



三菱电机工业机器人

CR750/CR751/CR760 系列控制器

操作说明书

功能和操作的详细说明

本书记载对应 iQ Platform 的 CR750-Q/CR751-Q/CR760-Q 机器人控制器与独立设备型
CR750-D/CR751-D/CR760-D 机器人控制器的错误一览。

MELFA
BFP-A8985-B

安全上的注意

使用机器人前请务必熟读以下的注意事项及别册「安全手册」，了解如何使用机器人，以确保您及周遭环境的安全。

A. 确认使用的方法能遵守法律的规则。

注意

示教作业等因安全的考虑，必须由受过操作教育训练的人员操作使用。
(无切断电力的保养作业亦相同)
→安全教育的实施

注意

请将示教作业依机器人的操作方法及顺序、异常时及再起动时处理等制作相关作业规程、并遵守规章内容。
(无切断电力的保养作业亦相同)
→作业规程的制作

警告

示教作业请设定为可立即停止运行的装置。
(无切断电力的保养作业亦相同)
→紧急停止开关的设定

注意

示教作业中请将 [示教作业中] 的标示放置在起动开关上。
(无切断电力的保养作业亦相同)
→示教作业中的表示

危险

运行中请确认使用围离或栅栏将操作人员与机器人做隔离，防止直接接触机器人。
→安全栅栏的设置

注意

运行开始，对于相关人员的信号有固定的方法，请依此进行。
→运行开始的信号

注意

维护作业原则上请中断电力进行、请将 [保养作业中] 的标语放置在起动开关上。
→维护作业中的表示

注意

作业开始前请详细的检查，确认机器人及紧急停止开关、相关装置等无异常状况。
→作业开始前的检查

B. 摘录别册的「安全手册」重点注意事项。
详细内容请参照「安全手册」

-  **危险** 使用复数的控制机器（GOT、PLC、按钮开关）执行机器人自动运行的情况下，各机器的操作权等的互锁请客户端自行设计。
-  **注意** 请在规格范围内的环境中使用机器人。除此之外的场合是容易造成机器人故障原因。（温度、湿度、空气、噪声环境等）
-  **注意** 请依照机器人指定的搬运姿势进行搬运或移动机器人。指定以外的搬运方式有可能因为掉落而造成人身安全或机器人故障。
-  **注意** 请确实将机器人固定在底座上。不稳定的姿势有可能产生位置偏移或发生振动。
-  **注意** 电线是产生噪声的原因，请尽可能将配线拉开距离。
太过接近有可能造成位置偏移及错误动作。
-  **注意** 请勿用力拉扯接头或过度的卷屈电线。
因有可能造成接触不良及电线断裂量的情况。
-  **注意** 抓手所包含的工件重量请勿超出额定负载及最大力矩。
超出重量的情况下有可能发生报警及故障。
-  **警告** 请确保抓手、工具的取放及工件的抓握牢固。否则运作中工件有可能飞散出去而造成人员及物品的损伤。
-  **警告** 机器人及控制器的接地请确认连接。否则容易因为噪声而做错误动作或导致触电事故发生。
-  **注意** 机器人在动作中时请标示为运行状态。没有标示的情况下容易导致人员接近或有错误的操作。
-  **警告** 在机器人的动作范围内做示教作业时，请务必确保机器人的控制有优先权。
否则由外部指令使机器人起动作，有可能造成人员及物品的损伤。
-  **注意** JOG 速度请尽量以低速进行、并请勿在操作中将视线离开机器人。
否则容易干涉到工件及周边装置。
-  **注意** 程序编辑后的自动运行前，请务必确认单步运行动作。若无确认有可能发生程序错误而与周边装置干涉。



注意

自动运行中安全栅栏的出入口门打开被锁住的情况下，机器人会自动停止。否则会发生人员的损伤。



注意

请勿因个人意思做机械的改造及使用指定以外的零件。否则可能导致机械故障或损坏。



警告

从外部用手将机器人手臂做推动的情况下，请勿将手或指头放入开口部位。有可能会夹伤手或指头。



注意

请勿用将机器人控制器的主电源关闭的方式来使机器人停止或紧急停止。在自动运行中将控制器的主电源关闭有可能使机器人精度受到影响。且，有可能发生手臂掉落或松动而干涉到周边装置的情况。



注意

重写控制器内程序或参数等内部数据时，请勿将控制器的主电源关闭。自动运行中或程序参数添写中时，若关闭控制器主电源，则有可能破坏控制器的内部数据。



危险

使用本产品的 GOT 直接连接功能的情况下，请勿连接手持式 GOT。GOT 会使机器人自动运行，无论操作权的有效 / 无效，可能发生物品手持式及人员损伤事故。



危险

在 CR750-Q/CR751-Q/CR760-Q 使用对应 IQPlatform 产品的情况下，请勿在 PLC 上连接手持式 GOT。手编 GOT 会使机器人自动运行，无论操作权的有效 / 无效，可能发生物品及人员损伤事故。



危险

多重 CPU 系统及伺服放大器的电源投入时，请勿拔除 SSCNET III 的接线。请勿直视运动 CPU 及伺服放大器的 SSCNET III 接头及 SSCNET III 电线前端所发出的光。光线直射眼睛，会使眼睛产生失调感。
(SSCNET III 的光源相当于 J1SC6802、IEC60825-1 所规定的等级 1。)



危险

机器人控制器的电源投入时，请勿拔除 SSCNET III 的接线。请勿直视 SSCNET III 接头及 SSCNET III 电线前端所发出的光。光线直射眼睛，会使眼睛产生失调感。
(SSCNET III 的光源相当于 J1SC6802、IEC60825-1 所规定的等级 1。)



危险

SSCNET III 接线取下后，若没有在 SSCNET III 接头部位装上护套的话，会使接头附着灰尘及杂物。会使机器人特性变差、发生错误动作等。



注意

应注意不要出现配线错误。进行了不符合规格的连接的情况下，有可能导致紧急停止无法解除等的误动作。

为了防止误动作，配线完毕后，必须对机器人控制器操作面板、示教单元紧急停止、用户紧急停止、门开关等的各种功能是否正常动作进行确认。



注意

将控制器的 **USB** 与市面销售的设备 (笔记本电脑、计算机、LAN 用集线器等) 连接使用时，有可能与本公司设备的匹配性及温度・噪声等的 **FA** 环境不符合。使用时，有时需要采取 **EMI** 对策 (**Electro-Magnetic Interference**) 或添加铁氧体磁芯等其它对策，用户应进行充分的动作确认。

此外，对于与市面销售设备连接时的动作保障・维护等本公司将不予承担。

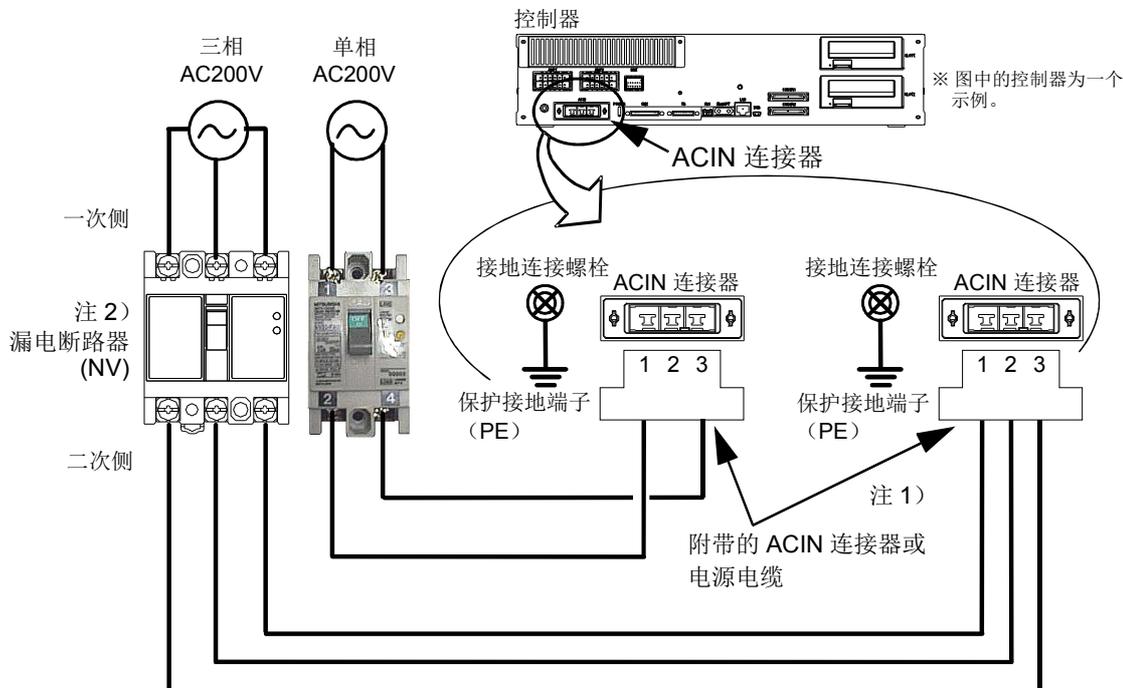
*CR751-D 或 CR751-Q 控制器

基本构成的注意事项如下所示。



注意

为了漏电保护，应对 CR751-D 控制器或 CR751-Q 驱动模块的一次侧供应电源设置漏电断路器。否则有可能导致触电事故。



注 1) 附带的 ACIN 连接器，建议通过压铆进行连接。（也可通过焊接）

推荐铆接工具：234171-1(Tyco Electronics)

注 2) 漏电断路器为用户自备品。请务必使用端子盖板。

推荐：单相用NV30FAU-2P-10A-AC100-240V-30mA（端子盖板：TCS-05FA2）

三相用NV30FAU-3P-10A-AC100-240V-30mA（端子盖板：TCS-05FA3）

- 1) 请准备漏电断路器（带端子盖板）、一次侧电源连接用电缆（AWG #14(2mm²) 以上）、一次侧电源接地用电缆（AWG #12(3.5mm²) 以上）。
二次侧电源连接用电缆虽然会根据产品的规格附带单相或三相用电源电缆（带 ACIN 连接器），但是，使用附带的 ACIN 连接器和 ACIN 端子，根据用户的环境制作电缆时，请准备二次侧电源连接用电缆（AWG #14(2mm²) 以上）。
- 2) 请确认一次侧电源符合电源规格。
- 3) 请确认已切断一次侧电源、及漏电断路器的电源开关为 OFF 状态。
- 4) 连接二次侧电源连接用电缆。
 - a) 使用附带的电源电缆（带 ACIN 连接器）时
请按照上图，从漏电断路器的二次侧进行连接。
 - b) 使用附带的 ACIN 连接器和 ACIN 端子制作电源电缆时
连接 ACIN 端子和二次侧电源连接用电缆（用户自备），将 ACIN 端子插入至 ACIN 连接器的下述针号。ACIN 端子，建议通过压铆进行连接。
单相用：1、3
三相用：1、2、3
请按照上图，从漏电断路器的二次侧进行连接。
- 5) 将电源电缆的 ACIN 连接器连接到控制器前面的 ACIN 连接器上。
- 6) 请将一次侧电源接地用电缆连接到控制器箱体的保护接地端子（M4 螺栓）上。
- 7) 请将一次侧电源连接用电缆连接到漏电断路器的一次侧端子上。

修订记录

印刷日期	手册编号	修改内容
2013年6月	BFP-A8985	第一版
2015年11月30日	BFP-A8985-B	<ul style="list-style-type: none"> • 追加指令 SpdOpt、EHServo On/OFF、EMvSpl、SplWrt、SplFWrt • 修改指令 CmpG、Def Act、Return、GetM、If...Then...Else...EndIf、Mvs、WthIf、Error、Act、Fine P、XClr、CallP、For ~ Next、While ~ WEnd、WithIf、Fine、XLoad、DefPlt、MvSpl • 追加参数 J1OFFSET、CMPJCLL、SPDOPT、MECAR、OPTOVC、WTHFUNC、MVTERM、ESCMODE、PRGMODE、SVDATA、JOGMENO、DRVMODE、NVJBTOUT、CATEGORY • 修改参数 EMGERR、HANDENA、OVRDTB、OVRDENA、ROMDRV、CTN、NETIP、AUTOENA、PRGOUT、LINEOUT、OVRDOUT、ERROUT、NETIP、OPPSL、RMTPSL • 追加机器人（系统）状态变量 M_LdMFax、M_ErCode、M_DIn32、M_DOut32、M_RCInfo、M_ESpd、P_ECord • 修改机器人（系统）状态变量 C_Mecha、C_Prg、M_In32、M_Out32、M_OPOvrd • 追加专用输入输出 DOORSTS1、DOORSTS2、DOORSTS • 追加机型 RV-4FJL、RH-3FHR、RV-50F 系列 (RV-35F、RV-50F、RV-70F)、CR760 控制器 • 追加功能 样条插补功能、Ex-T 控制功能 • 修改说明 2.1 操作面板（O/P）的功能（CR750 控制器） 3.10 无法解除报警的暂时报警复位操作 5.9 自由平面限制 5.11 电源开启的程序自动执行方法 5.14 关于输出信号复位模式 5.15 关于通信设定（以太网） 5.21 关于冲突检知功能 5.24 关于干涉回避功能 6.4 信号的有效无效状态 • 追加函数 ACos、ASin、SplSpd、SplPos、SplECord • 更改企业标志。

■ 前言

非常感谢您购买三菱电机工业机器人。
本使用说明书详细记载了控制器和示教单元（R32TB/R33TB（选购件））的功能和操作方法的相关说明，以及 MELFA-BASIC V 程序语言及规格的相关说明。
适用于对应 iQ Platform 的 CR750-Q/CR751-Q/CR760-Q 控制器和独立设备型的 CR750-D/CR751-D/CR760-D 控制器。特别是个别附带的功能，仅记载“CR750”及“CR750-D”。

此外，本手册中根据控制器前面的操作面板上的按键操作对机器人程序的启动、停止等的操作进行说明。使用未安装操作面板的机器人时，以上操作通过外部信号（专用输入输出信号）进行。在下一页中总结了与操作面板的按键对应的专用输入输出信号。请通过参数设置将专用输入输出信号分配到通用输入输出信号中使用。

操作之前请务必阅读本手册，在充分理解内容的基础上灵活使用机器人。
本手册中对特殊使用也尽量进行了记载，对于本手册中未记载的事项应理解为“不能进行”。此外，本手册所记载的内容以以下的软件版本为对象。

CR750-Q/CR751-Q/CR760-Q: Ver.R6b 以上
CR750-D/CR751-D/CR760-D: Ver.S6b 以上
T/B: Ver.1.7 以上

- 禁止未经允许转载本手册的部分或全部内容。
- 本手册的内容以后有可能在未通知的状况下进行更改，请予以谅解。
- 本手册的内容尽量做到完整无缺，如果发现了疑问点、错误、漏记等，请与所购买的销售商或三菱公司 MELFA 电话中心联系。
- 本规格书为原版。
- 因特网、以太网为印刷公司的登录商标。
- AutoCAD 是美国 Autodesk, Inc. 在美国以及其他国家的注册商标。
- 其他记载的公司名称·产品名为各公司的商标或登录商标。

致使用未安装操作面板的机器人的用户

机器人程序的启动、停止等的操作通过外部信号（专用输入输出信号）进行。本书中，以控制器或驱动模块前面安装有操作面板的机器人为基础，对其按键操作进行介绍。请将各按键操作对应的专用输入输出信号通过参数设置分配到通用输入输出信号中，通过信号操作进行。

下表所示为本书所介绍的对应操作面板各按键的专用输入输出信号。请以此为参考，进行信号的分配。

参数的详情请参照第 516 页的「6.3 专用输出」，各信号的时序图请参照第 527 页的「6.5 外部信号的时序图」，参数的设定方法请参照第 84 页的「3.14 参数画面的操作」。

表：对应操作面板各按键的专用输入输出信号

操作面板按钮、指示灯	参数名称	分类	功能	出厂设置值
START 按钮 START 按钮指示灯	START	输入	启动程序。	3,0
		输出	对程序处于运行中的状态进行输出。	
STOP 按钮 STOP 按钮指示灯	STOP	输入	停止运行中的程序。	0,-1
		输出	对程序处于中断中的状态进行输出。	
RESET 按钮 RESET 按钮指示灯	ERRRESET	输入	解除出错状态。	2,2
		输出	对处于出错状态进行输出。	
	SLOTINIT	输入	解除程序的中断中状态，将执行行返回起始处。进入程序可选择的状态。	-1,-1
		输出	对程序中断中的状态进行输出。	
CHNG DISP 按钮 UP/DOWN 按钮	PRGSEL	输入	本信号 ON 时，选择分配到 IODATA 的数值输入的信号中所输入的值作为程序编号。	-1,
		输出	无	
	PRGOUT	输入	本信号 ON 时，将所选择的程序编号输出到分配到 IODATA 的数值输出中的信号。	-1,-1
		输出	表示程序编号输出正在输出到数值输出中。	
	OVRDSEL	输入	本信号 ON 时，设置输入到分配到 IODATA 的数值输入中的信号的值作为手工变动值。	-1,
		输出	无	
	OVRDOUT	输入	本信号 ON 时，对分配到 IODATA 的数值输出中的信号所设置的手工变动值进行输出。	-1,-1
		输出	表示手工变动值正在输出到数值输出中。	
	LINEOUT	输入	本信号 ON 时，将当前的行编号输出到分配到 IODATA 的数值输出中的信号。	-1,-1
		输出	表示行编号正在输出至数值输出。	
	ERROUT	输入	本信号 ON 时，将出错编号输出到分配到 IODATA 的数值输出中的信号。	-1,-1
		输出	表示出错编号正在输出至数值输出。	
	IODATA	输入	上述程序编号和手工变动值作为二进制值进行读取。	-1,-1, -1,-1
		输出	上述程序编号、手工变动值、行编号以及出错编号作为二进制值进行输出。	
END 按钮 END 按钮指示灯	CYCLE	输入	进行循环停止。	-1,-1
		输出	对循环停止动作中状态进行输出。	
SVO.ON 按钮 SVO.ON 按钮指示灯	SRVON	输入	将伺服电源置为 ON。	4,1
		输出	对伺服电源处于 ON 状态进行输出。	
SVO.OFF 按钮 SVO.OFF 按钮指示灯	SRVOFF	输入	将伺服电源置为 OFF。	1,-1
		输出	对伺服电源不可 ON 的状态进行输出。（响应返回）	

目录

页

1 开始使用	1-1
1.1 操作说明书的使用方法	1-1
1.1.1 各操作说明书的内容	1-1
1.1.2 操作说明书中的符号	1-2
1.2 安全上的注意	1-3
1.2.1 确认使用的方法能遵守法律的规则。	1-3
1.2.2 安全手册注意事项	1-4
2 功能的说明	2-7
2.1 操作面板（O/P）的功能（CR750 控制器）	2-7
(1) 操作面板的按钮说明	2-7
(2) STATUS 编号（显示面板）的说明	2-7
(3) 机型复位操作的说明	2-8
2.2 操作面板（O/P）的功能（CR760 控制器）	2-9
(1) 操作面板的按钮说明	2-9
(2) 机型复位操作的说明	2-10
2.3 示教单元的（T/B）的功能	2-11
2.3.1 关于操作权	2-13
2.4 动作及控制相关的功能	2-14
3 操作方法的说明	3-16
3.1 示教单元的菜单画面操作	3-16
(1) 画面树形图	3-16
(2) 数字 / 文字的输入	3-20
(3) 菜单的选择方法	3-21
3.2 JOG 前进（概略说明）	3-22
3.2.1 JOG 前进的种类	3-22
3.2.2 JOG 前进的速度	3-24
3.2.3 关节 JOG	3-25
3.2.4 直交 JOG	3-25
3.2.5 TOOL JOG	3-26
3.2.6 三轴直交 JOG	3-26
3.2.7 圆筒 JOG	3-27
3.2.8 工件 JOG	3-27
3.2.9 工具数据的切换	3-28
3.2.10 世界坐标系的切换（基本坐标号码的指定）	3-29
3.2.11 JOG 操作中的冲突检知	3-31
(1) JOG 操作时的冲突检知基准调整	3-31
3.3 抓手开闭	3-32
3.4 整列抓手	3-33
3.5 程序编辑	3-34
3.5.1 程序的制作	3-34
(1) 开启程序编辑画面	3-34
(2) 程序的制作	3-35
(3) 程序制作的结束储存	3-37
(4) 程序的修正方法	3-38
(5) 现在位置的示教	3-40
(6) 位置变量的删除	3-43
(7) 位置数据的确认	3-44
(8) 位置数据的 MDI（Manual Data Input）登录（修正）	3-45
3.6 排错操作	3-46
(1) 单步前进	3-46
(2) 步编号返回	3-47
(3) 其它插槽的单步前进	3-48
(4) STEP 跳转	3-50
3.7 自动运行	3-51

3.7.1 动作速度的设定	3-51
(1) 在 O/P 的操作	3-51
(2) 在示教单元的操作	3-51
3.7.2 程序号码的选择	3-51
3.7.3 自动运行的开始	3-52
(1) 通过 O/P 启动	3-52
(2) 通过示教单元启动	3-53
3.7.4 停止	3-56
(1) 在 O/P 的操作	3-56
(2) 在示教单元的操作	3-56
3.7.5 停止之后重开启自动运行	3-57
(1) 右面板 (O/P) 再开	3-57
(2) 通过示教单元重启	3-57
3.7.6 程序复位	3-58
(1) 在操作面板 (O/P) 的操作	3-58
(2) 在示教单元的操作	3-58
3.8 伺服 ON/OFF	3-59
3.9 报警复位操作	3-60
3.10 无法解除报警的暂时报警复位操作	3-60
3.11 管理·编辑画面的操作	3-61
(1) 程序一览表示	3-61
(2) 复制程序 (复制)	3-62
(3) 变更程序名称 (重命名)	3-63
(4) 删除程序 (删除)	3-64
(5) 保护 (Protect)	3-65
(6) 程序的选择	3-66
3.12 运行画面的操作	3-67
3.12.1 执行行的显示	3-67
(1) 确认画面的选择	3-67
(2) 单步运行	3-67
(3) 单步跳转	3-69
(4) 在其它插槽的单步前进	3-70
(5) 确认画面的结束	3-70
3.12.2 单步运行画面	3-71
(1) 单步运行画面的选择	3-71
3.12.3 操作面板画面的操作	3-72
3.13 监视画面的操作	3-73
(1) 输入信号监视	3-73
(2) 输出信号监视	3-75
(3) 输入寄存器监视	3-77
(4) 输出寄存器监视	3-78
(5) 变量监视	3-81
(6) 报警历史	3-83
3.14 参数画面的操作	3-84
3.15 原点·制动器画面的操作	3-86
(1) 原点	3-86
(2) 刹车	3-86
3.16 设定 / 初始化画面的操作	3-88
(1) 程序的初始化	3-88
(2) 参数的初始化	3-89
(3) 电池的初始化	3-90
(4) 操作	3-91
(5) 时间设定	3-92
(6) 版本	3-92
3.17 扩展功能的操作	3-93
(1) PLC 直接	3-93

(2) 工件坐标	3-93
3.18 初始设定画面的操作	3-94
(1) 显示语言的设定	3-94
(2) 明亮度的设定示教单元	3-96
4 MELFA-BASIC V	4-98
4.1 MELFA-BASIC V 的功能	4-98
4.1.1 机器人的动作控制	4-99
(1) 关节插补动作	4-99
(2) 直线插补动作	4-100
(3) 圆弧插补动作	4-101
(4) 连续动作	4-103
(5) 加减速时间和速度控制	4-104
(6) 往目的位置的到达确认	4-106
(7) 高轨迹精度控制	4-107
(8) 抓手·TOOL 控制	4-108
4.1.2 Pallet 运算	4-109
4.1.3 程控	4-113
(1) 无条件分支·条件分支·待机	4-113
(2) 循环	4-115
(3) 插入	4-116
(4) 子程序	4-117
(5) 定时器	4-118
(6) 停止	4-119
4.1.4 外部信号的输出输入	4-120
(1) 信号输入	4-120
(2) 信号输出	4-120
4.1.5 通信	4-121
4.1.6 运算	4-122
(1) 运算符一览	4-122
(2) 位置数据的相对运算（乘算）	4-123
(3) 位置数据的相对运算（加算）	4-123
4.1.7 附随句	4-124
4.2 MELFA-BASIC V 和 MELFA-BASIC IV 的差异	4-125
4.2.1 关于 MELFA-BASIC V	4-125
4.2.2 MELFA-BASIC V 的特点	4-125
4.2.3 和 MELFA-BASIC IV 的比较	4-125
4.3 多重任务处理功能（Multitask）	4-126
4.3.1 所谓多重任务处理功能	4-126
4.3.2 多重任务的执行方法	4-127
4.3.3 各多重插槽运行状态	4-127
4.3.4 多任务程序制作上的注意点	4-129
(1) 任务数目和处理时间的关系	4-129
(2) 指定并列执行程序行数的最大值	4-129
(3) 在全局变量的程序间的数据的交接方法	4-129
(4) 用机器人（系统）状态变量的程序运行状态的确认	4-129
(5) 机器人动作的程序基本上在任务插槽 1 执行	4-129
(6) 在平时执行程序的初始化处理	4-129
4.3.5 多任务插槽程序运行上的注意点	4-130
(1) 多任务的开始	4-130
(2) 运行状态的表示	4-130
4.3.6 多任务的使用例	4-131
(1) 机器人的作业内容	4-131
(2) 到多任务为止的步骤	4-132
4.3.7 关于程序容量	4-133
(1) 程序储存领域	4-133
(2) 程序编辑领域	4-133

(3) 程序执行领域	4-133
4.4 MELFA-BASIC V 的详细规格	4-134
(1) 程序名制作	4-134
(2) 指令	4-134
(3) 变量	4-134
4.4.1 文	4-135
4.4.2 附随文	4-135
4.4.3 行	4-135
4.4.4 行号码	4-135
4.4.5 标签	4-135
4.4.6 可以在程序内使用的文字种类	4-136
4.4.7 有特别意义的文字	4-137
(1) 大写、小写的区别	4-137
(2) 下划线 ()	4-137
(3) 撇号 (')	4-137
(4) 星号 (*)	4-137
(5) 逗号 (,)	4-137
(6) 句号 (.)	4-137
(7) 空白	4-137
4.4.8 数据类型	4-138
4.4.9 常数	4-138
4.4.10 数值常数	4-138
(1) 10 进制	4-138
(2) 16 进制	4-138
(3) 2 进制	4-138
(4) 常数类型	4-138
4.4.11 字符串常数	4-138
4.4.12 位置常数	4-139
(1) 坐标·姿势·附加轴资料的形式和意义	4-139
(2) 构造标志数据的形式和意义	4-139
4.4.13 关节常数	4-140
(1) 各轴数据的形式和意义	4-140
4.4.14 角度值	4-141
4.4.15 变量	4-141
4.4.16 数值变量	4-142
4.4.17 字符串变量	4-142
4.4.18 位置变量	4-142
4.4.19 关节变量	4-143
4.4.20 输出输入变量	4-143
4.4.21 数组变量	4-143
4.4.22 全局变量	4-144
4.4.23 程序全局变量	4-144
4.4.24 使用者定义全局变量	4-145
4.4.25 使用者基本程序的制作	4-145
4.5 机器人的坐标系说明	4-146
4.5.1 关于机器人的坐标系	4-146
4.5.2 关于基本变换	4-147
4.5.3 关于位置数据	4-148
4.5.4 关于 TOOL 坐标系 (机械 I/F 坐标系)	4-149
(1) 机械 I/F 坐标系	4-149
(2) TOOL 坐标系	4-150
(3) TOOL 坐标系的使用效果	4-151
4.6 机器人 (系统) 状态变量	4-153
4.6.1 逻辑数	4-156
4.7 函数	4-156
(1) 使用者定义函数	4-156
(2) 内建函数	4-156

目录

	页
4.8 指令一览	4-158
(1) 动作控制关连的指令	4-158
(2) 程序控制相关的指令	4-159
(3) 定义指令	4-160
(4) 多任务相关	4-160
(5) 其它	4-160
4.9 运算符号	4-161
4.10 运算的优先级	4-162
4.11 程序的控制构造深度	4-162
4.12 保留字	4-162
4.13 指令的详细说明	4-163
4.13.1 记载项目的见解	4-163
4.13.2 各指令的说明	4-163
4.14 机器人（系统）状态变量的详细说明	4-287
4.14.1 记载项目的说明	4-287
4.14.2 各机器人状态变量（系统状态变量）的说明	4-287
4.15 函数的详细说明	4-366
4.15.1 记载项目的说明	4-366
4.15.2 各函数的说明	4-366
5 参数的设定功能	5-407
5.1 动作参数	5-407
5.2 信号参数	5-418
5.2.1 关于多 CPU 输入偏置（仅限 CR7xx-Q 控制器）	5-422
(1) 案例（A）	5-422
(2) 案例（B）	5-423
5.3 操作参数	5-425
5.4 程序参数	5-427
5.5 通信参数	5-430
5.6 关于标准 TOOL 坐标	5-432
5.7 关于标准基本坐标	5-434
5.8 关于使用者定义领域	5-435
5.8.1 坐标系的选择	5-436
5.8.2 领域的定义	5-437
5.8.3 对象机器的选择	5-439
5.8.4 指定使用者定义领域内的举动	5-439
5.8.5 设定示例	5-439
5.9 自由平面限制	5-440
5.9.1 自由平面极限的定义	5-440
5.9.2 自由平面极限的坐标系选择	5-441
5.10 中断时的 JOG 前进后的自动回归设定	5-442
5.11 电源开启的程序自动执行方法	5-443
5.12 关于抓手种类	5-444
(1) 电磁阀的种类和信号号码	5-444
5.13 关于抓手初始状态	5-445
(1) CR750/CR751 控制器	5-445
(2) CR760 控制器	5-446
5.14 关于输出信号复位模式	5-447
5.15 关于通信设定（以太网）	5-449
5.15.1 各参数的详细内容	5-449
(1) NETIP（控制器的 IP 地址）	5-449
(2) NETMSK（子网掩码）	5-449
(3) NETPORT（端口号码）	5-449
(4) CPRCE11 ~ 19（协议）	5-450

(5) COMDEV(对应 COM1 : ~ 8 的设备定义)	5-450
(6) NETMODE (服务器指定)	5-450
(7) NETHSTIP (数据通信对象的服务器的 IP 地址)	5-450
(8) MXTTOUT (执行 realtime 外部控制指令时的超时设定)	5-450
5.15.2 参数设定示例 1 (使用支持软件时)	5-451
5.15.3 参数设定示例 2-1 (使用数据链路功能: 控制器为服务器时)	5-452
5.15.4 参数设定示例 2-2 (使用数据链路功能: 控制器为客户机时)	5-453
5.15.5 参数设定示例 3 (使用实时外部控制功能)	5-454
5.15.6 连接的确认	5-455
5.15.7 通过 Windows 的 ping 指令确认连接的方法	5-455
5.16 关于抓手、工件条件设定 (最佳加减速设定)	5-456
5.17 关于特异点接近警告	5-458
5.18 关于高速 RAM 运行功能	5-459
5.19 关于暖机运行模式	5-461
5.20 关于特异点通过功能	5-467
5.21 关于冲突检知功能	5-472
(1) 功能的概要	5-472
(2) 关连参数	5-473
(3) 使用冲突检知功能时	5-474
5.22 关于过负载 Level 的最佳化	5-479
5.23 Pallet 定义指令的多旋转限制	5-480
5.24 关于干涉回避功能	5-481
5.24.1 操作步骤	5-482
5.24.2 机器的准备和连接	5-483
5.24.3 干涉确认用模型的登录	5-484
5.24.4 自由平面极限的登录	5-492
5.24.5 附加轴的对应	5-493
5.24.6 共享存储器扩展功能的设定 (仅限机器人之间的干涉确认时)	5-494
5.24.7 机器人之间的校准 (仅限机器人之间的干涉确认时)	5-496
5.24.8 干涉回避功能的有效 / 无效设定	5-498
5.24.9 干涉回避功能的使用	5-499
5.24.10 采样程序	5-500
5.25 PLC 输入输出单元直接控制	5-502
(1) 规格	5-502
(2) 操作步骤的概要	5-502
(3) 参数的说明	5-503
(4) PLC 的“多 CPU 设定”	5-504
(5) 机器人状态变量的说明	5-504
5.26 在机器人 CPU 之间直接通信	5-506
(1) 规格	5-506
(2) 利用方法	5-506
(3) 机器人状态变量的说明	5-506
5.27 关于冗余系统错误的动作选择参数	5-507
6 外部输入输出的功能	6-508
6.1 种类	6-508
6.2 PLC 连接输入输出功能	6-509
6.2.1 参数设定	6-509
(1) PLC CPU 的参数设定	6-509
(2) 机器人 CPU 的参数设定	6-510
6.2.2 CPU 共有内存和机器人输入输出信号的对应	6-512
6.2.3 PLC 梯形图的范例	6-512
6.2.4 专用输入输出信号的分配 (工厂出货时设定)	6-514
6.3 专用输入输出	6-516
6.4 信号的有效无效状态	6-526

	页
6.5 外部信号的时序图	6-527
6.5.1 各信号的个别时序图	6-527
6.5.2 时序图例	6-536
(1) 外部信号操作时序图 (其 1)	6-536
(2) 外部信号操作时序图 (其 2)	6-537
(3) 外部信号操作时序图 (其 3)	6-538
(4) 外部信号操作时序图 (其 4)	6-539
(5) 外部信号操作时序图 (其 5)	6-540
6.6 通过外部信号选择·执行程序的方法	6-541
6.6.1 种类	6-541
6.6.2 执行方法的选择	6-541
6.6.3 相关的输入输出参数	6-541
6.6.4 操作步骤	6-542
6.7 紧急停止输入	6-544
6.7.1 紧急停止输入时的机器人的动作	6-544
7 附录	附录 -545
7.1 关于构造标志	附录 -545
7.2 多功能电动抓手的使用方法	附录 -547
7.2.1 概要	附录 -547
(1) 设备构成 (例)	附录 -547
7.2.2 规格	附录 -548
7.2.3 限制事项	附录 -548
(1) 可使用的设备和软件版本	附录 -548
(2) 站号设定	附录 -548
(3) 多功能电动抓手的驱动电源 (与机器人本体的关系)	附录 -549
7.2.4 操作注意事项	附录 -549
(1) 原点回归	附录 -549
(2) 关于在 ALWAYS (始终执行) 程序中的执行	附录 -551
7.2.5 操作步骤	附录 -552
7.2.6 参数的设定	附录 -552
(1) 执行器类型的设定	附录 -552
(2) 控制参数的设定	附录 -553
7.2.7 多功能电动抓手的操作	附录 -554
(1) 抓手号码的切换	附录 -555
(2) 原点回归	附录 -555
(3) 抓手开闭操作	附录 -556
(4) 动作位置示教	附录 -556
(5) 向示教位置移动	附录 -558
(6) 把持动作	附录 -559
(7) 动作速度的设定	附录 -560
(8) 力 (把持力) 的设定	附录 -560
7.2.8 机器人语言的规格	附录 -561
(1) 指令一览	附录 -561
(2) 状态变量一览	附录 -561
(3) 指令的详细说明	附录 -562
(4) 状态变量的详细说明	附录 -572
7.2.9 参数一览	附录 -576
7.2.10 报警一览	附录 -578
7.3 多重抓手的使用方法	附录 -580
7.3.1 概要	附录 -580
(1) 设备构成 (例)	附录 -580
(2) 可使用的机器人和软件版本	附录 -580
7.3.2 操作步骤	附录 -581
(1) 参数的设定	附录 -581
(2) 抓手开闭操作 (R32TB)	附录 -584
(3) 工具号码 (TOOL 坐标参数) 与多重抓手的关系	附录 -585

目录

	页
(4) 关于报警	附录 -586
7.4 关于样条插补	附录 -587
7.4.1 概要	附录 -587
(1) 概要	附录 -587
(2) 特征	附录 -587
(3) 必要的设备与软件版本	附录 -588
(4) 术语	附录 -588
7.4.2 规格	附录 -589
(1) 基本规格	附录 -589
(2) 限制事项	附录 -590
(3) 样条插补中的机器人的动作	附录 -590
(4) 路径点相关的检查	附录 -592
7.4.3 功能的说明	附录 -593
(1) 路径补偿	附录 -593
(2) 动作模式	附录 -596
(3) 信号输出	附录 -597
(4) 数值设定	附录 -598
(5) 帧转换	附录 -598
7.4.4 作业步骤	附录 -600
7.4.5 样条文件的创建	附录 -602
(1) 新建	附录 -602
(2) 通过 DXF 文件导入功能创建	附录 -603
(3) 用机器人语言新建	附录 -609
(4) 打开现有的样条文件	附录 -609
(5) 样条文件编辑画面的说明	附录 -611
(6) 样条文件编辑时的菜单栏	附录 -613
(7) 路径点数据的内容	附录 -615
(8) 样条文件的编辑	附录 -616
(9) 样条文件的保存	附录 -622
(10) 样条文件的删除	附录 -624
(11) 样条号的更改	附录 -624
(12) 样条文件的复制	附录 -625
(13) 样条文件管理	附录 -626
(14) 导入·导出功能	附录 -627
(15) 编辑的辅助功能	附录 -636
(16) 样条曲线的显示	附录 -638
(17) 编辑内容的确认	附录 -638
(18) 机器人程序	附录 -639
7.4.6 机器人程序的创建	附录 -642
7.4.7 动作的确认	附录 -645
7.4.8 保存至机器人控制器	附录 -645
7.4.9 调整作业	附录 -645
(1) 位置补偿功能	附录 -645
(2) 帧转换功能	附录 -647
(3) 跳位	附录 -650
(4) 参数 SPLOPTGC	附录 -651
7.5 关于 Ex-T 控制	附录 -652
7.5.1 概要	附录 -652
(1) 特征	附录 -652
(2) 规格	附录 -653
(3) 必要的设备与软件版本	附录 -653
7.5.2 Ex-T 坐标的设定	附录 -653
(1) 设定方法	附录 -653
7.5.3 Ex-T JOG	附录 -654
(1) 工件 JOG 中的姿势成分动作	附录 -655
(2) Ex-T JOG 中的姿势成分动作	附录 -655
(3) Ex-T JOG 的操作	附录 -656
(4) RV6 轴型的工件 JOG 动作	附录 -657

目录

	页
(5) RH4 轴型的工件 JOG 动作	附录 -659
(6) RH4 轴垂吊型的工件 JOG 动作	附录 -661
7.5.4 机器人程序的创建	附录 -663
(1) Ex-T 控制相关指令·变量的一览	附录 -663
(2) 编程示例	附录 -663

1 开始使用

本章说明操作说明书的内容、使用方法、基本用语及安全事项等。

另外，各操作说明书中的示教单元 (T/B) 的安装、操作方法皆是以 R32TB(R33TB) 为基本记载。使用 R56TB(R57TB) 等其它的示教单元时，请参照各别的示教单元所附的操作说明书。

1.1 操作说明书的使用方法

1.1.1 各操作说明书的内容

以下为本产品所添附的文件内容及目的等说明。请依据用途加以活用。另外，特殊规格的情况下会以分册的说明书方式加以说明特殊部份。

安全手册	为了确保机器人相关操作者的安全，本手册说明机器人的处理、系统设计及制作等共通的注意事项及安全对策。
标准规格书	说明产品的标准规格及工厂出货时的特殊规格、构成配件、保养备品等。且说明在机器人的导入的时候，安全上、技术上的注意事项。
机器人本体设定到保养	关于机器人本体、可动作前的程序（开封、搬运、设置、动作确认）及保养、检查等做说明。
控制器设定及基本操作、保养	关于控制器、可操作前的程序（开封、搬运、设置、动作确认）及程序制作到自动运行的基本操作、保养、检查等做说明。
功能和操作说明	各功能的说明及操作方法、写程序所使用的 MELFA-BASIC V 指令的说明、与外部输入输出机器的接续方法、参数的说明、功能操作的详细说明等。
障碍排除	说明当报警发生时，报警号码所代表的原因及对策。
附加轴 I/F 功能	说明在机器人本体附加行走轴等使用的情况下，伺服放大器的控制功能。
跟踪功能说明书	说明传送带跟踪的规格、功能、使用方法。
扩展功能说明书	对 iQ Platform 支持的机器人中的可编程控制器，或独立设备型机器人中的 GOT 与机器人之间的存储器的数据构成及监视、操作步骤进行详细说明。

1.1.2 操作说明书中的符号

说明书中符号所表示的意义如表 1-1 中所示。

请确认后再阅读说明书。

表 1-1: 操作说明书中的记号

区分	用语 / 记号	意义
用语	iQ Platform	
	控制器	是对机器人本体进行控制的控制器。 是机器人 CPU 系统、驱动单元所构成的控制部的总称。
	机器人 CPU 单元 或 机器人 CPU	是安装在三菱公司 MELSEC-Q 系列的 PLC 基板 (Q3 □ DB) 的插槽上的机器人用 CPU 单元。与驱动单元通过专用电缆连接。
	机器人 CPU 系统	多 CPU 系统。 由 PLC 基板、PLC CPU 单元、机器人 PLC 单元、其他 MELSEC 单元所构成。
	驱动单元	是搭载了机器人用伺服放大器、安全回路等的机箱。
用语	独立设备型	
	控制器	是对机器人本体进行控制的控制器。 是将机器人 CPU、伺服放大器、安全回路等的控制部集中在一起的机箱。
记号	 危险	假设在错误处理的情况下, 会导致使用者有死亡或重伤的危险的注意事项。 为了让机器人安全的被使用, 请务必注意危险符号。
	 警告	假设在错误处理的情况下, 会导致使用者有死亡或重伤的危险的注意事项。 为了让机器人安全的被使用, 请务必注意危险符号。
	 注意	假设在错误处理的情况下, 会导致使用者有受伤或物品损坏的情形 的注意事项。为了让机器人安全的被使用, 请务必注意危险符号
	例) [JOG]	有加上 [] 符号的表示控制器前面操作参数的开关以及示教单元上的按键。
	例) [RESET] + [EXE] (A) (B)	表示在保持按住 (A) 键后再按下 (B) 键。 此例子为保持按住 [RESET] 键后再按下 [EXE] 键。
	T/B	表示为示教单元。 本书是以 R33TB 为基础说明。
	O/P	显示为驱动器元或机器人控制器前面的操作面板。
	CR751 (薄型) CR751 (厚型)	CR751 控制器有薄型 (高 98mm) 和厚型 (高 174mm) 两种。 薄型: CR751-03HD/Q、CR751-06HD/Q、CR751-12HD/Q、CR751-20HD/Q、 CR751-03HRD/Q、CR751-02VD/Q、CR751-04VD/Q、CR751-04VJD/ Q、CR751-07VD/Q 厚型: CR751-13VD/Q、CR751-20VD/Q、CR751-07VLD/Q ※CR751 控制器的外形图, 请参照另一手册“标准规格书”。

1.2 安全上的注意

使用机器人前请务必熟读以下的注意事项及别册「安全手册」，了解如何使用机器人，以确保您及周遭环境的安全。

1.2.1 确认使用的方法能遵守法律的规则。



注意

示教作业等因安全的考虑，必须由受过操作教育训练的人员操作使用。
(无切断电力的保养作业亦相同)
→安全教育的实施



注意

请将示教作业依机器人的操作方法及顺序、异常时及再启动时处理等制作相关作业规程、并遵守规章内容。
(无切断电力的保养作业亦相同)
→作业规程的制作



警告

示教作业请设定为可立即停止运行的装置。
(无切断电力的保养作业亦相同)
→紧急停止开关的设定



注意

示教作业中请将[示教作业中]的标示放置在起动开关上。
(无切断电力的保养作业亦相同)
→示教作业中的表示



危险

运行中请确认使用围离或栅栏将操作人员与机器人做隔离，防止直接接触机台。
→安全栅栏的设置



注意

运行开始，对于相关人员的信号有固定的方法，请依此进行。
→运行开始的信号



注意

维护作业原则上请中断电力进行、请将[保养作业中]的标语放置在起动开关上。
→维护作业中的表示



注意

作业开始前请详细的检查，确认机器人及紧急停止开关、相关装置等无异常状况。
→作业开始前的检查

1.2.2 安全手册注意事项

本节摘录别册 [安全操作手册] 中的重点提示。详细的内容请阅读 [安全操作手册]。

危险

使用复数的控制机器（GOT、PLC、按钮开关）执行机器人自动运行的情况下，各机器的操作权等的互锁请客户端自行设计。

注意

请在规格范围内的环境中使用机器人。除此之外的环境是容易造成机器人的故障原因。（温度、湿度、空气、噪声环境等）

注意

请依照机器人指定的搬运姿势进行搬运或移动机器人。指定以外的搬运方式有可能因为掉落而造成人身安全或机器人故障。

注意

请确实将机器人固定在底座上。不稳定的姿势有可能产生位置偏移或发生振动。

注意

电线是产生噪声的原因，请尽可能将配线拉开距离。太过接近有可能造成位置偏移及错误动作。

注意

请勿用力拉扯接头或过度的卷屈电线。因有可能造成接触不良及电线断裂的情况。

注意

抓手所包含的工件重量请勿超出额定负载及最大力矩。超出重量的情况下有可能发生报警及故障。

警告

请确实紧抓住抓手、安装工具及夹放的工件。以免因运行中使物件甩开而导致人员及物品的损伤。

警告

机器人及控制器的接地请确认连接。否则容易因为噪声而做错误动作或导致触电事故发生。

注意

机器人在动作中时请标示为运行状态。没有标示的情况下容易导致人员接近或有错误的操作。

警告

在机器人的动作范围内做示教作业时，请务必确保机器人的控制有优先权。否则由外部指令使机器人起动，有可能造成人员及物品的损伤。

注意

JOG 速度请尽量以低速进行、并请勿在操作中将视线离开机器人。否则容易干涉到工件及周边装置。

注意

程序编辑后的自动运行前，请务必确认 **step** 运行动作。若无确认有可能发生程序错误而与周边装置干涉。

-  **注意** 自动运行中安全栅栏的出入口门打开被锁住的情况下，机器人会自动停止。否则会发生人员的损伤。
-  **注意** 请勿因个人意思做机械的改造及使用指定以外的零件。否则可能导致机械故障或损坏。
-  **警告** 从外部用手将机器人手臂做推动的情况下，请勿将手或指头放入开口部位。有可能会夹伤手或指头。
-  **注意** 请勿用将机器人控制器的主电源关闭的方式来使机器人停止或紧急停止。在自动运行中将控制器的主电源关闭有可能使机器人精度受到影响。且，有可能发生手臂掉落或松动而干涉到周边装置的情况。
-  **注意** 重写控制器内程序或参数等内部数据时，请勿将控制器的主电源关闭。自动运行中或程序参数添写中时，若关闭控制器主电源，则有可能破坏控制器的内部数据。
-  **危险** 使用本产品的 GOT 直接连接功能的情况下，请勿连接手编 GOT。手编 GOT 会使机器人自动运行，无论操作权的有效 / 无效，可能发生物品及人员损伤事故。
-  **危险** 在 CR750-Q/CR751-Q/CR760-Q 使用对应 iQ Platform 产品的情况下，请勿在 PLC 上连接手编 GOT。手编 GOT 会使机器人自动运行，无论操作权的有效 / 无效，可能发生物品及人员损伤事故。
-  **危险** 多重 CPU 系统及伺服放大器的电源投入时，请勿拔除 SSCNET III 的接线。请勿直视 MOTION CPU 及伺服驱动器的 SSCNET III 接头及 SSCNET III 电线前端所发出的光。光线直射眼睛，会使眼睛产生失调感。（SSCNET III 的光源相当于 J1SC6802、IEC60825-1 所规定的等级 1。）
-  **危险** 机器人控制器的电源投入时，请勿拔除 SSCNET III 的接线。请勿直视 SSCNET III 接头及 SSCNET III 电线前端所发出的光。光线直射眼睛，会使眼睛产生失调感。（SSCNET III 的光源相当于 J1SC6802、IEC60825-1 所规定的等级 1。）
-  **危险** SSCNET III 接线取下后，若没有在 SSCNET III 接头部位装上护套的话，会使接头附着灰尘及杂物。会使机器人特性变差、发生错误动作等。
-  **注意** 应注意不要出现配线错误。进行了不符合规格的连接的情况下，有可能导致紧急停止无法解除等的误动作。为了防止误动作，配线完毕后，必须对机器人控制器操作面板、示教单元紧急停止、用户紧急停止、门开关等的各种功能是否正常动作进行确认。



将控制器的 USB 与市面销售的设备（笔记本电脑、计算机、LAN 用集线器等）连接使用时，有可能与本公司的设备的匹配性及温度・噪声等的 FA 环境不符合。使用时，有时需要采取 EMI 对策（Electro-Magnetic Interference）或添加铁氧体磁芯等其它对策，用户应进行充分的动作确认。此外，对于与市面销售设备连接时的动作保障・维护等本公司将不予承担。

2 功能的说明

2.1 操作面板 (O/P) 的功能 (CR750控制器)

(1) 操作面板的按钮说明

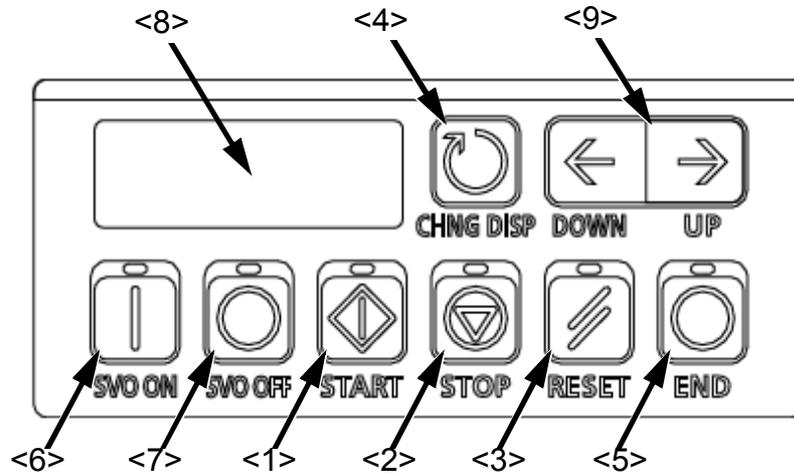


图 2-1: 操作面板各部位的名称 (控制器)

- <1> 开始按钮 执行程序使机器人动作。程序连续运行。
- <2> 停止按钮 立刻使机器人停止。伺服不会关闭。
- <3> RESET 按钮 解除报警。解除程序中中断状态并复位程序。
- <4> 显示切换按钮 按 <速度比例> → <行号码> → <程序号码> → <使用者情报> → <制造商情报> 的顺序切换显示面板的显示内容。
- <5>END 按钮 将执行中的程序停止在最终行或 End 语句。
- <6>SVO.ON 按钮 输入伺服电源。(SERVO ON)
- <7>SVO.OFF 按钮 切断伺服电源。(SERVO OFF。)
- <8>STATUS. 编号 (显示面板)
显示报警号码、程序号码、速度比例值 (%) 等。
- <9>UP/DOWN 按钮 在显示面板的内容显示上页、下页。

(2) STATUS 编号 (显示面板) 的说明

程序名中使用了字母的情况下，在操作面板的 7 段 LED 中显示时的简易显示记号如下所示。

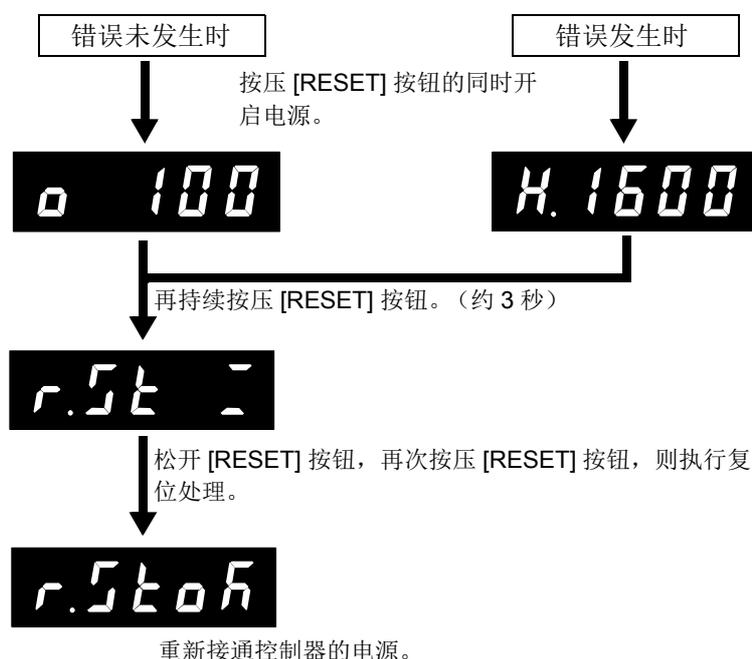
A	R	B	b	C	c	D	d	E	E	F	F	G	5	H	H	I	,
J	u	K	5	L	L	M	n	N	n	O	o	P	P	Q	q	R	r
S	5	T	t	U	u	V	u	W		X		Y	4	Z	2	?	-

显示程序名时，起始处会带有字母 "P"，因此最多可显示 4 个字符。因此，程序名应在 4 个字符以内。

程序名超过 4 个字符时，无法通过操作面板进行选择。对于通过机器人语言的 CallP 指令执行的子程序，其程序名可以超过 4 个字符。

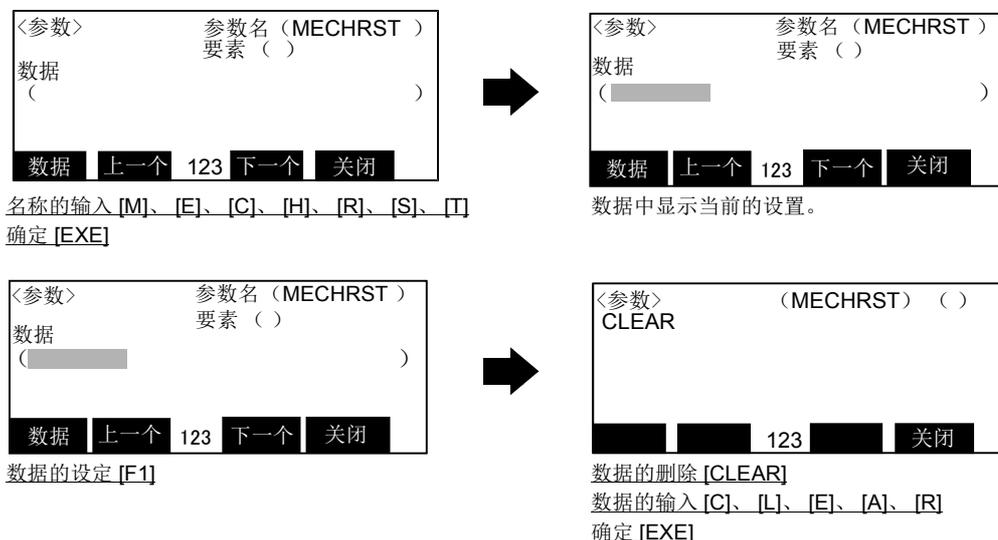
(3) 机型复位操作的说明

如因电池的消耗等导致机型信息消失，则会发生错误（H1600：机械未设置 等），操作将无法进行。该情况下，如进行以下机型的复位操作，则会变为出厂状态，可设置正确的机型信息。



控制器的软件版本为 R5q(F-Q 系列)/S5q(F-D 系列) 以上时，可通过示教单元或 RT ToolBox2 进行机型复位操作。通过示教单元进行机型复位操作的方法如下所示。

1) 在参数 MECHRST 中设置 CLEAR。



2) 重新接通控制器的电源。

⚠ 注意

如进行本操作，则会变为出厂状态，参数、程序和各種日志数据会被删除，因此除发生错误（H1600）以外，请勿实施本操作。被删除的这些数据中如有 RT ToolBox / RT ToolBox2 (选配件) 的批量备份数据，则可通过恢复功能进行复原。

2.2 操作面板（O/P）的功能（CR760控制器）

(1) 操作面板的按钮说明

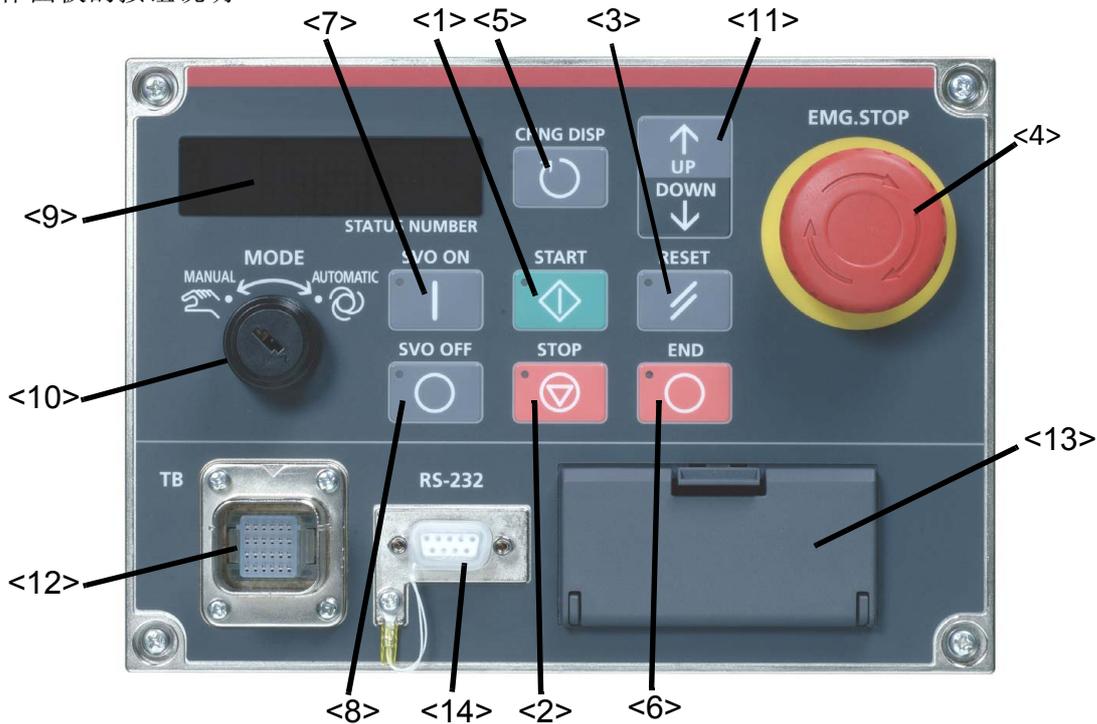
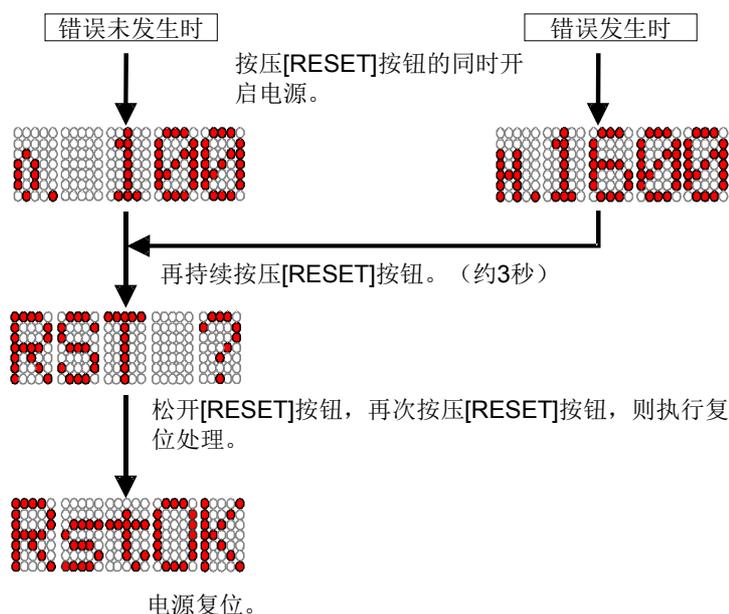


图 2-2: 操作面板各部位的名称（控制器）

- <1> 开始按钮 执行程序使机器人动作。程序连续运行。
- <2> 停止按钮 立刻使机器人停止。伺服不会 OFF。
- <3> 复位按钮 解除错误。此外，解除程序中断状态并复位程序。
- <4> 紧急停止开关 使机器人紧急停止。伺服 OFF。
- <5> 显示切换按钮 按 < 倍率修调 > → < 行号 > → < 程序号 > → < 用户信息 > → < 生产厂商信息 > 的顺序切换显示面板的显示内容。
- <6> 结束按钮 停止在执行中的程序的最终行或 End 语句。
- <7>SVO.ON 按钮 接通伺服电源。（伺服 ON）
- <8>SVO.OFF 按钮 切断伺服电源。（伺服 OFF）
- <9>STATUS.NUMBER(显示面板)
显示报警号码、程序号码、倍率修调值（%）等。
- <10> 模式切换开关 对机器人的操作权进行切换。
AUTOMATIC 操作面板或外部设备的操作有效。无法进行需要示教单元操作权的操作。操作面板与外部设备的连接中，需要对操作权用的参数进行设定。详细内容请参照“第 13 页的「2.3.1 关于操作权」”。
- MANUAL 示教单元有效时，仅限通过示教单元进行的操作有效。不能进行需要外部设备及操作面板的操作权的操作。
- <11>UP/DOWN 按钮 使显示面板的显示内容进行上翻页、下翻页显示。
- <12> 示教单元连接用连接器 . 连接示教单元的专用连接器。不使用示教单元时，连接附带的虚拟连接器。
- <13> 接口盖板 盖板内部配备有 USB 接口和电池。
- <14>RS-232 连接器 连接个人计算机的 RS-232 规格的连接器的。
CR760-Q 控制器中无此配备。

(2) 机型复位操作的说明

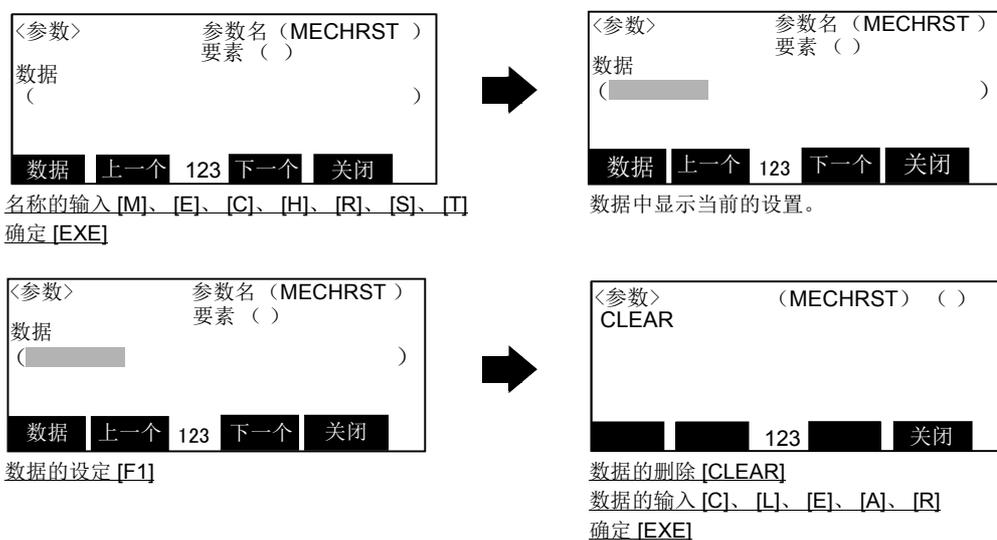
如因电池的消耗等导致机型信息消失，则会发生错误（H1600：机械未设置 等），操作将无法进行。该情况下，如进行以下机型的复位操作，则会变为出厂状态，可设置正确的机型信息。



通过示教单元或 RT ToolBox2 可进行机型复位操作。

通过示教单元进行机型复位操作的方法如下所示。

1) 在参数 MECHRST 中设置 CLEAR。



2) 重新接通控制器的电源。



注意

如进行本操作，则会变为出厂状态，参数、程序和各種日志数据会被删除，因此除发生错误（H1600）以外，请勿实施本操作。被删除的这些数据中如有 RT ToolBox / RT Toolbox2 (选配件) 的批量备份数据，则可通过恢复功能进行复原。

2.3 示教单元的 (T/B) 的功能

本章记述 R32TB/R33TB (配件) 的功能。

(1) 各按键的功能

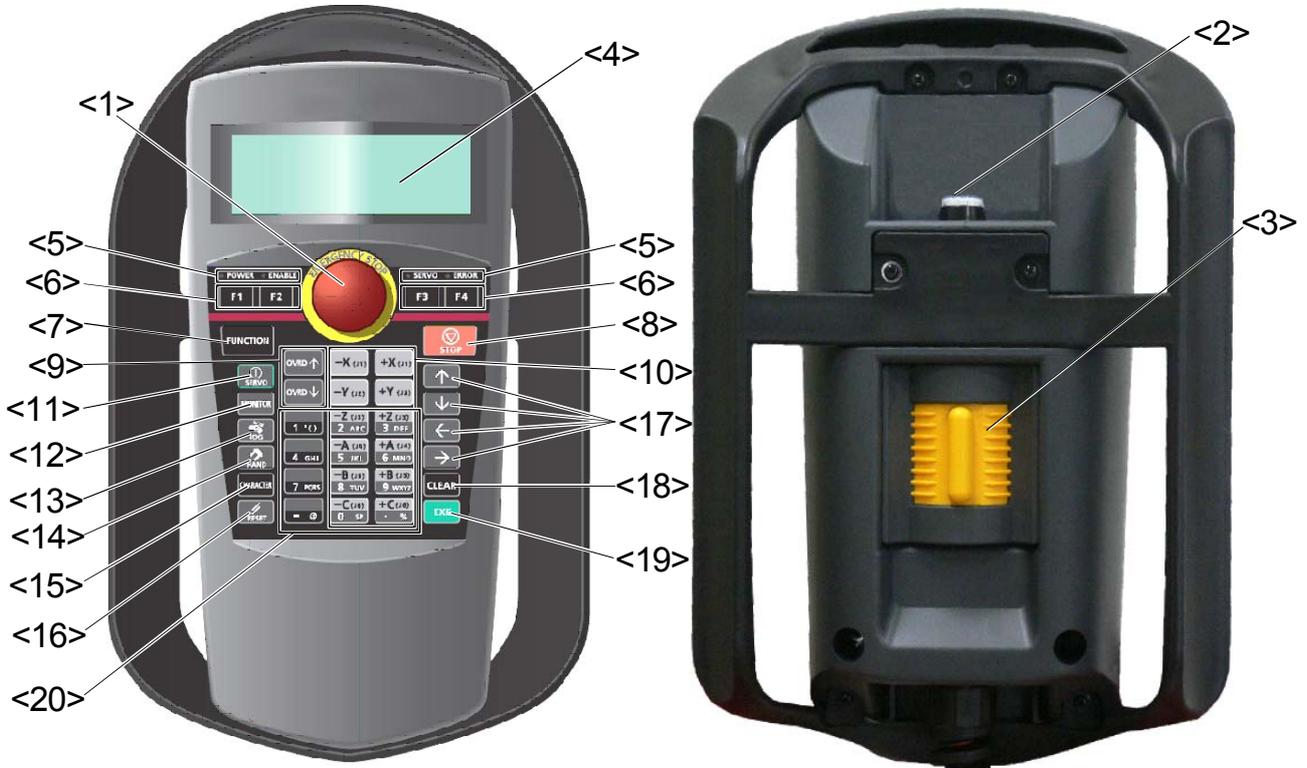


图 2-3: T/B 外观图

<1>[EMG.STOP] 开关

是一个有上锁功能的按钮式开关，紧急停止时使用。按下此钮则伺服 OFF，且无论示教单元在有效 / 无效状态，机器人都会立刻停止。

要取消此状态请将开关往右边顺时针旋转，或往自己的方向拉起。（紧急停止开关按下后，机器人会变成报警状态。开关解除后，请进行报警的复位）

<2>[TB ENABLE] 开关

为切换示教单元的按键操作为有效或无效的开关。此开关为交替的开关，示教单元有效时，开关内的灯会亮起，且，前面的 ENABLE 灯也会亮起。使用示教单元操作机器人时，务必使示教单元为有效。

示教单元有效时，示教单元的操作有优先权，无法由控制器或其它外部控制。若要以控制器或其它外部控制，请将示教单元设定为无效“DISABLE”。

<3>使能开关 (3位置开关)

在背面为3位置开关，MANUAL模式时，放开此开关或强按下此开关，则会使伺服关闭。JOG操作和单步操作等在伺服开启状态下操作的功能，请在轻押此开关下进行。

另外，执行紧急停止及伺服关闭操作时，伺服是关闭状态，只按下本开关，伺服也不会启动。请重新执行开启伺服。

<4>显示面板

显示以示教单元按键所做的程序内容或机器人状态。

<5>显示状态灯

显示示教单元及机器人的状态。

[POWER]: 示教单元有电源供给时，绿色灯亮起。

[ENABLE]: 示教单元为有效状态时，绿色灯亮起。

[SERVO]: 机器人在伺服开启中时，绿色灯亮起。

[ERROR]: 机器人在报警状态时，红色灯亮起。

<6>[F1][F2][F3][F4] 键

执行显示面板的功能显示部显示的功能。

<7>[FUNCTION] 键

在 1 个的操作、在 [F1][F2][F3][F4] 键分配功能有 5 个以上时，按下此键可以切换功能显示，在 [F1][F2][F3][F4] 键变更分配功能。在画面的最下段有用反白字显示的 MENU。

这些 [F1][F2][F3][F4] 键以左到右的顺序分配。按下对应的功能键，则可以选择显示的 MENU。

另外，MENU 的右端有显示 "⇒" 的情况下，现在的显示也还有显示其它的 MENU，按下 [FUNCTION] 键，可以切换显示 MENU。

<8>[STOP] 键

中断运行中的程序，使移动中的机器人减速停止。另外，有程序在执行中的话，会中断执行。

和控制器前面的 [STOP] 按键功能相同，有连接示教单元的状态，没有按下 [使能] 开关的情况下，（[ENABLE] 灯没有亮起的情况）也可以使用。

<9>[OVRD ↑][OVRD ↓] 键

可改变机器人的速度比例。按下 [OVRD ↑] 键，则速度比例增加。按下 [OVRD ↓] 键，则速度比例减少。以此键操作的速度比例变更，其变化也会显示在控制器前面的速度比例。

<10>[JOG 操作] 键，从 [-X(J1)] ~ [+C(J6)] 的 12 个键] 示教单元

在 JOG 模式时，可以此键执行 JOG 操作。示教单元

在抓手模式时，可以此键执行抓手操作。

<11>[SERVO]键

一边轻轻握住（3位置开关）按键，一边按下此键，则机器人会开启伺服。

<12>[MONITOR]键

按下此键，会变成接口模式，显示接口MENU。在接口模式时按下此键，则会回到接口模式前的画面。

<13>[JOG]键

按下此键，会变成JOG模式，显示JOG画面。在JOG画面按下此键，会回到JOG模式前的画面。

<14>[HAHD]键

按下此键，会变成抓手模式，显示抓手操作画面。在抓手画面按下此键，会回到抓手操作模式前的画面。

另外，按下此键2秒以上时，会变成TOOL选择画面，可进行TOOL数据选择，变更模式。在TOOL选择模式时，按下此键2秒以上，会回到前一个画面。

<15>[CHARACTER]键示教单元

在可以输入文字或数字的时候，使用[数字/文字]键的功能，切换数字及文字间的输入。

<16>[RESET]键

机器人在报警状态时，可解除报警。（也有无法解除的报警）

另外，一边按下此键，一边按下[EXE]键，则会执行程序复位。

<17>[↑][↓][←][→]键

将移标移动到各个方向。

<18>[CLEAR]键

可输入数字或文字时，按下此键的话，可删除光标上的1个文字。另外，长时间按住时，会删除光标输入范围的所有文字。

<19>[EXE]键

确定输入操作。

此外，直接执行时，继续按住此键时，机器人会动作。

<20>[数字 / 文字]键

可输入数字或文字时，按下此键会显示数字或文字。

2.3.1 关于操作权

即使在控制器连接示教单元及计算机等复数的机器，但同时控制器上的操作（运行、开启伺服等的指令），只限 1 个为有效机器。此限制的机器就称为「取得操作权」。

程序起动和开启伺服等将机器人起动、或可起动的因素，没有取得操作权的话则无法操作。相反的，停止和伺服关闭等使机器人停止的操作，为了安全上的操作，即使没有操作权，也可以执行。

表 2-1: 设定开关和操作权的关系

○：有操作权 ×：无操作权

设定开关	示教单元 [ENABLE]	无效 ^{注1)}		有效	
	控制器的模式	AUTOMATIC	MANUAL	AUTOMATIC	MANUAL
操作权	示教单元	×	×	× ^{注2)}	○
	控制器操作面板	○ ^{注3)}	×	× ^{注2)}	×
	计算机	○ ^{注3)}	×	× ^{注2)}	×
	外部信号	○ ^{注3)}	×	× ^{注2)}	×

注 1) “AUTOMATIC” 模式时在示教单元中显示 < 操作面板 > 画面时，操作权转移到示教单元。（示教单元的状态显示灯 [ENABLE] 闪烁）

请注意，在此期间无法进行需要其他操作机器操作权的操作。

注 2) 示教单元在 "有效" 状态下，控制器的模式设为 "AUTOMATIC" 的话，会发生报警 "5000"。

注 3) 根据操作权输入信号：IOENA 的设定可变更有效机器。

表 2-2: 必须有操作权的操作

操作权：○ = 操作权必要 × = 不要操作权

区分	操作权	操作
运行 ^{注1)}	○	伺服开启
	×	伺服关闭
	○	程序起动。除了控制器操作面板，通过示教单元的操作也可起动。
	×	程序停止 / 循环停止
	○	程序复位
	×	报警复位
	○	变更速度比例。但是使用示教单元可以随时变更。
	×	读取速度比例
	○	程序号码变更
信号 ^{注2)}	×	读出输入输出信号
	×	写入输出信号
	○	专用输入起动 / 复位 / 伺服开启 / 刹车 / 切换手动模式 / 通用输出复位 / 指定程序号码 / 指定行号码 / 指定速度比例
	×	专用输入停止 / 伺服关闭 / 连续、循环 / 操作权输入信号 / 要求输出程序号码 / 要求输出行号码 / 要求输出速度比例要求报警号码 / 数值输入
	×	读取抓手输出信号输入输出
	○	写入抓手输出信号
程序编辑 ^{注3)}	○	行登录 / 读出 / 呼叫 / 位置追加 / 补偿 / 读取、变量写入 / 读出
	○	STEP 前进 / 后退、执行
	○	STEP UP/DOWN
	○	STEP 跳过、直接执行、JOG
文件操作	×	读出程序一览 / 保护设定 / 复制 / 删除 / 变更名称 / 初始化
维修操作	×	读出参数、时间设定 / 读出、操作持间计算读出、报警历史读出
	○	原点设定、参数变更

注 1) 通过示教单元的 < 操作面板 > 操作时，操作权根据控制器的模式如下所示。

- "MANUAL"..... 需要按下示教单元的 [TB ENABLE] 将其设为有效。
- "AUTOMATIC"..... 即使不按示教单元的 [TB ENABLE]，在 < 操作面板 > 画面显示中，操作权也会自动转移到示教单元。（示教单元的状态显示灯 [ENABLE] 闪烁）

注 2) 通过示教单元显示 < 操作面板 > 画面过程中，通过 [MONITOR] 按键进行的操作无效。

注 3) 一台机器在线编辑中时，无法通过其他机器进行编辑。

2.4 动作及控制相关的功能

在本控制器里有如表 2-3 中所示特征功能。

表 2-3: 有特征的功能

功能	说明	关连页次
最佳速度控制	此功能指当机器人在 2 点间移动时, 可以依机器人的姿势, 在必须保持姿势的情况下, 在轨迹途中控制速度。防止因速度过大产生报警。但是当此功能为有效时, 机器人的顶端速度不固定。	请参照第 268 页的「Spd (Speed)」。
最佳加减速控制	依照机器人开始动作及停止时的加减速时间、抓手的重量及重心, 设定工件的有无等, 自动设定最佳的加减速时间的功能。与提高节拍时间相关连, 依条件也有可能使节拍时间降低。	请参照第 250 页的「Oadl (Optimal Acceleration)」, 及第 231 页的「LoadSet (Load Set)」。
直交伺服弹性	在原来由伺服来的回馈数据, 柔和控制机器人的功能。在工件的镶合上有效的功能。可沿着机器人直交座标学习。但是, 依工件的条件, 也有无法使用的情况。	请参照第 180 页的「Cmp Tool」。
冲突检知	机器人的 TOOL 和手臂和周边机器干涉的情况下, 会紧急停止, 将 TOOL 和手臂的损害降到最小。在自动运行及 JOG 运行时也有效的功能。(只对 RV-SQ/SD 系列。除了部份特殊规格) 注) 请注意本功能无法同时与多重机器控制使用。	请参照第 186 页的「ColChk (Col Check)」, 第 407 页的「动作参数」。
干涉回避功能	本功能是在 CR7xx-Q 系列控制器中, 利用机器人 CPU 间直接通信进行 2 台或 3 台机器人间的干涉确认的同时使机器人动作的功能。 在 JOG 操作或自动运行时, 通过事先检测出机器人之间的干涉并使其停止, 可以降低对机器人的损伤。	请参照第 481 页的「关于干涉回避功能」。
维修预报	以机器人的动作状态为基础, 预测机器人的电池、皮带、润滑油等维修情报。 此功能在计算机支持软件配件里, 可以确认维修情报。 (只对 RV-S 系列。除了部份特殊规格) 注) 请注意本功能无法同时与多重机器控制使用。	请利用配件: 计算机支持软件。
支持位置回归	关节轴的偏移、更换电机、抓手变形、机器人的底座偏移的情况发生时, 依最大可补偿 10 点计算原点位置及 TOOL、底座的补偿值, 进行补偿的功能。此功能请在配件的计算机支持软件中进行。可在垂直多关节机器人中使用。(除了部份特殊规格以外)	请利用配件: 计算机支持软件。
连续轨迹控制	可在复数的位置抑制连续的加减速, 使机器人动作的功能。 有效提升节拍时间。	请参照第 103 页的「(4) 连续动作」、第 184 页的「Cnt (Continuous)」。
多重任务程序运行	可将机器人的动作程序及与外部机器通信的程序等分割并列执行。有效的缩短输出处理。另外, 依据周边夹具的控制用程序的制作, 可构筑不需要使用 PLC 的系统。	请参照第 126 页的「4.3.1 所谓多重任务处理功能」及第 283 页的「XRun (x Run)」等的 X*** 指令。
程序的持续执行功能	从控制器的电源开启时, 即执行程序。 此功能利用多重任务功能, 可在机器人程序里代替 PLC 执行。	请参照第 427 页的「5.4 程序参数」的「SLTn」参数的起动属性 (ALWAYS)。
继续功能	将电源关闭时的状态储存, 可在下次开启电源时, 由电源关闭时的状态连接开启。	请参照第 427 页的「5.4 程序参数」的「CIN」参数。
附加轴控制	可做机器人的附加轴控制, 最多 2 轴。 因可记忆机器人示教数据中附加轴的位置, 因此可以完全同时控制。另外, 可以边移动附加轴 (行走轴), 边执行圆弧插补。	请参照别册: 「附加轴 I/F 操作说明书」。
多重机器控制	在标准的机器人以外, 依伺服电机所组成的机器人使用者机器, 最大为 2 台。(除了标准机器人外)	请参照别册: 「附加轴 I/F 操作说明书」。
外部机器通信功能	和外部机器的通信, 有下列的方法: 控制器的控制及程序内的互锁功能 1) 通过输入输出信号的方法 (CR7xx-Q: PLC 链接输入输出: 8192/8192 点) (CR7xx-D: 并行输入输出: 256/256 点 (最大)) 2) 与外部机器的数据链路功能 (※ 仅限 CR7xx-D) 3) 通过以太网的通信 参考: 所谓数据链路, 是指与外部机器 (视觉传感器等) 收发补偿量等数据的功能。	请参照第 314 页的「M_In/M_Inb/M_In8/M_Inw/M_In16」、第 324 页的「M_Out/M_Outb/M_Out8/M_Outw/M_Out16」变量。

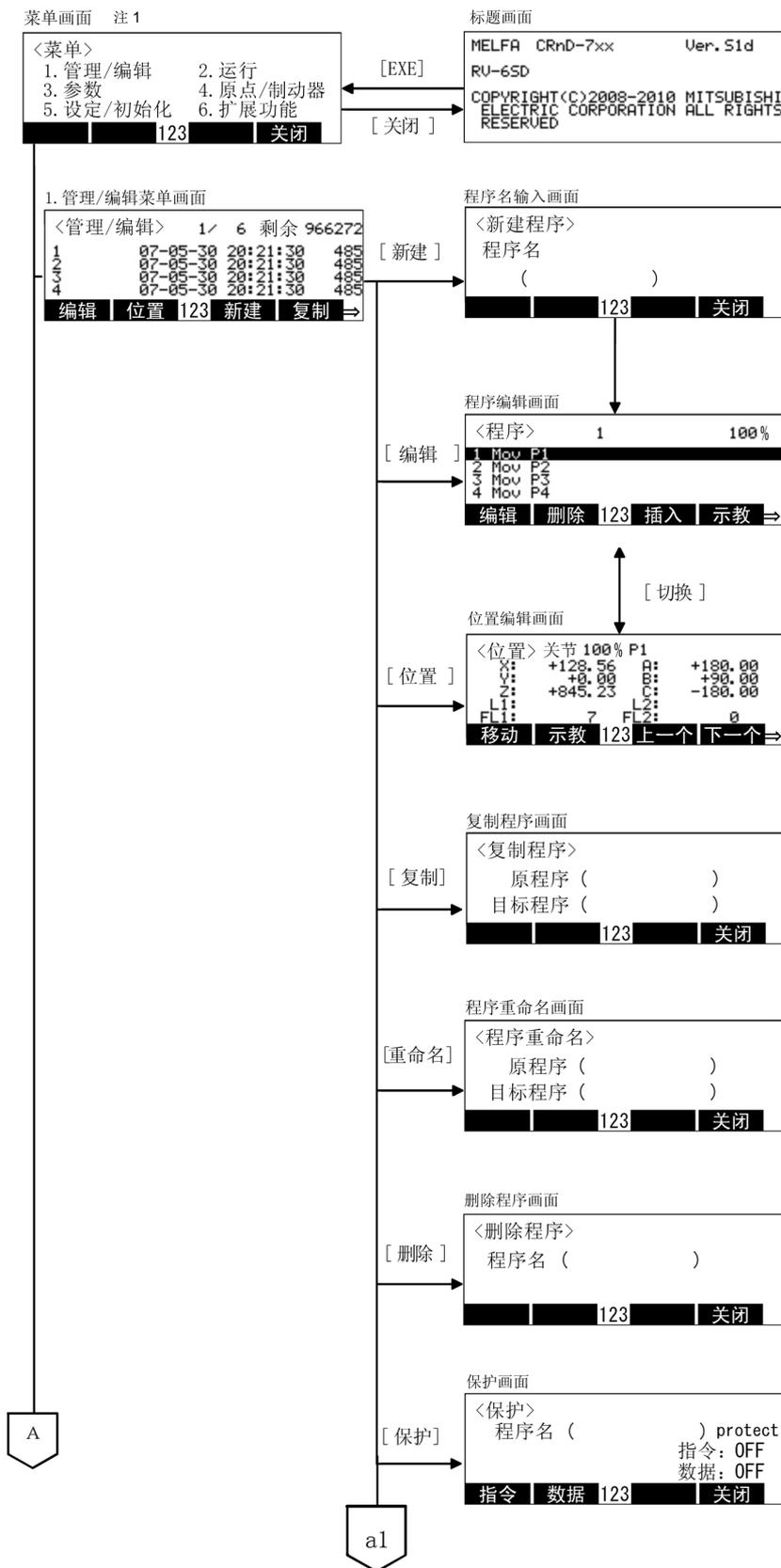
功能	说明	关连页次
中断监控	可以在程序动作中，监视信号等条件一致的情况下，将现在的处理中断，插入处理。对工件搬运时工件掉落等有效。	请参照第 190 页的「Def Act」、第 165 页的「Act」。
程序间跳转	使用 Call 指令将程序由程序里呼叫出来。	请参照第 170 页的「Call P」。
Pallet 运算	可以依据运算，求得配置在格子状里的工作的件置及 cassette 内的玻璃基板的位置。有助于示教量的轻减。可能是列 X 行形式、1 行形式、圆弧形式。	请参照第 109 页的「4.1.2Pallet 运算」第 198 页的「Def Plt (Define 码垛)」、第 256 页的「Plt (码垛)」。
使用者定义领域	可以指定最多 32 个区域的任意空间，在其领域内用 realtime 可监视 ROBOT 的尖端是否有进入，可以将其装态输出到外部、以程序示教、及发出报警。此外，在 ROBOT 程序内也拥有相同的功能的函数 (Zone、Zone2、Zone3)。	请参照第 435 页的「5.8 关于使用者定义领域」、第 339 页的「M_Uar」、第 340 页的「M_Uar32」变量。 请参照第 404 页的「Zone」、第 405 页的「Zone2」、第 405 页的「Zone3」。
关节动作范围 直交动作范围 自由平面限制	机器人的动作范围可以用 3 种方法限制。 关节动作范围：可限制各轴的动作范围。 直交动作范围：可在 ROBOT 的直交坐标限制动作范围。 自由平面限制：定义任意的平面，依据平面的前面或后面，限制其动作范围。	请参照第 407 页的「5.1 动作参数」的「MEJAR」、「MEPAR」参数。 请参照第 440 页的「5.9 自由平面限制」。

3 操作方法的说明

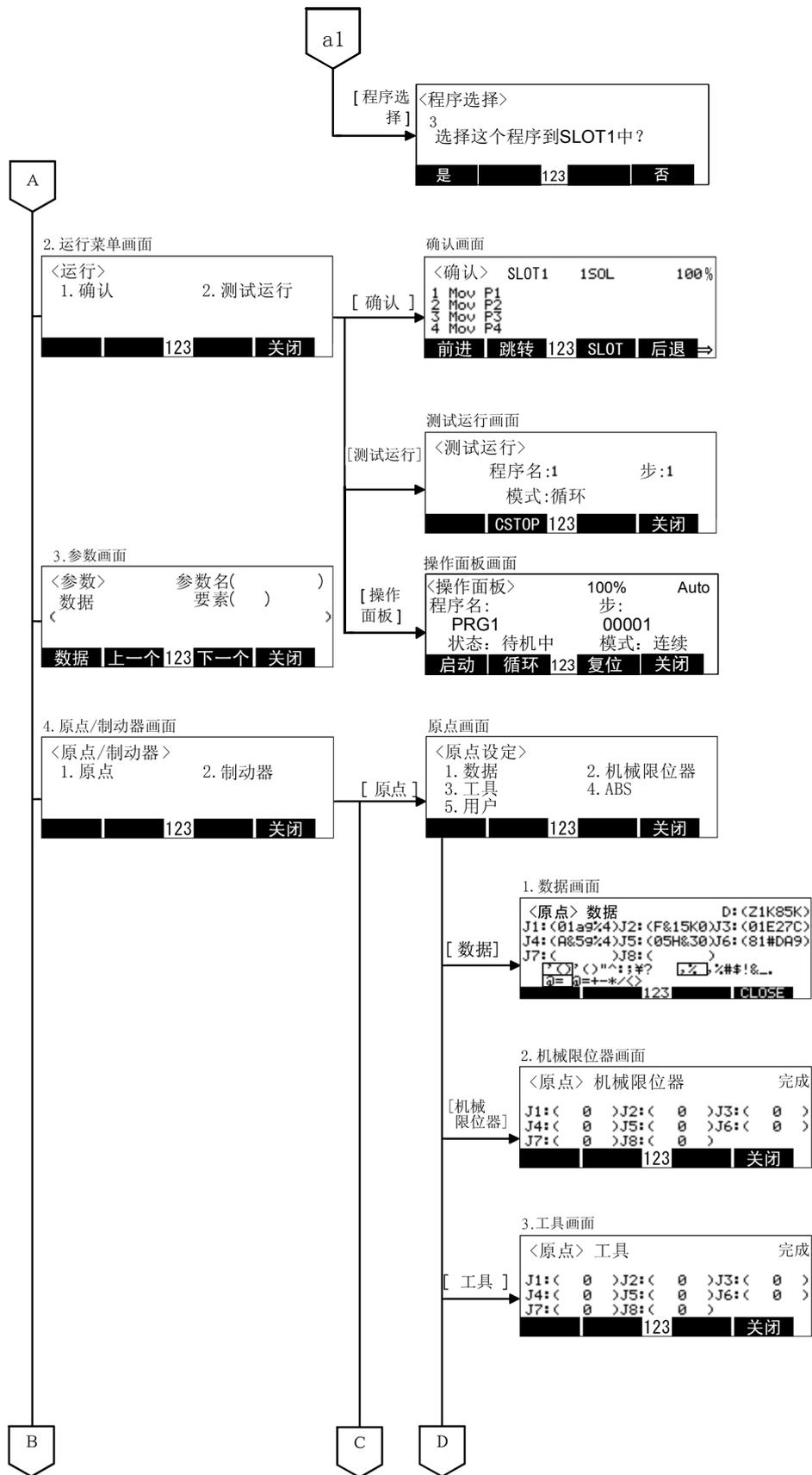
记述 R32TB（配件）的操作方法。

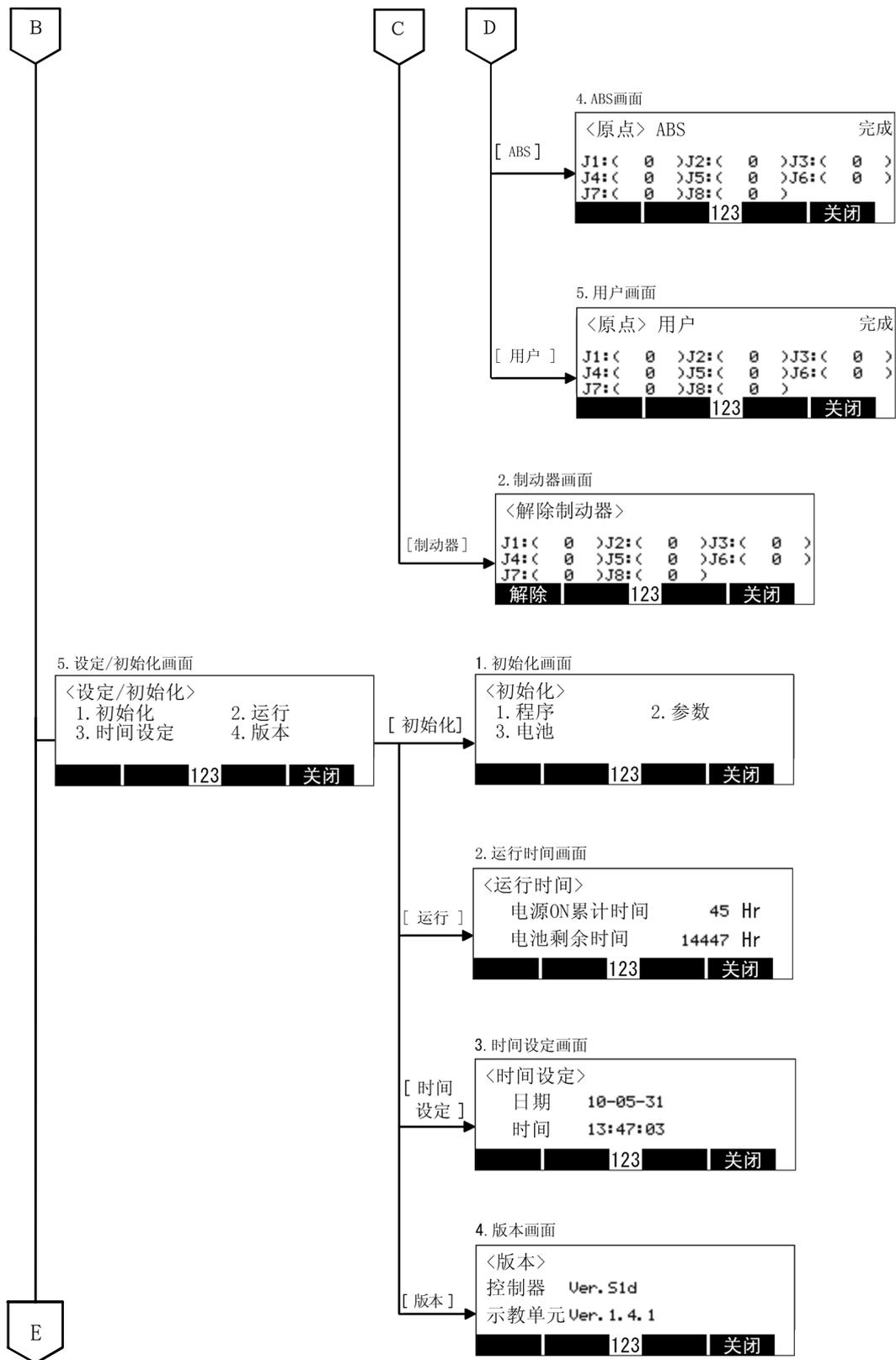
3.1 示教单元的菜单画面操作

(1) 画面树形图



注 1) 关于 6. 扩展功能中的“1. 可编程控制器直接”，请参照另一手册「使用说明书 /CR750-Q/CR751Q 系列、CRnQ-700 系列 iQ Platform 对应扩展功能说明书 (BFP-A8757)」。





(2) 数字 / 文字的输入

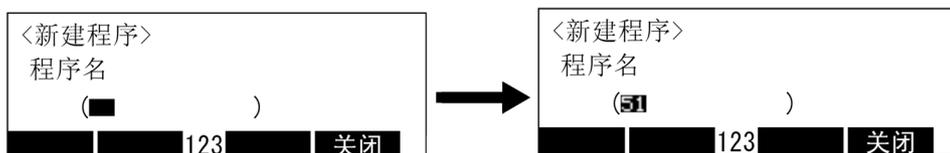
每按下一次 [CHARACTER] 键，会切换数字输入模式及文字输入模式。画面右下角会显示现在的输入模式，“123”表示为数字输入模式、“ABC”表示为文字输入模式。

1) 数字的输入

按下各数字（“-”（负号）、“.”（包含小数点））键左下角显示的键即可输入。

按下 [CHARACTER] 键，在画面右下角会显示“123”状态下，按下数字键输入。

例）输入程序名为“51”的情况。



数字的输入 [CHARACTER] [5] [1]

2) 文字的输入

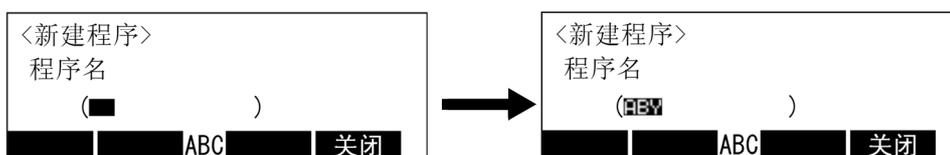
在各键的右下角有显示文字。按下此键即可以输入文字。

按下 [CHARACTER] 键，在画面右下角显示为“ABC”状态下输入文字。按键上有复数的文字显示的情况下，每按一次键，输入的文字会变换。

例）[ABC] 键的情况，每按一次此键，“A” “B” “C” “a” “b” 会重复。

相同键上的文字要连续输入的情况下，按下 [→] 键，光标会前进。

例）要输入“ABY”的情况，按下 [ABC]、[→]、[ABC]2 次，按下 [WXYZ]3 次。



文字的输入 [CHARACTER] [ABC] [→] [ABC] [ABC] [WXYZ] [WXYZ] [WXYZ]

可以输入没有显示在键上的文字。没有被显示的文字所分配的按键及那个按键可以输入的文字如下所示：

a) [' ()] 键 ' → (→) → " → ^ → : → ; → ¥ → ?

b) [@ =] 键 @ → = → + → - → * → / → < → >

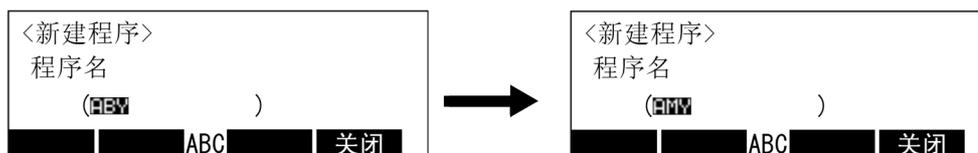
c) [, %] 键 , → % → # → \$ → ! → & → _ → .

3) 删除文字

文字输入错误时，按下 [CLEAR] 键，可将光标所在位置的 1 个文字删除。

例）要将“ABY”的“B”变更为“M”，变成“AMY”的情况下。

将光标移到文字“B”，按下 [CLEAR] 键删除后，再输入“M”、“Y”。



输入文字的修正 [←] [CLEAR] [MNO] [WXYZ] [WXYZ] [WXYZ]

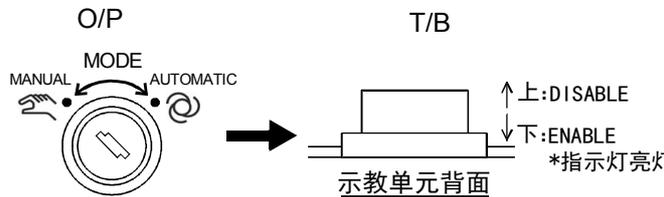
长押住 [CLEAR] 键的话，光标内的文字会全部删除。

(3) 菜单的选择方法

菜单的选择方法有下列 2 种，请选择其中一种执行。

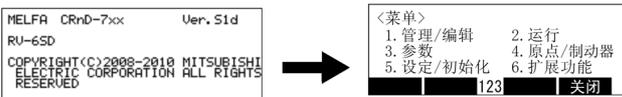
- 按下想选择项目的数字键。
- 将光标移到想选择的项目后按下 [EXE] 键。

显示由各个方法由菜单画面选择“1. 管理·编辑”的情况。



1) 使机器人控制器的 [MODE] 开关为“MANUAL”，按下示教单元的 [ENABLE] 开关使示教单元有效。
(开关和 ENABLELED 灯亮)

由<大标题>画面显示菜单画面



2) 在出现<大标题>画面时按下任一个键，(例 [EXE] 键)，则会显示<菜单>画面。

菜单画面的显示 [EXE]

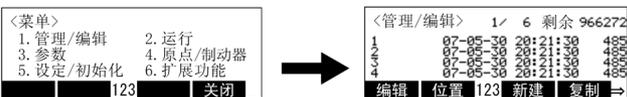
a) 用数字键选择菜单



按下 [CHARACTER] 键，在数字输入模式按下 [1] 键，显示<管理·编辑>画面。

菜单画面的显示 [1]

b) 用箭头键选择菜单



按下箭头键 ([↑]、[↓]、[←]、[→]) 将光标移动到“1. 管理·编辑”后，按下 [EXE] 键。会显示<管理·编辑>画面。

菜单画面的显示 [↑][↓][←][→]+EXE

其它的菜单画面也可用相同操作进行。

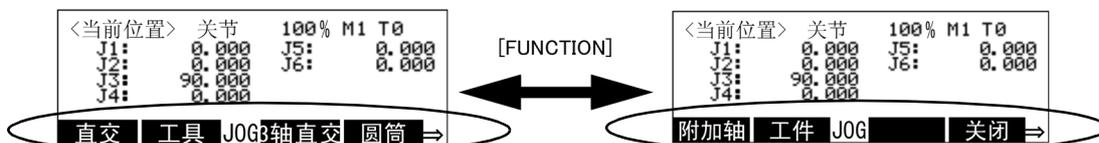
◆◆◆使用示教单元时◆◆◆

将机器人控制器的 [MODE] 开关调为“MANUAL”。除此之外情况下只可以做特定的操作 (在 JOG 画面显示现在位置、变更速度比例、监视)。

◆◆◆功能键◆◆◆

画面最下段有反白显示的菜单。由左开始的顺序以 [F1]、[F2]、[F3]、[F4] 键分配。按下对应的功能键，可以选择显示的菜单。

另外，画面右端有显示“=>”符号的情况下，表示现在有其它的菜单显示，请按下 [FUNCTION] 键，可以切换显示到另一个菜单。



3.2 JOG前进（概略说明）

JOG 前进是指以手动方式操作机器人，使机器人动作。

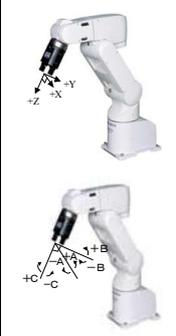
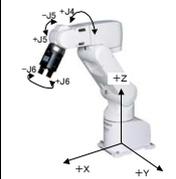
下列以垂直多关节型机器人做概略说明。

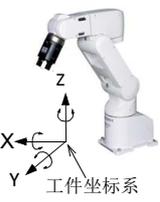
因会依机器人的机型，轴的构成而有所差异，因此请参照各别机型的「机器人本体设定及保养」。

3.2.1 JOG前进的种类

JOG 前进有下列 5 种类。（下列机器人的照片为参考机型。）

表 3-1: JOG 前进的种类

种类	操作	说明
关节 JOG 	<ol style="list-style-type: none"> 1) 将 [ENABLE] 开关调为 ENABLE。 2) 按下 Enable 开关（3 位置开关）。 3) 按下 [SERVO] 键使伺服开启。 4) 按下 [JOG] 键，再按下 [F1]（关节），选择关节 JOG 模式。 5) 按下各轴的按键。 J1 ~ J6 轴。 	<p>使机器人各轴移动。 使 J1 ~ J6 轴及附加轴 J7, J8 各别独立动作。</p> <p>轴数因机器人机型而有所差异。</p> <p>附加轴键: [J1] 键对应 J7 轴、[J2] 键对应 J8 轴。</p>
直交 JOG 	<ol style="list-style-type: none"> 1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“直交”各轴的功能键 ([F1] 到 [F4])，选择直交 JOG 模式。 5) 按下各轴的键。 X, Y, Z, A, B, C 轴。 	<p>按下 X, Y, Z 键则机器人基本坐标会以 XYZ 方向为基准往抓手的方向维持直线动作。 按下 A, B, C 键则以 XYZ 轴为中心旋转，抓手的方向会改变。 尖端的位置为固定。 但是，请务必以参数 MEXTL 设定正确的 TOOL 长度。</p>
TOOL JOG 	<ol style="list-style-type: none"> 1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“TOOL”各轴的功能键 ([F1] 到 [F4])，选择 TOOL JOG 模式。 5) 按下各轴的键。 X, Y, Z, A, B, C 轴。 	<p>按下 X, Y, Z 键则会以机器人的尖端为原点，以 TOOL 坐标为基准，维持抓手的方向，在前后、左右、上下方向以直线移动。按下 A, B, C 键则以 TOOL 坐标的 X, Y, Z 轴为中心旋转。 抓手的方向会改变。尖端的位置为固定。 但是，请务必以参数 MEXTL 设定正确的 TOOL 长度。 抓手尖端的 TOOL 坐标会依机器人的机型而有所不同 垂直多关节型机器人由机械 I/F 到抓手尖端方向为 +Z。 水平多关节型机器人由机械 I/F 到上方向为 +Z。</p>
三轴直交 JOG 	<ol style="list-style-type: none"> 1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“3 轴直交”的各功能键，([F1] 到 [F4])，选择 3 轴直交模式。 5) 按下各轴的键。 X, Y, Z, A, B, C 轴。 	<p>按下 X, Y, Z 键则会以机器人的基本坐标为基准，在 X, Y, Z 方向以直线动作。 按下 A, B, C 键则会和关节 JOG 相同，以关节为基准动作，但为了保持控制点的位置 (X Y Z 值)，姿势会改变。（和直交 JOG 不同，因为在 X, Y, Z, J4, J5, J6 里插补，因此不会保持姿势） 请务必以参数 MEXTL 设定正确的 TOOL 长度。</p>

种类	操作	说明
<p>圆筒 JOG</p> 	<p>1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“圆筒”的任意一个功能键，由 [F1] 到 [F4]，选择圆筒 JOG 模式。 5) 按下各轴的键。 X, Y, Z, A, B, C 轴。</p>	<p>以机器人的基本座标为基准，使抓手在圆筒方向动作的情况下使用。 X 轴由基本座标的原点开始往放射线方向移动。 Y 轴则以 J1 轴为中心旋转，和关节 JOG 同样动作。 (圆弧动作) Z 轴和直交 JOG 的 Z 相同。(上下动作) 按下 A, B, C 的键，会和直交轴的动作相同。 水平多关节型机器人亦有效。</p>
<p>工件 JOG (工件 JOG 模式)</p> 	<p>1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“工件”的任意一个功能键 (由 [F1] 到 [F4])，选择工件 JOG 模式。 5) 按下各轴的键。 X, Y, Z, A, B, C 轴。</p>	<p>进行该 JOG 操作时，需要事先将参数 WKnJOGMD (n=1 ~ 8) 设定为“0 (工件 JOG 模式)”。 按下 X、Y、Z 的键，即可以预先设定的机器人工件坐标系为基准保持抓手往 X、Y、Z 轴方向直线动作。 按下 A、B、C 的键，即可以同一工件坐标系的 X、Y、Z 轴为中心进行回转并改变抓手的方向。顶部位置是固定的。 但是，必须通过参数 MEXTL 正确设定工具长度。 注) 工件坐标系：用户事先进行设定。 (最多可设定 8 种) 没有设置工件坐标系的情况下则为直交 JOG 的动作。</p>
<p>工件 JOG (Ex-T JOG 模式)</p> <p>本功能可用于以下的软件版本。 F-Q 系列： R5 以上 F-D 系列： S5 以上</p> 	<p>1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“工件”的任意一个功能键 ([F1] 至 [F4]) 选择工件 JOG 模式。 5) 按下各轴的键。 X、Y、Z、A、B、C 轴。</p>	<p>进行该 JOG 操作时，需要事先将参数 WKnJOGMD(n=1 ~ 8) 设定为“1 (Ex-T JOG 模式)”。 按下 X、Y、Z 键，即可以预先设定的机器人工件坐标系为基准，保持抓手方向往 X、Y、Z 轴方向直线动作。 按压 A、B、C 键，即可以同一工件坐标系的 X、Y、Z 轴为中心旋转，并在改变抓手方向的同时移动控制点。 但是，需要预先通过参数 MEXTL 正确设定 TOOL 长度。 注) 工件坐标系：用户事先进行设定。 (最多可设定 8 种) 没有设定工件坐标系时，XYZ 为直交 JOG 的动作。 ABC 为以机器人的原点坐标系为中心旋转的动作。</p>

此外，在表 3-1 显示的 JOG 内，TOOL JOG、直交 JOG、圆筒 JOG 动作时，机器人的控制点接近特异点时，会发出蜂鸣声，同时，示教单元的画面会出现警告符号，促使正在操作机器人的人注意。本功能可以使用参数 (MESNGLSW)，自行设定为有效或无效。(请参照第 407 页的「5 参数的设定功能」)
 本功能的详细内容记载在第 458 页的「5.17 关于特异点接近警告」，请参照章节内容。

3.2.2 JOG前进的速度

可使用 [OVRD ↑] 和 [OVRD ↓] 键变更速度。现在的速度会以 % 显示在画面的右上角。

JOG 前进的速度有下列种类：

←		「OVRD ↓」键			「OVRD ↑」键			→	
LOW	HIGH	3%	5%	10%	30%	50%	70%	100%	

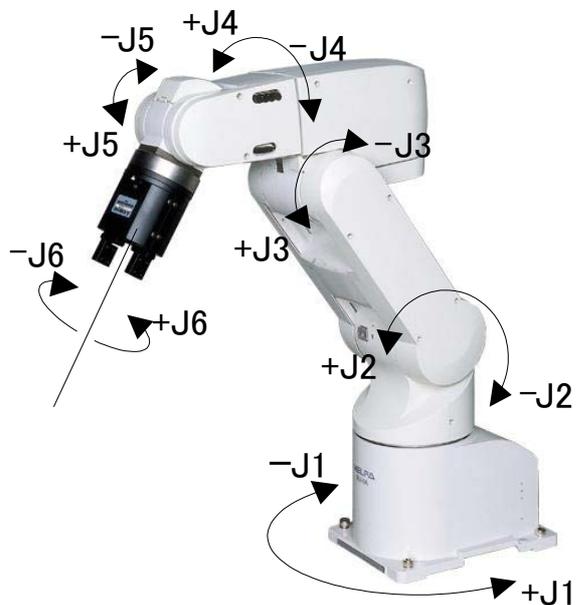
LOW 和 HIGH 为定寸前进。定寸前进为每按一次键，则机器人会以一定的量动作，移动量依机器人机型而有所不同。

表 3-2：固定尺寸示例

	关节 JOG	TOOL、直交 JOG
LOW	0.01 度	0.01mm
HIGH	0.10 度	0.10mm

3.2.3 关节JOG

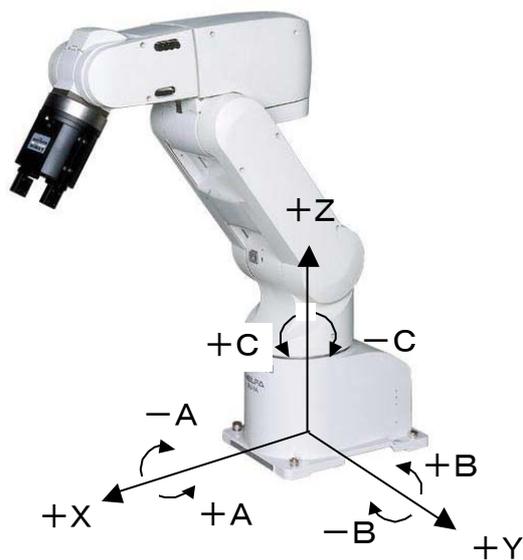
在各轴以角度单位使轴动作。



3.2.4 直交JOG

沿着机器人座标的方向移动。

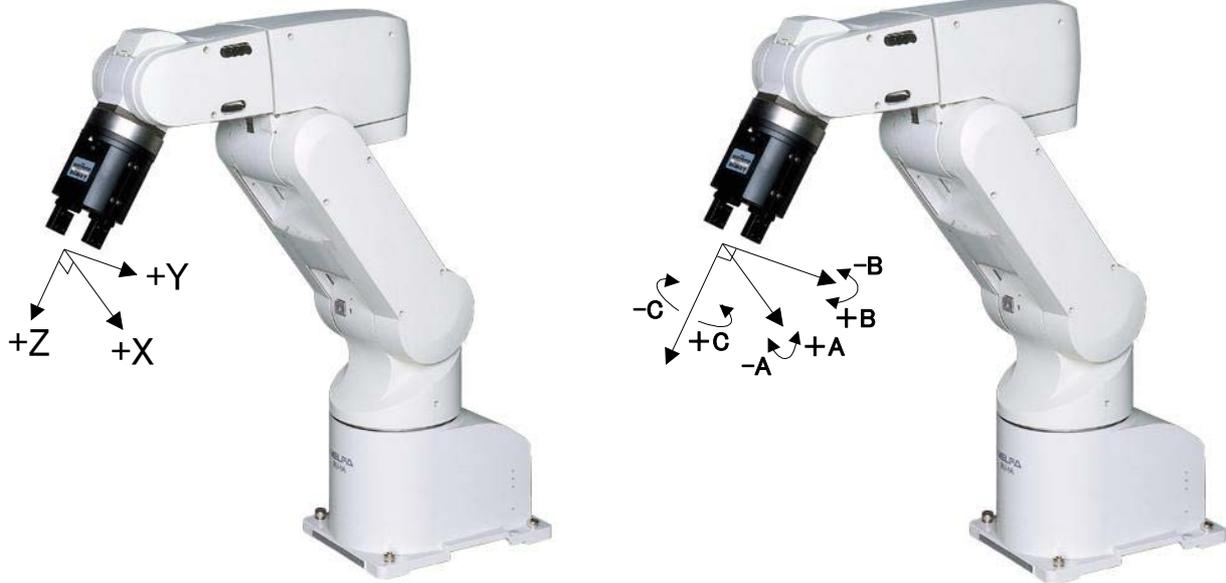
X、Y、Z 以 mm 单位移动。A、B、C 以角度单位移动。



3.2.5 TOOL JOG

沿着抓手尖端的方向移动。

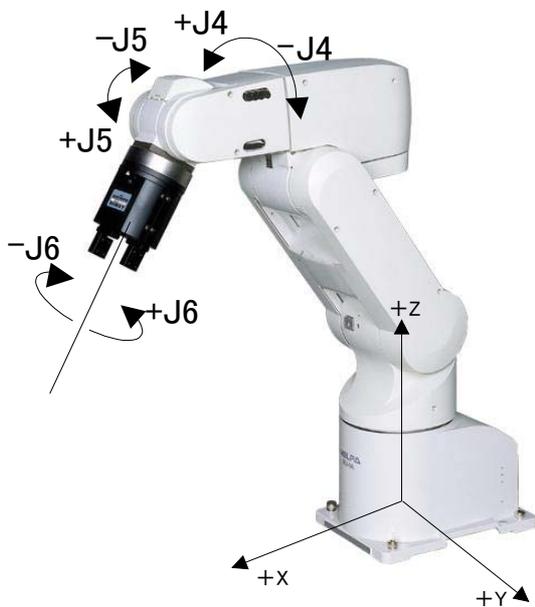
X、Y、Z 以 mm 单位移动。A、B、C 以角度单位移动。



3.2.6 三轴直交JOG

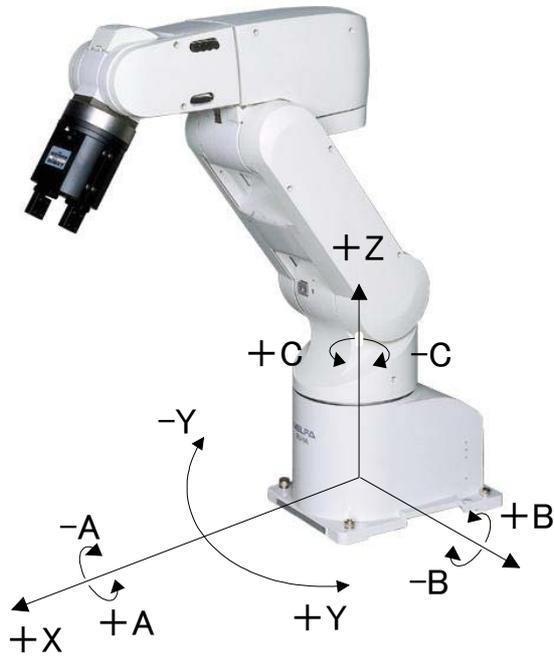
XYZ 和直交 JOG 相同沿着机器人座标方向移动。ABC 和关节 JOG 相同，以关节为基准动作，但是为了保持控制点的位置（XYZ 值），姿势会改变。

X、Y、Z 以 mm 单位移动。A、B、C 以角度单位移动。



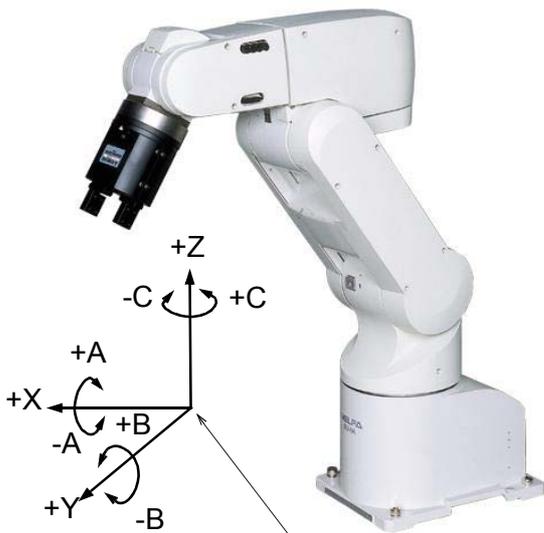
3.2.7 圆筒JOG

X 由机器人基本座标的原点开始以放射状在远离方向 / 接近方向移动。Y 是以 J1 轴为中旋转。Z 是往基本座标的 Z 方向移动。ABC 轴会变成和直交 JOG 相同动作。X、Y、Z 以 mm 单位移动。A、B、C 以角度单位移动。



3.2.8 工件JOG

沿着预先设定的工件坐标系（对准对象坐标的坐标系）的方向移动。XYZ 以 mm 为单位移动。ABC 以角度为单位移动。



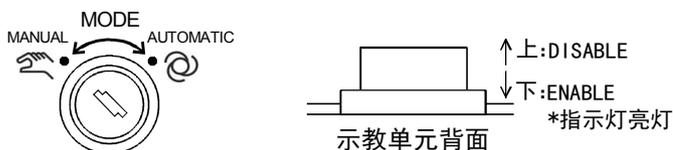
工件坐标系：对准工件或作业台等的坐标系

3.2.9 工具数据的切换

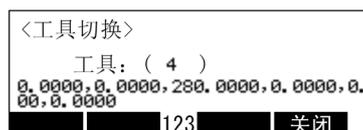
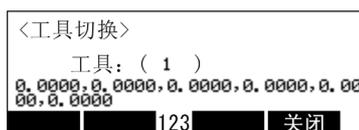
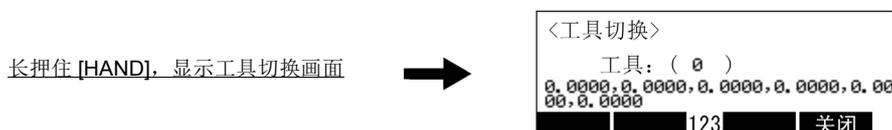
使用复数抓手的情况下，必须先预先登录各个工具数据，正确选择对应抓手的工具数据，才能使用。可使用 4 参数登录。

将要使用的工具的数据，以参数：MEXTL1 ~ 4 设定好，依下列操作选择使用的工具号码。

- 1) 将驱动器单元的 [MODE] 开关设定为“MANUAL”，示教单元的 [使能] 开关按下，使示教单元为有效。（开关和 ENABLE LED 灯亮）

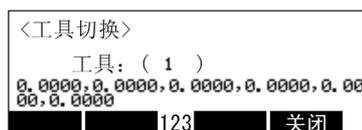


- 2) 长按住 [HAND] 键，会显示〈工具切换〉画面。
- 3) 按下希望的数字后，按下 [EXE] 键，则会切换到工具数据。参数的 MEXTL1 ~ 4 以数字的 1 ~ 4 对应。



工具数据的切换 [1] ~ [4] [EXE]

- 4) 在“关闭”按下被分配的功能键，结束。



结束 [F4]

- 5) JOG 画面的右上角会显示现在的工具号码 (T1 ~ T4)。

⚠ 注意

在程序的自动运行时，一边切换工具数据（MEXTL1 ~ 4 参数），一边使机器人移动到已示教位置的情况下，请在必要的场所，以工具号码取代 M_TOOL 变量，切换到工具数据运行。示教时的工具数据和单步运行时的工具号码不一致的话，会有意想不到的动作出现，请特别注意。

⚠ 注意

在程序的单步运行时，切换工具数据情况下，示教时的工具数据和单步运行时的工具号码不一致的话，会有意想不到的动作出现，请特别注意。

◆◆◆ TOOL 号码的确认◆◆◆

现在的 TOOL 号码可以用 TOOL 切换画面、JOG 面或 M_TOOL 变量进行确认。

◆◆◆ 相关资讯◆◆◆

MEXTL, MEXTL1, MEXTL2, MEXTL3, MEXTL4.....MEXTL16 参数

Tool 指令、M_Tool 变量

MEXTL 参数为在那个时候，将 TOOL 数据保持的参数。请注意使用 MEXTL1 ~ 4 的情况下，在选择 TOOL 号码的时候，MEXTL 参数已被写上。

请使用 TOOL 指令将 TOOL 号码回复为 0。

3.2.10 世界坐标系的切换（基本坐标号码的指定）

通过示教单元的操作可以轻松地更改作为机器人位置控制基准的世界坐标系。

在使用基本转换功能（Base 指令）的情况下进行示教作业时，是很方便的功能。

将要指定的基本坐标系预先设定在参数 WK1CORD ~ WK8CORD 中。（参照另一手册「从机器人本体安装到维护」的 2.4 动作的确认 (6) 工件 JOG 操作，及「功能和操作的详细说明」的 [5.1 动作参数]）

注意

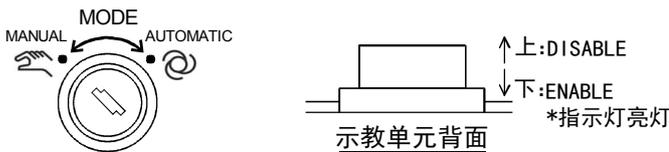
若通过本功能切换世界坐标系，则虽然机器人不动，但当前坐标值将发生变化。在进行位置示教前，请确认程序中使用的 Base 指令所对应的示教位置（位置变量）和当前的 JOG 操作中设定的基本坐标号码是否正确。

如果以错误的基本坐标号码进行示教，则执行程序时机器人将发生预想不到的动作，从而造成财产损失或人身事故。

请将基本坐标号码与对应的位置变量名关联起来，正确管理。

以下对操作步骤进行说明。

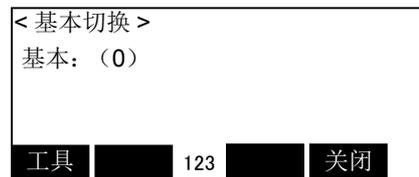
- 1) 将控制器的模式设为“MANUAL”，按下示教单元的 [ENABLE] 开关使示教单元有效。（开关和 ENABLE LED 亮灯）



- 2) 长按「HAND」键以显示 <基本切换> 画面。

显示 <工具切换> 画面时，按压画面下方的“Base”相应的功能键 [F1]。

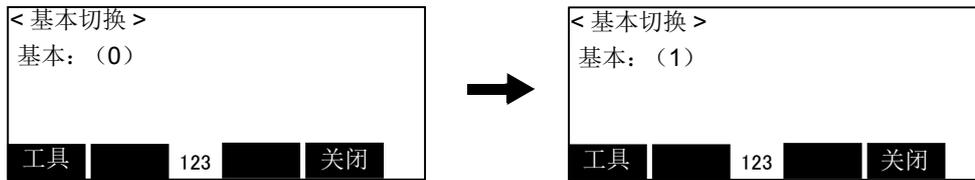
<基本切换> 画面的显示：长按 [HAND] 键



3) 输入想要的基本坐标号码，按压 [EXE] 键，则世界坐标系将被切换。

1 ~ 8: 基本坐标号码 (对应参数 WK1CORD ~ WK8CORD)

0: 返回出厂时的状态。(无基本转换的状态)



(图为切换至基本坐标号码 1 的示例)

通过基本坐标号码设定世界坐标系 [0], [1] ~ [8], [EXE]

4) 按压被分配为“关闭”的功能键，结束设定。



结束 [F4]

5) JOG 画面的右上方会显示当前的基本坐标号码 (B0 ~ B8)。

3.2.11 JOG操作中的冲突检知

备有冲突检知功能的机型，可以参数切换有效/无效。检知到冲突时，会发生报警号码 101n（最后 1 位的 n 为轴号码）。

JOG 操作时也可以使本功能有效，但是依机型的不同，初始设定状态会有差异。

表 3-3: 冲突检知参数

参数	名称	因素数	内容	初始值
冲突检知 ^{注1)}	COL	整数 3	定义冲突检知的使用为：“可以”或“不可以”及电源开启后的“有效”或“无效”。 要素 1: 冲突检知功能的使用：可以为 (1)、不可以为 (0) 要素 2: 程序运行时初始状态：有效为 (1)、无效为 (0) 要素 3: JOG 操作中有效为 (1) / 不设定为 (0) / NOERR 模式 (2) NOERR 模式下即使检知到冲突也不会有报警显示，只会使伺服关闭。请在冲突检知时，频繁发生报警、操作困难的情况下使用。除了程序运行时（含位置跳转、单步前进）以外，请依靠 JOG 操作用（要素 3）的设定。	RH-3FH/6FH/12FH/ 20FH 系列： 1,0,1 RV-F 系列： 0,0,1 RH-3FHR 系列： 1,1,1
JOG 时的检知标准	COLLVLJG	整数 8	设定每个关节轴在 JOG 操作时，（包含中断中）的检知标准（感应度）。单位 (%) 检知基准（感应度）升高的情况下，请将数值调小。 JOG 操作中若无发生冲突却发生冲突检知报警时，请将数值调大。 设定范围为 1 ~ 500 (%)	会因机型而有所差异。
抓手条件	HNDDAT0	实数 7	设定抓手的初始条件。（用 TOOL 座标指定），电源开启后的 JOG 操作会采用此设定值。 在 JOG 操作时使用冲突检知的情况下，请设定实际的抓手条件。没有设定的话，无法正确检知。 （重量，大小 X，大小 Y，大小 Z，重心 X，重心 Y，重心 Z） 单位：Kg, mm	值会因机型而有所差异。负载可设定为最大负载值。
工件条件	WRKDAT0	实数 7	设定工件的初始条件。（以 TOOL 座标指定）。电源开启后的 JOG 操作会采用此设定值。 （重量，大小 X，大小 Y，大小 Z，重心 X，重心 Y，重心 Z） 单位：Kg, mm	0,0,0,0,0,0,0,0,0,0,0,0

注 1) 无法和多重机制控制同时使用。

(1) JOG 操作时的冲突检知基准调整

操作时的冲突检知感应度设定较低。若要设定较高的冲突检知感应度的话，请使用参数的 COLLVLJG 做调整。另外，请正确设定使用参数 HNDDAT0 及 WRKDAT0。没有正确设定的情况下，进行 JOG 操作时，会因姿势而有错误检知的可能。

◆◆◆冲突检知功能的注意点◆◆◆

即使冲突检知功能有效，也无法百分之百的阻止因与周边机器的干涉而造成机器人、抓手、工件等的损伤。基本上还是请小心操作机器人，勿与周边机器发生干涉。

◆◆◆冲突后的操作◆◆◆

在抓手和周边机器发生干涉的情况下，若直接开启伺服电源，有可能再度变成冲突检知状态而无法开启伺服。若循环多次将伺服开启后，仍然发生报警的情况下，请暂且将刹车解除，使手臂闪避后再开启伺服。请参照第 60 页的「3.10 无法解除报警的暂时报警复位操作」，将伺服开启后再以 JOG 操作使手臂闪避。

◆◆◆和自动运行用的冲突检知的关系◆◆◆

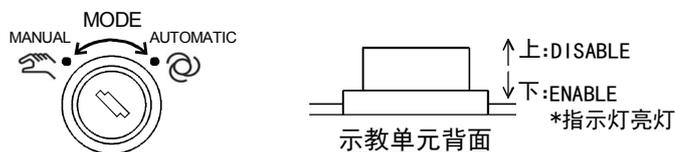
JOG 操作用的冲突检知和自动运行用的冲突检知是独立设定的。
无程序运行的情况下，会采用 JOG 操作用的设定。JOG 操作用的设定为有效的情况下，即使在程序中设定自动运行用的冲突检知为无效，也会中断运行，切换到 JOG 操作用的设定。

3.3 抓手开闭

说明安装在机器人上的抓手的开闭操作。

抓手最多可控制 6 个。X, Y, Z, A, B, C 轴的键, 各别分被为抓手 1, 2, 3, 4, 5, 6。抓手开为“+”侧的键, 抓手闭为“-”侧的键, 按下即可操作。

- 1) 将驱动器单元的 [MODE] 开关设定为“MANUAL”, 示教单元的 [ENABLE] 开关按下, 使示教单元为有效。(开关和 ENABLE LED 灯亮)



- 2) 按下 [HAND] 键可以显示〈抓手〉画面。

OUT-900 显示抓手的开闭状态, IN-900 显示抓手确认输入信号的 ON/OFF 状态。

要打开抓手 1 请按下 [+X] 键、要关闭请按下 [-X] 键。其它的抓手则和 Y, Z, A, B, C 轴的键操作相同。

显示抓手画面 [HAND]

<抓手>	±C:抓手1	±Z:抓手4
	±B:抓手2	±Y:抓手5
	±A:抓手3	±X:抓手6
OUT-900	76543210	IN-900 76543210
退避点	整列	HND 关闭

OUT-900 ~ OUT-907	7	6	5	4	3	2	1	0
开 / 闭	闭	开	闭	开	闭	开	闭	开
抓手号码	4		3		2		1	

IN-900 ~ IN-907	7	6	5	4	3	2	1	0
输入信号号码	907	906	905	904	903	902	901	900

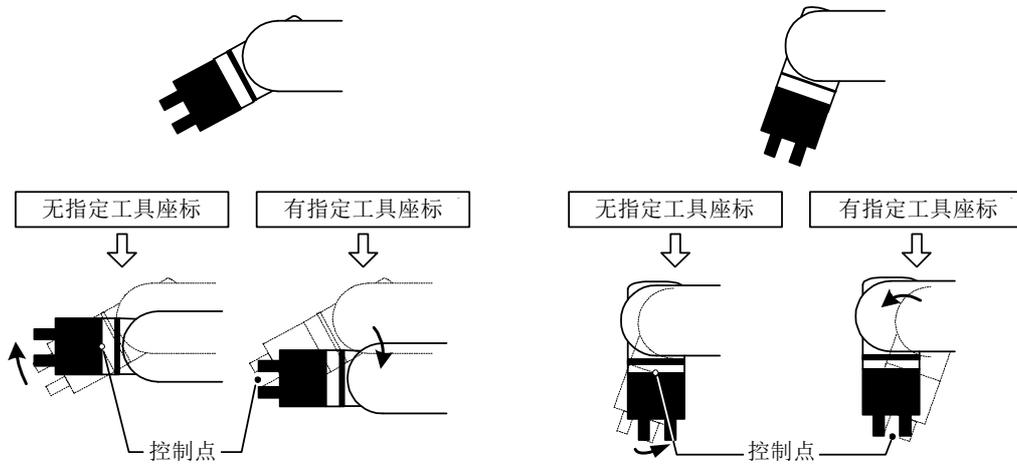
◇◆◇自动运行用的冲突检知的关系◆◆◇

机器人的抓手部份可以拿取各式各样的工件。使用气泄制电磁阀的双螺线型电导管, 在抓手开闭操作, 控制抓手信号的 2 位。

抓手信号的详细内容请参照第 444 页的「5.12 关于抓手种类」及第 445 页的「5.13 关于抓手初始状态」。

3.4 整列抓手

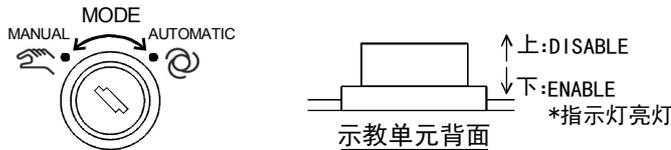
安装在机器人上的抓手的姿势，可以用 90 度单位的姿势做整列动作。
此功能下，机器人会移动到当前位置的 A、B、C 成份中离 90 度单位最近的值。



用工具指令及参数指定工具座标的情况下，会在指定的工具座标上进行抓手整列。没有指定工具座标的情况下，会在机械 1/F 进行抓手整列。上图为垂直多关节型机器人的例子，[有指定工具座标]，为在抓手尖端指定工具座标的情况。关于了 TOOL 座标，请参照第 432 页的「5.6 关于标准 TOOL 坐标」。

抓手整列的步骤如下所示。

- 1) 将驱动器单元的 [MODE] 开关设定为“MANUAL”，示教单元的 [ENABLE] 开关按下，使示教单元为有效。（开关和 ENABLE LED 灯亮）

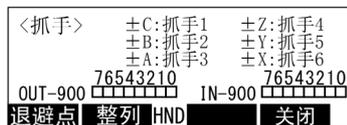


- 2) 按住 ENABLE 开关（3 位置开关）后再按下 [SERVO] 键，使伺服开启。
- 3) 按下 [抓手] 键，会显示〈抓手〉画面。

显示抓手画面 [抓手]



- 4) 按住 Enable 开关（3 位置开关），再按下被分配为“整列”的功能键。当继续按着的时候，机器人会执行抓手整列动作。动作中驱动器单元的 [START]LED 会亮灯。
动作途中放开任一个键，则机器人动作停止。



抓手整列的执行 [整列]

⚠ 注意

依据抓手整列的操作，使姿势成分（A、B、C）为 180 度的情况下，尽管是相同的姿势但是成分的值会变成 +180 度或 -180 度的其中一个。这是因为内部的运算误差所造成的结果，会变成那一个值并没有规则性。以此位置做为 Pallet 定义指令（DEF PLT）的位置数据使用的情况下，相同的姿势成分里混合了 +180 度和 -180 度，因此 Pallet 运算为了在 -180 度 ~ +180 度间算出分配位置，抓手回转会变成预期外的动作。在姿势成分 180 度的位置数据使用在 Pallet 的定义时，请将 180 度的符号 + 或 - 的任何一个准备好。此外，将此位置数据直接使用在插补指令的目的位置的情况下，无论那一个值都不会有问题的进行动作。

3.5 程序编辑

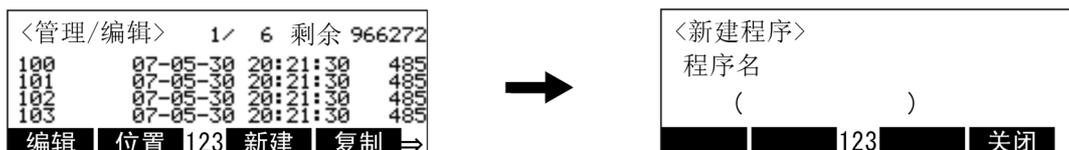
本机器人 CPU 里所用 MELFA-BASIC V，依据丰富的运算功能，可记述高级的作业。此外，在此说明使用示教单元作程序编辑的方法。（关于 MELFA-BASIC V 的指令、记述等，请参照本书 / 第 163 页的「4.13 指令的详细说明」。）

3.5.1 程序的制作

(1) 开启程序编辑画面

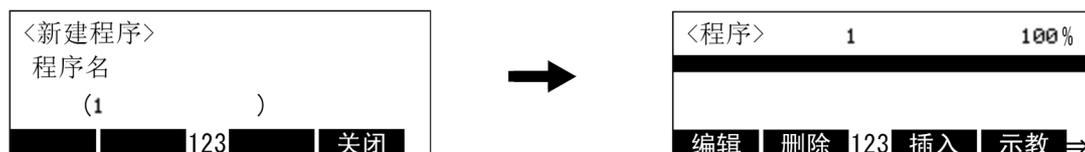
开启制作新程序的编辑画面。下列为建立程序名为“1”的示范例子。

- 1) 在〈菜单〉画面，选择“1. 管理 / 编辑”画面。
- 2) 按下对应“新规”的功能键。显示程序名输入画面。



菜单的选择 [F3]

- 3) 输入程序名。显示指令编辑画面。
(输入已有的程序名的情况下，会开启已存的程序)



输入程序号码 1 [1] [EXE]

◇◆◇平常执行程序（ALWAYS 属性）的编辑◇◆◇

在 SLTn 参数编辑在平常执行（ALWAYS）属性上设定的程序时，请暂时解除平常执行的属性后再编辑。因为平常执行属性的程序为被平常执行，因此无法编辑。在 SLTn 参数将 ALWAYS 变更为 START，然后重开启机器人 CPU 和驱动单元的电源，使平常执行停止。

◇◆◇开启已存程序◇◆◇

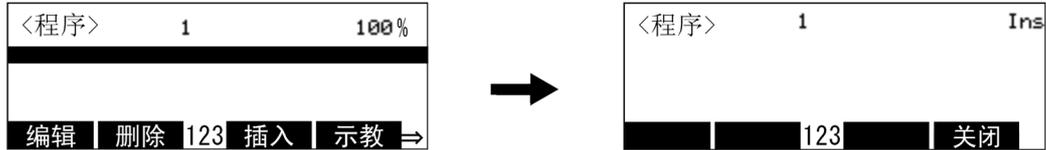
如上列所述，有把程序名输入即有的程序名的方法，在〈管理・编辑〉画面按下 [↑]、[↓] 键，在选择要开启程序的状态，按下对应“编辑”的功能键

(2) 程序的制作

下列显示输入 3 单步的程序的情况下的操作键。

- 1 Mov P1
- 2 Mov P2
- 3 End

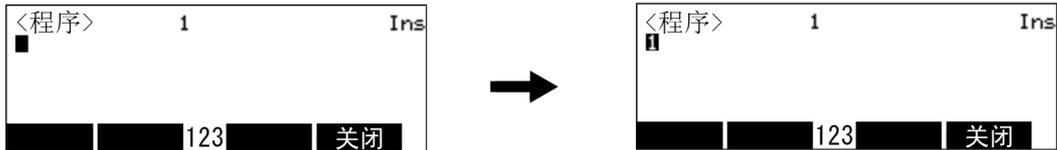
1) 在指令编辑画面，按下对应“插入”的功能键 ([F3])。



单步的插入 [F3]

2) 步号“1”的输入

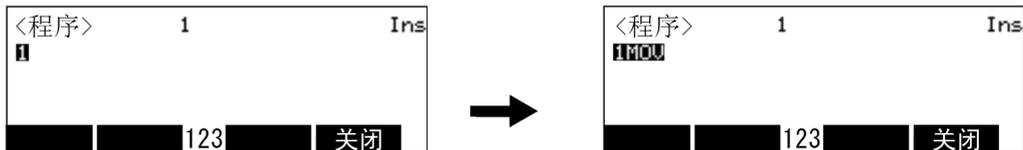
按下 [CHARACTER] 键，在数字输入模式按下 [1] 键。
可省略步号和指令后的中间的空间可以省略。



步号的输入 [1]

3) 指令“MOV”输入

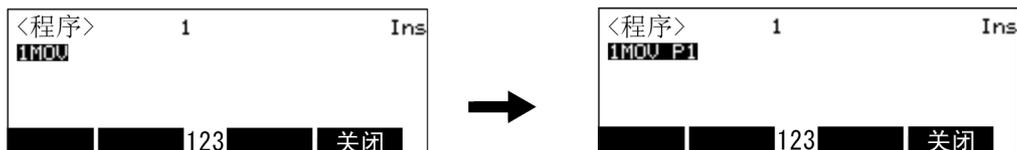
按下 [CHARACTER] 键，在文字输入模式，依按下
3 次 [MNO] (“M”)、[→]、[MNO]3 次 (“O”)、[TUV]、3 次 (“V”)、[→] 键的顺序。



MOV 的输入 [CHARACTER] [MNO] [→] [MNO] [MNO] [MNO] [TUV] [TUV] [TUV]

4) 附带在指令上“P1”数据的输入

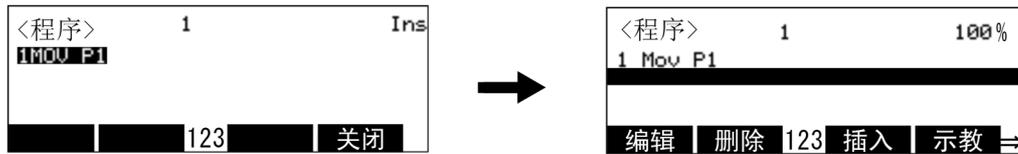
按下 [SP] (“SPACE”)、[PQRS] (“P”) 键。
按下 [CHARACTER] 键，在数字输入模式下按下 [1]。
指令和指令附带数据间必须有空隔 (SPACE)。



P1 的输入 [SP] [PQRS] [CHARACTER] [1]

5) 单步 1 的确定

按下 [EXE] 键则确定。



单步 1 的确定 [EXE]

6) 以下，同样的输入单步 2、3。



以上，则可以输入程序。

◇◆◇使单步的前后显示◇◆◇

按下 [↑]、[↓] 键则画面可以滚动。

◇◆◇显示特定单步◇◆◇

按下对应 "跳转" 的功能键，在显示单步号码输入画面里，输入希望显示的等单步号码，然后按下 [EXE] 键，则会显示输入的单步。

◇◆◇插入时可以省略单步号码◇◆◇

省略的情况下，会插入在光标的下一行。

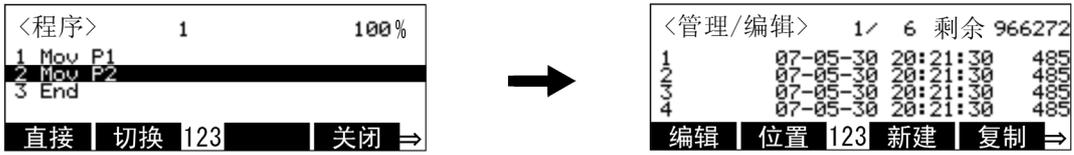
◇◆◇可自动变换大写字母及小写字母◇◆◇

在 MELFA BASIC V，预定语及变量名由大小写字母组合显示。即使由示教单元输入时，按大小字母原来的样子输入，也会在行的确认时自动的变换。

(3) 程序制作的结束储存

按下对应 "关闭" 的功能键, 则结束程序储存制作。

没有显示 "关闭" 的情况下按下 [FUNCTION] 键显示。



程序的结束和储存 [F4]

 注意

在程序编辑中切断电源的话, 包含位置数据等的编辑内容会被撤消, 请特别注意。
另外, 请在电脑里做备份数据保存。
建议在配件的电脑支持软件里做程序管理。

(4) 程序的修正方法

进行程序的修正前，请先参照本书第 34 页的「3.5.1 程序的制作」的「(1) 开启程序编辑画面」，开启程序的编辑画面。

以下为将 STEP 5 的 "Mov 5" 修正为 "Mvs P5" 的操作示范。



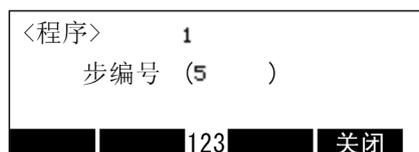
1) 呼出 STEP 5

按下对应 "跳转" 的功能键，在显示的 STEP 号码输入画面里，按下 [5]、[EXE] 键。将光标移动到 STEP 5。

使用 [↑]、[↓] 键将光标移到 STEP 5 也可以呼出 STEP 5。



呼出 STEP5 [F2]



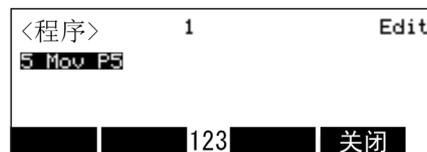
呼出 STEP 5 [5] [EXE]

2) 修正指令语。

按下对应 "编辑" 的功能键。

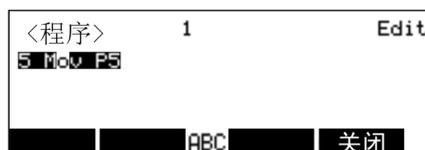
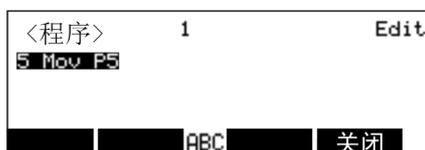


修正指令语 [F1]



3) 按下 [→] 键 3 次，将光标对准 "o"。

按下 [TUV] 键 3 次 ("v" 的输入)、[→] 键 1 次、[PQRS] 键 4 次 ("s" 的输入)，则会变成 "5 Mv s P5" 然后按 [EXE] 键确定。



[TUV] [TUV] [TUV] [→]
修正指令语 [PQRS] [PQRS] [PQRS] [PQRS]



指令语的修正 [EXE]

◇◆◇将光标符合修正◇◆◇

使用 [↑]、[↓] 键，将光标符合单步 5、按下对应 "编辑" 的功能键，即可进行修正。

◇◆◇中止修正◇◆◇

在修正途中按下对应 "关闭" 的功能键，即可中止修正。

◇◆◇修正文字◇◆◇

将光标移动到错误的文字上面，按下 [CLEAR] 键，往左边删除后再输入正确文字。

◇◆◇若修正程序的话◇◆◇

若进行程序修正请务必做储存动作 (按下对应 "关闭" 的功能键 [F4] 或按下示教单元背面的 [使能] 开关，使示教单元成为无效。然后进行内容相关的单步运行，确认修正正确。

(5) 现在位置的示教

使用 JOG 操作等将机器人往作业位置移动，可以在此位置用正在程序中被使用的位置变量做示教（登录）。已经示教完毕的情况下，可以在上面书写（修正）。有在指令编辑画面的示教和在位置编辑画面的示教。

(a) 在指令编辑画面的示教

呼叫出示教位置变量使用的单步。

下列为将单步 5"Mov P5" 的位置变量，示教为 P5 现在位置的操作步骤。请事先以 JOG 操作等，将机器人移动到作业位置。

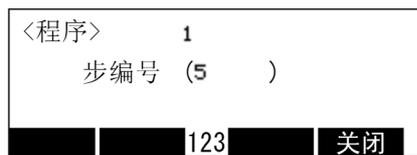
1) 呼叫出单步 5。

按下对应"跳转"的功能键，在显示的步号输入画面按下 [5]、[EXE] 键。光标移动到单步 5。

使用 [↑]、[↓] 键，使光标符合单步 5，也可以呼出单步 5。



呼出单步 5 [F2]



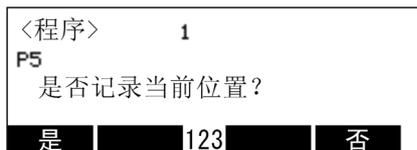
呼出单步 5 [5] [EXE]

2) 现在位置的示教

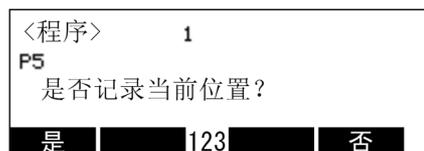
按下对应"示教"的功能键 ([F4])，显示确认画面。



现在的位置登录 [F4]



3) 按下对应"是"的功能键，机器人的现在位置会示教为 P5，回到原来指令编辑的画面。按下对应"否"的功能键 ([F1]) 的话，可以中止示教。



登录现在的位置 [F1]



以上，则现在位置的示教操作完成。

◇◆◇只有一个位置变量为对象◇◆◇

读出的单步 "Mov P1+P2" 和 "P1=P10" 等等，做为复数的位置变量使用的情况下，最左侧的位置变量为示教的对象。

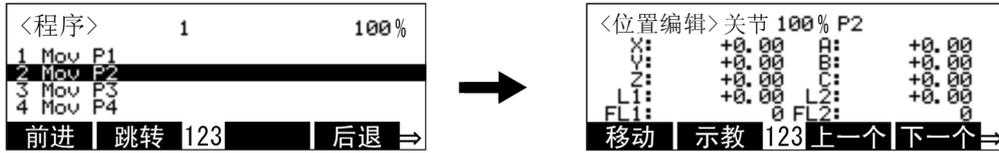
要示教其它的变量的情况下，请参照下一页的"在 (b) 位置编辑画面的示教"。

(b) 在位置编辑画面的示教

下列为将现在位置示教到位置变量 P5 的操作步骤。请预先使用 JOG 操作等将机器人移动到作业位置。

1) 位置编辑画面的表示

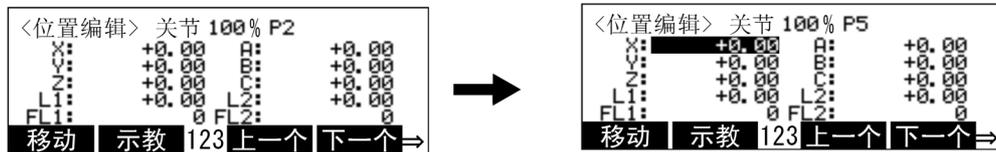
在指令编辑画面按下对应"切替"的功能键 ([F2])，则会显示位置编辑画面。



位置编辑的表示 [F2]

2) 呼出 P5。

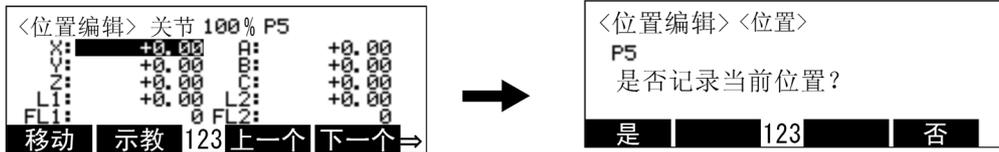
按下对应"上一个", "下一个"的功能键 ([F3]), ([F4]), 呼出"P5"。



呼出 P5 [F3] [F4]

3) 现在位置的示教

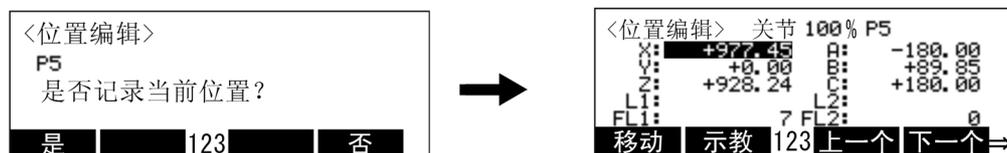
按下对应"示教"的功能键 ([F2])，则会显示确认画面。



对 P5 显示教导 [F2]

4) 按下对应"是"的功能键，则会显示机器人的现在位置数据为 P5，回复到原来的位置编辑画面。P5 则显示机器人的现在位置数据。

按下对应"否"的功能键 ([F1]) 的话，可以中止示教。

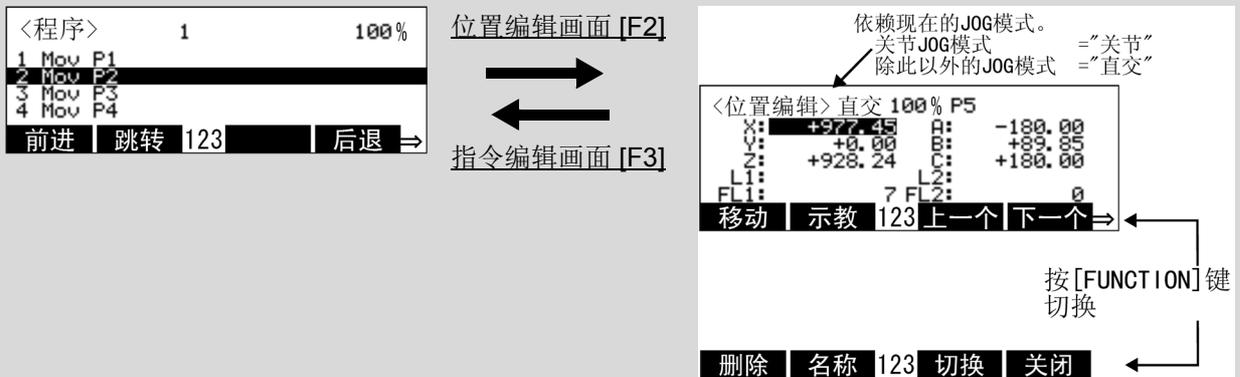


登录现在的位置 [F1]

以上，则现在位置的示教操作完成。

◇◆◇指令编辑画面和位置编辑画面的切换◇◆◇

按下对应"切替"的功能键（F2），则可以相互切换指令编辑画面和位置编辑画面。
画面没有显示"切替"的情况下，请按下 [FUNCTION] 键，则"切替"会被显示出来。MENU 的右端没有出现"⇒"的情况下，请按下 [FUNCTION] 键，则会显示可切换 MENU 的状态。



可以用上一个（F3），下一个（F4）前后的位置变量依次呼叫出来。通常只可以呼叫位置变量，但是切换功能键则可以用名字（F2）呼出关节变量。呼出关节变量后，可以用上一个（F3），下一个（F4）前后的关节变量依次呼叫出来。

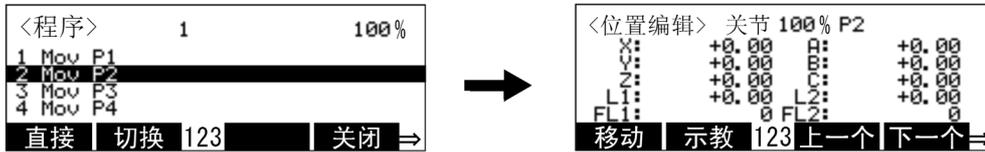
在关节变量显示最前或最后的话，在下一个显示会回到位置变量。

(6) 位置变量的删除

下列为删除位置变量的操作步骤。
只限程序中没有被使用的变量可以删除。

1) 位置编辑画面的表示

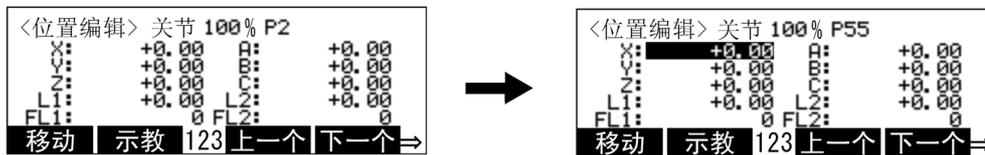
在指令编辑画面里，按下对应“切替”的功能键（[F2]），则会显示位置编辑画面。



位置编辑画面的表示 [F2]

2) 删除位置变量的表示

按下对应“上一个”或“下一个”的功能键，（[F3]）或（[F4]），显示要删除的变量位置。

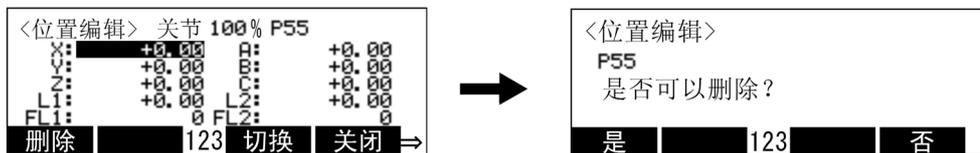


呼出 P55 [F3] [F4]

将 P55 删除的情况

3) 位置变量的删除

按下对应“删除”的功能键（[F1]），则会显示确认画面。
（没有显示“删除”的时候，请按下 [FUNCTION] 叫键，就会显示出来。）

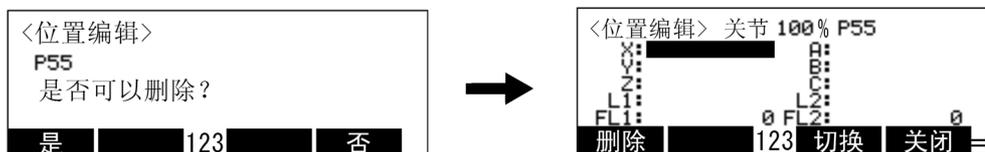


呼出 P55 [F1]

删除位置变量 [F1]

4) 位置变量的删除

按下对应“是”的功能键，则位置变量会被删除。



删除位置变量 [F1]

登录现在位置 [F1]

(7) 位置数据的确认

可以将机器人移动到要被示教的位置变量的地方。
在移动模式里有关节模式和直交模式。像下列一样依靠已设定的 JOG 模式。

表 3-4: 移动到指定位置数据的方法

名称	移动方法
关节模式	使用关节插补移动到画面显示的位置变量的地方。 JOG 模式为关节插补时, 会变成此移动方法。 变成和 Mov 指令同样的动作。
直交模式	以直关插补移动到指定位置数据的地方。因此和现在位置指定的位置的构造标志不同的话, 无法移动。JOG 模式为直交、3 轴直交、圆筒及 TOOL JOG 时, 会变成此移动方法。 变成和 Mvs 指令同样的动作。

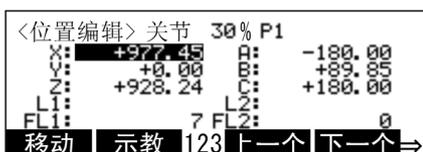
下列为操作方法示范。

本操作请在伺服 ON、轻押住 Enable 开关 (3 位置开关), 维持伺服 ON 的状态下进行。

1) 预先显示想移动的位置变量。

按下对应 "移动" 的功能键, 按住键的时候, 机器人会往表示的位置变量的地方移动。

放开对应 "移动" 的功能键, 则机器人会停止。此外, 放开 Enable 开关 (3 位置开关), 再强力按下的话, 伺服 OFF (关闭), 机器人停止。



显示位置编辑画面 [F1]



注意

机器人以本操作动作。

即使机器人往显示在画面上的位置变量场方移动, 也要进行确认与周边将置等没有干涉。
建议最初将速度放慢进行。

此外, 请边依据移动模式 (关节、直交) 边预测机器人的动作轨迹。

◇◆◇动作速度◇◆◇

可以按下 [OVRD ↑] 键 (速度调快)、[OVRD ↓] 键 (速度调慢) 来调整移动速度。

(8) 位置数据的 MDI (Manual Data Input) 登录 (修正)

所谓 MDI 是指在位置数据的各轴成分数据里输入直接数值的登录方法。

将位置数据加总计算, 作为基准位置开始的相对移动量 (差分) 所使用位置变量的登录, 以及微调已登录完毕的位置数据等情况下, 有效的登录方法。

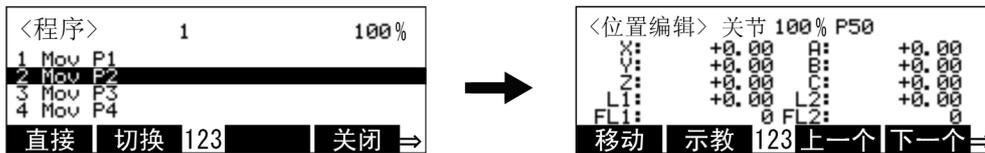
参考) 作为相对移动量的位置数据

例) 将 P1 做为基准位置, 由此开始在 50mm 上空, 用关节插补做移动的情况下, P1 用现在位置的示教登录, P50 在 Z 轴成分 "50.00"、对于其它轴的成份, MDI 将 " 0.00" 登录为位置变量, 所以可能执行 Mov P1+P50。

下列为上述例子, 将 P50 在 MD 登录的情况的操作方法示范。

1) 位置编辑画面的显示

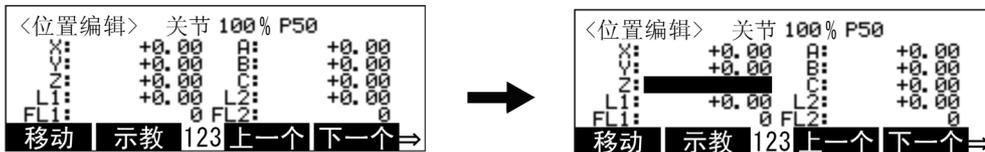
在指令编辑画面按下对应 "切换" 的功能键, 会显示位置编辑画面。



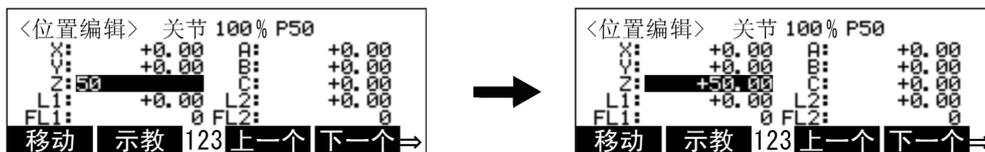
位置编辑画面的表示 [F2]

2) 在 Z 轴输入 "50"。

按下 [↓] 键 2 次, 将光标往 Z 轴移动。长押住 [CLEAR] 键, 会将显示的 "+0.00" 删除。按下 [5]、[0]、[EXE] 键。位置变量 P50 里, 只有 Z 轴的值会作为 50mm 被登录。



移动光标和值的删除 [↓] [CLEAR]



值 50 的输入 [5] [0] [EXE]

3.6 排错操作

所谓排错（Debug）是指随著制作的程序进行是否正常的动作的测试，异常时，在修正错误的作业中，可以用示教单元的排错功能确认。可以执行的排错功能如下所示。程序制作的情况下，务必进行排错，确定程序没有错误后，再执行运行。

(1) 单步前进

使用示教单元，将程序一个单步一个单步的执行（单步运行），以确认动作。

进行排错操作前，请参照第 34 页的「3.5.1 程序的制作」的第 34 页的「(1) 开启程序编辑画面」，显示要做排错的程序编辑画面。

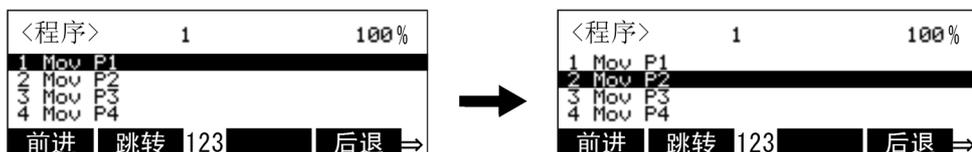
机器人的动作指令请在伺服开启操作后，轻押住示教单元的使能开关 [3 位置开关]，维持在伺服开启状态下执行。

1) 单步前进的执行

按下对应“前进”的功能键（[F]），持续押住的时候，光标的地方会执行等祭。

途中放开功能键的话，执行中断。另外，在机器人移动中放开或更强力按下使能开关（3 位置开关）的话，可以使伺服关闭，且中断执行。

执行中操作面板的 [开始] 开关的 LED 灯会亮起。1 单步的执行完毕后，[开始] 开关的灯会熄灭、且 [停止] 开关的 LED 灯会亮。放开 [F1] 键的话，示教单元画面的光标会往下一个单步移动。



单步前进 [F1]

每当按下对应“前进”的功能键时，会往下一个单步前进。

⚠ 注意

请特别注意动作中的机器人。在与周边装置的干涉等异常发生时，请以下列方式使机器人停止：

1. 请放开对应的功能键：“前进”。
2. 放开或强按下使能开关（3 位置开关）。

◇◇◇ 所谓单步运行 ◇◇◇

将程序 1 个单步 1 个单步的执行，则称为单步运行。动作速度缓慢，因为执行每一单步就停止，有效的进行程序的确认及动作位置的确认。

执行有光标的单步。

执行结束指令及 911 指令的话，单步不会前进。

◇◇◇ 执行单步的变更 ◇◇◇

使用箭头符号键移动光标及用跳转操作（“跳”），皆可以变更单步。

◇◇◇ 将动作中的机器人直接停止 ◇◇◇

- ① 按下 [紧急停止] 开关。

将伺服关闭且将在动作中发生紧急停止报警的机器人直接停止。

继续请以复位报警→伺服 ON→单步运行操作的顺序进行。

- ② 放开或者是更强大的按下使能开关（3 位置开关），会使伺服 OFF 且将动作中发生报警 2000 的机器人直接停止。

继续请以复位报警→伺服 ON→单步运行操作的顺序进行。

(2) 步编号返回

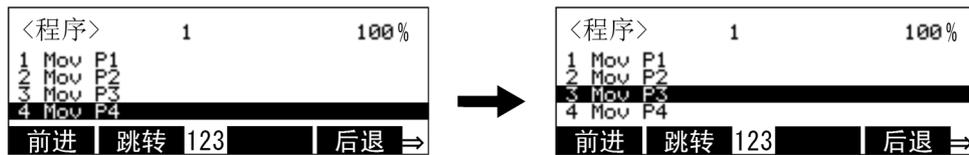
以步编号前进及自动运行将执行结束的步编号，或中断的步编号限制用插补命令执行在 1 个步编号 1 个步编号往返回方向执行。而且，可以返回步编号，最多 4 个。

1) 步编号返回的执行

按下对应 "后退" 的功能键，持续按住的时候，会往步编号的返回方向只执行 1 个步编号。

途中放开功能键的话，执行中断。另外，在机器人移动中放开或更强力按下 Enable 开关（3 位置开关），的话，可以使伺服关闭，且中断执行。

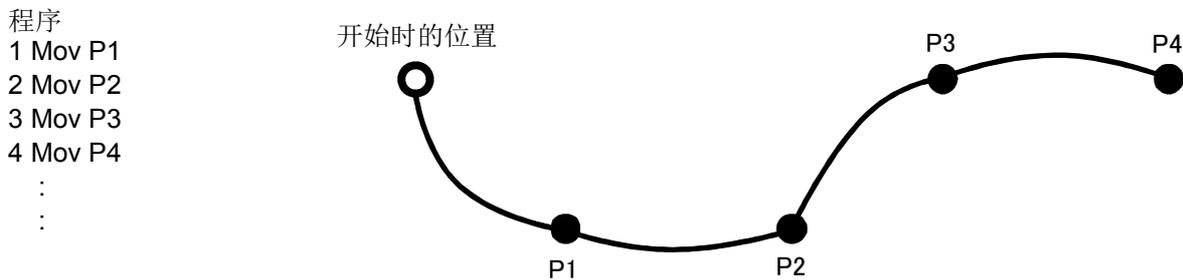
执行中操作面板的 [START] 开关的 LED 灯会亮起。1 步编号的执行完毕后，[START] 开关的灯会熄灭，且 [步编号] 开关的 LED 灯会亮。示教单元画面的光标在步编号返回方向，往下一个插补指令的步编号移动。



步编号返回 [F4]

每当按下对应 "后退" 的功能键时，会返回到前一个步编号。

[补充] 下列为要将程序步编号前进到步编号 4 以后，再进行步编号返回的情况。返回到 P1 后再进行步编号返回。执行步编号返回到开始时的位置为止。（开始时的位置指开始执行步编号 1 的位置。）



注意

请特别注意动作中的机器人。在与周边装置的干涉等异常发生时，请以下列方式使机器人停止：

1. 请放开对应的功能键：“后退”。
2. 放开或强按下 Enable 开关（3 位置开关）。

◆◆◆将动作中的机器人直接停止◆◆◆

①按下 [紧急停止] 开关。

将伺服关闭且将在动作中发生紧急停止报警的机器人直接停止。

继续请以复位报警→伺服 ON →步编号运行操作的顺序进行

②放开或者是更强大的按下 Enable 开关（3 位置开关），会使伺服 OFF 且将动作中发生报警 2000 的机器人直接停止。

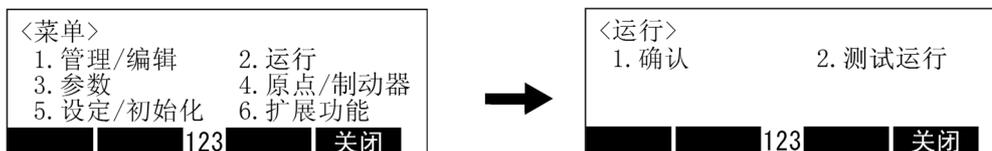
继续请以复位报警→伺服 ON →步编号运行操作的顺序进行。

(3) 其它插槽的单步前进

确认多重任务的程序的情况下，可以在运行菜单画面做单步前进。

1) 运行菜单的选择

在菜单画面按下 [2] 键，选择 "2 运行"。

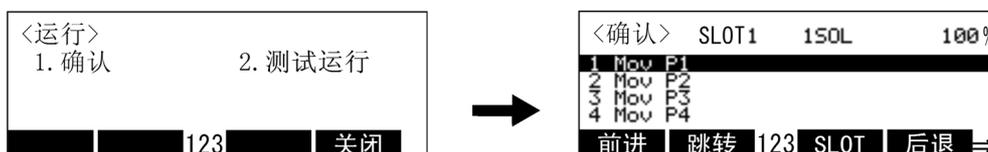


运行画面的选择 [2]

2) 选择确认画面

在运行画面按下 [1] 键后，选择 "1. 确认"。

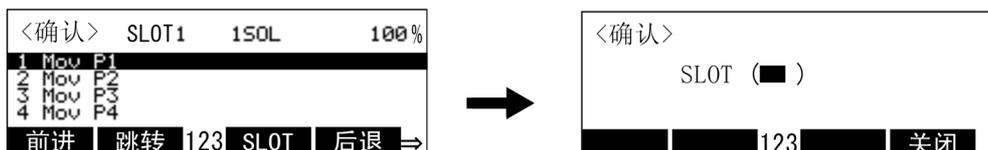
在插槽 1 会显示已设定的程序。插槽号码上会显示连续程序名称。



运行画面的选择 [1]

3) 变更插槽

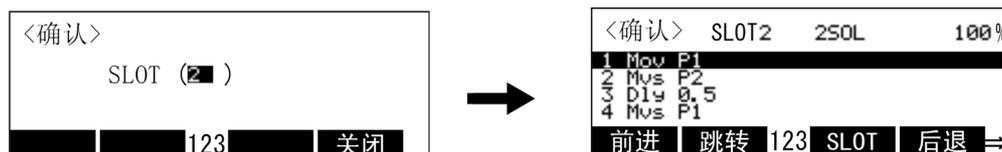
按下对应 "SLOT" 的功能键 ([F3])，则显示插槽号码指定画面。



变更插槽 [F3]

希按下希望的插槽按号后再按下 [EXE] 键。

显示输入的插槽号码程序。（下列为指定插槽 2 的情况）



变更插槽 [2] [EXE]

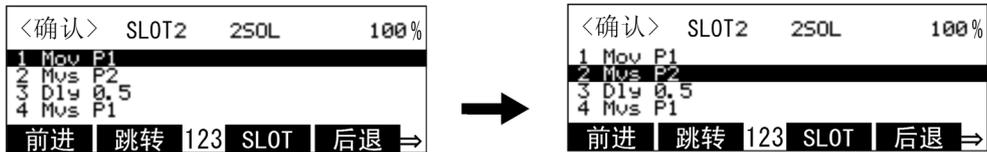
4) 执行单步运行

单步前进和单步回复可以在指令编辑画面和单步运行操作同样执行。

按下对应 "前进" 的功能键 ([F1])，则单步前进。按下对应 "后返" 的功能键 ([F4]) 键，则单步回复。当持续按下各功能键时，会在有光标的地方执行单步行。

中途放开功能键，则执行中断。另外，在机器人的移动中放开或强力按下 **Enable** 开关（3 位置开关）的话，会使伺服 **OFF**（关闭），将执行中断。

执行中操作面板的 **[START]** 开关的 LED 灯会亮。1 步编号的执行完毕，**[START]** 开关的 LED 灯会熄灭，**[步编号]** 开关的 LED 灯会亮。将示教单元画面的光标移动到下一个步编号。



步编号前进 **[F1]**（步编号后退 **[E4]**）

⚠ 注意

请特别注意动作中的机器人。在与周边装置的干涉等异常发生时，请以下列方式使机器人停止：

1. 请放开对应的功能键：“前进”或“后退”。
2. 放开或强按下 **Enable** 开关（3 位置开关）。

◇◆◇执行步编号的变更◇◆◇

可以使用箭头键移动光标及用跳转操作（“跳转”）变更执行步编号。
请参照后述的 [第 50 页的「\(4\)STEP 跳转」](#)。

(4) STEP 跳转

可以使用步编号运行变更程序号码。

以下为执行步编号 5 到步编号运行的操作示范。

1) 呼出步编号 5。

按下对应“跳转”的功能键（[F2]），进入步编号号码输入画面，在此画面按下 [5]、[EXE] 键，将光标移动到步编号 5。



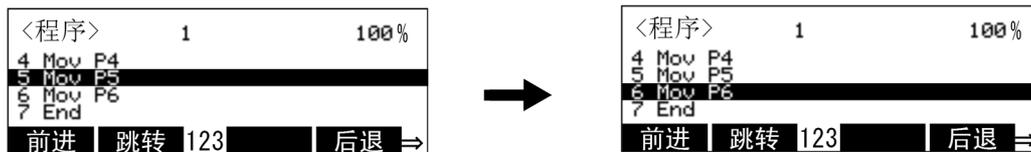
呼出 STEP 5 [F2]

呼出 STEP 5 [5] [EXE]

即使没有使用 [↑]、[↓] 键使光标在步编号 5 的情况下，也可以呼出步编号 5

2) STEP 前进

和上述的第 46 页的「(1) 单步前进」相同，按下对应“前进”的功能键（[F1]），则可由步编号 5 开始执行步编号前进。



步编号前进 [F1]

⚠ 注意

在变更执行步编号的步编号运行的情况下，请预先确认机器人和周边装置没有发生干涉。另外，请特别注意动作中的机器人。在与周边装置的干涉等异常发生时，请以下列方式使机器人停止：

1. 请放开对应的功能键：“前进”或“后退”。
2. 放开或强按下 Enable 开关（3 位置开关）。

3.7 自动运行

3.7.1 动作速度的设定

在控制器（驱动单元）的动作速度设定，请在 O/P（操作面板）或示教单元（T/B）进行。

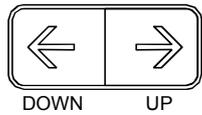
自动运行时的实际动作速度为：

动作速度 = （控制器（驱动单元）或示教单元的设置值） X （程序的设置值）。

(1) 在 O/P 的操作



速度比例的显示



速度比例的设定

1) 按下控制器（驱动单元）的 [CHNG DISP] 开关 2 次，STATUS 编号显示面板上会出现“速度比例”。

2) 每按下一次 [UP] 钮，则速度比例会依（10 → 20 → 30 → 40 → 50 → 60 → 70 → 80 → 90 → 100%）的顺序增加。按下 [DOWN] 钮，则会以相反顺序减少。

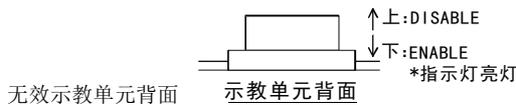
(2) 在示教单元的操作

1) 每按下一次 [OVRD ↑] 键则速度比例会以（3 → 5 → 10 → 30 → 50 → 70 → 100%）的顺序增加。

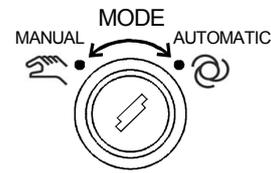
按下 [OVRD ↓] 键则会以相返顺序减少。

3.7.2 程序号码的选择

驱动器单元的准备



无效示教单元背面

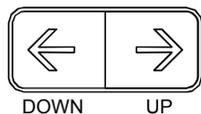


驱动单元有效

程序号码的选择



程序号码的显示



程序号码的显示

1) 使示教单元的 [ENABLE] 开关为 "DISABLE"。

2) 将控制器内的（驱动单元）的 [MODE] 开关设定为 "AUTOMATIC"。

3) 按下 [CHNG DISP] 钮，STATUS NUMBER 的显示面板上会出现“程序号码”。（开头以 "P" 表示。）

4) 按下 [UP] 钮则会依登录程序顺序往上显示、一按下 [DOWN] 钮则会以相反顺序向下显示。

※ 显示成为自动运行对象的程序号码。程序名为 5 个文字以上的情况下不会被显示。由外部机器选择的情况下，则会显示为 "P----"。

◇◆◇示教单元中的程序选择◇◆◇

示教单元的软件版本为 1.3 以上时，通过示教单元的操作可以进行程序选择。

操作方法的详细内容，请参照第 66 页的「(6) 程序的选择」。

3.7.3 自动运行的开始

(1) 通过 O/P 启动

**注意**

执行自动运行前，请务必确认下列项目。没有确认而执行自动运行的情况下，有可能造成物品损失及人员伤害。

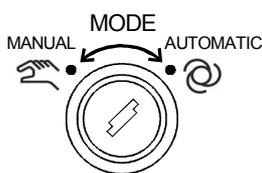
- 机器人的周围没有人员。
- 安全栅栏已上锁，防止不注意的人员接近。
- 机器人的安全范围内没有放置工具等无必要的物品。
- 工件放置在正确的位置。
- 确认在单步运行下，程序的动作正确。

控制器的准备

示教单元无效



1) 确认示教单元的 [ENABLE] 开关为 "DISABLE"。



控制器有效

2) 确认控制器（驱动单元）的模式为 "AUTOMATIC"。

开启伺服

SVO ON

开启伺服

3) 按下控制器（驱动单元）的 [SVO ON] 按钮。开启伺服电源。SVO ON 灯亮起。

自动运行的开始

START

开始（重复运行）



END

4) 按下控制器（驱动单元）的 [START] 按钮，则开始自动运行。（连续运行）
在连续运行的状态下按下 [END] 按钮，则停止在第 1 循环。在循环停止中 LED 灯会闪烁。在循环停止中再次按下 [END] 按钮会回复到连续运行。

**注意**

自动运行开始时，请务必确认已选择作为目的的程序号码。

**注意**

自动运行中请充分注意机器人的动作。
发现异常时，请按下 [EMG.STOP] 开关，立刻使机器人停止。

**注意**

刚开始自动运行时，请以低速状态动作，然后边确认与周边装置没有干涉，边慢慢地提升速度。

(2) 通过示教单元启动

示教单元软件 1.7 之后的版本，可通过在示教单元中的操作开始程序的自动运行。（R57TB：Ver. 3.0 以后）

操作通过 < 菜单 > → < 运行 > 中可选择的 <OPERATION> 画面进行。

通过参数：TBOP 的设定也可使本功能无效。



注意

“AUTOMATIC” 模式时在示教单元中显示 < 操作面板 > 画面时，操作权转移到示教单元。（示教单元的状态显示灯 [ENABLE] 闪烁）

此时，请注意无法进行通过控制器的 O/P（操作面板）及外部信号的需要操作权的操作。

（参照第 13 页的「2.3.1 关于操作权」）

以下所示为通过示教单元的 <OPERATION> 画面可执行的功能、及显示 <OPERATION> 画面时可执行的操作。

通过 <OPERATION> 画面执行的操作（各功能键）

- ①程序的选择 " 选择 "
- ②伺服 ON/OFF "SV.ON"/"SV.OFF"
(控制器的模式为 "AUTOMATIC" 时，ENABLE 开关无需 ON)
- ③自动运行的执行 " 启动 "
- ④运行模式（连续 / 循环）的变更 "CONT."/"CYCLE"
- ⑤程序复位 " 复位 "

其他按键操作

- ①动作速度的变更 [OVRD ↑] / [OVRD ↓] 键
注) 控制器的模式设定为 "AUTOMATIC" 时，以常规速度动作；设定为 "MANUAL" 时，以低速动作（用于动作确认）。
请将控制器切换到所需的模式。
- ②伺服 ON [SERVO] 键
(控制器的模式为 "AUTOMATIC 时，ENABLE 开关无需 ON)
- ③报警复位 [RESET] 键

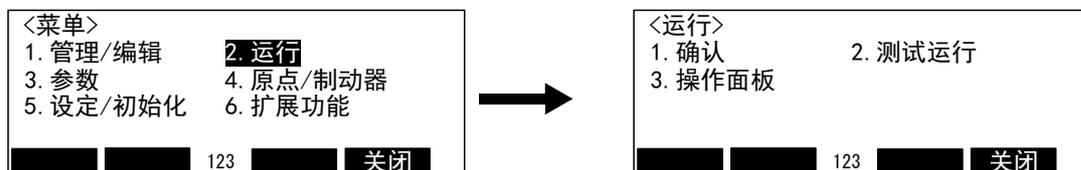
※ 通过 [EMG. STOP] 开关及 [STOP] 键停止机器人的功能始终有效。

通过 [JOG]、[HAND]、[MONITOR] 键进行的操作变为无效。

下面介绍通过示教单元执行自动运行时的操作方法。

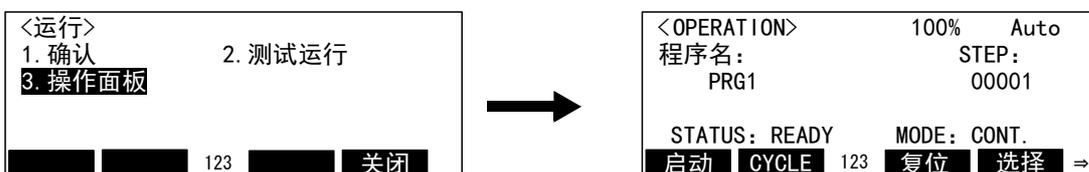
将控制器的模式设定为 "MANUAL" 执行自动运行（确认运行）时，请按下示教单元的 [TB ENABLE] 开关，将示教单元设为有效状态。

1) 在菜单画面中按 [2] 键，会显示运行菜单画面。



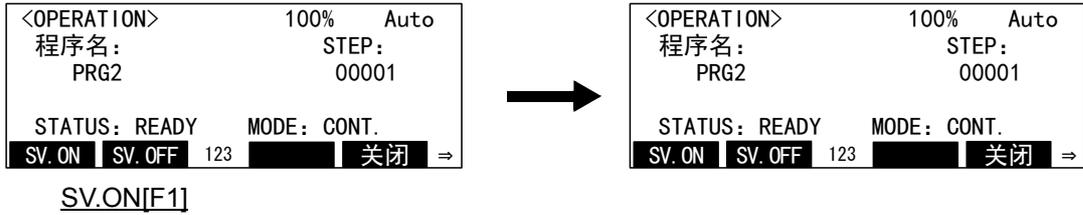
运行菜单的选择 [2]

2) 在运行菜单画面中按 [3] 键，会显示 <OPERATION> 画面。



操作面板的选择 [3]

5) 按下对应 "SV.ON" 的功能键 [F1], 开启伺服电源。(未显示 "SV.ON" 的功能时, 按 [FUNCTION] 键)

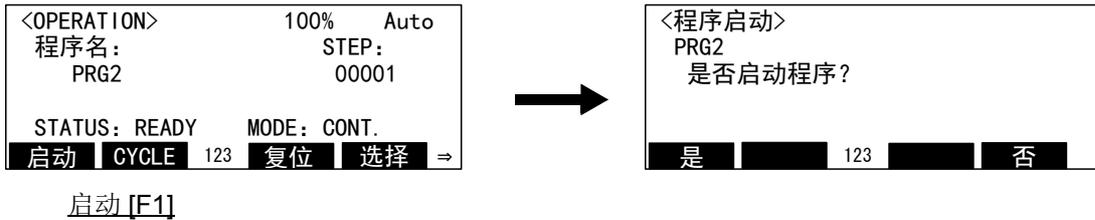


注意 自动运行开始时, 请务必确认已选择作为目的的程序号码。

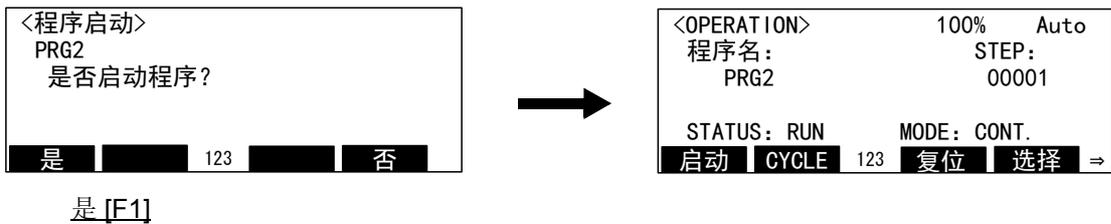
注意 自动运行中请充分注意机器人的动作。发现异常时, 请按下 [EMG.STOP] 开关, 立刻使机器人停止。

注意 刚开始自动运行时, 请以低速状态动作, 然后边确认与周边装置没有干涉, 边慢慢地提升速度。可通过示教单元的 [OVRD ↑][OVRD ↓] 键变更动作速度。

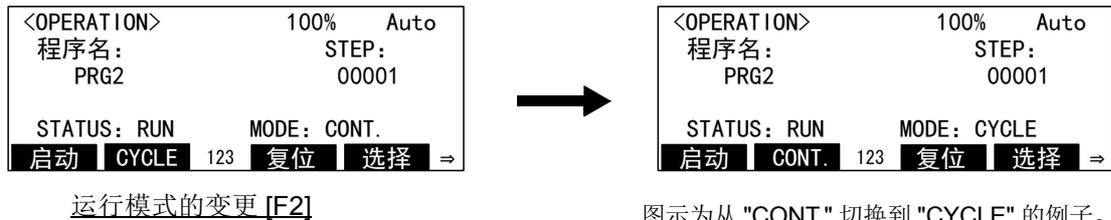
6) 按下对应 "启动" 的功能键 [F1], 即显示确认画面。



7) 按下对应 "是" 的功能键 [F1], 当前选择的程序即会开始自动运行。
返回到 <OPERATION> 画面。



8) 运行模式取决于画面上显示的模式。
请根据需要变更模式。



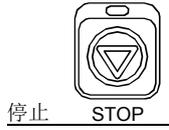
图示为从 "CONT." 切换到 "CYCLE" 的例子。

至此, 通过示教单元进行的自动运行的开始操作完成。

3.7.4 停止

使运行中的程序直接停止。将移动中的机器人减速停止。

(1) 在 O/P 的操作



1) 按下 [STOP] 钮。

(2) 在示教单元的操作

停止 [STOP]

1) 按下 [STOP] 键

◇◆◇停止操作与操作权无关◇◆◇

停止操作与操作权无关，会一直保持在有效状态。

3.7.5 停止之后重开启自动运行

(1) 右面板 (O/P) 再开

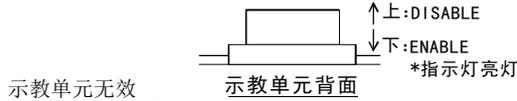


注意

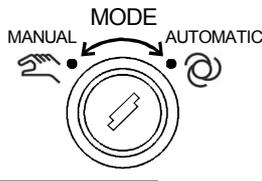
执行自动运行前，请务必确认下列项目。没有确认而执行自动运行的情况下，有可能造成物品损失及人员伤害。

- 机器人的周围没有人员。
- 安全栅栏已上锁，防止不注意的人员接近。
- 机器人的安全范围内没有放置工具等无必要的物品。
- 工件放置在正确的位置。
- 确认在单步运行下，程序的动作正确。

驱动单元的准备



1) 使示教单元的 [ENABLE] 开关为 "DISABLE"。



2) 将控制器内的 (驱动单元) 的 [MODE] 开关切换为 "AUTOMATIC"。

自动运行的再开



3) 按下控制器 (驱动单元) 的 [START] 按钮，重开启自动运行。(重复运行) 保持运行 /1 循环运行前的状态。



注意

重新开启自动运行时，请务必确认已选择做为目的的程序号码。



注意

自动运行中请注意机器人的动作。
发现异常的情况下，请直接按下 [EMG. STOP] 开关，使机器人停止。



注意

自动运行中请勿将机器人 CPU 的电源中断。中断的情况下，会使机器人 CPU 内的记忆异常，可能导致程序损坏。要使机器人立即停止的情况下，请使用紧急停止。

(2) 通过示教单元重启

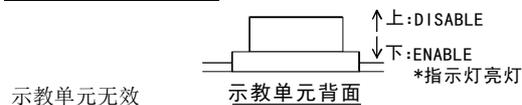
操作方法与自动运行的开始操作相同。请参照第 52 页的「3.7.3 自动运行的开始」，通过示教单元重启。

3.7.6 程序复位

解除程序的中断状态，后退到执行行的最前面。

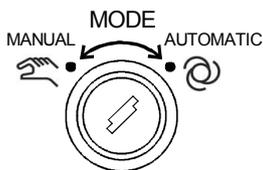
(1) 在操作面板（O/P）的操作

驱动器单元的准备



1) 将示教单元的 [ENABLE] 开关调为 "DISABLE"。

驱动器单元有效



2) 将控制器（驱动单元）的 [MODE] 开关调为 "AUTOMATIC"。

程序号码的表示



3) 按下控制器（驱动单元）的 [CHNG DISP] 钮，显示 "程序号码"

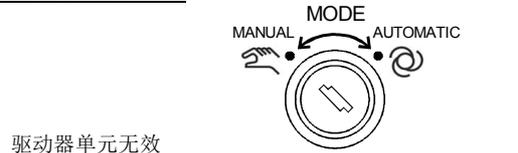
程序复位的执行



4) 按下控制器（驱动单元）的 [RESET] 钮。STOP 灯会熄灭，并解除程序的中断状态。

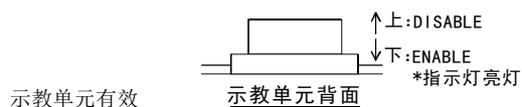
(2) 在示教单元的操作

示教单元的准备



驱动器单元无效

1) 将控制器（驱动单元）的 [MODE] 开关调为 "MANUAL"。



2) 将示教单元的 [ENABLE] 开关调为，"ENABLE"。

程序复位的执行

程序复位 [RESET] + [EXE]

3) 请按住 [RESET] 键后，按下 [EXE] 键。后退到执行行的最前面，使程序复位。

◇◆◇程序复位只在程序的中断中有效◇◆◇

程序运行中无法执行程序复位操作。请在中断中操作。

另外，从驱动器单元的操作面板上执行程序复位的情况下，请在 STATUS NUMBER 编号显示里显示 "程序号码" 后再做复位操作。

◇◆◇ STOP 灯灭◇◆◇

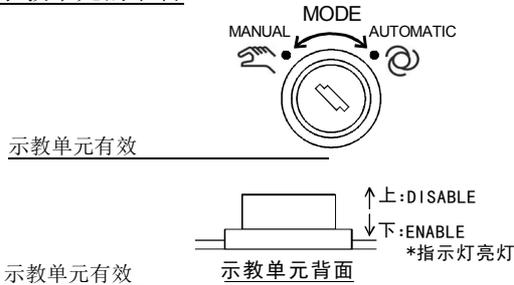
执行程序复位的话，STOP 灯不会亮起。

3.8 伺服ON/OFF

伺服电源在示教模式中，为了安全考虑，只有在轻押住示教单元里的使能开关（3位置开关）时，才会开启（ON）。在示教单元上的操作请在轻押住使能开关（3位置开关）的状态下执行。

■在示教单元的操作

示教单元的准备



伺服开启的执行

伺服 ON 操作 [SERVO]

伺服关闭的执行

伺服 OFF 操作，放开或强力按下使能开关（3位置开关）

1) 将控制器（驱动单元）的 "MANUAL" 开关。

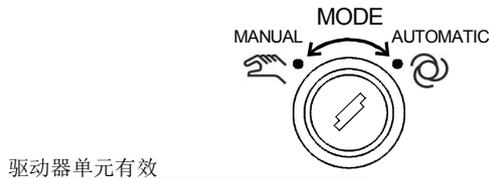
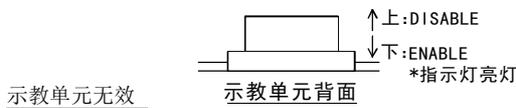
2) 将示教单元的 [使能] 开关调为 "ENABLE"。

3) 按下 [SERVO] 键使伺服开启。

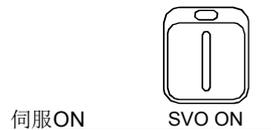
4) 放开或更加强力的按下使能开关（3位置开关）的话，则伺服关闭。

■在操作面板的操作

驱动单元的准备



伺服开启的执行



伺服关闭的执行



1) 将示教单元的 [ENABLE] 开关调为 "DISABLE"。

2) 将控制器（驱动单元）的 [MODE] 开关调为 "AUTOMATIC"。

3) 按下 [SERVO ON] 键使伺服开启，伺服 ON 灯亮。

4) 按下 [SERVO OFF] 键使伺服关闭，伺服 OFF 灯亮。

◇◆◇刹车动作◆◆◇

伺服关闭的话，刹车会自动的动作。依据机器人的机型，也有没有刹车的轴。

3.9 报警复位操作

■ 由操作面板的报警复位操作

解除报警



- 1) 按下控制器（驱动单元）的 [RESET] 钮。
示教单元侧的报警没有被复位的情况下，无法由示教单元执行复位操作。

■ 由示教单元的报警复位操作

解除报警

报警复位 [RESET]

- 1) 按下示教单元的 [RESET] 键。

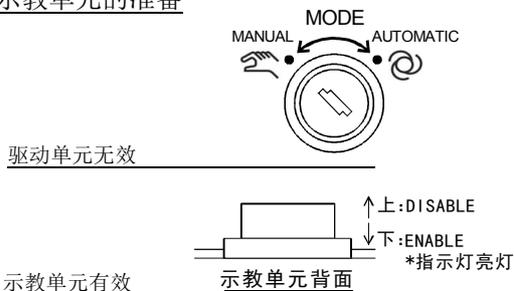
3.10 无法解除报警的暂时报警复位操作

依据机器人的机型别，轴在动作范围外的情况下，会发生无法解除的报警。此种情况下，一般的操作无法使伺服开启及 JOG 操作。可以下列步骤做暂时报警解除。动作范围外的情况等，暂时的解除报警的时候，请执行 JOG 操作，将轴移动到动作范围内。

此外 [RESET] 键相当于 R56/57TB 中的 [CAUTION] 键。使用 R56/57TB 时，请使用 [CAUTION] 键进行以下操作。

■ 由示教单元的暂时报警解除操作

示教单元的准备



- 1) 控制器（驱动单元）的 [MODE] 开关调为 "MANUAL"。

将示教单元的 [使能] 开关调为 "ENABLE"。

- 2) 轻握住使能开关（3 位置开关），一边按下 [SERVO] 键，一边持续按住 [RESET] 键。

解除报警 [SERVO] + [RESET]

依据上列操作可以暂时的复位报警。
因为放开键的话会再度发生报警，因此请不要放开按键。
执行 JOG 操作的情况，也请按住 [RESET] 状态下操作。

3.11 管理·编辑画面的操作

在此，对于程序管理相关做下列操作方法的说明

- (1) 程序一览表示
- (2) 复制程序（复制）
- (3) 变更程序名称（重命名）
- (4) 删除程序（删除）
- (5) 保护（Protect）
- (6) 程序的选择

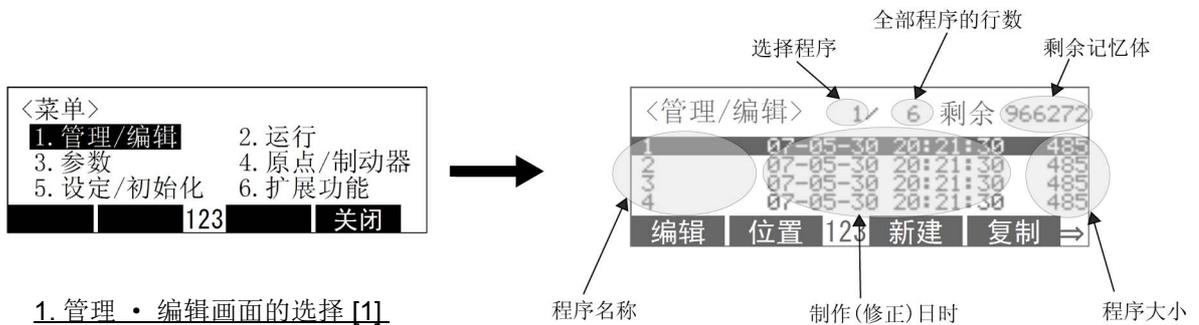
关于程序的编辑方法，请参照第 34 页的「3.5 程序编辑」。

(1) 程序一览表示

显示机器人 CPU 内的程序登录状态画面。

1) 管理·编辑菜单的选择。

在菜单画面按下 [1] 键。选择“1. 管理·编辑”，显示程序的一览表。



也可以使用 [↑]、[↓] 键，使光标对准“1. 管理·编辑”后按下 [EXE] 键来执行。

在程序一览画面按下 [↑]、[↓] 键的话，会显示其它的程序。

此外，可以将程序选择为各操作的对象。

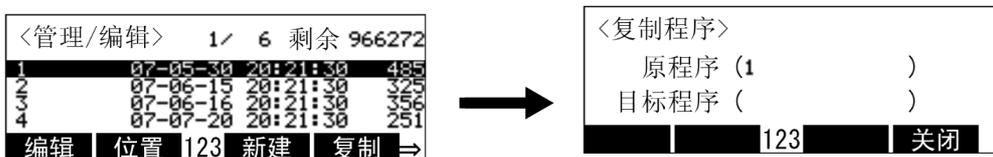
画面的下方有显示对应功能键的选单（“编辑”、“位置”、“新建”、“复制”）。里面的“编辑”、“位置”、“新规”为程序的编辑功能。关于这些操作方法请参照第 34 页的「3.5 程序编辑」。

按下 [FUNCTION] 键的话，会显示“重命名”、“删除”、“保护”、“关闭”。

(2) 复制程序（复制）

1) 选择复制 MENU

在程序一览显示里按下对应“复制”的功能键（[F4]）。显示复制画面。

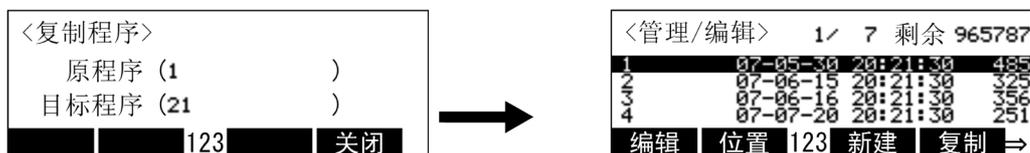


复制的选择 [F4]

2) 复制完毕程序的指定和执行

在原副本的括号内，会显示在事先程序一览表示里已选择的程序名称。（在图里程序名称“1”），要变更的情况下，使用箭头符号键移动光标即可变更。

在原副本的括号内输入预先复制起来的程序名称，然后按下 [EXE] 键。



程序的名称 [2] [1] [EXE]

此情况下，从程序“1”复制到“2 1”。

◇◆◇保护（保护）设定无法被复制◇◆◇

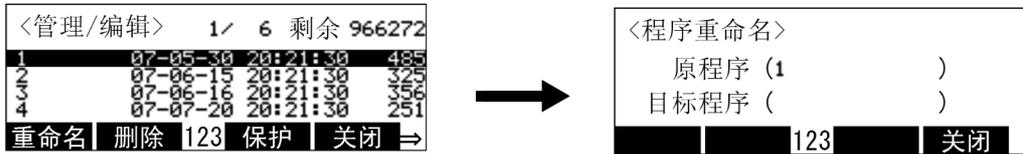
复制操作里，程序保护数据及变量保护数据无法被复制。

必要时请再度设定。关于保护（保护）的设定方法请参照第 65 页的「(5) 保护（Protect）」。

(3) 变更程序名称（重命名）

1) 重新命名菜单画面的选择

在程序一览显示里按下对应“重命名”的功能键（[F1]）。显示复制画面。“重命名”里菜单没有显示的情况下，请按下 [FUNCTION] 键使菜单显示。

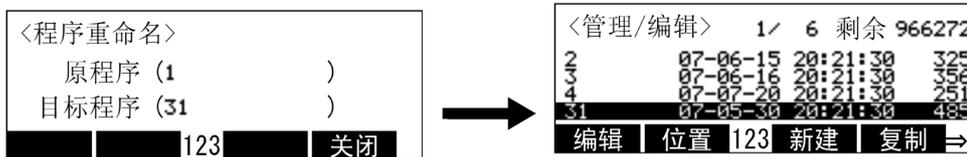


重命名的选择 [F1]

2) 指定要变更名称的程序

在原重新命名的括号内，会显示在事先程序一览表示里已选择的程序名称。（在图里程序名称“1”），要变更的情况下，使用箭头符号键移动光标即可变更。

在原重新命名的括号内输入新的程序名称，然后按下 [EXE] 键。



此情况下，从程序“1”重新命名为“31”。

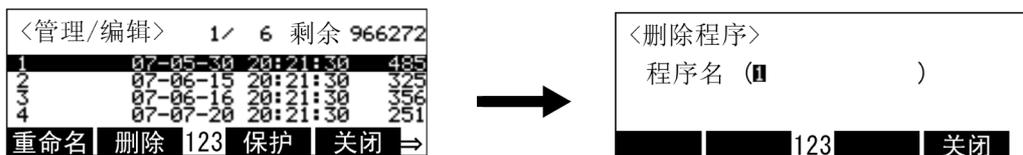
◇◆◇保护（保护）设定无法被变更程序名◇◆◇

保护命令设定后，对应的程序名称不可被变更。请解除保护命令。保护命令的解除方法请参照第 65 页的「(5) 保护 (Protect)」。

(4) 删除程序（删除）

1) 删除菜单的选择

按下在程序一览显示里对应“删除”的功能键（[F2]）。会显示删除画面。没有显示“删除”菜单的情况下，请按下 [FUNCTION] 键，使菜单显示出来。

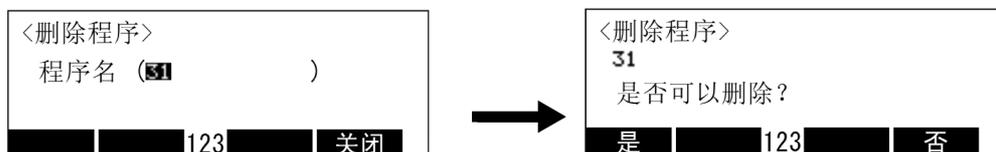


删除的选择 [F2]

2) 删除程序的指定

在“程序名”的括号内，会显示在事先程序一览表示里已选择的程序名称。（在图里程序名称“1”），要变更的情况下，请输入正确的程序名称。

按下 [EXE] 键的话，会显示确认画面。



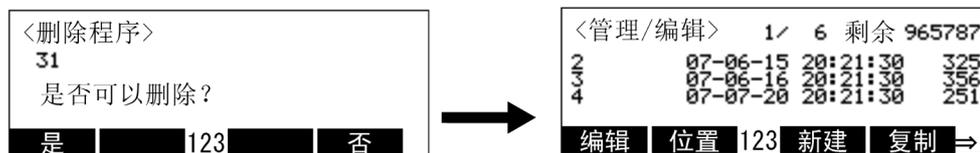
程序名称的设定 [3] [1] [EXE]

此情况下，删除程序名称“31”。

3) 程序的删除

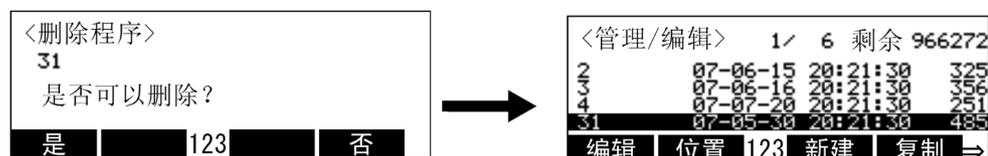
按下对应“是”的功能键，删除指定程序且会后退到程序一览显示。

没有显示删除的情况下，请按下对应“否”的功能键。后退到删除画面。



删除 [F1]

此情况下，删除程序名称“31”。



不删除 [F4]

此情况下，程序无法被删除。

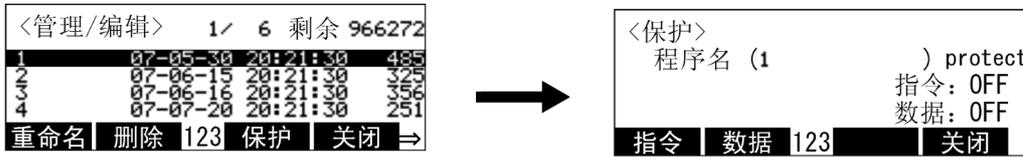
◇◆◇保护（保护）设定程序无法被删除◇◆◇

保护程序被设定后，程序无法被删除。请解除保护命令。保护命令的解除方法请参照「(5) 保护 (Protect)」。

(5) 保护 (Protect)

1) 保护菜单的选择

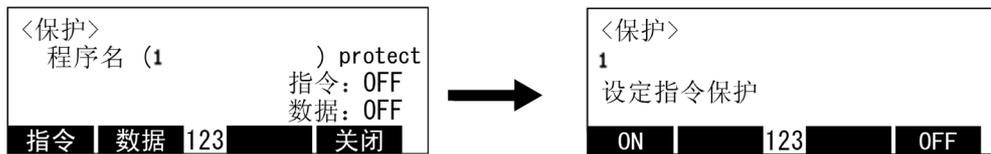
在保护一览显示里，按下对应“保护”的功能键 ([F3])。会显示保护画面。
 没有显示“保护”菜单的情况下，请按下 [FUNCTION] 键，使菜单显示。



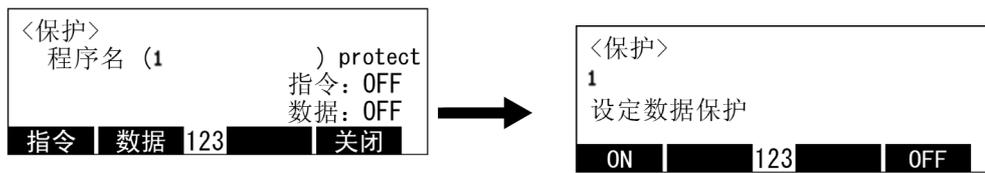
菜单的选择 [F3]

2) 保护的设定

程序的保护，指令和数据（变量值）可以个别的指定。设定指令的保护指定的情况下，请按下对应“指令”的功能键、设常数据的保护指定的情况下，请按下对应“数据”的功能键。

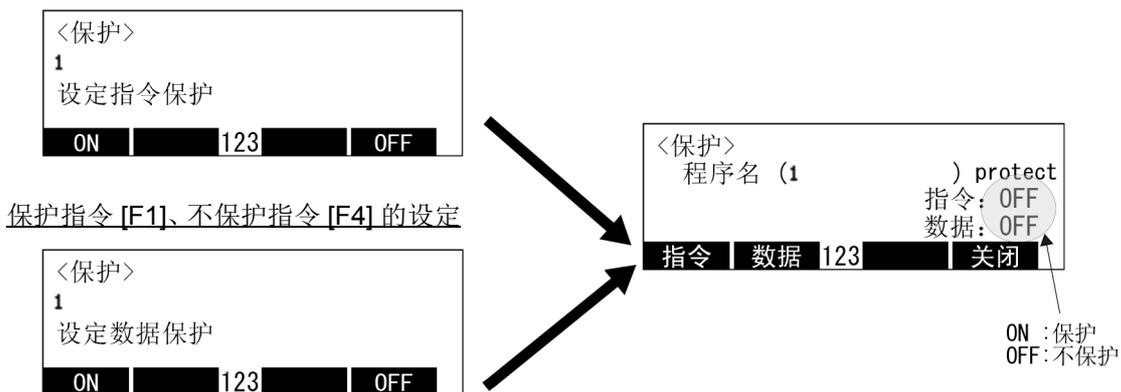


指令的保护指定 [F1]



数据的保护指定 [F2]

分别按下对应“ON”、“保护”、“OFF”功能键的话，“不保护”会被设定。



保护指令 [F1]、不保护指令 [F4] 的设定



保护数据 [F1]、不保护数据 [F4] 的设定

◆◆◆所谓指令保护◆◆◆

保护程序的删除、名称变更、名称变更及指令的变更因错误操作的功能。

- 在复制操作，保护数据无法复制。
- 在初始化操作，会忽略保护数据，进行初始化。

◆◆◆所谓数据保护◆◆◆

依据错误位置数据的登录、变更及依据错误程序的执行，保护由各变量所代入的变量的功能。

- 在复制操作，保护数据无法复制。
- 在初始化操作，会忽略保护数据，进行初始化。

(6) 程序的选择

本功能在 1.3 版本以后的示教单元中可使用。

可以选择作为单步运行和自动运行对象的程序。

和控制器的操作面板中的「3.7.2 程序号码的选择」的功能相同，选择程序后操作面板也将显示“STATUS NUMBER”的程序编号。

操作方法如下所示。

1) 程序选择

在程序一览显示中，将光标对准通过 [↑]、[↓] 键操作进行选择的程序。

<管理/编辑>				1/	6	剩余	966272
1	07-05-30	20:21:30	485				
2	07-06-15	20:21:30	325				
3	07-06-16	20:21:30	356				
4	07-07-20	20:21:30	251				
重命名			删除	123	保护	关闭	⇒

程序的选择 [↑]、[↓]

<管理/编辑>				1/	6	剩余	966272
1	07-05-30	20:21:30	485				
2	07-06-15	20:21:30	325				
3	07-06-16	20:21:30	356				
4	07-07-20	20:21:30	251				
重命名			删除	123	保护	关闭	⇒

图为选择了程序 3 的示例。

2) 选择程序的确定

按下示教单元的 [FUNCTION]+[EXE] 键。显示确认画面。

<管理/编辑>				1/	6	剩余	966272
1	07-05-30	20:21:30	485				
2	07-06-15	20:21:30	325				
3	07-06-16	20:21:30	356				
4	07-07-20	20:21:30	251				
重命名			删除	123	保护	关闭	⇒

<程序选择>	
3	选择这个程序到SLOT1中?
是	123
否	

选择程序的确定 [FUNCTION]+[EXE]

确认显示中的程序名（编号）并执行选择时，按下对应“是”的功能键（[F1]）。

要中止时请按下对应“否”的功能键（[F4]）。将返回到程序一览显示。

<程序选择>	
3	选择这个程序到SLOT1中?
是	123
否	

程序选择的执行: [F1] (是)
中止: [F4] (否)

<管理/编辑>				1/	6	剩余	966272
1	07-05-30	20:21:30	485				
2	07-06-15	20:21:30	325				
3	07-06-16	20:21:30	356				
4	07-07-20	20:21:30	251				
重命名			删除	123	保护	关闭	⇒

至此，程序的选择完成。

3.12 运行画面的操作

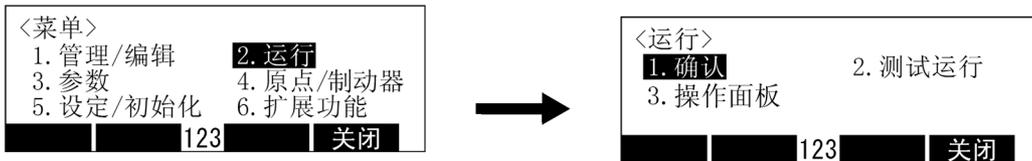
下列，说明各功能的操作方法。

- (1) 执行行的显示 1. 确认 : 显示执行中的程序的执行行及执行单步运行。
- (2) 单步运行画面 2. 单步运行 : 显示已选择的程序名称及执行中的步号。
程序执行中，将连续运行模式切换到循环停止模式。

3.12.1 执行行的显示

(1) 确认画面的选择

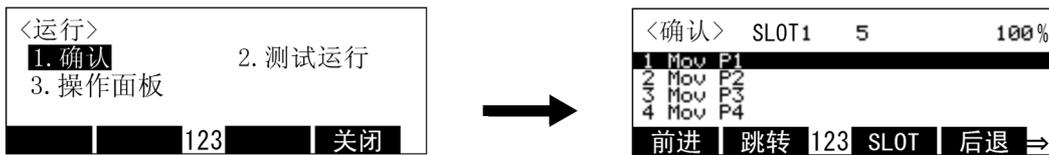
1) 在菜单画面按下 [2] 键的话，会显示运行菜单画面。



运行菜单的选择 [2]

2) 在菜单画面里按下 [1] 键的话，会显示确认画面。

在画面里显示设定在插槽 1 里的程序。在插槽号码里显示连续的程序名称。



确认的选择 [1]

程序执行中，在执行行移动光标。

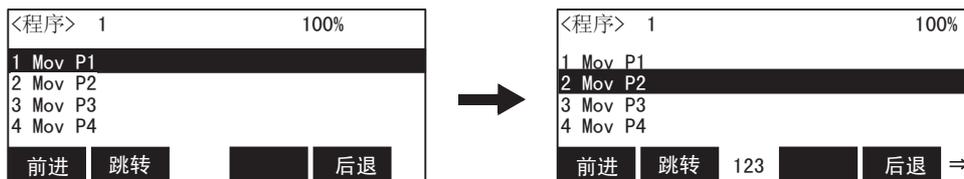
(2) 单步运行

执行和 [第 46 页的「3.6 排错操作」](#) 里显示的，和单步及前进单步后退相同的操作。

1) 单步前进

在确认画面按下对应 "前进" 的功能键 ([F1]) 的话，只有当按住的时候，会执行有光标的光标。中途放开功能键的话，执行会中断。另外，当机器人移动中放开或更强力按下使能开关 (3 位置开关)，也可以中断执行。

执行中操作面板的 [START] 开关的 LED 灯会亮。1 个单步的执行完毕的话，[START] 开关的 LED 灯会熄灭，且 [STOP] 开关的 LED 灯会亮。示教单元画面的光标会往下一个单步移动。



单步前进 [F1]

每按下功能键 ([F1])，会往下一个单步前进。



注意

工作中请特别注意机器人的动作。在与周边装置的干涉等异常发生时，请放开对应的功能键、放开或强按下 **Enable** 开关（3 位置开关）。使机器人停止。

◇◆◇所谓 STEP 运行◇◆◇

将程序 1 个 STEP 1 个 STEP 的执行，则称为 STEP 运行。动作速度缓慢，因为执行每一 STEP 就停止，有效的进行程序的确认及动作位置的确认。

执行有光标的 STEP。

执行 END 指令及 HLT 指令的话，超过的 STEP 不会前进。

◇◆◇执行 STEP 的变更◇◆◇

使用箭头符号键移动光标及用跳转操作（“跳转”），皆可以变更 STEP。

◇◆◇将动作中的机器人直接停止◇◆◇

①按下 [紧急停止] 开关。

将伺服关闭且将在动作中发生紧急停止报警的机器人直接停止。继续请以复位报警 → 伺服 ON → STEP 运行操作的顺序进行。

②放开或者是更强大的按下 **Enable** 开关（3 位置开关），会使伺服 OFF 且将动作中发生报警 2000 的机器人直接停止。继续请以复位报警 → 伺服 ON → STEP 运行操作的顺序

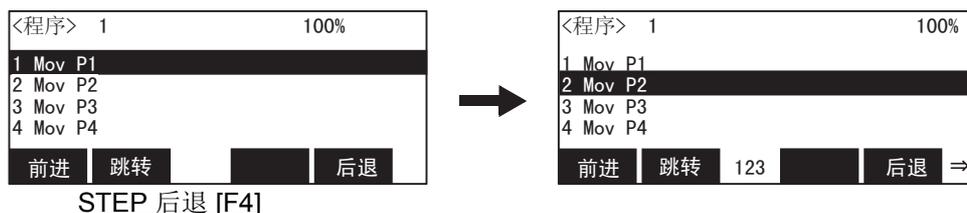
2) 单步后退

将完成 STEP 前进及自动运行的 STEP、或者将中断的 STEP，限制在插补指令往返回方向各 1 个 STEP。

按下对应“后退”的功能键（[F4]）的话，只有持续按住的时候，往 STEP 的返回方向只执行 1 个 STEP。

中途放开功能键的话，执行会中断。另外，当机器人移动中放开或更强力按下 **Enable** 开关（3 位置开关），伺服关闭且可以中断执行。

执行中操作面板的 [START] 开关的 LED 灯会亮。1 个 STEP 的执行完毕的话，[START] 开关的 LED 灯会熄灭，且 [STOP] 开关的 LED 灯会亮。示教单元画面的光标在 STEP 的返回方向，往下一个插补指令的 STEP 移动。



每按下功能键（[F4]），会返回到前面的 STEP。



注意

动作中请特别注意机器人的动作。在与周边装置的干涉等异常发生时，请放开对应的功能键、放开或强按下 **Enable** 开关（3 位置开关）。

使机器人停止。

◇◆◇将动作中的机器人直接停止◇◆◇

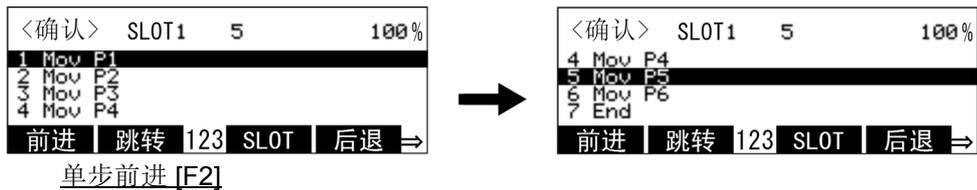
- ① 按下 [紧急停止] 开关。
将伺服关闭且将在动作中发生紧急停止报警的机器人直接停止。继续请以复位报警 → 伺服 ON → 单步运行操作的顺序进行。
- ② 放开或者是更强大的按下使能开关（3 位置开关），会使伺服 OFF 且将动作中发生报警 2000 的机器人直接停止。继续请以复位报警 → 伺服 ON → 单步运行操作的顺序。

(3) 单步跳转

可以变更执行单步运行的步号。
以下为执单步行 5 到单步运行情况下的操作示范例。

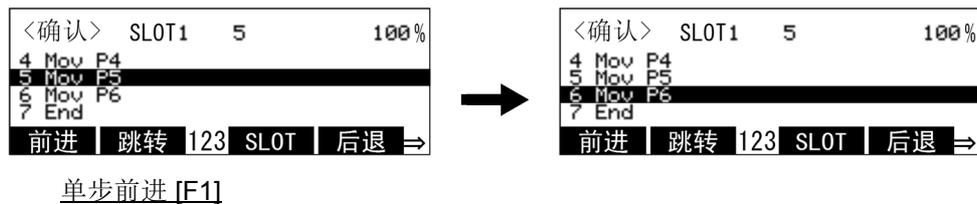
1) 呼出单步 5。

按下对应“跳转”功能键的（[F2]），在显示步号输入画面里按下 [5]、[EXE] 键。
光标移动到单步 5。



2) 单步前进

和上述第 67 页的「(2) 单步运行」相同的，按下对应“前进”的功能键（[F1]），则可以执行从单步 5 到单步前进。



⚠ 注意

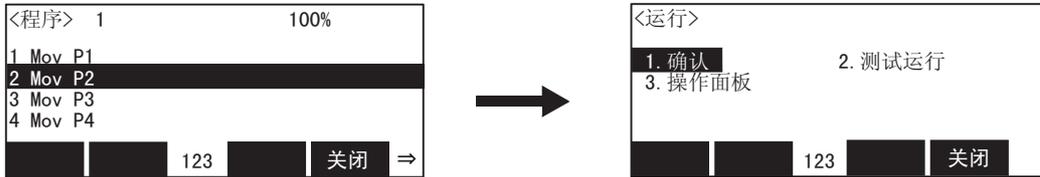
将执行的单步变更后，执行单步运行的情况下，请确认机器人不会与周边装置等发生干涉。
动作中请特别注意机器人的动作。在与周边装置的干涉等异常发生时，请放开对应的功能键、放开或强按下使能开关（3 位置开关）。
使机器人停止。

(4) 在其它插槽的单步前进

确认多任务的程序的情况下，可以在确认画面里做单步前进。操作方法请参照第 70 页的「(4) 在其它插槽的单步前进」。

(5) 确认画面的结束

1) 按下对应“关闭”的功能键（[F4]）的话，会后退到 MENU 运行画面。

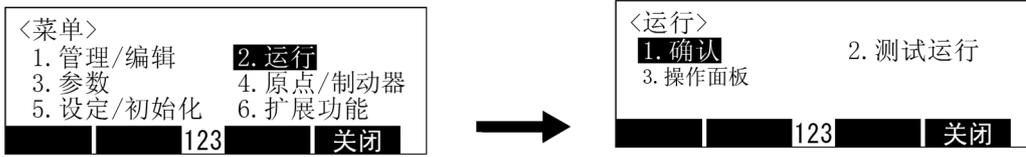


单步运行画面的结束 [F4]

3.12.2 单步运行画面

(1) 单步运行画面的选择

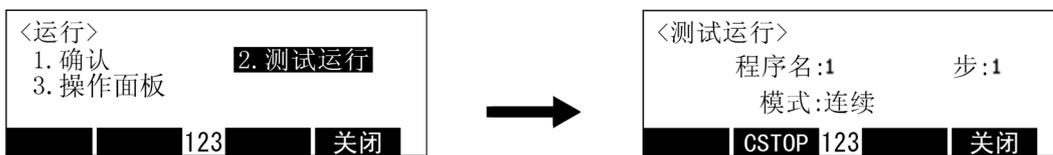
1) 在菜单画面按下 [2] 键的话，会显示运行菜单画面。



运行菜单的选择 [2]

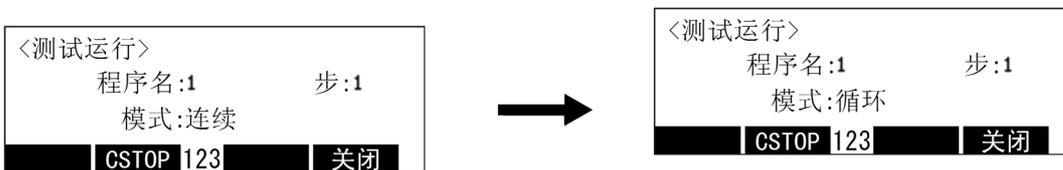
2) 在运行菜单画面按下 [2] 键的话，会显示单步运行画面。

显示在画面插槽 1 里设定的程序名及执行步号、以及现在的运行模式。



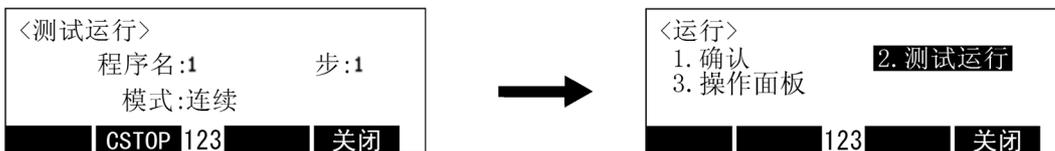
测试运行的选择 [2]

3) 在程序执行中按下对应“CSTOP”的功能键（[F2]）的话，可以将连续运行模式切换到循环运行模式。模式的后面显示“循环”的话，驱动单元操作面板的 [END] 钮灯会熄灭，且在指令或者程序的最终行执行后，程序的运行结束。



变更为循环运行模式 [F2]

4) 按下对应“关闭”的功能键（[F4]）的话，会后退到运行菜单画面。



测试运行画面的结束 [F4]

◇◆◇将程序的执行停止及变成连续运行模式◇◆◇

在循环运行模式中，按下 [STOP] 键，使程序的执行停止的情况下，会切换到连续运行模式。用循环模式继续执行程序的情况下，请按下 [START] 键后再次按下 [F4] 键。（即使按下驱动单元操作面板的 [END] 钮，也会变成循环运行模式）

3.12.3 操作面板画面的操作

进行伺服电源的 ON/OFF、程序的选择和自动运行的启动等。

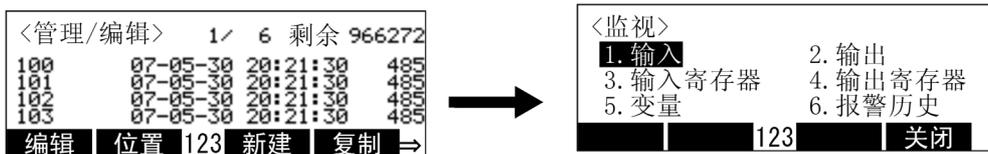
详细内容请参照第 52 页的「3.7.3 自动运行的开始」中的「(2) 通过示教单元启动」。

