



三菱电机工业机器人

CR800 系列控制器

操作说明书

功能和操作的详细说明

本书详细记载了 iQ Platform 对应的 CR800-R/CR860-R/CR800-Q/CR860-Q 控制器、独立设备型 CR800-D/CR860-D 控制器的功能及操作方法。

MELFA
BFP-A3576-V

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)
→紧急停止开关的设置

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篱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** 包含抓手在内的工件质量应不超过额定负载及允许转矩。超过时会导致发生错误或故障等。
-  **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

通过 CR800-R/Q 系列使用 iQ Platform 对应产品时，请勿将手持式 GOT 连接至可编程控制器。无论操作权有效 / 无效，手持式 GOT 都可以使机器人自动运行，可能导致设备损坏及人身事故。



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，接线完毕后，务必对示教单元紧急停止、用户紧急停止、门开关等的各种功能能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（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。

注意

为了保证机器人及系统的网络安全（可用性、完整性、机密性），对于来自不可信网络或经由网络的设备的非法访问、拒绝服务攻击 (DoS^{*1} 攻击) 以及计算机病毒等其他网络攻击，应采取设置防火墙与虚拟专用网络 (VPN)，以及在计算机上安装杀毒软件等对策。

因非法访问、拒绝服务攻击 (DoS 攻击)、计算机病毒以及其他网络攻击引发的机器人及系统方面的各种问题，三菱电机不承担责任。

*1 DoS: 耗费目标计算机的资源或使其安全性变得脆弱，导致其无法提供正常服务，以及此种状态。

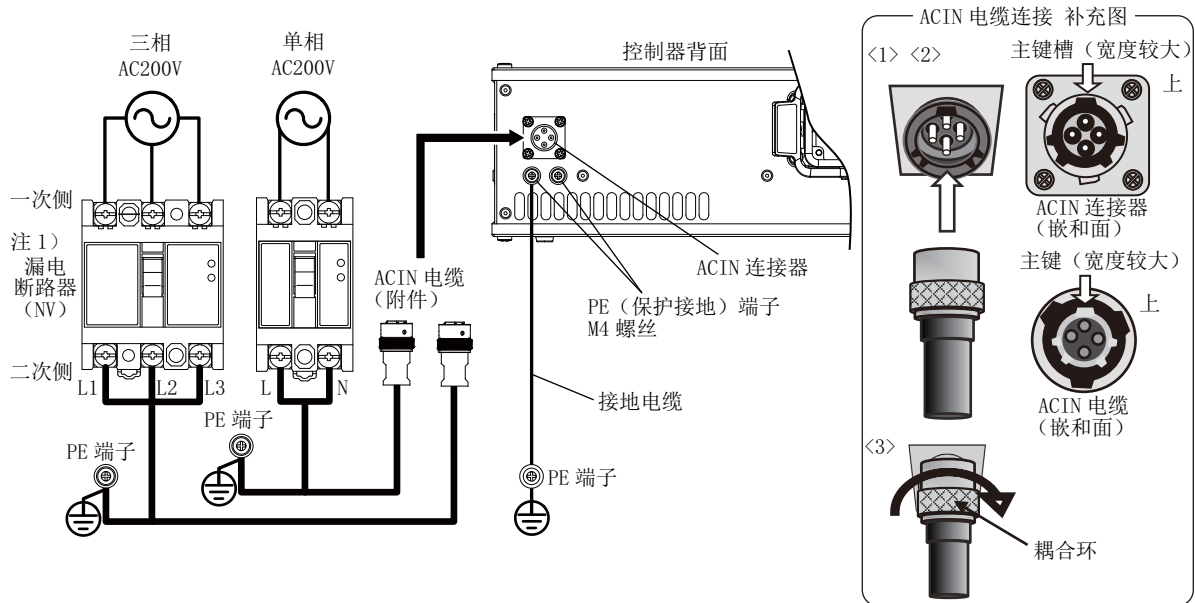
*CR800-D/R/Q 控制器

基本构成的注意事项如下所示。



注意

为了漏电保护，应对控制器的一次电源安装漏电断路器。否则有可能导致触电事故。



注 1) 务必安装端子盖板来使用漏电断路器。

1) 应自备以下产品。

产品名称	规格	备注
漏电断路器	以下记载推荐品。 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)	用户自备
一次侧电源连接用电缆	AWG #14 (2mm ²) 以上	用户自备。 端子紧固螺丝的紧固力矩: 2 ~ 3N·m
接地电缆	AWG #14 (2mm ²) 以上	用户自备。 端子紧固螺丝的紧固力矩: 2 ~ 3N·m
ACIN 电缆	端子尺寸: M5、电缆长度 3m	产品附带

- 2) 应确认一次侧电源符合电源规格。
- 3) 确认已切断一次侧电源和漏电断路器的电源开关已变为 OFF 状态。
- 4) 连接 ACIN 电缆。
将 ACIN 电缆的电源端子连接到漏电断路器的二次侧端子上。另外，应接地连接 FG 端子。
- 5) 应将 ACIN 电缆连接到控制器背面的 ACIN 连接器上。
<1> ACIN 电缆插头的主键 (宽度较大) 朝上。(参照 ACIN 电缆连接补充图)
<2> 与 ACIN 连接器侧的主键槽 (宽度较大) 对准，沿水平方向将 ACIN 电缆插入最里端。
插入时如果键槽吻合或处于倾斜状态，则可能导致连接器损坏。
<3> 应将 ACIN 电缆的耦合环向右旋转锁紧。
- 6) 由于 EN61800-5-1 的 Touch Current 在 3.5mA 以上，因此应将接地电缆连接到控制器的 PE (保护接地) 端子上，另一端接地连接 (2 点接地)。
- 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。

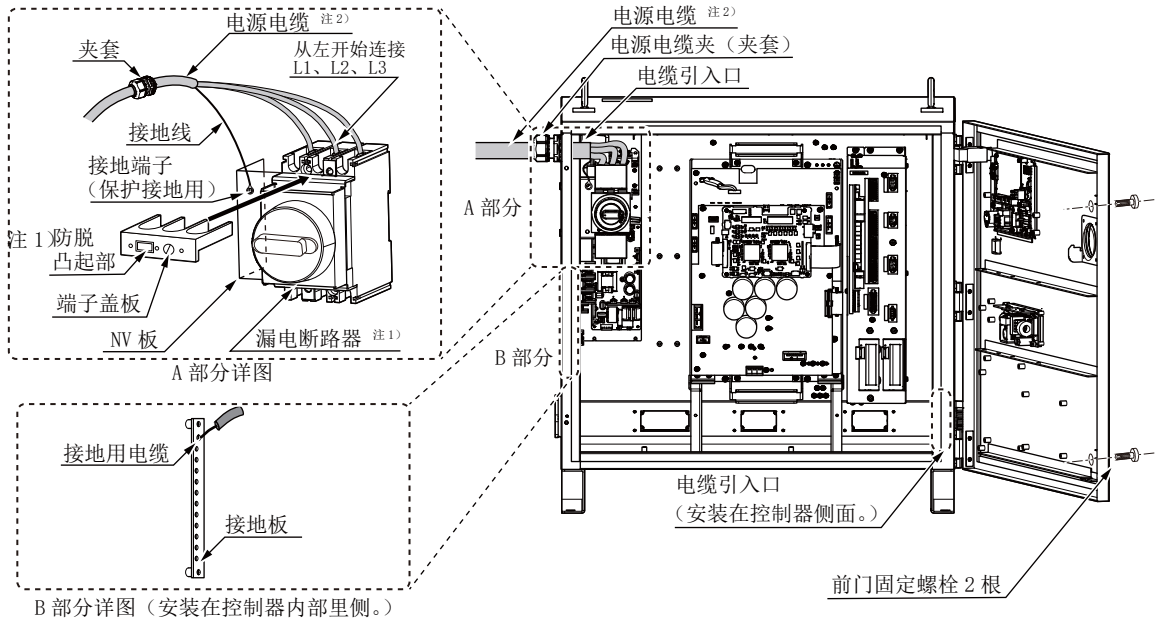
*CR860-D/R/Q 控制器

基本构成的注意事项如下所示。



注意

在机器人控制器内部安装模块或连接电缆时，请勿直接接触导电部分、基板及电子部件。否则将导致控制器的误动作或发生故障。



B 部分详图（安装在控制器内部里侧。）

注 1) 务必安装端子盖板来使用漏电断路器。

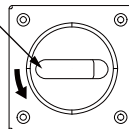
注 2) 关于一次侧电源的噪声（浪涌）对策的示例，请参照“CR860 Controller RV-35FR/50FR/80FR Series Standard Specifications Manual (BFP-A3779)”。

• 电源电缆的连接

1) 应准备电源用（AWG#8（8mm²）以上 3 芯）和接地用（AWG#8（8mm²）以上 1 芯）的电源电缆（电缆外径：19mm ~ 23mm）及保护接地用接地线（AWG#8（8mm²））。电源电缆应使用电源线与接地线为一体的多芯电缆。

2) 反时针方向旋转控制器正面的旋钮，即可开门。

反时针方向旋转
漏电断路器的旋钮。



3) 用指尖拨动漏电断路器的端子盖板表面的防脱凸起部以将其拔出。

4) 应确认一次侧电源符合规格。

5) 应确认已切断一次侧电源及控制器的漏电断路器为 OFF 状态。

6) 应从控制器侧面的电缆引入口插入电源电缆，并用电源电缆夹（夹套）进行固定。

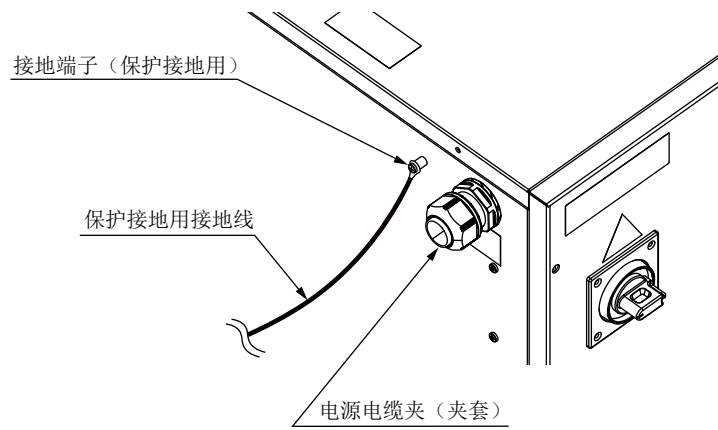
7) 将电源电缆连接到漏电断路器的 L1、L2、L3 端子（M5 螺栓）上。

8) 应将电源电缆的接地线连接到 NV 板的接地端子（保护接地用）（M6 螺栓）上。

9) 应将电源电缆的接地线连接到 NV 板的接地端子（保护接地用）（M6 螺栓）上。

10) 关闭控制器的前门，拧紧固定螺栓将其固定。如果未拧紧前门的固定螺栓，则不符合 IP54 标准。

- 11) 将保护接地用的接地线连接到电源电缆夹（夹套）旁边的接地端子（保护接地用）（M6 螺栓）上。



- 电源电缆的连接
需进行功能接地时，应将接地用电缆连接至 B 部分的接地板的空位上。请勿拆除最初连接的电缆。

修订记录

印刷日期	手册编号	修改内容
2017-09-30	BFP-A3576	• 第一版
2017-12-27	BFP-A3576-A	• 修订“6.2.1 参数设定”的“(1) 可编程控制器 CPU 的参数设定”。 • 修改抓手输入输出类型参数 (HIOTYPE) 的出厂时设定值。
2018-01-15	BFP-A3576-B	• 修改 PrmWrite 指令的例文。 • 补充指令语句的说明。(If...Then...ElseIf...Else...EndIf、Select Case) • 在多功能电动抓手状态变量 M_EHCur 的说明中追加 5)。修改例文。
2018-01-31	BFP-A3576-C	• 追加与不正确访问对策相关的记述。 • 删除连续功能的说明。 • 追加“7.8 日志功能”。
2018-03-01	BFP-A3576-D	• 追加 CR800-Q 控制器的说明。
2018-09-01	BFP-A3576-E	• 补充弹性伺服模式的说明。(Cmp Jnt、Cmp Pos、Cmp Tool、Cmp Off)
2018-10-15	BFP-A3576-F	• 追加样条插补指令速度高速化功能的说明。
2018-12-25	BFP-A3576-G	• 追加 ACIN 电缆连接的补充说明。
2019-04-19	BFP-A3576-H	• 修订软元件分配的说明。 • 修改错误记述。(错误: MELFA-BASIC V)
2020-01-24	BFP-A3576-J	• 追加参数。(TOOLSPEC、RCDUVER) • 修改说明内容。(Base、Tool、Def Work、M_Tool)
2020-10-30	BFP-A3576-K	• 变更非法访问对策的相关注意事项。 • 修改 #Include 语句的例文。(图 4-6) • 修订“3.11 无法解除错误的暂时错误复位操作”。 • 在模型设定参数中追加 CR 系列的值以及修改错误记述。 • 修改 Cmp Off 指令的例文。 • 追加“2.4 机器人 CPU 的 LED 显示”。 • 修改不统一的退避表示。 • 其他的错误记述修改及部分更改。
2021-02-19	BFP-A3576-M	• 补充指令语句的说明。(Def Act、Function...FEnd) • 追加 FUNCSPEC 参数。
2021-04-01	BFP-A3576-N	• 追加“5.28 关于安全功能”。 • 追加机器人(系统)状态变量 M_AmpInfoA。 • 追加参数。(NETIPFLT、NETIPFLS、NETIPFLE) • 在“1.1.1 各操作说明书的内容”中追加安全通信功能使用说明书。
2021-09-30	BFP-A3576-P	• 修订“4.4.2 关于基本转换”。 • 修改 GetM 命令的错误记述。 • 其他的错误记述修改及部分更改。
2022-01-31	BFP-A3576-R	• 其他的错误记述修改及部分更改。
2022-02-24	BFP-A3576-S	• 支持软件版本 C2d。 追加 HNDCHK 参数。
2023-04-17	BFP-A3576-T	• 支持软件版本 C2j。 在“4.12.2 各指令语句的说明”的指令语句“JRC”中追加说明。 在“5.1 动作参数”中追加参数。(UORGSPEC) • 其他的错误记述修改及部分更改。
2023-09-14	BFP-A3576-U	• 追加 CR860 控制器。 • 其他的错误记述修改及部分更改。
2024-03-08	BFP-A3576-V	• 修订“2.1 示教单元(T/B)的功能”。 • 修订“3.7 排错操作”。 • 其他的错误记述修改及部分更改。

■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（选件））的功能和操作方法的相关说明，及 MELFA-BASIC VI 程序语言的功能及规格的相关说明。

以下对本书中使用的系列与控制器的表述进行说明。

名称	说明
CR800 系列	是指 CR800 控制器和 CR860 控制器。
CR800-D 系列	是指 CR800-D 控制器和 CR860-D 控制器。
CR800-Q 系列	是指 CR800-Q 控制器和 CR860-Q 控制器。
CR800-R 系列	是指 CR800-R 控制器和 CR860-R 控制器。
CR800 控制器	是指 CR800-D 控制器、CR800-Q 控制器和 CR800-R 控制器。
CR860 控制器	是指 CR860-D 控制器、CR860-Q 控制器和 CR860-R 控制器。

适用于对应 iQ Platform 的 CR800-R/Q 系列和独立设备型的 CR800-D 系列。特别是个别附带的功能，记载为“仅限 CR800-R”或“仅限 CR800-D”、“仅限 CR860-R”及“仅限 CR860-D”。

此外，本手册中根据示教单元的按键操作与运行画面中的操作对机器人程序的启动、停止等的操作进行说明。关于通过外部信号（专用输入输出信号）进行的操作，在下一页中总结了与示教单元操作对应的专用输入输出信号，请通过参数设定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”。

- 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联系。
- 以太网、ETHERNET 是富士胶片商业创新株式会社的注册商标。Ethernet 是美国 Xerox Corporation 的注册商标。
- 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。
- 使用说明书的图片可能会与实物有所不同。

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。本手册中根据示教单元的按键操作与运行画面中的操作进行说明。请将各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。参数的详细内容请参照第 616 页的“6.3 专用输入输出”，各信号的时序图请参照第 627 页的“6.5 外部信号的时序图”，参数的设定方法请参照第 93 页的“3.15 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

示教单元操作	参数名称	分类	功能	出厂设置值
启动 RUN	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] 键 待机中	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] 键 ERROR 指示灯 程序复位	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。	
程序选择 [OVRD ↑] 键 [OVRD ↓] 键 步显示	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。	
CONT. 循环	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。	
[SERVO] 键 SERVO 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。	
[SERVO] 键 SERVO 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 示教单元 (T/B) 的功能	2-7
2.1.1 关于操作权	2-9
2.1.2 示教单元的使用	2-10
(1) 示教单元的安装	2-11
(2) 示教单元的拆卸	2-11
2.2 动作及控制相关的功能	2-12
2.3 机型复位操作的说明	2-14
2.4 机器人 CPU 的 LED 显示	2-15
2.4.1 R 型 CPU 模块 (R16RTCPU) 的显示	2-15
2.4.2 Q 型 CPU 模块 (Q172DSRCPU) 的显示	2-16
2.5 操作面板 (O/P) 的功能 (仅限 CR860-D/R/Q)	2-17
2.5.1 操作面板各部位的名称 (CR860 控制器)	2-17
2.5.2 STATUS NUMBER (显示面板) 的说明	2-18
(1) STATUS NUMBER (显示面板) 的切换	2-18
(2) 各种状态显示	2-19
(3) 机型复位操作的说明	2-20
3 操作方法的说明	3-21
3.1 示教单元的菜单画面操作	3-21
(1) 画面树形图	3-21
(2) 数字 / 文字的输入	3-25
(3) 菜单的选择方法	3-26
3.2 JOG 前进 (概略说明)	3-27
3.2.1 JOG 前进的种类	3-27
3.2.2 JOG 前进的速度	3-29
3.2.3 关节 JOG	3-30
3.2.4 直交 JOG	3-30
3.2.5 工具 JOG	3-31
3.2.6 三轴直交 JOG	3-31
3.2.7 圆筒 JOG	3-32
3.2.8 工件 JOG	3-32
3.2.9 工具转换数据的切换	3-33
3.2.10 世界坐标系的切换 (基本坐标编号的指定)	3-34
3.2.11 JOG 操作中的冲突检知	3-36
(1) JOG 操作时的碰撞检测标准调整	3-36
3.3 抓手开闭	3-37
3.4 退避点复归	3-38
3.5 抓手整列	3-39
3.6 程序编辑	3-40
3.6.1 程序的制作	3-40
(1) 程序编辑画面的开启	3-40
(2) 程序的制作	3-41
(3) 程序制作的结束、保存	3-43
(4) 程序的修正方法	3-44
(5) 当前位置的示教	3-46
(6) 位置变量的删除	3-49
(7) 位置数据的确认	3-50

目录

	页
(8) 位置数据的 MDI (Manual Data Input) 登录 (修正)	3-51
(9) 直接执行	3-52
3.7 排错操作	3-53
(1) 单步前进	3-53
(2) 单步返回	3-55
(3) 其他插槽中的单步前进	3-56
(4) 单步跳转	3-58
3.8 自动运行	3-59
3.8.1 动作速度的设定	3-59
(1) 示教单元中的操作	3-59
(2) 操作面板中的操作 (仅限 CR860-D/R/Q)	3-59
3.8.2 自动运行的开始	3-60
(1) 示教单元中的操作	3-60
(2) 操作面板中的操作 (仅限 CR860-D/R/Q)	3-64
3.8.3 停止	3-65
(1) 示教单元中的操作	3-65
(2) 操作面板中的操作 (仅限 CR860-D/R/Q)	3-65
3.8.4 停止之后重新开启自动运行	3-65
(1) 示教单元中的操作	3-65
(2) 操作面板中的操作 (仅限 CR860-D/R/Q)	3-65
3.8.5 程序复位	3-66
(1) 示教单元中的操作	3-66
(2) 操作面板中的操作 (仅限 CR860-D/R/Q)	3-66
3.9 伺服 ON/OFF	3-67
(1) 示教单元中的操作	3-67
(2) 操作面板中的操作 (仅限 CR860-D/R/Q)	3-67
3.10 错误复位操作	3-67
(1) 通过示教单元进行错误复位操作	3-67
(2) 通过操作面板进行错误复位操作 (仅限 CR860-D/R/Q)	3-67
3.11 无法解除错误的暂时错误复位操作	3-68
3.12 管理 • 编辑画面的操作	3-70
(1) 程序的一览显示	3-70
(2) 程序的复制 (复制)	3-71
(3) 程序的名称变更 (重命名)	3-72
(4) 程序的删除 (删除)	3-73
(5) 程序的保护 (保护)	3-74
(6) 程序的选择	3-75
3.13 运行画面的操作	3-76
3.13.1 执行行的显示	3-76
(1) 确认画面的选择	3-76
(2) 单步运行	3-76
(3) 单步跳转	3-78
(4) 其他插槽中的单步前进	3-79
(5) 确认画面的结束	3-79
3.13.2 测试运行画面	3-80
(1) 测试运行画面的选择	3-80
3.13.3 操作面板画面的操作	3-81
3.14 监视画面的操作	3-82
(1) 输入信号监视	3-82
(2) 输出信号监视	3-84
(3) 输入寄存器监视	3-86
(4) 输出寄存器监视	3-87
(5) 变量监视	3-90
(6) 错误履历	3-92
3.15 参数画面的操作	3-93
3.16 原点 • 制动闸画面的操作	3-95

目录

	页
(1) 原点	3-95
(2) 制动闸	3-95
3.17 设定・初始化画面的操作	3-97
(1) 程序的初始化	3-97
(2) 参数的初始化	3-98
(3) 电池的初始化	3-99
(4) 运行	3-99
(5) 时间设定	3-100
(6) 版本	3-100
3.18 扩展功能的操作	3-101
(1) PLC 直接	3-101
(2) 工件坐标	3-101
3.19 初始设定画面的操作	3-102
(1) 显示语言的设定	3-102
(2) 对比度的设定	3-104
4 MELFA-BASIC VI	4-106
4.1 MELFA-BASIC VI的功能	4-106
4.1.1 机器人的动作控制	4-107
(1) 关节插补动作	4-107
(2) 直线插补动作	4-108
(3) 圆弧插补动作	4-109
(4) 连续动作	4-111
(5) 加减速时间和速度控制	4-112
(6) 确认到达目的位置	4-114
(7) 高轨迹精度控制	4-115
(8) 抓手・工具控制	4-116
4.1.2 托盘运算	4-117
4.1.3 程序控制	4-121
(1) 无条件分支・条件分支・待机	4-121
(2) 循环	4-123
(3) 中断	4-124
(4) 子程序	4-125
(5) 定时器	4-126
(6) 停止	4-127
4.1.4 外部信号的输入输出	4-128
(1) 信号输入	4-128
(2) 信号输出	4-128
4.1.5 通信	4-129
4.1.6 公式和运算	4-130
(1) 运算符一览	4-130
(2) 位置数据的相对运算（乘算）	4-131
(3) 位置数据的相对运算（加算）	4-131
4.1.7 附随句	4-132
4.2 多任务功能	4-133
4.2.1 所谓多任务功能	4-133
4.2.2 多任务的执行方法	4-134
4.2.3 各任务插槽运行状态	4-134
4.2.4 多任务程序创建上的注意点	4-136
(1) 任务个数和处理时间的关系	4-136
(2) 指定并列执行程序个数的最大值	4-136
(3) 在外部变量的程序间的数据的传输方法	4-136
(4) 确认机器人（系统）状态变量下的程序运行状态	4-136
(5) 机器人动作的程序基本上在任务插槽 1 执行	4-136
(6) 在始终执行程序的初始化处理	4-136
4.2.5 多任务程序运行上的注意点	4-137
(1) 多任务的开始	4-137

目录

	页
(2) 运行状态的显示	4-137
4.2.6 多任务的使用例	4-138
(1) 机器人的作业内容	4-138
(2) 到多任务执行为止的步骤	4-139
4.2.7 关于程序容量	4-140
(1) 程序储存领域	4-140
(2) 程序编辑领域	4-140
(3) 程序执行领域	4-140
4.3 MELFA-BASIC VI的详细规格	4-141
(1) 程序名	4-141
(2) 指令语句	4-141
(3) 变量	4-141
4.3.1 语句	4-142
4.3.2 附随语句	4-142
4.3.3 行	4-142
4.3.4 步号	4-142
4.3.5 标签	4-142
4.3.6 可以在程序内使用的字符种类	4-143
4.3.7 有特别意义的字符	4-144
(1) 大写、小写的区别	4-144
(2) 下划线 (_)	4-144
(3) 撇号 (')	4-144
(4) 星号 (*)	4-144
(5) 逗号 (,)	4-144
(6) 句号 (.)	4-144
(7) 空格	4-144
4.3.8 数据类型	4-145
4.3.9 常数	4-145
4.3.10 数值常数	4-145
(1) 10 进制	4-145
(2) 16 进制	4-145
(3) 2 进制	4-145
(4) 常数的类型	4-145
4.3.11 字符串常数	4-145
4.3.12 位置常数 (直交 / 工件坐标常数)	4-146
(1) 坐标 · 姿势 · 附加轴数据的形式和意义	4-146
(2) 结构标签数据的形式和意义	4-146
4.3.13 关节常数	4-147
(1) 各轴数据的形式和意义	4-147
4.3.14 角度值	4-147
4.3.15 变量	4-148
4.3.16 数值变量	4-149
4.3.17 字符串变量	4-149
4.3.18 位置变量 (直交 / 工件坐标变量)	4-149
4.3.19 关节变量	4-150
4.3.20 输入输出变量	4-150
4.3.21 排列变量	4-150
4.3.22 外部变量	4-151
4.3.23 程序外部变量	4-151
4.3.24 用户定义外部变量	4-152
4.3.25 用户基本程序的创建	4-152
4.3.26 有效范围	4-153
4.3.27 Function 过程	4-153
4.3.28 #Include 陈述句	4-154
4.4 机器人的坐标系说明	4-155
4.4.1 关于机器人的坐标系	4-155
4.4.2 关于基本转换	4-156
4.4.3 关于位置数据	4-157

目录

	页
4.4.4 关于工具坐标系（机械接口坐标系）	4-158
(1) 机械接口坐标系	4-158
(2) 工具坐标系	4-159
(3) 工具坐标系的使用效果	4-160
4.5 机器人（系统）状态变量	4-162
4.5.1 逻辑数	4-165
4.6 函数	4-165
(1) 用户定义函数	4-165
(2) 内建函数	4-165
4.7 指令语句一览	4-167
(1) 动作控制相关的指令	4-167
(2) 程序控制相关的指令	4-168
(3) 定义指令	4-169
(4) 多任务相关	4-170
(5) 通信	4-170
(6) 其他	4-170
4.8 运算符	4-171
4.9 运算的优先级	4-172
4.10 程序的控制构造深度	4-172
4.11 保留字	4-172
4.12 指令的详细说明	4-173
4.12.1 记载项目的阅读方法	4-173
4.12.2 各指令语句的说明	4-173
4.13 机器人（系统）状态变量的详细说明	4-333
4.13.1 记载项目的说明	4-333
4.13.2 各机器人状态变量（系统状态变量）的说明	4-333
4.14 函数的详细说明	4-425
4.14.1 记载项目的说明	4-425
4.14.2 各函数的说明	4-425
5 参数的设定功能	5-468
5.1 动作参数	5-468
5.2 信号参数	5-482
5.2.1 关于多 CPU 输入偏置（仅限 CR800-R/Q 系列）	5-487
(1) 案例（A）	5-487
(2) 案例（B）	5-488
5.3 操作参数	5-490
5.4 程序参数	5-493
5.5 通信参数	5-497
5.6 关于标准 TOOL 坐标	5-500
5.7 关于标准基本坐标	5-502
5.8 关于使用者定义领域	5-503
5.8.1 坐标系的选择	5-504
5.8.2 领域的定义	5-505
5.8.3 对象机器的选择	5-507
5.8.4 指定使用者定义领域内的举动	5-507
5.8.5 设定示例	5-507
5.9 自由平面限制	5-508
5.9.1 自由平面极限的定义	5-508
5.9.2 自由平面极限的坐标系选择	5-509
5.10 中断时的 JOG 前进后的自动回归设定	5-510
5.11 电源开启的程序自动执行方法	5-511
5.12 关于抓手种类	5-512
(1) 电磁阀的种类和信号号码	5-512

目录

	页
5.13 关于抓手初始状态	5-513
5.14 关于输出信号复位模式	5-514
5.15 关于通信设定 (以太网)	5-516
5.15.1 各参数的详细内容	5-516
(1) NETIP (控制器的 IP 地址)	5-516
(2) NETMSK (子网掩码)	5-516
(3) NETPORT (端口号码)	5-516
(4) CPRCE11 ~ 19 (协议)	5-517
(5) COMDEV (对应 COM1 : ~ 8 的设备定义)	5-517
(6) NETMODE (服务器指定)	5-517
(7) NETHSTIP (数据通信对象的服务器的 IP 地址)	5-517
(8) MXTTOUT (执行 realtime 外部控制指令时的超时设定)	5-517
5.15.2 参数设定示例 1 (使用支持软件时)	5-518
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)	5-519
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)	5-520
5.15.5 参数设定示例 3 (使用实时外部控制功能)	5-521
5.15.6 连接的确认	5-522
5.15.7 通过 Windows 的 ping 指令确认连接的方法	5-522
5.16 关于抓手、工件条件设定 (最佳加减速设定)	5-523
5.17 关于特异点接近警告	5-525
5.18 关于高速 RAM 运行功能	5-526
5.19 关于暖机运行模式	5-527
5.20 关于特异点通过功能	5-533
5.21 关于冲突检知功能	5-538
(1) 功能的概要	5-538
(2) 关连参数	5-539
(3) 使用冲突检知功能时	5-540
5.22 关于过负载 Level 的最佳化	5-545
5.23 Pallet 定义指令的多旋转限制	5-546
5.24 关于干涉回避功能	5-547
5.24.1 操作步骤	5-548
5.24.2 机器的准备和连接	5-549
5.24.3 干涉确认用模型的登录	5-551
5.24.4 自由平面极限的登录	5-559
5.24.5 附加轴的对应	5-560
5.24.6 CPU 缓冲存储器扩展功能的设定 (仅限机器人之间的干涉确认时)	5-561
5.24.7 机器人之间的校准 (仅限机器人之间的干涉确认时)	5-564
5.24.8 干涉回避功能的有效 / 无效设定	5-566
5.24.9 干涉回避功能的使用	5-567
5.24.10 采样程序	5-568
5.25 可编程控制器输入输出模块直接控制	5-570
5.25.1 CR800-R 系列	5-570
(1) 作业流程	5-570
(2) 硬件的构成示例	5-570
(3) 设定系统构成示例 1 的参数	5-571
(4) 设定系统构成示例 2 的参数	5-577
(5) 使用状态变量进行控制	5-582
5.25.2 CR800-Q 系列	5-584
(1) 作业流程	5-584
(2) 硬件的构成示例	5-584
(3) 设定系统构成示例 1 的参数	5-585
(4) 设定系统构成示例 2 的参数	5-591
(5) 关于机器人参数的详细内容	5-598
(6) 使用状态变量进行控制	5-599
5.26 在机器人 CPU 之间直接通信	5-601
(1) 规格	5-601

目录

	页
(2) 机器人状态变量的说明	5-601
5.27 关于冗余系统错误的动作选择参数	5-602
5.28 关于安全功能	5-603
(1) IP 地址过滤功能	5-603
6 外部输入输出的功能	6-604
6.1 种类	6-604
6.2 PLC 连接输入输出功能	6-605
6.2.1 参数设定	6-605
(1) PLC CPU 的参数设定	6-605
(2) 机器人 CPU 的参数设定	6-609
6.2.2 CPU 缓冲存储器与机器人输入输出信号的对应	6-612
6.2.3 PLC 梯形图的范例	6-612
6.2.4 专用输入输出信号的分配 (工厂出货时设定)	6-614
6.3 专用输入输出	6-616
6.4 信号的有效无效状态	6-626
6.5 外部信号的时序图	6-627
6.5.1 各信号的个别时序图	6-627
6.5.2 时序图例	6-636
(1) 外部信号操作时序图 (其1)	6-636
(2) 外部信号操作时序图 (其2)	6-637
(3) 外部信号操作时序图 (其3)	6-638
(4) 外部信号操作时序图 (其4)	6-639
(5) 外部信号操作时序图 (其5)	6-640
6.6 通过外部信号选择 • 执行程序的方法	6-641
6.6.1 种类	6-641
6.6.2 执行方法的选择	6-641
6.6.3 相关的输入输出参数	6-641
6.6.4 操作步骤	6-642
6.7 紧急停止输入	6-644
6.7.1 紧急停止输入时的机器人的动作	6-644
6.8 软元件	6-645
6.8.1 软元件一览	6-645
6.8.2 软元件分配	6-646
6.8.3 可编程控制器软元件分配功能	6-649
7 附录	附录 -653
7.1 关于构造标志	附录 -653
7.2 关于样条插补	附录 -655
7.2.1 概要	附录 -655
(1) 概要	附录 -655
(2) 特征	附录 -655
(3) 必要的设备与软件版本	附录 -656
(4) 术语	附录 -656
7.2.2 规格	附录 -657
(1) 基本规格	附录 -657
(2) 限制事项	附录 -658
(3) 样条插补中的机器人的动作	附录 -658
(4) 路径点相关的检查	附录 -660
7.2.3 功能的说明	附录 -661
(1) 路径补偿	附录 -661
(2) 动作模式	附录 -664
(3) 信号输出	附录 -665
(4) 数值设定	附录 -666
(5) 帧转换	附录 -666
7.2.4 作业步骤	附录 -668

7.2.5 样条文件的创建	附录 -670
(1) 新建	附录 -670
(2) 通过 DXF 文件导入功能创建	附录 -671
(3) 用机器人语言新建	附录 -678
(4) 打开现有的样条文件	附录 -678
(5) 样条文件编辑画面的说明	附录 -680
(6) 样条文件编辑时的功能区	附录 -682
(7) 路径点数据的内容	附录 -685
(8) 样条文件的编辑	附录 -686
(9) 样条文件的保存	附录 -692
(10) 样条文件的删除	附录 -694
(11) 样条号的更改	附录 -694
(12) 样条文件的复制	附录 -695
(13) 样条文件管理	附录 -696
(14) 导入 · 导出功能	附录 -697
(15) 编辑的辅助功能	附录 -706
(16) 样条曲线的显示	附录 -708
(17) 编辑内容的确认	附录 -708
(18) 机器人程序	附录 -709
7.2.6 机器人程序的创建	附录 -712
7.2.7 动作的确认	附录 -715
7.2.8 保存至机器人控制器	附录 -715
7.2.9 调整作业	附录 -715
(1) 位置补偿功能	附录 -715
(2) 帧转换功能	附录 -717
(3) 跳位	附录 -720
(4) 参数 SPLOPTGC	附录 -721
7.2.10 样条插补指令速度高速化	附录 -722
(1) 概要	附录 -722
(2) 必要的软件版本	附录 -722
(3) 规格	附录 -722
(4) RT ToolBox3 的设定	附录 -723
(5) 注意事项 · 制约事项	附录 -723
7.3 关于 Ex-T 控制	附录 -724
7.3.1 概要	附录 -724
(1) 特征	附录 -724
(2) 规格	附录 -725
(3) 必要的设备与软件版本	附录 -725
7.3.2 Ex-T 坐标的设定	附录 -725
(1) 设定方法	附录 -725
7.3.3 Ex-T JOG	附录 -726
(1) 工件 JOG 中的姿势成分动作	附录 -727
(2) Ex-T JOG 中的姿势成分动作	附录 -727
(3) Ex-T JOG 的操作	附录 -728
(4) RV6 轴型的工件 JOG 动作	附录 -729
(5) RH4 轴型的工件 JOG 动作	附录 -731
(6) RH4 轴垂吊型的工件 JOG 动作	附录 -733
7.3.4 机器人程序的创建	附录 -735
(1) Ex-T 控制相关指令 · 变量的一览	附录 -735
(2) 编程示例	附录 -735
7.4 关于高速位置获取功能	附录 -737
7.4.1 概要	附录 -737
7.4.2 GPS 功能规格	附录 -737
7.4.3 数字输入信号规格	附录 -738
7.4.4 数字输入电气的规格	附录 -739
7.4.5 GPS 功能的使用方法	附录 -740
(1) 获取传感器输入时的位置数据	附录 -740
(2) 判断盒内有无工件	附录 -741

目录

	页
7.5 伺服软件升级	附录 -743
7.6 日志功能	附录 -744
7.6.1 概要	附录 -744
7.6.2 规格	附录 -745
(1) 基本规格	附录 -745
(2) 日志文件规格	附录 -745
7.6.3 日志保存对象的错误编号	附录 -746
7.6.4 参数的设定	附录 -746
7.6.5 日志文件内容的确认	附录 -747
(1) 错误履历	附录 -747
(2) 事件履历	附录 -748
(3) 程序执行履历	附录 -750
7.7 特殊设备	附录 -751
7.7.1 CR800-R 系列	附录 -751
7.7.2 CR800-D 系列	附录 -752
7.7.3 CR800-Q 系列	附录 -753

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 为基本记载。使用 R56TB 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。

手册名称	内容
安全手册	为了确保机器人相关的所有作业人员的安全，对机器人的使用、系统设计及制作的通用注意事项及安全措施进行了说明。
标准规格书	对产品的标准规格及出厂特殊规格、选件构成、维护部件等有关内容进行了说明。此外，还介绍了导入机器人之前的安全方面、技术方面的注意事项有关内容。
从机器人本体安装到维护	对机器人本体相关的投运之前的步骤（开箱、搬运、安装、动作确认）及其维护、点检有关内容进行了说明。
从控制器安装及基本操作到维护	对控制器相关的操作前的准备步骤（开箱、搬运、安装、动作确认）及从程序创建到自动运行的基本操作及维护、点检有关内容进行了说明。
功能及操作的详细说明	对各功能的说明及操作方法、程序中使用的 MELFA-BASIC VI 指令的说明、与外部输入输出设备的连接方法、参数的说明等功能・操作的详细内容进行了介绍。
故障排除	对发生错误时该错误编号对应的原因及措施进行了说明。
附加轴功能	对与控制器组合使用的通用伺服放大器控制功能进行了说明。
跟踪功能	对传送带跟踪的规格、功能、使用方法进行了说明。
GOT 扩展功能	对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、操作步骤进行了详细说明。
iQ Platform 对应扩展功能	对 iQ Platform 对应的机器人中的可编程控制器与机器人之间的存储器的数据构成及监视、操作步骤进行了详细说明。
安全通信功能	对通过与安全可编程控制器进行安全通信来扩展机器人的安全功能的安全通信功能进行了说明。请参照英文版使用说明书。
以太网功能	对可使用 TCP/IP 协议与以太网上的计算机通信的方法进行了说明。

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 对应类型	
	控制器	是对机器人本体进行控制的控制器。
	机器人 CPU 单元 或 机器人 CPU	表示安装在本公司 MELSEC iQ-R/MELSEC-Q 系列的可编程控制器基板模块中的机器人用 CPU 模块。使用专用电缆与控制器进行连接。
	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。
	独立设备型	
	控制器	表示对机器人本体进行控制的控制器。
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]	用 [] 围住的内容表示示教单元的按键。
	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 R32TB 为基础说明。

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** 包含抓手在内的工件质量应不超过额定负载及允许转矩。超过时会导致发生错误或故障等。
-  **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的，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** 通过 CR800-R/Q 系列使用 iQ Platform 对应产品时，请勿将手持式 GOT 连接至可编程控制器。无论操作权有效 / 无效，手持式 GOT 都可以使机器人自动运行，可能导致设备损坏及人身事故。
-  **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，接线完毕后，务必对示教单元紧急停止、用户紧急停止、门开关等的各种功能能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（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。

 注意

为了保证机器人及系统的网络安全（可用性、完整性、机密性），对于来自不可信网络或经由网络的设备的非法访问、拒绝服务攻击（DoS^{*1} 攻击）以及计算机病毒等其他网络攻击，应采取设置防火墙与虚拟专用网络（VPN），以及在计算机上安装杀毒软件等对策。

因非法访问、拒绝服务攻击（DoS 攻击）、计算机病毒以及其他网络攻击引发的机器人及系统方面的各种问题，三菱电机不承担责任。

*1 DoS：耗费目标计算机的资源或使其安全性变得脆弱，导致其无法提供正常服务，以及此种状态。

2 功能的说明

2.1 示教单元 (T/B) 的功能

本章记述 R32TB (配件) 的功能。



图 2-1: 示教单元外观图

<1> [EMG. STOP] 开关

附带上锁功能的紧急停止用按钮式开关。按下此开关，无论示教单元在有效 / 无效状态，机器人都会伺服 OFF 并立刻停止。

解除紧急停止状态时，应将开关向右旋转，或向自己的方向拉起。（按下紧急停止开关后，机器人将变为错误状态。解除开关后，应进行错误的复位操作。）

<2> [TB ENABLE] 开关

通过示教单元将机器人的操作设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。此外，前面的 ENABLE 灯也会亮起。使用示教单元操作机器人时，务必将示教单元设为有效。

将示教单元设为有效后，示教单元的操作有优先权，在示教单元中可进行操作的同时，将无法从外部进行操作（MANUAL 模式）。从外部进行操作时，应将示教单元设为无效状态（AUTOMATIC 模式）。

<3> 有效开关（3 位置开关）

在背面的 3 位置的开关。MANUAL 模式时，放开或强力按下此开关后伺服 OFF。JOG 操作及单步执行等在伺服 ON 状态下操作的功能，应在轻按此开关的情况下进行。

在 CR800 系列控制器的 S/W 版本为 Ver. D1 以上的情况下，进行单步执行时，无论机器人的动作 / 非动作指令如何，都应在轻按此开关的状态下进行操作。

另外，在进行紧急停止及伺服 OFF 操作，且伺服为 OFF 的情况下，只按下本开关也不会变为伺服 ON。应重新进行伺服 ON 操作。

<4> 显示面板

通过示教单元按键操作，显示程序内容及机器人状态。

<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 ON 时，绿色灯亮起。

[ERROR]：机器人在错误状态时，红色灯亮起。

- <6>[F1][F2][F3][F4] 键
执行显示面板的功能显示部中显示的功能。
- <7>[FUNCTION] 键
在 1 个操作中，[F1][F2][F3][F4] 键中存在 5 个以上分配的功能时，按下此键对功能显示进行切换，并变更 [F1][F2][F3][F4] 键中分配的功能。
在画面的最下段有用反白字显示的菜单。这些菜单以左到右的顺序分配到 [F1][F2][F3][F4] 键。通过按下对应的功能键可选择显示的菜单。
此外，菜单的右端显示“=>”时，表示除当前显示外还有其他菜单，每次按下 [FUNCTION] 键，即可切换显示菜单。
- <8>[STOP] 键
如果机器人处于动作中，立即减速，停止机器人的动作。此外，如果有程序在执行中，会中断执行。如果示教单元处于连接状态，即使没有按下 [ENABLE] 开关时（[ENABLE] 灯没有亮起时）也可使用。
- <9>[OVRD ↑][OVRD ↓] 键
改变机器人的速度倍率修调。按下 [OVRD ↑] 键时倍率修调将增加。按下 [OVRD ↓] 键时倍率修调将减少。
- <10>[JOG 操作] 键（[-X (J1)] ~ [+C (J6)] 的 12 个键）
示教单元为 JOG 模式时，通过此键进行 JOG 操作。
此外，示教单元为抓手模式时，通过此键进行抓手操作。
- <11>[SERVO] 键
轻轻握住（3 位置开关）的同时按下此键，机器人将会伺服 ON。
- <12>[MONITOR] 键
按下此键将变为监视模式，显示监视菜单。监视模式时按下此键，将返回监视模式前的画面。
- <13>[JOG] 键
按下此键，将变为 JOG 模式，显示 JOG 画面。JOG 模式时按下此键，将返回 JOG 模式前的画面。
- <14>[HAHD] 键
按下此键，将变为抓手操作模式，显示抓手操作画面。抓手操作模式时按下此键，将返回抓手操作模式前的画面。
此外，持续按下此键 2 秒以上，将变为工具选择画面，变为进行工具转换数据选择的模式。工具选择模式时持续按下此键 2 秒以上，将返回前一个画面。
- <15>[CHARACTER] 键
示教单元可进行文字输入或数字输入时，使用 [数字 / 文字] 键的功能切换数字与文字间的输入。
- <16>[RESET] 键
机器人为错误状态时，解除错误。（也有无法解除的错误）
此外，按下此键的同时按下 [EXE] 键，将进行程序复位。
- <17>[↑][↓][←][→] 键
将光标向各个方向移动。
- <18>[CLEAR] 键
可进行数字输入或文字输入时，按下此键，将删除光标所在位置的 1 个文字。
此外，通过长时间按住，删除光标所在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，持续按下此键期间，机器人将动作。
- <20>[数字 / 文字] 键
可进行数字输入或文字输入时，按下此键将显示数字或文字。

2.1.1 关于操作权

控制器即使连接了示教单元及计算机等多个设备，同时在控制器上的操作（运行、伺服 ON 等的指令），有效的设备也仅限 1 个。此限定的 1 个设备就称为“取得操作权”。程序启动及伺服 ON 等使机器人启动、或可引起其启动的操作，没有操作权则无法执行。但反之，停止及伺服 OFF 等使机器人停止的操作，出于安全考虑，即使没有操作权也可执行。

表 2-1：设定开关与操作权的关系 ○：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 (模式选择开关)	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> 画面期间，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，将控制器的模式设为“AUTOMATIC”后，将发生错误 5000。
 注 3) 根据操作权输入信号：IOENA 的设定有效设备将发生变化。

表 2-2：需要有操作权的操作 操作权：○ = 需要操作权、× = 不需要操作权

区分	操作权	操作
运行 注1)	○	伺服 ON
	×	伺服 OFF
	○	程序启动。除了控制器操作面板，通过示教单元的操作也可启动。
	×	程序停止 / 循环停止
	○	程序复位
	×	错误复位
	○	倍率修调变更。但是使用示教单元可以随时变更。
	×	倍率修调读取
	○	程序编号变更
输入输出信号 注2)	×	程序编号 / 行编号读取
	×	输入输出信号读取
	×	输出信号写入
	○	专用输入 - 启动 / 复位 / 伺服 ON / 刹车 / 手动模式切换 / 通用输出复位 / 程序编号指定 / 行编号指定 / 倍率修调指定
	×	专用输入 - 停止 / 伺服 OFF / 连续 · 循环 / 操作权输入信号 / 程序编号输出要求 / 行编号输出要求 / 倍率修调输出要求 / 错误编号要求 / 数值输入
程序编辑 注3)	×	抓手输入输出信号读取
	○	抓手输出信号写入
	○	行登录 / 读取 / 调用、位置追加 / 补偿 / 读取、变量写入 / 读取
	○	单步前进 / 返回、执行
文件操作	○	单步 UP/DOWN
	○	单步跳过、直接执行、JOG
维修操作	×	程序一览读取 / 保护设定 / 复制 / 删除 / 名称变更 / 初始化
	×	参数读取、时钟设定 / 读取、运行时间读取、报警履历读取
	○	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”：即使不按下示教单元的 [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> 画面期间，通过 [MONITOR] 键进行的操作无效。
 通过示教单元显示 <操作面板> 画面期间，通过 [MONITOR] 键进行的操作无效。

注 3) 一台设备在线编辑中时，无法通过其他设备进行编辑。

2.1.2 示教单元的使用

示教单元的拆装在将机器人控制器的控制电源设为 OFF 的状态下进行。如果在控制电源 ON 的状态下进行示教单元的拆装，将发生紧急停止报警。

此外，AUTOMATIC 模式时，在轻轻抓握示教单元的 3 位置有效开关的状态（位置 2 的状态）下放开后的 5 秒以内拔出示教单元连接器，可在不发生紧急停止报警的情况下从控制器上卸除示教单元。注 1)

注 1) 使用 CR860-D/R/Q 时，根据外部紧急停止按钮的接线方法，在不发生紧急停止报警的情况下可能无法拆除示教单元。请确认另一手册“CR860 Controller RV-35FR/50FR/80FR Series Standard Specifications Manual (BFP-A3779)”的安全措施示例的接线示例。



注意

强行拉扯或过度弯曲示教单元的电缆，有可能导致电缆断线或连接器破损，应加以注意。

进行拆装时，应握住连接器本身进行操作，请勿对电缆施加压力。

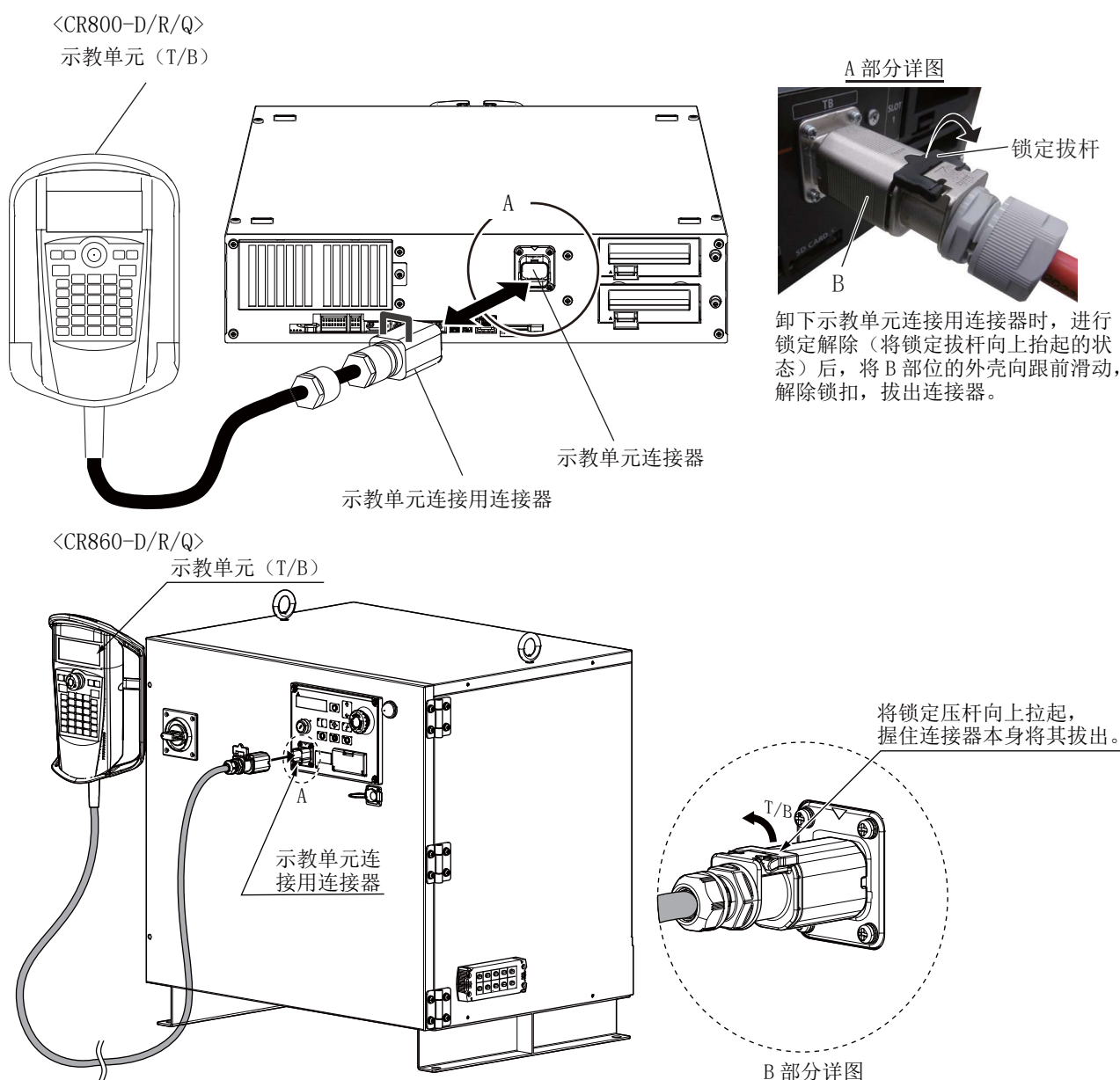


图 2-2: 示教单元的拆装

(1) 示教单元的安装

以下对示教单元的安装方法进行说明。

- 1) 确认控制器的电源处于 OFF 状态。
- 2) 图 2-2 参照，将示教单元的连接器连接到控制器的 TB 连接器。
将锁定压杆向上抬起，压入连接器直至发出咔嚓声。

至此，示教单元的安装结束。

(2) 示教单元的拆卸

以下对示教单元的拆卸方法进行说明。

■ MANUAL 模式时

- 1) 确认控制器的电源处于 OFF 状态。
- 2) 图 2-2 参照，将连接器上部的锁定压杆向上拉起，握住连接器本身将其拔出。

至此，示教单元的拆卸结束。

■ AUTOMATIC 模式时

- 1) 处于轻轻抓握示教单元的 3 位置有效开关的状态（位置 2 的状态）。
- 2) 1) 应在 5 秒以内拔出示教单元连接器。图 2-2 参照，将连接器上部的锁定压杆向上拉起，握住连接器本身将其拔出。

至此，示教单元的拆卸结束。

2.2 动作及控制相关的功能

在本控制器中有如表 2-3 所示特征功能。

表 2-3: 有特征的功能

功能	说明	相关页次
最佳速度控制	本功能可在机器人在 2 点间移动时, 根据机器人的姿势必须控制速度的情况下在轨迹途中控制速度, 防止因速度过快发生错误。但是当此功能为有效时, 机器人的前端速度不固定。	请参照第 312 页的“Spd (Speed)”。
最佳加减速控制	对于机器人开始动作及停止时的加减速时间, 根据抓手的重量及重心、设定工件的有无, 自动设定最佳的加减速时间进行动作的功能。与提高节拍时间相关连。根据条件也有可能使节拍时间降低。	请参照第 289 页的“Oad1 (Optimal Acceleration)”、第 260 页的“LoadSet (Load Set)”。
直交伺服弹性	依据来自伺服的反馈数据, 灵活控制机器人的功能。对工件的嵌合有效的功能。可沿着机器人直交坐标系学习。但是根据工件的条件, 也有可能无法使用。	请参照第 190 页的“Cmp Tool”等。
冲突检知	机器人的工具及机械臂干涉了周边设备时, 会紧急停止并将工具及机械臂的损害降到最低。在自动运行及 JOG 操作时也可将功能设为有效。(部分特殊规格除外) 注) 请注意本功能无法同时与多机器人控制使用。	请参照第 196 页的“ColChk (Col Check)”、第 468 页的“动作参数”的 COL。
干涉回避功能	本功能是在 CR800-R/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 操作及自动运行时, 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 547 页的“5.24 关于干涉回避功能”。
维修预报	以机器人的动作状态为基准, 预测机器人的电池、皮带、润滑油的维修信息。此功能在计算机支持软件配件中可确认维修信息。(部分特殊规格除外)	请使用计算机支持软件配件。
支持位置回归	发生关节轴的偏移、更换电机、抓手变形、机器人的底座偏移时, 最多仅可补偿 10 几点, 计算原点数据及工具、底座的补偿值并对位置偏移进行补偿的功能。此功能在计算机支持软件配件中进行。可在垂直多关节机器人中使用。(部分特殊规格除外)	请使用计算机支持软件配件。
连续轨迹控制	可在多个位置连续抑制加减速, 使机器人动作的功能。是有效提升节拍时间的办法。	请参照第 111 页的“(4) 连续动作”、第 194 页的“Cnt (Continuous)”。
多任务程序运行	可将机器人的动作程序、与外部设备通信的程序等分割并列执行。对缩短输入输出处理有效。此外, 根据周边夹具的控制用程序的制作, 可构筑无可编程控制器的系统。	请参照第 133 页的“4.2.1 所谓多任务功能”、第 328 页的“XRun (x Run)”等的 X*** 指令。
程序的始终执行功能	从控制器的电源 ON 时开始, 始终执行程序。此功能利用多任务功能, 在机器人程序中代替可编程控制器执行时有效。	请参照第 493 页的“5.4 程序参数”的“SLTn”参数的启动属性 (ALWAYS)。
附加轴控制	作为机器人的附加轴最多可控制 2 轴。还会记忆机器人示教数据中附加轴的位置, 因此可以完全同步控制。此外可边移动附加轴 (行走轴), 边执行圆弧插补。	请参照另一手册“附加轴功能使用说明书”。
多机器人控制	标准机器人以外, 最多可控制 2 台 (标准机器人除外) 根据伺服电机所组成的机器人 (用户机械)。	请参照另一手册“附加轴功能使用说明书”。
外部设备通信功能	与外部设备的通信, 有以下方法。 控制器的控制用及程序内的互锁功能 1) 通过输入输出信号的方法 (CR800-R/Q: 可编程控制器链接输入输出: 8192/8192 点) (CR800-D: 并行输入输出: 256/256 点 (最多)) 2) 与外部设备的数据链接功能 (※ 仅限 CR800-D) 3) 通过以太网的通信 参考: 所谓数据链接, 是指与外部设备 (视觉传感器等) 收发补偿量等数据的功能)	请参照、第 368 页的“M_In/M_Inb/M_In8/M_Inw/M_In16/M_In32” 变量请参照第 379 页的“M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32”。
中断监视	在程序动作中监视信号等, 当条件一致时, 可中断当前的处理, 执行中断处理。 对工件搬运时的工件掉落监视等有效。	请参照第 202 页的“Def Act”、第 175 页的“Act”。

功能	说明	相关页次
程序间跳转	通过 CallP 指令可从程序里调用程序。	请参照第 180 页的“CallP”。
托盘运算	可以根据运算，算出配置在格子状里的工件的位置及盒内的玻璃基板的位置。有助于示教量的轻减。可以是列 × 行形式、1 列形式及圆弧形式。	请参照第 117 页的“4.1.2 托盘运算”、第 216 页的“Def P1t (Define 码垛)”、第 296 页的“P1t”。
使用者定义区域	最多可指定 32 个区域的任意空间，在其领域内实时监视机器人的前端位置是否进入，可将该状态输出到外部，通过程序示教或报错。此外，在机器人程序内也存在有相同功能的函数（Zone、Zone2、Zone3）。	请参照第 503 页的“5.8 关于使用者定义领域”、第 393 页的“M_Uar”，第 394 页的“M_Uar32” 变量。 请参照第 465 页的“Zone”、第 466 页的“Zone2”、第 466 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：可通过机器人的直交坐标系限制动作范围。 自由平面限制：定义任意的平面，可根据平面的近处或远处限制动作范围。	请参照第 468 页的“5.1 动作参数”的“MEJAR”、“MEPAR”参数。 请参照第 508 页的“5.9 自由平面限制”。

2.3 机型复位操作的说明

由于电池的消耗等导致机型信息消失时，将发生错误（H1600：生产厂商未设定 等），无法进行操作。此时，进行以下的机型复位操作时，将变为出厂状态，可设定正确的机型信息。

可通过示教单元或 RT ToolBox3 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 中设定 CLEAR。

<参数>	参数名 (MECHRST)
数据	要素 ()
()	
数据	上一个 123 下一个 关闭

名称的输入 [M]、[E]、[C]、[H]、[R]、[S]、[T]
确定 [EXE]



<参数>	参数名 (MECHRST)
数据	要素 ()
()	
数据	上一个 123 下一个 关闭

显示数据中当前的设定。

<参数>	参数名 (MECHRST)
数据	要素 ()
()	
数据	上一个 123 下一个 关闭

数据的设定 [F1]



<参数>	(MECHRST) ()
CLEAR	
	123 关闭

数据的删除 [CLEAR]
数据的输入 [C]、[L]、[E]、[A]、[R]
确定 [EXE]

- 2) 重新接通控制器的电源。

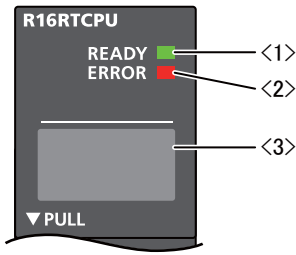


注意

由于进行本操作时将变为出厂状态，参数、程序与各种日志数据将被消除，因此请勿在发生错误（H1600）以外时实施。被消除的数据若存在 RT ToolBox3（选配件）的批量备份资料，则可通过恢复功能还原。

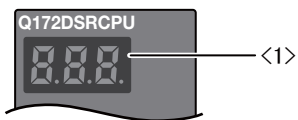
2.4 机器人CPU的LED显示



2.4.1 R型CPU模块（R16RTCPU）的显示



No.	名称	状态	说明
<1>	READY LED (绿色)	熄灯	电源OFF、电源ON后立刻在进行初始化 (约30秒) 或硬件异常
		闪烁	初始化中 (系统软件启动中)
		亮灯	正常动作中 (系统软件启动完成)
<2>	ERROR LED (红色)	熄灯	无错误或硬件异常
		闪烁或亮灯	当前发生了错误
<3>	点阵显示	H-1	电源ON或刚刚复位 经过数秒后依然显示当前内容时, 可能是发生了故障。
		00~99	硬件引导、OS启动中
		i19~i00	系统软件启动中
		E00~E60	系统异常 重新接通电源时无法恢复的情况下, 可能是发生了故障。
		Rdy	运行准备完成
		Run	程序运行中
		Stp	程序停止中
		AL H** ** L** ** C** **	当前发生了错误 在“AL”闪烁显示3次后, 4位的错误编号将分2次闪烁显示。第1次是H/L/C (错误等级) 与错误编号的高位2位, 第2次是错误编号的低位2位。

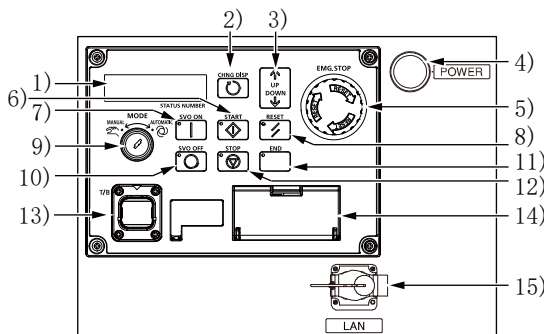
2.4.2 Q型CPU模块（Q172DSRCPU）的显示



No.	名称	状态	说明
<1>	7段LED显示	b00~o99	电源ON或刚刚复位、硬件引导、OS初始化中（约10秒） 经过1分钟后依然显示当前内容时，可能是发生了故障。
		qrw、qin、rin	可编程控制器CPU与机器人CPU之间的初始化处理中 经过1分钟后依然显示当前内容时，可能是发生了故障。
		i24~i00 8.8.8.	系统软件启动中 经过1分钟后依然显示当前内容时，可能是发生了故障。
		Rdy	运行准备完成
		Run	程序运行中
		Stp	程序停止中
		AL H** ** L** ** C** **	当前发生了错误 在“AL”闪烁显示3次后，4位的错误编号将分2次闪烁显示。第1次是H/L/C（错误等级）与错误编号的高位2位，第2次是错误编号的低位2位。
			最低位的点闪烁 正常动作中
	3个点保持亮灯状态 发生看门狗错误。硬件异常或软件异常。 即使重新接通电源也仍然发生该错误的情况下，可能是发生了故障。		

2.5 操作面板 (O/P) 的功能 (仅限CR860-D/R/Q)

2.5.1 操作面板各部位的名称 (CR860控制器)



- 1) 显示面板 (STATUS. NUMBER)
显示报警号码、程序名称、倍率修调值 (%) 等。电源接通后, 显示面板需要大概一分钟才会亮灯。
- 2) 显示切换按钮 (CHNG DISP)
按“倍率修调”→“行号码”→“程序名称”→“用户情报”→“生产厂商情报”的顺序切换显示面板的显示内容。
- 3) UP/DOWN 按钮..... 显示面板的显示内容进行上翻页、下翻页显示。
- 4) POWER 指示灯..... 表示主电源 (一次侧电源) 的状态。
• 亮灯: 主电源 (一次侧电源) ON
• 熄灯: 主电源 (一次侧电源) OFF
- 5) 紧急停止开关 (EMG. STOP)
使机器人紧急停止并将伺服 OFF。
- 6) 开始按钮 (START) 执行程序, 使机器人动作。程序连续运行。
- 7) SVO. ON 按钮..... 将伺服电源置为 ON。(伺服 ON)
- 8) 复位按钮 (RESET) 解除错误显示。或将已中断执行的程序进行复位。
- 9) 模式切换开关 (MODE) ... 切换机器人的操作权。
• 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。(程序的自动运行开始操作除外)
• MANUAL: 示教单元有效时, 仅限通过示教单元进行的操作有效。无法进行需要外部设备或操作面板的操作权的操作。
- 10) SVO. OFF 按钮..... 将伺服电源置为 OFF。(伺服 OFF)
- 11) 结束按钮 (END) 使执行中的程序停止在最终行或通过结束按钮使程序停止。
- 12) 停止按钮 (STOP) 使正在执行程序的机器人停止。不进行伺服 OFF。
- 13) 示教单元连接用连接器 (T/B)
未连接示教单元时, 根据使用环境要求应安装端盖或虚拟连接器 (另售)。
- 14) PC PC 连接用连接器。请打开盖板后连接 USB 电缆。
CR860-R/CR860-Q 控制器无法使用。
- 15) LAN 以太网连接用连接器。CR860-R/CR860-Q 控制器无法使用。

2.5.2 STATUS NUMBER (显示面板) 的说明

(1) STATUS NUMBER (显示面板) 的切换

可以通过显示切换按钮 (CHNG DISP)、UP/DOWN 按钮切换显示面板的显示。

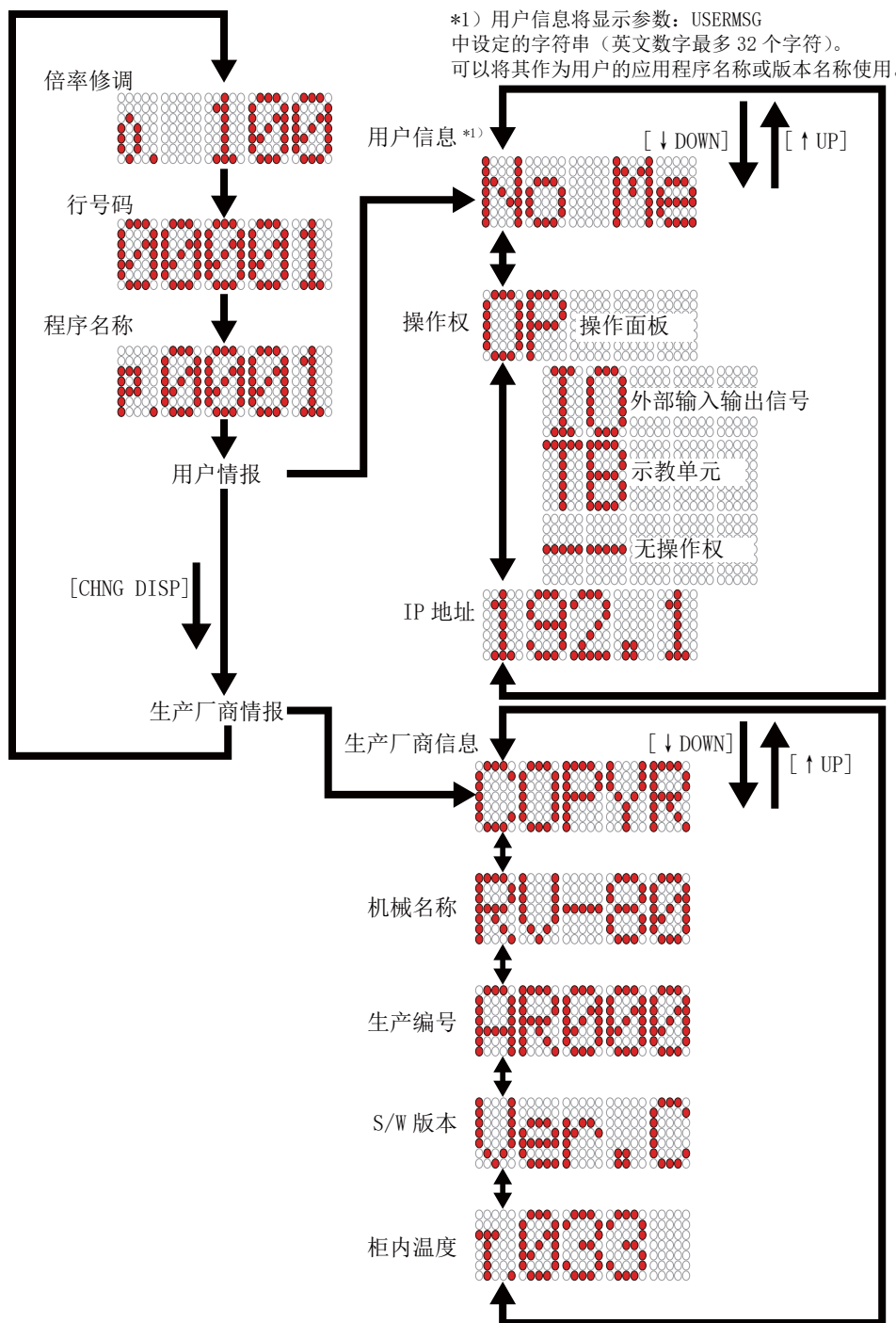


表 2-4: 显示面板的显示内容

显示内容		显示示例	说明
倍率修调		o. 100	显示当前倍率修调的值。
程序名称		p. 1234	显示设定在插槽 1 中的程序名称。 但是, PLC 直接执行中显示为“DMODE”。 ※ 程序名称为 5 个字符以上时, 不显示。从示教单元或 RT ToolBox3 等选择了程序名称为 5 个字符以上的程序时, 显示为“p. - - -”。或在插槽 1 中未设定程序时, 显示为“p. 0000”。
行号码		12345	显示当前所选择的程序的行号码。
用户情报	用户信息	No Message	显示用户信息。用户信息为6个字符以上时滚动显示。 用户信息为参数“USERMSG”中设定的字符串(最多可设定32个字符)。此外, 用户信息中不能使用[“], [,], [;]。“¥”显示为反斜线。并且, 不支持全角字符、半角片假名字符和空白字符(空格)。
	操作权	OP	显示取得控制器操作权的机器名称。 操作面板: OP 示教单元: TB 外部输入输出功能: IO 未获取操作权: - -
	IP 地址	192. 168. 0. 20	滚动显示控制器的IP地址。
生产厂商情报	生产厂商信息	COPYRIGHT(C) 2017-20xx MITSUBISHI ELECTRIC CORPORATION ALL RIGHTS RESERVED	滚动显示用户信息。
	机械名称	RV-35FR-D	滚动显示机器人的型号。
	生产编号	B10050088M	滚动显示控制器的生产编号。
	S/W 版本	Ver. C2g	滚动显示控制器的S/W版本。
	柜内温度	T. 035	控制器柜内温度显示为[单位: °C]。 CR860-R/D时, 机器人CPU内的温度显示为[单位: °C]。
错误号码显示		H. 0060	发生错误时, 错误号码闪烁显示。 如果同时发生多个错误, 则显示最高等级和最新的错误号码。此外, 按下UP/DOWN按钮, 可切换错误号码。

◇◆◇关于错误号码显示◇◆◇

错误号码显示中时按下显示切换按钮 (CHNG DISP) 则切换至常规的显示画面 (倍率修调、行号码、程序名称、用户情报、生产厂商情报)。此外, 如果松开显示切换按钮 (CHNG DISP), 则返回错误号码显示。再次按下显示切换按钮 (CHNG DISP) 时, 则显示下一个状态的显示画面 (倍率修调→行号码→程序名称→用户情报→生产厂商情报→倍率修调)。

(2) 各种状态显示

以左边的一个字符显示各种状态。

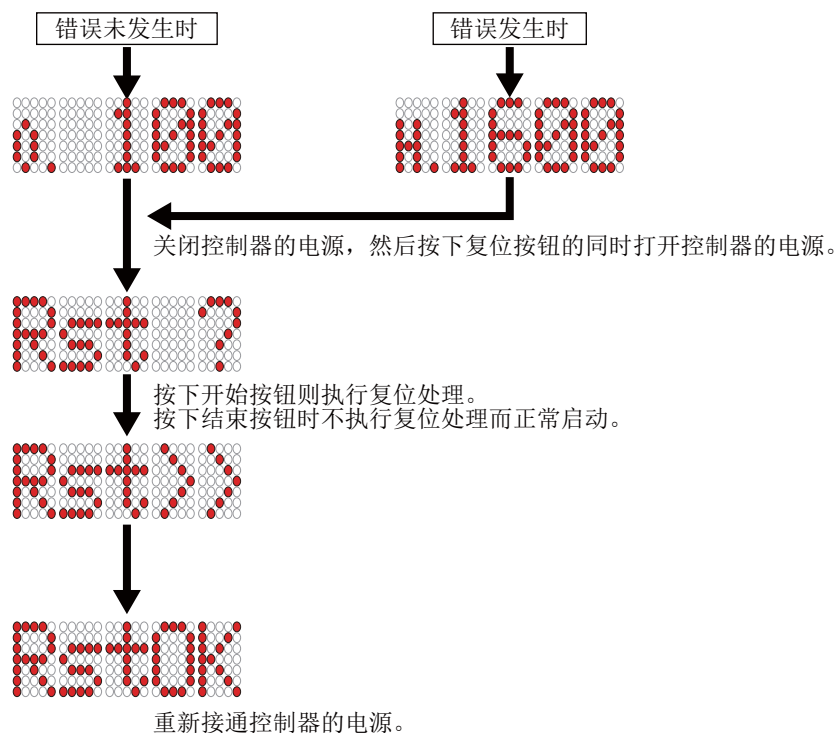
表 2-5: 冲突检知参数

倍率修调	程序名称	控制器柜内温度	高级别错误	低级别错误	警告	O/P 操作权 注1)

注1) LED 显示状态为倍率修调、程序名称、柜内温度时, 操作面板侧如果有操作权则左上方的点会亮起。

(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。



 注意

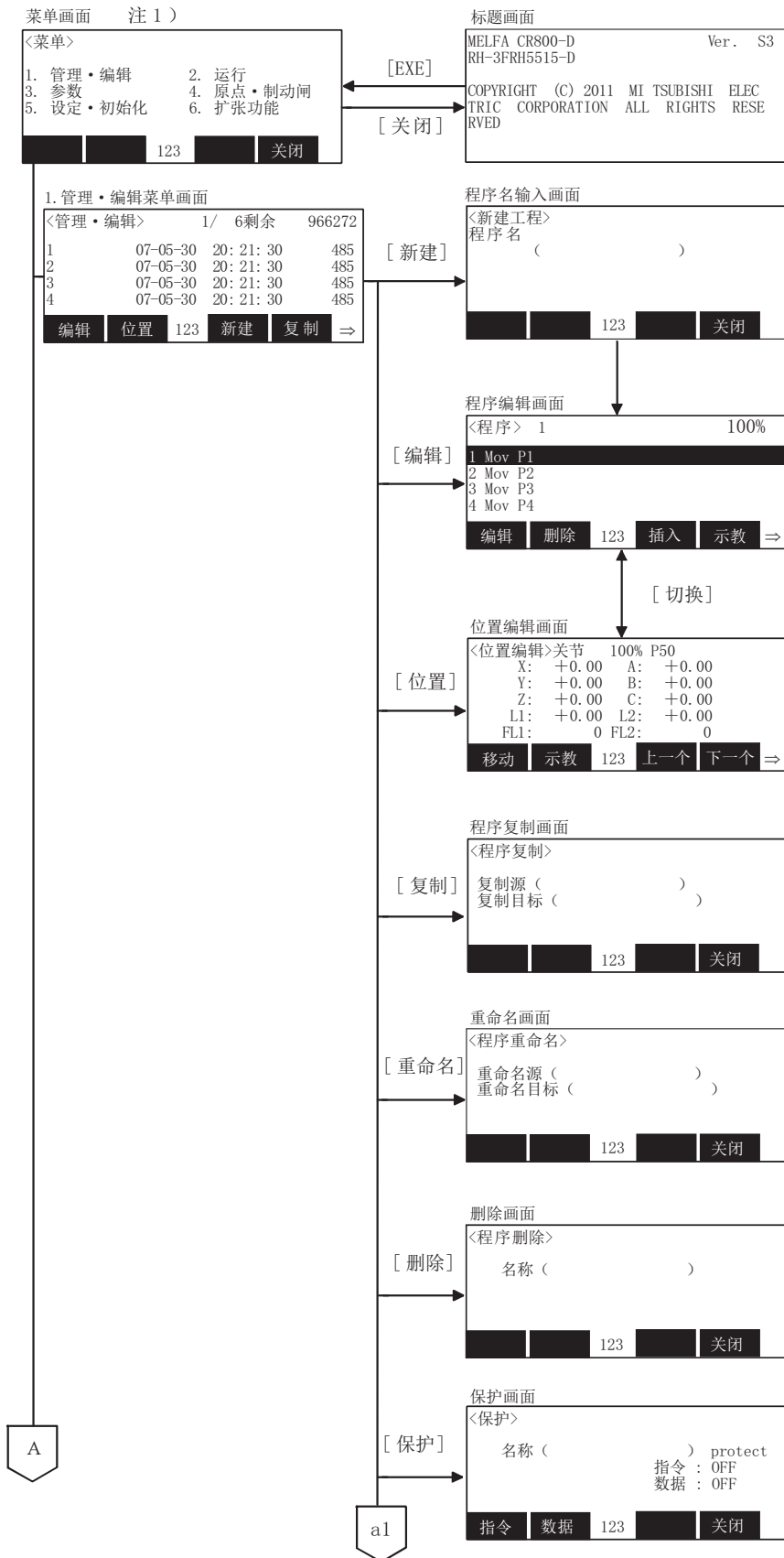
如进行本操作，则会变为出厂状态，参数、程序和各種日志数据会被删除，因此除发生错误（H1600）以外，请勿实施本操作。被删除的这些数据中如有 RT ToolBox3（选购件）的批量备份数据，则可通过恢复功能进行复原。

3 操作方法的说明

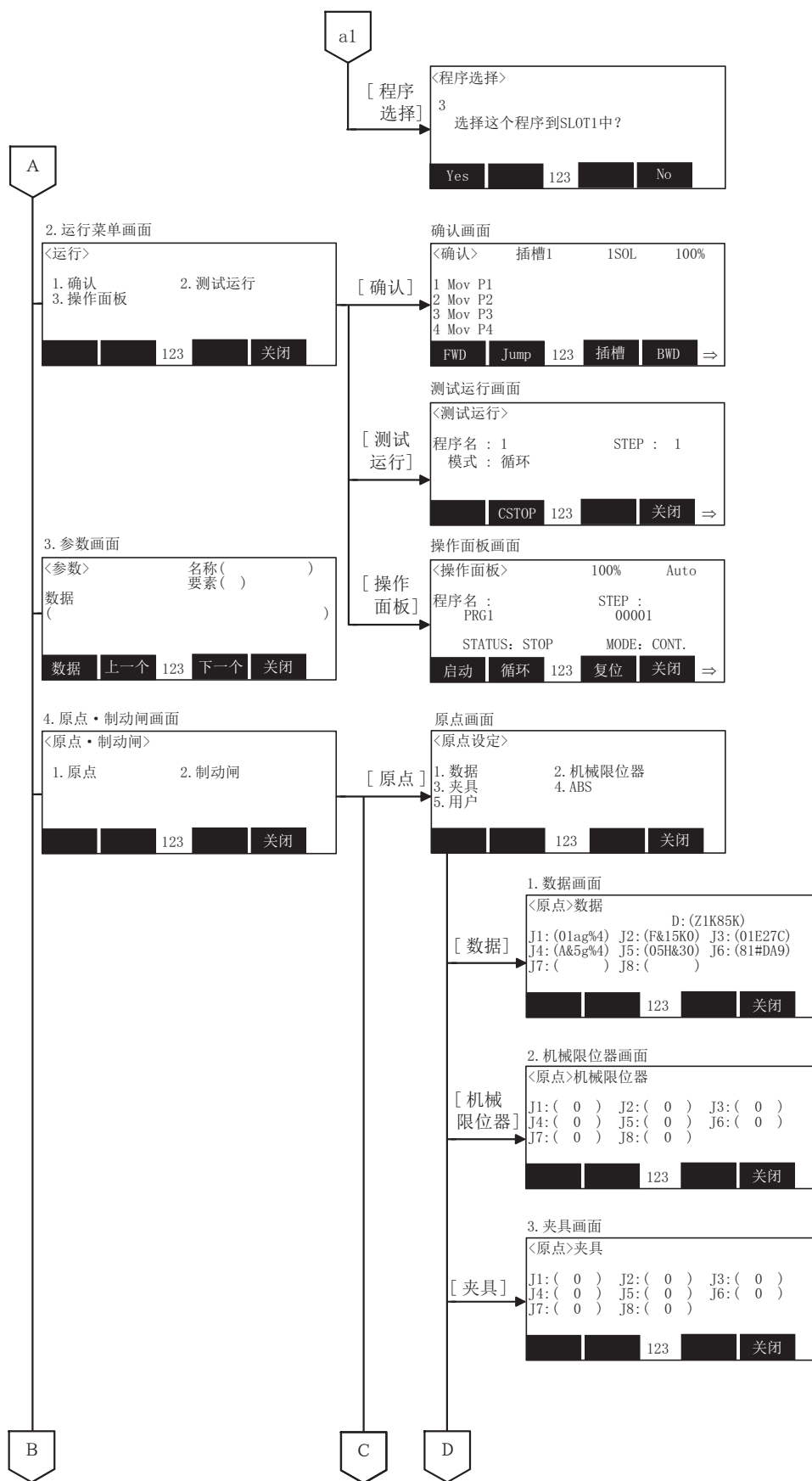
记述 R32TB（配件）的操作方法。

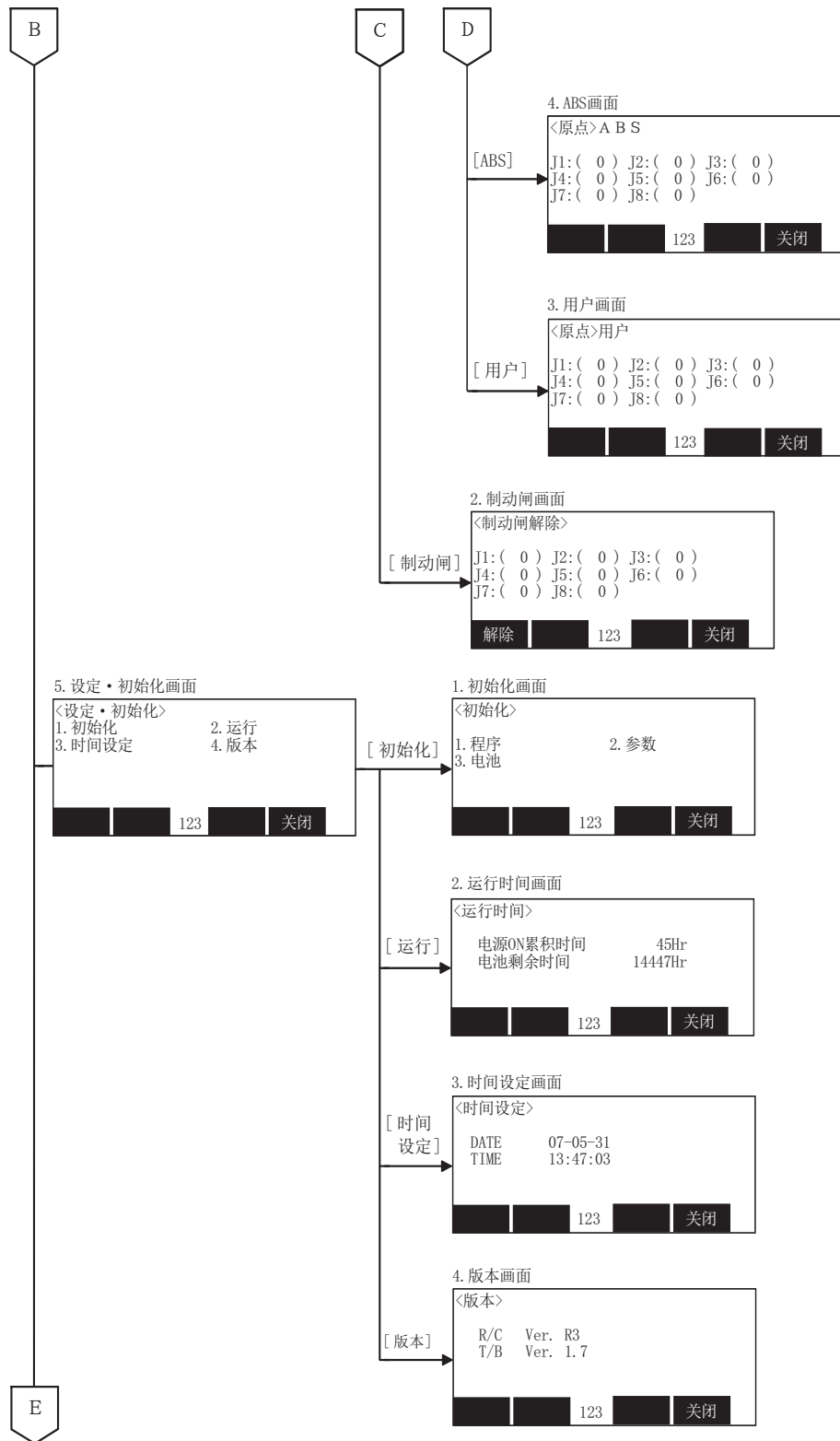
3.1 示教单元的菜单画面操作

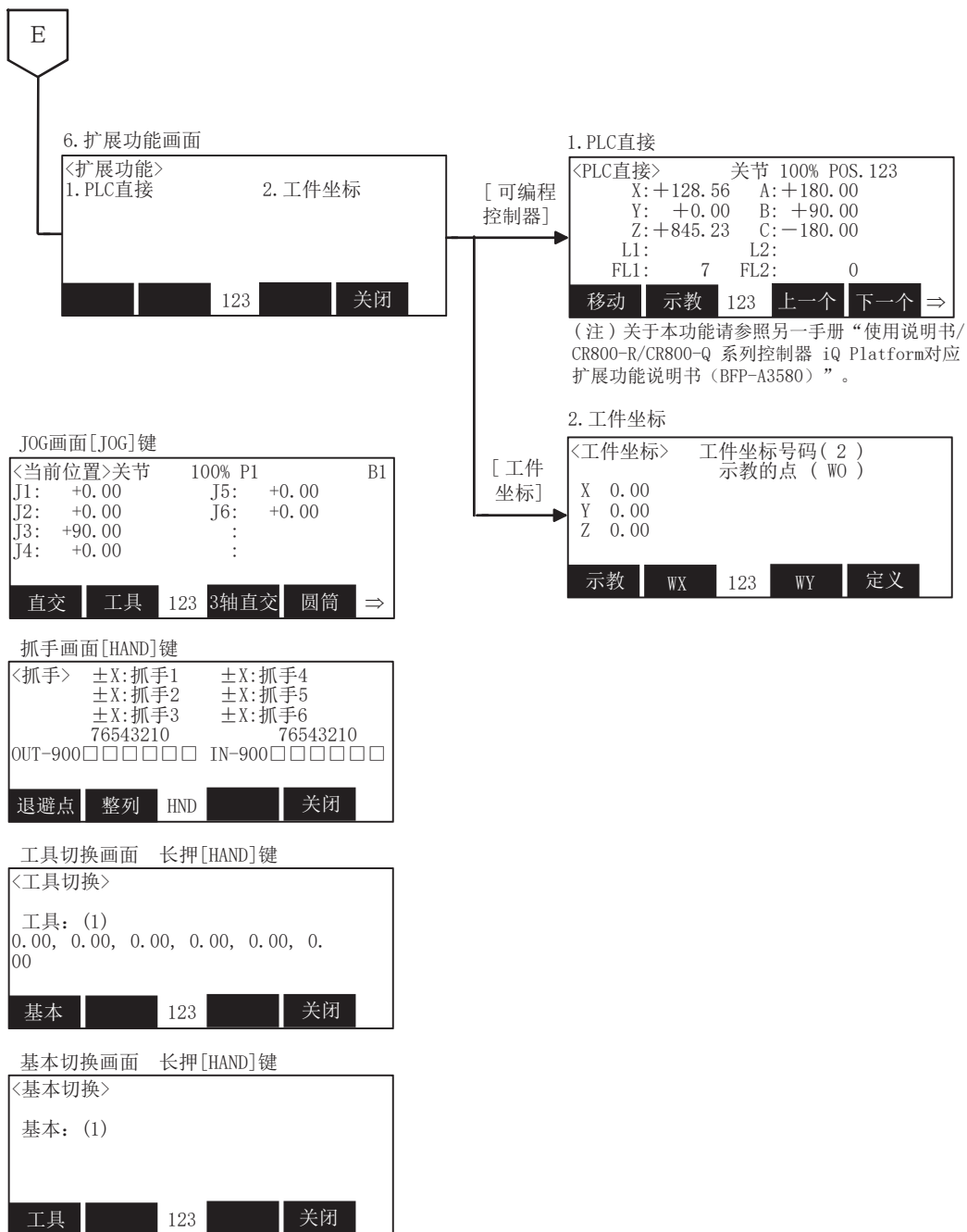
(1) 画面树形图



注1)关于6.扩展功能中的“1.PLC直接”，请参照另一手册“使用说明书/CR800-R/CR800-Q系列控制器 iQ Platform对应扩展功能说明书 (BFP-A3580)”。







(2) 数字 / 文字的输入

每次按下 [CHARACTER] 键，会切换数字输入模式与文字输入模式。

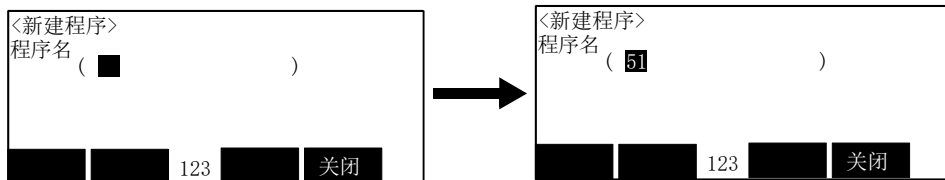
画面右下角会显示当前的输入模式，“123”表示数字输入模式，“ABC”表示文字输入模式。

1) 数字的输入

按下各键左下角显示的键可输入数字（包含“-”（负）、“.”（小数点））。

按下 [CHARACTER] 键，在画面右下角显示“123”的状态下，按下数字键进行输入。

例) 输入程序名为“51”时。



数字的输入 [CHARACTER] [5] [1]

2) 文字的输入

在各键的右下角显示文字。按下此键即可输入文字。

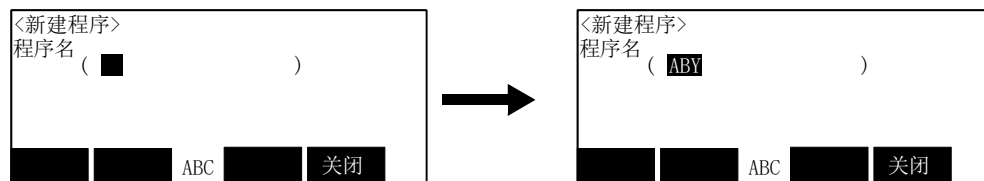
按下 [CHARACTER] 键，在画面右下角显示“ABC”的状态下，按下文字键进行输入。

每次按下有多个文字显示的键，输入的文字会变换。

例) [ABC] 键时，每次按下此键会重复“A”“B”“C”“a”“b”……。

要连续输入相同键上显示的文字时，应按下 [→] 键使光标前进。

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



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

可以输入按键上未显示的文字。分配了未显示的文字的键及通过该键可输入的文字如下所示。

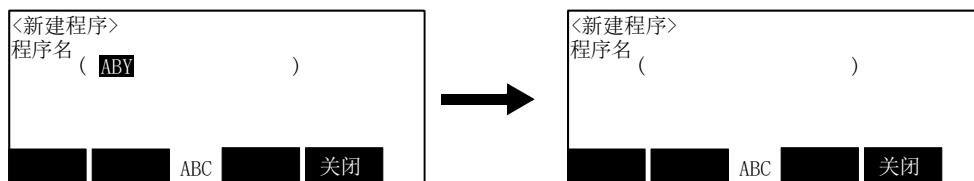
- a) [' ()] 键 ... ' → (→) → ” → ^ → : → ; → \ → ?
- b) [@ =] 键 @ → = → + → - → * → / → < → >
- c) [, %] 键 , → % → # → \$ → ! → & → _ → .

3) 文字的删除

错误输入的文字，按下 [CLEAR] 键，可删除光标所在位置的 1 个文字。

例) 要将“ABY”的“B”变更为“M”，变成“AMY”时。

将光标移动到文字“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]

长按住 [CLEAR] 键后，括号内的文字将全部删除。

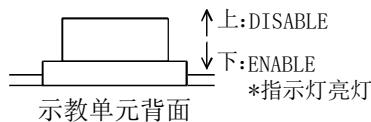
(3) 菜单的选择方法

菜单的选择方法有以下 2 种。请选择其中一种方法执行。

- 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。

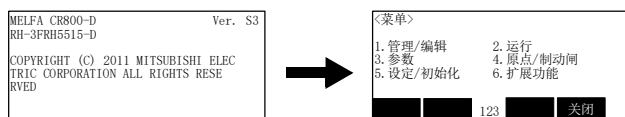
显示通过各个方法根据菜单画面选择“1. 管理 · 编辑”时的情况。

示教单元



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

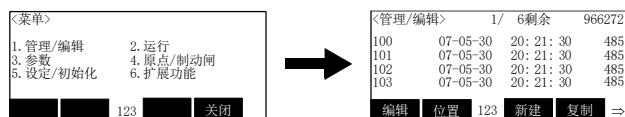
在标题画面中显示菜单画面。



2) 通过在出现 <标题> 画面的状态下按下任一 个键（例 [EXE] 键）显示 <菜单> 画面。

菜单画面的显示 [EXE]

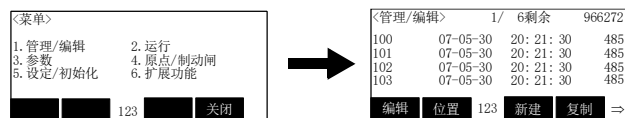
a) 用数字键选择菜单



按下 [CHARACTER] 键设为数字输入模式，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。显示 <管理 · 编辑> 画面。

菜单画面的显示 [↑] [↓] [←] [→] + [EXE]

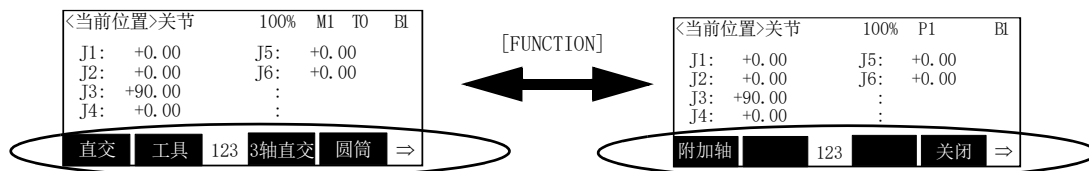
其他的菜单画面也可进行相同操作。

◆◆◆使用示教单元时◆◆◆

将控制器的模式设为“MANUAL”。除此以外仅可进行特定的操作（在 JOG 画面显示当前位置、变更倍率修调、监视）。

◆◆◆功能键◆◆◆

在画面的最下段有用反白字显示的菜单。这些菜单以从左到右的顺序分配到 [F1]、[F2]、[F3]、[F4] 键。通过按下对应的功能键可选择显示的菜单。此外，菜单的右端显示“=>”时，表示除当前显示外还有其他菜单，每次按下 [FUNCTION] 键，显示菜单将切换。



3.2 JOG前进（概略说明）

JOG 前进是指以手动方式使机器人运动的操作。

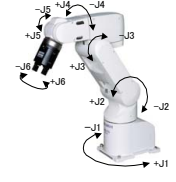


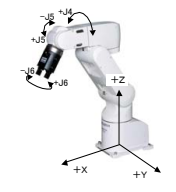
这里以垂直多关节机器人为基准对概略进行说明。

由于根据机器人的机型，轴的构成会有所差异，因此关于各个机型请参照另一手册“从机器人本体安装到维护”篇。

3.2.1 JOG前进的种类

JOG 前进有以下 6 种。（以下显示的机器人的图以垂直多关节机器人 6 轴型为示例。）

表 3-1：JOG 前进的种类

种类	操作	说明
 <p>关节 JOG</p>	<ol style="list-style-type: none"> 1) 将 [TB ENABLE] 开关设为 ENABLE。 2) 按下有效开关（3 位置开关）。 3) 按下 [SERVO] 键使伺服 ON。 4) 按下 [JOG] 键，再按下 [F1]（关节）键，选择关节 JOG 模式。 5) 按下各轴的键。 J1 ~ J6 轴。 	<p>使机器人移动到各轴。 使 J1 ~ J6 轴及附加轴 J7、J8 各自独立运动。 轴数根据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>
 <p>直交 JOG</p>	<ol style="list-style-type: none"> 1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“直交”的任一功能键（从 [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，以机器人的世界坐标系为基准，保持抓手的方向朝 X、Y、Z 方向进行直线动作。 按下 A、B、C 键，以世界坐标系的 X、Y、Z 轴为中心旋转，改变抓手的方向。前端的位置固定。 但是，需要先通过参数 MEXTL 正确设定工具长度。</p>
 <p>工具 JOG</p>	<ol style="list-style-type: none"> 1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“工具”的任一功能键（从 [F1] 到 [F4]），选择工具 JOG 模式。 5) 按下各轴的键。 X、Y、Z、A、B、C 轴。 	<p>按下 X、Y、Z 键，以将机器人抓手前端作为原点的工具坐标系为基准，保持抓手的方向朝前后、左右、上下方向进行直线移动。 按下 A、B、C 键，以工具坐标系的 X、Y、Z 轴为中心旋转，改变抓手的方向。前端的位置固定。但是，需要先通过参数 MEXTL 正确设定工具长度。 抓手前端的工具坐标系根据机器人机型而有所差异。 垂直多关节机器人由机械接口面到抓手前端方向为 +Z。 水平多关节机器人由机械接口面到上方向为 +Z。</p>
 <p>三轴直交 JOG</p>	<ol style="list-style-type: none"> 1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“3 轴直交”的任一功能键（从 [F1] 到 [F4]），选择 3 轴直交 JOG 模式。 5) 按下各轴的键。 X、Y、Z、A、B、C 轴。 	<p>按下 X、Y、Z 键，以机器人的世界坐标系为基准，向 X、Y、Z 方向进行直线动作。 按下 A、B、C 键，会和关节 JOG 一样以关节为基准进行动作，但为了保持控制点的位置（XYZ 值），姿势会改变。（和直交 JOG 不同，由于在 X、Y、Z、J4、J5、J6 中插补，因此不会保持姿势） 需要先通过参数 MEXTL 正确设定工具长度。</p>

种类	操作	说明
圆筒 JOG 	1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“圆筒”的任一功能键（从 [F1] 到 [F4]），选择圆筒 JOG 模式。 5) 按下各轴的键。 X、Y、Z、A、B、C 轴。	以机器人的世界坐标系的原点为基准，在抓手向圆筒方向动作的情况下使用。 X 轴从世界坐标系的原点向放射线方向移动。 Y 轴的动作和使 J1 轴旋转的关节 JOG 相同。（圆弧动作） Z 轴和直交 JOG 的 Z 相同。（上下动作） 按下 A、B、C 键，将进行和直交 JOG 相同的动作。 在水平多关节机器人中也可能有效。
工件 JOG （工件 JOG 模式） 	1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“工件”的任一功能键（从 [F1] 到 [F4]），选择工件 JOG 模式。 5) 按下各轴的键。 X、Y、Z、A、B、C 轴。	进行该 JOG 操作时，需要先将参数 WKnJOGMD (n=1 ~ 8) 设定为“0（工件 JOG 模式）”。 按下 X、Y、Z 键，以事先设定的机器人的工件坐标系为基准，保持抓手的方向朝 X、Y、Z 轴方向进行直线动作。 按下 A、B、C 键，以相同工件坐标系的 X、Y、Z 轴为中心旋转，改变抓手的方向。前端的位置固定。 但是，需要先通过参数 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） 未设定工件坐标系时，变为直交 JOG 中的动作。
工件 JOG （Ex-T JOG 模式） 	1) 1) ~ 3) 和关节 JOG 相同。 4) 按下对应“工件”的任一功能键（从 [F1] 到 [F4]），选择工件 JOG 模式。 5) 按下各轴的键。 X、Y、Z、A、B、C 轴。	进行该 JOG 操作时，需要先将参数 WKnJOGMD (n=1 ~ 8) 设定为“1（Ex-T JOG 模式）”。 按下 X、Y、Z 键，以事先设定的机器人的工件坐标系为基准，保持抓手的方向朝 X、Y、Z 轴方向进行直线动作。 按下 A、B、C 键，以相同工件坐标系的 X、Y、Z 轴为中心旋转，改变抓手的方向的同时移动控制点。 但是，需要先通过参数 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） 未设定工件坐标系时，XYZ 变为直交 JOG 中的动作。 ABC 进行以机器人的原点坐标系为中心旋转的动作。

此外，在表 3-1 的种类中所示的 JOG 内，直交 JOG、工具 JOG、圆筒 JOG、工件 JOG 动作时，机器人的控制点接近特异点后，会发出蜂鸣声，同时，示教单元的画面会显示警告符号，提醒机器人的操作人员注意。本功能可根据参数（MESNGLSW）设定为有效 / 无效。（参照第 468 页的“5 参数的设定功能”）

此外，本功能的详细内容如请参照第 525 页的“5.17 关于特异点接近警告”所示。

3.2.2 JOG前进的速度

可通过 [OVRD ↑]/[OVRD ↓] 键变更速度。当前的设定速度在画面右上方显示为 %。
 JOG 前进的速度有以下种类。

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

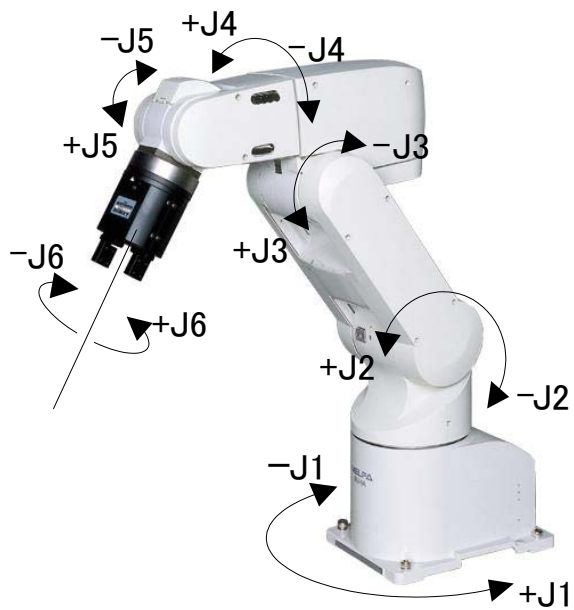
LOW 和 HIGH 为定寸前进。定寸前进是指每按 1 次键，机器人会以一定的量动作。移动量根据机器人机型而有所差异。

表 3-2: 固定尺寸示例

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

3.2.3 关节JOG

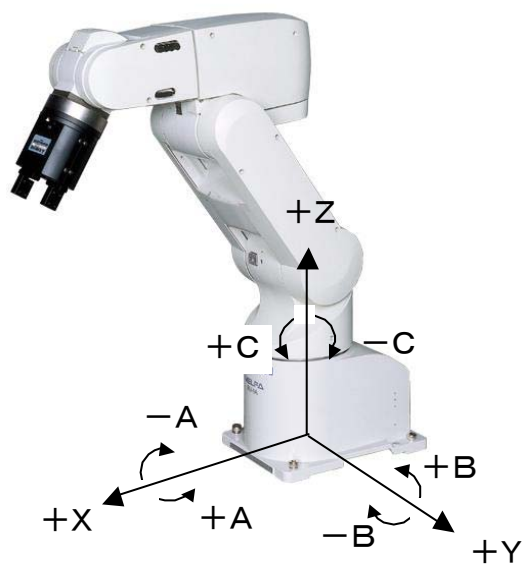
在各轴以角度为单位使轴动作。



3.2.4 直交JOG

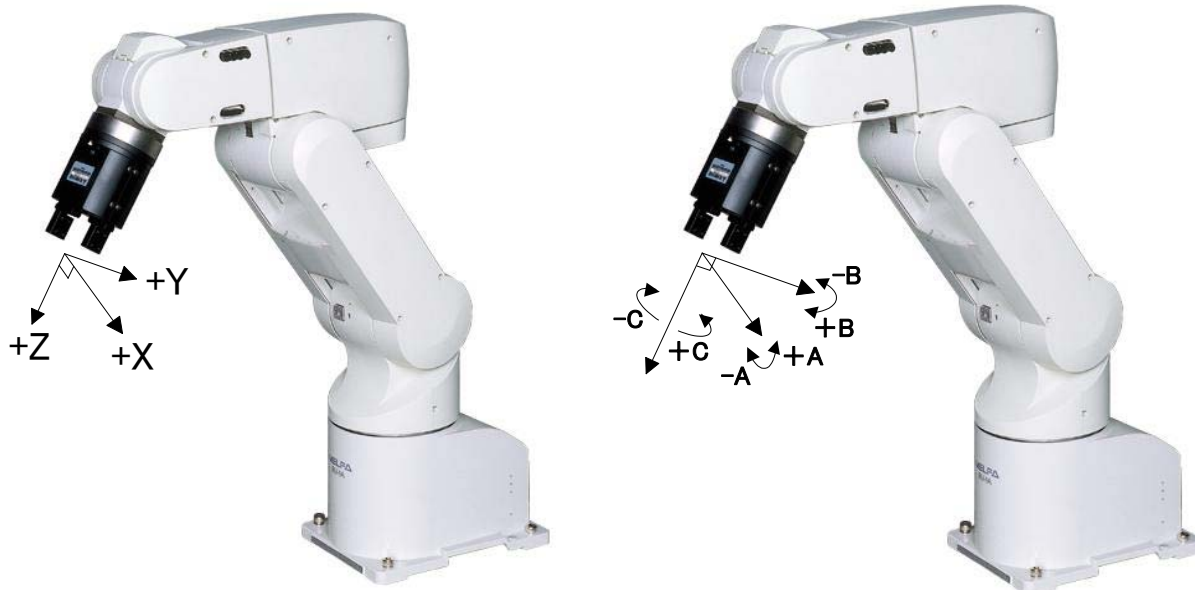
沿着世界坐标系的方向移动。

XYZ 以 mm 为单位移动。ABC 以角度为单位移动。



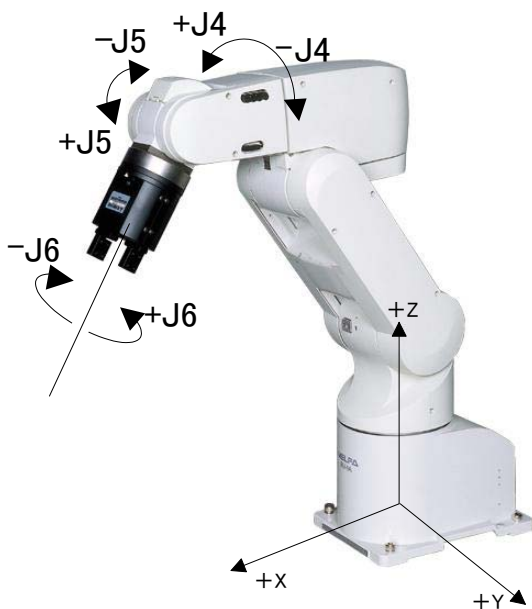
3.2.5 工具JOG

沿着抓手前端的方向移动。
XYZ 以 mm 为单位移动。ABC 以角度为单位移动。



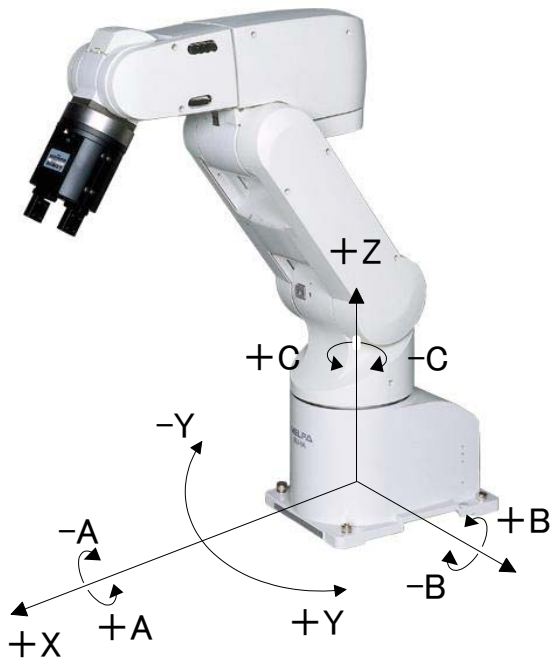
3.2.6 三轴直交JOG

XYZ 和直交 JOG 一样沿着世界坐标系的方向移动。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。
XYZ 以 mm 为单位移动。ABC 以角度为单位移动。



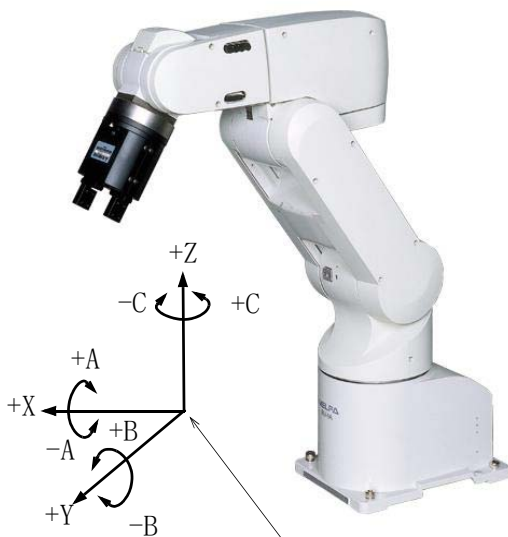
3.2.7 圆筒JOG

X 从机器人的世界坐标系的原点以放射状向远离方向 / 接近方向移动。Y 以 J1 轴为中心旋转。Z 向世界坐标系的 Z 方向移动。ABC 轴的动作和直交 JOG 相同。X、Z 以 mm 为单位移动。Y、A、B、C 以角度为单位移动。



3.2.8 工件JOG

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



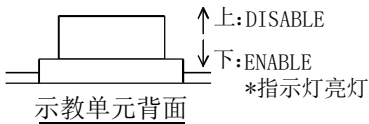
工件坐标系：对准工件及作业台等的坐标系

3.2.9 工具转换数据的切换

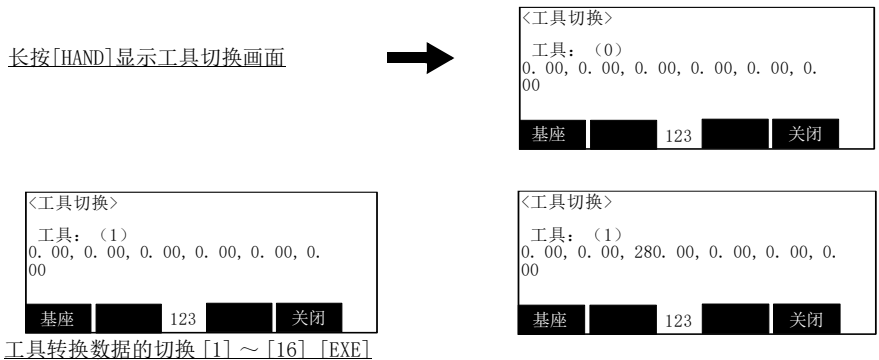
使用多个抓手时，需要预先登录各自的工具转换数据，选择正确对应抓手的工具转换数据再使用。最多可登录 16 种参数。

将要使用的工具数据先设定为参数：MEXTL1 ~ 16，按照下列操作选择要使用的工具编号。

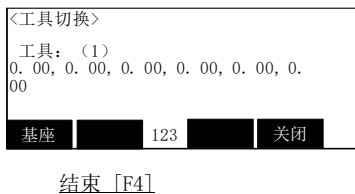
- 1) 将控制器的模式设为“MANUAL”，按下示教单元的 [ENABLE] 开关，将示教单元设为有效。（开关和 ENABLE LED 亮灯）



- 2) 长按 [HAND] 键显示 <工具切换> 画面。
- 3) 按下希望的数字键后，按下 [EXE] 键，将切换工具转换数据。参数 MEXTL1 ~ 16 对应数字 1 ~ 16。



- 4) 按下分配到“关闭”的功能键，结束。



- 5) JOG 画面的右上角显示当前的工具编号 (T1 ~ T16)。

⚠ 注意

程序自动运行时，切换工具转换数据（MEXTL1 ~ 16 参数）的同时使机器人移动到已示教的位置的情况下，应在必要的位置将工具编号代入 M_TOOL 变量，切换工具转换数据并运行。如果示教时的工具转换数据和运行时的工具编号不一致，会向意想不到的方向动作，请充分注意。

⚠ 注意

程序单步运行时，在切换工具转换数据动作的情况下，如果示教时的工具转换数据和单步运行时的工具编号不一致，会出现意想不到的动作，请充分注意。

◇◆◇工具编号的确认◇◆◇

当前的工具编号可通过工具切换画面、JOG 画面或 M_TOOL 变量进行确认。

◇◆◇相关信息◇◆◇

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

Tool 指令、M_Tool 变量

MEXTL 参数是保持当时工具转换数据的参数。请注意使用 MEXTL1 ~ 16 时，在选择工具编号时，MEXTL 参数将被覆盖。

此外，为了将工具编号恢复为 0，应执行 Tool 指令

3.2.10 世界坐标系的切换（基本坐标编号的指定）

通过示教单元的操作可以轻松地变更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为参数 WK1CORD ~ WK8CORD。（参照另一手册“从机器人本体安装到维护”的“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。

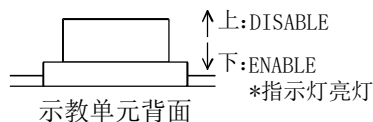
在进行位置示教前，应确认程序中使用的 Base 指令所对应的示教位置（位置变量）和当前的 JOG 操作中设定的基本坐标编号是否正确。

如果以错误的基本坐标编号进行示教，执行程序时机器人将发生意想不到的动作，导致设备损坏及人身事故。

应将基本坐标编号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，将示教单元设为有效。（开关和 ENABLE LED 亮灯）



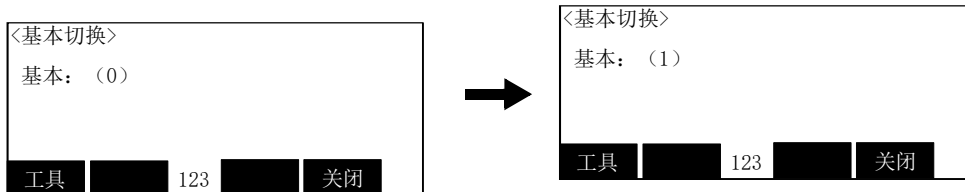
- 2) 长按 [HAND] 键显示〈基本切换〉画面。

显示〈工具切换〉画面时，按下画面下方的“Base”相应的功能键 [F1]。

〈基本切换〉画面的显示：长按 [HAND] 键



- 3) 输入希望的基本坐标编号后，按下 [EXE] 键，将切换世界坐标系。
1 ~ 8: 基本坐标编号 (对应参数 WK1CORD ~ WK8CORD)
0: 恢复出厂时的状态。(无基本转换的状态)



(图为切换至基本坐标编号 1 的示例)

通过基本坐标编号设定世界坐标系 [0]、[1] ~ [8]、[EXE]

- 4) 按下分配到“关闭”的功能键，结束。



结束 [F4]

- 5) JOG 画面的右上方会显示当前的基本坐标编号 (B0 ~ B8)。

3.2.11 JOG操作中的冲突检知

具备碰撞检测功能的机型，可通过参数切换其有效 / 无效。检测到冲突时，会发生错误编号 101n（低 1 位的 n 为轴编号）。

JOG 操作时也可将本功能设为有效，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	要素数	内容	初始值
冲突检知 注 1)	COL	整数 3	定义冲突检知的可使用 / 不可使用及电源 ON 后的有效 / 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 操作中设为有效为 (1) / 不设为有效为 (0) / NOERR 模式为 (2) NOERR 模式下即使检测到冲突也不会显示错误。只会变为伺服 OFF。应在检测到碰撞时频繁发生错误导致操作困难的情况下使用。 除了程序运行时（包含位置跳转、单步前进）以外，依存于 JOG 操作作用（要素 3）的设定。	RH-3FRH/6FRH/12FRH/ 20FRH 系列： 1, 0, 1 RV-FR 系列： 0, 0, 1 RH-3FRHR 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 (%) 升高检知标准（感应度）时应将数值调小。 JOG 操作中若未冲突却发生冲突检知错误时，应将数值调大。 设定范围 1 ~ 500 (%)	根据机型而有所差异。
抓手条件	HNDDATO	实数 7	设定抓手的初始条件。（通过工具坐标系指定） 电源 ON 后即执行的 JOG 操作将采用此设定值。 在 JOG 操作时使用碰撞检测的情况下，应设定实际的抓手条件。若未设定，可能会无法正确检知。 （质量，大小 X，大小 Y，大小 Z，重心 X，重心 Y，重心 Z） 单位：Kg, mm	根据机型而有所差异。
工件条件	WRKDATO	实数 7	设定工件的初始条件。（通过工具坐标系指定） 电源 ON 后即执行的 JOG 操作将采用此设定值。 （质量，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, 0.0

注 1) 本功能无法同时与多重机器人控制使用。

(1) JOG 操作时的碰撞检测标准调整

JOG 操作时的碰撞检测感应度设定较低。希望设定较高的碰撞检测感应度时，应调整参数 COLLVLJG 后使用。此外也应正确设定参数 HNDDATO 及 WRKDATO 后使用。在未正确设定的情况下进行 JOG 操作时，有些姿势可能会出现错误检测。

◆◆◆碰撞检测功能的注意点◆◆◆

即使将碰撞检测功能设为有效，也无法 100% 阻止由于外围设备的干涉导致的机器人、抓手、工件等的损伤。基本上还是应小心操作机器人，勿与周边设备发生干涉。

◆◆◆碰撞后的操作◆◆◆

在抓手及机械臂与周边设备发生了干涉时，若直接伺服 ON，可能会再次变成碰撞检测状态，而无法伺服 ON。多次重复伺服 ON 后仍然发生错误的情况下，应暂时进行制动闸解除操作，使机械臂闪避后再伺服 ON，或者执行第 68 页的“3.11 无法解除错误的暂时错误复位操作”，伺服 ON 后再通过 JOG 操作使机械臂闪避。

