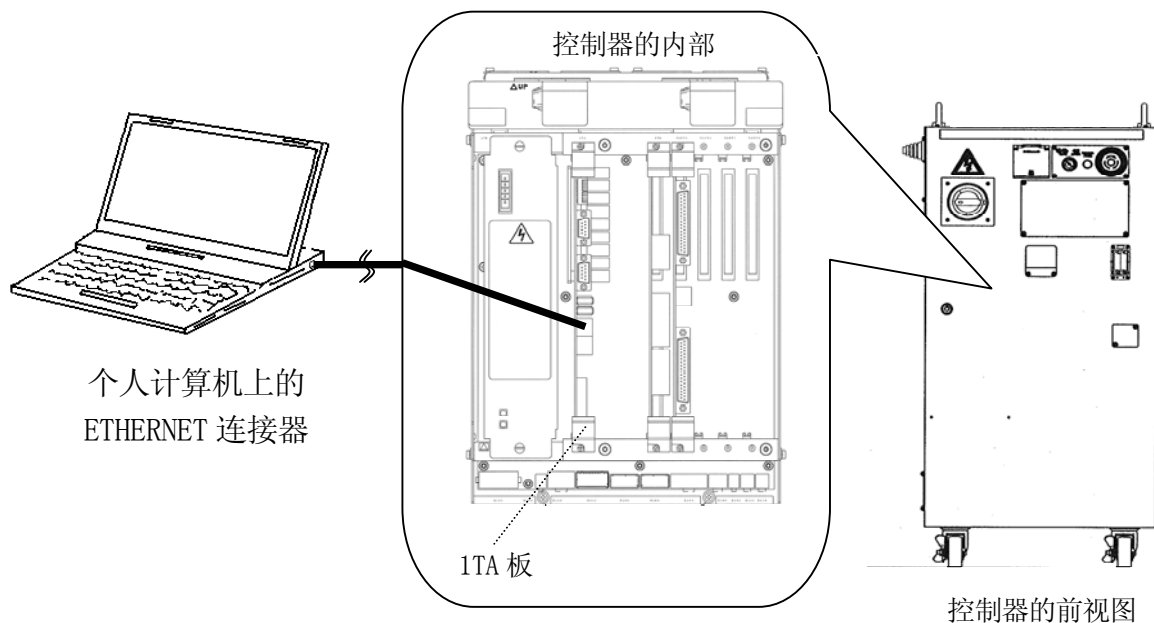
5. 开启控制器的控制器电源。  
(请参阅“操作手册” 3.1 电源开的步骤)
6. 在提示符“>”后，KRterm 的初始化界面将显示在屏幕上。

如果控制器电源在连接到 PC 机之前已开启，屏幕上仅出现提示符“>”，而不出现初始化画面，但 KRterm 照常工作。

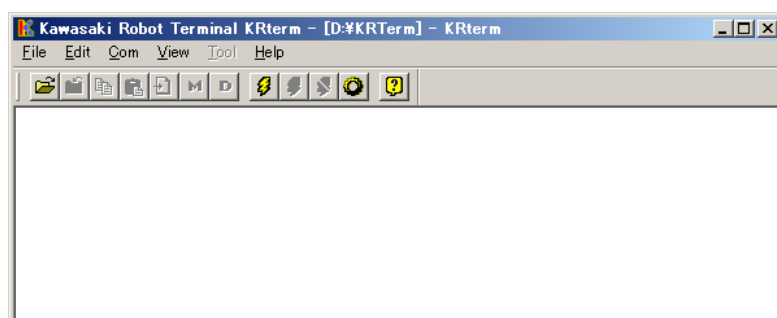
### 2.6.1.2 用以太网联接多台机器人

使用 KRterm 或 KCwin32/KCwinTCP 软件，可以通过以太网进行来自个人计算机控制器的操作。

#### 1. 连接电缆

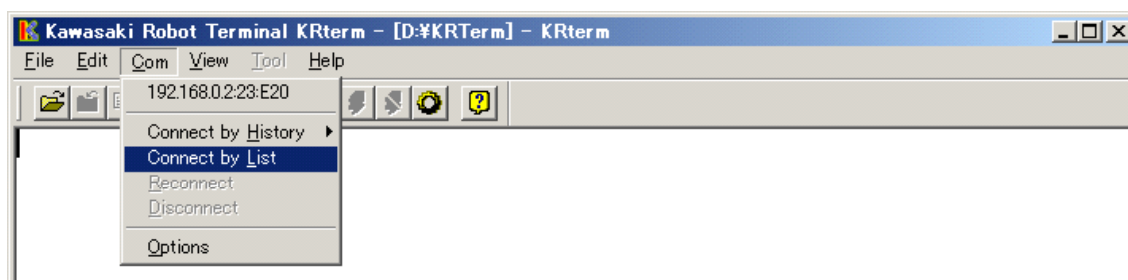


#### 2. 打开控制器电源并双击终端软件的图标。

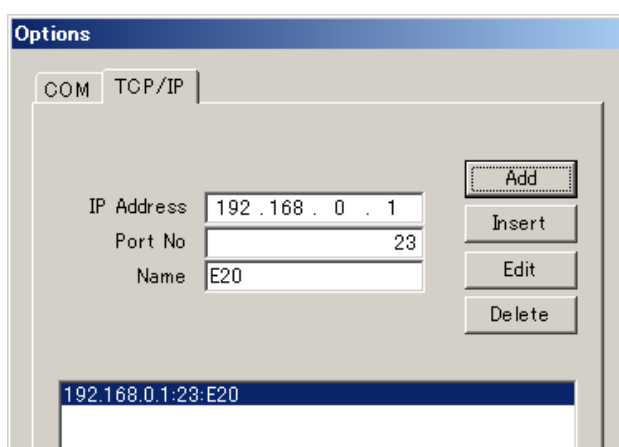


#### 3. 设置机器人的 IP 地址。

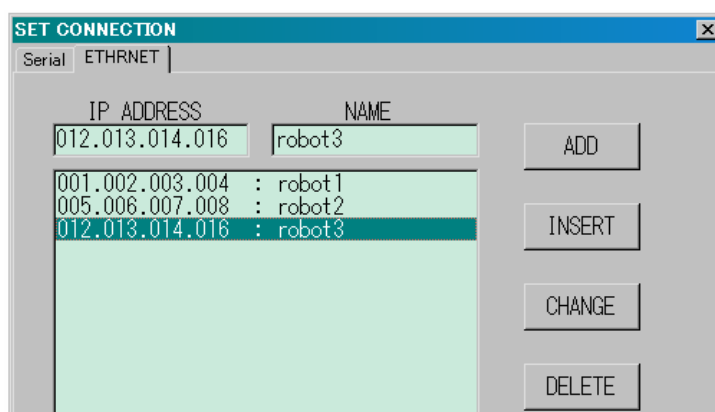
在菜单条上选择[Com(munication)] → [Options]。



对于 KRterm 软件，点击[TCP/IP] 并输入要联网的机器人控制器的 IP 地址和名称（可选的）。点击<Add>。

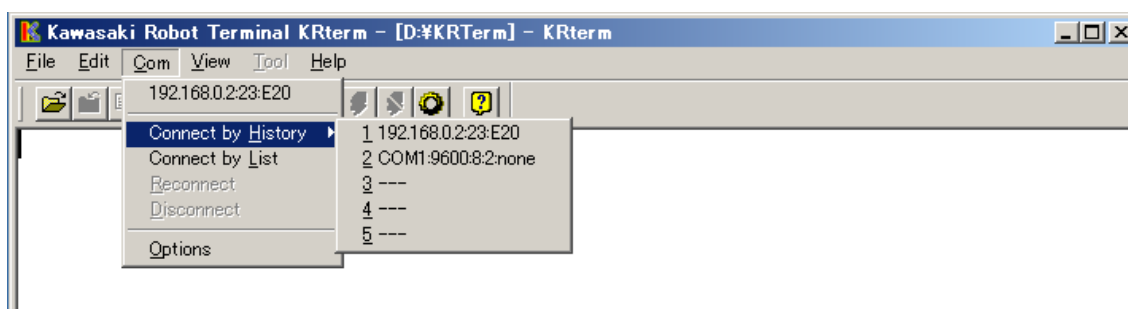


对于 KCwinTCP 软件，屏幕显示如下。输入要联网的机器人控制器的 IP 地址和名称（可选的），然后点击<Add>。

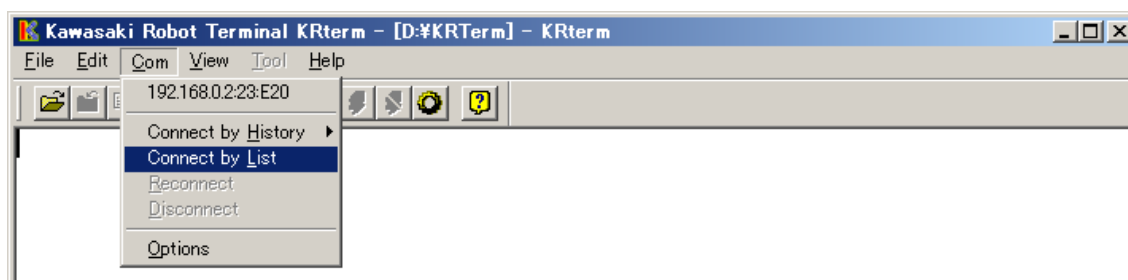


4. 按如下步骤联接机器人。

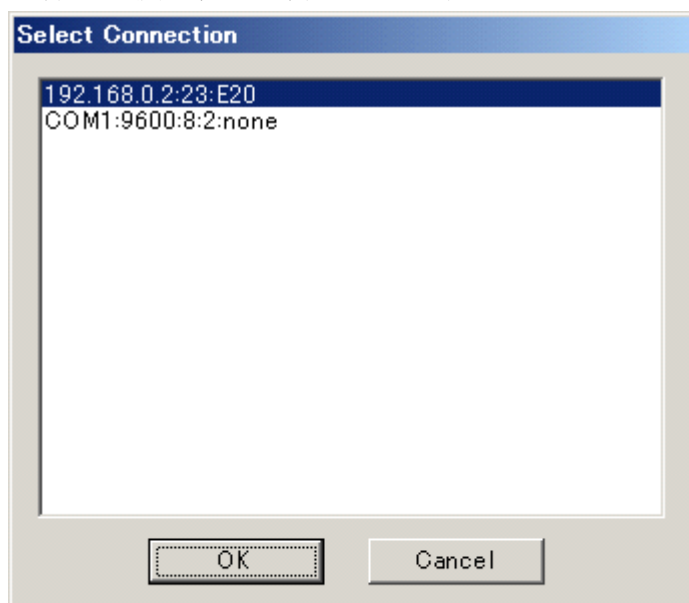
- (1) 当在菜单条中选择[Com]([通信])时，最近联接过的机器人就会显示在下拉式菜单中，可以在该下拉菜单中点击需联接的机器人名称。
- (2) 选择[Com] → [Connect by History]，显示出最近用过的机器人一览表。从此表中选择要联接的机器人。



- (3) 要联接以上的机器人一览表没有的机器人时，请选择 [Com(C)] → [Connect by List]。



选择要联接的机器人并点击<OK>按钮。





5. 如果联接建立，机器人的信息，如机器人的名称会显示在信息 login :后面。在该信息后输入“as”，将显示从机器人返回的提示符“>”。

提示符显示后，在其后输入任意的 AS 监控指令。

## 2.6.2 上传和下载数据

### (1) SAVE 指令

用 SAVE 指令将数据保存到计算机中(参看 5.3, SAVE 指令)。

实例 >SAVE test.pg

本操作以 KRterm

(KCwin32/KCwinTCP) 的安装目录为  
路径保存数据。

>SAVE My Documents\test.pg

把数据保存到指定文件夹。

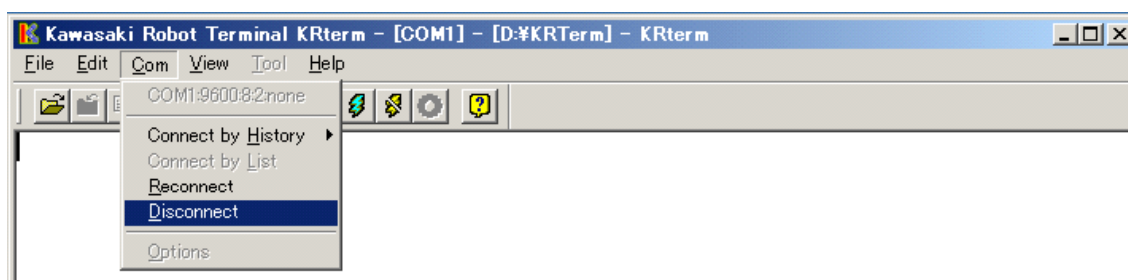
### (2) LOAD 指令

用 LOAD 指令，可将数据从计算机装载到机器人存储器中。

实例 >LOAD data01.as

## 2.6.3 系统关闭

1. 在机器人被接入时，选择菜单条中的 [Com ()] → [Disconnect]  
([通信] → [断开])，以断开机器人的联接。

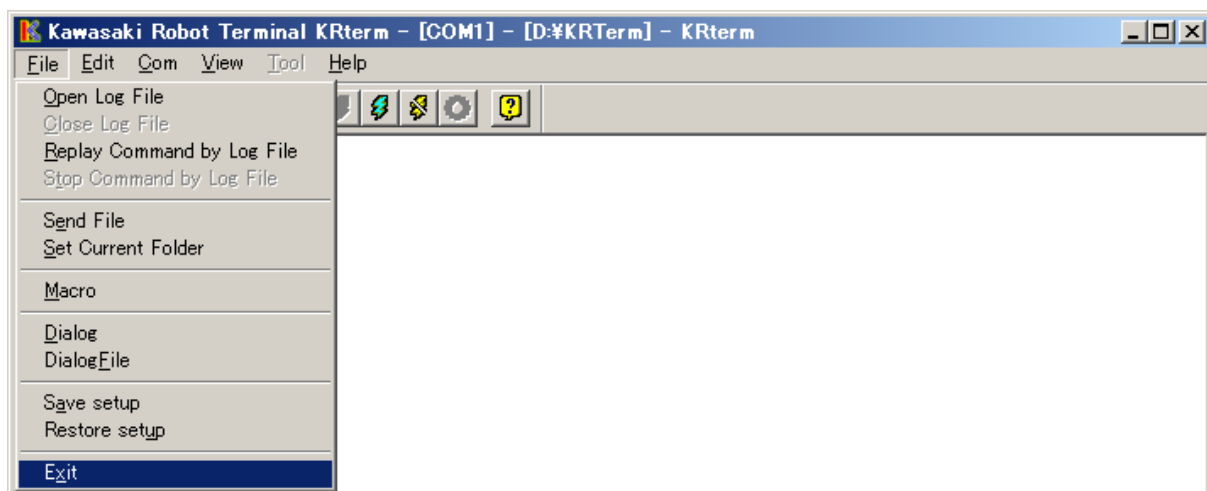


2. 关闭机器人控制器电源。(参看“操作手册” 3.2 电源关的步骤)

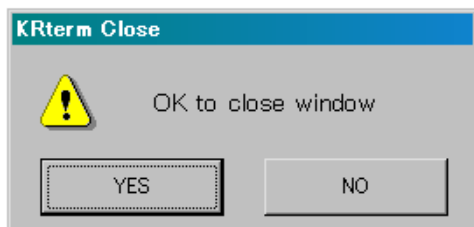
- (1) 将 HOLD/RUN 状态由 RUN (运行) 改变为 HOLD (保持)。
- (2) 按下紧急停止按钮，关断马达电源。
- (3) 关断控制器电源开关。

3. 关闭 KRterm 软件

- (1) 选择菜单条 [File] → [End] ([文件] → [结束])。



(2) 点击 <YES>(<是>)。



4. 关闭计算机电源。
5. 如果没有必要保持 PC 与控制器的连接，在确保控制器和 PC 电源关闭后，断开电缆连接。

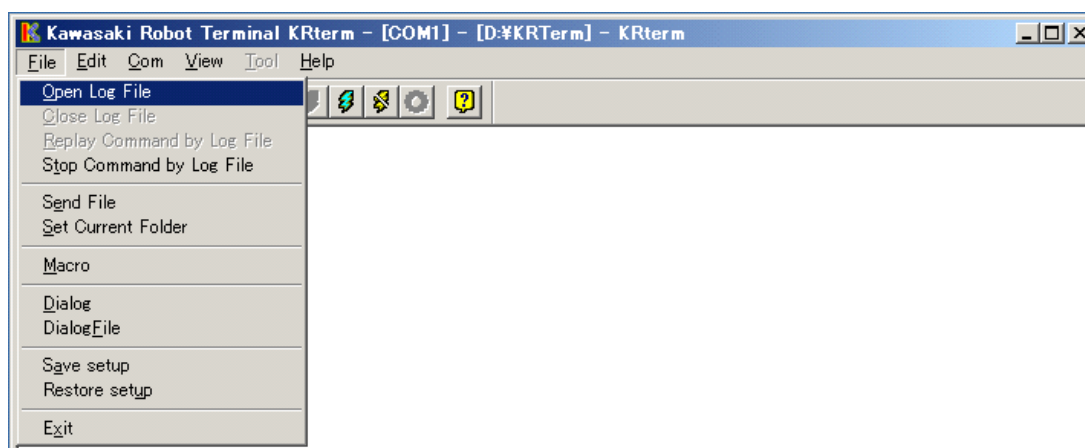
## 2.6.4 KRTERM 软件的功能

### 2.6.4.1 创建日志文件

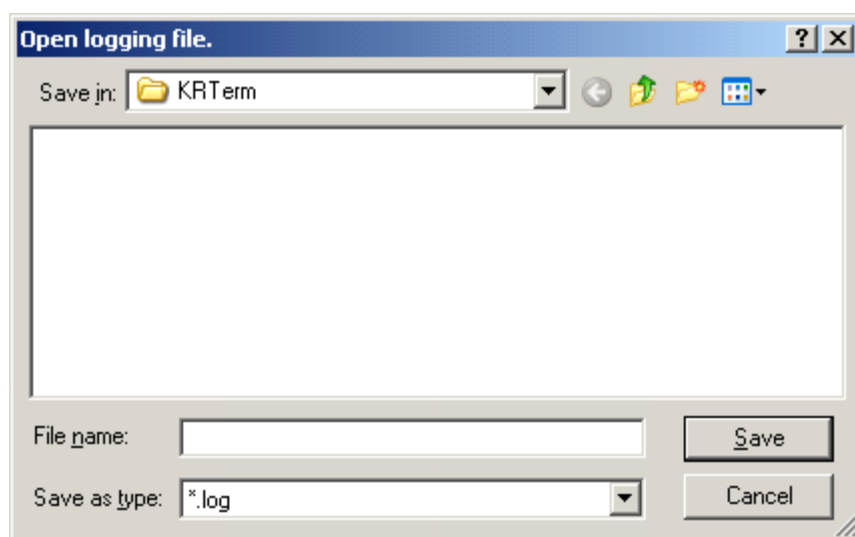
显示在 KRterm 画面中的信息可以保存到一个日志文件中。这有助于把机器人操作步骤打印出来。

#### 1. 启动日志

- (1) 选择菜单条 [File] → [Open Log File] ([文件] → [打开日志文件])。



(2) 选择要保存日志文件的文件夹，并给出文件名。



(3) 标题条上显示[Logging Now]([记录中])信息。显示器上的内容被记录下来直到日志文件被关闭。

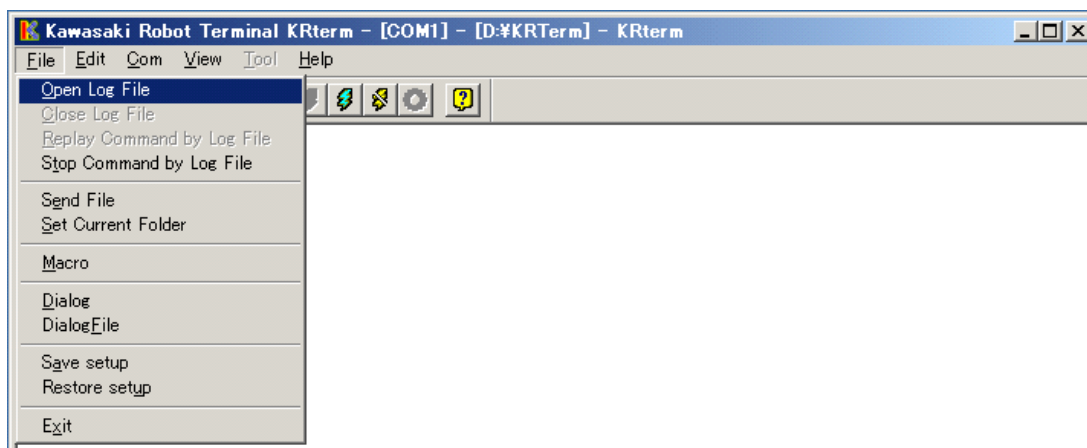


该信息显示时，显示器上的内容被记录下来。

## 2. 结束记录

一旦日志记录启动，KRterm 显示的所有内容都将被记录，直至日志文件关闭。

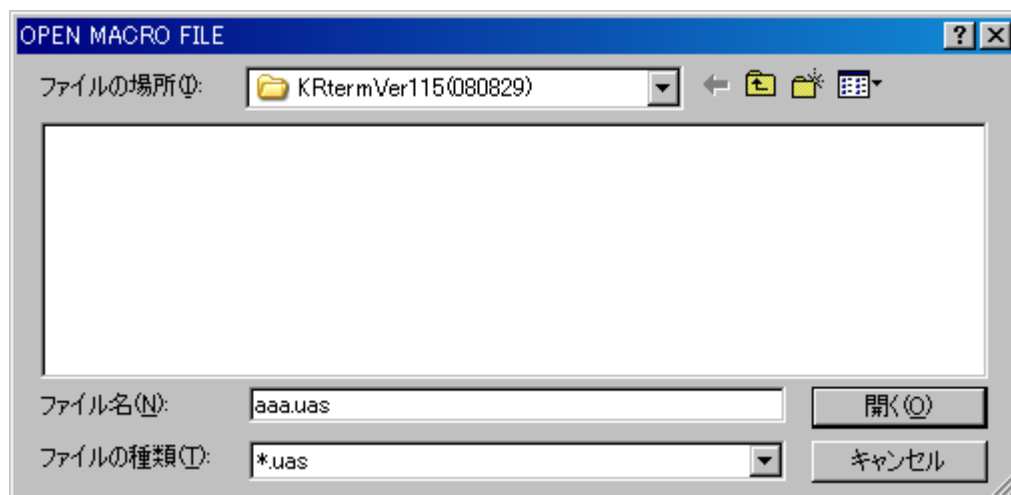
要关闭日志文件，结束日志记录，请选择菜单条[File] → [Stop Command by Log File]。



#### 2.6.4.2 宏功能

KRterm 和 KCwin32/KCwinTCP 系统提供有宏的功能。如果一个任务需要重复执行，把执行该任务所需的一系列指令或命令记录在宏中，这是非常有用的，可以提高效率。

要录制一个宏，请选择菜单条[FILE(F)] → [MACRO(M)]([文件(F)] → [宏(M)])，输入文件名，保存该宏。要运行一个宏，请在 KRterm 画面中使用 SEND 指令。



更多详情请参阅 KRterm 或 KCwin32/KCwinTCP 中的帮助。

### 3.0 AS 语言的信息表达


本章介绍了 AS 语言中信息和变量的类型。


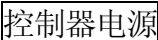
#### 3.1 符号和规则

##### 1. 大写字母和小写字母

为了便于理解，本手册中使用下述大写和小写字母应用规则：所有 AS 关键字(指令，命令等)用大写字母显示；可以被指定的变量和其他项目用小写字母显示。但是，在 AS 终端输入时大小写字母均可使用。

##### 2. 键和开关

示教器或计算机键盘上的键和控制器上的开关，在手册中，用他们的名字加一个方框  来表示。

例如 、

##### 3. 缩写

关键词能被缩写。例如，EXECUTE 命令可以被缩写为 EX。见附录 5 AS 语言列表




##### 4. 空格，Tab(制表格)键

命令之间(或指令)和参数\*之间至少有一个空格或制表格，以示分隔。同时，在多个没有用逗号或其他分隔符分开的参数之间，也需要用空格或制表格来分隔。多余的空格和制表格将被系统忽略。

**注释\*** 参数是一个完整命令或功能(函数)所必需的数据。例如，SPEED 命令中，需要一个参数数据用于指定机器人的速度。如果指令或功能(函数)需要使用几个参数时，请用逗号或空格将这些参数进行分隔。

例如 SPEED 50

##### 5. ENTER 键

监控指令和程序命令在按下键后，才会被处理。本手册中，键将用表示。

##### 6. 省略的参数

许多监控指令和程序命令有可以被省略的参数。如果在这些可选择的参数后面有一个逗号，即使在参数被省略时，逗号也必须保留。但如果所有的参数都被省略，这时逗号也可以省略。

## 7. 数值

如果没有另外指明的话, 数值是用十进制数表达的。数学表达式使用这些值作为 AS 监控指令和程序命令的自变量。但是, 请注意可接受的值是受限制的。下面的规则将说明各种情况下是如何解释这些值的。

### (1) 距离

用来定义机器人在两个点之间移动的长度。距离的单位是毫米(mm); 输入时单位被省略。输入值可以是正值或负值。

### (2) 角度

在指定的位姿(位置)处定义和修正机器人的姿态, 描述机器人各关节的旋转值。此值可以为正、负值, 最大角度为 180 度或 360 度, 取决于所用的指令。

### (3) 数量变量

除非另外指明, 这些变量表示为实数。变量值的范围为 $-3.4\text{E}+38$  到  $3.4\text{E}+38$  ( $-3.4 \times 10^{38}$  到  $3.4 \times 10^{38}$ )。当它超过 $\pm 999999$  时, 表达为 $x\text{E}+y$  ( $x$  是尾数,  $y$  是指数)。

### (4) 关节编号

以 1 到可用关节数量(标准型有 6 个关节)的整形数, 表达机器人的关节。关节从基座关节开始, 按照顺序进行编号。(通常表达为 JT1, JT2 ....)

### (5) 信号编号

用于标识二进制(开/关)信号。此值以整形数给出并遵循以下的范围。

	标准范围	最大范围
外部输出信号	1—32	1—960
外部输入信号	1001—1032	1001—1960
内部信号	2001—2256	2001—2960

负的信号编号表示 OFF 状态。

## 8. 关键字

总的来说, 在 AS 系统中可以任意给变量命名。但是, AS 系统中定义了命令、指令等的关键字被保留, 不能用作位姿数据、变量等的名称。

## 3.2 位姿信息，数字信息和字符信息

在 AS 系统中有 3 种类型的信息：位姿\*信息、数字信息和字符信息

**注释\*** “位姿(Pose)” 以前称为“位置(Location)”，但为了与国际标准(ISO)一致，本手册中，用一个词“位姿”来表示机器人的位置和姿态。

### 3.2.1 位姿信息

位姿信息，用来指定给定工作区域中机器人的位置和姿态。如果没有其他说明的话，机器人的位置和定向指的是机器人工具中心点(Tool center point, 缩写为 TCP)的位置和工具(坐标)的定向。此位置和姿态被称为机器人的位姿(位置)。

位姿由机器人的位置和面向方向来决定，因此，机器人被命令移动时，这两个方面被同时执行。

1. 机器人的 TCP(工具中心点)移动到指定位置。
2. 机器人的工具坐标旋转到指定的姿态

位姿数据用一套关节位移值(各轴值)或变换值(transformation values)来描述：

#### 1. 关节位移值(各轴值)

位姿信息以来自每个机器人轴的一套角度或直线位移值给出。使用编码器值，角度位移和直线位移分别是由角度和毫米来计算并描述的。一旦决定了关节位移值，TCP(工具中心点)的位置和定向是唯一被指定好的。

**例如** 关节位移值由 JT1, ...JT6 按顺序表达，每个关节的位移值显示在关节编号之下。

	JT1	JT2	JT3	JT4	JT5	JT6
#pose =	0.00,	33.00,	-15.00,	0,	-40,	30

#### 2. 变换值 (X, Y, Z, O, A, T)

用与参考坐标系的关系描述坐标系的位姿。除非其他指定，它是指机器人的工具坐标系相对于基础坐标系的变换值。位置由基础坐标系的 TCP(工具中心点)的 XYZ 值给定，定向由基础坐标系的工具坐标的欧拉 OAT 角度\*给定。通常使用的转换值是：工具变换值，描述工具坐标系相对于空工具坐标系的姿态，基于工件的变换值，描述工具坐标系相对于工件坐标系的姿态。

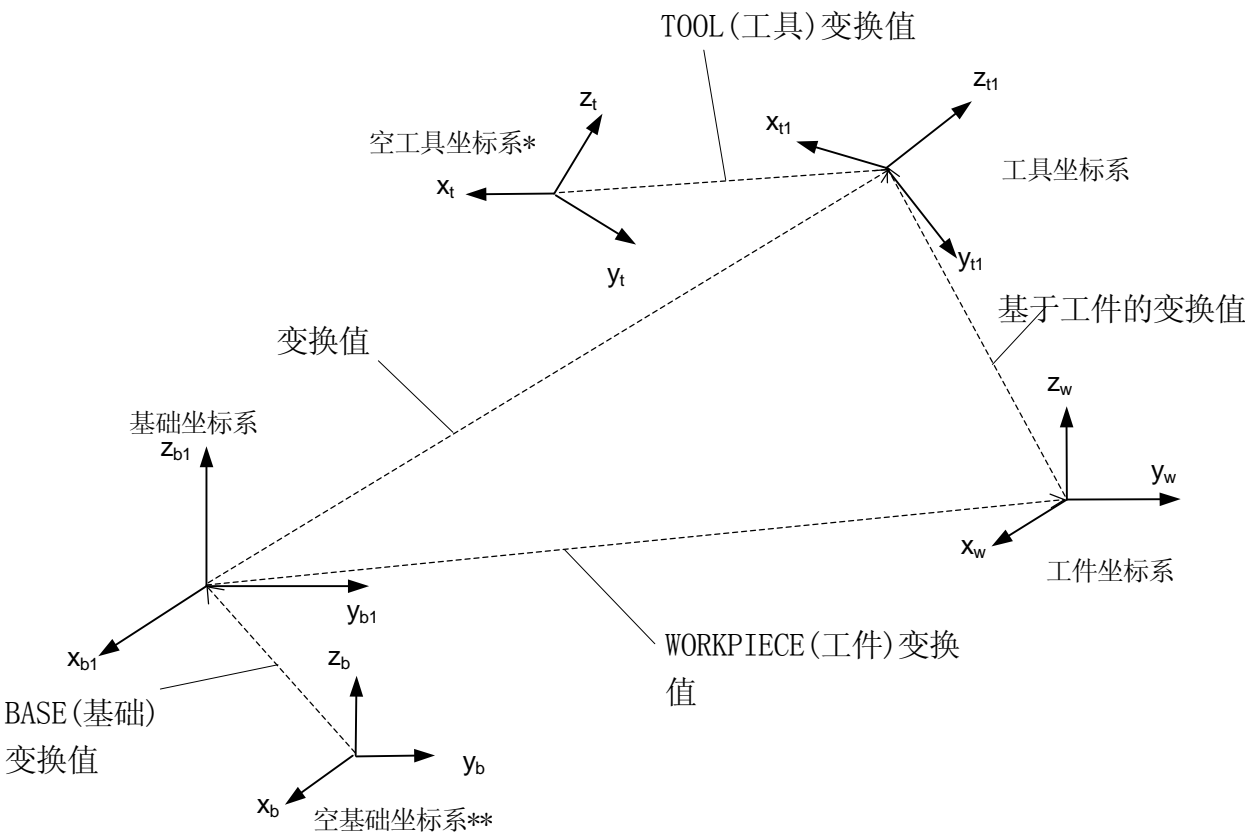
**注释\*** 见附录 3 欧拉 O, A, T 角度



例如	X	Y	Z	0	A	T
pose =	0,	1434,	300,	0,	0,	0

如果机器人有多于 6 个轴，额外轴的值以变换值显示。

例如	X	Y	Z	0	A	T	JT7
pose =	0,	1434,	300,	0,	0,	0	1000



**注释\***
空工具坐标系有他们自己的原点，此原点在机器人工具安装法兰表面的中心，它们用 T00L(工具)变换值(0, 0, 0, 0, 0, 0) 描述。

**注释\*\***
空基础坐标系被设置成机器人的默认值，用 BASE(基础)变换值(0, 0, 0, 0, 0, 0) 描述。

关节位移值和变换值各有利弊。请根据你的需要选用他们。

	关节位移值	变换值
优点	<ul style="list-style-type: none"> <li>· 再生精度高, 机器人位姿形态不模糊。</li> </ul>	<ul style="list-style-type: none"> <li>· 即使工具被改变, 用于再现模式的工具坐标系原点也不会改变。(空工具坐标系平移)</li> <li>· 可以使用相对坐标系(例如工件坐标系)</li> <li>· 数据用 XYZOAT 值显示, 可便于处理。</li> </ul>
缺点	<ul style="list-style-type: none"> <li>· 工具更换时 TCP 改变(空工具坐标系保持相同)</li> <li>· 不能使用相对坐标系(例如工件坐标系等)</li> </ul>	<ul style="list-style-type: none"> <li>· 如果 BASE (基础) 或 TOOL (工具) 变换值改变, 坐标系也将改变, 因此必须全面理解任何改变对安全使用的影响。</li> <li>· 如果在再现运转前没有进行设置, 机器人形态可能会改变。</li> </ul>
建议用法	<ul style="list-style-type: none"> <li>· 规定程序的开始位姿</li> <li>· 在用变换值描述位姿时或之前, 规定机器人的形态</li> <li>· 用于其他共同的位姿</li> </ul>	<ul style="list-style-type: none"> <li>· 描述相对坐标, 例如工件坐标系</li> <li>· 描述需要使用数字值和功能(函数)例如 SHIFT 进行改变的位姿</li> <li>· 描述需要通过传感器信息进行改变的位姿</li> </ul>

— [ 注 意 ] —

1. 如果使用关节位移值定义的位姿，机器人的形态将是唯一确定的。与关节位移值定义的位姿不同，如果位姿用变换值定义，机器人对应于那个位姿可能有不同的形态。这是因为变换值仅设置了机器人工具坐标系的 XYZOAT 值，并没有定义每个关节的轴数据。因此，在再现模式启动机器人前，务必使用形态指令 (LEFTY 等) 或记录关节位移值来固定机器人的形态。
2. 因为变换值是通过 BASE (基础) 坐标系来描述的，如果基础坐标系用 BASE 指令/命令进行了平移，机器人的 TCP 将会被平移相同的量。这是用变换值的优点之一，但是必须注意，基础坐标系的改变将会影响变换值点的位置。如果没有注意到这个影响，将会导致事故，例如与外围设备发生干涉等。

在使用 TOOL 指令/命令时要注意同样的问题。

### 3.2.2 数字信息

在 AS 系统中，数字值和表达式可以用作数字信息。数字表达式是用数字和变量同运算符和函数相结合的一个值。数字表达式不仅仅用于数学计算，也可作为参数用于监控指令和程序命令。

例如 DRIVE 指令中，需要指定 3 个参数：关节编号、运动量和速度。参数不但可用数值，也可以使用表达式来表达。如下例中所示：

DRIVE 3, 45, 75                      关节编号 3 以 75% 的速度移动 45° 。

DRIVE joint, (start+30)/2, 75      指定 joint=2, start=30 后，关节 2 以 75 % 的速度移动+30° 。

用于 AS 系统的数字值，可以分为三种类型：

#### 1. 实数

实数含有整数和小数。它可以是在  $-3.4 \times 10^{38}$  和  $3.4 \times 10^{38}$  ( $-3.4 \times 10^{38}$  和  $3.4 \times 10^{38}$ ) 之间的整数、负数或 0。实数可以用科学计数法来表示。用符号 E 将尾数和指数两部分分开。指数可以是负数 (1/10 的乘方) 或正数 (10 的乘方)。

例如	8. 5E3	$8.5 \times 10^3$	(表达式中的+被省略)
	6. 64	$6.64 \times 10^0$	(E、0 被省略)
	-9E-5	$-9.0 \times 10^{-5}$	(小数点被省略)
	-377	$-377 \times 10^0$	(小数点、E、0 被省略)

注意前 7 个数字有效，但是通过计算过程后有效数字的数量可能会减少。

没有小数部分的实数叫做整数(整个的数)。范围为-16, 777, 216 到+16, 777, 215，对于那些超过此限制的，前 7 位数字有效。整数通常用十进制数输入，虽然有时用二进制或十六进制表达更加方便。^B 表明数据用二进制方式输入。^H 表明数据用十六进制方式输入。

例如	^B101	(十进制为 5)
	^HC1	(十进制为 193)
	-^B1000	(十进制为-8)
	-^H1000	(十进制为-4096)

2. 逻辑值

逻辑值只有两种状态：ON(开)和 OFF(关)，或 TRUE(真)和 FALSE(假)。值-1.0 被赋值给 TRUE 或 ON 状态，值 0(或 0.0)被赋值给 FALSE 或 OFF 状态。ON、OFF、TRUE 和 FALSE 作为 AS 语言被保留。

逻辑真=TRUE, ON, -1.0  
逻辑假=FALSE, OFF, 0.0

3. ASCII 值

显示一个 ASCII 字符的数字值。字符用前缀(‘)来区分于其他值。

‘ A ‘ 1 ‘ v ‘ %

### 3.2.3 字符信息

AS 系统中的字符信息以用引号(“”)括起来的一串 ASCII 字符来表示。用于引号表示字符串的开始和结束，它不能用作字符串的一部分。ASCII 控制字符(CTRL, CR, LF 等)也不能包含在字符串中。

例如

>PRINT	“KAWASAKI”
↗	↖
指令	字符串

### 3.3 变量

在 AS 系统中，位姿信息、数字信息和字符信息都可以赋予名称。这些名称叫做变量，变量可以分为两种类型：全局变量和局部变量。除非另有声明，这里的变量指的是全局变量。

#### 3.3.1 变量(全局变量)

用做位姿信息、数字信息、字符信息的变量分别叫做位姿变量、实型变量\*、字符串变量。几个变量可以成组并用数组变量将它们指定在一个名称下。

**注释\*** 因为 AS 中使用的大多数数值是实数，所以数字变量指的是实数变量或实型变量。但是请注意，整数、逻辑值和 ASCII 值都用实数值来表达。因此，一个实型变量可能指这些值中的任意一个。

一旦变量被定义，它将其数值保存在存储器中。因此，它可以在任何程序中使用。

#### 3.3.2 局部变量

与上述全局变量相比，局部变量在每次执行程序时都被重新定义，并且不保存在存储器中。局部变量以名字前带有一个“.”(点)的变量来表示。

局部变量在几个程序使用相同的变量名、并且这些变量在每次程序运行时改变的情况下非常有用。局部变量也能用作为子程序的参数(见 4.4.2 带参数的子程序)。

[ 注 意 ]

1. 局部变量不能用监控指令定义。
2. 因为局部变量不保存在存储器中，局部变量.pose 的值不能用下面的命令来显示。

>POINT .pose

要查看局部变量的当前值，可以在局部变量定义所在的程序中，将它的值赋给一个全局变量，再用 POINT 指令即可。

POINT a=.pose

在使用 POINT 指令前，执行定义局部(本地)变量的程序。

>POINT a

X[mm]	Y[mm]	Z[mm]	O[deg]	A[deg]	T[deg]
-------	-------	-------	--------	--------	--------

xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx	xxxxxxx
---------	---------	---------	---------	---------	---------

变化吗? (放弃请按 RETURN 键)

### 3.4 变量名

变量名字必须以字母开头，并且只能包括字母、数字、点和下划线。字符可以以大写或小写字母(它将在屏幕上以小写字母显示)来输入。变量名字的长度限于 15 个字符之内。更长的名字仅前 15 个字符有效。下面是一些不可使用名字的例子：

3p.....第一个字符不是字母

part#2.....“#”是关节位移变量名的前缀，不能用在变量名的中间

random.....关键字

#### [ 注 意 ]

1. 描述关节位移值的变量前面带有符号“#”，以区别于变换值。字符串变量前面带有符号“\$”，以区别于实型变量。

pick (变换值变量)	#pick (关节位移值变量)
count (实型变量)	\$count (字符串变量)

2. 所有变量可以用作为数组变量。数组由相同名字下的、并且彼此之间用下标值区分的几个值组成。数组中的每一个值叫做数组元素。要指定数组元素，可在括号中附上元素下标值。例如，“part [7]”表示数组“part”的第 7 个元素。下标可以使用 0 到 9999 范围内的整数。三维数组使用类似的语法：part [7, 1, 1]=1。

3. 当一个变量被定义时，该变量可以在各种程序中使用。因此，小心避免对在不同程序中使用的变量做不必要的改变。



### 3.5 定义位姿变量

那些描述位姿信息的变量被称为位姿变量。位姿变量仅在有赋值时才被定义。只有当位姿变量在被创建了名字并被赋予数值时，它才被定义。如果执行了含未定义变量的程序，会报出错。

位姿变量在以下的情况中是很有用的。

1. 相同的位姿数据被重复利用而没有必要每次示教其位姿。
2. 一个已定义的位姿变量可在不同的程序中使用。
3. 一个已定义位姿变量可以被用来或被改变来定义一个不同的位姿。
4. 数值可作为位姿信息被直接输入，以取代使用示教器对机器人进行耗时的示教工作。
5. 位姿变量可以自由命名，使程序变得更易读。

位姿变量可按如下定义：

#### 3.5.1 用监控指令定义

1. HERE (意思为“这里”)指令把机器人的当前位姿数据存储在指定名称下。

##### 例 1 使用关节位移值

变量名以#开头，以区别于变换值。紧跟在指令后面，显示当前位姿的关节位移值：

```
> HERE #pose 
JT1      JT2      JT3      JT4      JT5      JT6
xxxxxxx  xxxxxxx  xxxxxxx  xxxxxxx  xxxxxxx  xxxxxxx
变化吗？(放弃请按 RETURN 键) 
>
```

##### 例 2 使用变换值

紧跟在指令后面，显示当前位姿的变换值：

```
> HERE pose 
X[mm]    Y[mm]    Z[mm]    O[deg]    A[deg]    T[deg]
xxxxxxx  xxxxxxx  xxxxxxx  xxxxxxx  xxxxxxx  xxxxxxx
变化吗？(放弃请按 RETURN 键) 
>
```

2. POINT(意思为“点”)指令是用另一个已定义的位姿变量来定义一个位姿，或通过终端输入数值来定义它。

### 例 1 使用关节位移值

- (1) 定义一个新的、未定义过的变量

```
>POINT #pose 
      JT1      JT2      JT3      JT4      JT5      JT6
      0.000    0.000    0.000    0.000    0.000    0.000
变化吗？(放弃请按 RETURN 键) 
>
输入新值，各值之间以逗号为分隔：
xxx, xxx, xxx, xxx, xxx, xxx
```

- (2) 改变一个已定义变量的值

```
>POINT #pose 
      JT1      JT2      JT3      JT4      JT5      JT6
      10.000   20.000   30.000   40.000   50.000   40.000
变化吗？(放弃请按 RETURN 键) 
输入要改变的值：
30, , , , 20, ;将 JT1 和 JT5 的值改变为 30 和 20
```

- (3) 替换一个已定义变量的值

```
>POINT pose_1=pose_2 
      JT1      JT2      JT3      JT4      JT5      JT6
      10.000   20.000   30.000   40.000   50.000   40.000
变化吗？(放弃请按 RETURN 键) 
```

显示要定义给 pose\_1 的值(Pose\_2 的最新值)。按  键，将这个值设置给 pose\_1，或用与上述(2)相同的步骤修改他们。

### 例 2 使用变换值

按照上述相同的步骤，只是变量名不可以#开头。

### 3.5.2 用程序命令定义

1. HERE 指令将机器人的当前位姿数据存储到指定名称下。

HERE pose

2. POINT 指令可用一个已定义的位姿来替代一个位姿变量。

POINT pose\_1=pose\_2

变量“pose\_1”被已定义的变量“pose\_2”的值代替。如果“pose\_2”未定义则报出错。

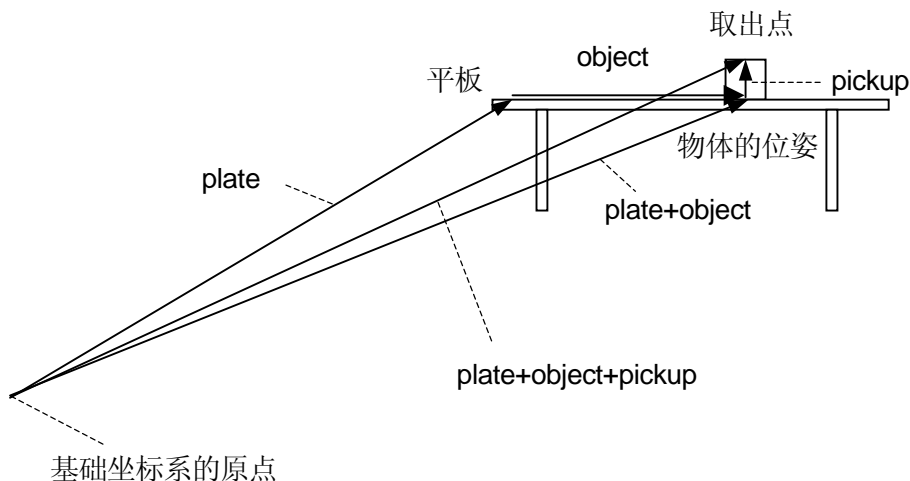
如果是关节位移值变量, 注意量名以#开始。  
如果是变换值变量, 定义变量其变量名没有前缀#。

### 3.5.3 使用复合变换值

在两个坐标系之间的变换值, 可以用在两个或多个过渡坐标系之间的变换值的组合来表达。这被叫做复合变换值或相对变换值。

比如说“plate”是相对于基础坐标系的变换值的变量名, 描述了平板上的坐标系。然后, 如果有个相对于位置“plate”的物体的位姿被定义为“object”, 那么该物体 object 相对于机器人基础坐标系的复合变换值, 可以用“plate+object”来描述。

在下例中，即使位姿“plate”改变(例如平板移动)，只有“plate”的变换值需要更新，其他的可以像原来一样使用。



复合变换值可以用定义位姿变量的任何命令和指令来定义。(用 HERE 最容易)

首先，用示教器将机器人工具点动到要命名为“plate”的位姿。然后输入以下指令将此位姿定义为 plate。

```
>HERE plate 
```

接下来，将机器人工具点动到要命名为“object”的位姿，并输入：

```
>HERE plate+object 
```

变换值“object”现在定义为相对于“plate”的当前位姿\*(如果“plate”在此时仍未定义，“object”将不能被定义并且报出错)。

**注释\*** 输入 HERE 命令后屏幕上出现的是最右边变量(如本例中为“object”)姿态的变换值。这不是“plate + value”值。要查看“plate + object”值，请当机器人在那个位姿时，使用 WHERE 指令来查看。

最后，将机械手移动到捡起取出 object (物体)的位姿，并输入：

```
>HERE plate+object+pickup 
```

最后的那个指令定义相对于变换值“object”的位姿变量“pickup”。

如上所示，复合变换值通过几个独立变换值的“+”结合来定义。请在“+”和变换值之间不要留任何空格。用这种方法，你可以结合任意多的变换值。

如果机器人要在指定为相对于“object”的“pickup”位姿处捡起 object (物体)，程序应该这样写：

JMOVE plate+object+pickup  
或 LMOVE plate+object+pickup

[ 注 意 ]

1. 不要改变相对变换值表达式中的变换值变量的顺序。例如，如果变换值变量“b”被定义为相对于变换值变量“a”，表达式应该为“a+b”，而不是“b+a”。
2. 上例中的位姿数据“object”和“pickup”被定义为了相对于其他位姿数据。因此，不要使用“JMOVE object”或“LMOVE pickup”指令，除非你确信它在程序中的目的和影响。

在重复使用复合变换值时，请使用 POINT 命令来减少计算复合变换值的时间。例如，要接近位姿“pickup”、然后向“pickup”位姿移动，你可能输入：

JAPPRO	plate+object+pickup, 100	接近“pickup”上方 100 mm
LMOVE	plate+object+pickup	直线运动到“pickup”

然而，如果你输入以下指令，将可以节省计算时间：

POINT	x=plate+object+pickup	计算目标位姿
JAPPRO	x, 100	接近目标上方 100 mm
LMOVE	x	直线运动到目标

这两个程序的结果动作相同，但是后者计算一次混合转换，因此执行时间更短。在这个非常简单例子中，执行时间的差异是很小的，但是在更复杂的程序中，它可产生很大的差异，并且可以缩短整个循环时间。

[ 注 意 ]

对于带有 7 个关节的机器人，请注意下述问题：

1. 在使用 POINT 命令时，应注意 JT7 的值。例如，在

**POINT p=p1+p2**

“p”的 JT7 值将会是 JT7 在“p2”处的值。表达式右侧最右边的变量值将被赋予“=”左侧的变量。

2. 当要赋给 JT7 一个特殊值时，请在 POINT 命令的末尾增加“/7”，例如

**POINT/7 p=TRANS(,,,,,value)**

值“value”被赋给变量“p”的 JT7 值。

### 3.6 定义实型变量

实型变量用赋值命令(=)来定义。给实型变量赋值的格式是：

`Real_variable_name=numeric_value`

例如            `a=10.5`  
                 `count=i*2+8`  
                 `Z[2]=Z[1]+5.2`

左侧的变量可以是数量变量(如计数)，也可以是数组元素(如 Z[2])。变量仅在赋值时定义。在赋值前，它一直处于未定义状态，如果执行了含有未定义变量的程序，将报出错。

右侧的数值可以是一个常数、一个变量或者是一个数学表达式。在处理赋值命令时，先计算赋值命令的右侧值，然后把这个值赋给左侧的变量。

如果命令左侧的变量是一个新变量或以前从未被赋值过，右边的值将被自动赋值给该变量。如果左侧变量是个已定义的变量，这个新值将替换此变量的当前值。

例如，指令“x=3”赋值 3 给变量“x”。它读作“赋值 3 给 x”，而不是“x 等于 3”。下面的例子将清晰地解释其处理过程：

`x=x+1.`

如果此例是一个数学等式，读作“x 等于 x 加 1”，但它没有意义。作为赋值命令，它应该读作“赋值 x 加 1 给 x”。这样，先计算“x”的当前值和 1 的和，然后把结果值作为一个新的数值赋值给“x”。因此，这个等式要求首先定义 x，如下所示：

`x=3`  
`x=x+1`

这时，“x”的结果值为 4。

### 3.7 定义字符串变量

字符串变量用赋值命令(=)定义。给字符串变量赋值的格式是：

**\$string\_variable=string\_value**

例如

**\$a1=\$a2**

**\$error mess[2]="time over"**

左边的字符变量可以是一个变量(如\$name)，也可以是数组元素(如\$line[2])。只有当变量在被创建了名字并被赋予数值时，它才被定义。在赋值前，它一直处于未定义状态，如果执行了含有未定义变量的程序，将报出错。

右边的字符串可以是字符串常数、字符串变量或者是字符表达式。在处理赋值命令时，先计算赋值命令右侧的值，然后把这个值赋给左侧的变量。

**\$name="KAWASAKI HEAVY INDUSTRIES LTD."**

在上述命令中，把“ ”括起来的字符串赋值给变量“\$name”。如果指令左侧的变量以前未被用过，此字符串将被自动赋值。如果左侧的变量是个已定义的变量，这个命令将用右侧的新字符串替换当前的字符串。



### 3.8 数学表达式

数学表达式由数字、变量、指定函数(功能)或其他和运算符结合的数学表达式组成。所有的数学表达式由系统求值，得到实数值结果。数学表达式可替代数值用在任何地方。它们可以用作监控指令和程序命令的参数，也可以用作数组的下标。

值的解释取决于表达式出现的上下文。例如，一个数组下标的表达式将被缩位解释为一个整数值。一个逻辑值表达式，当其值为 0 时，被解释为逻辑假(FALSE)，而当值为非零值时，被解释为逻辑真(TRUE)。

#### 3.8.1 运算符

为描述表达式提供有算术、逻辑、和二进制等运算符。所有这些运算符将两个值结合，以获得一个结果值。例外：有两个运算符(NOT 和 COM)只操作一个值，运算符(-)操作一个或两个值。运算符的描述如下：

算术运算符	+ - * / ^ MOD	加 减或负 乘 除 乘方 取余
关系运算符	< <=, =< = = <> >=, => >	小于 小于或等于 等于 不等于 大于或等于 大于
逻辑运算符	AND NOT OR XOR	逻辑与 逻辑非 逻辑或 逻辑异或
二进制运算符	BAND BOR BXOR COM	二进制与 二进制或 二进制异或 取反

[ 注 意 ]

1. 关系运算符 “= ” 是检查两个值是否相等的运算符，不同于赋值指示 “= ”。
2. 二进制运算符 BOR 对于两个数的各个二进制位执行 OR 操作(本例中，数值以二进制格式表达，但是此运算适用于任何格式的数据)。

$\text{^B101000 BOR ^B100001} \rightarrow \text{^B101001}$

这个结果和 OR 操作的结果不同。

$\text{^B101000 OR ^B100001} \rightarrow -1(\text{TRUE})$

这里， $\text{^B101000}$  和  $\text{^B100001}$  被解释成逻辑值，因为两者都不是 0(逻辑假, FALSE)，所以表达式的计算结果为逻辑真(TRUE)。

### 3.8.2 操作顺序

表达式按照优先顺序进行计算。优先权如下所示，从 1 到 14。注意操作的顺序可以用括号把各要素成组控制。在含有括号的表达式中，最里一对面括号中的表达式被最先计算，然后系统向最外面一对括号的方向工作。

1. 计算函数和数组
2. 处理涉及字符串的关系运算符(见 3.9 字符串表达式)
3. 处理乘方运算符 “^”
4. 处理一元运算符 “-” (负)、NOT(非)、COM(取反)
5. 从左到右地处理乘法 “\*” 和除法 “/”
6. 从左到右地计算取余数操作(MOD 操作)
7. 从左到右地处理加法“+”和减法“-”
8. 从左到右地处理关系运算符
9. 从左到右地处理 BAND 运算符
10. 从左到右地处理 BOR 运算符
11. 从左到右地处理 BXOR 运算符
12. 从左到右地处理 AND 运算符
13. 从左到右地处理 OR 运算符
14. 从左到右地处理 XOR 运算符

### 3.8.3 逻辑表达式

逻辑表达式的结果为真(TRUE)或假(FALSE)。逻辑表达式可在程序中使用,用来作为决定下一步操作的条件。在下面的例子中,一个简单的逻辑表达式“ $x > y$ ”,被用在子程序中,决定两个变量中的哪一个赋值给变量“max”。

```
IF x>y GOTO 10
max=y
GOTO 20
10 max=x
20 RETURN
```

在计算逻辑表达式时,值 0 被认为是假(FALSE),所有非 0 值都被认为是真(TRUE)。因此,所有实数和实数表达式都可以用作逻辑值。

例如,下面两个语句的意思相同,但是,第二个更便于理解。

```
IF x GOTO 10
IF x<>0 GOTO 10
```

3.9 字符串表达式

字符串表达式由字符串、字符变量、指定函数或其他用运算符结合在一起的字符串表达式组成。下述运算符用在字符串表达式中。

字符串运算符	+	结合
关系运算符	<	小于
	<=, =<	小于或等于
	==	等于
	<>	不等于
	>=, =>	大于或等于
	>	大于

用字符串运算符的结果将是字符串，用关系运算符的结果为实数值。

在使用字符串的关系运算符时，字符串从字符串的第一个字符开始进行字符比较。如果所有字符相同，这两个字符串被认为相等，但是即使由一个不同，含更大字符编码的字符串被认为是大的字符串。如果字符串中一个更短，短的那个被计算为小。在字符串关系运算中，空格和制表符被当作字符处理。

```
"AA"      <    "AB"  
"BASIC" == "BASIC"  
"PEN."   >    "PEN"  
"DESK"   <    "DESKS"
```

[ 注 意 ]

大写字母和小写字母在字符串表达式中被认作为不同的字符。



## 4.0 AS 程序

本章对 AS 程序进行说明。对程序的编程、执行和机器人运动进行说明。为了更好的理解，请在学习本章时实际操作系统或 PC-ROSET\*。

**注释\*** PC-ROSET 是一种个人计算机机器人模拟器，与 AS 系统兼容。

### 4.1 AS 程序的类型

程序是一连串的命令，告诉机器人怎样运动、输出信号、做计算等。程序名称由不超过 15 个的字符构成，必须以字母开头，只能包含字符、数字和点符号。只要存储器够用，你可以创建任意多的程序。程序通常用 AS 系统的编辑模式创建，但是你也可以使用一台装有 KRterm 或 KCwin32/KCwinTCP 终端软件或 PC-ROSET 的计算机来创建，并且可事后将程序装载到机器人存储器中。

#### 4.1.1 机器人控制程序

机器人控制程序是那些控制机器人动作的程序。你可以用包括机器人运动命令在内的所有程序命令来创建这些程序。

#### 4.1.2 PC 程序(过程控制程序)

PC 程序或称过程控制程序，是指与机器人控制程序同时执行的那些程序。PC 程序一般被用来控制和监控外部 I/O 信号。PC 程序和机器人控制程序可以通过共同使用的变量和内部信号来相互通信。

PC 程序和机器人控制程序使用相同的命令。因此，在一些情况下，一个 PC 程序可以被当作机器人控制程序来执行。但是，除了 BRAKE 命令之外的运动命令不能在 PC 程序中使用。BASE 和 TOOL 功能(函数)也不能用于 PC 程序。

### 4.1.3 自动启动

PC 程序可以被设置成在控制电源开启时自动启动。

1. 把系统开关 AUTOSTART.PC (或 AUTOSTART2.PC – AUTOSTART5.PC) 设置为 ON。
2. 创建你想要自动启动的程序，并将其命名为 AUTOSTART.PC (或 AUTOSTART2.PC – AUTOSTART5.PC)

有些监控指令也能通过使用程序命令 MC 在程序中执行。

例如 MC CONTINUE 等。(见 6.9 MC 程序命令)

这是一个示例 autostart (自动启动) 程序。在本例中，在控制器电源打开后，机器人监控马达电源，当其为 ON 时，执行程序 pg1。为了便于理解，这里忽略了安全检查，但是实际应用中，务必包括安全检验步骤。

```
autostart.pc( )  
1 WAIT SWITCH(POWER) ; 等待控制器电源 ON  
2 WAIT SIG(27) ; 检查机器人是否在原点位姿*  
3 MC EXECUTE pg1 ; 执行 pg1 (机器人动作程序)
```

**注释\*** 在执行此程序前，请设置原点位姿，并分配专用信号 HOME1 给信号 27。

## 4.2 创建和编辑程序

本节中，以一个简单的程序为例命令机器人执行一个任务。一个程序是一个叫机器人如何工作的进程表。当通过 AS 系统执行一个程序时，程序的步骤(行)是按照由上到下的顺序处理的，并且各步骤中定义的操作由机器人来实现。

### 4.2.1 AS 程序格式

AS 语言程序的各行(步骤)用如下格式表达。

步骤编号	标签	程序命令	;说明
------	----	------	-----

1. 步骤编号
- 步骤编号是自动分配给程序的各行的。步骤编号是从 1 开始连续编码的，无论何时插入或删除一些行，都会自动重新编号。
2. 标签
- 程序中的标签用来分支程序。标签可以是 1-9999 的整数，也可以是长度不超过 15 个的字母数字字符、点符号或下划线(以字母开头)，后面跟着冒号(:)。标签被插入在程序行的开头部分，位于右侧步骤编号之后。标签可被用作为程序中任何地方的分支(跳转)的目的地。
3. 说明
- 用分号(;)来表示分号右边的信息是说明文(注释文)。在程序执行时，不会对注释像程序命令那样处理。你也可以使的一个程序行只有一个分号而没有任何标签或命令。空白行也可以用来提高程序的可读性。(空白行至少要包含分号和分号后面一个空格或制表符)。



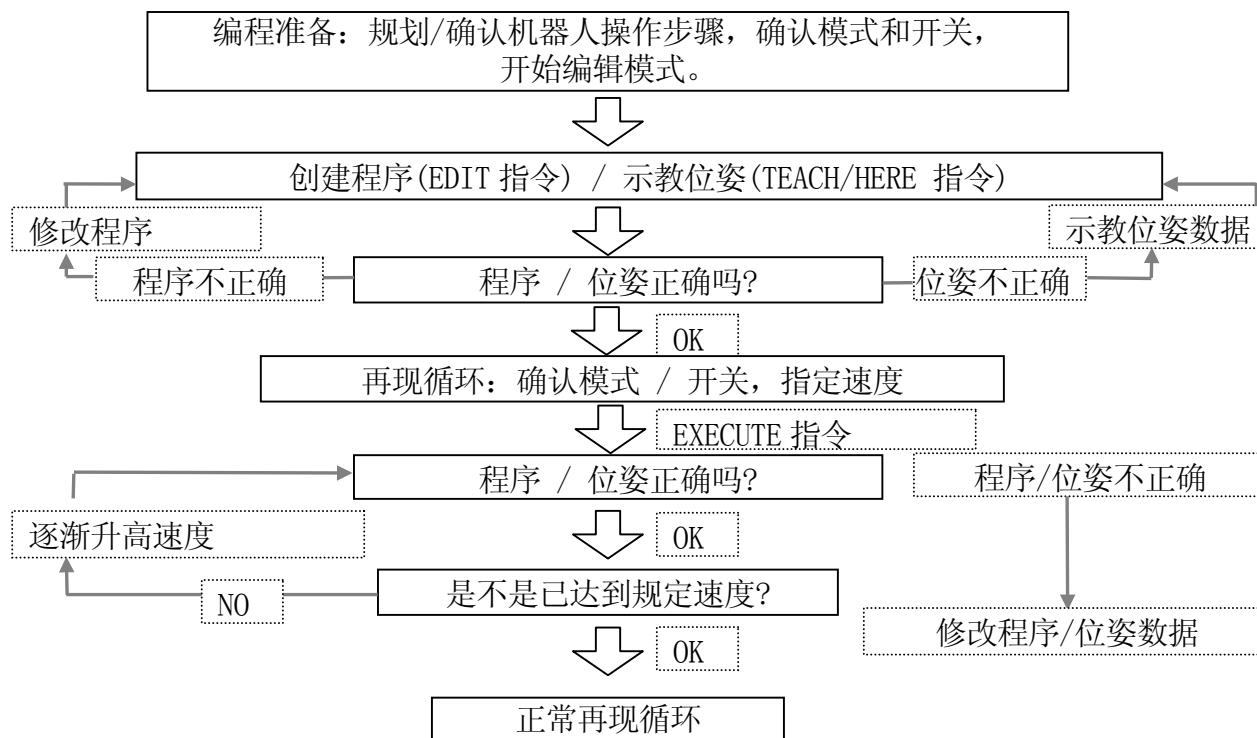
### 4.2.2 编辑指令

下面的编辑指令可用来创建或编辑程序。(高亮的参数可以省略)

EDIT 程序名, 步骤	启动编辑模式。
程序命令	用一个新的命令替代当前步骤。
登录键(  )	不更改当前步骤, 进入下一步。
D 步骤数量	删除指定数量的程序步骤。(D:Delete, 删除)
E	退出编辑模式, 返回监控模式。(E:Exit, 退出)
F 字符串	搜索字符并显示该行。(F:Find, 查找)
I	插入一个新的步骤。(I:Insert, 插入)
L	显示前一步骤。(L:Last, 最后一行)
M /已存在的字符/新字符	用新的字符(串)取代已存在的字符(串)。 (M:Modify, 修改)
O	将光标放在当前步骤上进行编辑。 (O:One line, 一行)
P 步骤数量	显示指定数量的程序步骤。(P:Print, 打印)
R 字符串	在一个步骤中替换字符。(R:Replace, 替换)
S 步骤编号	选择程序步骤。(S:Step, 步骤)
XD	剪切选中的一个或多个步骤, 并保存在剪贴板上。
XY	复制选中的一个或多个步骤, 并保存在剪贴板上。
XP	粘贴剪贴板的内容。
XQ	以相反的顺序粘贴剪贴板的内容(反向粘贴)。
XS	显示剪贴板的内容。
T	在编辑模式中进行示教(选项)。

### 4.2.3 编程流程

请按如下步骤完成编程：



### 4.2.4 创建程序

在一个 AS 程序中，两件事情必须示教给机器人：

1. 机器人的工作条件
2. 机器人工具要遵循的路径(位姿)。

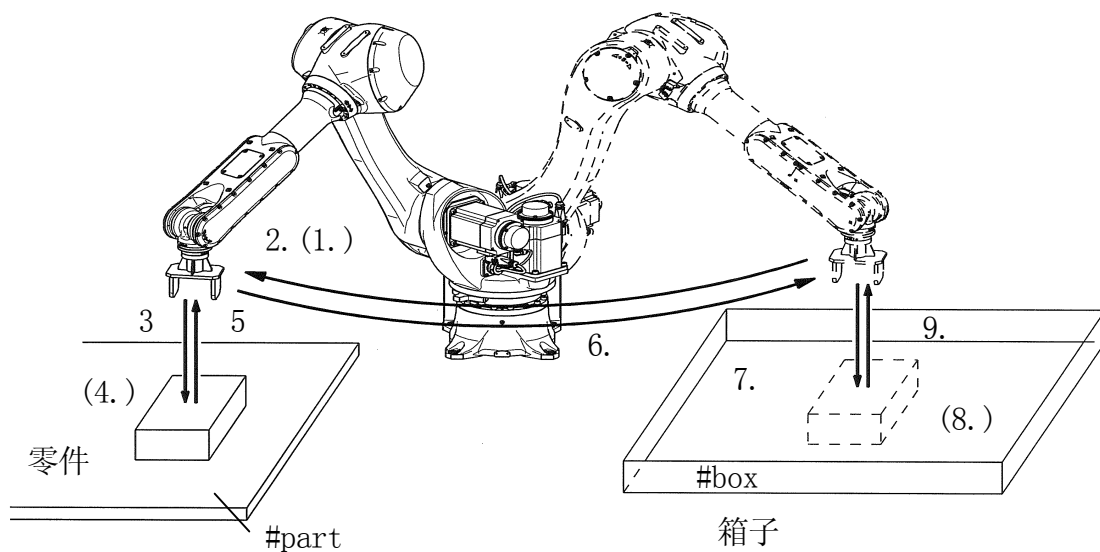
这里有一个示例程序。机器人将执行下页所示的任务：捡起给料器(传送装置)送来的零件，并把它放到箱子中。

首先定义完成任务所需的全部动作(运动)：

1. 检查手爪是否已打开。
2. 移动到给料器上零件(#part)上方 50 mm 的位姿。
3. 直线向下移动到零件(#part)处。
4. 关闭手爪抓住零件。
5. 直线向上移动到供料器上方 150 mm 处。
6. 移动到箱子(#box)上方 200 mm 处。

7. 向下移动零件到箱子中。
8. 打开手爪释放零件。
9. 移回到箱子上方 180 mm 处。

变量#part 和#box 表达了机器人的位置和姿态，在 AS 系统中被称为位姿数据。在执行程序之前，请按第 3 章中所示内容，指定位姿变量的名称。



程序通过 AS 编辑模式创建和编辑。创建名为“demo”的程序，输入“EDIT demo ”。  
屏幕显示如下内容：

```
> EDIT demo   
. PROGRAM demo  
1 ?
```

现在 AS 等待输入第一个步骤。在“1?”后面输入“OPENI ”

```
> EDIT demo  
. PROGRAM demo  
1 ? OPENI   
2 ?
```

接下来输入第二个步骤“JAPPRO #part, 50 ”

```
> EDIT demo
. PROGRAM demo
1 ? OPENI
2 ? JAPPRO #part, 50 
3 ?
```

用同样的方法输入程序的其他部分。在输入步骤时，可按 **Backspace** (**退格**) 键来修正错误，完成后按 。

如果在输入了有错误的步骤后按  键，将显示出错信息，并且该步骤被拒绝。这时，请重新输入步骤。在整个程序输入完成后，屏幕应该显示如下内容：

```
>EDIT demo
. PROGRAM
1 ? OPENI
2 ? JAPPRO #part, 50
3 ? LMOVE #part
4 ? CLOSEI
5 ? LDEPART 150
6 ? JAPPRO #box, 200
7 ? LMOVE #box
8 ? OPENI
9 ? LDEPART 180
10 ? E 
>
```

最后一步输入的“E ”不是机器人指令，而是退出编辑模式的指令（见 4.2.1 中的表格）。现在程序完成了。在程序执行时，AS 系统将按照步骤顺序依次执行，从步骤 1 到步骤 9。

关于如何创建程序的更多信息，请参阅 11. 示例程序。

### 4.3 执行程序


机器人控制程序和 PC 程序以不同的方式执行。

#### 4.3.1 执行机器人控制程序

要执行一个程序，请将 **TEACH/REPEAT** (示教/再现) 开关打到 REPEAT (再现) 位置。接下来，确认示教器的 **示教锁定** 开关在 OFF 位置。然后开启马达电源，并将 HOLD/RUN (保持/运转) 的状态由 HOLD (保持) 改变为 RUN (运转)。

##### 1. 通过 EXECUTE (执行) 指令运行程序

首先，设置监控速度。机器人将在执行程序时按此速度运动。速度必须设定得低于 30%，一开始的速度请设置为 10%。

```
> SPEED 10 
```


请用 EXECUTE 指令开始执行。输入如下指令：


```
> EXECUTE demo 
```


这时，机器人应该执行选定的任务。如果它不像预期的那样运动，请将 HOLD (保持) 状态改变为 RUN (运转) 状态。机器人将减速、停止。情况紧急时，请按下控制面板上或示教器上的 **紧急停止** 按钮。刹车动作、机器人立即停止。

如果机器人正以 10% 的速度动作，请逐渐升高速度。

```
> SPEED 30 
```


```
> EXECUTE demo  机器人以 30% 的速度操作。
```

```
> SPEED 80 
```

```
> EXECUTE demo  机器人以 80% 的速度操作。
```

在用发出 EXECUTE 指令一次后，就可以使用控制面板上的 **A** + **CYCLE START** (循环启动) 按钮来执行程序。


如果要执行程序多于一次循环，请在程序名字后面输入重复循环的次数。



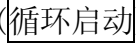
```
> EXECUTE demo, 5  执行 5 次。
```

```
> EXECUTE demo, -1  连续不断地执行程序。
```

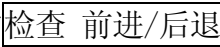
## 2. 通过 PRIME 指令(预先准备好)运行程序


用 EXECUTE 指令相同的方式设置监控速度，并执行 PRIME 指令。

>PRIME demo 

现在机器人准备执行此程序。按操作面板上的  +  () 开始执行。  
执行也可以用 CONTINUE 指令来起动。

## 3. 通过 STEP(步骤)指令或键运行程序

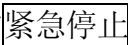
可以通过一步一步的单步执行程序，来检查程序的动作和内容。用 STEP(步骤)监控指令 或示教器上的 键\*。

**注释\*** 在用 键时，机器人在每个运动指令执行完毕后暂停执行。

在机器人执行过程中，有些监控指令不能使用。同样，在执行过程中也不能再次输入 EXECUTE 指令。

### 4.3.2 停止程序

有几种方法停止程序的运行。下面按从最紧急到最不紧急的顺序，说明三种方法。

1. 按控制面板或示教器上的按钮。刹车动作、机器人立即停止。除非有紧急情况，否则最好使用方法 2 和 3。
2. 将 HOLD/RUN(保持/运转)的状态由 RUN(运转)改变为 HOLD(保持)。机器人减速并停止。
3. 输入 ABORT 指令，使机器人在完成当前步骤(动作命令)后停止。

> ABORT 

HOLD 指令也可以用来停止程序执行。

> HOLD 

### 4.3.3 继续机器人控制程序

取决于程序是如何停止的，这里有几种恢复继续执行程序的方法。

1. 如果机器人是用[紧急停止]按钮停止的，请释放[紧急停止]的锁定，并开启马达电源。按 [A] + [CYCLE START (循环启动)] 按钮后，机器人开始运动。
2. 当 [HOLD (保持)] 状态用于来停止机器人时，按 [A] + [RUN] 将其状态改变为 RUN (运转)。
3. 要从 ABORT 或 HOLD 指令后或程序执行因出错而中止的情况中恢复执行，请用 CONTINUE 指令恢复继续执行程序。（当从出错状态重新执行时，在恢复继续程序前，错误必须被复位。）

> CONTINUE [ ]

### 4.3.4 执行 PC 程序

PC 程序通过 PCEXECUTE 监控指令执行，或通过机器人控制程序内部程序命令的执行来执行。PCABORT 指令可在任何时候停止 PC 程序的执行。PCEND 指令用于在当前的循环完成后结束程序的执行。

PCCONTINUE 指令恢复因 PCABORT 或出错而中止的程序继续执行。（当从出错状态重新执行时，在恢复继续执行程序前，错误必须被复位）

## 4.4 程序的执行过程

程序命令是从程序的顶部到底部按顺序执行的。但是如果出现像 GOTO 或 IF...GOTO 这样命令，顺序流程就会出现改变。CALL 指令调出并执行一个不同的程序，但是这并不改变流程的顺序。执行到 RETURN 指令时，进程返回主叫程序并从离开的位置继续执行。

WAIT(等待)命令停止程序进入下一个步骤，直到指定的条件被满足。PAUSE(暂停)和 HALT(立定)命令在使用这些命令的步骤处停止程序。

STOP 命令在有些情况下会不停止执行。如果指定的周期还有剩余，将从主程序的第一步继续执行。(即使 STOP 指令在子程序中被执行到，执行将返回到主程序的开始位置继续)。如果已没有剩余周期，执行将停止在使用该命令的步骤处。

### 4.4.1 子程序

主程序可以被临时挂起，调用并执行一个不同的被叫做子程序的程序。使用子程序，可以使程序变成一个更易理解的模块化结构程序。

### 4.4.2 带参数的子程序

参数可更方便地和子程序一起使用。例如，如果有一个计算需要输入不同的数据，并且重复执行，请为此计算创建一个子程序。使用 CALL 命令跳向子程序，并且使用输入数据作为计算中的参数。(见下面的例子 1、2)

最多可以使用 25 个实型变量、位姿变量或字符串变量参数。变量的类型在主程序和子程序中必须相同。当指派一个参数名给变换值时，为了区分于实数变量，要在参数变量名称前加一个符号“&”。同时，在 CALL 的目标程序(子程序)中使用局部变量。



**例 1** 实数变量“c”的值是输入数据“a”和“b”之和。

```
main()
1  a=1
2  b=2
3  CALL calc(a, b, c)
4  TYPE c
calc(. aa, . bb, . cc)
1  . cc=. aa+. bb
```

**例 2** 变换值“c”的值是变换值“a”和“b”之和。

```
pose()
1  point a=trans(10)
2  point b=trans(0, 20)
3  CALL add(&a, &b, &c)
4  point d=c
add(. &aa, . &bb, . &cc)
1  point . cc=. aa+. bb
```

[ 注 意 ]

为了在子程序中设置参数，请像上述例 1 那样，输入“EDIT calc, 0”，然后出现如下显示：

0. ()

0?