3.13 监视画面的操作

下列为功能的操作方法说明。

- (1) 输入信号的监视 1. 输入 : 参数输入信号的监视。
- (2) 输出信号的监视 2. 输出 : 参数输出信号的监视和 ON/OFF 的设定。
- (3) 输入寄存器的监视 3. 输入寄存器 : 使用 CC-Link 时, 监视输入寄存器的值。
- (4) 输出寄存器的监视 4. 输出寄存器 : 使用 CC-Link 时, 监视输出寄存器的值。
- (5) 变量的监视 5. 变量 : 在程序里使用的变量值的监视及值的设定。
- (6) 报警的历史显示 6. 报警历史 : 显示已发生报警的历史。

上述的任一个操作都可以在按下示教单元的 [MONITOR] 键, 显示监视菜单画面后进行。示教单元没有操作权的情况下也可以执行的功能。那时显示的画面是任意无关的, 在程序 (指令) 编辑画面, 没有变量监视功能。



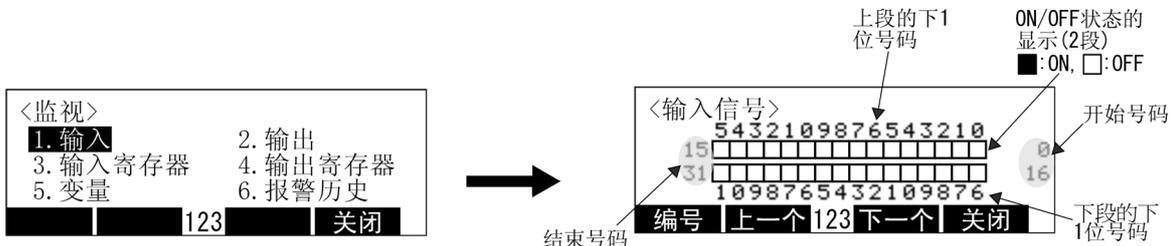
监视画面的选择 [MONITOR]

※ 显示的画面为随机。

(1) 输入信号监视

将从外部来的输入信号的 ON/OFF 状态做通常监视功能。

- 1) 在监视 MENU 画面按下 [1] 键的话, 会显示输入信号的监视画面。可以在 1 画面将 32 点的输入信号监视。

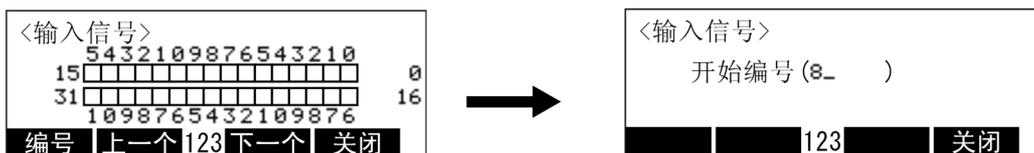


输入画面的选择 [1]

以下为确认输入号码由 8 变成 15 的状态的状况。

- 2) 按下对应 “号码” 功能键 [F1]。

因为会显示开始号码指定画面, 在开始号码设定为 “8”。(按下 [8]、[EXE] 键)



信号号码的设定 [F1]

开始号码的设定 [8]、[EXE]

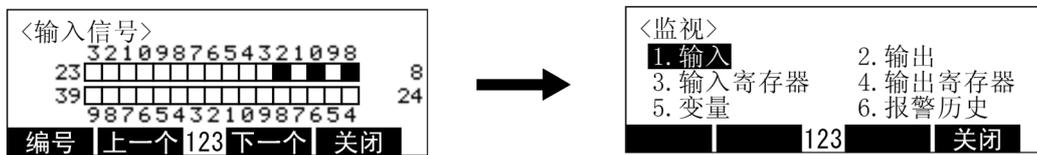
3) 在输入号码 8 号的前面，显示 32 点的 ON/OFF 状态。涂黑表示为 ON、返白表示为 OFF。



在图里，8,10,12 号为 ON，其它显示为 OFF。

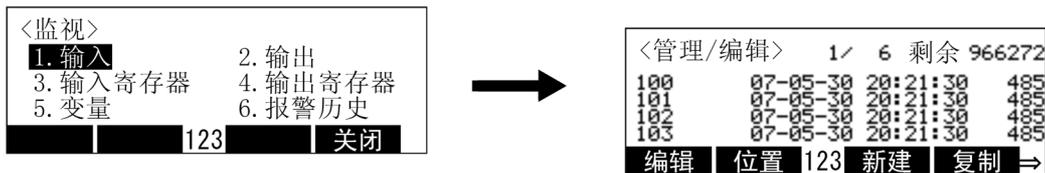
按下对应“其次”的功能键（[F3]）的话，会显示下一个输入信号的状态（下个画面），按下对应“前”的功能键（[F2]）的话，会显示前个输入信号的状态（前画面）。

4) 按下对应“关闭”的功能键（[F4]）的话，会后退到监视菜单画面。



信号号码的设定 [F4]

5) 在监视菜单画面按下对应“关闭”的功能键 [F4] 的话，监视会结束且后退到原始的画面。



信号号码的设定 [F4]

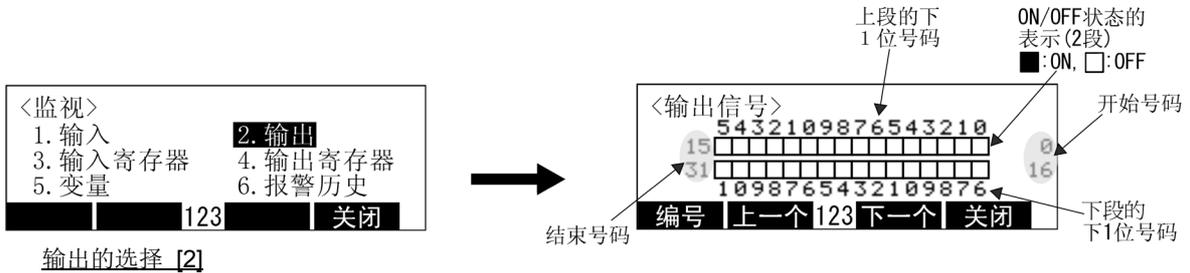
◇◆◇监视的结束◆◆◇

在输入寄存器的状态显示等的监视画面显示中，按下 [MONITOR] 键，无论何时都可以将监视结束且回到原本的画面。

(2) 输出信号监视

通常监视往外部的输出信号的 ON/OFF 状态，有将信号强制输出的功能。

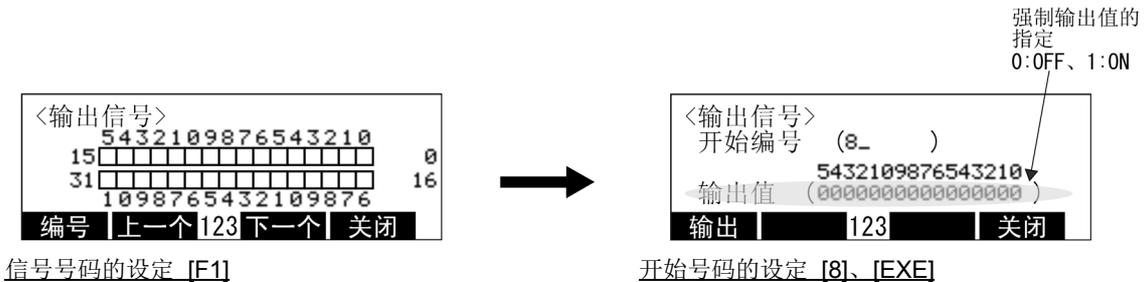
1) 在监视菜单画面按下 [2] 键的话，会显示输出信号的监视画面。在 1 画面可以监视 32 点的输出信号。



以下为确认输出码由 8 变成 15 的状态的状况。

2) 按下对应“号码”的功能键 [F1]。

因为会显示设定开始号码及输出值的画面，将开始号码设定为“8”。（按下 [8]、[EXE] 键）



在显示上的输出值，会显示现在的输出状态，但是在此，在设定信号的强制输出值部份，不会变成通常显示。

按下对应“关闭”的功能键 ([F4]) 的话，在输出号码 8 号的前面，显示 32 点的输出信号监视画面涂黑表示为 ON、返白表示为 OFF。



在图里，8 号为 ON，除此以外皆显示为 OFF。

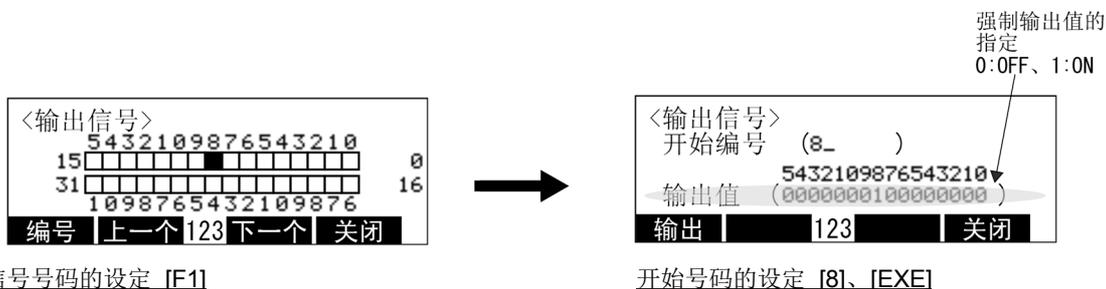
按下对应“其次”的功能键 ([F3])，则显示下个信号的状态（下个画面），按下对应“前”的功能键 ([F2])，则显示下个输入信号的状态（前画面）。

3) 输出信号的强制输出。

下列为将输出信号 8 号强制的关闭情况下的操作方法。

按下对应“号码”的功能键 ([F1])。

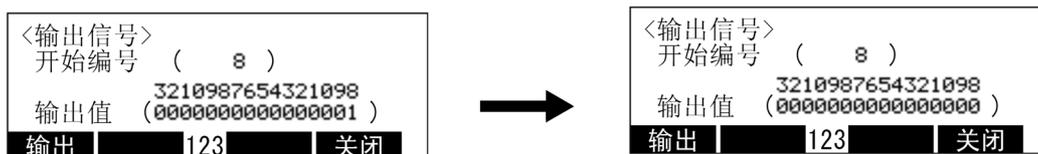
因为会显示设定开始号码及输出值的画面，在开始号码设定为“8”。(按下 [8]、[EXE] 键)



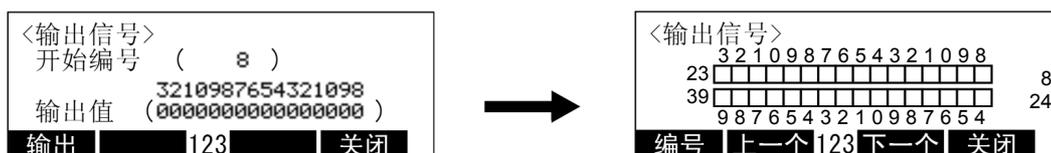
4) 使用箭头符号键，将光标移动到输出值的“8”的位置。

现在，因为输出信号 8 号码为开启，会显示值“1”。

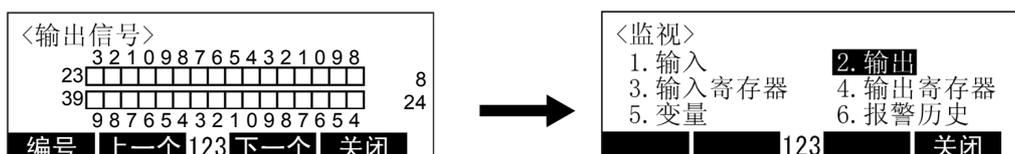
将此值关闭以变更显示为“0”([0] 键)，按下对应“出力”的功能键 ([F1]) 的话，实际上的同输出信号会变成关闭。



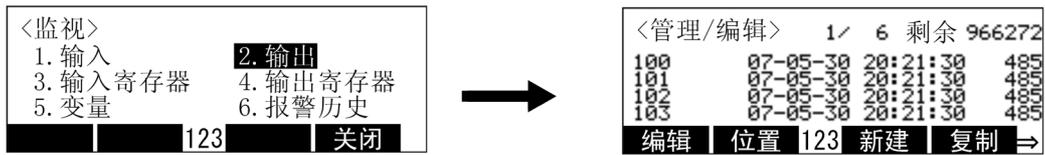
5) 按下对应“关闭”的功能键 ([F4]) 的话，会后退到原始的输出信号监视画面。



6) 按下对应“关闭”的功能键 ([F4]) 的话，会后退监视菜单画面。



7) 在监视菜单画面按下对应“关闭”的功能键 [F4] 的话，监视会结束且会后退到原始的画面。



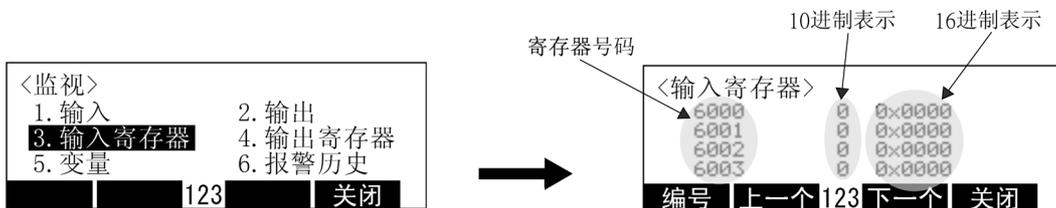
信号号码的设定 [F4]

(3) 输入寄存器监视

在使用 CC-Link 使用时，将输入寄存器的值做通常监视的功能。

注) 在 CR7xx-Q 系列，因为没有 CC-Link 配件，所以无法使用本功能。

1) 在监视菜单画面按下 [3] 键的话，会显示输入寄存器的监视画面。在 1 画面里可以监视 4 个的输入寄存器。

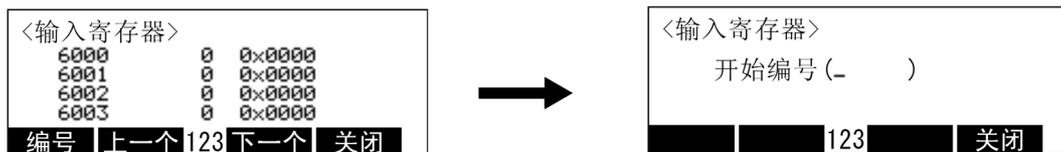


输入寄存器的选择 [3]

下列示范为确认输入 Register8000 号的状态。

2) 按下对应“号码”的功能键 [F1]。

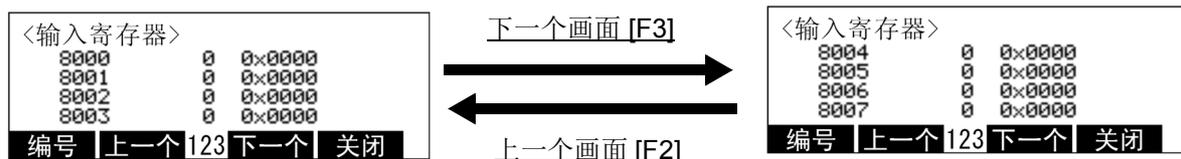
因为会显示开始号码指定画面，在开始号码上设定“8000”（按下 [8]、[0]、[0]、[0]、[EXE] 键）



寄存器号码的设定 [F1]

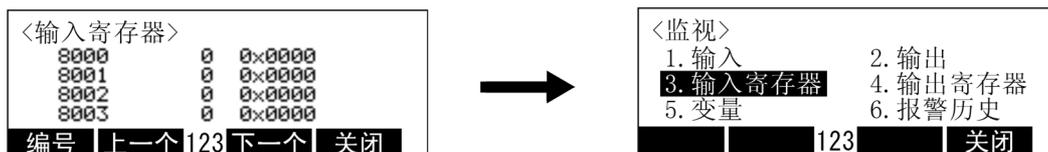
开始号码的设定 [8]、[0]、[0]、[0]、[EXE]

3) 在输入寄存器号的前面，显示 4 个寄存器的值。



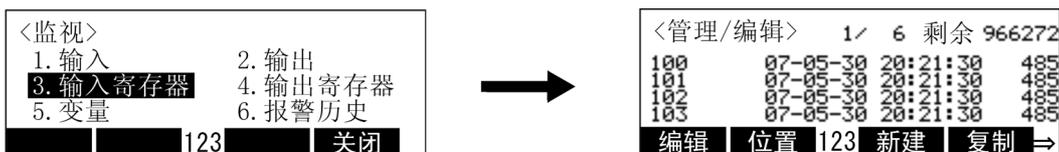
按下对应“其次”的功能键 ([F3])，会显示下一个输入寄存器的状态 (下个画面)，按下对应“前”的功能键 ([F2]) 的话会显示前一个输入寄存器的状态 (前画面)。

4) 按下对应“关闭”的功能键 ([F4]) 的话, 会后退监视菜单画面。



结束 [F4]

5) 在监视菜单画面按下对应“关闭”的功能键 [F4] 的话, 会后退到监视结束前的画面。



信号号码的设定 [F4]

◇◆◇监视的结束◇◆◇

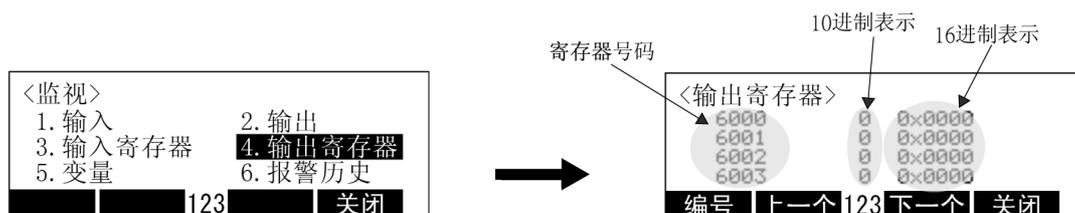
在输入寄存器的状态显示等的监视画面显示中, 按下 [MONITOR] 键, 无论何时都可以将监视结束且回到原本的画面。

(4) 输出寄存器监视

在使用 CC-Link 使用时, 将输出寄存器的值做经常监视功能。

注) 在 CR7xx-Q 系列, 因为没有 CC-Link 配件, 无法使用本功能。

1) 在监视菜单画面按下 [4] 键的话, 会显示输出寄存器的监视画面。在 1 画面里可以监视 4 个的输出寄存器。



输入寄存器的选择 [4]

下列示范为确认输出。Register 8000 号的状态。

2) 按下对应“号码”的功能键 [F1]。

因为会显示开始号码指定画面，在开始号码上设定“8000”。（按下 [8]、[0]、[0]、[0]、[EXE] 键）

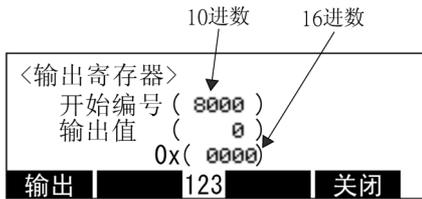


寄存器号码的设定 [F1]



开始号码的设定 [8]、[0]、[0]、[0]、[EXE]

3) 8000 号的现在输出值，在输出值的连续括号内以十进制数表示。在下面的 0x 里连续括号的值为 16 进数。



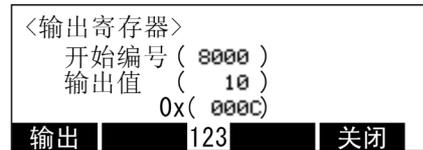
按下对应“关闭”的功能键 ([F4]) 的话，可以以输出寄存器的 8000 号为起点，后退到监视画面，但是，可以在现在的 8000 画面里，变更输出值。以下为将输出 Register 8000 号的值，设定为 12（10 进数）的示范。

4) 值的设定可以是 10 进数或 16 进数。

用 10 进数设定的情况下，使用箭头符号，将光标移动到输出值，输入“10”。按下 ([1]、[0] 键) 不要的文字按下 [CLEAR] 键会删除。

用 16 进数设定的情况下，使用箭头符号，将光标移动到 0x，输入“C”。按下 ([C] 键) 不要的文字按下 [CLEAR] 键会删除。

按下对应“出力”的功能键 ([F1]) 的话，会实际输出已设定的值。



输出值的设定

用 10 进数指定 [矢印]、[1]、[0]

用 16 进数指定 [矢印]、[C]

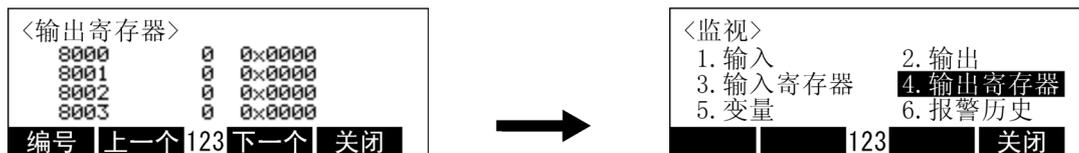
设定值的输出 [F1]

5) 按下对应“关闭”的功能键 ([F4]) 的话，会后退到原始的输出寄存器监视画面。



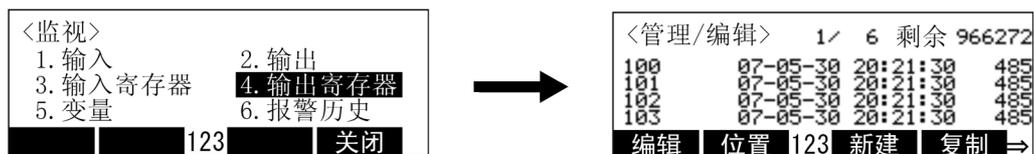
强制输出的结束 [F4]

6) 再一次按下对应“关闭”的功能键（[F4]）的话，会后退到原始的菜单画面。



输出寄存器监视的结束 [F4]

7) 在监视菜单画面按下对应“关闭”的功能键 [F4]，会将监视结束且后退到原始的画面。



信号号码的设定 [F4]

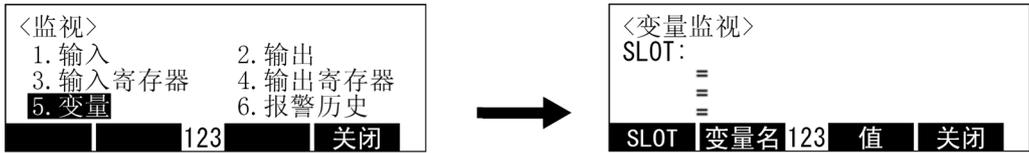
◇◆◇监视的结束◇◆◇

在输入寄存器的状态显示等的监视画面显示中，按下 [MONITOR] 键，无论何时都可以将监视结束且回到原本的画面。

(5) 变量监视

显示在程序里使用的变量内容或变更的功能。

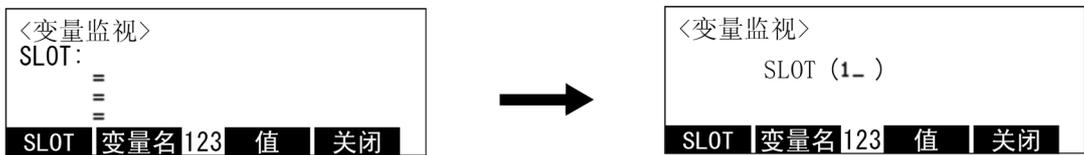
1) 在监视菜单画面按下 [5] 键的话，会显示变量监视画面。



变量的选择 [5]

2) 将要做为监视对象的程序用插槽号码指定。
按下对应“SLOT”的功能键，然后输入插槽号码。

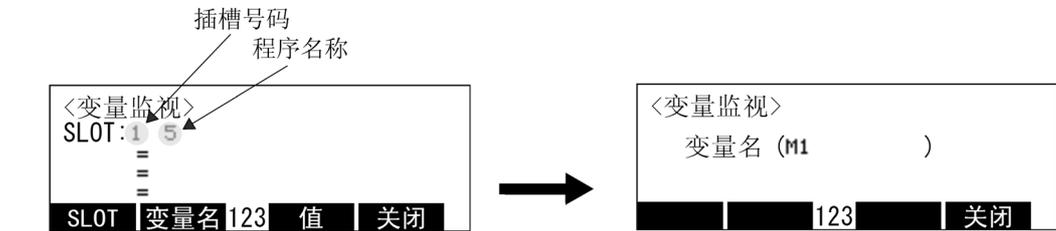
没有使用多重任务功能的情况下，请设定为“1”（按下 [1]、[EXE] 键）



插槽的选择 [F1]

插槽号码的设定 [1]、[EXE]

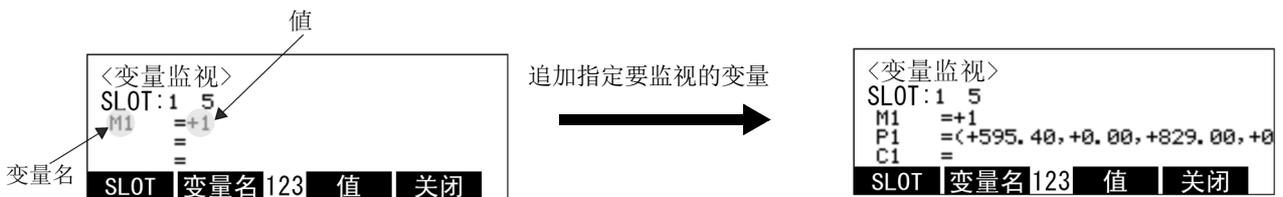
3) 在“SLOT: ”的后面会显示插槽号码和程序名称。
按下对应“名前”的功能键 ([F2])，然后输入要监视的变量。



变量名的选择 [F2]

变量名的输入 [M]、[1]、[EXE]
(监视数值变量 M1 的情况)

4) 在画面显示数值变量 M1 的值。
按下箭头符号，使光标移动到空的行，循环的执行上述 3) 的操作，即可以追加要监视的变量。
可以同时监视的变量为 3 个。



名字的选择 [F2]

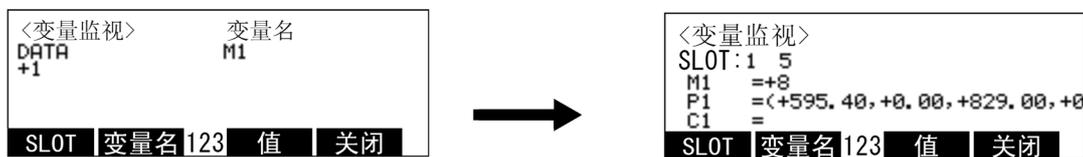
可同时监视 3 个

5) 变量值的变更

可变更显示中的变量的值。

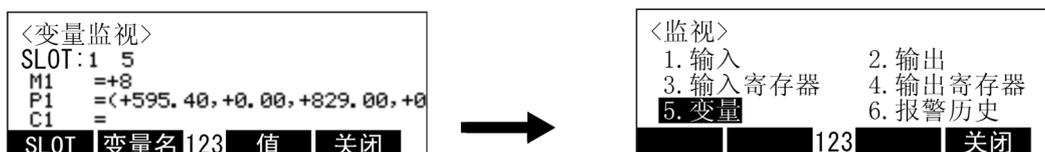
按下箭头符号，将光标移动到要变更的变量名上，按下对应“值”的功能键（[F3]）。

显示现在的值（数据），但是可以输入变更。



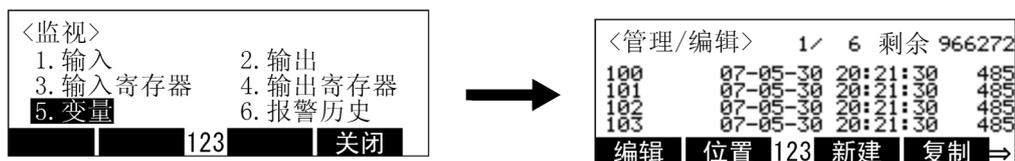
值的设定 [8]、[EXE]
(值设定为 8 的情况)

6) 按下对应“关闭”的功能键（[F4]）的话，会后退到原始的菜单画面。



变量监视的结束 [F4]

7) 在监视菜单画面按下对应“关闭”的功能键（[F4]），则会后退到监视结束前的画面。



信号号码的设定 [F4]

◇◆◇无关操作权◇◆◇

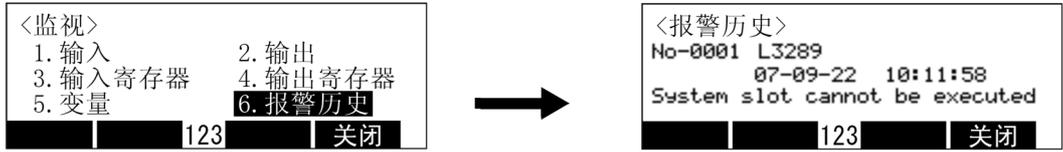
本操作，即使示教单元没有操作权也可以执行。

此外，即使在自动运行中也可以变更变量的值（数据）。

(6) 报警历史

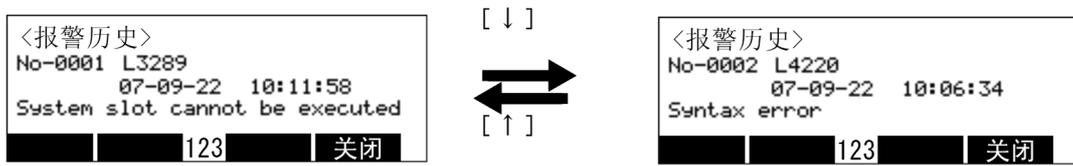
显示在机器人发生报警时的历史的功能。请作为障碍发生时的参考。

1) 在监视菜单画面按下 [6] 键的话，会显示报警历史。



报警历史的选择 [6]

以箭头符号表示报警历史的前后。



显示下一个报警历史 [↓]、显示前一个报警履表 [↑]

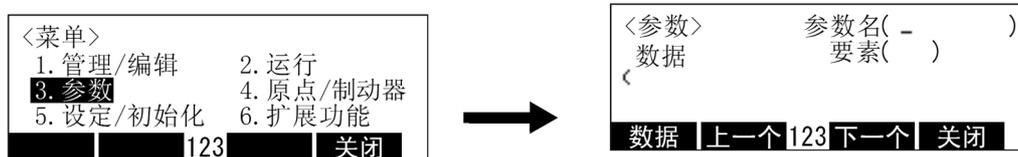
◇◆◇无关操作权◇◆◇

本操作，即使示教单元没有操作权也可以执行。

3.14 参数画面的操作

参数 I/O 的专用输出设定及 TOOL 长度等的设定，被当做参数做设定，机器人动作是以各参数的设定值为基础。显示各参数的设定值及登录的功能。

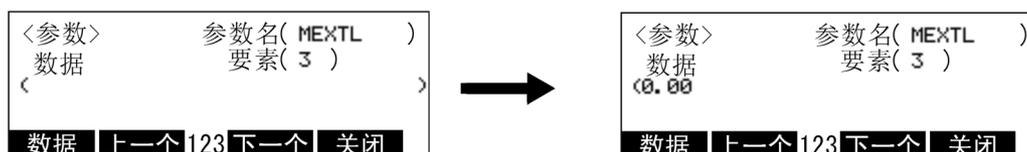
- 1) 在菜单画面按下 [3]，会显示参数画面。



参数的选择 [3]

以下为将参数「MEXTL (TOOL 数据)」的 Z 轴 (第 3 号的要素) 的设定值由 0 变更到 100mm 情况的示范。

- 2) 在名字输入“MEXTL”、要素输入“3”。
会显示现在已设定的数据。

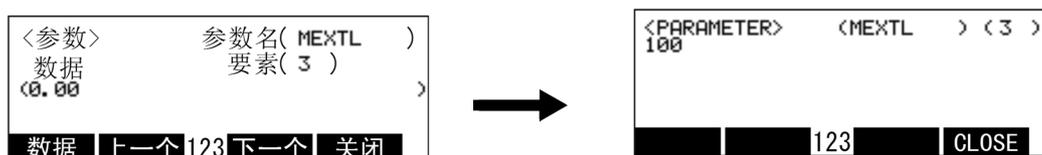


名字的输入 [M]、[E]、[X]、[T]、[L]、[↓]

要素的输入 [3]

确定 [EXE]

- 3) 按下对应“数据”的功能键 ([F1])，输入新的设定值“100”。
不要的数字的请用 [CLEAR] 键删除。



数据的设定 [F1]

数据的输入 [1]、[0]、[0]

确定 [EXE]

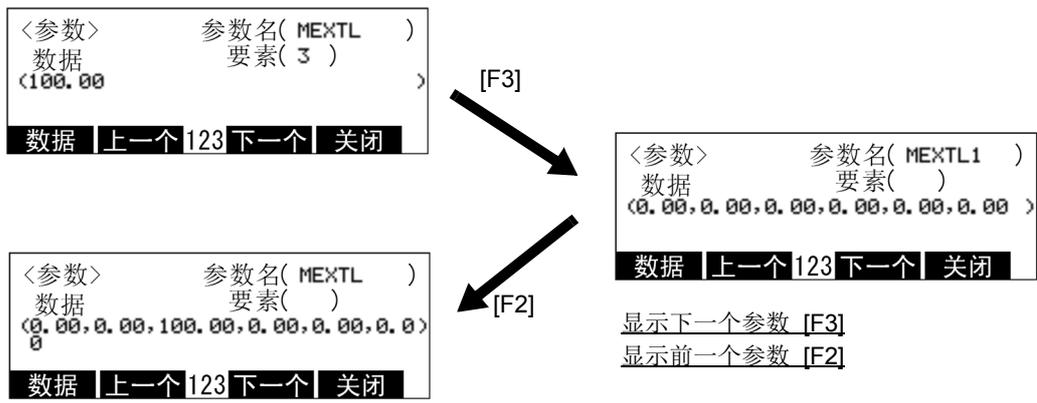
(取消的情况为 [F4])

按下 [EXE] 键，则确定蜂鸣器的鸣响值且后退到参数的显示画面。

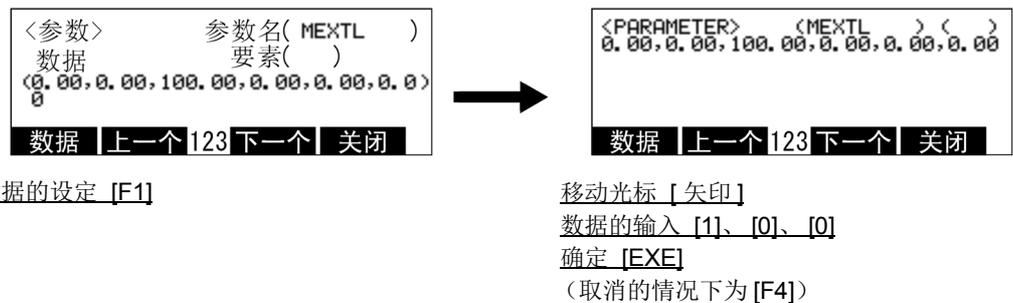
即使输入新的值后，再按下对应“关闭”的功能键 ([F4])，不用变更设定值也可以后退到参数显示画面。

按下对应“其次”的功能键 ([F3]) 的话，将下一个参数按下对应“前”的功能键 ([F2]) 的话，会显示前面的参数。

此情况下，显示所有被名字显示的参数，且在那个参数单位，为了显示前后，将要素号码的指定删除。



即使这个状态下，也可以变更值。
 按下对应“数据”的功能键（[F1]），用箭头符号将光标移动到要变更的要素号码的位置，输入新的设定值。不要的数字请使用 [CLEAR] 键删除。



按下 [EXE] 键的话，蜂鸣器的鸣响值确定，且会后退到参数的显示画面。
 即使输入新的值后，再按下对应“关闭”的功能键（[F4]），不用变更设定值也可以后退到参数显示画面。

- ◇◆◇电源的进入必须要修正◇◆◇
 由于修正进入机器人 CPU 和驱动单元的电源，变更的参数才会变有效。
- ◇◆◇只有程序执行中才能显示◇◆◇
 在程序的执行中，变更参数的设定值的话，会发生报警。
 （即使发生报警，执行的执行也不会停止）
- ◇◆◇在输入的参数里面，显示接近的参数◇◆◇
 参数的名字即使没有输入全部正确的文字，也会自动显示和输入的名字接近的参数。
 根据此点，输名长名字参数里的部份、或按下对应“其次”或“前”的功能键，则可以显示目的参数。

3.15 原点・制动器画面的操作

(1) 原点

因为参数的消失及机器人的干涉等，使原点位置消失或偏移的情况下，必须依据本操作进行原点修正设定。操作方法请参照别册的「操作说明书 / 机器人本体设定到保养」。

(2) 刹车

在刹车关闭状态，解除伺服电机的刹车的功能。

伺服关闭操作请参照第 59 页的「3.8 伺服 ON/OFF」进行。使用在用手直接移动机器人手臂的情况。

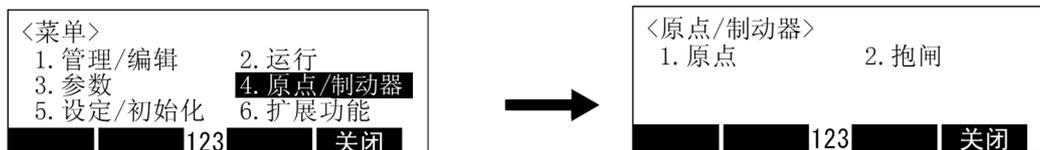


注意

当执行刹车解除的时候，依据机器人的构成及解除轴，手臂会因本身重量而落下。请事先做好在示教单元的操作者或其它的人，做好不让手臂落下的处理、及示教单元的操作者在给彼此信号的基础下，执行本操作。

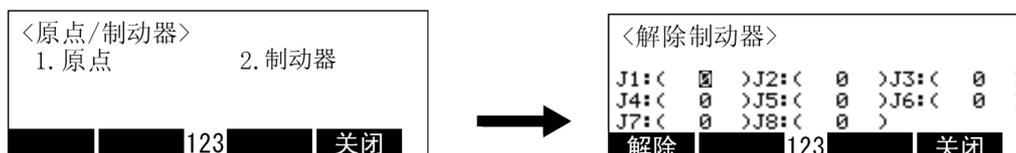
以下为操作方法示范。本操作请在轻按下使能开关（3 位置开关）的状态下进行。

1) 在菜单画面按下 [4] 键的话，会显示原点 / 制动器画面。



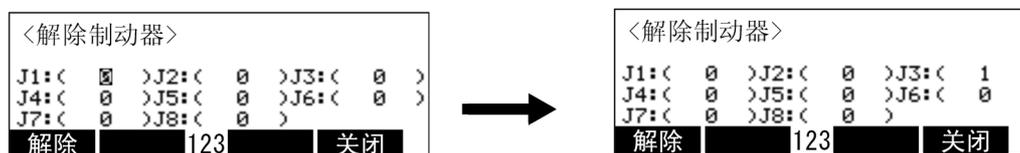
原点 / 制动器画面的选择 [4]

2) 在原点 / 制动器画面按下 [2] 键的话，会显示刹车解除画面。



原点 / 制动器画面的选择 [2]

3) 在要解除刹车的轴输入“1”。



制动器解除轴的设置 [矢印]、[1]

⚠ 注意

可以同时设定复数的轴的刹车解除，但是依据机器人的轴构成，由于合成的本身重量，手臂会落下。
 请事先做好在示教单元的操作者或其它的人，做好不让手臂落下的处理、及示教单元的操作者在给彼此信号的基础下，执行本操作。

4) 在轻押住使能开关的状态下，按下对应“解除”的功能键（[F1]）的话，只有在押住的时候，指定轴的刹车会被解除。

使刹车动作，请放开 [F1] 键或更强力的按下使能开关（3 位置开关）后再放开。

- RH-3FH 系列、RV-2F/4F/7F 系列、RH-3FHR 系列：
制动器解除为连续性的。
- RH-6FH/12FH/20FH 系列：
制动器断续重复解除 / 锁定。
- RV-13F 系列、RV-20F 系列、RV-50F 系列：
J2 轴、J3 轴制动器断续重复解除 / 锁定。其他轴的制动器连续解除。

〈解除制动器〉			
J1:	< 0 >	J2:	< 0 >
J4:	< 0 >	J5:	< 0 >
J7:	< 0 >	J8:	< 0 >
解除	123	关闭	

〈解除制动器〉			
J1:	< 0 >	J2:	< 0 >
J4:	< 0 >	J5:	< 0 >
J7:	< 0 >	J8:	< 0 >
解除	123	关闭	

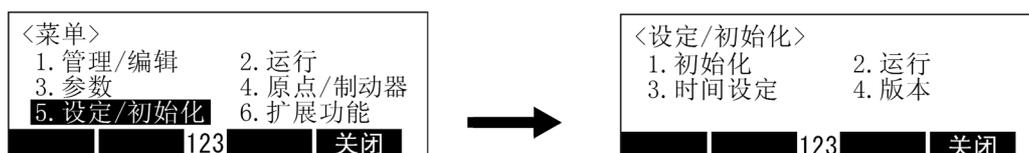
制动器解除 [ENABLE]、[F1]

3.16 设定/初始化画面的操作

下列说明以下功能的操作方法。

- (1) 初始化 1. 程序：删除全部的程序。
 2. 参数：将参数回复到出货时的设定值。
 3. 电池：复位电池的消耗时间。
- (2) 操作 显示电源 ON 的累积时间和电池的剩余时间。
- (3) 时间设定 进行日期和时间的显示、设定。
 在机器人 CPU 内有计时功能，使用在程序登录・变更时间及报警时间的表示等。与现在的日期・时间偏离的时候，请变更为正确的日期、时间。
- (4) 版本 表示机器人 CPU 和示教单元的软件版本。

在菜单画面按下 [5] 键的话，会显示设定 / 初始化菜单画面。

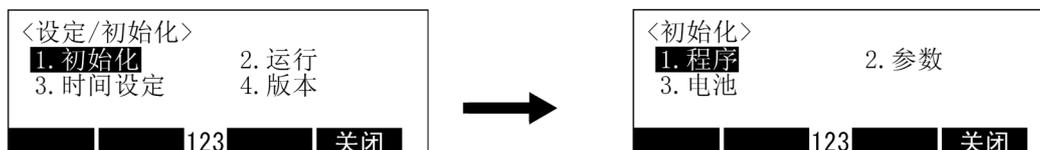


设定 / 初始化菜单画面的选择 [5]

(1) 程序的初始化

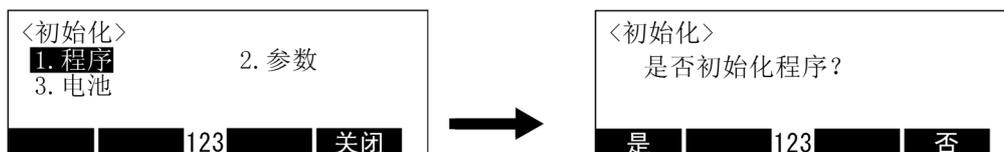
删除全部的程序。

- 1) 在设定 / 初始化菜单画面按下 [1] 键的话，会显示初始化菜单画面。



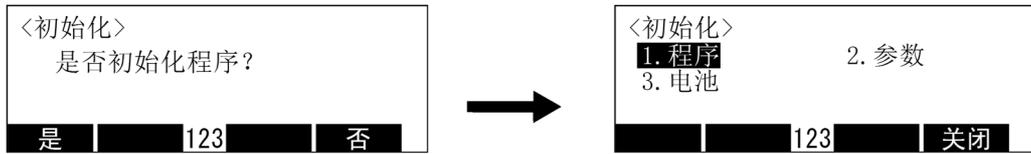
初始化菜单画面的选择 [1]

- 2) 在初始化菜单画面按下 [1] 键，选择“程序”显示确认的画面。



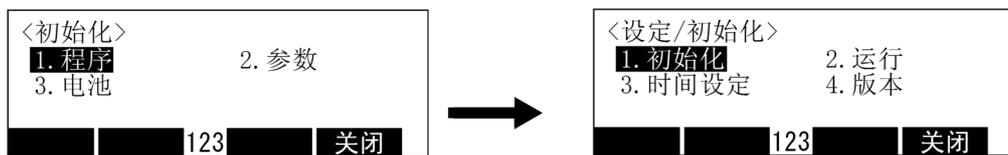
程序初始化确认画面的选择 [1]

- 3) 执行初始化时，按下对应“是”的功能键（[F1]），不执行时按下对应“否”的功能键（[F4]）。画面会返回到初始化 MENU 画面。



初始化的确认
执行情况 [F1]
没有执行情况 [F4]

- 4) 按下对应“关闭”的功能键（[F4]）的话，会后退到设定初始化画面。



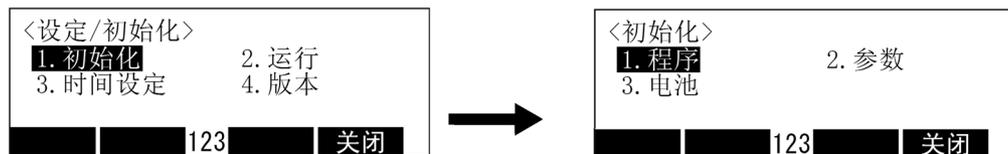
初始化画面的结束 [F4]

◇◆◇被保护住也可以执行◇◆◇
请注意，程序的初始化，在程序保护或变量保护设定为开启的情况下也可以执行。

(2) 参数的初始化

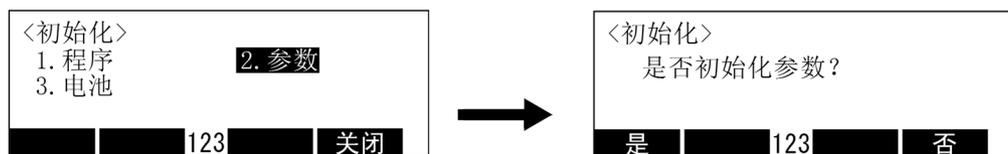
将参数后退到出货时的设定。

- 1) 在设定 / 初始化菜单画面里按下 [1] 键的话，会显示初始化菜单画面。



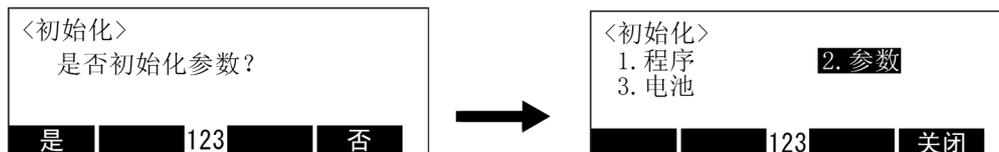
初始化菜单画面的选择 [1]

- 2) 在初始化菜单画面里按下 [2] 键，选择“参数”。
显示确认的画面。



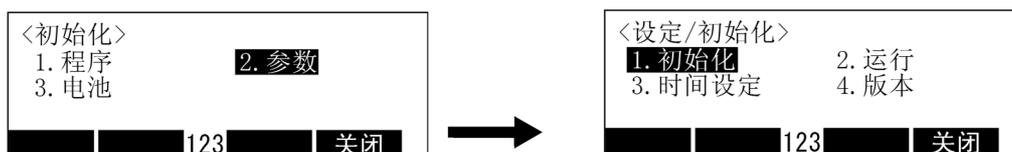
参数选择 [2]

- 3) 热行初始化的情况下，按下对应“是”的功能键（[F1]），不执行的情况下按下对应“否”的功能键（[F4]）。画面会后退到初始化菜单画面。



初始化的确认
执行的情况 [F1]
不执行的情况 [F4]

- 4) 按下对应“关闭”的功能键（[F4]）的话会后退到设定 / 初始化菜单画面。

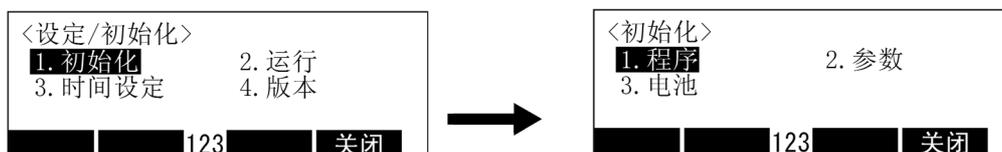


初始化画面的结束 [F4]

(3) 电池的初始化

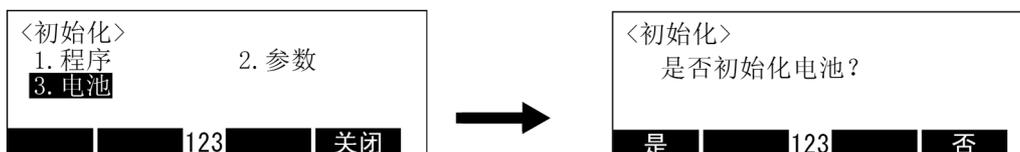
将电池的消耗时间复位。

- 1) 在设定 / 初始化菜单画面按下 [1]，会显示初始化菜单画面。



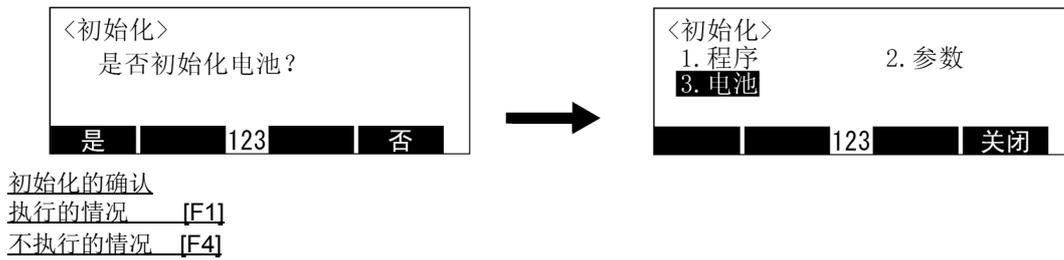
初始化菜单画面的选择 [1]

- 2) 在初始化菜单画面里按下 [3] 键选择“电池”。
显示确认的画面。

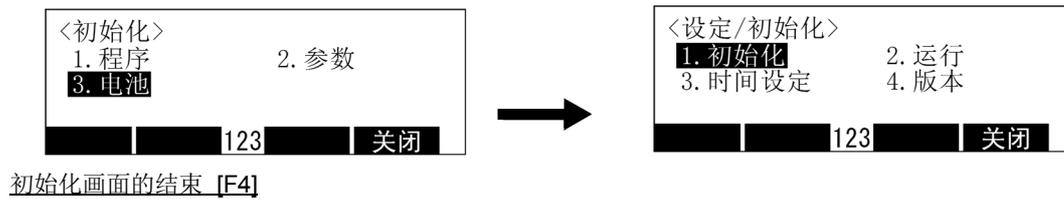


初始化菜单画面的选择 [1]

- 3) 执行初始化时，按下对应“是”的功能键（[F1]），不执行时按下对应“否”的功能键（[F4]）。画面会返回到初始化 MENU 画面。



- 4) 按下对应“关闭”的功能键（[F4]）的话会后退到设定 / 初始化画面。



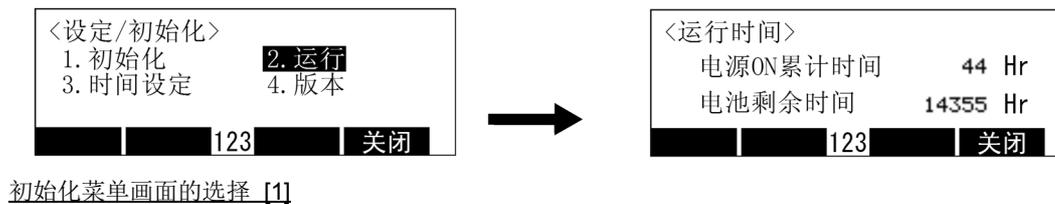
◇◆◇只有在电池更换时，请务必执行◇◆◇

在机器人 CPU 内部，会计算电池的使用时间，在消耗时，会显示出注意讯息。为了使注意讯息正确的显示，在更换电池时，请务必执行本初始化的操作，将电池消耗时间初始化。此外，在没有更换电池时，执行本初始化操作时，注意讯息的显示时间会偏离，因此请只在更换电池后执行。

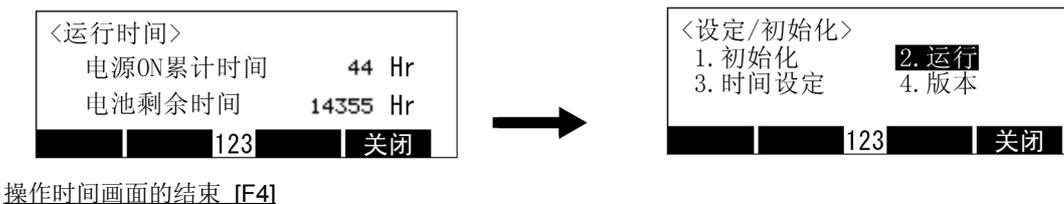
(4) 操作

显示电源 ON 的累积时间和电池的剩余时间。

- 1) 在设定 / 初始化菜单画面按下 [2] 键的话，会显示操作时间画面。



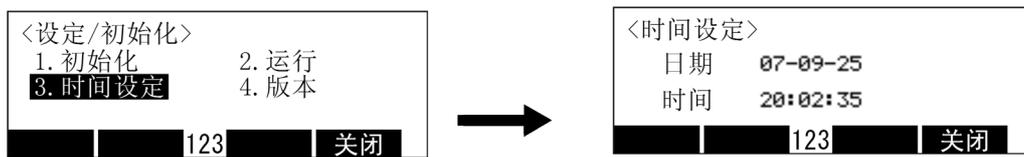
- 2) 按下对应“关闭”的功能键（[F4]）的话会后退到设定 / 初始化画面。



(5) 时间设定

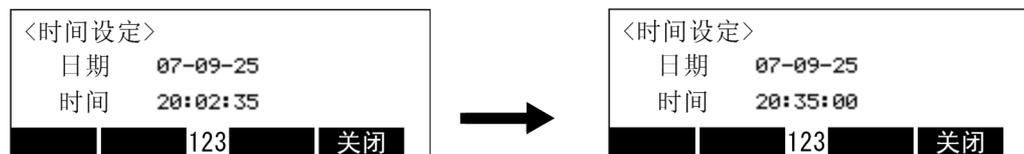
进行日期和时间的显示、设定。

- 1) 在设定 / 初始化菜单画面按下 [3] 键的话，会显示时间设定画面，显示现在的日期及时间。



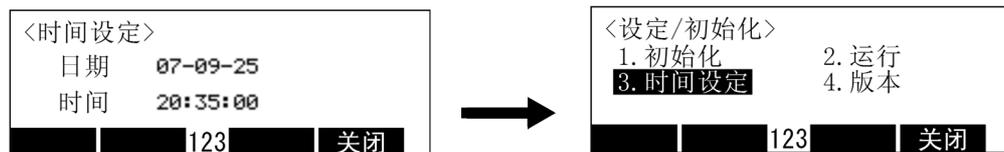
时间设定画面的选择 [3]

- 2) 可以在时间设定画面设定日期和时间。
使用箭头符号将光标移动，请输入现在的日期和时间。



日期、时间的输入 [矢印]、[数值]
日期、时间的确定 [EXE]

- 3) 按下对应“关闭”的功能键 ([F4]) 的话会返回到设定 / 初始化画面。

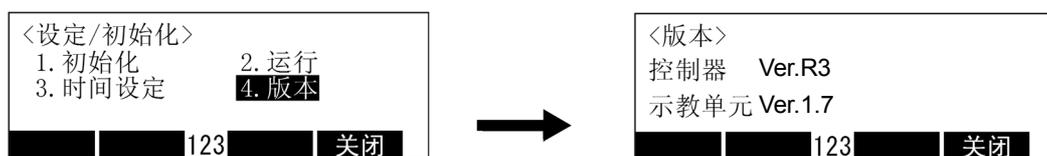


时间设定画面的结束 [F4]

(6) 版本

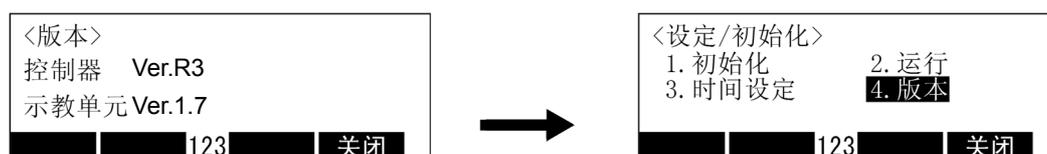
显示机器人 CPU 和示教单元的软件版本。

- 1) 在设定 / 初始化菜单画面按下 [4] 键，显示版本画面。



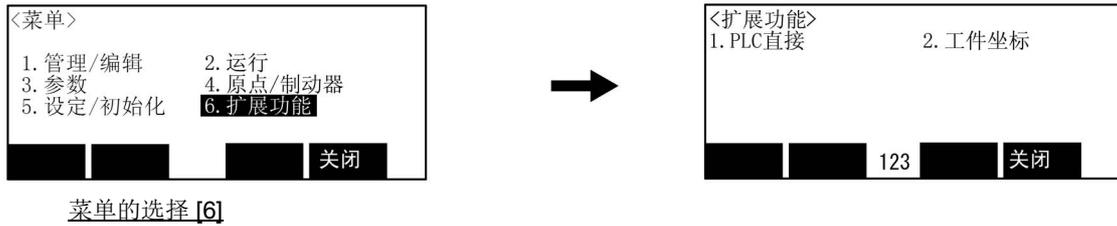
版本画面的选择 [4]

- 2) 按下对应“关闭”的功能键 ([F4]) 的话会返回到设定 / 初始化画面。



版本画面的 [F4]

3.17 扩展功能的操作



(1) PLC 直接

本功能是通过 PLC 的程序直接控制机器人的功能。仅 CR7xx-Q 控制器可以使用此功能。

关于其规格和操作方法的详细内容，请参照另一手册「使用说明书 / CR750-Q 系列、CRnQ-700 系列 iQ Platform 对应扩展功能说明书（BFP-A8757）」。

(2) 工件坐标

是工件 JOG 操作所需工件坐标系的定义画面。使用工件 JOG 时，请在本画面中，对对象工件坐标系进行定义。

详细的操作方法记载在 JOG 操作的说明位置。请参照并加以利用。

【参照位置】

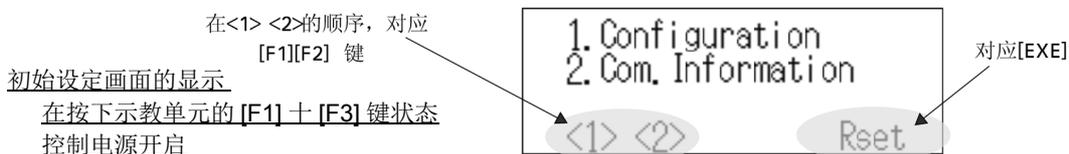
- 1) 工件坐标的设定、工件 JOG 操作：另一手册「使用说明书 / 从机器人本体安装到维护」
- 2) JOG 前进的种类：本书 / 第 22 页的「3.2.1JOG 前进的种类」
- 3) 相关参数：本书 / 第 407 页的「5.1 动作参数」的「工件坐标系」

3.18 初始设定画面的操作

下列示范有初始设定的功能。

- (1) 显示语言的设定 可以在示教单元的画面，设定显示的文字为中文或英文的任一种。
- (2) 对比度调整 示教单元的画面明亮度有 16 段可以调整。

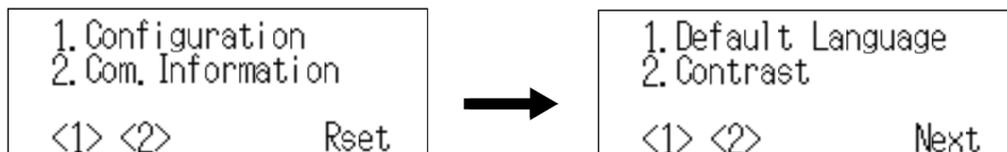
此操作在双重按下示教单元的 [F1] 键和 [F3] 键状态，将控制电源开启显示的初始设定画面里操作。



(1) 显示语言的设定

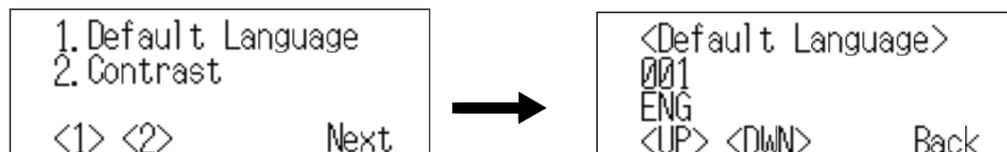
在示教单元的画面，显示文字可以设定为中文或英文中的其中一个。

- 1) 在初始设定画面按下 [F1] 键，选择 “1. Configuration”。



“1. Configuration” 的选择 [F1]

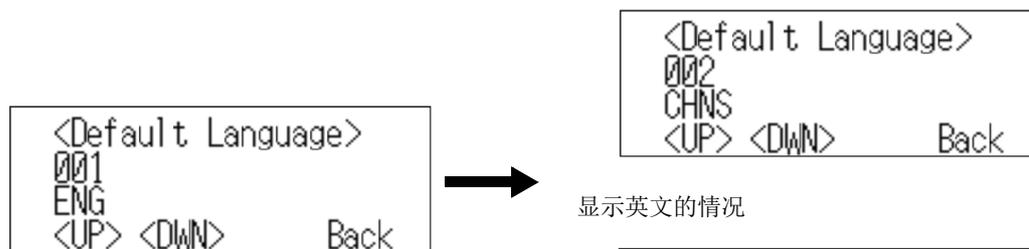
- 2) 按下 [F1] 键选择 “1. Default Language”。



“1. Default Language” 的选择 [F1]

- 3) 按下 [F1] 或 [F2] 键，在画面上要显示中文的情况，画面显示 “CHNS”，在画面上显示英文则显示 “ENG”。

显示中文的情况



显示英文的情况

显示语言的选择 [F1]、[F2]

4) 按下 [EXE] 键则确定。

显示中文的情况

```
<Default Language>
002
CHNS
<UP> <DWN>      Back
```

显示英文的情况

```
<Default Language>
001
ENG
<UP> <DWN>      Back
```



```
1. Default Language
2. Contrast
<1> <2>          Next
```

显示语言的确定 [EXE]

5) 按下 [EXE] 键，显示结束画面。

```
1. Default Language
2. Contrast
<1> <2>          Next
```



```
1. Save and Exit
2. Exit without Save
<1> <2>          Prev
```

设定语言的结束 [EXE]

6) 按下 [F1] 键的话，将设定储存。按下 [F2] 键的话，设定不储存。

在那一个初始设定画面都会后退。

此外，按下 [EXE] 键的话，可以修正设定。

```
1. Save and Exit
2. Exit without Save
<1> <2>          Prev
```



```
1. Configuration
2. Com. Information
<1> <2>          Rset
```

储存设定的情况 [F1]

不储存设定的情况 [F2]

时候设定情况 [EXE]

7) 按下 [EXE] 键的话，设定语言会在示教单元上开始显示。

```
1. Configuration
2. Com. Information
<1> <2>          Rset
```



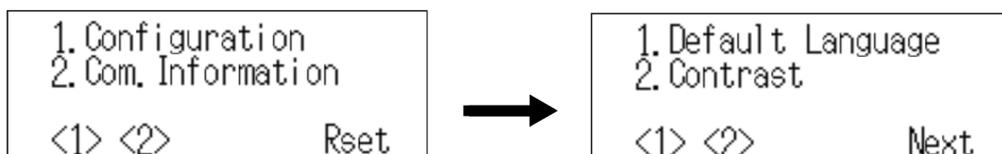
```
MELFA CRnD-7xx      Ver. 51d
RV-6SD
COPYRIGHT(C)2008-2010 MITSUBISHI
ELECTRIC CORPORATION ALL RIGHTS
RESERVED
```

初始设定画面的结束 [EXE]

(2) 明亮度的设定示教单元

的画面明亮度有 16 段可以调整。

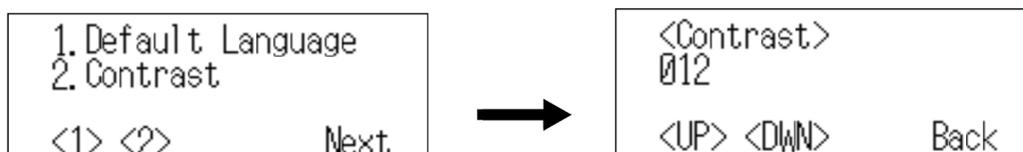
- 1) 在初始设定画面按下 [F1] 键，选择“1.Configuration”。



“1. Configuration” 的选择 [F1]

- 2) 按下 [F2] 键，选择“1.Contrast”。

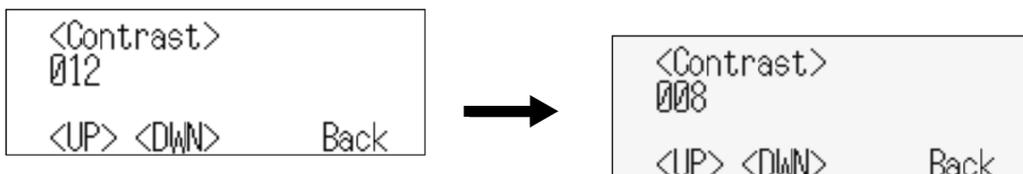
显示现在被设定的明亮度，以从 0 到 15 的数值显示。



“1. Contrast” 的选择 [F2]

- 3) 画面要调亮的情况下请按下 [F1] 键，调暗的情况下按下 [F2] 键，设定适当的明亮度。

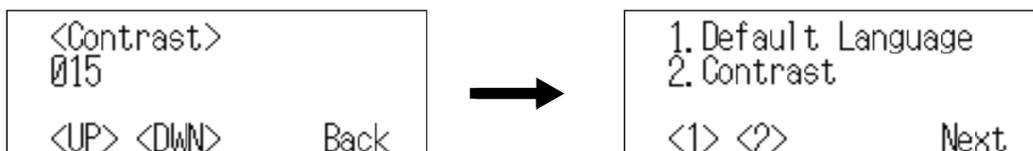
数值越大越明亮。



明亮的情况 [F1]

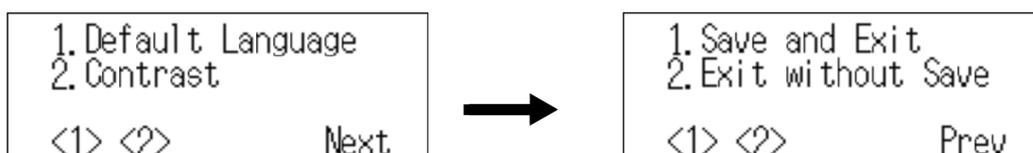
暗的情况 [F2]

- 4) 按下 [EXE] 键确定。



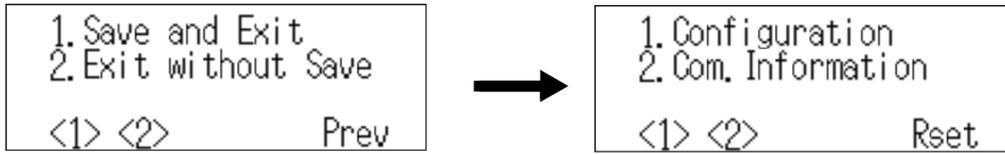
明亮度的确定 [EXE]

- 5) 按下 [EXE] 键，显示结束画面。



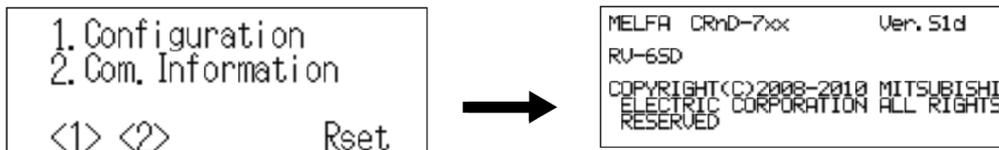
明亮度设定结束 [EXE]

- 6) 按下 [F1] 键则会储存设定。
按下 [F2] 键则不会储存设定。任何一个都会后退到初始设定画面。
另外，按下 [EXE] 键则可以修正设定。



储存设定的情况 [F1]
不储存设定的情况 [F2]
时候设定的情况 [EXE]

- 7) 按下 [EXE] 键的话，设定语言会在示教单元上开始显示。



初始设定画面的结束 [EXE]

4 MELFA-BASIC V

在此说明关于程序语言「MELFA-BASIC V」的功能及详细语言规格。

4.1 MELFA-BASIC V的功能

在此说明关于程序语言「MELFA-BASIC V」的概要。说明机器人的基本动作、信号的输出输入、条件分支方法等。

表 4-1: 说明项目一览

No.	项目	内容	相关指令等
1	4.1.1 机器人的动作控制	(1) 关节插补动作	Mov
2		(2) 直线插补动作	Mvs
3		(3) 圆弧插补动作	Mvr, Mvr2, Mvr3, Mvc
4		(4) 连续动作	Cnt
5		(5) 加减速时间和速度控制	Accel、Oadl
6		(6) 往目的位置的到达确认	Fine、Mov 和 Dly
7		(7) 高轨迹精度控制	Prec
8		(8) 抓手·TOOL 控制	HOpen、HClose、Tool
9	4.1.2Pallet 运算	-----	Def Plt、Plt
10	4.1.3 程控	(1) 无条件分支·条件分支·待机	GoTo、If Then Else、Wait 另外
11		(2) 循环	For Next、While Wend
12		(3) 插入	Def Act、Act
13		(4) 子程序	GoSub、Callp、On GoSub 另外
14		(5) 定时器	Dly
15		(6) 停止	End (1 循环停止)、Hlt
16	4.1.4 外部信号的输出输入	(1) 信号输入	M_In、M_Inb、M_Inw 另外
17		(2) 信号输出	M_Out、M_Outb、M_Outw 另外
18	4.1.5 通信	-----	Open、Close、Print、Input 另外
19	4.1.6 运算	(1) 运算符一览	+、-、*、/、<>、<、> 等
20		(2) 位置数据的相对运算 (乘算)	P1*P2
21		(3) 位置数据的相对运算 (加算)	P1+P2
22	4.1.7 附随句	-----	Wth、Wthlf

和指令的详细说明请参照第 163 页的「4.13 指令的详细说明」。

4.1.1 机器人的动作控制

(1) 关节插补动作

在往指定的位置、各关节轴单位的插补移动。(因为在各关节轴单位做插补, 尖端的轨迹不会变直线。)

■ 指令语

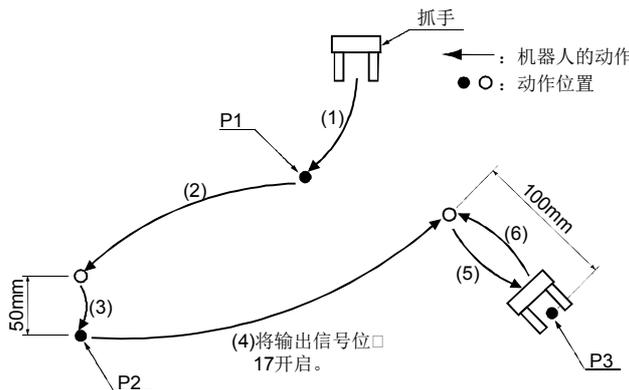
指令语	说明
Mov	用关节插补往指定位置移动。在 Type 无法指定插补形式。而且, 可以指定 Wth、Wthlf 的附随句。

■ 例子

例子	说明
Mov P1	' 往 P1 移动。
Mov P1+P2.....	' 移动到将 P1 和 P2 的各坐标成份加算起来的位置。参照第 123 页。
Mov P1*P2.....	' 往从 P1 到 P2 已变换的相对位置移动。参照第 123 页。
Mov P1, -50 (注)	' 移动到从 P1 开始在抓手方向后退 50mm 位置。
Mov P1 Wth M_Out (17) =1	' 往 P1 的移动开始, 同时将输出信号位 17 开启。
Mov P1 Wthlf M_In (20) =1, Skip.....	' 在往 P1 移动中, 开启输入信号 20, 往 P1 的移动插入, 前进到下一个 STEP。
Mov P1 Type 1, 0 (初期值: 绕道)	' 各轴动作在超过 180° 情况下, 做绕道 (或超近路) 的指定。

■ 程序例

• 机器人的动作



注意 (注) 的前进 / 后退的指定例子中, 程序示范例为垂直 6 轴机器人 (RV-12S 等) 的情况。抓手的前进 / 后退的方向, 根据在每个机型中 TOOL 坐标的 Z 轴方向 (+/- 方向) 决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标, 指定正确的方向。

• 程序的例子

程序	说明
1 Mov P1	' (1) 往P1移动。
2 Mov P2, -50 (注)	' (2) 往从P2开始, 在方向后退50mm的位置移动。
3 Mov P2	' (3) 往P2移动。
4 Mov P3, -100 Wth M_Out (17) =1	' (4) 开始往从P3开始, 在抓手后退100mm的位置移动, 同时将输出信号17开启。
5 Mov P3	' (5) 往P3移动。
6 Mov P3, -100 (注)	' (6) 返回到从P3到在抓手方向后退100mm位置。
7 End	' 程序结束。

■ 相关功能

功能	说明个所
指定动作速度.....	第 104 页的「(5) 加减速时间和速度控制」
指定加减速时间.....	第 104 页的「(5) 加减速时间和速度控制」
确认往目的位置的到达.....	第 106 页的「(6) 往目的位置的到达确认」
不在目标位置止而以连续动作往下一个目的位置.....	第 103 页的「(4) 连续动作」
在直线动作.....	第 100 页的「(2) 直线插补动作」
以画圆和圆弧动作.....	第 101 页的「(3) 圆弧插补动作」
在移动指令附加处理.....	第 279 页的「Wth (With)」

(2) 直线插补动作

将抓手尖端以直线插补移动到已指定位置。

■ 指令

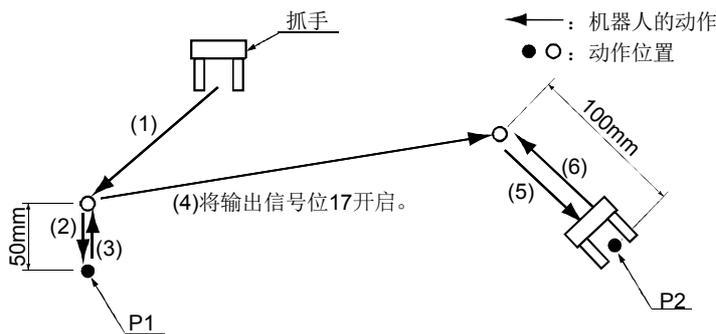
指令	说明
Mvs	以直线插补往指定位置移动。可以在 Type 指定插补形式。可以指定 Wth、Wth If 的附随句。

■ 例子

例子	说明
Mvs P1	' 往 P1 移动。
Mvs P1+P2	' 移动到将 P1 和 P2 的各坐标成份加算起来的位置。参照第 123 页。
Mvs P1*P2	' 往从 P1 到 P2 已变换的相对位置移动。参照第 123 页。
Mvs P1, -50 (注)	' 移动到从 P1 开始在抓手方向后退 50mm 位置。
Mvs P1 Wth M_Out (17) =1	' 往 P1 的移动开始, 同时将输出信号位 17 开启。
Mvs P1 Wthif M_In (20) =1, Skip	' 在往 P1 移动中, 开启输入信号 20, 往 P1 的移动插入, 前进到下一个 STEP。
Mvs P1 Type 0, 0	' 以等价旋转移移动到 P1。
Mvs P1 Type 0, 1	' 以 3 轴直交插补移动到 P1。

■ 程序例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定例子中, 程序示范例为垂直 6 轴机器人 (RV-12S 等) 的情况。抓手的前进 / 后退的方向, 根据在每个机型中 TOOL 坐标的 Z 轴方向 (+/- 方向) 决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标, 指定正确的方向。

• 程序例

程序	说明
1 Mvs P1, -50 (注)	' (1) 以直线插补从 P1 移动到抓手方向后退 50mm 位置。
2 Mvs P1	' (2) 以直线插补往 P1 移动。
3 Mvs, -50 (注)	' (3) 以直线插补从现在位置 (P1) 移动到抓手方向后退 50mm 位置。
4 Mvs P2, -100 Wth M_Out (17) =1 (注)	' (4) 开始移动的同时, 开启输出信号位 17。
5 Mvs P2	' (5) 以直线插补往 P2 移动。
6 Mvs, -100 (注)	' (6) 以直线插补移动到从 P2 到往抓手方向后退 100mm 位置。
7 End	' 程序结束。

■ 相关功能

功能	说明
指定动作速度。.....	第 104 页的「(5) 加减速时间和速度控制」
指定加减速时间。.....	第 104 页的「(5) 加减速时间和速度控制」
确认到目的位置的到达。.....	第 106 页的「(6) 往目的位置的到达确认」
不在目标位置止而以连续动作往下一个目的位置。.....	第 103 页的「(4) 连续动作」
以关节插补动作。.....	第 99 页的「(1) 关节插补动作」
以画圆和圆弧动作。.....	第 101 页的「(3) 圆弧插补动作」
在移动指令附加处理。.....	第 279 页的「Wth (With)」

(3) 圆弧插补动作

以三维圆弧插补，在 3 点指定的圆弧上移动。

现在位置从圆弧开始起点偏离的情况下，从直线动作到起点为止，做圆弧动作。

■ 指令

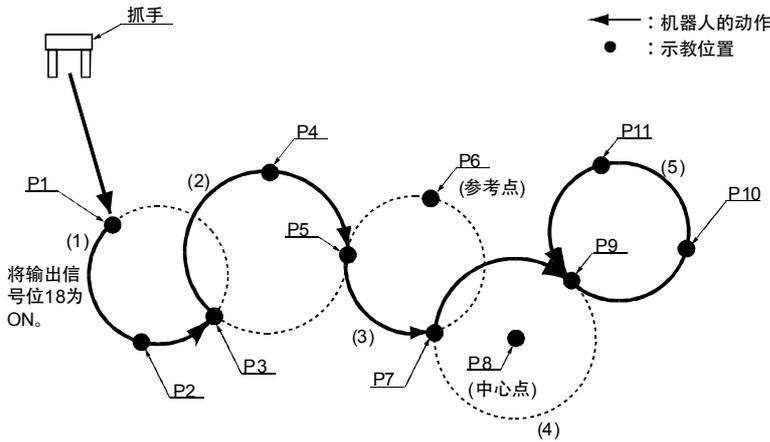
指令	说明
Mvr	指定起点、通过点、终点后，以圆弧插补依照起点→通过点→终点的顺序移动。 可在 Type 指定插补形式。可指定 Wth、Wth If 的附随句。
Mvr2	指定起点、终点、参考点后，以圆弧插补从起点→终点，不通过参考点的方式移动。 可在 Type 指定插补形式。可指定 Wth、Wth If 的附随句。
Mvr3	指定起点、终点、中心点后，以圆弧插补从起点→终点移动。从起点到终点的扇角为 0 度 < 扇角 < 180 度。 可在 Type 指定插补形式。可指定 Wth、Wth If 的附随句。
Mvc	指定起点（终点）、通过点 1、通过点 2 后，以圆弧插补在起点→通过点 1 →通过点 2 →终点的顺序做圆周移动。可指定 Wth、Wth If 的附随句。

■ 例子

例子	说明
Mvr P1, P2, P3.....	’ 以圆弧插补移动 P1 → P2 → P3。
Mvr P1, P2, P3 Wth M_Out (17) =1..	’ 在 P1 → P2 → P3 的圆弧插补动作开始时也开启输出信号位 17。
Mvr P1, P2, P3 Wthlf M_In (20) =1, Skip	’ 假设在 P1 → P2 → P3 的圆弧插补动作中，开启输入信号位 20，则会中止往 P1 的圆弧插补动作，且往下一个 STEP 前进。
Mvr P1, P2, P3 Type 0, 1	’ P1 → P2 → P3 以圆弧插补移动。
Mvr2 P1, P3, P11.....	’ P1 → P3 以不通过 P11 的方向，用圆弧插补移动。P11 为参考点。
Mvr3 P1, P3, P10.....	’ 在 P1 → P3 扇角的较小方向以圆弧插补移动。P10 为中心点。
Mvc P1, P2, P3.....	’ 将 P1 → P2 → P3 → P1 以圆周移动。

■ 程序例

• 机器人的动作



• 程序例

程序	说明
1 Mvr P1, P2, P3 Wth M_Out (18) =1	’ (1) 将P1→P2→P3以圆弧动作。因为动作前的机器人现在位置偏离起点，最初以直线动作移动到起点。圆弧动作开始及同时的开启 (P1) 输出信号位18。
2 Mvr P3, P4, P5	’ (2) 将P3→P4→P5以圆弧动作。
3 Mvr2 P5, P7, P6	’ (3) 起点 (P5)，参考点 (P6)，终点 (P7) 在指定的圆周上，起点开始以圆弧到终点为止不通过参考点的动作。
4 Mvr3 P7, P9, P8	’ 在 (4) 中心点 (P8)，起点 (P7)，终点 (P9) 指定的圆周上，以圆弧动作从起点到终点。
5 Mvc P9, P10, P11	’ (5) 以圆弧在P9→P10→P11→P9动作。(为1周动作。)
6 End	’ 程序结束。

■ 相关功能

功能	说明
指定动作速度。.....	第 104 页的 「(5) 加减速时间和速度控制」
指定加减速时间。.....	第 104 页的 「(5) 加减速时间和速度控制」
确认到目的位置的到达。.....	第 106 页的 「(6) 往目的位置的到达确认」
不在目标位置止而以连续动作往下一个目的位置。.....	第 103 页的 「(4) 连续动作」
以关节插补动作。.....	第 99 页的 「(1) 关节插补动作」
以直线动作。.....	第 100 页的 「(2) 直线插补动作」
在移动指令附加处理。.....	第 279 页的 「With (With)」

(4) 连续动作

每个动作位置不停止，连续移动复数的动作位置。

在指令里，指定连续动作的开始和结束。即使在连续动作中，也可以变更速度。

■ 指令

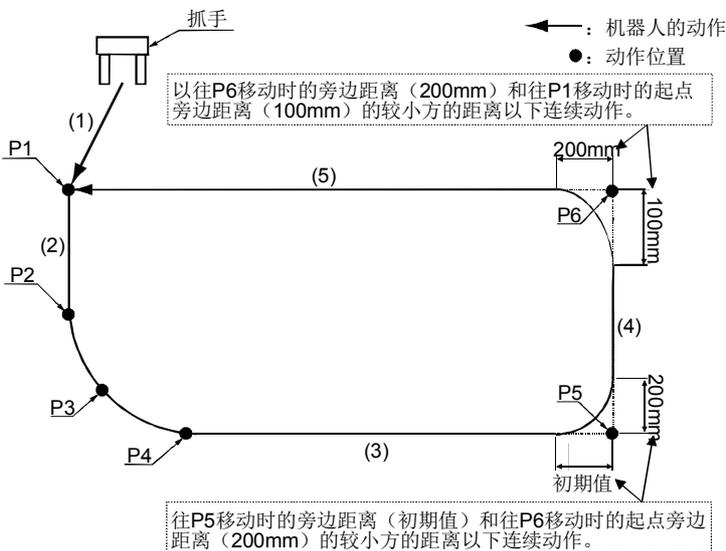
指令	说明
Cnt	指定连续动作的开始和结束。

■ 例子

例子	说明
Cnt 1	' 指定连续动作的开始。起点旁边距离为减速开始位置、终点旁边开始位置为加速结束位置（视机型会有所差异）。
Cnt 1, 100, 200	' 连续动作的开始指定和指定它的起点旁边距离=100mm、终点旁边距离=200mm。
Cnt 0	' 指定连续动作的结束 1。

■ 程序例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定例子中，程序示范例为垂直 6 轴机器人 (RV-12S 等) 的情况。抓手的前进 / 后退的方向，根据在每个机型中 TOOL 坐标的 Z 轴方向 (+/- 方向) 决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标，指定正确的方向。

• 程序例

程序	说明
1 Mov P1	' (1) 以关节插补往P1移动。
2 Cnt 1	' 使连续动作有效。(此后的移动会变成连续动作)
3 Mvr P2, P3, P4	' (2) 直线动作到P2为止且连续做圆弧动作到P4为止。
4 Mvs P5	' 连续圆弧动作，往P5直线动作。
5 Cnt 1, 200, 100	' (3) 在连续动作的起点旁边距离设定为200mm，终点旁边距离设定为100mm。
6 Mvs P6	' (4) 在前面往P5的移动连续，以直线动作往P6。
7 Mvs P1	' (5) 连续，以直线动作往P1。
8 Cnt 0	' 使连续动作无效。
9 End	' 程序结束。

■ 相关功能

功能	说明
指定动作速度.....	第 104 页的「(5) 加减速时间和速度控制」
指定加减速时间.....	第 104 页的「(5) 加减速时间和速度控制」
确认往目的位置的到达.....	第 106 页的「(6) 往目的位置的到达确认」
在关节插补动作.....	第 99 页的「(1) 关节插补动作」
在直线动作.....	第 100 页的「(2) 直线插补动作」
以画圆和圆弧动作.....	第 101 页的「(3) 圆弧插补动作」

(5) 加减速时间和速度控制

可以指定对加减速时的最高加减速度的比例及动作速度。

■ 指令

指令	说明
Accel	将移动速度时的加、减速度，以对最高速度的比例（%）指定。
Ovrd	将在程序全体的动作速度，以对最高速度的比例（%）指定。
JOvrd	将关节插补动作时的速度，以对最高速度的比例（%）指定。
Spd	将直线、圆弧插补动作时的速度、以抓手尖端速度（mm/s）指定。
Oadl	指定最佳加减速功能为有效 / 无效。

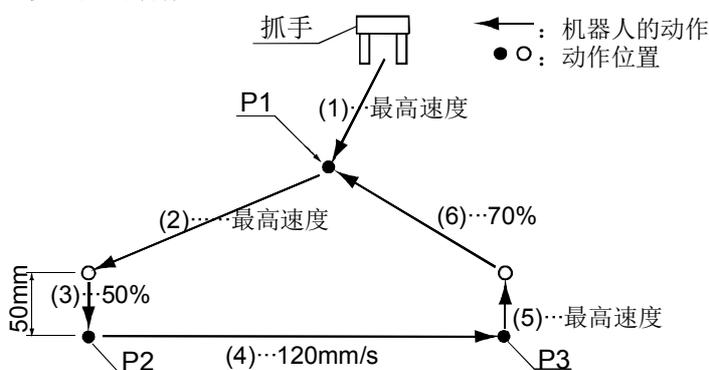
■ 例子

例子	说明
Accel	’ 加减速全部以 100% 设定。
Accel 60, 80.....	’ 加速度以 60%、减速度为 80% 设定。 (最高加减速时间为 0.2 秒的情况加速时间 $0.2 \div 0.6 = 0.33$ 、 减速度时间 $0.2 \div 0.8 = 0.25$ 秒。)
Ovrd 50	’ 关节插补、直线插补、圆弧插补动作都以最高速度的 50% 设定。
JOvrd 70.....	’ 将关节插补动作设定为最高速度的 70%。
Spd 30	’ 直线插补、圆弧插补动作时的速度设定为 30mm/s。
Oadl ON	’ 使最佳加减速功能为有效。

- 关节插补时的动作速度 控制器（T/B）的设定值 × Ovrd 指令的设定值 × JOvrd 指令的设定值
- 直线、圆弧插补时的动作速度 ... 控制器（T/B）的设定值 × Ovrd 指令的设定值 × Spd 指令的设定值

■ 程序例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定例子中，程序示范例为垂直 6 轴机器人（RV-12S）的情况。抓手的前进 / 后退的方向，根据在每个机型中 TOOL 坐标的 Z 轴方向（+/- 方向）决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标，指定正确的方向。

• 程序例

程序	说明
1 Ovrd 100	’ 将全体相关的动作速度设定为最大。
2 Mvs P1	’ (1) 以最高速度往 P1 移动。
3 Mvs P2, -50 (注)	’ (2) 以最高速度移动到从 P2 开始往抓手方向后退 50mm 的位置。
4 Ovrd 50	’ 将全体相关的动作速度设定为最高速度的一半。
5 Mvs P2	’ (3) 以初始设定速度的一半，用直线动作到 P2。
6 Spd 120	’ 将尖端速度设定为 120mm/s。（因为速度比例为 50%，但实际以 60mm/s 动作。）
7 Ovrd 100	’ 为了使实际的尖端速度为 120mm/s，请将动作速度的比例设为 100%。
8 Accel 70, 70	’ 加减速速度也设定为最高加减速度的 70%。
9 Mvs P3	’ (4) 以尖端速度 120mm/s 直线动作到 P3。
10 Spd M_NSpd	’ 将尖端速度后退到初始值。

程序	说明
11 JOvrd 70	' 将关节插补动作时的速度设定为70%。
12 Accel	' 加减速度后退到最高加减速度。
13 Mvs , -50 (注)	' (5) 以直线动作时的初始设定速度, 直线移动到从现在位置 (P3) 到在抓手方向后退50mm的位置。
14 Mvs P1	' (6) 以最高速度的70%往P1移动。
15 End	' 程序结束。

■ 相关功能

功能	说明
在关节插补动作。.....	第 99 页的 「(1) 关节插补动作」
在直线动作。.....	第 100 页的 「(2) 直线插补动作」
以画圆和圆弧动作。.....	第 101 页的 「(3) 圆弧插补动作」
不在目标位置止而以连续动作往下一个目的位置。.....	第 103 页的 「(4) 连续动作」

(6) 往目的位置的到达确认

以脉冲数指定决定位置完成条件。(Fine 指令)连续动作时,本指定为无效。

■ 指令

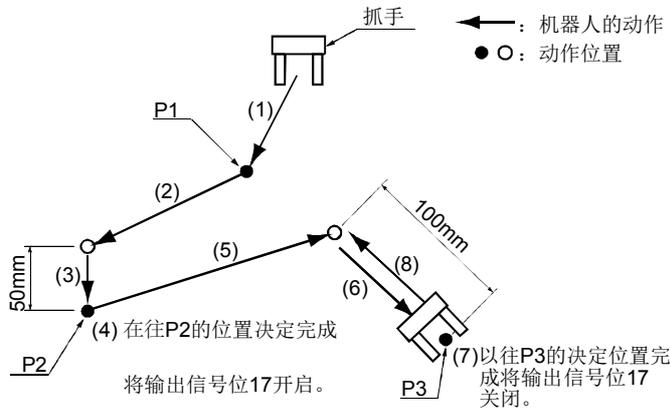
指令	说明
Fine	以脉冲数指定定位完成条件。指定脉冲数越小的话,越可以正确指定位置。
Mov 和 Dly	在 Mov 的动作指令,以 Dly 指令(定时器)做定位完成。 (在以皮带驱动方式的机器人为有效。例) RP-1AH/3AH/5AH 等)

■ 例子

例子	说明
Fine 100.....	将定位完成条件设定为 100 脉冲。
Mov P1.....	以关节插补往 P1 移动。(以指令值标准完成。)
Dly 0.5.....	动作指令后的定位以定时器来执行。 (在以皮带驱动方式的机器人为有效。例) RP-1AH/3AH/5AH 等)

■ 程序例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定例子中,程序示范例为垂直 6 轴机器人 (RV-12S 等) 的情况。抓手的前进 / 后退的方向,根据在每个机型中 TOOL 坐标的 Z 轴方向 (+/- 方向) 决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标,指定正确的方向。

• 程序例

程序	说明
1 Cnt 0	’ Fine指令只在Cnt指令关闭中有效。
2 Mvs P1	’ (1) 以关节插补往P1移动。
3 Mvs P2, -50 (注)	’ (2) 以最高速度移动到从P2开始往抓手方向后退50mm的位置。
4 Fine 50	’ 将定位完成脉冲设定为50。
5 Mvs P2	’ (3) 以直线插补往P2移动。(定位完成脉冲在50以下, Mvs结束)
6 M_Out (17) =1	’ (4) 定位脉冲为50脉冲时,开启输出信号17。
7 Fine 1000	’ 将定位完成脉冲设定为1000。
8 Mvs P3, -100 (注)	’ (5) 以直线移动到从P3开始往抓手方向后退100mm的位置。
9 Mvs P3	’ (6) 以直线移动到P3。
10 Dly 0.1	’ 定位以定时器执行。
11 M_Out (17) =0	’ (7) 将输出信号关闭。
12 Mvs , -100 (注)	’ (8) 以直线移动到从现在位置 (P3) 开始往抓手方向后退100mm的位置。
13 End	’ 程序结束。

■ 关连功能

功能	说明
以关节插补动作。.....	第 99 页的「(1) 关节插补动作」
以直线动作。.....	第 100 页的「(2) 直线插补动作」
不在目标位置停止而以连续动作往下一个目的位置。.....	第 103 页的「(4) 连续动作」

(7) 高轨迹精度控制
 能将机器人的动作轨迹提升的动作。

■ 指令

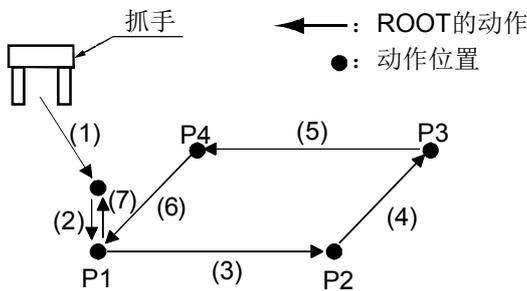
指令	说明
Prec	指定高精度模式的有效、无效。

■ 例子

例子	说明
Prec On.....	' 使高精度模式为有效。
Prec Off.....	' 使高精度模式为无效。

■ 程序例

• 机器人的动作



注意 (注) 抓手的前进/后退的指定例子中, 程序示范例为垂直 6 轴机器人 (RV-12S 等) 的情况。抓手的前进/后退的方向, 根据在每个机型中 TOOL 坐标的 Z 轴方向 (+/- 方向) 决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标, 指定正确的方向。

• 程序例

程序	说明
1 Mov P1, -50 (注)	' (1) 以关节插补从P1移动在抓手方向后退50mm位置。
2 Ovrd 50	' 将动作速度设定为最高速度的一半。
3 Mvs P1	' (2) 直线往P1移动。
4 Prec On	' 使高轨迹模式为有效。
5 Mvs P2	' (3) 从P1到P2以高轨迹精度移动。
6 Mvs P3	' (4) 从P2到P3以高轨迹精度移动。
7 Mvs P4	' (5) 从P3到P4以高轨迹精度移动。
8 Mvs P1	' (6) 从P4到P1以高轨迹精度移动。
9 Prec Off	' 使高轨迹模式为无效。
10 Mvs P1, -50	' (7) 以直线插补后退到从P1移动在抓手方向后退50mm位置。
11 End	' 程序结束。

注意

Prec 指令可以提升机器人的尖端轨迹精度, 但因为降低动作的加减速度, 因此可能延长定时器时间。
 不加入 Cnt 指令精度会提升。但是, 无法保证尖端的速度。

(8) 抓手·TOOL 控制

指定抓手的开闭及 TOOL 的形状。

■ 指令

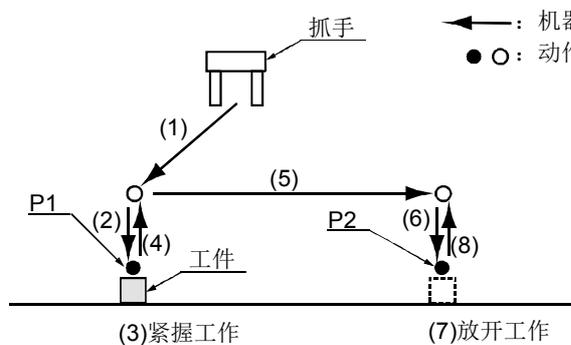
指令	说明
HOpen	打开指定抓手。
HClose	关闭指定抓手。
Tool	设定使用 TOOL 的形状，并符合控制点。

■ 例子

例子	说明
HOpen 1.....	' 打开 1 号的抓手。
HOpen 2.....	' 打开 2 号的抓手。
HClose 1.....	' 关闭 1 号的抓手。
HClose 2.....	' 关闭 2 号的抓手。
Tool (0, 0, 95, 0, 0, 0)	' 将机器人的控制点的位置设定在从法兰面的延长方向 95mm 的地方。

■ 程序例

• 机器人的动作



注意 (注) 抓手的前进/后退的指定例子中，程序示范例为垂直 6 轴机器人 (RV-12S 等) 的情况。抓手的前进/后退的方向，根据在每个机型中 TOOL 坐标的 Z 轴方向 (+/- 方向) 决定。请参照别册「机器人本体设定到保养」的「动作的确认」中所显示的 TOOL 坐标，指定正确的方向。

• 程序例

程序	说明
1 Tool (0, 0, 95, 0, 0, 0)	' 抓手长设定为95mm。
2 Mvs P1, -50 (注)	' (1) 以关节插补从P1移动在抓手方向后退50mm位置。
3 Ovr 50	' 将动作速度设定为最高速度的一半。
4 Mvs P1	' (2) 直线往P1移动。(去抓取工件)
5 Dly 0.5	' 为目的位置到达完成，等待0.5秒。
6 HClose 1	' (3) 关闭抓手1。(抓住工件)
7 Dly 0.5	' 等待0.5秒。
8 Ovr 100	' 将动作速度设定为最大。
9 Mvs, -50 (注)	' (4) 以直线动作从现在位置 (P1) 移动在抓手方向后退50mm位置。(抓住工件向上)
10 Mvs P2, -50 (注)	' (5) 以关节插补动作从P2移动在抓手方向后退50mm位置。
11 Ovr 50	' 将动作速度设定为最高速度的一半。
12 Mvs P2	' (6) 直线往P2移动。(要放置工件)
13 Dly 0.5	' 为目的位置到达完成，等待0.5秒。
14 HOpen 1	' (7) 打开抓手1。(放开工件)
15 Dly 0.5	' 等待0.5秒。
16 Ovr 100	' 将动作速度设定为最大。
17 Mvs, -50 (注)	' (8) 以直线动作从现在位置 (P2) 移动在抓手方向后退50mm位置。(放开工件)
18 End	' 程序结束。

■ 关连功能

功能	说明
附随句.....	第 279 页的「Wth (With)」

4.1.2 Pallet运算

将工件规则正确的排列 (Palletiz) 作业及取出规则正确排列的工件 (Depalletize) 作业的情况下, 码垛功能会变成使用基准, 只示教工件的位置, 可用运算求得剩余的位置。

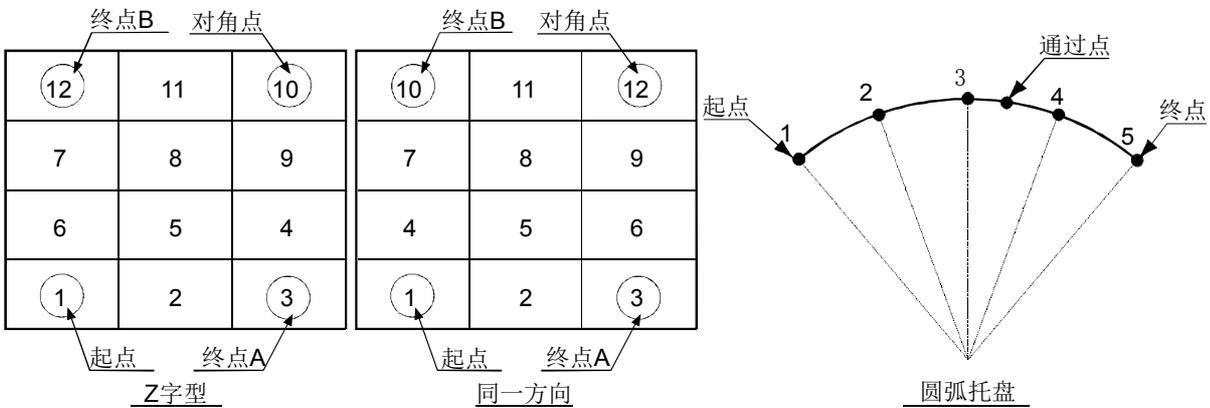
■ 指令

指令	说明
Def Plt	定义使用的 Pallet。
Plt	用运算求得 Pallet 上的指定位置。

■ 例子

例子	说明
Def Plt 1, P1, P2, P3, P4, 4, 3, 1.....	' 定义在指定托盘号码 1, 有起点 =P1、终点 A=P2、终点 B=P3、对角点 =P4 的 4 点地方和这样大小的尺寸内, 个数 A=4、个数 B=3 的合计 12 个 (4×3) 的作业位置, 用托盘模型 =1 (Z 字型) 进行运算。
Def Plt 2, P1, P2, P3, , 8, 5, 2.....	' 定义在托盘号码 2, 有起点 =P1、终点 A=P2、终点 B=P3 的 3 点指定地方在数量大的中间, 有 个数 A=8、个数 B=5 的合计 40 个 (5×8) 作业位置, 用托盘模型 =2 (同一方向) 进行运算。
Def Plt 3, P1, P2, P3, , 5, 1, 3.....	' 定义在托盘号码 3, 在有起点 =P1、通过点 =P2、终点 =P3 的 3 点指定圆弧上, 合计 5 个的作业位置, 使用圆弧托盘定义。
(Plt 1, 5)	' 运算托盘号码 1 的第 5 个位置。
(Plt 1, M1)	' 显示数值变量 M1 的值, 运算托盘号码 1 内的位置。

位置的指定和托盘模型的关系如下所示。



托盘模型 =1 (Z 字型)
(姿势均分)
托盘模型 =11 (Z 字型)
(姿势固定)

托盘模型 =2 (同一方向)
(姿势均分)
托盘模型 =12 (同一方向)
(姿势固定)

托盘模型 =3 (圆弧托盘)
(姿势均分)
托盘模型 =13 (圆弧托盘)
(姿势固定)

<定义码垛时，有关位置数据的姿势注意事项>

⚠ 注意

将姿势成份（A，B，C）为 180 度附近的位置数据视为码垛定义的<起点>、<终点 A，B>、<对角点>的情况下，请确认下列「[说明](#)」内容后实施。

■ 说明

姿势成份（A，B，C）变成在 180 度的位置，即使是相同的姿势，成份的值也会变成 +180 度或 -180 度的其中一个。这是因为内部的运算误差所造成，会变成那一个值并没有规则性。

将此位置作为码垛定义的<起点>、<终点>、<对角点>的情况下，在同一个姿势成份里有混合了 +180 度和 -180 度，托盘的格子点的位置，为了将 -180 度 ~ +180 度之间分割计算出，抓手会旋转且会有预期外的动作发生。

因为姿势成份的 +180 度和 -180 度变成相同的姿势，因此在码垛定义使用的位置数据，请使用符号 + 或符号 - 的其中一个备齐。

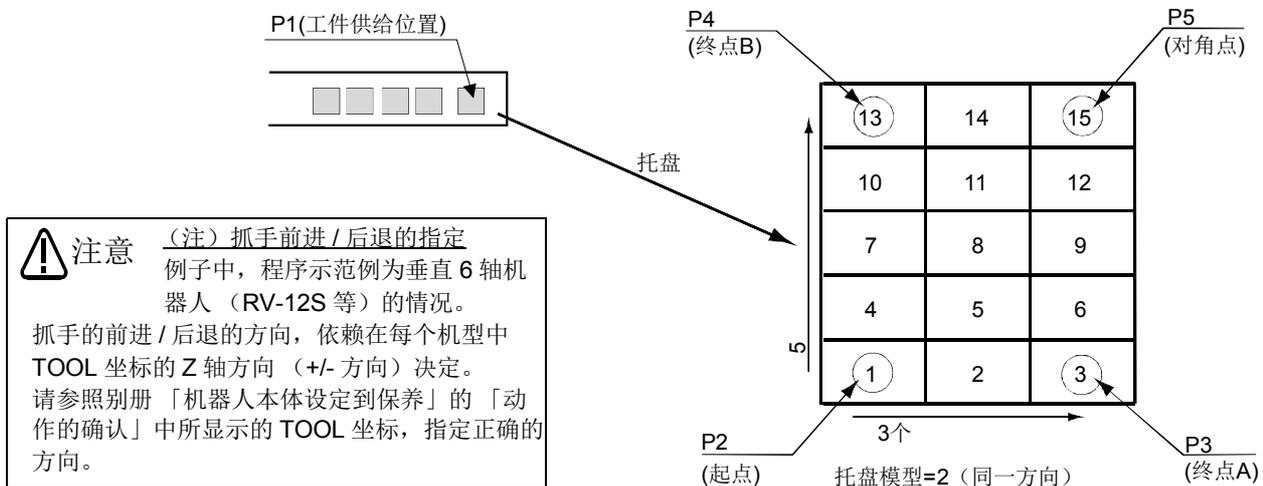
此外，姿势成份在 ± 180 度附近（例如：+179 度和 -179 度）的情况下，也会变成有符号混合的现象。此情况下，在姿势成份里加算或减算 360 度，请备齐符号将值补偿。（例如：在 -179 度补上 + 的符号的情况下，加上 360 度为，请补偿在 +181 度。）

在「[程序例 1](#)」，在托盘的全格子点，在抓手的方向为相同的情况下，（A，B，C 轴的值为相同）、将终点（P3、P4）和对角点（P5）的姿势成份对准起点（P2）的例子（行号码 10 ~ 90 号）和在「[程序的例 2](#)」，在 Pallet 定义位置的姿势成份为 ± 180 度附近，终点（P3、P4）和对角点（P5）的 C 轴的值为比 -178 度还小，或比 +178 度还大的情况下，在和起点（P2）相同符号补偿值的例子（行号码 10 ~ 100）。（将 ± 178 度列为需补偿的值），请在码垛的精度不良、抓手的方向偏离时，不得不补偿情况下的参考。

此外，在托盘模型里指定 11 ~ 13 的话，在码垛运算求到的位置变量的姿势会被代入<起点>、也可以固定抓手的方向。（指定 1 ~ 3 情况下的姿势数据，会被代入<起点> - <终点>之间均分的值）

■ 程序例子

• 机器人的动作



 注意

依据码垛运算（Pit 指令）所算出格子点的构造标志（位置数据的 FI1），采用码垛定义的起点的值。

因此，在码垛定义的各点构造标志的不同位置数据使用的话，会变成和希望的码垛动作不同的动作。

请使用在码垛定义的起点、终点 A、B 对角点里，构造标志值全部相同的位置数据。此外，关于格子点的多旋转标志（位置数据的 FI2）也是，采用码垛定义的起点的值。在码垛的各点多旋转标志的不同位置数据使用的话，因码垛动作的经过，因此机器人的位置及插补指令的种类（关节插补、直线插补等），抓手会旋转且产生预期外的动作。那样的情况下，请使用插补指令的自变量 Type，适当的将姿势的绕道 / 近处动作做适当的设定，将抓手调整为希望的动作。

• 程序例 1

关于码垛的全格子点，抓手的动作相同情况（A，B，C 轴的值为相同）

程序	说明
1 P3.A=P2.A	在P3的姿势成份（A）里代入P2的姿势成份（A）。
2 P3.B=P2.B	在P3的姿势成份（B）里代入P2的姿势成份（B）。
3 P3.C=P2.C	在P3的姿势成份（C）里代入P2的姿势成份（C）。
4 P4.A=P2.A	在P4的姿势成份（A）里代入P2的姿势成份（A）。
5 P4.B=P2.B	在P4的姿势成份（B）里代入P2的姿势成份（B）。
6 P4.C=P2.C	在P4的姿势成份（C）里代入P2的姿势成份（C）。
7 P5.A=P2.A	在P5的姿势成份（A）里代入P2的姿势成份（A）。
8 P5.B=P2.B	在P5的姿势成份（B）里代入P2的姿势成份（B）。
9 P5.C=P2.C	在P5的姿势成份（C）里代入P2的姿势成份（C）。
10 Def Pit 1, P2, P3, P4, P5, 3, 5, 2	定义码垛。 托盘号码=1、起点=P2、终点A=P3、终点B=P4、对角点=P5、 个数A=3、个数B=5、托盘模型=2（同一方向）
11 M1=1	在数值变量M1里代入值1。（M1在计数器使用）
12 *LOOP	作为跳转对象，将标准LOOP指定。
13 Mov P1, -50（注）	以关节插补动作从P1移动在抓手方向后退50mm位置。
14 OvrD 50	将动作速度设定为最高速度的一半。
15 Mvs P1	往P1直线移动。（去抓取工件）
16 HClose 1	关闭抓手1。（抓住工件）
17 Dly 0.5	等待0.5秒。
18 OvrD 100	将动作速度设定为最大。
19 Mvs, -50（注）	（4）以直线动作从现在位置（P1）移动在抓手方向后退50mm位置。（抓住工件向上）
20 P10=（Pit 1, M1）	显示数值变量M1的值，运算Pallet号码1内的位置，将结果代入P10。
21 Mov P10, -50（注）	以关节插补动作从P10移动在抓手方向后退50mm位置。
22 OvrD 50	将动作速度设定为最高速度的一半。
23 Mvs P10	以直线移动到P10。（去放置工件）
24 HOpen 1	打开抓手1。（放开工件）
25 Dly 0.5	等待0.5秒。
26 OvrD 100	将动作速度设定为最大。
27 Mvs, -50（注）	以直线动作从现在位置（P10）移动在抓手方向后退50mm位置。 （放开工件）
28 M1=M1+1	在数值变量M1的值补足1。（将码垛计数器前进）
29 If M1<=15 Then *LOOP	数值变量M1的值在15以下的话，循环往标准LOOP跳转处理。如果那样的话， 往下一行前进。
30 End	程序结束。

• 程序的例 2
姿势成份为 ±180 度附近的情况的补偿

程序	说明
1 If Deg (P2.C) <0 Then GoTo*MINUS	' 确认P2的姿势成份 (C) 的符号, 为- (负) 的话, 跳转到 Level MINUS。
2 If Deg (P3.C) <-178 Then P3.C =P3.C+Rad (+360)	' P3的姿势成份为-180度附近的情况下, 加算360度补偿在+值。
3 If Deg (P4.C) <-178 Then P4.C =P4.C+Rad (+360)	' P4的姿势成份为-180度附近的情况下, 加算360度补偿在+值。
4 If Deg (P5.C) <-178 Then P5.C=P5.C+Rad (+360)	' P5的姿势成份为-180度附近的情况下, 加算360度补偿在+值。
5 GoTo *DEFINE	' 无条件跳转到Level DEFINE的行。
6 *MINUS	' 作为跳转处, 指定Level MINUS。
7 If Deg (P3.C) > +178 Then P3.C =P3.C-Rad (+360)	' P3的姿势成份为+180度附近的情况下, 加算360度补偿在-值。
8 If Deg (P4.C) > +178 Then P4.C =P4.C-Rad (+360)	' P4的姿势成份为+180度附近的情况下, 加算360度补偿在-值。
9 If Deg (P5.C) > +178 Then P5.C =P5.C-Rad (+360)	' P5的姿势成份为+180度附近的情况下, 加算360度补偿在-值。
10 *DEFINE	作为跳转处, 指定Level DEFINE。
11 Def Plt 1, P2, P3, P4, P5, 3, 5, 2	' 定义码垛。 码垛号码=1、起点=P2、终点A=P3、 终点B=P4、对角点=P5、个数A=3、个数B=5、 Pallet Patten=2 (同一方向)
12 M1=1	' 在数值变量M1代入值1。 (M1使用在计数时)
13 *LOOP	' 作为跳转处, 指定Level LOOP。
14 Mov P1, -50 (注)	' 从P1在抓手方向后退50mm的位置以关节插补移动。
15 OvrD 50	' 动作速度设定在最高速度的一半。
16 Mvs P1	' 往P1以直线移动。(去握住工件)
17 HClose 1	' HAND1闭。(握住工件)
18 Dly 0.5	' 等待0.5秒。
19 OvrD 100	' 动作速度设定在最大。
20 Mvs, -50 (注)	' 从现在位置 (P1) 往抓手方向后退50mm位直线移动。 (握住工件向上)
21 P10= (Plt 1, M1)	' 显示数值变量M1的值, 运算码垛号码1内的位置, 将结果代入P10。
22 Mov P10, -50 (注)	' 从P10往抓手方向后退50mm位置以关节插补移动。
23 OvrD 50	' 动作速度设定在最高速度的一半。
24 Mvs P10	' 往P10直线移动。(去放置工件)
25 HOpen 1	' HAND1开。(放开工件)
26 Dly 0.5	' 等待0.5秒。
27 OvrD 100	' 动作速度设定在最大。
28 Mvs, -50	' 从现在位置 (P10) 往抓手方向后退50mm的直线移动。 (放开工件)
29 M1=M1+1	' 数值变量M1的值加上1。 (码垛计数增进)
30 If M1<=15 Then *LOOP	' 数值变量M1的值为15以下的话, 跳转到Level LOOP, 重复处理。如果不是那样的话, 往下一行前进。
31 End	' 程序结束。

4.1.3 程控

可以控制分支、插入、定时器、呼叫子程序、停止等程序的流动。

(1) 无条件分支·条件分支·待机

可以无条件的将程序往指定行流动，或条件判断后分支。

■ 指令

指令	说明
GoTo	在指定的标签无条件跳转。
On GoTo	对应指定变量的值，做跳转。值的条件为整数顺序。(1,2,3,4,.....)
If Then Else (在 1 行的记述)	对应指定条件指令做执行。值的条件可任意的指定。 条件成立的情况，执行 Then 的后面的记述、条件不成立的情况，执行 Else 的后面记述。 以一行记述。
If Then Else Endif (在复数行的记述)	对应指定变量及其值的指定条件进行复数行处理。值的条件可任意的指定。 条件的种类，每 1 指令 1 个种类。 条件成立的情况，执行从 Then 的下一行开始到 Else 为止的行、条件不成立的情况下，执行从 Else 的后面行开始到 Endif 为止的行。
Select Case End Select	对应指定变量及其值的指定条件进行跳转。值的条件可任意的指定。 条件的种类，每 1 指令可以指定复数种类。
Wait	等待到变量成为指定的值为止。

■ 例子

例子	说明
GoTo *FIN	' 无条件跳转到标准 FIN 的行。
On M1 GoTo *L1, *L2, *L3	' 数值变量 M1 的值为 1 的话，跳转到标准 L1，为 2 的话，跳转到标准 2，为 3 的话，跳转到标准 3 在值不相当时，往下一行前进。
If M1=1 Then *L100.....	' 数值变量 M1 的值为 1 的话，往标准 *L1 分支，假如不是那样的话会往下一行前进。
If M1=1 Then *L10 Else *L20.....	' 数值变量 M1 的值为 1 的话，往标准 *L10 分支，假如不是那样的话会标准 *L20 分支。
If M1=1 Then	' 数值变量 M1 的值为 1 的话，执行 M2=1、M3=2、M1 为 1 以外的，执行 M2=-1, M3=-2。
M2=1	
M3=2	
Else	
M2=-1	
M3=-2	
Endif	
Select M1.....	' 依据变量 M1 的值，分支到适当的 Case。
Case 10	' 值为 10 的话，只在 Case 10 和下一个的 Case IS 11 之间执行。
:	
Break	
Case 11.....	' 值为 11 的话，只在 Case11 和下一个的 Case IS < 5 之间执行。
:	
Break	
Case IS < 5.....	' 值比 5 小的话，只在 Case IS < 5 和下一个的 Case 6 TO 9 之间执行。
:	
Break	
Case 6 TO 9	' 值为 6 到 9 的话，只在 Case 6 TO 9 和下一个的 Default 之间执行。
:	
Break	
Default	' 值皆不符合上列任一个时，只在 Default 和下一个的 End Select 之间执行。
:	
Break	
End Select.....	' Select Case 文结束。
Wait M_In (1) =1.....	' 等待到输入信号位 1 开启为止。

■ 相关功能

功能	说明
重复.....	第 115 页的 [(2) 循环]
插入.....	第 116 页的 [(3) 插入]
子程序.....	第 117 页的 [(4) 子程序]
外部信号的输入.....	第 120 页的 [(1) 信号输入]

(2) 循环

可对应指定条件，重复执行复数的指令。

■ 指令

指令	说明
For Next	将 For 文和 Next 文之间，重复执行到满足指定条件。
While WEnd	将 While 文和 WEnd 文之间，重复执行到满足指定条件。

■ 例子

例子	说明
For M1=1 To 10 : Next	' For 文和 Next 文之间重复执行 10 次。 数值变量 M1 的值，最初代入 1，每次重复就加上 1。
For M1=0 To 10 Step 2..... : Next	' For 文和 Next 文之间重复执行 6 次。 数值变量 M1 的值，最初代入 0，每次重复就加上 2。
While (M1>=1) And (M1<=10) : WEnd	' 数值变量 M1 的值为 1 以上 10 以下之间，重复执行 While 文和 WEnd 文之间。

■ 相关功能

功能	说明
无条件分支·条件分支.....	第 113 页的 [(1) 无条件分支·条件分支·待机]
插入.....	第 116 页的 [(3) 插入]
输入信号等待.....	第 120 页的 [(1) 信号输入]

(3) 插入

指定条件成立的话，执行中的指令插入，且指定行可以分支。

■ 指令

指令	说明
Def Act	定义插入条件和插入时发生时的处理。
Act	指定插入的许可 / 禁止。
Return	将插入处理视为子程序的呼出时，后退到原来插入的行。

■ 例子

例子	说明
Def Act 1, M_In (10) =1 GoSub *L100.....	' 定义假设在插入号码 1，输入信号位 10 开启的话，在机器人减速停止后，呼出标准 *L100 的子程序。减速时间依赖 Accel 指令及 Ovrd 指令。
Def Act 2, M_In (11) =1 GoSub *L200, L.....	' 定义假设在插入号码 2，输入信号位 11 开启的话，执行中的指令完成后，呼出标准 *L200 的子程序。
Def Act 3, M_In (12) =1 GoSub *L300, S.....	' 定义假设在插入号码 3，输入信号位 12 开启的话，将机器人在最短时间、最短距离减速停止后，呼出标准 *L300 的子程序。
Act 1=1.....	' 许可优先号码 1 的插入。
:	
Act 2=0.....	' 禁止优先号码 2 的插入。
:	
*L100	' 发生插入，后退到行。
:	
Return 0.....	
*L200	' 发生插入，后退到下一行。
:	
Return 1.....	
*L300	' 发生插入，后退到行。
:	
Return 0.....	

■ 相关功能

功能	说明
无条件分支·条件分支.....	第 113 页的 [(1) 无条件分支·条件分支·待机]
子程序.....	第 117 页的 [(4) 子程序]
通信.....	第 121 页的 [4.1.5 通信]

(4) 子程序

可以使用子程序和子程序。

使用本功能可以共享程序，以节省单步数且将程序做成阶层构造，更浅而易懂。

■ 指令

指令	说明
GoSub	呼出指定标准的子程序。
On GoSub	呼出对应指定变量的值的子程序。值的条件为整数值顺序。(1,2,3,4,.....)
Return	用指令 GoSub 呼出，且后退到下一行。
CallP	呼出叫已指定的程序。在被呼出的程序的 End 指令里，后退到原始程序的下一个行。可以将调用程序，作为自变量数据传送。
FPm	在指令 CallP 呼叫出的程序里，接受自变量。

■ 例子

例子	说明
GoSub *GET.....	' 呼出由卷标 GET 开始的副程序。
On M1 GoSub *L1, *L2, *L3.....	' 数值变量 M1 的值为 1 的话会呼出标签 L1，为 2 的话标签 L2，为 3 的话标签 L3 的副程序。
:	当值不相当时，往下一个行前进。
:	
:	
*L1	
:	
Return	
*L2	
:	
Return	
*L3	
:	
Return.....	' 使用 GoSub 指令呼出后，后退到下一个行。
Return.....	' 使用 GoSub 指令呼出后，后退到下一个行。
CallP "10"	' 呼出 10 号的程序。
CallP "20", M1, P1	' 在往 20 号的程序里呼出数值变量 M1 和位置变量 P1。
FPm M10, P10.....	' 在子程序里，以 CallP 指令将数值变量传递在 M10、位置变量接收为 P10。

■ 相关功能

功能	说明
插入	第 116 页的「(3) 插入」
通信	第 121 页的「4.1.5 通信」
无条件分支	第 113 页的「(1) 无条件分支·条件分支·待机」

(5) 定时器

可以只等待指定时间，且可以在输出信号指定时间宽度，输出脉冲。

■ 指令

指令	说明
Dly	作为指定时间的定时器功能。

■ 例子

例子	说明
Dly 0.05	只等待 0.05 秒。
M_Out (10) =1 Dly 0.5	等待 0.5 秒后输出信号位 10 开启。

■ 相关功能

功能	说明
脉冲信号输出	第 120 页的「(2) 信号输出」

(6) 停止

可以停止程序的执行。移动中的机器人减速停止。

■ 指令

指令	说明
Hlt	将机器人停止，且中断程序的执行。起动由下一个行执行。
End	定义程序的 1 环的结束。连续运行时，在 End 指令执行再度由前头行起动。循环运行时，循环停止在 End 指令，程序结束。

■ 例文

例文	说明
Hlt	' 中断程序的执行。
If M_In (20) =1 Then Hlt	' 输入信号位置 20 开启后，程序中断。
Mov P1 WthIf M_In (18) =1, Hlt	' 在往 P1 移动中，输入信号位 18 开启后，程序中断。
End	' 即使在程序途中也会将程序结束。

■ 关连功能

功能	说明
附随句	第 279 页的「Wth (With)」

4.1.4 外部信号的输出

在机器人依外部机器（PLC等）控制情况，对一般的信号控制方法做说明。

(1) 信号输入

可以读取由 PLC 等的外部机器输入的信号。

确认机器人（系统）状态变量（M_In（）等）所使用的输入信号。关于机器人（系统）状态变量的详细内容请参照第 287 页的「4.14 机器人（系统）状态变量的详细说明」。

■ 指令

指令	说明
Wait	输入信号在变成指定状态前待机。

■ 系统状态变量（机器人状态变量）

M_In, M_Inb, M_Inw, M_DIn

■ 例文

例文	说明
Wait M_In (1) =1	输入信号位 1 到开启前待机。
M1=M_Inb (20)	在数值变量 M1 里将输入信号位，以从 20 到 27 的 8 个位的状态为数值代入。
M1=M_Inw (5)	在数值变量 M1 里将输入信号位，以从 5 到 20 的 16 个位的状态为数值代入。

■ 关连功能

功能	说明
信号输出	第 120 页的「(2) 信号输出」
依输入信号分支	第 113 页的「(1) 无条件分支·条件分支·待机」
依输入信号插入	第 116 页的「(3) 插入」

(2) 信号输出

可以将信号输出到 PLC 等的外部机器。

确认机器人（系统）状态变量（M_Out（）等）所使用的输入信号。关于机器人（系统）状态变量的详细内容请参照第 153 页的「4.6 机器人（系统）状态变量」。

■ 指令

指令	说明
Clr	以参数的输入输出信号复位为基础，将通用输出信号清除。

■ 系统状态变量（机器人状态变量）

M_Out, M_Outb, M_Outw, M_DOut

■ 例文

例文	说明
Clr 1	输出复位模式为基础清除。
M_Out (1) =1	将输出信号位开启。
M_Outb (8) =0	将输出信号位从 8 到 15 的 8 个位关闭。
M_Outw (20) =0	将输出信号位从 20 到 35 的 16 个位关闭。
M_Out (1) =1 Dly 0.5	将输出信号位在 0.5 秒间开启。（脉冲输出）
M_Outb (10) =&H0F	出将输出信号位从 10 到 13 的 4 个位开启，从 14 到 17 的 4 个位关闭。

■ 关连功能

功能	说明
信号输入	第 120 页的「(1) 信号输入」
定时器	第 118 页的「(5) 定时器」

4.1.5 通信

可以和计算机等的外部机器间的数据通信。

■ 指令

指令	说明
Open	将通信端口开启。
Close	将通信端口关闭。
Print#	将数据以 ASCII 形式输出。做为结束 CODE 将 CR 输出。
Input#	将数据以 ASCII 形式输入。做为结束 CODE 将 CR 输出。
On Com GoSub	在从通信端口发生插入时，定义呼出子程序。插入会发生在由外部机器将数据输入的情况。
Com On	许可从通信端口的插入处理。
Com Off	禁止从通信端口的插入处理。即使有插入也会变成无效。
Com Stop	停止从通信端口的插入处理。即使有插入的情况下也会记住，得到许可后执行。

■ 例文

例文	说明
Open "COM1: " As #1	' 将通信端口 COM1 作为文件号码 1 开启。
Close #1	' 将文件号码 1 关闭。
Close	' 将已开启的文件全部关闭。
Print# 1, "TEST".....	' 将文字列 "TEST" 输出到文件号码 1。
Print# 2, "M1="; M1.....	' 将在文字列 "M1=" 连续 M1 的值输出到文件号码 2。 输出数据例子: "M1=1"+CR (M1 的值为 1 的情况)
PRINT# 3, P1.....	' 将位置变量 P1 的坐标值输出到文件号码 3。 输出数据例子: " (123.7, 238.9, 33.1, 19.3, 0, 0) (1, 0) "+CR (X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0 的情况)
Print# 1, M5, P5.....	' 将数值变量 M5 的值和位置变量 P5 的坐标值输出到文件号码 1。 M5 和 P5 以逗号 (16 进 2C) 区分。 输出数据例子: "8, (123.7, 238.9, 33.1, 19.3, 0, 0) (1, 0) "+CR (M5=8、P5 为 X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0 的情况)
Input# 1, M3.....	' 将输入的数据转为数值，代入数值变量 M3。 输入数据例子: "8"+CR (代入值 8 的情况)
Input# 1, P10.....	' 输入的数据转为坐标值，代入位置变量 P10。 输入数据例子: " (123.7, 238.9, 33.1, 19.3, 0, 0) (1, 0) "+CR (P5 会变为 X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0)
Input# 1, M8, P6	' 将输入的最初数据转为数值，在数值变量 M8，将逗号后下一个数据视为坐标值，代入位置变量 P6。 M8 和 P6 以逗号 (16 进 2C) 区分。 输入数据例子: "7, (123.7, 238.9, 33.1, 19.3, 0, 0) (1, 0) "+CR (M8=7、P6 会变为 X=123.7、Y=238.9、Z=33.1、A=19.3、B=0、C=0、FL1=1、FL2=0)
On Com (2) GoSub *RECV.....	' 数据输入到通信端口 COM2 的话，定义呼出标准 RECV 的行的子程序。
Com (1) On.....	' 允许由通信端口 COM1 的插入。
Com (2) Off.....	' 禁止 (无效) 通信端口 COM2 的插入。
Com (1) Stop.....	' 停止 (保留) 由通信端口 COM1 的插入。

■ 关连功能

功能	说明
子程序	第 117 页的「(4) 子程序」
插入	第 116 页的「(3) 插入」

4.1.6 运算

下列为可以使用的运算符及其意义、例文示范。

(1) 运算符一览

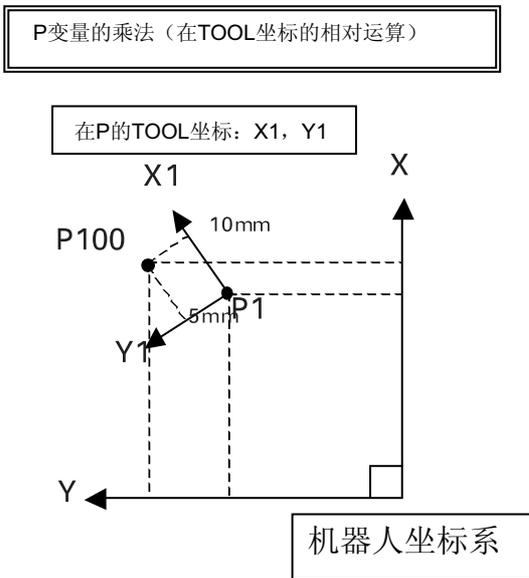
区分	运算	意义	例文
代入	=	将右边代入左边	P1=P2 ; 将 P2 代入位置变量 P1。 P5=P_Curr ; 将现在的坐标值代入现在位置变量 P5。 P10.Z=100.0 ; 将位置变量 P10 的 Z 坐标值设为 100.0。 M1=1 ; 将值 1 代入数值变量。 STS\$="OK" ; 将字符串变量 STS\$ 代入称为 OK 的字符串。
数值运算	+	加算	P10=P1+P2 注1) ; 将 P1 和 P2 各坐标成份加算结果代入位置变量 P10。 Mov P8+P9 ; 移动到将位置变量 P8 和 P9 的各坐标成份加算的位置。 M1=M1+1 ; 数值变量 M1 的值加 1。 STS\$="ERR"+"001" ; 在字符串变量 STS\$ 代入结合 ERR 及 001 的字符串。
	-	减算	P10=P1-P2 ; 在位置变量 P10 代入从 P1 到 P2 的各坐标成份减算结果。 Mov P8-P9 ; 移动到位置变量从 P8 到 P9 的各坐标成份减算的位置。 M1=M1-1 ; 从数值变量 M1 的值减去 1。
	*	乘算	P1=P10*P3 注2) ; 将从 P1 到 P10 的相对变换结果代入位置变量 P1。 M1=M1*5 ; 将数值变量 M1 的值乘上 5 倍。
	/	除算	P1=P10/P3 ; 将从 P1 到 P10 的相对变换结果代入位置变量 P1。 M1=M1/2 ; 将数值变量 M1 的值除 2。
	^	指数运算	M1=M1 ^ 2 ; 自乘数值变量 M1 的值。
	\	整数除算	M1=M1\3 ; 将数值变量 M1 的值除以 3 后取整数。(小数点以下舍去)
	Mod	余数运算	M1=M1 Mod 3 ; 将数值变量 M1 的值除以 3 后取余数。
	-	符号反转	P1=-P1 ; 将位置变量 P1 的各坐标成份符号反转。 M1=-M1 ; 将数值变量 M1 值的符号反转。
比较运算	=	比较是否等于	If M1=1 Then *L2 ; 数值变量 M1 的值为 1 的话, 往 L2 分支。 If STS\$="OK" Then *L1 ; 若字符串变量 STS\$ 为 OK 的字符串, 则往 L1 分支。
	<> 或 ><	比较是否不相等	If M1<>2 Then *L3 ; 若数值变量 M1 的值不为 2 的话, 往 L3 分支。 If STS\$<>"OK" Then *L9 ; 若字符串变量 STS\$ 为 OK 的字符串, 则往 L9 分支。
	<	比较是小于	If M1<10 Then *L3 ; 数值变量 M1 的值小于 10 的话, 往 L3 分支。 If Len (STS\$) <3 Then *L1 ; 字符串变量 STS\$ 的字符串数小于 3 的话, 往 L1 分支。
	>	比较是否大于	If M1>9 Then *L2 ; 数值变量 M1 的值大 9 的话, L2 分支。 If Len (STS\$) >2 Then *L3 ; 字符串变量 STS\$ 的字符串数大于 2 的话, 往 L3 分支。
	=< 或 或小于等于 <=	比较是否等于小于 或小于等于	If M1<=10 Then *L2 ; 数值变量 M1 的值为 10 或比 10 小的话, 往 L2 分支。 If Len (STS\$) <=5 Then *L3 ; 字符串变量 STS\$ 的字符串数为 5 或比 5 小的话, 往 L3 分支。
	=> 或 或大于等于 >=	比较是否等于大于 或大于等于	If M1>=11 Then *L2 ; 数值变量 M1 的值为 11 或比 11 小的话, 往 L2 分支。 If Len (STS\$) >=6 Then *L3 ; 字符串变量 STS\$ 的字符串数为 6 或比 6 小的话, 往 L3 分支。
	逻辑运算	And	逻辑与运算
Or		逻辑或运算	M_Outb (20) =M1 Or &H80 ; 在输出信号位 20 到 27, 输出数值变量 M1 的值。此时, 输出信号位 27 会变成常开。
Not		否定运算	M1=Not M_Inw (1) ; 将输入信号位 1 到 16 的状态取反后代入数值变量。
Xor		异或	M2=M1 Xor M_Inw (1) ; 在数值变量 M2, 将 M1 和输入信号位从 1 到 16 状态和以数值代入。
<<		逻辑左移动运算	M1=M1 << 2 ; 将数值变量 M1 向左移动 2 位。
>>		逻辑右移动运算	M1=M1 >> 1 ; 将数值变量 M1 向右移动 1 位。

注 1) 请参照第 123 页的「位置数据的相对运算 (加算)」。

注 2) 请参照第 123 页的「位置数据的相对运算 (乘算)」。

(2) 位置数据的相对运算（乘算）

数值变量的运算为一般的四则运算，但是，位置变量的运算并不是单纯的四则运算，要执行坐标变换。下列为说明的简单例子。



相对运算的例子（乘算）

```

1 P2= (10,5,0,0,0,0) (0,0)
2 P100=P1*P2
3 Mov P1
4 Mvs P100
P1= (200,150,100,0,0,45) (4,0)
    
```

在此例子中的示教位置 P1，在 P1 的 TOOL 坐标内，使相对移动情况的例子。

P2 的 X、Y 的值变成在 TOOL 坐标内的移动量。

P 变量的乘算会变成相对运算。

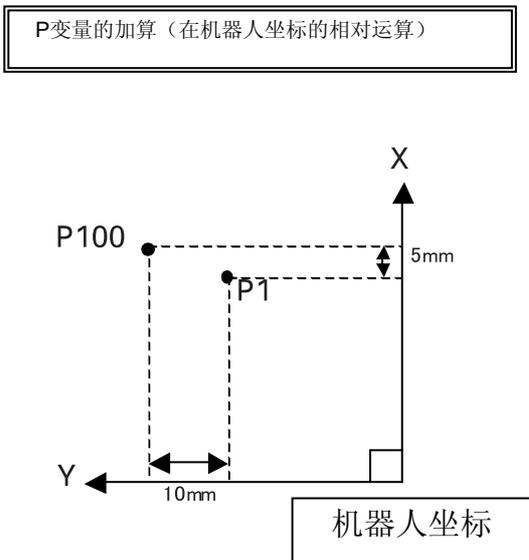
请注意依照乘算的顺号，意义会不同。

设定了相对移动量的变量（P2）会在后面记述。

P2 的姿势轴部份（A,B,C）为 0 的情况下，直接以 P1 的姿势使用。有值的情况下，将 P1 的姿势后退变成在 Z, Y, X 旋转（C, B, A 顺序）里旋转姿势。

乘算为在 TOOL 坐标内的加算，除算的情况下，在 TOOL 坐标内会变成减算。

(3) 位置数据的相对运算（加算）



相对运算的例子（加算）

```

1 P2= (5,10,0,0,0,0) (0,0)
2 P100=P1+P2
3 Mov P1
4 Mvs P100
P1= (200,150,100,0,0,45) (4,0)
    
```

此例子为在示教位置 P1，在机器人坐标内相对移动情况的例子。

P2 的 X, Y 的值会变成在机器人坐标内的移动量。

P 变量量的加算会变成相对演算。

在 P2 的 C 轴部份，输入值的话，也可变更 P100 的 C 轴的旋转成份。在 P1 的 C 轴成份会变成 P2 的 C 轴成份加算后的值。

注意) 在上列例子中，为了简单说明，所以采用二次元说明，但是实际上是以三次元运算。此外，TOOL 坐标会依机器人的姿势而有所不同。

4.1.7 附随句

可以在移动指令附加处理。

■ 附随句

附随句	说明
Wth	在移动指令附加无条件处理。
Wthlf	在移动指令附加条件处理。

■ 例文

例文	说明
Mov P1 Wth M_Out (20) =1	往 P1 的移动开始且同时将输出信号位 20 开启。
Mov P1 Wthlf M_In (20) =1, Hlt	往 P1 的移动中，输入信号位 20 为开启的话，则会停止。
Mov P1 Wthlf M_In (19) =1, Skip	往 P1 的移动中，输入信号位 19 为开启的话，中断往 P1 的移动，往下一行前进。

■ 关连功能

功能	说明
关节插补动作	第 99 页的「(1) 关节插补动作」
直线插补动作	第 100 页的「(2) 直线插补动作」
圆弧插补动作	第 101 页的「(3) 圆弧插补动作」
停止	第 119 页的「(6) 停止」

4.2 MELFA-BASIC V和MELFA-BASIC IV的差异

4.2.1 关于MELFA-BASIC V

CR7xx 系列机器人控制器，在机器人语言有搭载 MELFA-BASIC V。此 MELFA-BASIC V 将以往搭载在 CRn-500 系列机器人控制器里的 MELFA-BASIC IV 做为基础，更为使用简单及记述容易。在此，说明其差异。

4.2.2 MELFA-BASIC V的特点

MELFA-BASIC V 和 MELFA-BASIC IV 比较的话，拥有下列特点。

- (1) 不需要行号码。以往的行号码作为 STEP 号，自动编号。
因为去除了加上行号码的繁锁，使程序连接的效率提升及减低排错时因为行号码的错误，使损失减少。
- (2) 在指令及变量名等可以使用小写。依据此点，可大幅提升可读性。
- (3) 追加指令和函数，将功能提升。

4.2.3 和MELFA-BASIC IV的比较

MELFA-BASIC V 和 IV 的比较如表 4-2 所示。

表 4-2: 和 MELFA-BASIC IV 的比较

项目	MELFA-BASIC V	MELFA-BASIC IV
程序名	12 文字以内的英文大写和数字 (为了在操作面板上显示，建议使用 4 个文字以内)	
可使用文字	<ul style="list-style-type: none"> • 英文字 (大写、小写) • 数字 • 记号 	<ul style="list-style-type: none"> • 英文字 (只有大写。小写只能在指令和字符串数据) • 数字 • 记号
步号 (行号码)	作为步号，在程序登录时会自动采号。	作为行号码，在程序连接时，一定要输入。
1 行的长度	240 文字以内	127 文字以内
变量名	16 文字以内。在变量名可以使用英文大小写。大小写并没有区别，但是，在读出时会变换为在最初登录表记。	8 文字以内。在变量名里使用的英文字全部变换为大写。
标签名	16 文字以内。在标签名可以使用英文大小写。大小写并没有区别，但是，读初时变换为最初登录的表记。	8 文字以内。在标签名使用的英文字全部变换为大写。
指令	定义为英文字大小写的组合。在登录时并没有大小写区别，但是，读出时，会变换为登录在系统的表记。	全部定义为大写。登录时也用大写登录。
函数		
系统状态变量		
分支指令 (GoTo、GoSub) 对象指定的方法	用标签指定。	用卷标或行号码指定。

4.3 多重任务处理功能（Multitask）

4.3.1 所谓多重任务处理功能

所谓多重任务为复数的程序可以被并列执行的功能，可以用机器人的程序，控制周边机器的功能。多任务的执行，是指将并列执行的程序放入一个叫做多重任务插槽（全部有 32，出货时设定为 8）的容器内执行。

多重任务执行的开始，请由操作面板开始起动及专用输出入开始的起动或多重任务相关指令执行。

在图 4-1 显示多重任务的执行环境。

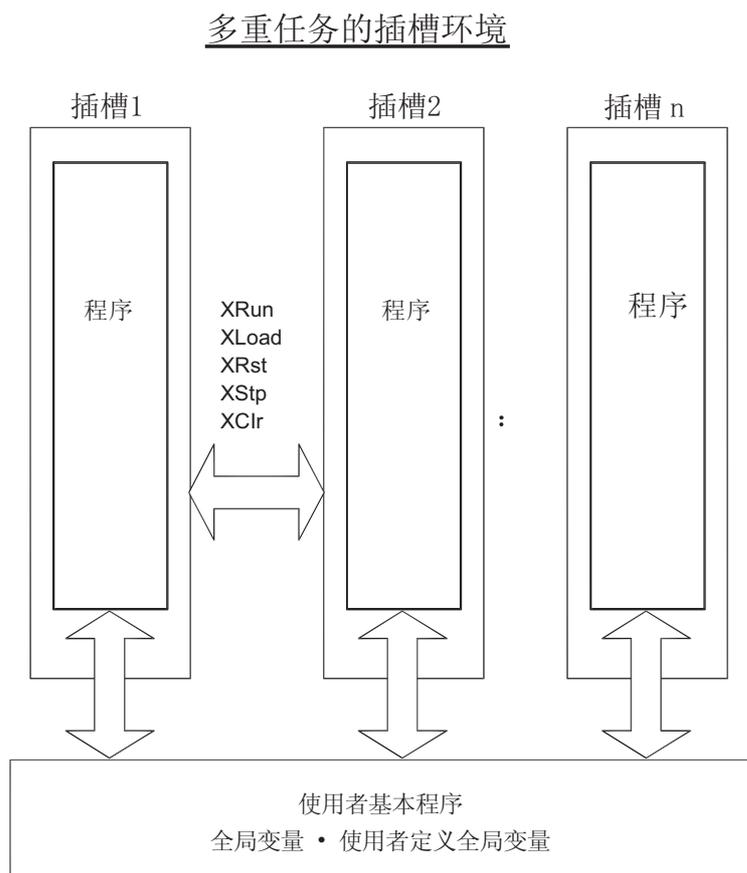


图 4-1：多重任务的执行环境

◇◆◇关于程序的执行◇◆◇

多任务的执行，是指将要执行的程序放入一个叫做多重任务插槽内执行。

例如：在运行一个程序的情况（通常在控制器的操作面板选择程序后起动的情况），控制器系统将在操作面板里选择的程序，放入多重插槽 1 中执行。

4.3.2 多重任务的执行方法

多重任务的执行方法如表 4-3 所示，有 3 种类。

表 4-3: 多重任务的执行方法

	执行种类	说明
1	从程序开始的执行	从程序中任意的位位置开始，用 MELFA-BASIC 指令，开始程序的并行执行的方法。可以指定并行执行程序及停止并行执行中的程序。 这个方法，依程序的流程，在选择并行执行程序的情况下为有效。相关指令有 XLoad (X Load)、XRun (x Run)、XStp (X Stop)、XRst (X Reset)、XClr (X clear)。详细内容请参照本书 / 第 163 页的「4.13 指令的详细说明」。
2	从控制器的操作面板、或来自外部输入输出信号的执行。	根据参数「SLT*」的设定情报（起动条件为“START”、或者“ERROR”），由于起动操作，将并行执行开始和通常并行执行、或在报警发生时，将并行执行开始的方法。请务必预先设定参数「SLT*」。以此方法，不会依赖程序的流程，以事先决定形同时执行，或是依次执行为有效。
3	在电源开启时，自动执行。	可以在控制器的电源开启后，直接通常执行。在 SLT* 参数的起动条件上，指定“ALWAYS”，则程序在控制器电源开启后，会通常执行。 依据此点，输入输出信号监视用的任务插槽等，不需要由 PLC 侧开始起动。 再来，也可以将动作用的程序由程序执行通常执行。 此情况下，请将参数的「ALWENA」值设为 1，即可执行从通常执行程序内 XRun, Xload 等的 X** 指令、Servo 指令、Reset 指令。

4.3.3 各多重插槽运行状态

各多重插槽的运行状态为依操作及指令，如图 4-2 的状态变化。各别的状态，可以用机器人（系统）状态及外部输出信号确认。

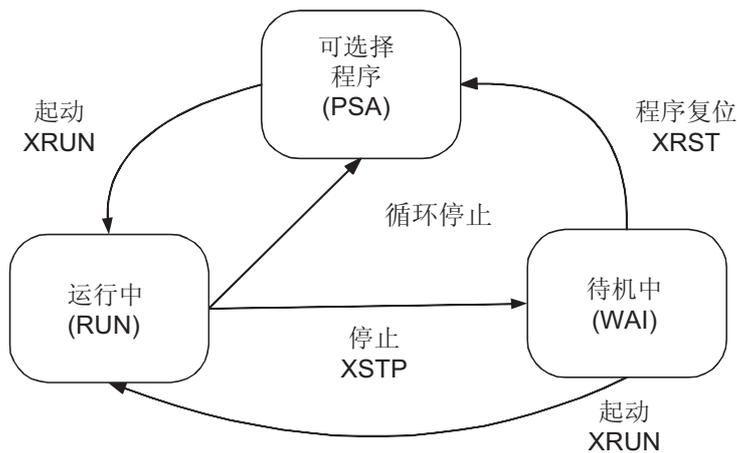


图 4-2: 各多重插槽的运行状态

< 关于多重插槽相关参数 >

每个工作插槽有 32 个 SL1~SL32（出货时 8 个），使用于设定程序名称、运行形态、起动条件、优先度。

参数的详细请参照第 407 页的「5 参数的设定功能」。

■ 指定形式

SLT* = ①程序名 ②运行形态 ③起动条件 ④优先度

■ 各设定值的意义

参数的项目	初始值	设定值说明	说明
①程序名	SLT1: 在操作面板选择的程序 SLT2 ~ 32: 在参数里设定的程序名	可设定已登录完成的程序	在多重任务运行里执行已决定的程序的情况下，以参数设定。 依据执行的程序，成为可变的情况下，可在程序里使用 XLoad、XRun 指令等指定。 SLT1 则在操作面板被选择的程序设定。
②运行形态	REP	•REP: 连续运行	在指定 REP 情况下，在程序的最终行或 End 指令执行，再一次从头开始被执行。
		•CYC: 1 循环运行	在指定 CYC 的情况下，1 循环运行后停止，且解除被选择状态。若想留下程序选择状态的情况下，请变更参数的 SLOTON。 详细内容请参照第 407 页的「5 参数的设定功能」SLOTON 项目。
③起动条件	START	•START: 依据操作面板的 START 按钮、IO 的 START 信号执行	START 使用于想要在平常起动的情况。 ^{注 1)}
		•ALWAYS: 在控制器的电源开启时执行	ALWAYS 使用于想平时执行的情况。 但是，无法执行 Mov 等的动作指令。平时执行程序的停止可以使用 XStp 指令。无法接受由操作面板或外部输入信号的停止及紧急停止。
		•ERROR: 在控制器发生报警时执行	ERROR 是想要在报警发生时执行的情况下指定。 无法使用在 Mov 指令等让机器人动作的指令执行。运行形态 (REP/CYC) 不管其设定值，都会变成 1 循环运行 (CYC)。
④优先度 (优先执行行数)	1	•1 ~ 31: 在多重任务运行时，一次执行的行数	数字越大时，多重插槽的同时执行行数也会增加。 例如：指定 SLT1 为 10、SLT2 为 5、SLT3 为 1 的情况下，SLT1 的程序执行 10 行后，SLT2 的程序执行 5 行、SLT3 的程序执行 1 行，之后会变成重复执行。

注 1) 从操作面板的起动操作及从专用输入信号的 START 信号的起动，起动条件设定为 "START"，开始同时执行全部的任务插槽。在个别起动的情况下，从专用输入信号 (S1START~S32START) 开始，用各任务插槽单位起动。

此情况下，在同专用输出参数里，预先将信号号码分配使用。关于专用输出的分配，请参照本书第 509 页的「6.2PLC 连接输出功能」。

■ 设定例

在多重插槽 2 指定的情况下的参数设定范例。

指定内容 程序名: 5
运行形态: 连续运行
起动条件: 通常执行
优先度: 10
SLT2=5, REP, ALWAYS, 10

4.3.4 多任务程序制作上的注意点

(1) 任务数目和处理时间的关系

多任务时，看起来像是将复数的机器人程序并列处理，但实际上是一边转换程序，一边执行每 1 行（行数可变更，请参照第 407 页的「5 参数的设定功能」里的「SLTn」参数）。任务增加的话，全体的程序执行时间会延长。使用多任务的情况下，请将任务数降到必要的最小限数。

但是，动作指令（Mov 和 Mvs 指令等）执行中的其它任务的程序，会随时处理

(2) 指定并列执行程序行数的最大值

并列执行参数的行数，是以参数「TASKMAX」设定。（请参照第 407 页的「5 参数的设定功能」）。（初始值为 8）并列执行 9 行以上的程序的情况下，请变更此参数。

(3) 在全局变量的程序间的数据的交接方法

在多任务运行程序间的数据交接，请使用 M_00 及 P_00 等的程序全局变量（参照第 144 页的「4.4.23 程序全局变量」）及使用者定义全局变量（第 145 页的「4.4.24 使用者定义全局变量」）执行。

以下为示范例子。在此例中，将输入信号 8 的开启 / 关闭状态，由多任务插槽 2 的程序判断，且将程序开启时使用全局变量的 M_00, 传达到程序的任务插槽 1。

```

< 任务插槽 1 >
1 M_00=0                ; M_00 里代入 0。
2 *L2: If M_00=0 Then *L2 ; M_00 的值没有变为 0 前，等待。
3 M_00=0                ; M_00 里代入 0。
4 Mov P1                 ; 前进下列目的作业。
5 Mov P2
   :
100 GoTo *L2             ; 从 STEP 号码 2 开始重复。

```

```

< 任务插槽 2> （信号及变量的程序）
1 If M_In (8) <>1 Then *L4 ; 输入信号 8 没有开启的话，往 L4 分支。
2 M_00=1                  ; M_00 里代入 1。
3 M_01=2                  ; 前进下列目的作业。
4 *L4
   :

```

(4) 用机器人（系统）状态变量的程序运行状态的确认

多任务运行程序的状态是使用机器人（系统）状态变量（M_Run、M_Wai、M_Err），从哪一个任务插槽都可以参照。例）M1=M_Run（2）得到任插槽 2 的运行中状态。

关于机器人（系统）状态变量的详细内容，请参照第 153 页的「4.6 机器人（系统）状态变量」。

(5) 机器人动作的程序基本上在任务插槽 1 执行

Mov 指令等等，记述机器人本体的动作，基本上设定在任务插槽 1 后执行。

要在任务插槽 1 以外动作的情况下，请务必使用机器人本体的获得和开放指令（Get M、Rel M）。

指令的详细内容，请参照本书 / 第 163 页的「4.13 指令的详细说明」。

(6) 在平时执行程序的初始化处理

将起动条件设定为 ALWAYS 的任务插槽的程序，将其运行形态指定为 REP 的话，会连续运行（重复运行）。在此程序执行初始化处理的情况下，设定初始化完成标志使条件分支等，制作为使初始化处理不会处理 2 次以上。

（将运行形态指定为 CYC（1 循环运行）的任务插槽，因为 1 次也不会执行，因此无需做此设定处理。）

◇◆◇机构 1 被分配到任务插槽 1 ◇◆◇

在初始状态，机构 1 会自动的被分配到任务插槽 1（标准系统的机器人本体）。

因此，即使在插槽 1 即使没有获得机器（即使 Get M 指令没有执行）也可以执行移动指令，但是，若于插槽 1 以外执行移动指令的情况下，请解除任务插槽 1 的机构取得状态（执行 Rel M 指令），然后使执行移动指令的任务插槽取得机器。（请执行 Get M 指令。）

4.3.5 多任务插槽程序运行上的注意点

(1) 多任务的开始

从操作面板的起动操作及从专用输入信号的 START 信号的起动，在任务插槽参数的起动条件设定为 "START"，开始同时执行全部的任务插槽程序。

从专用输入信号（S1START~S32START）的起动，可以开始执行各个任务插槽。此情况下，在同专用输入输出参数里，预先将信号号码分配使用。关于专用输出输入的分配，请参照本书第 509 页的「6.2PLC 连接输入输出功能」。

(2) 运行状态的表示

操作面板的 [START]、[STOP] 开关的 LED 及专用输入输出信号的 START、STOP, 会显示有在参数「SLT*」的起动条件设定为 "START" 的任务插槽的程序运行状态。

只要有 1 个程序在运行，[START] 开关的 LED 灯会亮起、且专用输入输出信号的 START 会开启。

此外，全部的程序停止的话，[START] 开关的 LED 灯会熄灭、且专用输入输出信号的 START 会关闭。

专用输入输出信号的 S1 START ~ S32 START、以及 S1 STOP ~ S32 STOP，输出每一个各别的运行状态。

必要得知各别的运行状态的情况下，请在相同专用输出输入参数，分配信号号码里确认外部信号。

关于详细的专用输出输入的分配，请参照本书 / 第 509 页的「6.2PLC 连接输入输出功能」。

起动条件设定为 ALWAYS、ERROR 的程序状态，对 [START]、[STOP] 开关的 LED 没有影响。

想要确认通常执行程序的运行状态，可以在计算机支持软件（选配）的监视 TOOL 里确认。

4.3.6 多任务的使用例

介绍多任务的执行范例。

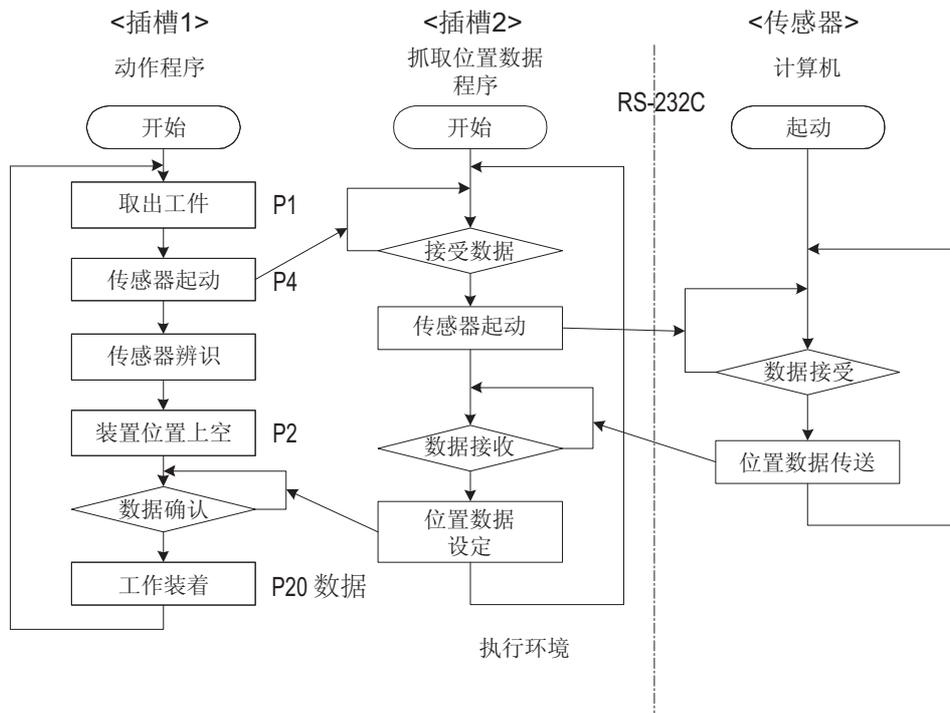
(1) 机器人的作业内容

机器人程序分为「动作程序」和「位置数据取得程序」2种。

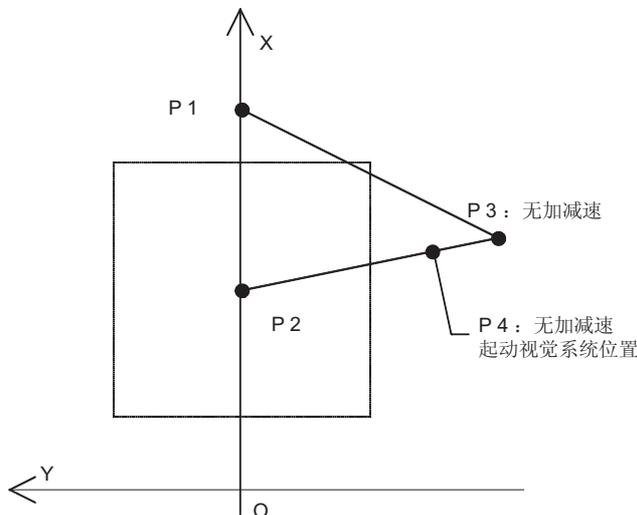
「动作程序」在任务插槽1执行，「位置数据取得程序」在任务插槽2执行。

机器人在动作途中，往感应器输出起动指令的话，会介由位置数据取得程序向计算机做出数据要求。计算机根据数据要求，将位置数据传送到机器人。机器人介由数据取得程序，取得补偿数据。

< 处理流程 >



- P1: 取得工件位置 (吸着定时器 Dly 0.05)
- P2: 放置工件位置 (解除定时器 Dly 0.05)
- P3: 视觉系统前位置 (不停留通过点 Cnt)
- P4: 视觉系统快门位置 (不停留通过点 Cnt)
- P_01: 视觉系统补偿数据
- P20: 将视觉系统补偿数据的位置加在 P2 上 (相对运算)



(2) 到多任务为止的步骤

■步骤 1: 程序的制作

<1> 动作程序 (程序名称: 1)

```

1 Cnt 1 ' 轨迹连续动作为有效
2 Mov P2,10 ' 往 P2 上空 +10mm 移动
3 Mov P1,10 ' 往 P1 上空 +10mm 移动
4 Mov P1 ' 往 P1 工件抓取位置
5 M_Out (10) =0 ' 抓取工件
6 Dly 0.05 ' 定时器 0.05 秒
7 Mov P1,10 ' 往 P1 上空 +10mm 移动
8 Mov P3 ' 往视觉系统前位置 P3 移动
9 Spd 500 ' 设定直线速度 500mm/ 秒
10 Mvs P4 ' 在 P4 通过时起动视觉抓取
11 M_02#=0 ' 在连锁变量 (M_01=1/M_02=0)
12 M_01#=1 ' 在环境处理时, 数据取得开始
13 Mvs P2,10 ' 往 P2 上空 +10mm 移动
14 *L2: If M_02#=0 Then GoTo *L2 ' 等待到变成连锁变量 M_02=1 为止
15 P20=P2*P_01 ' P20 加上视觉补偿 P_01, 往 P20 上空 +10mm 移动
16 Mov P20,10 ' 往 P20 上空 +10mm 移动
17 Mov P20 ' 往 P20 工件放置位置
18 M_Out (10) =1 ' 放置工件
19 Dly 0.05 ' 定时器 0.05 秒
20 Mov P20,10 ' 往 P20 上空 +10mm 移动
21 Cnt 0 ' 轨迹连续移动为无效
22 End ' 1 循环结束

```

<2> 位置数据取得程序 (程序名: 2)

```

1 *L1: If M_01#=0 Then GoTo *L1 ' 等待到变成全局变量 M_01=1 为止
2 Open "COM1:" As #1 ' 以太网回路开启
3 Dly M_03# ' 假设处理定时器 (0.05 秒)
4 Print #1,"SENS" ' 将字符串 "SENS" 传送到以太网 (视觉系统侧)
5 Input #1,M1,M2,M3 ' 等待视觉补偿值 (相对数据) 的取得
6 P_01.X=M1 ' 代入 Δ X 坐标
7 P_01.Y=M2 ' 代入 Δ Y 坐标
8 P_01.Z=0.0 '
9 P_01.A=0.0 '
10 P_01.B=0.0 '
11 P_01.C=Rad (M3) ' 代入 Δ C 坐标
12 Close ' 以太网回路关闭
13 M_01#=0 ' 全局变量 M_01=0
14 M_02#=1 ' 全局变量 M_02=1
15 End ' 处理结束

```

■步骤 2: 任务插槽参数的设定

插槽参数设定如下所示。

参数	程序名	运行模式	运行形态	执行行数
SLT1	1	REP	START	1
SLT2	2	REP	START	1

■步骤 3: 任务插槽参数的反映

为了使参数 (SLT 1, SLT 2) 变成有效, 请关闭电源后再开启。

■步骤 4: 起动

操作面板的起动开始程序 1 和程序 2 的运行。

4.3.7 关于程序容量

在机器人程序处理领域，分为储存、编辑、执行 3 种。各领域的容量，请参照「表 4-4 程序各领域的容量」。

(1) 程序储存领域

为储存程序的领域。标准可以储存合计 940K 位的程序。

(2) 程序编辑领域

在程序的编辑、单步执行动作确认时使用的领域。程序编辑领域为 179K 位，相当于 1 个程序的最大值。

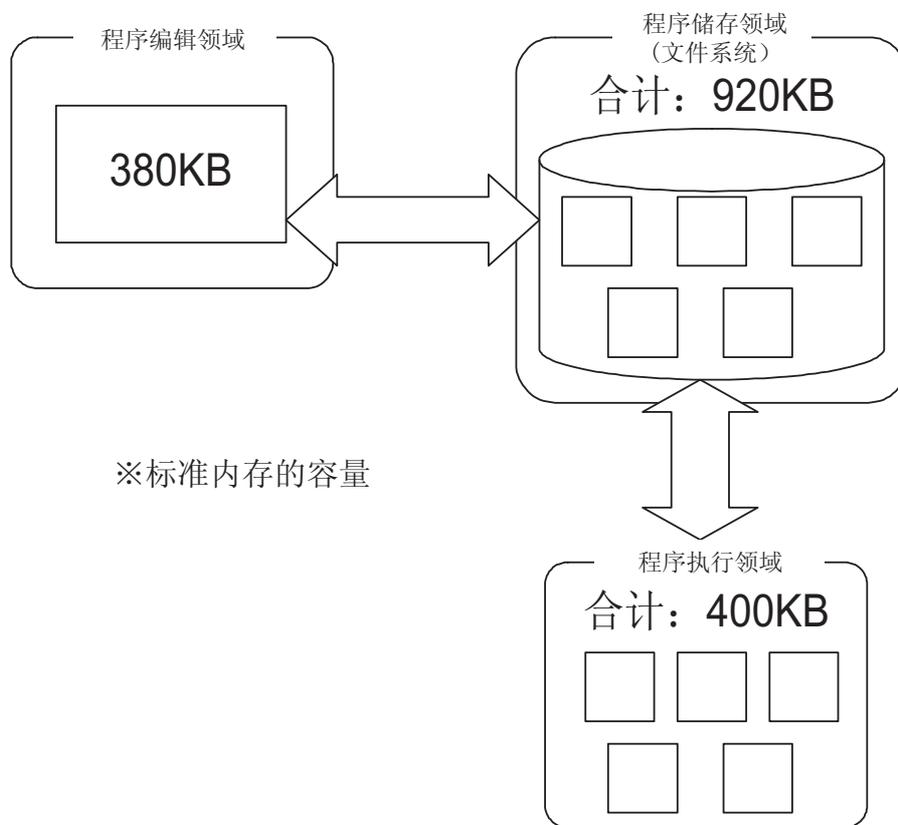
(3) 程序执行领域

程序执行领域为使程序自动运行时使用的领域。程序执行领域为 400K 位。依据使用者基本程序、多任务、XRun、GallP 指令，在同时执行领域承载的程序合计容量，必须为 400K 位以下。

表 4-4：程序各领域的容量

名称	容量
	标准内存
(1) 程序存储领域	920K 位元
(2) 程序编辑领域	380K 位
(3) 程序执行领域	400K 位

各程序的容量可以在示教单元及计算机支持软件的程序管理画面做确认。



4.4 MELFA-BASIC V 的详细规格

在此说明 MELFA-BASICV 的格式及构成等的相关文法，以及各指令的详细功能。
下列说明指令的构成要素。

(1) 程序名制作

程序名请使用 12 个文字以内。但是，操作面板的显示最多到 4 个文字，因此建议，制作程序在 **4 文字以内**。可使用的文字如下所示。

区分	可使用的文字
英文字	ABCDEFGHIJKLMNOPQRSTUVWXYZ (只能使用大写。登录小写的话，会有无法正常执行的情况)
数字	0 1 2 3 4 5 6 7 8 9 制作

文字在 4 个文字以上的情况，无法从操作面板选择。另外，从外部输入信号选择的情况下，请使用号码制作。**Call P** 指令作为子程序执行的情况下，可以使用 4 个文字以上的英文名字。但是一样无法从操作面板做选择。

(2) 指令

指令的构成范例

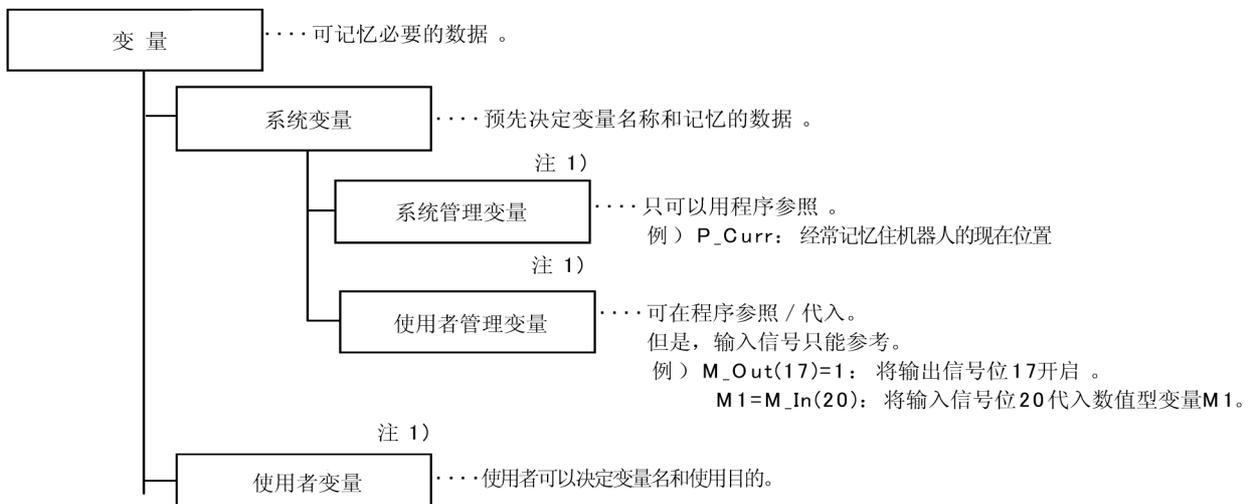
1 Mov P1 Wth M_Out(17)=1

① ② ③ ④

- ①步号：决定程序的执行顺序的号码。以号码的上升顺序执行。
- ②指令：指定机器人的动作及作业的指令。
- ③数据：每个指令所需的变量及数值等的的数据。
- ④附随句：根据需要附加对应机器人的作业时指定。

(3) 变量

程序可以用下列所示种类的变量。



位置型变量	····· 记忆机器人的直交坐标。变量名用“P”开始。 例) Mov P1 : 往以变量名P1记忆住的位置移动。
关节型变量	····· 记忆机器人的关节角度。变量名用“J”开始。 例) Mov J1 : 往以变量名J1记忆住的位置移动。
数值型变量	····· 记忆数值(整数、实数等)。变量名用“M”开始。 例) M1=1 : 变量名M1代入值1。
文字型变量	····· 记忆字符串。在变量名的最后加上“\$”。 例) C1\$="ERROR" : 在变量名C1\$代入称为“ERROR”的字符串。

4.4.1 文

文是构成程序的最小单位，由指令和那个所给予的数据所构成。

例) Mov P1
 指令 数据
 指令

4.4.2 附随文

只限定在移动指令，依据附随句，可以串连指令。
 依据此点，可以执行和移动指令并列的部份指令。

例) Mov P1 Wth M Out(17)=1
 指令 随附句 指令

详细内容请参照第 279 页的「Wth (With)」或第 279 页的「Wthlf (With lf)」及各移动指令 (Mov (Move), Mva (Move Arch), Mvs (Move S), Mvr (Move R), Mvr2 (Move R2), Mvr3 (Move R3))。

4.4.3 行

行是由行号码和 1 个指令所构成的。但是，使用附随句的情况下，指令会变成 2 个。一行的长度最多 240 个文字。(不包含行末文字。)

◇◆◇ 1 行只有 1 个指令◆◆◇

依照一般的 BASIC，复数的指令以分号做区分，无法在一行记述。

4.4.4 行号码

MELFA-BASIC V 不采用以往在 MELFA-BASIC 所使用的行号码。取而代之的是，是使用行号码。

看起来行号码相同，但是，开头会自动附上 1 个号码。行号码会变成从 1 开始到 3 2 7 6 7 为止的整数。在 MELFA-BASIC V，无法指定 GoTo、GoSub 等的跳转步号（行号码）。

指定的话，会变成句法报警。请使用卷标取代行号码。

4.4.5 标签

卷标是作为分支端的记号，可以独自附上的名称。

卷标在行号码的后面，在 * 加上英文大写、小写、数字构成，但是开头必须要用英文字，且以 16 个文字以内记述。可以定义 1 文字的卷标，会变成预约文等文字 (F、J、L、P、S、T 等为预约文)，建议使用 2 个文字以上，容易判断的标签名。标签名会直接照输入时大写、小写文字登录。但是，大写、小写并无区别。

在机器人控制器登录后，读出时，会统一被登录的标签名。

※ 无法在卷标使用文字如下所示。

- 预定语 (命令、函数等)
- 以记号及数字开始的名字
- 在变量名、函数名中被使用的名字

例) 1 GoTo *LBL
 :
 8 *LBL

4.4.6 可以在程序内使用的文字种类

可以在程序内使用的文字如下所示。但是，在程序名、变量名及标签名的使用上有限制。
可使用的文字为○、无法使用的文字为×、有限制使用文字则以△表示。

表 4-5: 可使用的文字一览

区分	程序可使用的文字	程序名	变量名	标签名
英文字	ABCDEFGHIJKLMNOPQRSTUVWXYZ	○	○	○
	abcdefghijklmnopqrstuvwxyz	×	○	○
数字	0123456789	○	△注1)	△注2)
记号	" ' & () * + - . , / : ; = < > ? @ ` [\] ^ { } ~	×	×	×
	% & ! # \$	×	可以作为型指定记号	×
	_ (下划线)	×	△注3)	○
空白	空白文字	×	×	×

注1) 变量名的开头只能使用英文字。第2个文字以后可以使用数字。

注2) 程序中的卷标名的开头只能使用英文字。第2个文字以后可以使用数字。

注3) 第2个文字以后可以使用。在第2文字使用'_'(下划线)的变量会变成全局变量。

关于详细的程序名请参照第134页的「(1) 程序名制作」、变量名请参照第141页的「4.4.15 变量」卷标名请参照第135页的「4.4.5 标签」。

4.4.7 有特别意义的文字

(1) 大写、小写的区别

MELFA-BASIC V 也接受英文小写。

变量名、标签名可就照输入时的大写、小写登录。但是，大写小写并没有分别，将以往登录的变量名、标签名统一后接受。因此，在读取程序时，以往登录名称会被变换。

另外，指令和文字列数据时，就照原样被登录。

关于指令和系统状态变量名也一样，没有大写和小写的分别，程序读出时，会被登录系统变换名称。

(2) 下划线 (_)

在识别符号（变量的名称）的第 2 个文字加上 _（下划线）的话，变量会在程序间变成有效，作为全局变量识别。

详细内容请参照第 144 页的「4.4.22 全局变量」。

例) P_Curr、M_01、M_ABC

(3) 撇号 (')

将撇号以后的文字视为注释（指令）。又，加在文的前头的话，会变成用 Rem 文取代。

例) 100 Mov P1 ' TORU ; TORU 会为成注释。

150 ' BUHIN TORIDASHI ; 和 150 Rem BUHIN TORIDASHI 相同。

(4) 星号 (*)

附在变成分支处的标签名前头。

例) 200 *KAKUNIN

(5) 逗号 (,)

作为指令的参数（自变量）及添加字的区分。

例) P1=(200, 150,....)

(6) 句号 (.)

从小数点、位置变量、关节变量等的复数数据，只取得成份数据情况下使用。

例) M1=P2.X ; 只将位置变量 P2 的 X 坐标代入数值变量 M1。

(7) 空白

包含在字符串常数及注释文的东西会被视为有文字意义。在行号码及指令的后面，数据的划分等为必要。在「4.13 指令的详细说明」的【书式】，在必要的场所用 "□" 表示空白。

4.4.8 数据类型

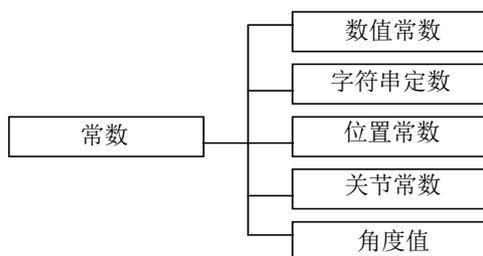
在 MELFA-BASIC V 里，数据的可分为数值、位置、关节、字符串等 4 种，将各别称为 [数据类型]。数值型分为实数型和整数型。在各数据类型有变量、常数。



例) 数值型 M1[数值变量]、1[数值常数] (整数)、1.5[数值常数] (实数)
 位置型 P1[位置变量]、(0, 0, 0, 0, 0, 0)(0, 0) [位置常数]
 关节型 J1[关节变量]、(0, 0, 0, 0, 0, 0)[关节常数]
 字符串型 C1\$[字符串变量]、"ABC"[字符串变量]

4.4.9 常数

常数分为 6 种：数值常数、字符串常数、位置常数、关节常数、角度值



4.4.10 数值常数

数值常数的结构如下所示。数值常数如下所示。

(1) 10 进制

例) 1、1.7、-10.5、+1.2E+5 (指数记载)

有效范围 -1.7976931348623157e+308 ~ 1.7976931348623157e+308

(2) 16 进制

例) &H0001、&HFFFF

有效范围 &H0000 ~ &HFFFF

(3) 2 进制

例) &B0010、&B1111

有效范围 &B0000000000000000 ~ &B1111111111111111

(4) 常数类型

可以依照在常数文字附加的记号，指定常数类型。

例) 10% (整数)、10000& (长精度整数)、1.0005! (单精度实数)、10.000000003# (双精度实数)

4.4.11 字符串常数

字符串常数是指用 (") 双引号记号圈起来的文字。

例) "ABCDEFGHJKLMN" "123"

◆◆◆字符串最多为 240 文字◆◆◆

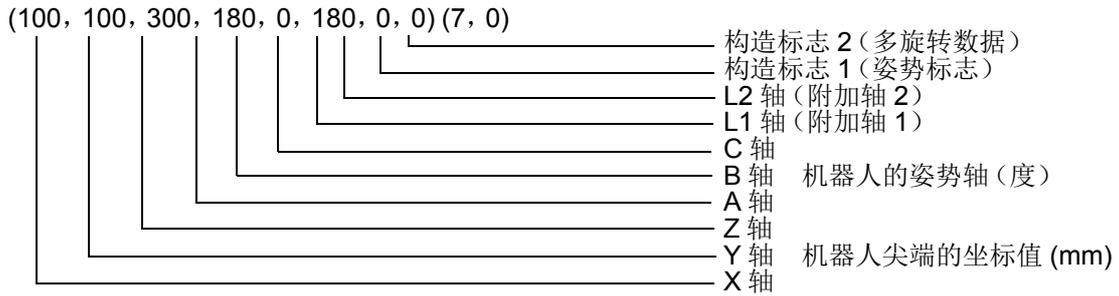
可输入的字符串长度，包含行号码、双引号在内，在 240 个文字以内。

将双引号本身包含在字符串的情况下，请连续输入 2 个双引号。

例如输入 AB"GD 字符串的情况下，请输入 "AB""GD"。

4.4.12 位置常数

位置常数的构造如下所示。在位置常数内无法记述变量。



例)

- P1=(300, 100, 400, 180, 0, 180, 0, 0)(7, 0)
- P2=(0, 0, -5, 0, 0, 0)(0, 0) [无行走轴数据的情况]
- P3=(100, 200, 300, 0, 0, 90)(4, 0) [4 轴的水平多关节机器人的情况]

(1) 坐标 · 姿势 · 附加轴资料的形式和意义

【形式】 X, Y, Z, A, B, C, L1, L2

【意义】 X, Y, Z: 坐标数据。机器人尖端的位置放置在直交坐标 (单位为 mm)

A, B, C: 姿势数据。表示姿势角度 (单位为 deg) 注 1)

L1, L2: 附加轴数据。附加轴 1、附加轴 2 的坐标值 (单位为 mm 或 deg)

注 1) 示教单元、计算机支持软件的单位以 deg 表示, 但是, 在程序的代入及运算, 单位是以弧度运算。

(2) 构造标志数据的形式和意义

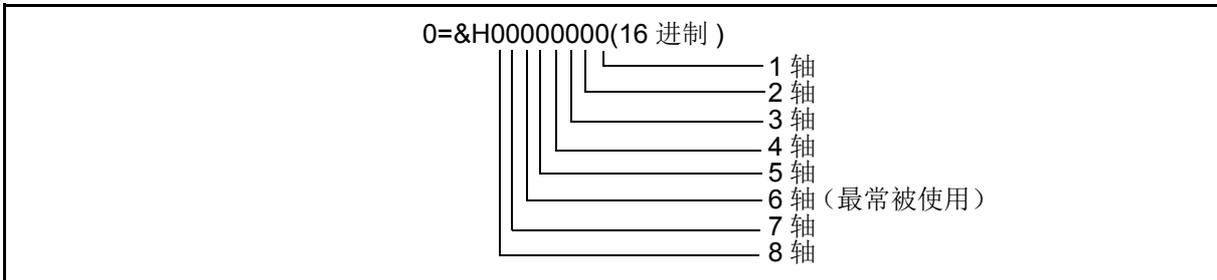
【形式】 FL1, FL2

【意味】 FL1: 姿势标志。表示在直交坐标的机器人手臂的姿势。



FL2: 多旋转数据。在直交坐标表示位置 (X Y Z) 和姿势 (A B C), 有相关各关节轴的旋转角度的情报。

初始值 =0 (范围从 0 开始至 +4294967295^111*4 个位元表示共有 8 个轴。每 1 轴 (-8~7) 有在计算机以 10 进制显示画面及每轴的旋转数以 16 进制显示画面。



多旋转数据的值

各轴的角度	-900	-540	-180	0	180	540	900
多旋转数据的值	...	-2 (E)	-1 (F)	0	1	2	...

在直交坐标的手轴尖端轴 (垂直多关节型机器人的 J6 轴), 即使旋转 1 周 (360 度), 也会显示一样的值。因此使用 FL2 来做区别。

◇◆◇关于轴数的指定◇◆◇

- ①没有必要全部记载 8 轴的坐标·姿势资料，但是，省略的情况下，省略的轴会被认为是未定义。
4 轴机器人 (X, Y, Z, C 轴构成) 的情况下，以 (X,Y,Z...C) 或 (X,Y,Z,0,0,C) 记述。
②省略全部的轴的情况下，像 (,) 这样，请至少加入一个", " (逗号)。

◇◆◇关于对位置成份数据的变量使用◇◆◇

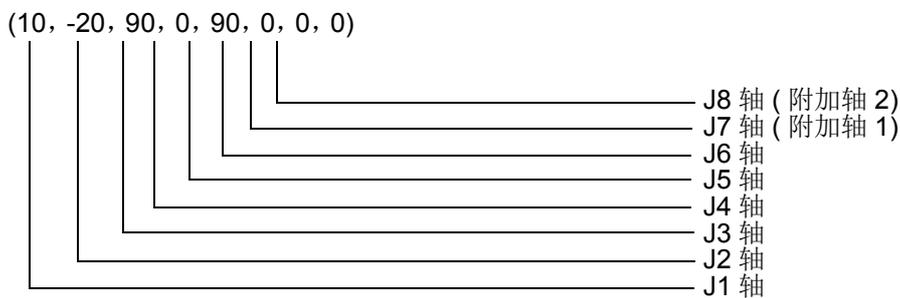
可以调用坐标·姿势·附加轴资料和构造 FLAG 的数据。
将位置常数构成位置成份数据里，无法包含变量。

◇◆◇构造标志数据的省略◇◆◇

将构造数据省略的时候，初始数据 ((7,0) 依机型的不同) 会被适用。

4.4.13 关节常数

关节常数的构造如下所示。在关节常数内无法记述变量。



例) 6 轴 机器人 J1=(0, 10, 80, 10, 90, 0)
6 轴 + 附加轴 J1=(0, 10, 80, 10, 90, 0, 10, 10)
5 轴 机器人 J1=(0, 10, 80, 0, 90, 0)
5 轴 + 附加轴 J1=(0, 10, 80, 0, 90, 0, 10, 10)
4 轴 机器人 J1=(10, 20, 90, 0)
4 轴 + 附加轴 J1=(10, 20, 90, 0, , , 10, 10)

(1) 各轴数据的形式和意义

【形式】 J1,J2,J3,J4,J5,J6,J7,J8

【意义】 从 J1 到 J6: 机器人各轴数据 (单位为 mm 或 deg)

J7.J8: 附加轴数据，可以省略 (单位为 mm 或 deg, 依参数的设定。) 水平多关节机器人的 J3 轴为直动轴的情况下，单位并非角度，而是会变成 mm。

◇◆◇关于关节成份资料的变量的使用◇◆◇

各轴数据称为关节成份数据。
将位置常数构成位置成份数据里，无法包含变量。

4.4.14 角度值

角度值非不是指 "弧度", 而是在表现 "度" 时使用。

记述为 100Deg 的话, 会变成角度值, 可以在三角函数等的自变量等使用。

例) Sin(90Deg) 表示为 90 度正弦。

4.4.15 变量

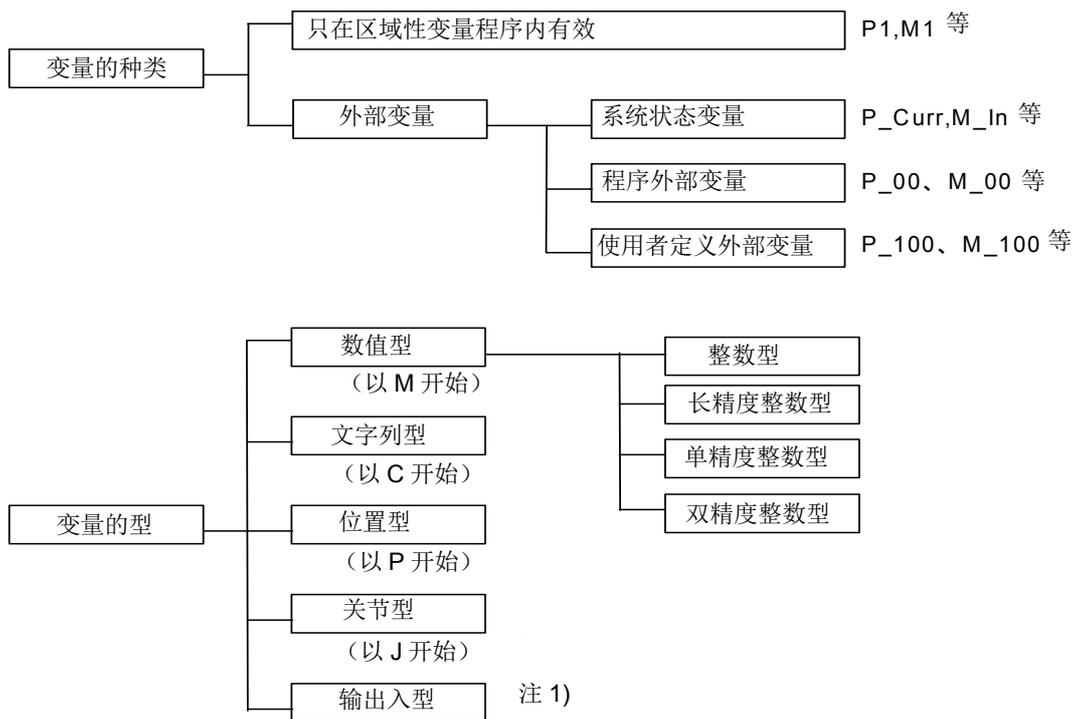
变量名的制作, 文字最大为 16。可以使用 1 个英文字变量名, 但是, 是处于预约语文字状态 (F, J, L, P, S, T 等), 因此建议使用 2 个文字以上容易判断的名称。

变量的类型分为数值型、字符串型、位置型、关节型、输出型。各别称为「变量的类型」。变量的类型决定于识别符号 (变量的名称) 的开头文字。

而且, 数值型又分为整数型、长精度整数型、单精度实型、双精度实数型。

此外, 作为数据的有效范围, 分为以下 2 种类。

- ①只在 1 个程序内有效的区域性变量 (Local Variable)
- ②在程序间间接有效的机器人 (系统) 状态变量、程序全局变量、使用者定义全局变量 (在变量名的第 2 文字附上 _ 的全局变量。请参照第 144 页的「4.4.22 全局变量」。)



注 1) 识别符号有以机器人 (系统) 状态变量决定的符号 (M_In, M_Out 等) 和以 Defio 指令用程序宣布的符号。

◇◆◇变量不会被初始化◇◆◇
变量在被生成时、程序下载或复位时, 不会被清 0。

4.4.16 数值变量

数值变量为使用 P, J, C 的文字以外的文字开始的变量。在 MELFA-BASICW 里, 大部份是以 M 为开头做说明。M 为 mathematics(数字)的开头文字。

```
例) M1=100
    M2!=-1.73E+10
    M3#=0.123
    ABC=1
```

1) 在数值型后置符号, 可以定义变量的类型。省略的情况下会变成单精度实数型。

表 4-6: 数值变量的数据范围

数值型后置符号	型	范围	備考
%	整数	-32768 ~ 32767	
&	长精度整数型	-2147483648 ~ 2147483647	
!	单精度实数型	-3.40282347e+38 ~ 3.40282347e+38	注) e 表示为 10 的次方。
#	双精度实数型	-1.7976931348623157e+308 ~ 1.7976931348623157e+308	

2) 在已登录完毕的变量后面, 无法加上数值型后置符号。请在最初制作新程序的记述时, 将符号写上。

3) 单精度 = 双精度的执行时, 值若超过的话, 会发生报警。

4.4.17 字符串变量

字符串变量以 "C"(或 "c") 开始, 最后要加上 "\$"。以 Def Char 指令定义的情况下, 用 "C"(或 "c") 以外的文字开始, 也可以加上变量名。

```
例) C1$="ABC"
    CS$=C1$
    Def Char MOJI
    MOJI="MOJIMOJI"
```

4.4.18 位置变量

字符串变量以 "P"(或 "p") 开始的变量。以 Def Pos 指令定义的情况下, 用 "P"(或 "p") 以外的文字开始, 也可以加上变量名。

可参照位置数据。在变量名的后面加上 "." 和成份名 "X" 等。

P1.X、P1.Y、P1.Z、P1.A、P1.B、P1.C、P1.L1、P1.L2 角度成份的 ABC 的单位为 (rad)。在度的变换时, 请使用 Deg 函数。例) P1=PORG

```
例) P1=PORG
    Dim P3(10)
    M1=P1.X      (单位 mm)
    M2=Deg(P1.A) (单位 DEG)
    Def Pos L10
    Mov L10
```

4.4.19 关节变量

关节变量以 "J"(或 "j") 开始的变量。以 Def Jnt 指令定义的情况下, 用 "J"(或 "j") 以外的文字开始, 也可以加上变量名。

可参照关节变量的成份数据。在变量名的后面加上 "." 和成份名 "J1" 等。

JDATA.J1、 JDATA.J2、 JDATA.J3、 JDATA.J4、
JDATA.J5、 JDATA.J6、 JDATA.J7、 JDATA.J8

成份数据的单位为 (rad)。在度变换时, 请使用 Deg 函数。

例) JSTART=(0, 0, 90, 0, 90, 0, 0, 0)

```
JDATA=JSTART
Dim J3(10)
M1=JDATA.J1      (单位 RAD)
M2=Deg(JDATA.J2) (单位 DEG)
Def Jnt K10
Mov K10
```

4.4.20 输出输入变量

输出输入变量有下列种类。此为在机器人 (系统) 状态变量中预先准备好的。

轮出入变量名	说明
M_In	输入信号的位参照用
M_Inb	输入信号的位 (8 位) 参照用
M_Inw	输入信号的单字 (16 位) 参照用
M_Out	输出信号的位参照 / 代入用
M_Outb	输出信号的位 (8 位) 参照 0 代入用
M_Outw	输出信号的单字 (16 位) 参照 0 代入用

请参照机器人 (系统) 状态变量的第 314 页的「M_In/M_Inb/M_In8/M_Inw/M_In16」、第 324 页的「M_Out/M_Outb/M_Out8/M_Outw/M_Out16」及第 306 页的「M_DIn/M_DOut」。

4.4.21 数组变量

数值变量、字符串变量、位置变量、关节变量可以配列使用。在变量的添字部份指定配列要素。配列要素用 Dim 指令记述。最大可配列三维。

例) 数组变量的定义例子

```
Dim M1(10)      单精度实数型
Dim M2%(10)    整数型
Dim M3&(10)    长精度整数型
Dim M4!(10)    单精度实数型
Dim M5#(10)    双精度实数型
Dim P1(20)
Dim J1(5)
Dim ABC(10, 10, 10)
```

配列的添字从 1 开始。

在输入输出信号的机器人 (系统) 状态变量中, 只有输入输出信号用的变量 (M_In、M_Out 等), 添字会变成从 Q 开始。

依据空的内存的容量, 确保决定领域。

4.4.22 全局变量

所谓全局变量，是在识别符号（变量的名称）的第 2 个文字加上 "_"（下划线）的变量，其值在程序间为有效。（使用者定义全局变量必须要登录在基本程序里）。因此，在多任务的程序间，全局变量和第 138 页的「4.4.8 数据类型」中所示一样的有数值、位置、关节、文字 4 种类型。全局变量有下列的 3 类。

表 4-7：全局变量的种类

全局变量	说明	例
程序全局变量	事先决定名称。	P_01, M_01, P_100(1) 等
使用者定义全局变量	使用者可以任意决定名称。 在使用者程序利用 Def Pos, Def Jnt, Def Char, Def Inte/ Long/Float/Double 宣布。	P_GENTEN, M_MACHI
机器人（系统） 状态变量	机器人（系统）状态变量以系统管理，事先对于名称的 内容（用途）做决定。	M_In, M_Out, P_Curr, M_PI 等

4.4.23 程序全局变量

在控制器，最初开始使用的程序全局变量显示在表 4-8。

如表所示，变量的名称已决定但是其用途是在使用者决定。

表 4-8：程序全局变量

数据类型	变量名	个数	备注
位置	P_00 ~ P_39	40	
位置数组（要素数 10）	P_100() ~ P_109()	10	数组的要素请使用 1 维。
关节	J_00 ~ J_39	40	
关节数组（要素数 10）	J_100() ~ J_109()	10	数组的要素请使用 1 维。
数值	M_00 ~ M_39	40	变量的类型为双精度实数。
数值数组（要素数 10）	M_100() ~ M_109()	10	数组的要素请使用 1 维。变量的类型为双精度实数。
字符串	C_00 ~ C_39	40	
字符串（要素数 10）	C_100() ~ C_109()	10	数组的要素请使用 1 维。

4.4.24 使用者定义全局变量

在只有上述的程序全局变量却不足的情况下，想要定义附有独立名称的变量时，可以利用使用者基本程序，以使用者定义将程序全局变量。

使用者定义全局变量的步骤
1) 首先将使用者基本程序制作。在变量的第 2 个文字加上 “_”。
2) 在参数「PRGUSR」设定程序名后，重新开启电源。
3) 利用使用者定义全局变量，制作平时的程序。

- (1) 使用者基本程序^{注)}中的 DLM 文里，依照在识别符号的第 2 个文字里加上 “_” (底线)，此变量会变成使用者定义全局变量。
- (2) 不需要执行使用者基本程序。
- (3) 使用者基本程序中，请只记述变量的声明行。
- (4) 使用者基本程序里定义数组变量，作为全局变量使用的情况下，即使是使用中的程序侧，也要再次的在 Dim 指令 单做配列声明。区域性变量 (只在程序内有效的变量) 则没有必要再声明。

例) 使用者定义全局变量的使用范例

主要程序 (程序名 1) 侧

1 Dim P_200(10)	' 全局变量的再声明
2 Dim M_200(10)	' 全局变量的再声明
3 Mov P_100(1)	
4 If M_200(1)=1 Then Hlt	
5 M1=1	' 区域性变量

使用者基本程序 (程序名 100) 侧

1 Def Pos P_900, P_901, P_902, P_903	
2 Dim P_200(10)	' 使用的程序侧也需要再度定义。
3 Def Inte M_100	
4 Dim M_200(10)	' 使用的程序侧也需要再度定义。

参数名	值
PRGUSR	100

4.4.25 使用者基本程序的制作

注)

◆◆◆何谓使用者基本程序?◆◆◆

使用者基本程序是指使用使用者定义全局变量的情况下，为了定义那个变量所使用，但是，没有必要执行本程序。只登录将声明文记录的程序所制作的参数「PRGUSR」。

将参数变更后，需要再一次启动电源。

◆◆◆用计算机支持软件登录新规使用者基本程序的方法◆◆◆

最初只能从计算机支持软件将往机器人控制器的指令写入，接着只写入位置数据。

使用者基本程序，和普通的程序同样的可以用示教单元、计算机支持软件的其中一种制作，但是，使用计算机支持软件制作的情况下，请依下列步骤执行。

- 1) 将在使用者基本里所制作的程序，储存在计算机里。
- 2) 从计算机支持软件的程序编辑里启动程序管理。
- 3) 在程序管理的传送处，指定以 1) 制作的程序，将传送处视为机器人，执行「复制」操作。此时，去除「位置变量」的确认，只在「指令」确认的状态执行。
- 4) 复制完成后，再执行 3) 的操作。这回将「指令」的确认去除，只在「位置变量」确认的状态下执行。
- 5) 以程序管理将程序删除再登录的情况下，也请先将使用者基本程序写入机器人控制器。

4.5 机器人的坐标系说明

4.5.1 关于机器人的坐标系

机器人的坐标系有如下 4 种。

①世界坐标系：原点为 ※1

表明机器人的现在位置^{注 1)}的基准坐标系。

②基本坐标系：原点为 ※2（机器人底面的 J1 轴旋转中心轴）

以机器人的基座安装面为基准确定的坐标系。将从世界坐标系观察到的机器人的安装中心位置（基本变换数据）设定到参数：MEXBS，或通过执行 Base 指令进行设定。

初始设定时，基本变换数据为 0，因此世界坐标系和基本坐标系一致。

③机械接口坐标系：原点为 ※3（工具安装面上的 J6 轴旋转中心轴）

以机器人的机械 I/F 为基准确定的坐标系。

④工具坐标系：原点为 ※4

由机器人的机械 I/F 上安装的工具（抓手）确定的坐标系。它与机械 I/F 坐标系之间的关系取决于工具数据（参数：MEXTL 的设定、或 Tool 指令的执行）。

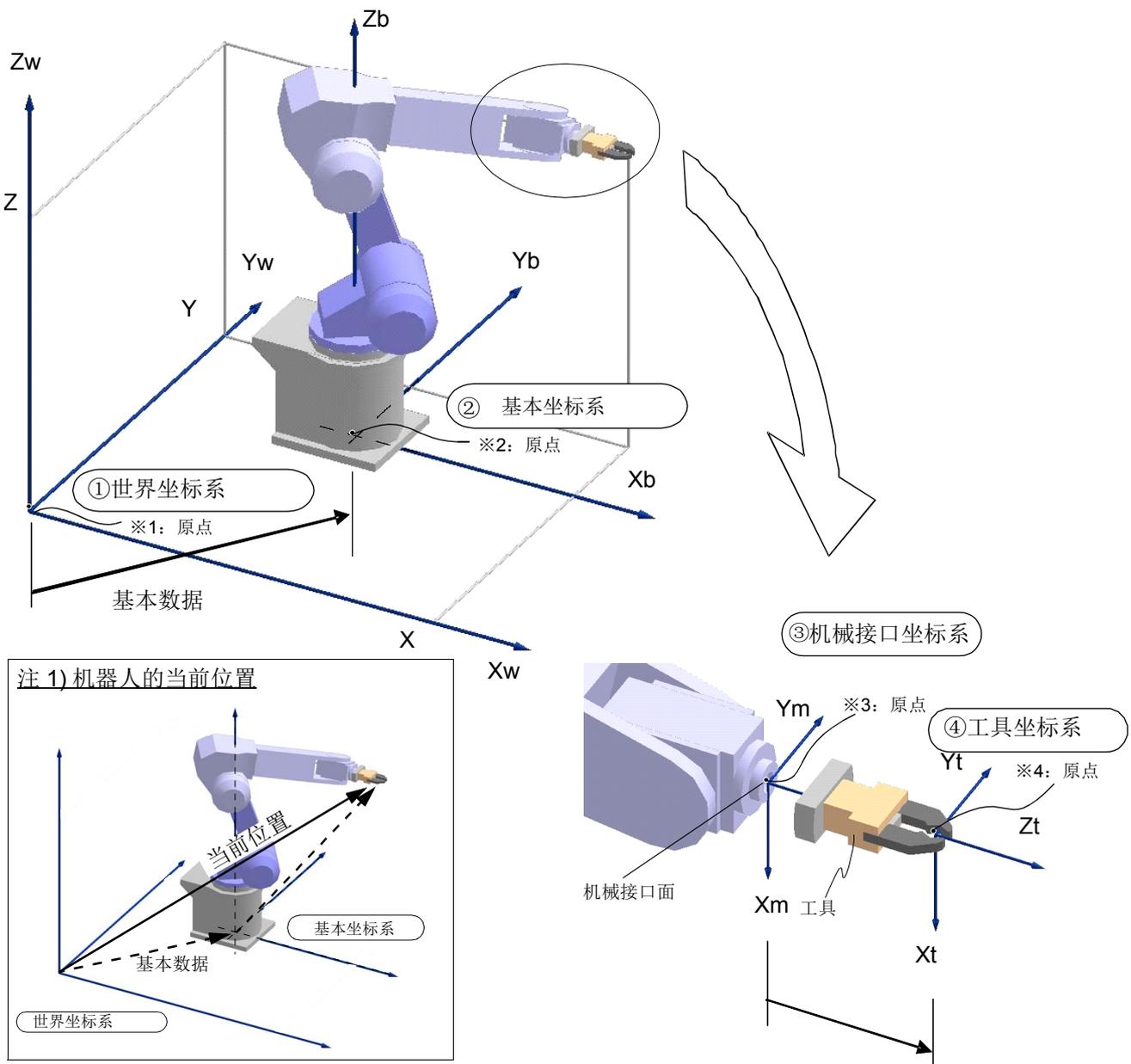


图 4-3: 机器人的坐标系

4.5.2 关于基本变换

可根据需要将世界坐标系移动到工作台的基准位置或工件的基准位置。
 使用该功能时，机器人的当前位置将作为工作台或工件基准的位置进行处理，因此相对于该基准，机器人的动作位置关系相同的作业组存在多个时，无需对作业组的所有动作一一示教，只需变更世界坐标系，即可进行相同作业（动作）。该世界坐标系的变更即被称为基本变换，通过对参数：**MEXBS**的基本变换数据（坐标值）的设定、或执行 **Base** 指令进行。

要设定的基本变换数据为，从新设的世界坐标系观察时的基本坐标系的原点位置数据。因此，当使用机器人的当前位置进行设定时（**Fram** 函数等），要对位置数据进行逆变换并进行设定。（例：**Base Inv(P1)**）
 使用 **Base** 指令指定工件坐标系参数（**WK1CORD** ~ **WK8CORD**）时，会在内部处理中实施逆变换，因此，用户无需实施逆变换处理。（例：**Base1** ~ **Base8**）

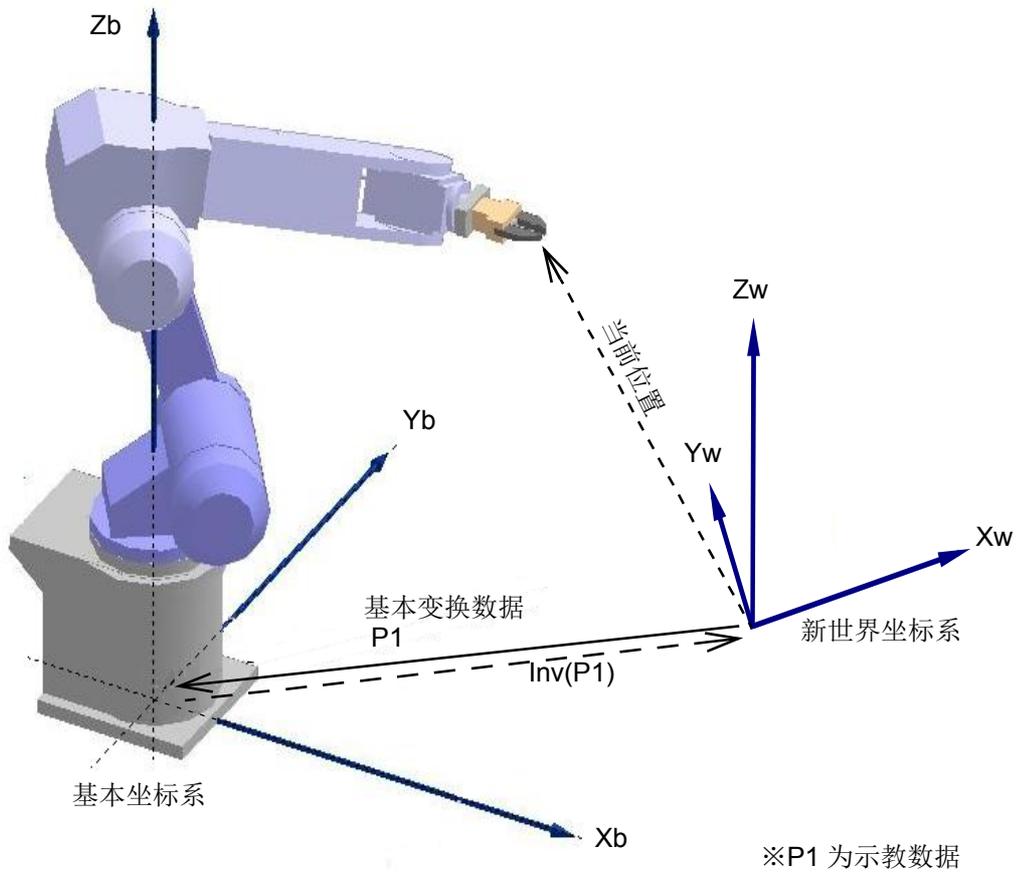


图 4-4：基本变换

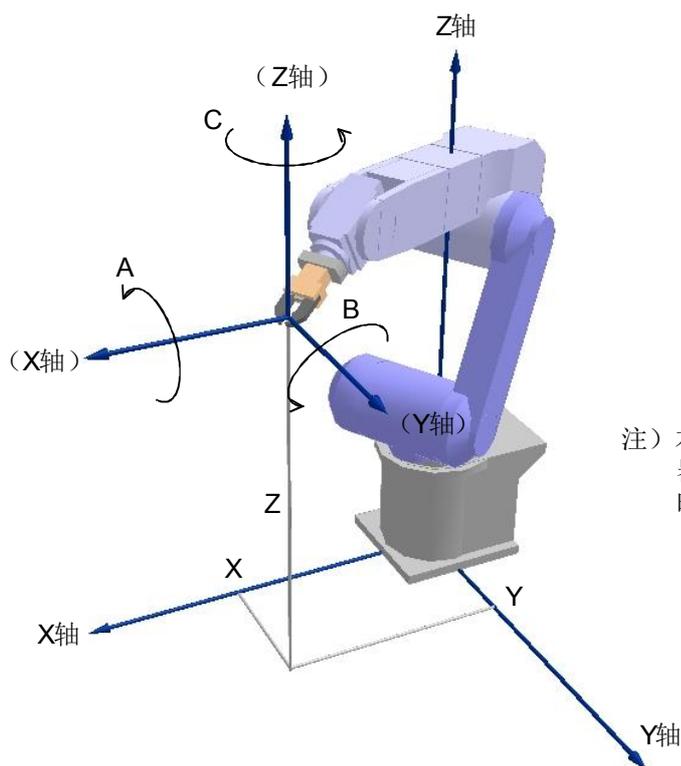
注意

实施基本转换后，机器人的当前位置变为以新设定的世界坐标系为基准的值。移动指令中指定的目的位置也作为新设定的世界坐标系内的位置处理。
 因此，之前的示教数据可能无法直接使用。示教时的坐标系需要和新设定的坐标系相同。
 如果坐标系不一致，则机器人会移动至预想不到的位置，从而造成财产损失或人身事故。
 要使用基本转换功能时，请对要转换的基本坐标系和示教位置的关系进行妥善管理，使机器人正确动作，更有效地利用基本转换功能。

4.5.3 关于位置数据

机器人的位置数据是指抓手前端（未进行工具设定时为机械 I/F 中心）的位置（X，Y，Z）与姿势（A，B，C）的 6 个数据以及构造标志的合并数据。
各数据的值均以世界坐标系为基准。

- 【含义】 X, Y, Z：坐标数据。世界坐标系中机器人的抓手前端的位置（单位为 mm）
姿势数据。表示姿势的角度（单位为 deg）
- A → 绕 X 轴旋转的旋转角
 - B → 绕 Y 轴旋转的旋转角
 - C → 绕 Z 轴旋转的旋转角



注）本图是假设尚未设定基本数据，即机器人的世界坐标系和基本坐标系一致时而绘制的。

姿势角的定义：

A, B, C 表明抓手前端（未进行工具设定时为法兰中心）的坐标系的姿势，它们分别表示世界坐标系的 X 轴、Y 轴、Z 轴旋转的旋转角度。相对各坐标轴的 + 方向，右螺纹的旋转方向为 + 方向。此外，已确定旋转的顺序，即按照 Z 轴旋转→Y 轴旋转→X 轴旋转的顺序旋转并进行计算（控制）。

4.5.4 关于TOOL坐标系（机械I/F坐标系）

将机器人的控制点设为机器人上安装的抓手前端时，需要设定工具数据。工具数据是以设定于法兰上的机械 I/F 坐标系为基准，对工具前端位置进行定义的数据。因此，下面首先对机械 I/F 坐标系进行说明。

为了更容易理解 TOOL 坐标系，这里我们将以垂直 6 轴机器人为例，对 TOOL 坐标系进行说明，关于其他机种（水平关节型机器人等），请参照第 432 页的「5.6 关于标准 TOOL 坐标」。

(1) 机械 I/F 坐标系

如图 4-5 所示、以法兰中心为原点的坐标系被称为机械 I/F 坐标系。机械 I/F 坐标系的 X 轴用 X_m 、Y 轴用 Y_m 、Z 轴用 Z_m 表示。

Z_m 为穿过法兰中心，垂直于法兰面的轴。从法兰面向外的方向为 + 方向。 X_m 、 Y_m 被设定在法兰面内。法兰中心和定位销孔的连线即为 X_m 轴。 X_m 轴的 + 方向为从中心到销孔相反方向。

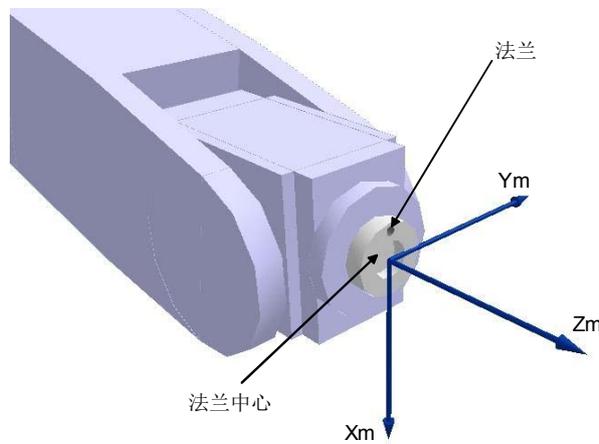


图 4-5: 机械 I/F 坐标系

法兰转动，机械 I/F 坐标系亦随之转动。（图 4-6）

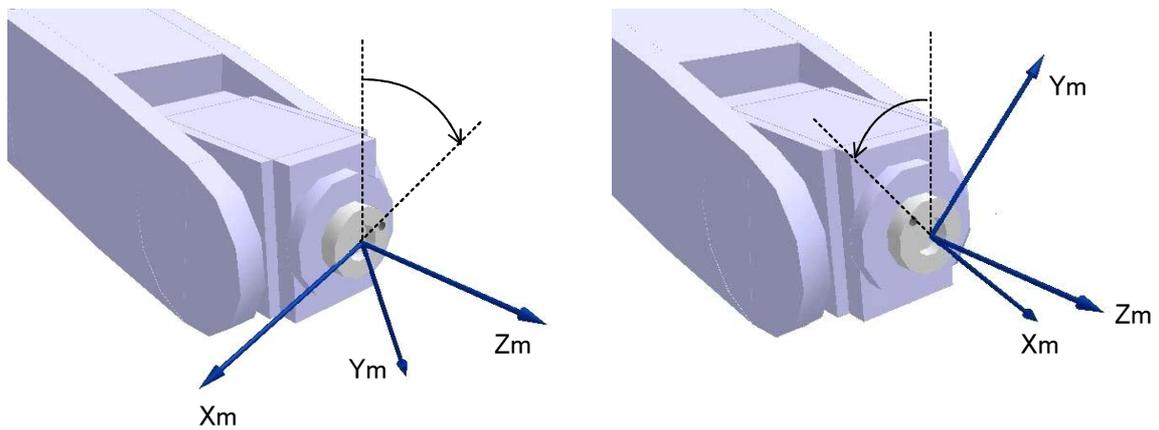


图 4-6: 法兰转动和机械 I/F 坐标系

(2) TOOL 坐标系

TOOL 坐标系是定义在抓手前端（抓手控制点）的坐标系，它将机械 I/F 坐标系的原点移到了抓手前端（控制点），并增加了任意旋转的成分。

TOOL 坐标系的 X 轴用 X_t 、Y 轴用 Y_t 、Z 轴用 Z_t 表示。

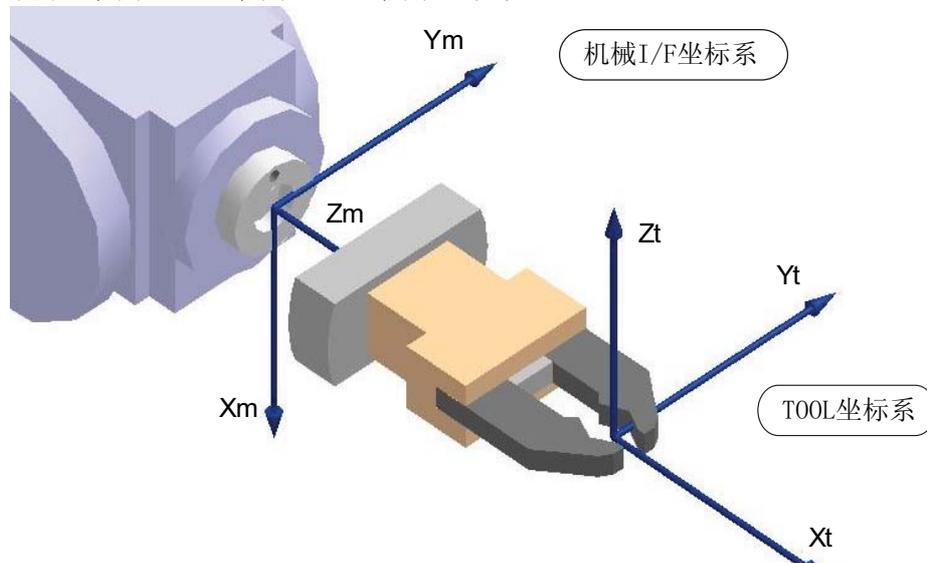


图 4-7：机械 I/F 坐标系与 TOOL 坐标系

TOOL 数据具有与位置数据相同的成分。

X, Y, Z：移动量。从机械 I/F 坐标系原点到 TOOL 坐标系原点的移动量。（单位为 mm）

A, B, C：坐标轴的旋转角度。（单位为 deg）

- A → 绕 X 轴旋转的旋转角
- B → 绕 Y 轴旋转的旋转角
- C → 绕 Z 轴旋转的旋转角

(3) TOOL 坐标系的使用效果

1) JOG 动作、示教操作时

以工具 JOG 模式使机器人动作时，可根据抓手面的方向使其动作。针对对象工件调整抓手姿势、变更夹持工件的姿势变得更为简单。

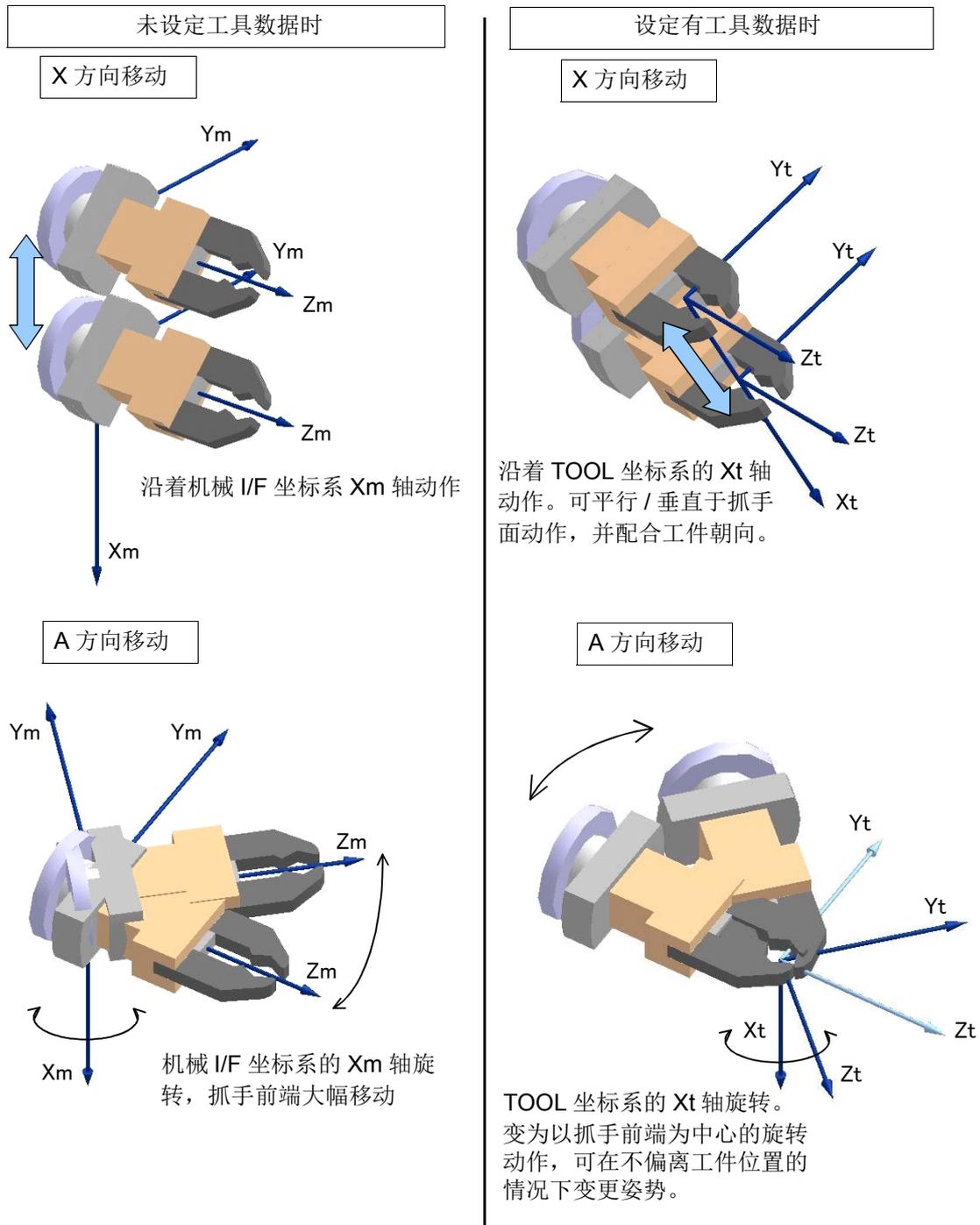


图 4-8: 基于工具数据有无的工具 JOG 动作

2) 自动运行时

使用移动指令可指定靠近距离 / 脱离距离，可简单方便地设定工件取出、搬运时的动作。靠近 / 脱离方向为 TOOL 坐标系的 Z 方向。

如图 4-9 所示、移动到工件搬运位置上空 50mm 的位置时，记述为 `Mov P1,50`。

表示为在 P1 的 Z 轴方向（TOOL 坐标系）上向 +50mm 的位置移动。

根据工件的朝向或动作状态设定 TOOL 坐标系的 Z 方向，可提高操作性。

在图 4-9 的例子中，为了使抓手侧横以进行工件的插入 / 拔出操作，TOOL 坐标系的 Z 轴方向与工件的朝向要一致。

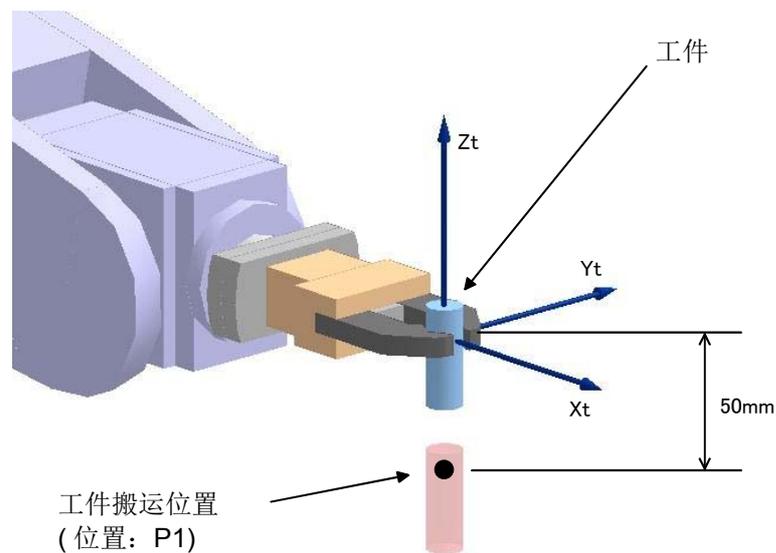


图 4-9: 靠近 / 脱离动作

在进行工件的相位对准等工件的姿势变更时，如设定有工具数据，则操作起来会更加方便。

如图 4-10 所示、绕工件的中心轴旋转，对准相位时，记述为 `Mov P1*(0,0,0,0,45)`。

「`*(0,0,0,0,45)`」中，「*」表示 TOOL 坐标系上的位置运算，让 (X,Y,Z,A,B,C) 的 C 旋转 45°。C 为绕 Z 轴的旋转，反映在机器人的动作上为，P1 绕 Z 轴（绕 TOOL 坐标系的 Zt 轴旋转）旋转 45°。

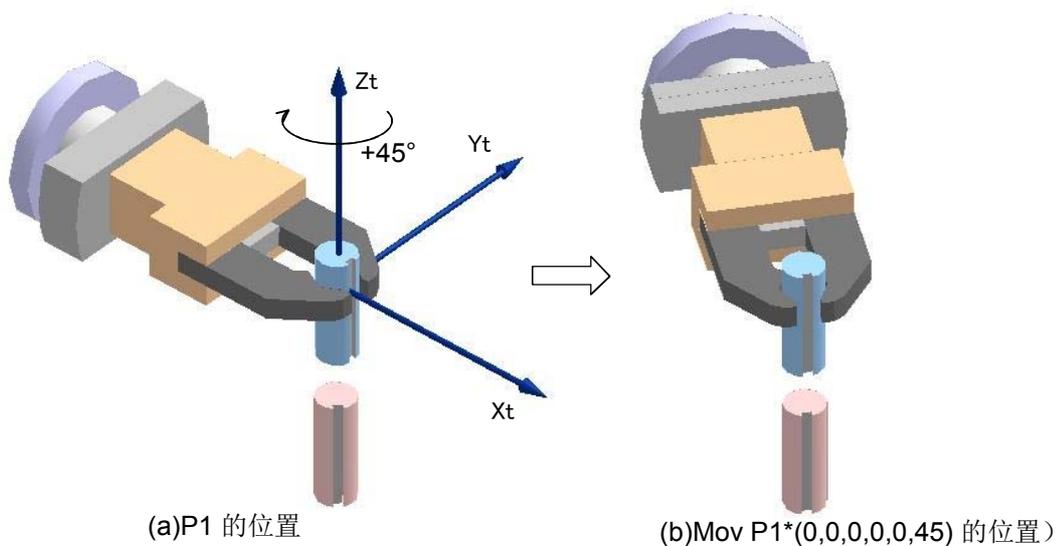


图 4-10: TOOL 坐标系上的旋转动作

4.6 机器人（系统）状态变量

预先准备的机器人的状态变量如表 4-9 所示。如表所示，预先决定变量的名称及用途。使用这些变量的话，可以调查机器人的状态，也可以使状态变化。

表 4-9: 机器人（系统）状态变量

No	变量名称	配列要素数 ^{注1)}	内容	属性 ^{注2)}	数据的型，单位	页
1	P_Curr	机器号码(1~3)	现在位置(直交)	R	位置型	357
2	J_Curr	机器号码(1~3)	现在位置(关节)	R	关节型	292
3	J_ECurr	机器号码(1~3)	现在检出器脉冲位置	R	关节型	296
4	J_Fbc	机器号码(1~3)	以原来从伺服来的回馈值制作的关节位置	R	关节型	297
5	J_AmpFbc	机器号码(1~3)	电流的回馈值	R	关节型	297
6	P_Fbc	机器号码(1~3)	以原来从伺服来的回馈之制作的关节位置	R	位置型	360
7	M_Fbd	机器号码(1~3)	指令位置和回馈位置的距离	R	整数型, mm	312
8	M_CmpDst	机器号码(1~3)	伺服弹性功能执行时的指令值和实际位置的偏移量	R	单精度实数型, mm	302
9	M_CmpLmt	机器号码(1~3)	伺服弹性指令值超过限制而变成报警时, 使用插入处理, 从报警状态复归时使用。	R	整数型	303
10	P_Tool	机器号码(1~3)	现在以指定的TOOL变换数据	R	位置型	361
11	P_Base	机器号码(1~3)	现在以指定的基本变换数据	R	位置型	353
12	P_NTool	机器号码(1~3)	系统初期值(TOOL变换数据)	R	位置型	361
13	P_NBase	机器号码(1~3)	系统初期值(TOOL变换数据)	R	位置型	353
14	M_Tool	机器号码(1~3)	TOOL号码(1~4)	RW	整数型	338
15	J_ColMxl	机器号码(1~3)	推断转矩和实际转矩的差异	R	关节型、%	293
16	M_ColSts	机器号码(1~3)	冲突检知状态(1: 冲突中、0: 其它)	R	整数型	304
17	P_ColDir	机器号码(1~3)	冲突时的动作方向	R	位置型	355
18	P_CordR	机器号码(1~3)	通过干涉回避功能, 返回从通用坐标到本机器人基本坐标为止的位置。	R	位置型	356
19	P_CurrR	机器号码(1~3)	通过干涉回避功能, 返回从通用坐标观察到的本机器人的现在位置。	R	位置型	358
20	M_Cavsts	机器号码(1~3)	通过干涉回避功能, 返回检测到干涉的机器人的CPU号码。	RW	整数型	301
21	P_CavDir	机器号码(1~3)	通过干涉回避功能, 返回检测到干涉时的机器人的动作方向。	R	位置型	354
22	M_OPOvrd	无	操作面板的速度比例(0~100%)	R	整数型、%	318
23	M_Ovrd	插槽号码(1~32)	现在指定的程序内速度比例(0~100%)	R	整数型、%	318
24	M_JOvrd	插槽号码(1~32)	现在指定的关节速度比例(0~100%)	R	整数型、%	318
25	M_NOvrd	插槽号码(1~32)	系统初始值(M_OVRD的初始值)(%)	R	单精度实数型、%	318
26	M_NJovrd	插槽号码(1~32)	系统初始值(M_JOVRD的初始值)(%)	R	单精度实数型、%	318
27	M_Wupov	机器号码(1~3)	暖机运行速度比例(50~100%)	R	单精度实数型、%	345
28	M_Wuprt	机器号码(1~3)	到解除暖机状态为止的时间(秒)		单精度实数型、秒	346
29	M_Wupst	机器号码(1~3)	到解除暖机状态为止的时间(秒)		单精度实数型、秒	347
30	M_Ratio	插槽号码(1~32)	到现在动作中的目的位置为止的到达率(%)	R	整数型、%	329
31	M_RDst	插槽号码(1~32)	现在动作中的剩余距离(只考虑X,Y,Z的3次元 mm)	R	单精度实数型, mm	330
32	M_Spd	插槽号码(1~32)	现在指定的速度(只在直线/圆弧插补时有效)	R	单精度实数型、mm/s	333
33	M_NSpd	插槽号码(1~32)	系统初始值(M_SPD的初始值)(mm/s)	R	单精度实数型、mm/s	333
34	M_RSpd	插槽号码(1~32)	现在的指令速度(mm/s)	R	单精度实数型、mm/s	333
35	M_Acl	插槽号码(1~32)	现在指定的加速率(%)	R	单精度实数型、%	298
36	M_DAcL	插槽号码(1~32)	现在指定的减速率(%)	R	单精度实数型、%	298
37	M_NAcL	插槽号码(1~32)	系统初始值(M_ACL的初始值)(%)	R	单精度实数型、%	298
38	M_NDAcL	插槽号码(1~32)	系统初始值(M_DACL的初始值)(%)	R	单精度实数型、%	298
39	M_AclSts	插槽号码(1~32)	现在的加减速状态 0= 停止中 1= 加速中 2 = 定速中 3 = 减速中	R	整数型	298
40	M_SetAdl	轴数(1~8)	指定各轴的加减速时间比例。(%)	RW	单精度实数型、%	331
41	M_LdFact	轴数(1~8)	各轴的伺服电机的负载率(%)	R	单精度实数型、%	320
42	M_Run	插槽号码(1~32)	运行中状态(1: 运行中 0: 没有运行中)	R	整数型	330