◆◆◆与自动运行用的碰撞检测的关系◆◆◆

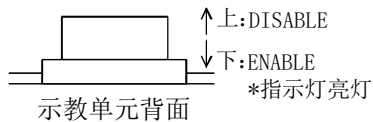
JOG 操作用的碰撞检测和自动运行用的碰撞检测是独立设定的。
未进行程序运行时将采用 JOG 操作用的设定。JOG 操作用的设定为有效时，即使在程序中将自动运行用的碰撞检测设定为无效，只要中断运行，就会切换成 JOG 操作用的设定（碰撞检测有效）。

3.3 抓手开闭

对安装在机器人上的抓手的开闭操作进行说明。

最多可控制 6 个抓手。X、Y、Z、A、B、C 轴的键，各自分配为抓手 6、5、4、3、2、1，抓手开为“+”侧的键，抓手闭为“-”侧的键，按下即可操作。

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



- 2) 按下“HAND”键显示 <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 时按下 [+C] 键、要关闭时按下 [-C] 键。其他抓手可通过 X、Y、Z、A、B 轴的键进行相同的操作。

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	
	76543210	76543210	
OUT-900	■ ■ ■ ■ ■ ■ ■ ■	IN-900	□ □ □ □ □ □ □ □
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
抓手编号	4		3		2		1	

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

⚠ 注意

抓手画面中监视画面显示的输入输出信号为 900 ~ 907。
 由于与 CR860 控制器的抓手输入输出信号不同，示教单元的抓手画面无法监视抓手信号的状态。
 抓手输入信号号码：764 ~ 775
 抓手输出信号号码：764 ~ 771

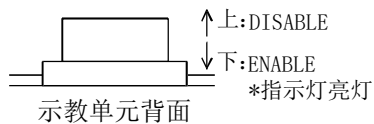
◇◆◇与自动运行用的碰撞检测的关系◆◆◇
 机器人的抓手部分可安装各式各样的工具。通过气动控制在电磁阀中使用双螺线型电导管时，通过抓手开闭操作来控制抓手信号的 2 位。
 抓手信号的详细内容请参照第 512 页的“5.12 关于抓手种类”及第 513 页的“5.13 关于抓手初始状态”。

3.4 退避点复归

可使机器人向退避点（通过参数 JSAFE 指定的位置）移动。

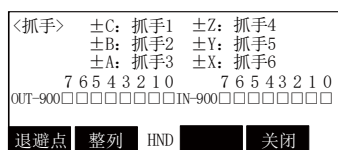
退避点复归的步骤如下所示。

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

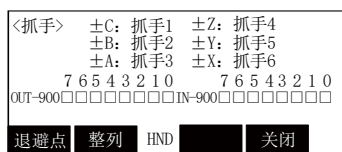


- 2) 按下有效开关（3 位置开关），再按下 [SERVO] 键，伺服 ON。
- 3) 按下“HAND”键显示 < 抓手 > 画面。

显示抓手画面 [HAND]



- 4) 按住有效开关（3 位置开关），再持续按下分配到“退避点”的功能键。持续按键期间，机器人将进行退避点复归动作，动作中控制器的 [START]LED 亮起。动作途中放开任一个键机器人都将停止。

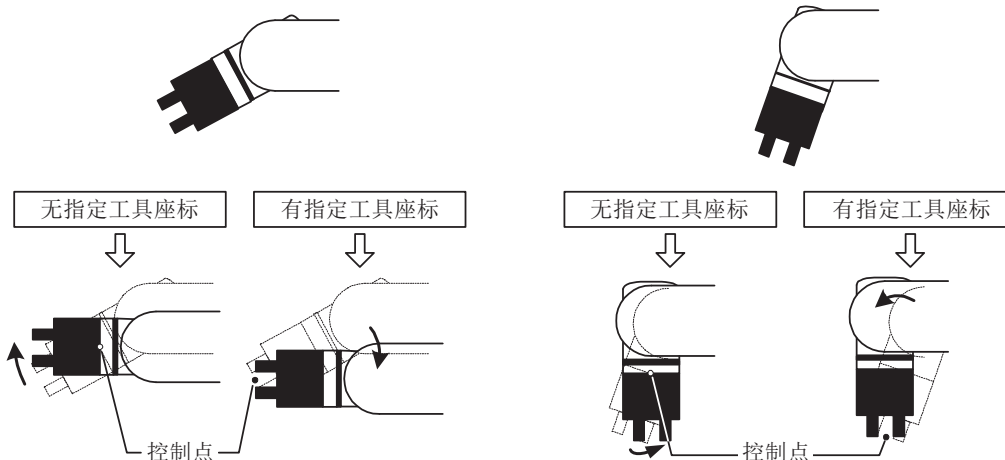


退避点复归的执行 [退避点]

3.5 抓手整列

安装在机器人上的抓手的姿势，可整列为 90 度单位的姿势。

此功能下，机器人会移动到当前位置的 A、B、C 成分中最接近 90 度单位的值。

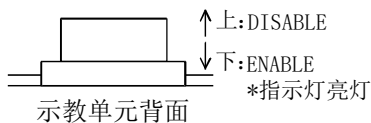


通过工具指令及参数指定工具座标时，将在指定的工具座标上进行抓手整列。未指定工具座标时，将在机器人接口中心进行抓手整列。

上图为垂直多关节机器人的示例，[有指定工具座标]是在抓手前端已指定工具座标的情况。关于工具座标请参照第 500 页的“5.6 关于标准 TOOL 坐标”。

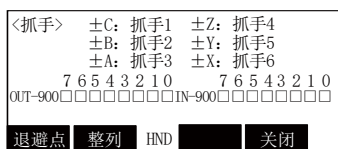
抓手整列的步骤如下所示。

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

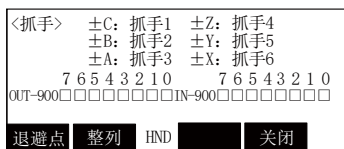


- 2) 按下有效开关（3 位置开关），再按下 [SERVO] 键，伺服 ON。
- 3) 按下“HAND”键显示 < 抓手 > 画面。

显示抓手画面 [HAND]



- 4) 按住有效开关（3 位置开关），再持续按下分配到“整列”的功能键。持续按键期间，机器人将进行整列动作，动作中控制器的 [START]LED 亮起。动作途中放开任一个键机器人都将停止。



抓手整列的执行 [整列]

⚠ 注意

根据抓手整列的操作，姿势成分（A、B、C）变为 180 度时，即使姿势相同，成分的值也将变为 +180 度或 -180 度的其中之一。这是由于内部的运算误差所造成，会变为哪一个值并没有规则性。将此位置作为 Pallet 定义指令（Def Plt）的位置数据使用时，如果相同的姿势成分中 +180 度和 -180 度同时存在，由于 Pallet 运算会分割 -180 ~ +180 度来计算位置，导致抓手旋转并引发意想不到的动作。将姿势成分为 180 度的位置数据在 Pallet 的定义中使用，应将 180 度的符号统一为 + 或 - 的其中之一。此外，将此位置数据作为插补指令的目的位置直接使用时，无论用哪一个值都可以进行动作。

3.6 程序编辑

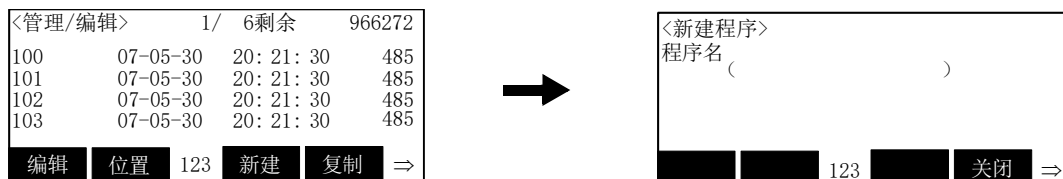
本机器人 CPU 中使用的 MELFA-BASIC VI 可根据丰富的运算功能记述高级的作业。此外，在此对使用了示教单元的程序编辑方法进行说明。关于 MELFA-BASIC VI 的指令、记述等，请参照本说明书 / 第 173 页的“4.12 指令的详细说明”的相关页。

3.6.1 程序的制作

(1) 程序编辑画面的开启

打开新建程序的编辑画面。开启程序名为“1”的示例如下所示。

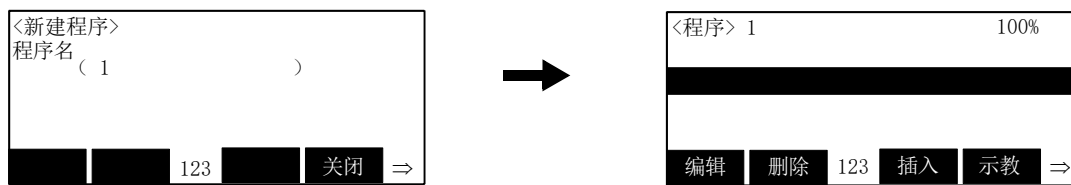
- 1) 在〈菜单〉画面选择“1. 管理・编辑”画面。
- 2) 按下对应“新建”的功能键。显示程序名输入画面。



菜单的选择 [F3]

- 3) 输入程序名。显示指令编辑画面。

(输入现有的程序名时，将开启现有的程序)



程序编号1的输入 [1] [EXE]

◇◆◇始终执行程序 (ALWAYS 属性) 的编辑◇◆◇

通过 SLTn 参数编辑已设定为始终执行 (ALWAYS) 属性的程序时，应先解除始终执行属性再编辑。由于始终执行属性的程序始终被执行，因此无法编辑。应通过 SLTn 参数将 ALWAYS 变更为 START，重新开启控制器的电源，停止始终执行。

◇◆◇开启现有的程序◇◆◇

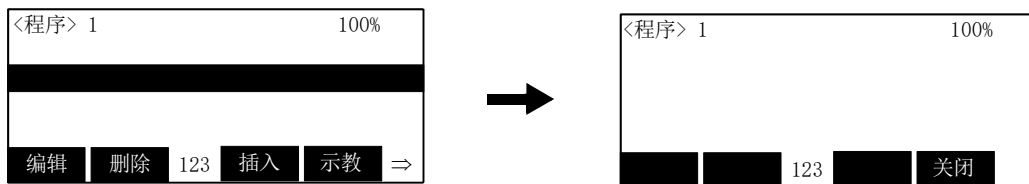
如上所述，可以在程序名中输入现有的程序名，也可以在〈管理・编辑〉画面按下 [↑]、[↓] 键，在选中要开启的程序的状况下，按下分配到“编辑”的功能键。

(2) 程序的制作

输入 3 单步的程序时的按键操作如下所示。

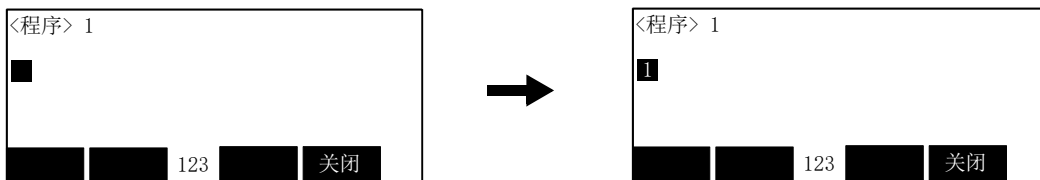
- 1 Mov P1
- 2 Mov P2
- 3 End

1) 在指令编辑画面，按下对应“插入”的功能键 ([F3])。



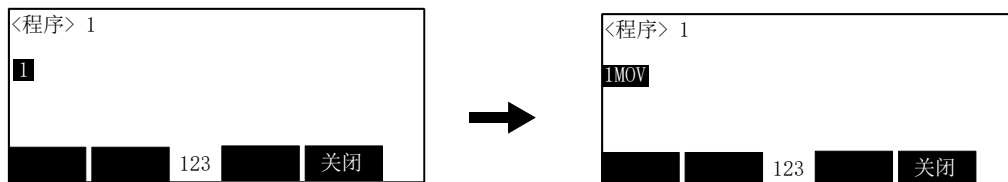
单步的插入 [F3]

2) 步号“1”的输入
 按下 [CHARACTER] 键设为数字输入模式，按下 [1] 键。
 可省略步号和指令后之间的空格。



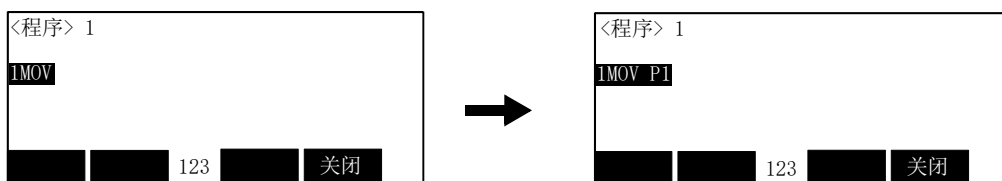
步号的输入 [1]

3) 指令语“Mov”的输入
 按下 [CHARACTER] 键，设为文字输入模式，
 按照 3 次 [MNO] (“M”)、[→]、[MNO]，3 次 (“0”)、[TUV]， (“V”)、[→] 的顺序按键。



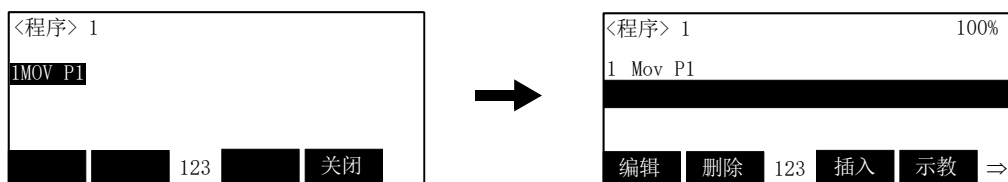
Mov的输入 [CHARACTER] [MNO] [→] [MNO] [MNO] [MNO] [TUV] [TUV] [TUV]

4) 指令语上附带的数据“P1”的输入
 按下 [SP] (“SPACE”)、[PQRS] (“P”) 键。
 按下 [CHARACTER] 键设为数字输入模式，按下 [1] 键。
 指令语和指令附带的数据之间要有空格。



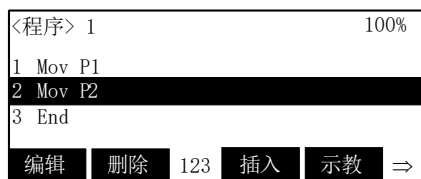
P1的输入 [SP] [PQRS] [CHARACTER] [1]

- 5) 单步 1 的确定
按下 [EXE] 键进行确定。



单步1的确定 [EXE]

- 6) 以下，按同样的方式输入单步 2、3。



至此，程序的输入完毕。

◇◆◇显示单步的前后◇◆◇

按下 [↑]、[↓] 键可使画面卷动。

◇◆◇显示特定单步◇◆◇

按下对应“JUMP”的功能键，在显示的单步编号输入画面中输入希望显示的单步编号，然后按下 [EXE] 键，显示输入的单步。

◇◆◇插入时可省略单步编号◇◆◇

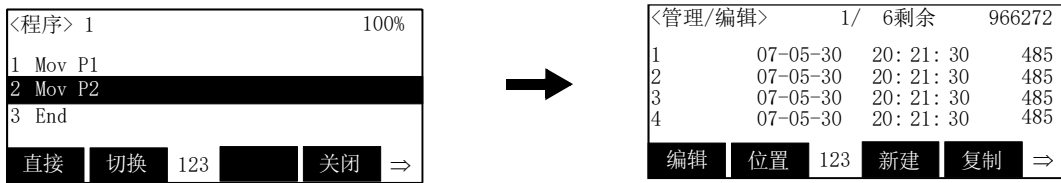
已省略时，将插入到光标的下一行。

◇◆◇可自动转换大小写字母◇◆◇

在 MELFA BASIC VI 中，预定语及变量名由大小写字母组合显示。从示教单元输入时，即使保持大写字母（小写字母）的状态，也会在确认行时自动转换。

(3) 程序制作的结束、保存

按下对应“关闭”的功能键，保存程序并结束制作。
未显示“关闭”时，按下 [FUNCTION] 键显示。



程序的结束和保存 [F4]

 注意

如果在程序编辑过程中切断电源，包含位置数据的编辑内容将被撤消，请特别注意。此外，应尽量也保存在计算机里进行备份。建议在配件的电脑支持软件中进行程序管理。

(4) 程序的修正方法

进行程序的修正前，请先参照本说明书 / 第 40 页的“3.6.1 程序的制作”的「(1) 程序编辑画面的开启」再开启程序的编辑画面。

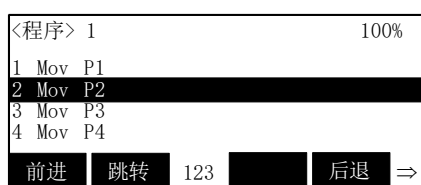
下面以将单步 5 的“Mov P5”修正为“Mvs P5”为例进行操作说明。



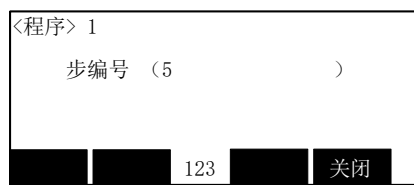
1) 单步 5 的调用。

按下对应“JUMP”的功能键，在显示的单步编号输入画面中按下 [5]、[EXE] 键。光标移动到单步 5。

通过 [↑]、[↓] 键将光标移到单步 5，也可调用单步 5。



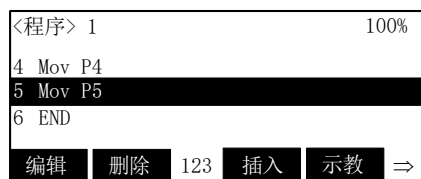
单步5的调用 [F2]



单步5的调用 [5] [EXE]

2) 指令语的修正。

按下对应“编辑”的功能键。

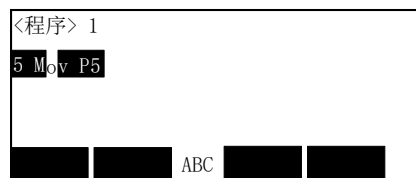
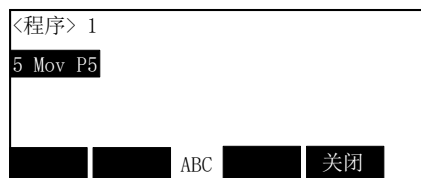


指令语的修正 [F1]



3) 按 3 次 [→] 键，将光标移到“o”。

按 3 次 [TUV] 键 (“v”的输入)、1 次 [→] 键、4 次 [PQRS] 键 (“s”的输入)，将变成“5 Mvs P5”，按下 [EXE] 键进行确定。

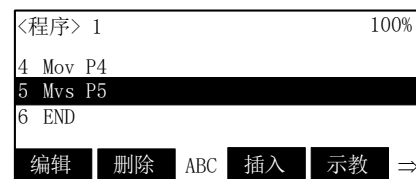


[TUV] [TUV] [TUV] [→]

指令语的修正 [PQRS] [PQRS] [PQRS] [PQRS]



指令语的修正 [EXE]



◇◆◇对齐光标进行修正◇◆◇

也可通过 [↑]、[↓] 键，将光标移到单步 5，按下对应“编辑”的功能键进行修正。

◇◆◇中止修正◇◆◇

在修正途中按下对应“关闭”的功能键，可中止修正。

◇◆◇修正文字◇◆◇

将光标移动到错误的文字之上，按下 [CLEAR] 键，向左边删除后再输入正确的文字。

◇◆◇程序修正之后◇◆◇

程序修正之后，务必进行保存操作（按下对应“关闭”的功能键 [F4] 或按下示教单元背面的 [ENABLE] 开关，将示教单元设为无效），并对内容进行单步运行，确认已正确修正。

(5) 当前位置的示教

可通过 JOG 操作等使机器人向作业位置移动，将此位置示教（登录）为程序中正在使用的位置变量。已经示教完毕的情况下可覆盖（修正）。分为在指令编辑画面中的示教和在位置编辑画面中的示教。

(a) 在指令编辑画面中的示教

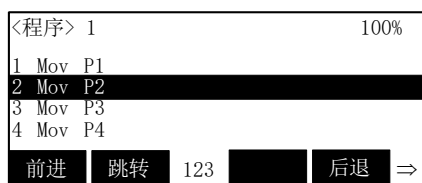
调用使用示教位置变量的单步。

在单步 5 “Mvs P5” 的位置变量 P5 示教当前位置时的操作步骤如下所示。预先通过 JOG 操作等，将机器人移动到作业位置。

1) 单步 5 的调用。

按下对应“JUMP”的功能键，在显示的单步编号输入画面中按下 [5]、[EXE] 键。光标移动到单步 5。

通过 [↑]、[↓] 键将光标移到单步 5，也可调用单步 5。



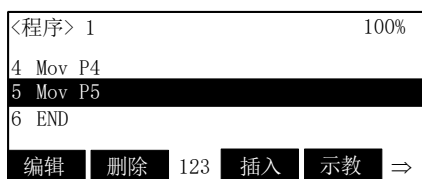
单步5的调用 [F2]



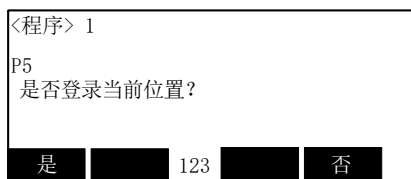
单步5的调用 [5] [EXE]

2) 当前位置的示教

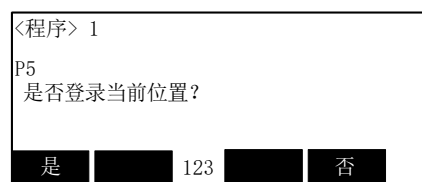
按下对应“示教”的功能键 ([F4])，显示确认画面。



当前位置的登录 [F4]



3) 按下对应“是”的功能键，机器人的当前位置数据将在 P5 示教，返回到原来的指令编辑画面。按下对应“否”的功能键 ([F1])，可中止示教。



当前位置的登录 [F1]



至此，当前位置的示教操作完成。

◇◆◇只有一个位置变量为对象◇◆◇

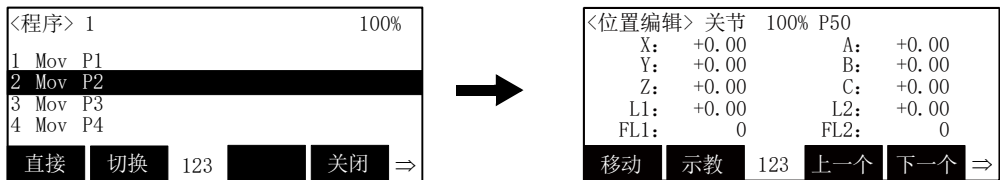
读出的单步使用“Mov P1+P2”及“P1=P10”等多个位置变量时，最左侧的位置变量变为示教的对象。此外，像“Mov p1+P2”一样大小写字母的位置变量同时存在时，大写字母的位置变量变为示教的对象。示教其他变量时，请参照下一页的“（b）在位置编辑画面的示教”。

(b) 在位置编辑画面的示教

向位置变量 P5 示教当前位置时的操作步骤如下所示。预先通过 JOG 操作等，将机器人移动到作业位置。

1) 位置编辑画面的显示

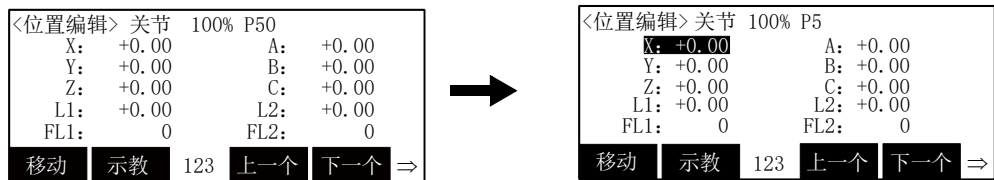
在指令编辑画面按下对应“切换”的功能键（[F2]），显示位置编辑画面。



位置编辑的显示 [F2]

2) P5 的调用。

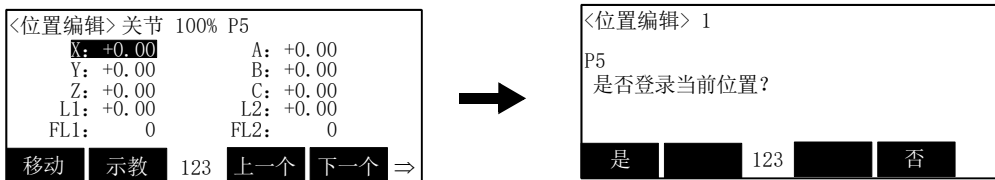
按下对应“Prev”、“Next”的功能键（[F3]）、（[F4]），调用“P5”。



单步5的调用 [F3] [F4]

3) 当前位置的示教

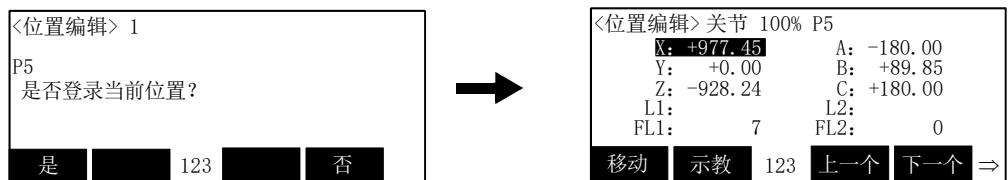
按下对应“示教”的功能键（[F2]），显示确认画面。



单步5的调用 [F2]

4) 按下对应“是”的功能键，机器人的当前位置数据将在 P5 示教，返回到原来的位置编辑画面。P5 中显示机器人的当前位置数据。

按下对应“否”的功能键（[F1]），可中止示教。



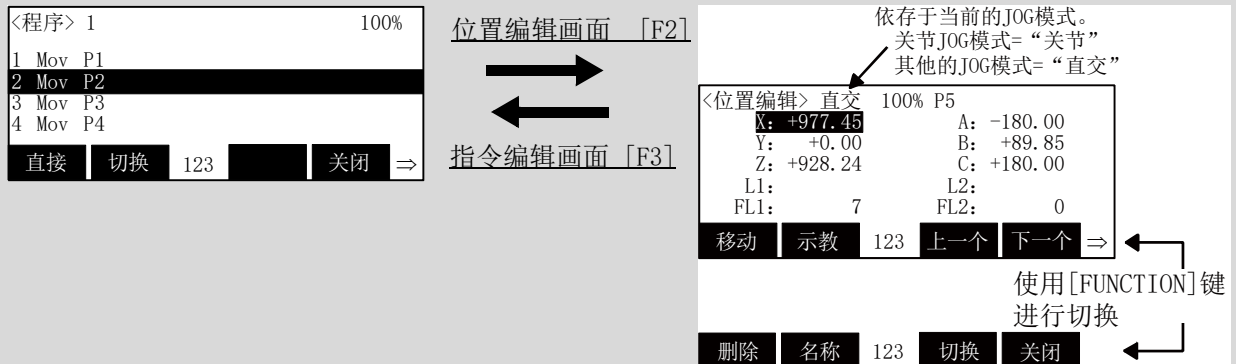
当前位置的登录 [F1]

至此，当前位置的示教操作完成。

◇◆◇指令编辑画面与位置编辑画面的切换◇◆◇

按下对应“切换”的功能键（F2），可相互切换指令编辑画面和位置编辑画面。

画面未显示“切换”时，按下 [FUNCTION] 键，将显示“切换”。菜单的右端显示“⇒”时，表示按下 [FUNCTION] 键即可切换菜单的状态。



可通过上一个（F3），下一个（F4）依次调用前后的位置变量。通常仅可调用位置变量，但可切换功能键，用名称（F2）调用位置变量以及关节变量。调用关节变量后，可通过上一个（F3），下一个（F4）依次调用前后的关节变量。

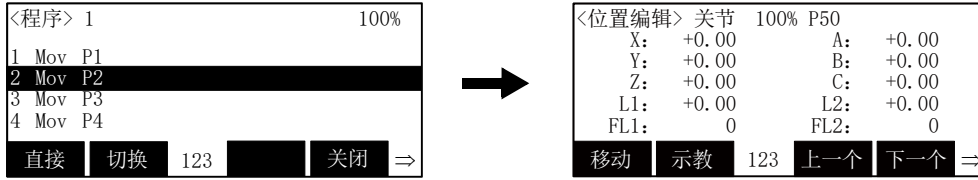
关节变量显示到起始或最后的值时，下一个显示将返回位置变量。

(6) 位置变量的删除

删除位置变量的操作步骤如下所示。
 仅可删除程序中未使用的变量。

1) 位置编辑画面的显示

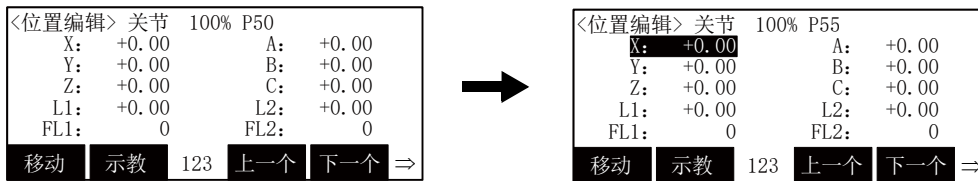
在指令编辑画面按下对应“切换”的功能键（[F2]），显示位置编辑画面。



位置编辑画面的显示 [F2]

2) 删除的位置变量的显示

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

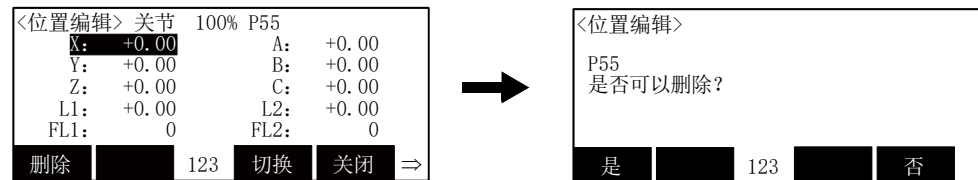


单步5的调用 [F3] [F4]

要删除P55时

3) 位置变量的删除

按下对应“删除”的功能键（[F1]），显示确认画面。
 （未显示“删除”时，按下[FUNCTION]键即可显示）

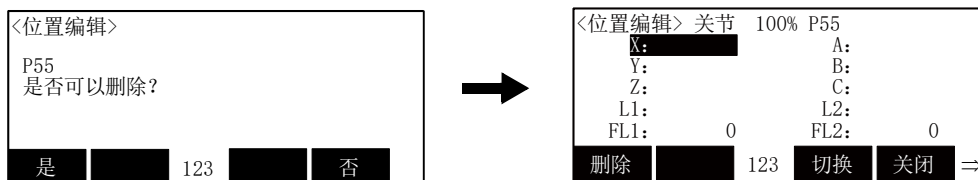


单步5的调用 [F1]

当前位置的登录 [F1]

4) 位置变量的删除

按下对应“是”的功能键（[F1]），位置变量将被删除。



单步5的调用 [F1]

当前位置的登录 [F1]

(7) 位置数据的确认

可将机器人移动到示教的位置变量的位置。

在移动模式中存在关节模式和直交模式，如下所示依存于当前设定的 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	通过关节插补移动到画面显示的位置变量的位置。 JOG 模式为关节模式时，将变为此移动方法。 变为和 Mov 指令相同的动作。
直交模式	通过直线插补移动到指定位置数据的位置。因此如果和指定的当前位置的构造标志不同时无法移动。JOG 模式为直交、3 轴直交、圆筒及工具 JOG 时，将变为此移动方法。 变为和 Mvs 指令相同的动作。

操作方法如下所示。

在伺服 ON 后轻按住有效开关（3 位置开关），维持伺服 ON 的状态下进行本操作。

1) 预先显示希望移动的位置变量。

按下对应“移动”的功能键，持续按键期间，机器人将移动至显示的位置变量的位置。

放开对应“移动”的功能键，机器人将停止。此外，放开有效开关（3 位置开关），或更强力按下时，伺服 OFF，机器人将停止。

位置编辑		关节	30% P1
X:	+977.45	A:	-180.00
Y:	+0.00	B:	+89.85
Z:	+928.24	C:	+180.00
L1:		L2:	
FL1:	7	FL2:	0
移动	示教	123	上一个 下一个 ⇒

位置编辑画面的显示 [F1]

注意

通过本操作使机器人动作。

即使机器人向画面上显示的位置变量的位置移动，也要先确认与周边装置等没有干涉后再进行。

建议一开始将速度放慢进行。此外，应在预测移动模式（关节、直交）下机器人动作轨迹的同时进行。

◇◆◇动作速度◇◆◇

按下 [OVRD ↑] 键（速度调快）、[OVRD ↓] 键（速度调慢），可调整移动速度。

(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中直接输入数值登录的方法。

在合计相同位置数据以作为从基准位置起的相对移动量 (差分) 所使用的位置变量的登录、及对已登录完毕的位置数据进行微调等情况下是有效的登录方法。

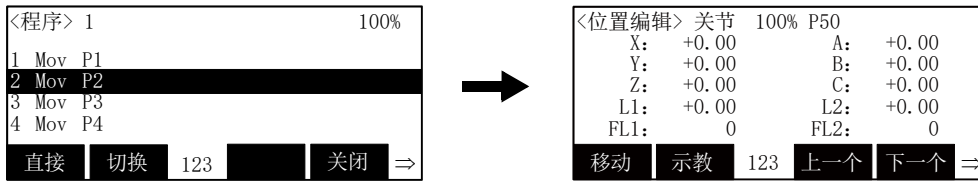
参考) 作为相对移动量的位置数据

例) 将P1作为基准位置, 由此开始在50mm上空, 通过关节插补移动时, P1通过当前位置的示教登录, 使用P50在Z轴成分中将“50.00”、其他轴成分中将“0.00”进行MDI登录的位置变量, 可执行Mov P1+P50。

上述示例中通过MDI登录P50的操作方法如下所示。

1) 位置编辑画面的显示

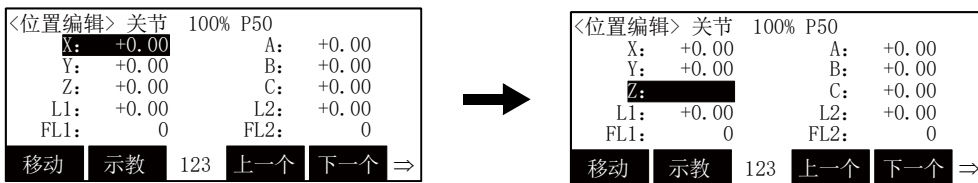
在指令编辑画面按下对应“切换”的功能键, 显示位置编辑画面。



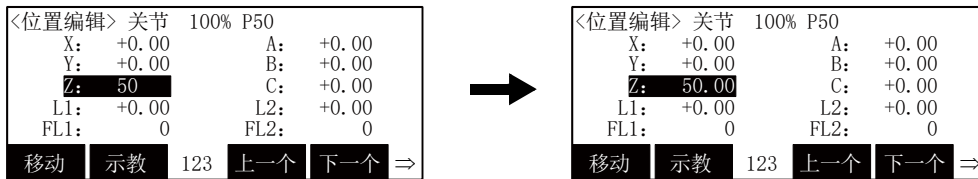
位置编辑画面的显示 [F2]

2) 在Z轴输入“50”。

按2次[↓]键, 使光标移动至Z轴。长按[CLEAR]键, 删除显示的“+0.00”。按下[5]、[0]、[EXE]键。仅登录位置变量P50Z轴的值50mm。



光标移动和值的清除 [↓] [CLEAR]



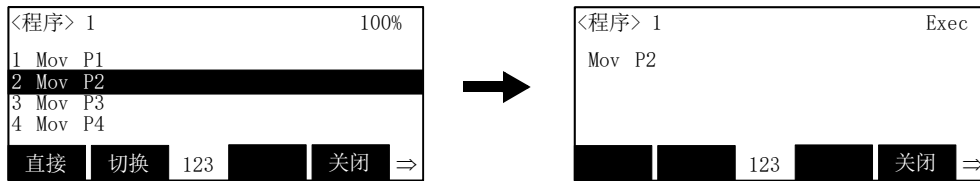
值50的输入 [5] [0] [EXE]

(9) 直接执行

所谓直接执行是指直接执行已输入的指令语句的方法。
在程序的编辑中也可确认机器人的动作。
直接执行 Mov 指令时的操作步骤如下所示。

1) 直接执行画面的显示

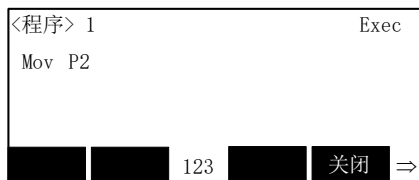
在指令编辑画面按下对应“直接”的功能键（[F1]），显示直接执行画面。



直接执行画面的显示 [F1]

2) 根据需要变更、执行指令

变更指令时，按下 [→] 键将光标移到变更位置后进行编辑。
完成编辑后按下 [EXE] 键。按住 [EXE] 键的期间，执行指令。
放开 [EXE] 键后，停止执行指令。



3.7 排错操作

所谓排错是指对已创建的程序是否正常动作进行测试的同时，在异常时修正该错误的作业中，可通过示教单元的排错功能进行确认。可执行的排错功能如下所示。已创建程序时，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，逐个单步执行程序（单步运行）并确认动作。

进行排错操作前，请参照第 40 页的“3.6.1 程序的制作”中的第 40 页的“(1) 程序编辑画面的开启”，显示排错对象程序的编辑画面。

在机器人的动作指令中，伺服 ON 操作后保持轻按住示教单元的有效开关（3 位置开关）不变，以在维持伺服 ON 的状态下执行。

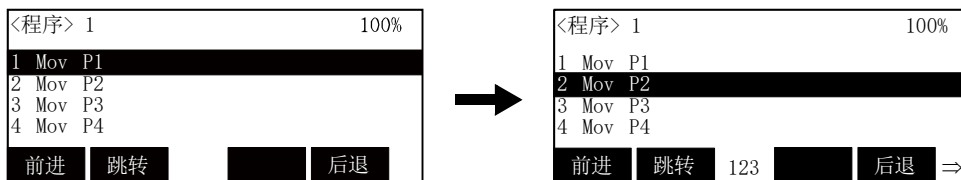
在 CR800 系列控制器的 S/W 版本为 Ver. D1 以上的情况下，进行单步执行时，无论机器人的动作 / 非动作指令如何，都应在轻按此开关的状态下进行操作。

1) 单步前进的执行

按下对应“前进”的功能键（[F1]）时，仅在持续按住的期间执行光标所在的单步。

中途松开功能键时，将中断执行。此外，通过在机器人移动中松开或更强力按下有效开关（3 位置开关），可以使伺服 OFF 以中断执行。

执行中操作面板的 [START] 开关的 LED 将亮灯。1 单步的执行完成后，[START] 开关的 LED 将熄灯，[STOP] 开关的 LED 将亮灯。松开 [F1] 键时，示教单元画面的光标移动至下一个单步。



单步前进 [F1]

每次按下对应“前进”的功能键时，都将前进至下一个单步。

⚠ 注意

动作中应充分注意机器人的动向。在发生与周边装置的干涉等异常时，应松开对应“前进”的功能键，或者松开有效开关（3 位置开关），或强力按下以使机器人停止。

⚠ 注意

根据当时的指定速度，机器人可能会以不同的轨迹动作。

尤其是弯角部分，其内环距离有可能会发生变化，因此，在开始自动运行时应先以低速运行，在留意未与周边装置发生干涉的同时逐渐提升运行速度。

◇◆◇所谓单步运行◇◆◇

逐个单步执行程序称为单步运行。由于动作速度缓慢，每执行 1 单步都将停止，因此对程序及动作位置的确认作业有效。

执行光标所在的单步。

执行 End 指令及 Hlt 指令时，单步前进不会再前进。

◇◆◇执行单步的变更◇◆◇

通过箭头键移动光标及跳转操作（“跳转”），可变更执行单步。

◇◆◇要立即停止动作中的机器人时◇◆◇

①按下 [EMG. STOP] 开关。

伺服 OFF 后，发生紧急停止错误使动作中的机器人立即停止。
之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

②松开或更强力按下有效开关（3 位置开关）。

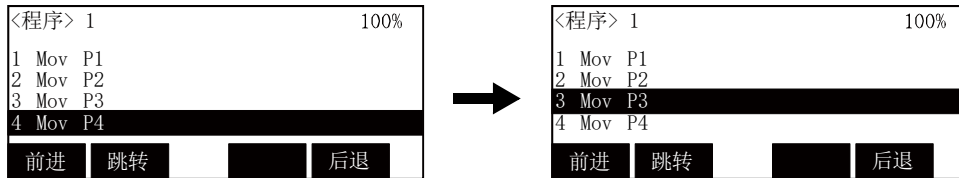
伺服 OFF 后，发生错误 2000，动作中的机器人立即停止。
之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

(2) 单步返回

将通过单步前进及自动运行结束执行的单步或中断的单步，在仅限插补指令下逐个单步向返回方向执行。此外，最多可返回 4 个单步。

1) 单步返回的执行

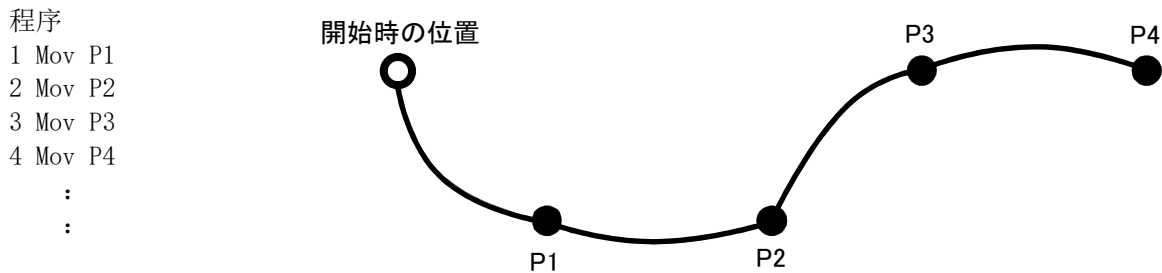
按下对应“后退”的功能键时，仅在持续按住期间向单步的返回方向仅执行 1 个单步。中途松开功能键时，将中断执行。此外，通过在机器人移动中松开或更强力按下有效开关（3 位置开关），可以使伺服 OFF 以中断执行。执行中操作面板的 [START] 开关的 LED 将亮灯。1 单步的执行完成后，[START] 开关的 LED 将熄灯，[STOP] 开关的 LED 将亮灯。示教单元画面的光标在单步返回方向，移动至下一个插补指令的单步。



单步返回 [F4]

每次按下对应“后退”的功能键时，都将返回至前一个单步。

[补充] 将如下程序单步前进到单步 4 后再进行单步返回时，返回到 P1 后再进行单步返回后，将返回到执行单步前进开始时的位置。（所谓开始时的位置是指开始执行单步 1 的位置）



注意

动作中应充分注意机器人的动向。在发生与周边装置的干涉等异常时，应松开对应“后退”的功能键，或者松开有效开关（3 位置开关），或强力按下以使机器人停止。

◆◆◆要立即停止动作中的机器人时◆◆◆

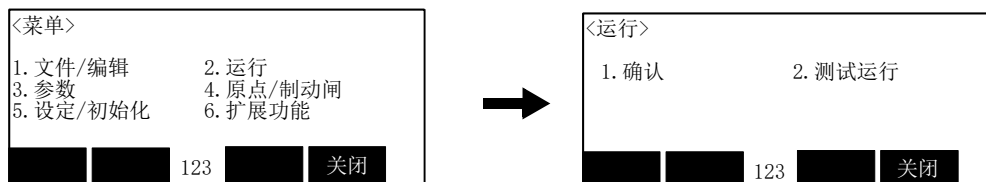
- ①按下 [EMG. STOP] 开关。
伺服 OFF 后，发生紧急停止错误，动作中的机器人立即停止。之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。
- ②松开或更强力按下有效开关（3 位置开关）。
伺服 OFF 后，发生错误 2000，动作中的机器人立即停止。之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

(3) 其他插槽中的单步前进

确认多任务的程序时，可在运行菜单的确认画面进行单步前进。

1) 运行菜单的选择

在菜单画面按下 [2] 键，选择“2. 运行”。

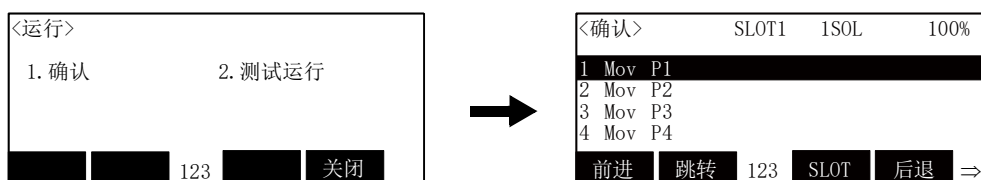


运行画面的选择 [2]

2) 确认画面的选择

在运行画面按下 [1] 键，选择“1. 确认”。

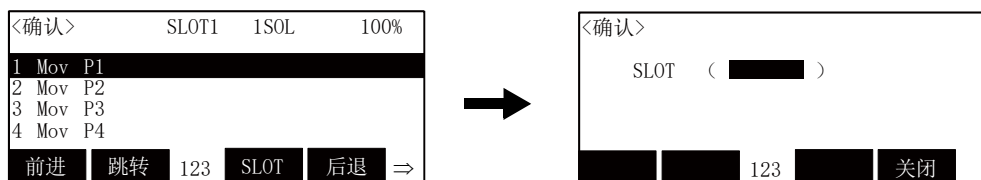
显示设定在插槽 1 中的程序。插槽编号上将连续显示程序名称。



运行画面的选择 [1]

3) 插槽的变更

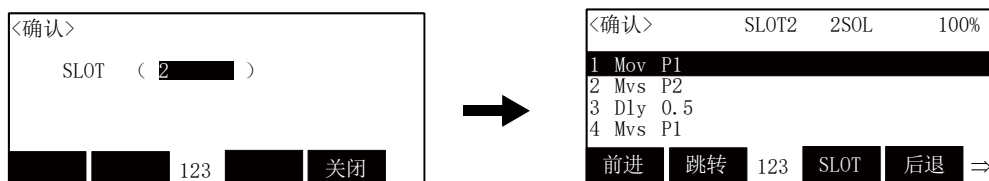
按下对应“SLOT”的功能键 ([F3])，则显示插槽编号指定画面。



插槽的变更 [F3]

输入希望的插槽编号后再按下 [EXE] 键。

显示输入的插槽编号的程序。(指定了插槽 2 的情况如下所示)



插槽的变更 [2] [EXE]

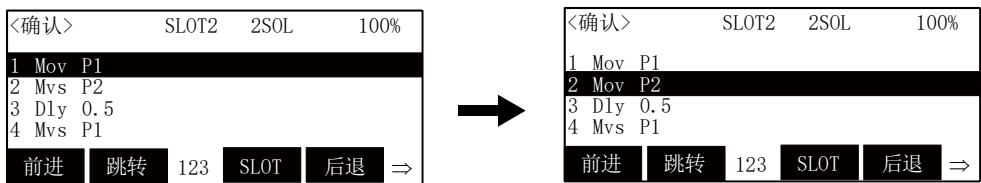
4) 单步运行的执行

单步前进和单步返回可执行与在指令编辑画面中的单步运行相同的操作。

按下对应“前进”的功能键 ([F1])，执行单步前进，按下对应“后退”的功能键 ([F4])，执行单步返回，仅在持续按下各功能键期间，执行光标所在的单步。

中途松开功能键时，将中断执行。此外，通过在机器人移动中松开或更强力按下有效开关 (3 位置开关)，可使伺服 OFF 以中断执行。

执行中操作面板的 [启动] 开关的 LED 将亮灯。1 单步的执行完成后，[启动] 开关的 LED 将熄灯，[停止] 开关的 LED 将亮灯。示教单元画面的光标将移动至下一个单步。



单步前进 [F1] (单步返回 [E4])

⚠ 注意

动作中应充分注意机器人的动向。在发生与周边装置的干涉等异常时，应松开对应“前进”或“后退”的功能键，或者松开有效开关（3 位置开关），或强力按下以使机器人停止。

◇◆◇执行单步的变更◇◆◇

通过箭头键移动光标及跳转操作（“跳转”），可变更执行单步。
请参照后述的[第 58 页的“（4）单步跳转”](#)。

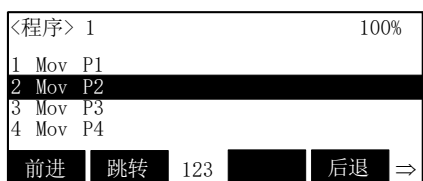
(4) 单步跳转

可变更进行单步运行的单步编号。

从单步 5 开始进行单步运行时的操作示例如下所示。

1) 单步 5 的调用。

按下对应“跳转”的功能键（[F2]），在显示的单步编号输入画面中按下 [5]、[EXE] 键。光标移动到单步 5。



单步5的调用 [F2]

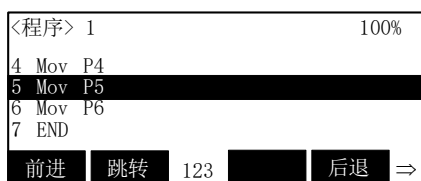


单步5的调用 [5] [EXE]

即使通过 [↑]、[↓] 键将光标移到单步 5，也可调用单步 5。

2) 单步前进

和上述的[第 53 页](#)的“(1) 单步前进”相同，按下对应“前进”的功能键（[F1]），则可从单步 5 开始执行单步前进。



单步前进 [F1]



⚠ 注意

变更所执行单步进行单步运行时，应预先确认机器人和周边装置等未发生干涉。此外，动作中应充分注意机器人的动向，在发生与周边装置的干涉等异常时，应松开对应“前进”或“后退”的功能键，或者松开有效开关（3 位置开关），或强力按下以使机器人停止。

3.8 自动运行

3.8.1 动作速度的设定

机器人的动作速度的设定，在 T/B（示教单元）或控制器的 O/P（操作面板）中进行。但是，仅 CR860-D/R/Q 支持 O/P 的操作。自动运行时的实际动作速度为，
 动作速度 = （控制器或示教单元的设定值）× （程序的设定值）。

(1) 示教单元中的操作

- 1) 每次按下 UP 按钮，倍率修调将按照（10 → 20 → 30 → 40 → 50 → 60 → 70 → 80 → 90 → 100%）的顺序增加。按下 DOWN 按钮时将按照相反顺序减少。

根据各设定开关的状态，可执行的机器人的操作和其动作速度如下表所示。

设定开关	控制器	控制器的模式 (模式选择开关)	AUTOMATIC	MANUAL
	示教单元	TB ENABLE 开关	OFF	ON
有效开关 (3 位置有效开关)				
从示教单元可执行的机器人操作			自动运行 (程序的执行)	
动作速度			程序指定速度 (最高 100%)	安全速度 (250mm/s) 以下

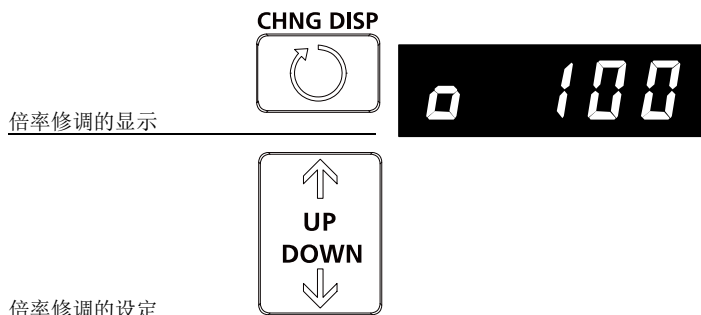
从示教单元可执行的自动运行以外的机器人操作中，有 JOG 运行、单步运行、位置跳转。这些操作在设定开关为下表所示的状态下可进行。动作速度为安全速度（250mm/s）以下。

设定开关	控制器	控制器的模式 (模式选择开关)	MANUAL
	示教单元	TB ENABLE 开关	ON
有效开关 (3 位置有效开关)			
动作速度			安全速度 (250mm/s) 以下

⚠ 注意

在 MANUAL 模式下使行走轴上的机器人运行时，应控制包括行走轴在内的机器人的运行速度（合成速度），使其不超过安全速度（250mm/s）。
 关于行走轴的速度设定，请参照另一手册“附加轴功能使用说明书”。

(2) 操作面板中的操作（仅限 CR860-D/R/Q）



- 1) 按两次控制器的显示切换按钮（CHNG DISP），显示面板（STATUS. NUMBER）上将显示“倍率修调”。
- 2) 每次按下 UP 按钮，倍率修调将按照（10 → 20 → 30 → 40 → 50 → 60 → 70 → 80 → 90 → 100%）的顺序增加。按下 DOWN 按钮时将按照相反顺序减少。

3.8.2 自动运行的开始

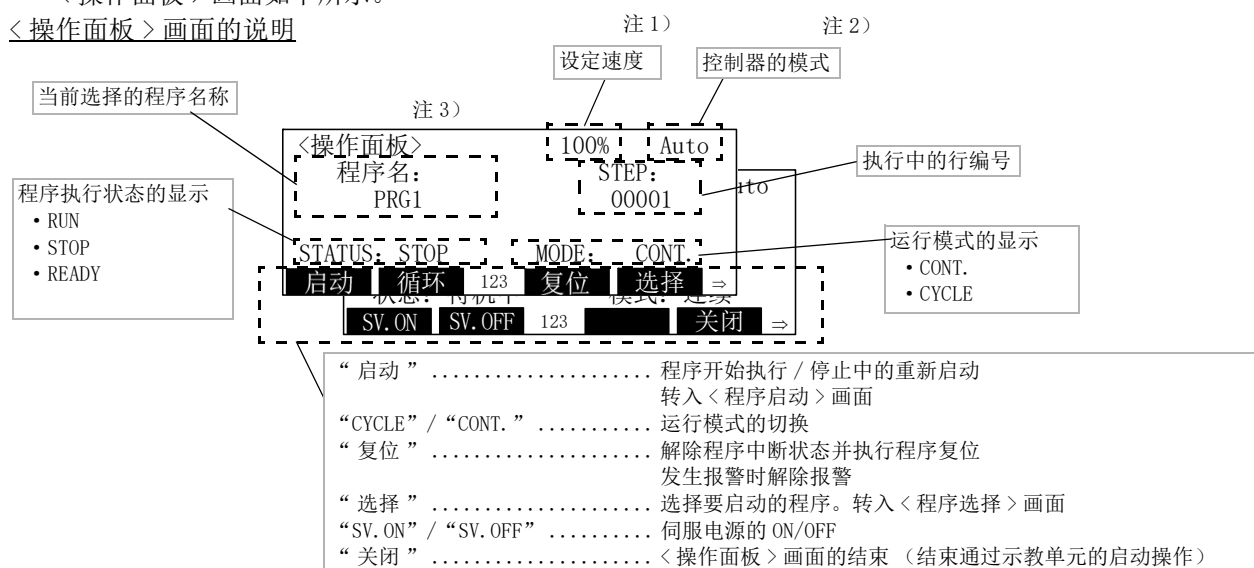
(1) 示教单元中的操作

操作通过〈菜单〉→〈运行〉中可选择的〈操作面板〉画面进行。

通过参数：TBOP 的设定也可将本功能设为无效。

〈操作面板〉画面如下所示。

〈操作面板〉画面的说明



注 1) 控制器的模式为“MANUAL”时，即使设定速度为 100%，也会变为低速动作。（用于动作确认）

注 2) 控制器的模式为“AUTOMATIC”时，示教单元的状态显示灯 [ENABLE] 闪烁，表示示教单元处于有效状态。结束〈操作面板〉画面，则示教单元返回到无效状态。

注 3) 〈操作面板〉画面显示中，[JOG]、[HAND]、[MONITOR] 键变为无效。

通过示教单元的〈操作面板〉画面可执行的功能、及显示〈操作面板〉画面时可执行的操作如下所示。

通过〈操作面板〉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，有效开关无需 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，机器人以用户指定的速度动作。控制器的模式设定为“MANUAL”时，机器人以安全速度以下动作（用于动作确认）。请将控制器切换到希望的模式。
- ②伺服 ON [SERVO] 键
(控制器的模式为“AUTOMATIC”时，有效开关无需 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。

⚠ 注意

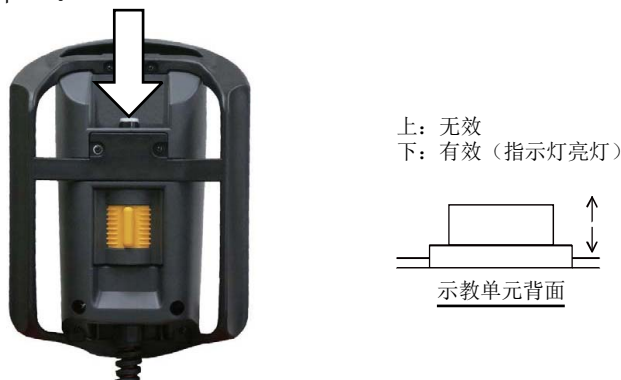
执行自动运行前，请务必确认下列项目。没有确认而执行自动运行的情况下，有可能造成物品损失及人身事故。

- 开始自动运行前，请确认安全防护区域内（安全栅栏内）没有人员。
- 请务必从安全防护区域外（安全栅栏外）进行自动运行的开始操作。
- 开始自动运行前，请务必确认作业内容与选择程序正确一致。
- 自动运行过程中对程序进行中断，并通过 MANUAL 模式对程序内容或位置数据进行修改时，务必进行一次确认运行之后再重启自动运行。

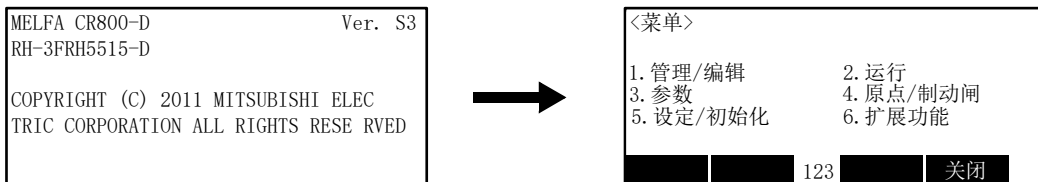
- 请根据自动运行中动作中的机械臂的离心力，或根据执行紧急停止时的停止时的冲击力，避免夹持的工件及工具意外放出或飞散的危险性。
- 自动运行中由于程序错误等机器人意外以高速与工件及周边装置发生冲突时，工件及周边装置、机器人机械臂及该减速机及传达机构等可能会破损。

以下对通过示教单元执行自动运行时的操作方法进行说明。

- 1) 按压示教单元 (T/B) 背面的 TB ENABLE 开关，将示教单元操作设为无效 (TB ENABLE 开关熄灯)，将用户准备的控制器的模式选择开关设为“**AUTOMATIC**”。保持示教单元的 TB ENABLE 为有效的状态，将模式选择开关切换为 **AUTOMATIC** 时，将发生错误 H5000 “**示教作业中**”。

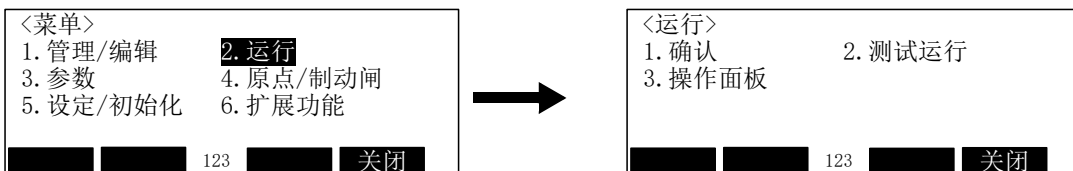


- 2) 在显示了标题画面的状态下按压 [EXE] 键，显示 < 菜单 > 画面。



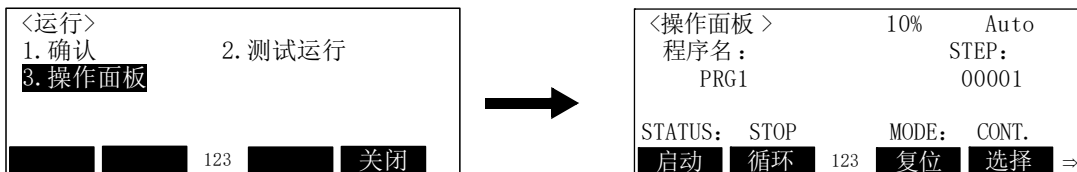
菜单画面的显示 [EXE]

- 3) 在菜单画面中按 [2] 按键，显示 < 运行 > 菜单画面。



运行菜单的选择 [2]

- 4) 在运行菜单画面中按下 [3] 键，将显示 < 操作面板 > 画面。



操作面板的选择 [3]



示教单元前面的状态显示灯“ENABLE”闪烁。

5) 请按压 [OVRD ↑] 键、[OVRD ↓] 后设定为安全的速度。

<操作面板>	100%	Auto
程序名:	STEP:	
PRG1	00001	
STATUS: STOP	MODE: CONT.	
启动	循环	123 复位 选择 ⇒



<操作面板>	10%	Auto
程序名:	STEP:	
PRG1	00001	
STATUS: STOP	MODE: CONT.	
启动	循环	123 复位 选择 ⇒

速度的设定 [OVRD ↑]、[OVRD ↓]

将速度更改为 10% 时

6) 在操作面板画面中按压 [F4] 键 (“选择”) 后, 将显示 <程序的选择> 画面。

<操作面板>	10%	Auto
程序名:	STEP:	
PRG1	00001	
STATUS: STOP	MODE: CONT.	
启动	循环	123 复位 选择 ⇒



<程序选择>		
程序名		
(PRG1)		
		123 关闭

程序的选择 [F4]

显示当前选择的程序名称。

7) 在程序名的括号内输入要启动的程序名称, 按下 [EXE] 键。

显示新的程序, 返回到 <操作面板> 画面。

※ 请务必确认已选择作为目的的程序编号。

※ 程序执行状态的显示为 “停止中” 时无法进行程序的更改。请按压 [复位] (F3 键), 将状态显示设为 “待机中” 后重新操作。

<程序选择>		
程序名		
(PRG2)		
		123 关闭



<操作面板>	10%	Auto
程序名:	STEP:	
PRG2	00001	
STATUS: STOP	MODE: CONT.	
启动	循环	123 复位 选择 ⇒

程序名称的输入, [EXE] 键

图示为将程序名变更为 “PRG2” 的示例。

8) 未显示 “SV.ON” 的功能时, 按压 [FUNCTION] 键切换功能菜单显示部。

<操作面板>	10%	Auto
程序名:	STEP:	
PRG2	00001	
STATUS: STOP	MODE: CONT.	
启动	循环	123 复位 选择 ⇒



<操作面板>	10%	Auto
程序名:	STEP:	
PRG2	00001	
STATUS: STOP	MODE: CONT.	
SV.ON	SV.OFF	123 关闭 ⇒

功能菜单的切换 [FUNCTION]

9) 在进行伺服 ON 之前, 应确认机器人的动作范围内无人存在后, 按压 [F1] 键 (“伺服 ON”)。

※ 即使用示教单元的 [SERVO] 按钮也可进行伺服 ON。

※ 该操作无需握住 “有效开关”。

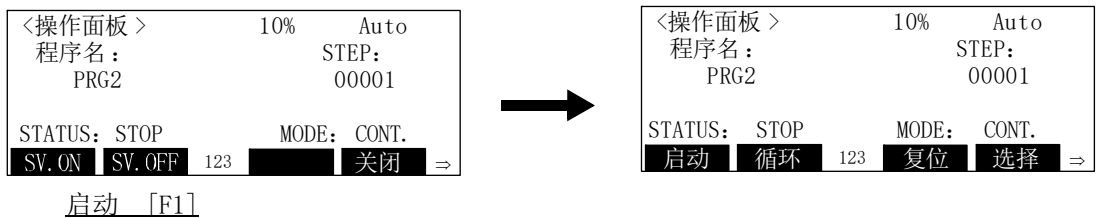
<操作面板>	10%	Auto
程序名:	STEP:	
PRG2	00001	
STATUS: STOP	MODE: CONT.	
SV.ON	SV.OFF	123 关闭 ⇒



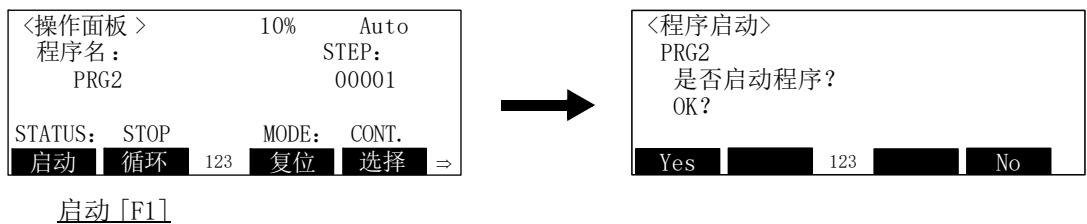
<操作面板>	10%	Auto
程序名:	STEP:	
PRG2	00001	
STATUS: STOP	MODE: CONT.	
SV.ON	SV.OFF	123 关闭 ⇒

SV.ON[F1]

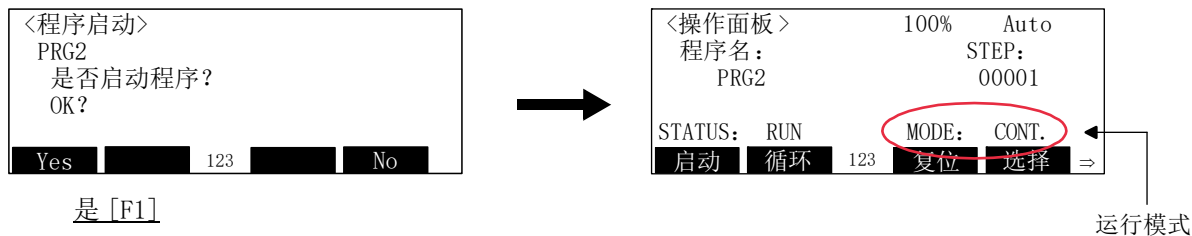
10) 按压 [FUNCTION] 键，切换功能菜单显示部。



11) 按压 [F1] 键 (“启动”), 显示确认画面。



12) 按压 [F1] 键 (“Yes”) 时, 开始当前选择的程序的自动运行。运行模式以画面上显示的模式为基准。



至此, 通过示教单元进行的自动运行的开始操作结束。

(2) 操作面板中的操作 (仅限 CR860-D/R/Q)

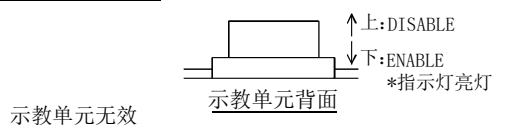
**注意**

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身事故。

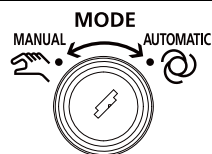
- 机器人的周围没有人员。
- 安全栅栏已上锁，防止人员不慎接近。
- 机器人的动作范围内没有放置工具等不需要的物品。
- 工件放置在正确的位置。
- 确认在单步运行下，程序在正确动作。

**注意**

使用示教单元打开操作面板时，操作面板中除可以进行停止按钮（STOP）的操作，其他任何按钮均无法进行操作。如果使用操作面板进行操作时，应将示教单元的画面切换至操作面板以外的画面后再进行操作。

控制器的准备

- 1) 确认示教单元的 [ENABLE] 开关为“DISABLE”。

控制器有效

- 2) 确认控制器的模式切换开关（MODE）为“AUTOMATIC”。

伺服 ON

- 3) 按下控制器的 SVO.ON 按钮，打开伺服电源。SVO ON 指示灯亮灯。

自动运行的开始

- 4) 下控制器的开始按钮（START），则开始自动运行。（连续运行）在连续运行的状态下按下结束按钮（END），则运行 1 循环后停止。在循环停止中 LED 灯会闪烁。在循环停止中再次按下结束按钮（END），则恢复连续运行。

- 5)

**注意**

开始自动运行时，请务必确认选择了目标程序名称。

**注意**

自动运行中请充分注意机器人的动作。

发现异常时，应按下紧急停止开关（EMG. STOP）使机器人立即停止动作。

**注意**

开始自动运行时，请先以低速状态动作，之后确认未与周边装置发生干涉的同时逐渐地提升速度。

3.8.3 停止

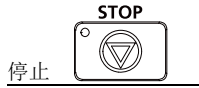
(1) 示教单元中的操作

使运行中的程序立刻停止，使移动中的机器人减速停止。

停止 [STOP]

1) 按下 [STOP] 键。

(2) 操作面板中的操作 (仅限 CR860-D/R/Q)



1) 按下停止按钮 (STOP)。

◇◆◇停止操作与操作权无关◇◆◇

停止操作与操作权的有无无关，将始终有效。

3.8.4 停止之后重新开启自动运行

(1) 示教单元中的操作

操作方法与自动运行的开始操作相同。第 60 页的“3.8.2 自动运行的开始”请参照，进行通过示教单元的重启操作。

(2) 操作面板中的操作 (仅限 CR860-D/R/Q)

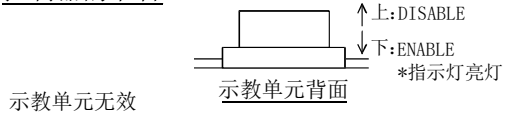


注意

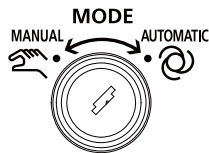
执行自动运行前，请务必确认下列项目。没有确认而执行自动运行的情况下，有可能造成物品损失及人身事故。

- 机器人的周围没有人员。
- 安全栅栏已上锁，防止人员不慎接近。
- 机器人的动作范围内没有放置工具等不需要的物品。
- 工件放置在正确的位置。
- 确认在单步运行下，程序在正确动作。

控制器的准备



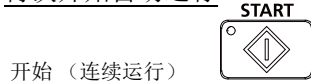
1) 应将示教单元的 [ENABLE] 开关置为“DISABLE”。



2) 将控制器的模式切换开关 (MODE) 置为“AUTOMATIC”。

控制器有效

再次开始自动运行



3) 按下控制器的开始按钮 (START)，则再次开始自动运行。(连续运行)
连续运行 /1 循环运行将保持之前的状态。



注意

开始自动运行时，请务必确认选择了目标程序名称。



注意

自动运行中请充分注意机器人的动作。
发现异常时，应按下紧急停止开关 (EMG. STOP) 使机器人立即停止动作。



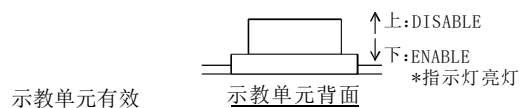
注意

自动运行中请勿切断机器人 CPU 的电源。否则可能导致机器人 CPU 内的存储器出现异常，从而造成程序损坏。要使机器人立即停止时，请使用紧急停止。

3.8.5 程序复位

(1) 示教单元中的操作

解除程序中中断状态，返回到执行行起始处。

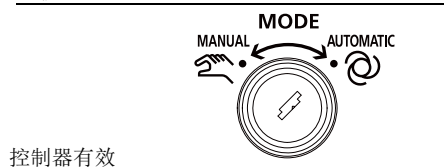
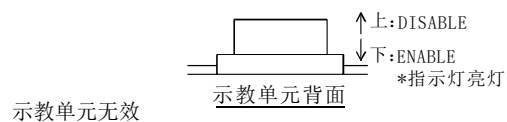


程序复位的执行

程序复位 [RESET] + [EXE]

(2) 操作面板中的操作 (仅限 CR860-D/R/Q)

控制器的准备



程序名称的显示



程序复位的执行



- 1) 将控制器的模式设为“MANUAL”。
- 2) 将示教单元的 [ENABLE] 开关设为“ENABLE”。

- 3) 按住 [RESET] 键的同时按下 [EXE] 键。执行行返回到起始处，程序将被复位。

- 1) 应将示教单元的 [ENABLE] 开关置为“DISABLE”。

- 2) 将控制器的模式切换开关 (MODE) 置为“AUTOMATIC”。

- 3) 按下控制器的显示切换按钮 (CHNG DISP)，以显示“程序名称”。

- 4) 按下控制器的复位按钮 (RESET)。STOP 指示灯熄灯，解除程序的中断状态。

◇◆◇程序复位仅在程序中中断中有效◇◆◇

程序运行中无法执行程序的复位操作。应在中断中进行操作。

此外，通过控制器的操作面板进行程序复位时，使显示面板 (STATUS. NUMBER) 中显示“程序名称”后再进行复位操作。

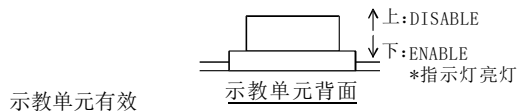
◇◆◇STOP 指示灯熄灯◇◆◇

进行了程序复位，则 STOP 指示灯熄灯。

3.9 伺服ON/OFF

(1) 示教单元中的操作

伺服电源在示教模式中为了安全，仅在轻按示教单元背面的有效开关（3 位置开关）的期间为 ON。示教单元中的操作应在轻按有效开关（3 位置开关）的状态下进行。



伺服 ON 的执行

伺服 ON 操作 [SERVO]

伺服 OFF 的执行

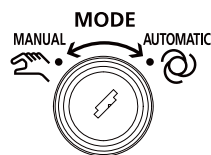
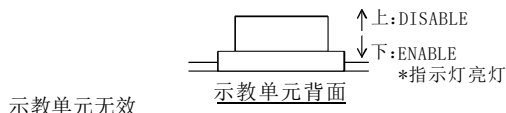
伺服 OFF 操作 有效开关（3 位置开关）
松开或强力按下

- 1) 将控制器的模式设为“MANUAL”。
- 2) 将示教单元的 [ENABLE] 开关设为“ENABLE”。

- 3) 按下 [SERVO] 键，伺服 ON。
- 4) 松开或更强力按下有效开关（3 位置开关），伺服 OFF。

(2) 操作面板中的操作（仅限 CR860-D/R/Q）

控制器的准备

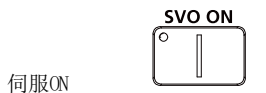


控制器有效

- 1) 应将示教单元的 [ENABLE] 开关置为“DISABLE”。

- 2) 将控制器的模式切换开关（MODE）置为“AUTOMATIC”。

伺服 ON 的执行



伺服ON

- 3) 按下 SVO.ON 按钮后则伺服 ON，并且 SVO ON 指示灯亮灯。

伺服 OFF 的执行



伺服OFF

- 4) 按下 SVO.OFF 按钮后则伺服 OFF，并且 SVO OFF 指示灯亮灯。

◇◆◇制动闸运行◆◆◇

伺服 OFF 时，制动闸将自动运行。根据机器人的机型，也存在无制动闸的轴。

3.10 错误复位操作

(1) 通过示教单元进行错误复位操作

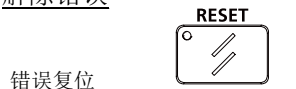
解除错误

错误复位 [RESET]

- 1) 按下示教单元的 [RESET] 键。

(2) 通过操作面板进行错误复位操作（仅限 CR860-D/R/Q）

解除错误



错误复位

- 1) 按下控制器的复位按钮（RESET）。
如果示教单元侧的错误未复位，则通过示教单元进行错误复位操作。

3.11 无法解除错误的暂时错误复位操作

根据机器人的机型，轴在动作范围外等情况下，有可能会发生无法解除的错误。此时，常规的操作下无法进行伺服 ON 及 JOG 操作。此时，按照下列步骤可暂时解除错误。动作范围外等情况下，暂时解除错误期间，应执行 JOG 操作，并移动到动作范围内。

此外，即使在使用干涉回避功能中，伺服 OFF 状态下由于人工导致机器人进入了干涉领域等情况下，也可通过按住 [RESET] 键进行 JOG 操作，解除干涉回避功能并使机器人运行。

此外，[RESET] 键相当于 R56TB 中的 [CAUTION] 键。使用 R56TB 时，应使用 [CAUTION] 键进行以下操作。

示教单元的准备



- 1) 将控制器的模式设为“MANUAL”。
- 2) 将示教单元的 [ENABLE] 开关置为“ENABLE”。
- 3) 按压 [JOG] 键打开 JOG 画面。
使用力觉功能时，即使是通过 JOG 画面打开的力觉功能画面，也可以一时性地进行错误复位操作。
- 4) 轻轻握住使能开关（3 位置开关）。
- 5) R32TB 的情况下持续按压 [RESET] 键，R56TB 的情况下持续按压 [CAUTION] 键。
- 6) 按下 [SERVO] 键使伺服 ON。
- 7) 通过 JOG 操作使机器人动作。

通过以上操作可暂时复位错误。

如果进行以下任意一项操作，则一时性的错误复位状态将变为无效。

- ① 松开 [RESET] 键（[CAUTION] 键）。
- ② 将示教单元的 [ENABLE] 开关设为“DISABLE”，将控制器的模式设为“AUTOMATIC”。
- ③ 打开 JOG 画面、力觉功能画面以外的画面。（仅限 R32TB）

如果一时性的错误复位状态无效时满足了发生错误的条件，则将再次发生错误。

通过 JOG 操作使机器人动作时，应在按下 [RESET] 键（[CAUTION] 键）的情况下进行操作。

注意

一时性的错误复位操作过程中进行的 JOG 操作，无法使机器人通过动作范围限制停止。请勿向动作范围外的方向移动机器人。

【参考】控制器 LED 的显示
 在控制器前面配置有显示控制器状态的 4 个 LED。

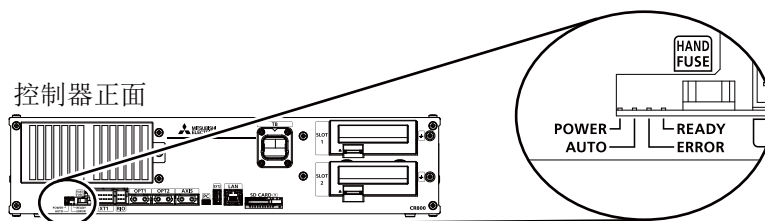


图 3-1：CR800-D/R/Q 控制器状态显示 LED

表 3-5：各 LED 的显示

LED	内容
POWER	显示控制电源的状态。 亮灯：控制电源 ON 熄灯：控制电源 OFF
AUTO	显示控制器的模式。 亮灯：AUTOMATIC 模式 熄灯：MANUAL 模式
ERROR	显示异常发生的状态。 亮灯：低级别错误、警告发生 快速闪烁：高级别错误发生 熄灯：正常动作中
READY	显示动作状态。 亮灯：控制器启动完成 慢速闪烁：运行中 快速闪烁：中断中

表 3-6：各功能的 LED 显示状态

控制器的状态	LED 显示状态		
低级别错误发生 警告发生	ERROR 亮灯	■ → ■ → ■ → ■	
高级别错误发生	ERROR 闪烁	■ → □ → ■ → □	快速闪烁。
SD 卡及 USB 存储器写入中	AUTO 闪烁	■ → □ → ■ → □	AUTO 和 READY 同时快速闪烁。
	READY 闪烁	■ → □ → ■ → □	
启动中 (从电源 ON 到启动完成为止)	AUTO 闪烁	□ → ■ → □ → ■	AUTO 和 READY 交替慢速闪烁。 启动完成时 1 秒内全部亮灯。
	READY 闪烁	■ → □ → ■ → □	
运行中	READY 闪烁	■ → □ → ■ → □	慢速闪烁。
中断中	READY 闪烁	■ → □ → ■ → □	快速闪烁。

3.12 管理・编辑画面的操作

在此，对关于程序管理的以下操作方法进行说明。

- (1) 程序的一览显示
- (2) 程序的复制（复制）
- (3) 程序的名称变更（重命名）
- (4) 程序的删除（删除）
- (5) 程序的保护（保护）
- (6) 程序的选择

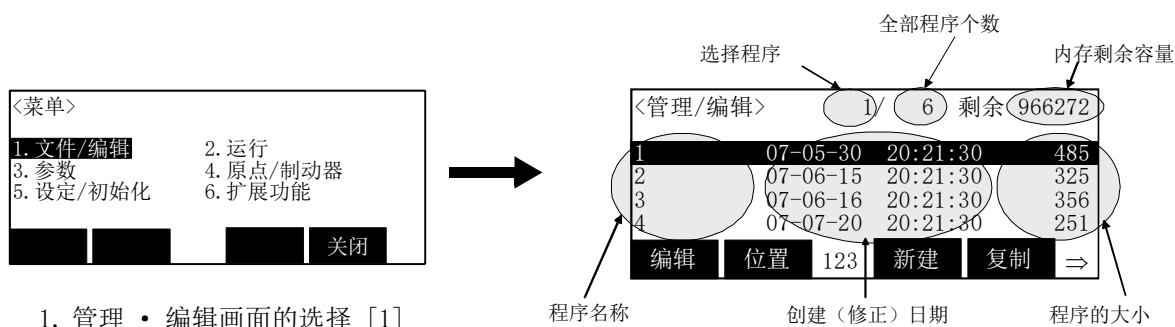
关于程序的编辑方法，请参照第 40 页的“3.6 程序编辑”。

(1) 程序的一览显示

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。



通过 [↑]、[↓] 键，将光标对准“1. 管理・编辑”后按下 [EXE] 键也可进行相同的操作。

在程序一览画面按下 [↑]、[↓] 键，将显示其他程序。

此外，也可预先选择作为各操作对象的程序。

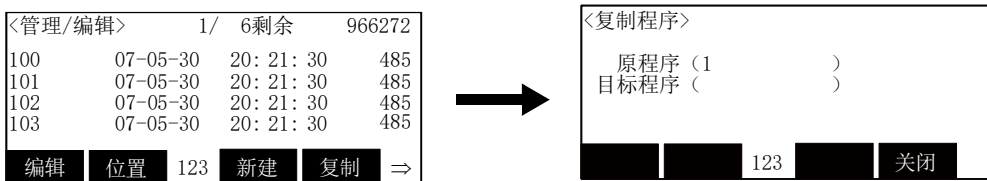
画面的下方显示对应功能键的菜单（“编辑”、“位置”、“新建”、“复制”）。其中，“编辑”、“位置”、“新建”为程序的编辑功能。关于这些操作方法请参照第 40 页的“3.6 程序编辑”。

按下 [FUNCTION] 键，将显示“重命名”、“删除”、“保护”、“关闭”。

(2) 程序的复制（复制）

1) 复制菜单的选择

在程序一览显示中按下对应“复制”的功能键（[F4]）。显示复制画面。

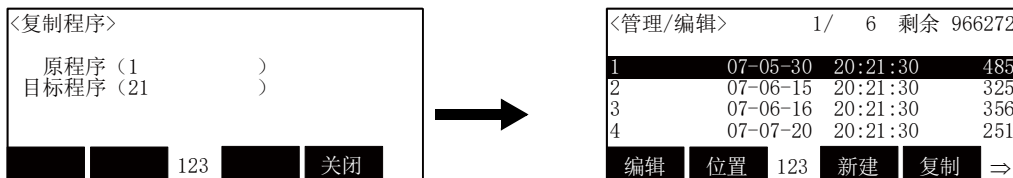


复制的选择 [F4]

2) 要复制的程序的指定和执行

在原程序的括号内，将在程序一览显示中预先显示已选择的程序名称。（图中程序名称“1”）要变更时，可通过箭头键移动光标进行变更。

在目标程序的括号内输入被复制的程序名称，按下 [EXE] 键。



程序的名称 [2] [1] [EXE]

此时，将程序“1”复制到“21”。

◆◆◆保护（保护）设定无法被复制◆◆◆

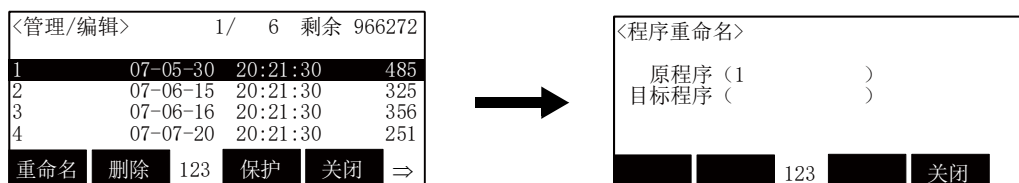
复制操作中，程序保护信息及变量保护信息将无法被复制。

应根据需要再次设定。保护（保护）的设定方法请参照第 74 页的“（5）程序的保护（保护）”。

(3) 程序的名称变更（重命名）

1) 重命名菜单的选择

在程序一览显示中按下对应“重命名”的功能键（[F1]）。显示复制画面。未显示“重命名”菜单时，按下 [FUNCTION] 键使其显示。

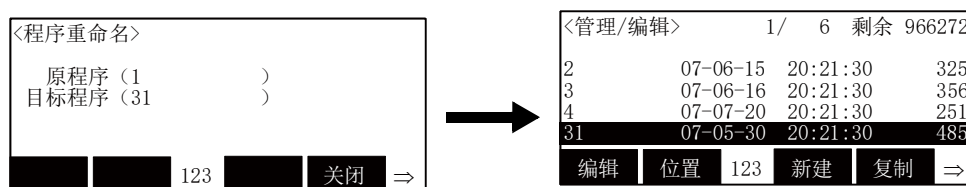


重命名的选择 [F1]

2) 要变更名称的程序的指定

在原程序的括号内，将在程序一览显示中预先显示已选择的程序名称。（图中程序名称“1”）要变更时，可通过箭头键移动光标进行变更。

在目标程序的括号内输入新程序名称，按下 [EXE] 键。



此时，将程序名称“1”变更为“31”。

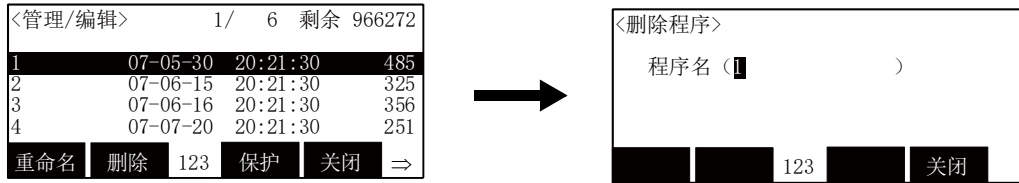
◇◆◇被保护（保护）的程序名称无法变更◇◆◇

设定了指令保护的程序名称无法变更。应解除指令保护后再执行。指令保护的解除方法请参照第 74 页的“（5）程序的保护（保护）”。

(4) 程序的删除 (删除)

1) 删除菜单的选择

在程序一览显示中按下对应“删除”的功能键 ([F2])。显示删除画面。未显示“删除”菜单时，按下 [FUNCTION] 键使其显示。

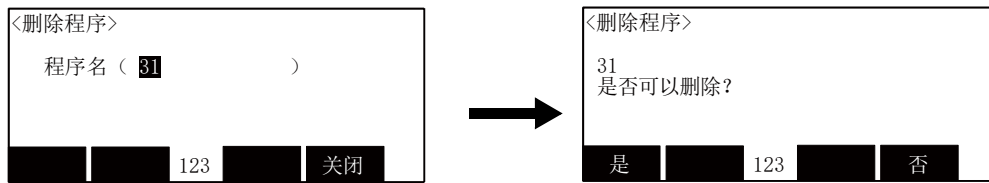


删除的选择 [F2]

2) 要删除的程序的指定

在程序名的括号内，将在程序一览显示中预先显示已选择的程序名称。(图中程序名称“1”)要变更时，输入正确的程序名称。

按下 [EXE] 键，显示确认画面。



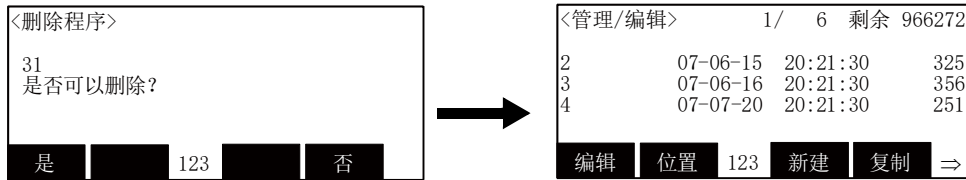
程序名称的设定 [3] [1] [EXE]

此时，将程序名称“31”删除。

3) 程序的删除

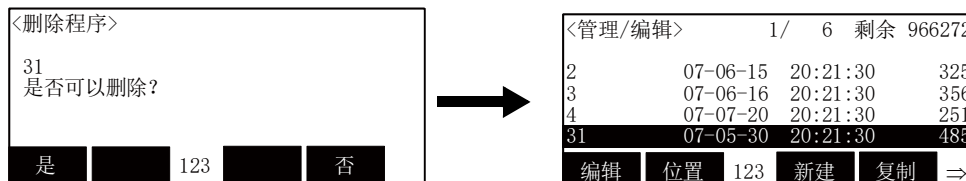
按下对应“是”的功能键，删除指定程序并返回到程序一览显示。

不删除时，应按下对应“否”的功能键。返回到删除画面。



删除 [F1]

此时，将程序名称“31”删除。



不删除 [F4]

此时，程序将无法被删除。

◆◆◆被保护 (保护) 的程序无法删除◆◆◆

设定了指令保护的程序无法删除。应解除指令保护后再执行。指令保护的解除方法请参照“ (5) 程序的保护 (保护) ”。

(5) 程序的保护（保护）

1) 保护菜单的选择

在程序一览显示中按下对应“保护”的功能键（[F3]）。显示保护画面。未显示“保护”菜单时，按下 [FUNCTION] 键使其显示。

<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	⇒

菜单的选择 [F3]

<保护>			
程序名 (1)	protect	
		指令 : OFF	
		数据 : OFF	
指令	数据	123	关闭

2) 保护的设定

程序的保护，可对该指令语句和数据（变量值）进行个别指定。

设定指令语句的保护指定时，按下对应“指令”的功能键；设定数据的保护指定时，按下对应“数据”的功能键。

<保护>			
程序名 (1)	protect	
		指令 : OFF	
		数据 : OFF	
指令	数据	123	关闭

指令语句的保护指定 [F1]

<保护>			
1	设定指令保护		
ON		123	OFF

<保护>			
程序名 (1)	protect	
		指令 : OFF	
		数据 : OFF	
指令	数据	123	关闭

数据的保护指定 [F2]

<保护>			
1	设定数据保护		
ON		123	

按下对应“ON”的功能键时，设定为“保护”；按下对应“OFF”的功能键时，设定为“不保护”。

<保护>			
1	设定指令保护		
ON		123	OFF

保护指令 [F1]、不保护指令 [F4] 的设定

<保护>			
1	设定数据保护		
ON		123	OFF

保护数据 [F1]、不保护数据 [F4] 的设定

<保护>			
程序名 (1)	protect	
		指令 : OFF	
		数据 : OFF	
指令	数据	123	关闭

ON : 保护
OFF : 不保护

◆◆◇所谓指令保护◆◆◇

对程序的删除、名称变更及指令变更的误操作进行保护的功能。

- 在复制操作中，保护信息无法被复制。
- 在初始化操作中，将忽略保护信息并执行初始化。

◆◆◇所谓数据保护◆◆◇

对由于误操作导致的位置数据的登录、变更及由于错误程序的执行导致的对各变量代入的变量进行保护的功能。

- 在复制操作中，保护信息无法被复制。
- 在初始化操作中，将忽略保护信息并执行初始化。

(6) 程序的选择

可选择作为单步运行及自动运行对象的程序。

操作方法如下所示。

1) 程序选择

在程序一览显示中，操作 [↑]、[↓] 键使光标对准要选择的程序。

<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 =>

图示为选择了程序 3 的示例。

2) 选择程序的决定

按下示教单元的 [FUNCTION]+[EXE] 键。将显示确认画面。

<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

选择程序的决定 [FUNCTION]+[EXE]

确认被显示的程序名（编号）并执行选择时，按下对应“是”的功能键（[F1]）。要中止时按下对应“否”的功能键（[F4]）。两者都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



<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 =>

程序选择的执行：[F1]（是）
中止：[F4]（否）

至此，程序的选择完成。

3.13 运行画面的操作

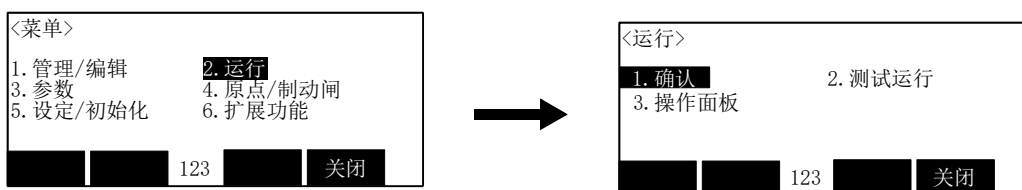
在此，对以下功能的操作方法进行说明。

- (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) 测试运行画面 2. 测试运行 : 显示已选择的程序名称及执行中的单步编号。此外，在程序执行中将连续运行模式切换到循环停止模式。

3.13.1 执行行的显示

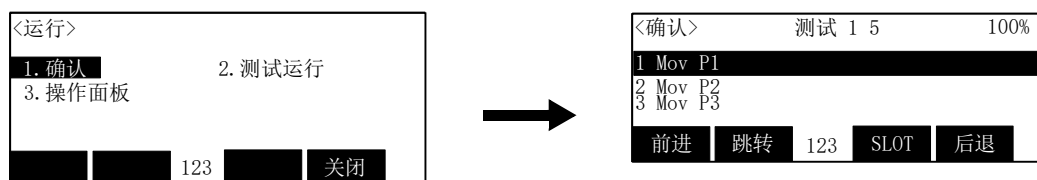
(1) 确认画面的选择

- 1) 在菜单画面中按下 [2] 键，将显示运行菜单画面。



运行菜单的选择 [2]

- 2) 在菜单画面中按下 [1] 键，将显示确认画面。
 画面中显示设定在插槽 1 中的程序。插槽编号上将连续显示程序名称。



确认的选择 [1]

程序执行中，光标移动到执行行。

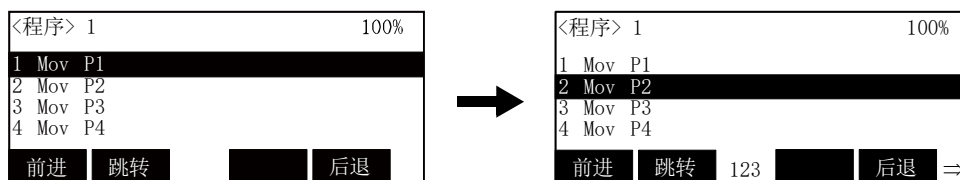
(2) 单步运行

可执行和第 53 页的“3.7 排错操作”所示的单步前进及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]），仅在按住的期间执行光标所在的单步。中途松开功能键时，将中断执行。此外，通过在机器人移动中松开或更强力按下有效开关（3 位置开关），可以使伺服 OFF 以中断执行。

1 个单步的执行完成后，示教单元画面的光标将移动至下一个单步。



单步前进 [F1]

每次按下功能键（[F1]）时，都将前进至下一个单步。

**注意**

动作中应充分注意机器人的动向。与周边装置发生干涉等异常时，应松开功能键，或者松开 / 强力按下有效开关（3 位置开关），以使机器人停止。

◇◆◇所谓单步运行◇◆◇

将逐个单步执行程序称为单步运行。由于动作速度缓慢，每执行 1 单步都将停止，因此对程序及动作位置的确认作业有效。

执行光标所在的单步。

执行 END 指令及 HLT 指令时，单步前进不会再进行。

◇◆◇执行单步的变更◇◆◇

通过箭头键移动光标及跳转操作（“跳转”），可变更执行单步。

◇◆◇要立即停止动作中的机器人时◇◆◇

①按下 [EMG. STOP] 开关。

伺服 OFF 后，发生紧急停止错误使动作中的机器人立即停止。

之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

②松开或更强力按下有效开关（3 位置开关）。

伺服 OFF 后，发生错误 2000 使动作中的机器人立即停止。

之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

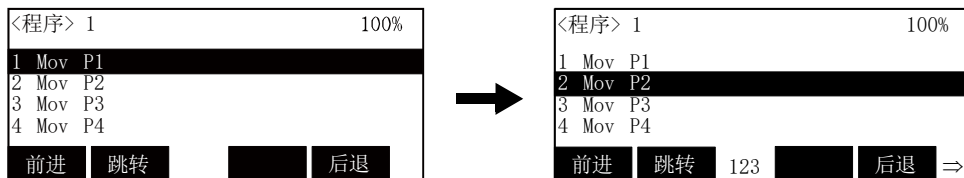
2) 单步返回

将通过单步前进及自动运行结束执行的单步或中断的单步，在仅限插补指令下逐个单步向返回方向执行。此外，最多可返回 4 个单步。

按下对应“后退”的功能键（[F4]）时，仅在持续按住期间向单步的返回方向仅执行 1 个单步。

中途松开功能键时，将中断执行。此外，通过在机器人移动中松开或更强力按下有效开关（3 位置开关），可以使伺服 OFF 以中断执行。

1 个单步的执行完成后，示教单元画面的光标在单步返回方向，移动下一个插补指令的单步。



单步前进 [F4]

每次按下功能键（[F4]）时，都将返回到前一个单步。

**注意**

动作中应充分注意机器人的动向。与周边装置发生干涉等异常时，应松开功能键，或者松开 / 强力按下有效开关（3 位置开关），以使机器人停止。

◇◆◇要立即停止动作中的机器人时◇◆◇

①按下 [EMG. STOP] 开关。

伺服 OFF 后，发生紧急停止错误使动作中的机器人立即停止。
之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

②松开或更强力按下有效开关（3 位置开关）。

伺服 OFF 后，发生错误 2000 使动作中的机器人立即停止。
之后，以错误复位 → 伺服 ON → 单步运行操作的顺序进行。

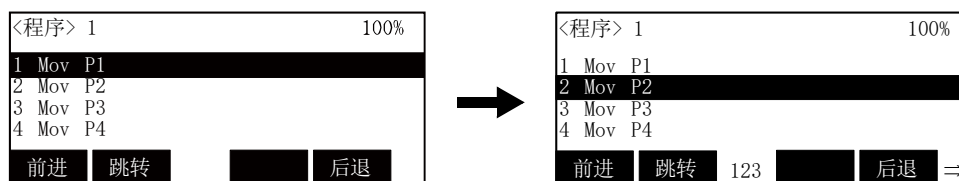
(3) 单步跳转

可变更进行单步运行的单步编号。

从单步 5 开始进行单步运行时的操作示例如下所示。

1) 单步 5 的调用。

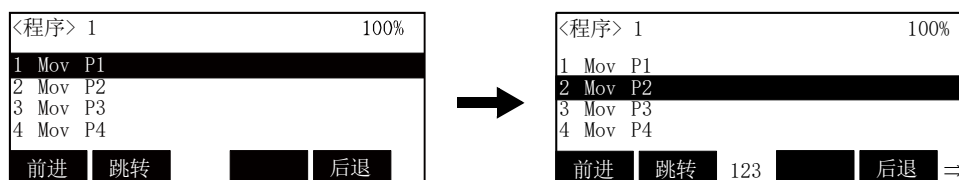
按下对应“跳转”的功能键（[F2]），在显示的单步编号输入画面中按下 [5]、[EXE] 键。光标移动到单步 5。



单步前进 [F2]

2) 单步前进

和上述的第 76 页的“(2) 单步运行”相同，按下对应“前进”的功能键（[F1]），则可从单步 5 开始执行单步前进。



单步前进 [F1]

⚠ 注意

在变更所执行单步进行单步运行时，应预先确认机器人与周边装置等未发生干涉。
此外，动作中应充分注意机器人的动向，发生与周边装置的干涉等异常时，应松开功能键，或者松开 / 强力按下有效开关（3 位置开关），以使机器人停止。

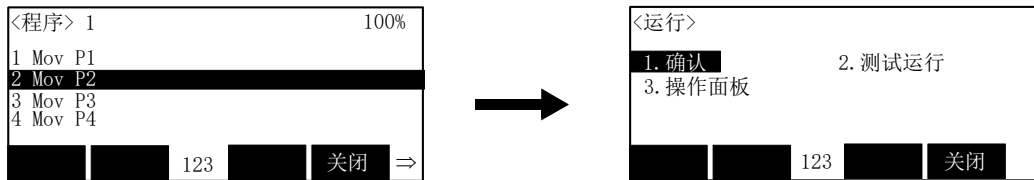
(4) 其他插槽中的单步前进

确认多任务的程序时，可在运行菜单的确认画面进行单步前进。

操作方法请参照第 79 页的“(4) 其他插槽中的单步前进”。

(5) 确认画面的结束

1) 按下对应“关闭”的功能键（[F4]），将返回到运行菜单画面。

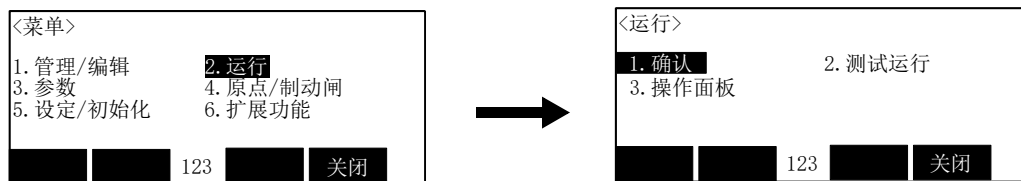


测试运行画面的结束 [F4]

3.13.2 测试运行画面

(1) 测试运行画面的选择

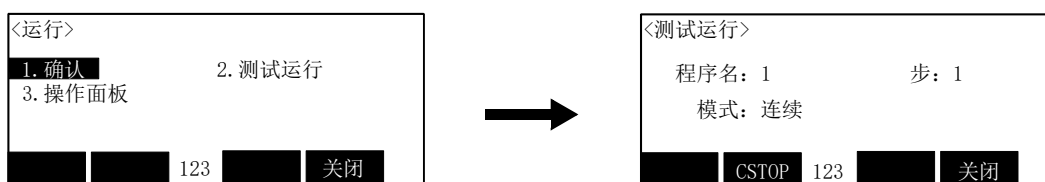
1) 在菜单画面中按下 [2] 键，将显示运行菜单画面。



运行菜单的选择 [2]

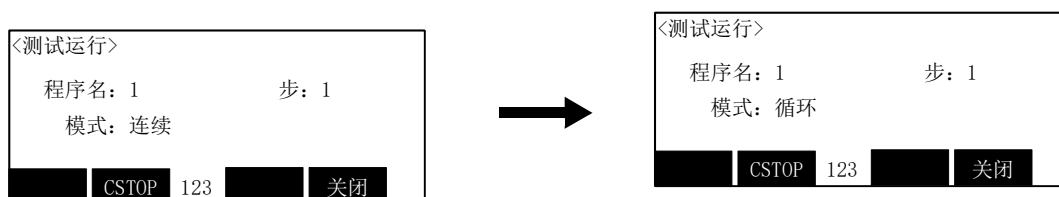
2) 在运行菜单画面中按下 [2] 键，将显示测试运行画面。

画面中显示设定在插槽 1 中的程序名、执行单步编号及当前的运行模式。



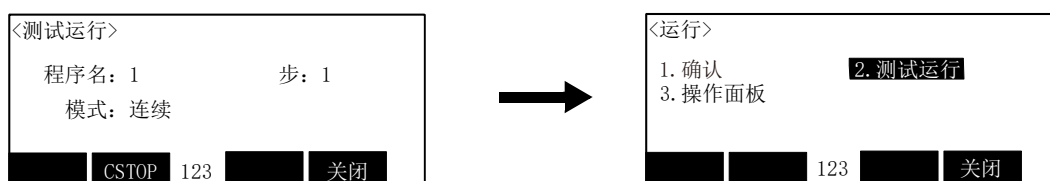
确认的选择 [2]

3) 程序执行中按下对应“CSTOP”的功能键 ([F2])，可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”，执行 END 指令或程序的最终行后，结束程序的运行。



变更为循环运行模式 [F2]

4) 按下对应“关闭”的功能键 ([F4])，将返回到运行菜单画面。



测试运行画面的结束 [F4]

◇◆◇停止程序的执行后将变为连续运行模式◇◆◇

在循环运行模式中，按下 [STOP] 键等，使程序的执行停止时，将切换到连续运行模式。在循环运行模式中继续执行程序时，应按下 [START] 按钮后再次按下 [F4] 键。

3.13.3 操作面板画面的操作

进行伺服电源的 ON/OFF、程序的选择及自动运行的启动等。
详细内容请参照第 60 页的“3.8.2 自动运行的开始”。

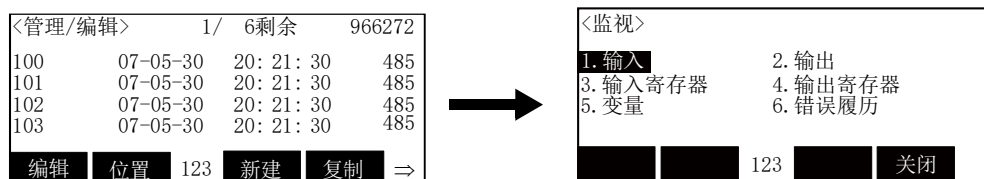
3.14 监视画面的操作

在此，对以下功能的操作方法进行说明。

- | | | |
|-------------------|----------|---------------------------|
| (1) 输入信号的监视 | 1. 输入 | : 并行输入信号的监视。 |
| (2) 输出信号的监视 | 2. 输出 | : 并行输出信号的监视与 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 | : 程序中使用的变量值的监视及值的设定。 |
| (6) 错误的履历显示 | 6. 错误履历 | : 显示已发生错误的履历。 |

上述的任一操作都可以通过按下示教单元的 [MONITOR] 键，使监视菜单画面显示后进行。示教单元即使无操作权也可实现该功能。

此时显示的画面可以是任意的，但在程序（指令）编辑画面中无变量监视功能。



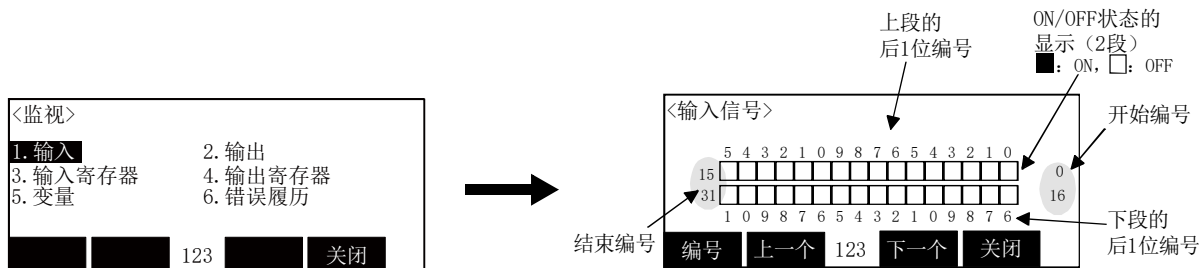
监视画面的选择 [MONITOR]

※ 显示的画面为任意。

(1) 输入信号监视

始终监视来自外部的输入信号的 ON/OFF 状态的功能。

- 1) 在监视菜单画面中按下 [1] 键，将显示输入信号的监视画面。可在 1 个画面中监视 32 点的输入信号。

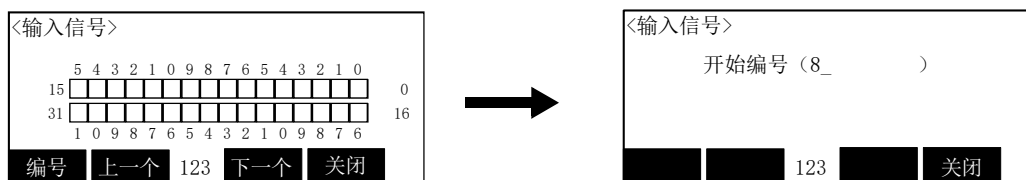


输入画面的选择 [1]

对输入信号编号 8 到 15 的状态进行确认的情况如下所示。

- 2) 按下对应“编号”的功能键 [F1]。

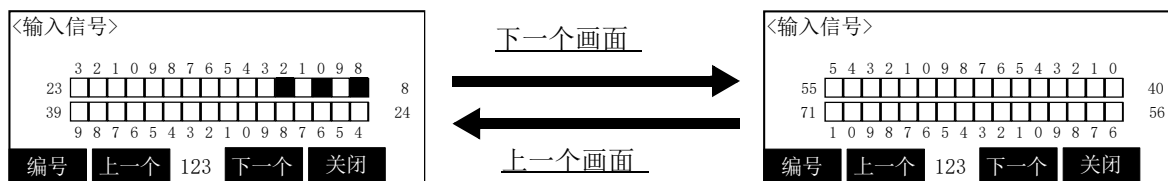
由于会显示开始编号指定画面，因此将“8”设定为开始编号。（按下 [8]、[EXE] 键）



信号编号的设定 [F1]

开始编号的设定 [8]、[EXE]

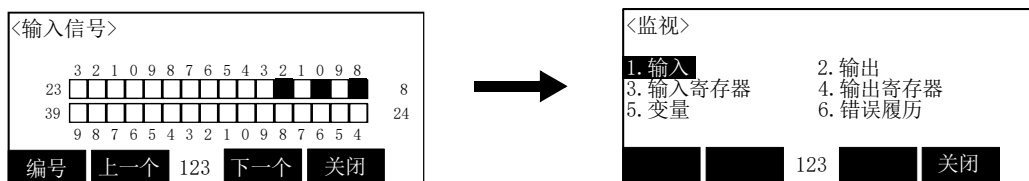
3) 将输入信号 8 号设为起始，显示 32 点的 ON/OFF 状态。涂黑表示 ON、反白表示 OFF。



在图中，8、10、12 号表示 ON，其他表示 OFF。

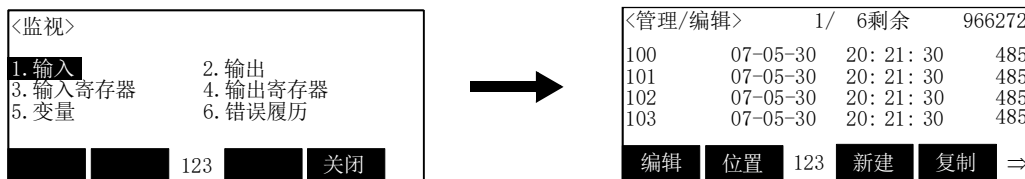
按下对应“下一个”的功能键（[F3]），将显示下一个输入信号的状态（下个画面），按下对应“上一个”的功能键（[F2]），将显示上一个输入信号的状态（上个画面）。

4) 按下对应“关闭”的功能键 [F4]，将返回到监视菜单画面。



信号编号的设定 [F4]

5) 在监视菜单画面按下对应“关闭”的功能键 [F4]，将结束监视并返回到原来的画面。



信号编号的设定 [F4]

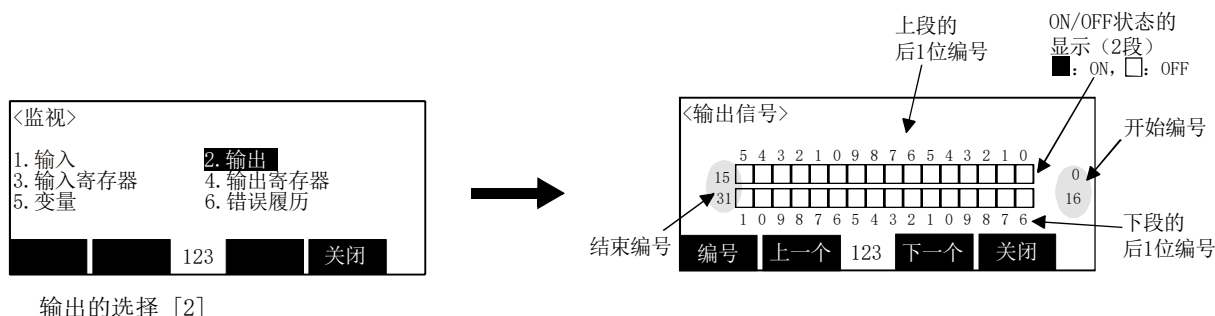
◆◆◆ 监视的结束 ◆◆◆

在输入信号的状态显示等的监视画面显示中，按下 [MONITOR] 键，无论何时都可结束监视并返回到原来的画面。

(2) 输出信号监视

有始终监视向外部的输出信号的 ON/OFF 状态的功能和将信号强制输出的功能。

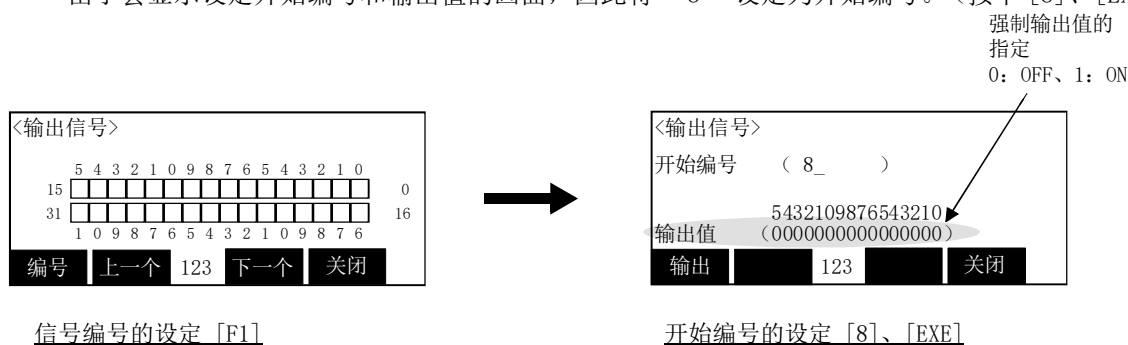
1) 在监视菜单画面中按下 [2] 键, 将显示输出信号的监视画面。可在 1 个画面中监视 32 点的输出信号。



对输出信号编号 8 到 15 的状态进行确认的情况如下所示。

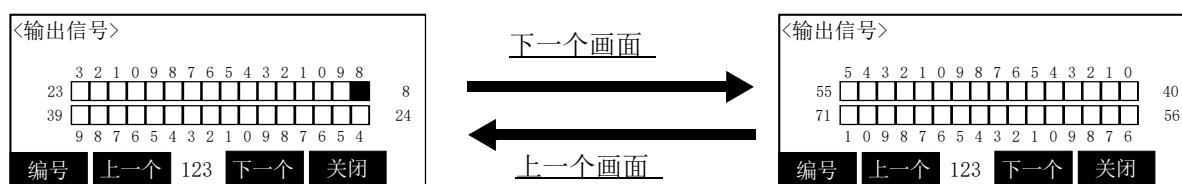
2) 按下对应“编号”的功能键 [F1]。

由于会显示设定开始编号和输出值的画面, 因此将“8”设定为开始编号。(按下 [8]、[EXE] 键)



在显示的输出值中, 会显示当前的输出信号的状态, 但是此处为设定信号的强制输出值的部分, 因此将不会始终显示。

按下对应“关闭”的功能键 ([F4]), 将显示以输出信号 8 号为起始的 32 点的输出信号监视画面。(始终显示)。涂黑表示 ON、反白表示 OFF。



在图中, 8 号表示 ON, 除此之外的都表示 OFF。

按下对应“下一个”的功能键 ([F3]), 将显示下一个输入信号的状态 (下个画面), 按下对应“上一个”的功能键 ([F2]), 将显示上一个输入信号的状态 (上个画面)。

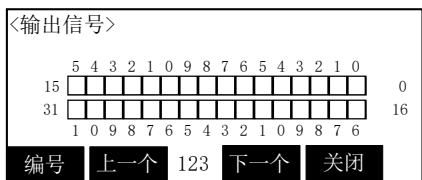
3) 输出信号的强制输出

将输出信号 8 号强制关闭时的操作方法如下所示。

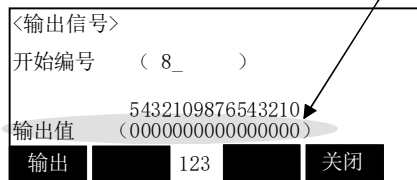
按下对应“编号”的功能键 [F1]。

由于会显示设定开始编号和输出值的画面，因此将“8”设定为开始编号。（按下 [8]、[EXE] 键）

强制输出值的指定
0: OFF、1: ON



信号编号的设定 [F1]

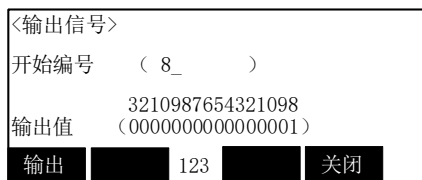


开始编号的设定 [8]、[EXE]

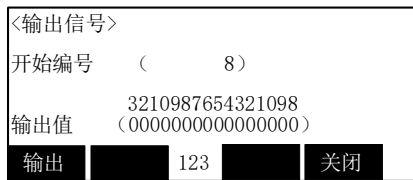
4) 可通过箭头键将光标移动到输出值“8”的位置。

由于当前输出信号 8 号为 ON，因此将显示为值“1”。

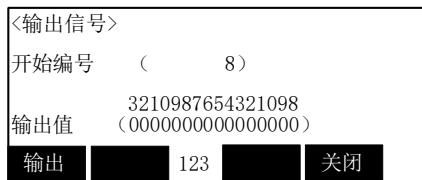
将此值变更为表示 OFF 的“0”（[0] 键），按下对应“输出”的功能键（[F1]）时，实际上相同的输出信号将变为 OFF。



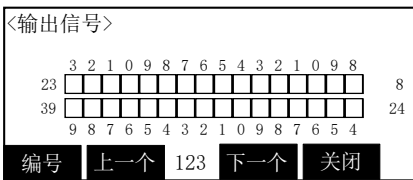
输出值的设定 [箭头]、[0]
设定值的输出 [F1]



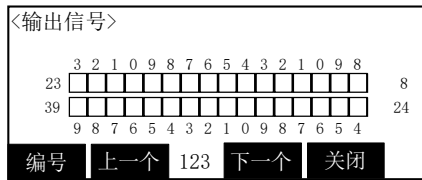
5) 按下对应“关闭”的功能键（[F4]），将返回到原来的输出信号监视画面。



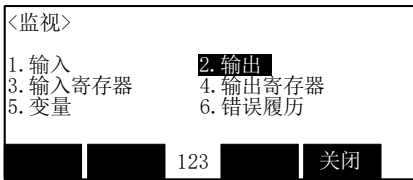
强制输出的结束 [F4]



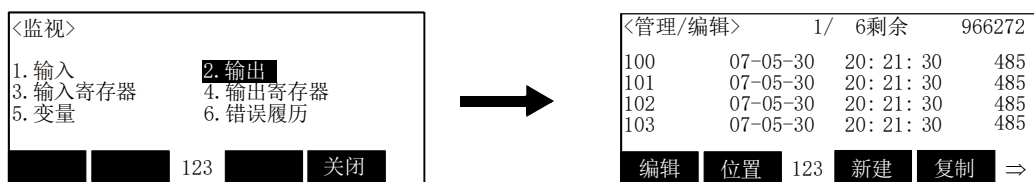
6) 按下对应“关闭”的功能键 [F4]，将返回到监视菜单画面。



信号编号的设定 [F4]