在 ? 之后输入 (. aa, . bb, . cc)。

#### 4. 4. 3 异步处理(中断)

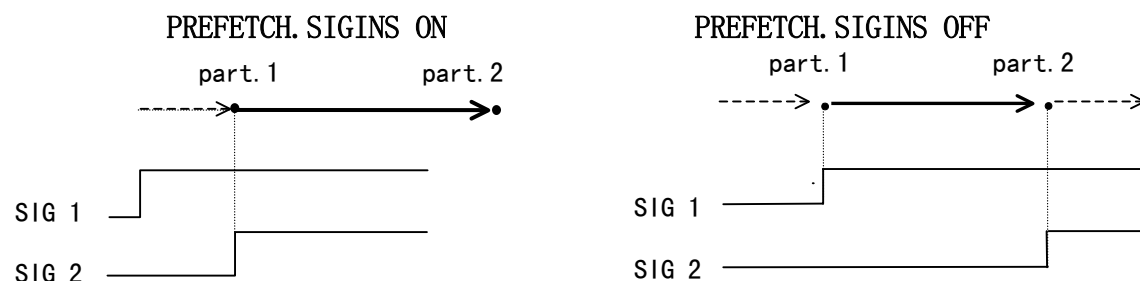
在一定的条件下，例如当出现故障或当指定的外部信号输入时，程序执行可被中断而执行另一个程序。这在主程序的执行流程之外独立出现，被称为异步处理(中断)。一旦检测到指定信号(例如一个外部信号或一个错误)，中断发生，而不管主程序的执行。这个过程用 ON(或 ONI) ...CALL 命令激活。

## 4.5 机器人动作

### 4.5.1 机器人动作和程序步骤执行的时序

在 AS 系统中，程序执行和机器人动作的时序可以通过系统开关的设置而改变。例如，当 PREFETCH. SIGINS 开关为 ON (允许提前处理信号 I/O 指令或 OFF 时 (不允许提前处理信号 I/O 指令)，步骤执行的时序出现变化。

```
JMOVE part1  
SIGNAL 1  
JMOVE part2  
SIGNAL 2
```



当 PREFETCH. SIGINS 为 ON 时，只要机器人开始向 part1 移动，外部信号 1 (SIGNAL 1) 立即被输出。当程序到达第二个 JMOVE 命令时，等待直到机器人到达 part1，然后执行该命令。一旦机器人到达 part1，它开始移向 part2，同时，外部信号 2 (SIGNAL 2) 被输出。

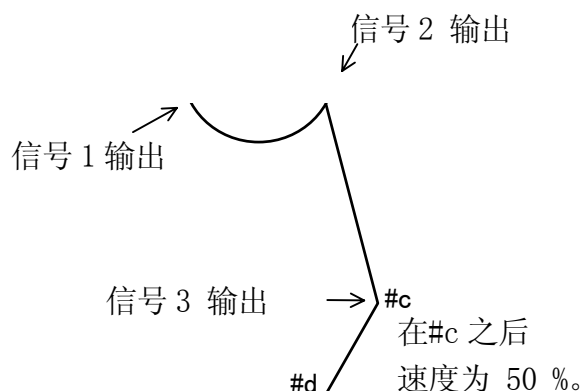
当 PREFETCH. SIGINS 为 OFF 时，在机器人到达运动命令的目的地并且轴一致之后，输出信号。

下面的例子演示了当 PREFETCH. SIGINS 为 ON 时，程序步骤是怎样在 AS 系统中执行的。

```

1 JMOVE    #b
2 SIGNAL    1
3 a=2
4 LMOVE    #c
5 SIGNAL    2
6 SPEED     50
7 LMOVE    #d
8 SIGNAL    3

```



当 PREFETCH. SIGINS 为 ON 时，事先处理信号，因此，在机器人开始执行当前的运动指令时，立即执行到下一个运动指令前的所有指令。如果上述程序在机器人在#a 处时执行，执行步骤顺序如下：

1. 在#a 处，机器人为 JMOVE #b 作运动规划，并且开始向#b 运动。
2. 一旦运动开始，下一步骤 SIGNAL 1 被马上执行，也就是机器人离开#a 后，信号 1 立即变成 ON。
3. 执行进程到达步骤 4，规划 LMOVE #c 并等待机器人到达#b。
4. 一旦机器人到达#b，机器人立即开始向#c 运动。执行进程到达步骤 7(规划 LMOVE #d 的运动)，并等待机器人到达#c。

#### [ 注 意 ]

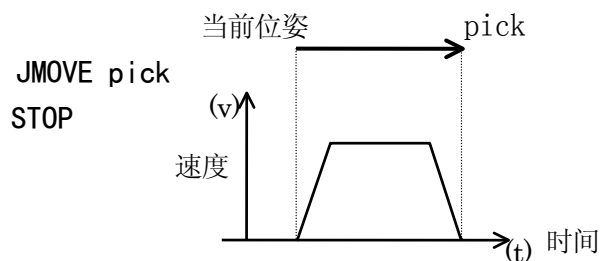
当 PREFETCH. SIGINS 为 ON 时，程序处理下一步直到不得不等待机器人到达指定的位姿。但是，时序受其他设置和指令/命令的影响，如 WAIT 命令或 CP 开关。WAIT 指令挂起步骤的进程直到条件被满足。当 CP 开关为 OFF 时，程序处理运动命令步骤前的所有步骤，并在处理前在那里停止运动。在编程时，请时刻关注系统开关和命令的设置。

如同这里的演示，重要的是注意 AS 系统处理程序的时序、程序中机器人运动受系统开关设置的影响、以及一些稳态程序命令。在编程过程中，请特别注意信号的输出时序。

关于各系统开关的详情，请参阅 7.0 AS 系统开关或操作手册。

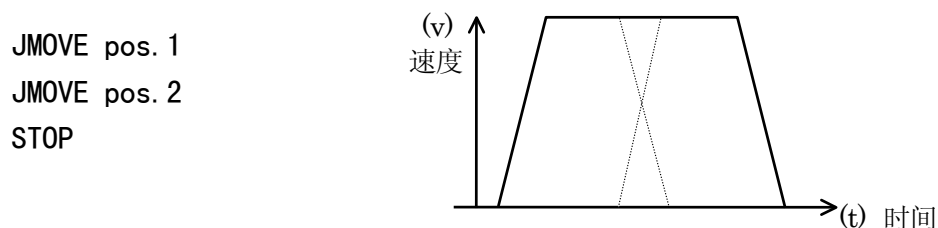
### 4.5.2 连续路径 (CP) 运动

本例揭示了运动命令的执行方法。



当执行上述运动命令时，机器人在向“pick”移动的同时，平滑地加速到当前设置的速度。当机器人接近“pick”时，它逐渐减速直到停在那个位姿上。一系列这样的动作，由一条运动命令执行，被称为一个“运动段”。

在下图所示的情况下，如果 CP 系统开关为 ON，机器人先加速到指定速度，但在接近 pos. 1 时并不减速。取而代之的是，它平滑地转向 pos. 2 移动。当机器人接近 pos. 2 时，它逐渐减速并停在那个点上。这个运动由 2 个运动命令组成，因此由两个运动段构成。



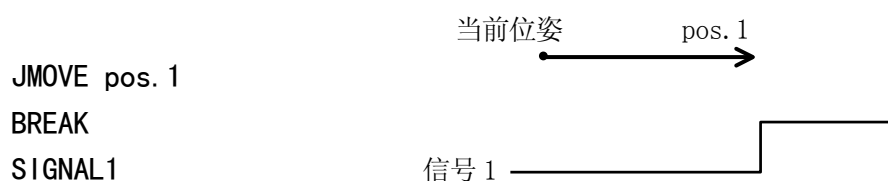
像这样的运动，机器人执行了一系列的运动，在运动段之间作出了平滑的转向，而并不在各目标位姿停止，被称为 CP (连续路径) 运动。将 CP 系统开关设置为 OFF，可禁止 CP 功能。如果 CP 开关为 OFF，机器人将在每个动作段的结束处减速并停止。（见 5.6 中 SWITCH 和 ON/OFF 指令，6.9 中如何设置 CP 开关的 ON/OFF 命令）

CP 运动能用于直线运动和关节插补运动 (各轴插补、或称各轴轨迹) 或者他们的组合。例如，CP 运动可以用于下列所有步骤的全过程：

直线运动 (如 LDEPART) → 关节插补运动 (如 JAPPRO) → 直线运动 (如 LMOVE)

### 4.5.3 CP 运动的中断

有些命令可以暂停程序的执行直到机器人真正到达目标位姿。这被称为 CP 运动的中断 (Break in CP motions)。如果机器人必须在执行稳态操作时 (如关闭手爪) 保持静止, 就可以使用执行命令。



JMOVE 命令开始移动机器人向 pos. 1 运动。接下来, 执行 BREAK 命令。这条命令挂起程序的执行, 直到 pos. 1 的移动完成。因此, 外部信号在机器人停止前不输出。

下面的命令挂起 (暂停) 程序的执行, 直到机器人运动完成。但是, 请小心不要在必须保持运动的地方使用这些命令。

BASE	BREAK	BRAKE	CLOSEI	HALT	OPENI	PAUSE	RELAXI	TOOL
ABOVE	BELOW	DWRIST	UWRIST	LEFTY	RIGHTY			

除了上述的之外, ONI 指令也中断程序的执行, 但是注意, 由 ONI 命令设置的打断可以发生在运动段的任何地方。

#### [ 注 意 ]

1. 如果在当前运动完成前未给出一个命令, 机器人将减速、停止。引起这种情况的原因有:
  - (1) 执行了 WAIT 命令, 但是继续程序的条件在机器人运动完成前未满足。
  - (2) 下一运动命令之前的程序步骤未在当前运动完成前执行完毕。
2. 在以 CP 运动移动时, 计算运动段之间的平稳转向 (过渡) 需要一定的时间。因此, 如果两个位置之间的距离太短, 在到达第二个位置之前计算可能还未完成, 因此导致机器人在运动段之间停顿。为避免发生这样的情况, 必须减慢速度。如果不能减速, 不要指定太靠近的位姿。

#### 4.5.4 CP 开关和 ACCURACY、ACCEL、DECEL 命令之间的关系

ACCURACY命令：设置各运动段末端处机器人的定位精度。（当机器人进入此命令设置的范围内时，它就认为已经到达了目标位姿，并开始向下一个目标运动。）

·ACCEL 命令： 设置机器人运动起始处的加速度。

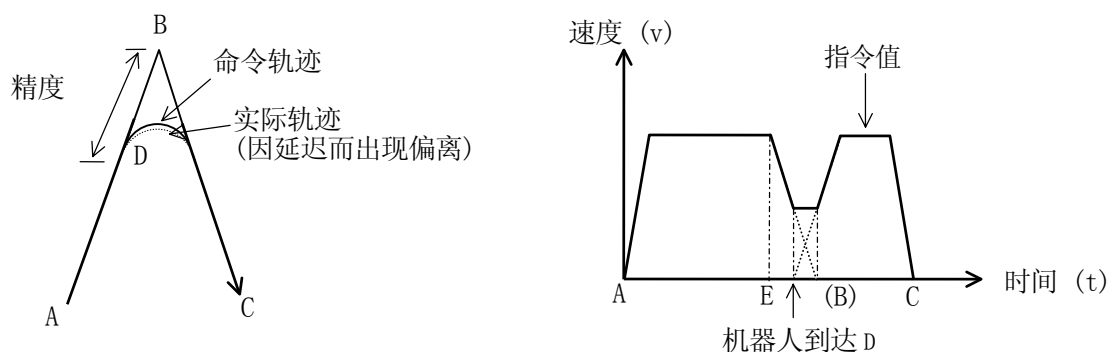
·DECEL 命令： 设置机器人运动结束处的减速度。

·CP 开关： 允许或禁止CP运动。

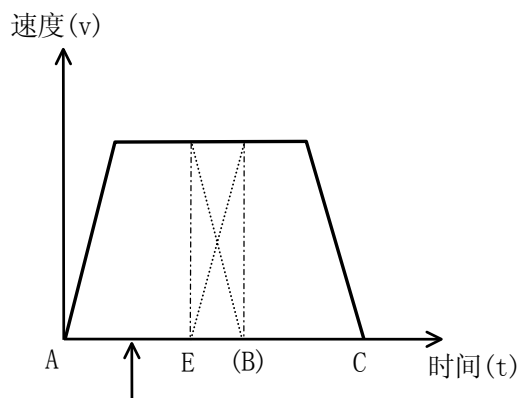
##### 4.5.4.1 CP ON: 运动类型 1(标准)

例如机器人在 CP 开关 ON 时进行如下运动：A →B→C。

一旦机器人的当前姿态值进入精度范围(例如机器人到达点 D)，当前运动路径值和下一路径的运动指令值的重叠开始。机器人将根据这些命令值，持续地向下一个路径改变运动。

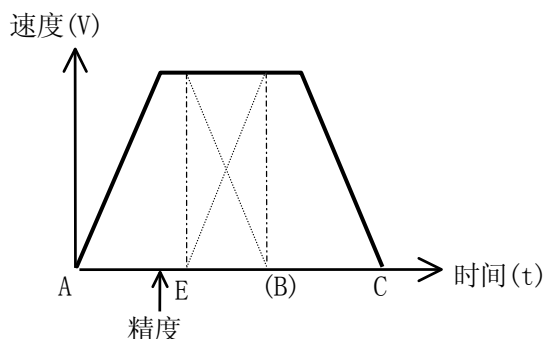


由 ACCURACY 指定得精度范围越大，重叠开始得越早。但是，下一个路径的加速不会在机器人开始减速的地方(点 E)之前开始。因此可以说，ACCURACY 作用在到达一定值时饱和，例如如果设置的精度值大于了点 B 和 E 之间的距离，此精度值将不起作用。（见下图）

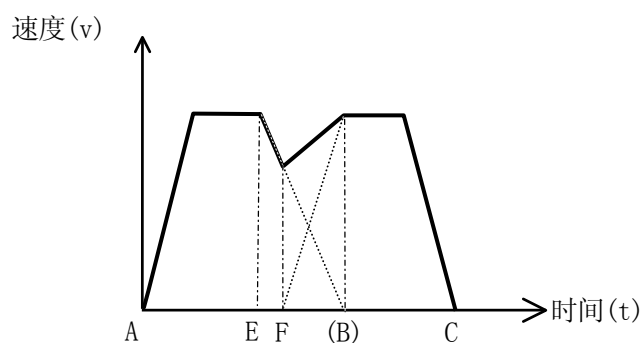


即使命令值此刻已到达精度点，下一个路径的加速仍不会开始，直到点 E 处的减速开始。

如果加速和减速的值设置得越小，重叠开始得就越早，机器人将以更大半径的轨迹运动，但是它到达点 C 所花时间的差别不会很大。



即使减速度减小而下一个路径的加速度增加，复合速度也将不会超过指定的最大速度，因为机器人直到点 F(加速度开始的点)重叠才开始。换句话说，完成减速和加速的时间是相同的(点 B)。



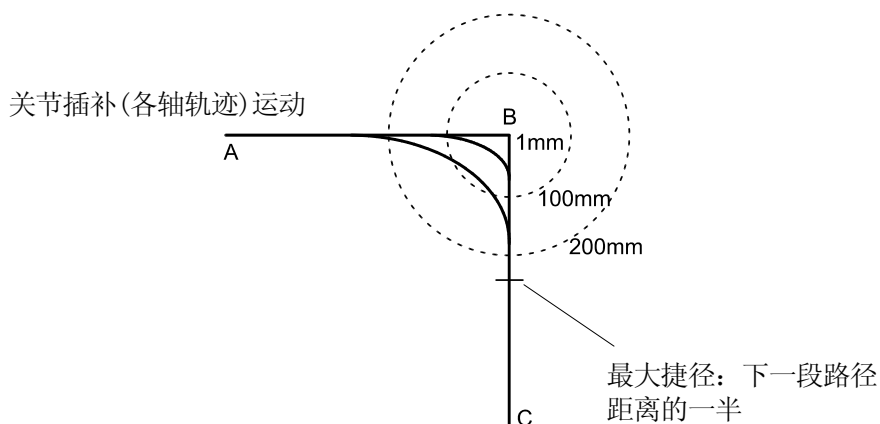
#### 4.5.4.2 CP ON: 运动类型 2

在运动类型 2 中，直线运动和圆弧运动中的精度和速度的概念不同于上面描述的运动类型 1。运动类型 1 和运动类型 2 可以使用相同的程序而无需修改，但是实际运动路径和运动速度将有变化。

##### 1. 精度设置

##### (1) 关节插补(各轴轨迹)运动中的精度

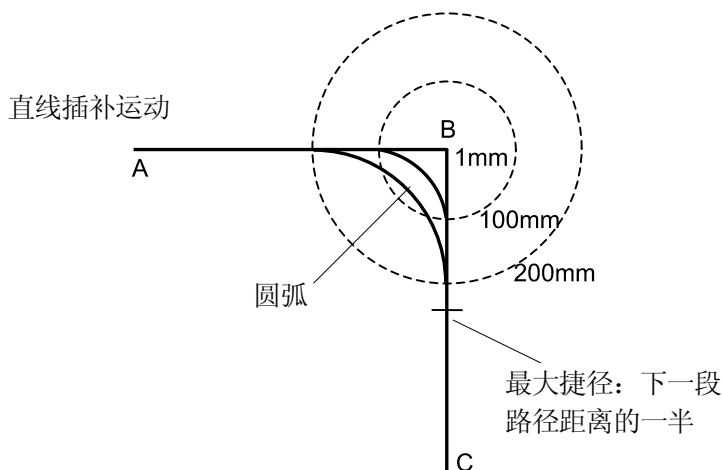
对应于精度设置的机器人的运动路径如下图所示。例子中点 B 处的精度值为 1 mm、100 mm、和 200 mm。与标准运动相同的方式，机器人在到达点 B 前开始走捷径，但不必在进入精度范围的点处开始转向。转向前机器人要如何接近点 B，是由各轴角度按照精度值的比例进行计算的。通过设置更大的精度值，机器人可以走更短的捷径，包括当前路径剩余距离更缩短或将下一段路径从 B 到 C 的距离缩短为一半。



## (2) 直线和圆弧插补运动中的精度

对应于精度设置的机器人的运动路径如下图所示。此例中点 B 处的精度值为 1 mm、100 mm、和 200 mm。机器人在进入精度范围的点处开始转向。机器人将沿落在精度范围内的圆弧轨迹运动。

通过设置更大的精度范围，机器人可以走更短的捷径，包括当前路径剩余距离更缩短或将下一段路径从 B 到 C 的距离缩短为一半。精度值可被设置到等于第二条路径一半距离的值。



通过走捷径，可缩短周期时间。但是，在设置了下述条件时，精度设置的处理与标准运动的相同：

- 在点 B 处执行一个等待命令时 (TWAIT、SWAIT 等)
- 在点 B 处，更换工件/工具时
- 在下一点的插补模式变为关节插补(各轴轨迹)时
- 在点 B 处运动模式改变时(正常模式↔基于固定工具坐标系的运动)
- 由于使用了条件命令如 IF 指令等，而使进程出现分支时

## 2. 速度设置

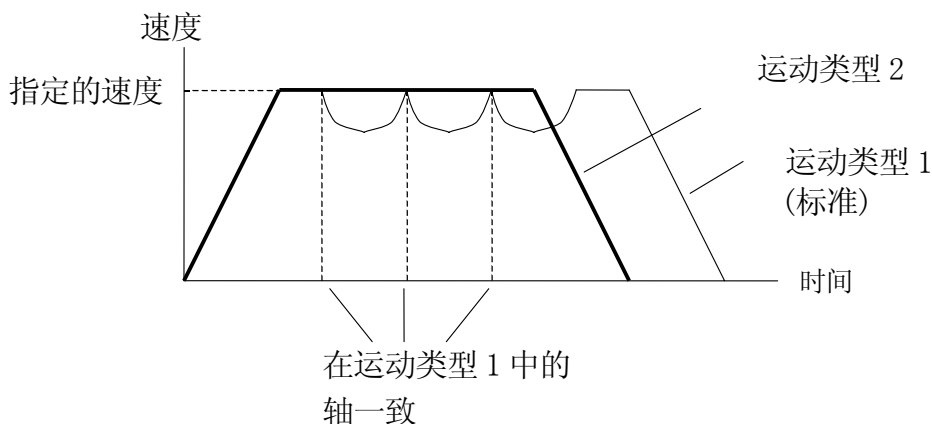
### (1) 关节插补运动中的速度

与运动类型 1 相同

### (2) 直线和圆弧插补运动中的速度



在运动类型 2 中，如果精度值设置的很大，机器人的形态两个定义的位姿之间不改变，即使两个位姿之间的距离很小，也可以获得指定的速度。



但是，如果设置了下列条件时，处理将与运动类型 1 相同：

- 在点 B 处执行一个等待命令时 (TWAIT、SWAIT 等)
- 在点 B 处，更换工件/工具时
- 在下一点的插补模式变为关节插补 (各轴轨迹) 时
- 在点 B 处运动模式从普通模式 (工件固定，工具移动) 改变时
- 由于使用了条件命令如 IF 指令等，而使进程出现分支时

### 3. 圆弧插补中的速度

在运动类型 2 中，最大速度被按照机器人执行正确的圆弧插补运动的能力自动设置。

#### [ 注 意 ]

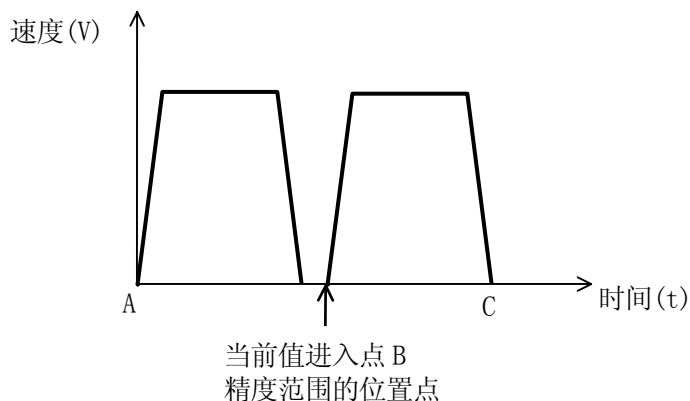
当试图执行一个在短距离内需机器人姿态有很大改变的程序时，改变姿态所花的时间将超过以指定速度移动那段距离的时间。这种条件下，关节移动优先，因此该运动将不会到达指定的速度。

在运动类型 2 中，机器人将沿精度范围内的圆弧轨迹运动。此轨迹的最大速度由机器人的能力进行设置。

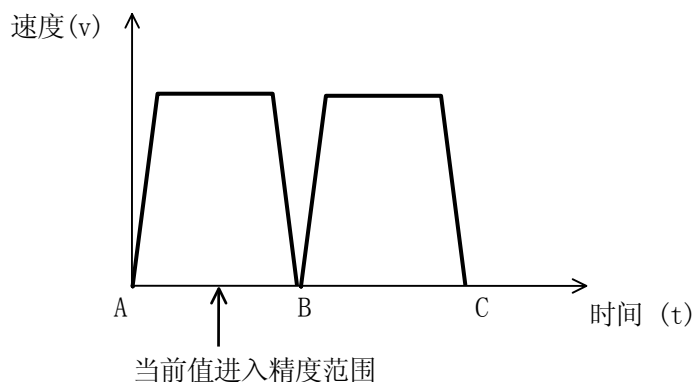
#### 4.5.4.3 CP OFF

当 CP 开关为 OFF 时，没有运动的重叠。在第一个运动段完成，并且当前值进入 ACCURACY\* 范围之后，第二个路径才开始加速。

**注释\*** 例如，对于 FS10 来说，默认值是 1 mm。



当 CP 开关为 OFF 时，第二条路径只有在第一个运动的速度减到 0 时才开始运动，即使精度范围被设置得大于第一条路径的末端。



#### 4.5.5 沿指定路径的运动

直线插补和关节插补运动是所有机器人的标准功能。但是，有时需要机器人沿着指定或计算的路径运动。AS 系统可以在机器人移动过程中进行计算，生成可能的复杂运动。这个特点被称为“沿指定路径的运动”。

系统可通过程序循环实现此运动，在运动命令执行的同时，程序循环进行一系列短距离的连续计算，这样一个程序循环是可以的，因为 AS 可以在机器人运动的同时执行非运动命令。计算出来的运动段用 CP 功能可以平滑地连接起来。

下面是沿指定路径运动的一个程序例子。机器人工具将沿着一系列由数组变量“path”定义的位置数据运动。

```
FOR  index=0 TO 10  
LMOVE  path[index]  
END
```

数组变量 path[0]到 path[10]可由手动示教或计算来定义。

#### 4.5.5.1 设置负荷数据

通过设置机器人当前动作的负荷数据，可自动为该负荷确定最佳的加速和减速。请按照机器人当前的动作，设置正确的负荷数据。



#### 小 心

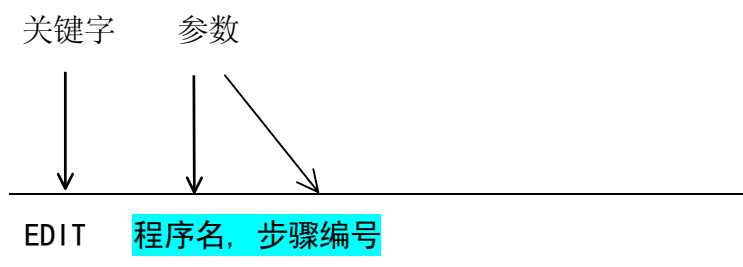
一定要设置正确的负荷质量和重心的位置。不正确的数据会削弱或缩短零件的寿命，或者会引起超载/偏差故障。更多详情，请参阅 WEIGHT 指令/命令。

负载数据可以用辅助功能 AUX 0406 自动负荷检测来自动设置。详见操作手册。

## 5.0 监控指令

本章将监控指令分组成如下目录，并详细描述了各指令。监控指令由表达指令的关键字和关键字后续的参数组成，如下面的示例所示。

### 示例



具有        标记的参数可以省略  
在关键字和参数之间必须输入空格。  
各例子中的    代表 登录 键。

## 5.1 编辑指令

EDIT	启动程序的编辑。
C	结束当前程序的编辑，切换到另一个程序。 (C: Change, 更改)
S	选择要显示程序步骤。(S: Step, 步骤)
P	显示指定数量的程序步骤。(P: Print, 打印)
L	选择前一步骤。(L: Last, 最后一行)
I	插入一个新的步骤。(I: Insert, 插入)
D	删除程序步骤。(D: Delete, 删除)
F	搜索字符。(F: Find, 查找)
M	替换字符。(M: Modify, 修改)
R	替换字符。(R: Replace, 替换)
O	将光标放在当前步骤上。(O: One line, 一行)
E	退出编辑模式。(E: Exit, 退出)
XD	剪切选中的一个或多个步骤，并保存在剪贴板上。
XY	复制选中的一个或多个步骤，并保存在剪贴板上。
XP	粘贴剪贴板的内容。
XQ	以相反的顺序粘贴剪贴板的内容(反向粘贴)。
XS	显示剪贴板的内容。
T	在编辑模式中示教运动指令(选项)。

---

**EDIT 程序名 , 步骤编号**

---

### 功能

进入可以创建和编辑程序的编辑模式。

### 参数

#### 程序名

选择一个要编辑的程序。如果未指定程序名，则打开编辑上一次编辑或停止(或因错误停止)的程序。如果指定的程序不存在，创建一个新的程序。

#### 步骤编号

选择开始编辑的步骤编号。如果不指定步骤编号，从被编辑过的上一步开始编辑。如果上一次程序执行过程中出现错误，那么选择出错处的那个步骤。

---

[ 注 意 ]

---

不能编辑正在执行过程中的程序。

不能执行或删除正在编辑中的程序。如果一个程序调用了正在编辑中的程序，将报出错，该程序的执行停止。

---

**C** 程序名, 步骤编号

---

**功能**

改变编辑模式中当前选择的程序。(C: Change, 更改)

**参数**

程序名

选择要编辑的程序。

步骤编号

选择要开始编辑的步骤编号。如果未指定步骤，选择程序的第一步。

---

**S** 步骤编号

---

**功能**

选择和显示指定的步骤进行编辑。(S: Step, 步骤)

**参数**

步骤编号

如果未指定步骤编号，则自动选择程序的第一步。如果步骤编号大于程序中的步骤数，则选择程序最后一步后的一个新步。

---

**P** 步骤数量

---

**功能**

显示当前步骤开始的指定数量的步骤。(P: Print, 打印)

**参数**

步骤数量

设置要显示的步骤的数量。如果步骤数量未指定，则只显示当前步骤。

**说明**

仅显示指定数量的步骤。列表中的最后一步准备好进行编辑。

---

**L**

---

**功能**

显示前一个步骤(上一步)进行编辑。([当前步骤编号] - 1 = [要显示的步骤编号])

(L: Last, 最后一行)



---


## I

---

### 功能



插入行到当前步骤之前。(I: Insert, 插入)

### 说明




插入行之后的步骤被重新编号。要退出插入模式，按  键。所有在退出插入模式前写入的行被插入到了程序中。

### 示例

插入 CLOSEI 命令到步骤 3 和 4 之间。

1?	OPENI	
2?	JAPPRO #PART, 500	
3?	LMOVE #PART	
4?	LDEPART 1000	
5?	S 4	;显示步骤 4，希望在此步骤前插入一行。
4	LDEPART 1000	
4?	I	;键入 I 指令。
4I	CLOSEI 	;为插入的行键入命令。
5I		;按 enter(登录)结束插入行
5	LDEPART 1000	;原步骤 4 现在被重新编号为步骤 5。
5?		

### [ 注意 ]

要插入空白行，请在插入模式中，按  键或  (制表符)，然后按  。

---

## D 步骤数量

---

### 功能

删除包括当前步骤在内的指定数量的步骤。

### 参数

步骤数量

指定从当前步骤开始的要删除的步骤数量。如果未指定数量，仅删除当前步骤。

### 说明

仅删除从当前步骤开始的指定数量的步骤。一旦删除，所有遗留的步骤将自动重新编号和显示。

### [ 注 意 ]

如果指定的步骤数量大于程序的步骤数量，删除当前步骤之后的全部步骤。

---

## F 字符串

---

### 功能

在当前程序中，从当前步骤到最终步骤，搜索(查找)指定的字符串，并且显示包含此字符串的第一个步骤。（F: Find, 查找）


### 参数

字符串

指定要搜索的字符串。

### 示例

在当前步骤之后的步骤中搜索字符串“abc”，并显示包含此字符串的步骤。

```
1?F abc   
3      JMOVE abc  
3?
```

---

## **M /已存在的字符/新字符**

---

### **功能**

修改当前步骤中的字符(串)。(M: Modify, 修改)

### **参数**

已存在字符(串)


指定当前步骤中的那个字符(串)需要改写。

新字符(串)

指定用于取代已存在字符(串)的新字符(串)。

### **示例**

将步骤 4 中的位姿变量 abc 修改为 def。

```
4   JMOVE abc  
4?M/abc/def   
4   JMOVE def  
4?
```

---

## R 字符串

---

### 功能

用指定的字符(串)替换当前步骤中已存在的字符(串)。(R: Replace, 替换)





### 参数


字符串

指定新字符(串)用于替换已存在的字符

### 说明


使用 R 指令的过程如下所示:

1. 用  键, 移动光标到要替换的第一个字符下面。
2. 按  键, 然后 .
3. 输入新的替换字符(串)。注意输入的字符(串)并不是替换光标上面的字符, 而是往左两格, 也就是从 R 上面的字符开始。(见下例)
4. 按 .

一旦按下  键, AS 系统立即检查此行是否正确。如果有错误, 输入将被忽略。

### 示例

用 R 指令改变速度从 20 到 35。

```
1      SPEED 20 ALWAYS
1?           R 35 
1      SPEED 35 ALWAYS
1?
```

---



## 0





---

### 功能

将光标放在当前步骤上进行编辑。(0: One line, 一行, (注意不是零))

### 示例

用 0 指令把姿态变量 abc 改成 def。用  或  键移动光标。

```
3      JMOVE abc
3?O 
3      JMOVE abc  ; 用  删除 “abc”
3      JMOVE def  ; 输入 “def”
3      JMOVE def
3?
```

### [ 注 意 ]

此指令在示教器上不能使用。

---

## E

---

### 功能

从编辑模式退出并返回监控模式。(E: Exit, 退出)

---

<b>XD</b>	<b>步骤数量</b>
-----------	-------------

---

### 功能

从程序中剪切指定数量的步骤，并将它们存储在粘贴缓冲器(剪贴板)中。

### 参数

#### 步骤数量

指定剪切的步骤数量，剪切的步骤从当前步骤开始。步骤数量最多为 10，即最多可剪切 10 个步骤。如果未指定，仅剪切当前步骤。

### 说明

剪切指定数量的步骤，并将它们存储在粘贴缓冲器中。

XY 指令只是复制，不剪切步骤，但是 XD 指令剪切步骤。程序中的剩余步数被相应地重新编号。

---

<b>XY</b>	<b>步骤数量</b>
-----------	-------------

---

### 功能

复制指定数量的步骤，并将它们存储在粘贴缓冲器(剪贴板)中。

### 参数

#### 步骤数量

指定复制并存储到粘贴缓冲器中的步骤数量。步骤数量最多为 10，即最多可复制 10 个步骤。如果此数量未指定，仅复制当前步骤。

### 说明

复制包括当前步骤的指定数量的步骤，并存储他们在粘贴缓冲器中。

XD 指令剪切步骤，而 XY 指令是复制步骤。在使用 XY 指令后，程序保持相同，步骤数量不改变。

---

## **XP**

---

### **功能**

在当前步骤前，插入粘贴缓冲器中的内容。

### **说明**

在此指令前，请先用 XD 或 XY 指令把想要的内容存储到粘贴缓冲器中。

---

## **XQ**

---

### **功能**

在当前步骤前，以相反的顺序插入粘贴缓冲器中的内容。

### **说明**

以和 XP 指令相反的顺序插入粘贴缓冲器中的内容。

---

**XS**

---

**功能**

显示粘贴缓冲器的内容。

**说明**

显示粘贴缓冲器的当前内容。如果粘贴缓冲器是空的，则不显示任何内容。



T 位姿变量

选项

功能

在编辑模式中，允许使用示教器进行运动命令(JMOVE、LMOVE 等)的示教。

参数

位姿变量


指定要示教的目标位姿变量名，用变换值或关节位移值(各轴值)表达。如果用 A[ ]的形式指定，它将被读作为数组变量。这时，变量不能用于元素编号(下标)。如果省略，把当前关节位移值(各轴值)示教为常数(位姿常数)。

说明

请在编辑模式中输入本指令。执行时，示教器显示一个专用的示教画面。这里示教的动作被记录为程序中的命令，并被写在输入 T 指令的那个步骤上。当示教多个步骤时，变量被用增加变量名字最后的数字来重命名。更多详情请参阅操作手册。


示例

带位姿变量


```
2 JAPPRO #a
3? T pos          用示教器示教，按  键返回 AS。
                   (这里示教三个步骤)


3 JMOVE pos0
4 JMOVE pos1
5 LMOVE pos2
  :
```

不带位姿变量

```
2 JAPPRO #a
3? T              用示教器示教关节值(各轴值)，按  键结束。
                   (这里示教两个步骤)

3 JMOVE #[0, 10, 20, 0, 0, 0]
4 JMOVE #[10, 10, 20, 0, 0, 0]
  :
```

 警告

要使用此指令，示教器必须与控制器连接。同时，机器人必须在示教模式，示教锁定为 ON。

## 5.2 程序和数据控制指令

USB_FDIR	USB 闪存中程序名和变量名列表。
LIST	显示所有程序步骤和变量值。
LIST/P	显示所有程序步骤。
LIST/L	显示所有位姿变量和他们的值。
LIST/R	显示所有实型变量和他们的值。
LIST/S	显示所有的字符串变量和他们的数据。
DELETE	删除机器人存储器中的程序和变量。
DELETE/P	删除机器人存储器中的程序。
DELETE/L	删除机器人存储器中的位姿变量。
DELETE/R	删除机器人中的实型变量。
DELETE/S	删除机器人存储器中的字符串变量。
USB_FDEL	删除 USB 闪存中的文件。
RENAME	改变程序的名称。
USB_RENAME	改变 USB 闪存中的程序名。
XFER	从一个程序复制步骤到另一个程序。
COPY	复制程序。
USB_COPY	复制 USB 闪存中的文件。
TRACE	开/关 TRACE (跟踪) 功能。(选项)
SETTRACE	为日志保留存储器。(选项)
RETRACE	释放 SETTRACE 保留的存储器。(选项)
LSTRACE	显示日志数据。(选项)

---

USB_FDIR	文件夹名
----------	------

---

### 功能

显示USB闪存中的文件夹名。



### 说明

文件夹名

指定文件夹，其包括要显示的文件列表。如果省略，则可以显示 USB 闪存上的根文件夹中的所有文件。

### 示例

>USB\_FDIR  显示 USB 闪存上所有的文件名称。

当 SCREEN 开关为 ON 时，显示不会滚动，并停在画面的底部。要继续显示，可按  键。  
要结束显示，按 。

LIST	程序名, .....
LIST/P	程序名, .....
LIST/L	位姿变量, .....
LIST/R	实型变量, .....
LIST/S	字符串变量, .....

功能

显示指定的程序和数据。





参数

程序名 (/P)，位姿变量 (/L)，实型变量 (/R)，字符串变量 (/S)  
指定要显示的数据类型。如果未指定，显示存储器中的所有数据。如果选择了数组变量，把数组变量的所有成员显示在屏幕上。

说明

LIST 指令显示所有的程序名、他们的子程序和变量。换句话说，LIST/P 指令仅显示主程序的内容。

示例

- >LIST 显示全部程序的内容，包括变量和他们的值。
- >LIST/L 显示全部位姿变量和他们的值。
- >LIST/R 显示全部实型变量和他们的值。
- >LIST test\* 显示全部以“test”开头的程序的内容、他们的子程序及变量。

当 SCREEN 开关为 ON 时，显示不会滚动，并停在画面的底部。要继续显示，可按键。  
要结束显示，按.

---

<b>DELETE</b>	<b>程序名,</b>	.....
<b>DELETE/P</b>	<b>程序名,</b>	.....
<b>DELETE/L</b>	<b>位姿变量,</b>	.....
<b>DELETE/R</b>	<b>实型变量[数组元素],</b>	.....
<b>DELETE/S</b>	<b>字符串变量[数组元素],</b>	.....

---

**功能**  
从存储器中删除指定的数据。

**参数**  
程序名 (/P)，位姿变量 (/L)，实型变量 (/R)，字符串变量 (/S)  
指定要删除的数据类型。





**说明**  
DELETE 指令完全删除指定的程序；也就是主程序本身及其如果使用的下列数据。(但是，不删除其他程序也使用的数据)。

- 所有此程序及其子程序调用的子程序。
- 所有此程序及其子程序使用的位姿变量。
- 所有此程序及其子程序使用的实型变量。
- 所有此程序及其子程序使用的字符串变量。

DELETE/P 指令，和 DELETE 指令不同，仅删除程序本身，不删除其子程序和该程序使用的变量。

如果在使用DELETE/R 和 DELETE/S 指令时未指定数组元素,将删除数组变量中的全部元素。如果指定了元素，仅删除指定的元素。

**示例**

>DELETE test		删除程序“test”，全部子程序和它们使用的变量。
>DELETE/P pg11, pg12		删除程序“PG11”和“PG12”。(不删除子程序和变量)。
>DELETE/R a		删除数组变量“a”的所有元素成员。
>DELETE/R a[10]		删除数组变量“a”的第 10 个元素成员。

---

**USB\_FDEL 文件名, .....**

---

**功能**

从 USB 闪存中删除指定的文件。

**参数**

文件名

指定要删除文件的名称。为了指定包括文件在内的文件夹，在文件名前添加“文件夹名¥”。  
在文件名前写“CF¥”来指定在控制器中的袖珍闪存的文件。

**说明**

完全删除指定文件中的程序。

---

### RENAME 新程序名=已存在的程序名

---

#### 功能

更改当前保存在存储器中的一个程序的名称。

#### 参数

新程序名

为程序设置新名称。


已存在的程序名

指定程序的当前名称。

#### 说明

如果新程序名已经存在，RENAME 指令导致出错。

#### 示例

>RENAME test=test.tmp  更改软盘中程序的名称，从“test.tmp”改到“test”。

---

### USB\_RENAME 新文件名=已经存在的文件名

---

#### 功能

改变在 USB 闪存中的当前有的文件名。

#### 参数

新的文件名

给文件设定一个新的名。为了指定包括文件在内的文件夹，在文件名前添加“文件夹名¥”。  
在文件名前写“CF¥”来指定在控制器中的袖珍闪存的文件。

已经存在的文件名

指定文件的当前名。为了指定包括文件在内的文件夹，在文件名前添加“文件夹名¥”。  
在文件名前写“CF¥”来指定在控制器中的袖珍闪存的文件。

#### 说明

如果新的文件名已经存在的话，USB\_RENAME 指令的结果将出现错误。

#### 示例

>USB\_RENAME file\_new = file  把USB闪存中的文件名由“file”改变为  
“file\_new”。

---

**XFER 目标程序名, 步骤编号 1=源程序名, 步骤编号 2, 步骤数量**

---

### 功能

从一个程序复制和传送步骤到另一个程序。

### 参数

#### 目标程序名

设定接收复制数据的程序。如果该名称的程序不存在，数据被传送到以该名称命名的新程序中。

#### 步骤编号 1

设定步骤编号，复制的数据将被插入该步数编号之前。如果步骤编号未指定，数据将被插入在指定程序的末尾。

#### 源程序名

设定程序的名称，数据将从此程序复制。

#### 步骤编号 2

设定源程序中开始复制数据处的步骤编号。如果未指定，数据将从源程序的开头复制。

#### 步骤数量

设定从源程序复制的步骤数量，从上面设定的步骤编号处开始(参数：步骤编号 2)。如果不指定，将复制程序中所有的剩余步骤。

### 说明

从指定程序中指定步骤编号处复制指定的步骤数量，然后将数据插入到目标程序的指定步骤之前。

#### [ 注 意 ]

如果目标程序正在被 STATUS 或 PCSTATUS 指令显示，或者正在被编辑(EDIT 指令)，那么就不能使用 XFER 指令。



---

**COPY 新程序名=源程序名+源程序名+**

---

### 功能

复制完整程序到一个新程序。

### 参数

#### 新程序名

指定放置复制的程序的程序名称。此程序名称必须指定。

#### 源程序名

指定要复制的程序名。必须指定至少一个程序。

### 说明

当指定了两个或更多的程序时，这些程序结合成一个以新程序名命名的程序。指定的新文件名不能是一个已存在的程序。

---

**USB\_COPY 新的文件名=源文件名**

---

### 功能

复制指定的文件到 USB 闪存中的新文件。

### 参数

#### 新的文件名

给新创建的文件指定一个新名。此名必须是被指定的。为了指定包括文件在内的文件夹，在文件名前添加“文件夹名¥”。在文件名前写“CF¥”来指定在控制器中的袖珍闪存的文件。

#### 源文件名

指定将要被复制的文件名。此名必须是被指定的。为了指定包括文件在内的文件夹，在文件名前添加“文件夹名¥”。在文件名前写“CF¥”来指定在控制器中的袖珍闪存的文件。

### 说明

新文件的指定名不能是已经存在的文件名。

---

**TRACE** **stepper 编号:** ON/OFF

---

### 功能

开始（ON）或结束（OFF）机器人的记录或 PC 程序，允许程序跟踪。

### 参数

Stepper (步骤器) 编号

用数字指定要跟踪的程序类型。

1: 机器人程序

1001: PC 程序 1

1004: PC 程序 4

1002: PC 程序 2

1005: PC 程序 5

1003: PC 程序 3

如果不指定类型，所有程序被跟踪记录。

ON/OFF

开始/结束记录。

### 说明

如果在 TRACE ON 前，未用 SETTRACE 指令保留必要的存储器，将报出错（P2034）“未定义的内存”。

---

SETTRACE	步骤数量
----------	------

---

### 功能

保留必要的内存来记录程序跟踪的数据。

### 参数

#### 步骤数量

指定记录的步骤数(设置范围：1 到 9999)。如果不指定，保留 100 步用的存储器。

### 说明

一部分用户存储器被留出，提供给已存在机器人和 PC 程序的当前编号和指定的步骤数量。

如果未保留记录存储器，就执行 TRACE ON 和 LSTRACE 指令，将报出错(P2034)“未定义的内存”。如果在记录过程中，使用了 SETTRACE 指令，报出错(P2033)“日志记录运行中”，并且全部跟踪被关闭(结束)。

---

## RESTRACE

---

### 功能

释放大用 SETTRACE 指令保留的存储器。

### 说明

如果在记录过程中使用 RESTRACE 指令，报出错(P2033)“日志记录运行中”。

LSTRACE

stepper 编号: 日志编号

**功能**  
显示指定机器人程序或 PC 程序的日志记录数据。

**参数**  
stepper(步骤器) 编号  
通过数字选择，指定要显示程序的类型：

- 1: 机器人程序
- 1001: PC 程序 1

1002: PC 程序 2

1003: PC 程序 3
- 1004: PC 程序 4

1005: PC 程序 5
- 如果未指定类型，显示机器人程序的日志记录。

**日志编号**  
指定要开始显示的日志数据的行编号。如果未指定，选择第一行。

**说明**  
如果在执行 LSTRACE 前，未用 SETTRACE 指令保留必要的存储器，将报出错（P2034）“未定义的内存”）。

如果在记录过程中，使用了 LSTRACE 指令，将报出错（P2033）“日志记录运行中”。

执行 LSTRACE 指令后，日志数据显示。在数据的后面出现提示符，并且可以输入如下指令：

N  显示后面 9 行。

L  显示前面 9 行。

S **编号** ; 显示指定的日志行编号，以及该行前后各 4 行日志(共 9 行)。如果未指定行号，显示 1-9 行。如果输入的编号比存在的行数要大，显示最高的行编号。

F **字符** ; 显示包含指定字符的那一行，以及该行前后各 4 行日志(共 9 条)。如果未指定字母，使用以前和 F 指令输入的字母。如果字符在数据中未被找到，不显示。

E  (END, 结束)；结束显示返回 AS 监控模式。  
 输入后显示下面 9 行。

示例

```
91    pg1    31 JOINT SPEED9 ACCU1 TIMER0 TOOL1 WORK0 CLAMP1 (OFF, 0, 0, 0) 2
92    pg1    32 SIGNAL 14;sig on
93    pg1    33 JOINT SPEED9 ACCU1 TIMER0 TOOL1 WORK0 CLAMP1 (OFF, 0, 0, 0) 2
94    pg1    34 CALL "sub1"
95    sub1    1 PRINT "SUB1"
96    sub1    2 xyz:
97    sub1    3 JMOVE a
98    sub1    4 JMOVE b
99    sub1    5 JMOVE c
```

----N: 下一页, L: 前一页, S 编号: 跳到编号, F 文字: 文字检索, E: 退出----

### 5.3 程序和数据存储指令

SAVE	保存指定的程序。
SAVE/P *	保存程序。
SAVE/L *	保存位姿变量。
SAVE/R *	保存实型变量。
SAVE/S *	保存字符串变量。
SAVE/A *	保存辅助信息。
SAVE/SYS *	保存系统数据。
SAVE/ROB *	保存机器人数据。
SAVE/ELOG *	保存错误日志数据。
LOAD *	装载程序和数据到机器人存储器。
USB_MKDIR	在 USB 闪存中创建新的文件夹。

**注释\*** 这些指令保存数据到个人计算机中。保存数据到 USB 闪存中，请给这些指令添加前缀 USB\_。更多详情，请见各指令的说明。

SAVE/SEL	文件名=程序名, .....
USB_SAVE/SEL	文件名=程序名, .....

**功能**  
SAVE 指令把程序和变量数据存储到计算机硬件上。（仅当 PC 与机器人控制器相连时才可使用）。  
USB\_SAVE 指令把程序和变量数据存储到 USB 闪存中。


**参数**  
**文件名**  
将指定的程序保存到此文件名下。如果未指定扩展名,扩展名“. as”将被自动加到文件名上。为了指定包括文件在内的文件夹,在文件名前添加“文件夹名¥”。在文件名前写“CF¥”来指定在控制器中的袖珍闪存的文件。

**序名**  
选择要保存的程序。如果未指定,将保存存储器中的全部程序。

**说明**  
指令 SAVE/P、SAVE/L、SAVE/R、SAVE/S、SAVE/SYS 指令存储各数据类型（分别是程序、位姿变量、实型变量、系统数据）在各自的文件中。使用不带参数的 SAVE 指令可存储全部五种数据类型到一个文件中。

SAVE 指令（不带 /SEL）可以存储指定的程序,包括程序使用的全部变量和子程序。

SAVE/SEL 指令仅存储程序,不存储程序使用的子程序和变量。

**示例**  
          >SAVE f3=cycle,motor       将系统数据、两个程序“cycle”和“motor”、这两个程序调用的子程序和程序中使用的变量存储在“f3.as”文件名下。

[ 注 意 ]

如果指定的文件名已经存在于存储器中,已存在的文件被自动地以扩展名前带“b”的新文件名来重新命名。例如,如果“file1.as”已经存在于存储器中,如果执行了指令 >SAVE file1,那个老文件被重新命名为“file1.bas”,然后新创建的文件被命名为“file1.as”。

SAVE/P/SEL	文件名=程序名, .....
SAVE/L/SEL	文件名=程序名, .....
SAVE/R/SEL	文件名=程序名, .....
SAVE/S/SEL	文件名=程序名, .....
SAVE/A	文件名
SAVE/SYS	文件名
SAVE/ROB	文件名
SAVE/ELOG	文件名

---

USB_SAVE/P/SEL	文件名=程序名, .....
USB_SAVE/L/SEL	文件名=程序名, .....
USB_SAVE/R/SEL	文件名=程序名, .....
USB_SAVE/S/SEL	文件名=程序名, .....
USB_SAVE/A	文件名
USB_SAVE/SYS	文件名
USB_SAVE/ROB	文件名
USB_SAVE/ELOG	文件名

功能

把如下的内容存储在磁盘文件中：程序 (/P)、位姿变量 (/L)、实型变量 (/R)、字符串变量 (/S)、辅助信息 (/A)、系统数据 (/SYS)、机器人数据 (/ROB) 和错误日志 (/ELOG)。

如同 SAVE 指令，USB\_SAVE/ 指令用于保存文件到 USB 闪存中。（仅当 PC 连接到机器人控制器时，用 SAVE/ 指令。见 2.6.2 上传和下载数据）

参数

文件名

保存数据到此文件名下。如果不指定扩展名，下述扩展名按照文件的数据类型自动被加到文件名上。

程序	. PG	系统数据	. SY
姿态信息	. LC	机器人数据	. RB
实型变量	. RV	错误日志	. EL
字符串变量	. ST		
辅助信息	. AU		

程序名

选择要保存的程序名。如果不指定，存储器中的全部程序和数据将被保存在磁盘文件中。



## 说明

### 1. SAVE/P

将被选定的程序和那些程序调用的子程序（包括被子程序调用的子程序）存储在指定的磁盘文件中。

被保存到文件中的程序名将显示在系统终端上。可能会出现一些不是 SAVE 指令的指定程序名的附加程序名。这些是指定程序调用的子程序的名字。这些子程序将和程序一起被存储在同一文件中。

程序将按照字母顺序存储在文件中，与他们被保存的先后无关。

### 2. SAVE/L, SAVE/R, SAVE/S

仅存储指定程序及其调用的子程序使用的变量。（/L: 仅存储位姿变量, /R: 仅存储实型变量, /S: 仅存储字符串变量）

### 3. SAVE/A

存储辅助信息

### 4. SAVE/SYS

存储系统数据

### 5. SAVE/ROB

存储与机器人相关的数据（机器人数据）


### 6. SAVE/ELOG

存储错误日志。本指令不能与其他的 SAVE/ 指令一起输入。例如，SAVE/ELOG/R 不起作用。

7. 如果/SEL 与/P, /L, /R, /S, 一起被输入，那么仅存储主程序和主程序所使用的变量。不存储子程序和子程序所使用的变量。

如果指定的文件名已经存在于存储器中，已存在的文件被自动地以扩展名前带“b”的新文件名来重新命名。（详见 SAVE 指令的[ 注意 ]框）

## 示例

>SAVE/L file2=pg1, pg2 

用在程序 pg1 和 pg2 中的位姿变量被存储在文件名“file2.lc”下。

---

## LOAD/Q 文件名

---

---

## USB\_LOAD/Q 文件名

---

### 功能

LOAD 指令装载计算机存储器中的文件到机器人存储器中。（仅当 PC 与机器人连接时使用）  
USB\_LOAD 指令装载文件到 USB 闪存中。

### 参数

文件名

保存此文件名下的指定程序。如果不指定扩展名，扩展名 “.as” 被自动添加给文件名。为了指定包括文件在内的文件夹，在文件名前添加 “文件夹名¥”。在文件名前写 “CF¥” 来指定在控制器中的袖珍闪存的文件。

### 说明

本指令从指定文件中装载数据（系统数据、程序和变量）到机器人存储器中。试图装载已经存在于存储器中的程序名将导致出错，以及 LOAD 指令执行中止。

#### [ 注 意 ]

当装载已经存在于存储器中的位姿变量、实型变量、或字符串变量时，存储器中的数据将被覆盖而没有任何警告。（程序不被覆盖）

当正在覆盖存储器数据时，如果 LOAD 指令被取消，原来的数据将被删除。

对于具有/Q 的 LOAD 指令，以下信息将在各系统数据或程序装载前出现：


转载吗？（1: 是, 0: 不/ 2: 全部转载, 3: 退出）

选择如下：

- 1: 装载数据
- 0: 不装载此数据并转向下一个数据
- 2: 装载此数据，并不加询问地装载全部文件中剩余的数据。
- 3: 不装载此数据，并结束 LOAD 指令。

如果程序中有不可读或不正确的步骤时，将出现下述信息：“步骤格式不正确（0: 继续装载 1: 删除程序并退出）”。如果输入 “0” 继续了此操作，可早程序装载完成后，用编辑器修正这些步骤。

示例

>LOAD pallet 

装载文件“pallet.as”中的数据到存储器中。

转载中

系统数据

程序 a1()

程序 test()

:


变换值

各轴值

实数变量

文件转载完毕。

>

>LOAD f3.pg 

装载文件“f3.pg”中的全部程序到存储器中。

---

## USB\_MKDIR 文件夹名

---

### 功能

以指定名在 USB 闪存中创建一个文件夹。

### 参数

文件夹名

以此名创建文件夹。

### 说明

用指定名在 USB 闪存中创建一个新的文件夹。如果在内存中同名的文件或文件夹已经存在，则显示信息“不能创建文件夹”，文件夹不能被创建。

#### 5.4 程序控制指令

SPEED	设定监控速度。
PRIME	准备执行的程序。
EXECUTE	执行程序。
STEP	执行程序的一个步骤。
MSTEP	执行一条机器人运动命令。
ABORT	当前步骤完成后停止执行。
HOLD	停止执行
CONTINUE	继续程序的执行。
STPNEXT	用单步模式执行程序。
KILL	初始化执行堆栈。
DO	执行单条程序命令。

**SPEED 监控速度**

**功能**

用百分比设定监控速度。

**参数**

**监控速度**

设定监控速度的百分比。如果设定值是 100，那么监控速度将是最大速度的 100 %。如果设定值是 50，那么监控速度将是最大速度的一半。

**说明**

监控速度(用此指令设定)和程序速度(在程序中用 SPEED 命令设定)的乘积决定机器人运动速度。例如，如果监控速度设定为 50、且程序中设定的速度为 60，那么机器人的最大速度是 30 %。

**[ 注 意 ]**





如果监控速度和程序速度的乘积超过 100，那么机器人的最大速度自动设定为 100%。

监控速度的默认设定是 10 %。

本指令不会影响正在执行的运动的速度。新的速度设定会在当前运动和已计划的运动完成后生效。

**示例**

如果程序速度设定为 100%:

>SPEED	30		机器人运动速度被设定为最大速度的 30 %。
>SPEED	50		机器人运动速度被设定为最大速度的 50 %。
>SPEED	100		机器人运动速度被设定为最大速度的 100 %。
>SPEED	200		机器人运动速度被设定为最大速度的 200 %。

---

PRIME	程序名	执行循环数	步骤编号
-------	-----	-------	------

---

### 功能

准备系统，让程序可以使用  $\boxed{A} + \boxed{\text{CYCLE START}}$  (循环启动) 按钮执行。单独使用此指令不会执行程序。

### 参数

#### 程序名

选择希望准备执行的程序。如不指定，则将选中上一次执行过或 prime 指令使用过的的程序。

#### 执行循环数

设置程序执行的次数。如果未指定，则假定为 1。若希望连续执行程序，则输入负数(-1)。

#### 步骤编号

选择步骤编号，程序将从此步骤开始执行。如果未指定，执行将从程序的第一步开始。

### 说明

本指令仅为程序的执行准备系统，而不执行程序。使用 PRIME 指令准备好系统后，可以用 CONTINUE 指令或者用  $\boxed{A} + \boxed{\text{CYCLE START}}$  (循环启动) 按钮来执行程序。

### [ 注 意 ]

使用本指令时，机器人存储器中的执行堆栈被初始化；也就是说任何暂停（例如，通过 HOLD 指令或因为出错）指示的程序信息将被丢失。例如，如果程序在执行其子程序时被暂停（信息被存储在堆栈中），随后用 PRIME 指令和 CYCLE START (循环启动) 按钮或 CONTINUE 指令（堆栈被初始化）来执行子程序，那么由于堆栈已被初始化，程序运行将无法返回主程序。

EXECUTE

程序名, 执行循环数, 步骤编号


**功能**  
执行一个机器人程序。

**参数**  
程序名  
选择希望执行的程序。如果未指定，则将选择上次所执行的(通过 EXECUTE, PRIME, STEP 或 MSTEP 指令)程序。

执行循环数  
设定执行程序次数。如果未指定，假定为 1。希望连续执行程序，则输入负数(-1)。最大限度为 32767。

步骤编号  
选择步骤编号，程序将从此步骤开始执行。如果未指定，执行将从程序的第一步开始。  
选择开始执行的步骤。如果未指定，则从程序的第一步开始执行。如果程序执行超过一次循环，那么从第 2 次循环开始，程序执行将从第一步开始。

**说明**  
从指定的步骤开始执行一个指定的机器人程序。并且将重复执行指定循环次数。

 小 心

在使用本条指令时，下列条件被自动设置：

SPEED 100 ALWAYS

ACCURACY 1 ALWAYS

STOP 命令或程序的最后一步标志着一次循环的结束。

**示例**

>EXECUTE test,-1

连续执行名为“test”的程序。(程序连续执行直至被 HALT 等指令停止，或错误发生时停止。)

>EXECUTE

执行上次执行的程序(仅一个循环)。



STEP	程序名, 执行循环数, 步骤编号
MSTEP	程序名, 执行循环数, 步骤编号

功能

执行机器人程序的一个步骤。

参数

程序名

选择希望执行的程序。如果未指定，则将选择当前被暂停的程序或上次执行的程序。

执行循环数

设定执行程序次数。如果未指定，假定为 1。

步骤编号

选择从那里开始执行的步骤编号。如果未指定，则从程序的第一步开始执行。如果未指定参数，则选择上次所执行步骤的下一步。


说明

此指令若无参数，仅可在以下情况时执行：

- 1. PAUSE 命令后，
- 2. 程序因非出错的原因停止时，
- 3. 当以前用 STEP 指令执行过程序命令后。

MSTEP 指令执行一个运动段(也就是说一个运动命令以及下一个运动命令前的步骤)。STEP 指令仅执行一个步骤的程序(机器人不一定运动)。

示例

>STEP assembly, , 23 

仅执行程序“assembly”的步骤 23。如在此执行后立即输入不带参数的 STEP 指令，则会执行程序“assembly”的步骤 24。

---

## ABORT

---

### 功能

停止机器人程序的执行。

### 说明

在当前步骤执行完成后停止机器人程序的执行。如果机器人在运动中，则机器人在完成当前进行的运动后停止。用 CONTINUE 指令继续(恢复)程序的执行。

—— [ 注 意 ] ——

在 AS 系统中，机器人的运动和执行中的步骤不一定始终相同。因此，如果步骤处理比机器人运动快，机器人可能在当前运动之后、停止之前，多执行一个运动。

---

## HOLD

---

### 功能

立即停止机器人程序的执行。

### 说明

机器人运动被立即停止。不像紧急停止按钮，马达电源不会关断(OFF)。本指令与把 HOLD/RUN(保持/运转)的状态由 RUN(运转)改变为 HOLD(保持)具有相同的效果。用 CONTINUE 指令可继续程序执行。

---

CONTINUE **NEXT**

---

### 功能

继续(恢复)执行由因 PAUSE 命令、ABORT 或 HOLD 指令、或因出错而停止的程序。本指令也可用来启动由 PRIME、STEP 或 MSTEP 指令做好执行准备的程序。

### 参数

#### NEXT

如果不输入 NEXT，程序将从执行停止的步骤继续执行。如果输入，则从停止步骤的下一步骤继续执行。

### 说明

关键字 NEXT 对程序重启的影响，将随程序被停止的情况不同而有所不同。

1. 程序在一个步骤或一个运动的执行过程中停止：  
CONTINUE 重新启动程序并重新执行被中断的步骤。  
CONTINUE NEXT 在程序停止的步骤的下一个步骤处重新开始执行。
2. 程序执行在一个步骤或一个运动完成后停止：  
CONTINUE 和 CONTINUE NEXT 从所完成的步骤紧接的下一个步骤处，重新执行程序，不考虑 NEXT。
3. 程序因 WAIT、SWAIT 或 TWAIT 命令而暂停时：  
CONTINUE NEXT 指令跳过上面的命令，从下一步骤开始继续执行。

---

[ 注 意 ]

---

CONTINUE 指令在以下条件下不能继续程序执行：

- 程序正常结束
- 程序被 HALT 命令停止
- 使用了 KILL 指令

---

## STPNEXT

---

### 功能

当系统开关 STP\_ONCE 为 ON 时，用于执行下一步骤。

### 说明

当系统开关 STP\_ONCE 为 ON 时，程序将以单步的增量来执行。本指令推进执行到程序中的下一个步骤。

---

## KILL

---

### 功能

初始化机器人程序的堆栈。

### 说明

如果程序因 PAUSE 命令、ABORT 指令或出错而停止，程序堆栈将保持当前的状态。KILL 指令用来初始化堆栈。一旦使用了 KILL 指令，CONTINUE 指令将是无效的，因为在堆栈中已经没有程序。

---

## D0 程序命令

---

### 功能

执行单条程序命令。(有些程序命令不能此指令执行)。

### 参数

程序命令

执行指定的程序命令。如果省略，则再次执行上次 D0 指令执行过的程序命令。

### 说明

程序命令一般被编写在程序内，并作为程序步骤来执行。然而 D0 指令可以执行单条的命令，而不必创建一个程序来运行该命令。

### 示例

>D0 JMOVE safe  机器人以关节插补动作移动到位姿“safe”。

>D0 HOME  机器人以关节插补动作移动到原点位姿(HOME 位姿)。

## 5.5 位姿信息指令

HERE	以当前位姿指定一个位姿变量。
POINT	定义一个位姿变量。
POINT/X	设定一个变换值变量的 X 值。
POINT/Y	设定一个变换值变量的 Y 值。
POINT/Z	设定一个变换值变量的 Z 值。
POINT/OAT	设定一个变换值变量的 OAT 值。
POINT/O	设定一个变换值变量的 O 值。
POINT/A	设定一个变换值变量的 A 值。
POINT/T	设定一个变换值变量的 T 值。
POINT/7	设定一个变换值变量的第 7 轴值。

---

## HERE 位姿变量

---

### 功能

用当前的位姿，来指定一个位姿变量名。位姿可以用变换值，关节位移值或复合变换值的形式表达。

### 参数

位姿变量


变量值可以指定为变换值、关节位移值或复合变换值的形式。

### 说明

位姿可以用变换值，关节位移值或复合变换值的形式表达。

#### [ 注 意 ]

仅定义复合变换值中最右边的变量。（见下面的示例）。如果复合值的其他变量未定义过，本指令会导致出错。


变量的值显示在终端显示的“变化吗?”。此值可以通过输入逗号分隔的数值进行修改。不用修改的值可以跳过。在信息“变化吗?”后按  键，结束数值编辑。

如果变量以关节位移值的形式定义(变量名以#开始)，则显示当前位姿的关节值。如果变量是变换值，则显示 XYZOAT 值。XYZ 值描述工具坐标系原点相对于基础坐标系的位置。OAT 值描述工具坐标系的姿态。


### 示例

>HERE #pick 

定义机器人的当前位姿为“#pick”（关节位移值）

>HERE place 

定义机器人的当前位姿为“place”（变换值）

HERE plate+object 

定义机器人的当前位姿为“object”，此位姿变量相对于位姿“plate”（复合变换值）。如果“plate”未定义则出错。

---

**POINT 位姿变量 1=位姿变量 2, 关节位移值变量**

---

**功能**

将“=”右边的位姿变量 2 赋值给“=”左边的位姿变量 1。

**参数**

位姿变量 1

指定要定义的位姿变量(关节位移值变量、变换值变量或复合变换值变量)。

位姿变量 2

指定已知的位姿变量的值。如果不指定，符号“=”也应省略。


关节位移值变量

指定用关节位移值形式定义的变量。如果左边的位姿变量是关节位移值，并且位姿变量是变换值时，则设定此参数（如果左边的参数不是关节位移值，则不能设定此参数。）在此指定的关节位移值表示机器人在位姿的形态。如果不指定，则以当前机器人的形态来定义位姿变量。

**说明**

设定使位姿变量 1 的值与位姿变量 2 的值相等。给变量设定新的数值，将显示其新的数值。

不指定位姿变量 2 而执行此指令时，如果位姿变量 1 已经被定义了，将显示其变量的值。如果位姿变量 1 还没被定义，则显示的值为全成分零（0，0，0，0，0，0）。

在显示的值的后面出现信息“变化吗？”。对此，如果输入新的数值，数值将被修改。信息“变化吗？”，到修改结束为止（不输入数值仅按  键），反复显示。

如果位姿变量 1 的值是关节位移值，则显示表示各轴的角度的数值。如果位姿变量 1 的值是变换值，则显示 XYZOAT 值。XYZ 值描述工具坐标系原点相对于基础坐标系的位置。

OAT 值描述工具坐标系的姿态。如果位姿变量 1 是复合变换值时，则决定其复合变换值变量的右端的变换值变量的值。其复合变换值变量中右端以外的变换值变量的值如果还未被定义的话，将出现错误。



[ 注 意 ]

当”=” 左边和右边的数值类型不同时，此指令的作用如下：

- POINT 变换值变量=关节位移值变量  
将右边的关节位移变量值转变成变换值并赋值给左边的位姿变量 1。
- POINT 关节位移值变量=变换值变量， 关节位移值变量  
右边的变换值变量值被转换成关节位移值并赋值给左边的位姿变量 1。如果指定了右边的关节位移值变量，则以关节位移值变量的关节位移值的形态来求位姿变量 1 的关节位移值。如果未指定，则以机器人当前的形态来求关节位移值变量值。

当指定数值时，最多可以输入 9 位小数位的数字。超过小数为 9 位数字的精度无法保证。

示例

>POINT #park

显示关节位移值变量 “#park” 的值。  
(如果未定义，则显示 0, 0, 0, 0, 0, 0)

JT1	JT2	JT3	JT4	JT5	JT6
10.000	15.000	20.000	30.000	50.000	40.000

变化吗?(放弃请按 RETURN 键)  
...-15

JT1	JT2	JT3	JT4	JT5	JT6
10.000	15.000	20.000	-15.000	50.000	40.000

变化吗?(放弃请按 RETURN 键) ☐

>POINT pick1=pick ☐

将 “pick” 的变换值赋值给 “pick1” 的变换值变量，并显示此值以供修改。

>POINT pos0=#pos0 ☐

将关节位移值变量 “#pos0” 转换成变换值并将它赋值给变量 “pos0”。

>POINT #pos1=pos1, #pos2 ☐

用#pos2 给定的机器人形态，将变换值的变量 “pos1” 转换为关节位移值，并将它赋值给变量 “#pos1”。

POINT/X	变换值变量 1=变换值变量 2
POINT/Y	变换值变量 1=变换值变量 2
POINT/Z	变换值变量 1=变换值变量 2
POINT/OAT	变换值变量 1=变换值变量 2
POINT/O	变换值变量 1=变换值变量 2
POINT/A	变换值变量 1=变换值变量 2
POINT/T	变换值变量 1=变换值变量 2
POINT/7	变换值变量 1=变换值变量 2

功能

将变换值变量 2 的变换值分量赋值给与变换值变量 1 的变换值相应的分量。这些值将显示在终端上以供编辑。

参数

变换值变量 1

选择要定义的变换值变量（变换值变量或复合变换值变量）

变换值变量 2

如果不指定，则“=”也应省略。

说明

仅赋值变换值的变量指定分量(X, Y, Z, O, A, T)。一旦命名执行，将显示每个分量的值以及信息“变化吗?”和提示。这时，可以编辑这些值。如果在提示后仅按 $\square$  键，退出。

示例

下述指令将 a1 的 OAT 赋值给 a2。a1 和 a2 的变换值如下：

	X	Y	Z	O	A	T
变换值变量 a1=	(1000,	2000,	3000,	10,	15,	30)
a2=	未定义					

```

>POINT/OAT a2=a1
X[mm]   Y[mm]   Z[mm]   O[deg]   A[deg]   T[deg]
  0.      0.      0.      10.      15.      30.
变化吗? (放弃请按 RETURN 键)  $\square$ 

```

## 5.6 系统控制指令

STATUS	显示系统状态。
WHERE	显示目前机器人的当前位姿数据。
IO	显示二进制信号的状态。
FREE	显示空余存储器的数量。
TIME	显示和设置当前时间和日期。
ULIMIT	设定机器人运动的上限。
LLIMIT	设定机器人运动的下限。
BASE	改变基础坐标系变换值。
TOOL	定义工具变换值。
SET_TOOLSHAPE	设定工具形状的数据。
ENA_TOOLSHAPE	根据工具形状设定速度控制的有效/无效。
TOOLSHAPE	根据工具形状设定速度控制的数据。
SETHOME	设定原点位置。
SET2HOME	设定原点位置 2。
ERRLOG	显示出错情况的历史。
OPLOG	显示操作的历史。
SWITCH	显示系统开关设置。
ON	开启系统开关(变为 ON )。
OFF	关闭系统开关(变为 OFF)。
ZSIGSPEC	设定和显示 I/O 信号的总数。
HSETCLAMP	设定默认的夹紧规格。
DEFSIG	显示和设定软件专用信号。
ZZERO	显示或设定调零数据。
ERESET	复位出错状态。
SYSINIT	初始化整个系统。
HELP	显示 AS 语言命令/指令列表。
ID	显示软件的版本信息。
WEIGHT	设定负荷重量数据。

ENCCHK_EMG	设定紧急停止时机器人的位姿与机器人重启时的位姿之间的偏差检查的容许范围。
ENCCHK_PON	设定电源开启时编码器值与上次电源关闭时值之间差异的可接受范围。
SLOW_REPEAT	设定低速再现模式中的速度。
REC_ACCEPT	允许/禁止程序的记录和/或更改。
ENV_DATA	设定自动伺服关断计时和示教器连接/断开。
ENV_2DATA	设定终端连接/断开。
CHSUM	清除校验和错误。
TPLIGHT	开启示教器的背光灯。
IPEAKLOG	显示峰值电流值。(选项)
IPEAKCLR	复位峰值电流值。(选项)
OPEINFO	显示操作信息。
OPEINFOCLR	清除操作信息。
REFFLTSET_STATUS	显示指令值的平均化时间值。
FFSET_STATUS	设定机器人速度/加速度前馈增益。

STATUS

功能

显示系统和当前机器人程序的状态。

说明

系统和机器人程序状态以下列格式显示：

1. ...	机器人状态 再现模式			
2. ...	环境设定状态 监控速度 (%) = 10.0 ALWAYS 程序速度 (%)= 100.0 ALWAYS 精度 [mm] = 1.0			
3. ...	Stepper 状态 程序正在运行			
4. ...	程序运行次数 已运行完成次数: 3 剩余运行次数: 无限次			
5. ...	程序名	优先级	步骤号	
	Test	0	1	WAIT sig(1001)

1. 机器人状态

当前机器人状态是下述情况之一：  
出错中；出现了一个错误；请尝试错误复位操作。  
马达电源 OFF；马达电源处于 OFF 状态。  
示教模式；马达电源为 ON；机器人由示教器控制。  
再现模式；马达电源为 ON；机器人由机器人程序控制。  
再现模式循环开始 ON；马达电源为 ON；机器人程序运行中。  
程序等待中；马达电源为 ON；机器人程序正在运行并处于等待状态 (执行一个 WAIT、SWAIT 或 TWAIT 命令)

2. 环境设定状态

当前监控速度 (用百分比显示)

3. Stepper 状态

步骤执行的当前状态。

4. 程序运行次数

已运行完成次数：        完成循环；已经完成的执行循环 (0-32767)。  
剩余运行次数：        剩余循环；剩余的执行循环。如果在 EXECUTE 指令中为执行循环数指定了负数，则显示“无限次”。

5. 程序名

当前正在执行或处于等待状态的程序名或步骤。

WHERE


显示模式

功能

显示当前的机器人位姿。


参数

显示模式

选择显示数据的模式。共有下面所示的 16 种模式。(模式 7-16 是选项)。如果不指定模式，则显示 TCP 在基础坐标系中的变换值和各关节角度(JT1, JT2, ..., JT3)。显示模式保持不变直到再次按下。

- WHERE
- ..... 显示基础坐标系中当前机器人位姿的变换值和关节角度(JT1, JT2, ..., JT3)。
- WHERE 1
- ..... 显示当前位姿的关节角度。
- WHERE 2
- ..... 显示基础坐标系中当前位姿的 XYZOAT 值(mm, deg)。
- WHERE 3
- ..... 显示当前命令值(deg)。
- WHERE 4
- ..... 显示与命令值的偏差(bit)。
- WHERE 5
- ..... 显示各关节的编码器值(bit)
- WHERE 6
- ..... 显示各关节的速度(deg/s)
- WHERE 7
- ..... 显示包括外部轴的当前位姿。(选项)
- WHERE 8
- ..... 显示固定工件坐标系中的当前位姿。(选项)
- WHERE 9
- ..... 显示变换值的各关节命令值
- WHERE 10
- ..... 显示马达电流。
- WHERE 11
- ..... 显示马达速度。
- WHERE 12
- ..... 显示用另一机器人的基础坐标系表达的当前变换值。(选项)
- WHERE 13
- ..... 示用另一机器人的工具坐标系表达的当前变换值。(选项)
- WHERE 14
- ..... 显示马达电流的命令值。
- WHERE 15
- ..... 示编码器的本来数据。
- WHERE 16
- ..... 显示 TCP 的速度。

示例

>WHERE 					
JT1	JT2	JT3	JT4	JT5	JT6
9.999	0.000	0.000	0.000	0.000	0.000
X[mm]	Y[mm]	Z[mm]	0[deg]	A[deg]	T[deg]
15.627	88.633	930.000	-9.999	0.000	0.000

I0/E	信号编号
------	------

**功能**  
显示所有外部和内部 I/O 信号的当前状态。

**参数**  
信号编号  
1.....显示 1－32, 1001－1032, 2001－2032  
2.....显示 33－64, 1033－1064, 2033－2064  
3.....显示 65－96, 1065－1096, 2065－2096  
4.....显示 97－128, 1097－1128, 2097－2128  
如果未指定, 则显示 1－32, 1001－1032, 2001－2032


**说明**  
如果系统开关 DISPIO\_01 为 OFF, 用“o”显示信号 ON 状态的信号, 用“x”显示 OFF 状态的信号。专用信号用大写字母(“O”和“X”)显示。如果系统开关 DISPIO\_01 为 ON, 用“1”显示 ON 状态的信号, 用“0”显示 OFF 状态的信号。“-”显示未安装的外部 I/O 信号。

如果在指令中输入了“/E”, 信号编号 3001 及其以上的, 将和 1－, 1001－, 2001－编号的信号一同显示。(选项)


显示持续刷新, 直到用[STOP]键结束显示。  
(见 7.0, DISPIO\_01 系统开关)

**例如**  
当 DISPIO\_01 为 OFF 时

```
>I0 [STOP]
  32 — 1      xxxx  xxxx  xxxx  xxxx  xxxx  xxxx  xxxx  xxxo  xxxo
1032 — 1001  xxxx  xxxx  xxxx  xxxx  xxxx  xxxx  xxxx  Oxxx
2032 — 2001  xxxx  xxxx  xxxx  xxxx  xxxx  xxxx  xxxx  xxxx
>
```

```
>IO/E 
  32 —    1    xxxx    xxxx    xxxx    xxXX    xxxx    XXXX    XXX0    XXX0
1032 — 1001    xxxx    xxxx    xxxx    xxXX    xxxx    XXXX    XXX0    XXX0
2032 — 2001    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx
3032 — 3001    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx    xxxx
>
```

当 DISPIO\_01 为 ON 时

```
>IO 
  32 —    1    0000    0000    0000    0000    0000    0000    0001    0001
1032 — 1001    0000    0000    0000    0000    0000    0000    0000    1000
2032 — 2001    0000    0000    0000    0000    0000    0000    0000    0000
>
```

---


FREE

---

功能

用百分比和字节数，显示当前未被使用的存储器大小。

示例

```
>FREE 
全部存储容量8192 kbytes
可使用存储容量8191 kbytes (99%)
```



---

**TIME** 年-月-日 小时: 分钟: 秒

---

### 功能

设定和显示当前的时间和日期。

### 参数

年-月-日 小时:分钟:秒


用以下描述的格式设定时间和日期。当设置“小时:分钟:秒”时，参数“年一月一日”不能省略。显示设定值后面跟有信息“变化吗?”。当省略了所有参数时，显示当前的时间和日期。

### 说明


本指令设定机器人内部的日历。各元素的范围是：

年        (00—99)  
月        (01—12)  
日        (01—31)

小时     (0—23)  
分钟     (0—59)  
秒        (0—59)

显示当前时间或输入值以及后面的跟随信息“变化吗?”。欲改变日期则需输入新值。按下  键结束指令。

### 示例

```
>TIME 02-04-29 09:45:46 
```

```
>TIME
```

```
现在时间 02-04-29 09:47:33
```

```
变化吗? (放弃请按 RETURN 键)
```

```
02-05-17
```

```
现在时间 02-05-17 (星期三) 09:47:33
```

```
变化吗? (放弃请按 RETURN 键)
```

```
> 
```

ULIMIT 关节位移值变量

LLIMIT 关节位移值变量

功能


设定和显示机器人动作范围的上限/下限。

参数

关节位移值变量

用关节位移值设定软件限度（上限或下限）。如果参数未指定，显示当前值。

说明

如果指定了参数，画面上显示出指定位姿变量，之后显示出信息“变化吗?”在此信息后输入需要的值，像在POINT指令中做的那样。要结束此指令，按  键。

如果参数未指定，则显示当前设定的极限值，后面跟随信息“变化吗?”。


示例

>ULIMIT

显示当前的设置。

	JT1	JT2	JT3	JT4	JT5	JT6	
最大值	120.00	60.00	60.00	190.00	115.00	270.00	(最大允许极限值)
现在值	30.00	15.00	25.00	-40.00	60.00	15.00	(当前设置值)
变化吗? (放弃请按 RETURN 键)							


>110, 50

	JT1	JT2	JT3	JT4	JT5	JT6
最大值	120.00	60.00	60.00	190.00	115.00	270.00
现在值	110.00	50.00	25.00	-40.00	60.00	15.00
变化吗? (放弃请按 RETURN 键) 						

>ULIMIT #upper 

将软件上极限设定为以变量 “#upper” 定义的位姿。

:

>LLIMIT #low 

将软件下极限设定为以变量 “#low” 定义的位姿。

:

BASE

变换值变量

**功能**  
定义基础变换值，它指定了基础坐标系和空基础坐标系之间的位姿关系。

**参数**  
变换值变量  
为定义新的基础坐标系的变换值或复合变换值用位姿变量来指定。在此的位姿变量描述了相对于空基础坐标系的基础坐标系的位姿，以空基础坐标系中的值表达。如果没有任何说明，则显示当前的基础变换值。

**说明**  
如果为参数指定了“NULL”，那么基础变换值将被设定为“空基础”，(XYZOAT=0, 0, 0, 0, 0, 0,)。在系统被初始化时，基础变换值将被自动设定为空基础。

设置一个新的基础变换值后，将显示值(XYZOAT)和信息“变化吗?”。要改变数值，则请输入由逗号隔开的新数值，并按 $\boxed{\text{↵}}$ 键。如果未指定参数，则显示当前值。

当机器人向由变换值定义的位姿移动时或在基础模式中由手动操作时，系统将考虑这里定义的基础变换值，并自动计算出机器人位姿。

当使用位姿变量作为参数，而且位姿变量被重新定义时，请注意，基础转换必须也用 BASE 指令重新定义且使用新定义的位姿作为参数。位姿变量的改变将被反映到基础转换中。

BASE 指令不影响由关节位移值定义的位姿。

[ 注 意 ]

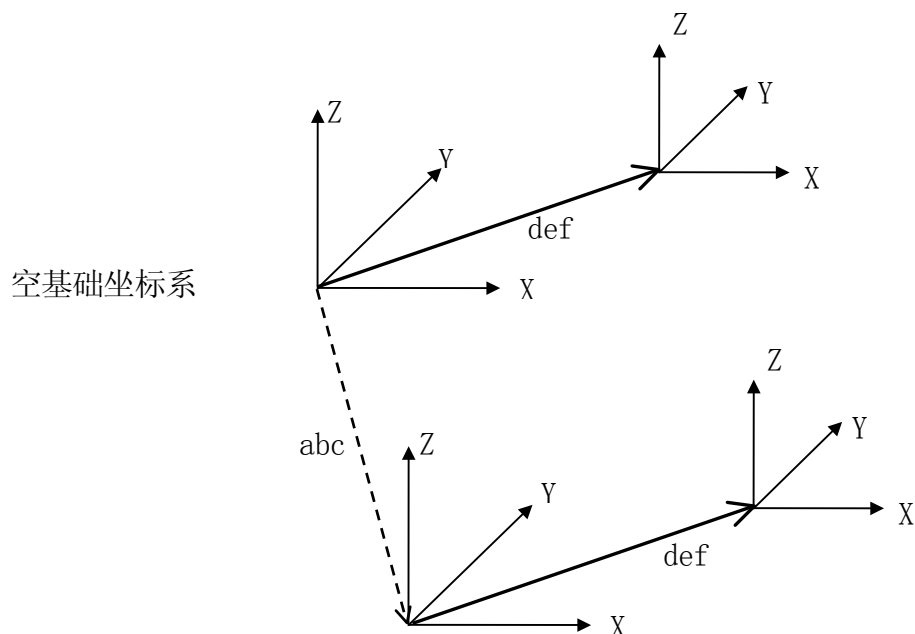
BASE   abc $\boxed{\text{↵}}$

DO   JMOVE   def $\boxed{\text{↵}}$

BASE   NULL $\boxed{\text{↵}}$

DO   JMOVE   def $\boxed{\text{↵}}$

即使上述两个例子中的位姿信息 def 是一样的，根据不同的基础转换，机器人的目标位姿也会有所区别。如下图所示。



### 示例

>BASE  显示当前基础变换值。  
 X[mm] Y[mm] Z[mm] O[deg] A[deg] T[deg]  
 -300. 0. 0. 0. 0. 0.  
 变化吗? (放弃请按 RETURN 键)

>BASE NULL  将基础变换值更改为空基础。  
 (X, Y, Z, O, A, T)=(0, 0, 0, 0, 0, 0)

>BASE abc  将基础坐标系位姿更改为由基础变换值变量 abc 描述的位姿。

---

## TOOL 变换值变量, 工具形状编号

---

### 功能

定义工具变换值。这个值指定了工具坐标系与空工具坐标系之间的位姿关系。

### 参数

#### 变换值变量


为定义新的工具坐标系的变换值或复合变换值用位姿变量来指定。此位姿变量描述了相对于空工具坐标系的工具坐标系的位姿，以空工具坐标系中的值表达。如果未输入值，则显示当前的工具变换值, 并且其能修改。

#### 工具形状编号

指定用于在示教和检查模式下速度控制的工具形状。如果未输入值，则显示当前的工具形状编号, 并且其能修改。

### 说明




如果“NULL”被指定给参数，则设定工具变换值为“空工具” (XYZOAT=0, 0, 0, 0, 0, 0,)。空工具坐标系的原点在工具安装法兰盘的中心，而且各坐标轴平行于机器人最后一个关节的轴。当系统被初始化时，工具变换值自动设置为空工具。

设定新的工具变换值后，则显示值 (XYZOAT) 和信息“变化吗?”。要改变值，输入由逗号隔开的新数值，并按  键。如果未指定参数，则显示当前值。

当机器人向由变换值定义的位姿移动时或在基础模式或工具模式下手动操作时，系统将考虑这里定义的工具变换值，并自动计算出机器人位姿。

当使用位姿变量作为参数时，如果位姿变量被重新定义，那么请注意：工具转换必须也用 TOOL 指令重新定义且将新定义的位姿作为参数。位姿变量的改变模式将被反映到工具转换中。见 11.4 工具变换值。

### 示例

- |  |  |
|--|--|
| >TOOL grip      | 将工具坐标系的位姿更改为位姿变量“grip”描述的位姿。                           |
| >TOOL NULL      | 将工具变换值更改为空工具。<br>(X, Y, Z, 0, A, T)=(0, 0, 0, 0, 0, 0) |
| >TOOL tool1, 1  | 选择 1 给工具变换值“tool1”的工具的工具形状编号。                          |

## SET\_TOOLSHAPE 工具形状编号=变换值变量 1, 变换值变量 2, ..., 变换值变量 8

### 功能

注册用于在示教模式和检查模式下速度控制的工具形状。

### 参数

工具形状编号

指定注册用的工具形状编号。设定范围：1 到 9。

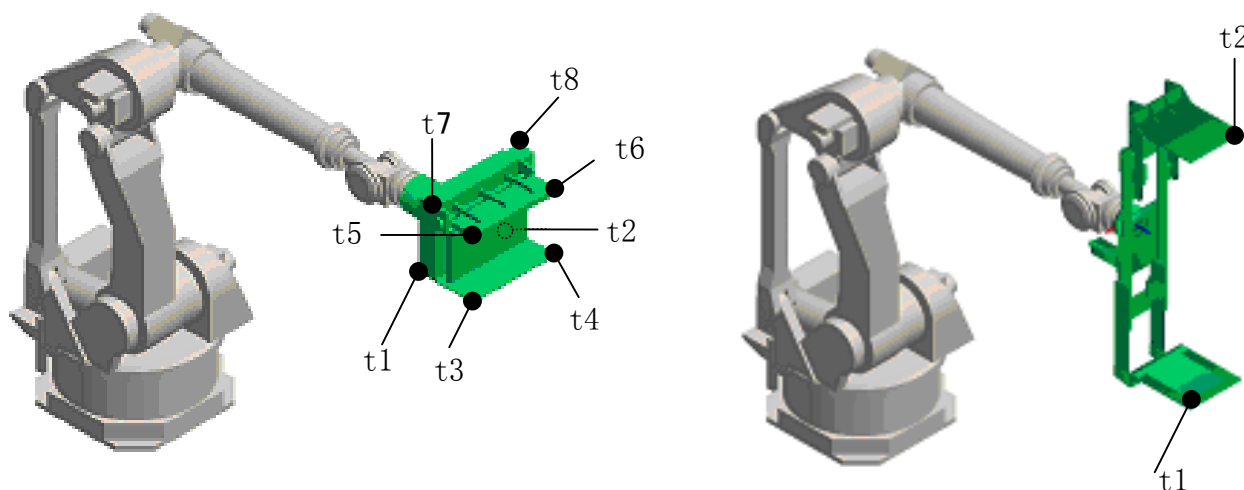
变换值变量 1-8

为工具形状指定变换值变量数。其最大位姿变量数可以指定为 8 个。从法兰盘中心看到的位置用变换值来指定。但只有 X, Y, Z 值用于工具形状注册。

### 说明

在示教模式或检查模式下，进行速度控制时，把可参照的工具形状最多用 8 个变换值变量来设定（见下图 t1 到 t8）。例如，从法兰盘面到顶端的距离比到 TCP 远时，或考虑工具上安装的制品的形状，想要控制速度时，请注册包括制品在内的工具的形状。

在辅助功能 304 中所注册的工具，当在同一辅助功能 304 的屏幕中，点击〈工具形状〉时，注册的工具形状通过屏幕显示。



### 示例

```
> SET_TOOLSHAPE 1=t1,t2, [Enter]
```

用以变换值变量 t1, t2 来指定与工具形状编号 1 对应的工具的端点位置。

```
> TOOL tool1,1 [Enter]
```

限制工具形状编号 1 的工具端点的速度。

---

## **ENA\_TOOLSHAPE 工具形状编号=TRUE/FALSE**

---

### **功能**

对指定工具形状编号的工具端点设定在示教模式和检查模式下速度控制的有效/无效。

### **参数**

工具形状编号

指定转换有效/无效的工具形状编号 1 到 9。

TRUE/FALSE

根据工具形状使速度控制有效时指定 TRUE，使其无效时指定 FALSE。

### **说明**

选择在指定工具形状编号的工具端点是否进行在示教和检查模式下的速度控制。所有工具形状的初始值都被设定为 FALSE（无效）。如果工具端点位置一个也没设定，而工具形状设定为 TRUE（有效）时，机器人在示教・检查模式下运转时，会产生错误 E1356 工具形状未设定。为了避免这种情况，用 SET\_TOOLSHAPE 指令时必须指定一个以上的工具端点位置或用本指令把设定由 TRUE（有效）变更为 FALSE（无效）后，用 TOOL 指令或 TOOLSHAPE 指令来设定适当的工具形状编号。（一旦，设定为 TRUE，如果不实行 TOOL/TOOLSHAPE 指令的话，此设定不会由 TRUE 变更为 FALSE。）

---

## TOOLSHAPE 工具形状编号

---

### 功能

选择用于在示教模式和检查模式下控制速度的工具形状。

### 参数



工具形状编号

给控制速度的工具形状指定编号。设定范围：1 到 9。

### 说明

根据工具形状进行在示教模式和检查模式下的速度控制时，必须要执行 ENA\_TOOLSHAPE 命令/指令（ENA\_TOOLSHAPE n=TRUE）。如果工具端点位置一个也没设定（所有点都设定为 0）的话，会产生错误 E1356 工具形状未设定。

### 示例

> TOOL tool1 	在指定工具 1 的工具变换值为 tool1 的同时，
> TOOLSHAPE 1 	对工具形状编号 1 的工具端点进行速度控制。



SETHOME   精度, HERE

SET2HOME   精度, HERE

功能

设定和显示原点位姿 (HOME 位姿)。

参数



精度

以毫米为单位设定原点位姿的精度范围。当机器人接近原点位姿到这里指定的距离时，就认为机器人处于原点位置了。如果未指定，默认值为 1mm。

HERE


设定当前的位姿为 HOME。

说明

如果未输入参数，则显示当前值以及信息“改变?”，输入需要的值，并按  键。如果不做任何改变，仅按 。

在 AS 系统中，可以设定两个原点位姿 (HOME1 和 HOME2)。用 SETHOME 指令设定 HOME 1，用 SET2HOME 指令设定 HOME 2。

示例

>SETHOME 2 


设定 2mm 的精度，通过输入新值改变 HOME 位置。

JT1	JT2	JT3	JT4	JT5	JT6	精度 [mm]
0.	0.	0.	0.	0.	0.	2.