No	变量名称	配列要素数 ^{注1)}	内容	属性 ^{注2)}	数据的型, 单位	页
43	M_Wai	插槽号码 (1 ~ 32)	中断中状态(1: 中断中 0: 没有中断中)	R	整数型	344
44	M_Psa	插槽号码 (1 ~ 32)	返回在指定的插槽中程序选择的可能与否。(1: 可选择, 0: 中断中时 不可选择)	R	整数型	328
45	M_Cys	插槽号码 (1 ~ 32)	循环运行状态 (1: 循环运行中, 0: 非循环运行中)	R	整数型	305
46	M_Cstp	无	循环停止动作状态 (1: 循环停止中 0: 非循环停止中)	R	整数型	305
47	C_Prg	插槽号码 (1 ~ 32)	执行程序名	R	字符串型	290
48	M_Line	插槽号码 (1 ~ 32)	现在执行中的行号码	R	整数型	319
49	M_SkipCq	插槽号码 (1 ~ 32)	执行包含最后执行的SKIP指令在内的行, 其执行结果, 进行SKIP的情况为1、不进行SKIP的情况则为0	R	整数型	332
50	M_BrkCq	无	BREAK指令结果 (1: BREAK、0: 无)	R	整数型	300
51	M_Err	无	报警中 (1: 报警发生中 0: 非报警发生中)	R	整数型	310
52	M_ErrLvl	无	读取报警等级 • S/W版本R1d之前(SQ系列)/S1d之前(SD系列)报警 无/警告/低等级/高等级=0/1/2/3 • S/W版本R1d之后(SQ系列)/S1d之后(SD系列)报警 无/警告/低等级/高等级/警告1/低等级1/高等级1=0/1/2/3/4/5/6	R	整数型	310
53	M_Errno	无	读取报警号码。	R	整数型	310
54	M_Svo	机器号码 (1 ~ 3)	伺服电机电源开启中 (1: 伺服电源开启中 0: 非伺服电源开启中)	R	整数型	336
55	M_Uar	机器号码 (1 ~ 3)	位数据 (1: 使用者定义领域内、0: 使用者定义领域外) (位0: 区域1 ~ 位15: 区域16)	R	整数型	339
56	M_Uar32	机器号码 (1 ~ 3)	位数据 (1: 使用者定义领域内、0: 使用者定义领域外) (位0: 区域1 ~ 位31: 区域32)	R	整数型	340
57	M_In	输入号码 (0 ~ 32767)	获取外部输入信号 (以位为单位)。通用位设备: 位信号输入0=off 1=on	R	整数型	314
58	M_Inb/ M_In8	输入号码 (0 ~ 32767)	获取外部输入信号 (以8位为单位)。通用位设备: 字节信号输入	R	整数型	314
59	M_Inw/ M_In16	输入号码 (0 ~ 32767)	获取外部输入信号 (以16位为单位)。通用位设备 : 字信号输入	R	整数型	314
60	M_In32	输入号码 (0 ~ 32767)	通过数值获取外部输入信号 (以32位为单位)。通用位设备: 双字信号输入	R	整数型	316
61	M_Out	输出号码 (0 ~ 32767)	输出外部输出信号 (以位为单位)。通用位设备: 位信号输出0=off 1=on	RW	整数型	324
62	M_Outb/ M_Out8	输出号码 (0 ~ 32767)	输出外部输出信号 (以8位为单位)。通用位设备: 字节信号输出	RW	整数型	324
63	M_Outw/ M_Out16	输出号码 (0 ~ 32767)	输出外部输出信号 (以16位为单位)。通用位设备: 字信号输出	RW	整数型	324
64	M_Out32	输出号码 (0 ~ 32767)	向外部输出信号 (以32位为单位) 中输出数值。通用位设备: 双字信号输出	RW	整数型	326
65	M_DIn	输入号码 (6000 ~)	CR7xx-Q系列无法使用。 (CC-Link的远程寄存器: 输入寄存器的读出)	R	整数型	306
66	M_DOut	输出号码 (6000 ~)	CR7xx-Q系列无法使用。 (CC-Link的远程寄存器: 输出寄存器的写入、读出)	RW	整数型	306
67	M_HndCq	输入号码 (1 ~ 8)	返回抓手确认输入信号。	R	整数型	313
68	P_Safe	机器号码 (1 ~ 3)	返回待避点位置。	R	位置型	361
69	J_Origin	机器号码 (1 ~ 3)	返回原点设定时的关节坐标数据。	R	关节型	297
70	M_Open	文件号码 (1 ~ 8)	指返回已指定的文件或通信端口的开启状态。	R	整数型	323
71	C_Mecha	机器号码 (1 ~ 3)	返回机器人的型号。	R	字符串型	290
72	C_Maker	无	显示厂商方面的情报。(字符串最多64个文字)	R	字符串型	289
73	C_User	无	返回参数"USERMSG"的内容。(字符串最多64个文字)	R	字符串型	291
74	C_Date	无	将现在的日期以"年/月/日"的形式显示。	R	字符串型	289
75	C_Time	无	将现在的时间以"小时: 分: 秒"的形式显示。	R	字符串型	291
76	M_BTime	无	返回电池剩余容量时间。(小时)	R	整数型、时间	300

No	变量名称	配列要素数 ^{注1)}	内容	属性 ^{注2)}	数据的型, 单位	页
77	M_Timer	定时器数 (1 ~ 8)	通常计时。也可以设定值。 [ms]依据程序的使用, 可以测定正确的节拍时间。	RW	单精度实数型	337
78	P_Zero	无	位置变量的要素(X,Y,Z,A,B,C,FL1,FL2)全部变为0的变量	R	位置型	365
79	M_PI	无	圆周率(3.1415...)	R	双精度实数型	328
80	M_Exp	无	自然对数的底(2.71828...)	R	双精度实数型	312
81	M_G	无	重力常数(9.80665)	R	双精度实数型	313
82	M_On	无	平时设定为1。	R	整数型	322
83	M_Off	无	平时设定为0。	R	整数型	322
84	M_Mode	无	操作面板的按键开关的状态 MANUAL/AUTOMATIC (操作面板)/AUTOMATIC (外部机器)=(1/2/3)	R	整数型	322
85	M_SplPno	机械号码 (1 ~ 3)	在样条插补动作中, 返回最新的已通过的路径点号码。	R	整数型	334
86	M_SplVar	机械号码 (1 ~ 3)	在样条插补动作中, 返回设定为路径点的任意数值。此外, 到再设定值为止的期间, 可保持任意的值。	RW	整数型	335
87	P_WkCord	工件坐标号码 (1 ~ 8)	当前设定的工件坐标的读取、工件坐标的设定	RW	位置型	364
88	M_ESpd	机械号码 (1 ~ 3)	在Ex-T控制/Ex-T样条插补动作中, 返回当前的Ex-T坐标的通过速度。	R	单精度实数型、mm/s	311
89	P_ECord	机械号码 (1 ~ 3)	在Ex-T控制/Ex-T样条插补动作中, 返回使用中的Ex-T坐标系原点的值。	R	位置型	359

注 1) 机器号码..... 1 ~ 3 指定对应多任务处理功能的机器号码。
 插槽号码..... 1 ~ 32 指定对应多任务功能的插槽号码。
 输入号码..... 0 ~ 32767 (理论值), 指定输入信号的位号码。
 输出号码..... 0 ~ 32767 (理论值), 指定输出信号的位号码。
 注 2) R..... 0 ~ 32767 (理论值), 指定输出信号的位号码。
 RW..... 可读取、写入。

4.6.1 逻辑数

表示比较、输入的结果等。

以整数评价，不为 0 的话是真，为 0 的话是假。在代入时，真的情况给予 1。在表 4-10 中，显示使用逻辑数的处理。

表 4-10: 逻辑数的真假对照值

在逻辑数的 1 显示	在逻辑数的 0 显示
<ul style="list-style-type: none"> • 比较运算的结果 (真的情况) • 逻辑运算的结果 (真的情况) • 开关启动 • 输出信号的开启 • 抓手开 (在抓手流过电流) • 设定插入等的允许 / 有效 	<ul style="list-style-type: none"> • 比较运算的结果 (假的情况) • 逻辑运算的结果 (假的情况) • 开关关闭 • 输出信号的关闭 • 抓手开 (在抓手不流过电流) • 设定插入等的禁止 / 无效

4.7 函数

所谓函数，是针对所给予的自变量，将进行某执行运算的结果，以数值型或字符串型返回的数，有预先编入的内建函数和用使用者定义的使用者定义函数。

(1) 使用者定义函数

以 Def FN 文定义。

例) Def FNMADD(MA, MB)=MA+MB

..... 定义求得 2 个值的合的函数 FNMADD。

函数名用 FN 开始，在第 3 个文字显示数据的型，记述识别文字 (C: 字符串、M: 数值、P: 位置、J: 函数)，且指定 8 个文字以内。

(2) 内建函数

内建函数一览如表 4-11 所示。

表 4-11: 内建函数一览

区分	函数名(书式)	功能	页	结果
数值函数	Abs (< 算式 >)	给予绝对值	367	数值
	CInt (< 算式 >)	将算式的小数部位四舍五入且变换为整数	374	
	Deg (< 算式: 弧度 >)	将角度单位 弧度 (rad) 变换为度 (deg)	377	
	Exp (< 算式 >)	给予算式的指数函数的值	378	
	Fix (< 算式 >)	给予整数部	379	
	Int (< 算式 >)	给予算式的值不能超过最大的整数	381	
	Len (< 文字列式 >)	给予文字串的长度	383	
	Ln (< 算式 >)	给予自然对数	384	
	Log (< 算式 >)	给予常用对数	384	
	Max (< 算式 >...)	从任意的个数的自变量中求得最大值	385	
	Min (< 算式 >...)	从任意的个数的自变量中求得最小值	386	
	Rad (< 算式: 度 >)	角度单位 将度 (deg) 变换为弧度 (rad)	390	
	Sgn (< 算式 >)	调查算式的符号	398	
	Sqr (< 算式 >)	给予平方根	402	
	Strpos(< 字符串式 >, < 字符串式 >)	求得在第 1 自变量字符串内, 第 2 自变量字符串的位置	402	
	Rnd (< 算式 >)	产生随机数	393	
	Asc(< 字符串式 >)	给予字符串式的最初的文字字符码	370	
	Cvi(< 字符串式 >)	将 2 文字的字符串式变换为整数	376	
	Cvs(< 字符串式 >)	将 2 文字的字符串变换为单精度实数	376	
	Cvd(< 字符串式 >)	将 8 文字的字符串变换为双精度实数	377	
	Val(< 字符串式 >)	字符串以数值变换	403	
	SplSpd(< 样条号码 >)	可从任意的样条文件中登录的路径点信息, 求出 MvSpl、EMvSpl 指令中不会发生错误的可指定的最大速度。		

区分	函数名 (书式)	功能	页	结果	
三角函数	ACos(< 算式 >)	给予反余弦 (Arc Cosine) 单位: 弧度 定义域: -1.0 ~ +1.0 值域: 0 ~ π	368	数值	
	ASin(< 算式 >)	给予反正弦 (Arc Cosine) 单位: 弧度 定义域: -1.0 ~ +1.0 值域: -π/2 ~ π/2	371		
	Atn(< 算式 >)	给予反正切 (Arc Tangent) 单位: 弧度 定义域: 数值值域: -π/2 ~ +π/2	371		
	Atn2(< 算式 >, < 算式 >)	给予反正切 (Arc Tangent) 单位: 弧度 Θ=ATN2(Δ y, Δ x) 定义域: Δ y, Δ x 的其中一个不为 0 的数值 值域: -π ~ +π	371		
	Cos(< 算式 >)	给予余弦 (Cosine) 单位: 弧度 定义域: 数值的范围 值域: -1 ~ +1	375		
	Sin(< 算式 >)	给予正弦 (Sine) 单位: 弧度 定义域: 数值的范围 值域: -1 ~ +1	399		
	Tan(< 算式 >)	给予正切函数 (Tangent) 单位: 弧度 定义域: 数值的范围 值域: 数值的范围	403		
字符串函数	Bin\$(< 算式 >)	将算式的值变换为 2 进制字符串	372	字符串	
	Chr\$(< 算式 >)	将算式的字符码给予持有文字	374		
	Hex\$(< 算式 >)	将算式的值变换为 16 进制字符串	381		
	Left\$(< 字符串式 >, < 算式 >)	从第 1 自变量的字符串例左边开始, 求得用第 2 自变量所指定的字符串长度	383		
	Mid\$(< 字符串式 >, < 算式 >, < 算式 >)	第 1 自变量字符串中, 从第 2 自变量指定位置开始, 用第 3 自变量求得指定长度的字符串	385		
	Mirror\$(< 字符串式 >)	执行字符串的 2 进制的位镜像反转	386		
	Mki\$(< 算式 >)	将算式的值变换为 2 个文字的字符串	387		
	Mks\$(< 算式 >)	将算式的值变换为 4 个文字的字符串	387		
	Mkd\$(< 算式 >)	将算式的值变换为 8 个文字的字符串	388		
	Right\$(< 字符串式 >, < 算式 >)	从第 1 自变量的字符串例右边开始, 求得用第 2 自变量所指定的字符串长度	393		
	Str\$(< 算式 >)	将算式的值变换为 10 进制字符串	402		
	CkSum(< 字符串式 >, < 算式 >, < 算式 >)	制作字符串的检查和 (校检) 用第 1 自变量的字符串, 返回到加算从第 2 自变量的位置到第 3 自变量的位置的字符串结果的下一个位置	375		数值
	位置变量	Dist(< 位置 >, < 位置 >)	求得 2 点间的距离		378
Fram(< 位置 1>, < 位置 2>, < 位置 3>)		计算在 3 点指定的坐标。位置 1 为平面的原点、位置 2 为 +X 轴上的点、位置 3 为 +Y 轴方向的平面上的点, 从 3 个的 XYZ 的坐标开始, 返回到平面的原点位置和将姿势以返回值 (位置) 返回。以在机构构造无关系的 6 轴三次元运算。在 5 轴型机器人中 ABC 的姿势数据意思不同, 因此无法使用。	380		
Rdf1(< 位置 >, < 数值 >)		将指定的位置构造标志, 用文字数据返回。 自变量的 (数值) 0=R/L, 1=A/B, 2=F/N 会变成 Return	391	文字	
Setf1(< 位置 >, < 文字 >)		变更指定位置的构造标志。变更数据以文字指定。 (R/L/A/B/F/N)	394		
Rdf2(< 位置 >, < 数值 >)		将指定位置的多旋转数据以数值 (-2 ~ 1) 返回。 自变量 (数值) 返旋转号码 (1 ~ 8)。	392	数值	
Setf2(< 位置 >, < 数值 >, < 数值 >)		将指定位置的多旋转数据以数值 (-2 ~ 1) 变更。 式为左边变更的轴号码、右边为设常数值。	395		
Align(< 位置 >)		返回位置 1 的姿势轴 (A,B,C), 会取得最接近的直交姿势 (0, ±90, ±180) 值。 在 5 轴型机器人中 ABC 的姿势数据意思不同, 因此无法使用。	369		
Inv(< 位置 >)		求得逆矩阵。	382	位置	
PtoJ(< 位置 >)		将位置数据变换为关节数据。	390	关节	
JtoP(< 位置 >)		将关节数据变换为位置数据。	382	位置	

区分	函数名 (书式)	功能	页	结果
位置变量	Zone(<位置 1>,<位置 2>,<位置 3>)	位置 1 是确认是否正进入以位置 2 和位置 3 的 2 点构成的空间 (立方体) 范围外 =0、范围内 =1 关于没有确认或不存在的成份, 单位为 deg 的情况下, 位置 2 设定为 -360, 位置 3 设定为 360 单位为 mm 的情况下, 位置 2 设定为 -10000, 位置 3 设定为 10000	404	数值
	Zone2(<位置 1>,<位置 2>,<位置 3> 数值 1, 数值 2, 数值 3, 位置 4)	位置 1 是确认是否正进入以位置 2 和位置 3 的 2 点构成的空间 (圆筒体) 范围外 =0、范围内 =1 对像只有 XYZ 的坐标值, ABC 的姿势数据会被忽略。	405	数值
	Zone3(<位置 1>,<位置 2>,<位置 3>, <位置 4> 数值 1, 数值 2, 数值 3)	位置 1 是确认是否正进入以位置 2, 位置 3, 位置 4 的 3 点以及数值 1, 数值 2, 数值 3 所构成的空间 (立方体) 范围外 =0, 范围内 =1 对像只有 XYZ 的坐标值, ABC 的姿势数据会被忽略。	405	数值
	PosCq(<位置 >)	确认 <位置 > 是否在动作范围内。	389	数值
	PosMid(<位置 1>,<位置 2>, 数值 1, 数值 2)	运算 <位置 1> 和 <位置 2> 的中间位置。	389	位置
	CalArc(<位置 1>,<位置 2>,<位置 3> 数值 1, 数值 2, 数值 3, 位置 4)	输出从 <位置 1>, <位置 2>, <位置 3> 可做圆弧的情报。	373	数值
	SetJnt(<J1 轴 >,<J2 轴 >,<J3 轴 >,<J4 轴 > <J5 轴 >,<J6 轴 >,<J7 轴 >, <J8 轴 >)	设定往关节变量的值	396	关节
	SetPos(<X 轴 >,<Y 轴 >,<Z 轴 >,<A 轴 > <B 轴 >,<C 轴 >,<L1 轴 >, <L2 轴 >)	设定往位置变量的值	397	位置
	SplPos(<样条号码 >, <路径点号码 >、<帧转换 >)	取出登录在任意样条文件中的路径点数据。	400	位置
SplECord(<样条号码 >, <帧转换 >)	取出登录在任意样条文件中的 Ex-T 坐标系原点数据。	399	位置	

4.8 指令一览

下列为各指令的说明页一览。以使用频率高的顺序表示。

(1) 动作控制关连的指令

指令	说明	页
Mov (Move)	关节插补	232
Mvs (Move S)	直线插补	242
Mvr (Move R)	圆弧插补	236
Mvr2 (Move R2)	圆弧插补 2	238
Mvr3 (Move R3)	圆弧插补 3	240
Mvc (Move C)	圆插补	235
Mva (Move Arch)	圆弧移动插补	233
MvSpl (Move Spline)	样条插补	244
Mxt	通过来自外部的指令进行的直接动作	249
MvTune	指定与用途最匹配的动作模式	247
Ovrd	全体的速度指定	255
Spd (Speed)	直线, 圆弧插补动作时的速度指定	268
JOvrd (J 速度比例)	关节插补动作时的速度指定	228
Cnt (Continuous)	指定连续轨迹模式	184
Accel	指定加减速度	164
ColChk (Col Check)	冲突检知	186

指令	说明	页
CavChk On	干涉回避	173
Cmp Jnt (Comp Joint)	在关节坐标指定弹性伺服	176
Cmp Pos	在直交坐标指定弹性伺服	178
Cmp Tool	在 TOOL 坐标指定弹性伺服	180
Cmp Off	弹性伺服设定无效	182
CmpG (Composition Gain)	指定弹性伺服的增益	183
Oadl (Optimal Acceleration)	指定最佳加减速	250
LoadSet (Load Set)	抓手附加条件指定	231
Prec (Precision)	高精度模式指定	257
Torq (Torque)	各轴的转矩指定	276
JRC (Joint Roll Change)	尖端轴可以多旋转	229
Fine	指定机器人的定位范围	216
Fine J	指定机器人的定位范围 (关节轴)	217
Fine P	指定机器人的定位范围 (直线距离)	218
Servo (Servo)	伺服电机电源的 ON/OFF	265
SpdOpt (Speed optimize)	直线插补动作时的速度调整	269
Wth (With)	往动作指令的附随指令	279
WthIf (With If)	往动作指令的附条件附随指令	279
EMvs (E Move S)	Ex-T 控制直线插补	209
EMvc (E Move C)	Ex-T 控制圆形插补	203
EMvr (E Move R)	Ex-T 控制圆弧插补	204
EMvr2 (E Move R2)	Ex-T 控制圆弧插补 2	206
EMvr3 (E Move R3)	Ex-T 控制圆弧插补 3	207
EMvSpl (E Move Spline)	Ex-T 样条插补	211

(2) 程序控制相关的指令

指令	说明	页
Rem(Remarks)	指令 (')	260
If...Then...Else...EndIf	条件分支	225
Select Case	可以做复数的分支	264
GoTo	跳转	223
GoSub(~ Return) (Go Subrouine)	子程序	222
Reset Err(Reset Error)	将报警复位 (禁止使用 Default)。	261
Call P	调用程序	170
FPrm	调用程序的自变量定义	220
Dly (Delay)	定时器	202
Hlt (Halt)	程序中断	223
End	程序转束	214
On... GoSub (ON Go Subroutine)	依据数值子程序跳转	252
On...GoTo	依据数值跳转	253
For ~ Next	循环	219
While ~ WEnd (While End)	附条件循环	278
Open	开启文件	254

指令	说明	页
Print	输出数据	258
Input	输入数据	227
Close	关闭文件	174
ColChk (Col Check)	冲突检知功能的有效 / 无效	186
HOpen/HClose (抓手 Open/Close)	抓手的开闭	224
Error	使用者报警	215
Skip(Skip)	动作中的跳转	267
Wait (Wait)	条件等待	277
Clr (Clear)	清零信号	175
SplWrt (Spline Write)	创建样条文件	272
SplFWrt (Spline Frame Write)	样条文件内帧转换信息改写	271

(3) 定义指令

指令	说明	页
Dim	数组变量声明	201
Def Plt (Define 码垛)	码垛声明	198
Plt (码垛)	码垛位置计算	256
Def Act	插入定义	190
Act	中断监控开始, 结束	165
Def Arch	定义弧型移动动作的弧形形状	192
Def Jnt (Define Joint)	定义关节位置变量	197
Def Pos (Define Position)	定义直交位置变量	200
Def Float/Def Double/Def Inte/Def Long	定义整数, 实数变量	195
Def Char (Define Character)	定义文字变量	193
Def IO (Define IO)	定义信号变量	196
Def FN (Define Function)	定义使用者函数	194
Title (Title)	程序标题的设定	274
Base	设定机器人基本位置	167
Tool (Tool)	设定 TOOL 长度	275
SetCalFrm (Set Calibration Frame)	帧转换所使用的坐标系的设定	266

(4) 多任务相关

指令	说明	页
XLoad (X Load)	下载其它任务插槽的程序	281
XRun (x Run)	执行其它任务插槽的程序	283
XStp (X Stop)	停止其它任务插槽的程序	284
XRst (X Reset)	将中断中的其它任务插槽的程序复位	282
XClr (X clear)	从指定的任务插槽程序解除程序下载	280
GetM (Get Mechanism)	取得机器控制权	221
RelM (Release Mechanism)	解放机器控制权	260
Priority	变更插槽任务的优先度	259
Reset Err(Reset Error)	报警复位 (禁止使用 Default)	261

(5) 其它

指令	说明	页
ChrSrch (Character Search)	从文字配列中检索字符串	172
GetPos (Get Position)	预留	-

4.9 运算符号

不执行数值的实数或整数的声明。取代的是依据运算的种类，强制的变换型。（参照表 4-12。）依照左项和右项型的组合，运算结果的型如下所示。

例)	左项	运算	右项	运算结果
	• 15	And	256	15
	(数值型)		(数值型)	(数值型)
	• P1	*	M1	P2
	(位置型)		(数值型)	(位置型)
	• M1	*	P1	记述报警
	(数值型)		(位置型)	

表 4-12: 依据运算型的数据变换表

左项型	运算	右项型				
		字符串	数值		位置	关节
			整数	实数		
字 串	代入 =	字符串	-	-	-	-
	加算 +	字符	-	-	-	-
	比较 (比较运算符号)	串整数	-	-	-	-
整 数	加算 +	-	整数	实数	-	-
	减算 -	-	整数	实数	-	-
	乘算 *	-	整数	实数	-	-
	除算 /	-	整数	实数	-	-
	整除除算 ¥	-	整数	整数	-	-
	剩余 Mod	-	整数	整数	-	-
	指数 ^	-	整数	实数	-	-
	代入 =	-	整数	整数	-	-
	比较 (比较运算符号)	-	整数	整数	-	-
逻辑 (逻辑运算符号)	-	整数	整数	-	-	
实 数	加算 +	-	实数	实数	-	-
	减算 -	-	实数	实数	-	-
	乘算 *	-	实数	实数	-	-
	除算 /	-	实数	实数	-	-
	整除除算 ¥	-	整数	整数	-	-
	剩余 Mod	-	整数	整数	-	-
	指数 ^	-	实数	实数	-	-
	代入 =	-	整数	整数	-	-
	比较 (比较运算符号)	-	整数	整数	-	-
逻辑 (逻辑运算符号)	-	整数	整数	-	-	
位 置	加算 +	-	-	-	位置	-
	减算 -	-	-	-	位置	-
	乘算 *	-	位置	位置	位置	-
	除算 /	-	位置	位置	位置	-
	整除除算 ¥	-	-	-	-	-
	剩余 Mod	-	-	-	-	-
	指数 ^	-	-	-	位置	-
	代入 =	-	-	-	-	-
	比较 (比较运算符号)	-	-	-	-	-
逻辑 (逻辑运算符号)	-	-	-	-	-	
关 节	加算 +	-	-	-	-	关节
	减算 -	-	-	-	-	关节
	乘算 *	-	关节	关节	-	-
	除算 /	-	关节	关节	-	-
	整除除算 ¥	-	-	-	-	-
	剩余 Mod	-	-	-	-	-
	指数 ^	-	-	-	-	关节
	代入 =	-	-	-	-	-
	比较 (比较运算符号)	-	-	-	-	-
逻辑 (逻辑运算符号)	-	-	-	-	-	
只有右项 (单项)	反转 - 否定 Not	-	整数 整数	整数 整数	位置 -	关节 -

反转: 符号反转 否定: 逻辑否定 代入: 代入运算 剩余: 剩余运算
比较: 比较运算 逻辑: 逻辑运算 (除了逻辑否定以外)

[注意]

- 以 '—' 记述部份的运算，没有做定义。
- 整数和整数的乘除算的结果，乘算的情况会变成整数型，除算的情况会变成实数型。
- 右项为 0 的除算情况下，（以 0 除算的情况），无法运算。
- 关于余数运算、逻辑运算（含否定），执行时，会将实数全部强制的变换为整数（四舍五入）后运算。

4.10 运算的优先级

式子中包含复数的运算符号的情况下，以下列表 4-13 中所示的优先级式进行运算。

表 4-13: 演运算的优先级

运算 (运算符号)	运算的种类	优先级
1) 括号 () 内的运算		高
2) 函数	函数	:
3) 指数 (^)	数值运算	:
4) 一元运算符号 (+,-)	数值运算	:
5) * /	数值运算	:
6) %	数值运算	:
7) Mod	数值运算	:
8) + -	数值运算	:
9) < >	逻辑运算	:
10) 比较运算符号 (=,<>,><,<=,<=>=>)	比较运算	:
11) Not	逻辑运算	:
12) And	逻辑运算	:
13) Or	逻辑运算	:
14) Xor	逻辑运算	低

4.11 程序的控制构造深度

组合程序的情况下，必须考虑控制构造的深度。

使用表 4-14 所示指令的话，程序的控制构造会变的更深。各指令的控制构造的深度有所限制，超过此限制的情况下执行时会发生报警。

表 4-14: 控制构造的深度限制

	段数	对应处理指令
程序内用户堆栈	16 段	▪ 循环控制 (For ~ Next, While ~ WEnd)
	8 段	▪ 调用函数 (CallP)
	800 段 (最大)	▪ 调用子程序 (GoSub) 依据 For ~ Next, While ~ WEnd, Call 指令的使用频率减少。

4.12 保留字

谓保留字是指系统里已被使用的字。程序上无法使用与保留字相同的名称。

指令、函数、系统状态变量（机器人状态变量）等会变成保留字。

4.13 指令的详细说明

4.13.1 记载项目的见解

- 【功能】 : 表示指令的功能。
- 【格式】 : 表示指令自变量的输入方法。
<>表示为自变量。
[]表示可省略。
□表示必须有空白。
- 【用语】 : 表示自变量的意义、范围等。
- 【例文】 : 表示参考例句。
- 【说明】 : 表示详细功能及注意事项。
- 【可使用的机器人机型】 : 表示可以使用的机器人机型。
- 【关连参数】 : 表示相关参数。
- 【关连系统状态变量】 : 表示相关系统变量。
- 【相关指令】 : 表示相关指令。

4.13.2 各指令的说明

下列为以英文顺序说明各指定。

Accel

【功能】

指定机器人动作时的加速度及减速度的比例（%）。
最佳加减速时也有效。

※ 最佳加减速时的加减速时间是以使用 `Oadl` 指令时算出最佳的时间，考虑 `M_SetAcl` 变量的值。

【格式】

```
Accel □ [<加速度比例 (%)>], [<减速度比例 (%)>]
      , [<Mva 指令上升时加速度比例 (%)>], [<Mva 指令上升时减速度比例 (%)>]
      , [<Mva 指令下降时加速度比例 (%)>], [<Mva 指令下降时减速度比例 (%)>]
```

【用語】

- <加速・减速度比例> 1~100(%)。速度从0开始到最高速度为止的加减速以比例指定。可用常数或变量记述・省略时以100, 100变成最高加减速速度。单位：[%]
- <上升时加减速比例> 指定Mva指令的圆弧动作上升时的加减速比例。省略时为100。可用常数或变量记述。
- <下降时加减速比例> 指定Mva指令的圆弧动作下降时的加减速比例。省略时为100。可用常数或变量记述。

【例文】

- | | | |
|--------------------------------|--|--|
| 1 Accel 50,100 | | ’ 指定重负载（标准加减速时间为0.2秒的情况下，加速时间会变成0.4秒、减速度时间会变成0.2秒） |
| 2 Mov P1 | | |
| 3 Accel 100,100 | | ’ 指定标准负载 |
| 4 Mov P2 | | |
| 5 Def Arch 1,10,10,25,25,1,0,0 | | |
| 6 Accel 100,100,20,20,20,20 | | ’ Mva 命指令动作时的上升、下降时的速度比例设为20° |
| 7 Mva P3,1 | | |

【说明】

- (1) 标准加减速时间，决定在每个机器人，设定对应的比例（%）。
系统初始值为100%、100%（最高加减速速度）
- (2) 执行本指令后变更加速比例，在程序复位时、及执行 `End` 文时，再设定系统初始值。
- (3) 可以记述100%以上，但是，依据机型，也有内部的上限为100%。超过100%的情况下，会影响机器人本体的寿命。此外，也会容易发生过度速度报警及过负载报警，因此在设定为100%以上时请特别注意。
- (4) `Cnt` 有效时的圆滑动作，依据加速度及动作速度，轨迹路径会有不同。另外，在一定速度执行圆滑动作的情况下，请将加速度和减速度视为同一值。在初始状态时，`Cnt` 会变成无效。
- (5) 在最佳加减速控制时 (`Oadl On`) 也会变成有效。

【相关指令】

[Oadl \(Optimal Acceleration\)](#)、[LoadSet \(Load Set\)](#)

【关连系统状态变量】

[M_Acl/M_DAcI/M_NAcI/M_NDAcI/M_AcISts](#)

【关连参数】

JADL

Act

【功能】

依据动作中的信号等，指定插入处理的许可 / 禁止。

【书式】

Act □ < 优先号码 >=<1/0/-1>

【用语】

<优先号码> 0: 插入全部许可或禁止。

1 ~ 8: 指定以Def Act文定义的插入优先号码。

请务必在优先号码Act指令的后面空出间隔记述。

当Act1继续记述时，会变成变量名的声明文。

<1/0/-1> 1: 插入许可的情况

0: 禁止使用时，变量也可以。

-1: <优先号码>为1~8时，对中断处理中进行解除。此外，中断监视也停止。

【例文】

①从 P1 往 P2 的移动中，输入信号 1 开启的话 (变成 1 的话)，到此信号变成 0 为止，会执行环路 (LOOP)

```
1 Def Act 1,M_In(1)=1 GoSub *INTR      ' 在插入 1 的条件下，分配输入信号 1。
2 Mov P1
3 Act 1=1                              ' 许可插入 1。
4 Mov P2
5 Act 1=0                              ' 禁止插入 1。
:
10 *INTR                                '
11 If M_In(1)=1 GoTo *INTR             ' M_IN(1) 信信号在变成 0 为止会执行环路 (LOOP)
12 Return 0                            '

```

②从 P1 往 P2 的移动中，输入信号 1 开启的话 (变成 1 的话)，动作中断且输出信号 10 开启

```
1 Def Act 1,M_In(1)=1 GoSub *INTR      ' 在插入 1 的条件下，分配输入信号 1。
2 Mov P1
3 Act 1=1                              ' 许可插入 1。
4 Mov P2
:
10 *INTR                                '
11 Act 1=0                              ' 禁止插入 1。
12 M_Out(10)=1                         ' 将输出信号 10 开启。
13 Return 1                            ' 发生插入，返回下一行。

```

③从 P1 向 P2 的移动中，输入信号 1 开启时 (变为 1 时) 动作中断，且当信号变为 0 之后再返回最初

```
1 *Start
:
11 Def Act1, M_In(1)=1 Goto *INTR2     ' 向中断 1 的条件中分配输入信号 1。
12 Mpv P1
13 Act 1=1                              ' 允许中断 1。
14 Mov P1
:
21 *INTR2
22 Act 1=0                              ' 将中断 1 设为禁止。
23 *Loop
24 If M_In(1) <> 0 Then Goto *Loop
25 Act 1=-1                             ' 对中断处理中进行解除。此外，中断监视也停止。
26 Goto *Start

```

【说明】

- (1) 优先号码 0 在程序开始运行时为“许可”的状态。将<优先号码>设定为 0 的话，即使优先号码 1 ~ 8 变成“许可”，也不会执行插入。
- (2) 优先号码 1 ~ 8 在程序运行开始时为“禁止”的状态。
- (3) 插入在下列的条件成立时进行。
 - <优先号码> 0 为“许可”
 - Def Act 文的条件成立
 - 将 Def Act 指定的<优先号码>以 Act 文变成有效时
- (4) 从插入处理开始的返回，是以 Return 0 或 Return 1 的其中一个记述，且指定插入处理结束后的返回地点。但是，从插入开始以 Return 1 返回到下一行的情况下，请禁止在插入处理内插入。不禁止插入，直接在插入条件成立的情况下，为了再次执行插入处理后返回到下一行，而不执行行，而进行跳越(SKIP)。
- (5) 即使在机器人插补动作中也中断的情况，会执行以 Def Act 文定义的插入处理。
- (6) 在插入处理中，在<优先号码>为“禁止”的状态执行。
在 Return 指令，回到从插入处理的话，会再次回到“许可”状态。
- (7) 至于通信插入 (Com) 会比以 Def Act 文定义的插入优先级变的更高。
- (8) 其它的插入的优先级关系如下所示。
Com > Act > Wth If(Wth)
- (9) 指定 Act 0=-1 时，发生 L3110 错误。
- (10) -1 的指定无法用于版本 R5d/S5d 以上。

【相关指令】

Def Act、Return

Base

【功能】

可以变更机器人现在位置控制基准的世界坐标系（移动及旋转）。有 2 种方法，一是直接指定基本变换数据，二是指定事先定义的世界坐标系的号码。

本功能会对机器人的示教数据或 JOG 动作产生较大影响。请在阅读「[4.5 机器人的坐标系说明](#)」之后，小心使用。

【格式】

Base □ < 基本变换数据 >	’ 直接指定基本变换数据
Base □ < 基本坐标号码 >	’ 通过基本坐标号码（工件坐标系号码）间接指定基本变换数据。

【用语】

- <基本变换数据> 以位置常数或位置变量指定基本变换数据。
要设定的值(坐标值)为从新设定的世界坐标系观察到的基本坐标系原点的位置数据。
- <基本坐标号码> 将系统初始值或参数(工件坐标系)中设定的值作为基本变换数据指定。
以数值型的常数或变量,在0~8的范围内指定。
0: 指定P_NBase(系统初始值)(因P_NBase=0, 0, 0, 0, 0, 0, 故清除基本变换)
1~8: 参数 工件坐标系 对应WK1CORD~WK8CORD
注)以实数或双精度实数指定时,小数点以下的值将被舍去。

【例文】

以基本变换数据指定

- | | |
|--------------------------|---|
| 1 Base (50,100,0,0,0,90) | ’ 以通过常数进行的变换数据设定新的世界坐标系 |
| 2 Mvs P1 | ’ 以新的世界坐标系向 P1 移动 |
| 3 Base P2 | ’ 以通过变量进行的变换数据设定新的世界坐标系 |
| 4 Mvs P1 | ’ 以新的世界坐标系向 P1 移动 |
| 5 Base 0 | ’ 将世界坐标系返回初始值。
(将 P_Nbase (系统初始值) 设定为基本变换数据) |

以基本坐标号码指定

- | | |
|----------|---|
| 1 Base 1 | ’ 将工件坐标系 1 (参数: WK1CORD) 作为新的世界坐标系 |
| 2 Mvs P1 | ’ 以新的世界坐标系向 P1 移动 |
| 3 Base 2 | ’ 将工件坐标系 2 (参数: WK2CORD) 作为新的世界坐标系 |
| 4 Mvs P1 | ’ 以新的世界坐标系向 P1 移动 |
| 5 Base 0 | ’ 将世界坐标系返回初始值。
(将 P_Nbase (系统初始值) 设定为基本变换数据) |

【说明】

(1) 要进行基本变换的值（坐标值）为从新设定的世界坐标系观察到的基本坐标系原点的位置数据。因此，使用机器人的现在位置，将通过 Fram 函数等定义的坐标值设定为基本变换数据时，对坐标值进行逆向变换后设定。（例：Base Inv(P1)）

此外，指定工件坐标系号码时，会自动在内部进行上述逆向变换处理。

位置数据的 X、Y、Z 成分表示从世界坐标系的原点开始到基本坐标系的原点为止的平行移动量。此外，

A、B、C 成分表示基本坐标系相对于机器人坐标系倾斜了多少。

X… 往 X 轴方向平行移动的距离

Y… 往 Y 轴方向平行移动的距离

Z… 往 Z 轴方向平行移动的距离

A… 围绕 X 轴的旋转角度

B… 围绕 Y 轴的旋转角度

C… 围绕 Z 轴的旋转角度

从世界坐标系的原点往正方向看，顺时针为正的旋转方向来获取 A、B、C。

(2) 构造标志内容没有意义。

(3) 通过本指令变更的基本坐标系会被存储在参数 MEXBS 中，控制器的电源关闭后也会被保持。

(4) 实施基本变换后，机器人的当前位置变为以新设定的世界坐标系或工件坐标系为基准的值。移动指令中指定的目的位置也作为新设定的世界坐标系或工件坐标系内的位置处理。

(5) 根据机器人的机型（本体的构造），基本变换数据的有效轴成分会有所不同。请参照第 432 页的「5.6 关于标准 TOOL 坐标」。

此外，与本指令相关的说明位置如下所示，请一并参照。

第 146 页的「4.5.1 关于机器人的坐标系」

第 147 页的「4.5.2 关于基本变换」

(6) 该数据的系统初始值为 P_NBase=(0,0,0,0,0,0)(0,0)。

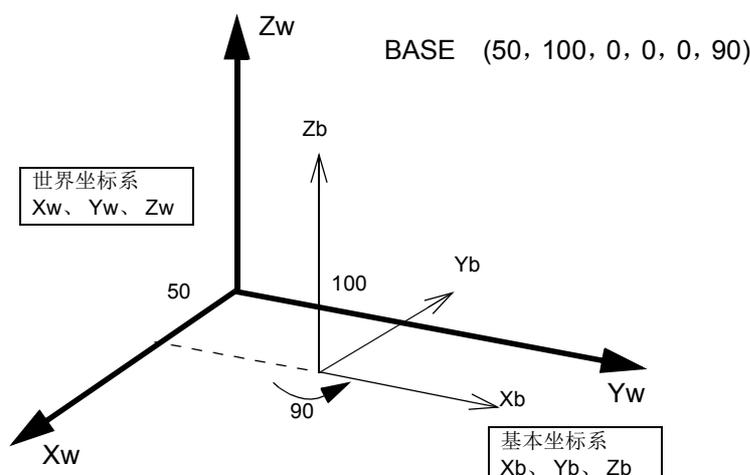


图 4-11: 坐标系的概念图

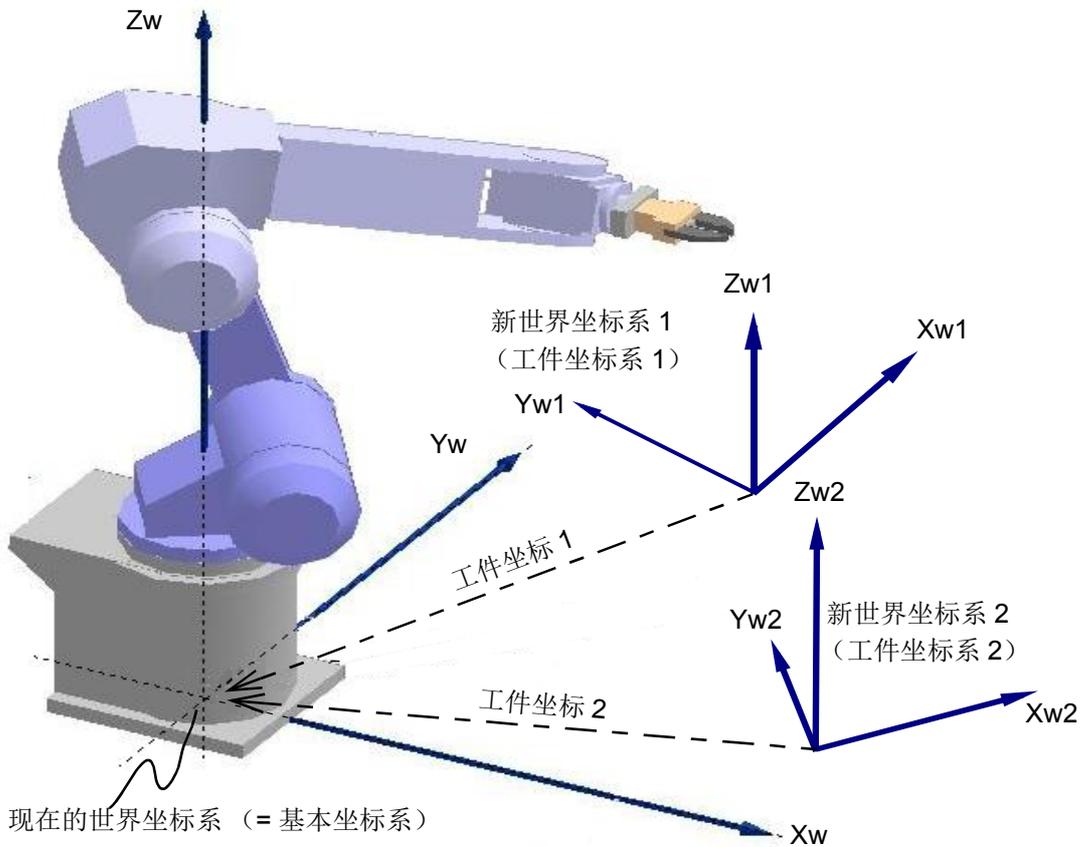


图 4-12: 指定了工件坐标系号码的基本变换

⚠ 注意

实施基本转换后，机器人的当前位置变为以新设定的世界坐标系或工件坐标系为基准的值。移动指令中指定的目的位置也作为新设定的世界坐标系或工件坐标系内的位置处理。

因此，之前的示教数据可能无法直接使用。示教时的坐标系需要和新设定的坐标系相同。

如果坐标系不一致，则机器人会移动至预想不到的位置，从而造成财产损失或人身事故。

使用基本变换功能时，请对要转换的基本坐标系和示教位置的关系进行妥善管理，使机器人正确动作，更有效地利用基本变换功能。

【相关参数】

MEXBS、WKnCORD (n 为 1 ~ 8)、MEXBSNO

【相关系统状态变量】

M_BsNo、P_Base/P_NBase、P_WkCord

Call P

【功能】

执行指定的程序的程序。(GoSub 的程序调用版)
用子程序, 用 End 指令或在最终行返回到主程序。

【格式】

Call P " □ <程序名>" [, <自变量>[, <自变量>]…]

【用语】

<程序名> 在字符串常数或字符串变量指定程序。
关于程序名的基准请参照第134页的「(1)程序名制作」。

<自变量> 程序被呼叫出时, 程序会指定替换的变量或常数。
自变量的最大个数为16。

【例文】

① 在调用程序传递自变量时

主程序

```
1 M1=0
2 CallP "10" ,M1,P1,P2
3 M1=1
4 CallP "10" ,M1,P1,P2
  :
10 CallP "10", M2,P3,P4
  :
15 End
```

"10" 子程序侧

```
1 FPrm M01, P01,P02
2 If M01<>0 Then GoTo *LBL1
3 Mov P01
4 *LBL1
5 Mvs P02
6 End          ' 在此会回返回到主程序
```

※ 在主程序的步号 2、4 被执行时, M1, P1, P2 各别被设定为子程序的 M01, P01, P02, 在主程序的步号 10 被执行时, M2, P3, P4 被设定为子程序的 M01, P01, P02。

② 在调用程序没有传递自变量时

主程序

```
1 Mov P1
2 CallP "20"
3 Mov P2
4 CallP "20"
5 End
```

"20" 子程序侧

```
1 Mov P1          ' 子程序的 P1 和主程序的 P1 不同。
2 Mvs P002
3 M_Out(17)=1
4 End          ' 在此会回返回到主程序
```

【说明】

- (1) 在 Call P 指令里被执行的子程序，以 End 指令结束，并返回到主程序。
(和 GoSub 执行时 Return 指令相当)。没有 End 指令等的情况下，最终行执行后，返回到主程序。
- (2) 将自变量替换的情况下，以 CallP 指令记述自变量的同时，必须在子程序的前头以 F Prm 指令定义自变量。
- (3) 主程序和子程序，变量的类型及个数不同的情况下，执行时会发生报警。
- (4) 将程序复位的情况下，会返回到主程序的前头控制。
- (5) 在主程序里执行的定义文 (Def Act, Def FN, Def Plt, Dim 指令)，在子程序为无效。
从子程序返回的时点会再度变成有效。
- (6) 工具数据在子程序中也有效。Accel、Spd、Ovrd、JOvrd 的值无效。
Oadl 的模式有效。
- (7) 在子程序中使用 CallP, 并且可以执行别的子程序。
但是，无法在主程序及已经有在别的任务插槽中，呼叫出执行中的程序。而且，也无法呼叫出本身的程序。
- (8) 可以从最初的主程序开始，执行 8 阶段 (阶层) 的子程序 (Call P)。
- (9) 从主程序往子程序，可以依据自变量替换变量的值，但是无法在子程序的处理结果代入自变量，然后替换到主程序。将子程序的处理结果使用在主程序的情况下，请使用全局变量替换值。

【相关指令】

FPrm

ChrSrch (Character Search)

【功能】

从文字排列中检索字符串。

【格式】

```
ChrSrch □ < 字符串数组 >, < 字符串 >, < 检索结果储存处 >
```

【用语】

< 字符串数组 > 指定变成检索的字符串排列。
 < 字符串 > 指定检索的字符串。
 < 检索结果储存处 > 设定检索字符串找到的要素号码。

【例文】

```
1 Dim C1$(10)
2 C1$(1)="ABCDEFG"
3 C1$(2)="MELFA"
4 C1$(3)="BCDF"
5 C1$(4)="ABD"
6 C1$(5)="XYZ"
7 C1$(6)="MELFA"
8 C1$(7)="CDF"
9 C1$(8)=" 机器人 "
10 C1$(9)="FFF"
11 C1$(10)="BCD"
12 ChrSrch C1$(1), " 机器人 ", M1        ' M1 设定为 8。
13 ChrSrch C1$(1), "MELFA", M2        ' M2 设定为 2。
```

【说明】

- (1) 从字符串数组检索字符串，将完全一致的字符串排列的要素号码设定在〈检索结果储存处〉。部份一致的不会被检索。
 在上述例文中即使记述为
 ChrSrch C1\$(1), "ROBO", M1
 一致的字符串也不会被检索。
- (2) 找不到检索字符串的情况下，〈检索结果储存处〉会被设定为 0。
- (3) 字符串的检索会从要素号码 1 开始，依号码顺序执行，最先找到的要素号码会被设定。
 在上述例文中，即使记述为
 ChrSrch C1\$(3), "MELFA", M2
 在 M2 会被设定为 2。（在 C1\$(2) 和 C1\$(6) 设定相同字符串）
- (4) 成为检索对象〈字符串数组〉的，只有一次元排列。指定二维以上的变量的情况下，执行时会发生报警。

CavChk On

【功能】

使干涉回避的停止功能有效。

本功能对对应机型有所限定。关于详细内容请参照 [「5.24 关于干涉回避功能」](#)。

【格式】

CavChk □ <On/Off>[,< 机器人 CPU 号码 >[,NOErr]]

【用语】

<On/Off>	On: 使干涉回避的停止功能有效。 Off: 使干涉回避的停止功能无效。
< 机器人 CPU 号码 >	指定使干涉回避的停止功能有效 / 无效的机器人 CPU。 省略或指定了 0 时, 所连接的所有机器人均为对象。 0 ~ 3、以常数或变量表述。 此外, 该自变量仅在 CR7xx-Q 控制器中起作用。
NOErr	干涉检知时, 不发生报警。 省略时, 发生报警 (L4931)。

【例文】

参照第 500 页的 [「5.24.10 采样程序」](#)。

【说明】

- (1) 使干涉回避的停止功能有效。初始状态取决于参数 CAV 的设定。
参数 CAV 中设定为不可使用时, 发生报警 (L4930)。
- (2) 省略了机器人 CPU 号码时, 在已安装的机器人 CPU 间之中干涉回避有效的 CPU 间进行确认。
- (3) 指定了机器人 CPU 号码时, 指定的机器人 CPU 为干涉回避无效时, 发生报警 (L4930)。
- (4) 抓手、工件模型可通过 LoadSet 指令指定。
- (5) 跟踪有效时, 不能使其有效。(发生报警 (L4936))

Close

【功能】

将指定文件号码的文件（CRnD 含通信端口）关闭。

【格式】

```
Close □ [[#] < 文件号码 > [, [# < 文件号码 >]
```

【用语】

<文件号码> 指定1~8想要关闭的文件。
只有数值常数。省略时，会关闭开启中的全部文件。

【例文】

1 Open "temp.txt" For Append As #1 : 将称为 temp.txt 的文件，在 Append 模式，作为文件号码 1 开启。
2 Print #1, "abc" : 在文件上写上 "abc"。
3 Close #1 : 关闭文件。

【说明】

(1) 将依据 Open 指令所开启的文件关闭。缓存内有剩余数据的情况下，会进行清除。
关闭文件时，在缓存内的剩余数据，请以下列方式处理。

表 4-15: 文件关闭时的各缓存处理动作

缓存种类	文件关闭时的处理
通信回路接收缓存	缓存内容被废弃。通信端口
通信回路传送缓存	(因为在 PRINT 指令执行时已被传送，因此传送缓存内没有剩余数据)
文件读出缓存	缓存内容被废弃。
文件写入缓存	将缓存内容写入文件后，关闭文件

(2) 依据 End 文的执行也会被关闭。
(3) 省略文件号码的情况下，关闭全部的文件。

【相关指令】

Open、Print、Input

Clr (Clear)

【功能】

删除通用输出信号 / 程序内区域性数值变量 / 数值全局变量。

【格式】

```
Clr □ <TYPE>
```

【用语】

<机型>

可指定常数或变量。

0: 执行以下的1~3。

1: 根据输出复位模式将通用输出信号删除。输出复位模式以参数：
ORS10000~ORS1816指定。

请参照第447页的「5.14关于输出信号复位模式」

(0: OFF、1: ON、*: 保持)

2: 将程序内使用的区域性数值变量及数值数组全部0清除。

3: 将外部数值变量及外部数值数组全部（系统全局变量和使用者定义全局变量）0清除。全局变量不会清除。

【例文】

① 根据输出复位模式将通用输出信号清除。

```
1 Clr 1
```

② 将程序内使用的区域性数值变量及数值数组0清除。

```
1 Dim MA(10)
```

```
2 Def Inte IVAL
```

```
3 Clr 2           ' MA(1)~MA(10)、IVAL、及程序内区域数值变量0清除。
```

③ 外部数值变量及外部数值数组全部0清除。

```
1 Clr 3
```

④ 同时执行上述①~③

```
1 Clr 0
```

【关连参数】

ORS10000 ~ ORS1816

【关连系统状态变量】

M_In/M_Inb/M_In8/M_Inw/M_In16、 M_Out/M_Outb/M_Out8/M_Outw/M_Out16

Cmp Jnt (Comp Joint)

【功能】

在关节坐标指定轴，并开始柔软控制模式 (弹性伺服模式)。

注) 本指令有限定的对应机型。请参照 【可使用的机器人机型】。

【格式】

Cmp □ Jnt, < 轴指定 >

【用语】

<轴指定> 将柔软控制的轴以位模式指定。
1: 有效、0: 无效 &B000000
对应654321轴。

【例文】

```
1 Mov P1
2 CmpG 0.0,0.0,1.0,1.0, , , ,      ' 柔软度的设定
3 Cmp Jnt,&B11                      ' 使 J1 轴和 J2 轴成柔软状态
4 Mov P2
5 HOpen 1
6 Mov P1
7 Cmp Off                            ' 返回到平常状态
```

【说明】

- (1) 在关节坐标的轴单位，可以执行机器人柔软控制。例如，以水平多关节机器人在上下方向插入 PIN 的情况，依据将 J1 轴和 J2 轴变得柔软，使在 PIN 插入的程度上变得更顺畅 (参考上列例文)。
- (2) 柔软的程序，依据设定弹性常数的 CmpG 指令来指定。机器人为 RH-SDH 系列的情况下，在水平轴的 J1、J2 轴指定 0.0=、使成为无伺服功能 (弹性常数 =0)，(垂直轴即使设定为 0.0，也不会变成无伺服。此外，在此功能下，移动到超过动作限制位置的话，请注意偏移量不要过大)。另外，使用此无伺服机功能的情况下，则没有下列 (4)，(5) 的功能。
- (3) 在柔软状态，即使在机器人的程序执行完成后，也会执行 Cmp Off 指令、或持续到电源切断为止。
- (4) 在柔软状态下执行压制的时候，无法移动到超过各关节轴的动作限制位置。
- (5) 依据压制，本来的目的位置和实际的机器人位置的偏移量超过 200mm 的话，机器人不会继续动作，程序会移动到下一行执行。
- (6) Cmp Jnt/Pos/Tool 无法同时使用。也就是说在 Cmp Jnt 指令动作时，若执行 Cmp Pos 指令或 Cmp Tool 指令的话，会发生报警。请解除 Cmp Off 指令后再执行这些指令。
- (7) 本指令在动作时，从伺服关闭变成开启的话，机器人的位置会变化，请特别注意。
- (8) 可以在弹性伺服模式里执行 JOG 操作。但是，因为无法在示教单元上解除弹性伺服模式的设定，因此请在程序执行本指令或是在示教单元的程序编辑画面的直接执行设定。
- (9) 变更轴指定的情况下，再次以 Cmp Off 指令解除伺服弹性模式后，再执行 Cmp Jnt 指令。
- (10) 弹性伺服模式只在机器人本体的轴有效，在附加轴指定也不会动作。
- (11) 在弹性伺服模式作用时，依据 Fine 指令指定决置决定完成条件的话，根据动作，机器人无法到达目的位置的定位完成脉冲，会等待动作指令的完成，无法执行程序。请不要同时使用伺服弹性模式和 Fine 指令。
- (12) 在关节坐标系的弹性模式下动作时发生误差过大 1 错误 (H096n) 的情况下，增大参数 CMPJCLL 的设定值可以抑制错误的发生。
(设定范围：1 ~ 10)
请逐渐增大该参数的设定值，并设定为不会发生误差过大 1 错误的值。
(该功能在控制器的软件版本为 R4b 版 /S4b 版以上时可使用。)

 **注意**

弹性伺服模式会执行 **Cmp Off** 指令，或继续执行到电源切断为止。在执行程序号码的变更及 **JOG** 操作时，请特别注意。

 **注意**

以 **Cmp Jnt** 指令作为弹性伺服模式状态执行 **JOG** 操作的情况下，请在关节 **JOG** 模式操作。在其它的 **JOG** 模式操作的话，因为 **JOG** 操作和弹性伺服模式控制坐标的方向不同，因此会有想移动的方向和机器人动作方向不同的情况发生。

 **注意**

在弹性伺服模式的状态示教位置时，请将伺服关闭之再执行。直接就伺服开启的状态执行示教操作的话，并不是实际的机器人位置，而是原本的指令位置。机器人会往和示教时不同的位置运行，请特别注意。

【可使用的机器人机型】

RH-F 系列

【关连系统状态变量】

M_CmpDst

【相关指令】

Cmp Off、**CmpG (Composition Gain)**、**Cmp Tool**、**Cmp Pos**

【相关参数】

CMPJCLL

Cmp Pos

【功能】

在直交坐标指定轴方向，并使指定机器人开始柔软控制模式（弹性伺服模式）。

注）本指令有限定的对应机型。请参照【可使用的机器人机型】。

【格式】

Cmp □ Pos, < 轴指定 >

【用语】

<轴指定> 将柔软控制的轴以位模式指定。
1: 有效、0: 无效 &B000000
对应CBAZYX轴。

【例文】

1 Mov P1	' 往部品插入位置上空移动
2 CmpG 0.5, 0.5, 1.0, 0.5, 0.5, , ,	' 柔软度的设定
3 Cmp Pos, &B011011	' 使 X,Y,A,B 轴成柔软状态
4 Mvs P2	' 往部品插入位置移动
5 M_Out(10)=1	' 定位用拉锁闭指示
6 Dly 1.0	' 拉锁闭完成等待
7 HOpen 1	' 抓手开
8 Mvs, -100	' 往 TOOL 坐标的 Z- 方向 100mm 待避
9 Cmp Off	' 返回到平常状态。

【说明】

- (1) 在直交坐标，可以执行机器人柔软控制。例如，在上下方向插入 PIN 的情况，依据将 X,Y,A,B 轴方向变得柔软，使在 PIN 插入的程度上变得更顺畅。
- (2) 柔软的程序，依据 CmpG 指令来指定。
- (3) 在柔软状态，即使在机器人的程序执行完成后，也会执行 Cmp Off 指令、或持续到电源切断为止。
- (4) 在柔软状态下执行压制的时候，无法移动到超过各关节轴的动作限制位置。
- (5) 指令位置和实际位置的偏移量，可以依据 M_CmpDst 来读出。可以此变量来确认 PIN 的插入成功与否。
- (6) 依据压制，本来的目的位置和实际的机器人位置的偏移量超过 200mm 的话，机器人不会继续动作，程序会移动到下一行执行。
- (7) Cmp Jnt/Pos/Tool 无法同时使用。也就是说在 Cmp Pos 指令动作时，若执行 Cmp Jnt 指令或 Cmp Tool 指令的话，会发生报警。请解除 Cmp Off 指令后再执行这些指令。
- (8) 本指令在动作时，从伺服关闭变成开启的话，机器人的位置会变化，此外，重力方向保持的轴，会有落下的可能。
- (9) 可以在弹性伺服模式里执行 JOG 操作。但是，因为无法在示教单元上解除弹性伺服模式的设定，因此请在程序执行本指令或是示教单元盒的程序编辑画面的直接执行设定。
- (10) 变更轴指定的情况下，再次以 Cmp Off 指令解除伺服弹性模式后，再执行 Cmp Pos 指令。
- (11) 使机器人在接近特异点的地方动作的话，会发生报警且机器人有可能会发生大动作及振动，因此，请不要接近特异点地方动作。假如发生此状态的情况下，请在伺服关闭状态下，先执行 Cmp Off 指令，（或者重新启动电源）然后解除弹性伺服模式后，将机器人从特异点远离后，重新使弹性伺服模式为有效。
- (12) 弹性伺服模式只在机器人本体的轴有效，在附加轴指定也不会动作。
- (13) 在弹性伺服模式作用时，依据 Fine 指令指定决置决定完成条件的话，根据动作，机器人无法到达目的位置的定位完成脉冲，会等待动作指令的完成，无法执行程序。请不要同时使用伺服弹性模式和 Fine 指令。

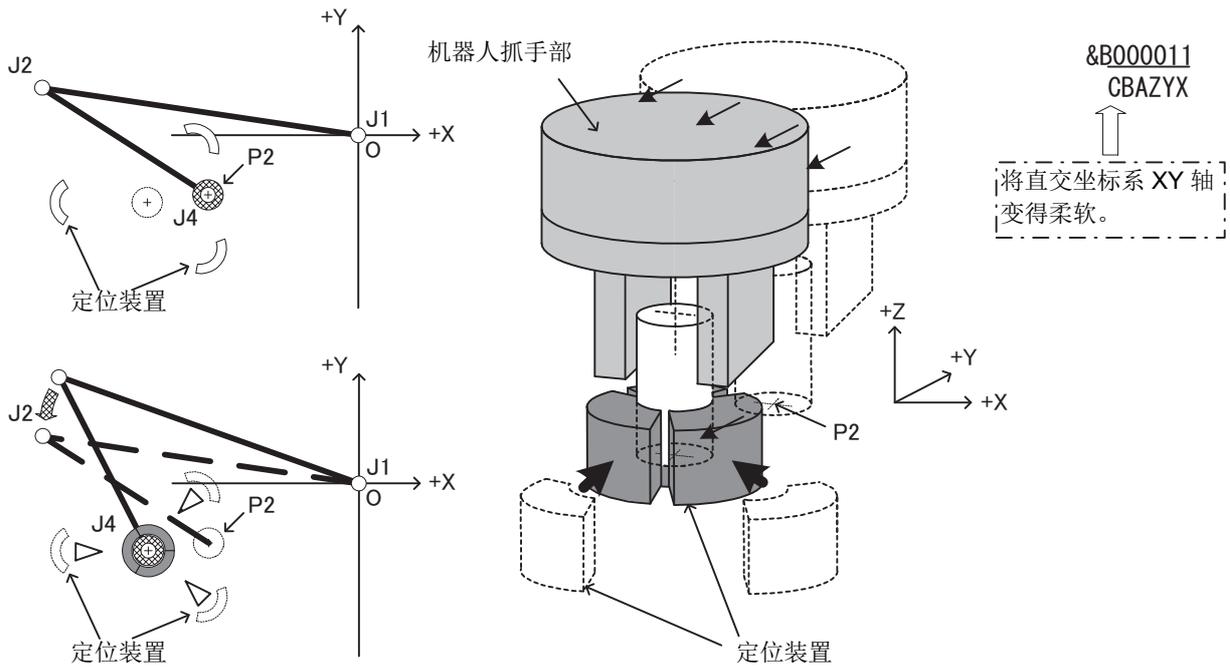


图 4-13: 伺服弹性模式使用例

⚠ 注意

弹性伺服模式会执行 **Cmp Off** 指令，或继续执行到电源切断为止。在执行程序号码的变更及 **JOG** 操作时，请特别注意。

⚠ 注意

以 **Cmp Pos** 指令作为弹性伺服模式状态执行 **JOG** 操作的情况下，请在直交 **JOG** 模式操作。在其它的 **JOG** 模式操作的话，因为 **JOG** 操作和弹性伺服模式控制坐标的方向不同，因此会有想移动的方向和机器人动作方向不同的情况发生。

⚠ 注意

在弹性伺服模式的状态示教位置时，请将伺服关闭之后再执行。直接就伺服开启的状态执行示教操作的话，并不是实际的机器人位置，而是示教原本的指令位置。机器人会往和示教时不同的位置运行，请特别注意。

【关连系统状态变量】

M_CmpDst

【相关指令】

Cmp Off、**CmpG (Composition Gain)**、**Cmp Tool**、**Cmp Jnt (Comp Joint)**

Cmp Tool

【功能】

在 TOOL 坐标指定机器人方向，并使指定机器人开始柔软控制模式（弹性伺服模式）。
注) 本指令有限定的对应机型。请参照【可使用的机器人机型】。

【格式】

Cmp □ Tool, < 轴指定 >

【用语】

<轴指定> 将柔软控制TOOL坐标的轴以位模式指定。
1: 有效、0: 无效 &B000000
对应CBAZYX轴。

【例文】

1 Mov P1	' 往部品插入位置上空移动
2 CmpG 0.5, 0.5, 1.0, 0.5, 0.5, , ,	' 柔软的设定
3 Cmp Tool, &B011011	' 将 TOOL 坐标的 X,Y,A,B 轴变成柔软状态
4 Mvs P2	' 往部品插入位置移动
5 M_Out(10110)=1	' 位置决定用拉锁闭指示
6 Dly 1.0	' 拉锁闭完成等待
7 HOpen 1	' 抓手开
8 Mvs, -100	' 往 TOOL 坐标的 Z- 方向 100mm 待避
9 Cmp Off	' 返回到平常状态。

【说明】

- (1) 在 TOOL 坐标，可以执行机器人柔软控制。关于相关的 TOOL 坐标请参照第 432 页的「5.6 关于标准 TOOL 坐标」。
- (2) 例如，在 TOOL 坐标的 Z 轴方向插入 PIN 的情况，依据将 X,Y,A,B 轴变得柔软，使在 PIN 插入上变得更顺畅。
- (3) 柔软的程序，依据 CmpG 指令来指定。
- (4) 在柔软状态，即使在机器人的程序执行完成后，也会执行 Cmp Off 指令、或持续到电源切断为止。
- (5) 在柔软状态下执行压制的时候，无法移动到超过各关节轴的动作限制位置。
- (6) 指令位置和实际位置的偏移量，可以依据 M_CmpDst 来读出。可以此变量来确认 PIN 的插入成功与否。
- (7) 依据压制，本来的目的位置和实际的机器人位置的偏移量超过 200mm 的话，机器人不会继续动作，程序会移动到下一行执行。
- (8) Cmp Jnt/Pos/Tool 无法同时使用。也就是说在 Cmp Pos 指令动作时，若执行 Cmp Jnt 指令或 Cmp Tool 指令的话，会发生报警。请解除 Cmp Off 指令后再执行这些指令。
- (9) 本指令在动作时，从伺服关闭变成开启的话，机器人的位置会变化，此外，重力方向保持的轴，会有落下的可能。
- (10) 可以在弹性伺服模式里执行操作。但是，因为无法在示教单元上解除弹性伺服模式的设定，因此请在程序执行本指令或是示教单元盒的程序编辑画面的直接执行设定。
- (11) 变更轴指定的情况下，再次以 Cmp Off 指令解除伺服弹性模式后，再执行 Cmp Tool 指令。
- (12) 垂直 5 轴型机器人时，仅 X 轴、Z 轴可以进行轴指定。
- (13) 使机器人在接近特异点的地方动作的话，会发生报警且机器人有可能会发生大动作及振动，因此，请不要接近特异点地方动作。假如发生此状态的情况下，请在伺服关闭状态下，先执行 Cmp Off 指令，（或者重新启动电源）然后解除弹性伺服模式后，将机器人从特异点远离后，重新使弹性伺服模式为有效。
- (14) 弹性伺服模式只在机器人本体的轴有效，在附加轴指定也不会动作。

(15) 在弹性伺服模式作用时，依据 Fine 指令指定决置决定完成条件的话，根据动作，机器人无法到达目的位置的定位完成脉冲，会等待动作指令的完成，无法执行程序。请不要同时使用伺服弹性模式和 Fine 指令。

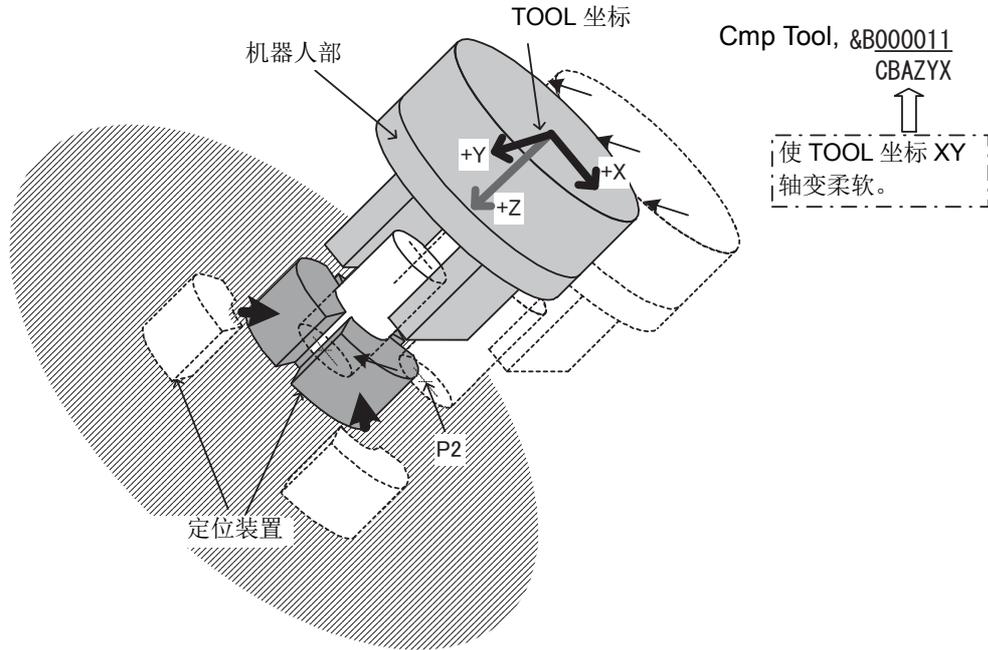


图 4-14：伺服弹性模式使用例

⚠ 注意

弹性伺服模式会执行 Cmp Off 指令，或继续执行到电源切断为止。在执行程序号码的变更及 JOG 操作时，请特别注意。

⚠ 注意

以 Cmp TOOL 指令作为弹性伺服模式状态执行 JOG 操作的情况下，请在 TOOL JOG 模式操作。在其它的 JOG 模式操作的话，因为 JOG 操作和弹性伺服模式控制坐标的方向不同，因此会有想移动的方向和机器人动作方向不同的情况发生。

⚠ 注意

在弹性伺服模式的状态示教位置时，请将伺服关闭之再执行。直接就伺服开启的状态执行示教操作的话，并不是实际的机器人位置，而是示教原本的指令位置。机器人会往和示教时不同的位置运行，请特别注意。

【关连系统状态变量】

M_CmpDst

【相关指令】

Cmp Off、CmpG (Composition Gain)、Cmp Pos、Cmp Jnt (Comp Joint)

Cmp Off

【功能】

解除机器人柔软控制模式（弹性伺服模式）。

【格式】

Cmp □ Off

【例文】

1 Mov P1	' 往部品插入位置上空移动
2 CmpG 0.5, 0.5, 1.0, 0.5, 0.5, , ,	' 柔软度的设定
3 Cmp Pos, &B011011	' 使 X,Y,A,B 轴成柔软状态
4 Mvs P2	' 往部品插入位置移动
5 M_Out(10110)=1	' 定位用拉锁闭指示
6 Dly 0.5	' 拉锁闭完成等待
7 HOpen 1	' 抓手开
8 Mvs, -100	' 往 TOOL 坐标的 Z- 方向 100mm 待避
9 Cmp Off	' 返回到平常状态。

【说明】

- (1) 解除以 Cmp Tool、Cmp Pos、Cmp Jnt 开始的弹性伺服状态。
- (2) 解除在弹性伺服模式的 JOG 操作，请在本指令在程序里执行或是在示教单元的程序编辑画面里直接执行。

【相关指令】

[CmpG \(Composition Gain\)](#)、[Cmp Tool](#)、[Cmp Pos](#)、[Cmp Jnt \(Comp Joint\)](#)

CmpG (Composition Gain)

【功能】

指定机器人柔软控制。

注) 本指令有限定的对应机型。请参照 **【可使用的机器人机型】**。

【格式】

Cmp Pos、Cmp Tool 时

```
CmpG □ [<X 轴增益 >], [<Y 轴增益 >], [<Z 轴增益 >], [<A 轴增益 >],
      [<B 轴增益 >], [<C 轴增益 >], ,
```

Cmp Jnt 时

```
CmpG □ [<J1 轴增益 >], [<J2 轴增益 >], [<J3 轴增益 >],
      [<J4 轴增益 >], [<J5 轴增益 >], [<J6 轴增益 >], ,
```

【用语】

<X~C轴增益>

<J1~J6轴增益>

以常数指定。
 可指定各轴的柔软度。
 在1.0为平常的状态，最低值会依据机型而有所不同。
 各机型的最低值请参照**【说明】(7)**。
 省略值的轴会维持现在的设定值。

【例文】

1 CmpG , ,0.5, , , , , ' 只设定 Z 轴。省略轴请空白，只用逗号记述。

【说明】

- (1) 能以各轴单位指定柔软度。
- (2) 没有使 Cmp Pos、Cmp Tool 为有效的话，无法实际变成柔软控制状态。
- (3) 以指令位置和实际的位置为比例，像弹簧一样的产生力量。CmpG 为指定弹簧常数的指令。
- (4) 指令位置和实际位置的偏移量，可以依据 M_CmpDst 来读出。可以此变量来确认 PIN 的插入成功与否。
- (5) 增益值调小的话，以 Cmp Pos、Cmp Tool、Cmp Jnt 指令设定柔软控制状态时，机器人的动作位置会下降，因此，请慢慢地调整为柔软状态，并且请一边确认一边设定。
- (6) 在柔软控制状态，依照本指令将增益值调大的话，为了使原来的目的位置和实际的机器人位置的偏移量变小的控制动作起作用，机器人会动作。将增益值调大的时候，请暂时依据 Cmp Off 指令，将伺服弹性模式解除后再进行设定。
- (7) 增益值即使在最低值以下，也会被视为最低值控制。
 此外，增益值可设定为小数点2位数以上。各机型的最低值如下所示。

机型	Cmp Pos、Cmp Tool 时	Cmp Jnt 时
RH-F 系列	0.20, 0.20, 0.20, 0.20, 0.20, 0.20	0.00, 0.00, 0.20, 0.00, 1.00, 1.00
RV-F 系列	0.01, 0.01, 0.01, 0.01, 0.01, 0.01	不可使用

- (8) 弹性伺服模式只在机器人本体的轴有效，在附加轴 (J7,J8 或 L1,L2) 指定也不会动作。

Cnt (Continuous)

【功能】

指定插补的连续动作。依据连续动作，可以缩短动作时间。

【格式】

```
Cnt □ <1/0>[, <数值 1>][, <数值 2>]
```

【用语】

- <1/0> 指定连续动作、加减速动作。指定以常数执行。
- 1: 连续动作指定。使插补和插补圆滑的连续动作。
在插补连续场所，减少速度的降低，轨迹变成沿着内侧转。
 - 0: 加减速动作（初始值）到达插补的目的位置后，执行下一个插补。
在插补连接场所，会暂时降低速度。
- <数值1> 将轨迹变换时的下一个插补开始的最大接近距离以mm单位指定。
可以使用常数或变量。
- <数值2> 轨迹变换时的前一个插补结束的最大接近距离，使用mm单位指定。
可以使用常数或变量。

【例文】

指定轨迹变换时的最大接近距离的情况

- | | |
|-----------------|---|
| 1 Cnt 0 | ' 将 Cnt(连续动作) 设定为无效 |
| 2 Mvs P1 | ' 有加减速的执行动作 |
| 3 Cnt 1 | ' 将 CNT (连续动作) 设定为有效 (此行以后的插补为连续动作) |
| 4 Mvs P2 | ' 和下一个插补的连续为连续动作 |
| 5 Cnt 1,100,200 | ' 指定在开始侧 100mm、结束侧 200mm 做连续动作 |
| 6 Mvs P3 | ' 用指定距离在插补的前后连续动作 |
| 7 Cnt 1,300 | ' 指定在开始侧 300mm、结束侧 300mm 做连续动作 |
| 8 Mov P4 | ' 在开始侧以 300mm 连续动作 |
| 9 Cnt 0 | ' 将 Cnt(连续动作) 设定为无效 |
| 10 Mov P5 | ' 有加减速的执行动作 |

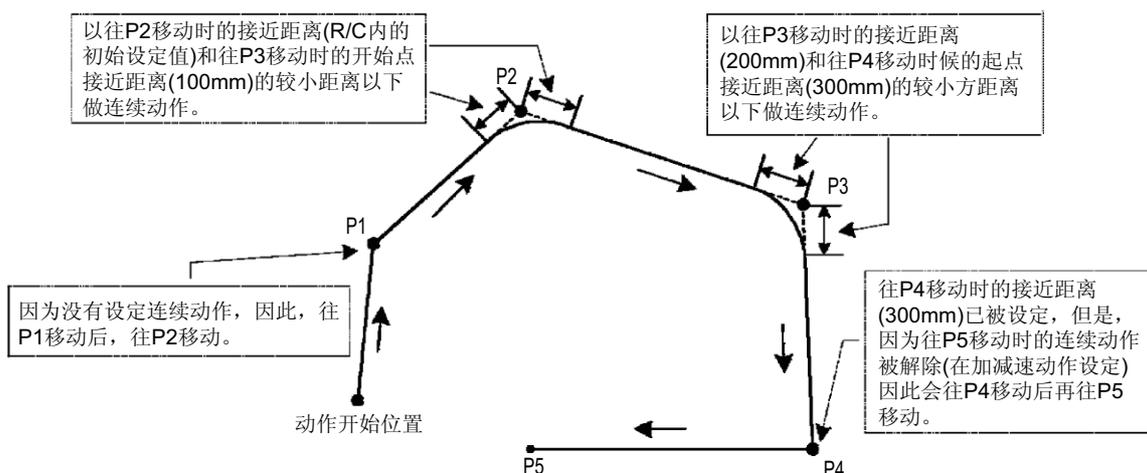


图 4-15: 连续轨迹动作的例子

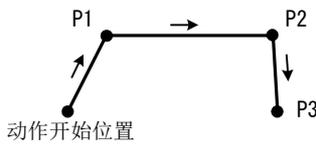
⚠ 注意

机器人根据当时的指定速度，可能会以不同的轨迹动作。尤其是转角部分，其内环距离有可能会发生变化，因此，在最初开始自动运行时应先以低速运行，然后一边留意与周边装置之间的干涉，一边逐渐提升运行速度。

【解说】

- (1) 被 Cnt 1 ~ Cnt 0 围起来的插补 (例文步号 4 ~ 8) 会成为连续动作的对像。
- (2) 系统的初始值为 Cnt 0 (非连续动作)。
- (3) 省略数值 1、数值 2 的情况下, 开始从减速开始位置和下一个插补之间的连接。
- (4) 如图 4-16 中所示, 在加减速动作模式, 在目的位置前执行减速, 往目的位置移动后, 开始往下一个目的位置的加速。对应此点, 连续动作模式在目的位置前进行减速, 但是不会完全的停止, 为了往下一个目的位置的移动而开始加速。因此, 不会通过各目的位置, 只会从附近通过。

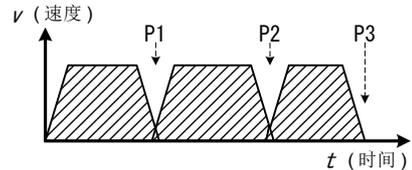
加减速动作模式



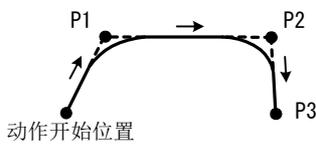
```

10 Mov P1
20 Mov P3
30 Mov P3
    
```

往 P1, P2, P3, 分别执行加速 · 减速
往目的位置移动后, 再往下一个目的位置移动。



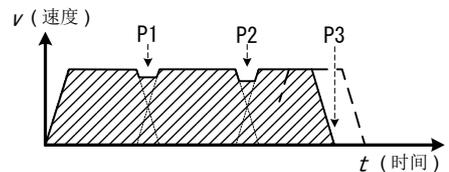
连续动作模式



```

10 Cnt 1
20 Mov P1
30 Mov P2
40 Mov P3
50 Cnt 0
    
```

通过 P1, P2 的附近, 往 P3 移动。



※ 上面所示为范例。依据移动距离和速度, 插补连续时, 会发生加减速的情况。

图 4-16: 加减速动作和连续动作

- (5) 所谓接近距离, 是指往下一个目的位置的插补动作的变换距离。省略此接近距离 (数值 1、数值 2) 的情况下, 加减速开始位置会变成往下个插补的变换位置。此情况下, 会通过从目的位置脱离的位置, 但是动作时间会变成最短。想要通过最接近目的位置的话, 请设定接近距离 (数值 1、数值 2)。

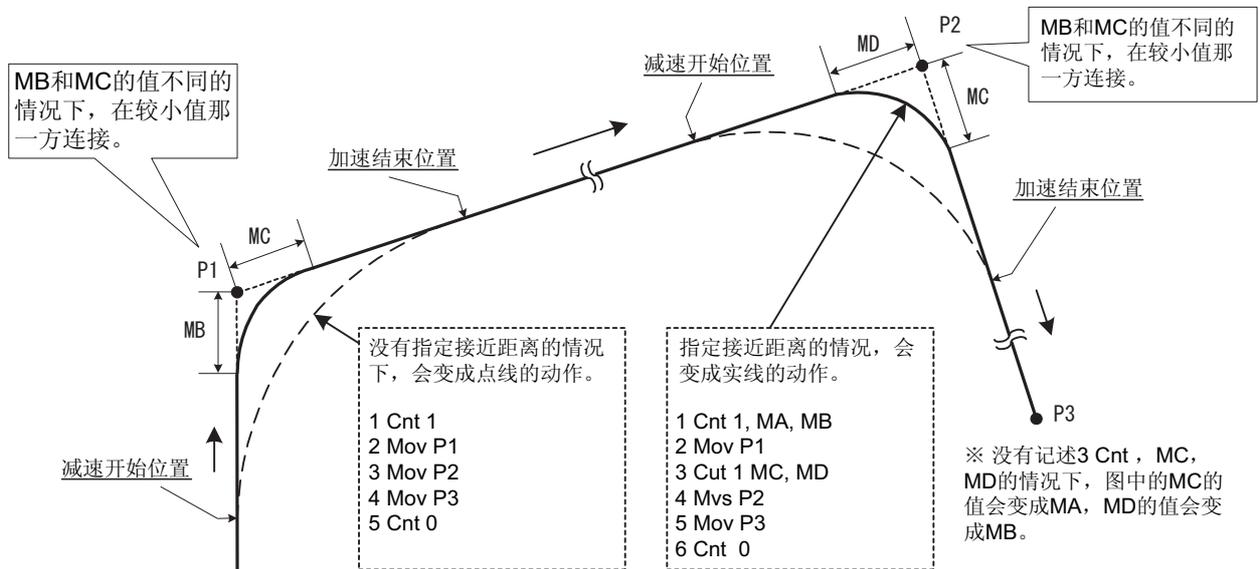


图 4-17: 接近距离的设定

- (6) 数值 1、数值 2 的指定不同的情况下, 会在较小方的位置 (距离) 做连续动作。
- (7) 省略数值 2 的情况下, 数值 2 会被设定和数值 1 同值。
- (8) 指定连续动作时, 根据 Fine 指令所做的位置决定完成指定会变成无效。
- (9) 将接近距离 (数值 1、数值 2) 调小的话, 会比 Cnt 0 的状态变成动作时间更长的情况。
- (10) 即使指定连续动作时, 在插补方式指定特异点通过的插补指令, 也会变成加减速动作。

ColChk (Col Check)

【功能】

在自动运行里设定冲突检知功能的有效 / 无效。

冲突检知功能是在机器人的抓手和手臂与周边机器发生干涉时，迅速停止机器人，且将机器人抓手的 Tool 部份及周边机器的破损及变形等伤害，抑制到最小。但是，无法 100% 的阻止。冲突检知功能有限制使用机型。（请参照【可使用的机器人机型】）。

【格式】

```
ColChk □ On[, NOErr]/Off
```

【用语】

On 使冲突检知为有效。检测到冲突的话，紧急停止机器人，且发出1010的报警号码，关闭伺服。
Off 使冲突检知为无效。
NOErr 检测到冲突的话，不会发生报警。（省略时会发生报警）

【例子 1】

发生冲突的话，产生报警的情况

```
1 ColLvl 80,80,80,80,80,80,,            ' 指定冲突检知的容许标准
2 ColChk On                                ' 使冲突检知为有效
3 Mov P1
4 Mov P2
5 Dly 0.2                                 ' 等待到动作完成为止（在 Fine 指令也 OK）
6 ColChk Off                               ' 使冲突检知为无效
7 Mov P3
```

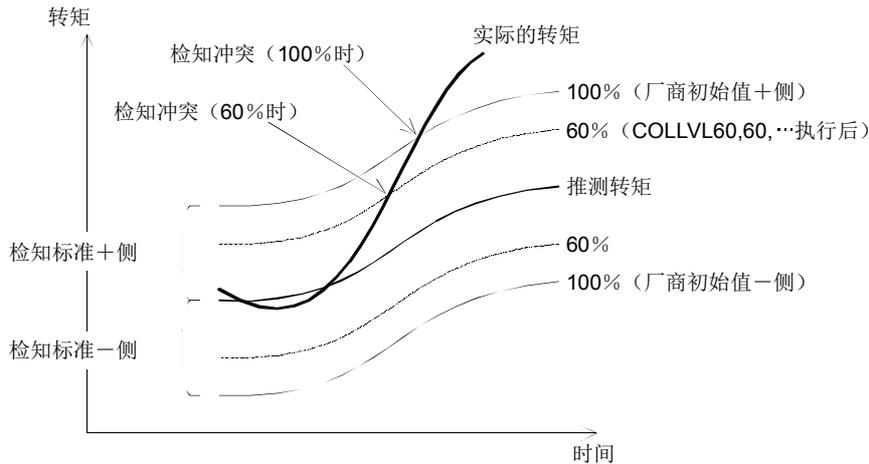
【例子 2】

发生冲突的话，使用中断处理的情况

```
1 Def Act 1,M_ColSts(1)=1 GoTo *HOME,S   ' 在检测到冲突的时候，将执行处理以中断来定义
2 Act 1=1
3 ColChk On,NOErr                         ' 在报警非发生模式，将冲突检知设为有效
4 Mov P1
5 Mov P2                                 ' 在 40 ~ 70 行执行中检测到冲突的话，跳转到中断处理
6 Mov P3
7 Mov P4
8 ColChk Off
9 Act 1=0
   :
100 *HOME                                 ' 冲突检知时的中断处理
101 ColChk Off                             ' 使冲突检知为无效
102 Servo On                              ' 开启伺服
103 PESC=P_ColDir(1)*(-2)                ' 制作待避动作的移动量
104 PDST=P_Fbc(1)+PESC                    ' 制作待避位置
105 Mvs PDST                              ' 往待避位置移动
106 Error 9100                             ' 让使用者定义的 L 标准报警发生，停止运行
```

【说明】

(1) 所谓冲突检知是推定以动作指令在移动中，对那个程度轴发生转矩，且推定转矩和实际转矩的差异，超过容许范围的情况下，则判断为冲突，使机器人紧急停止的功能。



- (2) 电源刚开启之后，冲突检知为无效的状态。请将参数 COL 设定为有效后使用。
此外，本指令为指定程序运行（含单步前进、单步跳转）冲突检知的有效/无效。中断中和 JOG 操作中，没有执行程序时的有效/无效状态，则是依赖参数 COL 的要素 3 的设定。
- (3) 检知标准可以用 Col Lvl 指令调整。检知标准的初始值会变成参数 COLLVL 的设定值。
- (4) 以本指令将冲突检知为有效后，以 ColChk Off 指令设为无效，除非将程序复位、执行 End 指令、关闭电源，否则此状态会被持续。
- (5) 即使在本指令设定冲突检知为无效，以 ColLvl 指令设定的冲突检知标准也会被保持。
- (6) 连续功能有效时，即使将电源关闭，也会在下次电源开启时，恢复以前设定的冲突检知设定状态。
- (7) 指定 NOErr（报警非发生模式）的时候，依据状态变量 M_ColSts 插入（插入条件为 M_ColSts(*)=1 的插入。* 为机器号码）没有变成有效的话，会发生报警 3950。请参照【例子 2】。另外，在报警非发生模式中，使此插入处理为无效的话，会发生报警 3960。
- (8) 在报警非发生模式里检知到冲突的话，机器人会关闭伺服且停止，但是不会变成报警，也不会中断运行。但是，在报警历史里会记录检知到冲突的事。（记录历史，只有在没有合并发生其它报警的情况下。）
- (9) 在无法使用冲突检知功能的机器人上执行 ColChk On 和 ColChk On, ONErr 的话，会发生低标准报警 3970。ColChk Off 的情况则不会发生报警，会变成无处理。
- (10) 在 Cmp 指令的伺服弹性有效中及以 Torq 指令的转矩限制有效中，无法使冲突检知变有效。此时，将冲突检知设定为有效的话，会发生报警 3940。相反的，在冲突检知有效中使 Cmp 指令及 Torq 指令为有效的话，会发生报警 3930。
- (11) 在动作指令之后记述 Col Chk Off 的话，会有无法在动作的最后停止后近检知到冲突的情况。请参照例文 1，在动作指令和 Col Chk Off 指令之间以 Dly 和 Fine 指令将定位完成后，再使 Col Chk Off。
- (12) 没有正确设定抓手重量（HNDDATn 参数）和工件重量（WRKDATn 参数）的话，会有冲突检知无法正确动作的情况。请正确使用参数。
- (13) 依据本指令使冲突检知为有效的话，会因为程序而使节拍时间（TACT TIME）变长的情况。并不是将全部程序的冲突检知功能变有效，请使用在只针对有可能对周边机器发生干涉的动作。
- (14) 本功能无法和多机器控制同时使用。

【关连系统状态变量】

M_ColSts、J_ColMxl、P_ColDir

【相关指令】

ColLvl (ColLevel)

【关连参数】

COL、COLLVL、COLLVLJG、HNDDATn、WRKDATn

ColLvl (ColLevel)

【功能】

在自动运行中设定冲突检知功能的检知标准。

冲突检知功能有限定使用机型。（请参照【可使用机器人机型】）。

【格式】

```
ColLvl [<J1 轴 >], [<J2 轴 >], [<J3 轴 >], [<J4 轴 >],
        [<J5 轴 >], [<J6 轴 >], ,
```

【用语】

<J1~J6轴> 检知标准以%指定，1~500%，省略时会保持前一个值。
 J7, J8轴没有此功能。
 初始值会变成参数COLLVL的设定值。

【例子】

```
1 ColLvl 80,80,80,80,80,,            ' 指定冲突检知的容许标准
2 ColChk On                            ' 使冲突检知为有效
3 Mov P1
4 ColLvl ,50,50,,,,                  ' 变更冲突检知的 J2, J3 轴容许标准
5 Mov P2
6 Dly 0.2                              ' 往 P2 到达的冲突检知为无效
7 ColChk Off                          ' 使冲突检知为无效
8 Mov P3
```

【说明】

- (1) 在程序运行时，设定冲突检知功能的各轴冲突容许标准。
- (2) 本指令为在自动运行时（含单步前进、单步跳转）冲突检知功能动作。程序没有运行时，（中断中和 JOG 操作中），则是依赖参数 COLLVLJG 的设定标准。
- (3) 平常电源开启后的容许标准设定值，会变成参数 COLLVLJ 的设定值。
参数 COLLVL 的设定值全轴全开始 100%。
- (4) 值调大的话，检知标准（敏感度）会迟钝，调小的话，检知标准会提升。
- (5) 检知标准调高的话，误检知的发生率会提高，因此请调整不要调太高。同时，依据姿势及动作速度，也会有误检知的情况发生。此情况下，请将检知标准调低。
- (6) 使用的控制器的软件版本为 R6b 版 /S6b 版以上、且使用 RT ToolBox2 Ver.3.40S 以上时，可通过 RT ToolBox2 的示波器功能选择“碰撞检测等级参考值”。
由此可以实现 ColLvl 指令设定值的调整作业简易化。详细内容请参照第 472 页的「5.21 关于冲突检知功能」。
- (7) 没有正确设定抓手重量（HNDDATn 参数）和工件重量（WRKDATn 参数）的话，会有冲突检知无法正常动作的情况。请正确使用参数。
- (8) 连续功能有效时，即使将电源关闭，也会在下一电源开启时，恢复以前设定的冲突检知设定状态。
- (9) 执行程序复位、End 指令时，容许标准会被参数 COLLVL 的设定值复位。
- (10) 即使在无法使用冲突检知功能的机器人上使用本指令，也不会发生报警，会被忽视。
- (11) 对应 J7、J8 轴，冲突检知不会有作用。
- (12) 即使同一机型也会因为机器人的个体差异，而有设定值改变的情况。请在个别的机器人上做动作的确认。

【关连系统状态变量】

M_ColSts、J_ColMxl、P_ColDir

【相关指令】

ColChk (Col Check)

【关连参数】

COL、COLLVL、HNDDATn、WRKDATn

※ 详细说明请参照第 472 页的「5.21 关于冲突检知功能」请一并参照。

此外，J_ColMxl 表示自动设定冲突检知 Level 的采样程序。

Com On/Com Off/Com Stop (Communication ON/OFF/STOP)

【功能】

- Com On : 允许从通信端口的插入。
- Com Off : 禁止从通信端口的插入。
- Com Stop : 暂时停止从通信端口的插入。(数据接受)。
在下一次的 Com On 进行插入跳转。

【格式】

Com[(< 文件号码 >)] □ On

Com[(< 文件号码 >)] □ Off

Com[(< 文件号码 >)] □ Stop

【用语】

<文件号码> 在通信端口里, 记述被分配到1~3的号码。(省略时为1)

【例子】

请参照第 251 页的「[On Com GoSub \(ON Communication Go Subroutine\)](#)」。

【说明】

- (1) 执行 Com Off 后, 即使有通信也不会发生插入。
- (2) 关于文件号码, 请参照第 254 页的「[Open](#)」。
- (3) 执行 Com Stop 后, 即使有通信也不会发生插入, 但是, 因为会记忆接收数据和插入, 因此依据 Com On 中断再开始时, 直接会执行中断处理。

Def Act

【功能】

定义插入处理的插入条件和处理。插入是在程序执行中，监视输入信号等，当指定条件成立时，想要将执行优先处理时使用。

【格式】

Def □ Act □ < 优先号码 >, < 式 > □ < 处理 >[, < 种类 >]

【用语】

- < 优先号码 > 插入的优先号码。1~8的号码以常数设定。
- < 式 > 作为插入的条件，以下列的格式记述。
 <数值型数据> <比较运算符> <数值型数据> 或、
 <数值型数据> <逻辑运算符> <数值型数据>
 ※所谓<数值型数据>是指下列的数。
 <数值型常数> | <数值变量> | <数值数组> | <成份数据>
- < 处理 > 为了执行插入发生时的处理，请以GoTo文或GoSub文记述。
- < 类型 > 省略：停止种类1
 在外部速度比例100%执行时，停止在已假想的停止位置。
 在外部速度比例较小时，到停止为止的时间延长，但是经常会停止在同样位置。
 S : 停止种类2
 不依赖外部速度比例，以最短时间·最短距离减速停止。
 L : 执行完成停止
 到达目的位置为止的动作后（执行中的1行完成）后，执行插入处理。

【例子】

```

1 Def Act 1,M_In(17)=1 GoSub *L100      ' 通用输入信号 17 号的输入信号为开启状态的话，
                                         ' 定义呼出 * L100（步号 10）的子程序。
2 Def Act 2,MFG1 And MFG2 GoTo *L200   ' MFG1 和 MFG2 的逻辑与为真的话，
                                         ' 定义往 * L 200（步号 20）跳转。
3 Def Act 3,M_Timer(1)>10500 GoSub *LBL ' 经过 10.5 秒后，往 * LBL（步号 30）的子程序跳转。
:
10 *L100:M_Timer(1)=0                   ' 定时器设定在 0。
11 Act 3=1                               ' 使 Act 3 为有效。
12 Return 0
:
20 *L200: Mov P_Safe
21 End
:
30 *LBL
31 M_Timer(1)=0                         ' 将定时器返回到 0。
32 Act 3=0                               ' 使 Act 3 为无效。
32 Return 0

```

【说明】

- (1) 在中断里被呼叫出的跳转处的处理最后以 Return 指令记述。
- (2) 从插入处理以 Return 1 返回到下一个单步的情况下，请在插入处理内禁止插入。没有禁止插入的话，会直接在插入条件成立的情况下，再度执行插入处理，为了返回到下一个单步，会有单步不被执行却做了单步的事。关于中断处理请参照第 165 页的「Act」。
- (3) 中断的优先级是以 < 优先号码 > 来决定，以 1~8 的顺序，数字越小的一方优先级越高。
- (4) 插入的设定，最多可以同时 8。依据 < 优先号码 > 做区别。
- (5) 只能使用在 < 式 > 等单纯的逻辑运算或比较运算（运算符号为 1 个），括号也无法使用。

- (6) 记述相同优先号码 Def Act 指令的情况下，以后者定义的一方为有效。
- (7) 因为 Def Act 指令只能执行插入的定义，因此插入的许可 / 禁止，请依 Act 指令来指定。
- (8) 以通信插入 (Com) 优先级会比任何一个 Def Act 指令定义的插入优先顺序高。
- (9) Def Act 指令只在已定义的程序内有效。在以 CallP 指令（程序间呼叫）呼叫出的程序（子程序）内使用插入处理的情况下，必须在子程序上再度定义。
- (10) 以 Def Act 指令在 < 处理 > 中指定 GoTo 指令的情况下，发生插入的话，会变成直接在以后的程序执行中处理插入，只接受优先级高的插入。以 GoTo 指令的插入处理中，以 End 指令来解除。
- (11) 在 < 式 > 中无法以 (M1 And &H001)=1 的逻辑运算的组合条件来记述。
- (12) 在圆或圆弧插补 (Mvc.Mvr.Mvr2,Mvr3) 的执行中，有插入条件，以 Return 0 返回到原本的以单步控制的情况下，机器人会返回到圆或圆弧的起点，再次执行圆、圆弧插补。
- (13) 在弧形插补的执行中，有插入，以 Return 0 返回到原本的情况。机器人会从那时的位置开始执行弧形插补。



注意

插入的停止时间，请依用途做正确的设定。
 机器人执行移动指令的时候，想依据插入在最短时间、最短距离将机器人停止的话，请将停止种类设定为“S”。

机器人在以动作指令的移动中，插入条件成立的时候，执行程序停止的 3 种类概念图如图 4-18 所示。

停止种类	外部速度比例 100% (最高速)	外部速度比例 50%
停止种类 1 (省略时) S1=S2		
停止种类 2 (S)		
执行完成停止 (L) S3=S4		

图 4-18: 停止时的概念图

【相关指令】
 Act

Def Arch

【功能】

定义 Mva 指令的 Arch 运动动作的弧形形状。

【格式】

```
Def Arch □ <弧形号码>, [<上升移动量>], [<下降移动量>]
          , [<下降移动量>], [<下降待避量>], [<插补形式>]
          , [<插补种类 1>, <插补种类 2>]
```

【用语】

<弧形号码> Arch 运动动作模式的号码。将1~4为止的号码以常数或变量设定。

<上升移动量>

<下降移动量> 如右图可以常数或变量指定

<上升待避量>

<下降待避量>

<插补形式>

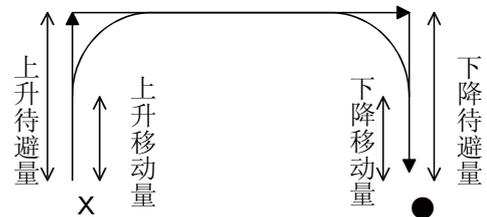
上升及下降的动作补插补式
直线/关节=1/0

<补插补类1>

绕远/走进路 =1/0

<补插补类2>

3轴直交/等量旋转 =1/0



省略弧形号码以外的情况下，会采用初始值。

初始值以下列 参数设定。值可修正。

参数名	弧形号码	上升移动量 (mm)	下降移动量 (mm)	上升待避量 (mm)	下降待避量 (mm)
ARCH1S	1	0.0	0.0	30.0	30.0
ARCH2S	2	10.0	10.0	30.0	30.0
ARCH3S	3	20.0	20.0	30.0	30.0
ARCH4S	4	30.0	30.0	30.0	30.0

垂直多关节机器人 (RV-F 系列)

参数名	弧形号码	插补形式	插补种类 1	插补种类 2
ARCH1T	1	1	0	0
ARCH2T	2	1	0	0
ARCH3T	3	1	0	0
ARCH4T	4	1	0	0

水平多关节机器人 (RH-F 系列)

参数名	弧形号码	插补形式	插补种类 1	插补种类 2
ARCH1T	1	0	0	0
ARCH2T	2	0	0	0
ARCH3T	3	0	0	0
ARCH4T	4	0	0	0

【例子】

1 Def Arch 1,5,5,20,20

2 Mva P1,1

3 Dly 0.3

4 Mva P2,2

5 Dly 0.3

' 以在步号 1 所定义的形状，用 Arch 运动动作执行。

' 没有对应的 Def Arch 的情况下，以参数的初始值动作。

【说明】

- (1) 没有 Def Arch 指令，执行 Mva 指令的情况下，以在参数设定的弧形形状动作。
- (2) 在程序内，变更上升量等的时候使用。



注意

机器人根据当时的指定速度，可能会以不同的轨迹动作。尤其是转角部分，其内环距离有可能会发生变化，因此，在最初开始自动运行时应先从低速运行，然后一边留意与周边装置之间的干涉，一边逐渐提升运行速度。

【相关指令】

Mva (Move Arch)、Accel、Ovrd、
Mvs (Move S)(参照插补类 1、2)

Def Char (Define Character)

【功能】

定义字符串变量。在变量名的前面文字使用 "C"(或 "c") 以外的文字的时候。开头加上 "C" (或 "c") 的变量，无需再以 Def Char 定义。

【格式】

```
Def □ Char □ < 字符串变量名 > [, < 字符串变量名 >]...
```

【用语】

<字符串变量名> 指定变量名。

【例子】

1 Def Char MESSAGE	' 将「MESSAGE」视为字符串变量定义
2 MESSAGE = "WORKSET"	' 在 MESSAGE 变量里代入 "WORKSET"。
3 CMSG = "ABC"	' 在 CMSG 变量里代入 "ABC"。以 C 开头的变量不要加「DefChar」

【说明】

- (1) 变量名的文字数最大为 16 个文字。关于可以使用的文字请参照第 136 页的「4.4.6 可以在程序内使用的文字种类」。
- (2) 指定复数变量名的情况下，1 行里可以记述文字的最大值 (含指令 240 个文字)。
- (3) 在使用者基本程序登录的变量名 "C"(或 "c") 的后面加上 "_" 的话，会变成使用者定义全局变量 (在程序间共通的变量)。
变成使用此变量的情况下，请务必以本指令定义。
详细请参照第 145 页的「4.4.24 使用者定义全局变量」。

Def FN (Define Function)

【功能】

定义任意的函数。

【格式】

Def □ FN< 识别文字 >< 名称 >[(< 假设自变量 >[, < 假设自变量 >]...]=< 函数的定义式 >

【用语】

<识别文字> 识别文字有下列4种类。

数值型 : M或m

字符串型: C或c

位置型 : P或p

关节型 : J或j

<名称> 可以任意的字符串记述。(最大5个文字)

<假设自变量> 函数被呼叫出来时,为交给函数的值。能记述全部的变量,最大可以记述16个变量。

<函数的定义式> 记述为了作为函数要进行运算的式。

【例子】

```
1 Def FNMAve(ma,mb)=(ma+mb)/2 '定义 FNMAve 以求得 2 个数的平均。
2 MDATA1=20
3 MDATA2=30
4 MAVE=FNMAve(MDATA1,MDATA2) '将 20 和 30 的平均值 25 代入数值变量 MAVE。
5 Def FNpAdd(PA,PB)=PA+PB '位置型的加算
6 P10=FNpAdd(P1,P2)
```

【说明】

(1) FN+< 名称 > 会变成函数名。函数名含 FN 最大可到 16 个文字 (不包含 \$)。

例) 数值型 ...FNMMAX 识别文字: M

字符串型 ...FNCAMES\$ 识别文字: C(在语尾以 \$ 记述)

(2) 将用 Def FN 定义的函数称为使用者定义函数。函数的定义式,以一行可以记述为范围。

(3) 在函数的定义式中,可以记述编入函数及已定义完成的使用者定义函数。此情况下,可以记述 16 段为止 level 的使用者定义函数。

(4) 在函数的定义式中所使用的变量,在假设自变量中没有的情况下,此变量则使用以那个时点所持有的值。此外,执行时变量的类型(数值或字符串)或数和已声明的型或数不同的情况下,会发生报警。

(5) 使用者定义函数只在已定义的程序里有。无法使用在以 Call P 指令呼叫出的子程序中。

Def Float/Def Double/Def Inte/Def Long

【功能】

声明数值变量。 Inte 为整数、 Long 为长精度整数型、 Float 为单精度实数、 Double 为双精度实数。

【格式】

```
Def □ Inte □ <数值变量名> [, <数值变量名>]...
Def □ Long □ <数值变量名> [, <数值变量名>]...
Def □ Float □ <数值变量名> [, <数值变量名>]...
Def □ Double □ <数值变量名> [, <数值变量名>]...
```

【用语】

<数值变量名> 指定变量名。

【例子】

① 整数型变量的定义

```
1 Def Inte WORK1, WORK2 '将 WORK1、 WORK2 视为数值变量名执行声明
2 WORK1 = 100           '往 WORK1 的代入 100
3 WORK2 = 10.562       'WORK2 的值变成 11
4 WORK2 = 10.12        'WORK2 的值变成 10
```

② 长精度整数型变量的定义

```
1 Def Long WORK3
2 WORK3 = 12345
```

③ 单精度型实数的定义

```
1 Def Float WORK4
2 WORK3 = 123.468      'WORK4 变成 123.468000。
```

④ 双精度型实数的定义

```
1 Def Double WORK5'
2 WORK4 = 100/3       'WORK5 变成 33.333332061767599。
```

【说明】

- (1) 变量名的文字数最大为 16 个文字。关于可以使用的文字请参照第 136 页的「4.4.6 可以在程序内使用的文字种类」。
- (2) 指定复数变量名的情况下，1 行里可以记述文字的最大值（含指令 240 个文字）。
- (3) 以 Inte 定义的变量会变成整数型。（-32768 ~ +32767）
- (4) 以 Long 定义的变量会变成长精度整数型。（-2147483648 ~ 2147483647）
- (5) 以 Float 定义的变量会变成单精度整型。（±3.40282347e+38）
- (6) 以 Double 定义的变量会变成双精度型。（±1.7976931348623157e+308）

Def IO (Define IO)

【功能】

定义输入输出变量。想要设定特定的输入输出信号的位宽时使用。

1 位、8 位、16 位的输入输出信号可以在标准变量使用。

(1 位： M_In, M_Out/8 位： M_Inb, M_Outb/16 位： M_Inw, M_Outw)

本指令在已声明的变量里无法参照输出信号，请特别注意。

【格式】

```
Def □ IO □ < 输入输出变量名 > = < 指定型 >, < 输入输出信号号码 >
                                     [, < Mask 情报 >]
```

【用语】

< 输入输出变量名 > 指定变量名。
 < 指定型 > 指定位(1bit)、字节组(8bit)、字符(16bit)的其中一个
 < 输入输出信号号码 > 指定输入信号(参照时)或输出信号号码(代入时)。
 < Mask 情报 > 只在特定的信号有效时指定。

【例子】

① 将 PORT1 的输入变量，以位型分配到输入输出信号 6

```
1 Def IO PORT1 = Bit,6
  :
10 PORT1 = 1        ' 输出信号号码 6 开启。
  :
20 PORT1 = 0        ' 输出信号号码 6 关闭。(因为 2 的最低位为 0)
21 M1 = PORT1      ' 输入信号号码 6 的状态被设定在 M 1
```

② 将 PORT2 的输入输出变量，以数元组分配到输入输出信号号码 5，

Mask 情报指定为 16 进制 0F。

```
1 Def IO PORT2 = Byte, 5, &H0F        '
  :
10 PORT2 = &HFF                        ' 输出信号 5 - 8 为开启。
  :
20 M2 = PORT2                            ' 输入信号 5 - 8 的值代入变量 M2。
```

③ 将 PORT3 的输入变量，以字符型分配到输入输出信号号码 8，

Mask 情报指定为 16 进制 0FFF。

```
1 Def IO PORT3 = Word, 8, &H0FFF      '
  :
10 PORT3 = 9                            ' 输出信号 8 和 11 为开启。
  :
20 M3 = PORT3                            ' 输入信号 8 -19 的值代入变量 M3。
```

【说明】

- (1) 参照此变量的时候， 会看到输入信号。
- (2) 代入此变量的时候， 写在输出信号上。
- (3) 无法以此变量参照输出信号。参照输出信号的情况下，请在 M_Out 变量参照。
- (4) 变量名的文字数最大为 16 个文字。关于可以使用的文字请参照第 136 页的「4.4.6 可以在程序内使用的文字种类」。
- (5) 无法用此变量执行脉冲输出。脉冲输出请使用 M_Out。

(6) 指定掩码 (Mask) 情报的情况下，只对特定的信号有效。

例) 上述第 20 行的例子，因为将 8 位宽的输出数据以 16 进制 OF 指定掩码，以后使用 PORT2 的话：

- 作为输入信号使用的情况 (M1=PORT2)

输入 5 ~ 8 号，9 ~ 12 号经常视为 0。

12 5 号 (输入信号号码)

0 0 0 0 1 1 1 1

无效 有效

- 作为输出信号使用的情况 (PORT2=M1)

在 5 ~ 8 号输出这次输入的数据，9 ~ 12 号则保持现在已输出的状态。

12 5 号 (输出信号号码)

**** 1 1 1 1

↑

↑

保持现在输出状态 这次的输出数据

Def Jnt (Define Joint)

【功能】

定义关节型位置变量。在变量名的前面文字使用 "J" (或 "j") 以外的文字的时候。开头加上 "J" (或 "j") 的变量，无需再以 Def Jnt 定义。

【格式】

Def □ Jnt □ < 关节变量名 > [, < 关节变量名 >]...

【用语】

< 关节变量名 > 指定变量名。

【例子】

```
1 Def Jnt SAFE    ' 将「SAFE」视为关节变量定义
2 Mov J1          ' 在以 J 开始的关节型位置变量，不要定义「Def Jnt」
3 退避点 = (-50,120,30,300,0,0,0)
4 Mov SAFE        ' 往退避点移动
```

【说明】

- (1) 在开头文字为 "J" (或 "j") 以外，定义关节位置变量的时候使用。
- (2) 变量名的文字数最大为 16 个文字。关于可以使用的文字请参照第 136 页的「[4.4.6 可以在程序内使用的文字种类](#)」。指定复数变量名的情况下，1 行里可以记述文字的最大值 (含指令 240 个文字)。
- (3) 在使用者基本程序登录的变量名 "J" (或 "j") 的后面加上 "_" 的话，会变成使用者定义全局变量 (在程序间共通的变量)。变成使用此变量的情况下，请务必以本指令定义。详细请参照第 145 页的「[4.4.24 使用者定义全局变量](#)」。

Def Plt (Define 码垛)

【功能】

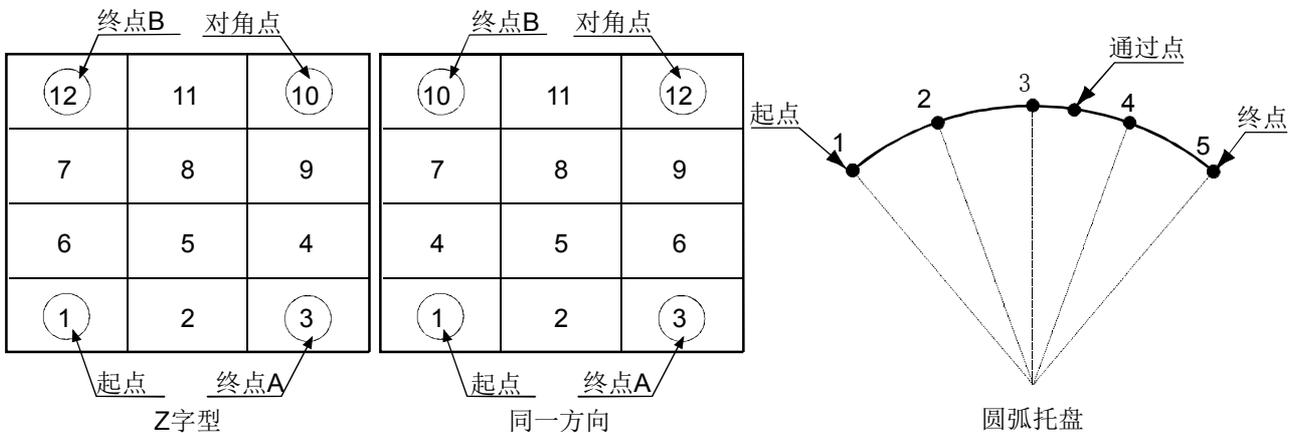
定义码垛。（3点码垛 / 4点码垛 / 圆弧码垛）

【格式】

```
Def □ Plt □ <码垛号码>, <起点>, <终点A>, <终点B>,
      [<对角点>], <个数A>, <个数B>, <码垛模版>
```

【用语】

- <码垛号码> 选择已设定的码垛号码（只有1~8的常数）。
- <起点> 码垛的起点。只有位置常数或位置变量。
- <终点A> 码垛一边的终点。圆弧码垛时为圆弧的通过点。只有位置常数或位置变量。
- <终点B> 码垛另一边的终点。圆弧码垛时为圆弧的终点。只有位置常数或位置变量。
- <对角点> 码垛起点的对角点。圆弧码垛时无意义。
- <个数A> 码垛起点终点间的工作个数。
- <个数B> 码垛时，码垛的起点和圆弧的终点间的工件个数。可以是变量。
- <码垛模版> 码垛的起点和终点8间的的工件个数。圆弧码垛时无意义。（必须指定1等）。可以是变量。
- 记述在被分配格子点，加上号码的码垛模版及姿势的固定]等分配。可以是变量。
- 1: Z字型（姿势等分割） 2: 同一方向（姿势等分割） 3: 圆弧码垛（姿势等分割）
 11: Z字型（姿势固定） 12: 同一方向（姿势固定） 13: 圆弧码垛（姿势固定）



【例子】

- 1 Def Plt 1,P1,P2,P3, ,3,4,1 '3点的码垛定义
 2 Def Plt 1,P1,P2,P3,P4,3,4,1 '4点的码垛定义

【说明】

- (1) 位置运算时的精度，4点码垛比3点码垛3点的精度更上升。
- (2) 只有在已执行的程序内有效。在以 CallP 指令呼叫出来的子程序中，无法使用已定义的码垛情报。使用在子程序的时候，请在子程序里再次定义。
- (3) 个数 A、B 视为非 0 的正数，若为 0 或者是负的数的情况下，会发生报警。
- (4) 个数 AX 个数 B 若超过 32767 的情况下，执行时发生报警。

- (5) 圆弧码垛的情况下，个数 **B** 的值并没有意义，但是因为无法省略，所以请设定为 0 等的值。此外，即使对角点有指定，也没有意义。
- (6) 抓手向下的情况下，请使起点、终点 **A**、终点 **B**、对角点的 **ABC** 轴的值的符号相同。抓手向下的情况，会变成 **A=180**(或 **-180**)、**B=0**、**C=180** (或 **-180**)，3 点的 **A** 轴、**C** 轴的符号不一致的情况下，在途中的位置，会有抓手旋转的情况发生。在这种情况下，请在示教单元的位置编辑画面将符号修正变为相同。**+180** 和 **-180** 会成为相同姿势，即使修正符号也不会有问题。
- (7) 在码垛模版指定 **11 ~ 13** 的话，以码垛运算求得的位置变量的姿势数据会被代入〈起点〉的姿势。指定 **1 ~ 3** 情况的姿势，会代入将〈起点〉 - 〈终点〉之间等分割的值。
- (8) **J1** 轴、**J2** 轴或 **J4** 轴可超过 ± 180 度的机器人机型中，无法指定 **J1** 轴、**J2** 轴或 **J4** 轴的关节角度跨 ± 180 度之类的码垛。如果进行了指定，则执行本指令时将发生 **L3750** 错误。此时，请将码垛分割使用。详细内容请参照第 109 页的「[4.1.2Pallet 运算](#)」。
- (9) 执行本指令时，对指定的起点相对的终点 **A**、终点 **B** 以及对角点（可省略）的位置关系构成的四角形是否在机器人的动作范围内进行检查。执行时发生报警 **L.3110** 时，请确认上述四角形在动作范围内。
- (10) 作为一系列码垛使用时，请将起点与终点 **B** 定义为同一点（或起点与终点 **A** 为同一点）。

⚠ 注意

将姿势成份 (**A, B, C**) ± 180 度附近的位置数据，作为码垛定义的〈起点〉、〈终点 **A, B**〉、〈对角点〉的时候，在位置数据的相同姿势成份里有不同符号混合的话，抓手的旋转会出现预期外的动作。将姿势成份使用在 ± 180 度附近的位置数据的情况下，请先确认第 109 页的「[4.1.2Pallet 运算](#)」的〈关于以码垛定义位置数据的姿势注意事项〉后，再使用。

⚠ 注意

依据码垛运算 (**Plt** 指令) 所算出的格子点的构造标志 (位置数据的 **FL1**)，采用码垛定义的起点值。因此，在码垛定义的各点使用构造标志不一样的位置数据的话，会变成和要求的动作不一样的动作。在码垛定义的起点、终点 **A**、**B**、对角点，请使用构造标志的值全部相同的位置数据。此外，关于格子点的多旋转标志 (位置数据的 **FL2**) 也一样，采用码垛定义的起点值。在码垛定义的各点，使用多旋转标志的不同数据的话，依据在码垛动作经由机器人的位置和插补指令的种类 (关节插补、直线插补等)，抓手的旋转会出现预期外的动作。在此情况下，使用插补指令的自变量类型，将姿势的绕道 / 抄近路等设定适当，将码垛调整成所要求的动作。

程序例请参照第 109 页的「[4.1.2Pallet 运算](#)」。

【相关指令】

Plt (码垛)

◇◆◇关于码垛的姿势◆◆◇

为了将码垛的格子点姿势设定为在平常是相同的方向，因此，请将指令的自变量的〈码垛 return〉设定在 **11 ~ 13**。使在〈起点〉指定的姿势相同。

Def Pos (Define Position)

【功能】

定义直角型位置变量。在变量名的前面文字使用 "P"（或 "p"）外的文字的时候。开头加上 "P"（或 "p"）的变量，无需再以 Def Pos 定义。

【格式】

```
Def □ Pos □ <位置变量名> [, <位置变量名>] ...
```

【用语】

<位置变量名>指定变量名。

【例子】

1 Def Pos WORKSET	' 将 「WORKSET」 作为直角型位置变量定义
2 Mov P1	' 以 P 开始的直角型位置变量不要定义 「Def Pos」
3 WORKSET=(250,460,100,0,0,-90,0,0)(0,0)	
4 Mov WORKSET	' 往 WORKSET 移动。

【说明】

- (1) 在开头文字为 "P"（或 "p"）以外，为定义直角型位置变量的时候使用。
- (2) 变量名的文字数最大为 16 个文字。关于可以使用的文字请参照第 136 页的「[4.4.6 可以在程序内使用的文字种类](#)」。
- (3) 指定复数变量名的情况下，1 行里可以记述文字的最大值（含指令 240 个文字）。
- (4) 在使用者基本程序登录的变量名 "P"（或 "p"）的后面加上 "_" 的话，会变成使用者定义全局变量（在程序间共通的变量）。变成使用此变量的情况下，请务必以本指令定义。详细请参照第 145 页的「[4.4.24 使用者定义全局变量](#)」。

Dim

【功能】

定义数组。可定义到 3 次元为止的排列。

【格式】

```
Dim □ <变量名> ( (<要素数> [, <要素数> [, <要素数>] ] )
    [, <变量名> ( (<要素数> [, <要素数> [, <要素数>] ] ) ) ] ...
```

【用语】

<变量名> 记述数组的名称。
<要素数> 将数组的要素的个数以常数记述。

【例子】

1 Dim PDATA(10)	' 定义持有 10 个要素的位置排列数组变量 PDATA。
2 Dim MDATA#(5)	' 定义持有 5 个要素的双精度实数型数组 MDATA#
3 Dim M1%(6)	' 定义持有 6 个要素的整数型数组 M1%
4 Dim M2!(4)	' 定义持有 4 个要素的单精度实数型数组 M2!
5 Dim M3&(5)	' 定义持有 5 个要素的长精度整数型数组 M3&
6 Dim CMOJI(7)	' 定义持有 7 个要素的字符串型变量 CMOJI
7 Dim MD6(2,3), PD1(5,5)	' 定义持有 2 X 3 的要素的 2 次元单精度实数型变量 MDATA ' 定义持有 5 X 5 的要素 2 次元的配置数组变量 PD1

【说明】

- (1) 可定义 1 次元排列、2 次元排列、3 次元排列。
- (2) 数值型变量的情况，依据在变量名上附加变量型表示记号，可以将整数型、长精度整数型、单精度实数型、双精度实数的变量分开使用。省略变量的类型的情况下，会变成单精度实数型变量。
Dim MABC(10) ' 定义持有 10 个要素的单精度实数型数组 MABC
- (3) 实际使用数组的时候，要素号码从 1 开始。在例子的步号 1 的 PDATA，要素号码变成 1 ~ 10。
- (4) <要素数> 以 1 ~ 999 的数值常数记述。无法使用数值表达式。要素数以实数指定的情况下，小数部位会四舍五入变成整数。
另外，依据系统内存的空间容量，有可能出现无法确保指定要素数的排列的情况。
此情况下，在行登录时，会发生报警。
- (5) 使用比已定义的要素数大的要素号码时，执行时会发生报警。
- (6) 数组在被定义的时点，变量值不定。
- (7) 使用数组的情况下，请务必以 Dim 指令定义。
- (8) 已使用 Dim 指令定义的排列，只在已定义的程序里有效。
在以 CallP 指令呼叫出的子程序中使用，必须再定义。
- (9) 数组可以和普通的变量一样使用。
但是，变量名及要素号码指定的文字数为超过 8 文字的变量，会无法在示教单元的监视变量画面、位置编辑画面里使用，请特别注意。
- (10) 在使用者基本程序登录的变量名的第 2 个字加上 "_" 的话，会变成使用者定义全局变量（在程序间共通的变量）。变成使用此变量的情况下，请务必以本指令定义。
详细请参照第 145 页的「4.4.24 使用者定义全局变量」。

Dly (Delay)

【功能】

① 单独指令的情况

指定的时间进行时间等待。在决定机器人的定位时使用，在取得输入输出信号的时序（Timing）时使用。

② 作为信号输出用的脉冲输出用，附随使用的情况下，会显示脉冲输出的输出时间。

【格式】

① 单独指令的情况

```
Dly □ <时间>
```

② 为脉冲输出用被附随使用情况

```
例) M_Out(1)=1 Dly □ <时间>
```

【用语】

<时间> 等待时间或脉冲输出时间可以用常数或变量来记述。单位：[秒]

最小值可以从0.01秒开始设定。0.00也可以设定。

最大值可以指定到单精度实数的最大值。

【例子】

① 等待时间

1 Dly 30 ' 执行 30 秒的等待时间。

② 信号的脉冲输出

1 M_Out(17)=1 Dly 0.5 ' 0.5 秒间，在通用输出信号 17 号执行信号输出（脉冲输出）。

2 M_Out(18)=1 Dly 0.5 ' 0.5 秒间，在通用输出信号 18 号执行信号输出（脉冲输出）。
经过 0.5 秒后，在 19-25 号，信号被输出。

③ 等待定位完成

1 Mov P1 ' 往 P1 移动

2 Dly 0.1 ' 往 P1 的定位。

④ 等待抓手开（闭）完成

1 HOpen 1 ' 打开抓手 1

2 Dly 0.5 ' 等待抓手 1 打开完毕

【说明】

(1) 在程序上设定等待时间。取得输入输出信号的时序，使用在动作指令的定位，记述在信号输出文中，（上记【例子】的 2 等），可以使用在设定脉冲输出时间的时候。

(2) 脉冲输出不会等待指定时间经过，会和下一个指令被并列执行。

(3) 脉冲输出可以在全部程序同时最多输出 50 个。超过的情况下，在执行时会出现报警。

(4) 脉冲输出在指定时间经过后，位会反转。因此，使用 M_Outb（8 位）和 M_Outw（16 位）的情况下，其位数量会反转。

(5) 脉冲输出在指定时间中，End 指令和程序的最终行被执行的情况下，在不等待时间经过下，结束程序的执行。但是，脉冲输出会在指定时间继续后关闭。

(6) 其他插入的优先关系如下所示：

Com > Act > Wthlf(Wth) > 脉冲输出（有设定时间）

(7) 在脉冲输出实行中，即使停止输入的情况下，脉冲输出的动作也不会停止。

注 1) 在下列程序里，即使在以步号 2 停止输入的情况下，也会保持输出信号状态，并停止执行。

```
1 M_Out(17)=1
```

```
2 Dly 10
```

```
3 M_Out(17)=0
```

注 2) 使用 M_Outb（8 位），M_Outw（16 位）执行脉冲输出的情况下，在指定时间后，以各个位宽反转。

M_Outb(1)=1 Dly 1.0 在此情况下，输出 00000001，且在 1 秒后输出 11111110。

EMvc (E Move C)

【功能】

按照始点、通过点 1、通过点 2、始点的顺序，沿着工件坐标系（Ex-T 坐标系）进行三维圆形插补动作。
（关于功能概要，请参照“第 652 页的「7.5 关于 Ex-T 控制」”。）

【格式】

EMvc □ < 工件坐标号码 >, < 始点 >, < 通过点 1 >, < 通过点 2 > [□ < 附带条件 >]

【用语】

- < 工件坐标号码 > 指定 1~8 的工件坐标号码。
- < 始点 > 圆形的始点及终点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。
- < 通过点 1 > 圆弧的通过点 1。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。
- < 通过点 2 > 圆弧的通过点 2。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。
- < 附带条件 > 对 Wth 节或 Wthlf 节进行记述。

【例文】

- ① 按照 P1、P2、P3、P1 的顺序沿着工件坐标 1 进行圆形插补移动
1 EMvc 1, P1, P2, P3
- ② 按照 P1、J2、P3、P1 的顺序沿着工件坐标 2 进行圆形插补移动 (使用关节变量的示例)
2 EMvc 2, P1, J2, P3
- ③ 按照 P1、P2、P3、P1 的顺序沿着工件坐标 1 进行圆形插补移动的同时，将输出信号 17 置为 ON
3 EMvs 2,P1 Wth M_Out(17)=1
- ④ 按照 P3、(Plt 1.5)、P4、P3 的顺序沿着工件坐标 4 进行圆形插补移动，在该移动过程中输入信号 20 为 ON 后，立即将输出信号 21 置为 ON
4 EMvc 4, P3, (Plt 1, 5), P4 Wthlf M_In(20)=1, M_Out(21)=1

【说明】

- (1) Ex-T 控制圆形插补，从所给的 3 点求出沿着指定工件坐标的圆形后，在圆周上移动。（360 度）
- (2) Ex-T 控制圆形插补中的姿势维持始点的姿势。不考虑通过点 1、2 的姿势。
- (3) 当前位置与始点不一致时，自动进行 Ex-T 控制直线插补移动至始点，之后进行 Ex-T 控制圆形插补。
- (4) 中断执行，通过 JOG 移动后重新开始执行时，移动至中断执行的位置后，重新开始执行剩余的圆弧插补。返回中断位置的插补方法（关节插补 / 直线插补）可通过 RETPATH 参数进行更改（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。
- (5) ALWAYS、ERROR 属性的程序中无法执行。（会发生 L3287 错误。）
- (6) 未设定工件坐标系的情况下，该初始值全部组成成分为 0(0, 0, 0, 0, 0, 0)。该状态下，如执行 Ex-T 控制直线插补，则变为沿着世界坐标系原点动作。
- (7) 附加了附加轴时，附加轴也会动作。但是，Ex-T 控制中不包含附加轴。因此，移动行走轴等机器人本体之类的情况下，设想工件坐标也会与机器人一起移动，并偏离本来的作业位置。因此，不建议通过本指令执行附加轴动作。此外，使用本指令时，请勿使用附加轴同步控制。

【相关指令】

EMvr (E Move R)、EMvr2 (E Move R2)、EMvr3 (E Move R3)、EMvs (E Move S)

【相关系统状态变量】

P_WkCord (工件坐标数据)

【相关参数】

WK1CORD ~ WK8CORD

EMvr (E Move R)

【功能】

从始点经由通过点至终点为止，沿着工件坐标系（Ex-T 坐标系）进行三维圆弧插补动作。
（关于功能概要，请参照“第 652 页的「7.5 关于 Ex-T 控制」”。）

【格式】

EMvr □ < 工件坐标号码 >, < 始点 >, < 通过点 >, < 终点 > [□ Type □ < 常数 1 >, < 常数 2 >]
[□ < 附带条件 >]

【用语】

< 工件坐标号码 > 指定 1~8 的工件坐标号码。

< 始点 > 圆弧的始点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。

< 通过点 > 圆弧的通过点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。

< 终点 > 圆弧的终点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。

< 常数 1 > 姿势旋转的指定。抄近道动作 = 0。省略时为 0。

< 常数 2 > 姿势插补方法的指定。等效旋转/三轴直交 = 0/1。省略时为 0。

< 附带条件 > 对 Wth 节或 Wthlf 节进行记述。

【例文】

① 从 P1 经由 P2 沿着工件坐标 1 向 P3 进行圆弧插补移动

1 EMvc 1, P1, P2, P3

② 从 P1 经由 J2 沿着工件坐标 2 向 P3 进行圆弧插补移动 (使用关节变量的示例)

2 EMvc 2, P1, J2, P3

③ 从 P1 经由 P2 沿着工件坐标 1 向 P3 进行圆弧插补移动的同时，将输出信号 17 置为 ON

3 EMvr 2, P1, P2, P3 Wth M_Out(17)=1

④ 从 P3 经由 (Plt 1,5) 沿着工件坐标 4 向 P4 进行圆弧插补移动，在该移动过程中输入信号 20 为 ON 后，立即将输出信号 21 置为 ON

4 EMvr 4, P3, (Plt 1, 5), P4 Wthlf M_In(20)=1, M_Out(21)=1

【说明】

- (1) Ex-T 控制圆弧插补，从所给的 3 点求出沿着指定工件坐标的圆弧后，在该圆弧上移动。
- (2) 姿势为从始点至终点的插补，不影响通过点的姿势。
- (3) 当前位置与始点不一致时，自动进行 Ex-T 控制直线插补移动至始点，之后进行圆弧插补。
- (4) 中断执行，通过 JOG 移动后重新开始执行时，移动至中断执行的位置后，重新开始执行剩余的圆弧插补。返回中断位置的插补方法 (关节插补 / 直线插补) 可通过 RETPATH 参数进行更改 (请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」)。
- (5) 等效旋转 (常数 2=0) 的情况下，如果始点与终点的结构标志不相同，则执行时会发生错误。
- (6) 指定的 3 点中有同一位置的情况下，或 3 点位于一条直线上的情况下，将从始点向终点进行 Ex-T 控制直线插补动作。不发生错误。
- (7) 在常数 2 中指定了三轴直交时，常数 1 为无效，将移动为与示教同样的姿势。
- (8) ALWAYS、ERROR 属性的程序中无法执行。(会发生 L3287 错误。)
- (9) 未设定工件坐标系的情况下，该初始值全部组成成分为 0(0, 0, 0, 0, 0, 0)。该状态下，如执行 Ex-T 控制直线插补，则变为沿着世界坐标系原点动作。
- (10) 附加了附加轴时，附加轴也会动作。但是，Ex-T 控制中不包含附加轴。因此，移动行走轴等机器人本体之类的情况下，设想工件坐标也会与机器人一起移动，并偏离本来的作业位置。因此，不建议通过本指令执行附加轴动作。此外，使用本指令时，请勿使用附加轴同步控制。
- (11) 常数 2 中指定了三轴直交时，动作与通过 Mvr 指令指定了三轴直交时相同。

【相关指令】

EMvc (E Move C)、EMvr2 (E Move R2)、EMvr3 (E Move R3)、EMvs (E Move S)

【相关系统状态变量】

P_WkCord (工件坐标数据)

【相关参数】

WK1CORD ~ WK8CORD

EMvr2 (E Move R2)

【功能】

在由始点、终点、参考点构成的圆弧上，从始点至终点，沿着工件坐标系（Ex-T 坐标系）进行三维圆弧插补动作。移动方向为不通过参考点的方向。

（关于功能概要，请参照“第 652 页的 [7.5 关于 Ex-T 控制]”。

【格式】

```
EMvr2 □ < 工件坐标号码 >,< 始点 >,< 终点 >,< 参考点 >[ □ Type □ < 常数 1 >,< 常数 2 >]
      [ □ < 附带条件 >]
```

【用语】

<工件坐标号码> 指定1~8的工件坐标号码。

<始点> 圆弧的始点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。

<终点> 圆弧的终点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。

<参考点> 圆弧的参考点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。

<常数1> 姿势旋转的指定。抄近道动作 = 0。省略时为0。

<常数2> 姿势插补方法的指定。等效旋转/三轴直交 = 0/1。省略时为0。

<附带条件> 对Wth节或Wthlf节进行记述。

【例文】

1 EMvr2 1, P1, P2, P3

2 EMvr2 2, P1, J2, P3

3 EMvr2 2, P1, P2, P3 Wth M_Out(17)=1

4 EMvr2 4, P3, (Plt 1, 5), P4 Wthlf M_In(20)=1, M_Out(21)=1

【说明】

- (1) 圆弧插补动作，从所给的 3 点求出圆弧后，在该圆弧上移动。
- (2) 姿势为从始点至终点的插补，不影响参考点的姿势。
- (3) 当前位置与始点不一致时，自动进行直线插补（三轴直交插补）至始点。
- (4) 中断执行，通过 JOG 移动后重新开始执行时，通过关节插补移动至中断执行的位置后，重新开始执行剩余的圆弧插补。返回中断位置的插补方法（关节插补 / 直线插补）可通过 RETPATH 参数进行更改（请参照第 442 页的 [5.10 中断时的 JOG 前进后的自动回归设定]）。
- (5) 移动方向为不通过参考点的方向。
- (6) 等效旋转（常数 2=0）的情况下，如果始点与终点的结构标志不相同，则执行时会发生错误。
- (7) 指定的 3 点中有同一位置的情况下，或 3 点位于一条直线上的情况下，将从始点向终点进行直线插补动作。不发生错误。
- (8) 在常数 2 中指定了三轴直交时，常数 1 为无效，将移动为与示教同样的姿势。
- (9) ALWAYS、ERROR 属性的程序中无法执行。（会发生 L3287 错误。）
- (10) 未设定工件坐标系的情况下，该初始值全部组成成分为 0(0, 0, 0, 0, 0, 0)。该状态下，如执行 Ex-T 控制直线插补，则变为沿着世界坐标系原点动作。
- (11) 附加了附加轴时，附加轴也会动作。但是，Ex-T 控制中不包含附加轴。因此，移动行走轴等机器人本体之类的情况下，设想工件坐标也会与机器人一起移动，并偏离本来的作业位置。因此，不建议通过本指令执行附加轴动作。此外，使用本指令时，请勿使用附加轴同步控制。
- (12) 常数 2 中指定了三轴直交时，动作与通过 Mvr 指令指定了三轴直交时相同。

【相关指令】

EMvc (E Move C)、EMvr (E Move R)、EMvr3 (E Move R3)、EMvs (E Move S)

【相关系统状态变量】

P_WkCord（工件坐标数据）

【相关参数】

WK1CORD ~ WK8CORD

EMvr3 (E Move R3)

【功能】

在由始点、终点、中心点构成的圆弧上，从始点至终点，沿着工件坐标进行三维圆弧插补动作。
 （关于功能概要，请参照“第 652 页的「7.5 关于 Ex-T 控制」”。）

【格式】

EMvr3 < 工件坐标号码 >,< 始点 >,< 终点 >,< 中心点 >[Type < 常数 1 >,< 常数 2 >]
 [< 附带条件 >]

【用语】

- < 工件坐标号码 > 指定 1~8 的工件坐标号码。
- < 始点 > 圆弧的始点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。
- < 终点 > 圆弧的终点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。
- < 中心点 > 圆弧的中心点。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。
- < 常数 1 > 姿势旋转的指定。抄近道动作 = 0。省略时为 0。
- < 常数 2 > 姿势插补方法的指定。等效旋转/三轴直交 = 0/1。省略时为 0。
- < 附带条件 > 对 Wth 节或 Wthlf 节进行记述。

【例文】

- 1 EMvr3 1, P1, P2, P3
- 2 EMvr3 2, P1, J2, P3
- 3 EMvr3 2, P1, P2, P3 Wth M_Out(17)=1
- 4 EMvr3 4, P3, (Plt 1, 5), P4 Wthlf M_In(20)=1, M_Out(21)=1

【说明】

- (1) 圆弧插补动作，从所给的 3 点求出圆弧后，在该圆弧上移动。
- (2) 姿势为从始点至终点的插补，不影响中心点的姿势。
- (3) 当前位置与始点不一致时，自动进行直线插补（三轴直交插补）至始点。
- (4) 中断执行，通过 JOG 移动后重新开始执行时，通过关节插补移动至中断执行的位置后，重新开始执行剩余的圆弧插补。返回中断位置的插补方法（关节插补 / 直线插补）可通过 RETPATH 参数进行更改（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。
- (5) 等效旋转（常数 2=0）的情况下，如果始点与终点的结构标志不相同，则执行时会发生错误。
- (6) 在常数 2 中指定了三轴直交时，常数 1 为无效，将移动为与示教同样的姿势。
- (7) 从始点至终点的中心角为 $0 < \text{中心角} < 180$ 度。
- (8) 应指定位置，使从中心点至始点的距离与从中心点至终点的距离之差为 0.01mm 以内。
- (9) 3 点位于同一直线上的情况下，或者是始点与中心点、或终点与中心点相同的情况下，会发生错误。
- (10) 始点与终点相同的情况下，或 3 点相同的情况下，不发生错误而执行下一个指令。
 但是，此时姿势不相同的情况下，仅进行姿势的插补。
- (11) ALWAYS、ERROR 属性的程序中无法执行。（会发生 L3287 错误。）
- (12) 未设定工件坐标系的情况下，该初始值全部组成成分为 0(0, 0, 0, 0, 0)。该状态下，如执行 Ex-T 控制直线插补，则变为沿着世界坐标系原点动作。
- (13) 附加了附加轴时，附加轴也会动作。但是，Ex-T 控制中不包含附加轴。因此，移动行走轴等机器人本体之类的情况下，设想工件坐标也会与机器人一起移动，并偏离本来的作业位置。因此，不建议通过本指令执行附加轴动作。此外，使用本指令时，请勿使用附加轴同步控制。
- (14) 常数 2 中指定了三轴直交时，动作与通过 Mvr 指令指定了三轴直交时相同。

【相关指令】

EMvc (E Move C)、EMvr (E Move R)、EMvr2 (E Move R2)、EMvs (E Move S)

【相关系统状态变量】

P_WkCord (工件坐标数据)

【相关参数】

WK1CORD ~ WK8CORD

EMvs (E Move S)

【功能】

从当前位置至目的位置，沿着工件坐标系（Ex-T 坐标系）进行直线插补动作。
（关于功能概要，请参照“第 652 页的「7.5 关于 Ex-T 控制」”。）

【格式】

EMvs □ < 工件坐标号码 >,< 移动目的位置 > [□ Type □ < 常数 1>,< 常数 2>]
[□ < 附带条件 >]

【用语】

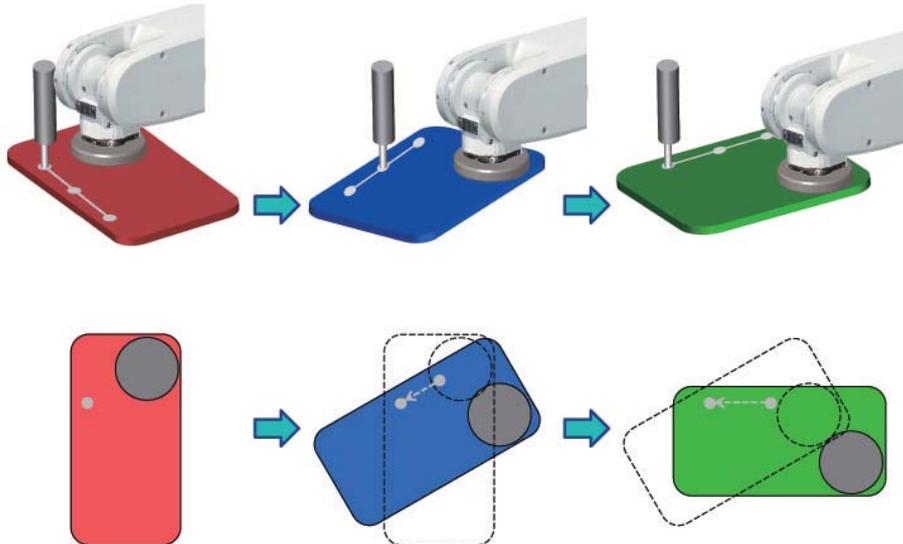
- < 工件坐标号码 > 指定 1~8 的工件坐标号码。
- < 移动目的位置 > 指定插补动作的最终位置。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。（输入机器人控制点的位置数据。）
- < 常数 1 > 姿势旋转的指定。抄近道动作 = 0。省略时为 0。
- < 常数 2 > 姿势插补方法的指定。等效旋转/三轴直交 = 0/1。省略时为 0。
- < 附带条件 > 对 Wth 节或 Wthlf 节进行记述。

【例文】

- ① 向目的位置 P1 沿着工件坐标 1 进行直线插补移动
1 EMvs 1,P1
- ② 向目的位置 (Plt 1,5) 沿着工件坐标 3 进行直线插补移动
1 EMvs 3,(Plt 1,5)
- ③ 向目的位置 P1 沿着工件坐标 2 进行直线插补移动的同时，将输出信号 17 置为 ON
1 EMvs 2,P1 Wth M_Out(17)=1
- ④ 向目的位置 P4 沿着工件坐标 1 进行直线插补移动，在该移动过程中输入信号 18 为 ON 后，立即将输出信号 20 置为 ON
1 EMvs 1,P4 Wthlf M_In(18)=1, M_Out(20)=1

【说明】

- (1) EMvs 指令（Ex-T 控制直线插补）为沿着指定的工件坐标系从机器人当前位置移动至移动目的位置的动作形态。
通过 EMvs 指令进行的动作示例如图所示。沿着 Ex-T 控制点（Ex-T 坐标系原点）改变姿势的同时进行直线插补的示例。



- (2) 姿势为从工件坐标看到的从始点至终点的插补。
- (3) 工件坐标号码 1~8 对应 WK1CORD ~ WK8CORD。
- (4) 仅 MELFA-BASIC V 可以使用。

- (5) 通过 Spd 指令可以指定 Ex-T 控制直线插补的动作速度。此外，也可以通过 OvrD 指令及 O/P 倍率修调设定值进行更改。
- (6) Ex-T 控制直线插补的加减速为 Oadl On 时（初始值）的最佳加减速。Accel 指令有效。
- (7) 通过 MvTune 指令可以切换动作模式。
- (8) 也可以通过 Cnt 指令进行轨迹连接、通过 Fine 指令进行定位精度指定。
- (9) Ex-T 控制直线插补动作过程中可以中断 / 重新执行。因中断后重新开始而发生的移动至中断点的动作，不是 Ex-T 控制直线插补动作，而是常规的插补动作。移动至该中断点的插补形式，因 RETPATH 参数的设定而异。初始设定为“通过关节插补向中断位置的移动”。
- (10) ALWAYS、ERROR 属性的程序中无法执行。（会发生 L3287 错误。）
- (11) 未设定工件坐标系的情况下，该初始值全部组成成分为 0(0, 0, 0, 0, 0, 0)。该状态下，如执行 Ex-T 控制直线插补，则变为沿着世界坐标系原点动作。
- (12) 附加了附加轴时，附加轴也会动作。但是，Ex-T 控制中不包含附加轴。因此，移动行走轴等机器人本体之类的情况下，设想工件坐标也会与机器人一起移动，并偏离本来的作业位置。因此，不建议通过本指令执行附加轴动作。此外，使用本指令时，请勿使用附加轴同步控制。
- (13) 常数 2 中指定了三轴直交时，动作与通过 Mvr 指令指定了三轴直交时相同。

【相关指令】

EMvc (E Move C)、EMvr (E Move R)、EMvr2 (E Move R2)、EMvr3 (E Move R3)

【相关系统状态变量】

P_WkCord（工件坐标数据）

【相关参数】

WK1CORD ~ WK8CORD

EMvSpl (E Move Spline)

【功能】

基于登录在指定的样条文件中的信息，沿着 Ex-T 坐标系原点执行样条插补。（关于样条插补的详细内容，请参照第 587 页的「7.4 关于样条插补」。）

注）该指令可在软件版本为 Ver.R6b 以上（F-Q 系列）/Ver.S6b 以上（F-D 系列）的版本中使用。

【格式】

```
EMvSpl □ <Ex-T 坐标号码>,<样条号码>,<速度>,<加减速距离>
      [,<帧转换> [,<姿势插补类型> [,<滤波器长度>]]]
```

【用语】

- <Ex-T 坐标号码> 以常数或数值变量指定作为控制点的 Ex-T 坐标号码。
 设定范围：
 0：使用登录在样条文件中的 Ex-T 坐标系原点数据。
 1～8：使用与号码对应的工件坐标系原点数据。
- <样条号码> 以常数或数值变量指定保持有希望使其动作的路径信息的样条文件的号码。
 设定范围：1～99
- <速度> 以常数或数值变量指定样条插补的动作速度。
 设定范围：比 0.0 大的数值 (mm/s)
- <加减速距离> 以常数或数值变量指定样条插补的加减速距离。
 加速距离为从动作开始加速至指定速度所需要的距离。
 减速距离为从指定速度开始减速至结束位置停止所需要的距离。
 样条插补中，加速距离与减速距离为通用的设定。
 设定范围：比 0.0 大的数值 (mm/s)
- <帧转换> 以常数或数值变量指定帧转换的执行内容。
 设定范围：
 0：不执行帧转换。
 1：使用样条文件中设定的坐标系执行帧转换。
 2：使用 SetCalFrm 指令中设定的坐标系执行帧转换。
 省略时：不执行帧转换。
- <姿势插补类型> 以常数或数值变量指定姿势的插补类型。
 设定范围：
 0（等效旋转动作）/1（三轴直交动作）
 省略时：等效旋转动作
- <滤波器长度> 以常数或数值变量指定施加在加减速动作上的滤波器长度。
 设定范围：0.0～1000.0 (ms)
 省略时：100.0ms

【例文】

- 1 EMvSpl 0,2,20,10 '通过样条文件 2 中设定的路径点数据生成曲线，让该曲线沿着登录在样条文件中的 Ex-T 坐标系原点动作（这个动作为 Ex-T 样条插补）
- 2 EMvSpl 7,2,30,10 '通过样条文件 2 中设定的路径点数据生成曲线，让该曲线沿着工件坐标 7 动作（这个动作为 Ex-T 样条插补）

【说明】

(1) <Ex-T 坐标号码> 指定为 1～8 时，执行遵从 Wk1CORD～WK8CORD 中设定的坐标系数据的样条插补。指定为 0 时，执行遵从登录在样条文件中的 Ex-T 坐标系原点数据的样条插补。

- (2) 样条文件中未进行 Ex-T 坐标设定时, 会发生 L2610(Ex-T 坐标设定不当) 错误。
- (3) 基于登录在 < 样条号码 > 对应的样条文件中的路径点数据等信息, 执行 Ex-T 样条插补。
- (4) 以从夹持的工件看到的 Ex-T 坐标系原点能够形成顺畅的曲线进行动作。
- (5) EMvSpl 指令开始执行时的机器人的当前位置与 Ex-T 样条插补的开始位置不一致时, 以 Ex-T 直线插补 (EMvs 指令) 动作至开始位置后, 执行 Ex-T 样条插补。
- (6) < 样条号码 > 对应的样条文件未保存在控制器中的情况下, 会发生 L2610(无法打开样条文件)。
- (7) Ex-T 样条插补的指令速度 (从工件看到的 Ex-T 坐标系的速度) 指定为以下算式。
指令速度 = (EMvSpl 指令的 < 速度 >) × (Ovrd 指令) × (操作面板的倍率修调)
不使用 Spd 指令与 JOvrd 指令的设定。此外, 即使 < 速度 > 中设定状态变量 M_NSpd, 最佳速度控制模式也不起作用。
- (8) Ex-T 样条插补的加减速通过 < 加减速距离 > 来指定。不使用 Accel 指令的设定。此外, 即使通过 Oadl 指令将最佳加减速控制设定为有效, 也不适用于 Ex-T 样条插补。
- (9) 加速动作将生成速度, 使从 Ex-T 样条插补的开始位置移动 < 加减速距离 > 即达到 < 速度 >。减速动作将生成速度, 使从 < 速度 > 移动 < 加减速距离 > 以在结束位置停止。
将 Ovrd 指令 · 操作面板的倍率修调设定为比 100% 小的数值时, 以比 < 加减速距离 > 更短的移动量达到指令速度。
(因滤波器等的影晌, 实际上是比通过 < 加减速距离 > 指定的数值更长的加减速距离。)
- (10) < 帧转换 > 指定为 “1(使用样条文件中设定的坐标系执行)” · “2(使用通过 SetCalFrm 指令设定的坐标系执行)” 时, 基于指定的方法依次转换路径点数据, 执行通过该路径点的 Ex-T 样条插补。
- (11) 即使 < 帧转换 > 已指定为 “1(使用样条文件中设定的坐标系执行)”, 当样条文件中未设定坐标系时, 也会发生 L2042(帧转换坐标未设定) 错误。
- (12) < 帧转换 > 指定为 “1(使用样条文件中设定的坐标系执行)” 时, 无法算出坐标系的情况下, 会发生 L2041(帧转换坐标无法算出) 错误。
- (13) < 帧转换 > 指定为 “2(使用通过 SetCalFrm 指令设定的坐标系执行)” 时, 则使用通过最后执行的 SetCalFrm 指令设定的坐标系执行帧转换。从未执行过 SetCalFrm 指令且未设定坐标系时, 会发生 L2042(帧转换坐标未设定) 错误。
- (14) 通过 < 姿势插补类型 > 的指定, 可以切换 “0(等效旋转动作)” 与 “1(三轴直交动作)”。
等效旋转动作, 以路径点间的姿势变化为最小角度进行动作。
三轴直交动作通过 J4、J5、J6 轴角度、而非 A、B、C 轴坐标值来生成姿势数据。因此, 路径点的姿势数据 (A、B、C 轴坐标值) 即使相同, 动作过程中机器人的姿势也会变化。
- (15) 不支持结构标志的跨越 (特异点通过)。路径点间结构标志不相同, 会发生 L2611(路径点的结构标志不相同) 错误。
- (16) 可以更改通过 < 滤波器长度 > 施加在加减速动作上的滤波器特性。希望加速 · 减速时的动作更为顺畅时, 可加长滤波器长度。但是, 加得过长则动作会变缓慢, 到 Ex-T 样条插补结束为止的时间会变长。
- (17) 路径点数据中未设定信号输出的情况下, 通过对象的路径点时, 将以指定的条件输出通用输出信号。
- (18) 指定了机器人的定位完成条件 (Fine 指令、Fine J 指令、Fine P 指令) 的情况下, 在 Ex-T 样条插补的结束位置上确认定位完成。
- (19) 即使通过 Cnt 指令指定了连续动作, 在 Ex-T 样条插补的开始位置 · 结束位置上也不会进行连续动作。
- (20) Def Act 指令 (中断的条件与处理的定义) 的 < 类型 > 即使指定为 “停止类型 1”, Ex-T 样条插补执行过程中发生了中断时, 也将以与 “停止类型 2” 相同的减速停止。
- (21) 附加轴不从 Ex-T 样条插补的开始位置进行动作。中途的路径点上即使设定了附加轴的位置, 也不用于 Ex-T 样条插补。
- (22) 对 EMvSpl 指令, 可以进行单步进给, 但无法进行单步退回。会发生 L2612 (无法执行单步退回) 错误。
- (23) 执行 EMvSpl 指令需要机械的控制权 (GetM 指令)。

- (24) 启动条件无法通过 ALWAYS·ERROR 的插槽执行 EMvSpl 指令。会发生 L2612 (该启动条件下无法执行) 错误。
- (25) EMvSpl 指令的自变量中设定了超出设定范围的值时, 会发生 L3110(自变量的值超出范围 (EMvSpl)) 错误。



如指定了相对 <速度> 而言过短的 <加减速距离>, 则会变成急剧的加减速, 将导致机器人振动或发生伺服错误, 因此应设定适当的加减速距离。

【相关指令】

SetCalFrm (Set Calibration Frame)

【相关系统状态变量】

M_SplPno、M_SplVar、P_WkCord、P_ECord

【相关参数】

WK1CORD ~ WK8CORD、SPLOPTGC

End

【功能】

定义程序的最终行。

在以子程序记述在程序的后半段的时候，在主处理的最后以 **End** 指令记述，且明白的表示程序的结束时也可以使用。在以 **Callp** 呼出子程序的情况下，以 **End** 指令的执行返回到主程序。

【格式】

End

【例子】

```

1 Mov P1
2 GoSub *ABC
3 End           ' 程序的结束
:
10 *ABC
11 M1=1
12 Return

```

【说明】

- (1) 定义程序的最终行。想要将程序在中途停止并呈中断状态的情况下，请使用 **Hlt** 指令。
- (2) 从操作面板执行的时候，因为是以连续运行的模式执行，所以即使有 **End** 指令也会再度从头开始执行。
想要以 **End** 指令结束的情况下，请按下操作面板的 **End** 键，使循环停止。
- (3) 即使在一个程序里记述复数的 **End** 指令也没有关系。
- (4) 在程序的终端即使没有记述 **End** 指令也没有关系。
- (5) 在以 **CallP** 呼出的程序里执行 **End** 指令的话，控制会返回到主程序。会变成和子程序（**GsSub** 指令）的 **Return** 指令同样的动作。
- (6) 以 **End** 指令的执行，被开启的文件及通信端口全部会被关闭。
- (7) 以 **End** 指令的执行，**Spd**、**Accel**、**Oadl**、**Jovrd**、**Ovrd**、**fine**、**Cut** 的设定会被初始化。

【相关指令】

[Hlt \(Halt\)](#)、[Call P](#)

Error

【功能】

用程序里使报警发生（9000号）。

【格式】

通常的程序

```
Error □ <异常号码>
```

始终执行（ALWAYS）程序

```
Error □ < 错误号码> , <条件>
```

【用语】

- <报警号码> 可以用常数或数值表达式设定。在号码9000~9299为止的范围指定。
- <条件> 0: 报警停止执行Error指令的程序。
1: 不报警停止执行Error指令的程序。

【例子】

① 使报警 9000 发生。

```
1 Error 9000
```

② 依据 M1 的值，改变发生的报警

```
4 If M1 <> 0 Then *LERR      ' M1 不为 0 时，往 *LERR 分支
:
14 *LERR
15 MERR=9000+M1*10          ' 依据 M1 的值，计算报警号码
16 Error MERR                ' 使计算出来的报警号码发生。
17 End
```

【说明】

- (1) 依据此指令的执行，可以使 9000 号的报警任意的发生。
- (2) 使低标准报警、高标准报警发生的情况下，其程序会中断。Error 指令以后的行不会被执行。警告则不会中断，且下一个行以后也会执行。
- (3) 依据号码，系统会有下列动作。
- (4) 依据参数 UER1 ~ UER20，最多可做成 20 个报警信息。
- (5) 报警号码、报警标准及系统的动作如下表所示。
指定下列报警号码范围以外的值的情况下，系统会发生报警。

号码	系统的动作
9000 ~ 9099 (高标准报警)	停止程序的执行并切断伺服电源。 以报警复位的输入来解除报警状态。
9100 ~ 9199 (低标准报警)	停止程序的执行。 以报警复位的输入来解除报警状态。
9200 ~ 9299 (警告)	继续程序的执行。 以报警复位的输入来解除报警状态。

【关连参数】

UER1 ~ 20

Fine

【功能】

以编码器脉冲数指定机器人的定位完成条件。
在连续动作控制（Cnt 1）中，Fine 指令为无效。

【格式】

Fine □ <脉冲数> [, <轴号码>]

【用语】

<脉冲数>	以变量或常数指定定位编码器脉冲。 指定0的话会变成无效。初始值会被设定为0。
<轴号码>	指定定位脉冲所指定的轴号码。省略的话，会变成全轴。可以常数或数值变量指定。

【例子】

1 Fine 300	' 将定位脉冲指定为 300
2 Mov P1	
3 Fine 100, 2	' 将 2 轴的定位脉冲指定为 100
4 Mov P2	
5 Fine 0, 5	' 将定位脉冲指定为无效
6 Mov P3	
7 Fine 0	' 将定位脉冲指定为 100
8 Mov P4	

【说明】

- (1) Fine 指令是将动作指令完成条件（定位精度）以反馈的脉冲数指定的指令。因为是脉冲数来判断动作完成，因此可以得到更正确的定位。
- (2) Fine 指令有效的所有轴的指令脉冲和反馈脉冲的差变为 <脉冲数> 以内时，视为动作完成。
- (3) 此外，也可以用 Dly 指令取代 Fine 指令做定位。这方面可以简单的指定。
 - 1 Mov P1
 - 2 Dly 0.1
- (4) 在初始状态，全轴 Fine 无效。暂时使 Fine 变有效的话，到无效为止，会持续有效状态。
- (5) 在程序结束（执行 Fine 指令、中断后的程序复位）时，Fine 会变成无效。
- (6) 变成连续动作控制有效状态（Cnt 1）的话，Fine 有效会暂时被忽视（无效，状态保持）。
- (7) 执行 Fine 指令时，Fine J 指令或 Fine P 指令为全轴无效。（不保持状态。）
- (8) 对应附加轴（通用伺服轴），可以切换 Fine 的有效 / 无效，但是，无法指定脉冲数。伺服放大器的参数「INP」的设定值会变成定位的脉冲数。换言之，指定 0 以外的整数的情况下，在伺服放大器的参数设定值，Fine 会变成有效，指定 0 的情况下，Fine 会变无效。
- (9) 在有伺服弹性模式功能时，依据 Fine 指令指定定位完成条件的话，依据动作，会使机器人无法到达目的位置的定位完成脉冲、持续等待动作指令的完成，而无法执行程序的前进。请不要同时使用伺服弹性模式及 Fine 指令。
- (10) 由于 Fine 指令是使用电机轴的编码器脉冲进行定位判定的，因此会受到编码器分辨率和齿轮比的影响。每个脉冲的输出角度因机型和轴而有所不同，通过输出轴单位（mm 或度）监视定位时，请使用 Fine P 指令、Fine J 指令。

Fine J

【功能】

机器人的定位完成条件以关节轴的值指定。
在连续动作控制（Cnt 1）中，Fine J 指令为无效。
执行 Fine J 指令时，Fine 指令或 Fine P 指令为全轴无效。

【格式】

Fine □ < 定位宽度 > , J[, < 轴号码 >]

【用语】

< 定位宽度 > 定位宽度以变量或常数指定。
指定0的话会变成无效。初始值会被设定为0。
单位取决于关节轴的单位系统，为[mm]和[deg]中的一个。
可指定的最小值为0.001。

< 轴号码 > 对指定定位脉冲的轴号码进行指定。省略的话，会变成全轴。
可以以常数或数值变量指定。

【例文】

1 Fine 1, J	' 将全轴的定位宽度指定为 1[mm] （或 [deg]）
2 Mov P1	
3 Fine 0.5, J, 2	' 将 2 轴的定位宽度变更为 0.5[mm] （或 [deg]）
4 Mov P2	
5 Fine 0, J, 5	' 将 5 轴的定位宽度指定设为无效
6 Mov P3	
7 Fine 0, J	' 将全轴的定位宽度指定设为无效
8 Mov P4	

【说明】

- (1) Fine J 指令是将动作指令完成条件（定位精度）以反馈的关节值指定的指令。因为是通过关节值来判断动作完成的，因此可以得到更正确的定位。
- (2) Fine J 指令有效的所有轴的指令关节位置和反馈关节位置的差变为 < 定位宽度 > 以内时，视为动作完成。
- (3) 此外，也可以用 Dly 指令取代 Fine J 指令进行定位。该指令可更加简单地进行指定。
1 Mov P1
2 Dly 0.1
- (4) 在初始状态，全轴 Fine J 无效。一旦将 Fine J 设为有效，有效状态会持续直到无效为止。
- (5) 在程序结束时（End 指令执行、中断后的程序复位），Fine J 会变为无效。
- (6) 变为连续动作控制有效状态（Cnt 1）的话，Fine J 有效会暂时被忽略（无效，保持状态）。
- (7) 执行 Fine J 指令时，Fine 指令或 Fine P 指令为全轴无效。（不保持状态。）
- (8) 对附加轴（通用伺服轴），也可以切换 Fine J 的有效 / 无效、指定 < 定位宽度 >。
- (9) 在伺服弹性模式生效时，通过 Fine J 指令指定定位完成条件的话，根据动作，可能会使机器人无法达到目的位置的定位完成脉冲，持续等待动作指令的完成，而导致程序无法执行下去。请勿同时使用伺服弹性模式及 Fine J 指令。

Fine P

【功能】

机器人的定位完成条件以直线距离指定。
在连续动作控制（Cnt 1）中，Fine P 指令为无效。
执行 Fine P 指令时，Fine 指令或 Fine J 指令为全轴无效。

【格式】

```
Fine □ < 直线距离 > , P
```

【用语】

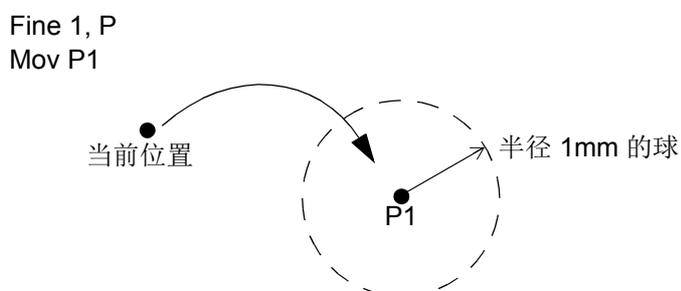
< 直线距离 > 定位直线距离[mm]以变量或常数指定。
指定0的话会变成无效。初始值会被设定为0。
可指定的最小值为0.001。

【例文】

```
1 Fine 1, P                    ' 将定位直线距离指定为 1mm
2 Mov P1
3 Fine 0, P                    ' 将定位直线距离指定设为无效
4 Mov P2
```

【说明】

- (1) Fine P 指令是将动作指令完成条件（定位精度）以反馈的直线距离指定的指令。因为是通过直线距离来判断动作完成的，因此可以得到更正确的定位。
- (2) 可从指令脉冲和反馈脉冲求得的，各机器人的现在位置之间的直线距离变为 < 直线距离 > 以内时，视为动作完成。
- (3) 此外，也可以用 Dly 指令取代 Fine P 指令进行定位。该指令可更加简单地进行指定。
1 Mov P1
2 Dly 0.1
- (4) 在初始状态，全轴 Fine P 无效。一旦将 Fine P 设为有效，有效状态会持续直到无效为止。
- (5) 在程序结束时（End 指令执行、中断后的程序复位），Fine P 会变为无效。
- (6) 变为连续动作控制有效状态（Cnt 1）的话，Fine P 有效会暂时被忽略（无效，保持状态）。
- (7) 执行 Fine P 指令时，Fine 指令或 Fine J 指令为全轴无效。（不保持状态。）
- (8) 对附加轴（通用伺服轴），无法切换 Fine P 的有效/无效。Fine P 始终无效。
- (9) 在伺服弹性模式生效时，通过 Fine P 指令指定定位完成条件的话，根据动作，可能会使机器人无法达到目的位置的定位完成脉冲，持续等待动作指令的完成，而导致程序无法执行下去。请勿同时使用伺服弹性模式及 Fine P 指令。



进入以 P1 为中心的半径为 1mm 的球内则定位完成。

图 4-19: Fine P 命令使用例

For ~ Next

【功能】

将 For 文和 Next 文间的程序，到满足结束条件前，循环的执行。

【书式】

```
For □ <计数器> = <初始值> □ To □ <结束值> □ [Step <增量>]
:
Next □ [<计数器>]
```

【用语】

<计数器>	作为循环控制的计数器，记述数值变量。
<初始值>	将循环控制的计数器的初始值以数值表达式设定。
<结束值>	将循环控制的计数器的结束值以数值表达式设定。
<增量>	将循环控制的计数器的增量以数值表达式设定。 将单步以后省略的话，增量会变成1。

【例子】

① 求出从 1 到 10 的求和程序

```
1 MSUM=0           ' 将合计 M1 初始化。
2 For M1=1 To 10   ' 使数值变量 M1，从 1 开始以计数每次增加 +1 到 10 为止。
3 MSUM=MSUM+M1    ' 在数值变量 MSUM 加上 M1 的值。
4 Next M1         ' 返回到步号 2。
```

② 将 2 个数值的相乘结果设定在 2 次元排列变量的程序

```
1 Dim MBOX(10,10) ' 确保 10×10 的排列领域。
2 For M1=1 To 10 单步 1 ' 使数值变量 M1，从 1 开始以计数每次增加 +1 到 10 为止。
3   For M2=1 To 10 单步 1 ' 使数值变量 M2，从 1 开始以计数每次增加 +1 到 10 为止。
4     MBOX(M1,M2)=M1*M2 ' 在排列变量 MBOX (M1, M2) 里代入 M1*M2 的值。
5   Next M2           ' 返回到步号 3。
6 Next M1            ' 返回到步号 2。
```

③ 从 For ~ Next 中通过 Break 可穿插至 Next 的下一行。

```
1 MSUM=0           ' 将合计 MSUM 进行初始化。
2 For M1=1 To 10   ' 数值变量 M1 从 1 开始到 10 为止每次 +1 进行计数。
3 MSUM=MSUM+M1    ' 在数值变量 MSUM 上加上 M1 的值。
4 If M_In(8)=1 Then Break ' 输入信号 8 开启则跳转到步号 6。
5 Next           ' 返回到步号 2。
6 If M_BrkCq=1 Then Hlt
```

【说明】

- (1) 在 For 文和 Next 文的中间可以记述其它的 For ~ Next。但是，一组的 For ~ Next 的控制会将程序内的控制构造更加深一段。程序内的控制构造深度，最高可以到 16 段的深度。超过 16 段的情况下，执行时会发生报警。
- (2) 在 For 文和 Next 文之间以 GoTo 指令强制的执行跳转的话，因为控制构造用内存（堆栈内存）的减少，在连续执行的情况下，迟早会发生报警。请使 For 文的条件成立，执行脱离循环（Loop）的程序。
- (3) 下列条件的情况下，执行时会发生报警。
 - 计数器的<初始值>比<结束值>大，<增量>为正的值的的情况
 - 计数器的<初始值>比<结束值>小，<增量>为负的值的的情况
- (4) For 文和 Next 文没有相对应的情况下，执行时会发生报警。
- (5) Next 文的计数器变量可以省略。在例子②里，可以省略步号 5 的 M2 和步号 6 的 M1。将计数器变量省略的话，某些处理会变快。
- (6) For ~ Next 中如果有 Break 语句，会跳转到 Next 的下一行，并从 For ~ Next 的处理中退出。

FPrm

【功能】

将程序作为有自变量的子程序（从主程序里以 CallP 指令呼出的程序）使用的情况下，定义自变量。

【格式】

```
FPrm □ <假设自变量> [ , <假设自变量> ] ...
```

【用语】

<假设自变量> 接受从主程序来的自变量值的变量，任意型的变量皆可使用。
最大可记述16个变量。

【例子】

<主程序>

1 M1=1

2 P2=P_Curr

3 P3=P100

4 CallP "100",M1,P2,P3 '也可以记述为「CallP "100", 1, P_Curr, P100」。

<子程序 "100" >

1 FPrm M1,P1,P2

2 If M1=1 Then GoTo *LBL

3 Mov P1

4 *LBL

5 Mvs P2

6 End

' 返回到主程序。

【说明】

- (1) 在呼叫子程序不使用自变量的情况下，不要 FPrm 文。
- (2) CallP 的自变量和以 FPrm 来定义的假设数的型和个数有差异的情况下，会发生报警。
- (3) 将在子程序的处理结果无法代入自变量传递给主程序。
将子程序的处理结果使用在主程序时，请使用全局变量来传递值。

【相关指令】

Call P

GetM (Get Mechanism)

【功能】

在使用多重任务时，以插槽 1 以外的程序执行机器人控制的情况、或者将附加轴作为使用者定义机器，执行多重机器控制的情况下，使用本指令。

指定控制机器人的机器号码，会获得控制权。解放控制权的情况下，使用 ReIM 指令。

【格式】

```
GetM □ <机器号码>
```

【用语】

< 机器号码 > 1~3，以常数或变量指定。
标准系统的机器人会变成机器1。

【例子】

① 任务插槽 1 将起动任务插槽 2 起动，且在任务插槽 2 中控制机器 1 任务插槽 1

```
1 ReIM          ' 为了在插槽 2 控制机器 1，解放机器。
2 XRun 2,"10"   ' 在插槽 2 选择程序 10
3 Wait M_Run(2)=1 ' 插槽 2 的起动确认等待
:
```

任务插槽 2 (程序 "10")

```
1 GetM 1      ' 获得机器 1 的控制权。
2 Servo On    ' 机器 1 的伺服开启
3 Mov P1
4 Mvs P2
5 P3=P_Curr   ' 机器 1 的现在位置代入 P3
6 Servo Off   ' 机器 1 的伺服关闭
7 ReIM        ' 解放机器 1 的控制权
8 End
```

【说明】

- (1) 因为通常 (单一任务) 是以初始状态获得机器 1 的控制权，因此没有必要使用 GetM。
- (2) 因为不能够以复数的任务同时获得 1 个的机器控制权，因此要在插槽 1 以外使机器人动作的情况下，务必执行下列步骤。
首先，以插槽 1 的程序，依据 ReIM 指令解放控制权。
再来，在使机器人动作的插槽程序里，依据 GetM 指令获得控制权。
在已经获得控制权的插槽里，再度执行 GetM 指令的话，会发生报警。
- (3) 控制权为必要的指令有电机电源 ON/OFF 指令、插补指令、速度加速度指定指令、TOOL/BASE 指令等。需要控制权的指令的详细内容，请参照以下内容。

机器人指令	Accel、Base、CavChk、Cmp Jnt、Cmp Pos、Cmp Tool、Cmp Off、CmpG、Cnt、ColChk、ColLvl、Def Plt、Plt、Fine、Fsc、FsGChg、HOpen、HClose、JRC、LoadSet、Mov、Mva、Mvc、Mvr、Mvr2、Mvr3、Mvs、MvSpl、MvTune、Mxt、Oadl、P_Base、Prec、Servo、SetCalFrm、Spd、SpdOpt、Tool、Torq、Trk、EMvc、EMvr、EMvr2、EMvr3、EMvs
函数	Align、PosCq、PosMid ※ 以下为即使从中间版本不进行 GetM 而使用时也不会出现错误，且通过机械 1 可使用的指令 JtoP、PtoJ Ver.N8/P8 ~ Inv..... Ver.R3a/S3a(特殊 ES Ver.R3dZ) Fram..... Ver.R3a/S3a(特殊 ES Ver.R3dZ)
机器人状态变量	J_Curr、J/Fbc、J_Origin、P_Curr、P_CurrR、P_Fbc、P_Safe、P_Tool、M_Rspd、M_SetAdl

- (4) 将必须指定机器的系统状态变量的自变量省略的情况下，现在取得的机器会被指定。

- (5) 程序中断的情况下，依据系统会自动的执行 ReIM，再起动机时，GetM 会被自动的执行。
 (6) 无法使用在通常执行程序内。

【相关指令】

ReIM (Release Mechanism)

GoSub(~ Return) (Go Subrouine)

【功能】

呼出指定 level 的子程序。

从子程序务必以 Return 指令返回。

【格式】

GoSub □ <呼叫处>

【用语】

<呼叫处> 记述level名。

【例子】

```

10 GoSub *LBL
11 End
:
100 *LBL
101 Mov P1
102 Return            ' 务必以 Return 指令返回。

```

【说明】

- (1) 从子程序，请务必以 Return 指令返回。以 GoTo 指令返回的话，机器构造用内存（堆栈内存）会减少，且会变成连续运行时发生报警的原因。
- (2) 从子程序中，可以依据 Gosub 再度呼出其它的子程序。子程序可以呼出的段数大约为 800 段。
- (3) 呼出处以 level 指定。呼出处的 level 不存在的情况下，执行时会发生报警。

【相关指令】

Return

GoTo

【功能】

无条件的分支到程序内指定的 level。

【格式】

```
GoTo □ <分支处 >
```

【用语】

<分支处> 记述level名。

【例子】

```
10 GoTo *LBL '分支到 level *LBL。
:
100 *LBL
101 Mov P1
```

【说明】

- (1) 在分支处指定 level。
- (2) 分支处的 level 不存在的情况下，执行时会发生报警。

Hlt (Halt)

【功能】

中断并停止程序的执行和机器人的动作。此时，已执行的程序会变成待机中。

【格式】

```
Hlt
```

【例子】

① 无条件在程序的途中，使机器人停止。

```
150 Hlt ' 无条件的中断程序。
```

② 满足某个条件时，使机器人停止。

```
100 If M_In(18)=1 Then Hlt ' 输入信号 18 开启的情况下，程序中断。
```

```
200 Mov P1 Wthlf M_In(17)=1, Hlt ' 往 P1 移动中，输入信号 17 开启的情况下，程序中断。
```

【说明】

- (1) 将程序的执行中断且将机器人减速后停止。此时系统会变成待机状态。
- (2) 在多任务使用的时候，只有执行 Hlt 的任务插槽会中断。
- (3) 再开启，会在操作面板的起动或外部开始的起动信号执行。程序会从 Hlt 文的下一行开始执行。但是，附随句（Wthlf 指令）的 Hlt 的情况下，会从执行中断的行再开始执行。

【相关指令】

[End](#)

注意

使用跟踪功能时

在跟踪动作中执行本 Hlt 指令，会停止跟踪动作（相当于 Trk Off 指令），中断程序的执行。使用多机器人时，对通过 GetM 指令取得机器人号码的机器人停止跟踪动作。使用重启（继续）继续跟踪动作时，请制作程序，以执行 Trk On 指令。

HOpen/HClose (抓手 Open/Close)

【功能】

指示抓手开、闭。

【格式】

HOpen □ < 抓手号码 > [, < 起动把持力 >, < 保持把持力 >, < 起动把持力持续时间 >]
 HClose □ < 抓手号码 >

【用语】

- | | |
|------------------------------|--|
| < 抓手号码 > | 选择1~8的数值的其中之一。以常数或变量指定。 |
| < 起动把持力 > ^{注)} | 在电动抓手的情况下为有效的参数。电动抓手以外的情况为无效。
为了起动抓手开闭, 将必要的把持力以常数或变量指定。
以从0到63阶段 (63=3.5kg) 设定把持力。初始值为63。省略时, 上一次的设定值为有效。 |
| < 保持把持力 > ^{注)} | 在电动抓手的情况下为有效的参数。电动抓手以外的情况为无效。
为了起动抓手开闭, 将必要的把持力以常数或变量指定。
以从0到63阶段 (63=3.5kg) 设定把持力。
初始值为63。省略时, 上一次的设定值为有效。 |
| < 起动把持力继续定时器 > ^{注)} | 只在电动抓手的情况下有效的参数。
将起动把持力持续的时间以常数或变量设定
可设定从0.00 (秒) 开始到精度实数的最大值为止。
初始值为0.3秒。 |

注) 仅在三菱公司制造的电动抓手中有效。

关于 TAIYO 公司制造的多功能电动抓手的控制, 请参照第 547 页的「7.2 多功能电动抓手的使用方法」。

【例子】

```
1 HOpen 1      ' 打开抓手 1。
2 Dly 0.2      ' 定时器为 0.2 秒 (打开抓手后等待)
3 HClose 1     ' 关闭抓手 1。
4 Dly 0.2      ' 定时器为 0.2 秒 (关闭抓手后等待)
5 Mov PUP      '
```

【说明】

- (1) 各抓手的动作设定 (单螺线管 / 双螺线管) 以参数抓手设定类型。
- (2) 在双螺线管设定的情况下, 抓手对应 1~4、双螺线管设定的情况下, 抓手对应 1~8。
- (3) 电源起动时的抓手出信号的状态以参数抓手 INIT 设定。
- (4) 抓手输入信号的确认以机器人状态变量 M_HndCq (' 抓手输入号码 ') 执行。此外, 900~907 号的输入信号 (机器为 1 台的情况) 也可以确认。

```
1 HClose 1
2 *LBL: If M_HndCq(1)<>1 Then GoTo *LBL
3 Mov P1
```
- (5) 在关连项目中有参数。请参照本书第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」及第 445 页的「5.13 关于抓手初始状态」。

【关连系统状态变量】

M_In (900 号)、M_Out (900 号)、M_HndCq

【相关指令】

LoadSet (Load Set)、Oadl (Optimal Acceleration)

【关连参数】

HANDTYPE、HANDINIT

请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」及第 445 页的「5.13 关于抓手初始状态」。

If...Then...Else...EndIf

【功能】

依据式的结果，选择处理并执行。

【格式】

```
If □ < 式 > □ Then □ < 处理 > □ [Else < 处理 >]
```

```
If □ < 式 > □ Then
  < 处理 >
  < 处理 >
  Break
  :
[Else]
  < 处理 >
  < 处理 >
  Break
  : ]
End If
```

【用语】

<式> 把成为比较对象的式，以比较表达式或逻辑表达式记述。

<处理> 记述在 Then继续处理比较的结果为真时的处理，及在Else继续处理为假的时候的处理。

【例子】

① 将 If...Then...Else... 以 1 行记述的时候

```
10 If M1>10 Then *L100
```

```
11 If M1>10 Then GoTo *L20 Else GoTo *L30
```

'M1 比 10 大的情况下，跳转到标准 L100

'M1 比 10 大的情况下，跳转，到标准 L20、
10 以下的情况下，跳转到标准 L30
Then、Else 后面的 GoTo 可以省略。

② 使用 If...Then...Else...EndIf 的联结构造时

```
10 If M1>10 Then
```

```
11 M1=10
```

```
12 Mov P1
```

```
13 Else
```

```
14 M1=-10
```

```
15 Mov P2
```

```
16 EndIf
```

③ 在 Then 或 Else 里，并且记述 If...Then...Else...EndIf 的联结情况下

```
30 If M1>10 Then
```

```
31 If M2 > 20 Then
```

```
32 M1 = 10
```

```
33 M2 = 10
```

```
34 Else
```

```
35 M1 = 0
```

```
36 M2 = 0
```

```
37 EndIf
```

```
38 Else
```

```
39 M1 = -10
```

```
400 M2 = -10
```

```
410 EndIf
```

④ 在 Then 或 Else 中，使用 Break 跳到 EndIf 的下一行。

```

30 If M1>10 Then
31  If M2 > 20 Then Break ' 条件成立的话，跳转到步号 39。
32  M1 = 10
33  M2 = 10
34 Else
35  M1 = -10
36  If M2 > 20 Then Break ' 条件成立的话，跳转到步号 39。
37  M2 = -10
38 EndIf
39 If M_BrkCq=1 Then Hlt
40 Mov P1

```

【说明】

- (1) If .. Then .. Else .. 以一行记述。
- (2) If .. Then .. Else .. EndIf 可以在复数行跨行记述。
- (3) Else 可以省略。
- (4) If .. Then .. Else .. EndIf 的情况下请务必使用 EndIf 记述。
- (5) 以 If .. Then .. Else .. EndIf 的条件分支处理中，用 GoTo 指令让其跳转的话，控制构造用内存（堆栈内存）会不足而发生报警。
- (6) If .. Then .. Else .. EndIf 的情况下，在 Then 或 Else 里，可以记述 If .. Then .. Else .. EndIf 。
（回路可到 8 段）
- (7) 在 Then 或 Else 的后面接连 GoTo 的情况下，可以将 GoTo 省略。
例) If M1 > 10 Then *L200 Else *L300
同时，只有在 Then 的后面接连 GoTo 的情况下可以省略或的其中一个。Else 无法省略。
例) If M1 > 10 Then GoTo *L200 （左边程序可以替换成下列样式）
=> If M1 > 10 Then *L200
=> If M1 > 10 GoTo *L200
- (8) 在 If Then EndIf 之中有 Break 文的话，往 EndIf 的下一行跳转，从 If Then EndIf 的处理中脱离。
- (9) 在条件式中可记述逻辑数。逻辑数如果不为 0 则是真（条件成立），为 0 表示假。因此可以进行以下记述。
例) If M_IN(900) Then M_Out(30)=1
输入信号 900 为 ON 时作为真 (M_In(900)=1) 处理，因此执行 Then 以后的记述。
输入信号 900 为 OFF 时作为假 (M_In(900)=0) 处理，因此不执行 Then 以后的记述。

Input

【功能】

从文件（CRnD 含通路回路）输入数据。接收的数据全部为 ASCII 文字。

【格式】

```
Input □ #< 文件号码 >, < 输入数据名 >[, < 输入数据名 >]...
```

【用语】

- <文件号码> 记述1~8为止的号码。
对应以Open指令分配的文件号码。
- <输入数据名> 记述将输入数据读取的变量名。可记述全部的变量。

【例子】

```
1 Open "temp.txt" For Input As #1      ' 在文件号码 1 分配为 "temp.txt"。
2 Input #1, CABC$                    ' 将从文件号码 1 到断行字符为止的字符串读入 CABC$
:
10 Close #1                          ' 关闭文件号码 1。
```

【说明】

- (1) 从以 Open 文打开的文件号码的文件开始输入数据后，代入变量。
Open 文没有被执行的情况下，会发生报警。
- (2) 输入的数据和代入的变量的类型必须相同。
- (3) 记述复数的变量名的时候，变量名之间必须以逗号（,）做划分。
- (4) 执行 Input 文的时候，会变成等待输入的状态。已输入的数据，和断行（CR, LF）字符输入的同时，代入变量。
- (5) 在被输入比 Input 文的自变量的数量更多的要素数的情况下，读取会被遗漏。在执行 End 或 Close 时，积存在缓冲区（buffer）里的数据会失效。
例）字符串和数值和位置混和输入的情况
1 Input #1,C1\$,M1,P1

从计算机侧传送数据的例子（只限在 CRnD）

（在机器人的标准 PORT 接收的情况：参数 "CRPC232" 为 0 的情况）

```
PRN MELFA,125.75,(130.5,-117.2,55.1,16.2,0,0)(1,0) CR
```

在 C1\$ 被代入 MELFA、M1 被代入 125.75、P1 被代入（130.5,-117.2,55.1,16.2,0,0）（1,0）。

【相关指令】

OpenOpen、Close、Print

JOvrd (J 速度比例)

【功能】

指定只有在机器人的关节移动时有效的速度比例。

【格式】

```
JOvrd □ < 指定速度比例 >
```

【用语】

< 指定速度比例 > 速度比例以实数指定。也可以用数值表达式记述。
单位: [%] (建议范围: 0.1~100.0)

【例子】

```
1 JOvrd 50  
2 Mov P1  
3 JOvrd M_NJOvrd      ' 设定初始值
```

【说明】

- (1) JOvrd 指令只在关节插补时有效。
- (2) 实际的速度比例 = (操作面板 (示教单元) 的速度比例设定值) × (速度比例程序 (Ovrd 指令) 关节速度比例 (JOvrd 指令))。JOvrd 指令只会使关节插补动作的速度比例变化。
- (3) < 指定速度比例 >, 100% 会变成成为机器人的能力的最大。通常系统初始值 (M_NOvrd) 会被设定为 100 %。以 End 文的执行或程序复位, 会返回到初始值。

【相关指令】

Ovrd、Spd (Speed)

【关连系统状态变量】

M_NJOvrd (系统的初始值)、M_JOvrd (现在的指定关节速度比例)

JRC (Joint Roll Change)

【功能】

- 在机器人本体的对象轴（参照【用语】的<轴号码>）的现在关节坐标值，加算 ± 360 度后替换成现在的坐标值。
- 使用者定义轴（附加轴、使用者定义机器）将指定轴的现在关节坐标值，以参数将指定值加算后替换为现在坐标值。
旋转轴，可使用直动轴的任何一个。也可在现在坐标值再度原点设定。

【格式】

JRC □ <[+] <数值>/-<数值>/0> [, <轴号码>]

【用语】

<数值>

指定增加减量的数（360度的倍数）。可用常数或变量记述。

例) +3: 将对象轴的角度增加+1080度。-2: 减少-720度。

指定+1/-1/0的情况下，有下列的意义。

+1: 将指定轴的现在关节角度以参数JRCQTT增加指定的量。

（符号可省略）机器人本体的优先轴的情况下，360度为固定。

-1: 将指定轴的现在关节角度以参数JRCQTT减少指定的量。

机器人本体的优先轴的情况下，360度为固定。

0: 将指定轴以参数JRCORG指定的值再做原点设定。

只能使用在使用者定义轴。

<轴号码>

将作为对象的轴以常数或变量指定。省略时会变成优先轴。

而且，将附加轴、使用者定义机器的轴做为对象时，无法省略。

[对象机型和对象轴]

(1) 机器人本体

RH-F 系列: J4 轴 (优先轴)

RV-F 系列: J6 轴 (优先轴)

(2) 全机型的附加轴

(3) 使用者定义机器的全轴

【例子】

- | | |
|----------|-------------------------------|
| 1 Mov P1 | ' 往 P1 移动。(J6 轴往正号方向移动) |
| 2 JRC +1 | ' 将对象轴的现在坐标值加算 360°。 |
| 3 Mov P1 | ' 往 P1 移动。 |
| 4 JRC +1 | ' 将对象轴的现在坐标值加算 360°。 |
| 5 Mov P1 | ' 往 P1 移动。 |
| 6 JRC -2 | ' 将对象轴的现在坐标值减算 -720°。(返回到原始值) |

【说明】

(1) 依据 JRC 1/-1 指令 (JRC n/-n)，可以指定增减现在已指定轴的现在关节坐标值。

依据 JRC 0 指令，将已指定轴再次做原点设定。

关节坐标的值改变，但是机器人不会动作。

(2) 使用本指令的情况下，为了依照指令执行不会变成动作范围外，请预先变更对象轴的动作范围。

可以变更关节动作范围参数 "MEJAR" 的相当轴的 - 侧、+ 侧的值。而且，旋转轴的动作范围请设定在 -2340 ~ +2340 度的范围。

(3) 指定轴省略的时候，优先轴会变成对象轴。优先轴会变成机器人尖端的旋转轴。

(4) 在优先轴不存在。(JRC 不可能的机器人) 的情况下，省略指定轴，且已指定的轴不是 JRC 对象轴的时候，执行时会发生报警。

(5) 在未设定原点时，执行时会发生报警。

(6) JRC 指令执行时，机器人停止。即使在 Cnt 有效的情况下也一样，执行此指令的话，插补的连续不会变成连续动作。

- (7) 使用 JRC 指令时，请务必设定下列的参数。
 作为 JRCEXE = 1。（JRC 执行可能）
 在 MEJAR 可以变更对象轴的动作范围。
 在 JRCQTT 设定执行 JRC 1/-1(JRC n/-n) 时的位置变化量。
 （只有附加轴、使用者定义机器的情况）
 在 JRCORG 设定 JRC 0 执行时的原点位置。（只有附加轴、使用者定义机器的情况）
- (8) 参数 JRCEXE = 0 的时候，即使执行 JRC 指令也会变成无处理。
- (9) 以参数 JRCQTT 指定的移动量在不满足脉冲数据 0 ~ 2147483647(0x7fffffff) 的情况下，初始值时会发生报警。
 此外，移动量→脉冲数据的换算，
 回旋转轴的情况：

$$\text{脉冲数据} = \text{移动量 (deg)} / 360 * \text{齿轮比分母} / \text{齿轮比分子} * \text{编码器脉冲数}$$

 直动轴的情况：

$$\text{脉冲数据} = \text{移动量 (mm)} * \text{齿轮比分母} / \text{齿轮比分子} * \text{编码器脉冲数}$$
- (10) 依据 JRC 的执行，原点数据会改变，因此会变成无法使用出货时的原点数据。在必须执行版本升级作业等，将控制器初始化的情况下，请预先在之前的状态，将参数备份。
- (11) JRC 指令无法使用单步返回操作。
- (12) 无法在通常执行程序内使用。
- (13) 指用本指令，伴随比较大的移动量的连续动作执行时，会发生「过负载报警」和「伺服放大器电机过热报警」。此情况下，调节速度和加减速比例，请在不会发生报警的条件下使用。

【关连参数】

JRCEXE

设定JRC的的执行可/不可。

执行不可=0(初始值)/执行可=1

JRCQTT

附加轴、使用者定义机器的时候，以JRC指令指定增减变化量（1deg/1mm单位）。机器人本体侧的JRC对象轴的情况不会是这个设定，固定在360度。

JRCORG (JRC 0执行时的原点/浮点型/要素数8/使用者参数)

在附加轴、使用者机器的时候，将JRC 0指定为执行时的原点坐标值。

各参数的详细内容请参照第407页的「5参数的设定功能」。

LoadSet (Load Set)

【功能】

指定 Oadl 指令执行时的抓手・工件的条件。

此外，使用干涉回避功能时，指定抓手号码、工件号码。（指定干涉确认的对象模型）

【格式】

LoadSet □ < 抓手条件号码 >, < 工件条件号码 >

【用语】

< 抓手条件号码 >

0~8。

将指定重量・大小的抓手条件参数 (HNDDAT0~8) 的号码以常数或变量指定。

同时在干涉回避功能中使用的抓手模型也被变更。对应参数 CAVKDH1~8 第一要素号码。

< 工件条件号码 >

0~8。

将指定重量・大小的工件条件参数 (WRKDAT0~8) 的号码以常数或变量指定。

同时在干涉回避功能中使用的工件模型也被变更。对应参数 CAVKDW1~8 第一要素号码。

【例子】

1 Oadl ON

2 LoadSet 1,1 ' 抓手 1 (HNDDAT1) ・ 工件 1 (WRKDAT1) 的条件

3 Mov P1

4 LoadSet 0,0 ' 抓手 0 (HNDDAT1) ・ 工件 0 (WRKDAT0) 的条件

5 Mov P2

6 Oadl Off

【说明】

- (1) 设定在最佳加减速使用抓手、及工件的条件。在每个重量不同工件上设定最佳加减速的情况下使用。此外，也与干涉回避功能中使用的抓手、工件号码联动。
- (2) 程序执行开始时，抓手会变成被设定最大负载状态。
此外，干涉回避功能中使用的抓手、工件的模型以初始值中设置的模型为对象。（CAVKDH1~8（抓手模型）、CAVKDW1~8（工件模型）各自的第1要素中已设定"0"的模型。）
- (3) 抓手条件以参数 (HNDDAT0~8) 指定重量、大小 (X,Y,Z)、重心位置 (X,Y,Z)。
- (4) 工件条件以参数 (WRKDAT0~8) 指定重量、大小 (X,Y,Z)、重心位置 (X,Y,Z)。
- (5) 以本指令执行并变更抓手条件和工件条件程序复位时、及在 END 文执行时，系统初始值会被再度设定。系统初始值使用 HNDDAT0、WRKDAT0。
- (6) 最佳加减速用的详细设定请参照第 456 页的「5.16 关于抓手、工件条件设定（最佳加减速设定）」。
关于干涉回避功能的详细设定，请参照第 481 页的「5.24 关于干涉回避功能」。

【相关指令】

Oadl (Optimal Acceleration)、HOpen/HClose (抓手 Open/Close)

【关连参数】

关于 HNDDAT0~8、WRKDAT0~8、HNDHOLD0~8 请参照第 456 页的「5.16 关于抓手、工件条件设定（最佳加减速设定）」。

关于 ACCMODE 请参照第 407 页的「表 5-1: 动作关连参数一览表」。

关于与干涉回避功能相关的参数，请参照第 481 页的「5.24 关于干涉回避功能」。

Mov (Move)

【功能】

从现在位置到移动目的位置执行关节插补位置。

【格式】

```
Mov □ < 移动目的位置 > [, < 接近距离 >]
      [ □类型□ < 常数 1>, < 常数 2>] □ [< 附加条件 >]
```

【用语】

- <移动目的位置> 补插补动作的最终位置。以位置型的变量和常数或关节变量记述。
- <接近距离> 指定此值的情况下,实际的移动目的位置会往以TOOL坐标的Z轴方向 (+/-方向)往指定距离的偏离位置动作。以常数或变量指定。
- <常数1> 1/0: 绕道 / 走近路方式动作。初始值为1 (绕道)。只有常数。
- <常数2> 无效 (请记述0)。只有常数。
- <附随条件> 可以使用Wth、WthIF。

【例子】

- 1 Mov P1 类型 1,0
- 2 Mov J1
- 3 Mov (Plt 1,10),100.0 Wth M_Out(17)=1
- 4 Mov P4+P5,50.0 类型 0,0 WthIf M_In(18)=1,M_Out(20)

【说明】

- (1) 将起点位置和终点位置的各轴关节角度做均等插补。因此尖端的轨迹无法保证。
- (2) 依据 Wth、WthIf 的并用,可以得到信号输出时序和动作的同步。
- (3) 类型的数值常数 1 为指定姿势的插补方式。
- (4) 在关节插补里称的绕道,是指以示教姿势做动作的意思。会有因示教时的姿势而变成绕道动作的情况。
- (5) 所谓走近路是指在起点・终点间的姿势,在动作量少的方向进行姿势的插补。
- (6) 绕道 / 走近路的指定,是指开始位置和目的位置的动作范围,有 ± 180 度上的移动量的意思。
- (7) 即使在有指定走近路的情况下,目的位置在动作范围外的時候,也会往返方向绕道动作。
- (8) 在关节插补,类型的数值常数 2 没有意义。
- (9) 通常执行程序内无法使用。
- (10) 在 Mov 指令执行时中断, JOG 前进,然后再开启的情况下,会返回到中断的位置再开始执行 Mov 指令。返回到中断位置插补方法 (关节插补 / 直线插补) 可以用 RETPATH 参数变更。此外,依据此 RETPATH 参数的变更,可以不返回到中断位置,而直接往目的位置移动。(请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」)。

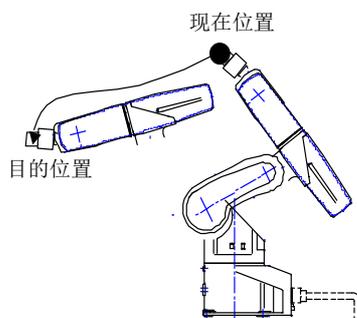


图 4-20: 关节插补的动作轨迹例

Mva (Move Arch)

【功能】

从现在位置以弧形运动动作（弧形插补）移动到目的位置。

【格式】

Mva □ < 移动目的位置 > [, < 弧形号码 >]

【用语】

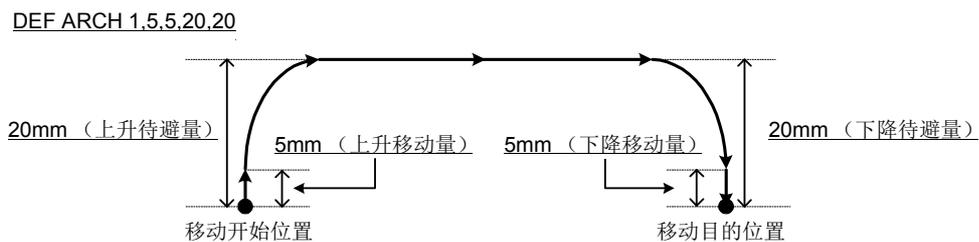
< 移动目的位置 > 插补动作的最终位置。以位置型的变量和常数或关节变量记述。
 < 弧形号码 > Def Arch (1~4)。省略时为1。

【例子】

- 1 Def Arch 1,5,5,20,20 ’ 定义弧形形状
- 2 OvrD 100,20,20 ’ 指定速度比例
- 3 Accel 100,100,50,50,50,50 ’ 设定加减速比例
- 4 Mov P0 ’ 往弧形运动动作的开始位置移动。
- 5 Mva P1,1 ’ 以步号 1 已定义的形状执行弧形运动动作。
- 6 Mva P2,2 ’ 以参数的初始值动作。

【说明】

- (1) 从现在位置在 Z 轴方向上升，持续的往目的位置的上空移动，且最后到下降到目的位置为止的动作。将所谓的弧形运动动作在一个指令里执行。
- (2) 在没有 Def Arch 指令下执行 Mva 指令的时候，以在参数设定的弧形形状动作。关于参数请参照第 192 页的「Def Arch」。
- (3) 因为插补形式、插补种类等也是以 Def Arch 指令定义，因此请参照第 192 页的「Def Arch」。
- (4) 通常执行程序内无法使用。
- (5) 在 Mva 指令执行中，中断 JOG 前进，然后再开启的情况下，会返回到中断的位置再开始执行 Mva 指令。返回到中断位置插补方法（关节插补 / 直线插补）可以用 RETPATH 参数变更。此外，依据此 RETPATH 参数的变更，可以不返回到中断位置，而直接往目的位置移动。（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。



※ 移动开始位置和目的位置的 Z 不同的情况下，会如下列的动作。

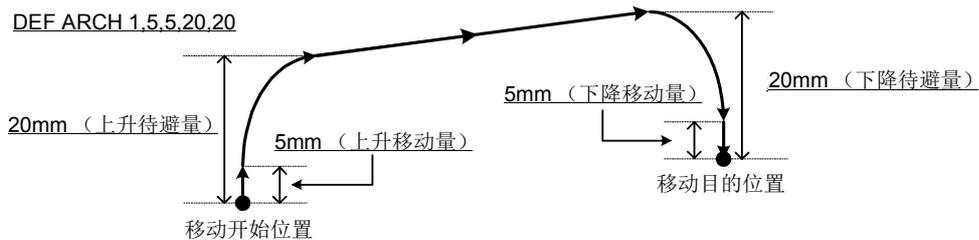


图 4-21：弧形插补的动作轨迹（从横面看的情况）



机器人根据当时的指定速度，可能会以不同的轨迹动作。尤其是转角部分，其内环距离有可能会发生变化，因此，在最初开始自动运行时应先以低速运行，然后一边留意与周边装置之间的干涉，一边逐渐提升运行速度。

【相关指令】

[Def Arch](#)、[Accel](#)、[Ovrd](#)

Mvc (Move C)

【功能】

以起点、通过点 1、通过点 2、起点的顺序执行三维圆弧插补动作。

【格式】

```
Mvc □ < 起点 >, < 通过点 1 >, < 通过点 2 > □ [< 附加条件 >]
```

【用语】

- < 起点 > 圆的起点及终点。记述位置型的变量和常数或关节型的变量。
- < 通过点 1 > 圆弧的通过点 1。记述位置型或关节型的常数及变量。
- < 通过点 2 > 圆弧的通过点 2。记述位置型或关节型的常数及变量。
- < 附随条件 > 记述 Wth 节或 WthIF 节。

【例文】

- 1 Mvc P1,P2,P3
- 2 Mvc P1,J2,P3
- 3 Mvc P1,P2,P3 Wth M_Out(17)=1
- 4 Mvc P3,(Plt 1,5),P4 Wthlf M_In(20)=1,M_Out(21)=1

【说明】

- (1) 圆插补动作是从被授予的 3 点开始求圆，移动圆周。(360 度)
- (2) 圆插补中的姿势会维持起点的姿势。通过点 1，2 的姿势不被考虑。
- (3) 现在位置和起点不一致的情况下，会自动的以直线插补（三轴直交插补）移动到起点为止，之后进行圆插补。
- (4) 中断执行，之后以 JOG 移动后再开启的情况下，会以关节插补移动到执行中断的位置为止，之后，再开始剩余的圆弧插补。返回到中断位置插补方法（关节插补 / 直线插补）可以用 RETPATH 参数变更。此外，依据此 RETPATH 参数的变更，可以不返回到中断位置，而直接往目的位置移动。（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。
- (5) 通常执行程序内无法使用。

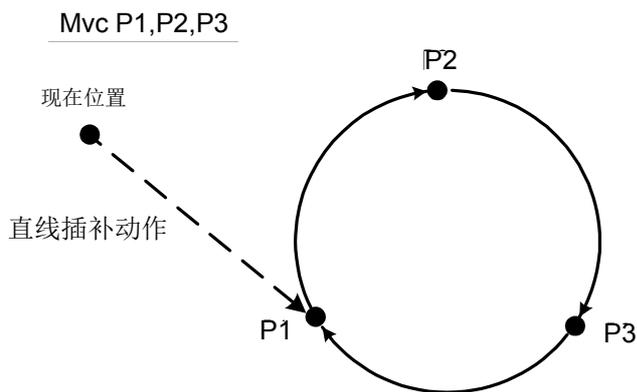


图 4-22: 圆插补的动作轨迹例

Mvr (Move R)

【功能】

从起点开始，经由通过点到终点为止，执行三维圆弧插补动作。

【格式】

Mvr □ < 起点 >, < 通过点 >, < 终点 >
 [□ Type □ < 常数 1 >, < 常数 2 >] □ [< 附加条件 >]

【用语】

< 起点 > 圆弧的起点。记述位置型的变量和常数、或关节变量。
 < 通过点 > 圆弧的通过点。记述位置型的变量和常数、或关节变量。
 < 终点 > 圆弧的终点。记述位置型的变量和常数、或关节变量。
 < 常数 1 > 走近路 / 绕道动作 = 0/1、初始值为 0。
 < 常数 2 > 等量旋转 / 三轴直交 / 特异点通过 = 0/1/2, 初始值为 0。
 < 附加条件 > 记述 Wth 或 Wthlf 节。

【例子】

```
1 Mvr P1,P2,P3
2 Mvr P1,J2,P3
3 Mvr P1,P2,P3 Wth M_Out(17)=1
4 Mvr P3,(Pit 1,5),P4 Wthlf M_In(20)=1,M_Out(21)=1
```

【说明】

- (1) 圆弧插补动作是从被授予的 3 点开始求圆，在那个圆弧上移动。
- (2) 姿势会变成从起点开始往终点的插补，通过点的姿势没有影响。
- (3) 现在位置和起点不一致的情况下，会自动的以直线插补（三轴直交插补）移动到起点为止。
- (4) 中断执行，之后以 JOG 移动后再开启的情况下，会以关节插补移动到执行中断的位置为止，之后，再开始剩余的圆弧插补。返回到中断位置插补方法（关节插补 / 直线插补）可以用 RETPATH 参数变更。请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」。
- (5) 等量旋转（常数 2=0）的情况下，起点和终点的构造标志不同时，执行时会发生异常。
- (6) 指定的 3 点内有相同位置的情况和 3 点在一直在线的情况时，会执行起点往终点的直线插补动作。不会发生报警。
- (7) 以常数 2 指定三轴直交的情况下，常数 1 无效并且以示教的姿势移动。
- (8) 常数 2 指定姿势插补的类型。由于三轴直交是在 (X、Y、Z、J4、J5、J6) 的坐标系上进行插补，因此通过特异点旁边时比较有效。仅有限定的机型可以支持特异点通过。关于详细内容请参照第 467 页的「5.20 关于特异点通过功能」。
- (9) 通常执行程序内无法使用。

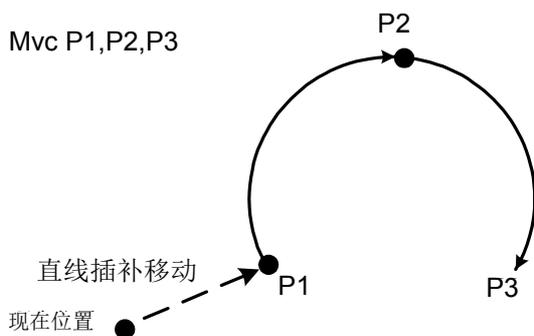


图 4-23: 圆弧插补的动作轨迹例其中之一

◇◆◇垂直 5 轴机器人中直线 (圆弧) 插补执行后的位置数据登录时的注意点◇◆◇

直线 (圆弧) 插补中, 动作开始位置的姿势数据与目的位置的姿势数据相差 +180 度或 -180 度以上时, 机器人会抄近道进行动作。此时, 垂直 5 轴机器人 (RV-4FJL) 的情况下, 到达目的位置后, 虽然姿势数据会显示登录的值, 但实际上姿势会偏移 +360 度或 -360 度。

例如, A 轴的值从 0 度的位置向 +200 度的位置进行直线插补时, 当前位置的姿势数据会显示为 +200 度, 但机器人会在 -160 度的位置动作。

如在该状态下登录位置数据, 则会登录与实际相差 +360 度或 -360 度的姿势数据。因程序的调试操作 (单步进给、MS 位置移动) 等导致在直线 (圆弧) 插补动作后要补偿位置数据时, 应加以注意。

此外, 通过用直线 (圆弧) 插补指令的自变量 Type 来指定绕远道动作, 或登录位置数据时暂时让伺服 OFF, 则可回避该现象。

Mvr2 (Move R2)

【功能】

从起点、终点、参考点开始所构成的圆弧上，从起点到终点为止，执行三维元圆弧插补动作。移动方向不通过参考点。

【格式】

Mvr2 □ < 起点 >, < 终点 >, < 参考点 >
 [□ Type □ < 常数 1 >, < 常数 2 >] □ [< 附加条件 >]

【用语】

< 起点 > 圆弧的起点。记述位置型的变量和常数、或关节变量。
 < 终点 > 圆弧的终点。记述位置型的变量和常数、或关节变量。
 < 参考点 > 圆弧的参考点。记述位置型的变量和常数、或关节变量。
 < 常数 1 > 走近路 / 绕道动作=0/1、初始值为0。
 < 常数 2 > 等量旋转 / 三轴直交 / 特异点通过=0/1/2, 初始值为0。
 < 附加条件 > 记述Wth或Wthlf节。

【例子】

1 Mvr2 P1,P2,P3
 2 Mvr2 P1,J2,P3
 3 Mvr2 P1,P2,P3 Wth M_Out(17)=1
 4 Mvr2 P3,(Plt 1,5),P4 Wthlf M_In(20)=1,M_Out(21)=1

【说明】

- (1) 圆插补动作是从被授予的 3 点开始求圆，在那个圆弧上移动。
- (2) 姿势会变成从起点开始往终点的插补，通过点的姿势没有影响。
- (3) 现在位置和起点不一致的情况下，会自动的以直线插补（三轴直交插补）移动到起点为止。
- (4) 中断执行，之后以 JOG 移动后再开启的情况下，会以关节插补移动到执行中断的位置为止，之后，再开始剩余的圆弧插补。返回到中断位置插补方法（关节插补 / 直线插补）可以用 RETPATH 参数变更。（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。
- (5) 移动方向为不通过参考点的方向。
- (6) 等量旋转（常数 2=0）的情况下，起点和终点的构造标志不同时，执行时会发生异常。
- (7) 指定的 3 点内有相同位置的情况和 3 点在一直在线的情况时，会执行起点往终点的直线插补动作。不会发生报警。
- (8) 以常数 2 指定三轴直交的情况下，常数 1 无效并且以示教的姿势移动。
- (9) 常数 2 指定姿势插补的类型。由于三轴直交是在 (X、Y、Z、J4、J5、J6) 的坐标系上进行插补，因此通过特异点旁边时比较有效。仅有限定的机型可以支持特异点通过。关于详细内容请参照第 467 页的「5.20 关于特异点通过功能」。
- (10) 通常执行程序内无法使用。

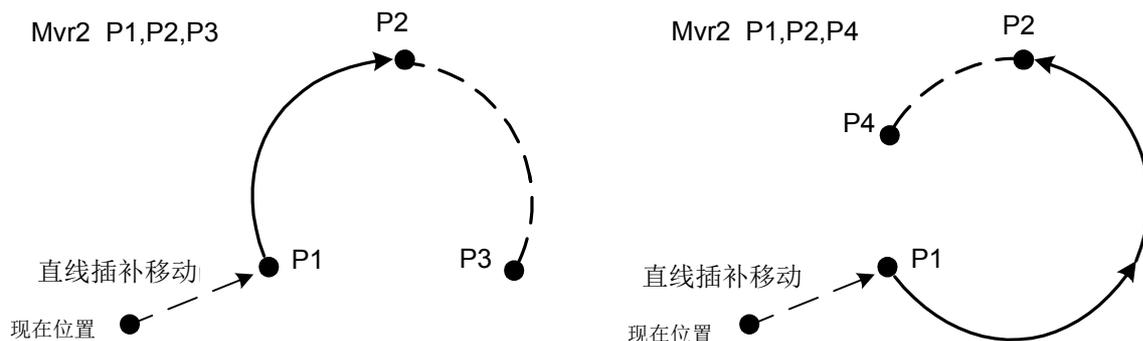


图 4-24: 圆弧插补的动作轨迹例子其 2

◇◆◇垂直 5 轴机器人中直线 (圆弧) 插补执行后的位置数据登录时的注意点◇◆◇

直线 (圆弧) 插补中, 动作开始位置的姿势数据与目的位置的姿势数据相差 +180 度或 -180 度以上时, 机器人会抄近道进行动作。此时, 垂直 5 轴机器人 (RV-4FJL) 的情况下, 到达目的位置后, 虽然姿势数据会显示登录的值, 但实际上姿势会偏移 +360 度或 -360 度。

例如, A 轴的值从 0 度的位置向 +200 度的位置进行直线插补时, 当前位置的姿势数据会显示为 +200 度, 但机器人会在 -160 度的位置动作。

如在该状态下登录位置数据, 则会登录与实际相差 +360 度或 -360 度的姿势数据。因程序的调试操作 (单步进给、MS 位置移动) 等导致在直线 (圆弧) 插补动作后要补偿位置数据时, 应加以注意。

此外, 通过用直线 (圆弧) 插补指令的自变量 Type 来指定绕远道动作, 或登录位置数据时暂时让伺服 OFF, 则可回避该现象。

Mvr3 (Move R3)

【功能】

从起点、终点、中心点开始所构成的圆弧上，从起点到终点为止，执行三维元圆弧插补动作。

【格式】

Mvr3 □ < 起点 >, < 终点 >, < 中心点 >
 [□ Type □ < 常数 1 >, < 常数 2 >] □ [< 附加条件 >]

【用语】

< 起点 > 圆弧的起点。记述位置型的变量和常数、或关节变量。
 < 终点 > 圆弧的终点。记述位置型的变量和常数、或关节变量。
 < 中心点 > 圆弧的中心点。记述位置型的变量和常数、或关节变量。
 < 常数 1 > 走近路 / 绕道动作 = 0/1、初始值为 0。
 < 常数 2 > 等量旋转 / 三轴直交 / 特异点通过 = 0/1/2, 初始值为 0。
 < 附加条件 > 记述 Wth 或 Wthlf 节。

【例子】

```
1 Mvr3 P1,P2,P3
2 Mvr3 P1,J2,P3
3 Mvr3 P1,P2,P3 Wth M_Out(17)=1
4 Mvr3 P3,(Plt 1,5),P4 Wthlf M_In(20)=1,M_Out(21)=1
```

【说明】

- (1) 圆插补动作是从被授予的 3 点开始求圆，在那个圆弧上移动。
- (2) 姿势会变成从起点开始往终点的插补，中心点的姿势没有影响。
- (3) 现在位置和起点不一致的情况下，会自动的以直线插补（三轴直交插补）移动到起点为止。
- (4) 中断执行，之后以 JOG 移动后再开启的情况下，会以关节插补移动到执行中断的位置为止，之后，再开始剩余的圆弧插补。返回到中断位置插补方法（关节插补 / 直线插补）可以用 RETPATH 参数变更。（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。
- (5) 等量旋转（常数 2=0）的情况下，起点和终点的构造标志不同时，执行时会发生异常。
- (6) 以常数 2 指定三轴直交的情况下，常数 1 无效并且以示教的姿势移动。
- (7) 常数 2 指定姿势插补的类型。由于三轴直交是在 (X、Y、Z、J4、J5、J6) 的坐标系上进行插补，因此通过特异点旁边时比较有效。仅有限的机型可以支持特异点通过。关于详细内容请参照第 467 页的「5.20 关于特异点通过功能」。
- (8) 起点开始到终点为止的中心角为， $0 < \text{中心角} < 180$ 度。
- (9) 请指定中心点开始到起点为止的距离和，中心点开始到终点为止的距离差在 0.01mm 以内的位置。
- (10) 3 点在同一直在线的情况、起点和中心点或终点和中心点相同的情况下，会发生报警。
- (11) 起点和终点为同相情况、3 点为相同的情况不会发生报警，会执行下一个指令。
- (12) 通常执行程序内无法使用。

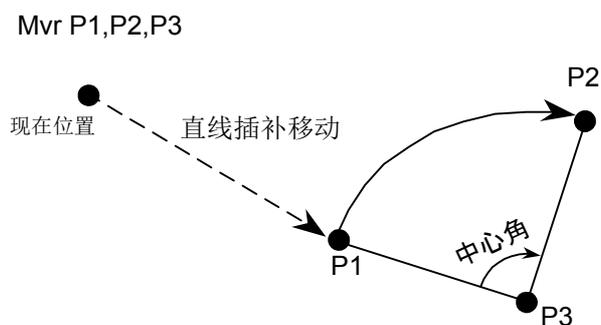


图 4-25: 圆弧插补的动作轨迹例子其 3

◇◆◇垂直 5 轴机器人中直线 (圆弧) 插补执行后的位置数据登录时的注意点◇◆◇

直线 (圆弧) 插补中, 动作开始位置的姿势数据与目的位置的姿势数据相差 +180 度或 -180 度以上时, 机器人会抄近道进行动作。此时, 垂直 5 轴机器人 (RV-4FJL) 的情况下, 到达目的位置后, 虽然姿势数据会显示登录的值, 但实际上姿势会偏移 +360 度或 -360 度。

例如, A 轴的值从 0 度的位置向 +200 度的位置进行直线插补时, 当前位置的姿势数据会显示为 +200 度, 但机器人会在 -160 度的位置动作。

如在该状态下登录位置数据, 则会登录与实际相差 +360 度或 -360 度的姿势数据。因程序的调试操作 (单步进给、MS 位置移动) 等导致在直线 (圆弧) 插补动作后要补偿位置数据时, 应加以注意。

此外, 通过用直线 (圆弧) 插补指令的自变量 Type 来指定绕远道动作, 或登录位置数据时暂时让伺服 OFF, 则可回避该现象。

Mvs (Move S)

【功能】

以直线插补动作从现在位置到移动目的位置为止。

【格式 1】

```
Mvs □ <移动目的位置> [, <近接距离>
      [ □ Type □ <常数 1>, <常数 2>] □ [<附加条件>]
```

【格式 2】

```
Mvs □, <脱离距离> [ □ Type □ <常数 1>, <常数 2>] □ [<附加条件>]
```

【用语】

- <移动目的位置> 插补动作的最终位置。记述位置型的变量和常数、或关节变量。
- <近接距离> 指定这个值的情况下,实际的移动目的位置为TOOL坐标的Z轴方向(+/-方向)往指定距离的偏离位置移动。
- <常数1> 走近路 / 绕道动作= 0/1、初始值为0。
- <常数2> 等量旋转 / 三轴直交 / 特异点通过= 0/1/2, 初始值为0。
- <附加条件> 记述Wth或Wthlf节。
- <脱离距离> 指定这个值的情况下,从现在的位置往TOOL坐标的Z轴方向(+/-方向)往指定距离的偏离位置移动。

【例子】

① 往目的位置P1直线插补移动

```
1 Mvs P1
```

② 往目的位置P1直线插补移动和同时将输出信号17开启

```
1 Mvs P1,100.0 Wth M_Out(17)=1
```

③ 往目的位置P4+P5(依据加算的相对运算位置)的TOOL坐标Z方向50mm,在直线插补移动中输入信号18开启的话,会开启输出信号20

```
2 Mvs P4+P5, 50.0 Wthlf M_In(18)=1, M_Out(20)=1
```

④ 从现在地往TOOL坐标Z方向直线插补移动50mm

```
3 Mvs ,50
```

【说明】

- (1) 直线插补是从机器人的现在位置开始往移动目的位置为止的控制点轨迹为直线的移动状态。
- (2) 姿势为从起点开始往终点的插补。
- (3) 在〈接近距离〉和〈脱离距离〉所指定的 TOOL 坐标依机器人的机型, Z 的 +/- 方向会不同。详细内容请参照第 432 页的「5.6 关于标准 TOOL 坐标」。

「图 4-26: 直线插补时的动作例」是一个例子。

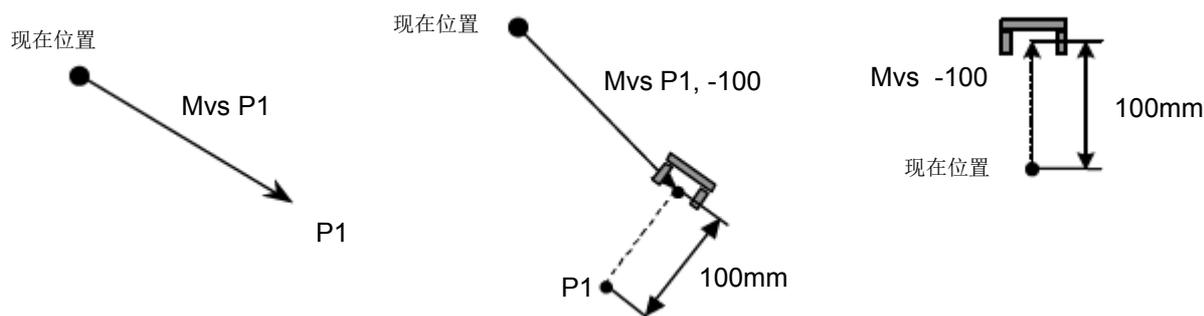
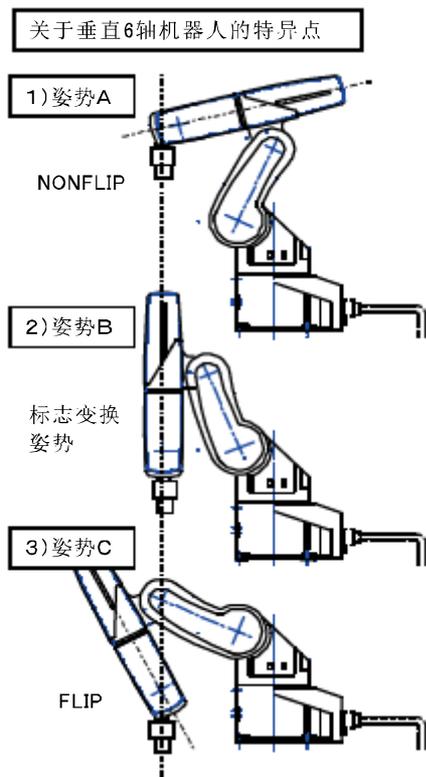


图 4-26: 直线插补时的动作例

- (4) 抄近道动作是指，关于起点・终点间的姿势，在动作量少的方向上进行姿势的插补。
- (5) 绕道动作是指，关于起点・终点间的姿势，在动作量多的方向上进行姿势的插补。
- (6) 指定绕道 / 抄近道时，目的位置在动作范围外时，会发生动作范围外的错误。
- (7) 中断执行，之后以 JOG 移动后再开启的情况下，会以关节插补移动到执行中断的位置为止，之后，再开始剩余的圆弧插补。返回到中断位置插补方法（关节插补 / 直线插补）可以用 RETPATH 参数变更。（请参照第 442 页的「5.10 中断时的 JOG 前进后的自动回归设定」）。
- (8) 通常执行程序内无法使用。
- (9) 等量旋转（常数 2=0）的情况下，起点和终点的构造标志不同时，执行时会发生异常。
- (10) 以常数 2 指定三轴直交的情况下，常数 1 无效并且以示教的姿势移动。
- (11) 常数 2 指定姿势插补的类型。由于三轴直交是在 (X、Y、Z、J4、J5、J6) 的坐标系上进行插补，因此通过特异点旁边时比较有效。仅有限的机型可以支持特异点通过。关于详细内容请参照第 467 页的「5.20 关于特异点通过功能」。
- (12) 特异点的说明



< 垂直 6 轴机器人的的情况 >

从姿势 A 通过姿势 B 到姿势 C 是无法以一般的直线插补 (Mvs) 动作。

姿势 A、B、C 都被限制在 J4 轴为 0 度的情况下，但是，姿势 A 的 J5 轴（手腕轴）的构造为 FLIP、姿势 C 为 NONFLIP。另外，因为姿势 B 的手腕为伸长状态，因此在同一直线上，因为有 J4、J6 轴，因此机器人无法做直线插补的位置运算。

想要在这样的姿势上做直线插补的情况下，请在 Mvs 的指令操作使用三轴直交（类型 0,1）方式（或特异点通过（类型 0,2）过方式）。但是，此轴直交为了将姿势 A 和姿势 C 的 J4、J5、J6 轴的关节角度做均等的插补，严格说起来，并没有保持姿势。因此可以预知的是移动途中，机器人抓手的姿势会前后动作。

此情况下，在途中追加 1 个点的话，可以减轻抓手姿势的变化量。

另外，在机器人坐标的 Z 轴上，有 J5 轴的中心点，手腕朝上的情况下也同样的会变成特异点。此状况下，J1 轴和 J6 轴被定位在同一轴上，可藉由运算得到机器人的位置。

图 4-27: 特异点其 1

◇◆◇垂直 5 轴机器人中直线 (圆弧) 插补执行后的位置数据登录时的注意点◇◆◇

直线 (圆弧) 插补中，动作开始位置的姿势数据与目的位置的姿势数据相差 +180 度或 -180 度以上时，机器人会抄近道进行动作。此时，垂直 5 轴机器人 (RV-F 系列) 的情况下，到达目的位置后，虽然姿势数据会显示登录的值，但实际上姿势会偏移 +360 度或 -360 度。

例如，A 轴的值从 0 度的位置向 +200 度的位置进行直线插补时，当前位置的姿势数据会显示为 +200 度，但机器人会在 -160 度的位置动作。

如在该状态下登录位置数据，则会登录与实际相差 +360 度或 -360 度的姿势数据。因程序的调试操作 (单步进给、MS 位置移动) 等导致在直线 (圆弧) 插补动作后要补偿位置数据时，应加以注意。

此外，通过用直线 (圆弧) 插补指令的自变量 Type 来指定绕远道动作，或登录位置数据时暂时让伺服 OFF，则可回避该现象。

MvSpl (Move Spline)

【功能】

基于登录在指定的样条文件中的信息，执行样条插补。

(关于样条插补的详细内容，请参照第 587 页的「7.4 关于样条插补」。)

注) 该指令可在软件版本为 Ver.R5 以上 (F-Q 系列) / Ver.S5 以上 (F-D 系列) 的版本中使用。

【格式】

```
MvSpl □ < 样条号码 >, < 速度 >, < 加减速距离 >
      [, < 帧转换 >[, < 姿势插补类型 >[, < 滤波器长度 >]]]
```

【用语】

- | | |
|----------|---|
| <样条号码> | 以常数或数值变量指定保持有希望使其动作的路径信息的样条文件的号码。
设定范围：1~99 |
| <速度> | 以常数或数值变量指定样条插补的动作速度。
设定范围：比0.0大的数值 (mm/s)，详细内容请参照【说明】(5)。 |
| <加减速距离> | 以常数或数值变量指定样条插补的加减速距离。
加速距离为从动作开始加速至指定速度所需要的距离。
减速距离为从指定速度开始减速至结束位置停止所需要的距离。
样条插补中，加速距离与减速距离为通用的设定。
设定范围：比0.0大的数值 (mm) |
| <帧转换> | 以常数或数值变量指定帧转换的执行内容。
设定范围：
0：不执行帧转换。
1：使用样条文件中设定的坐标系执行帧转换。
2：使用SetCalFrm指令中设定的坐标系执行帧转换。
省略时：不执行帧转换。 |
| <姿势插补类型> | 以常数或数值变量指定姿势的插补类型。
设定范围：
0 (等效旋转动作) / 1 (三轴直交动作)
省略时：等效旋转动作 |
| <滤波器长度> | 以常数或数值变量指定施加在加减速动作上的滤波器长度。
设定范围：0.0~1000.0 (ms)
省略时：100.0ms |

【例文】

- | | |
|----------------------|--|
| 1 OvrD 70 | ' 将程序倍率修调设定为 70% |
| 2 Mov P1 | ' 通过关节插补向位置 P1 的位置移动 |
| 3 MvSpl 2, 50, 10 | ' 执行通过样条文件 2 的路径点的样条插补 |
| 4 Mvs P2 | ' 通过直线插补向位置 P2 的位置移动 |
| 5 MvSpl 2, 50, 10, 1 | ' 通过对样条文件 2 的路径点实施了帧转换的路径点
' 执行样条插补 |

【说明】

- (1) 基于登录在 < 样条号码 > 对应的样条文件中的路径点数据等信息，执行样条插补。生成通过作为路径点登录的机器人位置·姿势的顺畅曲线 (样条曲线)，沿着该曲线动作。
- (2) MvSpl 指令开始执行时的机器人当前位置与样条插补的开始位置有偏离时，以直线插补动作至开始位置后，开始执行样条插补。
- (3) < 样条号码 > 对应的样条文件未保存在控制器中的情况下，会发生 L2610(无法打开样条文件) 错误。

(4) 样条插补的指令速度，指定为以下算式。

指令速度 = (MvSpl 指令的 <速度>) × (Ovrd 指令) × (操作面板的倍率修调)

不使用 Spd 指令与 JOvrd 指令的设定。此外，即使 <速度> 中设定状态变量 M_NSpd，最佳速度控制模式也不起作用。

(5) <速度> 的设定范围会根据登录在样条文件中的路径点间的距离而发生变化。详细内容请参照第 592 页的「表 7-27: 路径点相关的检查」。

(6) 样条插补的加减速通过 <加减速距离> 来指定。不使用 Accel 指令的设定。此外，即使通过 Oadl 指令将最佳加减速控制设定为有效，也不适用于样条插补。

(7) 加速动作将生成速度，使从样条插补的开始位置移动 <加减速距离> 即达到 <速度>。减速动作将生成速度，使从 <速度> 移动 <加减速距离> 以在结束位置停止。

将 Ovrd 指令 · 操作面板的倍率修调设定为比 100% 小的数值时，以比 <加减速距离> 更短的移动量达到指令速度。

(因滤波器等的影晌，实际上是比通过 <加减速距离> 指定的数值更长的加减速距离。)

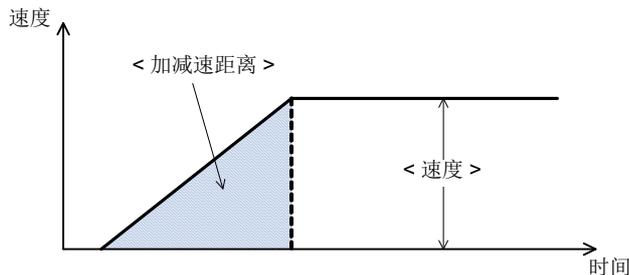


图 4-28: <加减速距离> 导致的加速

(8) <帧转换> 指定为 “1(使用样条文件中设定的坐标系执行)” · “2(使用通过 SetCalFrm 指令设定的坐标系执行)” 时，基于指定的方法依次转换路径点数据，执行通过该路径点的样条插补。

(9) 即使 <帧转换> 已指定为 “1(使用样条文件中设定的坐标系执行)”，当样条文件中未设定坐标系时，也会发生 L2042(帧转换坐标未设定) 错误。

(10) <帧转换> 指定为 “1(使用样条文件中设定的坐标系执行)” 时，无法算出坐标系的情况下，会发生 L2041(帧转换坐标无法算出) 错误。

(11) <帧转换> 指定为 “2(使用通过 SetCalFrm 指令设定的坐标系执行)” 时，则使用通过最后执行的 SetCalFrm 指令设定的坐标系执行帧转换。从未执行过 SetCalFrm 指令且未设定坐标系时，会发生 L2042(帧转换坐标未设定) 错误。

(12) 通过 <姿势插补类型> 的指定，可以切换 “0(等效旋转动作)” 与 “1(三轴直交动作)”。

等效旋转动作，以路径点间的姿势变化为最小的角度进行动作。

三轴直交动作通过 J4、J5、J6 轴角度、而非 A、B、C 轴坐标值来生成姿势数据。因此，路径点的姿势数据 (A、B、C 轴坐标值) 即使相同，动作过程中机器人的姿势也会变化。

(13) 不支持结构标志的跨越 (特异点通过)。路径点间结构标志不相同，会发生 L2611(路径点的结构标志不相同) 错误。

(14) 可以更改通过 <滤波器长度> 施加在加减速动作上的滤波器特性。希望加速 · 减速时的动作更为顺畅时，可加长滤波器长度。但是，加得过长则动作会变缓慢，到样条插补结束为止的时间会变长。

(15) 路径点数据中未设定信号输出的情况下，通过对象的路径点时，将以指定的条件输出通用输出信号。

(16) 指定了机器人的定位完成条件 (Fine 指令、Fine J 指令、Fine P 指令) 的情况下，在样条插补的结束位置上确认定位完成。

(17) 即使通过 Cnt 指令指定了连续动作，在样条插补的开始位置 · 结束位置上也不会进行连续动作。

(18) Def Act 指令 (中断的条件与处理的定义) 的 <类型> 即使指定为 “停止类型 1”，样条插补执行过程中发生了中断时，也将以与 “停止类型 2” 相同的减速停止。

(19) 附加轴不从样条插补的开始位置进行动作。中途的路径点上即使设定了附加轴的位置，也不用于样条插补。

- (20) 对 MvSpl 指令, 可以进行单步进给, 但无法进行单步退回。会发生 L2612 (无法执行单步退回) 错误。
- (21) 执行 MvSpl 指令需要机械的控制权 (GetM 指令)。
- (22) 启动条件无法通过 ALWAYS·ERROR 的插槽执行 MvSpl 指令。会发生 L2612 (该启动条件下无法执行) 错误。
- (23) MvSpl 指令的自变量中设定了超出设定范围的值时, 会发生 L3110(自变量的值超出范围 (MvSpl)) 错误。

注意

如指定了相对 < 速度 > 而言过短的 < 加减速距离 >, 则会变成急剧的加减速, 将导致机器人振动或发生伺服错误, 因此应设定适当的加减速距离。

【相关指令】

[SetCalFrm \(Set Calibration Frame\)](#)

【相关系统状态变量】

M_SplPno、M_SplVar

MvTune

【功能】

根据用途指定最佳的动作模式，可以提升机器人的动作性能。

指定的动作模式可从以下 4 种中任选一种。

另外，本指令根据以 LoadSet 指令指定的抓手 · 工件条件，将动作特性最佳化。请正确设定实际使用的抓手 · 工件的重量 · 形状 · 重心位置。

【格式】

MvTune □ < 动作模式 >

【用语】

< 动作特性模式 >

机器人的动作模式 (1~4) 以常数或数值变量指定。

1: 标准模式 (初始值)

2: 高速定位模式

3: 轨迹优先模式

4: 振动抑制模式

表 4-16: MvTune 的动作模式

动作模式		特征
1	标准模式 (初始值)	制造商标准设定。 适合于各种用途的标准动作特性。
2	高速定位模式	可缩短到达目的位置的时间。 请在希望缩短定位时间，提高作业效率时使用。 (用途: 跟踪作业、Palletiz 作业 等)
3	轨迹优先模式	可提高插补动作的轨迹精度。 请在对轨迹精度有较高要求的作业中使用。 (用途: 封口作业、熔敷作业、去毛刺作业 等)
4	振动抑制模式	具有抑制机器人机械臂振动 (共振) 的效果。 请在工件搬运过程中会发生振动的情况下使用。 (用途: 晶片搬运、精密部件搬运 等)

【例文】

```
LoadSet 1,1      ' 设定为抓手 1 · 工件 1
MvTune 2        ' 将动作模式变更为高速定位模式
Mov P1          ' 以高速定位模式动作
Mvs P1          ' 以高速定位模式动作
MvTune 3        ' 将动作模式变更为轨迹优先模式
Mvs P3          ' 以轨迹优先模式动作
```

【说明】

- (1) 根据以 LoadSet 指令指定的对应抓手 · 工件条件，将动作特性调整到最佳的状态。请正确设定实际使用的抓手 · 工件的重量 · 形状 · 重心位置。抓手 · 工件条件没有正确设定的话，可能无法达到最佳性能。(请参照第 456 页的「5.16 关于抓手、工件条件设定 (最佳加减速设定)」。)
- (2) 电源开启后，标准模式会指定为初始值。
- (3) 在程序结束 (End 指令执行、或中断中的程序复位) 时，动作模式会返回到标准模式。但是，在以 CallP 指令执行的子程序的 End 指令下，以及程序中断中的再启动，(也包含中断且程序编辑后的再启动)，动作模式会被保持。

(4) 各动作模式相对于标准模式的不同点如表 4-17 所示。

表 4-17: MvTune 标准模式的不同点

动作模式	比较项目			
	目标位置到达时间	轨迹精度 ^{注 1)}	振动抑制 ^{注 2)}	负载率 ^{注 3)}
1: 标准模式	○	○	○	○
2: 高速定位模式	◎	○ +	○	△
3: 轨迹优先模式	△	◎	○	○ +
4: 振动抑制模式	○ -	○ -	◎	○

(注) 表中的符号表示性能比较 Level 的程度。

◎: 提升、○+: 略微提升、○: 持平、○ -: 略微下降、△: 下降

注 1) 比较机器人对动作指令值的跟随性能

注 2) 比较对诱发振动的外部干扰的抑制能力

注 3) 比较电机的发热量

- (5) 最佳加减速控制（以 Oadl 指令或参数 ACCMODE 指定）为无效的情况下，以本指令执行会自动的变成有效。此外，在执行本指令后再执行 Oadloff 的情况下，只有最佳加减速控制会变无效，动作模式不变更。
- (6) 在高速定位模式中，加速·减速时的振动有时会比标准模式时更大。有故障的情况下，请选择标准模式。
- (7) 轨迹优先模式调整为通过中低速范围的动作速度可以得到最大的效果。因此，在高速执行描绘较小的圆的动作时，跟标准模式比起来会有振动增加的情况。此情况下请使用 Spd 指令降低动作速度，使振动变小。
- (8) 在振动抑制模式中，根据不同的使用条件，到达目的位置时的过冲量有可能会变大。在如等待定位完成的动作（使用了 Fine 指令的动作）中，节拍时间可能会出现延迟。（工件越重，过冲量越大）
- (9) 对于 JOG 动作不起作用。

【相关指令】

LoadSet (Load Set)、Oadl (Optimal Acceleration)、Prec (Precision)

【相关参数】

ACCMODE、HNDDAT0 ~ 8、WRKDAT0 ~ 8

Mxt

【功能】

每隔 1 个单位控制器控制时间（约 7.1msec），从以太网上连接的外部计算机等设备获取绝对位置数据，进行直接移动。

【格式】

Mxt < 文件号码 >, < 指令位置数据类型 >[, < 过滤器时间常数 >]

【用语】

<文件号码> 记述通过OPEN指令分配的1~8的号码。
未通过OPEN指令指定通信对象时，将出现错误，无法进行通信。
此外，通信对象以外的接收数据将被忽略。

<指令位置数据类型> 指定通过计算机等发出指令的位置数据的类型。
可指定直交 / 关节 / 电机脉冲的位置。
0: 直交坐标数据
1: 关节坐标数据
2: 电机脉冲坐标数据

<过滤器时间常数> 指定过滤器时间常数 (msec)。指定0时，为无过滤器。（省略时为0）
在接收位置数据上使用过滤器，创建低灵敏度的指令值，输出到伺服。

【例文】

10 Open "ENET: 192.168.0.2" AS #1	’ 设定以太网的通信对象的 IP 地址	
20 Mov P1	’ 向 P1 移动	
30 Mxt 1,1,50	’ 在 realtime 外部控制中，将过滤器时间常数设为 50msec，进行动作	
40 Mov P1	’ 向 P1 移动	
50 Hlt	’ 中断程序	

【说明】

- (1) 执行 Mxt 指令后，可以从网络上连接的计算机等设备读取用于动作控制的位置指令。（1 对 1 通信）
- (2) 1 个单位动作控制时间（约 7.1msec）可读取 1 个位置指令并进行动作。
- (3) Mxt 指令的动作
 - 1) 在控制器中执行本指令，控制器将变为可接收指令值的状态。
 - 2) 控制器在从计算机接收指令值后，将在接收指令值的下一个控制处理周期内将接收到的指令值输出到伺服。
 - 3) 将指令值发送到伺服后，控制器会向计算机发送现在位置等的控制器情报。
 - 4) 仅在从计算机向控制器发送了指令值的情况下，控制器才会向计算机回信。
 - 5) 没有收到数据时，保持现在位置。
 - 6) 接收到来自计算机的 realtime 外部指令结束的指令后，结束 Mxt 指令。
 - 7) 通过操作面板或外部输入停止后，将会中断 Mxt 指令，并中断收发信直至再启动为止。
- (4) 超时指定通过参数 MXTTOUT 进行。
- (5) 可在收发位置数据的同时一起收发 1 个任意指定（起始位·位宽）的输入输出信号。
- (6) 要在动作控制时间发出动作指令，需要处理速度足够快的计算机。
- (7) 推荐 WindowsXP 或 2000/Pentium III 1GHz 以上、主内存 512K 以上、控制台应用程序。

【相关指令】

Open

Oadl (Optimal Acceleration)

【功能】

对应机器人抓手的负载状态，自动执行最佳的加减速度的设定（最佳加减速控制）。

可以藉此缩短机器人的动作时间 (TACT)。

最佳加减速时的加减速速度如下列的计算公式：

$$\text{加减速速度 (秒)} = \text{最佳加减速速度 (秒)} \times \text{Accel 指令 (\%)} \times \text{M_SetAdl (\%)}$$

※ 最佳加减速速度是使用 Oadl 指令时所算出的最佳加减速时间。

【格式】

Oadl <On/Off>

【用语】

<On/Off>

On : 最佳加减速开始

Off : 最佳加减速结束

【例子】

```

1 Oadl On
2 Mov P1          ' 以最大负载动作
3 LoadSet 1, 1   ' 设定在抓手 1・工件 1
4 Mov P2          ' 以在抓手 1・工件 1 的负载动作
5 HOpen 1        '
6 Mov P3          ' 在抓手 1 的负载动作
7 HClose 1       '
8 Mov P4          ' 以在抓手 1+ 工件 1 的负载动作
9 Oadl Off

```

※ 参数 HNDHOLD1=0, 1 的情况

【说明】

- (1) 以 Loadset 指令选择对应抓手・工件条件，将动作特性调整到最佳的状态。
- (2) 抓手开闭时的工件把持 / 未把持的设定是以参数 HNDHOLD 1 ~ 8 设定。
- (3) Oadl 的初始设定可以依据参数 ACCMODE 做变更。（请参照第 407 页的「表 5-1: 动作关连参数一览表」）。
- (4) 将 Oadl 再次开启后，要到再度执行 Oadl Off 或执行程序 End 为止才会变有效。
- (5) 依据抓手和工件的状况，会比平常动作时间更延长的情况。
- (6) 可以依据 LoadSet 指令、Oadl 指令和 HNDDAT 1 (0) ~ 8、WRKDAT 1 (0) ~ 8 参数的设定值，执行最佳的加减速动作。这些参数的工厂出货时的设定会变成最大负载。请变更参数值以符合使用的条件。（请参照第 456 页的「5.16 关于抓手、工件条件设定（最佳加减速设定）」）。
- (7) 轴单位的加减速比例的值以参数 JADL 决定。S 系列依每个机型，值会有所不同。请参照第 414 页的「JADL」。

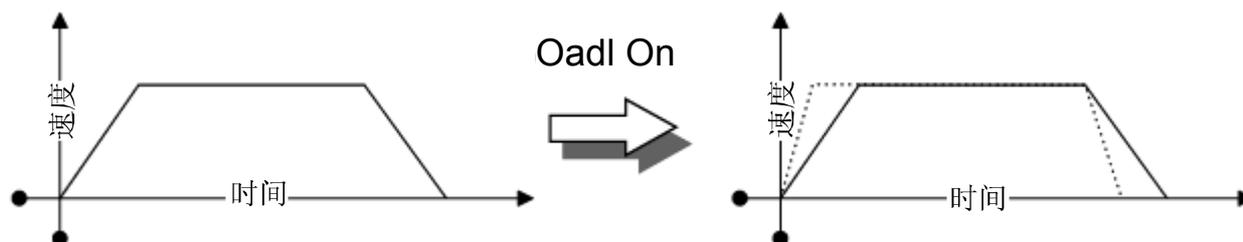


图 4-29: 轻负载时的加减速模式

【相关指令】

Accel、LoadSet (Load Set)、HOpen/HClose (抓手 Open/Close)

【关连参数】

HNDDAT0 ~ 8、WRKDAT0 ~ 8、HNDHOLD1 ~ 8、ACCMODE、JADL

On Com GoSub (ON Communication Go Subroutine)

【功能】

定义从指定的通信端口发生插入时，分支处理路径的开始行。

【格式】

```
On □ Com[(< 文件号码 >)] □ GoSub □ < 呼出对象 >
```

【用语】

<文件号码> 记述被分配在通信端口的1~3的号码。
<呼出对象> 记述LEVEL名。

【例子】

从文件号码 1 的通信端口 (COM1:) 的插入发生的话，执行 Level RECV 的处理。

```
1 Open "COM1:" As #1          ' 通信端口开启
2 On Com(1) GoSub *RECV      ' 插入的定义
3 Com(1) On                  ' 设定允许从文件号码 1 来的通信端口插入
4 '
:' << 此中间若发生通信的插入情况，往 Level RECV 分支。 >>
11 '
12 Mov P1
13 Com(1) Stop              ' 只从 P1 往 P2 移动，禁止插入。
14 Mov P2
15 Com(1) On                ' 在从 P1 到 P2 的移动中，有通信的情况下，在此发生插入
16 '
:' << 此中间若发生通信的插入情况，往 Level RECV 分支。 >>
26 '
27 Com(1) Off              ' 设定禁止从文件号码 1 来的通信端口插入
28 Close #1
29 End
:
:
40 *RECV                    ' 通信插入处理
41 Input #1, M0001          ' 接受的信息设定于 M0001、P0001。
42 Input #1, P0001
:
50 Return 1                  ' 返回到插入发生的下一行控制。
```

【说明】

- (1) 文件号码省略的情况下，文件号码 1 会被采用。
- (2) 插入的优先级以文件号码小的号码为优先。
- (3) 机器人动作中发生通信插入的话，在同一个插槽内动作的机器人会停止。依据 Com Stop 使插入停止，也可以让机器人不要停止。
- (4) 初始状态为插入禁止。使插入有效的话，请在本指令后执行 Com On 指令。
- (5) 在子程序请务必以 Return 返回。使用 Go To 返回的话，会使控制构造用内存（堆栈内存）减少，因内存不足而发生报警。

【相关指令】

Com On/Com Off/Com Stop (Communication ON/OFF/STOP)、Return、Open、Input、Print、Close

On...GoTo

【功能】

在指定值往对应 level 分支。

【格式】

On □ <式> □ GoTo □ [<分支处 >], [<分支处 >]...