7) 在监视菜单画面按下对应“关闭”的功能键 [F4]，将结束监视并返回到原来的画面。



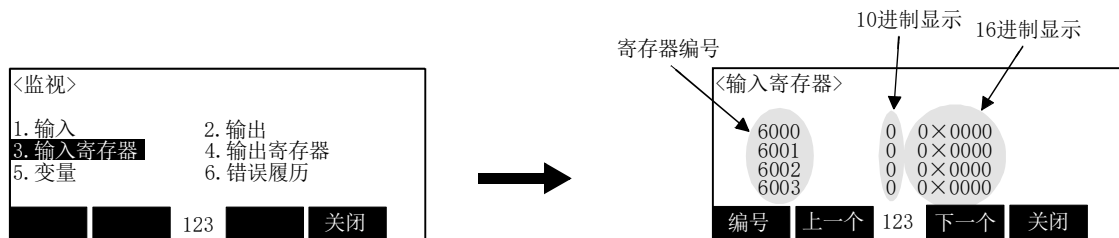
信号编号的设定 [F4]

(3) 输入寄存器监视

使用 CC-Link 时，始终监视该输入寄存器的值的功能。

注) 由于 CR800-R/Q 系列中没有 CC-Link 选购件，因此无法使用本功能。

1) 在监视菜单画面中按下 [3] 键，将显示输入寄存器的监视画面。可在 1 个画面中监视 4 个输入寄存器。

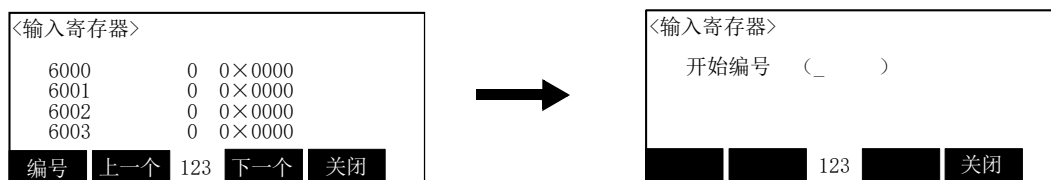


输入寄存器的选择 [3]

对输入寄存器 8000 号的状态进行确认的情况如下所示。

2) 按下对应“编号”的功能键 [F1]。

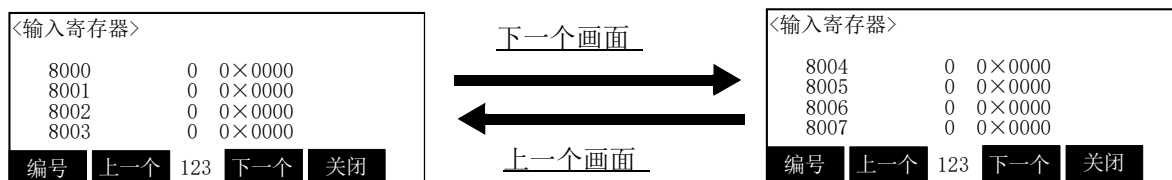
由于会显示开始编号指定画面，因此将“8000”设定为开始编号。（按下 [8]、[0]、[0]、[0]、[EXE] 键）



寄存器编号的设定 [F1]

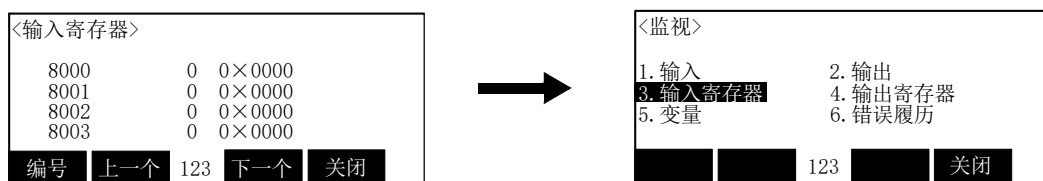
开始编号的设定 [8]、[0]、[0]、[0]、[EXE]

3) 将输入寄存器 8000 号设为起始，显示 4 个寄存器的值。



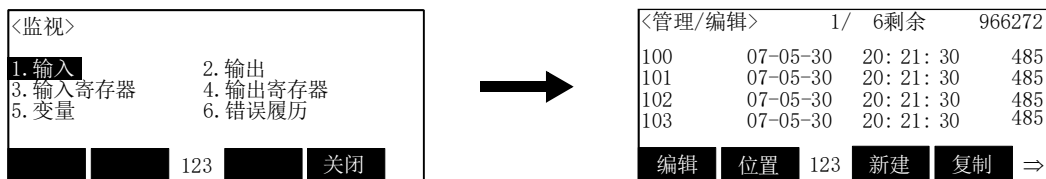
按下对应“下一个”的功能键 ([F3])，将显示下一个输入寄存器的状态 (下个画面)，按下对应“上一个”的功能键 ([F2])，将显示上一个输入寄存器的状态 (上个画面)。

4) 按下对应“关闭”的功能键 [F4]，将返回到监视菜单画面。



结束 [F4]

5) 在监视菜单画面按下对应“关闭”的功能键 [F4]，将结束监视并返回到原来的画面。



信号编号的设定 [F4]

◆◆◆ 监视的结束 ◆◆◆

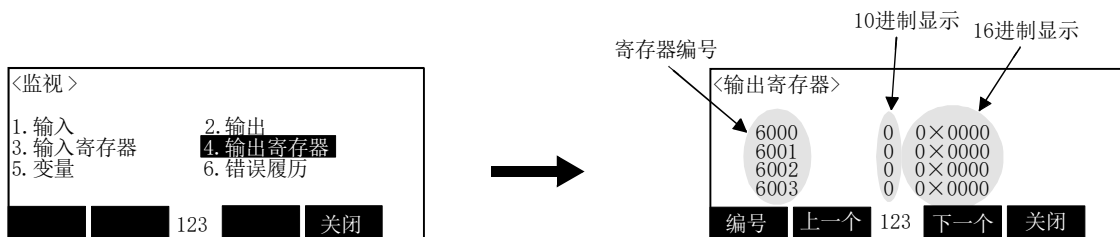
在输入寄存器的状态显示等的监视画面显示中，按下 [MONITOR] 键，无论何时都可结束监视并返回到原来的画面。

(4) 输出寄存器监视

使用 CC-Link 时，始终监视该输出寄存器的值的功能。

注) 由于 CR800-R/Q 系列中没有 CC-Link 选购件，因此无法使用本功能。

1) 在监视菜单画面中按下 [4] 键，将显示输出寄存器的监视画面。可在 1 个画面中监视 4 个输出寄存器。



输入寄存器的选择 [4]

对输出寄存器 8000 号的状态进行确认的情况如下所示。

- 2) 按下对应“编号”的功能键 [F1]。
 由于会显示开始编号指定画面，因此将“8000”设定为开始编号。（按下 [8]、[0]、[0]、[0]、[EXE] 键）

＜输出寄存器＞		
6000	0	0×0000
6001	0	0×0000
6002	0	0×0000
6003	0	0×0000
编号	上一个	123 下一个 关闭

寄存器编号的设定 [F1]

＜输出寄存器＞		
开始编号	(6000)
输出值	(0)
0×	(0000)
输出	123	关闭

开始编号的设定 [8]、[0]、[0]、[0]、[EXE]

- 3) 8000 号的当前输出值在输出值后的括号内显示为 10 进制。在下面的 0x 后的括号内的值为 16 进制。

＜输出寄存器＞		
开始编号	(8000)
输出值	(0)
0×	(0000)
输出	123	关闭

10进制数 16进制数

按下对应“关闭”的功能键（[F4]），也可返回到将输出寄存器的 8000 号设为起点的监视画面，但在当前画面中可变更输出值。

将输出寄存器 8000 号的值设定为 12（10 进制）的情况如下所示。

- 4) 值可设定为 10 进制或 16 进制。
 设定为 10 进制时，通过箭头键将光标移动到输出值，并输入“10”。（按下 [1]、[0] 键）应按下 [CLEAR] 键删除不需要的文字。
 设定为 16 进制时，通过箭头键将光标移动到 0x，并输入“C”。（按下 [C] 键）应按下 [CLEAR] 键删除不需要的文字。
 按下对应“输出”的功能键（[F1]），将实际输出已设定的值。

＜输出寄存器＞		
开始编号	(8000)
输出值	(0)
0×	(0000)
输出	123	关闭

＜输出寄存器＞		
开始编号	(8000)
输出值	(10)
0×	(000C)
输出	123	关闭

输出值的设定

通过 10 进制指定 [箭头]、[1]、[0]

通过 16 进制指定 [箭头]、[C]

设定值的输出 [F1]

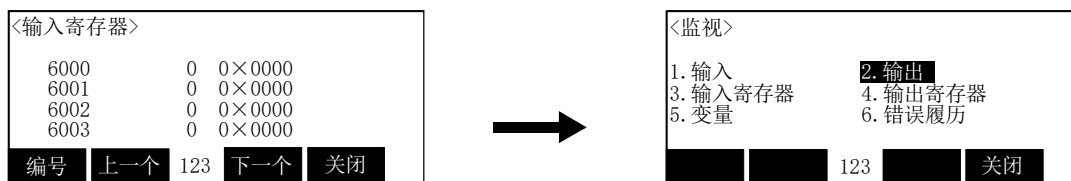
- 5) 按下对应“关闭”的功能键（[F4]），将返回到原来的输出寄存器监视画面。

＜输出寄存器＞		
开始编号	(8000)
输出值	(10)
0×	(000C)
输出	123	关闭

＜输出寄存器＞		
8000	0	0×0000
8001	0	0×0000
8002	0	0×0000
8003	0	0×0000
编号	上一个	123 下一个 关闭

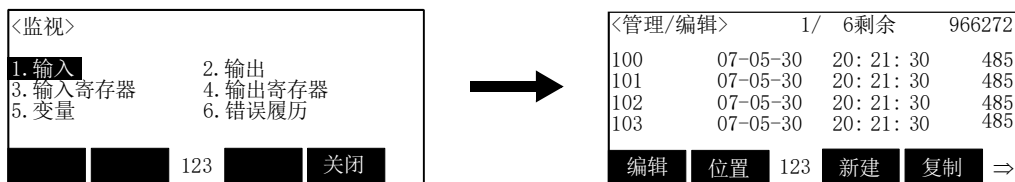
强制输出的结束 [F4]

6) 再次按下对应“关闭”的功能键（[F4]），将返回到原来的菜单画面。



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]，将结束监视并返回到原来的画面。



信号编号的设定 [F4]

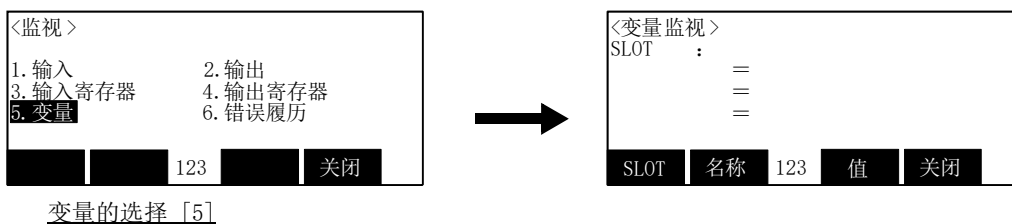
◇◆◇监视的结束◇◆◇

在输出寄存器的状态显示等的监视画面显示中，按下 [MONITOR] 键，无论何时都可结束监视并返回到原来的画面。

(5) 变量监视

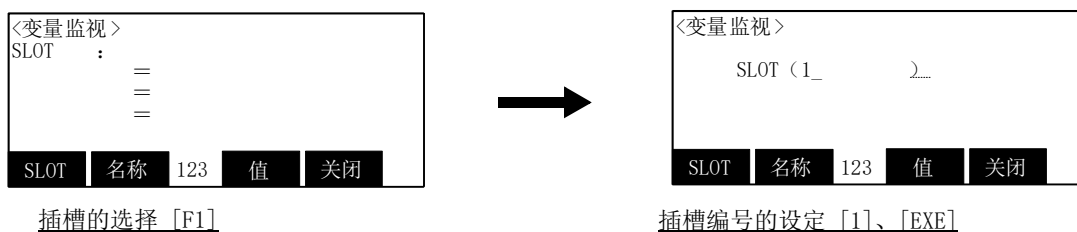
显示或变更在程序中使用的变量内容的功能。

- 1) 在监视菜单画面中按下 [5] 键，将显示变量监视画面。

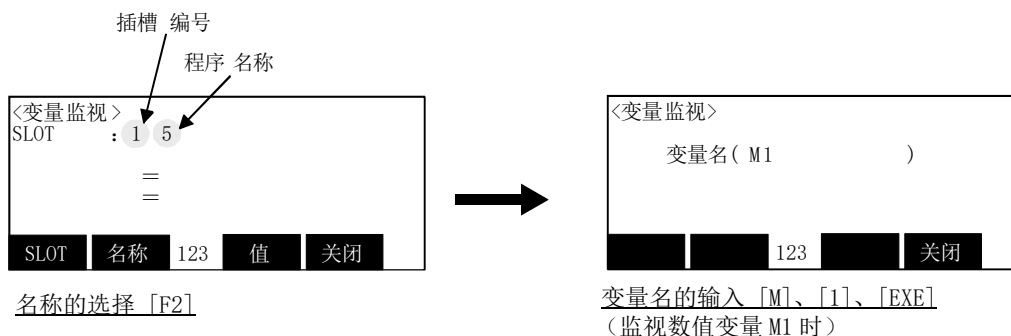


- 2) 将要作为监视对象的程序用插槽编号指定。
按下对应“SLOT”的功能键 ([F1])，输入插槽编号。

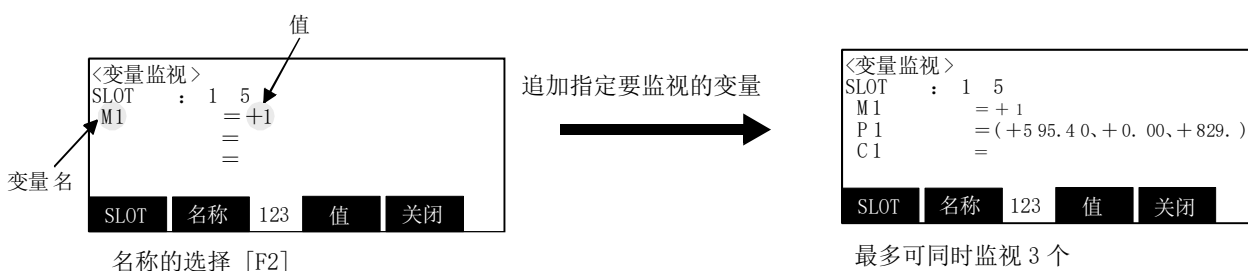
未使用多任务功能时，应设定为“1”。(按下 [1]、[EXE] 键)



- 3) 在“SLOT:”的后面将显示插槽编号和程序名称。
按下对应“名称”的功能键 ([F2])，输入要监视的变量名。



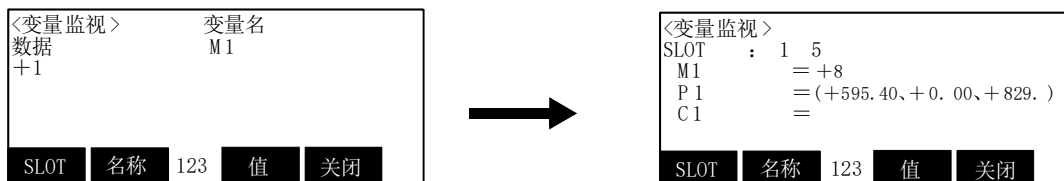
- 4) 画面中显示数值变量 M1 的值。
按下箭头键，使光标移动到空行，重复上述 3) 的操作，可追加要监视的变量。最多可同时监视 3 个变量。



5) 变量值的变更

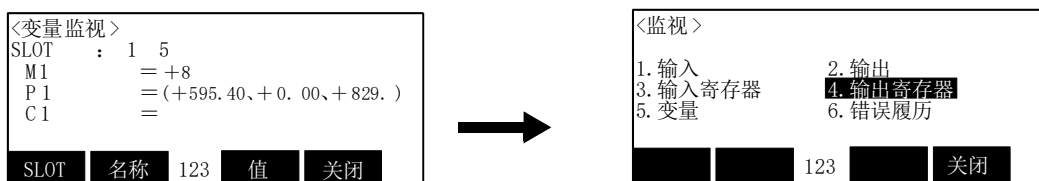
可变更显示中的变量的值。

按下箭头键，将光标移动到要变更的变量名上，按下对应“值”的功能键（[F3]）。
会显示当前值（数据），但可输入并进行变更。



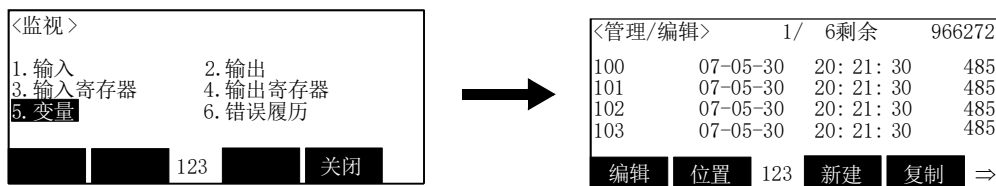
值的设定 [8]、[EXE]
(将值设定为8时)

6) 按下对应“关闭”的功能键（[F4]），将返回到原来的菜单画面。



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]，将结束监视并返回到原来的画面。



信号编号的设定 [F4]

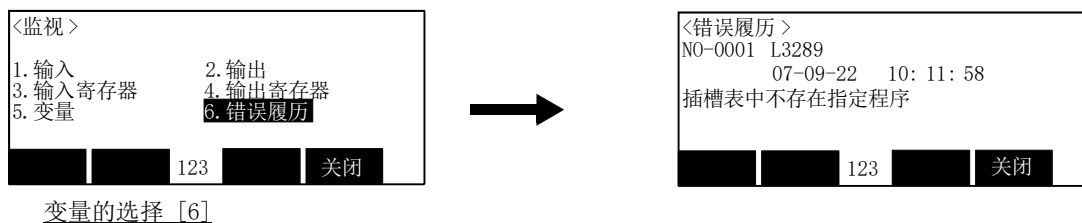
◇◆◇无关操作权◇◆◇

即使示教单元无操作权也可执行本操作。
此外，即使在自动运行中也可变更变量的值（数据）。

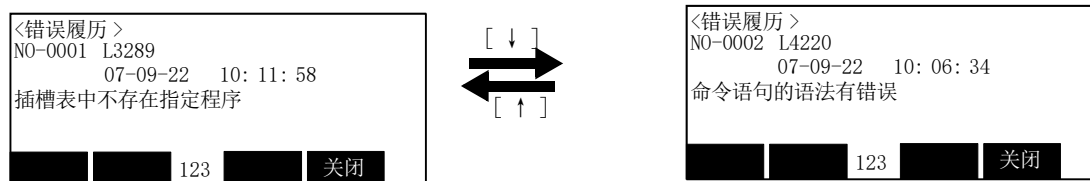
(6) 错误履历

显示在机器人中已发生的错误履历的功能。请作为故障发生时的参考。

1) 在监视菜单画面中按下 [6] 键，将显示错误履历画面。



通过箭头键显示错误履历的前后。



显示下一个错误履历 [↓]、显示上一个错误履历 [↑]

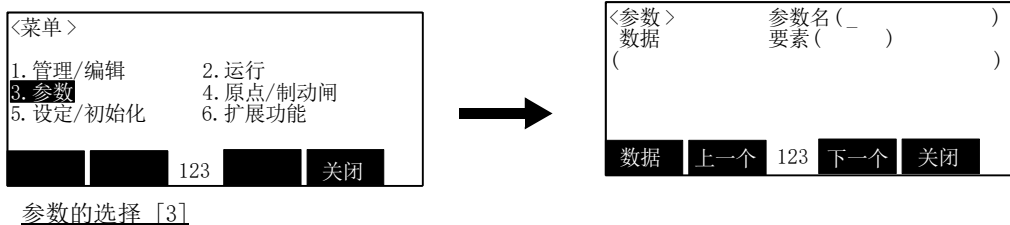
◇◆◇无关操作权◆◆◇

即使示教单元无操作权也可执行本操作。

3.15 参数画面的操作

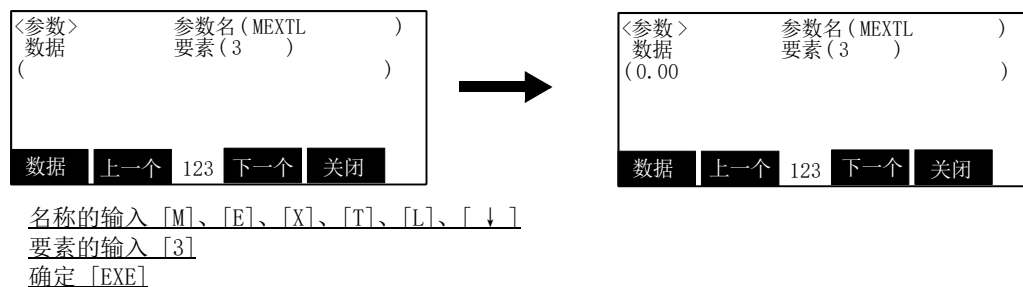
将并行 I/O 的专用输入输出设定及工具长度等的设定作为参数登录，机器人基于各参数的设定值进行动作。显示、登录各参数的设定值的功能。

1) 在菜单画面中按下 [3] 键，将显示参数画面。

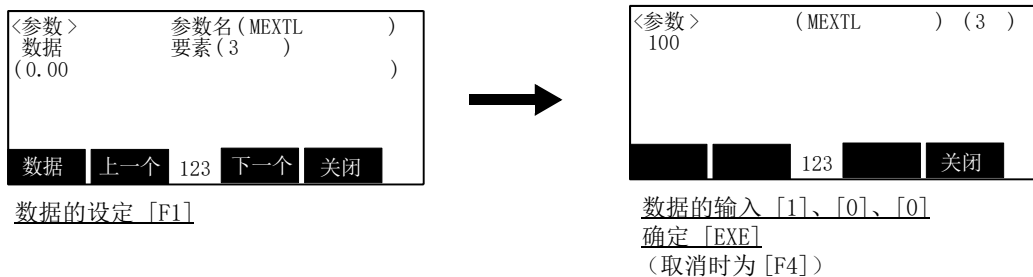


将参数“MEXTL（工具转换数据）”的 Z 轴（第 3 号的要素）的设定值由 0 变更为 100mm 时的示例如下所示。

2) 在名称中输入“MEXTL”、在要素中输入“3”。
将显示当前设定的数据。

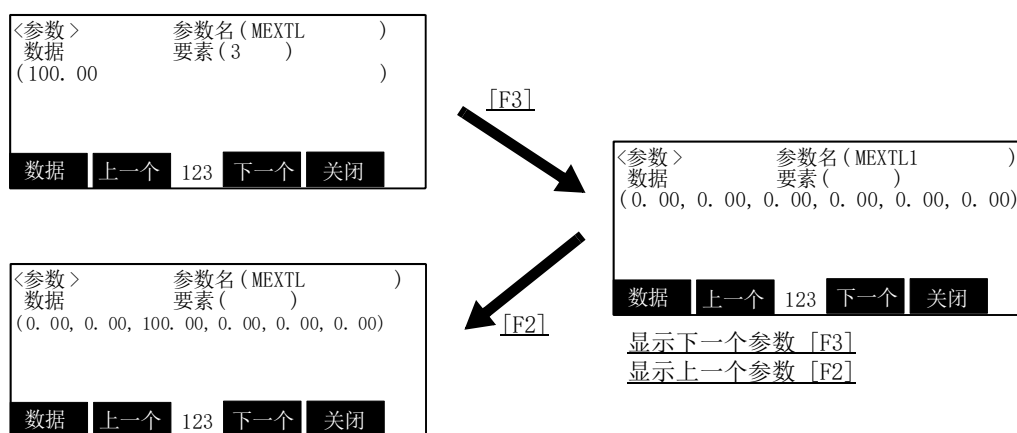


3) 按下对应“数据”的功能键 ([F1])，输入新的设定值“100”。
不需要的数字请用 [CLEAR] 键删除。



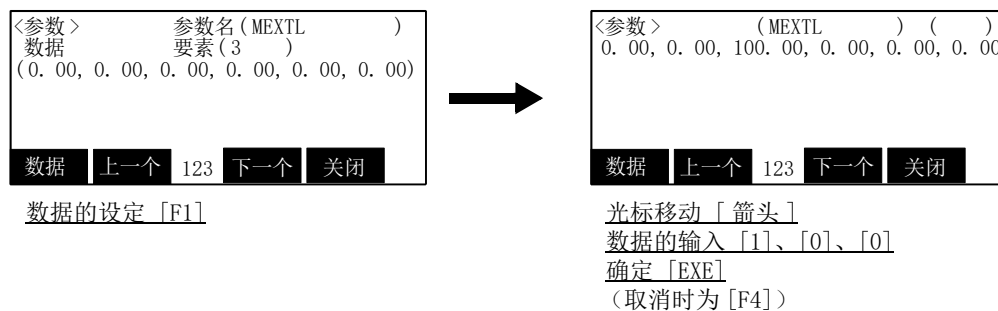
按下 [EXE] 键，蜂鸣器将鸣响，确定鸣响值后返回到参数的显示画面。
即使已输入新的设定值后，也可通过按下对应“关闭”的功能键 ([F4])，在不变更设定值下返回到参数显示画面。

此外，按下对应“下一个”的功能键 ([F3])，将显示下一个参数，按下对应“上一个”的功能键 ([F2])，将显示上一个参数。
此时，显示以名称显示的参数的全部要素，为了在该参数单位前后显示，将要素编号的指定删除。



即使在此状态下也可变更值。

按下对应“数据”的功能键（[F1]），通过箭头键将光标移动到要变更的要素编号的位置，输入新的设定值。不需要的数字请用 [CLEAR] 键删除。



按下 [EXE] 键，蜂鸣器将鸣响，确定鸣响值后返回到参数的显示画面。

即使已输入新的设定值后，也可通过按下对应“关闭”的功能键（[F4]），在不变更设定值下返回到参数显示画面。

◇◆◇需要重新接入电源◇◆◇

由于重新接入控制器的电源，因此已变更的参数变为有效。

◇◆◇程序中仅可显示◇◆◇

在程序的执行中变更参数的设定值，将发生错误。（即使发生错误，程序的执行也不会停止）

◇◆◇显示与已输入参数名称相近的参数◇◆◇

参数的名称即使没有正确输入全部的文字，也会自动显示和输入的名称相近的参数。

因此，先输入名称较长的参数的一部分，按下对应“下一个”或“上一个”的功能键，即可显示目的参数。

3.16 原点・制动闸画面的操作

(1) 原点

由于参数的消失及机器人的干涉等，使原点位置消失或偏移时，需要按照本操作重新设定机器人的原点。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”。

(2) 制动闸

在伺服 OFF 状态下，解除伺服电机制动闸的功能。请参照第 67 页的“3.9 伺服 ON/OFF”进行伺服 OFF 操作。

在用手直接移动机器人机械臂时使用。

CR860 时，如果不将用户自备的可用设备连接到控制器的紧急停止输入连接器（CNUSR11）的可用设备，则本操作将无效。

为了安全操作，务必连接可用设备。

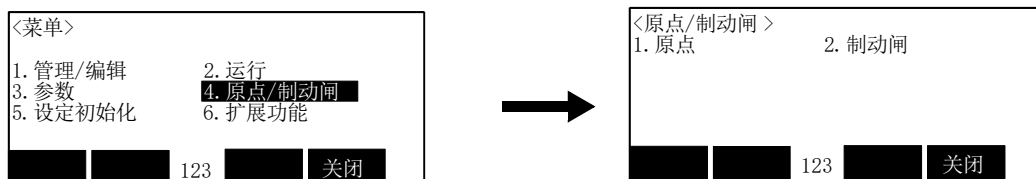


注意

执行制动闸解除时，由于机器人的构成、解除的轴将导致机器人机械臂因本身重量而落下。
为了安全起见，应预先采取支撑等处理以避免由于自重而落下。

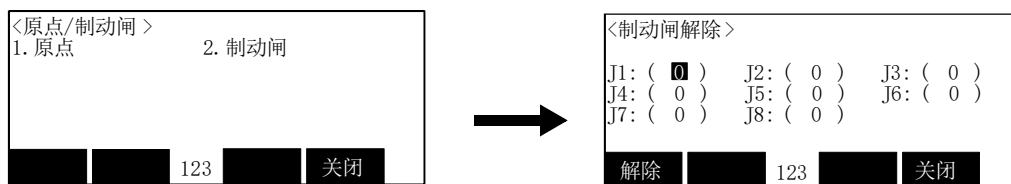
操作方法如下所示。本操作应在轻按住有效开关（3 位置开关）的状态下进行。

1) 在菜单画面中按下 [4] 键，显示原点・制动闸画面。



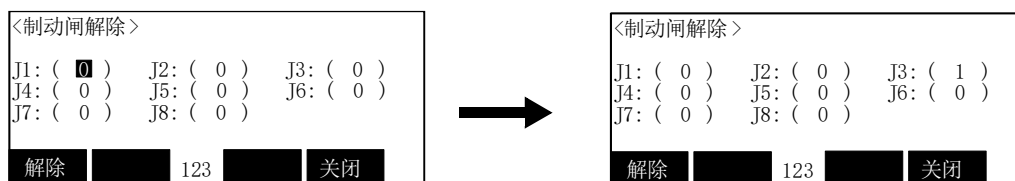
原点・制动闸画面的选择 [4]

2) 在原点・制动闸画面中按下 [2] 键，显示制动闸解除画面。



原点・制动闸画面的选择 [2]

3) 在要解除制动闸的轴中输入“1”。



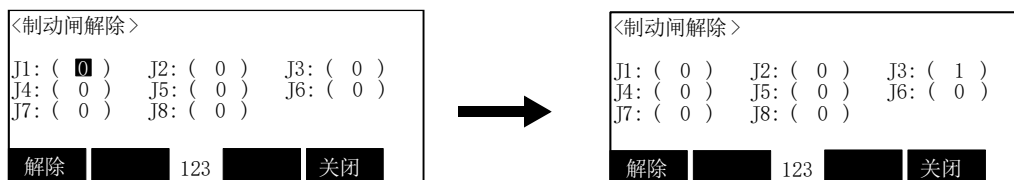
制动闸解除轴的设定 [箭头]、[1]

⚠ 注意

虽然可同时设定多个轴的制动闸解除，但是由于机器人的轴构成将导致机械臂因其合成的本身重量而落下。
为了安全起见，应预先采取支撑等处理以避免由于自重而落下。

- 4) 在轻按住有效开关（3位置开关）的状态下，按下对应“解除”的功能键（[F1]），仅在按住功能键（[F1]）期间，已指定的轴的制动闸将被解除。
- RH-3FRH 系列、RV-2FR/4FR/7FR 系列、RH-3FRHR 系列：
制动闸解除为连续性的。
 - RH-6FRH/12FRH/20FRH 系列：
制动闸重复间断性解除 / 锁定。
 - RV-13FR 系列、RV-20FR 系列：
J2 轴、J3 轴制动闸重复进行间断性解除 / 锁定。其他轴的制动闸为连续性解除。
 - RV-35FR/50FR/80FR 系列：
J2 轴、J3 轴、J4 轴、J5 轴、J6 轴制动闸重复进行间断性解除 / 锁定。J1 轴的制动闸为连续性解除。

要使制动闸动作时，应松开 [F1] 键，或者更强力按下或松开有效开关（3位置开关）。



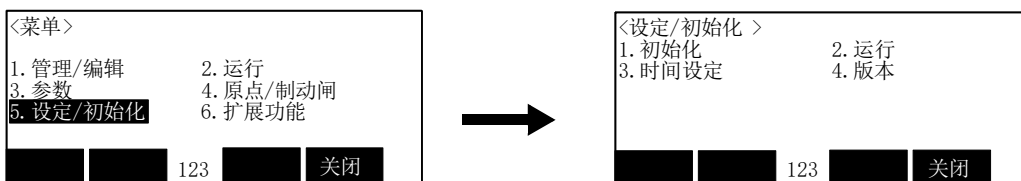
制动闸解除 [有效开关]、[F1]

3.17 设定・初始化画面的操作

在此，对以下功能的操作方法进行说明。

- (1) 初始化 1. 程序 : 删除全部程序。
2. 参数 : 将参数返回到出厂时的设定。
3. 电池 : 复位电池的消耗时间。
- (2) 运行 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。
机器人 CPU 内有时钟功能，用于程序登录・变更时间及错误时间的显示等。与当前的日期・时间偏离时，应变更为正确的日期、时间。
- (4) 版本 显示机器人 CPU 和示教单元的软件版本。

在菜单画面中按下 [5] 键，显示设定・初始化菜单画面。

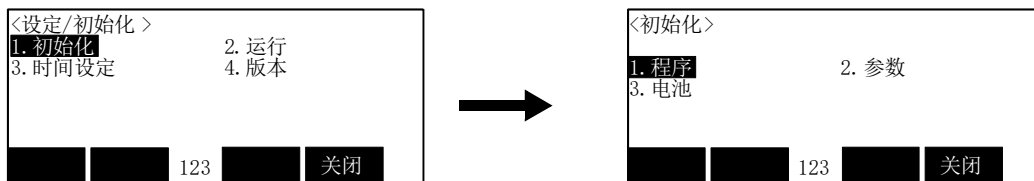


设定・初始化菜单画面的选择 [5]

(1) 程序的初始化

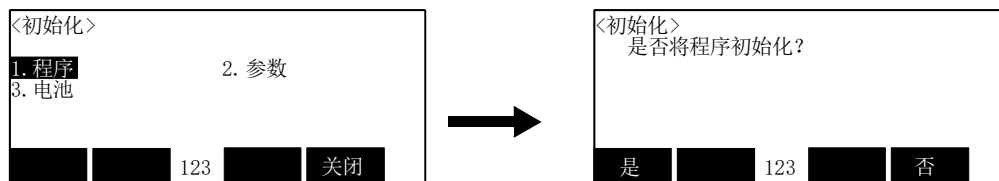
删除全部程序。

1) 在设定・初始化菜单画面中按下 [1] 键，显示初始化菜单画面。



初始化菜单画面的选择 [1]

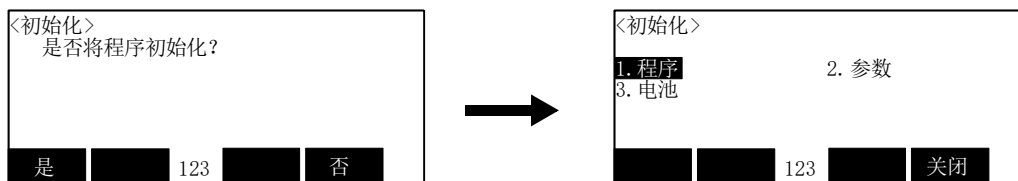
2) 在初始化菜单画面中按下 [1] 键，选择“程序”。
显示确认的画面。



初始化菜单画面的选择 [1]

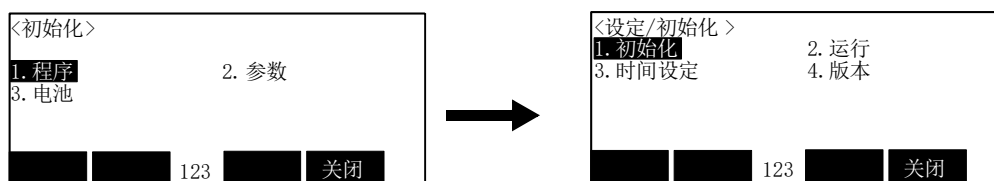
- 3) 执行初始化时，按下对应“是”的功能键（[F1]），不执行时按下对应“否”的功能键（[F4]）。

画面将返回到初始化菜单画面。



初始化的确认
执行时 [F1]
不执行时 [F4]

- 4) 按下对应“关闭”的功能键（[F4]）键，将返回到设定・初始化画面。



初始化画面的结束 [F4]

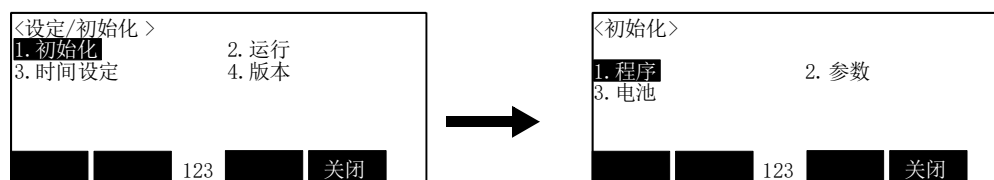
◇◆◇即使受保护也会执行◇◆◇

即使在程序保护或变量保护设定为 ON 时也会执行程序初始化，请加以注意。

(2) 参数的初始化

将参数返回到出厂时的设定。

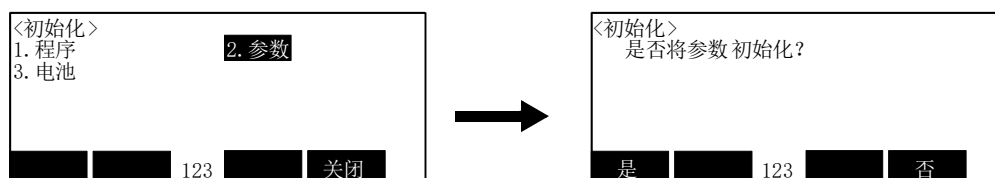
- 1) 在设定・初始化菜单画面中按下 [1] 键，显示初始化菜单画面。



初始化菜单画面的选择 [1]

- 2) 在初始化菜单画面中按下 [2] 键，选择“参数”。

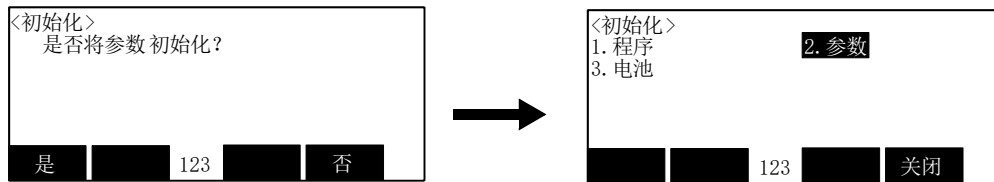
显示确认的画面。



参数选择 [2]

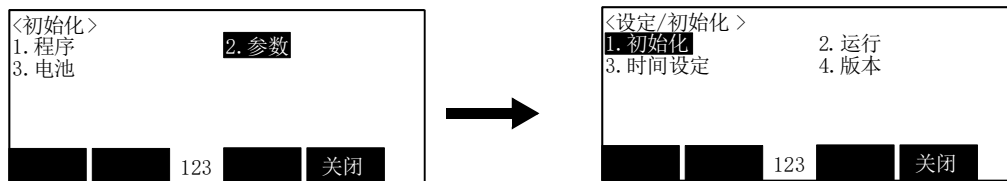
3) 执行初始化时，按下对应“是”的功能键（[F1]），不执行时按下对应“否”的功能键（[F4]）。

画面将返回到初始化菜单画面。



初始化的确认
 执行时 [F1]
 不执行时 [F4]

4) 按下对应“关闭”的功能键（[F4]）键，将返回到设定・初始化画面。



初始化画面的结束 [F4]

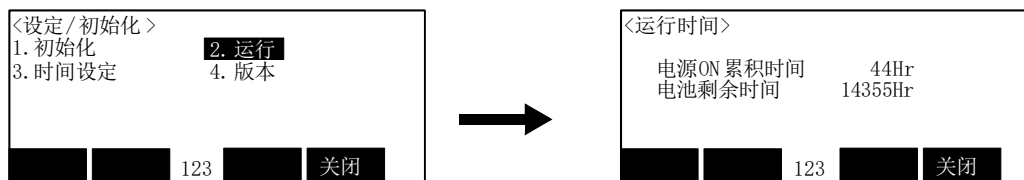
(3) 电池的初始化

CR800 系列中，通过检测电压自动对电池的消耗进行检查。因此，更换电池后无需进行初始化操作。即使进行初始化操作也无效。

(4) 运行

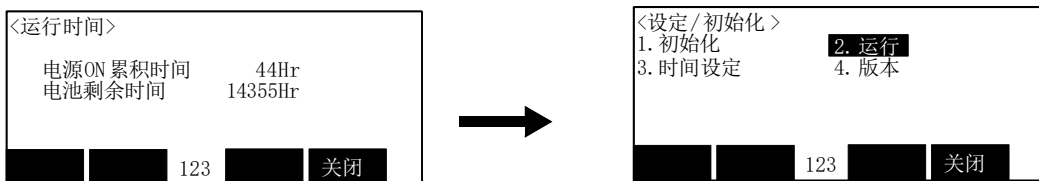
显示电源 ON 的累积时间和电池的剩余时间。

1) 在设定・初始化菜单画面中按下 [2] 键，显示运行时间画面。



初始化菜单画面的选择 [1]

2) 按下对应“关闭”的功能键（[F4]）键，将返回到设定・初始化画面。

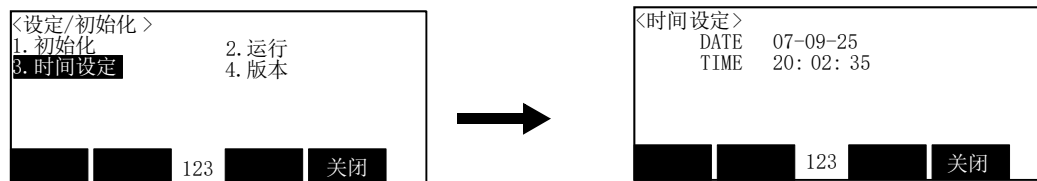


运行时间画面的结束 [F4]

(5) 时间设定

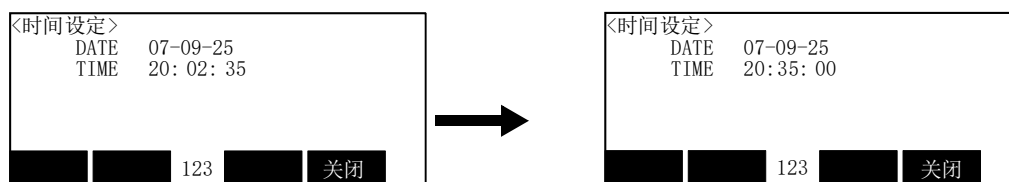
进行日期和时间的显示、设定。

- 1) 在设定・初始化菜单画面中按下 [3] 键，将显示时间设定画面和当前的日期和时间。



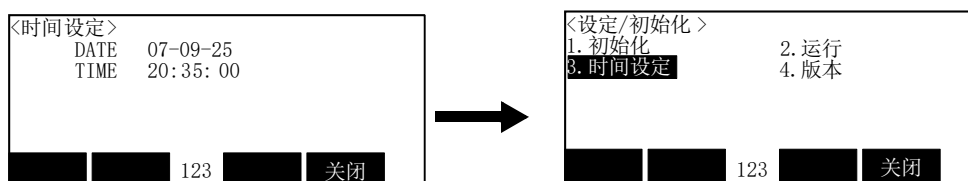
时间设定画面的选择 [3]

- 2) 可在时间设定画面中设定日期和时间。
应通过箭头键移动光标，输入当前的日期和时间。



日期、时间的输入 [箭头]、[数值]
日期、时间的确定 [EXE]

- 3) 按下对应“关闭”的功能键 ([F4]) 键，将返回到设定・初始化画面。

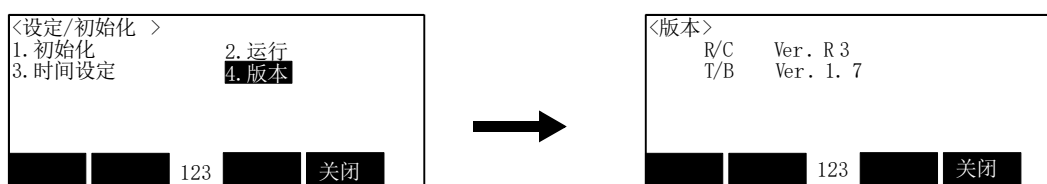


时间设定画面的结束 [F4]

(6) 版本

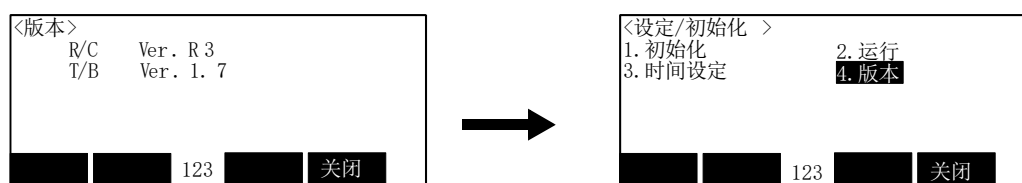
显示机器人 CPU 和示教单元的软件版本。

- 1) 在设定・初始化菜单画面中按下 [4] 键，将显示版本画面。



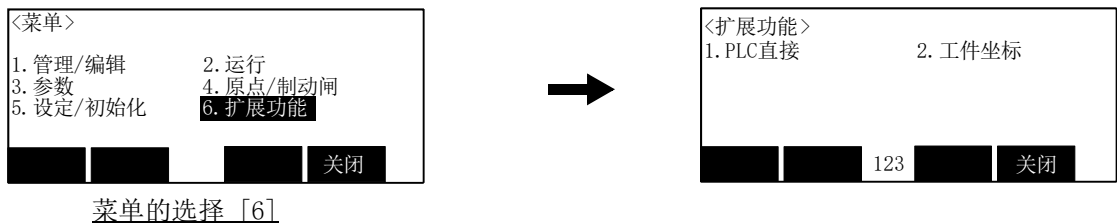
版本画面的选择 [4]

- 2) 按下对应“关闭”的功能键 ([F4]) 键，将返回到设定・初始化画面。



版本画面的结束 [F4]

3.18 扩展功能的操作



(1) PLC 直接

本功能是通过可编程控制器的程序直接控制机器人的功能。CR800-R/Q 系列中可以使用。

关于其规格・操作方法的详细内容，请参照另一手册“使用说明书/CR800-R/CR800-Q 系列控制器 iQ Platform 对应 扩展功能说明书（BFP-A3580）”。

(2) 工件坐标

是定义工件 JOG 操作所需的工件坐标系的画面。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】

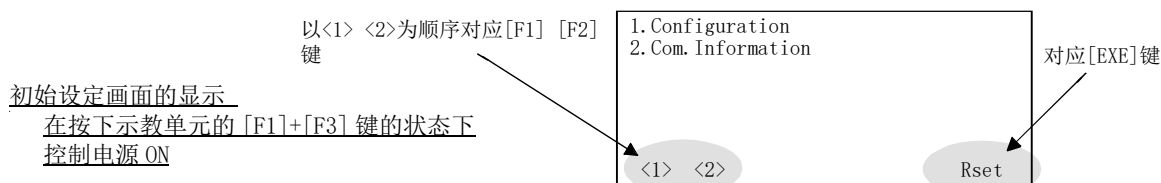
- 1) 工件坐标的设定、工件 JOG 操作：另一手册“使用说明书 / 从机器人本体安装到维护”
- 2) JOG 前进的种类：本书 / 第 27 页的“3.2.1JOG 前进的种类”
- 3) 相关参数：本书 / 第 468 页的“5.1 动作参数”的“工件坐标系”

3.19 初始设定画面的操作

初始设定的功能如下所示。

- (1) 显示语言的设定 可将示教单元画面中显示的文字设定为中文或英文的任一种。
- (2) 对比度调整 示教单元画面的亮度可通过 16 个等级进行调整。

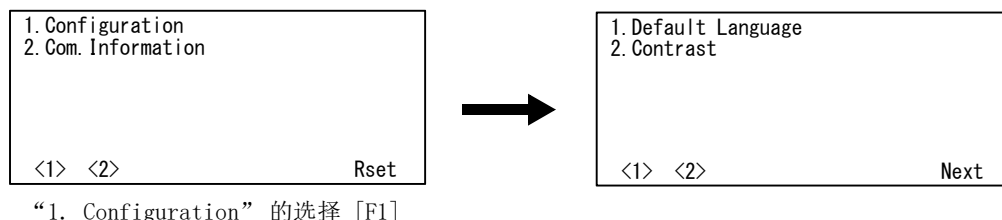
在同时按下示教单元的 [F1] 键和 [F3] 键的状态下，将控制电源设为 ON，在显示的初始设定画面中进行此操作。



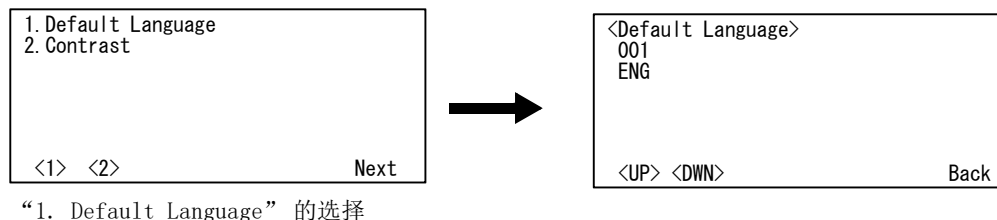
(1) 显示语言的设定

将示教单元画面中显示的文字设定为中文或英文的任一种。

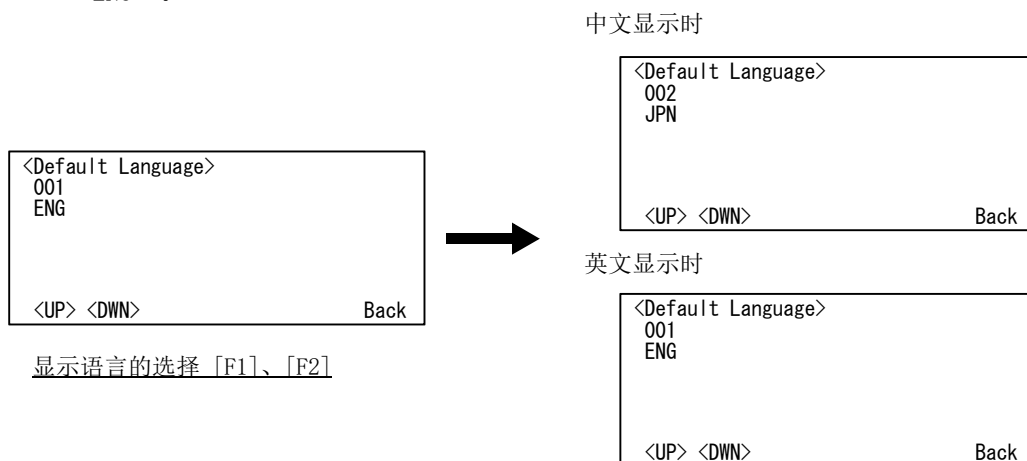
- 1) 在初始设定画面按下 [F1] 键，选择 “1. Configuration”。



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

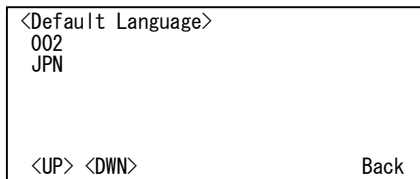


- 3) 按下 [F1] 或 [F2] 键，要设为中文显示时画面上显示 “CHNS”，设为英文显示时画面上显示 “ENG”。

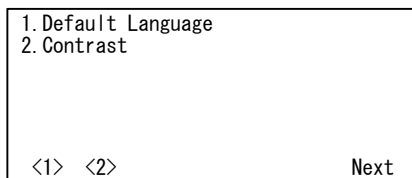
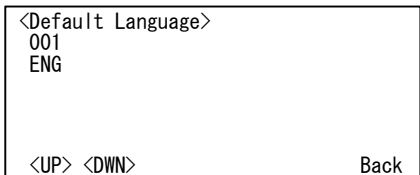


4) 按下 [EXE] 键进行确定。

中文显示时

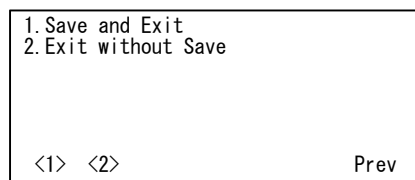
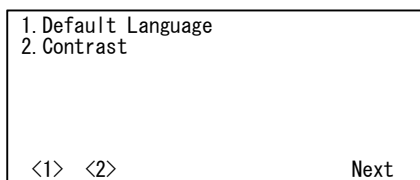


英文显示时



显示语言的确定 [EXE]

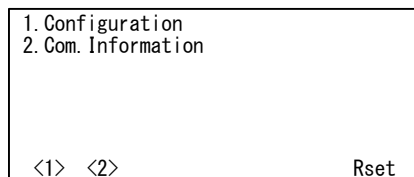
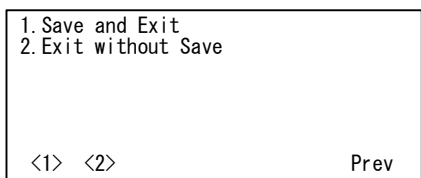
5) 按下 [EXE] 键，显示结束画面。



语言设定的结束 [EXE]

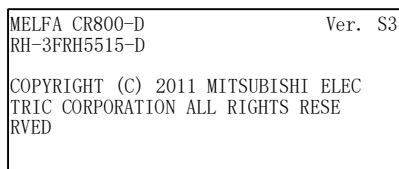
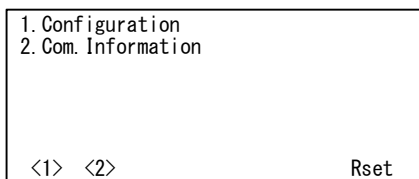
6) 按下 [F1] 键对设定进行保存。

按下 [F2] 键不对设定进行保存。无论按下哪一个键都将返回到初始设定画面。
此外，按下 [EXE] 键，可重新设定。



对设定进行保存时 [F1]
不对设定进行保存时 [F2]
重新设定时 [EXE]

7) 按下 [EXE] 键时，示教单元将以被设定的语言启动。

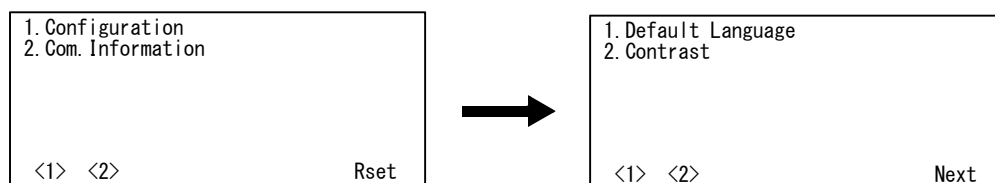


初始设定画面的结束 [EXE]

(2) 对比度的设定

示教单元画面的亮度可通过 16 个等级进行调整。

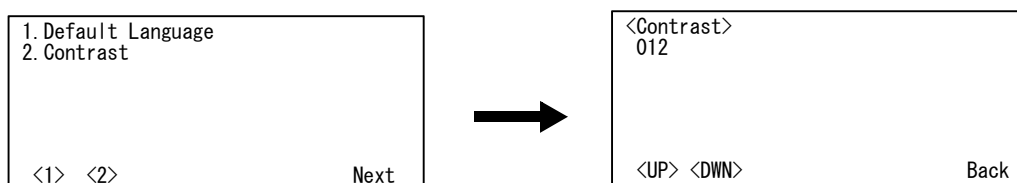
- 1) 在初始设定画面按下 [F1] 键，选择“1.Configuration”。



“1.Configuration” 的选择 [F1]

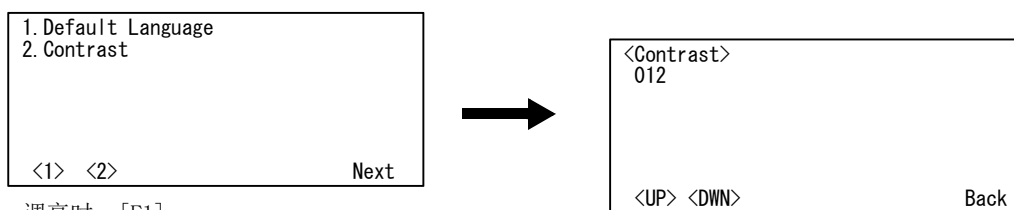
- 2) 按下 [F2] 键，选择“2.Contrast”。

当前被设定的亮度以 0 到 15 的数值显示。



“1.Contrast” 的选择 [F2]

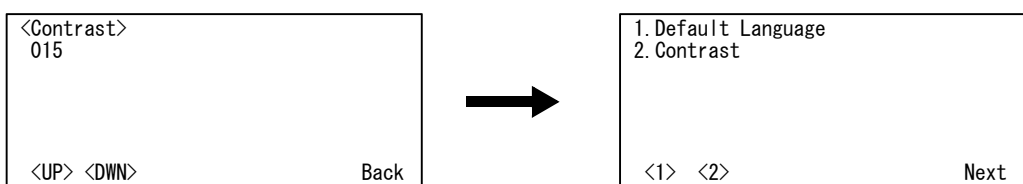
- 3) 将画面调亮时按下 [F1] 键，调暗时按下 [F2] 键，设定适当的亮度。数值越大越明亮。



调亮时 [F1]

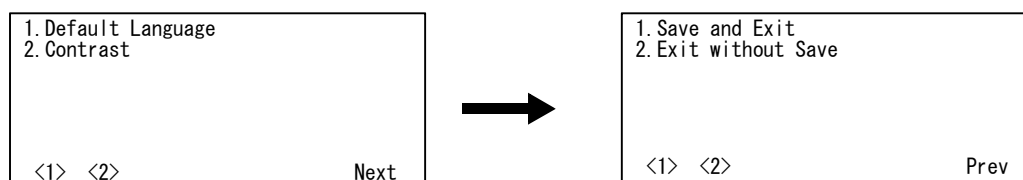
调暗时 [F2]

- 4) 按下 [EXE] 键进行确定。



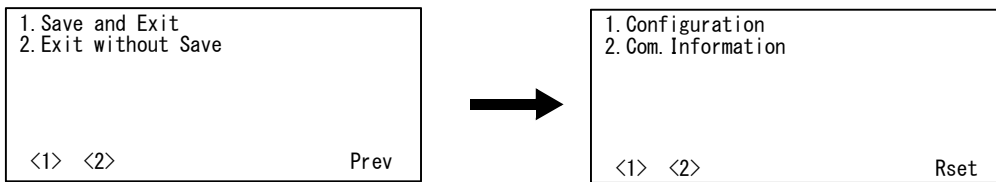
亮度的确定 [EXE]

- 5) 按下 [EXE] 键，显示结束画面。



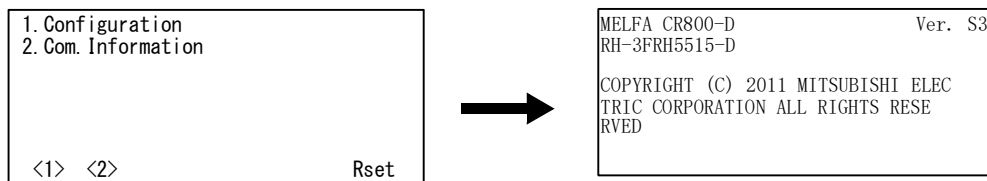
对比度设定的结束 [EXE]

- 6) 按下 [F1] 键对设定进行保存。
按下 [F2] 键不对设定进行保存。无论按下哪一个键都将返回到初始设定画面。
此外，按下 [EXE] 键，可重新设定。



对设定进行保存时 [F1]
不对设定进行保存时 [F2]
重新设定时 [EXE]

- 7) 按下 [EXE] 键时，示教单元将以被设定的语言启动。



初始设定画面的结束 [EXE]

4 MELFA-BASIC VI

在此说明关于程序语言“MELFA-BASIC VI”的功能及详细语言规格。

4.1 MELFA-BASIC VI的功能

在此说明关于程序语言“MELFA-BASIC VI”的概要。说明机器人的基本动作、信号的输入输出、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) 确认到达目的位置	Fine、Mov 和 Dly
7		(7) 高轨迹精度控制	Prec
8		(8) 抓手・工具控制	HOpen、HClose、Tool
9	4.1.2 托盘运算	-----	Def Plt、Plt
10	4.1.3 程序控制	(1) 无条件分支・条件分支・待机	GoTo、If Then Else、Wait 其他
11		(2) 循环	For Next、While Wend
12		(3) 中断	Def Act、Act
13		(4) 子程序	GoSub、Callp、On GoSub 其他
14		(5) 定时器	Dly
15		(6) 停止	End (1 循环停止)、Hlt
16	4.1.4 外部信号的输入输出	(1) 信号输入	M_In、M_Inb、M_Inw 其他
17		(2) 信号输出	M_Out、M_Outb、M_Outw 其他
18	4.1.5 通信	-----	Open、Close、Print、Input 其他
19	4.1.6 公式和运算	(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、WthIf

各指令的详细说明请参照第 173 页的“4.12 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

通过各关节轴单位的插补移动至指定的位置。(由于在各关节轴单位进行插补,因此前端的轨迹不会变直线。)

■ 指令语句

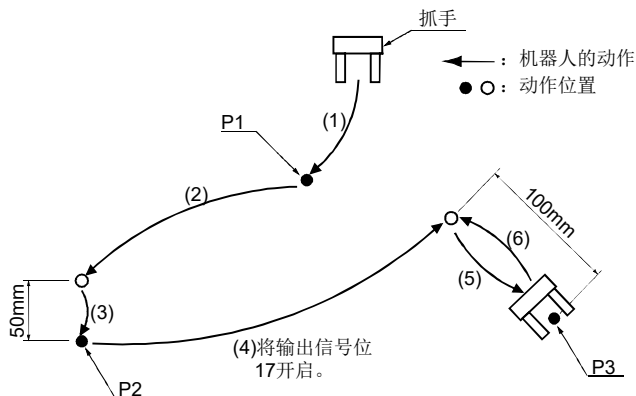
指令语句	说明
Mov	通过关节插补向指定位置移动。可以在 Type 指定插补形式。此外,可以指定 Wth、WthIf 的附随句。

■ 语句示例

语句示例	说明
Mov P1.....	’ 向P1移动。
Mov P1+P2.....	’ 移动到将 P1 和 P2 的各坐标成分加算起来的位置。第 131 页参照。
Mov P1*P2.....	’ 从P1移动到P2已变换的相对位置。第 131 页参照。
Mov P1, -50 (注).....	’ 从P1开始移动到抓手方向后退50mm的位置。
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 WthIf M_In(20)=1, Skip.....	’ 移动至P1中输入信号20变为ON时,则中止移动至P1,前进到下一个。
Mov P1 Type 1, 0 (初始值: 绕道))	’ 各轴动作在超过180°情况下,进行绕道(或抄近路)的指定。

■ 程序示例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定
 示例中, 程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向, 根据每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。请参照另一手册“从机器人本体安装到维护”的“动作的确认”中所显示的坐标系, 指定正确的方向。

• 程序示例

程序	说明
1 Mov P1	’ (1) 向P1移动。
2 Mov P2, -50 (注)	’ (2) 从P2开始移动到抓手方向后退50mm的位置。
3 Mov P2	’ (3) 向P2移动。
4 Mov P3, -100 Wth M_Out(17)=1	’ (4) 从P3开始移动到抓手方向后退100mm的位置, 同时开始输出信号17。
5 Mov P3	’ (5) 向P3移动。
6 Mov P3, -100 (注)	’ (6) 从P3开始返回到抓手方向后退100mm的位置。
7 End	’ 程序结束。

■ 相关功能

功能	说明位置
指定动作速度.....	第 112 页的“(5) 加减速时间和速度控制”
指定加减速时间.....	第 112 页的“(5) 加减速时间和速度控制”
确认到达目的位置.....	第 114 页的“(6) 确认到达目的位置”
不在目标位置停止, 而连续动作至下一个目的位置.....	第 111 页的“(4) 连续动作”
以直线动作.....	第 108 页的“(2) 直线插补动作”
以画圆和圆弧动作.....	第 109 页的“(3) 圆弧插补动作”
在移动指令附加处理.....	第 324 页的“Wth (With)”

(2) 直线插补动作

将抓手前端以直线插补移动到已指定位置。

■ 指令语句

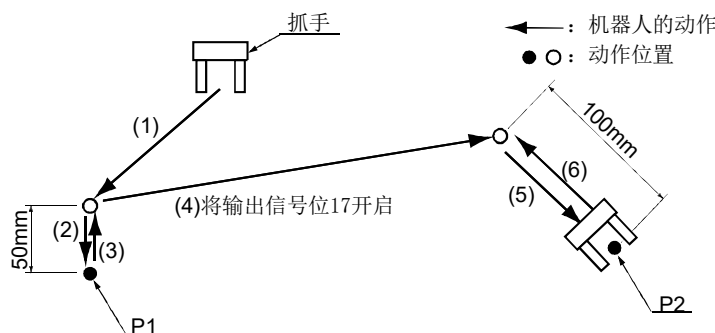
指令语句	说明
Mvs	通过直线插补向指定位置移动。可以在 Type 指定插补形式。可以指定 Wth、WthIf 的附随句。

■ 语句示例

语句示例	说明
Mvs P1.....	’ 向 P1 移动。
Mvs P1+P2.....	’ 移动到将 P1 和 P2 的各坐标成分加算起来的位置。第 131 页参照。
Mvs P1*P2.....	’ 从 P1 移动到 P2 已变换的相对位置。第 131 页参照。
Mvs P1, -50 (注)	’ 从 P1 开始移动到抓手方向后退 50mm 的位置。
Mvs , -50 (注)	’ 从当前位置开始移动到抓手方向后退 50mm 的位置。
Mvs P1 Wth M_Out(17)=1.....	’ 开始向 P1 移动的同时，输出信号位 17 设为 ON。
Mvs P1 WthIf M_In(20)=1, Skip.....	’ 移动至 P1 中输入信号 20 变为 ON 时，则中止移动至 P1，前进到下一个单步。
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。

■ 程序示例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定示例中，程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向，由每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。请参照另一手册“从机器人本体安装到维护”的“动作的确认”中所示的工具坐标系，指定正确的方向。

• 程序示例

程序	说明
1 Mvs P1, -50 (注)	’ (1) 以直线插补从 P1 开始移动到抓手方向后退 50mm 的位置。
2 Mvs P1	’ (2) 以直线插补移动到 P1 移动。
3 Mvs , -50 (注)	’ (3) 以直线插补从当前位置 (P1) 开始移动到抓手方向后退 50mm 的位置。
4 Mvs P2, -100 Wth M_Out(17)=1 (注)	’ (4) 开始移动的同时，输出信号位 17 设为 ON。
5 Mvs P2	’ (5) 以直线插补移动到 P2。
6 Mvs , -100 (注)	’ (6) 以直线插补从当前位置 (P2) 开始移动到抓手方向后退 100mm 的位置。
7 End	’ 程序结束。

■ 相关功能

功能	说明位置
指定动作速度.....	第 112 页的“(5) 加减速时间和速度控制”
指定加减速时间.....	第 112 页的“(5) 加减速时间和速度控制”
确认到达目的位置.....	第 114 页的“(6) 确认到达目的位置”
不在目标位置停止，而连续动作至下一个目的位置.....	第 111 页的“(4) 连续动作”
以关节插补动作.....	第 107 页的“(1) 关节插补动作”
以画圆和圆弧动作.....	第 109 页的“(3) 圆弧插补动作”
在移动指令附加处理.....	第 324 页的“Wth (With)”

(3) 圆弧插补动作

以三次元圆弧插补，在 3 点指定的圆弧上移动。
 当前位置从圆弧开始起点偏离的情况下，直线动作到起点为止后，进行圆弧动作。

■ 指令语句

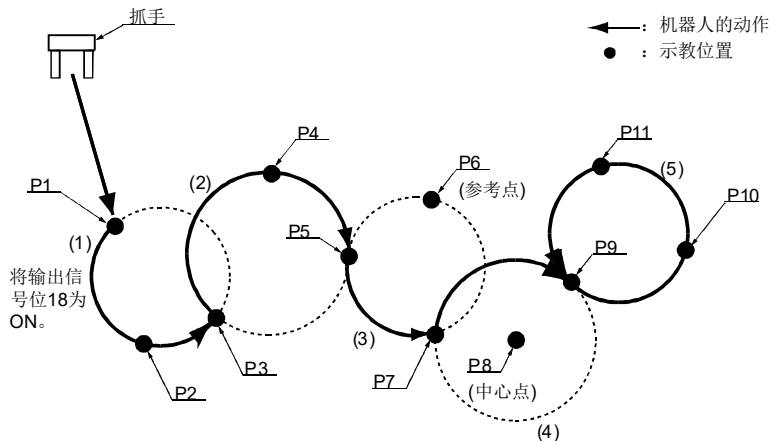
指令语句	说明
Mvr	指定起点、通过点、终点后，以圆弧插补依照起点→通过点→终点的顺序移动。可以在 Type 指定插补形式。可以指定 Wth、WthIf 的附随句。
Mvr2	指定起点、终点、参考点后，以圆弧插补不通过参考点向起点→终点的方向移动。可以在 Type 指定插补形式。可以指定 Wth、WthIf 的附随句。
Mvr3	指定起点、终点、中心点后，以圆弧插补从起点→终点移动。从起点到终点的扇角为 0 度 < 扇角 < 180 度。可以在 Type 指定插补形式。可以指定 Wth、WthIf 的附随句。
Mvc	指定起点（终点）、通过点 1、通过点 2 后，以圆弧插补依照起点→通过点 1→通过点 2→终点的顺序进行圆周移动。可以指定 Wth、WthIf 的附随句。

■ 语句示例

语句示例	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 WthIf 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，则会中止对 P1 的圆弧插补动作，且前进到下一个单步。
Mvr P1, P2, P3 Type 0, 1	' 以圆弧插补移动 P1 → P2 → P3。
Mvr2 P1, P3, P11	' 在 P1 → P3 不通过 P11 的方向进行圆弧插补移动。P11 为参考点。
Mvr3 P1, P3, P10	' 将 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。

■ 程序示例

• 机器人的动作



• 程序示例

程序	说明
1 Mvr P1, P2, P3 Wth M_Out(18)=1	' (1) 将 P1 → P2 → P3 以圆弧动作。由于动作前机器人的当前位置偏离起点，因此最初以直线动作到起点。圆弧动作开始的同时，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) 在起点 (P5)，参考点 (P6)，终点 (P7) 指定的圆周上，从起点开始到终点为止不通过参考点的方向以圆弧动作。
4 Mvr3 P7, P9, P8	' (4) 在中心点 (P8)，起点 (P7)，终点 (P9) 指定的圆周上，从起点开始到终点为止以圆弧动作。
5 Mvc P9, P10, P11	' (5) 将 P9 → P10 → P11 → P9 以圆弧动作。(为 1 周动作。)
6 End	' 程序结束。

■ 相关功能

功能	说明位置
指定动作速度。.....	第 112 页的 “(5) 加减速时间和速度控制”
指定加减速时间。.....	第 112 页的 “(5) 加减速时间和速度控制”
确认到达目的位置。.....	第 114 页的 “(6) 确认到达目的位置”
不在目标位置停止，而连续动作至下一个目的位置。.....	第 111 页的 “(4) 连续动作”
以关节插补动作。.....	第 107 页的 “(1) 关节插补动作”
以直线动作。.....	第 108 页的 “(2) 直线插补动作”
在移动指令附加处理。.....	第 324 页的 “Wth (With)”

(4) 连续动作

不在每个动作位置停止，连续移动多个动作位置。

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。

■ 指令语句

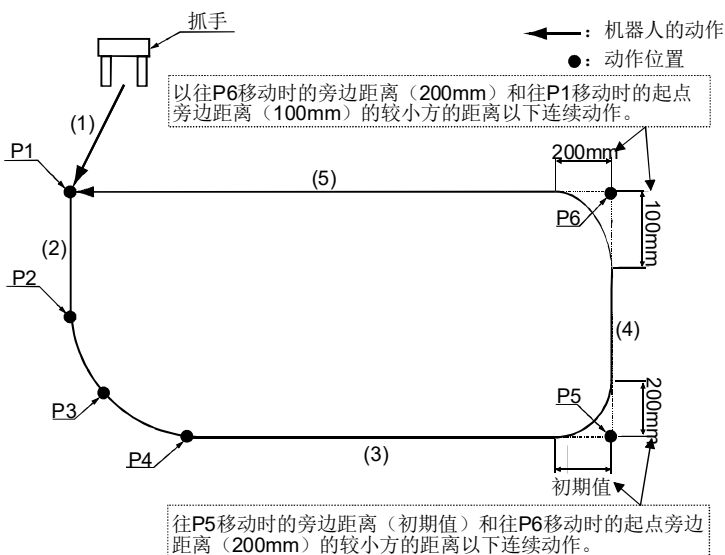
指令语句	说明
Cnt	指定连续动作的开始和结束。

■ 语句示例

语句示例	说明
Cnt 1	' 指定连续动作的开始。起点接近距离为减速开始位置、终点接近开始位置为加速结束位置（因机型而异）。
Cnt 1, 100, 200	' 指定连续动作的开始和指定它的起点接近距离 =100mm、终点接近距离 =200mm。
Cnt 0	' 指定连续动作的结束。

■ 程序示例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定
 示例中，程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向，由每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。请参照另一手册“从机器人本体安装到维护”的“动作的确认”中所显示的工具坐标系，指定正确的方向。

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。尤其是转角部分可能会采取内环的轨迹，因此，开始自动运行时最初应以低速运行，并在留意与外围装置之间的干涉的同时，逐渐提升运行速度。

• 程序示例

程序	说明
1 Mov P1	' (1) 以关节插补向P1移动。
2 Cnt 1	' 将连续动作设为有效。(此后的移动会变为连续动作。)
3 Mvr P2, P3, P4	' (2) 直线动作到P2为止，并连续进行圆弧动作到P4为止。
4 Mvs P5	' 连续进行圆弧动作后向P5直线动作。
5 Cnt 1, 200, 100	' (3) 将连续动作的起点接近距离设定为200mm，终点接近距离设定为100mm。
6 Mvs P6	' (4) 连续移动至上述的P5后，向P6直线动作。
7 Mvs P1	' (5) 连续向P1直线动作。
8 Cnt 0	' 将连续动作设为无效。
9 End	' 程序结束。

■ 相关功能

功能	说明位置
指定动作速度.....	第 112 页的“(5) 加减速时间和速度控制”
指定加减速时间.....	第 112 页的“(5) 加减速时间和速度控制”
确认到达目的位置.....	第 114 页的“(6) 确认到达目的位置”
以关节插补动作.....	第 107 页的“(1) 关节插补动作”
以直线动作.....	第 108 页的“(2) 直线插补动作”
以画圆和圆弧动作.....	第 109 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。

■ 指令语句

指令语句	说明
Accel	以相对于最高加减速度的比例 (%) 指定移动时的加速度和减速度。
Ovrd	以相对于最高速度的比例 (%) 指定程序整体的动作速度。
JOvrd	以相对于最高速度的比例 (%) 指定关节插补动作时的速度。
Spd	以抓手前端速度 (mm/s) 指定直线、圆弧插补动作时的速度。
Oadl	指定最佳加减速功能为有效 / 无效。

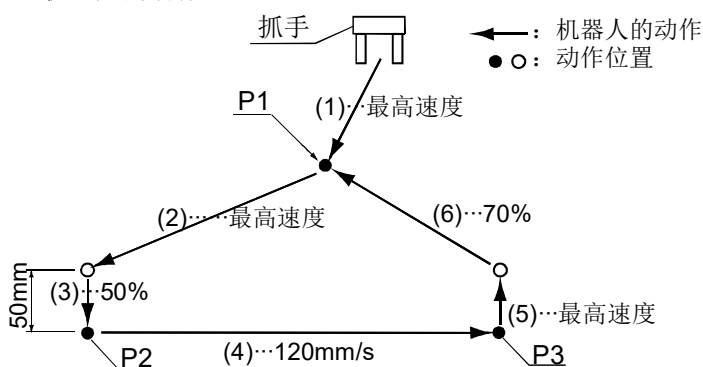
■ 语句示例

语句示例	说明
Accel	’ 加减速全部设定为 100%。
Accel 60, 80.....	’ 加速度设定为 60%，减速度设定为 80%。 (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、圆弧插补动作都设定为最高速度的 50%。
JOvrd 70	’ 将关节插补动作设定为最高速度的 70%。
Spd 30	’ 将直线插补、圆弧插补动作时的速度设定为 30mm/s。
Oadl ON	’ 将最佳加减速功能设为有效。

- 关节插补时的动作速度..... 控制器 (示教单元) 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 (示教单元) 的设定值 × Ovrd 指令的设定值 × Spd 指令的设定值

■ 程序示例

- 机器人的动作



注意 (注) 抓手的前进 / 后退的指定
 示例中, 程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向, 由每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。
 请参照另一手册 “从机器人本体安装到维护” 的 “动作的确认” 中所示的工具坐标系, 指定正确的方向。

- 程序示例

程序	说明
1 Ovrd 100	’ 将全体的动作速度设定为最大。
2 Mvs P1	’ (1) 以最高速度向P1移动。
3 Mvs P2, -50 (注)	’ (2) 以最高速度从P2开始移动到抓手方向后退50mm的位置。
4 Ovrd 50	’ 将全体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, 向P2直线动作。
6 Spd 120	’ 将前端速度设定为120mm/s。(由于倍率修调为50%, 因此实际以60mm/s动作。)
7 Ovrd 100	’ 为了使实际的前端速度为120mm/s, 请将动作速度的比例设为100%。
8 Accel 70, 70	’ 加减速都设定为最高加减速度的70%。
9 Mvs P3	’ (4) 以前端速度120mm/s向P3直线动作。
10 Spd M_NSpd	’ 将前端速度返回至初始值。
11 JOvrd 70	’ 将关节插补动作时的速度设定为70%。

程序	说明
12 Accel	' 加减速都返回至最高加减速。
13 Mvs , -50 (注)	' (5) 以直线动作时的初始设定速度, 从当前位置 (P3) 直线移动到抓手方向后退50mm的位置。
14 Mvs P1	' (6) 以最高速度的70%向P1移动。
15 End	' 程序结束。

■相关功能

功能	说明位置
以关节插补动作。.....	第 107 页的 “(1) 关节插补动作 ”
以直线动作。.....	第 108 页的 “(2) 直线插补动作 ”
以画圆和圆弧动作。.....	第 109 页的 “(3) 圆弧插补动作 ”
不在目标位置停止, 而连续动作至下一个目的位置。.....	第 111 页的 “(4) 连续动作 ”

(6) 确认到达目的位置

以脉冲数可对定位完成条件进行指定。(Fine 指令) 连续动作时, 本指定为无效。

■ 指令语句

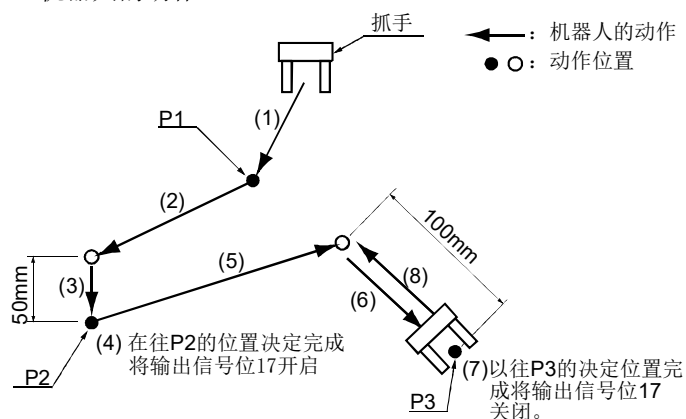
指令语句	说明
Fine	以脉冲数指定定位完成条件。指定的脉冲数越小, 越可以正确定位。
Mov 和 Dly	Mov 的动作指令后, 以 Dly 指令 (定时器) 完成定位。 (皮带驱动方式的机器人为有效。)

■ 语句示例

语句示例	说明
Fine 100.....	' 将定位完成条件设定为 100 脉冲。
Mov P1.....	' 以关节插补向 P1 移动。(以指令值级别完成。)
Dly 0.5.....	' 动作指令后的定位以定时器执行。 (皮带驱动方式的机器人为有效。)

■ 程序示例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定
 示例中, 程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向, 由每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。
 请参照另一手册 “从机器人本体安装到维护” 的 “动作的确认” 中所示的工具坐标系, 指定正确的方向。

• 程序示例

程序	说明
1 Cnt 0	' Fine指令仅在Cnt指令关闭中有效。
2 Mvs P1	' (1) 以关节插补向P1移动。
3 Mvs P2, -50 (注)	' (2) 以关节插补从P2开始移动到抓手方向后退50mm的位置。
4 Fine 50	' 将定位完成脉冲设定为50。
5 Mvs P2	' (3) 以直线插补向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) 以直线从P3开始移动到抓手方向后退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) 关闭输出信号17。
12 Mvs , -100 (注)	' (8) 以直线从当前位置 (P3) 开始移动到抓手方向后退100mm的位置。
13 End	' 程序结束。

■ 相关功能

功能	说明位置
以关节插补动作。.....	第 107 页的 “(1) 关节插补动作”
以直线动作。.....	第 108 页的 “(2) 直线插补动作”
不在目标位置停止, 而连续动作至下一个目的位置。.....	第 111 页的 “(4) 连续动作”

(7) 高轨迹精度控制
可提升机器人的动作轨迹后进行动作。

■ 指令语句

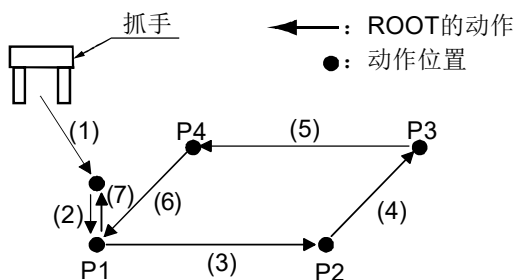
指令语句	说明
Prec	指定高精度模式的有效、无效。

■ 示例

示例	说明
Prec On	将高精度模式设为有效。
Prec Off	将高精度模式设为无效。

■ 程序示例

• 机器人的动作



注意 (注) 抓手的前进/后退的指定
示例中，程序示例为垂直6轴机器人的情况。抓手的前进/后退的方向，由每个机型中确定的工具坐标系的Z轴方向(+/-方向)决定。请参照另一手册“从机器人本体安装到维护”的“动作的确认”中所显示的工具坐标系，指定正确的方向。

• 程序示例

程序	说明
1 Mov P1, -50 (注)	’ (1) 以关节插补从P1开始移动到抓手方向后退50mm的位置。
2 Ovrdr 50	’ 将动作速度设定为最高速度的一半。
3 Mvs P1	’ (2) 直线向P1移动。
4 Prec On	’ 将高轨迹模式设为有效。
5 Mvs P2	’ (3) 以高轨迹精度从P1移动到P2。
6 Mvs P3	’ (4) 以高轨迹精度从P2移动到P3。
7 Mvs P4	’ (5) 以高轨迹精度从P3移动到P4。
8 Mvs P1	’ (6) 以高轨迹精度从P4移动到P1。
9 Prec Off	’ 将高轨迹模式设为无效。
10 Mvs P1, -50	’ (7) 以直线插补从P1开始后退到抓手方向后退50mm的位置。
11 End	’ 程序结束。

注意

Prec 指令可以提升机器人的前端轨迹精度，但由于降低动作的加减速度，因此可能延长节拍时间。
不加入 Cnt 指令精度会提升。但是，无法保证前端的速度。

(8) 抓手 · 工具控制
可指定抓手的开闭及工具的形状。

■ 指令语句

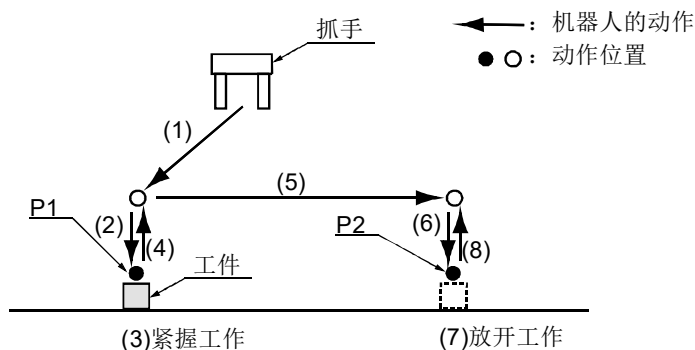
指令语句	说明
HOpen	打开指定抓手。
HClose	关闭指定抓手。
Tool	设定使用工具的形状，并对准控制点。

■ 示例

示例	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法兰面的延长方向 95mm 的位置。

■ 程序示例

• 机器人的动作



注意 (注) 抓手的前进 / 后退的指定
示例中，程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向，由每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。请参照另一手册“从机器人本体安装到维护”的“动作的确认”中所显示的工具坐标系，指定正确的方向。

• 程序示例

程序	说明
1 Tool (0, 0, 95, 0, 0, 0)	' 将抓手长度设定为95mm。
2 Mvs P1, -50 (注)	' (1) 以关节插补从P1开始移动到抓手方向后退50mm的位置。
3 OvrD 50	' 将动作速度设定为最高速度的一半。
4 Mvs P1	' (2) 直线向P1移动。(去抓取工件)
5 Dly 0.5	' 为完成到达目的位置，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 OvrD 100	' 将动作速度设定为最大。
9 Mvs , -50 (注)	' (4) 以直线动作从当前位置 (P1) 开始移动到抓手方向后退50mm的位置。(提起工件)
10 Mvs P2, -50 (注)	' (5) 以关节插补动作从P2移动到抓手方向后退50mm的位置。
11 OvrD 50	' 将动作速度设定为最高速度的一半。
12 Mvs P2	' (6) 直线向P2移动。(去放置工件)
13 Dly 0.5	' 为完成到达目的位置，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 OvrD 100	' 将动作速度设定为最大。
17 Mvs , -50 (注)	' (8) 以直线动作从当前位置 (P2) 开始移动到抓手方向后退50mm的位置。(放开工件)
18 End	' 程序结束。

■ 相关功能

功能	说明位置
附随句.....	第 324 页的“Wth (With)”

4.1.2 托盘运算

将工件规则正确的排列作业 (Palletize) 及取出规则正确排列的工件作业 (Depalletize) 的情况下, 使用托盘功能能够仅示教基准工件的位置, 并通过运算求得剩余的位置。

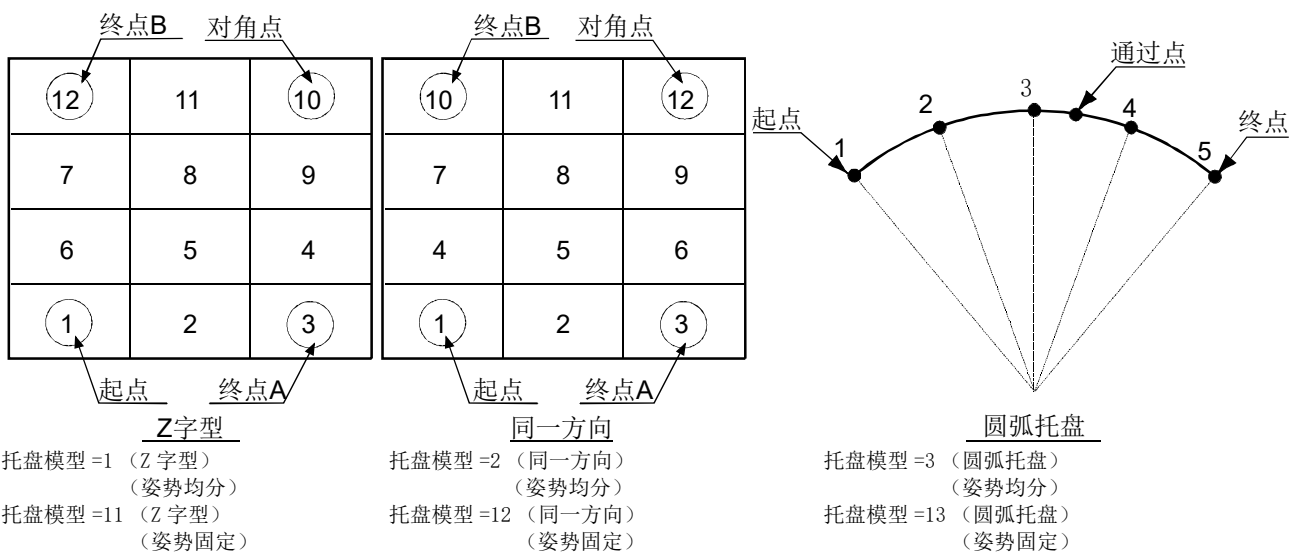
■ 指令语句

指令语句	说明
Def Plt	定义使用的托盘。
Plt	通过运算求得托盘上的指定位置。

■ 示例

示例	说明
Def Plt 1, P1, P2, P3, P4, 4, 3, 1.....	' 定义以起点=P1、终点A=P2、终点B=P3、对角点=P4的4点在指定的位置和大小尺寸中, 托盘编号1存在个数A=4、个数B=3的合计12个(4×3)作业位置, 通过托盘模型=1(Z字型)进行运算并使用。
Def Plt 2, P1, P2, P3, , 8, 5, 2.....	' 定义以起点=P1、终点A=P2、终点B=P3的3点在指定的位置和大小尺寸中, 托盘编号2存在个数A=8、个数B=5的合计40个(5×8)作业位置, 通过托盘模型=2(相同方向)进行运算并使用。
Def Plt 3, P1, P2, P3, , 5, 1, 3.....	' 定义以起点=P1、通过点=P2、终点=P3的3点在指定的圆弧上, 托盘编号3存在合计5个作业位置并使用圆弧托盘。
(Plt 1, 5).....	' 运算托盘编号1的第5个位置。
(Plt 1, M1).....	' 显示数值变量M1的值, 运算托盘编号1内的位置。

位置的指定和托盘模型的关系如下所示。



< 进行托盘定义的位置数据姿势相关的注意事项 >

⚠ 注意

将姿势成分 (A, B, C) 为 ± 180 度附近的位置数据设为托盘定义的 < 起点 >、< 终点 A, B >、< 对角点 > 的情况下, 请确认如下所示 “**说明**” 后使用。

■ 说明

姿势成分 (A, B, C) 变为在 180 度的位置, 即使是相同的姿势, 成分的值也会变为 $+180$ 度或 -180 度的其中一个。这是由于内部的运算误差所造成, 会变为哪一个值并没有规则性。

将此位置设为托盘定义的 < 起点 >、< 终点 >、< 对角点 > 的情况下, 在相同的姿势成分中 $+180$ 度和 -180 度同时存在时, 为了将 -180 度 ~ $+180$ 度之间进行分割并计算, 托盘的格子点的位置上抓手会旋转且会发生预期外的动作。

由于姿势成分的 $+180$ 度和 -180 度变为相同的姿势, 因此在托盘定义中使用的位置数据, 请在备齐符号 + 或 - 的其中一个后使用。

此外, 姿势成分在 ± 180 度附近 (例如, $+179$ 度和 -179 度) 的情况下, 也会出现符号同时存在的现象。此情况下, 在姿势成分里加算或减算 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, 补偿为加算 360 度的 $+181$ 度。)

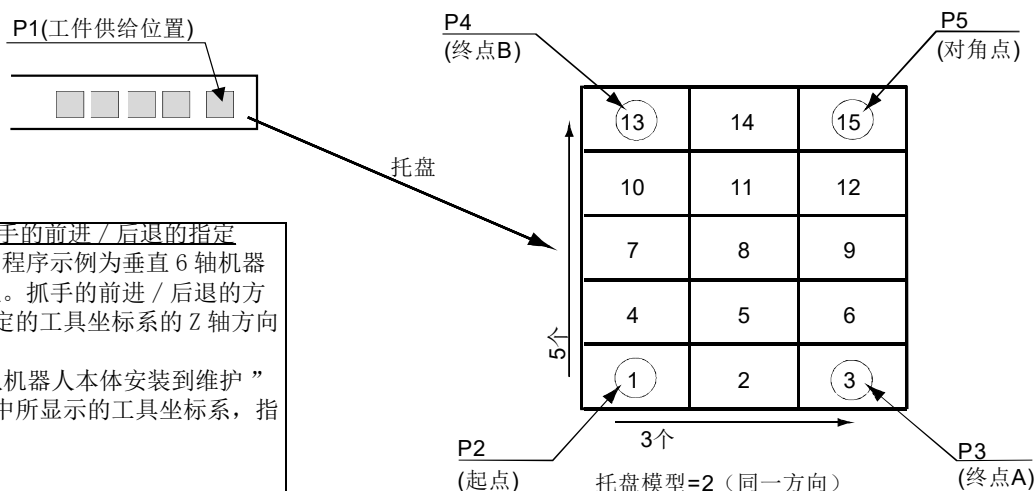
“**程序示例 1**” 那么, 在托盘的全格子点中, 在抓手的方向相同的情况下 (A, B, C 轴的值相同), 将终点 (P3、P4) 和对角点 (P5) 的姿势成分对准起点 (P2) 的示例 (行编号 10 ~ 90 号) 和 “**程序示例 2**” 中, 在托盘定义位置的姿势成分为 ± 180 度附近, 终点 (P3、P4) 和对角点 (P5) 的 C 轴的值小于 -178 度, 或大于 $+178$ 度的情况下, 在和起点 (P2) 相同符号中补偿值的示例 (行编号 10 ~ 100) 如下所示。(将 ± 178 度设为补偿的限界值) 请设为在托盘的精度不良、必须略微补偿抓手的方向情况下的参考。

此外, 在托盘模型中指定 11 ~ 13 时, 通过托盘运算求得的位置变量的姿势数据

< 起点 > 的姿势将被代入并得以固定抓手的方向。(指定 1 ~ 3 情况下的姿势数据, 会被代入将 < 起点 > - < 终点 > 之间进行均分的值)

■ 程序示例

- 机器人的动作



注意

(注) 抓手的前进 / 后退的指定示例中, 程序示例为垂直 6 轴机器人的情况。抓手的前进 / 后退的方向, 由每个机型中确定的工具坐标系的 Z 轴方向 (+/- 方向) 决定。请参照另一手册 “从机器人本体安装到维护” 的 “动作的确认” 中所显示的工具坐标系, 指定正确的方向。

⚠ 注意

依据托盘运算 (P1t 指令) 所算出格子点的构造标志 (位置数据的 F11), 采用托盘定义的起点的值。

因此, 使用托盘定义的各点中构造标志不同的位置数据时, 会变为和希望的托盘动作不同的动作。

在托盘定义的起点, 终点 A、B、对角点中, 请使用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 F12), 也采用托盘定义的起点的值。在托盘定义的各点中使用多旋转标志不同的位置数据时, 根据经由托盘动作的机器人位置及插补指令的种类 (关节插补、直线插补等), 抓手会旋转并发生预期外的动作。届时请使用插补指令的自变量 Type, 适当地将姿势的绕远道 / 抄近道动作进行设定, 将抓手调整为希望的动作。

• 程序示例 1

托盘的全格子点中, 抓手的方向相同时 (A、B、C 轴的值相同)

程序	说明
1 P3.A=P2.A	' 在P3的姿势成分 (A) 中代入P2的姿势成分 (A)。
2 P3.B=P2.B	' 在P3的姿势成分 (B) 中代入P2的姿势成分 (B)。
3 P3.C=P2.C	' 在P3的姿势成分 (C) 中代入P2的姿势成分 (C)。
4 P4.A=P2.A	' 在P4的姿势成分 (A) 中代入P2的姿势成分 (A)。
5 P4.B=P2.B	' 在P4的姿势成分 (B) 中代入P2的姿势成分 (B)。
6 P4.C=P2.C	' 在P4的姿势成分 (C) 中代入P2的姿势成分 (C)。
7 P5.A=P2.A	' 在P5的姿势成分 (A) 中代入P2的姿势成分 (A)。
8 P5.B=P2.B	' 在P5的姿势成分 (B) 中代入P2的姿势成分 (B)。
9 P5.C=P2.C	' 在P5的姿势成分 (C) 中代入P2的姿势成分 (C)。
10 Def P1t 1, P2, P3, P4, P5, 3, 5, 2	' 定义托盘。 托盘编号=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中代入值1。(M1在计数器使用)
12 *LOOP	' 作为跳转对象, 指定标签LOOP。
13 Mov P1, -50 (注)	' 以关节插补动作从P1移动到抓手方向后退50mm的位置。
14 OvrD 50	' 将动作速度设定为最高速度的一半。
15 Mvs P1	' 向P1直线移动。(去抓取工件)
16 HC1ose 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)	' 以直线动作从当前位置 (P1) 移动到抓手方向后退50mm的位置。 (提起工件)
20 P10=(P1t 1, M1)	' 显示数值变量M1的值, 运算托盘编号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开始移动到抓手方向后退50mm的位置。
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 H0pen 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)	' 以直线动作从当前位置 (P10) 移动到抓手方向后退50mm的位置。 (放开工件)
28 M1=M1+1	' 在数值变量M1的值补足1。(将托盘计数器前进)
29 If M1<=15 Then *LOOP	' 数值变量M1的值在15以下时, 跳转至标签LOOP后重复处理。否则前进至下一行。
30 End	' 程序结束。

• 程序示例 2
姿势成分为 ±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的姿势成分(C)的符号,为-(负)时,跳转到标签MINUS。
2 If Deg(P3.C)<-178 Then P3.C =P3.C+Rad(+360)	' P3的姿势成分为-180度附近的情况下,加算360度后补偿为+值。
3 If Deg(P4.C)<-178 Then P4.C =P4.C+Rad(+360)	' P4的姿势成分为-180度附近的情况下,加算360度后补偿为+值。
4 If Deg(P5.C)<-178 Then P5.C=P5.C+Rad(+360)	' P5的姿势成分为-180度附近的情况下,加算360度后补偿为+值。
5 GoTo *DEFINE	' 无条件跳转到标签DEFINE的行。
6 *MINUS	' 作为跳转对象,指定标签MINUS。
7 If Deg(P3.C) > +178 Then P3.C =P3.C-Rad(+360)	' P3的姿势成分为+180度附近的情况下,加算360度后补偿为-值。
8 If Deg(P4.C) > +178 Then P4.C =P4.C-Rad(+360)	' P4的姿势成分为+180度附近的情况下,加算360度后补偿为-值。
9 If Deg(P5.C) > +178 Then P5.C =P5.C-Rad(+360)	' P5的姿势成分为+180度附近的情况下,加算360度后补偿为-值。
10 *DEFINE	' 作为跳转处,指定标签DEFINE。
11 Def P1t 1, P2, P3, P4, P5, 3, 5, 2	' 定义托盘。 托盘编号=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)
12 M1=1	' 在数值变量M1中代入值1。 (M1在计数器使用)
13 *LOOP	' 作为跳转对象,指定标签LOOP。
14 Mov P1, -50 (注)	' 以关节插补动作从P1移动到抓手方向后退50mm的位置。
15 OvrD 50	' 将动作速度设定为最高速度的一半。
16 Mvs P1	' 向P1直线移动。(去抓取工件)
17 HClose 1	' 关闭抓手1。(抓住工件)
18 Dly 0.5	' 等待0.5秒。
19 OvrD 100	' 将动作速度设定为最大。
20 Mvs , -50 (注)	' 以直线动作从当前位置(P1)移动到抓手方向后退50mm的位置。(提起工件)
21 P10=(P1t 1, M1)	' 显示数值变量M1的值,运算托盘编号1内的位置,将结果代入P10。
22 Mov P10, -50 (注)	' 以关节插补动作从P10开始移动到抓手方向后退50mm的位置。
23 OvrD 50	' 将动作速度设定为最高速度的一半。
24 Mvs P10	' 以直线移动到P10。(去放置工件)
25 HOpen 1	' 打开抓手1。(放开工件)
26 Dly 0.5	' 等待0.5秒。
27 OvrD 100	' 将动作速度设定为最大。
28 Mvs , -50	' 以直线动作从当前位置(P10)移动到抓手方向后退50mm的位置。(放开工件)
29 M1=M1+1	' 在数值变量M1的值补足1。 (将托盘计数器前进)
30 If M1<=15 Then *LOOP	' 数值变量M1的值在15以下时,跳转至标签LOOP后重复处理。否则前进至下一行。
31 End	' 程序结束。

4.1.3 程序控制

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

可以将程序的流动向指定行进行无条件，或条件判断后分支。

■ 指令语句

指令语句	说明
GoTo	在指定的标签无条件跳转。
On GoTo	根据指定变量的值进行跳转。值的条件为整数顺序。(1, 2, 3, 4, ……)
If Then Else (在 1 行的记述)	根据指定条件执行指令。值的条件可任意的指定。 条件成立的情况下，执行 Then 后面的记述，条件不成立的情况下，执行 Else 后面的记述。 用一行记述。
If Then Else EndIf (多行记述)	根据指定变量及其值的指定条件进行多行处理。值的条件可任意的指定。条件的种类，每 1 指令为 1 个种类。 条件成立的情况下，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 的下一行开始到 EndIf 为止的行。
Select Case End Select	根据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 指令可以指定多个种类。
Wait	变量变为指定的值前待机。

■ 示例

示例	说明
GoTo *FIN	' 无条件跳转到标签 FIN 的行。
On M1 GoTo *L1, *L2, *L3	' 数值变量 M1 的值为 1 时，跳转到标签 L1，为 2 时，跳转到标签 2，为 3 时，跳转到标签 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 时，往标签 *L1 分支，否则前往下一行。
If M1=1 Then *L10 Else *L20	' 数值变量 M1 的值为 1 时，往标签 *L10 分支，否则会分支为标签 *L20。
If M1=1 Then	' 数值变量 M1 的值为 1 时，执行 M2=1、M3=2，M1 为 1 以外时，执行 M2=-1、M3=-2。
M2=1	
M3=2	
Else	
M2=-1	
M3=-2	
EndIf	
Select M1	' 依据数值变量 M1 的值，分支到适当的 Case 文。
Case 10	' 值为 10 时，仅在 Case 10 和下一个的 Case IS 11 之间执行。
:	
Break	
Case 11	' 值为 11 时，仅在 Case 11 和下一个的 Case IS < 5 之间执行。
:	
Break	
Case IS < 5	' 值小于 5 时，仅在 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 时，仅在 Case 6 TO 9 和下一个的 Default 之间执行。
:	
Break	
Default	' 值不符合上述任意值时，仅在 Default 和下一个 End Select 之间执行。
:	
Break	
End Select	' Select Case 文结束。
Wait M_In(1)=1	' 输入信号位 1 开启前待机。

■ 相关功能

功能	说明位置
循环.....	第 123 页的 “(2) 循环 ”
中断.....	第 124 页的 “(3) 中断 ”
子程序.....	第 125 页的 “(4) 子程序 ”
外部信号的输入.....	第 128 页的 “(1) 信号输入 ”

(2) 循环

可根据指定条件，重复执行多个指令。

■ 指令语句

指令语句	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	在 While 文和 WEnd 文之间，满足指定条件期间重复执行。

■ 示例

示例	说明
For M1=1 To 10 : Next	将 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	将 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	说明位置
无条件分支 · 条件分支.....	第 121 页的“(1) 无条件分支 · 条件分支 · 待机”
插入.....	第 124 页的“(3) 中断”
输入信号等待.....	第 128 页的“(1) 信号输入”

(3) 中断

指定条件成立的话，可中断执行中的指令，并使其向指定行分支。

■ 指令语句

指令语句	说明
Def Act	定义插入条件和插入发生时的处理。
Act	指定插入的允许 / 禁止。
Return	将插入处理视为子程序的调用时，返回至原来插入的行。

■ 示例

示例	说明
Def Act 1, M_In(10)=1 GoSub *L100	' 定义在插入编号 1 中输入信号位 10 开启时，在机器人减速停止后调用标签 *L100 的子程序。减速时间依赖于 Accel 指令及 Ovrld 指令。
Def Act 2, M_In(11)=1 GoSub *L200, L.....	' 定义插入编号 2 中输入信号位 11 开启的话，执行中的指令完成后调用标签 *L200 的子程序。
Def Act 3, M_In(12)=1 GoSub *L300, S.....	' 定义插入编号 3 中输入信号位 12 开启的话，将机器人在最短时间、最短距离减速停止后调用标签 *L300 的子程序。
Act 1=1.....	' 允许优先编号 1 的插入。
:	
Act 2=0.....	' 禁止优先编号 2 的插入。
:	
*L100	' 返回发生插入的行。
:	
Return 0.....	
*L200	' 返回发生插入的下一行。
:	
Return 1.....	
*L300	' 返回发生插入的行。
:	
Return 0.....	