变化吗? (放弃请按 RETURN 键)

, 90, -90

JT1	JT2	JT3	JT4	JT5	JT6	精度 [mm]
0.	90.	-90.	0.	0.	0.	2.

变化吗? (放弃请按 RETURN 键) 

>

>SETHOME   10, HERE 

将当前位姿设定为 HOME 位姿。精度被设定为 10mm；  
也就是说当机器人到达HOME位姿周围10mm范围内时，  
将输出表示机器人到达 HOME 位姿的 HOME 专用信号。

---

## ERRLOG

---

### 功能

显示错误日志(出错履历)。

### 说明

显示最近的 100 个错误。当显示到屏幕底端时，按`空格`键继续查看。错误按照年月日顺序列表。(辅助功能 0702)

### 示例

```
>ERRLOG 
```

```
1-[02/07/17 09:55:45 (SIGNAL 信号:00)  
(D1016)
```

---

## OPLOG

---

### 功能

显示操作日志(操作履历)。

### 说明

显示最近 100 个操作。当显示到达屏幕底端时，按`空格`键继续查看。(辅助功能 0703)

### 示例

```
>OPLOG 
```

```
1-[02/07/17 10:04:46] (SIGNAL 信号:00) [ PNL ]
```

SWITCH	开关名, ....., 开关名=ON
SWITCH	开关名, ....., 开关名=OFF

功能

显示和改变系统开关及其设置。

参数

开关名

显示指定的开关。如果未指定，则显示所有的开关。可以输入多个用逗号隔开的开关名。

ON 或 OFF

开(ON)或关(OFF)指定的系统开关。如果未输入参数，则显示开关的设置。

示例

```

>SWITCH                                显示所有的系统开关及其设置。
*POWER      ON      *REPEAT      ON
*RUN         ON      *CS          OFF
*RGSO        OFF     *ERROR        OFF
*TRIGGER      ON     *TEACH_LOCK  OFF
CHECK. HOLD   OFF     CP           ON
CYCLE. STOP   OFF     OX. PREOUT   ON
PREFETCH. SIGINS  OFF  QTOOL       OFF
REP_ONCE      OFF     RPS          OFF
STP_ONCE      OFF     AFTER. WAIT. TMR      ON
MESSAGES      ON      SCREEN       ON
AUTOSTART. PC  OFF     AUTOSTART2. PC OFF
AUTOSTART3. PC OFF     ERRSTART. PC  OFF
DISPIO_01     OFF     HOLD. STEP   OFF
FLOWRATE      OFF     SPOT_OP OFF
>[Enter]

>SWITCH  SCREEN, MESSAGE = OFF      关闭(OFF) SCREEN 和 MESSAGE。
SCREEN OFF
MESSAGE      OFF
>[Enter]

```

---

开关名, ..... ON

---

### 功能

开启 (ON) 指定的系统开关。

### 参数

开关名

开启 (ON) 这里指定的开关。可以输入多个用逗号隔开的开关名。

可用 SWITCH 指令检查开关的当前设置。

### 示例

>MESSAGES ON 

开启 (ON) MESSAGES 开关

>SCREEN, MESSAGES ON 

开启 (ON) MESSAGES 和 SCREEN 开关。

---

开关名, ..... OFF

---

### 功能

关闭 (OFF) 指定的系统开关。

### 参数

开关名

将这里指定的开关变成 OFF。可以输入多个用逗号隔开的开关名。

可用 SWITCH 指令检查开关的当前设置。

### 示例

>MESSAGES OFF 

关闭 (OFF) MESSAGES 开关。

>SCREEN, MESSAGES OFF 

关闭 (OFF) MESSAGES 和 SCREEN 开关。

ZSIGSPEC



功能

显示和改变外部和内部 I/O 信号的总数。

说明

显示当前的设置和信息“改变?”。此指令仅改变软件设置。请确定信号数与硬件设置相一致。


示例

```
>ZSIGSPEC   
    DO,  DI,  INT          (DO=外部输出信号, DI=外部输入信号, INT=内部信号)  
    64   64   128  
    变化吗? (放弃请按 RETURN 键)  
    32, 32, 32  
    DO,  DI,  INT  
    32   32   32  
    变化吗? (放弃请按 RETURN 键) 
```

HSETCLAMP

功能  
给操作材料搬运夹紧分配信号编号。

示例  
在下述例子中，夹紧 3 被设置为双线圈。

>HSETCLAMP 

	夹紧 1	夹紧 2	夹紧 3	夹紧 4
	点焊	搬运	未使用	未使用
'ON' 定义输出信号	10	0	24	24
'OFF' 定义输出信号	9	11	0	0
	夹紧 5	夹紧 6	夹紧 7	夹紧 8
	未使用	未使用	未使用	未使用
'ON' 定义输出信号	24	24	24	24
'OFF' 定义输出信号	0	0	0	0

(夹紧编号 1~8, 只按 ENTER 键: 无变化, ^C:退出) 3 ; 选择夹紧编号

设定为搬运夹紧?

(1: 设定为搬运夹紧, 0: 不使用, 仅登录: 不更改, CTRL+C: 结束) 1 ; 输入 1, 定义为搬运夹紧

使用单电控电磁阀，请定义一个信号  
对于双控电磁阀，请指定两个。

' ON' 定义输出信号  
(0: 不使用, 仅登录: 不更改, CTRL+C: 结束) 更改吗? 12 ; 设置 ON 时输出 ch12。

' OFF' 定义输出信号  
(0: 不使用, 仅登录: 不更改, CTRL+C: 结束) 更改吗? 13 ; 设置 ON 时输出 ch13。

	夹紧 1	夹紧 2	夹紧 3	夹紧 4
	点焊	搬运	搬运	未使用
‘ON’ 定义输出信号	10	0	12	24
‘OFF’ 定义输出信号	9	11	13	0
	夹紧 5	夹紧 6	夹紧 7	夹紧 8
	未使用	未使用	未使用	未使用
‘ON’ 定义输出信号	24	24	24	24
‘OFF’ 定义输出信号	0	0	0	0

(夹紧编号 1~8, 只按 ENTER 键: 无变化, ^C:退出) 3 ; 选择夹紧编号

[ 注 意 ]

必须按从 1 到 8 的顺序使用夹紧。例如，没有使用夹紧 4 时，不能使用夹紧 5。

[ 注 意 ]

Ctrl+C (退出) 无法从示教器的键盘画面上使用。

DEFSIG INPUT

DEFSIG OUTPUT

**功能**  
显示和改变软件专用信号的当前设定。

**参数**  
INPUT, OUTPUT （输入, 输出）  
OUTPUT (或仅 O) 显示输出信号, INPUT (或仅 I) 显示输入信号。当输入此参数时, 可以改变设置。如果无参数输入, 将列表显示当前用作专用信号的信号。

**说明**  
下述列表中的信号可作为专用信号。

软件专用输入信号	软件专用输出信号
外部马达电源 ON	马达电源 ON
外部错误复位	错误发生
外部循环启动	自动
外部程序复位	循环启动
外部程序选择有效（ JUMP-ON, JUMP-OFF, RPS-ON, 外部程序编号）	示教模式 原点位置 1, 原点位置 2
外部暂停	电源 ON
外部低速再现模式	RGS0
	外部程序选择有效（JMP_ST, RPS_ST）

下述代码可以和此指令一起使用。  
L: 回到以前的信号  
N: 到下一个信号  
Q: 取消操作 (数据输入被忽略)  
E: 退出



[ 注 意 ]

1. 外部程序选择

(1) 当选择 JMP 作为专用信号时，信号 JMP-ON, JMP-OFF, JMP-ST 也被自动设置为专用信号。JMP-ST 是输出信号，但由 DEFSIG INPUT 指令设置。

(2) 当选择 RPS 作为专用信号时，信号 RPS-ON, RPS-ST 也被自动设置为专用信号。RPS-ST 是输出信号，但设置在 DEFSIG INPUT 指令之下。

2. RPS 代码

如果下述信号中至少一个被选为专用信号，提示出现并且必须输入 RPS 代码。

JMP, RPS 或 EXT. PROGRAM RESET

3. 信号编号

信号可以设定在以下范围之内：

专用输出信号：1～安装的信号数。

专用输入信号：1001～安装的信号数。

4. 其他

如果一个信号编号已经被指定给了专用信号，它不能被指派给其他专用信号或用作通用信号

示例

下述例子显示当前选择的软件专用信号。

>DEFSIG

当前设定的专用信号

外部马达电源 ON = 1032

外部故障复位 = 1031

外部循环启动 = 1030

马达电源已开启 = 32

出错 = 31

自动模式 = 30

条件：运行状态

条件：再现模式

条件：循环连续

条件：步骤连续

自动运行中 = 29

示教模式 = 28

第 1 原点 = 27

>

下述例子重新安排专用输出信号 MOTOR\_ON 的选择，将 AUTOMATIC 的信号编号改为 30，选择 TEACH MODE 作为专用信号并设置信号编号为 3。

>DEFSIG OUTPUT

马达电源已开启 专用解除? (输入 1 解除) 1

出错 专用解除? (输入 1 解除)

信号编号 31 更改吗? (1 - 32)

自动模式 专用解除? (输入 1 解除)

信号编号 2 更改吗? (1 - 32) 30

自动运行中 专用解除? (输入 1 解除)

信号编号 29 更改吗? (1 - 32)

示教模式 专用化? (输入 1 专用化) 1

信号编号 0 更改吗? (1 - 32) 3

第 1 原点 专用解除? (输入 1 解除)

>

## ZZERO 关节编号

### 功能

把在机器人的机械原点的位置检出器（编码器）的值作为调零值来设定。此指令也能表示当前的调零数据和编码器回转量计数器的值。

### 参数

关节编号

(1) 复位编码器旋转计数器:

输入关节编号加 100。例如，复位关节 2 的编码器旋转计数器，输入:

ZZERO 102 

如果输入“100”作为关节编号，所有编码器同时被复位为 0°。

(2) 设定调零数据:

为关节 2 设定编码器调零数据(零位数据)，输入:

ZZERO 2 

如果输入“0”作为关节编号，所有的关节都被调零、设置为 0°。

如果未指定关节编号，则显示当前的编码器值和调零数据。

#### [ 注 意 ]

在设定调零数据前，应先复位编码器旋转计数器。



#### 危 险

此指令仅用于下述目的:

1. 当手臂的位置不正常时，检查调零数据是否已经改变。
2. 当它被意外改变时，修正此调零数据。

当调零数据改变时，机器人位姿的检测值也同时改变。因此，请注意：  
在调零数据被改变前后，相同的程序会导致不同的目标姿态和轨迹。

## 示例

1. 下述指令显示调零数据:

```
>ZZERO 
      JT1      JT2      JT3      JT4      JT5      JT6
设定值    268435456  268435456  268435456  268435456  268435456  268435456
现在值    268435456  268435456  268435456  268435456  268435456  268435456
变化吗? (放弃请按 RETURN 键) 

      JT1      JT2      JT3      JT4      JT5      JT6
OFFSET      0        0        0        0        0        0
变化吗? (放弃请按 RETURN 键) 
>
```

2. 下述指令复位全部关节的编码器旋转计数器。

```
>ZZERO 100
**编码器旋转计数器复位 (全部轴) **
执行吗? (输入 1, 执行)  1 
设定完毕
>
```

3. 下述指令在指定关节位置 (关节 2) 调零设定, 复位其编码器旋转计数器。

```
>ZZERO 102
**编码器值旋转计数器复位 (第 2 轴) **
现在角度值? (deg, mm)  0 
执行吗? (输入 1, 执行)  1 
设定完毕
>
```

4. 下述指令同时设定所有关节为调零数据。

```
>ZZERO 0   
          JT1      JT2      JT3      JT4      JT5      JT6  
设定值 268427264 268427264 268427264 268427264 268427264 268427264  
现在值 268427264 268427264 268427264 268427264 268427264 268427264  
设定各轴现在值作为调零数值吗? (输入 1 设定.) 1   
设定完毕。 
```

5. 设定调零数据，以便关节 2 可以显示当前位姿的指定值。

```
>ZZERO 2   
现在角度值 (deg. mm)?      0  
变化吗? (放弃请按 RETURN 键)   
编码器值? (现在=268435456, 输入 1 设定设定现在值) 1   
调零值=268435456 (268419072-268451840) OK? (输入 0, 更改)   
设定完毕。   
>
```

---

## ERESET

---

### 功能

复位错误状态。与操作面板上的 **ERROR RESET**（错误复位）按钮功能相同。

### 说明

当执行 ERESET 指令时，ERROR\_RESET 信号被输出。但是在错误连续发生时，此指令无效。

---

## SYSINIT

---

### 功能

删除存储器中所有程序和数据并初始化定义的参数。

### 说明

初始化系统并删除所有程序、位姿变量、数字变量和字符串变量。

————— [ 注 意 ] —————

当本指令执行时，所有程序和变量被从存储器删除。

HELP	字母字符
HELP/M	字母字符
HELP/P	字母字符
HELP/F	字母字符
HELP/PPC	字母字符
HELP/MC	字母字符
HELP/DO	字母字符
HELP/SW	字母字符

功能

显示 AS 语言指令和命令的列表。

参数

指定指令、命令的开头字母。如果省略，则显示所有的指令和命令。

例如，输入 HELP 字母字符指令显示以该字母开头的程序命令或监控指令。输入 HELP/F 字母字符指令显示以那个字母开头的函数。

说明

- 仅输入 HELP，显示监控指令和程序命令的列表。
- HELP/M 列出监控指令。
- HELP/P 列出程序命令。
- HELP/F 列出函数。
- HELP/PPC 列出可以在 PC 程序中使用的程序命令。(选项)
- HELP/MC 列出可以和 MC 命令一起使用的监控指令。(选项)
- HELP/DO 列出可以和 DO 指令一起使用的程序命令。(选项)
- HELP/SW 显示系统开关列表。(选项)

对有些指令和命令，也可以显示参数。

示例

```
>HELP/M 
  ABORT  BASE   BITS  BATCHK  CONTINUE  COPY      DEFSIG
  DELETE  DIRECTORY  DLYSIG  DO    EDIT    ERESET   ERRLOG

>HELP/F 
  #DEST    #PPOINT  $CHR    $DECODE  $ENCODE  $ERROR
  $LEFT    $MID    $RIGHT  $SPACE  ABS      ASC
```

ID
----

功能

显示机器人控制器中安装的软件版本信息。

说明


显示下列信息。

- 机器人名: 当前连接的机器人名称
- 轴数: 机器人关节的数目
- 编号: 机器人的序列编号
- 软件版本: AS 软件的版本编号
- 伺服: 伺服软件的版本编号
- 信号数: 系统中可用的输出、输入和内部信号的总数
- 夹紧数: 系统中可使用的夹紧总数。
- 运作类型: 机器人的运动类型
- 伺服类型: 伺服软件的类型

[ 注 意 ]

如果上述信息与实际机器人不符，请立即与我们联系。请不要开启马达电源也不要使机器人做任何运动操作。

示例

```
>ID   
机器人名      : JS005-E001  轴数: 6   编号 No. 1  
伺服版本      : version  000004-04...97/01/27 13:11  
伺服          : SA0A00-UX120-01  
信号数: 输出信号=32   输入信号=32   内部信号=256  
夹紧数: 2   动作类型: 1   伺服类型:1
```



WEIGHT	负荷质量, 重心位置 X, 重心位置 Y, 重心位置 Z, 绕 X 轴的惯性力矩, 绕 Y 轴的惯性力矩, 绕 Z 轴的惯性力矩
--------	--

功能

设定负荷质量数据(工具和工件的重量)。这个数据用来确定机器人手臂的最佳加速度。

参数

负荷质量

工具和工件的质量(以千克计)。范围：0.0 到最大负载能力(kg)。

重心位置(单位= mm)


- X        在工具坐标系中重心的 x 值
- Y        在工具坐标系中重心的 y 值
- Z        在工具坐标系中重心的 z 值

绕 X 轴的惯性力矩, 绕 Y 轴的惯性力矩, 绕 Z 轴的惯性力矩 (选项)

设定绕各个轴的惯性力矩。单位是kg • m<sup>2</sup>。绕各个轴的惯性力矩被定义为围绕平行于空工具坐标系的各坐标轴的力矩，它的旋转中心在工具的重心。

说明

如果未指定参数，则显示当前值以及信息“更改？”。

 危 险

请始终设定正确的负荷质量和重心位置。错误的数据可能削弱或缩短零件的寿命或引起超载/偏差错误。

---

### ENCCHK\_EMG

---

#### 功能

设置紧急停止时机器人的位姿与机器人重启时的位姿之间的偏差检查的容许范围

#### 说明

偏差=|(马达电源再次上电后的位姿) - (紧急停止后的位姿)|

可接受的偏差范围可以在各关节处设置。如果范围设置为 0.0，不执行偏差检查。如果范围设置得太小，那么马达电源在紧急停止后重新上电时，即使机器人在其性能规格内操作，也有可能引发错误。

---

### ENCCHK\_PON

---

#### 功能

设定电源开启 (ON) 时编码器值与上次电源关闭 (OFF) 时值之间差异的可接受范围。

#### 说明

容许范围=|(控制电源开启时的值) - (上次控制电源关闭时的值)|

可接受的偏差范围可以在各关节处设置。如果范围设定得太小，那么马达电源在紧急停止后重新上电时，即使机器人在其性能规格内操作，也有可能引发错误。

---

## SLOW\_REPEAT

---

### 功能

设定低速再现模式中的速度

### 说明

>SLOW\_REPEAT

低速再现模式 (1~25%)

(仅登录: 不变化 ^C: 结束): 现在 10 更改吗?)

如果不作任何改变, 仅需按  键。要改变速度。输入新的值并按  键。

[ 注 意 ]

+ (结束) 不能在示教器的键盘画面上使用。

---

## REC\_ACCEPT

---

### 功能

允许或禁止 RECORD 和 PROGRAM CHANGE 功能。

### 说明

>REC\_ACCEPT

记录 (0: 允许, 1: 禁止)

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

程序变更 (0: 允许, 1: 禁止)

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

输入 0 允许 RECORD 或 PROGRAM CHANGE 选项, 输入 1 禁止选项。

[ 注 意 ]

1. + (结束) 不能在示教器的键盘画面上使用。
2. 如果 PROGRAM CHANGE 被禁止, 执行 EDIT 指令时出现下列信息: (P1036) “程序修改被禁止, 设置[ACCEPT], 并重新运行。”) REC\_ACCEPT 指令不能用在 EDIT (编辑) 模式中。

---

## ENV\_DATA

---

### 功能

设定硬件环境数据。(自动伺服 OFF 计时和示教器安装的状态)

### 说明

>ENV\_DATA

自动伺服关闭时间(0: 伺服不关断))

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

如果不作任何改变, 仅需按下  键。输入 0 使得自动伺服OFF计时无效。要使得计时生效, 输入多少时间(以秒为单位)后伺服关闭。

接下来, 显示有关示教器的提示。

示教器(0: 连接, 1: 未连接))

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

如果不作任何改变, 仅需按下  键。要不连接示教器而操作机器人, 则输入 1。在断开示教器后, 连接短路插头。

#### [ 注 意 ]

+ (退出)不能在示教器的键盘画面上使用。

---

## ENV2\_DATA

---

### 功能

设定软件环境数据。

### 说明

>ENV2\_DATA

示教器(0: 连接, 1: 不连接)

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

如果不作任何改变, 仅需按下  键。接下来, 显示终端设置的提示。

标准终端(0: 连接, 1: 不连接)

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

如果不作任何改变, 仅需按下  键。通常个人计算机被设置为终端。

#### [ 注 意 ]

+ (退出)不能在示教器的键盘画面上使用。

---

## CHSUM

---

### 功能

允许或禁止异常检查和错误的复位。

### 说明

>CHSUM

清除检查和出错 (0: 无效, 1: 有效)

(仅登录: 不变化 ^C: 结束): 现在 0 更改吗?)

如果输入“0”，错误不能被复位。如果输入“1”，则错误被复位。默认值是“0”。控制电源关闭时 CHSUM 被重置为“0”。

如果错误不能被复位，出现下述信息：

>CHSUM

不能清除检查和出错。请查看下一个指令或辅助功能数值。

ZZERO

DEFSIG

:

:

>

如果任何数据中仍包含一个异常的检查，第一个例子中的信息清除检查和出错不会出现。相反地，会显示一个信息以识别另外的故障(与第二个例子中一样)。

### [ 注 意 ]

Ctrl+C (退出)不能在示教器的键盘画面上使用。

---

## **TPLIGHT**

---

### **功能**

开启示教器的背光灯。

### **说明**

如果示教器屏幕的背光灯为 OFF，本指令打开(ON)灯。当背景灯为 ON 时执行本指令，灯光将再亮 600 秒后熄灭。

IPEAKLOG

选项


功能

显示各个关节的峰值电流值。

说明

显示各个轴在马达力矩到达最高时的程序名、步骤编号、有效电流值[手臂]、以及峰值相对于机械极限或马达极限的比率。

示例

```
>IPEAKLOG   
计测开始 00/1/24 13:44:23
```

轴	程序	步骤	电流峰值		日期
JT1	pg223	5	4.1[Arms]	29.5[%]	00/1/24 13:44
JT2	pg223	13	1.4[Arms]	9.8[%]	00/1/24 13:44
JT3	pg223	10	13.7[Arms]	98.2[%]	00/1/24 13:44
JT4	pg223	1	2.1[Arms]	55.5[%]	00/1/24 13:45
JT5	pg223	7	2.4[Arms]	64.8[%]	00/1/24 13:45
JT6	pg223	4	1.0[Arms]	27.8[%]	00/1/24 13:45

IPEAKCLR

选项


功能

清除峰值电流值日志并重新开始记录值。

说明

用 IPEAKCLR 指令、SYSINI 指令或通过开启 1KA 板上的 No.8 拨动开关初始化系统，都可以复位清除日志值。

示例

```
>IPEAKCLR  
确认吗？(是：1，否：0) 1   
>
```

OPEINFO

机器人编号: 关节编号

功能  
显示操作信息。

参数  
机器人编号  
指定一个机器人，如果有多个的机器人由一个控制器控制。

关节编号  
指定要显示的关节编号。如果未指定，则显示所有关节的数据。

示例

```
>OPEINFO [  ]
运行信息 (02/1/14  9:54:1 - ) (FILE LOAD  02/1/14); 描述从什么时间开始数据累积。
控制电源 ON 时间      0.2 [时间]
伺服 ON 时间          0.1 [时间]
马达 ON 次数          4
伺服 ON 次数          10
紧急停止(运行中)次数  2
JT1
移动时间              0.1 [时间]
移动时间              301.28 [x100 deg, mm]
JT2
移动时间              0.1 [时间]
移动时间              193.84 [x100 deg, mm]
:
```

[ 注 意 ]

- 数据累积的限制
- 1. 小时 [H]: 69 年
  - 2. 操作次数: 约 2000 百万次(如果一天操作 10000 次, 则需 1176 年)
  - 3. 距离[deg, mm]: 12.5 年, 以 500mm/s 的速度。

OPEINFOCLR

功能  
复位清除操作信息。



---

## REFFLTSET\_STATUS

---

### 功能

显示指令值的平均化时间的默认值和当前值。

### 说明

通过 REFFLTSET 命令来确认指令值的平均化时间的默认值和当前值。

### 示例

在下面的样本程序中，指令值的平均化时间的默认值为 48 ms。当机器人从#p1 移动到#p2 时，如果执行 REFFLTSET\_STATUS 指令，则屏幕显示如下。

### 程序示例

```
1 JMOVE #p1
2 REFFLTSET 128
3 JMOVE #p2
```

屏幕显示例（当正在执行步骤 3 时，执行 REFFLTSET\_STATUS 命令。）

```
>REFFLTSET_STATUS
默认值 Default value [ms] = 48 48 48 24
现在值 [ms] = 128 128 128 64
```

---

## FFSET\_STATUS

---

### 功能

显示机器人的速度/加速度前馈增益的默认值和现在值。

### 说明

通过 FFFSET 命令来确认速度/加速度前馈增益的默认值和现在值。

### 示例

在下面的样本程序中，速度/加速度前馈增益的默认值为 0.7。当机器人从#p1 移动到#p2 时，如果执行 FFFSET\_STATUS 指令，则屏幕显示如下。

### 程序示例

```
1 JMOVE #p1
2 FFFSET 0.5
3 JMOVE #p2
```

屏幕显示例（当正在执行步骤 3 时，执行 FFFSET\_STATUS 命令。）

>FFSET\_STATUS

默认值: KVFF = 0.70, 0.70, 0.70, 0.70, 0.70, 0.70

现在值: KVFF = 0.50, 0.50, 0.50, 0.50, 0.50, 0.50

默认值: KAFF = 0.70, 0.70, 0.70, 0.70, 0.70, 0.70

现在值: KAFF = 0.50, 0.50, 0.50, 0.50, 0.50, 0.50

## 5.7 二进制信号指令

E 系列机器人控制器使用两种类型的二进制信号：机器人和外部设备之间的外部 I/O 信号，和机器人内的内部 I/O 信号。内部 I/O 信号用于机器人和 PC 程序之间、或在实际连接外部设备前作为测试信号检查程序。

二进制信号用以下指令控制或定义。

RESET	关断全部外部输出信号(变为 OFF)。
SIGNAL	开启(ON)/关断(OFF)外部 I/O 信号和内部信号。
PULSE	开启(ON)输出信号指定的时间量。
DLYSIG	在指定的时间过去后开启输出信号(变为 ON)。
BITS	设定信号等于指定值(至多 16 个信号)。
BITS32	设定信号等于指定值(至多 32 个信号)。
SCNT	在读到一个指定值时输出计数器信号。
SCNRESET	清除计数器信号编号。
SFLK	打开(ON)/关闭(OFF)闪烁信号。
SFLP	用 SET/RESET 信号打开(ON)/关闭(OFF)信号。
SOUT	指定条件满足时输出信号。
STIM	指定信号为 ON 时开启(ON)计时信号。
SETPICK	设定开始夹具闭合控制的时间。
SETPLACE	设定开始夹具打开控制的时间。
HSENSESET	开始指定传感器信号的监控。(选项)
HSENSE	读由 HSENSESET 存储的数据。(选项)

---

## RESET

---

### 功能

关断全部外部输出信号(变为 OFF)。专用信号、夹紧信号和多功能 OX/WX 互斥信号不受此指令的影响。

通过用选项设置，接口面板画面所用的信号不受此指令的影响。(选项)

### [ 注 意 ]

小心，即使在再现模式中，此指令将关闭除了上述那些之外所有信号。

SIGNAL 信号编号, .....

**功能**  
开启 (ON) (或关断 (OFF)) 指定的外部或内部信号。

**参数**  
信号编号  
选择外部输出信号或内部信号的编号。正编号开启信号 (ON)，负编号关断信号 (OFF)。

**说明**  
信号编号决定了信号是外部信号还是内部信号。

容许的信号编号	
外部输出信号	1—实际信号编号
1 内部信号	2001—2960
外部输入信号	不能定义

如果信号编号是正，信号被开启 (ON)；如果为负，信号被关断 (OFF)。如果信号编号是“0”则所有信号被关断（OFF）。专用信号、夹紧信号和多功能 OX/WX 互斥信号不受此指令的影响。

**示例**

>SIGNAL -1, 4, 2010

外部输出信号 1 为 OFF，4 为 ON，内部信号 2010 为 ON。

>SIGNAL -reset, 4

如果变量“reset”的值是正值，由那个值决定的输出信号被关断 (OFF)，输出信号 4 被开启为 ON。

PULSE 信号编号, 时间

功能

开启 (ON) 指定的信号维持给定的时间长度。

参数

信号编号

选择外部输出信号或内部信号编号 (仅为正值)。如果信号已经被用作专用信号则报出错。

容许的信号编号

外部输出信号	1—实际信号数或 64 (取两者中的较小值)
内部信号	2001—2960

2. 时间

设定信号输出的时间长度 (以秒为单位)。如果未指定, 则自动设定为 0.2 秒。

DLYSIG 信号编号, 时间

功能

在给定的时间过去后输出指定信号。

参数

信号编号

选择外部输出信号或内部信号编号。如果信号编号是正值, 信号被开启 (ON); 如果为负值, 信号被关闭 (OFF)。如果信号编号已被用作专用信号则报出错

容许的信号编号

外部输出信号	1—实际信号数或 64 (取两者中的较小值)
内部信号	2001—2960

时间

指定暂停信号输出的时间, 以秒为单位。

**BITS 开始信号编号, 信号数=值**

**功能**

以二进制形式安排一组外部输出信号或内部信号。按照指定值的二进制值，设定信号状态 ON/OFF。如果未指定值，则显示当前信号状态。

**参数**

开始信号编号

指定要设定信号状态的第一个信号。

信号数

指定要被设定 ON/OFF 状态的信号数量。允许的最大值为 16。要设定超过 16 个信号，请用下面说明的 BITS32 指令。

值

指定用来设定想要的 ON/OFF 信号状态的值。此值被转换为二进制数值，二进制数的每一位设定信号的状态。最低位对应于最小信号编号的信号。如果这个值的二进制数位数多于信号数量，仅设定给定数量的信号状态(从指定的信号编号开始)，而其余位被忽略。

如果此参数省略，则显示当前信号状态。

**说明**

按照给定值设定(或复位)一个或多个外部输出信号或内部信号的信号状态。


容许的信号编号	
外部输出信号	1—实际信号数
内部信号	2001—2960

指定信号编号大于实际安装的信号数时将导致出错。选择一个专用信号也会导致错误。

示例

>BITS 2001, 3 

显示内部信号 2001-2003 的值  
(从信号编号 2001 开始的 3 位)。

>BITS 1, 8=100 

外部输出信号 1-8 被设置为输出 01100100  
(100 的二进制数)。



**BITS32 开始信号编号, 信号数=值**

**功能**

用二进制形式安排一组外部输出信号或内部信号。按照指定值的二进制数值设定信号状态 ON/OFF。如果未指定值，则显示当前信号状态。

**参数**

开始信号编号  
指定要设定信号状态的第一个信号。

信号数量  
指定被设定 ON/OFF 的信号数量。允许的最大数量是 32。

十进制值  
指定用来设定想要的 ON/OFF 信号状态的值。此值被转换为二进制值，二进制数的每一位设定信号的状态。最低位对应于最小信号编号的信号。如果这个值的二进制数位数多于信号数量，仅设定给定数量的信号状态(从指定的信号编号开始)，而其余位被忽略。

如果此参数省略，则显示当前信号状态。


**说明**

按照给定值设定(或复位)一个或多个外部输出信号或内部信号的信号状态。

容许的信号编号	
外部输出信号	1－实际信号数
内部信号	2001－2960

指定信号编号大于实际安装的信号数时将导致出错。选择一个专用信号也会导致错误。

**示例**

>BITS32 1, 32=<sup>^</sup>H7FFFFFFF  外部输出信号 1-32 被设定为对应于 7FFFFFFF 的二进制数。外部输出信号 1-31 变为 ON。

---

**SCNT 计数器信号编号=加信号, 减信号, 计数器清除信号, 计数器值**

---

**功能**

当指定的计数值到达时, 输出计数器信号。

**参数****计数器信号编号**

指定要输出的信号编号。计数器信号编号的设置范围: 3097 到 3128。

**加信号**

通过信号编号或逻辑表达式指定。每次这个信号由 OFF 变为 ON 时, 计数器值增加 1。

**减信号**

通过信号编号或逻辑表达式指定。每次这个信号由 OFF 变为 ON 时, 计数器值减少 1。

**计数器清除信号**

通过信号编号或逻辑表达式指定。如果此信号变为 ON, 内部计数器被复位为 0。

**计数器值**

当内部计数器到达这个值时, 指定的信号输出。如果给定为“0”, 计数器信号关闭(OFF)。

**说明**

当 SCNT 指令执行时, 如果加信号从 OFF 变为 ON, 那么内部计数器值增加 1。如果减信号由 OFF 变为 ON 时, 内部计数器值减少 1。当内部计数器的值到达参数中指定的值时(计数器值), 输出计数器信号。如果计数器清除信号被输出时, 内部计数器的值被设为 0。每个计数器信号有其单独的计数器值。要强制复位内部计数器的值为 0, 可用 SCNTRESET 指令。

要检查信号 3001 到 3128 的状态, 请用 IO/E 指令。(选项)

---

### **SCNRESET 计数信号编号**

---

#### **功能**

复位指定计数器信号编号的内部计数器值为 0。

#### **参数**

计数器信号编号

选择要复位的计数器信号编号。计数器信号编号的设定范围为：3097 到 3128。

---

### **SFLK 信号编号=时间**

---

#### **功能**

在指定的时间周期中开/关(闪动)指定信号。

#### **参数**

信号编号

指定要闪烁的信号编号。设定范围为：3065 到 3096。

时间

指定循环 ON/OFF 的时间(实数)，如果设定为负值，闪烁被取消。

#### **说明**

一个 ON/ OFF 的过程被当作一个周期，此周期在指定时间内执行。

---

**SFLP 输出信号=置位信号表达式, 复位信号表达式**

---

**功能**

用置位信号和复位信号开 (ON) / 关 (OFF) 一个输出信号。

**参数****输出信号**

指定输出的信号编号。正值表示开启 (ON) 信号；负值则表示关闭 (OFF)。只能指定输出信号的信号编号。(从 1 到实际的信号数量)

**置位信号表达式**

指定信号编号或逻辑表达式以设定输出信号。

**复位信号表达式**

指定信号编号或逻辑表达式以复位输出信号。

**说明**

如果置位信号为 ON，则输出信号被开启为 ON。如果复位信号为 ON，则输出信号被关闭为 OFF。输出信号在 SFLP 指令执行的时候开启或关闭，而不是在置位信号或复位信号变为 ON 的时刻。

SOUT 信号编号=信号表达式

功能

当指定的条件被置位时，输出指定的信号。

参数

信号编号

指定要输出的信号编号。仅可指定输出信号的信号编号。(1 到实际的信号数量)

信号表达式

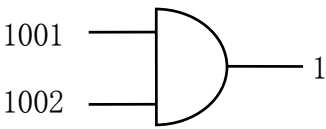
指定一个信号编号或逻辑表达式。

说明

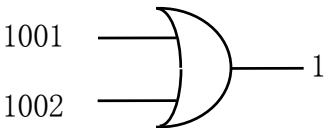
本指令用于信号的逻辑计算。使用如 AND 和 OR 等的逻辑表达式。当条件被置位时，指定的信号输出。

示例

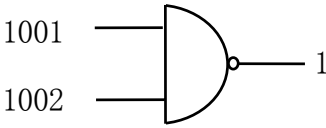
SOUT 1=1001 AND 1002



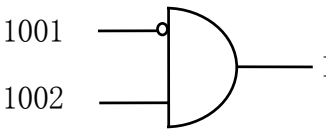
SOUT 1=1001 OR 1002



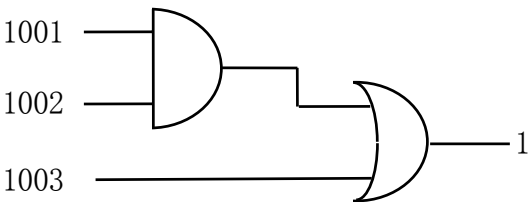
SOUT -1=1001 AND 1002  
SOUT 1=NOT (1001 AND 1002)



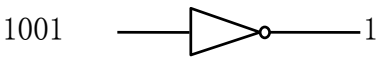
SOUT 1= -1001 AND 1002



SOUT 1=(1001 AND 1002) OR 1003



SOUT -1=1001 or SOUT 1= -1001 或  
SOUT 1=NOT (1001)



---

## STIM 计时信号=输入信号编号, 时间

---

### 功能

如果指定的输入信号为 ON 并持续了给定的时间, 则开启计时信号为 ON。

### 参数

#### 计时信号

选择要开启 (ON) 的信号。容许的信号编号从 3001 到 3064。

#### 输入信号编号

用整数的输入信号编号或逻辑表达式来指定, 用于监控开启计时信号的条件。此值不能超过实际安装的信号数量。

#### 时间

用实数指定时间 (秒); 在开启计时信号前, 输入信号必须在 ON 状态并持续了此指定的时间。

### 说明

被监控的输入信号必须为连续 ON 状态, 才能开启 (ON) 计时信号。如果在给定的时间过去前输入信号变为 OFF 状态, 那么当那个信号再次变为 ON 状态时, 计时将重新开始。如果输入信号关闭 (OFF), 计时信号也将立即关闭 (OFF)。但是, 输入信号仅在 STIM 执行时才影响计时信号。除非执行了 STIM, 否则即使输入信号关闭 (OFF), 计时信号仍将保持为 ON。

要检查信号 3001 到 3128 的状态, 请用 IO/E 指令。

### 示例

```
STIM 3001=1, 5
```

如果 sig1 保持 ON 状态 5 秒, sig2 将变为 ON。

```
SOUT 2=3001
```

```
>PCEXECUTE
```

---

SETPICK	时间 1,	时间 2,	时间 3,	时间 4,	...	时间 8
SETPLACE	时间 1,	时间 2,	时间 3,	时间 4,	...	时间 8

---

### 功能

为 8 个夹紧中的每个夹紧 (Clamp)，设定开始夹紧闭合控制 (SETPICK) 或夹紧打开控制 (SETPLACE) 的时间。

### 参数

时间 1—8

设定打开/关闭夹紧 1 到 8 的控制时间，以秒为单位。设定范围：0.0–10.0 秒。

### 说明

参看 CLAMP 命令。

---

**HSENSESET 监视编号=输入信号编号, 输出信号编号, 信号输出耽误的时间**

---

选项

**功能**

向 AS 系统声明信号检测的开始。当此命令执行时, AS 系统开始监视传感器信号并在信号变化时将位姿等数据积存到缓存中。保存在缓存中的数据可以用 HSENSE 命令来读取。缓存最多可以保存 20 个数据。

**参数**

**监视编号**

指定监视编号, 最多可以监视两个输入信号。每个信号的命令写作 HSENSESET 1 或 HSENSESET 2。

**输入信号编号**

设定要监视的信号编号。设置零(0)停止监视。

**输出信号编号**

设定当系统得到关节角度之后要输出信号的编号。指定的信号将开启 0.2 秒时间。此参数可以省略。

**信号输出耽误的时间**

设定在取得位姿数据后, 信号输出耽误的时间。设定范围是 0 到 9999 ms。可以省略。

— [ 注 意 ] —

即使在观察过程中控制电源被关断, 缓冲存储器仍将保留读取的数据。在再次开启控制电源后, 仍可以用 HSENSE 指令读取被保留的数据。但是监视不会自动重新开始, 因此必须再次执行 HSENSESET。

**示例**

>HSENSESET 1=wx\_sense 开始对输入信号 wx\_sense 进行监视。



---

**HSENSE 视编号 结果变量, 信号状态变量, 关节位移值变量, 位姿变量,  
内存使用量变量**

---

选项

### 功能

读取由 HSENSESET 命令保存在缓存中的数据。

### 参数

#### 监视编号

设定监视编号。要读取由 HSENSESET 1 保存的数据, 请指定 HSENSE1。要读取由 HSENSESET 2 保存的数据, 请指定 HSENSE 2。

#### 结果变量

为监视结果指定实型变量。在 HSENSE 命令执行后, 数值将被保存到此变量中。当 AS 系统没有检测到信号变化时, 将零(0)保存到此变量中。当 AS 系统检测到信号变化时, 将-1 保存到此变量中。

#### 信号状态变量

指定实型变量, 用于保存信号变化的状态。在 HSENSE 命令执行后, 数值被保存到此变量中。当信号从 OFF 变为 ON 时, ON (-1) 被保存在此变量中。当信号被从 ON 变为 OFF 时, OFF (0) 被保存在此变量中。

#### 位姿变量

指定在信号被检测到时, 为保存位姿数据的变量或排列变量。

#### 错误变量

指定为保存错误状态的实型变量。缓冲器很满时, 会产生错误。没有产生错误时, 保存为 0。缓冲器溢出时, 会产生错误, 保存为 0 以外的数值。缓冲器积存的数据超过 20 个后溢出。

#### 内存使用量变量

指定实型变量, 用来保存缓冲器中使用内存量。保存在此变量中的值显示了缓冲器中的已使用的内存量。当仅使用了一个内存, 保存为 0。当使用了所有的内存时, 变量值保存为 19。

## 5.8 信息显示指令

PRINT	显示数据。
TYPE	显示数据。
IFPWPRINT	在显示窗口中显示指定的字符串。
IFPLABEL	设定接口控制板上图标的标签。
IFPTITLE	设定接口控制板上指定页的标题。
SETOUTDA	指定模拟量输出环境。（选项）
OUTDA	按照设定的条件输出电压。（选项）

PRINT	设备编号: 打印数据, .....
TYPE	设备编号: 打印数据, .....

功能

在终端上显示参数中指定的打印数据。

参数

设备编号

选择用于显示数据的设备:

- 1: 个人计算机
- 2: 示教器

如果未指定, 则在当前选择的设备上显示数据。

打印数据

从下面选择一个或多个。当指定多个时, 用逗号隔开数据。

(1) 字符串	例如	“count=”
(2) 实数值表达式 (计算并显示数值)	例如	count
(3) 格式信息 (控制输出信息的格式)	例如	/D, /S

如果未指定参数则显示一个空行。

说明

如果输入“2”作为设备号, 则示教器画面自动切换到键盘画面。按下<下一页>返回常规画面。

下述代码用来指定数字表达式的输出格式。在指定不同的代码前, 一直使用相同的格式。在任何格式中, 如果数值太大以至于不能以给定的宽度显示, 将用星号(\*)填满空格。这时, 请改变显示字符的个数。一行中所能显示的最大字符数为 128。如果要在一行中显示超过 128 个字符, 请用下一页说明的/S 代码。

[ 注 意 ]

如果 MESSAGES 开关为 OFF, 则不显示任何消息在终端屏幕上。

## 格式指定代码

- /D            使用默认格式。除了数值后的 0 和所有空格（不包括数值之间的那个空格）都被删除了之外，/D 与指定格式为/G15.8 时相同。
- /Em. n        用科学计数法（例如-1.234 E +02）显示数值。“m”描述在终端上显示的字符总数，“n”表示小数点位置。“m”应该比 n 大 5 或更多。
- /Fm. n        用小数点固定的方法显示数值（例如-1.234）。“m”描述在终端上显示的字符总数，“n”表示小数部分的位数。
- /Gm. n        如果数值大于 0.01 而且能用 m 位的 Fm. n 格式显示，则用 Fm. n 格式显示。否则，数值用 Gm. n 格式显示。
- /Hn           将数值显示成 n 位的十六进制数。
- /In           将数值显示成 n 位的十进制数。

下述参数用来在字符串间插入确定的字符。

- /Cn           在打印数据的前或后，在代码输入的地方插入 n 次换行。如果此代码被置于打印数据中，则插入 n-1 个空行。
- /S            此行不换行。
- /Xn           插入 n 个空格。
- /Jn           将数值显示成 n 位的十六进制数。空白部分显示为 0。（选项）
- /Kn           将数值显示成 n 位的十进制数。空白部分显示为 0。（选项）
- /L            除了删除全部空格外，此代码与/D 相同。（选项）

### 示例

在本例中，实型变量“i”的值是 5，数组变量“point”的第 5 个元素是 12.66666。

```
>PRINT "point", i, "=", /F5.2, point [i] ↵
```

```
point 5=12.67      将像左边这样显示。
```

```
point 5=*****
```

如果 point[5]的数值是 1000 (1000.00)，则会显示成左边这样。数值太大导致无法显示（即数字位数大于 5）。

下述例子中/S 代码用来以数据后不换行显示数据。

```
>PRINT "ABC"  
>PRINT/S, "DEF"  
>PRINT "GHI" ↵
```

```
| ABC  
| DEFGHI |
```

应该如左边所示的显示，“GHI”将显示在“DEF”的同一行上。

**IFWPRINT 窗口, 行, 列, 背景颜色, 标签颜色=“字符串”, “字符串”, .....**

## 功能

在用辅助功能 0509 接口面板设置的字符串窗口中显示指定的字符串。

## 参数

### 窗口

选择在哪个窗口中来显示字符串。从辅助功能 0509 中的 1-8 (标准) 中选择。

### 行

为显示字符串, 指定窗口中的行。容许范围是 1—4, 但这取决于窗口尺寸的大小。如未指定, 则设定为 1。

### 列

为显示字符串, 指定窗口中的列。容许范围是 1—70, 但这取决于窗口尺寸的大小。如未指定, 则设定为 1。

### 背景颜色

选择选定窗口的背景颜色。容许范围是 0—15。如果未指定, 背景为白色。

No.	颜色	No.	颜色	No.	颜色	No.	颜色
0	灰	4	绿	8	粉红	12	深蓝
1	蓝	5	淡蓝	9	白	13	红棕
2	红	6	黄	10	黑	14	深绿
3	橙	7	白	11	蓝绿	15	淡紫

### 标签颜色

选择字符串显示的颜色。容许范围是 0~15。(见上表)。如果未指定, 字符串以黑色显示。

### 字符串

指定要显示的字符串。第一个字符串后面的全部字符串将从下一行的指定数位开始显示。IFWPRINT 的执行, 将从指定窗口中清除指定的字符串外的所有项目。

## 说明

IFWPRINT 指令仅在交互面板可用时才能使用。如果参数未指定, 则选用该特定窗口的上次设置。(如果是第一次使用, 则设置为上述默认值)。如果字符串无法在一行中显示, 那么将溢出到下一行中显示(会缩进到选定的列)。超出窗口尺寸的部分不显示。字符串中的控制字符将显示为空格。

---

**IFPLABEL 位置, “标签 1”, “标签 2”, “标签 3”, “标签 4”**

---

**功能**

设定并更改在指定位置的接口控制板图标标签。

**参数**

**位置**

指定要设定・更改的标签的接口控制板图标的表示位置。设定范围：1～112。

“标签 1”, “标签 2”...

指定标签上要表示的字符串。不更改的标签可以省略。

**说明**

设定・更改接口控制板上表示的图标的标签。当指定图标的没被设定的位置，或没有标签的图标时，什么都不设定。

**示例**

```
>IFPLABEL 10, , "标签 3"
```

只有接口控制板的位置 10 上的图标被更改为“标签 3”（标签 1，标签 2 没有更改，是由于其参数被省略）。

---

**IFPTITLE 页码, "标题"**

---

**功能**

指定并更改接口控制板的指定页的标题。

**参数**

页码.

指定要变更标题的接口控制板的页码。设定范围：1～4。

**“标题”**

指定指定页中的标题要表示的字符串。默认值设定为“接口控制板”。当指定 NULL 字符(“”)时，也表示为此默认值。

**说明**

设定 • 更改接口控制板的指定页所表示的标题。

**示例**

>IFPTITLE 1, "第 1 页"                      接口控制板的第 1 页的标题被更改为“第 1 页”。

>IFPTITLE 2, "" 接口控制板的第 2 页的标题被更改为“接口控制板”。



---

**SETOUTDA 端口编号=LSB, 位数, 逻辑, 最高电压, 最低电压**

---

### 功能

指定模拟量输出环境：端口编号和 LSB(最低位)、位数、信号输出逻辑电压、最大和最小电压。

### 参数

#### 端口编号

指定端口编号。设定范围：1 到 10 之间的整数。

#### LSB

用整数设定 D/A 转换输出的首信号编号。设定范围：OUT 1—OUT 125、2001—2125、3000 (1TW 端口的第一频道)、3001 (1TW 端口的第二频道)，以后增值的 ITW 端口的 DA 频道数。默认值为 3000。如果没有任何指定，前一次的设定值继续有效。

#### 位数

用整数设定 D/A 转换信号的位数。设定范围：4 位～16 位。当上面的参数 LSB 设定为 3000～[3000+1TW 端口的 DA 频道数]时，请设定 12 位。默认值为 8 位。如果没有任何指定，前一次的设定值继续有效。

#### 逻辑

设定输出数据是正逻辑(1)还是负逻辑(0)。默认值为 0 (负逻辑)。如果不指定，前一次的设定值继续有效。请为 1TW 设定为正逻辑。

#### 最高电压

指定硬件(D/A 输出)的最高电压。设定范围：-15—+15V。单位：V。默认值为 13V。化整到第一小数位。如果不指定，前一次的设定值继续有效。

#### 最低电压

指定硬件(D/A 输出)的最低电压。设定范围：-15—+15V。单位：V。默认值为 13V。化整到第一小数位。如果不指定，前一次的设定值继续有效。

参看 6.8 SETOUTDA, OUTDA 命令。

### [ 注 意 ]

1. 实际的输出电压依存于所使用的硬件。
2. 如果最高电压设定得比最低电压低时，将报出错。

---

**OUTDA 电压, 端口编号**

---

选项

**功能**

按照设定的条件向指定端口编号输出电压。

**参数**

**电压**

指定从 D/A 端口输出的电压。设定范围：-15—+15V。单位：0.1V。化整到第一小数位。

**端口编号**

指定端口编号。设定范围：1 到 16 之间的整数。如果未指定，默认为“1”。

———— [ 注 意 ] ————

请确认指令电压和实际输出电压一致，通过 SETOUTDA 命令(指令)将输出环境设置成与硬件设置相一致。



## 6.0 程序命令

本章按以下项目将程序命令分类，并详细介绍各命令。程序命令由一个表示命令的关键字和其后的参数组成，示例显示如下。

### 示 例

关键字 参数

↓ ↓ ↘

JMOVE 位姿变量, 夹紧编号

带        标记的参数可以省略。

关键字和参数之间必须用空格隔开。

示例中的  代表  键。

## 6.1 运动命令

JMOVE	以关节(各轴)插补动作移动机器人。
LMOVE	以直线插补动作移动机器人。
DELAY	停止机器人运动指定长度的时间。
STABLE	各轴与设置一致时(轴一致)停止机器人运动指定长度的时间。
JAPPRO	以关节(各轴)插补方式动作接近目的地。
LAPPRO	以直线插补动作动作接近目的地。
JDEPART	以关节插补动作离开当前位姿。
LDEPART	以直线插补动作离开当前位姿。
HOME	移动到原点位姿。
DRIVE	以单根轴的方向移动。
DRAW	沿基础坐标系 X, Y, Z 轴移动指定的量。
TDRAW	沿工具坐标系 X, Y, Z 轴移动指定的量。
ALIGN	将工具坐标系的 Z 轴和基础坐标系的轴对齐。
HMOVE	以直线插补动作移动 (腕关节以关节插补动作移动)。
XMOVE	以直线插补动作向指定位姿移动。
C1MOVE	以圆弧插补动作移动 (选项)。
C2MOVE	以圆弧插补动作移动 (选项)。

<b>JMOVE</b>	<b>位姿变量,</b>	<b>夹紧编号</b>
<b>LMOVE</b>	<b>位姿变量,</b>	<b>夹紧编号</b>

功能

移动机器人到指定位姿。

JMOVE：机器人以关节插补动作移动。

LMOVE：机器人以直线插补动作移动。

参数

位姿变量

指定机器人的目标位姿(可以是变换值变量、复合变换值变量、关节位移值变量或位姿信息函数等形式)。

夹紧编号

指定在目标位姿处要打开或闭合的夹紧编号。正数字闭合夹紧，负数字打开夹紧。任何夹紧编号(其最大值用 HSETCLAMP 命令或辅助功能 0605 设定)都可设定。如果省略，夹紧既不打开也不闭合(夹紧不动作)。

说明

当 JMOVE 命令执行时，机器人以关节插补动作移动。移动时，在机器人从起始位姿到结束位姿的整个运动过程中，各关节移动的行程相对于总行程的比例是相等的。

当 LMOVE 命令执行时，机器人以直线插补动作移动。工具坐标系原点(TCP)沿着直线轨迹移动。

示例

<b>JMOVE</b>	<b>#pick</b>	以关节插补动作，移动至由关节位移值“#pick”描述的位姿。
<b>LMOVE</b>	<b>ref+place</b>	以直线插补动作，移动至由复合变换值“ref + place”描述的位姿。
<b>LMOVE</b>	<b>#pick, 1</b>	以直线插补动作，移动至关节位移值“#pick”位姿处。到达该位姿时，1 号夹紧闭合。

---

## **DELAY 时间**

---

### **功能**

延迟；停止机器人运动指定的时间。

### **参数**

#### **时间**

以秒为单位指定机器人停止运动的时间。

### **说明**

在 AS 系统中，DELAY 命令被认为是一种“不向任何地方移动”的运动命令。

即使机器人动作由 DELAY 命令停止，下一个运动命令之前的所有程序步骤都会被执行。

### **示例**

DELAY 2.5                      停止机器人动作 2.5 秒

---

## **STABLE 时间**

---

### **功能**

推迟下一运动命令的执行，直至轴一致后指定的时间流逝。(等待至机器人稳定)

### **参数**

#### **时间**

以秒为单位指定机器人运动保持稳定状态的时间。

### **说明**

在机器人被此命令停止，而如果轴一致失败时，时间将从再次轴一致时开始计时。

<b>JAPPRO</b>	<b>位姿变量, 距离</b>
<b>LAPPRO</b>	<b>位姿变量, 距离</b>

功能

在工具坐标系 Z 方向上，移动到离示教位姿指定距离处。

JAPPRO: 以关节插补动作移动。

LAPPRO: 以直线插补动作移动。

参数

位姿变量

指定目标位姿(变换值变量或关节位移值变量)。

距离

指定目标位姿和机器人在工具坐标系 Z 轴方向上实际到达位姿之间的偏移值(单位: 毫米)。  
如果指定的距离为正值，机器人向工具坐标系 Z 轴的负方向运动。如果指定的距离为负值，机器人向工具坐标系 Z 轴的正方向运动。

说明

在这些命令中，工具的定向被设定为指定位姿处的定向，其位置被设定在工具坐标系 Z 轴方向上离指定位姿的指定距离处。

示例

<b>JAPPRO place, 100</b>	以关节插补动作，向工具坐标系 Z 轴方向上的、离位姿“place”100mm 处的位姿运动。“place”是用变换值描述的位姿。
<b>LAPPRO place, offset</b>	以直线插补动作，向工具 Z 轴方向上的离“place”位姿指定距离的位姿处移动，该距离值由变量“offset”定义。“place”是用变换值描述的位姿。



---

<b>JDEPART</b>	<b>距离</b>
<b>LDEPART</b>	<b>距离</b>

---

**功能**

沿工具坐标系 Z 轴，移动机器人到离当前位姿指定距离处的位姿。

JDEPART : 以关节插补动作移动。

LDEPART : 以直线插补动作移动。

**参数**

距离

以毫米为单位，沿着工具坐标系 Z 轴，指定当前位姿和目标位姿之间的距离。如果指定的距离为正值，机器人向后移动或向工具坐标系 Z 轴的负方向移动；如果指定的距离为负值，机器人向前移动或向工具坐标系 Z 轴的正方向运动。

**示例**

<b>JDEPART 80</b>	机器人工具以关节插补动作、向工具坐标系-Z 方向上的 80mm 处后退移动。
<b>LDEPART 2*offset</b>	机器人工具以直线插补动作，向工具坐标系-Z 方向上的 2*offset (200 mm，如果 offset = 100) 处后退移动。

---

**HOME**   **原点位姿编号**

---

**功能**

以关节插补动作、移动至以 HOME 或 HOME2 定义的位姿处。

**参数**

原点位姿编号

指定原点(home)位姿编号(1 或 2)。如果省略，则选定 HOME 1(原点 1)。

**说明**

共可设置两个原点(HOME 1 和 HOME 2)。此命令将机器人以关节插补动作移动到其中一个原点处。原点位姿应预先用 SETHOME 或 SET2HOME 命令设置好。如果原点未被定义，空原点(所有关节都在 0° )被默认为原点位姿。

**示例**

- |        |                                    |
|--------|------------------------------------|
| HOME   | 以关节插补动作运动到用 SETHOME 命令设定的原点处。      |
| HOME 2 | 以关节插补动作运动到用 SET2HOME 命令设定的第 2 原点处。 |

---

## DRIVE 关节编号, 位移, 速度

---

### 功能

移动机器人的单个关节。

### 参数

#### 关节编号

指定要移动的关节编号。(在六关节的机器人中, 关节被编号为 1 至 6, 从离工具安装法兰最远的关节处开始。)

#### 位移

指定移动关节的量, 可为正值或负值。

此数值的单位与描述关节位姿值的单位相同, 也就是说, 如果关节为旋转关节, 数值以度 (°) 表示, 如果关节为滑行关节, 数值用距离 (mm) 表示。

#### 3. 速度

为该运动指定速度。像正常的程序速度一样, 此速度用监控速度的百分比表示。如果未指定, 则默认为监控速度的 100%。

### 说明

此命令仅移动指定的关节。

此命令的运动速度, 是由此命令中指定的速度和监控速度合成的。程序中设置的程序速度并不影响此命令。

### 示例

**DRIVE 2, -10, 75**                      将关节 2 (JT2) 从当前位姿转动 -10°, 速度为监控速度的 75%。

---

<b>DRAW</b>	<b>X 平移量, Y 平移量, Z 平移量, X 旋转量, Y 旋转量, Z 旋转量, 速度</b>
-------------	---

---

<b>TDRAW</b>	<b>X 平移量, Y 平移量, Z 平移量, X 旋转量, Y 旋转量, Z 旋转量, 速度</b>
--------------	---

---

## 功能

机器人从当前位姿，以直线插补动作，以指定的速度，向 X, Y, Z 轴方向上指定的距离处移动，并且绕各轴旋转指定的旋转量。DRAW 命令按基础坐标系移动机器人，TDRAW 命令按工具坐标系移动机器人。

## 参数

### X 平移量

以 mm 为单位，指定在 X 轴上的移动量。若未指定，则默认为 0 mm。

### Y 平移量

以 mm 为单位，指定在 Y 轴上的移动量。若未指定，则默认为 0 mm。

### Z 平移量

以 mm 为单位，指定在 Z 轴上的移动量。若未指定，则默认为 0 mm。

### X 旋转量

以度为单位，指定绕 X 轴旋转的角度量。容许范围小于 $\pm 180^\circ$ 。若未指定，则默认为 0 度。

### Y 旋转量

以度为单位，指定绕 Y 轴旋转的角度量。容许范围小于 $\pm 180^\circ$ 。若未指定，则默认为 0 度。

### Z 旋转量

以度为单位，指定绕 Z 轴旋转的角度量。容许范围小于 $\pm 180^\circ$ 。若未指定，则默认为 0 度。

### 速度

以%、mm/s、mm/min、cm/min、或 s 为单位，指定速度。若未指定，机器人以程序速度移动。

## 说明

机器人以直线插补动作从当前位姿移动到指定位姿。

## 示例

**DRAW 50, , -30**      从当前位姿出发，以直线动作，在基础坐标系的 X 轴方向上移动 50 mm，并且在 Z 轴方向上移动 -30 mm。

---

## ALIGN

---

### 功能

移动工具坐标系的 Z 轴，使之平行于最接近的基础坐标系的坐标轴。

### 说明

在各中应用中，如果参考运动方向被设置为沿工具 Z 方向，在示教位姿数据前，使用 D0 ALIGN 命令，可以非常方便地使工具方向和基础坐标系对齐。

---

## HMOVE 位姿变量, 夹紧编号

---

### 功能

移动机器人至指定位姿。机器人按混合运动方式运动：主要轴为直线插补，腕关节为关节插补。

### 参数

#### 位姿变量

指定机器人运动的目标位姿。(可以是变换值变量、复合变换值变量、关节位移值变量或位姿信息函数)

#### 夹紧编号

指定在目标位姿处要打开或闭合的夹紧编号。正数字闭合夹紧，负数字打开夹紧。任何夹紧编号都设置到 HSETCLAMP 命令 (或辅助函数 0605) 设置的最大值。如果省略，夹紧既不打开也不闭合。

### 说明

此命令是以直线插补动作移动机器人。工具坐标系的原点画一条直线轨迹，但腕关节以关节插补动作运动。当机器人将要按直线动作，而动作开始和结束之间的腕关节的角度变化很大时，使用此命令。

XMOVE

模式

位姿变量

TILL

信号编号

**功能**  
机器人以直线插补动作向指定位姿移动，当指定的信号条件置位时，即使未到达指定的位姿，机器人也会停止运动，并跳转到下一步骤。

**参数**  
**模式**  
(不是指定项)  
监视指定输入信号的上升沿或下降沿。正信号编号监视上升沿，负编号监视下降沿。

/ERR (选项)  
如果信号条件在监视开始之前已经置位，则会返回一个错误信息。

/LVL (选项)  
如果信号条件在监视开始之前已经置位，则立即跳转到下一步骤。

**位姿变量**  
指定机器人运动的目标位姿(可以是变换值变量、复合变换值变量、关节位移值变量或位姿信息函数)。

**信号编号**  
指定外部输入信号或内部信号的编号。

容许的信号编号

外部输入信号	1001 至实际安装的信号编号或 1064 (取两者中的较小值)
内部信号	2001 至 2960

[ 注 意 ]

在监视信号的上升和下降沿时，程序仅在信号状态出现变化时执行分支。因此，如果在 XMOVE 命令执行时，该信号已经变为 ON，则程序不会被中断，直到该信号再次由 OFF 变为 ON 时才中断。

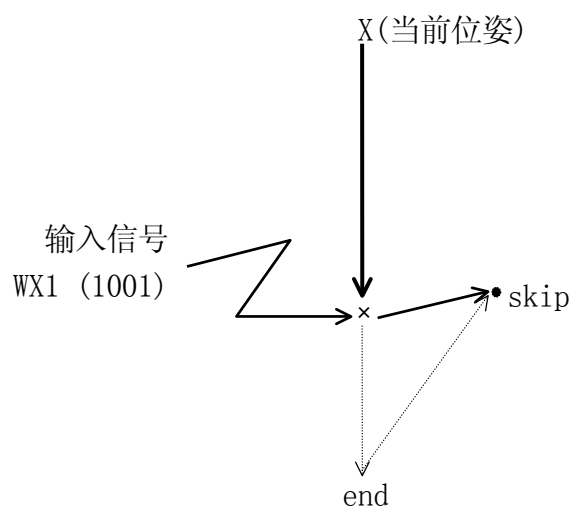
为保证精确监视，输入信号至少需要稳定 50 毫秒。

### 示例

XMOVE end TILL 1001

LMOVE skip

以直线插补动作，从当前位姿向“end”位姿处运动。一旦输入信号 1001 变为 ON，即使机器人还未到达“end”，程序的执行也跳转到下一步骤(LMOVE skip)。



C1MOVE	位姿变量, 夹紧编号	
C2MOVE	位姿变量, 夹紧编号	

选项

功能

在沿着圆弧路径，移动机器人至指定的位姿。

参数

位姿变量

指定机器人运动的目标位姿(可以是变换值变量、复合变换值变量、关节位移值变量或位姿信息函数)。

夹紧编号

指定在目标位姿处打开或闭合的夹紧编号。正数字闭合夹紧，负数字打开夹紧。任何夹紧编号都可指定，用 HSETCLAMP 命令(或辅助函数 0605)可设置该编号的最大值。如果省略，夹紧既不打开也不闭合。

说明

C1MOVE 命令将机器人移动到圆弧轨迹的中间位置点，C2MOVE 命令移动机器人至该圆弧轨迹的结束点。

要使机器人以圆弧插补动作进行移动，必须示教三个位姿点。在 C1MOVE 和 C2MOVE 命令中，这三个位姿是不同的。

- C1MOVE:
- 1. 上一条运动命令的位姿。
  - 2. 用作 C1MOVE 命令参数的位姿。
  - 3. 下一条运动命令的位姿(C1MOVE 或 C2MOVE 命令)。

- C2MOVE:
- 1. 上一条 C1MOVE 命令的位姿。
  - 2. 在 C1MOVE 命令之前的运动命令的位姿。
  - 3. C2MOVE 命令的位姿。



[ 注 意 ]

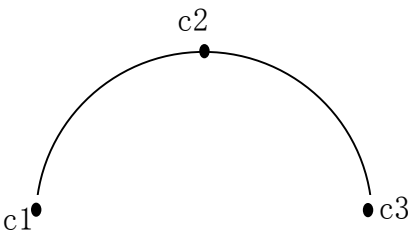
在 C1MOVE 命令之前，需要有如下命令：  
ALIGN, C1MOVE, C2MOVE, DELAY, DRAW, TDRAW, DRIVE, HOME,  
JMOVE, JAPPRO, JDEPART, LMOVE, LAPPRO, LDEPART, STABLE,  
XMOVE

C1MOVE 命令后必须跟有 C1MOVE 或 C2MOVE 命令。

C1MOVE 命令必须先于 C2MOVE 命令。

示例

JMOVE c1  
C1MOVE c2  
C2MOVE c3

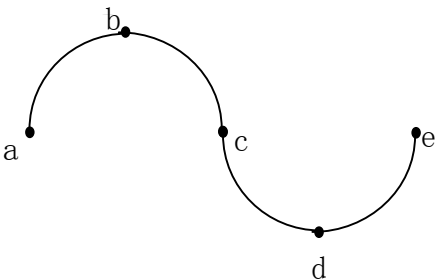


机器人以关节插补方式运动到 c1，然后沿着由 c1、c2、c3 形成的圆弧作圆弧插补运动。

JMOVE #a  
C1MOVE #b  
C2MOVE #c  
C1MOVE #d  
C2MOVE #e

} 圆弧 a、b、c

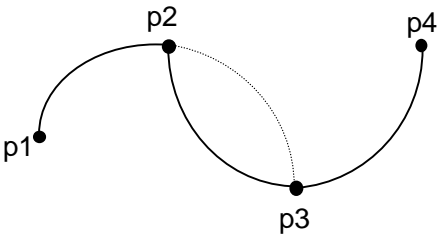
} 圆弧 c、d、e



LMOVE #p1  
C1MOVE #p2  
C1MOVE #p3  
C2MOVE #p4

} 圆弧 p1、p2、p3

} 圆弧 p2、p3、p4



## 6.2 速度和精度控制命令

SPEED	设定运动速度(程序速度)。
ACCURACY	设定精度范围。
ACCEL	设定加速度。
DECEL	设定减速度。
BREAK	暂停下一步骤的执行，直到当前动作完成。
BRAKE	停止当前动作并跳转到下一步骤。
BSPEED	设定一体化命令程序的速度(选项)。
REFFLTSET	指定机器人指令值的平均化时间。
REFFLTRESET	复位机器人的平均化时间。
FFSET	设定速度/加速度前馈增益。
FFRESET	复位机器人速度/加速度前馈增益。

---

**SPEED 速度, 旋转速度, ALWAYS**

---

### 功能

指定机器人的运动速度

### 参数

#### 速度

指定程序速度。通常以百分比用 0.01 至 100(%) 之间的值指定。绝对速度也可以通过用这些单位：MM/S 和 MM/MIN 来指定。单位 S(秒)指运动时间。如果单位被省略，则默认为百分比(%)。