【用语】

<式> 用数值运算式指定往第几号的LEVEL分支。
 <分支处> 记述LEVEL名。最大数为32。

【例子】

依据数值变量 M1 的值 (1 ~ 7) 分支。
(M1 为 1 的时候往 level ABC1 分支、为 2 的时候往 level LJMP 分支、为 3、4、5 的时候往 level LM1_345 分支、为 6、7 的时候往 level L67 分支)

```
10 On M1 GoTo *ABC1,*LJMP,*LM1_345,*LM1_345,*LM1_345,*L67,*L67
11 'M1 为 1 ~ 7 以外 (0 或 8 ~) 的时候, 移动到此行
```

```
100 *ABC1
101 ' 记述 M1=1 时候的处理
102 ' :
```

```
110 ' M1=2 的时候
111 *LJMP
112 ' 记述 M1=2 时候的处理
113 ' :
```

```
170 *L67
171 ' 记述 M1=6 或 M1=7 时候的处理
172 ' :
```

```
200 *LM1_345
201 ' 记述 M1=3、M1=4、M1=5 时候的处理
202 ' :
```

【说明】

- (1) 为 On GoSub 的 Go To 版。
- (2) <式> 的值比 <呼出处> 的个数大的情况下, 会跳转到下一行。例如 <式> 的值为 5, <呼出处> 只有写 3 个的情况下, 会跳转到下一行。
- (3) 呼出的 level 没有被使用的情况及 level 名重复被定义的情况下, 执行时会发生报警。

<式> 的值	处理 <控制 >
实数	将值四舍五入变换为整数后进行分支
0、或行号码 level 的个数超过的情况	控制移到下一次
负的数或比 32767 大	执行时报警

Open

【功能】

开启文件（CRnD 含通信端口）。

【格式】

Open □ "<文件名>" □ [For <模式>] □
As □ [#]<文件号码>

【用语】

<文件名> 记述文件名。
 • 在CRnD使用通信端口的情况下"<通信端口名>:"
 • 文件的情况下"<文件名>"

文件种类	文件名	连接方法种类
文件	在任意的 16 个文字以内记述	Input, Output, Append 通信端口
通信端口	COM1 : 在参数「COMDEV」的设定 : COM8 : 在参数「COMDEV」的设定	省略 = 只有随机模式
	ENET: 192.168.0.2 ^{注1)}	Mxt

注1) 是通过以太网接口使用 realtime 外部控制时的指定。"ENET:" 之后, 指定根据 Mxt 指令读取绝对位置数据的 IP 地址。

<模式> 指定往文件的连接方法。
 • 省略时: 使用随机模式、通信端口的情况下省略。
 • Input: 输入模式从已有的文件输入。
 • Output: 输出模式(新规文件)、制作新规文件, 然后输出。
 • Append: 输出模式(已有的文件)、已有的文件的最后追加输出。

<文件号码> 以1~8的常数记述。

【例子】

① 通信端口

```
1 Open "COM1:" As #1 ' 参数: 将在 COMDEV 的第 1 号的要素设定的通信端口以文件号码 1 开启。
2 Mov P_01
3 Print #1,P_Curr ' 将现在位置输出外部。 "(100.00,200.00,300,00,400.00)(7,0)" 的形式
4 Input #1,M1,M2,M3 ' 以 "101.00,202.00,303.00" 的 ASCII 形状从外部接受。
5 P_01.X=M1
6 P_01.Y=M2
7 P_01.C=Rad(M3) ' 复制到 Golbal 数据。
8 Close ' 关闭被开启的全部文件
9 End
```

② 文件操作 (在 R/C 内做成名称为 "txt.pmet" 的文件、写入 "abc")

```
1 Open "temp.txt" For Append As #1 ; 将名为 temp.txt 的文件在 Append 模式作为文件号码 1 开启。
2 Print #1, "abc" ; 在文件上写入 "abc"。
3 Close #1 ; 关闭文件。
```

【说明】

- (1) 在 (文件名) 记述文件利用指定的文件号码开启。
执行到文件的读写也需要使用这个文件号码。
- (2) 通信网络线也视为文件。

【相关指令】

Input、Print、Close

【关连参数】

COMDEV

Ovrd

【功能】

机器人动作的速度以 1 ~ 100% 指定。全部程序使用的速度比例。

【格式】

Ovrd □ < 速度比例 >

Ovrd □ < 速度比例 > [, < 上升时 > [, < 下降时速度比例 >]]

【用语】

<速度比例>

速度比例以实数指定。初始值为100。

单位: [%](范围: 0.01 ~ 100.0)

也可以用数值运算式记述。设定为0或100以上的话会发生报警。

<上升时/下降时速度比例>

指定在弧形运动指令(Mva)的上升、下降时的速度比例值。

【例子】

1 Ovrd 50

2 Mov P1

3 Mvs P2

4 Ovrd M_NOvrd ' 设定初始值

5 Mov P1

6 Ovrd 30,10,10 ' 弧形运动指令的上升、下降时的速度比例值设定为 10。

7 Mva P3,3

【说明】

(1) Ovrd 指令与插补的种类无关，为有效。

(2) 实际的速度比例如下所示。

- 关节插补动作时 = (操作面板 (T/B) 的速度比例设定值) × (程序速度比例 (Ovrd 指令)) × (关节速度比例 (JOvrd 指令))
- 直线插补动作时 = (操作面板 (T/B) 的速度比例设定值) × (程序速度比例 (Ovrd 指令)) × (直线指定速度 (Spd 指令))

(3) 速度比例指令只会使程序速度比例变化。100% 为机器人的能加最大值，通常系统初始值 (M_NOvrd) 会设定为 100%。程序中，在速度比例指令被执行为止，指定速度比例会采用系统初始值。

(4) 执行一次 Ovrd 指令的话，下次 Ovrd 指令会被执行，但是程序 End 的执行或程序复位为止会采用指定的速度比例。在 End 文的执行或程序复位会返回到初始值。

【相关指令】

JOvrd (J 速度比例) (关节插补用)、Spd (Speed) (直线插补、圆弧插补用)

【关连系统状态变量】

M_NOvrd(系统的初始值)、M_Ovrd(现在的指定速度)

Plt (码垛)

【功能】

运算码垛内的格子点位置。

【格式】

```
Plt □ < 码垛号码 >, < 格子点号码 >
```

【用语】

<码垛号码> 选择在Def Plt分设定完毕的1~8里，码垛号码。以变量或常数指定。

<格子点号码> 想求得的码垛内的位置号码。以变量或常数指定。

【例子】

1 Def Plt 1,P1,P2,P3,P4,4,3,1	'4 点码垛的定义 (P1,P2,P3,P4)
2 '	
3 M1=1	' M1(计数器)初始化
4 *LOOP	
5 Mov PICK, 50	' 往取出工件位置上空 50mm 移动
6 OvrD 50	
7 Mvs PICK	
8 HClose 1	' 抓手闭
9 Dly 0.5	' 抓手闭后等待 0.5 秒
10 OvrD 100	
11 Mvs,50	' 往现在位置上空 50mm 移动
12 PLACE = Plt 1, M1	' 计算第 M1 号的位置
13 Mov PLACE, 50	' 往 Pallt 上放置位置上空 50mm 移动
14 OvrD 50	
15 Mvs PLACE	
16 HOpen 1	' 抓手开
17 Dly 0.5	
18 OvrD 100	
19 Mvs,50	' 往现在位置上空 50mm 移动
20 M1=M1+1	' 计数器加算
21 If M1 <=12 Then *LOOP	' 计数在范围内的话，从 *LOOP 开始循环
22 Mov PICK,50	
23 End	

【说明】

- (1) 运算以 Def Plt 文定义的码垛的格子点位置。
- (2) 码垛号码最多可以同时 1 ~ 8, 8 个同时定义。
- (3) 请注意格子点的位置会依据码垛定义的指定方向而有所不同。
- (4) 指定超过以码垛定义文所定义的最大格子点号码的话，执行时会发生报警。
- (5) 将码垛的格子点使用在移动指令的目的位置的情况下，如下例没有用括号括起来的话，会发生报警。
Mov (Plt 1, 5)

详细内容请参照第 109 页的「4.1.2Pallet 运算」。

【相关指令】

Def Plt (Define 码垛)

Prec (Precision)

【功能】

使用于提升动作轨迹的情况。切换高精度模式的有效 / 无效。

【格式】

```
Prec  <On/Off>
```

【用语】

<On/Off>

On: 高精度模式有效时

Off: 高精度模式无效时

【例子】

```
1 Prec On          ' 使高精度模式为有效
2 Mvs P1
3 Mvs P2
4 Prec Off        ' 使高精度模式为无效
5 Mov P1
```

【说明】

- (1) 想将轨迹精度调高做插补动作时，以 **Prec On** 指令使高精度模式为有效。
- (2) 对应以 **Load Set** 指令指定的抓手、工件条件，调整为最佳的特性。抓手、工件条件没有正确设定的情况下，有可能无法显示出足够的性能。（请参照第 456 页的「[5.16 关于抓手、工件条件设定（最佳加减速设定）](#)」）。
- (3) **Prec** 指令会变成和 **MvTune** 指令相同的处理。**Prec Off** 对应 **MvTune1**(标准模式)、**Prec On** 对应 **MvTune3**(轨迹优先模式)。关于详细内容请参照 **MvTune** 指令的页次。

【相关指令】

[LoadSet \(Load Set\)](#)、[MvTune](#)

【关连参数】

HNDDAT0 ~ 8、WRKDAT0 ~ 8

Print

【功能】

在文件输出数据。数据全部为 ASCII 文字。

【格式】

```
Print □ #< 文件号码 > □ [, [< 式 >[: ]][...< 式 >[: ]]
```

【用语】

<文件号码> 记述以Open指令分配的1~8的号码。
<式> 可用数值表达式、位置表达式、字符串表达式记述。

【例子】 输出到 "temp.txt"。

```
1 Open "temp.txt" For APPEND As #1 ' 将 "temp.txt" 视为文件号码 1 开启。
2 MDATA=150 ' 在数值变量 MDATA 代入 150。
3 Print #1,"***Print TEST***" ' 输出字符串 "***Print TEST***"。
4 Print #1 ' 输出换行。
5 Print #1,"MDATA=",MDATA ' 输出字符串 "MDATA="、之后输出 MDATA 的值 (150)。
6 Print #1 ' 输出换行。
7 Print #1,"*****" ' 输出字符串 "*****"
8 End ' 程序结束。
```

输出结果如下所示。

```
***Print TEST***

MDATA=150

*****
```

【说明】

- (1) 没有记述 〈式〉 的情况下，只会输出断行字符。
- (2) 数据的输出形式 (例参考)
 - 〈式〉 的值和字符串输出领域会变成 14 个文字单位。输出复数值的情况下，使用逗号 (,) 做 〈式〉 和 〈式〉 的区别记号。输出时空出空间。在各领域单位的前面，若使用分号 (;) 的话，会在前面表示的值之后输出。此外，请务必在输出数据的最后加上断行字符。
- (3) Open 文没有被执行的情况下会发生报警。
- (4) 在含有双引号 (") 的时候，输出只到双引号为止。

例)

```
「1 M1=123.5
2 P1=(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0)」的时候
① 「3 Print #1,"OUTPUT TEST", M1, P1」记述的时候
会输出 OUTPUT TEST 123.5 (130.5,-117.2,55.1,16.2,0.0,0.0)(1,0)。
```

```
② 「3 Print #1,"OUTPUT TEST"; M1 ; P1」记述的时候
会输出 OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0)。
```

在最后的 〈式〉 的后面加上逗号或分号的话，不会执行换行的输出。

```
③ 「3 Print #1,"OUTPUT TEST",
4 Print #1,M1;
5 Print #1,P1」记述的时候
会输出 OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0) 。
```

【相关指令】

Open、Close、Input

Priority

【功能】

在多任务的程序运行，将复数的程序的行以顺序号码（初始值为一行一行）执行、动作。本指令为在多任务里指定程序执行时的优先度（优先执行行数）。

【格式】

Priority □ < 执行行数 > [, < 插槽号码 >]

【用语】

< 执行行数 > 将执行一次的行数以常数或变量指定。记述1~31的数值。
< 插槽号码 > 1~32，省略的时候会变成现在的插槽号码。以常数或变量指定。

【例子】

插槽 1
10 Priority 3 ' 现在的插槽的执行行数指定为 3。

插槽 2
10 Priority 4 ' 此插槽的执行行数指定为 4。

【说明】

- (1) 到指定的执行行数执行以前，其它的插槽的程序不会执行。例如像上述的例子在插槽 1 的程序里，指定 Priority3、在插槽 2 的程序里指定 Priority4 的情况，首先，执行插槽 1 的程序 3 行，之后执行插槽 2 的程序 4 行。之后会循环此操作。
- (2) 全部插槽的初始值为 1。因此，1 行执行后会往下一个插槽移动。
- (3) 在指定的任务插槽内没有程序的话，会发生报警。
- (4) 已指定的任务插槽的程序即使在运行中也可以变更。

RelM (Release Mechanism)

【功能】

在多任务运行跨越多插槽控制的情况下使用。在以 GetM 取得的机器开放时使用。

【格式】

```
Relm
```

【例子】

① 从任务插槽 1 开始起动任务插槽 2，且，在任务插槽 2 控制机器 1

任务插槽 1

```
1 RelM           ' 为了用插槽 2 控制机器 1， 开放机器。
2 XRun 2,"10"    ' 在插槽 2 选择程序 10
3 Wait M_Run(2)=1 ' 等待插槽 2 的起动确认
:
```

任务插槽 2(程序 "10")

```
1 GetM 1         ' 取得机器 1 的来源
2 Servo On      ' 开启机器 1 的伺服
3 Mov P1
4 Mvs P2
5 Servo Off     ' 关闭机器 1 的伺服
6 RelM          ' 开放机器 1 的来源
7 End
```

【说明】

- (1) 开放现在取得的机器来源。
- (2) 在取得机器的状态下，将执行的程序中中断使停止的情况下，会自动的开放取得的机器来源。
- (3) 通常执行程序内无法使用。

【相关指令】

[GetM \(Get Mechanism\)](#)

Rem(Remarks)

【功能】

使其以后的字符串变成指令。

【格式】

```
Rem □ [<指令>]
```

【用语】

<指令> 记述任意的字符串。
可以在一行的范围内记述。

【例子】

```
1 Rem ***MAIN PROGRAM***
2 ' ***MAIN PROGRAM***
3 Mov P1                ' 在 P1 移动
```

【说明】

- (1) Rem 可以用单引号省略记载。
- (2) 可以像例子的步号 3 一样在其指令后面记述。

Reset Err(Reset Error)

【功能】

将在机器人控制器上所发生的报警复位。在初始状态会变成使用禁止。
在警告报警以外发生的情况下，通常的程序无法运行但是通常执行程序可以运行。
此指令在常执行程序内使用的时候为有用。

【格式】

Reset Err

【例子】

在通常执行程序的执行例

1 If M_Err=1 Then Reset Err' 在控制器发生报警的情况下，将报警复位。

【说明】

- (1) 藉由参数「SLT*」，起动条件设定在通常执行 (ALWAYS) 想要在程序内复位机器人的系统报警的情况下使用。
- (2) 在参数「ALWENA」值从 0 变更为 1，控制器的电源再起动以后变成有效。

【关连参数】

ALWENA

【关连系统状态参数】

M_Err(1：报警发生中、 0：无报警)

Return

【功能】

- ①从通常的子程序 Return 的情况下，返回到 GoSub 的下一个单步控制。
- ②从插入处理用的子程序的 Return 的情况下，返回到发生插入的单步或下一个单步的控制。

【格式】

- ①从平常的子程序的 Return 的情况。

```
Return
```

- ②从插入处理用子程序 Return 的情况。

```
Return < 返回处指定号码 >
```

【用语】

- <返回处指定号码>插入发生时，在插入处理执行后，返回单步控制以常数或变量指定。
- 0...将控制返回到插入发生的单步。
 - 1...将控制返回到插入发生单步的下一个单步。

【例子】

① 从平常的子程序的 Return 例

```
1 '***MAIN PROGRAM***
2 GoSub *SUB_INIT      '子程序跳转到 level SUB_INIT。
3 Mov P1
4 :
100 '***SUB INIT***    '子程序
101 *SUB_INIT
102 PSTART=P1
103 M100=123
104 Return             '返回到从子过程调用的单步的下一个单步。
```

② 从插入处理用的子程序 Return 的例子。

通用输入信号 17 号的输入信号为开启状态的话，呼出 level Lact 的子程序

```
1 Def Act 1,M_In(17)=1 GoSub *Lact      ' Act 1 的插入定义执行。
2 Act 1=1                                ' 使 Act 1 为有效。
3 :
10 *Lact                                  ' Act 1 插入用子程序
11 Act 1=0                                ' 禁止插入。
12 M_Timer(1)=0                          ' 定时器设定为 0
13 Mov P2                                  ' 往 P2 移动
14 Wait M_In(17)=0                       ' 等待到输入信号 17 关闭为止。
15 Act 1=1                                ' 再设定插入。
16 Return 0                              ' 发生插入返回到单步。
```

【说明】

- (1) 以 GoSub 在被呼出的跳转处理的最后记述 Return 指令。
- (2) GoSub 没有在被呼出的跳转处理的最后执行 Return 的情况下会发生报警。
- (3) 以 GoSub 呼出的情况下请务必以 Return 返回。以 GoTo 指令返回的话控制构造用内存 (堆栈内存) 会不足而发生报警。
- (4) 在平常的子程序的 Return 有指定返回处号码的情况及从插入处理用的子程序的 Return 没有指定返回号码的情况，执行时会发生报警。

- (5) 在从插入处理的 **Return1** 返回到下一个单步的情况，请禁止在插入处理内插入。没有禁止插入使插入直接成立的话，会再度执行插入处理为了返回到下一次的单步，单步行不会被执行而是被跳过 (**Skip**)。关于插入处理请参照第 190 页的「**Def Act**」。
- (6) 圆、或圆弧插补 (**Mvc,Mvr,Mvr2,Mvr3**) 的执行中有插入进入、以 **Return0** 返回到原本的单步控制的时候，机器人会回到圆或圆弧的起点开点，再度执行圆、圆弧插补。
- (7) 在圆弧插补的执行中、以 **Return0** 返回到原本的单步控制的时候，机器人会从那时的位置开始进行圆弧插补。

【相关指令】

Act、**GoSub(~ Return) (Go Subrouine)**、**On... GoSub (ON Go Subroutine)**、
On Com GoSub (ON Communication Go Subroutine)、**Def Act**

Select Case

【功能】

遵照条件式的值，执行复数的叙述区块 (Statement Block) 的其中一个。

【格式】

```
Select □ < 条件 >
    Case □ < 式 >
        [< 处理 >]
        Break
    Case □ < 式 >
        [< 处理 >]
        Break
    :
    Default
        [< 处理 >]
        Break
End □ Select
```

【用语】

<条件> 记述数值表达式。
 <式> 式以下列的形式记述。型则必须和〈条件〉的型相同。
 • Is <比较运算> <常数>
 • <常数>
 • <常数> To <常数>
 <处理> 用以MELFA-BASIC V准备的指令记述（GoTo指令除外）。

【例子】

```
1 Select MCNT
2 M1=10 ' 此行不会被执行。
3 Case Is <= 10 ' MCNT <=10
4 Mov P1
5 Break
6 Case 11 ' MCNT=11 OR MCNT=12
7 Case 12
8 Mov P2
9 Break
10 Case 13 To 18 '13 <= MCNT <=18
11 Mov P4
12 Break
13 Default ' 上記以外
14 M_Out(10)=1
15 Break
16 End Select
```

【说明】

- (1) 条件和 Case 的其中一个一致的话，到 Break 或 Case 或 Default 或 End Select 为止的处理会被执行。
条件和 Case 的其中一个不一致的话，会执行被记述为 Default 的区域 (Block)。
- (2) 在没有 Default 的情况下，以无处理跳转到 End Select 的下一行。
- (3) Select Case 和 End Select 文必须要相互对应。在 Case 区域内以 Go To 指令跳转到 Select Case 以外的话，会因为控制构造用内存（堆栈内存）减少，而在以连续行的情况下，不知道在何时会发生报警。
- (4) 执行没有对应 Select Case 的 End Select 文的情况下，执行时会发生报警。
- (5) 可以在 Select Case 中，更加记述 Select Case(回路可以到 8 段)。
- (6) 在 Case 文中可以记述 While ~ Wend 和 For ~ Next。
- (7) 在〈式〉中使用比较运算(<、=、>等)的时候，使用 Case Is。
- (8) Break 可以省略。(Case 文里处理会遵照为 (1))

Servo (Servo)

【功能】

控制伺服电源的开启关闭。

【格式】

① 通常的程序

```
Servo □ <On/Off>
```

② 通常执行 (ALWAYS) 程序

```
Servo □ <On/Off> , < 机器号码 >
```

<On/Off>

On: 开启伺服电机的电源

Off: 关闭伺服电机的电源

<机器号码>

只在通常执行程序内有效。

以1~3、常数或变量记述。

【例子】

```
1 Servo On           ' 伺服开启
2 *L20:If M_Svo<>1 GoTo *L20   ' 等待伺服开启
3 Spd M_NSpd
4 Mov P1
5 Servo Off
```

【说明】

(1) 将机器人全体作为全轴对象，执行伺服电源的控制。

(2) 有附加轴的情况下，附加轴的伺服电源也会成为对象。

(3) 在通常执行程序内使用的情况下，将参数「ALWENA」的值从0变更为1，然后再次开启电源才会变的有效。

【关连系统状态变量】

M_Svo(1: ON、0: OFF)

【关连参数】

ALWENA

SetCalFrm (Set Calibration Frame)

【功能】

设定帧转换中使用的基准坐标系。

通过本指令设定 2 种（帧转换前与帧转换后）的基准坐标系。

注）该指令可在软件版本为 Ver.R5 以上 (F-Q 系列) / Ver.S5 以上 (F-D 系列) 的版本中使用。

【格式】

SetCalFrm □ <位置 1>, <位置 2>, <位置 3>, <位置 4>, <位置 5>, <位置 6>

【用语】

<位置1>	以位置变量或位置常数指定帧转换前基准坐标系中的X·Y·Z轴的原点位置数据。
<位置2>	以位置变量或位置常数指定帧转换前基准坐标系中的+X轴上的点位置数据。
<位置3>	以位置变量或位置常数指定帧转换前基准坐标系的X-Y平面上+Y方向的点位置数据。
<位置4>	以位置变量或位置常数指定帧转换后基准坐标系中的X·Y·Z轴的原点位置数据。
<位置5>	以位置变量或位置常数指定帧转换后基准坐标系中的+X轴上的点位置数据。
<位置6>	以位置变量或位置常数指定帧转换后基准坐标系的X-Y平面上+Y方向的点位置数据。

【例文】

1 PR1=(0, 0, 0, 0, 0, 0)(0, 0)	' 帧转换前的基准坐标系的原点位置
2 PR2=(1, 0, 0, 0, 0, 0)(0, 0)	' 帧转换前的基准坐标系的 +X 轴位置
3 PR3=(0, 1, 0, 0, 0, 0)(0, 0)	' 帧转换前的基准坐标系的 +Y 方向位置
4 PC1=(0, 0, 0, 0, 0, 0)(0, 0)	' 帧转换后的基准坐标系的原点位置
5 PC2=(1, 1, 0, 0, 0, 0)(0, 0)	' 帧转换后的基准坐标系的 +X 轴位置
6 PC3=(-1, 1, 0, 0, 0, 0)(0, 0)	' 帧转换后的基准坐标系的 +Y 方向位置
7 SetCalFrm PR1, PR2, PR3, PC1, PC2, PC3	' 设定帧转换中使用的基准坐标系
	' 帧转换后的基准坐标系为与转换前相比
	' 在 Z 轴周围旋转 +45 度的坐标系
8 MvSpl 5, 50, 20, 2	' 使用通过 SetCalFrm 指令设定的基准坐标系, 对样条
	' 文件 5 的路径点进行帧转换, 执行通过该路径点的
	' 样条插补

【说明】

(1) 设定对帧转换中使用的 2 种基准坐标系（帧转换前与帧转换后）进行定义的位置数据。为了定义一个坐标系，需要 3 个位置数据，因此要设定共计 6 点的位置数据。

在下图的例中，通过位置数据 PR1、PR2、PR3 定义帧转换前的基准坐标系“Xfr-Zfr-Yfr”、通过位置数据 PC1、PC2、PC3 定义帧转换后的基准坐标系“Xfc-Zfc-Yfc”。

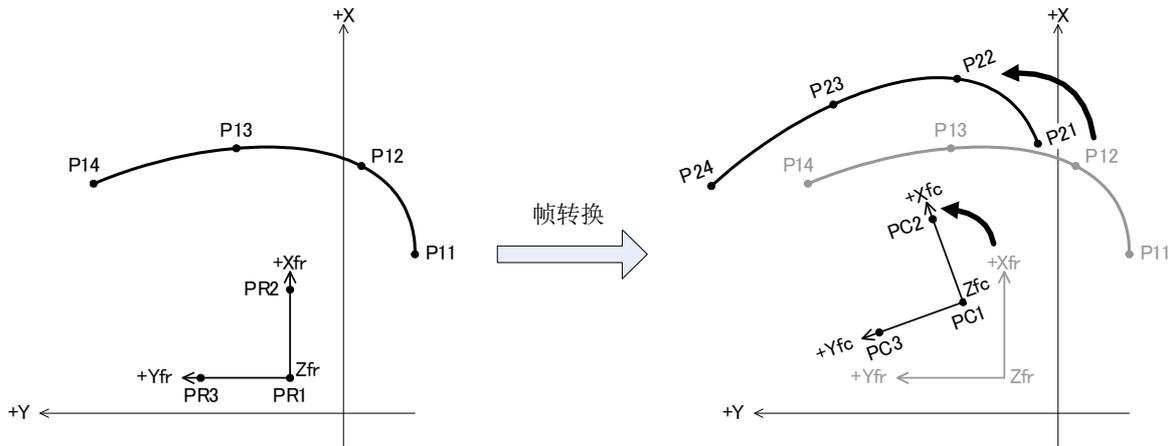


图 4-30: 帧转换的示例

- (2) 在坐标系的定义中使用各位置数据的 X、Y、Z 轴坐标值。A、B、C 轴坐标值等不使用其他的成分数据。
- (3) 定义坐标系的 3 个位置数据中含有同一点时，或 3 个位置数据并排成直线状时，将无法算出坐标系并发生 L2041（无法算出帧转换坐标）错误。
- (4) 接通控制器的电源之后，基准坐标系处于未设定的状态。此外，通过主程序的 End 指令・程序复位操作，基准坐标系将返回未设定的状态。
- (5) 执行 SetCalFrm 指令需要机械的控制权（GetM 指令）。
- (6) 启动条件通过 ALWAYS・ERROR 的插槽无法执行 SetCalFrm 指令。发生 L3287（启动条件为 ERR、ALW 时无法使用该指令）错误。

【相关指令】

MvSpl (Move Spline)

【関連関数】

Fram

Skip(Skip)

【功能】

将程序的控制移到下一行。

【格式】

Skip

【例子】

- 1 Mov P1 WthIf M_In(17)=1,Skip ’ 在位置变量 P1 以关节插补往显示位置移动中，输入信号 M_IN(17) 变成开启的时候，机器人的插补动作中断且此指令的执行会中断并执行下一个单步。
- 2 If M_SkipCq=1 Then Hlt ’ 执行 Skip 的话，程序会中断。

【说明】

(1) 使用在 Wth 节或 WTHiF 节。此情况下，会自动中断其单步的执行自动的往下一个控制移动。是否已 Skip 可以藉由 M_SkipCq 得知。

【关连系统状态变量】

M_SkipCq (1: 已 Skip、0: 没有 Skip)

Spd (Speed)

【功能】

指定机器人的直线移动、圆弧移动时的速度。另外指定最佳速度控制模式。

【格式】

Spd □ < 指定速度 >
Spd □ M_NSpd(最佳速度控制模式)

【用语】

< 指定速度 > 速度以实数指定。单位: [mm/s]

【例子】

```
1 Spd 100
2 Mvs P1
3 Spd M_NSpd            ' 设定初始值 ( 最佳速度控制模式 )
4 Mov P2
5 Mov P3
6 Ovrđ 80                ' 最佳速度模式中的速度过速报警对策
7 Mov P4
8 Ovrđ 100
```

【说明】

- (1) Spd 指令只有在直插补、圆弧插补时有效。
- (2) 实际的速度比例 = (操作面板 (T/B) 的速度比例设定值) × (程序速度比例 (Ovrđ 指令)) × (直线指定速度 (Spd 指令))。
- (3) Spd 指令只会使直线、圆弧指定速度变化。
- (4) 指定速度以 M_NSpd(初始值: 值为 10000。) 指定的情况下, 机器人会经常以最高速度动作, 因此线速无法保持一定 (最佳速度控制)。
- (5) 即使在最佳速度, 也会依据机器人的姿势发生报警。如果过速度的报警发生的话, 在其动作指令前插入 Ovrđ 指令且只降低那个区间的速度使用。
- (6) 程序中, 到实行 Spd 指令为止的指定速度会采用系统的初始值。执行一次 Spd 指令的话, 到下一个 Spd 指令执行为止, 会采用其指定的速度。
- (7) 执行程序 End 文的话, 指定速度会被设定在系统初始值。

【关连系统状态变量】

M_NSpd(系统的初始值 最佳速度控制模式 10000)
M_Spd(现在的指定速度)
M_RSpd(现在的实际速度)

SpdOpt (Speed optimize)

【功能】

以通过原点 (X=Y=0、机器人的特异点之一) 附近的水平方向的直线插补动作来调整速度，使得不出现超速。

注) 本指令可用于以下的机器人中。

机型：RH-3FHR 系列

【格式】

SpdOpt □ <On/Off>

【用语】

<On/Off> On : 将速度调整功能设为有效
 Off: 将速度调整功能设为无效

【例文】

```
1 Mov P1
2 SpdOpt On           ' 将速度调整功能设为有效
3 Mvs P2
4 Mvs P3
5 SpdOpt Off         ' 将速度调整功能设为无效
6 Mvs P6
```

【说明】

(1) 保持控制点的速度以直线插补进行动作的情况下，如图 4-31 所示想要通过原点 (机器人的特异点之一) 附近时，则 J1 轴必须进行高速旋转，根据指定的速度，有时会发生超速错误，但执行 SpdOpt On 后，即自动调整速度使得不会发生错误。

例如，在指令速度 V 下的动作过程中，靠近原点，一直不会超速的情况下，如图 4-32 的 A 所示，自动降低速度以防止发生超速。之后，通过原点附近，到可提高速度后，如图 4-32 的 B 所示向指令速度 V 开始加速。

(2) 与 Ovrđ 或 Spđ 指令的关系

以与本功能中的调整速度相比更慢的方式动作。

- 指定速度 (Ovrđ 或 Spđ 指令) 更慢时 以指定速度进行动作
- 指定速度 (Ovrđ 或 Spđ 指令) 更快时 以本功能中的调整速度进行动作

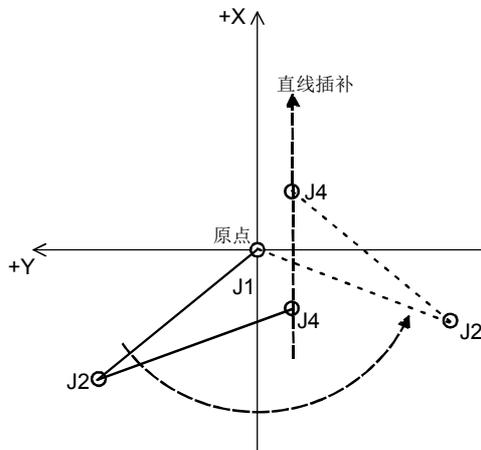


图 4-31: 原点附近的直线插补通过 (示例)

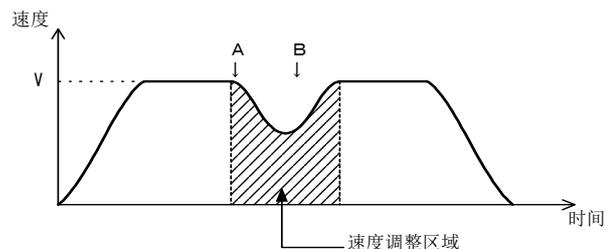


图 4-32: 速度调整时的速度样子

- (3) 本指令起作用的仅有直线插补动作。在关节插补及圆弧插补中不起作用。
此外，即使速度调整功能变为有效，J4 轴不通过 [图 4-33 速度调整区域与特异点区域] 所示区域的直线插补的情况下，也不起作用。

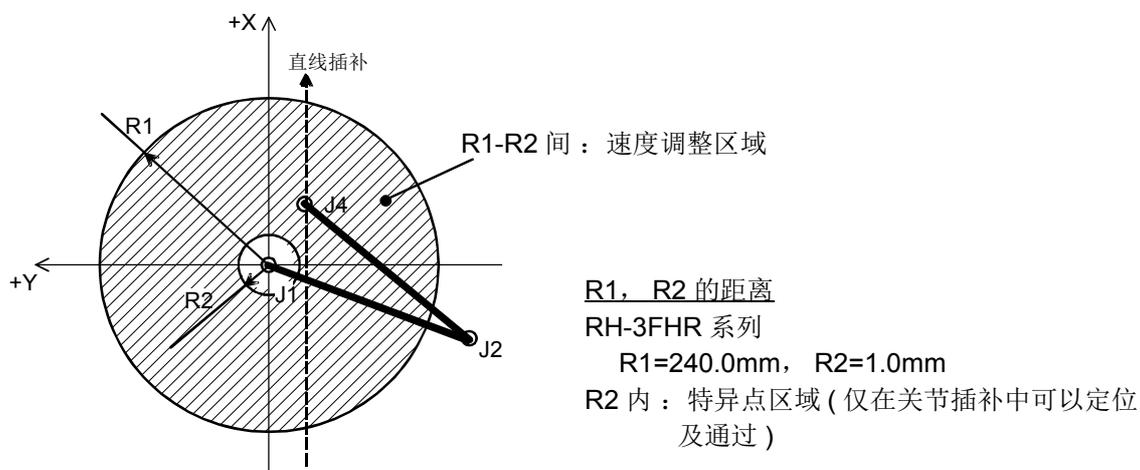


图 4-33: 速度调整区域与特异点区域

- (4) 电源接通后的速度调整功能的初始状态可以通过参数 SPDOPT 进行更改。该参数也限定了可以使用的机型。
对象机型中的初始值为 SPDOPT=1(速度调整有效)。
- (5) 执行 End 指令、程序复位操作后，速度调整功能的状态即返回电源接通后的初始状态。
- (6) 速度调整功能有效时，如欲执行 J4 轴通过图 4-33 所示的特异点区域之类的直线插补，则会发生 2804 错误并中断运行。
- (7) 在非对象机型中，即使在程序中记述了本指令，也将被忽略。
- (8) 在 origin 附近将 Cnt 指令设为有效以进行轨迹连接或执行大的姿势变化之类的直线插补动作时，即使速度调整功能为有效，也可能会发生超速错误。该情况下，应使进行轨迹连接的位置远离原点，或通过 OvrD 指令等调整速度。
- (9) 水平方向仅略微动作而垂直方向会发生大动作之类的直线插补的情况下，速度调整功能一变为有效，与无效时相比动作速度有可能会大幅降低。该情况下，应将速度调整功能设为无效，或通过关节插补 (Mov 指令) 进行动作。

【可使用的机器人机型】

RH-3FHR 系列

【相关参数】

SPDOPT

SplFWrt (Spline Frame Write)

【功能】

在指定的样条文件中登录帧转换信息。

注) 该指令可在软件版本为 Ver.R6b 以上 (F-Q 系列) / Ver.S6b 以上 (F-D 系列) 的版本中使用。

【格式】

SplFWrt □ < 样条号码 >, < 位置 1 >, < 位置 2 >, < 位置 3 >, < 位置 4 >, < 位置 5 >, < 位置 6 >

【用语】

- | | |
|----------|----------------------------------|
| < 样条号码 > | 指定登录帧转换信息的样条文件的号码。
设定范围: 1~99 |
| < 位置 1 > | 指定帧转换中使用的基准坐标系的原点。 |
| < 位置 2 > | 指定帧转换中使用的基准坐标系X轴上的位置。 |
| < 位置 3 > | 指定帧转换中使用的基准坐标系的XY平面上+Y方向的位置。 |
| < 位置 4 > | 指定帧转换中使用的补偿后的原点。 |
| < 位置 5 > | 指定帧转换中使用的补偿后的X轴上的位置。 |
| < 位置 6 > | 指定帧转换中使用的补偿后的XY平面上+Y方向的位置。 |

【例文】

SplFWrt 1, PR1, PR2, PR3, PC1, PC2, PC3 ' 在样条文件 1 中设定帧转换中使用的基准坐标系 (PR1, PR2, PR3) 与补偿后的坐标系 (PC1, PC2, PC3)。

【说明】

- (1) 在 < 样条号码 > 指定的样条文件中登录帧转换信息。
- (2) 设定对帧转换中使用的 2 种基准坐标系 (帧转换前与帧转换后) 进行定义的位置数据。定义 1 个坐标系, 因此需要 3 个位置数据。
- (3) 在坐标系的定义中使用各位置数据的 X、Y、Z 轴坐标值。A、B、C 轴坐标值等不使用其他的成分数据。
- (4) 定义坐标系的 3 个位置数据中含有同一点时, 或 3 个位置数据并排成直线状时, 将无法算出坐标系并发生 L2041 (无法算出帧转换坐标) 错误。
- (5) < 样条号码 > 对应的样条文件未保存在控制器中的情况下, 会发生 L2610(无法打开样条文件) 错误。

SplWrt (Spline Write)

【功能】

创建含有指定文件信息的样条文件。

注) 该指令可在软件版本为 Ver.R6b 以上 (F-Q 系列) / Ver.S6b 以上 (F-D 系列) 的版本中使用。

【格式】

```
SplWrt □ <样条号码>, <文件名 1> [, <动作模式>[, <样条取消角度>
      [, <块长比率>[, <Ex-T 坐标号码>[, <文件版本>]]]]]
```

【用语】

- <样条号码> 指定要创建的样条文件的号码。
设定范围：1～99
- <文件名> 指定包含有要登录在样条文件中的路径点信息的文件名。
应在任意16字符内进行记述。
- <动作模式> 指定样条插补下的动作模式(线速固定/线速可变)。
设定范围：
0：线速固定模式
1：线速可变模式
省略时：在线速固定模式下创建样条文件。
- <帧转换> 以常数或数值变量指定帧转换的执行内容。
设定范围：
0：不执行帧转换。
1：使用样条文件中设定的坐标系执行帧转换。
2：使用SetCalFrm指令中设定的坐标系执行帧转换。
省略时：不执行帧转换。
- <样条取消角度> 指定取消样条插补的角度。
设定范围：0～180(度)
省略时：120(度)
- <块长比率> 指定使样条曲线按直线动作的块长比率。
设定范围：0～100(倍)
省略时：8倍
- <Ex-T坐标号码> 指定用作控制点的工件坐标的号码。
设定范围：0～8
省略时：将Ex-T样条功能看作无效。
- <文件版本> 指定要创建的样条文件的版本。
设定范围：1～2
省略时：文件版本设为2以创建样条文件。(看作控制器对应的样条文件的最新版本)

【例文】

SplWrt 1,"01.csv" '基于"01.csv"中登录的路径点信息生成样条文件 1。

SplWrt 2,"02.csv", 0, 120, 8, 1, 2" 基于"01.csv"中登录的路径点信息生成样条文件 1。

【说明】

- (1) 创建 <样条号码> 指定的样条文件。
- (2) 将已登录至 <文件名> 指定的文件中的路径点信息登录至样条文件。

- (3) 将 < 动作模式 > 指定的模式登录至样条文件。省略时作为 0(线速固定) 被登录至样条文件。
- (4) 将 < 样条取消角度 > 指定的角度登录至样条文件。省略时作为 120(度) 被登录至样条文件。指定为 0 时, 样条取消不起作用。
- (5) 将 < 块长比率 > 指定的比率登录至样条文件。省略时作为 8(倍) 被登录至样条文件。指定为 0 时, 块长比率不起作用。
- (6) <Ex-T 坐标号码 > 指定为 1 ~ 8 时, 将 WK1CORD ~ WK8CORD 中设定的坐标系数数据登录至样条文件。指定为 0 时, 将 Ex-T 样条功能看作无效。省略时看作 0(Ex-T 样条功能无效)。
- (7) 创建 < 文件版本 > 指定的版本的样条文件。省略时登录为 2(控制器对应的样条文件的最新版本)。
- (8) 本指令执行后, < 文件名 > 指定的文件将从控制器中被删除。
- (9) 通过别的插槽的样条相关指令 (MvSpl、EMvSpl、SplPos、SplSpd、SplECord) 在打开对象的样条文件的状态下执行本指令时, 会发生 L2610(无法创建样条文件) 错误。
- (10) 无法打开 < 文件名 > 指定的路径点文件时, 会发生 L2611(无法打开路径点文件) 错误。
- (11) < 文件名 > 中登录的路径点信息与格式不相同 (标记数不相同、标记名不相同) 时, 会发生 L2611(文件的格式不相同) 错误。
- (12) < 文件名 > 中登录的路径点的数量出现异常 (不足 4 点 /5001 点以上) 时, 会发生 L2611(路径点的个数不当) 错误。
- (13) < 文件名 > 中登录的 M_SplVar 的值超出范围时, 会发生 L2615(M_SplVar 的设定值超出范围) 错误。
- (14) < 文件名 > 中登录的允许公差的值超出范围时, 会发生 L2615(允许公差的设定值超出范围) 错误。
- (15) < 文件名 > 中登录的输出信号的值超出范围时, 会发生 L2615(输出信号的设定值超出范围) 错误。
- (16) 自变量的值超出范围时, 将会发生 L3110(自变量的值超出范围 (SplWrt)) 错误。
- (17) 自变量的个数出现异常 (多 / 少) 时, 会发生 L4220(所输入的指令语句的结构有错误) 错误。
- (18) 控制器中没有创建样条文件的容量时, 会发生 C7070(存储容量不足) 错误。

【相关系统状态变量】

P_WkCord

【相关参数】

WK1CORD ~ WK8CORD

Title (Title)

【功能】

在程序加上 title。可以在选配的计算机支持软件里，在机器人控制器的一览表示栏里显示想要表示的文字。

【格式】

Title □ < 文字 >

【用语】

<文字>成为title的讯息。

【例子】

1 Title " 机器人 Loader Program"
2 Mvs P1
3 Mvs P2

【说明】

(1) 可以在程序的一行登录可以登录的最多文字数，但是在计算机支持软件的机器人控制器的程序一览显示栏里会变成最多可登录 20 个文字。

Tool (Tool)

【功能】

指定TOOL变换数据。设定TOOL(抓手)的长度、从机械I/F的控制点的位置、姿势。

【格式】

Tool □ <Tool 数据 >

【用语】

<Tool 数据> 以位置表达式（位置常数、位置变量等）指定Tool数据。

【例子】

① 设定直接数值

1 Tool (100,0,100,0,0,0) ' 以 Tool 坐标在 X 轴 100mm、 Z 轴 100mm 变更控制点
 2 Mvs P1
 3 Tool P_NTool ' 将控制点返回到初始值。（机械 I/F 位置、法兰面）

② 设定在直交的位置变量

（在 PTL01 设定为（100, 0, 100, 0, 0, 0, 0, 0）的话，会变成和①相同意思。）
 1 Tool PTL01
 2 Mvs P1

【说明】

- (1) Tool 指令是用于在使用 Double 抓手的系统里，想要在各抓手的尖端设定控制点的时候。抓手为 1 种类的情况下，并非使用 Tool 指令，请使用参数「MEXTL」设定。
- (2) 依据 Tool 指令变更的 Tool 变更数据会被储存在参数 MEXTL 里，控制器的电源关闭后即被储存。
- (3) 到 Tool 指令执行为止，系统初始值（P_NTool）会被采用。
 执行一次 Tool 指令的话，到下个 Tool 指令被执行为止，会采用已指定的 Tool 变换数据。与厂商构造无关，会以 6 轴三次元运算。
- (4) 示教时和自动运行时的 Tool 数据不同的话，会有在预料外动作的情况发生。
 请务必使运行时和示教时的设定一致化。
 此外，依据机器人的机型，有效轴成份会有所不同。
 请参照第 433 页的「表 5-8：对应机器人的形式 TOOL 变换数据有效的轴成份」，做适当的数据设定。
- (5) 可以使用 M_Tool 变量将 METL1 ~ 4 的参数设定为 Tool 数据。

【关连参数】

MEXTL、MEXTL1 ~ 4 详细内容请参照第 432 页的「5.6 关于标准 TOOL 坐标」。

【关连系统状态变量】

P_NTool(初始值)、P_Tool(现在的指定 Tool 变换数据)、M_Tool

Torq (Torque)

【功能】

指定各轴的转矩限制。依据转矩限制的指定，可以使工件等不要负载过大（过重）。
转矩限制值率超过的话，会发生误差过大的报警。

【格式】

Torq □ < 轴号码 >, < 转矩限制率 >

【用语】

<轴号码> 轴号码以数值常数指定。（从1到6）以常数或变量指定。
<转矩限制率> 从轴发生的限制以%指定。（从1到100）以常数或变量指定。

【例子】

```

1 Def Act 1,M_Fbd>10 GoTo *SUB1,S '指令位置和反馈位置的差在 10mm 以上插入
2 Act 1=1                          '插入有效
3 Torq 3,10                         '以转矩指令将 3 轴的转矩限制设定在通常的 10%
4 Mvs P1                             '动作
5 Mov P2
.
.
100 *SUB1
101 Mov P_Fbc                        '使指令位置和反馈位置一致
102 M_Out(10)=1                     '输出信号 10 号
103 End                              '在到达差的时点停止。

```

【说明】

- (1) 限制指定轴的转矩，不要在指定以上的转矩动作。值指定在对应标准时的转矩比例。标准的转矩会变成由制造商决定的值。
- (2) 依据机器人的机型，有效的转矩限制率会不同。会变成在伺服电机轴单位的设定，因此并不会限制在实际的机器人尖端的控制点的转矩限制率，请用各式的比率做测试。
- (3) 指定本指令，使转矩在限制状态停止的话，会有在指令位置和反馈位置偏离的状态停止的情况（依据摩擦）。此情况下，因为动作再开启时可能会发生误差过大的报警，因此请像上记例子中的步号 101 一样，在动作开始前使程序移动到反馈位置，就可以确实的动作。
- (4) 本指令只在标准的机器人轴有效。无法使用在通用伺服的轴（附加轴和使用者定义的机器）。想要做同样的动作请变更通用伺服侧的参数。

【关连系统状态参数】

M_Fbd、 P_Fbc

Wait (Wait)

【功能】

待机直到变量变成指定的值。

【格式】

```
Wait □ < 数值变量 > = < 数值常数 >
```

【用语】

<数值变量> 指定变量。经常使用输入输出信号变量(M_In、M_Out等)。
<数值常数> 指定数值常数。

【例子】

① 信号的状态

```
1 Wait M_In(1)=1          ' 和 1 *L10:If M_In(1)=0 Then GoTo *L10 意义相同。  
2 Wait M_In(3)=0
```

② 多插槽的状态

```
3 Wait M_Run(2)=1
```

③ 变量的状态

```
4 Wait M_01=100
```

【说明】

- (1) 等待信号输入和在多任务执行状态里使用连锁装置。
- (2) 成为指定的值时，往下一行移动。
- (3) 在多任务执行状态下，以 Wait 指令在复数的任务里同时执行时，其间的处理时间 (=Tact Time) 会变长，影响到系统。此情况下，请不要使用 wait 指令，更换为 If~Then 指令使用。
例) 50 Wait M_ABC=0 → 50 *LBL50:If M_ABC<>0 Then GoTo *LBL50
- (4) 在 Wait 指令中记述的条件个数为 1。如果记述了 2 个以上，则会出现错误判定或执行时发生错误。
错误记述示例)Wait M_in(38)=1 Or M_IN(39)=1
⇒ 这种情况下，请勿使用 Wait 指令，而应替换成 If ~ Then 指令。
例)*Loop
 If M_in(38)=1 Or M_IN(39)=1 Then *Next Else *Loop
 *Next

While ~ WEnd (While End)

【功能】

将 While 文和 WEnd 文之间的程序，满足循环 (loop) 条件时循环执行。

【格式】

```
While □ < 循环条件 >
:
WEnd
```

【用语】

<循环条件> 记述数值表达式。(参照第161页的「4.9运算符」)

【例子】

①数值变值 M1 的值在 -5 到 +5 的范围之间，循环处理，超越范围的情况下，移往 WEnd 的下一行控制。

```
1 While (M1>=-5) And (M1<=5)      '数值变值 M1 的值在 -5 到 +5 的范围之间，循环处理。
2   M1=M1+1                        ' M1 加上 1。
3   M_Out(8)=M1                    '输出 M1 的值。
4 WEnd                              '返回到 While 文 (单步 1)。
5 End                               '程序结束。
```

②从 While ~ WEnd 中通过 Break 可穿插至 WEnd 的下一行。

```
1 While (M1>=-5) And (M1<=5)      '1 While (M1>=-5) And (M1<=5) '数值变量 M1 的值在 -5 到 +5 之
   间重复处理。
2   M1=-(M1+1)                    ' M1 加上 1，反转符号。
3   M_Out(8)=M1                    '输出 M1 的值。
4   If M_In(8)=1 Then Break        '输入信号 8 开启则跳转到步号 6。
5 WEnd                              '返回 While 语句 (步 1)。
6 If M_BrkCq=1 Then Hlt
```

【说明】

- (1) 循环执行 While 文和 WEnd 文之间的程序。
- (2) <式> 的结果为真 (不为 0) 的期间，控制移到 While 文的下一行，循环处理。
- (3) <式> 的结果不为真 (为 0) 的情况下，控制移到 WEnd 文的下一行。
- (4) 从 While 文和 WEnd 之间以 GoTo 指令强制的跳转的话，控制构造用内存 (堆栈内存) 会减少，而在以连续行的情况下，不知道在何时会发生报警。请使程序的 While 文条件成立，脱离循环。
- (5) While ~ WEnd 中如果有 Break 语句，会跳转到 WEnd 的下一行，并从 While ~ WEnd 的处理中退出。

Wth (With)

【功能】

在插补动作附加处理。

【格式】

例) Mov P1 Wth □ <处理>

【用语】

<处理>

记述附加的处理。可以记述的指令如下所示。

①<数值型数据B> <代入运算符> <数值型数据A> [代入·信号修饰指令(参照第161页的「4.9运算符」)]

【例子】

① 在往 P1 的移动开始的同时，只将输出信号 17 号的数值变量 M1 所显示的值 +2 秒。

10 Mov P1 Wth M_Out(17)=1 Dly M1+2 '

【说明】

- (1) 只可以使用在移动指令的附随条件。
- (2) Wth 单独会发生报警。
- (3) 和动作的开始同时执行处理。
- (4) 和其它的插入的优先级关系如下所示。
Com > Act > Wthlf(Wth) > 代入脉冲

Wthlf (With lf)

【功能】

在插补动作指令有附加条件的附加处理。

【格式】

例) Mov P1 Wthlf □ <附加条件>, <处理>

【用语】

<附加条件>

记述处理附加条件。(和 Act 条件式相同)

<处理>

在附加条件成立的情况下, 附加处理。(和 Wth 相同)

作为处理可记述的指令如下所示。

① <数值型数据 B> <代入运算符> <数值型数据 A>

例) M_Out(1)=1、 P1=P2

② Hlt 文

③ Skip 文

【例子】

① 输入信号 17 开启的话, 会停止。

10 Mov P1 Wthlf M_In(17)=1, Hlt

② 现在的指令速度超过 200mm/sec 的情况下, 将输出信号 17 会 +2 秒的时候开启。

20 Mvs P2 Wthlf M_RSpd>200, M_Out(17)=1 Dly M1+2

③ 往 P3 直线插补移动中, 到达率变成 15% 的话, 将输出信号 1。

30 Mvs P3 Wthlf M_Ratio>15, M_Out(1)=1

【说明】

- (1) 只可以使用在移动指令的条件附加的附随条件。
- (2) 开始动作的开始, 同时条件监视。
- (3) 在处理的时候, 无法记述 Dly 指令。
- (4) 以 Hlt 指令和 Skip 指令使机器人停止动作时, 会与 Def Act 指令的“停止类型 1”或“停止类型 2”同样为减速停止。(参照第 190 页的「Def Act」)。Hit 为程序停止, Skip 为继续程序运行。
将参数 WTHFUNC 设定为 0 即可选择“停止类型 1”, 设定为 1 即可选择“停止类型 2”。初始设定为“停止类型 1”(WTHFUNC=0)。

XClr (X clear)

【功能】

解除从程序上指定任务插槽的程序选择状态。在指定任务插槽中，设为可执行新程序的状态。使用在多个任务运行时。

【格式】

XClr □ < 插槽号码 >

【用语】

<插槽号码> 指定1~32的插槽号码。以常数或变量指定。

【例子】

```

1 XRun 2,"1"      ' 在任务插槽 2 执行 1 号的程序

10 XStp 2         ' 中断任务插槽 2 的程序
11 Wait M_Wai(2)=1 ' 等待直到任务插槽 2 的程序中断为止
12 XRst 2        ' 解除插槽 2 的程序的 中断中状态 （在程序可选择状态）
13 Wait M_Psa(2)=1 ' 任务插槽 2 等待到变为可选择程序为止
14 XClr 2        ' 解除任务插槽 2 的程序选择状态
15 End

```

【说明】

- (1) 在指定的插槽内没有被选择的程序，会发生报警。
- (2) 已指定插槽在运行中的情况下会发生报警。
- (3) 已指定插槽在 中断的情况下会发生报警。
- (4) 在通常执行程序内使用的情况下，以参数「ALWENA」将值从 0 变更为 1，将控制器的电源关闭再开启，使它成为有效。

【相关指令】

XLoad (X Load)、XRst (X Reset)、XRun (x Run)、
XStp (X Stop)

【关连参数】

ALWENA

XLoad (X Load)

【功能】

从程序上在指定任务插槽加载指定程序。在多任务运行时使用。

【格式】

```
XLoad □ < 插槽号码 >, < 程序名 >
```

【用语】

<插槽号码> 指定1~32的插槽号码。以常数或变量指定。
<程序名> 指定程序名。以字符串常数指定（无法使用字符串变量）。

【例子】

```
1 If M_Psa(2)=0 Then *LblRun          ' 确认插槽 2 的程序可以选择状态
2 XLoad 2,"10"                        ' 在插槽 2 选择程序 10
3 *L30:If C_Prg(2)<>"10" Then GoTo *L30 ' 到载入前等待。
4 XRun 2                               ' 起动插槽 2
5 Wait M_Run(2)=1                     ' 等待插槽 2 的起动确认。
6 *LblRun
7 ' 插槽 2 在运行中的情况，从这里开始执行。
```

【说明】

- (1) 指定的程序不存在的情况下，会发生报警。
- (2) 指定的程序被其它的插槽选择的情况下，执行时会发生报警。
- (3) 指定的程序在编辑中的情况下，执行时会发生报警。
- (4) 指定的插槽在运行中的情况下，执行时会发生报警。
- (5) 程序名的指定以双引号（“程序名”）圈起来指定。
- (6) 在通常执行程序内，使用的情况下，以参数「ALWENA」将值从0变更为1，将控制器的电源关闭再开启，使它成为有效。
- (7) 在XLoad执行后，执行XRun的话，因为程序在加载中因此会发生报警，必要的情况下，请像例子的步号3一样，加载完成后执行确认。

【相关指令】

[XClr \(X clear\)](#)、[XRst \(X Reset\)](#)、[XRun \(x Run\)](#)、[XStp \(X Stop\)](#)

【关连参数】

ALWENA

XRst (X Reset)

【功能】

由程序指定的任务插槽的程序中断的情况下，返回到执行行的前面行。（程序复位）。
多任务运行时使用。

【格式】

XRst □ < 插槽号码 >

【用语】

<插槽号码> 指定1~32的插槽号码。以常数或变量指定。

【例子】

```

1 XRun 2           ' 起动
2 Wait M_Run(2)=1 ' 等待起动完成
  :
10 XStp 2          ' 停止
11 Wait M_Wai(2)=1 ' 等待停止完成
  :
15 XRst 2          ' 程序的执行开始行设为前头行。
16 Wait M_Psa(2)=1 ' 等待程序复位完成
  :
20 XRun 2          ' 再开启
21 Wait M_Run(2)=1 ' 等待再起动完成

```

【说明】

- (1) 只有在插槽变成中断状态下有效。
(插槽在运行中的状况或程序未选择时会变成报警。)
- (2) 在通常执行程序内，使用的情况下，以参数「ALWENA」将值从0变更为1，将控制器的电源关闭再开启，使它成为有效。

【相关指令】

XClr (X clear)、XLoad (X Load)、XRun (x Run)、
XStp (X Stop)

【关连参数】

ALWENA

【关连系统状态参数】

M_Psa(插槽号码)(1 : 程序可以选择 0 : 不可选择)
M_Run(插槽号码)(1 : 执行中 0 : 执行中以外)
M_Wai(插槽号码)(1 : 中断中 0 : 中断中以外)

XRun (x Run)

【功能】

由程序上指定的程序并列执行。使用在多任务运行时。

【格式】

```
XRun □ < 插槽号码 >[, ["< 程序名 >"], < 运行模式 >]]
```

【用语】

< 插槽号码 >	指定1~32的插槽号码。以常数或变量指定。
< 程序名 >	指定程序名。以字符串常数或字符串变量指定。 在插槽的程序里已有程序已有设定的情况下可以省略。
< 运行模式 >	0=连续运行、 1=循环停止运行、省略时会变成现在的运行模式。以常数或变量指定。

【例子】

① 以 Xrun 指令指定动作程序的情况 (连续运行)

```
1 XRun 2,"1"          ' 程序以插槽 1 起动
2 Wait M_Run(2)=1    ' 等待起动完成
```

② 以 Xrun 指令指定动作程序的情况 (循环运行)

```
1 XRun 3,"2",1       ' 将程序 2 在插槽 3 以循环运行模式起动
2 Wait M_Run(3)=1    ' 等待起动完成
```

③ 使用 Xload 指令，指定动作程序的情况 (连续运行)

```
1 XLoad 2, "1"        ' 将程序 1 在插槽 2 选择
2 *LBL: If C_Prg(2)<>"1" Then GoTo *LBL ' 等待到载入完成为止
3 XRun 2              ' 将插槽 2 起动
```

④ 使用 XLoad 指令，指定动作程序的情况 (连续运行)

```
1 XLoad 3, "2"        ' 将程序 2 在插槽 3 选择
2 *LBL: If C_Prg (3)<>"2" Then GoTo *LBL ' 等待到载入完成为止
3 XRun 3,,1          ' 将插槽 2 以循环运行模式起动
```

【说明】

- (1) 指定的程序不存在的情况下，会发生报警。
- (2) 指定的插槽号码已经被使用的情况下，执行时会发生报警。
- (3) 在任务插槽里程序没有被加载的情况下，因为会在本指令执行加载，所以即使没有执行 Xload 指令也可以运行。
- (4) 在程序的途中，已停止的「中断中」状态，执行 Xmn 的话，执行会继续。
- (5) 程序名的指定以双引号 ("程序名") 圈起来指定。
- (6) 运行模式省略时会以现状的运行模式执行。
- (7) 在通常执行程序内，使用的情况下，以参数「ALWENA」将值从 0 变更为 1，将控制器的电源关闭再开启，使它成为有效。
- (8) 在 Xload 执行后，执行 Xrun 的话，会有程序加载中的报警发生，因此必要的情况下，如例子③的步号 2、④的步号 2 一样，加载完成后执行确认。

【相关指令】

XClr (X clear)、XLoad (X Load)、XRst (X Reset)、
XStp (X Stop)

【关连参数】

ALWENA

【关连系统状态参数】

M_Run(插槽号码) (1: 执行中、0: 执行中以外)

XStp (X Stop)

【功能】

由程序上将所指定的任务插槽的程序运行中止。若已指定的任务插槽使机器人动作的情况下，机器人停止。在多任务运行时使用。

【格式】

XStp □ < 插槽号码 >

【用语】

<插槽号码> 指定1~32的插槽号码。以常数或变量指定。

【例子】

```

1 XRun 2          ' 执行
:
10 XStp 2         ' 停止
11 Wait M_Wai(2)=1 ' 等待停止完了
:
20 XRun 2         ' 再开启
  
```

【说明】

- (1) 在已经变成中断的情况下，报警不会发生。
- (2) XStp 可以使通常执行程序停止。
- (3) 在通常执行程序内，使用的情况下，以参数「ALWENA」将值从0变更为1，将控制器的电源关闭再开启，使它成为有效。

【相关指令】

XClr (X clear)、 XLoad (X Load)、 Xrst (X Reset)、
XRun (x Run)

【关连参数】

ALWENA

【关连系统状态参数】

M_Wai(插槽号码) (1: 中断中、 0: 中断中以外)

代入

【功能】

在变量或数组变量将运算的结果代入。

【格式】

```
< 变量名 >=< 式 1>
```

脉冲代入的情况

```
< 变量名 >=< 式 1> Dly < 式 2>
```

【用语】

<变量名> 记述想要代入值的变量名。
(变量的种类请参照第141页的「4.4.15变量」)

<式1> 代入值。记述数值表达式。

<式2> 脉冲计数器。记述数值表达式。

【例子】

① 变量运算结果的代入

```
10 P100=P1+P2*2
```

② 信号的输出

```
20 M_Out(10)=1 ' 将输出信号 10 开启
```

③ 信号的脉冲输出

```
30 M_Out(17)=1 Dly 2.0 ' 将输出信号 17 在 2 秒间开启
```

【说明】

- (1) 脉冲输出用和下一个行以后的指令执行并列被执行。
- (2) 脉冲输出、在指定时间经过后反转输出。以 M_Outb, M_Outw 输出脉冲的情况下，会以 8 位单位、16 位单位反转，请注意。无法以任意位数作为单位反转。
- (3) 脉冲输出在指定时间中，End 指令和程序的最终行被执行的情况下，不会等待指定时间经过，程序的执行结束。但是，在指定时间后输出会反转。

(Label)**【功能】**

表示分支处。

【格式】

```
*<Label 名 >
```

```
*<Label 名 >[ : <指令行 >]
```

【用语】

<Label 名> 记述从英文字开始的字符串，开头请务必以英文字。

最高请在16个文字以内记述(含*最高17个文字)。

<指令行> 可以在Label后的冒号(：)记述继续指令行。

【例子】

```
1 *SUB1
```

```
2 If M1=1 Then GoTo *SUB1
```

```
3 *LBL1:If M_In(19)=0 Then GoTo *LBL1 '直到输入信号 10 号开启为止，在步号 3 等待。
```

【说明】

(1) 在程序中即使没有参照也不会发生报警。

(2) 在一个程序内，同一个 Label 复数定义的话，会发生重复定义报警。

(3) 保留字无法使用在 Label。

(4) 在 Label 名使用底线("_")的情况下，(ex.*L_LABEL)、在第 1 个文字可以使用的只有"L"。

使用"L"以外的文字的情况下(ex.*A_LABEL)，会发生异常。

例) 使用底线时的正确 Label 例(开头文字为"L")

```
*L_ABC, *L12_345, *LABEL_1
```

底线使用时的错误 Label 例

```
*H_ABC,*ABC_123,*NG_,*_LABEL
```

(5) 在 Label 后面的逗号(：)可以记述继续指令行。但是，在指令行的后面记述冒号则无法记述再度指令行。

4.14 机器人（系统）状态变量的详细说明

4.14.1 记载项目的说明

- 【功能】 : 显示变量的功能。
- 【格式】 : 显示变量的自变量的输入方法。[] 表示可以省略的意义。
请参照系统状态变量（机器人状态变量）、在代入以外也可以使用条件式，但是，为了在格式的例子中单纯的标记，只可以参照或代入记述。
- 【例子】 : 显示使用变量的程序例。
- 【用语】 : 显示自变量的意义、范围等。
- 【说明】 : 显示详细的功能及注意事项。
- 【相关指令】 : 显示相关指令。
- 【关连参数】 : 显示相关的参数。
- 【关连系统状态变量】 : 显示相关连的系统变量。

4.14.2 各机器人状态变量（系统状态变量）的说明

以下各变量的说明以字母顺序显示。

C Com

【功能】

通过 **Open** 指令，对要开启的端口的参数进行设定。在变更通信对象时使用。

【格式】

例) C_Com (< 通信端口号码 >) = "ETH: < 服务器侧 IP 地址 > [, < 端口号码 >]"

【用语】

ETH: 表示对象为以太网的识别符号

< 通信端口号码 > 指定通过 **Open** 指令指定的 COM 的号码（端口种类通过参数 **COMDEV** 分配）1 ~ 8。

< 服务器侧 IP 地址 > 服务器侧的 IP 地址（不能省略）

< 端口号码 > 服务器侧的端口号码（省略时，使用参数 **NETPORT** 的设定值）

【例文】

在参数 **COMDEV** 的第 2 要素中设定了 "OPT12" 时的例子

```

1 C_Com(2)="ETH:192.168.0.10,10010" '将通信端口 COM2 对应的通信对象的 IP 地址设定为
                                     192.168.0.10，端口号码设定为 10010
2 Open "COM2:" AS #1                '开启上述端口
3 If M_Open(1)<>1 Then 110           '不能连接到服务器时，执行环路
4 Print #1, "HELLO"                 '发送字符串
5 Input #1, C1$                      '接收字符串
6 Close #1                          '端口关闭
7 C_Com(2)="ETH:192.168.0.11,10011" '将通信端口 COM2 对应的通信对象的 IP 地址设定为
                                     192.168.0.11，端口号码设定为 10011
8 Open "COM2:" AS #1                '开启上述端口
9 If M_Open(1)<>1 Then 170           '不能连接到服务器时，执行环路
10 Print #1, C1$                     '发送字符串
11 Input #1, C2$                     '接收字符串
12 Close #1                          '端口关闭
13 Hlt                               '暂停程序
14 End                               '结束

```

【说明】

- (1) 通过参数 **NETHSTIP**、**NETPORT** 指定控制器的通信对象，只要不变更该通信对象，就无需使用本指令。
- (2) 仅在以太网的数据链路的客户机时有效的功能。
- (3) 由于需要对 **Open** 指令的通信参数进行设定，因此需要在 **Open** 指令之前执行。
- (4) 接通电源时，使用通过参数 **NETHSTIP**、**NETPORT** 指定的设定值。执行本指令时，将会暂时性的变更由同一参数指定的值。有效状态持续到关闭电源为止。重新开启电源后，返回到通过参数设定的值。
- (5) 即使在 **Open** 指令之后执行，此时开启的状态也不会发生变化。这种情况下，需要先通过 **Close** 指令关闭端口，然后再执行 **Open** 指令。
- (6) 格式错误时，在程序编辑阶段不会报警，但在执行程序时会报警。

【相关参数】

NETHSTIP、**NETPORT**

C_Date

【功能】

将现在的日期以年 / 月 / 日的形式返回。

【格式】

例) <字符串变量> = C_Date

【例子】

1 C1\$=C_Date ' 在 C1\$ 输入 "2007/04/01"。

【说明】

- (1) 输入现在的日期。
- (2) 读取专用。设定日期的情况下请在示教单元设定。

【关连系统状态参数】

[C_Time](#)

C_Maker

【功能】

返回机器人控制器的制造商侧的情报。

【格式】

例) <字符串变量> = C_Maker

【例子】

1 C1\$=C_Maker ' 在 C1\$ 代入 "COPYRIGHT2007....."。

【说明】

- (1) 返回机器人控制器的制造商侧的情报。
- (2) 读取专用。

【关连系统状态参数】

[C_Mecha](#)

C_Mecha

【功能】

返回获得控制权的机械名（机器人的型号）。

【格式】

例) <字符串变量> = C_Mecha [(<数值>)]

【用语】

<字符串变量> 指定代入的字符串变量。
<数值> 输入机器号码。1 ~ 3，省略时为 1。

【例子】

1 C1\$=C_Mecha(1) '在 C1\$ 代入 "RV-12SQ"。
(插槽 1 获得机械 1 (“RH-3FH5515-D”) 的控制权时)

【说明】

- (1) 返回指定的任务插槽获得控制权的机械名（机器人的型号）。
- (2) 指定了未获得机械控制权的任务插槽时，返回 “”（无）。
- (3) 读取专用。

【相关指令】

[GetM \(Get Mechanism\)](#)、[RelM \(Release Mechanism\)](#)

C_Prg

【功能】

返回被选择的程序名（号码）。

【格式】

例) <字符串变量> = C_Prg [(<数值>)]

【用语】

<字符串变量> 指定代入的字符串变量。
<数值> 1 ~ 32，代入任务插槽号码。省略时为 1。

【例子】

1 C1\$=C_Prg(1) '在 C1\$ 代入 "10"。(程序名为 “10” 时)

【说明】

- (1) 在指定的任务插槽中代入已被设定（加载）的程序名（号码）。
- (2) 以单一任务使用的情况下，任务插槽号码会变成 1。
- (3) 在操作面板设定时，会变为该程序名（号码）。
- (4) 读取专用。
- (5) 指定程序没有被加载的任务插槽的情况下，执行时会发生报警。

C_Time

【功能】

将现在的时间，以时 / 分 / 秒的形式返回（24 小时表记）。

【格式】

例) <字符串变量>=C_Time

【例子】

1 C1\$=C_Time ' 在 C1\$ 代入 "01/05/20"。

【说明】

- (1) 返回现在的时间。
- (2) 读取专用。
- (3) 时间的变更在示教单元执行。

【关连系统状态参数】

[C_Date](#)

C_User

【功能】

返回在参数 "USERMSG" 设定的数据。

【格式】

例) <字符串变量>=C_User

【例子】

1 C1\$=C_User ' 在 C1\$ 代入参数「USERMSG」所设定的文字。

【说明】

- (1) 返回在参数 "USERMSG" 所设定的数据。
- (2) 读取专用。
- (3) 参数的变更请以计算机支持软件或示教单元执行。

J_Curr

【功能】

返回现在位置的关节型的数据。

【格式】

例) <关节型变量>=J_Curr [(<机器号码>)]

【用语】

<关节型变量> 指定代入的关节型变量。
<机器号码> 代入机器号码。1～3，省略时为1。

【例子】

1 J1=J_Curr '在1代入现在的关节位置。

【说明】

- (1) 代入指定机器号码的现在位置的关节型数据。
- (2) 读取专用。

【关连系统状态参数】

P_Curr

J_ColMxl

【功能】

返回冲突检知功能有效中的推测转矩及实际转矩差的最大值。

【格式】

例) <关节型变量>=J_ColMxl [(<机器号码>)]

【用语】

<关节型变量> 指定代入的关节型变量。
 <机器号码> 代入机器号码。1 ~ 3，省略时为 1。

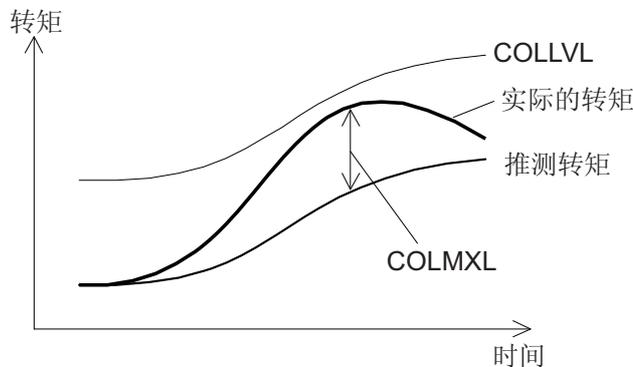
【文例】

```

1 M1=100 ' 设定各轴的冲突容许标准的初始值
2 M2=100
3 M3=100
4 M4=100
5 M5=100
6 M6=100
7 *LBL
8 ColLvl M1,M2,M3,M4,M5,M6,, ' 设定各轴的冲突容许标准
9 ColChk On ' 使冲突检知有效 (转矩误差最大值的计算开始)
10 Mov P1
    .
    .
50 ColChk Off ' 冲突检知无效 (转矩误差最大值的计算结束)
51 M1=J_ColMxl(1).J1+10 ' 计算使各轴都保有 10% 的剩余冲突容许 Level
52 M2=J_ColMxl(1).J2+10 (10% 为参考值, 并非实际的保证值)
53 M3=J_ColMxl(1).J3+10
54 M4=J_ColMxl(1).J4+10
55 M5=J_ColMxl(1).J5+10
56 M6=J_ColMxl(1).J6+10
57 GoTo *LBL
    
```

【说明】

(1) 通过 ColChk On ~ Off 为止的动作，保持各轴的推测转矩及实际转矩的误差最大值。



(2) 此值为 100% 的时候，表示误差最大值和冲突容许 Level 的厂商初始值相同。

(3) 冲突检知的使用禁止的机器人，全轴以 0.0 被返回。

(4) 在 ColChk On、ColLvl 指令执行时、伺服开启时，最大值会被初始化为 0.0。

(5) 为关节型的变量，作为关节变量直接就那样的读取的话，值会变换成 rad → deg。使用的时候请参照例子，将各轴要素代入变量后再使用。

(6) 读取专用。

【相关指令】

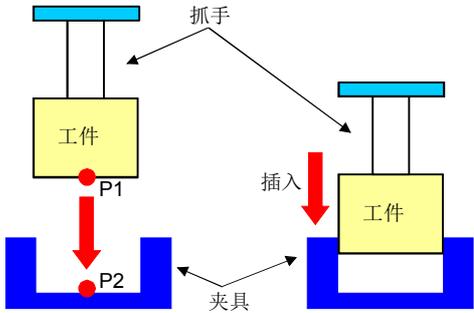
ColChk (Col Check)、 ColLvl (ColLevel)

【相关系统状态变量】

M_ColSts、 P_ColDir

【自动设定冲突检知 Level 的采样程序】

以下为自动设定冲突检知 Level 的程序。

采样程序	说明（包含注释）
 <pre> ***** 冲突检知 Level 自动设定 ***** 'GoSub *LEVEL ' 冲突检知 Level 自动设定程序 'HLT ***** *MAIN Oadl ON LoadSet 2,2 ColLvl M_01,M_02,M_03,M_04,M_05,M_06,, Mov PHOME Mov P1 Dly 0.5 ColChk ON Mvs P2 Dly 0.5 ColChk OFF Mov PHOME End </pre>	<p>自动设定如左图所示的从开始点 P1 移动到结束点 P2 之间的冲突检知 Level。</p> <p>执行冲突检知 Level 自动设定子程序的指令。仅在进行自动设定时，取消起始处的注释，以便使用。</p> <p>以最佳加减速动作。 读取抓手和工件的情报。（加减速时间的最佳化）</p> <p>重新设定冲突检知 Level。</p> <p>移动到 PHOME（待机点）。 移动到 P1（开始点）。</p> <p>将冲突检知设为有效。</p> <p>将冲突检知设为无效。</p> <p>返回到 PHOME（待机点）。 程序的结束行。</p>

采样程序	说明（包含注释）
<pre> ***** LEVEL FIX ***** *LEVEL Mov PHOME M1=0 'J1 轴的冲突容许 Level（初始化） M2=0 'J2 轴的冲突容许 Level（初始化） M3=0 'J3 轴的冲突容许 Level（初始化） M4=0 'J4 轴的冲突容许 Level（初始化） M5=0 'J5 轴的冲突容许 Level（初始化） M6=0 'J6 轴的冲突容许 Level（初始化） ColLvl 500,500,500,500,500,500,, ' 设定冲突检知 Level(500%) For MCHK=1 To 10 Dly 0.3 Mov P1 Dly 0.3 Colhk ON ' 冲突检知有效 Mvs P2 Dly 0.3 ColChk OFF ' 冲突检知无效 If M1<J_ColMxl(1).J1 Then M1=J_ColMxl(1).J1 If M2<J_ColMxl(1).J2 Then M2=J_ColMxl(1).J2 If M3<J_ColMxl(1).J3 Then M3=J_ColMxl(1).J3 If M4<J_ColMxl(1).J4 Then M4=J_ColMxl(1).J4 If M5<J_ColMxl(1).J5 Then M5=J_ColMxl(1).J5 If M6<J_ColMxl(1).J6 Then M6=J_ColMxl(1).J6 Next MCHK M_01=M1+10 M_02=M2+10 M_03=M3+10 M_04=M4+10 M_05=M5+10 M_06=M6+10 ColLvl M_01,M_02,M_03,M_04,M_05,M_06,, Mvs P1 Mov PHOME RETURN ***** </pre>	<p>冲突检知 Level 自动调整子程序。</p> <p>从 J1 轴到 J6 轴，冲突检知 Level 设定为 500%（最高值）。（确认动作时，在经由通路上没有障碍物）</p> <p>可自动检测冲突检知 Level，但考虑到误差问题，请执行多次。此处的参考值为重复 10 次。 J_COLMXL: 冲突检知有效中的推断转矩和实际转矩的误差的最大值 考虑到误差问题，在 10 次计测中记忆其最大值。</p> <p>一般电源开启后，检知 Level 将被代入参数「ColLvl」的值。因此，需将自动设定的值记录在程序外部变量中。此处的 "+10" 是在通过自动检知找出的值的基础上加上 10% 的余量。 ※10% 仅供参考。根据使用的系统的不同，可能会有无法正常动作的情况。 建议在通过实际机器进行确认的基础上，进行最佳的调整。请参照第 163 页的「4.13 指令的详细说明」的 ColLvl (ColLevel)。</p>

J_ECurr

【功能】

返回现在的编码器脉冲。

【格式】

例) <关节型变量> = J __ ECurr [(<机器号码>)]

【用语】

<关节型变量> 指定代入关节型变量。(为脉冲值, 请利用关节型变量。)
<机器号码> 代入机器号码。 1 ~ 3、省略时为 1。

【例子】

1 JA=J_ECurr(1) '在 JA 代入机器的编码器脉冲值。
2 MA=JA.J1 '在 MA 读出 J1 轴的编码器脉冲值。

【说明】

- (1) 返回的值为脉冲值, 但是代入型的值请使用关节型。之后, 指定关节成份数据且利用数值变量代入。
- (2) 读取专用。

J Fbc/J AmpFbc

【功能】

J_Fbc: 返回到依据编码器的反馈脉冲所生成的关节型现在位置。

J_AmpFbc: 返回各轴现在电流反馈值。

【格式】

例) <关节型变量> = J __ Fbc [(<机器号码>)]

例) <关节型变量> = J __ AmpFbc [(<机器号码>)]

【用语】

<关节型变量> 指定代入关节型变量。

<机器号码> 将机器号码代入。1 ~ 3、省略时为1。

【例子】

1 J1=J_Fbc ' 在 J1 代入依据现在伺服的反馈值所生成的关节现在位置。

2 J2=J_AmpFbc ' 在 J2 代入现在电流反馈值。

【说明】

(1) J_Fbc 为返回到编码器的反馈脉冲所生成的关节型现在位置。

(2) J_Fbc 为确认往伺服的指令值和实际的伺服的延迟的差。

(3) J_Fbc 可以确认在 Cmp Jnt 指令有差的情况。

(4) 读取专用。

【关连系统状态变量】

P_Fbc

J Origin

【功能】

返回到已设定原点时的关节数据。

【格式】

例) <关节型变量> = J __ Origin [(<机器号码>)]

【用语】

<关节型变量> 指定代入关节型变量。

<机器号码> 将机器号码代入。1 ~ 3、省略时为1。

【例子】

1 J1=J_Origin(1) ' 在 J1 代入机器 1 的原点设定位置

【说明】

(1) 返回到已设定原点时的关节数据。

(2) 使用在机器人位置偏移等情况，做原点位置确认用。

(3) 读取专用。

M_Acl/M_DAcI/M_NAcI/M_NDAcI/M_AcISts

【功能】

返回加减速时间相关情报。

M_Acl: 返回现在的加速时间比率。(%)

M_DAcI: 返回现在的减速时间比率。(%)

M_NAcI: 返回加速时间的初始值。(100%)

M_NDAcI: 返回减速时间的初始值。(100%)

M_AcISts: 返回现在位置的加减速状态。

(现在的加减速状态 0= 停止中 1= 加速中 2= 匀速中 3= 减速中)

【格式】

例) <数值变量> = M_Acl [(<数式>)]

例) <数值变量> = M_DAcI [(<数式>)]

例) <数值变量> = M_NAcI [(<数式>)]

例) <数值变量> = M_NDAcI [(<数式>)]

例) <数值变量> = M_AcISts [(<数式>)]

【用语】

<数值变量>

指定代入数值变量。

<数式>

1 ~ 32、将任务插槽号码代入。省略时为现在的插槽。

【例子】

1 M1=M_Acl ' 在 M1 代入在任务插槽 1 设定的加速时间比率。

2 M1=M_DAcI(2) ' 在 M1 代入在任务插槽 2 设定的减速时间比率。

3 M1=M_NAcI ' 在 M1 代入在任务插槽 1 设定的加速时间初始值比率。

4 M1=M_NDAcI(2) ' 在 M1 代入在任务插槽 2 设定的减速时间初始值比率。

5 M1=M_AcISts(3) ' 在 M1 代入在任务插槽 3 代入现在的加减速状态。

【说明】

(1) 加减速时间比率为对应各个机器人的最高加减速时间 (初始值) 的比率。50% 的情况下, 加减速时间会变成加一倍, 会变成慢慢的执行加减速的意思。

(2) M_NAcI, M_NDAcI 通常会返回到 100(%)。

(3) 读取专用。

M BsNo

【功能】

返回现在的基本坐标号码。

【格式】

例) <数值变量> = M__BsNo [(<机器号码>)]

【用语】

<数值变量> 指定要代入的数值变量。
 <机器号码> 指定 1 ~ 3 的机器号码。(省略时为 1)
 可使用常数、变量、逻辑 / 算术式、函数。
 指定了实数 / 双精度实数时, 小数点以后进行四舍五入。

【例文】

```
1 M1=M_BsNo      ' 将机器号码 1 的基本坐标号码代入变量 M1。
2 If M1=1 Then    ' 基本坐标号码为 1 时, 向 P1 移动。
3 Mov P1
4 Else            ' 基本坐标号码为 1 以外时, 向 P2 移动。
5 Mov P2
6 Endlf
```

【说明】

- (1) 读取现在设定的基本坐标号码 (参数: MEXBSNO)。
- (2) 根据读取值, 设定为以下的坐标系。
 - a) 0: 系统初始值 (P_Nbase)
 - b) 1 ~ 8: 工件坐标系号码 1 ~ 8 (参数: WK1CORD ~ WK8CORD)
 - c) -1: 基本变换设定为上述以外。
 (通过 Base 指令指定基本变换数据, 或直接编辑参数: MEXBS)

【相关指令】

Base

【相关参数】

MEXBSNO、WK n CORD (n 为 1 ~ 8)、MEXBS

M_BrkCq

【功能】

返回最后已执行的含 **Break** 指令的行的执行结果。

- 1: 有 **Break** 时
- 0: 没有 **Break** 时

【格式】

例) <数值变量> = M__BrkCq [(<数式>)]

【用语】

- <数值变量> 指定代入数值变量。
- <数式> 1 ~ 32、指定任务插槽号码。省略时为现在的插槽。

【例子】

```

1 While M1<>0                    '
2 If M2=0 Then Break            '
3 WEnd
4 If M_BrkCq=1 Then Hlt        '如果 While 内的 Break 被执行的话，进行 Hlt。

```

【说明】

- (1) 确认 **Break** 指令是否已被执行的状态。
- (2) 读取专用。
- (3) 即使只参照一次 **M_BRrkCq** 变量的话，**Break** 状态会被删除（值会变成 0）。因此，想留下状态的情况下，请代入数值变量后储存。
- (4) 即使参照示教单元的监视画面，**Break** 状态也会被删除。

M_BTime

【功能】

返回电池剩余容量时间（单位：时间）。

【格式】

例) <数值变量> = M__BTime

【用语】

- <数值变量> 指定代入的数值变量。

【例子】

```

1 M1=M_BTime                    '在 M1 代入电池剩余容量时间。

```

【说明】

- (1) 返回在现时点电池所持有的剩余时间。
- (2) 电池的持有时间的初始值依机型会有所不同。
- (3) 将机器人控制器的电源关闭中的时间积算后，返回到从电池保持时间初始值 14600 开始减掉的时间。
- (4) 读取专用。

M_CavSts

【功能】

返回检测到干涉的机器人的 CPU 号码。

- 1 ~ 3: 干涉确认检测中
- 0: 未检测到干涉确认

本功能对对应机型有所限定。关于详细内容请参照 [「5.24 关于干涉回避功能」](#)。

【格式】

```
例) Def Act 1,M_CavSts [ (<机器号码> ) ] <=0 GoTo *LCAV,S
```

【用语】

<机器号码> 填入机器号码。1 ~ 3、省略时为 1。

【例文】

参照第 500 页的 [「5.24.10 采样程序」](#)。

【说明】

- (1) 检测到机器人间的干涉确认时，写入检测到干涉的机器人 CPU 号码（1 ~ 3）。检测到与自由平面极限的干涉确认时，写入 1。
- (2) 在执行 End 指令之前不会自动清零，因此需要通过中断处理进行清零。
- (3) 保持执行 End 指令或程序复位、或通过 CavChk On 指令执行 "CavChk Off"（将干涉回避的停止功能设为无效）之前的值。
- (4) 在以 NoErr 模式使用时的 Def Act 指令中，用于中断条件。
- (5) 可参照及写入。

M_CmpDst

【功能】

返回伺服弹性功能执行时的指令值和实际的机器人的偏移量 (mm)。

【格式】

例) <数值变量> = M__CmpDst [(<机器号码>)]

【用语】

<数值变量> 指定代入的数值变量。
<机器号码> 将机器号码代入。1 ~ 3、省略时为 1。

【例子】

1 Mov P1
2 CmpG 0.5,0.5,1.0,0.5,0.5, , , ' 柔软の設定
3 Cmp Pos, &B00011011 ' 使成为柔软状态
4 Mvs P2
5 M_Out(10)=1
6 Mvs P1
7 M1=M_CmpDst(1) ' 在 M1 代入动作指令的指令位置和实际的现在位置的差。
8 Cmp Off ' 返回通常状态。

【说明】

- (1) 确认伺服弹性功能执行时的位置偏移量时使用。
- (2) 读取专用。

M_CmpLmt

【功能】

返回伺服弹性功能执行时的指令值是否有超过各种限制。

- 1: 超过限制
- 0: 没有超过限制。

【格式】

例) Def Act 1, M_CmpLmt [(<机器号码>)]=1 GoTo *LMT

【用语】

<机器号码> 代入机器号码。 1 ~ 3、省略时为 1。

【例子】

```

1 Def Act 1, M_CmpLmt(1)=1 GoTo *LMT      ' 作为插入 1 的条件定义
2 '
3 '

10 Mov P1
11 CmpG 1,1,0,1,1,1,1,1
12 Cmp Pos, &B100      ' 使伺服弹性模式有效
13 Act 1=1             ' 使插入 1 有效
14 Mvs P2              '
15 '

100 *LMT
101 Mvs P1             ' 将往 P2 的动作中断且返回到 P1
102 Reset Err         ' 将发生中的报警复位
103 Hlt               ' 将运行中断

```

【说明】

- (1) 在伺服弹性模式的指令值超过限制使报警发生的时候，使用插入处理从报警状态回归的时候使用。
- (2) 作为各种限制，确认伺服弹性模式的指令值的关节动作范围和动作速度、指令位置与实际的位置偏移量。
- (3) 将伺服电源关闭使伺服弹性模式为无效的情况下，会变成 0。
- (4) 读取专用。

M_ColSts

【功能】

返回冲突检知的状态。

- 1: 冲突检知中
- 0: 冲突未检出

【格式】

例) Def Act 1, M_ColSts [(<机器号码>)]=1 GoTo *LCOL,S

【用语】

<机器号码> 代入机器号码。 1 ~ 3、省略时为 1。

【例子】

```

1 Def Act 1,M_ColSts(1)=1 GoTo *HOME,S ' 在检测到冲突的时候，定义将执行处理做中断
2 Act 1=1
3 ColChk On,NOErr ' 在报警非发生模式，冲突检知为有效
4 Mov P1
5 Mov P2 ' 在 40 ~ 70 行执行中检测到冲突的话，跳转到中断处理
6 Mov P3
7 Mov P4
8 Act 1=0
  .
  .
100 *HOME ' 冲突检知时的中断处理
101 ColChk Off ' 冲突检知为无效
102 Servo On ' 伺服开启
103 PESC=P_ColDir(1)*(-2) ' 制作待避动作的移动量
104 PDST=P_Fbc(1)+PESC ' 制作待避位置（在冲突的位置（P_Fbc）加算退避移动量）
105 Mvs PDST ' 往待避位置移动
106 Error 9100 ' 让使用者定义的低标准报警发生，运行停止。

```

【说明】

- (1) 在检测到冲突时会变成 1，伺服关闭后解除冲突状态的话会变成 0。
- (2) 在 NOErr 模式的 Def Act 指令做插入条件使用。
- (3) 读取专用。

M_Cstp

【功能】

返回程序是否在循环停止中的状态。

1: 循环停止被输入, 循环停止运行实际效力中
(操入面板的 END 键输入、或循环停止信号输入)

0: 上述以外

【格式】

例) <数值变量> = M_Cstp

【用语】

<数值变量> 指定代入的数值变量。

【例子】

1 M1=M_Cstp ' 在 M1 输入 1。(循环停止中的情况)

【说明】

- (1) 在将程序连续运行的状态, 从操作面板按下 END 键的话, 会变成循环停止状态。将此时的状态作为 1 返还。
- (2) 读取专用。

M_Cys

【功能】

返回程序是否在循环中的状态。

1: 循环运行中 (以插槽参数 SLT* 运行模式~...)
(操入面板的 END 键输入、或循环停止信号输入)

0: 上記以外

【格式】

例) <数值变量> = M_Cys

【用语】

<数值变量> 指定代入的数值变量。

【例子】

1 M1=M_Cys ' 在 M1 输入 1。(循环运行中的情况)

【说明】

- (1) 将程序起动时, 可以用参数等设定连续运行或循环运行的运行模式。将此运行模式返还。
- (2) 即使指定插槽参数为 CYC, 以 XRun 指定连续运行的情况下, 值会变 0。
- (3) 读取专用。

M DIn/M DOut

【功能】

无法在 CR7xx-Q 系列使用此状态变量。

使用在往 CC-Link（选配）的远程寄存器的写入或参照时使用。

M_DIn: 参照输入寄存器的值。

M_DOut: 输出或参照到寄存器。

【格式】

例) <数值变量> = M_DIn (<数式 1 >)

例) <数值变量> = M_DOut (<数式 2 >)

【用语】

<数值变量> 指定将 CC-Link 的寄存器的值代入数值变量。

<数式 1 > 指定 CC-Link 的寄存器号码。(6000 ~)

<数式 2 > 指定 CC-Link 的寄存器号码。(6000 ~)

【例子】

1 M1=M_DIn(6000) ' 在 M1 代入 CC-Link 的输入寄存器 6000 号的值。

2 M1=M_DOut(6000) ' 在 M1 代入 CC-Link 的输出寄存器 6000 号的值。

3 M_DOut(6000)=100 ' 将 100 写入 CC-Link 的输出寄存器 6000 号。

【说明】

(1) M_DIn 为读取专用。

M DIn32

【功能】

从外部设备经由 CC-Link 的寄存器获取 32 位数据。

【格式】

```
例) <数值变量>=M_DIn32(<算式 / 输入寄存器号码>)
```

【用语】

- <数值变量> 指定要代入的长精度整数的数值变量。
- <算式 / 输入寄存器号码> 可指定的 CC-Link 的寄存器号码的范围为 6000 ~ 6254。

【例文】

- 1 M1&=M_DIn32(6000) '取出 CC-Link 的寄存器号码 6000 与 6001 开始的 32bit 的值，该值加入 M1& 中。
- 2 M2%=6002 'M2% 中加入 6002。
- 3 M3&=M_DIn32(M2%) And &H7FFFF '取出 M2% 中指定的 CC-Link 寄存器号码开始的 32bit 的值，该值的低位 19bit 的值加入 M3& 中。

【说明】

- (1) 从指定的 CC-Link 寄存器号码开始使用 2 点，获取 32bit 数据。
- (2) <数值变量> 中应使用用于长精度整数的变量。其他情况下，将无法正常获取信息。
- (3) CC-Link 的寄存器号码应指定为 6000 ~ 6254。指定超出范围的号码时，将全部返零，不会发生错误。
- (4) M_DIn32 为读取专用。

M_ErCode

【功能】

返回当前发生的错误的详细错误号码。

【格式】

例) <数值变量>=M_ErCode

【用语】

<数值变量> 指定 32 位长精度整数。
(16 位单精度整数中代入时会发生超出范围错误)
读出的值的含义如 **【说明】 (2)** 所示。

【例文】

1 *LBL: If M_Err=0 Then *LBL	' 等待至发生错误为止。
2 MD&=M_ErCode	' 读出详细错误号码 (代入长精度整数中)
3 MS%=Int(MD&/1000) Mod 100	' 取出详细错误号码的 2 位位数。

【说明】

- (1) 发生多个错误时，返回错误等级最高的信息。(仅限 1 个)
- (2) 详细错误号码为最多 9 位数的号码。
 XXXXYYYY .. XXXX : O/P 或 T/B 中显示的 4 位数的错误号码
 YYYYY : 详细错误号码 (内容取决于错误)

M_Err/M_ErrLvl/M_ErrNo

【功能】

返还机器人所发生关于报警的情报。

M_Err: 返还报警是否在发生中

M_ErrLvl: 返还已发生报警的 Level

M_ErrNo: 返还发生的报警号码

【格式】

例) <数值变量> = M_Err

例) <数值变量> = M_ErrLvl

例) <数值变量> = M_ErrNo

【用语】

<数值变量> 指定代入的数值变量。

M_Err : 0/1 = 无报警发生 / 报警发生中

M_ErrLvl : 0/1/2/3/4/5/6 = 无报警发生 / 警告 / 低等级报警 / 高等级报警
/ 警告 1 / 低等级报警 1 / 高等级报警 1

注) 各用语和意义见表 4-18

M_ErrNo : 当前发生的报警的编号

【例子】

1 *LBL: If M_Err=0 Then *LBL ' 等待到报警发生为止。

2 M2=M_ErrLvl ' 在 M2 输入报警的 Level。

3 M3=M_ErrNo ' 在 M3 输入报警号码。

【说明】

(1) 通常的程序在发生报警 (caution 除外) 的时候会中断, 但是利用 SLT* 参数在起动条件设定通常执行 (ALWAYS) 的程序, 可以利用此变量监视控制器的报警状态。通常执行程序即使在其它的程序发生报警, 也不会停止。

(2) 发生多个报警时, 返回报警等级最高的报警情报。

(3) M_ErrLvl 返回的报警等级及其意义如下所示。

表 4-18: 报警等级及其意义

报警等级	用语	意义	报警复位
0	无报警	未发生报警	-
1	警告	发生警告, 程序继续	[RESET] 键
2	低等级报警	执行中的程序将中断	[RESET] 键
3	高等级报警	执行中的程序将中断, 且切断伺服电源	[RESET] 键
4	警告 1	发生警告, 程序继续	电源复位
5	低等级报警 1	执行中的程序将中断	电源复位
6	高等级报警 1	执行中的程序将中断, 且切断伺服电源	电源复位

【相关指令】

Error、Reset Err(Reset Error)

M_ESpd

【功能】

返回 Ex-T 控制 /Ex-T 样条插补指令执行时的 Ex-T 坐标通过速度 [mm/s]。

【格式】

例) <数值变量> = M_ESpd [(<机械号码>)]

【用语】

- <数值变量> 指定要代入的数值变量。
- <机械号码> 设定正在执行 Ex-T 控制 /Ex-T 样条插补的机械号码。
 设定范围：1 ~ 3
 省略时：1
 指定了不存在的机械号码的情况下，执行时会发生 L3870（指定的机械号码为无效的值）错误。

【例文】

```
1 Wait M_00=1            ' 等待 Ex-T 样条插补开始
2 *L1:If M_ESpd<50 Then GoTo *L1
3 M_Out(100)=1          'Ex-T 坐标通过速度为 50[mm/s] 以上时，将输出 100 置为 ON
```

【说明】

- (1) 返回 Ex-T 控制 /Ex-T 样条插补指令执行时的 Ex-T 坐标通过速度 [mm/s]。
- (2) 通过在多任务中浏览 M_ESpd 的值，可以根据 Ex-T 坐标通过速度执行运算处理及信号输出等处理。
- (3) 未执行 Ex-T 控制 /Ex-T 样条插补指令时，返回 0.0。
- (4) M_ESpd 根据状况返回下表所示的值。

表 4-19: M_ESpd 返回的值

状况	M_ESpd 返回的值
电源接通后	0.0
Ex-T 控制 /Ex-T 样条插补执行中	插补执行时的 Ex-T 坐标通过速度
Ex-T 控制 /Ex-T 样条插补以外的插补执行中	0.0
插补中断时	0.0
主程序的 End 指令执行后	0.0
程序复位操作后	0.0
不支持样条插补的机械	0（写入执行后也是 0）

(5) 读取专用。

【相关指令】

EMvc (E Move C)、EMvr (E Move R)、EMvr2 (E Move R2)、EMvr3 (E Move R3)、EMvs (E Move S)、EMvSpl (E Move Spline)

M_Exp

【功能】

返还自然对数的底 (2.71828182845905)。

【格式】

例) <数值变量> = M_Exp

【用语】

<数值变量> 指定代入的数值变量。

【例子】

1 M1=M_Exp ' 在 M1 代入自然对数的底 (2.71828182845905)。

【说明】

- (1) 利用在处理指数函数、对数函数的时候。
- (2) 读取专用。

M_Fbd

【功能】

指令位置和反馈位置的差。

【格式】

例) <数值变量> = M_Fbd [(<机器号码>)]

【用语】

<数值变量> 指定代入的数值变量。
<机器号码> 输入机器号码。1 ~ 3、省略时为 1。

【例子】

```

1 Def Act 1,M_Fbd>10 GoTo *SUB1,S ' 指令位置和反馈位置的差在 10mm 以上的话, 中断
2 Act 1=1 ' 插入有效
3 Torq 3,10 ' 以转矩指令将 3 轴设定在 10% 以下
4 Mvs P1 ' 动作
5 End

10 *SUB1
11 Mov P_Fbc ' 使指令位置和反馈位置一致
12 M_Out(10)=1 ' 输出信号 10 号
13 End ' 在误差为零的时候停止。

```

【说明】

- (1) 返还在动作指令的位置位置和由电机的反馈位置的差的函数。
差为机器人本体的 XYZ 轴的合成值。(不含行走轴的差)
在使用转矩指令等的情况下, 不想发出误差过大报警 (960,970 等) 的时候, 请和 Def Act 指令组合使用。
- (2) 读取专用。

【相关指令】

[Torq \(Torque\)](#)

【关连系统状态变量】

[P_Fbc](#)

M_G

【功能】

返还重力常数 (9.80665)。

【格式】

例) <数值变量> = M_G

【用语】

<数值变量> 指定代入的数值变量。

【例子】

1 M1=M_G ' 在 M1 代入重力常数 (9.80665)。

【说明】

- (1) 利用在重力相关运算的情况。
- (2) 读取专用。

M_HndCq

【功能】

返还抓手确认输入信号的值。

【格式】

例) <数值变量> = M_HndCq (<数式>)

【用语】

<数值变量> 指定代入的数值变量。
<数式> 输入抓手输入信号号码。
 1 ~ 8 (相当于从输入信号 900 到 907)。

【例子】

1 M1=M_HndCq(1) ' 在 M1 输入抓手 1 的状态。

【说明】

- (1) 返还抓手确认输入信号状态 (感应器等) 的 1 个位。
- (2) M_HndCq (1) 相当于输入信号 900 号。即使使用 M_In(900) 也会变成相同的结果。
- (3) 读取专用。

M_In/M_Inb/M_In8/M_Inw/M_In16

【功能】

返回输入信号的值。

M_In: 返回位信号。

M_Inb 或 M_In8: 以字节为单位 (8 位量) 返回。

M_Inw 或 M_In16: 以字为单位 (16 位量) 返回。

【格式】

例) <数值变量>=M_In(<数式>)

例) <数值变量>=M_Inb(<数式>) 或 M_In8(<数式>)

例) <数值变量>=M_Inw(<数式>) 或 M_In16(<数式>)

【用语】

<数值变量> 指定要代入的数值变量。补充内容见表 4-20。

<数式> 指定信号号码。补充内容见表 4-21。

(1)CR7xx-Q 系列

10000 ~ 18191: 多 CPU 间共有设备

716 ~ 731: 多重抓手

900 ~ 907: 抓手

(2)CR7xx-D 系列

0 ~ 255: 并行输入输出

716 ~ 731: 多重抓手

900 ~ 907: 抓手

2000 ~ 5071: PROFIBUS

6000 ~ 8047: CC-Link

【例文】

1 M1%=M_In(10010)

'在 M1 中输入输入信号 10010 号的值 (1 或 0)。

2 M2%=M_Inb(900)

'在 M2 中输入从输入信号 900 号起 8 位量的值。

3 M3%=M_Inb(10300) And &H7

'在 M3 中输入从输入信号 10300 号起 3 位量的值。

4 M4%=M_Inw(15000)

'在 M4 中输入从输入信号 15000 号起 16 位量的值。

【说明】

(1) 返回输入信号的状态。

(2) 从指定的信号号码中, M_In 返回 1 位、M_Inb/M_In8 返回 8 位、M_Inw/M_In16 返回 16 位量的数据。

(3) 输入信号号码到 32767 为止, 但是只有硬件实际存在的信号号码的值会返回。硬件不存在的信号号码的情况下不进行处理。

(4) 读取专用。



注意

为了保证机器人和周边装置间的信号传输, 请务必实施信号的互锁。如果未实施互锁, 信号将无法正确传输, 从而造成误动作。

【补充】

表 4-20: 可填入 < 数值变量 > 的变量

○: 可使用、×: 不可使用

位宽	数值变量的类型				其他变量		
	整数 例)M1%	长精度整数 例)M1&	单精度 实数 例)M1!	双精度 实数 例)M1#	位置 ^{注1)} 例)P1.X	关节 ^{注1)} 例)J1.J1	字符串 例)C1\$
M_In	○	○	○	○	○	○	×
M_Inb/M_In8	○	○	○	○	○	○	×
M_Inw/M_In16	○	○	○	○	○	○	×

注 1) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因输入信号是以弧度单位的数值存储的, 因此将无法明确掌握信号的 ON/OFF 状态。(在监视器等的显示中, 将变换为度的单位显示)
 例)P1.A=M_In(8) 且输入信号 8 为 ON 时, P1.A 显示为 “57.296”, 无法明确掌握信号的 ON/OFF 状态。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

表 4-21: 可填入 < 数式 > 的常数 · 变量

○: 可使用、×: 不可使用

位宽	常数的类型			数值变量的类型				其他变量		
	数值 ^{注1)} 例)12	2 进制数 例)&B1100	16 进制数 例)&HC	整数 例)M1%	长精度整数 例)M1&	单精度 实数 ^{注1)} 例)M1!	双精度 实数 ^{注1)} 例)M1#	位置变量 ^{注1) 注2)} 例)P1.X	关节变量 ^{注1) 注2)} 例)J1.J1	字符串 例)C1\$
M_In	○	○	○	○	○	○	○	○	○	×
M_Inb/M_In8	○	○	○	○	○	○	○	○	○	×
M_Inw/M_In16	○	○	○	○	○	○	○	○	○	×

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因设定的值将以弧度单位的数值存储处理, 因此信号号码的指定将变得困难。(在监视器等的显示中, 将变换为度的单位显示, 因此会显示与设定值相同的值)
 例) 即使为了指定输入信号 8 号而在 P1.A 中设定 8 (通过 T/B 等的按键输入), 内部处理也会对 0.14 (rad) 进行四舍五入而变为 0, 因此, 将会变为指定输入信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

【相关系统状态变量】

M_Out32

【相关指令】

Def IO (Define IO)

M_In32

【功能】

将 32 位宽的外部输入信号以数值方式读取。

【格式】

例) <数值变量>=M_In32(<数式>)

【用语】

<数值变量> 指定要代入的数值变量。补充内容见表 4-22。

<数式> 指定信号号码。补充内容见表 4-23。

(1)CR7xx-Q 系列

10000 ~ 18191: 多 CPU 间共有设备

716 ~ 731: 多重抓手

900 ~ 907: 抓手

(2)CR7xx-D 系列

0 ~ 255: 并行输入输出

716 ~ 731: 多重抓手

900 ~ 907: 抓手

2000 ~ 5071: PROFIBUS

6000 ~ 8047: CC-Link

【例文】

1 *ack_wait

2 If M_In(7)=0 Then *ack_wait '等待直到输入信号 7 开启为止（互锁）

3 M1&=M_In32(10000) '以数值方式向长精度数值变量 M1 中存储从输入信号 1000 开始的 32 位量的数据。

4 P1.Y=M_In32(10100)/1000.0 '以数值方式向位置变量 P1 的 Y 成分输入从输入信号 1000 开始的 32 位量的数据，乘以 1/1000（实数化）后存储。
（外部机器侧，作为设定到机器人的位置变量 P1 的成分数据 Y 的数值，将实数值乘以 1000，使其变为整数后输出到机器人时的示例）

【说明】

(1) 以数值方式从信号号码返回 32 位量的数据。

(2) 输入信号号码到 32767 为止，但是只有硬件实际存在的信号号码会返回值。硬件不存在的信号号码的情况下不进行处理。

(3) 请为 <数值变量> 指定长精度整数型或实数型的变量。

(4) 读取专用。



注意

为了保证机器人和周边装置间的信号传输，请务必实施信号的互锁。如果未实施互锁，信号将无法正确传输，从而造成误动作。

【补充】

表 4-22: 可填入 <数值变量> 的变量

○: 可使用、×: 不可使用

位宽	数值变量的类型				其他变量		
	整数 例)M1%	长精度整数 例)M1&	单精度 实数 例)M1!	双精度 实数 例)M1#	位置 ^{注1)} 例)P1.X	关节 ^{注1)} 例)J1.J1	字符串 例)C1\$
M_In32	×	○	○	○	○	○	×

注 1) 当变量所处理的数值是角度时（位置变量的成分数据 A、B、C 和关节变量的所有成分数据），32 位的输入数据将直接作为弧度单位的数值存储。在监视器等显示中，将转换为度的单位显示。

例)P1.A=M_In32(16)且该输入数据为 5（10 进制数）时，P1.A 显示为

“286.479”（deg）。

表 4-23: 可填入 < 数式 > 的常数・变量

○: 可使用、×: 不可使用

位宽	常数的类型			数值变量的类型				其他变量		
	数值 注1) 例)12	2 进制数 例)&B1100	16 进制数 例)&HC	整数 例)M1%	长精度整数 例)M1&	单精度实数 注1) 例)M1!	双精度实数 注1) 例)M1#	位置 注1) 注2) 例)P1.X	关节 注1) 注2) 例)J1.J1	字符串 例)C1\$
M_In32	○	○	○	○	○	○	○	○	○	×

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因设定的值将以弧度单位的数值存储处理, 因此信号号码的指定将变得困难。(在监视器等的显示中, 将变换为度的单位显示, 因此会显示与设定值相同的值)

例) 即使为了指定输入信号 8 号而在 P1.A 中设定 8 (通过 T/B 等的按键输入), 内部处理也会对 0.14 (rad) 进行四舍五入, 而变为 0, 因此, 将会变为指定输入信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

【相关系统状态变量】

M_In/M_Inb/M_In8/M_Inw/M_In16

【相关指令】

Def IO (Define IO)

M JOvrd/M NJOvrd/M OPOvrd/M Ovrđ/M NOvrd

【功能】

返还速度比例值

M_JOvrd: 关节插补用的速度比例以指令 JOvrd 指定的值

M_NJOvrd: 关节插补用的速度比例的初始值 (100%)

M_OPOvrd: 操作面板的速度比例值

M_Ovrđ: 现在的速度比例值, 以 Ovrđ 指令指定的值

M_NOvrd: 速度比例的初始值 (100%)

【格式】

例) <数值变量> = M_JOvrd [(<数式>)]

例) <数值变量> = M_NJOvrd [(<数式>)]

例) <数值变量> = M_OPOvrd

例) M_OPOvrd = <算式>

例) <数值变量> = M_Ovrđ [(<数式>)]

例) <数值变量> = M_NOvrd [(<数式>)]

【用语】

<数值变量>

指定代入的数值变量。

<算式>

在 M_JOvrd/M_NJOvrd/M_Ovrđ/M_NOvrd 算式中, 加入 1 ~ 32、任务插槽号码。省略时为当前的插槽。

在 M_OPOvrd 算式中, 加入倍率修调值。

【例子】

1 M1=M_Ovrđ

' 在 M1 输入现在的速度比例值。

2 M2=M_NOvrd

' 在 M2 输入现在的速度比例的初始值 (100%)。

3 M3=M_JOvrd

' 在 M3 输入现在的关节速度比例值。

4 M4=M_NJOvrd

' 在 M4 输入现在的关节速度比例的初始值。

5 M5=M_OPOvrd

' 在 M5 输入现在的 OP(操作面板) 速度比例值。

6 M6=M_Ovrđ(2)

' 在 M6 输入插槽 2 的现在的值。

7 M_OPOvrd=20

'OP(操作面板) 倍率修调中设定为 20。

注) 对 M_OPOvrd 的设定在以下的软件版本中有效。

F-Q 系列: R5 以上、F-D 系列: S5 以上

【说明】

(1) 在自变量省略的情况下, 返还现在的插槽状态。

(2) M_JOvrd/M_NJOvrd/M_Ovrđ/M_NOvrd 仅可读取。

M_OPOvrd 可读取、写入。

M_Line

【功能】

返还现在正执行的步号。

【格式】

例) <数值变量> = M_Line [(<数式>)]

【用语】

<数值变量> 指定代入的数值变量。
<数式> 1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

【例子】

1 M1=M_Line(2) ' 在 M1 输入插槽 2 的执行步号。

【说明】

- (1) 多任务运行时，监视其它的任务执行单步时使用。
- (2) 读取专用。

M_LdFact

【功能】

可以参照各关节单位的负载率。

【格式】

例) <数值变量> = M_LdFact (<轴号码>)

【用语】

<轴号码> 指定轴号码。(1 ~ 8)
<数值变量> 各轴的负载率被代入, 0 ~ 100(%)。

【例子】

```

1 Accel 100,100          ' 将全体的减速时间设为 100%。
2 *Label
3 Mov P1
4 Mov P2
5 If M_LdFact(2)>90 Then
6   Accel 50,50          ' 将加减速比下降 50%
7   M_SetAdl(2)=50      ' 将 J2 轴的加减速比再度设定为 50%。(实际为 50%×50%=25%)
8 Else
9   Accel 100,100       ' 返回加减速时间
10 EndIf
11 GoTo *Label

```

【说明】

- (1) 可参照各轴的负载率。
- (2) 负载率往各轴电机流动的电流, 其从流通时间所导出的。
- (3) 以高负载严苛的姿势长时间动作的话, 负载率会上升。
- (4) 负载率变成 100% 的话, 会发生过负载报警。在上述例子里, 负载率超过 90% 的情况下, 加减速时间会暂时下降到 50% 的例子。
- (5) 使负载率下降可以调降加减速时间或使机器人在不严刻的姿势待机、或将伺服电源切断, 皆是有效对策。
- (6) 作为对象的机器号码的初始值为机器 1。执行了 ReIM 指令后, 或是在多任务控制中, 在插槽 1 以外的程序中将机器 1 作为对象时, 无需通过 GetM 指令指定机器号码。以其他机器为对象时, 请执行 GetM 指令。

【関連命令】

Accel、Ovrd

【相关系统状态变量】

M_SetAdl

M_LdFMax

【功能】

可以浏览各关节轴单位的最大负载率。

【格式】

例) < 数值变量 >=M_LdFMax(< 轴号码 >)

【用语】

< 轴号码 > 指定轴号码。(1 ~ 8)
< 数值变量 > 代入各轴的负载率。为 0 ~ 100(%)。

【例文】

```
1 Accel 100,100                    ' 将整体的减速时间设为 100%。  
2 *Label  
3 Mov P1  
4 Mov P2  
5 If M_LdFMax(2)>90 Then Goto *LdErr Else Goto *Label  
6  
7 *LdErr  
8 Error 9101                        ' 用户错误输出  
9 End
```

【说明】

- (1) 可以浏览各轴的最大负载率。可以检查机器人动作的严峻性。
- (2) 负载率可以通过流向各轴电机的电流和电流流过的时间导出。
- (3) 高负载且长时间以严峻的姿势进行动作，则负载率会上升。
- (4) 负载率为 100% 时，会发生过负载错误。以上述例文为例，在负载率超出 90% 时，暂时将加减速时间降低至 50%。
- (5) 要降低负载率，以下的对策较为有效：降低加减速时间，或以合理的姿势让机器人待机，或切断伺服电源。
- (6) 对象机械号码的初始值为机械 1。执行 ReIM(Release Mechanism) 指令后，或在多任务控制中通过插槽 1 以外的程序将机械 1 作为对象时，不需要通过 GetM 指令进行机械号码指定。将其他的机械作为对象时，应执行 GetM 指令。

【相关指令】

Accel、Ovrd

M Mode

【功能】

返还操作面板的按钮开关的模式。

- 1: MANUAL
- 2: AUTO

【格式】

例) <数值变量> = M_Mode

【用语】

<数值变量> 指定代入的数值变量。

【例子】

1 M1=M_Mode ' 在 M1 输入按钮开关的状态。

【说明】

- (1) 多任务运行时，可利用在 ALWAYS 被执行（一直执行）的程序等。
- (2) 读取专用。

M On/M Off

【功能】

返还通常为 1(M_On) 或 0(M_Off)。

【格式】

例) <数值变量> = M_On

例) <数值变量> = M_Off

【用语】

<数值变量> 指定代入的数值变量。

【例子】

1 M1=M_On ' 在 M1 输入 1。
2 M2=M_Off ' 在 M2 输入 0。

【说明】

- (1) 返还通常为 1 或 0。
- (2) 读取专用。

M_Out/M_Outb/M_Out8/M_Outw/M_Out16

【功能】

写入到外部输出信号或进行参照。

M_Out: 输出信号的位

M_Outb 或 M_Out8: 输出信号的字节 (8 位置)

M_Outw 或 M_Out16: 输出信号的字 (16 位置)

【格式】

例) M_Out(< 数式 1>)=< 数值 2>
 例) M_Outb(< 数式 1>) 或 M_Out8(< 数式 1>)=< 数值 3>
 例) M_Outw(< 数式 1>) 或 M_Out16(< 数式 1>)=< 数值 4>
 例) M_Out(< 数式 1>)=< 数值 2> Dly < 时间 >
 例) < 数值变量 >=M_Out(< 数式 1>)

【用语】

< 数式 1> 指定输出信号的号码。补充内容见表 4-24。

(1)CR7xx-Q 系列

10000 ~ 18191: 多 CPU 间共有设备

716 ~ 723: 多重抓手

900 ~ 907: 抓手

(2)CR7xx-D 系列

0 ~ 255: 并行输入输出

716 ~ 723: 多重抓手

900 ~ 907: 抓手

2000 ~ 5071: PROFIBUS

6000 ~ 8047: CC-Link

< 数值 2>,< 数值 3>,< 数值 4>

将要输出的数据以数值变量、常数或数值运算式记述。补充内容见表 4-25。

数值的范围

< 数值 2>: 0 或 1 (&H0 或 &H1)

< 数值 3>: -128 ~ +127 (&H80 ~ &H7F)

< 数值 4>: -32768 ~ +32767 (&H8000 ~ &H7FFF)

< 时间 >

将脉冲输出的输出时间以常数或数值运算式记述。单位: [秒]

< 数值变量 >

指定存储输出信号的数值变量。补充内容见表 4-26。

【例文】

1 M_Out(902)=1 ' 开启输出信号 902(1 位)。

2 M_Outb(10016)=&HFF ' 开启从输出信号 10016 开始的 8 位置。

3 M_Outw(10032)=&HFFFF ' 开启从输出信号 10032 开始的 16 位置。

4 M4=M_Outb(10200) And &H0F ' 在 M4 中输入从输出信号 10200 开始的低位 4 位的值。

【说明】

(1) 在写入到外部输出信号或进行参照时使用。

(2) 从指定的信号号码中, M_Out 输出 1 位、M_Outb/M_Out8 输出 8 位、M_Outw/M_Out16 输出 16 位置的数据。

(3) 关于脉冲输出请参照第 202 页的「Dly (Delay)」。

(4) 通过将参数: SYNCIO 设定为高速模式, 可以加快向外部输出信号的更新周期。为了提高性能, 建议在实施信号互锁的基础上, 在高速模式下使用。详细内容请参照第 418 页的「5.2 信号参数」的 SYNCIO。



注意

为了保证机器人和周边装置间的信号传输, 请务必实施信号的互锁。如果未实施互锁, 信号将无法正确传输, 从而造成误动作。

【补充】

表 4-24: 可填入 < 数式 1> 的常数・变量

○: 可使用、×: 不可使用

位宽	常数的类型			数值变量的类型				其他变量		
	数值 ^{注1)} 例)12	2 进制数 例)&B1100	16 进制数 例)&HC	整数 例)M1%	长精度整数 例)M1&	单精度 实数 ^{注1)} 例)M1!	双精度 实数 ^{注1)} 例)M1#	位置 ^{注1)} , ^{注2)} 例)P1.X	关节 ^{注1)} , ^{注2)} 例)J1.J1	字符串 例)C1\$
M_Out	○	○	○	○	○	○	○	○	×	×
M_Outb/M_Out8	○	○	○	○	○	○	○	○	×	×
M_Outw/ M_Out16	○	○	○	○	○	○	○	○	×	×

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因设定的值将以弧度单位的数值存储处理, 因此信号号码的指定将变得困难。(在监视器等的显示中, 将变换为度的单位显示, 因此会显示与设定值相同的值)

例) 为了指定输出信号 8 号, 即使在 P1.A 中代入 8 (P1.A=8), 内部处理也会对 0.14 (rad) 进行四舍五入, 值将变为 0, 因此, 将会变为指定输出信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

表 4-25: 可填入 < 数值 2>、< 数值 3>、< 数值 4> 的常数・变量

○: 可使用、×: 不可使用

位宽	常数的类型			数值变量的类型				其他变量		
	数值 例)12	2 进制数 例)&B1100	16 进制数 例)&HC	整数 例)M1%	长精度整数 例)M1&	单精度 实数 例)M1!	双精度 实数 例)M1#	位置 例)P1.X	关节 例)J1.J1	字符串 例)C1\$
M_Out	○	○	○	○	×	×	×	×	×	×
M_Outb/M_Out8	○	○	○	○	×	×	×	×	×	×
M_Outw/M_Out16	○	○	○	○	×	×	×	×	×	×

表 4-26: 可填入 < 数值变量 > 的变量

○: 可使用、×: 不可使用

位宽	数值変数の型				その他の変数		
	整数 例)M1%	长精度整数 例)M1&	单精度 实数 例)M1!	双精度 实数 例)M1#	位置 ^{注1)} 例)P1.X	关节 ^{注1)} 例)J1.J1	字符串 例)C1\$
M_Out	○	○	○	○	○	○	×
M_Outb/M_Out8	○	○	○	○	○	○	×
M_Outw/M_Out16	○	○	○	○	○	○	×

注 1) 当变量所处理的数值是角度时 (位置变量的 A、B、C 成分数据和关节变量的所有成分数据), 以弧度单位的数值对现在的输出状态进行存储, 运算等的处理将以弧度单位进行。在监视器等显示中, 将变换为度的单位显示。

【相关系统状态变量】

M_Out32

【相关指令】

Def IO (Define IO)

【相关参数】

SYNCIO

M_Out32

【功能】

向 32 位宽的外部输出信号写入数值或进行参照。

【格式】

例) M_Out32(<数式 1>)=<数值>
例) <数值变量>=M_Out32(<数式 1>)

【用语】

- <数式 1> 指定输出信号的号码。补充内容见表 4-24。
- (1)CR7xx-Q 系列
10000 ~ 18191: 多 CPU 间共有设备
716 ~ 723: 多重抓手
900 ~ 907: 抓手
- (2)CR7xx-D 系列
0 ~ 255: 并行输入输出
716 ~ 723: 多重抓手
900 ~ 907: 抓手
2000 ~ 5071: PROFIBUS
6000 ~ 8047: CC-Link
- <数值> 将要输出的数据以数值变量、常数或数值运算式记述。补充内容见表 4-25。
数值的范围: -2147483648 ~ +2147483647 (&H80000000 ~ &H7FFFFFFF)
- <数值变量> 指定存储输出信号的数值变量。补充内容见表 4-26。

【例文】

- ```
1 M_Out32(10000)=P1.X * 1000 '将位置变量 P1 的 X 坐标值乘以 1000 (整数化) 输出到从输出信号
 号码 10000 开始的 32 位量中。
2 *ack_wait
3 If M_In(7)=0 Then *ack_wait '等待直到输入信号 7 开启为止 (互锁)
4 P1.Y=M_In32(10100)/1000.0 '以数值方式向位置变量 P1 的 Y 成分中输入从输入信号 10100 开始的
 32 位量的数据, 乘以 1/1000 (实数化) 后存储。
 (外部机器侧, 作为设定到机器人的位置变量 P1 的成分数据 Y 的数值, 将实数值乘以 1000, 使其变为整数后输出到机器人时的示例)
```

### 【说明】

- (1) 在写入到 32 位宽的外部输出信号或进行参照时使用。
- (2) 从指定的信号号码输出 32 位量的数据。
- (3) 通过将参数: SYNCIO 设定为高速模式, 可以加快向外部输出信号的更新周期。为了提高性能, 建议在实施信号互锁的基础上, 在高速模式下使用。详细内容请参照第 418 页的「5.2 信号参数」的 SYNCIO。

## 注意

为了保证机器人和周边装置间的信号传输, 请务必实施信号的互锁。如果未实施互锁, 信号将无法正确传输, 从而造成误动作。

【補足】

表 4-27: 可填入 < 数式 1 > 的常数・变量

○: 可使用、×: 不可使用

| 位宽      | 常数的类型      |                   |                 | 数值变量的类型     |                |                    |                    | 其他变量         |               |               |
|---------|------------|-------------------|-----------------|-------------|----------------|--------------------|--------------------|--------------|---------------|---------------|
|         | 数值<br>例)I2 | 2 进制数<br>例)&B1100 | 16 进制数<br>例)&HC | 整数<br>例)M1% | 长精度整数<br>例)M1& | 单精度<br>实数<br>例)M1! | 双精度<br>实数<br>例)M1# | 位置<br>例)P1.X | 关节<br>例)J1.J1 | 字符串<br>例)C1\$ |
| M_Out32 | ○          | ○                 | ○               | ○           | ○              | ○                  | ○                  | ○            | ×             | ×             |

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因设定的值将以弧度单位的数值存储处理, 因此信号号码的指定将变得困难。(在监视器等的显示中, 将变换为度的单位显示, 因此会显示与设定值相同的值)

例) 即使为了指定输出信号 8 号而在 P1.A 中代入 8 (P1.A=8), 内部处理也会对 0.14 (rad) 进行四舍五入而变为 0, 因此, 将会变为指定输出信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

表 4-28: 可填入 < 数值 > 的常数・变量

○: 可使用、×: 不可使用

| 位宽      | 常数的类型      |                   |                 | 数值变量的类型     |                |                   |                    | 其他变量         |               |               |
|---------|------------|-------------------|-----------------|-------------|----------------|-------------------|--------------------|--------------|---------------|---------------|
|         | 数值<br>例)I2 | 2 进制数<br>例)&B1100 | 16 进制数<br>例)&HC | 整数<br>例)M1% | 长精度整数<br>例)M1& | 单精度<br>实数<br>例)M1 | 双精度<br>实数<br>例)M1# | 位置<br>例)P1.X | 关节<br>例)J1.J1 | 字符串<br>例)C1\$ |
| M_Out32 | ○注 1)      | ○注 1)             | ○注 1)           | ○注 1)       | ○              | ○                 | ○注 2)              | ○            | ○             | ×             |



注意

注 1)

请注意: 指定的常数为容纳在 2 进制数的 16 位中的数值 (-32768 ~ +32767) 时, 位 15 (第 16 位) 开启的数值作为负的数值处理, 高位 16 位被符号扩展, 输出全部 ON 状态的信号。

例) 指定 -32768(&B1000000000000000) 后, 会输出 &B11111111111111110000000000000000 (2 进制数)。

[对策]

如下所示, 在将常数代入长精度整数型的数值变量的基础上, 再代入本状态变量 M\_YDevD 后, 可输出 &B00000000000000001000000000000000 (2 进制数)。

- 1 M1&=32768
- 2 M\_YDevD(&H20)=M1&

注 2) 可输出数值的范围为 -2147483648 ~ 2147483647。

表 4-29: 可填入 < 数值变量 > 的变量

○: 可使用、×: 不可使用

| 位宽      | 数值变量的类型     |                |                    |                    | 其他变量         |               |               |
|---------|-------------|----------------|--------------------|--------------------|--------------|---------------|---------------|
|         | 整数<br>例)M1% | 长精度整数<br>例)M1& | 单精度<br>实数<br>例)M1! | 双精度<br>实数<br>例)M1# | 位置<br>例)P1.X | 关节<br>例)J1.J1 | 字符串<br>例)C1\$ |
| M_Out32 | ×           | ○              | ○                  | ○                  | ○            | ○             | ×             |

注 1) 当变量所处理的数值是角度时 (位置变量的 A、B、C 成分数据和关节变量的所有成分数据), 将以弧度单位的数值存储现在的输出状态。在监视器等显示中, 将变换为度的单位显示。

【相关系统状态变量】

M\_Out/M\_Outb/M\_Out8/M\_Outw/M\_Out16

【相关指令】

Def IO (Define IO)

【相关参数】

SYNCIO

## M\_PI

### 【功能】

返回圆周率 (3.14159265358979)。

### 【格式】

例) <数值变量> = M\_PI

### 【用语】

<数值变量>      指定代入的数值变量。

### 【例子】

1 M1=M\_PI      ' 在 M1 输入 3.14159265358979。

### 【说明】

- (1) 被代入的变量会变成实数。
- (2) 读取专用。

## M\_Psa

### 【功能】

已指定的任务插槽返还为程序可选择。

1: 可以选择程序

0: 不可以 (程序为中断状态的时候)

### 【格式】

例) <数值变量> = M\_Psa [ (<数式> ) ]

### 【用语】

<数值变量>      指定代入的数值变量。

<数式>            1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

### 【例子】

1 M1=M\_Psa(2)      ' 在 M1 输入任务插槽 2 的程序可以选择状态。

### 【说明】

- (1) 已指定的任务插槽返还为程序可选择。
- (2) 读取专用。



## M\_RDst

### 【功能】

返还机器人移动中的到目的位置为止的剩余距离 (mm)。

### 【格式】

例) <数值变量> = M\_RDst [ (<数式> ) ]

### 【用语】

<数值变量>      指定代入的数值变量。  
<数式>            1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

### 【例子】

1 Mov P1 Wthlf M\_RDst<10, M\_Out(10)=1    ' 在往 P1 移动中，离目的距离的剩余距离在 10mm 以下的话，将输出信号 10 号开启。

### 【说明】

- (1) 想要在机器人移动中的特定位置执行处理的情况下使用。
- (2) 读取专用。

## M\_Run

### 【功能】

返还已指定的任务插槽的程序是否在执行中的情报。

- 1: 执行中  
0: 执行中以外 (中断中或停止中)

### 【格式】

例) <数值变量> = M\_Run [ (<数式> ) ]

### 【用语】

<数值变量>      指定代入的数值变量。  
<数式>            1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

### 【例子】

1 M1=M\_Run(2)                    ' 在 M1 输入插槽 2 的实际有效状态。

### 【说明】

- (1) 已指定的插槽为执行中的情况为 1、停止中 (或中断中) 的情况为 0。
- (2) 判断程序是否在停止中 (执行行为前头的情况)，请利用 M\_Run 和 M\_Wai 的组合。
- (3) 读取专用。

## M\_SetAdl

### 【功能】

最佳加减速为有效 (Oadl On) 时, 设定指定轴的加减速时间比例。因为每个轴可以单独设定因此可以降低负载高的轴的电机负载。此外, 可像 OvrD 指令、Spd 指令和 Accel 指令一样的在全轴统一设定的方法不同, 对节拍的影响极少。初始值会变成参数 JADL 的值。

本状态变量有限定使用机型 (请参照【可以使用的机器人机型】)。

### 【格式】

例) M\_SetAdl (<轴号码>) = <数值变量>

### 【用语】

<轴号码>            指定轴号码。  
 <数值变量>        指定标准的加减速时间的比例, 1 ~ 100。单位为%。  
                       初始值为最佳加减速补偿率参数 (JADL) 值

### 【例子】

```

1 Accel 100,50 ' 全体的减速比例设定在 50%。
2 If M_LdFact(2)>90 Then ' J2 轴的负载率超过 90% 的话
3 M_SetAdl(2)=70 ' J2 轴的加减速比例设定 70%。
4 EndIf ' 加速 70%(=100%×70%)、减速 35%(=50%×70%)
5 Mov P1
6 Mov P2
7 M_SetAdl(2)=100 ' 将 J2 轴的加减速比例返还到 100%。
8 Mov P3 ' 加速 100%、减速 50%
9 Accel 100,100 ' 将全体的加减速比例返还到 100%。
10 Mov P4

```

### 【说明】

- (1) 可以用轴单位设定最佳加减速有效时的加减速时间比例。指定为 100% 的情况下, 加减速会变成最短时间。
- (2) 依据本状态变量, 可以像减低过负载报警和过热报警发生时轴的负载那样的设定加减速时间。
- (3) 本状态变量的设定可以在加速时间和减速时间两边做设定。
- (4) 和 Accel 指令合并使用的情况, 针对以最佳加减速所算出的加减速时间, 而且有做 Accel 指令的加减速分配时适用。
- (5) 由于 Accel 指令, 以指定的比例加减速时间会有变化。在本状态的指定, 由于是从每轴有独立设定以及考虑电机的负载后所算出的加减速时间, 所以加减速时间的变化, 会和已指定的比例出现少许不同的值。

### 【相关指令】

Accel、OvrD、Spd (Speed)

### 【关连系统变量】

M\_LdFact

## M\_SkipCq

### 【功能】

返还在最后包含已执行的 **Skip** 指令行的实际执行结果。

1: 已 **Skip** 的情况

0: 没有 **Skip** 的情况

### 【格式】

例) <数值变量> = M\_SkipCq [ (<数式> ) ]

### 【用语】

<数值变量>      指定代入的数值变量。

<数式>            1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

### 【例子】

1 Mov P1 Wthlf M\_In(10)=1,Skip      '在往 P1 移动开始时，输入输入信号 10 号的话，将 Mov 指令跳过。

2 If M\_SkipCq=1 Then GoTo \*Lskip      '已执行 skip 的话，跳转到 \*Lskip。

:

10 \*Lskip

### 【说明】

(1) 确认 **skip** 指令被执行时的状态。

(2) 读取专用。

(3) 即使只参照 **M\_SkipCq** 变量一次，**skip** 状态会被清除（值会变成 0）。因此，想要留下状态的情况下，请代入数值变量并储存。

## M\_Spd/M\_NSpd/M\_RSpd

### 【功能】

返还直线插补、关节插补时的速度情报。

M\_Spd: 现在被设定的速度

M\_NSpd: 初始值 (最佳速度控制)

M\_RSpd: 现在的指令速度

### 【格式】

例) <数值变量> = M\_Spd [( <数式> )]

例) <数值变量> = M\_NSpd [( <数式> )]

例) <数值变量> = M\_RSpd [( <数式> )]

### 【用语】

<数值变量> 指定代入的数值变量。

<数式> 1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

### 【例子】

1 M1=M\_Spd ' 在 M1 输入现在被设定的速度。

2 Spd M\_NSpd ' 将速度返回到最佳速度控制模式。

### 【说明】

(1) M\_RSpd 返还机器人动作中的现在的指令速度。

(2) M\_RSpd 利用在多任务程序或 Wth、WthIf 的时候里。

(3) 读取专用。

## M\_SplPno

### 【功能】

在样条插补执行中，返回最新的已通过的路径点号码。

样条插补的开始位置为 1。

注) 该指令可在软件版本为 Ver.R5 以上 (F-Q 系列) / Ver.S5 以上 (F-D 系列) 的版本中使用。

### 【格式】

例) <数值变量>=M\_SplPno[(<机械号码>)]

### 【用语】

<数值变量> 指定要代入浏览结果的数值变量。

<机械号码> 设定正在执行样条插补的机械号码。

设定范围：1 ~ 3

省略时：1

指定了不存在的机械号码的情况下，执行时会发生 L3870（指定的机械号码为无效的值）错误。

### 【例文】

```

1 Wait M_00=1 ' 等待样条插补开始
2 *L1:If M_SplPno < 5 Then GoTo *L1 ' 等待直至通过路径点 5
3 M1=M_Inw(100) ' 通过路径点 5 后，获取输入信号 100 ~ 115 的状态
4 *L2:If M_SplPno < 10 Then GoTo *L2 ' 等待直至通过路径点 10
5 M_01=M1 ' 通过路径点 10 后，将 M1 的内容代入 M_01

```

### 【说明】

- (1) 在样条插补执行中，返回已通过的路径点中最新的路径点号码。
- (2) 通过在多任务中浏览 M\_SplPno 的值，可以根据样条插补的进展，执行运算处理及信号输出等处理。
- (3) M\_SplPno 根据状况返回下表所示的值。

表 4-30: M\_SplPno 返回的值

| 状况             | M_SplPno 返回的值      |
|----------------|--------------------|
| 电源接通后          | 0                  |
| 样条插补执行中        | 1 ~ (与样条插补的进展对应的值) |
| 样条插补结束后        | 样条插补的结束位置的路径点号码    |
| 主程序的 End 指令执行后 | 0                  |
| 程序复位操作后        | 0                  |
| 不支持样条插补的机械     | 0                  |

- (4) M\_SplPno 为读取专用的状态变量。

## M\_SplVar

### 【功能】

在样条插补执行中，返回最新的已通过的路径点中所设定的数值设定值。通过写入，也可以更改为任意的值。

注) 该指令可在软件版本为 Ver.R5 以上 (F-Q 系列) / Ver.S5 以上 (F-D 系列) 的版本中使用。

### 【格式】

例) <数值变量 1>=M\_SplVar [<机械号码>]

例) M\_SplVar [<机械号码>]=<数值变量 2>

### 【用语】

- <数值变量 1> 指定要代入浏览结果的数值变量。  
 <机械号码> 设定正在执行样条插补的机械号码。  
 设定范围：1 ~ 3  
 省略时：1  
 指定了不存在的机械号码的情况下，执行时会发生 L3870（指定的机械号码为无效的值）错误。
- <数值变量 2> 指定要在 M\_SplVar 中设定的值。  
 设定范围：0 ~ 32767  
 设定了超出范围的值的情况下，执行时会发生 L2615（M\_SplVar 的设定值超出范围）错误或 L3110（自变量的值超出范围）错误。

### 【例文】

```

1 Wait M_00=1 ' 等待样条插补开始
2 *L1
3 Select M_SplVar
4 Case 1 ' 通过 M_SplVar 的值设定为 1 的路径点
5 M_SplVar=0 ' 将 M_SplVar 的值复位为 0
6 M_Out(100)=1 ' 将输出信号 100 置为 ON
7 Break
8 Default ' M_SplVar 的值不为 1
9 M_Out(100)=0 ' 将输出信号 100 置为 OFF
10 Break
11 End Select
12 If M_00=1 Then Goto *L1 ' 样条插补反复进行，直至结束

```

### 【说明】

- (1) 在样条插补执行中，返回已通过的路径点里最新的路径点中所设定的数值设定值。数值设定在样条文件的路径点数据中指定。
- (2) 通过在多任务中浏览 M\_SplVar 的值，可以根据样条插补的进展，执行运算处理及信号输出等处理。与 M\_SplPno 不同，即使是不相同的路径点也可以返回相同的值，因此如【例文】一样，反复进行同一输出信号的 ON/OFF 时，可以简化程序。
- (3) M\_SplVar 返回的值为 0 ~ 32767 的整数。此外，样条文件的数值设定中设定的“-1”的路径点，即使通过该路径点，M\_SplVar 的值也不会变化而是继续保持通过路径点时的值。

(4) M\_SplVar 根据状况返回表 4-31 所示的值。

表 4-31: M\_SplVar 返回的值

| 状况             | M_SplVar 返回的值                 |
|----------------|-------------------------------|
| 电源接通后          | 0                             |
| 样条插补执行中        | 与样条插补的进展对应的值<br>(样条文件内的数值设定值) |
| 通过指定为“-1”的路径点时 | 继续保持此时的值                      |
| 样条插补结束后        | 样条插补结束时的值                     |
| 主程序的 End 指令执行后 | 0                             |
| 程序复位操作后        | 0                             |
| M_SplVar 中写入值后 | 写入的值                          |
| 不支持样条插补的机械     | 0(写入执行后也是 0)                  |

(5) 通过至 M\_SplVar 的写入操作, 在 0 ~ 32767 的范围中, 可以设定任意的值。保持该值不变, 直至因样条插补或其他操作·处理导致值被更改。

## M\_Svo

### 【功能】

返还现在的伺服电源的状态。

1: 伺服电源开启 ON

0: 伺服电源关闭 OFF

### 【格式】

例) <数值变量> = M\_Svo [ (<机器号码> ) ]

### 【用语】

<数值变量> 指定代入位置变量。

<机器号码> 输入机器号码。1 ~ 3、省略时为 1。

### 【例子】

1 M1=M\_Svo(1) ' 在 M1 输入现在的伺服电源的状态。

### 【说明】

(1) 可以确认机器人的伺服状态。

(2) 读取专用。

## M\_Timer

### 【功能】

以毫秒单位计数时间。可以用于机器人的动作时间测定及计测正确的时间的情况。

### 【格式】

例) <数值变量> = M\_Timer (<数式>)

### 【用语】

<数值变量>      指定代入数值变量。  
<数式>            输入 1 ~ 8 的号码。括号无法省略。

### 【例子】

1 M\_Timer(1)=0  
2 Mov P1  
3 Mov P2  
4 M1=M\_Timer(1)      ' 在 M1 输入从 P1 到 P2 为止的移动时间 (ms)。  
                          例 ) 5.346 秒的情况下, M1 的值会变成 5346。  
5 M\_Timer(1)=1.5      ' 设定为 1.5。

### 【说明】

- (1) 也可以代入。代入时的单位被设定为秒。
- (2) 因为可以用毫秒 (ms) 单位计测因此可以测定正确的节拍时间。

## M\_Tool

### 【功能】

指定的号码的 TOOL 数据 (MEXTL1 ~ 4) 可以作为现在的 TOOL 数据使用的同时, 会设定于参数 MEXTL。且, 也可以读出现在的 TOOL 号码。

### 【格式】

例) <数值变量> = M\_Tool [ (<机器号码>) ]     ' 现在的 TOOL 号码的参照  
 例) M\_Tool [ (<机器号码>) ] = <数式>     ' 设定 TOOL 号码

### 【用语】

<数值变量>     指定代入位置变量。  
 <机器号码>     输入机器号码。1 ~ 3、省略时为 1。  
 <数式>     输入 1 ~ 4 的 TOOL 号码。

### 【例子】

#### ① TOOL 数据的设定

```
1 Tool (0,0,100,0,0,0) ' 指定 TOOL 数据 (0,0,100,0,0,0), 在 MEXTL 写入。
2 Mov P1
3 M_Tool=2 ' 将 TOOL 数据变更为 TOOL 号码 2(MEXTL2) 的值。
4 Mov P2
```

#### ② TOOL 号码的参照

```
1 If M_In(900)=1 Then ' 依据抓手输入信号切换 TOOL 数据。
2 M_Tool=1 ' 将 TOOL1 设定到 TOOL 数据。
3 Else
4 M_Tool=2 ' 将 TOOL2 设定到 TOOL 数据。
5 EndIf
6 Mov P1
```

### 【说明】

- (1) 将 TOOL 用的参数设定在 MEXTL1、MEXTL2、MEXTL3、MEXTL4 的值反映在 TOOL 数据, 也在参数 MEXTL 里写入。
- (2) TOOL 数据 1 ~ 4 对应 MEXTL1 ~ 4。
- (3) 参照时会读出现在被设定 TOOL 数据。
- (4) 读出的值为 0 的时候, MEXTL1 ~ 4 以外的 TOOL 数据会被视为现在的 TOOL 数据设定。
- (5) 示教单元的 TOOL 设定画面也可以做同样的设定。详细请参照第 28 页的「[3.2.9 工具数据的切换](#)」。

### 【关连参数】

MEXTL、MEXTL1、MEXTL2、MEXTL3、MEXTL4

### 【相关指令】

[Tool \(Tool\)](#)

## M\_Uar

**【功能】**

返回是否进入使用者定义领域。  
 位 0 ~ 15 对应领域 1 ~ 16，且各个位显示如下情报。  
 1: 使用者定义领域内、  
 0: 使用者定义领域外

**【格式】**

例) <数值变量> = M\_Uar [( <机器号码> )]

**【用语】**

<数值变量>            指定要代入的数值变量。  
 <机器号码>            填入机器号码。1 ~ 3、省略时为 1。

**【例文】**

1 M1 = M\_Uar(1) AND &H0004 ' 在 M1 中仅代入领域 3 的结果。  
 2 If M1<>0 Then M\_Out(10)=1 ' 在领域 3 范围内，即开启输出信号 10。

**【说明】**

- (1) 使用者定义领域的使用方法请参照第 435 页的「5.8 关于使用者定义领域」。
- (2) 对象为领域 1 ~ 16。如需要领域 17 ~ 32 的情报，请使用 M\_Uar32 获取。
- (3) 对象为在参数 AREAnAT(n 为领域号码 (n=1 ~ 16)) 中指定了 1 (信号输出) 的领域。
- (4) 要进行比较运算、逻辑运算时，以 10 进制数表示则位 15 为 1 时将会变为负数，因此建议使用 16 进制表示。
- (5) 读取专用。

**【相关系统变量】**

M\_Uar32

**【M\_Uar 的值和使用者定义领域的对应关系】**

| 位 | 领域 | 10 进制数 | 16 进制数 |
|---|----|--------|--------|
| 0 | 1  | 1      | &H0001 |
| 1 | 2  | 2      | &H0002 |
| 2 | 3  | 4      | &H0004 |
| 3 | 4  | 8      | &H0008 |
| 4 | 5  | 16     | &H0010 |
| 5 | 6  | 32     | &H0020 |
| 6 | 7  | 64     | &H0040 |
| 7 | 8  | 128    | &H0080 |

| 位  | 领域 | 10 进制数 | 16 进制数 |
|----|----|--------|--------|
| 8  | 9  | 256    | &H0100 |
| 9  | 10 | 512    | &H0200 |
| 10 | 11 | 1024   | &H0400 |
| 11 | 12 | 2048   | &H0800 |
| 12 | 13 | 4096   | &H1000 |
| 13 | 14 | 8192   | &H2000 |
| 14 | 15 | 16384  | &H4000 |
| 15 | 16 | -32768 | &H8000 |

例) 在使用者定义领域 5 及 10 内时，表示领域 5 的值 H0010 和表示领域 10 的值 H0400 相加的值 &H0200 作为 M\_Uar 的值返回。

## M\_Uar32

### 【功能】

返回是否有进入使用者定义领域。

位 0 ~ 31 对应领域 1 ~ 32，且各别的位情报表示如下。

1: 使用者定义领域内

0: 使用者定义领域外

### 【格式】

例) <数值变量> = M\_Uar32 [ (<机器号码> ) ]

### 【用语】

<数值变量> 指定代入位置变量。

<机器号码> 输入机器号码。1 ~ 3，省略时为 1。

### 【例子】

1 Def Long M1

2 M1& = M\_Uar32(1) AND &H00080000 ' 在 M1 只代入领域 20 的结果。

3 If M1&<>0 Then M\_Out(10)=1 ' 有进入到领域 20 的话，将输入信号 10 开启。

### 【说明】

(1) 使用者定义领域的使用者方法请参照第 435 页的「5.8 关于使用者定义领域」。

(2) 在<数值变量>使用 16 位整数型，值为超过 (OVER) 的情况下会发生报警。

请使用 32 位整数型。

(3) 在参数 AREAnAT(n 为领域号码 (n=1 ~ 32)) 指定为 1(信号输出) 的领域会成为对象。

(4) 执行比较运算、逻辑运算的情况下，以 10 进制记载，位 31 为 1 的情况下会变成负的数，因此，建议使用 16 进制记载。

(5) 读取专用。

### 【关连系统变量】

M\_Uar

### 【M\_Uar32 的值和使用者定义领域的对应】

| 位  | 领域 | 10 进制数 | 16 进制数     |
|----|----|--------|------------|
| 0  | 1  | 1      | &H00000001 |
| 1  | 2  | 2      | &H00000002 |
| 2  | 3  | 4      | &H00000004 |
| 3  | 4  | 8      | &H00000008 |
| 4  | 5  | 16     | &H00000010 |
| 5  | 6  | 32     | &H00000020 |
| 6  | 7  | 64     | &H00000040 |
| 7  | 8  | 128    | &H00000080 |
| 8  | 9  | 256    | &H00000100 |
| 9  | 10 | 512    | &H00000200 |
| 10 | 11 | 1024   | &H00000400 |
| 11 | 12 | 2048   | &H00000800 |
| 12 | 13 | 4096   | &H00001000 |
| 13 | 14 | 8192   | &H00002000 |
| 14 | 15 | 16384  | &H00004000 |
| 15 | 16 | 32768  | &H00008000 |

| 位  | 领域 | 10 进制数      | 16 进制数     |
|----|----|-------------|------------|
| 16 | 17 | 65536       | &H00010000 |
| 17 | 18 | 131072      | &H00020000 |
| 18 | 19 | 262144      | &H00040000 |
| 19 | 20 | 524288      | &H00080000 |
| 20 | 21 | 1048576     | &H00100000 |
| 21 | 22 | 2097152     | &H00200000 |
| 22 | 23 | 4194304     | &H00400000 |
| 23 | 24 | 8388608     | &H00800000 |
| 24 | 25 | 16777216    | &H01000000 |
| 25 | 26 | 33554432    | &H02000000 |
| 26 | 27 | 67108864    | &H04000000 |
| 27 | 28 | 134217728   | &H08000000 |
| 28 | 29 | 268435456   | &H10000000 |
| 29 | 30 | 536870912   | &H20000000 |
| 30 | 31 | 1073741824  | &H40000000 |
| 31 | 32 | -2147483648 | &H80000000 |

例) 进入使用者定义 5 及 10 的情况下，领域 5 将显示值 &H00000010 和领域 10 将显示值 &H00000400 有加上值 &H00000410，会视为 M\_Uar 的值返还。

## M\_UDevW/ M\_UDevD

### 【功能】

通过 CR7xx-Q 系列的控制器，向多 CPU 间共有内存直接写入或直接参照。（仅限 CR7xx-Q 系列控制器）  
（可不用经由 PLC 的梯形图程序，在多个机器人间更高速地获取互锁等的接口。此外，还可以获取运动控制器 CPU 等机器人 CPU 以外的共有内存情报。（参照第 506 页的「5.26 在机器人 CPU 之间直接通信」））

M\_UDevW: 以字为单位（16 位量）写入 / 参照。

M\_UDevD: 以双字为单位（32 位量）写入 / 参照。

### 【格式】

#### 参照

<数值变量> = M\_UDevW (<起始输入输出号码>, <共有内存地址>)

<数值变量> = M\_UDevD (<起始输入输出号码>, <共有内存地址>)

#### 写入

M\_UDevW (<起始输入输出号码>, <共有内存地址>) = <数值>

M\_UDevD (<起始输入输出号码>, <共有内存地址>) = <数值>

### 【用语】

<数值变量>

指定要代入输入输出信号的值的数值变量。

<起始输入输出号码>

通过常数或数值变量指定 CPU 单元的输入输出号码。

（指定的值为以 16 进制表示起始输入输出号码、且省略低位 1 位的值。）

范围：16 进制表示时为 &H3E0 ~ &H3E3（10 进制表示时为 992 ~ 995）

1 号机：&H3E0（10 进制表示时为 992）

2 号机：&H3E1（10 进制表示时为 993）

3 号机：&H3E2（10 进制表示时为 994）

4 号机：&H3E3（10 进制表示时为 995）

注）数据的写入仅限本号机有效。

<共有内存地址>

通过常数或数值变量指定 CPU 单元的共有内存地址。

各状态变量的有效范围如下所示：（10 进制数）

M\_UDevW: 10000 ~ 24335

M\_UDevD: 10000 ~ 24334

<数值>

通过常数或数值变量指定要写入的数据。

各状态变量的有效范围如下所示：

M\_UDevW: -32768 ~ 32767(&H8000 ~ &H7FFF)

M\_UDevD: -2147483648 ~ 2147483647

(&H80000000 ~ &H7FFFFFFF)

### 【例文】

1 M\_UDevW(&H3E1, 10010)=&HFFFF

' 在 2 号机 CPU（本号机）的共有内存地址 10010 中写入 &HFFFF（16 进制数）

2 M\_UDevD(&H3E1, 10011)=P1.X \* 1000

' 在 2 号机 CPU（本号机）的共有内存地址 10011/10012 的双字中写入位置变量 P1 的 X 坐标值乘以 1000 后的值

3 M1%=M\_UDevW(&H3E2, 10001) And &H7

' 在 M1 中代入 3 号机 CPU 的共有内存地址 10001 的低位 3 位的值

### 【说明】

(1) 向 PLC 的多 CPU 间共有内存写入或进行参照。

(2) 通过起始输入输出号码和共有内存地址指定对象共有内存。

(3) 写入的数据及参照所返回的数据，都是整数。

- (4) 从指定的共有内存地址中，M\_UDevW 以 1 字（16 位）量的数据为对象，M\_UDevD 以双字（32 位）量的数据为对象。
- (5) 在进行指定时，以 16 进制表示时为起始输入输出号码 &H3E0 ~ &H3E3（10 进制表示时为 992 ~ 995）；以 10 进制表示时为共有内存地址的写入或参照地址 10000 ~ 24335。
- (6) 仅本号机 CPU 的共有内存地址可写入。即使指定其他号机 CPU 的地址写入数据，也将不会更新。
- (7) 通过 M\_UdevD 为了防止 32 位数据分离，访问数据时请将共有内存的偶数地址置于起始处。

## 【补充】

表 4-32: 可填入 &lt; 数值变量 &gt; 的变量

○: 可使用、×: 不可使用

| 位宽      | 数值变量的类型      |                 |                     |                     | 其他变量                         |                               |                |
|---------|--------------|-----------------|---------------------|---------------------|------------------------------|-------------------------------|----------------|
|         | 整数<br>例 )M1% | 长精度整数<br>例 )M1& | 单精度<br>实数<br>例 )M1! | 双精度<br>实数<br>例 )M1# | 位置 <sup>注1)</sup><br>例 )P1.X | 关节 <sup>注1)</sup><br>例 )J1.J1 | 字符串<br>例 )C1\$ |
| M_UDevW | ×            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_UDevD | ×            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |

注 1) 当变量所处理的数值是角度时（位置变量的 A、B、C 成分数据和关节变量的所有成分数据），以弧度单位的数值对现在的输出状态进行存储，运算等的处理将以弧度单位进行。在监视器等显示中，将变换为度的单位显示。

表 4-33: 可填入 &lt; PLC 输入信号号码 &gt; &lt; 共有内存地址 &gt; 的常数・变量

| 位宽      | 常数的类型                      |                    |                  | 数值变量的类型      |                 |                                    |                                    | 其他变量                                 |                                       |                  |
|---------|----------------------------|--------------------|------------------|--------------|-----------------|------------------------------------|------------------------------------|--------------------------------------|---------------------------------------|------------------|
|         | 数值 <sup>注1)</sup><br>例 )12 | 2 进制数<br>例 )&B1100 | 16 进制数<br>例 )&HC | 整数<br>例 )M1% | 长精度整数<br>例 )M1& | 单精度<br>实数 <sup>注1)</sup><br>例 )M1! | 双精度<br>实数 <sup>注1)</sup><br>例 )M1# | 位置<br><sup>注1), 注2)</sup><br>例 )P1.X | 关节<br><sup>注1), 注2)</sup><br>例 )J1.J1 | 字符串<br>例 )C1\$\$ |
| M_UDevW | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                    | ×                                     | ×                |
| M_UDevD | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                    | ×                                     | ×                |

○: 可使用、×: 不可使用

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时（位置变量的成分数据 A、B、C 和关节变量的所有成分数据），因设定的值将以弧度单位的数值存储处理，因此信号号码的指定将变得困难。（在监视器等的显示中，将变换为度的单位显示，因此会显示与设定值相同的值）

例) 即使为了指定输出信号 8 号而在 P1.A 中代入 8 (P1.A=8)，内部处理也会对 0.14 (rad) 进行四舍五入而变为 0，因此，将会变为指定输出信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm，因此不会出现这种状态。

表 4-34: 可填入 &lt; 数值 &gt; 的常数・变量

○: 可使用、×: 不可使用

| 位宽      | 常数的类型                      |                    |                  | 数值变量的类型          |                 |                     |                     | 其他变量          |                |                  |
|---------|----------------------------|--------------------|------------------|------------------|-----------------|---------------------|---------------------|---------------|----------------|------------------|
|         | 数值 <sup>注1)</sup><br>例 )12 | 2 进制数<br>例 )&B1100 | 16 进制数<br>例 )&HC | 整数<br>例 )M1%     | 长精度整数<br>例 )M1& | 单精度<br>实数<br>例 )M1! | 双精度<br>实数<br>例 )M1# | 位置<br>例 )P1.X | 关节<br>例 )J1.J1 | 字符串<br>例 )C1\$\$ |
| M_UDevW | ○                          | ○                  | ○                | ○                | ×               | ×                   | ×                   | ×             | ×              | ×                |
| M_UDevD | ○ <sup>注2)</sup>           | ○ <sup>注2)</sup>   | ○ <sup>注2)</sup> | ○ <sup>注2)</sup> | ○               | ○                   | ○ <sup>注3)</sup>    | ○             | ○              | ×                |

注 1) 实数值将四舍五入。

注 2)  注意

请注意：指定的常数为容纳在 2 进制数的 16 位中的数值（-32768 ~ +32767）时，位 15（第 16 位）开启的数值作为负的数值处理，高位 16 位被符号扩展，输出全部 ON 状态的信号。

例) 指定 -32768(&B1000000000000000) 后，会输出  
&B11111111111111111000000000000000（2 进制数）。

[对策]

如下所示，在将常数代入长精度整数型的数值变量的基础上，再代入本状态变量 M\_YDevD 后，可输出 &B00000000000000001000000000000000（2 进制数）。

1 M1&=32768

2 M\_YDevD(&H20)=M1&

注 3) 可输出数值的范围为 -2147483648 ~ 2147483647。

**【参考】** 关于号机间数据的数据保证

根据本号机的读取和其他号机的数据写入 / 接收其他号机信息的时机，各号机的数据中可能会出现旧数据和新数据同时存在（数据的分离）的问题。

在通过使用 CPU 共有内存的程序进行通信中，防止数据分离的方法如图 4-34 所示。

1) 防止 32 位数据的分离

通过将偶数地址（例：地址 10002）置于起始处以访问 CPU 共有内存的使用者自由区域，可防止 32 位数据的分离。

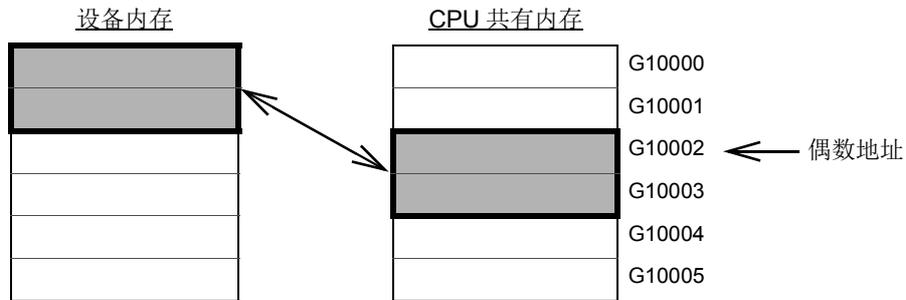


图 4-34：防止 32 位数据的分离

2) 防止超过 32 位的数据的分离

通过程序进行的读取从使用者自由区域的起始处开始按顺序读取。

此外，通过写入指令，从使用者自由区域的结束地址向起始地址写入发送数据。

因此，在要进行通信的数据的起始处设置用于互锁的设备，可防止进行通信的数据的分离。

## M\_Wai

### 【功能】

返还指定任务插槽的程序待机状态。

1: 中断中（程序为中断状态）

0: 中断中以外（运行中为停止中）

### 【格式】

|                              |
|------------------------------|
| 例) <数值变量> = M_Wai [( <数式> )] |
|------------------------------|

### 【用语】

<数值变量> 指定代入的数值变量。

<数式> 1 ~ 32、输入任务插槽号码。省略时为现在的插槽号码。

### 【例子】

1 M1=M\_Wai(1) ' 在 M1 输入插槽 1 的待机状态。

### 【说明】

(1) 确认程序是否在中断状态时使用。

(2) 判断程序是否在停止中（执行行为开头的情况），请将 M\_Run 和 M\_Wai 组合使用。

(3) 读取专用。

### 【关连系统变量】

M\_Wupov、M\_Wuprt、M\_Wupst

## M\_Wupov

### 【功能】

暖机运行状态的时候，为了将动作速度降低，请将指令速度乘以速度比例（暖机运行速度比例，单位：%）的值返还。

注) 关于暖机模式的详细内容请参照第 461 页的「5.19 关于暖机运行模式」。

### 【格式】

|                                  |
|----------------------------------|
| 例) <数值变量> = M_Wupov [( <机器号码> )] |
|----------------------------------|

### 【用语】

|        |                      |
|--------|----------------------|
| <数值变量> | 指定代入位置变量。            |
| <机器号码> | 输入机器号码。1 ~ 3、省略时为 1。 |

### 【例子】

1 M1=M\_Wupov(1) ' 在 M1 输入暖机运行速度比例的值。

### 【说明】

- (1) 机器人变成暖机运行状态（将速度自动的下降的动作状态）时，在为了将动作速度降低，请将指令速度乘以速度比例（暖机运行速度比例）的值返还时使用。
- (2) 暖机运行模式为无效的情况，及控制器前面的 MODE 开关变成“MANUAL”的时候，机器 LOCK 中通常会变成 0。
- (3) 从平常的状态转变到暖机运行状态时，或电源起动后直接变成暖机运行状态的情况下，在参数 WUPOVRD 的第 1 要素（暖机运行速度比例的初始值）会被视为初始值设定，依据机器人的动作，M\_Wupov 的值会变大大然后暖机运行状态在被解除时，M\_Wupov 的值会变成 100。
- (4) 在暖机运行状态，实际的速度比例如下所示。
  - 关节插补动作时 = ( 操作面板 (T/B) 的速度比例设定值 ) × ( 程序速度比例 (Ovrđ 指令) ) × ( 关节速度比例 (JOvrđ 指令) ) × 暖机运行速度比例
  - 直线插补动作时 = ( 操作面板 (T/B) 的速度比例设定值 ) × ( 程序速度比例 (Ovrđ 指令) ) × ( 直线指定速度 (Spđ 指令) ) × 暖机运行速度比例
- (5) 读取专用。

## M\_Wuprt

### 【功能】

为了解除暖机运行状态，请返还对象轴必须动作的时间（秒）。

注) 关于暖机模式的详细内容请参照第 461 页的「5.19 关于暖机运行模式」。

### 【格式】

|                                   |
|-----------------------------------|
| 例) <数值变量> = M_Wuprt [ (<机器号码> ) ] |
|-----------------------------------|

### 【用语】

<数值变量>            指定代入位置变量。  
<机器号码>            输入器号码。1 ~ 3、省略时为 1。

### 【例子】

10 M1=M\_Wuprt(1)    ' 在 M1 输入对象轴必须动作的时间。

### 【说明】

- (1) 机器人变成暖机运行状态（将速度自动的下降的动作状态）时，在确认以参数 WUPAXIS（暖机运行模式对象轴）指定的关节轴还有多少的时间动作的话，暖机运行状态才能被解除时使用。
- (2) 暖机运行模式为无效的情况，通常会返还 0。
- (3) 从平常的状态转变到暖机运行状态时，或电源起动后直接变成暖机运行状态的情况下，在参数 WUPTIME 的第 1 要素（暖机运行模式的有效时间）会被视为初始值设定，依据机器人的动作，M\_Wuprt 的值会变小然后值变成 0 的时候，暖机运行状态会被解除。
- (4) 暖机运行模式的对象轴有复数存在的情况下，返还其中动作时间最短的轴。  
例如：在某对象轴 (A) 动作剩 20 秒就会解除暖机运行状态的时候 (M\_Wuprt=20 的时候)、持续停止的其它对象轴 (B) 从平常的状态往暖机运行状态变化的话，(B) 会变成动作最短的轴（动作时间 0 秒），因此 (B) 的必须动作时间（=暖机运行模式的有效时间、初始值为 60 秒）会变成此状态变量的值 (M\_Wuprt=60)。
- (5) 读取专用。

## M\_Wupst

### 【功能】

返还在暖机运行状态被解除后，再度变为暖机状态为止的时间（秒）。

注）关于暖机模式的详细内容请参照第 461 页的「5.19 关于暖机运行模式」。

### 【格式】

例) <数值变量> = M\_Wupst [ (<机器号码> ) ]

### 【用语】

<数值变量>            指定代入位置变量。  
<机器号码>            输入机器号码。 1 ~ 3、省略时为 1。

### 【例子】

1 M1=M\_Wupst(1)    ' 在 M1 输入再次变成暖机运行状态为止的时间。。

### 【说明】

- (1) 机器人的暖机运行状态（将速度自动的下降的动作状态）被解除时，确认以参数 WUPAXIS（暖机运行模式对象轴）指定的关节轴的停止时间还要持续多久才会变成暖机运行状态时使用。
- (2) 暖机运行模式为无效的情况，返还以参数 WUPTIME 的第 2 要素（暖机运行模式再开时间）指定的时间。
- (3) 暖机运行状态在被解除时，对象轴动作的话，于参数 WUPTIME 的第 2 要素（暖机运行模式再开时间）所指定的时间会被视为初始值、对象轴停止的话，M\_Wupst 的值会变小。然后当值变为 0 的时候，会成为暖机运行状态。
- (4) 对象轴有复数存在的情况下，返还其中停止持续最久的那个轴的值。
- (5) 读取专用。

## M\_XDev/M\_XDevB/M\_XDevW/M\_XDevD

### 【功能】

在 CR7xx-Q 系列的控制器中参照 PLC 输入信号 (X) 的值。(仅限 CR7xx-Q 系列控制器)

(直接参照其他号机管理的输入输出单元 / 输入输出混合单元的输入信号。(参照第 502 页的「5.25PLC 输入输出单元直接控制」))

M\_XDev: 以位为单位参照。

M\_XDevB: 以字节为单位 (8 位量) 参照。

M\_XDevW: 以字为单位 (16 位量) 参照。

M\_XDevD: 以双字为单位 (32 位量) 参照。

### 【格式】

```
<数值变量> = M_XDev [(< PLC 输入信号号码 >)]
<数值变量> = M_XDevB [(< PLC 输入信号号码 >)]
<数值变量> = M_XDevW [(< PLC 输入信号号码 >)]
<数值变量> = M_XDevD [(< PLC 输入信号号码 >)]
```

### 【用语】

<数值变量>

指定要代入输入信号的值的数值变量。

< PLC 输入信号号码 >

通过常数或数值变量指定要参照的输入信号号码。

设定范围以 16 进制表示为 &H0 ~ &HFFF (以 10 进制表示为 0 ~ 4095), 各状态变量分别如下所示。

M\_XDev : &H0 ~ &HFFF(0 ~ 4095)

M\_XDevB: &H0 ~ &HFF8(0 ~ 4088)

M\_XDevW: &H0 ~ &HFF0(0 ~ 4080)

M\_XDevD: &H0 ~ &HFE0(0 ~ 4064)

注) 通过实数的数值变量指定时, 四舍五入后设定。

### 【例子】

|                             |                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------|
| 1 M1%=M_XDev(1)             | ' 在 M1 中代入 PLC 输入信号 1 号的值 (1 或 0)                                                    |
| 2 M2%=M_XDevB(&H10)         | ' 在 M2 中代入从 PLC 输入信号 10 号 (16 进制数) 开始 8 位量的值                                         |
| 3 M3%=M_XDevW(&H20) And &H7 | ' 在 M3 中代入从 PLC 输入信号 20 号 (16 进制数) 开始 3 位量的值                                         |
| 4 M4%=M_XDevW(&H20)         | ' 在 M4 中代入从 PLC 输入信号 20 号 (16 进制数) 开始 16 位量的值                                        |
| 5 M5&=M_XDevD(&H100)        | ' 在 M5 中代入从 PLC 输入信号 100 号 (16 进制数) 开始 32 位量的值                                       |
| 6 P1.Y=M_XDevD(&H100)/1000  | ' 以整数方式向位置变量 P1 的 Y 成分中输入从 PLC 输入信号 100 号 (16 进制数) 开始的 32 位量的数据, 乘以 1/1000 (实数化) 后代入 |

### 【说明】

- (1) 以整数返回 PLC 输入信号 (X) 的状态。
- (2) 从指定的 PLC 输入信号号码中, M\_XDev 返回 1 位、M\_XDevB 返回 8 位、M\_XDevW 返回 16 位、M\_XDevD 返回 32 位量的数据。
- (3) 指定 PLC 输入信号号码时, 请使参照的信号范围为以 16 进制表示时 &H0 ~ &HFFF (以 10 进制表示时 0 ~ 4095)。省略或超出范围时, 将会发生 L3110 (参数的值超出范围)。
- (4) 需要事先通过参数 QXYREAD 将输入信号的参照设为有效。
- (5) 对应的 PLC 单元未连接时, 将会返回 0。

【补充】

表 4-35: 可填入 < 数值变量 > 的变量

○: 可使用、×: 不可使用

| 位宽      | 数值变量的类型      |                 |                     |                     | 其他变量                         |                               |                |
|---------|--------------|-----------------|---------------------|---------------------|------------------------------|-------------------------------|----------------|
|         | 整数<br>例 )M1% | 长精度整数<br>例 )M1& | 单精度<br>实数<br>例 )M1! | 双精度<br>实数<br>例 )M1# | 位置 <sup>注1)</sup><br>例 )P1.X | 关节 <sup>注1)</sup><br>例 )J1.J1 | 字符串<br>例 )C1\$ |
| M_XDev  | ○            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_XDevB | ○            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_XDevW | ○            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_XDevD | ×            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |

注 1) 当变量所处理的数值是角度时 (位置变量的 A、B、C 成分数据和关节变量的所有成分数据), 以弧度单位的数值对现在的输出状态进行存储、运算等的处理将以弧度单位进行。在监视器等显示中, 将变换为度的单位显示。

表 4-36: 可填入 < PLC 输入信号号码 > 的常数 · 变量

○: 可使用、×: 不可使用

| 位宽      | 常数的类型                      |                    |                  | 数值变量的类型      |                 |                                    |                                    | 其他变量                                 |                                       |                |
|---------|----------------------------|--------------------|------------------|--------------|-----------------|------------------------------------|------------------------------------|--------------------------------------|---------------------------------------|----------------|
|         | 数值 <sup>注1)</sup><br>例 )12 | 2 进制数<br>例 )&B1100 | 16 进制数<br>例 )&HC | 整数<br>例 )M1% | 长精度整数<br>例 )M1& | 单精度<br>实数 <sup>注1)</sup><br>例 )M1! | 双精度<br>实数 <sup>注1)</sup><br>例 )M1# | 位置<br><sup>注1), 注2)</sup><br>例 )P1.X | 关节<br><sup>注1), 注2)</sup><br>例 )J1.J1 | 字符串<br>例 )C1\$ |
| M_XDev  | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                    | ×                                     | ×              |
| M_XDevB | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                    | ×                                     | ×              |
| M_XDevW | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                    | ×                                     | ×              |
| M_XDevD | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                    | ×                                     | ×              |

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因设定的值将以弧度单位的数值存储处理, 因此信号号码的指定将变得困难。(在监视器等的显示中, 将变换为度的单位显示, 因此会显示与设定值相同的值)

例) 即使为了指定输出信号 8 号而在 P1.A 中代入 8 (P1.A=8), 内部处理也会对 0.14 (rad) 进行四舍五入而变为 0, 因此, 将会变为指定输出信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

## M\_YDev/M\_YDevB/M\_YDevW/M\_YDevD

### 【功能】

通过 CR7xx-Q 系列的控制器向 PLC 输出信号 (Y) 写入或进行参照。(仅限 CR7xx-Q 系列控制器)  
(进行相应设定, 使机器人 CPU 对指定的输入输出单元 / 输入输出混合单元进行管理, 进行输出信号的直接参照或直接写入。(参照第 502 页的「5.25PLC 输入输出单元直接控制」))

M\_YDev: 以位为单位写入 / 参照。

M\_YDevB: 以字节为单位 (8 位量) 写入 / 参照。

M\_YDevW: 以字为单位 (16 位量) 写入 / 参照。

M\_YDevD: 以双字为单位 (32 位量) 写入 / 参照。

### 【格式】

#### 参照

<数值变量> = M\_YDev [( < PLC 输出信号号码 > )]

<数值变量> = M\_YDevB [( < PLC 输入信号号码 > )]

<数值变量> = M\_YDevW [( < PLC 输入信号号码 > )]

<数值变量> = M\_YDevD [( < PLC 输入信号号码 > )]

#### 写入

M\_YDev [( < PLC 输入信号号码 > )] = <数值>

M\_YDevB [( < PLC 输入信号号码 > )] = <数值>

M\_YDevW [( < PLC 输入信号号码 > )] = <数值>

M\_YDevD [( < PLC 输入信号号码 > )] = <数值>

### 【用语】

<数值变量>

指定要代入参照的输出信号的值的数值变量。

< PLC 输入信号号码 >

以常数或数值变量指定参照或写入的输出信号号码。设定范围以 16 进制表示为 &H0 ~ &HFFF (以 10 进制表示为 0 ~ 4095), 各状态变量分别如下所示。

M\_YDev: &H0 ~ &HFFF(0 ~ 4095)

M\_YDevB: &H0 ~ &HFF8(0 ~ 4088)

M\_YDevW: &H0 ~ &HFF0(0 ~ 4080)

M\_YDevD: &H0 ~ &HFE0(0 ~ 4064)

注) 通过实数的数值变量指定时, 四舍五入后设定。

<数值>

通过常数或变量指定要输出的数据。

各状态变量的有效范围如下所示:

M\_YDev: 0 或 1(&H0 ~ &H1)

M\_YDevB: -128 ~ 127(&H80 ~ &H7F)

M\_YDevW: -32768 ~ 32767(&H8000 ~ &H7FFF)

M\_YDevD: -2147483648 ~ 2147483647  
(&H80000000 ~ &H7FFFFFFF)

注) 通过实数指定时, 四舍五入后设定。

### 【例文】

1 M\_YDev(1)=1 ' 开启 PLC 输出信号 1 号

2 M\_YDevB(&H10)=&HFF ' 开启从 PLC 输出信号 10 号 (16 进制数) 开始的 8 位量

3 M\_YDevW(&H20)=&HFFFF ' 开启从 PLC 输出信号 20 号 (16 进制数) 开始的 16 位量

4 M\_YDevD(&H100)=P1.X \* 1000 ' 将位置变量 P1 的 X 坐标值乘以 1000 后存储到从 PLC 输出信号 100 号 (16 进制数) 开始的 32 位量中

5 M1%=M\_YDevW(&H20) And &H7' 在 M1 中代入从 PLC 输出信号 20 号 (16 进制数) 开始的 3 位的值

### 【说明】

- (1) 向 PLC 输出信号 (Y) 写入或进行参照。
- (2) 写入的数据及参照所返回的数据，都是整数值。
- (3) 从指定的 PLC 输出信号号码中，M\_Ydev 以 1 位、M\_YdevB 以 8 位、M\_YdevW 以 16 位、M\_YdevD 以 32 位量的数据为对象。
- (4) 指定 PLC 输出信号号码时，请使写入或参照的信号范围为以 16 进制表示时 &H0 ~ &HFFF（以 10 进制表示时 0 ~ 4095）。省略或超出范围时，将会发生 L3110（参数的值超出范围）。
- (5) 需要事先通过参数 QXYREAD 将输入信号的参照设为有效，或通过参数 QXYUNITn(n=1 ~ 4) 设定机器人 CPU 管理的输入输出单元。
- (6) 不支持配合使用 Diy 指令输出脉冲。使用 Diy 指令时，会出现 L4220（输入的指令语句的结构有误。）。
- (7) 写入时，如果未连接对应的 PLC 单元，则信号不会发生变化。  
参照时，对应的 PLC 单元未连接时，将会返回 0。

【补充】

表 4-37: 可填入 <数值变量> 的变量

○: 可使用、×: 不可使用

| 位宽      | 数值变量的类型      |                 |                     |                     | 其他变量                         |                               |                |
|---------|--------------|-----------------|---------------------|---------------------|------------------------------|-------------------------------|----------------|
|         | 整数<br>例 )M1% | 长精度整数<br>例 )M1& | 单精度<br>实数<br>例 )M1! | 双精度<br>实数<br>例 )M1# | 位置 <sup>注1)</sup><br>例 )P1.X | 关节 <sup>注1)</sup><br>例 )J1.J1 | 字符串<br>例 )C1\$ |
| M_YDev  | ○            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_YDevB | ○            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_YDevW | ○            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |
| M_YDevD | ×            | ○               | ○                   | ○                   | ○                            | ○                             | ×              |

注 1) 当变量所处理的数值是角度时（位置变量的 A、B、C 成分数据和关节变量的所有成分数据），以弧度单位的数值对现在的输出状态进行存储，运算等的处理将以弧度单位进行。在监视器等显示中，将变换为度的单位显示。

表 4-38: 可填入 <PLC 输入信号号码> 的常数・变量

○: 可使用、×: 不可使用

| 位宽      | 常数的类型                      |                    |                  | 数值变量的类型      |                 |                                    |                                    | 其他变量                                             |                                                   |                |
|---------|----------------------------|--------------------|------------------|--------------|-----------------|------------------------------------|------------------------------------|--------------------------------------------------|---------------------------------------------------|----------------|
|         | 数值 <sup>注1)</sup><br>例 )12 | 2 进制数<br>例 )&B1100 | 16 进制数<br>例 )&HC | 整数<br>例 )M1% | 长精度整数<br>例 )M1& | 单精度<br>实数 <sup>注1)</sup><br>例 )M1! | 双精度<br>实数 <sup>注1)</sup><br>例 )M1# | 位置<br><sup>注1)</sup> 、 <sup>注2)</sup><br>例 )P1.X | 关节<br><sup>注1)</sup> 、 <sup>注2)</sup><br>例 )J1.J1 | 字符串<br>例 )C1\$ |
| M_YDev  | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                                | ×                                                 | ×              |
| M_YDevB | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                                | ×                                                 | ×              |
| M_YDevW | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                                | ×                                                 | ×              |
| M_YDevD | ○                          | ○                  | ○                | ○            | ○               | ○                                  | ○                                  | ○                                                | ×                                                 | ×              |

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时（位置变量的成分数据 A、B、C 和关节变量的所有成分数据），因设定的值将以弧度单位的数值存储处理，因此信号号码的指定将变得困难。（在监视器等显示中，将变换为度的单位显示，因此会显示与设定值相同的值）

例) 即使为了指定输出信号 8 号而在 P1.A 中代入 8 (P1.A=8)，内部处理也会对 0.14 (rad) 进行四舍五入而变为 0，因此，将会变为指定输出信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm，因此不会出现这种状态。

表 4-39: 可填入 &lt; 数值 &gt; 的常数・变量

○: 可使用、×: 不可使用

| 位宽      | 常数的类型          |                   |                 | 数值变量的类型     |                |                    |                    | 其他变量         |               |               |
|---------|----------------|-------------------|-----------------|-------------|----------------|--------------------|--------------------|--------------|---------------|---------------|
|         | 数值 注1)<br>例)12 | 2 进制数<br>例)&B1100 | 16 进制数<br>例)&HC | 整数<br>例)M1% | 长精度整数<br>例)M1& | 单精度<br>实数<br>例)M1! | 双精度<br>实数<br>例)M1# | 位置<br>例)P1.X | 关节<br>例)J1.J1 | 字符串<br>例)C1\$ |
| M_YDev  | ○              | ○                 | ○               | ○           | ×              | ×                  | ×                  | ×            | ×             | ×             |
| M_YDevB | ○              | ○                 | ○               | ○           | ×              | ×                  | ×                  | ×            | ×             | ×             |
| M_YDevW | ○              | ○                 | ○               | ○           | ×              | ×                  | ×                  | ×            | ×             | ×             |
| M_YDevD | ○注2)           | ○注2)              | ○注2)            | ○注2)        | ○              | ○                  | ○注3)               | ○            | ○             | ×             |

注 1) 实数值将四舍五入。

注 2)  **注意**

请注意: 指定的常数为容纳 2 进制数的 16 位中的数值 (-32768 ~ +32767) 时, 位 15 (第 16 位) 开启的数值作为负的数值处理, 高位 16 位被符号扩展, 输出全部 ON 状态的信号。

例) 指定 -32768 (&B1000000000000000) 后, 会输出  
&B11111111111111111000000000000000 (2 进制数)

[ 对策 ]

如下所示, 在将常数代入长精度整数型的数值变量的基础上, 再代入本状态变量 M\_YDevD 后, 可输出 &B00000000000000001000000000000000 (2 进制数)。

1 M1&=32768

2 M\_YDevD(&H20)=M1&

注 3) 可输出数值的范围为 -2147483648 ~ 2147483647。

## P Base/P NBase

### 【功能】

返还基本数据相关情报。

P\_Base: 返还现在被设定的 BASE 变换数据。

P\_NBase: 返还初始值 (0,0,0,0,0,0)(0,0)。

### 【格式】

例) <位置变量> = P\_Base [ (<机器号码> ) ]

例) <位置变量> = P\_NBase

### 【用语】

<位置变量> 指定代入位置变量。

<机器号码> 输入机器号码。1 ~ 3, 省略时为 1。

### 【例子】

1 P1=P\_Base ' 在 P1 输入现在被设定的基本变换数据。

2 Base P\_NBase ' 将基本变换数据返回为初始值。

### 【说明】

(1) 在 P\_NBase 输入 (0,0,0,0,0,0)(0,0)。

(2) 基本变换会对示教数据产生影响, 因此请慎重使用。

(3) 变更基本位置的情况下, 请使用 Base 指令。

(4) 读取专用。

## P\_CavDir

### 【功能】

返回通过干涉确认检测到干涉时的机器人的动作方向。  
本功能对对应机型有所限定。关于详细内容，请参照「[5.24 关于干涉回避功能](#)」。

### 【格式】

例) <位置变量> = P\_CavDir [( <机器号码> )]

### 【用语】

<位置变量>            指定要代入的位置变量。  
<机器号码>            填入机器号码。1 ~ 3、省略时为1。

### 【例文】

参照第 500 页的「[5.24.10 采样程序](#)」。

### 【说明】

- (1) 在干涉确认后的自动恢复动作中确认机器人的动作方向时使用。
- (2) 将检测到干涉的瞬间的机器人的动作方向，以最大移动轴的值  $\pm 1.0$  的比例表示。  
例) (X轴方向: Y轴方向) = (2: -1) 的比例动作时 ..... P\_CavDir=(1,-0.5,0,0,0,0)(0,0)
- (3) 姿势轴和构造标志始终为 (\*,\*,\*,0,0,0,0) (0,0)。( \* 为任意值)
- (4) 检测到干涉时计算出值，保持该值直到下次检测到干涉。
- (5) 机器人静止时，通过其他机器人的移动检测到了干涉时，全轴为 0.0。
- (6) 本变量是以动作指令的目标位置为基础计算动作方向的，因此当在目标位置附近发生了冲突时，所有要素将可能为 0.0。
- (7) 读取专用。
- (8) 禁止使用干涉确认的机器人，始终返回全轴 0.0。
- (9) 可读出的值的单位与 P\_ColDir 通用。

## P\_ColDir

### 【功能】

返还检测到冲突时的机器人的动作方向。

### 【格式】

例) <位置变量> = P\_ColDir [ (<机器号码> ) ]

### 【用语】

<位置变量>      指定代入的位置变量。  
<机器号码>      输入机器号码。 1 ~ 3、省略时为 1。

### 【例子】

参照 ColChk 的例子

### 【说明】

- (1) 以冲突检知后的自动回归动作，确认机器人的动作方向时使用。
- (2) 将检测到冲突的瞬间机器人的动作方向以最大移动轴的值  $\pm 1.0$  的比例表示。例：想要以 (X 轴方向 : Y 轴方向) = (2: -1) 的比例动作时 ... P\_ColDir=(1,-0.5,0,0,0,0)(0,0)
- (3) 姿势轴和构造标志通常为 (\*,\*,\*,0,0,0,0,0)(0,0)。 ("\*" 为任意的值)
- (4) 算出检测到冲突时的值以及保持此值直到下一次检测到冲突为止。
- (5) 在机器人静止时，检测到从外部被物品碰撞到的情况下，全轴会变成 0.0。
- (6) 本变量将动作命令的目的位置作为动作方向的基础计算，因此在目的位置的附近发生冲突的情况下，全要素会变成 0.0。
- (7) 读取专用。
- (8) 在禁止使用冲突检知的机器人上，全轴会一直返还为 0.0。

### 【相关指令】

ColChk (Col Check)、 ColLvl (ColLevel)

### 【关连系统状态变量】

M\_ColSts、 J\_ColMxl

## P\_CordR

### 【功能】

返回从通用坐标到本机器人基本坐标为止的位置。

本功能对对应机型有所限定。关于详细内容，请参照「[5.24 关于干涉回避功能](#)」。

### 【格式】

例) <位置变量> = P\_CordR [ (<机器号码> ) ]

### 【用语】

<位置变量>            指定要代入的位置变量。  
<机器号码>            填入机器号码。1 ~ 3、省略时为1。

### 【例文】

参照第 500 页的「[5.24.10 采样程序](#)」。

### 【说明】

- (1) 参照从通用坐标到本机器人的基本坐标为止的位置。(参数 RBCORD 的设定值)
- (2) 参数 RBCORD 为初始值 (0,0,0,0,0) 时，所有坐标的读取值为 0。
- (3) 读取专用。
- (4) 使用者机器始终返回 "0"。

## P\_Curr

### 【功能】

返还现在位置 (X,Y,Z,A,B,C,L1,L2)(FL1,FL2)。

### 【格式】

例) <位置变量> = P\_Curr [ (<机器号码>) ]

### 【用语】

<位置变量> 指定代入的位置变量。  
<机器号码> 输入机器号码。 1 ~ 3、省略时为 1。

### 【例子】

```

1 Def Act 1,M_In(10)=1 GoTo *LACT ' 定义插入
2 Act 1=1 ' 插入有效
3 Mov P1
4 Mov P2
5 Act 1=0 ' 插入无效

100 *LACT
101 P100=P_Curr ' 在插入信号输入时点读取现在位置。
102 Mov P100,-100 ' 在 P100 的 100mm 上空 (在 TOOL 的 Z 方向 -100mm) 移动
103 End ' 程序结束

```

### 【说明】

- (1) 想知道现在位置时使用。
- (2) 读取专用。

### 【关连系统状态变量】

J\_Curr、 P\_Fbc

## P\_CurrR

### 【功能】

返回从通用坐标观察到的本机器人的现在位置。

本功能对对应机型有所限定。关于详细内容，请参照 [「5.24 关于干涉回避功能」](#)。

### 【格式】

|                                   |
|-----------------------------------|
| 例) <位置变量> = P_CurrR [ (<机器号码> ) ] |
|-----------------------------------|

### 【用語】

|        |                      |
|--------|----------------------|
| <位置变量> | 指定要代入的位置变量。          |
| <机器号码> | 填入机器号码。1 ~ 3、省略时为 1。 |

### 【例文】

参照第 500 页的 [「5.24.10 采样程序」](#)。

### 【说明】

- (1) 参照从通用坐标观察到的本机器人的现在位置。  
返回通过参数 RBCORD 对 P\_Curr 进行了变换的值。
- (2) 读取专用。
- (3) 使用者机器始终返回 "0"。

## P\_ECord

**【功能】**

在 Ex-T 控制 /Ex-T 样条插补指令执行中，返回所使用的 Ex-T 坐标系原点数据。

**【格式】**

例) <位置变量> = P\_ECord [ (<机械号码>)]

**【用语】**

- <位置变量>            指定要代入的位置变量。
- <机械号码>            设定正在执行 Ex-T 控制 /Ex-T 样条插补的机械号码。  
                           设定范围：1 ~ 3  
                           省略时：1  
                           指定了不存在的机械号码的情况下，执行时会发生 L3870（指定的机械号码为无效的值）错误。

**【例文】**

1 P1 = P\_ECord            ' 将当前正在进行插补的 Ex-T 坐标系原点数据代入至位置变量 P1

**【说明】**

- (1) 在 Ex-T 控制 /Ex-T 样条插补执行中，返回最后使用的 Ex-T 坐标系原点数据。
- (2) 通过多任务中浏览 P\_ECord 的值，可以根据插补的进展，执行运算处理及信号输出等处理。
- (3) P\_ECord 根据状况返回下表所示的值。

表 4-40: P\_ECord 返回的值

| 状况                        | P_ECord 返回的值                   |
|---------------------------|--------------------------------|
| 电源接通后                     | (0.0,0.0,0.0,0.0,0.0,0.0)(0,0) |
| Ex-T 控制 /Ex-T 样条插补执行中     | 插补中所使用的 Ex-T 坐标系原点数据           |
| Ex-T 控制 /Ex-T 样条插补结束后     | 最后使用的 Ex-T 坐标系原点数据             |
| 主程序的 End 指令执行后            | (0.0,0.0,0.0,0.0,0.0,0.0)(0,0) |
| 程序复位操作后                   | (0.0,0.0,0.0,0.0,0.0,0.0)(0,0) |
| 不支持 Ex-T 控制 /Ex-T 样条插补的机械 | (0.0,0.0,0.0,0.0,0.0,0.0)(0,0) |

(4) 读取专用。

**【相关指令】**

EMvc (E Move C)、EMvr (E Move R)、EMvr2 (E Move R2)、EMvr3 (E Move R3)、EMvs (E Move S)、EMvSpl (E Move Spline)

## P\_Fbc

### 【功能】

将从伺服来的反馈值返还到原来的现在位置 (X,Y,Z,A,B,C,L1,L2)(FL1,FL2)。

### 【格式】

|                                 |
|---------------------------------|
| 例) <位置变量> = P_Fbc [ (<机器号码> ) ] |
|---------------------------------|

### 【用语】

|        |                       |
|--------|-----------------------|
| <位置变量> | 指定代入的位置变量。            |
| <机器号码> | 输入机器号码。 1 ~ 3、省略时为 1。 |

### 【文例】

1 P1=P\_Fbc                    ' 在 P1 代入反馈现在位置。

### 【说明】

- (1) 将从伺服来的反馈值返还到原来的现在位置。
- (2) 读取专用。

### 【相关指令】

[Torq \(Torque\)](#)

### 【关连系统状态变量】

[P\\_Curr](#)、[J\\_Fbc/J\\_AmpFbc](#)、[M\\_Fbd](#)

## P\_Safe

**【功能】**

返回待避点位置（参数 JSAFE 的直交位置）。

**【格式】**

例) <位置变量> = P\_Safe [ (<机器号码> ) ]

**【用语】**

<位置变量>      指定代入的位置变量。  
<机器号码>      输入机器号码。 1 ~ 3、省略时为 1。

**【例子】**

1 P1=P\_Safe                    ' 在 P1 代入被设定的待避点位置。

**【说明】**

- (1) 返还用参数 JSAFE 登录的关节位置转变而来的直交位置。
- (2) 读取专用。

## P\_Tool/P\_NTool

**【功能】**

返回 TOOL 变换数据。  
P\_Tool: 返回现在被设定的 TOOL 变换数据。  
P\_NTool: 返回初始值 (0,0,0,0,0,0,0,0)(0,0)。

**【格式】**

例) <位置变量> = P\_Tool [ (<机器号码> ) ]  
例) <位置变量> = P\_NTool

**【用语】**

<位置变量>      指定代入的位置变量。  
<机器号码>      输入机器号码。 1 ~ 3、省略时为 1。

**【例子】**

1 P1=P\_Tool                    ' 在 P1 代入 TOOL 变换数据。

**【说明】**

- (1) P\_Tool 使用 Tool 指令或是参数的 MEXTL 返回 TOOL 变换数据。
- (2) 变更 TOOL 数据的时候请使用 Tool 指令。
- (3) 读取专用。

## P\_UDev

### 【功能】

通过 CR7xx-Q 系列的控制器，向多 CPU 间共有内存写入位置数据或进行参照。（仅限 CR7xx-Q 系列控制器）  
以位置数据单位（32 位 x10 量）写入 / 参照。

### 【格式】

参照  
<位置变量> = P\_UDev (<起始输入输出号码>, <共有内存地址>)

写入  
P\_UDev (<起始输入输出号码>, <共有内存地址>) = <位置数据>

### 【用语】

<位置变量> 指定要代入的位置变量。

<起始输入输出号码> 通过常数或数值变量指定 CPU 单元的输入输出号码。  
(指定的值为以 16 进制表示起始输入输出号码、且省略低位 1 位的值。)  
范围：16 进制表示时 &H3E0 ~ &H3E3 (10 进制表示时 992 ~ 995)  
1 号机：&H3E0(10 进制表示时为 992)  
2 号机：&H3E1(10 进制表示时为 993)  
3 号机：&H3E2(10 进制表示时为 994)  
4 号机：&H3E3(10 进制表示时为 995)  
注) 数据的写入仅限本号机有效。

<共有内存地址> 通过常数或数值变量指定 CPU 单元的共有内存地址。  
有效范围为 10000 ~ 24312 (10 进制数)。

<位置数据> 指定要写入的位置数据。  
可通过常数、变量、逻辑 / 算术式及函数指定。

### 【例子】

1 P\_UDev(&H3E1, 10010)=P\_Curr' 向 2 号机 CPU 的共有内存地址 10010 中写入现在位置。  
2 P1=P\_UDev(&H3E2, 10001) ' 读取 3 号机 CPU 的共有内存地址 10001 的位置数据，将其代入位置变量 P1 中。

### 【说明】

- (1) 向 PLC 的多 CPU 间共有内存写入位置数据或进行参照。  
写入的数据及参照所返回的数据，都是位置数据。
- (2) 通过起始输入输出号码和共有内存地址指定对象共有内存。
- (3) 以从指定的共有内存地址开始 20 字 (32 位 x10) 量的数据为对象。
- (4) 在进行指定时，为以 16 进制表示时为起始输入输出号码 &H3E0 ~ &H3E3 (10 进制表示时为 992 ~ 995)；以 10 进制表示时为共有内存地址的写入或参照地址 10000 ~ 24316。
- (5) 仅本号机 CPU 的共有内存地址可写入。即使指定其他号机 CPU 的地址写入数据，也将不会更新。
- (6) 参照时，对应的共有内存地址未通过多 CPU 高速通信区域设定 (参数: QMLTCPUn) 进行分配时，返回 0。

### 【补充】

表 4-41: 可填入 <PLC 输入输出信号号码><共有内存地址> 的常数・变量

| 变量的类型 | 常数的类型                     |                   |                 | 数值变量的类型     |                |                                   |                                   | 其他变量                                            |                                                  |               |
|-------|---------------------------|-------------------|-----------------|-------------|----------------|-----------------------------------|-----------------------------------|-------------------------------------------------|--------------------------------------------------|---------------|
|       | 数值 <sup>注1)</sup><br>例)12 | 2 进制数<br>例)&B1100 | 16 进制数<br>例)&HC | 整数<br>例)M1% | 长精度整数<br>例)M1& | 单精度<br>实数 <sup>注1)</sup><br>例)M1! | 双精度<br>实数 <sup>注1)</sup><br>例)M1# | 位置<br><sup>注1)</sup> , <sup>注2)</sup><br>例)P1.X | 关节<br><sup>注1)</sup> , <sup>注2)</sup><br>例)J1.J1 | 字符串<br>例)C1\$ |
| 可否使用  | ○                         | ○                 | ○               | ○           | ○              | ○                                 | ○                                 | ○                                               | ×                                                | ×             |

○: 可使用、×: 不可使用

注 1) 实数值将四舍五入。

注 2) 当变量所处理的数值是角度时 (位置变量的成分数据 A、B、C 和关节变量的所有成分数据), 因设定的值将以弧度单位的数值存储处理, 因此信号号码的指定将变得困难。(在监视器等显示中, 将变换为度的单位显示, 因此会显示与设定值相同的值)

例) 即使为了指定输出信号 8 号而在 P1.A 中代入 8 (P1.A=8) 时, 内部处理也会对 0.14 (rad) 进行四舍五入而变为 0, 因此, 将会变为指定输出信号 0 号。位置变量的成分数据 X、Y、Z 的单位为 mm, 因此不会出现这种状态。

**【参考】关于号机间数据的数据保证**

根据本号机的读取和其他号机的数据写入 / 接收其他号机信息的时机, 各号机的数据中可能会出现旧数据和新数据同时存在 (数据的分离) 的问题。

在通过使用 CPU 共有内存的程序进行通信中, 防止数据分离的方法如图 4-35 所示。

1) 防止 32 位数据的分离

通过将偶数地址 (例: 地址 10002) 置于起始处, 对 CPU 共有内存的使用者自由区域进行访问, 可防止 32 位数据的分离。

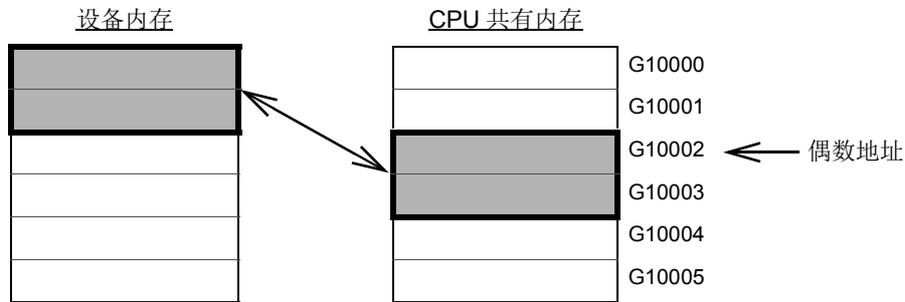


图 4-35: 防止 32 位数据的分离

2) 防止超过 32 位的数据的分离

通过程序进行的读取从使用者自由区域的起始处开始按顺序读取。

此外, 通过写入指令, 从使用者自由区域的结束地址向起始地址写入发送数据。

因此, 在要进行通信的数据的起始处设置用于互锁的设备, 可防止进行通信的数据的分离。

## P\_WkCord

### 【功能】

进行现在设定的工件坐标数据的参照及新工件坐标的设定。

参照、设定的对象为参数：WK1CORD ~ WK8CORD。

(关于功能概要请参照第 652 页的「7.5Ex-T 制御について」。)

### 【格式】

|                                 |      |
|---------------------------------|------|
| 例) <位置变量> = P_WkCord (<工件坐标号码>) | ' 参照 |
| P_WkCord (<工件坐标号码>) = <工件坐标数据>  | ' 设定 |

### 【用语】

|          |                                                                           |
|----------|---------------------------------------------------------------------------|
| <位置变量>   | 指定要代入的位置变量。                                                               |
| <工件坐标号码> | 指定 1 ~ 8 的工件坐标号码。<br>可使用常数、变量、逻辑 / 算术式、函数。<br>指定了实数 / 双精度实数时，小数点以后进行四舍五入。 |
| <工件坐标数据> | 以位置常数或位置变量指定工件坐标数据。<br>设定的值 (坐标值) 为从基本坐标系观察到的工件坐标系原点的位置数据。                |

### 【例子】

①在已经设定的工件坐标 1 的基础上新设定工件坐标 2，并将其作为世界坐标系的示例

|                  |                                      |
|------------------|--------------------------------------|
| 1 PW=P_WkCord(1) | ' 读取工件坐标 1 (参数: WK1CORD 的设定值) 代入 PW。 |
| 2 PW.X=PW.X+100  | ' 在读取的 X 坐标值上加上 100。                 |
| 3 PW.Y=PW.Y+100  | ' 在读取的 Y 坐标值上加上 100。                 |
| 4 P_WkCord(2)=PW | ' 将上述运算结果设定到工件坐标 2。(设定到参数: WK2CORD)  |
| 5 Base 2         | ' 将工件坐标 2 作为新的世界坐标系。                 |
| 6 Mov P1         |                                      |

②将 PA 设定为工件坐标 4，并沿着工件坐标 4 以 EX/T 控制直线插补动作的示例

|                  |                                   |
|------------------|-----------------------------------|
| 1 P_WkCord(4)=PA | ' 将 PA 设为工件坐标 4。(参数: 设定为 WK4CORD) |
| 2 EMvs 4,P1      | ' 沿着工件坐标 4，以 Ex-T 控制直线插补向 P1 移动。  |

### 【说明】

- 指定工件坐标号码，读取现在设定的工件坐标值，或进行设定。要指定的工件坐标号码 1 ~ 8 与参数：WK1CORD ~ WK8CORD 对应。
- 工件坐标数据的 X、Y、Z 成分表示从基本坐标系的原点开始到工件坐标系的原点为止的平行移动量。此外，位置数据 A、B、C 成分表示工件坐标系相对于机器人坐标系倾斜了多少。
  - X ... 往 X 轴方向平行移动的距离
  - Y ... 往 Y 轴方向平行移动的距离
  - Z ... 往 Z 轴方向平行移动的距离
  - A ... 围绕 X 轴的旋转角度
  - B ... 围绕 Y 轴的旋转角度
  - C ... 围绕 Z 轴的旋转角度
 从工件坐标系原点往正方向看，顺时针为正的旋转方向来获取 A、B、C 成分。
- 构造标志内容没有意义。
- 通过本指令设定工件坐标后，将清除对应的工件坐标号码的 WO、WX、WY 数据 (示教工件坐标的 3 点的坐标值。参数: WKnWO, WKnWX, WKnWY(n:1 ~ 8))。
 

例) 执行例文①的第 4 步 (P\_WkCord(2)=PW) 后，值 "0" 将被设定到 WK2WO、WK2WX、WK2WY 中。
- 仅 MELFA-BASIC V 可以使用。
- Ex-T 控制插补指令，是根据已指定的工件坐标号码，参照该变量并作为 Ex-T 控制点。因此，对 P\_WkCord 的设定也是对 Ex-T 控制中的控制点坐标的设定。

【相关指令】

Base、EMvc (E Move C)、EMvr (E Move R)、EMvr2 (E Move R2)、EMvr3 (E Move R3)、EMvs (E Move S)

【相关参数】

MEXBSNO、WKncORD(n:1 ~ 8)、WKnWO, WKnWX, WKnWY(n:1 ~ 8)

## P\_Zero

【功能】

通常返回 (0,0,0,0,0,0,0,0)(0,0)。

【格式】

|                    |
|--------------------|
| 例) <位置变量> = P_Zero |
|--------------------|

【用语】

<位置变量>      指定代入的位置变量。

【例子】

1 P1=P\_Zero      ' 在 P1 代入 (0,0,0,0,0,0,0,0)(0,0)。

【说明】

- (1) 想将 P 变量初始化时使用。
- (2) 读取专用。

## 4.15 函数的详细说明

### 4.15.1 记载项目的说明

- 【功能】 : 表示函数的功能。
- 【格式】 : 表示函数的自变量输入方法。
- 【例子】 : 表示使用函数的程序范例。
- 【用语】 : 表示自变量的意义、范围等。
- 【说明】 : 表示详细的功能及注意事项。
- 【参照】 : 表示相关的函数。
- 【相关指令】 : 表示相关的指令。
- 【关连参数】 : 表示相关的参数。

### 4.15.2 各函数的说明

以下各函数的说明以字母顺序表示。

## Abs

**【功能】**

返回被给予值的绝对值。

**【格式】**

|                     |
|---------------------|
| <数值变量> = Abs (<数式>) |
|---------------------|

**【例子】**

1 P2.C=Abs(P1.C)        ' 在 P2.C 代入取得 P1.C 符号的值。  
2 Mov P2  
3 M2=-100  
4 M1=Abs(M2)        ' 在 M1 代入 100。

**【说明】**

(1) 返回被给予值的绝对值（符号为正的数）。

**【参照】**

[Sgn](#)

## ACos

### 【功能】

从指定的余弦值（数值）返回反余弦值（Arc Cosine）。

### 【格式】

|                       |
|-----------------------|
| < 数值变量 >=ACos(< 算式 >) |
|-----------------------|

### 【用语】

|          |                                                               |
|----------|---------------------------------------------------------------|
| < 数值变量 > | 计算并返回指定算式的值的反余弦 (Arc Cosine) 值。<br>输出值的单位为弧度。<br>范围：0 ~ $\pi$ |
| < 算式 >   | 指定余弦值。<br>设定范围：-1.0 ~ +1.0                                    |

### 【例文】

1 MRad=ACos(0.6)        ' 将 0.6 的反余弦值 (0.927295218001612 rad) 代入至 MRad。

### 【说明】

(1) 返回指定 < 算式 > 的反余弦值。单位为弧度。

## Align

**【功能】**

将位置的姿势轴（A，B，C轴）变换到最近的直交姿势（0，±90，±180）。

Align 只执行数值的输出，实际的动作请使用 Mov 等的动作指令。

**【格式】**

```
<位置变量> = Align (<位置>)
```

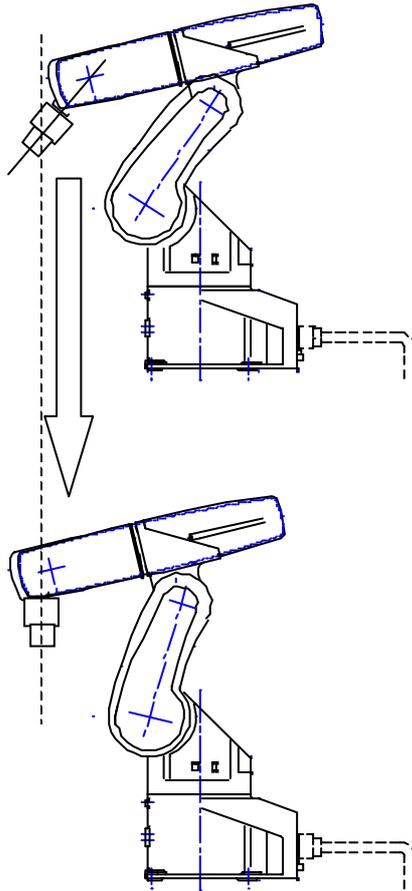
**【例子】**

```
1 P1=P_Curr
2 P2=Align(P1)
3 Mov P2
```

**【说明】**

- (1) 将位置的姿势轴 A，B，C 轴变换到最近的直交姿势（0，±90，±180）。
- (2) 因为返回值为位置数据，因此左边为关节变量的情况下，会发生报警。
- (3) 本功能无法使用在垂直多关节 5 轴机器人。

下列显示范例为 B 轴的例子。



## Asc

### 【功能】

返回字符串的最初文字的字符码 (character code)。

### 【格式】

|                      |
|----------------------|
| <数值变量> = Asc (<字符串>) |
|----------------------|

### 【例子】

1 M1=Asc("A")                    ' 在 M1 代入 &H41。

### 【说明】

- (1) 返回字符串的最初文字的字符码 (character code)。
- (2) 字符串为空字符串的情况下会发生报警。

### 【参照】

[Chr\\$](#)、[Val](#)、[Cvi](#)、[Cvs](#)、[Cvd](#)



## **Bin\$**

### **【功能】**

将数值变换为 2 进制字符串。

### **【格式】**

|                        |
|------------------------|
| <字符串变量> = Bin\$ (<数式>) |
|------------------------|

### **【例子】**

1 M1=&B11111111

2 C1\$=Bin\$(M1)            ' 在 C1\$ 代入 "11111111" 的字符串。

### **【说明】**

(1) 将数式的值变换为 2 进制字符串。

(2) 数式不为整数的情况下，将小数点四舍五入后的整数值变换成 2 进制数字字符串。

(3) Val 为执行与此动作相反的指令。

### **【参照】**

[Hex\\$](#)、[Str\\$](#)、[Val](#)

## CalArc

### 【功能】

提供以指定的 3 点所构成圆弧的情报。

### 【格式】

```
<<数值变量 4 > = CalArc (<位置 1 >, <位置 2 >, <位置 3 >,
 <数值变量 1 >, <数值变量 2 >, <数值变量 3 >, <位置变量 1 >)
```

### 【用语】

- <位置 1 >      指定圆弧的起点。
- <位置 2 >      指定圆弧的通过点。和 MVR 指令的 3 点相同意义。
- <位置 3 >      指定圆弧的终点。
- <数值变量 1 >    计算并返回指定的圆弧半径。(mm)
- <数值变量 2 >    计算并返回指定的圆弧中心角。(rad)
- <数值变量 3 >    计算并返回指定的圆弧长。(mm)
- <位置变量 1 >    计算并返回指定的圆弧的中心坐标。(位置型、ABC 为 0)
- <数值变量 4 >    返回值。
  - 1: 可正常的计算时。
  - 1: 位置 1、位置 2、位置 3 的其中 2 点为同一点、3 点为一直在线的情况。
  - 2: 3 点几乎为同一点的情况。

### 【例子】

```
1 M1=CalArc(P1,P2,P3,M10,M20,M30,P10)
2 If M1<>1 Then End ' 异通常结束
3 MR=M10 ' 半径
4 MRD=M20 ' 圆弧角度
5 MARCLN=M30 ' 圆弧长
6 PC=P10 ' 中心点坐标值
```

### 【说明】

- (1) 提供从指定的位置 1、位置 2、位置 3 的 3 点所生成的圆弧的信息。
- (2) 圆弧生成成功，各数据在被计算的情况下，返回值会变成 1。
- (3) 如果为同一点且 3 点在一直上排列的情况下，将 -1 作为返回值返回。在此情况将返回圆弧长从起点到终点的直线长度，半径为 -1、中心角为 0、中心点为 (0,0,0)。
- (4) 圆弧生成失败的话，将 -2 作为返回值返回。圆弧无法生成的情况下，返回半径为 -1、中心角为 0、圆弧长为 0、中心点为 (0,0,0)。
- (5) 在<位置 1 >，<位置 2 >，<位置 3 >，<数值变量 1 >，<数值变量 2 >，<数值变量 3 >，<位置变量 1 >无法记述有自变量的函数。执行时会发生报警。

## Chr\$

### 【功能】

返回拥有指常数式的值的字符码（character code）文字。

### 【格式】

```
<字符串变量> = Chr$ (<数式>)
```

### 【例子】

```
1 M1=&H40
2 C1$=Chr$(M1+1) ' 在 C1$ 代入 "A"。
```

### 【说明】

- (1) 将拥有数式的值的字符码文字返回。
- (2) 数式的值不是整数的情况下，返回将小数点部份四舍五入后的值的字符码。

### 【参照】

[Asc](#)

## CInt

### 【功能】

将数式的小数点部份四舍五入，变换为整数。

### 【格式】

```
<数值变量> = CInt (<数式>)
```

### 【例子】

```
1 M1=CInt(1.5) ' 在 M1 代入 2。
2 M2=CInt(1.4) ' 在 M2 代入 1。
3 M3=CInt(-1.4) ' 在 M3 代入 -1。
4 M4=CInt(-1.5) ' 在 M4 代入 -2。
```

### 【说明】

- (1) 将数式的值的小数部位四舍五入的值返回。

### 【参照】

[Int](#)、[Fix](#)

## CkSum

### 【功能】

计算字符的校检。

### 【格式】

$$\langle \text{数值变量} \rangle = *CkSum (\langle \text{字符串} \rangle, \langle \text{数式 1} \rangle, \langle \text{数式 2} \rangle)$$

### 【用语】

$\langle \text{字符串} \rangle$             指定计算和校检的字符串。  
 $\langle \text{数式 1} \rangle$             指定计算和校检的开始文字位置。  
 $\langle \text{数式 2} \rangle$             指定计算和校检的结束文字位置。

### 【例子】

1 M1=CkSum("ABCDEFGH",1,3) ' 在 M1 代入 &H41("A")+&H42("B")+&H43("C")=&HC6。

### 【说明】

- (1) 从字符串中的开始位置到结束位置，加算各文字的文字码，返还 0 ~ 255 的值。
- (2) 开始位置在字符串外的范围的情况下，会发生报警。
- (3) 结束位置超过字符串的范围外的情况下，求出从开始位置到字符串的最后的文字的和校检。
- (4) 加算结果超过 255 的情况下，会退缩到变成 255 或比 255 小的值。
- (5) 在  $\langle \text{字符串 1} \rangle$ ， $\langle \text{数式 1} \rangle$ ， $\langle \text{数式 2} \rangle$  无法记述有自变量的函数。执行时会发生报警。

## Cos

### 【功能】

给予余弦 (Cosine)。

### 【格式】

$$\langle \text{数值变量} \rangle = Cos (\langle \text{数式} \rangle)$$

### 【例子】

1 M1=Cos(Rad(60))

### 【说明】

- (1) 计算数式值的余弦 (Cosine) 的值。
- (2) 自变量的范围会变成数值的值可以取得的全部范围。
- (3) 返还值的范围变成 -1 ~ 1。
- (4) 自变量的单位为弧度。

### 【参照】

Sin、Tan、Atn/Atn2

## Cvi

### 【功能】

将字符串的最初 2 个文字的各个文字码视为数值转变为整数。

### 【格式】

<数值变量> = Cvi (<字符串式>)

### 【例子】

1 M1=Cvi("10ABC")                    ' 在 M1 代入 &H3031。

### 【说明】

- (1) 将字符串的最初 2 个文字的各个文字码视为数值转换为整数。
- (2) 字符串在 1 个文字以下时会发生报警。
- (3) 将数值变换为字符串时，请使用 MKI\$。
- (4) 在外部机构做数据输入输出等的数值数据传送时，为减少数据量时使用。

### 【参照】

Asc、Cvs、Cvd、Mki\$、Mks\$、Mkd\$

## Cvs

### 【功能】

将字符串的最初 4 个文字的各个文字码转换为单精度实数。

### 【格式】

<数值变量> = Cvs (<字符串式>)

### 【例子】

1 M1=Cvs("FFFF")                    ' 在 M1 代入 12689.6。

### 【说明】

- (1) 将字符串的最初 4 个文字的各个文字码转换为单精度实数。
- (2) 字符串在 3 个文字以下时会发生报警。
- (3) 将数值变换为字符串时，请使用 Mks\$。

### 【参照】

Asc、Cvi、Cvd、Mki\$、Mks\$、Mkd\$

## Cvd

**【功能】**

将字符串的最初 8 个文字的各个文字码转换为双精度实数。

**【格式】**

|                       |
|-----------------------|
| <数值变量> = Cvd (<字符串式>) |
|-----------------------|

**【例子】**

1 M1=Cvd("FFFFFFFF") ' 在 M1 代入 +3.52954E+30。

**【说明】**

- (1) 将字符串的最初 8 个文字的各个文字码转换为双精度实数。
- (2) 字符串在 7 个文字以下时会发生报警。
- (3) 将数值变换为字符串时，请使用 MKD\$。

**【参照】**

[Asc](#)、[Cvi](#)、[Cvs](#)、[Mki\\$](#)、[Mks\\$](#)、[Mkd\\$](#)

## Deg

**【功能】**

将角度单位从弧度 (rad) 变换为度 (deg)。

**【格式】**

|                     |
|---------------------|
| <数值变量> = Deg (<数式>) |
|---------------------|

**【例子】**

```
1 P1=P_Curr
2 If Deg(P1.C) < 170 Or Deg(P1.C) > -150 Then *NOErr1
3 Error 9100
4 *NOErr1
```

**【说明】**

- (1) 将数式的值从弧度 (rad) 变换为度 (deg)。
- (2) 将位置数据的姿势角度以位置常数表示的情况下，((500,0,600,180,0,180)(7,0)) 的单位为 DEG。  
像 P1.C 一样的直接参照位置变量的回转成份的时候，单位会变为弧度 (rad)。P1.C 的值可以用 DEG 处理。此情况下，请将参数「PRGMDEG」设定为 1。

**【参照】**

[Rad](#)

## Dist

### 【功能】

求得 2 点间（位置变量）的距离。

### 【格式】

$$\langle \text{数值变量} \rangle = \text{Dist} (\langle \text{位置 1} \rangle, \langle \text{位置 2} \rangle)$$

### 【例子】

1 M1=Dist(P1,P2)                    ' 在 M1 代入 P1 ~ P2 的距离。

### 【说明】

- (1) 返回位置 1 和位置 2 的距离 (mm)。
- (2) 位置数据的角度位置被忽视，只使用 X, Y, Z 的数据计算。
- (3) 无法使用关节变量，使用时会发生报警。
- (4) 在  $\langle \text{位置 1} \rangle$ ,  $\langle \text{位置 2} \rangle$  无法记述有自变量的函数。执行时会发生报警。

## Exp

### 【功能】

计算指数函数（将 e 作为底的数式的次方）。

### 【格式】

$$\langle \text{数值变量} \rangle = \text{Exp} (\langle \text{数式} \rangle)$$

### 【例子】

1 M1=Exp(2)                    ' 在 M1 代入  $e^2$ 。

### 【说明】

- (1) 返回数式值的指数函数。

### 【参照】

[Ln](#)



## ***Fram***

### 【功能】

计算由 3 个的位置数据所指定的坐标位置数据。

### 【格式】

<位置变量 4 > = Fram (<位置 1 >, <位置 2 >, <位置 3 >)

### 【用语】

<位置 1 > 会变成以 3 个位置指定的平面的 X, Y, Z 的原点。变量或常数。  
 <位置 2 > 会变成以 3 个位置指定的平面的 X 轴的点。变量或常数。  
 <位置 3 > 会变成以 3 个位置指定的 X-Y 平面的 +Y 方向的点。变量或常数。  
 <位置变量 4 > 将结果代入变量。构造标志代入<位置 1 >的值。

### 【例子】

```
1 Base P_NBase
2 P10=Fram(P1,P2,P3) ' 以 P1,P2,P3 的位置为基础制作 P10 的坐标
3 P10=Inv(P10)
4 Base P10 ' 将 P10 的位置利用为机器人原点。
:
```

### 【说明】

- (1) 可以于定义基本坐标和作为基准坐标时使用。
- (2) 由 3 个位置变量的 X, Y, Z 坐标值做成平面, 计算原点的位置·平面的倾斜, 作为位置变量返回。此位置变量的位置数据 X, Y, Z 和位置变量 1 的值相同, 因此 A, B, C 会变成以 3 个的位置指定的平面倾斜。
- (3) 无法在自变量使用关节变量, 使用时会发生报警。
- (4) 在<位置 1 >, <位置 2 >, <位置 3 >无法记述有自变量的函数。执行时会发生报警。

**NG 例** P10=Fram(FPrm(P01,P02,P03), P04, P05)

### 【关连参数】

关于相对变换 (运算\*), 请参照第 434 页的「5.7 关于标准基本坐标」。

## Hex\$

### 【功能】

将数式的值（-32768 ~ 32767）变换为 16 进制字符串。

### 【格式】

<字符串变量> = Hex\$ (<数式> [, <输出文字数>])

### 【例子】

10 C1\$=Hex\$(&H41FF)            ' 在 C1\$ 代入 "41FF"。  
20 C2\$=Hex\$(&H41FF,2)        ' 在 C2\$ 代入 "FF"。

### 【说明】

- (1) 将数式的值变换为 16 进制字符串。
- (2) 指定<输出文字数>的话，变换字符串的右侧的文字会输出指定的尺寸量。
- (3) 数值不为整数的情况下，小数点部份会四舍五入后变换为 16 进制字符串的整数值。
- (4) 相反的动作请使用 Val 指令。
- (5) 指定<输出文字数>的情况下，无法在<数式>记述有自变量的函数。执行时会发生报警。  
**NG 例** C1\$=Hex\$(Asc("a"),1)

### 【参照】

[Bin\\$](#)、[Str\\$](#)、[Val](#)

## Int

### 【功能】

返回不超过数式值的最大整数。

### 【格式】

<数值变量> = Int (<数式>)

### 【例子】

1 M1=Int(3.3)                    ' 在 M1 代入 3。

### 【说明】

- (1) 返回不超过数式值的最大整数。
- (2) 数式的值为正的情况下，返回和 Fix 相同的值。
- (3) 数式的值为负的情况下，会变成例如：Fix(-2.3)=-3.0。

### 【参照】

[CInt](#)、[Fix](#)



## Left\$

**【功能】**

从字符串的左端，取得指定长度的字符串。

**【格式】**

|                                |
|--------------------------------|
| <字符串变量> = Left\$ (<字符串>, <数式>) |
|--------------------------------|

**【例子】**

1 C1\$=Left\$("ABC",2) '在 C1\$ 带入 "AB"。

**【说明】**

- (1) 从字符串的左端，取得指定长度的字符串。
- (2) 数值为负的情况及比字符串的长度还长的情况下，会发生报警。
- (3) <字符串>, <数式>中无法记述有自变量的函数，执行时会发生报警。

**【参照】**

[Mid\\$](#)、[Right\\$](#)

## Len

**【功能】**

返回字符串的长度。

**【格式】**

|                      |
|----------------------|
| <数值变量> = Len (<字符串>) |
|----------------------|

**【例子】**

1 M1=Len("ABCDEFGH") '在 M1 代入 7。

**【说明】**

- (1) 返回参数字串的长度。

**【参照】**

[Left\\$](#)、[Mid\\$](#)、[Right\\$](#)

## Ln

**【功能】**

给予自然对数。（底为 e。）

**【格式】**

|                                                                       |
|-----------------------------------------------------------------------|
| $\langle \text{数值变量} \rangle = \text{Ln} (\langle \text{数式} \rangle)$ |
|-----------------------------------------------------------------------|

**【例子】**

1 M1=Ln(2)                    ’ 在 M1 代入 0.693147。

**【说明】**

- (1) 返回数式的值的自然对数。
- (2) 数值式为 0 或是负的情况下会发生报警。

**【参照】**

[Exp](#)、[Log](#)

## Log

**【功能】**

给予常用对数。（底为 10。）

**【格式】**

|                                                                        |
|------------------------------------------------------------------------|
| $\langle \text{数值变量} \rangle = \text{Log} (\langle \text{数式} \rangle)$ |
|------------------------------------------------------------------------|

**【例子】**

1 M1=Log(2)                    ’ 在 M1 代入 0.301030。

**【说明】**

- (1) 返回数式的值的常用对数。
- (2) 数值式为 0 或是负的情况下会发生报警。

**【参照】**

[Exp](#)、[Ln](#)

## Max

**【功能】**

求得最大值。

**【格式】**

|                                      |
|--------------------------------------|
| <数值变量> = Max (<数式 1 >, <数式 2 >, ...) |
|--------------------------------------|

**【例子】**

1 M1=Max(2,1,3,4,10,100) '在 M1 代入 100。

**【说明】**

- (1) 从任意的个数的自变量当中返回最大值。
- (2) 可以记述在一行的可登录文字数 (240 文字) 为止的范围内。
- (3) 在<数式 1 >, <数式 2 >, 无法记述有自变量的函数。执行时会发生报警。

**【参照】**

[Min](#)

## Mid\$

**【功能】**

由字符串的指定位置将指定长度的字符串返回。

**【格式】**

|                                           |
|-------------------------------------------|
| <字符串变量> = Mid\$ (<字符串>, <数式 1 >, <数式 3 >) |
|-------------------------------------------|

**【例子】**

1 C1\$=Mid\$("ABCDEFGH",3,2) '在 C1\$ 代入 "CD"。

**【说明】**

- (1) 将第 1 自变量字符串的以从第 2 自变量指定位置到以第 3 自变量所指定的字符串长度返回。
- (2) 数式 2 或数式 3 的值为 0 或负的情况下, 会发生报警。
- (3) 取出位置的结束值, 比第 1 自变量的字符串长度大的情况下, 会发生报警。
- (4) 在<字符串>, <数式 2 >, <数式 3 >无法记述有自变量的函数。执行时会发生报警。

**【参照】**

[Left\\$](#)、[Right\\$](#)、[Len](#)

## Min

### 【功能】

求得最小值。

### 【格式】

<数值变量> = Min (<数式 1 >, <数式 2 >, ...)