■ 相关功能

功能	说明位置
无条件分支 · 条件分支。.....	第 121 页的“(1) 无条件分支 · 条件分支 · 待机”
子程序。.....	第 125 页的“(4) 子程序”
通信。.....	第 129 页的“4.1.5 通信”

(4) 子程序

可使用子程序及子程序。

通过使用本功能共享程序并节省单步数，以便将程序做成阶层构造使其更浅显易懂。

■ 指令语句

指令语句	说明
GoSub	调用指定标签的子程序。
On GoSub	调用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	返回到以 GoSub 指令调用的下一行。
CallP	调用已指定的程序。以被调用的程序的 End 指令返回到原来程序的下一行。可以作为自变量将数据传送到要调用的程序。
FPrm	在以 CallP 指令调用的程序中接收自变量。

■ 示例

示例	说明
GoSub *GET	' 调用来自标签 GET 的子程序。
On M1 GoSub *L1, *L2, *L3	' 数值变量 M1 的值为 1 时调用标签 L1 的子程序，为 2 的话调用标签 L2 的子程序，为 3 的话调用标签 L3 的子程序。
:	在值不相当时，往下一行前进。
:	
*L1	
:	
Return	
*L2	
:	
Return	
*L3	
:	
Return	' 使用 GoSub 指令调用后，后退到下一个行。
Return	' 使用 GoSub 指令调用后，后退到下一个行。
CallP "10"	' 调用 10 号的程序。
CallP "20", M1, P1	' 在向 20 号的程序中传送数值变量 M1 和位置变量 P1 后进行调用。
FPrm M10, P10	' 在子程序中，以 CallP 指令将被传送的数值变量接收为 M10、位置变量接收为 P10。

■ 相关功能

功能	说明位置
中断	第 124 页的 “(3) 中断”
通信	第 129 页的 “4.1.5 通信”
无条件分支	第 121 页的 “(1) 无条件分支 · 条件分支 · 待机”

(5) 定时器

可以仅等待指定时间，且可以在指定时间宽度内对输出信号进行脉冲输出。

■ 指令语句

指令语句	说明
Dly	发挥指定时间的定时器功能。

■ 示例

示例	说明
Dly 0.05.....	' 仅等待 0.05 秒。
M_Out(10)=1 Dly 0.5.....	' 仅在 0.5 秒间开启输出信号位 10。

■ 相关功能

功能	说明位置
脉冲信号输出.....	第 128 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。

■ 指令语句

指令语句	说明
Hlt	将机器人停止，且中断程序的执行。启动由下一个行执行。
End	定义程序的 1 循环的结束。连续运行时，在 End 指令执行再度由起始行启动。循环运行时、循环停止时以 End 指令结束程序。

■ 示例

示例	说明
Hlt	' 中断程序的执行。
If M_In(20)=1 Then Hlt	' 输入信号位 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。

■ 相关功能

功能	说明位置
附随句	第 324 页的“Wth (With)”

4.1.4 外部信号的输入输出

对机器人由外部机器（可编程控制器等）控制的情况下最普遍的信号控制方法进行说明。

(1) 信号输入

可读取由可编程控制器等外部机器输入的信号。

使用机器人（系统）状态变量（M_In()等）确认输入信号。关于机器人（系统）状态变量的详细内容请参照第333页的“4.13 机器人（系统）状态变量的详细说明”。

■ 指令语句

指令语句	说明
Wait	输入信号变为指定状态前待机。

■ 系统状态变量（机器人状态变量）

M_In, M_Inb, M_Inw, M_DIn

■ 示例

示例	说明
Wait M_In(1)=1.....	’ 输入信号位1开启前待机。
M1=M_Inb(20).....	’ 在数值变量M1里将输入信号位20到27之间的8位的状态转为数值代入。
M1=M_Inw(5).....	’ 在数值变量M1里将输入信号位5到20之间的16位的状态转为数值代入。

■ 相关功能

功能	说明位置
信号输出.....	第128页的“(2) 信号输出”
依输入信号分支.....	第121页的“(1) 无条件分支·条件分支·待机”
依输入信号插入.....	第124页的“(3) 中断”

(2) 信号输出

可以将信号输出到可编程控制器等的外部机器。

使用机器人（系统）状态变量（M_Out()等）确认输入信号。关于机器人（系统）状态变量的详细内容请参照第162页的“4.5 机器人（系统）状态变量”。

■ 指令语句

指令语句	说明
Clr	以参数的输出信号复位为模式基础，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

■ 示例

示例	说明
Clr 1	’ 以输出复位模式为基础清除。
M_Out(1)=1	’ 将输出信号位1开启。
M_Outb(8)=0	’ 将输出信号位8到15之间的8位关闭。
M_Outw(20)=0	’ 将输出信号位20到35之间的16位关闭。
M_Out(1)=1 Dly 0.5	’ 在0.5秒间开启输出信号位1。（脉冲输出）
M_Outb(10)=&H0F	’ 将输出信号位10到13之间的4位开启，14到17之间的4位关闭。

■ 相关功能

功能	说明位置
信号输入.....	第128页的“(1) 信号输入”
定时器.....	第126页的“(5) 定时器”

4.1.5 通信

可以和计算机等的外部机器间进行数据通信。

■ 指令语句

指令语句	说明
Open	将通信线路开启。
Close	将通信线路关闭。
Print#	将数据以 ASCII 形式输出。作为结束 CODE 将 CR 输出。
Input#	将数据以 ASCII 形式输入。作为结束 CODE 将 CR 输出。
On Com GoSub	在从通信线路发生插入时，定义所调用子程序。从外部机器输入数据时发生插入。
Com On	允许从通信线路的插入处理。
Com Off	禁止从通信线路的插入处理。即使有插入也会变为无效。
Com Stop	停止从通信线路的插入处理。即使有插入的情况下也会记住，得到允许后执行。

■ 示例

示例	说明
Open "COM1: " As #1.....	' 将通信线路 COM1 作为文件编号 1 开启。
Close #1.....	' 将文件编号 1 关闭。
Close.....	' 将已开启的文件全部关闭。
Print# 1, "TEST".....	' 将字符串 "TEST" 输出到文件编号 1。
Print# 2, "M1="; M1.....	' 将在字符串 "M1=" 连续 M1 的值输出到文件编号 2。 输出数据示例: "M1="+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编号 3。 输出数据示例: "(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编号 1。 M5 和 P5 以逗号 (16 进 2C) 区分。 输出数据示例: "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。 输入数据示例: "8"+CR (代入值 8 的情况)
Input# 1, P10.....	' 将输入的数据转为坐标值，代入位置变量 P10。 输入数据示例: "(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，代入数值变量 M8，将逗号后下一个数据设为坐标值，代入位置变量 P6。M8 和 P6 以逗号 (16 进 2C) 区分。 输入数据示例: "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.....	' 数据输入到通信线路 COM2 时，定义所调用标签 RECV 的行的子程序。
Com(1) On.....	' 允许由通信线路 COM1 的插入。
Com(2) Off.....	' 禁止 (无效) 通信线路 COM2 的插入。
Com(1) Stop.....	' 停止 (保留) 通信线路 COM1 的插入。

■ 相关功能

功能	说明位置
子程序.....	第 125 页的“(4) 子程序”
中断.....	第 124 页的“(3) 中断”

4.1.6 公式和运算

可以使用的运算符及其意义、示例如下所示。

(1) 运算符一览

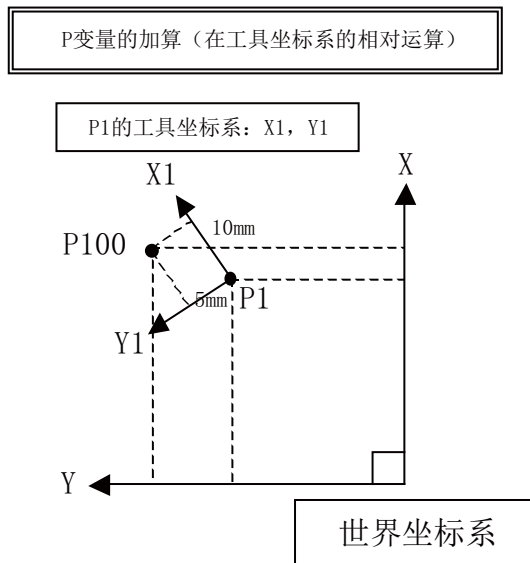
区分	运算符	意义	示例	
代入	=	将右边代入左边	P1=P2 ; 将 P2 代入位置变量 P1。 P5=P_Curr ; 将当前的坐标值代入当前位置变量 P5。 P10.Z=100.0 ; 将位置变量 P10 的 Z 坐标值设为 100.0。 M1=1 ; 将值 1 代入数值变量 M1。 STS\$="OK" ; 将 OK 字符串代入字符串变量 STS\$。	
数值运算	+	加算	P10=P1+P2 注1) ; 将 P1 和 P2 各坐标成分加算结果代入位置变量 P10。 Mov P8+P9 ; 移动到将位置变量 P8 和 P9 各坐标成分加算的位置。 M1=M1+1 ; 数值变量 M1 的值加 1。 STS\$="ERR"+"001" ; 将 ERR 字符串和 001 字符串结合后代入在字符串变量 STS\$。	
	-	减算	P10=P1-P2 ; 将 P1 到 P2 各坐标成分加算结果代入位置变量 P10。 Mov P8-P9 ; 移动到将位置变量从 P8 到 P9 的各坐标成分减算的位置。 M1=M1-1 ; 从数值变量 M1 的值减去 1。	
	*	乘算	P1=P10*P3 注2) ; 将从 P10 到 P3 的相对转换结果代入位置变量 P1。 M1=M1*5 ; 将数值变量 M1 的值乘以 5 倍。	
	/	除算	P1=P10/P3 ; 将从 P10 到 P3 的相对转换结果代入位置变量 P1。 M1=M1/2 ; 将数值变量 M1 的值除以 2。	
	^	指数运算	M1=M1 ^ 2 ; 数值变量 M1 的值的平方。	
	\	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 的各坐标成分符号反转。 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 时, 往标签 L2 分支。 If STS\$="OK" Then *L1 ; 若字符串变量 STS\$ 呈 OK 字符串, 则往标签 L1 分支。	
	<>	比较是否不相等	If M1<>2 Then *L3 ; 若数值变量 M1 的值不为 2 时, 往标签 L3 分支。 If STS\$<>"OK" Then *L9 ; 若字符串变量 STS\$ 不呈 OK 字符串, 则往标签 L9 分支。	
	<	比较是否小于	If M1<10 Then *L3 ; 数值变量 M1 的值小于 10 时, 往标签 L3 分支。 If Len(STS\$)<3 Then *L1 ; 字符串变量 STS\$ 的字符数小于 3 时, 往标签 L1 分支。	
	>	比较是否大于	If M1>9 Then *L2 ; 数值变量 M1 的值大于 9 时, 往标签 L2 分支。 If Len(STS\$)>2 Then *L3 ; 字符串变量 STS\$ 的字符数大于 2 时, 往标签 L3 分支。	
	=<	比较是否小于等于	If M1<=10 Then *L2 ; 数值变量 M1 的值为 10 或小于 10 时, 往标签 L2 分支。 If Len(STS\$)<=5 Then *L3 ; 字符串变量 STS\$ 的字符数为 5 或小于 5 时, 往标签 L3 分支。	
	=>	比较是否大于等于	If M1>=11 Then *L2 ; 数值变量 M1 的值为 11 或大于 11 时, 往标签 L2 分支。 If Len(STS\$)>=6 Then *L3 ; 字符串变量 STS\$ 的字符数为 6 或大于 6 时, 往标签 L3 分支。	
	逻辑运算	And	逻辑乘运算	M1=M_Inb(1) And &H0F ; 将输入信号位 1 到 4 的状态转为数值后代入数值变量 M1。(输入信号位 5 到 8 的状态为关闭。)
	Or	逻辑加运算	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 会变为常开。	
Not	否定运算	M1=Not M_Inw(1) ; 将输入信号位 1 到 16 的状态反转, 并转为数值后代入数值变量 M1。		
Xor	异或逻辑加运算	M2=M1 Xor M_Inw(1) ; 在数值变量 M2, 将 M1 和输入信号位从 1 到 16 的状态的异或逻辑加以数值代入。		
<<	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) 第 131 页的“位置数据的相对运算(加算)”请参照。

注 2) 第 131 页的“位置数据的相对运算(乘算)”请参照。

(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，要执行坐标转换。以下对简单的示例进行说明。



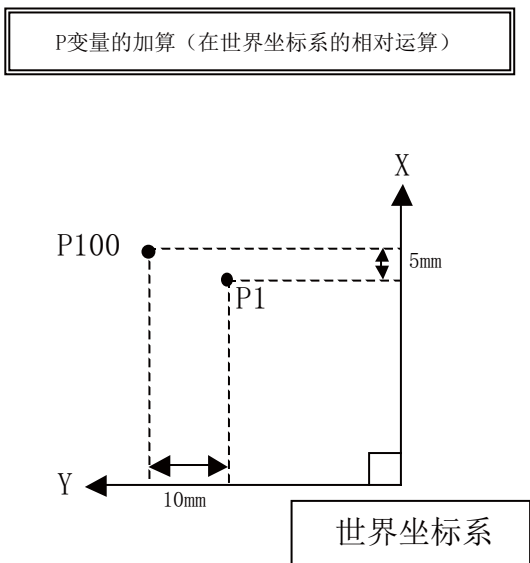
相对运算的示例（乘算）

```

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)
    
```

此示例为在示教位置 P1 中 P1 的工具坐标系内使其相对移动情况下的示例。
 P2 的 X、Y 的值变为在工具坐标系内的移动量。P 变量的乘算会变为相对运算。
 请注意依照乘算的顺序，否则意义会不同。
 设定了相对移动量的变量（P2）会在后面记述。
 P2 的姿势轴部分（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 的姿势后退变为在 Z, Y, X 旋转（C, B, A 顺序）里旋转姿势。乘算为在工具坐标系内的加算，但除算的情况下，会变为在工具坐标系内的减算。

(3) 位置数据的相对运算（加算）



相对运算的示例（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)
    
```

此示例为在示教位置 P1 中世界坐标系内使其相对移动情况下的示例。
 P2 的 X、Y 的值会变为在世界坐标系内的移动量。P 变量的加算会变为相对运算。
 在 P2 的 C 轴部分，输入值时，也可变更 P100 的 C 轴的旋转成分。会变成在 P1 的 C 轴成份加算 P2 的 C 轴成份后的值。

注意) 在上列示例中，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，工具坐标系会因机器人的姿势而异。
 注意) 由于 5 轴机器人无法正确进行相对运算，因此无法使用。

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。
WthIf	在移动指令附加条件处理。

■ 示例

示例	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 WthIf M_In(20)=1, Hlt	' 往 P1 的移动中，输入信号位 20 为开启时，则会停止。
Mov P1 WthIf M_In(19)=1, Skip	' 往 P1 的移动中，输入信号位 19 为开启时，中断往 P1 的移动，往下一行前进。

■ 相关功能

功能	说明位置
关节插补动作.....	第 107 页的 “(1) 关节插补动作”
直线插补动作.....	第 108 页的 “(2) 直线插补动作”
圆弧插补动作.....	第 109 页的 “(3) 圆弧插补动作”
停止.....	第 127 页的 “(6) 停止”

4.2 多任务功能

4.2.1 所谓多任务功能

在此对多任务功能进行说明。

所谓多任务为多个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任务插槽（全部有 32 个，出货时设定为 8 个）内执行。

多任务执行的开始，通过执行操作面板开始的起动及专用输入信号开始的起动、或多任务相关指令后进行。

图 4-1 显示多任务的执行环境。

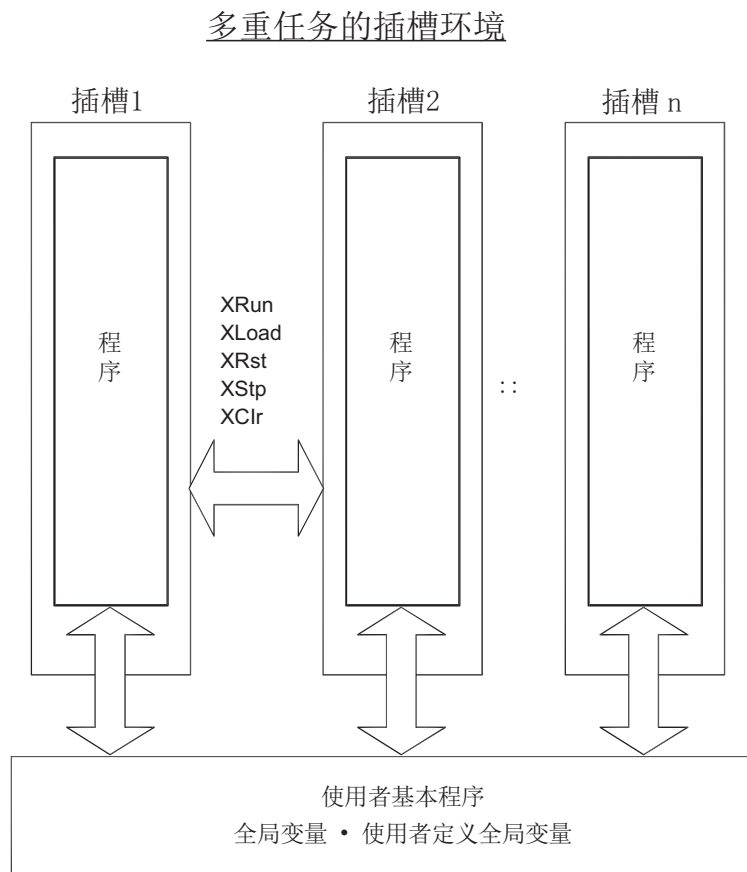


图 4-1：多任务的执行环境

◇◆◇关于程序的执行◇◆◇

多任务的执行，是指将要执行的程序放入“任务插槽”内执行。

例如，在运行一个程序时（通常在示教单元的〈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，默认放入任务插槽 1 中执行。

4.2.2 多任务的执行方法

多任务的执行方法如表 4-2 所示，有 3 个种类。

表 4-2：多任务的执行方法

	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 VI 指令，开始程序的并列执行的方法。可以指定并列执行程序及停止并列执行中的程序。这个方法，依程序的流程，在选择并列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 173 页的“4.12 指令的详细说明”各页。
2	从示教单元的 < 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（启动条件为“START”、或者“ERROR”），由于起动操作，并列执行的开始及始终并列执行、或在错误发生时开始并列执行的方法。需要预先设定参数“SLT*”。以此方法，不会依存程序的流程，以预先决定的形状同时执行，或是依次执行时有效。
3	在电源开启时自动执行	可以在控制器的电源开启后始终执行。在 SLT* 参数的起动条件上，指定为“ALWAYS”的程序在控制器的电源开启后，会始终执行。依据此点，输入输出信号监视用的任务插槽等，不需要由可编程控制器侧开始启动。此外，也可以将动作用的程序由始终执行的程序执行。该情况下，通过将参数的“ALWENA”值设为 1，即可从始终执行程序内执行 XRun, Xload 等的 X** 指令及 Servo 指令、Reset 指令。

4.2.3 各任务插槽运行状态

依据操作及指令，各任务插槽的运行状态的状态变化如图 4-2 所示。各自的状态，可以通过机器人（系统）状态变量及外部输出信号确认。

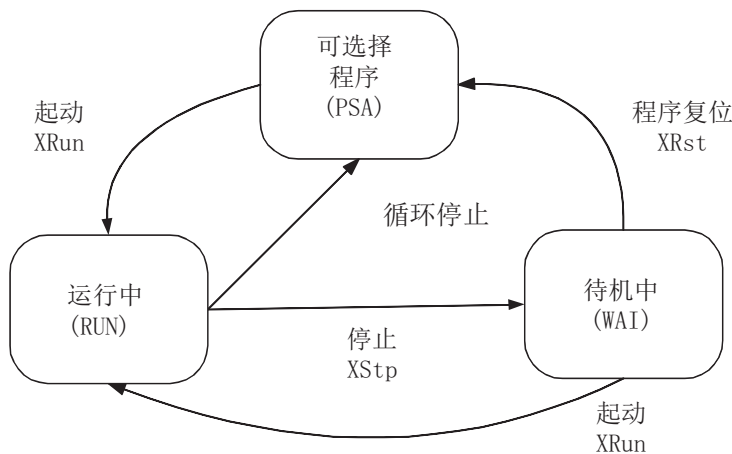


图 4-2：各任务插槽运行状态

< 关于任务插槽相关参数 >

每个任务插槽有 32 个 SLT1 ~ SLT32（出厂时 8 个），用于设定使用的程序名称、运行形态、启动条件、优先级。

参数的详细内容请参照第 468 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启动条件、④优先级

■ 各设定值及其意义

参数的项目	初始值	设定值	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	在多任务运行里执行已决定的程序的情况下，以参数设定。 视其状况，要执行的程序变为可变的情况下，可在程序里通过 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，且选择状态被解除。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 468 页的“5 参数的设定功能” SLOTON 项目。
③启动条件	START	• START: 依据示教单元的 <操作面板> 的 IO 的 START 信号执行	试图正常启动时设定 START。 ^{注1)}
		• ALWAYS: 在控制器的电源开启时执行	试图始终执行时使用 ALWAYS。 但是，无法执行 Mov 等的动作指令。始终执行程序停止可以使用 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: 在控制器变为错误状态时执行	ERROR 是在错误发生时试图执行的情况下指定。无法执行 Mov 指令等让机器人动作的指令。运行形态（REP/CYC）与其设定值无关，都会变为 1 循环运行（CYC）。
(4) 优先级 (优先执行行数)	1	1 ~ 31: 在多任务运行时，同时执行的行数	数字越大时，任务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，之后会变为重复执行。

注1) 从示教单元的<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”的全部任务插槽的程序将同时开始执行。

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。该情况下，在同专用输入输出参数里，预先将信号编号分配使用。关于专用输入输出的分配的详细内容，请参照本书 / 第 605 页的“6. 2PLC 连接输出功能”。

■ 设定示例

在任务插槽 2 中进行指定的情况下的参数设定示例如下所示。

指定内容) 程序名称: 5

运行形态: 连续运行

启动条件: 始终执行

优先级 : 5

SLT2=5, REP, ALWAYS, 5

4.2.4 多任务程序创建上的注意点

(1) 任务个数和处理时间的关系

多任务时，看起来像是将多个机器人程序并列处理，但实际上是一边切换程序一边执行每 1 行（行数可变更，请参照第 468 页的“5 参数的设定功能”的“SLTn”参数）。因此，任务个数增加时，全体的程序执行时间会延长。使用多任务的情况下，请将任务个数降到必要的最小限数。

但是，动作指令（Mov 及 Mvs 指令等）执行中会随时处理其它任务的程序。

(2) 指定并列执行程序个数的最大值

并列执行程序个数，是以参数“TASKMAX”（请参照第 468 页的“5 参数的设定功能”）设定。并列执行（初始值为 8）9 个以上的程序的情况下，请变更此参数。

(3) 在外部变量的程序间的数据的传输方法

在多任务运行程序间的数据交接，请使用 M_00 及 P_00 等的程序外部变量（参照第 151 页的“4.3.23 程序外部变量”）及用户定义外部变量（参照第 152 页的“4.3.24 用户定义外部变量”）执行。

示例如下所示。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外部变量的 M_00，传达到任务插槽 1 的程序。

```

< 任务插槽 1 >
1 M_00=0 ; M_00 代入 0。
2 *L2: If M_00=0 Then *L2 ; 等待直到 M_00 的值不为 0。
3 M_00=0 ; M_00 代入 0。
4 Mov P1 ; 前进下列目的作业。
5 Mov P2
:
100 GoTo *L2 ; 从 STEP 编号 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 没有开启时，往标签 L4 分支。
2 M_00=1 ; M_00 代入 1。
3 M_01=2 ; 前进下列目的作业。
4 *L4
:

```

(4) 确认机器人（系统）状态变量下的程序运行状态

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 162 页的“4.5 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，记述机器人本体的动作的程序，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，需要使用机器人本体的获得和开放指令（GetM、RelM）。指令的详细内容，请参照本书 / 第 173 页的“4.12 指令的详细说明”。

(6) 在始终执行程序的处理

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

（由于将运行形态指定为 CYC（1 循环运行）的任务程序仅执行 1 次，因此无需做此设定处理。）

◆◆◇机械 1 被分配到任务插槽 1 ◇◆◆

在初始状态，机械 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，即使在任务插槽 1 中没有获得机械（即使 GetM 指令没有执行）也可以执行移动指令，但是，在任务插槽 1 以外执行移动指令的情况下，请解除任务插槽 1 的机械获取状态后（执行 RelM 指令），在执行移动指令的任务插槽中取得机械。（请执行 GetM 指令。）

4.2.5 多任务程序运行上的注意点

(1) 多任务的开始

从示教单元的〈操作面板〉的起动操作及从专用输入信号的 START 的起动，任务插槽参数的启动条件设定为“START”的全部任务插槽的程序将同时开始执行。

从专用输入信号 S1START ~ S32START 的起动，可以开始执行各个任务插槽。该情况下，在同专用输入输出参数里，预先将信号编号分配使用。关于专用输入输出的分配的详细内容，请参照本书 / 第 605 页的“6.2PLC 连接输出功能”。

(2) 运行状态的显示

示教单元的〈操作面板〉的运行中、待机中，以及专用输入输出信号的 START、STOP，会显示在参数“SLT*”的启动条件设定为“START”的任务插槽的程序运行状态。只要有 1 个程序在运行，示教单元的〈操作面板〉的运行状态会变为运行中、且专用输出信号的 START 会开启。此外，全部程序停止时，示教单元的〈操作面板〉的运行状态会变为待机中、且专用输出信号的 STOP 会开启。

专用输出信号的 S1START ~ S32START、以及 S1STOP ~ S32STOP，输出每一个任务插槽的个别运行状态。需要得知个别的运行状态的情况下，请在相同专用输入输出参数分配信号编号后确认外部信号。关于专用输入输出的分配的详细内容，请参照本书 / 第 605 页的“6.2PLC 连接输出功能”。

启动条件设定为 ALWAYS、ERROR 的程序状态，对示教单元的〈操作面板〉的运行中、待机中的显示没有影响。想要确认始终执行程序运行状态，可以通过计算机支持软件（选配）的监视工具确认。

4.2.6 多任务的使用例

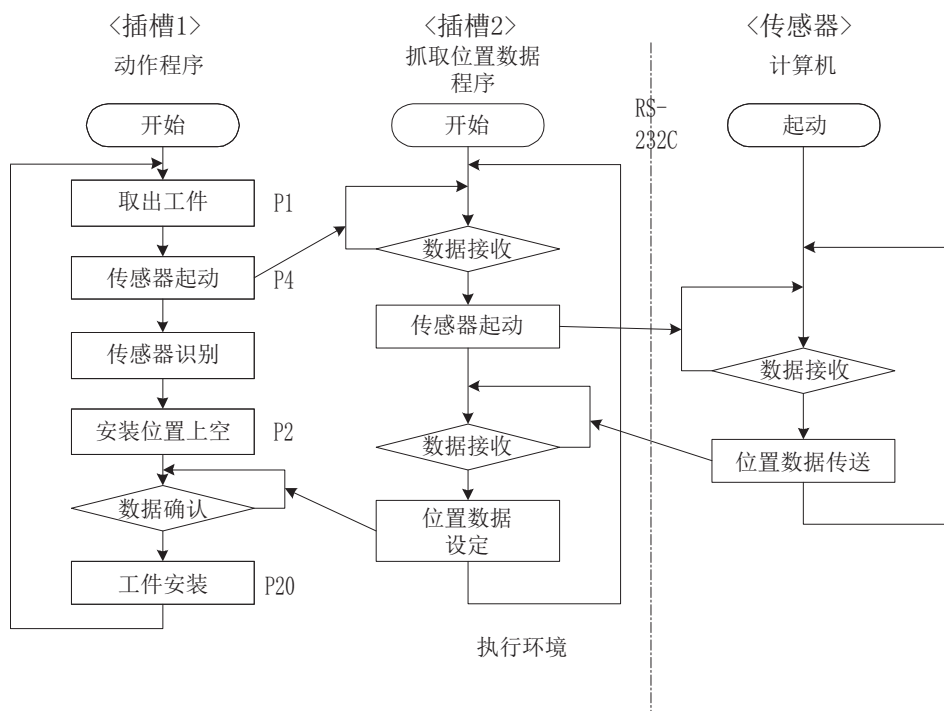
在此对多任务的执行示例进行介绍。

(1) 机器人的作业内容

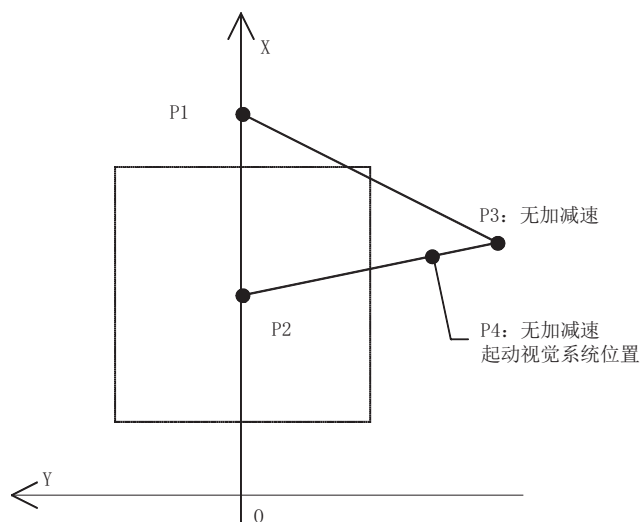
机器人程序分为“动作程序”和“位置数据取得程序”2种。

“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，往传感器输出启动指令时，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。机器人侧介由数据取得程序，取得补偿数据。

< 处理流程 >



- P1: 抓取工件位置 (吸着定时器 D1y 0.05)
- P2: 放置工件位置 (解除定时器 D1y 0.05)
- P3: 视觉前位置 (不停留在通过点 Cnt)
- P4: 视觉快门位置 (不停留这通过点 Cnt)
- P_01: 视觉补偿数据
- P20: 将视觉补偿数据加在 P2 上的位置 (相对运算)



(2) 到多任务执行为止的步骤

■步骤 1: 程序的创建

<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 通过时启动视觉抓取
11 M_02#=0       ' 互锁变量 (M_01=1/M_02=0) 中
12 M_01#=1       ' 在背景处理时, 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' 等待直到互锁变量 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 ' 等待直到互锁变量 M_01=1 为止
2 Open "COM1:" As #1           ' 以太网线路开启
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 ΔX 坐标
7 P_01.Y=M2                     ' 代入 ΔY 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 ΔC 坐标
12 Close                         ' 以太网线路关闭
13 M_01#=0                     ' 互锁变量 M_01=0
14 M_02#=1                     ' 互锁变量 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。

参数	程序名	运行模式	运行形态	优先级
SLT1	1	REP	START	1
SLT2	2	REP	START	1

■步骤 3: 任务插槽参数的反映

为了将参数 (SLT1, SLT2) 设为有效, 请关闭电源后再开启。

■步骤 4: 起动

在操作面板的起动下开始程序 1 和程序 2 的运行。

4.2.7 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 4-3 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 920K 字节的程序。

(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 380K 字节，相当于 1 个程序的最大值。

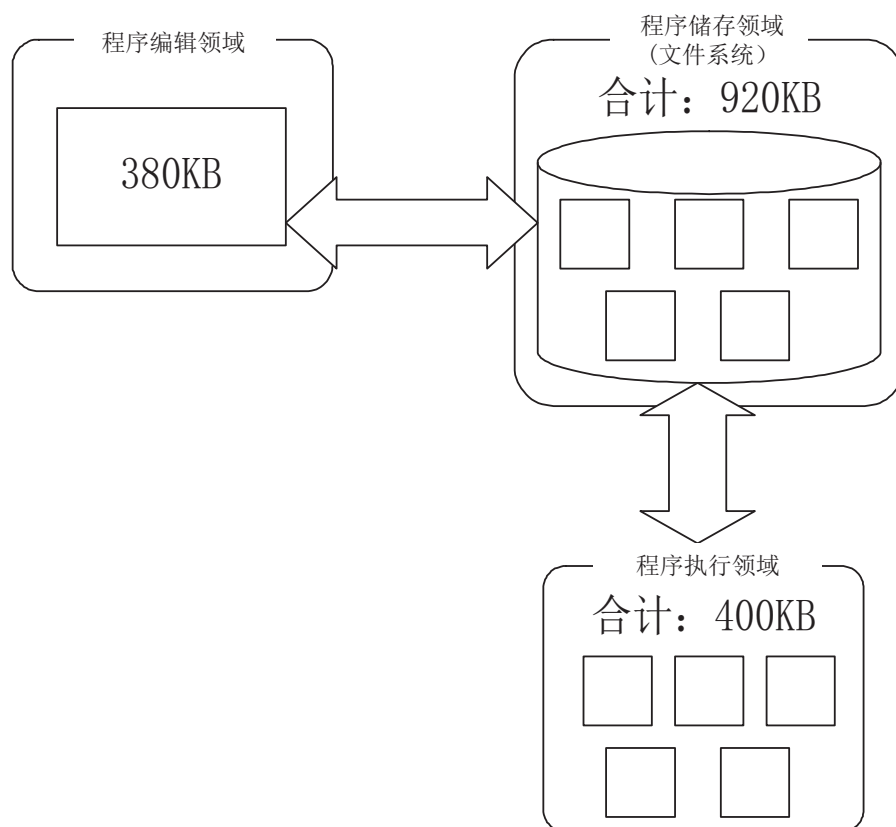
(3) 程序执行领域

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 字节。依据用户基本程序、多任务、XRun、GallP 指令，在同时执行领域加载的程序合计容量，需要为 400K 字节以下。

表 4-3：程序各领域的容量

名称	容量
	标准存储器时
(1) 程序储存领域	920K 字节
(2) 程序编辑领域	380K 字节
(3) 程序执行领域	400K 字节

各程序的容量可以在示教单元及计算机支持软件（RT ToolBox3）的程序管理画面做确认。



4.3 MELFA-BASIC VI的详细规格

在此详细说明 MELFA-BASIC VI 的格式及构成等的相关文法，以及各指令语句的详细功能。下列说明指令语句的构成要素。

(1) 程序名

程序名请用 12 个字符以内创建。可使用的字符如下所示。

区分	可使用的字符
英字符 (半角)	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z (只能使用半角大写。登录小写或全角字符时，会有无法正常执行的情况)
数字 (半角)	0 1 2 3 4 5 6 7 8 9

从外部输入信号选择的情况下，请使用编号创建。

(2) 指令语句

指令语句的构成示例。

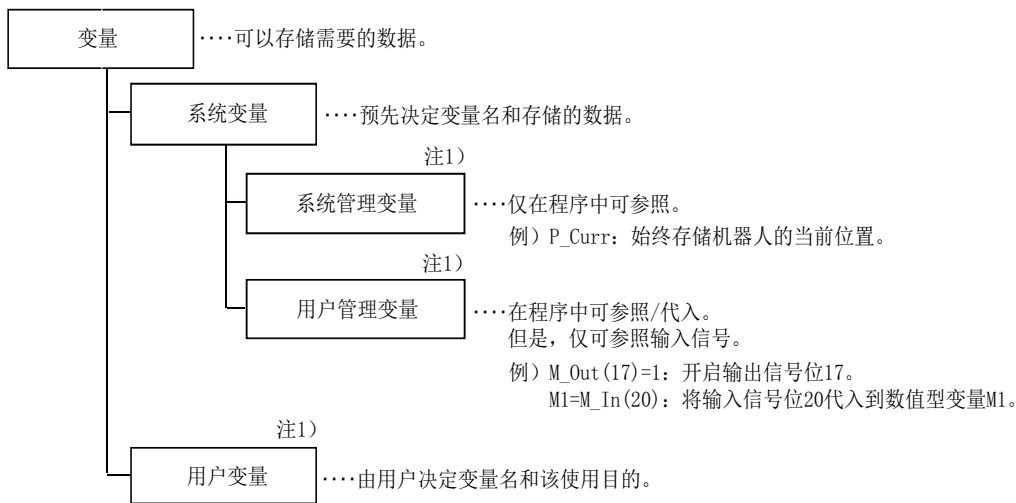
① Mov ② P1 ③ Wth ④ M_Out(17)=1

① ② ③ ④

- ①步号：决定程序的执行顺序的编号。以编号的上升顺序执行。
- ②指令语句：指定机器人的动作及作业的指令。
- ③数据：每个指令语句所需的变量及数值等的的数据。
- ④附随句：根据需要在附加机器人的作业时进行指定。

(3) 变量

程序可以用下列所示种类的变量。



注1) 各变量以以下各个种类进行分类。

- 位置型变量 ……存储机器人的直交/工件坐标型的数据。
直交变量/工件坐标变量名以 ‘P’ (或 ‘p’) / ‘W’ (或 ‘w’) 开始。
例) Mov P1: 移动至变量名P1中存储的位置。
- 关节型变量 ……存储机器人的关节角度。变量名以 “J” 开始。
例) Mov J1: 移动至变量名J1中存储的位置。
- 数值型变量 ……存储数值 (整数、实数等)。变量名以 “M” 开始。
例) M1=1: 在变量名M1中代入值1。
- 字符型变量 ……存储字符串。变量名的最后附带 “\$”。
例) C1\$="ERROR": 变量名C1\$中代入“ERROR”字符串。

4.3.1 语句

语句是构成程序的最小单位，由指令语句和所给予的数据所构成。

例) Mov P1
 指令语句 数据
 指令语句

4.3.2 附随语句

只限定在移动指令，依据附随句，可以串连指令语句。

依据此点，可以执行和移动指令并列的部分指令。

例) Mov P1 Wth M Out(17)=1
 指令语句 附随句 指令语句

详细内容请参照第 324 页的“Wth (With)”或第 324 页的“WthIf (With If)”及各移动指令 (Mov (Move), Mva (Move Arch), Mvs (Move S), Mvr (Move R), Mvr2 (Move R2), Mvr3 (Move R3))。

4.3.3 行

行是由步号和 1 个指令语句所构成的。但是，使用附随句的情况下，指令语句会变为 2 个。一行的长度最多 240 个字符。(不包含行末字符。)

◇◆◇ 1 行只有 1 个指令语句 ◇◆◇

依照一般的 BASIC，多个指令语句以分号进行区分，无法在一行记述。

4.3.4 步号

MELFA-BASIC VI 不采用以往在 MELFA-BASIC 所使用的行编号。取而代之使用步号。看起来与行编号相同，但是，开头会自动附上 1 个编号。步号会变为从 1 开始到 32767 为止的整数。

在 MELFA-BASIC VI 中，无法指定 GoTo、GoSub 等的跳转步号（行编号）。指定时，会变为句法错误。请使用标签取代步号。

4.3.5 标签

标签是作为分支端的记号，可以独自附上的名称。

标签跟在步号的后面，可在 * 后加上英文大写、小写、数字构成，但是开头必须要用英字符，且以 16 个字符以内记述。可以定义 1 个字符的标签，但是由于存在变为保留字的字符 (F、J、L、P、S、T 等为保留字)，因此建议使用 2 个字符以上容易判断的标签名。

标签名会直接登录输入时大写、小写字符。但是，大写、小写并无区别。控制器登录后，读取时会统一成已登录的标签名。

※ 无法在标签中体现的内容如下所示。

- 保留字（指令语句、函数名等）
- 以符号及数字开始的名字
- 在变量名、函数名中已被使用的名字
- “_”（下划线）无法用于标签名的第 2 个字符。

例) 1 GoTo *LBL
 :
 8 *LBL

4.3.6 可以在程序内使用的字符种类

可以在程序内使用的字符如下表所示。但是，在程序名、变量名、标签名中可使用的字符上有限制。可使用的字符以○表示、无法使用的字符以×表示、有限制使用的字符以△表示。

表 4-4：可使用的字符一览

区分	程序可使用的字符	程序名	变量名	标签名
英字符	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	○	○	○
	a b c d e f g h i j k l m n o p q r s t u v w x y z	×	○	○
数字	0 1 2 3 4 5 6 7 8 9	○	△注1)	△注2)
符号	“ ’ & () * + - . , / : ; = < > ? @ ` [\] ^ { } ~	×	×	×
	% & ! # \$	×	可以作为型指定符号使用	×
	_ (下划线)	×	△注3)	○注4)
空白	空白字符	×	×	×

注1) 变量名的开头只能使用英字符。第2个字符以后可以使用数字。

注2) 程序中的标签名的开头只能使用英字符。第2个字符以后可以使用数字。

注3) 第2个字符以后可以使用。在第2个字符使用“_”(下划线)的变量会变为外部变量。

注4) 标签名的第3个字符之后(不包括开头的“*”)可以使用。

关于程序名的详细内容请参照第141页的“(1) 程序名”、变量名的详细内容请参照第148页的“4.3.15 变量”、标签名的详细内容请参照第142页的“4.3.5 标签”。

4.3.7 有特别意义的字符

(1) 大写、小写的区别

MELFA-BASIC VI也接受英文小写。

变量名、标签会直接登录输入时大写、小写字符。但是，大写小写并没有区别，将被统一成已登录的变量名、标签名后处理。因此，在读取程序时，会转换为已登录的名称。

另外，注释和字符串数据时也直接登录。

关于指令语句和系统状态变量名也一样，没有大写和小写的分别，程序读取时，会转换为系统已登录名称。

(2) 下划线 (_)

通过在识别符号（变量的名称）的第2个字符加上_（下划线），该变量会跨程序识别为有效外部变量。详细内容请参照第151页的“4.3.22 外部变量”。

例) P_Curr、M_01、M_ABC

(3) 撇号 (')

将撇号以后的字符视为注释（解说）。此外，如果加在文的开头，将替代Rem文。

例) 100 Mov P1 ' TORU ; TORU会成为注释。
150 ' BUHIN TORIDASHI ; 与150 Rem BUHIN TORIDASHI相同。

(4) 星号 (*)

附在变为分支处的标签名的开头。

例) 200 *KAKUNIN

(5) 逗号 (,)

作为指令的参数（自变量）及下标的区分使用。

例) P1=(200, 150,)

(6) 句号 (.)

仅从小数点、位置变量、关节变量等多个数据中取得成分数据情况下使用。

例) M1=P2.X ; 只将位置变量P2的X坐标成分代入数值变量M1。

(7) 空格

包含在字符串常数及注释文的东西会被视为有字符意义。在行编号及指令语句后，需要每个数据区分开来。“4.12 指令的详细说明”的【格式】中，在必要的场所用“□”表示空格。

4.3.8 数据类型

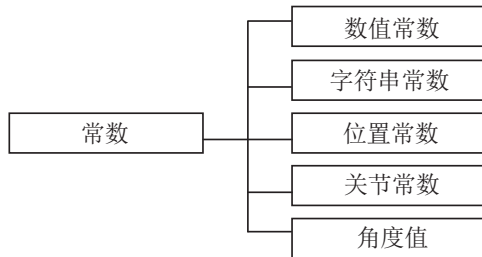
在 MELFA-BASIC VI 中，数据类型可分为数值、位置、关节、字符串这 4 种，各自称为其 [数据类型]。数值型分为实数型和整数型。各数据类型中有变量、常数。



- 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" [字符串常数]

4.3.9 常数

常数有数值常数、字符串常数、位置常数、关节常数、角度值这 5 种。



4.3.10 数值常数

数值常数的句法如下所示。数值常数如下所示。

(1) 10 进制

例) 1、1.7、-10.5、+1.2E+5 (标明指数)

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) 常数的类型

通过在常数字符附加符号可以指定常数类型。

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

4.3.11 字符串常数

字符串常数是指用 (") 双引号符号圈起来的字符。

例) "ABCDEFGHJKLMN" "123"

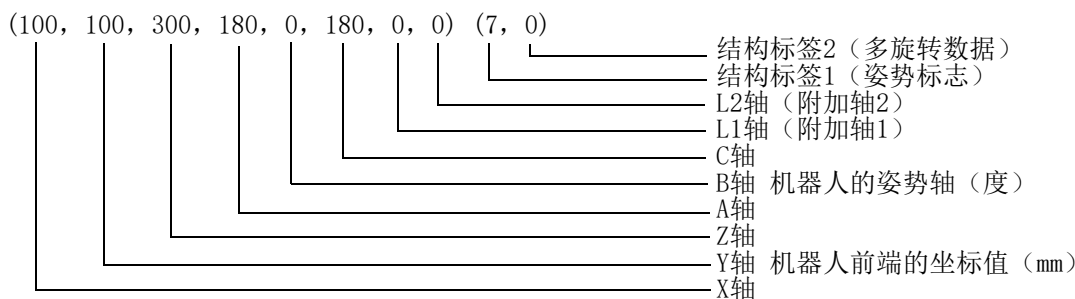
◆◆◆字符串最多为 240 个字符◆◆◆

可输入的字符串长度，包含行编号、双引号在内，在 240 个字符以内。

将双引号本身包含在字符串的情况下，请连续输入两个双引号。输入 AB"CD 字符串的情况下，请输入 "AB" "CD"。

4.3.12 位置常数（直交/工件坐标常数）

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 · 附加轴数据的形式和意义

【形式】X, Y, Z, A, B, C, L1, L2

【意义】X, Y, Z: 坐标数据。直交坐标系中机器人抓手前端的位置 / 工件坐标系中工件位置（单位为 mm）
 A, B, C: 姿势数据。表示姿势的角度。直交坐标系中机器人前端的姿势 / 工件坐标系中工件姿势（单位为 deg）^{注1)}

L1, L2: 附加轴数据。附加轴 1、附加轴 2 各自的坐标值（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中是以弧度为单位进行运算。

(2) 结构标签数据的形式和意义

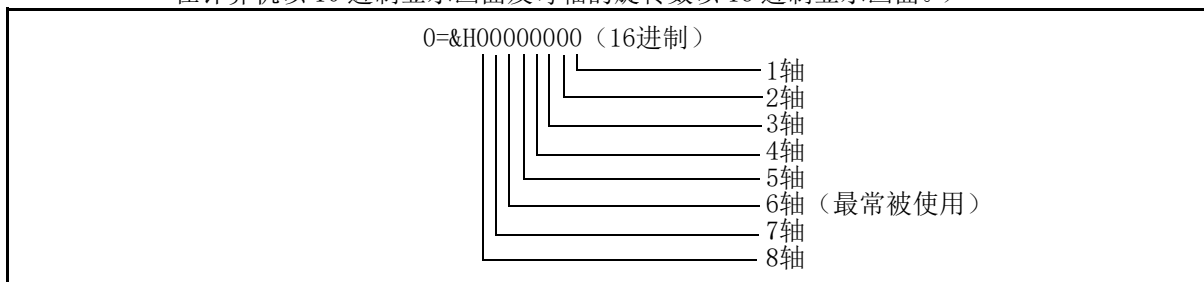
【形式】FL1, FL2

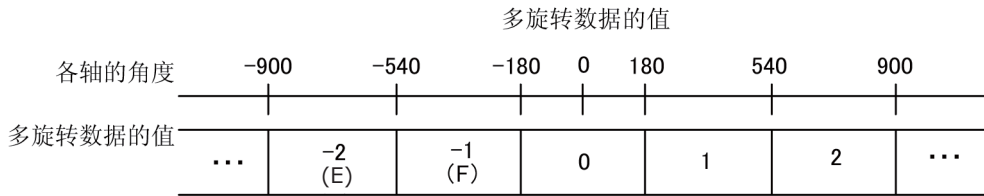
【意义】FL1: 姿势标志。表示在直交坐标系中机器人机械臂的姿势。



FL2: 多旋转数据。具备直交坐标系中表示位置（XYZ）和姿势（ABC）下各关节轴的旋转角度的信息。

初始值 =0（范围从 0 开始至 +4294967295 · · 1 轴 4 位构成的 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。）





在直角坐标系中的腕部前端轴（垂直多关节机器人的 J6 轴），即使旋转 1 周（360 度），也会显示一样的值。因此使用 FL2 来进行区别。

◇◆◇关于轴数的指定◇◆◇

①没有必要记述全部 8 轴的坐标 · 姿势、附加轴数据，但是，省略的情况下，以后的轴数据会被作为未定义进行处理。

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省略全部轴的情况下，应像 (,) 这样至少加入一个 “,” (逗号)。

◇◆◇关于使用位置成分数据的变量◇◆◇

坐标 · 姿势 · 附加轴数据和结构标签数据统称为位置成分数据。

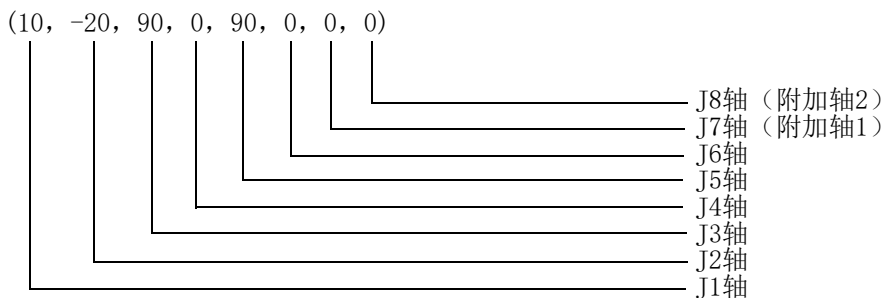
构成位置常数的位置成分数据里，无法包含变量。

◇◆◇结构标签数据的省略◇◆◇

将构造数据省略时，初始数据 ((7, 0) 因机型而异) 会被适用。

4.3.13 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，而是会变为 mm。

◇◆◇关于使用关节成分数据的变量◇◆◇

各轴数据称为关节成分数据。

构成关节常数的关节成分数据里无法包含变量。

4.3.14 角度值

角度值不是指 “弧度”，而是在表现 “度” 时使用。

记述为 100Deg 时，会变为角度值，可以在三角函数等的自变量使用。

例) Sin(90Deg) 表示为 90 度正弦。

4.3.15 变量

变量名的创建，字符最多为 32。可以使用 1 个英文字的变量名，但是，由于存在变为保留字的字符（F、J、L、P、S、T、W 等），因此建议使用 2 个字符以上容易判断的名称。

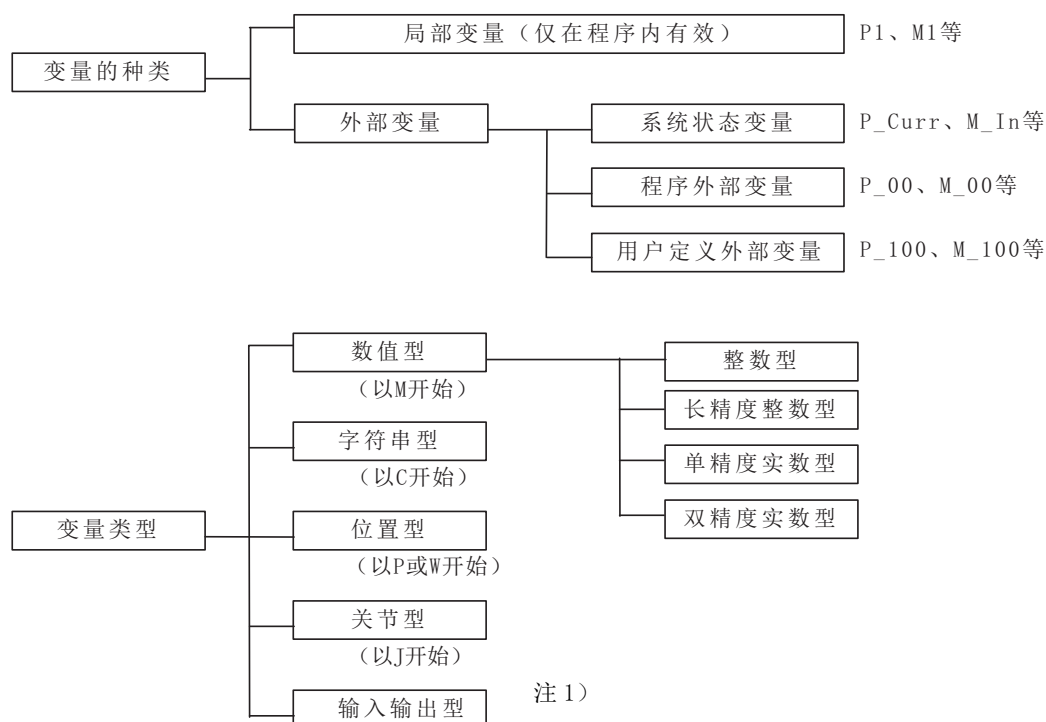
变量的类型分为数值型、字符串型、位置型、关节型、输入输出型。各自称为其“变量的类型”。变量的类型决定于识别符号（变量的名称）的开头字符。

此外，数值型分为整数型、长精度整数型、单精度实数型、双精度实数型。

此外，作为数据的有效范围，分为以下 2 个种类。

① 只在 1 个程序内有效的局部变量

② 跨程序有效的机器人（系统）状态变量、程序外部变量、用户定义外部变量（在变量名的第 2 字符附上 _ 的外部变量。第 151 页的“4.3.22 外部变量”请参照。）



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 Def I0 指令用程序声明的符号。

◇◆◇变量不会被初始化◇◆◇

在被生成、程序加载或复位时，变量不会被清零。

4.3.16 数值变量

数值变量为使用 P、J、C、W 的字符以外的字符开始的变量。在 MELFA-BASIC VI 里，大部分是以 M 为开头进行说明。M 为 mathematics（数学）的开头字符。

```
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。省略的情况下会变为单精度实数型。

表 4-5: 数值变量的类型和数据范围

数值型后置符号	类型	范围	备注
%	整数型	-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。请在创建新建程序时进行最初记述时，将符号写上。
- 3) 将长精度整数型数据代入整数型的数值变量时，值若超过时，执行时会发生错误（L3110）。将双精度实数型数据代入单精度实数型的数值变量时，值若超过时，会代入无限大的值（附带符号“Inf”）。

4.3.17 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，也可以加上以“C”（或“c”）以外的字符开始的变量名。

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

4.3.18 位置变量（直交/工件坐标变量）

直交变量 / 工件坐标变量为以‘P’（或‘p’）/‘W’（或‘w’）开始的变量。以 Def Pos 指令 / Def Work 指令定义的情况下，也可以加上以‘P’（或‘p’）/‘W’（或‘w’）以外的字符开始的变量名。可参照位置变量的成分数据。可以在变量名的后面加上“.”和成分名“X”等。

```
P1.X、 P1.Y、 P1.Z、 P1.A、 P1.B、 P1.C、 P1.L1、 P1.L2
```

角度成分的 ABC 的单位为（rad）。请使用 Deg 函数以转换成度。

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

4.3.19 关节变量

关节变量为以“J”（或“j”）开始的变量。以 Def Jnt 指令定义的情况下，也可以加上以“J”（或“j”）以外的字符开始的变量名。

可参照关节变量的成分数据。在变量名的后面加上“.”和成分名“J1”等。

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

成分数据的单位为（rad）。请使用 Deg 函数以转换成度。

```
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
```

4.3.20 输入输出变量

输入输出变量有以下种类。此为在机器人（系统）状态变量中预先准备好的。

输入输出变量名	说明
M_In	输入信号的位参照用
M_Inb	输入信号的字节（8位）参照用
M_Inw	输入信号的单字（16位）参照用
M_Din	输入信号的双字（32位：支持 CC_Link）参照用
M_Out	输出信号的位参照 / 代入用
M_Outb	输出信号的字节（8位）参照 / 代入用
M_Outw	输出信号的单字（16位）参照 / 代入用
M_Dout	输出信号的双字（32位：支持 CC_Link）参照 / 代入用

请参照机器人（系统）状态变量的第 368 页的“M_In/M_Inb/M_In8/M_Inw/M_In16/M_In32”、第 379 页的“M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32”及第 353 页的“M_DIn/M_DOut”。

4.3.21 排列变量

数值变量、字符串变量、位置变量、关节变量可以排列使用。在变量的下标部分指定排列要素。排列变量用 Dim 指令记述。最大可排列 3 次元。

例) 排列变量的定义示例

```
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)
```

排列的下标从 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），下标会变为从 0 开始。

由剩余内存的容量决定能否确保区域。

4.3.22 外部变量

所谓外部变量，是在识别符号（变量的名称）的第2个字符加上“_”（下划线）的变量，其值跨多个程序仍为有效（用户定义外部变量需要登录在基本程序里）。因此，在多任务的程序间交接传输数据时是有效的。

外部变量和第145页的“4.3.8数据类型”中所示相同的有数值、位置、关节、字符4种类型。外部变量有下列的3个种类。

表 4-6：外部变量的种类

外部变量	说明	例
程序外部变量	程序外部变量预先决定名称。	P_01、M_01、P_100(1)等
用户定义外部变量	用户可以任意决定名称。 在用户基本程序利用Def Pos, Def Jnt, Def Char, Def Inte/Long/Float/Double 进行声明。	P_GENTEN、M_MACHI
机器人（系统）状态变量	机器人（系统）状态变量由系统管理，事先决定名称的内容（用途）。	M_In、M_Out、P_Curr、M_PI等

4.3.23 程序外部变量

控制器最初准备的程序外部变量如表4-7所示。如表所示，变量的名称已决定，但是其用途是由用户决定。

表 4-7：程序外部变量

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

4.3.24 用户定义外部变量

在只有上述的程序外部变量不足的情况下，想要定义附有独立名称的变量时，可以利用用户基本程序，由用户定义程序外部变量。

到使用用户定义外部变量为止的步骤
1) 首先创建用户基本程序。在变量的第 2 个字符加上 “_”。
2) 在参数 “PRGUSR” 设定程序名后，重新接通电源。
3) 创建已使用用户定义外部变量的常规程序。

- (1) 用户基本程序^{注)}中的 Def 语句里，通过定义识别符号的第 2 个字符里带有 “_” (下划线) 的变量，此变量会变为用户定义外部变量。
- (2) 无需执行用户基本程序。
- (3) 用户基本程序中，请只记述变量的声明行。
- (4) 用户基本程序里定义排列变量，作为外部变量使用的情况下，即使是使用中的程序侧，也需要再次以 Dim 指令进行排列声明。局部变量 (只在程序内有效的变量) 则无需再次声明。

例) 用户定义外部变量的使用示例

主程序 (程序名 1) 侧

1 Dim P_200(10)	' 外部变量的再次声明
2 Dim M_200(10)	' 外部变量的再次声明
3 Mov P_100(1)	
4 If M_200(1)=1 Then Hlt	
5 M1=1	' 局部变量

用户基本程序 (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次声明。
3 Def Inte M_100	
4 Dim M_200(10)	' 使用的程序侧也需要再次声明。

参数名	值
PRGUSR	100

4.3.25 用户基本程序的创建

注)

◇◆◇何谓用户基本程序？◇◆◇

用户基本程序是指使用用户定义外部变量的情况下，为了定义该变量所使用，但无需执行本程序。创建已记述声明文的程序并只登录在参数 “PRGUSR” 中。将参数变更后，请再一次接通电源。

◇◆◇用计算机支持软件登录新用户基本程序的方法◇◆◇

最初只从计算机支持软件往控制器写入指令，接着只写入位置数据。

用户基本程序和常规程序相同，可以用示教单元、计算机支持软件的任意一个创建，但是，使用计算机支持软件创建的情况下，请按以下步骤执行。

- 1) 将用于用户基本所创建的程序，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) 创建的程序，将传送目标视为机器人，执行 “复制” 操作。此时，去除 “位置变量” 的确认，只在 “指令语句” 确认的状态下执行。
- 4) 复制完成后，再次执行 3) 的操作。这次将 “指令语句” 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，也请先将用户基本程序写入控制器。

4.3.26 有效范围

变量的有效范围是指小范围内的“过程内→过程外（程序内→程序外）→用户定义外部变量 / 程序外部变量 / 系统状态变量”。（参照图 4-3）

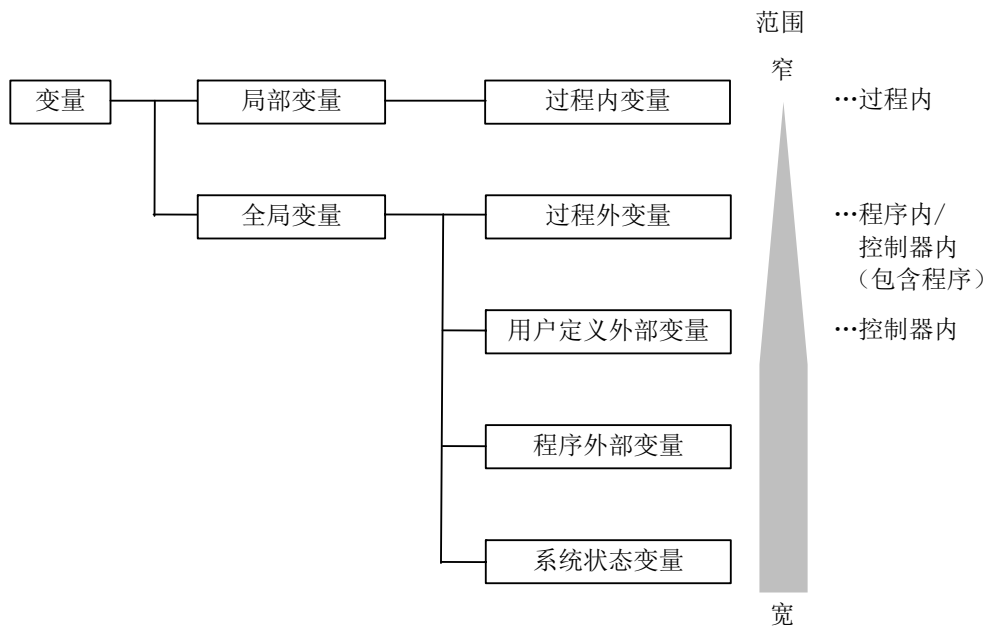


图 4-3: 变量有效范围

4.3.27 Function过程

Function 过程是指，可以作为一步处理的独立使用函数（处理）。在 Function 过程中，可在外部独立处理使用的变量，也能定义函数的返回值，因此能够像零件一样使用各处理，也能使程序的推测性更好。

如果要定义 Function 过程，程序的入口点^{注 1)}为 Function Main 过程。

※ 如果要定义过程，需要定义 Function Main 过程。

※ 如果要定义过程，在入口点^{注 1)}之前执行预处理程序^{注 2)}。

注 1) 入口点是指，指定程序名启动程序时，执行程序开头、或程序内的 Function Main 过程（无假设自变量）。程序的执行点被称为程序的“入口点”。

注 2) 预处理程序是指，执行 Function 过程外记述的指令语句的处理。

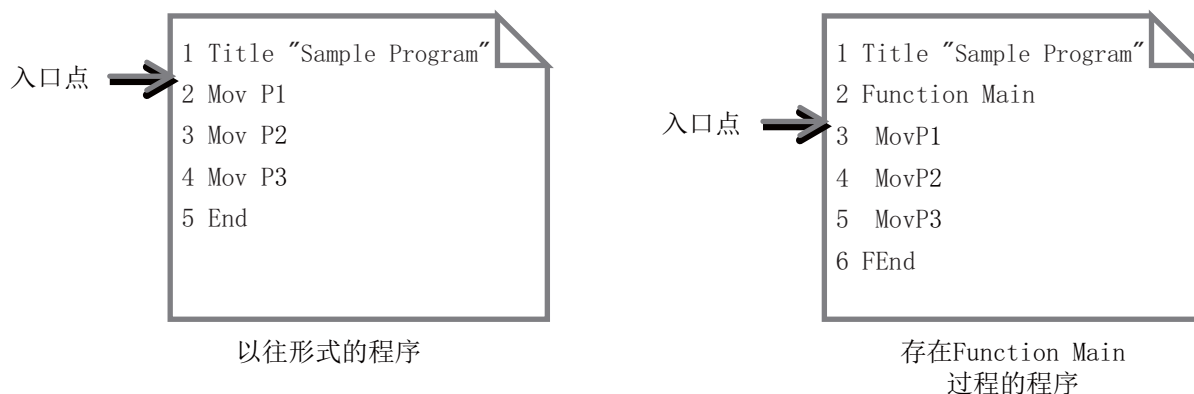


图 4-4: 入口点

由 Function 过程记述的情况下，处理步骤如图 4-5 所示。

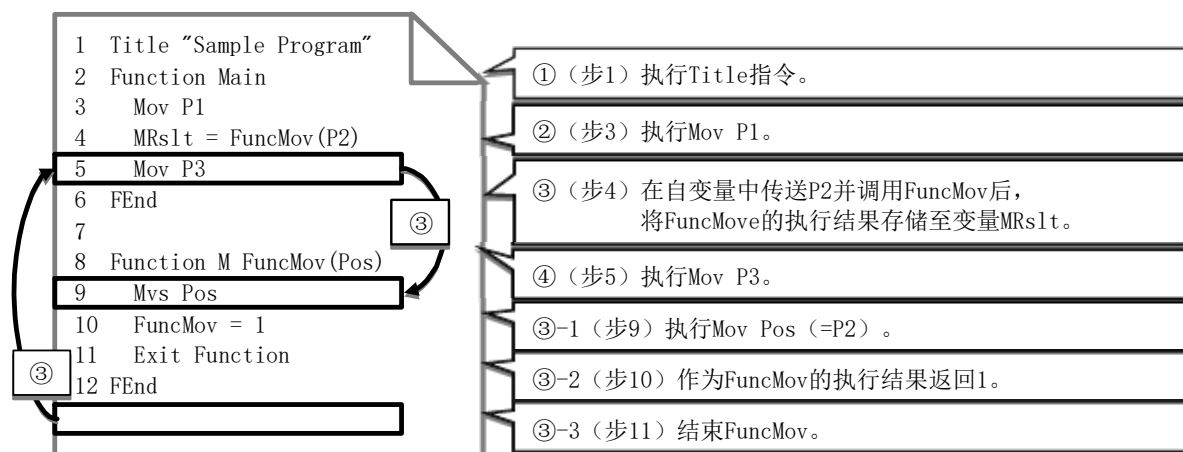


图 4-5: 由 Function 过程记述时的处理步骤

关于 Function 过程的格式，请参照“指令的详细说明 Function 过程的项目”。

4.3.28 #Include 陈述句

能够从程序读取指定程序。将经常使用的程序（function）及通用程序（function）作为程序库管理，根据需要使用该程序等时将会更加方便。

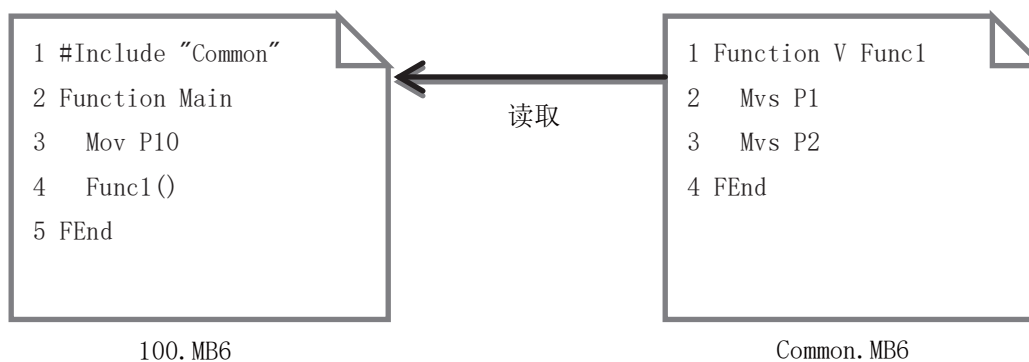


图 4-6: 程序的读取（#Include 陈述句）

※ 能同时从多个程序中指定并读取相同的程序。
详细情况请参照“指令的详细说明 Include 陈述句的项目”。

4.4 机器人的坐标系说明

4.4.1 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

表明机器人的当前位置^{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（基本转换数据）设定到参数：MEXBS，或通过执行Base指令进行设定。

初始设定时，基本转换数据为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）

以机器人的机械接口面为基准确定的坐标系。

④工具坐标系：原点为※4

由机器人的机械接口面上安装的工具（抓手）确定的坐标系。它与机械接口坐标系之间的关系取决于工具转换数据（参数：MEXTL的设定、或工具指令的执行）。

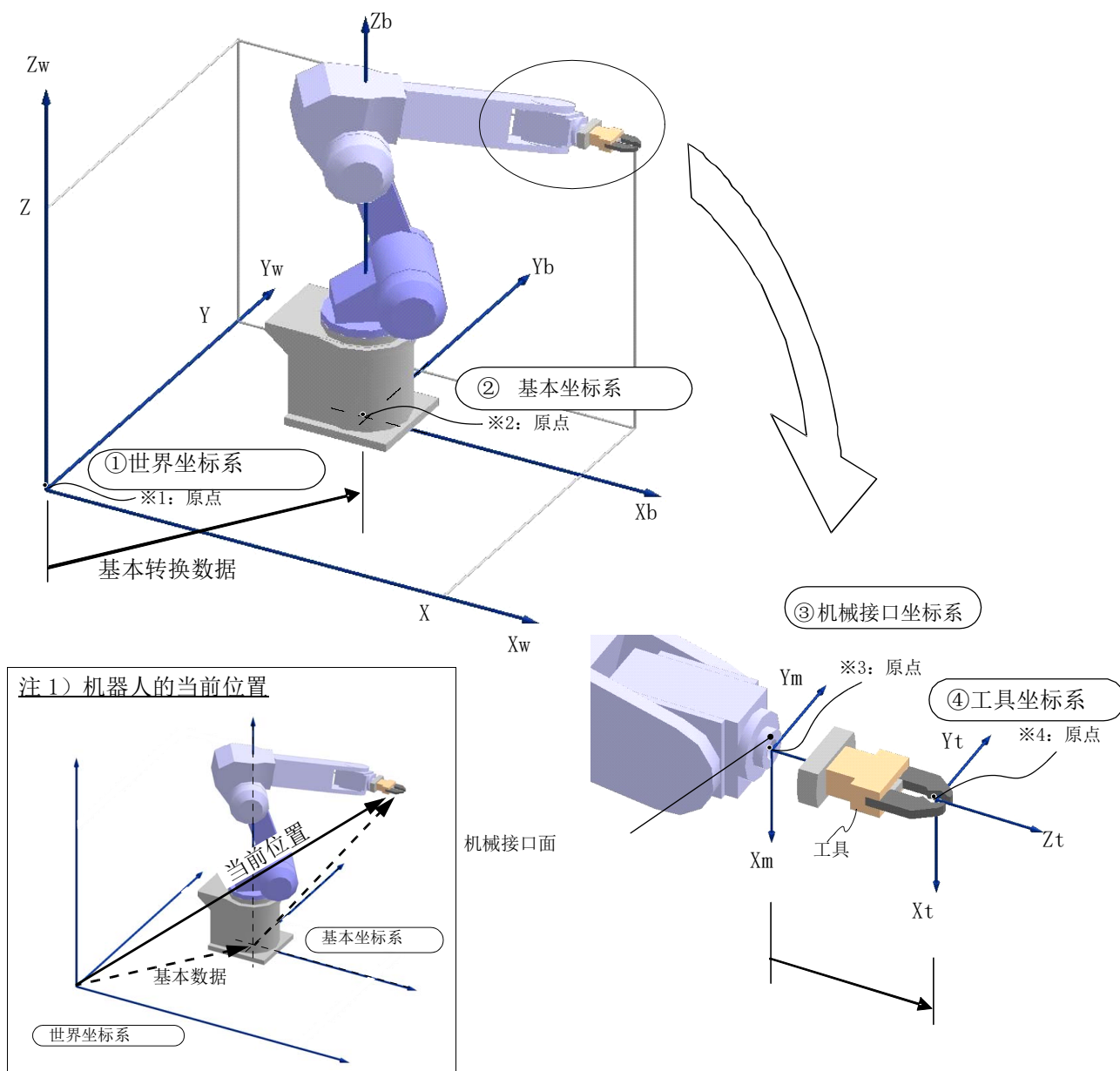


图 4-7：机器人的坐标系

4.4.2 关于基本转换

可根据需要将世界坐标系移动到作业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作业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转换，通过对参数：MEXBS 的基本转换数据（坐标值）的设定、或执行 Base 指令进行。

要设定的基本转换数据为，从新设定的世界坐标系观察到的基本坐标系的原点位置数据。因此，使用机器人的位置数据进行设定时（Fram 函数等），要对位置数据进行逆转换并进行设定。（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转换，因此，用户无需实施逆转换处理。（例：Base1 ~ Base8）

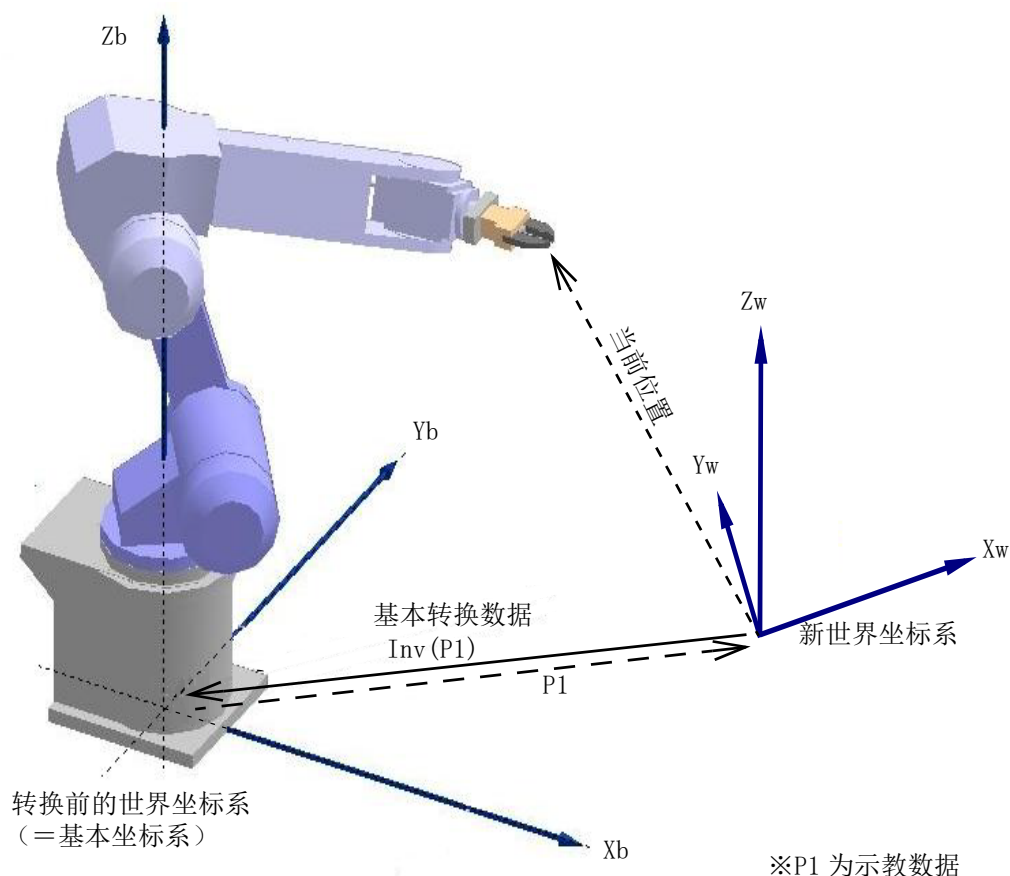


图 4-8：基本转换

⚠ 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。

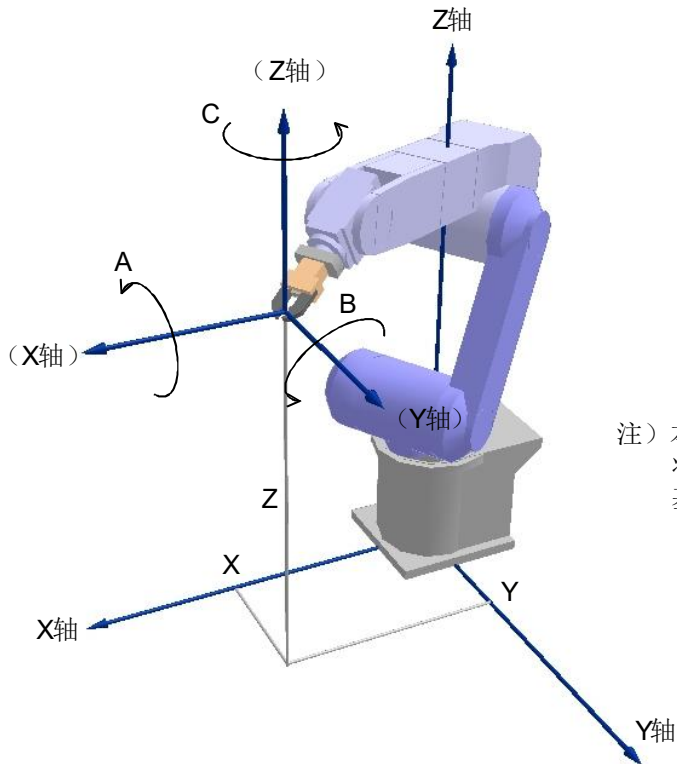
如果坐标系不一致，则机器人会移动至预想不到的位置，从而造成财产损失或人身事故。

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。

4.4.3 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接口中心）的位置（X, Y, Z）与姿势（A, B, C）的6个数据以及结构标签的合并数据。
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）
A, B, C: 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转换数据的状态, 即假设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。相对各坐标轴的 + 方向, 右螺栓的旋转方向为 + 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）。

4.4.4 关于工具坐标系（机械接口坐标系）

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

为了更容易理解工具坐标系，这里我们将以垂直 6 轴机器人为例进行说明，关于其他机型（水平关节型机器人等），请参照第 500 页的“5.6 关于标准 TOOL 坐标”。

(1) 机械接口坐标系

图 4-9 正如所示，以法兰中心为原点的坐标系被称为机械接口坐标系。机械接口坐标系的 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。法兰中心和定位用针孔的连线即为 X_m 轴。 X_m 轴的 + 方向为从中心到定位用针孔相反方向。

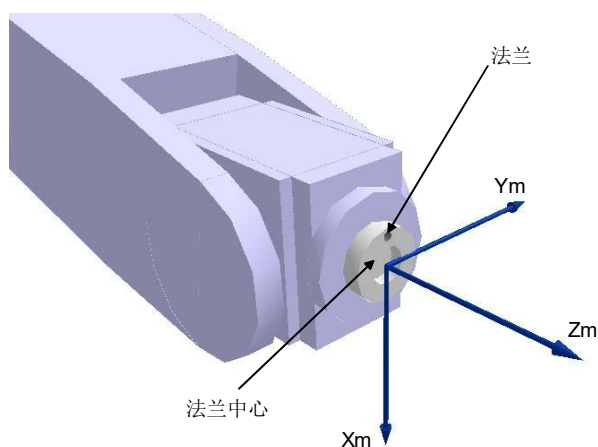


图 4-9：机械接口坐标系

法兰旋转，机械接口坐标系亦随之旋转。（图 4-10）

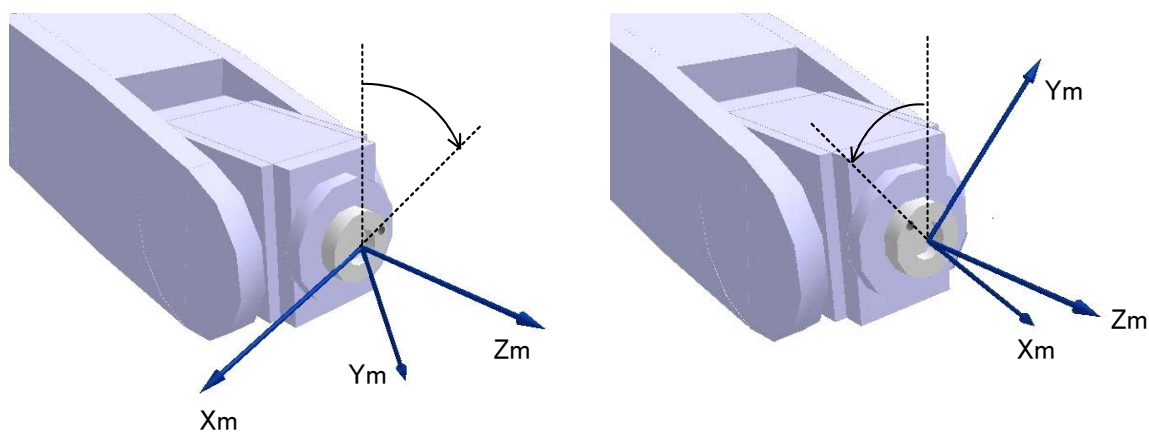


图 4-10：法兰旋转和机械接口坐标系

(2) 工具坐标系

工具坐标系是定义在抓手前端（抓手控制点）的坐标系，它将机械接口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。

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

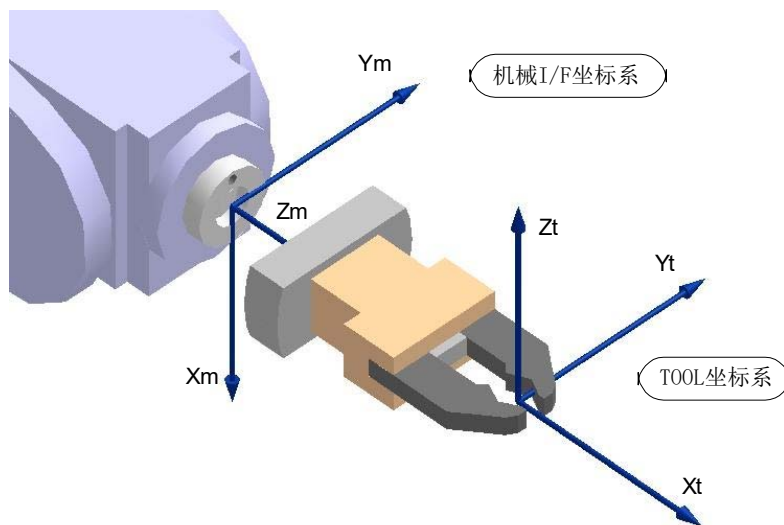


图 4-11：机械接口坐标系与工具坐标系

工具转换数据具有与位置数据相同的成分。

X, Y, Z : 移动量。从机械接口坐标系原点到工具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)

$A \rightarrow$ 绕 X 轴旋转的旋转角

$B \rightarrow$ 绕 Y 轴旋转的旋转角

$C \rightarrow$ 绕 Z 轴旋转的旋转角

(3) 工具坐标系的使用效果

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。更易于调整相对于对象工件的抓手姿势及变更夹持工件的姿势。

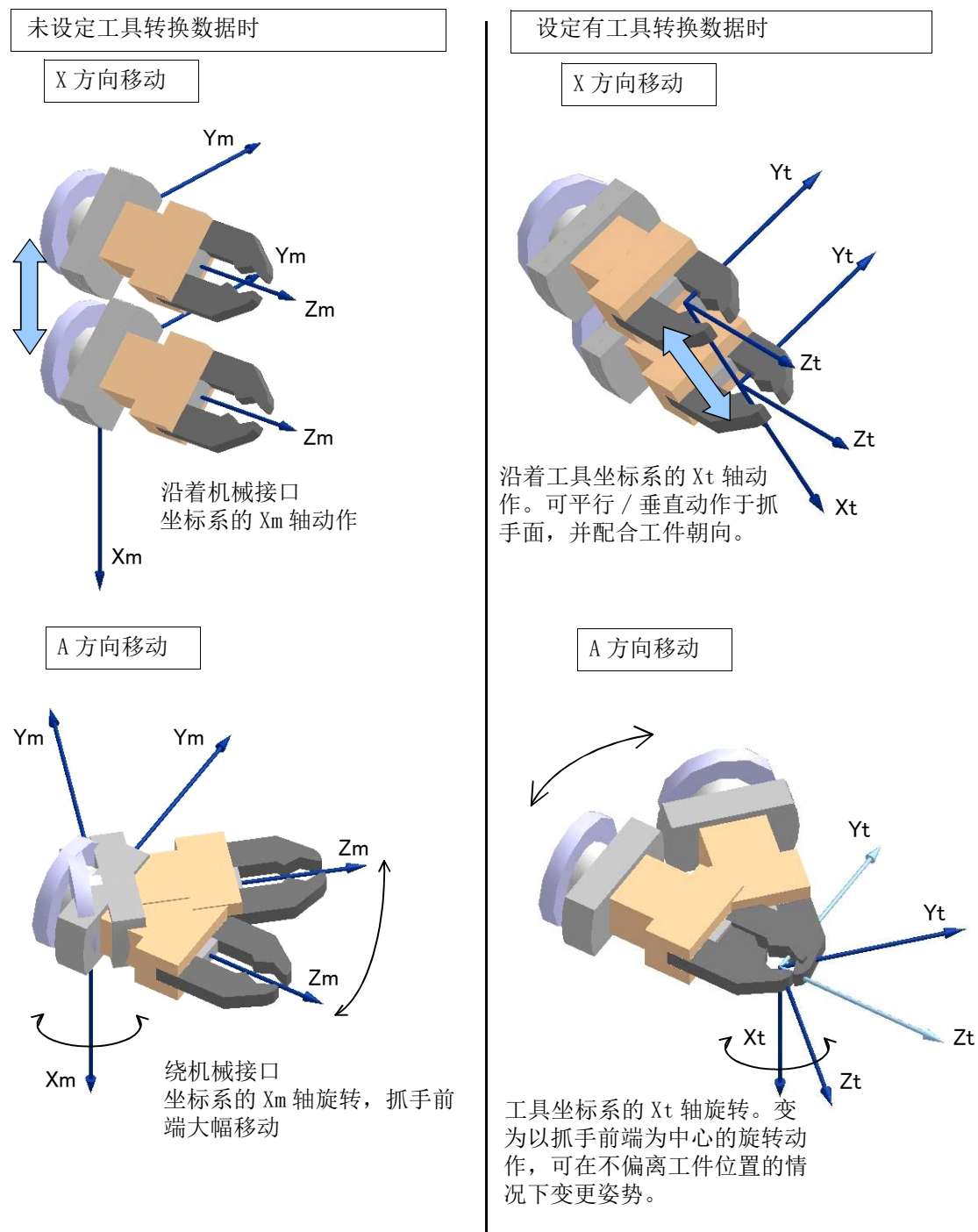


图 4-12：基于有无工具转换数据的工具 JOG 动作

2) 自动运行时

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

图 4-13 正如图所示，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（工具坐标系）上向 +50mm 的位置移动。
根据工件的朝向或动作状态设定工具坐标系的 Z 方向，可提高操作性。

图 4-13 的示例中，为了使抓手侧横以进行工件的插入 / 拔出操作，工具坐标系的 Z 轴方向与工件的朝向要一致。

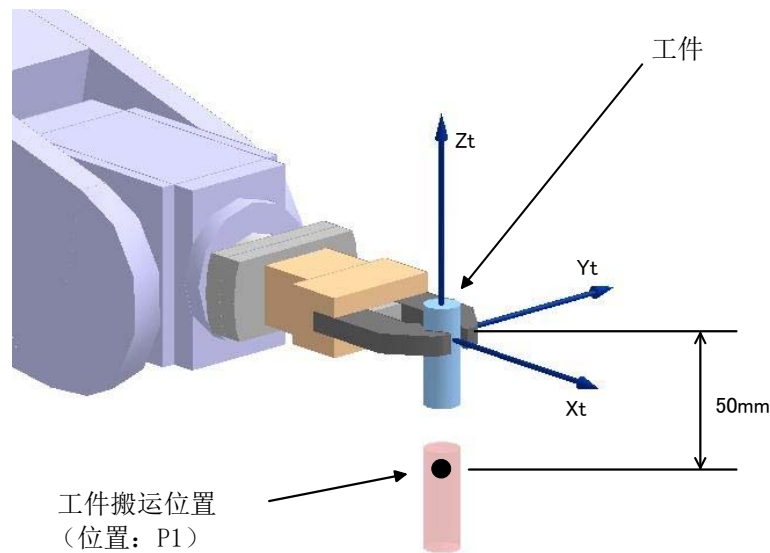


图 4-13: 靠近 / 脱离动作

在进行工件的位相对准等工件的姿势变更时，如设定有工具转换数据，则操作起来会更加方便。

图 4-14 所示，绕工件的中心轴旋转，对准位相时，记述为

$Mov P1*(0, 0, 0, 0, 0, 45)$ 。

“ $(0, 0, 0, 0, 0, 45)$ ”中，“*”表示工具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（绕工具坐标系的 Zt 轴旋转）旋转 45° 。

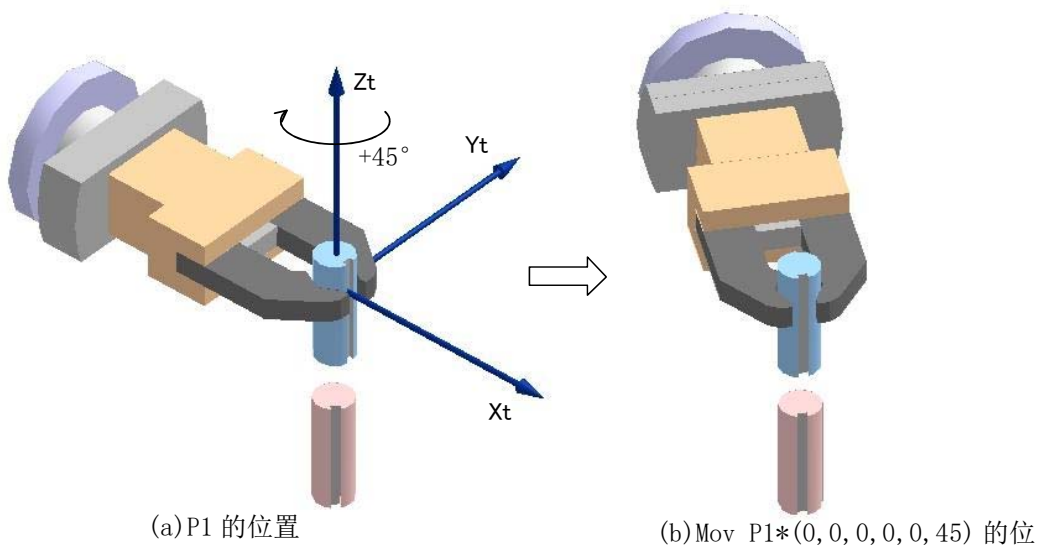


图 4-14: 工具坐标系上的旋转动作

4.5 机器人（系统）状态变量

预先准备的机器人的状态变量如表 4-8 所示。如表所示，预先决定变量的名称及用途。使用这些变量时，可以调查机器人的状态，也可以使状态变化。

表 4-8：机器人（系统）状态变量

No	变量名称	排列要素数 ^{注1)}	内容	属性 ^{注2)}	数据的类型、单位	页
1	P_Curr	机械编号 (1~3)	当前位置(直交)	R	位置型	411
2	J_Curr	机械编号 (1~3)	当前位置(关节)	R	关节型	338
3	J_ECurr	机械编号 (1~3)	当前编码器脉冲位置	R	关节型	342
4	J_Fbc	机械编号 (1~3)	从伺服来的反馈制作的关节位置	R	关节型	343
5	J_AmpFbc	机械编号 (1~3)	电流的反馈值	R	关节型	343
6	P_Fbc	机械编号 (1~3)	以从伺服来的反馈值为依据形成的直交位置	R	位置型	414
7	M_Fbd	机械编号 (1~3)	指令位置和反馈位置的距离	R	整数型, mm	359
8	M_CmpDst	机械编号 (1~3)	弹性伺服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	349
9	M_CmpLmt	机械编号 (1~3)	弹性伺服模式的指令值超过限制而变为错误时, 使用插入处理从错误状态复归时使用。	R	整数型	350
10	P_Tool	机械编号 (1~3)	当前指定的工具转换数据	R	位置型	420
11	P_Base	机械编号 (1~3)	当前指定的基本转换数据	R	位置型	407
12	P_NTool	机械编号 (1~3)	系统初始值(工具转换数据)	R	位置型	420
13	P_NBase	机械编号 (1~3)	系统初始值(基本转换数据)	R	位置型	407
14	M_Tool	机械编号 (1~3)	工具编号(1~16)	RW	整数型	392
15	J_ColMxl	机械编号 (1~3)	推断转矩和实际转矩的差异	R	关节型, %	339
16	M_ColSts	机械编号 (1~3)	碰撞检测状态(1: 碰撞中, 0: 其它)	R	整数型	351
17	P_ColDir	机械编号 (1~3)	碰撞时的动作方向	R	位置型	409
18	P_CordR	机械编号 (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	410
19	P_CurrR	机械编号 (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当前位置。	R	位置型	412
20	M_Cavsts	机械编号 (1~3)	通过干涉回避功能, 返回检测到干涉的机器人的CPU编号。	RW	整数型	348
21	P_CavDir	机械编号 (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	408
22	M_OPOvrd	无	操作面板的速度比例(0~100%)	R	整数型、%	370
23	M_Ovrd	插槽编号 (1~32)	现在指定的程序内速度比例(0~100%)	R	整数型、%	370
24	M_Jovrd	插槽编号 (1~32)	现在指定的关节速度比例(0~100%)	R	整数型、%	370
25	M_NOvrd	插槽编号 (1~32)	系统初始值(M_OVRD的初始值)(%)	R	单精度实数型、%	370
26	M_NJovrd	插槽编号 (1~32)	系统初始值(M_JOVRD的初始值)(%)	R	单精度实数型、%	370
27	M_Wupov	机械编号 (1~3)	暖机运行速度比例(50~100%)	R	单精度实数型、%	399
28	M_Wuprt	机械编号 (1~3)	到解除暖机运行状态为止的时间(秒)		单精度实数型、秒	400
29	M_Wupst	机械编号 (1~3)	再次到暖机运行状态为止的时间(秒)		单精度实数型、秒	401
30	M_Ratio	插槽编号 (1~32)	到当前动作中的目的位置为止的到达率(%)	R	整数型、%	383
31	M_RDst	插槽编号 (1~32)	当前动作中的剩余距离(只考虑X, Y, Z的3次元 mm)	R	单精度实数型, mm	384
32	M_Spd	插槽编号 (1~32)	当前指定的速度(只在直线/圆弧插补时有效)	R	单精度实数型、mm/s	387
33	M_NSpd	插槽编号 (1~32)	系统初始值(M_SPD的初始值)(mm/s)	R	单精度实数型、mm/s	387
34	M_RSpd	插槽编号 (1~32)	当前的指令速度(mm/s)	R	单精度实数型、mm/s	387
35	M_Acl	插槽编号 (1~32)	当前指定的加速率(%)	R	单精度实数型、%	344
36	M_Dacl	插槽编号 (1~32)	当前指定的减速率(%)	R	单精度实数型、%	344
37	M_NAcl	插槽编号 (1~32)	系统初始值(M_ACL的初始值)(%)	R	单精度实数型、%	344
38	M_NDacl	插槽编号 (1~32)	系统初始值(M_DACL的初始值)(%)	R	单精度实数型、%	344
39	M_AclSts	插槽编号 (1~32)	当前的加减速状态0=停止中 1=加速中 2=恒速中 3=减速中	R	整数型	344
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、%	385
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、%	372
42	M_Run	插槽编号 (1~32)	运行中状态(1: 运行中 0: 非运行中)	R	整数型	384
43	M_Wai	插槽编号 (1~32)	中断中状态(1: 中断中 0: 非中断中)	R	整数型	398

No	变量名称	排列要素数 ^{注1)}	内容	属性 ^{注2)}	数据的类型、单位	页
44	M_Psa	插槽编号 (1 ~ 32)	通过指定的插槽返回可否选择程序。(1: 可选择, 0: 不可选择 中断中时)	R	整数型	382
45	M_Cys	插槽编号 (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	353
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	352
47	C_Prg	插槽编号 (1 ~ 32)	执行程序名	R	字符串型	336
48	M_Line	插槽编号 (1 ~ 32)	当前执行中的行编号	R	整数型	371
49	M_SkipCq	插槽编号 (1 ~ 32)	包含最后执行的SKIP指令在内的行的执行结果, SKIP时为1, 未SKIP时为0。	R	整数型	386
50	M_BrkCq	无	BREAK指令结果 (1: BREAK, 0: 无)	R	整数型	347
51	M_Err	无	错误中 (1: 错误发生中 0: 非错误发生中)	R	整数型	357
52	M_ErrLvl	无	读取错误等级 • S/W版本R1d之前 (SQ系列)/S1d之前 (SD系列) 无错误/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) 无错误/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	357
53	M_Errno	无	读取错误编号。	R	整数型	357
54	M_Svo	机械编号 (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	390
55	M_Uar	机械编号 (1 ~ 3)	位数据 (1: 用户定义区域内、0: 用户定义区域外) (位0: 区域1~位15; 区域16)	R	整数型	393
56	M_Uar32	机械编号 (1 ~ 3)	位数据 (1: 用户定义区域内、0: 用户定义区域外) (位0: 区域1~位31; 区域32)	R	整数型	398
57	M_In	输入编号 (0 ~ 32767)	获取外部输入信号 (以位为单位)。 通用位软元件: 位信号输入 0=off 1=on	R	整数型	368
58	M_Inb/ M_In8	输入编号 (0 ~ 32767)	获取外部输入信号 (以8位为单位)。 通用 " : 字节信号输入	R	整数型	368
59	M_Inw/ M_In16	输入编号 (0 ~ 32767)	获取外部输入信号 (以16位为单位)。 通用 " : 字信号输入	R	整数型	368
60	M_In32	输入编号 (0 ~ 32767)	通过数值获取外部输入信号 (以32位为单位)。 通用 " : 双字信号输入	R	整数型	368
61	M_Out	输出编号 (0 ~ 32767)	输出外部输出信号 (以位为单位)。 通用位软元件: 位信号输出 0=off 1=on	RW	整数型	379
62	M_Outb/ M_Out8	输出编号 (0 ~ 32767)	输出外部输出信号 (以8位为单位)。 通用 " : 字节信号输出	RW	整数型	379
63	M_Outw/ M_Out16	输出编号 (0 ~ 32767)	输出外部输出信号 (以16位为单位)。 通用 " : 字信号输出	RW	整数型	379
64	M_Out32	输出编号 (0 ~ 32767)	向外部输出信号 (以32位为单位)中输出数值。 通用 " : 双字信号输出	RW	整数型	379
65	M_DIn	输入编号 (6000 ~)	以字为单位从CC-Link和EtherCAT的输入寄存器读取数据。 CR800-R/Q系列无法使用。 (CC-Link的远程寄存器: 输入寄存器的读取)	R	整数型	353
66	M_DOut	输出编号 (6000 ~)	以字为单位对CC-Link和EtherCAT的输出寄存器进行数据的写入和读取。 CR800-R/Q系列无法使用。 (CC-Link的远程寄存器: 输出寄存器的写入、读取)	RW	整数型	353
67	M_DIn32	输入编号 (6000 ~ 6254)	以双字为单位从CC-Link和EtherCAT的输入寄存器读取数据。	R	整数型	354
68	M_DOut32	输出编号 (6000 ~ 6254)	以双字为单位对CC-Link和EtherCAT的输出寄存器进行数据的写入和读取。	RW	整数型	355
69	M_HndCq	输入编号 (1 ~ 8)	返回抓手确认输入信号。	R	整数型	368
70	P_Safe	机械编号 (1 ~ 3)	返回退避点位置。	RW	位置型	419
71	J_Origin	机械编号 (1 ~ 3)	返回原点设定时的关节坐标数据。	R	关节型	343
72	M_Open	文件编号 (1 ~ 8)	返回已指定的文件或通信线路的开启状态。	R	整数型	378
73	M_NvOpen	视觉传感器编号 (1 ~ 8)	返回视觉传感器的线路连接状态。	R	整数型	376
74	C_Mecha	机械编号 (1 ~ 3)	返回机器人的型号。	R	字符串型	336
75	C_Maker	无	显示厂商方面的信息。(字符串最多64个字符)	R	字符串型	335
76	C_User	无	返回参数“USERMSG”的内容。(字符串最多32个字符)	R	字符串型	337
77	C_Date	无	将当前的日期以“年/月/日”的形式显示	R	字符串型	335

No	变量名称	排列要素数 ^{注1)}	内容	属性 ^{注2)}	数据的类型、单位	页
78	C_Time	无	将当前的时间以“小时:分:秒”的形式显示。	R	字符串型	337
79	M_BTime	无	返回电池剩余容量时间。(小时)	R	整数型、时间	347
80	M_Timer	定时器数量 (1~8)	始终计时。也可以设定值。[ms] 通过在程序中使用,可以测定正确的节拍时间。	RW	单精度实数型	391
81	P_Zero	无	位置变量的要素(X, Y, Z, A, B, C, FL1, FL2)全部为0的变量	R	位置型	424
82	M_PI	无	圆周率(3.1415...)	R	双精度实数型	382
83	M_Exp	无	自然对数的底(2.71828...)	R	双精度实数型	359
84	M_G	无	重力常数(9.80665)	R	双精度实数型	360
85	M_On	无	总是设定为1。	R	整数型	377
86	M_Off	无	总是设定为0。	R	整数型	377
87	M_Mode	无	模式选择开关的状态 MANUAL/AUTOMATIC(操作面板)/AUTOMATIC(外部机器))=(1/2/3)	R	整数型	375
88	M_SplPno	机械编号 (1~3)	在样条插补动作中,返回最新的已通过的路径点编号。	R	整数型	388
89	M_SplVar	机械编号 (1~3)	在样条插补动作中,返回路径点中已设定的任意数值。 此外,到重新设定值为止的期间,可保持任意的值。	RW	整数型	389
90	P_WkCord	工件坐标编号 (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	423
91	M_ESpd	机械编号 (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	358
92	P_ECord	机械编号 (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	413
93	M_Mxt	对象号机 (2~4)	在机器人间同步使用	RW	整数型	375
94	P_GCurr	对象号机 (2~4)	主站侧机器人的当前位置	R	位置型	415
95	M_Gps	监视编号 (1~8)	在高速位置获取功能(GPS功能)中,返回由Def Gps指令定义的监视编号P_Gps1~P_Gps8中保存的位置数据的个数。	R	整数型	364
96	P_Gps1 ~ P_Gps8	位置编号 (1~400)	在高速位置获取功能(GPS功能)中,用直交坐标返回Def Gps指令定义的条件成立时的位置数据。	R	整数型	417
97	M_Map1 ~ M_Map8	盒段数 (1~130)	在高速位置获取功能(GPS功能)中,返回Def Gps指令定义的条件下的工件存在段数。	R	整数型	374

注1) 机械编号..... 1~3、指定对应多任务处理功能的机械编号。
 插槽编号..... 1~32、指定支持多任务功能的插槽编号。
 输入编号..... 0~32767 (理论值),指定输入信号的位编号。
 输出编号..... 0~32767 (理论值),指定输出信号的位编号。
 注2) R..... 只可读取。
 RW..... 可读取、写入。

4.5.1 逻辑数

表示比较、输入输出的结果等。

以整数评价，不为 0 表示真，为 0 表示假。在代入时，真的情况给予 1。表 4-9 显示使用逻辑数的处理。

表 4-9：逻辑数的真假对应值

用逻辑数的 1 表示	用逻辑数的 0 表示
<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） • 开启开关 • 开启输入输出信号 • 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） • 关闭开关 • 关闭输入输出信号 • 抓手开（在抓手不流过电流） • 设定插入等的禁止 / 无效

4.6 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用户定义的用户定义函数。

(1) 用户定义函数

以 Def FN 语句定义。

例) Def FNMADD(MA, MB)=MA+MB

..... 定义求得 2 个值的和的函数 FNMADD。

函数名以 FN 开始，记述在第 3 个字符中显示数据类型的识别字符（C：字符串、M：数值、P：位置、J：函数），且指定在 8 个字符以内。

(2) 内建函数

内建函数一览如表 4-10 所示。

表 4-10：内建函数一览

区分	函数名（格式）	功能	页	结果
数值函数	Abs（<算式>）	给出绝对值	426	数值
	CInt（<算式>）	对算式的小数部分进行四舍五入，转换成整数	433	
	Deg（<算式：弧度>）	将角度单位 弧度（rad）转换为度（deg）	436	
	Exp（<算式>）	给出算式的指数函数的值	437	
	Fix（<算式>）	给出整数部	438	
	Int（<算式>）	给出不超过算式值的最大整数	440	
	Len（<字符串式>）	给出字符串的长度	442	
	Ln（<算式>）	给出自然对数	443	
	Log（<算式>）	给出常用对数	443	
	Max（<算式>...）	从任意个数的自变量中求出最大值	444	
	Min（<算式>...）	从任意个数的自变量中求出最小值	445	
	Rad（<算式：度>）	将角度单位 度（deg）转换成弧度（rad）	449	
	Sgn（<算式>）	检查算式的符号	457	
	Sqr（<算式>）	给出平方根	461	
	Rnd（<算式>）	产生随机数	452	
	SpISpd（<样条编号>）	可从任意的样条文件中登录的路径点信息，求出 MvSp1、EMvSp1 指令中不会发生错误的可指定的最大速度。	460	

区分	函数名 (格式)	功能	页	结果	
三角函数	ACos(<算式>)	给出反余弦 (Arc Cosine) 单位: 弧度 定义域: $-1.0 \sim +1.0$ 值域: $0 \sim \pi$	427	数值	
	ASin(<算式>)	给出反正弦 (Arc Sine) 单位: 弧度 定义域: $-1.0 \sim +1.0$ 值域: $-\pi/2 \sim \pi/2$	430		
	Atn(<算式>)	给出反正切 (Arc Tangent) 单位: 弧度 定义域: 数值 值域: $-\pi/2 \sim +\pi/2$	430		
	Atn2(<算式>, <算式>)	给出反正切 (Arc Tangent) 单位: 弧度 $\Theta = \text{ATN2}(\Delta y, \Delta x)$ 定义域: $\Delta y, \Delta x$ 的任意一个不为 0 的数值 值域: $-\pi \sim +\pi$	430		
	Cos(<算式>)	给出余弦 (Cosine) 单位: 弧度 定义域: 数值的范围 值域: $-1 \sim +1$	434		
	Sin(<算式>)	给出正弦 (Sine) 单位: 弧度 定义域: 数值的范围 值域: $-1 \sim +1$	458		
	Tan(<算式>)	给出正切 (Tangent) 单位: 弧度 定义域: 数值的范围 值域: 数值的范围	464		
	字符串函数	Asc(<字符串式>)	给出字符串式的第一个字符的字符代码		429
Bin\$(<算式>)		将算式的值转换成 2 进制字符串	431	字符串	
Chr\$(<算式>)		给出持有算式值字符代码的字符	433		数值
Cvi(<字符串式>)		将 2 个字符的字符串式转换为整数	435	字符串	
Cvs(<字符串式>)		将 4 个字符的字符串式转换为单精度实数	435		字符串
Cvd(<字符串式>)		将 8 个字符的字符串式转换为双精度实数	436	字符串	
Hex\$(<算式>)		将算式的值转换成 16 进制字符串	440		字符串
Left\$(<字符串式>, <算式>)		从第 1 自变量的字符串左部求出第 2 自变量中指定的长度的字符串	442	字符串	
Mid\$(<字符串式>, <算式>, <算式>)		从第 1 自变量的字符串中求出从第 2 自变量中指定的位置到第 3 自变量中指定的长度的字符串	444		字符串
Mirror\$(<字符串式>)		进行字符串的 2 进制数的位的镜像反转	445	字符串	
Mki\$(<算式>)		将算式的值转换成 2 个字符的字符串	446		字符串
Mks\$(<算式>)		将算式的值转换成 4 个字符的字符串	446	字符串	
Mkd\$(<算式>)		将算式的值转换成 8 个字符的字符串	447		字符串
Right\$(<字符串式>, <算式>)		从第 1 自变量的字符串右部求出第 2 自变量中指定的长度的字符串	452	数值	
Val(<字符串式>)		将字符串转换成数值	464		字符串
Str\$(<算式>)		将算式的值转换成 10 进制字符串	463	字符串	
Strpos(<字符串式>, <字符串式>)		在第 1 自变量字符串内求出第 2 自变量字符串的位置	462		字符串
StrLwr(<字符串>)		将 <字符串> 转换成小写。	461	字符串	
StrUpr(<字符串>)		将 <字符串> 转换成大写。	462		数值
CkSum(<字符串式>, <算式>, <算式>)		生成字符串的校验和 返回第 1 自变量的字符串中从第 2 自变量的位置到第 3 自变量的位置的字符的加算结果的低位字节的值	434	位置	
位置变量		Dist(<位置>, <位置>)	求得 2 点间的距离		437
		Fram(<位置 1>, <位置 2>, <位置 3>)	计算在 3 点指定的坐标系。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, 将平面的原点位置和姿势以返回值 (位置) 返回。以在机械构造无关系的 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。	439	字符
		Rdf11(<位置>, <数值>)	将指定的位置结构标签, 用字符数据返回。 自变量的 <数值>=R/L, 1=A/B, 2=F/N 会变为 Return	450	
	Setf11(<位置>, <字符>)	变更指定位置的结构标签。变更数据以字符指定。(R/L/A/B/F/N)	453	数值	
	Rdf12(<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> 返回轴编号 (1 ~ 8)。	451		数值
	Setf12(<位置>, <数值>, <数值>)	将指定位置的多旋转数据以数值 (-2 ~ 1) 变更。 公式的左边为变更的轴编号、右边为设定的数值。	454		

区分	函数名 (格式)	功能	页	结果
位置变量	Align(<位置>)	位置 1 的姿势轴 (A, B, C) 将返回取得最接近的直交姿势 (0, ±90, ±180) 的值。 在 5 轴型机器人中 ABC 的姿势数据意思不同, 因此无法使用。	428	数值
	Inv(<位置>)	求出反行列。	441	位置
	PtoJ(<位置>)	将位置数据转换成关节数据。	449	关节
	JtoP(<位置>)	将关节数据转换成位置数据。	441	位置
	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分, 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 对应的成分。	465	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 (圆柱体) 范围外 =0、范围内 =1 对象只有 XYZ 的坐标值, ABC 的姿势数据会被忽略。	466	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 对象只有 XYZ 的坐标值, ABC 的姿势数据会被忽略。	466	数值
	PosCq(<位置>)	确认 <位置> 是否在动作范围内。	448	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。	448	位置
	CalArc(<位置 1>, <位置 2>, <位置 3> 数值 1, 数值 2, 数值 3, 位置 4)	输出由 <位置 1>, <位置 2>, <位置 3> 所得的圆弧的情报。	432	数值
	SetJnt(<J1 轴>, <J2 轴>, <J3 轴>, <J4 轴> <J5 轴>, <J6 轴>, <J7 轴>, <J8 轴>)	设定至关节变量的值。	455	关节
	SetPos(<X 轴>, <Y 轴>, <Z 轴>, <A 轴> <B 轴>, <C 轴>, <L1 轴>, <L2 轴>)	设定至位置变量的值。	456	位置
	SplPos(<样条编号>, <路径点编号>, <帧转换>)	取出登录在任意样条文件中的路径点数据。	459	位置
SplECord(<样条编号>, <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。	458	位置	

4.7 指令语句一览

下列为各指令语句的说明页一览。以使用频率高的顺序表示。

(1) 动作控制相关的指令

指令	说明	页
Mov (Move)	关节差补	261
Mvs (Move S)	直线差补	273
Mvr (Move R)	圆弧插补	267
Mvr2 (Move R2)	圆弧插补 2	269
Mvr3 (Move R3)	圆弧插补 3	271
Mvc (Move C)	圆形插补	266
Mva (Move Arch)	圆弧动作插补	262
MvSpl (Move Spline)	样条插补	275
Mxt	通过来自外部的指令进行的直接动作	280
MvTune	指定与用途相适应的最佳动作模式	278
Ovrd	指定整体的速度	295

指令	说明	页
Spd (Speed)	指定直线、圆弧插补动作时的速度	312
JOvrd	指定关节插补动作时的速度	257
Cnt (Continuous)	指定连接轨迹模式	194
Accel	指定加减速速度	174
ColChk (Col Check)	碰撞检测	196
CavChk On	干涉回避	183
Cmp Jnt (Comp Joint)	在关节坐标系指定弹性伺服	186
Cmp Pos	在直交坐标系指定弹性伺服	188
Cmp Tool	在工具坐标系指定弹性伺服	190
Cmp Off	弹性伺服设定无效	192
CmpG	指定弹性伺服的增益	193
Oadl (Optimal Acceleration)	指定最佳加减速	289
LoadSet (Load Set)	指定抓手附加条件	260
Prec (Precision)	指定高精度模式	297
Torq (Torque)	指定各轴的转矩	321
JRC (Joint Roll Change)	前端轴可以多旋转	258
Fine	指定机器人的定位范围 (脉冲)	240
Fine J	指定机器人的定位范围 (关节轴)	241
Fine P	指定机器人的定位范围 (直线距离)	242
Servo (Servo)	伺服电机电源的开启 / 关闭	309
SpdOpt (Speed optimize)	直线插补动作时的速度调整	313
Wth (With)	往动作指令的附随指令	324
WthIf (With If)	往动作指令的附条件附随指令	324
EMvs (E Move S)	Ex-T 控制直线插补	233
EMvc (E Move C)	Ex-T 控制圆形插补	227
EMvr (E Move R)	Ex-T 控制圆弧插补	228
EMvr2 (E Move R2)	Ex-T 控制圆弧插补	230
EMvr3 (E Move R3)	Ex-T 控制圆弧插补	231
EMvSpl (E Move Spline)	Ex-T 样条插补	235

(2) 程序控制相关的指令

指令	说明	页
Rem(Remarks)	注释 (')	303
If...Then...ElseIf...Else...EndIf	条件分支	253
Select Case	可多个分支	308
GoTo	跳转	249
GoSub (~ Return) (Go Subrouine)	子程序跳转	248
Reset Err(Reset Error)	将错误复位 (禁止使用默认)。	304
CallP	调用程序	180
FPrm	调用程序的自变量定义	244
Dly (Delay)	定时器	221
Function... FEnd	Function 过程定义	245

指令	说明	页
Exit	脱离过程 / 重复处理。	239
Hlt (Halt)	程序中斷	251
Include	#Include 陈述句声明	255
End	程序结束	237
On... GoSub (ON Go Subroutine)	依据数值子程序跳转	291
On...GoTo	依据数值跳转	292
For ~ Next	循环	243
While ~ WEnd (While End)	附条件循环	323
Open	开启文件	293
Print	输出数据	298
Input	输入数据	256
Close	关闭文件	184
Remove	删除文件	304
Save	程序储存	307
ColChk (Col Check)	碰撞检测功能的有效 / 无效	196
HOpen/HClose (抓手 Open/Close)	抓手的开闭	252
Error	用户错误	238
Skip(Skip)	动作中的跳转	311
Wait (Wait)	条件等待	322
Clr	信号清除	185
SplWrt (Spline Write)	创建样条文件	316
SplFWrt (Spline Frame Write)	样条文件内帧转换信息改写	315
GpsChk (Get position check)	Def Gps 指令, 或是 Def Map 指令定义的条件下的监视开始 / 结束	250

(3) 定义指令

指令	说明	页
Dim	排列变量声明	220
Def Plt (Define 码垛)	托盘声明	216
Plt	托盘位置计算	296
Def Act	中断定义	202
Act	中断监视开始、结束	175
Def Arch	定义圆弧动作的弧形形状	205
Def Jnt (Define Joint)	定义关节位置变量	212
Def Pos (Define Position)	定义直交位置变量	218
Def Work	定义工件坐标变量	219
Def Inte/Def Long/Def Float/Def Double	定义整数、实数变量	210
Def Char (Define Character)	字符变量的定义	206
Def IO (Define IO)	信号变量的定义	211
Def FN (Define Function)	用户函数的定义	207
Const	常数的定义	200
Static	Static 变量的声明	318
Title (Title)	设定程序标题	319
Base	设定机器人基本位置	177
Tool (Tool)	设定工具长度	320
SetCalFrm (Set Calibration Frame)	设定帧转换所使用的坐标系	310

指令	说明	页
Def Gps (Define get position)	通过高速位置获取功能 (GPS 功能) 定义获取位置情报时的监视条件	208
Def Map	通过高速位置获取功能 (GPS 功能) 定义盒内有无工件 (映射处理) 的监视条件。	213

(4) 多任务相关

指令	说明	页
XLoad (X Load)	对其它任务插槽的程序加载	326
XRun (x Run)	执行其他任务插槽的程序	328
XStp (X Stop)	停止其他任务插槽的程序	330
XRst (X Reset)	程序复位中断中的其他任务插槽	327
XClr (X clear)	解除指定任务插槽的程序加载	325
GetM (Get Mechanism)	取得机械控制权	247
RelM (Release Mechanism)	释放机械控制权	303
Priority	更改任务插槽的优先级	299
Reset Err (Reset Error)	错误复位 (禁止使用默认)	304

(5) 通信

指令	说明	页
Com Off	禁止来自于通信线路的中断。	199
Com On	允许来自于通信线路的中断。	199
Com Stop	暂时停止来自于通信线路的中断。(接收数据。)	199
On Com GoSub (ON Communication Go Subroutine)	通信中断的定义	290
Open	开启文件	293
Close	关闭文件	184
Input	输入文本数据	256
Print	输出文本数据	298
NVOpen	与视觉传感器连接, 登录到视觉传感器。	283
NVClose	切断与视觉传感器的连接。	281
NVLoad	将指定的视觉程序设为可启动状态。	282
NVRun	启动指定的视觉程序。	285
NVTrg	向视觉传感器请求摄像, 在指定时间后取得编码器值。	287
EBRead	指定视觉传感器的符号名并读取数据。	222
EBWrite	指定视觉传感器的符号名并写入数据。	225

(6) 其他

指令	说明	页
ChrSrch (Character Search)	从字符排列中检索字符串。	182
PrmRead	读取参数值	300
PrmWrite	写入参数	301
GetPos (Get Position)	备用	-
PVScal	使用视觉传感器校准数据, 将视觉传感器的图像坐标转换为机器人的世界坐标	302

4.8 运算符

不执行数值的实数或整数的类型声明。取而代之，依据运算的种类，强制的转换类型。（参照表 4-11。）
 依照左项和右项的数据类型的组合，运算结果的类型如下所示。

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

表 4-11: 基于运算的数据类型转换表

左项型	运算	右项型				
		字符串	数值		位置	关节
			整数	实数		
字符串	代入 = 加算 + 比较 (比较运算符)	字符串 字符串 整数	- - -	- - -	- - -	- - -
整数	加算 + 减算 - 乘算 * 除算 / 整数除算 \ 余数 Mod 指数 ^ 代入 = 比较 (比较运算符) 逻辑 (逻辑运算符)	- - - - - - - - - -	整数 整数 整数 整数 整数 整数 整数 整数 整数 整数	实数 实数 实数 实数 整数 整数 实数 整数 整数 整数	- - - - - - - - - -	- - - - - - - - - -
实数	加算 + 减算 - 乘算 * 除算 / 整数除算 \ 余数 Mod 指数 ^ 代入 = 比较 (比较运算符) 逻辑 (逻辑运算符)	- - - - - - - - - -	实数 实数 实数 实数 整数 整数 实数 整数 整数 整数	实数 实数 实数 实数 整数 整数 实数 整数 实数 整数	- - - - - - - - - -	- - - - - - - - - -
位置	加算 + 减算 - 乘算 * 除算 / 整数除算 \ 余数 Mod 指数 ^ 代入 = 比较 (比较运算符) 逻辑 (逻辑运算符)	- - - - - - - - - -	- - 位置 位置 - - - - - -	- - 位置 位置 - - - - - -	位置 位置 位置 位置 - - - 位置 - -	- - - - - - - - - -
关节	加算 + 减算 - 乘算 * 除算 / 整数除算 \ 余数 Mod 指数 ^ 代入 = 比较 (比较运算符) 逻辑 (逻辑运算符)	- - - - - - - - - -	- - 关节 关节 - - - - - -	- - 关节 关节 - - - - - -	- - - - - - - - - -	关节 关节 - - - - - - 关节 - -
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 余数: 余数运算
 比较: 比较运算 逻辑: 逻辑运算 (除了逻辑否定以外)

[注意]

- 以 ' - ' 记述部分的运算，未被定义。
- 整数和整数的乘除算的结果，乘算的情况会变为整数型，除算的情况会变为实数型。
- 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转换为整数（四舍五入）后运算。

4.9 运算的优先级

公式中包含多个运算符的情况下，按以下表 4-12 中所示的优先级进行运算。

表 4-12: 运算的优先级

运算（运算符）	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) 单项运算符 (+, -)	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 (=, <, >, <=, >=, =>)	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

4.10 程序的控制构造深度

组合程序的情况下，需要考虑控制构造的深度。

表 4-13 使用所示指令时，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错误。

表 4-13: 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。

4.11 保留字

所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）等会变为保留字。

4.12 指令的详细说明

4.12.1 记载项目的阅读方法

- 【功能】 : 表示指令语句的功能。
- 【格式】 : 表示指令语句自变量的输入方法。
 - < > 表示为自变量。
 - [] 表示可省略。
 - 表示需要有空格。
- 【术语】 : 表示自变量的意义、范围等。
- 【程序示例】 : 表示程序示例。
- 【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。
- 【相关参数】 : 表示相关参数。
- 【相关系统状态变量】 : 表示相关系统变量。
- 【相关指令】 : 表示相关指令语句。

4.12.2 各指令语句的说明

如下所示为以英文顺序说明各指令。

Accel

【功能】

机器人动作时的加速度及减速度以比例（%）指定。
最佳加减速度时也有效。

※ 最佳加减速度时的加减速度时间是以使用 Oad1 指令时算出最佳的时间，考虑 M_SetAd1 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 (%)>]
```

【术语】

- <加速・减速度比例> 1~100（%）。从速度0到最高速度为止以比例指定夹爪的加减速度。可用常数或变量记述。省略时为100，100为最高加减速度。单位：[%]
- <上升时加减速度比例> 指定Mva指令的圆弧动作上升时的加减速度比例。省略时为100。可用常数或变量记述。
- <下降时加减速度比例> 指定Mva指令下的圆弧动作下降时的加减速度比例。省略时为100。可用常数或变量记述。

【程序示例】

```
1 Accel 50,100                            ' 指定重负载（标准加减速度时间为 0.2 秒的情况下，加速时间为 0.4 秒、减速度时间为 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 指令动作时的上升、下降时的倍率修调设为 20
7 Mva P3,1
```

【说明】

- (1) 标准加减速度时间取决于每个机器人，设定与此对应的比例（%）。系统初始值为 100%、100%。（最高加减速度）
- (2) 执行本指令后更改的加速比例，在程序复位时及执行 End 语句时会被重新设定为系统初始值。
- (3) Cnt 有效时的圆滑动作的轨迹路径根据加速度及动作速度会有所不同。此外，以一定速度执行圆滑动作的情况下，请将加速度和减速度设为相同值。初始状态下，Cnt 为无效。
- (4) 在最佳加减速度控制时 (Oad1 On) 也会为有效。

【相关指令】

Oad1 (Optimal Acceleration)、LoadSet (Load Set)

【相关系统状态变量】

M_Acl/M_DAc1/M_NAc1/M_NDAcl/M_AclSts

【相关参数】

JADL

Act

【功能】

指定根据动作中的信号等进行的 interrupt 处理的允许 / 禁止。

【格式】

Act □ < 优先编号 >=<1/0/-1>

【术语】

<优先编号> 0 : 允许或禁止全部中断。
 1~8 : 指定以 Def Act 语句定义的中断优先编号。
 请务必在 Act 指令的后面空出空格记述优先编号。
 继续记述 Act1 时, 会变为变量名的声明语句。

<1/0/-1> 1 : 允许中断的情况
 0 : 禁止使用时, 变量也可以。
 -1 : <优先编号> 为 1~8 时, 对中断处理中进行解除。此外, 中断监视也停止。

【程序示例】

①从 P1 往 P2 的移动过程中, 输入信号 1 开启时 (变为 1 时), 到此信号变为 0 为止, 会执行环路

```
1 Def Act 1, M_In(1)=1 GoSub *INTR      ' 向中断 1 的条件中分配输入信号 1。
2 Mov P1
3 Act 1=1                               ' 允许中断 1。
4 Mov P2
5 Act 1=0                               ' 禁止中断 1。
:
10 *INTR
11 If M_In(1)=1 GoTo *INTR              ' M_IN(1) 信号在变为 0 为止会执行环路
12 Return 0
```

②从 P1 往 P2 的移动中, 输入信号 1 开启时 (变为 1 时), 将中断动作并开启输出信号 10 开启

```
1 Def Act 1, M_In(1)=1 GoSub *INTR      ' 向中断 1 的条件中分配输入信号 1。
2 Mov P1
3 Act 1=1                               ' 允许中断 1。
4 Mov P2
:
10 *INTR
11 Act 1=0                               ' 将中断 1 设为禁止。
12 M_Out(10)=1                          ' 开启输出信号 10。
13 Return 1                              ' 返回至发生中断的下一行。
```

③从 P1 向 P2 的移动过程中, 输入信号 1 开启时 (变为 1 时), 将中断动作并在等待信号变为 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 Mov P1
13 Act 1=1                               ' 允许中断 1。
14 Mov P2
:
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) 优先编号 0 在程序开始运行时为“允许”的状态。将〈优先编号〉0 设定为禁止时，即使优先编号 1～8 变为“允许”，也不会执行中断。
- (2) 优先编号 1～8 在程序运行开始时为“禁止”的状态。
- (3) 中断在下列的条件成立时进行。
 - 〈优先编号〉0 为“允许”
 - Def Act 语句的条件成立
 - 将通过 Def Act 指定的〈优先编号〉以 Act 语句设为有效时
- (4) 从中断处理开始的返回，是以 Return 0 或 Return 1 的其中一个进行记述，且指定中断处理结束后的返回地点。但是，从中断处理开始以 Return 1 返回到下一行的情况下，请禁止在中断处理内中断。不禁止中断而直接在中断条件成立的情况下再次执行中断处理后返回到下一行，因此有可能不执行行而进行跳转。
- (5) 即使机器人在插补动作中也中断，并执行以 Def Act 语句定义的中断处理。
- (6) 在中断处理中，将以“禁止”状态执行该〈优先编号〉。
以 Return 指令从中断处理返回时，会再次返回到“允许”状态。
- (7) 通信中断 (Com) 比以 Def Act 语句定义的中断优先级别更高。
- (8) 与其它中断的优先顺序关系如下所示。
Com > Act > WthIf (Wth)
- (9) 指定 Act 0=-1 时，将发生 L3110 错误。

【相关指令】

Def Act、Return

Base

【功能】

可以变更机器人当前位置控制基准的世界坐标系（移动及旋转）。有直接指定基本转换数据的方法与指定事先定义的工件坐标系编号的方法这2种方法。

本功能会对机器人的示教数据及JOG动作产生较大影响。请在阅读“4.4 机器人的坐标系说明”之后，小心使用。

【格式】

Base □ <基本转换数据>	’ 直接指定基本转换数据
Base □ <基本坐标编号>	’ 通过基本坐标编号（工件坐标系编号）间接指定基本变换数据

【术语】

- <基本转换数据> 以位置常数或位置变量指定基本转换数据。
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。
- <基本坐标编号> 将系统初始值或参数(工件坐标系)中设定的值指定为基本转换数据。
以数值型的常数或变量,在值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。

【程序示例】

以基本转换数据指定

- | | |
|-------------------------------|---|
| 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 | ’ 在新的世界坐标系中向 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 | ’ 在新的世界坐标系中向 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) 设定为基本转换数据) |

以基本坐标编号指定

- | | |
|----------|---|
| 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 | ’ 在新的世界坐标系中向 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 | ’ 在新的世界坐标系中向 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) 设定为基本转换数据) |

【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，使用机器人的当前位置将通过 Fram 函数等定义的坐标值指定为基本转换数据时，将对坐标值进行逆向变换后设定。（例：Base Inv(P1)）

此外，指定工件坐标系编号时，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 · · · 向 X 轴方向平行移动的距离

Y · · · 向 Y 轴方向平行移动的距离

Z · · · 向 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，以顺时针为正的旋转方向来获取 A、B、C。

- (2) 结构标志的内容没有含义。

- (3) 通过本指令变更的〈基本转换数据〉会被保存在参数 MEXBS 中，〈基本坐标编号〉会被保存在参数 MEXBSNO 中。保存后的值，即使在控制器的电源 OFF 后也会保持。

使用连续反复执行 Base 指令、Tool 指令、M_Tool 的程序时，可能会无法即时进行参数保存，从而发生错误 C7091（参数保存失败），应加以注意。发生错误 C7091 时，应如下所示变更执行 Base 指令、Tool 指令的位置。

例)

*MAIN	Base PB	Base PB
Base PB	Tool PT	Tool PT
Tool PT	➔	*MAIN
If M_In(20)=1 Then		If M_In(20)=1 Then
GoSub *SUB1		GoSub *SUB1
EndIf		EndIf
GoTo *MAIN		GoTo *MAIN

- (4) 执行基本转换后，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。

- (5) 根据机器人的机型（本体的构造），基本转换数据的有效轴成分会有所不同。第 500 页的“5.6 关于标准 TOOL 坐标”请参照。

此外，与本指令相关的说明位置如下所示，请一并参照。

第 155 页的“4.4.1 关于机器人的坐标系”

第 156 页的“4.4.2 关于基本转换”

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

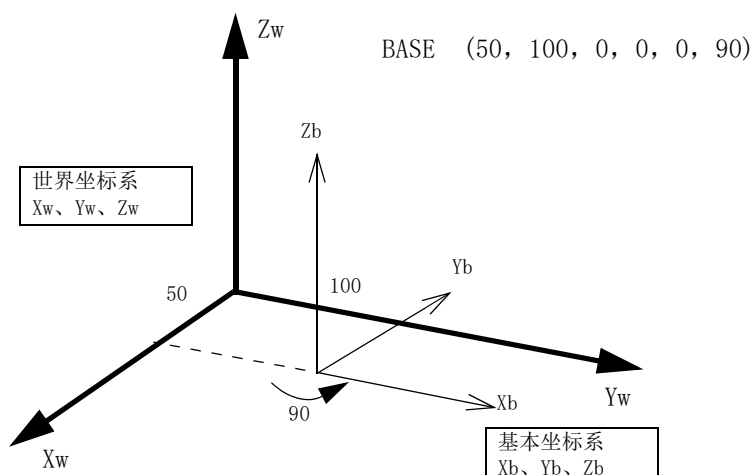


图 4-15: 坐标系的观念图

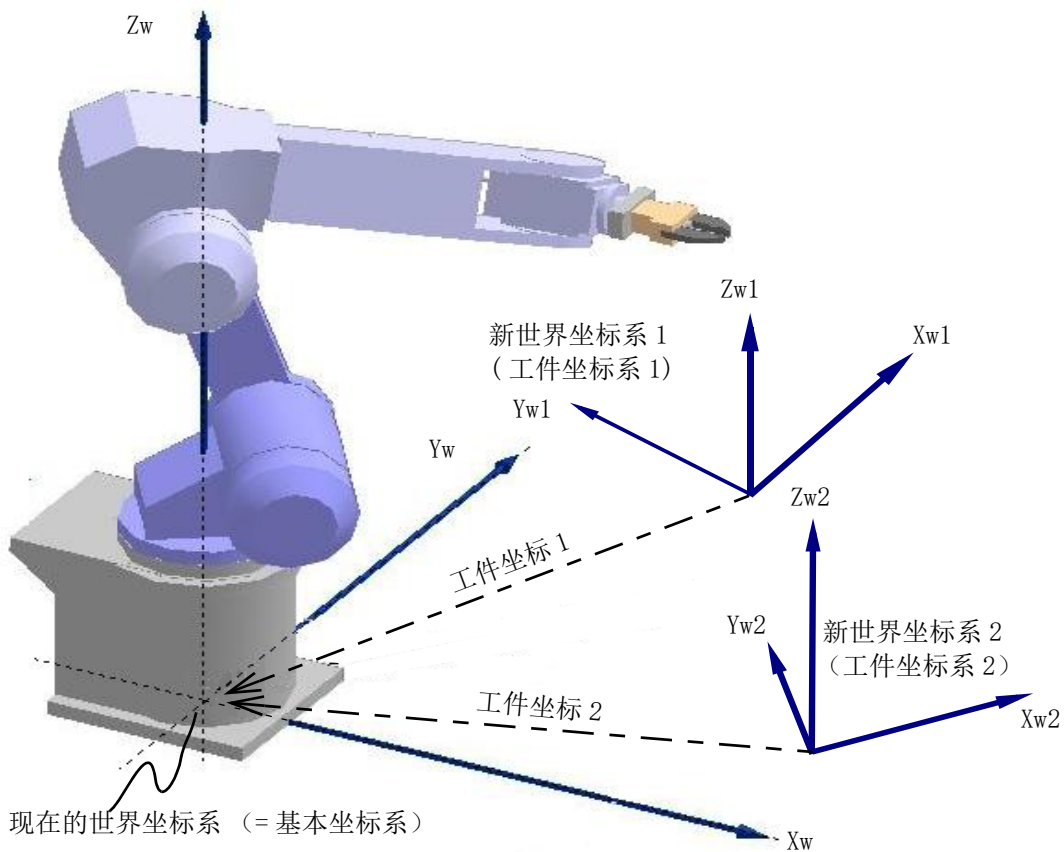


图 4-16: 指定了工件坐标系编号的基本变换

⚠ 注意

执行基本转换后，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，从而造成设备损坏或人身事故。

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。

【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

CallP

【功能】

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

【格式】

CallP □ "<程序名>" [, <自变量>[, <自变量>] . . .]

【术语】

<程序名> 以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141页的“(1) 程序名”。
<自变量> 程序被调用时，将指定传递给该程序中的变量或常数。自变量的最大个数为16。

【程序示例】

① 在调用程序中传递自变量时

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                   ' 在此会返回到主程序。