#### 旋转速度(选项)

指定在直线插补和圆弧插补动作时工具定向的旋转速度。通常指定为 0.01 至 100(%) 之间。绝对速度可以通过指定速度单位：DEG/S 和 DEG/MIN 来指定。如果单位被省略，则默认为百分比(%)。如果本参数省略，旋转速度则默认为 100%。

#### ALWAYS

ALWAYS 的中文意思是：始终；若本参数被输入，则此命令指定的速度值将保持有效，直到执行到下一 SPEED 命令。若未输入，则该速度值仅对下一条运动命令有效。

### 说明

机器人动作的实际速度，由监控速度和此命令指定的运动速度之乘积来决定(监控速度×程序速度)。然而，在如下情况下不能保证全速：

1. 当两个示教点之间的距离过近时。
2. 当示教的直线运动超出了轴的最大旋转速度时。

关节插补运动和直线运动的运动速度的决定方式不同。在关节插补运动中，运动速度以各关节轴的最大速度的百分比来确定。而在直线插补运动中，运动速度以工具坐标系原点的最大速度的百分比来确定。

当速度以单位时间内的移动距离或以秒来指定时，则此指定的速度为工具坐标系原点的直线运动速度。当机器人以关节插补动作进行移动时，则以百分比设置速度。(即使速度是以绝对速度或运动时间指定的，机器人也不会以指定的速度运动。代之，速度将被处理为给定速度与最大速度的百分比。)

绝对速度以 MM/S 和 MM/MIN 为单位的值来表达，时间指定的速度以带 S 的值来表达，所描述的是当监控速度为 100%时的速度。如果监控速度降低，这些速度将以同比例降低。

— [ 注 意 ] —

即使程序速度和由 SPEED 命令(监控速度)指定的速度的乘积超过了 100 %，但实际的运动速度也不会超过 100 %。

旋转速度必须在旋转速度控制选项为 ON 时才可指定。如果此选项不为 ON 时指定，就会出错。

示例

当监控速度为 100 %时，如下设定速度。

<b>SPEED 50</b>	将下一条运动的速度指定为最大速度的 50 %。
<b>SPEED 100</b>	将下一条运动的速度指定为最大速度的 100 %。
<b>SPEED 200</b>	将下一条运动的速度指定为最大速度的 100 % (速度超 100 %时被看作为 100 %)。
<b>SPEED 20MM/S ALWAYS</b>	工具坐标系原点(TCP)的速度被指定为 20 mm/sec，直至它被另一 SPEED 命令改变(当监控速度为 100 %时)。
<b>SPEED 6000 MM/MIN</b>	机器人下一条运动的速度被指定为 6000 mm/min(该速度为工具坐标系原点的速度，假定监控速度为 100 %)。
<b>SPEED 5 S</b>	设定下一条机器人运动的速度，使其在 5 秒种内到达。(该速度是监控速度为 100 %时的工具坐标系原点的速度)。
<b>SPEED 100 MM/S, 10 DEG/S</b>	指定下一条运动的速度。到达目标位姿所需时间长者优先。

---

ACCURACY	距离	ALWAYS	FINE
----------	----	--------	------

---

### 功能

指定判断机器人位姿时的精度。

### 参数

距离

以毫米为单位指定精度范围的距离。

#### ALWAYS

ALWAYS 的中文意思是：始终；若本参数被输入，则此精度设定保持有效，直到下一条 ACCURACY 命令执行。若未输入，则此精度设定仅对下一条运动命令有效。

#### FINE

如果输入本参数，不论“距离”参数是否设定，仅当当前值和示教值一致时，决定机器人的位姿。

### 说明

当输入 ALWAYS 参数时，所有的后继运动将由本命令指定的精度控制。

缺省的精度设定为 1 mm。

精度设定的影响是有限的，因为在 AS 系统中，精度检查并不在机器人接近示教的位姿而开始减速前开始。（参看 4.5.4 CP 开关和 ACCURACY、ACCEL、DECEL 命令之间的关系）。

#### [ 注意 ]

当精度指定为 1 mm 时，机器人在移动到每个动作命令的位姿后，将在两个运动段之间有一个暂停。如果要保证 CP(连续路径)运动，可将精度范围设定得大些。

请注意，不要把精度范围设得过小，否则可能会导致轴一致失败。

本命令指定的精度不是重复精度而是机器人的定位精度，因而请不要指定 1 mm 或更小的值。

### 示例

ACCURACY 10 ALWAYS

将此命令的所有后继运动命令的精度范围设定为 10 mm。

---

ACCEL	加速度	ALWAYS
DECEL	减速度	ALWAYS

---

## 功能

设定机器人运动的加速度(或减速度)。

## 参数

加速度 (ACCEL) / 减速度 (DECEL)

以最大加速度(减速度)的百分比，来指定加速度或减速度。容许范围从 0.01 到 100。数值超过此范围时被默认为 100，低于此范围的被默认为 0.01。

## ALWAYS

ALWAYS 的中文意思是：始终；若输入本参数，用此命令指定的加速度(或减速度)在下一条 ACCEL (或 DECEL) 命令执行前一直保持有效。若未输入，则此命令仅影响下一条运动命令。

## 说明

ACCEL 命令是以最大加速度的百分比，来指定机器人开始一个运动时的加速度。

DECEL 命令是以最大加速度的百分比，来指定机器人在动作结束处的减速度。

## 示例

ACCEL 80 ALWAYS            指定此命令后所有运动的加速度为 80 %。

DECEL 50                    指定下一条运动命令的减速度为 50 %。

---

## **BREAK**

---

### **功能**

暂停程序中的下一步骤的执行，直到当前机器人运动完成后再继续。

### **说明**

此命令有如下两种影响：

1. 暂停程序中下一步骤的执行，直到机器人到达当前运动命令的目标位姿后再继续。
2. 打断从当前运动到下一运动的连续运动 (CP)。机器人在两个运动段之间中停顿一下。

---

## **BRAKE**

---

### **功能**

停止当前的机器人动作。

### **说明**

立即停止当前的机器人运动，并跳到程序中的下一步骤。

**BSPEED 速度**

选项

**功能**

设定机器人的运动速度(一体化命令速度)。机器人运动速度计算如下：监控速度×程序速度×一体化命令速度。

**参数**

**速度**

指定速度(有效范围：1—1000 %)。此命令设定的速度值在下一 BSPEED 命令执行之前一直保持有效。

**说明**

机器人运动速度是由监控速度×程序速度×一体化命令速度计算得来的。但是总速度不能超过 100 %。各速度可以输入最大达 1000 的值，但如果总速度超过了 100 %，将自动降为 100 %。例如，如果监控速度为 100 %、程序速度为 50 %，运动速度计算如下：100 %×50 %×一体化命令速度。如果一体化命令速度小于 200 %，总速度由以上公式决定，但是如果大于 200 %，那么运动速度总是为 100 %。

[ 注 意 ]

1. 当程序选择复位时，并且一个新的程序执行时（由 EXECUTE 命令或通过示教器选择程序），一体化速度的默认值设定为 100%。当程序是由外部选择的，如果程序是通过外部程序复位选中，而不是被 RPS 和 JUMP 信号选中的，则一体化命令速度设定为默认值。
2. 请注意如果程序不是从头开始执行或有步骤被跳过时， 机器人将可能不按指定的程序速度运动。在下面例子中，机器人在第 3 步处停止，在程序跳转至第 25 步后继续运动。那么第 25 步处的一体化命令速度为 block1 的速度值。

Step 1	BSPEED block1	； 设置一体化命令速度为 block1
Step 2	Joint Speed 9……	
Step 3	Linear Speed 9……	
Step 12	BSPEED block2	； 设置一体化命令速度为 block2
Step 13	Joint Speed 9……	
Step 14	Linear Speed 9……	
Step 24	BSPEED block3	； 设置一体化命令速度为 block3
Step 25	Joint Speed 9……	
Step 26	Linear Speed 9……	



### 示例

编写如下的程序，实现通过外部信号的 4 个位来改变速度。

```
a=BITS(first signal for external speed selection, 4)
BSPEED block1[a]
```

以下程序可以从外部设备选择速度

```
BSPEED block1                ;设置一体化速度的缺省值。
IF SIG(External_speed ON) THEN ;判断外部速度选择是否有效。
a=BITS(first signal for external speed selection, 4);获取用于外部选择的数字。
IF (a<11) THEN                ;如果 a 大于等于 11，则设置无效。
BSPEED block11[a]             ;设置选择的一体化命令速度
END
END
Joint Speed 9.....          ;以选择的一体化命令速度运动
Joint Speed 9.....
```

实数变量 block1 必须预先定义。

```
block1=50
block111[0]=10
block111[1]=20
block111[2]=30
block111[3]=40
```

例如，如果外部程序选择的第一个信号为 1010，且输入的信号为：

```
1010• • • OFF
1011• • • ON
1012• • • OFF
1013• • • OFF
```

那么  $a = 2$ ，因而 block11[2] 被选中，运动速度变为 30 %。

---

**REFFLTSET 关节值平均化时间, 位置平均化时间, 姿势平均化时间, 信号平均化时间**

---

### 功能

指定机器人指令值的平均化时间。此命令仅在指令值的平均化选项有效时有效。

### 参数

#### 关节值平均化时间

指定在各轴插补动作下机器人的关节角度的平均化时间。单位：[ms]。容许范围：1-254 的整数。根据 AS 系统的控制周期调高指定值为整数。（参数 2-4 也同样）。

#### 位置平均化时间

指定在直线插补・圆弧插补动作下机器人的位置的平均化时间。单位：[ms]。容许范围：1-254 的整数。如果被省略了的话，设置与各轴平均化时间相同的值。

#### 姿势平均化时间

指定在直线插补・圆弧插补动作下机器人的姿势的平均化时间。单位：[ms]。容许范围：1-254 的整数。如果被省略了的话，设定与位置平均化时间相同的值。

#### 信号平均化时间(选项)

指定信号输出的平均化时间。单位：[ms]。容许范围：1-254 的整数。如果被省略了的话，该参数值设定为由各轴平均化时间的默认值和现在值的比，与信号平均化时间的默认值相乘的值。

### [ 注 意 ]

上述 4 个参数之间的相对关系被适当地设定。因此，当做任何修改时，通常仅有必要更改关节值平均化时间。这样一来，这些参数间的相对关系将保持适当的值。

各参数的默认值和现在值可以由 REFFLTSET\_STATUS 指令确认。

### 说明

平均化时间是为了使机器人顺利地到达指定的位姿而设定的。

如果该时间设定的长，那么机器人的振动将减少，并且机器人将更顺利地运动。但，周期时间将变长，并且机器人有比示教的路径走捷径的倾向。

另一方面，如果该时间设定的短，那么周期时间将缩短，机器人将比示教的路径走更精确的轨迹。但，机器人的震动有增加的倾向。

[ 注意 ]

由于使用此命令将改变其动特性，因此通常不使用此命令。当使用此命令时，要遵循以下的步骤：

- 逐步更改设定值（以约 8 ms 的增值），并在更改后确认机器人的运动。
- 当确认机器人运动时，以约 20%的低速监控速度开始，并逐渐的提高速度。

[ 注意 ]

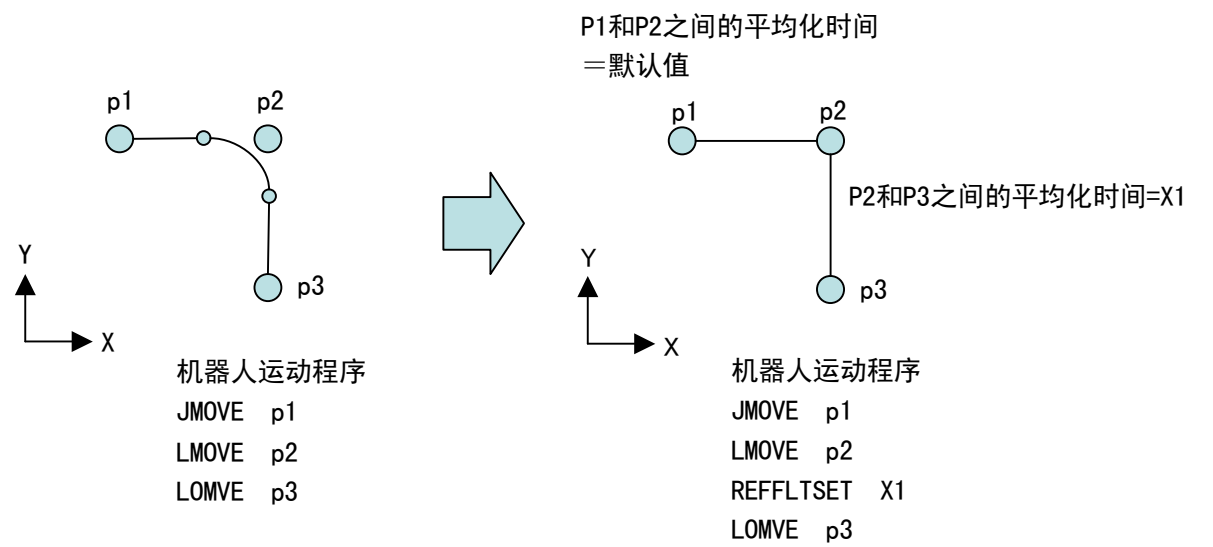
执行此命令设定的值，适用于所有的机器人运动，并与 SPEED ALWAYS 命令相同。如果要复位默认值，则使用 REFFLTRESET 命令。

当执行 REFFLTSET 命令时，在当前运动和下一运动命令之间的连续路径运动被中断。即，该两个运动不跟随在一个连续的运动。但，在进入第二运动之前，在第一运动结束时，机器人将暂时停止。

- 示例
- REFFLTSET 64

设定机器人的关节值・位置・姿势平均化时间为 64 ms。  
信号平均化时间被设定为默认值。
- REFFLTSET 64, 64, 64, 32

设定机器人的关节值・位置・姿势平均化时间为 64 ms。  
设定信号平均化时间为 32 ms。



遵循以下的步骤执行减少振动的对策：

- 1) 通过减小加减速来减少振动。用 AS 命令 ACCEL/DECEL 来更改加减速。

对于简易示教设定，通过 A-0301 速度的加速/减速设定来更改加减速。（仅当[加减速设定]在 A-0399 中设定为[有效]时，A-0301 设定可能。）

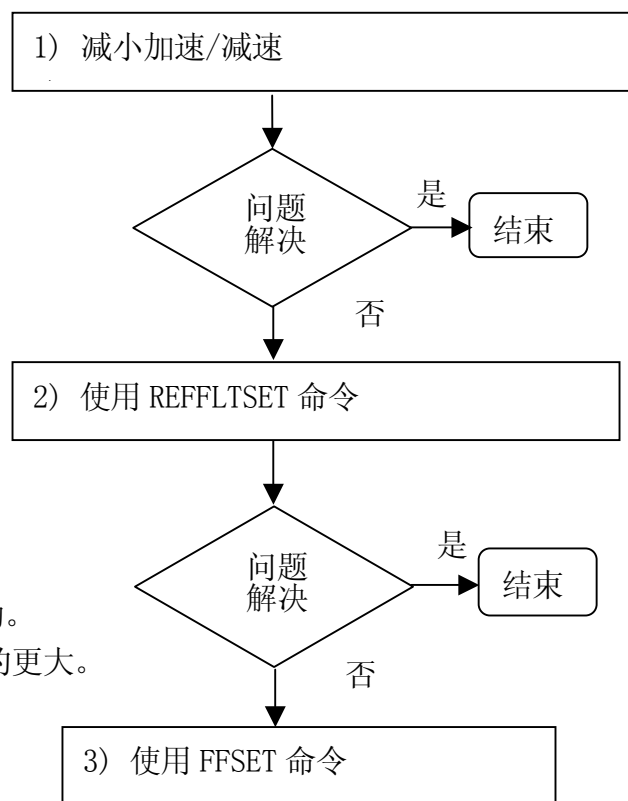
当振动未减小或周期时间太长，复位其到原设定并执行下一调节。

- 2) 用 REFFLTSET 命令顺利进行机器人运动。  
用 REFFLTSET 命令将平均化时间设定的更大。  
在平均化时间设定为更大的区间（在 REFFLTSET 和 REFFLTRESET 部分）满足以下的条件时有效：

- 其大于一个步骤。
- 当前位姿轴一致的步骤少。
- 短距离（小于 50 mm）的步骤多。
- 在 REFFLTSET/REFFLTRESET 命令前的运动步骤中的精度设定小。

建议您与平均化时间一同调节加减速。执行 REFFLTSET 命令来改变机器人的路径，因此要小心确认运动。

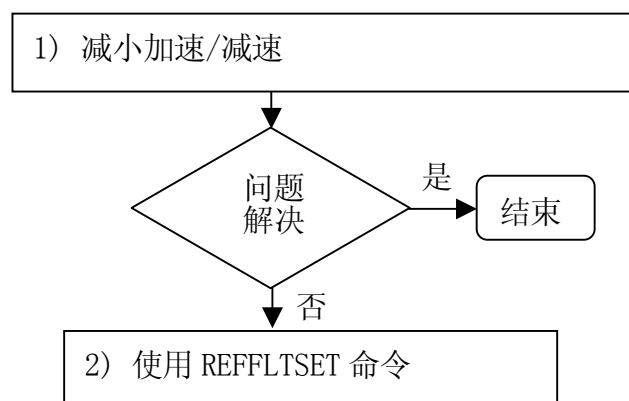
在使用 REFFLTSET 命令更改加减速后，如果振动不减少或周期时间太长，则复位 REFFLTSET 命令和加速/减速设定到原设定并用 FFSET 命令执行下一调节。FFSET 命令在本节的最后介绍。



遵循以下的步骤执行精度提高对策：

- 1) 以低速确认路径精度。  
使用 AS 命令 SPEED 或在简易示教设定的速度命令来更改速度。

如果路径精度未提高或速度设定不符合应用条件，设定速度到原设定并执行下一调节。



- 2) 通过 REFFLTSET 命令提高路径精度。使用 REFFLTSET 命令来减小平均化时间并提高路径精度。

此时，振动有增加的倾向，因此要小心确认运动。

---

## REFFLTRESET

---

### 功能

将机器人的平均化时间复位到默认值。

### 参数

将由 REFFLTSET 命令更改的平均化时间复位到默认值。

与 REFFLTSET 命令相同，当执行此命令时，在当前运动和下一运动命令之间的连续路径运动被中断。即，该两个运动不跟随在一个连续的运动。但，在进入第二运动之前，在第一运动结束时，机器人将暂时停止。

### 示例

在下面的样本程序中，从 p2 到 p3 的平均化时间为 X1，在 p3 和 p1 之间的平均化时间为默认值。

机器人运动程序

```
JMOVE p1  
LMOVE p2  
REFFLTSET X1  
LOMVE p3  
REFFLTRESET  
JMOVE p1
```

FFSET JT1 增益, JT2 增益, JT3 增益, JT4 增益, JT5 增益, JT6 增益, JT7 增益, JT8 增益, JT9 增益

功能  
设定机器人开始运动时的速度/加速前馈增益。

参数  
JT1 增益, JT2 增益, JT3 增益, JT4 增益, JT5 增益, JT6 增益, JT7 增益, JT8 增益, JT9 增益  
用实数值指定各轴的速度/加速前馈增益。容许范围：0 -1 （到小数点第三位有效）。除 JT1 轴以外可以省略。省略时，被设定如下。

当省略机器人轴的参数时：设定 JT1 的值。  
当省略外部轴的参数时： 设定为 JT1 的默认值与现在值的比，与省略的外部轴的默认值相乘的值。

[ 注 意 ]  
设定机器人轴的增益与 JT1 的值相等。用 FFSET\_STATUS 指令确认各参数的默认值和现在值。

示例  
当速度/加速前馈增益设定的越小，那么机器人的振动将减少，并且机器人将更顺利地运动。但，周期时间将变长，并且机器人有比示教的路径走捷径的倾向。  
  
另一方面，当该增益设定的越大，那么周期时间将缩短，机器人将比示教的路径走更精确的轨迹。但，机器人的振动有增加的倾向。

— [ 注 意 ] —

由于使用此命令将改变其动特性，因此通常不使用此命令。当使用此命令时，要遵循以下的步骤：

- 逐步更改设定值（以约 0.1 增值），并在更改后确认机器人的运动。
- 当确认机器人运动时，以约 20%的低速监控速度开始，并逐渐的提高速度。

此命令更改动特性；因此确认未突然更改开始和结束时的机器人速度。

— [ 注 意 ] —

执行此命令设定的值，适用于所有的机器人运动，并与 SPEED ALWAYS 命令相同。如果要复位默认值，则使用 FFRESET 命令。

例

FFSET 0.5            设定所有速度/加速前馈值为 0.5。

FFSET 0.5,0.49      除 JT2 外，设定所有轴的速度/加速前馈值为 0.5。设定 JT2 的速度/加速前馈值为 0.49。

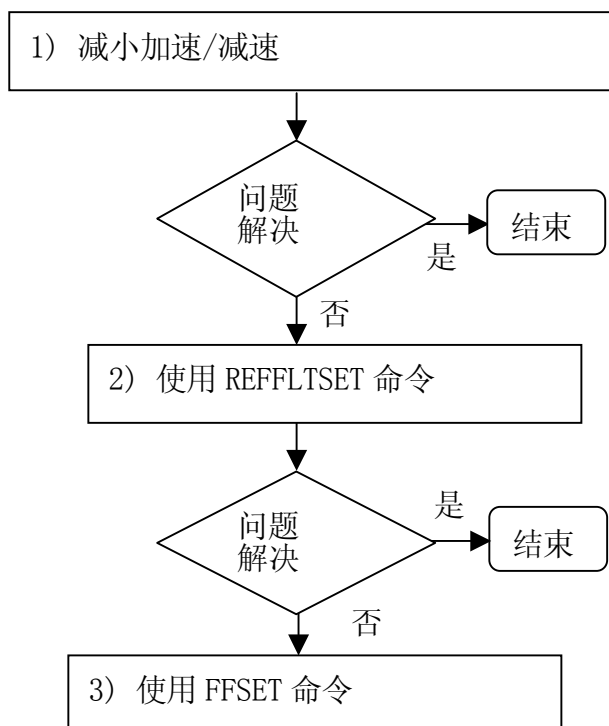
在下面的样本机器人运动程序中，在#p1 轴一致后，更改速度/加速前馈增益为 0.5。

```
JMOVE #p1
FFSET 0.5
JMOVE #p2
FFRESET
JMOVE #p3
```



遵循以下的步骤执行减少振动的对策：

- 1) 通过减小加减速来减少振动。  
遵循 REFFLTSET 命令所介绍的步骤。
- 2) 用 REFFLTSET 命令顺利进行机器人运动。  
遵循 REFFLTSET 命令所介绍的步骤。
- 3) 用 FFSET 命令顺利地运动。  
用 FFSET 命令设定速度/加速前馈增益来顺利进行机器人动作。  
执行 FFSET 命令来改变机器人的路径，因此要小心确认运动。



---

## FFRESET

---

### 功能

复位机器人速度/加速前馈增益到默认值。

### 说明

将由 FFSET 命令更改的机器人的速度/加速前馈增益复位到默认值。其与由 FFSET 命令设定默认值的方法相同。

### 示例

在下面的样本程序中，在#p1 轴一致后，更改速度/加速前馈增益为 0.5。在#p2 轴一致后，复位前馈增益为默认值。

```
JMOVE #p1  
FFSET 0.5  
JMOVE #p2  
FFRESET  
JMOVE #p3
```

### 6.3 夹紧控制命令

OPEN	在下一运动命令开始时，输出夹紧打开信号。
OPENI	在当前运动命令完成时，输出夹紧打开信号。
CLOSE	在下一运动命令开始时，输出夹紧闭合信号。
CLOSEI	在当前运动命令完成时，输出夹紧闭合信号。
RELAX	在下一运动命令开始时，关断夹紧信号。
RELAXI	在当前运动命令完成时，关断夹紧信号。
OPENS	在运动命令执行过程中，输出夹紧打开信号。 (选项)
CLOSES	在运动命令执行过程中，输出夹紧闭合信号。 (选项)
RELAXS	在运动命令执行过程中，关断夹紧信号。 (选项)
GUNON	开启射枪信号(变为 ON) ，用距离控制射枪输出延时。 (选项)
GUNOFF	关断射枪信号(变为 OFF) ，用距离控制射枪输出延时。 (选项)
GUNONTIMER	用计时控制射枪输出 ON 延时 。(选项)
GUNOFFTIMER	用计时控制射枪输出 OFF 延时 。(选项)

---

OPEN	夹紧编号
OPENI	夹紧编号

---

**功能**  
打开机器人夹紧(输出夹紧打开信号)。

**参数**  
夹紧编号  
指定夹紧的编号。若省略，默认为 1。

**说明**  
此命令输出信号给气动手爪的控制阀门，以打开夹紧。

对于 OPEN 命令，不输出信号，要到下一动作开始时才输出信号。

使用 OPENI 命令时的信号输出时序如下：

1. 如果机器人当前正在运动中，信号在运动结束后输出。如果机器人正在作 CP 运动，则 CP 动作暂停(相当于 BREAK 命令)。
2. 如果机器人不在运动中，信号被立刻送至控制阀。

**示例**

OPEN	当机器人开始下一运动时，将夹紧打开信号送至夹紧 1 的控制阀。
OPENI	一旦机器人完成当前运动时，立即将夹紧打开信号送至夹紧 2 的控制阀。

---

CLOSE	夹紧编号
CLOSEI	夹紧编号

---

功能

闭合机器人夹紧(输出夹紧闭合信号)

参数

夹紧编号

指定夹紧的编号。若省略，默认为 1。

说明

此命令输出信号给气动手爪的控制阀门，以闭合夹紧。

对于 CLOSE 命令，不输出信号，要到下一动作开始时才输出信号。

使用 CLOSEI 命令时的信号输出时序如下：

1. 如果机器人当前正在运动中，信号在运动结束后输出。如果机器人正在作 CP 运动，则 CP 动作暂停(相当于 BREAK 命令)。
2. 如果机器人不在运动中，信号被立刻送至控制阀。

示例

CLOSE 3                    当机器人开始下一运动时，将夹紧闭合信号送至夹紧 3 的控制阀。

CLOSEI                    一旦机器人完成当前运动时，立即将夹紧闭合信号送至夹紧 1 的控制阀。

---

RELAX	夹紧编号
RELAXI	夹紧编号

---

### 功能

将气动电磁阀的打开和闭合信号同时关断为 OFF (将夹紧信号变为 OFF, 在双头电磁阀中, 将夹紧打开和闭合信号都变为 OFF)。

### 参数

夹紧编号

指定夹紧编号, 如果省略, 默认为 1。

### 说明

对于 RELAX 命令, 信号要在下一运动命令开始时输出。

使用 RELAXI 命令时的信号输出时序如下:

1. 如果机器人当前正在运动中, 信号在运动结束后输出。如果机器人正在作 CP 运动, 则 CP 动作暂停 (相当于 BREAK 命令)。
2. 如果机器人不在运动中, 信号被立刻送至控制阀。

OPENS	夹紧编号
CLOSES	夹紧编号
RELAXS	夹紧编号

选项

**功能**  
将气动电磁阀的打开和闭合信号变为 ON 或 OFF。

**参数**  
夹紧编号  
指定夹紧的编号。如果省略，默认为 1。

**说明**  
此命令在以下方面不同于 OPEN/ CLOSE/ RELAX 和 OPENI/ CLOSEI/ RELAXI 命令：

- 1. OPEN/ CLOSE/ RELAX 命令：  
信号在下一运动命令开始时输出。
- 2. OPENI/CLOSEI/RELAXI 命令：  
如果机器人正在运动中，则信号在该运动完毕时输出。CP 运动暂停(相当于 BREAK 命令)。
- 3. OPENS/CLOSES/RELAXS 命令：  
此命令在执行后，信号立刻输出。

这些命令不受 PREFETCH. SIGINS 系统开关的影响。

---

<b>GUNON</b>	射枪编号,	距离
<b>GUNOFF</b>	射枪编号,	距离

---

选项

### 功能

将射枪信号变成 ON/OFF，并通过指定距离来控制射枪输出延时。

### 参数

射枪编号

指定射枪编号为 1 或 2。

距离

指定距离(单位: mm)，以调整射枪开/关的延时。负值提取延时，正值则延迟延时。如未指定，则默认值为 0。

### 说明

射枪信号是在 GUNON/GUNOFF 命令之后的运动命令执行时变为 ON/OFF 的。输出的延时取决于在此命令中指定的距离，输出时间由 GUNONTIMER/GUNOFFTIMER 命令设定。

### 示例

**GUNON 2, 100**

以 100 mm 延迟 ON 延时后，将射枪 2 的开信号变为 ON。



---

GUNONTIMER	射枪编号,	时间
GUNOFFTIMER	射枪编号,	时间

---

选项

功能

按指定的时间量调整射枪输出的延时(射枪开/关的时间)

参数

射枪编号

设定射枪编号为 1 或 2。

时间

指定时间量(单位：秒)，以调整射枪 ON/OFF 的延时。负值提前延时，正值则延迟延时。  
如果未指定，默认为 0 秒。

说明

调整时间取决于射枪系统的外部环境(如：阀门至射枪末端的距离、涂料类型、气候等)。  
因此请在程序开始时设置此延时。要在程序外部改变此延时，请为时间参数使用一个变量  
(例如，当该延时必须随涂料的颜色或粘性而改变时)。

本命令仅仅调整射枪的输出延时，并不实际控制射枪的开/关。

示例

GUNONTIMER 1, -0.5      提前射枪 1 开信号的延时 0.5 秒。

#### 6.4 形态命令

RIGHTY	改变形态，使机器人手臂像人的右手臂。
LEFTY	改变形态，使机器人手臂像人的左手臂。
ABOVE	改变形态，使肘关节处于上部位姿。
BELOW	改变形态，使肘关节处于下部位姿。
UWRIST	改变形态，使 JT5 的角度为正值。
DWRIST	改变形态，使 JT5 的角度为负值。

---

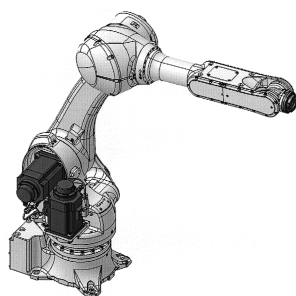
**RIGHTY**  
**LEFTY**

---

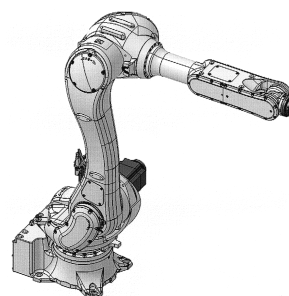
**功能**

强制机器人在下一运动过程中改变形态，使机器人的手臂像人的右臂 (RIGHTY) 或左臂 (LEFTY)。在直线插补运动过程中或者下一运动的目标是用关节位移值来表达的，那么机器人的形态不能改变。(参看 11.7 设置机器人形态)

**示例**



LEFTY



RIGHTY

---

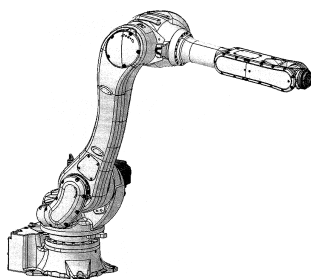
**ABOVE**  
**BELOW**

---

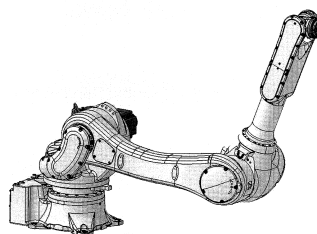
**功能**

强制机器人在下一运动过程中改变形态，使机器人的“肘关节” (关节 3) 可以像人的手臂一样，处于腕关节的上方或下方。在直线插补运动过程中或者下一运动目标是用关节位移值来表达的，那么机器人的构造不能改变。(参看 11.7 设置机器人形态)

**示例**



ABOVE



BELOW

---

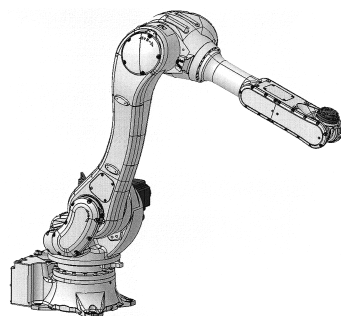
**UWRIST**  
**DWRIST**

---

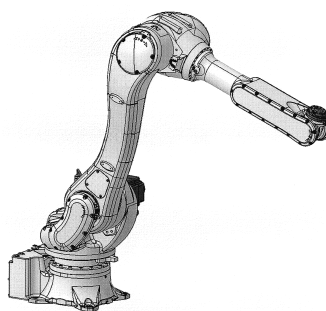
**功能**

强制机器人在下一步运动过程中改变形态，使机器人关节 5(JT5)的角度为正值或负值。在直线插补运动过程中或者下一运动目标是用关节位移值来表达的，那么机器人的构造不能改变。(参看 11.7 设置机器人形态)

**示例**



UWRIST  
(关节 5 为  $90^\circ$ )



DWRIST  
(关节 5 为  $-90^\circ$ )\*  
注释\* 关节 4 旋转了  $180^\circ$

## 6.5 程序控制命令

GOTO	跳转到指定的标签处。
IF	当指定的条件满足时，跳转到指定的标签处。
CALL	分支到一个子程序。
RETURN	返回子程序的调用程序。
WAIT	将程序的执行置于待机状态，直到条件满足。
TWAIT	将程序的执行置于待机状态，直到指定的时间流逝。
MVWAIT	将程序的执行置于待机状态，直到到达给定的位置或时间。
LOCK	改变机器人控制程序的优先级。
PAUSE	暂停程序执行。
HALT	停止程序执行(不可恢复)。
STOP	停止执行循环。
SCALL	分支到一个子程序。
ONE	在出错时调用程序。
RETURNE	从出错步骤后面的步骤开始执行。

---

GOTO 标签 IF 条件

---

**功能**  
跳转到带指定标签的程序步骤。

**参数**  
**标签**  
指定要跳转至的程序标签。标签为 0 至 32767 之间的任一整数。

**条件**  
指定跳转的条件，该参数和关键字 IF 可以省略。若省略，则程序在执行此命令时就跳转。

**说明**  
跳转至由标签指定的程序步骤。若指定了跳转条件，则程序在条件满足时跳转；若条件不满足，本命令继续向此命令的下一步骤执行。

请注意标签和步骤编号是不同的。程序所有的步骤编号都是由系统自动分配的，标签是有目的地在步骤编号后输入的。

当指定了条件时，此命令的功能与 IF GOTO 命令的功能相同。.

**示例**

GOTO 100	无条件跳转到标签 100 处。若无标签 100，则出错。
GOTO 200 IF n==3	当变量“n”等于 3 时，程序跳转至标签 200 处。如果不等于 3，则执行下一步骤。

---

## IF 条件 GOTO 标签

---

### 功能

当给定的条件满足时，跳转到带指定标签的程序步骤处。

### 参数

#### 条件

指定表达式条件，如： $n = 0$ 、 $n > 3$ 、 $m + n < 0$ 。

#### 标签

指定程序要跳转至的标签(不是步骤编号)。标签必须位于同一程序中。

### 说明

当条件满足时，程序跳转到由标签指定的步骤。如果条件不满足，则执行此命令后的步骤。

如果指定的标签不存在，则出错。

### 示例

<code>IF n&gt;3 GOTO 100</code>	如果整数变量“n”的值大于 3，程序跳转到带有标签 100 的步骤；如果“n”不大于 3，则执行此步骤后的步骤。
---------------------------------	--

<code>IF flag GOTO 25</code>	如果整数变量“flag”的值不等于 0，程序跳转至标签 25 所在的步骤；如果变量“flag” 的值等于 0，则执行此步骤后的步骤。此命令也可写为： <code>IF flag&lt;&gt;0 GOTO 25</code> 。
------------------------------	--

---

## CALL 程序名

---

### 功能

暂停当前程序的执行，并跳转至一个新的程序(子程序)。当子程序执行完毕后，返回原来的程序并继续执行 CALL 命令后的程序步骤。

### 参数

程序名

指定要执行的子程序名

### 说明

本命令暂停当前程序的执行，并跳转到指定子程序的第一步骤。

---

#### [ 注意 ]

---

同一个子程序不能被机器人控制程序和 PC 程序同时调用。同时，子程序也不能调用自己。

调用子程序时的程序嵌套最多可达 20 层。

### 示例

CALL sub1            跳入名为“sub1”的子程序，当执行到子程序“sub1”中的 RETURN 命令时，返回原来的程序并执行 CALL 命令后的程序。

---

## RETURN

---

### 功能

结束子程序的执行，并返回到子程序的调用程序中的 CALL 命令的下一步骤。

### 说明

此命令结束子程序的执行并返回到子程序的调用程序中。如果该子程序不是被其他程序调用的(如：当子程序是由 EXECUTE 命令执行时)，则程序执行结束。

在子程序的末尾处，即使没有 RETURN 命令，执行完子程序后，仍会回到原来的程序。但是，RETURN 命令还是应该被编写为子程序的最后步骤(或需要结束子程序的任何地方)。



---

## WAIT 条件

---

### 功能

使程序执行等待，直到指定的条件得到满足。

### 参数

#### 条件

指定待机(程序等待)条件。(实数表达式)

### 说明

此命令，在指定的条件被满足之前，暂停程序的执行。CONTINUE NEXT 命令可以在条件满足之前恢复程序的执行(跳过正在执行的 WAIT 命令)。

### 示例

**WAIT SIG (1001, - 1003)**      暂停程序的执行，直到外部输入信号 1001 (WX1)为 ON、  
并且 1003 (WX3)为 OFF。

**WAIT TIMER(1)>10**              暂停程序的执行，直到计时 1 的值超过 10(秒)。

**WAIT n>100**                      暂停程序的执行，直到变量“n”的值大于 100。(本例中，  
假设变量“n”的值是由 PC 程序或程序中断来累加的)

---

## **TWAIT 时间**

---

### **功能**

暂停程序执行，直到指定的时间流逝。

### **参数**

时间

以秒为单位指定暂停程序执行的时间。

### **说明**

此命令在指定的时间内，暂停程序的执行。

执行中的 TWAIT 命令，可以用 CONTINUE NEXT 命令来跳过。

WAIT 命令可用来代替 TWAIT 命令，获得的效果相同。

### **示例**

**TWAIT 0.5**                      等待 0.5 秒。

**TWAIT deltat**                      等待到变量“deltat”设置的时间流逝。

---

## MVWAIT 数值

---

### 功能

暂停程序的执行，直到当前运动的剩余距离(或时间)小于指定的距离(或时间)。

### 参数

数值

指定距离或时间，距离用毫米(mm)为单位表达，时间为秒(S)。若未指定单位，默认为毫米。

### 说明

此命令用来使程序执行与机器人动作同步。但是，请注意由于此命令根据指令值来监视剩余距离(或时间)，它可能由于响应滞后而不同于实际的剩余距离(或时间)。当机器人以关节插补动作移动时，指定的距离和实际距离可能相差较大。如果此命令执行时，当前运动已完成，将直接执行下一命令而不等待。可用 CONTINUE NEXT 命令跳过执行中的 MVWAIT 命令。

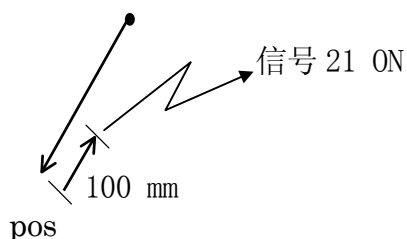
#### [ 注意 ]

MVWAIT 命令不能用在 PC 程序中。同时，此命令也不能与 DO 指令一起使用。

### 示例

在下图中，机器人向位姿“pos”移动，当其离“pos”小于 100 mm 时，信号 21 变为 ON。这种情况仅在系统开关 PREFETCH. SIGINS 为 ON(在轴一致前预读信号)时，以及机器人在精度范围之内时出现。

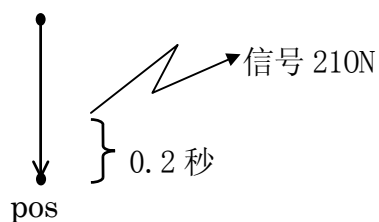
```
LMOVE pos  
MVWAIT 100 mm  
SIGNAL 21
```



### 示例

在下图中，机器人向位姿“pos”移动，当其离“pos”小于 0.2 秒时，信号 21 变为 ON。这种情况仅在系统开关 PREFETCH. SIGINS 为 ON(在轴一致前预读信号)时，以及机器人在精度范围之内时出现。

```
LMOVE pos  
MVWAIT 0.2 秒  
SIGNAL 21
```



---

## LOCK 优先级

---

### 功能

改变堆栈中当前被选中程序的优先级。

### 参数

优先级

用 0 至 127 之间的实数指定优先级。

### 说明

通常，机器人程序的优先级为 0，用此命令可改变优先级，数值越大优先级越高。

### 示例

LOCK 2                      把优先级改成 2。

---

## PAUSE

---

### 功能

暂时停止(暂停)程序的执行。

### 说明

此命令暂停程序的执行，并在终端上显示一条信息。可用 CONTINUE 命令恢复程序的执行。

此命令在检查程序时很方便。当 PAUSE 命令暂停程序时，可以检查各变量的值。

---

## HALT

---

### 功能

停止程序执行，此命令执行后，程序不能恢复运行。

### 说明

不管剩余步骤的情况如何，此命令停止程序的执行。在终端显示一条信息。

用此命令停止的程序不能用 CONTINUE 命令恢复运行。

---

## STOP

---

### 功能

结束当前的执行循环。

### 说明

如果有未完成的循环，则返回到第一步骤；否则结束执行。此命令标记了执行路径的结束，其作用与 HALT 命令不同。

如果还有未完成的循环，恢复执行时将从主程序的第一步骤开始\* (即使在子程序或另一个中断程序执行过程中，处理过 STOP 命令，也将返回到主程序)。

**注释\*** 主程序是用 EXECUTE、STEP、PCEXECUTE 命令执行的程序，子程序是被另一程序用 CALL、ON 或 ONI 命令调用的程序。

主程序中 RETURN 命令的功能和 STOP 命令的功能相同。

用 STOP 命令停止的程序，不能用 CONTINUE 指令恢复运行。

---

## SCALL 字符串表达式, 变量

---

### 功能

跳转到字符串表达式给出名称的子程序。

### 参数

字符串表达式

以字符串表达式形式指定子程序名。

### 变量

如果子程序调用正常执行，数值 0 被赋于此变量。如果在子程序调用中出现了某些异常，错误代码(≠ 0)被赋于此变量。如果省略，当在子程序调用中出现异常时，程序的执行将出错停止。

### 说明

除了文件名用字符串形式表达之外，此命令的功能与 CALL 命令的相同(见 CALL 命令)。

### 示例

```
$prog="sub1"
```

```
SCALL $prog
```

跳入名为“sub1”的子程序。

```
num=12
```

```
$temp1=$ENCODE (/12, num)
```

```
$temp2=""
```

```
FOR i=1 to LEN($temp1)
```

```
$temp3=$MID($temp1, i, 1)
```

```
IF $temp3<> "" THEN
```

```
$temp2=$temp2+$temp3
```

```
END
```

```
END
```

```
SCALL "sub"+$temp2
```

把赋给“num”的实数值转变成一个字符串表达式，并跳入名为“sub12”的子程序。

---

## ONE 程序名

---

### 功能

当出错时调用指定的程序。

### 说明

此命令在错误发生时调用指定的程序。可以从 PC 程序调用，即，可以调用指定的程序。

可用 RETURN(或 RETURNE) 命令返回至原来的程序。RETURN 命令返回到出错的步骤处。RETURNE 命令返回到出错步骤的后面步骤处。(如果程序中既没有 RETURN 也没有 RETURNE 命令存在，循环的执行将停止在被调用程序的尾部)

由 ONE 命令调用的程序中不能含有运动命令。

如果在由 ONE 命令调用的程序中出错，程序执行将在错误处停止。

### [ 注意 ]

只要包含了 ONE 命令的主程序在执行中，此命令也在子程序出错时有效，就像主程序一样。主程序停止执行时，ONE 命令也变成无效。

当错误出现时，如果有一个程序被 ONE 命令调用了，则错误指示灯就不会点亮。



---

## RETURNE

---

### 功能

返回到出错步骤的下一步骤。

### 说明

此命令一般和 ONE 命令配对使用。当错误发生时，ONE 命令使程序跳转到某子程序中，然后，当执行了子程序中的 RETURNE 命令后，返回到原来程序中出错步骤之后的步骤。

## 6.6 程序结构命令

IF.....THEN...ELSE.....END

WHILE.....DO.....END

DO.....UNTIL

FOR.....END

CASE.....OF.....VALUE.....ANY.....END

SCASE.....OF.....VALUE.....ANY.....END

---

```
IF 逻辑表达式 THEN
程序命令(1)
ELSE
程序命令(2)
END
```

---

### 功能

根据逻辑表达式的结果执行一组程序步骤。

### 参数

逻辑表达式

逻辑表达式或实数表达式，测试该值为“真”（非 0）或“假”（0）。

程序命令(1)

如果以上逻辑表达式为真，则执行这里输入的程序命令组。

程序命令(2)

如果以上逻辑表达式为假，则执行这里输入的程序命令组。

### 说明

根据逻辑表达式的值，该流程控制结构执行两组命令中的一组。执行流程如下：

1. 计算逻辑表达式，如果其结果为 0(假)，跳转到流程 4。
2. 计算逻辑表达式，如果其结果为 1(真)，执行程序命令(1)。
3. 跳转到流程 5。
4. 如果有 ELSE 语句，执行程序命令(2)。
5. 从 END 后面的步骤，继续执行程序。

#### [ 注意 ]

1. ELSE 和 END 语句必须各自独占一行。
2. IF...THEN 结构必须以 END 语句结束。

### 示例

在下例中，如果 n 大于 5，则将程序速度设定为 10 %，否则设定为 20 %。

```
21    IF n>5 THEN
22        sp=10
23    ELSE
24        sp=20
25    END
26    SPEED sp ALWAYS
```

下面的程序首先检查变量 “m” 的值，如果 “m” 不为 0，则程序检查外部输入信号 1001(WX1)，并根据信号的状态显示不同的信息。本例中，外面的 IF 结构没有 ELSE 语句。

```
71    IF m THEN
72        IF SIG(1001) THEN
73            PRINT"Input signal is TRUE"
74        ELSE
75            PRINT"Input signal is FALSE"
76        END
77    END
```

---

**WHILE 条件 DO**  
**程序命令**  
**END**

---

### 功能

当指定的条件为真(TRUE)时, 程序命令被执行。如果条件为假(FALSE), WHILE 语句被跳过。

### 参数

#### 条件

逻辑表达式或实数表达式, 测试该值为“真”(非 0)或“假”(0)。

#### 程序命令

指定一组在条件为真(TRUE)时执行的命令。

### 说明

当指定的条件为“真”时, 此流程控制命令重复执行给定的那组程序步骤。执行流程如下:

1. 计算逻辑表达式, 如果其结果为 0(假), 则跳转到流程 4。
2. 计算逻辑表达式, 如果其结果为 1(真), 则执行程序命令。
3. 跳转到流程 1。
4. 从 END 后面的步骤, 继续执行程序。

### [ 注意 ]

与 DO 结构不同, 如果条件为假, WHILE 结构内的任何程序步骤都不执行。

使用本结构时, 条件最终必须从 TRUE 变成 FALSE。

### 示例

下例中, 监视输入信号 1001 和 1002, 机器人运动根据它们的状态停止。当从两个送料机来的信号中一个输入信号变成 0(送料机为空)时, 机器人停止, 并且程序从 END 语句之后的步骤处继续执行(本例中的步骤 27)。

如果在 WHILE 结构开始执行时, 送料机之一为空(外部输入信号 OFF=0), 则不执行结构中的任何步骤, 程序流程跳转到步骤 27。

```
20      .  
21      .  
22      .  
23      WHILE SIG(1001, 1002) DO  
24          CALL part1  
25          CALL part2  
26      END  
27      .  
28      .  
29      .  
30      .
```

---

**D0**  
**程序命令**  
**UNTIL 逻辑表达式**

---

**功能**

创建一个 D0 循环。

**参数**

**程序命令**

只要逻辑表达式为假 (FALSE)，程序命令将重复执行。

**逻辑表达式**

逻辑表达式或实数表达式。当逻辑表达式结果变为真 (TRUE) 时，停止该结构中的程序命令的执行。

**说明**

该流程控制结构命令在给定的条件 (逻辑表达式) 为假时，执行一组程序命令。

执行流程如下：

1. 执行程序命令。
2. 检查逻辑表达式的值，如结果为假，则重复流程 1；如为真，跳转到流程 3。
3. 继续执行 UNTIL 语句后面的程序步骤。

当逻辑表达式的值从假变为真时，退出 D0 结构的执行。

— [ 注 意 ] —

与 WHILE 结构不同，D0 结构中的程序命令至少执行一次。

位于 D0 语句和 UNTIL 语句之间的程序命令可以省略。若没有命令，UNTIL 之后的逻辑表达式被重复地求值。当逻辑值变成真时，程序执行退出循环，继续执行 D0 结构之后的步骤。

D0 结构必须以 UNTIL 语句结束。

### 示例

在下例中，D0 结构控制以下任务：拾取一个零件，放到储料器中。当储料器满时，二进制输入信号“buffer.full”变为 ON。当信号 ON 时，机器人停止并开始另一个不同的操作。

```
10      .  
11      .  
12      .  
13      D0  
14          CALL get.part  
15          CALL put.part  
16      UNTIL SIG(buffer.full)  
17      .  
18      .  
19      .
```



---

**FOR 循环变量=起始值 TO 结束值 STEP 步进值**  
**程序命令**  
**END**

---

### 功能

重复程序执行。

### 参数

#### 循环变量

变量或实数值。该变量首先被设置了一个初始值，每次执行循环时加 1。

#### 起始值

实数或表达式。指定循环变量的初始值。

#### 结束值

实数或表达式。此值将和循环变量的当前值比较，如果循环变量的值达到了此值，则程序退出循环。

#### 步进值

实数或表达式，可以省略。在每次循环之后将循环变量加上或减去此值。使用 STEP 语句时输入此参数，除非循环变量以 1 为单位递增。如果步进值未指定，则将默认值 1 加到循环变量上；这种情况下，STEP 语句也可以省略。

### 说明

此流程控制结构命令重复执行位于 FOR 和 END 语句之间的程序命令。在每次循环执行之后，循环变量增加给定的步进值。

执行流程如下：

1. 赋给循环变量起始值。
2. 计算结束值和步进值。
3. 比较循环变量和结束值。
  - a. 如果步进值为正，且循环变量大于结束值，程序跳转到流程 7。
  - b. 如果步进值为负，且循环变量小于结束值，程序跳转到流程 7。上述 a 和 b 以外，程序跳转到流程 4。
4. 执行 FOR 语句之后的程序命令。
5. 当执行到 END 语句时，步进值被加到循环变量上。
6. 返回至流程 3。
7. 执行 END 语句之后的程序命令（在流程 3 中比较测试的循环变量的值不变）。

[ 注 意 ]

对于每个 FOR 语句必须有一个与之对应的 END 语句。

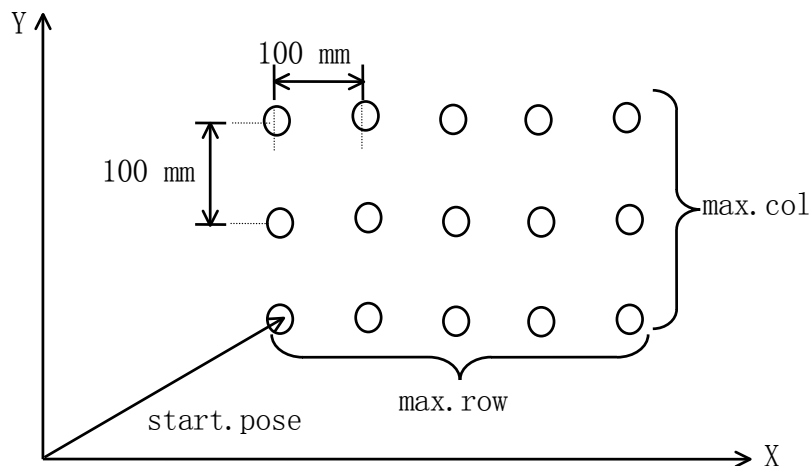
请注意，如果循环变量在首次检查时就大于(或小于，如果步进值为负)结束值，则将不执行 FOR 和 END 之间的程序命令。

循环次数值(循环变量)不允许由 FOR 循环中的其他程序(操作符、表达式等)改变。

示例

子程序“pick.place”拾取一个零件并放置到“hole”。零件放置方法如下图所示(托盘的放置平行于世界坐标系的 X、Y 轴，零件之间的距离为 100 mm)。

```
FOR row=1 TO max.row  
POINT hole=SHIFT (start.pose BY (row-1)*100, 0, 0)  
FOR col=1 TO max.col  
CALL pick.place  
POINT hole=SHIFT(hole BY 0, 100, 0)  
END  
END
```



---

```
CASE 索引变量 OF
VALUE 情况值 1, .....:
程序命令
VALUE 情况值 2, .....:
程序命令
:
VALUE 情况值 n, .....:
程序命令
ANY :
程序命令
END
```

---

### 功能

根据特定的情形编号执行程序。

### 参数

索引变量

实数变量和表达式。根据此索引变量的值来决定执行哪个 CASE 结构。

程序命令

当索引变量的值等于 VALUE 语句之后的某一个值时，执行这些程序命令。

### 说明

该结构使程序能从多个命令组中选择并执行选中的命令组。在 AS 语言中，它是一个强有力的工具，在一个程序中需作出多种选择时非常方便。

程序执行流程如下：

1. 检查 CASE 语句后输入的索引变量值。
2. 在 VALUE 步骤中检查并找出包含有与索引变量值相等的值的第一个步骤。
3. 执行该 VALUE 步骤之后的命令。
4. 继续执行 END 语句之后的命令。

如果没有与索引变量匹配的值，则执行 ANY 语句之后的程序命令。如果没有 ANY 语句，则不执行 CASE 结构中的步骤。

[ 注意 ]

ANY 语句和它的程序命令可以省略。

ANY 语句只能在结构中出现一次，且必须出现在结构的底部，如下例所示。

ANY 语句后的冒号“:”可以省略。输入冒号时，要在 ANY 后留一个空格。若没有空格，则 ANY: 将被视为一个标签。

ANY 和 END 语句必须独占一行。

示例

在下面的程序中，如果实型变量 x 为负数，程序执行在显示信息后停止；如果值为正数，程序按如下 3 种情况执行：

1. 如果值为 0 到 10 之间的偶数
2. 如果值为 1 到 9 之间的奇数
3. 如果值是上述之外的正数

```
IF x<0 GOTO 10
```

```
CASE x OF
```

```
VALUE 0, 2, 4, 6, 8, 10:
```

```
PRINT "The number x is EVEN"
```

```
VALUE 1, 3, 5, 7, 9:
```

```
PRINT "The number x is ODD"
```

```
ANY :
```

```
PRINT "The number x is larger than 10"
```

```
END
```

```
STOP
```

```
10 PRINT "Stopping because of negative value"  
STOP
```

```
SCASE 索引变量 OF
SVALUE 字符串_1, .....:
程序命令
SVALUE 字符串_2, .....:
程序命令
:
SVALUE 字符串_n, .....:
程序命令
ANY :
程序命令
END
```

选项

**功能**  
按照指定的字符串执行程序。

**参数**  
索引变量  
指定字符串变量或表达式。根据此变量的字符串来决定执行哪个 SCASE 结构。

**程序命令**  
当索引变量的字符串等于某个 SVALUE 语句之后的值时，则执行这些程序命令。

**说明**  
与前面所描述的 CASE 结构不同，SCASE 结构的执行条件用字符串设置，也请见 CASE 结构。  
  
如果没有与索引变量匹配的字符串，则执行 ANY 语句之后的程序命令。如果没有 ANY 语句，将不执行 SCASE 结构中的任何步骤。

[ 注 意 ]

ANY 语句和它的程序命令可以省略。

ANY 语句只能在结构中出现一次，且必须出现在结构的底部。

ANY 语句后的冒号 “:” 可以省略。输入冒号时，要在 ANY 后留有一个空格。  
若没有空格，ANY: 将被视为一个标签。

ANY 和 END 语句必须独占一行。

### 示例

在下面的程序中，如果字符串变量\$str 等于字符串\$a+” c” ，则执行程序 pc；如果字符串变量等于字符串\$a+” g”，则执行程序 pg。

```
SCASE $ s t r OF  
SVALUE $a+” c” :  
CALL pc  
SVALUE $a+” g” :  
CALL pg  
END
```

## 6.7 二进制信号命令

RESET	关断全部外部输出信号(变为 OFF)。
SIGNAL	开启(ON)/ 关断(OFF)外部 I/O 信号和内部信号。
PULSE	开启(ON)输出信号指定的时间量。
DLYSIG	在指定的时间过去之后, 改变信号。
RUNMASK	指定要屏蔽的信号。
BITS	设定一组信号来等于指定的值。(最大 16 信号)
BITS32	设定一组信号来等于指定的值。(最大 32 信号)
SWAIT	在满足指定的条件之前, 挂起程序的执行。
EXTCALL	调用由外部信号选定的程序。
ON	设定中断条件。
ONI	设定中断条件。
IGNORE	取消 ON 或 ONI 命令。
SCNT	在到达指定的计数器值时, 输出计数器信号。
SCNTRESET	清除计数器信号编号。
SFLK	在指定时间周期内, 开(ON)/ 关(OFF) 闪烁信号。
SFLP	根据 SET/RESET 信号, 开/关信号。
SOUT	当指定条件满足时, 输出信号。
STIM	当指定的信号为 ON 并持续了指定的时间时, 开启(ON) 计时信号。
SETPICK	设定开始夹紧闭合控制的时间。(选项)
SETPLACE	设定开始夹紧打开控制的时间。(选项)
CLAMP	控制夹紧信号的打开或闭合。(选项)
HSENSESET	开始对指定传感器信号进行监视。(选项)
HSENSE	读取由 HSENSESET 命令保存在缓冲区中的数据。(选项)

RESET

**功能**  
关断全部外部输出信号 (变为 OFF)。此指令对专用信号、夹紧信号和多功能 OX/WX 互斥信号没有影响。

利用选项设置，在接口面板屏幕上使用的信号将不受此命令的影响。(选项)

SIGNAL 信号编号, .....

**功能**  
开启 (ON) (或关断 (OFF)) 指定的外部信号 (OX) 或内部信号。

**参数**  
**信号编号**  
选择外部输出信号或内部信号的编号。如果信号编号是 “0”，则所有信号被关断 (OFF)。专用信号、夹紧信号和多功能 OX/WX 互斥信号不受此指令的影响。

容许的信号编号	
外部输出信号	1—实际信号的数量
内部信号	2001—2960

参看 5.7 SIGNAL 监控指令。



PULSE 信号编号, 时间

功能

开启 (ON) 指定的外部输出信号或内部信号，并保持给定的时间长度。

参数

信号编号

选择外部输出信号或内部信号的编号，选择专用信号会导致出错。

容许的信号编号	
外部输出信号	1—实际信号的数量
内部信号	2001—2960

时间

指定信号输出多久的时间(单位：秒)。若未指定，自动指定为 0.2 秒。

参看 5.7 PULSE 监控指令。

DLYSIG 信号编号, 时间

功能

在给定的时间过去后，输出指定的信号。

参数

信号编号

选择外部输出信号和内部信号的编号，如果信号编号为正数，信号变为 ON；如果为负数，信号变为 OFF。选择专用信号会导致出错。

容许的信号编号	
外部输出信号	1—实际信号的数量
内部信号	2001—2960

时间

指定延迟信号输出的时间，单位为秒。

参看 5.7 DLYSIG 监控指令。

**RUNMASK 起始信号编号, 信号的数量**

**功能**

仅当程序执行时允许信号为 ON。用 SIGNAL、PULSE 或 DLYSIG 命令可以开启这些信号为 ON，但是当程序执行停止时，这些信号将变为 OFF (如果未使用此命令，则信号一旦为 ON 后，将保持为 ON)。

**参数**

**起始信号编号**

指定需屏蔽的一组外部输出信号或内部信号中的首个信号的编号。输入一个负的编号将取消那个信号编号的屏蔽功能，且当程序停止时该信号将不会变为 OFF。

**容许的信号编号**

外部输出信号	1—实际信号的数量
内部信号	2001—2960

**信号的数量**

指定需要屏蔽多少个信号，若未指定，默认值为 1。

**说明**

当程序执行停止时，本命令选定的信号一直变成 OFF，但是，专用信号不受此命令的影响。

如果程序执行被中断，屏蔽的信号将变为 OFF。当用 CONTINUE 命令恢复程序执行时，信号将返回到当初程序运行时的状态。当用 DO 命令或 STEP 命令时会出现同样的情况 (用 EXECUTE 命令重新启动程序时将使 RUNMASK 命令失效)。

**示例**

**RUNMASK 5, 2**      屏蔽外部输出信号 5 和下一个信号 6，指定 2 个位。当程序运行时，这些信号可用 SIGNAL、PULSE、或 DLYSIG 命令变为 ON。当程序执行停止时，他们变为 OFF。

**BITS 起始信号编号, 信号的数量=十进制数值**

**功能**

以二进制格式安排一组外部输出信号或内部信号。根据与指定的十进制数值相等的二进制的值将这些信号的状态设定成 ON 或 OFF。

**参数**

起始信号编号

指定需设定信号状态的第一个信号。

容许的信号编号	
外部输出信号	1—实际信号的数量
内部信号	2001—2960

**信号的数量**

指定要设定成 ON/OFF 的信号的数量。最大允许值为 16，要设定的信号多于 16 个时，需用 BITS32 命令，说明见下节。

**十进制数值**

指定用于设定所要 ON/OFF 信号状态的十进制数值。十进制数值被转换成二进制数，二进制数值的每一位从最低位开始设定信号状态。如果该值的二进制数的位数多于信号的数量，仅有给定数量的信号(从指定的信号编号开始算起)被设定，其余的位被忽略。

参看 5.7 BITS 监控指令。

---

### **BITS32 起始信号编号, 信号的数量=数值**

---

#### **功能**

以二进制格式安排一组外部输出信号或内部信号。根据与指定的十进制数值相等的二进制的值将这些信号的状态设定成 ON 或 OFF。如果数值未指定, 则显示这些信号的当前状态。

#### **参数**

起始信号编号

指定需设定信号状态的首个信号。

信号的数量

指定被设为 ON/OFF 的信号的数量。最大允许值为 32。

数值

指定用于设定所要的 ON/OFF 信号状态的十进制数值。十进制数值被转换成二进制符号, 二进制数值的每一位指定信号的状态。二进制数值的最低位对应于最小的信号编号, 依此类推。如果该值的二进制符号的位数多于信号的数量, 仅有给定数量的信号(从指定的信号编号开始算起)被设定, 其余的位被忽略。

#### **说明**

根据给定的值, 置位(或复位)一个或多个外部输出信号或内部信号。

许用信号编号	
外部输出信号	1—实际信号的数量
内部信号	2001—2960

当设置的信号编号大于实际安装的信号数量时会导致错误。选择了专用信号也会产生错误。

参看 5.7 BITS、BITS32 指令。

SWAIT 信号编号, .....

**功能**  
在指定的外部 I/O 信号或内部信号满足置位条件之前，一直等待。

**参数**  
**信号编号**  
指定要监视的外部 I/O 信号或内部信号的编号，负编号表示当信号为 OFF 时条件满足。

容许的信号编号	
外部输出信号	1—实际信号的数量
内部信号	2001—2960

**说明**  
如果全部指定的信号都满足指定的条件时，此命令结束，程序执行下一步骤。如果条件不满足，程序一直等在那一步骤处，直至条件满足。

可以用 CONTINUE NEXT 命令，跳过执行中的 SWAIT 命令。

用 WAIT 命令可以达到相同的效果。

<b>示例</b>		
SWAIT	1001, 1002	在外部输入信号 1001 (WX1) 和 1002 (WX2) 都变成 ON 之前，一直等待。
SWAIT	1, -2001	在外部输出信号 1 (OX1) 为 ON 和内部信号 2001 (WX1) 为 OFF 之前，一直等待。

EXTCALL

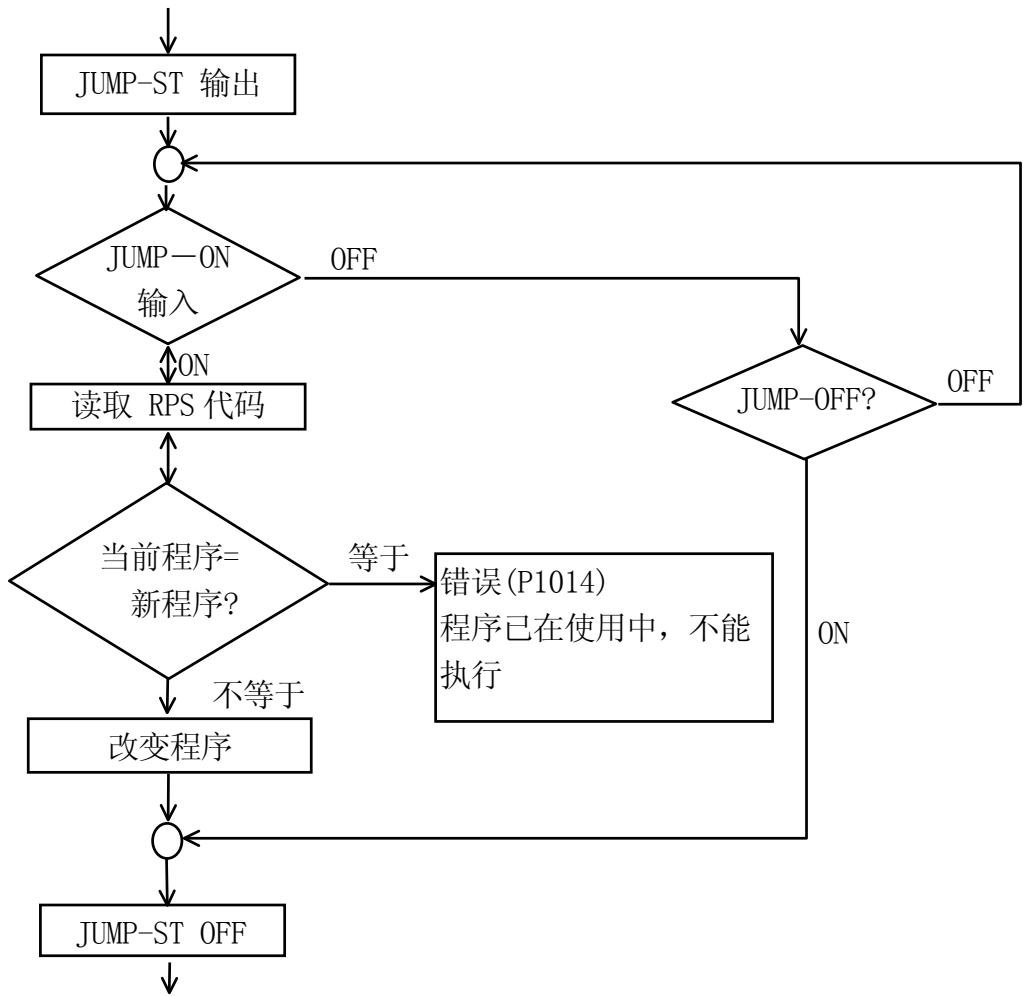
功能

调用由外部输入信号选定的程序。

说明

EXTCALL 命令的执行过程，如以下流程图所示：

- 1. 输出 JUMP-ST 信号，允许输入一个外部程序。
- 2. 等待 JUMP-ON 信号的输入。
- 3. 当 JUMP-ON 输入后，读入由 RPS-CODE (RPS 代码) 输入的程序编号。如果输入的编号为 100 或更大，程序 pgxxx 被调用。如果编号为 99-10，程序 pgxx 被调用，编号小于 9，则调用 pgx 。



— [ 注 意 ] —

在等待 JUMP\_ON 信号时，输入 CONTINUE NEXT 指令，可以跳过本命令的执行。

仅当 RPS 模式为 ON 及 RPS 信号被设置成软件专用信号时，此命令才有效。当 RPS 未被定义为专用信号时，如果执行此命令将产生一个错误。

如果 RPS 模式为 OFF，此命令将被忽略(不执行)。

EXTCALL 用于调用子程序。完成该子程序后(或当子程序中执行了一条 RETURN 命令)，将返回原来程序执行。

ON	模式	信号编号	CALL	程序名, 优先级
ON	模式	信号编号	GOTO	标签, 优先级
ONI	模式	信号编号	CALL	程序名, 优先级
ONI	模式	信号编号	GOTO	标签, 优先级

功能

监视指定的外部输入信号或内部信号，且在信号输入的同时，分支到指定的子程序 (CALL)，或跳转到指定的标号处 (GOTO)。

ONI 停止当前运动命令；而 ON 在跳转到子程序或标号之前，将等待当前运动完成。

参数

模式	
(不指定)	监视指定信号的上升沿和下降沿。
/ERR (选项)	如果信号的状态在监视开始时已经满足指定的条件，则返回一个错误。
/LVL (选项)	如果信号的状态在监视开始时已经满足指定的条件，则立即跳转到指定的子程序或标号处。

信号编号

指定要监视的信号编号

如果编号为正，将监视信号的上升沿或从 OFF 到 ON 的变化。如果编号为负，则监视下降沿或从 ON 到 OFF 的变化。

容许的信号编号	
外部输入信号	1001—实际信号的数量
内部信号	2001—2960

程序名称

指定当输入指定信号时执行的子程序的名称。若省略，程序继续执行下一步骤，而不转到子程序中。

标签

指定当输入指定信号时所要跳转到的标签。

优先级

设置程序的优先级，范围：1 到 127。若未指定，默认值为 1。编号越大，优先级越高。当输入了标签作为跳转目标时，优先级将被忽略。



### 说明

对于 ON...CALL 命令，如果检测到被监视信号有所改变，则程序中断而执行指定的子程序。该功能与在检测到被监视信号之后执行的 CALL 命令是相同的(参看 6.5 CALL 程序命令)。

如果在被调用的子程序中执行了 RETURN 命令，程序的执行将返回到子程序调用之前执行的程序步骤后面的那一步骤(参看 11.3 外部联锁)。

ONI 命令只能用在机器人运动程序中，而不能在 PC 程序中使用。

在任一如下情况下将取消信号监视：

1. 对 ON 和 ONI 命令中的信号执行了 IGNORE 命令。
2. ON 和 ONI 命令已执行，且程序已跳转到子程序。
3. 一个新的 ON 或 ONI 命令指定了与以前 ON (ONI)命令相同的信号(旧的设置被取消)。

### [ 注意 ]

1. 当监视信号的上升沿和下降沿时，程序仅在信号状态改变时执行分支。因而，要检测信号的边界时，如果信号在 ON 命令执行时已经为 ON，则不执行分支。在信号变为 OFF，然后再一次变为 ON 之前，程序不执行分支。
2. 为了精确检测信号的改变，信号至少需要稳定 50 毫秒。
3. ON (ONI) 命令一经执行，监视就马上开始。因为在 AS 系统中，将非运动命令与之前的运动命令一起读入并执行，信号的监视与紧跟在 ON (ONI)命令执行之前的机器人运动同时开始。
4. 当程序未执行时，信号不会被监视。

### 示例

**ONI -1001 CALL alarm**

监视外部输入信号 1001(WX1)。一旦信号从 ON 变成 OFF (信号编号为负数，故检测下降沿)时，运动停止，并执行“alarm”子程序。

**ON test CALL delay**

监视变量“test”中分配的信号。如果信号如期改变(改变条件取决于“test”中的值，因为它有可能为负数，也可能为正数)，在当前运动步完成后，程序执行“delay”子程序。在“delay”子程序执行完毕后，返回至原来程序。

IGNORE 信号编号

**功能**  
取消 ON 或 ONI 命令设置的信号监视。

**参数**  
信号编号：  
指定要取消监视的信号的编号。

容许的信号编号	
外部输入信号	1001—实际信号的数量
内部信号	2001—2960

**说明**  
此命令使最近的 ON 或 ONI 命令指定的信号无效(参看 11.3 外部联锁)。

[ 注 意 ]

ON(ONI) 命令的监视功能仅在实际安装的二进制 I/O 信号作为输入信号时有效。

**示例**

IGNORE 1005      取消对外部输入信号(通道 5)的监视。

IGNORE test      取消对由变量“test”的值指定的信号的监视。

---

**SCNT 计数器信号编号=计数增加信号, 计数减少信号, 计数器清除信号, 计数器值**

---

**功能**

当到达指定的计数值时, 输出计数器信号。

**参数**

**计数器信号编号**

指定要输出的信号编号。计数器信号编号的设置范围是: 3097 到 3128。

**计数增加信号**

用信号编号或逻辑表达式来指定。信号每次从 OFF 变成 ON, 计数器就加计数 1。

**计数减少信号**

用信号编号或逻辑表达式来指定。信号每次从 OFF 变成 ON, 计数器就减计数 1。

**计数器清除信号**

用信号编号或逻辑表达式来指定。如果该信号变为 ON, 则内部计数器被复位为 0。

**计数器值**

当内部计数器达到此值时, 输出指定的计数器信号。如果给定值为 0, 信号变为 OFF。

**说明**

当执行 SCNT 命令时, 如果计数增加信号从 OFF 变成 ON, 则内部计数器值加 1。如果计数减少信号从 OFF 变成 ON, 则内部计数器数值减 1。当内部计数器数值达到在参数中指定的值(计数器值)时, 则输出计数器信号。如果输出了计数器清除信号, 内部计数器的数值被设置成 0。每个计数器信号有其自己的计数器值。用 SCNTRESET 命令可以强制内部计数器清 0。

用 IO/E 命令可检查信号 3001 到 3128 的状态。(选项)

参看 5.7 SCNT 监控指令。

---

### SCNTRESET 计数器信号编号

---

#### 功能

将与指定的计数器信号对应的内部计数器数值置为 0(复位)。

#### 参数

计数器信号编号

选择要复位的计数器信号编号。计数器信号编号的指定范围是：3097 到 3128。

参看 5.7 SCNTRESET 监控指令。

---

### SFLK 信号编号=时间

---

#### 功能

在指定的时间周期内，将指定的信号变成 ON/OFF(闪烁)。

#### 参数

信号编号

指定要闪烁的信号的编号，设置的范围是：3065 到 3096。

时间

指定 ON/OFF 周期变化的时间(实数值)。如果设置成负数值，则取消闪烁。

#### 说明

ON/ OFF 变化的过程被看作为一个周期，此周期在指定的时间内执行。

参看 5.7 SFLK 监控指令。

---

### **SFLP 输出信号=置位信号表达式, 复位信号表达式**

---

#### **功能**

用一个信号置位表达式和一个信号复位表达式，将输出信号变为 ON 或 OFF。

#### **参数**

##### **输出信号**

指定输出信号的编号。正数使信号变为 ON，负数使信号变为 OFF。只能指定输出信号(范围：1 到实际信号的数量)。

##### **置位信号表达式**

指定来置位输出信号的信号编号或逻辑表达式。

##### **复位信号表达式**

指定来复位输出信号的信号编号或逻辑表达式。

#### **说明**

如果置位信号为 ON，则输出信号变为 ON。如果复位信号为 ON，则输出信号变为 OFF。若置位信号和复位信号都为 ON, 则输出信号为 OFF。输出信号是在执行 SFLP 命令时变为 ON 或 OFF 的，而不是在置位信号或复位信号变为 ON 的时候。

参看 5.7 SFLP 监控指令。

SOUT 信号编号=信号表达式

**功能**  
当指定的条件满足时，输出指定的信号。

**参数**  
**信号编号**  
 指定要输出的信号的编号，只能指定输出信号(范围：1 到实际信号的数量)。

**信号表达式**  
 指定一个信号编号或逻辑表达式。

**说明**  
 此命令用于信号的逻辑计算。将使用逻辑表达式，如：AND 和 OR。当指定的条件满足时，输出指定的信号。(参看 5.7 SOUT 监控指令。)

**示例**

SOUT 1 = 1001 AND 1002

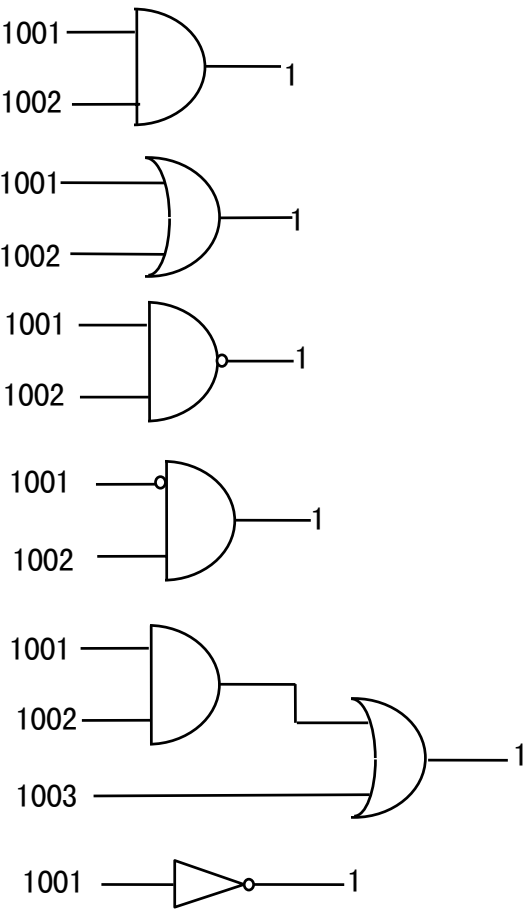
SOUT 1 = 1001 OR 1002

SOUT -1 = 1001 AND 1002  
 SOUT 1 = NOT(1001 AND 1002)

SOUT 1 = -1001 AND 1002

SOUT 1 = (1001 AND 1002) OR 1003

SOUT -1 = 1001 或 SOUT 1= -1001 或  
 SOUT 1 = NOT(1001)



**STIM  计时信号=输入信号编号, 时间**

**功能**  
如果指定的输入信号为 ON 并保持给定的时间，则计时信号变成 ON。

**参数**  
**计时信号**  
选择要变成 ON 的信号编号，设定范围：3001 到 3064。

**输入信号编号**  
指定整数形式的输入信号编号或逻辑表达式，作为将计时信号变成 ON 的条件来监视。该值不能超过实际安装的信号编号。

**时间**  
以实数值形式指定输入信号为 ON 的时间(单位：秒)。

参看 5.7 STIM 监控指令。

SETPICK	时间 1, 时间 2, 时间 3,  ...,  时间 8
SETPLACE	时间 1, 时间 2, 时间 3,  ...,  时间 8

选项

**功能**  
为 8 个夹紧的每一个，设定开始夹紧闭合控制(SETPICK)或夹紧打开控制(SETPLACE)的时间。

**参数**  
**时间 1 到 8**  
以秒为单位设定打开/闭合夹紧 1 到 8 的控制时间，设定范围为：0.0 到 10.0 秒。

**说明**  
参看 CLAMP 命令。

---

CLAMP	夹紧编号 1, 夹紧编号 2, 夹紧编号 3, 夹紧编号 4, ....., 夹紧编号 8,
-------	--

---

选项

### 功能

输出由参数夹紧编号 x 指定的手爪的打开/闭合的夹紧信号。用 SETPICK/SETPLACE 命令设定输出的时间；也就是说，当前运动完成之前 x 秒，输出信号。

### 参数

夹紧编号 1 到 8

指定夹紧编号。如果编号为正，机器人手爪打开；如果编号为负，机器人手爪闭合。

### 说明

此命令输出打开和闭合气动手爪控制阀的信号。若机器人当前不在运动中或剩余的运动时间少于 SETPICK/SETPLACE 命令设定的时间时，立即输出信号。当轴一致时，如果下一运动在 SETPICK/SETPLACE 命令设定时间到达之前开始，造成动作重叠，则输出信号。如果命令设定不合理，如：“CLAMP 1, -1”，后面的夹紧编号将有效。

### 示例

```
12 SETPICK 4, 3, 2, 1
13 SETPLACE 0.2, 0.4, 0.6, 0.8
14 LMOVE a
15 CLAMP -1, 2, 3, -4
```

执行以上程序，机器人将动作如下：

到达位姿 a 之前 3 秒钟，夹紧 2 闭合。

到达位姿 a 之前 2 秒钟，夹紧 3 闭合。

到达位姿 a 之前 0.8 秒钟，夹紧 4 打开。

到达位姿 a 之前 0.2 秒钟，夹紧 1 打开。



---

**HSENSESET 监视编号=输入信号编号, 输出信号编号, 信号输出耽误的时间**

---

选项

**功能**

向 AS 系统声明信号检测的开始。当此命令执行时, AS 系统开始监视传感器信号并在信号变化时将位姿等数据积存到缓存中。保存在缓存中的数据可以用 HSENSE 命令来读取。缓存最多可以保存 20 个数据。

**参数**

**监视编号**

指定监视编号, 最多可以监视两个输入信号。每个信号的命令写作 HSENSESET 1 或 HSENSESET 2。

**输入信号编号**

设定要监视的信号编号。设置零(0)停止监视。

**输出信号编号**

指定在系统取得关节角度后, 要输出的信号编号。指定的信号 ON 0.2 秒, 本项也可省略。

**信号输出耽误的时间**

设定在取得位姿数据后, 信号输出耽误的时间。设定范围是 0 到 9999 ms。可以省略。

— [ 注 意 ] —

即使在观察过程中控制电源被关断, 缓冲区里仍将保留读取的数据。在再次开启控制电源后, 仍可以用 HSENSE 命令读取被保留的数据。但是监视不会自动重新开始, 因此必须再次执行 HSENSESET。

请参看 5.7 HSENSESET 监控指令。

**示例**

HSENSESET 1=wx\_sens            开始监视输入信号 wx\_sens。

---

<b>HSENSE</b>	<b>监视编号</b>	<b>结果变量,</b>	<b>信号状态变量,</b>	<b>位姿变量,</b>	<b>错误变量,</b>	<b>内存使用量变量</b>
---------------	-------------	--------------	----------------	--------------	--------------	----------------

---

选项

## 功能

读取由 HSENSESET 命令保存在缓存中的数据。

## 参数

### 编号

设定监视器编号。要读取由 HSENSESET 1 保存的数据, 请指定 HSENSE1。要读取由 HSENSESET 2 保存的数据, 请指定 HSENSE 2。

### 结果变量

为监视结果指定实型变量。在 HSENSE 命令执行后, 数值将被保存到此变量中。当 AS 系统没有检测到信号变化时, 将零(0)保存到此变量中。当 AS 系统检测到信号变化时, 将-1 保存到此变量中。

### 信号状态变量

指定实型变量或数组变量, 用于保存信号变化的状态。在 HSENSE 命令执行后, 数值被保存到此变量中。当信号从 OFF 变为 ON 时, ON (-1)被保存在此变量中。当信号被从 ON 变为 OFF 时, OFF (0)被保存在此变量中。

### 位姿变量

在信号被检测到时, 为保存位姿数据指定关节位移值变量或数组变量。

### 错误变量

指定缓冲器溢出出错时的实型变量。如果没有错误, 将 0 保存在变量中。当缓冲存储器溢出时, 将把一个数值(非 0)保存在此变量中。缓存在积存数据超过了 20 个后溢出。

### 内存使用量变量

指定实型变量名, 用来保存缓冲器中内存使用量。保存在此变量中的值显示了缓冲器中的已使用的内存量。当仅使用了一个内存, 保存为 0。当使用了所有的内存时, 变量值保存为 19。

### 示例

在此程序中，当机器人从#p2\_1 运动到 #p4\_1 时，监视传感器信号 wx\_sense，并在检测到该信号时将 JT3 的位姿数据保存在数组 hsens\_jt[ ] 中。该程序用到了许多局部变量(局部变量的名称以“.”开始)。

```
.err=0                                ;初始化
IF SIG(wx_sense) THEN                 ;当传感器信号保持 ON
  .err=4                               ;错误的开始点
  RETURN
END
;
HSENSESET 1=wx_sense                  ;开始监视

JMOVE #p2_1                           ;移动到开始点
BREAK
SPEED sens_sp
ABS. SPEED ON
JMOVE #p4_1                           ;移动结束点

.num=0
loop:
HSENSE 1 .stat,hsens_onoff[.num+1],#hsens[.num+1] ,.serr,.rest
IF .serr <> 0 THEN
  .err = 3                             ;缓冲区存储器溢出”(HSENS)”
  RETURN
END
IF .stat==ON THEN
  hsens_jt[.num+1]=DEXT(#hsens[.num+1],3) ;检测到信号时，保存 Z 方向上的位姿。
  .num = .num + 1
END
IF .rest GOTO loop                    ;当缓冲区保存了数据后，再循环。
IF DISTANSE (DEST, HERE) > 0.1 GOTO loop
;
HSENSESET 1=0                         ;结束监视
```

## 6.8 信息控制命令

PRINT	在终端上显示信息。
TYPE	在终端上显示信息。
PROMPT	在终端上显示信息，并等待键盘的输入。
IFPWPRINT	在显示窗口中显示指定的字符串。
IFPLABEL	设定接口控制板上图标的标签。
IFPTITLE	设定接口控制板上指定页的标题。
SETOUTDA	指定模拟量输出环境。(选项)
OUTDA	按照设定的条件输出电压。(选项)

---

PRINT	设备编号: 打印数据, .....
TYPE	设备编号: 打印数据, .....