### 【例子】

1 M1=Min(2,1,3,4,10,100)                    ' 在 M1 代入 1。

### 【说明】

- (1) 从任意的个数的自变量当中返回最小值。
- (2) 可以记述在一行的可登录文字数（240 文字）为止的范围内。
- (3) 在<数式 1 >, <数式 2 >, 无法记述有自变量的函数。执行时会发生报警。

### 【参照】

[Max](#)

## Mirror\$

### 【功能】

将字符串的各字符码 2 进制表示的 big string 反转求得字符码化的字符串。

### 【格式】

<字符串变量> = Mirror\$ (<字符串式>)

### 【例子】

1 C1\$=Mirror\$("BJ")                    ' 在 C1\$ 代入 "RB"。  
                                           ' "BJ"=&H42,&H4A=&B01000010,&B01001010  
                                           ' 反转 =&H52,&H42=&B01010010,&B01000010  
                                           ' 输出 ="RB"

### 【说明】

- (1) 将字符串的各字符码 2 进制表示的 big string 反转，求得字符码化的字符串。

## Mki\$

### 【功能】

将数式（整数）的值转换为 2 字节的字符串。

### 【格式】

```
<字符串变量> = Mki$ (<数式>)
```

### 【例子】

```
1 C1$=Mki$(20299) ' 在 C1$ 代入 "OK"。
2 M1=Cvi(C1$) ' 在 M1 代入 20299。
```

### 【说明】

- (1) 将数式（整数）的值的下级 2 字节转换为字符串式。
- (2) 请使用 Cvi 将字符串转换为数值。
- (3) 和外部之间以数值数据传输时，为了减少数据量时使用。

### 【参照】

[Asc](#)、[Cvi](#)、[Cvs](#)、[Cvd](#)、[Mks\\$](#)、[Mkd\\$](#)

## Mks\$

### 【功能】

将数式（单精度实数）的值转换为 4 字节的字符串。

### 【格式】

```
<字符串变量> = Mks$ (<数式>)
```

### 【例子】

```
1 C1$=Mks$(100.1)
2 M1=Cvs(C1$) ' 在 M1 代入 100.1。
```

### 【说明】

- (1) 将被写入数式（单精度实数）的值的下级 4 字节量转换为字符串式。
- (2) 请使用 Cvs 将字符串转换为数值。
- (3) 和外部之间以数值数据传输时，为了减少数据量时使用。

### 【参照】

[Asc](#)、[Cvi](#)、[Cvs](#)、[Cvd](#)、[Mki\\$](#)、[Mkd\\$](#)

## Mkd\$

### 【功能】

将数式（双精度实数）的值转换为 8 字节的字符串。

### 【格式】

|                        |
|------------------------|
| <字符串变量> = Mkd\$ (<数式>) |
|------------------------|

### 【例子】

1 C1\$=Mkd\$(10000.1)

2 M1=Cvd(C1\$)                    ' 在 M1 代入 10000.1。

### 【说明】

- (1) 将被写入数式（双精度实数）的值的下级 8 字节量转换为字符串式。
- (2) 请使用 Cvd 将字符串转换为数值。
- (3) 和外部之间以数值数据传输时，为了减少数据量时使用。

### 【参照】

Asc、Cvi、Cvs、Cvd、Mki\$、Mks\$

## PosCq

### 【功能】

调查被给予的位置是否有进入动作范围内。

### 【格式】

$\langle \text{数值变量} \rangle = \text{PosCq} (\langle \text{位置变量} \rangle)$

### 【例子】

1 M1=PosCq(P1)                    'P1 的位置有在动作范围内的话，在 M1 代入 1。

### 【说明】

- (1) 用自变量调查被给予的位置数据是否有进入机器人的动作范围内。进入机器人的动作范围内的话返回 1，动作范围外的话则返回 0。
- (2) 自变量必须给予位置数据型、关节数据型的其中一个。

## PosMid

### 【功能】

求得在被给予的 2 点间做直线插补时的中间位置数据。

### 【格式】

$\langle \text{位置变量} \rangle = \text{PosMid} (\langle \text{位置变量} 1 \rangle, \langle \text{位置变量} 2 \rangle, \langle \text{数式} 1 \rangle, \langle \text{数式} 2 \rangle)$

### 【例子】

1 P1=PosMid(P2,P3,0,0)            '在 P1 代入 P2,P3 的中间点的位置数据（含姿势）。

### 【说明】

- (1) 在 2 个位置数据间做直线插补的时候，求得中点的位置数据。
- (2) 第 1 自变量给予直线插补的起点，第 2 自变量指定直线插补的终点。
- (3) 第 3 自变量、第 4 自变量相当于 Mvs 指令的 2 个类型自变量。
- (4) 起点和终点的自变量必须以指定的插补种类在直线插补可能的位置数据。例如：起点和终点的构造标志不同的情况下会发生报警。
- (5) 在  $\langle \text{位置变量} 1 \rangle$ ， $\langle \text{位置变量} 2 \rangle$ ， $\langle \text{数式} 1 \rangle$ ， $\langle \text{数式} 2 \rangle$ ，无法记述有自变量的函数。执行时会发生报警。

## PtoJ

### 【功能】

将被给予的位置数据转换为关节数据。

### 【格式】

<关节变量> = PtoJ (<位置变量>)

### 【例子】

1 J1=PtoJ(P1)                    ' 在 J1 代入将 P1 (直交型的位置变量) 的值转换为关节型。

### 【说明】

- (1) 将位置数据做坐标变换, 并转换为关节数据。
- (2) 在自变量无法使用关节变量 (J 变量)。使用的情况下会发生报警。
- (3) 因为返回值为关节数据, 因此在左边使用位置变量的时候会发生报警。

### 【参照】

[JtoP](#)

## Rad

### 【功能】

将角度单位度 (deg) 转换为弧度 (rad)。

### 【格式】

<数值变量> = Rad (<数式>)

### 【例子】

```

1 P1=P_Curr
2 P1.C=Rad(90)
3 Mov P1 ' 只往将 C 轴从现在位置变为 90 度的 P1 移动。

```

### 【说明】

- (1) 将数式的值由角度单位度 (deg) 转换为弧度 (rad)。
- (2) 常使用于代入位置变量的姿势成份的情况及计算三角函数的情况。

### 【参照】

[Deg](#)

## Rdf11

### 【功能】

将指定的位置构造标志以文字数据 "R"/"L", "A"/"B", "N"/"F" 返回。

### 【格式】

<字符串变量> = Rdf11 (<位置变量>, <数式>)

### 【用语】

<位置变量>      指定想取出构造标志的位置变量。  
 <数值>            指定想取出的哪个构造标志。  
 0="R"/"L", 1="A"/"B", 2="N"/"F"

### 【例子】

1 P1=(100,0,100,180,0,180)(7,0)      ' 由于构造 flag7(&B111) 为 RAN  
 2 C1\$=Rdf11(P1,1)                      ' 在 C1\$ 代入 "A"。

### 【说明】

- (1) 在第 1 自变量的位置数据的构造标志中, 取出以第 2 自变量指定的值。
- (2) 本函数摘取自位置数据的 FL1 的要素情报。
- (3) 在<位置变量>, <数式>无法记述有自变量的函数。执行时会发生报警。

### 【参照】

[Rdf12](#)、[Setf11](#)、[Setf12](#)

## Rdf12

### 【功能】

将指定的关节轴多回转情报以数值返回。

### 【格式】

<数值变量> = Rdf12 (<位置变量>, <数式>)

### 【用语】

<位置变量> 指定想摘取多回转情报的位置变量。  
<数式> 指定想摘取多回转情报的关节轴的值。(1 ~ 8)

### 【例子】

1 P1=(100,0,100,180,0,180)(7,&H00100000) ' '  
2 M1=Rdf12(P1,6) ' 在 M1 代入 1。

### 【说明】

- (1) 在第 1 自变量的位置数据的多回转情报中，摘取以第 2 自变量指定的关节轴的值。
- (2) 返回的值范围会变成 -8 ~ 7。
- (3) 本函数摘取自位置数据的 FL2 的要素情报。
- (4) 构造标志 2 (多回转情报) 以 32 位构造，相当于 1 轴 4 位 8 轴量被设定。
- (5) 当显示在示教单元时，多回转为负的情况，将 -1 ~ -8 转换为 F ~ 8 (4 位符号附 16 进制标记) 显示。  
<在示教单元的多回转情报的表示例> 87654321 轴<相当于 1 轴的多回转数和显示的关系>  
J6 轴为多回转 +1 的情况: FL2=00100000      ..... -2 -1 0 +1 +2 .....  
J6 轴为多回转 -1 的情况: FL2=00F00000      ..... E F 0 1 2 .....
- (6) 在<位置变量>, <数式>无法记述有自变量的函数。执行时会发生报警。

### 【参照】

[Rdf11](#)、[Setf11](#)、[Setf12](#)

### 【相关指令】

[JRC \(Joint Roll Change\)](#)

## Rnd

### 【功能】

使随机数发生。

### 【格式】

<数值变量> = Rnd (<数式>)

### 【用语】

<数式>                    设定随机数的初始值。设定为 0 的情况下，不会执行随机数的初始值设定，而会继续摘取随机数。

<数值变数>                以 0.0 ~ 1.0 的范围将值返回。

### 【例子】

```

1 Dim MRND(10)
2 C1=Right$(C_Time,2) ' 每次执行都会得到不同的顺序、
3 MRNDBS=Cvi(C1) ' 使用时间值将随机数初始化。
4 MRND(1)=Rnd(MRNDBS) ' 设定随机数的初始值，摘取随机数的第 1 个。
5 For M1=2 To 10 ' 摘取剩余的 9 个随机数。
6 MRND(M1)=Rnd(0)
7 Next M1

```

### 【说明】

- (1) 用自变量以被给予数式的值将随机数初始化后摘取随机数。
- (2) 用自变量以被给予数式的值为 0 的情况下，不会执行随机数而会继续摘取随机数。
- (3) 以相同数值执行随机数的初始化时，得到的随机数列会变成相同值。

## Right\$

### 【功能】

从字符串的右端摘取出指定长度的字符串。

### 【格式】

<字符串变量> = Right\$ (<字符串>, <数式>)

### 【例子】

```

1 C1$=Right$("ABCDEFGF",3) ' 在 C1$ 代入 "EFG"。

```

### 【说明】

- (1) 从字符串的右端摘取出指定长度的字符串。
- (2) 第 2 自变量的数值为负的情况及比第 1 字符串长度大的情况下，会发生报警。
- (3) 在<字符串>, <数式>无法记述含有自变量的函数。执行时会发生报警。

### 【参照】

[Left\\$](#)、[Mkd\\$](#)、[Len](#)



## Setfl2

**【功能】**

变更指定位置的多回转数据（构造标志 2）。

**【格式】**

<位置变量> = Setfl2 (<位置变量>, <数式 1 >, <数式 2 >)

**【用語】**

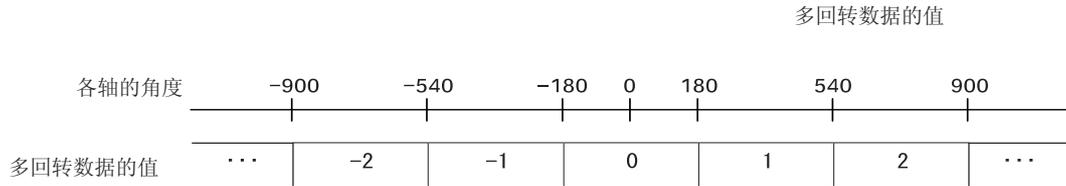
- <位置变量> 指定将多回转数据（构造 Flag2）变更的位置变量。
- <数值 1 > 指定将多回转数据（构造 Flag2）变更的轴号码。(1 ~ 8)
- <数值 2 > 定变更多回转数据值。(-8 ~ 7)

**【例子】**

```
10 Mov P1
20 P2=Setfl2(P1,6,1)
30 Mov P2
```

**【说明】**

- (1) 将以位置数据的数式 1 指定的关节轴的多回转情报返还以数式 2 将指定值变更的位置数据。
- (2) 本函数变更位置数据的 FL2 的要素情报。  
相当于位置变量的 (100, 0, 300, 180, 0, 180) (7, 0) 的 "0" 的地方。
- (3) 以自变量被给予的位置数据内容 (X,Y,Z,A,B,C,FL1) 不会被改变。



- (4) 在<位置变量>, <数式 1 >, <数式 2 >无法记述含有自变量的函数。执行时会发生报警。

**【参照】**

[Rdfl1](#)、[Rdfl2](#)、[Setfl1](#)

## SetJnt

### 【功能】

设定往关节变量的值。

### 【格式】

```
<关节变量>= SetJnt (< J1 轴> [, < J2 轴> [, < J3 轴>
 [, < J4 轴> [, < J5 轴> [, < J6 轴> [, < J7 轴> [, < J8 轴>]]]]]])
```

### 【用语】

<关节变量>                    指定变更的关节变量。  
 < J1 轴> ~ < J8 轴>        单位为 RAD( 直动轴为 mm)。

### 【例子】

```
1 J1=J_Curr
2 For M1=0 To 60 Step 10
3 M2=J1.J3+Rad(M1)
4 J2=SetJnt(J1.J1, J1.J2, M2) ' 只使 J3 轴的值每 10 度回转, J4 轴以后为相同值
5 Mov J2
6 Next M1
7 M0=Rad(0)
8 M90=Rad(90)
9 J3=SetJnt(M0,M0,M90,M0,M90,M0)
10 Mov J3
```

### 【说明】

- (1) 可以变更关节变量的各轴的值。
- (2) 在自变量可以记述变量。
- (3) J1 轴以外可以省略。省略会变成以后的轴全部省略。无法像 (SetJnt(10,10,,,10) 一样的记述。)
- (4) 自变量: 无法记述含有自变量的函数。执行时会发生报警。

### 【参照】

[SetPos](#)

### 【关连参数】

AXUNT、PRGMDEG

## SetPos

### 【功能】

设定位置变量的值。

### 【格式】

```
<位置变量> = SetPos (< X 轴> [, < Y 轴> [, < Z 轴>
 [, < A 轴> [, < B 轴> [, < C 轴> [, < L1 轴> [, < L2 轴>]]]]]])
```

### 【用语】

- <位置变量> 指定变更的位置变量。
- < X 轴>~< Z 轴 e > 单位为 mm。
- < A 轴>~< C 轴> 单位为 RAD。（可使用 PRGMDEG 参数转换为 DEG。）
- < L1 轴>、< L2 轴> 单位依存在 AXUNT 参数。

### 【例子】

```
1 P1=P_Curr
2 For M1=0 To 100 Step 10
3 M2=P1.Z+M1
4 P2=SetPos(P1.X, P1.Y, M2) ' 只使 Z 轴的值上升 10mm, A 轴以后为相同值
5 Mov P2
6 Next M1
```

### 【说明】

- (1) 可以变更位置变量的各轴的值。
- (2) 在自变量可以记述变量。
- (3) X 轴以外可以省略。省略会变成以后的轴全部省略。无法像 (SetPos(10,10,,,10) 一样记述。)
- (4) 自变量：无法记述含有自变量的函数。执行时会发生报警。

### 【参照】

[SetJnt](#)

### 【关连参数】

AXUNT、PRGMDEG

## Sgn

**【功能】**

调查数式的符号。

**【格式】**

<数值变量> = Sgn (<数式>)

**【例子】**

1 M1=-12

2 M2=Sgn(M1) ' 在 M2 代入 -1。

**【说明】**

(1) 调查被给予的数式值的符号，返回如下列的值。

|     |    |
|-----|----|
| 正的数 | 1  |
| 0   | 0  |
| 负的数 | -1 |

## Sin

**【功能】**

计算正弦（sin）。

**【格式】**

$$\langle \text{数值变量} \rangle = \text{Sin} (\langle \text{数式} \rangle)$$

**【例子】**

1 M1=Sin(Rad(60))                    ' 在 M1 代入 0.86603。

**【说明】**

- (1) 计算数式的值的正弦（sin）的值。
- (2) 值的范围会变成数值的值可以取得的全部范围。
- (3) 返回值的范围会变成 -1 ~ 1。
- (4) 自变量的单位为弧度。

**【参照】**

Cos、Tan、Atn/Atn2

## SplECord

**【功能】**

将登录在样条文件中的 Ex-T 坐标系原点数据保存至位置变量。

**【格式】**

$$\langle \text{位置变量} \rangle = \text{SplECord}(\langle \text{样条号码} \rangle[, \langle \text{帧转换} \rangle])$$

**【用语】**

- <位置变量>            要代入结果的变量。
- <样条号码>            以常数或数值变量指定保持有希望获取的路径点信息的样条文件的号码。  
设定范围：1~99
- <帧转换>              以常数或数值变量指定帧转换的执行内容。  
设定范围：  
0：不执行帧转换。  
1：使用样条文件中设定的坐标系执行帧转换。  
2：使用SetCalFrm指令中设定的坐标系执行帧转换。  
省略时：不执行帧转换。

**【例文】**

- 1 P\_WkCord(1)=SplECord(8)            ' 将样条文件 1 的路径点 5 号的位置数据代入至位置变量 P1
- 2 P1=SplECord(5)                    ' 将登录在样条文件 5 中的 Ex-T 坐标系原点数据代入至位置变量 P1
- 3 P2=SplECord(10,1)                ' 对登录在样条文件 10 中的 Ex-T 坐标系原点数据进行了帧转换后，将数据代入至 P2。

**【说明】**

- (1) 返回 <样条号码> 中指定的样条文件中所登录的 Ex-T 坐标系原点数据。
- (2) <样条号码> 对应的样条文件未保存在控制器中的情况下，会发生 L2610(无法打开样条文件) 错误。

- (3) < 帧转换 > 指定为 “1(使用样条文件中设定的坐标系执行)” 或 “2(使用通过 SetCalFrm 指令设定的坐标系执行)” 时，基于指定的方法转换 Ex-T 坐标系原点数据，返回补偿后的数据。
- (4) 即使 < 帧转换 > 已指定为 “1(使用样条文件中设定的坐标系执行)”，样条文件中未设定坐标系时，也会发生 L2042(帧转换坐标未设定) 错误。
- (5) < 帧转换 > 指定为 “1(使用样条文件中设定的坐标系执行)” 时，无法算出坐标系的情况下，会发生 L2041(帧转换坐标无法算出) 错误。
- (6) 若 < 帧转换 > 指定为 “2(使用通过 SetCalFrm 指令设定的坐标系执行)”，则使用通过最后执行的 SetCalFrm 指令设定的坐标系执行帧转换。从未执行过 SetCalFrm 指令且未设定坐标系时，会发生 L2042(帧转换坐标未设定) 错误。
- (7) 样条文件中未进行 Ex-T 坐标设定时，会发生 L2610(Ex-T 坐标设定不当) 错误。

## SplPos

### 【功能】

将登录在样条文件中的任意路径点数据保存至位置变量。

### 【格式】

```
< 位置变量 > = SplPos(< 样条号码 >, < 路径点号码 >[, < 帧转换 >])
```

### 【用语】

- < 位置变量 > 要代入结果的变量。结构标志将代入指定的样条文件中所登录的路径点数据的值。
- < 样条号码 > 以常数或数值变量指定保持有希望获取的路径点信息的样条文件的号码。  
设定范围：1～99
- < 路径点号码 > 以常数或数值变量指定希望获取的路径点的号码。  
设定范围：1～5000
- < 帧转换 > 以常数或数值变量指定帧转换的执行内容。  
设定范围：  
0：不执行帧转换。  
1：使用样条文件中设定的坐标系执行帧转换。  
2：使用SetCalFrm指令中设定的坐标系执行帧转换。  
省略时：不执行帧转换。

### 【例文】

- 1 P1=SplPos(1,5)            ’ 将样条文件 1 的路径点 5 号的位置数据代入至位置变量 P1
- 2 P2=SplPos(1,10,1)        ’ 对样条文件 1 的路径点 10 号的位置数据进行了帧转换后，将数据代入至位置变量 P2

### 【说明】

- (1) 返回 < 样条号码 > 中指定的样条文件中所登录的 < 路径点号码 > 对应的路径点的位置数据。
- (2) 结构标志的值为样条文件中所登录的路径点数据的值。
- (3) < 样条号码 > 对应的样条文件未保存在控制器中的情况下，会发生 L2610(无法打开样条文件) 错误。
- (4) < 路径点号码 > 对应的路径点信息未登录在样条文件的情况下，会发生 L2611(路径点数据未登录) 错误。
- (5) < 帧转换 > 指定为 “1(使用样条文件中设定的坐标系执行)” 或 “2(使用通过 SetCalFrm 指令设定的坐标系执行)” 时，基于指定的方法转换路径点数据，返回该路径点。

- (6) 即使 < 帧转换 > 已指定为 “1( 使用样条文件中设定的坐标系执行 )”，样条文件中未设定坐标系时，也会发生 L2042( 帧转换坐标未设定 ) 错误。
- (7) < 帧转换 > 指定为 “1( 使用样条文件中设定的坐标系执行 )” 时，无法算出坐标系的情况下，会发生 L2041( 帧转换坐标无法算出 ) 错误。
- (8) 若 < 帧转换 > 指定为 “2( 使用通过 SetCalFrm 指令设定的坐标系执行 )”，则使用通过最后执行的 SetCalFrm 指令设定的坐标系执行帧转换。从未执行过 SetCalFrm 指令且未设定坐标系时，会发生 L2042( 帧转换坐标未设定 ) 错误。

## SplSpd

### 【功能】

通过样条插补指令 (MvSpl、EMvSpl 指令) 算出不会发生错误且可指定的最大速度 [mm/s]。

### 【格式】

< 数值变量 > = SplSpd(< 样条号码 >)

### 【用语】

- < 数值变量 >      要代入结果的变量。
- < 样条号码 >      以常数或数值变量指定保持有希望获取的路径点信息的样条文件的号码。  
设定范围：1～99

### 【例文】

- 1 MS=SplSpd(8)      ’ 使用样条文件 8 的路径点数据执行样条插补时，将不会发生错误且可指定的最大速度代入至 MS。
- 2 MvSpl 8,MS,50      ’ 执行以 MS 中设定的速度通过样条文件 8 的路径点的样条插补

### 【说明】

- (1) 从 < 样条号码 > 指定的样条文件中所登录的路径点数据，算出执行样条指令 (MvSpl、EMvSpl 指令) 时不会发生错误且可指定的最大速度。
- (2) 所算出的最大速度为以程序倍率修调为 100% 时的速度。
- (3) 使用算出的速度执行了样条插补的情况下，机器人的姿势若发生较大变化时，则有可能发生超速。
- (4) < 样条号码 > 对应的样条文件未保存在控制器中的情况下，会发生 L2610( 无法打开样条文件 ) 错误。

## Sqr

**【功能】**

计算数式的平方根。

**【格式】**

```
<数值变量> = Sqr (<数式>)
```

**【例子】**

1 M1=Sqr(2)                    ' 在 M1 代入 1.41421。

**【说明】**

- (1) 计算数式的值的平方根。
- (2) 在自变量被给予数式的值为负的情况下，会发生报警。

## Strpos

**【功能】**

在字符串内检索指定的字符串。

**【格式】**

```
<数值变量> = Strpos (<字符串 1 >, <字符串 2 >)
```

**【例子】**

1 M1=Strpos("ABCDEFGF","DEF")    ' 在 M1 代入 4。

**【说明】**

- (1) 从第 1 自变量的字符串中，返回以第 2 自变量指定字符串中最初出现的位置。
- (2) 字符串 2 的长度为 0 的情况下会发生报警。
- (3) 例如：字符串 1 为 "ABCDEFGF"、字符串 2 为 "DEF" 的时候，会返回 4。
- (4) 找不到字符串的情况下会返回 0。
- (5) 在<字符串 1>，<字符串 2>无法记述含有自变量的函数。执行时会发生报警。

## Str\$

**【功能】**

将数式的值转换为 10 进制字符串。

**【格式】**

```
<字符串变量> = Str$ (<数式>)
```

**【例子】**

1 C1\$=Str\$(123)                    ' 在 C1\$ 代入 "123"。

**【说明】**

- (1) 将数式的值转换为 10 进制字符串。
- (2) 相反的动作请使用 Val 指令。

**【参照】**

Bin\$、Hex\$、Val

## Tan

**【功能】**

计算正切 (tangent)。

**【格式】**

```
<数值变量> = Tan (<数式>)
```

**【例子】**

1 M1=Tan(Rad(60))                    ’ 在 M1 代入 1.73205。

**【说明】**

- (1) 返回数式的值的正切 (Tangent) 的值。
- (2) 以自变量给予值的范围，会变成数值的值可以取得的全部范围。
- (3) 返回值的范围，会变成数值的值可以取得的全部范围。
- (4) 自变量的单位为弧度。

**【参照】**

[Sin](#)、[Cos](#)、[Atn/Atn2](#)

## Val

**【功能】**

将字符串的值转换为数值。

**【格式】**

```
<数值变量> = Val (<字符串式>)
```

**【例子】**

1 M1=Val("15")  
2 M2=Val("&B1111")  
3 M3=Val("&HF")

**【说明】**

- (1) 将被给予的字符串式转换为数值。
- (2) 在字符串里可以使用 2 进制 (&B)、10 进制、16 进制 (&H)。
- (3) 在上述例子中 M1,M2,M3 会变成相同的值 (15)。

**【参照】**

[Bin\\$](#)、[Hex\\$](#)、[Str\\$](#)

## Zone

### 【功能】

确认指定的位置是否有进入指定领域内（以 2 点指定的长方体）。

### 【格式】

```
<数值变量> = Zone (<位置 1>, <位置 2>, <位置 3>)
```

### 【用语】

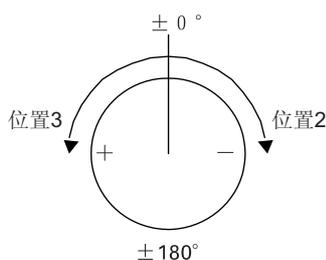
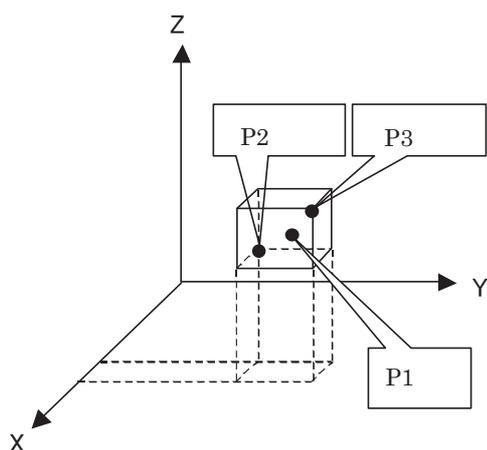
<位置 1> 想确认的位置  
 <位置 2> 指定领域的第 1 点的位置  
 <位置 3> 指定领域的第 2 点的位置（对角点）  
 位置 1 ~ 3 设定为直交坐标的变量（P 变量 X,Y,Z,A,B,C,L1,L2）。

### 【例子】

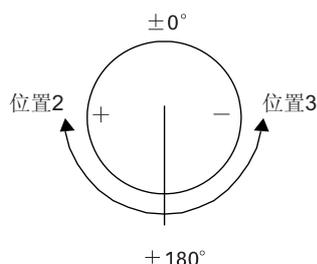
```
1 M1=Zone(P1,P2,P3)
2 If M1=1 Then Mov P_Safe Else End
```

### 【说明】

- 确认位置 1 是否有进入以位置 2 和位置 3 的 2 点所指定的长方体（2 点会变成成立方体的对角点）的空间，且在进入时返回 1，没有进入时返回。
- 确认位置 1 是否有进入空间里，每一个位置 1 的要素 (X,Y,Z,A,B,C,L1,L2) 的值会确认是否在位置 1 和位置 2 的值之间。
- 关于姿势角 (A,B,C) 是确认是否有进入从位置 2 的角度在正的方向回转，到到达位置 3 的角度为止的范围内。例) P2.A 为 -100，P3.A 为 +100 的情况下，P1.A 为 50 的话会变成范围内。也同样确认 B,C 轴（参照下图）。
- 对于没有确认或不存在的位置成份，单位为 deg 的情况下，位置 2 设定在 -360、位置 3 设定在 360、单位为 mm 的情况下，位置 2 设定在 -10000、位置 3 设定在 10000。
- 在<位置 1>，<位置 2>，<位置 3>无法记述含有自变量的函数。执行时会发生报警。



例) -90~0~+90的情况下  
 在<位置2>的ABC中代入-侧的值。  
 在<位置3>的ABC中代入+侧的值。



例) -90~180~+90的情况下  
 在<位置2>的ABC中代入+侧的值。  
 在<位置3>的ABC中代入-侧的值。

## Zone2

### 【功能】

确认指定的位置是否有进入指定领域内（以 2 点构成的圆筒体）。

### 【格式】

<数值变量> = Zone2 (<位置 1 >, <位置 2 >, <位置 3 >), <数式>

### 【用语】

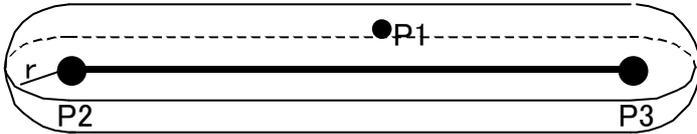
|         |              |
|---------|--------------|
| <位置 1 > | 想确认的位置       |
| <位置 2 > | 指定领域第 1 点的位置 |
| <位置 3 > | 指定领域第 2 点的位置 |
| <数式>    | 两端半球的半径      |

### 【例子】

```
1 M1=Zone2(P1,P2,P3,50)
2 If M1=1 Then Mov P_Safe Else End
```

### 【说明】

- (1) 确认位置 1 是否有进入以位置 2 和位置 3 的 2 点所指定的以半径构成的空间（参照下图），且在进入时返回 1，没有进入时返回 0。
- (2) 此函数是确认位置（X，Y，Z 的各坐标）是否有进入指定空间内的函数，不考虑姿势成份等。



- (3) 在<位置 1 >, <位置 2 >, <位置 3 >, <数式>无法记述含有自变量的函数。执行时会发生报警。

## Zone3

### 【功能】

确认指定的位置是否有进入指定领域内（以 3 点构成平面为基础的长方体）。

### 【格式】

<数值变量> = Zone3 (<位置 1 >, <位置 2 >, <位置 3 >, <位置 4 >, <数式 W >, <数式 H >, <数式 L >)

### 【用语】

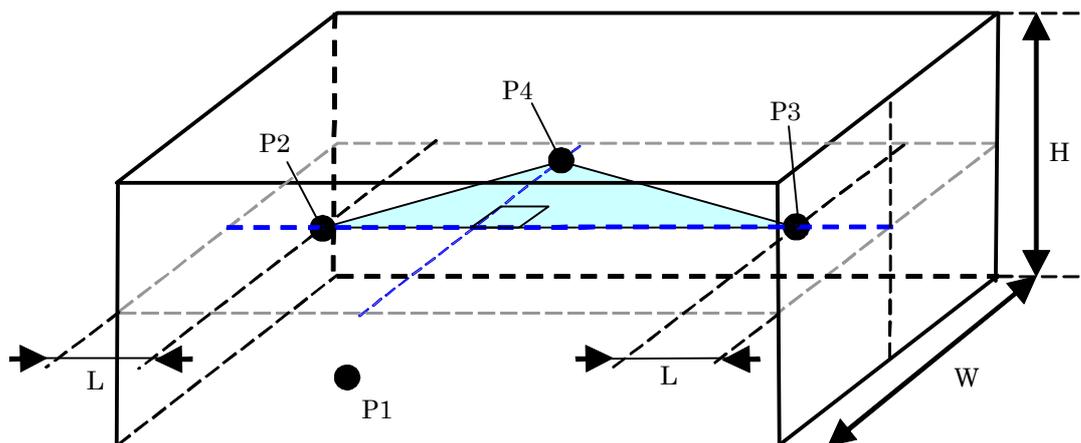
|         |                                     |
|---------|-------------------------------------|
| <位置 1 > | 想确认的位置                              |
| <位置 2 > | 指定领域第 1 点的位置                        |
| <位置 3 > | 指定领域第 2 点的位置                        |
| <位置 4 > | 随着<位置 2 >、<位置 3 >指定构成领域的平面指定点的位置    |
| <数式 W > | 构成领域长方体的宽 [mm]                      |
| <数式 H > | 构成领域长方体的高 [mm]                      |
| <数式 L > | 构成领域长方体的<位置 2 >、<位置 3 >开始的各别纵深 [mm] |

## 【例子】

```
1 M1=Zone3(P1,P2,P3,P4,100,100,50)
2 If M1=1 Then Mov P_Safe Else End
```

## 【说明】

- 1) 执行位置 1 是否有进入以位 2、位置 3、位置 4 的 3 点及数式 W、数式 H、数式 L 所构成的空间内（参照下图），进入的情况下返回 1，没有进入的情况下返回 0。
- 2) 此函数是确认位置 <位置 1>（X、Y、Z 的各坐标）是否有进入指定空间内的函数，不考虑姿势成份等。



- 3) 在 <位置 1> ~ <位置 4>、<数式 W> ~ <数式 L> 无法记述含有自变量的函数。执行时会发生报警。
- 4) 在 <数式 W>、<数式 H> 输入负的情的情况下会发生报警。
- 5) 在 <位置 2> ~ <位置 4> 同一位置或同一直在线的位置输入的话，因为不能指定制作领域，因此不会执行确认返回 -1。  
在 <数式 L> 为负的数，绝对值为 <位置 2> 和 <位置 3> 的中间的距离的一半的数的情况下，不会执行确认的返回 -1。

## 5 参数的设定功能

本控制器有表 5-1 所显示的各种参数。依据参数的设定变更，可以变更各种功能及初始设定。

| No. | 分类   | 内容                                           | 参照页次    |
|-----|------|----------------------------------------------|---------|
| 1   | 动作参数 | 机器人的动作范围、坐标、抓手关连的设定用的参数。                     | 第 407 页 |
| 2   | 信号参数 | 信号关连用的参数。关于专用信号请参照第 509 页的「6.2PLC 连接输出输入功能」。 | 第 418 页 |
| 3   | 操作参数 | 控制器、示教单元等的操作相关参数。                            | 第 425 页 |
| 4   | 语言参数 | 机器人语言相关参数。                                   | 第 427 页 |
| 5   | 通信参数 | 通信相关参数                                       | 第 430 页 |

关于专用信号请参照第 509 页的「6.2PLC 连接输出输入功能」。

变更参数的情况下请务必将机器人控制器的电源重新启动（OFF → ON）。若没有重新启动电源的话，参数的设定内容不会变有效。参数的详细的操作方法请参照本书第 84 页的「3.14 参数画面的操作」。



### 注意

参数的变更，请确定其功能和设定值后再执行。

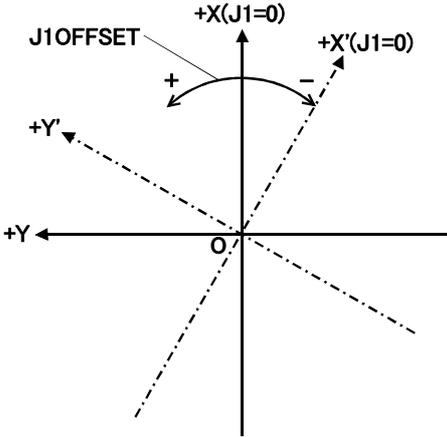
没有确认就执行的情况下，机器人会有预期以外动作出现而导致人身事故及物品损伤。

### 5.1 动作参数

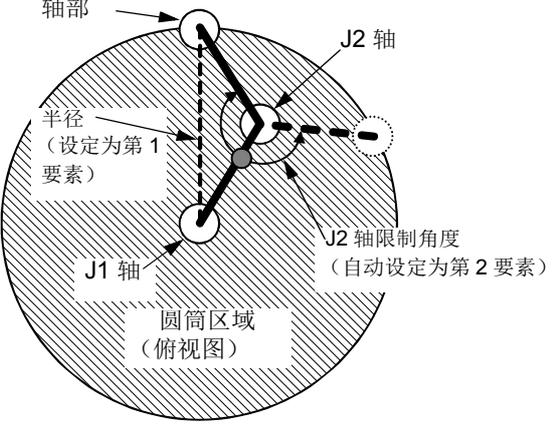
机器人的动作范围、坐标、抓手关连的设定用的参数。

表 5-1: 动作关连参数一览表

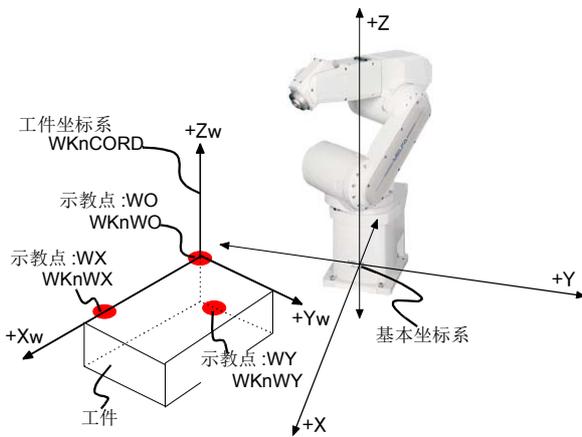
| 参数                                        | 参数名     | 排列数<br>文字数 | 内容说明                                                                                                                                                            | 出货时设定值                                                                 |
|-------------------------------------------|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| 关节动作范围                                    | MEJAR   | 实数 16      | 指定关节坐标的速度比例限制值。<br>设定各轴的动作范围。不建议将动作范围扩大。可能会撞上机械限位器。<br>注)垂直5轴型机器人中设定了J1偏置角度(J1OFFSET)时, 请注意无法更改J1轴的关节动作范围。<br>正号、负号的2方向 (-J1,+J1,-J2,+J2.....-J8,+J8) 单位: deg   | 每个机器人的设定值                                                              |
| 直交动作范围                                    | MEPAR   | 实数 6       | 指定直交坐标的速度比例限制值。<br>将机器人的动作范围以直交坐标限制。将机器人设置于设备内, 在手动操作时为了不想碰撞到外围设备的情况下设定。<br>负号、正号的2方向 (-X, +X, -Y, +Y, -Z, +Z) 单位: mm                                           | (-X,+X,-Y,+Y,-Z,+Z)=<br>-10000,10000,<br>-10000,10000,<br>-10000,10000 |
| 窄角广角限制功能<br>※ 仅限 RV-4F/7F/<br>13F/20F 系列。 | MELTEXS | 整数 1       | 窄角广角限制功能用于保护机械臂前端不对基座产生干涉, 可对该功能进行有效/无效的切换。<br>机械臂前端需要在基座附近位置动作时, 请将本参数设定为"0"。此时, 请注意勿使机械臂前端和基座产生干涉。<br>(关于窄角广角限制领域, 请参照记载于「附册/标准规格书」上的动作范围图)<br>0: 无效<br>1: 有效 | 1 (有效)<br>仅限RV-4F/7F/13F/<br>20F系列。<br>其他机型为0 (无效)                     |

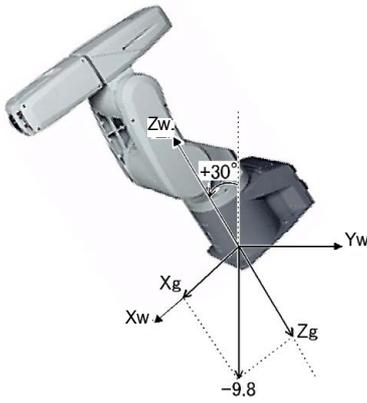
| 参数                                                   | 参数名                    | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                       | 出货时设定值                                             |
|------------------------------------------------------|------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| J1 轴偏置角度                                             | J1OFFSET               | 实数 2       | <p>设定垂直 5 轴型机器人的 J1 轴偏置角度。<br/>在其他类型的机器人中不起作用。<br/>注) 设定 J1OFFSET 即会更改 J1=0 度的方向, 因此会自动补偿关节动作范围 (MEJAR)。此外, 设定了 J1OFFSET 时, 请注意无法更改 J1 轴的关节动作范围。</p> <p>第1要素: 指定本功能的有效/无效。将功能设为有效时请设定 1.0, 设为无效时请设定 0.0。<br/>第2要素: 可以将 J1=0 度 (X 轴) 的方向仅偏置该参数中设定的角度。(单位: 度)<br/>设定范围: <math>-360 \leq \text{第2要素} \leq +360</math></p>  | 0.0, 0.0                                           |
| 标准工具坐标<br>请参照「4.5 机器人的坐标系说明」、<br>「5.6 关于标准 TOOL 坐标」。 | MEXTL                  | 实数 6       | <p>对抓手前端 (控制点) 与机械接口 (抓手安装面) 的关系设定初始值。<br/>出厂时设定机械接口为控制点。<br/>希望将安装抓手控制点设为抓手前端时请更改。(直交或工具 JOG 时可通过抓手前端进行姿势控制)<br/>(X,Y,Z,A,B,C) 单位: XYZ 为 mm, ABC 为 deg</p>                                                                                                                                                                                                                                                | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
| TOOL 坐标 1 ~ 16<br>请参照「M_Tool」。                       | MEXTL1<br>:<br>MEXTL16 | 实数 6       | <p>在 M_TOOL 变量中代入 1~16 后, 可通过各自的参数值切换 TOOL 数据。</p>                                                                                                                                                                                                                                                                                                                                                         | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
| 标准基本坐标<br>请参照「4.5 机器人的坐标系说明」、<br>「5.7 关于标准基本坐标」。     | MEXBS                  | 实数 6       | <p>设定基本坐标和机器人坐标的位置关系。<br/>出货时设定基本坐标和机器人坐标为一致。作为装置全体的坐标要变更时的设定。此参数不太需要做变更。在想使装置全体坐标统一等的时候设定。<br/>(X, Y, Z, A, B, C) 单位: XYZ 为 mm、ABC 为 deg<br/>注) 程序处于运行中或中断中时, 不能变更值。</p>                                                                                                                                                                                                                                  | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
| 基本坐标号码<br>请参照「4.5 机器人的坐标系说明」。                        | MEXBSNO                | 整数 1       | <p>指定基本坐标号码, 设定世界坐标系。(基本变换)<br/>另外, 显示现在的设定状态。<br/>设定值的内容:<br/>0: 指定 P_NBase (系统初始值) (P_NBase=0, 0, 0, 0, 0, 0, 因此清除基本变换)<br/>1~8: 指定工件坐标系 1~8 (参数: WK1CORD~WK8CORD 的值) 的设定值<br/>-1: 使用 Base 指令直接指定了基本变换数据, 或使用标准基本坐标参数: MEXBS 进行了直接指定。<br/>(注: 设定值 -1 为仅读取有效)<br/>注) 程序处于运行中或中断中时, 无法变更值。</p>                                                                                                             | -1                                                 |

| 参数                            | 参数名                                | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 出货时设定值 |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
|-------------------------------|------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----|-----|---------|--------------|----------|----------------|------------|------|-----------------------|---------|--------|--------------------------------------------------------------------------------|
| 使用者定义领域<br>请参照「5.8关于使用者定义领域」。 | 指定使用者定义领域的设定（最大32领域）及在该领域内的机器人的举动。 |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
|                               | AREA*CS<br>*为1~32                  | 整数 1       | 设定使用者定义领域*的坐标系。（* = 1~32）<br>0: 世界坐标系（向上兼容）<br>1: 基本坐标系                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0      |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
|                               | AREA*P1<br>*为1~32                  | 实数 8       | 设定使用者定义领域*的对角点1的位置坐标以及姿势数据・附加轴坐标。从第1要素开始，按X、Y、Z、A、B、C、L1、L2的顺序定义。<br><table border="1" style="margin: 10px auto;"> <thead> <tr> <th></th> <th>内容</th> <th>单位</th> </tr> </thead> <tbody> <tr> <td>X、Y、Z成分</td> <td>设定对角点1的位置坐标</td> <td>mm</td> </tr> <tr> <td>A、B、C成分</td> <td>设定姿势领域</td> <td>deg</td> </tr> <tr> <td>L1、L2成分</td> <td>设定附加轴领域</td> <td>mm、deg</td> </tr> </tbody> </table><br><注意><br>•请设定通过AREA*CS指定的坐标系中的值。<br>•不进行姿势确认时，请将A、B、C坐标设定为-360。<br>•使用附加轴时，请设定L1.L2成分。<br>•关于X、Y、Z、L1、L2成分，即使与AREA*P2对换设定，定义的领域也不会发生变化。      |        | 内容  | 单位  | X、Y、Z成分 | 设定对角点1的位置坐标  | mm       | A、B、C成分        | 设定姿势领域     | deg  | L1、L2成分               | 设定附加轴领域 | mm、deg | (X, Y, Z, A, B, C, L1, L2) =<br>0.0, 0.0, 0.0, -360.0,<br>-360.0, -360.0, 0, 0 |
|                               | 内容                                 | 单位         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| X、Y、Z成分                       | 设定对角点1的位置坐标                        | mm         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| A、B、C成分                       | 设定姿势领域                             | deg        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| L1、L2成分                       | 设定附加轴领域                            | mm、deg     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
|                               | AREA*P2<br>*为1~32                  | 实数 8       | 设定使用者定义领域*的对角点2的位置坐标以及姿势数据・附加轴坐标。从第1要素开始，按X、Y、Z、A、B、C、L1、L2的顺序定义。<br><table border="1" style="margin: 10px auto;"> <thead> <tr> <th></th> <th>内容</th> <th>单位</th> </tr> </thead> <tbody> <tr> <td>X、Y、Z成分</td> <td>设定对角点2的位置坐标</td> <td>mm</td> </tr> <tr> <td>A、B、C成分</td> <td>设定姿势领域</td> <td>deg</td> </tr> <tr> <td>L1、L2成分</td> <td>设定附加轴领域</td> <td>mm、deg</td> </tr> </tbody> </table><br><注意><br>•请设定通过AREA*CS设定的坐标系中的值。<br>•不进行姿势确认时，请将A、B、C坐标设定为+360。<br>•使用附加轴时，请设定L1.L2成分。<br>•关于X、Y、Z、L1、L2成分，即使与AREA*P1对换设定，定义的领域也不会发生变化。      |        | 内容  | 单位  | X、Y、Z成分 | 设定对角点2的位置坐标  | mm       | A、B、C成分        | 设定姿势领域     | deg  | L1、L2成分               | 设定附加轴领域 | mm、deg | (X, Y, Z, A, B, C, L1, L2) =<br>0.0, 0.0, 0.0, +360.0,<br>+360.0, +360.0, 0, 0 |
|                               | 内容                                 | 单位         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| X、Y、Z成分                       | 设定对角点2的位置坐标                        | mm         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| A、B、C成分                       | 设定姿势领域                             | deg        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| L1、L2成分                       | 设定附加轴领域                            | mm、deg     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| 使用者定义领域<br>请参照「5.8关于使用者定义领域」。 | AREA*ME<br>*为1~32                  | 整数 1       | 指定使用者定义领域*有效的机器号码。<br>机器号码的设定为1~3，通常设定为1。<br>0: 无效<br>1: 机器1（通常的设定）<br>2: 机器2<br>3: 机器3<br>（设定为0时，不进行领域确认）                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0      |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
|                               | AREA*AT<br>*为1~32                  | 整数 1       | 指定机器人进入使用者定义领域内时的举动<br>0: 无效<br>1: 信号输出（专用输出・状态变量输出）<br>2: 报警输出<br><设定内容><br><table border="1" style="margin: 10px auto;"> <thead> <tr> <th>设定</th> <th>领域内</th> <th>领域外</th> </tr> </thead> <tbody> <tr> <td rowspan="2">信号输出</td> <td>专用输出信号开启（※1）</td> <td>专用输出信号关闭</td> </tr> <tr> <td>状态变量的对应位开启（※2）</td> <td>状态变量的对应位关闭</td> </tr> <tr> <td>报警输出</td> <td>因报警输出导致的停止（发生H2090报警）</td> <td>-</td> </tr> </tbody> </table><br>※1: 系统状态变量(M_Uar32、M_Uar)<br>※2: 专用输入输出的信号号码通过USRAREA设定<br><注意><br>选择了报警输出时，将会忽略姿势领域确认和附加轴确认领域，而仅在位置领域进行确认。 | 设定     | 领域内 | 领域外 | 信号输出    | 专用输出信号开启（※1） | 专用输出信号关闭 | 状态变量的对应位开启（※2） | 状态变量的对应位关闭 | 报警输出 | 因报警输出导致的停止（发生H2090报警） | -       | 0（无效）  |                                                                                |
| 设定                            | 领域内                                | 领域外        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| 信号输出                          | 专用输出信号开启（※1）                       | 专用输出信号关闭   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
|                               | 状态变量的对应位开启（※2）                     | 状态变量的对应位关闭 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |
| 报警输出                          | 因报警输出导致的停止（发生H2090报警）              | -          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |     |     |         |              |          |                |            |      |                       |         |        |                                                                                |

| 参数                                                                                                       | 参数名                                                                       | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                  | 出货时设定值                                                                       |
|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 自由平面限制<br>请参照「5.9 自由平面限制」。                                                                               | 在自由平面中设置的速度比例限制值。<br>由3点的坐标制作平面且将不包含原点的一侧的领域设为动作范围外。使用下列的3种参数，最大可以设定8个限制。 |            |                                                                                                                                                                                                                                                                                                                                       |                                                                              |
|                                                                                                          | SFC*P<br>*为1~8                                                            | 实数 9       | 指定做成平面的 3点<br>X1,Y1,Z1: 平面的原点位置<br>X2,Y2,Z2: 平面的X轴上的位置<br>X3,Y3,Z3: 平面的X-Y平面上的+Y方向的位置                                                                                                                                                                                                                                                 | (X1,Y1,Z1,X2,Y2,Z2,<br>X3,Y3,Z3)=0.0,0.0,<br>0.0,0.0,0.0,0.0,0.0,<br>0.0,0.0 |
|                                                                                                          | SFC*ME<br>*为1~8                                                           | 整数 1       | 指定使自由平面限制有效的机器号码。<br>机器号码的设定为1~3, 通常设定为1。                                                                                                                                                                                                                                                                                             | 0                                                                            |
|                                                                                                          | SFC*AT<br>*为1~8                                                           | 整数 1       | 指定已设定的自由平面限制的有效/无效。<br>0: 无效(初始值)<br>1: 有效(动作可能领域为机器人的坐标原点侧)<br>-1: 有效(动作可能领域为无机器人的坐标原点侧)                                                                                                                                                                                                                                             | 0(无效)                                                                        |
| 圆筒限制<br>※ 仅在以下的 S/W Ver 与机型中有效。<br>S/W Ver.<br>F-Q 系列: R4c 以上<br>F-D 系列: S4c 以上<br><br>机型:<br>RH-3FHR 系列 | 可以在以J1轴为中心的圆筒区域对动作加以限制。<br>将以J1轴为中心的半径设定为参数。                              |            |                                                                                                                                                                                                                                                                                                                                       |                                                                              |
|                                                                                                          | MECAR                                                                     | 实数 2       | 参数的构成: (半径、J2轴限制角度)<br>将半径设定为第1要素。单位: mm (小数点以后2位)<br>半径为0时, 本功能无效。<br>半径为负值、或者为伸展机械臂也无法够到的距离时, 输入时会发生错误。<br>以半径定义的圆筒限制将会限制J2轴的动作。J2轴的限制角度将自动被计算, 并以绝对值输入第2要素。(仅读取时无法更改)<br>※与关节动作范围参数(MEJAR)相比, 较小的一方受到动作限制。<br><br><br><br>注) 本功能限于RH-3FHR系列机型。 | 0, 0                                                                         |
| 待避点位置                                                                                                    | JSAFE                                                                     | 实数 8       | 指定待避点位置。<br>依据机器人程序Mov P_SAFE的执行或输入外部信号的SAFEPOS信号, 往待避点位置动作xz<br>(J1,J2,J3,J4,J5,J6,J7,J8) 单位: deg                                                                                                                                                                                                                                    | 依机型会有所不同。                                                                    |
| 机械 限位器                                                                                                   | MORG                                                                      | 实数 8       | 指定机械限位器原点位置。<br>(J1,J2,J3,J4,J5,J6,J7,J8) 单位: deg                                                                                                                                                                                                                                                                                     | 依机型会有所不同。                                                                    |
| 使用者指定原点                                                                                                  | USERORG                                                                   | 实数 8       | 指定使用者指定原点位置。通常没有必要设定。<br>(J1,J2,J3,J4,J5,J6,J7,J8) 单位: deg                                                                                                                                                                                                                                                                            | 依机型会有所不同。                                                                    |
| 选择特异点接近警告功能<br>请参照「5.17 关于特异点接近警告」。                                                                      | MESNGLSW                                                                  | 整数 1       | 指定特异点接近警告的有效 / 无效。<br>(无效 / 有效 = 0 / 1)<br>本参数被设定为“有效”的情况下, 参数: 即使BZR(蜂鸣的ON/OFF)设定为OFF, 本警告音也会响起。                                                                                                                                                                                                                                     | 1(有效)                                                                        |

| 参数        | 参数名     | 排列数<br>文字数 | 内容说明                                                                                                                                                            | 出货时设定值   |
|-----------|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| JOG 设定    | JOGJSP  | 实数 3       | 指定关节JOG的前进量及单步运行的速度。<br>(定寸H, 定寸L, 最大速度比例)<br>定寸H: JOG速度High表示时的前进量 单位: deg<br>定寸L: JOG速度Low表示时的前进量 单位: deg<br>最大速度比例: 以OP 速度比例×最大速度比例动作。                       | 每个机型的设定值 |
|           | JOGPSP  | 实数 3       | 指定直交JOG的前进量及单步运行的速度。<br>(定寸H, 定寸L, 最大速度比例)<br>定寸H: JOG速度High表示时的前进量 单位: mm<br>定寸L: JOG速度Low表示时的前进量 单位: mm<br>最大速度比例: 以OP 速度比例×最大速度比例动作。<br>最大速度超过250mm/s的话无法动作。 | 每个机型的设定值 |
| JOG 速度限制值 | JOGSPMX | 实数 1       | 限制示教模式时机器人动作速度。单位: mm/s<br>即使设定比250大的值, 最大值也会被限制在250。                                                                                                           | 250.0    |

| 参数                                                                                   | 参数名                   | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 出货时设定值                                  |
|--------------------------------------------------------------------------------------|-----------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 工件坐标系                                                                                | WKnCORD<br>n 为 1 ~ 8  | 实数 6       | <p>是工件坐标的坐标值。<br/>(X, Y, Z, A, B, C) 单位: mm或度</p> <p>作为工件JOG操作时的基准坐标或工件坐标数据使用。<br/>在软件版本R5(F-Q系列)/S5(F-D系列)以上, 作为Ex-T控制中的控制点(Ex-T坐标)使用。<br/>关于Ex-T请参照第652页的「7.5关于Ex-T控制」。</p> <p>作为工件坐标数据使用时, 有效的轴成分因机器人机型而有所差异。请参照第434页的「5.7关于标准基本坐标」。</p> <p>在示教单元中的工件坐标设定(定义)操作中定义的坐标值会被设定为工件坐标, 而通过在本参数中输入坐标值也可以定义工件坐标。此时, 根据以下示教将用于定义工件坐标的3点(参数: WKnWO, WKnWX, WKnWY(n为1~8))的各坐标值清零。</p>  <p>注) 在基本变换为初始值的状态(示教坐标系和基本坐标系一致的状态)下进行工件坐标的示教, 将更方便管理。尤其在定义多个工件坐标时, 建议以此来防止发生混乱</p> | (0.00, 0.00, 0.00,<br>0.00, 0.00, 0.00) |
|                                                                                      | WKnWO<br>n 为 1 ~ 8    | 实数 3       | <p>是用于定义工件坐标的示教位置, 也是工件坐标的原点位置。<br/>(对应于示教单元中的示教操作的"WO"。参照上图)</p> <p>(X, Y, Z) 单位: mm</p> <p>注) 仅输入该坐标值无法定义工件坐标。请通过示教单元工件坐标设定画面的[定义]操作或RTToolBox2的工件坐标参数画面的[写入]操作进行工件坐标的计算处理。</p>                                                                                                                                                                                                                                                                                                                                                                      | (0.00, 0.00, 0.00)                      |
|                                                                                      | WKnWX<br>n 为 1 ~ 8    | 实数 3       | <p>是用于定义工件坐标的示教位置, 也是工件坐标的+X轴上的位置。(对应于示教单元中的示教操作的"WX"。参照上图)</p> <p>(X, Y, Z) 单位: mm</p> <p>注) 仅输入该坐标值无法定义工件坐标。请通过示教单元工件坐标设定画面的[定义]操作或RTToolBox2的工件坐标参数画面的[写入]操作进行工件坐标的计算处理。</p>                                                                                                                                                                                                                                                                                                                                                                        | (0.00, 0.00, 0.00)                      |
|                                                                                      | WKnWY<br>n 为 1 ~ 8    | 实数 3       | <p>是用于定义工件坐标的示教位置, 也是工件坐标的X-Y平面上的+Y轴侧的位置。(对应于示教单元中的示教操作的"WY"。参照上图)</p> <p>(X, Y, Z) 单位: mm</p> <p>注) 仅输入该坐标值无法定义工件坐标。请通过示教单元工件坐标设定画面的[定义]操作或RTToolBox2的工件坐标参数画面的[写入]操作进行工件坐标的计算处理。</p>                                                                                                                                                                                                                                                                                                                                                                 | (0.00, 0.00, 0.00)                      |
| 工件 JOG 动作模式<br>※ 仅在以下的 S/W<br>Ver 中有效。<br>S/W Ver.<br>F-Q 系列: R5 以上<br>F-D 系列: S5 以上 | WKnJOGMD<br>n 为 1 ~ 8 | 整数 1       | <p>对每个工件坐标设定工件JOG操作中的动作模式。</p> <p>0: 工件JOG (A、B、C的动作为与工件坐标的X轴、Y轴、Z轴平行旋转。控制点位置不变。)</p> <p>1: Ex-T JOG (A、B、C的动作为以工件坐标的X轴、Y轴、Z轴为中心移动控制点并同时旋转。)</p>                                                                                                                                                                                                                                                                                                                                                                                                       | 0                                       |

| 参数                                                       | 参数名                                               | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 出货时设定值          |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
|----------------------------------------------------------|---------------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------------------------------------|--------------|------------------------|----|------------------------|-------------------|------------------------|----------------------|---------------------|--------------------|
| 中断时 JOG 前进后的自动回归设定<br><br>请参照「5.10 中断时的 JOG 前进后的自动回归设定」。 | RETPATH                                           | 整数 1       | 设定在程序运行中，依据停止使程序中断后再以JOG动作等移动的情况下，重新启动时会回到中断位置重新启动程序。<br>使功能无效的情况下，以从现在的位置到下一个点为止已设定的动作指令移动。不会返回到中断点。<br>0: 无效<br>1: 以关节插补返回<br>2: 以直线插补返回<br>注) 以直线插补返回时，会以3轴直交插补做抄近路动作。<br>注) 在圆弧插补指令和Mva指令中，即使被设定为0，也会变成和1一样的动作。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1               |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
| 重力方向<br><br>※ 本功能可以使用在 RV-SQ/SD 系列。                      | MEGDIR                                            | 实数 4       | 对应安装姿势，将作用在机器人上的重力加速度及大小设定在机器人坐标的 X 轴、Y 轴、Z 轴。(单位: mm/秒 <sup>2</sup> )<br>要素数为 4 个，由左开始的顺序安装姿势，X 轴的动力加速度、Y 轴的动力加速度、Z 轴的重力加速度。<br><table border="1" style="margin: 10px auto;"> <thead> <tr> <th>安装姿势</th> <th>设定值<br/>(安装姿势, X 轴的重力加速度,<br/>Y 轴的重力加速度, Z 轴的重力加速度)</th> </tr> </thead> <tbody> <tr> <td>放置地板<br/>(标准)</td> <td>( 0.0, 0.0, 0.0, 0.0 )</td> </tr> <tr> <td>壁挂</td> <td>( 1.0, 0.0, 0.0, 0.0 )</td> </tr> <tr> <td>垂吊<sup>注1)</sup></td> <td>( 2.0, 0.0, 0.0, 0.0 )</td> </tr> <tr> <td>任意的姿势<sup>注2)</sup></td> <td>( 3.0, **, **, ** )</td> </tr> </tbody> </table> 注 1) RH-3FHR 系列应在第 1 要素为“0.0”的情况下直接使用。<br>注 2) 在 ** 里代入数值。<br>关于任意的姿势的设定值，请参照下列范例。<br>重力加速度的设定例如下所示。<br>例) 机器人往前方 30 度倾斜姿势的情况 (参照下图)<br>$X$ 轴方向重力加速度 ( $Xg$ ) = $9.8 \times \sin(30 \text{度}) = 4.9$<br>$Z$ 轴方向重力加速度 ( $Zg$ ) = $9.8 \times \cos(30 \text{度}) = 8.5$<br>但是，因为机器人坐标的 $Z$ 轴方向相反，所以会变成 -8.5。<br>$Y$ 轴方向重力加速度 ( $Yg$ ) = 0.0<br>因此，设定值会变成 (3.0, 4.9, 0.0, -8.5)。 <div style="text-align: center;">  </div> | 安装姿势            | 设定值<br>(安装姿势, X 轴的重力加速度,<br>Y 轴的重力加速度, Z 轴的重力加速度) | 放置地板<br>(标准) | ( 0.0, 0.0, 0.0, 0.0 ) | 壁挂 | ( 1.0, 0.0, 0.0, 0.0 ) | 垂吊 <sup>注1)</sup> | ( 2.0, 0.0, 0.0, 0.0 ) | 任意的姿势 <sup>注2)</sup> | ( 3.0, **, **, ** ) | 0.0, 0.0, 0.0, 0.0 |
| 安装姿势                                                     | 设定值<br>(安装姿势, X 轴的重力加速度,<br>Y 轴的重力加速度, Z 轴的重力加速度) |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                 |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
| 放置地板<br>(标准)                                             | ( 0.0, 0.0, 0.0, 0.0 )                            |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                 |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
| 壁挂                                                       | ( 1.0, 0.0, 0.0, 0.0 )                            |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                 |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
| 垂吊 <sup>注1)</sup>                                        | ( 2.0, 0.0, 0.0, 0.0 )                            |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                 |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
| 任意的姿势 <sup>注2)</sup>                                     | ( 3.0, **, **, ** )                               |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                 |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |
| 抓手初始状态<br><br>请参照「5.13 关于抓手初始状态」。                        | HANDINIT                                          | 整数 8       | 设定电源投入时的气动抓手I/F的输出。<br>本参数为指定在抓手的尖端的抓手专用信号900号台的电源开启时的初始值。<br>将抓手控制以通用IO (10000~18191号以外) 控制的情况下设定 (在 HANDTYPE 参数900号以外的信号指定) 的电源开启时的初始状态，请不要使用此HANDINIT参数，请使用ORS* 参数。<br>以此ORS* 参数设定的值会变成电源开启时的信号初始值。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 1,0,1,0,1,0,1,0 |                                                   |              |                        |    |                        |                   |                        |                      |                     |                    |

| 参数                                                          | 参数名                                                                                                                     | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                          | 出货时设定值                                                                                       |
|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| 抓手种类<br>请参照「5.12 关于抓手种类」。                                   | HANDTYPE                                                                                                                | 字符串 8      | 设定单/双线螺管的抓手种类和输出信号号码<br>(D:双线螺管 S:单线螺管)。设定连接在抓手的信号号码D900的情况下, 信号号码会输出900和901。<br>D(双线螺管)的情况下, 请设定使信号不会重迭。                                                                                                                                                                     | CR750/CR751:<br>D900,D902,D904,D<br>906<br>,,,,<br>CR760:<br>D764,D766,D768,D<br>770<br>,,,, |
| 抓手输入输出类型                                                    | HIOTYPE                                                                                                                 | 整数 1       | 设定内置气动抓手接口的电气规格(源型/漏形)。请根据所使用的电磁阀等的规格进行设定。<br>0: 源型/1: 漏形                                                                                                                                                                                                                     | 1                                                                                            |
| 抓手、工件条件 (在最佳加减速、冲突检知使用)<br>请参照「5.16 关于抓手、工件条件设定 (最佳加减速设定)」。 | 设定程序在Oadl On时的抓手条件・工件条件。<br>最大可设定到8, 使用何种条件的组合, 可依据LoadSet指令选择。<br>注) 请正确设定抓手、工件条件。如果搭载负载的设定低于实际负载, 有可能会缩短机器人机构部件的使用寿命。 |            |                                                                                                                                                                                                                                                                               |                                                                                              |
|                                                             | HNDDAT0                                                                                                                 | 实数 7       | 设定抓手的初始条件。(以TOOL坐标指定。)<br>电源一开启后, 此设定值马上被采用。使用冲突检知在JoD操作时, 使用前请设定实际的抓手条件。没有设定的情况下会发生错误检知的情况。<br><br>(重量,大小Z,大小Y,大小Z,重心X,重心Y,重心Z)<br>单位: Kg,mm                                                                                                                                 | 依机型会有所不同。                                                                                    |
|                                                             | HNDDAT*<br>*为1~8                                                                                                        | 实数 7       | 设定抓手的条件。(以TOOL坐标指定。)<br>(重量,大小Z,大小Y,大小Z,重心X,重心Y,重心Z)<br>单位: Kg,mm                                                                                                                                                                                                             | 标准负载<br>,0.0,0.0,0.0,0.0,0.0,<br>0.0                                                         |
|                                                             | WRKDAT0                                                                                                                 | 实数 7       | 设定工件的初始值条件。(以TOOL坐标指定。)<br>电源一开启后, 此设定值马上被采用。<br>(重量,大小Z,大小Y,大小Z,重心X,重心Y,重心Z)<br>单位: Kg,mm                                                                                                                                                                                    | 0.0,0.0,0.0,0.0,0.0,<br>0.0,0.0                                                              |
|                                                             | WRKDAT*<br>*为1~8                                                                                                        | 实数 7       | 设定工件的条件。(以TOOL坐标指定。)<br>(重量,大小Z,大小Y,大小Z,重心X,重心Y,重心Z)<br>单位: Kg,mm                                                                                                                                                                                                             | 0.0,0.0,0.0,0.0,0.0,<br>0.0,0.0                                                              |
|                                                             | HNDHOLD*<br>*为1~8                                                                                                       | 整数 2       | 在执HOPEN*和HCLOSE*时, 设定工件的把持/未把持。<br>(OPEN时的设定, CLOSE时的设定)<br>(未把持/把持=0/1)                                                                                                                                                                                                      | 0,1                                                                                          |
| 最佳加减速设定<br>请参照「5.16 关于抓手、工件条件设定 (最佳加减速设定)」。                 | ACCMODE                                                                                                                 | 整数 1       | 设定最佳加减速模式的电源一开启的有效/无效。<br>(无效/有效=0/1)                                                                                                                                                                                                                                         | 1                                                                                            |
| 最佳加减速补偿率                                                    | JADL                                                                                                                    | 实数 8       | 设定最佳加减速时的加减速补偿率的初始值(电源投入时的值)(%)。这是在最佳加减速控制计算对应加减速的比例。将本设定值调大可以高速动作, 但是, 就这样动作一定时间后会发生过负载报警及过热报警。报警出现的情况下, 请将设定值调小。初期状态会设定在预防过负荷报警及过热报警发生的初始值。加速度、减速度两边皆适用。<br><br>※所谓过负载报警<br>为防止因为在高速回转的热, 使电机破损, 负载率在某个值以上的时候会发生报警。<br>※所谓过热报警<br>为防止因为在高速回转的热, 使位置检测器破损, 温度在某个值以上的时候会发生报警。 | 依机型会有所不同。                                                                                    |

| 参数                                                                                   | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                    | 出货时设定值                                                                             |
|--------------------------------------------------------------------------------------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 加减速最佳化模式选择<br>※ 仅限 RH-F 系列<br>(RH-3FHR 系列除外)                                         | MAPMODE  | 整数 1       | 选择将 RH-F 系列 (RH-3FHR 系列除外) 中的与轴 (J3 轴) 高度对应的加减速最佳化模式设为标准加减速模式还是高速加减速模式。<br>0: 标准加减速模式/1: 高速加减速模式<br><br>初始设定为标准加减速模式, 在该模式下可在运行时将轴 (J3 轴) 前端振动 (含残留振动) 控制在最小限度。在用户使用机器人进行作业时, 可在该振动不会造成恶劣影响的范围内, 选择高加减速模式, 使机器人进行高速动作。关于模式的详细内容, 请参照另一手册「标准规格书」。                                        | 0                                                                                  |
| 最佳速度控制调整系数<br><br>※ 仅在以下的 S/W Ver 中有效。<br>S/W Ver.<br>F-Q 系列: R5 以上<br>F-D 系列: S5 以上 | OPTOVC   | 实数 1       | 设定最佳速度控制的调整系数。<br>设定范围: 0.30~1.00<br><br>最佳速度控制是指, 在不通过 Spd 指令进行速度指定的直线插补动作/圆弧插补动作中, 以最佳速度执行动作的功能。可以通过增大设定值使机器人高速动作, 但在特异点旁边等机器人动作姿势变化较大时, 容易发生超过速度极限错误。此外, 发生超过速度极限错误时, 可以通过降低本设定值抑制错误的发生, 但机器人的动作速度会变慢。<br><br>※ 超过速度极限错误是指<br>错误号码为 H213n 的错误。(n 表示轴号码 1~8)<br>对电机的速度指令超过允许值时将发生该错误。 | 依机型会有所不同。                                                                          |
| 速度调整插补功能开关<br>※ 仅在以下机型中有效。<br>机型: RH-3FHR 系列                                         | SPDOPT   | 整数 1       | 设定速度调整插补功能的有效/无效。<br>1: 有效 (接通电源时, 速度调整插补功能为有效)<br>0: 无效 (接通电源时, 速度调整插补功能为无效)<br>1或0时, 可以在程序中使用 SpdOpt 指令切换速度调整插补功能的有效/无效。<br><br>注) 本功能限于 RH-3FHR 系列机型。                                                                                                                                | RH-3FHR 系列<br>1<br><br>其他机型<br>0                                                   |
| 冲突检知                                                                                 | COL      | 整数 3       | 定义冲突检知的使用可 / 不可和电源一开启后的有效 / 无效。<br>要素 1: 冲突检知功能的使用可 (1) / 不可 (0)<br>要素 2: 作为自动运行时初始状态有效 (1) / 无效 (0)<br>要素 3: JOG 中设为有效 (1) / 无效 (0) / NOERR 模式 (2)<br><br>在 NOERR 模式即使检知到冲突也不会有报警显示。只会使伺服关闭。当因为频繁的报警发生使操作困难的情况下, 请使用 NOERR 模式。<br>自动运行 (含位置跳转、单步运行) 以外时, 依存 JOG 操作用 (要素 3) 的设定。        | RH-3FH/6FH/12FH/<br>20FH 系列:<br>1,0,1<br>RV-F 系列:<br>0,0,1<br>RH-3FHR 系列:<br>1,1,1 |
| 检知 LEVEL                                                                             | COLLVL   | 整数 8       | 设定在自动运行时, 各关节轴的各轴的检知 LEVEL (敏感度) 的初始值。<br>设定范围: 1 ~ 500 单位: %<br>※ 在设定范围外时, 会以最近的设定范围内的值处理。                                                                                                                                                                                             | 依机型会有所不同。                                                                          |
| JOG 时的检知 LEVEL                                                                       | COLLVLJG | 实数 8       | 设定在 JOG 操作时 (含中断中) 各关节轴的检知 LEVEL (敏感度)。单位: (%)<br>要调高检知 LEVEL (敏感度) 的时候, 请将数值调小。<br>JOG 操作中没有发生冲突却有冲突检知报警的情况下, 请将值调大。<br>设定范围: 1 ~ 500 单位: %<br>※ 在设定范围外时, 会以最近的设定范围内的值处理。                                                                                                              | 依机型会有所不同。                                                                          |

| 参数                   | 参数名     | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                            | 出货时设定值    |
|----------------------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 腕部旋转角 (A 轴)<br>坐标系选择 | RCD     | 整数 1       | 切换垂直 5 轴型机器人的腕部旋转角 (直角坐标 A 轴) 的控制与显示方式。在其他类型的机器人中不起作用。<br>2: 一般角方式<br>如果动作的前后 A 轴的值都相同, 则控制 A 轴, 以保持抓手的姿势。但是, 由于腕部的倾斜 (直交坐标 B 轴), 有可能无法保持抓手的姿势。一般不更改出厂设定, 按该方式使用。                                                                                                                                                                               | 2 (一般角方式) |
| 设定暖机运行模式             | WUPENA  | 整数 1       | 指定暖机运行模式的有效 / 无效。<br>0: 无效<br>1: 有效<br>注) 设定上述以外的值的情况下, 全部会视为无效。<br>注) 多机器的情况下, 会设定在各机器上。                                                                                                                                                                                                                                                       | 0 (无效)    |
| 暖机运行模式对象轴            | WUPAXIS | 整数 1       | 将在暖机运行模式中, 作为控制对象的关节轴, 依据位的 ON/OFF 以 16 进制 (下位 bit 开始 J1、J2,...) 进行指定。<br>bit ON : 对象轴<br>bit OFF: 非对象轴<br>将在低温下的操作时会发生误差过大报警的关节轴视为对象轴。<br>注) 不存在的轴的位即使开启, 也不会变成对象轴。<br>注) 没有对象轴的情况下, 暖机运行模式会变成无效。<br>注) 多机器的情况下, 会设定在各机器上。                                                                                                                     | 0         |
| 暖机运行模式控制时间           | WUPTIME | 整数 2       | 指定在暖机运行模式的处理时间。<br>(有效时间、重新开始时间) 单位: 分<br><br>有效时间: 指定机器人在暖机运行状态且将动作调降的动作时间。<br>(设定范围: 0 ~ 60)<br>重新开始时间: 指定在暖机运行状态被解除后, 对象轴继续停止的时候, 到再重新开始暖机运行状态的时间。<br>(设定范围: 1 ~ 1440)<br>注) 指定在设定范围外的值的情况下, 会以最近的设定范围内的值处理。<br>注) 有效时间为 0 分的情况下, 暖机运行模式会成为无效。<br>注) 多机器的情况下, 会设定在各机器上。                                                                      | 1,60      |
| 暖机运行速度比例             | WUPOVRD | 整数 2       | 执行暖机运行状态相关的速度设定。<br>(初始值、值固定时间的比例) 单位: %<br><br>初始值: 指定暖机运行状态时, 适用在动作速度上的速度比例的初始值 (暖机运行速度比例)。<br>(设定范围: 50 ~ 100)<br>固定时间的比例: 暖机运行模式时, 动作速度的速度比例会从初始值开始没有变化时间, 指定此比例在有效时间上。<br>(设定范围: 0 ~ 50)<br><br>暖机运行速度比例的值和各要素的设定值的对应如下图所示。<br><br><br>注) 指定在设定范围外的值的情况下, 会以最近的设定范围内的值处理。<br>注) 速度比例的初始值为 100% 的情况下, 暖机运行模式会变成无效。<br>注) 多机器的情况下, 会设定在各机器上。 | 70, 50    |

| 参数                       | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                     | 出货时设定值                                                                                      |
|--------------------------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 伺服弹性报警的功能设定              | CMPERR   | 整数 1       | <p>依据本参数的设定, 能使报警 2710 ~ 2740 (以伺服弹性控制所生成的位置指令异常所发生的报警) 不要发生。</p> <p>1: 报警发生有效<br/>0: 报警发生无效</p> <p>对象及发生的报警内容如下所示。</p> <p>2710: 从原来的位置指令的位移过大。<br/>2720: 伺服弹性指令的超过关节限制<br/>2730: 伺服弹性指令的超速<br/>2740: 伺服弹性指令的坐标变换报警</p> <p>发生这些异常, 因为弹性伺服控制不是在正常的功能状态, 因此为了不使报警发生必须修正原来的示教位置及程序的内容。本参数变更为 0 (报警发生无效), 只在依据报警, 即使不中断运行也可以判断运行上的问题时使用。</p> | 1<br>(报警发生有效)                                                                               |
| Cmp Jnt 用电流限制等级          | CMPJCLL  | 整数 1       | <p>更改关节坐标系的弹性模式 (Cmp Jnt 指令) 下各轴电机的电流限制等级。</p> <p>设定范围: 1(限制高) ~ 10(限制低)</p> <p>在关节坐标系的弹性模式下动作时发生误差过大 1 错误 (H096n) 的情况下, 增大参数 CMPJCLL 的设定值可以抑制错误的发生。(更改后的设定值将从下一次执行 Cmp Jnt 指令开始反映。)</p>                                                                                                                                                 | 1                                                                                           |
| 过负载检知 Level 的最佳化         | OLTMX    | 整数 1       | <p>设定机器人使用环境中的周围温度的上限值。</p> <p>以该设定值为基础使机器人动作时的过负载检知 Level 最佳化。(单位: °C)</p> <p>设定范围: 0 ~ 40</p> <p>关于与本参数相关的功能的详细内容, 请参照第 479 页的「5.22 关于过负载 Level 的最佳化」。</p>                                                                                                                                                                             | RH-3FH/6FH/12FH/<br>20FH: 40<br>RV-2F 系列: 40<br>RV-4F/7F/13F/20F/<br>50F 系列、RH-<br>3FHR: 30 |
| 电池消耗时间报警发生时间间隔           | ITBATERR | 整数 1       | <p>指定电池消耗时间报警的发生时间间隔。</p> <p>(单位: 小时)</p> <p>设定范围: 1 ~ 336。</p> <p>设定值小于 1 时视作 1, 大于 336 时视作 336。</p>                                                                                                                                                                                                                                    | 24                                                                                          |
| 与 PLC 的时间同步 (仅限 CR7xx-Q) | TIMESYNC | 整数 1       | <p>选择是否使机器人控制器的设定时间和 PLC 的设定时间同步。</p> <p>(同步 / 不同步 =1/0)</p>                                                                                                                                                                                                                                                                             | 0                                                                                           |
| WTHIF 指令规格               | WTHFUNC  | 整数 1       | <p>指定 WthIf 指令的停止类型。</p> <p>0: 停止类型 1<br/>1: 停止类型 2</p> <p>关于停止类型, 请参照 Def Act 指令的说明。</p>                                                                                                                                                                                                                                                | 0                                                                                           |
| Cnt 无效时的动作指令结束条件指定       | MVTERM   | 整数 1       | <p>指定 Cnt 无效时的动作指令结束条件。</p> <p>F 系列 (RV-2F/2FL、RH-3FHR 除外) 中, 为了提高 Cnt0 时的目的位置到达性, 更改了动作指令的结束条件。F 系列之前的机型 (SD/SQ 系列) 中创建的程序直接用于 F 系列 (RV-2F/2FL、RH-3FHR 除外) 时, 动作时间可能会变长。动作速度优先时, 请更改本参数, 在需要定位的位置上使用 Fine P 或 Dly 指令等。</p> <p>0: 速度指令输出结束<br/>1: 相当于 Cnt 1<br/>2: 速度指令生成结束</p>                                                          | 0: F 系列<br>(RV-2F/2FL、<br>RH-3FHR 除外)<br>2: RV-2F/2FL、<br>RH-3FHR、SD/<br>SQ 系列              |
| 退避点复归模式                  | ESCMODE  | 整数 1       | <p>退避点复归时, 选择将当前位置设定为工具定义位置还是机械接口原点。</p> <p>0: 设定为 TCP 位置 (工具定义位置)<br/>1: 设定为机械接口原点</p>                                                                                                                                                                                                                                                  | 0                                                                                           |
| 控制器 CPU 运行模式             | DRVMODE  | 整数 1       | <p>使用软件进行模拟时, 指定是否需要与机器人连接。</p> <p>F-Q 系列时, 指定是否需要与驱动模块连接。</p> <p>0: 通常模式<br/>(需要与机器人或驱动模块连接)<br/>1: 机器人伺服分离模式<br/>(不需要与机器人或驱动模块连接)</p>                                                                                                                                                                                                  | 0                                                                                           |

## 5.2 信号参数

信号关连用的参数。专用信号相关请参照第 516 页的「6.3 专用输出」。

表 5-2: 信号关连参数一览表

| 参数                            | 参数名        | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 出货时设定值     |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
|-------------------------------|------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------|--|--|----|-----|------|---|---|---|---|---|---|---|----|---|---|----|---|---|---|----|----|---|----|---|---|---|----|---|----|---|----|----|---|---|----|----|----|------------------|
| 专用输入输出信号                      |            |            | 关于专输入输出信号的参数请参照第516页的「6.3专用输出」。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 在起动信号的输入时读取从数值输入的程序号码         | PST        | 整数 1       | 通常从外部输入信号选择程序是于数值输入信号 (IODATA) 中设定程序号码, 并在程序选择信号 (PRGSEL) 确定号码, 以起动信号动作。使本功能为有效的话, 程序选择信号会变成不需要, 在起动信号 (START) 开启的时候, 从数值输入信号 (IODATA) 读取程序号码。<br>(功能无效/有效=0/1)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0 (无效)     |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 停止输入 B 触点指定                   | INB        | 整数 1       | 对停止输入信号的A触点/B触点进行切换。<br>(A触点/B触点=0/1)<br>切换的对象为, 专用输入信号: STOP、STOP2和控制器专用停止输入: SKIP。<br>※关于SKIP的详细内容, 请参照另一手册「标准规格书」或「特殊规格书」。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0 (A触点)    |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 机器人报警输出                       | ROBOTERR   | 整数 1       | <p>设定开放EMGOUT接头的机器人报警输出端子的报警 Level。</p> <p>在工厂出货时设定中, 无论发生何种报警Level都会开放。例如, 为了在忽略警告Level而发生了Low Level报警或High Level报警中的一种, 或是两者都发生时, 开放该输出端子, 需设定3。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2">设定值</th> <th colspan="3">发生报警 Level</th> </tr> <tr> <th>警告</th> <th>Low</th> <th>High</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>开放</td> </tr> <tr> <td>2</td> <td>-</td> <td>开放</td> <td>-</td> </tr> <tr> <td>3</td> <td>-</td> <td>开放</td> <td>开放</td> </tr> <tr> <td>4</td> <td>开放</td> <td>-</td> <td>-</td> </tr> <tr> <td>5</td> <td>开放</td> <td>-</td> <td>开放</td> </tr> <tr> <td>6</td> <td>开放</td> <td>开放</td> <td>-</td> </tr> <tr> <td>7</td> <td>开放</td> <td>开放</td> <td>开放</td> </tr> </tbody> </table> <p>关于EMGOUT接头的详细内容, 请参照以下说明书。<br/>使用说明书从控制器设置和基本操作到保养「2.2.5 外部紧急停止的连接」</p> | 设定值        | 发生报警 Level |  |  | 警告 | Low | High | 0 | - | - | - | 1 | - | - | 开放 | 2 | - | 开放 | - | 3 | - | 开放 | 开放 | 4 | 开放 | - | - | 5 | 开放 | - | 开放 | 6 | 开放 | 开放 | - | 7 | 开放 | 开放 | 开放 | 7 (任何报警Level均开放) |
| 设定值                           | 发生报警 Level |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
|                               | 警告         | Low        | High                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 0                             | -          | -          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 1                             | -          | -          | 开放                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 2                             | -          | 开放         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 3                             | -          | 开放         | 开放                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 4                             | 开放         | -          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 5                             | 开放         | -          | 开放                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 6                             | 开放         | 开放         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 7                             | 开放         | 开放         | 开放                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| CC-Link报警解除许可<br>(※只限CR7xx-D) | E7730      | 整数 1       | 即使附有 CC-Link 配件, 但是不接续 CC-Link 而使用控制器的情况下, 会发生报警 7730, 且会变成无法操作。此报警平常无法解除, 但是可以使用本参数, 暂时的解除报警 (可以暂时解除报警 / 不可以解除报警 = 1/0)<br>本参数在从示教单元或计算机支持软件变更后就会变成有效, 因此没有必要重新启动电源。相反的, 为了不储存本参数的变更内容, 请再度投入电源返回到 0 (变成无法解除)。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0 (报警解除不可) |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |

| 参数                                                | 参数名                               | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 出货时设定值                                               |      |   |           |   |           |   |           |         |
|---------------------------------------------------|-----------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|------|---|-----------|---|-----------|---|-----------|---------|
| 输出信号复位模式<br>请参照「5.14 关于输出信号复位模式」                  |                                   |            | 设定Clr指令及专用输入（OUTRESET）等的通用输出信号复位时的动作。在电源开启时，在此设定的模式也会被信号输出。<br>在下列的参数里，信号以232位单位设定。<br>（OFF / ON / 保持=0 / 1 / *）<br>注）设定为保持“*”的位在电源ON时变为OFF。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                      |      |   |           |   |           |   |           |         |
|                                                   | ORST0                             | 字符串 4      | 设定信号号码0~31。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 00000000,<br>00000000,<br>00000000,<br>00000000      |      |   |           |   |           |   |           |         |
|                                                   | ORST32<br>:<br>:<br>:<br>ORST8016 | 字符串 4      | 设定信号号码32~63。<br>:<br>设定信号号码8016~8047。<br>注）使用多重抓手时，使用输出信号716~723。<br>输出信号700~715为系统保留领域，因此无法设定。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 00000000,<br>00000000,<br>00000000,<br>00000000<br>: |      |   |           |   |           |   |           |         |
|                                                   | ORS10000<br>:<br>ORS18160         | 字符串 4      | 设定信号号码10000~10031。<br>:<br>设定信号号码18160~18191。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 00000000,<br>00000000,<br>00000000,<br>00000000<br>: |      |   |           |   |           |   |           |         |
| 复位的输出复位                                           | SLRSTIO                           | 整数 1       | 指定在程序复位时执行通用输出信号复位的功能。<br>（无效 / 有效=0 / 1）                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0（无效）                                                |      |   |           |   |           |   |           |         |
| 多 CPU 台数设定<br>（仅限 CR7xx-Q）                        | QMLTCPUN                          | 整数 1       | 在多CPU系统，设定在基本的基本单元安装的CPU单元台数。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 2                                                    |      |   |           |   |           |   |           |         |
| 多 CPU <sub>n</sub> 号机<br>设定高速通信领域<br>（仅限 CR7xx-Q） | QMLTCPUn<br>n=1 ~ 4               | 整数 4       | <p>在多CPU系统，设定在1~4号机的多CPU间高速通信功能的各CPU单元间执行受信点数。关于全CPU必须符合参数设定值。<br/>没有符合参数设定值的话，因为会在PLC CPU发生报警，因此请将各CPU的参数设定值符合。</p> <p>要素1：使用者自由领域的容量（K点）<br/>范围：1~14（最大※）<br/>※如下表依据CPU台数，最大值会有所不同。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CPU 台数</th> <th>设定范围</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>0 ~ 14K 点</td> </tr> <tr> <td>3</td> <td>0 ~ 13K 点</td> </tr> <tr> <td>4</td> <td>0 ~ 12K 点</td> </tr> </tbody> </table> <p>要素2：自动更新的点数（点）<br/>范围：0~14335<br/>因为机器人CPU不对应自动更新，因此自动更新的点数请设定为0。</p> <p>要素3：系统领域的容量（K点）<br/>范围：1或2</p> <p>要素4：和多CPU同步开始（1：执行、0：不执行）的机器人CPU因为在开始花费时间，因此以基本的1（同步），请不要改变设定。</p> | CPU 台数                                               | 设定范围 | 2 | 0 ~ 14K 点 | 3 | 0 ~ 13K 点 | 4 | 0 ~ 12K 点 | 1,0,1,1 |
| CPU 台数                                            | 设定范围                              |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                      |      |   |           |   |           |   |           |         |
| 2                                                 | 0 ~ 14K 点                         |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                      |      |   |           |   |           |   |           |         |
| 3                                                 | 0 ~ 13K 点                         |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                      |      |   |           |   |           |   |           |         |
| 4                                                 | 0 ~ 12K 点                         |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                      |      |   |           |   |           |   |           |         |

| 参数                           | 参数名               | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 出货时设定值   |
|------------------------------|-------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 多 CPU 输入偏置<br>(仅限 CR7xx-Q)   | QMLTCPUS          | 整数 1       | <p>设定CR7xx-Q控制器中对多CPU的机器人输入信号的偏置。</p> <p>以1K字为单位指定来自G10000的偏置,从指定的共有内存作为R/C的输入读取。混合其他支持iQ Platform的CPU(运动控制器CPU或NCCPU)使用、不使各CPU中使用的共有内存重复时,根据需要设定。</p> <p>设定值范围: -1~14(整数)</p> <p>(-1: 不使用 / 0~14 K字)</p> <p>(A) 设定为-1时,偏置值会自动根据安装的插槽固定下来(与以往版本[N4a以前]兼容)。</p> <p>(B) 设定为0~14时,根据值被设定为输入偏置。</p> <p>※参照下述「5.2.1关于多CPU输入偏置(仅限CR7xx-Q控制器)」案例(A)、(B)。</p> <p>此外,当连接多台机器人,且该参数都设为相同值(-1以外)时,也可使PLC几乎同时向多台机器人输入相同的信号状态。</p> <p>关于多CPU系统,请参照QCPU用户手册(多CPU系统篇)SH(名)-080475-F。</p>                                              | -1       |
| 信号输出的处理模式<br>注1)             | SYNCIO            | 整数 1       | <p>指定基于M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32/M_Dout及Def Io的信号输出时的处理模式。</p> <p>向上兼容模式/高速模式1/高速模式2=0/1/2</p> <p>向上兼容模式: 不加快信号更新周期,以兼容旧版的方式处理。</p> <p>高速模式1: 对基于M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32的信号输出进行加速。</p> <p>高速模式2: 在高速模式1的基础上,对基于M_Dout的信号输出进行加速。</p>                                                                                                                                                                                                                                                | 2: 高速模式2 |
| PLC 输入输出信号的取入设定 (仅限 CR7xx-Q) | QXYREAD           | 整数 2       | <p>对PLC的多CPU系统中由其他号机CPU管理的输入输出单元(包括输入输出混合单元)输入输出信号的获取的有效/无效进行设定。</p> <p>※获取信号时指定的信号号码依据1号机CPU的I/O分配设定。</p> <p>不更新输入输出单元以外的信号。</p> <p>要素1: 输入信号(X)的获取[0: 无效/1: 有效]</p> <p>要素2: 输出信号(Y)的获取[0: 无效/1: 有效]</p>                                                                                                                                                                                                                                                                                                              | 0,0      |
| PLC 输入输出单元设定 (仅限 CR7xx-Q)    | QXYUNITn<br>n=1~4 | 整数 7       | <p>对PLC的多CPU系统中,以机器人CPU作为管理CPU的输入输出单元/输入输出混合单元进行设定。</p> <p>要素1: 单元类型</p> <p>0: 无对象输入输出单元</p> <p>1: 未使用(无意义)</p> <p>2: 输出单元</p> <p>3: 输入输出混合单元</p> <p>要素2: 起始输入输出号码</p> <p>0~4080(10进制数)</p> <p>要素3: 基板号码</p> <p>0: 主基板</p> <p>1~7: 扩展基板</p> <p>要素4: 插槽号码</p> <p>0~11(10进制数)</p> <p>要素5: 输入输出点数</p> <p>0: 16点/1: 32点/2: 48点/3: 64点/4: 128点/5: 256点/6: 512点/7: 1024点</p> <p>要素6: 报警时输出模式(输出·输入输出混合单元用)</p> <p>0: 清除</p> <p>1: 保持</p> <p>要素7: I/O响应时间1(输入·输入输出混合单元用)</p> <p>0: 10ms/4: 1.0ms/5: 5.0ms/6: 20ms/7: 70ms</p> |          |

注 1) 本参数可加快通过系统状态变量 M\_Out 等进行的外部输出信号的处理。  
在以下所示的程序例 1 中，会加快步编号 1 及步编号 4 的输出信号处理。

< 程序例 1>

```
1 M_Out (9) =1 ' 开启输出信号 9
2 *ack_check '
3 If M_In (7) =0 Then *ack_check' 等待直到输入信号 7 开启为止 (互锁)
4 M_Out (9) =0 ' 关闭输出信号 9
5 End
```

\* 高速化的参数值：在上述程序例中，处理时间将可缩短约 80%。

但是，在 CR7xx-D 控制器的 CC-Link、PROFIBUS、并行输入输出接口（卡）中，在有连续多行的外部信号输出指令时效果明显。

< 程序例 2>

```
1 M_Out (9) =1 ' 开启输出信号 9
2 M_Out8 (10) =&H1F ' 从输出信号 10 向 8 位输出 &H1F
3 M_Out16 (18) =&H3FFF ' 从输出信号 18 向 16 位输出 &H3FFF
4 M_Out32 (33) =&H7FFFFFFF' 从输出信号 33 向 32 位输出 &H7FFFFFFF
5 End
```



## 注意

为了保证机器人和周边装置间的信号传输，请务必实施信号的互锁。如果未实施互锁，信号将无法正确传输，从而造成误动作。

5.2.1 关于多CPU输入偏置 (仅限CR7xx-Q控制器)

(1) 案例 (A)

不对输入使用偏置时 (参数: QMLTCPUS = -1 时)

表 5-3: CPU 共有内存和机器人输入输出信号的对应关系

| PLC (字软元件) |                           | 机器人 (位软元件) |                              |
|------------|---------------------------|------------|------------------------------|
| 输出         | U3E0¥G10000 ~ U3E0¥G10511 | 输入         | 机器人 CPU No.1 / 10000 ~ 18191 |
|            | U3E0¥G10512 ~ U3E0¥G11023 |            | 机器人 CPU No.2 / 10000 ~ 18191 |
|            | U3E0¥G11024 ~ U3E0¥G11535 |            | 机器人 CPU No.3 / 10000 ~ 18191 |
| 输入         | U3E1¥G10000 ~ U3E1¥G10511 | 输出         | 机器人 CPU No.1 / 10000 ~ 18191 |
|            | U3E2¥G10000 ~ U3E2¥G10511 |            | 机器人 CPU No.2 / 10000 ~ 18191 |
|            | U3E3¥G10000 ~ U3E3¥G10511 |            | 机器人 CPU No.3 / 10000 ~ 18191 |

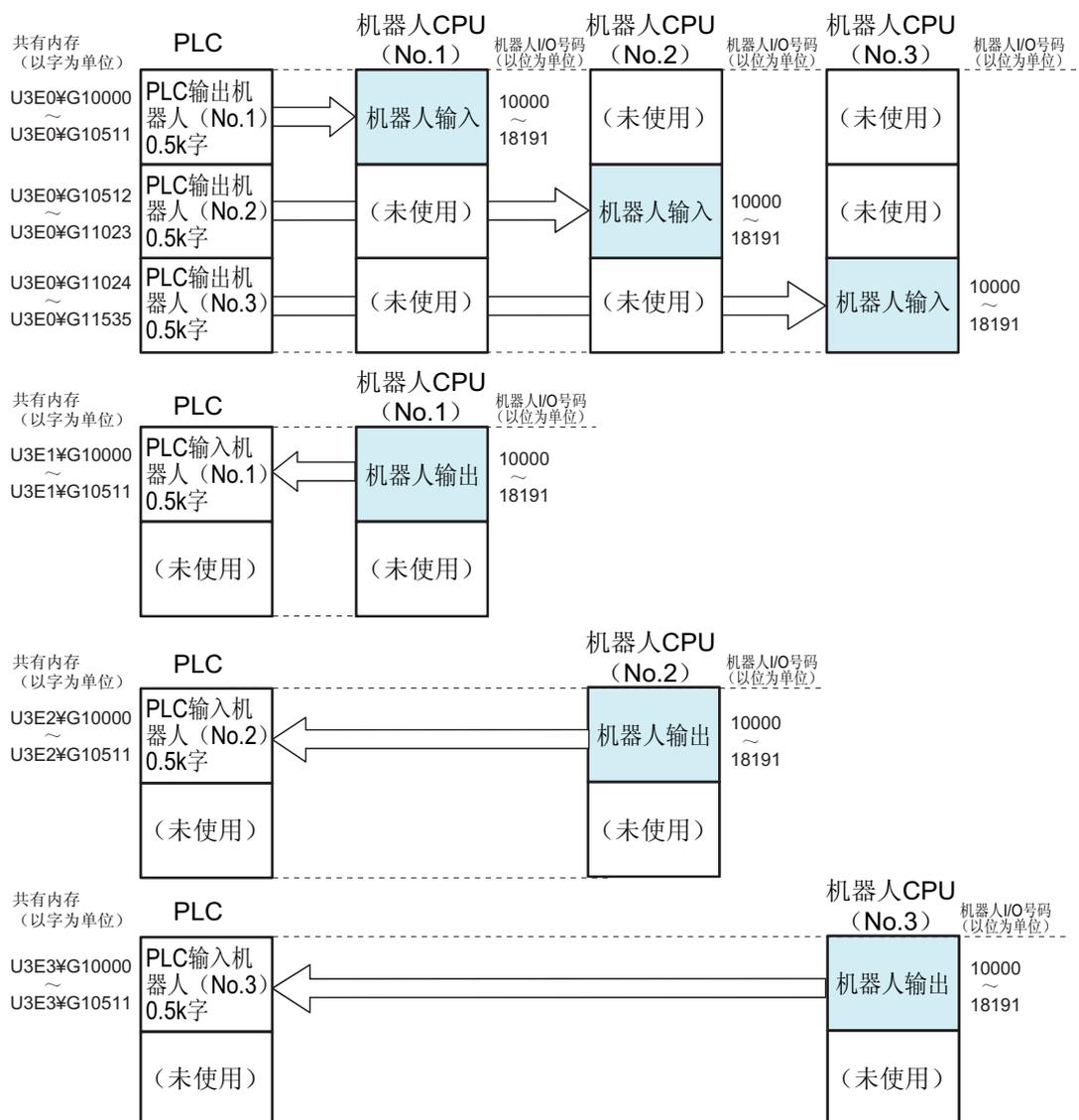


图 5-1: CPU 共有内存和机器人输入输出信号的对应关系 (案例 (A))

(2) 案例 (B)

对输入使用偏置时 (参数 QMLTCPUS = 0 ~ 14 时)

$$\alpha (1) = (\text{机器人 CPU No.1 的 QMLTCPUS}) * 1024$$

$$\alpha (2) = (\text{机器人 CPU No.2 的 QMLTCPUS}) * 1024$$

$$\alpha (3) = (\text{机器人 CPU No.3 的 QMLTCPUS}) * 1024$$

表 5-4: CPU 共有内存和机器人输入输出信号的对应关系

| PLC (字软元件) |                                                       | 机器人 (位软元件) |                              |
|------------|-------------------------------------------------------|------------|------------------------------|
| 输出         | U3E0¥G10000+ $\alpha$ (1) ~ U3E0¥G10511+ $\alpha$ (1) | 输入         | 机器人 CPU No.1 / 10000 ~ 18191 |
|            | U3E0¥G10000+ $\alpha$ (2) ~ U3E0¥G10511+ $\alpha$ (2) |            | 机器人 CPU No.2 / 10000 ~ 18191 |
|            | U3E0¥G10000+ $\alpha$ (3) ~ U3E0¥G10511+ $\alpha$ (3) |            | 机器人 CPU No.3 / 10000 ~ 18191 |
| 输入         | U3E1¥G10000 ~ U3E1¥G10511                             | 输出         | 机器人 CPU No.1 / 10000 ~ 18191 |
|            | U3E2¥G10000 ~ U3E2¥G10511                             |            | 机器人 CPU No.2 / 10000 ~ 18191 |
|            | U3E3¥G10000 ~ U3E3¥G10511                             |            | 机器人 CPU No.3 / 10000 ~ 18191 |

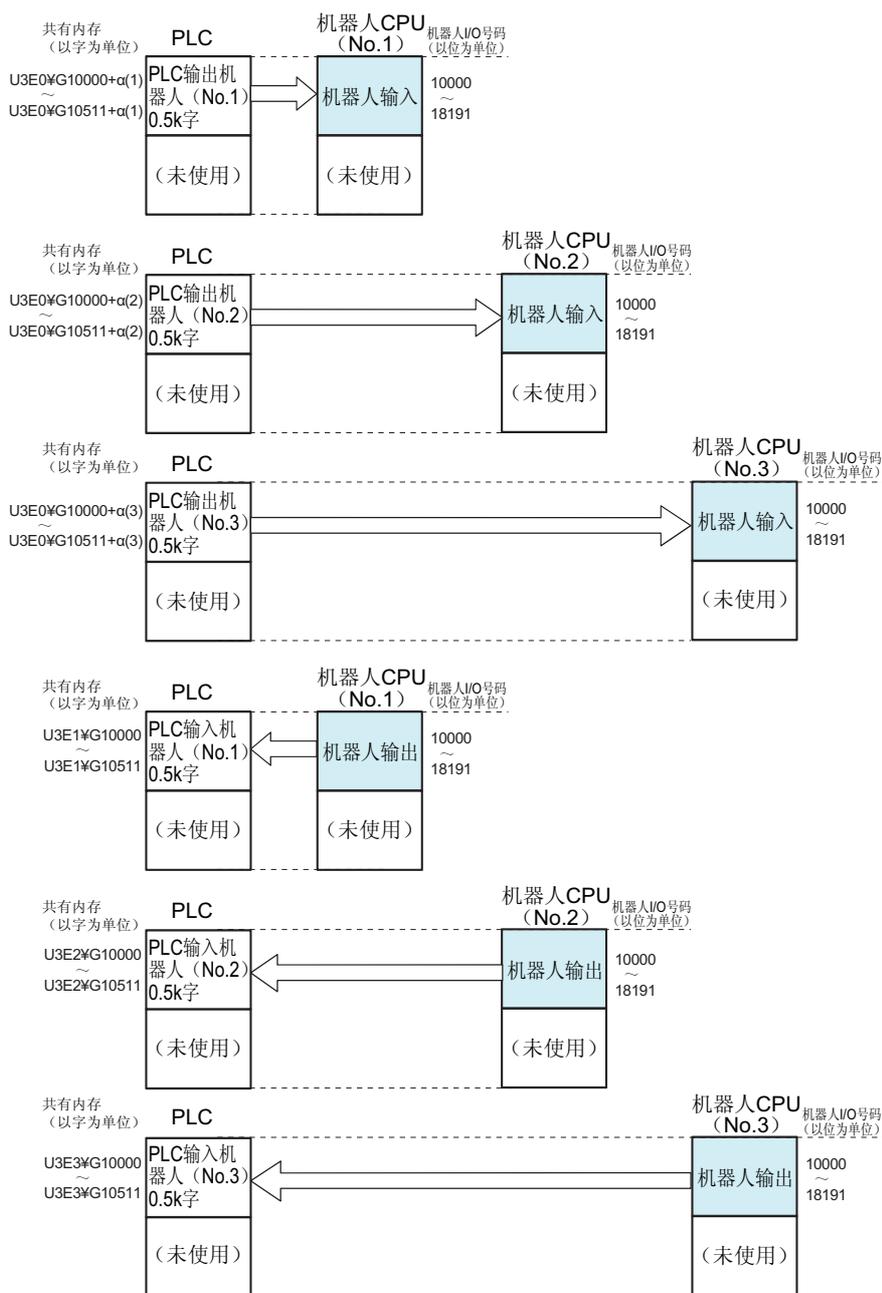


图 5-2: CPU 共有内存和机器人输入输出信号的对应关系 (案例 (B))

### 5.3 操作参数

控制器、示教单元等的操作相关参数。

表 5-5: 操作相关参数一览

| 参数                     | 参数名              | 排列数<br>文字数       | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 出货时设定值          |           |           |           |      |                  |        |                  |                  |        |
|------------------------|------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------|-----------|-----------|------|------------------|--------|------------------|------------------|--------|
| 蜂鸣器的 ON/OFF            | BZR              | 整数 1             | 指定在机器人控制器发生报警等响起的蜂鸣音的ON/OFF。<br>(OFF / ON=0 / 1)                                                                                                                                                                                                                                                                                                                                                                                                                   | 1 (ON)          |           |           |           |      |                  |        |                  |                  |        |
| 程序复位操作权                | PRSTENA          | 整数 1             | 程序复位的操作的操作权必要或不要。<br>(必要 / 不要=0 / 1) 操作权不要的情况下, 会变成从那里都可以将程序复位。但是, 为了安全考虑, 无法在示教模式。                                                                                                                                                                                                                                                                                                                                                                                | 0 (必要)          |           |           |           |      |                  |        |                  |                  |        |
| 按键开关切换时程序复位            | MDRST            | 整数 1             | 操作按键开关时, 解除程序中断状态。<br>(无效 / 有效=0 / 1)                                                                                                                                                                                                                                                                                                                                                                                                                              | 0 (无效)          |           |           |           |      |                  |        |                  |                  |        |
| 操作面板显示模式               | OPDISP           | 整数 1             | 按键开关切换时的5位数LED显示模式的设定<br>0: 在按键开关切换时会变成显示速度比例。(初始值)<br>1: 即使切换按键开关也会保持现状的显示模式                                                                                                                                                                                                                                                                                                                                                                                      | 0 (速度比例显示)      |           |           |           |      |                  |        |                  |                  |        |
| 设定程序选择权                | OPPSL            | 整数 1             | 操作面板有操作权时, 通过外部信号选择程序的操作也设为有效。<br>(外部 / OP=0 / 1)                                                                                                                                                                                                                                                                                                                                                                                                                  | 1 (OP)          |           |           |           |      |                  |        |                  |                  |        |
|                        | RMTPSL           | 整数 1             | 外部输入输出有操作权时, 通过操作面板选择程序的操作也设为有效。<br>(外部 / OP=0 / 1)                                                                                                                                                                                                                                                                                                                                                                                                                | 0 (外部)          |           |           |           |      |                  |        |                  |                  |        |
| 示教单元的速度比例操作权           | OVRDTB           | 整数 1             | 指定从示教单元的速度比例变更时的操作权为必要。<br>(不要 / 必要=0 / 1)<br>注) 设定为OVRDTB=1 (需要操作权) 时, 根据OVRDENA 的设定及控制器的模式, 示教单元的操作权的要 (示教单元有效时) / 不要 (示教单元无效时) 有所不同。请参照下表。<br><table border="1" style="margin-left: 20px;"> <thead> <tr> <th>控制器的模式</th> <th>OVRDENA=0</th> <th>OVRDENA=1</th> </tr> </thead> <tbody> <tr> <td>AUTOMATIC</td> <td>不可更改</td> <td>可更改<br/>(示教单元无效时)</td> </tr> <tr> <td>MANUAL</td> <td>可更改<br/>(示教单元有效时)</td> <td>可更改<br/>(示教单元有效时)</td> </tr> </tbody> </table> | 控制器的模式          | OVRDENA=0 | OVRDENA=1 | AUTOMATIC | 不可更改 | 可更改<br>(示教单元无效时) | MANUAL | 可更改<br>(示教单元有效时) | 可更改<br>(示教单元有效时) | 1 (必要) |
| 控制器的模式                 | OVRDENA=0        | OVRDENA=1        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |           |           |           |      |                  |        |                  |                  |        |
| AUTOMATIC              | 不可更改             | 可更改<br>(示教单元无效时) |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |           |           |           |      |                  |        |                  |                  |        |
| MANUAL                 | 可更改<br>(示教单元有效时) | 可更改<br>(示教单元有效时) |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |           |           |           |      |                  |        |                  |                  |        |
| 模式变更时的速度设定             | OVRDMD           | 整数 2             | 模式变更时自动的设定速度比例。<br>第1要素...从示教变更到自动模式时的速度比例值<br>第2要素...从自动变更示教模式时的速度比例值<br>设定为0的情况下会维持现状。                                                                                                                                                                                                                                                                                                                                                                           | 0,0             |           |           |           |      |                  |        |                  |                  |        |
| 速度比例变更操作权              | OVRDENA          | 整数 1             | 操作面板及由外部的速度比例变更时操作权为必要的设定。<br>(必要 / 不要=0 / 1)<br>注) 即使设定为OVRDENA=1 (不需要操作权), 控制器的模式为“MANUAL”时, 也无法进行倍率修调的更改。                                                                                                                                                                                                                                                                                                                                                       | 0 (必要)          |           |           |           |      |                  |        |                  |                  |        |
| 切换程序的存取对象 (ROM/RAM 运行) | ROMDRV           | 整数 1             | 将程序的存取对象在 ROM RAM 间切换。<br>0=RAM 模式 (初始值使用 SRAM 内存)<br>1=ROM 模式<br>2= 高速 RAM 模式 (使用 DRAM 内存)<br><br>高速RAM模式与继续功能无法同时使用。使用高速RAM模式时, 请将继续功能 (参数CTN) 设为无效。                                                                                                                                                                                                                                                                                                             | 2 (DRAM)        |           |           |           |      |                  |        |                  |                  |        |
| 将 RAM 领域的情报复制到 ROM 领域  | BACKUP           | 字符串 1            | 将程序、参数、共通变量、报警日志从 RAM 领域复制到 ROM 领域。<br><br>请不要变更本参数的值。                                                                                                                                                                                                                                                                                                                                                                                                             | SRAM→FLROM (固定) |           |           |           |      |                  |        |                  |                  |        |
| 将 RAM 领域的情报返回写入 ROM 领域 | RESTORE          | 字符串 1            | 将 ROM 领域的程序、参数、共通变量、报警日志从 ROM 领域返回写入到 RAM 领域。<br><br>请不要变更本参数的值。                                                                                                                                                                                                                                                                                                                                                                                                   | FLROM→SRAM (固定) |           |           |           |      |                  |        |                  |                  |        |

| 参数                                                                                                   | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                              | 出货时设定值                                             |
|------------------------------------------------------------------------------------------------------|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| 维修预报的执行间隔                                                                                            | MFINTVL  | 整数 2       | 设定维修预报用的数据的收集间隔。<br>第1要素...数据的收集LEVEL(低)~5(高)<br>第2要素...预报确认的执行间隔(单位:时间)                                                                                          | 1(低)、6(时间)                                         |
| 维修预报的通知方法                                                                                            | MFREPO   | 整数 2       | 设定维修预报的通知方法。想要阻止警告及信号输出时请设定在0。<br>第1要素...使警告发生/不发生(1/0)<br>第2要素...专用信号的输出/不输出(1/0)                                                                                | 1(出现警告)<br>0(不会输出信号)                               |
| 维修预报的复位<br>注)将本参数由示教单元读出的时候,请将全部的参数名输入后再读出                                                           | MFGRST   | 整数 1       | 将维修预报的润滑油相关累积数据复位。<br><br>※ 维修预报功能的润滑油补给催促警告(7530番台)发生,且必须在实施润滑油补给的轴,将在控制器里累积的润滑油相关数据复位。<br><br>基本上是在计算机支持软件的维修预报画面进行复位操作,但是,无法准备计算机的情况下,依据从示教单元输入本参数,可以将累积的数据复位。 | 0: 全轴复位<br>1~8: 在指定轴单位复位                           |
|                                                                                                      | MFBRST   | 整数 1       | 将维修预报的皮带相关累积数据复位。<br>※ 发生维修预报功能的皮带更换催促警告(7540号),实施了皮带更换的轴需要对控制器中累积的皮带相关数据进行复位。<br>基本上是在计算机支持软件的维修预报画面进行复位操作,但是,无法准备计算机的情况下,依据从示教单元输入本参数,可以将累积的数据复位。               | 0: 全轴复位<br>1~8: 在指定轴单位复位                           |
| 支持位置回归                                                                                               | DJNT     | 实数 8       | 依据支持位置回归 TOOL 将求得的原点的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。只可以在计算机支持软件的参数专用画面里参照。                                                                                          | 依机型会有所不同。                                          |
|                                                                                                      | MEXDTL   | 实数 6       | 依据支持位置回归 TOOL 将求得的标准 TOOL 的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。                                                                                                          | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
|                                                                                                      | MEXDTL1  | 实数 6       | 依据支持位置回归 TOOL 将求得的 TOOL 号码 1 用的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。                                                                                                      | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
|                                                                                                      | MEXDTL2  | 实数 6       | 依据支持位置回归 TOOL 将求得的 TOOL 号码 2 的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。                                                                                                       | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
|                                                                                                      | MEXDTL3  | 实数 6       | 依据支持位置回归 TOOL 将求得的 TOOL 号码 3 的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。                                                                                                       | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
|                                                                                                      | MEXDTL4  | 实数 6       | 依据支持位置回归 TOOL 将求得的 TOOL 号码 4 的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。                                                                                                       | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
|                                                                                                      | MEXDBS   | 实数 6       | 依据支持位置回归 TOOL 将求得的基本用的补偿数据代入。请不要在支持位置回归 TOOL 以外做变更。                                                                                                               | (X,Y,Z,A,B,C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
| 示教单元的启动画面的有效/无效                                                                                      | TBOP     | 整数 1       | 设定是否可以使用通过示教单元进行的程序启动。<br>0: 不能使用/1: 可以使用                                                                                                                         | 1                                                  |
| 与冗余系统错误相关的 [RESET] 键输入时的动作选择<br>※ 仅在以下的 S/W Ver 中有效。<br>S/W Ver.<br>F-Q 系列: R6b 以上<br>F-D 系列: S6b 以上 | CATEGORY | 整数 1       | 设定在发生冗余系统错误时通过[RESET]键发生的动作。<br>3: 可错误复位<br>4: 不可错误复位<br>作为对象的错误如下所示。<br>H0039(门开关开路信号配线异常)<br>H0046(可用设备配线异常)<br>H0051(外部紧急停止配线异常)                               | 3: 可错误复位                                           |

## 5.4 程序参数

程序执行及机器人相关语言参数。

表 5-6: 程序执行相关参数一览

| 参数                 | 参数名                | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 出货时设定值         |
|--------------------|--------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| 多任务件数              | TASKMAX            | 整数 1       | 指定同时执行程序数。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 8              |
| 插槽表<br>(在多任务运行时设定) | SLT*<br>* 为 1 ~ 32 | 字符串 4      | <p>指定「程序名」,「运行模式」,「起动条件」,「优先顺序」。<br/>程序名: 被选择的程序名。使用英文字母时请使用大写。无法识别小写。<br/>运行模式: 连续 / 1 循环=REP/CYC<br/>REP: 程序被循环执行。<br/>CYC: 执行 1 个循环后结束。(程序依据Goto指令等成为无限循环时将不会结束。)<br/>起动条件: 通常 / 报警时 / 通常=START/ERROR/ALWAYS<br/>START: 依据操作面板的START按钮及信号的起动执行。<br/>ALWAYS: 控制器的电源开启后立即被执行。此程序在起动中等的状态没有影响。<br/>编辑设定属性在ALWAYS的程序, 请先将暂时通常执行的属性解除后再编辑。<br/>因为通常属性的程序会在通常被执行, 因此无法编辑。<br/>请将ALWAYS变更为START且将控制器的电源再投入后使通常执行停止。<br/>ERROR: 在报警发生时被执行。此程序在起动中等的状态没有影响。利用起动条件将ALWAYS、ERROR设定的程序无法执行下列的动作指令。执行时会发生报警。<br/>Mov,Mvs,Mvr,Mvr2,Mvr3,Mvc,Mva,<br/>Drive,Getm,Relm,JRC<br/>优先顺序: 1 ~ 3 1 (3 1 为最大)<br/>此数值表示在一次执行的行数。和Priority指令的执行行数相同意义。<br/>例如: 在使用 2 个插槽执行的情况下, 在 S L T 1 指定为 1、S L T 2 指定为 2 的话, 会执行 S L T 1 的程序 1 行后, 再执行 S L T 2 的程序 2 行。<br/>因此, S L T 2 的程序会被执行的较多, 结果使 S L T 2 的优先顺序提升。</p> | "",REP,START,1 |
| 程序选择记忆             | SLOTN              | 整数 1       | <p>指定是否在程序选择时将程序名以参数记忆或是否在循环运行结束时保持程序选择状态等。<br/>(1) 程序选择时的程序名记忆有效<br/>(位0、记忆有效 / 记忆无效=1 / 0)<br/>记忆有效: 在插槽 1 的程序选择时, 在STL 1 参数储存现在的程序名。且, 在电源开启时, 选择在STL 1 参数设定的程序。<br/>记忆无效: 在插槽 1 的程序选择时, 在STL 1 参数不会储存现在的程序名。在电源开启时, 和记忆有效时相同, 选择在STL 1 参数设定的程序。<br/>(2) 循环运行结束时程序保持<br/>(bit1、保持/非保持=1 / 0)<br/>保持 : 在循环运行结束时保持程序选择状态。<br/>不会变成P.0000。<br/>非保持 : 在循环运行结束时不会保持程序选择状态。<br/>会变成P.0000。<br/>设定值和动作<br/>0: 记忆无效、非保持<br/>1: 记忆有效、非保持 (初始值)<br/>2: 记忆无效、保持<br/>3: 记忆有效、保持</p>                                                                                                                                                                                                                                                                                                                   | 1 (记忆有效、非保持)   |

| 参数                                         | 参数名             | 排列数<br>文字数    | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 出货时设定值                                                                                             |
|--------------------------------------------|-----------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 将在通常执行程序内的X**指令、SERVO指令、RESET ERR指令设定为可以执行 | ALWENA          | 整数 1          | 在SLT*参数里设定为通常执行程序（起动条件为ALWAYS）内将Xrun、Xload、Xstp、Servo、Xrst、Reset Error指令设为执行可能。<br><br>可 / 不可=1（0以外） / 0                                                                                                                                                                                                                                                                                                                                                                                                                          | 0（不可）                                                                                              |
| 使用者基本程序<br>请参照「4.4.24 使用者定义全局变量」。          | PRGUSR          | 字符串 1         | 设定使用者基本程序名。使用英文字母时请使用大写。使用小写的话无法识别。使用者基本程序是在想要使用使用者定义全局变量时使用的程序。内容只记述Def Inte及Dim等的变量定义指令。在使用者基本程序里用Dim指令声明数组时，必须再度以Dim指令定义相同变量名。没有排列变量的情况下，不需要再定义。                                                                                                                                                                                                                                                                                                                                                                               | ""（无）                                                                                              |
| 继续功能                                       | CTN             | 整数 1          | 整数 1 关于程序执行插槽，储存电源关闭时的状态且可以在下一次电源开启时，从储存状态继续开始动作。被储存的数据为程序执行环境（速度比例、执行单步行、程序内变量等）和输出信号状态。本功能为有效时，在运行中关闭电源的话，会在下回电源开启时，在待机中的状态中开始。继续的情况下，请将伺服电源投入后再输入起动。<br>(功能无效 / 有效=0 / 1)<br><br><注意事项><br>1) 使用本功能时请务必留意，对于有不带制动器的轴的机器人而言，在关闭电源后，机械臂有时会因自重而下降或旋转。<br>2) 可使用继续功能继续的程序仅限加载在任务插槽1中的程序。插槽2之后的不会继续执行，而会以程序复位状态启动。<br>3) 在使本功能有效之后，以下参数将无法变更，因此请在变更本参数前先变更以下参数。<br>SLTn、SLOTn、TASKMAX。<br>4) 在使本功能有效之后，即使变更了插槽（SLT*）参数，也不会反映到插槽表中。请在暂时使继续功能无效后，先关闭再打开电源，然后再变更插槽表（SLT*）参数。<br>5) 高速RAM模式与继续功能无法同时使用。使用继续功能时，请将高速RAM模式（参数ROMDRV）设为无效。 | 0（无效）                                                                                              |
| JRC 指令<br>(轴的多旋转功能)                        |                 |               | 设定JRC指令的执行状态。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                    |
|                                            | JRCXEXE         | 整数 1          | 指定JRC指令的执行可 / 不可。<br>执行可 / 不可 = (1 / 0)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0（执行不可）                                                                                            |
|                                            | JRCQTT          | 实数 8          | 将以JRC指令增减的变化量，从开头要素以J1,J2,J3的顺序设定到J8轴为止。只对使用者定义轴。机器人附加轴在J7,J8轴的设定会变得有效、机器人附加轴在任意轴的设定会变得有效。<br>单位依存在参数AXUNT。                                                                                                                                                                                                                                                                                                                                                                                                                       | JRC执行可能<br>机器人<br>0,0,0,0,0,360,0,0<br>或<br>0,0,0,360,0,0,0,0<br>JRC执行不可<br>机器人<br>0,0,0,0,0,0,0,0 |
|                                            | JRCORG          | 实数 8          | 执行JRC指令后设定原点设定时的原点坐标值。<br>只在使用者定义轴有效。<br>单位依存在参数AXUNT。                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0,0,0,0,0,0,0,0                                                                                    |
| 附加轴设定                                      | AXUNT           | 整数 16         | 设定附加轴的单位。<br>角度 (degree) / 长度 (mm) = 0 / 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                                                                    |
| 使用者报警设定                                    | UER1 ~<br>UER20 | 整数 1<br>字符串 3 | 依据Error指令，设定报警的讯息、原因、复位方法。<br>最多可以设定20的使用者报警。<br>第1要素...执行设定的报警号码（9000~9299为有效）无法使用初始值的9900。请变更后使用<br>第2要素...报警讯息<br>第3要素...原因<br>第4要素...回归方法<br>在讯息中包含空白的话，请将全体文字用""圈住<br>例) 9000,"Time Out","No Signal","Check Button"                                                                                                                                                                                                                                                                                                          | 9900,"message",<br>cause","treat"                                                                  |

| 参数                  | 参数名     | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                              | 出货时设定值                             |
|---------------------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| 位置数据的旋转成份的单位设定      | PRGMDEG | 整数 1       | 指定在机器人程序内位置数据的旋转成份的单位。<br>0: RAD<br>1: DEG<br>例) M1=P1.A (指定此时的单位。)<br>(在Default参照成份数据时候为弧度。)<br><br>位置常数 (P1= (100,0,300,180,0,180) (7,0)) 的旋转成份以Default的DEG。与此参数无关。                                                                                             | 0 (RAD)                            |
| 机器人语言               | RLNG    | 整数 1       | 选择使用的机器人语言。<br>2: MELFA-BASIC V<br>1: MELFA-BASIC IV                                                                                                                                                                                                              | 2                                  |
| 显示语言 <sup>注1)</sup> | LNG     | 字符串 1      | 设定显示语言。<br>"JPN": 显示日文<br>"ENG": 显示英文<br><br>下列所表示的语言会被变更。<br>(1) 计算机支持软件<br>· 机器人的报警讯息<br>· 参数说明表<br>(2) 以外部通信从机器人读出的报警讯息<br>(机器人 CPU 的以太网 I/F)<br>※ 使用 R33TB 或 R57TB 的情况下, 在各别专用的画面里设定。<br>(R33TB 请参照本书的第 94 页的「3.18 初始设定画面的操作」、R57TB 请参照别册的操作说明书「R57TB 操作说明书」) | 日文规格的情况为"JPN"<br><br>英文规格的情况为"ENG" |
| 扩展全局变量              | PRGGBL  | -          | 将本参数设为 0, 重新接通电源后, 程序外部变量的容量将减半, 但程序保存领域会从 920KB 增加到 940KB。                                                                                                                                                                                                       | 0                                  |
| 编译模式                | PRGMODE | 整数 1       | 通过将编译执行模式设为有效, 提高程序处理速度。<br>0: 编译无效<br>1: 编译有效                                                                                                                                                                                                                    | 1                                  |

注 1) 备有其它语言的操作说明书 (日文或英文), 需要时请另外洽购。

此外, 贴在机器人和控制器上的注意贴纸是依据所订购规格的语言制作。将本参数变更语言切换时请特别注意。

## 5.5 通信参数

通信相关参数。

表 5-7: 通信相关参数一览

| 参数                   | 参数名                                                                                             | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                      | 出货时设定值                                                                                                                                                                                    |
|----------------------|-------------------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RT Tool Box 2 通信方式设定 | COMSPEC                                                                                         | 整数 1       | 指定控制器和RT Tool Box 2之间的通信方式（以往的通信方式/高可靠性通信方式）。<br>0: 以往的通信方式<br>1: 高可靠性通信方式<br>与以往的通信方式相比，虽然速度变慢了，但在噪声较多的环境中可以获得可靠性更高的通信。                                                                                                  | 1                                                                                                                                                                                         |
| 通信设置                 | COMDEV                                                                                          | 字符串 8      | 对在MELFA-BASIC V的Open指令使用通信端口的情况下COM1:COM2:上分配哪个端口进行设定。<br>此参数在执行数据链路（以Open指令使用）时必须设定。<br>以程序的Open语句指定的COMn:所对应的设备，以此参数指定（n为1~8）。<br>参数要素从左开始的顺序为COM1:、COM2:....COM8:。<br><br>执行数据链路时需要设定。<br>分配OPT11~OPT19。（参照以下以太网相关的参数） | , , , , , , , ,                                                                                                                                                                           |
| 以太网                  | NETIP                                                                                           | 字符串 1      | 控制器的IP地址<br>使用F-D系列控制器时，无法设定192.168.100.xxx的地址。<br>在RT ToolBox2中恢复了参数时，通过将电源复位2次使NETIP变为有效。                                                                                                                               | F-D系列:<br>192.168.0.20<br>F-Q系列:<br>192.168.100.1                                                                                                                                         |
|                      | NETMSK                                                                                          | 字符串 1      | 子网掩码                                                                                                                                                                                                                      | 255.255.255.0                                                                                                                                                                             |
|                      | NETPORT                                                                                         | 数值 10      | 端口号码 范围 0~32767<br>realtime外部控制用，<br>对应COMDEV的OPT11~19                                                                                                                                                                    | -----<br>10000 ,<br>(OPT11) 10001 ,<br>(OPT12) 10002 ,<br>(OPT13) 10003 ,<br>(OPT14) 10004 ,<br>(OPT15) 10005 ,<br>(OPT16) 10006 ,<br>(OPT17) 10007 ,<br>(OPT18) 10008 ,<br>(OPT19) 10009 |
|                      | CPRCE11<br>CPRCE12<br>CPRCE13<br>CPRCE14<br>CPRCE15<br>CPRCE16<br>CPRCE17<br>CPRCE18<br>CPRCE19 | 数值 9       | 协议 0:无步骤, 1:有步骤, 2:数据链路<br>(1:现在无有步骤功能)<br>对应COMDEV的OPT11~19                                                                                                                                                              | (OPT11) 0 ,<br>(OPT12) 0 ,<br>(OPT13) 0 ,<br>(OPT14) 0 ,<br>(OPT15) 0 ,<br>(OPT16) 0 ,<br>(OPT17) 0 ,<br>(OPT18) 0 ,<br>(OPT19) 0                                                         |

| 参数            | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                             | 出货时设定值                                                                                                                                       |
|---------------|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 以太网           | NETMODE  | 数值 9       | 服务器指定 (1:服务器、0:客户机)<br><br>(OPT11)<br>(OPT12)<br>(OPT13)<br>(OPT14)<br>(OPT15)<br>(OPT16)<br>(OPT17)<br>(OPT18)<br>(OPT19)                       | 1,<br>1,<br>1,<br>1,<br>1,<br>1,<br>1,<br>1,<br>1                                                                                            |
|               | NETHSTIP | 字符串 9      | 数据通信对象的服务器的IP地址<br>※ 仅在通过NETMODE指定了客户机时有效<br><br>(OPT11)<br>(OPT12)<br>(OPT13)<br>(OPT14)<br>(OPT15)<br>(OPT16)<br>(OPT17)<br>(OPT18)<br>(OPT19) | 192.168.0.2,<br>192.168.0.3,<br>192.168.0.4,<br>192.168.0.5,<br>192.168.0.6,<br>192.168.0.7,<br>192.168.0.8,<br>192.168.0.9,<br>192.168.0.10 |
|               | MXTTOUT  | 数值 1       | 执行realtime外部控制指令时的超时时间<br>(7.1msec的倍数、-1时无超时)                                                                                                    | -1                                                                                                                                           |
| Job Load 超时时间 | NVJBTOUT | 整数 1       | 设定网络视觉传感器的Job Load超时时间<br>(单位: 秒, 设定范围: 1~32767)                                                                                                 | 90                                                                                                                                           |

## 5.6 关于标准TOOL坐标

在机器人安装抓手时，将机器人的控制点设置在抓手尖端时，必须要设定 TOOL 数据。  
设定方法如下列 3 点。

- 1) 以参数 MEXTL 设定。
- 2) 在机器人程序内以 TOOL 指令设定。
- 3) 将 TOOL 号码设定到 M\_Tool 变量。以 MEXTL1 ~ 16 参数设定的值会变成 TOOL 数据。  
详细内容请参照第 338 页的「M\_Tool」

出货时设定在 0，控制点为机械 I/F（法兰面）。

TOOL 数据的构造：X, Y, Z, A, B, C

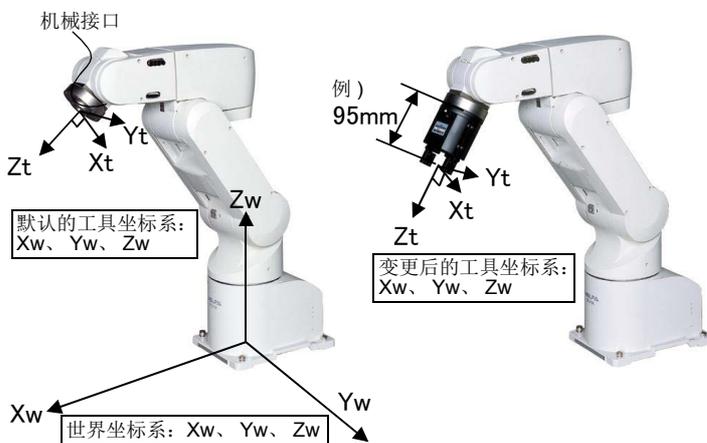
X, Y, Z 轴：以从机械 I/F 来的 TOOL 坐标的值

A 轴：TOOL 坐标的 X 轴回转

B 轴：TOOL 坐标的 Y 轴回转

C 轴：TOOL 坐标的 Z 轴回转

### 垂直 6 轴机器人的情况

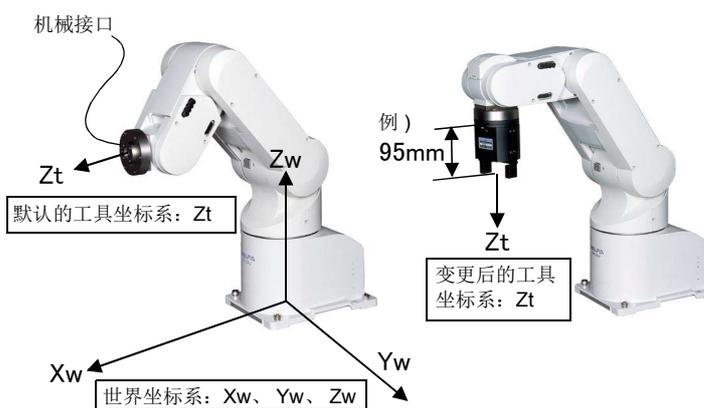


### < 垂直 6 轴机器人的情况 >

- 1) 参数的设定例  
参数名: MEXTL  
值: 0, 0, 95, 0, 0, 0
- 2) Tool 指令的设定例  
1 Tool (0,0,95,0,0,0)

6 轴机器人在动作范围内的话，各种姿势都可能。

### 垂直 5 轴机器人的情况



### < 垂直 5 轴机器人的情况 >

- 1) 参数的设定例  
参数名: MEXTL  
值: 0, 0, 95, 0, 0, 0
- 2) Tool 指令的设定例  
1 Tool (0,0,95,0,0,0)

5 轴机器人因为动作范围的关系，只对 Z 轴成份有效。即使在其它轴输入数据也会被忽视。

水平 4 轴机器人的示例



< 水平 4 轴机器人的情况 >

1) 在参数的设定例

参数名: MEXTL

值: 0, 0, -10, 0, 0, 0

2) TOOL 指令的设定例

1 Tool (0,0,-10,0,0,0)

水平 4 轴机器人基本上可以平行移动的 OFFSET 由于和垂直机器人的 TOOL 坐标方向不同, 请特别注意。

TOOL 变换数据的轴成份依据机器人的机型会有有效的和没有的东西。请参照表 5-8 做适当数据的设定。

表 5-8: 对应机器人的形式 TOOL 变换数据有效的轴成份

| 形式                                                  | 轴数 | TOOL 变换数据的轴成份 <sup>注 1)</sup> |   |   |   |   |   |
|-----------------------------------------------------|----|-------------------------------|---|---|---|---|---|
|                                                     |    | X                             | Y | Z | A | B | C |
| RV-2F、RV-4F/4FL、RV-7F 系列、RV-13F 系列、RV-20F、RV-50F 系列 | 6  | ○                             | ○ | ○ | ○ | ○ | ○ |
| RV-4FJL                                             | 5  | △                             | △ | ○ | △ | △ | × |
| RH-3FH/6FH/12FH/20FH 系列、RH-3FHR 系列                  | 4  | ○                             | ○ | ○ | △ | △ | ○ |

注 1) ○: 有效。△: 无意义、即使设定也会被忽视。×: 设定值请务必固定为 0、设定为 0 以外的话, 会有动作障碍出现。

## 5.7 关于标准基本坐标

出厂时世界坐标系位置设定为 0，基本坐标系（机器人的安装位置）与世界坐标系（作为机器人当前位置的基准的坐标系）一致。

通过利用基本转换功能，可以将世界坐标系原点设定在 J1 轴的中心位置以外。

由于执行基本转换，则世界坐标系和基本坐标系的位置关系会改变，因此如果将机器人移动至之前的示教位置则会移动至与以前不同的位置，所以请切实管理示教位置与基本转换的关系，有效利用基本转换功能。设定方法有以下 4 种。

- 1) 对参数：MEXBS 直接设定基本转换数据
- 2) 通过基本坐标号码对参数：MEXBSNO 进行设定
- 3) 通过参数：J1OFFSET 设定 J1 轴偏置角度（仅限垂直 5 轴型的机器人）
- 4) 通过在机器人程序中执行 Base 指令进行设定

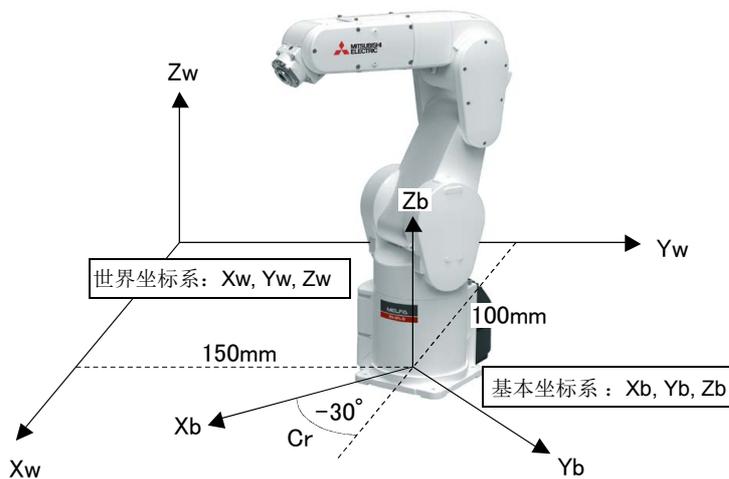
基本坐标系数据的构造：X, Y, Z, A, B, C

X, Y, Z 轴：从世界坐标原点看到的基本坐标系（机器人的安装位置）的位置

A 轴：世界坐标系的 X 轴旋转

B 轴：世界坐标系的 Y 轴旋转

C 轴：世界坐标系的 Z 轴旋转



注) 图示为垂直 6 轴机器人的例子。水平多关节型机器人也一样。

(例)

1) 在参数的设定例

参数名：MEXBS

值：100,150,0,0,0,-30

2) Base 指定的设定例

1 Base (100,150,0,0,0,-30)

通常不变更基本坐标也不会有问题，但是，变更的情况下请参考上列例子后再做设定。在机器人程序内以 Base 指令做变更的情况下，会往预期以外的位置移动，请特别注意。

基本变换数据的轴成份，会因机器人的机型而有所不同，因此分为有效及无效。请参照表 5-9 做适当的数据设定。

表 5-9：关于机器人的形式基本变换数据的有效轴成份

| 形式                                                  | 轴数 | 基本变换数据的轴成份 <sup>注1)</sup> |   |   |   |   |                  |
|-----------------------------------------------------|----|---------------------------|---|---|---|---|------------------|
|                                                     |    | X                         | Y | Z | A | B | C                |
| RV-2F、RV-4F/4FL、RV-7F 系列、RV-13F 系列、RV-20F、RV-50F 系列 | 6  | ○                         | ○ | ○ | ○ | ○ | ○                |
| RV-4FJL 系列                                          | 5  | ○                         | ○ | ○ | △ | △ | ○ <sup>注2)</sup> |
| RH-3FH/6FH/12FH/20FH 系列、RH-3FHR 系列                  | 4  | ○                         | ○ | ○ | △ | △ | ○                |

注 1) ○：有效、△：无意义、×：设定值请固定在 0。

注 2) 请对参数：J1OFFSET 设定值。

### ⚠ 注意

进行基本变换后，机器人的现在位置基准将发生变化，因此，之前的示教数据将无法直接使用了。

若错误地使机器人动作到基本变换前的示教位置，则机器人将会移动到意想不到的位置，从而造成财产损失或人身事故。

要使用基本变换功能时，请对要转换的基本坐标系和示教位置的关系加以妥善管理，在注意让机器人正确动作的同时运用基本变换功能。

## 5.8 关于使用者定义领域

使用者定义领域是随时对机器人控制点是否进入了由参数设定的任意位置领域内进行监视的功能。可使用专用输入输出或状态变量，从输出领域内 / 外的状态的功能和领域内时报警停止的功能中选择，对于与周边装置共享作业空间时与周边装置的协作及与周边装置的干涉回避等有效。此外，在位置领域以外，也可以用于机器人的姿势领域判定及附加轴领域判定。

通过以下参数设定，即可使用本功能。

- 1) 选择作为基准的坐标系（参数：AREAnCS）
- 2) 使用者定义领域的设定（参数：AREAnP1、AREAnP2）
- 3) 对象机器的指定（参数：AREAnME）
- 4) 指定使用者定义领域内的举动（参数：AREAnAT）

下面介绍各参数的详细设定内容。

### 5.8.1 坐标系的选择

该功能为在使用 Base 指令等变更基本坐标系后使用时，选择使用者定义领域是同时移动还是固定的功能。选择通过参数：AREAnCS 实现，指定「世界坐标系」（移动）、或「基本坐标系」（固定）之一作为基准坐标系。

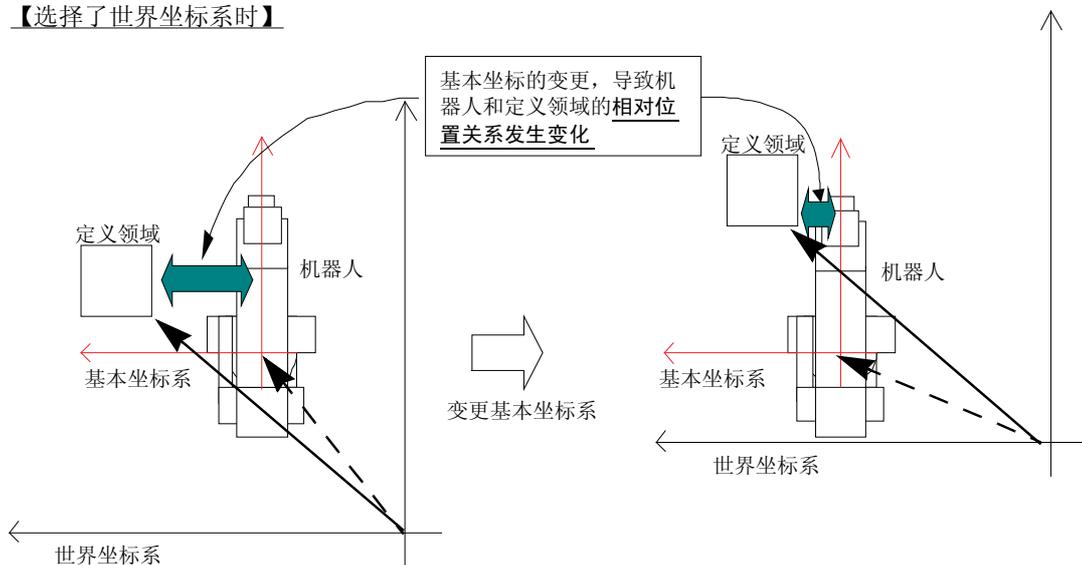
不变更基本坐标系时，无论选择哪一个，使用者定义领域都不会变化。

关于坐标系的详细内容，请参照「4.5 机器人的坐标系说明」。

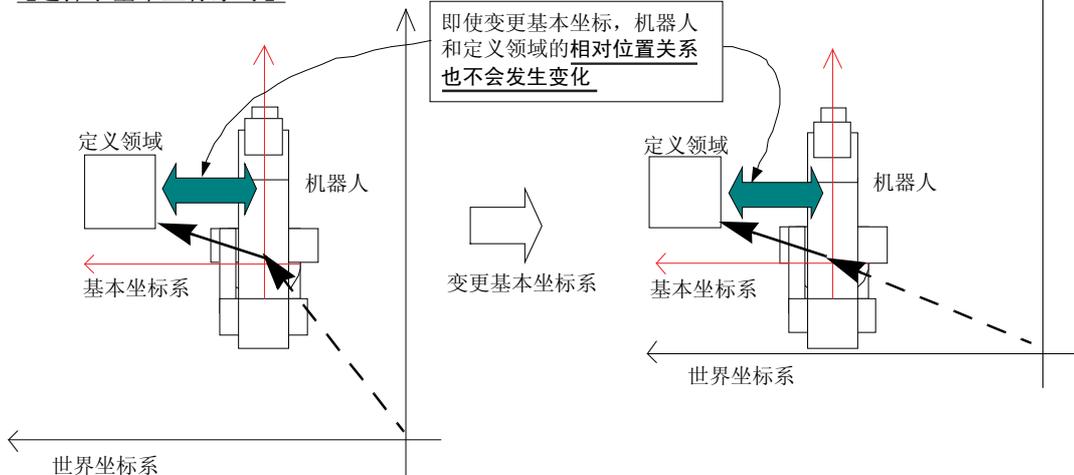
表 5-10: 作为基准的坐标系

| 坐标系   | 说明（特征）                                                                                              |
|-------|-----------------------------------------------------------------------------------------------------|
| 世界坐标系 | 变更基本坐标系，则使用者定义领域也同时移动。机器人本体和使用者定义领域的相对位置关系发生变化。                                                     |
| 基本坐标系 | 即使变更基本坐标系，使用者定义领域也不会移动。机器人本体和使用者定义领域的相对位置关系固定不变。<br>在需要变更基本坐标系后使用，但又想保持机器人本体和使用者定义领域的相对位置关系固定不变时有效。 |

#### 【选择了世界坐标系时】



#### 【选择了基本坐标系时】



注) 图示为以垂直 6 轴机器人为例，从正上方观察到的图像。水平多关节型机器人也相同。

图 5-3: 在世界坐标系和基本坐标系中的区别

## 5.8.2 领域的定义

领域分为位置领域、姿势领域和附加轴领域。  
下面介绍各领域的详细设定内容。

## (1) 位置领域的设定

通过以 AREA\*P1·AREA\*P2 (\*=1~32) 的 X、Y、Z 成分设定的对角点坐标，设定使用者定义领域的位置领域。设定的坐标值是以通过 AREA\*CS (\*=1~32) 选择的坐标系作为基准的值。

## &lt; 注意 &gt;

1) 使用 Base 等指令变更世界坐标系后使用，且在使用者定义领域的坐标系的选择中选择了「基本坐标系」时，在设定领域时请注意以下内容。

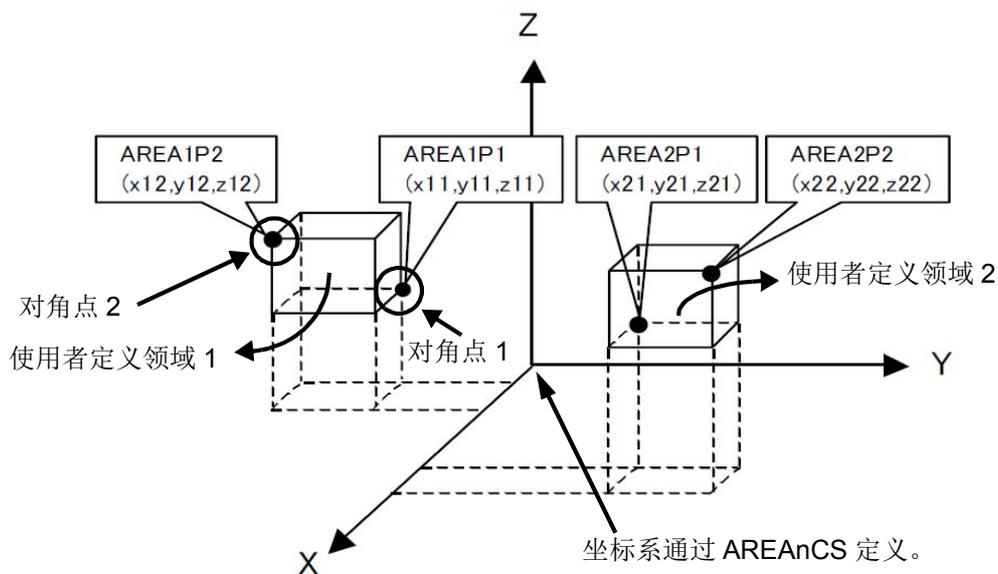
设定到 AREAnP1、AREAnP2 的 X、Y、Z 成分的坐标值，必须为在 AREAnCS 中选择的坐标系中的值。

示教单元或 RT ToolBox 等中显示的直角坐标值是以世界坐标系为基准的值。因此，当在 AREAnCS 中选择了「基本坐标系」时，所显示的坐标系和应设定的坐标值是不同的，需要特别注意。

此时，需要将显示中的坐标值变换成基本坐标系后进行设定，或是临时使世界坐标系返回初始状态后进行设定。（出货时，基本坐标系和世界坐标系一致）

2) 使用者定义领域的内侧或外侧的判定以 0.001mm 及 0.001 度的单位进行。因此，当在边界线上时，有时判定结果将不确定。

3) 即使对换 AREAnP1 和 AREAnP2 的 X、Y、Z 成分设定，定义领域也相同。



## (2) 姿势领域的设定

通过 AREAnP1·AREAnP2 的 A、B、C 成分的设定，设定使用者定义领域的姿势领域。  
设定的值是以通过 AREAnCS 选择的坐标系作为基准的值。

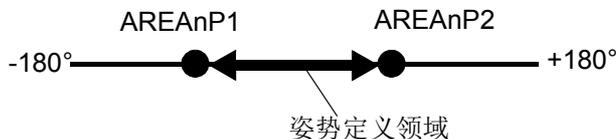
### ⚠ 注意

使用 6 轴规格的机器人时，B 轴的现在坐标值为  $\pm 90$  度附近时，微小的手腕姿势移动量也会导致 A 轴及 C 轴的坐标值符号反转等的大变化。这起因于机器人的控制。  
因此，当机器人在上述 B 轴  $=\pm 90$  度附近时，A 轴及 C 轴的姿势领域判定结果有可能不因机器人的移动量而发生变化，所以本领域判定功能不适用。  
请在 B 轴的现在坐标值不会出现在  $\pm 90$  度附近的机器人动作中使用本领域判定功能。

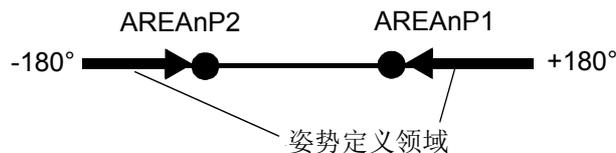
#### < 注意 >

- 1) 使用 Base 等指令变更世界坐标系后使用，且在使用者定义领域的坐标系的选择中选择了「基本坐标系」时，在设定领域时请注意以下内容。  
设定到 AREAnP1、AREAnP2 的 A、B、C 成分的值，必须为在 AREAnCS 中选择的坐标系中的值。示教单元或 RT ToolBox 等中显示的直角坐标值是以世界坐标系为基准的值。因此，当在 AREAnCS 中选择了「基本坐标系」时，所显示的坐标系和应设定的坐标值是不同的，需要特别注意。  
此时，需要将显示中的坐标值变换成基本坐标系后进行设定，或是临时使世界坐标系返回初始状态后进行设定。（出货时，基本坐标系和世界坐标系一致）
- 2) 根据 AREAnP1 和 AREAnP2 的 A、B、C 各成分的大小关系，设定的领域也会不同。（参照下图）
- 3) 不对姿势领域进行确认时，AREAnP1 的 A、B、C 成分都设为  $-360^\circ$ ，AREAnP2 的 A、B、C 成分都设为  $+360^\circ$ 。
- 4) 使用者定义领域的内侧或外侧的判定以  $0.001\text{mm}$  及  $0.001$  度的单位进行。因此，当在边界线上时，判定结果可能会不确定。

#### 【姿势成分的大小关系设定为 AREAnP2 > AREAnP1 时】



#### 【姿势成分的大小关系设定为 AREAnP1 > AREAnP2 时】



## (3) 附加轴领域的设定

通过 AREAnP1·AREAnP2 的 L1,L2 成分的设定，设定使用者定义领域的附加轴领域。  
设定了附加轴领域时，只有当位置领域·姿势领域·附加轴领域全部都在领域内时，才会判定为使用者定义领域内。

#### < 注意 >

- 1) AREAnP1、AREAnP2 的 L1、L2 成分不会受到在 AREAnCS 中定义的坐标系的影响。
- 2) 关于 L1、L2，即使对 AREAnP1 和 AREAnP2 对换设定，定义领域也相同。
- 3) 设定了附加轴领域时，只有当位置领域·姿势领域·附加轴领域全部都在领域内时，才会判定为使用者定义领域内。
- 4) 使用者定义领域的内侧或外侧的判定以  $0.001\text{mm}$  及  $0.001$  度的单位进行。因此，当在边界线上时，判定结果可能会不确定。
- 5) 未使用附加轴（J7,J8 轴）时，无需设定。

### 5.8.3 对象机器的选择

通过参数 AREA\*ME 指定进行领域确认的对象机器。通常设定机器 1("1")。正在使用多重机器等时，设定对应的机器号码。

### 5.8.4 指定使用者定义领域内的举动

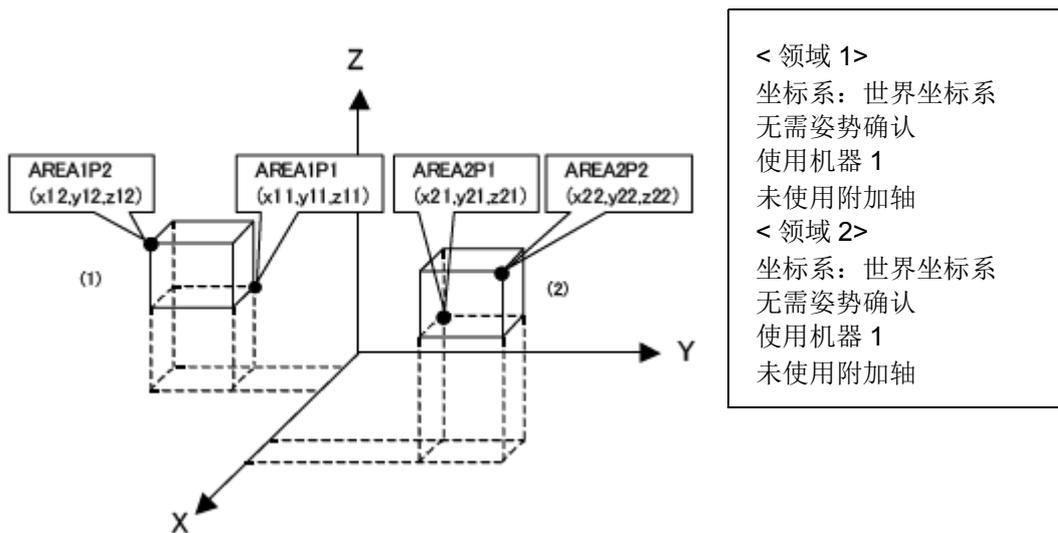
从表 5-11 中选择机器人是否处于使用者定义领域内的状态设定，通过参数 AREAnAT 的设定进行指定。

表 5-11：指定使用者定义领域内的举动

| 设定             | 领域内                                                                                                             | 领域外                                                  |
|----------------|-----------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 0: 无效          | 不进行状态设定。                                                                                                        | 不进行状态设定。                                             |
| 1: 信号输出及状态变量设定 | 专用输出信号 USRAREA 开启。<br>开启系统状态变量 (M_Uar32,M_Uar) 的对应位。                                                            | 专用输出信号 USRAREA 关闭。<br>关闭系统状态变量 (M_Uar32,M_Uar) 的对应位。 |
| 2: 报警输出        | 发生报警: H2090 并停止。<br>※ 此时, 仅执行位置领域的确认, 而忽略姿势领域、附加轴领域。<br>※ 要将机器人移动到领域外, 请对无法解除的报警执行临时性的报警复位操作, 然后通过 JOG 前进移动机器人。 | -                                                    |

### 5.8.5 设定示例

需在下图的领域 (1) 中, 在作业中输出输出信号号码 10, 在领域 (2) 中, 在作业中输出输出信号号码 11 时, 参数设定如下所示。

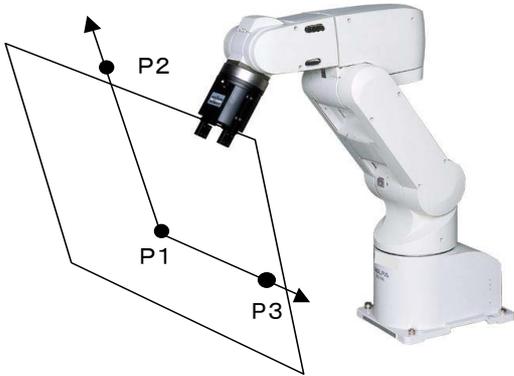


| 参数名     | 值的意义                                | 值                              |
|---------|-------------------------------------|--------------------------------|
| AREA1CS | 选择领域 1 的坐标系                         | 0                              |
| AREA1P1 | 领域 1 的对角点 1 位置数据: X、Y、Z、A、B、C、L1、L2 | x11,y11,z11,-360,-360,-360,0,0 |
| AREA1P2 | 领域 1 的对角点 2 位置数据: X、Y、Z、A、B、C、L1、L2 | x12,y12,z12,+360,+360,+360,0,0 |
| AREA1ME | 领域 1 的对象机器号码: 通常为 1                 | 1                              |
| AREA1AT | 领域 1 (无效 / 信号输出 / 报警): 0/1/2        | 1                              |
| AREA2CS | 选择领域 2 的坐标系                         | 0                              |
| AREA2P1 | 领域 2 的对角点 1 位置数据: X、Y、Z、A、B、C、L1、L2 | x11,y11,z11,-360,-360,-360,0,0 |
| AREA2P2 | 领域 2 的对角点 2 位置数据: X、Y、Z、A、B、C、L1、L2 | x12,y12,z12,+360,+360,+360,0,0 |
| AREA2ME | 领域 2 的对象机器号码: 通常为 1                 | 1                              |
| AREA2AT | 领域 2 (无效 / 信号输出 / 报警): 0/1/2        | 1                              |
| USRAREA | 输出信号: 开始号码、结束号码                     | 10,11                          |

## 5.9 自由平面限制

在机器人坐标内定义任意的平面，执行判定那个平面是在前面或是后面，使发出自由平面限制报警。

### 5.9.1 自由平面极限的定义



如左图在 3 点 (P1, P2, P3) 定义任意的平面后，判定那面是否在前面（有机器人原点侧）。依据本功能可以防止与地面冲突及和周边的干涉。

最多可监视 8 面。  
没有平面的界限。

| 参数值             | 说明                                                                                                                               |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------|
| SFCnP(n=1 ~ 8)  | 指定为了制作平面的 3 点。<br>P 1 的坐标值的 X1,Y1,Z1: 平面的原点位置<br>P 2 的坐标值的 X2,Y2,Z2: 平面的 X 轴上的位置<br>P 3 的坐标值的 X3,Y3,Z3: 平面的 X - Y 平面上的 + Y 方向的位置 |
| SFCnME(n=1 ~ 3) | 指定使自由平面限制为有效的机器号码。通常设定为 1。<br>多重机器的情况下指定其机器号码。                                                                                   |
| SFCnAT(n=1 ~ 8) | 指定已设定自由平面限制的有效 / 无效。<br>0: 无效 (初始值)<br>1: 有效 (可动作领域为机器人坐标原点侧)<br>-1: 有效 (可动作领域为无机器人坐标原点侧)                                        |

设定上述参数的话请再度投入控制器的电源。  
上述超过平面的情况下会发生自由平面限制报警。

### 5.9.2 自由平面极限的坐标系选择

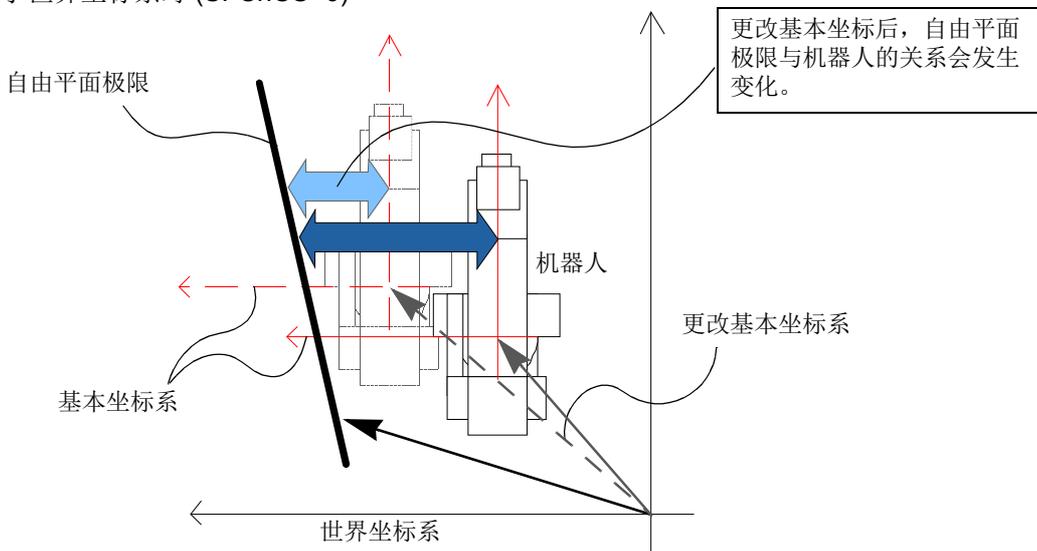
该功能是，使用 **Base** 指令等更改基本坐标系以使用时，选择是同时移动自由平面极限还是将其固定的功能。选择时是通过参数：**SFCnCS** 将基准坐标系指定为“世界坐标系”（移动）或“基本坐标系”（事先固定）的任意一个。

不更改基本坐标系时，无论选择哪个，自由平面极限均不会发生变化。

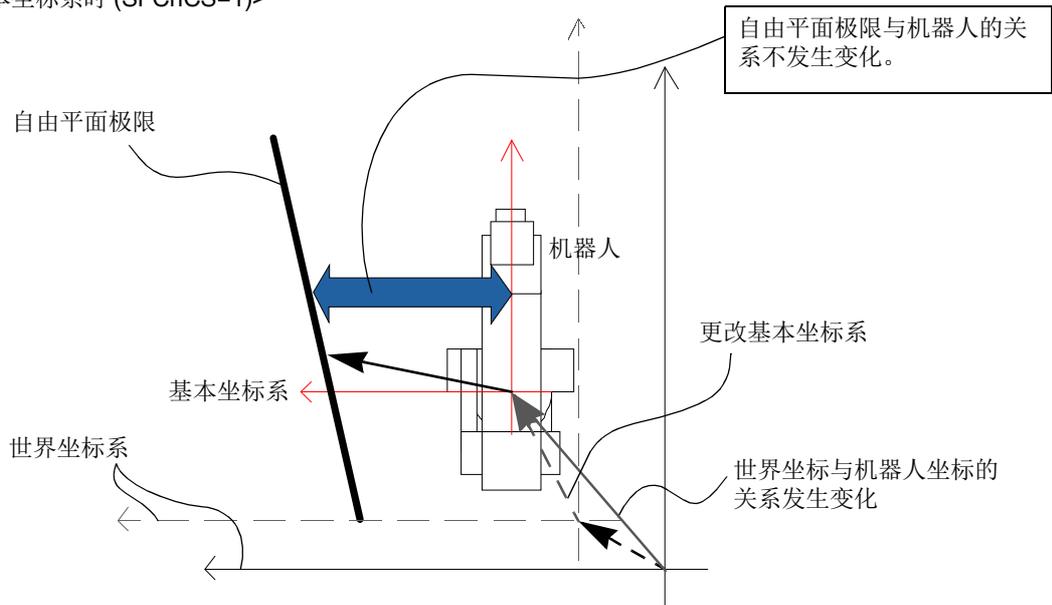
关于坐标系的详细内容，请参照第 146 页的「4.5 机器人的坐标系说明」。

| 坐标系               | 说明                                                                                    |
|-------------------|---------------------------------------------------------------------------------------|
| 世界坐标系<br>SFCnCS=0 | 更改基本坐标系后，自由平面极限也会同时移动。机器人本体与自由平面极限的相对位置关系会发生变化。                                       |
| 基本坐标系<br>SFCnCS=1 | 即使更改基本坐标系，自由平面极限也不会移动。机器人本体与自由平面极限的相对位置关系是固定的。要更改基本坐标系使用，但想将机器人本体与自由平面极限的相对位置关系固定时有效。 |

< 选择了世界坐标系时 (SFCnCS=0)>



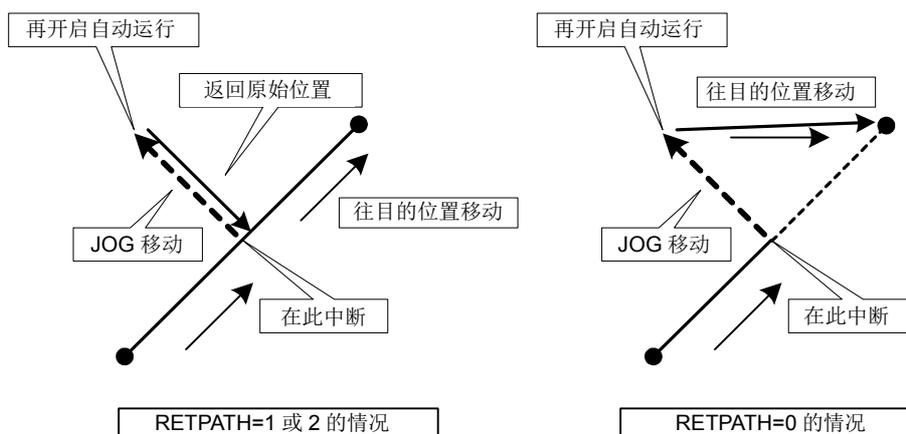
< 选择了基本坐标系时 (SFCnCS=1)>



## 5.10 中断时的JOG前进后的自动回归设定

指定将机器人在自动运行中或单步前进中中断，利用示教单元往别的位置 JOG 前进，从那个位置开始再开启自动运行，再度执行 STEP 前进的情况的轨迹动作。请参考下列图。

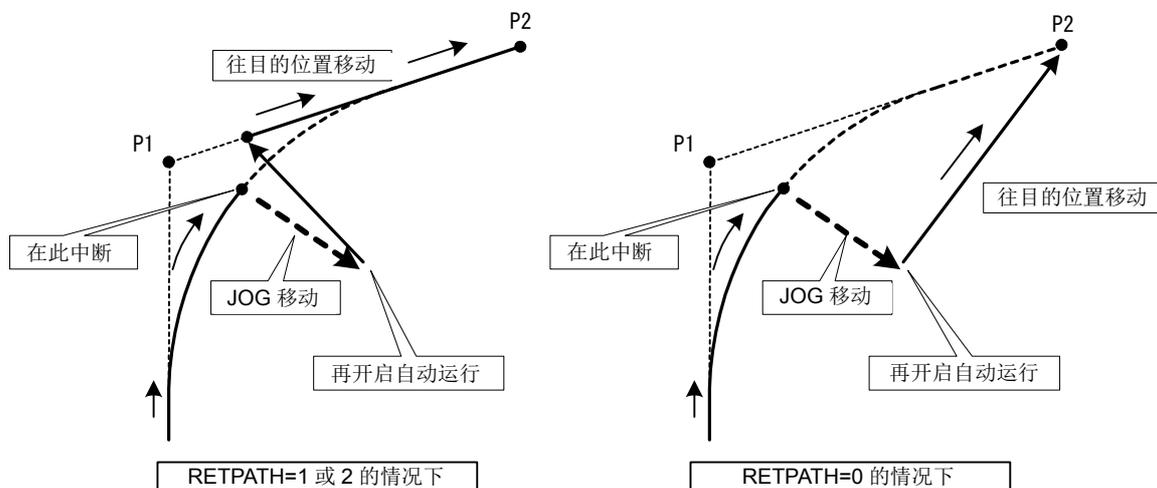
| 参数和值                | 动作说明                                                  |
|---------------------|-------------------------------------------------------|
| RETPATH=1 (Default) | 1) 以关节插补返回到原来的中断位置。<br>2) 重新开始中断的行。                   |
| RETPATH=0           | 从以 JOG 移动的位置开始再开始中断行，因此，从现在的位置到下一个目的位置以执行中的指令的插补形式移动。 |
| RETPATH=2           | 1) 以直线插补返回到原来的中断位置。<br>2) 重新开始中断的行。                   |



[注意] 在参数「RETPATH」设定为 1，以关节 JOG 以外（直交、TOOL、圆筒等）移动的情况下，因为返回到中断位置时是以关节插补动作，因此请注意与周边装置的干涉。

[注意] 在参数「RETPATH」设定为 2 的时候，构造数据为有效的机器人及有多旋转的机器人的情况下，从中断位置以关节 JOG 移动的话，本来的构造数据和多旋转数据会变不同的位置，会无法返回到中断位置。此情况下，请符合中断时的位置后再重新开始。

使用 Cnt 指令执行着连续动作（轨迹接续动作）的情况下，如下图所示，被设定为「RETPATH=1 或 2」的时候，不会返回到中断位置，而是返回到从 P1 开始到 P2 的移动路径上的位置。设定为「RETPATH=0」时，从现在的位置往目的位置移动。



### 5.11 电源开启的程序自动执行方法

只开启控制器的电源将机器人程序自动的执行的执行的方法如下所示。  
但是，因为只开启电源机器人就会动作，因此运用的时候请特别注意。

表 5-12: 相关参数

| 参数和值   | 动作说明                                                   |
|--------|--------------------------------------------------------|
| SLT*   | 例) SLT2=2,ALWAYS,REP<br>指定程序名、起动条件、运行形态。<br>这次起动条件为重点。 |
| ALWENA | 0 → 1<br>可以在通常执行程序内执行 XRun、Xload 等的多任务相关的指令及 Servo 指令。 |

1) 首先，制作通常执行用的程序和动作用程序。

< 程序 2 号 通常执行用程序 >

```

1 ' Auto Start Sample Program
2 '
3 ' 按键开关为 AUTO(Ext) 的时候，执行 Prg.1。
4 ' 按键开关为 AUTO(Ext) 以外的时候，停止程序且返回到执行行的开头。
5 '
6 IF M_Mode<>2 And (M_Run(1)=1 Or M_Wai(1)=1) Then GoSub *MTSTOP
7 IF M_Mode=2 And M_Run(1)=0 And M_Wai(1)=0 Then GoSub *MTSTART
8 IF M_Mode=1 Then Hlt ' for DEBUG
9 End
10 '
11 *MTSTART
12 XRun 1,"1"
13 Return
14 '
15 *MTSTOP
16 XStp 1
17 Wait M_Run(1)=0
18 XRst 1
19 Return

```

< 程序 1 号 动作用程序 > (任意内容)

```

1 'Main Program (位置数据为 RV-2AJ 的情况)
2 Servo On
3 M_Out(8)=0
4 Mov P1
5 M_Out(8)=1
6 Mov P2
7 End
P1=(+300.00,-200.00,+200.00,+0.00,+180.00,+0.00)(6,0)
P2=(+300.00,+200.00,+200.00,+0.00,+180.00,+0.00)(6,0)

```

2) 设定参数。

| 参数     | 动作说明                                                  |
|--------|-------------------------------------------------------|
| SLT2   | SLT2=2,ALWAYS,REP 将程序 2 号以通常执行模式执行。                   |
| ALWENA | 0 → 1<br>可以在通常执行程序内执行 XRUN、XOAD 等的多任务相关的指令及 Servo 指令。 |

设定后请将控制器的电源关闭。

3) 开启电源。

在上述示范例中开启电源，使按键开关为 AUTO(Ext.) 的话，程序 1 号会被执行、机器人动作。

## 5.12 关于抓手种类

### (1) 电磁阀的种类和信号号码

根据所使用的电磁阀的种类和连接的输出信号，设定参数。

可以设定的内容如下所示。

- a) 电磁阀的漏型 / 源型的设定..... 参数: HIOTYPE  
 注) 本参数的设定同时会将抓手输入信号的逻辑设定成漏型或源型。
- b) 电磁阀的单线螺管种类 / 双线螺管种类的设定..... 参数: HANDTYPE
- c) 驱动电磁阀的输出信号号码 ..... 参数: HANDTYPE

工厂出货时，按使用漏型且为双线螺管种类的电磁阀的情况设定。

使用的电磁阀与本设定不同时，或是使用通用输出信号时，请变更表 5-13 所示的参数。

表 5-13: 工厂出货时的参数设定

| 参数名      | 值・说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HIOTYPE  | 工厂出货时的设定值: 1<br><br>进行电磁阀的漏型 / 源型的设定和抓手输入信号逻辑的漏型 / 源型的设定。<br>1: 漏型<br>0: 源型                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| HANDTYPE | 工厂出货时的设定值: D900,D902,D904,D906, , , ,<br><br>设定电磁阀的单线螺管种类 / 双线螺管种类和驱动电磁阀的输出信号号码。<br>从值的左侧起按顺序对应抓手号码 1、2.. 初始值如下所示。<br>抓手 1 = 存取信号号码 900、901<br>抓手 2 = 存取信号号码 902、903<br>抓手 3 = 存取信号号码 904、905<br>抓手 4 = 存取信号号码 906、907<br>此抓手 1 ~ 4 (或 8) 作为指令 (HOpen、HClose) 的参数使用。<br><br>< 设定方法 ><br>使用双线螺管种类的情况下，在信号号码的前面加上 'D' 并指定号码。<br>双线螺管的情况下会变成抓手 1 ~ 4。<br>使用单线螺管种类的情况下，在信号号码的前面加上 'S' 并指定号码。<br>单线螺管的情况下会变成抓手 1 ~ 8。<br><br>< 设定示例 ><br>※ 下示均为通过外部接线将用户准备的阀门连接到通用输出信号的例子。<br>①使用双线螺管种类将通用信号从 10 号开始将抓手分配为 2 个的时候<br>HANDTYPE=D10,D12, , , , ,<br>②使用单线螺管种类将通用信号从 10 号开始将抓手分配为 3 个的时候<br>HANDTYPE=S10,S11,S12, , , , ,<br>③在抓手 1 号以双线螺管分配 10 号、在抓手 2 号以单线螺管分配 12 号的时候<br>HANDTYPE=D10,S12, , , , , |

### 5.13 关于抓手初始状态

#### (1) CR750/CR751 控制器

工厂出货时的设定如下所示。

| 抓手的种类                     | 状态       | 输出信号号码的状态 |
|---------------------------|----------|-----------|
| 安装气动抓手 I/F 时<br>(假设为双线螺管) | 抓手 1 = 开 | 900=1     |
|                           | 抓手 2 = 开 | 901=0     |
|                           | 抓手 3 = 开 | 902=1     |
|                           | 抓手 4 = 开 | 903=0     |
|                           |          | 904=1     |
|                           |          | 905=0     |
|                           |          | 906=1     |
|                           |          | 907=0     |

为了在起动电源时使全部的抓手在开的状态作用，初始参数请设定如下。

| 参数名      | 值                               |
|----------|---------------------------------|
|          | 900,901,902,903,904,905,906,907 |
| HANDINIT | 1, 0, 1, 0, 1, 0, 1, 0          |

标准构成（连接 1 台）的情况如上列所示。

例如：在电源开启时，务必将抓手 1 关闭的情况下请使用下列的设定。

| 参数名      | 值                         |
|----------|---------------------------|
| HANDINIT | 0, 1, 1, 0, 1, 0, 1, 0, 1 |

[注意] 将抓手初始状态设定在「开」的话，在电源开启时会有工件掉落的情况发生，请特别注意。

[注意] 本参数为指定在机器人的尖端的抓手专用信号（900 号）的电源开启时的初始值。

设定将抓手控制以通用 IO（900 号）控制时（在 HANDTYPE 参数指定 900 号以外的信号）的电源开启时的初始状态，并非使用此 HANDINIT 参数，请使用 ORS\* 参数。以 ORS\* 设定的值在电源开启时会变成信号初始值。

## (2) CR760 控制器

出厂时的设定如下所示。

| 抓手的类型             | 状态       | 输出信号号码的状态 |
|-------------------|----------|-----------|
| 假设通过气动抓手的双电磁阀进行驱动 | 抓手 1 = 开 | 764=0     |
|                   | 抓手 2 = 开 | 765=0     |
|                   | 抓手 3 = 开 | 766=0     |
|                   | 抓手 4 = 开 | 767=0     |
|                   |          | 768=0     |
|                   |          | 769=0     |
|                   |          | 770=0     |
|                   |          | 771=0     |

| 类型     | 参数名称    | 值                                                                                                                     |
|--------|---------|-----------------------------------------------------------------------------------------------------------------------|
| 抓手输出信号 | ORST764 | 信号号码<br>764----771 772----779 780----787 788----795<br>                               <br>*****, *****, ***** , ***** |

各值按从左到右的顺序与位对应。要设定的值为“0”或“1”或“\*”这3种。

将“0”设定为 OFF

将“1”设定为 ON

“\*” = 维持现状，不更改。(电源 ON 时为 OFF)

例如，电源 ON 时，希望仅关闭抓手 1 时，务必进行如下设定。

| 参数名称    | 值                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------|
| ORST764 | 信号号码<br>764----771 772----779 780----787 788----795<br>                               <br>01*****, ***** , ***** , ***** |

[注意] 事先将抓手初始状态设定为“开”，则在电源 ON 时有可能出现工件掉落，应充分加以注意。

### 5.14 关于输出信号复位模式

工厂出货时的设定，全部的通用输出信号皆以 OFF(0) 开始。为了变更电源开启时的通用输出信号的状态，请变更下列的参数。请注意此参数也兼用在通用输出信号复位操作（专用输入信号等执行的东西）及 Clr 指定执行时的复位模式。

| 种类                                             | 参数名                                 | 值（工厂出货时全部设定为 0）                                                                      |
|------------------------------------------------|-------------------------------------|--------------------------------------------------------------------------------------|
| 远<br>程<br>/<br>脉<br>冲<br>I<br>O                | ORST0                               | 信号号码<br>0-----7 8-----15 16-----23 24-----31<br>00000000、00000000、00000000、00000000  |
|                                                | ORST32                              | 32-----40 41-----49 50-----57 58-----66（以下相同）<br>00000000、00000000、00000000、00000000 |
|                                                | ORST96                              | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST64                              | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST128                             | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST160                             | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST192                             | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST224                             | 00000000、00000000、00000000、00000000                                                  |
| P<br>R<br>O<br>F<br>I<br>B<br>U<br>S<br>配<br>件 | ORST2000                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2032                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2064                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2096                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2128                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2160                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2192                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2224                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2256                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST2288                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | :                                   | :                                                                                    |
|                                                | :                                   | :                                                                                    |
|                                                | :                                   | :                                                                                    |
| ORST5008                                       | 00000000、00000000、00000000、00000000 |                                                                                      |
| ORST5040                                       | 00000000、00000000、00000000、00000000 |                                                                                      |
| C<br>C<br>I<br>L<br>I<br>N<br>K<br>配<br>件      | ORST6000                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6032                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6064                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6096                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6128                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6160                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6192                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6224                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6256                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | ORST6288                            | 00000000、00000000、00000000、00000000                                                  |
|                                                | :                                   | :                                                                                    |
|                                                | :                                   | :                                                                                    |
|                                                | :                                   | :                                                                                    |
| ORST7984                                       | 00000000、00000000、00000000、00000000 |                                                                                      |
| ORST8016                                       | 00000000、00000000、00000000、00000000 |                                                                                      |

| 种类                                         | 参数名      | 值 (工厂出货时全部设定为 0)                                                                                                            |
|--------------------------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------|
| P<br>L<br>C<br><br>L<br>I<br>N<br>K<br>注1) | ORS10000 | 信号号码<br>1000---10007 10008---10015 10016---10023 10024---10031<br>               <br>00000000、 00000000、 00000000、 00000000 |
|                                            | ORS10032 | 10032---10039 10040---10047 10048---10055 10056---10063<br>               <br>00000000、 00000000、 00000000、 00000000        |
|                                            | ORS10064 | 00000000、 00000000、 00000000、 00000000                                                                                      |
|                                            | ORS10096 | 00000000、 00000000、 00000000、 00000000                                                                                      |
|                                            | ORS10128 | 00000000、 00000000、 00000000、 00000000                                                                                      |
|                                            |          | 00000000、 00000000、 00000000、 00000000                                                                                      |
|                                            |          | 00000000、 00000000、 00000000、 00000000                                                                                      |
|                                            | ORS18160 | 00000000、 00000000、 00000000、 00000000                                                                                      |

注 1) PLC LINK 只有 CR7xx-Q 系列。

各值从左开始的顺序对应位。设定的值为『0』或『1』或『\*』3 种类。

『0』 = 设定为 OFF

『1』 = 设定为 ON

『\*』 = 维持现状，不变更。(电源开启时会变成 OFF)

例如：在电源开启时，一定要使通用信号的 10138、10139、10140 号及通用输出信号的 10160、10161、10168 成为开启的状态时，请设定如下。

| 参数名      | 值                 |                    |               |                      |
|----------|-------------------|--------------------|---------------|----------------------|
| ORS10128 | 10128---10135     | 10136---10143      | 10144---10151 | 10152---10159        |
|          |                   |                    |               |                      |
|          | 00000000、         | 00000000、          | 00000000、     | 00000000..... 出货时的值  |
|          | 00000000、         | 00 <b>11</b> 0000、 | 00000000、     | 00000000..... 这次的设定值 |
| ORS10160 | 10160---10167     | 10168---10175      | 10176---10183 | 10184---10191        |
|          |                   |                    |               |                      |
|          | 00000000、         | 00000000、          | 00000000、     | 00000000..... 出货时的值  |
|          | <b>11</b> 000000、 | <b>10</b> 000000、  | 00000000、     | 00000000..... 这次的设定值 |

如上所示且在通用输出信号复位操作时，将 10148、10149、10150 开启的话，就会就维持在 ON、为关闭的话，就会维持 OFF，像这样维持现状不复位的时候，请做以下设定。

| 参数名      | 值                                   |
|----------|-------------------------------------|
| ORS10128 | 00000000、00111000、0000***0、00000000 |
| ORS10160 | 11000000、10000000、00000000、00000000 |

上述的情况，在电源开启时 10148、10149、10150 会以 0(OFF) 开始。在电源 ON 时设为 1(ON)，在通用输出信号复位作时，无法设定为维持现状。

[注意] 编辑参数的情况下，请不要弄错 0 的个数，0 的个数过多或不足的情况下，在下次的电源开启时会发生报警。

## 5.15 关于通信设定（以太网）

用于通信的端口号如表 5-14 所示。

表 5-14: 使用端口号

| 端口号           | 用途                     | 参数 <sup>注 1)</sup> |
|---------------|------------------------|--------------------|
| 20            | 预留                     |                    |
| 21            | 预留                     |                    |
| 23            | COGNEX 视觉传感器           | 不可更改               |
| 111           | 预留                     |                    |
| 502           | 预留                     |                    |
| 1024          | 预留                     |                    |
| 5001          | GOT 通信                 | GOTPORT            |
| 10000         | Mxt 指令                 | NETPORT            |
| 10001         | RT ToolBox2（数据链接也可使用）  | NETPORT            |
| 10002 ~ 10009 | 数据链接（RT ToolBox2 也可使用） | NETPORT            |
| 11000         | R56TB/R57TB            | 不可更改 (TBPORT)      |
| 12000         | 示波器功能                  | MONPORT            |
| 43815         | 预留                     |                    |
| 43816         | 预留                     |                    |

注 1) 端口号除一部分以外均可通过参数进行更改。设定范围为 0 ~ 65535。

### 5.15.1 各参数的详细内容

各参数的详细内容如下所示。

#### (1) NETIP（控制器的 IP 地址）

设定控制器的 IP 地址。IP 地址就如同邮件的地址。IP 地址以 4 个 0 ~ 255 之间的数字及 4 个数字之间的点 (.) 表示。例如，设定为 192.168.0.1 或 10.97.11.31。

控制器和网络计算机之间以 1 对 1 的方式直接连接时，IP 地址可维持初始值（任意）状态，但如果连接到本地局域网（LAN）时，请按照客户的 LAN 系统管理员的要求设定 IP 地址。IP 地址重复时，功能无法正确动作，设定时，请避免出现 IP 地址重复的问题。

此外，与控制器进行通信的计算机必须在同一网络中。

#### (2) NETMSK（子网掩码）

设定控制器的子网掩码。子网掩码是 IP 地址中为定义子网而设定的地址。子网掩码以 4 个 0 ~ 255 之间的数字及 4 个数字之间的点 (.) 表示。

例如，设定为 255.255.255.0 或 255.255.0.0。通常情况下可保持初始值状态不变。连接到本地局域网（LAN）时，请按照客户的 LAN 系统管理员的要求设定子网掩码。

#### (3) NETPORT（端口号码）

设定控制器的端口号码。端口号码就如同邮件的姓名。9 种要素以不同的数值表示端口号码。第 1 个（要素号码 1）为实时控制用，第 2 个到第 9 个（要素号码 2 ~ 9）为支持软件用或数据链路用。通常情况下，可保持初始值状态而无需变更。此外，请避免使端口号码出现重复。

**(4) CPRCE11 ~ 19 (协议)**

使用数据链路功能时，必须设定。设定通信时的协议（步骤）。协议有 3 种，分别是无步骤、有步骤、数据链路。

设定值对应以下所示的 0 ~ 2。

- 0... 无步骤                   : 使用计算机支持软件时的协议。
- 1... 有步骤                   : 预留（无此功能，请勿错误设定）
- 2... 数据链路               : 使用机器人程序的 Open/Input/Print 指令进行通信时的协议。

**(5) COMDEV( 对应 COM1 : ~ 8 的设备定义 )**

使用数据链路功能时，必须设定。设定对应 COM1 : ~ 8 的设备定义。COM1 : ~ 8 通过机器人程序的 Open 指令使用。仅在协议设定 (CPRCE11 ~ 19) 中指定了数据链路时，必须设定。在以太网接口选项中设定 OPT11 ~ OPT19 的设备名 (字符串)。OPT11 ~ OPT19 的设定值对应如下表所示的在参数 NETPORT 中设定的端口号码。

※ 在以下的参数 NETOPORT (n) 及 COMDEV (n) 中，数值 n 表示该参数的要素号码。

| 数值 n | "COMDEV (n) 中设定的设备名" | 端口号码                 |
|------|----------------------|----------------------|
| 1    | OPT11                | NETPORT(2) 中指定的端口号码  |
| 2    | OPT12                | NETPORT(3) 中指定的端口号码  |
| 3    | OPT13                | NETPORT(4) 中指定的端口号码  |
| 4    | OPT14                | NETPORT(5) 中指定的端口号码  |
| 5    | OPT15                | NETPORT(6) 中指定的端口号码  |
| 6    | OPT16                | NETPORT(7) 中指定的端口号码  |
| 7    | OPT17                | NETPORT(8) 中指定的端口号码  |
| 8    | OPT18                | NETPORT(9) 中指定的端口号码  |
| 9    | OPT19                | NETPORT(10) 中指定的端口号码 |

例如，将 NETPORT (3) 中设定的端口号码作为通信端口文件名 COM3: 分配给数据链路时，

- COMDEV(3) = OPT13                    ※ 设定 COMDEV 的第 3 要素为 OPT13
- CPRCE13 = 2                            ※ 作为数据链路设定。

**(6) NETMODE (服务器指定)**

使用数据链路功能时设定。

将控制器的数据链路功能中的 TCP/IP 通信设定为服务器或客户机。需要根据控制器上连接的对象侧的应用程序进行切换。

**(7) NETHSTIP (数据通信对象的服务器的 IP 地址)**

在数据链路功能中将控制器作为客户机使用时，进行设定。在数据链路功能中，指定控制器连接的对象服务器的 IP 地址。仅在 NETMODE 的服务器指定中将控制器设为「客户机」时进行设定。

**(8) MXTTOUT (执行 realtime 外部控制指令时的超时设定)**

使用 realtime 外部控制指令，需要设定与控制器之间的通信超时时间时进行变更。指定 1 个控制周期 (约 7.11msec) 的倍数。执行了 realtime 外部控制指令后，计算机没有对控制器发出通信数据时，开始对超时时间计时，当到达该 MXTTOUT 的设定值时，将通过报警 (H7820) 停止。例如，想要在约 7 秒内没有通信即报警时，设定为 1000。工厂出货时，设定为无超时时间 (-1)。

## 5.15.2 参数设定示例1（使用支持软件时）

使用支持软件时的设定示例如下所示。

进行控制器的参数设定和所使用计算机的 OS 的网络设定。

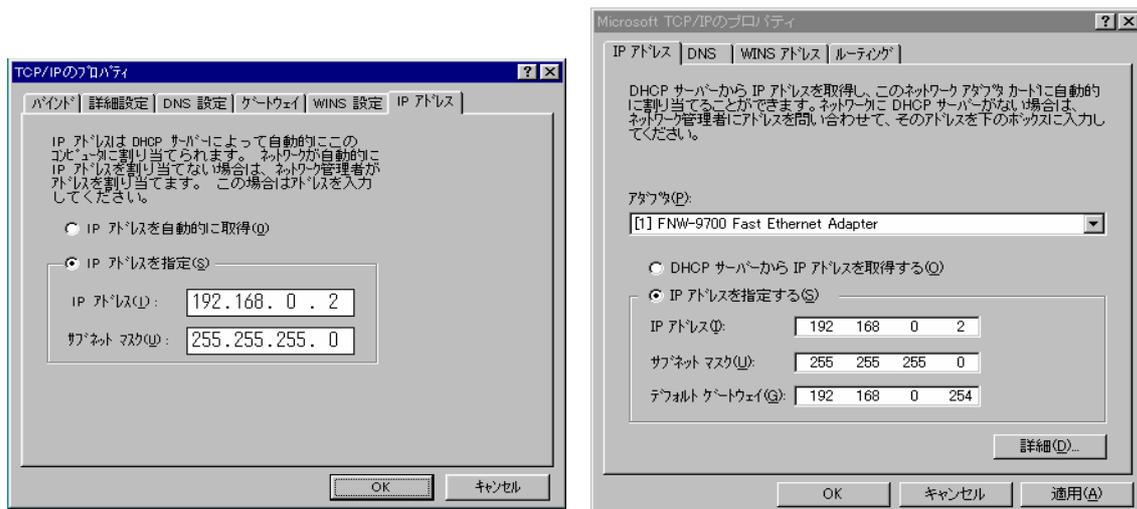
| 项目         | 设定值          |
|------------|--------------|
| 控制器的 IP 地址 | 192.168.0.20 |
| 计算机的 IP 地址 | 192.168.0.2  |
| 控制器的端口号码   | 10001        |

按下表所示，设定控制器的参数。

如为工厂出货时的设定，则可以用不用变更参数，而直接使用。

| 要变更的参数名 | 变更前后 | 参数的值         |
|---------|------|--------------|
| NETIP   | 前    | 192.168.0.20 |
|         | 后    | "（维持初始值不变）   |
| NETPORT | 前    | 10001        |
|         | 后    | "（维持初始值不变）   |

接着，将计算机的 IP 地址设定为 192.168.0.2。在网络的属性中设定。Windows95（左下图）、WindowsNT4.0（右下图）可供参考。



在 Windows 的 TCP/IP 属性的网络设定（网络计算机的属性）中设定计算机的 IP 地址。不同版本的 Windows 的设定画面也会有所不同，详细的设定方法请参照 Windows 附带的手册等资料。

此外，有关计算机支持软件的设定方法及使用方法，请参照计算机支持软件所附带的使用说明书。

## 5.15.3 参数设定示例2-1（使用数据链路功能：控制器为服务器时）

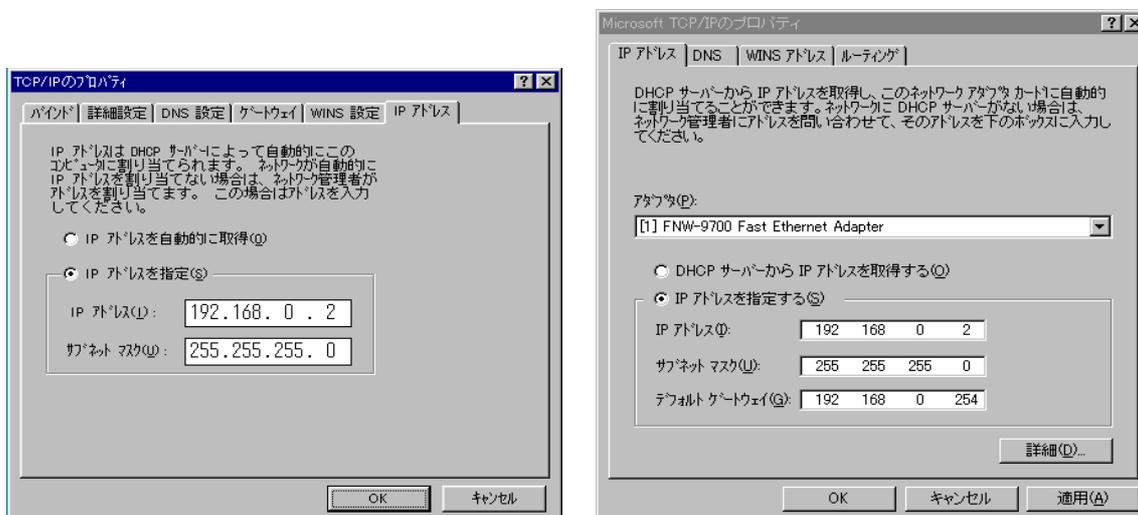
数据链路功能中，控制器为服务器时的设定示例如下所示。

| 项目                        | 设定值          |
|---------------------------|--------------|
| 控制器的 IP 地址                | 192.168.0.20 |
| 计算机的 IP 地址                | 192.168.0.2  |
| 控制器的端口号码                  | 10003        |
| 通信端口号码<br>OPEN 指令的 COM 号码 | COM3         |

| 要变更的参数名 | 变更前・后 | 参数的值                                                        |
|---------|-------|-------------------------------------------------------------|
| NETIP   | 前     | 192.168.0.20                                                |
|         | 后     | " (维持初始值不变)                                                 |
| NETPORT | 前     | 10000,10001,10002,10003,10004,10005,10006,10007,10008,10009 |
|         | 后     | " (维持初始值不变)                                                 |
| CPRCE13 | 前     | 0                                                           |
|         | 后     | <u>2</u>                                                    |
| COMDEV  | 前     | RS232, , , , , ,                                            |
|         | 后     | RS232, , <u>OPT13</u> , , , , ,                             |

将计算机的 IP 地址设定为 192.168.0.2。在网络的属性中设定。

Windows95（左下图）、WindowsNT4.0（右下图）可供参考。



在 Windows 的 TCP/IP 属性的网络设定（网络计算机的属性）中设定计算机的 IP 地址。不同版本的 Windows 的设定画面也会有所不同，详细的设定方法请参照 Windows 附带的手册等资料。

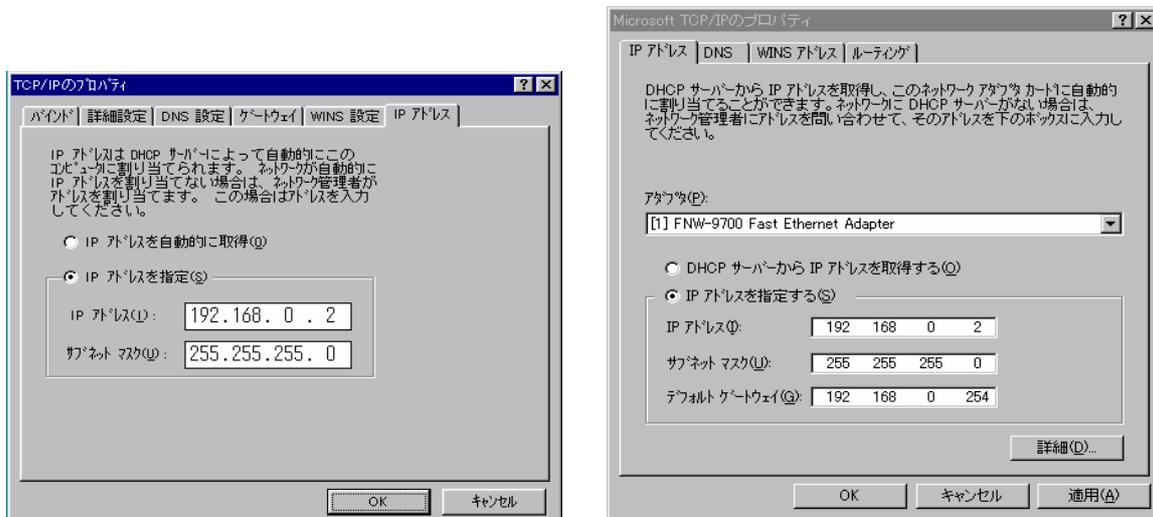
5.15.4 参数设定示例2-2（使用数据链路功能：控制器为客户机时）

数据链路功能中，控制器为客户机时的设定示例如下所示。

| 项目                        | 设定值          |
|---------------------------|--------------|
| 控制器的 IP 地址                | 192.168.0.20 |
| 计算机的 IP 地址                | 192.168.0.2  |
| 控制器的端口号码                  | 10003        |
| 通信端口号码<br>OPEN 指令的 COM 号码 | COM3         |

| 要变更的参数名  | 变更前・后 | 参数的值                                                                                                                         |
|----------|-------|------------------------------------------------------------------------------------------------------------------------------|
| NETIP    | 前     | 192.168.0.20                                                                                                                 |
|          | 后     | " (维持初始值不变)                                                                                                                  |
| NETPORT  | 前     | 10000,10001,10002,10003,10004,10005,10006,10007,10008,10009                                                                  |
|          | 后     | " (维持初始值不变)                                                                                                                  |
| CPRCE13  | 前     | 0                                                                                                                            |
|          | 后     | <b>2</b>                                                                                                                     |
| COMDEV   | 前     | RS232, , , , , , ,                                                                                                           |
|          | 后     | RS232, , <b>OPT13</b> , , , , ,                                                                                              |
| NETMODE  | 前     | 1,1,1,1,1,1,1,1,1                                                                                                            |
|          | 后     | 1,1, <b>0</b> ,1,1,1,1,1,1                                                                                                   |
| NETHSTIP | 前     | 192.168.0.2, 192.168.0.3, 192.168.0.4, 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10         |
|          | 后     | 192.168.0.2, 192.168.0.3, <b>192.168.0.2</b> , 192.168.0.5, 192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9, 192.168.0.10 |

将计算机的 IP 地址设定为 192.168.0.2。在网络的属性中设定。  
Windows95（左下图）、WindowsNT4.0（右下图）可供参考。



在 Windows 的 TCP/IP 属性的网络设定（网络计算机的属性）中设定计算机的 IP 地址。不同版本的 Windows 的设定画面也会有所不同，详细的设定方法请参照 Windows 附带的手册等资料。

## 5.15.5 参数设定示例3（使用实时外部控制功能）

使用 **realtime** 外部控制功能时的设定示例如下所示。

| 项目         | 设定值          |
|------------|--------------|
| 控制器的 IP 地址 | 192.168.0.20 |
| 计算机的 IP 地址 | 192.168.0.2  |
| 控制器的端口号码   | 10000        |

| 要变更的参数名 | 变更前・后 | 参数的值                                                        |
|---------|-------|-------------------------------------------------------------|
| NETIP   | 前     | 192.168.0.20                                                |
|         | 后     | " (维持初始值不变)                                                 |
| NETPORT | 前     | 10000,10001,10002,10003,10004,10005,10006,10007,10008,10009 |
|         | 后     | " (维持初始值不变)                                                 |
| MXTTOUT | 前     | -1                                                          |
|         | 后     | " (维持初始值不变)                                                 |

将计算机的 IP 地址设定为 192.168.0.2。在网络的属性中设定。

Windows95（左下图）、WindowsNT4.0（右下图）可供参考。



在 Windows 的 TCP/IP 属性的网络设定（网络计算机的属性）中设定计算机的 IP 地址。不同版本的 Windows 的设定画面也会有所不同，详细的设定方法请参照 Windows 附带的手册等资料。

### 5.15.6 连接的确认

在使用之前，请再次确认以下项目。

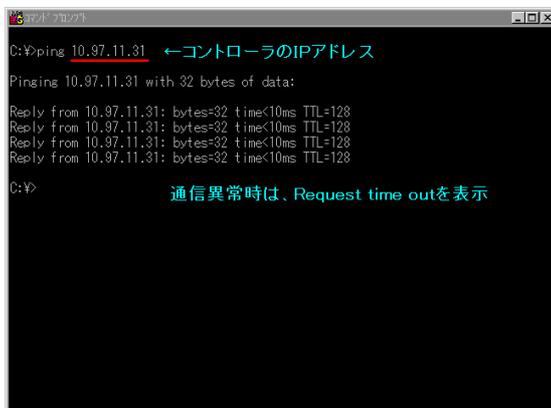
| 编号 | 确认项目                                                          | 确认 |
|----|---------------------------------------------------------------|----|
| 1  | 示教单元是否固定牢靠？                                                   |    |
| 2  | 控制器和计算机间的以太网电缆是否正确连接？                                         |    |
| 3  | 是否使用了合适的以太网电缆？（计算机和控制器 1 对 1 直接连接时，使用交叉电缆。LAN 中使用集线器时，使用直通电缆） |    |
| 4  | 控制器的参数设定是否正确？                                                 |    |
| 5  | 参数设定完成后，是否关闭过一次电源？                                            |    |

### 5.15.7 通过Windows的ping指令确认连接的方法

以下介绍使用 Windows 的 ping 指令确认连接的方法。

从 Windows 的「开始」—「所有程序 (P)」菜单中，启动 MS-DOS 提示符（命令提示符），如下所示，指定控制器的 IP 地址。

通信正常时，会如下图所示，显示 Reply from ...。异常时，会显示 Request Time out。



## 5.16 关于抓手、工件条件设定（最佳加减速设定）

最佳加减速设定依据 **Loadset** 指令、**Oadl** 指令，对应机器人尖端的负载，自动的变成最佳的加减速。为了得到最佳的加减速，预先正确的设定下列的参数为必要条件。