```

※ 在主程序的步号 2、4 被执行时，M1、P1、P2 被分别设定为子程序的 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 在调用程序中没有传递自变量时

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                   ' 在此会返回到主程序。

```

【说明】

- (1) 以 CallP 指令被执行子程序，将以 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 指令的情况下，最终行执行后，返回到主程序。
- (2) 传递自变量的情况下，以 CallP 指令记述自变量的同时，需要在子程序的前头以 FPrm 指令定义自变量。
- (3) 主程序和子程序，自变量的类型及个数不同的情况下，执行时会发生错误。
- (4) 将程序复位的情况下，控制会返回到主程序的前头。
- (5) 以主程序执行的定义语句（Def Act、Def FN、Def Plt、Dim 指令）在子程序中为无效。从子程序返回时会再次变为有效。
- (6) 工具转换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) 可在子程序中使用 CallP，并且可以执行其他的子程序。
但是，无法调用主程序及已经在其他的任务插槽中正在执行的程序。此外，也无法调用自身的程序。
- (8) 可以从最初的主程序开始，执行 8 阶段（阶层）的子程序（CallP）。
- (9) 可以依据自变量从主程序向子程序传递变量的值，但是无法将子程序中的处理结果代入自变量，并传递给主程序。将子程序的处理结果使用在主程序的情况下，请使用外部变量来传递值。

 注意

通过 CallP 指令执行子程序时，在返回主程序时数据将被写入至控制器内部的非易失性存储器。

重复执行 CallP 指令会增加控制器内部非易失性存储器的写入次数，可能会导致故障，因此应将参数 AUTOSAVE 设定为 0（不保存）。详细内容请参照“5.4 程序参数”中的 AUTOSAVE。

例：程序 1 至程序 2 高速重复执行。

< 程序 1 >

```
1 *LOOP
2 CallP "2"
3 Goto *LOOP
```

< 程序 2 >

```
1 PS=P_Curr
2 End
```

【相关指令】

[FPrm](#)

ChrSrch (Character Search)

【功能】

从字符串数组中检索字符串。

【格式】

ChrSrch □ < 字符串数组变量 >, < 字符串 >, < 检索结果存储位置 >

【术语】

< 字符串数组变量 >	指定检索对象的字符串数组。
< 字符串 >	指定检索的字符串。
< 检索结果存储位置 >	将设定找到了检索字符串的要素编号。

【程序示例】

```

1 Dim C1$(10)
2 C1$(1)="ABCDEFGG"
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$(8)="ROBOT"
10 C1$(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) 从字符串数组变量检索字符串，将完全一致的字符串数组的要素编号设定在 < 检索结果储存处 >。部分一致的不会被检索。
在上述程序示例中，即使记述为
ChrSrch C1\$(1), "ROBO", M1,
也检索不到一致的字符串。
- (2) 找不到检索字符串的情况下，< 检索结果存储位置 > 会被设定为 0。
- (3) 字符串的检索会从要素编号 1 开始依编号顺序执行，将设定为最初找到的要素编号。
在上述程序示例中，即使记述为
ChrSrch C1\$(3), "MELFA", M2,
M2 也会被设定为 2。(C1\$(2) 和 C1\$(6) 已设定为相同字符串)
- (4) 检索对象的 < 字符串数组变量 > 仅为一维数组。指定了二维以上的变量的情况下，执行时会发生错误。

CavChk On

【功能】

将干涉回避的停止功能设为有效。

本功能限定为对应机型。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 编号>[, NOErr]]

【术语】

<On/Off>	On: 将干涉回避的停止功能设为有效。 Off: 将干涉回避的停止功能设为无效。
<机器人 CPU 编号>	指定使干涉回避的停止功能有效 / 无效的机器人 CPU。 省略或指定为 0 时, 所连接的所有机器人均为对象。 以 0 ~ 3、常数或变量进行记述。
NOErr	CR800-D 系列请指定为 0。指定 0 以外时将发生报警 (L4930)。 检测到干涉时, 不发生错误。 省略时, 将发生错误 (L240n)。

【程序示例】

参照第 568 页的“5.24.10 采样程序”。

【说明】

- (1) 将干涉回避的停止功能设为有效。初始状态取决于参数 CAV 的设定。
设定为参数 CAV 中不可使用时, 将发生错误 (L4930)。
- (2) 省略了机器人 CPU 编号时, 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 编号时, 指定位置的机器人 CPU 为干涉回避无效时, 将发生错误 (L4930)。
- (4) 抓手、工件模型可通过 LoadSet 指令指定。
- (5) 跟踪有效中时, 无法使其有效。(发生错误 (L4936))

Close

【功能】

关闭指定文件编号的文件（含通信线路）。

【格式】

```
Close □ [[#]<文件编号>[, [[#]<文件编号>•••]]]
```

【术语】

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

【程序示例】

```
1 Open "temp.txt" For Append As #1 ' 将称为 temp.txt 的文件以 Append 模式作为文件编号 1 开启。
2 Print #1, "abc" ' 在文件中写入 "abc"。
3 Close #1 ' 关闭文件。
```

【说明】

- (1) 根据 Open 指令将所开启的文件关闭。缓存内有剩余数据的情况下，会进行清除。
关闭文件时，缓存内剩余的数据将被进行以下处理。

表 4-14：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
通信线路 接收缓存	缓存内容被删除。
通信线路 发送缓存	（在 PRINT 指令执行时立即发送，因此发送缓存内没有剩余数据）
文件 读出缓存	缓存内容被删除。
文件 写入缓存	将缓存内容写入文件后，关闭文件。

- (2) 依据 End 语句的执行也会被关闭。
(3) 省略文件编号的情况下，关闭全部的文件。

【相关指令】

Open、Print、Input

Clr

【功能】

对通用输出信号 / 程序内区域性数值变量 / 数值外部变量进行清除。

【格式】

Clr □ <类型>

【术语】

<类型>

可指定常数或变量。

0：执行以下的1~3的全部。

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：ORST0~ORS18160指定。
请参照第514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变量全部清零。

3：将外部数值变量及外部数值数组变量全部(系统外部变量和用户定义外部变量)清零。外部位置变量不会清除。

【程序示例】

① 根据输出复位模式输出通用输出信号。

```
1 Clr 1
```

② 将程序内的区域性数值变量、数值数组变量清零。

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

```
2 Def Inte IVAL
```

```
3 Clr 2          ' MA(1)~MA(10)、IVAL及程序内区域性数值变量清零
```

③ 将外部数值变量及外部数值数组变量全部清零。

```
1 Clr 3
```

④ 同时执行上述①~③。

```
1 Clr 0
```

【相关参数】

ORST0 ~ ORS18160

【相关系统状态变量】

M_In/M_Inb/M_In8/M_Inw/M_In16/M_In32、M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32

Cmp Jnt (Comp Joint)**【功能】**

在关节坐标系中指定轴并开始机器人的柔软控制模式（弹性伺服模式）。

注）本指令限定为对应机型。请参照【可使用的机器人机型】。

【格式】

Cmp □ Jnt, < 轴指定 >

【术语】

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

【程序示例】

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

【说明】

- (1) 在关节坐标系中可以以轴为单位执行机器人柔软控制。例如，水平多关节机器人中在上下方向插入针脚的情况下，通过将 J1 轴和 J2 轴变得柔软，使针脚插入在一定程度上可以更顺畅进行（参照上记程序示例）。
- (2) 柔软度可依据设定弹性常数的 CmpG 指令来指定。机器人为 RH-FRH/RH-CRH 系列的情况下，在水平轴的 J1、J2 轴指定 0.0，使其变为无伺服相当的功能（弹性常数=0）（垂直轴即使设定为 0.0，也不会变为无伺服相当）。此外，在此功能下，移动到超过动作限制位置时，请注意偏移量不要过大）。并且，使用此无伺服相当的功能的情况下，以下 (4)、(5) 的功能无法使用。
- (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等时，无法移动到超过各关节轴的动作限制位置。
- (5) 依据压制等，本来的目的位置和实际的机器人位置的偏移量超过 200mm 时，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 指令时，会发生错误。请解除 Cmp Off 指令后再执行这些指令。
- (7) 本指令在动作时，从伺服关闭变为开启时，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，由于无法在示教单元中解除弹性伺服模式的设定，因此请通过在程序执行本指令或在示教单元的程序编辑画面直接执行进行设定。
- (9) 变更轴指定的情况下，请以 Cmp Off 指令解除伺服弹性模式后，再次执行 Cmp Jnt 指令。
- (10) 弹性伺服模式仅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 指令指定定位完成条件时，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伺服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）
请将该参数的设定值逐渐变更为较大的值，并设定为不会发生误差过大 1 错误的值。

⚠ 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。在执行程序编号的变更及 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 模式下操作时，由于 JOG 操作和弹性伺服模式下控制的坐标系方向不同，因此机器人会向与想移动的方向不同的方向动作。

 **注意**

在弹性伺服模式的状态示教位置时，请将伺服关闭后再执行。伺服开启的状态下直接执行示教操作时，并不是实际的机器人位置，而是原本的指令位置，机器人会向和示教运行时不同的位置动作，因此请特别注意。

 **注意**

请在机器人静止时开始弹性伺服模式。如果在机器人静止前开始弹性伺服模式，当解除弹性伺服模式时，机器人可能会发生较大振动。

【可使用的机器人机型】

RH-FRH 系列、RH-CRH 系列

【相关系统状态变量】

M_CmpDst

【相关指令】

Cmp Off、CmpG、Cmp Tool、Cmp Pos

【相关参数】

CMPJCLL

Cmp Pos

【功能】

在直角坐标系指定轴方向，并开始柔软控制机器人的模式（弹性伺服模式）。

【格式】

Cmp □ Pos, < 轴指定 >

【术语】

<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。

【程序示例】

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 轴变为柔软状态
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	' 往工具坐标系的 Z- 方向 100mm 退避
9 Cmp Off	' 返回到平常状态。

【说明】

- (1) 在直角坐标系，可以执行机器人柔软控制。例如，在上下方向插入针脚的情况下，通过将 X、Y、A、B 轴方向变得柔软，使针脚插入在一定程度上变得顺畅。
- (2) 柔软度可依据 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等时，无法移动到超过各关节轴的动作限制位置。
- (5) 指令位置 and 实际位置的偏移量，可以依据 M_CmpDst 来读出。可以使用此变量来确认针脚的插入成功与否等。
- (6) 依据压制等，本来的目的位置和实际的机器人位置的偏移量超过 200mm 时，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 指令时，会发生错误。请解除 Cmp Off 指令后再执行这些指令。
- (8) 本指令在动作时，从伺服关闭变为开启时，机器人的位置会变化。此外，保持重力方向的轴，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，由于无法在示教单元中解除弹性伺服模式的设定，因此请通过在程序执行本指令或在示教单元的程序编辑画面直接执行进行设定。
- (10) 变更轴指定的情况下，请以 Cmp Off 指令解除伺服弹性模式后，再次执行 Cmp Pos 指令。
- (11) 使机器人在接近特异点的地方动作时，会发生错误且机器人有可能会发生大动作・振动，因此，请不要在接近特异点地方动作。假如发生此状况的情况下，请在伺服关闭状态下，执行（或者重新启动电源）Cmp Off 指令并解除弹性伺服模式后，将机器人从特异点远离后，重新将弹性伺服模式设为有效。
- (12) 弹性伺服模式仅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 指令指定定位完成条件时，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。

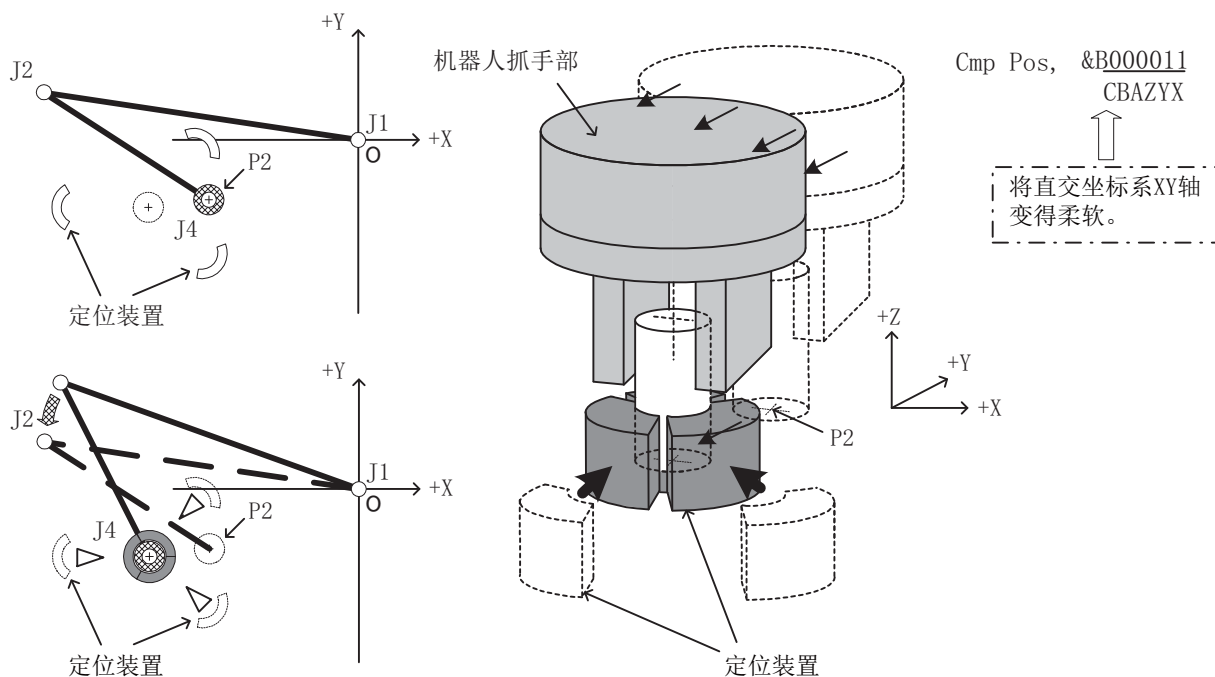


图 4-17: 伺服弹性模式使用例

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。在执行程序编号的变更及 JOG 操作时，请特别注意。

注意

以 Cmp Pos 指令作为弹性伺服模式的状态下执行 JOG 操作的情况下，请在直交 JOG 模式下操作。在其它的 JOG 模式下操作时，由于 JOG 操作和弹性伺服模式下控制的坐标系方向不同，因此机器人会向与想移动的方向不同的方向动作。

注意

在弹性伺服模式的状态示教位置时，请将伺服关闭后再执行。伺服开启的状态下直接执行示教操作时，并不是实际的机器人位置，而是原本的指令位置，机器人会向和示教运行时不同的位置动作，因此请特别注意。

注意

请在机器人静止时开始弹性伺服模式。如果在机器人静止前开始弹性伺服模式，当解除弹性伺服模式时，机器人可能会发生较大振动。

【相关系统状态变量】

M_CmpDst

【相关指令】

Cmp Off、CmpG、Cmp Tool、Cmp Jnt (Comp Joint)

Cmp Tool

【功能】

在工具坐标系指定机器人方向，并使指定机器人开始柔软控制模式（弹性伺服模式）。

【格式】

Cmp □ Tool, < 轴指定 >

【术语】

<轴指定> 将柔软控制工具坐标系的轴以位模式指定。
1:有效、0:无效 &B000000
对应CBAZYX轴。

【程序示例】

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 Tool, &B011011	' 将工具坐标系的 X、Y、A、B 轴变为柔软状态
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	' 往工具坐标系的 Z- 方向 100mm 退避
9 Cmp Off	' 返回到平常状态。

【说明】

- (1) 在工具坐标系，可以执行机器人柔软控制。关于工具坐标系请参照第 500 页的“5.6 关于标准 TOOL 坐标”。
- (2) 例如，在工具坐标的 Z 轴方向插入针脚的情况下，通过将 X、Y、A、B 轴变得柔软，使针脚插入可以更顺畅进行。
- (3) 柔软度可依据 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等时，无法移动到超过各关节轴的动作限制位置。
- (6) 指令位置 and 实际位置的偏移量，可以依据 M_CmpDst 来读出。可以使用此变量来确认针脚的插入成功与否等。
- (7) 依据压制等，本来的目的位置和实际的机器人位置的偏移量超过 200mm 时，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 指令时，会发生错误。请解除 Cmp Off 指令后再执行这些指令。
- (9) 本指令在动作时，从伺服关闭变为开启时，机器人的位置会变化。此外，保持重力方向的轴，会有落下的可能。
- (10) 可以在弹性伺服模式里执行 JOG 操作。但是，由于无法在示教单元中解除弹性伺服模式的设定，因此请通过在程序执行本指令或在示教单元的程序编辑画面直接执行进行设定。
- (11) 变更轴指定的情况下，请以 Cmp Off 指令解除伺服弹性模式后，再次执行 Cmp Tool 指令。
- (12) 垂直 5 轴型机器人 (RV-4FRJL) 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时，会发生错误且机器人有可能会发生大动作・振动，因此，请不要在接近特异点地方动作。假如发生此状况的情况下，请在伺服关闭状态下，执行（或者重新启动电源）Cmp Off 指令并解除弹性伺服模式后，将机器人从特异点远离后，重新将弹性伺服模式设为有效。
- (14) 弹性伺服模式仅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 指令指定定位完成条件时，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。

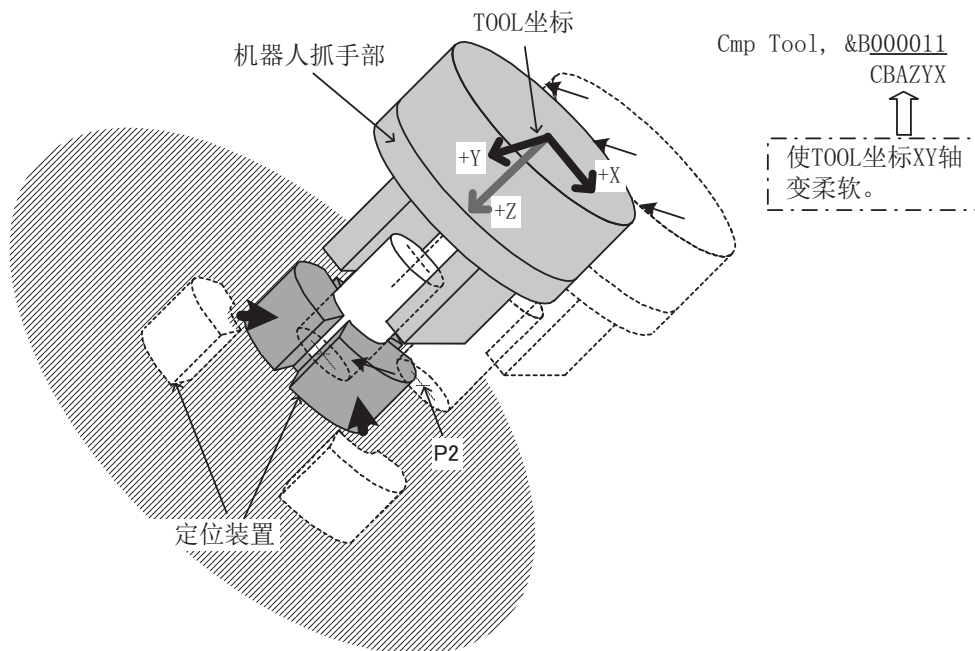


图 4-18：伺服弹性模式使用例

注意

弹性伺服模式会持续到执行 Cmp Off 指令、或电源切断为止。在执行程序编号的变更及 JOG 操作时，请特别注意。

注意

以 Cmp Tool 指令作为弹性伺服模式的状态下执行 JOG 操作的情况下，请在工具 JOG 模式下操作。在其它的 JOG 模式下操作时，由于 JOG 操作和弹性伺服模式下控制的坐标系方向不同，因此机器人会向与想移动的方向不同的方向动作。

注意

在弹性伺服模式的状态示教位置时，请将伺服关闭后再执行。伺服开启的状态下直接执行示教操作时，并不是实际的机器人位置，而是原本的指令位置，机器人会向和示教运行时不同的位置动作，因此请特别注意。

注意

请在机器人静止时开始弹性伺服模式。如果在机器人静止前开始弹性伺服模式，当解除弹性伺服模式时，机器人可能会发生较大振动。

【相关系统状态变量】

M_CmpDst

【相关指令】

Cmp Off、CmpG、Cmp Pos、Cmp Jnt (Comp Joint)

Cmp Off

【功能】

解除机器人柔软控制状态（弹性伺服模式）。

【格式】

Cmp □ Off

【程序示例】

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	' 位置决定用拉锁闭指示
6 Dly 0.5	' 拉锁闭完成等待
7 HOpen 1	' 抓手开
8 Mvs, -100	' 往工具坐标系的 Z- 方向 100mm 退避
9 Mov P_Fbc	' 向反馈位置移动并进行补偿（参照【说明】(3)）
10 Dly 0.5	' 机器人静止等待（参照【说明】(4)）
11 Cmp Off	' 返回到平常状态。
12 Hlt	' 停止

【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下的 JOG 操作时，请通过在程序执行本指令或在示教单元的程序编辑画面直接执行进行设定。
- (3) 执行 Cmp Off 指令时，如果发生 H115n（伺服放大器 通信数据 指令值异常）的情况，在 Cmp Off 之前执行 Mov P_Fbc 能得到改善。
- (4) 如果在机器人静止前开始弹性伺服模式，当执行 Cmp Off 时机器人可能会发生较大振动。请在机器人静止时开始弹性伺服模式。

【相关指令】

CmpG、Cmp Tool、Cmp Pos、Cmp Jnt（Comp Joint）

CmpG

【功能】

指定机器人柔软控制。

【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>], ,

【术语】

<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。

【程序示例】

1 CmpG , , 0.5, , , , ' 只设定 Z 轴。省略轴请空白, 只用逗号记述。

【说明】

- (1) 可以以各轴单位指定柔软度。
- (2) 没有使 Cmp Pos、Cmp Tool 指令为有效状态时, 无法实际变为柔软控制状态。
- (3) 以指令位置和实际的置的偏差为比例, 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。可以使用此变量来确认针脚的插入成功与否等。
- (5) 增益调小时, 以 Cmp Pos、Cmp Tool、Cmp Jnt 指令设定柔软控制状态时, 机器人的位置会下降, 因此, 请逐渐调整为柔软状态, 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时, 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起作用, 机器人会动作。将增益值调大时, 请先依据 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-FRH/RH-CRH 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-FR 系列	0.01, 0.01, 0.01, 0.01, 0.01, 0.01	不可使用

- (8) 弹性伺服模式仅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>]
```

【术语】

- <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。
在插补连续位置，减少速度的降低，轨迹变为沿着内侧转。
 - 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。
在插补连接位置，会暂时降低速度。
- <数值1> 将轨迹变换时的前一个插补结束的最大接近距离以mm为单位指定。
可以使用常数或变量。
- <数值2> 将轨迹变换时的前一个插补结束的最大接近距离以mm为单位指定。
可以使用常数或变量。

【程序示例】

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 300mm 连续动作
9 Cnt 0 ' 将 Cnt（连续动作）设为无效
10 Mov P5 ' 有加减速的执行动作

```

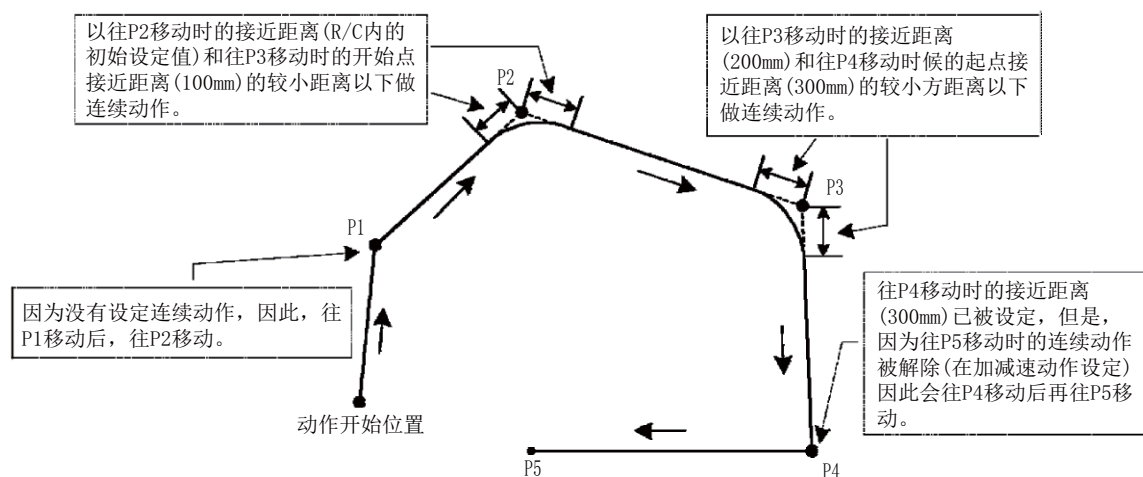


图 4-19: 连续轨迹动作的示例

## ⚠ 注意

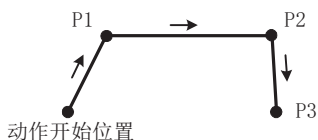
机器人根据当时的指定速度，可能会以不同的轨迹动作。

尤其是转角部分，其内环距离有可能会发生变化，因此，在开始自动运行时最初应先以低速动作，然后一边注意与周边装置之间的干涉，一边逐渐提升速度。

【说明】

- (1) 被 Cnt 1 ~ Cnt 0 围起来的插补（程序示例步号 4 ~ 8）会成为连续动作的对象。
- (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) 图 4-20 如图所示，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。

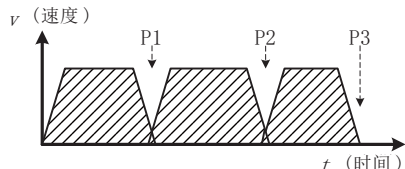
加减速动作模式



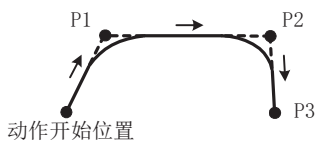
```

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。

```



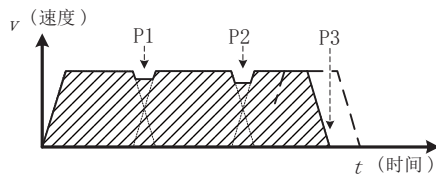
连续动作模式



```

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 4-20：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为往下一个插补的变换位置。此情况下，会通过从目的位置脱离的位置，但是动作时间会变为最短。想要通过最接近目的位置时，请设定接近距离（数值 1、数值 2）。

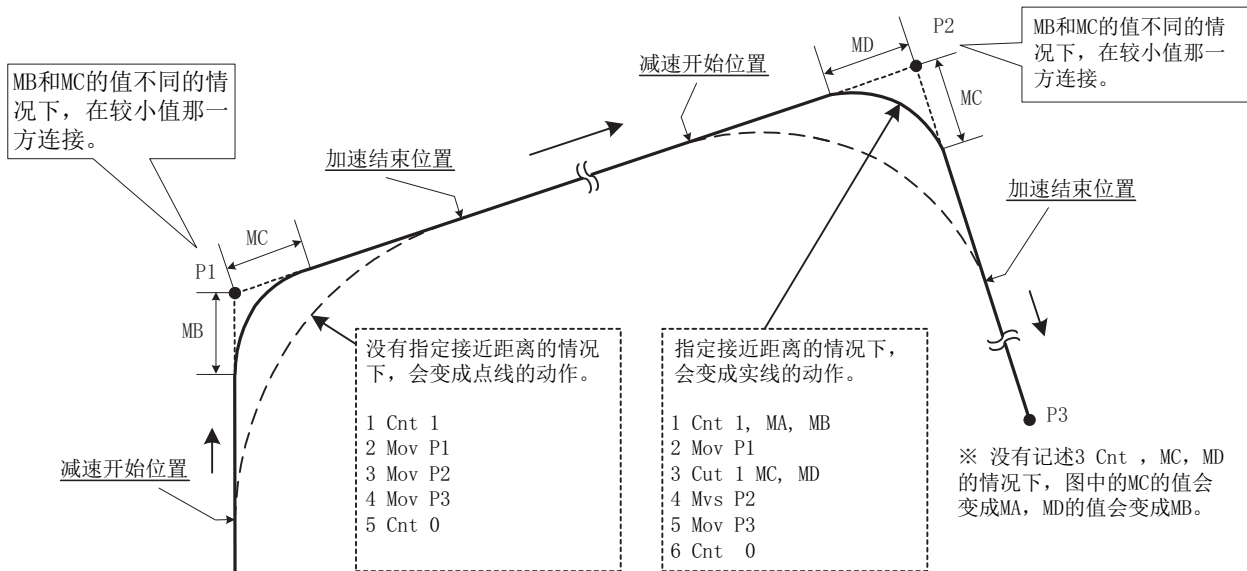


图 4-21：接近距离的设定

- (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 会被设定为和数值 1 相同的值。
- (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为无效。
- (9) 将接近距离（数值 1、数值 2）调小时，会变为比 Cnt 0 的状态动作时间更长。
- (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，也会变为加减速动作。



## 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及手臂与周边机器发生干涉时，迅速停止机器人，且将机器人的工具部分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。

### 【格式】

```
ColChk □ On[, NOErr]/Off
```

### 【术语】

On                    将冲突检知设为有效。检测到冲突时，紧急停止机器人，且发生错误编号1010，关闭伺服。  
Off                    将冲突检知设为无效。  
NOErr                检测到冲突时，不会发生错误。（省略时会发生错误）

### 【程序示例 1】

发生冲突时，发生错误的情况

```
1 ColLvl 80,80,80,80,80,80,, ' 指定冲突检知的容许标准
2 ColChk On ' 将冲突检知设为有效
3 Mov P1
4 Mov P2
5 Dly 0.2 ' 等待到动作完成为止 (Fine 指令也 OK)
6 ColChk Off ' 将冲突检知设为无效
7 Mov P3
```

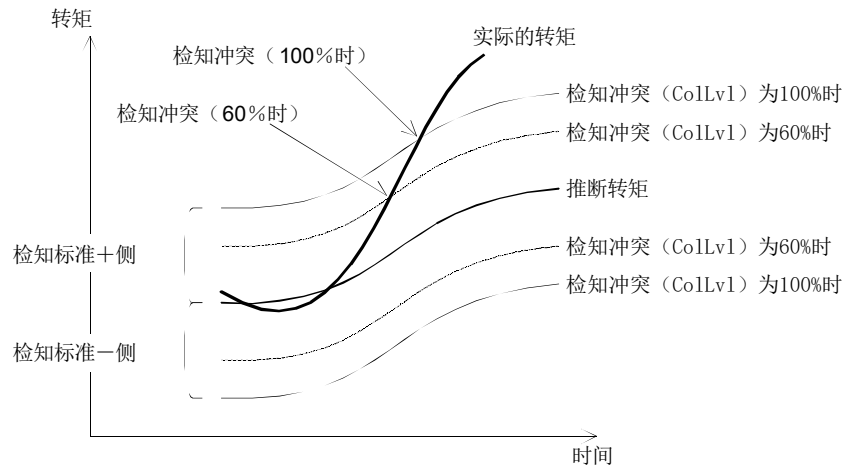
### 【程序示例 2】

发生冲突时，使用中断处理的情况

```
1 Def Act 1,M_ColSts(1)=1 GoTo *HOME,S ' 在检测到冲突时，将执行处理以中断来定义
2 Act 1=1
3 ColChk On,NOErr ' 在错误非发生模式，将冲突检知设为有效
4 Mov P1
5 Mov P2 ' 4 ~ 7 行执行中检测到冲突时，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 ' 将冲突检知设为无效
102 Servo On ' 开启伺服
103 PESC=P_ColDir(1)*(-2) ' 制作退避动作的移动量
104 PDST=P_Fbc(1)+PESC ' 制作退避位置
105 Mvs PDST ' 往退避位置移动
106 Error 9100 ' 让使用者定义的 L 等级错误发生，停止运行
```

### 【说明】

(1) 所谓冲突检知，是以动作指令在移动中，推定对轴一定程度发生转矩，且推定转矩和实际转矩的差异超过容许范围的情况下，判断为冲突，并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 操作中等，没有执行程序时的有效 / 无效状态，则依存参数 COL 的要素 3 的设定。
- (3) 检知标准可以用 ColLvl 指令调整。检知标准的初始值会变为参数 COLLVL 的设定值。
- (4) 以本指令将冲突检知设为有效后，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) 即使以本指令将冲突检知设为无效，以 ColLvl 指令设定的冲突检知标准也会被保持。
- (6) 指定 NOErr（错误非发生模式）时，依据状态变量 M\_ColSts 中断（中断条件为 M\_ColSts(\*)=1 的中断。\* 为机械编号）没有变为有效时，会发生错误（L3950）。请参照【程序示例 2】。此外，在错误非发生模式中，将此中断处理设为无效时，会发生错误（L3960）。
- (7) 在错误非发生模式里检知到冲突时，机器人会关闭伺服且停止，但是不会发生错误，也不会中断运行。但是，在错误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错误的情况下）
- (8) 在无法使用冲突检知功能的机器人上执行 ColChk On 和 ColChk On, NOErr 时，会发生错误（L3970）。ColChk Off 的情况则不会发生错误，会变为无处理。
- (9) 在通过 Cmp 指令的伺服弹性有效中及通过 Torq 指令的转矩限制有效中，无法将冲突检知设为有效。此时，要将冲突检知设为有效时，会发生错误（L3940）。相反的，要在冲突检知有效中将 Cmp 指令及 Torq 指令设为有效时，会发生错误（L3930）。
- (10) 在动作指令之后记述 ColChk Off 时，会有无法在动作的最后停止后近检知到冲突的情况。如程序示例 1 所示，在动作指令和 ColChk Off 指令之间以 Dly 和 Fine 指令将定位完成后，再使 ColChk Off。
- (11) 没有正确设定抓手重量（HNDDATn 参数）和工件重量（WRKDATn 参数）时，会有冲突检知无法正确动作的情况。请正确设定参数后使用。
- (12) 依据本指令将冲突检知功能设为有效时，会由于程序而使执行时间（节拍时间）变长的情况。并不是将全部程序的冲突检知功能设为有效，请使用在只针对有可能对周边机器发生干涉的动作。
- (13) 本功能无法和多机械控制同时使用。
- (14) 检知后若伺服开启，请将参数 COLSERVO 设定为 1，并指定 NOErr（错误非发生模式）。但是，根据受到冲突时的压延量，有可能发生过负载错误。

【相关系统状态变量】

M\_ColSts、J\_ColMxl、P\_ColDir

【相关指令】

ColLvl (ColLevel)

【相关参数】

COL、COLLVL、COLLVLJG、HNDDATn、WRKDATn、COLSERVO

## CollVl (ColLevel)

### 【功能】

在自动运行中设定冲突检知功能的检知标准。

### 【格式】

```
CollVl [<J1 轴 >], [<J2 轴 >], [<J3 轴 >], [<J4 轴 >],
 [<J5 轴 >], [<J6 轴 >], ,
```

### 【术语】

<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。  
初始值会变为参数COLLVL的设定值。

### 【程序示例】

```
1 CollVl 80,80,80,80,80,80,, ' 指定冲突检知的容许标准
2 ColChk On ' 将冲突检知设为有效
3 Mov P1
4 CollVl ,50,50,,,, ' 更改冲突检知的 J2、J3 轴容许标准
5 Mov P2
6 Dly 0.2 ' 将往 P2 到达的冲突检知设为无效
7 ColChk Off ' 将冲突检知设为无效
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、单步跳转操作）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，会变为参数 COLLVL 的设定值。机型不同，参数 COLLVL 的初始值设定不同。
- (4) 值调大时，检知标准（敏感度）会迟钝，调小时，检知标准会提升。
- (5) 检知标准调高时，误检知的发生率会提高，因此请不要调整得过高。  
此外，依据姿势及动作速度，初始值也会有误检知的情况发生。此情况下，请将检知标准调低后使用。
- (6) RT ToolBox3 的示波器功能选择“冲突检知等级参考值”。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 538 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）时，会有冲突检知无法正确动作的情况。请正确设定参数后使用。
- (8) 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。
- (9) 即使在无法使用冲突检知功能的机器人上执行本指令，也会被忽视，不会发生错误。
- (10) 对应 J7、J8 轴，冲突检知不会有作用。
- (11) 即使同一机型也会因机器人的个体差异，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。

### 【相关系统状态变量】

M\_ColSts、J\_ColMxl、P\_ColDir

### 【相关指令】

ColChk (Col Check)

### 【相关参数】

COL、COLLVL、HNDDATn、WRKDATn

※ 详细说明请参照第 538 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标准的采样程序。

## Com On/Com Off/Com Stop (Communication ON/OFF/STOP)

### 【功能】

- Com On : 允许从通信线路的中断。
- Com Off : 禁止从通信线路的中断。
- Com Stop : 暂时停止从通信线路的中断（数据接收）。  
在下一次的 Com On 时立即进行中断跳转。

### 【格式】

```
Com[(<文件编号>)] □ On
```

```
Com[(<文件编号>)] □ Off
```

```
Com[(<文件编号>)] □ Stop
```

### 【术语】

<文件编号> 在通信线路里，记述被分配到1~3的编号。（省略时为1）

### 【程序示例】

请参照第 290 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中断。
- (2) 关于文件编号，请参照第 293 页的“Open”。
- (3) 执行 Com Stop 后，即使有通信也不会发生中断，但是，由于会记忆接收数据和中断，因此依据 Com On 中断再开始时，直接会执行中断处理。

## Const

### 【功能】

对常数（不可写入的变量）进行定义。

### 【格式】

```
Const □ Def □ Inte □ <数值变量名> [= <数值>][, <数值变量名> [= <数值>][, ...]]
Const □ Def □ Long □ <数值变量名> [= <数值>][, <数值变量名> [= <数值>][, ...]]
Const □ Def □ Float □ <数值变量名> [= <数值>][, <数值变量名> [= <数值>][, ...]]
Const □ Def □ Double □ <数值变量名> [= <数值>][, <数值变量名> [= <数值>][, ...]]
```

<数值变量名> 指定变量名。  
 <数值> 指定数值变量设定的值。

```
Const □ Def □ Char □ <字符串变量名> [= <字符串>][, <字符串变量名> [= <字符串>][, ...]]
```

<字符串变量名> 指定变量名。  
 <字符串> 指定字符串变数设定的字符串。

```
Const □ Def □ Jnt □ <关节变量名>[, <关节变量名>[, ...]]
Const □ Def □ Pos □ <直交变量名>[, <直交变量名>[, ...]]
Const □ Def □ Work □ <工件坐标变量名>[, <工件坐标变量名>[, ...]]
```

<关节变量名> 指定变量名。  
 <直交变量名> 指定变量名。  
 <工件坐标变量名> 指定变量名。

```
Const □ 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>])][, ...]]
```

<变量名> 指定数组变量名。  
 <要素数> 以常数指定数组变量的要素个数。

```
Const □ <变量名> = <公式>
```

<变量名> 指定变量名。  
 <公式> 指定代入值、数值运算式。

**【程序示例】**

```
1 Const Def Inte M1 = 10000, M2 = 200 ' 用常数 10000 定义 M1、常数 200 定义 M2。
2 Const Def Float M3 = 3.3 ' 用常数 3.3 定义 M3。
3 Const Def Char Message = "Error" ' 用字符串常数 "Error" 定义 Message。
4 Const Dim cmd$(5) ' 用常数定义 cmd$。
5 Const cmd$(1) = "Cmd01" ' 用字符串常数 "Cmd01" 定义 cmd$(1)。
6 Const Def P1 ' 用常数定义 P1。
7 Const P1 = (100, 0, 0) ' 用直交常数 (100.00, 0.00, 0.00) 定义 P1。
```

**【说明】**

- (1) Const 定义的变量将作为写入禁止（常数）被处理。  
如果对 Const 变量进行写入会发生错误 L4320(00000)。但是，如果用记述的方式替换 Const，则能正常替换（写入）。
- (2) 与变量替换相同，Const 定义时会根据机型进行范围内检查。  
此外，在整数型通过实数进行初始化的情况，会定义四舍五入后的值。
- (3) Const 定义后的变量，即使进行 Clr 指令也不会被清零。
- (4) 使用者定义外部变量也能进行 Const 定义。  
在用户基础程序上设定使用者定义外部变量的值时，需要 Def 指定。  
设定数组或直交 / 关节 / 工件坐标变量的值时，需要通过用户基础程序以外的程序替换。
- (5) 单独变量的情况下，如果在用户基础程序进行 Const 定义，即使不通过参照程序定义，也能使用 Const 变量。
- (6) 数组变量的情况下，如果在用户基础程序进行 Const 定义，即使不通过参照程序进行 Constd 定义，也作为 Const 变量处理。

**【相关指令】**

Def Inte/Def Long/Def Float/Def Double、Def Char (Define Character)、Def Jnt (Define Joint)、  
Def Pos (Define Position)、Def Work、Dim、Static

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 > I < 数值变量 > I < 数值数组 > I < 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, S ' 如果通用输入信号 17 号的输入信号进入开启状态
2 ' 调用 *L100 的子程序。
3 Def Act 2, MFG1 AND MFG2 GoTo *L200 ' 如果 MFG1 和 MFG2 的逻辑变为真，
4 ' 跳转至 *L200。
5 Def Act 3, M_Timer(1) > 10500 GoSub *L300, S ' 经过 10.5 秒后，调用 *L300 的
6 ' 子程序。
7 Act 1=1 ' 将 Act 1 设为有效
8 Act 2=1 ' 将 Act 2 设为有效
9 Dly 10
10 End
11 ' 优先编号 1 的处理
12 *L100: M_Timer(1) = 0 ' 定时器返回到 0。
13 Act 3 = 1 ' 将 Act 3 设为有效。
14 Return 0
15 ' 优先编号 2 的处理
16 *L200: Mov P_Safe
17 End
18 ' 优先编号 3 的处理
19 *L300: M_Timer(1) = 0 ' 定时器返回到 0。
20 Act 3 = 0 ' 将 Act 3 设为无效。
21 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 175 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 指令仅在定义的程序内有效。在通过 CallP 指令（程序间调用）调用的程序（子程序）中或通过 #Include 陈述句指定的程序所定义的函数执行中使用中断处理时，需要在子程序 / 指定的程序上重新定义。
- (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 返回到原本的以单步控制的情况。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移动的过程中，中断条件成立时的执行程序停止的3种概念图如图4-22所示。

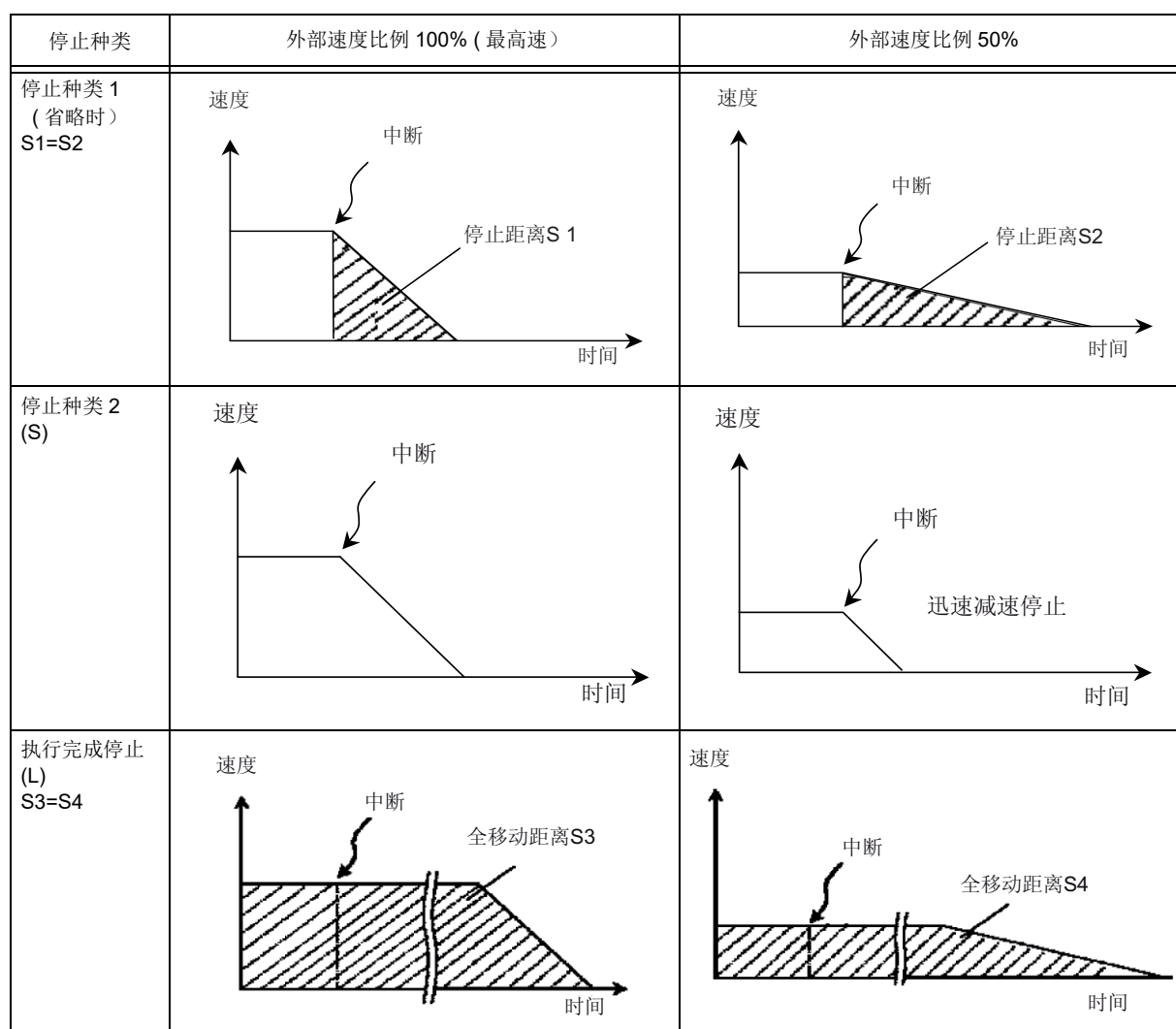


图 4-22：停止时的概念图

【相关指令】

Act

## Def Arch

**【功能】**

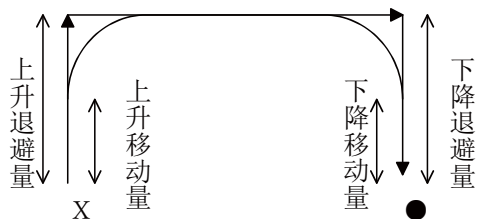
定义 Mva 指令的 Arch 运动动作的弧形形状。

**【格式】**

Def Arch □ < 弧形号码 >, [< 上升移动量 >], [< 下降移动量 >]  
 , [< 上升退避量 >], [< 下降退避量 >], [< 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
- < 上升退避量 >
- < 下降退避量 >
- < 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) | 上升退避量 (mm) | 下降退避量 (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-FR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-FRH/RH-CRH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 , 以在步号 1 所定义的形状，用 Arch 运动动作执行。
- 3 Dly 0.3
- 4 Mva P2, 2 , 没有对应的 Def Arch 的情况下，以参数的初始值动作。
- 5 Dly 0.3

**【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】**

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 MSG = "ABC"         | ’ 在 MSG 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 143 页的“4.3.6 可以在程序内使用的字符种类”。
- (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 152 页的“4.3.24 用户定义外部变量”。

## 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 |
| 工件坐标型 | : W或w |

< 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 ...FNCAME\$ 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。同样, 在过程内定义的用户定义函数无法在定义过程以外使用。在多个过程中通用时, 应进行全局定义 (在过程外定义)。

## Def Gps (Define get position)

### 【功能】

根据高速位置获取功能（GPS 功能）定义取得位置数据时的监视条件。

### 【格式】

```
Def □ Gps □ < 监视编号 >, < 输入编号 >, < 条件 >, [< 机械编号 >]
```

### 【术语】

< 监视编号 > 要监视的监视编号。以常数指定1~8的编号。

< 输入编号 > 指定要监视的输入信号编号。

SKIP输入：801~803

< 条件 > 指定为On或Off。

记录< 输入编号 >中指定的信号为ON或OFF时的机器人的位置数据。

| 设定值 | 条件          |
|-----|-------------|
| On  | 指定的输入信号的上升沿 |
| Off | 指定的输入信号的下降沿 |

< 机械编号 > 指定取得位置数据的机械编号。省略时为1。

### 【示例】

```
1 Def Gps 1, 801, On, 1 ‘ 监视编号1中记录801号的信号为ON时机械1的位置数据
 数据
2 GpsChk On, 1 ‘ 开始监视编号1的条件监视
3 Mvs P1 ‘ 向位置P1移动
4 GpsChk Off, 1 ‘ 结束监视编号1的条件监视
5 M1 = M_Gps(1) ‘ 取得P_Gps1中存储的位置数据的个数
6 If M1 = 0 Then Error 9000 ‘ 若位置数据无法记录, 则发生错误9000
7 Mvs P_Gps1(1) ‘ 向801号首次设为ON时的位置移动。
8 Hlt ‘ 停止
```

### 【说明】

- (1) < 输入编号 > 中指定的输入信号满足 < 条件 > 中指定的条件时，存储 < 机械编号 > 中指定的机器人的当前位置数据。但是，执行 GpsChk On 指令前，未开始本指令中定义条件的监视。
- (2) 已存储的机器人的位置数据在 GpsChk Off 指令执行时存储于 P\_Gps1() ~ P\_Gps8() 中（1 ~ 8 对应监视编号）。
- (3) 存储于 P\_Gps1() ~ P\_Gps8() 中的位置数据在执行对应监视编号的 GpsChk On 指令时清零。
- (4) 使用多任务时，每个插槽可分别根据本指令定义监视条件。定义的内容仅在定义的插槽内有效。全部插槽合计最多可定义包含 Def Map 指令在内的 8 个指令。
- (5) 本指令不支持启动条件设为 ALWAYS、ERROR 的程序。
- (6) < 监视编号 > 中设定了多个条件时，之后执行的条件的设定将覆盖之前的条件的设定。
- (7) 程序复位操作、执行 End 指令及重新接通控制器电源时通过本指令定义的内容清零。
- (8) 在以 CallP 指令呼叫的子程序上也会保持本指令中定义的内容。此外，在子程序上再定义时，将覆盖定义内容。
- (9) 指定与 < 监视编号 > 中 Def Map 指令的 < 监视编号 > 相同的编号时，采用之后执行的指令的监视条件，删除之前执行的指令的监视条件。

(10) 根据 GpsChk On 指令执行监视时，不可根据本指令更改与监视中监视编号对应的定义内容。更改定义内容时，请在结束监视后再通过 GpsChk Off 指令进行更改。

**【相关指令】**

GpsChk (Get position check)、Def Map

**【相关系统状态变量】**

M\_Gps、P\_Gps1 ~ P\_Gps8

Def Inte/Def Long/Def Float/Def Double**【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 WORK4 = 123.468 ' WORK4 变成 123.468000。
```

④ 双精度型实数的定义

```
1 Def Double WORK5'
2 WORK5 = 100/3 ' WORK5 变成 33.333332061767599。
```

**【说明】**

- (1) 变量名的文字数最大为 16 个文字。关于可以使用的字符请参照第 143 页的“4.3.6 可以在程序内使用的字符种类”。
- (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)、双字符 (32bit) 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。
```

④ 将 PORT4 的输入变量，以双字符型分配到输入输出信号编号 16，Mask 信息指定为 16 进制 3FFFFFF。

```
1 Def IO PORT4 = DWord, 16, &H3FFFFFF '
 :
10 PORT4 = 65536 ' 输出信号 32 开启。
```



**【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。关于可以使用的字符请参照第 143 页的“4.3.6 可以在程序内使用的字符种类”。
- (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 0F 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, 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 143 页的“4.3.6 可以在程序内使用的字符种类”。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 152 页的“4.3.24 用户定义外部变量”。

## Def Map

### 【功能】

根据高速位置获取功能（GPS 功能）定义盒内的工件有无判定（映射处理）的监视条件。

### 【格式】

Def □ Map □ < 监视编号 >, < 输入编号 >, < 条件 >, [< 机械编号 >],  
 < 最下段的位置 >, < 最上段的位置 >, < 段数 >, < 感应度 >

### 【术语】

< 监视编号 > 指定要监视的监视编号。以常数指定1~8的编号。

< 输入编号 > 指定要监视的输入信号编号。

SKIP输入：801~803

< 条件 > 指定为On或Off。

记录<输入编号>中指定的信号为ON或OFF时的机器人的位置数据。

| 设定值 | 条件          |
|-----|-------------|
| On  | 指定的输入信号的上升沿 |
| Off | 指定的输入信号的下降沿 |

< 机械编号 > 指定取得位置数据的机械编号。省略时为1。

< 最下段的位置 > / < 最上段的位置 >

指定盒的最下段与最上段的工件检测位置。

要设定的位置需要位于检测时机器人动作的直线轨道上。

将设定为最下段的位置定义为盒 第1段。替换最下段•最上段并进行设定时，将盒最上段定义为第1段。

< 段数 >

指定盒的段数。

根据指定的段数，在平均分割<最下段的位置>与<最上段的位置>的位置上判断各段存在的位置。

设定范围：1~130[段]

< 灵敏度 >

为了提升工件有无的判别精度，指定传感器的感应区域。（参照图4-23）

取得感应区域外的位置时，该位置被撤销，不用于工件有无判别。

设定范围：0.1~100[mm]

（但是，旋转1轴的多机械情况下，设定范围为0.1~100[deg]）

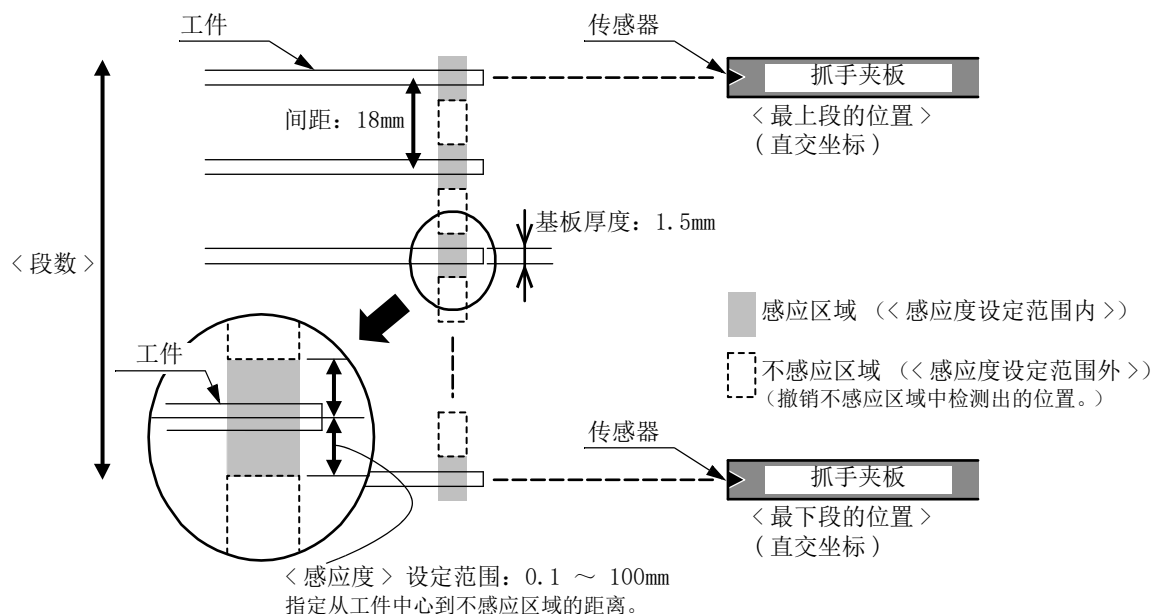


图 4-23: 传感器感应区域

## 【示例】

```
1 Def Map 3, 801, On, 1, PC1, PC2, 20, 10
```

```
2 Mov PM1
```

```
3 GpsChk On, 3
```

```
4 Mvs PM2
```

```
5 GpsChk Off, 3
```

```
6 M1 = M_Gps(3)
```

```
7 If M1 = 0 Then Error 9000
```

```
8 For M2=1 To 20
```

```
9 M_Out(6100+M2)=M_Map3(M2)
```

```
10 Next M2
```

‘监视编号3记录801号为ON时机械1的位置数据, 根据指定的条件执行映射处理。

PC1: 表示盒第1段的位置、PC2: 表示盒最上段的位置、20:20段平均分割、10: 传感器感应度10mm宽

‘移动至映射开始位置

‘开始监视编号3的条件监视。

‘记录801号为ON时的机械1的位置数据。通过M\_Map3(130)表现从该位置数据起工件存在的段数。

‘移动至映射结束位置

‘结束监视编号3的条件监视。算出位置数据中存在工件的段数, 收纳至M\_Map3中

‘取得P\_Gps3中存储的位置数据的个数

‘若位置数据无法记录, 则发生错误9000

‘从输出信号6101号开始按顺序输出映射结果。

‘段数部分循环

## 【说明】

- 执行 GpsChk Off 指令时, 基于 <最下段的位置> 与 <最上段的位置> 中指定的位置、<段数> 及 <灵敏度> 中指定的范围, 计算从取得输入信号时的位置数据起的工件存在的段数。计算结果在 M\_Map1() ~ M\_Map8() 中存储 “1” (有工件) 或 “0” (无工件)。此外, M\_Map1() ~ M\_Map8() 的要素段数从 <最下段位置> 起, 作为 “1” 段输出结果。(M\_Map1(1) 为第 1 段的数据。)
- <输入编号> 中指定的输入信号满足 <条件> 中指定的条件时, 将 <机械编号> 中指定的机器人的当前位置数据、取得的位置数据的个数分别存储于 P\_Gps1() ~ P\_Gps8() 与 M\_Gps(n) 中。但是, 执行 GpsChk On 指令前, 未开始本指令中定义的条件监视。(输入信号有反应的全部位置数据与该个数存储于 P\_Gps1() ~ P\_Gps8()、M\_Gps(n) 中。与 <感应度>、<段数> 的设定无关联。)
- 存储于 P\_Gps1() ~ P\_Gps8()、M\_Map1() ~ M\_Map8()、M\_Gps(n) 中的位置数据在执行对应监视编号的 GpsChk On 指令时清零。

- (4) <最下段的位置>与<最上段的位置>为同一位置时将被视为1段,并计算该段中是否存在工件。
- (5) 使用多任务时,每个插槽可分别根据本指令定义监视条件。定义的内容仅在定义的插槽内有效。全部插槽合计最多可定义包含 Def Gps 指令在内的8个指令。
- (6) 本指令不支持启动条件设为 ALWAYS、ERROR 的程序。
- (7) 相同<监视编号>中设定了多个条件时,之后执行的条件的设定将覆盖之前的条件的设定。
- (8) 程序复位操作、执行 End 指令及重新接通控制器电源时通过本指令定义的内容清零。
- (9) 在以 CallP 指令呼叫的子程序上也会保持本指令中定义的内容。此外,在子程序上再定义时,将覆盖定义内容。
- (10) 指定与<监视编号>中 Def Gps 指令的<监视编号>相同的编号时,采用之后执行的指令的监视条件,删除之前执行的指令的监视条件。
- (11) 根据 GpsChk On 指令执行监视时,不可根据本指令更改与监视中监视编号对应的定义内容。更改定义内容时,请在结束监视后再通过 GpsChk Off 指令进行更改。

**【相关指令】**

GpsChk (Get position check)、Def Gps (Define get position)

**【相关系统状态变量】**

M\_Gps、P\_Gps1 ~ P\_Gps8、M\_Map1 ~ M\_Map8

## Def Plt (Define 码垛)

### 【功能】

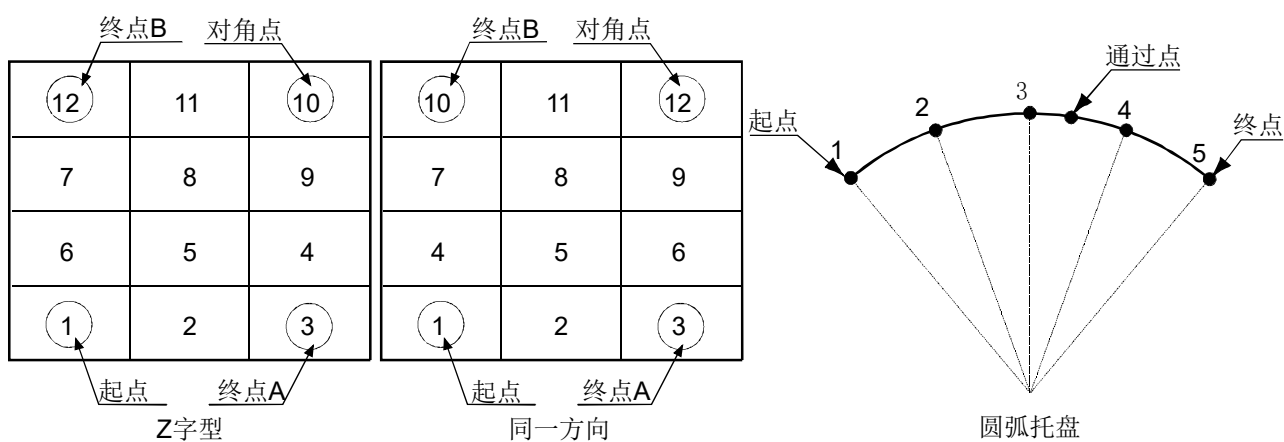
定义码垛。（3点码垛 / 4点码垛 / 圆弧码垛）

### 【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作个数。可以是变量。
- <码垛模版> 码垛的起点和终点间的的工作个数。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 117 页的“4.1.2 托盘运算”。希望使其跨 ±180 度动作时，可通过更改参数执行。详细内容请参照参数 PLTSPEC 的说明。
- (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。  
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 117 页的“4.1.2 托盘运算”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，依据在码垛定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。

程序示例请参照第 117 页的“4.1.2 托盘运算”。

### 【相关指令】

Plt

### ◆◆◆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〈码垛模版〉设定在 11 ~ 13。使在〈起点〉指定的姿势相同。

## Def Pos (Define Position)

### 【功能】

定义直交型位置变量。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」作为直交型位置变量定义       |
| 2 Mov P1                                          | ' 以 P 开始的直交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，为定义直交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。关于可以使用的字符请参照第 143 页的“4.3.6 可以在程序内使用的字符种类”。
- (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。  
详细内容请参照第 152 页的“4.3.24 用户定义外部变量”。

## Def Work

**【功能】**

定义工件坐标变量。变量名开头的字符使用 ‘W’ （或 ‘w’ ）外的字符的时候。

**【格式】**

```
Def □ Work □ <工件坐标变量名>[, <工件坐标变量名>[, ...]]
```

**【术语】**

<工件坐标变量名>            指定变量名。

**【示例】**

1 Def Work Box1            ’ 将 Box1 定义为工件坐标变量。

**【说明】**

- (1) 起始字符以 ‘W’ （或 ‘w’ ）以外的字符定义工件坐标变量时使用。
- (2) 变量名的字符数最多为 32 个字符。关于可以使用的字符请参照第 143 页的 [“4.3.6 可以在程序内使用的字符种类”](#)。
- (3) 在变量名 ‘W’ （或 ‘w’ ）的后面加上下划线 ‘\_’ 登录到用户基本程序后，会变成用户定义外部变量（在程序间共通的变量）。  
利用此变量的情况下，需要以本指令定义。  
详细内容请参照 [“4.3.24 用户定义外部变量”](#)。



## *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> ] ] )<br>[ , <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<br>' 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 ~ 2500 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编号的情况下，执行时会发生 L4370 错误。
- (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。  
详细内容请参照第 152 页的“4.3.24 用户定义外部变量”。

### 【相关指令】

Const

## 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\_Outb(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 > WthIf(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。

## EBRead

### 【功能】

指定视觉传感器的符号名并读取数据。

将从视觉传感器读取的数据存储在指定的变量中。

在 COGNEX 公司产视觉传感器的视觉工具 EasyBuilder 中创建视觉程序（作业）时，应使用本指令，指定符号名并读取数据。

### 【格式】

```
EBRead □ #< 视觉传感器编号 >, [< 符号名 >], < 变量名 1 > [, < 变量名 2 >] . . . [, < 超时 >]
```

### 【用语】

- < 视觉传感器编号 >（不可省略）： 指定控制的视觉传感器的编号。  
设定范围：1 ~ 8
- < 符号名 >（可省略）： 指定存储有视觉传感器的读取数据符号标记的名称。  
省略时，将指定参数 EBRDTAG（初始值为定制格式的符号名“Job.Robot.FormatString”）的设定值。
- < 变量名 >（不可省略）： 指定用于展开从视觉读取的数据的变量。  
可通过逗号（,）分隔记述多个变量。  
可指定数值型变量、位置型变量、字符串型变量。  
指定了位置型变量时，视觉数据设定的成分为 X、Y、C 成分，未设定数据的成分的值 0。
- < 超时 >（省略为 10）： 指定超时时间（单位 s）。  
设定范围：1 ~ 32767 的整数

### 【程序示例】

```
100 If M_NvOpen(1)<>1 Then ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1 ' 与连接至 COM2: 的视觉传感器连接，并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1 ' 与视觉传感器编号 1 连接，等待至登录完成
140 NVLoad #1,"TEST" ' 加载“TEST”程序。
150 NVRun #1,"TEST" ' 加载“TEST”程序。
160 EBRead #1,,MNUM,PVS1,PVS2 ' 读取“Job.Robot.FormatString”符号的数据并保存至变量 MNUM、
 PVS1、PVS2 中。
170
300 NVClose #1 ' 切断与 COM2: 连接的视觉传感器的线路。
```

## 【说明】

- (1) 从指定的视觉传感器的激活的视觉程序中指定符号名并读取数据。
- (2) 从视觉传感器中读取的数据存储在指定的变量中。
- (3) 视觉传感器的数据在以逗号 (,) 分隔的多个值 (字符串) 时, 以逗号 (,) 分隔并罗列指定的变量名时按照记述变量名的顺序存储数据。那时有必要将对象数据与变量的类型设为相同。
- (4) 指定了位置型变量时, 在 X、Y、Z 成分中设定视觉数据, 数据未设定的成分的值设为 0。在 C 成分中设定 Rad 转换的值。
- (5) 指定的变量数比接收数据少时, 仅向指定的变量设定值。
- (6) 指定的变量数比接收数据多时, 不更新比数据数多的部分的变量。
- (7) 省略符号名时, 设定参数 EBRDTAG 的设定值 (出厂设定为“Job.Robot.FormatString”)
- (8) 可通过数值指定超时时间。超时时间内, 从视觉传感器到接收数据为止, 不向下一步移动。但是, 停止机器人程序时, 中断本指令。重启时从中断状态执行处理。
- (9) 多任务中使用本指令时, 使用的任务中需要执行 NVOpen 指令与 NVRun 指令。此外, 在此使用的 <视觉传感器编号>, 应使用通过 NVOpen 指令指定的编号。
- (10) 程序的启动条件不支持『ALWAYS』。
- (11) 执行本指令过程中中断条件成立时, 将立即执行中断处理。中断处理结束后执行处理。
- (12) 由于节拍时间的缩短, 执行 NVRun 指令后实施其他作业, 可在需要的时机执行 EBRead。
- (13) 在 NVRun 指令后记述 EBRead 指令时, 请将参数 NVTRGTMG 的值设为 1。**

**参数 NVTRGTMG 为出场设定 (NVTRGTMG = 2) 时, NVRun 指令不等待视觉识别处理完成, 移动至下一个指令处理。因此继续执行 EBRead 指令时可能取得上一次的识别结果。**

- (14) NVRun 与 EBRead 之间, 程序停止时, 执行了 NVRun 时的结果与执行了 EBRead 的结果可能不同, 请注意。

## &lt; 变量中设定的值 &gt;

根据 EBRead 指令的执行在变量中代入的值如下所示。

- (1) 指定符号 (Pattern\_1.Number\_Found) 的内容 10
  - 执行 EBRead #1, "Pattern\_1.Number\_Found", MNUM 时的设定值  
→ MNUM=10
  - 执行 EBRead #1, "Pattern\_1.Number\_Found", CNUM 时的设定值  
→ CMNUM="10"
- (2) 指定符号 (Job.Robot.FormatString) 的内容 2, 125.75, 130.5, -117.2, 55.1, 0, 16.2
  - 执行 EBRead #1, , MNUM, PVS1, PVS2 时的设定值  
→ MNUM=2  
PVS1. X=125.75 PVS1. Y=130.5 PVS1. C=-117.2  
PVS2. X=55.1 PVS2. Y=0, PVS2. C=16.2  
※ 视觉数据未设定的成分 (X, Y, C 成分以外) 的值为 0
  - 执行 EBRead #1, , MNUM, MX1, MY1, MC1, MX2, MY2, MC2 时的设定值  
→ MNUM=2  
MX1=125.75 MY1=130.5 MC1=-117.2  
MX2=55.1 MY2=0 MC2=16.2
  - 执行 EBRead #1, , CNUM, CX1, CY1, CC1, CX2, CY2, CC2 时的设定值  
→ CNUM="2"  
CX1="125.75" CY1="130.5" CC1="-117.2"  
CX2="55.1" CY2="0" CC2="16.2"
- (3) 指定符号 (Job.Robot.FormatString) 的内容 2, 125.75, 130.5
  - 执行 EBRead #1, , MNUM, PVS1 时的设定值  
→ MNUM=2  
PVS1. X=125.75 PVS1. Y=130.5  
※ 视觉数据未设定的成分 (X, Y 成分以外) 的值为 0

**【错误】**

- (1) 变量的数据类型不同时，将发生“输入的指令语句的语法有误”错误。
- (2) 指令的变量的个数异常（多/少）时，将发生“变量的个数不正确。”错误。
- (3) <视觉传感器编号>为1~8以外时，将发生“超出变量值范围”错误。
- (4) 以<视觉传感器编号>中设定的编号未执行NVOpen指令时，将发生“未执行NVOPEN指令”错误。
- (5) 接收的字符串数据与将其代入的变量的类型不同时，将发生“EBREAD接收数据格式异常”错误。
- (6) <超时>为1~32767以外时，将发生“超出变量值范围”错误
- (7) <超时>中指定的时间以内，或省略<超时>并在初始值的10s以内从视觉传感器无应答时，将发生“视觉传感器响应超时”错误。
- (8) 执行本指令过程中通信线路被切断时，将发生“通信异常”错误，并关闭机器人控制器侧的线路。
- (9) 不存在于指定的符号名为激活的视觉程序中时，将发生“指定的符号名不正确”错误。
- (10) 请将指定的变量的数量最多设为31个（识别个数+坐标值（X、Y、Z）×10个）。指定超过该数量时，根据指令的记述将发生“输入的指令语句的语法有误”错误。
- (11) <视觉程序名>超过15个字符的字符数时，将发生“视觉程序名异常”错误。
- (12) <视觉程序名>中使用了“0”~“9”、“A”~“Z”、“-”、“\_”以外（也包含小写字母的字母）时，将发生“视觉程序名异常”错误。
- (13) <视觉程序名>中指定的程序不存在于视觉传感器内时，将发生“不存在视觉程序”错误。
- (14) <视觉程序名>中指定的程序若未通过NVRun指令启动，将发生“视觉程序名异常”错误。
- (15) <识别个数单元>与<开始单元>、<结束单元>的指定中，字母为“A”~“Z”以外或数值为“0”~“399”以外时，将发生“变量值超出范围”错误。
- (16) <识别个数单元>中指定的单元中没有值时，将发生“识别个数单元中指定的值不正确”错误。
- (17) <开始单元>与<结束单元>逆转时，将发生“指定的单元值超出范围”错误。
- (18) <开始单元>与<结束单元>中指定的单元中包含的数据数超过90个时，将发生“指定的单元值超出范围”错误。
- (19) <开始单元>与<结束单元>中指定的范围超出行30、列10时，将发生“指定的单元值超出范围”错误。
- (20) <类型>为“0”~“7”以外时，将发生“变量值超出范围”错误。

**⚠ 注意**

请设定参数NVTRGTMG。

在NVRun指令后记述EBRead指令时，请将参数NVTRGTMG的值设为1。

参数NVTRGTMG为出场设定（NVTRGTMG = 2）时，NVRun指令不等待视觉识别处理完成，移动至下一个指令处理。因此继续执行EBRead指令时可能获取上一次的识别结果。

## EBWrite

### 【功能】

指定视觉传感器的符号名并写入数据。

使用 COGNEX 公司产视觉传感器的视觉工具 EasyBuilder 创建了视觉程序（作业）后，可以使用本指令，在以符号名指定的单元中写入数据。

### 【格式】

```
EBWrite □ #< 视觉传感器编号 >, [< 符号名 >], < 写入数据 > [, < 超时 >]
```

### 【用语】

- < 视觉传感器编号 >:                   以数值常数指定所控制的视觉传感器的编号。  
                                          设定范围：1 ~ 8
- < 符号名 >:                            指定要写入数据的单元格标识符号的名称。  
                                          省略的情况下，指定为参数 EBWRTAG 的设定值。
- < 写入数据 >:                         指定视觉传感器中写入的数据。  
                                          数值常数、数值变量、字符串常数、字符串变量、位置成分数据、关节成分数据、算式、字符串式可以使用。
- < 超时 >:                             以数值常数指定超时时间（单位：秒）。  
                                          省略的情况下指定为 10 秒。  
                                          设定范围：1 ~ 32767 的整数值

### 【程序示例】

```
1 If M_NvOpen(1)<>1 Then ' 视觉传感器编号 1 登录未完成时
2 NVOpen "COM2:" As #1 ' 与连接至 COM2: 的视觉传感器连接，并将编号设为 1
3 Wait M_NvOpen(1) = 1 ' 等待视觉传感器编号 1 登录完成
4 EndIf
5 NVOpen #1, "TEST" ' 加载 "TEST" 程序（作业）
6 EBWrite #1, "Sample.Float", 5 ' 将 "Sample.Float" 符号的数据改写为 5
7 EBWrite #1, "Sample.String", "Test" ' 将 "Sample.String" 符号的数据改写为 "Test"
8 NVRun #1, "TEST" ' 启动 "TEST" 程序（作业）
 :
 :
20 End
```

### 【说明】

- (1) 对指定的视觉传感器的激活的视觉程序（作业），在以符号名指定的单元格中写入数据。
- (2) 对 < 视觉传感器编号 > 中指定的视觉传感器，未执行 NVOpen 指令时将发生错误（L. 3141）。
- (3) 省略 < 符号名 > 时，设定为参数 EBWRTAG 的设定值（出厂设定为“ ”（NULL））。
- (4) 指定的 < 符号名 > 在激活的视觉程序中不存在时，将发生错误（L. 8637）。
- (5) 根据 < 写入数据 > 的数据类型的不同，视觉传感器的单元格中写入的数据类型会发生变化。（即使指定了双精度实数时，也是转换为了单精度实数的值。）

| < 写入数据 > | 单元格中写入的数据类型  |
|----------|--------------|
| 数值型（整数）  | 整数（Int）      |
| 数值型（实数）  | 单精度实数（Float） |
| 字符串型     | 字符串（String）  |

- (6) 根据 < 写入数据 > 的数据类型和以 < 符号名 > 指定的视觉程序的单元格的值类型的组合情况，处理方式如下表所示。

| < 写入数据 > | 单元格的值类型    | 处理            |
|----------|------------|---------------|
| 数值型（整数）  | 布尔值编辑控制    | 单元格值更新（整数）    |
|          | 整数编辑控制     | 单元格值更新（整数）    |
|          | 浮动小数点数编辑控制 | 单元格值更新（单精度实数） |
|          | 文本编辑控制     | 执行错误（L. 8637） |

| < 写入数据 > | 单元格的值类型                                   | 处理                                                                             |
|----------|-------------------------------------------|--------------------------------------------------------------------------------|
| 数值型 (实数) | 布尔值编辑控制<br>整数编辑控制<br>浮动小数点数编辑控制<br>文本编辑控制 | 单元格值更新 (整数、小数点以下舍去)<br>单元格值更新 (整数、小数点以下舍去)<br>单元格值更新 (单精度实数)<br>执行错误 (L. 8637) |
| 字符串型     | 布尔值编辑控制<br>整数编辑控制<br>浮动小数点数编辑控制<br>文本编辑控制 | 执行错误 (L. 8637)<br>执行错误 (L. 8637)<br>执行错误 (L. 8637)<br>单元格值更新 (字符串)             |

- (7) 可通过数值常数指定超时时间。超时时间内，在接收到来自视觉传感器的写入结果的数据之前，不向下一步移动。
- (8) 停止执行机器人程序时，本指令的处理将中断。重新执行程序则会从中断状态开始继续进行处理。
- (9) 在多任务中使用本指令时，使用的任务插槽中需要执行 NVOpen 指令。此外，< 视觉传感器编号 > 应使用通过 NVOpen 指令指定的编号。
- (10) 任务插槽的启动条件为 『ALWAYS』 时无法使用。
- (11) 执行本指令过程中中断条件成立时，将立即执行中断处理。

## 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 724 页的“7.3 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 WthIf 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 2 进行圆形插补移动的同时，将输出信号 17 置为 ON  
 3 EMvc 2, P1, P2, P3 Wth M\_Out(17)=1
- ④ 按照 P3、(P1t 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, (P1t 1, 5), P4 WthIf 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 510 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 724 页的“7.3 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 WthIf 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 2 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 (P1t 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, (P1t 1, 5), P4 WthIf 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 510 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 724 页的“7.3 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WthIf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 WthIf 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 510 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 724 页的“7.3 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 WthIf 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 WthIf 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 510 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, 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 724 页的“7.3 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】

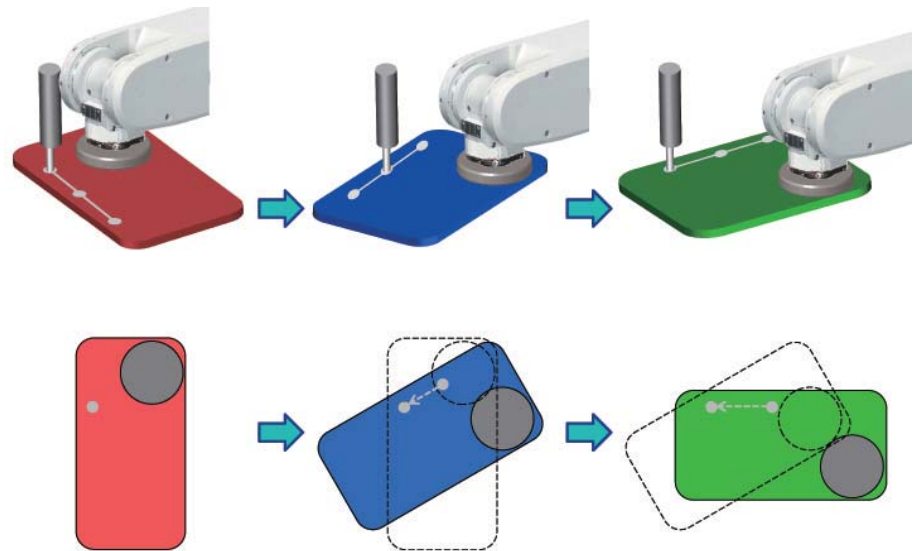
- <工件坐标号码> 指定1~8的工件坐标号码。
- <移动目的位置> 指定插补动作的最终位置。以位置型的变量与常数、逻辑/算术式与函数或关节变量进行记述。（输入机器人控制点的位置数据。）
- <常数1> 姿势旋转的指定。抄近道动作 = 0。省略时为0。
- <常数2> 姿势插补方法的指定。等效旋转/三轴直交= 0/1。省略时为0。
- <附带条件> 对Wth节或WthIf节进行记述。

### 【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 (P1t 1,5) 沿着工件坐标 3 进行直线插补移动  
 1 EMvs 3,(P1t 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 WthIf 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) 在 MELFA-BASIC V/VI 下可使用。

- (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 Ovrld 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 Oad1 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 655 页的“7.2 关于样条插补”。）

### 【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>) × (Ovrld 指令) × (操作面板的倍率修调)  
不使用 Spd 指令与 JOvrld 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 Oad1 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 Ovrld 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 响，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 L3287 (起动条件为 ERR、ALW 的情况下无法使用此指令。)
- (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。

## 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、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)、CallP

## 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<br>(高标准报警) | 停止程序的执行并切断伺服电源。<br>以报警复位的输入来解除报警状态。 |
| 9100 ~ 9199<br>(低标准报警) | 停止程序的执行。<br>以报警复位的输入来解除报警状态。        |
| 9200 ~ 9299<br>(警告)    | 继续程序的执行。<br>以报警复位的输入来解除报警状态。        |

### 【关连参数】

UER1 ~ 20

## Exit

### 【功能】

脱离 Function 过程。

### 【格式】

|                 |
|-----------------|
| Exit □ Function |
|-----------------|

### 【示例】

```
1 Function V Func
2 ' :
3 Exit Function ' 脱离函数的处理。
4 FEnd
```

### 【说明】

(1) 脱离函数的处理 (Function 过程)。

### 【相关指令】

Function... FEnd

## 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。<br>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】

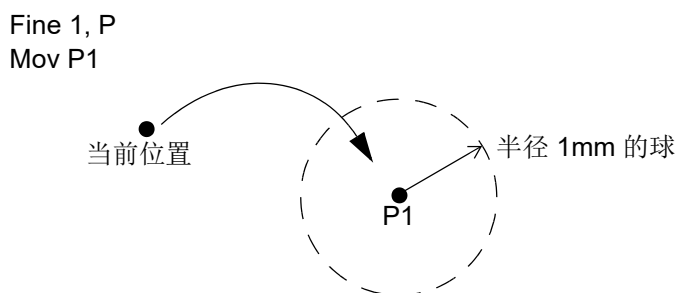
< 直线距离 >      定位直线距离[mm]以变量或常数指定。  
指定0的话会变成无效。初始值会被设定为0。  
可指定的最小值为0.001。

### 【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 4-24: 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 MSUM 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 ' 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 ' 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】

[CallP](#)

## Function... FEnd

### 【功能】

对 Function 过程（函数）进行定义。

### 【格式】

```
Function □ <返回值的类型> □ <过程名> ([<假设自变量>[, <假设自变量>]...])
 <函数的处理>
 <过程名> = <返回值>
Exit □ Function
FEnd
```

### 【术语】

|          |                                                                                                                 |
|----------|-----------------------------------------------------------------------------------------------------------------|
| <返回值的类型> | 指定返回值的类型。<br>R: 直交型<br>J: 关节型<br>W: 工件坐标型<br>M: 数值型(通过类型后置符号, 指定数值型。省略时, 为单精度实数型。)<br>G: 字符串型<br>V: 无返回值(void型) |
| <过程名>    | 任意字符串(半角英文数字、最多32个字符)                                                                                           |
| <假设自变量>  | 自变量名与对应“4.3.8数据类型”的5个数据类型的变量名相同。<br>交接方法为在自变量名的前面加上“ByVal”(交接值、可省略)/“ByRef”(参照交接)关键词。                           |
| <函数的处理>  | 记述函数的处理。                                                                                                        |
| <返回值>    | 通过指定的类型记述函数的返回值。                                                                                                |

### 【示例】

```
1 Function Main
2 Dim Mary(10)
3 For Idx=1 To 10 Step 1 ' 在 Mary 中存储值的处理
4 Mary(Idx) = Idx
5 Next
6 Msum = Sum(Mary, 10) ' 求出 Mary 的合计值。
7 FEnd
8 ' -----
9 ' 求出第 1 个自变量中指定的 1 次元排列的合计。
10 ' 返回值：合计值
11 ' 第 1 个自变量：1 次元排列
12 ' 第 2 个自变量：要素数
13 Function M Sum(ByRef Mary(), ByVal Melem)
14 Msum = 0
15 For Idx=1 To Melem Step 1
16 Msum = Msum + Mary(Idx)
17 Next
18 Sum = Msum ' 返回 Mary 的合计值。
19 Exit Function
20 FEnd
```

### 【说明】

- (1) Function 过程最多可定义 256 个。
- (2) 可省略返回值的指定。省略返回值的指定时为 void 型。
- (3) 假设自变量最多可定义 16 个。

- (4) 定义过程时，需要对程序文件中作为入口点的 Function Main 过程进行定义。（不存在 Function Main 过程的程序文件按惯例记述程序。）
- (5) 通过 CallP 指令调用存在 Function Main 过程的程序时，CallP 指令的自变量转为过程的假设自变量（不需要 FPrm 指令。）  
但是，定义假设自变量附带的 Function Main 过程时，不可定义没有假设自变量的 Function Main 过程。
- (6) 存在 Function Main 过程的程序中，指令语句（除了通过预处理程序可执行的指令）务必记述在过程内。  
※ 通过预处理程序可执行的指令如下所示。
- #Include 叙述
  - 定义指令 注1)
  - 条件分支 / 反复指令
  - 算数运算 / 比较运算
  - Tool/Base 指令
- 注1) 通过过程以外定义的指令（Def Act 指令等）无法记述调用子程序（GoSub...Return 语句）及 GoTo 语句的处理。
- (7) 通过不存在 Function 过程的程序定义的变量全部为程序内的全局变量。
- (8) 调用 Function 过程时，<过程名>的后面务必加上括号（）。
- (9) 在自变量中指定排列时，在自变量名中加上“（<要素数>[,<要素数>[,<要素数>]]）”进行声明。  
可通过省略<要素数>，交接要素数与次元数未确定的排列。（※ 最多3次元）  
关于指定排列时的假设自变量与实际自变量的对应，请参照下表。

|       |                | 实际自变量 |                 |                |
|-------|----------------|-------|-----------------|----------------|
|       |                | 非排列   | 排列<br>(要素数确定)   | 排列<br>(要素数未确定) |
| 假设自变量 | 非排列            | OK    | NG              | NG             |
|       | 排列<br>(要素数确定)  | NG    | 仅限要素数与次元数相同时 OK | NG             |
|       | 排列<br>(要素数未确定) | NG    | OK              | OK             |

- (10) 在自变量中指定排列并省略<要素数>时，此后次元的<要素数>也未确定。
- (11) 在实际自变量中指定成分数据时，务必转为值。
- (12) 不可在自变量中指定状态变量。
- (13) 函数内已经声明・定义的变量只在函数内为有效的局部变量。
- (14) 与局部变量同名的变量在函数外已经声明・定义时，局部变量将优先。
- (15) 静态变量可示教。  
按照以下方式显示在变量编辑器中。  
“<过程名>.:<变量名>”
- (16) 在函数完成后撤销局部变量值。
- (17) 函数完成后仍保持局部变量值时，变量声明时在变量名前加上 Static 关键词变为 static 变量（静态变量）。
- (18) 未处理返回值时，返回返回值类型的初始值。  
各数据类型的初始值如下所示。
- 直交型 / 工件坐标型：“(0, 0, 0, 0, 0, 0, 0, 0, 0)(, )”
  - 关节型：“(0, 0, 0, 0, 0, 0, 0, 0, 0)”
  - 数值型：0
  - 字符串型：“”
- (19) 通过 Exit Function，脱离函数的处理。
- (20) 以下值继承在已调用函数（反向返回调用元时）中也设定的值。  
Cnt, Fine, MvTnue, Tool, Base, Accel, Spd, Ovr, JOvr
- (21) 同时使用其他程序读取指令（CallP、XLoad、XRun、#Include）的情况下，应将参数 FUNCSPEC 设定为 1。

#### 【关连参数】

FUNCSPEC

## 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 Re1M 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 Re1M ' 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 Re1M ' 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 Re1M 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<br>※ 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<br>JtoP、PtoJ ..... Ver.N8/P8 ~<br>Inv..... Ver.R3a/S3a(特殊 ES Ver.R3dZ)<br>Fram..... Ver.R3a/S3a(特殊 ES Ver.R3dZ)                                                                              |
| 机器人状态变量 | M_RSpd、M_ESpd                                                                                                                                                                                                                                                      |

- (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。

## *GpsChk (Get position check)*

### 【功能】

高速位置获取功能（GPS 功能）中，通过 Def Gps 指令或 Def Map 指令开始 / 结束定义的条件监视。  
执行本指令前，需要通过 Def Gps 指令或 Def Map 指令定义监视条件。

### 【格式】

|                           |
|---------------------------|
| GpsChk □ <On/Off>, <监视编号> |
|---------------------------|

### 【术语】

<On/Off>      On: 通过<监视编号>中指定的Def Gps指令或Def Map指令开始定义的条件监视。  
将相当于M\_Gps(n)、P\_Gps1()~P\_Gps8()、M\_Map1()~M\_Map8()中指定的监视编号的数据清零。  
Off: 结束监视中的输入信号的监视。  
生成对应<监视编号>的M\_Gps(n)、P\_Gps1()~P\_Gps8()、M\_Map1()~M\_Map8()中的数据。

<监视编号>      指定通过Def Gps指令或Def Map指令指定的1~8的监视编号。

### 【示例】

|                       |                            |
|-----------------------|----------------------------|
| 1 Def Gps 7,852,Off,1 | ‘监视编号7记录852号为OFF时机械1的位置数据。 |
| 2 GpsChk On, 7        | ‘开始监视编号7的条件监视              |
| 3 Mvs P1              | ‘向位置P1移动                   |
| 4 GpsChk Off, 7       | ‘结束监视编号7的条件监视              |

### 【说明】

- 通过 Def Gps 指令或 Def Map 指令开始、结束已定义的输入信号的监视。
- 将执行 GpsChk On 指令时 <监视编号> 中指定的 P\_Gps1() ~ P\_Gps8()、M\_Gps() 数据清零。
- 生成执行 GpsChk Off 指令时 <监视编号> 中指定的 P\_Gps1() ~ P\_Gps8()、M\_Gps(n) 数据。使用通过高速位置获取功能（GPS 功能）取得的数据前，务必执行 GpsChk Off 指令。
- 本指令不支持启动条件设为 ALWAYS、ERROR 的程序。

### 【相关指令】

Def Gps (Define get position)、Def Map

### 【相关系统状态变量】

M\_Gps、P\_Gps1 ~ P\_Gps8、M\_Map1 ~ M\_Map8

## 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 WthIf 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 (WthIf 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。使用多机器时，对通过 GetM 指令取得机器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 >

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。

**【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 510 页的“5.10 中断时的 JOG 前进后的自动回归设定”及第 513 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

CR860 控制器时，抓手输入信号 / 输出信号中的 900 号分别对应 764 号。

**【相关指令】**

LoadSet (Load Set)、Oadl (Optimal Acceleration)

**【关连参数】**

HANDTYPE、HANDINIT

请参照第 510 页的“5.10 中断时的 JOG 前进后的自动回归设定”及第 513 页的“5.13 关于抓手初始状态”。

If...Then...ElseIf...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
 :
[Elseif □ < 公式 > □ 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 ' M1 比 10 大的情况下，跳转到标准 L100
11 If M1>10 Then GoTo *L20 Else GoTo *L30 ' 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 .. ElseIf .. Else .. EndIf 可以在复数行跨行记述。
- (3) Else 可以省略。
- (4) If .. Then .. ElseIf .. Else .. EndIf 的情况下请务必使用 EndIf 记述。
- (5) ElseIf 可进行多行记述。
- (6) 以 If .. Then .. ElseIf .. 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。
- (7) If .. Then .. ElseIf .. Else .. EndIf 的情况下，在 Then 或 Else 里，可以记述 If .. Then .. ElseIf .. Else .. EndIf 。（回路可到 8 段）
- (8) 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
- (9) 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。
- (10) 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。
- (11) 在 ElseIf 前记述 Else 时将发生错误。

## Include

### 【功能】

读取指定的程序。

### 【格式】

```
#Include □ “<程序名>”
```

### 【术语】

<程序名>                      指定要读取的程序名。

### 【示例】

```

1 #Include “LIB1” ’ 读取程序。
2 ’
3 Function Main
4 Mvs PData1 ’ 利用已读取的程序的通用变量。
5 Func1() ’ 调用已读取的程序的函数。
6 FEnd

1 ’ LIB1.MB6
2 Def Pos PData1, PData2
3 Function V Func1
4 Mvs P1
5 Mvs P2
6 Mvs P3
7 FEnd

```

### 【说明】

- (1) #Include 叙述包含多重 Include（通过 #Include 叙述在读取程序内进一步声明 #Include 叙述时）最多可声明 64 个。
- (2) 已指定的程序读取到记述了 #Include 叙述的步骤中。
- (3) 在 Function 过程内声明 #Include 叙述时，执行时发生错误。
- (4) 多次指定同一个文件时，不读取第 2 次以后的文件。  
(在通过 #Include 叙述指定的程序目标中也包含已经读取的程序。)
- (5) 已指定读取源（已记述 #Include 叙述）程序时，将被无视。
- (6) 读取目标（已通过 #Include 叙述指定）程序中存在同名的变量 / 函数时，读取源（已记述 #Include 叙述）程序将会优先。
- (7) 读取目标程序间存在同名的变量 / 函数时，顺序靠后的程序将会优先（已记述 #Include 叙述）。
- (8) 程序间的同名函数中函数的签名不同时，作为重载函数进行处理。

## Input

### 【功能】

从文件（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

（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】

Open、Close、Print

## JOvrd

### 【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  $\pm 360$  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-FRH/RH-CRH 系列: J4 轴 (优先轴) |
| RV-FR 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，调节速度和加减速度比例，请在不会发生报警的条件下使用。
- (14) 请勿在执行 JRC 指令的过程中切断控制器的电源。原点数据异常，可能发生错误 C1761 (机器人本体内的原点数据不正确)。此时需要重新进行原点设定。

#### 【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 468 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。<br>将指定重量·大小的抓手条件参数 (HNDDATO~8) 的号码以常数或变量指定。<br>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。<br>将指定重量·大小的工件条件参数 (WRKDATO~8) 的号码以常数或变量指定。<br>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 (WRKDATO) 的条件
5 Mov P2
6 Oadl Off

```

### 【说明】

- 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。
- 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。）
- 抓手条件以参数 (HNDDATO~8) 指定重量、大小 (X, Y, Z)、重心位置 (X, Y, Z)。
- 工件条件以参数 (WRKDATO~8) 指定重量、大小 (X, Y, Z)、重心位置 (X, Y, Z)。
- 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 HNDDATO、WRKDATO。
- 最佳加减速用的详细设定请参照第 523 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 547 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 HNDDATO ~ 8、WRKDATO ~ 8、HNDHOLD0 ~ 8 请参照第 523 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 468 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 547 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 (P1t 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)=1

### 【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  $\pm 180$  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话, 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 510 页的“5.10 中断时的 JOG 前进后的自动回归设定”)。

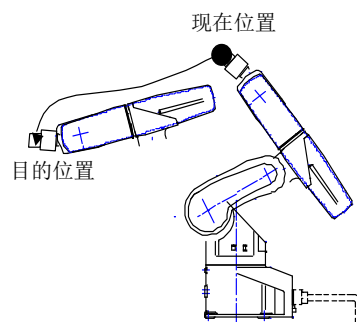


图 4-25: 关节插补的动作轨迹示例

## 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】

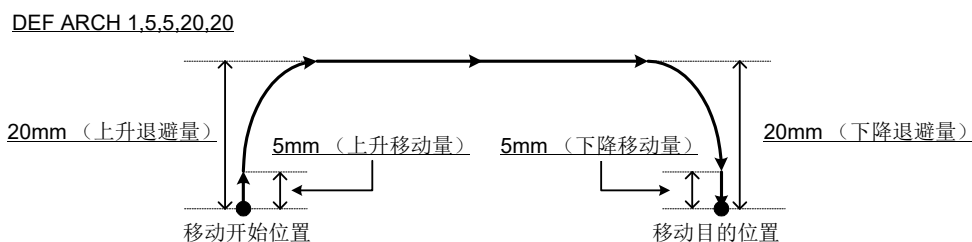
<移动目的位置> 插补动作的最终位置。以位置型的变量和常数或关节变量记述。  
<弧形号码> Def Arch (1~4)。

### 【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 ' 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 205 页的“Def Arch”。
- (3) 因为插补形式、插补类型等也是以 Def Arch 指令定义的，因此请参照第 205 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 510 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。

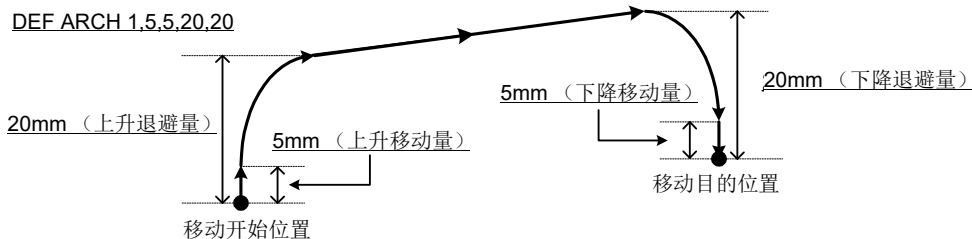


图 4-26：弧形插补的动作轨迹示例（从水平方向看的情况）

## 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。

### 【相关指令】

[Def Arch](#)、[Accel](#)、[Ovr](#)d

## Mva2 (Move Arch2)

### 【功能】

以圆弧动作（弧形插补）从当前位置移动到目的位置。

注）本指令限定为对应机型。请参照【可使用的机器人机型】。

### 【格式】

Mva2 <移动目的位置> [, <下降量> [, <接近距离 1> [, <接近距离 2>  
[, <定位距离> [, <整定时间> ]]]]]

### 【术语】

|          |                                                                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <移动目的位置> | 插补动作的结束位置。以位置型的变量和常数、或关节变量进行记述。                                                                                                                 |
| <下降量>    | 指定 Z 轴动作限制的下降量。单位：[mm]<br>设定范围：0 以下的实数                                                                                                          |
| <接近距离 1> | 指定或省略了 0 时，上升至 Z 轴动作限制位置。<br>以常数或变量可以指定第 1 转角的接近距离。单位：[mm]<br>设定范围：-1/0 以上的实数<br>指定了 0 时，至上升位置的动作完成后，进行下一个动作。<br>指定了 -1 时，变为与 Cnt1 相同的动作。       |
| <接近距离 2> | 省略时，使用参数“MVACNT1”的第 1 要素的值。<br>以常数或变量可以指定第 2 转角的接近距离。单位：[mm]<br>设定范围：-1/0 以上的实数<br>指定了 0 时，至目的位置上方位置的动作完成后，进行下一个动作。<br>指定了 -1 时，变为与 Cnt1 相同的动作。 |
| <定位距离>   | 省略时，使用参数“MVACNT1”的第 2 要素的值。<br>将至目的位置的定位完成条件以直线距离进行指定。单位：[mm]<br>设定范围：0 以上的实数（可指定的最小值为 0.001。）<br>指定了 0 时，定位为无效。                                |
| <整定时间>   | 省略时，使用参数“MVACNT1”的第 3 要素的值。<br>指定目的位置到达完成后的整定时间。单位：[秒]<br>设定范围：0 以上的实数<br>省略时，使用参数“MVACNT1”的第 4 要素的值。                                           |

### 【程序示例】

|                              |                                                     |
|------------------------------|-----------------------------------------------------|
| 1 Mov P0                     | ' 移动到圆弧动作的开始位置。                                     |
| 2 Mva2 P1                    | ' 上升至 Z 轴的动作限制位置后，以圆弧向 P1 移动。                       |
| 3 Mva2 P2, -20               | ' 上升至 Z 轴的动作限制 -20mm 位置后，以圆弧向 P2 移动。                |
| 4 Mva2 P3, 0, 20, 20, 0.1, 0 | ' 上升至 Z 轴的动作限制位置后，以圆弧向 P3 移动。<br>' 目的位置的定位距离为 0.1mm |

### 【说明】

- (1) 从当前位置向 Z 轴方向上升，接着移动至目的位置的上方为止，最后下降移动至目的位置。以一个指令执行圆弧动作。
- (2) <下降量>、<接近距离 1>、<接近距离 2>、<定位距离>、<沉降时间> 可省略。
- (3) 省略了 <接近距离 1>、<接近距离 2>、<定位距离>、<沉降时间> 时使用参数“MVACNT1”的值。详细内容请参照参数“MVACNT1”的说明。
- (4) Mva2 指令的插补形式为关节插补、就近、等价旋转。
- (5) Mva2 指令执行时，暂时忽略（无效、状态保持）Cnt 指令。
- (6) Mva2 指令执行时，暂时忽略（无效、状态保持）Fine 指令。
- (7) 在弹性伺服功能有效时，指定 <定位距离> 时，根据动作，机器人无法到达目的位置的定位完成条件，通过等待指令的完成，无法执行程序。弹性伺服功能有效时，请勿进行 <定位距离> 的指定。
- (8) Mva2 指令限定为对应机型。通过其他类型的机器人使用时将发生错误 H3391。关于对应机型，请参照【可使用的机器人机型】。
- (9) 在始终执行程序内无法使用。
- (10) Mva2 指令执行中中断并在 JOG 前进后重新开始运行时，返回至中断位置后再重新开始执行 Mva2 指令。返回至中断位置的插补方法（关节插补 / 直线插补）可以通过 RETPATH 参数进行更改。

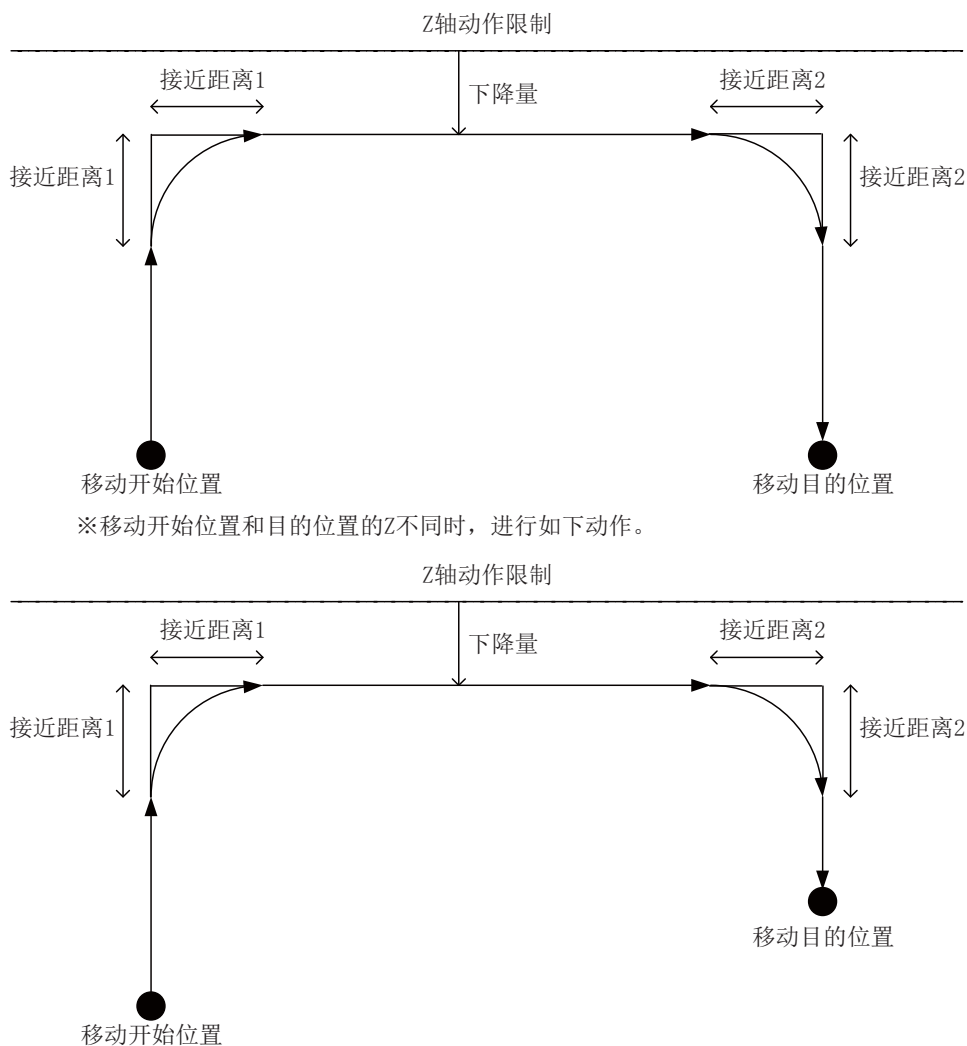
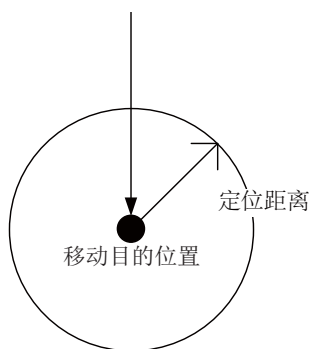


图 4-27: Mva2 指令的动作轨迹 (从水平方向看的情况)



进入〈定位距离〉中已指定的半径的球时定位完成。

图 4-28: Mva2 指令的定位距离

**【可使用的机器人机型】**

RH-FRH 系列 (RH-3FRHR 系列除外)

**【相关参数】**

MVACNT1

## 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,(P1t 1,5),P4 WthIf 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，移动圆周。(360 度)
- (2) 圆弧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，之后进行圆弧插补。
- (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 510 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。

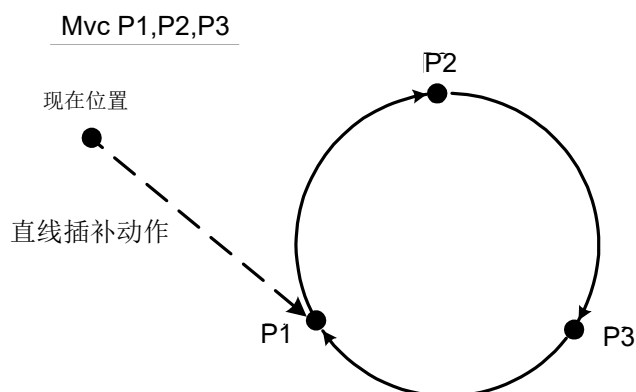


图 4-29：圆插补的动作轨迹示例

## 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WthIf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, (Plt 1, 5), P4 WthIf 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 510 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 533 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。

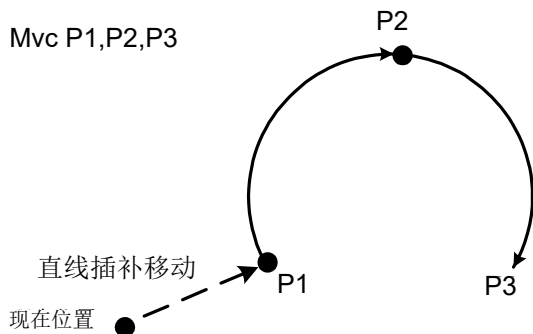


图 4-30：圆弧插补的动作轨迹示例其 1



## ◇◆◇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-4FRJL）的情况下，到达目的位置后，虽然姿势数据会显示已登录的值，但实际为偏移了 +360 度或 -360 度的姿势。

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WthIf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,(P1t 1,5),P4 WthIf 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 510 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 533 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。

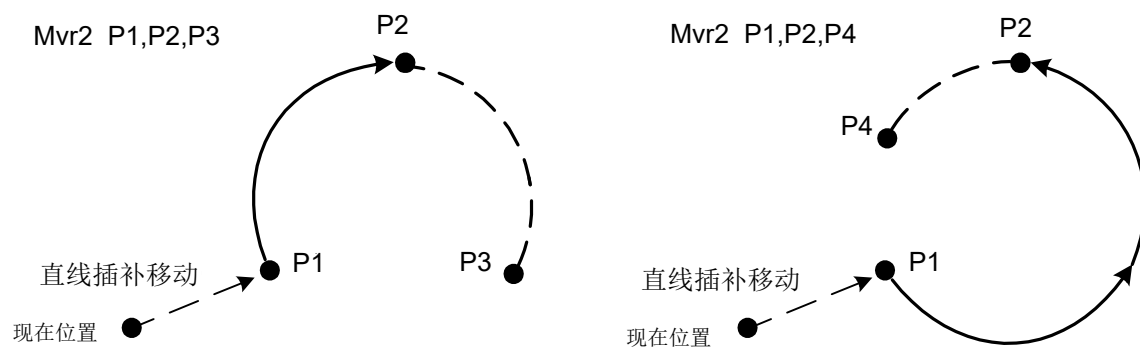


图 4-31：圆弧插补的动作轨迹示例其 2

## ◇◆◇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-4FRJL）的情况下，到达目的位置后，虽然姿势数据会显示已登录的值，但实际为偏移了 +360 度或 -360 度的姿势。

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WthIf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,(P1t 1,5),P4 WthIf 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510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。仅有限定的机型可以支持特异点通过。关于详细内容请参照第533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 < 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。

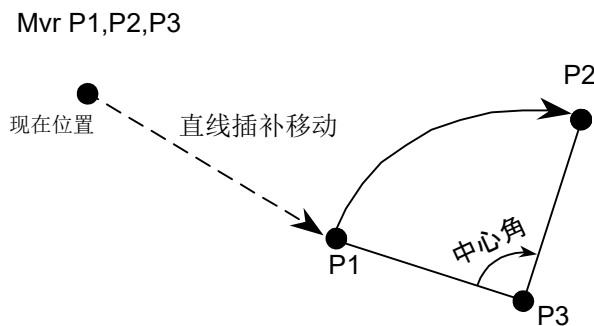


图 4-32：圆弧插补的动作轨迹示例其 3

## ◇◆◇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-4FRJL）的情况下，到达目的位置后，虽然姿势数据会已显示登录的值，但实际为偏移了 +360 度或 -360 度的姿势。

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WthIf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 WthIf 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 500 页的“5.6 关于标准 TOOL 坐标”。

“图 4-33: 直线插补时的动作示例”是一个例子。

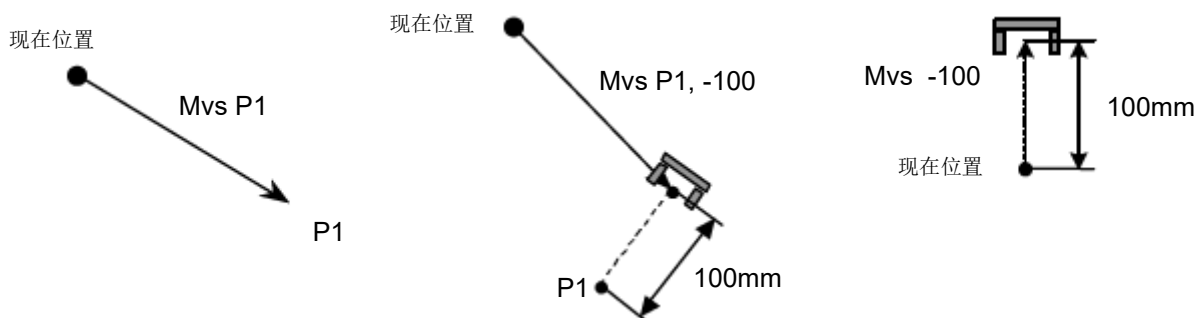
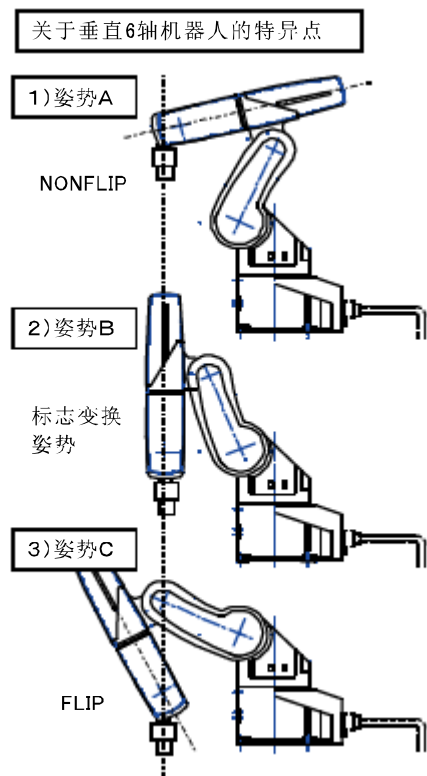


图 4-33: 直线插补时的动作示例

- (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 510 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。仅有限定的机型可以支持特异点通过。关于详细内容请参照第 533 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 4-34: 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-FR 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 655 页的“7.2 关于样条插补”。)

### 【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 Ovrld 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>) × (Ovrld 指令) × (操作面板的倍率修调)
 不使用 Spd 指令与 JOvrld 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 660 页的“表 7-5: 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 Ovrld 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。)

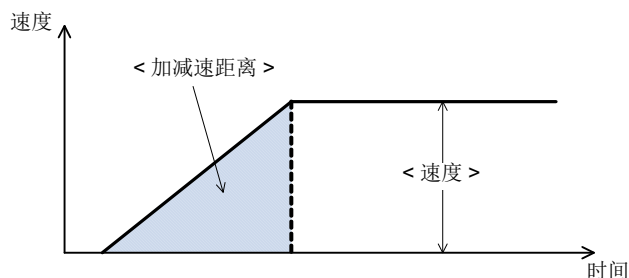


图 4-35: <加减速距离>导致的加速

- (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 MvSp1 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 MvSp1 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 MvSp1 指令。会发生 L3287（启动条件为 ERR、ALW 的情况下无法使用此指令。）
- (23) MvSp1 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（MvSp1））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 4-15: 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的动作特性。
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 523 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如表 4-16 所示。

表 4-16: 与 MvTune 标准模式的不同

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 Oad1 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 Oad1Off 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)、Oad1 (Optimal Acceleration)、Prec (Precision)

【相关参数】

ACCMODE、HNDDATO ~ 8、WRKDATO ~ 8

Mxt

【功能】

每隔 1 个单位控制器控制时间，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。
控制器控制时间在 CR800-Q 下约 7.1ms、在 CR800-D/R 下约 3.5ms（设定用户机械时约 7.1ms）。

【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可读取 1 个位置指令并进行动作。动作控制时间在 CR800-Q 下约 7.1ms、在 CR800-D/R 下约 3.5ms（设定用户机械时约 7.1ms）。
- (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起动为止。
- (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

NVClose

【功能】

切断与指定的视觉传感器的线路。

【格式】

NVClose □ [[#]<视觉传感器编号> [, [[#]<视觉传感器编号> · · ·]

【用语】

<视觉传感器编号> (可省略): 指定 1 ~ 8 的常数 (视觉传感器的编号)。以编号表示连接 <COM 编号> 中指定的 COM 的视觉传感器。
省略时, 切断通过 NVOpen 指令打开的全部线路 (视觉传感器用的线路)。
此外, 指定的 <视觉传感器编号> 可通过最多 8 个逗号分隔指定。
设定范围: 1 ~ 8

【程序示例】

```
100 If M_NvOpen(1)<>1 Then ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1 ' 与连接至 COM2: 的视觉传感器连接, 并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1 ' 与视觉传感器编号 1 连接, 等待至登录完成
140 . . . . .
300 NVClose #1 ' 切断与 COM2: 连接的视觉传感器的线路。
```

【说明】

- (1) 切断与通过 NVOpen 指令连接的视觉传感器之间的线路。
- (2) 省略 <视觉传感器编号> 时, 将切断全部线路。
- (3) 已经切断时, 进到下一步。
- (4) 可同时与最多 7 台视觉传感器连接, 因此为了识别切断哪台视觉传感器的线路, 要使用 <视觉传感器编号>。
- (5) 执行本指令中中断程序时, 将继续执行直至本指令的处理完成。
- (6) 多任务中使用本指令时, 将在使用的任务中执行 NVOpen 指令并仅切断打开的线路。此外, 此时使用的 <视觉传感器编号> 应使用通过 NVOpen 指令指定的编号。
- (7) 程序的启动条件不支持 『ALWAYS』。
- (8) 使用了 End 指令时, 将切断通过 NVOpen 指令或 Open 指令打开的全部线路。但是, 以 CallP 指令调用的程序内的 End 指令中不关闭线路。此外, 程序复位时也会关闭线路, 因此实施 End 指令或程序复位时, 不需要通过本指令切断路线。
- (9) 执行本指令中中断条件成立时, 本指令完成后将执行中断处理。

【错误】

- (1) 若 <视觉传感器编号> 中指定的数值为 “1” ~ “8” 以外, 将发生 “变量值超出范围” 错误。
- (2) 指令的变量的个数多于 8 时, 将发生 “变量的个数不正确。” 错误。

NVLoad

【功能】

将指定的视觉程序加载至视觉传感器。

【格式】

NVLoad □ #< 视觉传感器编号 >, < 视觉程序 (作业) 名 >

【用语】

< 视觉传感器编号 > (不可省略): 指定控制视觉传感器的编号。
 设定范围: 1 ~ 8

< 视觉程序 (作业) 名 > (不可省略): 指定要启动的视觉程序名。
 视觉程序的扩展名 (.job) 可省略
 可使用的字符仅为 '0' ~ '9'、'A' ~ 'Z'、'a' ~ 'z'、'-'、'_'。

【程序示例】

```

100 If M_NvOpen(1)<>1 Then      ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1      ' 与连接至 COM2: 的视觉传感器连接, 并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1        ' 与视觉传感器编号 1 连接, 等待至登录完成
140 NVLoad #1,"TEST"          ' 加载 "TEST" 程序。
150 NVRun #1,"TEST"           ' 加载 "TEST" 程序。
160 EBRead #1,,MNUM,PVS1,PVS2 ' 读取 "Job.Robot.FormatString" 符号的数据并保存至变量
                                MNUM、PVS1、PVS2 中。