---

## 功能

在终端上显示在参数中指定的打印数据。

## 参数

### 设备编号

从下面选择显示数据的设备:

- 1: 个人计算机
- 2: 示教器

如果未指定, 数据将显示在当前选中的设备上。

### 打印数据

选择以下一个或多个打印数据。有多个数据时, 用逗号分开。

- |                        |              |
|------------------------|--------------|
| (1) 字符串                | 如: "count =" |
| (2) 实数表达式 (先计算数值, 再显示) | 如: count     |
| (3) 格式信息 (控制输出信息的格式)   | 如: /D, /S    |

不指定参数时, 将显示空行。

## 说明

参看 5.8 PRINT/TYPE 监控指令。

### [ 注意 ]

如果 MESSAGES 系统开关为 OFF, 终端屏幕上将不显示任何信息。

---

**PROMPT**    **设备编号:** 字符串, 变量

---

### 功能

在终端显示指定的字符串，并以提示符”>”结尾，等待键盘的输入。

### 参数

设备编号

从下面选择显示数据的设备：

- 1: 个人计算机
- 2: 示教器

如果未指定，数据将显示在当前选中的设备上。


字符串

指定要在终端上显示的字符。

变量




指定接收键盘输入数据的变量。它可以是一组实型变量或一个单字符串变量。

### 说明

指定的字符串显示在终端上，并等待从键盘输入数据和  (回车键)。

输入的数据按以下方式之一进行处理。

1. 当 PROMPT 用于为一组实型变量输入数值时，系统将读取的一行以空格或逗号分开的一组数字的输入数据。每个输入数字根据其符号转换成内部表达，然后一个一个地分配给变量。
2. 如果输入数值的数量多于变量的数量，多出的数值将被忽略。如果输入数值的数量小于变量的数量，将把“0”赋值给余下的变量。若输入了非数值型数据，将产生一个错误，且程序停止执行。为了避免混乱和出错，建议用一个 PROMPT 命令只为一个变量分配一个数值。

当用一个字符串变量作为 PROMPT 的变量参数时，输入的字符被读作一个单独的数据单位，所有的字符都被赋值给字符串变量。在屏幕提示符后，如果输入了  键或  +  键，实型变量将被赋值为“0”，字符串变量将被赋值为空字符串。

若输入的设备编号为“2”，示教器将自动切换到键盘画面。按<下一页>可返回到常规画面。

### 示例

引号内的字符串将显示在终端上，并要求输入数据。当输入了数据(零件数量)并按下  键后，输入的数值被赋给变量 “part.count” 。然后程序继续执行。

```
PROMPT "Enter the number of parts: ", part.count
```

以下命令提问输入一个字符串变量值，可以输入文字数字字符而不会引起出错。

```
PROMPT "Enter the number of parts: ", $input
```

IFWPRINT 窗口, 行, 列, 背景颜色, 标签颜色=“字符串”, “字符串”, .....

选项

功能

在辅助功能 0509 接口面板中设置的字符串窗口中显示指定的字符串。

参数

窗口

选择显示字符串的窗口。从辅助功能 0509 中的 1—8(标准)中选择。

行

指定在选定窗口中的哪一行上显示字符串。输入行号为 1 至 4；可设置的行号取决于窗口尺寸大小。若不指定，默认值为 1。

列

指定在选定窗口中的哪一列上显示字符串。 输入列号为 1 至 70；可设置的列号取决于窗口尺寸大小。若不指定，默认值为 1。

背景颜色

选择所选窗口的背景颜色。颜色编号从 0 至 15。若不指定，背景为白色。

No.	颜色	No.	颜色	No.	颜色	No.	颜色
0	灰	4	绿	8	粉红	12	深蓝
1	蓝	5	淡蓝	9	白	13	红棕
2	红	6	黄	10	黑	14	深绿
3	橙	7	白	11	蓝绿	15	淡紫

标签颜色

选择要显示的字符的颜色。颜色编号从 0 至 15。若不指定，字符显示为黑色。

字符串

指定要显示的字符串。第一个字符串后的其余各字符串从下一行的指定列开始显示。IFWPRINT 命令的执行将清除指定窗口中的非显示区域。

说明

IFWPRINT 指令仅在接口面板可用时才能使用。如果参数未指定，则选用该特定窗口的上次设置。(如果是第一次使用，则设置为上述默认值)。若字符串长度大于一行，超出部分在下一行显示(按所选的列缩进)。超出窗口尺寸的部分不显示。字符串中的控制字符将显示为空格。



---

IFPLABEL	位置, “标签 1”, “标签 2”, “标签 3”, “标签 4”
----------	------------------------------------

---

### 功能

设定并更改在指定位置的接口控制板图标的标签。

### 参数

#### 位置

指定要设定・更改的标签的接口控制板图标的表示位置。设定范围：1～112。

“标签 1”, “标签 2” …

指定标签上要表示的字符串。不更改的标签可以省略。

### 说明

设定・更改接口控制板上表示的图标的标签。当指定图标的没被设定的位置, 或没有标签的图标时, 什么都不设定。

参看 5.8 IFPLABEL 监控指令。

---

**IFPTITLE 页码, "标题"**

---

**功能**

指定并更改接口控制板的指定页的标题。

**参数**

页码.

指定要变更标题的接口控制板的页码。设定范围：1～4。

**“标题”**

指定指定页中的标题要表示的字符串。默认值设定为“接口控制板”。当指定 NULL 字符（“”）时，也表示为此默认值。

**说明**

设定・更改接口控制板的指定页所表示的标题。

参看 5.8 IFPTITLE 监控指令。

---

**SETOUTDA 端口编号=LSB, 位数, 逻辑, 最高电压, 最低电压**

---

选项

### 功能

指定模拟量输出环境：端口编号和 LSB(最低位)、位数、信号输出逻辑电压、最大和最小电压。

### 参数

#### 端口编号

指定端口编号。设置范围：1 到 10 之间的整数。

#### LSB

用整数设定 D/A 转换输出的首信号编号。设定范围：OUT 1—OUT 125、2001—2125、3000 (1TW 端口的第一频道)、3001 (1TW 端口的第二频道)，以后增值的 1TW 端口的 DA 频道数。默认值为 3000。如果没有任何指定，前一次的设定值继续有效。

#### 位数

用整数设定 D/A 转换信号的位数。设定范围：4 位～16 位。当上面的参数 LSB 设定为 3000～[3000+1TW 端口的 DA 频道数]时，请设定 12 位。默认值为 8 位。如果没有任何指定，前一次的设定值继续有效。

#### 逻辑

设定输出数据是正逻辑(1)还是负逻辑(0)。默认值为 0 (负逻辑)。如果不指定，前一次的设定继续有效。请为 1TW 设定为正逻辑。

#### 最高电压

指定硬件(D/A 输出)的最高电压。设定范围：-15—+15V。单位：V。默认值为 13V。化整到第一小数位。如果不指定，前一次的设定继续有效。

#### 最低电压

指定硬件(D/A 输出)的最低电压。设定范围：-15—+15V。单位：V。默认值为 0V。化整到第一小数位。如果不指定，前一次的设定继续有效。

### 〔 注 意 〕

1. 实际的输出电压依存于所使用的硬件。
2. 如果最高电压设定得比最低电压低时，将报出错。

---

**OUTDA 电压, 端口编号**

---

选项

**功能**

按照设定的条件向指定端口编号输出电压。

**参数**

**电压**

指定从 D/A 端口输出的电压。设定范围：-15—+15V。单位：0.1V。化整到第一小数位。

**端口编号**

指定端口编号。设定范围：1 到 16 之间的整数。如果未指定，默认为“1”。

———— [ 注 意 ] ————

请确认指令电压和实际输出电压一致，通过 SETOUTDA 命令(指令)将输出环境设置成与硬件设置相一致。

## 6.9 位姿信息命令

HERE	以当前位姿定义一个位姿变量。
POINT	定义一个位姿变量。
POINT/X	设定变换值变量的 X 值。
POINT/Y	设定变换值变量的 Y 值。
POINT/Z	设定变换值变量的 Z 值。
POINT/OAT	设定变换值变量的 OAT 值。
POINT/O	设定变换值变量的 O 值。
POINT/A	设定变换值变量的 A 值。
POINT/T	设定变换值变量的 T 值。
POINT/7	设定变换值变量的第 7 轴的值。
DECOMPOSE	把位姿信息的组成部分赋值给数组变量的元素。
BASE	改变基础坐标系变换值。
TOOL	定义工具变换值。
SET_TOOLSHAPE	设定工具形状的数据。
ENA_TOOLSHAPE	根据工具形状设定速度控制的有效/无效。
TOOLSHAPE	根据工具形状设定速度控制的数据。
SETHOME	设定原点位姿。
SET2HOME	设定原点位姿 2。
ULIMIT	设定机器人运动的上限。
LLIMIT	设定机器人运动的下限。
TIMER	设定计时。
UTIMER	设定用户计时的值。
ON	开启系统开关(变为 ON)。
OFF	关闭系统开关(变为 OFF)。
NCHON	开启陷波滤波器 (变成 ON)。(选项)
NCHOFF	关闭陷波滤波器 (变成 OFF)。(选项)
WEIGHT	设定负荷重量数据。
MC	从 PC 程序中执行监控指令。
TPLIGHT	开启示教器的背光灯。
CURLIM	更改外部轴的马达电流限制值。

---

## HERE 位姿变量

---

### 功能

以当前的位姿，定义一个位姿变量。位姿可以是变换值、关节位移值或复合变换值。

### 参数

#### 位姿变量

用变换值变量、关节位移值变量，或复合变换值变量来指定现在位置的位姿变量。

#### [ 注 意 ]

仅定义复合变换值变量中最右边的变量。(见下面的示例)。如果复合值的其他变量未定义过，本指令会导致出错。

参看 5.5，HERE 监控指令。

---

**POINT 位姿变量 1=位姿变量 2, 关节位移值变量**

---

**功能**

将“=”号右边的位姿变量 2 的值赋值给“=”左边的位姿变量 1。

**参数**

**位姿变量 1**

指定要定义的位姿变量(关节位移值变量、变换值变量或复合变换值变量)。如果是复合变换值变量的话, POINT 指定最右侧的变量值。

**位姿变量 2**

指定已知的位姿变量的值。如果不指定, 符号“=”也应省略。

**关节位移值变量**

位姿变量 1 的值是关节位移值, 并且位姿变量 2 的值仅在其是变换值时有效。(位姿变量 1 不是关节位移值变量时, 不能指定。)设定此参数时, 以此关节位移变量指定的形态由变换值来求关节位移值。如果不指定, 则以当前机器人的形态来求位姿变量。

**说明**

如果位姿变量 2 未被定义, 将出现错误。

如果位姿变量 1 是复合变换值变量, 则根据此命令仅定义其复合变换值变量的右端的变换值变量。另外, 此时右端以外的变换值变量中其他任一变量如果未定义的话, 将出现错误。

参看 5.5 POINT 监控指令。

---

POINT/X	变换值变量 1=变换值变量 2
POINT/Y	变换值变量 1=变换值变量 2
POINT/Z	变换值变量 1=变换值变量 2
POINT/OAT	变换值变量 1=变换值变量 2
POINT/O	变换值变量 1=变换值变量 2
POINT/A	变换值变量 1=变换值变量 2
POINT/T	变换值变量 1=变换值变量 2
POINT/7	变换值变量 1=变换值变量 2

---

### 功能

将变换值变量 2 的变换值分量赋值给与变换值变量 1 的变换值相应的分量。这些值将显示在终端上以供编辑。

### 参数

变换值变量 1

选择要定义的变换值变量（变换值变量或复合变换值变量）

变换值变量 2

如果不指定，则“=”也应省略。

### 说明

如果“=”号右边有任意未定义的位姿变量，将出错。

如果位姿变量 1 指定了复合变换值，此命令仅定义复合变换值最右端的变换值。然而，在该复合变换值中除了最右端之外的其他任一变换值如果未定义，则将出错。

### 示例

POINT/X temp=tempx

将变换值“tempx”中的 x 分量赋值给变换值“temp”中的 x 分量。



---

**DECOMPOSE 数组变量[元素编号]=位姿变量**

---

**功能**

将指定的位姿变量值的每个分量(变换值为 X, Y, Z, O, A, T; 关节位移值为 JT1, JT2, JT3, JT4, JT5, JT6), 存储到一个数组变量的各个元素中。

**参数**

数组变量

指定数组变量名称, 用于保存各分量值。

元素编号

指定用于保存分量的第一个元素。

位姿变量

指定从中提取分量的位姿变量名称(变换值、关节位移值)。

**说明**

此命令将指定的位姿信息的分量赋值到数组变量的元素中。

变换值时, 6 个元素分别被赋予 XYZOAT 值; 以关节位移值表示时, 元素被赋予机器人手臂上各关节的角度值。

**示例**

**DECOMPOSE X[0]=part**

将变换值“part”的分量赋值给数组变量“x”的前 6 个元素。

**DECOMPOSE angles[4]=#pick**

把关节位移值“#pick”的分量赋值给数组变量“angles”中的第 4 个元素开始的元素。

例如, 在以上命令中, 如果#pick 的值为(10, 20, 30, 40, 50, 60), 那么:

**angles[4]=10 angles[7]=40**

**angles[5]=20 angles[8]=50**

**angles[6]=30 angles[9]=60**

---

## BASE 变换值变量

---

### 功能

定义基础变换值。

### 参数

变换值变量

为定义新的基础坐标系的变换值或复合变换值用位姿变量来指定。在此的位姿变量描述了相对于空基础坐标系的基础坐标系的位姿，以空基础坐标系中的值表达。

### 说明

此命令执行时，CP 运动停止(BREAK)，基础变换值被改为指定的变换值。

参看 5.6，BASE 监控指令。

---

## TOOL 变换值变量, 工具形状编号

---

### 功能

定义工具变换值。

### 参数

变换值变量

为定义新的工具坐标系的变换值或复合变换值用位姿变量来指定。此位姿变量描述了相对于空工具坐标系的工具坐标系的位姿，以空工具坐标系中的值表达。如果（变换值或符合变换值）参数指定为“NULL”，工具坐标将被设定为空工具坐标。

工具形状编号

指定用于在示教和检查模式下速度控制的工具形状。其可以被省略。如果未指定，不对工具形状进行速度控制。

### 说明

命令执行时，连续路径(CP)运动停止(BREAK)，工具变换值被改为指定的变换值。

参看 5.6 TOOL 监控指令。

---

**SET\_TOOLSHAPE 工具形状编号=变换值变量 1, 变换值变量 2, ..., 变换值变量 8**

---

**功能**

注册用于在示教模式和检查模式下速度控制的工具形状。

**参数**

工具形状编号。

指定注册用的工具形状编号。设定范围：1 到 9。

**变换值变量 1-8**

为工具形状指定变换值变量数。其最大位姿变量数可以指定为 8 个。从法兰盘中心看到的位置用变换值来指定。但只有 X, Y, Z 值用于工具形状注册。

**说明**

进行在示教模式或检查模式下速度控制时，把可参照的工具形状最多用 8 个变换值变量来设定。

参看 5.6 SET\_TOOLSHAPE 指令。

---

### **ENA\_TOOLSHAPE 工具形状编号=TRUE/ FALSE**

---

#### **功能**

对指定工具形状编号的工具端点设定在示教模式和检查模式下速度控制的有效/无效。

#### **参数**

工具形状编号

指定转换有效/无效的工具形状编号 1 到 9。

TRUE/FALSE

根据工具形状使速度控制有效时指定 TRUE，使其无效时指定 FALSE。

#### **说明**

选择在指定工具形状编号的工具端点是否进行在示教和检查模式下的速度控制。所有工具形状的默认值都被设定为 FALSE（无效）。如果工具端点位置一个也没设定，而工具形状设定为 TRUE（有效）时，机器人在示教・检查模式下运转时，会产生错误 E1356 工具形状未设定，用 SET\_TOOLSHAPE 指令必须指定一个以上的工具端点位置或用本指令把设定由 TRUE（有效）变更为 FALSE（无效）后，用 TOOL 指令或 TOOLSHAPE 指令来设定此工具形状编号。（一旦，设定为 TRUE，如果不实行 TOOL/TOOLSHAPE 指令的话，此设定不会由 TRUE 变更为 FALSE。）

---

## TOOLSHAPE 工具形状编号

---

### 功能

选择用于在示教模式和检查模式下控制速度的工具形状。

### 参数

工具形状编号

给控制速度的工具形状指定编号。设定范围：1 到 9。

### 说明

根据工具形状进行在示教模式和检查模式下速度控制时，必须要执行 ENA\_TOOLSHAPE 命令/指令 (ENA\_TOOLSHAPE n =TRUE)。如果工具端点位置一个也没设定（所有点都设定为 0）的话，会产生错误 E1356 工具形状未设定。

参看 5.6 TOOLSHAPE 指令。

---

## SETHOME 精度, 关节位移值变量 SET2HOME 精度, 关节位移值变量

---

### 功能

设定 HOME(原点)位姿。

### 参数

精度

以毫米为单位，设定 HOME 位姿的精度范围。

关节位移值变量

为设定 HOME 位姿，指定以关节位移值定义的位姿变量。

### 说明

通过指定各关节的位移值(以角度为单位)，设定机器人的 HOME 位姿。

参看 5.6, SETHOME/ SET2HOME 指令。

### 示例

SETHOME 10, #place            把#place 位姿的关节位移值记录为 HOME 位姿。HOME 的精度范围为不大于 10mm。

---

<b>ULIMIT</b>	<b>关节位移值变量</b>
<b>LLIMIT</b>	<b>关节位移值变量</b>

---

### 功能

设定和显示机器人动作范围的上/下限。

### 参数

关节位移值变量

以关节位移值变量设定软件限位值(上限或下限)。

### 说明

用关节位移值(单位为度)指定机器人运动范围的上(下)限。

参看 5.6 LIMIT/LLIMIT 监控指令。

---

<b>TIMER</b>	<b>计时编号=时间</b>
--------------	----------------

---

### 功能

设定指定计时的时间。

### 参数

计时编号

指定计时编号以指定时间，允许的编号为 1 到 10。

时间

以秒为单位指定计时的定时时间。

### 说明

此命令执行时，计时立即被置为指定的时间，可用 TIMER 函数检查计时的值。

### 示例

下例中，用 TIMER 命令和 TIMER 函数使程序执行暂停指定的时间。

<b>TIMER 1=0</b>	设置计时 1 时间为 0 (秒)
<b>WAIT TIMER(1)&gt;delay</b>	一直等待，直到计时 1 的值大于 delay 。

---

## UTIMER @计时变量=计时值

---

### 功能

为用户计时设置默认值。用户计时可以用计时变量自由命名称。多个用户计时可以同时使用。

### 参数

@计时变量

指定用作为计时名称使用的变量名称或数组变量名称。在整数变量前需输入@。

计时值

为计时指定默认值。允许的范围是 0 到 2147483647 (秒)。

---

开关名称,	.....ON
开关名称,	.....OFF

---

### 功能

开 (ON) / 关 (OFF) 指定的系统开关。

### 参数

开关名称

将这里指定的开关变成 ON (OFF)。可以输入多个开关，中间用逗号隔开。

可用 SWITCH 命令检查系统开关的当前设置状态。

参看 5.6 ON/OFF 监控指令。

NCHON	
NCHOFF	选项

**功能**  
在指定步骤的后续运动步骤中，开启 (或关闭) 陷波滤波器 。用 NCHON 时开启陷波滤波器 ，用 NCHOFF 时关闭。

**示例**

10	SPEED	100	ALWAYS	
20	HOME			
30	DRIVE5,	90		
40	NCHOFF			} 以陷波滤波器 的 OFF 状态进行直线移动。
50	SPEED	10		
60	DRAW	1000		
70	NCHON			
80	DRAW,	-500		

[ 注 意 ]

通常用 NCHON。

当低速操作中可以见到微小振动时；或有外力作用于机器人时，可用 NCHOFF。

EXECUTE 命令的执行将使滤波器复位。PRIME，STEP，MSTEP 等命令首次执行时滤波器也将被复位。滤波器缺省的设置为 ON(陷波滤波器 有效)。



---

<b>WEIGHT</b>	<b>负荷质量, 重心位置 X, 重心位置 Y, 重心位置 Z, 绕 X 轴的惯性力矩, 绕 Y 轴的惯性力矩, 绕 Z 轴的惯性力矩</b>
---------------	---

---

### 功能

设定负荷的质量数据(工具和工件的重量)。该数据用来确定机器人手臂最佳的加速度/减速度。

### 参数

#### 负荷质量

工具和工件的质量(单位: 千克)。范围: 0.0 到最大负载能力(kg)。

#### 重心位置(单位: mm)

X: 工具坐标系中重心的 x 值。

Y: 工具坐标系中重心的 y 值。

Z: 工具坐标系中重心的 z 值。

绕 X 轴的惯性力矩, 绕 Y 轴的惯性力矩, 绕 Z 轴的惯性力矩(选项)。

设定绕各个轴的惯性力矩, 单位为  $\text{kg} \cdot \text{m}^2$ 。绕各轴的惯性力矩定义为: 以工具重心为旋转中心、绕平行于空工具坐标的各坐标轴、所得到的力矩。

### 说明

在程序中使用 WEIGHT 命令时, 若未指定有关参数, 将以该型号机器人的最大负载能力作为默认值。



### 危 险

必须设定正确的负载质量和重心位置。不正确的数据可能会削弱或缩短零部件的寿命, 或引起过载/偏差错误。

---

## MC 监控指令

---

### 功能

从 PC 程序中启用执行监控指令。此命令可使用的监控指令有 ABORT、CONTINUE、ERESET、EXECUTE、HOLD、与 SPEED。

### 说明

当机器人程序从 AUTOSTART.PC 程序中执行(用 EXECUTE 命令)时，可使用此命令。MC 命令不能用在机器人程序中。

### 示例

用此命令可以从 AUTOSTART.PC 中执行机器人程序。当然，执行机器人程序前，必须先开启马达电源，因此应用 MC 命令时，要增加如下命令，以检查电源是否为 ON。

```
autostart.pc()  
1 10 IF SWITCH(POWER)==FALSE GOTO 10  
2 MC EXECUTE pg1  
. END
```

---

## TPLIGHT

---

### 功能

打开示教器的背光灯。

### 说明

如果示教器屏幕的背光灯为 OFF，那么此命令将灯打开。如果执行本命令时背景灯已开，灯光将再亮 600 秒。

---

**CURLIM 轴编号, 正电流限制值, 负电流限制值**

---

**功能**

更改外部轴的马达的电流限制值。

**参数****轴编号**

指定外部轴的编号。容许范围：7-9。

**正电流限制值**

指定马达的正电流的限制值。该限制值设置为外部轴伺服参数的电流限制值的百分比。单位：%。容许范围：大于 0，小于等于 100。

**负电流限制值**

指定马达的负电流的限制值。该限制值设置为外部轴伺服参数的电流限制值的百分比。单位：%。容许范围：大于 0，小于等于 100。

**说明**

此命令仅在带有 KHI 放大器的外部轴有效。

更改的电流限制值在以下的情况下，复位到外部轴伺服参数的电流限制值。

- 用 EXECUTE 指令执行程序时
- 重新选择并由循环开始执行一个新程序时
- 复位程序执行时

## 6.10 程序和数据控制命令

DELETE	删除机器人存储器中的程序和变量。
DELETE/P	删除机器人存储器中的程序。
DELETE/L	删除机器人存储器中的位姿变量。
DELETE/R	删除机器人存储器中的实型变量。
DELETE/S	删除机器人存储器中的字符串变量。
TRACE	开启 (ON) 或关闭 (OFF) TRACE 功能 。
NLOAD	将指定文件读入到机器人存储器中。
SLOAD	将指定文件读入到机器人存储器中，文件以字符串的形式之指定。

---

<b>DELETE</b>	<b>程序名, .....</b>
<b>DELETE/P</b>	<b>程序名, .....</b>
<b>DELETE/L</b>	<b>位姿变量, .....</b>
<b>DELETE/R</b>	<b>实型变量 [数组元素] , .....</b>
<b>DELETE/S</b>	<b>字符串变量 [数组元素] , .....</b>

---

### 功能

从存储器中删除指定的数据。

### 参数

程序名 (/P)，位姿变量 (/L)，实型变量 (/R)，字符串变量 (/S)；  
指定要删除的数据类型和名称。

参看 5.2 DELETE 监控指令。

---

**TRACE** **stepper 编号: ON/OFF**

---

### 功能

开始 (ON) 或结束 (OFF) 跟踪已记录的机器人程序或个人计算机程序。

### 参数

stepper 编号

用以下整数值指定程序类型:

1: 机器人程序

1001: PC 程序 1

1004: PC 程序 4

1002: PC 程序 2

1005: PC 程序 5

1003: PC 程序 3

若类型未指定, 则记录所有的程序。

ON/OFF 开/关

开始/结束跟踪记录。

### 说明

如果在 TRACE ON 之前未使用 SETTRACE 命令保存必要的存储器(内存), 将产生错误信息 (P2034) “未定义的内存”。

参看 5.2 TRACE/SETTRACE 监控指令。

---

**NLOAD/IF/ARC 设备编号=文件名+ 文件名+ ..., 状态变量**

---

### 功能

从外部存储器设备读取指定的文件，并装载到机器人存储器中。

### 参数

设备编号

指定哪里显示当前处理情况，设定值为 1 或 2：

- 1: 标准终端 (PC)
- 2: 示教器

省略时，显示在执行程序的设备上。

文件名

指定要装载到机器人存储器中的文件，若要指定多个文件时，用“+”隔开文件名。

状态变量

设定为 0 时，即使发生错误也允许程序继续执行，故障代码被保存；省略时，程序在错误发生时停止执行。

### 说明

将指定文件的内容(程序和变量)装载到机器人存储器中，若指定的程序与机器人存储器中已存在的程序同名，将导致出错，且不读入新的程序。

- 1. 设定成 NLOAD/IF 时，装载接口面板数据。
- 2. 设定成 NLOAD/ARC 时，装载弧焊焊接数据。

当正在装载的程序中有某步骤不能读取时，显示出错信息“步骤格式出错。(0：改变为注释后继续，1：删除程序 退出)”。当选择“0”继续时，在加载完成后，请用编辑器对程序进行修改。

#### — [ 注 意 ] —

姿态变量、实数变量、或字符串变量与读入中的变量同名时，将被新数据覆盖(重写)，而不警告。

### 示例

NLOAD Pallet	将文件 pallet.as 的内容读入机器人存储器中。
NLOAD f3.pg	将文件 f3.pg 的所有程序读入机器人存储器中。

---

**SLOAD/IF/ARC 设备编号=文件名, 状态变量**

---

### 功能

从外部存储器读取由字符串指定的文件，并加载到机器人存储器中。

### 参数

设备编号

指定在哪里显示当前的处理情况，设定值为 1 或 2：

1：标准终端（PC）

2：示教器

省略时，显示在执行程序的设备上。

字符串

指定文件，加载到机器人存储器中。只能指定一个文件。

状态变量

设定为 0 时，即使发生错误也允许程序继续执行，故障代码被保存；省略时，程序在错误发生时停止执行。

### 说明

将指定文件的内容(程序和变量)装载到机器人存储器中，若指定的程序与机器人存储器中已存在的程序同名，将导致出错，且不读入新的程序。

1. 设定成 SLOAD/IF 时，装载接口面板数据。

2. 设定成 SLOAD/ARC 时，装载弧焊焊接数据。

当正在装载的程序中有某步骤不能读取时，会显示出错信息“步骤格式出错。(0：改变为注释后继续，1：删除程序 退出)”。当选择“0”继续时，在加载完成后，请用编辑器对程序进行修改。

#### [ 注 意 ]

姿态变量，实数变量，或字符串变量与读取中的变量同名时，将被新数据覆盖(重写)，而不警告。

### 示例

SLOAD \$str1      当\$str1=“ pallet.as” 时，加载文件 pallet.as 的内容。

SLOAD \$str2      当\$str2=“ f3.pg” 时，加载文件 f3.pg 的内容。





## 7.0 AS 系统开关

本章描述各系统开关的功能。关于设置 ON/OFF 或检查开关的状态，请参阅 SWITCH、ON 和 OFF 监控指令/程序命令。

CP	启用或关闭连续路径 (CP) 功能。
CHECK. HOLD	当 HOLD/RUN (保持/运转) 在 HOLD 状态时, 允许或禁止从键盘输入指令。
CYCLE. STOP	用外部 HOLD 停止循环。
MESSAGES	允许或禁止终端输出。
OX. PREOUT	设定外部信号产生的时序。
PREFETCH. SIGINS	允许或禁止提前处理 AS 程序中的 I/O 信号。
QTOOL	允许或禁止在一体化示教过程中进行工件转换。
RPS	允许或禁止程序的随机选择。
SCREEN	控制终端显示。
REP_ONCE	设定再现循环一次还是连续。
STP_ONCE	设定执行一次一步步 (单步) 还是连续执行。
AUTOSTART. PC	启用控制电源开启时的 PC 程序自动启动。
ERRSTART. PC	决定出错时是否执行选定的 PC 程序。
TRIGGER	显示 <span style="border: 1px solid black;">握杆触发</span> 开关的 ON/OFF 状态。
CS	显示循环启动的 ON/OFF 状态。
POWER	显示 <span style="border: 1px solid black;">马达电源</span> 的 ON/OFF 状态。
RGSO	显示伺服的 ON/OFF 状态。
TEACH_LOCK	显示 <span style="border: 1px solid black;">示教锁定</span> 的 ON/OFF 状态
ERROR	显示程序是否在错误状态。
REPEAT	显示 <span style="border: 1px solid black;">TEACH/REPEAT</span> ( <span style="border: 1px solid black;">示教/再现</span> ) 开关的状态。
RUN	显示当前的保持/运转的状态。
PNL_CYCST	显示循环启动的 ON/ OFF 状态。
PNL_MPOWER	显示 <span style="border: 1px solid black;"> </span> 马达电源的 ON/ OFF 状态。

PNL_ERESET	显示操作面板上 <b>ERRORRESET</b> ( <b>错误复位</b> ) 的 ON/ OFF 状态。
TPKEY_A	显示示教器上 <b>A</b> 键的 ON/ OFF 状态。
DISPIO_01	改变 I0 指令的显示模式。
HOLD. STEP	允许程序暂停时的执行中步骤的显示。(选项)
WS_COMPOFF	改变 WS 信号的输出时序。(选项)
FLOWRATE	从流量控制模式切换到速度输出模式(反之亦然)。(选项)
WS. ZERO	改变 WS=0 时执行的焊接处理。(选项)
ABS. SPEED	允许绝对速度的使用。
SLOW_START	允许或禁止低速启动功能。
AFTER. WAIT. TMR	设定点焊控制器(timer)在一体化命令程序中如何开始。
STAT_ON_KYBD	显示在键盘屏幕上的身份数据。
WAITREL_AUTO	显示等待解除弹出式窗口。
REP_ONCE. RPS_LAST	选择反复一次操作时的结束步骤。
DEST_CIRINT	决定 DEST/#DEST 函数返回的位姿。
PROGDATE	决定程序的更改日期是否加入到程序信息中。

---

## CP

---

### 功能

启用或关闭连续路径 (CP, continuous path) 功能。

### 说明

此开关用来开启 (ON) (启用) 或关断 (OFF) (关闭) CP 功能。如果此开关在程序中改变, 将从下一个运动开始启用/关闭 CP 功能。此开关的默认设置是 ON。

### 示例

CP OFF      关闭 CP 功能

---

## CHECK. HOLD

---

### 功能

当 HOLD/RUN(保持/运转) 处于 HOLD(保持) 状态时, 允许或禁止使用键盘来输入 EXECUTE、DO、STEP、MSTEP 和 CONTINUE 指令。

### 说明

如果此开关为 ON, 仅当 HOLD/RUN(保持/运转) 处于 HOLD(保持) 状态时, 才接受从键盘输入的下列指令。(接受这些指令, 但是并不开始运动, 除非 HOLD/RUN(保持/运转) 转换到 RUN 状态。)

EXECUTE, DO, STEP, MSTEP, CONTINUE

如果此开关为 OFF, 无论 HOLD/RUN(保持/运转) 处于什么状态, 均接受上述指令。不通过键盘的操作不受这些指令的影响。(如果 HOLD/RUN(保持/运转) 处于 RUN 状态, 一旦输入指令机器人就立即开始运动)。

默认设置是 OFF。

---

## CYCLE STOP

---

### 功能

确定在外部保持后是否继续再现执行循环。

### 说明

如果此开关为 OFF，当外部保持(External HOLD)信号输入时，机器人停止，但 CYCLE START (循环启动) 仍保持为 ON。但当外部保持信号变为 OFF 时，机器人继续程序执行。

如果此开关为 ON，当外部保持信号输入时机器人停止，并且 CYCLE START 变为 OFF。因此，即使外部保持信号解除，CYCLE START 保持为 OFF，机器人也不能继续执行程序。要继续执行程序，需遵循正规的执行顺序(也就是说在示教器上按 **A** + **CYCLE START** 钮)。

此开关仅对外部停止有效。当 **HOLD/RUN** (保持/运转) 状态为 HOLD (保持) 状态时，不论此系统开关的状态如何，在 **CYCLE START** 保持为 ON，机器人停止。当 **HOLD/RUN** 被转换到 RUN 状态时，机器人继续执行程序。

默认设置是 OFF。

---

## MESSAGES

---

### 功能

允许或禁止在终端上输出信息。

### 说明

如果此开关为 ON，使用 PRINT 或 TYPE 指令时，信息将在终端上显示。如果此开关为 OFF，信息不在终端上显示。

默认设置为 ON。

---

## **OX. PREOUT**

---

### **功能**

确定在一体化命令中开启 (ON)/关断 (OFF) OX 信号的时序。

### **说明**

当运行中的程序是由一体化命令示教的程序、并且此开关为 ON 时，示教给给定位姿的 OX 信号，将在机器人开始向此位姿移动时，立即变为 ON。

如果此开关为 OFF，信号在该步骤的位姿轴一致后才变为 ON。

默认设置为 ON。

---

## **PREFETCH. SIGINS**

---

### **功能**

允许或禁止 AS 程序中信号输入和输出指令的提前处理。

### **说明**

如果此开关为 OFF，下列信号输入/输出和同步指令，直到当前运动命令的位姿轴一致后，才会被执行。

SWAIT, SIGNAL, TWAIT, PULSE, DLYSIG, RUNMASK, RESET, BITS

如果此开关为 ON，一旦开始向示教点移动，所有指令包括上述指令被立即执行，并且程序被一直执行到下一运动命令的前一步骤。

默认设置为 OFF。

## QTOOL

### 功能

允许或禁止在一体化示教过程中进行工具转换。

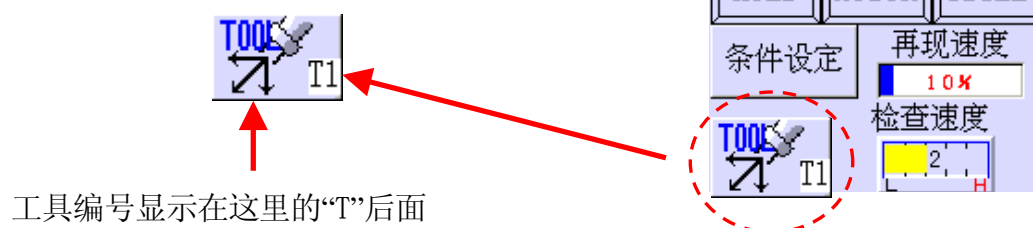
### 说明

如果此开关为 ON，下面的功能将影响正在示教中的工具坐标系：

#### 1. 工具坐标系的自动选择

- (1) 如果当前步骤是由一体化命令进行示教的，机器人将按照已注册的当前选定的工具（1—9）的工具数据进行动作。
- (2) 如果示教了一条 AS 语言命令在当前步骤中，工具坐标系保持不变。
- (3) 如果当前步骤中还没有示教内容，并且示教器正在显示辅助数据画面，机器人将按照已注册的当前画面显示的工具数据中的工具（1—9）的工具坐标系进行动作。
- (4) 如果当前步骤中还没有示教内容，示教器也不显示辅助数据，那么工具坐标系将保持不变。

当前有效的工具编号显示在示教器右上角的状态区中。



工具编号显示在这里的“T”后面

当使用 AS 语言中指定的工具时，将此开关关断 OFF，工具编号则显示为“T 0”。如果示教工作一直用 AS 语言进行，请保持此开关为 OFF。

如果此开关为 OFF，当执行了 TOOL 指令时，或当执行了程序中的 TOOL 命令或一体化命令时，工具坐标系改变。

默认设置为 ON。

---

## RPS

---

### 功能

允许或禁止程序（JMP，RPS）的随机选择。

### 说明

如果此开关为 OFF，将忽略 EXTCALL 命令和辅助数据的 JUMP/ END。

默认设置为 OFF。

---

## SCREEN

---

### 功能

允许或禁止屏幕画面滚动。

### 说明

如果此开关为 ON，当显示已经充满屏幕时，屏幕停止滚动。按 空格 键显示下一页。

默认设置为 ON。



---

### REP\_ONCE

---

#### 功能

确定程序是运行一次还是连续运行。

#### 说明

如果此开关为 ON，程序运行一次。

如果此开关为 OFF，程序连续运行。

默认设置为 OFF。

---

### STP\_ONCE

---

#### 功能

确定程序步骤是一次运行一步还是连续运行。

#### 说明

如果此开关为 ON，程序一次运行一个步骤。

如果此开关为 OFF，程序连续运行所有步骤。

默认设置为 OFF。

---

**AUTOSTART. PC**  
**AUTOSTART2. PC**  
**AUTOSTART3. PC**  
**AUTOSTART4. PC**  
**AUTOSTART5. PC**

---

### 功能

确定当控制电源开启时选定的 PC 程序是否自动启动运行。

### 说明

如果此开关为 ON，当控制电源开启时，名为 AUTOSTART. PC 的 PC 程序将自动启动。但，预先需创建程序 AUTOSTART. PC。可以选择五个不同的 PC 程序自动启动，它们的名称分别为：AUTOSTART. PC 和 AUTOSTART2. PC 到 AUTOSTART5. PC。分别创建程序名，并将各自对应的开关变为 ON，来启动需要的程序。

默认设置为 OFF。

### [ 注 意 ]

如果此开关为 ON，但是对应的程序不存在，则出错并显示信息“程序不存在”。

---

## ERRSTART. PC

---

选项

### 功能

确定在错误发生时是否执行选定 PC 程序。

### 说明

如果此开关为ON，当错误发生时自动执行命名为ERRSTART. PC的PC程序。预先创建名为ERRSTART. PC的PC程序。

一旦执行了ERRSTART. PC程序，此开关被自动关断为OFF。注意该ERRSTART. PC程序是作为第5个PC程序执行的，因此如果已经有5个PC程序处于运行中，就不能执行ERRSTART. PC。

要重新设定ERRSTART. PC程序自动执行，必须再次设定此开关到ON。

默认设置为OFF。

### [ 注 意 ]

如果此开关为 ON，但对应的 ERRSTART. PC 程序不存在，则出错并显示信息“程序不存在”。

---

## SWITCH (TRIGGER)

---

### 功能

显示示教器上的握杆触发开关是 ON 还是 OFF。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果握杆触发开关为 ON，返回-1；如果为 OFF，则返回 0。

---

### SWITCH (CS)

---

#### 功能

显示 CYCLE START (循环启动) 按钮是 ON 还是 OFF。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果 CYCLE START 为 ON，返回-1；如果为 OFF，则返回 0。

---

### SWITCH (POWER)

---

#### 功能

显示马达电源的开关是 ON 还是 OFF。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果马达电源的开关是 ON，返回-1；如果是 OFF，则返回 0。

---

### SWITCH (RGS0)

---

#### 功能

显示伺服马达电源是 ON 还是 OFF。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果伺服马达电源为 ON，返回-1；如果为 OFF，则返回 0。

---

### SWITCH (TEACH\_LOCK)

---

#### 功能

显示 示教锁定 开关是 ON 还是 OFF。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果此开关为 ON，返回-1；如果为 OFF，则返回 0。

---

### SWITCH (ERROR)

---

#### 功能

显示当前是否出错。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果有出错返回-1，如果无出错则返回 0。

---

### SWITCH (REPEAT)

---

#### 功能

显示在控制面板上的 TEACH/REPEAT (示教/再现) 开关是处于 TEACH 位置还是处于 REPEAT 位置。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果开关处于 REPEAT 位置返回-1，如果开关处于 TEACH 位置则返回 0。

---

### **SWITCH (RUN)**

---

#### **功能**

显示 RUN/HOLD（运转/保持）是处于 RUN 状态还是处于 HOLD 状态。

此功能不会开启/关断该开关。

在和 SWITCH 函数一起使用时，如果开关处于 RUN 状态返回-1，如果其处于 HOLD 状态则返回 0。

---

### **SWITCH (PNL\_CYCST)**

---

#### **功能**

显示控制面板上的 CYCLE START（循环启动）开关是被按下了 (ON) 还是没有 (OFF)。

开关不能用此命令开启 (ON) 或关断 (OFF)。

在和 SWITCH 函数一起使用时，当此开关为 ON 时返回-1，当为 OFF 时则返回 0。

---

### SWITCH (PNL\_MPOWER)

---

#### 功能

显示马达电源是被按下了 (ON) 还是没有 (OFF)。

开关不能用此命令开启 (ON) 或关断 (OFF)。

在和 SWITCH 函数一起使用时，当此开关为 ON 时返回-1，当为 OFF 时则返回 0。

---

### SWITCH (PNL\_ERESET)

---

#### 功能

显示控制面板上的 **ERROR RESET** (**错误复位**) 开关是被按下了 (ON) 还是没有 (OFF)。

开关不能用此命令开启 (ON) 或关断 (OFF)。

在和 SWITCH 函数一起使用时，当此开关为 ON 时返回-1，当为 OFF 时则返回 0。




---

## SWITCH (TPKEY\_A)

---

### 功能

显示示教器上的  开关是被按下了 (ON) 还是没有 (OFF)。

开关不能用此命令开启 (ON) 或关断 (OFF)。

在和 SWITCH 函数一起使用时，当此开关为 ON 时返回-1，当为 OFF 时则返回 0。

DISPIO\_01

功能

改变 I/O 指令执行时、信号 (外部 I/O 信号和内部信号) 的显示方式。

说明

如果系统开关 DISPIO\_01 为 OFF，用“o”显示 ON 的信号。用“x”显示 OFF 的信号。专用信号用大写字母 (“O” 和 “X”) 显示。

如果系统开关 DISPIO\_01 为 ON，用“1”显示 ON 的信号，用“0”显示 OFF 的信号。未安装的外部 I/O 信号用“-”显示。


默认设置为 OFF。（请参见 5.6 I/O 监控指令）

示例

当 DISPIO\_01 为 OFF 时

```
>IO 
32—      1      xxxx   xxxx   xxxx   xxXX   xxxx   XXXX   XXXO   XXXO
1032—1001      xxxx   xxxx   xxxx   xxXX   xxxx   XXXX   XXXX   OXXX
2032—2001      xxxx   xxxx   xxxx   xxxx   xxxx   xxxx   xxxx   xxxx
>
```

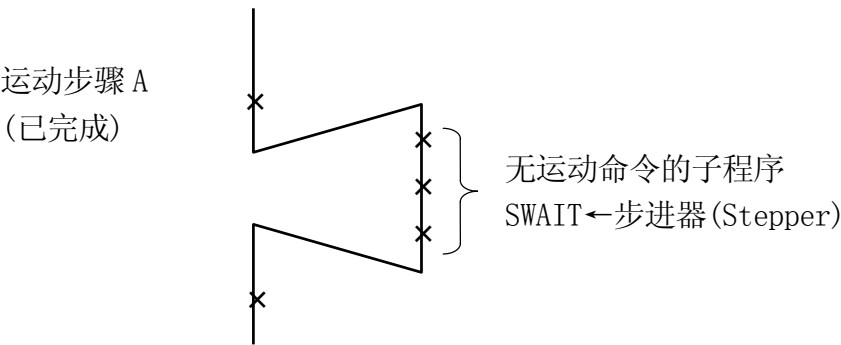
当 DISPIO\_01 为 ON 时

```
>IO 
32—      0      0000   0000   0000   0000   0000   0000   0001   0001
1032—1001      0000   0000   0000   0000   0000   0000   0000   1000
2032—2001      0000   0000   0000   0000   0000   0000   0000   0000
>
```

**HOLD. STEP**

**功能**  
选择当程序执行被暂停时要显示的步骤。

**说明**  
如果此开关为 ON，并且在非运动步骤执行过程中暂停了程序，将显示当前执行的步进器 (Stepper) 中的步骤，而不显示刚完成的运动命令。  
例如在下述情况中，如果此开关为 ON，显示步进器中的 SWAIT 步骤。(SWAIT 可以跳过)。  
如果此开关为 OFF，则显示运动步骤 A。



默认设置为 OFF。

**WS\_COMPOFF**

选项

**功能**  
改变焊接信号 (WS) 的输出条件。

**说明**  
如果此开关为 ON, WS 信号输出的时序将从存储器发生改变的时刻改为收到焊接完成信号输入的时刻。  
  
如果此开关为 OFF, WS 信号在每次存储器改变时输出。  
  
默认设置为 OFF。

---

## FLOWRATE

---

选项

### 功能

在流量控制模式与速度输出模式之间切换。

### 说明

如果此开关为 ON，选择流量控制模式。

如果此开关为 OFF，选择速度输出模式。

默认设置为 OFF。

---

## WS. ZERO

---

选项

### 功能

改变在 WS 为 0 时执行的操作。

### 说明

如果此开关为 ON，当 WS=0 执行焊接，就像 WS≠0 时一样。(加压和焊接)

如果此开关为 OFF，当 WS=0 不执行焊接。(仅加压)

默认设置为 OFF。

---

## ABS. SPEED

---

### 功能

允许或禁止绝对速度的使用。此功能允许以低的、预先定义的速度，执行运动步骤。它优先于设定的监控速度。

### 说明

如果此开关为 ON，当下述条件为真时，机器人以为程序指定的绝对速度运动。

$$\text{最大速度} \times \text{监控速度} > \text{程序速度}$$

### 示例

如果最大速度为 2400 mm/s，监控速度为 10 %，程序速度为 100 mm/s，那么

$$2400 \times 0.1 > 100$$

机器人将以 100 mm/s 的程序速度运动。

如果最大速度为 100 %，监控速度为 10 %，程序速度为 5 %，那么

$$100 \times 0.1 > 5$$

机器人将以 5 %的程序速度运动

如果最大速度为 2400 mm/s，监控速度为 2 %，程序速度为 100 mm/s，那么

$$2400 \times 0.02 < 100$$

机器人以 48 mm/s 的监控速度运动。

---

### **SLOW\_START**

---

#### **功能**

允许 (ON) 或禁止 (OFF) 低速启动功能。如果允许低速启动功能，第一条运动命令将以低速再现速度执行。

#### **说明**

如果此开关为 ON，低速启动功能生效。

如果程序被停止，然后重新启动，那么第一条运动步骤将以低速再现速度开始。

---

### **AFTER\_WAIT\_TIMR**

---

#### **功能**

为一体化命令中的计时，设定开始计时的时序。

#### **说明**

如果此开关为 OFF，当轴一致时，计时开始计时。如果它为 ON，当轴一致并且所有设定的条件 (例如 WX、WAIT、RPS ON) 都满足时，计时才开始计时。

默认设置为 OFF。

---

## **STAT\_ON\_KYBD**

---

### **功能**

设定状态信息是否显示在键盘屏幕上。

### **说明**

如果此开关为 OFF，不显示状态信息，并且键盘屏幕在整个屏幕上显示。如果此开关为 ON，当在示教屏幕下，状态信息显示在键盘屏幕的最上方。键盘屏幕由于只有 4 行，因此小于整个屏幕的显示。

默认设置为 ON。

---

## **WAITREL\_AUTO**

---

### **功能**

在示教或检查模式下，当机器人等待时，设定等待解除的弹出式窗口是否显示。

### **说明**

如果此开关为 ON，当机器人在等待状态下，出现弹出式窗口。如果此开关为 OFF，不显示弹出式窗口。

默认设置为 ON。

---

### **REP\_ONCE. RPS\_LAST**

---

#### **功能**

当程序用 RPS 功能重复一次时，设定哪一步用来结束执行程序。

#### **说明**

如果此开关为 ON，REP\_ONCE 开关为 ON（重复一次）时，重复一次程序后，结束步骤已示教，在还没有移动到下一程序时，结束执行程序。

如果此开关为 OFF，在执行结束步骤后，程序移动到下一程序并停止在其程序的第一步。

默认设置为 OFF。



## DEST\_CIRINT

### 功能

决定 DEST/#DEST 函数在运动类型 2 中的精度圆内的圆弧轨迹上返回的位姿。

### 说明

机器人在运动类型 2 中沿着 A→B→C 做直线插补运动时，其运动路径将显示如下图。该图例子中的点 B 的精度设置为 1 mm, 100 mm, 200 mm。机器人一进入设置了精度的圆内，就开始走捷径到下一段路径。机器人将沿精度圆内的圆弧轨迹运动。请参阅“4.5.4 CP 开关和 ACCURACY、ACCEL、DECEL 命令之间的关系”。

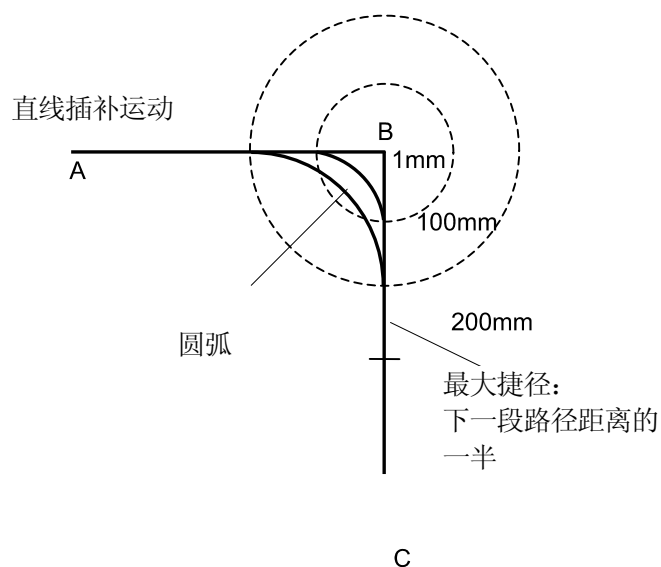
在以下两种情况下，使用 DEST/#DEST 函数时，通过设置此开关 ON 或 OFF 来更改返回位姿。

1. 当机器人在点 A 和开始圆弧运动的点之间时
2. 当机器人走圆弧轨迹时（在精度圆内）

如果此开关为 ON，精度设置为 200 mm，则在第一种情况下，返回点 A'，在第二种情况下，返回 B'。

如果此开关为 OFF，精度设置为 200 mm，则在第一种情况下，返回点 B，在第二种情况下，返回 C'。

默认设置为 ON。



---

**PROG. DATE**

---

**功能**

决定程序的更改日期是否加入到程序信息中。

**参数**

如果此开关为 ON，在程序表画面中将显示项目选择按钮。按此按钮，将显示执行次数的总数和程序更改日期。

如果此开关为 OFF，在程序表画面中将不显示项目选择按钮。代替程序表，将显示程序中的步骤数和其评论。

默认设置为 OFF。



## 8.0 运算符

本章描述 AS 语言中运算符的功能。这些运算符可与监控指令和程序命令联合使用。

### 8.1 算术运算符

算术运算符用来执行普通的数学计算。

运算符	功能	示例
+	加	$i=i+1$
-	减	$j=i-1$
*	乘	$i=i*3$
/	除	$i=i/2$
MOD	求余数	$i=i\text{MOD}2$
^	乘方	$i=i^3$

#### 示例

$i=i+1$       把  $i$  的值加上 1 赋值给  $i$ ；例如，当  $i$  是 5 时，6 将作为表达式  $i+1$  的结果被赋值给  $i$ ，

$i=i \text{ MOD } 2$       当  $i$  是 5 时，运算符计算  $5 \div 2$ ，并将余数 1 赋值给  $i$ 。

$i=i^3$       将  $i^3$  赋值给  $i$ 。当  $i$  是 2 时，将 8 赋值给命令左侧的  $i$ 。

早除法 (/) 和求余数 (MOD) 计算中，如果使用 0 作为命令最右边的值 (除数) 将导致错误。

#### 示例

$i=i/0$

$i=i \text{ MOD } 0$

## 8.2 关系运算符

关系运算符和 IF 和 WAIT 等命令一起使用，用于检验一个条件是否被满足。

运算符	功能	示例
<	当左边的值小于右边时为真(-1)	$i < j$
>	当左边的值大于右边时为真(-1)	$i > j$
<=	当左边的值小于或等于右边时为真(-1)	$i \leq j$
=<	与上述相同	$i = < j$
>=	当左边的值大于或等于右边时为真(-1)	$i \geq j$
=>	与上述相同	$i = > j$
==	当两边相等时为真(-1)	$i == j$
<>	当两边不等时为真(-1)	$i < > j$

### 示例

**IF  $i < j$  GOTO 10**      当  $j$  大于  $i$  时(也就是说命令  $i < j$  为真)，程序跳到带标签 10 的步骤处。如果不是，程序则继续执行下一步骤。

**WAIT  $t == 5$**       当  $t$  是 5 时(也就是说  $t == 5$  为真)，程序执行下一步骤。如果不是，程序执行被暂停直到条件被满足。

**IF  $i + j > 100$  GOTO 20**      当  $i + j$  大于 100 时(也就是说表达式  $i + j > 100$  为真)，程序跳到带标签 20 的步骤处。如果不是，程序则继续执行下一步骤。

**IF  $\$a == "abc"$  GOTO 20**      当  $\$a$  是 “abc” (也就是说  $\$a == "abc"$  为真)，程序跳到带标签 20 的步骤处。如果不是，程序则继续执行下一步骤。

### 8.3 逻辑运算符

逻辑运算符用在布尔运算中, 比如  $0+1=1$ ,  $1+1=1$ ,  $0+0=0$  (逻辑或 (OR)), 或  $0 \times 1=0$ ,  $1 \times 1=1$ ,  $0 \times 0=0$  (逻辑与 (AND))。AS 系统中有两种逻辑运算符, 逻辑运算符和二进制运算符。

逻辑运算符不用来计算数值, 而是用来确定一个值或表达式是真还是假。如果数值是 0, 则被认为是假 (OFF)。所有非零值都被认为是真 (ON)。请注意: 用这个运算符计算时, 返回 -1 作为真。

运算符	功能	示例
AND	逻辑与 (Logical AND)	i AND j
OR	逻辑或 (Logical OR)	i OR j
XOR	逻辑异或 (Exclusive logical OR)	i XOR j
NOT	逻辑非 (Logical complement)	NOT i

#### 示例

**i AND j**                      计算 i 与 j 之间逻辑与的值。变量 i 和 j 通常是逻辑值, 但也可以是实数值。这种情况下, 所有非 0 的实数值都被认为是 ON(真)。

i	j	结果
0	0	0 (OFF 假)
0	非 0	0 (OFF 假)
非 0	0	0 (OFF 假)
非 0	非 0	-1 (ON 真)

仅当两个值都是 ON(真)时, 结果是 ON(真)。

**i OR j**                      计算 i 与 j 之间逻辑或的值。

i	j	结果
0	0	0 (OFF 假)
0	非 0	-1 (ON 真)
非 0	0	-1 (ON 真)
非 0	非 0	-1 (ON 真)

当两个值都是或其中任意一个是 ON(真)时, 结果是 ON(真)。

**i XOR j**      计算 i 与 j 之间的逻辑异或值

i	j	结果
0	0	0 (OFF 假)
0	非 0	-1 (ON 真)
非 0	0	-1 (ON 真)
非 0	非 0	0 (OFF 假)

仅当两个值中的一个为 ON(真)时，结果是 ON(真)。

**NOT i**      计算 i 的逻辑非值

i	结果
0	-1 (ON 真)
非 0	0 (OFF 假)

在 AS 中，一个值或表达式的逻辑状态可能按以下所示来表达：

真： not 0, ON, TRUE (非 0, 开, 真)

假： 0, OFF, FALSE (0, 关, 假)

## 8.4 二进制运算符

二进制逻辑运算符为两个数值的每个位执行逻辑操作。例如，如果一个数字由 4 个位组成，那么将要进行计算的值可能是 0000, 0001, 0010, 0011, ……，1111 (在 AS 中，数值由 32 位组成)。

二进制表达式	十进制表达式
0000	0
0001	1
0010	2
0011	3
⋮	⋮
1111	15

运算符	功能	示例
BOR	二进制或 (Binary OR)	i BOR j
BAND	二进制与 (Binary AND)	i BAND j
BXOR	二进制异或 (Binary XOR)	i BXOR j
COM	二进制非 (Binary complement)	COM i

### 示例

i BOR j                      如果 i=5, j=9, 那么结果是 13。

$$\begin{array}{r} i=5 \quad 0101 \\ j=9 \quad 1001 \\ \hline 1101 \Rightarrow 13 \end{array}$$

i BAND j                      如果 i=5, j=9, 那么结果是 1。

$$\begin{array}{r} i=5 \quad 0101 \\ j=9 \quad 1001 \\ \hline 0001 \Rightarrow 1 \end{array}$$

i BXOR j                      如果 i=5, j=9, 那么结果是 12

$$\begin{array}{r} i=5 \quad 0101 \\ j=9 \quad 1001 \\ \hline 1100 \Rightarrow 12 \end{array}$$

COM i                      如果 i=5, 那么结果是-6

$$\begin{array}{r} i=5 \quad 0 \cdots 0101 \\ \hline 1 \cdots 1010 \Rightarrow -6 \end{array}$$



## 8.5 变换值运算符

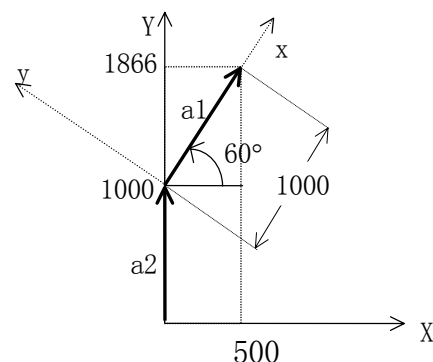
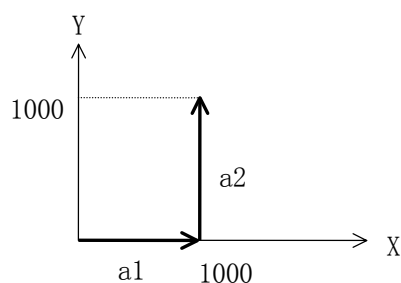
在 AS 系统中，运算符+ 和-用来确定复合变换值(XYZOAT 值)。但是注意，不像通常的加法和减法，互换法则并不适用于变换值运算操作。算术表达式“ $a+b$ ”和“ $b+a$ ”结果是一样的,但是“姿态  $a$ +姿态  $b$ ”与“姿态  $b$ +姿态  $a$ ”未必相等。这是因为在变换值操作中，轴的姿态被考虑了进去。下面是一个例子：

$$a1=(1000, \quad 0, \quad 0, \quad 0, \quad 0, \quad 0)$$

$$a2=( \quad 0, 1000, \quad 0, \quad 60, \quad 0, \quad 0)$$

$$a1+a2=(1000, 1000, \quad 0, \quad 60, \quad 0, \quad 0)$$

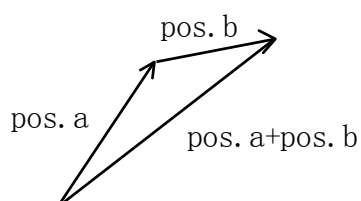
$$a2+a1=(500, 1866, \quad 0, \quad 60, \quad 0, \quad 0)$$



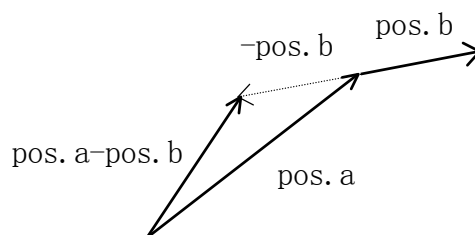
运算符	功能	示例
+	两个变换值相加	pos. a+pos. b
-	两个变换值相减	pos. a-pos. b

### 示例

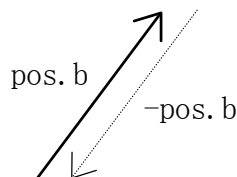
pos. a+ pos. b



pos. a-pos. b



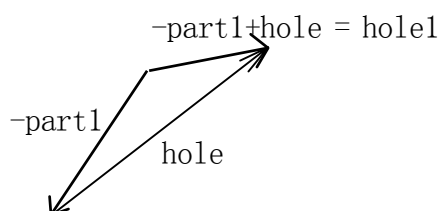
变换值运算符“-”用于一个值(例如 $-\text{pos. b}$ )，表示  $\text{pos. b}$  的反向值。例如，当变换值  $\text{pos. b}$  定义了对象 B 相对于对象 A 的姿态，那末  $-\text{pos. b}$  则定义了对象 A 相对于对象 B 的姿态。



在下例中，“hole1”被定义为相对于“part1”（事先已定义）。这可通过复合变换值  $\text{part1}+\text{hole}$  来实现。

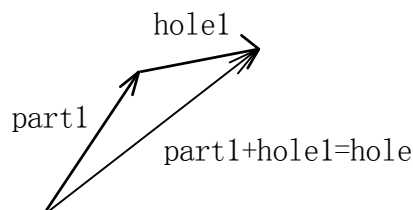
将机器人手动到希望定义为“hole1”的位姿，并用 HERE 指令把机器人当前位姿赋给“hole”。然后使用这个位姿（“hole”），定义“hole1”。

**POINT** hole1=  $-(\text{part1})+\text{hole}$



另一种方法不用运算符“-”也可以定义“hole1”，是把“hole1”写在 POINT 指令表达式的左边。下述指令也能定义“hole1”。

**POINT** part1+hole1=hole



## 8.6 字符串运算符

运算符	功能	示例
+	合并两个字符串	\$a = \$b + \$c

### 示例

\$a=\$b+\$c      合并\$b+\$c 并将它赋值给\$a。当\$b= “abc” ,  
\$c= “123” 时, \$a= “abc123” 。

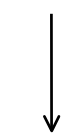
## 9.0 函数

本章介绍 AS 系统中使用的函数。这些函数通常结合监控指令和程序命令一起使用。它们用下面描述的格式表达。关键字指定了函数，括号中输入的参数决定了函数的值。

### 示例

关键字

参数



**SIG** (信号编号, .....)

用 [ ] 标记的参数可以省略。

关键字后面必须有一个空格。

## 9.1 实数函数

SIG	返回指定信号的逻辑与(AND)。
BITS	返回与信号的位格式对应的数值(最多可有 16 个信号)。
BITS32	返回与信号的位格式对应的数值(最多可有 32 个信号)。
TIMER	返回计时的当前值。
DISTANCE	返回两个点之间的距离。
DX, DY, DZ	返回变换值中的位移值(X, Y, Z)。
DEXT	返回一个位姿的指定分量(选项)。
ASC	返回 ASCII 码值。
LEN	返回字符串的长度。
TRUE, ON	返回值 -1.0, 代表逻辑真(TRUE)。
FALSE, OFF	返回值 0.0, 代表逻辑假(FALSE)。
VAL	把实数值返回到指定的字符串中。
INSTR	返回指定字符在字符串中的起始点之后所在的位置值。
MAXVAL	比较给定的数值。
MINVAL	比较给定的数值。
INT	返回一个数值的整数值。
MAXINDEX	返回指定维编号中的最大下标值。
MININDEX	返回指定维编号中的最小下标值。
SWITCH	返回系统开关的状态。
WHICHTASK	返回由程序或子程序选中的任务号。
TASK	返回程序执行的状态。
ERROR	返回错误代码。
PRIORITY	返回机器人程序的优先级。
UTIMER	返回计时的当前值。
MSPEED	返回机器人 1 的监控速度。
MSPEED2	返回机器人 2 的监控速度。
INRANGE	检查位姿是否处于运动范围内。
SYSDATA	返回 AS 参数(选项)。

EXISTJOINT	返回指定的以关节位移值定义的位姿变量是否存在。
EXISTTRANS	返回指定的以变换值定义的位姿变量是否存在。
EXISTREAL	返回指定的实数值变量是否存在。
EXISTCHAR	返回指定的字符串变量是否存在。
EXISTINTEGER	返回指定的整数值变量是否存在。
EXISTPGM	返回指定的程序是否存在。
STRTOPOS	返回指定字符串变量的位姿数据。
STRTOVAL	返回指定字符串变量的实数值。

SIG(信号编号, ……)

**功能**  
返回指定的二进制信号状态的逻辑与 (AND)。

**参数**  
信号编号  
指定外部或内部 I/O 信号的编号。

**说明**  
计算所有指定的二进制信号状态的逻辑与，并返回结果值。如果所有指定信号的状态为真，则返回-1 (真(TRUE) 的值)，否则返回值为 0 (假(FALSE) 的值)。可用下面显示的外部或内部 I/O 信号的编号指定。

容许的信号编号	
外部输出信号	1—实际信号数
外部输入信号	1001—实际信号数
内部信号	2001—2960

用正编号指定的信号为 ON 状态时，被看作为真。用负编号指定的信号为 OFF 状态时，被看作为假。对应于“0”的信号编号是没有的，所以该信号总被认为真。

— [ 注 意 ] —  
同时评估一个以上的信号时，有如下时间限制：多个信号同时输入时，每个信号间大约有 2 毫秒的稳定时间差。

**示例**  
如果二进制 I/O 信号 1001=ON, 1004=OFF, 20=OFF, 那么

	结果	
SIG(1001)	-1	真
SIG(1004)	0	假
SIG(-1004)	-1	真
SIG(1001, 1004)	0	假
SIG(1001, -1004)	-1	真
SIG(1001, -1004, -20)	-1	真

**BITS(起始信号编号, 信号数量)**

**功能**  
读取连续的二进制信号，并返回与指定的二进制信号位格式相对应的十进制数值。

**参数**  
起始的信号编号  
指定要读取的第一个信号。

信号数量  
指定要读入信号的数量。允许的最大数量为 16。读取的信号多于 16 个时，用 BIT32 函数，将在下一单元中说明。

**说明**  
该函数返回与指定信号位格式相对应的十进制数值。该函数返回的数值在二进制表达式中，最低位对应于起始信号编号。

容许的信号编号	
外部输出信号	1—实际信号数
外部输入信号	1001—实际信号数
内部编号	2001—2960

[ 注 意 ]

同时评估一个以上的信号时，有如下时间限制：多个信号同时输入时，每个信号间大约有 2 毫秒的稳定时间差。

**示例**  
如果信号状态如下，下面表达式的结果将为 5。  
x=BITS(1003, 4)

以 1003 起始的 4 个信号(如：1003, 1004, 1005 和 1006)的逻辑值被读作二进制位格式的 0101 或十进制的 5。

信号：	1008	1007	1006	1005	1004	1003	1002	1001
状态：	1	1	0	1	0	1	0	0



---

### **BITS32 (起始信号编号, 信号数量)**

---

#### **功能**

读取连续的二进制信号，并返回与指定的二进制信号位格式相对应的十进制数值。

#### **参数**

起始信号编号

指定要读取的第一个信号。

信号数量

指定要读入信号的数量。允许的最大数量为 32。

#### **说明**

参看 BITS 函数。

---

## TIMER(计时编号)

---

### 功能

以秒为单位返回指定计时的值。表达了执行该 TIMER 函数时刻的计时值。

### 参数

计时编号

指定要读取哪一个计时，允许的编号为 0—10。

### 说明

通过使用 TIMER 函数，可以在任意时刻读取计时的值。读取并返回从先前用 TIMER 命令设定的时间值算起所用的时间(以秒为单位)。若 TIMER 命令不曾设定过，则返回的计时值为 0。

编号 0 的计时(计时 0)用作系统时钟。其返回的值为系统启动后的时间。

### 示例

在下例中，用 TIMER 命令和实数函数测量子程序的执行时间。

```
TIMER (1)=0           设定计时 1 值为 0
CALL  test.routine     调用子程序
PRINT "Elapsed time=", TIMER(1),"seconds"
```

---

### **DISTANCE (变换值变量 1, 变换值变量 2)**

---

#### **功能**

计算以变换值表示的两个位姿之间的距离。

#### **参数**

变换值变量 1, 变换值变量 2

指定以变换值表示的两个位姿变量的名称, 计算他们之间的距离。

#### **说明**

以毫米为单位返回两个位姿之间的距离(两个位姿可以以任意次序输入)。

#### **示例**

k=DISTANCE (HERE, part)                      计算当前 TCP 至 “part” 之间的距离, 将结果赋给 k。

DX(变换值变量)

DY(变换值变量)

DZ(变换值变量)

功能

返回指定位姿的变换值的 (X, Y, Z) 分量。

参数

变换值变量

指定以变换值定义的位姿变量的名称，需要获取其 X、Y、或 Z 方向分量。

说明

这三个函数分别返回指定变换值变量值的 X、Y、或 Z 的分量。

[ 注 意 ]

变换值的各分量可用 DECOMPOSE 命令获取。用 DECOMPOSE 命令可以获得 O、A、和 T 值。

示例

如果变换值变量 start 有如下变换值：

X	Y	Z	O	A	T
125,	250,	-50,	135,	50,	75
x=DX(start)		DX 函数返回 x=125.00			
y=DY(start)		DY 函数返回 y=250.00			
z=DZ(start)		DZ 函数返回 z= -50.00			

DEXT (位姿变量, 元素编号)

**功能**  
返回指定位姿的指定元素。

**参数**  
**位姿变量**  
指定由关节位移值或变换值定义的位姿变量。

**元素编号**  
指定返回实数值的元素，见下表。

元素编号	位姿	
	变换值	关节位移值
1	X 分量	JT1
2	Y 分量	JT2
3	Z 分量	JT3
4	O 分量	JT4
5	A 分量	JT5
6	T 分量	JT6
7	JT7	JT7

**示例**  
如果“aa” 的变换值为 0, 0, 0, -160, 0, 0, 300，然后输入下面的函数：

```
type DEXT(aa, 7)
```

这将返回变量“aa” 的 JT7 值 300。

---

### ASC(字符串, 字符编号)

---

#### 功能

返回字符串表达式中指定字符的 ASCII 数值。

#### 参数

字符串

指定包含所需 ASCII 数值字符的字符串。如果该字符串为空串，或参数“字符编号”的指定编号超过了字符串中字符的实际编号，则返回值为-1。

字符编号

指定字符的编号，从字符串的第一个字符算起。如果未指定、或指定为 0 或者 1，则返回的值为该字符串中第一个字符的 ASCII 值。

#### 说明

ASCII 值以实数形式返回。

#### 示例

ASC("sample", 2)                      返回字符“a”的 ASCII 值。

ASC(\$name)                              返回字符串“\$name”第一个字符的 ASCII 值。

ASC(\$system, i)                        返回字符串变量“\$system”中第 i 个字符的 ASCII 值。

---

### LEN(字符串)

---

#### 功能

返回指定字符串中字符的数量。

#### 参数

字符串

指定一个字符串、字符串变量、或字符串表达式。

#### 示例

LEN("sample")                        返回字符串“sample”中字符的数量，其个数为 6。

---

. . . **TRUE**. . .  
. . . **ON**. . .