此外，**RV-SQ** 系列所搭载的冲突检知功能也是使用本参数。

在 **JOG** 操作时使用冲突检知功能的时候，请正确的设定 **HNDDAT0**，**WRKDAT0** 使用。

工厂出货时的设定如下所示。

|         | 参数名     | 值                            |
|---------|---------|------------------------------|
| 设定抓手的条件 | HNDDAT0 | 依机型会有所不同。                    |
|         | HNDDAT1 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT2 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT3 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT4 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT5 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT6 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT7 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
|         | HNDDAT8 | 最大负载、0.0、0.0、0.0、0.0、0.0、0.0 |
| 设定工件的条件 | WRKDAT0 | 依机型会有所不同。                    |
|         | WRKDAT1 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT2 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT3 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT4 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT5 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT6 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT7 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |
|         | WRKDAT8 | 0.0、0.0、0.0、0.0、0.0、0.0、0.0  |

参数值从左开始的顺序设定为（重量、大小 X、Y、Z、重心 X、Y、Z），抓手的条件、工件的条件分别最大可以设定 8 个。抓手的大小请将抓手进入长方体时的长度也加入。以 **LoadSet** 指令指定的抓手条件和工件条件，进行最佳加减速的计算。

| 参数名      | 值（工厂出货时） |
|----------|----------|
| HNDHOLD1 | 0、1      |
| HNDHOLD2 | 0、1      |
| HNDHOLD3 | 0、1      |
| HNDHOLD4 | 0、1      |
| HNDHOLD5 | 0、1      |
| HNDHOLD6 | 0、1      |
| HNDHOLD7 | 0、1      |
| HNDHOLD8 | 0、1      |

参数从左开始的顺序（**OPEN** 时，**CLOSE** 时）设定为把持·未把持。

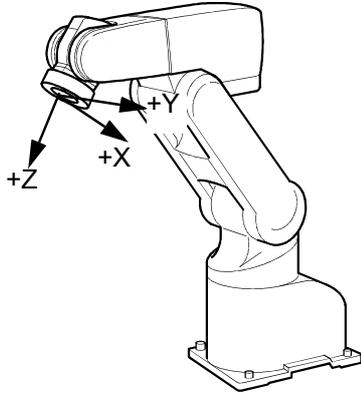
『0』 = 设定为未把持

『1』 = 设定为把持

- 对应抓手的开闭状态，进行只有抓手、抓手和工件切换的最佳加减速的计算。
- 抓手的开闭状态，依据 **HOpen/HClose** 指令的执行进行切换。

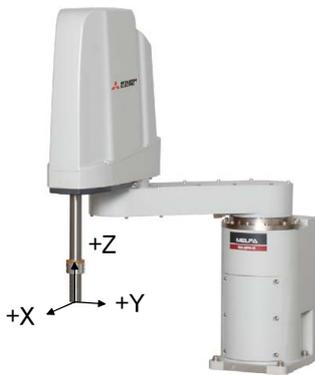
以下表示在机器人的形式别中，成为设定抓手的条件及工件的条件时的基准坐标轴。抓手的条件、工件的条件坐标轴的基准皆相同。此外，大小全部以正的值设定。

■垂直多关节型机器人



坐标轴的定义  
将 TOOL 坐标视为坐标轴。  
  
必要的设定轴：设定重心的 X、Y、Z 成份和  
大小的 X、Y、Z 方向

■水平多关节型机器人



坐标轴的定义  
将 TOOL 坐标作为坐标轴。  
  
必要的设定轴：设定重心的 X、Y、Z 成分和  
大小的 X、Y、Z 方向

## 5.17 关于特异点接近警告

在特异点存在的机器人，使用示教单元使机器人动作时，机器人的控制点往特异点接近的话，会发生警告，促使操作机器人的人注意。

尽管发生警告，如果动作可能的话，不会中断操作，机器人动作会继续。另外，从特异点的附近远离的时候，警告会自动的复位。下列为警告的详细内容。

### (1) 使警告发出的操作

使用示教单元进行下列的操作，在机器人动作时，机器人的控制点往特异点接近的话，就会发生警告。

- ① JOG 操作。（关节 JOG 模式以外）
- ② 单步前进、单步后退操作。
- ③ MS 位置移动操作。
- ④ 直接执行操作。

在上述的任一个操作往特异点接近的话，控制器的蜂鸣器会持续响起。（连续音）但是，操作面板的 STATUS. 编号显示不会有变化。

### (2) 不发生警告的操作

执行下列所示的操作及动作的情况下，即使机器人的控制点接近特异点也不会发生警告。

- ① 使用示教单元，从关节 JOG 模式进入附加轴 JOG 操作。
- ② 即使从示教单元来的操作，也在执行关节插补指令的情况。  
（Mov 指令等的执行、位置移动操作）
- ③ 程序自动运行的情况。
- ④ 根据专用输入信号（JOGENA, JOGM 等）的 JOG 操作。
- ⑤ 解除刹车，以外力移动机器人的情况。
- ⑥ 机器人为静止的情况。

## 5.18 关于高速RAM运行功能

高速 RAM 运行功能对于程序的动作、数据的保持存在部分限制，请在充分理解规格的基础上使用。

### (1) 概要

机器人程序保存到有电池后备的 RAM（SRAM）中。

执行程序的方式有 2 种，分别是 SRAM 运行和 DRAM 运行，可通过参数：ROMDRV 选择。出货时，设定为 DRAM 运行。使用 DRAM 运行（高速 RAM 运行）可以获得更高的执行速度（最大可比 SRAM 运行快 1.2 倍），但程序中所使用的变量在断电时的保存上存在部分限制。

#### 限制内容

SRAM 运行：保存所有变量。

DRAM 运行：保存程序外部变量。局部变量和使用者定义外部变量撤消。

使用时，可根据客户的用途，变更机器人程序的运行功能（DRAM 运行或 SRAM 运行）进行使用。请变更表 5-15 中所示参数：ROMDRV，并重新接通电源。关于参数的变更，请参照第 84 页的「3.14 参数画面的操作」

表 5-15: ROM/ 高速 RAM 运行参数一览

| 参数     | 说明和值                                                                |
|--------|---------------------------------------------------------------------|
| ROMDRV | 将程序的执行（运行）切换为 SRAM 运行 /DRAM 运行。<br>0= SRAM 运行<br>2= DRAM 运行（出货时设定值） |

表 5-16: 各内存的作用和 ROMDRV 参数的关系

| 内存种类 | 特征                       | ROMDRV 参数                                                                        |                                                                  |
|------|--------------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------|
|      |                          | 0 (SRAM 模式)                                                                      | 2 (DRAM 模式)                                                      |
| SRAM | 断电时不丢失<br>电池消耗后丢失<br>可读写 | 程序执行（执行结果保存）<br>程序管理操作 <sup>注1)</sup><br>系统数据读写 <sup>注2)</sup><br>共通变量读写<br>程序读写 | 程序管理操作 <sup>注1)</sup><br>系统数据读写 <sup>注2)</sup><br>共通变量读写<br>程序读写 |
| DRAM | 可高速执行<br>断电时丢失           | -                                                                                | 程序执行<br>（执行结果撤消）                                                 |

注 1) 程序管理操作指的是通过示教单元或计算机支持软件，对控制器内的程序进行复制、删除、程序名变更等的操作。

注 2) 系统数据保存包括「参数保存」、「报警日志保存」。包括程序（OPEN/PRINT/INPUT）所读写的数据文件。

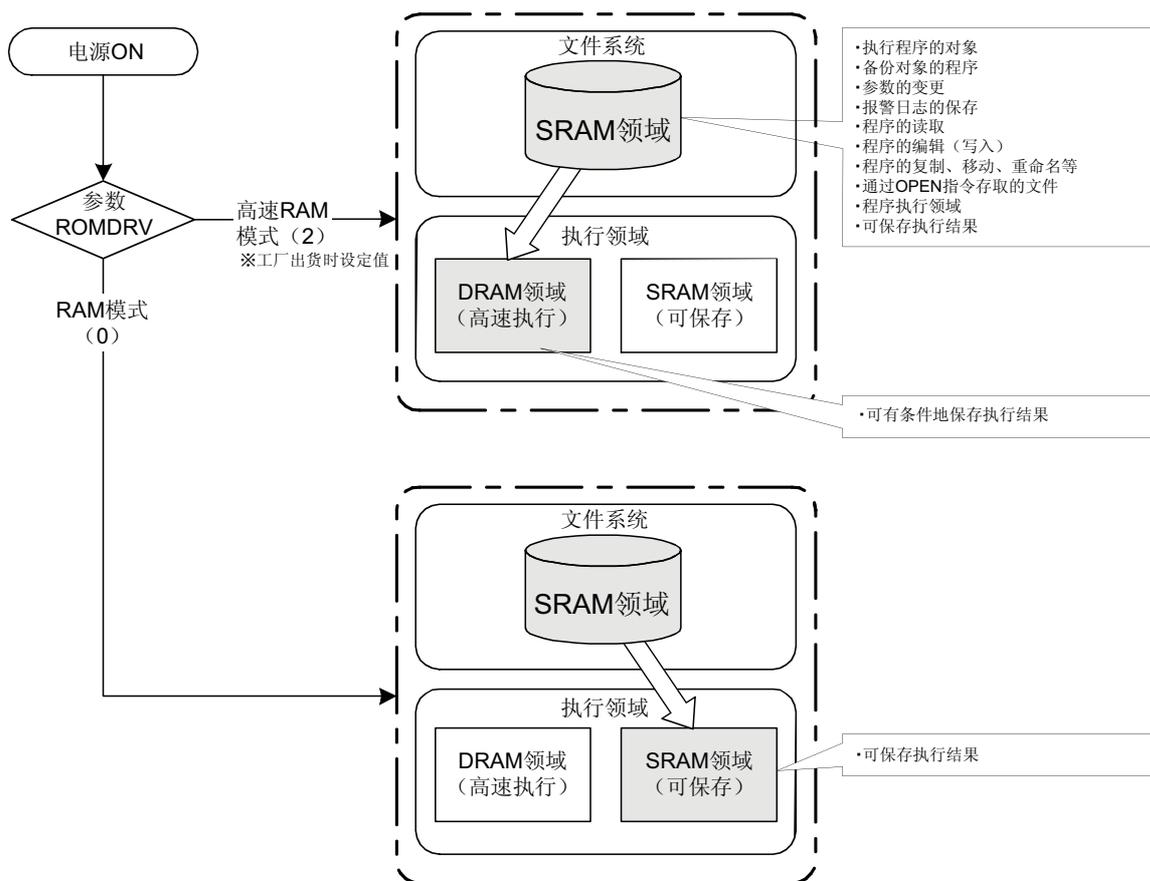


图 5-5: 高速 RAM 运行的示意图

高速 RAM 运行中使用 DRAM 内存执行。与使用 SRAM 内存执行相比，最快可以进行约 1.2 倍的高速处理。（根据程序内容而变化。）

但是，程序的执行或单步运行等机器人的操作虽然也可与 SRAM 运行一样执行，但是部分操作存在限制。请参照以下所示的注意事项。

## (2) 断电时的变量保存相关的注意事项

请注意，在 DRAM 运行中，虽然在程序执行过程中可以变更变量值，但是所变更的值在控制器断电时会被撤消。可以通过如下方法在断电时保存变量的值。

### ※ 断电时也需保存变量值的情况：

使用程序外部变量，或是在断电前实施程序选择，或是执行 CallP 指令使其结束（执行至 CallP 对象的 End 指令）。但使用者定义外部变量除外。

表 5-17: DRAM 运行 /SRAM 运行中的变量保存

| 变量 注 1)   | 高速 RAM 运行中<br>(DRAM 运行)                      | SRAM 运行中          |
|-----------|----------------------------------------------|-------------------|
| 局部变量      | 断电时执行的程序的变量值将被撤消。<br>程序选择时或 CALLP 指令结束时将被保存。 | 变量的值在断电后保持不变。注 2) |
| 程序外部变量    | 变量的值在断电后保持不变。                                |                   |
| 使用者定义外部变量 | 断电时执行的程序的变量值将被撤消。                            | 变量的值在断电后保持不变。     |

注 1) 各变量又可分为数值变量、字符串变量、位置变量、关节变量。

注 2) 但是，当以计算机支持软件改写了程序时，程序中使用的局部变量的值将被撤消。

## 5.19 关于暖机运行模式

### (1) 功能的概要

三菱的机器人，为了能在常温的环境下动作时能发挥最高的性能，所以有调整加减速及伺服系统等。

为此目的，在低温下及长期停止后的操作时，会因为使用在部品润滑用的润滑油的黏性的变化，而无法发挥本来的性能，会发生位置精度恶化及误差过大等的伺服报警。这种情况下，请利用低速执行适应运行（暖机运行）后再移转为本操作，为此，请另外准备暖机运行用的程序。

所谓暖机运行模式，控制器的电源投入之后将速度降低运行，动作时间的经过的同时也会慢慢恢复到原来的速度功能，这样的话，不用准备其它的程序也能简单的执行暖机运行。在低温下及长期停止后的操作时，发生误差过大报警的情况下，请将暖机运行模式设为有效。

#### ■使用暖机运行模式

使用暖机运行模式请在参数 WUPENA 指定为 1（有效）后再投入控制器的电源。

#### ■暖机运行模式为有效时

暖机运行模式为有效时，控制器的电源投入的话，会变成暖机状态（将速度自动的降低的状态）。在暖机状态下的运行会比指定的实际的动作速度慢，对象轴的动作时间的经过的同时会慢慢的返回到指定的动作速度。将速度降低比例称为暖机运行速度比例，此值为 100% 的时候，以机器人指定的速度动作。工厂出货时的参数设定，暖机运行速度比例会为了配合对象轴的动作时间而改变值（图 5-6）。

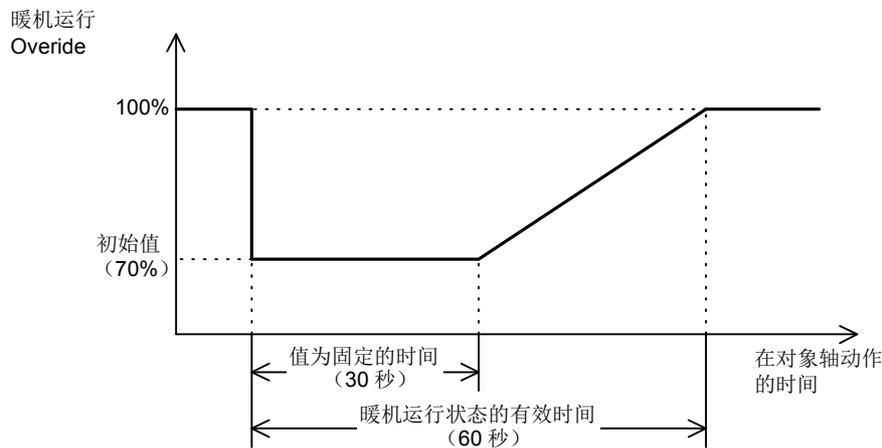


图 5-6: 暖机运行速度比例的变化

### ⚠ 注意

即使在暖机运行状态，若控制器前面的 MODE 切换开关为 "MANUAL" 的情况及依据 JOG 动作、Realtime 外部控制（MXT 指令）动作的情况下，速度不会降低，会以本来的指定速度动作。

### ⚠ 注意

在暖机状态，因为是比原来的指定速度更低速的动作，因此，与周边的 Interlock 请确实的实施。

### ⚠ 注意

对象轴的动作功率低下的话，即使使暖机运行模式为有效，也会发生误差过大等的伺服报警。此情况下，请变更程序将速度·加减速调低。

另外，在暖机状态的时候，将控制器前的 STATUS NUMBER 设为显示速度比例的话，会在从左开始的第 2 行显示底线（\_），可以确认在暖机运行状态。

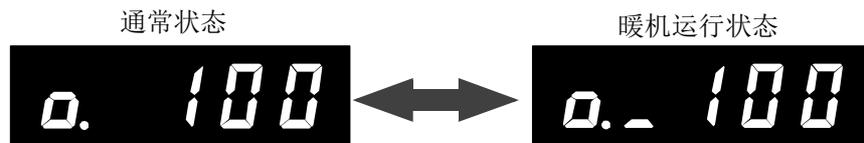


图 5-7: 暖机运行状态的速度比例表示

将对象轴动作的暖机运行状态解除的话，会以机器人的指定速度动作。另外，在解除后机器人停止持续的话，在低温下可再度冷却关节部份。因此，对象轴在长时间（工厂出货时的设定为 60 分）停止持续时，再度成为暖机运行状态，将速度调慢再动作。

注 1) 将控制器电源遮断 / 再投入的情况下，电源遮断的时间短的话，机器人的关节部份的温度不会那样的低下。因此，在暖机运行状态被解除后将电源遮断 / 再投入时，电源遮断的时间短的情况下，不会以暖机运行状态开启，而是以通常的状态开启。

注 2) 所谓对象轴，是指在暖机运行状态模式成为控制对象的关节轴，以参数 WUPAXIS 指定的关节轴。

## (2) 功能的详细

### 1) 暖机运行模式的参数、专用输入输出信号、状态变量

在暖机运行模式，追加下列的参数、专用输入输出信号、状态变量。关于这些详细内容请参照第 407 页的「5.1 动作参数」、第 509 页的「6.2PLC 连接输出功能」、第 98 页的「4MELFA-BASIC V」。

表 5-18: 暖机运行模式的参数一览表

| 参数名     | 说明和值                                                                                                                                                                 |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WUPENA  | 指定暖机运行模式的有效/无效。<br>0: 无效 / 1: 有效                                                                                                                                     |
| WUPAXIS | 在暖机运行模式将成为控制对象的关节轴，根据位的 ON/OFF 以 16 进制（从下位位开始 J1,J2,...）指定。<br>位 ON: 对象轴/位 OFF: 非对象轴                                                                                 |
| WUPTIME | 以暖机运行模式的处理，指定使用时间（单位：分）。在第 1 要素指定有效时间、第 2 要素指定再开启时间。<br>有效时间：指定成为暖机运行状态降低速度动作的时间。（设定范围：0~60）<br>再开始时间：暖机运行状态被解除后，在对象轴停止持续的情况，指定到再度成为暖机运行状态的时间。（设定范围：1~1440）          |
| WUPOVRD | 执行暖机运行状态相关速度。以第 1 要素指定初始值、第 2 要素指定值固定时间的比例。单位皆为%。<br>初始值：在暖机运行状态时指定动作速度乘以速度比例（暖机运行速度比例）的初始值。（设定范围：50~100）<br>值固定时间的比例：将暖机运行状态从初始值开始没有变化的时间，以对有效时间的比例进行指定。（设定范围：0~50） |

表 5-19: 暖机运行模式的输入输出信号一览表

| 参数名                            | 区分 | 功能                         |
|--------------------------------|----|----------------------------|
| MnWUPENA<br>(n=1~3)<br>(操作权必要) | 输入 | 使各机器的暖机运行模式有效。(n: 机器号码)    |
|                                | 输出 | 输出暖机运行模式有效。(n: 机器号码)       |
| MnWUPMD<br>(n=1~3)             | 输出 | 输出暖机运行模式将速度调降的动作。(n: 机器号码) |

表 5-20: 暖机运行模式的状态变量

| 状态变量    | 功能                                           |
|---------|----------------------------------------------|
| M_Wupov | 暖机运行状态的时候，为了降低速度动作，返回指定速度乘以速度比例（暖机运行速度比例）的值。 |
| M_Wuprt | 为了解除暖机运行状态，返回暖机运行模式的对象轴的必须动作时间。              |
| M_Wupst | 返回暖机运行状态被解除后，到再次变成暖机运行状态为止的时间。               |

2) 暖机运行模式使用时

使用暖机运行模式时，以参数指定功能有效。另外，可以用专用输入信号切换功能的有效 / 无效。

■以参数指定

以参数使暖机运行模式为有效时，在参数 WUPENA 设定为 1。参数变更后，依据控制器的电源再投入，暖机运行模式会成为有效。但是，下次像这样的情况，即使将参数 WUPENA 设定为 1，也不会成为暖机运行模式。

- 在参数 WUPAXIS 被设定为 0（暖机运行模式状态的对象轴不存在）情况
- 在参数 WUPTIME 的第 1 要素被设定为 0（在暖机运行时间为 0 分）情况
- 在参数 WUPOVRD 的第 1 要素被设定为 100（即使在暖机运行状态，速度也不会降低）情况使用暖机运行模式的情况下，请将此参数变更为适当的设定值。

■以专用输入信号切换

依据分配专用输入信号 MnWUPENA（n=1 ~ 3：机器号码），不用重新投入控制器的电源，也可以切换暖机运行模式的有效 / 无效。另外，现在的有效 / 无效的状态，可以用专用输出信号 MnWUPENA（n=1 ~ 3：机器号码）做确认

- 注 1) 使专用输入信号功能的话，必须事先依据前述的参数设定使暖机运行模式为有效。
- 注 2) 此专用输入信号必须有外部输入的操作权。此外，运行中及 JOG 动作中不接受输入。
- 注 3) 依据此专用输入信号设定的有效 / 无效的状态，即使没有外部输入的操作权也会被保持。

3) 暖机运行模式为有效时

暖机运行模式为有效时，将控制器的电源投入的话，会变成暖机运行状态。在暖机运行状态，依据对被指定的速度乘以暖机运行速度比例，可降低实际的动作运行速度。对象轴的动作时间的经过一起慢慢将动作速度返回被指定的速度，暖机状态被解除的话，会以机器人被指定速度动作。

■电源投入之后的初始状态

暖机运行模式为有效的时候，将控制器的电源投入的话，会变成暖机运行状态。但是，在暖机运行状态被解除后，将控制器的电源遮断 / 再投入时，电源遮断的时间短的情况下，因为机器人的关节部份的温度不会那样的低下，所以不会以暖机运行状态开启，而是以通常的状态开启。具体而言，满足下列的条件的话，会以通常的状态开始。

条件：从暖机运行状态解除到对象轴停止持续的时间，比在参数 WUPTIME 的第 2 要素（暖机运行状态的再起动时间）指定的时间还短的情况下，以通常的状态开始。

另外，依据专用输入信号 MnWUPENA（n=1 ~ 3：机器号码）将暖机运行模式切换为有效时，通常会变成暖机运行模式。

■确认是否在暖机运行状态的办法

现在是否在暖机状态，或者是在通常的状态，可以下列 3 种办法确认。

- 依据控制器前面的 STATUS 编号确认。  
依据将 STATUS 编号设定在速度比例来显示。在暖机运行状态从左边开始第 2 位数以底线（\_）来表示。

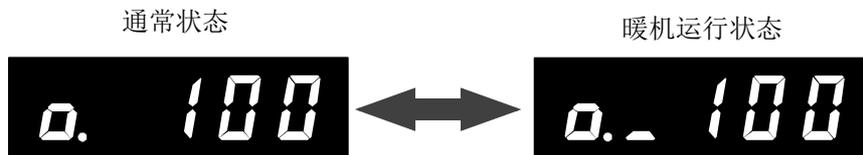


图 5-8: 暖机运行状态的速度比例表示

- 依据状态变量确认。  
根据监视状态变量 M\_Wupov（暖机运行速度比例的值）的值来进行确认。在通常的状态下，M\_Wupov 的值会成为 100%，但是在暖机运行状态会变成不满 100% 的值。
- 依据专用输入输出信号确认  
在暖机运行状态，专用输出信号 MnWUPMD（（n=1~3：机器号码）会被输出。

### ■ 通常状态 / 暖机状态的切换

暖机运行状态的时候，暖机运行模式的对象轴动作持续，此动作时间超过暖机运行状态的有效时间的话，暖机运行状态会被解除变成通常的状态。之后，因为机器人停止持续的话，在低温下关节部份会冷却，对象轴长时间停止持续，暖机运行状态的再开始时间超过的情况下，会从通常状态再度切换到暖机运行状态。

#### • 暖机运行状态的解除

对象轴的动作时间累积值超过暖机运行状态的有效时间的话，暖机运行状态会被解除变成通常的状态。暖机运行状态的有效时间以参数 **WUPTIME** 的第 1 要素指定。（工厂出货时的设定为 1 分）

对象轴为复数的情况下，在全部的对象轴超过有效时间时，暖机运行状态会被解除。

而且，依据状态变量 **M\_Wuprt**，可以确认之后对象轴动作还有多少时间及确认有否解除暖机运行状态。

#### • 从通常的状态切换到暖机运行状态

对象轴的停止持续时间，超过暖机运行状态的再开始时间的话，会从通常的状态切换到暖机运行状态。暖机运行状态的再开始时间以参数 **WUPTIME** 的第 2 要素指定。（工厂出货时的设定为 60 分）

对象轴为复数的情况下，只要有其中一轴超过暖机运行的再开始时间的话，会成为暖机运行状态。

而且，依据状态变量 **M\_Wupst**，可以确认对象轴还有多少停止持续时间及确认有否切换到暖机运行状态。

注) 即使 **ROBOT** 有动作，但是对象轴没有动作的情况，则判断为停止。

将通常状态 / 暖机运行状态的切换例子以时序表来表示，如图 5-9 所示。

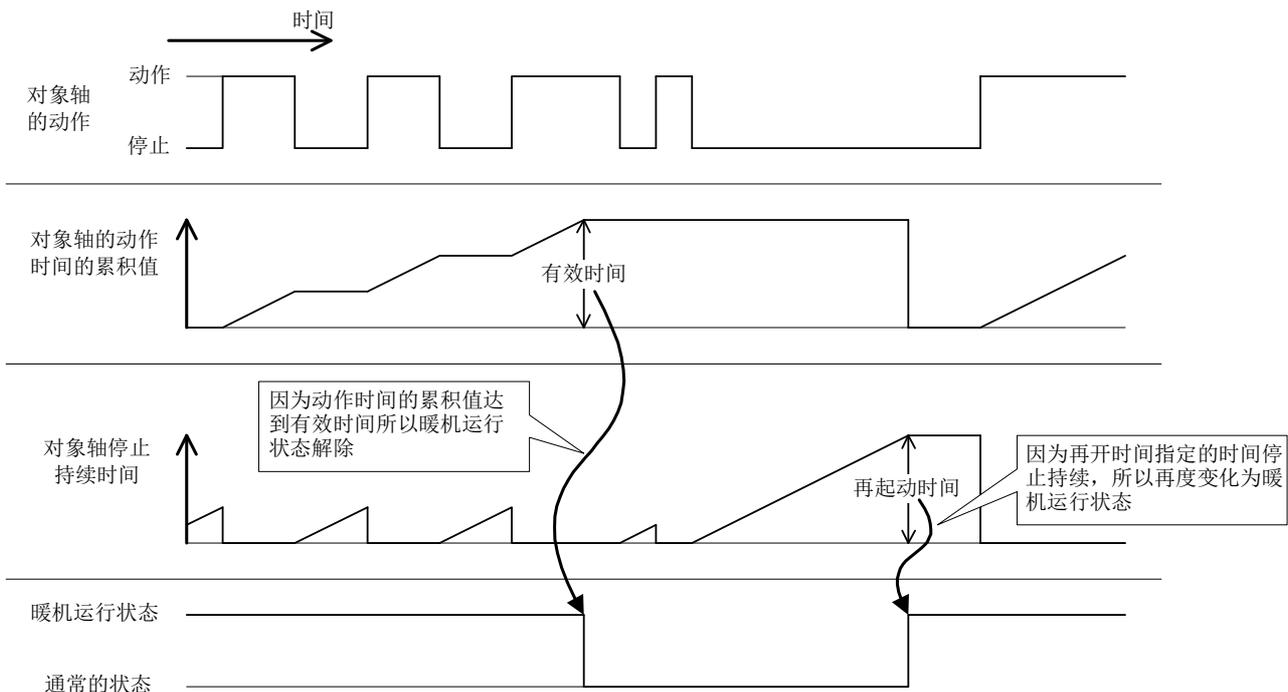


图 5-9: 通常的状态 / 暖机运行状态的切换例子

### ■ 关于暖机运行速度比例的值

暖机运行状态的时候，为了降低速度所以将动作速度的速度比例称为暖机运行速度比例。暖机运行速度比例是根据对象轴动作的时间，如下图所示进行变化，实时反映在机器人的动作上。

对应暖机运行速度比例的初始值和暖机运行状态的有效时间，速度比例不变化时间的比例，以参数 WUPOVRD 指定。（工厂出货时的设定，初始值为 70%、比例为 50%（=30 秒））

此外，此值可以用状态变量 M\_Wupov 做确认。

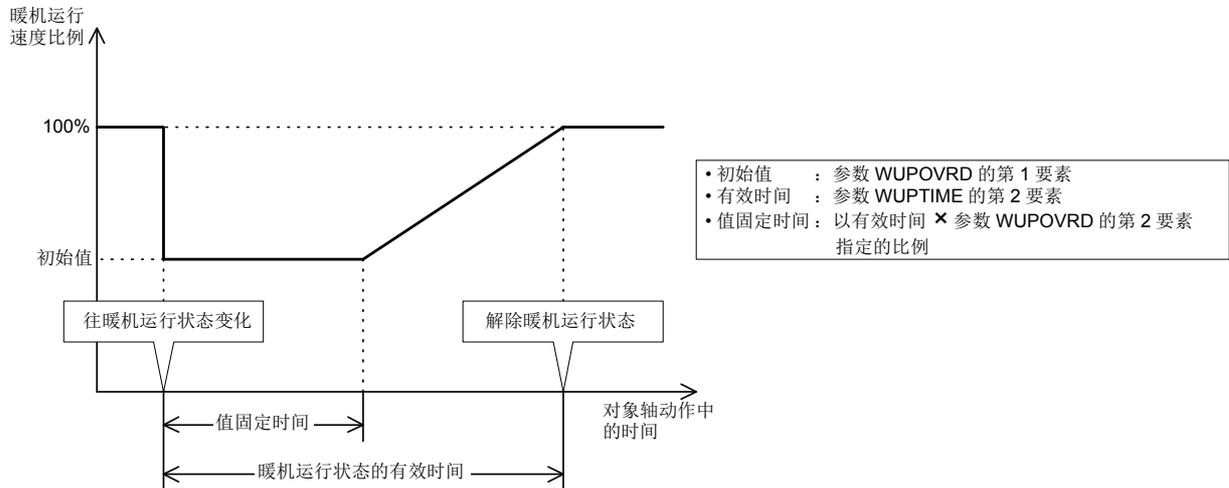


图 5-10: 暖机运行速度比例的变化

在暖机运行状态中，实际的速度比例如下所示。

- 关节插补动作时 = (操作面板 (T/B) 的速度比例设定值) × (程序速度比例 (OvrD 指令)) × (关节速度比例 (JOvrD 指令)) × 暖机运行速度比例
- 直线插补动作时 = (操作面板 (T/B) 的速度比例设定值) × (程序速度比例 (OvrD 指令)) × (直线指定速度 (Spd 命令)) × 暖机运行速度比例

注 1) 将控制器前面的 MODE 切换开关设定为 "MANUAL" 的情况及依据 JOG 动作、Realtime 外部控制 (Mxt 指令) 动作的情况下，不会反映在暖机运行速度比例，而以本来的指定速度动作。

注 2) 在暖机运行状态，因为以比本来的速度还低速的动作，因此请确实的实施与周边装置的互锁。

注 3) 对象轴为复数的情况下，暖机运行速度比例会算出对象轴中使用最小的动作时间。某对象轴不动作，状态变量 M\_Wuprt 的值不变化的情况下，其它的对象轴无论怎么动作，暖机运行速度比例的值也不会有变化。

此外，依据各对象轴的动作 / 停止的状况，值在变成 100% 前，会返回到初始值。

例如：暖机运行速度比例比初始值大的时候，某对象轴从通常的状态切换到暖机运行模式的话，因为其轴的动作时间为最小，（动作时间为 0 秒），暖机运行速度比例会返回到初始值。

### (3) 报警发生的情况

#### 1) 即使在暖机运行状态也发生误差过大的报警

- 在暖机运行速度比例为初始值时，变成报警的情况下，请将初始值的值（参数 WUPOVRD 的第 1 要素）调小。
- 暖机运行速度比例在往 100% 增加中发生报警的情况，应该是暖机运行状态的有效时间或者值固定的时间过短。请将参数 WUPTIME 的第 1 要素（有效时间）或者 WUPOVRD 的第 2 要素（值固定的时间比例）的值调大。
- 即使执行上述的处理仍然无法解除报警的情况下，请将动作程序做变更，速度·加减速度调降。

#### 2) 暖机运行状态被解除后发生误差过大的报警

- 请将参数 WUPTIME 的第 1 要素值调大、暖机运行状态的有效时间延长。
- 请确认机器人的负载及周围温度是否在规格范围内。

- 暖机运行状态解除后，请确认对象轴是否长时间停止持续。这样的情况下，请将参数 WUPTIME 的第 2 要素值调小、再度变成暖机运行状态的时间缩短。
  - 即使执行上述的处理仍然无法解除报警的情况下，请将动作程序做变更，速度·加减速度调降。
- 3) 无论何时都无法解除暖机运行状态
- 请确认全部动作全无动作关节轴是否为暖机运行模式的对象轴，参数 WUPAXIS 的设定值。
  - 请确认对象轴是否比暖机运行状态的再开始时间（参数 WUPTIME 的第 2 要素）长时间停止持续。
  - 请确认被指定的速度是否以非常地低速（以关节插补时的速度比例大概 3 ~ 5%）的方式持续动作。指定速度为低速的情况下，因为没有必要使用暖机运行模式，请使暖机运行模式为无效。

### 5.20 关于特异点通过功能

#### (1) 功能的概要

三菱公司的机器人，是使用直交坐标系中的位置数据进行直线插补动作的运算及示教位置的存储等。例如垂直 6 轴型的机器人，虽然是通过 X、Y、Z、A、B、C 的坐标值来表示位置数据，但即使是相同位置的数据，机器人也可以采取多个姿势。因此，通过与坐标值一起使用结构标志（表示姿势的标志），从多个姿势中将机器人的姿势确定为一个。

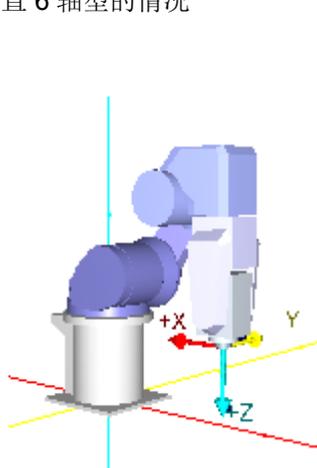
但是，即使使用结构标志，在该标志的切换位置上，有的关节轴的取得角度有无限个组合方式，因此无法让机器人在所希望的位置上所希望的姿势进行动作（例如，垂直 6 轴型的机器人的情况下，J5 轴 = 0 度时，J4 轴与 J6 轴无法确定）。该位置称为特异点，原先不可以以直交 JOG 或直线插补等方式通过。因此，目前为止都是通过调整布局使特异点不在作业区域中，或无论如何都需要通过特异点时，通过关节插补进行动作等措施加以对应。

特异点通过功能是指，可以通过直交 JOG 或直线插补等方式通过特异点的功能，通过该功能，可以以直线插补方式扩大作业区域以提高布局的自由度。

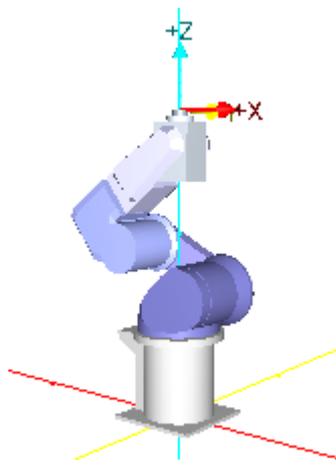
#### ■ 可通过的特异点位置

可以以特异点通过功能通过的特异点的位置如下所示。

#### 垂直 6 轴型的情况

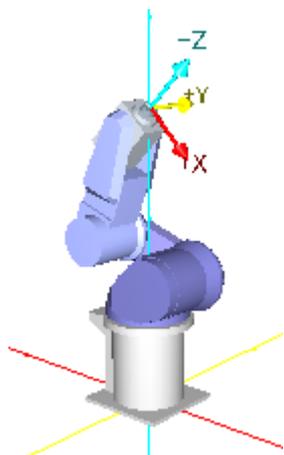


① J5 轴 = 0 度的位置  
结构标志的 NonFlip/Flip 切换的位置。



② J1 轴的旋转轴上存在 J5 轴中心的位置  
结构标志的 Right/Left 切换的位置。

#### 垂直 5 轴型的情况

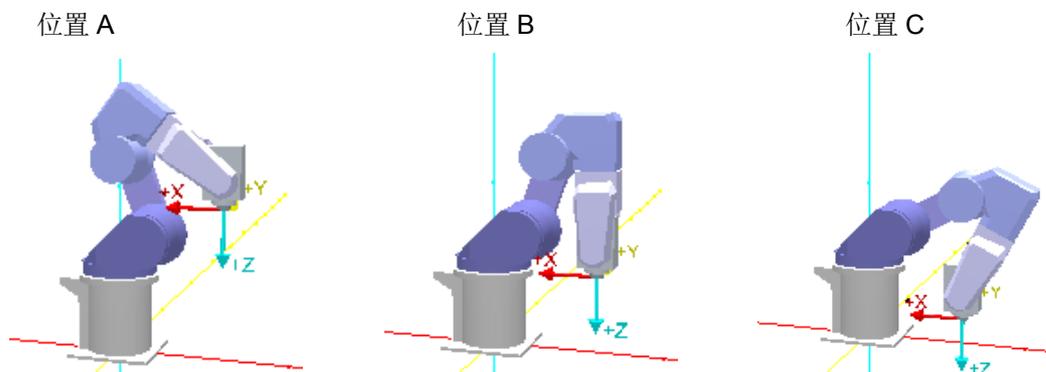


① J1 轴的旋转轴上存在机械接口中心的位置  
结构标志的 Right/Left 切换的位置。

### ■ 特异点通过功能有效时的动作

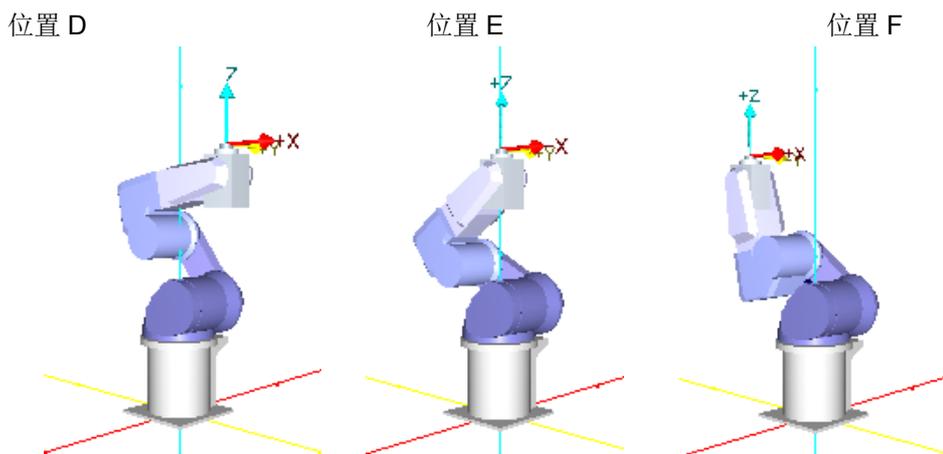
将特异点通过功能设为有效后，可以通过直交 JOG 或直线插补等方式从位置 A 经由位置 B（特异点位置）动作至位置 C（反过来也一样）。此时，在通过位置 B 前后，结构标志的值会切换。

特异点通过功能无效（或不支持）的情况下，在从位置 A 向位置 B 动作之前会因错误而停止。直交 JOG 的情况下，将在到达位置 B 之前停止。

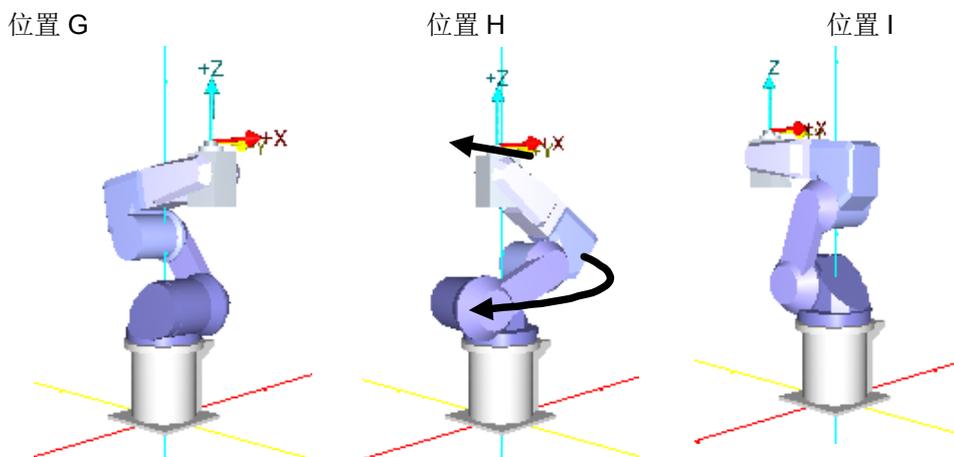


可以通过特异点的是机器人的动作轨迹通过特异点之上的情况。动作轨迹偏离特异点之上（通过特异点近旁）的情况下，动作时将不切换结构标志的值。

- 位置 D → E → F：通过特异点的情况（在位置 E 的前后，结构标志进行切换。）



- 位置 G → H → I：通过特异点近旁的情况（结构标志不进行切换。）





## 注意

通过特异点近旁时，机器人有可能如上图的位置 H 一样发生大旋转。进行位置示教等靠近机器人进行作业时，目光不得离开机器人，并应充分注意不可与其接触。

### ■ 为了使用特异点通过功能

要通过 JOG 操作使用特异点通过功能时，必须将参数 FSPJOGMD 指定为 1（有效），并重新接通控制器的电源。要通过自动运行使用时，则必须通过插补指令的 TYPE 指定将常数 2 指定为 2。

### ■ 对象机型

特异点通过功能对应的机型如下所示。

| 对应机型                                                               |
|--------------------------------------------------------------------|
| RV-2F、RV-4F 系列、RV-4FJL、RV-7F 系列、RV-7FLL、RV-13F 系列、RV-20F、RV-50F 系列 |

非对象机型中，即使将特异点通过功能设定为有效，JOG 操作也会为常规的动作（维持姿势不变、且不切换结构标志的动作），自动运行则会变为错误。

### ■ 限制事项

使用特异点通过功能时，存在以下的限制事项。

- (1) 将附加轴用作多机械时，不能使用特异点通过功能。
- (2) 使用机器人附加轴的同步控制时，不能使用特异点通过功能。
- (3) 弹性模式有效时，不能使用特异点通过功能。
- (4) 碰撞检测功能有效时，不能使用特异点通过功能。
- (5) 维护预报的信息收集等级应设为等级 1（出厂时的设定）。

## (2) JOG 动作的特异点通过功能

JOG 动作的特异点通过功能为通过参数 FSPJOGMD 指定有效（1）/无效（0）。

| FSPJOGMD      | 直交 JOG       | 工具 JOG       | 工件 JOG       | 3 轴直交 JOG                         | 圆筒 JOG                        | 关节 JOG  |
|---------------|--------------|--------------|--------------|-----------------------------------|-------------------------------|---------|
| 0<br>(出厂时设定值) | 常规的直交 JOG    | 常规的工具 JOG    | 常规的工件 JOG    | 设定值不影响。<br>(XYZ 将维持姿势不变，不切换结构标志。) | 设定值不影响。<br>(将维持姿势不变，不切换结构标志。) | 设定值不影响。 |
| 1             | 特异点通过的直交 JOG | 特异点通过的工具 JOG | 特异点通过的工件 JOG |                                   |                               |         |

① 常规的直交 / 工具 / 工件 JOG 是指维持姿势不变、不切换结构标志的 JOG 动作。

② 无法使用特异点通过功能的机器人中，即使更改了参数 FSPJOGMD 的设定值，也为常规的动作（维持姿势不变、不切换结构标志的动作）。

特异点通过功能对应的机型为 RV-F 系列。（详细内容参照上述的「**■对象机型**」）。

③ 特异点通过中，不能让多个轴同时进行 JOG 动作。某个轴在动作中时，即使想要使其他轴动作，该动作也将被忽略。

④ 使用示教单元进行 JOG 动作时，若靠近特异点，则会发出特异点近旁警告。关于特异点近旁警告，请参照第 458 页的「5.17 关于特异点接近警告」。

⑤ 在通过专用输入信号进行的 JOG 动作中，也将反映参数 FSPJOGMD 的指定。

- (3) 位置数据的确认操作（跳位）的特异点通过功能  
位置数据的确认操作（跳位）中也将反映参数 FSPJOGMD 的指定。

| FSPJOGMD      | MO 位置移动                  | MS 位置移动                       |
|---------------|--------------------------|-------------------------------|
| 0<br>(出厂时设定值) | 设定值不影响。<br>通过关节模式进行位置移动。 | 常规的位置移动<br>(将维持姿势不变、不切换结构标志。) |
| 1             |                          | 特异点通过的位置移动                    |



## 注意

通过示教单元直接执行插补指令（例：Mvs P1）时，参数 FSPJOGMD 变为 1（有效）时，则 Type 指定中即使未指定特异点通过，也将以特异点通过有效进行动作。

- (4) 自动运行的特异点通过功能  
要在自动运行中使用特异点通过功能，必须对每个对象插补指令通过 Type 指定进行指定。

## Type

### 【功能】

通过插补指令的 Type 指定来指定特异点通过功能。所对应的插补指令为直线插补（Mvs）、圆弧插补（Mvr、Mvr2、Mvr3）。关于可进行特异点通过的指定的机型，请参照第 469 页的「[对象机型](#)」。

### 【格式】

Type □ < 常数 1>, < 常数 2>

### 【用语】

<常数 1>            0/1    : 抄近道/绕远道动作  
<常数 2>            0/1/2: 等效旋转/三轴直交/特异点通过

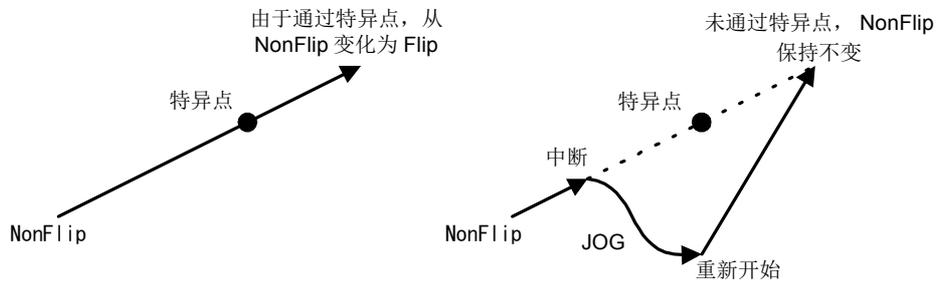
### 【例文】

1 Mvs P1 Type 0,2            ' 从当前位置起至 P1 为止，以特异点通过有效进行直线插补动作  
2 Mvr P1,P2,P3 Type 0,2       ' P1 ~ P3 为止，以特异点通过有效进行圆弧插补动作

### 【说明】

- (1) 无法使用特异点通过功能的机器人中将 < 常数 2> 指定为 2 时，执行时会发生错误。
- (2) 指定了特异点通过时，对始点·终点间的结构标志不进行检查。此外，由于无法确定目的位置的结构标志，因此对动作开始前的目的位置·中间位置的动作范围不进行检查。
- (3) 通过 Spd 指令指定了速度时，将上限作为指定速度，在特异点近旁自动降低至不会发生速度错误的速度来进行动作。
- (4) 指定了特异点通过的插补指令中不适用最佳加减速，而是以固定加减速进行动作。此时，因 Accel 指令的指定而导致加速时间与减速时间不相同，将较长的时间用于加速、减速两方。
- (5) 指定了特异点通过的插补指令中不适用 Cnt 指令的指定，以有加减速来进行动作。
- (6) 圆弧中，当前位置与始点位置不相同，即使通过 Type 指定指定了特异点通过，也会按照之前的三轴直交的直线插补来进行动作，直至始点。

(7) 将指定了特异点通过的插补动作中断，并通过 JOG 移动后重新开始动作时，将按照参数 RETPATH 进行动作，直至中断位置。参数 RETPATH 为 0（无效：不返回中断位置）的情况下，如下图所示重新开始后的动作路径如果不通过特异点，则结构标志不进行切换，插补结束时机器人的姿势有可能与不中断时的动作有所不同。



(8) 指定了特异点通过时，与常规的直线插补等相比，动作速度有可能变慢。此外，由于特异点通过功能要进行复杂的处理，因此有可能会影响程序执行速度。特异点通过的指定，应仅通过必要的插补指令来进行。

## 5.21 关于冲突检知功能

### (1) 功能的概要

使机器人执行各式各样的动作时，依据作业者的操作错误及动作程序的错误等，会造成机器人和周边装置干涉的情况。像这样的情况，以往是以控制机器人的电机驱动伺服保护功能（误差过大检知等）让机器人停止，将机器人的抓手和手臂部份、工件、周边装置等的损伤减轻。然而，依据机器人的高速化、使用电机的大容量化，干涉时所花费的力量比原来还大，只以伺服的保护功能来防止机器人的损伤会变得困难。

冲突检知功能，比以往的伺服保护功能更高感度的检知到干涉，依据更早使机器人停止，将损伤减轻为目的的功能。



**警告**

即使冲突检知功能为有效，也无法防止动作中的机器人和所接触到的作业者的负伤。不论冲突检知功能的有效 / 无效，请依照所规定的安全守则执行作业。



**注意**

即使使冲突检知功能为有效，也无法百分之百的阻止与周边装置的干涉而造成机器人、抓手、工件等的损伤。基本上请不要与周边装置发生干涉小心注意的操作、运用机器人。

### ■ 干涉的检知原理

机器人与周边装置等发生干涉的话，针对各关节轴的位置指令不追随实际的位置，因此依据伺服的反馈控制会发生更大的转矩。干涉没有被解除的话，发生的转矩会更增加，没有发生干涉的话，会成为更大差异的值。冲突检知功能，利用此伺服的特性，可以检知到干涉。首先，依据现在的位置指令各负载设定，在各关节轴将转矩依次比较。然后，其差值超过容许范围（检知 Level）的时候，会判断为干涉，伺服关闭且使机器人紧急停止。

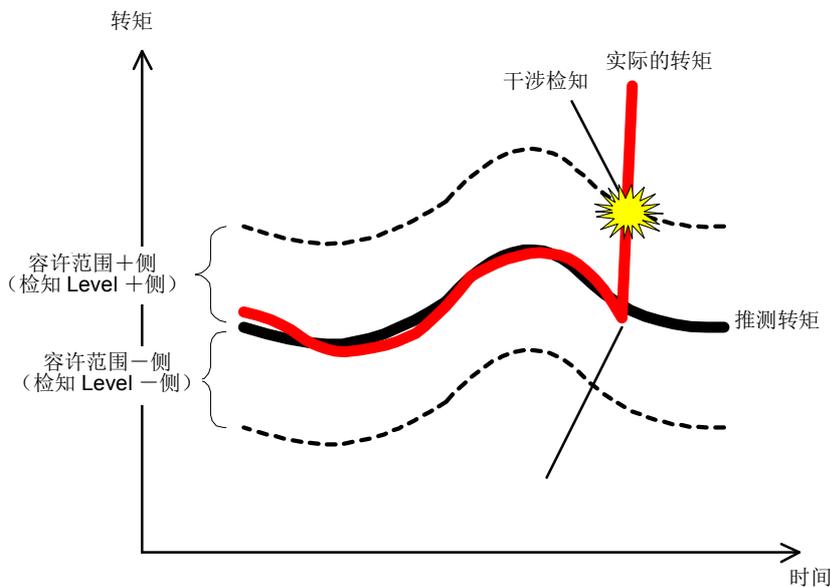


图 5-11：干涉的检知原理

## (2) 关连参数

关于冲突检知功能相关参数，有下列几种。关于详细内容请参照第 407 页的「5.1 动作参数」、第 456 页的「5.16 关于抓手、工件条件设定（最佳加减速设定）」。

表 5-21: 冲突检知功能相关参数

| 参数名                   | 说明和值                                                                                                                                                          | 出货时设定值                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| COL                   | 定义冲突检知功能的使用可 / 不可和电源投入后直接的有效 / 无效。<br>要素 1: 指定冲突检知功能的使用可 (1) / 不可 (0)<br>要素 2: 指定程序运行时的初始状态有效 (1) / 无效 (0)<br>要素 3: 指定 JOG 操作时的有效 (1) / 无效 (0) / NOERR 模式 (2) | RH-3FH/6FH/<br>12FH/20FH 系<br>列: 1,0,1<br>RV-F 系列: 0,0,1<br>RH-3FHR 系列:<br>1,1,1 |
| COLLVL                | 设定在程序运行时，各关节轴的各轴的检知 LEVEL (敏感度) 的初始值。<br>此值会成为冲突检知功能中所规定的 Level 基准值的倍率。<br>值小时，检知 LEVEL 会变高。<br>设定范围: 1 ~ 500 单位: %                                           | 依机型而有所不同。                                                                        |
| COLLVLJG              | 设定在 JOG 操作时 (含中断中) 各关节轴的检知 LEVEL (敏感度)。<br>此值会成为冲突检知功能中所规定的 Level 基准值的倍率。<br>值小时，检知 Level 会变高。<br>设定范围: 1 ~ 500 单位: %                                         | 依机型而有所不同。                                                                        |
| HNDDAT*<br>* 为 0 ~ 8  | 设定抓手的条件。(以 TOOL 坐标指定。)<br>HNDDAT0 电源一开启后，马上被视为初始条件采用。<br>(重量, 大小 X, 大小 Y, 大小 Z, 重心 X, 重心 Y, 重心 Z) 单位: kg, mm                                                  | 依机型而有所不同。                                                                        |
| WRKDAT*<br>* 为 0 ~ 8  | 设定工件的条件。(以 TOOL 坐标指定。)<br>WRKDAT0 电源一开启后，马上被视为初始条件采用。<br>(重量, 大小 X, 大小 Y, 大小 Z, 重心 X, 重心 Y, 重心 Z) 单位: kg, mm                                                  | 0.0, 0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0                                             |
| HNDHOLD*<br>* 为 1 ~ 8 | 执行 HOpen, HClose 指令的时候，指定工件的把持 (1) / 未把持 (0)。<br>要素 1: 指定 HOpen 指令执行时的状态。<br>要素 2: 指定 HClose 指令执行时的状态。                                                        | 0,1                                                                              |

### (3) 使用冲突检知功能时

在使用冲突检知功能时，先在参数 COL 的要素 1 指定为「使用可 (1)」，然后再投入控制器的电源。之后，对以 JOG 操作时和程序运行时执行各别检知功能的设定（有效 / 无效的指定和检知 Level）。

（请合并参照第 473 页的「表 5-21：冲突检知功能相关参数」。）

#### 1) JOG 操作时的使用方法

JOG 操作时的冲突检知功能，全部依据参数执行设定。因此，即使在控制器的电源投入中变更有效 / 无效等设定，到电源再投入为止都不会反应变更。执行设定的参数如表 5-22 所示。

表 5-22：在 JOG 操作时的冲突检知功能设定参数

| 参数名      | 说明和值                                                                                                                                       | 出货时设定值                                                                             |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| COL      | 定义冲突检知功能的可使用 / 不可使用和电源投入之后的有效 / 无效。<br>要素 1：冲突检知功能的使用可 (1) / 不可 (0) 设为 1。<br>要素 3：指定 JOG 操作时的有效 1) / 无效 (0)。有效 (1) / 无效 (0) / NOERR 模式 (2) | RH-3FH/6FH/<br>12FH/20FH 系<br>列：1,0,1<br>RV-F 系列：<br>0,0,1<br>RH-3FHR 系<br>列：1,1,1 |
| COLLVLJG | 在 JOG 操作时（含中断中）设定各关节轴的检知 Level（检知的敏感度）。                                                                                                    | 依机型而有所不同。                                                                          |
| HNDDATO  | 设定抓手的条件。（在 TOOL 坐标设定。）                                                                                                                     | 依机型而有所不同。                                                                          |
| WRKDAT0  | 设定工件的条件。（在 TOOL 坐标设定。）                                                                                                                     | 0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0, 0.0                                               |

#### ■ 碰撞检测等级的调整

JOG 操作时的检测等级（检测的灵敏度）设定得较低。需要更高的检测等级时，请调整参数 COLLVLJG 来使用。此外，为了高精度进行转矩的推断，也请正确设定参数 HNDDATO、WRKDAT0。

#### ◆◆◆ 重点 ◆◆◆

将检知 Level 调整太高（设定值过小）的话，依据机器人的位置和姿势，会有错误检知的情况。像这样的情况下，请将检知 Level 调低（设定值调高）。

#### ■ 检知到干涉时的动作

JOG 操作中检知到与周边装置等的干涉的话，会发生 1010 号（第 1 行为轴号码）的报警且机器人伺服关闭、动作停止。而且，NOERR 模式的情况下（在参数 COL 的要素 3 设定为 2 的情况），不会发生报警、伺服关闭后动作停止。（但是，在报警历史里会留下 1010 号台的报警。）

#### ■ 干涉后的操作

抓手及手臂与周边装置等维持在干涉的状态，伺服开启的话，会变成再度冲突检知状态，会有无法开启伺服的情况。即使重复伺服开启也还是变成报警的情况下，暂时将刹车解除，执行 JOG 操作使手臂逃避，请解除干涉状态。

#### ■ JOG 操作中使冲突检知暂时无效的方法

按住示教单元的「RESET」键进行伺服 ON 及 JOG 操作时，在按住「RESET」键期间，冲突检知无效。

## 2) 程序运行时的使用方法

程序运行时的冲突检知功能，以参数设定初始状态，但是实际上是使用 MELFA-BASIC V 的指令在程序中变更设定。以初始状态设定参数和冲突检知功能相关指令，各别如下表所示。而且，关于指令的详细请参照第 163 页的「4.13 指令的详细说明」、第 287 页的「4.14 机器人（系统）状态变量的详细说明」。

表 5-23: 在程序运行时的冲突检知功能设定的参数

| 参数名                   | 说明和值                                                                                                                  | 出货时设定值                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| COL                   | 定义冲突检知功能的可使用 / 不可使用和电源投入之后的有效 / 无效。<br>要素 1: 冲突检知功能的使用可 (1) / 不可 (0) 设为 1。<br>要素 2: 程序运行时的初始状态的有效 (1) / 无效 (0) 设定在 1。 | RH-3FH/6FH/<br>12FH/20FH 系<br>列: 1,0,1<br>RV-F 系列: 0,0,1<br>RH-3FHR 系列:<br>1,1,1 |
| COLLVL                | 设定在程序运行时各关节轴的检知 Level (检知的敏感度) 的初始值。                                                                                  | 依机型会有所不同。                                                                        |
| HNDDAT*<br>* 为 0 ~ 8  | 设定抓手的条件。(在 TOOL 坐标设定。)                                                                                                | 依机型会有所不同。                                                                        |
| WRKDAT*<br>* 为 0 ~ 8  | 设定工件的条件。(在 TOOL 坐标设定。)                                                                                                | 0.0, 0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0                                             |
| HNDHOLD*<br>* 为 1 ~ 8 | 设定执行 HOpen, HClose 指令的时候的工件把持 (1) / 未把持 (0)。                                                                          | 0,1                                                                              |

表 5-24: 在程序运行时的冲突检知功能使用 MELFA-BASIC V 指令和状态变量

| 指令 / 状态变量 | 说明                                                                                                                                                          |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ColChk    | 指定冲突检知功能的有效 / 无效 / NOERR<br>例: ColChk On ' 使冲突检知功能为有效。                                                                                                      |
| ColLvl    | 在每个关节轴设定冲突检知功能的检知 Level (检知的敏感度)。此数值和参数 ColLvl,,, 相同, 成为在冲突检知功能中所规定的检知 Level 基准值的倍率。(单位: %)<br>例: ColLvl 80, 120, 120, 120, 50, 80, ' 设定 J1 ~ J6 轴的检知 Level |
| LoadSet   | 指定抓手、工作的条件。程序运行中使用的抓手及把持工件变更情况下使用。<br>例: LoadSet 1, 0 ' 指定参数 HNDDAT1、WRKDAT0 的条件                                                                            |
| J_ColMxl  | 将假定转矩和实际的转矩的差的最大值变换为检知 Level 返回。在 ColLvl 指令的自变量调整时参照。(单位: %)                                                                                                |
| M_ColSts  | 在检知到干涉时将 1 返回。使用在 NOERR 模式的插入条件。                                                                                                                            |
| P_ColDir  | 返回将干涉检知时的机器人的动作方向 (X,Y,Z 的动作比例。在 NOERR 模式以退避动作使用。                                                                                                           |

## ◆◆◆ 重点 ◆◆◆

跨越整个程序使冲突检知功能为有效的话，伴随着所发生的错误检知的可能性也会变高。因为要误错误检知消失的话必须将检知 Level 降低，其结果，对冲突检知需要的动作来说，干涉的检知敏感度会变低。所以为了使检知敏感度提高，请只对其可能干涉的动作使用冲突检知功能。

## ◆◆◆ 重点 ◆◆◆

使冲突检知功能为有效的话，依据程序节拍时间 (Tact Time) 会有延长的情况。为了减低对 Tact time 的影响，请不要跨越整个程序使冲突检知为有效，而只要对其可能干涉的动作使用冲突检知功能。

### ■冲突检知 Level 的调整

对应机器人的动作，请调整程序运行时的检知 Level（检知的敏感度）。参考以前，调整步骤例如下所示。而且，为了将转矩的假定精度变好，请正确的设定工件条件和抓手条件。

表 5-25: 程序运行时的检知 Level 调整步骤的一例

| 步骤 | 内容                                                                                                                |
|----|-------------------------------------------------------------------------------------------------------------------|
| 1  | 在使用冲突检知功能动作的前后追加 ColLvl 指令和 ColChk 指令。                                                                            |
| 2  | 为了不误检知干涉，请将检知 Level 调低。（ColLvl 指令的自变量 300 等高的值）。                                                                  |
| 3  | 运行程序并监视变成对象的动作的 J_ColMxl 的值。此外，因为数值很分散，所以成为对象的动作复数次循环，并每次都控制 J_ColMxl 的值。                                         |
| 4  | 从复数的 J_ColMxl 的值开始求出在各关节值的最大值，在其值中加上极限（例如 20%）会作为 ColLvl 指令的自变量。                                                  |
| 5  | 将在步骤 4 求得的值设定在 ColLvl 指令并运行程序，使用冲突检知功能确认动作有无错误检知。错误检知的情况下，请将 ColLvl 指令自变量值慢慢提高，直到不会发生错误检知 Level 为止，再将检知 Level 降低。 |

#### ◇◆◇ 重点 ◇◆◇

将动作速度变更的话，检知 Level 的变更也变成必须变更。调整检知 Level 的时候，请将机器人以实际运用的速度做调整动作。

#### ◇◆◇ 重点 ◇◆◇

在复数台的机器人使用检知功能的时候，由于电机特性的偏差等使机器人的个体差和依据使用环境，即使同一动作也会需要调整检知 Level。另外，依据机器人的机型有所不同的情况，有必要调整每个检知 Level。

使用的控制器的软件版本为 R6b 版 /S6b 版以上、且使用 RT ToolBox2 Ver. 3.40S 以上时，可通过 RT ToolBox2 的示波器功能选择“碰撞检测等级参考值”。由此可以实现 ColLvl 指令设定值的调整作业简化。详细内容请参照第 477 页的「3）补充事项」。

### ■程序例

检知到干涉的话，依据中断处理执行退避动作。

```

1 Def Act 1,M_ColSts (1) =1 GoTo *HOME,S' 定义检知到干涉的时候将执行处理插入
2 Act 1=1
3 ColLvl 80,120,120,100,80,80,, ' 设定检知 Level
4 ColChk On,NOErr ' 在 NOERR 模式使冲突检知功能为有效
5 Mov P1
6 Mov P2 ' 在 5~8 行执行检知到干涉的话，跳转到插入处理
7 Mov P3
8 Mov P4
9 ColChk Off ' 使冲突检知为无效
10 Act 1=0
 :
11 *HOME ' 检知到干涉的中断处理
12 ColChk Off ' 使冲突检知为无效
13 Servo On ' 伺服开启
14 PESC=P_ColDir (1) * (-5) ' 算出退避动作的移动量（约 5mm 的反转动作）
15 PDST=P_Fbc (1) +PESC ' 制作退避位置
16 Mvs PDST ' 往退避位置移动
17 Error 9100 ' 让使用者定义的 L Level 报警发生中断
 :
```

3) 补充事项

■关于 JOG 操作时和程序运行时的区别

在 JOG 操作时和程序运行时，因为机器人的动作速度和作业内容会有很大的差异，所以为了让针对各别的冲突检知功能做最佳的动作，会变成独立设定。  
在此，所谓的“JOG 操作时”和“程序运行时”，是指以下的状态。

- JOG 操作时：JOG 动作中、自动运行中断中
- 程序运行时：自动运行中、单步前进 / 返回动作中，位置数据的确认动作中。

以上如图 5-12 一样的状态迁移。

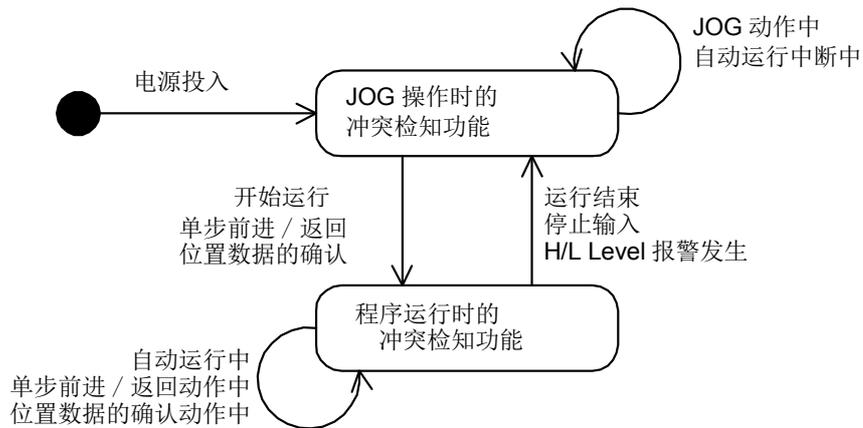


图 5-12: 程序运行时 /JOG 操作时的相关切换状态迁移

例如 JOG 操作时的冲突检知功能为有效的情况，在程序运行中即使将冲突检知设定为无效，只要按下停止钮，中断程序的话，会转换为 JOG 操作时的设定，冲突检知会变成有效。

■伺服关闭中的冲突检知功能

JOG 操作时和程序运行时都一样在伺服关闭中，冲突检知功能会暂时的变成无效。

■关于示波器功能的“碰撞检测等级参考值”

使用的控制器的软件版本为 R6b 版 /S6b 版以上、且使用 RT ToolBox2 Ver. 3.40S 以上时，可通过 RT ToolBox2 的示波器功能的实时监视器选择“碰撞检测水平参考值”。

“碰撞检测等级参考值”是通过碰撞检测水平 (CoILvl 指令或参数 COLLVL、COLLVLJG 的设定值) 表示的推断转矩与实际转矩的差值，并作为确定设定值时的参考值。

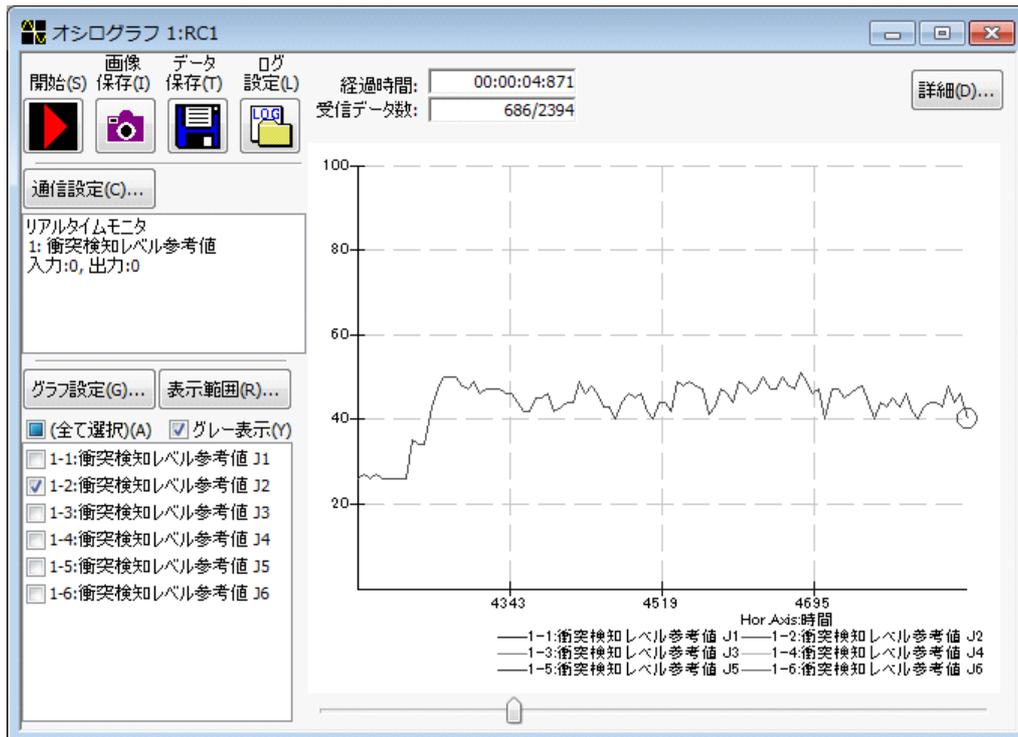


图 5-13: “碰撞检测等级参考值”的示波器显示

碰撞检测功能有效且伺服 ON 时，将如图 5-13 所示，碰撞检测等级会实时输出至示波器。对于将碰撞检测设为有效的动作，参考该值来设定碰撞检测等级。调整步骤的一个示例如下所示。此外，为了使转矩推断精度更好，应正确设定工件条件与抓手条件。

表 5-26: 程序运行时的碰撞检测等级调整步骤的一个示例

| 步骤 | 内容                                                                                                     |
|----|--------------------------------------------------------------------------------------------------------|
| 1  | 在使用碰撞检测功能的动作前后，增加 ColLvl 指令与 ColChk 指令。                                                                |
| 2  | 为了不错误检测出干涉，要降低 (ColLvl 指令的自变量中使用 300 等较大的值) 设定检测等级。                                                    |
| 3  | 运行程序，用示波器监视对象动作中的“碰撞检测等级参考值”。此外，数值有时会有偏差，因此要反复多次进行对象动作，记录下各轴的最大值。                                      |
| 4  | 将通过步骤 3 求出的各轴的最大值加上余量 (例如 20%) 得出的数值用作 ColLvl 指令的自变量。                                                  |
| 5  | 将通过步骤 4 求出的值设定在 ColLvl 指令中以运行程序，并通过使用碰撞检测功能的动作确认无错误检测。出现错误检测时，要逐渐加大 ColLvl 指令的自变量的值，降低检测等级，直至不再出现错误检测。 |

## 5.22 关于过负载Level的最佳化

当所使用机器人的实际周边温度在 40 °C 以下时，通过将实际的周边温度设定到参数：OLTMX 中，可使机器人动作时工作的过负载报警的检知 Level（对电机发热给予保护的功能）根据客户的使用环境达到最佳化，进一步提升连续动作性。

【对象机种】：RH-F 系列 /RV-F 系列

表 5-27 所示为参数：OLTMX 的说明。

机器人的周边温度被控制在 40 °C 以下时，请将实际的周边温度设定到本参数，有效使用机器人。

表 5-27：相关参数

| 参数               | 参数名   | 排列数<br>文字数 | 内容说明                                                                           | 出货时设定值                                                                                   |
|------------------|-------|------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| 过负载检知 Level 的最佳化 | OLTMX | 整数 1       | 设定机器人使用环境中的周围温度的上限值。<br>以该设定值为基础使机器人动作时的过负载检知 Level 最佳化。（单位：°C）<br>设定范围：0 ~ 40 | RH-3FH/6FH/<br>12FH/20FH：40<br>RV-2F 系列：40<br>RV-4F/7F/13F/<br>20F/50F 系列、<br>RH-3FHR：30 |

注) 周边温度设定的注意事项

若全年中，机器人周边温度会发生变化的，请设定最高温度。设定的温度低于实际温度时，在发生过负载报警之前，有可能会发生电机过热报警。

通过变更周边温度参数改善连续动作性的效果因机种・动作轴而异。

有时也会出现即使变更了参数，连续动作性能也没有变化的情况。

### 5.23 Pallet定义指令的多旋转限制

J1 轴或 J4 轴可超过  $\pm 180$  度的机器人机型 (RV-2F 等) 中, 使用 Pallet 定义指令: DEFPLT 时, 无法指定 J1 轴或 J4 轴的关节角度跨  $\pm 180$  度之类的 Pallet。如果指定了跨多旋转之类的位置, 执行 Pallet 定义指令时将发生错误。通过更改参数可以执行。

表 5-28: 相关的参数

| 参数        | 参数名称    | 数组数字字符数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                         | 出厂时设定值                |
|-----------|---------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Pallet 规格 | PLTSPEC | 整数 1    | <p>对于 Pallet 指定的各点, 将多旋转位置的错误检查设为有效。</p> <p>0: 不进行多旋转位置的错误检查<br/>1: 进行多旋转位置的错误检查</p> <p>设为 1(进行多旋转位置的错误检查)时, 会像 RV-2F 机器人那样, 在 J1 轴或 J4 轴的关节角度超出 <math>\pm 180</math> 度进行多旋转的机型中, 指定了跨多旋转位置的情况下, 或指定了 J1 轴或 J4 轴进行大旋转的位置的情况下, 执行 Pallet 定义指令时会发生错误。</p> <p>设为 0(不进行多旋转位置的错误检查)时, 不会发生错误, 但在移动指令的抄近道 (TYPE) 或多旋转标志 (FL2) 的指定中, 必须创建考虑到多旋转的程序。如指定错误, 关节轴会旋转 1 周, 有可能会干涉到外围装置等。对所有点的 Pallet, 应务必在低速下进行动作确认。</p> | 1: RV-F 系列<br>0: 其他机型 |

包括抓手尖端轴 (RV 机型中为 J6 轴、RH 机型中为 J4 轴), 旋转轴 J1 轴及 RV 机型的前臂偏转轴 J4 都可以超出  $\pm 180$  度进行多旋转。

Pallet 定义指令中, 不能进行跨这些轴的多旋转动作。通过 Pallet 定义指令指定多旋转标志不同的位置而进行动作时, 在指定绕远道的关节插补中, J1 轴将转动着进行相反的动作。根据情况, 可能会发生意想不到的动作, 因此在 Pallet 定义指令中有以下的限制。

#### 1) 始点的多旋转标志与终点 A 或终点 B 或对角点的多旋转标志不同的案例

对于多旋转轴 (RV 的 J6 轴机型的情况下为 J1 轴及 J4 轴, RV 的 5 轴机型与 RH 机型的情况下为 J1 轴) 的多旋转标志, 要通过 Pallet 定义的各点进行比较确认, 即使有一个多旋转标志不相同, 执行 Pallet 定义指令时也会发生 L3750 错误, 出现错误停止。以常规 Pallet 与圆弧 Pallet 两种为对象。

#### 2) 即使 1) 的多旋转标志一致, 关节角度也有很大不同的案例

对于多旋转轴 (RV 的 J6 轴机型的情况下为 J1 轴及 J4 轴, RV 的 5 轴机型与 RH 机型的情况下为 J1 轴) 的关节坐标位置, 要对 Pallet 定义的各点与始点进行比较, 关节角度相差  $\pm 180$  度以上的情况下, 执行 Pallet 定义指令时会发生 L3760 错误, 出现错误停止。以常规 Pallet 与圆弧 Pallet 两种为对象。

作为检查多旋转的轴, 以机器人的轴 (旋转轴 J1 轴、RV 机型的前臂偏转轴 J4 轴) 为检查对象, 关于附加轴及用户机械的多旋转, 不进行上述检查。

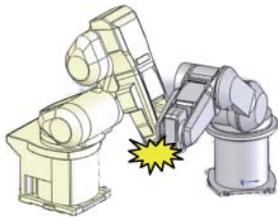
### 5.24 关于干涉回避功能

本功能是在进行干涉确认的同时使机器人动作的功能。干涉确认对象为①机器人之间（仅限 CR7xx-Q 控制器）和②机器人与自由平面极限之间。

在 JOG 操作或自动运行时，通过事先检测出机器人之间或机器人与自由平面极限之间的干涉并使其停止，可以降低对机器人及外围装置的损伤。

在事先检测到干涉时，机器人停止动作。可通过在程序中的指定，使机器人发生报警、或使其执行恢复动作。

- 【对象机种】：** 机器人之间的干涉确认仅限 RH-F/RV-F 系列的 CR7xx-Q 控制器。（不支持用户机械）  
机器人与自由平面极限之间的干涉确认仅限 RH-F/RV-F 系列。（不支持用户机械）  
※ 支持的控制器软件版本请参照第 483 页的「5.24.2 机器的准备和连接」。



#### 可回避的干涉例

利用本功能，防止如下机器人动作中的干涉于未然的控制将动作。

- 1) JOG 操作时，因误操作导致的对机器人的撞击。
- 2) 自动运行中，因程序错误导致的对机器人的撞击。
- 3) 恢复时，进入与设想动作不同的顺控程序，导致机器人之间或机器人与外围装置之间发生碰撞。
- 4) 以自动运行首次作高速动作时，因互锁问题导致冲突。
- 5) 使 2 台机器人在速度比例不同的状态下动作而导致机器人之间发生冲突。

图 5-14： 机器人动作时的干涉确认（自动运行・JOG 运行）

#### ⚠ 注意

本干涉回避功能是对机械臂、抓手、工件进行模型化处理，当这些模型之间发生了重叠时则判断为有发生干涉的危险，为避免实际发生干涉而对机器人的动作进行控制的功能。

使用时需要注意的是，使用此功能必须要登录与机器人动作相应的正确模型，因此并不能完全保证防止发生干涉。

### 5.24.1 操作步骤

下面介绍使用干涉回避功能时的操作步骤概要。

#### (1) 机器人之间的干涉确认（仅限 CR7xx-Q 控制器）

< 操作的流程 >



#### (2) 与自由平面极限之间的干涉确认

< 操作的流程 >



5.24.2 机器的准备和连接

表 5-29 中为使用本功能所需的机器，图 5-15 为其连接示例。  
 请参照图示，连接所需机器。

(1) 机器人之间的干涉确认（仅限 CR750-Q/CR751-Q 控制器）

表 5-29: 所需机器

| No. | 机器                                       | 备注                                                                                                                 |
|-----|------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| 1   | 机器人（CR750-Q/CR751-Q 控制器）<br>2 台或 3 台（最多） | 本功能使用经由 iQPlatform 的共有内存的机器人 CPU 间直接通信。<br>控制器的软件版本：<br>R3 版以上（模型形状：球形）<br>R3m 版以上（模型形状：球形、圆筒形）<br>注）请在相同的软件版本中使用。 |
| 2   | 安装了 RT ToolBox2 的计算机                     | 与机器人 CPU 连接。<br>RTToolBox2 的软件版本：2.1L 版以上。                                                                         |

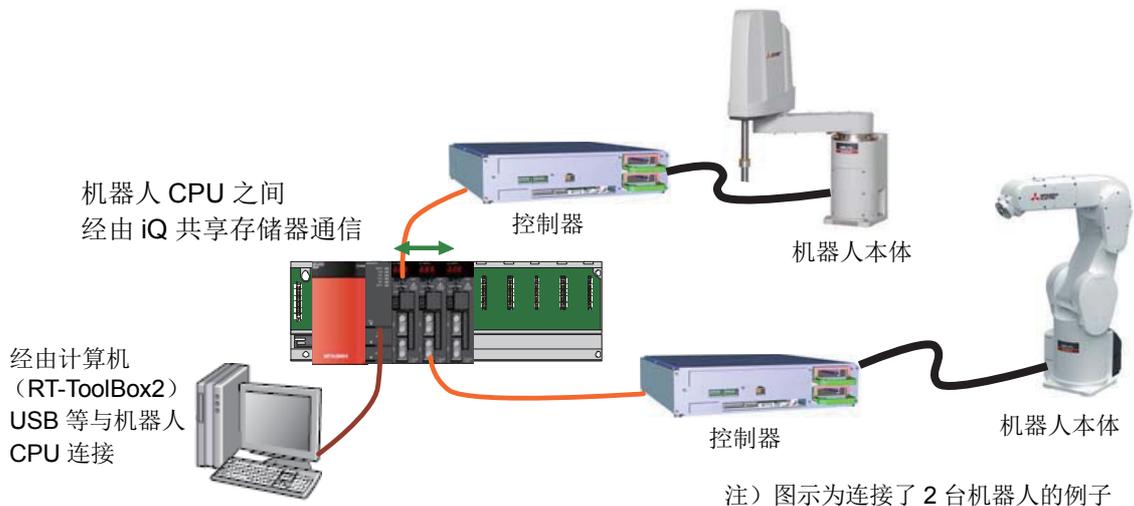


图 5-15: 机器的连接

(2) 自由平面极限之间的干涉确认

表 5-30: 所需设备

| No. | 设备                   | 备注                                                |
|-----|----------------------|---------------------------------------------------|
| 1   | 机器人 1 台              | 仅本号机中设定的自由平面极限进行干涉确认。<br>控制器的软件版本：<br>R6b/S6b 版以上 |
| 2   | 安装了 RT ToolBox2 的计算机 | 与机器人 CPU 连接。<br>RT ToolBox2 的软件版本：3.40G 版以上       |

<CR750-D/CR751-D 控制器 >



图 5-16: 设备的连接（CR750-D/CR751-D 控制器）

<CR750-Q/CR751-Q 控制器 >

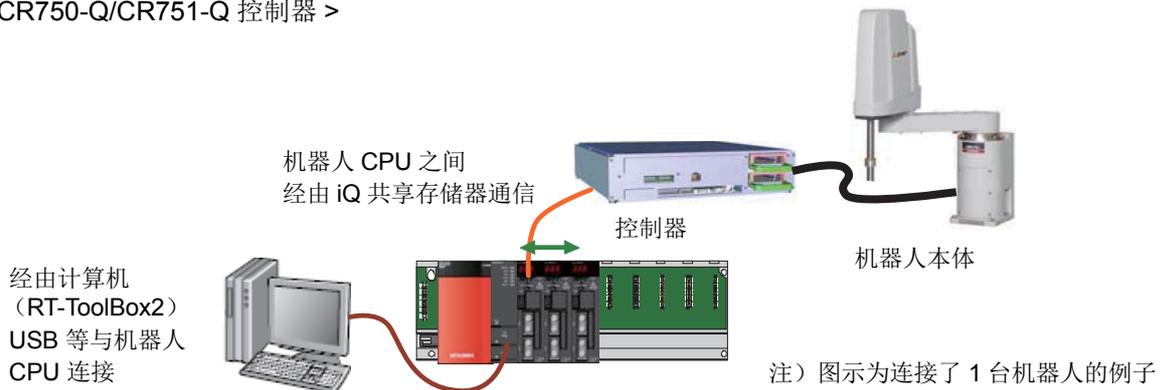
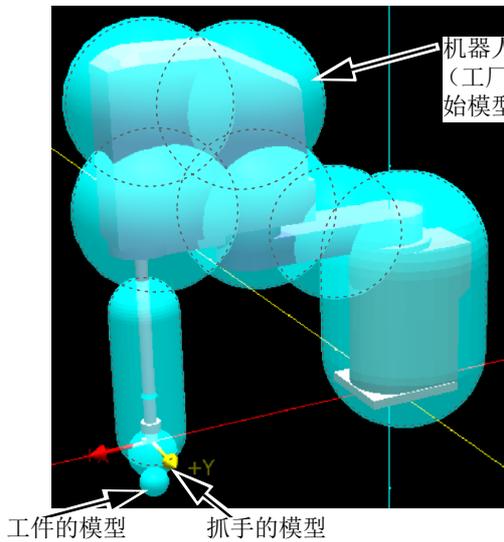


图 5-17: 设备的连接 (CR750-Q/CR751-Q 控制器)

5.24.3 干涉确认用模型的登录

以机器人本体为基准登录用于干涉确认的模型 (以下简称为“模型”)。

<RH-F 系列 >



<RV-F 系列 >

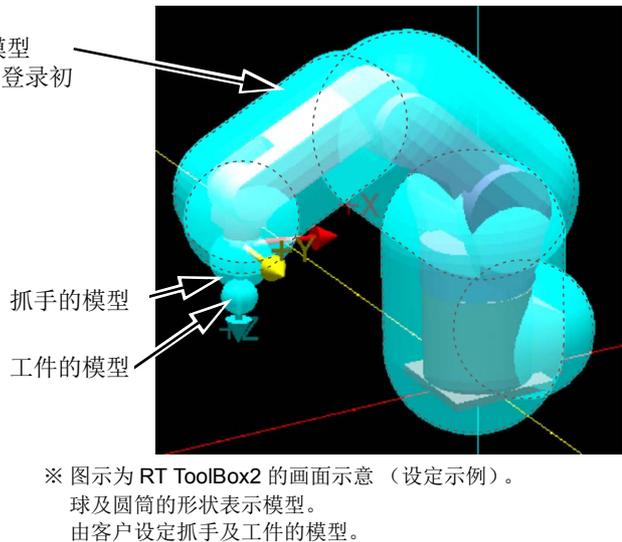


图 5-18: 模型的登录示例

必要的登录内容归纳为表 5-31。机器人本体、抓手、工件各可登录 8 个模型。

表 5-31: 模型的登录内容

| 模型的设定项目                                                                   | 模型的种类                                                 |                                                                                     |                                                                                     |
|---------------------------------------------------------------------------|-------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                                                                           | 机器人本体的模型注 1)                                          | 抓手的模型                                                                               | 工件的模型                                                                               |
| 关系<br>• 机器人本体:<br>设定将各模型登录到机器人本体的哪个部分<br>• 抓手、工件:<br>为各模型设定号码 (抓手号码、工件号码) | 设定作为机器人 Jn 轴登录的各模型与机器人本体之间的关系。(模型的位置设定为基座部、Jn 轴或是法兰部) | 设定的抓手号码与 LoadSet (Load Set) 指令的抓手条件号码联动。<br><br>注) 模型的形状为球形及圆筒形。<br>参数: CAVKDH1 ~ 8 | 设定的工件号码与 LoadSet (Load Set) 指令的工件条件号码联动。<br><br>注) 模型的形状为球形及圆筒形。<br>参数: CAVKDW1 ~ 8 |
|                                                                           | 注) 模型的形状为球形及圆筒形。<br>参数: CAVKDA1 ~ 8                   | LoadSet 指令可动态变更作为干涉确认对象的抓手及工件的模型。                                                   |                                                                                     |

| 模型的设定项目     | 模型的种类                                                           |                                                      |                |
|-------------|-----------------------------------------------------------------|------------------------------------------------------|----------------|
|             | 机器人本体的模型 <sup>注1)</sup>                                         | 抓手的模型                                                | 工件的模型          |
| 中心位置        | 将机器人本体安装面或各轴的旋转中心作为基准，以从此处起的距离指定模型的中心位置。<br>参数：CAVPSA1 ~ 8      | 以从机械接口坐标系的原点（J3轴的前端）起的距离指定模型的中心位置。<br>参数：CAVPSH1 ~ 8 | 参数：CAVPSW1 ~ 8 |
| 模型的大小       | 以半径设定各模型的大小。                                                    |                                                      |                |
|             | 参数：CAVSZA1 ~ 8                                                  | 参数：CAVSZH1 ~ 8                                       | 参数：CAVSZW1 ~ 8 |
| 各模型的有效 / 无效 | 设定各模型的有效 / 无效，及示教单元有效时是否临时将其设为无效。<br>注) 抓手的模型、工件的模型需在示教操作时设为无效。 |                                                      |                |
|             | 参数：CAVSCA1 ~ 8                                                  | 参数：CAVSCH1 ~ 8                                       | 参数：CAVSCW1 ~ 8 |

注 1) 工厂出货时，各机种都设定了相应的初始值。

下面为各参数的说明内容。

(1) 模型登录用参数

下面介绍前述表 5-31 中归纳的各参数的详细内容。

机器人本体、抓手、工件各可最多登录 8 个模型，它们以各参数名的最后 1 位的数字来作区别。

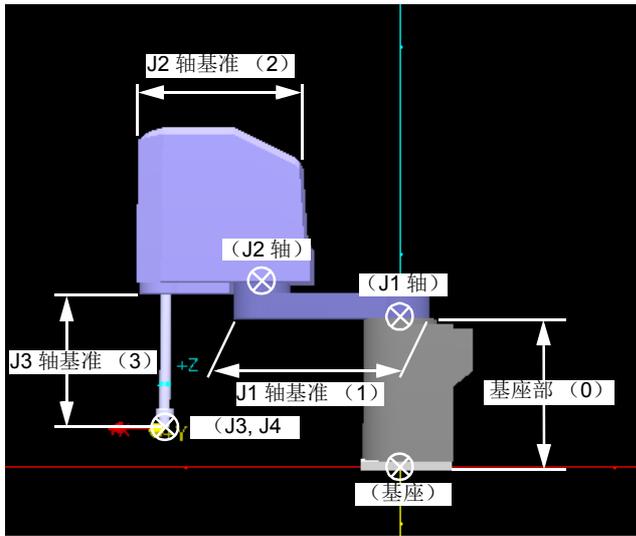
1) 机器人本体的模型

①模型登录部和形状：CAVKDA1 ~ 8

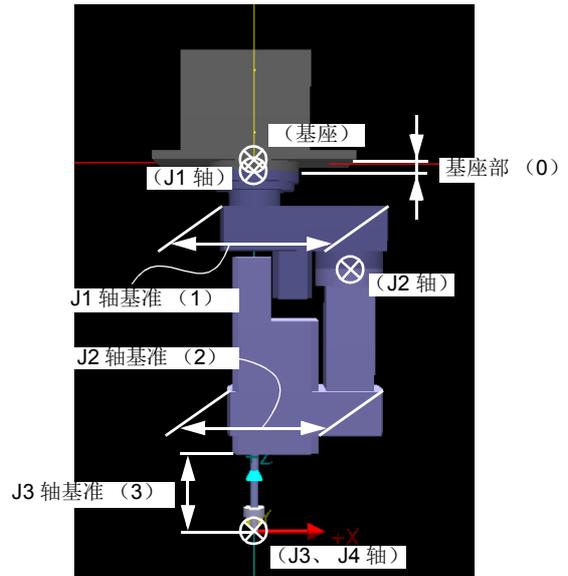
表 5-32: 模型设定参数（机器人本体：CAVKDA1 ~ 8）

| 参数                  | 参数名         | 排列数<br>文字数 | 内容说明                                                                                                                                 | 出货时设定值                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                        |
|---------------------|-------------|------------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 模型登录部和形状<br>(机器人本体) | CAVKDA1 ~ 8 | 整数 2       | 设定模型的登录部（基准轴）和形状。最多可登录 8 个模型。（各模型对应参数名的最后 1 位的号码（1~8））<br>第 1 要素：登录部（基准轴）<br>0 : 基座部<br>1~6 : Jn 轴<br>第 2 要素：形状<br>0 : 球形<br>1 : 圆筒形 | RH-3/6/12/20FH 系列：<br>CAVKDA1=0, 1<br>CAVKDA2=0, 0<br>CAVKDA3=1, 0<br>CAVKDA4=2, 0<br>CAVKDA5=2, 0<br>CAVKDA6=2, 0<br>CAVKDA7=2, 0<br>CAVKDA8=4, 1<br>注) RH-3FH35xx、<br>RH-6FH35xx、<br>RH-12FH55xx 为<br>CAVKDA3=0, 0。<br><br>RH-3FHR 系列：<br>CAVKDA1=0, 0<br>CAVKDA2=1, 1<br>CAVKDA3=1, 0<br>CAVKDA4=2, 1<br>CAVKDA5=2, 1<br>CAVKDA6=3, 1<br>CAVKDA7=0, 0<br>CAVKDA8=0, 0 | RV-F 系列：<br>CAVKDA1=0, 0<br>CAVKDA2=0, 1<br>CAVKDA3=2, 1<br>CAVKDA4=4, 1<br>CAVKDA5=5, 0<br>CAVKDA6=0, 0<br>CAVKDA7=0, 0<br>CAVKDA8=0, 0<br>注) RV-2F 为<br>CAVKDA2=0, 0<br>CAVKDA4=3, 0<br>CAVKDA5=4, 1<br>CAVKDA6=5, 0 |

<RH-3FH/6FH/12FH/20FH 系列 >



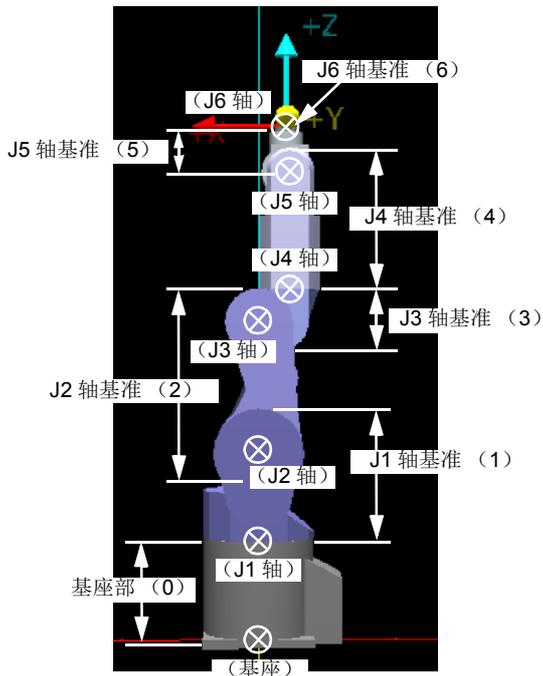
<RH-3FHR 系列 >



注) 括号内数字为参数: CAVKDA1 ~ 8 (模型登录部和形状) 的第 1 要素 (登录部 (基准轴)) 设定值

图 5-19: RH-F 系列模型登录部 (补充)

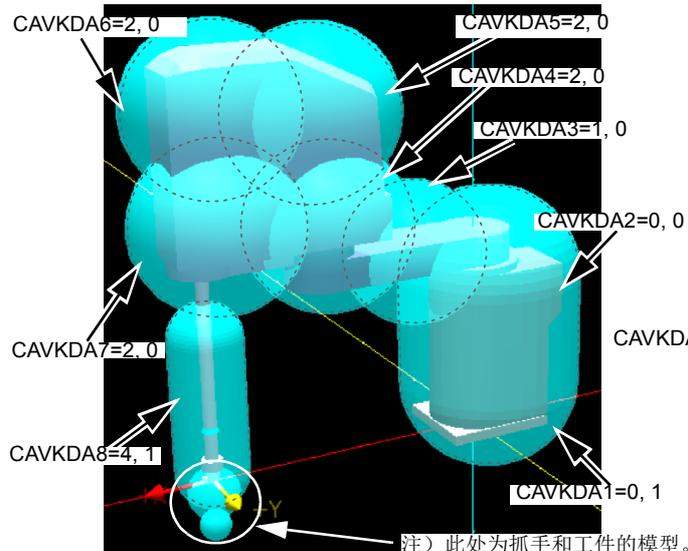
<RV-F 系列 >



注) 括号内数字为参数: CAVKDA1 ~ 8 (模型登录部和形状) 的第 1 要素 (登录部 (基准轴)) 设定值

图 5-20: RV-F 系列模型登录部 (补充)

<RH-F 系列>



<RV-F 系列>

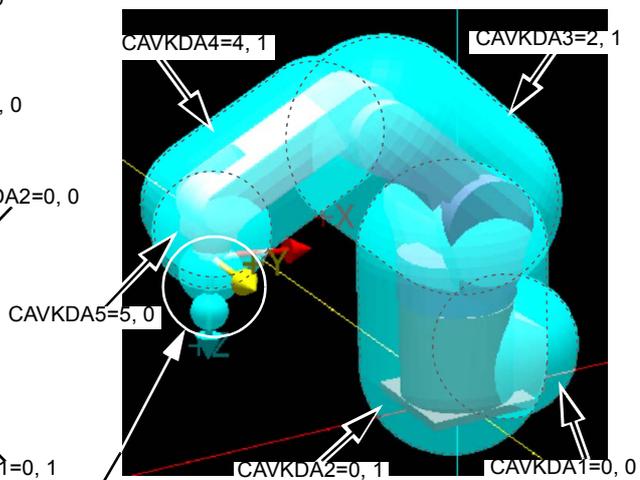
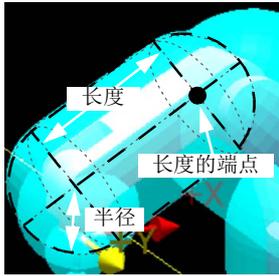


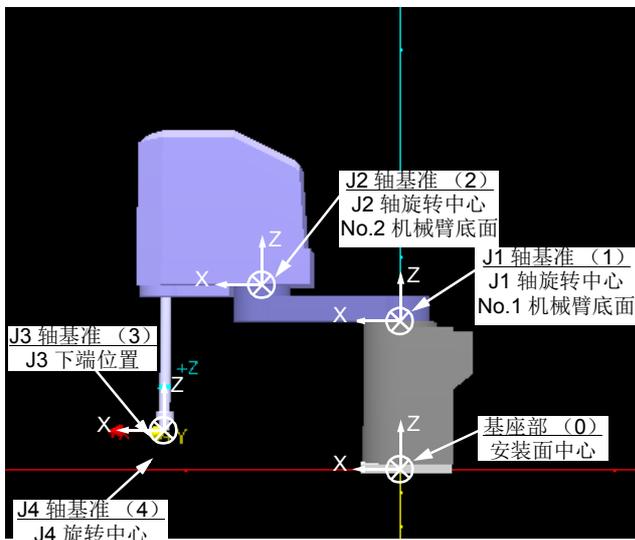
图 5-21: 登录部和模型形状的设置示例

②模型的位置：CAVPSA1 ~ 8

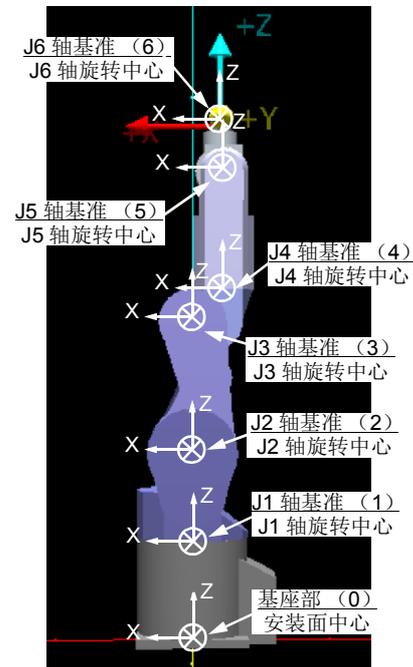
表 5-33: 模型设定参数 (机器人本体: CAVPSA1 ~ 8)

| 参数               | 参数名         | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 出货时设定值 |
|------------------|-------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 模型的位置<br>(机器人本体) | CAVPSA1 ~ 8 | 实数 6       | <p>以从各基准轴位置起的距离和旋转角设定要登录的各模型的位置。<br/>(各模型对应参数名的最后1位的号码(1~8))</p> <p>第1要素: X轴方向的距离 (mm)<br/>第2要素: Y轴方向的距离 (mm)<br/>第3要素: Z轴方向的距离 (mm)<br/>第4要素: 绕X轴旋转的旋转角 (deg)<br/>第5要素: 绕Y轴旋转的旋转角 (deg)<br/>第6要素: 绕Z轴旋转的旋转角 (deg)</p> <p>注) 旋转角按Z轴→Y轴→X轴的顺序反映。形状为球形时, 无需设定旋转角。</p> <p>&lt; 要设定的模型的场所 &gt;</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>球形</p> </div> <div style="text-align: center;">  <p>圆筒形</p> </div> </div> |        |

<RH-F 系列 >



<RV-F 系列 >



注) • 括号内的数字为参数: CAVKDA1 ~ 8 (模型登录部和形状) 的第 1 要素 (登录部 (基准轴)) 设定值  
• 上段: 基准轴  
下段: 基准轴的场所

图 5-22: 各模型的基准轴位置 XYZ 方向

[补充]: 各基准轴的 XYZ 方向

RH-F 系列

假设 J1, J2, J4 = 0 度、J3 = 下端的姿势时, 各基准轴的 XYZ 方向与基本坐标系一致。

RV-F 系列

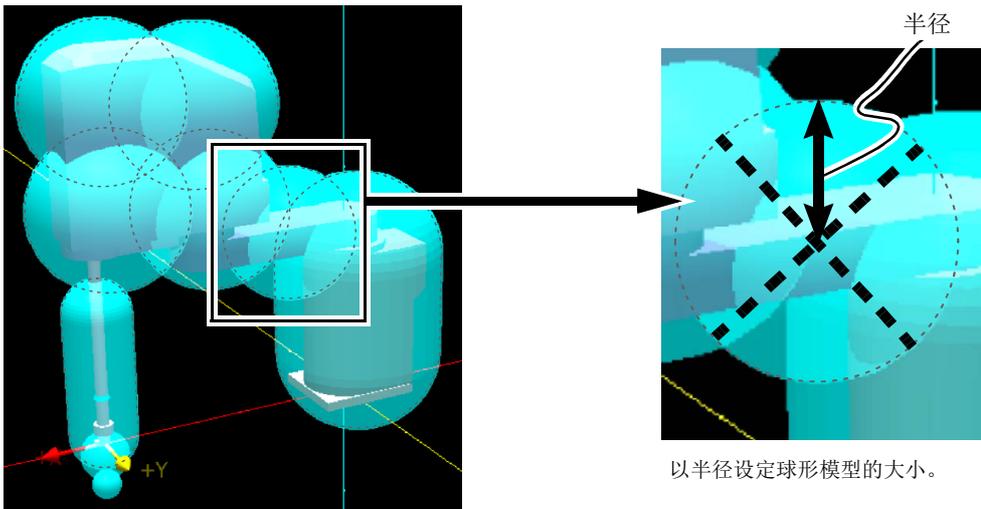
假设全轴为 0 度的姿势时, 各基准轴的 XYZ 方向与基本坐标系一致。

③模型的大小：CAVSZA1 ~ 8

表 5-34：模型设定参数（机器人本体：CAVSZA1 ~ 8）

| 参数               | 参数名         | 排列数<br>文字数 | 内容说明                                                                                                                | 出货时设定值 |
|------------------|-------------|------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 模型的大小<br>(机器人本体) | CAVSZA1 ~ 8 | 实数 4       | 设定各模型的大小。(各模型对应参数名的最后1位的号码(1~8))<br><br>第1要素：半径 (mm)<br>第2要素：长度 (mm)<br>第3要素：固定为0<br>第4要素：固定为0<br>注) 形状为球形时，无需设定长度。 |        |

<RH-F 系列>



<RV-F 系列>

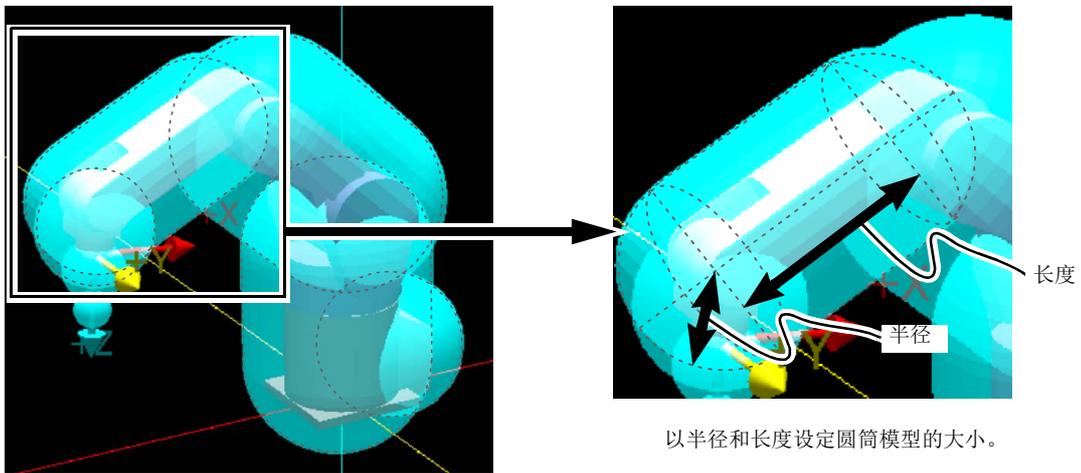


图 5-23：模型的大小（补充）

## ④模型的有效/无效：CAVSCA1～8

表 5-35：模型设定参数（机器人本体：CAVSCA1～8）

| 参数                  | 参数名       | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                  | 出货时设定值                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 模型的有效/无效<br>(机器人本体) | CAVSCA1～8 | 整数 3       | <p>对各模型设定是否作为干涉确认的对象（有效/无效）。（各模型对应参数名的最后1位的号码（1～8））</p> <p>第1要素：有效/无效的设定<br/>（0：无效、1：有效）</p> <p>第2要素：设定在JOG操作时的临时解除干涉回避功能的操作中是否设为无效<br/>（0：使无效、1：保持有效状态）</p> <p>注）在示教时的JOG操作中，对不得不需要干涉的抓手或工件的模型设定为“0：设为无效”可方便操作。</p> <p>第3要素：固定为0</p> | <p>RH-3/6/12/02FH系列：<br/>CAVSCA1=1, 0, 0<br/>CAVSCA2=0, 0, 0<br/>CAVSCA3=1, 0, 0<br/>CAVSCA4=1, 0, 0<br/>CAVSCA5=1, 0, 0<br/>CAVSCA6=1, 0, 0<br/>CAVSCA7=1, 0, 0<br/>CAVSCA8=1, 0, 0</p> <p>RV-F系列：<br/>CAVSCA1=1, 0, 0<br/>CAVSCA2=1, 0, 0<br/>CAVSCA3=1, 0, 0<br/>CAVSCA4=1, 0, 0<br/>CAVSCA5=1, 0, 0<br/>CAVSCA6=0, 0, 0<br/>CAVSCA7=0, 0, 0<br/>CAVSCA8=0, 0, 0</p> <p>RH-3FHR系列：<br/>CAVSCA1=1, 0, 0<br/>CAVSCA2=1, 0, 0<br/>CAVSCA3=1, 0, 0<br/>CAVSCA4=1, 0, 0<br/>CAVSCA5=1, 0, 0<br/>CAVSCA6=1, 0, 0<br/>CAVSCA7=0, 0, 0<br/>CAVSCA8=0, 0, 0</p> |

## 2) 抓手的模型

表 5-36 中为作为抓手登录的模型的设定参数。

表 5-36：模型设定参数（抓手）

| 参数              | 参数名       | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                           | 出货时设定值                                  |
|-----------------|-----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 抓手号码和形状<br>(抓手) | CAVKDH1～8 | 整数 2       | <p>设定作为抓手登录的各模型的抓手号码和形状。最多可登录8个模型。（各模型对应参数名的最后1位的号码（1～8））</p> <p>第1要素：抓手号码<br/>对应使用LoadSet (Load Set)指令变更模型时的抓手条件号码<br/>0：作为初始值设定的模型<br/>1～8：通过LoadSet (Load Set)指令指定的抓手条件号码</p> <p>第2要素：形状<br/>0：球形<br/>1：圆筒形</p>                             | 将所有参数（CAVKDH1～8）都设定为“0, 0”。             |
| 模型的中心位置<br>(抓手) | CAVPSH1～8 | 实数 6       | <p>对各模型指定从机械接口坐标系观察到的模型的位置和姿势。（各模型对应参数名的最后1位的号码（1～8））</p> <p>第1要素：X轴方向的距离（mm）<br/>第2要素：Y轴方向的距离（mm）<br/>第3要素：Z轴方向的距离（mm）<br/>第4要素：绕X轴旋转的旋转角（deg）<br/>第5要素：绕Y轴旋转的旋转角（deg）<br/>第6要素：绕Z轴旋转的旋转角（deg）</p> <p>注）旋转角按Z轴→Y轴→X轴的顺序反映。形状为球形时，无需设定旋转角。</p> | 将所有参数（CAVPSH1～8）都设定为“0, 0, 0, 0, 0, 0”。 |
| 模型的大小<br>(抓手)   | CAVSZH1～8 | 实数 4       | <p>设定各模型的大小。（各模型对应参数名的最后1位的号码（1～8））</p> <p>第1要素：半径（mm）<br/>第2要素：长度（mm）<br/>第3要素：固定为0<br/>第4要素：固定为0</p> <p>注）形状为球形时，无需设定长度。</p>                                                                                                                 | 将所有参数（CAVSZH1～8）都设定为“0, 0, 0, 0”。       |

| 参数                 | 参数名         | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                     | 出货时设定值                         |
|--------------------|-------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| 模型的有效 / 无效<br>(抓手) | CAVSCH1 ~ 8 | 整数 3       | 对各模型设定是否作为干涉确认的对象(有效/无效)。(各模型对应参数名的最后1位的号码(1~8))<br><br>第1要素: 有效/无效的设定(0: 无效、1: 有效)<br>第2要素: 设定在JOG操作时的临时解除干涉回避功能的操作中是否设为无效(0: 设为无效、1: 保持有效状态)。<br>注) 在示教时的JOG操作中, 对不得不需要干涉的抓手或工件的模型设定为“0: 设为无效”可方便操作。<br><br>第3要素: 固定为0 | 将所有参数(CAVSCH1~8)都设定为“0, 0, 0”。 |

◇◆◇变更执行程序时的确认对象模型(抓手、工件)◆◆◇  
 执行程序时, 可以使用 LoadSet 指令, 变更作为干涉确认对象的抓手和工件的模型, 以便根据实际使用的抓手或把持的工件执行干涉确认。使用 LoadSet 指令时, 指定事先在参数中设定的抓手号码、工件号码。

3) 工件的模型

表 5-37 中为作为工件登录的模型的设定参数。

表 5-37: 模型设定参数(工件)

| 参数                 | 参数名         | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                   | 出货时设定值                                  |
|--------------------|-------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 工件号码和形状<br>(工件)    | CAVKDW1 ~ 8 | 整数 2       | 设定作为工件登录的各模型的工件号码和形状。最多可登录8个模型。(各模型对应参数名的最后1位的号码(1~8))<br>第1要素: 工件号码<br>对应使用 LoadSet (Load Set) 指令变更模型时的工件条件号码<br>0 : 作为初始值设定的模型<br>1~8 : 通过 LoadSet (Load Set) 指令指定的工件条件号码<br>第2要素: 形状<br>0 : 球形<br>1 : 圆筒形                           | 将所有参数(CAVKDW1~8)都设定为“0, 0”。             |
| 模型的位置<br>(工件)      | CAVPSW1 ~ 8 | 实数 6       | 对各模型指定从机械接口坐标系观察到的模型的位置和姿势。(各模型对应参数名的最后1位的号码(1~8))<br><br>第1要素: X轴方向的距离(mm)<br>第2要素: Y轴方向的距离(mm)<br>第3要素: Z轴方向的距离(mm)<br>第4要素: 绕X轴旋转的旋转角(deg)<br>第5要素: 绕Y轴旋转的旋转角(deg)<br>第6要素: 绕Z轴旋转的旋转角(deg)<br>注) 旋转角按Z轴→Y轴→X轴的顺序反映。形状为球形时, 无需设定旋转角。 | 将所有参数(CAVPSW1~8)都设定为“0, 0, 0, 0, 0, 0”。 |
| 模型的大小<br>(工件)      | CAVSZW1 ~ 8 | 实数 4       | 设定各模型的大小。(各模型对应参数名的最后1位的号码(1~8))<br><br>第1要素: 半径(mm)<br>第2要素: 长度(mm)<br>第3要素: 固定为0<br>第4要素: 固定为0<br>注) 形状为球形时, 无需设定长度。                                                                                                                 | 将所有参数(CAVSZW1~8)都设定为“0, 0, 0, 0”。       |
| 模型的有效 / 无效<br>(工件) | CAVSCW1 ~ 8 | 整数 3       | 对各模型设定是否作为干涉确认的对象(有效/无效)。(各模型对应参数名的最后1位的号码(1~8))<br><br>第1要素: 有效/无效的设定(0: 无效、1: 有效)<br>第2要素: 设定在JOG操作时的临时解除干涉回避功能的操作中是否设为无效(0: 设为无效、1: 保持有效状态)。<br>注) 在示教时的JOG操作中, 对不得不需要干涉的抓手或工件的模型设定为“0: 设为无效”可方便操作。<br><br>第3要素: 固定为0               | 将所有参数(CAVSCW1~8)都设定为“0, 0, 0”。          |

## ◇◆◇变更执行程序时的确认对象模型（抓手、工件）◇◆◇

执行程序时，可以使用 **LoadSet (Load Set)** 指令，变更作为干涉确认对象的抓手和工件的模型，以便根据实际使用的抓手或把持的工件执行干涉确认。

使用 **LoadSet (Load Set)** 指令时，指定事先在定义模型的参数中设定的抓手号码、工件号码。

## ◇◆◇仅在进行把持时执行工件的干涉确认◇◆◇

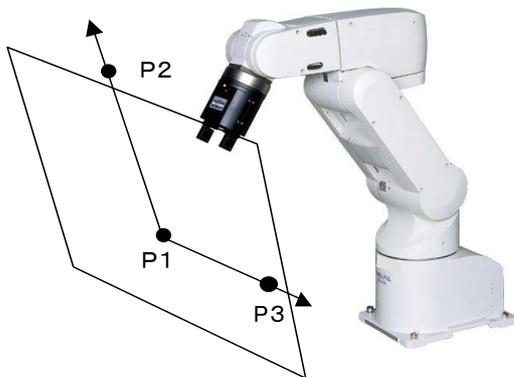
仅在参数 **HNDHOLD\*** 中设定的把持状态时，将工件模型作为干涉确认的对象。（与抓手开闭联动）

## 5.24.4 自由平面极限的登录

登录用于干涉确认的自由平面极限。必要的登录内容汇总在表 5-38 中。最多可以登录 8 个可进行干涉确认的自由平面极限。

**注意**

仅登录的机器人 CPU 进行干涉确认。



如左图所示，通过 3 点（P1，P2，P3）定义任意平面。

注) 图示为垂直 6 轴型机器人的例子。水平多关节型机器人也一样。

表 5-38: 自由平面极限登录内容

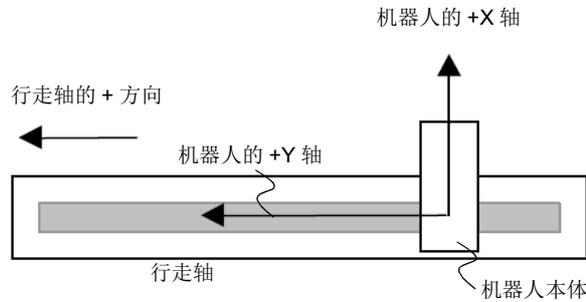
| 参数和值             | 说明                                                                                                                         |
|------------------|----------------------------------------------------------------------------------------------------------------------------|
| SFCnP(n=1 ~ 8)   | 指定用于创建平面的 3 点。<br>P1 的坐标值的 X1,Y1,Z1: 平面的原点位置<br>P2 的坐标值的 X2,Y2,Z2: 平面的 X 轴上的位置<br>P3 的坐标值的 X3,Y3,Z3: 平面的 X-Y 平面上的 +Y 方向的位置 |
| CAVSCFn(n=1 ~ 8) | 对设定的自由平面极限的干涉确认有效 / 无效进行指定。<br>0: 无效 (初始值)<br>1: 有效                                                                        |

5.24.5 附加轴的对应

在使用附加轴的机器人中，通过设定附加轴的同步控制参数，即可使用考虑到附加轴动作的干涉回避功能。（仅限行走轴等直动轴）

请根据使用状态，参照表 5-34 设定参数。

注) 对于后面的“[5.24.7 机器人之间的校准 (仅限机器人之间的干涉确认时)]”中设定的机器人之间的位置关系，请将行走轴的坐标值设定为“0”。



此时，参数为 AXDIR = 0.0, 0.0, -90.0。  
(机器人绕 Z 轴旋转 -90 度)

图 5-24: 行走轴的使用示例

表 5-39: 附加轴的同步控制参数

| 参数                 | 参数名       | 排列数<br>文字数 | 内容说明                                                                                                                                | 出货时设定值        |
|--------------------|-----------|------------|-------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 指定干涉回避附加轴 (行走轴) 号码 | CAVAX.JNO | 整数 1       | 设定干涉回避中需要考虑到附加轴 (行走轴) 的轴号码。<br>7、8轴以外的设定为非干涉确认对象。<br>设定值: 0、7或8                                                                     | 0             |
| 附件轴同步方向            | AXDIR     | 实数 3       | 将行走轴的 + 方向从作为 X 轴的坐标系向机器人坐标系变换<br>第1要素: 绕 X 轴旋转的旋转角<br>第2要素: 绕 Y 轴旋转的旋转角<br>第3要素: 绕 Z 轴旋转的旋转角<br>注) 在工厂出货设定中, 机器人的 X 轴与行走轴的 + 方向一致。 | 0.0, 0.0, 0.0 |

## 5.24.6 共享存储器扩展功能的设定（仅限机器人之间的干涉确认时）

使用参数设定共有内存扩展功能。

通过该设定，占用机器人 CPU 间通信所需的共有内存。（参照图 5-25）

## (1) 参数的设定

将参数：IQMEM 的位 4 设为“1”，将机器人间协调控制功能设定为有效。

或在参数：QMLTCPUN 中将多 CPU 系统中安装在主基板上的 CPU 单元的台数及参数：QMLTCPUn 的要素 1 设定为“2”。

表 5-40：共有内存扩展功能选择参数

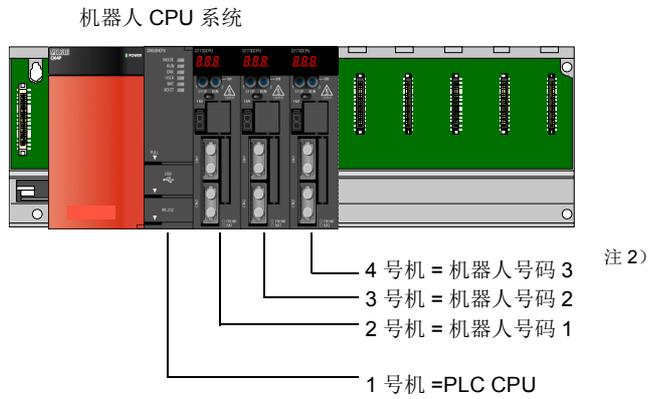
| 参数                            | 参数名                   | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 出货时设定值                 |                  |   |                      |   |                       |                      |           |         |
|-------------------------------|-----------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------------|---|----------------------|---|-----------------------|----------------------|-----------|---------|
| 选择共有内存扩展功能                    | IQMEM                 | 整数 1       | <p>选择共有内存扩展功能。<br/>对每个位分配功能。1/0= 有效 / 无效</p> <p>15<br/>0<br/>00000000 00000000</p> <table border="0"> <tr> <td style="border: 1px solid black; padding: 2px;">bit2-3、<br/>5-15<br/>未使用</td> <td style="padding: 2px;">   ...bit0: 扩展功能的使用</td> </tr> <tr> <td></td> <td style="padding: 2px;">   .....bit1:PLC 直接执行功能</td> </tr> <tr> <td></td> <td style="padding: 2px;">   .....bit4: 机器人间协调控制功能</td> </tr> </table>                                                                                                                                                                                                                                                                              | bit2-3、<br>5-15<br>未使用 | ...bit0: 扩展功能的使用 |   | .....bit1:PLC 直接执行功能 |   | .....bit4: 机器人间协调控制功能 | 00000000<br>00000000 |           |         |
| bit2-3、<br>5-15<br>未使用        | ...bit0: 扩展功能的使用      |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |
|                               | .....bit1:PLC 直接执行功能  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |
|                               | .....bit4: 机器人间协调控制功能 |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |
| 设定多 CPU 台数                    | QMLTCPUN              | 整数 1       | <p>设定在多 CPU 系统中，安装在主基板上的 CPU 单元的台数。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 2                      |                  |   |                      |   |                       |                      |           |         |
| 设定多 CPU <sub>n</sub> 号机高速通信领域 | QMLTCPUn<br>n=1 ~ 4   | 整数 4       | <p>设定在多 CPU 系统中，在 1~4 号机的多 CPU 间高速通信功能的各 CPU 单元间进行收发信的点数。<br/>所有 CPU 的参数设定值必须统一。如果参数设定值不正确，则 PLC CPU 将出现报警，请统一各 CPU 的参数设定值。</p> <p>要素1：使用者自由领域的大小（K点）<br/>范围：1~14（最大※）<br/>※如下表所示，根据多 CPU 的台数，最大值也有所不同。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CPU 台数</th> <th>设定范围</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>0 ~ 14K 点</td> </tr> <tr> <td>3</td> <td>0 ~ 13K 点</td> </tr> <tr> <td>4</td> <td>0 ~ 12K 点</td> </tr> </tbody> </table> <p>要素2：自动刷新的点数（点）<br/>范围：0~14335<br/>机器人 CPU 不支持自动刷新，因此，请将自动刷新的点数 <b>始终设为 0。</b></p> <p>要素3：系统领域的大小（K点）<br/>范围：1或2</p> <p>要素4：多 CPU 同步启动（1：同步启动、0：不同步启动）<br/>机器人 CPU 在启动上需要花费一定的时间，因此，原则上请保持 1（同步启动）的状态，<b>不要变更设定。</b></p> | CPU 台数                 | 设定范围             | 2 | 0 ~ 14K 点            | 3 | 0 ~ 13K 点             | 4                    | 0 ~ 12K 点 | 1,0,1,1 |
| CPU 台数                        | 设定范围                  |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |
| 2                             | 0 ~ 14K 点             |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |
| 3                             | 0 ~ 13K 点             |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |
| 4                             | 0 ~ 12K 点             |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                  |   |                      |   |                       |                      |           |         |

(2) 共有内存映射  
表示基于参数: IQMEM 的设定状态的共有内存上的机器人 CPU 输出领域的内存映射。

< 共有内存映射 >



< 插槽和机器人号码的关系 >



注 1) 机器人间协调控制功能的地址为固定不变, 与扩展功能的有效 / 无效状态无关。

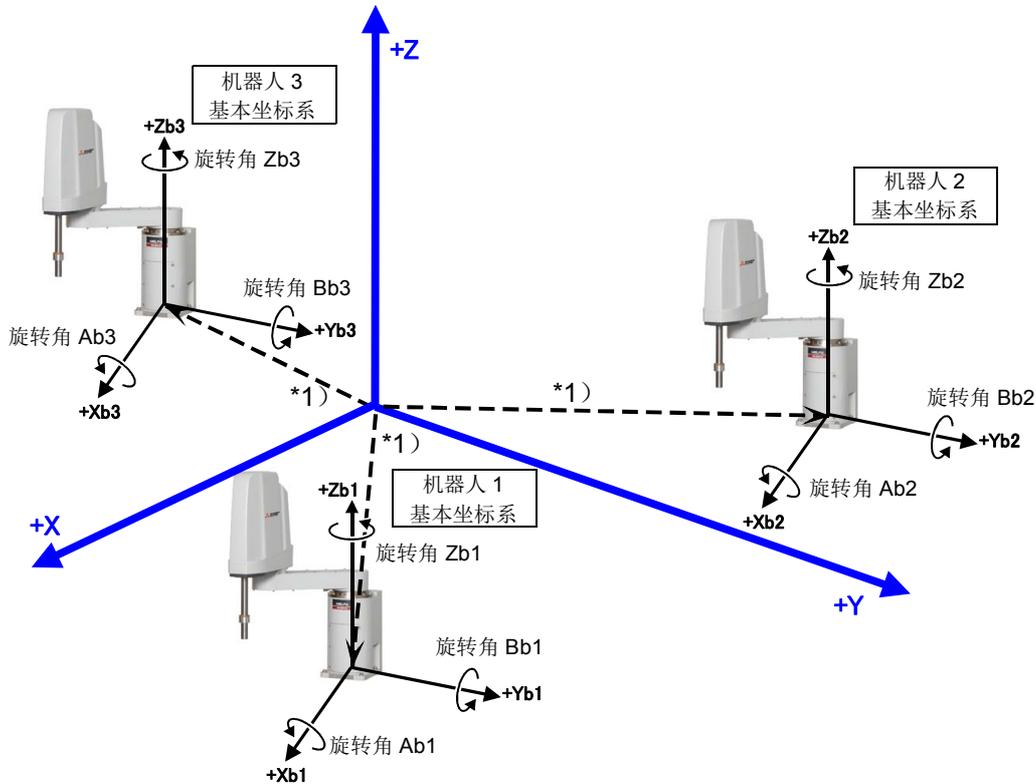
注 2) 共有内存地址中显示的 U3En¥ 的 “n” 对应机器人编号。  
机器人 CPU 系统 (iQPlatform) 的各插槽位置的机器人号码是固定的。

图 5-25: 共有内存映射

## 5.24.7 机器人之间的校准（仅限机器人之间的干涉确认时）

设定干涉回避功能有效的机器人间的位置关系。

根据装置的布局图等决定一个机器人间通用的坐标系原点。将从该坐标系观察到的各机器人的基本坐标系原点位置设定为参数：RBCORD。



\*1) 从通用的坐标系观察到的各机器人的基本坐标系原点。  
将该坐标值设定为各机器人的参数：RBCORD。

图 5-26：机器人间的校准示意图

## (1) 校准的设定

将从机器人间通用坐标观察到的机器人的基本坐标系原点位置以 X、Y、Z、A、B、C 的坐标值设定为参数：RBCORD。

注) 使用行走轴时，请设定行走轴的坐标值设为“0”时的位置关系。

表 5-41：机器人间的校准设定参数

| 参数       | 参数名    | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                         | 出货时设定值                                 |
|----------|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| 机器人间通用坐标 | RBCORD | 实数 6       | <p>从机器人间通用坐标观察到的本机器人的基本坐标系原点位置。<br/>(以 X、Y、Z、A、B、C 的坐标值指定)</p> <p>第1要素：X轴坐标值 (mm)<br/>第2要素：Y轴坐标值 (mm)<br/>第3要素：Z轴坐标值 (mm)<br/>第4要素：A轴坐标值 (deg) (绕X轴旋转的旋转角)<br/>第5要素：B轴坐标值 (deg) (绕Y轴旋转的旋转角)<br/>第6要素：C轴坐标值 (deg) (绕Z轴旋转的旋转角)</p> <p>注1) 请将A、B、C轴坐标值(旋转角)设定为按绕Z轴旋转→绕Y轴旋转→绕X轴旋转的顺序旋转后得到的值。</p> | 0.00, 0.00, 0.00,<br>0.00, 0.00, 0.00, |

(2) 校准设定结果的确认

确认各机器人已正确实施校准设定。

确认步骤如下所示。

- 1) 通过装置的布局图等为每个机器人确定一个基准位置。(图 5-27 的 a)。以下为基准点)
- 2) 通过 JOG 操作对准已确定各机器人的抓手前端的基准点。
- 3) 通过示教单元等确认机器人 (系统) 状态变量: **P\_CurrR** (从通用坐标系观察到的机器人的现在位置) 的值。
- 4) 对上述 **P\_CurrR** 的值与从布局图等上读取到的值进行比较。  
比较对象仅限 XYZ 的值。双方一致, 即表明校准正确。如果不同, 则请调整参数: **RBCORD** 的设定值。
- 5) 对所有干涉回避功能有效的机器人实施上述操作。

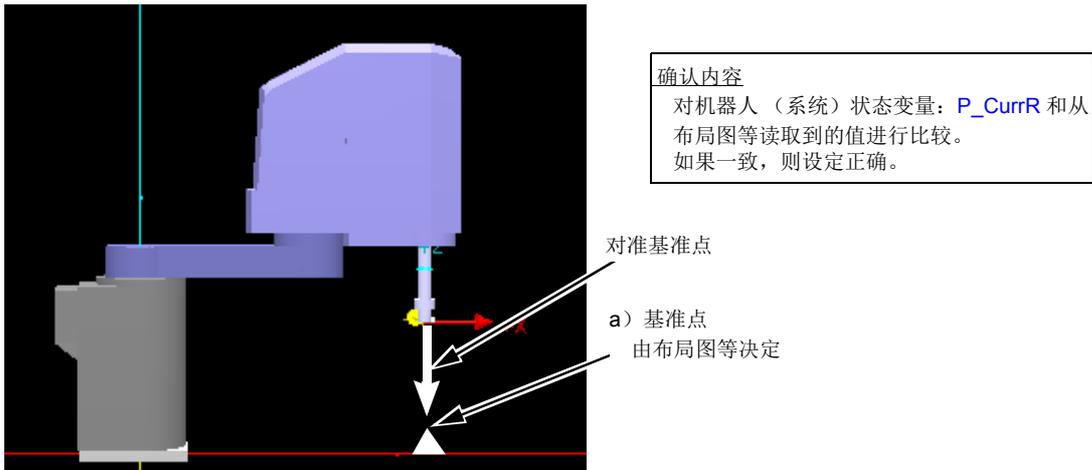


图 5-27: 确认机器人间的校准设定

## 5.24.8 干涉回避功能的有效/无效设定

干涉回避功能本身的可否使用和执行程序时及 JOG 操作时各自的有效 / 无效，可通过参数：CAV 进行设定。

要设定的参数的详细内容如表 5-42 所示。

表 5-42：干涉回避功能的有效 / 无效设定参数

| 参数       | 参数名 | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                            | 出货时设定值  |
|----------|-----|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 干涉回避功能设定 | CAV | 整数 3       | <p>设定干涉回避功能本身的使用可否。<br/>在设定为可以使用时，还要对执行程序时的有效/无效、JOG操作时的有效（包含干涉时的动作）/无效进行设定。</p> <p>第1要素：使用干涉回避功能的设定（0：不可使用/1：可以使用）<br/>第2要素：指定程序运行时的初始状态（0：无效、1：有效）<br/>第3要素：设定JOG操作时的有效/无效<br/>（0：无效、1：有效•发生报警、2：有效•仅蜂鸣音）</p> | 0, 1, 1 |

5.24.9 干涉回避功能的使用

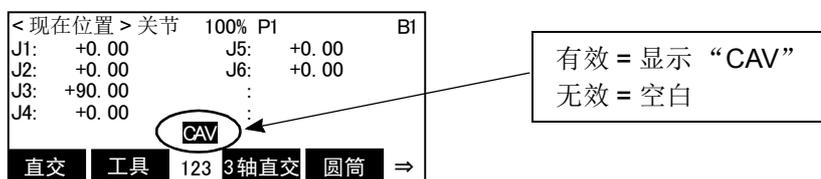
下面介绍 JOG 操作时及执行程序时的干涉回避动作。将参数 CAV 的第 1 要素设定为“1”（可以使用）。（请参照第 498 页的「5.24.8 干涉回避功能的有效 / 无效设定」）

(1) JOG 操作时的干涉回避

将参数：CAV 的第 3 要素设定为有效（“1”或“2”）时，可在 JOG 操作时使用干涉回避功能。一旦检测到干涉，机器人将减速停止，根据该参数的设定值，动作有如下之分。  
 设定值 =1：发生报警 (L241n)。重新开始 JOG 操作需要进行一次报警复位操作。  
 设定值 =2：蜂鸣音响起。可向不干涉的方向动作。

临时解除干涉回避功能的操作

模型的有效 / 无效设定参数：CAVSCA1 ~ 8（机器人本体）、CAVSCH1 ~ 8（抓手）、CAVSCW1 ~ 8（工件）的各第 2 要素设定为“0”（暂时设为无效）的模型，通过按示教单元的 [CLEAR] 键，可暂时将干涉确认设为无效。  
 每按 1 次 [CLEAR] 键，即在有效 / 无效间切换 1 次，有效状态下，示教单元画面中央下方显示“CAV”。



此外，通过 [CLEAR] 键的操作仍然不能从干涉回避中恢复时（伺服 OFF 中手动使机器人进入干涉领域等时），通过执行本手册第 60 页的「3.10 无法解除报警的暂时报警复位操作」（按住 [RESET] 键的同时进行 JOG 操作），可暂时解除干涉回避功能本身，使机器人动作。

注）但是，在该操作下，机器人不会在动作范围限制处停止。请注意不要向动作范围外动作。

(2) 执行程序时的干涉回避

将参数：CAV 的第 2 要素设定为有效（“1”），通过使用表 5-43 中所示的指令语句及表 5-44 中所示的外部变量，可在执行程序时（自动运行中）使用干涉回避功能。

表 5-43: 指令语句（干涉回避功能）

| 指令语句   | 内容                                                                           | 说明页   |
|--------|------------------------------------------------------------------------------|-------|
| CavChk | 进行干涉回避功能的有效/无效设定和干涉检知时的报警发生有无的设定。<br>CavChk <ON/OFF>[, <机器人CPU号码> [,NOErr] ] | 第173页 |

表 5-44: 机器人（系统）状态变量（干涉回避功能）

| 变量名      | 内容                                                      | 说明页   |
|----------|---------------------------------------------------------|-------|
| P_CordR  | 从通用坐标观察到的本机器人的基本坐标系原点位置<br>数据类型：位置型                     | 第356页 |
| P_CurrR  | 从通用坐标观察到的本机器人的现在位置<br>数据类型：位置型                          | 第358页 |
| P_CavDir | 检测到干涉时的机器人的动作方向<br>数据类型：位置型                             | 第354页 |
| M_CavSts | 干涉状态。<br>0: 无干涉<br>1 ~ 3: 检测到干涉时的干涉对象的机器人号码<br>数据类型：整数型 | 第301页 |

## 【补充】

一旦检测到干涉，机器人将减速停止，根据 **CavChk On** 指令有无指定“NOErr”，动作有如下之分。

无 NOErr 的指定：发生报警 (L240n)。重新开始自动操作需要进行一次报警复位操作。

有 NOErr 的指定：不发生报警，执行程序中指定的中断处理。

注) 指定 NOErr 时，必须要有检测到干涉时要处理的中断程序。

如果执行本指令前未定义中断处理，在执行本指令时将会发生报警。

中断处理的概要：当机器人（系统）状态变量 **M\_CavSts** 的值不为“0”时，即为检测干涉的状态。声明在该状态下进行中断处理。

将在第 500 页的「5.24.10 采样程序」中介绍采样程序。



## 注意

未完成原点设定的机器人会将现在的关节坐标值始终作为全轴 0 度进行干涉确认。

因此，本功能将无法正常工作。请务必在完成原点设定之后执行本功能。

## 5.24.10 采样程序

## (1) 开始和结束干涉回避功能（所有机器人）

< 程序例 > 注) 步号从略。

```

:
'--- 初始状态（干涉回避功能有效） ---；通过参数 CAV 设定为干涉确认有效时。
:
MVS P1 '干涉回避功能有效时的动作
:
'--- 所有机器人的干涉回避功能无效 ---
CavChk OFF '通过省略机器人号码，将所有机器人设为无效
:
MVS P2 '干涉回避功能无效时的动作
:
'--- 所有机器人的干涉回避功能有效 ---
CavChk ON '通过省略机器人号码，将所有机器人设为有效
:
MVS P3 '干涉回避功能有效时的动作
:

```

## (2) 开始和结束干涉回避功能（指定机器人）

< 程序例 > 注) 步号从略。

```

:
'--- 结束与 2 号机之间的干涉回避功能 ---
CavChk OFF,2 '结束与机器人号码 2 之间的干涉回避功能
:
MVS P5 '与机器人号码 2 之间的干涉回避功能无效时的动作
:
'--- 开始与 2 号机之间的干涉回避功能 ---
CavChk ON,2 '重新开始与机器人号码 2 之间的干涉回避功能
:
MVS P6 '与机器人号码 2 之间的干涉回避功能有效时的动作
:

```

## (3) 变更抓手、工件的模型

< 程序例 > 注) 步号从略。

```

:
'--- 变更抓手的模型 ---
LoadSet 2,2 ' 指定抓手、工件的模型 2
:
MVS P7 ' 以抓手、工件模型 2 执行干涉回避功能
:
'--- 变更干涉确认的模型 (工件) ---
LoadSet 1,2 ' 指定抓手模型 1、工件模型 2
:
MVS P8 ' 以抓手模型 1、工件模型 2 执行干涉回避功能
:
'--- 变更干涉确认的模型 (工件) ---
LoadSet 0,0 ' 将抓手、工件模型返回初始值
:
MVS P9 '
:

```

## (4) 检测到干涉后，执行待避动作 (中断处理)

< 程序例 > 注) 步号从略。

```

Def Act 1,M_CavSts<>0 GoTo *Home,S ' 通过中断定义检测到干涉时要执行的处理
Act 1=1
CavChk On,0,NOErr ' 在不发生报警的模式下将干涉回避功能设为有效。
Mov P1 ' 干涉回避功能有效时的动作
Mov P2 ' 干涉回避功能有效时的动作
Mov P3 ' 干涉回避功能有效时的动作
:
:
*Home ' 检测到干涉后中断的步
CavChk Off
M_CavSts=0 ' 清除干涉状态
MDist=Sqr (P_CavDir.X*P_CavDir.X+P_CavDir.Y*P_CavDir.Y+P_CavDir.Z*P_CavDir.Z)
 ' 求出移动量的比例
PESC=P_CavDir (1) * (-50) * (1/MDist)
 ' 生成待避动作的移动量 (从干涉位置退回 50mm)
PDST=P_Fbc (1) +PESC' 生成待避位置
Mvs PDST ' 待避动作
Mvs PHome ' 向待避位置移动
:

```

## 5.25 PLC输入输出单元直接控制

在 F-Q 系列的控制器中，可以参照由插入在基座单元中的其他号机管理的输入输出单元 / 输入输出混合单元的输入输出信号，也可指定机器人 CPU 管理的输出单元 / 输入输出混合单元，进行直接信号输出 ※1)。

下面介绍大致的规格和步骤。

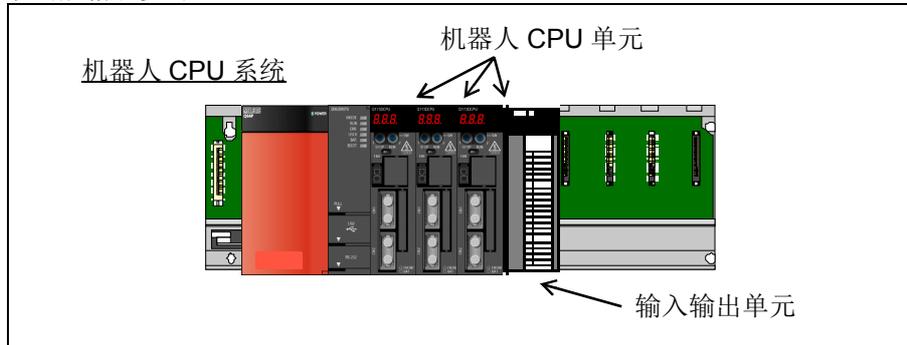


图 5-28: PLC 输入输出单元直接控制

### (1) 规格

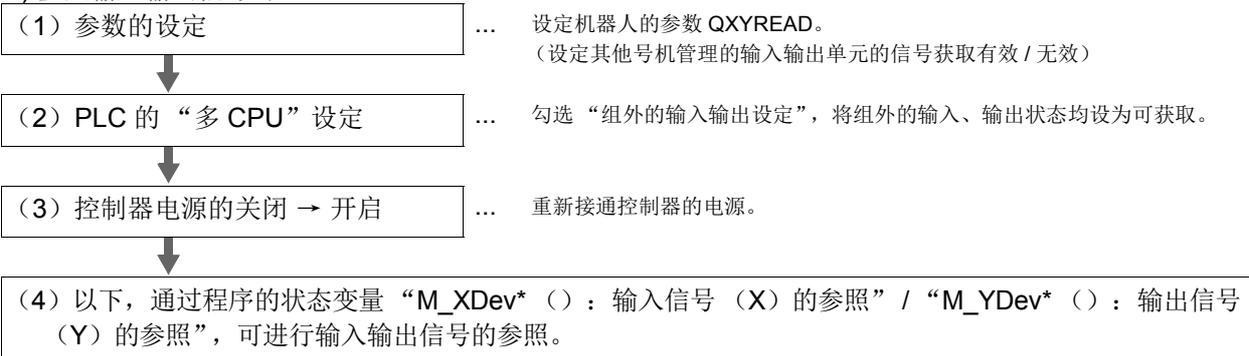
本功能通过参数 QXYREAD 进行由其他号机管理的输入输出单元 / 输入输出混合单元的输入输出信号参照的有效 / 无效的切换，以及通过参数 QXYUNIT 指定作为机器人 CPU 管理对象的输入输出单元加以使用。规格内容见表 5-45。

表 5-45: 规格

| No. | 项目     | 规格                                                                                  |
|-----|--------|-------------------------------------------------------------------------------------|
| 1   | 对象软件版本 | 在高于以下软件版本时，本功能有效。<br>本体: R2b<br>注) 对 R32TB/R33TB、R56TB/R57TB、RT ToolBox2 的软件版本没有限制。 |
| 2   | 对象插槽   | 支持主基板及扩展基板 (7 段) 的所有插槽。                                                             |
| 3   | 对象单元   | 支持三菱 iQ Platform 对应 PLC MELSEC-Q 系列所支持的所有输入输出单元。                                    |
| 4   | 指定设备范围 | 输入信号: X0 ~ XFFF (4096 点)<br>输出信号: Y0 ~ YFFF (4096 点)                                |

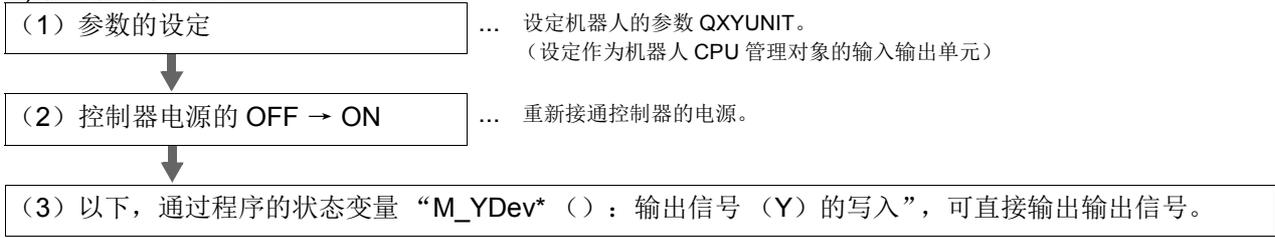
### (2) 操作步骤的概要

#### 1) 参照输入输出信号时



※1) 要通过 PLC CPU (或是连接在 PLC CPU 上的 GX Developer/GX Works) 参照机器人作为管理 CPU 的信号，需要将“获取组外的输入·输出状态”的设定设为有效。详细内容请参照「(4)PLC 的“多 CPU 设定”」。

2) 输出输出信号时



(3) 参数的说明

表 5-46 中对相关参数的概要进行说明。详细内容请参照第 418 页的「5.2 信号参数」的各说明页。

表 5-46: 相关参数

| 参数                  | 参数名                    | 排列数<br>文字数 | 内容说明                                                                                      | 说明页     |
|---------------------|------------------------|------------|-------------------------------------------------------------------------------------------|---------|
| PLC 输入输出信号的<br>获取设定 | QXYREAD                | 整数 2       | 选择其他号机 CPU 管理的输入输出单元（包括输入输出混合单元）的输入输出信号参照功能的有效 / 无效。<br>输入信号 (X) 和输出信号 (Y) 可分开设定。         | 第 420 页 |
| PLC 输入输出单元设定        | QXYUNITn<br>(n: 1 ~ 8) | 整数 7       | 设定以机器人 CPU 作为管理 CPU 的输入输出单元 / 输入输出混合单元。<br>指定单元类型、起始输入输出号码、基板号码、插槽号码、输入输出点数、报警时输出模式、响应时间。 | 第 420 页 |

## (4) PLC 的“多 CPU 设定”

同时勾选 PLC 侧的“多 CPU 设定”的“组外的输入输出设定”中的“获取组外的输入状态”和“获取组外的输出状态”，将组外的输入状态、输出状态均设为可获取。

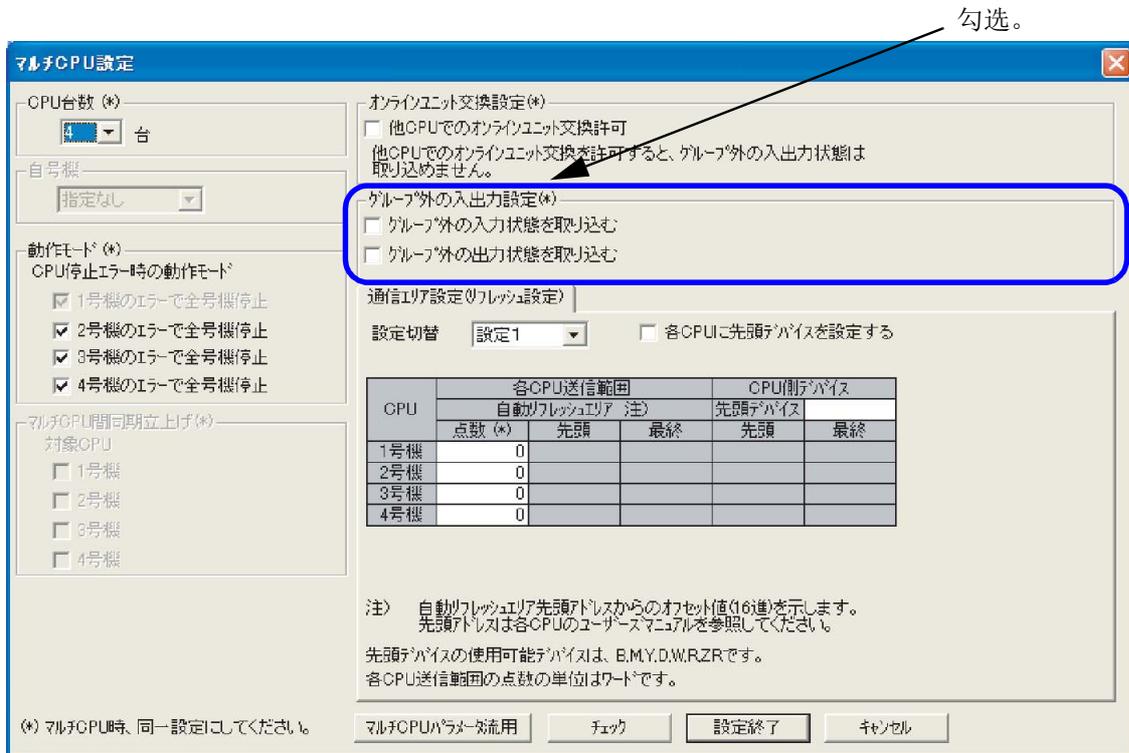


图 5-29: GX Developer 的 PLC 参数的多 CPU 设定画面

## 【补充】

要通过 PLC CPU（或是连接在 PLC CPU 上的 GX Developer/GX Works）参照机器人作为管理 CPU 的信号，需要使“获取组外的输入・输出状态”的设定有效。

（有关功能的详细内容，请参照 PLC 的使用说明书）

## (5) 机器人状态变量的说明

表 5-47 中对相关机器人状态变量的概要进行说明。详细内容请参照第 287 页的「4.14.2 各机器人状态变量（系统状态变量）的说明」的各说明页。

表 5-47: 相关的机器人状态变量

| 变量名     | 内容                                                                                                | 说明页     |
|---------|---------------------------------------------------------------------------------------------------|---------|
| M_XDev  | 以位为单位读取 PLC 输入信号 (X)。<br>例) 1 M1%=M_XDev (1) ' 在 M1 中代入 PLC 输入信号 1 号的值 (1 或 0)                    | 第 348 页 |
| M_XDevB | 以字节为单位读取 PLC 输入信号 (X)。<br>例) 1 M2%=M_XDevB (&H10) ' 在 M2 中代入从 PLC 输入信号 10 号 (16 进制数) 开始的 8 位的值    | 第 348 页 |
| M_XDevW | 以字为单位读取 PLC 输入信号 (X)。<br>例) 1 M4%=M_XDevW (&H20) ' 在 M4 中代入从 PLC 输入信号 20 号 (16 进制数) 开始的 16 位的值    | 第 348 页 |
| M_XDevD | 以双字为单位读取 PLC 输入信号 (X)。<br>例) 1 M5&=M_XDevD (&H100) ' 在 M5 中代入从 PLC 输入信号 100 号 (16 进制数) 开始的 32 位的值 | 第 348 页 |
| M_YDev  | 以位为单位读取 / 写入 PLC 输出信号 (Y)。<br>例) 1 M_YDev (2) =1 Dly 0.5 ' 使 PLC 输出信号 2 号开启 0.5 秒 (脉冲输出)          | 第 350 页 |
| M_YDevB | 以字节为单位读取 / 写入 PLC 输出信号 (Y)。<br>例) 1 M_YDevB (&H10) =&HFF ' 开启从 PLC 输出信号 10 号 (16 进制数) 开始的 8 位     | 第 350 页 |

| 变量名     | 内容                                                                                                                                 | 说明页     |
|---------|------------------------------------------------------------------------------------------------------------------------------------|---------|
| M_YDevW | 以字为单位读取 / 写入 PLC 输出信号 (Y)。<br>例) 1 M_YDevW (&H20) =&HFFFF ' 开启从 PLC 输出信号 20 号 (16 进制数) 开始的 16 位                                    | 第 350 页 |
| M_YDevD | 以双字为单位读取 / 写入 PLC 输出信号 (Y)。<br>例) 1 M_YDevD (&H100) =P1.X * 1000 ' 将位置变量 P1 的 X 座标值乘以 1000 后存储到从 PLC 输出信号 100 号 (16 进制数) 开始的 32 位中 | 第 350 页 |

## 5.26 在机器人CPU之间直接通信

在 F-Q 系列的控制器中，在多个机器人 CPU 之间直接进行信号收发的功能。可不用经由 PLC 的梯形图程序，在多个机器人之间更高速地获取互锁等接口。此外，还可以获取运动 CPU 等机器人 CPU 以外的共有内存情报。

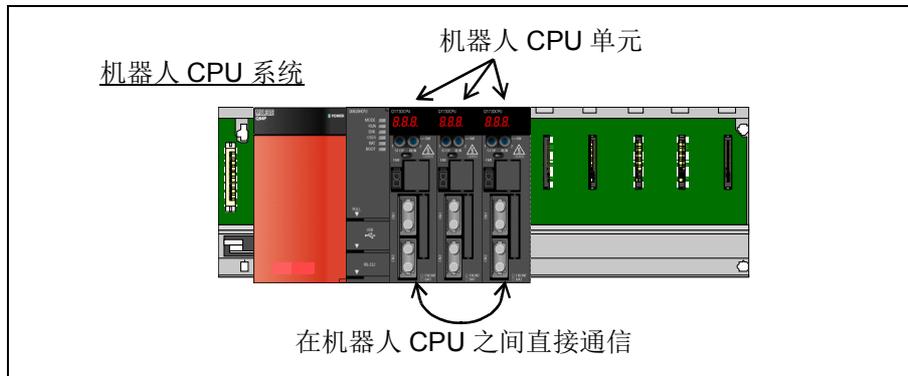


图 5-30: 机器人 CPU 间直接通信

### (1) 规格

本功能已加入到了表 5-48 中所示对象软件版本以后的软件中，可不用设定参数而直接使用。

规格内容见表 5-48。

表 5-48: 规格

| No | 项目          | 规格                                                                                                                      |
|----|-------------|-------------------------------------------------------------------------------------------------------------------------|
| 1  | 对象软件版本      | 在高于以下软件版本时，本功能有效。<br>本体: R2b<br>注) 对 R32TB/R33TB、R56TB/R57TB、RT ToolBox2 的软件版本没有限制。                                     |
| 2  | 指定设备范围      | 多 CPU 间共有设备 (U3En¥G10000 ~ G24335) 中可指定的范围。                                                                             |
| 3  | 专用输入输出信号的分配 | 通过专用信号进行的机器人控制，将通过 PLC (1 号机) 实施。<br>不能将专用输入输出参数分配到 2 号机以后的共有内存设备。<br>将专用输入输出信号分配到共有内存设备以使用时，请务必分配 PLC (1 号机) 的共有内存以使用。 |

### (2) 利用方法

通过使用程序的状态变量“M\_UDevW：以字为单位 (16 位量) 写入 / 参照”、“M\_UDevD：以双字为单位 (32 位量) 写入 / 参照”，可进行多 CPU 间共有设备 (U3En¥G □) 的读取 / 写入。

### (3) 机器人状态变量的说明

表 5-49 中对相关机器人状态变量的概要进行说明。详细内容请参照第 287 页的「4.14.2 各机器人状态变量 (系统状态变量) 的说明」的各说明页。

表 5-49: 相关的机器人状态变量

| 变量名     | 内容                                                                                                                                                    | 说明页     |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| M_UDevW | 以字为单位读取 / 写入多 CPU 间共有设备 (U3En¥G □)。<br>例) 1 M_UDevW (&H3E1, 10010) =&HFFFF' 在 2 号机 CPU 的共有内存地址 10010 中写入 &HFFFF (16 进制数)。                             | 第 341 页 |
| M_UDevD | 以双字为单位读取 / 写入多 CPU 间共有设备 (U3En¥G □)。<br>例) 1 M_UDevD (&H3E1, 10011) =P1.X * 1000' 将位置变量 P1 的 X 坐标值乘以 1000 后得出的值写入到 2 号机 CPU 的共有内存地址 10011/10012 的双字中。 | 第 341 页 |

## 5.27 关于冗余系统错误的动作选择参数

关于表 5-50 的冗余系统错误，通过设定参数：CATEGORY，可以选择发生错误时的 [RESET] 按键输入下的动作。

表 5-50：对象错误一览

| 错误号码  | 内容          |
|-------|-------------|
| H0039 | 门开关开路信号配线异常 |
| H0046 | 可用设备配线异常    |
| H0051 | 外部紧急停止配线异常  |

[RESET] 按键输入时的动作，为选择表 5-51 中的任意一个。

表 5-51：参数设定值相应的动作

| 参数设定值     | 内容             |
|-----------|----------------|
| 3（可错误复位）  | 可以进行错误复位。（初始值） |
| 4（不可错误复位） | 不可进行错误复位。      |

为了反映本设定内容，应务必重新接通控制器的电源。

### [注意]

本设定为对相关信号线或配线的异常消除后的动作进行选择。  
 残留有信号线及配线异常的情况下，无论是否设定有本参数，上述的错误均不会解除。

## 6 外部输入输出的功能

### 6.1 种类

- (1) 专用输出输入 . . . . . 显示机器人的程序执行、停止等的远程操作、执行中情报、伺服电源状态等的状态的输入输出信号。  
在各输入输出信号将功能分配使用。分配方法以各专用的参数设定使用信号号码。  
(请参照第 509 页的「6.2PLC 连接输出输入功能」) 及、紧急停止输入 (参照第 544 页的「6.7 紧急停止输入」)。有预先分配使用频率高的信号。可以追加、变更。
- (2) 通用输出输入 . . . . . 在机器人的程序里, 使用在 PLC 等的通信。在抓取周边装置的定位信号及确认机器人的位置时使用。
- (3) 抓手输出输入 . . . . . 抓手用的控制信号。使用在抓手的开闭指示及关于抓手上附的感应器情报的收集。可用使用者程控。机器人的抓手尖端附近有配线。(抓手输出输入为选配。)

表 6-1: 输入输出信号全体图

|            | 输入输出信号        | 使用方法                                                                                                                                                           |
|------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 抓手输出输入     | 900 ~ 907     | 可以在 M_In, M_Inb, M_Inw, M_Out, M_Outb, M_Outw 变量里参照 / 代入指令 HOpen, HClose<br><br>例) If M_In (900) =1 Then M_Out (900) =1<br>HOpen 1, HClose 1                   |
| PLC 连接输出输入 | 10000 ~ 18191 | 可以在 M_In, M_Inb, M_Inw, M_Out, M_Outb, M_Outw 变量里参照 / 代入指令<br>例) If M_In (10080) =1 Then M_Out (10080) =1<br><br>注) 专用输出的分配信号无法使用 M_Out, M_Outb, M_Outw 变量做输出。 |

## 6.2 PLC连接输出输入功能

本功能只在 CR7xx-Q 系列控制器中有效。QnUD (H) CPU (以后统称为 PLC CPU) 及 Q172DRCPU (以后统称为机器人 CPU) 是使用 CPU 间的共有内存, 依据 PLC 阶梯语言做相互通信。在相互通信是使用共有的内存「多 CPU 间高速通信区域<sup>注1)</sup>」。机器人 CPU 输入信号、输出信号也是使用 10000~18191 为止的信号号码。

### 6.2.1 参数设定

为了使用 PLC 连接功能, 必须各别在 PLC CPU 及机器人 CPU 做多重 CPU 关连的参数设定。机器人 CPU 以 RT ToolBox 或示教单元 (R32TB、R56TB) 做参数设定、PLC CPU 则用 GX Developer 做参数设定。详细内容请参照各设定 TOOL 的操作说明书。

#### (1) PLC CPU 的参数设定

使用 GX-Developer 执行多重 CPU 的参数设定。下列为系统设置 PLC CPU1 台、机器人 CPU1 台时的范例。

##### 1) CPU 台数

在 CPU 台数多重 CPU 系统设定在基本 BASE 单元上安装的 CPU 单元的台数。

##### 2) 多重 CPU 间同步开始

从机器人 CPU 电源投入开始到系统起动的为止大约花费 10 几秒。请在多重 CPU 系统设定同时开始 (确认进入状态)。

##### 3) 多重 CPU 间高速通信区域设定

将点数以字符单位设定。关于机器人 CPU, 因为只能使用不到 1K 字符, 因此请设定在 1K 字符。

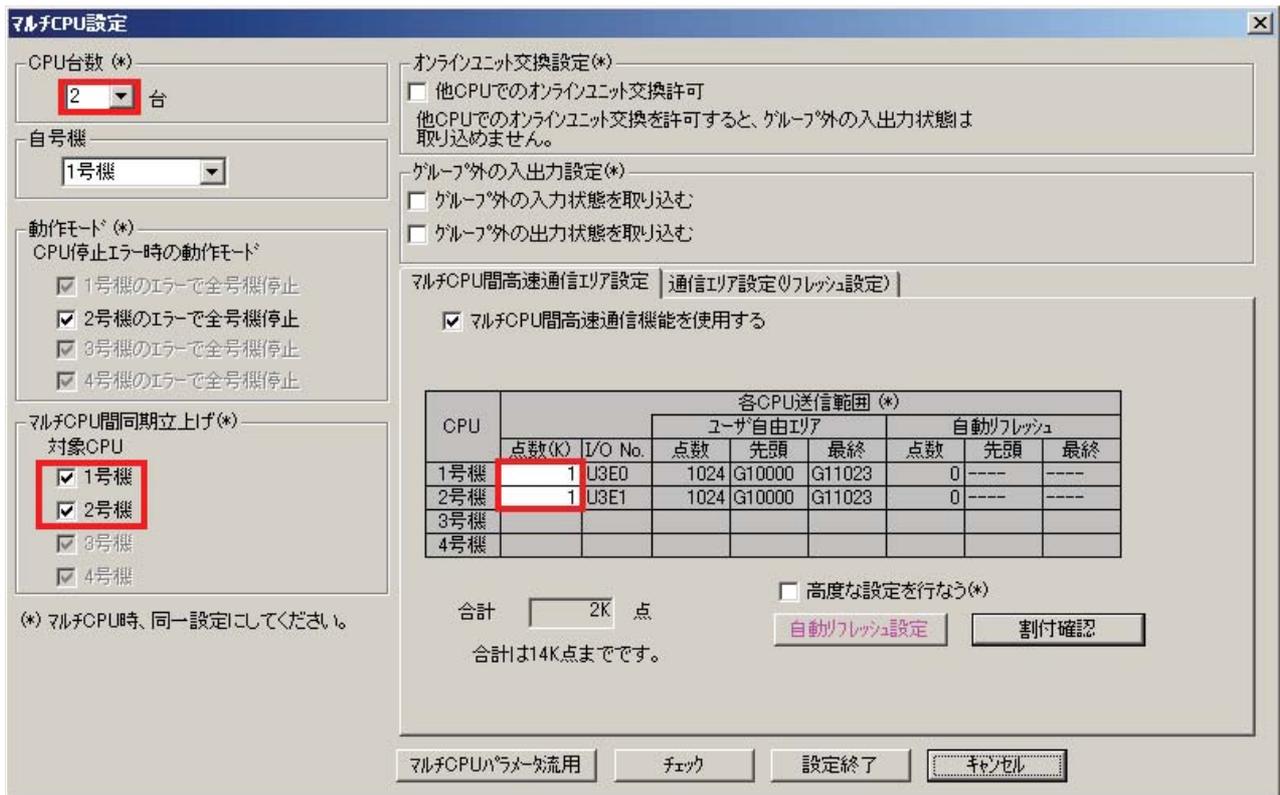


图 6-1: PLC CPU: 依据 GX Developer 设定画面范例

在多 CPU 间高速通信区域, 可设定使用者自由区域及自动更新区域, 但是, 机器人 CPU (Q172DRCPU) 因为没有对应自动更新区域, 自动更新区域的点数, 通常设定为 0。

注 1) 关于多 CPU 及多 CPU 间高速通信区域, 请参照 QCPU 的操作手册 (QCPU 使用者操作手册 多 CPU 系统编)。

## (2) 机器人 CPU 的参数设定

使用 RT ToolBox 执行多 CPU 的参数设定。

表 6-2: 机器人 CPU 的参数设定

| 参数名                    | 内容                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 出货时的设定值                                                                           |      |   |                   |   |           |                        |           |                                                                                   |                      |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------|---|-------------------|---|-----------|------------------------|-----------|-----------------------------------------------------------------------------------|----------------------|
| QMLTCPUN               | 多 CPU 台数设定<br>在多 CPU 系统设定在基本基础单元上安装 CPU 单元的台数。<br>范围: 1 ~ 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2                                                                                 |      |   |                   |   |           |                        |           |                                                                                   |                      |
| QMLTCPUn<br>n = 1~4    | 多重 CPU 号机高速通信区域设定 n= 1 ~ 4)<br>在多重 CPU 系统, 设定在 1 ~ 4 号间的多重 CPU 间高速通信功能的各 CPU 单元间执行收送信点数。<br>关于全 CPU, 必须符合参数设定值。没有符合参数设定值的情况下, PLC 会 CPU 会发生报警, 因此请符合各 CPU 的参数设定值。<br><br>要素 1: 使用者自由区域的大小 (K 点)<br>范围: 1 ~ 14 (最大)<br><br>表 6-3: CPU 台数别的设定范围<br><table border="1" data-bbox="443 846 865 1064"> <thead> <tr> <th>CPU 台数</th> <th>设定范围</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>0 ~ 14K 点</td> </tr> <tr> <td>3</td> <td>0 ~ 13K 点</td> </tr> <tr> <td>4</td> <td>0 ~ 12K 点</td> </tr> </tbody> </table><br>要素 2: 自动更新的点数 (点)<br>范围: 0~14335<br>但是同为机器人 CPU 没有对应自动更新, 因此机器人 CPU 的自动更新区域的点数通常设定为 0。<br>要素 3: 系统区域的大小 (K 点)<br>范围: 1 或 2<br>要素 4: 多重 CPU 同步开始 1: 执行、0: 不执行)<br>机器人 CPU 在开始会花费时间, 因此基本上就保持 1 (同步), 请不要改变设定。全 CPU 必须在同一设定。 | CPU 台数                                                                            | 设定范围 | 2 | 0 ~ 14K 点         | 3 | 0 ~ 13K 点 | 4                      | 0 ~ 12K 点 | 1,0,1,1                                                                           |                      |
| CPU 台数                 | 设定范围                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| 2                      | 0 ~ 14K 点                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| 3                      | 0 ~ 13K 点                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| 4                      | 0 ~ 12K 点                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| IQMEM <sup>注 1)</sup>  | 选择共有内存扩展功能。<br>各个位分别分配有功能。1/0= 有效 / 无效<br><table border="0" data-bbox="402 1482 1002 1684"> <tr> <td style="text-align: right;">15</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td colspan="3">00000000 00000000</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">bit2-3、<br/>5-15<br/>未使用</td> <td style="border-left: 1px solid black; padding: 2px;"> </td> <td style="padding: 2px;"> ...bit0: 扩展功能的使用<br/>  ....bit1: PLC 直接执行功能<br/> <br/>  .....bit4: 机器人间协调控制功能</td> </tr> </table>                                                                                                                                                                                                            | 15                                                                                | 0    |   | 00000000 00000000 |   |           | bit2-3、<br>5-15<br>未使用 |           | ...bit0: 扩展功能的使用<br>  ....bit1: PLC 直接执行功能<br> <br>  .....bit4: 机器人间协调控制功能        | 000000000000<br>0000 |
| 15                     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| 00000000 00000000      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| bit2-3、<br>5-15<br>未使用 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | ...bit0: 扩展功能的使用<br>  ....bit1: PLC 直接执行功能<br> <br>  .....bit4: 机器人间协调控制功能        |      |   |                   |   |           |                        |           |                                                                                   |                      |
| IQSPEC <sup>注 1)</sup> | 设定 CR7xx-Q 系列控制器的功能。<br>各个位分别分配有功能。1/0= 有效 / 无效<br><table border="0" data-bbox="402 1780 1225 2011"> <tr> <td style="text-align: right;">15</td> <td style="text-align: center;">0</td> <td></td> </tr> <tr> <td colspan="3">00000000 00000000</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">bit1-15<br/>未使用</td> <td style="border-left: 1px solid black; padding: 2px;"> </td> <td style="padding: 2px;"> ...bit0: 共有内存写入方向<br/>=0: 读取、写入均按从起始地址到结束地址的顺序实施<br/>=1: 读取按从起始地址开始、写入按从结束地址开始的顺序实施</td> </tr> </table>                                                                                                                                                                                                      | 15                                                                                | 0    |   | 00000000 00000000 |   |           | bit1-15<br>未使用         |           | ...bit0: 共有内存写入方向<br>=0: 读取、写入均按从起始地址到结束地址的顺序实施<br>=1: 读取按从起始地址开始、写入按从结束地址开始的顺序实施 | 000000000000<br>0001 |
| 15                     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| 00000000 00000000      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                   |      |   |                   |   |           |                        |           |                                                                                   |                      |
| bit1-15<br>未使用         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | ...bit0: 共有内存写入方向<br>=0: 读取、写入均按从起始地址到结束地址的顺序实施<br>=1: 读取按从起始地址开始、写入按从结束地址开始的顺序实施 |      |   |                   |   |           |                        |           |                                                                                   |                      |

注 1) 关于扩展功能（PLC 直接执行功能）的详细内容，请参阅另一手册“使用说明书 /CR750-Q 系列、CRnQ-700 系列 iQ Platform 对应扩展功能说明书（BFP-A8757）”。

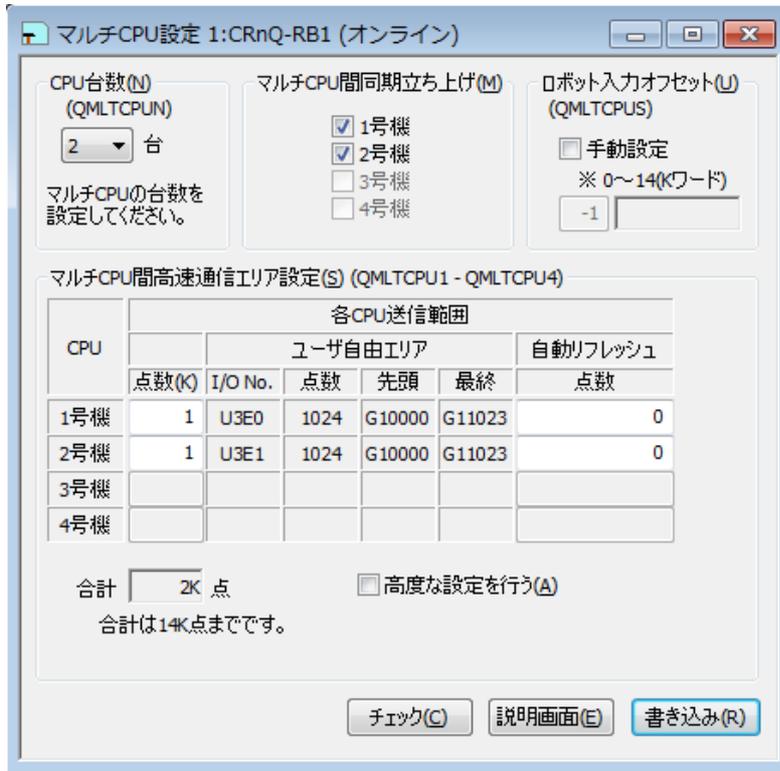


图 6-2: 机器人 CPU：通过 RT ToolBox 进行设定的画面示例

◆◆◆关于多 CPU 的对象◆◆◆

多 CPU 的对象为下列对应 iQ Platform 的 CPU、基板。（截止到 2010 年 8 月）

| 项目        | 型号                                                                                                                                                           | 备注                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| PLC CPU   | 通用型 QCPU<br>Q03UD (E) CPU、Q04UD (E)<br>HCPU、Q06UD (E) HCPU、<br>Q10UD (E) HCPU、Q13UD (E)<br>HCPU、Q20UD (E) HCPU、<br>Q26UD (E) HCPU、<br>Q50UDEHCPU、Q100UDEHCPU | <ul style="list-style-type: none"> <li>支持多 CPU 间高速通信的产品</li> <li>1 号机必须是 PLC CPU</li> </ul> |
| 机器人 CPU   | Q172DRCPU                                                                                                                                                    |                                                                                             |
| 运动控制器 CPU | Q172DCPU、Q173DCPU                                                                                                                                            |                                                                                             |
| NC CPU    | Q172NCCPU                                                                                                                                                    |                                                                                             |
| 基板        | 多 CPU 间高速主基板<br>Q38DB、Q312DB                                                                                                                                 | <ul style="list-style-type: none"> <li>支持多 CPU 间高速通信的产品</li> </ul>                          |

机器人 CPU 使用 RT ToolBox 或示教盒（R33TB、R57TB）、PLC CPU 使用 GX Developer、运动控制器 CPU 使用 MT Devevoper、NC CPU 使用 Remote monitor Tool 等进行参数设定。关于详细内容，请参阅各工具的使用说明书。

### 6.2.2 CPU共有内存和机器人输入输出信号的对应

PLC CPU 将 CPU 共有内存以 U3E0\G10000 的样子储存。机器人 CPU No.n 的 CPU 共有内存则以 U3En\G10000 的样子储存。

(n=1~3、最多可使用 3 台的机器人 CPU)

机器人 CPU 的输入输出信号号码各为 10000 ~18191。

请注意 PLC 侧会变成 Word 设备、机器人侧会变成 Bit 设备。

此外，CPU 共有内存各机器人输入输出信号的对应如下表所示，无法变更。

表 6-4: CPU 共有内存和机器人输入输出信号的对应

| PLC (Word 设备) |                           | 机器人 (Bit 设备) |                              |
|---------------|---------------------------|--------------|------------------------------|
| 输出            | U3E0\G10000 ~ U3E0\G10511 | 输入           | 机器人 CPU No.1 / 10000 ~ 18191 |
|               | U3E0\G10512 ~ U3E0\G11023 |              | 机器人 CPU No.2 / 10000 ~ 18191 |
|               | U3E0\G11024 ~ U3E0\G11535 |              | 机器人 CPU No.3 / 10000 ~ 18191 |
| 输入            | U3E1\G10000 ~ U3E1\G10511 | 输出           | 机器人 CPU No.1 / 10000 ~ 18191 |
|               | U3E2\G10000 ~ U3E2\G10511 |              | 机器人 CPU No.2 / 10000 ~ 18191 |
|               | U3E3\G10000 ~ U3E3\G10511 |              | 机器人 CPU No.3 / 10000 ~ 18191 |

### 6.2.3 PLC梯形图的范例

将操作盘的「机器人操作权有效钮」X0 开启，显示将操作的「机器人操作权有效中灯」Y20，输出机器人的操作权有效状态输出例。多重 CPU 的构成：1 号机：PLC QnUD (H) CPU、2 号机：机器人 Q172DRCPU。

**【说明】**

< 第 0 ~ 16 行 >

将 M100 ~ M131 写入 U3E0\G10000、U3E0\G10001 的共有设备内存，作为从 PLC 往机器人的写入。将 U3E1\G10000、U3E1\G10001 的共有设备内存，在 M200 ~ M231 的位元设备读出，作为从机器人往 PLC 的输出。

< 第 17 ~ 22 行 >

将 X0 和 M105 开启，且在 M105 对应 PLC 的 U3E0\G10000 的位 5 开启。于是，机器人的输入 10005 为开启，用专用输入信号被分配的操作权变成有效。

操作权变成有效的话，用专用输出信号所分配的机器人的输出 10005 会开启，且机器人的 U3E1\G10000 的位 5 会开启。于是，在 U3E1\G10000 的位 5 对应 PLC 的 M205 开启，且 Y20 开启。而且，在此例表示位设备 M201 (U3E0\G10000 的位 1/ 也就是机器人的输出 10001) 控制器电源开启完成 (将外部输入信号数据输出)。

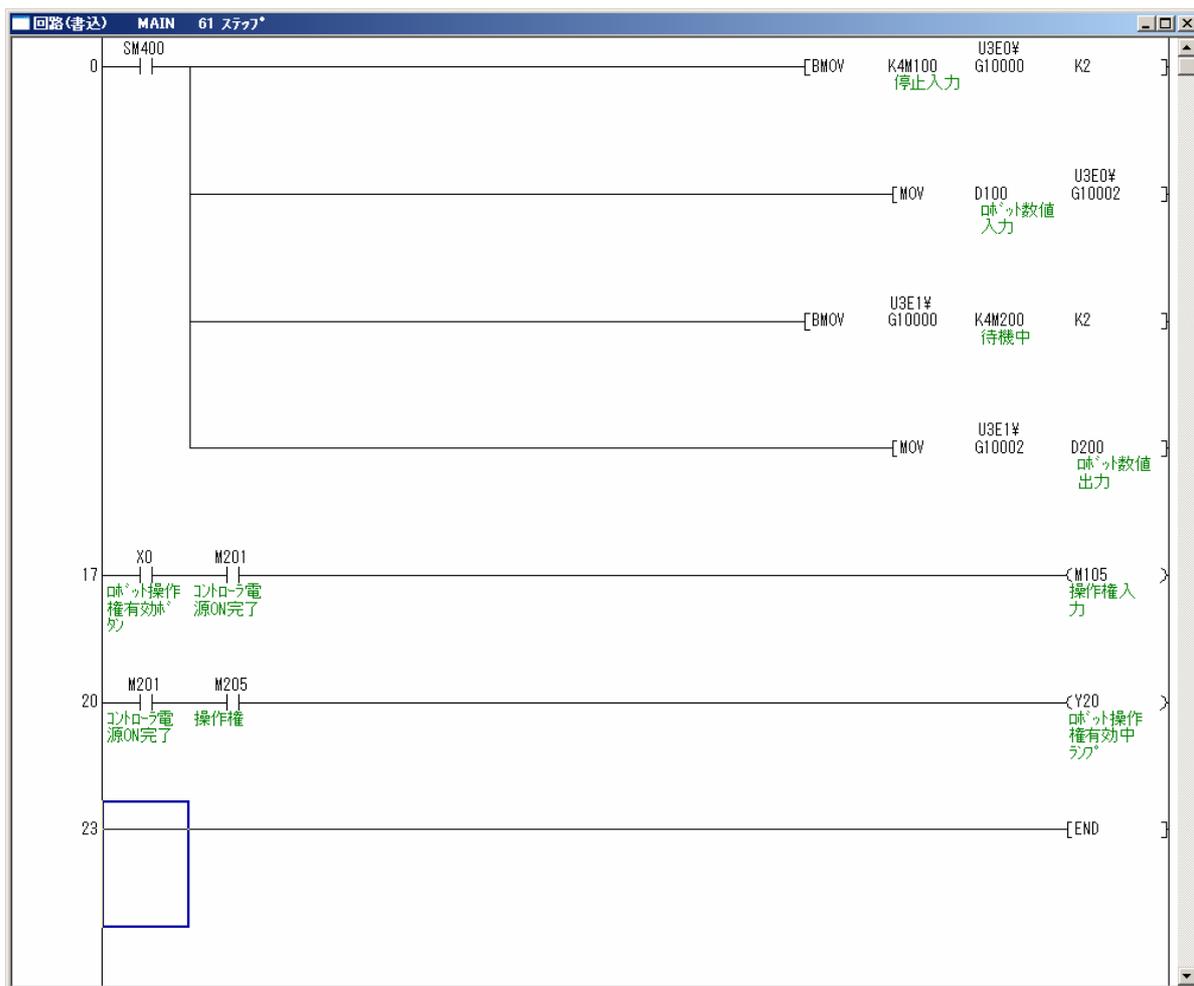


图 6-3: PLC 阶梯图例

## 6.2.4 专用输入输出信号的分配（工厂出货时设定）

工厂出货时的专用输出输入信号的分配如表 6-5 所示。

表 6-5: 专用输入输出信号的分配（工厂出货时设定）

| 参数名      | 输入信号名称（※ 操作权必要） | 输出信号名称        | 输入    | 输出    | G 设备<br>注 1) |
|----------|-----------------|---------------|-------|-------|--------------|
| STOP     | 停止输入（不可变更分配）    | 中断中输出（不可变更分配） | 10000 | 10000 | G10000       |
| RCREADY  | -               | 控制器电源 ON 完成   | -     | 10001 |              |
| ATEXTMD  | -               | 远程模式输出        | -     | 10002 |              |
| TEACHMD  | -               | 示教模式输出        | -     | 10003 |              |
| ATTOPMD  | -               | OP 模式输出       | -     | 10004 |              |
| IOENA    | 操作权输入           | 操作权输出         | 10005 | 10005 |              |
| START    | 起动输入（※）         | 运行中输出         | 10006 | 10006 |              |
| STOPSTS  | -               | 停止信号输入中       | -     | 10007 |              |
| SLOTINIT | 程序重置（※）         | 程序选择可能输出      | 10008 | 10008 |              |
| ERRRESET | 报警复位输入          | 报警发生中输出       | 10009 | 10009 |              |
| SRVON    | 伺服 ON 输入（※）     | 伺服 ON 中输出     | 10010 | 10010 |              |
| SRVOFF   | 伺服 OFF 输入       | 伺服 ON 不可输出    | 10011 | 10011 |              |
| CYCLE    | 循环停止输入          | 循环停止动作中输出     | 10012 | 10012 |              |
| SAFEPOS  | 退避点回归信号（※）      | 退避点回归中输出      | 10013 | 10013 |              |
| BATERR   | -               | 电池电压低下        | -     | 10014 |              |
| OUTRESET | 通用输出信号复位（※）     |               | 10015 | -     |              |
| HLVLERR  | -               | 高 Level 报警输出  | -     | 10016 |              |
| LLVLERR  | -               | 低 Level 报警输出  | -     | 10017 |              |
| CLVLERR  | -               | 警告 Level 报警输出 | -     | 10018 |              |
| EMGERR   | -               | 紧急停止输出        | -     | 10019 |              |
| PRGSEL   | 程序选择输入          | -             | 10020 | -     |              |
| OVRDSEL  | 速度比例选择输入        | -             | 10021 | -     |              |
| PRGOUT   | 程序号码输入要求        | 程序号码输出中       | 10022 | 10022 |              |
| LINEOUT  | 行号码输出要求         | 行号码输出中        | 10023 | 10023 |              |
| OVRDOUT  | 速度比例值输出要求       | 速度比例值输出中      | 10024 | 10024 |              |
| ERROUT   | 报警号码输出要求        | 报警号码输出中       | 10025 | 10025 |              |
| -        | -               | -             | -     | -     |              |
| -        | -               | -             | -     | -     |              |
| -        | -               | -             | -     | -     |              |
| -        | -               | -             | -     | -     |              |
| -        | -               | -             | -     | -     |              |
| -        | -               | -             | -     | -     |              |
| IODATA   | 数值输入 0          | 数值输出 0        | 10032 | 10032 | G10002       |
|          | 数值输入 1          | 数值输出 1        | 10033 | 10033 |              |
|          | 数值输入 2          | 数值输出 2        | 10034 | 10034 |              |
|          | 数值输入 3          | 数值输出 3        | 10035 | 10035 |              |

| 参数名      | 输入信号名称 (※ 操作权必要) | 输出信号名称       | 输入    | 输出    | G 设备<br>注 1) |
|----------|------------------|--------------|-------|-------|--------------|
| IODATA   | 数值输入 4           | 数值输出 4       | 10036 | 10036 | G10002       |
|          | 数值输入 5           | 数值输出 5       | 10037 | 10037 |              |
|          | 数值输入 6           | 数值输出 6       | 10038 | 10038 |              |
|          | 数值输入 7           | 数值输出 7       | 10039 | 10039 |              |
|          | 数值输入 8           | 数值输出 8       | 10040 | 10040 |              |
|          | 数值输入 9           | 数值输出 9       | 10041 | 10041 |              |
|          | 数值输入 10          | 数值输出 10      | 10042 | 10042 |              |
|          | 数值输入 11          | 数值输出 11      | 10043 | 10043 |              |
|          | 数值输入 12          | 数值输出 12      | 10044 | 10044 |              |
|          | 数值输入 13          | 数值输出 13      | 10045 | 10045 |              |
|          | 数值输入 14          | 数值输出 14      | 10046 | 10046 |              |
|          | 数值输入 15          | 数值输出 15      | 10047 | 10047 |              |
| HNDCNTL1 | -                | 抓手输出信号状态 900 | -     | 10048 | G10003       |
|          | -                | 抓手输出信号状态 901 | -     | 10049 |              |
|          | -                | 抓手输出信号状态 902 | -     | 10050 |              |
|          | -                | 抓手输出信号状态 903 | -     | 10051 |              |
|          | -                | 抓手输出信号状态 904 | -     | 10052 |              |
|          | -                | 抓手输出信号状态 905 | -     | 10053 |              |
|          | -                | 抓手输出信号状态 906 | -     | 10054 |              |
| HNDSTS1  | -                | 抓手输入信号状态 900 | -     | 10056 | G10003       |
|          | -                | 抓手输入信号状态 901 | -     | 10057 |              |
|          | -                | 抓手输入信号状态 902 | -     | 10058 |              |
|          | -                | 抓手输入信号状态 903 | -     | 10059 |              |
|          | -                | 抓手输入信号状态 904 | -     | 10060 |              |
|          | -                | 抓手输入信号状态 905 | -     | 10061 |              |
|          | -                | 抓手输入信号状态 906 | -     | 10062 |              |
|          | -                | 抓手输入信号状态 907 | -     | 10063 |              |
| USRAREA  | -                | 使用者定义领域 1    | -     | 10064 | G10004       |
|          | -                | 使用者定义领域 2    | -     | 10065 |              |
|          | -                | 使用者定义领域 3    | -     | 10066 |              |
|          | -                | 使用者定义领域 4    | -     | 10067 |              |
|          | -                | 使用者定义领域 5    | -     | 10068 |              |
|          | -                | 使用者定义领域 6    | -     | 10069 |              |
|          | -                | 使用者定义领域 7    | -     | 10070 |              |
| -        | 使用者定义领域 8        | -            | 10071 |       |              |

注 1) 多 CPU 间共有 Device 的地址。(从 PLC CPU 侧看到的地址)

### 6.3 专用输出输入

专用输出输入信号有表 6-6 中所示功能。

在此，在各别的参数使用参数输出单元来分配信号号码。信号号码的分配，在各参数以“输入信号”、“输出信号”的顺序指定信号号码。参数的设定方法请参照第 84 页的「3.14 参数画面的操作」。此外，分配信号号码指定 -1 的话，其信号会变成未使用。

输出输入的参数可以在示教单元的参数画面或计算机支持软件（选配）的维修 TOOL 里设定。此外，请参照第 536 页的「6.5.2 时序图例」中所示的信号时序图。

专用输入信号在控制器模式为“AUTOMATIC”，且操作权输入信号为 ON 时有效。

表 6-6: 专用输出输入一览表

| 参数名              | 区分 | 名称          | 功能                                                                                                                                                                                            | 信号 Level | 工厂出货信号号码<br>输入, 输出 |          |
|------------------|----|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------|----------|
|                  |    |             |                                                                                                                                                                                               |          | CR7xx-Q            | CR7xx-D  |
| RCREADY          | 输入 | -           | -                                                                                                                                                                                             |          | -1 (无意义)           | -1 (无意义) |
|                  | 输出 | 控制器电源 ON 完成 | 电源 ON 完成后, 输出外部输入信号为受理可能。                                                                                                                                                                     |          | 10001              | -1       |
| ATEXTMD          | 输入 | -           | -                                                                                                                                                                                             |          | -1 (无意义)           | -1 (无意义) |
|                  | 输出 | 远程模式输出      | 输出操作面板的按键的状态为以自动模式由外部的操作为有效。为远程模式。 <u>以 IO 控制的时候, 此信号开启状态会变成条件。</u>                                                                                                                           |          | 10002              | -1       |
| TEACHMD          | 输入 | -           | -                                                                                                                                                                                             |          | -1 (无意义)           | -1 (无意义) |
|                  | 输出 | 示教模式输出      | 输出操作面板的按键状态为示教模式。                                                                                                                                                                             |          | 10003              | -1       |
| ATTOPMD          | 输入 | -           | -                                                                                                                                                                                             |          | -1 (无意义)           | -1 (无意义) |
|                  | 输出 | 自动模式输出      | 输出以操作面板的按键的状态为自动模式的操作面板的有效。                                                                                                                                                                   |          | 10004              | -1       |
| IOENA            | 输入 | 操作权输入信号     | 使外部信号控制的操作权有效 / 无效。                                                                                                                                                                           | Level    | 10005,<br>10005    | 5,<br>3  |
|                  | 输出 | 操作权输出信号     | 输出外部信号控制的操作权有效状态。模式开关在以 AUTOMATIC 设定状态, 操作权输入信号开启而且在没有取得操作权的设备情况下, 依外部信号给予操作权。                                                                                                                |          |                    |          |
| START<br>(操作权必要) | 输入 | 起动输入        | 执行程序的起动。想起动指定的程序时, 在程序选择信号「PRGSEL」和数值输入「IODATA」, 请在选择程序后输入起动信号。但是, 将参数「PST」为有效的话, 数值输入 (IODATA) 开始以读入程序号码起动。<br>(变成不要选择程序。)<br>多任务运行时执行全部任务插槽。<br>但是, 在参数「SLT**」起动条件为 ALWAYS 和 ERROR 设定的插槽除外。 | Edge     | 10006,             | 3,       |
|                  | 输出 | 运行中输出       | 输出程序为运行中。多任务运行时在至少一个任务插槽为运行中时开启。<br>但是, 在参数「SLT**」起动条件为 ALWAYS 和 ERROR 设定的插槽除外。                                                                                                               |          | 10006              | 0        |

| 参数名                 | 区分 | 名称           | 功能                                                                                                                                                        | 信号 Level | 工厂出货信号号码<br>输入, 输出 |             |
|---------------------|----|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------|-------------|
|                     |    |              |                                                                                                                                                           |          | CR7xx-Q            | CR7xx-D     |
| STOP                | 输入 | 停止输入         | 停止运行中的程序。(起动条件为 ALWAYS 和 ERROR 设定的插槽除外。) 停止输入会固定在输入信号 0 号, 无法变更。<br>多任务运行时将全部任务插槽停止。但是, 以参数「SLT**」起动条件为 ALWAYS 和 ERROR 设定的插槽除外。依据参数 IND, A 接点 / B 接点可以变更。 | Level    | 10000<br>(不可变更)    | 0<br>(不可变更) |
|                     | 输出 | 中断中输出        | 输出程序为中断中。<br>多任务运行时, 没有运行中插槽, 至少 1 个中断中的时候开启。<br>但是, 以参数「SLT**」起动条件为 ALWAYS 和 ERROR 设定的插槽除外                                                               |          | 10000              | -1          |
| STOP2               | 输入 | 停止输入         | 停止运行中的程序。<br>(规格和 STOP 参数相同)<br>与 STOP 参数不同, 可以变更信号的号码。                                                                                                   | Level    | -1,                | -1,         |
|                     | 输出 | 中断中输出        | 输出程序为中断中。<br>(规格和 STOP 参数相同)                                                                                                                              |          | -1                 | -1          |
| STOPSTS             | 输入 | -            | -                                                                                                                                                         |          | -1 (无意义)           | -1 (无意义)    |
|                     | 输出 | 停止信号输入中      | 输出停止为输入中。<br>(会变成全部设备的逻辑和)                                                                                                                                |          | 10007              | -1          |
| SLOTINIT<br>(操作权必要) | 输入 | 程序复位         | 解除式的中断中状态, 返回到执行行前头。<br>依据程序复位, 会变成程序可以选择状态。<br>在多任务使用的情况下, 执行对应全任务插槽的程序复位。<br><br>但是, 以参数「SLT**」起动条件为 ALWAYS 和 ERROR 设定的插槽除外                             | Edge     | 10008,             | -1,         |
|                     | 输出 | 程序选择可能输出     | 输出可选择程序状态。<br>程序在运行中 / 没有中断中的情况开启 (ON)。<br>多任务运行时在全任务插槽运行中 / 没有运行中的情况开启 (ON)。<br><br>但是, 以参数「SLT**」起动条件为 ALWAYS 和 ERROR 设定的插槽除外                           |          | 10008              | -1          |
| ERRRESET            | 输入 | 报警复位输入信号     | 解除报警状态。                                                                                                                                                   | Edge     | 10009,             | 2,          |
|                     | 输出 | 报警发生中输出信号    | 输出报警状态。                                                                                                                                                   |          | 10009              | 2           |
| SRVON<br>(操作权必要)    | 输入 | 伺服 ON 输入信号   | 开启机器人的伺服电源。多机器的情况下会开启全部机器人的伺服电源。                                                                                                                          | Edge     | 10010,             | 4,          |
|                     | 输出 | 伺服 ON 中输出信号  | 在机器人的伺服电源为开启的时候会变成 ON。<br>伺服电源为关闭的时候会变成 OFF。<br>多任务运行时, 没有运行中插槽, 至少 1 个中断中的时候开启                                                                           |          | 10010              | 1           |
| SRVOFF              | 输入 | 伺服 OFF 输入信号  | 将机器人的伺服电源关闭 (对象为全机器)。本信号输入中时伺服开启会变成不可以。                                                                                                                   | Level    | 10011,             | 1,          |
|                     | 输出 | 伺服 ON 不可输出信号 | 输出伺服电源开启不可的状态。(echo back)                                                                                                                                 |          | 10011              | -1          |

| 参数名                 | 区分 | 名称             | 功能                                                                                                                                                                                                               | 信号 Level | 工厂出货信号号码<br>输入, 输出 |           |
|---------------------|----|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------|-----------|
|                     |    |                |                                                                                                                                                                                                                  |          | CR7xx-Q            | CR7xx-D   |
| AUTOENA             | 输入 | 自动运行可输入        | 禁止在非自动做自动运行。<br>该信号为非激活且为 AUTOMATIC 模式时, 会发生 L5010 低等级错误。<br>在取得操作面板的操入和 IO 的操作时使用。没有一定要使用的必要。                                                                                                                   | Level    | -1,                | -1        |
|                     | 输出 | 自动运行可输出        | 输出自动运行可状态。                                                                                                                                                                                                       |          | -1                 | -1        |
| CYCLE               | 输入 | 循环停止输入信号       | 执行循环停止。                                                                                                                                                                                                          | Edge     | 10012,<br>10012    | -1,<br>-1 |
|                     | 输出 | 循环停止动作中输出信号    | 输出循环停止中动作中。<br>循环停止完成的话, 会关闭。                                                                                                                                                                                    |          |                    |           |
| MELOCK<br>(需要操作权)   | 输入 | 机械锁输入信号        | 将全机器设定 / 解除在机械锁状态。<br>在全插槽为选择可状态时, 可以设定 / 解除。程序选择可状态时会变成 Level。                                                                                                                                                  | Level    | -1,                | -1,       |
|                     | 输出 | 机械锁中输出信号       | 输出机械锁状态。<br>至少一个的机器为机器 Lock 状态的时候开启 (ON)。<br>机器 LOCK 状态时, 机器人不会动作, 程序运行会变可能。                                                                                                                                     |          | -1                 | -1        |
| SAFEPOS<br>(需要操作权)  | 输入 | 待避点回归输入信号      | 要求待避点回归动作。往参数「JSAFE」的位置做关节插补动作。速度依存在速度比例。请注意与周边装置的干涉。                                                                                                                                                            | Edge     | 10013,             | -1,       |
|                     | 输出 | 待避点回归中输出信号     | 输出待避点回归动作中。                                                                                                                                                                                                      |          | 10013              | -1        |
| BATERR              | 输入 | -              | -                                                                                                                                                                                                                |          | -1 (无意义)           | -1,       |
|                     | 输出 | 电池电压低下         | 输出控制器的电池电压低下。更换电池后, 重新输入控制器的电源的话, 输出为 OFF。使下列条件也会输出。<br>• 控制器电源 OFF 累计时间超过 14600。将电池消耗时间初始化的话, 输出为 OFF。                                                                                                          |          | 10014              | -1        |
| OUTRESET<br>(需要操作权) | 输入 | 通用输出信号复位       | 将通用输出信号复位。<br>依据参数 ORS1000 ~ ORS1816, 设定输入时的动作。                                                                                                                                                                  | Edge     | 10015,             | -1,       |
|                     | 输出 | -              | -                                                                                                                                                                                                                |          | -1 (无意义)           | -1 (无意义), |
| HLVLERR             | 输入 | -              | -                                                                                                                                                                                                                |          | -1 (无意义),          | -1 (无意义), |
|                     | 输出 | 高 Level 报警输出信号 | 输出高 Level 报警发生中。                                                                                                                                                                                                 |          | 10016              | -1        |
| LLVLERR             | 输入 | -              | -                                                                                                                                                                                                                |          | -1 (无意义),          | -1 (无意义), |
|                     | 输出 | 低 Level 报警输出信号 | 输出低 Level 报警发生中。                                                                                                                                                                                                 |          | 10017              | -1        |
| CLVLERR             | 输入 | -              | -                                                                                                                                                                                                                |          | -1 (无意义),          | -1 (无意义), |
|                     | 输出 | 警告报警输出信号       | 输出警告报警发生中。                                                                                                                                                                                                       |          | 10018              | -1        |
| EMGERR              | 输入 | -              | -                                                                                                                                                                                                                |          | -1 (无意义),          | -1 (无意义), |
|                     | 输出 | 紧急停止输出信号       | 输出紧急停止为发生中。<br>[EMGERR 输出条件]<br>• 外部紧急停止错误: H0050、H0051 (冗余系统线路异常)<br>• 操作面板紧急停止错误: H0060、H0061 (冗余系统线路异常)<br>• 示教单元紧急停止错误: H0070、H0071 (冗余系统线路异常)<br>• 门开关错误: H0039、H0040 (冗余系统线路异常)<br>• EMGIN 连接器的配线异常: H0141 |          | 10019              | -1        |

| 参数名                              | 区分 | 名称                     | 功能                                                                                                                                                                                                      | 信号 Level | 工厂出货信号号码<br>输入, 输出                        |                                |
|----------------------------------|----|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------|--------------------------------|
|                                  |    |                        |                                                                                                                                                                                                         |          | CR7xx-Q                                   | CR7xx-D                        |
| SnSTART<br>(n=1 ~ 32)<br>(需要操作权) | 输入 | 插槽 n 起动输入              | 执行各插槽的起动。n=1 ~ 32                                                                                                                                                                                       | Edge     | -1,<br>-1                                 | -1,<br>-1                      |
|                                  | 输出 | 插槽 n 运行中输出             | 输出各插槽的运行中状态。n=1 ~ 32                                                                                                                                                                                    |          |                                           |                                |
| SnSTOP<br>(n=1 ~ 32)             | 输入 | 插槽 n 停止输入              | 执行各插槽的停止。n=1 ~ 32                                                                                                                                                                                       | Level    | -1,<br>-1                                 | -1,<br>-1                      |
|                                  | 输出 | 插槽 n 中断中输出             | 在各插槽为一时停止中, 输出程序中中断中。n=1 ~ 32                                                                                                                                                                           |          |                                           |                                |
| SnSRVOFF<br>(n=1 ~ 3)<br>(需要操作权) | 输入 | 机器 n 伺服关闭输入信号          | 将各机器伺服关闭 n=1 ~ 3<br>本信号为输入中不可开启伺服                                                                                                                                                                       | Level    | -1,<br>-1                                 | -1,<br>-1                      |
|                                  | 输出 | 机器 n 伺服开启不可输出信号        | 输出伺服开启不可状态。(回声)                                                                                                                                                                                         |          |                                           |                                |
| SnSRVON<br>(n=1 ~ 3)<br>(需要操作权)  | 输入 | 机器 n 伺服开启输入信号          | 将各机器伺服开启。n=1 ~ 3                                                                                                                                                                                        | Edge     | -1,<br>-1                                 | -1,<br>-1                      |
|                                  | 输出 | 机器 n 伺服开启中输出信号         | 输出伺服开启状态。n=1 ~ 3                                                                                                                                                                                        |          |                                           |                                |
| SnMELOCK<br>(n=1 ~ 3)<br>(需要操作权) | 输入 | 机器 n 机械锁输入信号           | 将各机器设定 / 解除在机械锁状态<br>n=1 ~ 3                                                                                                                                                                            | Edge     | -1,<br>-1                                 | -1,<br>-1                      |
|                                  | 输出 | 机器 n 机械锁中输出信号          | 输出机械锁状态<br>n=1 ~ 3                                                                                                                                                                                      |          |                                           |                                |
| PRGSEL<br>(需要操作权)                | 输入 | 程序选择输入信号               | 将以数值输入信号的设定值以程序号码指定。选择对应插槽 1 的程序号码。数值输入 (IODATA) 的输出维持 30ms 以上请开始输出。此外本信号也是 30ms 以上请输出到机器人。                                                                                                             | Edge     | 10020                                     | -1,                            |
|                                  | 输出 | -                      | -                                                                                                                                                                                                       |          |                                           |                                |
| OVRDSEL<br>(需要操作权)               | 输入 | 速度比例选择输入信号             | 将在数值输入信号的设定值视为速度比例。往数值输入 (IODATA) 的输出为请由 30ms 以上输出。且本信号 30ms 以上往机器人输出。                                                                                                                                  | Edge     | 10021                                     | -1,                            |
|                                  | 输出 | -                      | -                                                                                                                                                                                                       |          |                                           |                                |
| IODATA                           | 输入 | 数值输入<br>(开始位号码, 结束位号码) | 作为二进制读入<br>• 程序号码 (以 PRGSEL 读入、参数「PST」有效时在起动信号读入。<br>速度比例在 OVRDSEL 读入、位宽可以任意设定。但是, 超出已设定的位宽值输出时, 无法保证值。<br>往机器人的输入, 最少等待 30ms 以上输出后开始将 PRGSEL 等的设定信号输入。                                                 | Level    | 注 2)<br>10032<br>(开始位),<br>1047<br>(结束位), | 注 2)<br>-1 (开始位),<br>-1 (结束位), |
|                                  | 输出 | 数值输出<br>(开始位号码, 结束位号码) | 作为二进制输出<br>• 程序号码 (在 PRGSEL 输出)、<br>• 速度比例 (在 OVRDOUT 输出)、<br>• 行号码 (在 LINEOUT 输出)、<br>• 执行报警号码 (在 ERROUT 输出) 的输出<br>位元宽可以任意设定。但是, 超出已设定的位宽值输出时, 无法保证值。<br>请将程序号码 (PRGOUT) 信号等往机器人输入开始, 等待 30ms 以上开始读入。 |          |                                           |                                |
| PRGOUT                           | 输入 | 程序号码输出要求               | 将插槽 1 的程序号码以数值输出 (IODATA) 输出。<br>将本信号输出到机器人开始等待 30ms 以上将数值输出 (IODATA) 信号读入。                                                                                                                             | Edge     | 10022,                                    | -1,                            |
|                                  | 输出 | 程序号码输出中信号              | 在数值输出将「程序号码输出中」状态输出。                                                                                                                                                                                    |          |                                           |                                |

| 参数名                                                                          | 区分 | 名称                            | 功能                                                                                                                                                                                                                                                                                                                         | 信号 Level | 工厂出货信号号码<br>输入, 输出                                       |                                                          |
|------------------------------------------------------------------------------|----|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------------------------------------------------------|----------------------------------------------------------|
|                                                                              |    |                               |                                                                                                                                                                                                                                                                                                                            |          | CR7xx-Q                                                  | CR7xx-D                                                  |
| LINEOUT                                                                      | 输入 | 行号码输出要求                       | 将任务插槽 1 的行号码在数值输出 (IODATA) 输出。<br>将本信号输出到机器人开始等待 30ms 以上将数值输出 (IODATA) 信号读入。                                                                                                                                                                                                                                               | Edge     | 10023,                                                   | -1,                                                      |
|                                                                              | 输出 | 行号码输出中信号                      | 在数值输出将 [ 行号码输出中 ] 状态输出。                                                                                                                                                                                                                                                                                                    |          | 10023                                                    | -1                                                       |
| OVRDOUT                                                                      | 输入 | 速度比例值输出要求                     | 将 OP 速度通过数值 (IODATA) 输出。<br>将本信号输出到机器人开始等待 30ms 以上将数值输出 (IODATA) 信号读入。                                                                                                                                                                                                                                                     | Edge     | 10024,                                                   | -1,                                                      |
|                                                                              | 输出 | 速度比例值输出中信号                    | 在数值输出将 [ 速度比例输出中 ] 状态输出。                                                                                                                                                                                                                                                                                                   |          | 10024                                                    | -1                                                       |
| ERROUT                                                                       | 输入 | 报警号码输出要求                      | 将报警号码在数值输出 (IODATA) 输出。<br>将本信号输出到机器人开始等待 30ms 以上将数值输出 (IODATA) 信号读入。                                                                                                                                                                                                                                                      | Edge     | 10025,                                                   | -1                                                       |
|                                                                              | 输出 | 报警号码输出中信号                     | 输出数值输出 [ 报警号码输出中 ] 状态。                                                                                                                                                                                                                                                                                                     |          | 10025                                                    | -1                                                       |
| JOGENA<br>(需要操作权)                                                            | 输入 | JOG 有效输入信号                    | 将指定轴以被指定的模式执行 JOG 动作。<br>在此信号开启期间动作。                                                                                                                                                                                                                                                                                       | Level    | -1,                                                      | -1,                                                      |
|                                                                              | 输出 | JOG 有效中输出信号                   | 输出 JOG 动作状态。                                                                                                                                                                                                                                                                                                               |          | -1                                                       | -1                                                       |
| JOGM                                                                         | 输入 | JOG 模式输入<br>(开始号码, 结束号码)      | 指定 JOG 模式。<br>0/1/2/3/4= 关节, 直交, 圆筒, 3 轴直交, TOOL (EX-T)<br>注) 关于 EX-T 控制及 Ex-T JOG, 请参照第 652 页的「7.5 关于 Ex-T 控制」。                                                                                                                                                                                                           | Level    | 注 3)<br>-1 (开始位),<br>-1 (结束位),<br>-1 (开始位),<br>-1 (结束位), | 注 3)<br>-1 (开始位),<br>-1 (结束位),<br>-1 (开始位),<br>-1 (结束位), |
|                                                                              | 输出 | JOG 模式输出<br>(开始号码, 结束号码)      | 输出现在的 JOG 模式。                                                                                                                                                                                                                                                                                                              |          |                                                          |                                                          |
| JOGMENO<br>※ 仅在以下的 SW Ver 中有效。<br>S/W Ver.<br>F-Q 系列: R5 以上<br>F-D 系列: S5 以上 | 输入 | JOG 机械号码输入<br>(开始号码, 结束号码)    | 指定机械号码。<br>未设定本参数时, 机械 1 固定。                                                                                                                                                                                                                                                                                               | 等级       | -1 (开始位),<br>-1 (结束位),                                   | -1 (开始位),<br>-1 (结束位),                                   |
|                                                                              | 输出 | JOG 机械号码输出<br>(开始号码, 结束号码)    | 输出当前的机械号码。                                                                                                                                                                                                                                                                                                                 |          | -1 (开始位),<br>-1 (结束位),                                   | -1 (开始位),<br>-1 (结束位),                                   |
| JOG+                                                                         | 输入 | JOG 前进正号侧 8 轴<br>(开始号码, 结束号码) | 指定 JOG 动作的轴。<br>关节 JOG 时: 从开始号码 J1,J2,J3,J4,J5,J6,J7,J8 轴<br>直交 JOG 时: 从开始号码 X,Y,Z,A,B,C,L1,L2 轴<br>圆筒 JOG 时: 从开始号码 X,θ,Z,A,B,C,L1,L2 轴<br>三轴直交 JOG 时: 从开始号码 X,Y,Z,J4,J5,J6 轴<br>TOOL JOG 时: 从开始号码 X,Y,Z,A,B,C 轴<br>工件 JOG 时 (Ex-T JOG 时):<br>从开始号码 X,Y,Z,A,B,C 轴<br>注) 关于 EX-T 控制及 Ex-T JOG, 请参照第 652 页的「7.5 关于 Ex-T 控制」。 | Level    | 注 4)<br>-1,<br>-1                                        | 注 4)<br>-1,<br>-1                                        |
|                                                                              | 输出 | -                             | -                                                                                                                                                                                                                                                                                                                          |          |                                                          |                                                          |

| 参数名                   | 区分 | 名称                            | 功能                                                                                                                                                                                                                                                                                                                         | 信号 Level | 工厂出货信号号码<br>输入, 输出                                       |                                                          |
|-----------------------|----|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------------------------------------------------------|----------------------------------------------------------|
|                       |    |                               |                                                                                                                                                                                                                                                                                                                            |          | CR7xx-Q                                                  | CR7xx-D                                                  |
| JOG-                  | 输入 | JOG 前进负号侧 8 轴<br>(开始号码, 结束号码) | 指定 JOG 动作的轴。<br>关节 JOG 时: 从开始号码 J1,J2,J3,J4,J5,J6,J7,J8 轴<br>直交 JOG 时: 从开始号码 X,Y,Z,A,B,C,L1,L2 轴<br>圆筒 JOG 时: 从开始号码 X,θ,Z,A,B,C,L1,L2 轴<br>三轴直交 JOG 时: 从开始号码 X,Y,Z,J4,J5,J6 轴<br>TOOL JOG 时: 从开始号码 X,Y,Z,A,B,C 轴<br>工件 JOG 时 (Ex-T JOG 时):<br>从开始号码 X,Y,Z,A,B,C 轴<br>注) 关于 EX-T 控制及 Ex-T JOG, 请参照第 652 页的「7.5 关于 Ex-T 控制」。 | Level    | 注 4)<br>-1,<br>-1                                        | 注 4)<br>-1,<br>-1                                        |
|                       | 输出 | -                             | -                                                                                                                                                                                                                                                                                                                          |          |                                                          |                                                          |
| JOGWKND<br>(需要操作权)    | 输入 | 工件坐标编号                        | 在工件 JOG 动作中, 以 1 ~ 8 的数值指定作为基准的工件坐标号码 (Ex-T 坐标号码)。<br>注) 本输入信号根据 JOG 有效输入信号: JOGENA 的 Edge (从 OFF 到 ON 的变化) 进行获取。要变更工件坐标编号时, 请使 JOG 有效输入信号: JOGENA 暂时从 OFF 变为 ON<br>注) 关于 EX-T 控制及 Ex-T 坐标号码, 请参照第 652 页的「7.5 关于 Ex-T 控制」。                                                                                                 | Level    | 注 3)<br>-1 (起始位),<br>-1 (结束位),<br>-1 (起始位),<br>-1 (结束位), | 注 3)<br>-1 (起始位),<br>-1 (结束位),<br>-1 (起始位),<br>-1 (结束位), |
|                       | 输出 |                               | 输出所输入的工件坐标号码 (Ex-T 坐标号码)。<br>注) 关于 EX-T 控制及 Ex-T 坐标号码, 请参照第 652 页的「7.5 关于 Ex-T 控制」。                                                                                                                                                                                                                                        |          |                                                          |                                                          |
| JOGNER<br>(需要操作权)     | 输入 | JOG 时报警<br>一时忽视输入信号           | JOG 操作时将无法复位报警一时的忽视。<br>※ 本信号只以机器 1 为对象。                                                                                                                                                                                                                                                                                   | Level    | -1,                                                      | -1,                                                      |
|                       | 输出 | JOG 时报警<br>一时忽视输出中信号          | 输入报警一时忽视状态。<br>※ 本信号只以机器 1 为对象。                                                                                                                                                                                                                                                                                            |          | -1                                                       | -1                                                       |
| HNDCTRLn<br>(n=1 ~ 3) | 输入 | -                             | -                                                                                                                                                                                                                                                                                                                          |          |                                                          |                                                          |
|                       | 输出 | 机器 n 抓手输出信号<br>(开始号码, 结束号码)   | 将抓手输出 (n=1) 900~907 的状态输出<br>将抓手输出 (n=2) 910~917 的状态输出<br>将抓手输出 (n=3) 920~927 的状态输出<br>例) 想将 900~903 的 4 点在通用输出信号 3 号、4 号、5 号、6 号输出的情况下, 将 HNDCTRL1 设定在 (3,6)。                                                                                                                                                             |          | HNDCTRL1<br>10048<br>(起始位)<br>10055<br>(结束位)             | -1 (起始位),<br>-1 (结束位),                                   |
| HNDSTS n<br>(n=1 ~ 3) | 输入 | -                             | -                                                                                                                                                                                                                                                                                                                          |          |                                                          |                                                          |
|                       | 输出 | 机器 n 抓手输入信号<br>(开始号码, 结束号码)   | 将抓手输入 (n=1) 900~907 的状态输出<br>将抓手输入 (n=2) 910~917 的状态输出<br>将抓手输入 (n=3) 920~927 的状态输出<br>例) 想将 900~903 的 4 点在通用输出信号 3 号、4 号、5 号、6 号输出的情况下, 将 HNDSTS1 设定在 (3,6)。                                                                                                                                                              |          | HNDSTS1<br>10056<br>(起始位)<br>10063<br>(结束位)              | -1 (起始位),<br>-1 (结束位),                                   |

| 参数名                                    | 区分 | 名称                        | 功能                                                                                                                                                                                                                                                                                                                             | 信号 Level | 工厂出货信号号码<br>输入, 输出                        |                                |
|----------------------------------------|----|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------|--------------------------------|
|                                        |    |                           |                                                                                                                                                                                                                                                                                                                                |          | CR7xx-Q                                   | CR7xx-D                        |
| HANDENA                                | 输入 | 抓手控制允许输入                  | 允许或禁止通过外部信号控制机器人抓手。 1/0 = 允许 / 禁止<br><br>注) 自动运转中可进行机器人的抓手控制。为确保安全, 请切实做好机器人与 PLC 等外部机器的互锁。<br>允许通过外部信号进行机器人抓手控制时, 来自程序的“HOpen/HClose”变为无效。                                                                                                                                                                                    | Level    | -1,<br>-1                                 | -1,<br>-1                      |
|                                        | 输出 | 抓手控制允许输出                  | 输出通过外部信号控制机器人抓手的允许状态。 1/0 = 允许 / 禁止<br>抓手控制允许输入信号被开启时, T/B 无效的情况下本信号将开启。                                                                                                                                                                                                                                                       |          |                                           |                                |
| HANDOUT                                | 输入 | 抓手输出控制信号                  | 设定用于控制机器人抓手的外部输入信号范围。<br>此处所设定的输入信号与参数:<br>HANDTYPE 中设定的抓手信号按顺序一一对应。<br>要素 1: 抓手输出控制信号开始号码<br>要素 2: 抓手输出控制信号结束号码                                                                                                                                                                                                               | Edge     | -1,<br>-1                                 | -1,<br>-1                      |
|                                        | 输出 | -                         | -                                                                                                                                                                                                                                                                                                                              |          |                                           |                                |
| HNDERRn<br>(n=1 ~ 3)                   | 输入 | 机器 n 抓手警报输入信号             | 要求抓手报警发生。<br>LOW Level 报警 3 0 号发生。                                                                                                                                                                                                                                                                                             | Level    | -1,<br>-1                                 | -1,<br>-1                      |
|                                        | 输出 | 机器 n 抓手警报输出中信号            | 输出抓手报警发生中状态。                                                                                                                                                                                                                                                                                                                   |          |                                           |                                |
| AIRERRn<br>(n=1 ~ 5)                   | 输入 | 机器 n 空气压报警输入中信号           | 要求空压报警发生。<br>LOW Level 报警 3 1 号发生。                                                                                                                                                                                                                                                                                             | Level    | -1,<br>-1                                 | -1,<br>-1                      |
|                                        | 输出 | 机器 n 空气压报警输出中信号           | 输出空气压报警发生中状态。                                                                                                                                                                                                                                                                                                                  |          |                                           |                                |
| USRAREA<br><br>第 435 页的「5.8 关于使用者定义领域」 | 输入 | -                         | -                                                                                                                                                                                                                                                                                                                              |          | 注 5)<br>10064<br>(开始位),<br>10071<br>(结束位) | 注 5)<br>-1 (开始位),<br>-1 (结束位), |
|                                        | 输出 | 使用者定义领域 32 点 (开始号码, 结束号码) | 输出机器人为使用者定义领域内。<br>从开始号码侧以 1、2、3 的顺序被输出。<br>领域以参数 AREA1P1、AREA1P2 ~ AREA32P1、AREA32P2 设定。<br>设定例)<br>以 USRAREA 为例的话<br>只使用在领域 1 的情况 USRAREA:<br>8,8 → 设定有效<br>只使用在领域 1,2 的情况 USRAREA:<br>8,9 → 设定有效<br><br>USRAREA: -1,-1 → 设定无效<br>USRAREA: 8,-1 → 设定无效 (无报警)<br>USRAREA: -1,8 → 设定无效 (无报警)<br>USRAREA: 9,8 → 设定无效 (L6643 报警) |          |                                           |                                |
| MnPTEXC<br>(n=1 ~ 3)                   | 输入 | -                         | -                                                                                                                                                                                                                                                                                                                              |          | -1 (无意义),                                 | -1 (无意义),                      |
|                                        | 输出 | 维修品的更换时间警告                | 输出到达维修品的更换时间。                                                                                                                                                                                                                                                                                                                  | Level    | -1                                        | -1                             |
| MnWUPENA<br>(n=1 ~ 3)<br>(需要操作权)       | 输入 | 机器 n 暖机运行模式有效输入信号         | 使各机器的暖机运行模式有效。<br>(n=1 ~ 3)<br><br>注) 在本信号切换暖机运行模式的有效 / 无效, 必须要预先依据参数 (WUPENA 等) 使暖机运行模式为有效。依据参数使暖机运行模式为无效的情况下, 即使输入本信号也不会变成有效。                                                                                                                                                                                                | Level    | -1,                                       | -1,                            |
|                                        | 输出 | 机器 n 暖机运行模式输出信号           | 输出暖机运行模式变成有效。<br>(n=1 ~ 3)                                                                                                                                                                                                                                                                                                     |          | -1                                        | -1                             |

| 参数名                  | 区分 | 名称              | 功能                                                                                                                                                                                                                                       | 信号 Level     | 工厂出货信号号码<br>输入, 输出                                                          |           |
|----------------------|----|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------|-----------|
|                      |    |                 |                                                                                                                                                                                                                                          |              | CR7xx-Q                                                                     | CR7xx-D   |
| MnWUPMD<br>(n=1 ~ 3) | 输入 | -               | -                                                                                                                                                                                                                                        |              | -1 (无意义),                                                                   | -1 (无意义), |
|                      | 输出 | 机器 n 暖机运行状态输出信号 | 输出在以暖机运行状态将速度调降的动作。<br>(n=1 ~ 3)                                                                                                                                                                                                         |              | -1                                                                          | -1        |
| PSSL0T               | 输入 | 位置数据输出插槽号码指定    | 指定位置数据的输出对象插槽号码 (1 ~ 32)。此处所指定的插槽中事先加载有对象程序。<br>* 如果未在外部机器侧将参数: PSOUT 的输入信号 (位置数据输出指示) 关闭, 则插槽号码不会变更。                                                                                                                                    | Level        | -1 (输入起始位),<br>-1 (输入结束位),<br>-1 (输出起始位),<br>-1 (输出结束位)<br>* 输入输出均为最大 6 位宽  |           |
|                      | 输出 | 位置数据输出插槽号码输出    | 输出 (响应) 现在位置数据的输出对象插槽号码。                                                                                                                                                                                                                 |              |                                                                             |           |
| PSTYPE               | 输入 | 位置数据类型指定        | 指定作为位置数据的输出对象的数据的类型。<br>1/0= 关节型变量 / 位置型变量<br>* 如果未在外部机器侧将参数: PSOUT 的输入信号 (位置数据输出指示) 关闭, 则位置数据类型不会变更。                                                                                                                                    | Level        | -1, -1                                                                      |           |
|                      | 输出 | 位置数据类型输出        | 输出 (响应) 现在位置数据的输出对象数据类型。<br>1/0= 关节型变量 / 位置型变量                                                                                                                                                                                           |              |                                                                             |           |
| PSNUM                | 输入 | 位置数据号码指定        | 指定作为输出对象的位置数据的号码 (关节型变量 / 位置型变量的号码)。<br>位置数据的号码范围: 0 ~ 65535<br>例) 将 P100 作为输出对象时, 在输入起始位~输入结束位间指定数值 "100"。<br>(注) 不可指定像 "P001" 之类在高位中含有 "0" 的变量。<br>* 如未在外部设备侧将 PSOUT 输入信号 (位置数据输出指示) 关闭, 则位置数据号码不会变更。                                  | Level        | -1 (输入起始位),<br>-1 (输入结束位),<br>-1 (输出起始位),<br>-1 (输出结束位)<br>* 输入输出均为最大 16 位宽 |           |
|                      | 输出 | 位置数据号码输出        | 输出 (响应) 现在输出对象的位置数据的号码。<br>输出在输入侧指定的位置号码。                                                                                                                                                                                                |              |                                                                             |           |
| PSOUT                | 输入 | 位置数据输出指示        | 是要求输出出现在输出对象位置数据的指示。<br>该信号从 OFF 变为 ON 时 (Edge), 将现在输出对象位置数据向参数: PSPOS (位置数据输出位领域) 中指定的信号号码输出, 在该信号持续 ON 期间 (Level), 保持该输出状态。(不进行 realtime 更新)<br>(Edge: 位置数据的输出更新、<br>Level: 保持输出值)<br>0 → 1: 位置数据的输出更新<br>1: 保持输出状态<br>0: 可以变更输出对象位置数据 | Edge + Level | -1, -1                                                                      |           |
|                      | 输出 | 位置数据输出中         | 对正在输出指定的位置数据一事进行输出。<br>【输出信息】<br>OFF: 位置数据未输出<br>ON : 位置数据输出中                                                                                                                                                                            |              |                                                                             |           |

| 参数名      | 区分 | 名称            | 功能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 信号 Level | 工厂出货信号号码<br>输入, 输出        |            |
|----------|----|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------------------|------------|
|          |    |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          | CR7xx-Q                   | CR7xx-D    |
| PSPOS    | 输入 | -             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -        | -1, (无意义)                 |            |
|          | 输出 | 位置数据输出信号领域    | <p>指定输出位置数据的信号领域的起始号码。<br/>领域为 8 个轴的坐标值 (32 位 x 8 轴) 和构造标志 (32 位 x 2 要素), 需要共计 320 位的连续领域。可设定的范围如下所示。</p> <p>(1)CR7xx-Q 系列<br/>10000 ~ 17872: 多 CPU 间共有设备</p> <p>(2)CR7xx-D 系列<br/>2000 ~ 3632: PROFIBUS<br/>6000 ~ 7728: CC-Link</p> <p>输出的位置数据的排列<br/>位置型变量:<br/>X,Y,Z,A,B,C,L1,L2,FL1,FL2<br/>关节型变量: J1,J2,J3,J4,J5,J6,J7,J8<br/>各坐标值以 1000 倍的整数输出。<br/>(mm, 度) 在外部机器侧请按 1/1000 处理。但是, FL1、FL2 将直接输出原值。</p> <p>可输出的数据范围<br/>有符号整数 -2147483648 ~ 2147483647<br/>请参阅第 540 页的「(5) 外部信号操作时序图 (其 5)」中所示的时序图和注意事项。</p> | -        | -1                        |            |
| TMPOUT   | 输入 | 柜内温度输出请求      | 将机器人控制器的柜内温度输出到数值输出 (IODATA) 中。<br>温度以二进制整数形式输出。<br>请在向机器人输入本信号后等待 15ms 以上再读取数值输出 (IODATA) 信号。                                                                                                                                                                                                                                                                                                                                                                                                                      | 边缘       | -1, -1                    |            |
|          | 输出 | 柜内温度输出中       | 将“正在输出柜内温度”状态输出到数值输出中。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -        |                           |            |
| RSTBAT   | 输入 | 电池累计时间的复位     | 复位电池累计时间。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -        | -1、                       |            |
|          | 输出 | 复位结束          | 输出复位结束。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -        | -1                        |            |
| RSTGRS   | 输入 | 维护预报的复位 (润滑脂) | 复位维护预报的润滑脂信息。<br>※ 轴位模式指定为 IODATA 或 DIODATA。                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -        | -1、                       |            |
|          | 输出 | 复位结束          | 输出复位结束。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -        | -1                        |            |
| RSTBLT   | 输入 | 维护预报的复位 (皮带)  | 复位维护预报的皮带信息。<br>※ 轴位模式指定为 IODATA 或 DIODATA。                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -        | -1、                       |            |
|          | 输出 | 复位结束          | 输出复位结束。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -        | -1                        |            |
| SVDATA   | 输入 | -             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -        | -1 (无意义)                  | -1、<br>-1、 |
|          | 输出 | 负载率数据         | 从本参数的第 3 要素中指定的寄存器号码起, 将 J1 到 J8 轴的最大负载率 (%) 输出到第 4 要素所指定的号码中。<br>输出的负载率的更新周期为间隔 2 秒。<br>(支持 CC-Link 寄存器)                                                                                                                                                                                                                                                                                                                                                                                                           | -        | -1 (起始寄存器),<br>-1 (结束寄存器) |            |
| DOORSTS1 | 输入 | -             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -        | -1、                       | -1、        |
|          | 输出 | 门开关 1 状态      | 输出门开关第 1 系统的状态。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -        | -1                        | -1         |
| DOORSTS2 | 输入 | -             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -        | -1、                       | -1、        |
|          | 输出 | 门开关 2 状态      | 输出门开关第 2 系统的状态。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -        | -1                        | -1         |

| 参数名     | 区分 | 名称    | 功能                                                      | 信号 Level | 工厂出货信号号码<br>输入, 输出 |         |
|---------|----|-------|---------------------------------------------------------|----------|--------------------|---------|
|         |    |       |                                                         |          | CR7xx-Q            | CR7xx-D |
| DOORSTS | 输入 | -     | -                                                       | -        | -1,                | -1,     |
|         | 输出 | 门开关状态 | 输出门开关第 1 系统与第 2 系统的逻辑或。<br>仅在第 1、2 系统均为 ON 时, 该信号也为 ON。 | -        | -1                 | -1      |