170 . . . . .
300 NVClose #1                ' 切断与 COM2: 连接的视觉传感器的线路。

```

【说明】

- (1) 指定的视觉传感器中, 加载指定的视觉程序。
- (2) 本指令在视觉传感器中加载视觉程序时进到下一步。
- (3) 执行本指令的过程中程序中断时, 将立即中断。
- (4) 已加载指定的 < 视觉程序名 > 时, 以无处理结束指令。
- (5) 多任务中使用本指令时, 需要在使用的任务中执行 NVOpen 指令。此外, 请使用通过 NVOpen 指令指定的 < 视觉传感器编号 >。
- (6) 程序的启动条件不支持 『ALWAYS』。
- (7) 执行本指令过程中中断条件成立时, 将立即执行中断处理。

【错误】

- (1) 各变量的数据类型不同时, 将发生 “输入的指令语句的语法有误” 错误。
- (2) 指令的变量的个数异常 (多 / 少) 时, 将发生 “变量的个数不正确。” 错误。
- (3) < 视觉传感器编号 > 为 “1” ~ “8” 以外时, 将发生 “超出变量值范围” 错误。
- (4) 因 < 视觉传感器编号 > 中设定的编号未执行 NVOpen 指令时, 将发生 “视觉传感器编号指定异常” 错误。
- (5) < 视觉程序名 > 超过 15 个字符的字符数时, 将发生 “视觉程序名异常” 错误。
- (6) < 视觉程序名 > 中使用了 “0” ~ “9”、“A” ~ “Z”、“-”、“_” 以外 (也包含小写字母的字母) 时, 将发生 “视觉程序名异常” 错误。
- (7) < 视觉程序名 > 中指定的程序不存在于视觉传感器内时, 将发生 “不存在视觉程序” 错误。
- (8) 视觉传感器为 『离线』时, 将发生 “设为在线” 错误, 因此应将视觉传感器设为 『在线』。
- (9) 执行本指令过程中通信线路被切断时, 将发生 “通信异常” 错误, 并关闭机器人控制器侧的线路。

NVOpen

【功能】

与指定的视觉传感器连接，登录到该视觉传感器。

【格式】

NVOpen □ “<COM 编号>” □ As □ #<视觉传感器编号>

【用语】

<COM 编号> (不可省略): 指定与 Open 指令相同通信线路的编号。
设定范围: “COM1:” ~ “COM8:”

<视觉传感器编号> (不可省略): 指定 1 ~ 8 的常数 (视觉传感器的编号)。以编号表示连接 <COM 编号> 中指定的 COM 的视觉传感器。
此外, 应注意该编号与 Open 指令的 <文件编号> 共享。
设定范围: 1 ~ 8

【程序示例】

```
100 If M_NvOpen(1)<>1 Then ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1 ' 与连接至 COM2: 的视觉传感器连接, 并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1 ' 与视觉传感器编号 1 连接, 等待至登录完成
```

【说明】

- (1) 与连接 <COM 编号> 中指定的线路的视觉传感器连接, 登录到该视觉传感器。
- (2) 最多可同时连接 7 台视觉传感器。使用 <视觉传感器编号> 来识别与哪个视觉传感器通信。
- (3) 与 Open 指令并用时, 由于 Open 指令的 <COM 编号>、<文件编号> 与本指令的 <COM 编号>、<视觉传感器编号> 共享, 应使用 Open 指令的 <COM 编号>、<文件编号> 中指定外的编号。

(例) 正常的示例 变为错误的示例

```
10 Open "COM1:" As #1 10 Open "COM2:" As #1
20 NVOpen "COM2:" As #2 20 NVOpen "COM2:" As #2 ⇒使用 <COM 编号>
30 NVOpen "COM3:" As #3 30 NVOpen "COM3:" As #1 ⇒使用 <视觉传感器编号>
```

此外, 在 1 台机器人控制器与 1 台视觉传感器的构成中, 无法打开 2 个以上的线路。

设定参数【NETHSTIP】时设定了相同 IP 地址的情况下, 将发生“Ethernet 参数 NETHSTIP 设定异常”错误。

- (4) 需要“用户名”与“密码”来登录视觉传感器。需要将视觉传感器中设定的可全访问的用户名与该密码设定为机器人控制器的参数【NVUSER】与【NVPSWD】。
用户名、密码均可使用最多 15 个字符的数字 (0 ~ 9) 与字母 (A ~ Z)、“-”、“_”。(由于示教单元仅支持大写字母, 使用新用户时, 应以大写字符设定视觉传感器中设定的密码。)
购买网络视觉传感器时的全访问权限用户名为“admin”。该密码为“”。因此参数【NVUSER】与【NVPSWD】的初始值为【NVUSER】=“admin”【NVPSWD】=“”。
MELFA-Vision 中更改“admin”的密码时, 或登录新用户时, 应更改参数【NVUSER】与【NVPSWD】。更改时, 若要显示参数【NVPSWD】的内容, 会显示“****”。更改视觉传感器侧的密码时, 应打开参数【NVPSWD】, 直接更改显示的“****”。更改后, 复位机器人控制器的电源。

【注意事项】

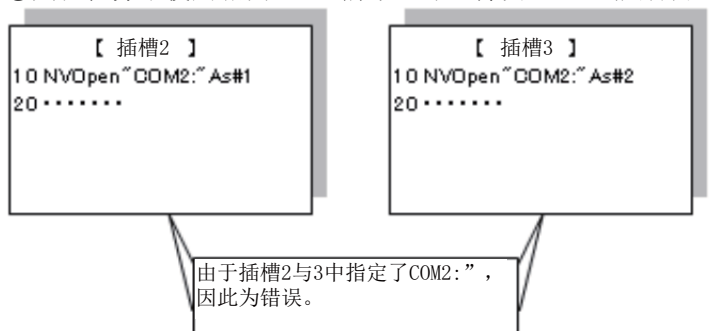
此外, 1 台机器人控制器连接多台视觉传感器时, 应将全部视觉传感器中设定的用户名与密码设为相同设定。

- (5) 可通过 M_NvOpen 确认与执行本指令时的网络视觉传感器的通信状态。详细内容请参照 M_NvOpen 的说明。
- (6) 执行本指令过程中中断程序时, 将立即停止。登录到视觉传感器, 需要复位机器人程序后再启动。

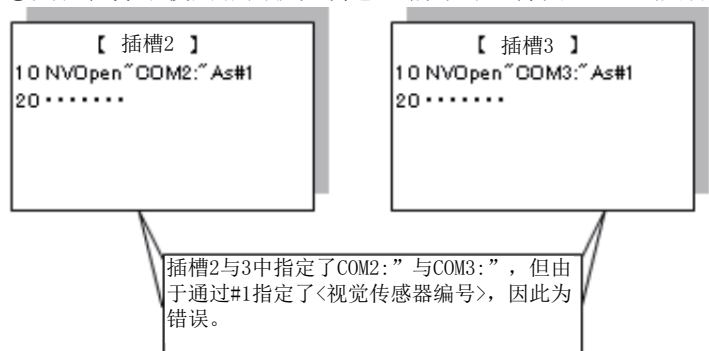
(7) 多任务中使用本指令时，有以下限制。

<COM 编号> 与 <视觉传感器编号> 在多任务的各个任务中无法重复。

①其他任务中使用相同 <COM 编号> 时，将发生“试图打开已打开的通信文件”错误。



②其他任务中使用相同视觉传感器编号时，将发生“试图打开已打开的通信文件”错误。



(8) 程序的启动条件不支持『ALWAYS』。

(9) 3 台机器人可同时控制相同视觉传感器。第 4 台机器人登录后第 1 台机器人的线路会被切断，因此构建系统时应充分注意。

(10) 以 Ca11P 指令调用的程序的 End 指令中不关闭线路，但在主程序的 End 指令中关闭线路。此外，在程序复位时也关闭线路。

(11) 执行本指令中中断条件成立时，即使在本指令处理中也会立即执行中断处理。

【错误】

(1) 各变量的数据类型不同时，将发生“输入的指令语句的语法有误”错误。

(2) 指令的变量的个数异常（多/少）时，将发生“变量的个数不正确。”错误。

(3) 若 <COM 编号> 中指定的字符为“COM2:”~“COM8:”以外，将发生“变量值超出范围”错误。

(4) 若 <视觉传感器编号> 中指定的数值为“1”~“8”以外，将发生“变量值超出范围”错误。

(5) 指定线路已连接的 <COM 编号>（也包含通过 Open 指令线路开路 <文件编号>）时，将发生“试图打开已打开的通信文件”错误。

(6) 线路打开的目标中视觉传感器未连接时，将发生“视觉传感器未连接”错误。（使用与 Ethernet 的规格相同设定生产厂商参数【COMTIMER】。当前“1s”）

(7) 其他任务中指定了相同 <COM 编号> 或相同 <视觉传感器编号> 时，将发生“试图打开已打开的通信文件”错误。

(8) 参数【NVUSER】（用户名）与【NVPSWD】（密码）中指定的用户名与密码不同时，将发生“密码异常”错误。

(9) 执行本指令过程中通信线路被切断时，将发生“通信异常”错误，并关闭机器人控制器侧的线路。

(10) 启动条件在『ALWAYS』的程序中使用，将发生“启动条件为 ERR、ALW 时无法使用该指令”错误。

NVRun

【功能】

启动指定的视觉程序。

【格式】

NVRun □ #< 视觉传感器编号 >, < 视觉程序 (作业) 名 >

【用语】

- < 视觉传感器编号 > (不可省略): 指定控制视觉传感器的编号。
 设定范围: 1 ~ 8
- < 视觉程序 (作业) 名 > (不可省略): 指定要启动的视觉程序名。
 视觉程序的扩展名 (.job) 可省略
 可使用的字符仅为 '0' ~ '9'、'A' ~ 'Z'、'a' ~ 'z'、'-'、'_'。

【程序示例】

```

100 If M_NvOpen(1)<>1 Then      ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1      ' 与连接至 COM2: 的视觉传感器连接, 并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1        ' 与视觉传感器编号 1 连接, 等待至登录完成
140 NVLoad #1,"TEST"          ' 加载 "TEST" 程序。
150 NVRun #1,"TEST"           ' 加载 "TEST" 程序。
160 EBRead #1,,MNUM,PVS1,PVS2 ' 读取 "Job.Robot.FormatString" 符号的数据并保存至变量
                                MNUM、PVS1、PVS2 中。

170 . . . . .
300 NVClose #1                ' 切断与 COM2: 连接的视觉传感器的线路。
    
```

【说明】

- (1) 指定的视觉传感器中, 启动指定的视觉程序。
- (2) **完成本指令处理的时机根据参数 NVTRGTMG 的设定有所不同。参数 NVTRGTMG 为出厂设定时, 在与视觉传感器的图像处理指令 (摄像请求) 的通信结束时移动至下一个指令执行。**
- (3) 执行本指令的过程中程序中中断时, 将立即中断。
- (4) 已加载指定的 < 视觉程序名 > 时, 仅执行摄像及图像处理。(不进行视觉程序的加载。)
- (5) 使用 EBRead 指令接收视觉传感器中的数据。
- (6) 多任务中使用本指令时, 需要在使用的任务中执行 NVOpen 指令。此外, 请使用通过 NVOpen 指令指定的 < 视觉传感器编号 >。
- (7) 程序的启动条件不支持 『ALWAYS』。
- (8) 请将 EasyBuilder 的画像取得设定的触发设定设定为 “外部”、“手动” 或 “网络”。
 (参数 NVTRGTMG 的设定值为 0、2 时即使是 “照相机” 也可进行)
- (9) 最多 3 台机器人可控制相同视觉传感器, 但本指令无法同时在多个机器人中使用。请在任意一台机器人中使用。
- (10) 执行本指令过程中中断条件成立时, 将立即执行中断处理。

【错误】

- (1) 各变量的数据类型不同时，将发生“输入的指令语句的语法有误”错误。
- (2) 指令的变量的个数异常（多/少）时，将发生“变量的个数不正确。”错误。
- (3) <视觉传感器编号>为“1”～“8”以外时，将发生“超出变量值范围”错误。
- (4) 因<视觉传感器编号>中设定的编号未执行 NVOpen 指令时，将发生“视觉传感器编号指定异常”错误。
- (5) <视觉程序名>超过 15 个字符的字符数时，将发生“视觉程序名异常”错误。
- (6) <视觉程序名>中使用了“0”～“9”、“A”～“Z”、“-”、“_”以外（也包含小写字母的字母）时，将发生“视觉程序名异常”错误。
- (7) <视觉程序名>中指定的程序不存在于视觉传感器内时，将发生“不存在视觉程序”错误。
- (8) EasyBuilder 的画像取得设定的触发设定设定为“外部”、“手动”或“网络”以外时，将发生“画像取得指定异常”错误。
（参数 NVTRGTMG 的设定值为 1 时，即使设定为“照相机”时也会变为相同错误）
- (9) 视觉传感器为『离线』时，将发生“设为在线”错误，因此应将视觉传感器设为『在线』。
- (10) 执行本指令过程中通信线路被切断时，将发生“通信异常”错误，并关闭机器人控制器侧的线路。

NVTrg

【功能】

在指定的视觉程序中请求摄像。

【格式】

```
NVTrg □ #< 视觉传感器编号 >, < 延迟时间 >, < 编码器 1 值读取变量 >[, [< 编码器 2 值读取变量 >]
[, [< 编码器 3 值读取变量 >] [, [< 编码器 4 值读取变量 >]
[, [< 编码器 5 值读取变量 >] [, [< 编码器 6 值读取变量 >]
[, [< 编码器 7 值读取变量 >] [, [< 编码器 8 值读取变量 >]
```

【用语】

- < 视觉传感器编号 > (不可省略): 指定控制视觉传感器的编号。
设定范围: 1 ~ 8
- < 延迟时间 > (不可省略): 向指定的视觉传感器输出摄像请求后, 指定取得编码器值为止的延迟时间 (单位: ms)。
设定范围: 0 ~ 150ms
- < 编码器 n 值读取变量 > (第 2 个以后可以省略): 指定复位读取的外部编码器 n 的值的精度数值变量。
注) n为1~8

【程序示例】

```
100 If M_NvOpen(1)<>1 Then      ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1      ' 与连接至 COM2: 的视觉传感器连接, 并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1        ' 与视觉传感器编号 1 连接, 等待至登录完成
140 NVLoad #1, "TEST"         ' 加载 "TEST" 程序。
150 NVTrg #1, 15, M1#, M2#     ' 向视觉传感器输出摄像请求, 在 15s 后取得编码器 1、2
160 EBRead #1, , MNUM, PVS1, PVS2 ' 读取 "Job.Robot.FormatString" 符号的数据并保存至变量
                                  MNUM、PVS1、PVS2 中。
170 . . . . .
300 NVClose #1                ' 切断与 COM2: 连接的视觉传感器的线路。
```

【说明】

- (1) 向指定的视觉传感器输出摄像请求, 在指定时间后取得编码器值。将取得的编码器值存储至指定的数值变量中。
- (2) **完成本指令处理的时机根据参数 NVTRGTMG 的设定有所不同。参数 NVTRGTMG 为出厂设定时, 在视觉传感器进行图像处理指令 (摄像请求), 视觉图像处理完成后移动至下一个指令执行。**
- (3) 执行本指令的过程中程序中中断时, 将立即中断。
- (4) 使用 EBRead 指令接收视觉传感器中的数据。
- (5) 多任务中使用本指令时, 需要在使用的任务中执行 NVOpen 指令。此外, 请使用通过 NVOpen 指令指定的 < 视觉传感器编号 >。
- (6) 程序的启动条件不支持 『ALWAYS』。
- (7) 请将 EasyBuilder 的画像取得设定的触发设定设定为 “外部”、“手动” 或 “网络”。
(参数 NVTRGTMG 的设定值为 0 时即使是 “照相机” 也可进行)
- (8) 最多 3 台机器人可控制相同视觉传感器, 但本指令无法同时在多个机器人中使用。请在任意一台机器人中使用。
- (9) 执行本指令过程中中断条件成立时, 将立即执行中断处理。

【错误】

- (1) 各变量的数据类型不同时，将发生“输入的指令语句的语法有误”错误。
- (2) 指令的变量的个数异常（多/少）时，将发生“变量的个数不正确。”错误。
- (3) <视觉传感器编号>为“1”～“8”以外时，将发生“超出变量值范围”错误。
- (4) 因<视觉传感器编号>中设定的编号未执行 NVOpen 指令时，将发生“视觉传感器编号指定异常”错误。
- (5) 视觉程序的画像取得指定被设定为“照相机”（所有触发指令）、“外部触发”、“手动触发”以外时，将发生“画像取得指定异常”错误。
- (6) EasyBuilder 的画像取得设定的触发设定设定为“外部”、“手动”或“网络”以外时，将发生“画像取得指定异常”错误
（参数 NVTRGTMG 的设定值为 1,2 时，即使设定为“摄像头”时也会变为相同错误）
- (7) 视觉传感器为『离线』时，将发生“设为在线”错误，因此应将视觉传感器设为『在线』。
- (8) 执行本指令过程中通信线路被切断时，将发生“通信异常”错误，并关闭机器人控制器侧的线路。

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 468 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 523 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 477 页的“JADL”。

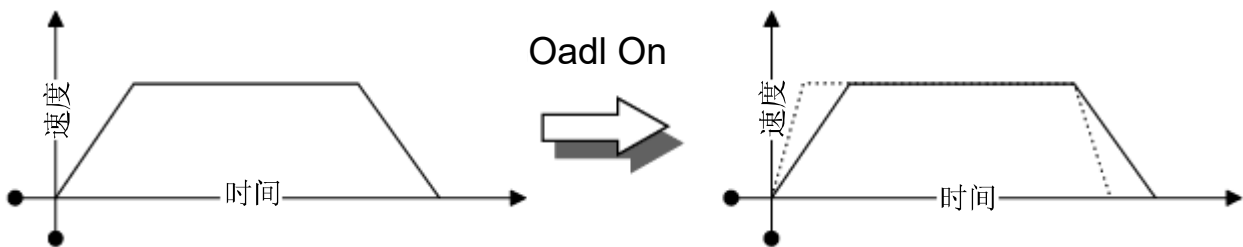


图 4-36: 轻负载时的加减速模式

【相关指令】

Accel、LoadSet (Load Set)、HOpen/HClose (抓手 Open/Close)

【关连参数】

HNDDATO ~ 8、WRKDATO ~ 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... GoSub (ON Go Subroutine)

【功能】

呼出指定值对应的 Level 的子程序。

【格式】

On < 式 > GoSub [< 呼出处 >], [< 呼出处 >]...

【用语】

< 式 > 用数值运算式指定往第几号的Level分支。
 < 呼出处 > 记述Level名。最大数为32。

【例子】

从输入信号 16 将 3 位量的值设定在 M1，依据其 M1 的值 (1 ~ 7) 进行分支。

(M1 为 1 时呼出 level ABC1、为 2 时呼出 level Lsub、为 3、4、5 时呼出 level LM1_345、为 6、7 时呼出 level L67)

```

1 M1 = M_Inb(16) And &H7
2 On M1 GoSub *ABC1,*Lsub,*LM1_345,*LM1_345,*LM1_345,*L67,*L67

100 *ABC1
101 ' 记述 M1=1 时的处理
102 Return ' 务必以 Return 返回。

121 *Lsub
122 ' 记述 M1=2 时的处理
123 Return ' 务必以 Return 返回。

170 *L67
171 ' 记述 M1=6 或 M1=7 时的处理
172 Return ' 务必以 Return 返回。

200 *LM1_345
201 ' 记述 M1=3、M1=4、M1=5 时的处理
202 Return ' 务必以 Return 返回。
    
```

【说明】

- (1) 以 < 式 > 的值决定呼出那个 Level 的子程序。
 例如：< 式 > 的值为 2 的情况下，会呼出以第 2 号记述的 level。
- (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。
- (4) 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。

< 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

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 LIMP 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,*LIMP,*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 *LIMP
```

```
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（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。
 • 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, Random, Binary
通信端口	COM1: 在参数「COMDEV」的设定 : COM8: 在参数「COMDEV」的设定	省略=TCP 通信 UDP=UDP 通信
	ENET: 192.168.0.2 ^{注1)}	Mxt
	MXT: QRBUS<CPU 号机编号 2~4> CPU 号机编号为 2~4, 指定主站侧机器人的 CPU 编号。	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、通信线路(TCP通信)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。
 • Binary: 指定二进制文件模式。
 • UDP : 使用UDP进行通信。
- <文件号码> 以1~8的常数记述。

【示例】

① 文件操作 (在控制器内创建名称为“temp.txt”的文件、写入“abc”。)

- ```
1 Open "temp.txt" For Append As #1' 将名为 temp.txt 的文件在 Append 模式下
 ' 作为文件编号 1 打开。
2 Print #1, "abc" ' 在文件中写入 "abc"。
3 Close #1 ' 关闭文件。
```

② 文件操作 (写入到二进制文件)

- ```
1 Open "temp.dat" For Binary As #1' 将 temp.dat 文件在二进制模式下
   ' 作为文件编号 1 打开。
2 BPrint #1, P_Curr.X, 4 ' 以二进制形式每 4 个字节一组写入当前位置。
3 BPrint #1, P_Curr.Y, 4
4 BPrint #1, P_Curr.Z, 4
5 BPrint #1, P_Curr.A, 4
6 BPrint #1, P_Curr.B, 4
7 BPrint #1, P_Curr.C, 4
8 Close #1 ' 关闭文件。
```

③ 通信线路

```

1 Open "COM1:" As #1 ' 将以参数 COMDEV 的第 1 个要素设定的通信线路
                      ' 作为文件编号 1 开放。

2 Mov P_01
3 Print #1, P_Curr    ' 将当前位置输出外部。(以下的形式)
                      ' "(100.00,200.00,300.00,400.00,500.00,600.00,700.00,800.00)(7,0)"

4 Input #1, M1, M2, M3 ' 以 "101.00,202.00,303.00" 的 ASCII 形式从外部接收。
5 P_01.X = M1
6 P_01.Y = M2
7 P_01.C = Rad(M3)   ' 复制到 Golbal 数据。
8 Close              ' 关闭被开放的全部文件。
9 End

```

④ 通信线路 (UDP 通信)

```

1 Open "COM1:" For UDP As #1 ' 对以参数 COMDEV 的第 1 个要素设定的
                              ' 通信线路使用 UDP 通信并作为文件编号 1
                              ' 开放。
                              ' 事先登录校准用的示教点。

2 Dim PRob(4)
3 For Mno=1 To 4
4   Mov PRob(Mno)
5   Print #1, "TRIGGER"      ' 向传感器发送触发器。
6   BInput #1, MX, 4         ' 以二进制形式取得传感器的识别数据。
7   BInput #1, MY, 4
8   BInput #1, MC, 4
9   PCam.X = MX
10  PCam.Y = MY
11  PCam.C = MC
12  VSSetCP 1, Mno, PCam, PRob(Mno) ' 设定传感器坐标系与机器人坐标系的对应点。
13 Next Mno
14 VSRegCD 1                ' 求出校准数据。
15 Close #1                 ' 关闭通信线路。
16 End

```

【说明】

- (1) 通过指定了〈文件名〉中记述的文件的文件编号开启。
执行到文件的读写需要使用这个文件编号。
- (2) 通信线路也视为文件。
- (3) 读写二进制数据时，使用 BInput / BPrint 指令。

【相关指令】

Input、Print、Close

【相关参数】

COMDEV

Ovrd

【功能】

机器人动作的速度以 0.01 ~ 100% 指定。涉及到全部程序的倍率修调。

【格式】

Ovr d □ < 速度比例 >

Ovr d □ < 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) 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 Ovr d 50
2 Mov P1
3 Mvs P2
4 Ovr d M_NOvr d          ' 设定初始值
5 Mov P1
6 Ovr d 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) Ovr d 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(Ovr d 指令)) × (关节速度比例(JOvr d 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(Ovr d 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_NOvr d)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Ovr d指令的话，下次Ovr d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】

JOvr d (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_NOvr d (系统的初始值)、M_Ovr d (现在的指定速度)

Plt

【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 Pa11t 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 117 页的“4.1.2 托盘运算”。

【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 523 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】

HNDDATO ~ 8、WRKDATO ~ 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中，按顺序执行多个程序的行（初始值为逐行执行）并进行动作。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。

PrmRead

【功能】

可读取参数值。

【格式】

[格式 1] PrmRead [< 机械编号 >], < 参数名 >, < 值存储目标变量 >

【术语】

< 机械编号 >

指定要读取的参数的机械编号。不依赖机械的参数即使机械编号指定为1, 2, 3也可读取。以常数或变量指定。

(可省略…可读取通用与机械1)

设定范围: 0 ~ 3 Q 通用参数(机械1即使指定为0也可读取)

< 参数名 >

指定要读取的参数名。

< 值存储目标变量 >

指定要设定已读取的参数值的变量。

数值型…仅读取第1要素, 整数时, 进行四舍五入

位置型…读取第1~8要素

关节型…读取第1~8要素

字符串型…以字符读取全部的要素数, 最多可读取127个

【示例】

- | | |
|--------------------------------|---------------------------|
| 1 PrmRead 0, "LNG", CLNG\$ | ' 读参数 LNG |
| 2 PrmRead 1, "LNG", CLNG\$ | ' 机械 1 也可读取 |
| 3 PrmRead 1, "MEJAR", CMEJAR\$ | ' 读参数 MEJAR, 但 128 字符以后丢失 |
| 4 Dim CDIM\$(16) | |
| 5 PrmRead 0, "ROMDRV", MDRV | ' 转换成数值读取 |
| 6 PrmRead 0, "AREA1P1", PAREA1 | ' 将参数 AREA1P1 读取至 P 变量中 |

【说明】

- (1) 仅可读取用户公开参数。

【相关指令】

[PrmWrite](#)

PrmWrite

【功能】

可写入参数值。

【格式】

PrmWrite [机械编号], 参数名, 参数值

【术语】

机械编号

指定要写入的参数的机械编号。不依赖机械的参数时，即使机械编号指定为1, 2, 3也可写入。

以常数或变量指定。

(可省略...可写入通用与机械1)

设定范围: 0 ~ 3 0; 通用参数(机械1即使指定为0也可读取)

参数名

指定要写入的参数名。

参数值

指定要写入的参数值。

可以数值指定。但是，字符转换后设定为字符型的参数。

数值型...仅写入第1要素，整数时进行四舍五入

位置型...写入第1~8要素

关节型...写入第1~8要素

字符串型...以字符写入全部的要素数，最多可写入127个

【示例】

1 PrmWrite 0, "LNG", CLNG\$	' 参数 LNG 写入
2 PrmWrite 1, "LNG", CLNG\$	' 即使在机器 1 也可写入
3 PrmWrite 1, "MEJAR", CMEJAR\$	' 参数 MEJAR 写入
4 PrmWrite 0, "ROMDRV", MDRV	' 转换为数值后进行写入
5 PrmWrite 0, "AREA1P1", PAREA1	' 将参数 AREA1P1 写入 P 变量中

【说明】

- (1) 可写入用户公开参数。
- (2) 参数写入后，需要重新接通电源。

【相关指令】

[PrmRead](#)

PVSCal

【功能】

使用以 RT ToolBox3 的 2D 视觉校准功能设定的视觉传感器校准数据（参数 VSCALB1 ~ 8），将视觉传感器的画像坐标转换为机器人的世界坐标。

【格式】

<位置变量>=PVSCal(<校准编号>, <视觉 X>,
<视觉 Y>, <视觉 θ>[, <基准位置变量>])

【术语】

<位置变量>	指定要代入的位置变量。 以机器人的世界坐标返回坐标转换的计算结果。
<校准编号>	以参数VSCALB 1 ~ 8的编号指定坐标转换中使用的视觉传感器校准数据。
<视觉X>	视觉传感器的画像坐标X[pixel]
<视觉Y>	视觉传感器的画像坐标Y[pixel]
<视觉θ>	视觉传感器的画像坐标θ [deg]
<基准位置变量>	以位置常数或位置变量指定基准位置。 在抓手中安装摄像头时,应将通过视觉传感器进行画像识别时的机器人的位置(画像识别时的机器人位置)指定为基准位置。进行以下相对运算。 <画像识别时的机器人位置>*<坐标转换的计算结果> 省略时为绝对坐标。

【示例】

```

1 ' 使用 Open/Print/Input 指令启动对象视觉传感器, 将取得的画像坐标 [pixel] 代入数值变量中。
2 ' MX= 视觉传感器 X[pixel]
3 ' MY= 视觉传感器 Y[pixel]
4 ' MT= 视觉传感器 θ [deg]
5 PVS=PVSCal (1, MX, MY, MT)          ' 使用校准编号 1 转换为机器人的坐标 (世界坐标)
6 PVS. Z=PDST. Z                      ' 指定 Z 的高度
7 Mov PVS, -50                        ' 向计算结果位置上空 50mm 移动
8 Mvs PVS                              ' 向计算结果位置移动

```

【说明】

- (1) 使用视觉传感器校准数据（参数 VSCALB1 ~ 8），将视觉传感器的画像坐标转换为机器人的世界坐标。坐标转换中使用的参数 VSCALB1 ~ 8 应使用 RT ToolBox3 的 2D 视觉校准功能预先进行设定。
- (2) 校准编号中设定 1 ~ 8 以外的数字时, 发生错误 L3110 (自变量的值在范围外)。
- (3) 自变量的个数不是四个或五个时, 发生错误 L3120 (执行时自变量个数)。
- (4) 自变量的类型不同时, 发生错误 L3810 (自变量的类型不同)。

【相关参数】

VSCALB1 ~ 8

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。

Remove

【功能】

删除文件。

【格式】

```
Remove □ "<文件名>"
```

【用语】

<文件名> 指定删除的文件名。

【示例】

1 Remove "temp.txt" ' 删除指定的文件。

【说明】

- (1) 删除通过 Open 指令创建的文件。
- (2) 不存在删除对象的文件时，执行时将变为错误 L7010。

【相关指令】

Open

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
  :
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。
  :
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)。关于中断处理请参照第 202 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

Save

【功能】

保存程序。

【格式】

Save □ <保存类型>[, <插槽编号>]

【术语】

<保存类型>	0: 保存插槽1程序与用户基本程序。 1: 保存用户基本程序。 2: 保存指定插槽的程序。
<插槽编号>	指定插槽编号。(1 - 32)以常数或变量指定。

【示例】

1 Save 0	' 保存插槽 1 与用户基本程序。
2 Save 1	' 保存用户基本程序。
3 Save 2, 3	' 保存插槽 3 的程序。

【说明】

- (1) 根据高速 DRAM 运行未保存运行时更改的变量值时，可使用本指令进行保存。
- (2) 程序保存完成前，程序在执行行待机。



注意

由于执行 Save 指令需要时间，因此使用本指令时对节拍时间有影响。

【相关参数】

AUTOSAVE

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>

记述数值运算式或字符串式。

<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>
- <字符串常数>

<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))
- (9) <条件> 与 <公式> 的类型不一致时，将发生错误 L3810 (自变量的种类不同)。

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。

【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”。

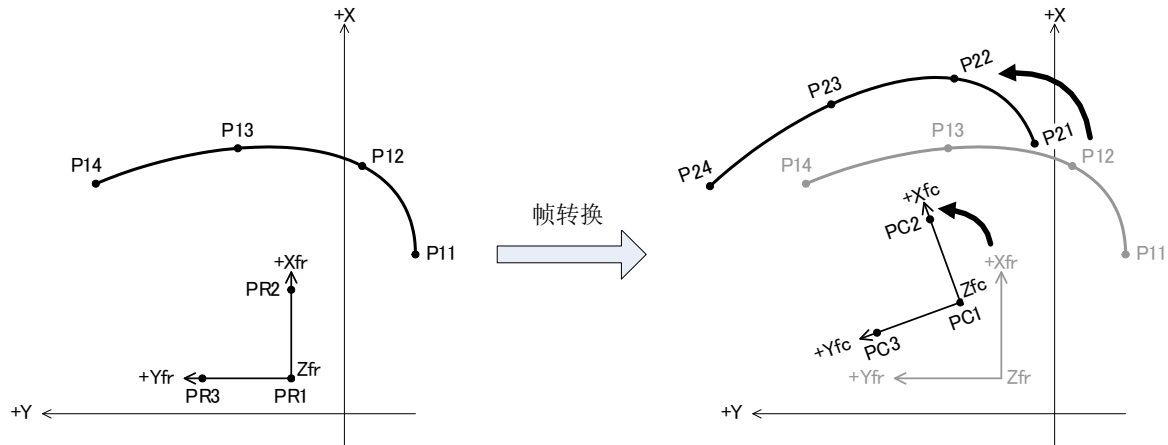


图 4-37：帧转换的示例

- (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。设定范围：0.01~10000[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超速报警对策
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) 通过执行程序 End 语句或程序复位，指定速度将被设定为系统初始值。

【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-3FRHR 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 4-38 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, 即自动调整速度避免发生错误。

例如, 通过指令速度 V 动作的过程中, 靠近原点, 保持动作可能会超速的情况下, 如图 4-39 的 A 所示, 自动降低速度以防止发生超速。此后, 通过原点附近, 可提高速度后, 如图 4-39 的 B 所示朝着指令速度 V 开始加速。

(2) 与 Ovrld 或 Spd 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 (Ovrld 或 Spd 指令) 更慢时 以指定速度进行动作
- 指定速度 (Ovrld 或 Spd 指令) 更快时 以本功能中的调整速度进行动作

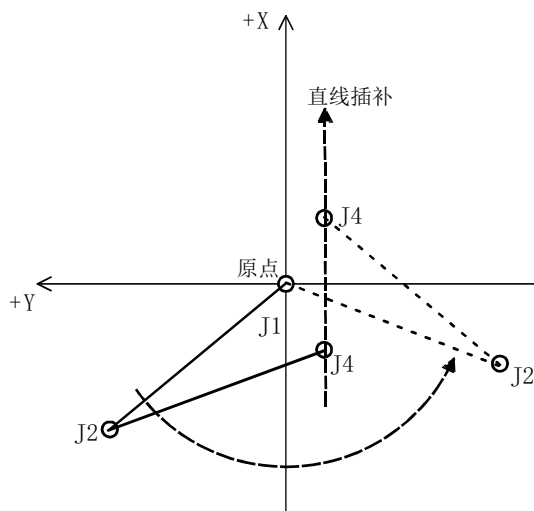


图 4-38: 原点附近的直线插补通过 (示例)

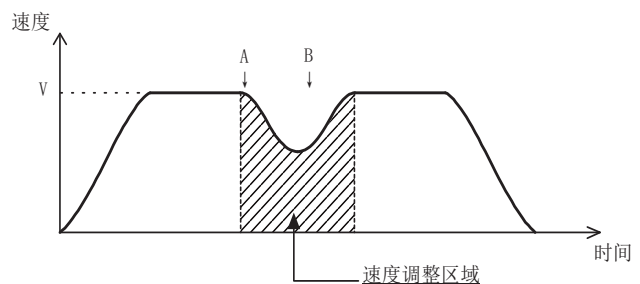


图 4-39: 速度调整时的速度状况

- (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，即使速度调整功能有效，J4 轴不通过“图 4-40 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。

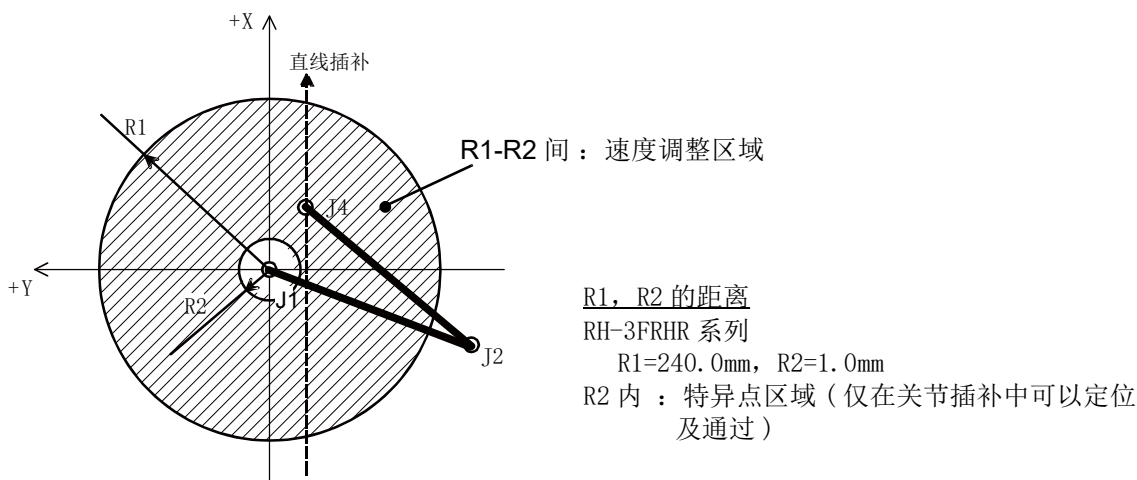


图 4-40：速度调整区域与特异点区域

- (4) 电源接通后的速度调整功能的初始状态可以通过参数 SPD OPT 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 SPD OPT=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，若想执行像 J4 轴通过图 4-40 的特异点区域那样的直线插补，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 origin，或通过 Ovr d 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-3FRHR 系列

【相关参数】

SPD OPT

SplFWrt (Spline Frame Write)

【功能】

在指定的样条文件中登录帧转换信息。

【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。

【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。
- <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

Static

【功能】

声明 Static 变量（静态变量）。

【格式】

```
Static □ Def □ Inte □ <数值变量名>[, <数值变量名>[, ...]]
Static □ Def □ Long □ <数值变量名>[, <数值变量名>[, ...]]
Static □ Def □ Float □ <数值变量名>[, <数值变量名>[, ...]]
Static □ Def □ Double □ <数值变量名>[, <数值变量名>[, ...]]
```

<数值变量名> 指定变量名。

```
Static □ Def □ Char □ <字符串变量名>[, <字符串变量名>[, ...]]
```

<字符串变量名> 指定变量名。

```
Static □ Def □ Jnt □ <关节变量名>[, <直交变量名>[, ...]]
Static □ Def □ Pos □ <直交变量名>[, <关节变量名>[, ...]]
Static □ Def □ Work □ <工件坐标变量名>[, <工件坐标变量名>[, ...]]
```

<关节变量名> 指定变量名。

<直交变量名> 指定变量名。

<工件坐标变量名> 指定变量名。

```
Static □ 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>]]) [, ...]]
```

<变量名> 指定排列变量名。

<要素数> 以常数指定排列变量的要素个数。

【示例】

```
1 Static Def Inte M1           ’ 将 M1 视为数值型的静态变量执行声明。
2 Static Def Char Message     ’ 将 Message 视为字符串的静态变量执行声明。
3 Static Def Pos P1           ’ 将 P1 视为直交型的静态变量执行声明。
4 Static Dim cmd$(5)         ’ 将 cmd 视为字符串型排列的静态变量执行声明。
```

【说明】

- (1) 声明静态变量。
- (2) Function 过程内声明的静态变量保持函数结束后的值。
- (3) 可与 Const 定义并用。

【相关指令】

Const、Def Inte/Def Long/Def Float/Def Double、Def Char (Define Character)、Def Jnt (Define Joint)、Def Pos (Define Position)

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】

指定工具转换数据。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 □ < 工具转换数据 >

【用语】

<工具转换数据> 以位置表达式(位置常数、位置变量等)指定工具转换数据。

【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) 通过本指令变更的 <工具转换数据> 会被保存在参数 MEXTL 中, 即使在控制器的电源 OFF 后也会保持。使用连续反复执行 Base 指令、Tool 指令、M_Tool 的程序时, 可能会无法即时进行参数保存, 从而发生错误 C7091 (参数保存失败), 应加以注意。发生错误 C7091 时, 应如下所示变更执行 Base 指令、Tool 指令的位置。

例)

<pre>*MAIN Base PB Tool PT If M_In(20)=1 Then GoSub *SUB1 EndIf GoTo *MAIN</pre>		<pre>Base PB Tool PT *MAIN If M_In(20)=1 Then GoSub *SUB1 EndIf GoTo *MAIN</pre>
--	---	--

- (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, 会采用已指定的工具转换数据。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工具转换数据不同的话, 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 501 页的“表 5-8: 对应机器人的形式工具转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 的参数设定为工具转换数据。

【关连参数】

MEXTL、MEXTL1 ~ 4 详细内容请参照第 500 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 (现在的指定工具转换数据)、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 指令中记述的条件的个数为 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，有 2 种对应方法。
 - ① 不使用 Wait 指令，而替换成 If ~ Then 指令。
例) *Loop
 If M_In(38)=1 Or M_In(39)=1 Then *Next1 Else *Loop
 *Next1
 - ② 将参数 PRSPEC 设定为 0 (条件指令方式)。
- (5) 多任务程序中，同时执行 Wait 指令与 CallP 指令时，可能对系统有影响。通过多任务可同时执行这些指令时，应将参数 PRSPEC 设定为 0 (条件指令方式)。

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171页的“4.8 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171页的“4.8 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 > WthIf(Wth) > 代入脉冲

WthIf (With If)

【功能】

在插补动作指令有附加条件的附加处理。

【格式】

例) Mov P1 WthIf □ <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 WthIf 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 WthIf 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 WthIf 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 202 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 □ < 插槽编号 >
```

【用語】

<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) 在指定的插槽内未选择程序的情况下，会发生错误 (L3370)。
- (2) 已指定插槽在运行中的情况下，会发生错误 (L3380)。
- (3) 已指定插槽在 中断的情况下，会发生错误 (L3380)。
- (4) 在始终执行的程序内使用的情况下，通过用参数“ALWENA”将值从 0 更改为 1，并将控制器的电源从 OFF 设为 ON，则变为有效。

【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 □ < 插槽编号 >, < 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，会发生错误（L4140）。
- (2) 其它的插槽中已选择其他程序的情况下，会发生错误（L3360）。
- (3) 指定的程序在编辑中的情况下，会发生错误（L3360）。
- (4) 指定的插槽在运行中的情况下，会发生错误（L3360）。
- (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) 如果在XLoad执行后立即执行XRun，因为程序在加载中可能会失败，所以必要的情况下，如例子的步号3所示，执行加载完成确认。

【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 □ < 插槽编号 >

【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及程序未选择时，会发生错误 (L3340、L3350)。
- (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 □ <槽号码>[, ["<程序名>"][, <运行模式>]]

【用语】

<槽号码>	指定1~32的槽号码。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，会发生错误（L4140）。
- (2) 指定的槽编号已经被使用的情况下，会发生错误（L3310）。
- (3) 在任务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) 如果在 XLoad 执行后立即执行 XRun，因为程序在加载中可能会失败，所以必要的情况下，如例子③的步号 2、例子④的步号 2 所示，执行加载完成确认。

**注意**

通过 XRun 指令指定的插槽中执行的程序结束时，数据将被写入至控制器内部的非易失性存储器。

重复执行 XRun 指令会增加控制器内部非易失性存储器的写入次数，可能会导致故障，因此应将参数 AUTOSAVE 设定为 0（不保存）。详细内容请参照“5.4 程序参数”中的 AUTOSAVE。

例：程序 1 至程序 2 重复循环运行。

```
< 程序 1 >  
1 *LOOP  
2 XRun 2, "2", 1  
3 Wait M_Run(2)=1  
4 End
```

```
< 程序 2 >  
1 PS=P_Curr  
2 End
```

【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) 未选择程序时，会发生错误（L3330）。在已经变成中断的情况下，执行结果不会失败。
- (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148页的“4.3.15 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(10)=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。

4.13 机器人（系统）状态变量的详细说明

4.13.1 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表记，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。

4.13.2 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\$ 中代入 “RH-3FRH5515-D”。
(插槽 1 获得机械 1 (“RH-3FRH5515-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_CoIMx1

【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_CoIMx1 [(<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。

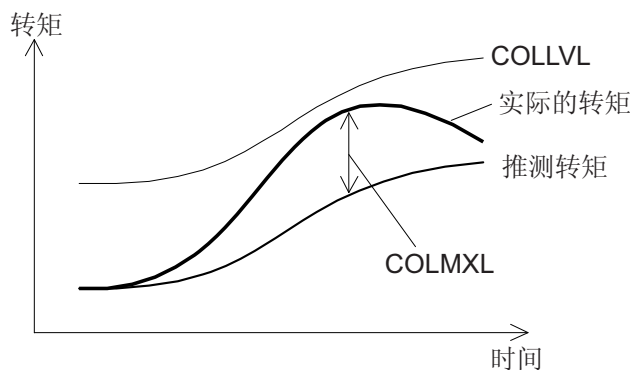
【文例】

```

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_CoIMx1 (1). J1+10 ' 计算使各轴都保有 10% 的剩余冲突容许 Level
52 M2=J_CoIMx1 (1). J2+10 (10% 为参考值, 并非实际的保证值)
53 M3=J_CoIMx1 (1). J3+10
54 M4=J_CoIMx1 (1). J4+10
55 M5=J_CoIMx1 (1). J5+10
56 M6=J_CoIMx1 (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】

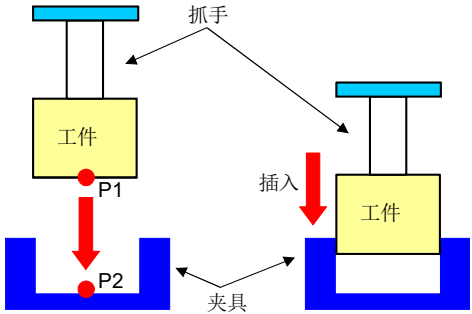
ColChk (Col Check)、ColLvl (ColLevel)

【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。

	<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>采样程序</p>	<p>说明（包含注释）</p>
<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>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 173 页的“4.12 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) J_AmpFbc 为相对额定电流的比例 (%)，以 0.1% 为单位。(1000=100%)。

(5) 读取专用。

【关连系统状态变量】

M_AmpInfoA、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_Ac1/M_DAc1/M_NAc1/M_NDAc1/M_Ac1Sts**【功能】**

返回加减速时间相关情报。

M_Ac1: 返回现在的加速时间比率。(%)

M_DAc1: 返回现在的减速时间比率。(%)

M_NAc1: 返回加速时间的初始值。(100%)

M_NDAc1: 返回减速时间的初始值。(100%)

M_Ac1Sts: 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_Ac1 [(<数式>)]

例) <数值变量> = M_DAc1 [(<数式>)]

例) <数值变量> = M_NAc1 [(<数式>)]

例) <数值变量> = M_NDAc1 [(<数式>)]

例) <数值变量> = M_Ac1Sts [(<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_Ac1 | ' 在 M1 代入在任务插槽 1 设定的加速时间比率。 |
| 2 M1=M_DAc1(2) | ' 在 M1 代入在任务插槽 2 设定的减速时间比率。 |
| 3 M1=M_NAc1 | ' 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_NDAc1(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_Ac1Sts(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_NAc1, M_NDAc1 通常会返回到 100(%)。
- (3) 读取专用。

M AmpInfoA

【功能】

以单位 Arms 回复当前的各轴的电流值信息。
 注) 本机器人状态变量可在控制器的软件版本为 C2 以上的情况下使用。

【格式】

例) <数值变量> = M_AmpInfoA (<信息 ID 号码>, <轴号码>)

【用语】

<数值变量> 指定代入的数值变量 (单精度实数)。
 <信息 ID 号码> 指定要读取的信息的信息 ID 号码。
 信息 ID 号码的详细内容, 请参照下列表格。

信息 ID 号码	
设定值	信息的内容
1	电流反馈值 (Arms)
2	最近 2 秒内的电流指令最大值 (Arms)
3	接通电源后的电流指令最大值 (Arms)

<轴号码> 对轴号码 1 ~ 6 进行指定。不可省略。

【例文】

- 1 M1=M_AmpInfoA (1, 1) ' 读取当前的 1 轴电流反馈值
- 2 M2=M_AmpInfoA (2, 2) ' 读取当前的 2 轴电流指令最大值 (最近 2 秒内)
- 3 M3=M_AmpInfoA (3, 3) ' 读取当前的 3 轴电流指令最大值 (接通电源后)

【说明】

- (1) 对信息 ID 号码和轴号码指定了超出设定范围的值时, 执行时会发生 L3110 错误。
- (2) 单位均为 Arms (实际值安培)。
- (3) 读取专用。

【关连系统状态变量】

[J_Fbc/J_AmpFbc](#)

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 EndIf
```

【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 指令指定基本转换数据, 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)、MEXSBS

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_BrkCq 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 568 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      ' 在检知到冲突的时候，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 行执行中检知到冲突的话，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)                    ' 制作退避动作的移动量
104 PDST=P_Fbc(1)+PESC                       ' 制作退避位置（在冲突的位置（P_Fbc）加算退避移动量）
105 Mvs PDST                                  ' 往退避位置移动
106 Error 9100                               ' 让使用者定义的低标准报警发生，运行停止。

```

【说明】

- (1) 在检知到冲突时会变成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 (参数的插槽表中设定了 CYC 时)
- 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。将此运行模式返回。
- (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 CR800-R/Q 系列使用此状态变量。

以字 (16 位) 为单位对 CC-Link 和 EtherCAT 的输入输出寄存器进行数据的参照和写入。

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> 指定将远程寄存器的值代入数值变量。

<数式 1 > 指定输入寄存器号码。(6000 ~)

<数式 2 > 指定输出寄存器号码。(6000 ~)

【例子】

- 1 M1=M_DIn(6000) ' 在 M1 中代入输入寄存器 6000 号的值。
- 2 M1=M_DOut(6000) ' 在 M1 中代入输出寄存器 6000 号的值。
- 3 M_DOut(6000)=100 ' 将 100 写入至输出寄存器 6000 号。

【说明】

- (1) M_DIn 为读取专用。

M DIn32

【功能】

从外部设备以 2 个字（32 位）为单位从 CC-Link 和 EtherCAT 的输入寄存器获取数据。

【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 >

可指定的寄存器号码的范围为 6000 ~ 6254。

【例文】

1 M1&=M_DIn32(6000)

' 取出输入寄存器号码 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% 中指定的输入寄存器号码开始的 32bit 的值，该值的低位 19bit 的值加入 M3& 中。

【说明】

- (1) 从指定的输入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，将无法获取信息。
- (3) 输入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 DOut32

【功能】

以 2 个字（32 位）为单位对 CC-Link 和 EtherCAT 的输出寄存器进行数据的写入。或以 2 个字为单位确认当前的输出信息。

【格式】

参照
 < 数值变量 >=M_DOut32(< 算式 / 输出寄存器号码 >)

写入
 M_DOut32(< 算式 / 输出寄存器号码 >)=< 数值 >

【用语】

< 数值变量（长精度整数）> 指定要代入的长精度整数的数值变量。
 < 算式 / 输出寄存器号码 > 可指定输出寄存器号码的范围为 6000 ~ 6254。
 < 数值 > 指定要输出的数据。
 数值的范围为 -2147483648 ~ 2147483647 (&H80000000 ~ &H7FFFFFFF)。

【例文】

1 M_DOut32(6000)=&H12345678 ' 向输出寄存器号码 6000 与 6001 中输出 16 进制表示的 12345678。
 向寄存器号码 6000 中输出 & H4567，向寄存器号码 6001 中输出 &H1234。

2 M1&=M_DOut32(6002) And &H7FFF ' 将从输出至输出寄存器号码 6002 与 6003 中的 32bit 数据开始的低位 19 位的值加入 M1& 中。

【说明】

- (1) 从指定的输出寄存器号码开始对 2 点输出指定的 32bit 数据。
- (2) 从指定的输出寄存器号码开始对 2 点获取当前的输出信息（32bit 数据）。
- (3) < 数值变量 > 与 < 数值 > 中应使用用于长精度整数的变量。其他情况下，有可能无法处理信息。
- (4) 输出寄存器号码应指定为 6000 ~ 6254。指定超出范围的号码的情况下，浏览时会全部返零，写入时不执行处理且不发生错误。但是，指定了负值时，会发生 L3110 错误。
- (5) 通过将参数 SYNCIO 设定为高速模式，可以加快寄存器输出的更新周期。为了提高性能，建议在采取互锁的基础上，以高速模式使用。
- (6) 指定的寄存器号码不是机器人占用的号码时，RT ToolBox3 等的寄存器输出监视上会显示输出的信息，但实际上并没有输出到网络。此外，指定了占用的最后的寄存器时，虽然监视上会显示输出的信息，但实际上只输出占用的最后的寄存器的信息（低位 16bit）。
- (7) 不支持组合了 Dly 指令的脉冲输出。使用 Dly 指令，会发生 L4220（所输入的指令语句的结构有错误）错误。
- (8) 输出至分配给参数 DIODATA 及 SVDATA 的专用输出的寄存器号码时，会发生 H0091（该信号已分配给专用输出）错误。

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。
 XXXXYYYYY.... 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】

返回机器人所发生关于报警的情报。

M_Err: 返回报警是否在发生中

M_ErrLvl: 返回已发生报警的 Level

M_ErrNo: 返回发生的报警号码

【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) 各术语和意义见表 4-17。

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 4-17: 错误等级及其意义

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、Remove

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 4-18: 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_GDev/M_GDevW/M_GDevD**【功能】**

通过 CR800-R/Q 系列的控制器写入 CPU 缓冲存储器或进行直接参照。
(仅限 CR800-R/Q 系列控制器)

M_GDev: 以位为单位 (1 位) 写入 / 参照。
M_GDevW: 以字为单位 (16 位) 写入 / 参照。
M_GDevD: 以双字为单位 (32 位) 写入 / 参照。

【格式】

参照

< 数值变量 >=M_GDev (< 起始输入输出编号 >, < 缓冲存储器地址 >, < 位编号 >)
< 数值变量 >=M_GDevW (< 起始输入输出编号 >, < 缓冲存储器地址 >)
< 数值变量 >=M_GDevD (< 起始输入输出编号 >, < 缓冲存储器地址 >)

写入

M_GDev (< 起始输入输出编号 >, < 缓冲存储器地址 >, < 位编号 >) =< 数值 >
M_GDevW (< 起始输入输出编号 >, < 缓冲存储器地址 >) =< 数值 >
M_GDevD (< 起始输入输出编号 >, < 缓冲存储器地址 >) =< 数值 >

【术语】

< 数值变量 >	指定要代入的数值变量。
< 起始输入输出编号 >	指定 CPU 模块的输入输出编号。 (指定的值为以 16 进制表示起始输入输出编号、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。
< 缓冲存储器地址 >	通过常数或数值变量指定 CPU 模块的缓冲存储器地址。 每个状态变量的有效范围如下所示: (10 进制) M_GDev: 10000 ~ 24335 M_GDevW: 10000 ~ 24335 M_GDevD: 10000 ~ 24334
< 位编号 >	通过常数或数值变量指定 CPU 模块的缓冲存储器的位编号。 10 进制表示时的设定范围如下所示。 M_GDev: 0 ~ 15
< 数值 >	通过常数或数值变量指定写入数据。 每个状态变量的有效范围如下所示: M_GDev: 0、1 M_GDevW: -32768 ~ 32767 (&H8000 ~ &H7FFF) M_GDevD: -2147483648 ~ 2147483647 (&H80000000 ~ &H7FFFFFFF)

【示例】

- 1 M_GDev(&H3E1, 10010, 2)=1 ' 将 2 号机 CPU 的缓冲存储器地址 10010 的位编号 2 的值设定为 1
- 2 M_GDevW(&H3E1, 10010)=&HFFFF ' 将 2 号机 CPU 的缓冲存储器地址 10010 的值设定为 FFFF
- 3 M_GDevD(&H3E1, 10011)=P1.X * 1000 ' 对 2 号机 CPU 的缓冲存储器地址 10011/10012 双字设定为位置变量 P1 的 X 坐标值扩大 1000 倍后的值
- 4 M1%=M_GDevW(&H3E2, 10001) And &H7 ' M1 中代入 3 号机 CPU 的缓冲存储器地址 10001 低位 3 位的值

【说明】

- (1) 向可编程控制器的缓冲存储器中写入或进行参照
- (2) 通过起始输入输出编号及缓冲存储器地址指定对象缓冲存储器。
- (3) 写入的数据或参照所返回的数据都是整数。
- (4) 从指定的缓冲存储器地址中，M_GDevW 以 1 字（16 位）的数据为对象，M_GDevD 以双字（32 位）的数据为对象。
- (5) 进行指定时，应指定起始输入输出编号以 16 进制表示时为 &H3E0 ~ &H3E3（以 10 进制表示时为 992 ~ 995）、缓冲存储器地址的写入或参照地址以 10 进制表示时为 10000 ~ 24335。
- (6) 仅本号机 CPU 的缓冲存储器地址可写入。即使指定其他号机 CPU 的地址来写入数据，也不会更新。
- (7) 要通过 M_GDevD 防止 32 位数据分离，请将缓冲存储器的偶数地址置于起始处进行访问。仅对应 CR800-R/Q 系列。（通过 CR800-D 系列执行时，不执行写入处理，参照返回 0。）

【补充】

表 4-19：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 例) P1.X	关节 例) J1.J1	字符串 例) C1\$
M_GDev	○	○	○	○	○	○	×
M_GDevW	○	○	○	○	○	○	×
M_GDevD	×	○	○	○	○	○	×

表 4-20：可填入〈起始输入输出编号〉〈缓冲存储器地址〉〈位编号〉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 例) 12	2 进制 例) &B1100	16 进制 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)} 例) P1.X	关节 ^{注 1)} 例) J1.J1	字符串 例) C1\$
M_GDev	○	○	○	○	○	○	○	○	○	×
M_GDevW	○	○	○	○	○	○	○	○	○	×
M_GDevD	○	○	○	○	○	○	○	○	○	×

○：可使用、×：不可使用


注 1) 实数值将四舍五入。

表 4-21: 可填入〈数值〉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 注1) 例) 12	2 进制 例) &B1100	16 进制 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	位置注1) 例) P1.X	关节注1) 例) J1.J1	字符串 例) C1\$
M_GDev	○	○	○	○	○	○	○	○	○	×
M_GDevW	○	○	○	○	○	○	○	○	○	×
M_GDevD	○注2)	○注2)	○注2)	○注2)	○	○	○注3)	○	○	×

○: 可使用、×: 不可使用

注 1) 实数值将四舍五入

注 2)  **注意** 指定的常数为储存在 2 进制的 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 位被扩展符号, 输出全部 ON 状态的信号。

例) 指定 -32768 (&B1000000000000000) 时
会输出 &B111111111111111100000000000000 (2 进制)

[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, 再代入到本状态变量 M_GDevD 后, 可输出 &B000000000000000010000000000000 (2 进制)。

- 1 M1&=-32768
- 2 M_GDevD(&H2)=M1&

注 3) 可输出的数值的范围为 -2147483648 ~ 2147483647。

M Gps

【功能】

高速位置获取功能（GPS 功能）中，返回以 Def Gps 指令定义的监视编号 P_Gps1 ~ P_Gps8 中存储的位置数据的个数。

【格式】

例) <数值变量>=M_Gps[(<数值>)]

【术语】

- <数值变量> 指定要代入的数值变量。
 返回<数值>中指定的监视编号对应的状态变量 P_Gps1 ~ P_Gps8（1 ~ 8 为监视编号）中存储的当前位置的个数。
 初始值为 0。此外，根据 GpsCchk On 指令为“0”。
- <数值> 指定以 Def Gps 指令定义的监视编号（1 ~ 8）。省略时为 1。

【示例】

- | | |
|---------------------------|------------------------------------|
| 1 Def Gps1, 851, 1 | ‘ 监视编号 1 记录 851 号为 ON 时机械 1 的位置数据。 |
| 2 GpsChk On, 1 | ‘ 开始监视编号 1 的条件监视 |
| 3 Mvs P1 | ‘ 向位置 P1 移动 |
| 4 GpsChk Off, 1 | ‘ 结束监视编号 1 的条件监视 |
| 5 M1=M_Gps(1) | ‘ 取得 P_Gps1 中存储的位置数据的个数 |
| 6 If M1=0 Then Error 9000 | ‘ 若位置数据无法记录，则发生错误 9000 |

【说明】

- (1) 通过 Def Gps 指令定义的条件成立时的机器人当前位置数据存储在 P_Gps1 ~ P_Gps8 中。存储在 P_Gps1 ~ P_Gps8 中的位置数据的个数存储在本状态变量中。
- (2) 可确认以 Def Gps 指令定义的条件成立的次数。
- (3) 根据 GpsChk On 指令数据将被清零。
- (4) 可在 GpsChk Off 指令发出后确认值。
- (5) 读取专用。
- (6) 重新接通电源时清零。

【相关指令】

Def Gps（Define get position）、Def Map、GpsChk（Get position check）

M_HGDev/M_HGDevW/M_HGDevD**【功能】**

以 CR800-R 系列的控制器写入 CPU 缓冲存储器（固定周期通信区域）或直接参照。

（仅限 CR800-R 系列控制器）

（可不用经由可编程控制器的梯形图程序，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 CPU 等机器人 CPU 以外的 CPU 缓冲存储器信息。（参照第 601 页的“在机器人 CPU 之间直接通信”））

M_HGDev: 以位为单位（1 位）写入 / 参照。

M_HGDevW: 以字为单位（16 位）写入 / 参照。

M_HGDevD: 以双字为单位（32 位）写入 / 参照。

【格式】

参照

<数值变量>=M_HGDev (<起始输入输出编号>, <缓冲存储器地址>, <位编号>)

<数值变量>=M_HGDevW (<起始输入输出编号>, <缓冲存储器地址>)

<数值变量>=M_HGDevD (<起始输入输出编号>, <缓冲存储器地址>)

写入

M_HGDev (<起始输入输出编号>, <缓冲存储器地址>, <位编号>) =<数值>

M_HGDevW (<起始输入输出编号>, <缓冲存储器地址>) =<数值>

M_HGDevD (<起始输入输出编号>, <缓冲存储器地址>) =<数值>

【术语】

<数值变量>

指定要代入的数值变量。

<起始输入输出编号>

指定 CPU 模块的输入输出编号。

（指定的值为以 16 进制表示起始输入输出编号、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。

<缓冲存储器地址>

通过常数或数值变量指定 CPU 模块的缓冲存储器地址。

每个状态变量的有效范围如下所示：（10 进制）

M_HGDev: 0 ~ 12287

M_HGDevW: 0 ~ 12287

M_HGDevD: 0 ~ 12286

<位编号>

通过常数或数值变量指定 CPU 模块的缓冲存储器的位编号。

10 进制表示时的设定范围如下所示。

M_HGDev: 0 ~ 15

<数值>

通过常数或数值变量指定写入数据。

每个状态变量的有效范围如下所示：

M_HGDev: 0、1

M_HGDevW: -32768 ~ 32767 (&H8000 ~ &H7FFF)

M_HGDevD: -2147483648 ~ 2147483647

(&H80000000 ~ &H7FFFFFFF)

【示例】

- | | |
|-----------------------------------|---|
| 1 M_HGDev(&H3E1, 30, 2)=1 | ‘将 2 号机 CPU 缓冲存储器地址 30 的位编号 2 的值设定为 1 |
| 2 M_HGDevW(&H3E1, 10)=&HFFFF | ‘将 2 号机 CPU 的缓冲存储器地址 10 的值设定为 FFFF |
| 3 M_HGDevD(&H3E1, 11)=P1.X * 1000 | ‘对 2 号机 CPU 的缓冲存储器地址 11/12 双字设定为位置变量 P1 的 X 坐标值扩大 1000 倍后的值 |
| 4 M1%=M_HGDevW(&H3E2, 1) And &H7 | ‘M1 中代入 3 号机 CPU 的缓冲存储器地址 1 低位 3 位的值 |

【说明】

- (1) 向可编程控制器的缓冲存储器（固定周期通信区域）中写入或进行参照。
- (2) 通过起始输入输出编号及缓冲存储器地址指定对象缓冲存储器。
- (3) 写入的数据或参照所返回的数据都是整数。
- (4) 从指定的缓冲存储器地址中，M_HGDevW 以 1 字（16 位）的数据为对象，M_HGDevD 以双字（32 位）的数据为对象。
- (5) 在进行指定时，应制定起始输入输出编号以 16 进制表示时为 &H3E0 ~ &H3E3（以 10 进制表示时为 992 ~ 995）；缓冲存储器地址的写入或参照地址以 10 进制表示时为 0 ~ 12287。
- (6) 仅本号机 CPU 的缓冲存储器地址可写入。即使指定其他号机 CPU 的地址来写入数据，也不会更新。
- (7) 仅对应 CR800-R 系列（通过 CR800-D/CR800-Q 系列执行时，不执行写入处理，参照返回 0）。
- (8) 要通过 M_HGDevD 防止 32 位数据分离，请将缓冲存储器的偶数地址置于起始处进行访问。

【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 例) P1.X	关节 例) J1.J1	字符串 例) C1\$
M_HGDev	○	○	○	○	○	○	×
M_HGDevW	○	○	○	○	○	○	×
M_HGDevD	×	○	○	○	○	○	×

表 4-23：可填入〈起始输入输出编号〉〈缓冲存储器地址〉〈位编号〉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 例) 12	2 进制 例) &B1100	16 进制 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)} 例) P1.X	关节 ^{注 1)} 例) J1.J1	字符串 例) C1\$
M_HGDev	○	○	○	○	○	○	○	○	○	×
M_HGDevW	○	○	○	○	○	○	○	○	○	×
M_HGDevD	○	○	○	○	○	○	○	○	○	×

○：可使用，×：不可使用


注 1) 实数值将四舍五入。

表 4-24: 可填入〈数值〉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 注1) 例) 12	2 进制 例) &B1100	16 进制 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	位置注1) 例) P1.X	关节注1) 例) J1.J1	字符串 例) C1\$
M_HGDev	○	○	○	○	×	×	×	×	×	×
M_HGDevW	○	○	○	○	×	×	×	×	×	×
M_HGDevD	○注2)	○注2)	○注2)	○注2)	○	○	○注3)	○	○	×

○: 可使用, ×: 不可使用

注 1) 实数值将四舍五入。

注 2)  **注意** 指定的常数为储存在 2 进制的 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 位被扩展符号, 输出全部 ON 状态的信号。

例) 指定 -32768 (&B1000000000000000) 时会输出 &B111111111111111100000000000000 (2 进制)

[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, 再代入到本状态变量 M_HGDevD 后, 可输出 &B000000000000000100000000000000 (2 进制)。

- 1 M1&=-32768
- 2 M_HGDevD(&H20)=M1&

注 3) 可输出的数值的范围为 -2147483648 ~ 2147483647。

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。
 CR800-D/R/Q: 1 ~ 8 (相当于输入信号 900 到 907)。
 CR860-D/R/Q: 1 ~ 8 (相当于输入信号 764 到 775)。

【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) CR800-D/Q/R 时, 相当于输入信号 900 号, 即使使用 M_In(900) 也会得到相同结果。
CR860-D/Q/R 时, 相当于输入信号 764 号, 即使使用 M_In(764) 时也会得到相同结果。
- (3) 读取专用。

M_In/M_Inb/M_In8/M_Inw/M_In16/M_In32

【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。
 M_In32: 以双字为单位 (32 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>)
 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 4-25。
 <数式> 指定信号号码。补充内容见表 4-26。
 (1)CR800-R/Q 系列
 1000 ~ 18191: 多 CPU 间共有设备
 764 ~ 775: 抓手 (CR860)
 900 ~ 907: 抓手 (CR800)
 (2)CR800-D 系列
 0 ~ 255: 并行输入输出
 764 ~ 775: 抓手 (CR860)
 900 ~ 907: 抓手 (CR800)
 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。
- 5 M5%=M_In32(16000) ' 在 M5 中输入从输入信号 16000 号起 32 位量的值。

【说明】

- (1) 返回输入信号的状态。
- (2) 从指定的信号号码开始，分别 M_In 返回 1 位、M_Inb/M_In8 返回 8 位、M_Inw/M_In16 返回 16 位、M_In32 返回 32 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) M_In32 时，应将 <数值变量> 指定为长精度整数型或实数型的变量。
- (5) 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 4-25: <数值变量> 中可记述的变量 ○: 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	○	○	○	○	○	○	×
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)，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 4-26: <数式> 中可记述的常数・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	○	○	○	○	○	○	○	○	○	×
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。

【相关指令】

Def I0 (Define I0)

M_JOvrđ/M_NJOvrđ/M_OPOvrđ/M_Ovrđ/M_NOvrđ**【功能】**

返还速度比例值

M_JOvrđ: 关节插补用的速度比例以指令 JOvrđ 指定的值

M_NJOvrđ: 关节插补用的速度比例的初始值 (100%)

M_OPOvrđ: 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_NOvrđ: 速度比例的初始值 (100%)

【格式】

例) <数值变量> = M_JOvrđ [(<数式>)]

例) <数值变量> = M_NJOvrđ [(<数式>)]

例) <数值变量> = M_OPOvrđ

例) M_OPOvrđ = <算式>

例) <数值变量> = M_Ovrđ [(<数式>)]

例) <数值变量> = M_NOvrđ [(<数式>)]

【用语】

<数值变量>

指定代入的数值变量。

<算式>

在 M_JOvrđ/M_NJOvrđ/M_Ovrđ/M_NOvrđ 算式中, 加入 1 ~ 32、任务插槽号码。省略时为当前的插槽。

在 M_OPOvrđ 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_NOvrđ

' 在 M2 输入现在的速度比例的初始值 (100%)。

3 M3=M_JOvrđ

' 在 M3 输入现在的关节速度比例值。

4 M4=M_NJOvrđ

' 在 M4 输入现在的关节速度比例的初始值。

5 M5=M_OPOvrđ

' 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_OPOvrđ=20

' OP (操作面板) 倍率修调中设定为 20。

【说明】

(1) 在自变量省略的情况下, 返还现在的插槽状态。

(2) M_JOvrđ/M_NJOvrđ/M_Ovrđ/M_NOvrđ 仅可读取。

M_OPOvrđ 可读取、写入。

M Line

【功能】

返回现在正执行的步号。

【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_SetAd1(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厉的姿势待机、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 Re1M 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

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。上述例文中，负载率超过 90% 时发生报警 9101 的示例。
- (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 RelM(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 Map1 ~ M Map8**【功能】**

高速位置获取功能（GPS 功能）中，返回通过 Def Gps 指令定义的条件下的工件存在段数。

可确认通过映射处理（通过滑动盒前安装在抓手上的传感器，将存储传感器有反应的盒段数中存在工件的信息）的实施取得的工件的有无信息。

【格式】

例) <数值变量>=M_Map1[(<数值>)]

【术语】

<数值变量> 指定要代入的数值变量。
 将<数值>中指定的段数的工件有无信息返回到位置变量中。
 初始值为 0。存在工件的段数中存储 1。
 此外，根据 GpsChk On 指令清零。

<数值> 指定要确认有无工件的盒的段数。
 将盒段数的最下段定义为第 1 段。
 设定范围：1 ~ 130（段）（指定 0 或负的值时，作为 1 进行处理。）

【示例】

```

1 Def Map 3, 851, On, 1, PC1, PC2, 20, 10
   ‘ 监视编号 3 记录 851 号为 ON 时机械 1 的位置数据，并根据已定义
   ‘ 的条件执行映射处理。
   PC1: 表示盒 第 1 段的位置、PC2: 表示盒最上段的位置、20: 20
   ‘ 段平均分割、10: 传感器灵敏度 10mm 宽
2 Mov PM1
   ‘ 移动至映射开始位置
3 GpsChk On, 3
   ‘ 开始监视编号 3 的条件监视。记录 851 号为 ON 时的位置数据的同
   ‘ 时，计算从该位置数据开始的工件存在的段数并通过 M_Map3
   ‘ (130) 表现。
4 Mvs PM2
   ‘ 移动至映射结束位置
5 GpsChk Off, 3
   ‘ 结束监视编号 3 的条件监视
6 M1=M_Gps(3)
   ‘ 取得 P_Gps3 中存储的位置数据的个数
7 If M1=0 Then Error 9000
   ‘ 若位置数据无法记录，则发生错误 9000
8 For M2=1 To 20
9 M_Out(6100+M2) = M_Map3(M2)
   ‘ 从输出信号 6101 号按顺序输出映射结果。
10 Next M2
   ‘ 段数部分重复

```

【说明】

- (1) 通过 Def Map 指令定义的条件成立时的机器人当前位置数据收存储在 P_Gps1() ~ P_Gps8() 中。根据以该信息与 Def Map 指令定义的条件，将工件存在于哪段存储至本状态变量中。
- (2) 根据输出信号将本状态变量的信息输出至外部装置中，由此可将工件的位置通知至外部装置。
- (3) 通过 GpsChk On 指令数据将被清零。
- (4) 可确认 GpsChk Off 指令后的值。
- (5) 读取专用。
- (6) 重新接通电源时清零。

【相关指令】

Def Map, GpsChk (Get position check)

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 Mxt

【功能】

进行协调动作的主站侧与从站侧机器人之间，用于与机器人应用程序同步。

【格式】

例) <数值变量>=M_Mxt (<对象号机>)

【术语】

<数值变量> 指定要代入的数值变量。
 <对象号机> 代入主站侧机器人的 CPU 号机编号。2 ~ 4。

【示例】

M_Mxt(2)=1 ' 通过多 CPU2 号机开始协调。
 Wait M_Mxt(2)=1 ' 等待多 CPU2 号机开始协调。

【说明】

- (1) 通过协调动作功能用于与机器人应用程序同步。
- (2) 仅本号机可读取 / 写入。本号机以外为读取专用。
- (3) 协调动作中，通过主站侧机器人将本状态变量从 1 变为 0，则从站侧机器人将自动结束 Mxt 指令。
- (4) 电源接通后的初始值为 0。

M_NvOpen

【功能】

表示视觉传感器用的线路连接状态。

【数组的意义】

数组元素 (1 ~ 8): 视觉传感器编号

【返回值的说明】

0: 线路连接中 (未登录) 1: 登录完成 -1: 未连接

【使用方法】

NVOpen 指令执行后, 视觉传感器与线路连接, 确认是否在视觉传感器中登录。

【程序示例】

```

100 If M_NvOpen(1)<>1 Then ' 视觉传感器编号 1 未登录完成时
110 NVOpen "COM2:" As #1 ' 与连接至 COM2: 的视觉传感器连接, 并将编号设为 1 号。
120 End If
130 Wait M_NvOpen(1)=1 ' 与视觉传感器编号 1 连接, 等待至登录完成
140 . . . . .
300 NVClose #1 ' 切断与 COM2: 连接的视觉传感器的线路。

```

【说明】

- 1) 通过 NVOpen 指令显示与网络视觉传感器连接的线路打开时的线路状态。
- 2) 初始值为“-1”。执行 NVOpen 指令, 线路连接时值为“0” (线路连接中), 网络视觉传感器登录完成时为“1” (登录完成)。
- 3) 本变量虽然和状态变量 M_Open 的状态相似, 但 M_Open 在连接确认完成时变为“1”, M_NvOpen 则要在视觉传感器登录完成后变为“1”。

【错误】

- (1) 数组元素中指定数据型不同时, 将发生“输入的指令语句的语法有误”错误。
- (2) 数组元素的个数异常 (多 / 少) 时, 将发生“变量的类型不同”错误。
- (3) 数组元素指定为“1” ~ “8” 以外时, 将发生“数组元素有误”错误。

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 Open

【功能】

返回已指定的文件或通信线路的开启状态。根据以 Open 指令所指定的文件种类，返回的值会有所不同。

【格式】

例) <数值变量> = M_Open [(<文件号码>)]

【用语】

<数值变量> 指定代入的数值变量。
 <文件号码> 以 Open 指令所开启的文件号码以 1 ~ 8 的常数指定。省略时文件号码会变成 1。执行 9 以上时会发生报警。

【例子】

1 Open "temp.txt" For Append As #1 ' 将 "temp.txt" 作为文件 1 开启
 2 *LBL:If M_Open(1)<>1 Then GoTo *LBL ' 等待文件安号码 1 到开启为止

【说明】

- (1) 读取专用。
- (2) 依据以 Open 指令所指定的文件种类的返还值如下所示。

Open 指令执行时的文件种类	意义		值
文件	还文件被开启时或文件没被开启的状态。 Open 指令执行后，在 Close 指令、End 指令、程序的 End 被执行为止，将 1 返还。		1 : 开启 -1 : 文件号码未定义 (未开启)
通信线路 以太网	表示是否连接至对象。	服务器设定时	1: 客户端已连接 0: 客户端未连接 -1: 文件编号未定义 (未 OPEN)
		客户机设定时	1: 已连接服务器 (已确定连接) 0: 未连接服务器 (未确定连接。OPEN 后，服务器相当于死机时) -1: 文件编号未定义 (未 OPEN 或在服务器死机状态下 OPEN 时)

【相关指令】

Open

M_Out/M_Outb/M_Out8/M_Outw/M_Out16/M_Out32**【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)

M_Out32: 输出信号的双字 (32 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_Out32(< 数式 1>)=< 数值 5>
 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 4-27。

(1) CR800-R/Q 系列

10000 ~ 18191: 多 CPU 间共享软元件

764 ~ 771: 抓手 (CR860)

9900 ~ 907: 抓手 (CR800)

(2) CR800-D 系列

0 ~ 255: 并行输入输出

764 ~ 771: 抓手 (CR860)

900 ~ 907: 抓手 (CR800)

2000 ~ 5071: PROFIBUS

6000 ~ 8047: CC-Link

< 数值 2 >, < 数值 3 >, < 数值 4 >, < 数值 5 >

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 4-28。

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)

< 数值 5 >: -2147483648 ~ +2147483647 (H80000000 ~ &H7FFFFFFF)

< 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 4-29。

【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。 |
| 5 M5=M_Out32(16000) | ' 在 M5 中输入从输出信号 16000 开始的 32 位置的值。 |

【说明】

- (1) 在写入到外部输出信号或进行参照时使用。
- (2) 从指定的信号号码开始, 分别 M_Out 输出 1 位、M_Outb/M_Out8 输出 8 位、M_Outw/M_Out16 输出 16 位、M_Out32 输出 32 位置的数据。
- (3) 关于脉冲输出请参照第 221 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 482 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 4-27: < 数式 1 > 中可记述的常数・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)I2	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	○	○	○	○	○	○	○	○	×	×
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) 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 4-28: < 数值 2 >、< 数值 3 >、< 数值 4 >、< 数值 5 > 中可记述的常数・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)I2	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	○	○	○	○	×	×	×	×	×	×
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 0N 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: <数值变量> 中可记述的变量

○: 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_Out	○	○	○	○	○	○	×
M_Outb/M_Out8	○	○	○	○	○	○	×
M_Outw/M_Out16	○	○	○	○	○	○	×
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), 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。

【相关指令】

Def I0 (Define I0)

【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 Ratio

【功能】

返还往机器人移动中的目的位置的到达率 (0 ~ 100%)。

【格式】

例) <数值变量> = M_Ratio [(<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 WthIf M_Ratio>80, M_Out(1)=1
' 在往 P1 移动中, 到目的位置的 80% 的距离的话, 将输出信号 1 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 RCInfo

【功能】

返回机器人控制器的各种信息。

【格式】

例) <数值变量> = M_RCInfo(<信息 ID 号码>, <要素号码>)

【用语】

- <数值变量> 指定要代入的数值变量。
- <信息 ID 号码> 指定要读取的信息的信息 ID 号码。
信息 ID 号码的详细内容, 请参照下列表格。
- <要素号码> 根据要读取的信息指定要素号码。
要素号码的详细内容, 请参照下列表格。

信息 ID 号码		要素号码	
设定值	信息的内容	设定值 (设定范围)	含义
1	各轴编码器温度 (°C)	1 ~ 8	表示机器人的轴号码。 ※ 不存在的轴以及附加轴 (7、8 轴) 时, 始终返回 0。
2	各轴编码器最大温度 (°C)	1 ~ 8	
3	控制器柜内温度 (°C)	1	-

【例文】

```
1 Dim METMP(8)                            ' 代入各轴编码器温度的变量
2 Dim METMPX(8)                          ' 代入各轴编码器最大温度的变量
3 For MT=1 To 8
4   METMP(MT)=M_RCInfo(1, MT)           ' 读取各轴编码器温度
5   METMPX(MT)=M_RCInfo(2, MT)         ' 读取各轴编码器最大温度
6 Next
7 MRCTMP=M_RCInfo(3, 1)                 ' 读取控制器柜内温度
```

【说明】

- (1) 对信息 ID 号码和要素号码指定设定了范围以外的值时, 执行时会发生 L3110 错误。

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 WthIf 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 Ovrld 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、Ovrld、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 WthIf 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。

【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。

【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 (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。可以使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)      ' M1 中, 从当前位置向 P1 移动, 并且代入已向 P2 移动的时间 (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】

指定的号码的工具转换数据 (MEXTL1 ~ 4) 可以作为现在的工具转换数据使用的同时, 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工具转换数据的设定

```
1 Tool (0, 0, 100, 0, 0, 0)     ' 指定工具转换数据 (0, 0, 100, 0, 0, 0), 在 MEXTL 写入。
2 Mov P1
3 M_Tool=2     ' 将工具转换数据变更为 TOOL 号码 2 (MEXTL2) 的值。
4 Mov P2
```


②TOOL 号码的参照

```
1 If M_In(900)=1 Then     ' 依据抓手输入信号切换工具转换数据。
2   M_Tool=1     ' 将 TOOL1 设定到工具转换数据。
3 Else
4   M_Tool=2     ' 将 TOOL2 设定到工具转换数据。
5 EndIf
6 Mov P1
```

【说明】

- (1) 将 TOOL 用的参数设定在 MEXTL1、MEXTL2、MEXTL3、MEXTL4 的值反映在工具转换数据, 也在参数 MEXTL 里写入。
 使用连续反复执行 Base 指令、Tool 指令、M_Tool 的程序时, 可能会无法即时进行参数保存, 从而发生错误 C7091 (参数保存失败), 应加以注意。发生错误 C7091 时, 应如下所示变更执行 Base 指令、M_Tool 的位置。

例)

<pre>*MAIN Base PB M_Tool=2 If M_In(20)=1 Then GoSub *SUB1 EndIf GoTo *MAIN</pre>		<pre>Base PB M_Tool=2 *MAIN If M_In(20)=1 Then GoSub *SUB1 EndIf GoTo *MAIN</pre>
---	---	---

- (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 以外的工具转换数据会被视为现在的工具转换数据设定。
- (5) 向工具编号中写入 0 后, 当前的工具参数 MEXTL 将被复位为初始值 (0, 0, 0, 0, 0, 0)。在设定为 0 时不想对当前的工具参数进行复位而是想要保持时, 应将参数 TOOLSPEC 从初始值 0 变更为 1 后, 再次接通电源。
 需要控制器的软件版本为 A4 版以上。
- (6) 示教单元的 TOOL 设定画面也可以做同样的设定。详细内容请参照第 33 页的“3.2.9 工具转换数据的切换”。

【关连参数】

MEXTL、MEXTL1 ~ 16、TOOLSPEC

【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) 用户定义区域的使用方法请参照第 503 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】

返还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) 用户定义区域的使用方法请参照第 503 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 CR800-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 CR800-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 601 页的“5.26 在机器人 CPU 之间直接通信”））

M\_UDevW: 以字为单位（16 位置）写入 / 参照。  
 M\_UDevD: 以双字为单位（32 位置）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】**

|                                                                                                                                                 |                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1 M_UDevW(&amp;H3E1, 10010)=&amp;HFFFF</p> <p>2 M_UDevD(&amp;H3E1, 10011)=P1.X * 1000</p> <p>3 M1%=M_UDevW(&amp;H3E2, 10001) And &amp;H7</p> | <p>’ 在 2 号机 CPU（本号机）的共有内存地址 10010 中写入 &amp;HFFFF（16 进制数）</p> <p>’ 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</p> <p>’ 在 M1 中代入 3 号机 CPU 的共有内存地址 10001 的低位 3 位的值</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**【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。仅对应 CR800-Q 系列。（通过 CR800-D/CR800-R 系列执行时，不执行写入处理，参照返回 0。）
- (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 4-37：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_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 4-38：可填入〈PLC 输入信号号码〉〈共有内存地址〉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>注1),注2)<br>例)P1.X | 关节<br>注1),注2)<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〈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 <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\_UDevD 后，可输出 &B00000000000000001000000000000000（2 进制数）。

1 M1&=-32768

2 M\_UDevD(&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 4-41 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。

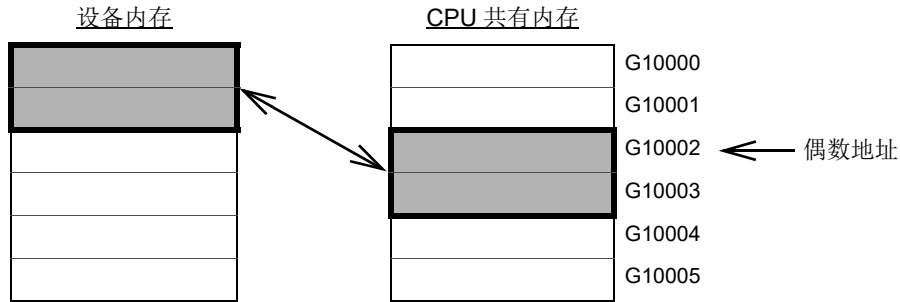


图 4-41：防止 32 位数据的分离

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】

返回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）关于暖机运行模式的详细内容请参照第 527 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 (Ovrld 指令)) × (关节速度比例 (JOvrld 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 (Ovrld 指令)) × (直线指定速度 (Spd 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）关于暖机运行模式的详细内容请参照第 527 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）关于暖机运行模式的详细内容请参照第 527 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 CR800-R/Q 系列的控制器中参照 PLC 输入信号 (X) 的值。(仅限 CR800-R/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 570 页的“5.25 可编程控制器输入输出模块直接控制”))

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 ~ &HFEE (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 4-40: 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 4-41: 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>注1), 注2)<br>例)P1.X | 关节<br>注1), 注2)<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 CR800-R/Q 系列的控制器向 PLC 输出信号 (Y) 写入或进行参照。(仅限 CR800-R/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 570 页的“5.25 可编程控制器输入输出模块直接控制”))

M\_YDev : 以位为单位写入 / 参照。  
M\_YDevB: 以字节为单位 (8 位置) 写入 / 参照。  
M\_YDevW: 以字为单位 (16 位置) 写入 / 参照。  
M\_YDevD: 以双字为单位 (32 位置) 写入 / 参照。

**【格式】**

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>参照</p> <p>&lt;数值变量&gt; = M_YDev [ (&lt; PLC 输出信号号码 &gt; ) ]</p> <p>&lt;数值变量&gt; = M_YDevB [ (&lt; PLC 输入信号号码 &gt; ) ]</p> <p>&lt;数值变量&gt; = M_YDevW [ (&lt; PLC 输入信号号码 &gt; ) ]</p> <p>&lt;数值变量&gt; = M_YDevD [ (&lt; PLC 输入信号号码 &gt; ) ]</p> <p>写入</p> <p>M_YDev [ (&lt; PLC 输入信号号码 &gt; ) ] = &lt;数值&gt;</p> <p>M_YDevB [ (&lt; PLC 输入信号号码 &gt; ) ] = &lt;数值&gt;</p> <p>M_YDevW [ (&lt; PLC 输入信号号码 &gt; ) ] = &lt;数值&gt;</p> <p>M_YDevD [ (&lt; PLC 输入信号号码 &gt; ) ] = &lt;数值&gt;</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**【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 ~ &HFEO (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。<br>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 位置                              |
| 3 M_YDevW (&H20)=&HFFFF        | ' 开启从 PLC 输出信号 20 号 (16 进制数) 开始的 16 位置                             |
| 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 位置中 |
| 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。

- (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 D1y 指令输出脉冲。使用 D1y 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 4-42: 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& | 单精度实数<br>例)M1! | 双精度实数<br>例)M1# | 位置注1)<br>例)P1.X | 关节注1)<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 4-43: 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& | 单精度实数注1)<br>例)M1! | 双精度实数注1)<br>例)M1# | 位置注1),注2)<br>例)P1.X | 关节注1),注2)<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 4-44: 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)12<br>注1) | 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&amp;=-32768

2 M\_YDevD(&amp;H20)=M1&amp;

注 3) 可输出数值的范围为 -2147483648 ~ 2147483647。

## P Base/P NBase

### 【功能】

返还基本转换数据相关情报。

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转换数据。  
2 Base P\_NBase ' 将基本转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 568 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) 以 (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)。(“\*”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】

1 P1=P\_CordR      ' 在 P1 中输入从通用坐标到本机器人基本坐标为止的位置。

### 【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, 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】

1 P1=P\_CurrR            ' 在 P1 中输入从通用坐标到本机器人基本坐标为止的位置。

### 【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 4-45: 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\_GCurr**【功能】**

在进行协调动作的主站侧与从站侧机器人之间，用于读出协调动作开始时主站侧机器人当前位置的系统状态变量。在开始协调动作前，读出主站侧机器人的当前位置并对准动作开始位置时使用。  
作为事先准备，需要将 iQ 扩展功能设为有效并将对象号机的当前位置设为可监视状态。

**【格式】**

|                              |
|------------------------------|
| 例) <位置变量>=P_GCurr[ (<对象号机>)] |
|------------------------------|

**【术语】**

<位置变量>            指定代入的位置变量。  
<对象号机>            输入主站侧机器人的 CPU 号机编号。2 ~ 4。

**【示例】**

Open "MXT:QRBUS2" As #1        ' 作为主站侧机器人，指定 CPU 缓冲存储器的多 CPU 2 号机  
Mov P\_GCurr(2)                ' 移动至与多 CPU2 号机相同的作业开始位置

**【说明】**

- (1) 用于通过协调动作功能取得应用程序的开始位置。
- (2) 读取专用。
- (3) 输入读出本变量时的对象 CPU 号机的当前位置。  
指定了 iQ 扩展功能无效的对象号机时，全部值始终为 0（与 P\_Zero 相同）。
- (4) 代入本状态变量的值会保持到断开电源为止。与 Open 指令无关可读出。

## P\_GDev

### 【功能】

通过 CR800-R/Q 系列的控制器，向 CPU 缓冲存储器写入位置数据，或直接进行参照。（仅限 CR800-R/Q 系列控制器）

以位置数据单位（32 位 × 10 量）写入 / 参照。

### 【格式】

参照  
 < 位置变量 >=P\_GDev (< 起始输入输出编号 >, < 缓冲存储器地址 >)

写入  
 P\_GDev (< 起始输入输出编号 >, < 缓冲存储器地址 >) =< 位置数据 >

### 【术语】

- |              |                                                                                                                                                                                                                                     |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| < 位置变量 >     | 指定代入的位置变量。                                                                                                                                                                                                                          |
| < 起始输入输出编号 > | CPU 模块的输入输出编号以常数或数值变量指定。<br>（指定的值为以 16 进制表示起始输入输出编号、且省略低位 1 位的值）<br>范围：16 进制表示时为 &H3E0 ~ &H3E3（10 进制表示时为 992 ~ 995）<br>1 号机：&H3E0（10 进制表示为 992）<br>2 号机：&H3E1（10 进制表示为 993）<br>3 号机：&H3E2（10 进制表示为 994）<br>4 号机：&H3E3（10 进制表示为 995） |
| < 缓冲存储器地址 >  | 通过常数或数值变量指定 CPU 模块的缓冲存储器地址。<br>有效范围为 0 ~ 524268（10 进制）。                                                                                                                                                                             |
| < 位置数据 >     | 指定写入的位置数据。<br>可通过常数、变量、逻辑 / 算术式及函数指定。                                                                                                                                                                                               |

### 【示例】

- 1 P\_GDev(&H3E1, 10)=P\_Curr    ’ 在当前位置设定 2 号机 CPU 的缓冲存储器地址 10 的值。  
 2 P1=P\_GDev(&H3E2, 1)        ’ 将 3 号机 CPU 的缓冲存储器地址 1 的位置数据代入 P1。

### 【说明】

- (1) 向可编程控制器的 CPU 缓冲存储器写入位置数据，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) 通过起始输入输出编号及缓冲存储器地址指定对象缓冲存储器。
- (3) 将已指定的缓冲存储器地址中 20 字（32 位 × 10）量的数据作为对象。
- (4) 在进行指定时，起始输入输出编号以 16 进制表示为 &H3E0 ~ &H3E3（10 进制表示为 992 ~ 995），缓冲存储器地址的写入或参照地址以 10 进制表示为 0 ~ 524268。
- (5) 仅自号机 CPU 的缓冲存储器地址可写入。即使指定其他号机 CPU 的地址写入数据，也将不会更新。
- (6) 仅对应 CR800-R/Q 系列（通过 CR800-D 系列执行时，不执行写入处理，参照返回 0）。

### 【补充】

表 4-46: < 可编程控制器输入信号编号 > < 缓冲存储器地址 > 中可记述的常数 · 变量

| 位宽   | 常数的类型               |                      |                 | 数值变量的类型      |                     |                         |                        | 其他变量              |                    |                |
|------|---------------------|----------------------|-----------------|--------------|---------------------|-------------------------|------------------------|-------------------|--------------------|----------------|
|      | 数值<br>注 1)<br>例) 12 | 2 进制<br>例)<br>&B1100 | 16 进制<br>例) &HC | 整数<br>例) M1% | 长精度<br>整数<br>例) M1% | 单精度<br>实数注 1)<br>例) M1& | 双精度<br>实数注 1)<br>例) M1 | 位置注 1)<br>例) P1.X | 关节注 1)<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。

P Gps1 ~ P Gps8**【功能】**

通过高速位置获取功能（GPS 功能），用直交坐标返回 Def Gps 指令定义的条件成立时的位置数据。（最多可保存 400 个位置数据。）

可确认以 Def Gps 指令定义的条件成立时的机器人的位置数据。（可通过 M\_Gps 确认以 Def Gps 指令定义的条件成立时的个数。）

**【格式】**

例) <位置变量>=P\_Gps1[ (<数值>)]

**【术语】**

- <位置变量> 指定代入的位置变量。  
 <位置变量>中返回<数值>中指定的位置编号对应的位置数据。  
 初始值为 P\_Zero（全部要素为 0）。此外，通过 GpsChk On 指令被清零，GpsChk Off 后条件成立时的位置数据将被收纳。
- <数值> 指定 GPS 功能中取得的位置编号。（1 ~ 400）  
 根据通过 Def Gps 指令定义的条件成立的顺序，按照 1、2、3... 的顺序指定位置编号。（初次取得成立的位置数据时，设定 1）  
 省略时为 1。

**【示例】**

- |                           |                                    |
|---------------------------|------------------------------------|
| 1 Def Gps1, 852, On, 1    | ‘ 监视编号 1 记录 852 号为 ON 时的机器 1 的位置数据 |
| 2 GpsChk On, 1            | ‘ 开始监视编号 1 的条件监视                   |
| 3 Mvs P1                  | ‘ 向位置 P1 移动                        |
| 4 GpsChk Off, 1           | ‘ 结束监视编号 1 的条件监视                   |
| 5 M1=M_Gps(1)             | ‘ 取得 P_Gps1 中存储的位置数据的个数            |
| 6 If M1=0 Then Error 9000 | ‘ 若不能记录位置将发生错误 9000                |
| 7 Mvs P_Gps1(1)           | ‘ 向 852 号首次为 ON 的位置移动。             |
| 8 Hlt                     | ‘ 停止                               |

**【说明】**

- (1) 以 Def Gps 指令定义的条件成立时的机器人当前位置数据将被收纳至本状态变量。Def Gps 指令的<监视编号>为 1 且该条件成立 3 次时，P\_Gps1 (1)、P\_Gps1 (2)、P\_Gps1 (3) 中机器人的位置数据将被保存。最初取得的位置数据将被收纳至 P\_Gps1 (1)。
- (2) 最多可保存 400 个位置数据（P\_Gpsn (1) ~ P\_Gpsn (400)）。无法取得 400 个以上的位置数据。
- (3) 返回排列要素中指定的数值的位置数据。
- (4) 通过 GpsChk On 指令数据将被清零。
- (5) 可确认 GpsChk Off 指令后的值。
- (6) 读取专用。
- (7) 重新开启电源时清零。

**【相关指令】**

Def Gps (Define get position)、Def Map、GpsChk (Get position check)

P\_HGDev

## 【功能】

C 通过 CR800-R 系列的控制器，向 CPU 缓冲存储器（固定周期通信区域）写入位置数据，或直接进行参照。（仅限 CR800-R 系列控制器）  
以位置数据单位（32 位 × 10 量）写入 / 参照。

## 【格式】

参照

< 位置变量 >=P\_HGDev (< 起始输入输出编号 >, < 缓冲存储器地址 >)

写入

P\_HGDev (< 起始输入输出编号 >, < 缓冲存储器地址 >) =< 位置数据 >

## 【术语】

&lt; 位置变量 &gt;

指定代入的数值变量。

&lt; 起始输入输出编号 &gt;

CPU 模块的输入输出编号以常数或数值变量指定。

（指定的值为以 16 进制表示起始输入输出编号、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：&amp;H3E0（10 进制表示为 992）

2 号机：&amp;H3E1（10 进制表示为 993）

3 号机：&amp;H3E2（10 进制表示为 994）

4 号机：&amp;H3E3（10 进制表示为 995）

&lt; 缓冲存储器地址 &gt;

通过常数或数值变量指定 CPU 模块的缓冲存储器地址。

有效范围为 0 ~ 12268（10 进制）。

&lt; 位置数据 &gt;

指定写入的位置数据。

可通过常数、变量、逻辑 / 算术式及函数指定。

## 【示例】

1 P\_HGDev(&H3E1, 10)=P\_Curr ' 在当前位置设定 2 号机 CPU 的缓冲存储器地址 10 的值。

2 P1=P\_HGDev(&H3E2, 1) ' 将 3 号机 CPU 的缓冲存储器地址 1 的位置数据代入 P1。

## 【说明】

- 向可编程控制器的 CPU 缓冲存储器（固定周期通信区域）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。
- 通过起始输入输出编号及缓冲存储器地址指定对象缓冲存储器。
- 将已指定的缓冲存储器地址中 20 字（32 位 × 10）量的数据作为对象。
- 在进行指定时，起始输入输出编号以 16 进制表示为 &H3E0 ~ &H3E3（10 进制表示为 992 ~ 995），缓冲存储器地址的写入或参照地址以 10 进制表示为 0 ~ 12268。
- 仅自号机 CPU 的缓冲存储器地址可写入。即使指定其他号机 CPU 的地址写入数据，也将不会更新。
- 参照时对应的 CPU 缓冲存储器地址未通过（参数：QMLTCPUn）分配时值返回 0。仅对应 CR800-R 系列。（通过 CR800-D/CR800-Q 系列执行时，不执行写入处理，参照返回 0。）

## 【补充】

表 4-47：&lt; 可编程控制器输入信号编号 &gt; &lt; 缓冲存储器地址 &gt; 中可记述的常数 · 变量

| 位宽   | 常数的类型               |                   |                 | 数值变量的类型      |                     |                             |                            | 其他变量              |                    |                |
|------|---------------------|-------------------|-----------------|--------------|---------------------|-----------------------------|----------------------------|-------------------|--------------------|----------------|
|      | 数值<br>注 1)<br>例) 12 | 2 进制<br>例) &B1100 | 16 进制<br>例) &HC | 整数<br>例) M1% | 长精度<br>整数<br>例) M1% | 单精度<br>实数<br>注 1)<br>例) M1& | 双精度<br>实数<br>注 1)<br>例) M1 | 位置注 1)<br>例) P1.X | 关节注 1)<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。

## P\_Safe

**【功能】**

向退避点位置（参数 JSafe 的直交位置）写入位置数据，或进行参照。

**【格式】**

参照

<位置变量> = P\_Safe [( <机器号码> )]

写入

P\_Safe[( <机械编号 > )] = <位置数据 >

**【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。               |
| <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 P1=P_Safe   | ' P1 中代入设定的退避点位置。 |
| 2 P_Safe = P2 | ' 将退避点位置更改为 P2。   |

**【说明】**

- (1) 将登录在参数 JSafe 中的关节位置转换为直交位置并返回。
- (2) 进行了写入时，参数 JSafe 的值也将改写。



## P Tool/P NTool

### 【功能】

返回工具转换数据。

P\_Tool: 返回当前指定的工具转换数据。

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 [ (<机器编号> ) ]

例) <位置变量> = P\_NTool

### 【用语】

<位置变量> 指定代入的位置变量。

<机器编号> 输入机器编号。1 ~ 3、省略时为 1。

### 【例子】

1 P1=P\_Tool        ’ 将工具转换数据代入 P1。

### 【说明】

(1) P\_Tool 返回 Tool 指令或参数 MEXTL 中设定的工具转换数据。

(2) 变更工具转换数据时应使用 Tool 指令。

(3) 读取专用。

### P UDev

**【功能】**

通过 CR800-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 CR800-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 ~ 24316 (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，返回值 0。仅对应 CR800-Q 系列。(通过 CR800-D/CR800-R 系列执行时，不执行写入处理，参照返回 0。)

**【补充】**

表 4-48: 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>实数注 1)<br>例)M1! | 双精度<br>实数注 1)<br>例)M1# | 位置<br>注 1), 注 2)<br>例)P1.X | 关节<br>注 1), 注 2)<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 4-42 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。

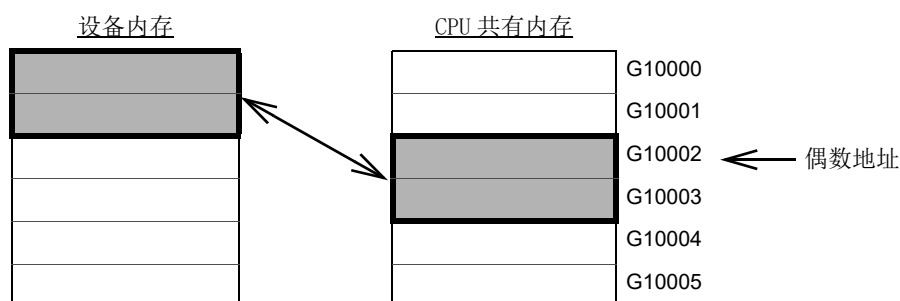


图 4-42: 防止 32 位数据的分离

##### 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 724 页的“7.3 关于 Ex-T 控制”。)

### 【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。
- 仅限 MELFA-BASIC V / VI 可使用。
- 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)。

**【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)。

**【说明】**

- (1) 想将 P 变量初始化时使用。
- (2) 读取专用。

## 4.14 函数的详细说明

### 4.14.1 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。

### 4.14.2 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>)

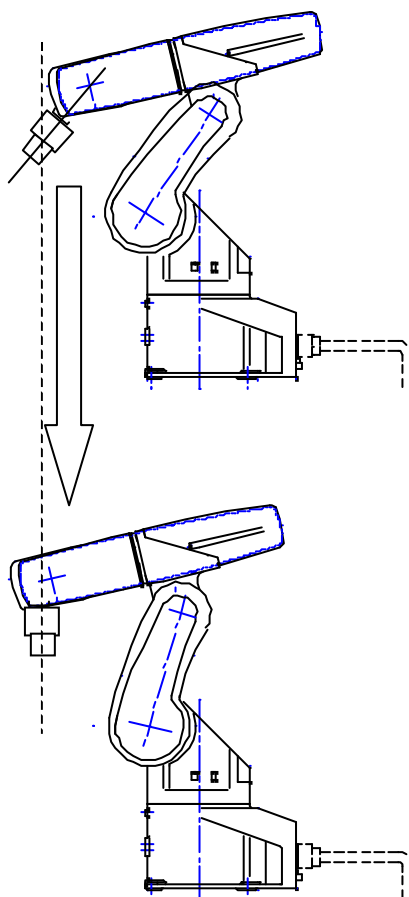
### 【例子】

```
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](#)

## ASin

### 【功能】

从指定的正弦值（数值）返回反正弦值（Arc sine）。

### 【格式】

$\langle \text{数值变量} \rangle = \text{ASin}(\langle \text{算式} \rangle)$

### 【用语】

$\langle \text{数值变量} \rangle$       计算并返回指定算式的值的反正弦（Arc sine）值。  
                           输出值的单位为弧度。  
                           范围： $-\pi/2 \sim +\pi/2$

$\langle \text{算式} \rangle$             指定正弦值。  
                           设定范围： $-1.0 \sim +1.0$

### 【例文】

1 MRad=ASin(-0.4)            ' 将 -0.4 的反正弦值 (-0.411516846067488 rad) 代入至 MRad。

### 【说明】

(1) 返回指定  $\langle \text{算式} \rangle$  的反正弦值。单位为弧度。

## Atn/Atn2

### 【功能】

计算余切（Arc Tangent）。

### 【格式】

$\langle \text{数值变量} \rangle = \text{Atn}(\langle \text{数式} \rangle)$   
 $\langle \text{数值变量} \rangle = \text{Atn2}(\langle \text{数式} 1 \rangle, \langle \text{数式} 2 \rangle)$

### 【用语】

$\langle \text{数值变量} \rangle$       依据指定的数式，余切（Arc Tangent）的值会被计算并返还。  
                           输出值的单位为弧度。

$\langle \text{数式} \rangle$              $\Delta Y / \Delta X$  的计算值

$\langle \text{数式} 1 \rangle$          $\Delta Y$

$\langle \text{数式} 2 \rangle$          $\Delta X$

### 【例子】

1 M1=Atn(100/100)            ' 在 M1 代入  $\pi/4$  弧度。  
 2 M2=Atn2(-100, 100)        ' 在 M2 代入  $-\pi/4$  弧度。

### 【说明】

- (1) 计算被给予的数式的余切。单位为弧度。
- (2) 得到的值为 Atn 的情况下， $-\pi/2 < \text{Atn} < \pi/2$ 。
- (3) 得到的值为 Atn2 的情况下， $-\pi < \text{Atn} < \pi$ 。
- (4) 在 Atn2 的情况下  $\langle \text{数式} 2 \rangle$  为 0 且  $\langle \text{数式} 1 \rangle$  为正的数时，会变成  $\pi/2$ 、负的数时会变成  $-\pi/2$ 。
- (5) 在 Atn2 的情况下， $\langle \text{数式} 1 \rangle$ 、 $\langle \text{数式} 2 \rangle$  将无法记述有自变量的函数。执行时会发生报警。

NG 例    M1=Atn2(Max(MA, MB), 100)  
              M1=Atn2(CInt(10.2), 100)

### 【参照】

Sin、Cos、Tan

## 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 中存在同一点、或 3 点位于一条直线上时。  
 -2: 圆弧生成失败时。

### 【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 MARCLEN=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】**

返回拥有指定数值的值的字符码（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) 将拥有数值的值的字符码文字返回。
- (2) 数值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】**

将数值的小数点部份四舍五入，转换为整数。

**【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) 将数值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】

<数值变量> = \*CkSum (<字符串>, <数式 1 >, <数式 2 >)

### 【用语】

|        |                 |
|--------|-----------------|
| <字符串>  | 指定计算和校检的字符串。    |
| <数式 1> | 指定计算和校检的开始文字位置。 |
| <数式 2> | 指定计算和校检的结束文字位置。 |

### 【例子】

1 M1=CkSum("ABCDEFG", 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<字符串 1 >, <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。

## Cos

### 【功能】

给予余弦 (Cosine)。

### 【格式】

<数值变量> = Cos (<数式>)

### 【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出入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](#)

## *Fix*

**【功能】**

返回数式的整数部。

**【格式】**

|                     |
|---------------------|
| <数值变量> = Fix (<数式>) |
|---------------------|

**【例子】**

1 M1=Fix(5.5)                      ’ 在 M1 输入 5。

**【说明】**

- (1) 返回数式的值的整数部份值。
- (2) 数式的值为正的情况下，返回和 Int 相同的值。
- (3) 数式的值为负的情况下，会变成例如 Fix(-2.3)=-2.0。

**【参照】**

[CInt](#)、[Int](#)

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 502 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](#)

### Inv

**【功能】**

求得位置变量的逆矩阵的位置数据。位置的相对运算时使用。

**【格式】**

$$\langle \text{位置变量} \rangle = \text{Inv} (\langle \text{位置变量} \rangle)$$

**【例子】**

1 P1=Inv (P2) ' 在 P1 代入 P2 的逆矩阵。

**【说明】**

- (1) 求得位置变量的逆矩阵的位置数据。
- (2) 无法在自变量使用关节变量，使用时会发生报警。
- (3) 因为返回值为位置数据，因此在左边使用关节变量的情况下，会发生报警。

### JtoP

**【功能】**

将被给予的关节数据变换为位置数据。

**【格式】**

$$\langle \text{位置变量} \rangle = \text{JtoP} (\langle \text{关节变量} \rangle)$$

**【例子】**

1 P1=JtoP (J1) ' 在 P1 代入 J1 (关节型) 的位置以直交型表示时的位置。

**【说明】**

- (1) 将被给予的关节数据执行坐标变换，变换为位置数据。
- (2) 无法在自变量使用关节变量，使用时会发生报警。
- (3) 因为返回值为位置数据，因此在左边使用关节变量的情况下，会发生报警。

**【参照】**

[PtoJ](#)

## Left\$

### 【功能】

从字串的左端，取得指定长度的字串。

### 【格式】

|                              |
|------------------------------|
| <字串变量> = Left\$ (<字串>, <数式>) |
|------------------------------|

### 【例子】

1 C1\$=Left\$("ABC",2)                    ' 在 C1\$ 带入 "AB"。

### 【说明】

- (1) 从字串的左端，取得指定长度的字串。
- (2) 数值为负的情况及比字串的长度还长的情况下，会发生报警。
- (3) <字串>, <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】

返回字串的长度。

### 【格式】

|                     |
|---------------------|
| <数值变量> = Len (<字串>) |
|---------------------|

### 【例子】

1 M1=Len("ABCDEFG")                    ' 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】**

|                    |
|--------------------|
| <数值变量> = Ln (<数式>) |
|--------------------|

**【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】**

|                     |
|---------------------|
| <数值变量> = Log (<数式>) |
|---------------------|

**【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\$("ABCDEFGG", 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\$ 中代入指定数式的值转换成的 2 字节字符串。 |
| 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) | ' 在 C1\$ 中代入指定数式的值转换成的 4 字节字符串。 |
| 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)                                 ' 在 C1\$ 中代入指定数式的值转换成的 8 字节字符串。
- 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】**

<数值变量> = PosCq (<位置变量>)

**【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】**

<位置变量> = PosMid (<位置变量 1>, <位置变量 2>, <数式 1>, <数式 2>)

**【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<位置变量 1>, <位置变量 2>, <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。

## PtoJ

**【功能】**

将被给予的位置数据转换为关节数据。

**【格式】**

$\langle \text{关节变量} \rangle = \text{PtoJ} (\langle \text{位置变量} \rangle)$

**【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】**

$\langle \text{数值变量} \rangle = \text{Rad} (\langle \text{数式} \rangle)$

**【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。<br>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 进制表记) 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$=Right$("ABCDEFGG", 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](#)

## Setf11

**【功能】**

将指定位置的姿势标志依据文字（"RAN" 等）变更。

**【格式】**

```
<位置变量> = Setf11 (<位置变量>, <字符串>)
```

**【用语】**

<位置变量> 指定将姿势标志变更的位置变量。  
 <字符串> 指定要变更的姿势标志。可以复数指定。  
 "R" or "L": Right/Left 的设定  
 "A" or "B": Above/Below 的设定  
 "N" or "F": Nonflip/Flip 的设定

**【例子】**

```
10 Mov P1
20 P2=Setf11(P1,"LBF")
30 Mov P2
```

**【说明】**

- (1) 返还将第 1 自变量的位置数据的姿势标志以第 2 自变量在指定的值变更的位置数据。
- (2) 本函数变更位置数据的 FL1 的要素情报。以自变量被给予的位置数据内容不会被改变。
- (3) 姿势标志的指定是从字符串的最后的文字开始被设定。因此，如果 "LR" 等被指定的情况下，结果的姿势标志会变成 "L"。
- (4) 以数值变更的情况下，请设定为 P1.FL1=7。
- (5) 姿势标志的意义会依机器人的机型而有所不同，详情请参照各机器人的「机器人本体设定到保养」。

姿势标志相当于位置常数 (100, 0, 300, 180, 0, 180) (Z, 0) 的 7 的部份。实际的位置会变成 bit 模式。

```
7=&B00000111
```

```

 1/0=N/F
 1/0=A/B
 1/0=R/L

```

- (6) 在<位置变量>, <字符串>无法记述含有自变量的函数。执行时会发生报警。

**【参照】**

[Rdf11](#)、[Rdf12](#)、[Setf12](#)

## Setf12

### 【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> = Setf12 (<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=Setf12(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。

多回转数据的值

|         |      |      |      |   |     |     |     |
|---------|------|------|------|---|-----|-----|-----|
| 各轴的角度   | -900 | -540 | -180 | 0 | 180 | 540 | 900 |
| 多回转数据的值 | ...  | -2   | -1   | 0 | 1   | 2   | ... |

- (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】

[Rdf11](#)、[Rdf12](#)、[Setf11](#)

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】

设定往位置变量的值。

### 【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】

<数值变量> = Sin (<数式>)

### 【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

## SpIECord

### 【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】

<位置变量> = SpIECord(<样条号码>[, <帧转换>])

### 【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)=SpIECord(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=SpIECord(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=SpIECord(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。<br>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。<br>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。<br>设定范围：<br>0：不执行帧转换。<br>1：使用样条文件中设定的坐标系执行帧转换。<br>2：使用SetCalFrm指令中设定的坐标系执行帧转换。<br>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。

## Sp1Spd

### 【功能】

通过样条插补指令 (MvSp1、EMvSp1 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 > = Sp1Spd(< 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=Sp1Spd(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 MvSp1 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 (MvSp1、EMvSp1 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。

## StrLwr

**【功能】**

将字符串转换为小写字符。

**【格式】**

<字符串变量> = StrLwr (<字符串>)

**【术语】**

<字符串变量>    指定要代入的字符串变量。  
<字符串>        指定要转换的字符串。

**【示例】**

1 C1\$ = StrLwr (“ABC” )’ C1\$ 为 “abc” 。

**【说明】**

- (1) 可将字符串转换为小写字符。

**【参照】**

[StrUpr](#)

## StrUpr

**【功能】**

将字符串转换为大写字符。

**【格式】**

```
<字符串变量> = StrUpr (<字符串>)
```

**【术语】**

<字符串变量> 指定要代入的字符串变量。

<字符串> 指定要转换的字符串。

**【示例】**

1 C1\$ = StrUpr ("abc") ' C1\$ 为 "ABC" 。

**【说明】**

(1) 可将字符串转换为大写字符。

**【参照】**

[StrLwr](#)

## 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("ABCDEFGG","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 "ABCDEFGG"、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。



## Tan

**【功能】**

计算正切（tangent）。

**【格式】**

$\langle \text{数值变量} \rangle = \text{Tan} (\langle \text{数式} \rangle)$

**【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】**

$\langle \text{数值变量} \rangle = \text{Val} (\langle \text{字符串式} \rangle)$

**【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】**

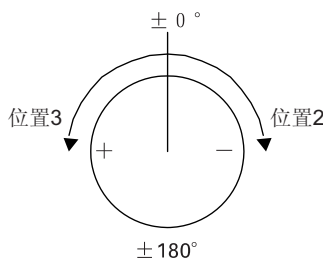
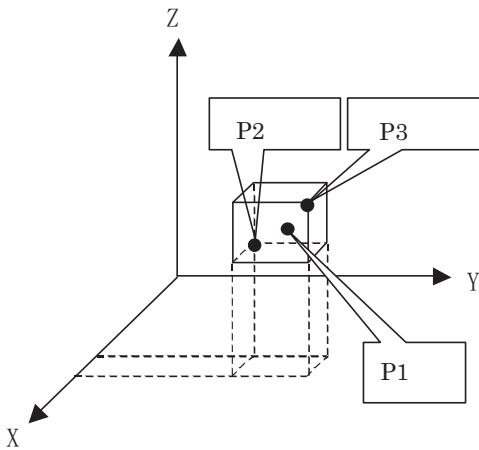
- <位置 1 >            想确认的位置
  - <位置 2 >            指定领域的第 1 点的位置
  - <位置 3 >            指定领域的第 2 点的位置（对角点）
- 位置 1 ~ 3 设定为直交坐标的变量（P 变量 X, Y, Z, A, B, C, L1, L2）。

**【例子】**

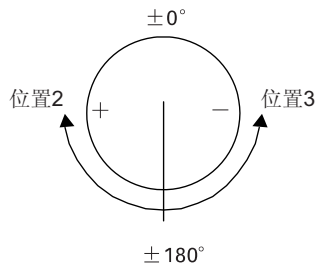
```
1 M1=Zone(P1,P2,P3)
2 If M1=1 Then Mov P_Safe Else End
```

**【说明】**

- (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。
- (2) 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。
- (3) 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）。
- (4) 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。
- (5) 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】

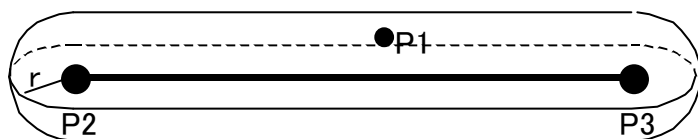
|         |              |
|---------|--------------|
| <位置 1 > | 想确认的位置       |
| <位置 2 > | 指定领域第 1 点的位置 |
| <位置 3 > | 指定领域第 2 点的位置 |
| <数式>    | 两端半球的半径      |

### 【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】

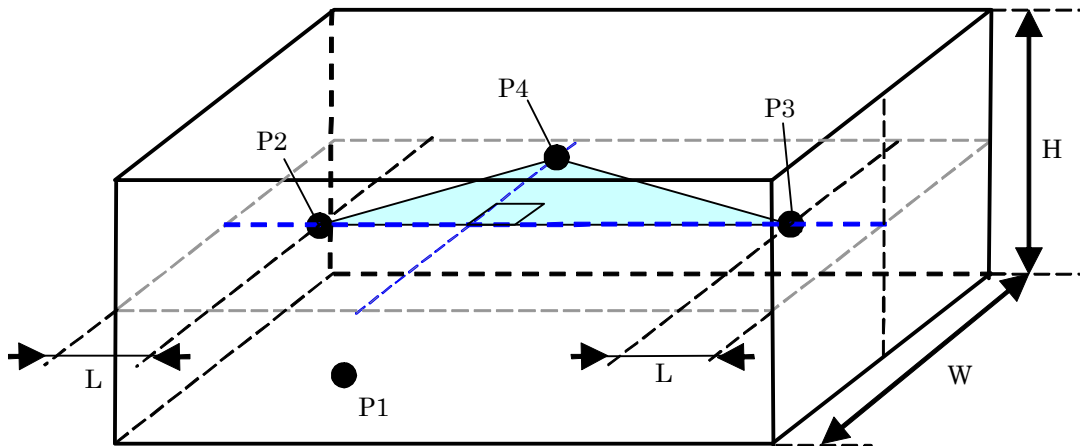
|         |                                     |
|---------|-------------------------------------|
| <位置 1 > | 想确认的位置                              |
| <位置 2 > | 指定领域第 1 点的位置                        |
| <位置 3 > | 指定领域第 2 点的位置                        |
| <位置 4 > | 随着<位置 2 >、<位置 3 >指定构成领域的平面指定点的位置    |
| <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 468 页 |
| 2   | 信号参数 | 信号关联用的参数。关于专用信号请参照第 605 页的“6.2PLC 连接输入输出功能”。<br>本章中存在“CPU 缓冲存储器”记载的部分为 CR800-Q 系列时，请改读为“CPU 间共享存储器”。 | 第 482 页 |
| 3   | 操作参数 | 控制器、示教单元等的操作相关参数。                                                                                    | 第 490 页 |
| 4   | 语言参数 | 机器人语言相关参数。                                                                                           | 第 493 页 |
| 5   | 通信参数 | 通信相关参数                                                                                               | 第 497 页 |