---

### 功能

返回 TRUE (真) (-1) 的逻辑值。

### 说明

当必须指定逻辑条件为真 (TRUE) 时，该函数非常方便。

TRUE 和 ON 函数的结果相同 (选择最适合程序需要的函数)。

---

. . . **FALSE**. . .  
. . . **OFF**. . .

---

### 功能

返回 FALSE (假) (0) 的逻辑值。

### 说明

当必须指定逻辑条件为假 (FALSE) 时，该函数非常方便。

FALSE 和 OFF 函数的结果相同 (选择最适合程序需要的函数)。

VAL (字符串, 代码)

功能

返回指定字符串中的实数值。

参数

字符串

指定字符串、字符串变量、或字符串表达式。

代码

指定用实数值或表达式表达的返回数值的进制。若未指定，或指定数值为 0、1 或 2 以外的其他数值，则默认值为 0。

数值	进制
0	十进制
1	二进制
2	十六进制

[ 注 意 ]

在字符串中可以使用科学计数法。

指定进制的代码(如：<sup>^</sup>B 和<sup>^</sup>H)可以加在字符串的开头。

读到不是数字或不是进制代码的任何字符，将被认为是字符串的结束标志。

示例

VAL ("123 ")	返回实数值 123。
VAL ("123abc ")	返回实数值 123。
VAL ("12 ab 3 ")	返回实数值 12。
VAL ("1. 2E-5")	返回实数值 0. 00001。
VAL (" ^ HFF")	返回实数值 255。( ^H 代表十六进制，16×15+15=255)



---

## INSTR(**开始点**, 字符串 1, 字符串 2)

---

### 功能

以实数值的形式返回指定字符串在给定字符串中的位置。

### 参数

#### 开始点

指定在字符串 1 中从哪里开始寻找字符串 2 的位置。若未指定，将从字符串 1 的首个字符开始查找。

#### 字符串 1

用字符串、字符串变量、或字符串表达式表示。指定要在里面搜索字符串 2 的字符串。

#### 字符串 2

用字符串、字符串变量、或字符串表达式表示，来指定待寻找的字符串。如果指定为空串，则返回起点的值。若字符串未指定，则返回值为 1。

### 说明

当字符串 2 包含在字符串 1 中时，本命令返回的值为字符串 2 在字符串 1 中起点的数值。

若字符串 2 没有包含在字符串 1 中，则返回值为 0。

如果指定的起点值小于或等于 1，则从字符串 1 首位开始搜寻。如果指定的起点值大于字符串 1 的字符数，则返回值为 0。

不区分字母的大小写。

### 示例

INSTR("file.ext", ".")	返回实数值 5。
INSTR("file", ".")	返回实数值 0。
INSTR("abcdefgh", "DE")	返回实数值 4。
INSTR(5, "1-2-3-4", "-")	返回实数值 6。

---

**MAXVAL (实数值 1, 实数值 2, .....)**

---

**功能**

比较给出的实数值，并返回其中最大的值。

**参数**

实数值 1, 实数值 2, .....

指定要比较的实数值。

---

**MINVAL (实数值 1, 实数值 2, .....)**

---

**功能**

比较给出的实数值，并返回其中最小的值。

**参数**

实数值 1, 实数值 2, .....

指定要比较的实数值。

---

INT (数值表达式)

---

**功能**  
返回指定数值表达式的整数值。

**参数**  
数值表达式

**说明**  
返回整数值 (如：如果值不是科学计数法格式时，返回值为小数点左边的数)。除非为 0，负号保留，与和整数一起返回。

**示例**

INT (0. 123)	返回 0
INT (10. 8)	返回 10
INT (-5. 462)	返回-5
INT (1. 3125E+2)	返回 131
INT (cost+0. 5)	返回 “cost” 经四舍五入后的整数值

---

MAXINDEX (字符串变量, 维编号)

---

**功能**  
返回数组中指定维编号的最大下标值。

**参数**  
字符串变量  
指定数组变量的名称。

**维编号**  
指定维编号 (1-3)。  
若指定的值不在 1 到 3 之间，则出错。若未指定，默认值为 1。

**说明**  
如果数组已定义，则返回数组中指定维编号的最大下标值。如果变量不是数组，返回 - 1；若变量未定义，则返回 - 2。

### 示例

变量#pos 是以关节位移值表示的，是一个从#pos[0] 到 #pos[100] 的一维数组。维编号被省略。

ret= MAXINDEX (“#pos”)                      变量 ret 的值为 100。

变量#place 是以关节位移值表示的，是一个从#place[1, 1] 到 #place[1, 5] 的二维数组。

ret=MAXINDEX (“#place”, 2)                  变量 ret 的值为 5。

变量#place 是以关节位移值表示的，是一个从#place[2, 1, 10] 到 #place[2, 1, 20] 的三维数组。

ret=MAXINDEX (“#place”, 3)                  变量 ret 的值为 20。

以下程序显示变量 pos 的变换值，pos 是一个一维数组。

```
. PROGRAM index ()
max=MAXINDEX ("pos", 1)
min=MININDEX ("pos", 1)
FOR j=min TO max
    $val="pos"+ "["+$ENCODE (/12, j)+"]"
    ret=EXISTTRANS ($val)
    IF ret==FALSE THEN
        GOTO continue
    END
    DECOMPOSE x[0]=pos[j]
    TYPE "pos[" , j, "]=", /F8. 3, x[0], /X3, /F8. 3, x[1], /X3, /F8. 3, x[2]
continue:
END
. END
```

---

## MININDEX(字符串变量, 维编号)

---

### 功能

返回数组中指定维编号的最小下标值。

### 参数

字符串变量

指定数组变量的名称。

维编号

指定维编号(1-3)。

若指定的值不在 1 到 3 之间，则出错。若未指定，默认值为 1。

### 说明

如果数组已定义，则返回数组中指定维编号的最小下标值。如果变量不是数组，返回 -1；若变量未定义，则返回 -2。

### 示例

变量#pos 是以关节位移值表示的，是一个从 #pos[0] 到 #pos[100] 的一维数组。

ret=MININDEX("#pos", 1)                      变量 ret 的值为 0。

变量#place 是以关节位移值表示的，是一个从 #place[1,1] 到 #place[1,5] 的二维数组。

ret=MININDEX("#place", 2)                      变量 ret 的值为 1。

变量#place 是以关节位移值表示的，是一个从 #place[2,1,10] 到 #place[2,1,20] 的三维数组。

ret=MININDEX("#place", 3)                      变量 ret 的值为 10。

**SWITCH(开关名称)**

**功能**

返回指定系统开关的当前状态。

**说明**

如果开关为 ON，返回值为-1；如果为 OFF，则返回值为 0。

**WHICHTASK 程序名称**

**功能**

返回由指定程序(子程序)选定的任务编号。

**参数**

程序名称以字符串变量的形式指定程序或子程序的名称。变量名称应以\$开头。

**说明**

返回任务编号的实数值。

1: 机器人程序 (机器人 1)

2: 机器人程序 (机器人 2)

1001: PC 程序 1

1004: PC 程序 4

1002: PC 程序 2

1005: PC 程序 5

1003: PC 程序 3

-1: 执行的任务不存在。

**示例**

`task_no=WHICHTASK ($pg_name)`

如果由程序 “\$pg\_name” 选定的任务存在，  
则把任务编号保存在变量中。若不存在则  
task\_no= -1。

---

### TASK (任务编号)

---

#### 功能

返回由任务编号指定的程序执行状态。

#### 参数

任务编号

1: 机器人 1

2: 机器人 2

1001:PC 程序 1                      1004:PC 程序 4

1002:PC 程序 42 1005:PC 程序 5

1003:PC 程序 43

#### 说明

此函数返回程序的执行状态。例如，此函数可以用于从机器人控制程序中监视 PC 程序的执行状态。然后可以根据 PC 程序的状态设置机器人操作的条件。

该函数返回的值如下：

0	未执行
1	程序运行中
2	程序执行暂停中
3	程序步骤的执行已完成，等待机器人动作完成。

---

### ERROR

---

#### 功能

返回当前错误的错误代码。

#### 说明

当前有错误发生时，返回当前出错中的错误代码。没有错误发生时返回值为 0。

#### 示例

TYPE \$ERROR (ERROR)                      TYPE 命令显示由 ERROR 函数返回的错误代码的出错信息。

---

## **PRIORITY**

---

### **功能**

返回当前机器人程序的优先级编号。

### **说明**

以实数形式返回堆栈内当前选定的机器人程序的优先级编号。在 PC 程序中没有优先级设置。

默认的机器人程序优先级编号为 0。可以通过 LOCK 命令改变优先级编号。

---

## **UTIMER (@计时变量)**

---

### **功能**

返回由 UTIMER 命令设定的@计时变量的当前值。

### **参数**

@计时变量

指定由 UTIMER 命令设定的变量的名称。把@符号加在变量名称的前面以便指定一个整数型变量。



---

**M SPEED**

**M SPEED2**

---

### 功能

返回当前的监控速度 (0—100 %)。

MSPEED 用于机器人 1，MSPEED2 用于机器人 2。

### INRANGE (位姿变量, 关节位移值变量)

## 功能

检查位姿是否在机器人的运动范围之内，并根据此检查结果返回一个数值(参看下表)。

### 参数

## 位姿变量

指定要检查的位姿(关节位移值变量、变换值变量、或复合变换值变量)。

关节位移值变量

仅在指定的位姿变量为变换值或复合变换值时，输入本参数。由给定的关节位移值计算出机器人的形态。如果未指定，则使用当前的机器人形态。

## 说明

该函数返回的数值如下表:

0	在运动范围之外。
1	JT1 在运动范围之外。
2	JT2 在运动范围之外。
4	JT3 在运动范围之外。
8	JT4 在运动范围之外。
16	JT5 在运动范围之外。
32	JT6 在运动范围之外。
16384	超出了碰撞检测范围。
32768	超出了机器人手臂到达范围。

〔注意〕—

该函数检查位姿是否在运动范围内，但不检查到达该位姿的路径是否在运动范围之内。

### 示例

```
IF INRANGE(pos1, #p) GOTO ERR STOP:
```

ERR STOP:

TYPE "pose pos1 is out of motion range."

PAUSE

如果位姿 pos1 超出运动范围，跳转到标签 ERR\_STOP 处，显示信息，并停止机器人。

---

## **SYSDATA(关键字, opt1, opt2)**

---

### 功能

根据给定的关键字返回 AS 系统中的指定参数。

### 参数

关键字、opt1、opt2

#### M. SPEED

返回监控速度(以百分比的形式)。如果不在执行运动步骤，返回-1。

Opt 1: 机器人编号(从 1 到机器人数量)。若未输入，默认为 1。

Opt 2: 不使用。

#### MSTEP

返回执行中的运动步骤的步骤编号，或者正在执行程序中的上一运动步骤的编号。若不存在这种步骤，返回-1。

Opt 1: 机器人编号(1 到机器人数量)。若未输入，默认为 1。

Opt 2: 不使用

#### STEP

返回执行中的运动步骤的步骤编号，或者正在执行程序中的上一运动步骤的编号。若不存在这种步骤，返回-1。

Opt 1: 机器人编号(1 到机器人数量)或 PC 任务号(1001 到 PC 程序数量)。若未输入，默认为 1。

Opt 2: 不使用

#### P. SPEED

返回当前运动或下一执行运动的运动速度(百分比)。如果速度单位被设置为时间，返回-1。

Opt 1: 机器人编号(1 到机器人数量)。若未输入，默认为 1。

Opt 2: 不使用

#### P. SPEED. M

返回当前运动或下一执行运动的运动速度(单位为 MM/S)，如果速度单位被设置为时间，返回-1。

Opt 1: 机器人编号(1 到机器人数量)。若未输入，默认为 1。

Opt 2: 不使用

**EXISTJOINT (“关节位移值变量名称”)**

**功能**

检查指定的关节位移值变量是否存在。

**参数**

关节位移值变量名称（以关节位移值定义的）  
以字符串的形式指定以关节位移值变量的名称。变量名称应包含在引号内，并以 # 开头。

**说明**

如果变量名称存在，返回-1。若不存在，则返回 0。

[ 注 意 ]

在指定数组变量时，有以下限制：

1. 指定数组元素的范围，不能省略。
2. 最小值写在冒号(:)的左边，最大值写在右边。

例如：如果实型变量 r 为一个三维数组，有三个元素 r[1, 1, 1]，r[1, 1, 2]，r[1, 1, 3]。

检查结果 OK 的将是	r[1, 1, 1:3]	
检查结果 NG 的将是	r[1, 1, 1:]	； 范围未指定
	r[1, 1, *]	； 范围未指定
	r[1, 1, 3:1]	； 指定范围次序错(从大到小)

**示例**

ret=EXISTJOINT (“#pos”)

如果以关节位移值变量#pos 存在，ret=-1，否则  
ret=0。

以下显示的位姿变量#place 是一个二维数组，从#place[1,1] 到 #place[1,5]。

ret=EXISTJOINT (“#place”)	ret 将等于-1。
ret=EXISTJOINT (“#place[1,1]”)	#place[1,1]存在，因此 ret=1。
ret=EXISTJOINT (“#place[1,1:5]”)	#place[1,1]到 #place[1,5]存在，因此 ret=1。
ret=EXISTJOINT (“#place[1,1:6]”)	#place[1,6]不存在，因此 ret=0。

---

### EXISTTRANS (“变换值变量名称”)

---

#### 功能

检查指定的以变换值变量是否存在。

#### 参数

变换值变量名称

以字符串的形式指定以变换值变量的名称，变量名称应该用引号括起来。

#### 说明

如果变量存在，返回-1。如果不存在，返回 0。

在指定数组变量时，请参看 EXISTJOINT 的限制和示例。

#### 示例

`ret=EXISTTRANS (“pos1”)`      如果以变换值变量 pos1 存在， `ret=-1`。  
如果其不存在， `ret=0`。

---

### EXISTREAL (“实型变量名称”)

---

#### 功能

检查指定的实型变量是否存在。

#### 参数

实型变量名称

以字符串形式指定实型变量的名称。变量名称应被括在引号内。

#### 说明

若变量名称存在，返回-1，若不存在，返回 0。

在指定数组变量时，请参看 EXISTJOINT 的限制和示例。

#### 示例

`ret=EXISTREAL (“pp”)`      若实型变量 pp 存在， `ret=-1`；若不存在，则 `ret=0`。

---

### EXISTCHAR(“字符串变量名称”)

---

#### 功能

检查指定的字符串变量是否存在。

#### 参数

字符串变量名称

以字符串形式指定字符串变量的名称。变量名称应包含在引号中，并以\$开头。

#### 说明

如果变量名称存在，返回-1；若不存在，则返回 0。

在指定数组变量时，请参看 EXISTJOINT 的限制和示例。

#### 示例

ret=EXISTCHAR(“\$val”)      如果字符串变量\$val 存在，ret=-1。  
                                 如果不存在，ret=0。

---

### EXISTINTEGER(“整型变量名称”)

---

#### 功能

检查指定的整型变量是否存在。

#### 参数

整型变量名称

以字符串形式指定整型变量的名称。变量名称应包含在引号中，并以@开头。

#### 说明

如果变量名称存在，返回-1；如果不存在，返回 0。

在指定数组变量时，请参看 EXISTJOINT 的限制和示例。

#### 示例

ret=EXISTINTEGER(“@abc”)      如果整型变量@abc 存在 ret=-1；如果不存在，ret  
                                 =0。

---

### EXISTPGM (“程序名称”)

---

#### 功能

返回指定的程序是否存在。

#### 参数

程序名称

以字符串形式指定程序(或子程序)的名称。

#### 说明

如果指定的程序或子程序存在，返回-1，如果不存在，返回 0。

#### 示例

ret=EXISTPGM(“pg1”)                      如果程序 pg1 存在，ret=-1；如果不存在，ret=0。

---

### STRTOPOS (字符串变量)

---

#### 功能

返回由字符串变量指定的位姿变量的值。

#### 参数

字符串变量指定一个字符串变量以获取指定的位姿值，字符串变量名称应以\$开头。

#### 说明

如果位姿变量已经定义了，则返回位姿变量值。如果位姿变量未定义，则产生错误。

#### 示例

```
HERE #pos  
$A=“#pos”
```

JMOVE STRTOPOS (\$A)

字符串 “\$A” 指定了 “#pos”。机器人向关节位移值  
“#pos” 描述的目标位姿移动。如果 “#pos” 没有定义，  
则产生错误。

---

### STRTOVAL (字符串变量)

---

#### 功能

返回指定的变量中的实数值。

#### 参数

字符串变量指定字符串变量以获取指定的实数值，字符串变量名称应以\$开头。

#### 说明

如果实型变量已经定义，则返回其实数值；如果实型变量未定义，则产生错误。

#### 示例

VAR=5

\$VA="VAR"

total=STRTOVAL (\$VA)+6

字符串变量“\$VA”指定了实型变量“VAR”。因为实型变量“VAR”值为 5，所以实型变量“total”值为 11。如果变量“VAR”未定义，则产生错误。



## 9.2 位姿值函数

DEST	以变换值的形式返回目标位姿。
#DEST	以关节位移值的形式返回目标位姿。
FRAME	返回框架坐标系的变换值。
NULL	返回空变换值。
HERE	返回当前位姿的变换值。
#HERE	返回当前位姿的关节位移值。
TRANS	返回由给定分量组成的变换值。
RX, RY, RZ	返回表示绕轴旋转的变换值。
#PPOINT	返回由给定分量组成的关节位移值。
SHIFT	返回由原来位姿平移产生的变换值。
AVE_TRANS	返回两个位姿的变换值的平均值。
BASE	返回基础变换值。
TOOL	返回工具变换值。
TRADD	返回的值为 X 分量加上走行轴的值(选项)。
TRSUB	返回的值为 X 分量减去走行轴的值(选项)。
#HOME	返回原点(HOME) 位姿的关节位移值。
CCENTER	返回由给定位姿描述的圆的圆心变换值(选项)。
CSHIFT	返回朝着由给定位姿描述的圆的圆心位置平移而得到的位姿的变换值(选项)。

DEST

#DEST

功能

DEST: 返回机器人当前动作的目标位姿的变换值。  
#DEST: 返回机器人当前动作的目标位姿的关节位移值。

说明

当机器人运动因某种原因中断后，使用这两个函数可以找出机器人的目标位置。这两个函数可以用于所有的机器人动作。

[ 注意 ]

机器人停止时的位姿和由 DEST/ #DEST 函数返回的位姿值并不总是相等。例如：如果 HOLD/RUN(运转/保持)的状态由 RUN 状态转换到 HOLD 状态，机器人立即停止，但是 DEST/ #DEST 函数返回的位姿是描述了机器人停止前要前往的位姿。

示例

当要恢复被 ONI...CALL 命令中断的运动时，使用 DEST/ #DEST 函数是很方便的。执行了下面子程序包含的命令之后，机器人可以恢复其被中断的动作。

POINT save=HERE	将当前位姿存储在“save”中。
POINT old=DEST	将目标位姿存储在“old”中。
JDEPART 50.0	后退 50 毫米(关节插补)。
:	
:	
JAPPRO save, 50.0	接近到“save”位姿上方 50 mm 的位姿(关节插补)。
LMOVE save	移动到“save”位姿处(直线插补)。
LMOVE old	移动到“old”位姿处(直线插补)。

---

**FRAME (变换值变量 1, 变换值变量 2, 变换值变量 3, 变换值变量 4)**

---

### 功能

返回相对于基础坐标系的框架坐标系(相对坐标系)的变换值。注意仅用位姿变量的变换值的变换分量作为决定框架坐标的位置信息。

### 参数

变换值变量 1, 变换值变量 2

指定以变换值变量来决定 X 轴的方向。框架坐标系的 X 轴, 被设定为通过这两点。X 轴的正方向为从变换值变量 1 的点面向变换值变量 2 的点的方向。

变换值变量 3

指定以变换值变量来决定 Y 轴的方向。设定框架坐标系的 Y 轴, 使这三个点: 位姿 1、位姿 2、位姿 3 在 XY 平面上, 位姿 3 在 Y 轴的正方向上。

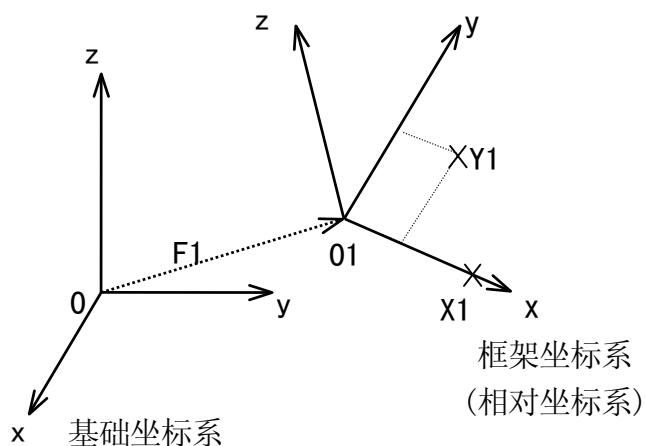
变换值变量 4

指定以变换值变量来指定框架坐标系的原点, 它等于此函数的 X, Y, Z 返回值。

### 说明

POINT F1=FRAME (01, X1, Y1, 01)

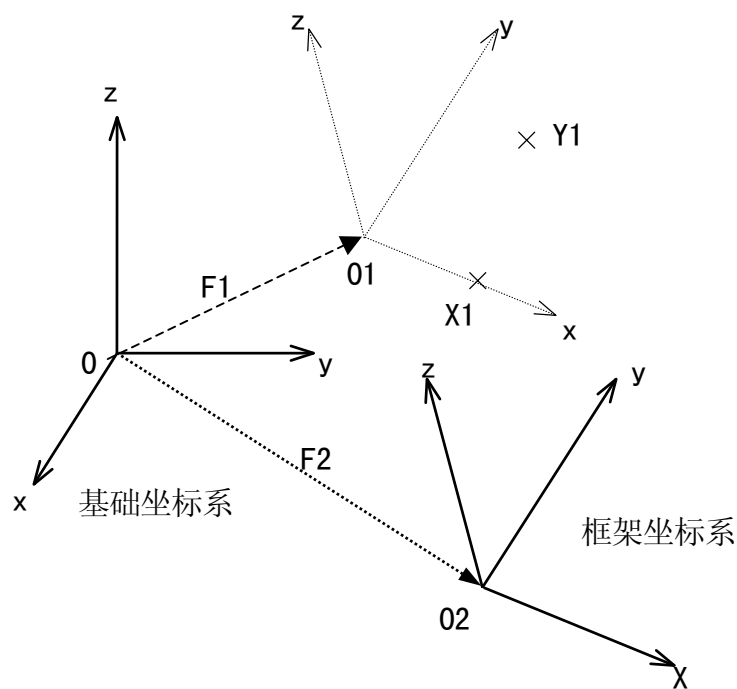
设定如下图的框架坐标系。



如果在框架坐标系中的位姿被示教为  $F1+A$ ，则当坐标系改变时(例如零件站移动了)，只需重新示教  $F1$ 。(参看 11.6 用框架坐标表示相对位姿)。

**POINT F2=FRAME (01, X1, Y1, 02)**

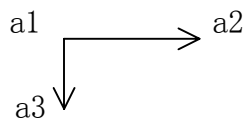
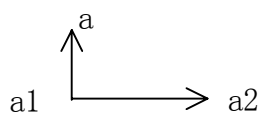
该命令设定的框架坐标系如下图所示。



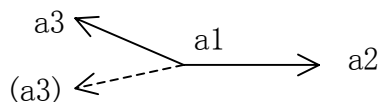
[ 注意 ]

定义框架坐标系时，注意以下几点：

**POINT F=FRAME(a1, a2, a3, a1)**



注意在以上两个框架坐标系中的 Y 轴和 Z 轴的方向相反，取决于 a3 示教的位置。因此，当 a3 点被示教得靠近 X 轴时(如下图)，Z 轴的方向不一定是所要的方向。因此程序在使用框架坐标系之前，必须要检查框架坐标系是否正确。



a1, a2, a3 三点定义了工具坐标系原点得位置。当重新定义框架坐标系时，工具变换值必须和 a1, a2, a3 最初示教的时候相同。

为了更加精确，应该将 a1, a2, a3 三点示教得尽可能分开。特别是 a3 点应该靠近 Y 轴而尽量远离 X 轴。

在示教 a1, a2, a3 点时，工具坐标系的原点应该被定义在易于观察的点上，例如在工具的顶点等。

对于某些工具，即使其已经用工具变换值平移，也很难确定工具坐标系的原点。在这种情况下，以空工具示教 a1, a2, a3，但请注意，这时工具坐标系的原点位于法兰面的中心。

---

## NULL

---

### 功能

返回空变换值。

### 说明

返回所有平移分量和旋转分量都为 0 ( $X=Y=Z=0$ ,  $0=A=T=0$ ) 的变换值。本函数和 SHIFT 函数一起使用时，可以方便地重新定义变换值。可以平移坐标而不改变旋转分量 (OAT)。

### 示例

```
POINT new=SHIFT(NULL BY x.shift, y.shift, z.shift)+old
```

通过沿基础坐标系平移位姿 “old” 指定的距离，来定义变量 “new”。

```
dist=DISTANCE(NULL, test.pos)
```

计算位姿 “test.pos” 和机器人的空原点 (0, 0, 0, 0, 0, 0) 之间的距离，并将该值赋给变量 dist。

---

**HERE**

---

---

**#HERE**

---

### 功能

HERE : 返回描述工具坐标系当前位姿的变换值。

#HERE : 返回描述工具坐标系当前位姿的关节位移值。

### 说明

在此函数执行的时刻，将读入编码器的值。因此请注意，此函数的返回值，代表了函数执行时机器人的所处的位姿。

#### [ 注 意 ]

“here” 不能用作程序名称或变量名称。

### 示例

dist=DISTANCE (HERE, pos1)

计算 “pos1” 和当前位姿之间的距离，并赋给变量 “dist”。

---

**TRANS(X 分量, Y 分量, Z 分量, O 分量, A 分量, T 分量)**

---

**功能**

返回含有指定的平移分量和旋转分量的变换值。

**参数**

X 分量, Y 分量, Z 分量

指定平移分量 X、Y、Z。未指定时默认为 0。

O 分量, A 分量, T 分量

指定旋转分量 O, A, T。未指定时默认为 0。

**说明**

变换值计算自各参数中指定的值。新的变换值可用来定义位姿变量，或者复合变换值中或运动命令中位姿变量。此函数和 DECOMPOSE 命令一起应用时很方便(参看#PPOINT 函数的示例)。

**示例**

`POINT temp.pos=TRANS(v[0], v[1]+100, v[2], v[3], v[4], v[5])`

数组变量 v[0]—v[5]，被转换成变换值 temp.pos，其中 v[1]加上 100 作为 Y 分量。



---

**RX(角度)**

**RY(角度)**

**RZ(角度)**

---

### 功能

返回代表绕指定轴旋转的变换值。

### 参数

角度

以度为单位指定旋转量。

### 说明

本函数中的 X、Y、Z 代表坐标系的坐标轴，返回的是绕指定轴的旋转量。此函数不返回平移量(X、Y、Z=0)。

### 示例

**POINT x\_rev=RX(30)**

返回代表绕 X 轴 30°旋转量的变换值，并将该值赋给  
“x\_rev”。

**#PPOINT(jt1, jt2, jt3, jt4, jt5, jt6)**

功能

返回指定的关节位移值。

参数

jt1	}	指定各关节(jt1—jt6)的值(旋转关节的角度)
jt2		
jt3		
:		
jt6		
		若未指定，默认为 0。

说明

此函数返回指定的关节位移值。各数值分别代表各关节的位移值，从关节 1(JT1)至最后关节(未必为 6)。

[ 注 意 ]

#PPOINT 函数仅用在关节位移值中，因此，一定要以“#”作为函数的开头。

示例

如下程序中，六关节机器人的第 2、3 关节从当前位姿移动指定的量。

```

HERE #ref          将当前位姿保存到存储器中(#ref)。
DECOMPOSE x[0]=#ref  将各分量值分解到数组变量 x[0], ……x[5]中。

JMOVE #PPOINT (X[0], x[1]+a, x[2]-a/2, x[3], x[4], x[5])

```

以下两个命令的执行结果与上面的程序一样，但与上面程序不同的是，两个关节不能同时运动。

```

DRIVE 2, a, 100      关节 2 移动 a°。
DRIVE 3, -a/2, 100   关节 3 移动 -a/2°。

```

---

**SHIFT(变换值变量 BY X 平移量, Y 平移量, Z 平移量)**

---

**功能**

返回将参数“变换值”描述的位姿沿着各基础坐标轴(X、Y、Z)平移指定的距离而得到的位姿的变换值。

**参数**

变换值变量

指定以变换值变量，用于平移的位姿。

X 平移量, Y 平移量, Z 平移量

指定基础坐标系的在 X, Y, Z 方向上的平移总量。如果省略，默认为 0。

**说明**

X 平移量、Y 平移量、Z 平移量被加至给定以指定变换值变量的 X、Y、Z 移动分量中。结果以变换值的形式返回。

**示例**

如果变换值变量“x”的变换值为 (200, 150, 100, 10, 20, 30)，那么

```
POINT y=SHIFT(x BY 5, -5, 10)
```

“x”按指定值平移到 (205, 145, 110, 10, 20, 30)，并且这些值被赋给变量“y”。

---

### **AVE\_TRANS (变换值变量 1, 变换值变量 2 )**

---

#### **功能**

返回两个变换值的平均值。

#### **参数**

变换值变量 1, 变换值变量 2

指定以变换值变量将求平均值的两个变换值。

#### **说明**

此函数计算变换值变量 1 和 2 各分量的中间点位姿的变换值。

此函数通常用于计算传感器检查时所得位姿信息的平均值。

#### **[ 注 意 ]**

对于 XYZ 分量, 平均值为各分量分别相加并除以 2 而得到。对于 OAT 分量, 不能用这种方式获取。

#### **示例**

POINT x=AVE\_TRANS (p, q)

JMOVE AVE\_TRANS (p, q)

---

## BASE

---

### 功能

返回当前基础变换值。

### 示例

`point a=BASE`      将当前基础变换值赋予变量 “a”。

---

## TOOL

---

### 功能

返回当前工具变换值。

### 示例

`point aa=TOOL`      将当前工具变换值赋予变量 “aa”。

---

### TRADD(变换值变量)

---

选项

#### 功能

返回走行轴位移值和以变换值变量的 X 分量的和。

#### 参数

变换值变量

指定用于加到走行轴上的 X 分量的，以变换值变量。

---

### TRSUB(变换值变量)

---

选项

#### 功能

返回从变换值的 X 分量中减去走行轴位移值后的值。

#### 参数

变换值变量

指定用于从中减去走行轴 X 分量的以变换值变量。

---

## #HOME (原点位姿编号)

---

选项

### 功能

返回当前设定的原点位姿的位置值。

### 参数

原点位姿编号

指定原点位姿编号。

- 1: 指定用 SETHOME 指令设定的原点位姿 1
  - 2: 指定用 SETHOME2 指令设定的原点位姿 2
- 若未指定，默认选择 1

### 说明

以关节位移值形式返回当前设定的原点位姿的位姿。

---

### [ 注 意 ]

---

#HOME 函数仅用关节位移值表示。因此，一定要以“#”作为函数的开头。

### 示例

在以下程序中，机器人不直接运动到原点位姿，而是先运动到与原点位姿相同高度处(仅在 Z 轴方向上)，然后再移动到原点位姿。

```
POINT homepos=#HOME(1)
IF DZ(homepos) > DZ(HERE) THEN
HERE tmp
POINT/Z tmp=homepos
LMOVE tmp
END
HOME
```

---

**CCENTER(变换值变量 1, 变换值变量 2, 变换值变量 3, 变换值变量 4)**

---

选项

**功能**

返回由 3 个指定的位姿形成的弧线的圆心。

**参数**

变换值变量 1, 2, 3

指定用于决定弧线上的 3 个点的以变换值变量。

变换值变量 4

指定用于确定机器人形态的变换值变量。

---

**CSHIFT(变换值变量 1, 变换值变量 2, 变换值变量 3, 变换值变量 4 BY 平移量)**

---

选项

**功能**

返回从目标位姿平移指定数值后得到的位姿。

**参数**

变换值变量 1, 2, 3

指定用于确定弧线上的 3 个点的以变换值变量。

变换值变量 4

指定用于以变换值形式指定目标位姿的以变换值变量。

平移量

以实数形式指定要平移的量。



### 9.3 数学函数

ABS	返回数学表达式的绝对值
SQRT	返回数学表达式的平方根
PI	返回常数 $\pi$
SIN	返回正弦值
COS	返回余弦值
ATAN2	返回反正切值
RANDOM	返回一个随机数

---

**ABS (实数)**

---

---

**SQRT (实数)**

---

---

**PI**

---

---

**SIN (实数)**

---

---

**COS (实数)**

---

---

**ATAN2 (实数 1, 实数 2)**

---

---

**RANDOM**

---

关键字	功 能	实 例
ABS	返回数学表达式的绝对值	ABS(数值)
SQRT	返回数学表达式的平方根	SQRT(数值)
PI	返回常数 $\pi$ (3.1415……)	PI
SIN	返回给定角度的正弦值	SIN(数值)
COS	返回给定角度的余弦值	COS(数值)
ATAN2	返回正切值等于 $v1/v2$ 的角度值(单位为度)	ATAN2(v1, v2)
RANDOM	返回一个 0.0 至 1.0 之间的随机数。	RANDOM

#### 示例

$x = \text{ABS}(y)$	将 $ y $ 值代入 $x$ 。
$x = \text{SQRT}(y)$	将 $y$ 的平方根值代入 $x$ 。
$en = 2 * \text{PI} * r$	将 $2\pi r$ 值代入 $en$ 。
$Z = (\text{SIN}(x) ^ 2) + (\text{COS}(y) ^ 2)$	将 $(\sin(x))^2 + (\cos(y))^2$ 值代入 $z$ 。
$\text{slope} = \text{ATAN2}(\text{rise}, \text{run})$	将 $\tan^{-1}(\text{rise}/\text{run})$ 值代入 $\text{slope}$ 。
$r = \text{RANDOM} * 10$	将 0 至 10 之间的一个随机数代入 $r$ 。

#### 9.4 字符串函数

<b>\$CHR</b>	返回指定数值的 ASCII 码。
<b>\$SPACE</b>	返回指定数量的空格。
<b>\$LEFT</b>	返回字符串中最左边的字符。
<b>\$RIGHT</b>	返回字符串中最右边的字符。
<b>\$MID</b>	返回指定数量的字符。
<b>\$DECODE</b>	提取指定的分隔符前的字符。
<b>\$ENCODE</b>	返回由打印数据产生的字符串。
<b>\$ERRORS</b>	返回指定错误代码的出错信息。
<b>\$ERROR</b>	返回由一个负数指定的错误代码的出错信息。
<b>DATE</b>	返回系统的日期。
<b>\$TIME</b>	以字符串的形式返回系统时间。

---

### **\$CHR (实数)**

---

#### **功能**

返回与指定数值相对应的 ASCII 码。

#### **参数**

实数 (或数学表达式)

指定要转变为 ASCII 码的数值，容许范围为 0—255。

#### **示例**

<b>\$CHR (65)</b>	返回与 ASCII 值 65 对应的字符 “A”。
<b>\$CHR (^ H61)</b>	返回与 ASCII 值 97 (16×6+1) 对应的字符 “a”。

---

### **\$SPACE (空格的数量)**

---

#### **功能**

返回指定数量的空格。

#### **参数**

空格的数量

指定空格的数量。指定 0 或一个正数。

#### **示例**

Type "a" + \$SPACE (1) + "dog"    显示 “a dog” (在 “a” 和 “dog” 之间有 1 个空格)

---

### **\$LEFT (字符串, 字符的数量)**

---

#### **功能**

返回自指定字符串最左端字符起的指定数量的字符。

#### **参数**

字符串

字符串、字符变量、或字符串表达式。

#### **字符的数量**

指定要从输入的字符串最左端(第一个字符)起返回的字符个数。如果为 0 或负数，则返回空格。如果指定的数值大于该字符串的字符数，则返回整个字符串。

#### **示例**

`$LEFT ("abcdefgh", 3)`          返回字符串 "abc" 。

`$LEFT ("*1*2*3*4*5", 15)`      返回 "\*1\*2\*3\*4\*5" (整个字符串)。

---

### **\$RIGHT (字符串, 字符的数量)**

---

#### **功能**

返回自指定字符串最右端字符起的指定数量的字符。

#### **参数**

字符串

字符串、字符串变量、或字符串表达式。

#### **字符的数量**

指定要从输入的字符串最右端(最后一个字符)起返回的字符个数。如果为 0 或负数，则返回空格。如果指定的数值大于字符串的字符数，则返回整个字符串。

#### **示例**

`$RIGHT ("abcdefgh", 3)`          返回字符串 "fgh" 。

---

### **\$MID(字符串, 实数值, 字符的数量)**

---

#### 功能

从指定字符串中返回指定数量的字符。

#### 参数

字符串

字符串、字符串变量、字符串表达式。

实数值(或数学表达式)

指定要提取的字符串的起始位置。

字符的数量

设定提取的字符的数量。

#### 说明

如果起始位置未指定，或指定值小于或等于 1，则从字符串的第一个字符开始提取字符。

如果指定的起始位置小于或等于 0，或数值大于字符串的字符数，则返回空格。

如果要提取的字符数量未指定，或者它大于字符串的字符数，则返回从指定起始位置开始到字符串最后一个字符之间的所有字符。

#### 示例

如下指令中，\$MID 函数返回“cd”（从字符串“abcdef”第三个字符开始提取两个字符）。其结果被赋给字符串变量\$substring。

```
$substring=$MID(“abcdef”,3,2)
```

---

## **\$DECODE (字符串变量, 分隔符, 模式)**

---

### 功能

返回由“分隔符”隔开的字符串。

### 参数

字符串变量

设定要从中提取字符的字符串。用该函数提取的字符将从该字符串中移除。

分隔符

指定读作为分隔符的字符(字符串的任一字符都可以被指定为分隔符)。

模式

指定该函数操作方法的实数。

如果模式为负数或 0, 或未指定, 则返回从字符串变量的第一个字符至分隔符之间的字符。返回的字符串从字符串变量中移除。如果模式指定为正数, 则返回字符串中出现的第一个分隔符。返回的分隔符将从字符串变量中移除。若在字符串中连续存在两个以上的分隔符, 所有这些分隔符都被返回, 并从字符串中移除。

### 说明

该函数搜寻指定字符串中的分隔符, 并提取从字符串首个字符至分隔符之间的字符。被提取的字符作为函数结果返回, 同时提取的字符从原字符串中移除。

作为该函数的结果返回的字符串(从原字符串移出的字符串)可以是分隔符之前的字符或是分隔符本身。

### — [ 注 意 ] —

该函数在返回字符的同时, 也改变了原来的字符串。

分隔符不分大小写。



### 示例

在下面的指令中，从字符串“\$input”中移走被逗号或空格隔开的数字。在 DO 结构中的第一个指令提取\$input 中第一组字符，并赋给变量“\$temp”。然后 VAL 函数将上一指令中提取的字符串转变为实数值。再将实数值赋给数组变量“value”。然后，程序继续执行下一个\$DECODE 指令，并搜寻下一个分隔符(\$input 中的前一个分隔符已被移走)。

```
i=0      ;计数器复位
DO
$temp=$DECODE($input, ",", 0)      ;提取“,”之前的字符
value[i]=VAL($temp)                ;将字符转变为实数值
if $input =="" GOTO 100
$temp=$DECODE($input, ",", 1)      ;提前分隔符“,”
i=i+1                                ;计数器增 1
UNTIL $input==""                    ;继续程序执行，直至没有字符
100 TYPE "END"
```

如果\$input 的值如下所示，中间用空格和逗号隔开，则上述程序的执行结果将为如下：

1234, 93465.2, .4358, 3458103,

value[0]	1234.0
value[1]	93465.2
value[2]	0.4358
value[3]	3458103.0

作为程序执行的结果，字符串变量\$input 的值变成“”（或空）。

---

## **\$ENCODE (打印数据, 打印数据, .....)**

---

### 功能

返回在参数中指定的打印数据产生的字符串。字符串的生成方式与使用 TYPE 指令相同。

### 参数

#### 打印数据

从下面选择一个或多个。指定数据大于一个时, 用逗号隔开各个打印数据。

- (1) 字符串
- (2) 实数表达式(该值被计算并显示)
- (3) 格式化信息(控制输出信息的格式)

### 说明

此函数与 TYPE 指令一样, 可以在程序中用打印数据生成字符串。与 TYPE 不同的是, \$ENCODE 函数不显示所产生的字符串, 但取而代之的是, 其结果被用作程序中的值。

以下代码用于指定数学表达式的输出格式。在指定下一个不同的格式以前, 前一格式一直有效。在任一格式下, 若数值太长而不能在给定宽度中显示时, 将显示星号(\*)代替该数值。

#### 格式规范代码

- |               |  |
|---------------|--|
| <b>/D</b>     | 采用默认格式。除了数值后的 0 及数值之间所有空格只留下一个数值之间的空格之外, 与指定/G15.8 格式相同。                               |
| <b>/Em. n</b> | 以科学计数法的形式表示数值(如: -1.234 E +02), “m” 描述在终端显示的所有字符的数量, “n” 为小数的位数。m 应比 n 至少大 6, 但不大于 32。 |
| <b>Fm. n</b>  | 以固定的小数点符号表示数值(如: -1.234), “m” 表示在终端显示的所有字符的位数, “n” 为小数位数。                              |
| <b>/Gm. n</b> | 如果数值可以表示为 m 位数字(包含小数点后的 n 位数字)的 Fm. n 格式, 则表示成该格式。否则, 表示为 Gm. n 格式                     |
| <b>/Hn</b>    | 将数值表示成 n 位的十六进制数。  |
| <b>/In</b>    | 将数值表示成 n 位的十进制数。   |

以下参数用于在字符串之间插入某些字符：

- /Cn        在输入该代码处插入 n 次换行，可以放在打印数据的前面或后面。若该代码位于打印数据之中，则插入 n-1 个空白行。
- /S        不换行。
- /Xn        插入 n 个空格。
- /Jn        将数值表示成 n 位数字的十六进制数。用 0 代替空格。(选项)
- /Kn        将数值表示成 n 位数字的十进制数，用 0 代替空格。(选项)
- /L        该代码除了移除所有的空格外，其余功能与/D 格式相同。(选项)

#### 示例

```
$output=$output + $ENCODE (/F6.2, count)
```

将实型变量“count”的值转变为/F6.2 格式的字符串，并加到字符串“\$output”的后面，然后将合成的字符串再赋值回字符串变量“\$output”。

---

### **\$ERRORS (错误代码)**

---

#### **功能**

返回指定错误代码的出错信息。错误代码作为含有出错信息的字符串被返回。

#### **参数**

错误代码

用以下格式指定错误代码： Pxxxx、Wxxxx、Exxxx、或 Dxxxx 。

---

### **\$ERROR (错误编号)**

---

#### **功能**

返回指定错误代码的出错信息。

#### **参数**

错误编号

用负数形式指定出错编号 (以负号“-”开头)。出错代码将转换成如下所示的负的出错编号：

Dxxxx	:	-4xxxx
Exxxx	:	-3xxxx
Wxxxx	:	-2xxxx
Pxxxx	:	-1xxxx

---

### **\$DATE (日期形式)**

---

#### **功能**

以指定的字符串格式返回系统日期。

#### **参数**

日期格式

用数字 1—3 指定要输出的日期格式。

#### **说明**

日期格式如下：

\$DATE (1)            月/日/年

(如果日期为 2008 年 7 月 10 日，则返回值为 07/10/2008。)

\$DATE (2)            日/月/年

(如果日期为 2008 年 7 月 10 日，返回值为 10/JUL/2008)。

“月”是从一月至十二月顺序排列的月份缩写：JAN/ FEB/ MAR/ APR/ MAY/ JUN/ JUL/ AUG/  
SEP/ OCT/ NOV/ DEC。

\$DATE (3)            年/月/日

(如果日期为 2008 年 7 月 10 日，返回值为 2008/07/10。)

---

### **\$TIME**

---

#### **功能**

按如下字符串格式返回系统时间：时:分:秒

#### **示例**

18: 27: 50

钟点采用从 0 至 23 的 24 小时制。

10.0 PC 程序

本章描述了用于过程控制程序(PC=Process Control)的监控指令和程序命令。在右边的括号中 M 代表监控指令，P 代表程序命令。那些既有 M 又有 P 的，即既可用作指令又可用作命令。

PCSTATUS	显示指定 PC 程序的状态。(M)
PCEXECUTE	执行指定的 PC 程序。(M, P)
PCABORT	立即停止指定 PC 程序的执行。(M, P)
PCKILL	初始化 PC 程序执行堆栈。(M)
PCEND	停止指定 PC 程序的执行。(M, P)
PCCONTINUE	继续 PC 程序的执行。(M)
PCSTEP	执行 PC 程序的单个步骤。(M)
PCSCAN	指定 PC 程序处理时间。(P)

示 例

关键字

参数

↓

↓

PCSTATUS

程序编号

带有 标记的参数可以省略。

关键字和参数之间一定要输入空格。

例子中的位姿变量代表 键。

PCSTATUS

PC 程序编号:

功能

显示 PC 程序的状态。(M)

参数

PC 程序编号

选择要显示的 PC 程序编号。容许范围：1-5。如果不指定，则设为 1。

说明

PC 程序状态以下面的格式显示。

(1)	• • •	PC 程序状态:	程序未运行
		程序运行次数:	
(2)	• • •	已运行完成次数:	11
(3)	• • •	剩余运行次数	
(4)	• • •	程序名	优选数 步骤号 No
(5)	• • •	pc_test0	1 PRINT "step1"

(1) PC 程序状态

PC 程序状态描述了下面状态之一：

程序未运行；程序目前不在运行。

程序运行中；程序目前正在运行。

程序等待中；程序在运行，但是正在等待“WAIT”指令中设置的条件被满足。

(2) 已运行完成次数

显示已执行完成的周期数量。

(3) 剩余运行次数

显示还未执行的周期数量。如果在“PCEXECUTE”指令中执行周期被设为负数(-1)，将显示“无限次”。

(4) 程序名

(5) 步骤

显示当前正在执行的步骤编号以及在此步骤中写入的命令。

PCEXECUTE	PC 程序编号	程序名称	执行周期	步骤编号
<b>功能</b>				
执行 PC 程序。(M, P)				
<b>参数</b>				
<b>PC 程序编号</b>				
选择希望执行的 PC 程序编号。容许范围：1 到 5。如果不指定，则设为 1。可以同时执行多达 5 个程序。PC 程序的编号并不代表优先级的顺序。				
<b>程序名称</b>				
选择以 PC 程序编号执行的程序的名称。如果不指定，则选择上一次用“PCEXECUTE”指令执行过的程序。				
<b>执行周期</b>				
指定 PC 程序的执行次数。如果不指定，则设为 1。如果输入-1，则连续执行程序。				
<b>步骤编号</b>				
选择从哪个步骤开始执行。如果不指定，则从程序的第一步开始执行。				
<b>说明</b>				
此指令除了执行的是 PC 程序而非机器人控制程序之外，它与“EXECUTE”监控指令是相同的。当前执行的 PC 程序以它的名称及后面添加闪烁的“*”来显示。				
PCEXECUTE 即可使用监控指令又可使用命令或机器人控制程序中的命令。				
<b>示例</b>				
PCEXECUTE control, -1		程序“control”被连续执行；即程序连续执行，直到执行了 PCABORT 指令、执行了程序中的 PAUSE 或 HALT 命令或有错误发生。		



---

<b>PCABORT</b>	<b>PC 程序编号:</b>
----------------	-----------------

---

### 功能

停止当前运行中的程序的执行。(M, P)

### 参数

PC 程序编号

选择要停止的 PC 程序编号。容许范围：1 到 5。如果不指定，则设为 1。

### 说明

PCABORT 指令除了停止的是 PC 程序而非机器人控制程序之外，它与 ABORT 指令是相同的。

当前运行中的程序被停止，用“PCCONTINUE”指令可以继续执行。

PCABORT 可以在机器人控制程序中，作为监控指令或命令使用。

---

<b>PCKILL</b>	<b>PC 程序编号:</b>
---------------	-----------------

---

### 功能

初始化 PC 程序的堆栈。(M)

### 参数

PC 程序编号

选择要初始化的 PC 程序编号。容许号码从 1 到 5。如果未指定，则设为 1。

### 说明

此指令初始化 PC 程序的程序堆栈。

当一个程序因为 PAUSE 或 PCABORT 指令或错误而暂停时，程序保存在程序堆栈中。只要程序在堆栈中，就不能将其删除(DELETE 指令)。这种情况下，首先要用 PCKILL 指令，将程序从堆栈中除去。

---

PCEND	PC 程序编号:任务编号
-------	--------------

---

### 功能

当执行到那个程序中的下一个 STOP 命令时，停止当前运行中的 PC 程序的执行。(M, P)

### 参数

#### PC 程序编号

选择要结束的 PC 程序编号。容许范围：1—5。如果未指定，则设为 1。

#### 任务编号

指定 1 或-1。如果未指定，则设为 1。

### 说明

如果任务编号未指定或指定为 1，只要执行到下一个 STOP 或 RETURN 命令(或类似的命令)时，立即停止程序的执行，而不管还有没有剩余周期。剩余的周期可以用 PCCONTINUE 命令来执行。

如果为任务编号指定-1，则取消以前输入的 PCEND 指令。当出现程序循环或程序无限运行而不带 STOP 命令时，PCEND 无效，并且必须用 PCEND-1 命令取消。(要取消循环，必须用 PCABORT)。

PCEND 可以在机器人控制程序中，作为监控指令或命令使用。

---

PCCONTINUE	PC 程序编号	NEXT
------------	---------	------

---

### 功能

继续执行一个暂停的 PC 程序。或者跳过 PC 程序中的 WAIT 命令。(M)

### 参数

#### PC 程序编号

选择要继续执行的 PC 程序编号。容许范围：1—5。如果未指定，则设为 1。

#### NEXT

如果指定了此参数，那么将从暂停步骤的下一步骤继续执行。如果未指定，则从被暂停的步骤继续执行。带有参数 NEXT，此指令可以被用来跳过正在运行的 PC 程序中的 WAIT 命令，并且继续那个 PC 程序的执行。

### 说明

PCCONTINUE 指令除了是用来继续 PC 程序而非机器人控制程序之外，它与 CONTINUE 指令是一样的。

程序从因 PAUSE、PCABORT 指令或错误引起停止的步骤处继续执行。如果指定了参数 NEXT，则从该停止步骤的下一步骤继续执行。

PCSTEP	PC 程序编号:程序名称,	执行周期,	步骤编号
--------	---------------	-------	------

功能

执行 PC 程序的单个步骤。(M)

参数

PC 程序编号

选择包含有需要步骤的 PC 程序编号。容许范围：1－5。如果未指定，则设为 1。

程序名称

选择以该 PC 程序编号执行的程序的名称。如果未指定，则选择当前执行的程序或上一次执行的程序。

执行周期

指定执行该程序步骤多少次。如果未指定，则设为 1。

步骤编号

选择要执行的程序步骤的编号。如果未指定，选择程序的第一步。如果参数均未指定，则执行下一步。

说明


PCSTEP 指令，与 PCCONTINUE 指令一样，仅能在以下条件下不带参数使用。

- 1. PCSTEP 指令在上一步骤处曾经使用过的时候
- 2. 在 PAUSE 命令后
- 3. 程序不是因出错而被暂停时

示例

>PCSTEP sequence, , 23 

执行 PC 程序编号 1、名为“sequence”的步骤 23 一次。

在这之后，输入 PCSTEP, 然后执行下一步(24 步)。

---

## PCSCAN 时间

---

### 功能

设定执行 PC 程序的周期时间。(P)

### 参数

#### 时间

设定程序重复循环需要的时间。时间可以指定为 0 秒或更大，以秒为单位。

### 说明

此指令用来在指定的周期时间中执行 PC 程序。如果执行时间大于指定时间，将忽略这里指定的时间。

### 示例

```
program
  PCSCAN 1
  IF sig(1) THEN
    SIGNAL -1
  ELSE
    SIGNAL 1
  END
```

如果用 PCEXECUTE 指令连续执行上述程序(执行周期: -1)，那么 SIGNAL 1 每秒钟从 ON → OFF 转换一次。

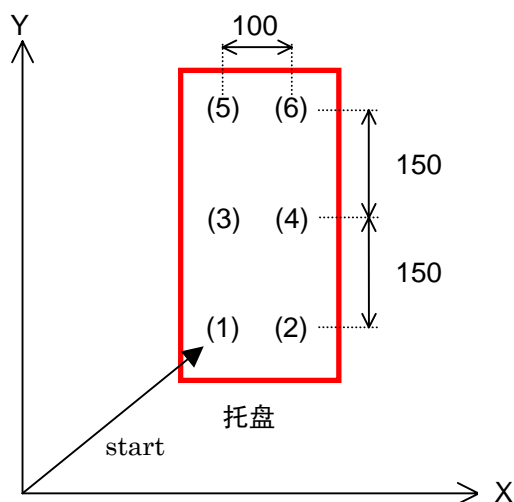
## 11.0 示例程序

本章展示了一些用 AS 语言系统编写的示例程序。

### 11.1 程序的初始化设置

为了简化编程，在机器人执行任何功能之前应进行如下设置：

- 将机器人移动到原点 (HOME) 位姿。
- 为每个任务定义必要的变量 (例如对于码垛任务，确定每个托盘上零件的个数)。
- 初始化计数器、标志等。
- 设置此任务中要用的工件坐标系。
- 设置此任务中要用的基础坐标系。



在上述例子中，零件按照从(1)到(6)的顺序进行码垛。这时，应该使用下面的程序来初始化设置。本例中托盘被放置成与机器人的基础坐标系平行。

```

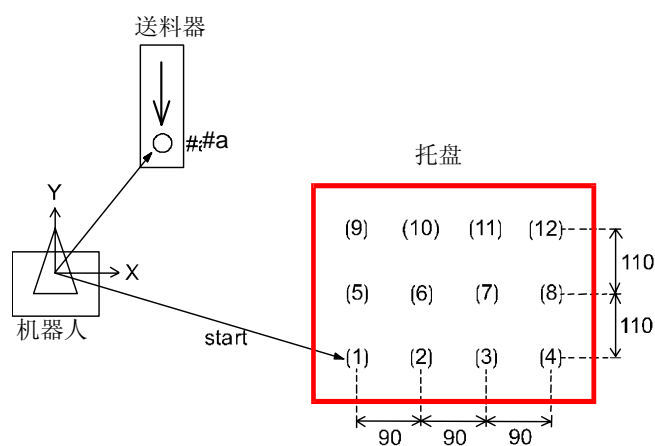
1  BASE  NULL           ;定义机器人基础坐标系(NULL)
2  TOOL  tool1          ;工具转换(工具 1)*
3  row.max=3            ;3 行
4  col.max=2            ;2 列
5  xs=100               ;设置 X 轴上的间距(ΔX=100 mm)
6  ys=150               ;设置 Y 轴上的间距(ΔY=150 mm)
7  POINT put=start      ;将 pose(1) 的值赋予变量 put。
8  OPENI                ;打开工具的手爪
9  HOME                 ;移动到原点位姿**
    
```

注释\* 工具变换值(tool 1)应该事先定义好。

注释\*\* 原点(HOME)应该事先定义好。

## 11.2 码垛

本例中，将零件从送料器上捡起，并放置在 3 行 (110 mm 间距) 4 列 (90 mm 间距) 的托盘上。为了简化说明，托盘和托盘上的零件都平行于机器人基础坐标系的 XY 平面。同样，也省略了用外部 I/O 信号 (SWAIT 命令, SIGNAL 命令等) 对送料器和机器人的同步处理。



- 托盘被放置成平行于机器人基础坐标系的 XY 平面。
- 应该在执行程序前定义好位姿#a (零件送料器)和位姿 “start” (放置第一个零件的地方)。

### 程序示例

```
. PROGRAM palletize
;      初始化设置(3 行, 4 列, ΔX=90, ΔY=110 等)
row.max=3
col.max=4
xs=90
ys=110
SPEED 100 ALWAYS
ACCURACY 100 ALWAYS
POINT put=start
OPENI
;
;      开始进行码垛
FOR row=1 TO row.max
FOR col=1 TO col.max
JAPPRO #a, 100
SPEED 30
ACCURACY 1
LMOVE #a
CLOSEI
LDEPART 200
;
JAPPRO put, 200
SPEED 30
ACCURACY 1
LMOVE put
OPENI
LDEPART 200
;
;      计算下一行中零件的位姿。
POINT put=SHIFT(put BY xs, 0, 0)
END
;
;      计算下一列中零件的位姿。
POINT put=SHIFT(start by 0, ys*row, 0)
END
. END
```

} 从送料器捡起零件。

} 放置零件到托盘上。



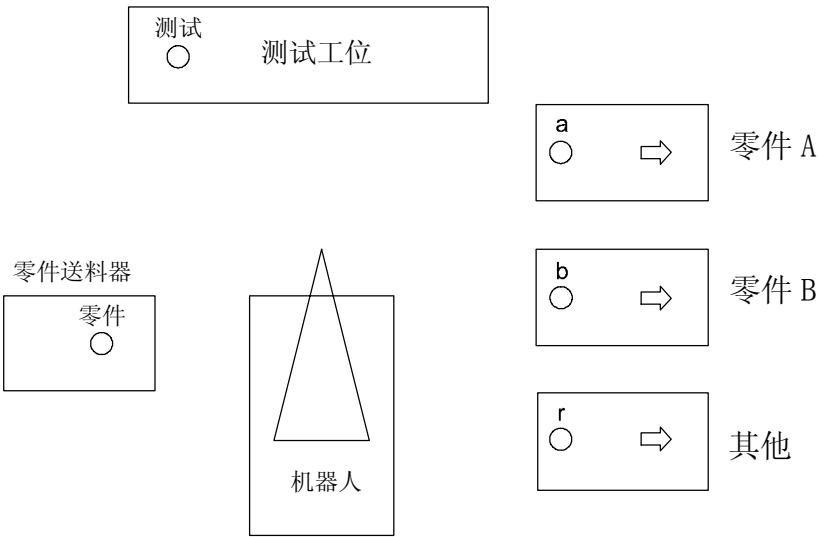
11.3 外部联锁

本例子示范了一个与外部设备同步执行的操作。本程序使用的命令有: SIGNAL, IF, SWAIT, ONI, IGNORE。

- 1. 两种类型的零件, A 和 B, 被以随机的顺序放置在送料器中。(放置完成的输入信号: IN1)
- 2. 机器人从送料器捡起一个零件并将其放置在测试工位上。(放置完成的输出信号: OUT1)
- 3. 在测试工位上, 零件被分类成零件 A、零件 B 或不属于 A 或 B 的。  
测试完成的输入信号: IN2  
零件类别的输入信号: IN3, IN4  
(IN3, IN4)=(1, 0) : 零件 A  
(IN3, IN4)=(0, 1) : 零件 B  
(IN3, IN4)=(0, 0) 或 (1, 1) : 其他
- 4. 机器人按照每个零件的类别放置零件。

如果在机器人从送料器捡起零件并运送到测试工位期间, 测试工位出现了任何故障, 程序立即暂定并转到故障排除子程序。故障出现的外部输入信号是 IN7。故障排除完成后输入信号 IN6, 一旦输入了此信号, 机器人立刻继续执行。

将执行上述操作的程序命名为 MAIN, 而将故障排除子程序命名为 EMERGENCY。



## 程序示例

```
; 定义变量
set.end=1001                ;将放置完成的信号(IN1)命名为 set.end
test.end=1002               ;将测试完成的信号(IN2)命名为 test.end
a.part=1003                 ;将零件 A 的信号(IN3)命名为 a.part
b.part=1004                 ;将零件 B 的信号(IN4)命名为 b.part
retry=1006                  ;将故障已排除的信号(IN6)命名为 retry
fault=1007                  ;将故障信号(IN7)命名为 fault
test.start= 1               ;将启动测试信号(OUT1)命名为 test.start

. PROGRAM main()
OPENI
10 JAPPRO part, 100
ONI fault CALL emergency    ;监视信号 fault，检测到时，立即跳转到子程
                              序 emergency。
SWAIT set.end               ;等待零件被放置在送料器中
LMOVE part                  ;移动到位姿 part (零件送料器)
CLOSEI
LDEPART 100
JAPPRO test, 100            ;运送零件到测试工位(位姿 test)
LMOVE test
BREAK
;
IGNORE fault                ;停止监视信号 IN7(fault)
SIGNAL test.start           ;开启信号 test.start 为 ON
TWAIT 1.0
SWAIT test.end              ;等待，直到测试完成
JDEPART 100
SIGNAL -test.start          ;关断信号 test.start 为 OFF
IF SIG(a.part, -b.part) GOTO 20 ;如果零件是零件 A，则跳转到标签 20
IF SIG(-a.part, b.part) GOTO 30 ;如果零件是零件 B，则跳转到标签 30
POINT n=r                   ;如果零件既不是 A 也不是 B，则将零件运送到 r
GOTO 40
20 POINT n=a                ;定义地方去放置零件 A
GOTO 40
30 POINT n=b                ;定义地方去放置零件 B
40 JAPPRO n, 100             ;运送零件到其放置位姿
```

```
LMOVE    n  
OPENI
```

```
LDEPART 100  
GOTO 10
```

```
. END
```

```
. PROGRAM emergency()
```

```
PRINT    "**ERROR**"
```

;输出出错信息到终端上

```
SWAIT    retry
```

;等待，直到故障解决

```
ONI fault CALL emergency
```

;再次启动对 fault 信号的监控

```
RETURN
```

;返回 main 程序

```
. END
```

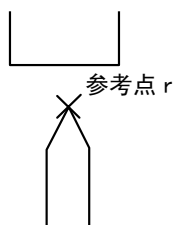
## 11.4 工具变换

本节说明如何来获得工具变换值以及如何应用它们来创建程序。

### 11.4.1 工具变换值-1(当工具尺寸未知时)

当因工具不规则而不知道工具的尺寸时，可以按如下所示来计算工具变换值。基础坐标系的 Z 轴，被设置为与地面垂直。

1. 选择一个带有尖点的物体。固定物体的尖端垂直于地面向上。这个点将成为参考点“r”。

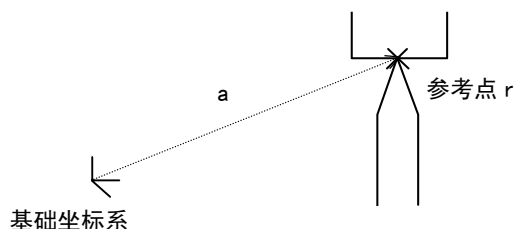


2. 手动移动机器人使安装工具的法兰面垂直向下。然后转到重复模式，输入以下指令：

```
>SPEED 10 
>TOOL NULL  ;设置工件坐标系为空
>DO ALIGN  ;将工件坐标系 Z 轴与基础坐标系 Z 轴对齐
```

3. 使用示教器上的基础模式，手动操作机器人，使法兰的中心垂直于参考点。接着，手动操作机器人，仅沿着基础坐标系的 X、Y、Z 轴移动。输入如下指令，将该位姿的变换值赋给变量“a”：

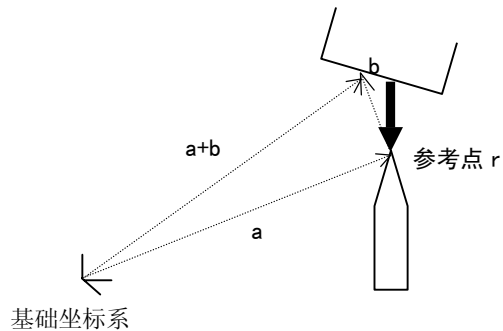
```
>HERE a 
```



4. 安装工具到法兰上，并手动移动工具中心点(TCP)到参考点，使新工具坐标系的 Z 轴垂直于基础坐标系的 X 轴和 Y 轴。

输入如下指令，将该位姿处的变换值示教为复合值“a+b”：

>HERE a+b



5. 从此复合值中，可以发现工具变换值就是“-b”。

输入：

>POINT t=-b

这将-b 的值赋给变量 t。

6. 用 t 指定工具变换值。

>TOOL t

7. 检查，按下述输入：

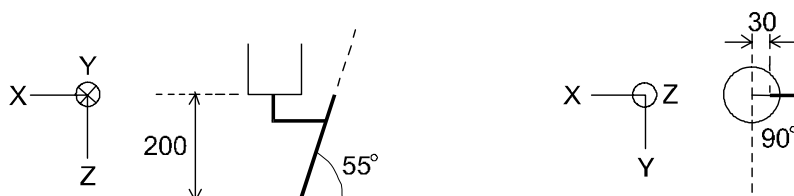
>DO JMOVE r

工具的顶端应该移到参考点 r。

一旦定义完毕，所有的执行都将基于这个工具变换值，除非工具被改变。

### 11.4.2 工具变换值-2(当工具尺寸已知时)

当工具尺寸已知时，工具变换值可以由下述过程获得。由这个过程获得的值通常比前一个过程(见上节 11.4.1)获得的值更精确。



上图中所示的 XYZ 轴为空工件坐标系。下面的过程把工具坐标系的原点设置在焊炬的顶端上，并且将 Z 轴设置成与焊枪方向一致。

(1) 用 POINT 指令定义工具变换值变量 “torch”：

```
>POINT torch 
X      Y  Z      0      A      T
0      0  0      0      0      0
```

变化吗？(放弃请按 RETURN 键)

```
>-30, 0, 200, 0, 35, 0 
X      Y      Z      0      A      T
-30  0      200      0      35      0
变化 
>
```

(2) 用变量 “torch” 设置工具变换值。

```
>TOOL torch 
```

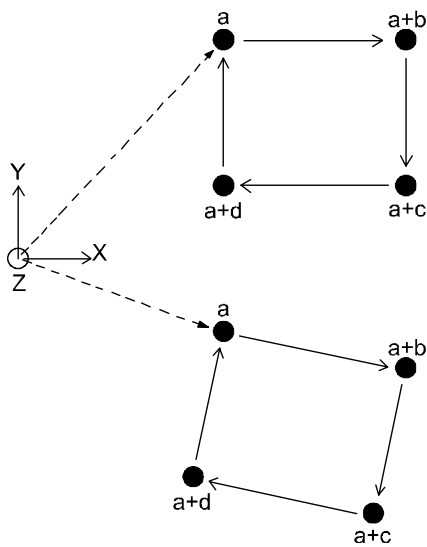
11.5 相对位姿

11.5.1 相对位姿的用法

位姿可以被定义成相对于一个参考点。当以这种方式定义时，即使该参考点被重新定义，但那个位姿与参考点之间的关系仍保持不变。

举例来说，如果示教一张桌子的 4 个角，机器人和桌子的位置关系取决于它们被放置在哪里，只要桌子的形状保持相同，4 个角的关系就不会改变。因此，为了指定机器人和桌子的绝对位置关系，将其中的一个角示教为参考点，而其它 3 个角，用相对于第一个角的方式进行示教，那么，在重新安置桌子的时候，只需要重新定义参考点即可。

示例



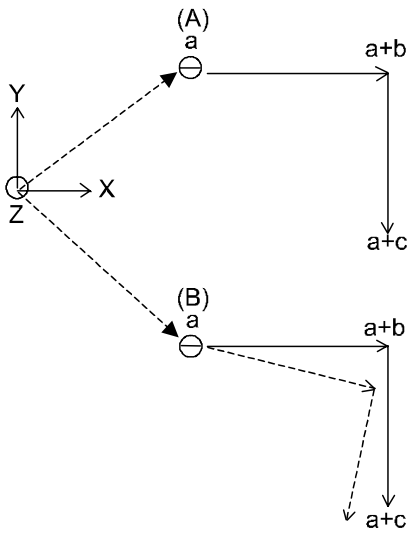
在右边的图(A)，示教了参考点 a 和其他角的复合变换值。在图(B)中，参考点 a 被重新定义。如果机器人工具的定向不重新设定(即与图(A)中保持相同的定向)，机器人将沿实线所示的轨迹移动。如果希望机器人沿虚线移动，就必须重新定义定向和位置。

示教

>HERE a  
>HERE a+b  
>HERE a+c  
>HERE a+d

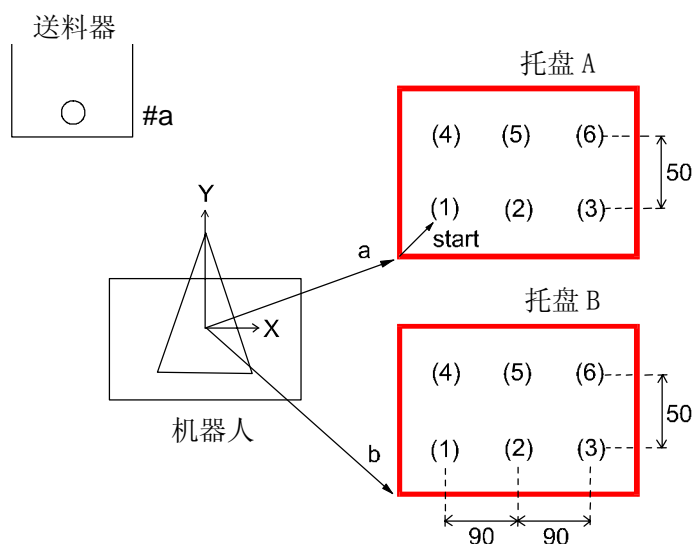
程序

JMOVE a  
LMOVE a+b  
LMOVE a+c  
LMOVE a+d  
LMOVE a



### 11.5.2 使用相对位姿编程的实例

本例中，零件像以前示例中一样被码垛在托盘上，但是这里用了两个托盘。两个托盘被放在不同的位置，但是每个托盘上参考点和零件放置位置之间的关系是相同的。本操作将零件从送料器放置到托盘 A 上。在放置了六个零件以后，机器人继续在托盘 B 上进行同样的操作。（这里省略了与送料器同步的程序）。



需要示教的位姿

- #a : 机器人从送料器捡起零件的位姿
- a : 托盘 A 上的参考位姿
- b : 托盘 B 上的参考位姿
- start : 托盘上第一个零件相对于参考点的位姿



### 程序示例

```
.PROGRAM relative.test
;初始化设置(2 行, 3 列, ΔX=90, ΔY=50, 等)
row.max=2
col.max=3
xs=90
ys=50
OPENI

flg=0 ;flg=0: Pallet A(托盘 A), flg=1: Pallet B(托盘 B)
POINT pallet=a ;定义 Pallet(托盘)A 的参考位姿
; 开始码垛
10 POINT put=start
FOR row=1 TO row.max
FOR col=1 TO col.max
JAPPRO #a, 100
LMOVE #a
CLOSEI
LDEPART 100
;
POINT put_pt=pallet+put
JAPPRO put_pt, 200
LMOVE put_pt
OPENI
LDEPART 200
;
POINT put=SHIFT(put BY xs, 0, 0) ;找出下一列零件的位置
END
;
POINT put=SHIFT(start BY 0, ys*row, 0) ;找出下一行零件的位置
END
;
IF flg<>0 GOTO 30 ;当托盘 B 码放完毕时(fl=1)跳转到结束步骤
flg=1
POINT pallet=b ;定义 Pallet(托盘)B 的参考位姿
GOTO 10
30 TYPE "*** end ***"
STOP
.END
```

## 11.6 用 FRAME 函数的相对位姿

在 11.5.1 节的示例中，在重新定义参考位姿时，必须修正工具的定向。但是，如果使用 FRAME 函数，就不需要那样做。示教 4 个点 (b, c, d, e) 来定义框架 (FRAME) 变换值 a。点 b 和 c 确定了 X 轴的方向，第 3 点 d 确定 XY 平面，点 e 为原点。在示教完这些点后，输入以下指令：

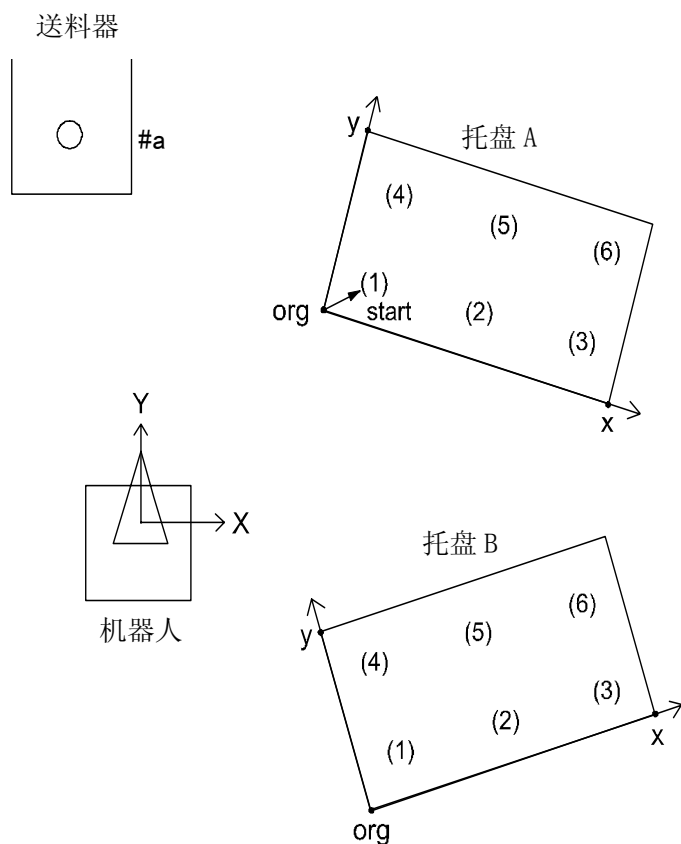
```
POINT a=FRAME (b, c, d, e)
```

然后，相对坐标系被定义为变量“a”。XYZ 值显示了相对坐标系原点的位置，而 OAT 值则显示了相对坐标系的定向。

此后，相对坐标系中的所有位姿都可以被表达为位姿 a + □。如果托盘位置改变，用上述方法再次示教点 b、c、d、e 以重新定义 a。

用 FRAME 函数定义的相对坐标系也叫做框架 (FRAME) 坐标系。

在下面的范例程序中，将使用框架坐标系来实现与 11.5.2 节中相同的操作。



第一步是将零件放在托盘 A 上。示教托盘的 3 个角，一个作为原点(origin)，其余两点分别作为 X 轴和 Y 轴上的点(见上图)。执行以下程序将零件码放在托盘 A 上(请注意：在定义完点以后，程序的其他部分与以前的示例程序是一样的)。为了在托盘 B 上码放零件，重新示教 3 个角，并执行相同的程序。框架坐标系将被重新定义，零件将像在托盘 A 上一样被码放在托盘 B 上。

### 程序示例

```
.PROGRAM frame.test
;初始化设置 (2 行, 3 列, ΔX=90, ΔY=50, 等)
row.max=2
col.max=3
xs=90
yx=50
OPENI
;
POINT pallet=FRAME(org, x, y, org)      ;定义托盘(Pallet)的框架坐标系
;                                       (3 个点: 原点 org, X、Y 轴)
POINT put=start                        ;开始码垛
FOR row=1 TO row.max
FOR col=1 TO col.max
JAPPRO #a, 100
LMOVE #a
CLOSEI
LDEPART 100
;
POINT put_pt=pallet+put
JAPPRO put_pt, 200
LMOVE put_pt
OPENI
LDEPART 200
;
POINT put=SHIFT(put BY xs, 0, 0)      ;找出下一列上零件的位置
END
;
POINT put=SHIFT(startBY 0, ys*row, 0) ;找出下一行上零件的位置
END
STOP
.END
```

## 11.7 设定机器人形态

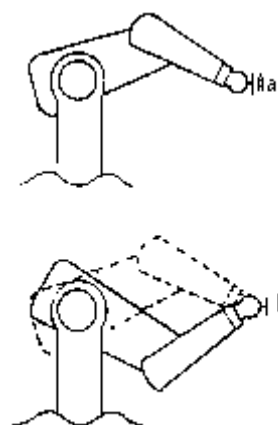
大多数机器人有 6 个关节，当一个位姿被用关节位移值示教时，给定了六个关节的各关节的位移量，因此机器人的位姿是被唯一定义的。另一方面，当一个位姿用变换值来定义时，就会有几种情况，根据机器人设备的形态不同，会有多于一套的角度值能给出相同的由变换值指定的位姿。在 AS 中，机器人基本上保持以前动作的形态，因此无需改变机器人的形态。但是，在以下情况中，应通过一个形态命令来指定机器人的形态：

1. 当机器人从一个不清楚形态的点向由变换值示教的点移动时，
2. 当第 5 关节(弯曲关节)以 SBS(SBS: swivel、bend、swivel, 旋转、弯曲、旋转)手腕形态通过零度位置( $0^\circ$ )时

例如，右图中所示，如果位姿#a 具有 ABOVE 形态(肘关节处于上方)，那末即使位姿 b 最初定义为 BELOW 形态(肘关节处于下方)，命令 JMOVE b 的结果仍将是 ABOVE(如图中虚线所示)。

```
JMOVE #a  
JMOVE b
```

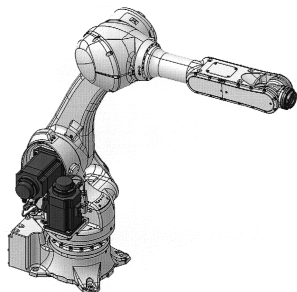
同样地，如果#a 被定义为 UWRIST (JT5>0, 即 JT5 的角度为正值)，位姿 b 的形态也将是 UWRIST，而不管该位姿在示教时的形态。