注 1) 信号 Level 的意义如下所示。

**Level:** → 在信号为 ON 状态时, 指定的功能为有效, 且在 OFF 状态时变为无效。信号的开启时间请最低确保在 15ms 以上。

**Edge:** → 信号为 OFF 状态开始变为 ON 状态, 在变化时指定的功能为有效, 且之后信号返回到 OFF 状态, 指定的功能会维持在原来的状态。



注 2) 输入开始号码、输入结束号码、输出开始号码、输出结束号码的顺序。作为实际的数值输入或输出使用时, 以 2 进制表示, 使用从开始号码到结束号码。表现开始号码为最下位位、结束号码为最上位位。为表现数值请只设定必要的数。

例如在程序选择使用, 程序为只到 1 ~ 6 的情况下, 设定 3 位的话, 可以表现。最大可设定到 16 位。

下列为分配例。

例) 将起动信号设定在在通用输入 10016, 运行中信号设定在在通用输出 10026 的情况。

参数 START= {10016, 10026}

例) 将数值输入设定在 10027 ~ 10030, 4 位量、数值输出设定在 10027 ~ 10031, 5 位量的情况。

参数 IODATA= {10027, 10030, 10027, 10031}

注 3) 输入开始号码、输入结束号码、输出开始号码、输出结束号码的顺序。

作为实际的 JOG 模式使用时, 从开始号码到结束号码以 2 进制表示。开始号码为最下位的位置、结束号码为最上位的位置。为表现数值请只设定必要的数。

例如只使用关节模式和直交模式的情况下, 如果设定 2 位话最好, 最大可设定到 3 位。

注 4) 输入开始号码、输入结束号码的顺序。开始号码指定为 J1/X 轴、结束号码指定为最大 J8/L2 轴。

例如使用 6 轴机器人的情况下, 如果设定 6 位最好。

即使 4 轴机器人的情况, 使用直交模式的时候, 为了使 C 轴为必要, 请务必设定为 6 位。最大可设定到 8 位。

注 5) 输出开始号码、输出结束号码的顺序。指始号码设定为区域 1、结束号码指定为最大区域 32。

例如只使用 2 个区域的情况下, 如果设定 2 位话最好, 最大可设定到 32 位。

注 6) 上述参数中可设定的值的范围为 0 ~ 255、2000 ~ 3951、6000 ~ 8047、10000 ~ 18191。

## 6.4 信号的有效无效状态

依据输入信号的种类，运行中及停止输入中等，依据那时的机器人的状态会有即使输入作为目的的信号也没有功能的情况。

表 6-7 为对应输入信号的有效 / 无效，机器人的状态的关系。

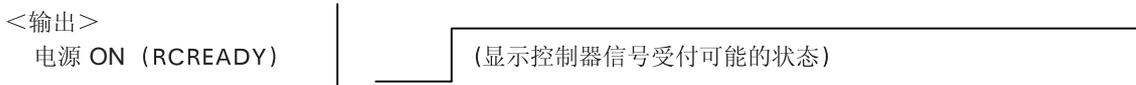
表 6-7：专用输入信号的有效 / 无效状态

| 参数名                             | 名称         | 依据机器人的状态左边标记信号的有效 / 无效              |                                     |
|---------------------------------|------------|-------------------------------------|-------------------------------------|
| SLOTINIT                        | 程序复位       | 运行中状态（START 输出 ON 状态）的时候没有功能。       |                                     |
| SAFEPOS                         | 待避点回归输入    |                                     |                                     |
| OUTRESET                        | 通用输出信号复位   |                                     |                                     |
| MnWUPENA                        | 暖机运行模式有效输入 |                                     |                                     |
| START<br>SnSTART<br>(n=1 ~ 32)  | 起动机输入      | 只在外部输入的操作权有效（IOENA 输出 ON 状态）的时候有功能。 |                                     |
| SLOTINIT                        | 程序复位       |                                     |                                     |
| SRVON<br>MnSRVON<br>(n=1 ~ 3)   | 伺服 ON 输入   |                                     |                                     |
| MELOCK<br>MnMELOCK<br>(n=1 ~ 3) | 机械锁输入      |                                     |                                     |
| SAFEPOS                         | 待避点回归输入    |                                     |                                     |
| PRGSEL                          | 程序选择输入     |                                     |                                     |
| OVRDSEL                         | 速度比例选择输入   |                                     |                                     |
| JOGENA                          | JOG 有效输入   |                                     |                                     |
| MnWUPENA                        | 暖机运行模式有效输入 |                                     |                                     |
| START                           | 起动机输入      |                                     | 在停止输入状态（STOPSTS 为 ON 状态）的时候没有功能。    |
| SAFEPOS                         | 待避点回归输入    |                                     |                                     |
| JOGENA                          | JOG 有效输入   |                                     |                                     |
| SRVON                           | 伺服 ON 输入   |                                     | 伺服关闭输入状态的时候没有功能。                    |
| MELOCK<br>MnMELOCK<br>(n=1 ~ 3) | 机械锁输入      |                                     | 只在程序选择可状态（SLOTINIT 输出 ON 状态）的时候有功能。 |
| PRGSEL                          | 程序选择输入     | 在中断中状态（STOP 输出 ON 状态）的时候没有功能。       |                                     |

### 6.5 外部信号的时序图

#### 6.5.1 各信号的个别时序图

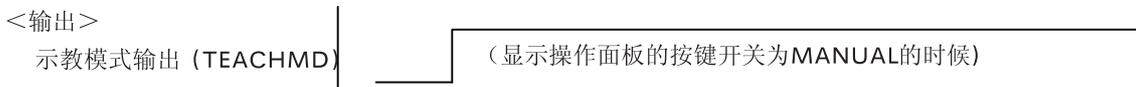
(1) RCREADY (控制器的电源 ON 完成输出)



(2) ATEXTMD (远程模式输出)



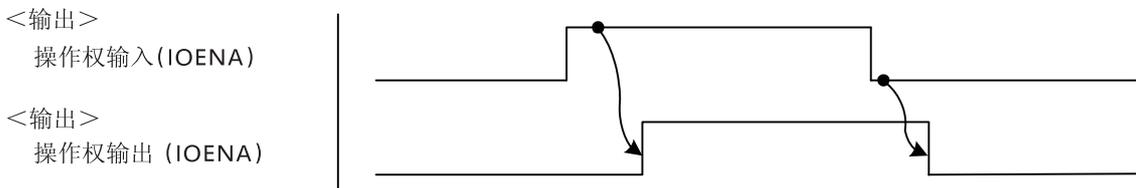
(3) TEACHMD (示教模式输出)



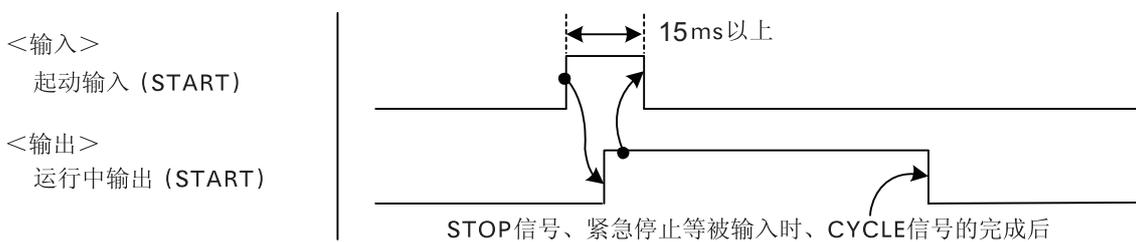
(4) ATTOPMD (自动模式输出)



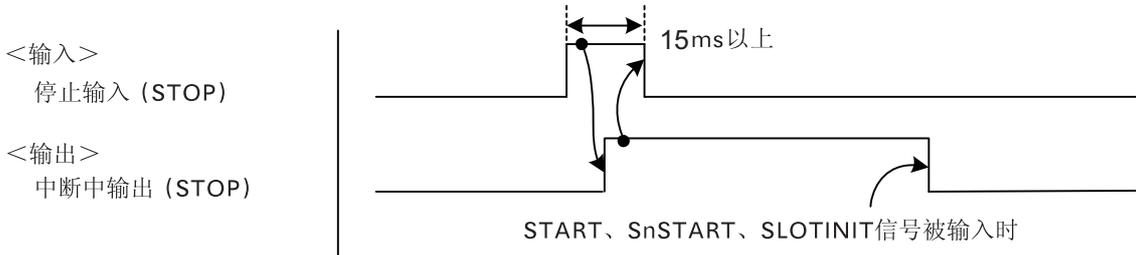
(5) IOENA (操作权输入信号 / 操作权输出信号)



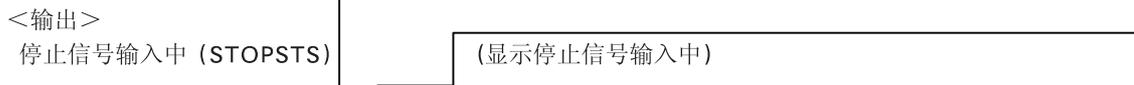
(6) START (起动输入 / 运行中输出)



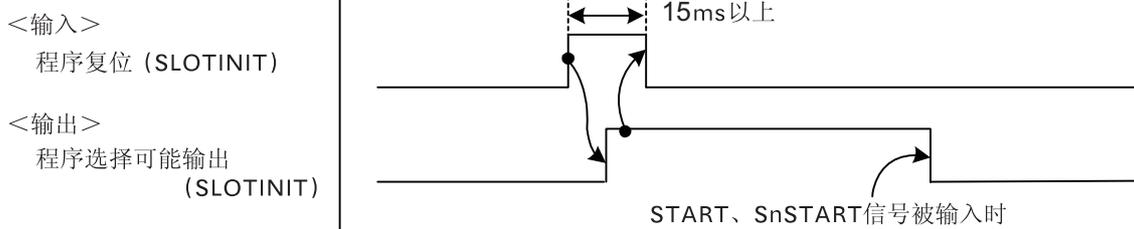
(7) STOP (停止输入 / 中断中输出)



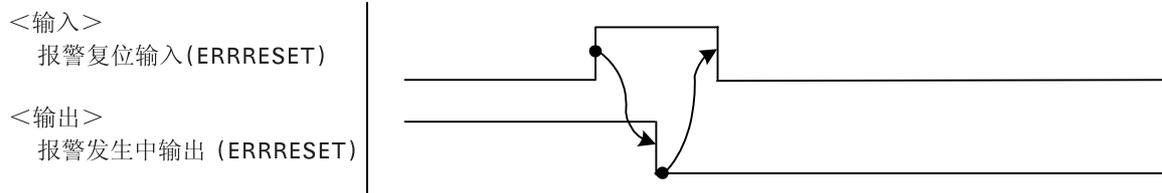
## (8) STOPSTS (停止信号输入中输出)



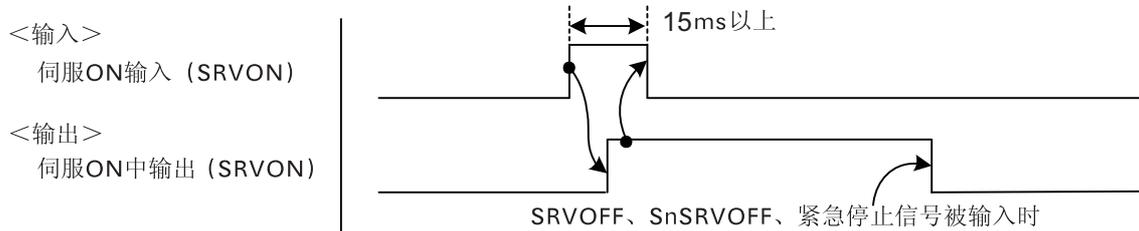
## (9) SLOTINIT (程序复位输入 / 程序选择可能输出)



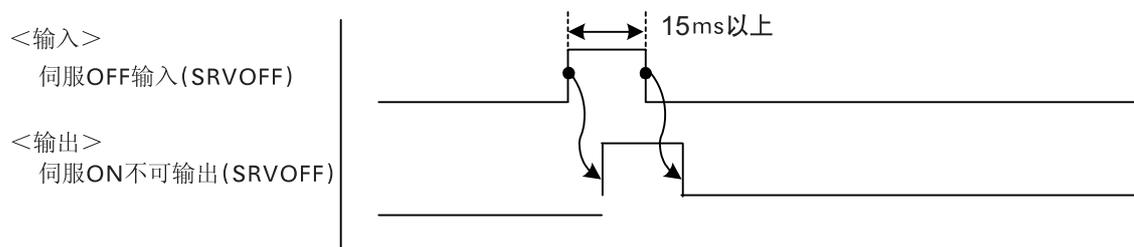
## (10) ERRRESET (报警复位输入 / 报警发生中输出)



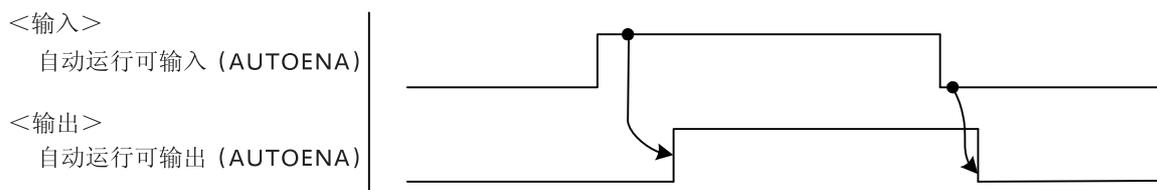
## (11) SRVON (伺服 ON 输入 / 伺服 ON 中输出)



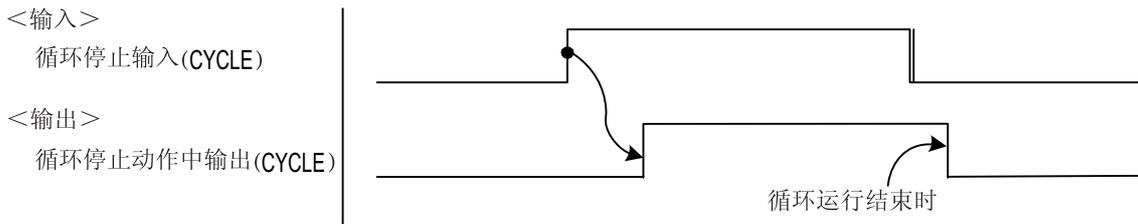
## (12) SRVOFF (伺服 OFF 输入 / 伺服 ON 不可输出)



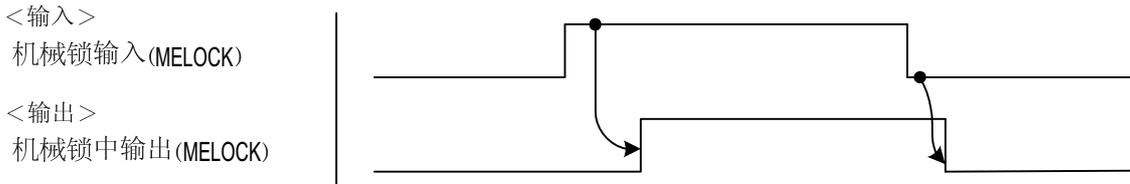
## (13) AUTOENA (自动运行输入 / 自动运行可输出)



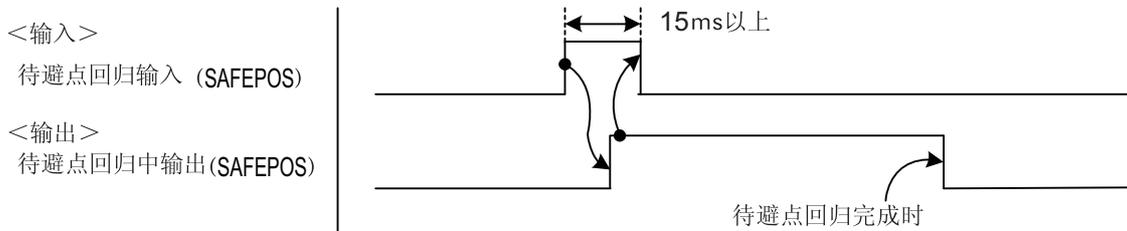
(14) CYCLE (循环停止输入 / 循环停止动作中输出)



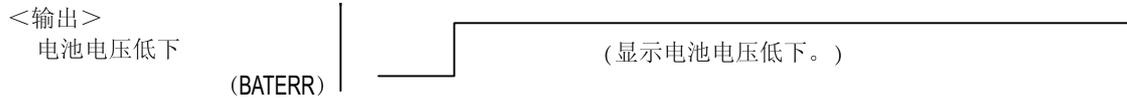
(15) MELOCK (机械锁输入 / 机械锁中输出)



(16) SAFEPOS (待避点回归输入 / 待避点回归中输出)



(17) BATERR (电池电压低下)



(18) OUTRESET (通用输出信号复位要求输入)



(19) HVLERR (高 Level 报警发生中输出)



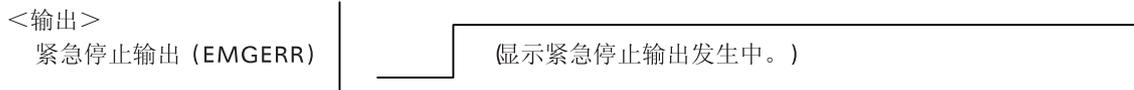
(20) LLVLERR (低 Level 报警发生中输出)



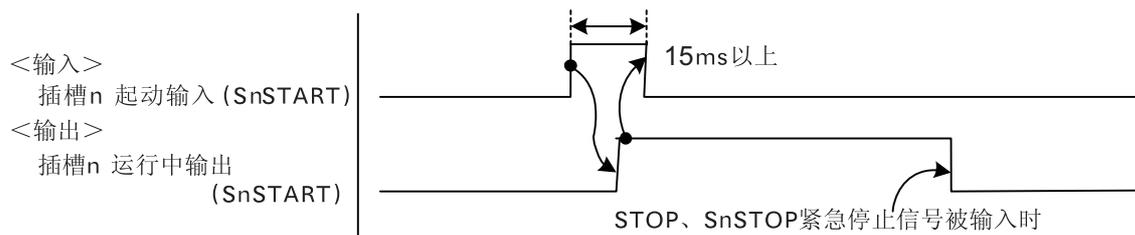
## (21) CLVLERR (警告 Level 报警发生输出中)



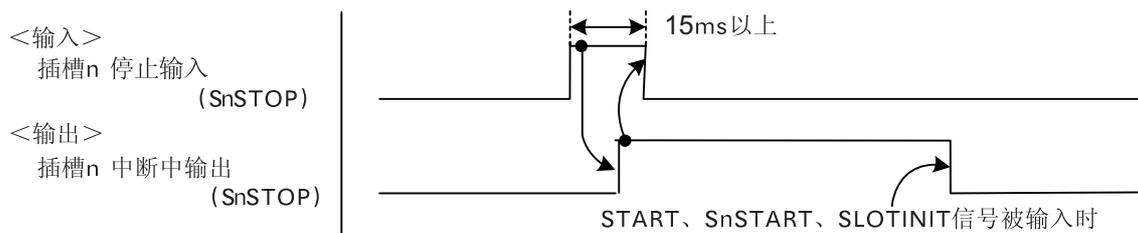
## (22) EMGERR (紧急停止中输出)



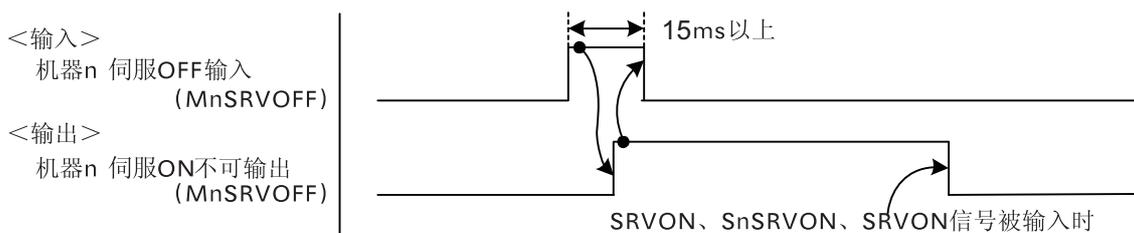
## (23) SnSTART (插槽 n 起动输入 / 插槽 n 运行中输出)



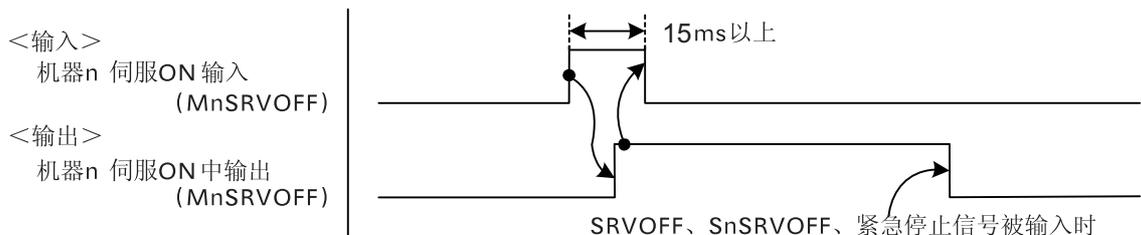
## (24) SnSTOP (插槽 n 停止输入 / 插槽 n 中断中输出)



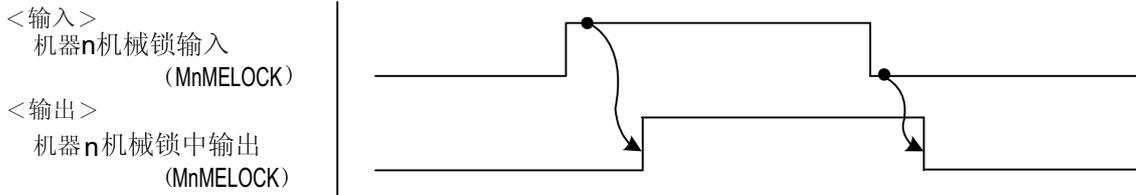
## (25) MnSRVOFF (机器 n 伺服 OFF 输入 / 机器 n 伺服 ON 不可输出)



## (26) MnSRVON (机器 n 伺服 ON 输入 / 机器 n 伺服 ON 中输出)

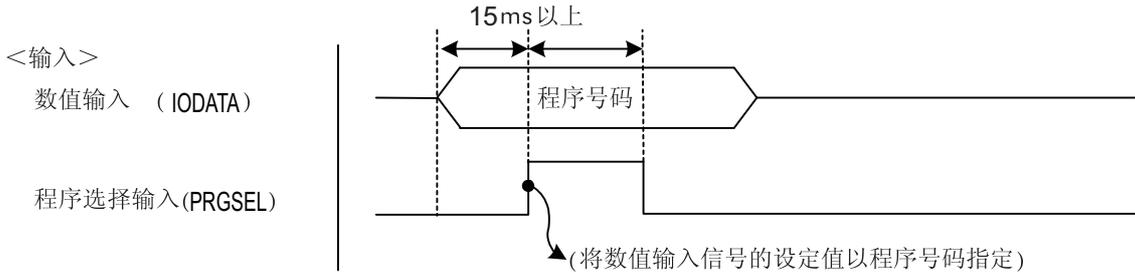


(27) MnMELOCK (机器 n 机械锁输入 / 机器 n 机械锁中输出)



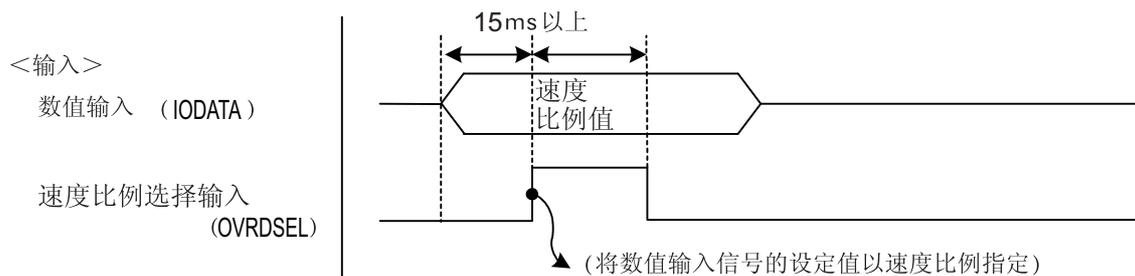
(28) PRGSEL (程序选择输入)

※ 和数值输出 (IODATA) 一起使用。



(29) OVRDSEL (速度比例选择输入)

※ 和数值输出 (IODATA) 一起使用。

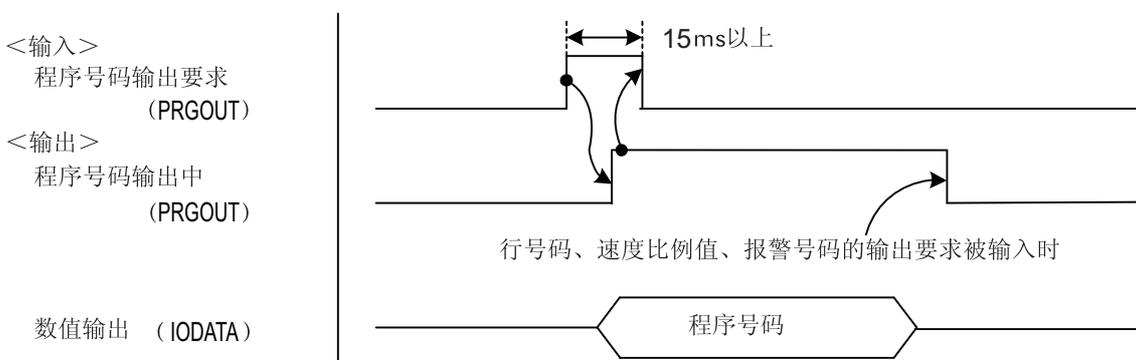


(30) IODATA (数值输入 / 数值输出)

※ 和 PRGSEL、OVRDSEL、PRGOUT、LINEOUT、OVRDOUT、ERROUT 一起使用。

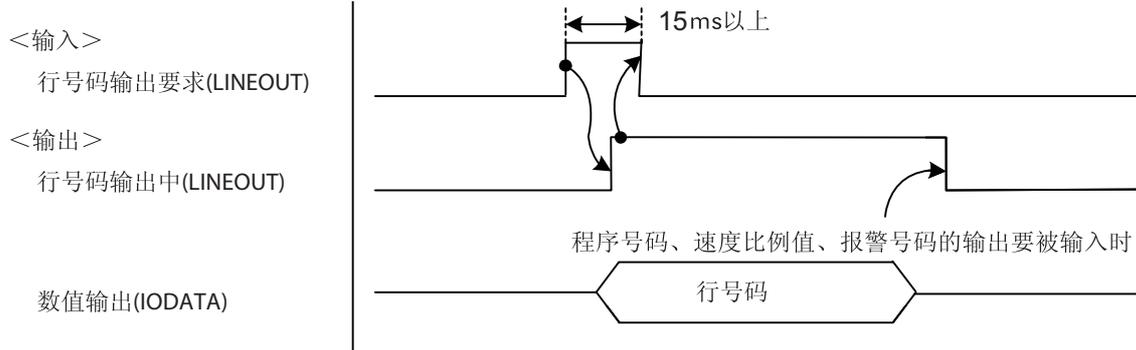
(31) PRGOUT (程序号码输出要求输入 / 程序号码输出中)

※ 和数值输出 (IODATA) 一起使用。



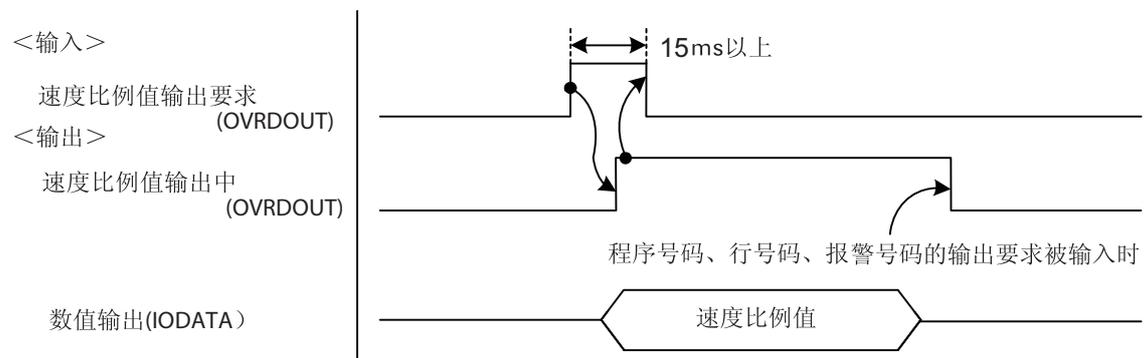
## (32) LINEOUT (行号码输出要求输入 / 行号码输出中)

\* 和数值输出 (IODATA) 一起使用



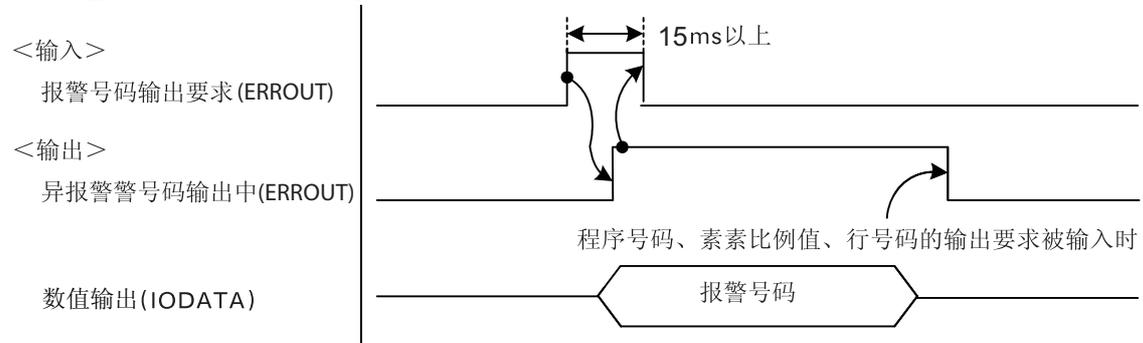
## (33) OVRDOUT (速度比例值输出要求 / 速度比例值输出中)

\* 和数值输出 (IODATA) 一起使用

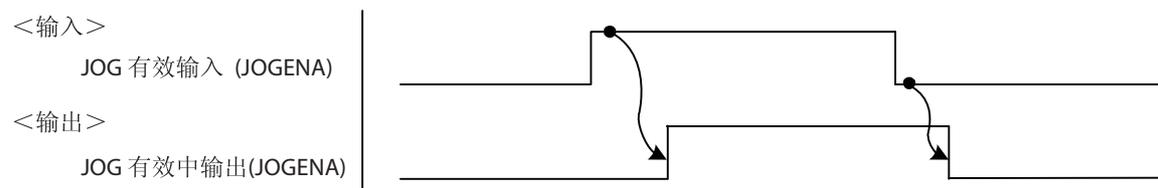


## (34) ERROUT (报警号码输出要求 / 报警号码输出中)

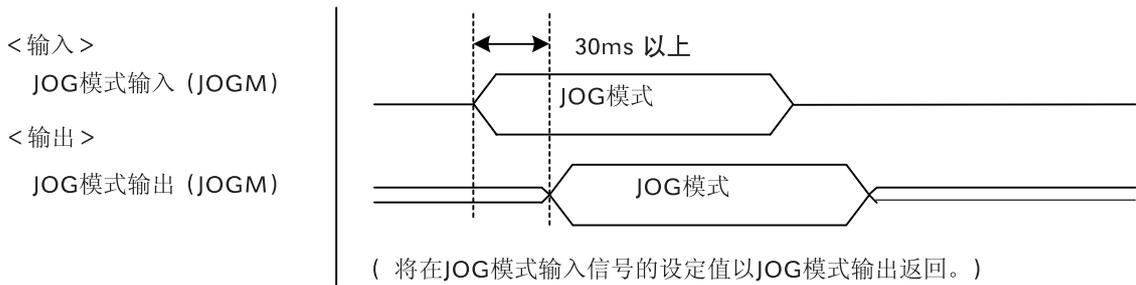
\* 和数值输出 (IODATA) 一起使用



## (35) JOGENA (JOG 有效输入 / JOG 有效中输出)



(36) JOGM (JOG 模式输入 /JOG 模式输出)



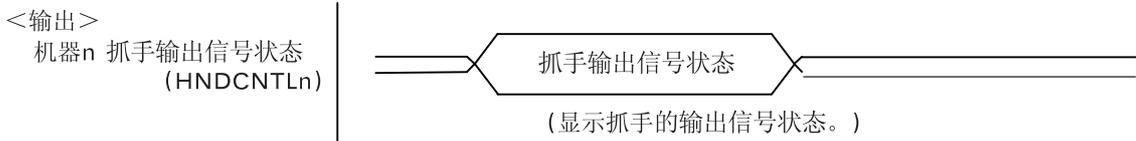
(37) JOG+ (在 JOG 前进正号侧输入 8 轴)



(38) JOG- (在 JOG 前进负号侧输入 8 轴)



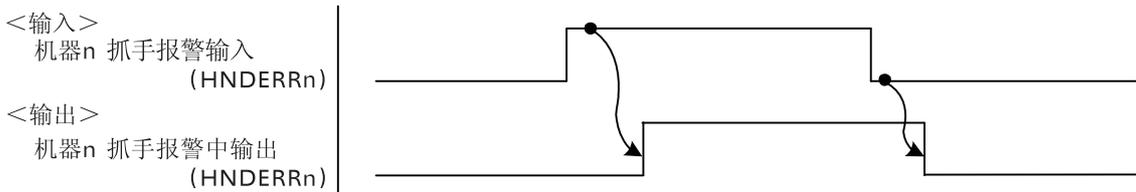
(39) HNDCTRLn (机器 n 抓手输出信号状态)



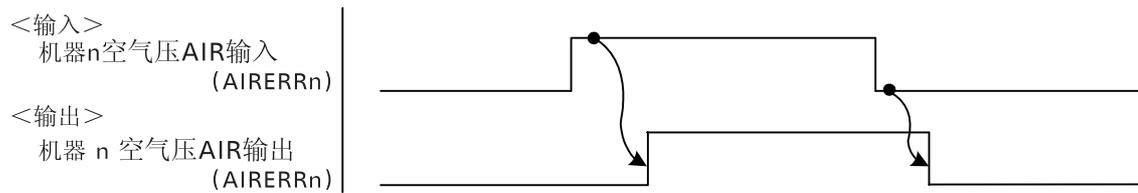
(40) HNDSTS n (机器 n 抓手输入信号状态)



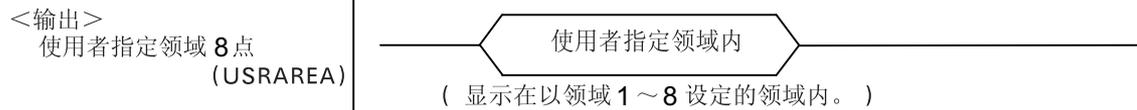
(41) HNDERRn (机器 n 抓手报警输入信号 / 机器 n 抓手报警中输出)



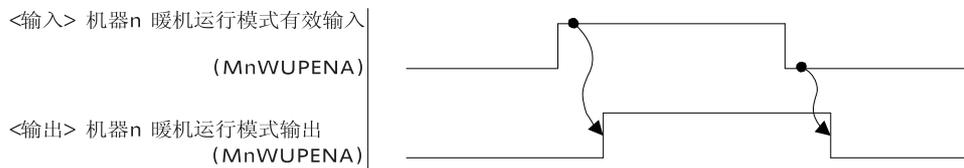
## (42) AIRERRn (机器 n 空气压 AIR 输入信号 / 机器 n 空气压 AIR 输出中)



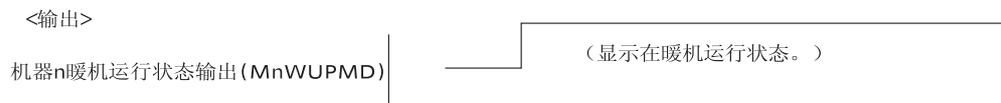
## (43) USRAREA (使用者定义领域 8 点输出)



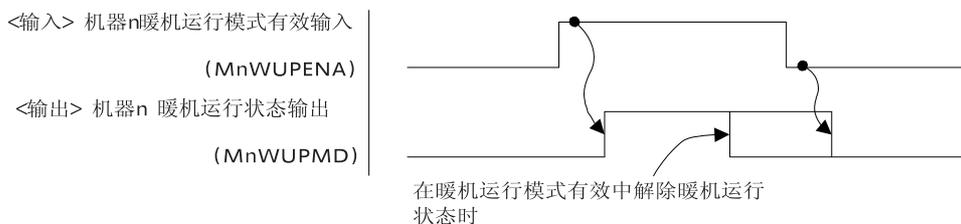
## (44) MnWUPENA (机器 n 暖机运行模式有效输入 / 机器 n 暖机运行模式输出)



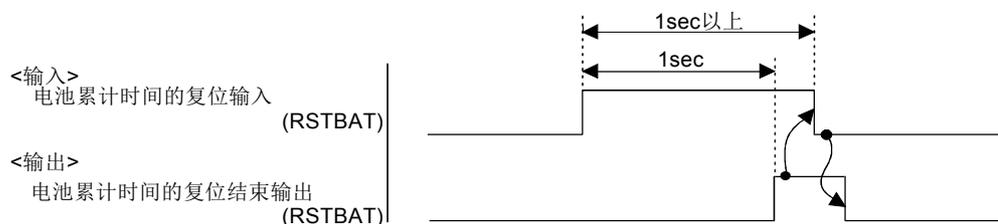
## (45) MnWUPMD (机器 n 暖机运行状态输出)



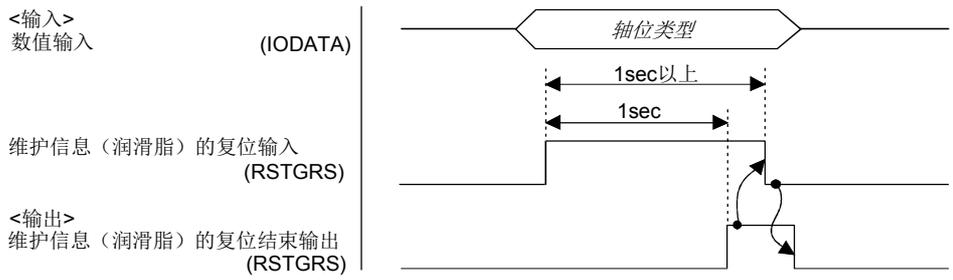
※ 机器 n 和暖机运行模式有效输入 (MnWUPENA) 同持分配时如下所示。



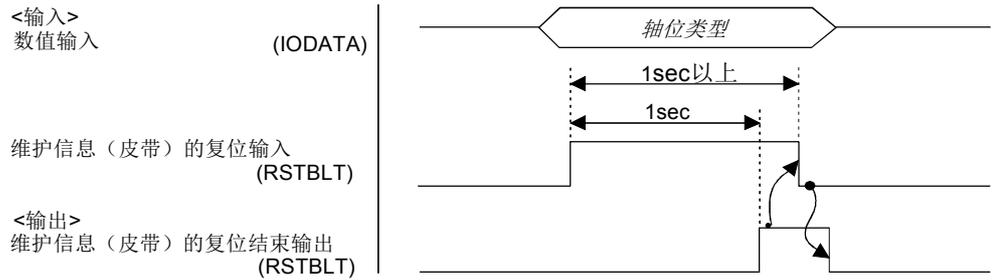
## (46) RSTBAT (电池累计时间的复位)



(47) RSTGRS (维护预报的复位 (润滑脂))



(48) RSTBAT (维护预报的复位 (皮带))



## 6.5.2 时序图例

## (1) 外部信号操作时序图（其1）

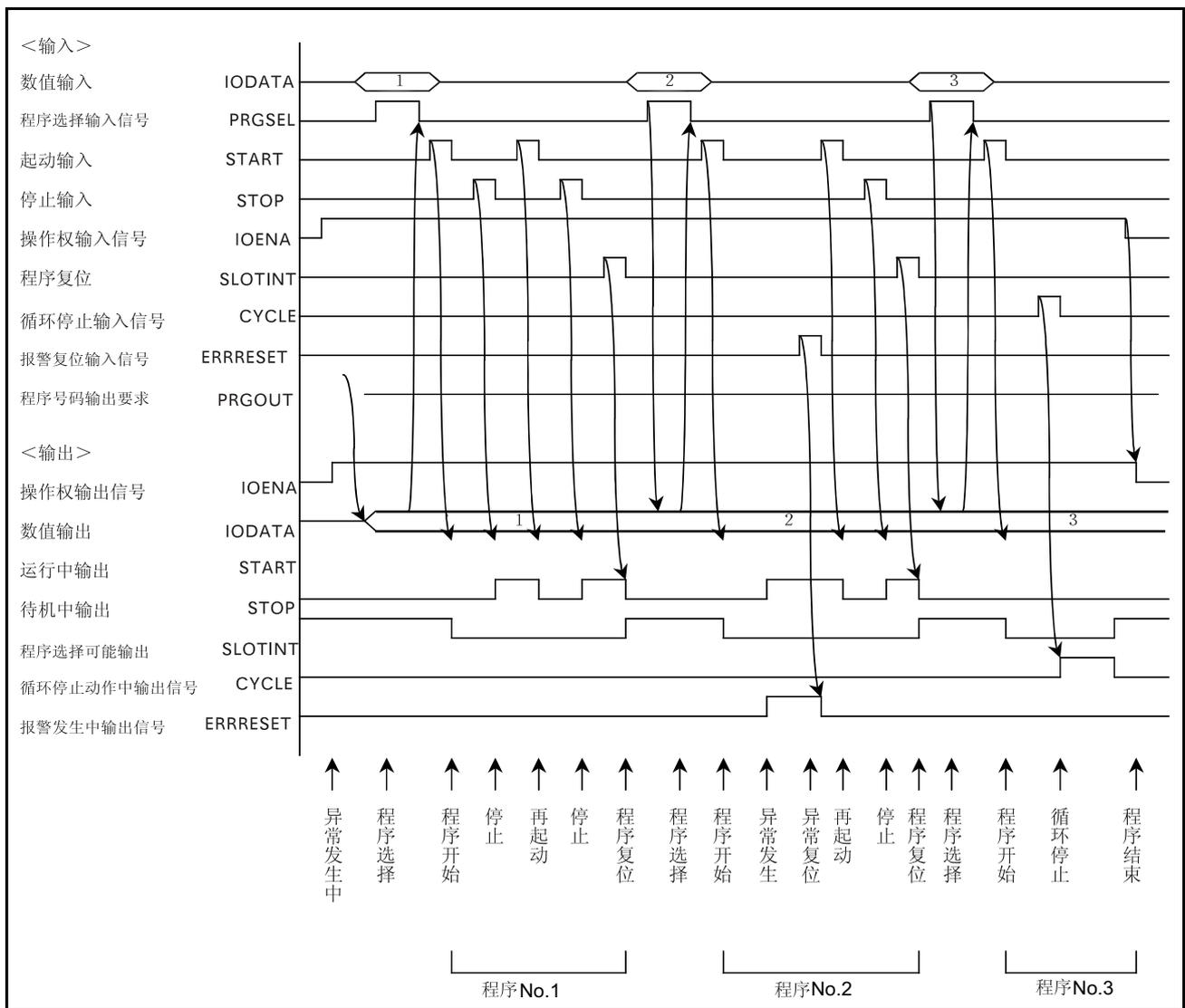


图 6-4: 外部操作时序图例（其1）

(2) 外部信号操作时序图（其2）

在图 6-5 外部信号，显示 Servo ON/OFF、程序选择、速度比例选择、起动、行号码输出等的时序图范例。

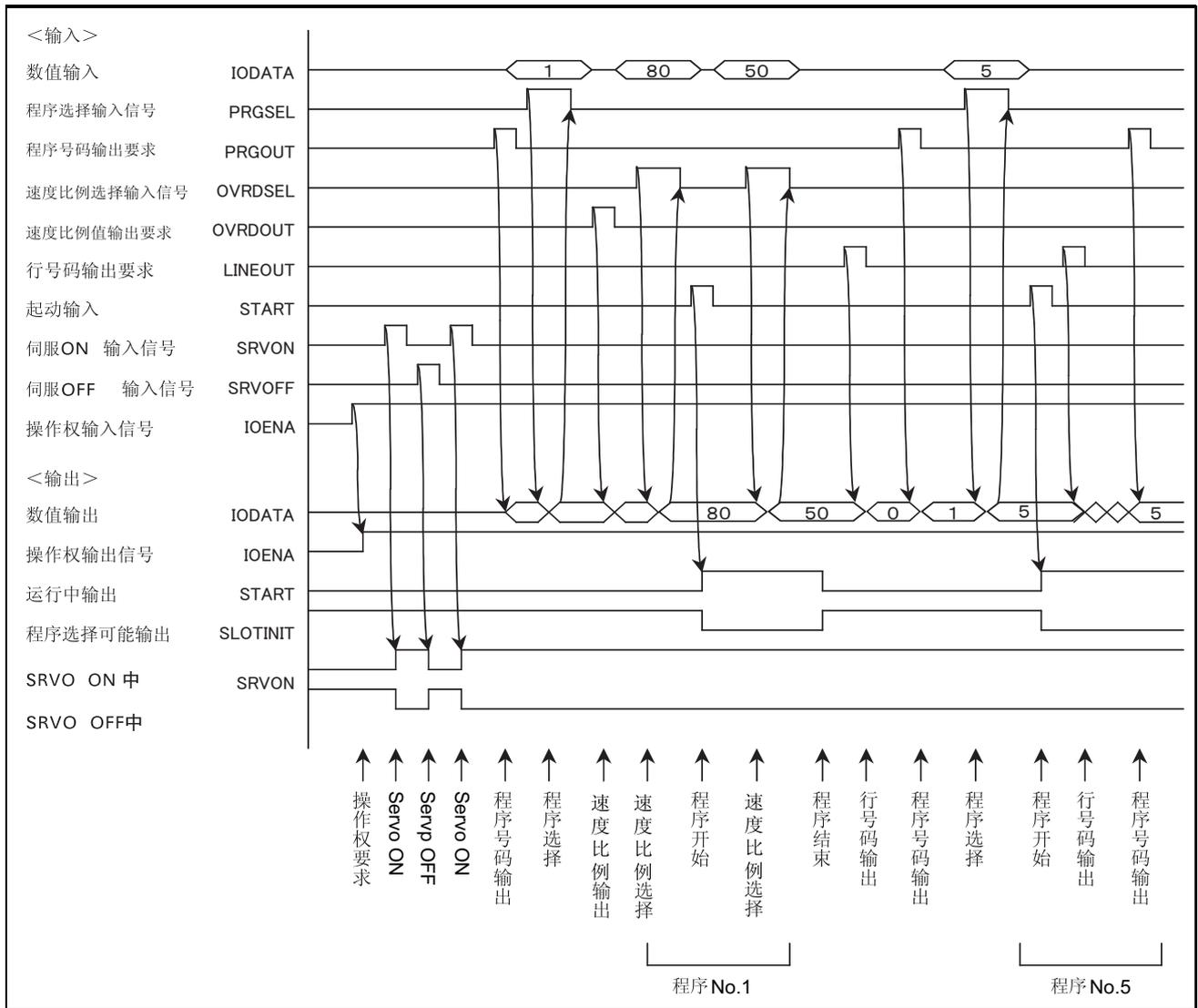


图 6-5: 外部信号操作时序图例（其2）

## (3) 外部信号操作时序图（其3）

在图 6-6 里显示来自外部信号的报警复位、通用输出复位、程序复位等的时序图例。

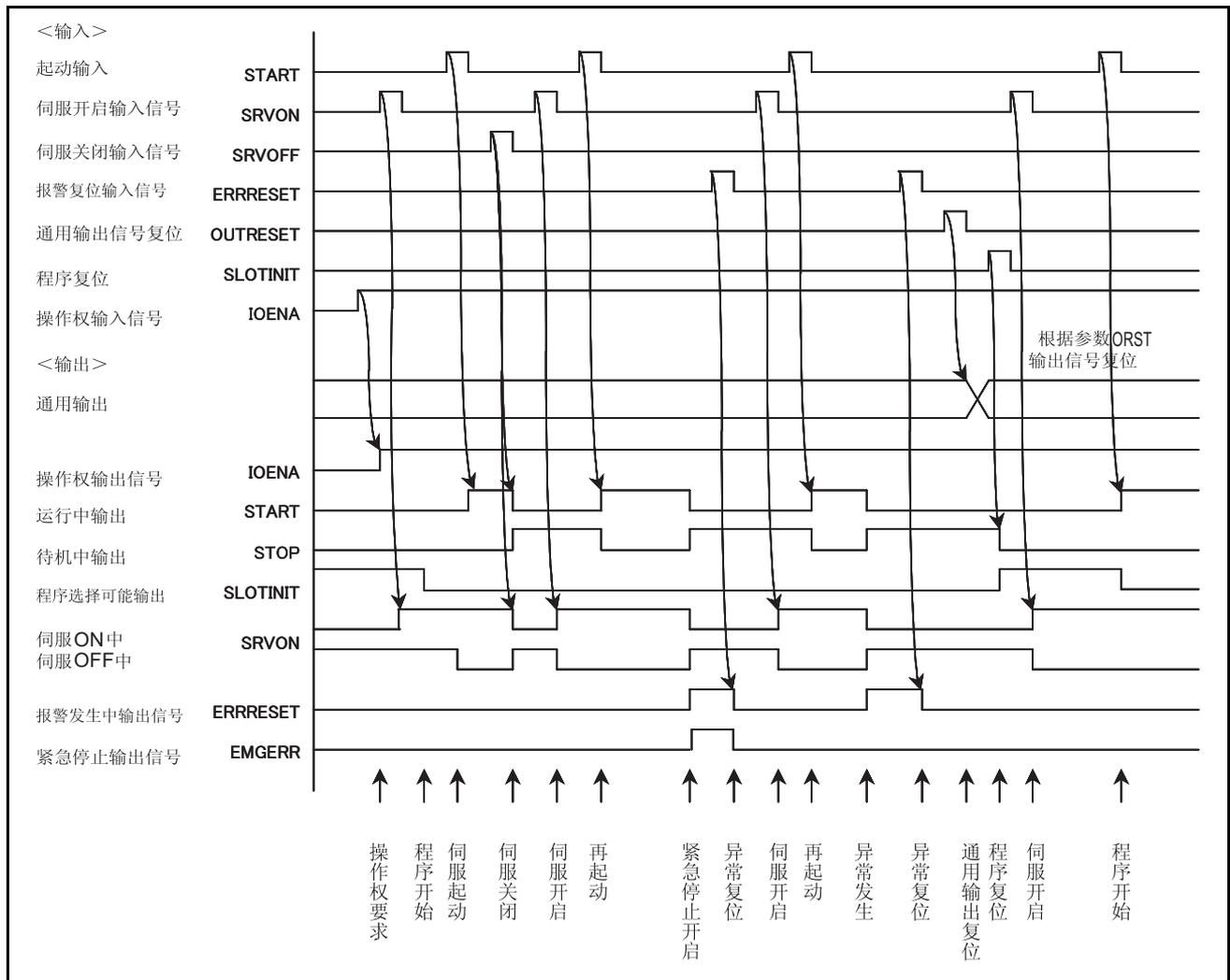


图 6-6: 外部信号操作时序图（其3）

(4) 外部信号操作时序图 (其 4)

在图 6-7 里显示来自外部信号的 JOG 操作、待避点回归、程序重置等的时序图例

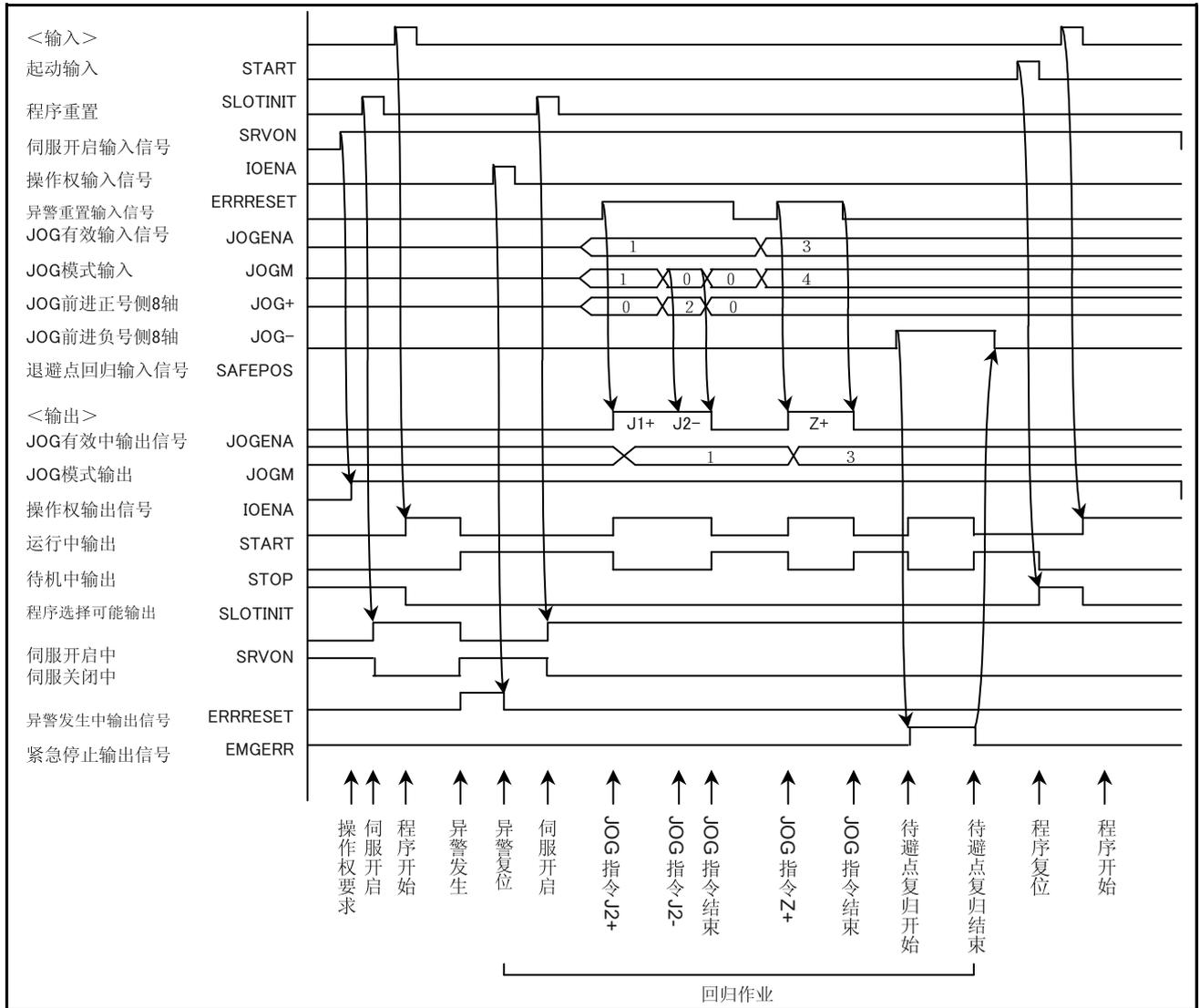


图 6-7: 外部信号操作时序图 (其 4)

## (5) 外部信号操作时序图（其 5）

下面是输出图 6-8 中指定的位置数据时的时序图示例。

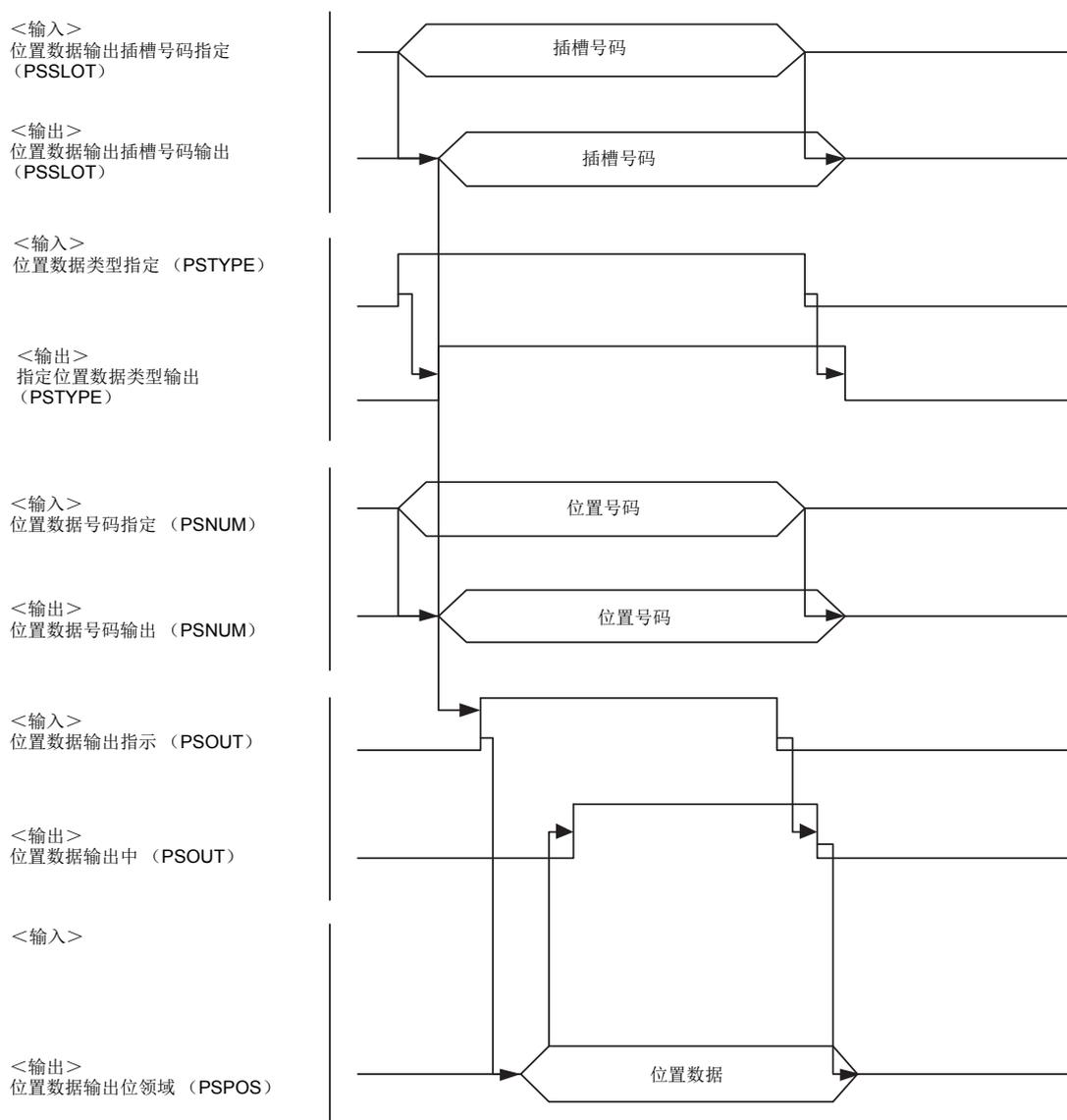


图 6-8: 外部信号操作时序图（其 5）

## [注意]

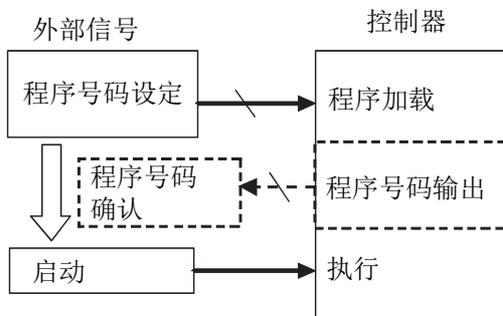
- (1) 参数「PSPOS」中指定的信号号码开始的 320 点的输出信号领域不存在时，将发生报警：7081（参数的值超出范围，因此无法写入），无法执行位置数据的监视。请务必确保 320 点的连续领域。
- (2) 参数「PSSLOT」中设定的信号号码的范围大于 6 位时，将发生报警：7081（参数的值超出范围，因此无法写入）。
- (3) 参数「PSNUM」中设定的位置数据号码的范围大于 16 位时，将发生报警：7081（参数的值超出范围，因此无法写入）。
- (4) 位置数据输出指示（PSOUT）输入中时，即使变更了插槽号码指定、位置数据类型指定、位置数据号码指定，其指定也将无效。此时，请将位置数据输出指示（PSOUT）的输入先关闭后再开启。请确认插槽号码输出（PSSLOT）、位置数据类型输出（PSTYPE）、位置数据号码输出（PSNUM）来了解输出的是哪个位置数据。
- (5) 指定的插槽中未加载程序时，各轴的值将输出为"&H7FFFFFFF"。
- (6) 指定的位置数据不存在时，各轴的值将输出为"&H80000000"。
- (7) 在位置数据输出中时，如果指定插槽中正在执行的程序发生了切换（CallP 指令或 Xrun 指令、参数「PRGSEL」），各轴的值将输出为"&H80000000"。

### 6.6 通过外部信号选择·执行程序的方法

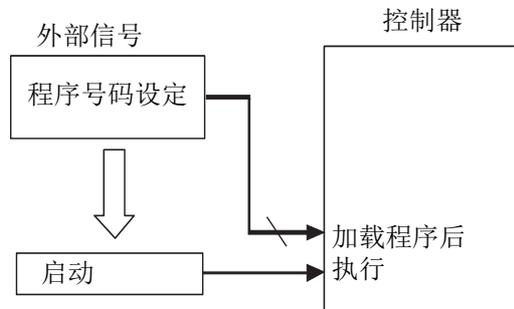
#### 6.6.1 种类

通过外部信号选择并执行程序的方法有以下 2 种。

- (a) 先确认指定的程序已加载后再开始执行的方法
- (b) 与启动信号同时指定程序号码后再开始执行的方法



(a)：确认指定程序的加载



(b)：与启动信号同时指定程序

#### 6.6.2 执行方法的选择

在设定表 6-8 的参数后指定通过外部信号执行程序的方法（上述 (a) 或 (b)）。

表 6-8：程序选择方法切换参数

| 参数名 | 功能概要                                                      |
|-----|-----------------------------------------------------------|
| PST | 切换程序选择方法。<br>0：先确认指定的程序已加载后再开始执行<br>1：与启动信号同时指定程序号码后再开始执行 |

#### 6.6.3 相关的输入输出参数

表 6-9：输入输出信号参数

| 参数名    | 区分      | 功能概要                                  |
|--------|---------|---------------------------------------|
| IOENA  | 输入      | 切换外部信号控制的操作有效 / 无效                    |
| PRGOUT | 输入      | 使任务插槽 1 的程序号码输出到数值输出 (IODATA) 中。      |
| IODATA | 输入 / 输出 | 设定数值输入 / 数值输出。设定为可以以二进制值读取所设定的范围的位状态。 |
| PRGSEL | 输入      | 将数值输入信号中的设定值指定为程序号码。                  |
| START  | 输入      | 启动程序。                                 |

## 6.6.4 操作步骤

作为举例，此处将输入输出信号按表 6-10 所示进行分配。

表 6-10: 输入输出信号分配

| 参数名    | 输入位    | 输出位    |
|--------|--------|--------|
| IOENA  | 5      | 3      |
| PRGOUT | 7      | -      |
| IODATA | 8 ~ 11 | 8 ~ 11 |
| PRGSEL | 6      | -      |
| START  | 3      | -      |

(1) 先确认指定的程序已加载后再开始执行时

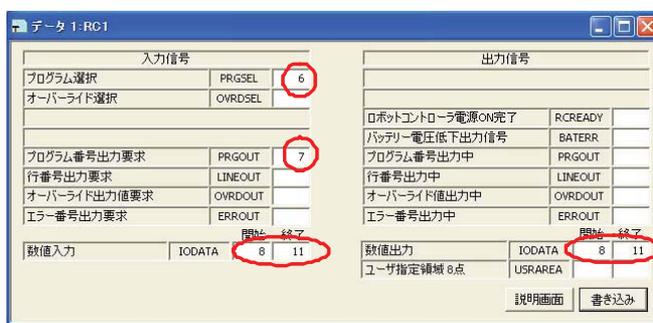
- 1) 将参数「PST」的设定值为 0。(初始设定值)
- 2) 将表 6-10 的输入输出位分配给参数。

下图为 ToolBox2 的【参数设定】→【专用输入输出分配】的设定画面。对图中圆圈标记部位进行设定。

【专用输入输出分配】：【通用 1】画面

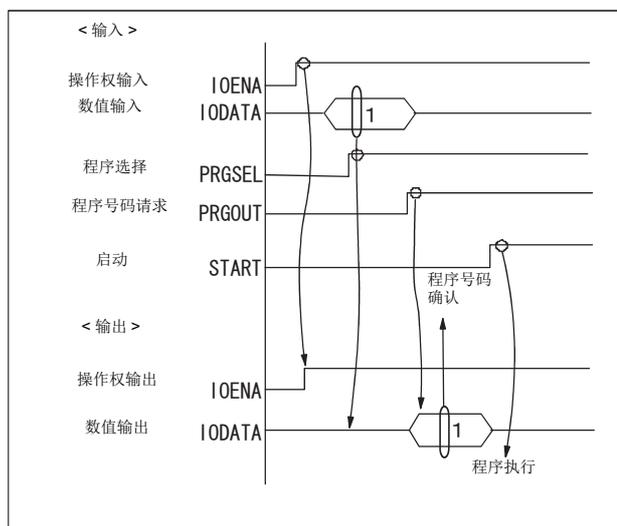


【专用输入输出分配】：【数据】画面



3) 选择・启动程序的步骤

- ① 开启输入信号 IOENA (位 5)，取得操作权。
- ② 确认输出信号 IOENA (位 3)，如果为 ON，则说明取得了操作权。
- ③ 在输入用 IODATA (位 8 ~ 11) 中将程序号码设定为二进制值。
- ④ 开启 PRGSEL (位 6)。  
将读取输入用 IODATA (位 8 ~ 11) 中设定的号码的程序。
- ⑤ 开启 PRGOUT (位 7)。  
所读取的程序号码被输出到输出用 IODATA (位 8 ~ 11) 中。可以确认号码是否一致。
- ⑥ 输入 START (位 3)，执行程序。



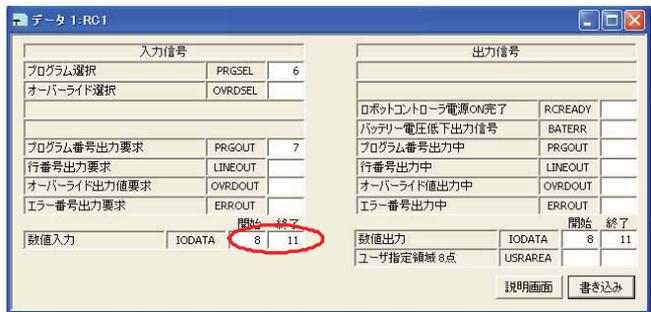
- (2) 与启动信号同时指定程序号码后再开始执行时
  - 1) 将参数「PST」的设定值为 1。(初始值为 0)
  - 2) 将表 6-10 的输入输出位分配给参数。

下图为 ToolBox2 的【参数设定】→【专用输入输出分配】的设定画面。对图中圆圈标记部位进行设定。

【专用输入输出分配】：【通用 1】画面

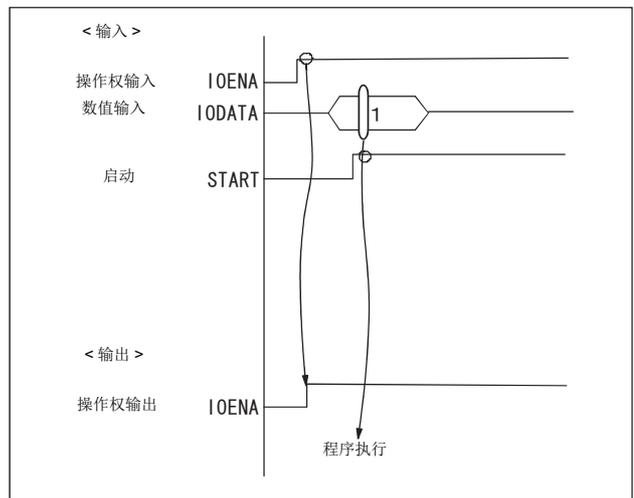


【专用输入输出分配】：【数据】画面



3) 选择 · 启动程序的步骤

- ① 开启输入信号 IOENA (位 5)，取得操作权。
- ② 确认输出信号 IOENA (位 3)，如果为 ON，则说明取得了操作权。
- ③ 在输入用 IODATA (位 8 ~ 11) 中将程序号码设定为二进制值。
- ④ 输入 START (位 3)，执行程序。



注 1) 变更了参数后，请重新接通控制器的电源。

注 2) 执行程序时，省略了伺服 ON、程序初始化等步骤。请参阅本书第 516 页的「6.3 专用输入输出」～第 526 页的「6.4 信号的有效无效状态」。

## 6.7 紧急停止输入

紧急停止输入配线等的相关数据请参照别册的「控制器设定和基本操作、保养」编。

### 6.7.1 紧急停止输入时的机器人的动作

在机器人动作中，输入紧停止信号的话，会将伺服电源硬性遮断。无法特别指定紧急停止后的机器人尖端的轨迹及停止位置。依据机器人的速度、**TOOL** 的负载状态，会发生速度比例。

## 7 附录

### 7.1 关于构造标志

构造标志是指机器人的姿势。

机器人抓手的前端会记忆 X, Y, Z, A, B, C 位置数据。但是,即使是相同的位置数据,但机器人可执行数种不同姿势。此姿势以构造标志表示。且位置藉由 FL1 记忆在位置常数 (X, Y, Z, A, B, C) (FL1、FL2) 内。以下显示构造标志的种类。

#### ■垂直多关型机器人的情况

##### (1) RIGHT/LEFT

表示 J1 轴的旋转中心垂直上方的法兰中心的位置 (R)。(5 轴型)

表示 J1 轴的旋转中心垂直上方的 J5 轴的旋转中心位置 (P)。(6 轴型)

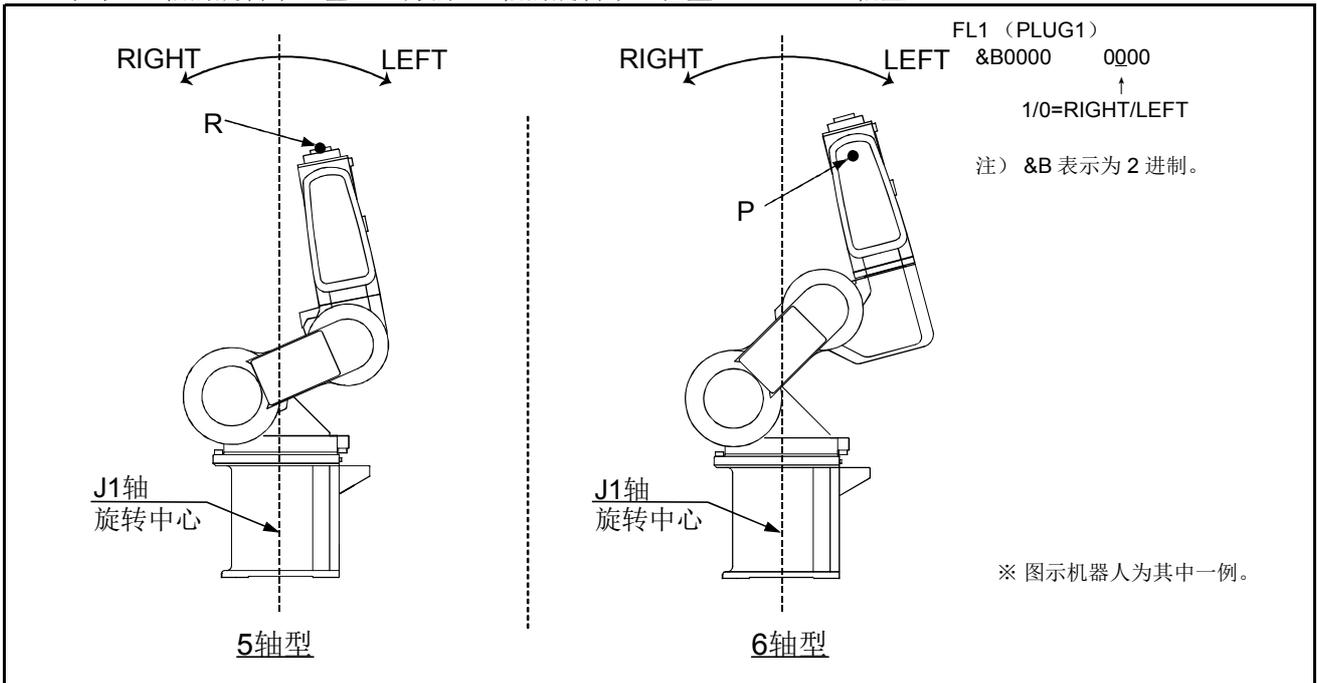


图 7-1: 构造 PLUG (RIGHT/LEFT)

##### (2) ABOVE/BELOW

表示通过从 J2 轴的旋转中心到 J3 轴的旋转中心的 J5 轴的旋转中心位置 (P)。

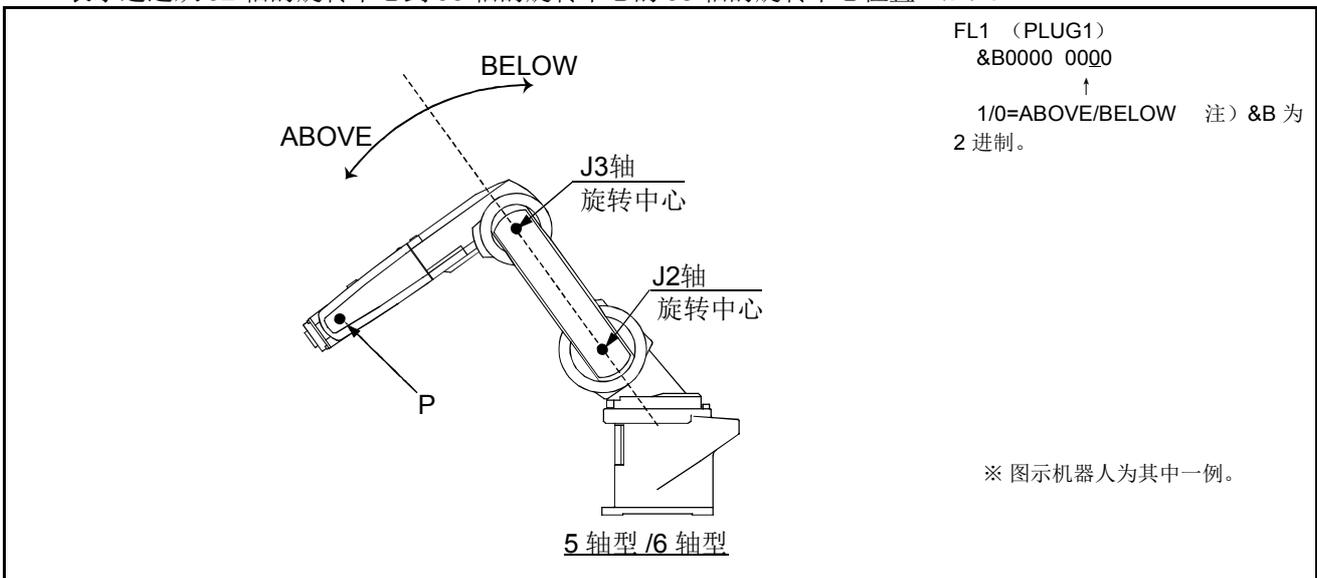


图 7-2: 构造 PLUG (ABOVE/BELOW)

## (3) NONFLIP/FLIP (只有 6 轴机型)

表示通过从 J4 轴的旋转中心到 J5 轴的旋转中心的法兰面的方向。

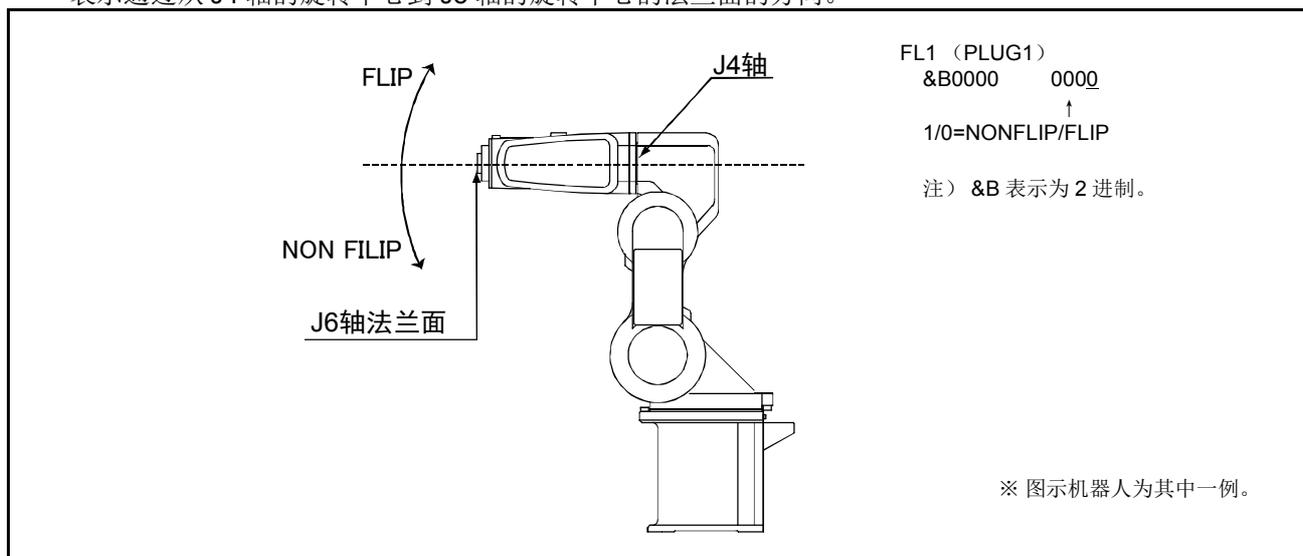


图 7-3: 构造 PLUG (NONFLIP/FLIP)

■ 水平多关节型机器人的情况

## (1) RIGHT/LEFT

表示通过从 J1 轴的旋转中心到 J2 轴的旋转中心的直线上相对的尖端轴位置。



图 7-4: 构造 PLUG (RIGHT/LEFT)

## 7.2 多功能电动抓手的使用方法

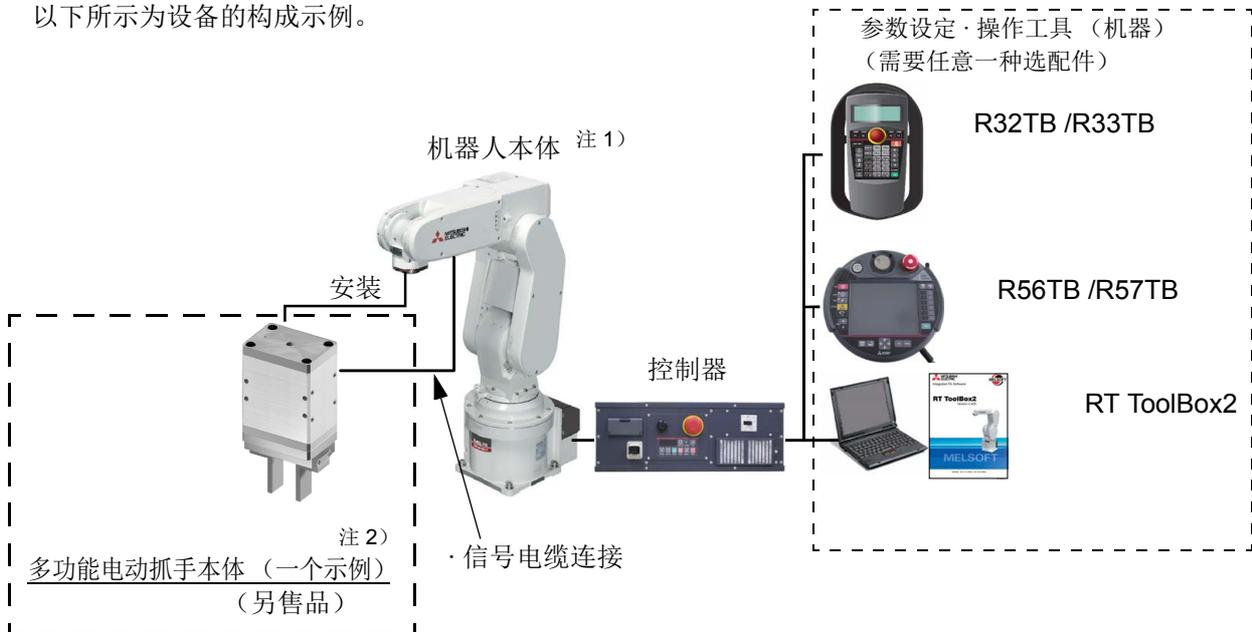
下面对使用 TAIYO 公司制造的多功能电动抓手时的设定及操作方法进行说明。本书中未记载的事项请参阅多功能电动抓手产品附带的使用说明书。

注) 本章下文中的“电动抓手”即表示“多功能电动抓手”。

### 7.2.1 概要

#### (1) 设备构成 (例)

以下所示为设备的构成示例。



注1) 图示为 RV-2F 的构成示例。

RV-2F 为标准规格的机器人，可使用多功能电动抓手。

其他机器人为特殊规格。(仅限垂直多关节 6 轴型) 请参阅另外发行的验收规格书。

注2) 多功能电动抓手本体的图仅记载一个示例。关于构成多功能电动抓手的附件以及规格等，请参阅多功能电动抓手附带的使用说明书。

### ⚠ 注意

在调整作业等对多功能电动抓手部分进行调整时，请务必关闭驱动电源。否则可能会发生手或手指被夹的危险。

另外，即使在机器人本体的伺服 OFF 操作中，多功能电动抓手仍然保持驱动电源 ON 状态。这是为了防止抓手所把持的工件意外掉落。

示教和调整作业中，请在把握多功能电动抓手的驱动电源 ON/OFF 状态的基础上，安全地进行作业。

可以通过来自示教单元的操作关闭多功能电动抓手的驱动电源。

通过执行抓手开 / 闭等的操作或动作指令，自动开启驱动电源。

图 7-5：使用多功能电动抓手的构成示例

## 7.2.2 规格

表 7-1：规格

| 项目         | 单位 | 规格                                                                   | 备注                                                                                                         |
|------------|----|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| 最大连接台数     | 台  | 1                                                                    | 表示抓手 1、抓手 2、抓手 3。                                                                                          |
| 连接规格       | -  | 远程 I/O 连接                                                            |                                                                                                            |
| 通道号码       | -  | 通道 1                                                                 |                                                                                                            |
| 站号（输入输出号码） | -  | 抓手 1：站号 5（700 ~ 731）<br>抓手 2：站号 6（732 ~ 763）<br>抓手 3：站号 7（764 ~ 795） | 站号通过多功能电动抓手本体的开关进行设定。<br>可以通过站号的设定来区分 3 种抓手。同时只能使用 1 台抓手。<br>※ 左侧所示输入输出号码的信号被多功能电动抓手占用。<br>无法用作用户程序等的信号输出。 |
| 示教点数（最大）   | 点  | 32                                                                   | 1 台抓手的最大示教点数。                                                                                              |

## 7.2.3 限制事项

### (1) 可使用的设备和软件版本

多功能电动抓手可以通过表 7-2 所示的机器人和设备、工具以及对象的软件版本来使用。

表 7-2：可使用的设备和软件版本

| 设备       | 控制器   | R32TB/R33TB | R56TB/R57TB | RT ToolBox2 |
|----------|-------|-------------|-------------|-------------|
| 可使用的软件版本 | R3 以后 | Ver. 1.6 以后 | Ver. 2.5 以后 | Ver. 2.1 以后 |

【注意】：多功能电动抓手在电源开启后开始动作前，需要进行一次原点回归。  
因此不对应继续功能（通过参数：CTN 指定）。

### (2) 站号设定

请通过多功能电动抓手的旋转开关将站号设定为 5、6、7 中任意一个使用。

以下所示为抓手号码与站号的关系。抓手号码与作为示教单元等的操作对象的抓手号码的切换、设定的参数名称和机器人语言、通过状态变量指定的抓手号码、以及报警号码建立了关联。

表 7-3：抓手号码与站号的关系

| 名称   | 抓手号码 | 站号 |
|------|------|----|
| 抓手 1 | 1    | 5  |
| 抓手 2 | 2    | 6  |
| 抓手 3 | 3    | 7  |

注) 同时只能使用 1 台抓手。

设定用旋转开关  
(出货时设定为 5)



图 7-6：站号设定

## (3) 多功能电动抓手的驱动电源（与机器人本体的关系）

多功能电动抓手的驱动电源 ON/OFF 状态与机器人本体的伺服 ON/OFF 不同步。即使将机器人本体紧急停止，多功能电动抓手的驱动电源也不会关闭。另外，即使在机器人本体的伺服 OFF 操作中，多功能电动抓手的驱动电源仍然保持 ON 状态。

多功能电动抓手的驱动电源 OFF 通过示教单元的操作进行。此外，驱动电源 ON 通过使其动作的操作和指令的执行自动开启。

表 7-4：多功能电动抓手的驱动电源 ON/OFF

| 操作              | 驱动电源 ON/OFF 状态 | 备注                          |
|-----------------|----------------|-----------------------------|
| 来自示教单元的抓手伺服 OFF | OFF            |                             |
| 机器人的紧急停止 ON     | 维持当时的状态        | 由于维持当时的状态，因此可以防止所把持的工件意外掉落。 |
| 机器人本体的伺服 OFF    | 维持当时的状态        |                             |
| 机器人本体的伺服 ON     | 维持当时的状态        |                             |
| 来自示教单元的抓手开闭操作   | ON             | 自动开启驱动电源。                   |
| 抓手动作指令的执行       | ON             |                             |

## 7.2.4 操作注意事项

## (1) 原点回归

多功能电动抓手在接通电源后需要执行一次原点回归动作。

在原点回归未完成的状态下驱动多功能电动抓手会发生报警，但机器人本体可通过程序的执行等进行动作。



## 注意

机器人动作时，可能会因为多功能电动抓手的开闭状态而导致与工件或周边设备等发生干涉。

为防止发生这样的干涉，请在充分注意的基础上进行操作和编程，确保在确认多功能电动抓手的状态的同时驱动机器人。

（备有用于通过程序确认多功能电动抓手的状态的状态变量）。

以下所示为确认原点回归完成状态的程序示例。（状态变量和指令的详细内容记载在第 561 页的「7.2.8 机器人语言的规格」中）

此外，在机器人把持工件的动作开始前，需要确认抓手是否已打开。

**【例 1】：**通过示教单元等的操作进行原点回归时

[功能] 如果程序启动时多功能电动抓手 1 的原点回归未完成，将发生报警 9100。

```

1 If M_EHOrg = 0 THEN '如果原点回归未完成（状态变量 M_EHOrg 的值为 "0"）
2 Error 9100 '发生报警 9100。
3 EndIf
4 以下，如果在作业程序 '步骤 1 中完成了原点回归，将执行本步骤以后的步骤。
5 :
6 :
```

补充：发生的报警事先通过第 428 页的「UER1 ~ UER20」参数进行设定。

**【例 2】：** 通过程序的执行进行原点回归时

[功能] 如果程序执行中多功能电动抓手 1 的原点回归处于未完成状态，将发生报警 9100。

```

1 EHOrg 1 ' 执行原点回归。
2 Def Act 1, M_EHOrg = 0 GoSub *ORGLOST ' 定义中断处理（如果原点回归处于未完成状态，则处理转为标签：*ORGLOST）
3 Wait M_EHOrg = 1 ' 等待原点回归完成。
4 Act 1 = 1 ' 将中断处理设为有效。
5 以下为作业程序
6 : ' （如果执行中原点回归处于未完成状态
7 : ' 处理转为标签：*ORGLOST）
:
999 *ORGLOST '
1000 Error 9100 ' 发生报警 9100。
1001 Return 0 ' 返回至中断时的步骤。

```

补充：①发生的报警事先通过第 428 页的「UER1 ~ UER20」参数进行设定。

②虽然在程序执行中原点回归未完成时，会发生与故障原因对应的报警（电机过载等），但通过执行上述程序，可以采取双重安全对策。

## (2) 关于在 ALWAYS（始终执行）程序中的执行

可以通过 ALWAYS 程序执行各电动抓手指令。可通过 ALWAYS 程序执行，因此没有需要进行设定的内容。

表 7-5：电动抓手控制指令（概要）

| 指令      | 功能         |
|---------|------------|
| EHOpen  | 抓手打开       |
| EHClose | 抓手关闭       |
| EHMov   | 向指定位置移动    |
| EHHold  | 定位把持动作     |
| EHOrg   | 原点回归       |
| EHStop  | 抓手停止       |
| EHTbl   | Table 动作指令 |

## 通过 ALWAYS 程序执行时的注意事项

- 1) 即使通过 ALWAYS（始终执行）程序使用电动抓手指令，在仅执行 ALWAYS 程序时，运行中的输出信号保持 OFF 不变。
- 2) 发生（重复执行、过载、反馈报警等）电动抓手的报警时，程序会继续运行。通过确认 M\_Err 变量的值，能够实现在发生报警时不执行指令的编程。
- 3) 即使开启紧急停止，使用了电动抓手指令的 ALWAYS（始终执行）程序会继续运行，不会停止。通过确认 M\_Err 变量的值，能够实现在发生报警时不执行指令的编程。
- 4) 电源开启后（操作面板变为速度比例显示的时机）约 2 秒钟内会向电动抓手发送初始数据，如果在此期间执行电动抓手的各项指令，将发生重复执行的报警。请等到 M\_EHBusy 变为 0 后再执行。

## ALWAYS 程序的示例

```

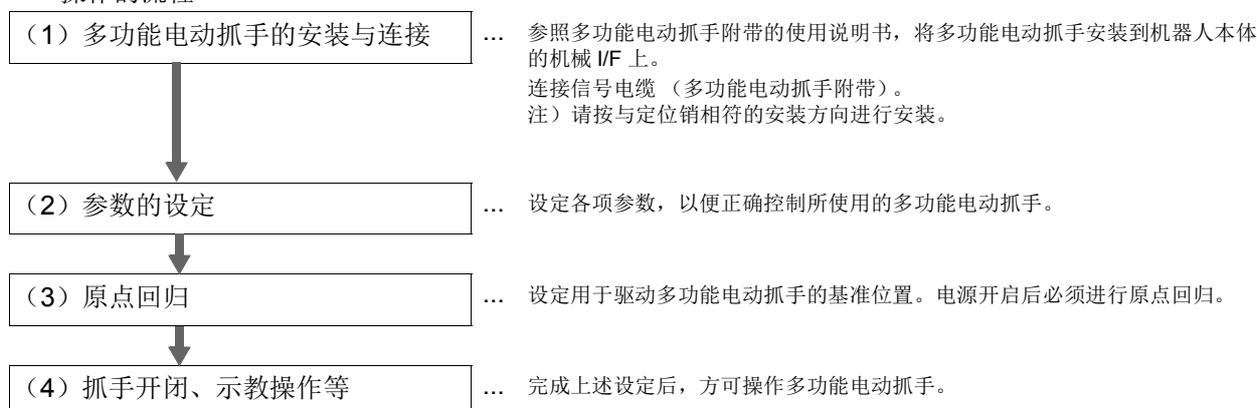
1 While M_EHBusy (1) =0 '等待电源开启时的初始化完成
2 If M_Err=0 Then '仅未发生报警时进行电动抓手的处理
3 If M_EHOrg (1) =0 Then '原点未回归时，执行原点回归
4 EHOrg 1
5 Wait M_EHOrg (1) =1
6 EndIf
7 If M_In (20) =1 Then '输入信号 20 号为 ON 时抓手打开
8 EHOpen 1,50,30 '打开抓手
9 EndIf
10 If M_In (21) =1 Then '输入信号 21 号为 ON 时抓手关闭
11 EHClose 1,50,30 '关闭抓手
12 EndIf
13 Wait _EHBusy (1) =0 '等待抓手的处理完成
14 EndIf
15 End

```

## 7.2.5 操作步骤

以下所示为多功能电动抓手的操作步骤概要。

< 操作的流程 >



## 7.2.6 参数的设定

设定多功能电动抓手的控制所需的参数。

通过示教单元（R32TB/R33TB、R56TB/R57TB）或 RT ToolBox2 进行设定。关于参数变更方法的详细内容，请参阅各产品附带的使用说明书。R56TB/R57TB 及 RT ToolBox2 有专用画面。

- R32TB/R33TB：本书「3.14 参数画面的操作」
- R56TB/R57TB：R56TB/R57TB 使用说明书（BFP-A8591）
- RT ToolBox2：RT ToolBox2 / RT ToolBox2 mini 使用说明书（BFP-A8617）

**【注意】** ①请正确指定与所使用的抓手号码（通过旋转开关进行的站号设定）对应的参数名。  
本书中，参数名中与抓手号码对应的字符用 "n" 表示。

< 例 > 执行器类型：EHnTYPE

设定抓手号码 1 的执行器类型时，参数名指定为 "EH1TYPE"。

②设定执行器类型后，控制参数将被初始化。

为应对机器人的更换等保养时，建议使用选配的 RT ToolBox2 对手动变更后的参数值进行备份。

③写入参数时，请务必连接电动抓手的信号电缆。

### (1) 执行器类型的设定

设定所使用的多功能电动抓手的类型。请参照所使用的多功能电动抓手附带的使用说明书进行设定。

表 7-6 所示为要设定的参数内容。

表 7-6：执行器类型设定参数

| 参数    | 参数名     | 排列数<br>文字数 | 内容说明                                                                                       | 出货时设定值 |
|-------|---------|------------|--------------------------------------------------------------------------------------------|--------|
| 执行器类型 | EHnTYPE | 整数 1       | 设定多功能电动抓手本体的类型（机种）。<br>设定范围为 0~65535。表 7-7 所示为多功能电动抓手本体的各类型（电动抓手型号）的设定值。<br>n：表示抓手号码（1~3）。 | 0      |

注）设定本参数后，会从多功能电动抓手本体读取控制参数并自动设定到控制器的参数中。（初始值）此前进行过变更的参数将被覆盖，建议事先使用选配的 RT ToolBox2 进行备份。

表 7-7： 电动抓手各型号的设置值

| 电动抓手型号      | 设定值 | 电动抓手型号      | 设定值 | 电动抓手型号     | 设定值 | 电动抓手型号  | 设定值 |
|-------------|-----|-------------|-----|------------|-----|---------|-----|
| SS-2010     | 110 | SS-2005-3N  | 111 | SS-2005-5N | 112 | ST-2013 | 118 |
| ST-2004     | 119 | SS-2815     | 120 | ST-2820    | 129 | SS-4225 | 130 |
| ST-4230     | 139 | SD-2005     | 210 | SD-2810    | 220 | SD-4220 | 230 |
| FS (T) 2020 | 310 | FS (T) 2840 | 320 |            |     |         |     |

## (2) 控制参数的设定

表 7-8 所示为用于控制多功能电动抓手的参数一览，表 7-12 所示为各参数的详细内容。

请根据需要进行变更。

进行变更时，请参阅多功能电动抓手附带的使用说明书。

表 7-8： 控制参数一览

| No | 参数       | 参数名注 1)               | 排列数<br>文字数 | 内容说明                                                                 | 参照页码 |
|----|----------|-----------------------|------------|----------------------------------------------------------------------|------|
| 1  | 软限制 + 侧  | EHnLMTp               | 整数 1       | 多功能电动抓手的软限制 + 侧<br>[0 ~ 9999] 以 0.1mm 为单位                            | 576  |
| 2  | 软限制 - 侧  | EHnLMTM               | 整数 1       | 功能电动抓手的软限制 - 侧<br>[-9999 ~ 0] 以 0.1mm 为单位                            | 576  |
| 3  | 行程       | EHnSTRK               | 整数 1       | 多功能电动抓手的行程<br>[0 ~ 9999] 以 0.1mm 为单位                                 | 576  |
| 4  | 原点回归方向   | EHnORGD               | 整数 1       | 多功能电动抓手的原点回归方向<br>[0: OPEN/1: CLOSE]                                 | 577  |
| 5  | 原点回归速度   | EHnORGV               | 整数 1       | 多功能电动抓手的原点回归速度<br>[程序速度 20 ~ 50]                                     | 577  |
| 6  | 原点偏移     | EHnORGSF              | 整数 1       | 多功能电动抓手的原点偏移<br>[-9999 ~ 9999] 以 0.01mm 为单位                          | 577  |
| 7  | 加速度      | EHnACC                | 整数 1       | 多功能电动抓手的加速度<br>[1 ~ 100] %                                           | 576  |
| 8  | 程序最高速度   | EHnVMAX               | 整数 1       | 多功能电动抓手的程序最高速度<br>[1 ~ 100] %                                        | 577  |
| 9  | 把持速度     | EHnVHLD               | 整数 1       | 多功能电动抓手的把持速度<br>[1 ~ 100] %                                          | 577  |
| 10 | 恒速移动距离   | EHnZNCV               | 整数 1       | 多功能电动抓手的恒速移动距离<br>[1 ~ 9999] 以 0.01mm 为单位                            | 577  |
| 11 | 限制宽度     | EHnLMTW               | 整数 1       | 多功能电动抓手的限制宽度<br>[1 ~ 9999] 以 0.01mm 为单位                              | 577  |
| 12 | 定位完成距离   | EHnPSCD               | 整数 1       | 多功能电动抓手的定位完成距离<br>[1 ~ 9999] 以 0.01mm 为单位                            | 576  |
| 13 | 原点回归方式   | EHnORGS               | 整数 1       | 多功能电动抓手的原点回归方式<br>[0: 行程端 /1: 行程端 +Z 相检测方向]                          | 577  |
| 14 | 点位数据     | EHnPOS1<br>~ EHnPOS32 | 整数 1       | 多功能电动抓手的点位数据设定<br>[-999.99 ~ 999.99] 以 0.01mm 为单位                    | 577  |
| 15 | Table 数据 | EHnTBL1<br>~ EHnTBL32 | 实数 7       | 多功能电动抓手的动作条件 Table 数据                                                | 577  |
| 16 | 示教单元动作模式 | EHOPEMD               | 整数 1       | 来自示教单元的动作模式初始值<br>[0: 点位指定 /1: Table 指定]                             | 578  |
| 17 | 现在位置获取模式 | EHCURMD               | 整数 1       | 示教单元的现在位置显示<br>[0: 始终显示, 1: MANUAL 模式始终显示, AUTOMATIC 模式在动作结束时显示 1 次] | 578  |

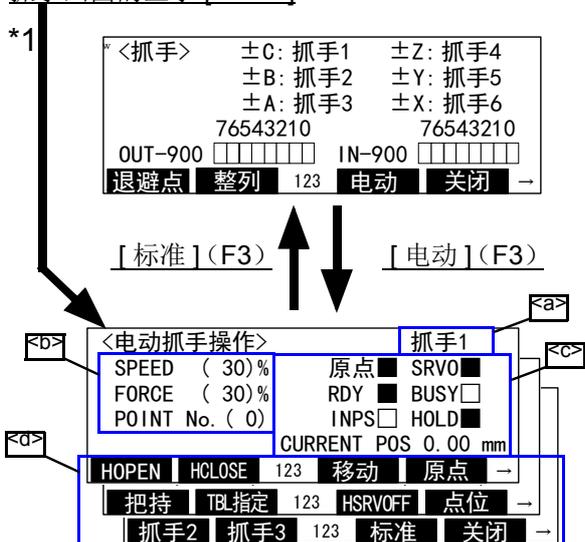
注 1) n : 表示抓手号码 (1 ~ 3)。

## 7.2.7 多功能电动抓手的操作

本节介绍的是使用 R32TB 的多功能电动抓手的操作方法。关于使用 R56TB/R57TB 进行的操作，请参阅另一手册「使用说明书 /R56TB/R57TB 使用说明书（BFP-A8591）」。

图 7-7 所示为 R32TB 上的多功能电动抓手操作画面。

### 抓手画面的显示 [HAND]



\*1) 连接了多功能电动抓手时，最初显示 <Electric Hand Operation> 画面。

<a>

显示现在对象的多功能电动抓手号码。  
通过按下与“抓手 n”（n 为抓手号码 1 ~ 3）对应的功能键 [F1] 或 [F2] 键来切换对象抓手。

<b>

设定多功能电动抓手动作的数据。  
• SPEED ..... 以 % 为单位设定速度 [1 ~ 100]  
• FORCE ..... 以 % 为单位设定力（把持力）[1 ~ 100]  
• POINT No. 或 Table No.  
※ 通过功能键 "PNT SET"、"TBL SET" 进行切换  
• POINT No ..... 指定示教位置号码 [1 ~ 32]  
• Table No ..... 指定 Table 编号 [1 ~ 32]

<c>

显示多功能电动抓手的状态。  
• 原点 .....  : 原点回归完成 /  : 未完成  
• SRVO .....  : 伺服 ON /  : OFF  
• RDY .....  : 准备完成 /  : 准备中 / 报警发生中  
• BUSY .....  : 动作中 /  : 非动作中  
• INPS .....  : 到达目标位置 /  : 未到达  
• HOLD .....  : 把持中 /  : 未把持  
• CURRENT POS ... 有符号的绝对位置（以 0.01mm 为单

<d>

按下与所显示的各功能对应的功能键（[F1] ~ [F4]）进行操作。

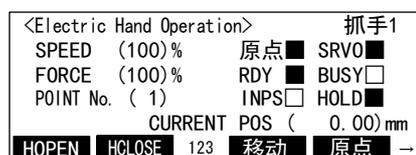
（未显示所需的功能时，请按 [FUNCTION] 键）

- “HOPEN” ..... 打开抓手。（如果松开按键，会在中途停止）
- “HCLOSE” ..... 关闭抓手。（如果松开按键，会在中途停止）
- “移动” ..... 向“POINT No.”中所示示教位置移动。（如果松开按键，会在中途停止）
- “原点” ..... 进行原点回归（如果松开按键，会在中途停止）
- “夹持”或“动作”（※ 每选择即切换）
  - “把持” .... 进行抓手把持动作。（如果松开按键，会在中途停止）
  - “动作” ..... 根据通过指定的 Table 编号设定的条件进行动作。
- “TBL SET”或“PNT SET”（※ 每次选择即切换）
  - “TBL SET” .. 指定 Table 编号（使之动作的条件）。（画面的 <b> 中显示“Table No.”）  
注）指定 Table 编号时，在 Table 中被指定的速度、力的条件有效。
  - “PNT SET” .. 指定示教位置编号。（画面的 <b> 中显示“POINT No.”）
- “HSRVOFF” .. 切断多功能电动抓手的驱动电源  
※ 可通过人力移动来进行直接示教。
- “点位” ..... 将多功能电动抓手的位置反映在示教画面上。
- “抓手 n” ..... 切换操作对象多功能电动抓手号码。（“n”为抓手号码）  
注）不显示不存在的多功能电动抓手号码。
- “标准” ..... 返回 < 抓手 > 画面。
- “关闭” ..... 关闭 <Electric Hand Operation> 画面及 < 抓手 > 画面，返回原来的画面。

图 7-7： 多功能电动抓手操作画面

以下所示为各项功能的操作方法。这里介绍的是在显示 <Electric Hand Operation> 画面状态下的操作方法。请事先连接多功能电动抓手，按下示教单元 ENABLE 开关以启用示教单元，按 [HAND] 键以显示 <Electric Hand Operation> 画面。

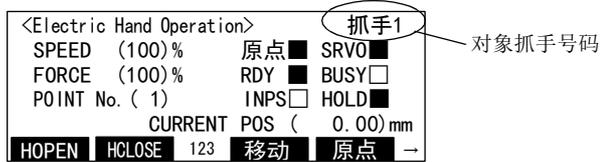
抓手画面的显示：  
[HAND]



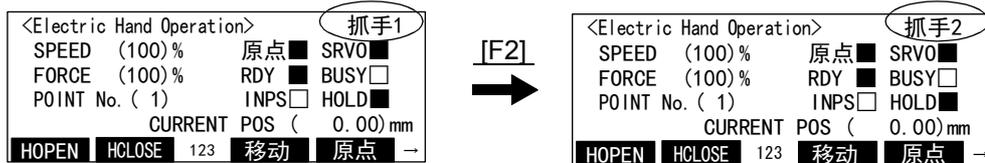
[注意] 外部输入信号的停止输入（STOP、STOP2）为 ON 时，抓手不动作。此外，抓手动作中如果停止输入信号开启，抓手会立即停止。

## (1) 抓手号码的切换

- 1) <Electric Hand Operation> 画面的右上角显示有现在对象抓手号码。  
抓手号码不同时，请通过以下方法进行切换。



- 2) 按下与“抓手 n”（n 为 1 ~ 3）对应的功能键来切换对象抓手号。



对象抓手号码的切换：[F1]、[F2]

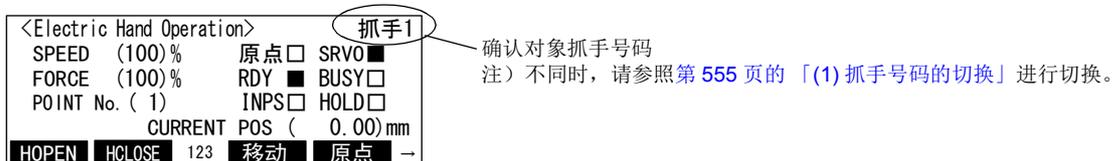
注) 图示为从抓手号码 1 切换到抓手号码 2 的示例。  
未显示“抓手 n”时，请按 [FUNCTION] 键。

至此，对象抓手号码的切换完成。

## (2) 原点回归

原点回归是设定用于驱动多功能电动抓手的基准位置的动作，在驱动抓手前必须实施原点回归。

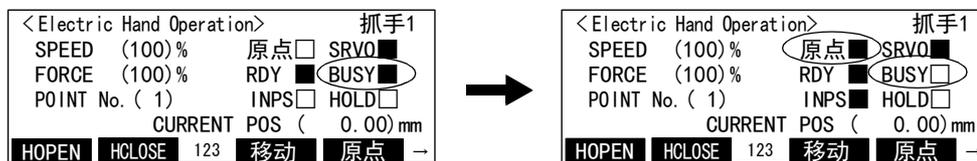
- 1) 请确认原点回归对象抓手号码。（<Electric Hand Operation> 画面右上角的显示）不同时请进行切换。



- 2) 长按与 <Electric Hand Operation> 画面的“原点”对应的功能键（[F4]）。

按下按键期间，多功能电动抓手执行原点回归动作。执行中，示教单元画面的“BUSY”显示将变为“■”。

原点回归完成后，“BUSY”显示将变为“□”，“原点”显示将变为“■”。



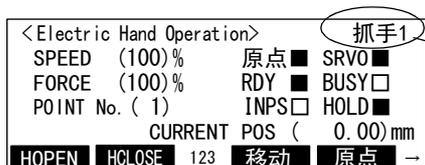
原点回归：[F4]  
\* 松开按键后动作停止。

原点回归完成。

至此，多功能电动抓手的原点回归操作完成。

## (3) 抓手开闭操作

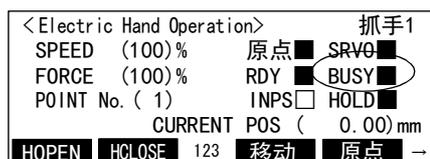
- 1) 请确认抓手开闭对象的抓手号码。（<Electric Hand Operation> 画面右上角的显示）不同时请进行切换。



确认对象抓手号码

注) 不同时, 请参照第 555 页的「(1) 抓手号码的切换」进行切换。

- 2) 在按下与 <Electric Hand Operation> 画面的 "HOPEN""HCLOSE" 对应的功能键期间, 抓手动作。动作中, "BUSY" 显示将变为 "■"。



打开抓手: "HOPEN" ([F1])  
关闭抓手: "HCLOSE" ([F2])

\* 松开按键后动作停止。

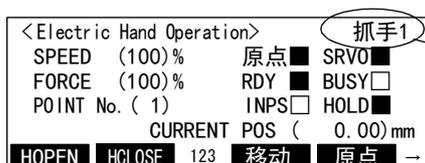
- 3) 至抓手开 / 闭端的移动完成后, "INPS" 显示将变为 "■"。  
注) 移动途中进行把持时, "HOLD" 显示将变为 "■"。

## (4) 动作位置示教

可以事先对多功能电动抓手的位置进行示教, 通过程序 (EHMov 指令) 或示教单元进行移动。

多功能电动抓手 1 到 3 分别可进行最多 32 点的示教, 通过号码进行指定。除了登录多功能电动抓手的现在位置的方法外, 还可以通过直接输入数值来登录示教。步骤如下所示。

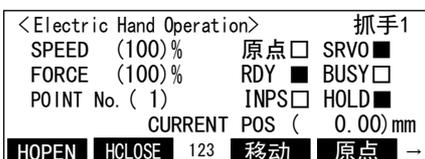
- 1) 请确认示教对象抓手号码。（<Electric Hand Operation> 画面右上角的显示）不同时请进行切换。



确认对象抓手号码

注) 不同时, 请参照第 555 页的「(1) 抓手号码的切换」进行切换。

- 2) 请通过 "HOPEN"、"HCLOSE" 等操作将多功能电动抓手移动到示教位置。



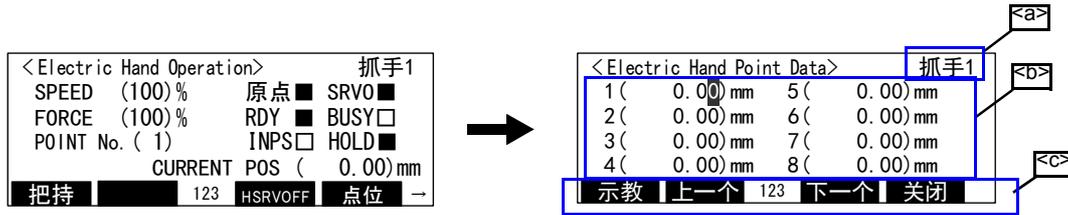
向示教位置移动

- 打开抓手: "HOPEN" ([F1])
- 关闭抓手: "HCLOSE" ([F2])

\* 松开按键后动作停止。

- 也可以切断驱动电源, 直接用手移动进行示教。

3) 按下与 <Electric Hand Operation> 画面的 " 点位 " 对应的功能键 ([F4])，以显示 <Electric Hand Point Data> 画面。



<Electric Hand Point Data> 画面的显示 : [F4]

- <a>  
显示示教对象多功能电动抓手号码。
- <b>  
显示位置数据。  
可通过方向箭头键移动光标，通过数值输入进行设定（微调）。
- <c>  
  - “示教”..... 对光标位置的位置数据进行示教。  
抓手的现在位置被设定为对象位置数据。
  - “上一个”... 每次按下按键时，对象位置数据按 1-8 ⇒ 25-32 ⇒ 17-24 ⇒ 9-16 ⇒ 1-8 顺序切换。
  - “下一个”... 每次按下按键时，对象位置数据按 1-8 ⇒ 9-16 ⇒ 17-24 ⇒ 25-32 ⇒ 1-8 顺序切换。
  - “关闭”..... 关闭 <Electric Hand Point Data> 画面，返回 <Electric Hand Operation> 画面。

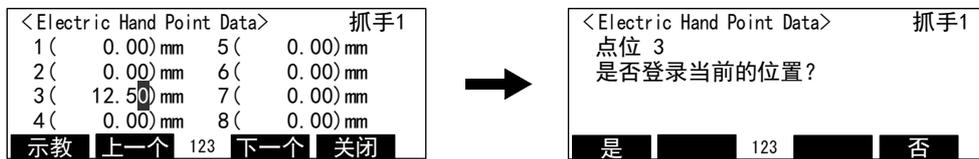
4) 对位置进行示教。

如下所示示教有， a) 登录多功能电动抓手的现在位置和 b) 输入数值进行登录的两种方法。

a) 登录多功能电动抓手的现在位置

① 按方向箭头键，将光标移动到要进行示教的点位号。未显示所需点位号时，按与 " 上一个 " 或 " 下一个 " 对应的功能键以使其显示。

按下与 " 示教 " 对应的功能键 ([F1])。将显示确认画面。

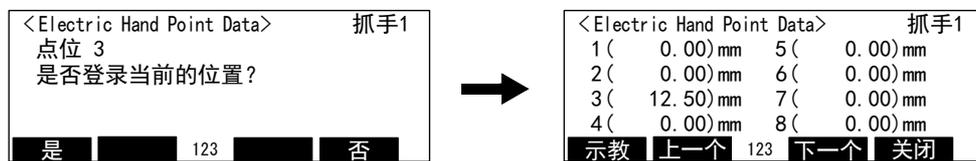


现在位置的示教 : [F1]

注) 图示为在抓手号码 1 的点位 3 进行现在位置的示教的示例。

② 按下与 " 是 " 对应的功能键 ([F1]) 后，现在位置即被登录。

按下与 " 否 " 相对应的功能键 ([F4]) 后，取消登录。

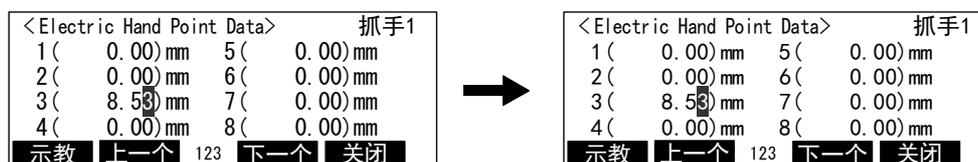


示教的执行 : [F1]，  
取消按 [F4]

至此，多功能电动抓手的现在位置即被登录。

## b) 输入数值进行登录

- ①按方向箭头键，将光标移动到要进行示教的点位号。未显示所需要的点位号时，按与"上一个"或"下一个"对应的功能键以使其显示。  
直接输入数值并按 [EXE] 键。



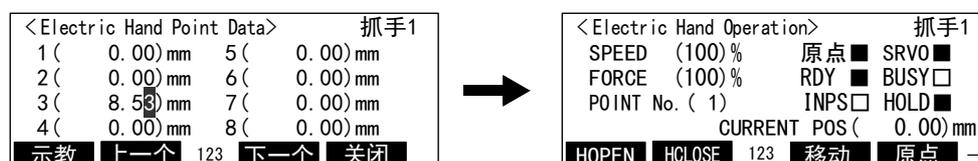
通过数值输入进行示教: [数字] 键、[EXE]

注) 图示为将抓手号码 1 的点位 3 设定为 "8.53"mm 的示例。

至此，通过数值输入进行的位置数据登录完成。

## 5) 关闭多功能电动抓手点位设定画面。

按下与"关闭"对应的功能键 ([F4]) 后，将返回 <Electric Hand Operation> 画面。



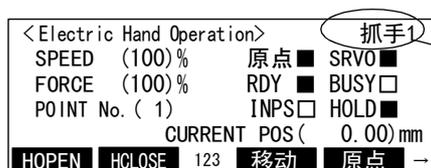
<Electric Hand Point Data> 画面的关闭: [F4]

至此，多功能电动抓手的动作位置示教完成。

## (5) 向示教位置移动

可以将多功能电动抓手向已示教位置移动。

- 1) 请确认移动对象抓手号码。( <Electric Hand Operation> 画面右上角的显示) 不同时请进行切换。

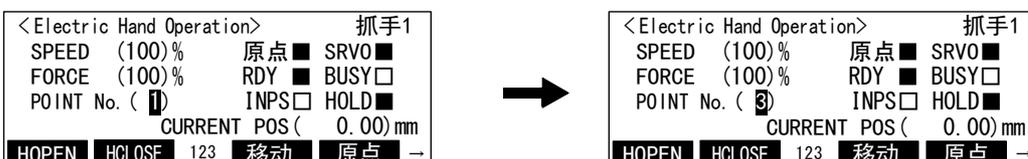


确认对象抓手号码

注) 不同时，请参照第 555 页的「(1) 抓手号码的切换」进行切换。

- 2) 在画面上的"POINT No."中指定要移动的点位号。

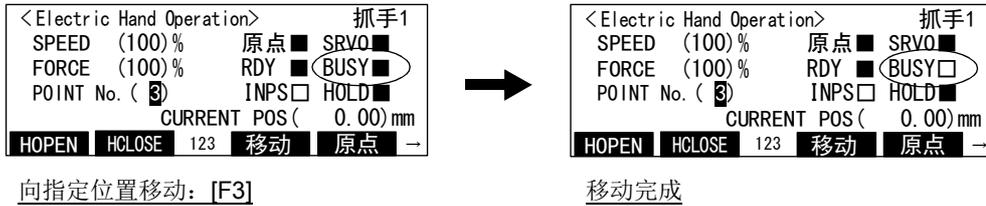
通过方向箭头键将光标移动到"POINT No."的括号内，输入数字 (1 ~ 32) 后按 [EXE] 键。



光标移动: [箭头]、数值输入[数字]

注) 图示为指定抓手号码 1 的点位 3 的示例。

- 3) 长按与画面上的 "移动" 对应的功能键 ([F3])。按住期间, 抓手向指定的位置号码移动。移动中, "BUSY" 显示将变为 "■"。  
向目标位置的移动完成后, "BUSY" 显示将变为 "□", "INPS" 显示将变为 "■"。

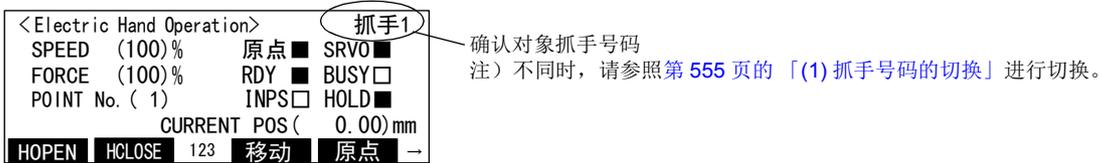


至此, 向示教位置的移动完成。

(6) 把持动作

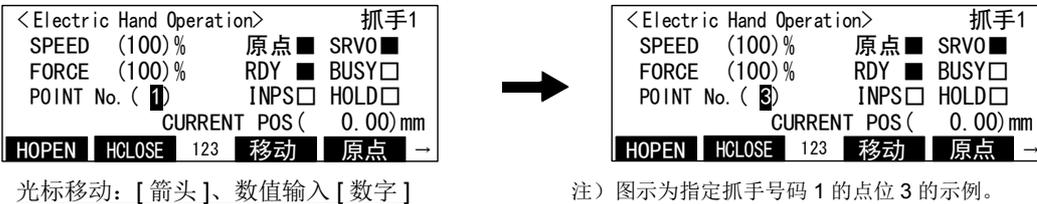
在已示教的位置执行把持动作。梯形速度、恒速移动距离等的把持动作模式以参数的设定值为基础。以下所示为把持动作的操作方法。

- 1) 请确认动作对象抓手号码。( <Electric Hand Operation> 画面右上角的显示) 不同时请进行切换。

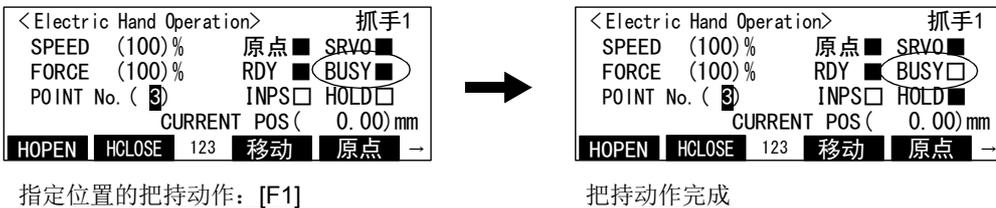


- 2) 在画面上的 "POINT No." 中指定要把持的点位号码。

通过方向箭头键将光标移动到 "POINT No." 的括号内, 输入数字 (1 ~ 32) 后按 [EXE] 键。



- 3) 长按与画面上的 "把持" 对应的功能键 ([F1])。按住期间, 抓手在指定的位置号码处执行把持动作。移动中, "BUSY" 显示将变为 "■"。



- 4) 把持工件后, "BUSY" 显示将变为 "□", "HOLD" 显示将变为 "■"。

注) 不把持工件而移动到目标位置时, "INPS" 显示将变为 "■"。

至此, 示教位置的把持动作完成。

## (7) 动作速度的设定

可以通过示教单元设定驱动多功能电动抓手时的速度。设定值为相对于最高速度的比例（%）。该设定值在各抓手号码之间通用。

设定操作方法如下所示。

- 1) 通过方向箭头键将光标移动到 "SPEED" 的括号内，输入数值后按 [EXE] 键。



速度的设定: [箭头]、[数字]、[EXE]



设定完成

注) 图示为将速度设定为 50% 的示例。

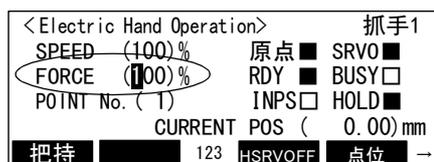
至此，动作速度的设定完成。

## (8) 力（把持力）的设定

可以通过示教单元设定驱动多功能电动抓手时的力（把持力）。设定值为相对于最高把持力的比例（%）。该设定值在各抓手号码之间通用。

设定操作方法如下所示。

- 1) 通过方向箭头键将光标移动到 "FORCE" 的括号内，输入数值后按 [EXE] 键。



力（把持力）的设定: [箭头]、[数字]、[EXE]



设定完成

注) 图示为将力设定为 30% 的示例。

注) 设定值较小时，有可能会因为抓手自身的摩擦而导致抓手不动作。

例) 多功能电动抓手: ESG1-SS-2815-11XW107 以 10% 以上的把持力进行动作。

至此，力的设定完成。

## 7.2.8 机器人语言的规格

### (1) 指令一览

表 7-9 所示为用于驱动多功能电动抓手的指令一览。

表 7-9：多功能电动抓手控制指令

| No | 指令                           | 说明                   | 参照页码 |
|----|------------------------------|----------------------|------|
| 1  | EHOpen/EHClose (抓手打开 / 抓手关闭) | 至行程开端的动作 / 至行程闭端的动作  | 562  |
| 2  | EHMov (抓手移动)                 | 向指定位置的定位动作           | 564  |
| 3  | EHHold (抓手保持)                | 在指定位置的定位把持动作         | 566  |
| 4  | EHOrg (抓手原点)                 | 原点回归动作               | 568  |
| 5  | EHServo On/Off (hand servo)  | 控制伺服电机电源的 ON/OFF     | 569  |
| 6  | EHStop (抓手停止)                | 动作的停止                | 570  |
| 7  | EHTbl (Hand Table)           | 以指定的 Table 号码的条件进行动作 | 571  |

### (2) 状态变量一览

表 7-10 所示为与多功能电动抓手有关的状态变量一览。

表 7-10：多功能电动抓手状态变量

| No | 变量名称                               | 配列要素数 <sup>注1)</sup> | 内容                                     | 属性 <sup>注2)</sup> | 数据类型、单位    | 参照页码 |
|----|------------------------------------|----------------------|----------------------------------------|-------------------|------------|------|
| 1  | M_EHPos1/<br>M_EHPos2/<br>M_EHPos3 | 点位号码 (1 ~ 32)        | 各多功能电动抓手 (1~3) 的点位数据                   | RW                | 单精度实数型, mm | 572  |
| 2  | M_EHCur                            | 抓手号码 (1 ~ 3)         | 多功能电动抓手的现在位置                           | R                 | 单精度实数型, mm | 573  |
| 3  | M_EHBusy                           | 抓手号码 (1 ~ 3)         | 多功能电动抓手的动作状态<br>[1: 执行中/0: 非执行中]       | R                 | 整数型        | 574  |
| 4  | M_EHInPs                           | 抓手号码 (1 ~ 3)         | 多功能电动抓手的目标位置到达状态<br>[1: 到达目标位置/0: 未到达] | R                 | 整数型        | 574  |
| 5  | M_EHHold                           | 抓手号码 (1 ~ 3)         | 多功能电动抓手的把持检测状态<br>[1: 检测到把持/0: 未检测到]   | R                 | 整数型        | 574  |
| 6  | M_EHZone                           | 抓手号码 (1 ~ 3)         | 多功能电动抓手的指定领域内<br>[1: 指定领域内/0: 指定领域外]   | R                 | 整数型        | 574  |
| 7  | M_EHOrg                            | 抓手号码 (1 ~ 3)         | 多功能电动抓手的原点回归完成状态<br>[1: 原点回归完成/0: 未完成] | R                 | 整数型        | 574  |
| 8  | M_EHTMd1/<br>M_EHTMd2/<br>M_EHTMd3 | Table 号码 (1 ~ 32)    | 多功能电动抓手指定的Table数据的动作模式                 | R                 | 整数型        | 575  |
| 9  | M_EHOPos                           | 抓手号码 (1 ~ 3)         | 多功能电动抓手的动作完成点位号码<br>[0: 原点未完成或移动中]     | R                 | 整数型        | 576  |

注 1) 点位号码 .....1 ~ 32, 用于指定位点数据 (示教号码)。

抓手号码 .....1 ~ 3, 用于指定抓手号码。

Table 号码 .....1 ~ 32, 用于指定抓手的 Table 号码。

注 2) R..... 只读。

RW..... 可读取、写入。

## (3) 指令的详细说明

各指令后的详细内容按不同功能显示。

**EHOpen/EHClose** (抓手打开/抓手关闭)**【功能】**

执行多功能电动抓手的开/闭动作。

**【格式】**

```
EHOpen □ < 抓手号码 >, <SPEED>, <FORCE> [, < 区段 ON 位置 >, < 区段 OFF 位置 >]
EHClose □ < 抓手号码 >, <SPEED>, <FORCE> [, < 区段 ON 位置 >, < 区段 OFF 位置 >]
```

**【术语】**

<抓手号码> ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。  
(注: 小数点以后四舍五入)

<SPEED>..... 以%单位的整数指定抓手的移动速度。以常数或数值变量进行指定。  
设定范围: 1 ~ 100 (%) (注: 小数点以后四舍五入)  
注) ①如果设定速度小于20%, 抓手有可能会出现振动。  
②如果设定速度大于50%, 有可能会发生过负载报警。

<FORCE>..... 以%单位的整数指定抓手的把持力。以常数或数值变量进行设定。  
设定范围: 1 ~ 100 (%) (注: 小数点以后四舍五入)

注) 如果设定值小于30%, 有可能会因为抓手自身的摩擦而导致抓手不动作。  
例) 多功能电动抓手: ESG1-SS-2815-11XW107以10%以上的把持力进行动作。

<区段ON位置>/<区段OFF位置>

要确认抓手是否在特定领域内时, 通过起始位置/结束位置指定该特定领域。  
(参照图7-8)以常数或数值变量进行指定。  
设定范围: -9999.99~9999.99 (mm) (以0.01mm为单位)

**【例子 1】** (无区段指定)

```
1 EHOpen 1, 100, 30 ' 打开抓手 1 (速度 =100%, 力 =30%)
2 Mov PUP ' 向把持位置上空移动
3 Wait M_EHInPs=1 ' 确认向开端的移动完成
4 Mvs PGET ' 向把持位置移动
5 Fine 0.5, P ' 等待机器人移动完成 注1)
6 EHClose 1, 50, 30 ' 关闭抓手 1 (速度 =50%, 力 =30%)
7 Wait M_EHBusy=0 ' 等待抓手的动作完成
8 If M_EHHold=1 Then ' 如果正在进行把持, 则向把持位置上空移动
9 Mvs PUP
10 EndIf
```

注 1) 通过使用 Fine 指令, 可以在机器人本体的动作完成后开始抓手的动作。但是, 通过 Fine 指令指定的脉冲数 (0.5) 为需要调整的值。(参阅第 216 页的 [Fine])  
此外, 通过 Dly 指令也能得到相同的效果。

## 【例子 2】（有区段指定）

|                                     |                                                |
|-------------------------------------|------------------------------------------------|
| 1 Mvs PCHK                          | ' 向尺寸确认位置移动                                    |
| 2 Fine 0.5, P                       | ' 等待机器人移动完成 注 1)                               |
| 3 EHClose 1, 50, 30, 20.00 22.00    | ' 关闭抓手 1<br>(速度 =50%, 力 =30%, 区段位置 =20 ~ 22mm) |
| 4 Wait M_EHBusy=0                   | ' 等待抓手的动作完成                                    |
| 5 If M_EHHold=1 AND M_EHZone=1 Then | ' 如果正在区段位置进行把持, 则在 MCHKOK (确认标志) 中代入 "1"。      |
| 6 MCHKOK=1                          |                                                |
| 7 Endlf                             |                                                |
| 8 EHMov 1, 0, 100, -10.00           | ' 将抓手 1 从现在位置打开 10mm (速度 =100%)                |
| 9 Wait M_EHInPs=1                   | ' 确认向指定位置的移动完成                                 |
| 10 Mvs PUP                          | ' 向尺寸确认位置上空移动                                  |

注 1) 通过使用 Fine 指令, 可以在机器人本体的动作完成后开始抓手的动作。但是, 通过 Fine 指令指定的脉冲数 (0.5) 为需要调整的值。(参阅第 216 页的「Fine」)  
此外, 通过 Dly 指令也能得到相同的效果。

## 【说明】

1) 指定速度、力来开 / 闭抓手。

以一定速度向开或闭方向移动到行程端为止。在行程途中检测到把持时, 将在该位置完成动作。

2) 自动运行中, 通过本指令使抓手动作开始后, 直接转入下个指令的执行。(不等待抓手动作结束)

a) 要等到抓手动作完成后再使机器人移动时, 请参照以下的状态变量。参照方法如上述例子所示。

- M\_EHBusy (抓手动作中) ..... 抓手动作中为 "1"
- M\_EHInPs (目标位置到达状态) .... 动作开始时为 "0"、到达开或闭端时为 "1"
- M\_EHHold (抓手把持状态) ..... 动作开始时为 "0"、行程途中进行把持时为 "1"

b) 抓手动作中, 执行了抓手动作指令的插槽停止时, 抓手动作也会暂停。

插槽重新启动后, 抓手动作也重新开始。

但是, 在插槽停止中时通过示教单元的手动操作等驱动对象抓手的情况下, 抓手动作不会重新开始。

c) 要通过程序中途停止 (结束) 抓手动作时, 请执行 EHStop 指令。

3) 单步运行或直接执行时, 将等待直到通过本指令执行的抓手动作结束为止。

4) 还可以指定区段位置以进行停止位置确认。

抓手在区段内时, 状态变量 M\_EHZone 变为 "1"。

例) 执行了 EHClose 1、50、30、5、8 时。(抓手的位置为 5mm ~ 8mm 时, M\_EHZone 的值变为 "1")

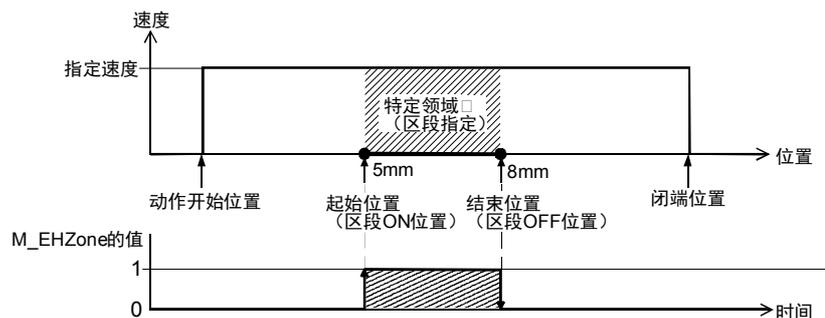


图 7-8：抓手动作和状态变量 (M\_EHZone) 的值 (例)

5) 在多功能电动抓手动作中执行了本指令时, 会发生报警 "抓手动作指令重复执行"。

有重复执行的可能性时, 请参照状态变量 M\_EHBusy (抓手动作中) 确认是否为抓手动作中后, 再执行本指令。

6) 控制器电源启动后, 请在使多功能电动抓手进行原点回归后再执行本指令。

不进行原点回归而执行本指令时, 会发生报警。

7) 购买多功能电动抓手后, 需要进行参数初始化 (机种选择)。

8) 不支持继续功能 (如果在动作中切断电源, 处理将被初始化)。

## EHMov (抓手移动)

### 【功能】

使多功能电动抓手向指定位置移动。

### 【格式】

EHMov □ < 抓手号码 >, <POINT No.>, <SPEED> [, <相对量 >]

### 【术语】

- <抓手号码> ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。  
(注: 小数点以后四舍五入)
- <POINT No.> ..... 指定多功能电动抓手的点位号码。以常数或数值变量进行指定。  
设定范围为0~32。(注: 小数点以后四舍五入)  
0为现在位置, 1~32将参数: EHNPOS1~32 (n: 抓手号码) 中设定的值设定为目的位置。
- <SPEED> ..... 以%单位的整数指定多功能电动抓手的移动速度。以常数或数值变量进行指定。  
设定范围: 1 ~100 (%) (注: 小数点以后四舍五入)  
注) 如果设定速度小于20%, 抓手有可能会出出现振动。
- <相对量> ..... 指定了该值时, 实际的移动目的位置变为按本设定值进行偏置后的位置。  
设定值以0.01mm为单位进行指定。以常数或数值变量进行指定。  
设定范围为-9999.99~9999.99 (mm)。(注: 0.01mm以后四舍五入)

### 【例子 1】 (无区段指定)

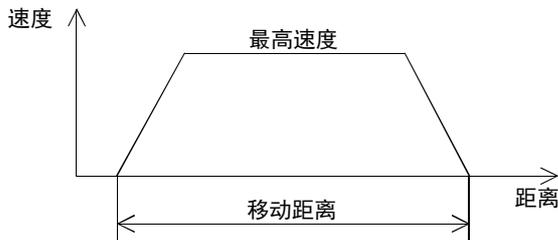
- |                      |                                                     |
|----------------------|-----------------------------------------------------|
| 1 EHMov 1, 5, 100    | ' 抓手 1 移动 (点位号码 =5, 速度 =100%)                       |
| 2 Mov PUP            | ' 向把持位置上空移动                                         |
| 3 Wait M_EHInPs=1    | ' 确认向指定位置的移动完成                                      |
| 4 Mvs PGET           | ' 向把持位置移动                                           |
| 5 Fine 0.5, P        | ' 等待机器人的移动完成 (参照 EHOOpen/EHClose (抓手打开 / 抓手关闭) 的例子) |
| 6 EHClose 1, 50, 30  | ' 抓手 1 闭 (速度 =50%, 力 =30%)                          |
| 7 Wait M_EHBusy=0    | ' 动作完成确认                                            |
| 8 If M_EHHold=1 Then | ' 把持确认                                              |
| 9 Mvs PUP            | ' 向把持位置上空移动                                         |
| 10 EndIf             |                                                     |

### 【例子 2】 (有区段指定)

- |                                     |                                             |
|-------------------------------------|---------------------------------------------|
| 1 Mvs PCHK                          | ' 向尺寸确认位置移动                                 |
| 2 EHClose 1, 50, 30, 20.00 22.00    | ' 抓手 1 闭 (速度 =50%, 力 =30%, 区段位置 =20 ~ 22mm) |
| 3 Wait M_EHBusy=0                   | ' 动作完成确认                                    |
| 4 If M_EHHold=1 AND M_EHZone=1 Then | ' 把持 & 区段确认                                 |
| 5 MCHKOK=1                          | ' 确认标志 =1                                   |
| 6 EndIf                             |                                             |
| 7 EHMov 1, 0, 100, -10.00           | ' 抓手 1 移动: 从现在位置打开 10mm (速度 =100%)          |
| 8 Wait M_EHInPs=1                   | ' 等待向指定位置的移动完成                              |
| 9 Mvs PUP                           | ' 向尺寸确认位置上空移动                               |

## 【说明】

- 1) 指定点位号码、速度，使多功能电动抓手向指定位置移动。  
通过梯形速度控制移动至指定位置。



- 2) 自动运行中，通过本指令使抓手动作开始后，直接转入下个指令的执行。（不等待抓手动作结束）
- 要等到抓手动作完成后再使机器人移动时，请参照以下的状态变量。参照方法如上述例子所示。
    - **M\_EHBusy**（抓手动作中）..... 抓手动作中为 "1"
    - **M\_EHInPs**（目标位置到达状态）..... 动作开始时为 "0"、到达指定位置时为 "1"
  - 抓手动作中，执行了抓手动作指令的插槽停止时，抓手动作也会暂停。  
插槽重新启动后，抓手动作也重新开始。但是，在插槽停止中时通过示教单元的手动操作等驱动对象抓手的情况下，抓手动作不会重新开始。
  - 要通过程序中途停止（结束）抓手动作时，请执行 **EHStop** 指令。
- 3) 单步运行或直接执行时，将等待直到通过本指令执行的抓手动作结束为止。
- 4) 如果指定相对量，可以使移动位置按相对量进行偏置。
- 5) 向移动位置移动途中因外力而停止时，会发生报警「多功能电动抓手模块报警（电机过负载）」。
- 6) 在多功能电动抓手动作中执行了本指令时，会发生报警"抓手动作指令重复执行"。有重复执行的可能性时，请参照状态变量 **M\_EHBusy**（抓手动作中）在确认是否为抓手动作中后，再执行本指令。
- 7) 控制器电源启动后，请在使多功能电动抓手进行原点回归后再执行本指令。  
不进行原点回归而执行本指令时，会发生报警。
- 8) 购买多功能电动抓手后，需要进行参数初始化（机种选择）。
- 9) 不支持继续功能（如果在动作中切断电源，处理将被初始化）。

## EHHold (抓手保持)

### 【功能】

实施多功能电动抓手的定位把持动作。

### 【格式】

```
EHHold □ < 抓手号码 >, <POINT No.>, <SPEED>, <FORCE>
```

### 【术语】

<抓手号码> ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。

注) 以实数的数值变量进行了指定时, 将在四舍五入后进行设定。

<POINT No.> ..... 指定多功能电动抓手的点位号码。以常数或数值变量进行指定。

设定范围为0~32。(注: 小数点以后四舍五入)

0为现在位置, 1~32将参数: EHNPOS1~32 (n: 抓手号) 中设定的值设定为目的位置。

<SPEED> ..... 以%单位的整数指定抓手的移动速度。

设定范围: 1 ~100 (%)

注) 如果设定速度小于20%, 抓手有可能会出出现振动。

<FORCE> ..... 以%单位的整数指定抓手的把持力。以常数或数值变量进行设定。

设定范围: 1 ~100 (%)

注) ①以实数的数值变量进行了指定时, 将在四舍五入后进行设定。

②如果设定值小于30%, 有可能会因为抓手自身的摩擦而导致抓手不动作。

例) 多功能电动抓手: ESG1-SS-2815-11XW107以10%以上的把持力进行动作。

### 【例子 1】

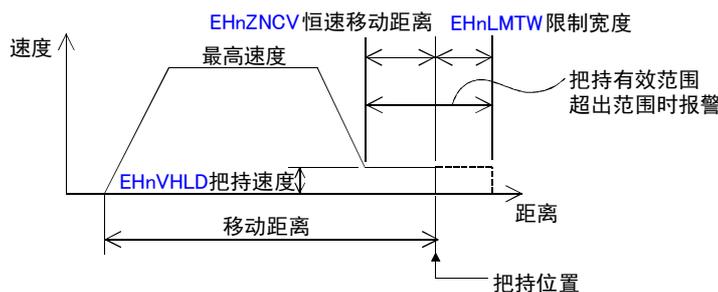
```
1 EHOOpen 1, 100, 30 ' 抓手 1 开 (速度 =100%, 力 =30%)
2 Mov PUP ' 向把持位置上空移动
3 Wait M_EHInPs=1 ' 确认向开端的移动完成
4 Mvs PGET ' 向把持位置移动
5 Fine 0.5, P ' 等待机器人的移动完成 (参照 EHOOpen/EHClose (抓手
 打开 / 抓手关闭) 的例子)
6 EHHold 1, 5, 100, 30 ' 抓手 1 定位把持 (点位号码 =5, 速度 =100%, 力 =30%)
7 Wait M_EHBusy=0 ' 动作完成确认
8 If M_EHHold=1 AND M_EHZone=1 Then ' 把持 & 区段确认
9 Mvs PUP ' 向把持位置上空移动
10 Endlf
```

### 【说明】

1) 指定点位号码、速度、力, 并实施多功能电动抓手的定位把持动作。

通过梯形速度控制移动到指定位置前, 在把持前以恒速进行动作。

把持前的恒速移动距离 (EHnZNCV) · 把持速度 (EHnVHLD) · 超过指定位置的有效距离 (限制宽度) (EHnLMTW) 通过参数进行设定。



- 2) 自动运行中，通过本指令使抓手动作开始后，直接转入下个指令的执行。（不等待抓手动作结束）
  - a) 要等到抓手动作完成后再使机器人移动时，请参照以下的状态变量。参照方法如上述例子所示。
    - **M\_EHBusy**（抓手动作中）..... 抓手动作中为 "1"
    - **M\_EHHold**（抓手动作中）..... 动作开始时为 "0"、行程途中进行把持时为 "1"
    - **M\_EHZone**（抓手动作中）..... 在指定领域（恒速移动距离 + 限制宽度内）中时为 "1"
    - **M\_EHInPs**（抓手动作中）..... 动作开始时为 "0"、到达目标位置时为 "1"
  - b) 抓手动作中，执行了抓手动作指令的插槽停止时，抓手动作也会暂停。  
插槽重新启动后，抓手动作也重新开始。  
但是，在插槽停止中时通过示教单元的手动操作等驱动对象抓手的情况下，抓手动作不会重新开始。
  - c) 要通过程序中途停止（结束）抓手动作时，请执行 **EHStop** 指令。
- 3) 单步运行或直接执行时，将等待直到通过本指令执行的抓手动作结束为止。
- 4) 在恒速移动距离和限制宽度范围内进行把持时，状态变量 **M\_EHHold** 和 **M\_EHZone** 将被设为 1。  
在恒速移动距离前进行把持时，仅状态变量 **M\_EHHold** 被设为 1。  
未能在限制宽度的范围内进行把持时，**M\_EHHold** 为 0 停止，**M\_EHInPs** 设为 1。
- 5) 在动作途中停止时，通过重启动作返回到移动开始位置后，再次实施定位把持动作。  
返回移动开始位置的动作作为 **EHMove** 指令（速度为通过 **EHHold** 指令指定的速度）动作。
- 6) 在多功能电动抓手动作中执行了本指令时，会发生报警 "抓手动作指令重复执行"。  
有重复执行的可能性时，请参照状态变量 **M\_EHBusy**（抓手动作中）确认是否为抓手动作中后，再执行本指令。
- 7) 控制器电源启动后，请在使多功能电动抓手进行原点回归后再执行本指令。  
不进行原点回归而执行本指令时，会发生报警。
- 8) 购买多功能电动抓手后，需要进行参数初始化（机种选择）。
- 9) 不支持继续功能（如果在动作中切断电源，处理将被初始化）。

## EHOrg (抓手原点)

### 【功能】

实施多功能电动抓手的原点回归。

### 【格式】

|                  |
|------------------|
| EHOrg □ < 抓手号码 > |
|------------------|

### 【术语】

< 抓手号码 > ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。  
注) 以实数的数值变量进行了指定时, 将在四舍五入后进行设定。

### 【例子 1】

```

1 If M_EHOrg=0 THEN ' 原点回归未完成状态
2 EHOrg 1 ' 抓手 1 原点回归
3 Wait M_EHOrg=1 ' 等待抓手 1 原点回归完成
4 EndIf
5 EHOrgOpen 1, 100, 30 ' 抓手 1 开 (速度 =100%, 力 =30%)

```

### 【说明】

- 1) 使多功能电动抓手进行原点回归。  
控制器电源启动后, 在进行多功能电动抓手的移动动作前, 需要进行原点回归。
- 2) 自动运行中, 通过本指令使抓手动作开始后, 直接转入下个指令的执行。(不等待抓手动作结束)
  - a) 请参照以下的状态变量进行抓手的动作完成确认。
    - M\_EHBusy (抓手动作中) ..... 抓手动作中为 "1"
    - M\_EHInPs (目标位置到达状态) ..... 动作开始时为 "0"、原点回归完成时为 "1"
    - M\_EHOrg (抓手原点回归完成状态) .... 原点回归完成时为 "1"
  - b) 抓手动作中, 执行了抓手动作指令的插槽停止时, 抓手动作也会暂停。  
插槽重新启动后, 抓手动作也重新开始。  
但是, 在插槽停止中时通过示教单元的手动操作等驱动对象抓手的情况下, 抓手动作不会重新开始。
  - c) 要通过程序中途停止 (结束) 抓手动作时, 请执行 EHStop 指令。
- 3) 进行原点回归时, 无法再现上一次的原点回归位置的情况下, 将发生报警「多功能电动抓手模块报警 (编码器 Z 相位置偏离)」。  
此时, 请在排除位置偏离的原因后再次实施原点回归。  
因指尖更换等导致原点位置发生变化的情况下, 请实施多功能电动抓手的参数初始化 (机种选择) 操作。
- 4) 在多功能电动抓手动作中执行了本指令时, 会发生报警「抓手动作指令重复执行」。  
有重复执行的可能性时, 请参照状态变量 M\_EHBusy (抓手动作中) 确认是否为抓手动作中后, 再执行本指令。
- 5) 购买多功能电动抓手后, 需要进行参数初始化 (机种选择)。
- 6) 不支持继续功能 (如果在动作中切断电源, 处理将被初始化)。
- 7) 不使用本指令而通过示教单元等进行原点回归时, 如果预先在程序的起始处记述如下所示的原点回归完成确认功能, 可以在原点回归未完成时事先发生报警。

原点回归完成确认功能 (例: 如果原点回归未完成, 将发生报警 9100)

```

1 If M_EHOrg = 0 THEN '
2 Error 9100
3 EndIf

```

补充: 发生的报警事先通过第 428 页的「UER1 ~ UER20」参数进行设定。



## 注意

在多功能电动抓手的原点设定动作中, 请勿驱动机器人。如果驱动机器人, 有可能会因为振动等的影响导致原点设定未正常完成。

## EHServo On/Off (hand servo)

### 【功能】

对多功能电动抓手的伺服电机电源的 ON/OFF 进行控制。

### 【格式】

EHServo □ < 抓手号码 >, <On/Off>

### 【用语】

<抓手号码> ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。  
注)以实数的数值变量指定时,四舍五入进行设定。

<On/Off> ..... On: 接通多功能电动抓手的伺服电机的电源的情况  
Off: 切断多功能电动抓手的伺服电机的电源的情况

### 【例文】

|                   |                                                           |
|-------------------|-----------------------------------------------------------|
| 1 EHMov 1, 1, 100 | '抓手 1 移动 (要点号码 =1、速度 =100%)                               |
| 2 Mov PUP         | '向夹持位置上空移动                                                |
| 3 Wait M_EHBusy=0 | '至指定位置的移动结束确认                                             |
| 4 EHServo 1, Off  | '抓手 1 伺服 OFF                                              |
| 5 Mvs P1          | '抓手 1 的卡爪及工件为锥形,卡爪接触到工件的情况下,抓手 1 的卡爪会仿照工件进行动作。(仅限单凸轮·三爪型) |

### 【说明】

- 1) 对多功能电动抓手的伺服电机电源进行控制。  
※ 执行 EHOpen/EHClose/EHMov/EHHold/EHOrg 指令时,将自动接通伺服电机的电源,因此没有必要执行 EHServo On。  
※ 若切断了伺服电机的电源,则可通过外部力量移动可动部位。(仅限单凸轮·三爪型)
- 2) 本指令在伺服电机电源的 ON/OFF 结束后转至执行下一个指令。
- 3) 如在多功能电动抓手动作过程中执行本指令,则有可能发生“抓手动作指令双重执行”错误。该情况下,应参照状态变量 M\_EHBusy(抓手动作中),确认抓手为非动作中之后,再执行本指令。

## EHStop (抓手停止)

### 【功能】

使多功能电动抓手停止。

### 【格式】

|                   |
|-------------------|
| EHStop □ < 抓手号码 > |
|-------------------|

### 【术语】

< 抓手号码 > ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。  
注) 以实数的数值变量进行了指定时, 将在四舍五入后进行设定。

### 【例子 1】

```

1 EHClose 1, 100, 30 ' 抓手 1 闭 (速度 =100%, 力 =30%)
2 *LOOP: If M_In (20) =1 Then EHStop' 输入 20 号 (传感器情报等) 开启时, 中途停止抓手动作
3 If M_EHBusy=1 Then Goto *LOOP ' 动作中为环路
4 Mov PUP ' 向上空位置移动

```

### 【说明】

- 1) 使多功能电动抓手停止。  
使通过 EHOpen/EHClose/EHMove/EHHold/EHOrg 指令执行的动作停止。
- 2) 购买多功能电动抓手后, 需要进行参数初始化 (机种选择)。
- 3) 如果在执行本指令后立即执行抓手动作指令 (EHOpen 等), 有可能会发生报警「抓手动作指令重复执行」。这种情况下, 请参照状态变量 M\_EHBusy (抓手动作中) 确认是否为抓手动作中后, 再执行抓手动作指令。

## EHTbl (Hand Table)

### 【功能】

以指定的 Table 号码的条件驱动多功能电动抓手。

### 【格式】

```
EHTbl □ < 抓手号码 >, <TABLE No.>
```

### 【术语】

< 抓手号码 > ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量进行指定。

<TABLE No.> ..... 指定多功能电动抓手的Table号码。

指定范围为1~32。

参数被设定为EHnTBL1~32中设定的Table情报。

### 【例子 1】

```

1 EHTbl 1, 1 ' 抓手 1 移动 (点位号码 =5, 速度 =100%, 相对量 =-3.0mm)
2 Mov PUP ' 向把持位置上空移动
3 Wait M_EHInPs=1 ' 确认向开端的移动完成
4 Mvs PGET ' 向把持位置移动
5 Fine 0.5, P ' 等待移动完成 (Dly 指令也可以)
6 EHTbl 1, 2 ' 抓手 1 定位把持 (点位号码 =5, 速度 =100%, 力 =30%)
7 Wait M_EHBusy=0 ' 动作完成确认
8 If M_EHHold=1 AND M_EHZone=1 Then ' 把持 & 区段确认
9 Mvs PUP ' 向把持位置上空移动
10 EndIf

```

#### < 参数设定 >

```

EH1TBL1=3.00, 5.00, -3.00, 100.00, 0.00, 0.00, 0.00
EH1TBL2=4.00, 5.00, 0.00, 100.00, 30.00, 0.00, 0.00

```

### 【说明】

1) 使多功能电动抓手进行 Table 指定执行。

根据参数 EHnTBL1 ~ 32 中设定的 Table 情报 (动作模式、点位号码、相对量、力、区段 ON 位置、区段 OFF 位置) 来驱动多功能电动抓手。

2) 自动运行中, 通过本指令使抓手动作开始后, 直接转入下个指令的执行 (不等待抓手动作结束)。请参照以下的状态变量进行抓手的动作完成确认。

< 动作模式: 开、闭时 >

- M\_EHBusy (抓手动作中) ..... 抓手动作中为 "1"
- M\_EHInPs (目标位置到达状态) ..... 动作开始时为 "0"、移动完成时为 "1"
- M\_EHHold (抓手把持状态) ..... 动作开始时为 "0"、行程途中进行把持时为 "1"

## (4) 状态变量的详细说明

各状态变量的详细内容按不同功能如下所示。

**M\_EHPos1/M\_EHPos2/M\_EHPos3****【功能】**

参照多功能电动抓手 1 ~ 3 的点位数据进行设定。

**【格式】**

|                                  |           |
|----------------------------------|-----------|
| < 数值变量 >=M_EHPos1 (<POINT No.>)  | ' 点位数据的参照 |
| M_EHPos1 (<POINT No.> )=< 数值数据 > | ' 点位数据的设定 |

**【用语】**

< 抓手号码 > ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量指定。

注) 通过实数的数值变量指定时, 四舍五入以进行设定。

< 数值变量 > ..... 指定要代入点位数据的变量。

点位数据值为-9999.99 ~ 9999.99 (mm)。

< 数值数据 > ..... 指定点位数据。以常数或数值变量指定。设定值以0.01mm为单位进行指定。

设定范围为-9999.99 ~ 9999.99 (mm)。(注: 0.01mm以后将四舍五入)

**【例文】**

1 MPos=M\_EHCur ' 代入抓手现在位置

2 MPos=MPos+0.5 ' 计算现在位置 +0.5mm 的位置

3 M\_EHPos1 (5) =MPos' 设定到点位 5

4 EHMov 1, 5, 100 ' 抓手 1 向点位 5 的位置移动 (点位号码 =5, 速度 =100%)

5 Wait M\_EHInPs=1 ' 确认到达指定位置

**【说明】**

- 1) 指定点位号码, 并设定 • 参照多功能电动抓手的点位数据 (参数 EHnPOS1 ~ EHnPOS32、 n : 抓手号码)。

## M\_EHCur

**【功能】**

返回多功能电动抓手的现在位置。

**【格式】**

```
<数值变量>=M_EHCur [(<抓手号码>)]
```

**【用语】**

<数值变量> ..... 指定要代入抓手现在位置的变量。

<抓手号码> ..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量指定。（省略时为抓手1）

注）通过实数的数值变量指定时，四舍五入以进行设定。

**【例文】**

- 1 MPos=M\_EHCur           ' 代入抓手现在位置
- 2 MPos=MPos+0.5         ' 计算现在位置 +0.5mm 的位置
- 3 M\_EHPos1 (5) =MPos' 设定到点位 5
- 4 EHMov 1, 5, 100       ' 抓手 1 向点位 5 的位置移动（点位号码 =5, 速度 =100%）
- 5 Wait M\_EHInPs=1       ' 确认到达指定位置

**【说明】**

- 1) 指定抓手号码，并返回多功能电动抓手的现在位置。
- 2) 现在位置的范围为 -9999.99 ~ 9999.99（mm）。
- 3) 未连接对象抓手时的值为 0。
- 4) 对象抓手未进行原点回归时的值不确定。

M\_EHBusyM\_EHInPsM\_EHHoldM\_EHZoneM\_EHOrg**【功能】**

返回多功能电动抓手状态。

- M\_EHBusy..... 动作状态 [1 : 动作执行中 / 0 : 无动作执行]  
 M\_EHInPs..... 目标位置到达状态 [1 : 到达目标位置 / 0 : 未到达]  
 M\_EHHold..... 把持状态 [1 : 检测到把持 / 0 : 未检测到]  
 M\_EHZone..... 指定领域内 [1 : 在指定领域内 / 0 : 不在]  
 M\_EHOrg..... 原点回归状态 [1 : 原点回归完成 / 0 : 未完成]

**【格式】**

<数值变量>=M\_EHOrg [ (<抓手号码>) ] ※ 其他状态变量也相同。

**【用语】**

- <数值变量>..... 指定要代入多功能电动抓手状态的变量。  
 <抓手号码>..... 以整数指定多功能电动抓手号码1~3。以常数或数值变量指定。（省略时为抓手1）  
 注）通过实数的数值变量指定时，四舍五入以进行设定。

**【例文】**

```

1 If M_EHOrg=0 THEN ' 原点回归未完成状态
2 EHOrg 1 ' 抓手 1 原点回归
3 Wait M_EHOrg=1 ' 等待抓手 1 原点回归完成
4 EndIf
5 EHOOpen 1, 100, 20 ' 抓手 1 开 （速度 =100%, 力 =20%）
6 Mov PUP ' 向把持位置上空移动
7 Wait M_EHInPs=1 ' 确认向开端的移动完成
8 Mvs PGET ' 向把持位置移动
9 Fine 0.5, P ' 等待机器人的移动完成 （参照 EHOOpen/EHClose （抓
 ' 手打开 / 抓手关闭）的例子）
10 EHClose 1, 50, 20, 20.00 22.00 ' 关闭抓手 1 （速度 =50%, 力 =20%, 区段位置 =20 ~
 ' 22mm）
11 Wait M_EHBusy=0 ' 动作结束确认
12 If M_EHHold=1 AND M_EHZone=1 Then ' 把持 & 区段确认
13 Mvs PUP ' 向把持位置上空移动
14 EndIf

```

**【说明】**

- 1) 指定抓手号码，并返回多功能电动抓手的各种状态。
- 2) 未连接对象抓手时为 0。
- 3) 各状态变量的说明如下所示。

表 7-11：多功能电动抓手的状态变量

| No | 状态变量     | 说明                                                                                                                                         |
|----|----------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | M_EHBusy | 动作状态[1: 动作执行中/0: 无动作执行]。<br>抓手动作处理中时设定为"1"。                                                                                                |
| 2  | M_EHInPs | 目标位置到达状态[1: 到达目标位置/0: 未到达]。<br>抓手动作开始时为"0", 到达目标位置(定位完成距离范围)后变为"1"。在目标位置期间设定为"1"。                                                          |
| 3  | M_EHHold | 把持状态[1: 检测到把持/0: 未检测到]。<br>抓手动作开始时为"0", 动作结束时如果以指定把持力进行了把持, 则变为"1"。在以指定把持力进行把持期间设定为"1"。                                                    |
| 4  | M_EHZone | 指定领域内[1: 在指定领域内/0: 不在]。<br>抓手动作开始时为"0", 进入指定领域后变为"1"。<br>在指定领域内期间设定为"1"。                                                                   |
| 5  | M_EHOrg  | 原点回归状态[1: 原点回归完成/0: 未完成]。<br>如果抓手原点回归动作正常完成, 则设定为"1"。<br>控制器电源启动后, 如果不进行原点回归, 则无法执行各种动作指令。请通过本状态变量确认原点回归完成状态, 在原点回归未完成的情况下执行原点回归指令(EHOrg)。 |

## M\_EHTMd1/M\_EHTMd2/M\_EHTMd3

### 【功能】

返回多功能电动抓手指定的 Table 数据的动作模式。

### 【格式】

```
<数值变量>=M_EHTMd1 [(<TABLE No.>)]' 参照 Table 数据的动作模式
```

### 【用语】

<数值变量> ..... 指定要代入 Table 数据的动作模式 (0~5) 的变量。

0: 未设定、1: 移动、2: 把持、3: 相对移动、4: 开 (Open)、5: 闭 (Close)

<TABLE No.> ..... 指定多功能电动抓手的 Table 号码。

设定范围为 1~32。通过常数或变量进行设定。

### 【例文】

```
M1=M_EHTMd1 (1) '将抓手 1 的 Table 数据 1 的动作模式代入 M1
```

```
M11=M_EHTMd2 (11) '将抓手 2 的 Table 数据 11 的动作模式代入 M11
```

```
If M11<>0 Then EHTbl 2, 11 '使抓手 2 按 Table 数据 11 的内容动作
```

### 【说明】

- 1) 指定抓手号码, 并设定 • 参照多功能电动抓手的 Table 数据 (参数 EHNtbl1 ~ EHNtbl32、n : 抓手号码) 的动作模式。

## M\_EHOPos

### 【功能】

返回多功能电动抓手的移动完成点位号码。

### 【格式】

```
<数值变量>=M_EHOPos [(<抓手号码>)]
```

### 【用语】

<数值变量> ..... 指定要代入移动完成点位号码 (1~32) 的变量。

### 【例文】

M1=MEHOPos (1) '将抓手 1 的完成点位号码代入 M1'  
If M\_EHOPos (2) =2 Then \*L01' 如果抓手 2 的完成点位号码为 2, 则跳转到 \*L01

### 【说明】

- 1) 指定抓手号码, 并返回多功能电动抓手的移动完成点位号码。
- 2) 开始执行 EHOrg/EHOpen/EHClose/EHMOV/EHHold 指令时为 0。
- 3) EHMov 的动作完成后通过 M\_EHInPs=1 输出点位号码。
- 4) EHHold 的动作完成后通过 M\_EHHold=M\_EHZone=1 或 M\_EHInPs=1 输出点位号码。
- 5) EHHold 的把持有效范围外的动作完成后的输出保持 0 不变。此外, 在将所把持的工件取下的状态下执行把持动作而变为 M\_EHInPs=1 的情况下, 输出保持 0 不变。
- 6) 执行 EHMov/EHHold 指令后, 移动中发生报警或中断的情况下, 输出保持 0 不变。
- 7) 未连接对象抓手时为 0。

## 7.2.9 参数一览

表 7-12 所示为参数一览。

表 7-12: 多功能电动抓手控制参数一览

| 参数             | 参数名 <sup>注1)</sup> | 排列数<br>文字数 | 内容说明                                                                                                         | 出货时设定值 |
|----------------|--------------------|------------|--------------------------------------------------------------------------------------------------------------|--------|
| 多功能电动抓手执行器类型   | EHnTYPE            | 整数 1       | 实施并存储多功能电动抓手本体的类型 (机种) 选择设定范围为 0~65535。<br>写入本参数时, 在实施了所连接的多功能电动抓手的参数初始化 (机种选择) 操作后, 将存储对象抓手的参数初始值。          | 0      |
| 多功能电动抓手软限制 (+) | EHnLMP             | 整数 1       | 设定多功能电动抓手的正向可动范围。<br>设定范围为 0~9999 (0.1mm 单位)。<br>将原定回归方向设定为打开侧时, 则关闭侧为 + 方向。<br>将原定回归方向设定为关闭侧时, 则打开侧为 + 方向。  | 0      |
| 多功能电动抓手软限制 (-) | EHnLTM             | 整数 1       | 设定多功能电动抓手的负向可动范围。<br>设定范围为 -9999~0 (0.1mm 单位)。<br>将原定回归方向设定为打开侧时, 则关闭侧为 + 方向。<br>将原定回归方向设定为关闭侧时, 则打开侧为 + 方向。 | 0      |
| 多功能电动抓手行程      | EHnSTRK            | 整数 1       | 设定多功能电动抓手的执行器的行程。<br>设定范围为 0~9999 (0.1mm 单位)。                                                                | 0      |
| 多功能电动抓手定位完成距离  | EHnPSCD            | 整数 1       | 向指定的点位移动时, 在到达指定距离前完成移动。<br>设定范围为 1~9999 (0.01mm 单位)。<br>要提升系统的节拍时间时可增大该值。                                   | 0      |
| 多功能电动抓手加速度     | EHnACC             | 整数 1       | 设定执行器的加速度。<br>将根据执行器类型和可动部重量的设定自动设定最佳的加速度。<br>请在考虑到执行器设定部和指尖部的刚性等而需要调低加速度的情况下进行变更。<br>设定范围为 1~100 (%)。       | 0      |

| 参数                     | 参数名 <sup>注1)</sup>    | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 出货时设定值        |
|------------------------|-----------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 多功能电动抓手最<br>高速度        | EHnVMAX               | 整数 1       | 设定最高速度。<br>设定范围为1~100(%)。                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0             |
| 多功能电动抓手把<br>持速度        | EHnVHLD               | 整数 1       | 设定定位把持动作(通过EHHold指令进行的动作)的把持<br>有效范围内的恒速速度。<br>设定范围为1~100(%)。                                                                                                                                                                                                                                                                                                                                                                                                                  | 0             |
| 多功能电动抓手恒<br>速移动距离      | EHnZNCV               | 整数 1       | 设定定位把持动作(通过EHHold指令进行的动作)的恒速<br>移动时的距离。<br>仅以距离目标位置相当于恒速移动距离的地方开始变为把<br>持速度。<br>设定范围为1~9999(0.01mm单位)。                                                                                                                                                                                                                                                                                                                                                                         | 0             |
| 多功能电动抓手限<br>制宽度        | EHnLMTW               | 整数 1       | 设定定位把持动作(通过EHHold指令进行的动作)的恒速<br>移动时的限制宽度。<br>以把持速度从目标位置开始仅移动相当于限制宽度的距<br>离。<br>超出把持有效范围的情况下,在移动量+限制宽度处停<br>止。<br>设定范围为1~9999(0.01mm单位)。                                                                                                                                                                                                                                                                                                                                        | 0             |
| 多功能电动抓手原<br>点回归方向      | EHnORGD               | 整数 1       | 设定原点回归方向。设定值为0: 打开侧/1: 关闭侧。<br>原点回归方向设定为「0: 打开侧」时,位置设定为+且向<br>关闭方向移动。                                                                                                                                                                                                                                                                                                                                                                                                          | 0             |
| 多功能电动抓手原<br>点回归速度      | EHnORGV               | 整数 1       | 设定原点回归速度。设定范围为20~50(%)。                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0             |
| 多功能电动抓手原<br>点移位        | EHnORGSF              | 整数 1       | 在原点回归完成的坐标位置上数据将发生相当于通过该参<br>数设定的值的移位。<br>设定范围为-9999~9999(0.01mm单位)。<br>因装置维护等而发生了位置偏离的情况下,通过设定该参<br>数,可以省略重新输入点位数据的步骤。请勿输入比较限<br>制更大的值。                                                                                                                                                                                                                                                                                                                                       | 0             |
| 多功能电动抓手原<br>点回归方式      | EHnORGS               | 整数 1       | 设定原点回归的方法。<br>设定值为[0: 行程端/1: 行程端+Z相检测方式]<br>行程端检测后,手指将反转直到Z相检测为止。<br>到Z相检测为止的反转距离成为问题的情况下,通过变更<br>为行程端检测,反转距离将变为0.5mm。                                                                                                                                                                                                                                                                                                                                                         | 0             |
| 多功能电动抓手点<br>位数据        | EHnPOS1<br>~ EHnPOS32 | 整数 1       | 设定(存储)多功能电动抓手的点位数据。<br>设定范围为-999.99~999.99(0.01mm单位)。                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             |
| 多功能电动 Hand<br>Table 数据 | EHnTBL1<br>~ EHnTBL32 | 实数 7       | 设定多功能电动抓手的Table数据。<br>第1要素: 动作模式<br>(0: 未设定、1: 移动、2: 把持、3: 相对移<br>动、4: 开(Open)、5: 闭(Close))<br>第2要素: 点位号码<br><设定范围> 动作模式/1: 0~32/2: 1~32<br>第3要素: 相对量[mm]<br><设定范围> -9999.99~9999.99<br>第4要素: 速度[%]<br><设定范围> 动作模式/4,5: 1~50/1,2,3: 1~<br>100<br>第5要素: 力[%]<br><设定范围> 1~100<br>第6要素: 区段ON位置[mm]<br>※仅动作模式为开(EHOpen)、闭(EHClose)时<br>有效<br><设定范围> -9999.99~9999.99<br>第7要素: 区段OFF位置[mm]<br>※仅动作模式为开(EHOpen)、闭(EHClose)时<br>有效<br>※第6、7要素均为"0"时,区段指定无效<br><设定范围> -9999.99~9999.99 | 0,0,0,0,0,0,0 |

| 参数                  | 参数名 <sup>注1)</sup> | 排列数<br>文字数 | 内容说明                                                                                  | 出货时设定值 |
|---------------------|--------------------|------------|---------------------------------------------------------------------------------------|--------|
| 多功能电动抓手<br>示教单元动作模式 | EHOPEMD            | 整数 1       | 设定多功能电动抓手的示教单元动作模式的初始值。<br>0：点位指定<br>1：Table指定                                        | 0      |
| 多功能电动抓手<br>现在位置获取模式 | EHCURMD            | 整数 1       | 设定多功能电动抓手的现在位置获取模式。<br>0：始终获取（向上兼容）<br>1：Manual模式时为始终获取。<br>Automati c模式时在动作结束时仅获取1次。 | 0      |

注 1) n：抓手号码（1～3）。

## 7.2.10 报警一览

表 7-13 所示为与多功能电动抓手有关的报警一览（原因和对策）。

通过报警号码的前 3 位区别对象的多功能电动抓手号码。

- H814\*：抓手 1
- H815\*：抓手 2
- H816\*：抓手 3

表 7-13：报警一览表

| 报警号码                    | 报警发生原因和对策 |                                                                                                                               |
|-------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------|
| H8140<br>H8150<br>H8160 | 报警信息      | 多功能电动抓手远程 I/O 通信端口异常                                                                                                          |
|                         | 原因        | 多功能电动抓手远程 I/O 通信中发生了 CRC 报警或连接报警。                                                                                             |
|                         | 对策        | 请检查通信电缆或确认连接机器的电源。                                                                                                            |
| L8141<br>L8151<br>L8161 | 报警信息      | 发生多功能电动抓手报警                                                                                                                   |
|                         | 原因        | 多功能电动抓手中发生了以下任意一项报警。<br>电机过载、电机过电流、编码器 Z 相位置偏离、电源电压过低、位置偏差溢出、反馈报警 1、2 和 3、电压异常、系统异常 1 和 2。                                    |
|                         | 对策        | 请参照多功能电动抓手的使用说明书排除报警原因后，实施机器人的报警复位操作。<br>无法通过报警复位操作解除报警的情况下，请重新接通机器人的电源。                                                      |
| L8142<br>L8152<br>L8162 | 报警信息      | 发生多功能电动抓手报警                                                                                                                   |
|                         | 原因        | 多功能电动抓手中发生了以下任意一项报警。<br>软限制超出、原点回归未完成、驱动电源 OFF、互锁时执行移动指令、运行中输入执行指令、数据写入过程中输入指令、原点回归时未发现 Z 相、FCS 确认报警、超出数据输入范围、执行器类型错误、内部通信异常。 |
|                         | 对策        | 请参照多功能电动抓手的使用说明书排除报警原因后，实施机器人的报警复位操作。无法通过报警复位操作解除报警的情况下，请重新接通机器人的电源。<br>通过程序确认原点回归的完成状态的例子见第 549 页的「操作注意事项」。                  |
| L8143<br>L8153<br>L8163 | 报警信息      | 未连接多功能电动抓手                                                                                                                    |
|                         | 原因        | 没有连接多功能电动抓手。                                                                                                                  |
|                         | 对策        | 请确认远程 I/O 的通信电缆的连接或确认站号的设定。                                                                                                   |
| L8144<br>L8154<br>L8164 | 报警信息      | 多功能电动抓手机种选择未完成。                                                                                                               |
|                         | 原因        | 没有完成多功能电动抓手机的机种选择。                                                                                                            |
|                         | 对策        | 请设定机器人的参数：执行器类型（EHnTYPE）。                                                                                                     |
| L8145<br>L8155<br>L8165 | 报警信息      | 多功能电动抓手原点回归未完成。                                                                                                               |
|                         | 原因        | 没有完成多功能电动抓手的原点回归。                                                                                                             |
|                         | 对策        | 请执行原点回归。<br>通过程序确认原点回归的完成状态的例子见第 549 页的「操作注意事项」。                                                                              |
| L8146<br>L8156<br>L8166 | 报警信息      | 多功能电动抓手响应超时。                                                                                                                  |
|                         | 原因        | 没有来自多功能电动抓手的响应。                                                                                                               |
|                         | 对策        | 请确认电缆的连接状态、多功能电动抓手状态并重新接通控制器的电源。                                                                                              |

| 报警号码                    | 报警发生原因和对策 |                                                         |
|-------------------------|-----------|---------------------------------------------------------|
| L8147<br>L8157<br>L8167 | 报警信息      | 多功能电动抓手动作指令重复执行。                                        |
|                         | 原因        | 重复执行了多功能电动抓手的动作指令。                                      |
|                         | 对策        | 请修正程序，避免重复执行动作指令。                                       |
| L8148<br>L8158<br>L8168 | 报警信息      | 多功能电动抓手 n 点位登录重复执行。                                     |
|                         | 原因        | 多任务程序中的设置或 T/B 与 RT ToolBox2 等的操作中重复进行了点位数据登录。          |
|                         | 对策        | 请修正程序或修改操作步骤。                                           |
| L8149<br>L8159<br>L8169 | 报警信息      | 多功能电动抓手系统异常 3。                                          |
|                         | 原因        | 多功能电动抓手发生了系统异常。                                         |
|                         | 对策        | 请在报警复位后执行原点回归。报警无法复位时，请重新接通电源。再次发生时请向生产厂商咨询。            |
| L0092                   | 报警信息      | 该信号正为电动抓手所使用。                                           |
|                         | 原因        | 无法使用电动抓手用的输出信号。                                         |
|                         | 对策        | 连接有多功能电动抓手的情况下，多功能电动抓手所使用的输出信号无法通过程序输出。请变更程序中使用的输出信号号码。 |

## 7.3 多重抓手的使用方法

下面将对将 IDEC 公司制造的多重抓手连接到机器人的 RIO 上使用时的设定及操作方式进行说明。请一并参照多重抓手产品附带的使用说明书。

### 7.3.1 概要

#### (1) 设备构成（例）

以下所示为设备的构成示例。



注1) 图示为 RV-2F 的构成示例。

RV-2F 为标准规格的机器人，可以使用多重抓手。

其他机器人为特殊规格。（仅限垂直多关节 6 轴型）请参阅另外发行的交货规格书。

注2) 多重抓手本体图记载的仅为一个示例。关于构成多重抓手的附件以及规格等，请参阅多重抓手附带的使用说明书。

注3) 为方便参数的设定操作，请使用 R56TB/R57TB 或 RT ToolBox2。通过 R32TB/R33TB 也可以方便地进行 JOG 操作、抓手开闭操作。

图 7-9：使用多重抓手时的构成示例

#### 【参考：多重抓手本体与机器人可控制的抓手（手指）的关系】

多重抓手本体可以分成基座单元和各抓手（手指）来考虑。基座单元具有通用性，但各抓手（手指）的形状通常因工件而异。

本机器人具备每次均将机器人的控制点变更为为抓手（手指）的前端（作业位置）的功能，使得能够按此时成为作业对象的抓手（手指）基准进行动作，例如按各抓手（手指）基准前进/后退等。

用户进行的 JOG 操作或自动运行中指定了工具号码后，控制点变更为指定的工具号码所对应的抓手（手指）的前端。（控制点的初始值为机械 I/F 的中心）

为了有效运用该功能，将预先在参数中设定各抓手（手指）与机械 I/F 中心的位置关系。（工具号码所对应的抓手（手指）的关联）要设定的参数的详细内容请参照第 581 页的「(1) 参数的设定」。

#### (2) 可使用的机器和软件版本

多重抓手可以通过表 7-14 所示的机器人和机器、工具以及对象的软件版本来使用。

表 7-14：可使用的机器和软件版本

| 机器       | 控制器   | R32TB/R33TB | R56TB/R57TB | RT ToolBox2 |
|----------|-------|-------------|-------------|-------------|
| 可使用的软件版本 | R3 以后 | Ver. 1.6 以后 | Ver. 2.5 以后 | Ver. 2.1 以后 |

### 7.3.2 操作步骤

以下所示为操作步骤的概要。



#### (1) 参数的设定

通过 T/B 或 RT ToolBox2 设定参数。关于操作方法的详细内容，请参阅各产品附带的使用说明书。

- R32TB/R33TB：本书 [7.3.2 操作步骤]
- R56TB/R57TB：R56TB/R57TB 使用说明书（BFP-A8591）
- RT ToolBox2：RT ToolBox2 / RT ToolBox2 mini 使用说明书（BFP-A8617）

#### 1) 抓手种类の設定

通过设定参数：抓手种类（HANDTYPE），可以从示教单元进行抓手操作或通过 HOpen/HClose 指令像通常的抓手一样进行操作。

通过 RIO 连接多重抓手时，为了使用通道 1 的站号 5，输入输出信号号码将如下所示。

- 输入信号号码..... 716 ~ 731（16 点）
- 输出信号号码..... 716 ~ 723（8 点）

注）700 ~ 715 号为系统保留领域，因此输入输出均不可使用。

此外，输入信号号码由多重抓手产品的规格决定。

设定示例）使用双线螺管且通过以下的输出信号进行抓手开闭时的设定示例如下所示

- 抓手 1 开 / 闭信号号码：716/717
- 抓手 2 开 / 闭信号号码：718/719
- 抓手 3 开 / 闭信号号码：720/721
- 抓手 4 开 / 闭信号号码：722/723

参数：HANDTYPE 的设定

HANDTYPE = D716, D718, D720, D722, , , ,

表 7-15：抓手种类设定参数

| 参数   | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                  | 出货时设定值                  |
|------|----------|------------|-------------------------------------------------------------------------------------------------------|-------------------------|
| 抓手种类 | HANDTYPE | 字符串 8      | 设定单/双线螺管的抓手种类和输出信号号码（D：双线螺管、S：单线螺管）。在抓手种类后设定信号号码。D900 的情况下，输出信号至信号号码900和901。D（双线螺管）的情况下，设定时请注意避免信号重叠。 | D900,D902,D904,D906,... |

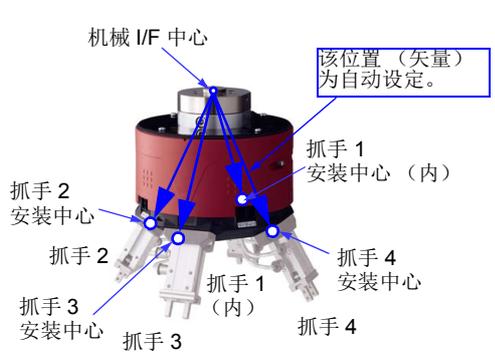
## 2) 抓手基座的型号设定

将所使用的多重抓手的基座的型号设定到参数：MHBnTYPE (n：基座号码1~8) 中。以该设定值为基础，自动设定参数：多重抓手基座基准坐标 (MHBnHCy。表 7-17)。

表 7-16：抓手基座设定参数

| 参数       | 参数名                         | 排列数<br>文字数 | 内容说明                                                                                                                                                                                         | 出货时设定值 |
|----------|-----------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 多重抓手基座型号 | MHBnTYPE<br>(n：基座号码<br>1~8) | 字符串 16     | 将所使用的多重抓手基座的型号分配(设定)到基座号码<br>1~8。<br>可同时登录8种型号。<br><br>【以下为出货时设定值】<br><多重抓手对应机种(包括垂直6轴特殊规格)><br>MHB1TYPE=MH1A-4NBY-*<br>MHB2TYPE=MHB8TYPE=未设定<br><上述以外的机种(包括使用者机器)><br>MHB1TYPE~MHB8TYPE=未设定 | 参照左栏   |

表 7-17：抓手基准坐标设定参数

| 参数         | 参数名                                          | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 出货时设定值 |
|------------|----------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 多重抓手基座基准坐标 | MHBnHCy<br>(n：基座号<br>码1~8、<br>y：抓手号码<br>1~4) | 实数 6       | 表示所使用的每个多重抓手基座的(每个基座号码)的各个抓手(手指)安装中心位置。(距离机械I/F中心的相对位置)<br>每个多重抓手基~座具有4个抓手(手指)安装位置。<br><br>【以下为出货时设定值】<br><多重抓手对应机种(包括垂直6轴特殊规格)><br>MHB1HC1=(0.00,+51.41,+107.41,0.00,+30.00,+90.00)<br>MHB1HC2=(+51.41,0.00,+107.41,0.00,+30.00,0.00)<br>MHB1HC3=(0.00,-51.41,+107.41,0.00,+30.00,-90.00)<br>MHB1HC4=(-51.41,0.00,+107.41,0.00,+30.00,+180.00)<br>MHB2HC1~MHB8HC4=(0.00,0.00,0.00,0.00,0.00,0.00)<br><上述以外的机种(包括使用者机器)><br>MHB1HC1~MHB8HC4=(0.00,0.00,0.00,0.00,0.00,0.00)<br><br> | 参照左栏   |

## 3) 多重抓手基座的分配(设定)

将所使用的多重抓手基座号码分别分配到可同时使用的4种多重抓手中。

表 7-18：抓手种类设定参数

| 参数       | 参数名                         | 排列数<br>文字数 | 内容说明                                                                                                                                       | 出货时设定值 |
|----------|-----------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 多重抓手基座号码 | MHnBNO<br>(n：多重抓手<br>号码1~4) | 整数 1       | 通过多重抓手号码n分配所使用的"多重抓手基座号码"(相当于参数MHBnTYPE的"n")。<br>设定值：0~8<br>0：设为未设定的状态。<br>1~8：指定要分配的多重抓手基座号码。<br><br>该设定值与参数：多重抓手偏置量(MHnOFSy)的各多重抓手号码相对应。 | 0      |

4) 抓手偏置量的设定

为 4 种抓手（手指）分别设定从其安装位置到作业位置（应作为控制点的位置）为止的距离。该设定值与通过参数：多重抓手基座型号中指定的多重抓手基座基准坐标相关联，变为 TOOL 坐标（从机械 I/F 到应作为控制点位置的位置关系）。

表 7-19： 抓手偏置量设定参数

| 参数      | 参数名                                    | 排列数<br>文字数 | 内容说明                                                                                            | 出货时设定值           |
|---------|----------------------------------------|------------|-------------------------------------------------------------------------------------------------|------------------|
| 多重抓手偏置量 | MHnOFSy<br>(n: 多重抓手号码 1~4、y: 抓手号码 1~4) | 实数 3       | 对多重抓手号码1~4, 分别按x、y、z设定抓手基座上安装的4个抓手(手指)的偏置量。(从抓手(手指)的安装位置到作业位置的距离)<br>设定值以多重抓手基座基准坐标系为基准。单位: mm。 | 0.00, 0.00, 0.00 |

【补充】：使用 R56TB/R57TB 或 RT ToolBox2 设定本参数后，会自动设定 TOOL 坐标参数：MEXTLn (n: 1~16) 的值。由此，可以通过切换工具号码来选择（变更）作为动作基准的工具。R32TB/R33TB 不具备该功能，因此请使用 R56TB/R57TB 或 RT ToolBox2。

5) 抓手初始状态的设定

根据不同的连接方法，请设定以下参数。

- RIO 连接..... Clr 指令和专用输入（OUTRESET）等的通用输出信号复位时的动作设定用参数：输出信号复位模式（ORST700）
- 并行 I/O 连接..... 电源接通时的气动抓手 I/F 的输出设定用参数：抓手初始状态（HANDINIT）

表 7-20： 抓手初始状态设定参数

| 参数       | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                  | 出货时设定值                                          |
|----------|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| 输出信号复位模式 |          |            | 设定Clr指令和专用输入（OUTRESET）等的通用输出信号复位时的动作。电源开启时仍以此处所设定的模式输出信号。<br>通过下列参数对各信号以32位为单位进行设定。<br>（OFF/ON/ 保持=0/1/*）                                                                                             |                                                 |
|          | ORST700  | 字符串 4      | 设定信号号码700~731。<br>注）多重抓手使用输出信号716~723。<br>输出信号700~715为系统保留领域，因此不可进行设定。                                                                                                                                | 00000000,<br>00000000,<br>00000000,<br>00000000 |
| 抓手初始状态   | HANDINIT | 整数 8       | 设定电源接通时的气动抓手I/F的输出。<br>本参数用于指定位于机器人前端的抓手专用信号（900~999号）的电源开启时的初始值。<br>要设定通过通用IO（10000~18191号以外）进行抓手控制的情况下（通过HANDTYPE参数指定900号以外的信号）的电源开启时的初始状态，请使用ORS*参数，不使用该HANDINIT参数。<br>通过该ORS*参数设定的值为电源开启时的信号的初始值。 | 1,0,1,0,1,0,1,0                                 |

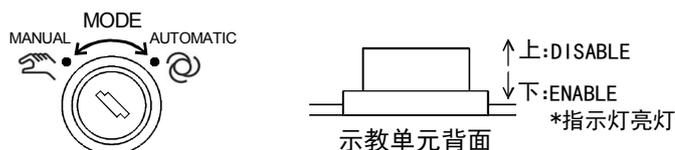
## (2) 抓手开闭操作 (R32TB)

以下所示为通过 R32TB 进行的抓手开闭操作画面。

仅在多重抓手进行 RIO 连接的情况下，除了画面下方显示的输入输出号码 (OUT-716、IN-716) 不同外，操作方法与通常的抓手的情况相同。最多可控制 6 个抓手，抓手 1、2、3、4、5、6 分别被分配给 X、Y、Z、A、B、C 轴的按键，多重抓手的情况下使用抓手 1 ~ 4。抓手打开可通过按 "+" 侧的按键、抓手关闭可通过按 "-" 侧的按键进行操作。

操作方法如下所示。

- 1) 将控制器的模式设为 "MANUAL"，按下示教单元的 [ENABLE] 开关使示教单元有效。（开关和 ENABLE LED 亮灯）



- 2) 按下「HAND」键以显示 < 抓手 > 画面。

OUT-716 表示抓手的开闭状态，IN-716 表示抓手确认输入信号的 ON/OFF 状态。

打开抓手 1 时按 [+X] 键，关闭抓手 1 时按 [-X] 键。其他抓手可通过 Y、Z、A 轴的按键进行相同的操作。

抓手画面的显示 [HAND]



可通过 [↓]、[↑] 键切换输入信号号码。



|                   |   |   |   |   |   |   |   |   |
|-------------------|---|---|---|---|---|---|---|---|
| OUT-716 ~ OUT-723 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 开 / 闭             | 闭 | 开 | 闭 | 开 | 闭 | 开 | 闭 | 开 |
| 抓手号码              | 4 |   | 3 |   | 2 |   | 1 |   |

|                 |     |     |     |     |     |     |     |     |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| IN-716 ~ IN-723 | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| 输入信号号码          | 723 | 722 | 721 | 720 | 719 | 718 | 717 | 716 |

注) 输入信号号码由多重抓手产品的规格决定。

(3) 工具号码 (TOOL 坐标参数) 与多重抓手的关系

为控制多重抓手, 将工具号码和 TOOL 坐标参数扩展至 1 ~ 16。请正确设定操作对象的工具号码, 以安全使用机器人。

**注意** 程序的自动运行时, 需要在切换工具数据 (MEXTL1 ~ 16) 的同时向示教位置动作的情况下, 请在需要的位置上将工具号码代入 M\_TOOL 变量并切换工具数据来运行。应充分注意如果示教时的工具数据与运行时的工具号码不一致, 会朝意想不到的方向动作。

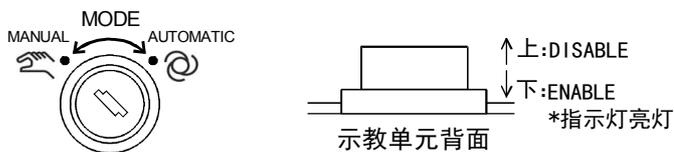
**注意** 程序的单步运行时, 要切换工具数据并进行动作的情况下, 应充分注意如果示教时的工具数据与单步运行时的工具号码不一致, 会朝意想不到的方向动作。

表 7-21: 工具号码与多重抓手的关系

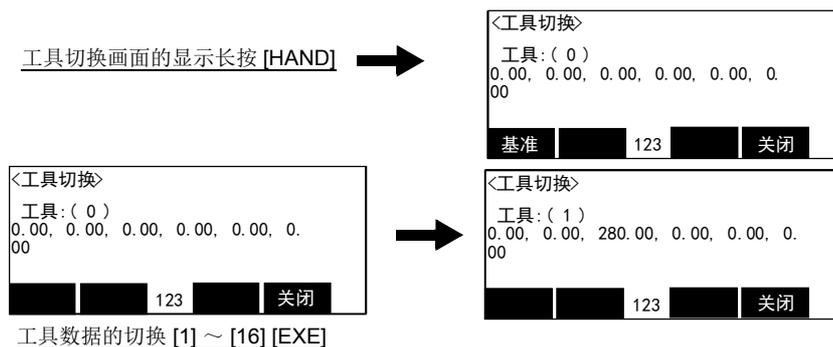
| 多重抓手号码 | 抓手 (手指) 号码 | 工具号码 (TOOL 坐标参数) |
|--------|------------|------------------|
| 多重抓手 1 | 抓手 1       | 工具 1 (MEXTL1)    |
|        | 抓手 2       | 工具 2 (MEXTL2)    |
|        | 抓手 3       | 工具 3 (MEXTL3)    |
|        | 抓手 4       | 工具 4 (MEXTL4)    |
| 多重抓手 2 | 抓手 1       | 工具 5 (MEXTL5)    |
|        | 抓手 2       | 工具 6 (MEXTL6)    |
|        | 抓手 3       | 工具 7 (MEXTL7)    |
|        | 抓手 4       | 工具 8 (MEXTL8)    |
| 多重抓手 3 | 抓手 1       | 工具 9 (MEXTL9)    |
|        | 抓手 2       | 工具 10 (MEXTL10)  |
|        | 抓手 3       | 工具 11 (MEXTL11)  |
|        | 抓手 4       | 工具 12 (MEXTL12)  |
| 多重抓手 4 | 抓手 1       | 工具 13 (MEXTL13)  |
|        | 抓手 2       | 工具 14 (MEXTL14)  |
|        | 抓手 3       | 工具 15 (MEXTL15)  |
|        | 抓手 4       | 工具 16 (MEXTL16)  |

工具号码的变更方法如下所示。

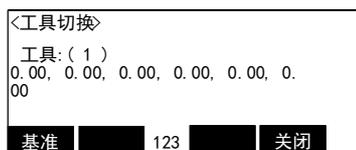
- 1) 将控制器的模式设为 "MANUAL", 按下示教单元的 [ENABLE] 开关使示教单元有效。(开关和 ENABLE LED 亮灯)



- 2) 长按 [HAND] 键以显示 < 工具切换 > 画面。
- 3) 按下所需的数字键并按 [EXE] 键后, 工具数据即发生切换。参数的 MEXTL1 ~ 16 与数字 1 ~ 16 对应。



4) 按下分配给 " 关闭 " 的功能键以退出。



退出 [F4]

5) JOG 画面的右上角会显示现在的工具号码 (T1 ~ T16)。

至此，工具号码的变更完成。

#### (4) 关于报警

与多重抓手有关的报警如表 7-22 所示。

表 7-22：报警一览表

| 报警号码  | 报警发生原因和对策 |                              |
|-------|-----------|------------------------------|
| C190  | 报警信息      | 连接中的示教单元与现在的工具号码不对应。         |
|       | 原因        | 连接中的示教单元与现在的工具号码不对应。         |
|       | 对策        | 请在确认现在的工具数据 (MEXTL) 后操作 TB。  |
| H8130 | 报警信息      | 远程 I/O 的通信端口异常。              |
|       | 原因        | 发生了多重抓手远程 I/O 的 CRC 报警或连接报警。 |
|       | 对策        | 请检查通信电缆或确认连接机器的电源。           |

## 7.4 关于样条插补

对机器人的动作指令之一的样条插补进行说明。

### 7.4.1 概要

#### (1) 概要

样条插补是指，沿着顺畅连接所指定的路径点的样条曲线，机器人以指定的速度进行动作的功能。

**Ex-T 样条插补**是指，将顺畅连接机器人所夹持的工件上指定的路径点的样条曲线，沿着任意的坐标系原点（**Ex-T** 坐标系原点），以指定的速度进行动作的（从工件看到的 **Ex-T** 坐标系原点以指定的速度进行相对的动作）功能。

在密封或研磨·去毛刺等作业中，将以往的直线·圆弧插补所无法对应的曲线状的轨迹描画的动作变为可能。

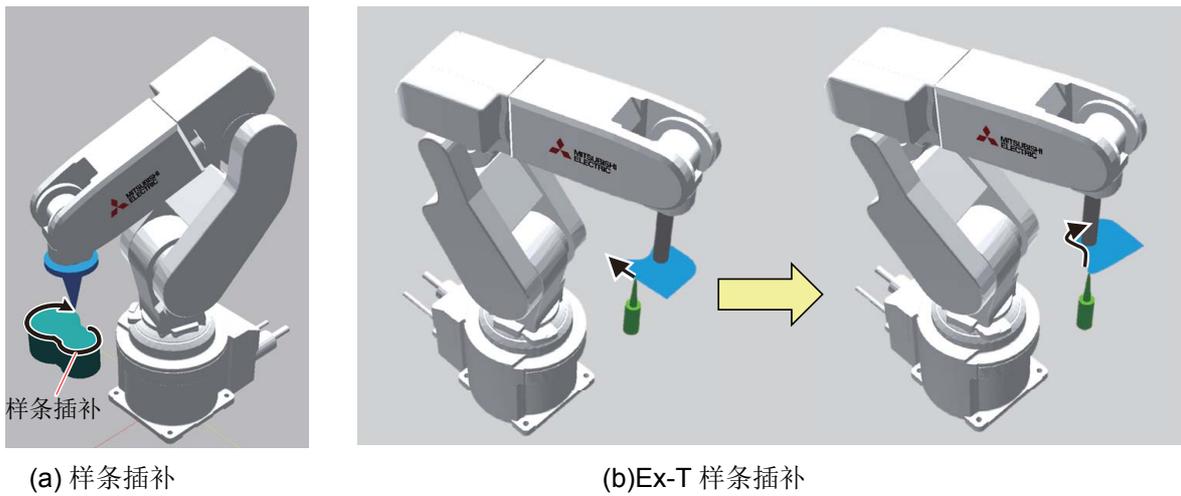


图 7-10: 样条插补的概要

#### (2) 特征

- 在各路径点之间生成顺畅的样条曲线，使得通过指定为路径点的机器人位置与姿势。机器人沿着该曲线以指定速度的线速进行动作。

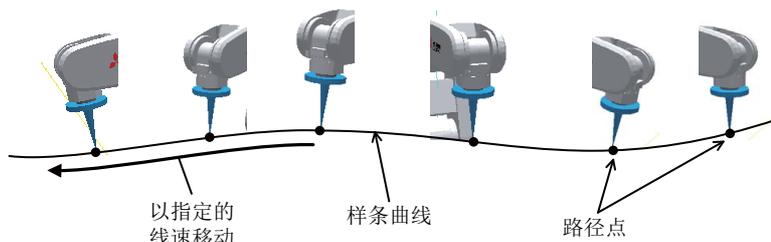


图 7-11: 样条插补

- 使用与机器人程序不同的、登录了路径点数据的样条插补专用的文件（样条文件）。该文件在 RT ToolBox2 的专用编辑画面（样条文件编辑画面）中创建·编辑，并写入至控制器。

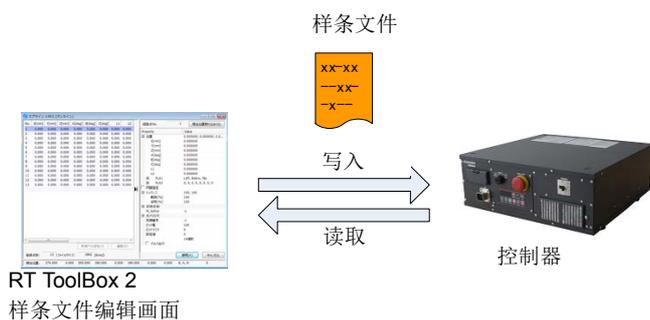


图 7-12: 将路径点数据进行专用文件化

- 通过使其独立为专用的文件，可在多个机器人程序中共享样条插补，并且样条插补也可以简单地切换。

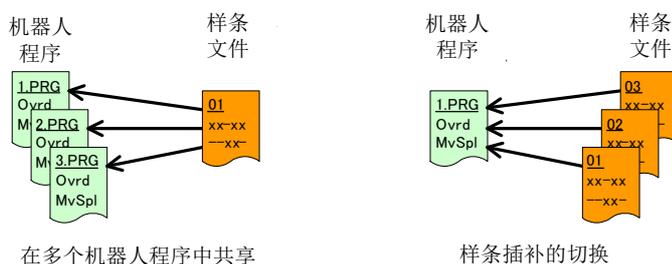


图 7-13: 基于专用文件化的特征

- 可以部分性补偿样条曲线的膨胀，或更改为直线·圆弧轨迹。
- 除机器人的位置以外，可以登录信号输出及任意的数值作为路径点的数据。

### (3) 必要的设备与软件版本

使用样条插补时，需要 RT ToolBox2。样条插补可以在表 7-23 所示的软件版本的设备中使用。

表 7-23: 可以使用样条插补的软件版本

| 设备       | 功能        | 控制器                                        | RT ToolBox2   |
|----------|-----------|--------------------------------------------|---------------|
| 可使用的软件版本 | 样条插补      | F-Q 系列: Ver. R5 以上<br>F-D 系列: Ver. S5 以上   | Ver. 3.10L 以上 |
|          | Ex-T 样条插补 | F-Q 系列: Ver. R6b 以上<br>F-D 系列: Ver. S6b 以上 | Ver.3.40S 以上  |

关于示教单元（R32TB/R33TB/R56TB/R57TB），无软件版本的指定。

### (4) 术语

样条插补的说明中所使用的术语如表 7-24 所示。

表 7-24: 术语的说明

| 术语   | 说明                                    |
|------|---------------------------------------|
| 路径点  | 为生成样条曲线而指定的机器人的位置数据（直角坐标值）。           |
| 块    | 是指相邻 2 个路径点间所生成的曲线或线段。                |
| 路径   | 是指通过样条插补指令生成的、通过路径点的样条曲线。             |
| 开始位置 | 表示开始样条插补动作的路径点。                       |
| 结束位置 | 表示结束样条插补动作的路径点。                       |
| 样条文件 | 存有路径点及执行时所需要的设定值等的文件，1 个文件对应 1 个样条插补。 |

| 术语         | 说明                                        |
|------------|-------------------------------------------|
| 样条文件编辑画面   | RT ToolBox2 中配备的用于创建·编辑·保存样条文件的专用画面。      |
| DXF 文件导入画面 | RT ToolBox2 中配备的用于从 DXF 文件创建·保存样条文件的专用画面。 |
| Ex-T 坐标系原点 | 是指机器人的外部被任意定义的控制对象的坐标系原点。                 |

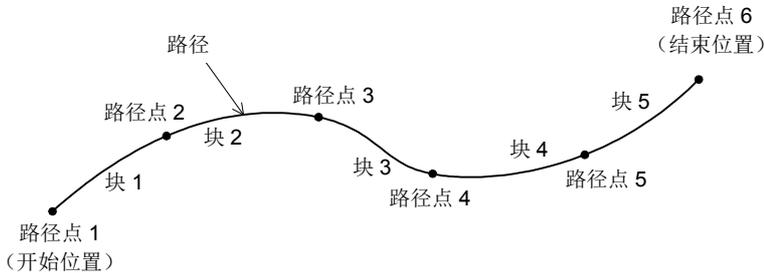


图 7-14: 样条插补的术语

## 7.4.2 规格

### (1) 基本规格

表 7-25: 基本规格

| 项目       |          | 规格                                                                                                                                                              |
|----------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 对应的机器人   |          | 垂直多关节（6 轴型）机器人、水平多关节（4 轴型）机器人<br>※ 在垂直多关节（5 轴型）机器人与用户机械中无法使用。                                                                                                   |
| 对应的机器人语言 |          | MELFA-BASIC V 中已增加样条插补的指令、函数、状态变量。<br>（MvSpl 指令、EMvSpl 指令、SetCalFrm 指令、SplPos 函数、SplSpd 函数、SplECord 函数、<br>状态变量 M_SplPno、M_SplVar）<br>※ 在 MELFA-BASIC IV 中无法使用。 |
| 动作方法     |          | 将样条文件登录至控制器，并执行样条插补的指令。<br>生成通过样条文件中登录的路径点的曲线后，让机器人沿着该曲线动作。                                                                                                     |
| 样条文件     | 文件登录数    | 最多 99 个<br>※ 实际可登录个数，取决于文件大小及保存区域的空余容量。                                                                                                                         |
|          | 路径点登录数   | 每个文件最多 5000 点<br>※ 5000 点约为 470KB 的文件大小。                                                                                                                        |
|          | 创建·编辑·保存 | 使用 RT ToolBox2 的样条文件编辑画面进行文件的创建·编辑·保存。<br>使用 SplWrt 指令、SplFWrt 指令进行文件的创建·编辑·保存。                                                                                 |
|          | 路径点的登录方法 | 在 RT ToolBox2 的样条文件编辑画面中，通过以下的操作登录路径点。<br>示教操作、MDI、CSV 文件的导入、MXT 文件的导入、DXF 文件的导入、机器人程序<br>（SplWrt 指令）                                                           |
| 路径补偿     | 允许公差指定   | 用块单位来指定曲线的膨胀情况。也可将任意的块看作直线，而不是样条曲线。                                                                                                                             |
|          | 圆弧指定     | 可指定连续 3 点的路径点，以看作圆弧轨迹，而不是样条曲线。<br>※ 根据样条文件的版本不同，指定圆弧的方法也不同。                                                                                                     |
|          | 样条取消     | 通过动作方向变化较大的路径点来分割样条曲线。                                                                                                                                          |
|          | 块长比率     | 将前后的块进行比较，线段长度为所规定的比率以上时，自动将该块看作直线。                                                                                                                             |
| 动作模式     | 线速固定     | 生成插补指令，使得以指定的速度进行等速动作。                                                                                                                                          |
|          | 线速可变     | 因指定的速度大·姿势变化大等原因导致动作中可能要发生超速时，自动降低速度进行动作。                                                                                                                       |
| 信号输出     |          | 通过路径点时，可以输出通用信号。                                                                                                                                                |
| 数值设定     |          | 可以给每个路径点设定任意的数值，并通过状态变量对此进行浏览。                                                                                                                                  |

## (2) 限制事项

表 7-26: 限制事项

| 项目          | 限制的内容                                                                                                                                                                                                                                                                                                                                                                                                             |      |    |       |              |        |              |         |               |        |                 |       |             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----|-------|--------------|--------|--------------|---------|---------------|--------|-----------------|-------|-------------|
| 附加轴         | 样条插补不以附加轴为对象。即使对各路径点设定了附加轴的坐标值，在样条插补执行过程中也不会从开始位置的坐标值开始进行动作。                                                                                                                                                                                                                                                                                                                                                      |      |    |       |              |        |              |         |               |        |                 |       |             |
| Cnt 指令      | 即使通过 Cnt 指令指定了插补的连续动作，对样条插补的开始位置·结束位置也不起作用。                                                                                                                                                                                                                                                                                                                                                                       |      |    |       |              |        |              |         |               |        |                 |       |             |
| Oadl 指令     | 即使因 Oadl 指令使得最佳加减速控制变为有效，对样条插补也不起作用。                                                                                                                                                                                                                                                                                                                                                                              |      |    |       |              |        |              |         |               |        |                 |       |             |
| MvTune 指令   | MvTune 指令的动作模式的指定，对样条插补不起作用。                                                                                                                                                                                                                                                                                                                                                                                      |      |    |       |              |        |              |         |               |        |                 |       |             |
| FsGChg 指令   | 无法将样条插补看作通过 FsGChg 指令（指定力觉控制的控制特性更改）指定 < 切换开始位置 > 的对象。                                                                                                                                                                                                                                                                                                                                                            |      |    |       |              |        |              |         |               |        |                 |       |             |
| 工具 / 基本切换   | 样条插补的执行中断时，无法切换工具数据 / 基本数据。                                                                                                                                                                                                                                                                                                                                                                                       |      |    |       |              |        |              |         |               |        |                 |       |             |
| Jrc 指令      | 样条插补的执行中断时，无法执行 Jrc 指令。                                                                                                                                                                                                                                                                                                                                                                                           |      |    |       |              |        |              |         |               |        |                 |       |             |
| 插补动作相关的状态变量 | 插补动作相关的以下的状态变量，不返回样条插补的状态。 <table border="1" data-bbox="528 768 1147 972"> <thead> <tr> <th>状态变量</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>M_Acl</td> <td>返回当前的加速时间比率。</td> </tr> <tr> <td>M_DAcI</td> <td>返回当前的减速时间比率。</td> </tr> <tr> <td>M_Ratio</td> <td>返回到达目的位置的到达率。</td> </tr> <tr> <td>M_RDst</td> <td>返回至目的位置为止的残余距离。</td> </tr> <tr> <td>M_Spd</td> <td>返回当前所设定的速度。</td> </tr> </tbody> </table> | 状态变量 | 功能 | M_Acl | 返回当前的加速时间比率。 | M_DAcI | 返回当前的减速时间比率。 | M_Ratio | 返回到达目的位置的到达率。 | M_RDst | 返回至目的位置为止的残余距离。 | M_Spd | 返回当前所设定的速度。 |
| 状态变量        | 功能                                                                                                                                                                                                                                                                                                                                                                                                                |      |    |       |              |        |              |         |               |        |                 |       |             |
| M_Acl       | 返回当前的加速时间比率。                                                                                                                                                                                                                                                                                                                                                                                                      |      |    |       |              |        |              |         |               |        |                 |       |             |
| M_DAcI      | 返回当前的减速时间比率。                                                                                                                                                                                                                                                                                                                                                                                                      |      |    |       |              |        |              |         |               |        |                 |       |             |
| M_Ratio     | 返回到达目的位置的到达率。                                                                                                                                                                                                                                                                                                                                                                                                     |      |    |       |              |        |              |         |               |        |                 |       |             |
| M_RDst      | 返回至目的位置为止的残余距离。                                                                                                                                                                                                                                                                                                                                                                                                   |      |    |       |              |        |              |         |               |        |                 |       |             |
| M_Spd       | 返回当前所设定的速度。                                                                                                                                                                                                                                                                                                                                                                                                       |      |    |       |              |        |              |         |               |        |                 |       |             |
| 中断处理        | 即使将“停止类型 1”指定为中断的停止类型，样条插补动作中的中断，也将与“停止类型 2”同样停止。                                                                                                                                                                                                                                                                                                                                                                 |      |    |       |              |        |              |         |               |        |                 |       |             |
| 跟踪功能        | 不能同时使用跟踪功能与样条插补。                                                                                                                                                                                                                                                                                                                                                                                                  |      |    |       |              |        |              |         |               |        |                 |       |             |

## (3) 样条插补中的机器人的动作

- 机器人的当前位置与样条插补的开始位置不相同，如执行样条插补，则机器人将以直线插补动作至开始位置后，再开始进行样条插补。直线插补将以 OvrD 指令·Spd 指令·倍率修调所指定的速度进行动作。  
对于附加轴，当前位置与开始位置不相同，也会动作至开始位置。

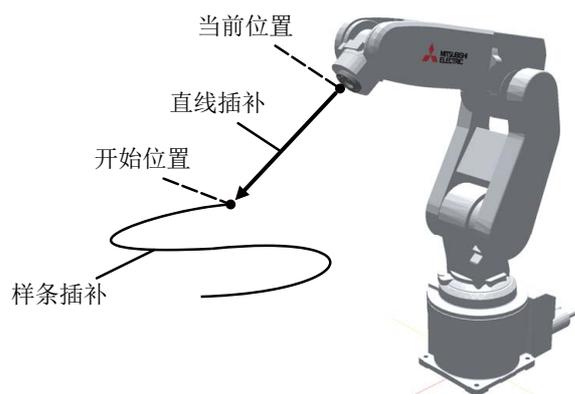


图 7-15: 当前位置与开始位置不相同 (样条插补)

- 机器人的当前位置与 Ex-T 样条插补的开始位置不相同，如执行 Ex-T 样条插补，则机器人将以 Ex-T 直线插补动作至开始位置后，再开始进行 Ex-T 样条插补。Ex-T 直线插补将以 OvrD 指令·Spd 指令·倍

率修调所指定的速度进行动作。  
对于附加轴，当前位置与开始位置不相同，也会动作至开始位置。

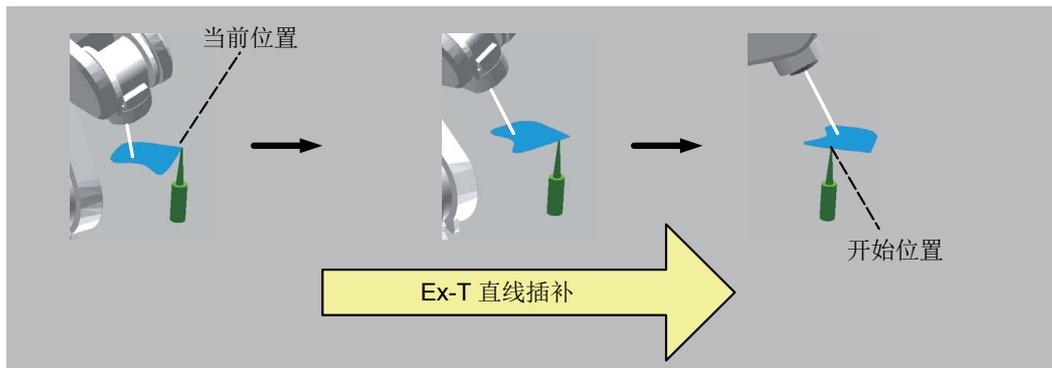


图 7-16: 当前位置与开始位置不相同 (Ex-T 样条插补)

- 样条插补执行过程中可以通过停止输入等使运行中断。重新开始运行后，将从中断的位置开始继续进行剩余的样条插补。
- 样条插补执行过程中通过停止输入等使运行中断后，通过 JOG 操作移动了机器人·附加轴时，如重新开始运行，样条插补将返回中断的位置后再继续运行。通过关节插补返回中断位置。不反映参数 RETPATH (中断时 JOG 进给后的自动复归设定) 的设定。

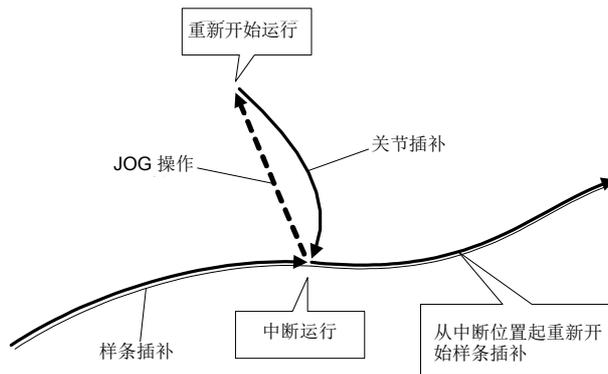


图 7-17: 中断·重新开始时的动作

- 样条插补执行过程中通过停止输入等使运行中断后，如将中断执行的机器人程序读取至示教单元·RT ToolBox2 的程序编辑画面，则样条插补的中断位置信息将被复位。因此，如从中断行开始重新运行，则机器人将向样条插补的开始位置移动之后再继续运行。

## (4) 路径点相关的检查

对各路径点进行表 7-27 中记载的检查。

表 7-27: 路径点相关的检查

| 种类          | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------|----|----------|----|----------|----|----------|-----|----------|-----|-----------|-----|-----------|
| 路径点间的距离     | <p>检查相邻的 2 点路径点是否相隔着适当的距离。适当的距离会根据指定的速度发生变化，对速度而言，2 点路径点过近时，会发生 L2611（路径点过近）错误。<br/>应参考下表及下图来决定指令速度与路径点间的距离。</p> <table border="1"> <thead> <tr> <th>指令速度 [mm/s]</th> <th>必要的路径点间的距离</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>0.6mm 以上</td> </tr> <tr> <td>20</td> <td>1.2mm 以上</td> </tr> <tr> <td>50</td> <td>3.0mm 以上</td> </tr> <tr> <td>100</td> <td>6.0mm 以上</td> </tr> <tr> <td>200</td> <td>12.0mm 以上</td> </tr> <tr> <td>500</td> <td>30.0mm 以上</td> </tr> </tbody> </table> <p>此外，该表的指令速度可通过以下的算式求出。<br/>指令速度 = (MvSpl 指令的 &lt;速度&gt;) × (Ovrd 指令)</p> | 指令速度 [mm/s] | 必要的路径点间的距离 | 10 | 0.6mm 以上 | 20 | 1.2mm 以上 | 50 | 3.0mm 以上 | 100 | 6.0mm 以上 | 200 | 12.0mm 以上 | 500 | 30.0mm 以上 |
| 指令速度 [mm/s] | 必要的路径点间的距离                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 10          | 0.6mm 以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 20          | 1.2mm 以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 50          | 3.0mm 以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 100         | 6.0mm 以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 200         | 12.0mm 以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 500         | 30.0mm 以上                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 姿势的变化量      | <p>检查相邻的 2 点路径点的姿势变化是否过大。姿势的变化角度大于 150 度时，会发生 L2611（姿势变化过大）错误。<br/>※Ex-T 样条插补时，检查工件上的路径点的姿势变化是否过大。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
| 结构标志的值      | <p>检查相邻的 2 点路径点的结构标志的值是否相同。结构标志的值不相同，会发生 L2611（路径点的结构标志不相同）错误。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |            |    |          |    |          |    |          |     |          |     |           |     |           |

### 7.4.3 功能的说明

#### (1) 路径补偿

通过使用路径补偿的功能，可以补偿样条曲线的形状。配备有“允许公差指定”、“圆弧指定”、“样条取消”、“块长比率”用作路径补偿。

##### ■ 允许公差指定

希望用块单位调整样条曲线的膨胀情况时，指定允许公差<sup>※1)</sup>。设定范围为 0 ~ 100%，初始状态为 100%（无调整）。设定值越小，曲线的膨胀情况越小，为 0% 时，对象块为直线。

对姿势也可指定允许公差。将设定值得越小，则姿势变化的膨胀情况变得越小，但通过路径点时的姿势变化的顺畅度也会降低。

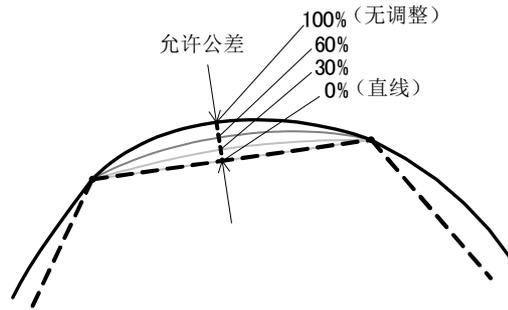
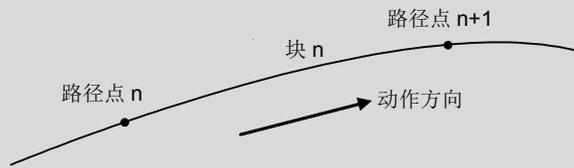


图 7-18: 允许公差

#### ◇◆◇ 允许公差对象的块 ◇◆◇

允许公差将作为路径点的数据进行设定。各块的开始位置侧的路径点中所设定的允许公差将被反映至该块中。下图的示例中，设定在路径点 n 中的允许公差将被反映至块 n 中。



##### ■ 圆弧指定

根据样条文件的版本不同，圆弧指定的方法也不同。

#### 1) 样条文件版本 02

如对连续的 2 个块进行圆弧指定（在块的开始点上指定圆弧），则由这 2 个块组成的曲线将成为圆弧，而不是样条。如用于像弯角部分那样需要圆弧状曲线的部位，则可用比样条曲线更少的路径点数描画出更准确的圆弧。

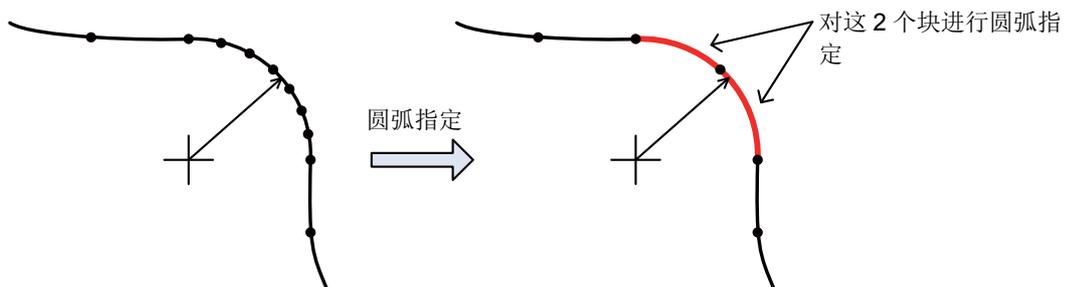


图 7-19: 圆弧指定（样条文件版本 02）

※1) 允许公差一般是表示允许误差 · 精度，但本功能中是指相对于连接路径点的直线（弦）的样条曲线的膨胀情况。

圆弧指定务必在 2 个连续的块中进行指定。仅在 1 个块中有指定时，会发生 L2613（圆弧的指定点数不够）错误。

此外，通过指定 2 个块，形成圆弧的 3 点并排在直线上而无法生成圆弧时，会发生 L2613（块数据算出异常（Cir.Arc））错误。

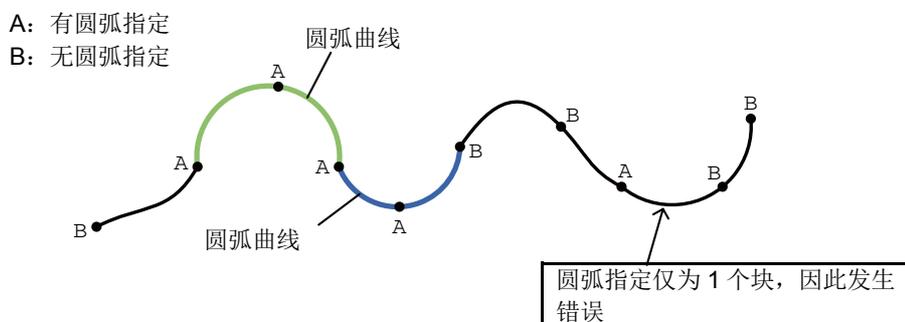


图 7-20: 因圆弧指定而导致发生错误的示例 (样条文件版本 02)

## 2) 样条文件版本 01

如对连续的 3 点路径点进行圆弧指定，则连接这 3 点的曲线将成为圆弧，而不是样条。如用于像弯角部分那样需要圆弧状曲线的部位，则可用比样条曲线更少的路径点数描画出更准确的圆弧。

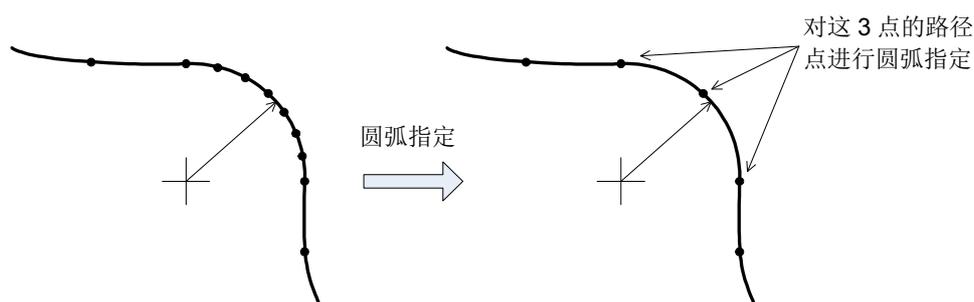


图 7-21: 圆弧指定 (样条文件版本 01)

圆弧指定务必在 3 点连续的路径点中进行指定。仅指定了 1 点时，将忽视该指定而描画出样条曲线，但仅指定了 2 点时，则会发生 L2613（圆弧的指定点数不够）错误。

此外，指定的 3 点并排在直线上而无法生成圆弧时，会发生 L2613（块数据算出异常（Cir.Arc））错误。

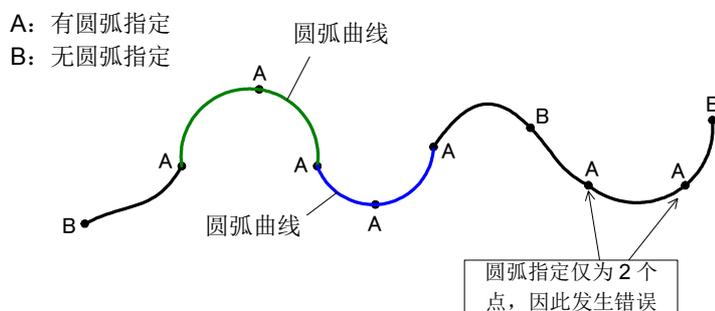
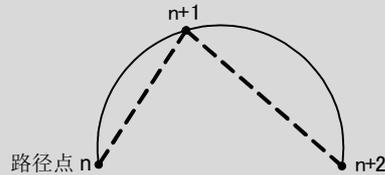


图 7-22: 因圆弧指定而导致发生错误的示例 (样条文件版本 01)

## ◇◆◇圆弧指定与其他的路径补偿功能的关系◆◆◇

允许公差指定·块长比率对描画圆弧曲线的块不起作用。样条取消对圆弧的通过点（路径点  $n+1$ ）不起作用。关于始点（路径点  $n$ ）与终点（路径点  $n+2$ ），前后的块形成的角度比取消角度大时，将暂时减速停止。



## ⚠注意

连续进行圆弧指定时，根据路径点的采点方法不同，有可能会如图所示变为折返的不松弛的圆弧。样条取消不起作用、在圆弧之间的连接点不能减速停止时，将出现急剧的反转动作，因此应充分注意所生成的圆弧曲线的形状。