关于专用输入输出信号的参数请参照第 605 页的“6.2PLC 连接输入输出功能”。  
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。参数的详细操作方法请参照本书 / 第 93 页的“3.15 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。

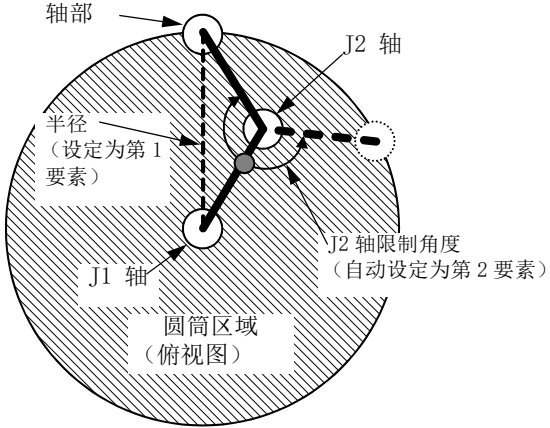
表 5-1：动作关连参数一览表

| 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：<br>deg | 每个机器的设定值                                                                        |
| 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 |
| 窄角广角极限功能<br>※ 仅限 RV-4FR/7FR/<br>13FR/20FR 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”。<br>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-4FR/7FR/<br>13FR/20FR 系列<br>其他机型为 0 (无效)                      |

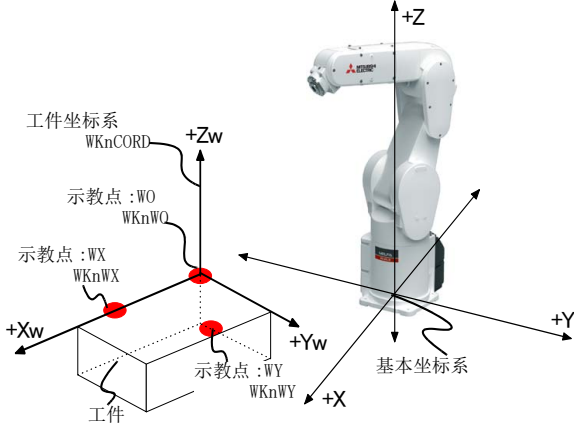
| 参数                                               | 参数名                    | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 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。在其他类型的机器人中无效，因此请勿设定。</p> <p>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。</p> <p>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：-360 ≤ 第2要素 ≤ +360</p> <p>设定了 J1OFFSET 时，需要设定 J1 轴的原点。按照以下的步骤实施原点设定。<br/>原点设定的方法请参照另一手册的“从机器人本体安装到维护”。</p> <ol style="list-style-type: none"> <li>1) 在 J1OFFSET 中设定值。</li> <li>2) 重新接通电源。</li> <li>3) J1 轴对准 0 度。</li> <li>4) 以夹具方式进行 J1 轴的原点设定。</li> <li>5) 在示教单元的 JOG 画面中显示关节坐标，确认 J1 轴的角度为在 1) 中设定的值。</li> <li>6) 更改的原点数据应记录到以下位置。 <ul style="list-style-type: none"> <li>• 机器人内部的原点数据表</li> </ul> </li> </ol> | 0.0, 0.0                                                |
| 标准工具坐标<br>请参照“4.4 机器人的坐标系说明”、“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。</p> <p>出厂时设定机械接口为控制点。</p> <p>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 |
| 工具坐标 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工具转换数据。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | (X, Y, Z, A, B, C) =<br>0.0, 0.0, 0.0,<br>0.0, 0.0, 0.0 |
| 工具功能设定                                           | TOOLSPEC               | 整数 1       | <p>工具编号选择了 0 时，选择工具参数 MEXTL 也同时复位 (初始化为所有部分都为 0) / 不复位。</p> <p>0: 复位<br/>1: 不复位<br/>需要控制器的软件版本为 A4 版以上。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0 (复位)                                                  |
| 标准基本坐标<br>请参照“4.4 机器人的坐标系说明”、“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。</p> <p>出货时设定基本坐标和机器人坐标为一致。作为装置整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的时候设定。</p> <p>(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 |

| 参数                            | 参数名                                | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 出货时设定值                                                  |    |    |         |             |    |         |        |     |         |         |        |                                                                                           |
|-------------------------------|------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|----|----|---------|-------------|----|---------|--------|-----|---------|---------|--------|-------------------------------------------------------------------------------------------|
| 基本坐标编号<br>请参照“4.4 机器人的坐标系说明”。 | MEXBSNO                            | 整数 1       | 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转换数据，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。                                                                                                                                                                                                                                                                                        | -1                                                      |    |    |         |             |    |         |        |     |         |         |        |                                                                                           |
| 系统基本坐标系                       | MEXSBS                             | 实数 6       | 设定世界坐标系和基本坐标系的位置关系。<br>出厂设定中世界坐标系和基本坐标系为一致。<br>本参数无法以Base指令切换。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | (X, Y, Z, A, B, C) =<br>0.0, 0.0, 0.0, 0.0,<br>0.0, 0.0 |    |    |         |             |    |         |        |     |         |         |        |                                                                                           |
| 系统工具坐标系                       | MEXSTL                             | 实数 6       | 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>本参数无法以Tool指令及M_Tool切换。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | (X, Y, Z, A, B, C) =<br>0.0, 0.0, 0.0, 0.0,<br>0.0, 0.0 |    |    |         |             |    |         |        |     |         |         |        |                                                                                           |
| 用户定义区域<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><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" data-bbox="646 967 1220 1108"> <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><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,<br>L1, L2) =<br>0.0, 0.0, 0.0, -<br>360.0,<br>-360.0, -360.0, 0<br>, 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" data-bbox="646 1406 1220 1547"> <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><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,<br>L1, L2) =<br>0.0, 0.0, 0.0,<br>+360.0, +360.0,<br>+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     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                         |    |    |         |             |    |         |        |     |         |         |        |                                                                                           |

| 参数                           | 参数名                                                                        | 排列数<br>文字数     | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 出货时设定值                                                                           |     |     |      |                  |              |                    |                |      |                             |   |
|------------------------------|----------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|-----|-----|------|------------------|--------------|--------------------|----------------|------|-----------------------------|---|
| 用户定义区域<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> <table border="1" style="margin-left: 20px;"> <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<br/>(※1)</td> <td>专用输出信号<br/>关闭</td> </tr> <tr> <td>状态变量的对应位开启<br/>(※2)</td> <td>状态变量的对<br/>应位关闭</td> </tr> <tr> <td>报警输出</td> <td>因报警输出导致的停止<br/>(发生 H2090 报警)</td> <td>-</td> </tr> </tbody> </table> ※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<br>(※1) | 专用输出信号<br>关闭 | 状态变量的对应位开启<br>(※2) | 状态变量的对<br>应位关闭 | 报警输出 | 因报警输出导致的停止<br>(发生 H2090 报警) | - |
| 设定                           | 领域内                                                                        | 领域外            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                  |     |     |      |                  |              |                    |                |      |                             |   |
| 信号输出                         | 专用输出信号开启<br>(※1)                                                           | 专用输出信号<br>关闭   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                  |     |     |      |                  |              |                    |                |      |                             |   |
|                              | 状态变量的对应位开启<br>(※2)                                                         | 状态变量的对<br>应位关闭 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                  |     |     |      |                  |              |                    |                |      |                             |   |
| 报警输出                         | 因报警输出导致的停止<br>(发生 H2090 报警)                                                | -              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                  |     |     |      |                  |              |                    |                |      |                             |   |
| 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, X3, Y3, Z3)=0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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)                                                                           |     |     |      |                  |              |                    |                |      |                             |   |

| 参数                                         | 参数名       | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                     | 出货时设定值                 |
|--------------------------------------------|-----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 圆筒限制<br>※ 仅在以下机型中有效。<br>机型：<br>RH-3FRHR 系列 | MECAR     | 实数 2       | <p>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。</p> <p>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。</p>  <p>注) 本功能限于RH-3FRHR系列机型。</p> | 0, 0                   |
| 退避点位置                                      | JSAFE     | 实数 8       | <p>指定退避点位置。<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，往退避点位置动作<sub>xz</sub><br/>(J1, J2, J3, J4, J5, J6, J7, J8) 单位：deg</p>                                                                                                                                                                                                                                                            | 依机型会有所不同。              |
| 选择调整类型                                     | ALIGN TYP | 整数 1       | <p>指定抓手排列动作与Align函数抓手姿势的计算方法。抓手排列动作中，以该抓手姿势运动。Align函数返回抓手排列的直交坐标值。</p> <p>0：通常<br/>关于直交坐标姿势的各成分（A，B，C），求出最近的直交姿势（0度、±90度、±180度）。<br/>RV类型机器人中，应指定此方式。</p> <p>1：放射状<br/>求出更改姿势轴（C）以确保沿着原点（(x, y)=(0, 0)）与控制点连接的直线放射的位置。<br/>RH类型的机器人中，在工件配置成圆筒状的情况下等，使用此方式时比较便利。<br/>RV类型中，请勿指定此方式。</p> <p>2～：预留</p>                                                                                      | 0                      |
| 原点设定完成状态                                   | MEINST    | 整数 1       | <p>显示原点设定完成的状态。<br/>本参数在原点设定时自动设定。</p>                                                                                                                                                                                                                                                                                                                                                   | 0                      |
| 原点设定时 0 位置的脉冲数                             | MEINSZ    | 整数 8       | <p>设定原点设定时0位置的脉冲数。<br/>本参数在原点设定时自动设定。</p>                                                                                                                                                                                                                                                                                                                                                | 0, 0, 0, 0, 0, 0, 0, 0 |
| 原点设定时 1 转内的脉冲                              | MEOFFZ    | 整数 8       | <p>设定原点设定时0位置1转内的脉冲位置。<br/>本参数在原点设定时自动设定。</p>                                                                                                                                                                                                                                                                                                                                            | 0, 0, 0, 0, 0, 0, 0, 0 |
| 原点设定的检查数据                                  | MEINSD    | 整数 1       | <p>MEINSZ的检查数据。<br/>本参数在原点设定时自动设定。</p>                                                                                                                                                                                                                                                                                                                                                   | 0                      |
| 机械限位器原点                                    | MORG      | 实数 8       | <p>指定机械限位器原点位置。<br/>(J1, J2, J3, J4, J5, J6, J7, J8) 单位：deg</p>                                                                                                                                                                                                                                                                                                                          | 依机型会有所不同。              |
| 用户指定原点                                     | USERORG   | 实数 8       | <p>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</p>                                                                                                                                                                                                                                                                                                                 | 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   |
|-----------------------------------------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 用户原点设定方式                                | UORGSPEC | 整数 1       | 设定用户原点设定方式。如果设定为高精度方式，则在设定用户原点时，考虑到作用于机器人的重力引起的挠曲而进行补偿。<br>0: 正常<br>1: 高精度<br>本参数的详细内容，请参照另一手册“从机器人本体安装到维护”。<br>※本参数支持C2j以上型号的机器人控制器。                           | 0        |
| 选择特异点接近警告功能<br><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><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)   |
| 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 速度规定值                               | 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J0以定义工件坐标。此时, G操作时的基准坐标或工件坐标数据使用。此外, 作为Ex-T控制中的控制点 (Ex-T坐标) 使用。<br/>关于Ex-T控制, 请参照第724页的“7.3关于Ex-T控制”。<br/>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502页的“5.7关于标准基本坐标”。<br/>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) 在基本转换为初始值的状态(世界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)<br/>(X, Y, Z) 单位: mm<br/>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RT ToolBox3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。<br/>(对应于示教单元中的示教操作的“WX”。参照上图)<br/>(X, Y, Z) 单位: mm<br/>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RT ToolBox3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。<br/>(对应于示教单元中的示教操作的“WY”。参照上图)<br/>(X, Y, Z) 单位: mm<br/>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RT ToolBox3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)                      |

| 参数                                                                | 参数名                                               | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 出货时设定值 |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |
|-------------------------------------------------------------------|---------------------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------------------------------------------------|----------|------------------------|-------------------|------------------------|-------------------|------------------------|----------------------|-----------------------|--------------------|
| 工件 JOG 动作模式                                                       | WKnJOGMD<br>n 为 1 ~ 8                             | 整数 1       | 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的动作为与工件坐标的X轴、Y轴、Z轴平行旋转。控制点位置不变。)<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并同时旋转。)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0      |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |
| 选择工件坐标编号                                                          | WKJOGNO                                           | 整数 1       | 设定选择中的工件坐标编号。<br>本参数在工件坐标切换操作时自动设定。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0      |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |
| 中断时 JOG 前进后的自动恢复设定<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>※ 仅限 RV-FR 系列 (RV-35FR/50FR/80FR 系列除外)、RV-8CRL、RV-5AS | MEGDIR                                            | 实数 4       | <p>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: m/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。</p> <table border="1" style="margin-left: 20px;"> <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(标准)</td> <td>( 0.0, 0.0, 0.0, 0.0 )</td> </tr> <tr> <td>壁挂<sup>注1)</sup></td> <td>( 1.0, 0.0, 0.0, 0.0 )</td> </tr> <tr> <td>垂吊<sup>注2)</sup></td> <td>( 2.0, 0.0, 0.0, 0.0 )</td> </tr> <tr> <td>任意的姿势<sup>注3)</sup></td> <td>( 3.0, ***, ***, ***)</td> </tr> </tbody> </table> <p>注 1) RV-5AS 无法壁挂使用。<br/>注 2) RH-3FRHR 系列应在第 1 要素为“0.0”的情况下直接使用。<br/>注 3) 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。</p> <p>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 x sin(30 度) = 4.9<br/>Z 轴方向重力加速度 (Zg) = 9.8 x cos(30 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)。</p>  | 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(标准) | ( 0.0, 0.0, 0.0, 0.0 ) | 壁挂 <sup>注1)</sup> | ( 1.0, 0.0, 0.0, 0.0 ) | 垂吊 <sup>注2)</sup> | ( 2.0, 0.0, 0.0, 0.0 ) | 任意的姿势 <sup>注3)</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(标准)                                                          | ( 0.0, 0.0, 0.0, 0.0 )                            |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |
| 壁挂 <sup>注1)</sup>                                                 | ( 1.0, 0.0, 0.0, 0.0 )                            |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |
| 垂吊 <sup>注2)</sup>                                                 | ( 2.0, 0.0, 0.0, 0.0 )                            |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |
| 任意的姿势 <sup>注3)</sup>                                              | ( 3.0, ***, ***, ***)                             |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |                                                   |          |                        |                   |                        |                   |                        |                      |                       |                    |



| 参数                                                        | 参数名                                                                                                                  | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                       | 出货时设定值                                                                                         |
|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| 抓手初始状态<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                                                                         |
| 抓手类型<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。                                                                                                  | CR800-D/R/Q:<br>D900, D902, D904, D906, , , ,<br>CR860-D/R/Q:<br>D764, D766, D768, D770, , , , |
| 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>-1: 未设定/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Oad1 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。 |            |                                                                                                                                                                                                            |                                                                                                |
|                                                           | HNDDATO                                                                                                              | 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X, 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X, 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, 0.0, 0.0                                                    |
|                                                           | WRKDATO                                                                                                              | 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X, 重心X, 重心Y, 重心Z）<br>单位: Kg, mm                                                                                                           | 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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X, 重心X, 重心Y, 重心Z）<br>单位: Kg, mm                                                                                                                                   | 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 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                                                                                           |
| 抓手条件设定检查                                                  | HNDCHK                                                                                                               | 整数 1       | 对HNDDAT参数未设定警告的有效/无效进行设定。<br>0: 无效<br>1: 有效<br>本参数有效（1）的情况下，接通控制器电源时，如果抓手条件参数HNDDAT*（*=0~8）均为初始值，即未进行设定，则会发生错误C0330。<br>如果本参数设定为无效（0），则不进行上述的设定检查，即使未设定抓手条件，也不会发生错误C0330。<br>在控制器的软件版本为C2d以上的版本中，已经追加了本参数。 | 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                                                                                              |

| 参数                                                 | 参数名     | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                          | 出货时设定值                                                                                                   |
|----------------------------------------------------|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| 最佳加减速补偿率                                           | JADL    | 实数 8       | <p>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进行高速动作，但是如果持续一定时间，可能会发生过负载报警、过热报警及再生过度报警等。报警出现的情况下，请将设定值调小。</p> <p>初期状态会设定为预防过负载报警、过热报警及再生过度报警发生的初始值。</p> <p>此外，如果将本设定值增大，可能会导致消耗的电力增多。加速度和减速度均适用。</p> <p>※所谓过负载错误报警<br/>为防止因高速旋转产生的热量导致电机破损，负载率超过一定程度时会发生该报警。</p> <p>※所谓过热报警<br/>为防止因高速旋转产生的热量导致位置检测器破损，温度上升到一定程度时会发生该报警。</p> <p>※所谓再生过度报警<br/>为防止再生电阻因高速旋转减速而产生的热量而损坏，超过一定再生水平时会发生该报警。</p> | 依机型会有所不同。                                                                                                |
| 加减速最佳化模式选择<br>※ 仅限 RH-FRH 系列（RH-CRH/RH-3FRHR 系列除外） | MAPMODE | 整数 1       | <p>选择将RH-FRH系列（RH-3FRHR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。</p> <p>0：标准加减速模式/1：高速加减速模式</p> <p>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」。</p>                                                                                                                                                                    | 0                                                                                                        |
| 最佳速度控制调整系数                                         | OPTOVC  | 实数 1       | <p>设定最佳速度控制的调整系数。</p> <p>设定范围：0.30~1.00</p> <p>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。</p> <p>※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。</p>                                                                                                                         | 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-3FRHR 系列       | SPDOPT  | 整数 1       | <p>设定速度调整插补功能的有效/无效。</p> <p>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。</p> <p>注）本功能限于RH-3FRHR系列机型。</p>                                                                                                                                                                                                                                                    | <p>RH-3FRHR 系列<br/>1</p> <p>其他机型<br/>0</p>                                                               |
| 冲突检知                                               | COL     | 整数 3       | <p>定义冲突检知的使用可 / 不可和电源一开启后的有效 / 无效。</p> <p>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)</p> <p>在 NOERR 模式即使检测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 操作作用（要素 3）的设定。</p>                                                                                                                        | <p>RH-3FRH/6FRH/<br/>12FRH/20FRH系列：<br/>1, 0, 1<br/>RV-FR系列：<br/>0, 0, 1<br/>RH-3FRHR系列：<br/>1, 1, 1</p> |

| 参数             | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                     | 出货时设定值    |
|----------------|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 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：%                                                                                                                                                                                                 | 依机型会有所不同。 |
| JOG 时的检知 LEVEL | COLLVLJG | 实数 8       | 设定每个关节轴在 JOG 操作时（包含中断中）的检知 LEVEL（感应度）。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：%                                                                                                                      | 依机型会有所不同。 |
| 碰撞检测后的伺服状态     | COLSERVO | 整数 1       | 指定撞击检测后的伺服状态。使用时，需要在 NOERR 模式下使用碰撞检测。<br>0：伺服 OFF<br>1：保持伺服 ON                                                                                                                                                                                           | 0         |
| 腕部旋转（A 轴）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>将在低温下的工作时会发生误差过大出现错误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。（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     |

| 参数              | 参数名     | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                        | 出货时设定值                                                                                            |
|-----------------|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 暖机运行倍率修调        | WUPOVRD | 整数 2       | <p>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: %</p> <p>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)</p> <p>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)</p> <p>暖机运行倍率修调的值和各要素的设定值的对应如下图所示。</p> <p>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。</p> | 70, 50                                                                                            |
| 柔量控制错误的功能设定     | 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。<br/>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。<br/>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。<br/>本参数更改为 0 (错误发生无效), 仅限于依据错误的不同, 可判断为即使不中断运行运行上也没问题时进行。</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。<br/>设定范围: 1(限制高) ~ 10(限制低)<br/>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。<br/>以该设定值为基础对机器人动作时的过载检知 Level 进行优化。<br/>(单位: °C)<br/>设定范围: 0 ~ 40<br/>关于与本参数相关的功能的详细内容, 请参照第 545 页的“5.22 关于过载 Level 的最佳化”。</p>                                                                                                                                                                                                     | RH-3FRH/6FRH/<br>12FRH/20FRH: 40<br>RV-2FR 系列: 40<br>RV-4FR/7FR/13FR/<br>20FR 系列、RH-<br>3FRHR: 30 |
| 电机过热警告          | MOHW    |            | <p>指定电机过热错误发生前是否发出了警告。<br/>0: 无效<br/>1: 有效</p>                                                                                                                                                                                                                                                                                                              | 1                                                                                                 |

| 参数                                | 参数名                | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                              | 出货时设定值                                                                                |
|-----------------------------------|--------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 电池使用时间错误发生时间间隔                    | ITBATERR           | 整数 1       | 指定电池使用时间错误的发生时间间隔。<br>(单位: 小时)<br>设定范围: 1 ~ 336。<br>设定值比 1 小时视为 1, 比 336 大时视为 336。                                                                                                                                                                                                                | 24                                                                                    |
| 与可编程控制器的时间同步<br>(仅限 CR800-R/Q 系列) | TIMESYNC           | 整数 1       | 选择是否将机器人控制器的设定时间与可编程控制器的设定时间设为同步。<br>(同步 / 不同步 =1/0)                                                                                                                                                                                                                                              | 1                                                                                     |
| Wait 指令的功能定义                      | PRSPEC             | 整数 1       | 设定 Wait 指令的执行方式。<br>0: 条件指令方式 (可记叙多个条件式)<br>1: 动作指令方式 (条件式为 1 个)                                                                                                                                                                                                                                  | 0                                                                                     |
| WthIf 指令规格                        | WTHFUNC            | 整数 1       | 指定 WthIf 指令的停止类型。<br>0: 停止类型 1<br>1: 停止类型 2<br>关于停止类型请参照 Def Act 指令的说明。                                                                                                                                                                                                                           | 0                                                                                     |
| Cnt 无效时的动作指令结束条件指定                | MVTERM             | 整数 1       | 指定 Cnt 无效时的动作指令结束条件。<br><br>FR 系列 (RV-2FR/2FRL、RH-3FRHR 除外) 及 RV-8CRL 中, 为提高 Cnt0 时的目的位置到达性, 变更了动作指令的结束条件。F 系列之前的机型 (SD/SQ 系列) 中制作的程序直接用于 FR 系列 (RV-2FR/2FRL、RH-3FRHR 除外) 及 RV-8CRL 时, 动作时间可能会变长。动作速度优先时, 应更改本参数, 在需要定位的位置上使用 Fine P 及 Dly 指令等。<br><br>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 | 0: FR 系列<br>(RV-2FR/2FRL、RH-3FRHR 除外)、RV-8CRL<br>2: RV-2FR/2FRL、RH-3FRHR、RH-3/6CRH 系列 |
| 退避点回归模式                           | ESCMODE            | 整数 1       | 退避点回归时, 选择将当前位置设定为工具定义位置或机器人接口原点。<br>0: 设定为 TCP 位置 (工具定义位置)<br>1: 设定为机械接口原点                                                                                                                                                                                                                       | 0                                                                                     |
| 控制器 CPU 运行模式                      | DRVMODE            | 整数 1       | 使用软件进行模拟时, 指定是否需要与机器人连接。<br>0: 通常模式<br>(需要与机器人连接)<br>1: 机器人伺服分离模式<br>(不需要与机器人连接)                                                                                                                                                                                                                  | 0                                                                                     |
| 反馈增益调整系数                          | FBGAIN             | 整数 8       | 调整各关节轴的速度环增益。<br>设定范围: 50 ~ 150<br>单位: [%]<br><br>通过降低设定值, 可抑制电机的振荡及驱动部外部干扰的影响。通过提高设定值, 可改善轨迹精度及沉降时间, 但可能会出现电机振荡的情况, 建议边确认状况边慢慢地提升。                                                                                                                                                               | 100, 100, 100, 100,<br>100, 100, 100, 100                                             |
| 固定增益化                             | PG1                | 整数 1       | 将可变增益控制设为无效, 使固定增益有效。在跟踪动作等的跟随性不佳时进行设定。<br>设定范围: 0.00 ~ 120.00<br><br>不足 1.0: 可变增益控制有效<br>1.0 ~ 120.0: 使固定增益有效<br><br>通常设定为 20 左右的值。发生振动的话, 请减小设定值。值设定过大时, 可能会发生振动。                                                                                                                               | 0.0                                                                                   |
| 伺服模拟功能                            | SRVSIM             | 整数 1       | 在 RT ToolBox3 的模拟中对负载率模拟的有效 / 无效进行指定。<br>0: 无效<br>1: 有效                                                                                                                                                                                                                                           | 1                                                                                     |
| 视觉传感器校准数据                         | VSCALBn<br>n=1 ~ 8 | 整数 12      | 通过 RT ToolBox3 的 2D 视觉校准功能定义的视觉传感器校准数据将被保存。<br>对应校准编号 n=1 ~ 8 的共计 8 个校准。                                                                                                                                                                                                                          | 0.0、0.0、0.0、0.0、<br>0.0、0.0、0.0、0.0、<br>0.0、0.0、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       |
|--------------------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| MVA2 指令的自变量省略时的设定值 | MVACNT1 | 实数 4       | 对省略 MVA2 指令的自变量时适用的设定值进行指定。<br>要素 1: 接近距离 1[mm]<br>设定范围: -1/0 以上的实数<br>要素 2: 接近距离 2[mm]<br>设定范围: -1/0 以上的实数<br>※ 指定为 -1 时变为与 Cnt1 相同的动作<br>要素 3: 定位距离 [mm]<br>设定范围: 0 以上的实数<br>要素 4: 整定时间 [sec]<br>设定范围: 0 以上的实数 | 0, 0, 0.1, 0 |

## 5.2 信号参数

信号关连用的参数。关于专用信号请参照第 616 页的“6.3 专用输入输出”。

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                            |            |            | 关于专用输入输出信号的参数请参照第616页的“6.3 专用输入输出”。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |            |  |  |    |     |      |   |   |   |   |   |   |   |    |   |   |    |   |   |   |    |    |   |    |   |   |   |    |   |    |   |    |    |   |   |    |    |    |                  |
| 在起动信号的输入时<br>读取从数值输入的程<br>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>设定开放机器人错误输出端子的错误Level。<br/>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, 或是两者都发生了时, 开放该输出端子, 需设定3。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <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>关于详细内容, 请参照另一手册的“使用说明书/从控制器安装及基本操作到维护”的“外部紧急停止输入输出/停止专用输入/门开关/模式选择开关输入的连接”。</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>(※仅限CR800-D系<br>列) | E7730      | 整数 1       | <p>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。<br/>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)</p> <p>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)。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 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, 00000000,<br>00000000, 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）输出信号700~715为系统保留领域，因此无法设定。<br>使用CR860-D/R/Q时，输出信号764~771分配至抓手输出信号，因此应在HANDINIT中进行设定。              | 00000000, 00000000,<br>00000000, 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, 00000000,<br>00000000, 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）      |
| 多 CPU 台数设定<br>（※ 仅限 CR800-R/Q<br>系列） | 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                                              |



| 参数                                                 | 参数名                 | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 出货时设定值     |      |   |           |   |           |   |           |  |
|----------------------------------------------------|---------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------|---|-----------|---|-----------|---|-----------|--|
| CPU 缓冲存储器固定周期通信区域设定<br>(※ 仅限 CR800-R 系列)           | QMLTCPUn<br>n=1 ~ 4 | 整数 4       | <p>多CPU系统中，从1号机读取通过1~4号机的CPU缓冲存储器固定周期通信区域设定进行发送接收的点数，并自动设定。无需更改值。</p> <p>要素1: 固定周期通信区域的大小 (k字)<br/>范围: 0~12<br/>※全号机合计最多24k字</p> <p>要素2: 自动刷新的点数 (字)<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: 系统备用</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>                                                                                                                                                                                                                                                                                                                                          | 1, 0, 1, 1 |      |   |           |   |           |   |           |  |
| 多 CPU <sub>n</sub> 号机高速通信区域设定<br>(※ 仅限 CR800-Q 系列) |                     |            | <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。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: 40px;"> <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 |  |
| 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 CR800-R 系列) | QMLTCPUS | 整数 1       | CR800-R 系列时, 对多 CPU 的机器人输入信号的偏置进行设定。<br><br>以 1K 字为单位指定来自 HGO 的偏置, 并从指定的 CPU 缓冲存储器作为 R/C 的输入获取。混合其他支持 iQ Platform 的 CPU (运动 CPU 及 NCCPU) 使用, 以确保各 CPU 中使用的 CPU 缓冲存储器不重复时, 根据需要设定。<br>设定值范围: -1~14 (整数)<br>(-1: 不使用 / 0~14 K 字)<br>(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)。<br>(B) 设定为 0~14 时, 根据值被设定为输入偏置。<br>※ 参照下述 “5.2.1 关于多 CPU 输入偏置 (仅限 CR800-R/Q 系列)” 案例 (A)、(B)。<br><br>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。<br>关于多 CPU 系统, 请参照 RCP 手册 (MELSEC iQ-R CPU 模块用户手册 (应用篇))。 | -1        |
| 多 CPU 输入偏置<br>(※ 仅限 CR800-Q 系列) |          |            | CR800-Q 系列时, 对多 CPU 的机器人输入信号的偏置进行设定。<br><br>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。<br>设定值范围: -1~14 (整数)<br>(-1: 不使用 / 0~14 K 字)<br>(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)。<br>(B) 设定为 0~14 时, 根据值被设定为输入偏置。<br>※ 参照下述 “5.2.1 关于多 CPU 输入偏置 (仅限 CR800-R/Q 系列)” 案例 (A)、(B)。<br><br>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。<br>关于多 CPU 系统, 请参照 QCPU 用户手册 (多 CPU 系统篇) SH (名) -080475。           |           |
| 信号输出的处理模式<br>注 1)               | SYNCIO   | 整数 1       | 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。<br>向上兼容模式/高速模式 1/高速模式 2=0/1/2<br>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。<br>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。<br>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。                                                                                                                                                                                                                                                      | 2: 高速模式 2 |
| 专用输入滤波器                         | IOFIL    | 整数 1       | 防止因专用输入信号的噪音等导致的误输入。<br>0: 滤波器无效<br>1: 滤波器有效                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0         |

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：在上述程序示例中，处理时间可缩短约 80%。

但是，在 CR800-D 系列的 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。在上述程序示例 2 中，处理时间可缩短约 75%。

<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CR800-R/Q系列）

(1) 案例（A）

不对输入使用偏置时（参数：QMLTCPUS = -1 时）

表 5-3：CPU 缓冲存储器与机器人输入输出信号的对应

| PLC（字软元件） |                              |                              | 机器人（位软元件） |                               |
|-----------|------------------------------|------------------------------|-----------|-------------------------------|
| 控制<br>器   | CR800-R 系列                   | CR800-Q 系列                   |           |                               |
| 输出        | U3E0\HG0 ~<br>U3E0\HG511     | U3E0\G10000 ~<br>U3E0\G10511 | 输入        | 机器人 CPU No. 1 / 10000 ~ 18191 |
|           | U3E0\HG512 ~<br>U3E0\HG1023  | U3E0\G10512 ~<br>U3E0\G11023 |           | 机器人 CPU No. 2 / 10000 ~ 18191 |
|           | U3E0\HG1024 ~<br>U3E0\HG1535 | U3E0\G11024 ~<br>U3E0\G11535 |           | 机器人 CPU No. 3 / 10000 ~ 18191 |
| 输入        | U3E1\HG0 ~<br>U3E1\HG511     | U3E1\G10000 ~<br>U3E1\G10511 | 输出        | 机器人 CPU No. 1 / 10000 ~ 18191 |
|           | U3E2\HG0 ~<br>U3E2\HG511     | U3E2\G10000 ~<br>U3E2\G10511 |           | 机器人 CPU No. 2 / 10000 ~ 18191 |
|           | U3E3\HG0 ~<br>U3E3\HG511     | U3E3\G10000 ~<br>U3E3\G10511 |           | 机器人 CPU No. 3 / 10000 ~ 18191 |

下图变为 CR800-R 系列时的存储器映射。CR800-Q 系列时，请参考表 5-3 改读为对应的设备编号。

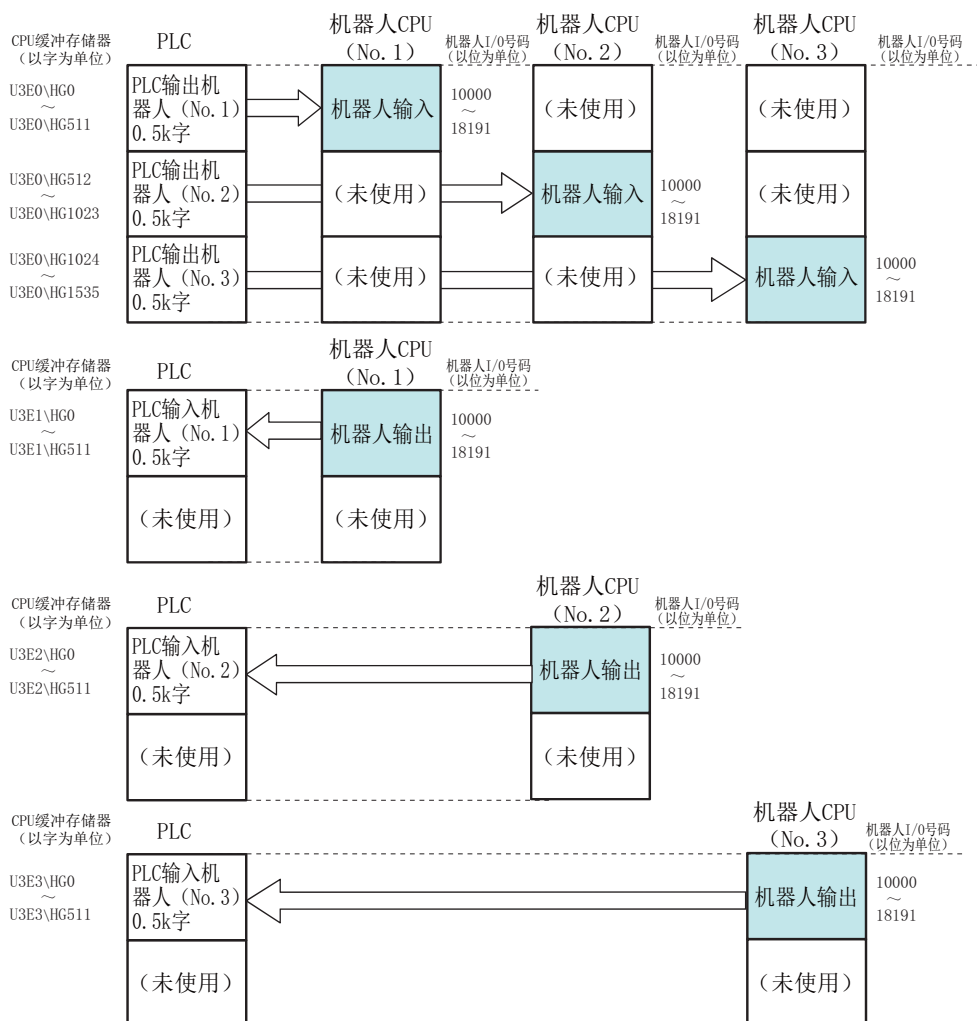


图 5-1：CPU 缓冲存储器与机器人输入输出信号的对应（案例（A））

## (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 缓冲存储器与机器人输入输出信号的对应

|         |                                                      | PLC (字软元件)                                               |            |                               |  |
|---------|------------------------------------------------------|----------------------------------------------------------|------------|-------------------------------|--|
| 控制<br>器 | CR800-R 系列                                           | CR800-Q 系列                                               | 机器人 (位软元件) |                               |  |
| 输出      | U3E0\HG0+ $\alpha$ (1) ~<br>U3E0\HG511+ $\alpha$ (1) | U3E0\G10000+ $\alpha$ (1) ~<br>U3E0\G10511+ $\alpha$ (1) | 输入         | 机器人 CPU No. 1 / 10000 ~ 18191 |  |
|         | U3E0\HG0+ $\alpha$ (2) ~<br>U3E0\HG511+ $\alpha$ (2) | U3E0\G10512+ $\alpha$ (2) ~<br>U3E0\G11023+ $\alpha$ (2) |            | 机器人 CPU No. 2 / 10000 ~ 18191 |  |
|         | U3E0\HG0+ $\alpha$ (3) ~<br>U3E0\HG511+ $\alpha$ (3) | U3E0\G11024+ $\alpha$ (3) ~<br>U3E0\G11535+ $\alpha$ (3) |            | 机器人 CPU No. 3 / 10000 ~ 18191 |  |
| 输入      | U3E1\HG0 ~ U3E1\HG511                                | U3E1\G10000 ~<br>U3E1\G10511                             | 输出         | 机器人 CPU No. 1 / 10000 ~ 18191 |  |
|         | U3E2\HG0 ~ U3E2\HG511                                | U3E2\G10000 ~<br>U3E2\G10511                             |            | 机器人 CPU No. 2 / 10000 ~ 18191 |  |
|         | U3E3\HG0 ~ U3E3\HG511                                | U3E3\G10000 ~<br>U3E3\G10511                             |            | 机器人 CPU No. 3 / 10000 ~ 18191 |  |

下图变为 CR800-R 系列时的存储器映射。CR800-Q 系列时，请参考表 5-4 改读为对应的设备编号。

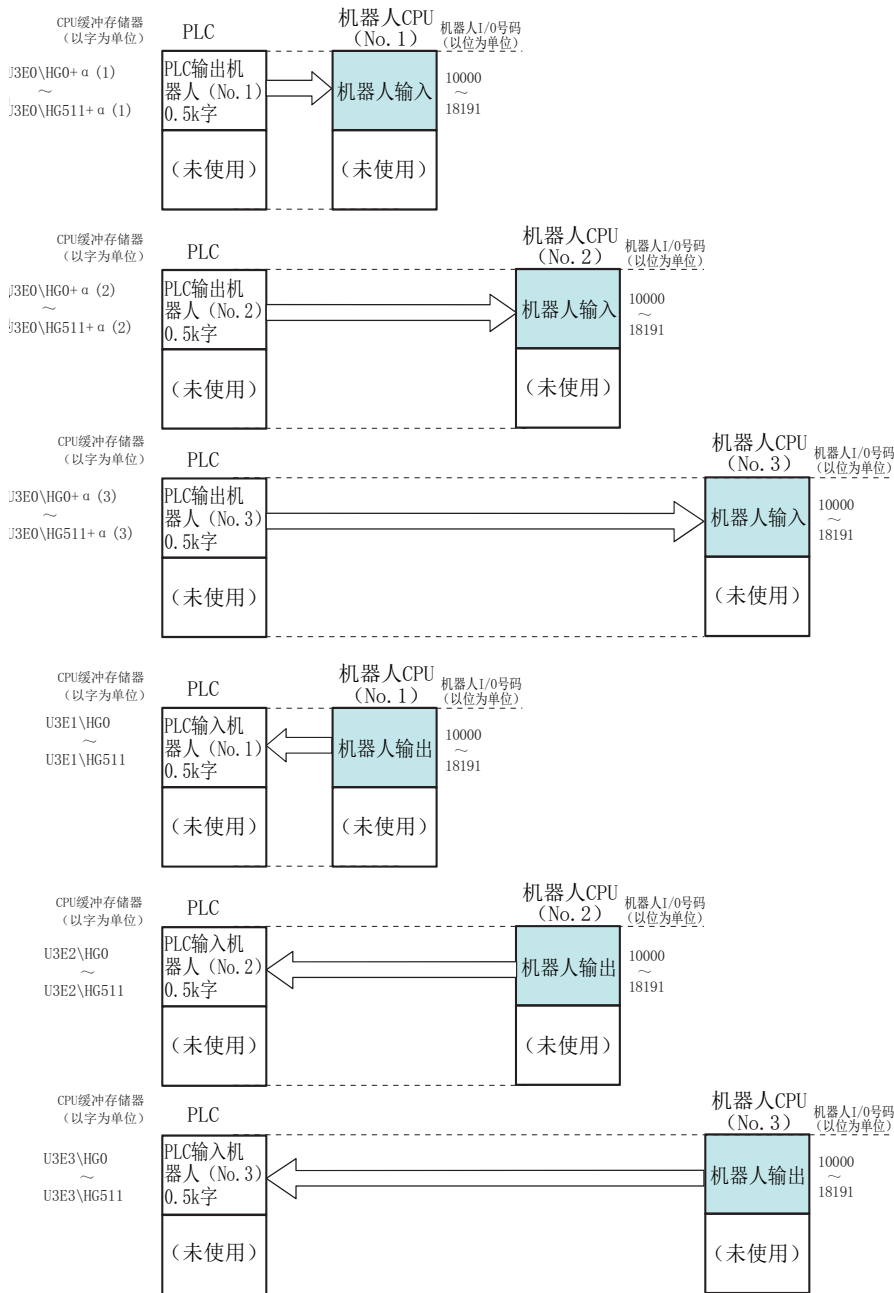


图 5-2: CPU 缓冲存储器与机器人输入输出信号的对应 (案例 (B))

## 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       | 切换控制器模式时, 5位LED显示模式的设定。<br>0: 更改模式时显示倍率修调。(初始值)<br>1: 即使更改模式, 也会继续保持当前的显示模式。<br>此外, 电源ON时的显示变为程序名称。                                                                                                                                                                                                                                                                                                                                                            | 0 (倍率修调显示) |           |           |                       |      |     |                    |     |     |        |
| 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)<br>如果设定为外部(0)时, 虽然操作面板有操作权(显示面板左上方的点状LED为亮灯状态), 但可以从外部信号选择程序。                                                                                                                                                                                                                                                                                                                                                    | 1 (OP)     |           |           |                       |      |     |                    |     |     |        |
|                       | RMTPSL    | 整数 1       | 外部输入输出有操作权时, 也可将通过操作面板选择程序的操作设为有效。<br>(外部/OP=0/1)<br>如果设定为OP(1)时, 即使操作面板没有操作权(显示面板左上方的点状LED为熄灯状态), 也可以从操作面板选择程序。                                                                                                                                                                                                                                                                                                                                               | 0 (外部)     |           |           |                       |      |     |                    |     |     |        |
| 限制从操作面板更改OVRD         | OPNOOVRD  | 整数 1       | 将通过控制器的操作面板进行的倍率修调更改设定为允许或禁止。<br>0: 允许<br>1: 禁止<br>设定为禁止(1)时, 即使操作面板为有操作权的状态(显示面板左上方的点状LED为亮灯状态), 也无法从操作面板更改倍率修调。                                                                                                                                                                                                                                                                                                                                              | 0 (允许)     |           |           |                       |      |     |                    |     |     |        |
| 显示面板的滚动速度             | OPSCRSPD  | 整数 1       | 更改操作面板LED显示的滚动速度。通常情况下, 应保持初始值状态。<br>调小时变快, 调大时变慢。<br>(仅限有操作面板的控制器)<br>设定范围: 1~1000[单位: ms]                                                                                                                                                                                                                                                                                                                                                                    | 120        |           |           |                       |      |     |                    |     |     |        |
| 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: auto; margin-right: auto;"> <thead> <tr> <th>控制器的模式</th> <th>OVRDENA=0</th> <th>OVRDENA=1</th> </tr> </thead> <tbody> <tr> <td>AUTOMATIC<br/>(示教单元无效)</td> <td>不可更改</td> <td>可更改</td> </tr> <tr> <td>MANUAL<br/>(示教单元有效)</td> <td>可更改</td> <td>可更改</td> </tr> </tbody> </table> | 控制器的模式     | OVRDENA=0 | OVRDENA=1 | AUTOMATIC<br>(示教单元无效) | 不可更改 | 可更改 | MANUAL<br>(示教单元有效) | 可更改 | 可更改 | 1 (必要) |
| 控制器的模式                | OVRDENA=0 | OVRDENA=1  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |            |           |           |                       |      |     |                    |     |     |        |
| AUTOMATIC<br>(示教单元无效) | 不可更改      | 可更改        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |            |           |           |                       |      |     |                    |     |     |        |
| MANUAL<br>(示教单元有效)    | 可更改       | 可更改        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |            |           |           |                       |      |     |                    |     |     |        |
| 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       | 设定从外部进行的倍率修调更改时是否需要操作权。<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)     |           |           |                       |      |     |                    |     |     |        |

| 参数                                           | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                            | 出货时设定值                                                  |
|----------------------------------------------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| 切换程序的存取对象                                    | ROMDRV   | 整数 1       | 程序的访问对象如下所示。<br>2= 高速 RAM 模式 (使用 DRAM 内存)<br>※ 不可更改                                                                                                             | 2 (DRAM)                                                |
| 维修预报设定                                       | MFENA    | 整数 1       | 对维护预报功能的有效 / 无效进行指定。<br>0: 无效<br>1: 有效<br>※ 指定为无效时, 不能进行维护预报功能相关的信息收集, 无法算出正确的维护时期。                                                                             | 1                                                       |
| 维修预报警告发生日                                    | MFWRNDAY | 整数 1       | 指定维护时间前的剩余天数。通过维护预报功能算出的剩余天数, 在本参数指定的天数以下的情况下, 发生警报。<br>设定范围: 1 ~ 365<br>单位: 日                                                                                  | 14                                                      |
| 每天的操作时间                                      | MFOPTIME | 实数 1       | 通过维护预报功能使用, 指定机器人 1 天的假设工作时间。<br>设定范围: 1 ~ 24<br>单位: 时间                                                                                                         | 16.0                                                    |
| 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>※ 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>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: 不可使用 / 1: 可使用                                                                                                                      | 1                                                       |



| 参数                           | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                   | 出货时设定值                                                      |
|------------------------------|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| 示教单元蜂鸣音                      | TBBZR    | 整数 1       | 对发生错误时示教单元蜂鸣音的ON/OFF进行指定。但是，无法关掉R32TB的按键音。<br>0: 蜂鸣音OFF<br>1: 蜂鸣音ON                                                                    | 1                                                           |
| 抓手开闭延迟时间                     | HNDT     | 整数 8       | 根据R32TB进行抓手关闭操作时,指定到抓手反应为止的延迟时间。在已指定的时间中需要持续按下按键,按下错误的按键时,抓手可能会立即不进行反应。<br>单位: [0.1秒]                                                  | (+C, -C, +B, -B, +A, -A, +Z, -Z)=<br>0, 0, 0, 0, 0, 0, 0, 0 |
| 与冗余系统错误相关的 [RESET] 键输入时的动作选择 | CATEGORY | 整数 1       | 设定在发生冗余系统错误时通过[RESET]键执行的动作。<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                                                    |
| 控制器版本信息<br>(仅限 CR800-R/Q 系列) | RCDUVER  | 字符串 1      | 在CR800-R/Q系列中,可确认控制器的软件版本。<br>在发生了版本不正确的错误(C0099)时,可用来确认版本信息。                                                                          | 根据版本不同而不同                                                   |

## 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」。</p> <p>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。</p> <p>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。)</p> <p>起动条件: 通常 / 报警时 / 通常=START/ERROR/ALWAYS<br/>           START: 依据&lt;操作面板&gt;的启动键及信号的启动信号执行。<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</p> <p>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 |

| 参数                                             | 参数名    | 排列数<br>文字数                                                               | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 出货时设定值                                                                                                                                        |
|------------------------------------------------|--------|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 程序选择记忆                                         | SLOTON | 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。</p> <p>(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。</p> <p>(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。</p> <p>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)                                                                                                                                  |
| 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                                                                     | <p>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。</p> <p>可 / 不可=1 / 0</p>                                                                                                                                                                                                                                                                                                                                                             | 0 (不可)                                                                                                                                        |
| 使用者基本程序<br>请参照“4.3.24 用户定义外部变量”。               | PRGUSR | 字符串 1                                                                    | <p>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。</p> <p>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。</p>                                                                                                                                                                                                                                                                                                    | " " (无)                                                                                                                                       |
| 继续功能                                           | CTN    | 整数 1                                                                     | 备用                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0 (无效)                                                                                                                                        |
| JRC 指令<br>(轴的多旋转功能)                            |        |                                                                          | 设定 JRC 指令的执行状态。                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                               |
|                                                | JRCXE  | 整数 1                                                                     | <p>指定 JRC 指令的执行可 / 不可。</p> <p>执行可 / 不可 = (1 / 0)</p>                                                                                                                                                                                                                                                                                                                                                                                                                          | 0 (执行不可)                                                                                                                                      |
|                                                | JRCQT  | 实数 8                                                                     | <p>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。</p> <p>单位依存在参数 AXUNT。</p>                                                                                                                                                                                                                                                                                                                                            | <p>JRC 执行可能<br/>机器人<br/>0, 0, 0, 0, 0, 360, 0, 0</p> <p>或<br/>0, 0, 0, 360, 0, 0, 0, 0</p> <p>JRC 执行不可<br/>机器人<br/>0, 0, 0, 0, 0, 0, 0, 0</p> |
| JRCORG                                         | 实数 8   | <p>执行 JRC 指令后设定原点设定时的原点坐标值。</p> <p>只在使用者定义轴有效。</p> <p>单位依存在参数 AXUNT。</p> | 0, 0, 0, 0, 0, 0, 0, 0                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                               |
| 附加轴设定                                          | AXUNT  | 整数 16                                                                    | <p>设定附加轴的单位。</p> <p>0: 角度 (degree)<br/>1: 长度 (mm)<br/>2: 长度 (mm) 线性伺服使用<br/>使用线性伺服时，请设定为 2。</p>                                                                                                                                                                                                                                                                                                                                                                               | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 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                               |
|---------------------|-----------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| 使用者报警设定             | UER1<br>~ UER20 | 整数 1<br>字符串 3 | 设置因Error指令导致的错误的信息、原因、复位方法。<br>最多可以设定20个用户错误。<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（更改时，也请再次将全体文字用“”圈住。）<br>例）9000,“Time Out”,“No Signal”,“Check Button”<br>· 信息中无法使用逗号（,）、分号（;）、货币符号（¥）。<br>· 第1位的编号可当作辅助编号。例如，将信息登录为9000号信息的话，发生9001~9009错误时，显示登录为9000号的信息。 | 9900,“message”,“cause”,“treat”       |
| 位置数据的旋转成份的单位设定      | PRGMDEG         | 整数 1          | 在机器人程序内指定位置数据的旋转成分的单位。<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)                              |
| 小数点以后的位数            | PRGDPNTM        | 整数 1          | 指定位置型变量及关节型变量小数点以后的位数。<br>指定2或3行。                                                                                                                                                                                                                                                                                                         | 3: RH-FRH、RH-CRH、RH-3FRHR<br>2: 上述以外 |
| 机器人语言               | RLNG            | 整数 1          | 选择使用的机器人语言。<br>3: MELFA-BASIC VI<br>1: MELFA-BASIC IV                                                                                                                                                                                                                                                                                     | 3                                    |
| 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 R32TB 或 R56TB 的情况下，在个别专用的画面里设定。<br>(R32TB 请参照本书的第 102 页的“3.19 初始设定画面的操作”、R56TB 请参照另一手册的操作说明书“R56TB/R57TB 操作说明书”)                                                                 | 日文规格的情况为“JPN”<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。                                                                                                                                                                                                                                                                                  | 1                                    |
| 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                                    |
| 程序冗余配置功能            | PRGDUP          | 整数 1          | 指定程序的冗余配置。由于冗余配置，因噪音等导致程序异常时，可自动修复。<br>0: 不进行冗余配置<br>1: 进行冗余配置<br>注) 由于程序的冗余配置，程序的保存容量减少为原来的 1/2。                                                                                                                                                                                                                                         | 0                                    |

| 参数                       | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                               | 出货时设定值  |
|--------------------------|----------|------------|------------------------------------------------------------------------------------------------------------------------------------|---------|
| 自动运行时的程序自动保存             | AUTOSAVE | 整数 1       | 指定程序运行后局部变量的自动保存。在以下时机进行自动保存。<br>· 通过 CallP 指令调用的子程序的 End 执行指令时<br>· 切换程序时<br>· 执行 XClr 指令时<br>· 循环运行后清除程序时<br><br>0: 不保存<br>1: 保存 | 1       |
| 其他程序读取指令与 Function 程序的并用 | FUNCSPEC | 整数 1       | 设定是否同时使用其他程序读取指令 (CallP、XLoad、XRun、#Include)。<br>变更为了 1 的情况下, 使用其他程序读取指令时, 节拍时间可能会变长。<br><br>0: 不使用<br>1: 使用                       | 0 (不使用) |

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。



| 参数                          | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                                  | 出货时设定值                                                                                                                                               |
|-----------------------------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ethernet 数据链接用<br>数据包规格     | NETPSPEC | 整数 1       | 指定通过Ethernet进行的数据链接通信的数据包接受时的规格。<br>0: 以往规格<br>指定为不使用结束CODE接收时。接收1个数据包的数据前, 等待Input指令。分割数据包的情况下, 无法正常接受。<br>1: 新规格<br>指定为使用结束CODE接收时。即使已分割数据包, 到接收结束CODE前, 等待Input指令。 | 0                                                                                                                                                    |
| Ethernet 通信协议终端<br>CODE     | NETTERM  | 整数 9       | 指定使用通过Ethernet进行的数据链接通信的终端CODE。<br>要素1~9对应COM1~COM9。<br>0: 无终端CODE<br>1: 有终端CODE                                                                                      | 0, 0, 0, 0, 0, 0, 0,<br>0, 0                                                                                                                         |
| GOT 端口编号                    | GOTPORT  | 整数 1       | 指定GOT直接连接中的端口编号。应符合GOT的端口编号。不更改GOT初始设定的情况下保持为初始设定的设定值。                                                                                                                | 5001                                                                                                                                                 |
| 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       | 实时外部控制指令执行时的超时时间<br>CR800-Q系列: 7.1ms的倍数<br>CR800-D/R系列: 3.5ms (设定了用户机械时为7.1ms)的倍数<br>-1时无超时                                                                           | -1                                                                                                                                                   |
| Job Load 超时时间               | NVJBOUT  | 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                                                                                                                                                   |
| 通过 EBRead 指令指定<br>的标记名的初始值  | EBRD TAG | 字符串 1      | 通过EBRead指令设定指定的视觉传感器标识符号名的初始值<br>(128字符以内)。<br>根据EBRead指令的记述省略符号名的情况下指定本参数的设定值。                                                                                       | "Job.Robot.<br>FormatString"                                                                                                                         |
| 通过 EBWrite 指令指定<br>的标记名的初始值 | EBWRTAG  | 字符串 1      | 通过EBWrite指令设定指定的视觉传感器标识符号名的初始值<br>(128字符以内)。<br>根据EBWrite指令的记述省略符号名的情况下指定本参数的设定值。                                                                                     | " " (无)                                                                                                                                              |

| 参数                     | 参数名       | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 出货时设定值                       |       |       |   |    |    |   |           |           |   |    |           |   |
|------------------------|-----------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-------|-------|---|----|----|---|-----------|-----------|---|----|-----------|---|
| 用户名                    | NVUSER    | 字符串 1      | 为登录视觉传感器而设定用户名(15个字符以内)。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | “admin”                      |       |       |   |    |    |   |           |           |   |    |           |   |
| 密码                     | NVPSWD    | 字符串 1      | 为登录视觉传感器而设定密码(15个字符以内)。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | “ ”                          |       |       |   |    |    |   |           |           |   |    |           |   |
| 触发时机                   | NVTRGTMG  | 整数 1       | <p>规定NVRun指令与NVTrg指令的处理内容。<br/>每个设定值的处理内容如下表所示。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>设置值</th> <th>NVRun</th> <th>NVTrg</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>触发</td> <td>触发</td> </tr> <tr> <td>1</td> <td>触发 + 图像处理</td> <td>触发 + 图像处理</td> </tr> <tr> <td>2</td> <td>触发</td> <td>触发 + 图像处理</td> </tr> </tbody> </table> <p>●触发 . . . 在与视觉传感器的图像处理指令(摄像请求)的通信结束时移动至下一个命令执行。<br/>视觉传感器进行图像处理期间用机器人实施其他作业时,进行本设定时可缩短节拍时间。</p> <p>●触发 + 图像处理 . . . 在视觉传感器进行图像处理指令(摄像请求),视觉图像处理完成后移动至下一个命令执行。<br/>请向视觉传感器请求图像处理,通过下一步取得识别结果时(执行EBRead指令时)进行本设定。</p> <p>在NVRun指令后记述EBRead指令时,请将参数NVTRGTMG的值设为1。<br/>参数NVTRGTMG为出场设定(NVTRGTMG = 2)时,NVRun指令不等待视觉识别处理完成,移动至下一个指令处理。因此继续执行EBRead指令时可能获取上一次的识别结果。</p> | 设置值                          | NVRun | NVTrg | 0 | 触发 | 触发 | 1 | 触发 + 图像处理 | 触发 + 图像处理 | 2 | 触发 | 触发 + 图像处理 | 2 |
| 设置值                    | NVRun     | NVTrg      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                              |       |       |   |    |    |   |           |           |   |    |           |   |
| 0                      | 触发        | 触发         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                              |       |       |   |    |    |   |           |           |   |    |           |   |
| 1                      | 触发 + 图像处理 | 触发 + 图像处理  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                              |       |       |   |    |    |   |           |           |   |    |           |   |
| 2                      | 触发        | 触发 + 图像处理  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                              |       |       |   |    |    |   |           |           |   |    |           |   |
| 通过 EBRead 指令指定的符号名的初始值 | EBRDTAG   | 字符串 1      | 设定通过EBRead指令指定的视觉传感器标识符号名的初始值(128字符以内)。<br>根据EBRead指令的记述省略符号名的情况下指定本参数的设定值。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | “Job.Robot.<br>FormatString” |       |       |   |    |    |   |           |           |   |    |           |   |
| IP 地址过滤功能有效 / 无效指定     | NETIPFLT  | 整数 1       | <p>对IP地址过滤功能的有效/无效进行指定。<br/>有效时,可以任意设定为放行或阻断。对象IP地址可以通过参数NETIPFLS和NETIPFLE进行指定。<br/>放行: 允许来自指定的IP地址的访问。<br/>阻断: 阻断来自指定的IP地址的访问。</p> <p>0 无效(不使用IP地址过滤功能。允许所有访问。)<br/>1 有效/放行(允许来自指定的IP地址的访问。)<br/>2 有效/阻断(禁止来自指定的IP地址的访问。)</p> <p>将本功能设为了有效时,应使起始地址与结束地址为同一网络地址,且地址大小不可逆转。设定错误的情况下,接通电源时将发生错误且本功能将为无效。</p>                                                                                                                                                                                                                                                                                                                                                                                                                            | 0                            |       |       |   |    |    |   |           |           |   |    |           |   |
| IP 地址过滤功能起始地址          | NETIPFLS  | 字符串 1      | 对IP地址过滤功能的起始地址进行指定。<br>在IP地址过滤功能的有效/无效指定的参数NETIPFLT为有效(1或2)时可以使用。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 192.168.0.2                  |       |       |   |    |    |   |           |           |   |    |           |   |
| IP 地址过滤功能结束地址          | NETIPFLE  | 字符串 1      | 对IP地址过滤功能的结束地址进行指定。<br>在IP地址过滤功能的有效/无效指定的参数NETIPFLT为有效(1或2)时可以使用。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 192.168.0.2                  |       |       |   |    |    |   |           |           |   |    |           |   |



## 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，必须要设定工具转换数据。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工具转换数据。

详细内容请参照第 392 页的“M\_Tool”

出货时设定在 0，控制点为机械 I/F（法兰面）。

工具转换数据的构造：X, Y, Z, A, B, C

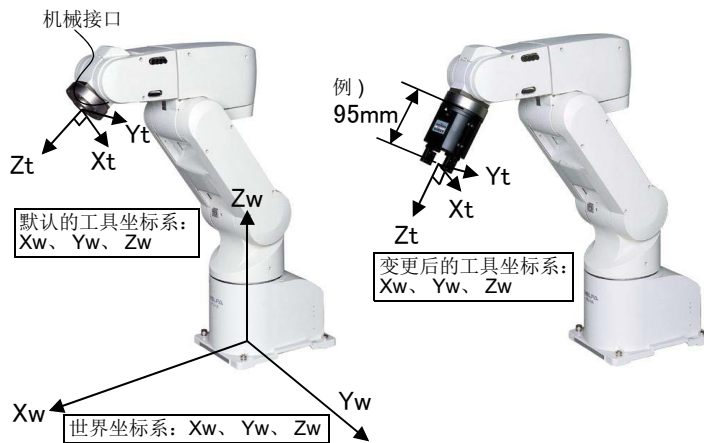
X, Y, Z 轴：以从机械 I/F 来的 TOOL 坐标的值

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

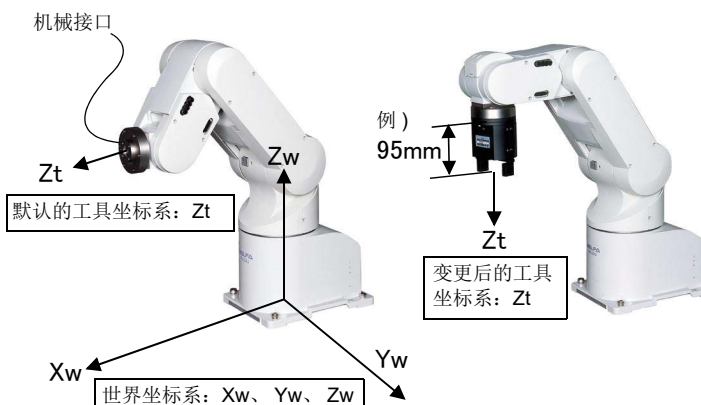
值: 0, 0, 95, 0, 0, 0

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工具坐标系的方向不同，请特别注意。

工具转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: 对应机器人的形式工具转换数据有效的轴成份

| 形式                                              | 轴数 | 工具转换数据的轴成份 <sup>注 1)</sup> |   |   |   |   |   |
|-------------------------------------------------|----|----------------------------|---|---|---|---|---|
|                                                 |    | X                          | Y | Z | A | B | C |
| RV-2FR、RV-4FR/4FRL、RV-7FR 系列、RV-13FR 系列、RV-20FR | 6  | ○                          | ○ | ○ | ○ | ○ | ○ |
| RV-4FRJL                                        | 5  | △                          | △ | ○ | △ | △ | × |
| RH-3FRH/6FRH/12FRH/20FRH 系列、RH-3FRHR 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

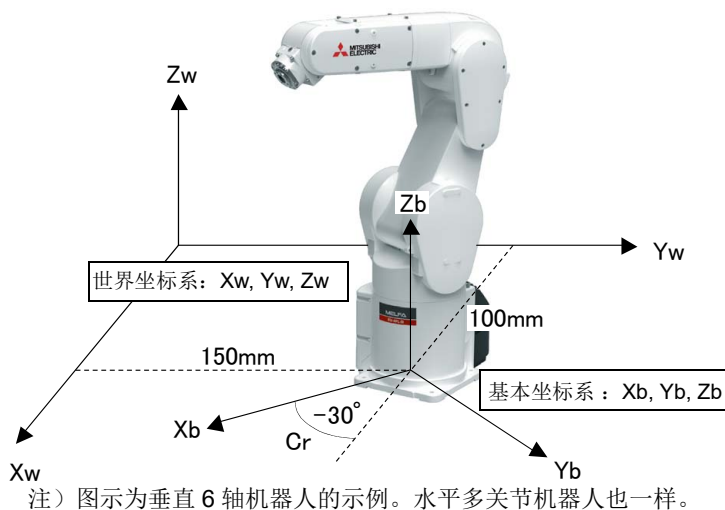
基本坐标系数据的构造：X, Y, Z, A, B, C

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



(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。

基本转换数据的轴成分，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转换数据的有效轴成份

| 形式                                              | 轴数 | 基本转换数据的轴成份 <sup>注1)</sup> |   |   |   |   |                  |
|-------------------------------------------------|----|---------------------------|---|---|---|---|------------------|
|                                                 |    | X                         | Y | Z | A | B | C                |
| RV-2FR、RV-4FR/4FRL、RV-7FR 系列、RV-13FR 系列、RV-20FR | 6  | ○                         | ○ | ○ | ○ | ○ | ○                |
| RV-4FRJL 系列                                     | 5  | ○                         | ○ | ○ | △ | △ | ○ <sup>注2)</sup> |
| RH-3FRH/6FRH/12FRH/20FRH 系列、RH-3FRHR 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。

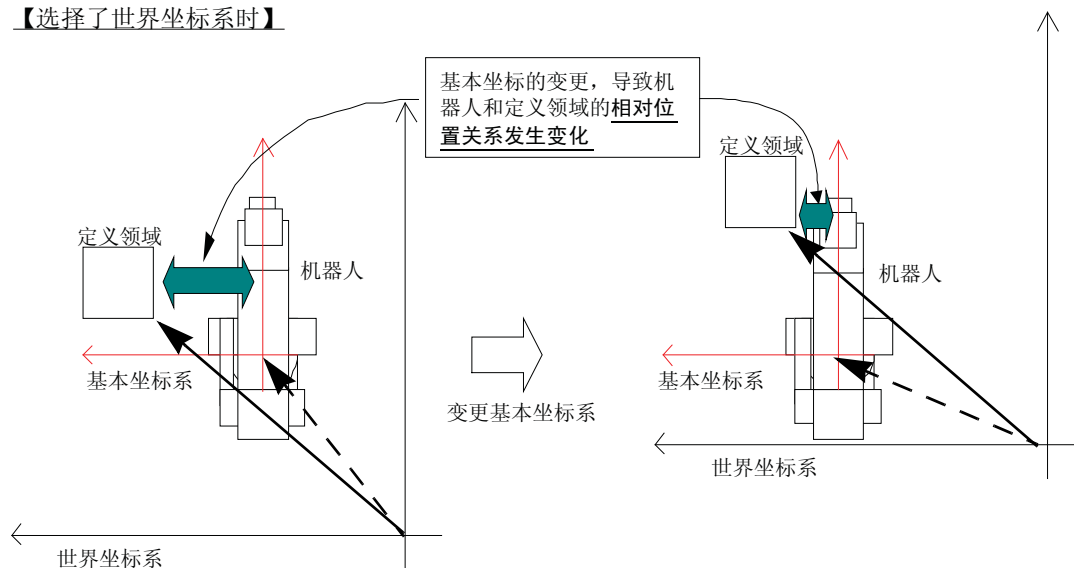
不变更基本坐标系时，无论选择哪一个，使用者定义领域都不会变化。

关于坐标系的详细内容，请参照“4.4 机器人的坐标系说明”。

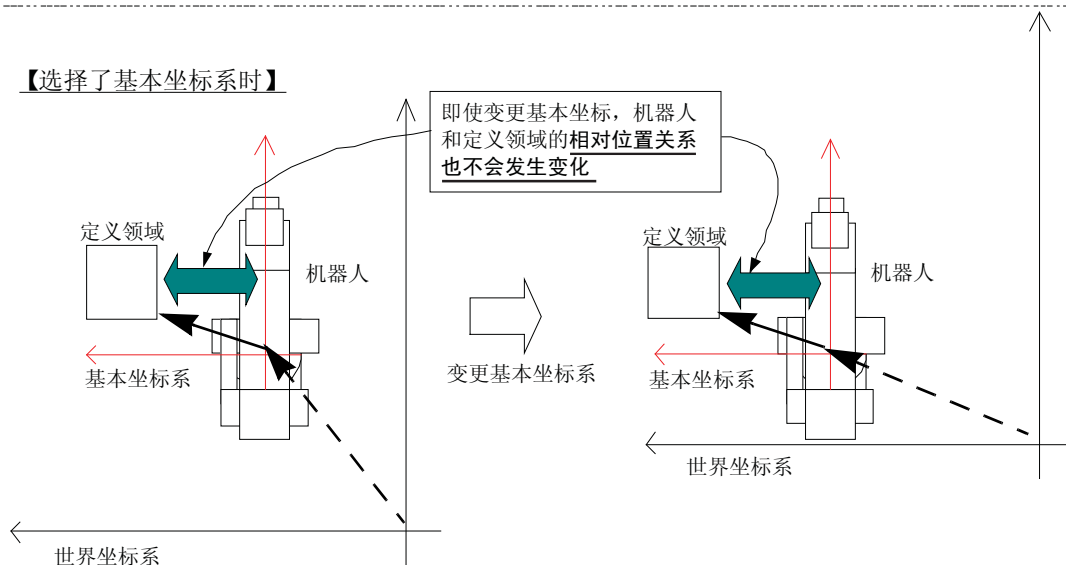
表 5-10：作为基准的坐标系

| 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) 图示以垂直 6 轴机器人为例，从正上方看到的图像。水平多关节机器人也一样。

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

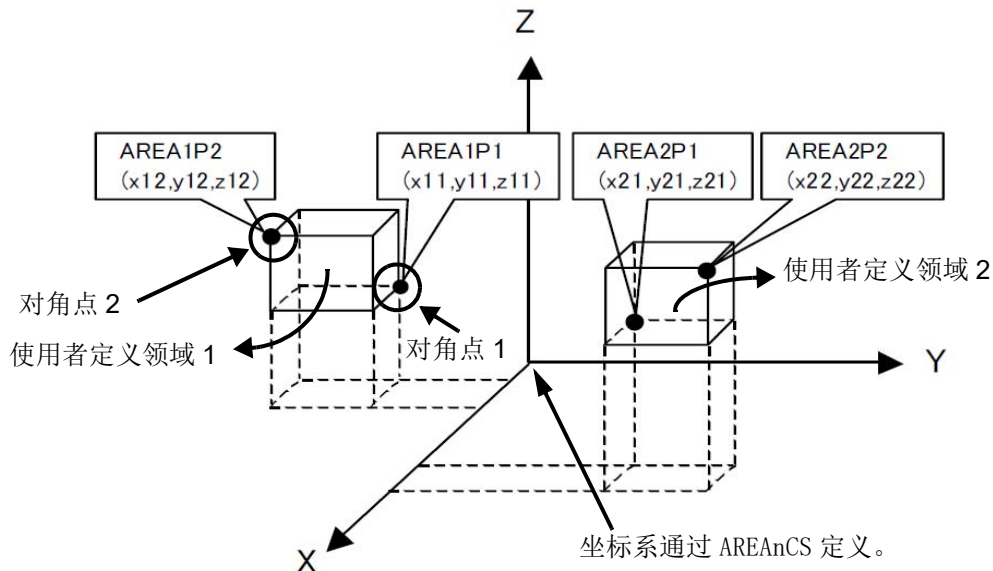
领域分为位置领域、姿势领域和附加轴领域。  
下面介绍各领域的详细设定内容。

## (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。

< 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 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。

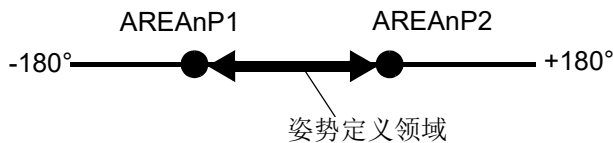
### ⚠ 注意

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。

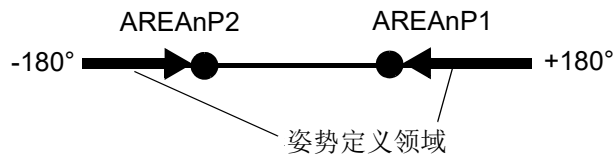
#### < 注意 >

- 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 0.001mm 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 0.001mm 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

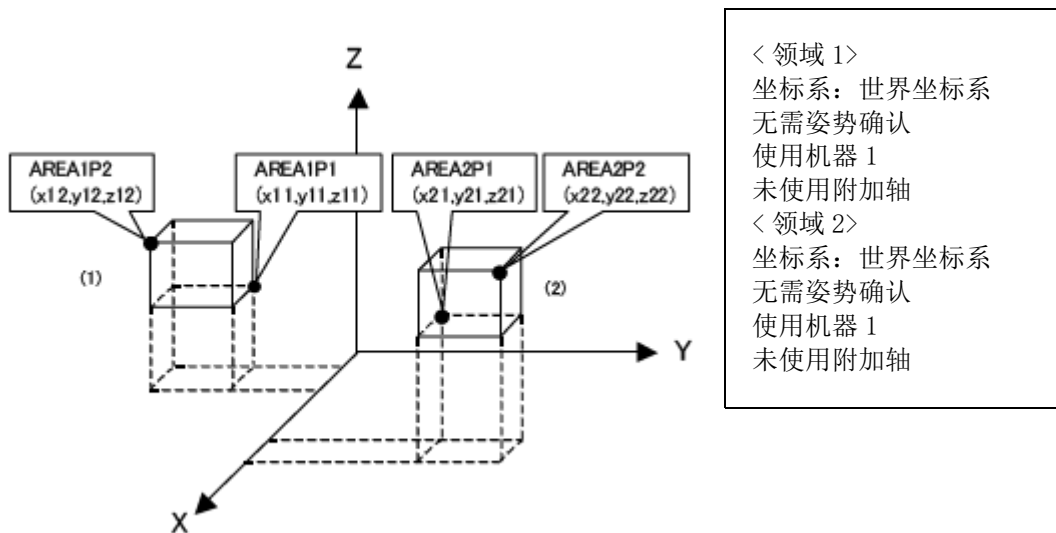
从表 5-11 中选择机器人是否处于使用者定义领域内的状态设定，通过参数 AREAnAT 的设定进行指定。

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。



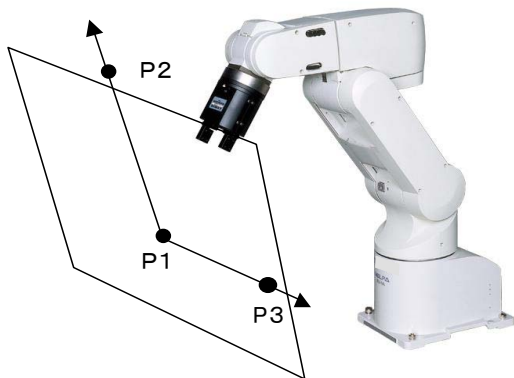
| 参数名     | 值的意义                                | 值                                     |
|---------|-------------------------------------|---------------------------------------|
| AREA1CS | 选择领域 1 的坐标系                         | 0                                     |
| AREA1P1 | 领域 1 的对角点 1 位置数据: X、Y、Z、A、B、C、L1、L2 | 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。

注) 图示为垂直 6 轴机器人的示例。水平多关节机器人也一样。

| 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

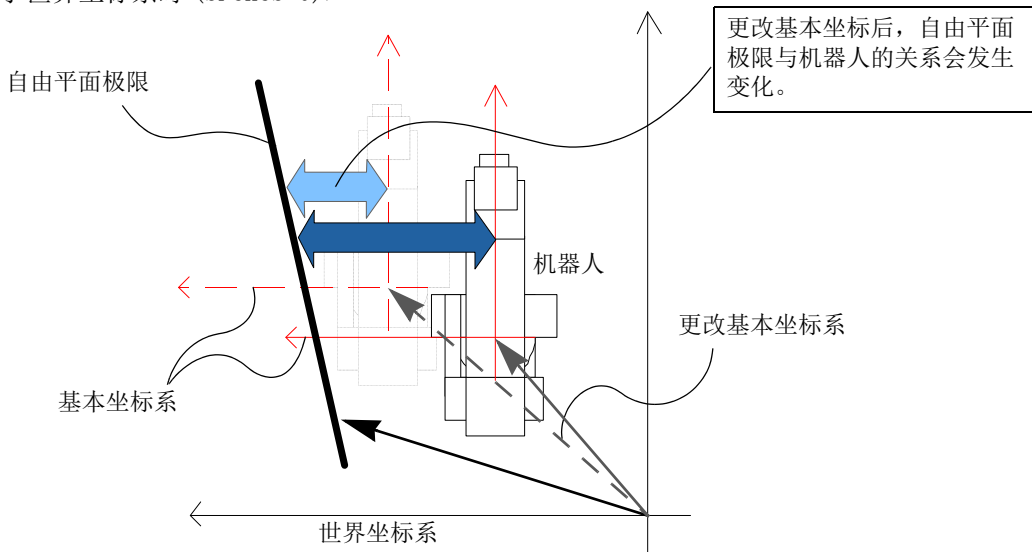
该功能是，使用 Base 指令等更改基本坐标系以使用时，选择是同时移动自由平面极限还是将其固定的功能。选择时是通过参数：SFCnCS 将基准坐标系指定为“世界坐标系”（移动）或“基本坐标系”（事先固定）的任意一个。

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。

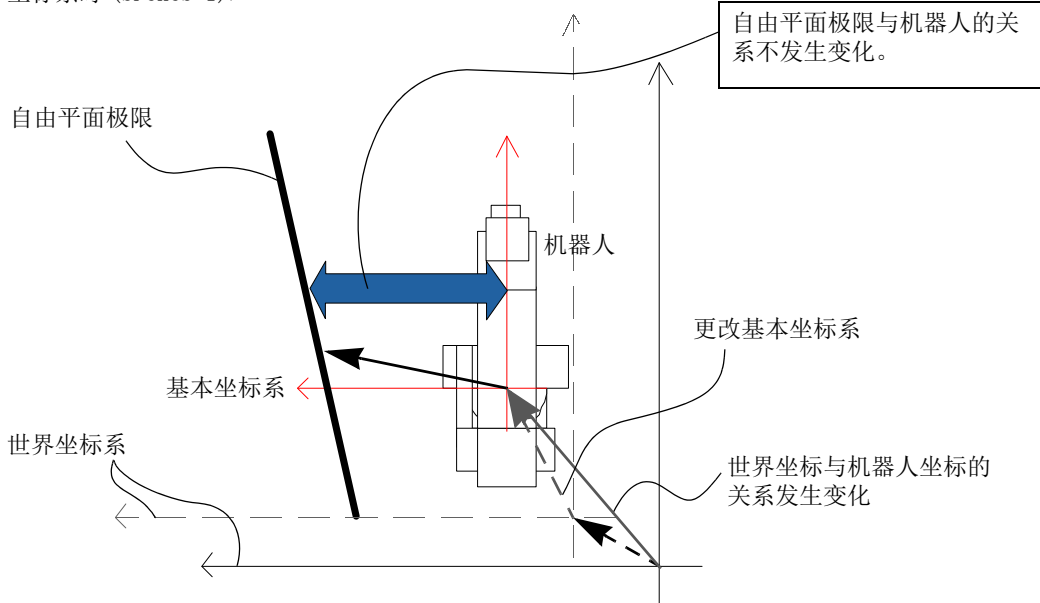
关于坐标系的详细内容，请参照第 155 页的“4.4 机器人的坐标系说明”。

| 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) >



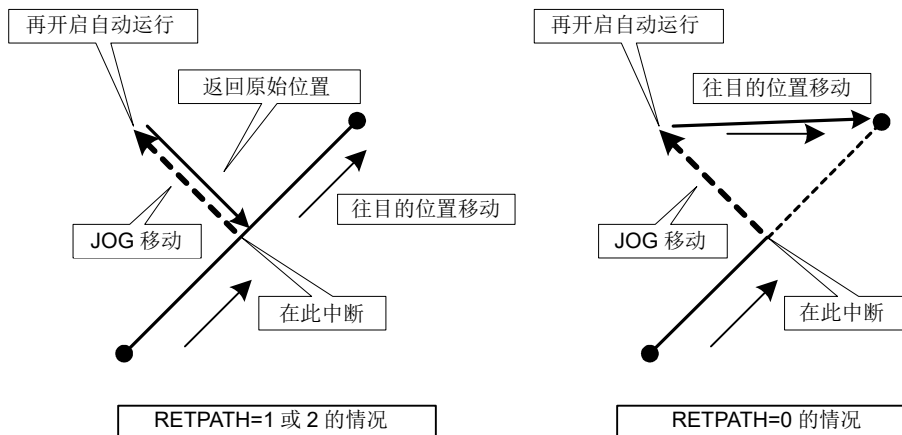
< 选择了基本坐标系时 (SFCnCS=1) >



## 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。

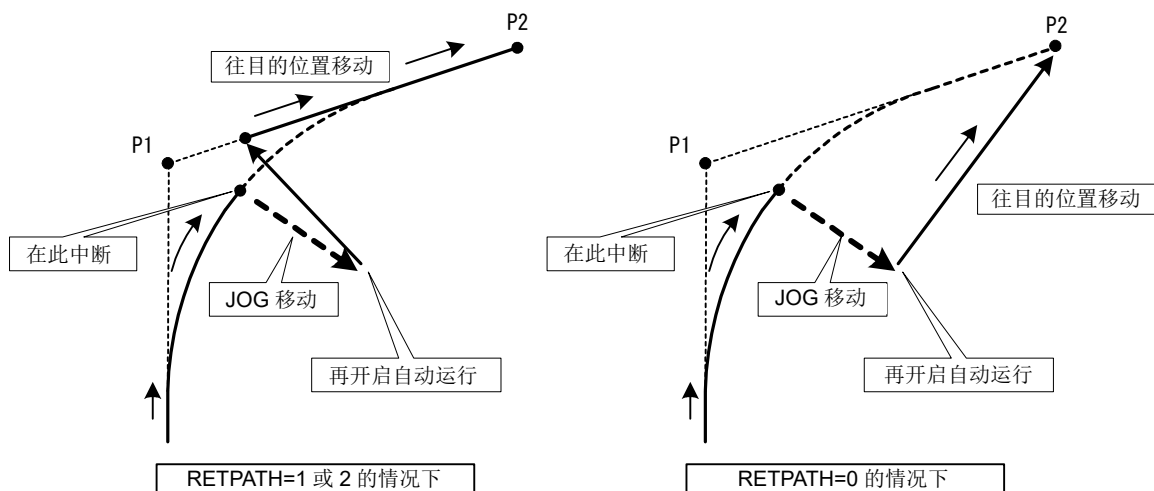
| 参数和值                | 动作说明                                                  |
|---------------------|-------------------------------------------------------|
| RETPATH=1 (Default) | 1) 以关节插补返回到原来的中断位置。<br>2) 重新开始中断的行。                   |
| RETPATH=0           | 从以 JOG 移动的位置开始再开始中断行，因此，从现在的位置到下一个目的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“AUTOMATIC”的时候，执行 Prg.1。
4 ' “AUTOMATIC”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控制器电源，使控制器模式变为“AUTOMATIC”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

出厂时，预设使用双螺线管类型的电磁阀。

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: 未设定<br>1: 漏型<br>0: 源型                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| HANDTYPE | 工厂出货时的设定值:<br>CR800-D/R/Q: D900, D902, D904, D906, , , ,<br>CR860-D/R/Q: D764, D766, D768, D770, , , ,<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>CR800-D/R/Q:<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>CR860-D/R/Q:<br>抓手 1 = 存取信号号码 764、765<br>抓手 2 = 存取信号号码 766、767<br>抓手 3 = 存取信号号码 768、769<br>抓手 4 = 存取信号号码 770、771<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

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 |           |
|---------------------------|----------|-----------|-----------|
|                           |          | CR800 控制器 | CR860 控制器 |
| 安装气动抓手 I/F 时<br>(假设为双线螺管) | 抓手 1 = 开 | 900=1     | 764=1     |
|                           |          | 901=0     | 765=0     |
|                           | 抓手 2 = 开 | 902=1     | 766=1     |
|                           |          | 903=0     | 767=0     |
|                           | 抓手 3 = 开 | 904=1     | 768=1     |
|                           |          | 905=0     | 769=0     |
|                           | 抓手 4 = 开 | 906=1     | 770=1     |
|                           |          | 907=0     | 771=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                                                                                                        |
|----------|----------------------------------------------------------------------------------------------------------|
|          | CR800 控制器 : 900, 901, 902, 903, 904, 905, 906, 907<br>CR860 控制器 : 764, 765, 766, 767, 768, 769, 770, 771 |
| 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 |

[ 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前端抓手专用信号的电源为 ON 时的初始值。

使用通用 IO 控制进行抓手控制的情况下，设定电源置为 ON 时的初始值时，不使用 HANDINIT 参数，应使用 ORS\* 参数进行设定。在 ORS\* 参数中设定的值将为电源置为 ON 时的信号初始值。

## 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> <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                         | ORS10000                            | 信号号码<br>10000---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                                                                                          |                                     |
|                                     | L<br>I<br>N<br>K<br>注1)             | ORS10128                                                                                                                     | 00000000、00000000、00000000、00000000 |
|                                     |                                     |                                                                                                                              | 00000000、00000000、00000000、00000000 |
| 00000000、00000000、00000000、00000000 |                                     |                                                                                                                              |                                     |
| 00000000、00000000、00000000、00000000 |                                     |                                                                                                                              |                                     |
| 00000000、00000000、00000000、00000000 |                                     |                                                                                                                              |                                     |
| ORS18160                            | 00000000、00000000、00000000、00000000 |                                                                                                                              |                                     |

注 1) 可编程控制器链接只有 CR800-R/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> 1000、 | 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>1</b> 0000000、  | 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 ToolBox3（数据链接也可使用）  | NETPORT           |
| 10002 ~ 10009 | 数据链接（RT ToolBox3 也可使用） | NETPORT           |
| 11000         | R56TB                  | 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 控制周期的倍数。1 控制周期在 CR800-Q 系列时约 7.1ms，在 CR800-D/R 系列时约 3.5ms (设定用户机械时约 7.1ms)。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，将通过错误 (H7820) 停止。例如，在约 7 秒内无通信时要设为错误时，设定为 2000 (CR800-D/R 系列) 或 1000 (CR800-Q 系列或 CR800-D/R 系列中设定用户机械时)。

出厂时，设定为无超时 (-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。

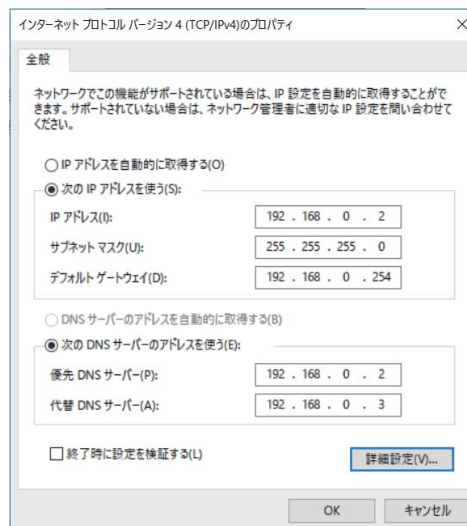
| 項目         | 設定値          |
|------------|--------------|
| 控制器的 IP 地址 | 192.168.0.20 |
| 计算机的 IP 地址 | 192.168.0.2  |
| 控制器的端口号码   | 10001        |

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。Windows10（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。

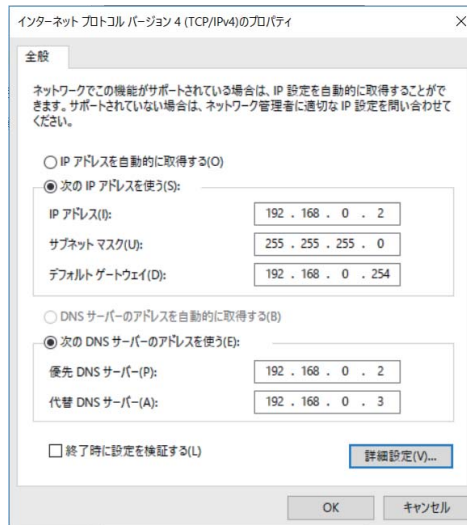
5.15.3 参数设定示例2-1（使用数据链路功能：控制器为服务器时）

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。  
Windows10（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。

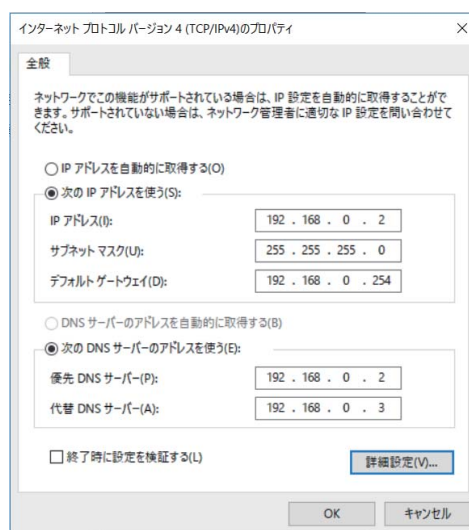
## 5.15.4 参数设定示例2-2（使用数据链路功能：控制器为客户机时）

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     | <u>2</u>                                                                                                                           |
| COMDEV   | 前     | RS232, , , , , ,                                                                                                                   |
|          | 后     | RS232, , <u>OPT13</u> , , , , ,                                                                                                    |
| NETMODE  | 前     | 1, 1, 1, 1, 1, 1, 1, 1, 1                                                                                                          |
|          | 后     | 1, 1, <u>0</u> , 1, 1, 1, 1, 1, 1                                                                                                  |
| NETHSTIP | 前     | 192.168.0.2, 192.168.0.3, 192.168.0.4, 192.168.0.5,<br>192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9,<br>192.168.0.10         |
|          | 后     | 192.168.0.2, 192.168.0.3, <u>192.168.0.2</u> , 192.168.0.5,<br>192.168.0.6, 192.168.0.7, 192.168.0.8, 192.168.0.9,<br>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。  
Windows10（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。

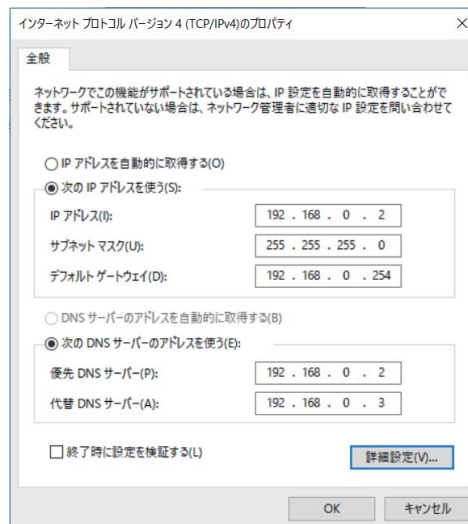
## 5.15.5 参数设定示例3（使用实时外部控制功能）

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。  
Windows10（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。

```

C:\>ping 10.97.11.31 ←コントローラのIPアドレス
Pinging 10.97.11.31 with 32 bytes of data:
Reply from 10.97.11.31: bytes=32 time<10ms TTL=128
Reply from 10.97.11.31: bytes=32 time<10ms TTL=128
Reply from 10.97.11.31: bytes=32 time<10ms TTL=128
Reply from 10.97.11.31: bytes=32 time<10ms TTL=128
C:\>
通信異常時は、Request time outを表示

```

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、Oad1 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

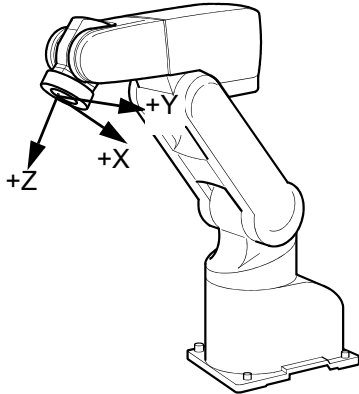
『1』 = 设定为把持

- 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 成份和  
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）

#### (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

为了提高处理速度，执行程序时进行 DRAM 运行（高速 RAM 运行）。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

机器人程序将被保存至 FLASH ROM 中。

程序的执行是用于提高处理速度的 DRAM 运行，程序中所使用的变量在电源切断时的保存上存在部分限制。

#### 限制内容

DRAM 运行：保存程序外部变量和使用者定义外部变量。本地变量被废弃。

### (2) 关于断电时的变量保存的注意事项

高速 RAM 运行时可变更程序执行中的变量值，但变更后的值会因控制器断电而被废弃，请注意。通过下述方法可在断电时保存变量值。

#### ※ 需要在断电时也保存变量值的情况：

使用程序外部变量和使用者定义外部变量，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（执行至 CallP 对象的 End 指令）。但使用者定义外部变量除外。

表 5-15: 高速 RAM 运行（DRAM 运行）中的变量保存

| 变量 注1)    | 高速 RAM 运行中<br>(DRAM 运行)                                                     |
|-----------|-----------------------------------------------------------------------------|
| 本地变量      | 断电时执行的程序的变量值将被废弃。<br>程序选择时或 CALLP 指令结束时将被保存。此外，通过使用 Save 指令可保存包含局部变量的机器人程序。 |
| 程序外部变量    | 变量的值在断电后也保持不变。                                                              |
| 使用者定义外部变量 | 变量的值在断电后也保持不变。                                                              |

注 1) 各个变量中有数值变量、字符串变量、位置变量、关节变量。

## 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回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。通过工厂出货时的参数设定，暖机运行倍率超调速度比例会配合对象轴的动作时间如（图 5-5）所示值发生变化。

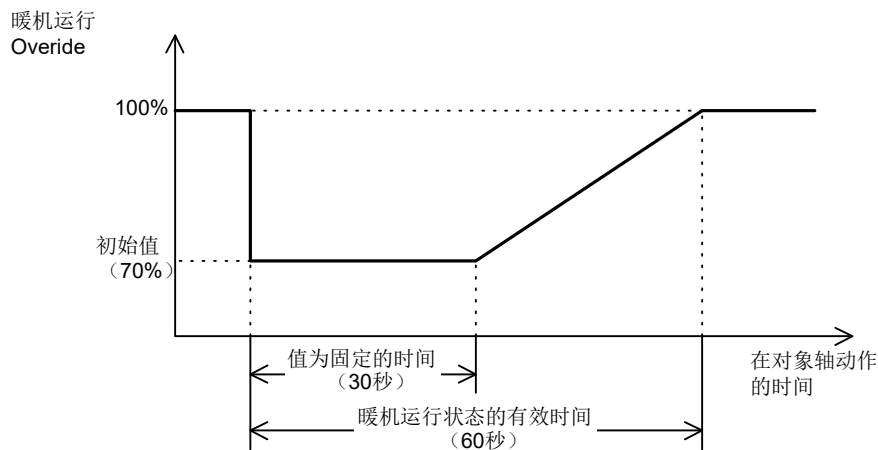


图 5-5：暖机运行倍率超调的变化

### ⚠ 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。

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。关于详细内容请参照第 468 页的“5.1 动作参数”、第 605 页的“6.2PLC 连接输入输出功能”、第 106 页的“4MELFA-BASIC VI”。

表 5-16: 暖机运行模式的参数一览

| 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 5-17: 暖机运行模式的专用输入输出信号一览

| 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 5-18: 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

现在是否在暖机运行状态，或者是在通常的状态，可以下列 2 种办法确认。

- 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，对像轴长时间停止持续，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示例以时序图来表示，如图 5-6 所示。

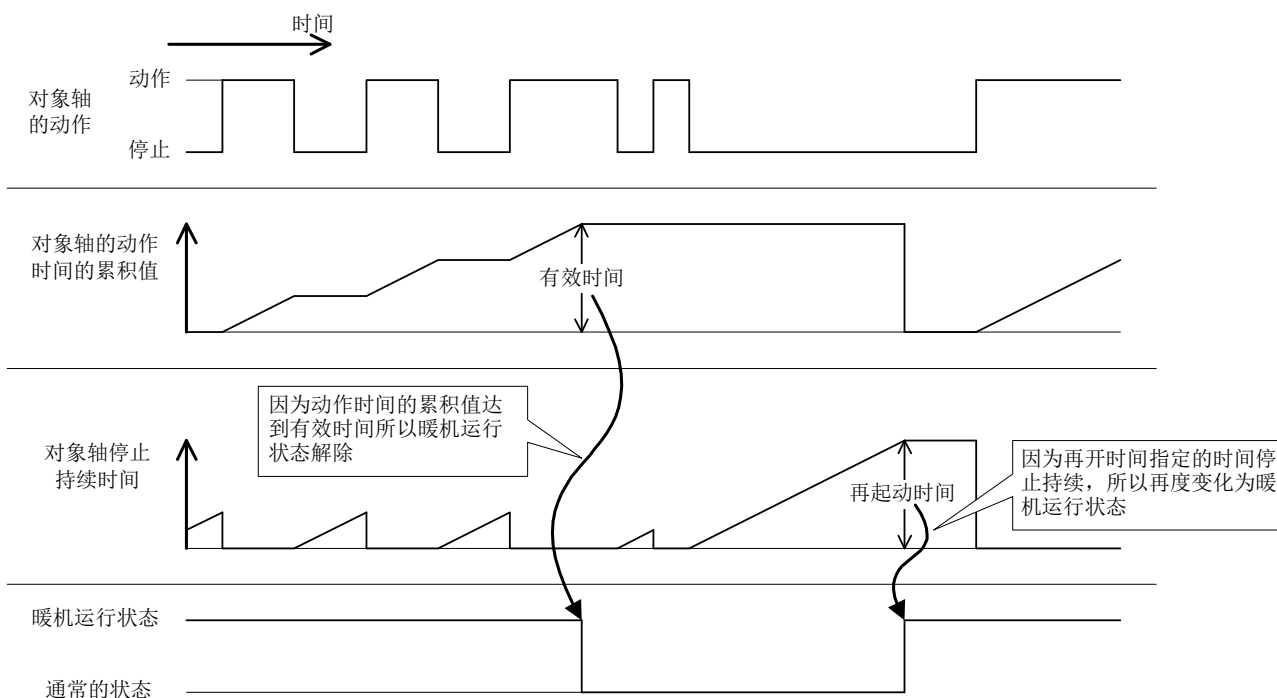


图 5-6：通常的状态 / 暖机运行状态的切换示例

### ■ 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。

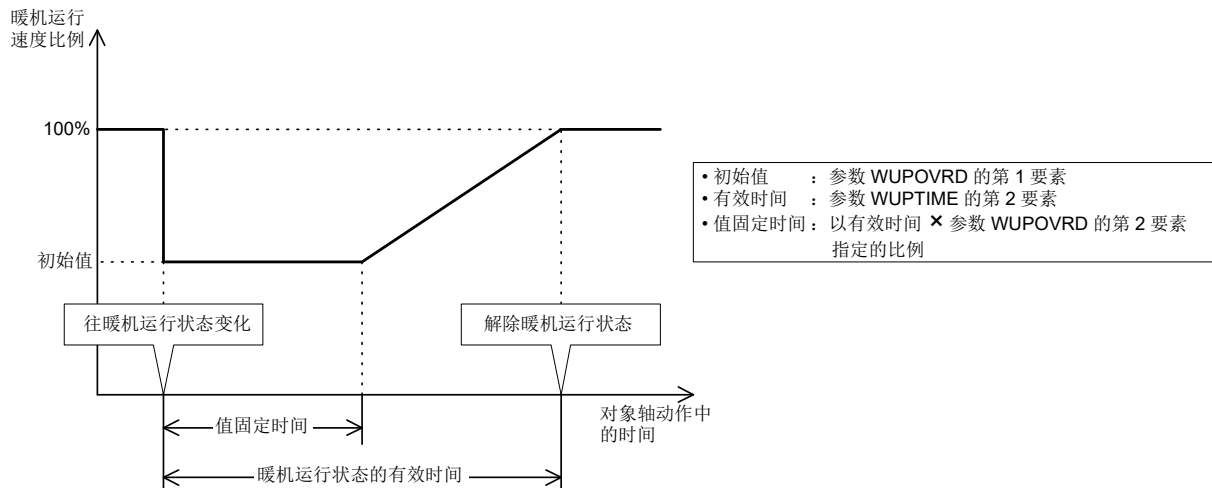


图 5-7：暖机运行倍率超调的变化

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 = (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 = (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。

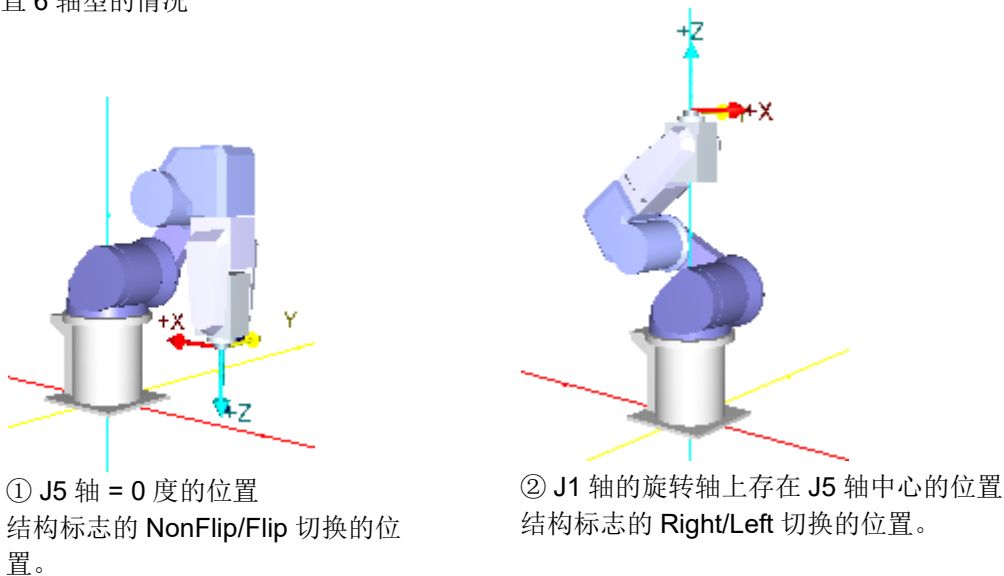
但是，即使使用结构标志，在该标志的切换位置上，有的关节轴的取得角度有无限个组合方式，因此无法让机器人在所希望的位置上以所希望的姿势进行动作（例如，垂直 6 轴型的机器人的情况下，J5 轴 = 0 度时，J4 轴与 J6 轴无法确定）。该位置称为特异点，原先不可以以直交 JOG 或直线插补等方式通过。因此，目前为止都是通过调整布局使特异点不在作业区域中，或无论如何都需要通过特异点时，通过关节插补进行动作等措施加以对应。

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。

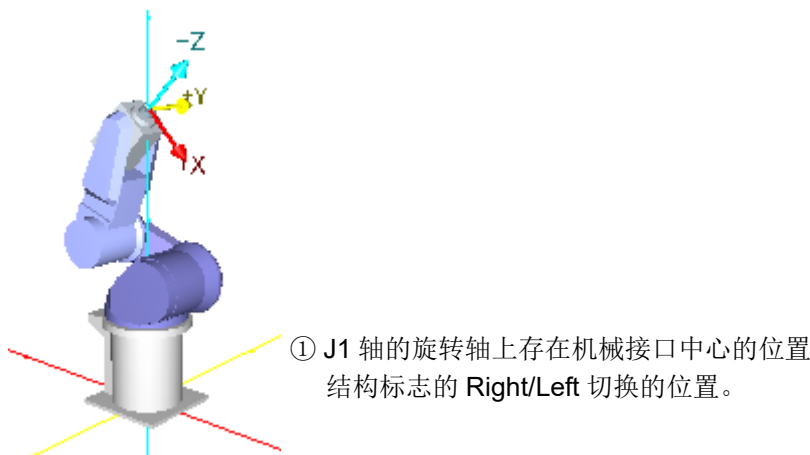
#### ■可通过的奇异点位置

可以以特异点通过功能通过的奇异点的位置如下所示。

垂直 6 轴型的情况



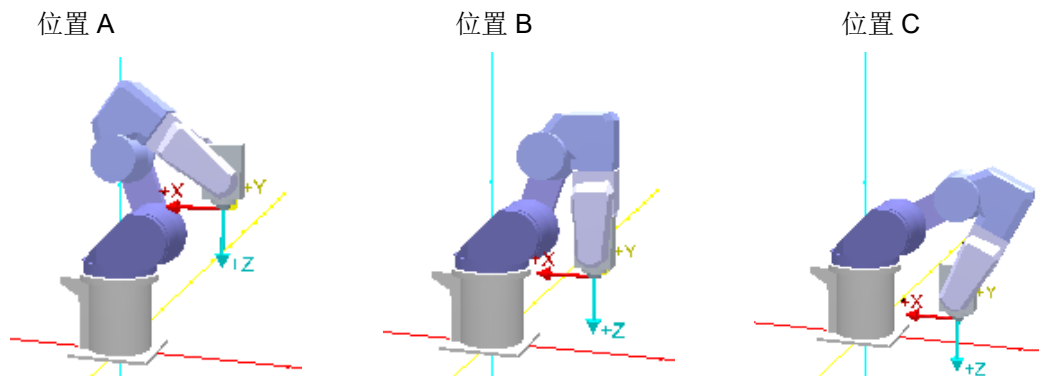
垂直 5 轴型的情况



### ■ 特异点通过功能有效时的动作

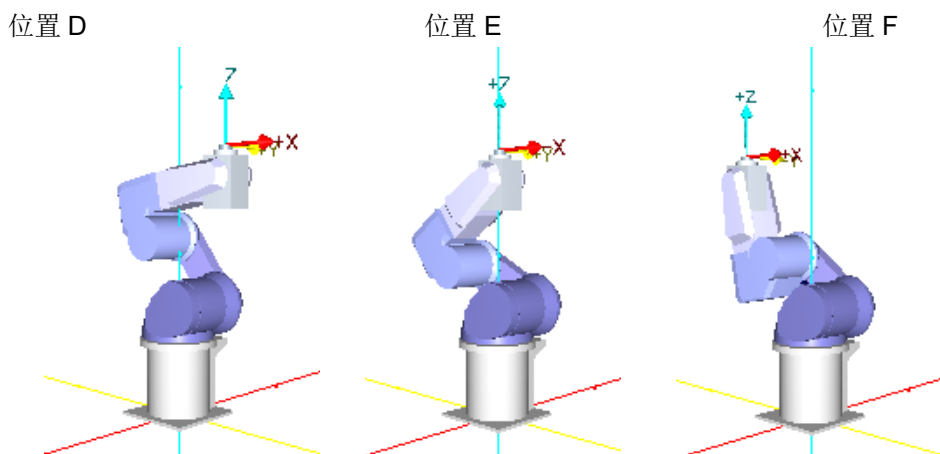
将特异点通过功能设为有效后，可以通过直交 JOG 或直线插补等方式从位置 A 经由位置 B（特异点位置）动作至位置 C（反过来也一样）。此时，在通过位置 B 前后，结构标志的值会切换。

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。

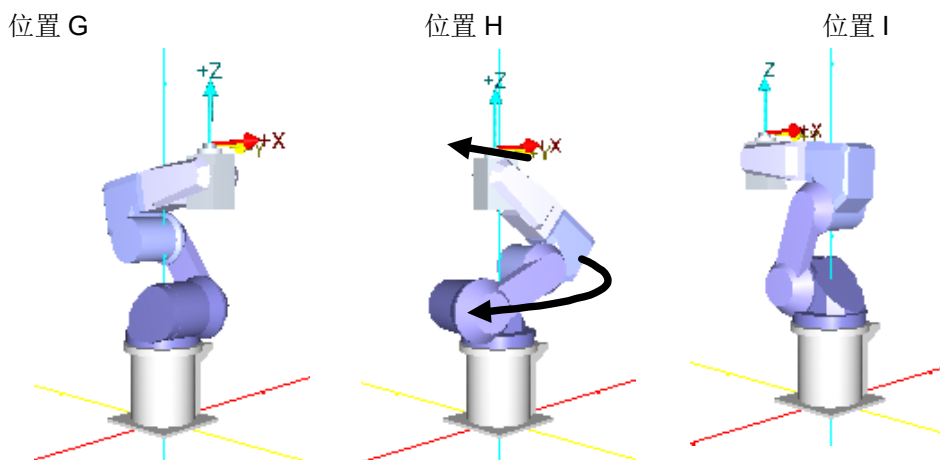


可以通过特异点的是机器人的动作轨迹通过特异点之上的情况。动作轨迹偏离特异点之上（通过特异点近旁）的情况下，动作时将不切换结构标志的值。

- 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-2FR、RV-4FR 系列、RV-7FR 系列、RV-7FRLL、RV-13FR 系列、RV-20FR、RV-8CRL |

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。                | 设定值不影响。            | 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 | (XYZ 将维持姿势不变，不切换结构标志。) | (将维持姿势不变，不切换结构标志。) | 设定值不影响。 |

- 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-FR 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。关于特异点接近警告，请参照第 525 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      | M0 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 535 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

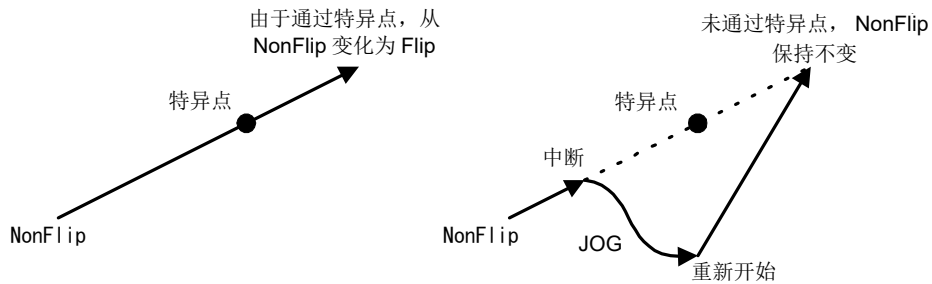
### 【例文】

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】

- 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。
- 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。
- 通过 Spd 指令指定了速度时，将以上限作为指定速度，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。
- 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。
- 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。
- 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，以往是以拉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。

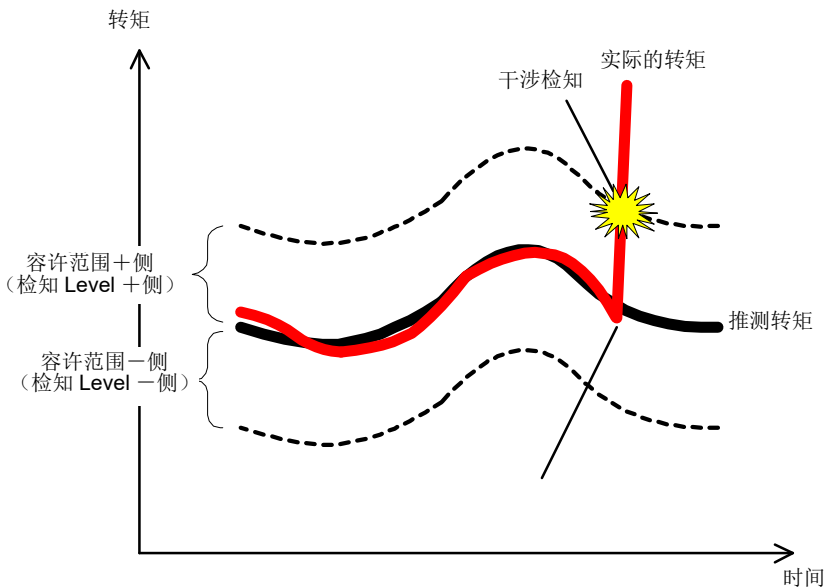


图 5-8：干涉的检测原理

## (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 468 页的“5.1 动作参数”、第 523 页的“5.16 关于抓手、工件条件设定（最佳加减速设定）”。

表 5-19：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-3FRH/6FRH/<br>12FRH/20FRH 系<br>列：1, 0, 1<br>RV-FR 系列：<br>0, 0, 1<br>RH-3FRHR 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>HNDDATO 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>WRKDATO 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,<br>0.0, 0.0, 0.0,<br>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)。

(请一并参照第 539 页的“表 5-19: 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 5-20 所示。

表 5-20: 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-3FRH/6FRH/<br>12FRH/20FRH 系<br>列: 1, 0, 1<br>RV-FR 系列:<br>0, 0, 1<br>RH-3FRHR 系列:<br>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。                                                                                       |
| WRKDATO  | 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,<br>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、WRKDATO。

## ◆◆◆ 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)。

可通过 RT ToolBox3 的示波器功能选择“碰撞检测等级参考值”。由此可以实现参数 COLLVLJG 设定值的调整作业简易化。详情内容请参照第 543 页的“3) 补充事项”

## ■ 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

程序运行时的碰撞检测功能，以参数设定初始状态，但是实际上是使用 MELFA-BASIC VI 的指令在程序中更改设定。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，关于指令的详细内容请参照第 173 页的“4.12 指令的详细说明”、第 333 页的“4.13 机器人（系统）状态变量的详细说明”。

表 5-21：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-3FRH/6FRH/<br>12FRH/20FRH 系<br>列：1, 0, 1<br>RV-FR 系列：<br>0, 0, 1<br>RH-3FRHR 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,<br>0, 0, 0, 0, 0,<br>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 5-22：程序运行时的碰撞检测功能中使用的 MELFA-BASIC VI 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。                                                                                                      |
| CollLvl   | 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 CollLvl ,,, 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：CollLvl 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 CollLvl 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 模式以回避动作使用。                                                                                                         |

◆◆◆ 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 5-23：程序运行时的检知 Level 调整步骤的示例

| 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。

可通过 RT ToolBox3 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 543 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 5-9 一样的状态迁移。

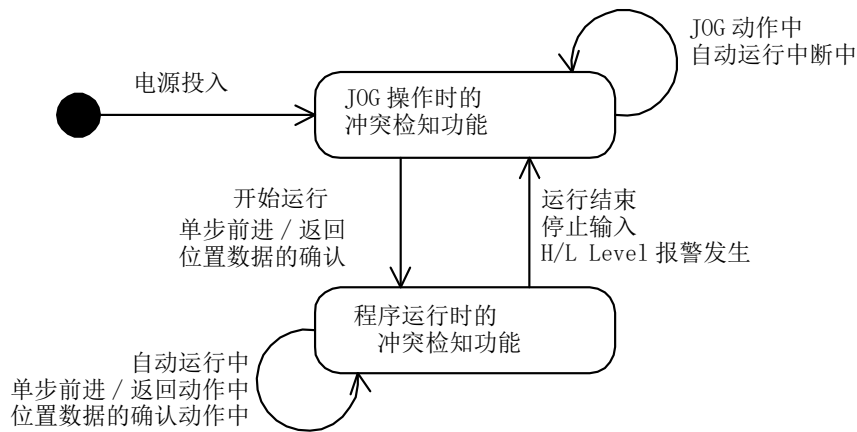


图 5-9：程序运行时 / JOG 操作时切换相关的状态迁移

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，只要按下停止按钮，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”

可通过 RT ToolBox3 的示波器功能的实时监控选择“碰撞检测等级参考值”。

“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。

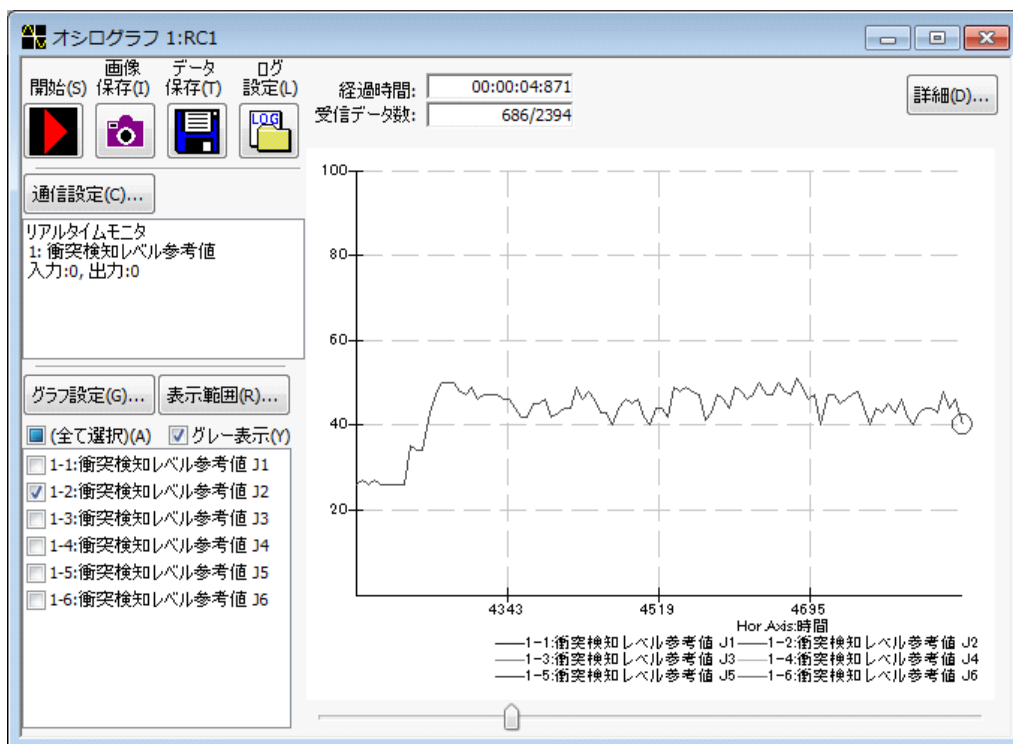


图 5-10: “碰撞检测等级参考值” 的示波器显示

碰撞检测功能有效且伺服 ON 时，将如图 5-10 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 5-24: 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 ColLv1 指令与 ColChk 指令。                                                                |
| 2  | 为了不错误检测出干涉，要降低（ColLv1 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 ColLv1 指令的自变量。                                                    |
| 5  | 将通过步骤 4 求出的值设定在 ColLv1 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 ColLv1 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-FRH 系列 //RH-CRH 系列 /RV-FR 系列

表 5-25 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 5-25：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-3FRH/6FRH/<br>12FRH/20FRH：40<br>RV-2FR 系列：40<br>RV-4FR/7FR/<br>13FR/20FR、<br>RH-3FRHR：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，配置部件可能提前出现故障，或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-2FR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 5-26: 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 (进行多转位置的错误检查) 时, 会像 RV-2FR 机器人那样, 在 J1 轴或 J4 轴的关节角度超出 <math>\pm 180</math> 度进行多转的机型中, 指定了跨多转位置的情况下, 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-FR 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

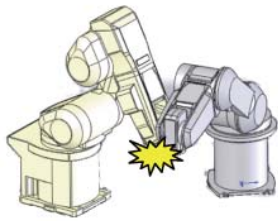
本功能是在进行干涉确认的同时使机器人动作的功能。干涉确认对象为①机器人之间（仅限 CR800-R/Q 系列）和②机器人与自由平面极限之间。