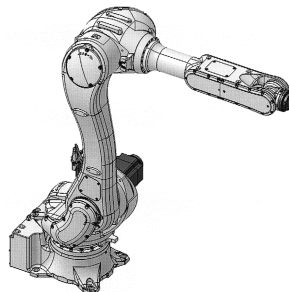
要解决这些问题，有必要改变机器人在运动中的形态。无论何时，当一条关节运动命令(关节插值运动命令：JMOVE, JAPPRO, JDEPART, DRIVE 等)以一个由变换值定义的点作为结束的时候，可以通过执行一条形态命令来实现这一改变。这里列出 6 条形态命令，程序示例在下几页的注意框中给出。

### LEFTY, RIGHTY

设定机器人前 3 个关节(JT1, JT2, JT3)的姿态。LEFTY 将机器人形态设定成类似人的左臂, 而 RIGHTY 则设定成类似右臂。



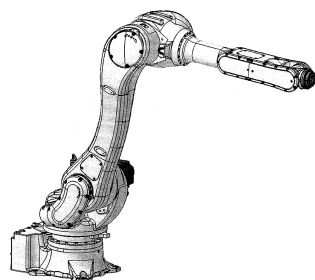
LEFTY



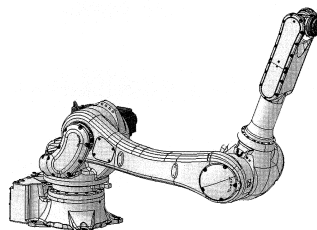
RIGHTY

### ABOVE, BELOW

设置机器人形态, 使得第 3 个关节(JT3)处于上方(ABOVE)或位于下方(BELOW)。



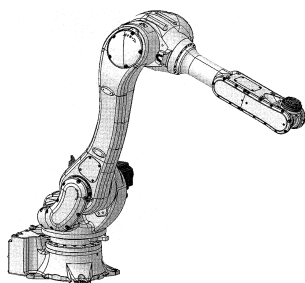
ABOVE



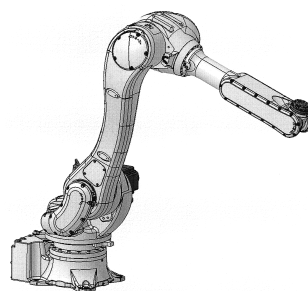
BELOW

### UWRIST, DWRIST

设定机器人形态, 为使得机器人保持相同的工具定向, 其第 5 关节(JT5)的值为正(UWRIST)或为负(DWRIST)。



UWRIST



DWRIST

— [ 注 意 ] —

1. 通常来说, 形态命令对关节位移值(位姿名称以#开头的)没有影响, 机器人会以示教时的姿态移动到示教的位置。然而, 当机器人以直线插补动作, 在起始位姿的形态与目标位姿的形态不同的两个位姿之间移动时, 会导致错误。
2. 机器人不会对形态命令立即做出反应。而是在执行下一个关节插补运动(JMOVE, JAPPRO, JDEPART, DRIVE 等)时, 才改变其形态。
3. 在常规程序中, 除非是故意要改变形态, 否则不必对它进行改动。形态命令用在下述情况下:
  - (1) 当程序不是以关节位移值定义的位姿为目标的运动命令开始时, 形态命令应写在程序的开头以确定机器人的姿态。
  - (2) 当 JAPPRO 命令按如下所示使用时, 应使用形态命令:

```
JMOVE    a
JAPPRO   #b, 100
JMOVE    #b
```

在执行运动命令“JAPPRO #b, 100”之后的姿态, 在有些情况下, 可能不同于#b 的姿态。如果手腕的姿态(JT5 角度的正负)不同, 那么执行下一步“JMOVE #b”时, 可能会引起 JT4 和 JT6 大量旋转。一个避免这种问题的方法是将位姿#b 上方 100mm 的位姿示教为#bb, 并使用如下的 JMOVE 命令。

```
JMOVE    a
JMOVE    #bb
JMOVE    #b
:
```

另一种避免 JT4 和 JT6 偏离的方法, 是用形态命令指定手腕姿态。本例中, #b 处的姿态被假定为 UWRIST(JT5 的值为正)。

```
MOVE     a
UWRIST
JAPPRO   #b, 100
JMOVE    #b
:
```

手腕的姿态在执行“JAPPRO #b, 100”后变为 UWRIST, 从而避免了关节 4 和 6 在执行“JMOVE #b”时的不必要的旋转。



## 附录 1 信号编号的限制

编号	M, P, F*		输出信号	输入信号	内部信号
1	BITS	MP	1 到最大信号**	-----	2001 到最大信号**
2	BITS	F	1 到最大信号**	1001 到最大信号**	2001 到最大信号**
3	DEFSIG	M	1 到最大信号**	1001 到最大信号**	-----
4	DLYSIG	MP	1 到最大信号**	-----	2001 到最大信号**
5	ON, ONI	P	-----	1001 到 1256***	2001 到 2256
6	PULSE	MP	1 到最大信号**	-----	2001 到最大信号**
7	RUNMASK	P	1 到最大信号**	-----	2001 到最大信号**
8	SIGNAL	MP	1 到最大信号**	-----	2001 到最大信号**
9	SIG	F	1 到最大信号**	1001 到最大信号**	2001 到最大信号**
10	SWAIT	P	1 到最大信号**	1001 到最大信号**	2001 到最大信号**
11	XMOVE	P	1 到 256***	1001 到 1256***	2001 到 2256
12					
13					

**注释\*** M= 监控指令, P= 程序命令, F= 函数

**注释\*\*** 最大信号: 已安装的外部信号的数量  
32 (标准), 最大 960 (选项)

**注释\*\*\*** 尽管标明为从 1 到 256 或从 1001 到 1256, 如果安装了标准 I/O 模块, 最大编号可以指定为 32 (1032)。



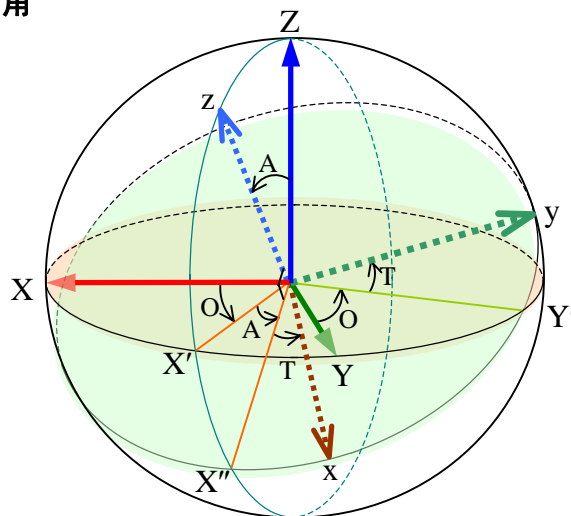
## 附录 2 ASCII 码

ASCII 字符	八进制	十进制	十六进制	说明
NULL	000	00	00	Null
SOH	001	01	01	Start of heading
STX	002	02	02	Start of text
ETX	003	03	03	End of text
EOT	004	04	04	End of transmission
ENQ	005	05	05	Enquiry
ACK	006	06	06	Acknowledge
BEL	007	07	07	Bell
BS	010	08	08	Backspace
HT	011	09	09	Horizontal tabulation
LF	012	10	0A	Line feed
VT	013	11	0B	Vertical tabulation
FF	014	12	0C	Form feed
CR	015	13	0D	Carriage return
SO	016	14	0E	Shift out
SI	017	15	0F	Shift in
DLE	020	16	10	Data link escape
DC1	021	17	11	Device control 1
DC2	022	18	12	Device control 2
DC3	023	19	13	Device control 3
DC4	024	20	14	Device control 4
NAK	025	21	15	Negative acknowledge
SYN	026	22	16	Synchronous idle
ETB	027	23	17	End of transmission block
CAN	030	24	18	Cancel
EM	031	25	19	End of medium
SUB	032	26	1A	Substitute
ESC	033	27	1B	Escape
FS	034	28	1C	File separator
GS	035	29	1D	Group separator
RS	036	30	1E	Record separator
US	037	31	1F	Unit separator
SP	040	32	20	Space

ASCII 字符	八进制	十进制	十六进制	ASCII 字符	八进制	十进制	十六进制
!	040	32	20	0	060	48	30
”	041	33	21	1	061	49	31
#	042	34	22	2	062	50	32
\$	043	35	23	3	063	51	33
%	044	36	24	4	064	52	34
&	045	37	25	5	065	53	35
'	046	38	26	6	066	54	36
(	047	39	27	7	067	55	37
)	050	40	28	8	070	56	38
*	051	41	29	9	071	57	39
+	052	42	2A	:	072	58	3A
,	053	43	2B	;	073	59	3B
-	054	44	2C	<	074	60	3C
	055	45	2D	=	075	61	3D

ASCII 字符	八进制	十进制	十六进制	ASCII 字符	八进制	十进制	十六进制
@	100	64	40		140	96	60
A	101	65	41	a	141	97	61
B	102	66	42	b	142	98	62
C	103	67	43	c	143	99	63
D	104	68	44	d	144	100	64
E	105	69	45	e	145	101	65
F	106	70	46	f	146	102	66
G	107	71	47	g	147	103	67
H	110	72	48	h	150	104	68
I	111	73	49	i	151	105	69
J	112	74	4A	j	152	106	6A
K	113	75	4B	k	153	107	6B
L	114	76	4C	l	154	108	6C
M	115	77	4D	m	155	109	6D
N	116	78	4E	n	156	110	6E
O	117	79	4F	o	157	111	6F
P	120	80	50	p	160	112	70
Q	121	81	51	q	161	113	71
R	122	82	52	r	162	114	72
S	123	83	53	s	163	115	73
T	124	84	54	t	164	116	74
U	125	85	55	u	165	117	75
V	126	86	56	v	166	118	76
W	127	87	57	w	167	119	77
X	130	88	58	x	170	120	78
Y	131	89	59	y	171	121	79
Z	132	90	5A	z	172	122	7A
[	133	91	5B		173	123	7B
\	134	92	5C		174	124	7C
]	135	93	5D		175	125	7D
	136	94	5E		176	126	7E
-	137	95	5F	DEL	177	127	7F

### 附录 3 欧拉 O, A, T 角



坐标系  $\Sigma(x, y, z)$  相对于基础坐标系  $\Sigma(X, Y, Z)$  的姿态，通常用欧拉 OAT 角来表达。如上图所示，三个角可以定义为如下。在上图中，两个坐标系  $\Sigma(x, y, z)$  和  $\Sigma(X, Y, Z)$  具有相同的原点。

O:  $Zz$  平面与  $XZ$  平面之间的夹角

A:  $z$  轴与  $Z$  轴之间的夹角

T:  $x$  轴与  $X''$  轴之间的夹角

$X''$  轴在  $Zz$  平面上，此轴和  $z$  轴之间的夹角是  $90^\circ$ 。

这三个角也可用被说成，代表了要使基础坐标系  $\Sigma(X, Y, Z)$  与坐标系  $\Sigma(x, y, z)$  相重合所需旋转的角度。旋转的顺序不能改变，否则结果会不同。

1. 坐标系  $\Sigma(X, Y, Z)$  绕  $Z$  轴的旋转 O。(将坐标系  $\Sigma(X, Y, Z)$  移动到坐标系  $\Sigma(X', Y', Z)$ 。)
2. 坐标系  $\Sigma(X', Y', Z)$  绕  $Y'$  轴的旋转 A。(将  $\Sigma(X', Y', Z)$  移动到  $\Sigma(X'', Y', Z)$ 。)
3. 坐标系  $\Sigma(X'', Y', z)$  绕  $z$  轴的旋转 T。(将  $\Sigma(X'', Y', z)$  移动到  $\Sigma(x, y, z)$ 。)

此外，也可以按照极坐标值来考虑：当 P 点，它在  $z$  轴上，与原点相距  $d$ ，被记为  $(d, A, 0)$ ；那么这些坐标值中的 O 和 A 与上述的 O 和 A 相等。Z 轴的方向将由这两个值描述。



## 附录 4 错误信息表

代码	错误信息
P0100	非法的输入数据。
P0101	变量太多。
P0102	输入数据过大。
P0103	非法的 PC 编号。
P0104	非法的机器人编号。
P0105	非法程序。
P0106	非法的优先级。
P0107	无效的坐标值。
P0108	语法错误。
P0109	无效语句。
P0110	请指明命令的全拼。
P0111	当前模式下不能使用该条命令/指令。
P0112	不能用 D0 指令执行。
P0113	不是程序指令。
P0114	非法表达式。
P0115	非法函数。
P0116	函数变量非法。
P0117	无较变量(或程序)名。
P0118	非法的变量型类。
P0119	数组下标错误。
P0120	括号不成对。
P0121	需要的是二进制算子。
P0122	非法常数。
P0123	非法的原定符。
P0124	无较标签。
P0125	缺字符。
P0126	非法的开关名称。
P0127	指定的开关名称需要全拼。
P0128	非法格式的分类符。
P0129	标签语句重复。
P0130	不能定义为数组。
P0131	数组的维数大于 3。
P0132	数组变量已存在。
P0133	数组变量不存在。

P0134	需要使用数组变量。
P0135	需要局部变量。
P0136	意外的数组下标。
P0137	调用子程序时自变量不匹配。
P0138	调用子程序时自变量的类型不匹配。
P0139	非法的控制结构。
P0140	步: XX 错误的 END 语句。
P0141	步: XX 多余的 END 语句。
P0142	步: XX 不能以 END 停止 D0。
P0143	步: XX 在 CASE 后没有 VALUE 语句。
P0144	步: XX 前面缺少 IF。
P0145	步: XX 前面缺少 CASE。
P0146	步: XX 前面缺少 D0。
P0147	步: XX 未发现 XX 的 END 语句。
P0148	步: XX 控制结构过多。
P0149	变量(或程序)名已存在。
P0150	不同类型的变量已存在。
P0151	复杂的表达式造成内部缓冲区溢出。
P0152	未定义的变量(或程序)。
P0153	非法的时钟值。
P0154	缺少'='。
P0155	缺少')'。
P0156	缺少']'。
P0157	缺少"TO"。
P0158	缺少"BY"。
P0159	缺少': '。
P0160	请指定"ON"或"OFF"。
P0161	必须指定机器人号。
P0162	该指令中不能修改位置数据。
P0163	程序名、变量名、文件名等未指定。
P0164	非法的机器人网络 ID。
P0165	步: XX 在 SCASE 后没有 SVALUE 语句。
P0166	步: XX 前面缺少 SCASE。
P1000	电机电源 OFF, 不能执行程序。
P1001	示教模式中不能执行程序。
P1002	示教模锁 ON, 不能执行程序。
P1003	外部保持信号输入中, 不能执行程序。

P1004	程序正在复位中，不能执行程序。
P1005	因为外部启动可用，程序不能执行。
P1006	因为外部启动禁用，程序不能执行。
P1007	在 RPS_END 步，启动信号未输入。
P1008	保持开关有故障，程序不能执行。
P1009	程序已在运行中。
P1010	机器人控制程序已在运行中。
P1011	程序不能继续，请用 EXECUTE。
P1012	机器人正在运动中。
P1013	出错中，不能执行，请先复位错误。
P1014	程序已在使用中，不能执行。
P1015	不能删除，正被其他命令使用中。
P1016	不能删除，正在被程序使用。
P1017	不能删除，编辑器中的程序。
P1018	用 KILL 或 PCKILL 取消程序。
P1019	PC 程序运行中。
P1020	示教器操作中，不能运行。
P1021	不能用 D0 命令执行。
P1022	不能用 MC 指令执行。
P1023	不能在机器人程序中执行。
P1024	语句不能被执行。
P1025	函数未设置，不能被执行。
P1026	不能 KILL 运动中的程序。
P1027	示教锁 ON，不能编辑程序。
P1028	不能粘贴。
P1029	程序名未指定。
P1030	程序被另一进程互锁。
P1031	无空闲内存。
P1032	无程序步。
P1033	程序名已存在。
P1034	该程序不可编辑。
P1035	记录被禁止，设定[Record Accept]并重新运行。
P1036	程序修改被禁止，设定[Accept]并重新运行。
P1037	程序名不能是“calib_load_”。
P1038	程序不存在。
P1039	示教器未连接。
P1040	在接口板中不能执行该指令。



P1041	自动监控指令失败。
P1042	NUM 程序运行中。
P1043	不能在再现模式下执行。
P1044	因为电机电源 ON，所以不能执行。
P1045	设置示教模式及示教锁 ON。
P1046	接通触发器开关。
P1047	未连接机器人，不能选择程序/步。
P1048	刹车检查中，不能操作。
P1049	程序被锁住。
P1050	保护程序已存在。
P1051	程序运行中不能解锁保护。
P1052	因存储器已满，不能复制程序。
P1053	因存储器已满，程序复制已被挂起。
P1054	请关断触发开关。
P1055	请示教在夹紧 ON 的步骤上轴锁定指示。
P1056	请示教在夹紧 ON 的步骤上解除轴锁定的指示。
P1059	不能在程序运行中记录。
P1060	在 PC 程序中不能执行。
P2000	关断电机电源。
P2001	把保持/运转关打到保持处。
P2002	无外部轴。
P2003	定位器型号非法。
P2004	不能修改，用户数据已存在。
P2005	图形区域错误。
P2006	选项被关闭。
P2007	正被其他设备执行中，所以不能被执行。
P2008	设备未准备好。
P2009	文件名非法。
P2010	磁盘未准备好。
P2011	无效的磁盘格式。
P2012	磁盘写保护。
P2013	磁盘已满。
P2014	文件过多。
P2015	不能在只读文件上写入。
P2016	不能打开文件。
P2017	不能关闭文件。
P2018	存储数据日志中。

P2019	ADC 函数已在使用中。
P2020	非法的设备编号。
P2021	在该终端上不能执行。
P2022	不能使用 DOUBLE 0X。
P2023	协调模式中。
P2024	无效的 X 坐标值。
P2025	无效的 Y 坐标值。
P2026	无效的 Z 坐标值。
P2027	不能使用已在接口板中被使用的信号。
P2028	手臂 ID 板忙。
P2029	轴设置数据不正确。
P2030	未知的辅助功能号。
P2031	被删除的步是 Jump、Call 指令的目标步。
P2032	WHERE 参数输入值错误。
P2033	日志记录运行中。
P2034	未定义的内存。
P2035	无数据。
P2036	内存校验错误。
P2037	实时路径调节已在运行中。
P2038	矩阵计算错误。
P2039	不能从 FN 指令开始循环。
P2040	卡未准备好。
P2041	加载的卡错误。
P2042	卡写保护。
P2043	卡电池电量不足。
P2044	卡未格式化。
P2045	该卡不能格式化。
P2046	卡初始化错误。
P2047	文件已打开。
P2048	卡中文件不存在。
P2049	试图打开的文件过多。
P2050	卡访问中意外错误。
P2051	文件 I/O 数据序列号非法。
P2052	[LSEQ] 程序包括不可用的指令。
P2053	[LSEQ] 步数太多。
P2054	[LSEQ] 信号变量类型错误。
P2055	[LSEQ] 程序已在运行中。

P2056	[LSEQ] 信号编号超出指定范围。
P2057	[SerialFlash] 不能打开文件。
P2058	[SerialFlash] 数据读取错误。
P2059	[SerialFlash] 数据写入错误。
P2060	[SerialFlash] 文件或路径不存在。
P2061	软盘中文件不存在。
P2062	[FDD/PC_CARD] 经校验，写入数据失败。
P2063	[FDD/PC_CARD] 校验功能错误响应。
P2064	[FDD] 无可利用空间。
P2065	[Multi Disks] 装载了错误磁盘。
P2066	引导闪存处于禁止写入状态。
P2067	[Serial Flash] 文件目录错误。
P2068	不能执行正在编辑的程序。
P2069	[FDD/PC_CARD] 设备正在使用中。
P2070	不能登录更多的数据。
P2071	C/S 开关设置为禁止。
P2072	[LSEQ] 最大的执行循环数。
P2073	[LSEQ] 其他程序等待执行。
P2074	软盘已损坏。
P2075	轴 XX 的通道号不正确。
P2076	SAVE/LOAD 进行中。
P2077	[Serial Flash] 访问错误。
P2078	[Serial Flash] 上载或下载失败。
P2079	卡已满。
P2080	不能执行，因为通道已被分配给轴号。
P2083	用户履历未创建。
P2084	用户履历登录数已改变。
P2085	用户履历不能登录，无空余存储空间。
P2086	用户履历未登录。
P2087	登录的用户履历数据与指定数据种类不同。
P2088	不能装入不正确的补偿参数。
P2090	没有伺服规格的伺服数据。
P2091	[Serial Flash] 文件或目录已存在。
P2092	[Serial Flash] 此目录不是空的。
P2093	[Serial Flash] 这里没有写入的空间了。
P2094	[Serial Flash] 不能访问只读文件。
P2095	可选 CPU 没有响应。

P2096	码垛动作中断后循环启动不能执行。
P2097	码垛动作过程中不能变更步骤。
P2098	此轴不能无限旋转。
P2099	不能将码垛状态变更为“开”。
P2100	宏的执行错误。
P2101	include 文件嵌套过深。
P2102	文件或文件夹不存在。
P2103	USB 闪存未插入。
P2104	软件下载失败。
P2105	USB 闪存的可使用空间不足。
P2106	小型闪存的可使用空间不足。
P2107	系统正在下载软件。
P2108	在 USB 闪存中没有软件。
P2109	由于同时操作信号在输入中，因此不能执行程序。
P2110	[USB/CF] 文件写入错误。
P2111	请把旋转轴返回到原点位置。
P2112	文件名太长。
P2113	由 KI 命令不能开始循环操作。
P4500	FIELD-BUS) 接口未被激活。
P4501	DEVNET) 节点 XX 不在扫描列表中。
P4502	DEVNET) 已在该模式。
P4503	信号编号重复。
P4504	FIELD-BUS) 超出最大信号数。(最大 XX)
P4505	CC-LINK) 版本不匹配。
P4506	EN/IP-M) 已经处于指定模式。.
P4507	FIELD-BUS) 不能执行老版本的 ANYBUS 卡固件。
P4508	FIELD-BUS) 不能与交互卡通信。
P4509	FIELD-BUS) 交互卡型号错误。
P4510	FIELD-BUS) 卡的初始化没有完成。
P5000	等待焊接完成。
P5001	等待缩回或伸出位置信号。
P5002	点焊程序正在运行。
P5003	外部轴型号和焊枪型号数据不匹配。
P6000	步 XX 的偏移位置超出范围。
P6001	源程序中的步 XX 超出运动范围。
P6002	指定的喷涂数据库不存在。
P6003	因为重现操作被挂起，程序不能执行。

P6004	吹扫处理中，不能执行。
P6005	因为机器人不连接，不能执行。
P6006	在喷涂区域的结束点不能指定圆弧插补。
P6007	喷涂区域内示教点过多。
P6008	喷涂区域内示教点之间的命令过多。
P6009	喷涂区域内示教点缺少。
P6010	pg 编号以外的程序名被选择。
P6011	不能动作。请更改到各轴插补或添加示教点。
P6012	示教锁为 OFF，因此，不能编辑程序。
P6500	不能生成工作路径方向。
P6501	非法工具姿态。
P6502	无焊接数据库。
P6503	不能改变焊接条件。
P6504	步：XX 缺少先导 L. START。
P6505	轴型类未设置成伺服焊枪。
P6506	不能在圆弧动作中使用平移功能。
P7000	因为不在原点位置 1 处，不能复位程序。
P7001	在力测量模式下，仅可用 NOP Interp。
P7002	因为夹具已 ON，不能改变行程。
P7003	伺服参数文件未找到
P7500	请开启马达电源。
P7501	请进入示教模式。
P7502	超出了中断处理的同时登录容许数。
P7503	不能在错误屏蔽中执行程序。
P7504	ONC/ONCI 通道已接收到数据。
P7505	不能在保存过程中执行。
P7506	机器人在动作中，不能接受记录。
P7507	在再现操作中的数据更改量太大。
P8400	CLAMP MODE 信号输入中，不能执行程序。
P8800	控制器编号重复。
P8801	IL 机器人编号重复。
P8802	IL 服务器正在处理中。
P8803	与 IL 服务器连接未被允许。
P8804	IL 服务器 IP 地址未设定。
P8805	不是示教模式。
P8806	请将伺服关断。
P8807	ILL) 通信超时。

P8808	ILL)PC 服务器处理完毕等待超时错误。
P8809	ILL)PC 服务器处理要求完毕等待超时错误。
P8810	ILL)Inter lock less 功能系统错误。
P8811	ILL)不能解除从动控制器的操作禁止锁定。
P8812	ILL)不能与 PC 服务期通信。
P8813	IL 机器人编号未登录。
P9000	限制方向输入错误。
P9001	限制距离输入错误。
P9002	参考点数据重复。
P9003	参考点 1、2、3 都在一条直线上。
P9004	参考点 4 在容许范围外。
P9005	因为示教锁定为 ON，不能运行。
W1000	在此姿态形状下，轴 XX 不能沿直线运动。
W1001	检查中关节轴的速度超出了最大值，请设低速度。
W1002	操作日志信息已被清除。
W1003	标定失败，请在改变姿态后重试。
W1004	轴 XX 超出运动范围，检查运动区域。
W1005	非法重心，已设置为缺省参数。
W1006	错误的负载力矩，已设置为缺省参数。
W1007	应用设置已改变，请关断控制电源，然后重开。
W1008	参数已改变，请关断控制电源，然后重开。
W1009	在最近的紧停时，轴 XX 的位置包络错误。
W1010	RAM 电池电压低。
W1011	PLC 报警。XX
W1012	伺服参数已改变，关断后再开启控制电源。
W1013	编码器电池电压低。伺服=XX
W1014	轴数已改变。请重新初始化。
W1015	可能发生失误。
W1016	轴 XX 的电机转矩超出极限值。轴 XX
W1017	编码器电池电压低。外部轴=XX
W1018	网络参数已改变，关断后再开启控制电源。
W1019	登录值超出了额定负载值。
W1020	发现错误磁盘扇区。
W1021	在当前位置不能得到最佳姿态。
W1022	未执行 ZRPAADSET 命令。
W1023	示教插头位置错误或 P-N 电压低。XX
W1024	动作开始位置与上次停止位置之间的偏差超出了限制设定。

W1025	(SSCNET) 轴 XX 再生过大警告。代码=XX
W1026	(SSCNET) 轴 XX 的电机过载警告。代码=XX
W1027	提升装置锁定中，不能运动。
W1028	负荷重心超出了允许值。可能会造成减速器故障。
W1029	负荷重心超出了允许值。JT5 不能在 0 度外使用。
W1030	第 XX 轴的刹车力矩已下降。
W1031	不能沿直线运动，除非 JTXX 的值是 0 度。
W1032	不能直线运动—因法兰面方向向上。
W1033	不能变更姿势。
W1034	编码器电源电压降低。(轴 XX)
W1035	编码器电池电压过低。确认调零位置。(轴 XX)
W1036	步骤数据不相同。
W1037	此轴不能无限旋转。
W1048	编码器通信异常报警。轴 XX
W1053	(FANXX) 风扇旋转速度降低。(伺服板 XX)
W1054	AVR 的剩余寿命不长。
W1055	视觉周期超时。
W1056	[主 CPU 板]CPU 温度超限。(XX 1/1000℃)
W1057	不能以现姿势做直线运动。
W1058	链接 3 干涉地面。
W1059	链接 5 干涉机器人的基础部分。
W1060	链接 6 干涉机器人的基础部分。
W1061	示教器连接变更了。确认当前的位姿并操作机器人。
W1062	示教器后灯的点灯时间超过极限。
W1063	MC 继电器的 ON/OFF 操作的次数超过极限。(伺服板 XX) (MCXX)
W1064	超过极限。(部件: XX)
W1071	第 XX 轴的当前位置偏离于停止位置。
W2901	SLOGIC 错误信息 #1
W2902	SLOGIC 错误信息 #2
W2903	SLOGIC 错误信息 #3
W2904	SLOGIC 错误信息 #4
W2905	SLOGIC 错误信息 #5
W2906	SLOGIC 错误信息 #6
W2907	SLOGIC 错误信息 #7
W2908	SLOGIC 错误信息 #8
W2909	SLOGIC 错误信息 #9
W2910	SLOGIC 错误信息 #10

W2911	SLOGIC 错误信息 #11
W2912	SLOGIC 错误信息 #12
W2913	SLOGIC 错误信息 #13
W2914	SLOGIC 错误信息 #14
W2915	SLOGIC 错误信息 #15
W2916	SLOGIC 错误信息 #16
W2917	SLOGIC 错误信息 #17
W2918	SLOGIC 错误信息 #18
W2919	SLOGIC 错误信息 #19
W2920	SLOGIC 错误信息 #20
W2921	SLOGIC 错误信息 #21
W2922	SLOGIC 错误信息 #22
W2923	SLOGIC 错误信息 #23
W2924	SLOGIC 错误信息 #24
W2925	SLOGIC 错误信息 #25
W2926	SLOGIC 错误信息 #26
W2927	SLOGIC 错误信息 #27
W2928	SLOGIC 错误信息 #28
W2929	SLOGIC 错误信息 #29
W2930	SLOGIC 错误信息 #30
W2931	SLOGIC 错误信息 #31
W2932	SLOGIC 错误信息 #32
W2933	SLOGIC 错误信息 #33
W2934	SLOGIC 错误信息 #34
W2935	SLOGIC 错误信息 #35
W2936	SLOGIC 错误信息 #36
W2937	SLOGIC 错误信息 #37
W2938	SLOGIC 错误信息 #38
W2939	SLOGIC 错误信息 #39
W2940	SLOGIC 错误信息 #40
W2941	SLOGIC 错误信息 #41
W2942	SLOGIC 错误信息 #42
W2943	SLOGIC 错误信息 #43
W2944	SLOGIC 错误信息 #44
W2945	SLOGIC 错误信息 #45
W2946	SLOGIC 错误信息 #46
W2947	SLOGIC 错误信息 #47



W2948	SLOGIC 错误信息 #48
W2949	SLOGIC 错误信息 #49
W2950	SLOGIC 错误信息 #50
W2951	SLOGIC 错误信息 #51
W2952	SLOGIC 错误信息 #52
W2953	SLOGIC 错误信息 #53
W2954	SLOGIC 错误信息 #54
W2955	SLOGIC 错误信息 #55
W2956	SLOGIC 错误信息 #56
W2957	SLOGIC 错误信息 #57
W2958	SLOGIC 错误信息 #58
W2959	SLOGIC 错误信息 #59
W2960	SLOGIC 错误信息 #60
W2961	SLOGIC 错误信息 #61
W2962	SLOGIC 错误信息 #62
W2963	SLOGIC 错误信息 #63
W2964	SLOGIC 错误信息 #64
W2965	最大允许负荷值为 XX。
W2966	负荷超出了最大允许力矩。
W2967	负荷超出了最大力矩。
W2968	请将组编号设置为 XX。
W3800	通信板存储器异常。(XX)
W3801	刹车已被释放，不能动作。
W3802	经过时间已超过设定值，请实施保养。
W3803	控制器电源的开机时间已超过设定值，请实施保养。
W3804	机器人的连接时间已超过设定值，请实施保养。
W3805	伺服开机时间已超过设定值，请实施保养。
W3806	第 XX 轴的动作距离合计已超过设定值，请实施保养。
W3807	MC 的动作次数已超过设定值，请实施保养。
W3808	伺服开启的次数已超过设定值，请实施保养。
W3809	紧急停止的次数已超过设定值，请实施保养。
W3810	第 XX 轴的电流三次方数值已超过设定值，请实施保养。
W3811	第 XX 轴的电流二次方数平均值已超过设定值，请实施保养。
W3812	NO. XX I/O 板的输入电源(1) 异常。
W3813	NO. XX I/O 板的输入电源(2) 异常。
W3814	NO. XX I/O 板的输出电源异常或保险丝烧断。
W4000	对故障信息写入，PLC 没有响应。

W4001	故障信息写入失败。[EC=XX]
W4002	对故障信息写入，PLC 的响应错误。
W4500	FIELD-BUS) 从端口离线。
W4501	FIELD-BUS) 主端口离线。
W4502	CC-LINK) 主控板的数据连接异常。XX
W5000	在力测量模式下，释放等待状态。
W5001	PLC 通信失误。
W5002	焊接控制器 XX 没有连接。
W5003	焊接控制器 XX 没有响应。
W5004	焊接控制器 XX 响应错误。
W5005	(点焊) RWC XX 未响应。
W5006	(点焊) RWC 响应错误。XX
W5007	(点焊) 焊接失误。XX
W5008	(点焊) 电缆断开失误。XX
W5009	(点焊) 内部泄漏。XX
W5010	(点焊) 主电流交换报警。XX
W5011	(点焊) RWC XX 未连接。
W5012	不能达到设定力值。
W5013	端部磨损超限。(移动侧)
W5014	端部磨损超限。(固定侧)
W5015	(点焊) 焊接电流下降。
W5016	出现焊接异常。(代码 XX)
W6000	请为减速齿轮和电机承加注润滑油。
W6001	更换机器人主电缆。
W6002	更换控制器内的冷却风扇。
W6003	更换控制器内的 DC(直流) 电源。
W6004	更换伺服电源单元。
W6005	更换机器人臂的电源放大器。
W6006	更换机器人手腕关节的电源放大器。
W6007	更换走行轴电源放大器。
W6008	内压联锁已被短接。
W6009	没有选择内压防爆机种。
W6010	因相互等待无效，不能执行枪间的相对距离检测(ID:XX)。
W6011	快门信号变量记录失败。
W6026	步骤 XX 的预计应力在范围外。
W7000	在压力测量模式下，除了伺服焊枪轴外其余不能运行。
W7001	检测到板间缝隙异常。

W7002	检测到板间缝隙异常。
W7003	焊极修磨中发现了异物。
W7004	修正值超出工件的异常范围。
W7009	工件未被检测到。
W7010	焊枪接触检测信号已被输入。
W7011	电极间的距离超过设定值。
W7012	面位置自动修正失败。
W7500	偏离了前停止位置，不能继续检查运动。
W7501	电源电压低，不能执行程序。
W8400	不能达到设轴 XX 定力值。
W8800	指令值超出了虚拟安全围栏。(球 XX、规定直线 XX)
W8801	指令值超出了虚拟安全围栏。(球 XX、规定区域 Z 上限)
W8802	指令值超出了虚拟安全围栏。(球 XX、规定区域 Z 下限)
W8803	指令值侵入限制区域。(球 XX、部分 XX 直线 XX)
W8804	指令值侵入限制区域。(球 XX、部分 XX Z 上限)
W8805	指令值侵入限制区域。(球 XX、部分 XX Z 下限)
W8806	指令值超出了虚拟安全围栏。(方形、规定直线 XX)
W8807	指令值超出了虚拟安全围栏。(方形、规定区域 Z 上限)
W8808	指令值超出了虚拟安全围栏。(方形、规定区域 Z 下限)
W8809	指令值侵入限制区域。(方形、部分 XX)
W8810	指令值超出了虚拟安全围栏。(LinkXX、规定直线 XX)
W8811	指令值超出了虚拟安全围栏。(LinkXX、规定区域 Z 上限)
W8812	指令值超出了虚拟安全围栏。(LinkXX、规定区域 Z 下限)
W8813	指令值侵入限制区域。(LinkXX、部分 XX 直线 XX)
W8814	指令值侵入限制区域。(LinkXX、部分 XX Z 上限)
W8815	指令值侵入限制区域。(LinkXX、部分 XX Z 下限)
W8851	检测到区域干涉。
W8852	检测到手臂干涉。(XX, XX)
W8853	ILL)检测到手臂干涉。(XX, XX)
W8854	ILL)通信超时。
W8855	ILL)顺序处理要求完成等待超时错误。
W8856	ILL)顺序处理完成等待超时错误。
W8857	ILL)顺序处理系统错误。
W8858	ILL)生成/设置处理完成等待超时错误。
W8859	ILL)Inter lock less 功能系统错误。
W8860	[手臂控制板]自 IL 服务器来的数据无效。
W8900	动作限制信号已输入，不能动作。

E0001	未知错误。
E0002	[伺服板 XX] CPU BUS 错误。
E0100	存在异常的注释语句。
E0101	标签不存在。
E0102	变量未定义。
E0103	位置数据未定义。
E0104	字符串变量未定义。
E0105	程序或标签未定义。
E0106	数值超出范围。
E0107	无数组下标。
E0108	除数为 0。
E0109	浮点数溢出。
E0110	字符串过长。
E0111	试图以负指数值进行运算。
E0112	表达式太过复杂。
E0113	没有可计算的表达式。
E0114	SQRT 参数为负数。
E0115	数组下标值超出范围。
E0116	自变量数值不完整或缺少。
E0117	错误的关节轴号。
E0118	子程序调用过多。
E0119	子程序不存在。
E0120	没有跳转目标程序。
E0121	不能指定与跳转源程序同名的目标程序。
E0900	辅助一体型步指令校验和错误。
E0901	步数据损坏。
E0902	表达式数据损坏。
E0903	系统数据校验和错误。
E1000	ADC 通道错误。
E1001	ADC 输入范围错误。
E1002	PLC 接口错误。
E1003	内置 PLC 未安装。
E1004	INTER-bus 板未准备好。
E1005	自旋轴编码器差值错误。
E1006	触摸拼开关短路。
E1007	电源程序板未安装。
E1008	第二电源程序板未安装。

E1009	第 XXI/O 板未安装。
E1010	电源程序检测错误。
E1011	内置式程序板未安装。
E1012	RI/O 板或 C-NET 板未安装。
E1013	INTER-BUS 板未安装。
E1014	通信用双端口内存未安装。
E1015	放大器接口板未安装。代码=XX
E1016	第 XX CC-LINK 板未安装。
E1017	PLC 错误。错误代码=十六进制码 XX
E1018	INTER-BUS 状态错误。
E1019	安全单元电源程序板未安装。
E1020	外部设备异常。
E1021	手臂 ID 板出错。代码=XX
E1022	电源程序板出错。代码=XX
E1023	机器人网络通信错误。
E1024	外部轴释放程序错误。代码=XX
E1025	外部轴连接程序错误。代码=XX
E1026	主 CPU ID 不匹配。
E1027	安全回路被切断。
E1028	轴 XX 电机过载。
E1029	编码器旋转数异常。轴 XX
E1030	编码器数据异常。轴 XX
E1031	编码器数据计算错误。轴 XX
E1032	ABS (绝对) 和 INC (增量) 编码器数据不一致。轴 XX
E1033	编码器线路失误。轴 XX
E1034	编码器初始化错误。轴 XX
E1035	编码器响应错误。轴 XX
E1036	编码器通信错误。轴 XX
E1037	编码器数据转换错误。轴 XX
E1038	编码器绝对值道错误。轴 XX
E1039	编码器增量脉冲错误。轴 XX
E1040	编码器的 MR—传感器失误。轴 XX
E1041	限位开关动作。轴 XX
E1044	目标位置在指定范围外。
E1045	(点焊) 焊枪和夹具不匹配。
E1046	起始点与终止点距离过短。
E1047	轴号不适用于传送带跟随模式。

E1048	调零偏差数据非法。
E1049	当前位置超出指定区域。
E1050	编码器和制动器的断电信号未专用。
E1051	错误的成对型 OX 输出。
E1052	工件检测信号未专用。
E1053	工件检测信号已输入。
E1054	不能执行运行指令。
E1055	圆弧的起始点位置错误。
E1056	主机器人已存在。
E1057	检查哪个机器人被指定为 MASTER/ALONE。
E1058	SLAVE (从) 机器人已存在。
E1059	不是协调运动指令。
E1060	不能在后退检查模式下执行。
E1061	不能在 ONE 程序中执行。
E1062	在运动到起始姿态时, 轴 JT2 和 JT3 互相干涉。
E1063	在运动到终止姿态时, 轴 JT2 和 JT3 互相干涉。
E1064	托盘号非法。
E1065	工件号非法。
E1066	非法的成型号。
E1067	成型类型非法。
E1068	工件数据非法。
E1069	托盘数据非法。
E1070	ON/ONI 信号已输入。
E1071	XMOVE 信号已输入。
E1072	原点位置数据未定义。
E1073	定时器编号非法。
E1074	超出最大信号编号。
E1075	夹具编号非法。
E1076	不能使用负的定时值。
E1077	无设置值。
E1078	信号编号非法。
E1079	不能使用专用信号。
E1080	非 RPS 模式。
E1081	不能使用负值。
E1082	超出运动范围的绝对下限值。
E1083	超出运动范围的绝对上限值。
E1084	超出运动范围的设定下限值。

E1085	超出运动范围的设定上限值。
E1086	轴 XX 的起始点超出运动范围。
E1087	轴 XX 的终止点超出运动范围。
E1088	目标位置超出运动范围。
E1089	在当前位置状态下不能执行直线运动。
E1090	外部调节数据未输入。
E1091	外部调节数据异常。
E1092	调节数据超出限值。
E1093	执行调节动作的运动指令不正确。
E1094	关节轴号非法。
E1095	在 PC 程序中不能执行运动指令。
E1096	辅助数据设置不正确。
E1097	缺少 C1MOVE 或 C2MOVE 指令。
E1098	C1MOVE (CIR1) 指令需先于 C2MOVE 指令。
E1099	不能创建弧形路径, 请检查 3 点的位置。
E1100	在焊接规格中不能执行。
E1101	只能在焊接规格中执行。
E1102	选件未设置, 不能被执行。
E1103	超出输送装置的位置。
E1104	SPINMOVE 指令太多。
E1105	目的位置在指定保护空间内。
E1106	在该机器人中不能执行。
E1107	不能使用 SEPARATE CONTROL。
E1108	机器人网络 ID 号重复。
E1109	传送装置接口板未安装。
E1110	组未预置。
E1111	因为运动限制, 轴 XX 不能移动。
E1113	没有收到工件检测信号。
E1114	在协调控制中中断。
E1115	协调控制的强制终止。
E1116	旋转轴未在各 360 度处停止。
E1117	处理超时。
E1118	轴 XX 的命令值突然改变。
E1119	轴 XX 的命令值超出运动范围。
E1120	当前命令造成第 2 轴和第 3 轴干涉。
E1121	其它机器人已在干涉区域。
E1122	电机电源意外 OFF。

E1123	轴 XX 速度错误。
E1124	轴 XX 的偏差错误。
E1125	轴 XX 的速率路包络错误。
E1126	轴 XX 的命令速度错误。
E1127	轴 XX 的命令加速度错误。
E1128	轴 XX 的目标位置与当前位置间的不一致错误。
E1129	外部轴 XX 在保持时被移动。
E1130	检测到 XX 的冲突。
E1131	检测到 XX 意外的震动。
E1132	电机电源关断，测量停止。
E1133	输送装置达到最大位置值。
E1134	传送装置的工件传送节距异常。
E1135	电机电源 OFF。
E1136	标准终端未连接。
E1137	不能对示教器输入输出。
E1138	辅助终端未连接。
E1139	DA 板未安装。
E1140	无传送装置轴。
E1141	传送装置的传送超出同步区域。
E1142	无走行轴。
E1143	未设置传送装置轴编号。
E1144	无手臂控制板。
E1145	不能使用指定通道，已在使用中。
E1146	[LSEQ] 处理过程因超时而终止。
E1147	不能打开安装文件，所以不能设置为出厂状态。
E1148	不能读取安装文件，所以不能设置为出厂状态。
E1149	不能打开安装数据，所以不能设置为出厂状态。
E1150	不能读取安装数据，所以不能设置未出厂状态。
E1151	设置为出厂状态所需的数据过多。
E1152	出厂状态的安装数据名称过长。
E1153	电源程序板检测错误。代码=XX
E1154	选件的 SIO 端口未安装。
E1155	A/D(模/数)转换器未安装。
E1156	[手臂控制板] 处理过程超时。
E1157	手臂 ID 接口板失误。代码=XX
E1158	(SSCNET) 轴 XX 的伺服失误。
E1159	(SSCNET) 伺服的失误代码为 XX。



E1160	(SSCNET) 轴 XX 的伺服失误及监视器设置错误。
E1161	该型号机器人不支持自动工具登录功能。
E1162	在 XX 重力补偿值通道中的缓冲区溢出。
E1163	机器人停止在检查操作区域中。
E1164	[LSEQ] 在控制电源开通时，程序发生执行错误。代码=XX
E1165	不能下载外部轴参数。(Jt-A)
E1166	通道未指定关节轴号。(Jt-A)
E1167	不能下载外部轴参数。(Jt-B)
E1168	通道未指定关节轴号。(Jt-B)
E1169	伺服参数改变程序错误。代码=XX
E1170	从站未准备好。
E1171	CC-LINK 通信板未安装。
E1172	焊接通信板未安装。
E1173	轴 XX 伺服通信错误。
E1174	0 号 AD 板未安装。
E1175	调零偏移数据非法。机器人=XX
E1176	(SSCNET) 外部轴参数下载错误。
E1177	(SSCNET) 通道未指定关节轴号。
E1179	当前的下扭曲补偿值过大。轴 XX
E1180	外部轴参数下载错误。轴 XX
E1181	编码器电池电压低。伺服=XX
E1182	编码器电池电压低。外部轴=XX
E1183	轴 JT5 不是 0 度，请将轴 JT5 设置为 0 度。
E1184	运动的配置非法。
E1185	轴 1 和轴 2 在起始位置干涉。
E1186	轴 1 和轴 2 在终止位置干涉。
E1187	轴 1 和轴 2 间的当前命令互相干涉。
E1188	(SSCNET) 伺服参数改变程序中的错误。代码=XX
E1189	(SSCNET) 轴 XX 再生错误。代码=XX
E1190	(SSCNET) 轴 XX 的速度错误。代码=XX
E1191	(SSCNET) 轴 XX 电机过载。代码=XX
E1192	(SSCNET) 轴 XX 的偏差错误。代码=XX
E1193	(SSCNET) 轴 XX 编码器电池电压低。代码=XX
E1194	(SSCNET) 轴 XX 的参数警告。代码=XX
E1195	(双重伺服) 主控轴与从动轴之间的偏差异常。
E1196	提升装置锁定中，不能运动。
E1197	补偿用 LS 信号未专用化。

E1198	刹车检查流程异常。
E1199	软件版本不支持刹车检查功能。
E1200	(双重伺服) 电流偏差不能补偿。(偏差 XX)
E1201	未安装干涉检查板。
E1202	声音记录器停止失败。
E1203	LS 基准位置未登录。
E1204	当前伸出量超出了允许值。
E1205	总伸出量超出了允许值。
E1207	手臂上 ID 板的种类不同。
E1208	伺服参数下载出错。轴 XX
E1209	伺服参数上载出错。轴 XX
E1210	未受保护程序不能执行。
E1211	存储器已满，不能复制程序。
E1212	存储器已满，程序复制已被挂起。
E1213	动作开始点的 JT4 干涉。
E1214	动作目标点的 JT4 干涉。
E1215	JT4 的指令值干涉。
E1216	动作开始点的 JT5、6 干涉。
E1217	动作目标点的 JT5、6 干涉。
E1218	JT5、6 的指令值干涉。
E1219	路径上信号输出命令过多。
E1220	信号输出点不在路径上。
E1221	信号编号过多。
E1222	动作开始点/结束点不存在。
E1223	前/后动作命令不合适。
E1224	多个信号输出点相同。
E1225	修正结束命令没有找到。
E1228	动作开始点的 JT4 不是 0 度。
E1229	动作目标点的 JT4 不是 0 度。
E1230	动作开始点的法兰面向上。
E1231	目标点的法兰面向上。
E1232	可选的 CPU 板未安装。
E1233	I 各轴/I 直线信号未指定。
E1234	I 各轴/I 直线信号未检测到。
E1235	分离运行用 I/O 板未安装。
E1236	修正距离过大。
E1237	视觉识别出错。

E1238	视觉通信出错。
E1239	FRAME (框架) 修正模式中不能使用此命令。
E1240	BASE FRAME (基准框架) 未从视觉单元送出。
E1241	FN481 的参数不正确。
E1242	BASE FRAME (基准框架) 登录不能超过 99。
E1243	照相机 XX 未连接，不能执行。
E1244	动作开始点的 JT1、2 和地面干涉。
E1245	动作目标点的 JT1、2 和地面干涉。
E1246	JT1、2 的指令值和地面干涉。
E1247	编码器数据不能确定。(轴 XX)
E1248	编码器的 EEPROM 访问标志为忙状态。(轴 XX)
E1249	编码器内部温度超出了报警温度。(轴 XX)
E1250	编码器的旋转数超出了检测可能范围。(轴 XX)
E1251	编码器内的 EEPROM 出现访问错误。(轴 XX)
E1252	第 XX 轴的旋转量数据 (编码器内部) 异常。(轴 XX)
E1253	编码器通信中的应答数据不一致。(轴 XX)
E1254	组 XX 的 MC 为 OFF，不能动作。
E1255	未选中机器人的马达电源被开启了。
E1256	机内电磁阀/传感器、错误复位接口板不存在。
E1257	组 XX 的 MC 在个别再现动作中被关断为 OFF。
E1258	MC 在动作中被关断为 OFF。
E1259	码垛命令的结构异常。
E1260	码垛动作过程中不能执行。
E1261	码垛动作已中断。
E1262	编码器旋转速度超限。(轴 XX)
E1263	编码器温度超限。(轴 XX)
E1264	无限旋转轴的速度偏差异常。(轴 XX)
E1267	编码器的初始设置异常。(轴 XX)
E1268	编码器线路上有断路或编码器波特率的设置失败。(轴 XX)
E1269	此程序是其他机器人用的。
E1270	位姿变量无效。
E1271	在动作起始点手臂与地面干涉。
E1272	在动作目标点手臂与地面干涉。
E1273	指令值的位姿与地面干涉。
E1274	在高负荷模式下第 XX 轴的速度超过限值。
E1275	在高负荷模式下第 XX 轴超出运动范围。
E1276	在高负荷模式下动作起始点的第 XX 轴超出运动范围。

E1277	在高负荷模式下动作目标点的第 XX 轴超出运动范围。
E1278	手腕不能再弯曲了。
E1279	请在动作起始点使手腕垂直向下。
E1280	请在动作目标点使手腕垂直向下。
E1281	JT4 的指令值超过限制。
E1282	因为组 XX(第 XX 轴)的 MC 为 OFF，不能动作。
E1283	错误分析)E1035 错误常常发生。JTXX
E1284	错误分析)E1035 错误和 E1029 错误同时发生。轴 XX
E1285	错误分析)E1035 错误和 E1036 错误同时发生。轴 XX
E1286	错误分析)E1035 错误和 E1032 错误同时发生。轴 XX
E1287	检测到第 XX 轴的电源模块错误（上）。
E1288	检测到第 XX 轴的电源模块错误（下）。
E1289	[伺服板 XX]检测到同步错误。(伺服 FPGA)
E1290	第 XX 轴电流传感器的电压超过上限值。
E1291	第 XX 轴电流传感器出错或线路断开。(U)
E1292	[伺服板 XX]从 MCXX 电源异常信号输入。
E1293	[伺服板 XX]电流 FB 增益设置值异常。
E1294	[伺服板 XX]IO 24 V 降低。
E1295	[伺服板 XX]机器阀 24 V 降低。
E1296	[伺服板 XX]安全电路 LS 状态信号不一致。
E1297	[伺服板 XX]安全电路中的内部气压降低跳线配线不一致。
E1298	[伺服板 XX]LS 超越控制开关的接点不一致。
E1299	[伺服板 XX]安全电路中的内部气压降低跳线不连接。
E1300	[伺服板 XX]DC 电源异常。(24 V)
E1301	[伺服板 XX]软件和伺服板之间的编码器型类不一致。
E1302	[MCXX]OFF 检测异常。(伺服板 XX)
E1303	[MCXX]安全继电器 OFF 检测异常。(伺服板 XX)
E1304	[MCXX]K1 错误动作。(伺服板 XX)
E1305	[MCXX]K2 错误动作。(伺服板 XX)
E1306	[MCXX]冲击控制继电器错误动作。(伺服板 XX)
E1307	[MCXX]安全继电器 KS1 错误动作。(伺服板 XX)
E1308	[MCXX]安全继电器 KS2 错误动作。(伺服板 XX)
E1309	[MCXX]安全继电器 KS3 错误动作。(伺服板 XX)
E1310	[MCXX]马达 ON 继电器错误动作。(伺服板 XX)
E1311	[MCXX]安全电路中的马达 OFF 继电器错误动作。(伺服板 XX)
E1312	[MCXX]安全电路中的马达 OFF 继电器状态信号不一致。(伺服板 XX)
E1313	[MCXX]安全电路中的接点状态信号不一致。(伺服板 XX)

E1314	[MCXX] 三相整流模块过热。(伺服板 XX)
E1315	[I/O 板 (NoXX)] 检测到看门狗错误。
E1316	[I/O 板 (NoXX)] 访问错误。(地址 XX)
E1317	[伺服板 (NoXX)] 监控的响应异常。[代码:XX]
E1318	[MCXX] DC20 V 电源异常。(伺服板 XX)
E1319	机器阀/传感器、错误复位接口板 NO.2 未安装。
E1321	[主 CPU 板] 与伺服板 (XX) 通讯异常。(代码: XX)
E1322	电源程序板和 MCXX 之间安全电路的单路/双路设置不一致。(伺服板 XX)
E1323	伺服板 XX 和 MCXX 之间安全电路的单路/双路设置不一致。
E1324	电源程序板和伺服板 XX 之间的安全电路断开。
E1325	伺服板 XX 和 MCXX 之间的安全电路断开。
E1326	安全栅栏打开。
E1327	[电源程序板] 安全电路中的马达 OFF 继电器状态信号不一致。
E1328	[电源程序板] 安全电路中的马达 OFF 继电器错误动作。
E1329	[电源程序板] 安全电路中的示教/再现开关状态异常。
E1330	[电源程序板] IO 24 V 降低。
E1331	[电源程序板] 控制盘内温度过高。
E1332	[电源程序板] 从伺服板 XX 电源异常信号输入。
E1333	马达 ON 状态信号切断, 为 OFF。(伺服板 XX) (MCXX)
E1334	示教/再现开关状态异常 (安全电路和监控之间不一致)。
E1335	马达电源切断, 为 OFF。(伺服板 XX) (MCXX) (代码 XX)
E1336	[伺服板 XX] 与主 CPU 板通讯异常。
E1337	[MCXX] 刹车电源异常。(伺服板 XX)
E1338	[MCXX] P-N 间电压降低。(伺服板 XX)
E1339	[MCXX] P-N 间电压过高。(伺服板 XX)
E1340	[MCXX] 再生时间超时。(伺服板 XX)
E1341	[MCXX] 再生电阻过热。(伺服板 XX)
E1342	分离马达线束断线或机器人手臂的温度过高。(MCXX)
E1343	刹车配线与软件设置不一致。(JtXX)
E1344	第 XX 轴电流传感器故障或线路断线。(V)
E1345	[伺服板 XX] 限位开关线路断线。
E1346	第 XX 轴编码器的全部数据获取失败。
E1347	[MCXX] 不是目的地 MC。(伺服板 XX)
E1348	[MCXX] MC 控制规格不匹配。(伺服板 XX)
E1349	[MCXX] MC 的防爆/不防爆规格与软件中设置的不一致。(伺服板 XX)
E1350	[MCXX] MC 的规格错误。[代码 XX] (伺服板 XX)
E1351	[MCXX] MC OFF 延迟规格不匹配。(伺服板 XX)

E1352	在软件和电源模块之间设置的代码不一致。(第 XX 轴)
E1353	[主 CPU 板]CPU 温度异常。
E1355	伺服接口代码通讯出错。(代码: XX)
E1356	工具形状未设定。
E1357	外部轴参数数据下载失败。(Jt-C)
E1358	轴编号没有分配到指定频道。(Jt-C)
E1359	第 XX 轴的 U 相电流超负荷。
E1360	第 XX 轴的 V 相电流超负荷。
E1361	第 XX 轴的 W 相电流超负荷。
E1362	[伺服板 XX]工具中心点的速度超出安全速度。
E1363	[伺服板 XX]法兰中心点的速度超出安全速度。
E1365	2 处以上的示教键开关为 ON。
E1366	在编号 XX ANYBUS 接口板看门狗错误。
E1367	K1481 的参数值出错。
E1377	[主 CPU 板]工具中心点的速度超出安全速度。
E1378	[主 CPU 板]法兰中心的速度超出安全速度。
E1379	[主 CPU 板]轴 XX 的偏差错误。
E1382	[伺服板 XX] 机内阀・传感器接口板未安装。
E1384	总电力消耗超过允许值。
E1385	限位开关(第 XX 轴)动作, 或 RSC 检测出异常。
E1386	限位开关信号线断开, 或 RSC 检测出异常。
E1387	(SSCNET) 访问错误的地址。(call:XX)(地址:XX)
E3800	第 XX 轴伺服放大器过热。
E3801	第 XX 轴放大器主回路电源电压过低。
E3803	第 XX 轴放大器速度控制异常。
E3804	第 XX 轴放大器速度反馈异常。
E3805	第 XX 轴放大器位置偏差异常。
E3806	第 XX 轴放大器伺服就绪信号没有开为 ON。
E3807	第 XX 轴放大器的 IPM 过热。
E3808	外部紧急停止开关被按下。
E3809	刹车释放信号异常。
E3810	电源顺序就绪信号关闭。
E3811	第 XX 轴放大器的指令值急变。
E3900	移动的工具与选择的工具的数据不一致。 z
E4000	数据通信错误。
E4001	数据读取错误。
E4002	数据写入错误。

E4003	文件访问中意外错误。
E4004	通信在试错误。
E4005	通信过程被停止。
E4006	请求后未收到数据。
E4007	接收数据过长。(最多=255 字符)
E4008	通信中接收到异常数据(EOT)。
E4009	通信超时错误。
E4010	终端已被使用。
E4011	通信端口已被使用。
E4012	等待 PROMPT 的输入，请连接输入装置。
E4013	TELNET) 发送错误。代码=XX
E4014	TELNET) 接收错误。代码=XX
E4015	TELNET) IAC(应用间通信)接收错误。代码=XX
E4016	TELNET) 关闭失败。代码=XX
E4017	TELNET) 主插口关闭失败。代码=XX
E4018	TELNET) 系统错误。代码=XX
E4019	TCPIP) 插口打开错误。代码=XX Dst. IP=XX. XX. XX. XX
E4020	TCPIP) 插口关闭错误。代码=XX Dst. IP=XX. XX. XX. XX
E4021	TCPIP) 通信错误。代码=XX Dst. IP=XX. XX. XX. XX
E4022	TCPIP) 消息太长。
E4023	TCPIP) 不能到达主机。
E4024	TCPIP) 通信时间超时。Dst. IP=XX. XX. XX. XX
E4025	TCPIP) 连接失败。
E4026	TCPIP) 无缓冲空间。
E4027	TCPIP) 坏的插口。
E4028	FTP) 数据收到错误。代码=XX
E4029	FTP) 数据发送错误。代码=XX
E4030	FTP) 服务器未识别命令。代码=XX
E4031	FTP) 与 FTP 服务器断开失败。代码=XX
E4032	FTP) 检测到未注册的 OS。
E4033	FTP) 与服务器连接失败。代码=XX
E4034	FTP) 接收 HOST OS 信息失败。代码=XX
E4035	FTP) TCP/IP 未初始化。
E4036	FTP) 当前的 FTP 服务繁忙。
E4037	FTP) 自动存盘失败。
E4053	通道错误。
E4054	TCPIP) 不能执行，因为以太网卡未安装。

E4055	TCP) 不能创建套接字。
E4056	TCP) 该端口不在 LISTEN(SOCK)。
E4057	TCP) 套接字 ID 非法。
E4059	ASCYCLE 通信接收错误。代码=XX
E4060	[手臂控制板] ASCYCLE 通信接收错误。代码=XX
E4061	接收到的计量孔数据超出允许值。
E4062	主控/从动数据未登录。
E4063	基准点(参考点)数据未登录。
E4064	3D 校准/测量模式两个都为 ON。
E4065	接收数据的变量未登录。
E4066	接收数据的变量已破坏。
E4067	接收的数据已破坏。
E4068	起始码不正确。
E4069	结束码不正确。
E4070	3D 摄像头组编号未指定。
E4071	3D 摄像头组编号不正确。
E4072	通信开始等待超时。
E4073	手臂接口板来的伺服 OFF 信号无应答。
E4074	[伺服板 XX]MCXX 没有响应。(代码:XX)
E4075	[伺服板 XX]与 MCXX 通讯异常。(代码:XX)
E4076	[MCXX]与伺服板 XX 通讯异常。(代码:XX)
E4077	[伺服板 XX]与主 CPU 板的通讯出现异常。(代码:XX)
E4078	[伺服板 XX]与主 CPU 板的指令通讯出现异常。(代码:XX)
E4082	[伺服板 XX] 与 Cubic-S 的通信异常。(代码:XX)
E4500	ANYBUS) IN-AREA 请求超时。XX
E4501	ANYBUS) OUT/FB。CTRL 释放超时。XX
E4510	DN) 主状态。XX
E4511	DN) 节点状态。XX
E4512	ABM-DN) 邮箱错误。
E4520	ABMA-PDP) 状态 STOP。XX
E4521	ABMA-PDP) 状态 OFFLINE。XX
E4522	ABMA-PDP) I/O 数据通信错误。XX
E4523	ABMA-PDP) 发送 I/O 数据超时。XX
E4524	ABMA-PDP) 接收 I/O 数据超时。XX
E4525	ABMA-PDP) 发送消息超时。XX
E4526	ABMA-PDP) 接收消息超时。XX
E4527	ABMA-PDP) 校验配置数据。XX



E4528	PROFIBUS) 检测到从站诊断错误响应。XX
E4529	PROFIBUS) 检测到统计计数器错误响应。XX
E4530	DN) Device Net 电缆未连接。
E4531	CC-LINK) 通信已断开。XX
E4532	CC-LINK) 环境设定错误。
E4533	CC-LINK) 看门狗超时。
E4534	CC-LINK) 参数设定错误。XX
E4535	CC-LINK) 参数设定超时。
E4536	CC-LINK) 主控板异常。XX
E4537	CC-LINK) 主控板初始化错误。XX
E4538	CANopen) 网络断开。
E5000	连接的允许信号未被开启。
E5001	RWC 的类型不是过程控制类型。
E5002	1GS 板不是过程控制类型。
E5003	非法的伸出(缩回)控制信号。
E5004	焊接完成信号已输入。
E5005	(点焊) 焊接顺序设置数据异常。
E5006	CLAMP SPEC 未被设置成 PULSE。
E5007	伺服焊枪未连接或连接了错误的焊枪。
E5008	端部磨损(STAGE1) 未执行。
E5009	工件检测信号(焊枪顶端接触信号) 未设置。
E5010	伺服焊枪机械参数未设置。
E5011	用于伺服焊枪轴的夹具号已被设置。
E5012	偏移数据异常, 不能改变焊枪。
E5013	同一步内不能切换多个焊枪。
E5014	焊枪连接到另一个轴上了, 不能执行。
E5015	焊枪状态数据与夹具条件不一致。
E5016	SRVPRESS 的数据错误。
E5017	磨损基本数据未登录。
E5018	未检测到焊接完成信号。
E5019	检测到焊接错误信号。
E5020	缩回位置监视错误。
E5021	伸出位置监视错误。
E5022	当前焊枪缩回位置与目标置不同。
E5023	磨损异常, 不能测量。
E5024	未检测到加压完成信号。
E5025	未检测到焊枪打开完成信号。

E5026	(点焊) RWC 错误。XX
E5027	机器人焊接中停止。
E5028	不能达到设置力值。
E5029	焊枪端部粘连。
E5030	铜板磨损超限。步号=XX
E5031	焊接完成信号未关断。
E5032	标定未正常结束。
E5033	厚度异常不能焊接。
E5034	点焊枪端部磨损超限。(移动侧)
E5035	点焊枪端部磨损超限。(固定侧)
E5036	焊枪状态数据错误。
E5037	端部磨损超限。XX:
E5038	电弧检测信号未关闭。
E5039	RWC 通信接口板没有响应。
E5040	射枪已连接, 不能再连接。
E5041	不能断开射枪, 因为射枪已断开。
E5042	射枪编号未定义或射枪型式不是伺服枪。
E5043	焊接机的通信错误。(编码 XX)
E5044	焊接数据获取失败(焊接机 XX)
E5045	焊接数据变更失败(焊接机 XX)
E5046	焊接异常发生。
E5047	焊接项目获取中。请等待完成。
E5048	焊接机未连接或焊接项目未取得。(焊接机 XX)
E5049	系列编号信号异常。
E5050	此焊接机无焊接结果追溯功能。
E5051	因未连接工具切换轴, 不能标定。
E5052	加压力测量值异常。
E5053	加压力传感器断线或有故障。
E5054	示教器的自动/手动选择开关设置为手动。
E5055	示教器的自动/手动选择开关设置为自动。
E5056	集成焊接机板未安装。
E5057	集成焊接机板的初始化失败。
E5058	集成焊接机(电元公司)未连接。(焊接机 XX)
E5059	集成焊接机(电元公司)的通信失败。(焊接机 XX)
E5060	保护集成焊接机板。(焊接机 XX)
E5061	集成焊接机(电元公司)板数据处理未执行。(焊接机 XX)
E5062	集成焊接机(电元公司)数据处理错误。(焊接机 XX)

E5063	出现焊接错误。（代码 XX）
E5064	集成焊接机（电元公司）的焊接取消。（焊接机 XX）
E5065	集成焊接机（电元公司）出错。（焊接机 XX）
E5066	焊接完毕的等待时间超时。（焊接机 XX）
E5067	磁铁控制异常。（焊接机 XX）
E5075	伺服修磨轴不存在。
E5500	视觉板未安装。
E5501	（视觉）摄像机未连接。
E5502	（视觉）参数错误。
E5503	（视觉）符号错误。
E5504	（视觉）名字错误。
E5505	（视觉）图像内存错误。
E5506	（视觉）柱状图数据错误。
E5507	（视觉）模式错误。
E5508	（视觉）不透明度（/颜色）错误。
E5509	（视觉）摄像机输入指定错误。
E5510	（视觉）摄像机通道号错误。
E5511	（视觉）窗口号错误。
E5512	（视觉）坐标数据错误。
E5513	（视觉）编号错误。
E5514	（视觉）图像代码错误(binary/multi)。
E5515	（视觉）阈值错误。
E5516	（视觉）PROTO（/TEMPLATE）未登录或已存在。
E5517	（视觉）标定数据未登录。
E5518	（视觉）图像指针未初始化。
E5519	（视觉）PROTO 对象的样本太多。
E5520	（视觉）检测到的目标太多。
E5521	（视觉）视觉命令未初始化。
E5522	（视觉）系统登录错误数据。
E5523	（视觉）图像处理错误。
E5524	（视觉）声音端口已指派给另一功能。
E5525	（视觉）数据存储区不够。
E5526	（视觉）错误的同步模式。
E5527	（视觉）视觉处理中。
E5528	（视觉）图像捕捉错误。
E5529	（视觉）超时或缓冲区溢出。
E5530	（视觉）闪存写入失败。

E5531	(视觉) Proto 数据异常, 所以初始化。
E5532	(视觉) 工件检测失败。
E5533	(视觉) 初始化错误。代码=XX
E5534	(视觉) 视觉系统错误。
E5535	(视觉) 指定的运动模式不正确。
E5536	(视觉) 摄像机/投光灯参数不适合。
E5537	(视觉) 摄像机开关指派不正确。
E5538	(视觉) 此平面已指派给另一摄像机。
E5539	(视觉) 边界未发现。
E5540	(视觉) HIS 数据不合适。
E5541	(视觉) H 的数据宽度超过 128。
E5542	(视觉) 摄像机的远距离图像输入单元未设置。
E5543	(视觉) 不能计算设置的边界点。
E5544	(视觉) 检查在配置中设置的颜色转换表的类型。
E5545	(视觉) 区域尺寸不正确。
E5546	(视觉) 狭缝图像不存在。
E5547	(视觉) 相关向量的编号错误。
E5548	(视觉) 向量数据不适合。
E5549	(视觉) X-Fit 环境未设置。
E5550	(视觉) 鼠标未初始化。
E5551	(视觉) 摄像机开关板未安装。
E6000	防爆示教器未连接。
E6001	XD (2) START 后的步必须是 LMOVE 或 HMOVE。
E6002	信号状态已输入。
E6003	开门检测信号未专用化。
E6004	未检测到位置数据。
E6005	barrier 单元设置错误。
E6006	未检测到信号。
E6007	腕关节不能进一步伸直(奇点 1)。
E6008	腕关节不能进一步弯曲(奇异点 2)。
E6009	吹扫气流流量不够。
E6010	超出 XYZ 运动区域。
E6011	内部气压低。
E6012	焊枪间的相对距离过近。
E6013	程序队列中无存储器空间。
E6014	延迟启动队列中无存储器空间。
E6015	专用信号未专门化。

E6016	机器人手臂伸展开（奇异点 3）
E6017	超出了机械 XYZ 动作极限。
E6018	涂装设备控制板异常。（代码 XX）
E6019	涂装设备控制板未安装。
E6020	监视机器人设定的 ID 重复。
E6021	机器人没有指定各自的 ID。
E6022	相互等待 ID 重复。
E6023	相互等待通信出错。
E6024	手腕不能再向旁边转（奇异点 1）
E6025	（传送装置同步通讯）传送装置位置数据接收错误。
E6026	（枪间的相对距离检测）X 方向距离过近。（ID:XX）
E6027	（枪间的相对距离检测）Y 方向距离过近。（ID:XX）
E6028	（枪间的相对距离检测）Z 方向距离过近。（ID:XX）
E6029	[伺服板 XX] 安全电路中的内压降低状态信号不一致。
E6030	[伺服板 XX] 内压降低。
E6031	监视机器人的 ID 无效。
E6032	[吹扫控制板] 内压低。（吹扫中）
E6033	喷涂机器控制过程出错。（代码 XX）
E6034	旋转台的旋转指令异常。
E6500	无焊接接口板。
E6501	未发现第二块焊接接口板。
E6502	弧焊失败。
E6503	焊丝粘连。
E6504	起弧失败。
E6505	弧焊绝缘不良。
E6506	焊枪干涉。
E6507	插补数据非法。
E6508	无用于极性比例控制的 D/A 板。
E6509	未检测到工件。
E6510	未定义的深侧方向。
E6511	深侧点数不足。
E6512	未定义的母工件或子工件。
E6513	深侧点过多。
E6514	工件规格错误。
E6515	指定的深侧点错误。
E6516	焊丝检测失败。
E6517	焊接条件编号错误。

E6518	未设置焊接条件数据。
E6519	焊接条件数据超出范围。
E6520	激光传感器跟踪值超限。
E6521	超出激光传感器跟踪能力。
E6522	激光传感器不能检测焊接点。
E6523	焊枪和摄像机间的标定数据未准备好。
E6524	激光传感器计算出的数据有错误。
E6525	不能检测焊点，激光传感器跟踪已设置。
E6526	激光传感器控制器没有响应。
E6527	激光传感器通信错误。代码=XX
E6528	激光传感器未发现起始点。
E6529	激光传感器未发现结束点。
E6530	当使用激光传感器功能时，不能使用圆弧插补。
E6531	电机电源 OFF，不能启动激光。
E6532	无连接到激光传感器的通信板。
E6533	无 RTPM 板。
E6534	RTPM 的示教点过多。
E6535	RTPM 电弧传感器失误。
E6536	RTPM 电流偏差错误。
E6537	RTPM 跟踪值超出范围。
E6538	超出 RTPM 跟踪能力。
E6539	AVC 跟踪值超出范围。
E6540	超出 AVC 跟踪能力。
E6541	无 AVC 板。
E6542	AVC 电压偏差错误。
E6543	AVC 的示教点过多。
E6544	超电弧跟踪值超出范围。
E6545	超出超电弧跟踪能力。
E6546	未发现 Bead end。
E6547	未发现 Finish end。
E6548	超电弧的旋转超出正常偏差。
E6549	超电弧喷灯标定错误。
E6550	超电弧 Z 相刻度错误。
E6551	无超电弧板。
E6552	超电弧板失误。代码=XX
E6553	超电弧电流传感器失误。
E6554	超电弧电压传感器失误。

E6555	超电弧电流偏差错误。
E6556	超电弧放大器失误。代码=XX
E6557	无送丝控制板。
E6558	送丝控制错误。代码=XX。
E6559	送丝速度偏差错误。
E6560	焊接过程中，不能重新校准。
E6561	重新校准过程中，不能焊接。
E6562	电极粘连。
E6563	KHITS 系统异常。代码=XX。
E6564	弧焊焊接命令顺序不正确。
E6565	电弧焊接口板 (1LN) 未安装。机器人=XX
E6566	FN 命令的执行顺序不正确。
E6567	KLS 跟踪系统出错。代码=XX
E6568	跟踪系统去的命令执行失败。(指令 = XX)
E6569	示教的数据超出了倾斜补偿的极限。
E6570	焊接进行中，不能执行。
E6571	焊丝进给/缩回进行中，不能执行。
E6572	圆弧动作的示教点不足。
E6573	焊接机异常。(代码=XX)
E6574	焊接凹槽感测：边界未发现。
E6575	焊接凹槽感测：缝隙异常。
E6576	焊接机没有准备启动。(代码=XX)
E6577	错误复位后，再开始无补偿的动作。
E6578	在感测执行中控制电源切断。指定步骤后重试一次。
E6579	KI 命令的执行顺序错误。
E7000	伺服焊枪未连接。
E7001	位置数据包含了释放的焊枪状态数据。
E7002	目的位置远离目标点。
E7003	焊枪 XX 的空隙距离被设置成了 0 毫米。
E7004	焊枪端部磨损量超出限值。(移动侧)
E7005	焊枪端部磨损量超出限值。(固定侧)
E7006	夹具编号或射枪编号不是伺服焊枪。
E7007	焊接头磨损率未设置、空打模式的焊接头磨损基准数据不能登录。
E7008	射枪轴独立动作未完成。
E7009	伺服焊接枪的电流限制值异常。
E7500	检测到第 XX 轴冲突。
E7501	检测到第 XX 轴冲击。

E7502	AC Fail Process Error = XX
E7503	电源顺序设定数据不正确。
E7504	动作开始点的第 XX 轴干涉角错误。
E7505	动作目标点的第 XX 轴干涉角错误。
E7506	第 XX 轴的指令值干涉角错误。
E7507	SC1MOVE 命令后面需要有 SC1MOVE 或 SC2MOVE 命令。
E7508	SC2MOVE 命令之前需要有 SC1MOVE 命令。
E7509	圆弧或直线插补的条件未满足，不能执行。
E7510	当前的姿势不能动作。
E7511	刹车控制比特编号重复。
E7512	L3C1MOVE 命令后面需要有 L3C1MOVE 命令或 L3C2MOVE 命令。
E7513	L3C2MOVE 命令前面需要有 L3C1MOVE 命令
E7514	指定的参数无效。
E8200	不在协调模式中。
E8201	协调模式下运动指令总数不匹配。
E8202	协调模式下运动指令步不匹配。
E8203	协调模式下不能使用该指令。
E8204	协调组编号不正确。
E8205	JMASTER 机器人未设定。
E8206	协调动作中不支持接触感测。
E8207	JMASTER 机器人已存在。
E8208	WSLAVE 机器人已存在。
E8209	协调动作中不支持固定作业点动作。
E8210	WSLAVE 机器人未设定。
E8211	同步计数器超出。
E8212	协调模式下不能再执行非协调动作命令。
E8213	MASTER 机器人没有指定。
E8214	SLAVE 机器人没有指定。
E8400	伺服手爪在夹具 ON 步被打开。夹具=XX
E8401	伺服手爪的夹具位置错误。夹具=XX
E8402	不能达到设轴 XX 定力值。
E8403	NC 轴锁定信号未 OFF。
E8404	不能执行各轴动作以外的动作。
E8405	轴锁定状态下，试图动作 Matehan 轴。
E8600	(FSJ) 结合异常 XX
E8601	剩余的板厚超过下限值。。
E8602	在设定的加工时间过去前，到达下压量。。



E8603	在设定的加工时间过去后，未到达下压量。。
E8604	加压电缆断线。
E8605	请输入两个以上加压力设定值。
E8606	请按升序输入数据。
E8607	FSJ 接合计数器报警 (XX)
E8608	(FSJ) 加工条件设定值异常。
E8609	设定的加压力超出了最大加压力。
E8610	设定的旋转速度超过上限值。。
E8611	FSW 加工记录缓冲器已满。
E8612	在设定的时间内销轴或肩未到达目标位置。代码 XX)
E8800	指令值超出了虚拟安全围栏。(球 XX、规定直线 XX)
E8801	指令值超出了虚拟安全围栏。(球 XX、规定区域 Z 上限)
E8802	指令值超出了虚拟安全围栏。(球 XX、规定区域 Z 下限)
E8803	指令值侵入限制区域。(球 XX、部分 XX 直线 XX)
E8804	指令值侵入限制区域。(球 XX、部分 XX Z 上限)
E8805	指令值侵入限制区域。(球 XX、部分 XX Z 下限)
E8806	指令值超出了虚拟安全围栏。(方形、规定直线 XX)
E8807	指令值超出了虚拟安全围栏。(方形、规定区域 Z 上限)
E8808	指令值超出了虚拟安全围栏。(方形、规定区域 Z 下限)
E8809	指令值侵入限制区域。(方形、部分 XX)
E8810	指令值超出了虚拟安全围栏。(LinkXX、规定直线 XX)
E8811	指令值超出了虚拟安全围栏。(LinkXX、规定区域 Z 上限)
E8812	指令值超出了虚拟安全围栏。(LinkXX、规定区域 Z 下限)
E8813	指令值侵入限制区域。(LinkXX、部分 XX 直线 XX)
E8814	指令值侵入限制区域。(LinkXX、部分 XX Z 上限)
E8815	指令值侵入限制区域。(LinkXX、部分 XX Z 下限)
E8820	当前值超出了虚拟安全围栏。(球 XX、规定直线 XX)
E8821	当前值超出了虚拟安全围栏。(球 XX、规定区域 Z 上限)
E8822	当前值超出了虚拟安全围栏。(球 XX、规定区域 Z 下限)
E8823	当前值侵入了限制区域。(球 XX、部分 XX 直线 XX)
E8824	当前值侵入了限制区域。(球 XX、部分 XX Z 上限)
E8825	当前值侵入了限制区域。(球 XX、部分 XX Z 下限)
E8826	当前值超出了虚拟安全围栏。(方形、规定直线 XX)
E8827	当前值超出了虚拟安全围栏。(方形、规定区域 Z 上限)
E8828	当前值超出了虚拟安全围栏。(方形、规定区域 Z 下限)
E8829	当前值侵入了限制区域。(方形、部分 XX)
E8830	当前值超出了虚拟安全围栏。(LinkXX、规定直线 XX)