## ■样条取消

希望形成弯角边而不是以样条曲线顺畅通过动作方向变化大的路径点时，可通过指定取消角度，用其路径点来分割样条曲线。

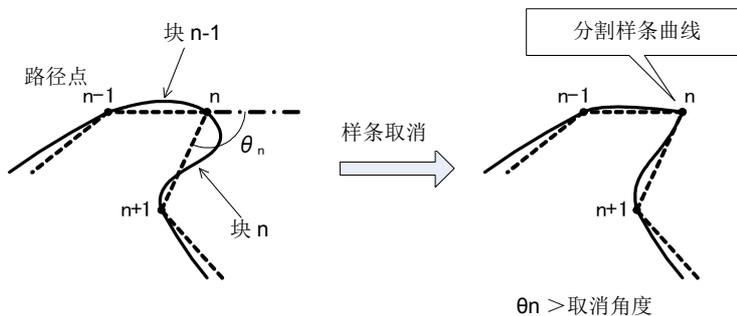


图 7-23: 样条取消

例如，希望通过图 7-23 的路径点  $n$  来形成弯角边时，如设定比路径点  $n$  的前后的块（块  $n-1$  与块  $n$ ）形成的角度  $\theta_n$  更小的角度，则可用路径点  $n$  来分割样条曲线。机器人在分割了样条曲线的路径点  $n$  暂时减速停止后，重新加速并执行剩余的插补。

取消角度的设定范围如下所示，每个样条插补均可分别设定。

设定范围：0 ~ 180 度（单位为 1 度）

※ 设定为 0 度时，样条取消不起作用。

※ 初始值为 120 度。

## ⚠注意

将样条取消设为无效时，或将取消角度更改为较大的值时，即使是动作方向发生大变化的路径点，样条曲线也不被分割，且机器人不会减速。因此，有可能会发生机器人振动或伺服错误。

此外，样条曲线出现急剧折弯时，有可能无法算出插补位置而发生 L2163（样条插补指令算出异常）错误。

应确认路径点的位置，设定适当的取消角度。

### ■ 块长比率

对于相邻的块，块长很长时，可使该块自动以直线动作，而不是样条曲线。

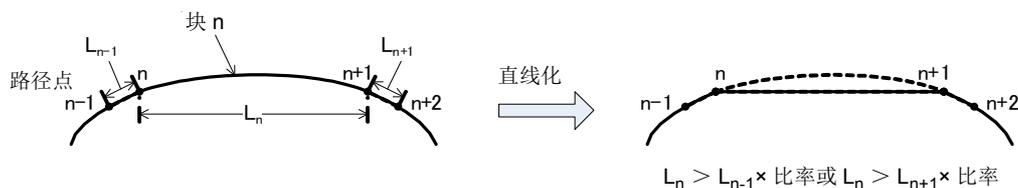


图 7-24: 块长比率导致的直线化

块  $n$  的长度  $L_n$  相对于相邻的块长  $L_{n-1}$  或  $L_{n+1}$ ，如果比指定的比率长，则块  $n$  为直线。

块长比率的设定范围如下所示，每个样条插补均可分别设定。

设定范围：0 ~ 100 倍（1 倍单位）

※ 指定为 0 倍时，块长比率不起作用。

※ 初始值为 8 倍。

此外，对因块长比率而形成直线的块，即使指定了允许公差，也将优先适用块长比率的功能。

### ◇◆◇ 路径补偿中的注意点 ◇◆◇

进行过路径补偿的曲线及其前后的曲线的连接部分，与未进行过补偿的样条曲线之间的连接相比，会丧失顺畅度。因此，根据连接的情况及指定速度，在曲线的连接部分有可能发生速度变动。

### (2) 动作模式

样条插补将保持指定的速度使机器人动作，以使控制点发生移动。因此，指定的速度较大时，或对于路径点间的距离而言姿势的变化量较大时，有可能在动作过程中出现超速。

该情况下，虽然不能保持指定的速度，但配备有可选择为动作时尽量不发生超速的动作模式。如表 7-28 所示，动作模式有 2 种。

表 7-28: 动作模式

| 动作模式 | 说明                                                                                                          |
|------|-------------------------------------------------------------------------------------------------------------|
| 线速固定 | 以指定的速度进行动作，保持一定的速度。根据指定的速度及姿势的变化量的大小，在动作过程中发生超速。                                                            |
| 线速可变 | 以指定的速度进行动作，保持一定的速度。但是，如判断为动作过程中有可能发生超速时，自动降低动作速度来进行控制，以尽量不发生超速。已无可能发生超速时，将恢复原来的指定速度。<br>※ 根据动作状况，有可能无法抑制超速。 |

每个样条插补均可分别设定动作模式。初始状态为线速固定。

在样条插补的动作中发生超速时，该作业未必要用固定的速度进行动作时，应尝试“线速可变”。

### ◇◆◇ 单步进给时 ◇◆◇

以单步进给执行样条插补时，在可能要超出规定速度的情况下，动作模式即使为“线速固定”，也会降低动作速度。

(3) 信号输出

通过路径点时，可以将任意的外部输出信号置为 ON/OFF。通过该功能，在样条插补的动作中，例如通过任意的路径点时向外围装置输出触发信号这样的操作，即使不对机器人程序进行记述也可以实现。

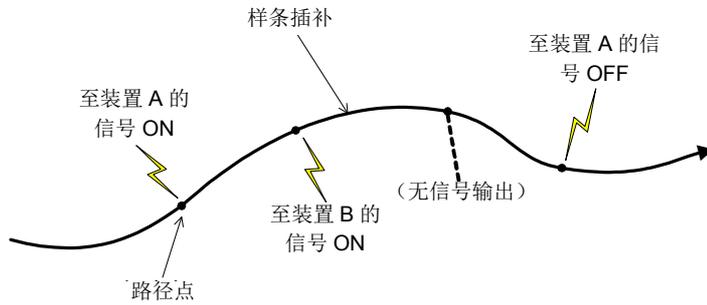


图 7-25: 信号输出

要进行信号输出，需要设定表 7-29 的项目作为路径点的数据。

表 7-29: 信号输出的设定项目

| 设定项目 | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |                  |   |     |   |        |    |          |    |              |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|------------------|---|-----|---|--------|----|----------|----|--------------|
| 起始号码 | 指定输出信号的起始号码。<br>设定范围：-1 ~ 32767<br>※-1 时，在该路径点中，本功能无效。<br>初始值为 -1。设定范围如上所示，但实际可输出的起始号码取决于所连接的软元件的种类。                                                                                                                                                                                                                                                                                                                                                                                    |    |                  |   |     |   |        |    |          |    |              |
| 位宽   | 指定要输出的信号的位宽。<br>设定范围：1、8、16、32 位<br>初始值为 1 位。                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |                  |   |     |   |        |    |          |    |              |
| 位掩码  | 对于指定的位宽，以位为单位对信号输出设为有效时的掩码模式进行指定。<br>设定范围根据位宽不同而发生变化。<br><table border="1" style="margin: 10px auto;"> <thead> <tr> <th>位宽</th> <th>输出数据设定范围 [16 进制]</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0、1</td> </tr> <tr> <td>8</td> <td>0 ~ FF</td> </tr> <tr> <td>16</td> <td>0 ~ FFFF</td> </tr> <tr> <td>32</td> <td>0 ~ FFFFFFFF</td> </tr> </tbody> </table> <p>对于位掩码为 1 的位，所指定的输出数据的 ON/OFF 状态将反映至信号输出中。<br/>对于位掩码为 0 的位，将保持此时的外部输出信号的 ON/OFF 状态。<br/>初始值为 0，因此要输出信号时需要设定位掩码。</p> | 位宽 | 输出数据设定范围 [16 进制] | 1 | 0、1 | 8 | 0 ~ FF | 16 | 0 ~ FFFF | 32 | 0 ~ FFFFFFFF |
| 位宽   | 输出数据设定范围 [16 进制]                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |    |                  |   |     |   |        |    |          |    |              |
| 1    | 0、1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |                  |   |     |   |        |    |          |    |              |
| 8    | 0 ~ FF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |                  |   |     |   |        |    |          |    |              |
| 16   | 0 ~ FFFF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                  |   |     |   |        |    |          |    |              |
| 32   | 0 ~ FFFFFFFF                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |                  |   |     |   |        |    |          |    |              |
| 输出数据 | 指定信号输出的数据。<br>设定范围根据位宽不同而发生变化。<br><table border="1" style="margin: 10px auto;"> <thead> <tr> <th>位宽</th> <th>输出数据设定范围 [16 进制]</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0、1</td> </tr> <tr> <td>8</td> <td>0 ~ FF</td> </tr> <tr> <td>16</td> <td>0 ~ FFFF</td> </tr> <tr> <td>32</td> <td>0 ~ FFFFFFFF</td> </tr> </tbody> </table> <p>初始值为 0。</p>                                                                                                                               | 位宽 | 输出数据设定范围 [16 进制] | 1 | 0、1 | 8 | 0 ~ FF | 16 | 0 ~ FFFF | 32 | 0 ~ FFFFFFFF |
| 位宽   | 输出数据设定范围 [16 进制]                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |    |                  |   |     |   |        |    |          |    |              |
| 1    | 0、1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |                  |   |     |   |        |    |          |    |              |
| 8    | 0 ~ FF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |                  |   |     |   |        |    |          |    |              |
| 16   | 0 ~ FFFF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                  |   |     |   |        |    |          |    |              |
| 32   | 0 ~ FFFFFFFF                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |                  |   |     |   |        |    |          |    |              |
| 脉冲输出 | 如将脉冲输出设为有效，则仅会在通过路径点后的约 14ms 间将上述信号输出的指定反映至外部输出信号中。经过约 14ms 后，外部输出信号的状态将恢复原来的状态。<br>初始值无效。                                                                                                                                                                                                                                                                                                                                                                                              |    |                  |   |     |   |        |    |          |    |              |

路径点中设定的信号输出的数据与外部输出信号的状态变化的示例如表 7-30 所示。

表 7-30: 信号输出的示例 (初始状态为全部信号 OFF)

| 路径点 | 起始号码 | 位宽 | 位掩码 | 输出数据 | 脉冲输出 | 外部输出信号的状态 [●: ON/○: OFF] |     |     |     |     |     |     |     |
|-----|------|----|-----|------|------|--------------------------|-----|-----|-----|-----|-----|-----|-----|
|     |      |    |     |      |      | 107                      | 106 | 105 | 104 | 103 | 102 | 101 | 100 |
| 1   | 100  | 1  | 1   | 1    | 无效   | ○                        | ○   | ○   | ○   | ○   | ○   | ○   | ●   |
| 2   | 102  | 1  | 1   | 1    | 无效   | ○                        | ○   | ○   | ○   | ○   | ●   | ○   | ●   |
| 3   | -1   | 1  | 0   | 0    | 无效   | ○                        | ○   | ○   | ○   | ○   | ●   | ○   | ●   |
| 4   | 100  | 8  | 5C  | FA   | 无效   | ○                        | ●   | ○   | ●   | ●   | ○   | ○   | ●   |
| 5   | 100  | 8  | FF  | AF   | 有效   | ●                        | ●   | ●   | ●   | ●   | ●   | ●   | ●   |
|     |      |    |     |      |      | 约 14ms 后 ⇒               |     |     |     |     |     | ○   | ●   |

此外, 如过信号输出的输出目标中分配有专用输出信号, 则在执行信号输出时会发生 L0091 (该信号已分配给专用输出) 错误。请勿向分配了专用输出的位输出信号。

此外, 通过信号输出指定的外部输出信号并不是被占用了。通过多任务在其他的机器人程序中更改对象的外部输出信号后, 此更改将会被反映, 因此应加以注意。

#### ◇◆◇路径点通过的判断◇◆◇

路径点的通过, 用位置指令等级来判断。不用实际的机器人位置 (反馈位置) 来判断。

#### (4) 数值设定

可设定任意的数值用作路径点的数据。

设定范围: -1 ~ 32767 (整数)

※-1 表示未设定。

※ 初始值为 -1。

浏览状态变量 M\_SplVar, 可以确认最新通过的路径点中设定的值。通过在多个路径点中事先设定同一数值, 并通过多任务浏览状态变量 M\_SplVar, 则可在样条插补动作中通过所规定的路径点时, 执行规定的处理。

通过未设定的路径点时, 状态变量 M\_SplVar 的值不发生变化。返回此时保持的值。

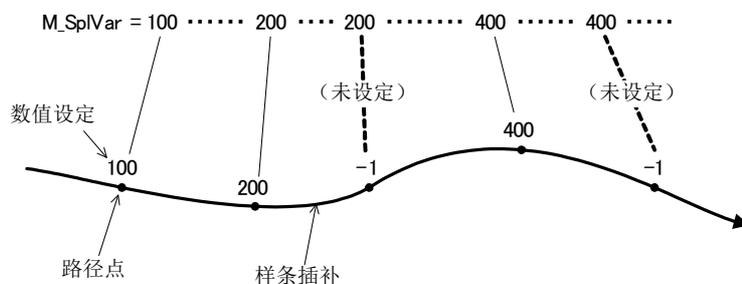


图 7-26: 数值设定

#### (5) 帧转换

帧转换是指对任意的路径, 保持其形状不变而将路径移动至其他位置的功能。

通过该功能,

- 从图纸上的位置转换至实际的位置
- 对作业对象进行偏位补偿
- 基于样条插补进行作业区域的更改

以上事项等均可对应。

■处理的概略

以图 7-27 为例对帧转换进行说明。

对任意的路径设定基准坐标系。坐标系的设定为，与 Fram 函数同样地指定 3 个位置数据。示例中，指定位置数据 PR1、PR2、PR3 后设定了基准坐标系 “Xfr-Zfr-Yfr”。

接着同样设定转换后的基准坐标系。示例中，指定位置数据 PC1、PC2、PC3 以设定转换后的基准坐标系 “Xfc-Zfc-Yfc”。

执行帧转换后，算出全部路径点，使转换前后的从基准坐标系的原点至路径点的相对变位相同。示例中，从路径点 P11 ~ P14 算出了 P21 ~ P24。

此外，转换后的各路径点的结构标志和多旋转标志、附加轴数据，将设定为与转换前的对应路径点相同的值。

此外，Ex-T 样条插补时，Ex-T 坐标系原点 “XE1-ZE1-YE1” 也将进行帧转换。

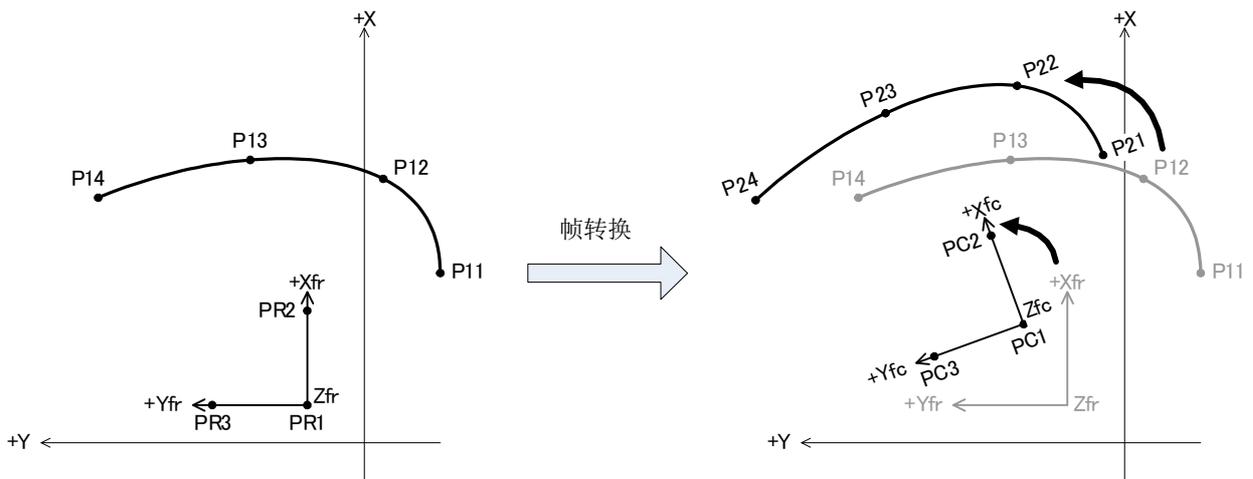


图 7-27: 帧转换 (样条插补)

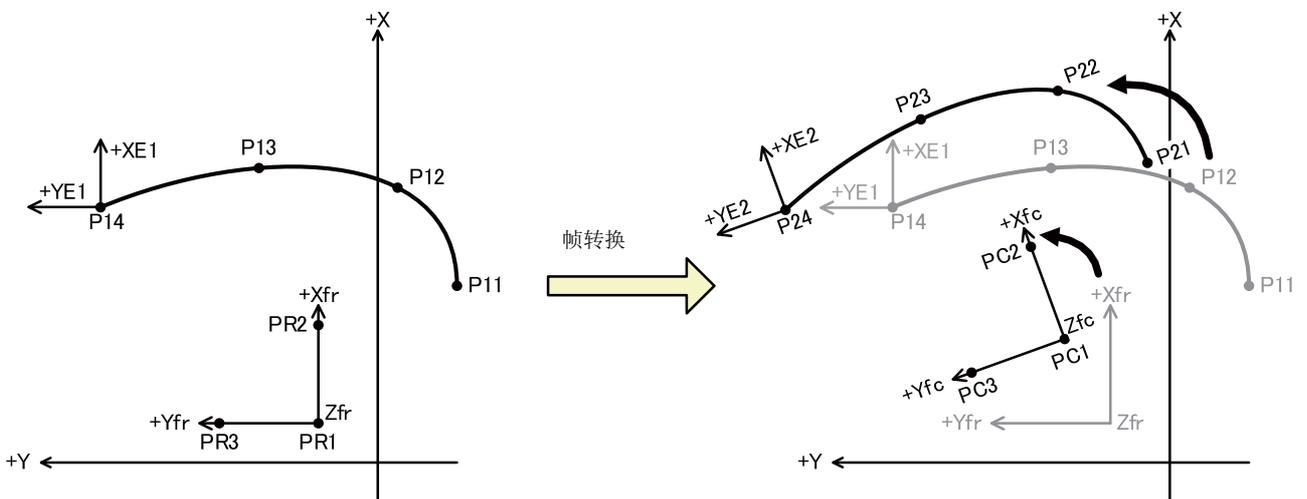


图 7-28: 帧转换 (Ex-T 样条插补)

■要设定的数据

为了定义坐标系，与 Fram 函数同样设定以下 3 个位置数据。

原点位置..... 相当于图 7-27 及图 7-28 的 PR1、PC1

X 轴上的位置..... 相当于图 7-27 及图 7-28 的 PR2、PC2

XY 平面的 +Y 方向的位置 ... 相当于图 7-27 及图 7-28 的 PR3、PC3

转换前与转换后共计设定 6 个位置数据的 X、Y、Z 轴坐标值。

此外，3 个位置数据中含有同一点时，或 3 个位置数据并排成直线状时，将无法定义坐标系并发生 L2041 (无法算出帧转换坐标) 错误。

## ■ 执行方法

配备有 3 种方法用作帧转换的执行方法。

表 7-31: 帧转换的执行方法

| 执行方法              | 说明                                                           | 使用案例                                                        |
|-------------------|--------------------------------------------------------------|-------------------------------------------------------------|
| 通过 RT ToolBox2 执行 | 在 RT ToolBox2 的样条文件编辑画面中执行帧转换, 将转换结果用作路径点, 创建样条文件。           | · 通过 JOG 操作等对转换后的路径点数据进行个别补偿时<br>· 希望将转换后的路径点数据用作其他路径的原本数据时 |
| 将坐标系保存至样条文件       | 将坐标系的设定保存至样条文件, 在执行样条插补时依次对路径点进行帧转换。                         | · 希望将图纸上的坐标值留作路径点的坐标值时<br>· 坐标系固定为 1 个时                     |
| 在机器人程序中设定坐标系      | 使用 SetCalFrm 指令, 在机器人程序上设定坐标系。执行样条插补时, 使用所设定的坐标系依次对路径点进行帧转换。 | · 希望对 1 个路径适用多个坐标系时<br>· 每次均使用传感器类对路径的偏移量进行补偿时              |

### 7.4.4 作业步骤

至执行样条插补为止的作业步骤如下所示。

表 7-32: 作业步骤

| No. | 步骤                     | 作业内容                                                                                                                                                                                          |
|-----|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | 样条文件的创建<br>(7.4.5 章)   | 使用 RT ToolBox2 的样条文件编辑画面、DXF 导入功能、SplWrt 指令的任意一个来创建样条文件。<br>(1) 通过位置的示教·导入功能登录路径点。<br>(2) 根据需要设定信号输出·数值设定。                                                                                    |
| 2   | 机器人程序的创建<br>(7.4.6 章)  | 创建包含样条插补动作的机器人程序。                                                                                                                                                                             |
| 3   | 动作的确认<br>(7.4.7 章)     | 使用模拟功能 <sup>注 1)</sup> 来确认样条插补的动作, 并适当修改路径点的位置·机器人程序。                                                                                                                                         |
| 4   | 保存至机器人控制器<br>(7.4.8 章) | 将机器人程序与样条文件写入至机器人控制器。                                                                                                                                                                         |
| 5   | 调整作业<br>(7.4.9 章)      | 使用 RT ToolBox2 的样条文件编辑画面对实际的系统进行微调。<br>(1) 实际的系统中, 通过调试操作来确认样条插补的动作。<br>(2) 将样条文件读取至样条文件编辑画面。<br>(3) 适当调整路径点的位置·路径补偿的设定。<br>(4) 将样条文件写入至控制器。<br>(5) 反复进行上述 (1) ~ (4), 并调整样条文件的数据, 直至达到所希望的路径。 |
| 6   | 运行                     | 运行机器人程序, 并执行样条插补。                                                                                                                                                                             |

注 1) RT ToolBox2 mini 版中无法使用。

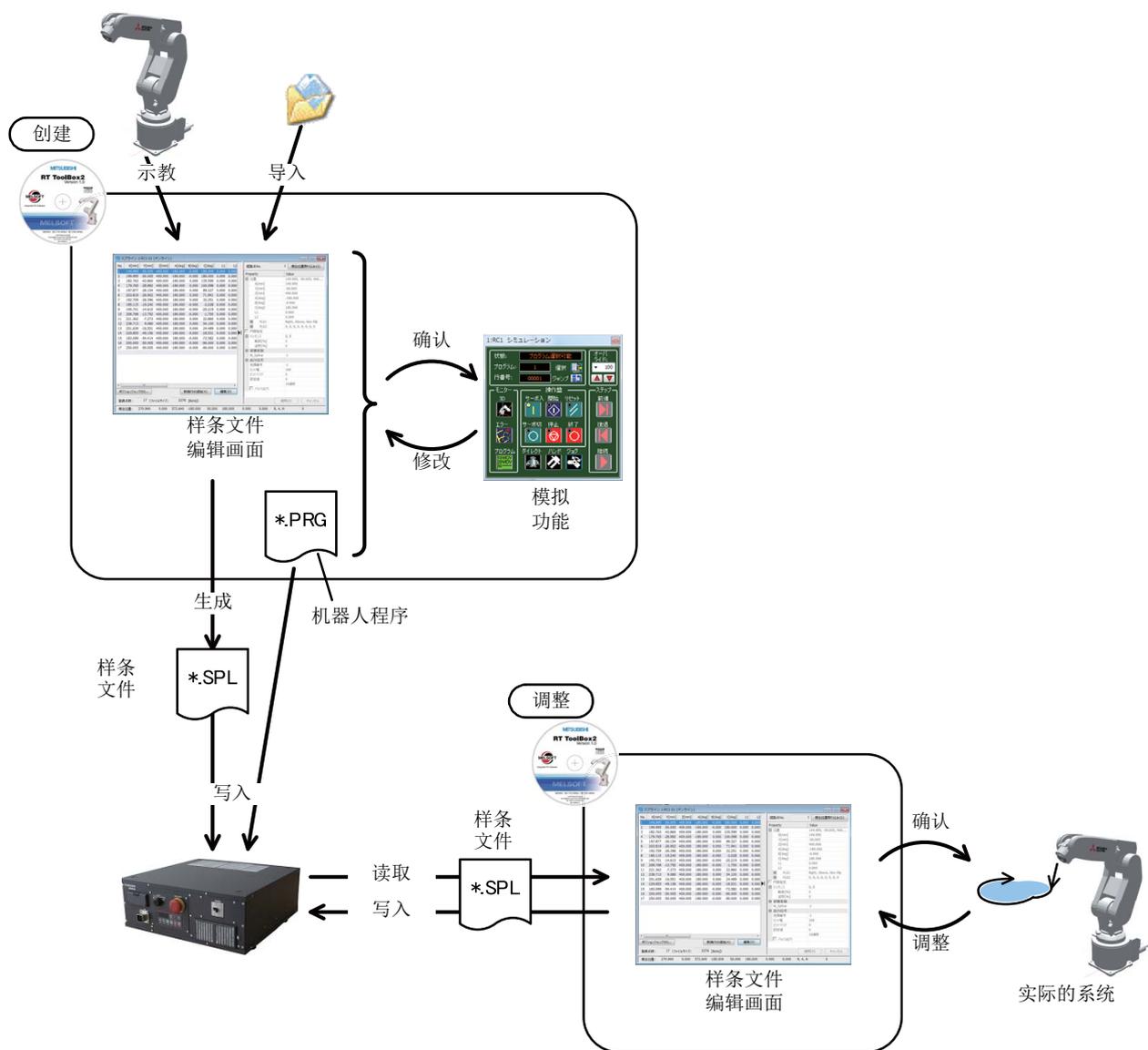


图 7-29: 作业内容的示意图

### 7.4.5 样条文件的创建

使用 RT ToolBox2 的样条文件编辑画面、DXF 导入功能、SplWrt 指令创建样条文件。以下对样条文件的创建方法进行说明。

#### (1) 新建

##### ■在计算机内新建

选择要新建的工程的 [ 离线 ] → [ 样条 ], 点击鼠标右键。将显示上下文菜单, 点击 [ 新建 ] 后, 样条文件编辑画面即启动。

(点击菜单 [ 文件 ] → [ 新建 ] 也可启动。)

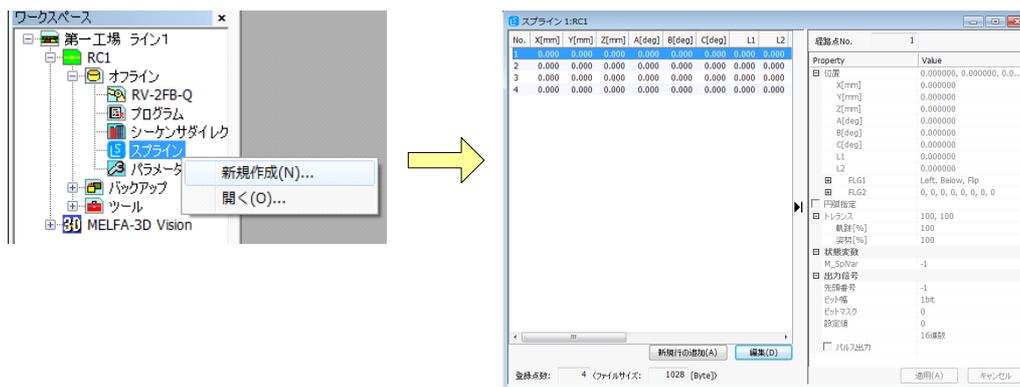


图 7-30: 在计算机内新建

##### ■在控制器内新建

选择要新建的工程的 [ 在线 ] → [ 样条 ], 点击鼠标右键。将显示上下文菜单, 点击 [ 新建 ] 后, 样条文件编辑画面即启动。

(点击菜单 [ 文件 ] → [ 新建 ] 也可启动。)

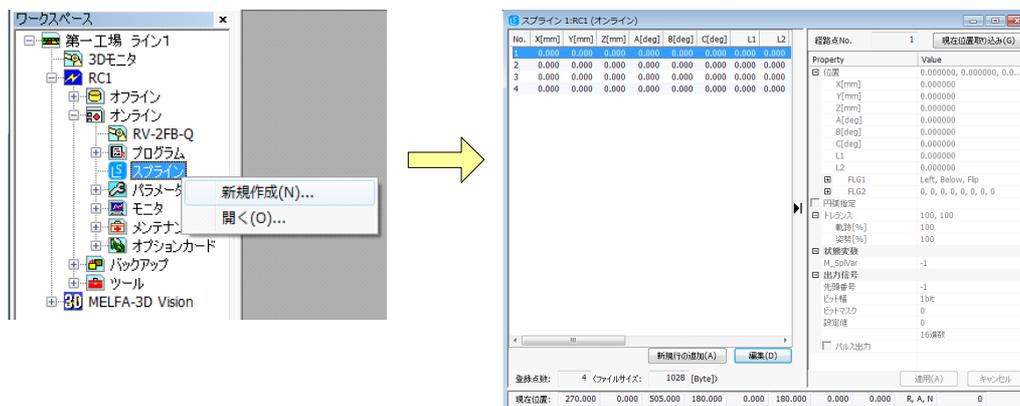


图 7-31: 在控制器内新建

#### ◆◆◆不支持样条插补的机器人时◆◆◆

对象工程的机器人为不支持样条插补的机型时, [ 在线 ] 工程树中不显示 [ 样条 ]。 [ 离线 ] 工程树中, 始终显示 [ 样条 ]

## (2) 通过 DXF 文件导入功能创建

该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。此外，DXF 文件支持至 AutoCAD2014。

## 1) DXF 文件选择

在工程树中选择 [ 工具 ] → [ DXF 文件导入 ] 来显示上下文菜单。选择 [ 打开 ] 菜单后，即显示打开 DXF 文件的对话框。

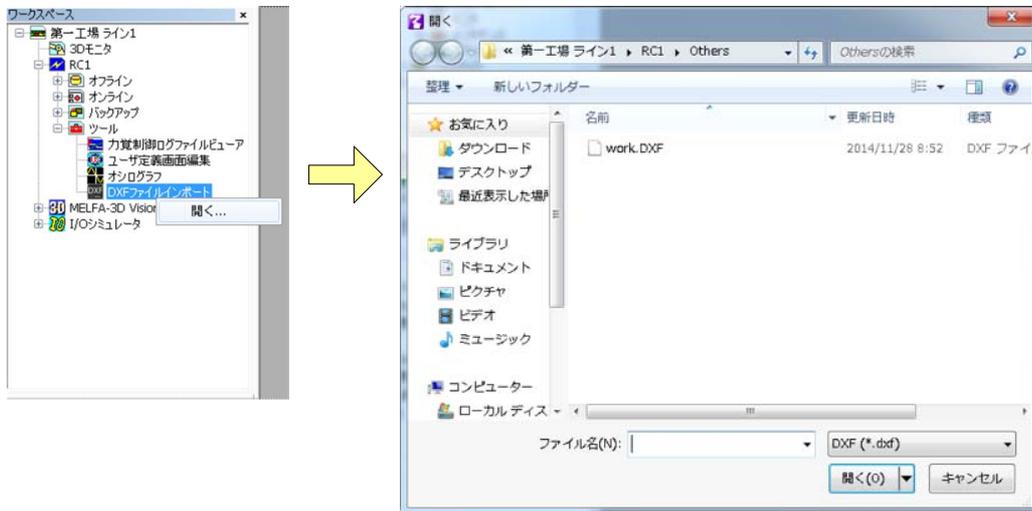


图 7-32: 文件选择画面

所选择的 DXF 文件的内容将显示在 DXF 文件阅读器中。

※DXF 文件中可导入的图形数据如下所示。

- LINE ( 线段 )
- ARC ( 圆弧 )
- CIRCLE ( 圆形 )
- SPLINE ( 样条 )
- POLYLINE ( 折线 )/LWPOLYLINE( 细折线 )

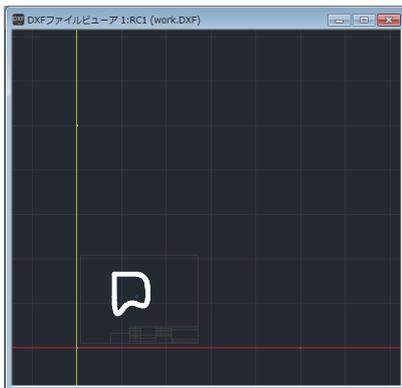


图 7-33: DXF 文件阅读器

## 2) 画层选择

从 [画层] 页的 [画层一览] 列表中选择要导入的画层。

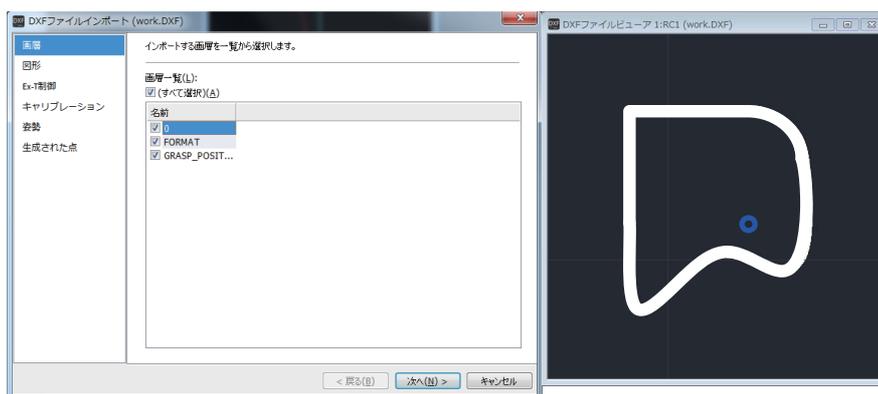


图 7-34: 画层选择画面

## 3) 图形选择

在 [图形] 页中, 选择要导入的图形。

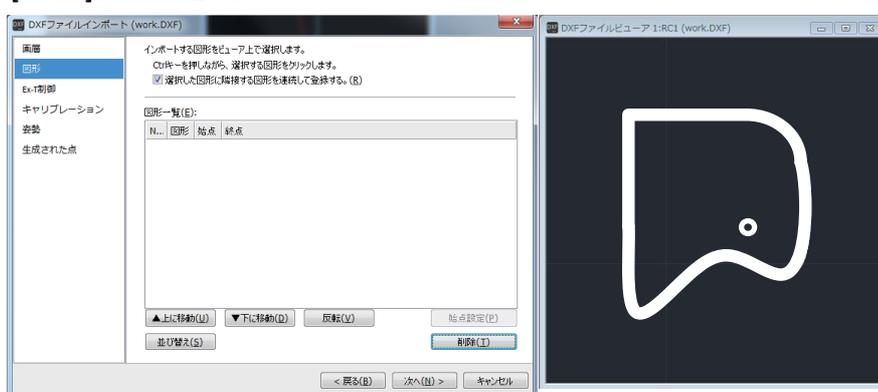


图 7-35: 图形选择画面

如按压 **Ctrl** 按键的同时点击 DXF 文件中要导入的图形, 则图形会被登录至 [图形一览] 列表。

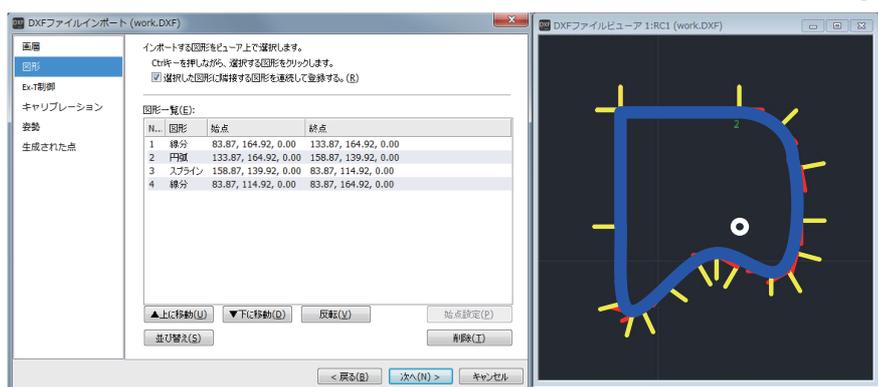


图 7-36: 图形选择

从起始的图形开始按顺序显示。前后的图形的始点 / 终点连续的图形将显示为蓝色。不连续的图形将显示为紫色。



图 7-37: 图形选择

从 [ 图形一览 ] 列表中选择图形并点击 [ 上移 ]/[ 下移 ] 按钮，即可调换所选择的图形的位置。

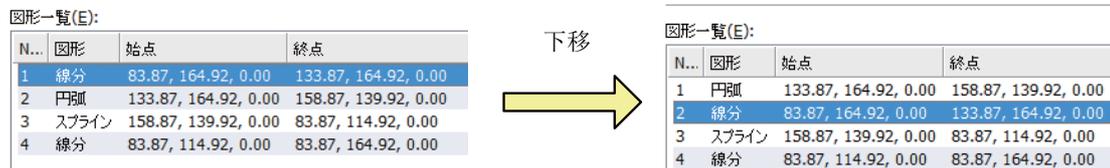


图 7-38: 图形选择 (下移)

从 [ 图形一览 ] 列表中选择图形并点击 [ 反转 ] 按钮，即可调换所选择图形的始点与终点。

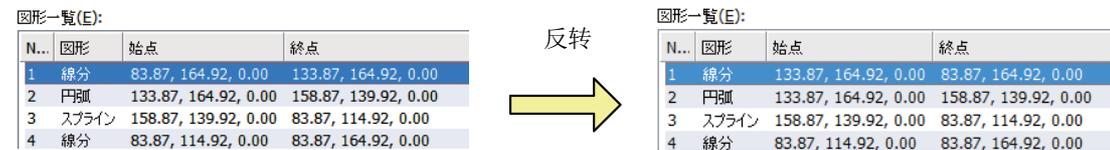


图 7-39: 图形选择 (反转)

点击 [ 排序 ] 按钮，即显示确认对话框。

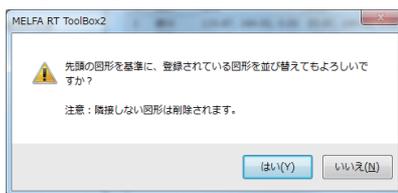


图 7-40: 排序确认对话框

点击确认对话框的 [ 是 ] 按钮，即以起始的图形为基准对登录在 [ 图形一览 ] 列表中的图形执行排序。  
※ [ 图形一览 ] 列表中登录有不相邻的图形时，该图形将被删除。



图 7-41: 图形选择 (排序)

#### 4) Ex-T 控制设定

[Ex-T 控制] 页中，对导入的图形的夹持位置与要登录在样条文件中的 Ex-T 坐标进行设定。

此处，使用 Ex-T 样条插补时，勾选 [使用 Ex-T 控制。] 勾选框。使用常规的样条插补时，不勾选勾选框。

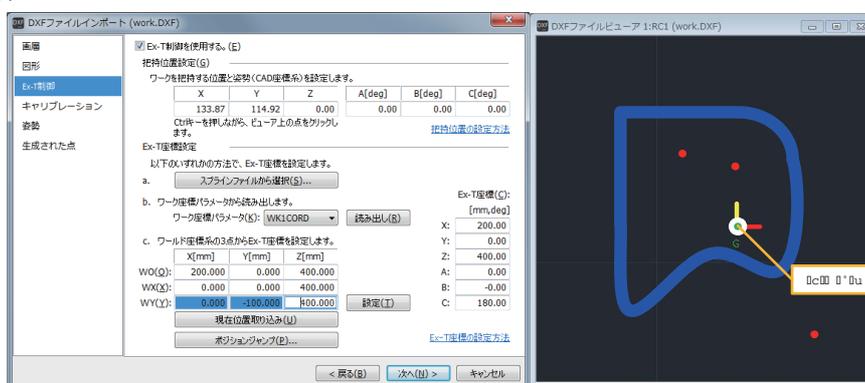


图 7-42: Ex-T 控制设定画面

如按压 **Ctrl** 按键的同时点击阅读器上的点（红色），则所选择的点被登录为夹持位置。

※ 画层 “GRASP\_POSITION” 的点 / 圆形（中心点）显示为夹持位置（X、Y、Z）的初始值。

Ex-T 坐标的设定方法有以下 3 种。

- 从现有的样条文件中选择  
点击 [从样条文件中选择] 按钮，即显示样条文件的选择画面。

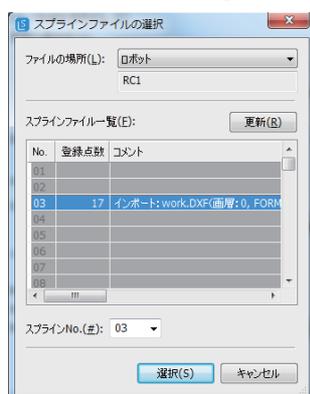


图 7-43: 样条文件的选择画面

读取保存在所选择的样条文件中的 Ex-T 坐标数据。

- 从工件坐标参数中读取  
对从下拉列表中读取的参数进行设定，并点击 [读取] 按钮，即读取所选择的参数中登录的工件坐标数据。
- 从世界坐标系的 3 点开始设定  
通过设定 Ex-T 坐标系的原点 (WO)、Ex-T 坐标系的 +X 轴上的位置 (WX)、Ex-T 坐标系的 X-Y 平面上的 +Y 轴侧的位置 (WY) 来设定 Ex-T 坐标数据。

#### 5) 校准

在 [校准] 页中，为了将坐标系从 CAD 坐标系转换为机器人的世界坐标系，要执行校准。

使用 Ex-T 样条插补时，不需要校准。



图 7-44: 校准画面

如按压 **Ctrl** 按键的同时点击阅读器上的点（红色），则所选择的点被登录为 CAD 坐标系的 CO/CX/CY。

※ 画层 “CALIBRATION” 的点 / 圆形（中心点）显示为 CAD 坐标系的初始值。

校准设定有以下 2 种方法。

- 从现有的样条文件中选择  
点击 [ 从样条文件中选择 ] 按钮，即显示与 Ex-T 控制设定同样的样条文件选择画面。读取通过画面所选择的样条文件中保存的校准数据。
- 从 CAD 坐标系、世界坐标系的 3 点开始设定  
通过设定 CAD 坐标系中的原点、+X 轴上的位置、X-Y 平面上的 +Y 轴侧的位置、世界坐标系中的原点、+X 轴上的位置、X-Y 平面上的 +Y 轴侧的位置来设定校准数据。

## 6) 姿势登录

在 [ 姿势 ] 页中，设定路径点的姿势。

将姿势类型选择为 [ 对控制点为固定 ]，即以各路径点的切线方向作为工具方向来登录姿势。



图 7-45: 姿势登录画面（对控制点为固定）

将姿势类型选择为 [ 固定 ]，即在所有的路径点上登录与开始位置相同的姿势。

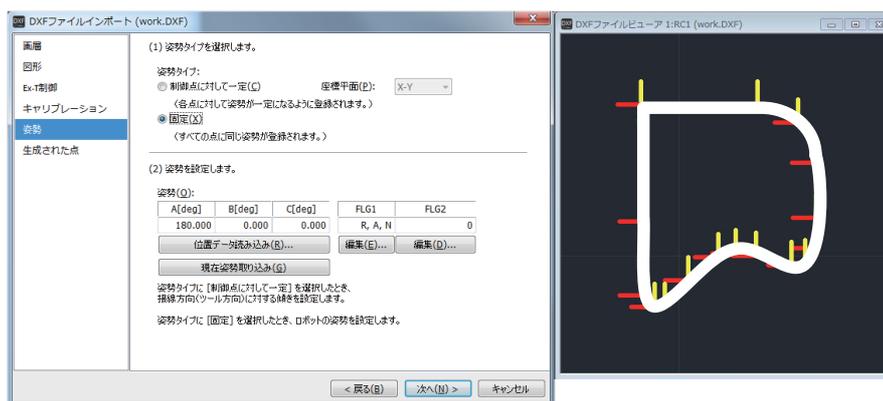


图 7-46: 姿势登录画面 (固定)

## 7) 输出至样条文件

在 [ 生成的点 ] 页中选择 [ 样条文件 ] 作为输出格式，并点击 [ 结束 ] 按钮，即向样条文件输出生成点阵数据。在导入结束对话框中点击 [ 是 ] 按钮，即显示样条文件的保存画面。通过指定要保存的样条文件名并点击 [ 保存 ] 按钮，可以保存样条文件。

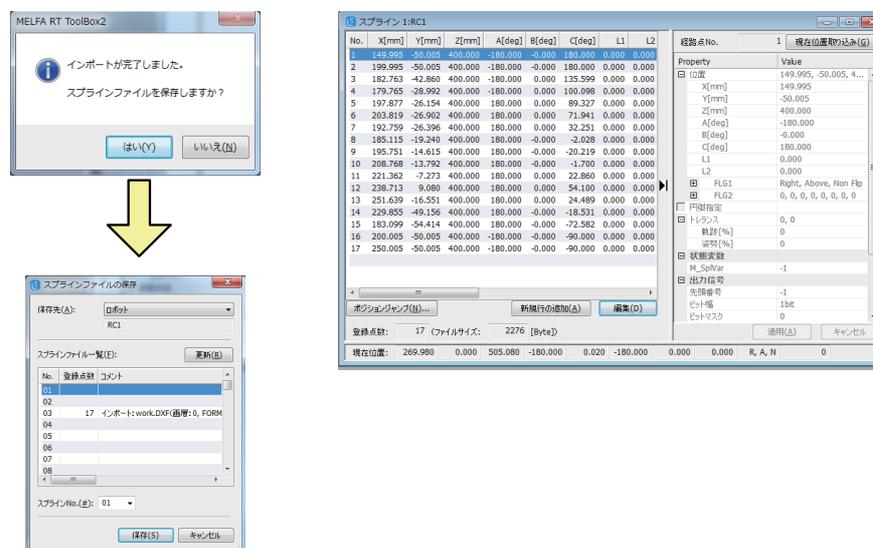
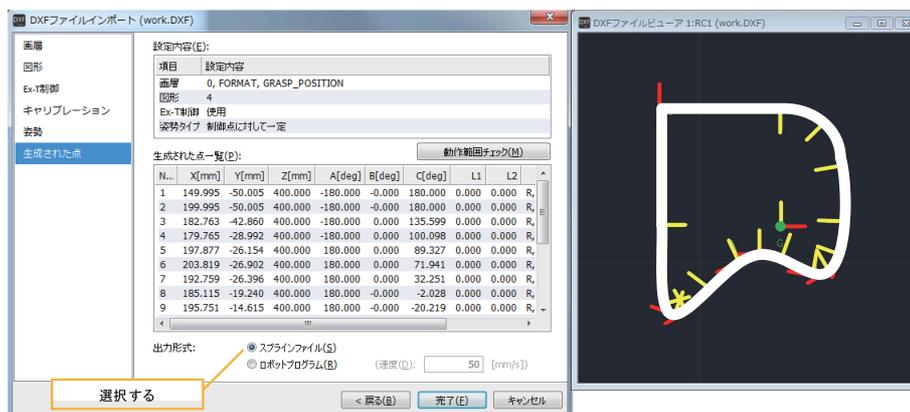


图 7-47: 生成的点画面

## (3) 用机器人语言新建

## ■用 SpiWrt 指令新建样条文件

在软件版本为 R6b 或 S6b 以上的机器人控制器中，使用 SpiWrt(样条写入)指令可以创建样条文件。指令的详细内容请参照 [SpiWrt \(Spline Write\)](#)。

在读取的文件中，对表 7-37 所示的数据以逗号分隔按顺序记述为路径点数据。此外，记述示例如图 7-80 所示。

第 1 行中应务必从 <X> 开始按顺序记述“数据识别标记”。未记述有“数据识别标记”时或各行的数据个数(20个)过多或过少时，会发生错误。

## ■用 SpiFWrt 指令编辑现有的样条文件

在软件版本为 R6b 或 S6b 以上的机器人控制器中，使用 SpiWrt(样条帧写入)指令可以在所保存的样条文件中登录帧转换坐标数据。指令的详细内容请参照 [SpiWrt \(Spline Write\)](#)。

## (4) 打开现有的样条文件

## ■打开保存在计算机中的样条文件

展开对象工程的 [离线] → [样条]。所保存的样条文件将显示在工程树中，双击要编辑的样条文件。(选择样条文件并点击鼠标的上下文菜单 [打开]，也可以打开。)

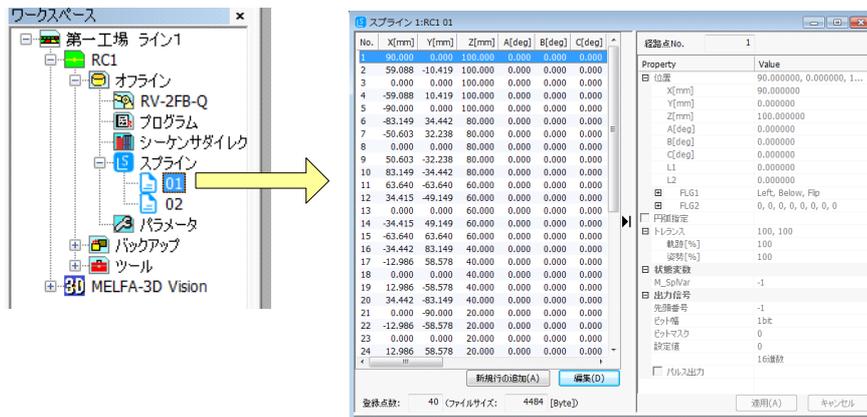


图 7-48: 打开保存在计算机中的样条文件

### ■ 打开控制器内的样条文件

展开对象工程的 [ 在线 ] → [ 样条 ]。保存在控制器中的样条文件将显示在工程树中，双击要编辑的样条文件。

(选择样条文件并点击鼠标的上下文菜单 [ 打开 ], 也可以打开。)

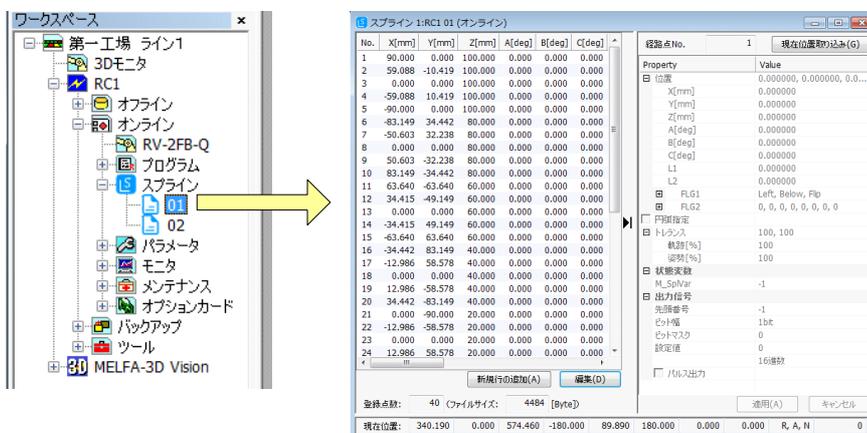


图 7-49: 打开控制器内的样条文件

### ■ 从一览中选择并打开样条文件

选择工程树的 [ 样条 ], 点击鼠标右键。

显示上下文菜单后, 点击 [ 打开 ], 即显示所保存的样条文件的一览。

(点击菜单 [ 文件 ] → [ 打开 ] 也可显示一览。)

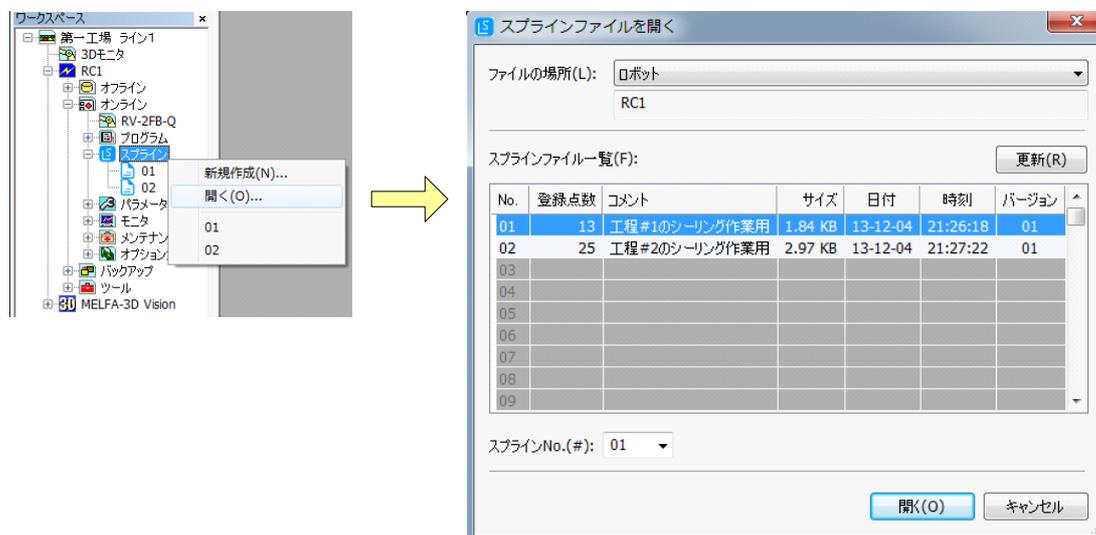


图 7-50: 从一览中选择并打开样条文件

一览中将显示登录在样条文件中的路径点数据的个数及注释, 以这些为参考选择对象样条文件, 按压 [ 打开 ] 按钮。

(5) 样条文件编辑画面的说明

以下对样条文件编辑画面进行说明。

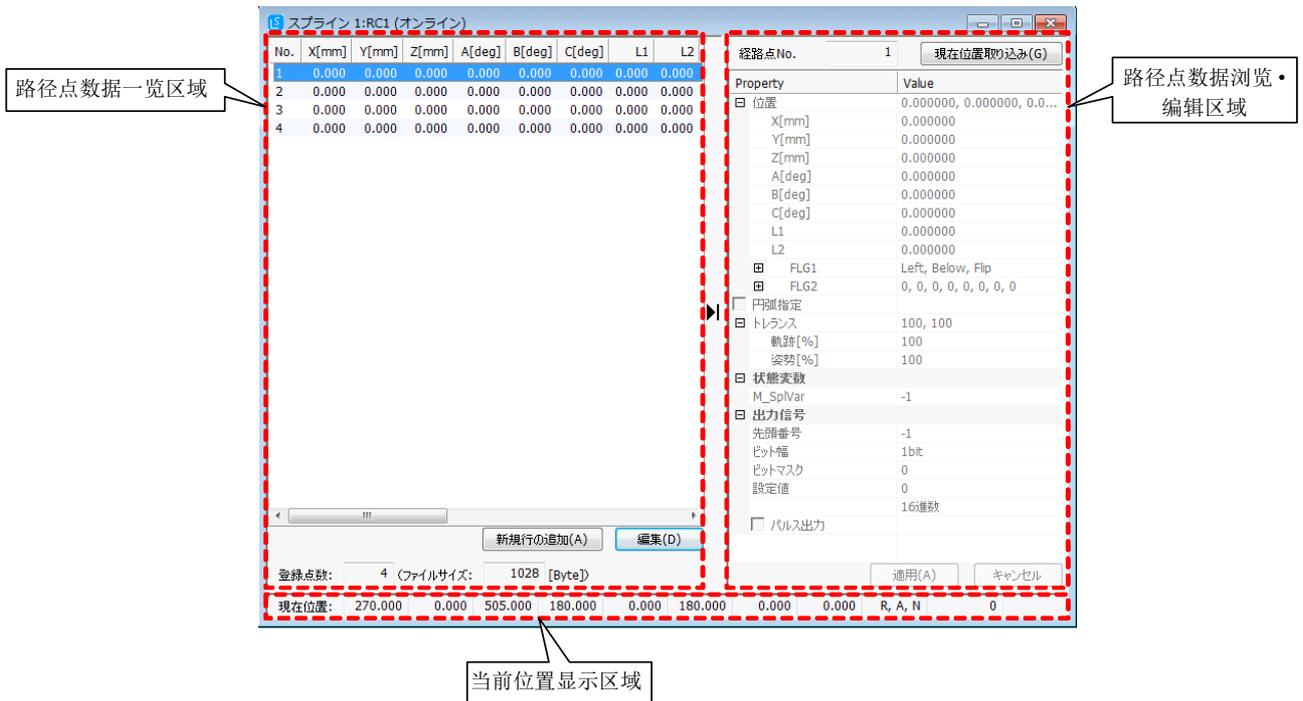


图 7-51: 样条文件编辑画面

样条文件编辑画面由 3 个区域构成。

■ 路径点数据一览区域

显示登录在样条文件中的路径点数据的一览。在该区域中进行以下的操作。

- 路径点数据的增加·删除
- 选择要编辑的路径点数据
- 登录点数、文件大小确认

点击位于样条文件编辑画面的中央附近的存储记号，路径点数据浏览·编辑区域即被存储，路径点数据一览区域的显示范围得以扩大。由此路径点间的数据变化的情况变得易于确认。

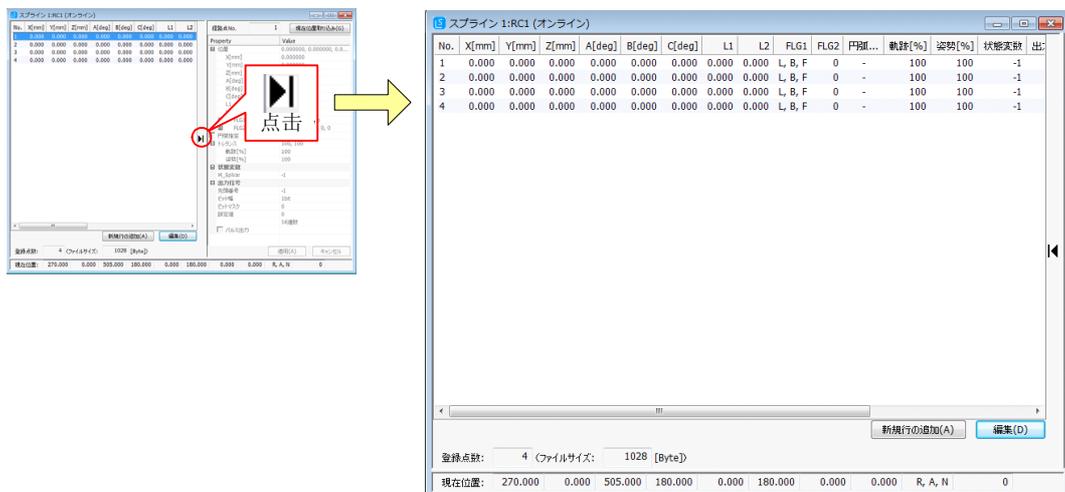


图 7-52: 路径点数据一览区域的显示范围扩大后的状态

可以通过计算机的键盘与鼠标的操作同时选择多个点的路径点数据。



图 7-53: 路径点数据的选择方法

### ■ 路径点数据浏览·编辑区域

在路径点数据一览区域中对所选择的路径点数据进行浏览·编辑。

如在路径点数据一览区域中选择路径点数据，则该内容以灰色的文字颜色显示在路径点数据浏览·编辑区域中。选择多个点时，将显示最后所选择的路径点数据。之后，如将路径点数据的编辑设为有效，则变为黑色的文字颜色，可编辑路径点数据。

(关于将编辑设为有效的方法，请参照第 617 页的「■ 路径点数据的编辑·登录」。)



图 7-54: 路径点数据浏览·编辑区域的浏览状态与编辑状态的切换

此外，如点击属性栏的存储 ([ - ]) · 展开 ([ + ]) 记号，则存储·展开并显示路径点数据。



图 7-55: 存储路径点数据的状态

### ■ 当前位置显示区域

显示对象工程中所连接的机器人的当前位置 ( 直角坐标系 ) 数据。

对象工程未与机器人连接 ( 离线模式 ) 时，不显示该区域与路径点数据浏览·编辑区域的 [ 获取当前位置 ] 按钮。

## (6) 样条文件编辑时的菜单栏

样条文件编辑时，菜单栏中将增加“文件(F)”·“编辑(E)”·“工具(T)”。



图 7-56: 样条文件编辑时的菜单栏

各个菜单的内容如下所示。

表 7-33: 增加的菜单的内容

| 菜单项目              |        | 说明                                     | 参照页码 |
|-------------------|--------|----------------------------------------|------|
| 文件                | 新建     | 启动用于新建样条文件的编辑画面。                       | 602  |
| 新規作成(N)... Ctrl+N | 打开     | 显示用于打开样条文件的专用画面。                       | 609  |
| 開く(O)... Ctrl+O   | 关闭     | 关闭激活的样条文件编辑画面。                         | -    |
| 閉じる(C)            | 保存     | 保存编辑中的样条文件。                            | 623  |
| 上書き保存(S) Ctrl+S   | 另存为    | 显示用于将编辑中的样条文件进行另存的专用画面。                | 622  |
| 名前を付けて保存(A)...    | 样条文件管理 | 打开样条文件管理画面。                            | 626  |
| スプラインファイル管理(M)    | 导入     | 从 CSV/MXT <sup>注1)</sup> 格式的文件获取路径点数据。 | 633  |
| インポート(I)...       | 导出     | 向 CSV 格式的文件写出路径点数据。                    | 636  |
| エクスポート(E)...      | 属性     | 为了进行注释的浏览·写入、文件版本的更改，显示样条文件的属性画面。      | 621  |
| プロパティ(P)...       |        |                                        |      |

| 菜单项目            |  | 说明             | 参照页码                                          |     |
|-----------------|--|----------------|-----------------------------------------------|-----|
| 编辑              |  | 撤销             | 取消对路径点数据最后适用的编辑内容并返回原来的状态。                    | 637 |
| 元に戻す(U) Ctrl+Z  |  | 恢复             | 再次反映通过 [ 撤销 ] 取消的内容。                          | 637 |
| やり直し(R) Ctrl+Y  |  | 剪切             | 将所选择的路径点数据存储为要剪切的数据。实施 [ 粘贴 ], 则原来的路径点数据将被剪切。 | 636 |
| 切り取り(T) Ctrl+X  |  | 复制             | 将所选择的路径点数据存储为要复制的数据。                          | 636 |
| コピー(C) Ctrl+C   |  | 粘贴             | 将实施了 [ 剪切 ]、[ 复制 ] 的路径点数据粘贴至指定的路径点数据。         | 636 |
| 貼り付け(P) Ctrl+V  |  | 复制 - 位置数据      | 将所选择的位置数据存储为要复制的数据。                           | 637 |
| コピー - 位置データ(V)  |  | 粘贴 - 位置数据      | 将存储的位置数据粘贴至指定的路径点数据。                          | 637 |
| 貼り付け - 位置データ(B) |  | 插入到所选行的前面      | 将新的路径点数据插入到所选择的路径点数据的前面。                      | 616 |
| 選択行の前に挿入(I)     |  | 所复制的路径点的插入     | 将实施了 [ 复制 ] 的路径点数据插入到指定的路径点数据的前面。             | 637 |
| コピーした経路点の挿入(S)  |  | 全选             | 将全部的路径点数据设为已选状态。                              | -   |
| すべて選択(A) Ctrl+A |  | 删除             | 删除所选择的路径点数据。                                  | 618 |
| 削除(D) Delete    |  | 新增行            | 在路径点数据的末尾增加新的路径点数据。                           | 616 |
| 新規行の追加(N)       |  | 编辑             | 将所选择的路径点数据设为可编辑状态。                            | 617 |
| 編集(E)           |  | FLG1 批量编辑      | 对所选择的路径点的 FLG1 的值进行批量编辑。                      | 618 |
| FLG1一括編集(1)...  |  | FLG2 批量编辑      | 对所选择的路径点的 FLG2 的值进行批量编辑。                      | 619 |
| FLG2一括編集(2)...  |  | 允许公差批量编辑       | 对所选择的路径点的允许公差量进行批量编辑。                         | 619 |
| トランス一括編集(O)...  |  | 圆弧批量指定         | 切换所选择的路径点的圆弧指定的 ON/OFF。                       | 619 |
| 円弧一括指定(L)       |  | 跳位             | 显示跳位功能的专用画面。                                  | 650 |
| 工具              |  | 插补设定           | 显示样条文件的插补设定画面。                                | 620 |
| ポジションジャンプ(P)... |  | Ex-T 控制设定      | 显示样条文件的 Ex-T 控制设定画面。                          | 621 |
| 補間設定(I)...      |  | 位置补偿           | 显示位置补偿功能的专用画面。                                | 645 |
| Ex-T制御設定(E)...  |  | 帧转换            | 显示帧转换功能的专用画面。                                 | 647 |
| 位置補正(A)...      |  | 动作范围检查         | 检查编辑中的样条文件是否在机器人的动作范围内。<br>※ 该功能仅在模拟时可以执行。    | 639 |
| フレーム変換(F)...    |  | 路径点检查          | 显示路径点检查功能的专用画面。                               | 638 |
| 動作範囲チェック(M)     |  | 算出最大速度         | 通过使用编辑中的样条文件的样条插补指令, 算出不会发生错误且可指定的最大速度。       | 639 |
| 経路点チェック(H)...   |  | 样条曲线           | 显示编辑中的样条文件的样条曲线。                              | 638 |
| 最大速度算出(S)       |  | 机器人程序 - 程序转换   | 将编辑中的样条文件转换为机器人程序。                            | 639 |
| スプライン曲線(C)      |  | 机器人程序 - 获取位置数据 | 将保存在机器人程序中的位置数据获取至编辑中的样条文件。                   | 641 |
| ロボットプログラム(R) ▶  |  |                |                                               |     |
| プログラム変換(C)...   |  |                |                                               |     |
| 位置データ取り込み(I)... |  |                |                                               |     |

注 1) MXT 文件是指具有从 MELFA-Works 输出的点阵数据的文件。

## (7) 路径点数据的内容

样条文件中登录的路径点数据的内容如下所示。

表 7-34: 路径点数据的内容

| 数据项目 |              | 内容                                                                       | 新增时的初始值                                           |   |
|------|--------------|--------------------------------------------------------------------------|---------------------------------------------------|---|
| 位置   |              | 通过直角坐标系 (Y、Z、A、B、C、L1、L2、FLG1、FLG2) 指定路径点的位置数据。<br>距离的单位为 mm，角度的单位为 deg。 | X ~ L2: 0.0<br>FLG1: Left, Below, Flip<br>FLG2: 0 |   |
| 圆弧指定 |              | 指定圆弧轨迹的路径点。<br>设定范围: 有检查 (指定有效)、无检查 (指定无效)                               | 无检查                                               |   |
| 允许公差 | 位置           | 指定位置 (轨迹) 的允许公差量。<br>设定范围: 0 ~ 100 [%]                                   | 100                                               |   |
|      | 姿势           | 指定姿势的允许公差量。<br>设定范围: 0 ~ 100 [%]                                         | 100                                               |   |
| 状态变量 |              | 指定要通过状态变量 M_SplVar 浏览的值。<br>设定范围: -1 ~ 32767 (-1 表示未设定。)                 | -1                                                |   |
| 输出信号 | 起始号码         | 指定输出信号的起始号码。<br>设定范围: -1 ~ 32767 (-1 表示信号输出无效。)                          | -1                                                |   |
|      | 位宽           | 指定要输出的信号的位宽。<br>设定范围: 1、8、16、32 [位]                                      | 1                                                 |   |
|      | 位掩码          | 指定将信号输出看作有效的位的掩码模式。                                                      |                                                   | 0 |
|      |              | 位宽                                                                       | 设定范围 [16 进制]                                      |   |
|      |              | 1                                                                        | 0、1                                               |   |
| 8    |              | 0 ~ FF                                                                   |                                                   |   |
| 16   | 0 ~ FFFF     |                                                                          |                                                   |   |
| 32   | 0 ~ FFFFFFFF |                                                                          |                                                   |   |
| 设定值  | 指定信号输出的数据。   |                                                                          | 0                                                 |   |
|      | 位宽           | 设定范围 [16 进制]                                                             |                                                   |   |
|      | 1            | 0、1                                                                      |                                                   |   |
|      | 8            | 0 ~ FF                                                                   |                                                   |   |
| 16   | 0 ~ FFFF     |                                                                          |                                                   |   |
| 32   | 0 ~ FFFFFFFF |                                                                          |                                                   |   |
| 脉冲输出 |              | 指定脉冲输出。<br>设定范围: 有检查 (指定有效)、无检查 (指定无效)                                   | 无检查                                               |   |

## (8) 样条文件的编辑

## ■ 路径点数据的增加

按压路径点数据一览表区域的 [ 新增行 ] 按钮，则将新的路径点数据增加至一览的末尾。

初始值为表 7-34 的“新增时的初始值”所示的值。

(点击菜单 [ 编辑 ] → [ 新增行 ] 也可以增加。)

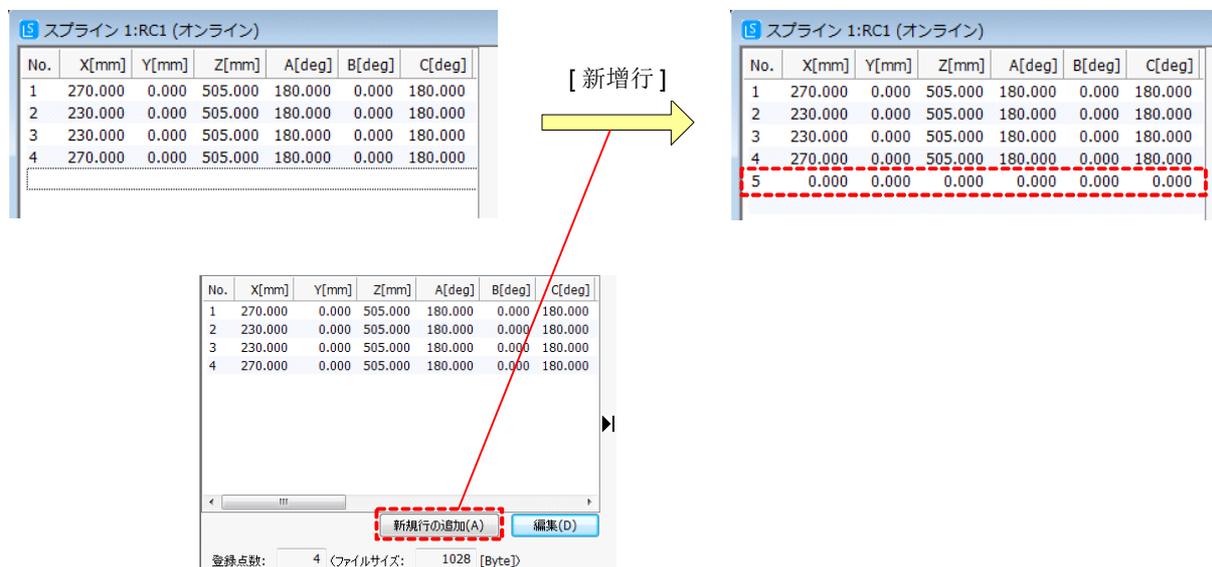


图 7-57: 路径点数据的增加

此外，选择任意的路径点数据并点击鼠标右键。显示上下文菜单后，点击 [ 插入到所选行的前面 ]，即可将新的路径点数据插入到所选择的路径点数据的前面。

(点击菜单 [ 编辑 ] → [ 插入到所选行的前面 ] 也可以插入。)

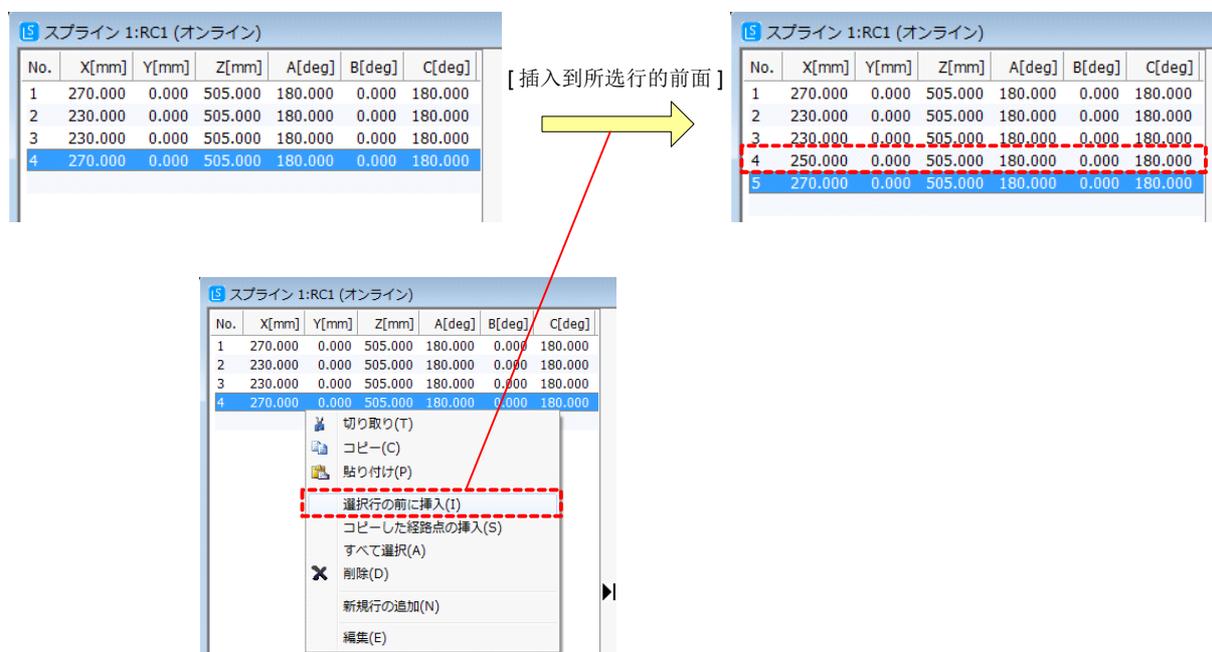


图 7-58: 路径点数据的插入

此时，根据所选择的路径点数据的行，所插入的路径点数据的初始值为表 7-35 所示的值。

表 7-35: 所插入的路径点数据的初始值

| 所选行   | 路径点数据的初始值                |                          |                             |
|-------|--------------------------|--------------------------|-----------------------------|
| 起始行   | 与表 7-34 的“新增时的初始值”为相同的值。 |                          |                             |
| 起始行以外 | 位置                       | X ~ L2                   | 为所选择的路径点数据与前 1 行的路径点数据的平均值。 |
|       |                          | FLG1、FLG2                | 与所选择的路径点数据为相同的值。            |
|       | 位置以外                     | 与表 7-34 的“新增时的初始值”为相同的值。 |                             |

### ■ 路径点数据的编辑·登录

编辑路径点数据时，在路径点数据一览区域中选择并双击编辑对象的路径点数据。通过双击，对象的路径点数据变为编辑有效状态，路径点数据浏览·编辑区域变为编辑状态。

(通过路径点数据一览区域下部的 [编辑] 按钮、菜单 [编辑] → [编辑] 也可将编辑设为有效。)

在路径点数据浏览·编辑区域中选择要编辑的项目并设定值。

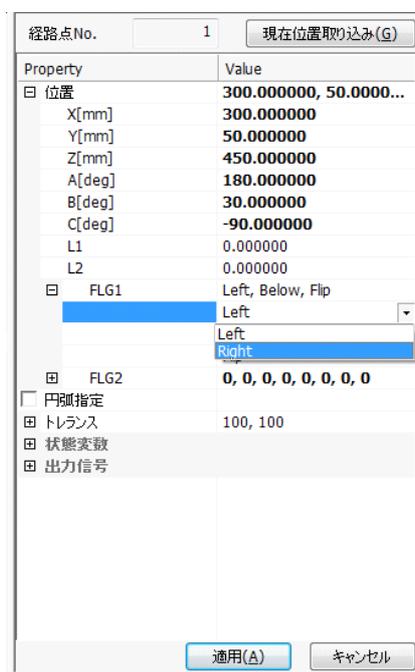


图 7-59: 路径点数据的编辑

与机器人连接时，在路径点数据浏览·编辑区域上部将显示 [获取当前位置] 按钮。按压该按钮，即将机器人的当前位置获取为路径点的位置数据。

要将设定的内容登录为路径点数据时，按压 [适用] 按钮。只要不按压 [适用] 按钮，设定的内容就不会被登录至路径点数据。不登录时，按压 [取消] 按钮。按压 [适用]·[取消] 按钮，编辑即变为无效，路径点数据浏览·编辑区域即变为浏览状态。

### ◇◆◇ 编辑状态下的路径点数据的选择 ◇◆◇

路径点数据浏览·编辑区域为编辑状态时，将无法在路径点数据一览区域中选择路径点数据。选择其他的路径点数据时，请按压 [适用]·[取消] 按钮将路径点数据浏览·编辑区域切换为浏览状态。

### ■ 路径点数据的删除

在路径点数据一览区域中选择路径点数据，并点击鼠标右键。显示上下文菜单后，点击[删除]，即可删除所选择的路径点数据。

(点击菜单[编辑]→[删除]也可以删除。)

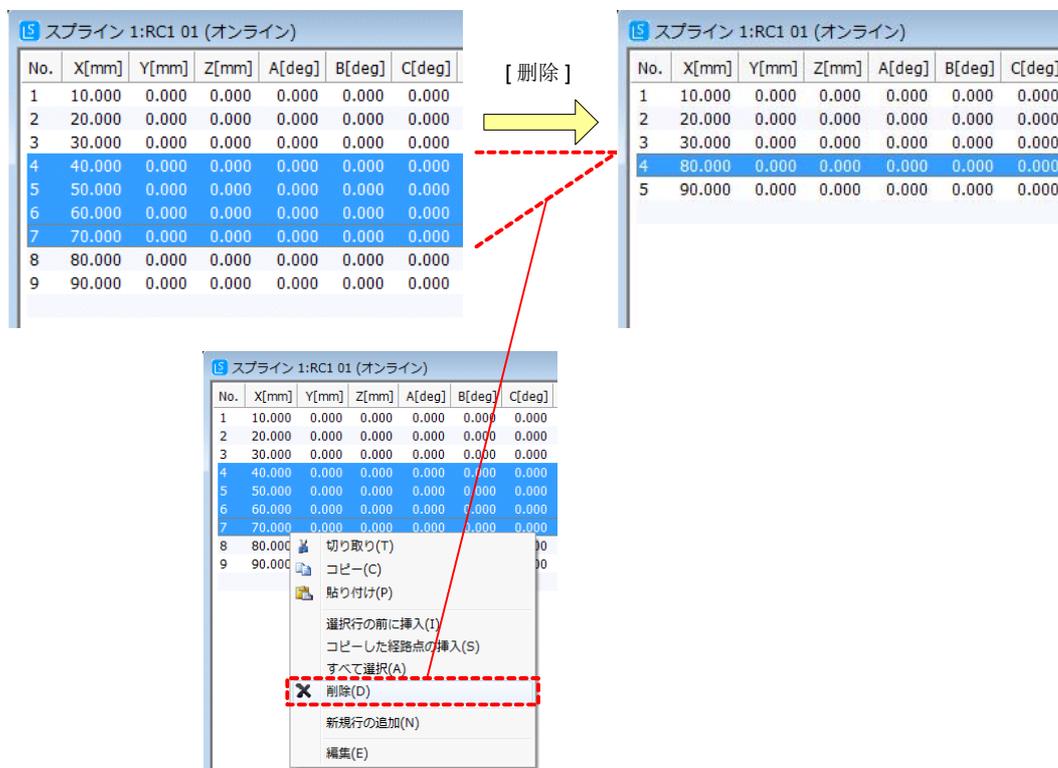


图 7-60: 路径点数据的删除

### ■ FLG1 批量编辑

对所选择的路径点的 FLG1 的值进行批量编辑。该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。

要批量编辑 FLG1 的值，需要在路径点数据一览区域中选择编辑对象的路径点数据并点击鼠标右键。显示上下文菜单后，点击[FLG1 批量编辑]，即显示 FLG1 批量编辑画面。

(点击菜单[编辑]→[FLG1 批量编辑]也可以显示。)

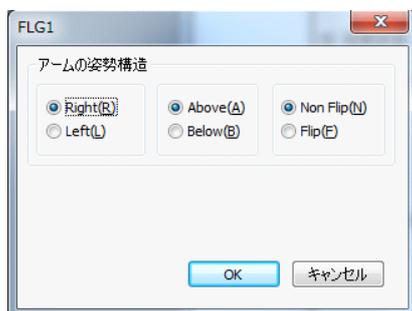


图 7-61: FLG1 批量编辑画面

如更改机械臂的姿势结构后按压[OK]按钮，则更改内容会被反映至所选择的路径点。按压[取消]按钮时，不反映更改内容而关闭 FLG1 批量编辑画面。

### ■ FLG2 批量编辑

对所选择的路径点的 FLG2 的值进行批量编辑。该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。

要批量编辑 FLG2 的值，需要在路径点数据一览区域中选择编辑对象的路径点数据并点击鼠标右键。显示上下文菜单后，点击 [FLG2 批量编辑]，即显示 FLG2 批量编辑画面。

（点击菜单 [编辑] → [FLG2 批量编辑] 也可以显示。）

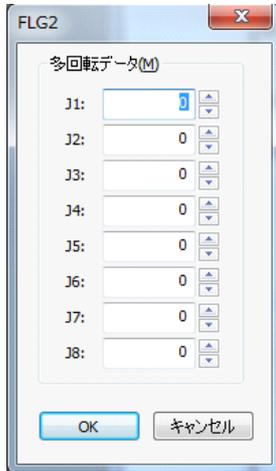


图 7-62: FLG2 批量编辑画面

如更改各轴的多旋转数据后按压 [OK] 按钮，则更改内容会被反映至所选择的路径点。按压 [取消] 按钮时，不反映更改内容而关闭 FLG2 批量编辑画面。

### ■ 允许公差的批量编辑

对所选择的路径点的允许公差量进行批量编辑。该功能可在 RT ToolBox2 Ver.3.20W 以上的版本中使用。

要批量编辑允许公差量，需要在路径点数据一览区域中选择编辑对象的路径点数据并点击鼠标右键。显示上下文菜单后，点击 [允许公差批量编辑]，即显示允许公差批量编辑画面。（点击菜单 [编辑] → [允许公差批量编辑] 也可以显示。）

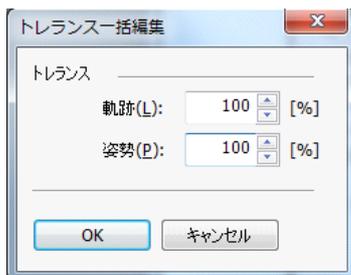


图 7-63: 允许公差的批量编辑画面

如更改允许公差量后按压 [OK] 按钮，则更改内容会被反映至所选择的路径点。按压 [取消] 按钮时，不反映更改内容而关闭允许公差批量编辑画面。

### ■ 圆弧批量指定

对所选择的路径点的圆弧指定进行批量编辑。该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。

要批量编辑圆弧指定，需要在路径点数据一览区域中选择编辑对象的路径点数据并点击鼠标右键。显示上下文菜单后，点击 [圆弧批量指定]，即可切换所选择的路径点的圆弧指定的 ON/OFF。（点击菜单 [编辑] → [圆弧批量指定] 也可以编辑。）

| No. | X[mm]   | Y[mm]   | Z[mm]   | A[deg]   | B[deg] | C[deg]  | L1    | L2    | FLG1    | FLG2 | 圆弧... | 軌跡[%] | 姿勢[%] | 状態変数 | 出力信 |
|-----|---------|---------|---------|----------|--------|---------|-------|-------|---------|------|-------|-------|-------|------|-----|
| 1   | 149.995 | -50.005 | 400.000 | -180.000 | -0.000 | 180.000 | 0.000 | 0.000 | R, A, N | 0    | -     | 0     | 0     | -1   |     |
| 2   | 199.995 | -50.005 | 400.000 | -180.000 | -0.000 | 180.000 | 0.000 | 0.000 | R, A, N | 0    | ON    | 100   | 100   | -1   |     |
| 3   | 182.763 | -42.860 | 400.000 | -180.000 | 0.000  | 135.599 | 0.000 | 0.000 | R, A, N | 0    | ON    | 100   | 100   | -1   |     |
| 4   | 179.765 | -28.992 | 400.000 | -180.000 | 0.000  | 100.098 | 0.000 | 0.000 | R, A, N | 0    | -     | 100   | 100   | -1   |     |

图 7-64: 圆弧批量指定

### ■ 插补设定

设定样条插补的“动作模式”·“取消角度”·“块长比率”。

点击菜单 [ 工具 ] → [ 插补设定 ], 激活的样条文件编辑画面的样条文件中所设定的插补设定的内容即显示在设定画面中。

補間設定

動作モード(O): 線速一定

スプラインキャンセル角度(C): 120 [deg] (0: 機能無効)

直線とみなすブロック長比率(S): 8 (0: 機能無効)

OK キャンセル

图 7-65: 插补设定画面

如更改设定值后按压 [OK] 按钮, 即存储更改内容, 关闭设定画面。按压 [取消] 按钮时, 不存储更改内容而关闭设定画面。各项目的设定范围如表 7-36 所示。

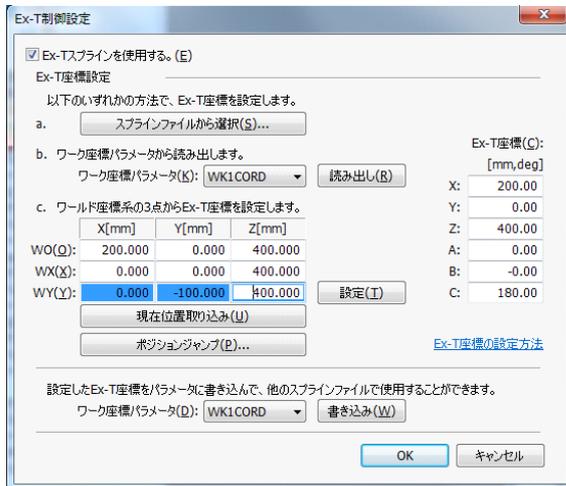
表 7-36: 插补设定项目的设定范围

| 项目   | 设定范围        | 初始值  |
|------|-------------|------|
| 动作模式 | 线速固定 / 线速可变 | 线速固定 |
| 取消角度 | 0 ~ 180 (度) | 120  |
| 块长比率 | 0 ~ 100 [倍] | 8    |

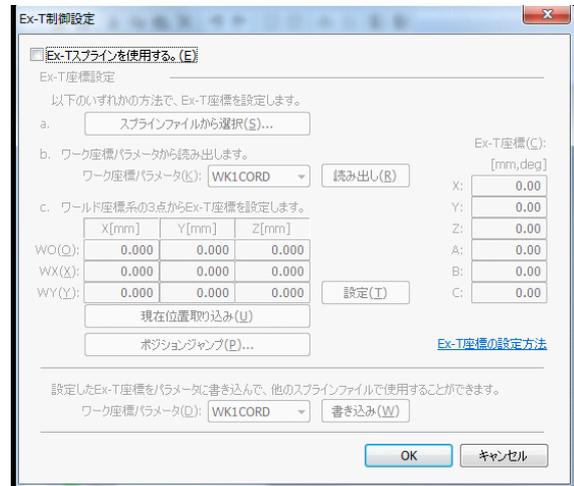
### ■ Ex-T 控制设定

进行样条插补的“Ex-T 控制设定”。

点击菜单 [ 工具 ] → [ Ex-T 控制设定 ]，激活的样条文件编辑画面的样条文件中所设定的 Ex-T 控制设定的内容即显示在设定画面中。



有 Ex-T 控制设定



无 Ex-T 控制设定

图 7-66: Ex-T 控制设定画面

### ■ 注释的写入

可在样条文件中写入注释。注释中所写入的内容，将在样条文件一览的注释栏中显示。事先将作业内容及条件等作为注释写入，将有助于选择样条文件。

点击菜单 [ 文件 ] → [ 属性 ]，激活的样条文件编辑画面的样条文件中所写入的注释即显示在属性画面中。

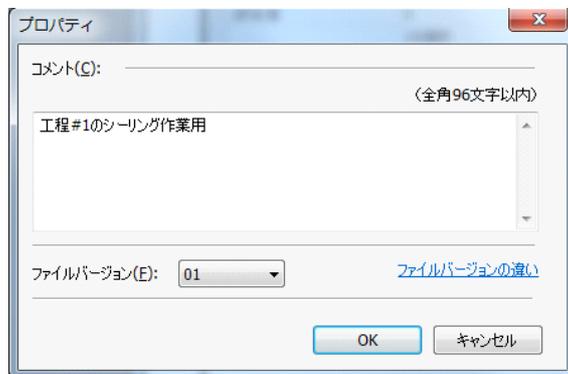


图 7-67: 属性画面

注释最多可写入 96 个全角字符（192 个半角字符）。但是，换行无法输入。初始状态为未写入。

写入注释后按压 [OK] 按钮，即存储该内容，关闭属性画面。按压 [取消] 按钮时，不存储所写入的注释而关闭属性画面。

### ■ 文件版本

可更改样条文件的版本。

点击菜单 [ 文件 ] → [ 属性 ]，激活的样条文件编辑画面的样条文件的文件版本即显示在属性画面中。

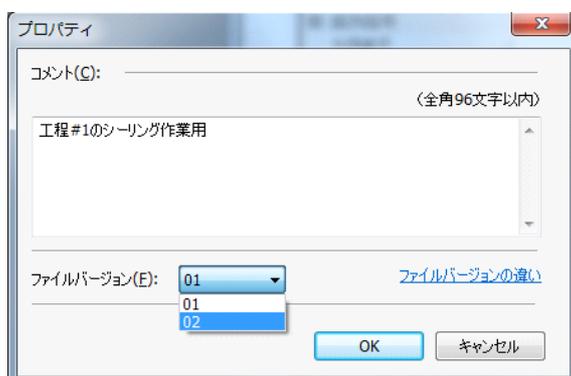


图 7-68: 属性画面

更改文件版本后按压 [OK] 按钮，即存储该内容，关闭属性画面。按压 [取消] 按钮时，不存储文件版本而关闭属性画面。

※ 文件版本导致的不同，请参照第 593 页的「■ 圆弧指定」。

### (9) 样条文件的保存

将样条文件编辑画面中编辑的内容反映并保存至样条文件。

#### ■ 点击菜单 [ 文件 ] → [ 另存为 ]

显示样条文件的保存画面。

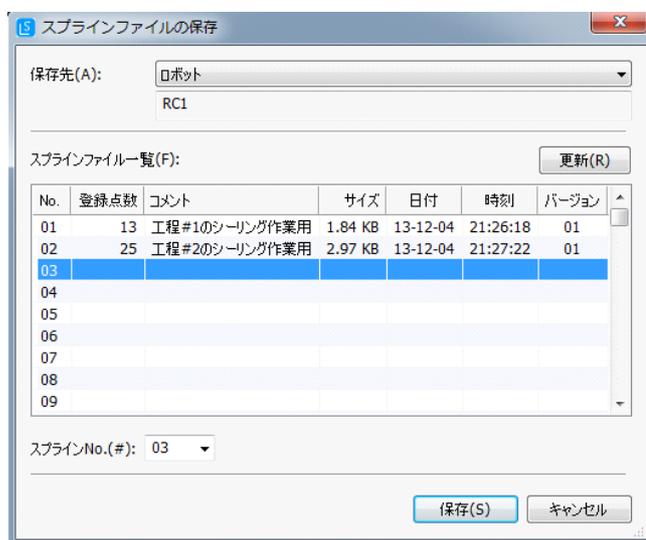


图 7-69: 样条文件的保存画面

从“计算机”·“机器人”中选择保存目标。（离线模式时，无法选择“机器人”。）

然后，从样条文件一览中选择要保存的样条号后按压 [保存] 按钮，即将编辑内容保存至样条文件并关闭保存画面。

按压 [取消] 按钮时，不保存而关闭保存画面。

## ◇◆◇关于样条文件保存目标文件夹◇◆◇

计算机内的样条文件以工作区的工程为单位进行管理。保存目标的文件夹如下。  
工作区作业文件夹\工程名\Spline。

## ◇◆◇关于样条文件的名称◇◆◇

样条文件的名称由以下的格式来决定。

SPLFILE\*\*.SPL (固定为大写字母)

“\*\*”中有 01 ~ 99 的样条号, 与 MvSpl 指令 /EMvSpl 指令的自变量 < 样条号 > 相对应。

样条文件编辑画面中将显示为 01 ~ 99 的样条号, 无须在意文件名。



## 注意

如在 RT ToolBox2 的文件管理画面或 Microsoft Windows 的资源管理器等中将样条文件的文件名更改为其他的名称, 将无法执行样条插补。  
请勿更改样条文件的名称。

## ■ 点击菜单 [文件] → [保存]

如果是现有的样条文件, 则将编辑内容保存至对象的样条文件。

如果是新建的样条文件, 则动作将与点击 [另存为] 时相同。

## ■ 按压样条文件编辑画面的关闭按钮

按压样条文件编辑画面右上角的 [×] (关闭) 按钮, 即显示确认对话框。

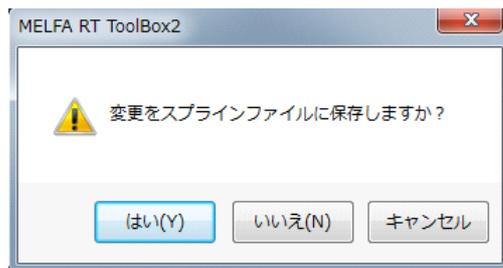


图 7-70: 保存确认对话框

按压确认对话框的 [是] 按钮, 所编辑的内容即反映并保存至样条文件。

如果是现有的样条文件, 则将编辑内容保存至对象的样条文件后, 关闭样条文件编辑画面。

如果是新建的样条文件, 则将显示样条文件的保存画面。

从样条文件一览中选择保存目标的样条号后按压 [保存] 按钮, 即将编辑内容保存至样条文件并关闭样条文件编辑画面。按压 [取消] 按钮时, 不保存而关闭保存画面, 返回样条文件编辑画面。

按压确认对话框的 [否] 按钮, 即删除所编辑的内容并关闭样条文件编辑画面。

如按压确认对话框的 [取消] 按钮, 则不保存而返回样条文件编辑画面。



## 注意

样条文件编辑画面中所编辑·存储的内容, 如不进行保存, 则不会反映至样条文件。  
编辑作业中应频繁进行保存至样条文件的操作。



**注意**

对样条插补中正在使用的样条文件无法进行保存。如要进行保存，则会发生 L2610（无法更改样条文件）错误。请勿对正在使用的样条文件进行保存操作。

### (10) 样条文件的删除

选择显示在工程树上的删除对象的样条文件，点击鼠标右键。  
显示上下文菜单后，点击 [删除]，即显示确认对话框。



图 7-71: 样条文件的删除

按压 [是] 按钮，即删除所指定的样条文件。

按压 [否] 按钮，即取消删除操作。

### (11) 样条号的更改

选择显示在工程树上的更改对象的样条文件，点击鼠标右键。  
显示上下文菜单后，点击 [样条 No. 的更改]，即显示用于指定新样条号的对话框。

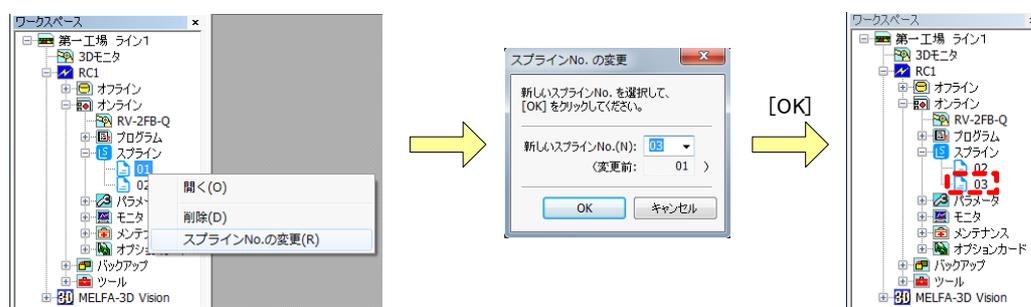


图 7-72: 样条号的更改

如指定新样条号后按压 [OK] 按钮，即更改所指定的样条文件的样条号。

按压 [取消] 按钮，即取消样条号的更改操作。

(12) 样条文件的复制

RT ToolBox Ver.3.20W 以上版本中，可以在项目树上复制样条文件。

在工程树上选择要复制的样条文件，拖放至复制目标的工程树的 [ 样条 ]/[ 在线 ]/[ 离线 ]，即弹出确认对话框。

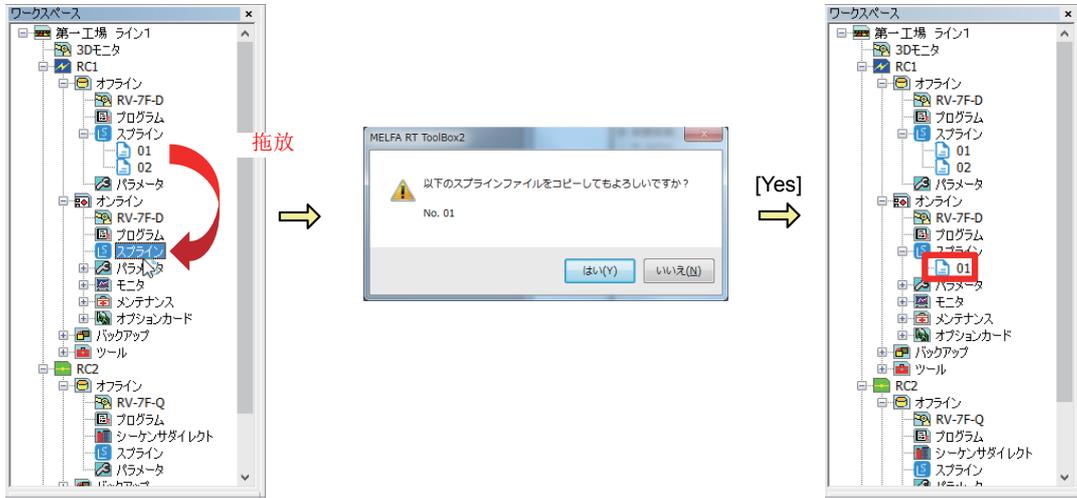


图 7-73: 样条文件的复制

按压 [ 是 ] 按钮，即复制所指定的样条文件。

按压 [ 否 ] 按钮，即取消复制操作。

要复制复制目标上已存在的样条号的样条文件时，将弹出覆盖确认对话框。

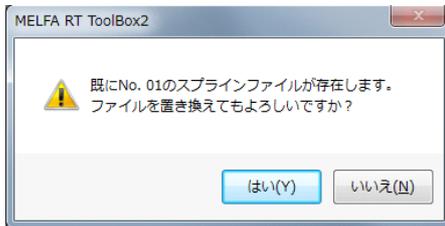


图 7-74: 覆盖确认对话框

## (13) 样条文件管理

RT ToolBox2 Ver.3.40S 以上版本中，备有样条文件的管理画面。

选择要进行样条管理的工程的 [在线] → [样条]，点击鼠标右键。显示上下文菜单后，点击 [样条文件管理]，即启动样条文件编辑画面。（菜单 [文件] → [样条文件管理]）

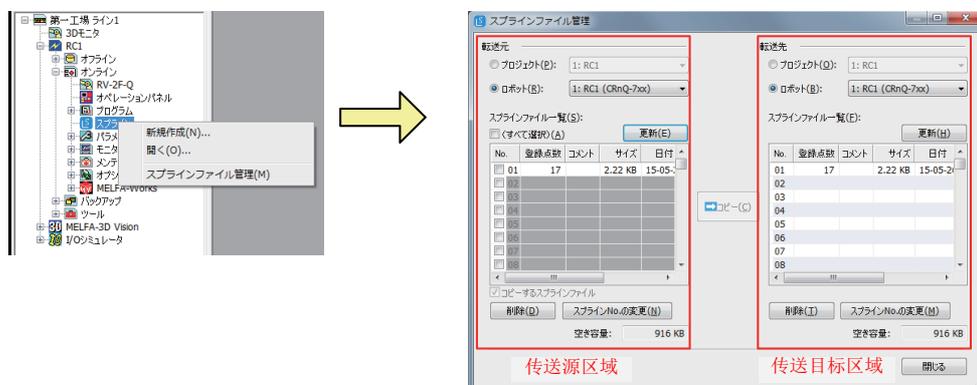


图 7-75: 样条文件管理画面

样条文件管理画面，由 2 两个区域构成，可以进行样条文件的复制、删除、样条 No. 的更改。

## ■ 样条文件的复制

勾选传送源中显示的样条文件的勾选框，点击 [复制] 按钮后，已勾选的样条文件将复制到传送目标中。传送目标中存在相同号码的样条文件时，可以更改并复制样条 No.。

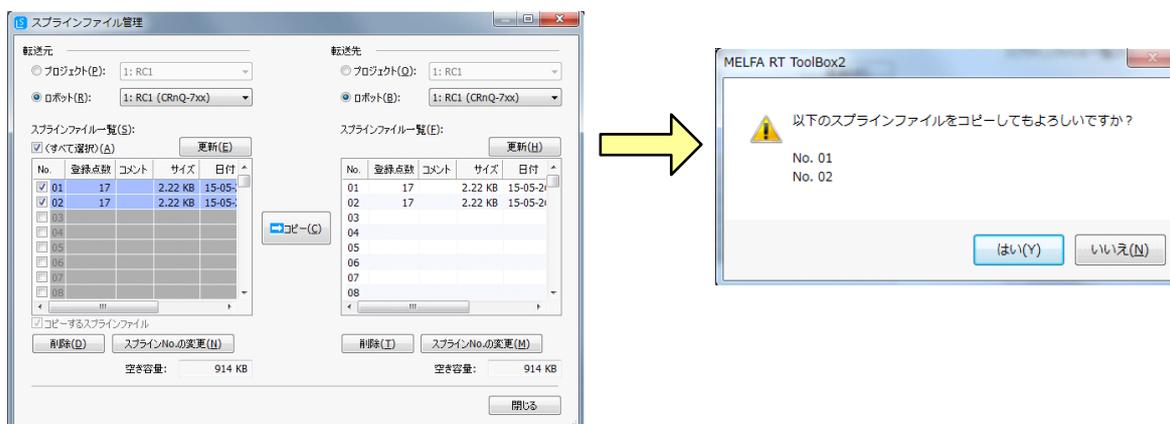


图 7-76: 样条文件管理画面上的文件复制

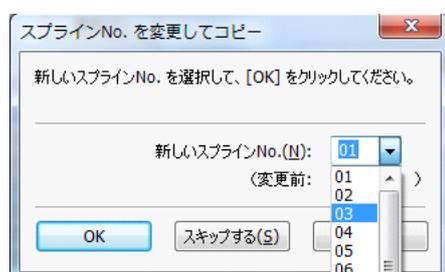


图 7-77: 更改并复制样条 No.

■ 样条文件的删除

从传送源或传送目标的样条文件一览表中选择要删除的文件，点击[删除]按钮，即可删除所选的样条文件。

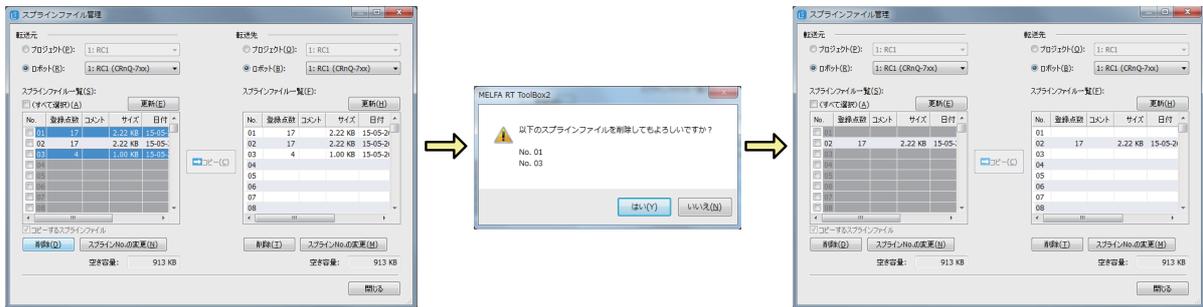


图 7-78：样条文件的删除

■ 样条 No. 的更改

从传送源或传送目标的样条文件一览表中选择要更改的文件，点击[样条 No. 的更改]按钮，即可更改所选的样条文件的 No.。

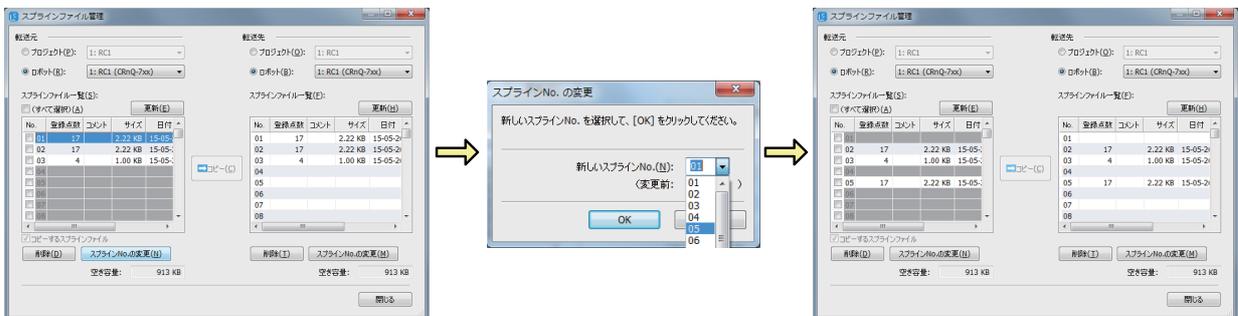


图 7-79：样条 No. 的更改

(14) 导入·导出功能

事先将路径点数据以规定形式的 CSV 文件记述，即可将该文件取得（导入）至样条文件编辑画面。通过此操作，将例如从 CAD 图纸中抽出的位置数据事先记述为 CSV 文件，即可批量登录为路径点数据。除了 CSV 文件外，也可导入从 MELFA-Works 中输出的点阵数据（MXT 文件）。

此外，也可从样条文件编辑画面将已登录的路径点数据作为 CSV 文件写出（导出）。通过此操作，即使不启动样条文件编辑画面，也可确认路径点数据的内容。

■ CSV 文件的形式

在 CSV 文件中，对表 7-37 所示的数据以逗号分隔按顺序记述为路径点数据。按照 1 行表示 1 个路径点数据进行记述。

表 7-37：记述在 CSV 文件中的数据

| 数据识别标记 | 内容                      |
|--------|-------------------------|
| <X>    | 指定路径点的 X 轴坐标值。<br>单位：mm |
| <Y>    | 指定路径点的 Y 轴坐标值。<br>单位：mm |
| <Z>    | 指定路径点的 Z 轴坐标值。<br>单位：mm |
| <A>    | 指定路径点的 A 轴坐标值。<br>单位：度  |
| <B>    | 指定路径点的 B 轴坐标值。<br>单位：度  |

| 数据识别标记     | 内容                                                                                                                                                                                                                                                                                                                                  |    |              |   |     |   |        |    |          |    |              |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|--------------|---|-----|---|--------|----|----------|----|--------------|
| <C>        | 指定路径点的 C 轴坐标值。<br>单位：度                                                                                                                                                                                                                                                                                                              |    |              |   |     |   |        |    |          |    |              |
| <L1>       | 指定路径点的 L1 轴（附加轴 1）坐标值。<br>单位：mm 或度                                                                                                                                                                                                                                                                                                  |    |              |   |     |   |        |    |          |    |              |
| <L2>       | 指定路径点的 L2 轴（附加轴 2）坐标值。<br>单位：mm 或度                                                                                                                                                                                                                                                                                                  |    |              |   |     |   |        |    |          |    |              |
| <FL1>      | 指定路径点的结构标志的值。<br>设定范围：0 ~ 7                                                                                                                                                                                                                                                                                                         |    |              |   |     |   |        |    |          |    |              |
| <FL2>      | 指定路径点的多旋转标志的值。<br>设定范围：0 ~ 4294967295                                                                                                                                                                                                                                                                                               |    |              |   |     |   |        |    |          |    |              |
| <Info>     | 指定圆弧指定的有无。<br>设定范围：有指定（1）、无指定（0）                                                                                                                                                                                                                                                                                                    |    |              |   |     |   |        |    |          |    |              |
| <Tol1>     | 以整数指定位置（轨迹）的允许公差量。<br>设定范围：0 ~ 100 [%]                                                                                                                                                                                                                                                                                              |    |              |   |     |   |        |    |          |    |              |
| <Tol2>     | 以整数指定姿势的允许公差量。<br>设定范围：0 ~ 100 [%]                                                                                                                                                                                                                                                                                                  |    |              |   |     |   |        |    |          |    |              |
| <Tol3>     | 未使用。请记述为 100。                                                                                                                                                                                                                                                                                                                       |    |              |   |     |   |        |    |          |    |              |
| <M_SplVar> | 以整数指定要通过状态变量 M_SplVar 浏览的值。<br>设定范围：-1 ~ 32767（-1 表示未设定。）                                                                                                                                                                                                                                                                           |    |              |   |     |   |        |    |          |    |              |
| <OutPort>  | 以整数指定输出信号的起始号码。<br>设定范围：-1 ~ 32767（-1 表示信号输出无效。）                                                                                                                                                                                                                                                                                    |    |              |   |     |   |        |    |          |    |              |
| <Bit>      | 指定要输出的信号的位宽。<br>设定范围：1、8、16、32 [位]                                                                                                                                                                                                                                                                                                  |    |              |   |     |   |        |    |          |    |              |
| <Mask>     | 以 16 进制数指定将信号输出看作有效的位的掩码模式。<br><table border="1" style="margin-left: 20px;"> <thead> <tr> <th>位宽</th> <th>设定范围 [16 进制]</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0、1</td> </tr> <tr> <td>8</td> <td>0 ~ FF</td> </tr> <tr> <td>16</td> <td>0 ~ FFFF</td> </tr> <tr> <td>32</td> <td>0 ~ FFFFFFFF</td> </tr> </tbody> </table> | 位宽 | 设定范围 [16 进制] | 1 | 0、1 | 8 | 0 ~ FF | 16 | 0 ~ FFFF | 32 | 0 ~ FFFFFFFF |
| 位宽         | 设定范围 [16 进制]                                                                                                                                                                                                                                                                                                                        |    |              |   |     |   |        |    |          |    |              |
| 1          | 0、1                                                                                                                                                                                                                                                                                                                                 |    |              |   |     |   |        |    |          |    |              |
| 8          | 0 ~ FF                                                                                                                                                                                                                                                                                                                              |    |              |   |     |   |        |    |          |    |              |
| 16         | 0 ~ FFFF                                                                                                                                                                                                                                                                                                                            |    |              |   |     |   |        |    |          |    |              |
| 32         | 0 ~ FFFFFFFF                                                                                                                                                                                                                                                                                                                        |    |              |   |     |   |        |    |          |    |              |
| <OutVal>   | 以 16 进制数指定要输出信号的数据。<br><table border="1" style="margin-left: 20px;"> <thead> <tr> <th>位宽</th> <th>设定范围 [16 进制]</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0、1</td> </tr> <tr> <td>8</td> <td>0 ~ FF</td> </tr> <tr> <td>16</td> <td>0 ~ FFFF</td> </tr> <tr> <td>32</td> <td>0 ~ FFFFFFFF</td> </tr> </tbody> </table>         | 位宽 | 设定范围 [16 进制] | 1 | 0、1 | 8 | 0 ~ FF | 16 | 0 ~ FFFF | 32 | 0 ~ FFFFFFFF |
| 位宽         | 设定范围 [16 进制]                                                                                                                                                                                                                                                                                                                        |    |              |   |     |   |        |    |          |    |              |
| 1          | 0、1                                                                                                                                                                                                                                                                                                                                 |    |              |   |     |   |        |    |          |    |              |
| 8          | 0 ~ FF                                                                                                                                                                                                                                                                                                                              |    |              |   |     |   |        |    |          |    |              |
| 16         | 0 ~ FFFF                                                                                                                                                                                                                                                                                                                            |    |              |   |     |   |        |    |          |    |              |
| 32         | 0 ~ FFFFFFFF                                                                                                                                                                                                                                                                                                                        |    |              |   |     |   |        |    |          |    |              |
| <Pulse>    | 指定脉冲输出。<br>设定范围：有指定（1）、无指定（0）                                                                                                                                                                                                                                                                                                       |    |              |   |     |   |        |    |          |    |              |

CSV 文件的示例如下所示。

| <X>    | <Y>     | <Z>    | <A>     | <B>    | <C>    | <L1> | <L2> | <FL1> | <FL2> | <Info> | <Tol1> | <Tol2> | <Tol3> | <M_SplVar> | <OutPort> | <Bit> | <Mask> | <OutVal> | <Pulse> |
|--------|---------|--------|---------|--------|--------|------|------|-------|-------|--------|--------|--------|--------|------------|-----------|-------|--------|----------|---------|
| 300.00 | 125.00  | 325.00 | 180.00  | 0.00   | 180.00 | 0.00 | 0.00 | 7     | 0     | 0      | 100    | 100    | 100    | 0          | 10100     | 8     | a      | ff       | 0       |
| 250.00 | 100.00  | 325.00 | 180.00  | 0.00   | 180.00 | 0.00 | 0.00 | 7     | 0     | 0      | 100    | 100    | 100    | 1          | -1        | 1     | 0      | 0        | 0       |
| 250.00 | 0.00    | 325.00 | 180.00  | 15.00  | 180.00 | 0.00 | 0.00 | 7     | 0     | 1      | 100    | 100    | 100    | 10         | 10100     | 8     | 5      | ff       | 1       |
| 300.00 | -50.00  | 325.00 | 165.00  | 0.00   | 180.00 | 0.00 | 0.00 | 7     | 0     | 1      | 100    | 100    | 100    | 100        | -1        | 1     | 0      | 0        | 0       |
| 350.00 | 0.00    | 325.00 | 180.00  | -15.00 | 180.00 | 0.00 | 0.00 | 7     | 0     | 1      | 100    | 100    | 100    | -1         | -1        | 1     | 0      | 0        | 0       |
| 300.00 | 50.00   | 325.00 | -165.00 | 0.00   | 180.00 | 0.00 | 0.00 | 7     | 0     | 1      | 100    | 100    | 100    | -1         | -1        | 1     | 0      | 0        | 0       |
| 250.00 | 0.00    | 325.00 | 180.00  | 15.00  | 180.00 | 0.00 | 0.00 | 7     | 0     | 1      | 100    | 100    | 100    | 0          | 10100     | 8     | ff     | 0        | 0       |
| 250.00 | -100.00 | 325.00 | 180.00  | 0.00   | 180.00 | 0.00 | 0.00 | 7     | 0     | 0      | 50     | 100    | 100    | -1         | -1        | 1     | 0      | 0        | 0       |
| 300.00 | -125.00 | 325.00 | 180.00  | 0.00   | 180.00 | 0.00 | 0.00 | 7     | 0     | 0      | 50     | 100    | 100    | -1         | -1        | 1     | 0      | 0        | 0       |

图 7-80: CSV 文件的示例

第 1 行中请务必从 <X> 开始按顺序记述表 7-37 的“数据识别标记”。未记述有“数据识别标记”时或各行的数据个数 (20 个) 过多或过少时, 将无法导入。

### ■ MXT 文件的形式

MXT 文件是指具有从 MELFA-Works 输出的点阵数据的文件。

按照以下步骤进行 MXT 文件的创建。

※ 关于 MELFA-Works 的使用方法, 请参照“MELFA-Works 使用说明书 (BFP-A8477)”。

※ 根据样条与 Ex-T 样条的不同, 有些步骤内容也不同, 请加以注意。

#### 1) MELFA-Works 的启动

启动 MELFA-Works。

#### 2) 工件的夹持

※ 该作业仅在 Ex-T 样条时需要。样条时, 请实施“4) 路径创建”的作业。

将夹持用的抓手安装在机器人上。

点击 [ 机器人设定 ] → [ 更改 ], 将显示机器人详细设定画面。

点击机器人详细设定画面的 [ 信号连接 ] 按钮, 将显示 Hand I/O 画面并夹持工件。

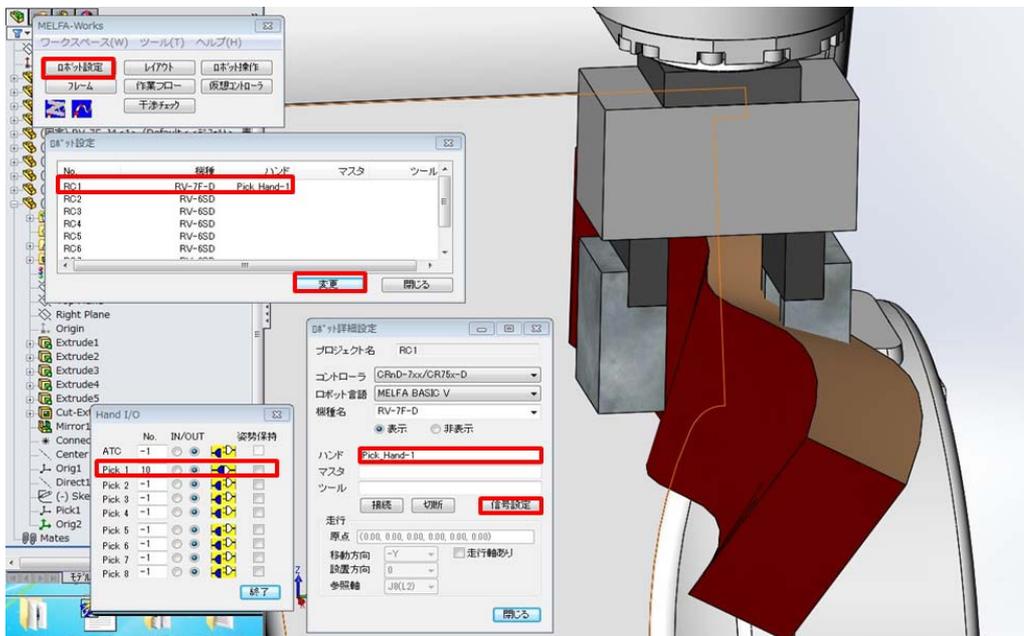


图 7-81: 工件夹持

## 3) 夹持位置的示教

※ 该作业仅在 Ex-T 样条时需要。样条时，请实施“4) 路径创建”的作业。

点击 [作业流程]，将显示作业流程画面。点击作业流程画面的 [导入] 按钮，示教夹持工件的位置。

※ 此处所示教的数据为夹持位置。

※ 示教夹持位置后，请勿移动工件。

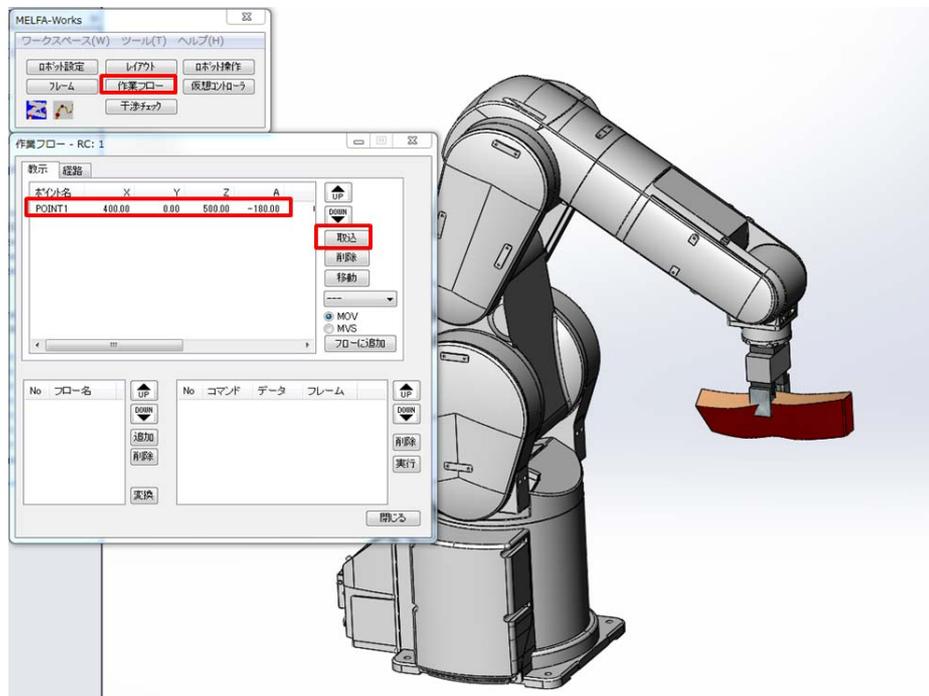


图 7-82: 夹持位置的示教

4) 路径创建

更换用于创建路径数据的 CAD 链接用抓手。

点击 [ 作业流程 ]，将显示作业流程画面。

选择作业流程画面的路径标签，点击 [ 增加 ] 按钮，即会增加加工路径。

选择已增加的加工路径，点击 [ 编辑 ] 按钮，即显示加工设定画面。

选择工件上的作业路径，点击 [ 增加 ] 按钮，即可登录加工路径。

※Ex-T 样条时，即使是垂直多关节（6 轴型）机器人也请勿勾选 Z 反转。

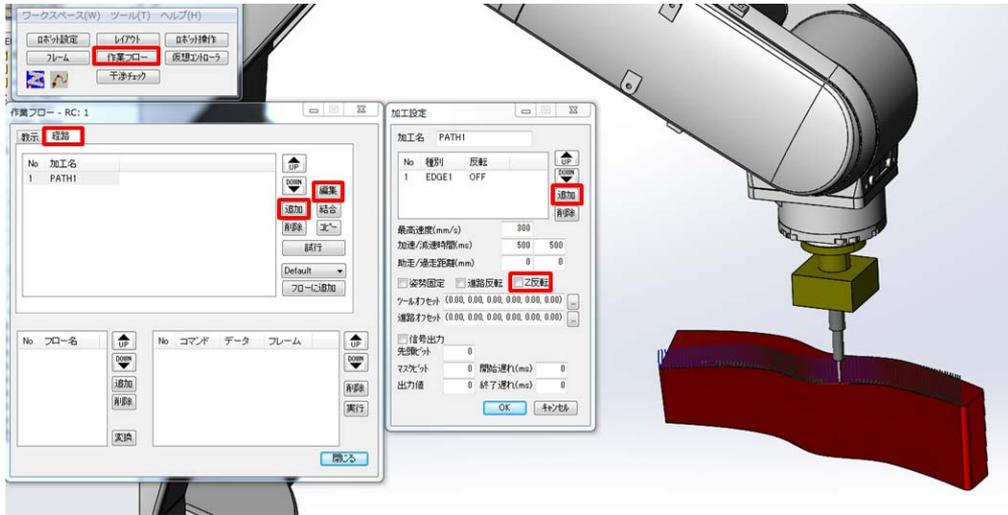


图 7-83: 路径创建

5) 作业流程的创建

进行作业流程的创建。

※ 根据样条与 Ex-T 样条的不同，作业流程的步骤也不同，请加以注意。

点击 [ 作业流程 ]，将显示作业流程画面。

点击流程的 [ 增加 ] 按钮，增加流程。

选择作业流程画面的路径标签。

选择加工路径，并点击 [ 增加至流程 ] 按钮，所选择的数据即增加至流程。

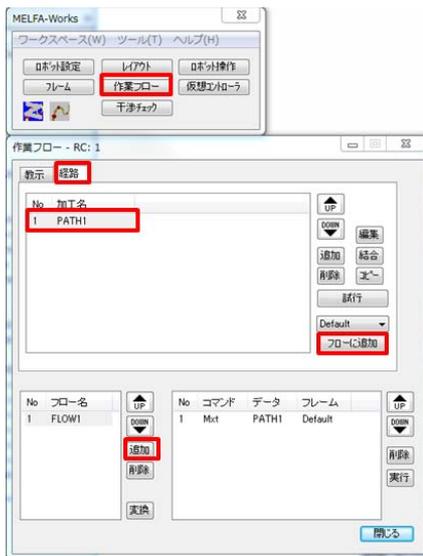


图 7-84: 流程创建（加工路径登录）

Ex-T 样条时，需要在作业流程中登录“3) 夹持位置的示教”中所示教的夹持位置。  
在作业流程画面中选择示教标签。  
选择示教的夹持位置，并点击[增加至流程]按钮，所选择的位置数据即增加至流程。



图 7-85: 流程创建（夹持位置登录）

#### 6) MXT 文件的输出

输出所创建的作业流程相对应的 MXT 文件。

点击作业流程画面的[转换]按钮，即显示输出目标设定对话框。

在程序名中输入任意的名称，并点击[OK]按钮，即可在通过 MELFA-Works 指定的工程文件夹的下述位置创建所输入的程序名的程序与 MXT 文件。

#### · 文件夹构成

MXT 文件: MELFA-Works\MXT01\_01.MXT

程序: FLOW.PRGM (※ 程序名设定为“FLOW”时)

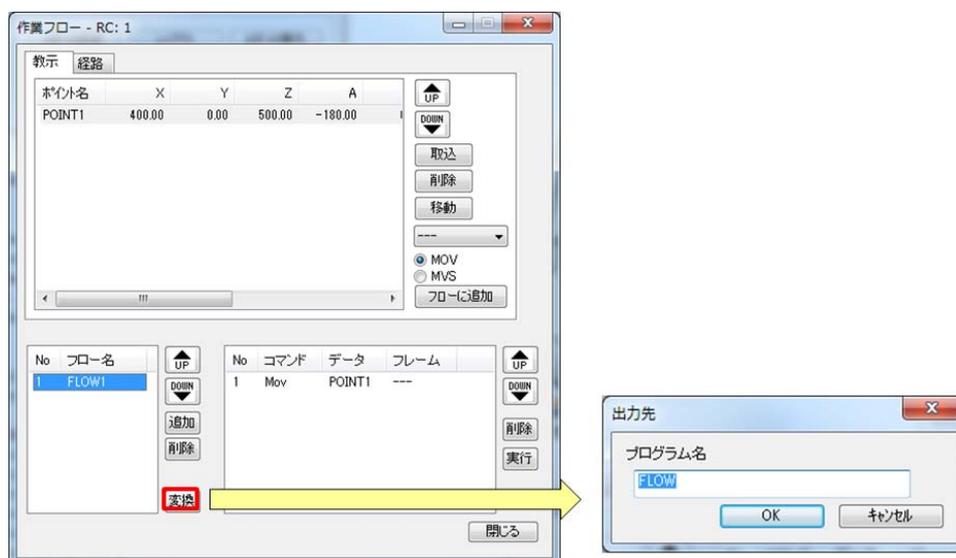


图 7-86: MXT 文件的输出

## ■ 导入

RT ToolBox2 Ver.3.20W 以上的版本不仅支持 CSV 格式，还支持 MXT 格式的文件。

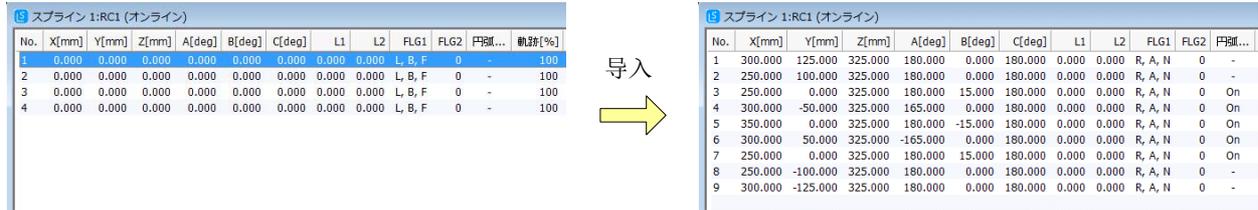


图 7-87: 导入



## 注意

通过导入，在样条文件编辑画面中登录的全部路径点数据，都会替换为新导入的路径点数据。

### • CSV 文件

点击菜单 [文件] → [导入]，即显示文件选择对话框。选择并打开希望打开的 CSV 文件，激活的样条文件编辑画面中即导入 CSV 文件的内容。

※Ex-T 样条时，数据导入后必须进行 Ex-T 控制设定。

### • MXT 文件

点击菜单 [文件] → [导入]，即显示文件选择对话框。选择并打开希望打开的 MXT 文件，即显示允许公差设定画面。

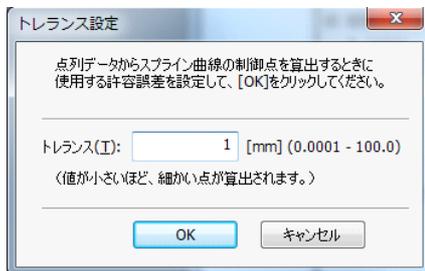


图 7-88: 允许公差设定画面

设定点阵数据与样条曲线之间的容许误差 (mm) 并点击 [OK] 按钮，即显示 Ex-T 控制设定画面。

样条时无需设定，因此点击 [OK] 按钮，MXT 文件的内容即导入至激活的样条文件编辑画面。(允许误差越小，路径点算出越细。)

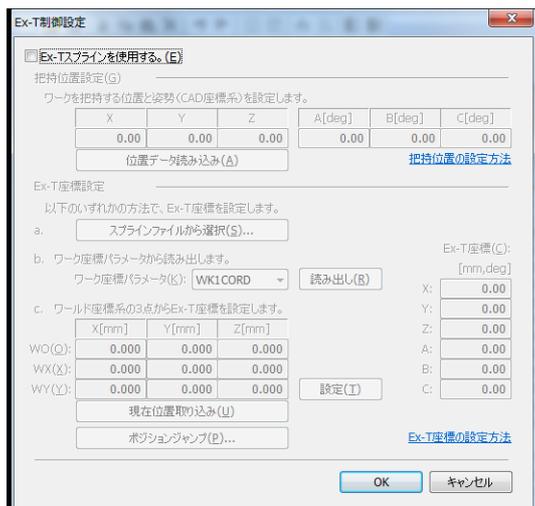


图 7-89: Ex-T 控制设定画面（初始画面）

Ex-T 样条时，需要设定夹持位置与 Ex-T 坐标。

勾选 [ 使用 Ex-T 样条 ] 勾选框。

点击 [ 位置数据读取 ] 按钮，即显示程序选择画面，从中选择 MXT 文件输出时创建的程序。

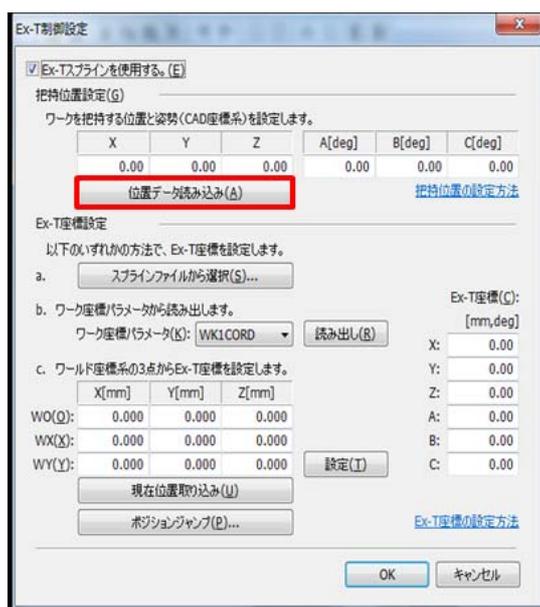


图 7-90: Ex-T 控制设定画面（设定方法）

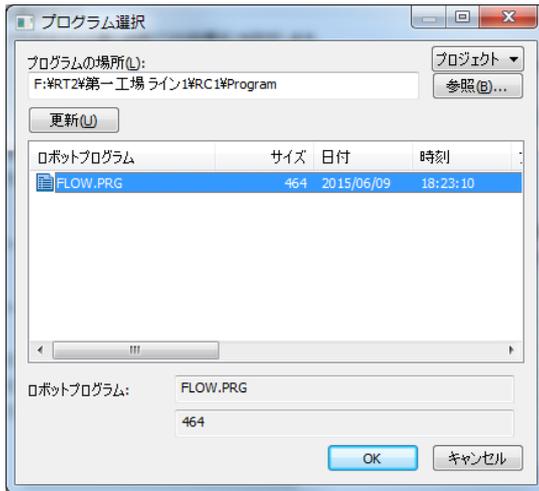


图 7-91: 程序选择画面

选择程序并点击 [OK] 按钮，显示直角位置数据选择画面。

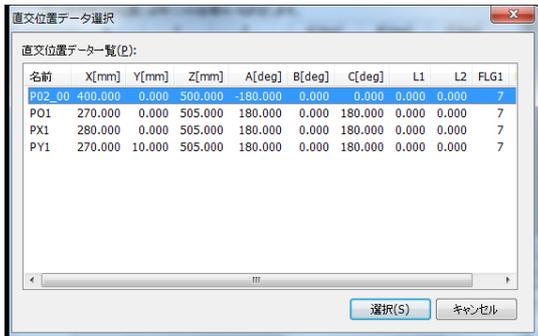


图 7-92: 直角位置数据选择画面

选择要读取的位置数据（MXT 文件创建时示教的夹持位置）并点击 [OK] 按钮，夹持位置即被设定。

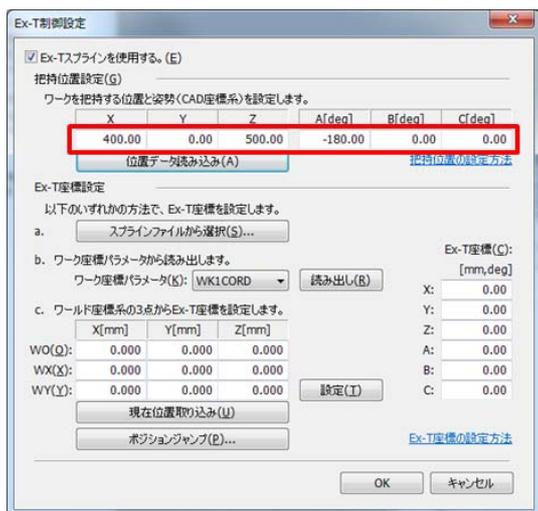


图 7-93: Ex-T 控制设定（夹持位置设定）

设定 Ex-T 坐标并点击 [OK] 按钮，MXT 文件的内容即被转换为用于 Ex-T 样条的路径点数据并被导入。

## ⚠ 注意

对应的点阵的数据类型仅有直交数据。  
因为不导入 MXT 文件的页眉信息的移动速度（加减速时间·最高速度），所以需要反映至机器人程序的 MvSpl 指令的自变量。

## ⚠ 注意

如减小允许公差的值，则所算出的路径点的精度会上升，但路径点间的距离会变短，因此无法加快速度。

### ■ 导出

点击菜单 [文件] → [导出]，即显示指定 CSV 文件的保存目标的对话框。若指定保存目标的文件夹与 CSV 文件的名称进行保存，则激活的样条文件编辑画面的路径点数据的内容会被写出至 CSV 文件。

## (15) 编辑的辅助功能

### ■ 剪切 & 粘贴

剪切所指定的路径点数据，粘贴至其他的路径点数据。

在路径点数据一览区域中选择剪切位置的路径点数据，点击菜单 [编辑] → [剪切]。然后，选择粘贴目标的路径点数据，并点击菜单 [编辑] → [粘贴]，通过 [剪切] 操作所存储的路径点数据即按顺序被粘贴至粘贴目标的路径点数据。与此同时，剪切位置的路径点数据将被从一览中删除。

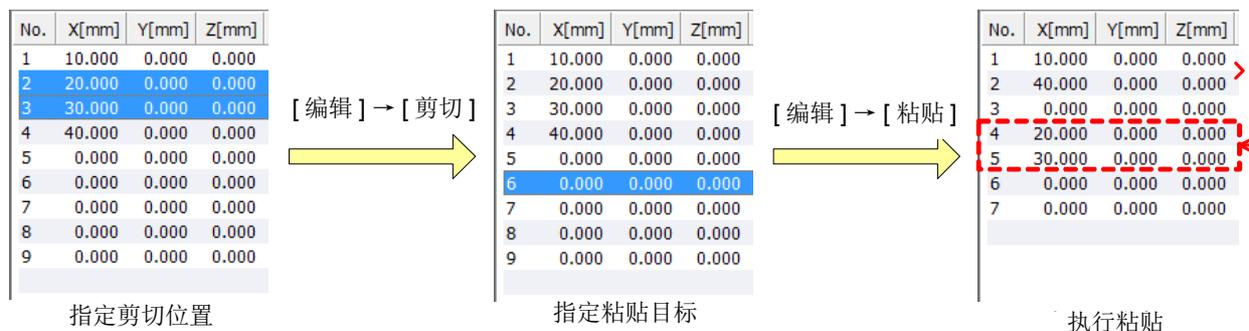


图 7-94: 剪切 & 粘贴

### ◇ ◆ ◆ 粘贴目标的路径点数据数较少时 ◆ ◆ ◆

对于指定为剪切位置的路径点数据的个数而言，粘贴目标以后的路径点数据数较少时，将不会粘贴没有粘贴目标的路径点数据。

### ■ 复制 & 粘贴

将所指定的路径点数据的内容复制到其他的路径点数据。

在路径点数据一览区域中选择作为复制位置的路径点数据，点击菜单 [编辑] → [复制]。然后，选择复制目标的路径点数据，并点击菜单 [编辑] → [粘贴]，通过 [复制] 操作所存储的路径点数据即按顺序被复制至复制目标的路径点数据。



图 7-95: 复制 & 粘贴

### ◇◆◇复制目标的路径点数据数较少时◇◆◇

对于指定为复制位置的路径点数据的个数而言，复制目标以后的路径点数据数较少时，将不会复制没有复制目标的路径点数据。

#### ■复制&粘贴（位置数据）

将所指定的位置数据的内容复制到样条文件的路径点数据。

在机器人程序中选择作为复制位置的位置数据，点击菜单[编辑]→[复制-位置数据]。然后，选择作为复制目标的路径点数据，并点击菜单[编辑]→[粘贴-位置数据]，通过[复制-位置数据]操作所存储的位置数据即按顺序被复制至复制目标的路径点数据。

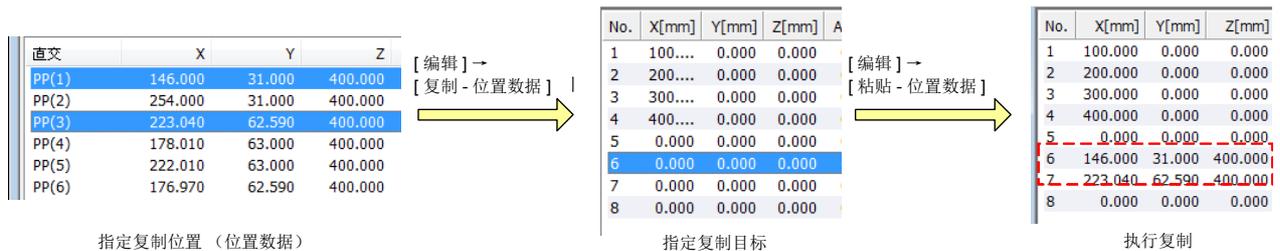


图 7-96: 复制&粘贴（位置数据）

#### ■所复制的路径点的插入

将所指定的路径点数据的内容作为新路径点数据插入其他的行中。

在路径点数据一览区域中选择作为复制位置的路径点数据，点击菜单[编辑]→[复制]。然后，选择作为插入目标的路径点数据，并点击菜单[编辑]→[所复制的路径点的插入]，在作为插入目标所选择的路径点数据之前将[复制]操作所存储的路径点数据按顺序插入。

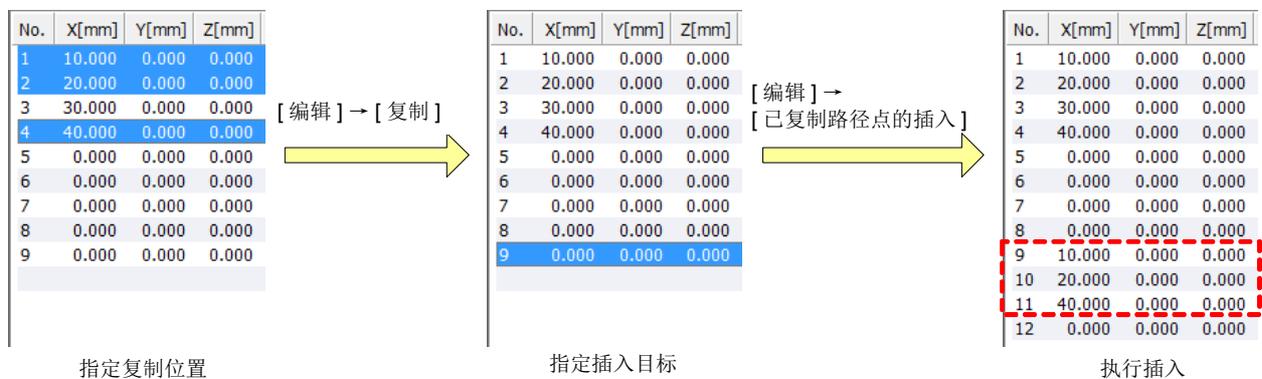


图 7-97: 所复制的路径点的插入

#### ■撤销（undo）

点击菜单[编辑]→[撤销]，即取消对路径点数据最后适用的编辑内容，恢复适用前的状态。具体来说，以下的操作为适用对象。

- 路径点数据的增加·插入·删除
- 路径点数据的更改（适用编辑内容）
- 路径点数据的剪切&粘贴·复制&粘贴·复制的路径点的插入
- 导入
- 位置补偿功能
- 帧转换（执行转换）

#### ■恢复（redo）

点击菜单[编辑]→[恢复]，通过[撤销]取消的编辑内容将重新适用于路径点数据。

## (16) 样条曲线的显示

显示样条曲线。该功能可在 RT ToolBox2 Ver.3.20W 以上的版本中使用。

点击菜单 [ 工具 ] → [ 样条曲线 ], 3D 监视画面打开后编辑中的样条文件的样条曲线会显示在 3D 监视画面中。

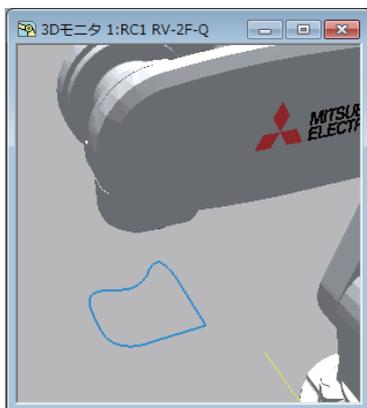


图 7-98: 样条曲线的显示

关于 3D 监视画面的使用方法, 请参照“RT ToolBox2/RT ToolBox2 mini 使用说明书”。

## (17) 编辑内容的确认

## ■ 路径点检查

事先对各路径点检查路径点间的距离·姿势的变化量·结构标志的值等。

该功能可在 RT ToolBox2 Ver.3.20W 以上的版本中使用。

点击菜单 [ 工具 ] → [ 路径点检查 ], 即显示激活的样条编辑画面的路径点检查画面。

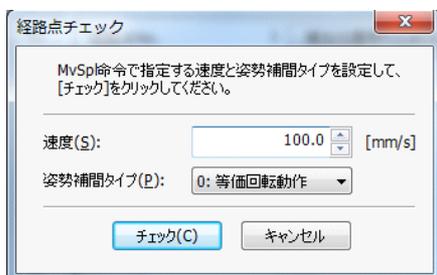


图 7-99: 路径点检查画面

设定通过 MvSpl 指令 /EMvSpl 指令指定的速度与姿势插补类型并按压 [ 检查 ] 按钮, 即执行各路径点的检查。正常结束后显示完成对话框。发生错误时, 将显示错误对话框。关于错误内容, 请参照另一手册“故障排除”。

※ 路径点检查时, 将以倍率修调 100% 动作时的速度进行检查。

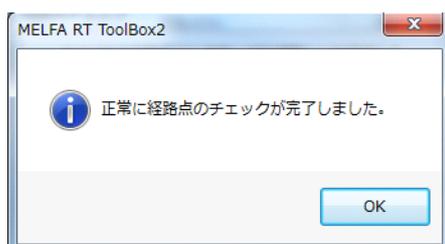


图 7-100: 路径点检查完成对话框 (左) / 错误对话框 (右)

**■ 动作范围检查**

事先对各路径点检查机器人可否动作。

该功能仅在 RT ToolBox2 Ver.3.40S 以上版本的模拟功能中使用。

点击菜单 [ 工具 ] → [ 动作范围检查 ]，即向激活的样条编辑画面的各路径点执行动作。正常结束后显示完成对话框。发生错误时，将显示错误对话框。关于错误内容，请参照另一手册“故障排除”。

※ 动作范围检查中，机器人的轨迹无法得到保证。

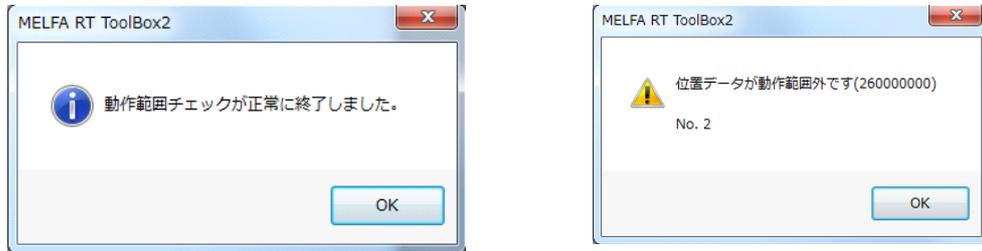


图 7-101: 动作范围检查完成对话框 (左) / 错误对话框 (右)

**■ 最大速度算出**

事先从路径点间的距离算出执行样条插补指令 (MvSpl 指令、EM v Spl 指令) 时不会发生错误的可执行的最大速度。

该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。

点击菜单 [ 工具 ] → [ 最大速度算出 ]，即显示激活的样条编辑画面的最大速度算出结果。

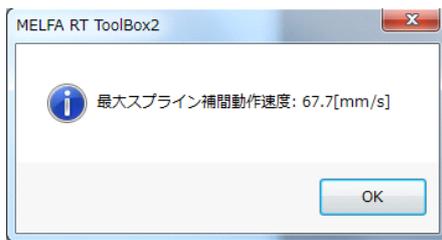


图 7-102: 最大速度算出结果

**(18) 机器人程序****■ 程序转换**

创建使用登录在样条文件中的路径点数据的机器人程序。

该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。

点击菜单 [ 工具 ] → [ 机器人程序 ] → [ 程序转换 ]，即显示激活的样条编辑画面的新机器人程序对话框。

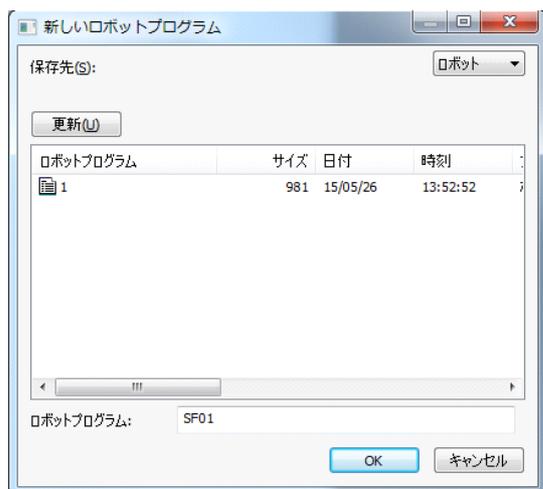


图 7-103: 新机器人程序画面

输入要保存的机器人程序名并点击 [OK] 按钮，即显示速度设定对话框。



图 7-104: 速度设定对话框

设定控制点的速度并点击 [OK] 按钮，样条文件的内容即转换为机器人程序。

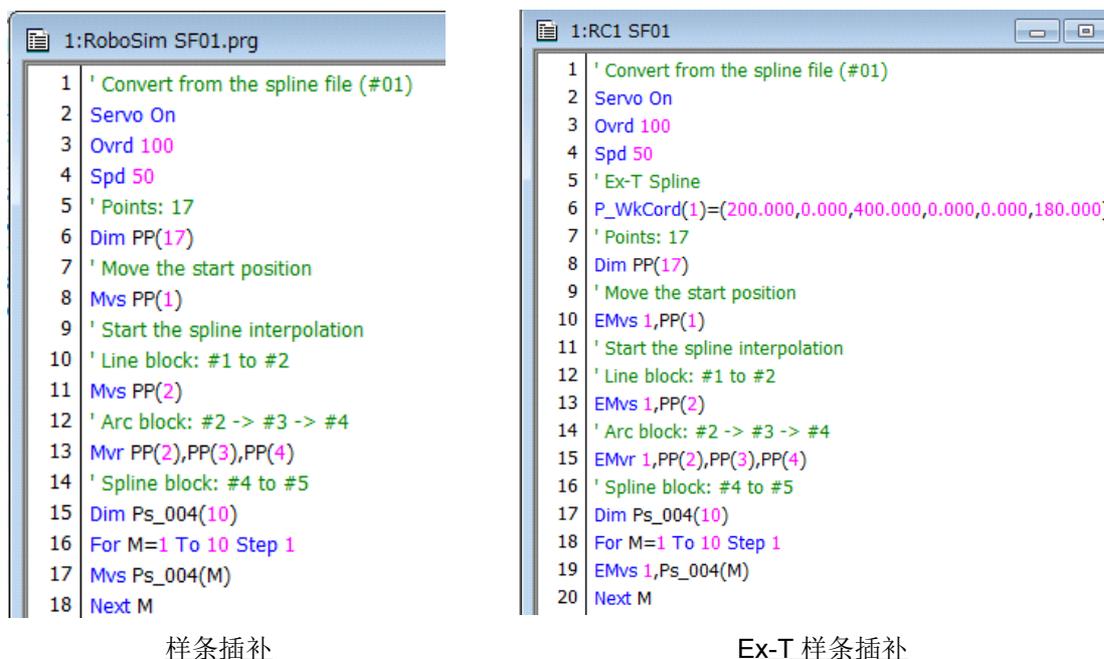


图 7-105: 通过各插补创建的程序

### ■ 位置数据导入

可将机器人程序的位置数据作为样条文件的路径点数据进行导入。

该功能可在 RT ToolBox2 Ver.3.40S 以上的版本中使用。

点击菜单 [ 工具 ] → [ 机器人程序 ] → [ 位置数据导入 ]，即显示激活的样条编辑画面中要导入的程序的选  
择画面。

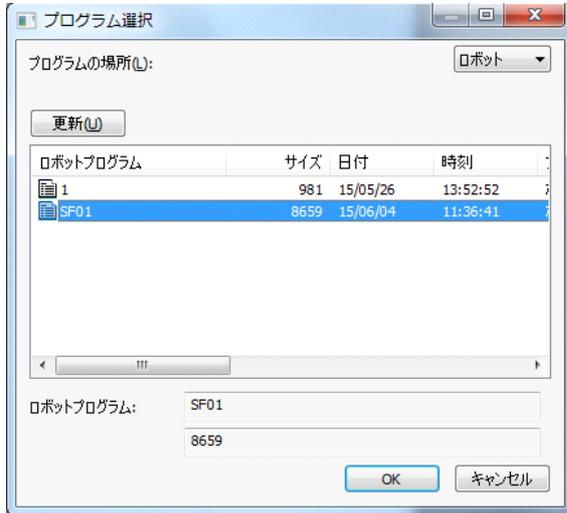


图 7-106: 程序选择画面

选择要导入位置数据的机器人程序并点击 [OK] 按钮，即在激活的样条编辑画面中导入位置数据。

※ 可导入的位置数据为 PP(n) (n: 路径点号) 或 P\* (\*: 路径点号)。

※ 激活的样条文件对应的路径点不存在时，新建路径点。(位置数据以外为初始值。)

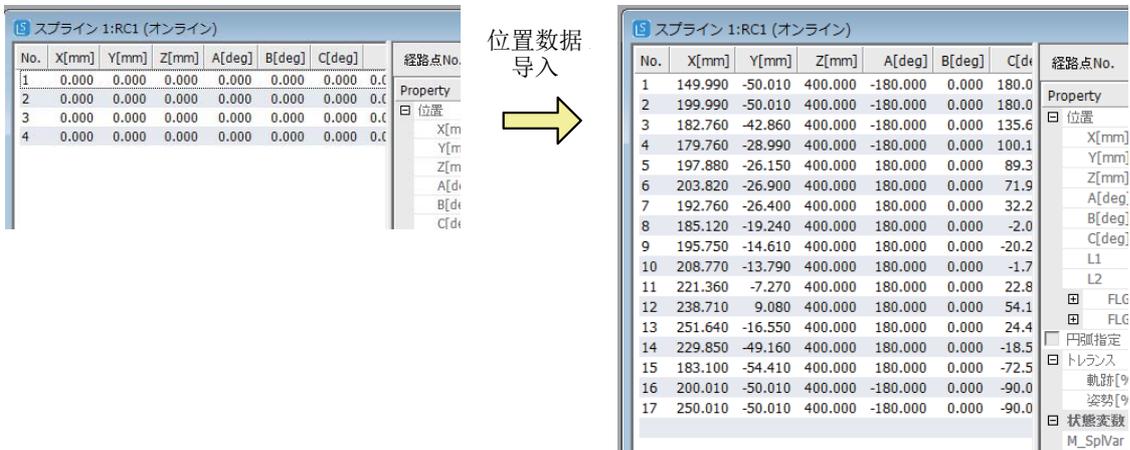


图 7-107: 位置数据导入

### 7.4.6 机器人程序的创建

在 RT ToolBox2 示教单元的程序编辑画面中创建机器人程序。

执行样条插补时，使用记载在表 7-38 与表 7-40 中的指令·机器人状态变量。关于详细内容，请参照第 163 页的「4.13 指令的详细说明」·第 287 页的「4.14 机器人（系统）状态变量的详细说明」。

表 7-38: 样条插补中使用的指令

| 指令语                               | 说明            | 参照页码 |
|-----------------------------------|---------------|------|
| MvSpl (Move Spline)               | 执行样条插补。       | 244  |
| EMvSpl (E Move Spline)            | 执行 Ex-T 样条插补。 | 211  |
| SetCalFrm (Set Calibration Frame) | 设定帧转换中使用的坐标系。 | 266  |

表 7-39: 样条插补中使用的编入函数

| 函数名称     | 说明                                                                         | 参照页码 |
|----------|----------------------------------------------------------------------------|------|
| SplPos   | 将登录在指定样条文件中的路径点数据代入至位置变量。                                                  | 400  |
| SplSpd   | 基于指定的样条文件的路径点数据，将在执行 MvSpl 指令 /EMvSpl 指令时不会发生 L2611( 路径点过近 ) 错误的速度代入至数值变量。 | 401  |
| SplECord | 将登录在指定样条文件中的 Ex-T 坐标系原点数据代入至位置变量。                                          | 399  |

表 7-40: 样条插补相关的机器人状态变量

| 变量名称     | 数组元素数       | 内容                                | 属性注 1) | 数据的类型 | 参照页码 |
|----------|-------------|-----------------------------------|--------|-------|------|
| M_SplPno | 机械号 (1 ~ 3) | 返回最新通过的路径点的号码。                    | R      | 整数型   | 334  |
| M_SplVar | 机械号 (1 ~ 3) | 返回最新的路径点数据的数值设定值。<br>通过写入，也可以更改值。 | RW     | 整数型   | 335  |

注 1) R ..... 可读取。

RW..... 可读取、写入。

■ 采样程序

路径 1 的路径点数据登录在样条文件 05 中。使用帧转换执行路径 1 与路径 2 的样条插补。此时，在插槽 2 的程序中使用路径点数据的数值设定，在动作过程中将输出信号为 100 号和 101 号的信号置为 ON/OFF。

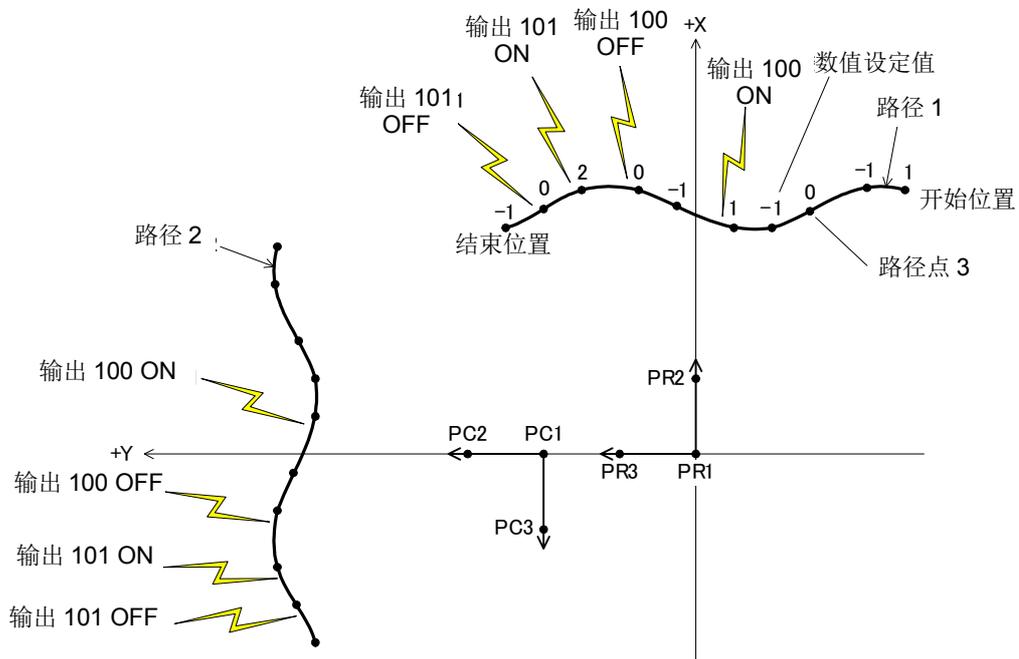


图 7-108: 采样程序的动作

插槽 1

|                                        |                                        |
|----------------------------------------|----------------------------------------|
| Ovrd 100                               |                                        |
| Mov P1                                 | ‘以关节插补向待机位置 (P1) 移动                    |
| ‘                                      |                                        |
| Fine 200                               | ‘将定位脉冲更改为 200                          |
| Spd 100                                | ‘将到开始位置为止的直线插补速度指定为 100mm/s            |
| M_00=1                                 | ‘将插槽 2 同步用的标志置为 ON                     |
| MvSpl 5, 50, 10                        | ‘以加减速距离 10mm、速度 50mm/s 进行路径 1 的样条插补    |
| M_00=0                                 | ‘将同步用的标志置为 OFF                         |
| Fine 0                                 | ‘将定位脉冲指定设为无效                           |
| Mvs P1                                 | ‘通过直线插补向待机位置移动                         |
| ‘                                      |                                        |
| PR1=(0, 0, 0, 0, 0, 0)(0, 0)           | ‘设定用于帧转换的基准坐标系的位置 (PR1 ~ PR3)          |
| PR2=(20, 0, 0, 0, 0, 0)(0, 0)          |                                        |
| PR3=(0, 20, 0, 0, 0, 0)(0, 0)          |                                        |
| PC1=(0, 40, 0, 0, 0, 0)(0, 0)          | ‘设定转换后的基准坐标系的位置 (PC1 ~ PC3)            |
| PC2=(0, 60, 0, 0, 0, 0)(0, 0)          | ‘(相对于转换前, 绕 Z 轴旋转 90 度·在 Y 轴方向上移动 40mm |
| PC3=(-20, 40, 0, 0, 0, 0)(0, 0)        | ‘的坐标)                                  |
| SetCalFrm PR1, PR2, PR3, PC1, PC2, PC3 | ‘算出帧转换用的坐标系并进行设定                       |
| ‘                                      |                                        |
| Fine 200                               |                                        |
| M_00=1                                 |                                        |
| MvSpl 5, 50, 10, 2                     | ‘对样条号 5 的路径点实施帧转换,                     |
|                                        | ‘进行路径 2 的样条插补                          |
| M_00=0                                 |                                        |
| Fine 0                                 |                                        |
| Mvs P1                                 |                                        |
| End                                    |                                        |

## 插槽 2

|                                 |                               |
|---------------------------------|-------------------------------|
| Def IO PORT1=Byte, 100, &H03    | ‘ 将输出信号 100 和 101 分配给变量 PORT1 |
| M_SplVar=0                      | ‘ 将 M_SplVar 的值复位为 0          |
| Wait M_00=1                     | ‘ 等待直至样条插补开始                  |
| *L1:If M_SplPno<3 Then GoTo *L1 | ‘ 等待直至通过路径点 3                 |
| *L2                             |                               |
| Select M_SplVar                 |                               |
| Case 1                          | ‘ 通过将数值设定值设定为 1 的路径点          |
| PORT1=1                         | ‘ 将输出信号 100 置为 ON             |
| Break                           |                               |
| Case 2                          | ‘ 通过将数值设定值设定为 2 的路径点          |
| PORT1=2                         | ‘ 将输出信号 101 置为 ON             |
| Break                           |                               |
| Default                         | ‘ 数值设定值不为 1、2                 |
| PORT1=0                         | ‘ 将输出信号 100 和 101 置为 OFF      |
| Break                           |                               |
| End Select                      |                               |
| If M_00=1 Then Goto *L2         | ‘ 反复进行，直至样条插补结束               |
| End                             |                               |

### 7.4.7 动作的确认

使用 RT ToolBox2 的模拟功能来确认所创建的机器人程序的样条插补的动作。与希望的动作不相同，要修改路径点数据的设定·机器人程序。

关于模拟功能的使用方法，请参照“RT ToolBox2/RT ToolBox2 mini 使用说明书”。

此外，在 RT ToolBox2 mini 中无法使用模拟功能。

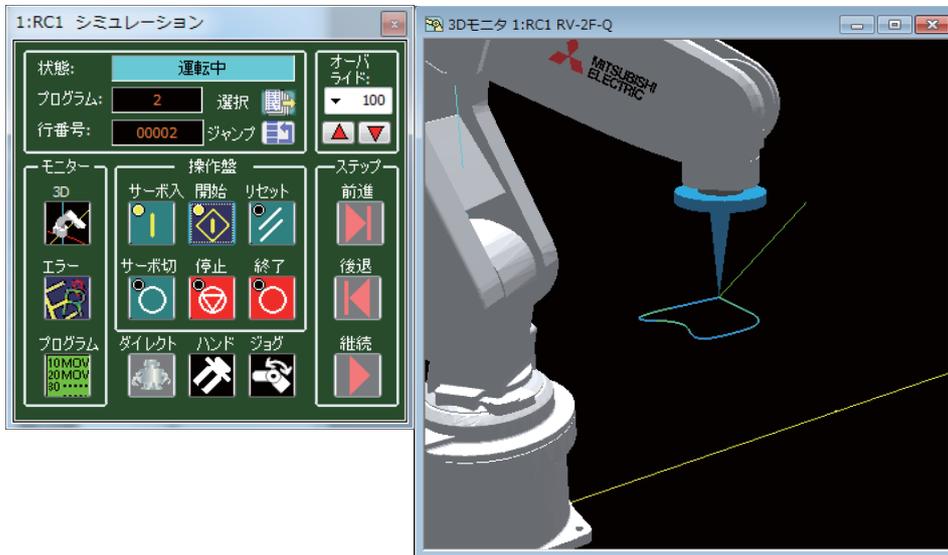


图 7-109: 模拟功能

#### ◇◆◇模拟所使用的机器人程序与样条文件◇◆◇

以模拟执行样条插补时，机器人程序与样条文件必须登录在 RT ToolBox2 内的虚拟控制器中。启动了模拟的工程的 [在线] 中，未登录机器人程序与样条文件时，应通过以下的方法复制到虚拟控制器中。

- 机器人程序：使用 RT ToolBox2 的程序管理。
- 样条文件：在样条文件编辑画面中打开对象文件，进行第 622 页的「(9) 样条文件的保存」。此外，进行第 625 页的「(12) 样条文件的复制」。

### 7.4.8 保存至机器人控制器

将机器人程序与样条文件保存至机器人控制器。

关于样条文件的保存方法，请参照第 622 页的「(9) 样条文件的保存」。

### 7.4.9 调整作业

实际的系统中，通过调试操作（单步进给）来确认样条插补的动作。

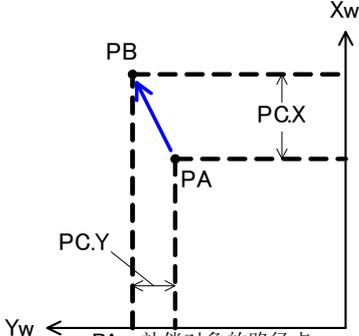
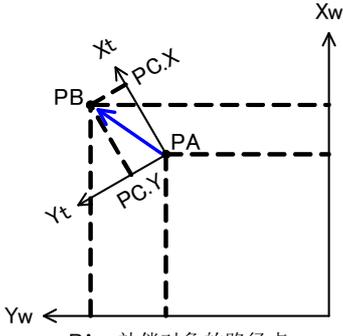
与希望的动作不相同，要修改路径点数据的设定·机器人程序。修改路径点数据时，需要将样条文件读取至 RT ToolBox2 的样条文件编辑画面，更改路径点数据的设定值后，写入至控制器。

下面对为了补偿位置数据而在 RT ToolBox2 中配备的“位置补偿功能”·“帧转换功能”·“跳位功能”和参数 SPLOPTGC（有源增益控制的增益补偿率）进行说明。

#### (1) 位置补偿功能

对路径点数据的机器人的位置，可以实施与 MELFA-BASIC V 的位置数据的相对运算同样的补偿。如表 7-41 所示，补偿方法有 2 种。

表 7-41: 位置的补偿方法

| 补偿方法           | 说明                                                                                                                                                                                                                                                                                                    |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 矢量和运算 (P+P)    | <p>对路径点数据的机器人的位置数据, 进行补偿数据的值的加法运算 (各坐标成分的加法运算)。是基于世界坐标系进行的补偿。<br/>结构标志·多旋转标志·附加轴数据不会更改, 保留原来的值。</p>  <p>PA: 补偿对象的路径点<br/>PB: 补偿结果<br/>PC: 补偿数据</p> <p>矢量和运算 (P+P)</p>                                                |
| 矢量乘积运算 (P x P) | <p>对路径点数据的机器人的位置数据, 进行补偿数据的值的乘法运算。<br/>(补偿后的位置 = 路径点的位置 × 补偿数据)<br/>是基于工具坐标系进行的补偿。<br/>结构标志·多旋转标志·附加轴数据不会更改, 保留原来的值。</p>  <p>PA: 补偿对象的路径点<br/>PB: 补偿结果<br/>PC: 补偿数据<br/>Xt-Yt: PA 中工具坐标的方向</p> <p>矢量乘积运算 (P×P)</p> |

使用位置补偿功能时, 点击菜单 [工具] → [位置补偿]。点击 [位置补偿] 后, 即显示位置补偿画面。

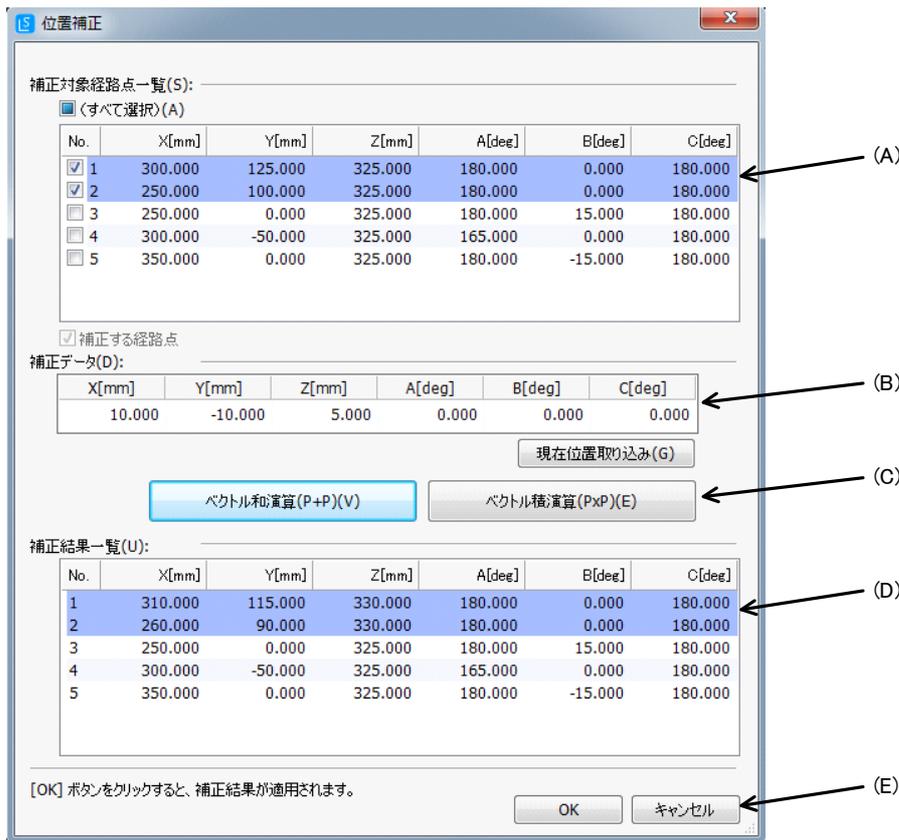


图 7-110: 位置补偿画面

位置补偿按照以下的步骤进行。

- (A) “补偿对象路径点一览”中将显示激活的样条文件编辑画面的路径点数据。从中勾选要实施位置补偿的路径点数据。可以选择多个路径点数据。
- (B) 设定补偿数据。  
与控制器连接时，将显示“当前位置导入”的按钮，按压该按钮，机器人的当前位置即被导入为补偿数据。
- (C) 选择补偿方法，按压按钮。
- (D) “补偿结果一览”中将显示补偿实施后的路径点数据。未选择为补偿对象的路径点数据与原来的值相同。
- (E) 按压 [OK] 按钮，即在样条文件编辑画面的路径点数据中反映补偿的结果，并关闭位置补偿画面。  
按压 [取消] 按钮，则删除补偿的结果并关闭位置补偿画面。

## (2) 帧转换功能

点击菜单 [工具] → [帧转换]，即显示用于对激活的样条文件编辑画面的路径点数据实施帧转换的画面。  
(关于帧转换，请参照第 598 页的「(5) 帧转换」。)



图 7-111: 帧转换画面

### ■ 坐标系の設定

图 7-111 (A) 的部分中, 设定对基准坐标系与转换后的基准坐标系进行定义的各 3 个位置 (原点·X 轴上的位置·X-Y 平面的 +Y 方向的位置) 的 X、Y、Z 轴坐标值。

关于基准坐标系, 按压 [从路径点一览中选择] 按钮, 可以从所登录的路径点数据中选择要设定的坐标值。

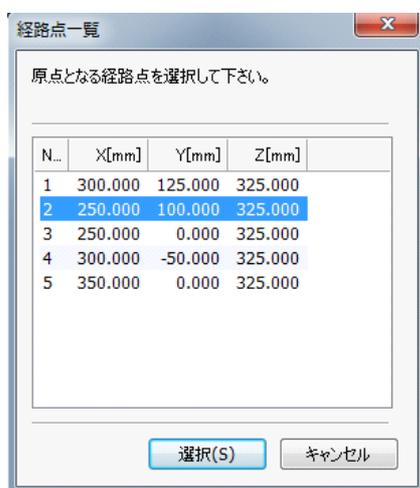


图 7-112: 从路径点一览中选择

与控制器连接时, 将显示 [当前位置导入] 按钮。按压该按钮, 即在所选择的转换后的基准坐标系的位置上设定机器人的当前位置。

### ■ 通过 RT ToolBox2 执行帧转换

设定坐标系后, 按图 7-111 (B) 的 [执行转换] 按钮, 即对全部的路径点数据实施帧转换, 并在图 7-111 (C) 的“转换结果一览”画面中显示该结果。

实施帧转换后, 图 7-111 (D) 的 [适用转换结果后结束] 按钮即变为有效。

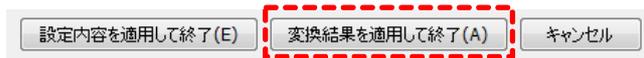


图 7-113: [ 适用转换结果后结束 ] 按钮

按压该按钮，即在样条文件编辑画面的路径点数据中反映转换结果，并关闭帧转换画面。

按压 [ 取消 ] 按钮，即删除转换结果并关闭帧转换画面。

此外，坐标系的设定不适当（同一点·3 点在直线上）时，将显示“帧设定不正确。”的对话框，无法实施帧转换。

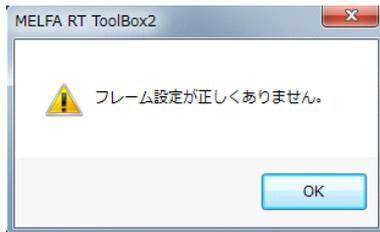


图 7-114: 坐标系的设定不正确时的对话框（执行转换时）

#### ■将坐标系保存至样条文件

设定坐标系后，按压图 7-111 (D) 的 [ 适用设定内容后结束 ] 按钮，即在样条文件编辑画面中存储对坐标系进行定义的位置数据，并关闭帧转换画面。

MvSpl 指令 /EMvSpl 指令的自变量 < 帧转换 > 指定为“1”的情况下，执行样条插补时使用此处设定的坐标系进行帧转换。

按压 [ 取消 ] 按钮，即删除坐标系的设定并关闭帧转换画面。（样条文件编辑画面中存储有坐标系时，该内容不会被更改。）

此外，坐标系的设定不适当（同一点·3 点在直线上）时，将显示表示坐标系的设定不正确的对话框。

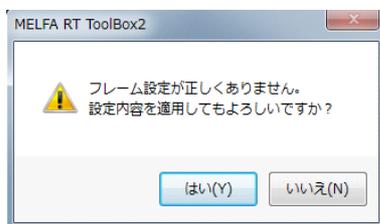


图 7-115: 坐标系的设定不正确时的对话框（保存坐标系时）

按压 [ 是 ] 按钮，即在样条文件编辑画面中存储该设定，并关闭帧转换画面。

按压 [ 否 ] 按钮，则在样条文件编辑画面中不进行存储而返回帧转换画面。

#### ■坐标系的设定清除

要将存储在样条文件编辑画面中的坐标系的设定设为未设定状态，需要按压图 7-111 (E) 的 [ 清除 ] 按钮。点击该按钮，即显示确认对话框。

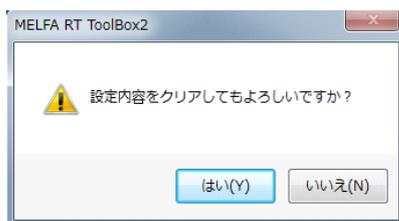


图 7-116: 坐标系清除的确认对话框

按压 [ 是 ] 按钮，存储在样条文件编辑画面中的坐标系的设定即变为未设定状态。

按压 [ 否 ] 按钮，则不更改设定而返回帧转换画面。

### (3) 跳位

跳位至所选择的路径点。

该功能可在 RT ToolBox2 Ver.3.20W 以上的版本中使用。

在调试状态下打开程序后，按压路径点数据一览区域下部的 [ 跳位 ] 按钮，即显示跳位画面。（点击菜单 [ 工具 ] → [ 跳位 ] 也可以显示。）

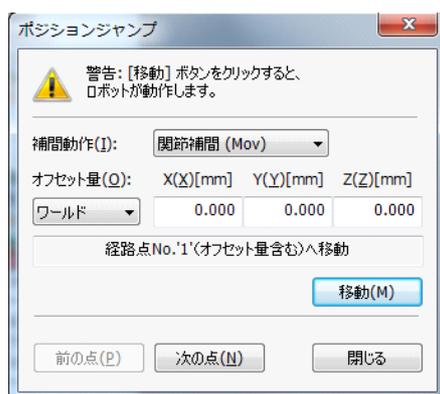


图 7-117: 跳位画面

按压 [ 下 1 点 ] ( [ 上 1 点 ] ) 按钮，对象的路径点即切换为下 1 点 ( 上 1 点 )。

按压 [ 移动 ] 按钮，机器人即以指定的插补动作移动至所选择的路径点的位置 ( 包含偏置量 )。

※RT ToolBox2 Ver.3.40S 以上版本中的偏置量，可从基于世界坐标系进行补偿的补偿量、沿着工具方向进行补偿的补偿量中进行选择。



### 注意

机器人将以与实际系统相同的速度进行动作，因此应注意机器人周围的安全。  
请在随时均可紧急停止的状态下使用。

对象的工程为离线模式时，不显示 [ 跳位 ] 按钮。（同样，无法点击菜单 [ 工具 ] → [ 跳位 ]。）

**(4) 参数 SPLOPTGC**

通过参数 SPLOPTGC，可以调整以样条插补进行动作时的机器人的控制特性。

通常使用出厂设定值，不必更改。

关于曲线部分的动作轨迹，希望更加改善对指令的内环量时，应增大设定值。增大设定值，则有可能改善轨迹精度。

希望抑制机器人的摇晃及振动时，相反地减小设定值则有可能得以改善。

表 7-42: 参数 SPLOPTGC

| 参数           | 参数名称     | 数组数字数 | 内容说明                                                                                                                                                                                                                                      | 出厂时设定值 |
|--------------|----------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 有源增益控制的增益补偿率 | SPLOPTGC | 整数 1  | <p>可以调整执行样条插补时的机器人的控制特性（伺服的响应性）。增大设定值，则可以改善轨迹精度，减小设定值，则可以抑制振动。<br/>设定范围为 1 ~ 200[%]。</p> <p>※ 设定较大的值，则电机有可能振荡或机器人有可能振动。相反设定较小的值，则机器人可能会无法动作而发生伺服错误。应一点一点地更改值后确认动作，在不发生异常的范围内进行设定。</p> <p>※ 力觉控制有效时不起作用。</p> <p>※ 本参数在设定值更改后，无须重新接通电源。</p> | 100    |

## 7.5 关于Ex-T控制

### 7.5.1 概要

#### (1) 特征

Ex-T 控制是以固定于外部的坐标系的原点为机器人的控制点，使机器人动作的功能。用于以下的用途。

- 研磨作业

机器人夹持着加工对象工件，通过使工件触碰到固定安装的砂轮或砂带来进行去毛刺或表面精加工。

- 涂敷作业

机器人夹持着工件，通过固定安装的点胶机对工件涂敷溶剂或粘合剂的作业。



图 7-118: 研磨作业的示例

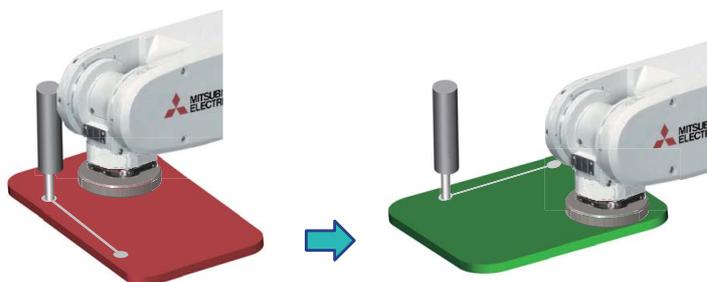


图 7-119: 涂敷作业的示例

对于如上图所示的被固定的工具（砂轮 / 点胶机等），机器人通过使夹持着的工件动作来进行加工（去毛刺 / 研磨 / 密封等）时，保持工具与工件的相对位置的同时，生成基于所指定的加工路径动作的程序是非常困难的。

Ex-T 控制是用于轻松地实现这些作业或编程的功能。可将砂轮或点胶机的位置登录到机器人，并以该位置为基准使其进行直线 / 圆弧动作。

(2) 规格

| 项目                | 规格                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |     |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----|----|-----------------|-------------|-----|-----------------|-------------|-----|-----------------|-------------|-----|-------------------|---------------|-----|-------------------|---------------|-----|------|-------|----|----------|------------|-----|
| 对应的机器人            | 垂直多关节（6轴型）机器人、水平多关节（4轴型）机器人<br>※ 在垂直多关节（5轴型）机器人与用户机械中无法使用。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| 对应的机器人语言          | MELFA-BASIC V<br>· 在 MELFA-BASIC V 中已增加以下的指令。<br><table border="1" style="margin-left: 20px;"> <thead> <tr> <th>指令</th> <th>说明</th> <th>页码</th> </tr> </thead> <tbody> <tr> <td>EMvs (E Move S)</td> <td>Ex-T 控制直线插补</td> <td>209</td> </tr> <tr> <td>EMvc (E Move C)</td> <td>Ex-T 控制圆形插补</td> <td>203</td> </tr> <tr> <td>EMvr (E Move R)</td> <td>Ex-T 控制圆弧插补</td> <td>204</td> </tr> <tr> <td>EMvr2 (E Move R2)</td> <td>Ex-T 控制圆弧插补 2</td> <td>206</td> </tr> <tr> <td>EMvr3 (E Move R3)</td> <td>Ex-T 控制圆弧插补 3</td> <td>207</td> </tr> </tbody> </table><br>· 已更改 MELFA-BASIC V 的以下的状态变量的功能。<br><table border="1" style="margin-left: 20px;"> <thead> <tr> <th>变量名称</th> <th>数组元素数</th> <th>页码</th> </tr> </thead> <tbody> <tr> <td>P_WkCord</td> <td>工件坐标号（1～8）</td> <td>364</td> </tr> </tbody> </table> | 指令  | 说明 | 页码 | EMvs (E Move S) | Ex-T 控制直线插补 | 209 | EMvc (E Move C) | Ex-T 控制圆形插补 | 203 | EMvr (E Move R) | Ex-T 控制圆弧插补 | 204 | EMvr2 (E Move R2) | Ex-T 控制圆弧插补 2 | 206 | EMvr3 (E Move R3) | Ex-T 控制圆弧插补 3 | 207 | 变量名称 | 数组元素数 | 页码 | P_WkCord | 工件坐标号（1～8） | 364 |
| 指令                | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 页码  |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvs (E Move S)   | Ex-T 控制直线插补                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 209 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvc (E Move C)   | Ex-T 控制圆形插补                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 203 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvr (E Move R)   | Ex-T 控制圆弧插补                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 204 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvr2 (E Move R2) | Ex-T 控制圆弧插补 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 206 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvr3 (E Move R3) | Ex-T 控制圆弧插补 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 207 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| 变量名称              | 数组元素数                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 页码  |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| P_WkCord          | 工件坐标号（1～8）                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 364 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| Ex-T 坐标设定个数       | 最多 8 个（通过参数 / 状态变量进行坐标系的设定）                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |

(3) 必要的设备与软件版本

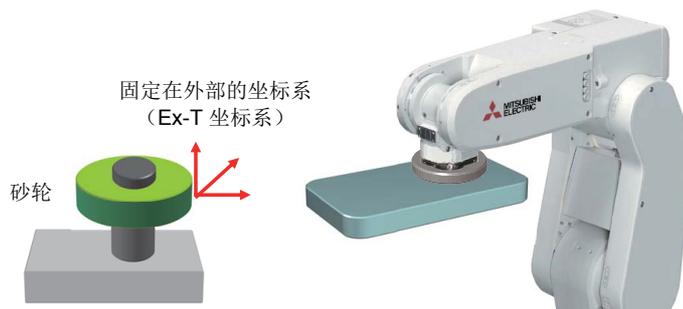
Ex-T 控制可在下表所示的机器人控制器的软件版本中使用。

表 7-43: 可使用 Ex-T 控制的软件版本

| 设备       | 控制器                                      | 示教单元                                               |
|----------|------------------------------------------|----------------------------------------------------|
| 可使用的软件版本 | F-Q 系列: Ver. R5 以上<br>F-D 系列: Ver. S5 以上 | R32TB/R33TB: Ver.1.3 以上<br>R56TB/R57TB: Ver.2.3 以上 |

7.5.2 Ex-T坐标的设定

使用 Ex-T 控制时，需要设定固定在外部的基准坐标（Ex-T 坐标）。



- 在用固定安装的砂轮进行的研磨作业中，要研磨的位置为 Ex-T 坐标的原点。
- 在用固定安装的点胶机进行的涂敷作业中，喷嘴前端的位置为 Ex-T 坐标的原点。

(1) 设定方法

Ex-T 坐标的设定使用与工件坐标相同的参数 / 状态变量进行设定。与工件坐标系相同，通过以下设定进行。

- 使用示教单元或 RT ToolBox2 进行参数设定
- 对机器人程序（MELFA-BASIC V）中的系统状态变量进行设定

Ex-T 坐标（工件坐标）相关的参数如下表所示。

| 参数名称                 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WKnCORD<br>n 为 1 ~ 8 | <p>工件坐标（Ex-T坐标）的坐标值。<br/>（X、Y、Z、A、B、C）单位：mm或度</p> <p>用作工件JOG操作中的基准坐标或工件坐标数据。也用作Ex-T控制中的控制点（Ex-T坐标）。<br/>用作工件坐标数据时，有效的轴成分会因机器人机型不同而有所不同。请参照第434页的【5.7关于标准基本坐标】。</p> <p>虽然可通过示教单元中的工件坐标设定（定义）操作设定定义的坐标值，但通过在本参数中输入坐标值，也可以定义工件坐标。此时，用于通过以下的示教来定义工件坐标的3点（参数：WKnWO、WKnWX、WKnWY（n为1~8））的各坐标值将清零。</p> <div style="text-align: center;"> <p>工件坐标系（Ex-T坐标系）KnCORD</p> <p>示教点：WO WKnWO</p> <p>示教点：WX WKnWX</p> <p>示教点：WY WKnWY</p> <p>基本坐标系</p> </div> <p>注) 工件坐标（Ex-T坐标）的示教，在基本转换为初始值的状态（世界坐标系与基本坐标系一致的状态）下进行将更易于管理。特别在定义多个工件坐标时，建议用于防止混乱</p> |
| WKnWO<br>n 为 1 ~ 8   | <p>为用于定义工件坐标（Ex-T坐标）的示教位置，是工件坐标（Ex-T坐标）的原点位置。（对应示教单元中的示教操作的“WO”。参照上图）<br/>（X、Y、Z）单位：mm</p> <p>注) 如仅输入该坐标值，则工件坐标（Ex-T坐标）不会被定义。请通过示教单元工件坐标设定画面的[定义]操作或RT ToolBox2的工件坐标参数画面的[写入]操作进行工件坐标的计算处理。</p>                                                                                                                                                                                                                                                                                                                                      |
| WKnWX<br>n 为 1 ~ 8   | <p>为用于定义工件坐标（Ex-T坐标）的示教位置，是工件坐标（Ex-T坐标）的+X轴上的位置。（对应示教单元中的示教操作的“WX”。参照上图）<br/>（X、Y、Z）单位：mm</p> <p>注) 如仅输入该坐标值，则工件坐标（Ex-T坐标）不会被定义。请通过示教单元工件坐标设定画面的[定义]操作或RT ToolBox2的工件坐标参数画面的[写入]操作进行工件坐标的计算处理。</p>                                                                                                                                                                                                                                                                                                                                   |
| WKnWY<br>n 为 1 ~ 8   | <p>为用于定义工件坐标（Ex-T坐标）的示教位置，是工件坐标（Ex-T坐标）的X-Y平面上的+Y轴侧的位置。（对应示教单元中的示教操作的“WY”。参照上图）<br/>（X、Y、Z）单位：mm</p> <p>注) 如仅输入该坐标值，则工件坐标（Ex-T坐标）不会被定义。请通过示教单元工件坐标设定画面的[定义]操作或RT ToolBox2的工件坐标参数画面的[写入]操作进行工件坐标的计算处理。</p>                                                                                                                                                                                                                                                                                                                            |

在使用 WKnWO、WKnWX、WKnWY 的 3 点示教中设定工件坐标（Ex-T 坐标）时，如设定工具数据使得示教的基准位置为控制点则会更加方便。

关于使用 R33TB 或 R32TB 进行 3 点示教的操作方法，请参照另一手册“使用说明书 / 从机器人本体安装到维护”的“2.4 动作的确认”的“（6）工件 JOG 操作”。

### 7.5.3 Ex-T JOG

Ex-T JOG 是以工件坐标（Ex-T 坐标）为控制点，基于工件坐标系（Ex-T 坐标系）进行 JOG 动作的功能。因为是基于工件坐标系进行动作，所以虽然与以往的工件 JOG 的动作相似，但 Ex-T JOG 与以往的工件 JOG 相比，姿势成分的相关动作会有所不同。

Ex-T JOG 操作通过示教单元的工件 JOG 操作进行。与以往的工件 JOG 的动作模式的切换是通过每个工件坐标系（Ex-T 坐标系）的参数 WK1JOGMD ~ WK8JOGMD 的设定来进行的。

| 参数名称                  | 内容说明                                                                                                                                  |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| WKnJOGMD<br>n 为 1 ~ 8 | 对每个工件坐标设定工件JOG操作中的动作模式。<br>0: 工件JOG (A、B、C的动作为绕工件坐标的X轴、Y轴、Z轴的平行轴的旋转。控制点的位置不变。)<br>1: Ex-T JOG (A、B、C的动作为以工件坐标的X轴、Y轴、Z轴为中心移动控制点的同时旋转。) |

参数 WK1JOGMD ~ WK8JOGMD 的设定导致的动作差异，按机器人机型不同分别如表所示。

#### RV6 轴型的工件 JOG 的动作

| 工件 JOG 的动作模式              | 工件 JOG                       | Ex-T JOG                       |
|---------------------------|------------------------------|--------------------------------|
| 参数 WKnJOGMD (n=1 ~ 8) 的设定 | 0 (初始值)                      | 1                              |
| XYZ 键的动作                  | 基于工件坐标系的各轴移动。                | 与以往的工件 JOG 相同。                 |
| ABC 键的动作                  | 在保持控制点位置不变的情况下，基于工件坐标系，改变方向。 | 以工件坐标系的各轴为中心轴，改变控制点位置的同时，改变方向。 |

#### RH4 轴型、RH4 轴垂吊型

| 工件 JOG 的动作模式              | 工件 JOG                       | Ex-T JOG                                           |
|---------------------------|------------------------------|----------------------------------------------------|
| 参数 WKnJOGMD (n=1 ~ 8) 的设定 | 0 (初始值)                      | 1                                                  |
| XYZ 键的动作                  | 基于工件坐标系的各轴移动                 | 与以往的工件 JOG 相同                                      |
| C 键的动作                    | 在保持控制点位置不变的情况下，基于工件坐标系，改变方向。 | 以工件坐标系 (EX-T 坐标系) 的 Z 轴 (Zw) 为中心轴，改变控制点位置的同时，改变方向。 |
| AB 键的动作                   | 机器人不动作。                      | 机器人不动作。                                            |

以往的工件 JOG 与 Ex-T JOG 相比，机器人的姿势成分的动作有所不同。下面以 C 成分的动作为例对差异进行说明。

#### (1) 工件 JOG 中的姿势成分动作

工件 JOG 中的姿势成分的动作，是在控制点的绕工件坐标的 XYZ 轴的平行轴的旋转。此时，位置为固定不变。

工件 JOG 中 C 成分动作示例如图 7-120 所示。

W0-Wx-Wy 表示工件坐标系 (看作从 +Wz 看到的工件坐标系的图)。●为机器人的控制点 (TCP)，圆角的四方形为机器人夹持的工件。虚线表示移动后的工件位置 (姿势)。

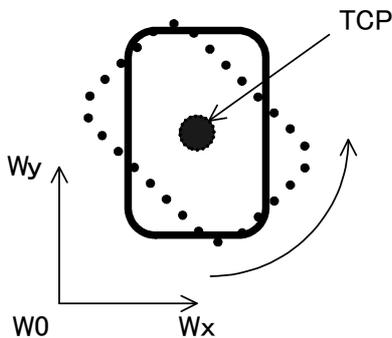


图 7-120: 工件 JOG 的 C 成分动作

#### (2) Ex-T JOG 中的姿势成分动作

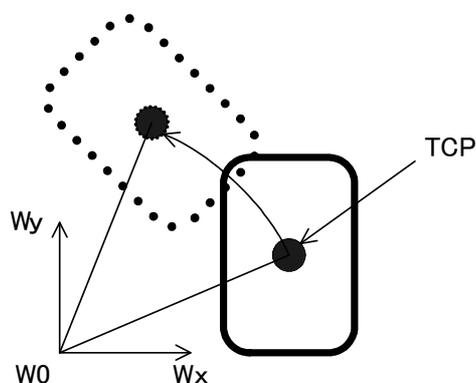
Ex-T JOG 中的姿势成分的动作，为绕 Ex-T 坐标系 (工件坐标系) 的 XYZ 轴的旋转。此时，机器人的位置也发生变化。

Ex-T JOG 中 C 成分动作示例如图 7-121 的 <动作示例 1>、<动作示例 2> 所示。

W0-Wx-Wy 表示 Ex-T 坐标系（工件坐标系）（看作从 +Wz 看到的 Ex-T 坐标系的图）。●为机器人的控制点（TCP），圆角的四方为机器人夹持的工件。虚线表示移动后的工件位置（姿势）。

<动作示例 1> 为 Ex-T 坐标系（工件坐标系）与工件分开的情况，<动作示例 2> 为 Ex-T 坐标系（工件坐标系）的原点（或 Z 轴）与工件接触的情况。两者均以 W0 为中心进行旋转动作。

<动作示例 1>



<动作示例 2>

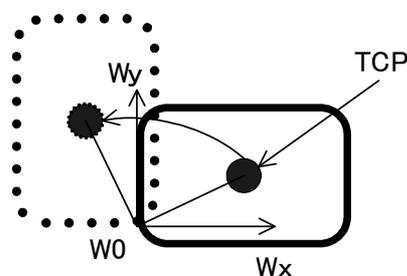


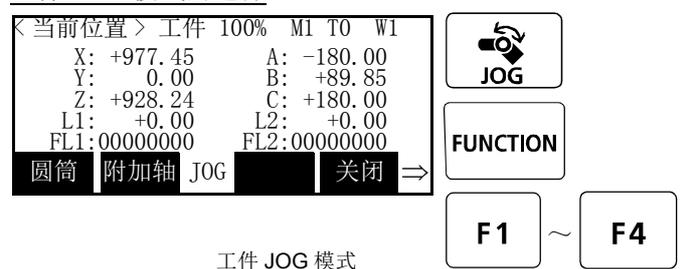
图 7-121: Ex-T JOG 的 C 成分动作

### (3) Ex-T JOG 的操作

Ex-T JOG 的操作与工件 JOG 的操作方法相同。

需要预先设定 EX-T 坐标（工件坐标）和动作模式（参数 WK1JOGMD ~ WK8JOGMD）。

#### 工件 JOG 模式的选择



注) 显示的坐标值是基于直交坐标系的值。

按压 [JOG] 键以显示 JOG 画面。（画面下侧显示“JOG”）

确认画面上方显示为 JOG 模式的“WORK（工件）”。

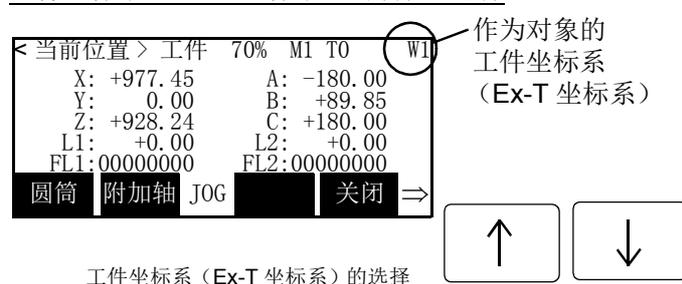
显示为其它 JOG 模式时

请按压“工件”对应的功能键。

（在画面下方未显示希望的 JOG 模式时，按压 [FUNCTION] 键可使其显示）

结束 JOG 操作时，应再次按压 [JOG] 键，或按压“关闭”对应的功能键。

#### 工件坐标系（Ex-T 坐标系）的确认·选择



请对工件 JOG（Ex-T JOG）动作的对象工件坐标系（Ex-T 坐标系）进行确认。在画面右上方的 8 个工件坐标系（Ex-T 坐标系）内，显示当前对象号码。（W1 ~ W8）

未显示希望的工件坐标系（Ex-T 坐标系）时，可通过方向键（[↑]、[↓]）进行更改。

每次按压 [↑] 键将以 W1 → W2...W7 → W8 的顺序增大，按压 [↓] 键时向相反方向减少。

### ⚠ 注意

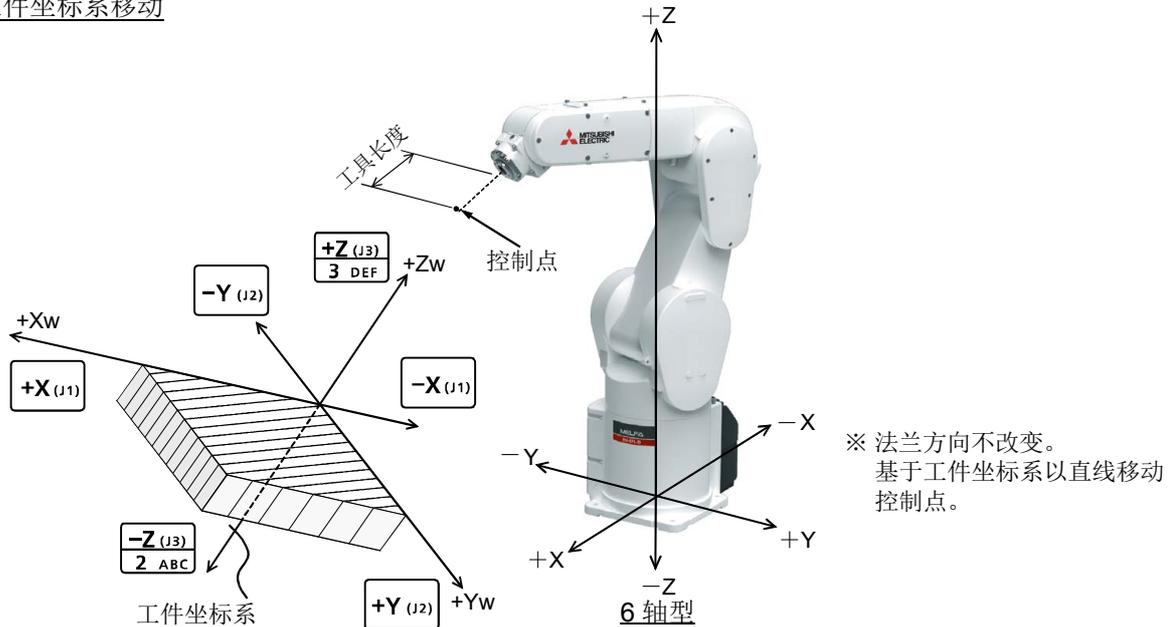
请务必确认对象工件坐标系（Ex-T 坐标系）的号码显示正确。（画面右上方的 W1 ~ W8 的显示）

在错误的工件坐标系（Ex-T 坐标系）中，机器人将向意料以外的方向执行动作，有可能导致设备损坏或人身事故。

(4) RV6 轴型的工件 JOG 动作

通过 XYZ 键进行的动作，在工件 JOG 模式与 Ex-T JOG 模式中，为相同的动作。

基于工件坐标系移动

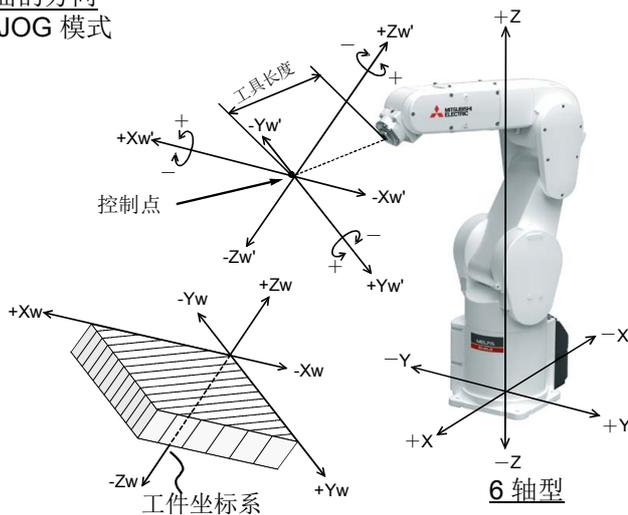


- 按压 **[+X(J1)]** 键时沿着工件坐标系的 X 轴 ( $X_w$ ) 的正方向移动。  
按压 **[-X(J1)]** 键时沿着负方向移动。
- 按压 **[+Y(J2)]** 键时沿着工件坐标系的 Y 轴 ( $Y_w$ ) 的正方向移动。  
按压 **[-Y(J2)]** 键时沿着负方向移动。
- 按压 **[+Z(J3)]** 键时沿着工件坐标系的 Z 轴 ( $Z_w$ ) 的正方向移动。  
按压 **[-Z(J3)]** 键时沿着负方向移动。

通过 ABC 键进行的动作，在工件 JOG 模式与 Ex-T JOG 模式中，为不同的动作。

改变法兰面的方向

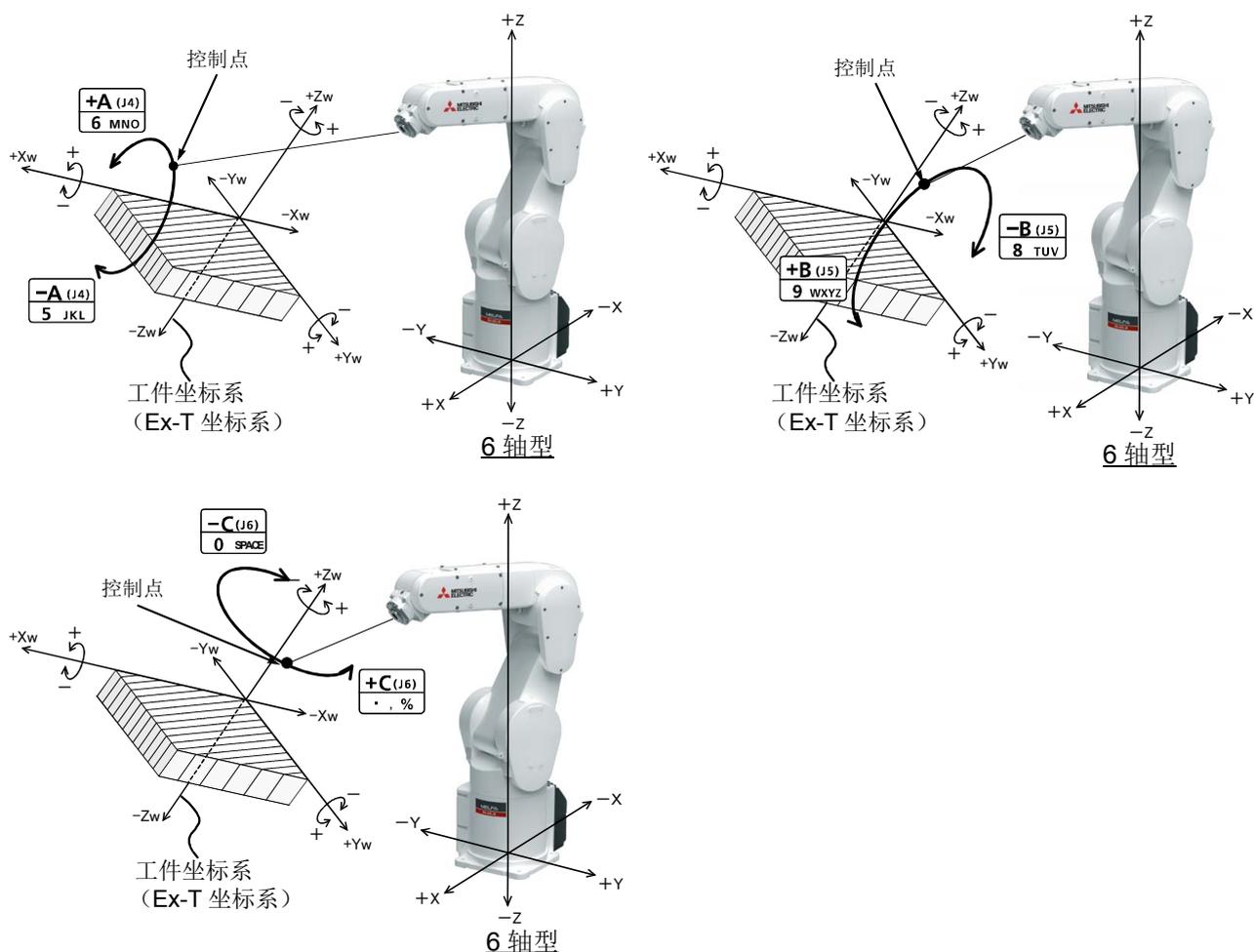
①工件 JOG 模式



※ 法兰的位置不改变。  
基于工件坐标系改变法兰的方向。

- 按压 **[+A(J4)]** 键时向 X 轴的正方向旋转。  
按压 **[-A(J4)]** 键时向负方向旋转。
- 按压 **[+B(J5)]** 键时向 Y 轴的正方向旋转。  
按压 **[-B(J5)]** 键时向负方向旋转。
- 按压 **[+C(J6)]** 键时向 Z 轴的正方向旋转。  
按压 **[-C(J6)]** 键时向负方向旋转。

## ② Ex-T JOG 模式



※ 控制点以工件坐标系（Ex-T 坐标系）的各轴为中心旋转。

使用 [+A (J4)] 键与 [-A (J4)] 键时以  $X_w$  轴为中心、使用 [+B (J5)] 键与 [-B (J5)] 键时以  $Y_w$  轴为中心、使用 [+C (J6)] 键与 [-C (J6)] 键时以  $Z_w$  轴为中心进行控制点旋转之类的动作。

- 按压 [+A(J4)] 键时，以工件坐标系（Ex-T 坐标系）的 X 轴（ $X_w$ ）为中心，控制点向正方向旋转。  
按压 [-A(J4)] 键时向负方向旋转。
- 按压 [+B(J5)] 键时，以工件坐标系（Ex-T 坐标系）的 Y 轴（ $Y_w$ ）为中心，控制点向正方向旋转。  
按压 [-B(J5)] 键时向负方向旋转。
- 按压 [+C(J6)] 键时，以工件坐标系（Ex-T 坐标系）的 Z 轴（ $Z_w$ ）为中心，控制点向正方向旋转。  
按压 [-C(J6)] 键时向负方向旋转。

## (5) RH4 轴型的工件 JOG 动作

通过 XYZ 键进行的动作，在工件 JOG 模式与 Ex-T JOG 模式中，为相同的动作。

基于工件坐标系移动

- 按压 **[+X(J1)]** 键时沿着工件坐标系的 X 轴 ( $X_w$ ) 的正方向移动。  
按压 **[-X(J1)]** 键时沿着负方向移动。
- 按压 **[+Y(J2)]** 键时沿着工件坐标系的 Y 轴 ( $Y_w$ ) 的正方向移动。  
按压 **[-Y(J2)]** 键时沿着负方向移动。
- 按压 **[+Z(J3)]** 键时沿着工件坐标系的 Z 轴 ( $Z_w$ ) 的正方向移动。  
按压 **[-Z(J3)]** 键时沿着负方向移动。

通过 C 键进行的动作，在工件 JOG 模式与 Ex-T JOG 模式中，为不同的动作。  
AB 键不动作。

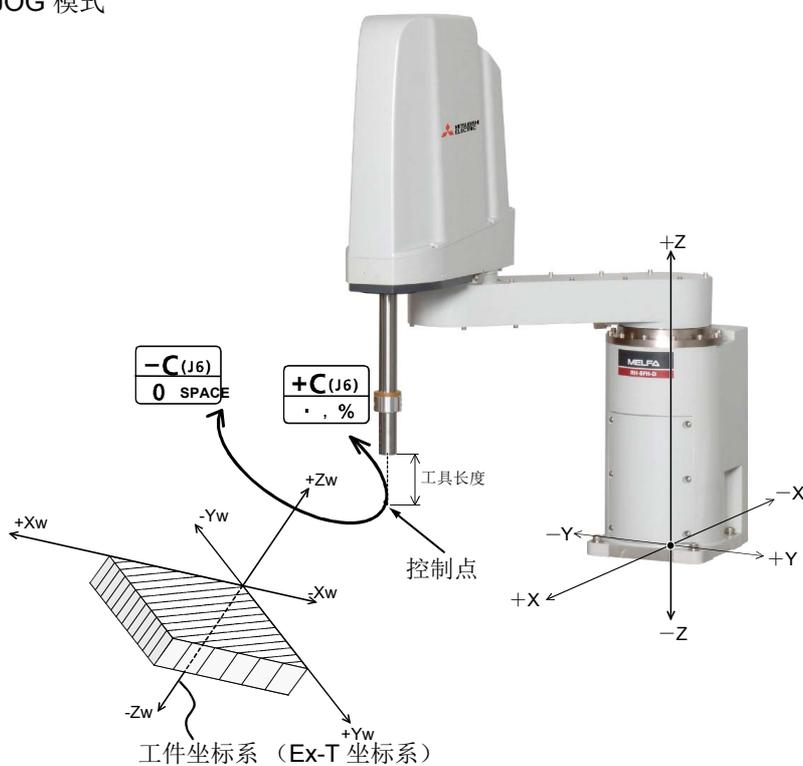
### 改变前端的方向

#### ① 工件 JOG 模式



- 按压 **[+C(J6)]** 键时前端向正方向旋转。  
按压 **[-C(J6)]** 键时前端向负方向旋转。

#### ② Ex-T JOG 模式

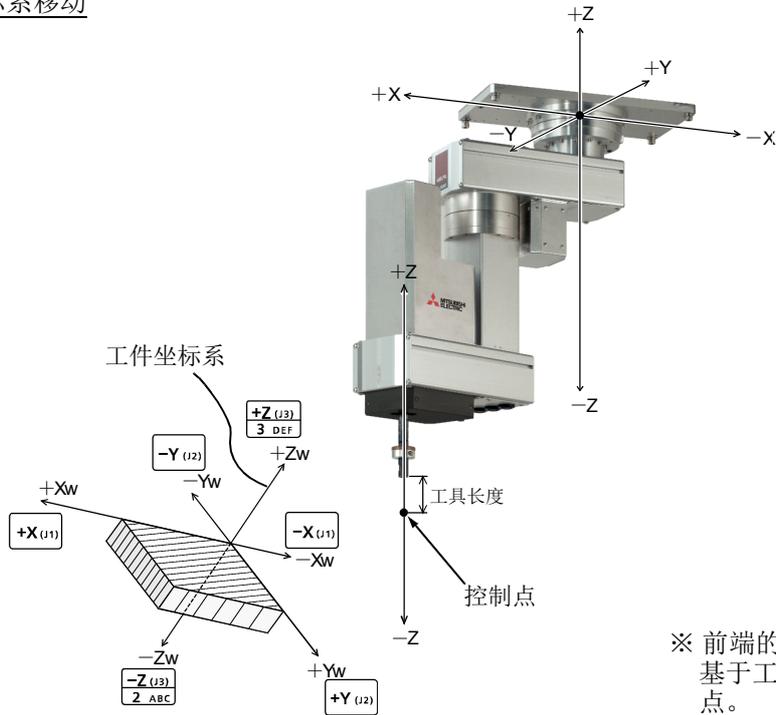


- 按压 **[+C(J6)]** 键时，以工件坐标系（Ex-T 坐标系）的 Z 轴（Zw）为中心，控制点向正方向旋转。  
按压 **[-C(J6)]** 键时，向负方向旋转。

## (6) RH4 轴垂吊型的工件 JOG 动作

通过 XYZ 键进行的动作，在工件 JOG 模式与 Ex-T JOG 模式中，为相同的动作。

## 基于工件坐标系移动

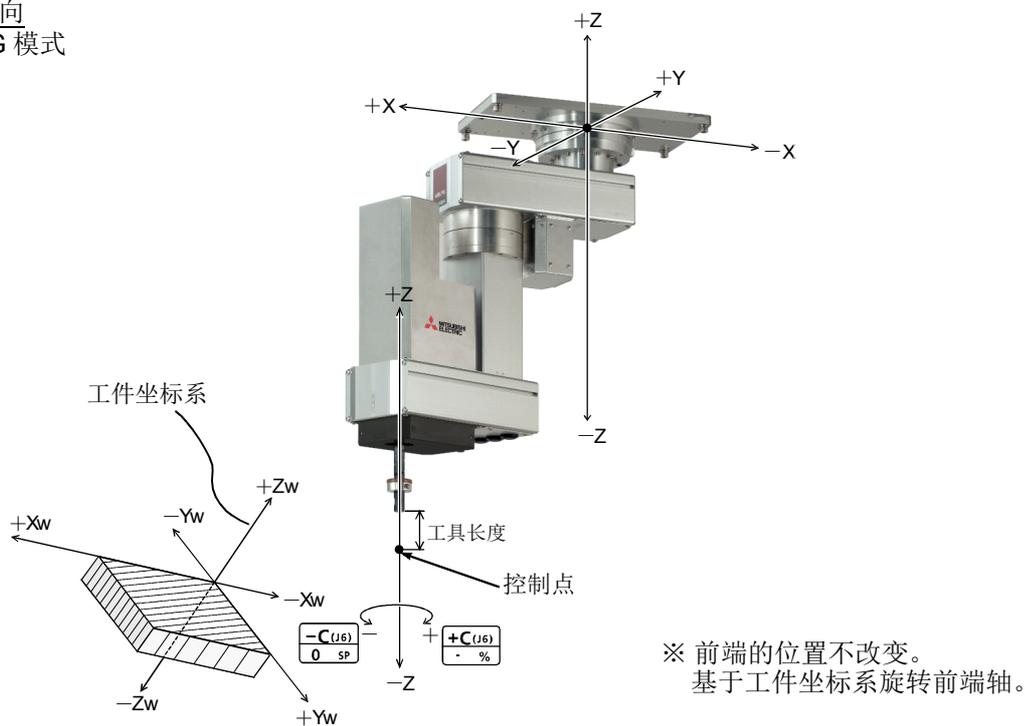


- 按压 **[+X(J1)]** 键时沿着工件坐标系的 X 轴 ( $X_w$ ) 的正方向移动。  
按压 **[-X(J1)]** 键时沿着负方向移动。
- 按压 **[+Y(J2)]** 键时沿着工件坐标系的 Y 轴 ( $Y_w$ ) 的正方向移动。  
按压 **[-Y(J2)]** 键时沿着负方向移动。
- 按压 **[+Z(J3)]** 键时沿着工件坐标系的 Z 轴 ( $Z_w$ ) 的正方向移动。  
按压 **[-Z(J3)]** 键时沿着负方向移动。

通过 C 键进行的动作，在工件 JOG 模式与 Ex-T JOG 模式中，为不同的动作。  
AB 键不动作。

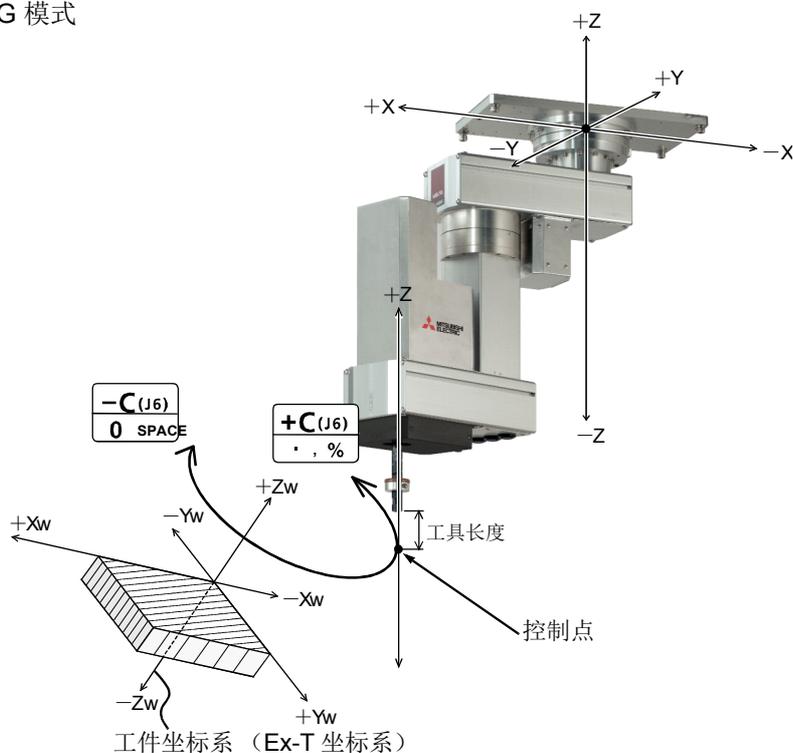
### 改变前端的方向

#### ①工件 JOG 模式



- 按压  $[+C(J6)]$  键时前端向正方向旋转。  
按压  $[-C(J6)]$  键时前端向负方向旋转。

#### ② Ex-T JOG 模式



- 按压  $[+C(J6)]$  键时，以工件坐标系 (Ex-T 坐标系) 的 Z 轴 ( $Zw$ ) 为中心，控制点向正方向旋转。  
按压  $[-C(J6)]$  键时，向负方向旋转。

## 7.5.4 机器人程序的创建

## (1) Ex-T 控制相关指令·变量的一览

Ex-T 控制相关的 MELFA-BASIC V 的指令与变量的一览如下所示。

指令的详细说明请参照表的参照页码。

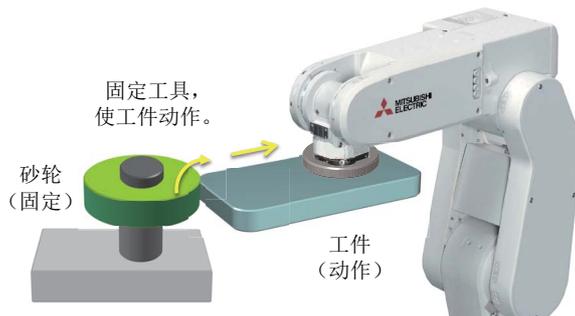
表 7-44: Ex-T 控制相关的指令

| No. | 指令的种类 | 增加指令              | 功能                     | 参照页码 |
|-----|-------|-------------------|------------------------|------|
| 1   | 直线插补  | EMvs (E Move S)   | 基于工件坐标 (Ex-T 坐标) 的直线插补 | 209  |
| 2   | 圆形插补  | EMvc (E Move C)   | 基于工件坐标 (Ex-T 坐标) 的圆形插补 | 203  |
| 3   | 圆弧插补  | EMvr (E Move R)   | 基于工件坐标 (Ex-T 坐标) 的圆弧插补 | 204  |
| 4   |       | EMvr2 (E Move R2) |                        | 206  |
| 5   |       | EMvr3 (E Move R3) |                        | 207  |

表 7-45: Ex-T 控制相关的状态变量

| No. | 变量名称     | 功能概要       | 补充说明         | 参照页码 |
|-----|----------|------------|--------------|------|
| 1   | P_WkCord | 工件坐标的读取、设定 | 也兼用 Ex-T 坐标。 | 364  |

## (2) 编程示例



用于进行如图所示动作的程序示例如下。

在固定的加工夹具上，机器人夹持工件进行描绘的动作。(图 1) ~ (图 5))

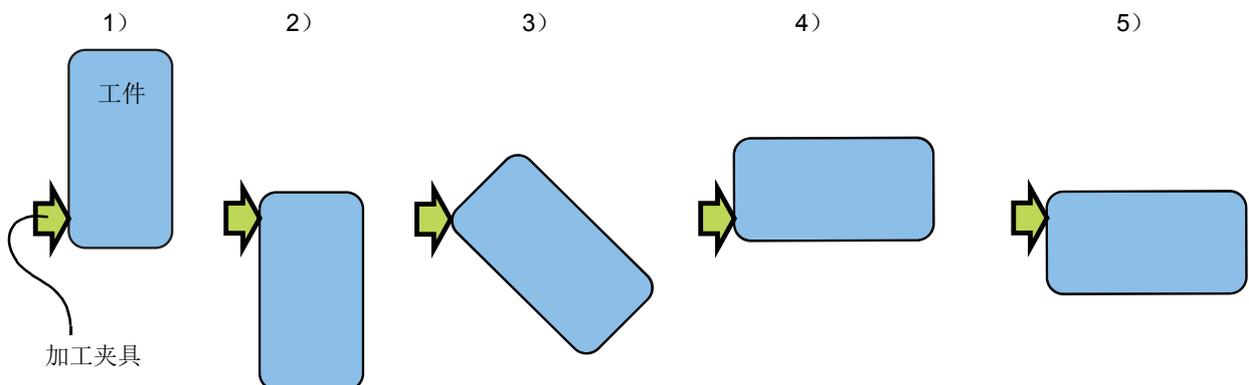


图 7-122: 作业示例 1

重叠该图则实际的动作如下所示。

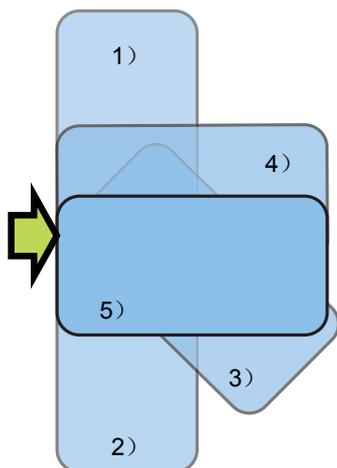
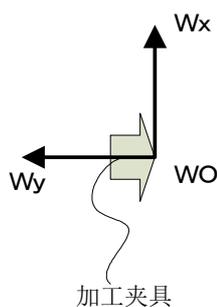


图 7-123: 作业示例 2

### ■ 步骤 1: 工件坐标 (Ex-T 坐标) 的设定

设定工件坐标 (Ex-T 坐标), 使图中加工夹具的与工件之间接触的部位成为工件坐标 (Ex-T 坐标) 原点。(此处, 看作是设定为工件坐标 1。)



此外, 为了可以进行基于该工件坐标系的 JOG 动作, 要将参数 WK1JOGMD 设定为「1 (Ex-T JOG 模式)」。

### ■ 步骤 2: 位置的示教

让机器人实际夹持工件, 进行位置的示教。

此处, 对图的 1) ~ 5) 的位置进行示教。

进行位置示教时, 如通过工件 JOG (Ex-T JOG) 进行机器人的 JOG 进给, 则可以进行基于加工夹具的 JOG 进给。

### ■ 步骤 3: 程序的创建

创建 MELFA-BASIC V 的程序。

(实际上, 还需要工件的夹持动作及信号输入输出等, 但此处省略。)

#### 【程序示例】

```

Mov P001 ' 向 1) 的位置移动
Dly 0.5
Spd 50 ' 将加工速度 (工件的移动速度) 设定为 50mm/sec
EMvs 1, P002 ' 沿着工件坐标 1 以 Ex-T 直线插补向 2) 的位置移动
EMvr 1, P002,P003,P004 ' 沿着工件坐标 1 以 Ex-T 圆弧插补从 2) 经由 3) 向 4) 移动。
EMvs 1, P005 ' 沿着工件坐标 1 以 Ex-T 直线插补向 5) 的位置移动
:

```

该示例中, 示教位置为 5 点, 但实际上有可能需要根据工件形状或加工的情况来增加示教位置。同时也需根据变化情况更改程序。





**mitsubishi electric corporation**