E8831	当前值超出了虚拟安全围栏。(LinkXX、规定区域 Z 上限)
E8832	当前值超出了虚拟安全围栏。(LinkXX、规定区域 Z 下限)
E8833	当前值侵入了限制区域。(LinkXX、部分 XX 直线 XX)
E8834	当前值侵入了限制区域。(LinkXX、部分 XX Z 上限)
E8835	当前值侵入了限制区域。(LinkXX、部分 XX Z 下限)
E8850	指令值生成已停止。
E8851	检测到区域干涉。
E8852	检测到手臂干涉。
E8853	指令值预测失败。
E8854	检测到停顿。
E8855	干涉检查板无响应。
E8856	干涉检查板和手臂控制板之间通信错误。
E8857	设定的机器人台数过多。
E8860	[手臂控制板]不能收到干涉检查板的数据。
E8861	IL 服务器和手臂控制板之间通信错误。
E8862	IL 服务器和手臂控制板之间通讯电缆断路。
E8900	工件未被正常握持。
E8901	工件已握持。
E8902	输入了动作限制信号，已停止。
E9000	JTXX 的控制杆断线。
E9300	因为第 XXd 轴不连接，不能旋转。
E9301	因为第 XX 轴无效，不能旋转。
E9302	第 XX 轴的旋转速度设置异常。
E9401	[Cubic-S]机器人控制器和 Cubic-S 间的 XX 不一致。
E9402	[Cubic-S]机器人控制器和 Cubic-S 间的 XX 不一致。
E9403	[Cubic-S (%d)]系统错误。(代码 XX XX)
E9404	[Cubic-S]参数可能被更改。关断后再开启控制器电源。
E9405	[Cubic-S]不能使用 Cubic-S。
E9406	[Cubic-S (XX)]XX (XX)从总是许可空间离开。(代码 XX)
E9407	[Cubic-S (XX)]XX (XX)进入总是禁止空间。(代码 XX)
E9408	[Cubic-S (XX)]XX (XX)从选择许可空间离开。(代码 XX)
E9409	[Cubic-S (XX)]XX (XX)进入选择禁止空间。(代码 XX)
E9410	[Cubic-S (XX)]第 XX 轴在总是轴监视范围外。(代码 XX)
E9411	[Cubic-S (XX)]第 XX 轴在选择轴监视范围外。(代码 XX)
E9412	[Cubic-S (XX)]XX (XX)的速度超过 250 mm/s。(代码 XX)
E9413	[Cubic-S (XX)]XX (XX)的速度超过设定速度。(代码 XX)
E9414	[Cubic-S (XX)]在停止监视中，第%-M 轴动作。(代码 XX)

E9415	[Cubic-S (XX)] X 的方向超出工具方向范围(XX)。(代码 XX)
E9416	[Cubic-S (XX)]保护停止中, 减速监视异常(第%-M 轴)。(代码 XX)
E9417	[Cubic-S (XX)]紧急停止中, 减速监视异常(第%-M 轴)。(代码 XX)
E9418	[Cubic-S (XX)]第 XX 轴偏差异常(代码 XX)。在示教模式下, 确认零位和原点位姿。
E9419	[Cubic-S (XX)]第 XX 轴的编码器值异常(代码 XX)。在示教模式下, 确认零位和原点位姿。
E9420	[Cubic-S (XX)]当控制器电源为开时, 第 XX 轴的指令值异常(代码 XX)。在示教模式下, 确认零位和原点位姿。
E9421	[Cubic-S (XX)]第 XX 轴的编码器值突然更改(代码 XX)。在示教模式下, 确认零位和原点位姿。
E9422	[Cubic-S (XX)]安全单元内的软件处理异常。(代码 XX XX)
E9423	[Cubic-S (XX)]安全单元异常。(代码 XX XX)
E9424	[Cubic-S (XX)]工具编号不一致(指令 XX 当前 XX)(代码 XX)
E9425	[Cubic-S (XX)]内部紧急停止继电器粘连。(代码 XX)
E9426	[Cubic-S (XX)]外部紧急停止继电器粘连。(代码 XX)
E9427	[Cubic-S (XX)]用于紧急停止继电器粘连检测的电路或配线异常。(代码 XX)
E9428	[Cubic-S (XX)]外部安全继电器(用户输出 XX)粘连。(代码 XX XX)
E9429	[Cubic-S (XX)]外部安全继电器(固定输出 XX)粘连。(代码 XX XX)
E9430	[Cubic-S (XX)]用于驱动紧急停止继电器的电路或配线异常。(代码 XX XX)
E9431	[Cubic-S (XX)]安全输出(用户输出 XX)的电路或配线异常。(代码 XX XX)
E9432	[Cubic-S (XX)]安全输出(固定输出 XX)的电路或配线异常。(代码 XX XX)
E9433	[Cubic-S (XX)]安全输入(用户输入 XX)的电路或配线异常。(代码 XX XX)
E9434	[Cubic-S (XX)]安全输入(固定输入 XX)的电路或配线异常。(代码 XX XX)
E9435	[Cubic-S (XX)]电源电压下降。(代码 XX)
E9436	[Cubic-S (XX)]与伺服板 XX 的通信异常。(代码 XX XX)(类型: %d)
E9437	[Cubic-S (XX)]第 XX 轴的编码器通信异常。(代码 XX XX)
E9438	[Cubic-S (XX)]与设定参数用的 PC 通信异常。(代码 XX XX)
E9439	[Cubic-S (XX)]第%轴的编码器异常。(代码 XX XX)
E9440	[Cubic-S (XX)]超越控制开关保持为开的状态。(代码 XX)
E9441	[Cubic-S (XX)]复位输入信号保持为开的状态。(代码 XX)
E9442	[Cubic-S (XX)]PC 中的参数值被改写。(代码 XX)
E9443	[Cubic-S]指定的工具数据不同于工具形状编号的工具数据。
E9445	[Cubic-S (XX)]安全输入(固定输入 XX)的状态不一致。(代码 XX XX)
E9446	[Cubic-S (XX)]用户安全输入电路异常或配线的接地故障。(代码 XX XX)
E9447	[Cubic-S (XX)]固定安全输入电路异常或配线的接地故障。(代码 XX XX)
E9448	[Cubic-S]无来自 Cubic-S 的应答。
E9449	[Cubic-S (XX)]第%-M 轴的速度超过 250 mm/s。(代码 XX)

E9450	[Cubic-S (XX)]第%-M 轴的速度超过设定速度。(代码 XX)
E9451	[Cubic-S (XX)]参数编号%d 的值在范围外。(代码 XX XX)
E9452	[Cubic-S (XX)]外部安全继电器(示教/快速检查模式固定输出)粘连。(代码 XX XX)
E9453	[Cubic-S (XX)]外部安全继电器(示教速度监视固定输出)粘连。(代码 XX XX)
E9454	[Cubic-S (XX)]外部安全继电器(超越控制固定输出)粘连。(代码 XX XX)
E9455	[Cubic-S (XX)]外部安全继电器(紧急停止固定输出)粘连。(代码 XX XX)
E9456	[Cubic-S (XX)]安全输出(示教/快速检查模式固定输出)的电路或配线异常。 (代码 XX XX)
E9457	[Cubic-S (XX)]安全输出(示教速度监视固定输出)的电路或配线异常。(代码 XX XX)
E9458	[Cubic-S (XX)]安全输出(超越控制固定输出)的电路或配线异常。(代码 XX XX)
E9459	[Cubic-S (XX)]安全输出(紧急停止固定输出)的电路或配线异常。(代码 XX XX)
E9460	[Cubic-S (XX)]安全输入(示教/快速检查模式固定输入)的电路或配线异常。 (代码 XX XX)
E9461	[Cubic-S (XX)]安全输入(示教速度监视固定输入)的电路或配线异常。(代码 XX XX)
E9462	[Cubic-S (XX)]安全输入(超越控制固定输入)的电路或配线异常。(代码 XX XX)
E9463	[Cubic-S (XX)]安全输入(紧急停止固定输入)的电路或配线异常。(代码 XX XX)
E9464	[Cubic-S (XX)]安全输入(示教/快速检查模式固定输入)的状态不一致。(代码 XX XX)
E9465	[Cubic-S (XX)]安全输入(示教速度监视固定输入)的状态不一致。(代码 XX XX)
E9466	[Cubic-S (XX)]安全输入(超越控制固定输入)的状态不一致。(代码 XX XX)
E9467	[Cubic-S (XX)]安全输入(紧急停止固定输入)的状态不一致。(代码 XX XX)
D0001	CPU 错误。PC=XX
D0002	主 CPU BUS 错误。PC=XX
D0007	[伺服板 XX]CPU 错误。(编码 XX)
D0008	[伺服板 XX]浮动小数点除外。(编码 XX)
D0009	[伺服板 XX]CPU 除外。(PC=XX)
D0010	[手臂接口板]CPU 错误。(PC=XX)
D0900	示教数据已损坏。
D0901	AS 闪存校验和错误。
D0902	伺服闪存校验和错误。
D0903	IP 板存储器错误。XX
D0904	因为 AC_FAIL, 存储器被锁住。
D1003	伺服控制软件初始化错误。
D1011	As 或伺服软件与机器人型号不兼容。
D1012	伺服类型不匹配, 请检查设置。
D1014	伺服系统失误。代码=XX
D1015	伺服数据文件不存在。
D1016	伺服数据文件中没有可用于机器人型号的数据。

D1017	伺服数据下载错误。
D1018	伺服软件版本不匹配。
D1020	[手臂控制板]两 CPU 之间同步错误。
D1021	伺服 FPGA 配置数据未找到。
D1022	伺服 FPGA 软件的配置错误。(代码 XX)
D1023	硬件和软件之间的反馈电流选择信号不一致。(轴 XX)
D1025	[伺服板 XX] 检测到看门狗错误。(伺服 FPGA)
D1026	[伺服板 XX] 从电源程序板电源异常信号输入。
D1027	[MCXX]检测到看门狗错误。
D1028	[伺服板 XX]检测到 DC 电源异常。(伺服 FPGA)
D1029	[伺服板 XX]检测到 AC 外部电源异常。(伺服 FPGA)
D1030	不能与伺服板 XX 通信。
D1031	伺服软件读取错误。
D1032	[伺服板 XX]伺服软件下载错误。(代码 XX)
D1033	连接端编号 (XX) 与伺服板编号 (XX) 的设定不同。
D1034	伺服数据文件未找到。(代码 XX)
D1035	[伺服板 XX]伺服软件初始化错误。(代码 XX)
D1036	[伺服板 XX]伺服数据下载错误。(代码 XX)
D1037	[伺服板 XX]伺服 FPGA 配置错误。(代码 XX)
D1038	[伺服板 XX]伺服软件初始化数据上载错误。(代码 XX)
D1039	[伺服板 XX]伺服软件初始化数据下载错误。(代码 XX)
D1040	[伺服板 XX]设备检查错误。(代码 XX)
D1041	第 XX 轴刹车解除电路异常。
D1042	伺服 FPGA2 的配置数据未找到。
D1043	伺服 FPGA2 的配置错误。(代码 XX)
D1500	轴 XX 编码器读取错误。
D1501	焊枪交换器连接不良或编码器通信失误。
D1502	轴 XX 放大器过流。
D1504	轴 XX 电流反馈异常。(放大器失误, 动力线缆断开)
D1507	AC 初级电源关断。
D1512	轴 XX 的制动器线失误。
D1516	保持备份时间的软/硬件设置不一致。
D1517	安全电路中紧停线路的保险线烧断。
D1518	安全电路上的紧停状态不一致。
D1520	安全电路中的示教/再现状态不一致。
D1521	安全电路中安全护栏状态不一致。
D1522	安全电路中启动设备的条件不一致。

D1523	安全电路中启动外部设备的条件不一致。
D1528	控制器温度超限。
D1529	信号线缆未连接或编码器电源失误。
D1530	轴 XX 电流极限值异常。
D1532	(SSCNET) 编码器通信失误。轴 XX 代码=XX
D1533	(SSCNET) 轴 XX 的绝对位置值已被擦除。代码=XX
D1534	(SSCNET) 轴 XX 的参数错误。代码=XX
D1535	(SSCNET) 轴 XX 报警。代码=XX
D1536	轴 XX 不能正常动作。
D1539	PWM 信号输出电路出现故障。
D1541	软件和手臂控制板之间设置的编码器型类不一致。
D1542	多点连接编码器初始化时, 旋转量数据获取失败。
D1543	[伺服板 XX]DC 电源异常。(5 V)
D1544	[伺服板 XX]DC 电源异常。(3.3 V)
D1545	[伺服板 XX]DC 电源异常。(12 V)
D1546	[伺服板 XX]DC 电源异常。(2.5 V)
D1547	[伺服板 XX]DC 电源异常。(1.2 V)
D1548	[伺服板 XX]DC 电源异常。(1.0 V)
D1549	[伺服板 XX]外部电源电压过低。
D1550	[伺服板 XX]外部电源电压过高。
D1551	[伺服板 XX]AC 外部电源切断。
D1552	[MCXX]DC 电源电压异常。(3.3 V)
D1553	[MCXX]DC 电源电压异常。(5 V)
D1557	[电源程序板]DC 电源异常(3.3 V)。
D1558	[电源程序板]DC 电源异常(5 V)。
D1559	[电源程序板]DC 电源异常(12 V)。
D1560	[电源程序板]DC 电源异常(24 V)。
D1561	[电源程序板]AC 外部电源切断。
D1562	[电源程序板]AC 外部电源电压过高。
D1563	[电源程序板]AC 外部电源电压过低。
D1564	[电源程序板]检测到远程电源 OFF 信号。
D1565	不能访问电源程序板。(代码 XX)
D1566	P-N 电容器未放电。(伺服板 XX) (MCXX)
D1567	[伺服板 XX]外部电源电压错误。
D1568	[伺服板 XX]PWM 输出电源电路故障。
D1571	第 XX 轴的电流反馈电路异常。
D2000	激光传感器的通信板没有响应。

D2001	RI/O 或 C-NET 板初始化错误。
D2002	手臂 ID 板没有响应。
D2003	手臂 ID 板中没有数据。
D2004	手臂 ID 板中的数据不匹配。
D2005	CC-LINK 软件版本不匹配。
D2006	防爆示教器通信板上的看门狗错误。
D2007	内置式 PLC 板没有响应。
D2008	XX 组的 PN 电磁接触器已粘连。
D2009	内部压力传感器失误。
D2010	用户接口和手臂控制板间同步出错。
D2011	用户接口和手臂控制板间参数下载错误。
D2012	软吸收错误，关断后再开启控制电源。
D2013	修改增益错误，关断后再开启控制电源。
D2014	机器人网络初始化错误。
D2015	组 XX（第 XX 轴）的 MC 粘连。
D2016	手臂控制板没有响应。
D2021	手臂数据文件未找到。
D2022	手臂数据未找到。
D2023	手臂数据载入失败。
D2024	[手臂控制板] 机器人型号设置失败。
D2025	软件中设置的机器人代码与手臂控制板上的不匹配。
D2026	软件中设置的代码和电流传感器接口板中的不匹配。
D2027	软件中设置的代码与动力模块中的不匹配。
D2028	(SSCNET) 初始化错误。代码=XX
D2035	程序执行错误。
D2036	(SSCNET) 1LP 接口板系统失误。代码=XX
D2037	安全单元回路异常。
D2038	(SSCNET) 接口板未安装。
D2039	(SSCNET) 轴 XX 初始化通信失误。
D2040	(SSCNET) 轴 XX 初始化错误。代码=XX
D2041	信号线缆连接失误。
D2042	伺服放大器和机器人手臂之间连接错误。
D2043	手臂接口板检测 AC-Fail。
D2044	[手臂控制板] 伺服 FPGA 软件没有响应。
D2045	[手臂控制板] 设备检查错误。(代码 XX)
D2046	吹扫控制板上的继电器异常。(继电器)
D2047	伺服 CPU 板上的跳线设定错误或安全继电器异常。

D2048	吹扫控制板上的 12 V 电源异常。
D2049	吹扫控制板上的联锁继电器驱动回路 (1) 过电流。
D2050	吹扫控制板上的联锁继电器驱动回路 (2) 过电流。
D2051	吹扫控制板通信异常。
D2052	外部轴放大器的硬件设置不匹配。机器人 = n
D2053	(FANXX) 风扇旋转速度异常。(伺服板 XX)
D2054	在软件和电源模块之间设置的代码不一致。(代码:XX)
D2055	[电源程序板]检测到看门狗错误。
D2056	[I/O 板 (NoXX)] 同一地址 ID 设置到有些板中。
D2057	[伺服板 XX] 伺服 FPGA 软件没有响应。
D2058	[主 CPU 板]DC 电源异常。(XX mV)
D2064	[吹扫控制板]内压低。
D2065	当内压低时隔绝刹车的安全继电器异常。
D2066	[吹扫控制板]DC 电源异常。(12 V)
D2067	[主 CPU 板]与吹扫控制板的通信异常。
D2068	[IO 板编号, XX]设备检查失败。(代码 XX)
D2069	[ANYBUS 接口板 (编号, XX)]几个板有相同的 ID 地址。
D3800	通信板存储器异常。(XX)
D3801	第 XX 轴放大器接口异常 1。
D3804	第 XX 轴放大器执行元件异常。
D3805	第 XX 轴放大器电流检测器异常。
D3806	第 XX 轴放大器主回路电压不一致。
D3807	第 XX 轴放大器存储器 (EEPROM) 异常。
D3808	第 XX 轴放大器内部 RAM 异常。
D3809	第 XX 轴放大器伺服处理器异常。
D3810	第 XX 轴放大器参数异常。
D3811	第 XX 轴放大器初始处理异常。
D3812	第 XX 轴放大器未定义错误 1。
D3813	放大器通信接口板的初始检查错误。(XX)
D3814	放大器通信接口板的未定义错误。(XX)
D3815	第 XX 轴放大器不能通信。
D3816	第 XX 轴放大器通信帧接收错误。
D3817	第 XX 轴放大器通信帧接收超时。
D3818	第 XX 轴放大器通信 BANK DATA 异常。
D3819	第 XX 轴放大器初始化超时。
D3820	第 XX 轴放大器通信未定义错误。
D3821	马达线束连接处异常。



D3822	第 XX 轴马达设定与控制器不一致。
D3823	控制器内的第 XX 风扇故障。
D3824	第 1I/O 板的第 XX 熔丝已开路。
D3825	第 2I/O 板的熔丝编号 XX 已开路。
D3826	机器人 DC 电源异常。
D3828	控制器型号错误。
D3830	P-N 间过电压。
D3831	P-N 间电压低。
D3832	再生吸收时间异常。
D3833	放电电阻过热。
D3834	电源板开关回路异常。
D3835	电源板突入电流防止回路异常。
D3836	DC 电源电压异常。(代码 XX)
D3837	第 XX 轴放大器的控制电源电压异常。
D3838	电源板异常。
D3840	电源板上的第 XX 风扇故障。
D3841	机器人 XX 伺服放大器没有安装。
D3842	电源装置的控制电源异常。
D3843	刹车释放设定异常。
D3844	第 XX 轴放大器通信中, PLL 接收故障。
D3845	第 XX 轴放大器的曼彻斯特代码错误。
D4000	[DIAG]在 RS232C 中检测出错误。(代码: XX)
D4001	[DIAG]在以太网中检测出错误。(代码: XX)
D4500	未发现现场总线接口板。
D4501	ABMA-PDP) 接口模块失误。代码=XX
D4502	FIELD-BUS-INIT) 错误应答。代码=XX
D4503	FIELD-BUS-INIT) 应答超时。代码=XX
D4504	ANYBUS) OUT/FB. CTRL 请求超时。代码=XX
D6000	Barrier 单元内温度过高。
D6001	相互等待数据初始化失败。
D6002	[喷涂设备控制软件]检测到看门狗错误。

## 附录 5 AS 语言一览表（按字母为序）

AS 语言的缩写可能无预先通知而改变。

功能后的字母为：

M: 监控指令, E: 编辑器指令, P: 程序命令, S: 开关, F: 函数, O: 运算符,  
K: 其他关键字

名称	缩写	功能		格式(参数)	节	页
ABORT	AB	当前步骤完成后停止执行。	M	ABORT	5.4	5-39
ABOVE	AB	改变形态, 使肘关节处于上部位姿。	P	ABOVE	6.4	6-40
ABS	ABS	返回数学表达式的绝对值	F	ABS(实数值)	9.3	9-47
ABS. SPEED	ABS. SPEED	允许绝对速度的使用。(选项)	S	... ABS. SPEED...	7.0	7-20
ACCEL	ACCE	设定加速度。	P	ACCEL 加速度 ALWAYS	6.2	6-19
ACCURACY	ACCU	设定精度范围。	P	ACCURACY 距离 ALWAYS FINE	6.2	6-18
AFTER. WAIT. TMR	AF	设置点焊控制器(timer)在一体化命令程序中如何开始。(选项)	S	... AFTER. WAIT. TMR...	7.0	7-21
ALIGN	AL	将工具坐标系的 Z 轴和基础坐标系的轴对齐。	P	ALIGN	6.1	6-10
AND	AND	逻辑与(Logical AND)	O	.... AND ....	8.3	8-3
ASC	ASC	返回 ASCII 码值。	F	ASC(字符串, 字符编号)	9.1	9-11
ATAN2	ATAN2	返回反正切值	F	ATAN2(实数值 1, 实数值 2)	9.3	9-47
AUTOSTART. PC	AUTOSTART.	启用控制电源开启时的 PC 程序自动启动。(选项)	S	... AUTOSTART. PC...	7.0	7-9
AUTOSTART2. PC	AUTOSTART2.	启用控制电源开启时的 PC 程序自动启动。(选项)	S	... AUTOSTART2. PC...	7.0	7-9
AUTOSTART3. PC	AUTOSTART3.	启用控制电源开启时的 PC 程序自动启动。(选项)	S	... AUTOSTART3. PC...	7.0	7-9
AUTOSTART4. PC	AUTOSTART4.	启用控制电源开启时的 PC 程序自动启动。(选项)	S	... AUTOSTART4. PC...	7.0	7-9
AUTOSTART5. PC	AUTOSTART5.	启用控制电源开启时的 PC 程序自动启动。(选项)	S	... AUTOSTART5. PC...	7.0	7-9
AVE_TRANS	AVE TRANS	返回两个位姿的变换值的平均值。	F	AVE_TRANS(变换值变量 1, 变换值变量 2)	9.2	9-41
BAND	BAND	二进制与(Binary AND)	O	... BAND ...	8.4	8-5
BASE	BA	改变基础坐标系变换值。	M	BASE 变换值变量	5.6	5-56
BASE	BA	改变基础坐标系变换值。	P	BASE 变换值变量	6.9	6-103
BASE	BASE	改变基础坐标系变换值。	F	BASE	9.2	9-42
BELOW	BE	改变形态, 使肘关节处于下部位姿。	P	BELOW	6.4	6-40
BITS	BI	设定信号等于指定值(至多 16 个信号)。	M	BITS 开始信号编号, 信号数值	5.7	5-92
BITS	BI	设定信号等于指定值(至多 16 个信号)。	P	BITS 起始信号编号, 信号的数量=十进制数值	6.7	6-72
BITS	BITS	设定信号等于指定值(至多 16 个信号)。	F	BITS (起始信号编号, 信号数量)	9.1	9-5

名称	缩写	功能		格式(参数)	节	页
BITS32	BITS32	设定信号等于指定值(至多 32 个信号)。	M	BITS32 开始信号编号, 信号数=值	5.7	5-94
BITS32	BITS32	设定信号等于指定值(至多 32 个信号)。	P	BITS 32 起始信号编号, 信号的数量=数值	6.7	6-73
BITS32	BITS32	设定信号等于指定值(至多 32 个信号)。	F	BITS32 (起始信号编号, 信号数量)	9.1	9-6
BOR	BOR	二进制或(Binary OR)	O	... BOR ...	8.4	8-5
BRAKE	BRA	停止当前动作并跳转到下一步骤。	P	BRAKE	6.2	6-20
BREAK	BRE	暂停下一步骤的执行, 直到当前动作完成。	P	BREAK	6.2	6-20
BSPEED	BSP	设定一体化命令程序的速度(选项)。	P	BSPEED 速度	6.2	6-21
BXOR	BX	二进制异或(Binary XOR)	O	... BXOR ...	8.4	8-5
BY	BY	返回将参数“变换值”描述的位姿沿着各基础坐标轴(X、Y、Z)平移指定的距离而得到的位姿的变换值。	K	SHIFT(变换值 BY X 平移量, Y 平移量, Z 平移量)	9.2	9-40
C	C	结束当前程序的编辑, 切换到另一个程序。	E	C 程序名, 步骤编号	5.1	5-4
C1MOVE	C1	以圆弧插补动作移动(选项)。	P	C1MOVE 位姿变量, 夹紧编号	6.1	6-13
C2MOVE	C2	以圆弧插补动作移动(选项)。	P	C2MOVE 位姿变量, 夹紧编号	6.1	6-13
CALL	CA	分支到一个子程序。	P	CALL 程序名	6.5	6-45
CASE	CASE	根据特定的情形编号执行程序。	P	CASE 索引变量 OF ... VALUE ... ANY... END	6.6	6-64
CCENTER	CCENTER	返回由给定位姿描述的圆的圆心变换值(选项)。	F	CCENTER(变换值变量 1, 变换值变量 2, 变换值变量 3, 变换值变量 4)	9.2	9-45
CHECK. HOLD	CH	当 HOLD/RUN(保持/运转)在 HOLD 位置时, 允许或禁止从键盘输入指令。	S	.... CHECK. HOLD....	7.0	7-3
\$CHR	\$CHR	返回指定数值的 ASCII 码。	F	\$CHR(实数值)	9.4	9-50
CHSUM	CH	清除校验和错误。	M	CHSUM	5.6	5-82
CLAMP	CLAMP	控制夹紧信号的打开或闭合。	P	CLAMP 夹紧编号 1, ..., 夹紧编号 8	6.7	6-85
CLOSE	CLOSE	在下一运动命令开始时, 输出夹紧闭合信号。	P	CLOSE 夹紧编号	6.3	6-34
CLOSEI	CLOSEI	在当前运动命令完成时, 输出夹紧闭合信号。	P	CLOSEI 夹紧编号	6.3	6-34
CLOSES	CLOSES	在运动命令执行过程中, 输出夹紧闭合信号。	P	CLOSES 夹紧编号	6.3	6-36
COM	COM	二进制非(Binary complement)	O	... COM ...	8.4	8-5
CONTINUE	CON	继续程序的执行。	M	CONTINUE NEXT	5.4	5-40
COPY	COP	复制程序。	M	COPY 新程序名=源程序名+源程序名+...	5.2	5-22
COS	COS	返回余弦值	F	COS(实数值)	9.3	9-47
CP	CP	启用或关闭连续路径(CP)功能。	S	..... CP.....	7.0	7-3

名称	缩写	功能		格式(参数)	节	页
CS	CS	显示 CYCLE START (循环启动) 的 ON/OFF 状态。	S	SWITCH(CS)	7.0	7-11
CSHIFT	CSHIFT	返回朝着由给定位姿描述的圆的圆心位置平移而得到的位姿的变换值(选项)。	F	CSHIFT(变换值变量 1, 变换值变量 2, 变换值变量 3, 变换值变量 4 BY 平移量)	9.2	9-45
CURLIM		更改外部轴的马达电流限制值。	P	CURLIM 轴编号, 正电流限制值, 负电流限制值	6.9	6-112
CYCLE. STOP	CY	用外部 HOLD 停止循环。	S	... CYCLE. STOP. ...	7.0	7-4
D	D	删除程序步骤。(D: Delete, 删除)	E	D 步骤数量	5.1	5-7
\$DATE	\$DATE	返回系统的日期。	F	\$DATE (日期形式)	9.4	9-58
DECEL	DECE	设置减速度。	P	DECEL 减速度 ALWAYS	6.2	6-19
\$DECODE	\$DECODE	提取指定的分隔符前的字符。	F	\$DECODE (字符串变量, 分隔符, 模式)	9.4	9-53
DECOMPOSE	DECO	把位姿信息的组成部分赋值给数组变量的元素。	P	DECOM 数组变量 [元素编号] = 位姿变量	6.9	6-102
DEFSIG	DEF	显示和设置软件专用信号。	M	DEFSIG OUTPUT/INPUT	5.6	5-69
DELAY	DEL	停止机器人运动指定长度的时间。	P	DELAY 时间	6.1	6-4
DELETE	DEL	删除机器人存储器中的程序和变量。	M	DELETE (/P) (/L) (/R) (/S) 数据, ...	5.2	5-18
DELETE	DEL	删除机器人存储器中的程序和变量。	P	DELETE (/P) (/L) (/R) (/S) 数据, ...	6.10	6-114
#DEST	#DEST	以关节位移值的形式返回目标位姿。	F	#DEST	9.2	9-31
DEST	DEST	以变换值的形式返回目标位姿。	F	DEST	9.2	9-31
DEST_CIRINT	DEST_CIRINT	决定 DEST/#DEST 函数返回的位姿。	S	DEST_CIRINT	7.0	7-24
DEXT	DEXT	返回一个位姿的指定分量(选项)。	F	DEXT(位姿变量名称, 元素编号)	9.1	9-10
DISPIO_01	DIS	改变 IO 指令的显示模式。	S	... DISPIO_01. ...	7.0	7-17
DISTANCE	DISTANCE	返回两个点之间的距离。	F	DISTANCE(变换值变量 1, 变换值变量 2)	9.1	9-8
DLYSIG	DL	在指定的时间过去后开启输出信号(变为 ON)。	M	DLYSIG 信号编号, 时间	5.7	5-91
DLYSIG	DL	在指定的时间过去后开启输出信号(变为 ON)。	P	DLYSIG 信号编号 时间	6.7	6-70
DO	DO	执行单条程序命令。	M	DO 程序命令	5.4	5-42
DO	DO	DO 循环。	P	DO 程序命令 UNTIL 逻辑表达式	6.6	6-60
DRAW	DRA	沿基础坐标系 X, Y, Z 轴移动指定的量。	P	DRAW X 平移量, Y 平移量, Z 平移量, X 旋转量, Y 旋转量, Z 旋转量, 速度	6.1	6-9
DRIVE	DRI	以单根轴的方向移动。	P	DRIVE 关节编号, 位移, 速度	6.1	6-8
DWRIST	DW	改变形态,使 JT5 的角度为负值。	P	DWRIST	6.4	6-41
DX	DX	返回变换值中的位移值(X)。	F	DX(变换值变量)	9.1	9-9
DY	DY	返回变换值中的位移值(Y)。	F	DY(变换值变量)	9.1	9-9
DZ	DZ	返回变换值中的位移值(Z)。	F	DZ(变换值变量)	9.1	9-9

名称	缩写	功能		格式(参数)	节	页
E	E	退出编辑器。(E: Exit, 退出)	E	E	5.1	5-10
EDIT	ED	启动程序编辑器。	M	EDIT 程序名, 步骤编号	5.1	5-3
ELSE	EL	根据逻辑表达式的结果执行一组程序步骤。	P	IF ... THEN ... ELSE ... END	6.6	6-56
ENA_TOOLSHAPE	ENA_TOOLSHAPE	根据工具形状设定速度控制的有效/无效。	M	ENA_TOOLSHAPE 工具形状编号=TRUE/FALSE	5.6	5-60
ENA_TOOLSHAPE	ENA_TOOLSHAPE	根据工具形状设定速度控制的有效/无效。	P	ENA_TOOLSHAPE 工具形状编号=TRUE/FALSE	6.9	6-105
ENCCHK_EMG	ENCCHK_E	设置紧急停止时机器人的位姿与机器人重启时的位姿之间的偏差检查的容许范围。	M	ENCCHK_EMG	5.6	5-79
ENCCHK_PON	ENCCHK_P	设置电源开启时编码器值与上次电源关闭时值之间差异的可接受范围。	M	ENCCHK_PON	5.6	5-79
\$ENCODE	\$ENCODE	返回由打印数据产生的字符串。	F	\$ENCODE (打印数据, 打印数据, .....	9.4	9-55
FOR.....END	EN	根据特定的情形编号执行程序。	P	FOR ... TO ... STEP ... END	6.6	6-62
ENV DATA	ENV	设置自动伺服关断计时和示教器连接/断开。	M	ENV_DATA	5.6	5-81
ENV2 DATA	ENV2	设定终端连接/断开。	M	ENV2_DATA	5.6	5-81
ERESET	ERE	复位出错状态。	M	ERESET	5.6	5-75
ERRLOG	ERR	显示出错情况的历史。	M	ERRLOG	5.6	5-63
ERROR	ERROR	显示程序是否在错误状态。	S	SWITCH(ERROR)	7.0	7-13
ERROR	ERROR	显示程序是否在错误状态。	F	ERROR	9.1	9-20
\$ERROR	\$ERROR	显示程序是否在错误状态。	F	\$ERROR (错误编号)	9.4	9-57
\$ERRORS	\$ERRORS	返回指定错误代码的出错信息。	F	\$ERRORS (错误代码)	9.4	9-57
ERRSTART.PC	ERRS	决定出错时是否执行选定的 PC 程序。	S	... ERRSTART.PC...	7.0	7-10
EXECUTE	EX	执行程序。	M	EXECUTE 程序名, 执行循环数, 步骤编号	5.4	5-37
EXISTCHAR	EXISTCHAR	返回指定的字符串变量是否存在。	F	EXISTCHAR("字符串变量名称")	9.1	9-27
EXISTINTEGER	EXISTINTEGER	返回指定的整数值变量是否存在。	F	EXISTINTEGER("整型变量名称")	9.1	9-27
EXISTJOINT	EXISTJOINT	返回指定的关节位移值变量是否存在。	F	EXISTJOINT("关节位移值变量名称")	9.1	9-25
EXISTPGM	EXISTPGM	返回指定的程序是否存在。	F	EXISTPGM("程序名称")	9.1	9-28
EXISTREAL	EXISTREAL	返回指定的实数值变量是否存在。	F	EXISTREAL("实型变量名称")	9.1	9-26
EXISTTRANS	EXISTTRANS	返回指定的变换值变量是否存在。	F	EXISTTRANS("变换值变量名称")	9.1	9-26
EXTCALL	EX	调用由外部信号选定的程序。	P	EXTCALL	6.7	6-75
F	F	搜索字符。(F: Find, 查找)	E	F 字符串	5.1	5-7
FFRESET	FFRESET	复位机器人速度/加速度前馈增益。	P	FFRESET	6.2	6-31

名称	缩写	功能		格式(参数)	节	页
FFSET	FFSET	设定速度/加速度前馈增益。	P	FFSET JT1 增益, JT2 增益, JT3 增益, JT4 增益, JT5 增益, JT6 增益, JT7 增益, JT8 增益, JT9 增益	6.2	6-28
FFSET_STATUS	FFSET_STATUS	设定机器人速度/加速度前馈增益。	M	FFSET_STATUS	5.6	5-87
FLOWRATE	FLOWRATE	从流量控制模式切换到速度输出模式(反之亦然)。(选项)	S	... FLOWRATE ...	7.0	7-19
FOR	FOR	重复程序执行。	P	FOR 循环变量=起始值 TO 结束值 STEP 步进值程序命令 END	6.6	6-62
FRAME	FRAME	返回框架坐标系的变换值。	F	FRAME (变换值变量 1, 变换值变量 2, 变换值变量 3, 变换值变量 4)	9.2	9-32
FREE	FR	显示空余存储器的数量。	M	FREE	5.6	5-53
GOTO	GO	跳转到指定的标签处。	P	GOTO 标签 IF 条件	6.5	6-43
GUNOFF	GUNOFF	关断射枪信号(变为 OFF), 用距离控制射枪输出延时。	P	GUNOFF 射枪编号, 距离	6.3	6-37
GUNOFFTIMER	GUNOFFTIMER	用计时控制射枪输出 OFF 延时。(选项)	P	GUNOFFTIMER 射枪编号, 时间	6.3	6-38
GUNON	GUNON	开启射枪信号(变为 ON), 用距离控制射枪输出延时。	P	GUNON 射枪编号, 距离	6.3	6-37
GUNONTIMER	GUNONTIMER	用计时控制射枪输出 ON 延时。(选项)	P	GUNONTIMER 射枪编号, 时间	6.3	6-38
HALT	HA	停止程序执行(不可恢复)。	P	HALT	6.5	6-51
HELP	HEL	显示 AS 语言命令/指令列表。	M	HELP	5.6	5-76
HELP/DO	HEL/DO	列出可以和 DO 指令一起使用的程序命令。	M	HELP/DO	5.6	5-76
HELP/F	HEL/F	列出函数。	M	HELP/F	5.6	5-76
HELP/M	HEL/M	列出监控指令。	M	HELP/M	5.6	5-76
HELP/MC	HEL/MC	列出可以和 MC 命令一起使用的监控指令。	M	HELP/MC	5.6	5-76
HELP/P	HEL/P	列出程序命令。	M	HELP/P	5.6	5-76
HELP/PPC	HEL/PPC	列出可以在 PC 程序中使用的程序命令。	M	HELP/PPC	5.6	5-76
HELP/SW	HEL/SW	显示系统开关列表。	M	HELP/SW	5.6	5-76
HERE	HE	以当前位姿定义一个位姿变量。	M	HERE 位姿变量	5.5	5-44
HERE	HE	以当前位姿定义一个位姿变量。	P	HERE 位姿变量	6.9	6-99
#HERE	#HERE	返回当前位姿的关节位移值。	F	#HERE	9.2	9-36
HERE	HERE	返回当前位姿的变换值。	F	HERE	9.2	9-36
HMOVE	HM	以直线插补动作移动(腕关节以关节插补动作移动)。	P	HMOVE 位姿变量, 夹紧编号	6.1	6-10
HOLD	HO	停止执行	M	HOLD	5.4	5-39
HOLD. STEP	HOLD. STEP	允许程序暂停时的执行中步骤的显示。(选项)	S	... HOLD. STEP. ...	7.0	7-18
HOME	HO	移动到原点位置(home position)。	P	HOME 原点位姿编号	6.1	6-7
#HOME	#HOME	返回原点(HOME)位置的关节位移值(选项)。	F	#HOME (原点位姿编号)	9.2	9-44



名称	缩写	功能		格式(参数)	节	页
HSENSE	HSENSE	读由 HSENSESET 存储的数据。	M	HSENSE 监视编号 结果变量, 信号状态变, 位姿变量, 错误变量, 内存使用量变量	5.7	5-102
HSENSE	HSENSE	读由 HSENSESET 存储的数据。	P	HSENSE 监视编号 结果变量, 信号状态变, 位姿变量, 错误变量, 内存使用量变量	6.7	6-87
HSENSESET	HSENSESET	开始指定传感器信号的监控。	M	HSENSESET 监视编号=输入信号编号, 输出信号编号, 信号输出耽误的时间	5.7	5-101
HSENSESET	HSENSESET	开始指定传感器信号的监控。	P	HSENSESET 监视编号 = 输入信号编号, 输出信号编号, 信号输出耽误的时间	6.7	6-86
HSETCLAMP	HS	设置默认的夹紧规格。	M	HSETCLAMP	5.6	5-67
I	I	插入一个新的步骤。(I: Insert, 插入)	E	I	5.1	5-6
ID	ID	显示软件的版本信息。	M	ID	5.6	5-77
IF	IF	当指定的条件满足时, 跳转到指定的标签处。	P	IF 条件 GOTO 标签	6.5	6-44
IF	IF	当指定的条件满足时, 跳转到指定的标签处。	P	IF... THEN... ELSE... END	6.6	6-56
IFPLABEL	IFPLABEL	设定接口控制板上的标签。	M	IFPLABEL 位置, “标签 1”, “标签 2”, “标签 3”, “标签 4	5.8	5-108
IFPLABEL	IFPLABEL	设定接口控制板上的标签。	P	IFPLABEL 位置, “标签 1”, “标签 2”, “标签 3”, “标签 4	6.8	6-94
IFPTITLE	IFPTITLE	设定接口控制板上的标题。	M	IFPTITLE 页码., “标题”	5.8	5-109
IFPTITLE	IFPTITLE	设定接口控制板上的标题。	P	IFPTITLE 页码., “标题”	6.8	6-95
IFWPRINT	IFWPRINT	在显示窗口中显示指定的字符串。	M	IFWPRINT 窗口, 行, 列, 背景颜色, 标签颜色 = “字符串”, 字符串”, ...	5.8	5-107
IFWPRINT	IFWPRINT	在显示窗口中显示指定的字符串。	P	IFWPRINT 窗口, 行, 列, 背景颜色, 标签颜色 = “字符串”, 字符串”, ...	6.8	6-93
IGNORE	IG	取消 ON 或 ONI 命令。	P	IGNORE 信号编号	6.7	6-79
INPUT	I	INPUT(或仅 I) 显示输入信号	K	DEFSIG INPUT	5.6	5-69
INRANGE	INRANGE	检查位姿是否处于运动范围内(选项)。	F	INRANGE(位姿变量 1, 关节位移值变量)	9.1	9-23
INSTR	INSTR	返回指定字符在字符串中的起始点之后所在的位置值。	F	INSTR(开始点, 字符串 1, 字符串 2)	9.1	9-14
INT	INT	返回一个数值的整数值。	F	INT(数值表达式)	9.1	9-16
IO	IO	显示二进制信号的状态。	M	IO/E 信号编号	5.6	5-52
IPEAKCLR	IPEAKCLR	复位峰值电流值。(选项)	M	IPEAKCLR	5.6	5-84
IPEAKLOG	IPEAKLOG	显示峰值电流值。(选项)	M	IPEAKLOG	5.6	5-84
JAPPRO	JA	以关节(各轴)插补方式动作接近目的地。	P	JAPPRO 位姿变量, 距离	6.1	6-5
JDEPART	JD	以关节插补动作离开当前位姿。	P	JDEPART 距离	6.1	6-6

名称	缩写	功能		格式(参数)	节	页
JMOVE	JM	以关节(各轴)插补动作移动机器人。	P	JMOVE 位姿变量名, 夹紧编号	6.1	6-3
KILL	KI	初始化执行堆栈。	M	KILL	5.4	5-41
L	L	选择前一步骤。(L: Last, 最后一行)	E	L	5.1	5-5
LAPPRO	LA	以直线插补动作动作接近目的地。	P	LAPPRO 位姿变量, 距离	6.1	6-5
LDEPART	LD	以直线插补动作离开当前位姿。	P	LDEPART 距离	6.1	6-6
\$LEFT	\$LEFT	返回字符串中最左边的字符。	F	\$LEFT(字符串, 字符的数量)	9.4	9-51
LEFTY	LE	改变形态, 使机器人手臂像人的左手臂。	P	LEFTY	6.4	6-40
LEN	LEN	返回字符串的长度。	F	LEN (字符串)	9.1	9-11
LIST	LI	显示所有程序步骤和变量值。	M	LIST (/P) (/L) (/R) (/S) 程序/数据, ...	5.2	5-17
LLIMIT	LL	设定机器人运动的下限。	M	LLIMIT 关节位移值变量	5.6	5-55
LLIMIT	LL	设定机器人运动的下限。	P	LLIMIT 关节位移值变量	6.9	6-107
LMOVE	LM	以直线插补动作移动机器人。	P	LMOVE 位姿变量名, 夹紧编号	6.1	6-3
LOAD	LO	装载程序和数据到机器人存储器。	M	LOAD/Q 文件名	5.3	5-31
LOCK	LO	改变机器人控制程序的优先级。	P	LOCK 优先级	6.5	6-50
LSTRACE	LSTRACE	显示日志数据。(选项)	M	LSTRACE stepper 编号: 日志编号	5.2	5-25
M	M	替换字符。(M: Modify, 修改)	E	M/ /已存在的字符/新字符	5.1	5-8
MAXINDEX	MAXINDEX	返回指定维编号中的最大下标值。	F	MAXINDEX(字符串变量, 维编号)	9.1	9-16
MAXVAL	MAXVAL	比较给定的数值。	F	MAXVAL(实数值 1, 实数值 2, ...)	9.1	9-15
MC	M	从 PC 程序中执行监控指令。	P	MC 监控指令	6.9	6-111
MESSAGES	ME	允许或禁止终端输出。	S	... MESSAGES. ....	7.0	7-4
\$MID	\$MID	返回指定数量的字符。	F	\$MID(字符串, 实数值, 字符的数量)	9.4	9-52
MININDEX	MININDEX	返回指定维编号中的最小下标值。	F	MININDEX(字符串变量, 维编号)	9.1	9-18
MINVAL	MINVAL	比较给定的数值。	F	MINVAL(实数值 1, 实数值 2, ...)	9.1	9-15
MM/MIN	MM/M	MM/MIN	K	... MM/M	6.2	6-16
MM/S	MM/S	MM/S	K	... MM/S	6.2	6-16
MOD	MOD	求余数	O	... MOD ...	8.1	8-1
MSPEED	MSPEED	返回机器人 1 的监控速度(选项)。	F	MSPEED	9.1	9-22
MSPEED2	MSPEED2	返回机器人 2 的监控速度(选项)。	F	MSPEED2	9.1	9-22
MSTEP	MS	执行一条机器人运动命令。	M	MSTEP 程序名, 执行循环数, 步骤编号	5.4	5-38
MVWAIT	MVWAIT	将程序的执行置于待机状态, 直到到达给定的位置或时间。	P	MVWAIT 数值	6.5	6-48
NCHOFF	NCHOF	关闭陷波滤波器 (变成 OFF)。(选项)	P	NCHOFF	6.9	6-109



名称	缩写	功能		格式(参数)	节	页
NCHON	NCHON	开启陷波滤波器 (变成 ON)。(选项)	P	NCHON	6.9	6-109
NEXT	N	下一个步骤执行。	K	CONTINUE NEXT	5.4	5-40
NLOAD	NLOAD	将指定文件读入到机器人存储器中。	P	NLOAD/IF/ARC 设备编号=文件名 + 文件名 + ..., 状态变量	6.10	6-116
NOT	NOT	逻辑非	O	... NOT ...	8.3	8-3
NULL	NULL	返回空变换值。	F	NULL	9.2	9-35
O	O	将光标放在当前步骤上。	E	O	5.1	5-10
OFF	OF	关闭系统开关(变为 OFF)。	M	开关名, ... OFF	5.6	5-65
OFF	OF	关闭系统开关(变为 OFF)。	P	开关名称, ... OFF	6.9	6-108
OFF	OFF	关闭系统开关(变为 OFF)。	F	... OFF. ...	9.1	9-12
ON	ON	开启系统开关(变为 ON)。	M	开关名, .... ON	5.6	5-65
ON	ON	开启系统开关(变为 ON)。	P	ON 模式 信号编号 CALL 程序名, 优先级	6.7	6-77
ON	ON	开启系统开关(变为 ON)。	P	ON 模式 信号编号 GOTO 标签, 优先级	6.7	6-77
ON	ON	开启系统开关(变为 ON)。	P	开关名称, .... ON	6.9	6-108
ON	ON	开启系统开关(变为 ON)。	F	..... ON. ....	9.1	9-12
ONE	ONE	在出错时调用程序。	P	ONE 程序名	6.5	6-53
ONI	ONI	设定中断条件。	P	ONI 模式 信号编号 CALL 程序名, 优先级	6.7	6-77
ONI	ONI	设定中断条件。	P	ONI 模式 信号编号 GOTO 标签, 优先级	6.7	6-77
OPEINFO	OPEINFO	显示操作信息。(选项)	M	OPEINFO 关节编号	5.6	5-85
OPEINFOCLR	OPEINFOCLR	清除操作信息。(选项)	M	OPEINFOCLR	5.6	5-85
OPEN	OPEN	在下一运动命令开始时, 输出夹紧打开信号。	P	OPEN 夹紧编号	6.3	6-33
OPENI	OPENI	在当前运动命令完成时, 输出夹紧打开信号。	P	OPENI 夹紧编号	6.3	6-33
OPENS	OPENS	在运动命令执行过程中, 输出夹紧打开信号。	P	OPENS 夹紧编号	6.3	6-36
OPLOG	OP	显示操作的历史。	M	OPLOG	5.6	5-63
OR	OR	逻辑或(Logical OR)	O	... OR ...	8.3	8-3
OUTDA	OUTDA	按照设定的条件输出电压。	M	OUTDA 电压, 端口编号	5.8	5-111
OUTDA	OUTDA	按照设定的条件输出电压。	P	OUTDA 电压, 端口编号	6.8	6-97
OUTPUT	O	(或仅 O) 显示输出信号	K	DEFSIG OUTPUT	5.6	5-69
OX. PREOUT	OX	设置外部信号产生的时序。	S	.... OX. PREOUT. ....	7.0	7-5
P	P	显示指定数量的程序步骤。(P: Print, 打印)	E	P 步骤数量	5.1	5-5
PAUSE	PA	暂停程序执行。	P	PAUSE	6.5	6-50
PCABORT	PCA	立即停止指定 PC 程序的执行。(M, P)	M	PCABORT PC 程序编号:	10.0	10-4
PCABORT	PCA	立即停止指定 PC 程序的执行。(M, P)	P	PCABORT PC 程序编号:	10.0	10-4
PCCONTINUE	PCC	继续 PC 程序的执行。(M)	M	PCCONTINUE PC 程序编号:NEXT	10.0	10-6
PCEND	PCEN	停止指定 PC 程序的执行。(M, P)	M	PCEND PC 程序编号:任务编号	10.0	10-5

名称	缩写	功能		格式(参数)	节	页
PCEND	PCEN	停止指定 PC 程序的执行。(M, P)	P	PCEND PC 程序编号:任务编号	10.0	10-5
PCEXECUTE	PCEX	执行指定的 PC 程序。(M, P)	M	PCEXECUTE PC 程序编号, 程序名称, 执行周期, 步骤编号	10.0	10-3
PCEXECUTE	PCEX	执行指定的 PC 程序。(M, P)	P	PCEXECUTE PC 程序编号, 程序名称, 执行周期, 步骤编号	10.0	10-3
PCKILL	PCK	初始化 PC 程序执行堆栈。(M)	M	PCKILL PC 程序编号:	10.0	10-4
PCSCAN	PCSC	指定 PC 程序处理时间。(选项) (P)	P	PCSCAN 时间	10.0	10-8
PCSTATUS	PCSTA	显示指定 PC 程序的状态。(M)	M	PCSTATUS PC 程序编号:	10.0	10-2
PCSTEP	PCSTE	执行 PC 程序的单个步骤。(M)	M	PCSTEP PC 程序编号:程序名称, 执行周期, 步骤编号	10.0	10-7
PI	PI	返回常数 $\pi$	F	PI	9.3	9-47
PNL_CYCST	PNL_CYCST	显示操作面板上 CYCLE START (循环启动) 的 ON/ OFF 状态。	S	SWITCH (PNL_CYCST)	7.0	7-14
PNL_ERESET	PNL_ERESET	显示操作面板上 ERRORRESET (错误复位) 的 ON/ OFF 状态。	S	SWITCH (PNL_ERESET)	7.0	7-15
PNL_MPOWER	PNL_MPOWER	显示操作面板上 MOTOR POWER (马达电源) 的 ON/ OFF 状态。	S	SWITCH (PNL_MPOWER)	7.0	7-15
POINT	P0	定义一个位姿变量。	M	POINT 位姿变量 1= 位姿变量 2, 关节位移值变量	5.5	5-45
POINT	P0	定义一个位姿变量。	P	POINT 位姿变量 1= 位姿变量 2, 关节位移值变量	6.9	6-100
POINT/7	P0/7	设定一个位姿变量的第 7 轴值。	M	POINT/7 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/7	P0/7	设定一个位姿变量的第 7 轴值。	P	POINT/7 变换值变量 1= 变换值变量 2	6.9	6-101
POINT/A	P0/A	设定一个位姿变量的 A 值。	M	POINT/A 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/A	P0/A	设定一个位姿变量的 A 值。	P	POINT/A 变换值变量 1= 变换值变量 2	6.9	6-101
POINT/O	P0/O	设定一个位姿变量的 O 值。	M	POINT/O 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/O	P0/O	设定一个位姿变量的 O 值。	P	POINT/O 变换值变量 1= 变换值变量 2	6.9	6-101
POINT/OAT	P0/OAT	设定一个位姿变量的 OAT 值。	M	POINT/OAT 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/OAT	P0/OAT	设定一个位姿变量的 OAT 值。	P	POINT/OAT 变换值变量 1= 变换值变量 2	6.9	6-101
POINT/T	P0/T	设定一个位姿变量的 T 值。	M	POINT/T 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/T	P0/T	设定一个位姿变量的 T 值。	P	POINT/T 变换值变量 1= 变换值变量 2	6.9	6-101
POINT/X	P0/X	设定一个位姿变量的 X 值。	M	POINT/X 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/X	P0/X	设定一个位姿变量的 X 值。	P	POINT/X 变换值变量 1= 变换值变量 2	6.9	6-101

名称	缩写	功能		格式(参数)	节	页
POINT/Y	PO/Y	设定一个位姿变量的 Y 值。	M	POINT/Y 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/Y	PO/Y	设定一个位姿变量的 Y 值。	P	POINT/Y 变换值变量 1= 变换值变量 22	6.9	6-101
POINT/Z	PO/Z	设定一个位姿变量的 Z 值。	M	POINT/Z 变换值变量 1= 变换值变量 2	5.5	5-47
POINT/Z	PO/Z	设定一个位姿变量的 Z 值。	P	POINT/Z 变换值变量 1= 变换值变量 2	6.9	6-101
POWER	POWER	显马达电源的 ON/OFF 状态。	S	SWITCH(POWER)	7.0	7-11
#PPOINT	#PPOINT	返回由给定分量组成的关节位移值。	F	#PPOINT (jt1, jt2, jt3, jt4, jt5, jt6)	9.2	9-39
PREFETCH. SIGINS	PR	允许或禁止提前处理 AS 程序中的 I/O 信号。	S	.... PREFETCH. SIGINS....	7.0	7-5
PRIME	PRIM	准备执行的程序。	M	PRIME 程序名, 执行循环数, 步骤编号	5.4	5-36
PRINT	PRIN	显示数据。	M	PRINT 设备编号: 打印数据, ...	5.8	5-103
PRINT	PRIN	显示数据。	P	PRINT 设备编号: 打印数据, ...	6.8	6-90
PRIORITY	PRIORITY	返回机器人程序的优先级(选项)。	F	PRIORITY	9.1	9-21
PROG. DATE	PROG. DATE	决定程序的更改日期是否加入到程序信息中。	S	PROG. DATE	7.0	7-25
PROMPT	PROM	在终端上显示信息, 并等待键盘的输入。	P	PROMPT 设备编号: 字符串, 变量	6.8	6-91
PULSE	PU	开启 (ON) 输出信号指定的时间量。	M	PULSE 信号编号, 时间	5.7	5-91
PULSE	PU	开启 (ON) 输出信号指定的时间量。	P	PULSE 信号编号, 时间	6.7	6-70
QTOOL	Q	允许或禁止在一体化示教过程中进行工件转换。	S	.... QTOOL....	7.0	7-6
R	R	替换字符。(R: Replace, 替换)	E	R 字符串	5.1	5-9
RANDOM	RANDOM	返回一个随机数	F	RANDOM	9.3	9-47
REC_ACCEPT	REC	允许/禁止程序的记录和/或更改。	M	REC_ACCEPT	5.6	5-80
REFFLTRESET	REFFLTRESET	复位机器人的平均化时间。	P	REFFLTRESET	6.2	6-27
REFFLTSET	REFFLTSET	指定机器人指令值的平均化时间。	P	REFFLTSET 关节值平均化时间, 位置平均化时间, 姿势平均化时间, 信号平均化时间	6.2	6-23
REFFLTSET_STATUS	REFFLTSET_STATUS	显示指令值的平均化时间值。	M	REFFLTSET_STATUS	5.6	5-86
RELAX	RELAX	在下一运动命令开始时, 关断夹紧信号。	P	RELAX 夹紧编号	6.3	6-35
RELAXI	RELAXI	在当前运动命令完成时, 关断夹紧信号。	P	RELAXI 夹紧编号	6.3	6-35
RELAXS	RELAXS	在运动命令执行过程中, 关断夹紧信号。	P	RELAXS 夹紧编号	6.3	6-36
RENAME	REN	改变程序的名称。	M	RENAME 新程序名=已存在的程序名	5.2	5-20

名称	缩写	功能		格式(参数)	节	页
REP_ONCE	REP	设置再现循环一次还是连续。	S	... REP_ONCE...	7.0	7-8
REP_ONCE.PRS_LAST	REP_ONCE.PRS_LAST	选择在重复一次操作时的结束步骤。		... REP_ONCE.PRS_LAST...	7.0	7-23
REPEAT	REPEAT	显示 TEACH/REPEAT 开关的状态。	S	SWITCH (REPEAT)	7.0	7-13
RESET	RES	关断全部外部输出信号。	M	RESET	5.7	5-89
RESET	RES	关断全部外部输出信号。	P	RESET	6.7	6-69
RETRACE	RETRACE	释放 SETTRACE 保留的存储器。(选项)	M	RETRACE	5.2	5-24
RETURN	RET	返回子程序的调用程序。	P	RETURN	6.5	6-45
RETURNE	RETURNE	从出错步骤后面的步骤开始执行。	P	RETURNE	6.5	6-54
RGSO	RGSO	显示伺服的 ON/OFF 状态。	S	SWITCH (RGSO)	7.0	7-12
\$RIGHT	\$RIGHT	返回字符串中最右边的字符。	F	\$RIGHT(字符串, 字符的数量)	9.4	9-51
RIGHTY	RI	改变形态, 使机器人手臂像人的右手臂。	P	RIGHTY	6.4	6-40
RPS	RP	允许或禁止程序的随机选择。	S	... RPS...	7.0	7-7
RUN	RUN	显示保持/运转开关的状态。	S	SWITCH (RUN)	7.0	7-14
RUNMASK	RU	指定要屏蔽的信号。	P	RUNMASK 开始信号编号, 信号的数量	6.7	6-71
RX	RX	返回表示绕轴旋转的变换值。	F	RX (角度)	9.2	9-38
RY	RY	返回表示绕轴旋转的变换值。	F	RY (角度)	9.2	9-38
RZ	RZ	返回表示绕轴旋转的变换值。	F	RZ (角度)	9.2	9-38
S	S	选择要显示程序步骤。	E	S 步骤编号	5.1	5-4
SAVE	SA	保存指定的程序。	M	SAVE/SEL 文件名=程序名, ...	5.3	5-28
SAVE/ELOG	SA/ELOG	保存错误日志数据。	M	SAVE/ELOG 文件名	5.3	5-29
SAVE/P, L, R, S, A	SA/P, ...	把如下的内容存储在磁盘文件中	M	SAVE (/P) (/L) (/R) (/S) (/A) /SEL 文件名=程序名, ...	5.3	5-29
SAVE/ROB	SA/ROB	保存机器人数据。	M	SAVE/ROB 文件名	5.3	5-29
SAVE/SYS	SA/SYS	保存系统数据。	M	SAVE/SYS 文件名	5.3	5-29
SCALL	SCA	分支到一个子程序。	P	SCALL 字符串表达式, 变量	6.5	6-52
SCASE	SCASE	按照指定的字符串执行程序。	P	SCASE 索引变量 OF ... SVALUE ... ANY... END	6.6	6-66
SCNT	SCN	在读到一个指定值时输出计数器信号。(选项)	M	SCNT 计数器信号编号=加信号, 减信号, 计数器清楚信号, 计数器值	5.7	5-95
SCNT	SCNT	在读到一个指定值时输出计数器信号。(选项)	P	SCNT 计数器信号编号=加信号, 减信号, 计数器清楚信号, 计数器值	6.7	6-80
SCNTRESET	SCNTR	清除计数器信号编号。(选项)	M	SCNTRESET 计数器信号编号	5.7	5-96
SCNTRESET	SCNTR	清除计数器信号编号。(选项)	P	SCNTRESET 计数器信号编号	6.7	6-81
SCREEN	SC	控制终端显示。	S	... SCREEN...	7.0	7-7
SET_TOOLSHAPE	SET_TOOLSHAPE	工具形状的注册。	M	SET_TOOLSHAPE 工具形状编号=变换值变量 1, 变换值变量 2, ..., 变换值变量 8	5.6	5-59
SET_TOOLSHAPE	SET_TOOLSHAPE	工具形状的注册。	P	SET_TOOLSHAPE 工具形状编号=变换值变量 1, 变换值变量 2, ..., 变换值变量 8	6.9	6-104

名称	缩写	功能		格式(参数)	节	页
SET2HOME	SET2	设定原点位姿 2。	M	SET2HOME 精度, HERE	5.6	5-62
SET2HOME	SET2HOME	设定原点位姿 2。	P	SET2HOME 精度, 关节位移值变量	6.9	6-106
SETHOME	SETH	设定原点位姿。	M	SETHOME 精度, HERE	5.6	5-62
SETHOME	SETHOME	设定原点位姿。	P	SETHOME 精度, 关节位移值变量	6.9	6-106
SETOUTDA	SETOUTDA	设定模拟输出环境。	M	SETOUTDA 端口编号=LSB, 位数, 逻辑, 最高电压, 最低电压	5.8	5-110
SETOUTDA	SETOUTDA	设定模拟输出环境。	P	SETOUTDA 端口编号=LSB, 位数, 逻辑, 最高电压, 最低电压	6.8	6-96
SETPICK	SETPICK	设定开始夹具闭合控制的时间。(选项)	M	SETPICK 时间 1, ..., 时间 8	5.7	5-100
SETPICK	SETPICK	设定开始夹具闭合控制的时间。(选项)	P	SETPICK 时间 1, ..., 时间 8	6.7	6-84
SETPLACE	SETPLACE	设定开始夹具打开控制的时间。(选项)	M	SETPLACE 时间 1, ..., 时间 8	5.7	5-100
SETPLACE	SETPLACE	设定开始夹具打开控制的时间。(选项)	P	SETPLACE 时间 1, ..., 时间 8	6.7	6-84
SETTRACE	SETTRACE	为日志保留存储器。(选项)	M	SETTRACE 步骤数量	5.2	5-24
SFLK	SFLK	打开(ON)/关闭(OFF)闪烁信号。(选项)	M	SFLK 信号编号=时间	5.7	5-96
SFLK	SFLK	打开(ON)/关闭(OFF)闪烁信号。(选项)	P	SFLK 信号编号=时间	6.7	6-81
SFLP	SFLP	用 SET/RESET 信号打开(ON)/关闭(OFF)信号。(选项)	M	SFLP 输出信号=置位信号表达式, 复位信号表达式	5.7	5-97
SFLP	SFLP	用 SET/RESET 信号打开(ON)/关闭(OFF)信号。(选项)	P	SFLP 输出信号=置位信号表达式, 复位信号表达式	6.7	6-82
SHIFT	SHIFT	返回由原来位姿平移产生的变换值。	F	SHIFT (变换值 BY X 平移量, Y 平移量, Z 平移量)	9.2	9-40
SIG	SIG	返回指定信号的逻辑与(AND)。	F	SIG (信号编号, ...)	9.1	9-4
SIGNAL	SI	开启(ON)/ 关断(OFF)外部 I/O 信号和内部信号。	M	SIGNAL 信号编号, ...	5.7	5-90
SIGNAL	SI	开启(ON)/ 关断(OFF)外部 I/O 信号和内部信号。	P	SIGNAL 信号编号, ...	6.7	6-69
SIN	SIN	返回正弦值	F	SIN (实数值)	9.3	9-47
SLOAD	SLOAD	将指定文件读入到机器人存储器中, 文件以字符串的形式之指定。	P	SLOAD/IF/ARC 设备编号=文件名, 状态变量	6.10	6-117
SLOW_REPEAT	SL	设置低速再现模式中的速度。	M	SLOW_REPEAT	5.6	5-80
SLOW_START	SLOW_START	允许或禁止低速启动功能(选项)。	S	... SLOW_START...	7.0	7-21
SOUT	S0	指定条件满足时输出信号。(选项)	M	SOUT 信号编号=信号表达式	5.7	5-98
SOUT	S0	指定条件满足时输出信号。(选项)	P	SOUT 信号编号=信号表达式	6.7	6-83
\$SPACE	\$SPACE	返回指定数量的空格。	F	\$SPACE (空格的数量)	9.4	9-50
SPEED	SP	设定监控速度。	M	SPEED 监控速度	5.4	5-35



名称	缩写	功能		格式(参数)	节	页
SPEED	SP	设定监控速度。	P	SPEED 速度, 旋转速度, ALWAYS	6.2	6-16
SQRT	SQRT	返回数学表达式的平方根	F	SQRT (实数值)	9.3	9-47
STABLE	STA	各轴与设置一致时(轴一致)停止机器人运动指定长度的时间。	P	STABLE 时间	6.1	6-4
STAT_ON_KYBD	STAT_ON_KYBD	在键盘屏幕上显示状态信息。	S	STAT_ON_KYBD	7.0	7-22
STATUS	STA	显示系统状态。	M	STATUS	5.6	5-50
STEP	STE	执行程序的一个步骤。	M	STEP 程序名, 执行循环数, 步骤编号	5.4	5-38
STIM	STI	指定信号为 ON 时开启(ON)计时信号。(选项)	M	STIM 计时信号=输入信号编号, 时间	5.7	5-99
STIM	STI	指定信号为 ON 时开启(ON)计时信号。(选项)	P	STIM 计时信号=输入信号编号, 时间	6.7	6-84
STOP	STO	停止执行循环。	P	STOP	6.5	6-51
STP_ONCE	ST	设置执行一次一步步(单步)还是连续执行。	S	... STP_ONCE ...	7.0	7-8
STPNEXT	STP	用单步模式执行程序。	M	STPNEXT	5.4	5-41
STRTOPOS	STRTOPOS	返回指定字符串变量的位姿数据。	F	STRTOPOS (字符串变量)	9.1	9-28
STRTOVAL	STRTOVAL	返回指定字符串变量的实数值。	F	STRTOVAL (字符串变量)	9.1	9-29
SWAIT	SW	在满足指定的条件之前, 挂起程序的执行。	P	SWAIT 信号编号, .....	6.7	6-74
SWITCH	SW	显示系统开关设置。	M	SWITCH 开关名, ... =ON (=OFF)	5.6	5-64
SWITCH	SWITCH	显示系统开关设置。	F	SWITCH(开关名称)	9.1	9-19
SYSDATA	SYSDATA	返回 AS 参数(选项)。	F	SYSDATA(关键字, opt1, opt2)	9.1	9-24
SYSINIT	SYS	初始化整个系统。	M	SYSINIT	5.6	5-75
T	T	在编辑器模式中示教运动指令(选项)。	E	T 位姿变量	5.1	5-14
TASK	TASK	返回程序执行的状态。	F	TASK(任务编号)	9.1	9-20
TDRAW	TD	沿工具坐标系 X, Y, Z 轴移动指定的量。	P	TDRAW X 平移量, Y 平移量, Z 平移量, X 旋转量, Y 旋转量, Z 旋转量, 速度	6.1	6-9
TEACH_LOCK	TEACH_LOCK	显示 TEACH LOCK(示教锁定)的 ON/OFF 状态	S	SWITCH(TEACH LOCK)	7.0	7-12
THEN	THEN	根据逻辑表达式的结果执行一组程序步骤。	K	IF 逻辑表达式 THEN ... ELSE ... END	6.6	6-56
TIME	TI	显示和设置当前时间和日期。	M	TIME 年-月-日 小时:分钟:秒	5.6	5-54
\$TIME	\$TIME	显示和设置当前时间和日期。	F	\$TIME	9.4	9-58
TIMER	TI	设定计时。	P	TIMER 计时编号=时间	6.9	6-107
TIMER	TIMER	设定计时。	F	TIMER(计时编号)	9.1	9-7
TO	TO	重复程序执行。	K	FOR ... TO ... STEP ... END	6.6	6-62
TOOL	TOOL	定义工具变换值。	M	TOOL 变换值变量, 工具形状编号	5.6	5-58
TOOL	TOOL	定义工具变换值。	P	TOOL 变换值变量, 工具形状编号	6.9	6-103
TOOL	TOOL	定义工具变换值。	F	TOOL	9.2	9-42
TOOLSHAPE	TOOLSHAPE	根据工具形状设定速度控制。	M	TOOLSHAPE 工具形状编号	5.6	5-61
TOOLSHAPE	TOOLSHAPE	根据工具形状设定速度控制。	P	TOOLSHAPE 工具形状编号	6.9	6-106

名称	缩写	功能		格式(参数)	节	页
TPKEY_A	TPKEY_A	显示示教器上 A 键的 ON/ OFF 状态。	S	SWITCH(TPKEY_A)	7.0	7-16
TPLIGHT	TPLIGHT	开启示教器的背光灯。(选项)	M	TPLIGHT	5.6	5-83
TPLIGHT	TPLIGHT	开启示教器的背光灯。(选项)	P	TPLIGHT	6.9	6-111
TRACE	TRACE	开/关 TRACE(跟踪)功能。(选项)	M	TRACE <b>stepper 编号</b> :ON/OFF	5.2	5-23
TRACE	TRACE	开/关 TRACE(跟踪)功能。(选项)	P	TRACE <b>stepper 编号</b> :ON/OFF	6.10	6-115
TRADD	TRADD	返回的值为 X 分量加上走行轴的值(选项)。	F	TRADD (变换值变量)	9.2	9-43
TRANS	TRANS	返回由给定分量组成的变换值。	F	TRANS (X 分量, Y 分量, Z 分量, 0 分量, A 分量, T 分量)	9.2	9-37
TRIGGER	TRIGGER	显示握杆触发开关的 ON/OFF 状态。	S	SWITCH (TRIGGER)	7.0	7-10
TRSUB	TRSUB	返回的值为 X 分量减去走行轴的值(选项)。	F	TRSUB (变换值变量)	9.2	9-43
TWAIT	TW	将程序的执行置于待机状态,直到指定的时间流逝。	P	TWAIT 时间	6.5	6-47
TYPE	TY	显示数据。	M	TYPE <b>设备编号</b> :打印数据, ...	5.8	5-104
TYPE	TY	显示数据。	P	TYPE <b>设备编号</b> :打印数据, ...	6.8	6-90
ULIMIT	UL	设定机器人运动的上限。	M	ULIMIT <b>关节位移值变量</b>	5.6	5-55
ULIMIT	UL	设定机器人运动的上限。	P	ULIMIT <b>关节位移值变量</b>	6.9	6-107
UNTIL	UN	创建一个 DO 循环。	P	DO ... UNTIL 逻辑表达式	6.6	6-60
USB_COPY	USB_COPY	复制 USB 闪存中的文件。	M	USB_COPY 新的文件名=源文件名	5.2	5-23
USB_FDEL	USB_FDEL	删除 USB 闪存中的文件。	M	USB_FDEL 文件名, ...	5.2	5-19
USB_FDIR	USB_FDIR	在编辑器模式中示教运动指令。	M	USB_FDIR 文件名	5.2	5-16
USB_LOAD	USB_LOAD	装载 USB 闪存中的程序到机器人存储器中。	M	USB_LOAD/ <b>Q</b> 文件名	5.3	5-31
USB_MKDIR	USB_MKDIR	在 USB 闪存中创建新的文件夹。	M	USB_MKDIR 文件夹名	5.3	5-33
USB_RENAME	USB_RENAME	改变用于 USB 闪存中的文件名。	M	USB_RENAME 新文件名= 已经存在的文件名	5.2	5-20
USB_SAVE	USB_SAVE	保存指定的程序到 USB 闪存中。	M	USB_SAVE/SEL 文件名= <b>程序名</b> , ...	5.3	5-28
USB_SAVE/A	USB_SAVE/A	保存辅助信息到 USB 闪存中。	M	USB_SAVE/A 文件名	5.3	5-29
USB_SAVE/ELOG	USB_SA/ELOG	保存错误日志到 USB 闪存中。	M	USB_SAVE/ELOG 文件名	5.3	5-29
USB_SAVE/P, L, R, S,	USB_SA/P, ...	保存指定数据到 USB 闪存中。	M	USB_SAVE (/P) (/L) (/R) (/S) /SEL 文件名= <b>程序名</b> , ...	5.3	5-29
USB_SAVE/ROB	USB_SAVE/ROB	保存系统数据到 USB 闪存中。	M	USB_SAVE/ROB 文件名	5.3	5-29
USB_SAVE/SYS	USB_SAVE/SYS	保存系统数据。	M	USB_SAVE/SYS 文件名	5.3	5-29
UTIMER	UTIMER	设定用户计时的值。	P	UTIMER @计时变量= 计时值	6.9	6-108
UTIMER	UTIMER	设定用户计时的值。	F	UTIMER (@计时变量名称)	9.1	9-21
UWRIST	UW	改变形态,使 JT5 的角度为正值。	P	UWRIST	6.4	6-41
VAL	VAL	把实数值返回到指定的字符串中。	F	VAL (字符串, <b>代码</b> )	9.1	9-13
WAIT	WA	将程序的执行置于待机状态,直到条件满足。	P	WAIT 条件	6.5	6-46
WAITREL_AUTO	WAITREL_AUTO	显示等待解除的弹出式窗口。	S	WAITREL_AUTO	7.0	7-22

名称	缩写	功能		格式(参数)	节	页
WEIGHT	WE	设定负荷重量数据。	M	WEIGHT 负荷质量, 重心位置 X, 重心位置 Y, 重心位置 Z, 绕 X 轴的惯性力矩, 绕 Y 轴的惯性力矩, 绕 Z 轴的惯性力矩	5.6	5-78
WEIGHT	WE	设定负荷重量数据。	P	WEIGHT 负荷质量, 重心位置 X, 重心位置 Y, 重心位置 Z, 绕 X 轴的惯性力矩, 绕 Y 轴的惯性力矩, 绕 Z 轴的惯性力矩	6.9	6-110
WHERE	W	显示目前机器人的当前位姿数据。	M	WHERE 显示模式	5.6	5-51
WHICHTASK	WHICHTASK	返回由程序或子程序选中的任务号。	F	WHICHTASK 程序名称	9.1	9-19
WHILE	WH	当指定的条件为真(TRUE)时, 程序命令被执行。如果条件为假(FALSE), WHILE 语句被跳过。	P	WHILE ... DO ... END	6.6	6-58
WS. ZERO	WS. ZERO	改变 WS=0 时执行的焊接处理。(选项)	S	... WS. ZERO ...	7.0	7-19
WS_COMPOFF	WS_COMPOFF	改变 WS 信号的输出时序。(选项)	S	... WS_COMPOFF ...	7.0	7-18
XD	XD	剪切选中的一个或多个步骤, 并保存在剪贴板上。	E	XD 步骤数量	5.1	5-11
XFER	XF	从一个程序复制步骤到另一个程序。	M	XFER 目标程序名, 步骤编号 1=源程序名, 步骤编号 2, 步骤数量	5.2	5-21
XMOVE	X	以直线插补动作向指定位姿移动。	P	XMOVE 模式 位姿变量名 TILL 信号编号	6.1	6-11
XOR	XOR	逻辑异或	O	XOR	8.3	8-3
XP	XP	粘贴剪贴板的内容。	E	XP	5.1	5-12
XQ	XQ	以相反的顺序粘贴剪贴板的内容(反向粘贴)。	E	XQ	5.1	5-12
XS	XS	显示剪贴板的内容。	E	XS	5.1	5-13
XY	XY	复制选中的一个或多个步骤, 并保存在剪贴板上。	E	XY 步骤数量	5.1	5-11
ZSIGSPEC	ZSIG	设定和显示 I/O 信号的总数。	M	ZSIGSPEC	5.6	5-66
ZZERO	ZZ	显示或设定调零数据。	M	ZZERO 关节编号	5.6	5-72
FALSE	FALSE	返回值 0.0, 代表逻辑假(FALSE)。	F	... FALSE ...	9.1	9-12
TRUE	TRUE	返回值 -1.0, 代表逻辑真(TRUE)。	F	... TRUE ...	9.1	9-12
+	+	加	O	.... + ...	8.1	8-1
-	-	减	O	.... - ....	8.1	8-1
*	*	乘	O	.... *. ....	8.1	8-1
/	/	除	O	.... / ....	8.1	8-1
^	^	乘方	O	.... ^ ....	8.1	8-1
+	+	两个变换值相加	O	.... + ...	8.5	8-6
-	-	两个变换值相减	O	.... - ....	8.5	8-6
+	+	合并两个字符串	O	.... + ...	8.6	8-8
<	<	当左边的值小于右边时为真(-1)	O	.... < ...	8.2	8-2



名称	缩写	功能		格式(参数)	节	页
<=	<=	当左边的值小于或等于右边时为真(-1)	0	.... <= ...	8.2	8-2
<>	<>	当两边不等时为真(-1)	0	.... <> ...	8.2	8-2
=<	=<	与上述相同	0	.... =< ...	8.2	8-2
==	==	当两边相等时为真(-1)	0	.... = ...	8.2	8-2
=>	=>	与上述相同	0	.... => ...	8.2	8-2
>	>	当左边的值大于右边时为真(-1)	0	.... > ...	8.2	8-2
>=	>=	当左边的值大于或等于右边时为真(-1)	0	.... >= ...	8.2	8-2

---

川崎机器人控制器 E 系列  
AS 语言参考手册

---

2009 年 8 月 : 第一版  
2011 年 5 月 : 第二版

出版 : 川崎重工业株式会社

90209-1022DCB