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-FRH/RV-FR 系列的 CR800-R/Q 系列。（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-FRH/CRH 系列、RV-FR 系列、RV-8CRL 及 RV-5AS。（不支持用户机械）

※ Q 系列中，CR700-Q 控制器与 CR800-Q 控制器在同一基本坐标系中同时存在时，即使将干涉回避功能设为有效也不动作。

**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 5-11：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。

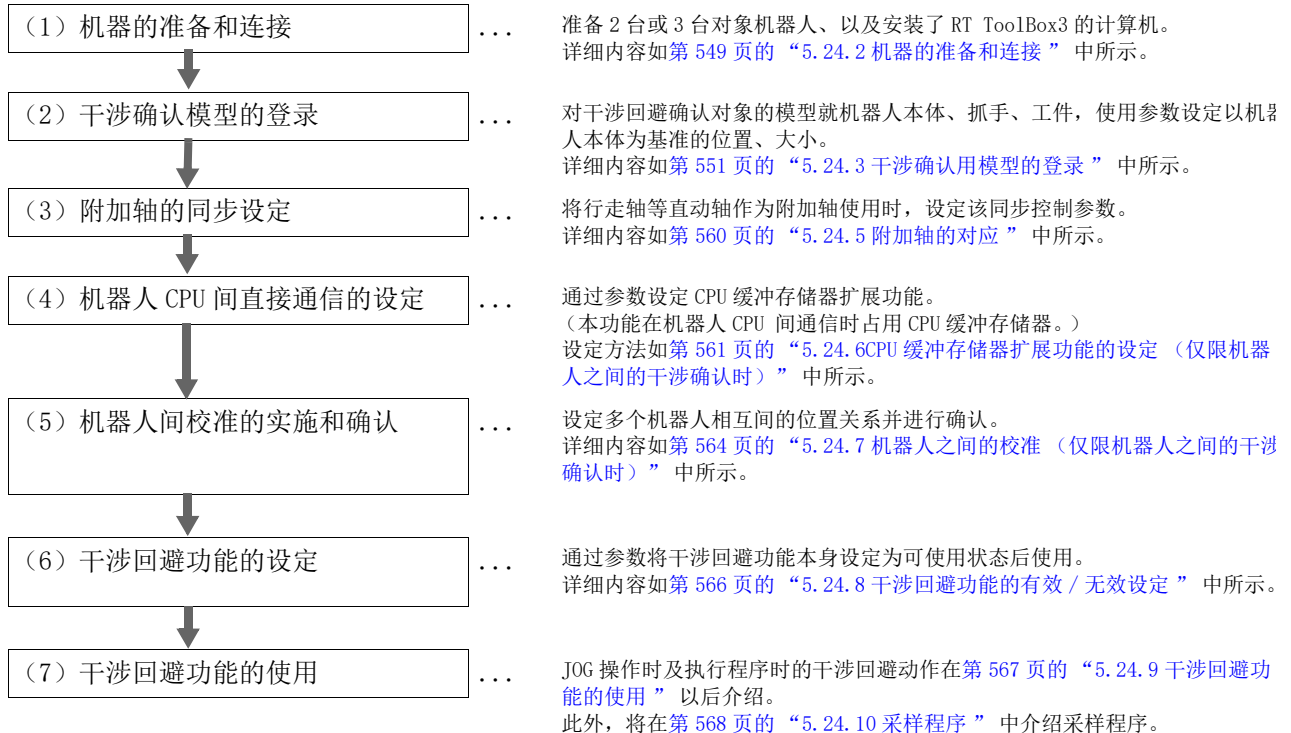


## 5.24.1 操作步骤

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 CR800-R/Q 系列）

〈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 5-27 中将使用本功能所需的机器，图 5-12 连接到的示例。  
 请参照图示，连接所需机器。

(1) 机器人之间的干涉确认（仅限 CR800-R/Q 系列）

■ CR800-R 控制器

表 5-27：所需机器（CR800-R 控制器）

| No. | 机器                               | 备注                                              |
|-----|----------------------------------|-------------------------------------------------|
| 1   | 机器人（CR800-R 控制器）<br>2 台或 3 台（最多） | 本功能使用经由 iQPlatform 的 CPU 缓冲存储器的机器人 CPU 间进行直接通信。 |
| 2   | 已安装 RT ToolBox3 的计算机             | 与机器人 CPU 连接。                                    |

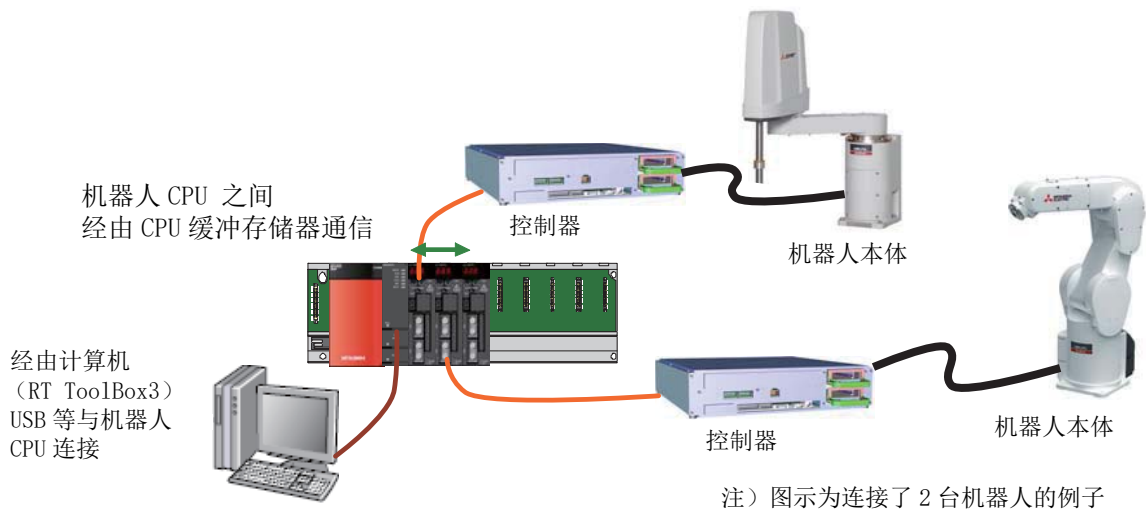


图 5-12：机器的连接(CR800-R 控制器)

■ CR800-Q 控制器

表 5-28：所需机器（CR800-Q 控制器）

| No. | 机器                               | 备注                                      |
|-----|----------------------------------|-----------------------------------------|
| 1   | 机器人（CR800-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。 |
| 2   | 安装了 RT ToolBox3 的计算机             | 与机器人 CPU 连接。                            |

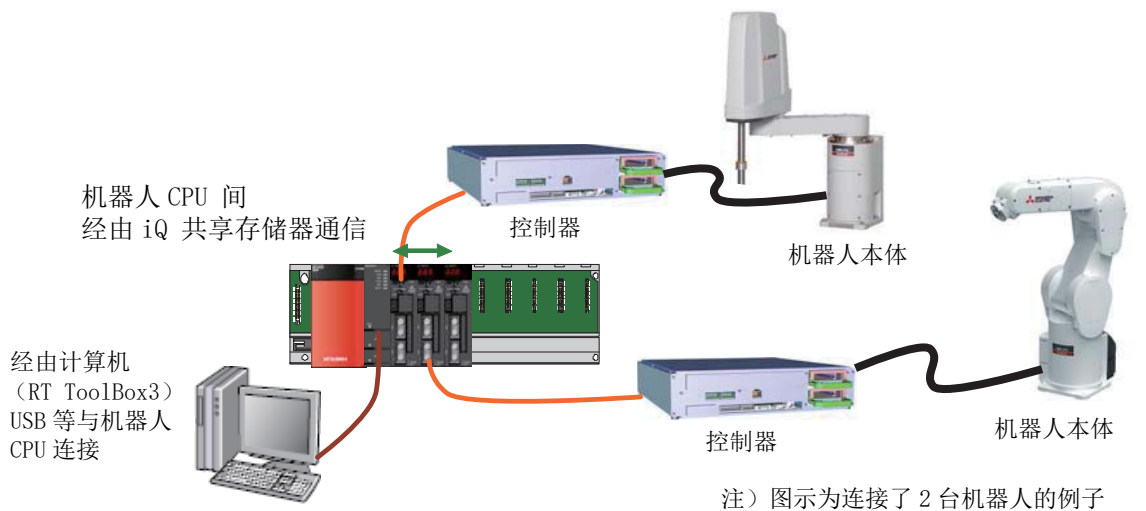


图 5-13：机器的连接(CR800-Q 控制器)

## (2) 自由平面极限之间的干涉确认

表 5-29: 所需的机器

| 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。 |
| 2   | 已安装 RT ToolBox3 的计算机 | 与机器人 CPU 连接。          |

## &lt;CR800-D 控制器 &gt;



图 5-14: 机器的连接 (CR800-D 控制器)

## &lt;CR800-R 控制器 &gt;

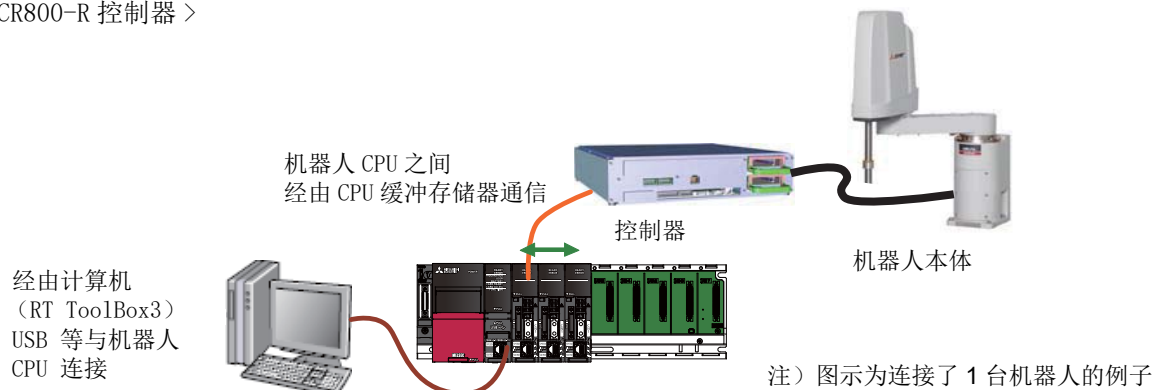


图 5-15: 机器的连接 (CR800-R 控制器)

## &lt;CR800-Q 控制器 &gt;

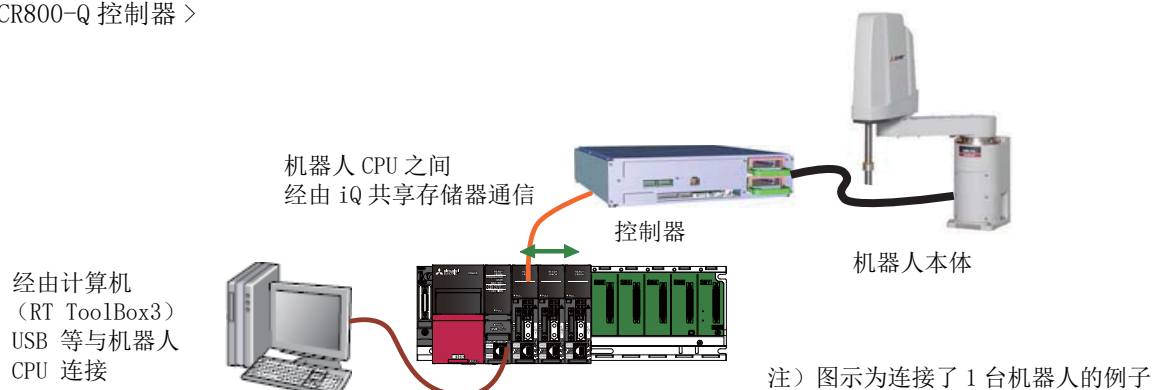
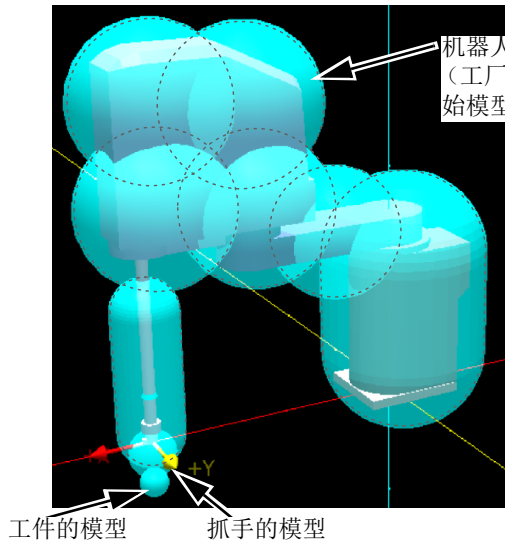


图 5-16: 设备的连接 (CR800-Q 控制器)

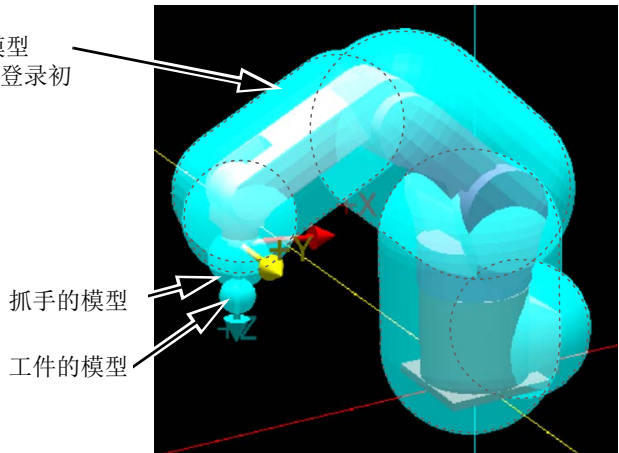
5.24.3 干涉确认用模型的登录

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-FRH 系列>



<RV-FR 系列>



※ 图示为 RT ToolBox3 的画面示意（设定示例）。球及圆筒的形状表示模型。由客户设定抓手及工件的模型。

图 5-17：模型的登录示例

需要的登录内容归纳在表 5-30 中。机器人本体、抓手、工件各可登录 8 个模型。

表 5-30：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 <sup>注1)</sup>                                                                                 | 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）<br><br>注) 模型的形状为球形及圆筒形。<br>参数: <b>CAVKDA1 ~ 8</b> | 设定的抓手号码与 <b>LoadSet (Load Set)</b> 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: <b>CAVKDH1 ~ 8</b> | 设定的工件号码与 <b>LoadSet (Load Set)</b> 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: <b>CAVKDW1 ~ 8</b> |
|                                                                          |                                                                                                         | 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 5-30 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。

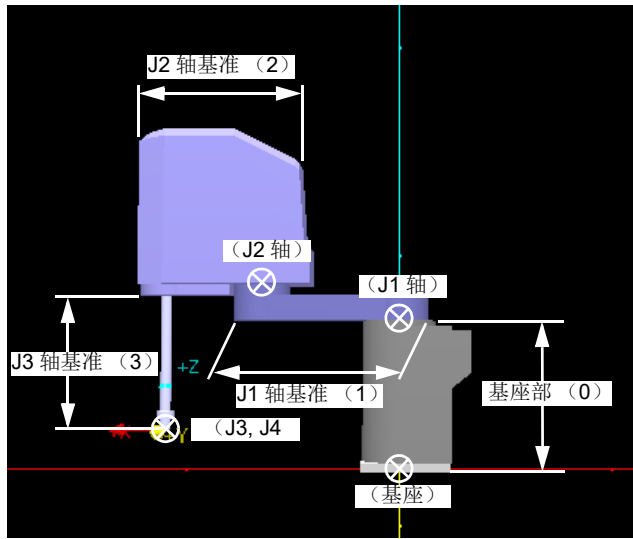
##### 1) 机器人本体的模型

①模型登录部和形状：CAVKDA1 ~ 8

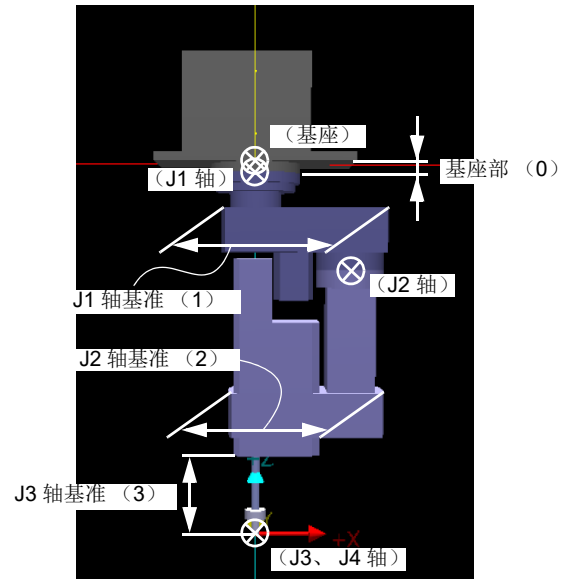
表 5-31：模型设定参数（机器人本体：CAVKDA1 ~ 8）

| 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-3/6/12/20FRH 系列、RH-3/6CRH 系列、<br>RV-FR 系列、RV-8CRL：<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-2FR 为<br>CAVKDA2=0, 0<br>CAVKDA4=3, 0<br>CAVKDA5=4, 1<br>CAVKDA6=5, 0<br>RV-8CRL 为<br>CAVKDA4=3, 0<br>CAVKDA5=4, 1<br>CAVKDA6=5, 1<br>RH-3FRHR 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 |

<RH-3FRH/6FRH/12FRH/20FRH 系列>



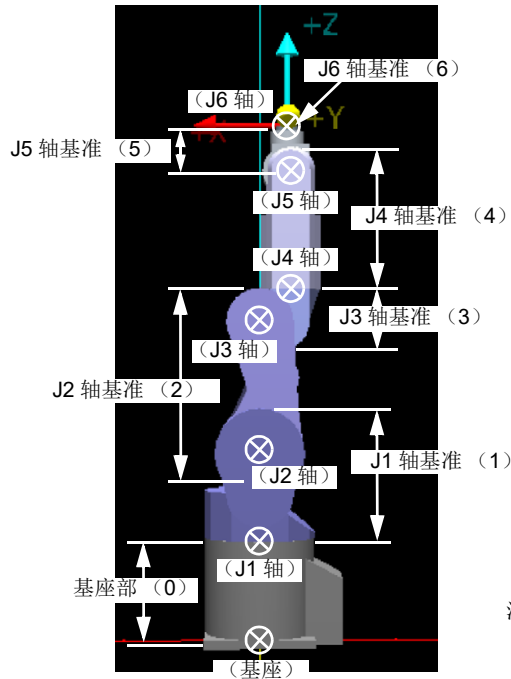
<RH-3FRHR 系列>



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 5-18: RH-FR 系列模型登录部 (补充)

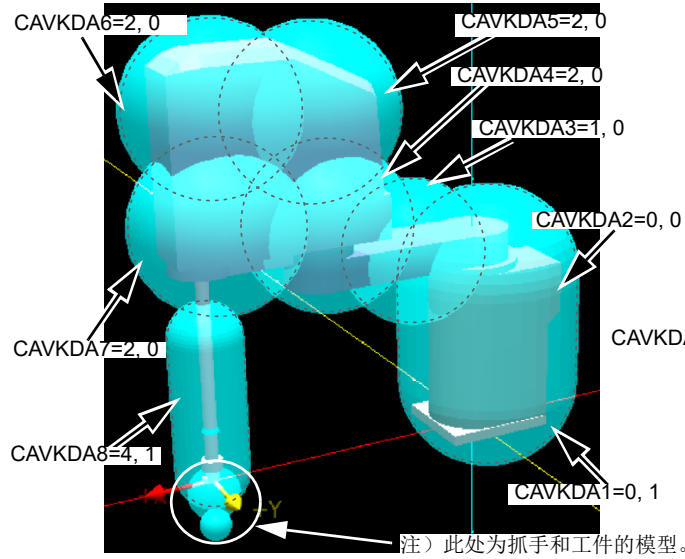
<RV-FR 系列>



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: RV-FR 系列模型登录部 (补充)

<RH-FRH 系列>



<RV-FR 系列>

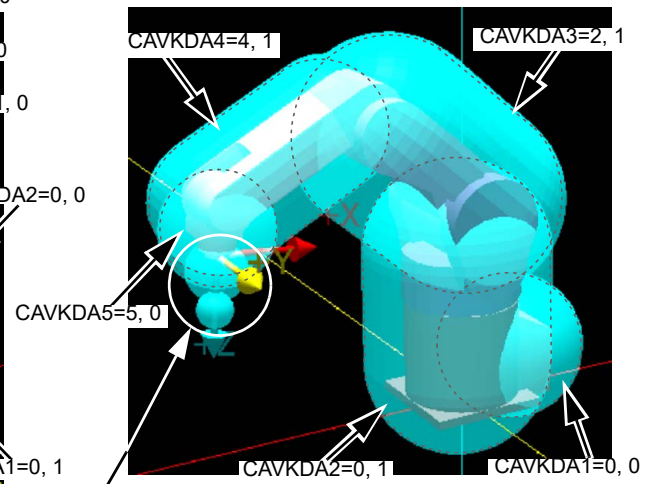
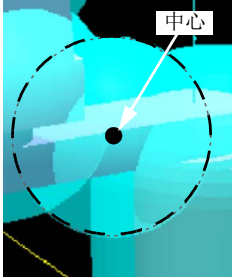
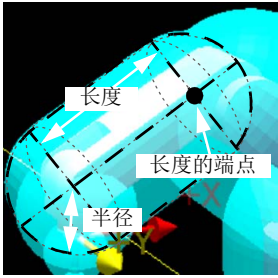


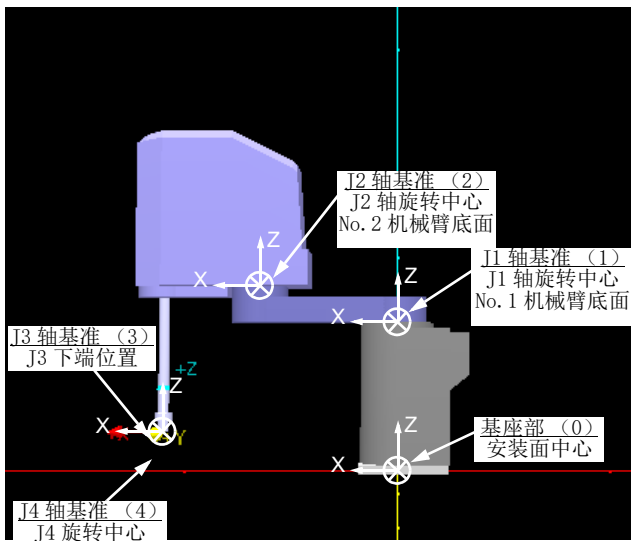
图 5-20: 登录部和模型形状的设置示例

②模型的位置：CAVPSA1 ~ 8

表 5-32：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> |        |

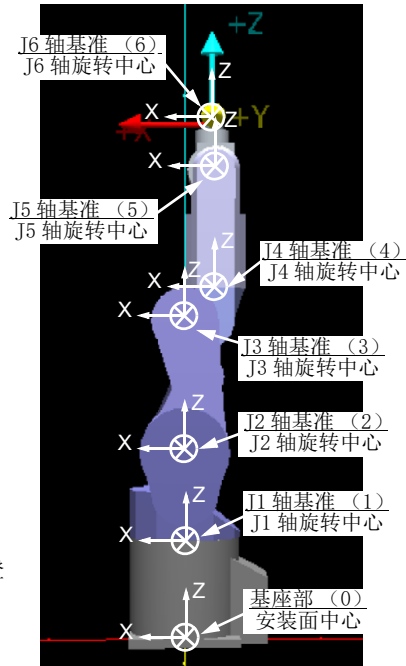
<RH-FRH 系列>



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 5-21：各模型的基准轴位置 XYZ 方向

<RV-FR 系列>



[ 补充 ]：各基准轴的 XYZ 方向

RH-FRH 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-FR 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。

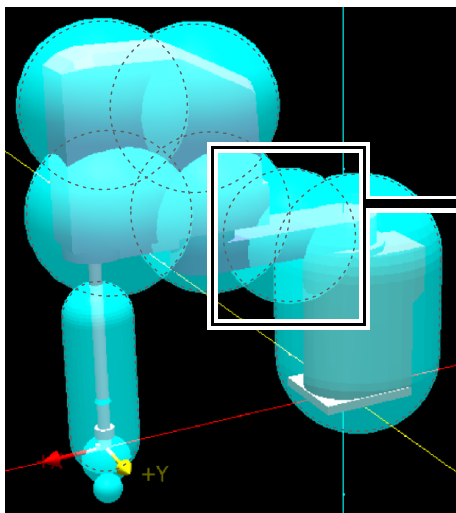


## ③模型的大小：CAVSZA1 ~ 8

表 5-33：模型设定参数（机器人本体：CAVSZA1 ~ 8）

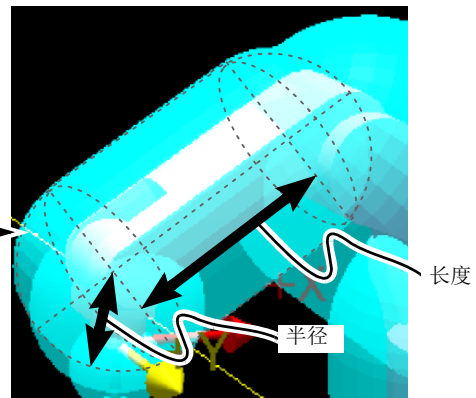
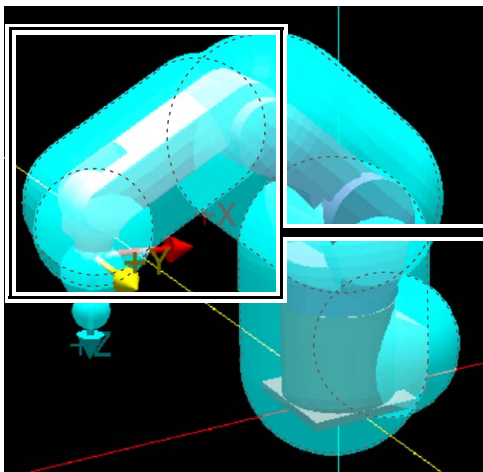
| 参数               | 参数名         | 排列数<br>文字数 | 内容说明                                                                                                                | 出货时设定值 |
|------------------|-------------|------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 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。 |        |

## &lt;RH-FRH 系列&gt;



以半径设定球形模型的大小。

## &lt;RV-FR 系列&gt;



以半径和长度设定圆筒模型的大小。

图 5-22：模型的大小（补充）

④模型的有效 / 无效：CAVSCA1 ~ 8

表 5-34：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<b>临时解除干涉回避功能的操作</b>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-3/6/12/20FRH系列、RH-3/6CRH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>注）RH-3FRH35xx、RH-6FRH35xx、RH-12FRH55xx为CAVSCA3=0, 0, 0。</p> <p>RH-3FRHR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> <p>RV-FR系列、RV-8CRL：<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>注）RV-2FR、RV-8CRL为CAVSCA6=1, 0, 0。</p> |

2) 抓手的模型

表 5-35 中为作为抓手登录的模型的设定参数。

表 5-35：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<b>LoadSet (Load Set)</b>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<b>LoadSet (Load Set)</b>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<b>机械接口坐标系</b>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 (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 <b>临时解除干涉回避功能的操作</b> 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>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 5-36 中为作为工件登录的模型的设定参数。

表 5-36: 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 (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 <b>LoadSet (Load Set)</b> 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 <b>LoadSet (Load Set)</b> 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 (工件)      | CAVPSW1 ~ 8 | 实数 6       | 对各模型指定从 <b>机械接口坐标系</b> 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 (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 (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 <b>临时解除干涉回避功能的操作</b> 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>第3要素: 固定为0                    | 将所有参数 (CAVSCW1~8) 都设定为 “0, 0, 0”。          |

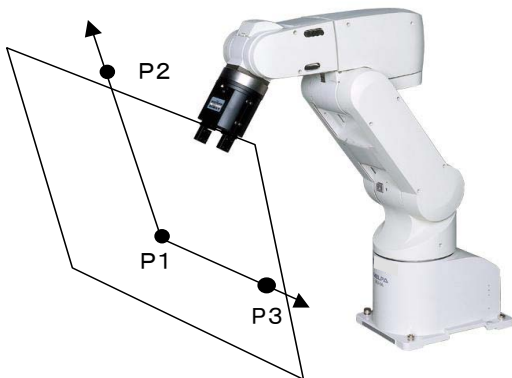
◇◆◇变更执行程序时的确认对象模型（抓手、工件）◇◆◇  
 执行程序时，可以使用 **LoadSet (Load Set)** 指令，变更作为干涉确认对象的抓手和工件的模型，以便根据实际使用的抓手或把持的工件执行干涉确认。  
 使用 **LoadSet (Load Set)** 指令时，指定事先在定义模型的参数中设定的抓手号码、工件号码。

◇◆◇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。需要的登录内容归纳在表 5-37 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 轴机器人的示例。水平多关节机器人也一样。

表 5-37: 自由平面极限登录内容

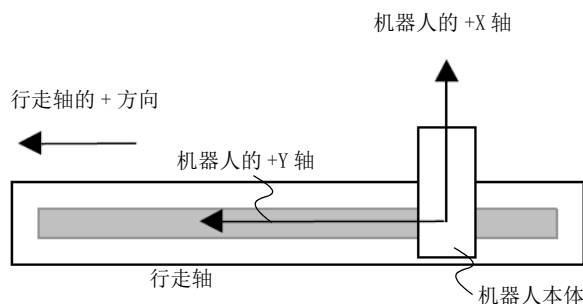
| 参数和值              | 说明                                                                                                                               |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------|
| 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）对于后面的第 564 页的“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 5-23：行走轴的使用示例

表 5-38：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 | CAVAXJNO | 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 CPU缓冲存储器扩展功能的设定（仅限机器人之间的干涉确认时）

通过参数设定缓冲存储器扩展功能。

通过该设定，占用机器人CPU间通信所需的CPU缓冲存储器。（参照图5-24）

(1) 参数的设定

参数：将 IQMEM 的位 4 设为“1”，将机器人间干涉回避功能设定为有效。

或参数：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 5-39：CPU 缓冲存储器扩展功能选择参数

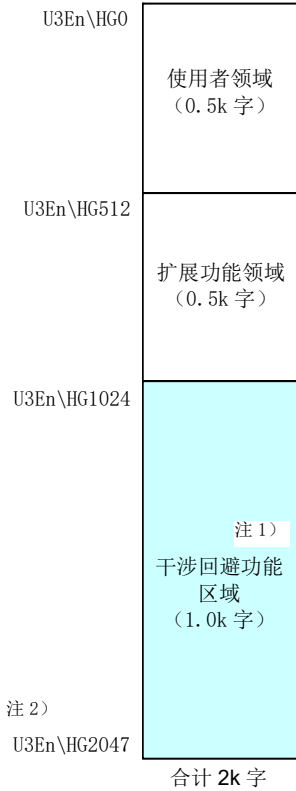
| 参数              | 参数名      | 排列数<br>文字数 | 内容说明                                                                                                                                                                              | 出货时设定值               |
|-----------------|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| CPU 缓冲存储器扩展功能选择 | IQMEM    | 整数 1       | 选择 CPU 缓冲存储器扩展功能。<br>对每个位分配功能。1/0= 有效 / 无效<br>15<br>0<br>00000000 00000000<br>   ...bit0: 扩展功能的使用<br>   .....bit1:PLC 直接执行功能<br>   <br>bit2-3、<br>5-15<br>未使用   .....bit4: 干涉回避功能 | 00000000<br>00000000 |
| 设定多 CPU 台数      | QMLTCPUN | 整数 1       | 设定在多CPU系统中，安装在主基板上的CPU单元的台数。                                                                                                                                                      | 2                    |

| 参数                                                     | 参数名                 | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 出货时设定值     |      |   |           |   |           |   |           |            |
|--------------------------------------------------------|---------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------|---|-----------|---|-----------|---|-----------|------------|
| CPU 缓冲存储器固定周期通信区域设定<br>(※ 仅限 CR800-R 系列)               | QMLTCPUn<br>n=1 ~ 4 | 整数 4       | <p>多CPU系统中，从1号机读取通过1~4号机的CPU缓冲存储器固定周期通信区域进行发送接收的点数，并自动设定。无需更改值。</p> <p>要素1: 固定周期通信区域的大小 (K字)<br/>范围: 0~12<br/>※全号机合计最多24K字</p> <p>要素2: 自动刷新的点数 (字)<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: 系统备用</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>                                                                                                                                                                                                                                                                                                                            | 1, 0, 1, 1 |      |   |           |   |           |   |           |            |
| 多 CPU <sub>n</sub> 号机<br>设定高速通信领域<br>(※ 仅限 CR800-Q 系列) |                     |            | <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。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" data-bbox="667 981 1031 1160"> <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<b>0。</b></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)，<b>请不要改变设定。</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           |            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |      |   |           |   |           |   |           |            |

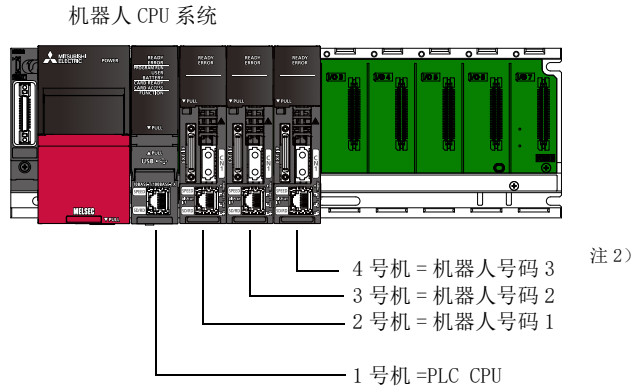
(2) CPU 缓冲存储器映射

表示基于参数：IQMEM 的设定状态的 CPU 缓冲存储器上的机器人 CPU 输出区域的存储器映射。  
 下图变为 CR800-R 系列时的存储器映射。CR800-Q 系列时，请参考表 5-3 改读为对应的设备编号。

<CPU 缓冲存储器映射>



<插槽和机器人号码的关系>



注 1) 干涉回避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) CPU 缓冲存储器地址中显示的 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。

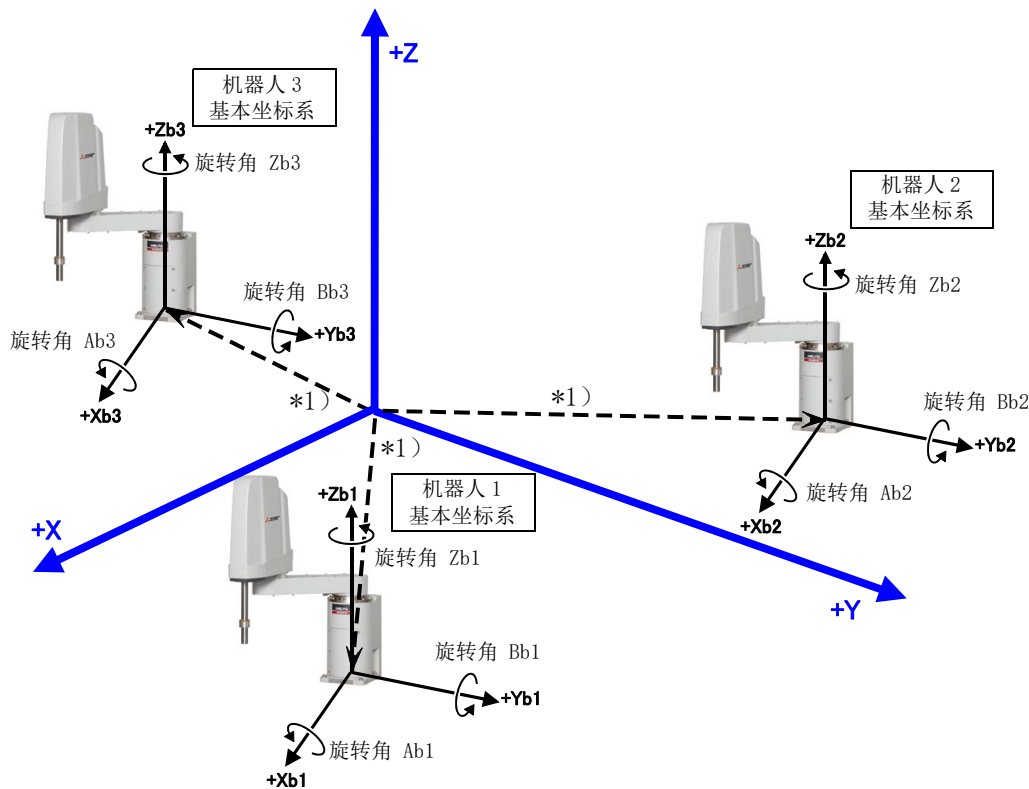
图 5-24: CPU 缓冲存储器映射



## 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 5-25：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 5-40：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,<br>0.00, 0.00,<br>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 5-26 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。

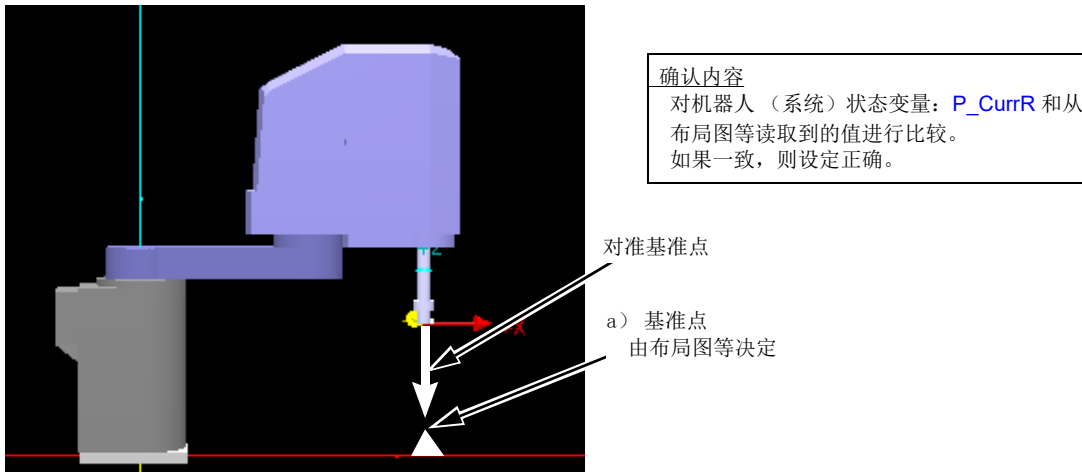


图 5-26: 确认机器人间的校准设定

## 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。

表 5-41 中设定的参数的详细内容如下所示。

表 5-41：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

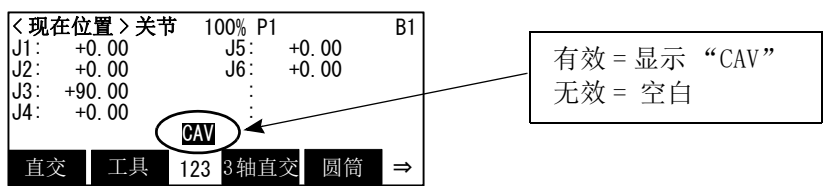
下面介绍 JOG 操作时及执行程序时的干涉回避动作。将参数 CAV 的第 1 要素设定为“1”（可以使用）。（参照第 566 页的“5.24.8 干涉回避功能的有效 / 无效设定”）

(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时等），通过执行本手册第 68 页的“3.11 无法解除错误的暂时错误复位操作”（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

通过将 CAV 的第 2 要素设定为有效（“1”），并使用表 5-42 中所示的指令语句及表 5-43 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 5-42：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183页 |

表 5-43：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409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412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408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348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 568 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, 执行退避动作 (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)
' 生成退避动作的移动量 (从干涉位置退回 50mm)
PDST=P_Fbc (1) +PESC ' 生成退避位置
Mvs PDST ' 退避动作
Mvs PHome ' 向退避位置移动
:

```

## 5.25 可编程控制器输入输出模块直接控制

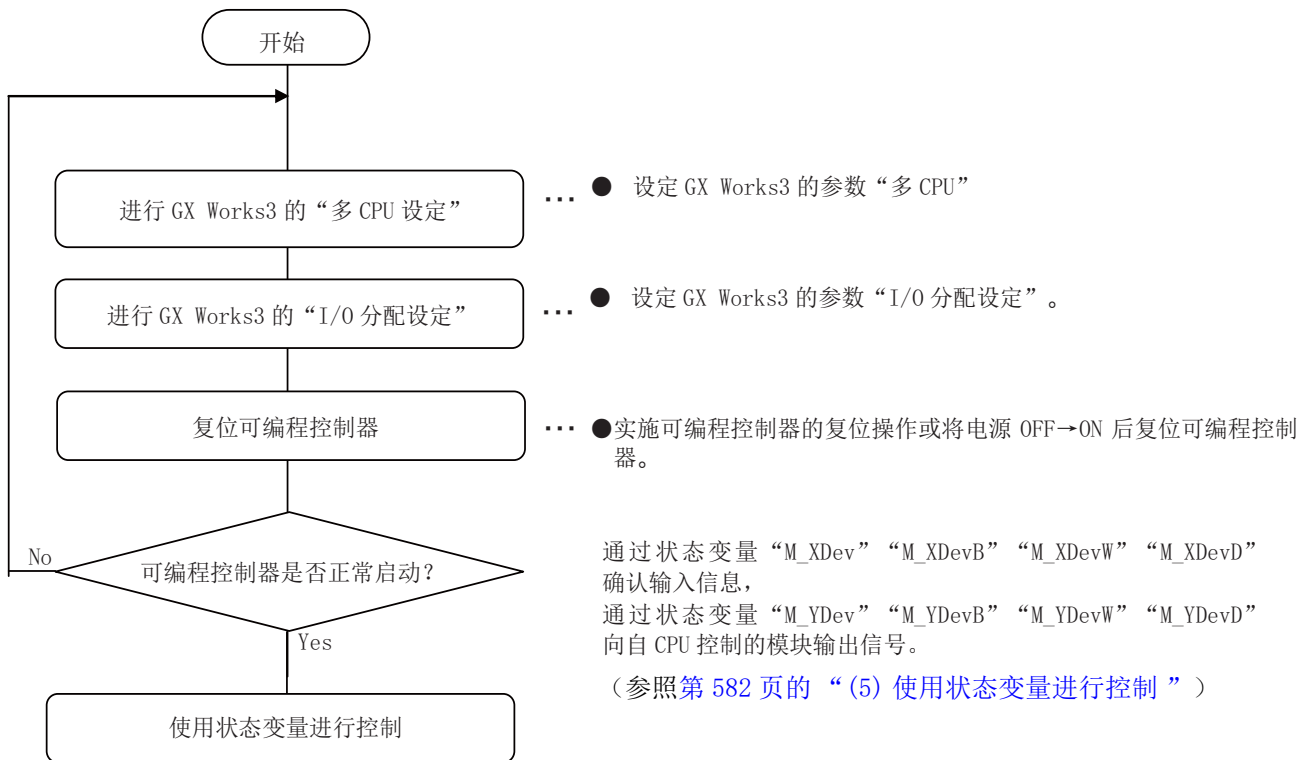
对机器人 CPU 控制可编程控制器用的“输入模块”、“输出模块”、“输入输出混合模块”3 个种类模块的功能进行说明。

CR800-R 系列和 CR800-Q 系列的设定步骤不同。CR800-R 系列时请参照第 570 页的“5.25.1CR800-R 系列”、CR800-Q 系列时请参照第 584 页的“5.25.2CR800-Q 系列”

### 5.25.1 CR800-R 系列

#### (1) 作业流程

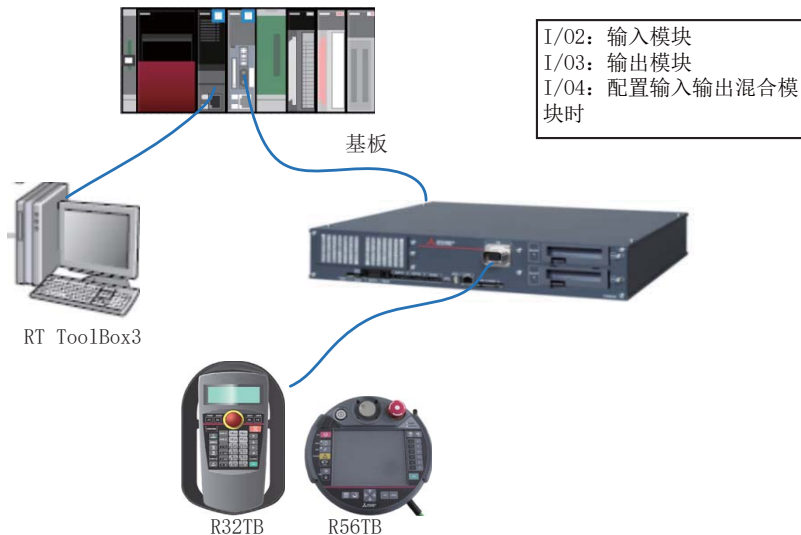
为了控制可编程控制器用输入输出模块，所需的作业流程如下所示。



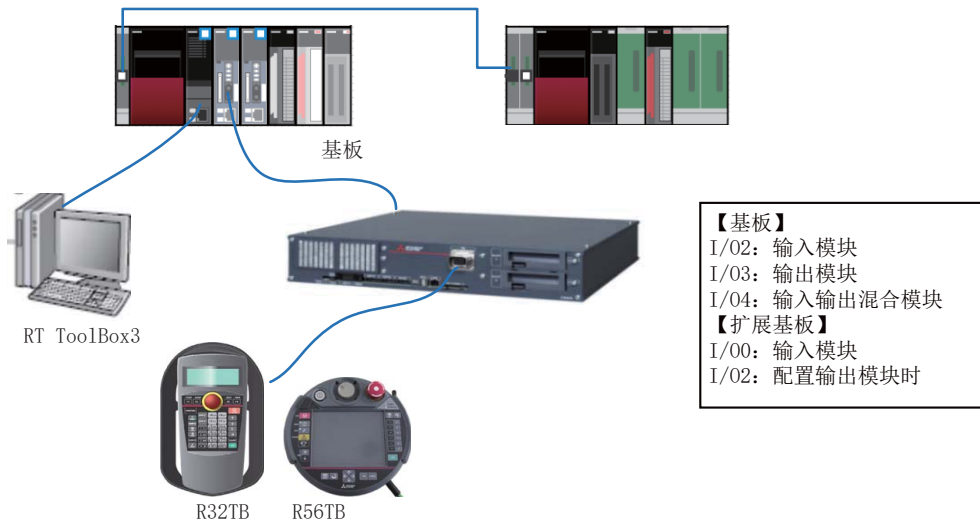
#### (2) 硬件的构成示例

将“机器人 CPU1 台+3 种输入输出模块”的构成作为系统构成示例 1、将“机器人 CPU2 台+3 种输入输出模块+扩展基板追加”的构成作为系统构成示例 2 进行以下说明。

#### ■ 系统构成示例 1：机器人 CPU1 台+3 种输入输出模块

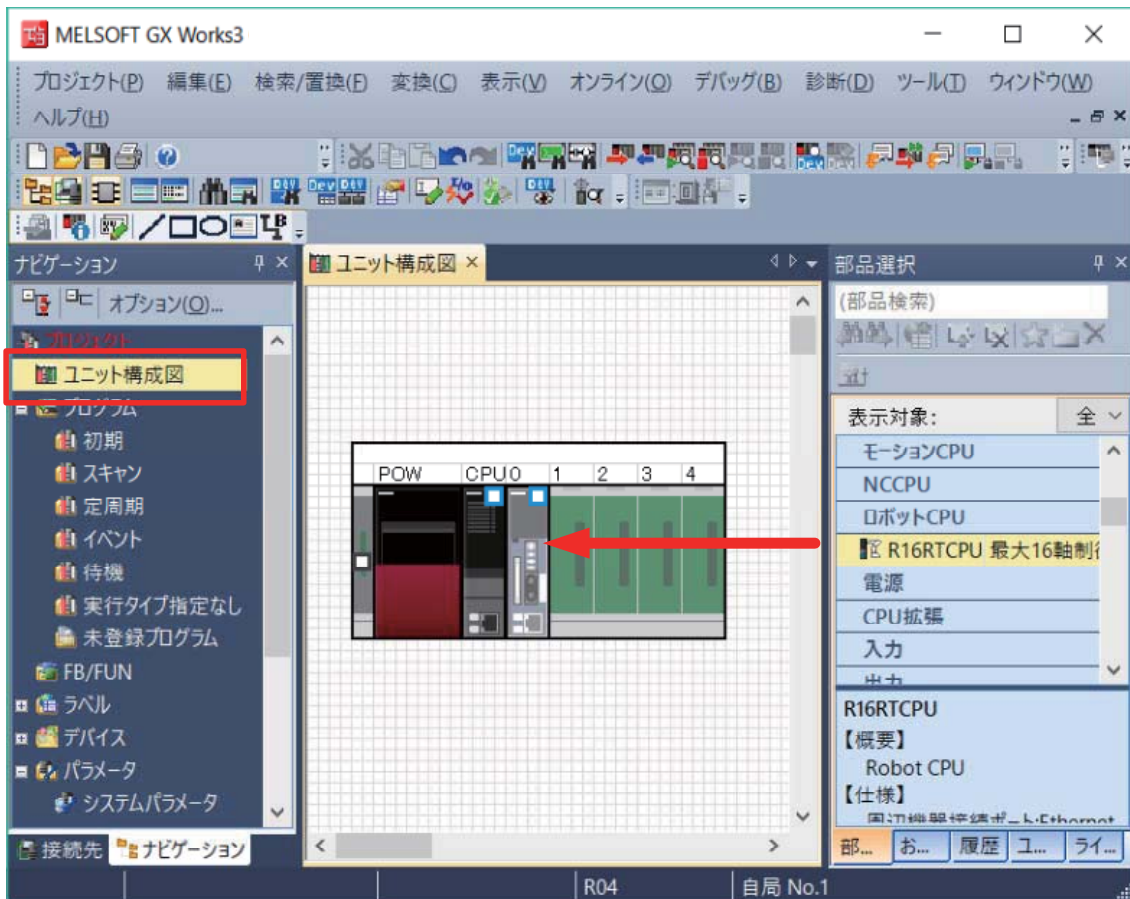


■系统构成示例 2：机器人 CPU1 台 + 3 种输入输出模块 + 扩展基板的追加



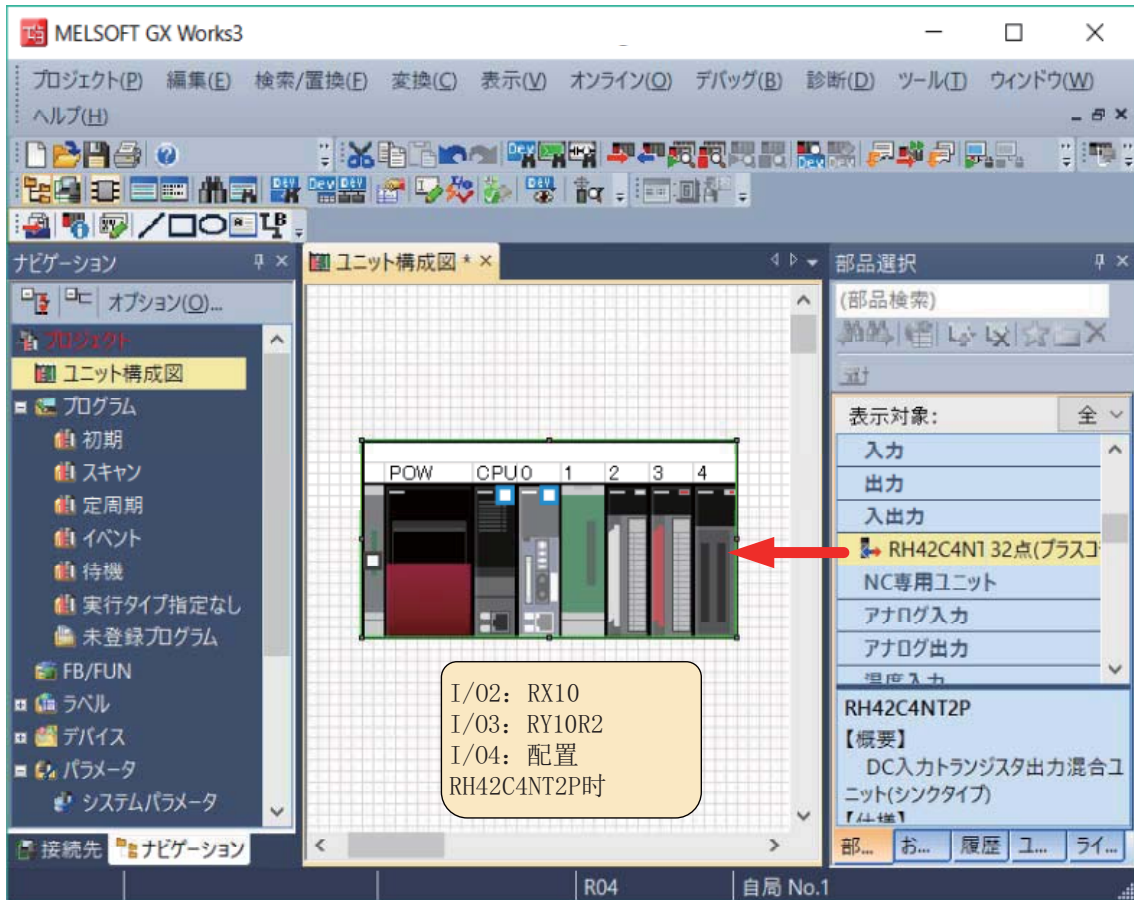
(3) 设定系统构成示例 1 的参数

根据系统构成示例 1（机器人 CPU1 台 + 3 种输入输出模块）的条件，配置 GX Works3 的“模块构成图”中使用的基座、电源模块、可编程控制器 CPU、机器人 CPU。

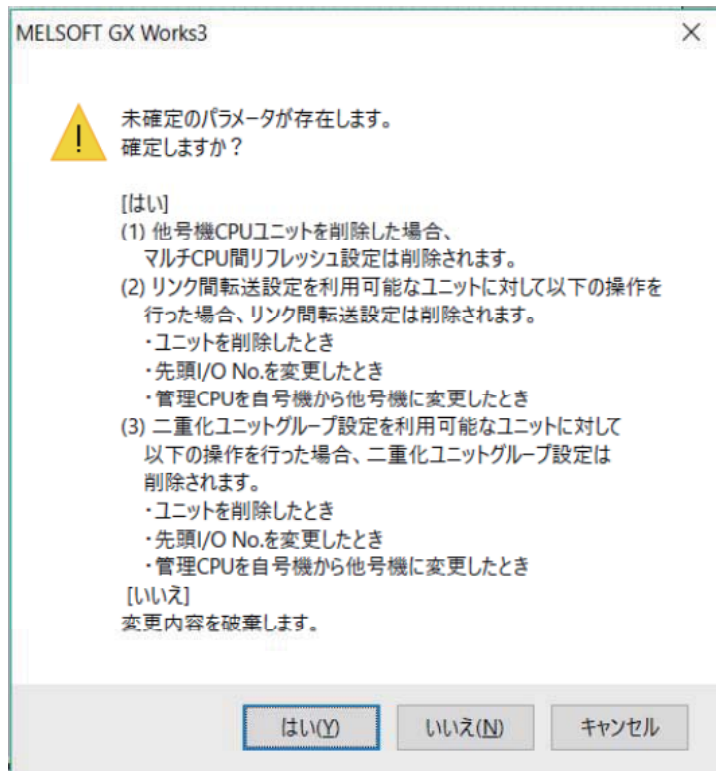




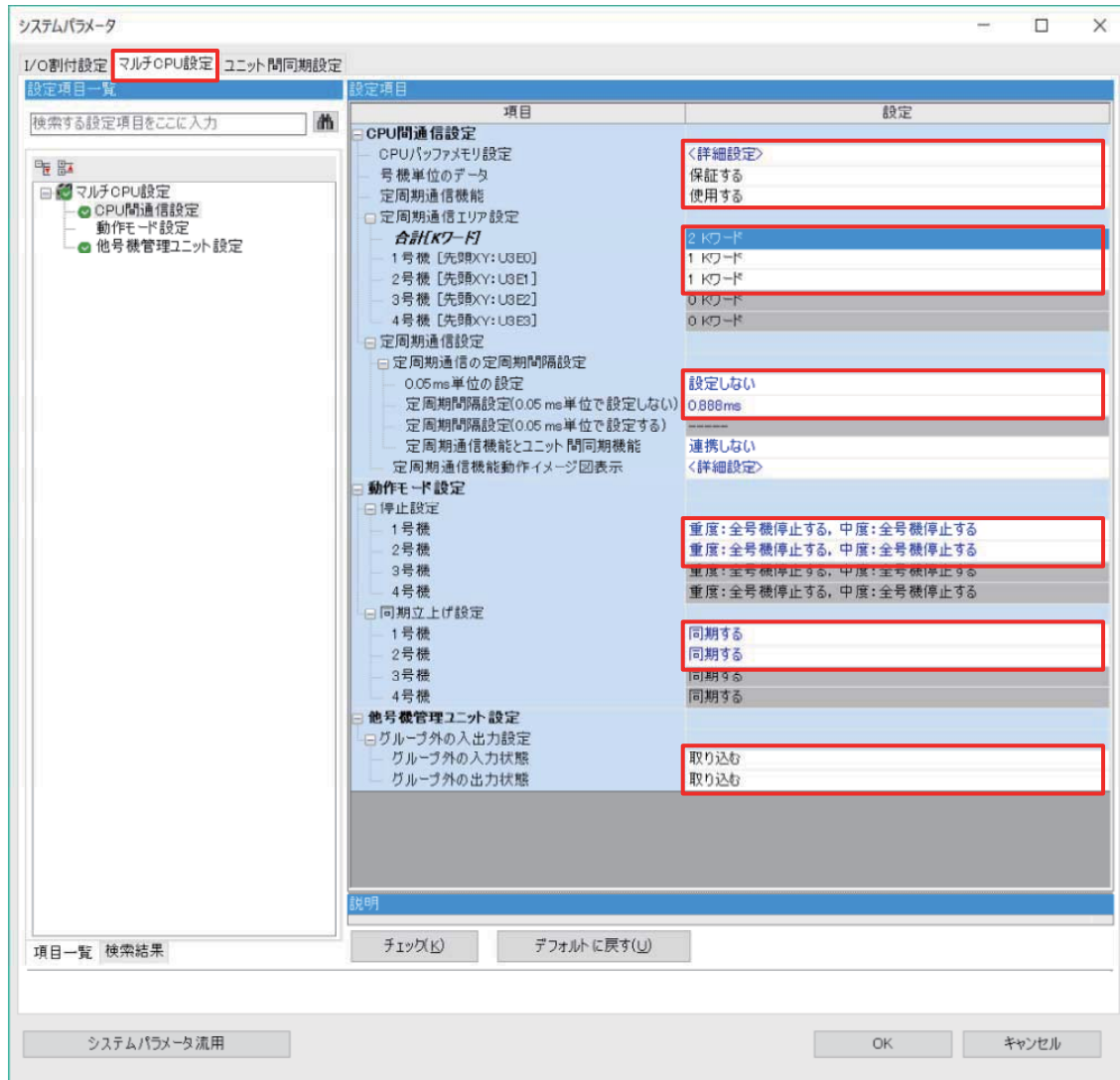
接着，配置使用的输入模块、输出模块、输入输出混合模块。



对模块进行配置后，关闭模块构成图画面，即显示下图的对话框，选择“是”。



接着，打开系统参数画面，实施多 CPU 的设定。



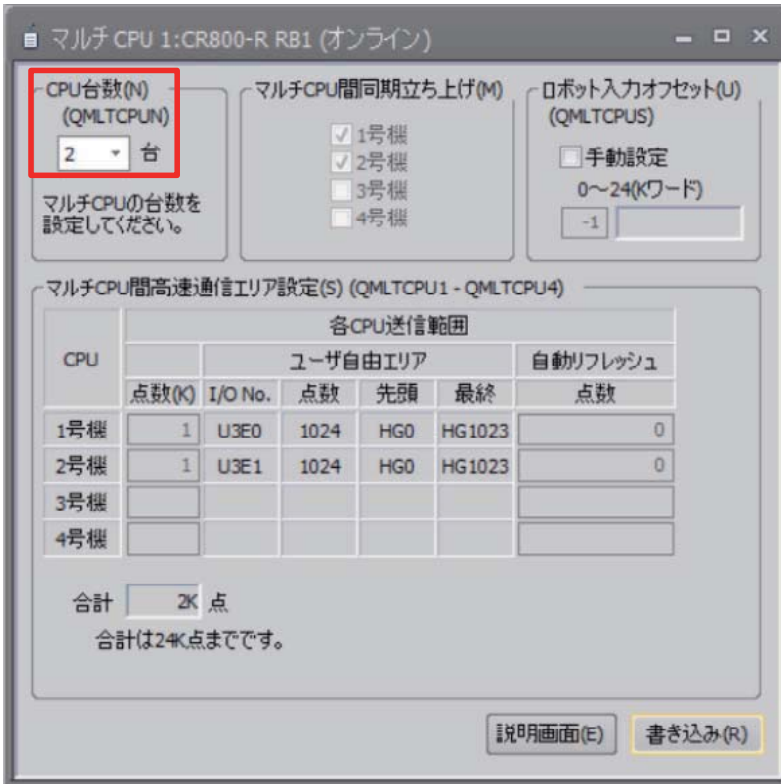
| 设定项目       |                              | 说明                                                                                                                                                                                                                | 设定值                                                                                                                                                                                                                                                                                         |
|------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU 间通信设定  | 号机单位的数据                      | 在通过 CPU 模块间刷新的数据通信中，防止各号机的数据分离并进行数据的发送接收时选择。处理 64 位数据以上的数据时可能发生数据的分离。                                                                                                                                             | 保证<br>不保证<br><br>机器人 CPU 的可编程控制器链接输入输出功能的更新周期如下所示。<br><号机单位的数据 - “不保证”><br>指定的固定周期通信间隔。<br><号机单位的数据 - “保证”><br>即使将固定周期通信间隔设定设为“0.444ms”以下，也会变为 0.888ms。<br>※ 其他数据的更新周期如各章记载所示。                                                                                                              |
|            | 固定周期通信功能                     | 设定是否使用固定周期通信功能。与机器人 CPU 组合时务必选择“使用”。                                                                                                                                                                              | 使用                                                                                                                                                                                                                                                                                          |
|            | 固定周期通信区域设定                   | 设定固定周期通信区域 <sup>注 1)</sup> 内各号机的传送区域范围。机器人所需区域如下所示。<br><CPU 缓冲存储器扩展功能 - 有效时><br>• 机器人输入区域 . . . . 1.0K<br>• 机器人输出区域 . . . . 1.0K<br><br><CPU 缓冲存储器扩展功能 - 无效时><br>• 机器人输入区域 . . . . 0.5K<br>• 机器人输出区域 . . . . 0.5K | <CPU 缓冲存储器扩展功能 - 有效><br>• 1号机 . . . . 向机器人发送数据的大小(1K)与向其他号机发送数据的大小的合计<br>• 机器人号机 . . . . 设定 1K<br>• 其他的号机 . . . . 设定自号机 CPU 的发送数据大小<br><br><CPU 缓冲存储器扩展功能 - 无效><br>• 1号机 . . . . 向机器人发送数据的大小(0.5K)与向其他号机发送数据的大小的合计<br>• 机器人号机 . . . . 设定 1K <sup>注 2)</sup><br>• 其他的号机 . . . . 设定自号机的发送数据大小 |
|            | 0.05ms 单位的设定                 | 设定是否以 0.05ms 为单位设定固定周期通信周期。与机器人 CPU 组合时请务必选择“未设定”。                                                                                                                                                                | 未设定                                                                                                                                                                                                                                                                                         |
|            | 固定周期通信间隔设定 (未以 0.05ms 为单位设定) | 从选择项目设定固定周期通信周期。请将仅使用固定周期通信功能的号机设定为同一固定周期通信周期。此外，使用机器人 CPU 的协调控制功能及干涉回避功能时，与有无号机单位的数据保证无关，请选择“0.888ms”以下。                                                                                                         | 0.222ms<br>0.444ms<br>0.888ms<br>1.777ms<br>3.555ms<br>7.111ms                                                                                                                                                                                                                              |
| 动作模式设定     | 停止设定                         | 各号机中发生重度异常或中度异常时，对全部号机是否停止动作进行设定。                                                                                                                                                                                 | 重度：全部号机停止、中度：全部号机继续<br>※ 应在全部 CPU 模块中设定。                                                                                                                                                                                                                                                    |
|            | 同步启动设定                       | 在多 CPU 系统中使 CPU 模块的启动时间同步时进行设定。※ 由于机器人 CPU 启动要花 60 秒，因此应设定为同步启动。                                                                                                                                                  | 同步                                                                                                                                                                                                                                                                                          |
| 其他号机管理模块设定 | 组外的输入输出设定                    | 设定为可获取管理外模块的输入 (X) 的 ON/OFF 数据和其他号机 CPU 的输出 (Y) 的 ON/OFF 数据。                                                                                                                                                      | 获取。                                                                                                                                                                                                                                                                                         |

注 1) 关于多 CPU 及固定周期通信区域，请参照 RCP 的手册 (MELSEC iQ-R CPU 模块用户手册 (应用篇))。

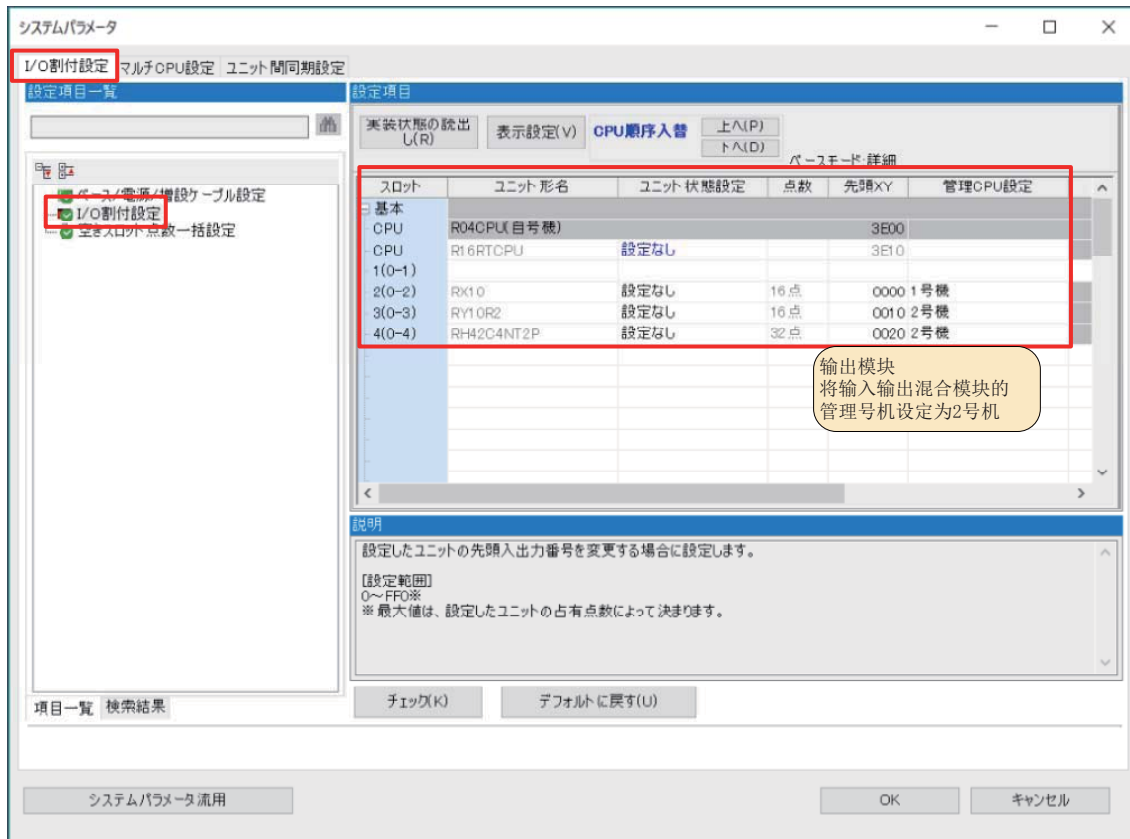
注 2) 由于区域只能以 1K 为单位进行设定，因此 0.5K 时分配为 1K。

接着，RT ToolBox3 的参数菜单的“多 CPU 设定”。

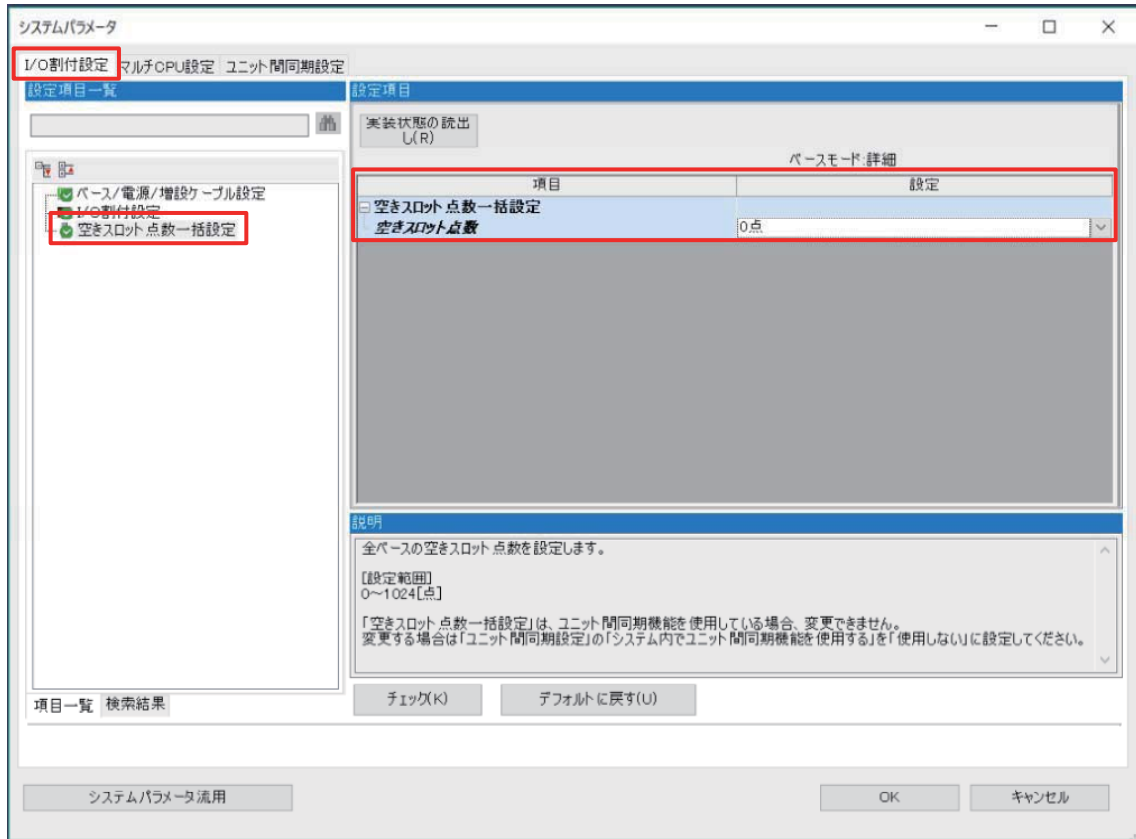
由于“多 CPU 间同步启动”及“多 CPU 间高速通讯区域设定”从可编程控制器 CPU 自动取得设定值，因此不需要设定。



然后，根据 GX Works3 的 CPU 参数，如下所示实施“I/O 分配设定”。



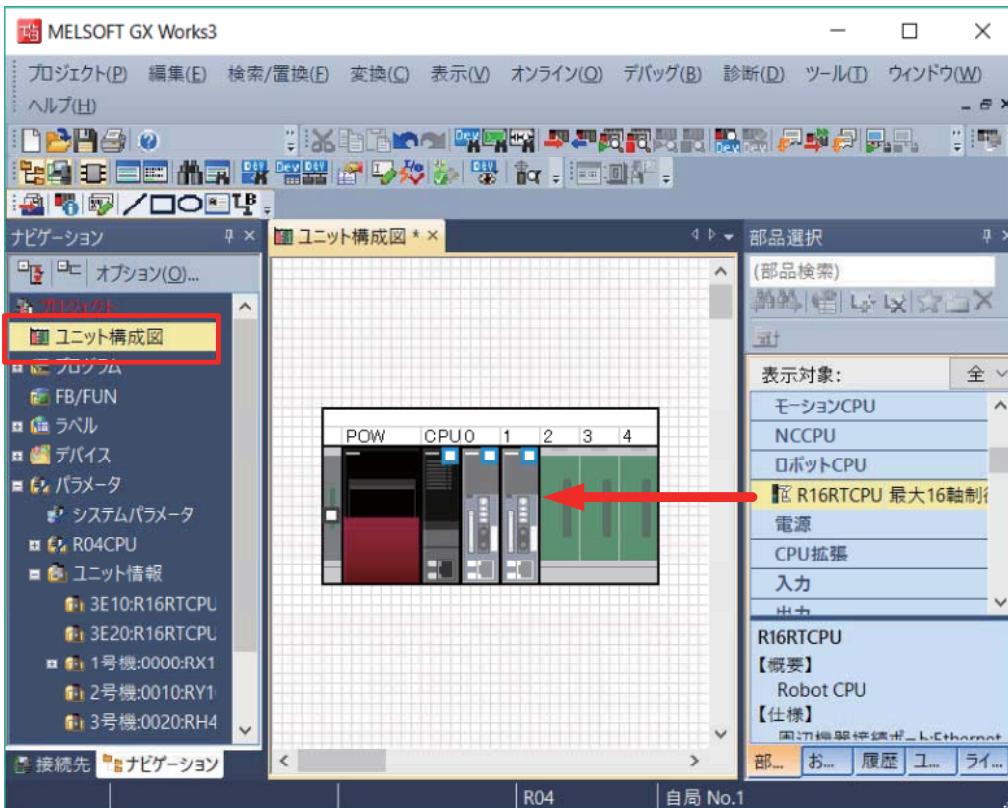
输入模块设定为可编程控制器管理，输出模块和输入输出混合模块设定为机器人 CPU1 管理。  
此时，存在空插槽时，如下所示实施“空插槽点数批量设定”。



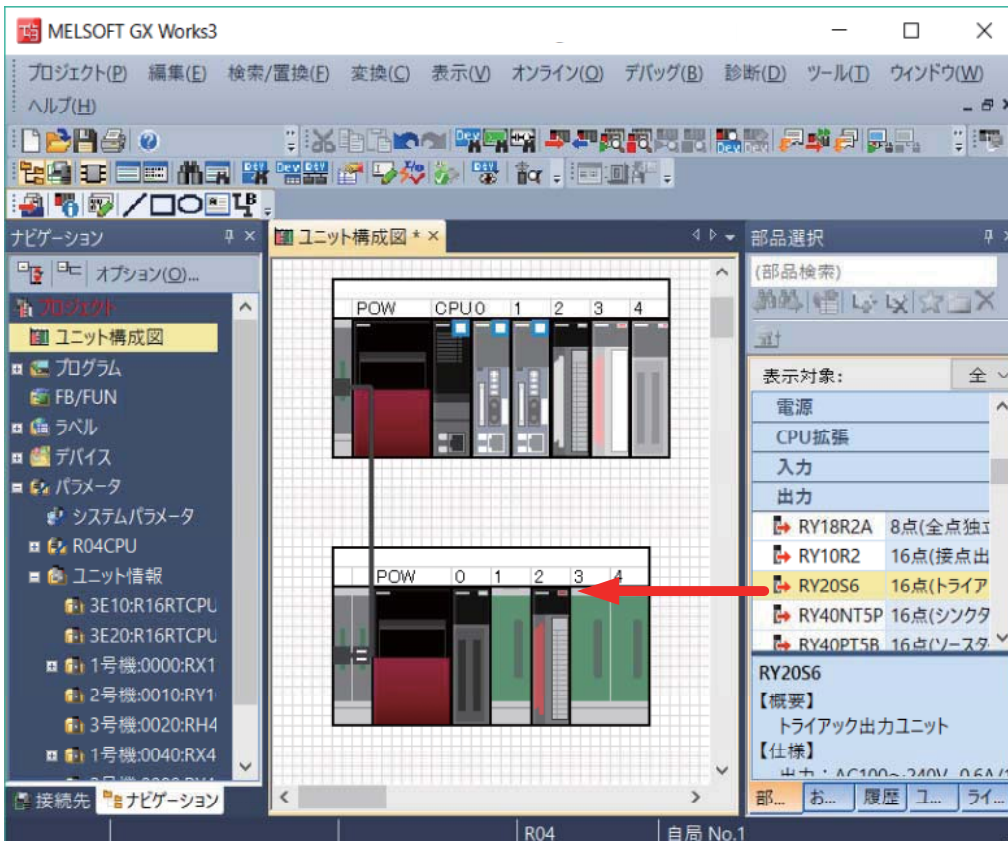
设定以上参数后，将可编程控制器及机器人 CPU 的电源复位。

(4) 设定系统构成示例 2 的参数

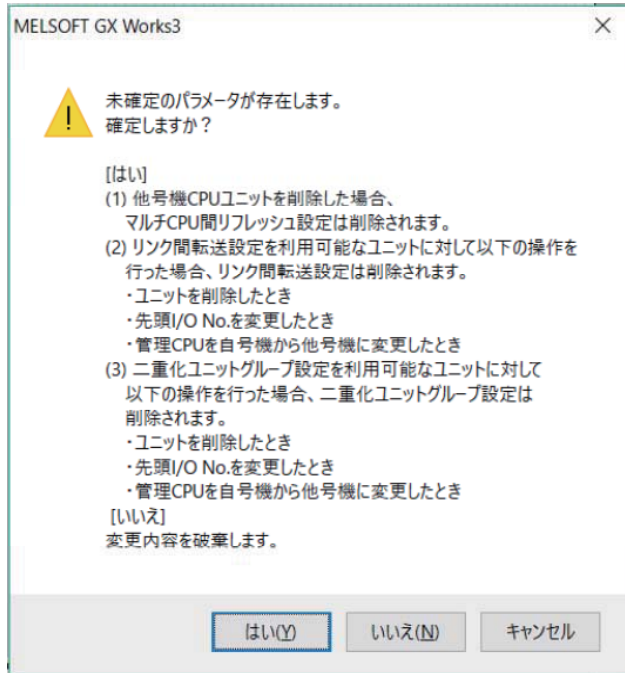
根据系统构成示例 2（机器人 CPU2 台 + 3 种输入输出模块 + 扩展基板）的条件，配置 GX Works3 的“模块构成图”中使用的基座、电源模块、可编程控制器 CPU、机器人 CPU。



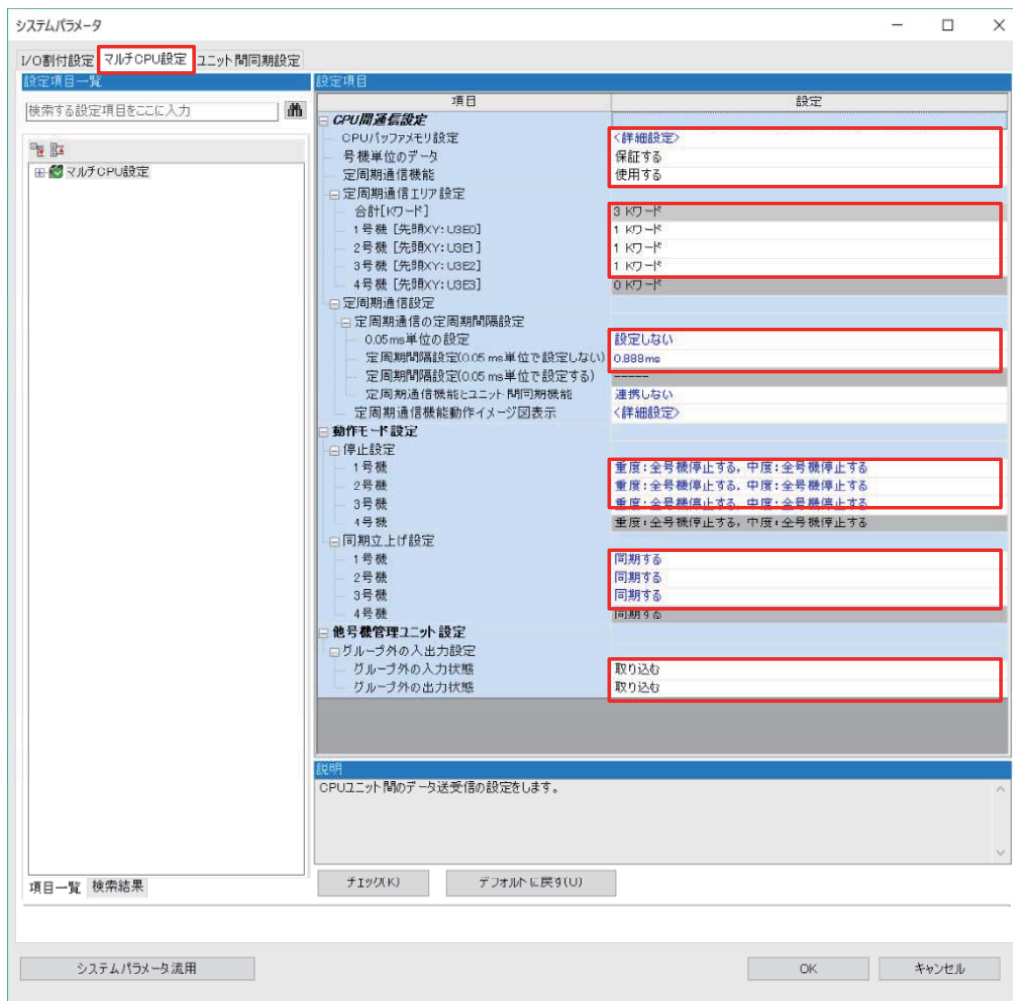
接着，配置使用的扩展基板、输入模块、输出模块、输入输出混合模块。



对模块进行配置后，关闭模块构成图画面，即显示下图的对话框，选择“是”。



接着，打开系统参数画面，实施多CPU的设定。



| 设定项目       |                                 | 说明                                                                                                                                                                                                                | 设定值                                                                                                                                                                                                                                                                                         |
|------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU 间通信设定  | 号机单位的数据                         | 在通过 CPU 模块间刷新的数据通信中，防止各号机的数据分离并进行数据的发送接收时选择。处理 64 位数据以上的数据时可能发生数据的分离。                                                                                                                                             | 保证<br>不保证<br><br>机器人 CPU 的可编程控制器链接输入输出功能的更新周期如下所示。<br><号机单位的数据 - “不保证”><br>指定的固定周期通信间隔。<br><号机单位的数据 - “保证”><br>即使将固定周期通信间隔设定设为“0.444ms”以下，也会变为 0.888ms。<br>※ 其他数据的更新周期如各章记载所示。                                                                                                              |
|            | 固定周期通信功能                        | 设定是否使用固定周期通信功能。与机器人 CPU 组合时务必选择“使用”。                                                                                                                                                                              | 使用                                                                                                                                                                                                                                                                                          |
|            | 固定周期通信区域设定                      | 设定固定周期通信区域 <sup>注 1)</sup> 内各号机的传送区域范围。机器人所需区域如下所示。<br><CPU 缓冲存储器扩展功能 - 有效时><br>• 机器人输入区域 . . . . 1.0K<br>• 机器人输出区域 . . . . 1.0K<br><br><CPU 缓冲存储器扩展功能 - 无效时><br>• 机器人输入区域 . . . . 0.5K<br>• 机器人输出区域 . . . . 0.5K | <CPU 缓冲存储器扩展功能 - 有效><br>• 1号机 . . . . 向机器人发送数据的大小(1K)与向其他号机发送数据的大小的合计<br>• 机器人号机 . . . . 设定 1K<br>• 其他的号机 . . . . 设定自号机 CPU 的发送数据大小<br><br><CPU 缓冲存储器扩展功能 - 无效><br>• 1号机 . . . . 向机器人发送数据的大小(0.5K)与向其他号机发送数据的大小的合计<br>• 机器人号机 . . . . 设定 1K <sup>注 2)</sup><br>• 其他的号机 . . . . 设定自号机的发送数据大小 |
|            | 0.05ms 单位的设定                    | 设定是否以 0.05ms 为单位设定固定周期通信周期。与机器人 CPU 组合时请务必选择“未设定”。                                                                                                                                                                | 未设定                                                                                                                                                                                                                                                                                         |
|            | 固定周期通信间隔设定<br>(未以 0.05ms 为单位设定) | 从选择项目设定固定周期通信周期。请将仅使用固定周期通信功能的号机设定为同一固定周期通信周期。此外，使用机器人 CPU 的协调控制功能及干涉回避功能时，与有无号机单位的数据保证无关，请选择“0.888ms”以下。                                                                                                         | 0.222ms<br>0.444ms<br>0.888ms<br>1.777ms<br>3.555ms<br>7.111ms                                                                                                                                                                                                                              |
| 动作模式设定     | 停止设定                            | 各号机中发生重度异常或中度异常时，对全部号机是否停止动作进行设定。                                                                                                                                                                                 | 重度：全部号机停止、中度：全部号机继续<br>※ 应在全部 CPU 模块中设定。                                                                                                                                                                                                                                                    |
|            | 同步启动设定                          | 在多 CPU 系统中使 CPU 模块的启动时间同步时进行设定。<br>※ 由于机器人 CPU 启动要花 60 秒，因此应设定为同步启动。                                                                                                                                              | 同步                                                                                                                                                                                                                                                                                          |
| 其他号机管理模块设定 | 组外的输入输出设定                       | 设定为可获取管理外模块的输入 (X) 的 ON/OFF 数据和其他号机 CPU 的输出 (Y) 的 ON/OFF 数据。                                                                                                                                                      | 获取。                                                                                                                                                                                                                                                                                         |

注 1) 关于多 CPU 及固定周期通信区域，请参照 RCP 的手册 (MELSEC iQ-R CPU 模块用户手册 (应用篇))。

注 2) 由于区域只能以 1K 为单位进行设定，因此 0.5K 时分配为 1K。



接着，RT ToolBox3 的参数菜单的“多 CPU 设定”。

由于“多 CPU 间同步启动”及“多 CPU 间高速通讯区域设定”从可编程控制器 CPU 自动取得设定值，因此不需要设定。

【机器人 CPU1 号机 (=2 号机)】

マルチ CPU 1:CR800-R RB1 (オンライン)

CPU台数(N) (QMLTCPUN)  
3 台  
マルチCPUの台数を設定してください。

マルチCPU間同期立ち上げ(M)  
 1号機  
 2号機  
 3号機  
 4号機

ロボット入力オフセット(U) (QMLTCPUS)  
 手動設定  
 0~24(Kワード)  
 -1

マルチCPU間高速通信エリア設定(S) (QMLTCPU1 - QMLTCPU4)

| CPU | 各CPU送信範囲 |         |      |     |        | 自動リフレッシュ<br>点数 |
|-----|----------|---------|------|-----|--------|----------------|
|     | 点数(K)    | I/O No. | 点数   | 先頭  | 最終     |                |
| 1号機 | 1        | U3E0    | 1024 | HG0 | HG1023 | 0              |
| 2号機 | 1        | U3E1    | 1024 | HG0 | HG1023 | 0              |
| 3号機 | 1        | U3E2    | 1024 | HG0 | HG1023 | 0              |
| 4号機 |          |         |      |     |        |                |

合計 3K 点  
合計は24K点までです。

説明画面(E) 書き込み(R)

【机器人 CPU2 号机 (=3 号机)】

マルチ CPU 2:CR800-R RB2 (オンライン)

CPU台数(N) (QMLTCPUN)  
3 台  
マルチCPUの台数を設定してください。

マルチCPU間同期立ち上げ(M)  
 1号機  
 2号機  
 3号機  
 4号機

ロボット入力オフセット(U) (QMLTCPUS)  
 手動設定  
 0~24(Kワード)  
 -1

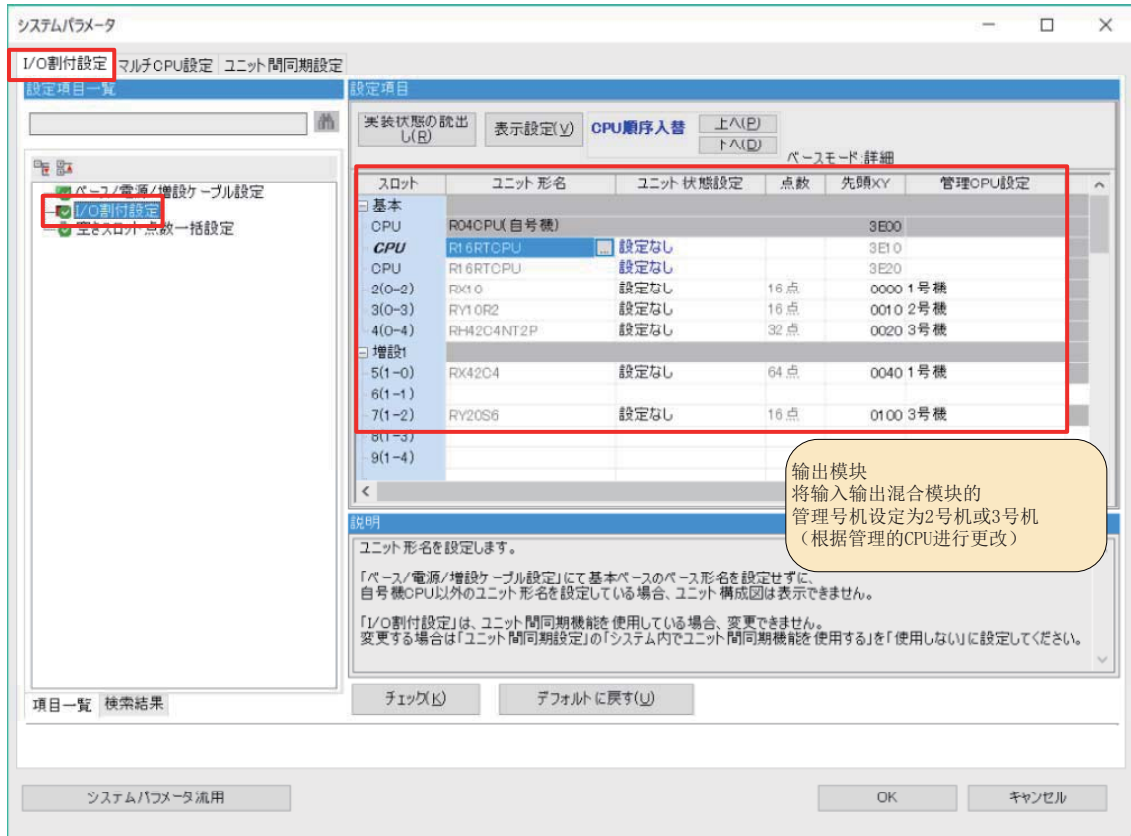
マルチCPU間高速通信エリア設定(S) (QMLTCPU1 - QMLTCPU4)

| CPU | 各CPU送信範囲 |         |      |     |        | 自動リフレッシュ<br>点数 |
|-----|----------|---------|------|-----|--------|----------------|
|     | 点数(K)    | I/O No. | 点数   | 先頭  | 最終     |                |
| 1号機 | 1        | U3E0    | 1024 | HG0 | HG1023 | 0              |
| 2号機 | 1        | U3E1    | 1024 | HG0 | HG1023 | 0              |
| 3号機 | 1        | U3E2    | 1024 | HG0 | HG1023 | 0              |
| 4号機 |          |         |      |     |        |                |

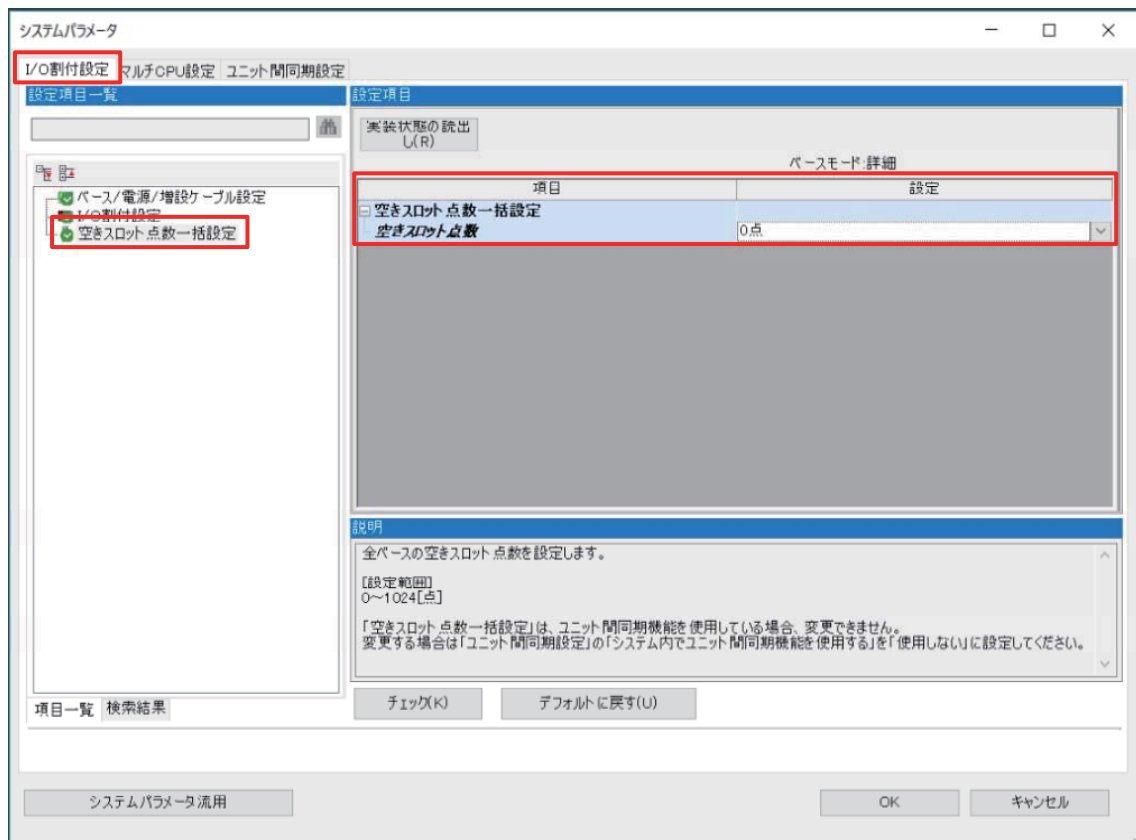
合計 3K 点  
合計は24K点までです。

説明画面(E) 書き込み(R)

然后，根据 GX Works3 的 CPU 参数，如下所示实施 “I/O 分配设定”。



输入模块设定为可编程控制器管理，输出模块和输入输出混合模块设定为机器人CPU的1号机和2号机管理。此时，存在空插槽时，如下所示实施 “空插槽点数批量设定”。



设定以上参数后，将可编程控制器及机器人CPU 的电源复位。

## (5) 使用状态变量进行控制

用于控制可编程控制器输入输出模块的状态变量存在以下类型

| 变量名     | 内容                                                                                                                                 |
|---------|------------------------------------------------------------------------------------------------------------------------------------|
| 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)                                                     |
| 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                                     |
| 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                                     |
| 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                                  |
| M_YDev  | 以位为单位读取 / 写入 PLC 输出信号 (Y)。<br>例) 1 M_YDev(2)=1 ' 将可编程控制器输出信号 2 号设为 ON                                                              |
| 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                                      |
| 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                                    |
| 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 |

例如，系统构成示例 2 时，进行如下设定。

|           | 先頭I/O No. | 点数  | ユニット形名     | シリーズ | エラー状態 | ユニット構成 | 管理 CPU | ネットワーク情報 (ポート1) | ネットワーク情報 (ポート2) |
|-----------|-----------|-----|------------|------|-------|--------|--------|-----------------|-----------------|
| 基本-電源     | -         | -   | R64P       | -    | -     | 電源     | -      | -               | -               |
| 基本-CPU    | 3E00      | -   | R04CPU     | iQ-R | -     | CPU    | -      | -               | -               |
| 基本-CPU    | 3E10      | -   | R16RTCPU   | iQ-R | -     | CPU    | -      | -               | -               |
| 基本-CPU    | 3E20      | -   | R16RTCPU   | iQ-R | -     | CPU    | -      | -               | -               |
| 基本-I/O 2  | 0000      | 16点 | RX10       | iQ-R | -     | 入力     | 1号機    | -               | -               |
| 基本-I/O 3  | 0010      | 16点 | RY10R2     | iQ-R | -     | 出力     | 2号機    | -               | -               |
| 基本-I/O 4  | 0020      | 32点 | RH42C4NT2P | iQ-R | -     | 入出力    | 3号機    | -               | -               |
| 増設1-電源    | -         | -   | R62P       | -    | -     | 電源     | -      | -               | -               |
| 増設1-I/O 0 | 0040      | 64点 | Rx42C4     | iQ-R | -     | 入力     | 1号機    | -               | -               |
| 増設1-I/O 1 | -         | -   | 空き         | -    | -     | -      | -      | -               | -               |
| 増設1-I/O 2 | 0100      | 16点 | RY20S6     | iQ-R | -     | 出力     | 3号機    | -               | -               |
| 増設1-I/O 3 | -         | -   | 空き         | -    | -     | -      | -      | -               | -               |
| 増設1-I/O 4 | -         | -   | 空き         | -    | -     | -      | -      | -               | -               |

想显示安装在 1 号机 (= 可编程控制器) 管理的扩展基板 1 的插槽 0 中的输入模块的输入信息时，在 2 号机 (= 机器人 CPU1 号机) 中，准备如下机器人程序。

```
M1&=M_XDevD(&H40)
```

指定对应的模块的“起始I/O”编号。

读取输入信号32bit的状态变量。

接着，对输出至自己管理的输出模块、或输入输出模块的情况进行说明。

|           | 先頭I/O No. | 点数  | ユニット形名     | シリーズ | エラー状態 | ユニット構成 | 管理CPU | ネットワーク情報 (ポート1) | ネットワーク情報 (ポート2) |
|-----------|-----------|-----|------------|------|-------|--------|-------|-----------------|-----------------|
| 基本-電源     | -         | -   | R64P       | -    | -     | 電源     | -     | -               | -               |
| 基本-CPU    | 3E00      | -   | R04CPU     | iQ-R | -     | CPU    | -     | -               | -               |
| 基本-CPU    | 3E10      | -   | R16RTCPU   | iQ-R | -     | CPU    | -     | -               | -               |
| 基本-CPU    | 3E20      | -   | R16RTCPU   | iQ-R | -     | CPU    | -     | -               | -               |
| 基本-I/O 2  | 0000      | 16点 | RX10       | iQ-R | -     | 入力     | 1号機   | -               | -               |
| 基本-I/O 3  | 0010      | 16点 | RY10R2     | iQ-R | -     | 出力     | 2号機   | -               | -               |
| 基本-I/O 4  | 0020      | 32点 | RH42C4NT2P | iQ-R | -     | 入出力    | 3号機   | -               | -               |
| 増設1-電源    | -         | -   | R82P       | -    | -     | 電源     | -     | -               | -               |
| 増設1-I/O 0 | 0040      | 64点 | RX42C4     | iQ-R | -     | 入力     | 1号機   | -               | -               |
| 増設1-I/O 1 | -         | -   | 空未         | -    | -     | -      | -     | -               | -               |
| 増設1-I/O 2 | 0100      | 16点 | RY20S6     | iQ-R | -     | 出力     | 3号機   | -               | -               |
| 増設1-I/O 3 | -         | -   | 空未         | -    | -     | -      | -     | -               | -               |
| 増設1-I/O 4 | -         | -   | 空未         | -    | -     | -      | -     | -               | -               |

例如，将信号输出至安装在上图所示扩展基板 1 的插槽 2 中的输出模块时，在 3 号机 (= 机器人 CPU2 号机) 中，准备如下机器人程序。

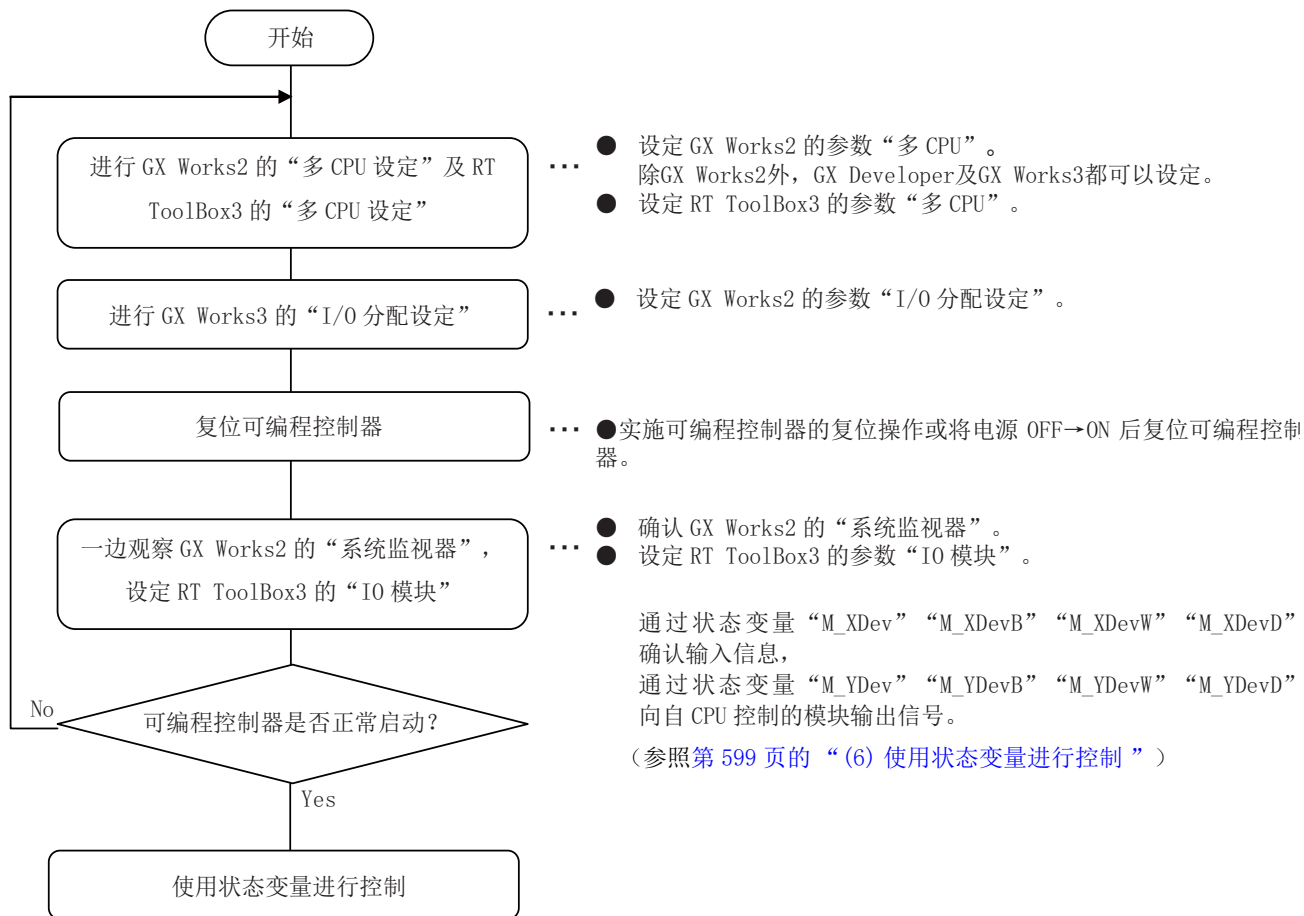
```
M_YDevD(&H100) = &H1234
```

- └─ 指定想输出的数据。
- └─ 指定对应的模块的“起始I/O”编号。
- └─ 将数据进行32bit输出的状态变量。

5.25.2 CR800-Q系列

(1) 作业流程

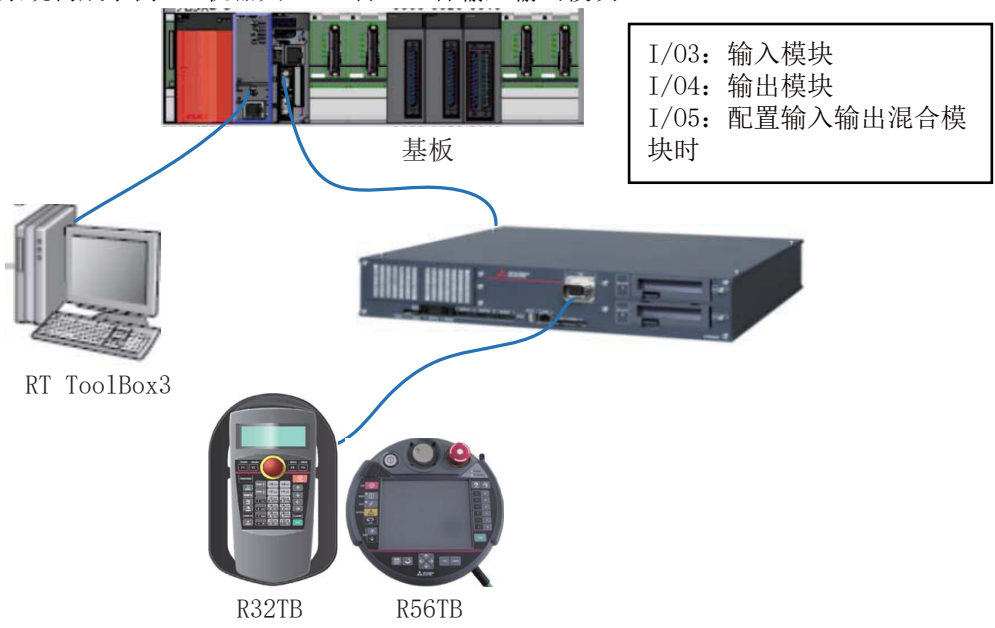
为了控制可编程控制器用输入输出模块，所需的作业流程如下所示。



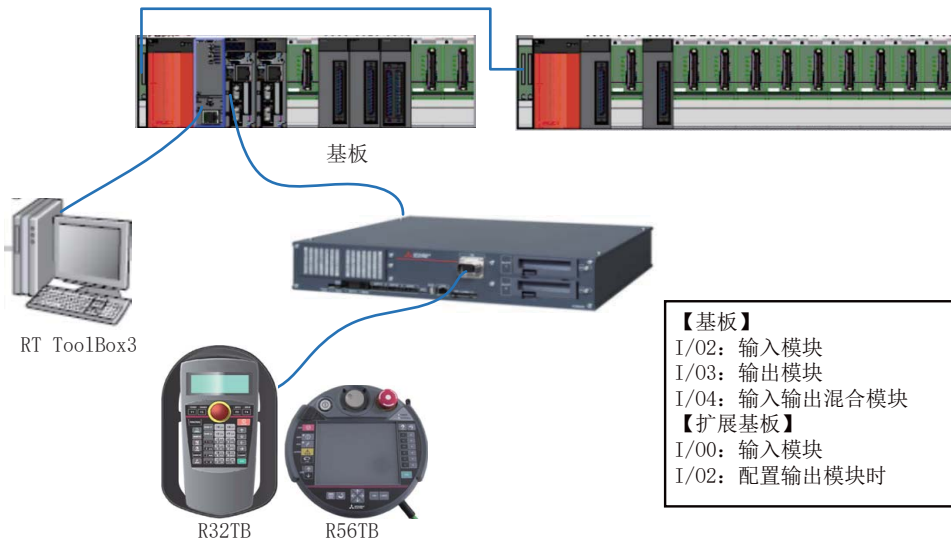
(2) 硬件的构成示例

将“机器人 CPU1 台 + 3 种输入输出模块”的构成作为系统构成示例 1、将“机器人 CPU2 台 + 3 种输入输出模块 + 扩展基板追加”的构成作为系统构成示例 2 进行以下说明。

■ 系统构成示例 1: 机器人 CPU1 台 + 3 种输入输出模块



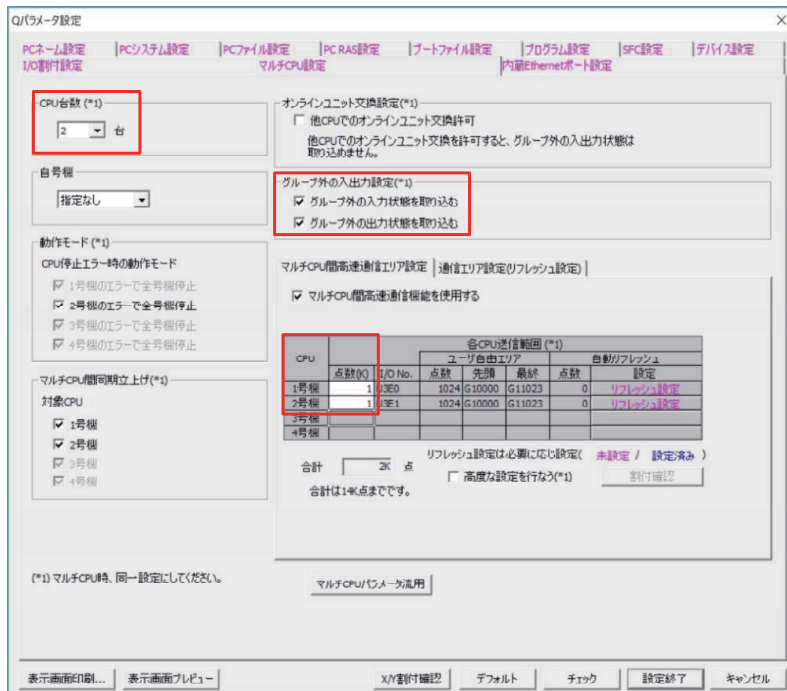
■ 系统构成示例 2: 机器人 CPU1 台 + 3 种输入输出模块 + 扩展基板的追加



(3) 设定系统构成示例 1 的参数

根据系统构成示例 1（机器人 CPU1 台 + 3 种输入输出模块）的条件，实施 GX Works2 的 PC 参数中的“多 CPU 设定”、及 RT ToolBox3 的“多 CPU 设定”。

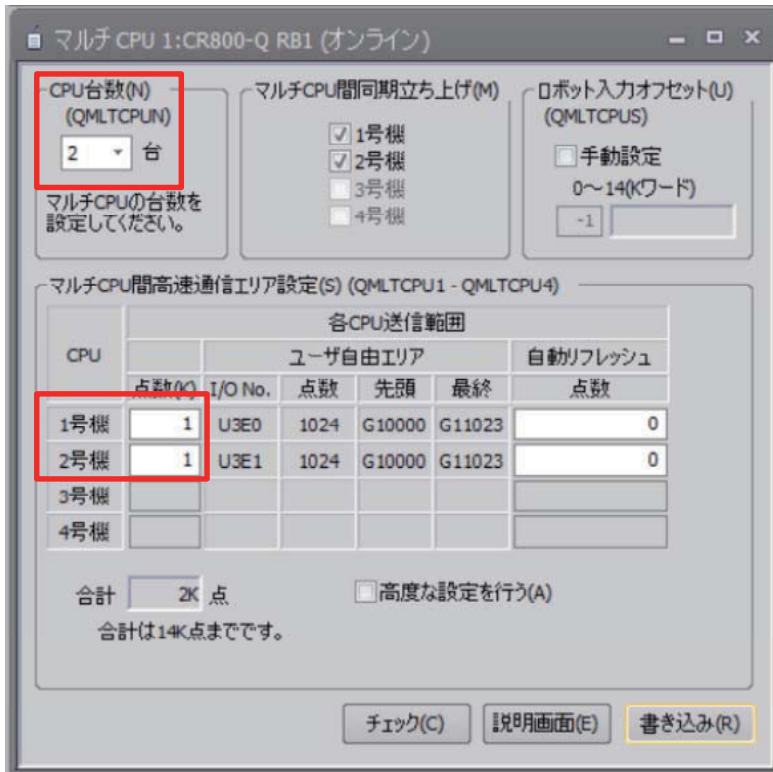
根据 GX Works3 的 CPU 参数，如下所示实施“多 CPU 设定”。



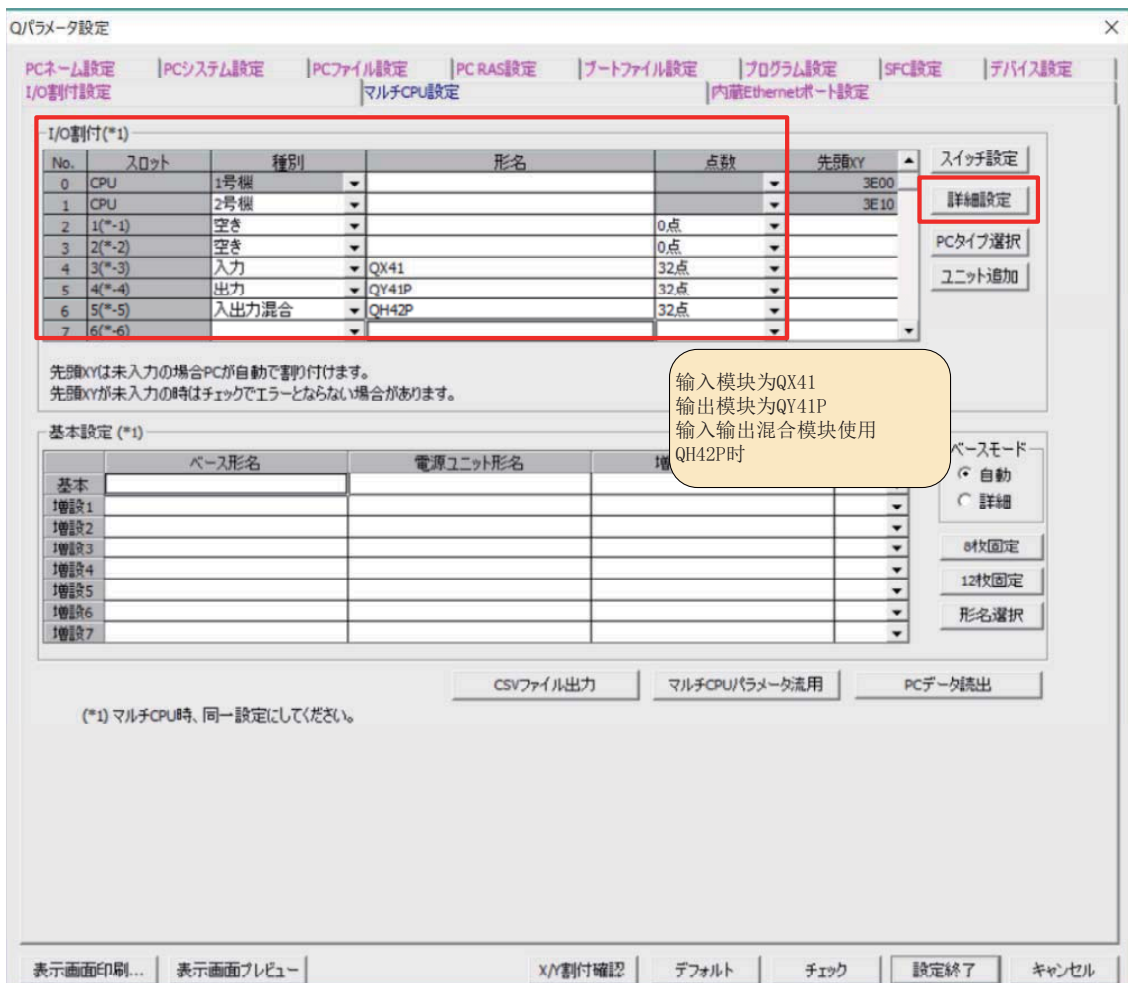
| 设定项目            | 说明                                                           | 设定值                                                                                                                                              |
|-----------------|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU 台数          | 设定在多 CPU 系统中使用的 CPU 模块的台数。                                   | 2                                                                                                                                                |
| 组外的输入输出设定       | 设定为可获取管理外模块的输入 (X) 的 ON/OFF 数据和其他号机 CPU 的输出 (Y) 的 ON/OFF 数据。 | 将检查设为 ON。                                                                                                                                        |
| 多 CPU 间高速通信区域设定 | 使用多 CPU 间高速通信区域 <sup>注 1)</sup> 进行数据传送时设定。                   | <ul style="list-style-type: none"> <li>• 1 号机 (= 可编程控制器) ... 向机器人发送数据的大小 (1K) 与向其他号机发送数据的大小的合计</li> <li>• 2 号机 (= 机器人 1 号机) ... 设定 1K</li> </ul> |

注 1) 关于多 CPU 及多 CPU 间高速通信区域，请参照 QCPU 的手册（QCPU 用户手册 多 CPU 系统篇）。

相当地，如下所示实施 RT ToolBox3 的参数菜单的“多 CPU 设定”。



然后，根据 GX Works2 的 PC 参数，如下所示实施“I/O 分配设定”。



点击 [ 详细设定 ] 按钮，设定如下管理 CPU。

I/Oユニット、インテリジェント機能ユニット詳細設定

|    | スロット     | 種別    | 形名    | エラー時<br>出力モード | H/Wエラー時<br>CPU動作モード | I/O応答時間 | 管理CPU(*1) |
|----|----------|-------|-------|---------------|---------------------|---------|-----------|
| 0  | CPU      | 1号機   |       |               |                     |         |           |
| 1  | CPU      | 2号機   |       |               |                     |         |           |
| 2  | 1(*-1)   | 空き    |       |               |                     |         | 1号機       |
| 3  | 2(*-2)   | 空き    |       |               |                     |         | 1号機       |
| 4  | 3(*-3)   | 入力    | QX41  |               |                     | 10ms    | 1号機       |
| 5  | 4(*-4)   | 出力    | QY41P | クリア           |                     |         | 2号機       |
| 6  | 5(*-5)   | 入出力混合 | QH42P | クリア           |                     | 10ms    | 2号機       |
| 7  | 6(*-6)   |       |       |               |                     |         | 1号機       |
| 8  | 7(*-7)   |       |       |               |                     |         | 1号機       |
| 9  | 8(*-8)   |       |       |               |                     |         | 1号機       |
| 10 | 9(*-9)   |       |       |               |                     |         | 1号機       |
| 11 | 10(*-10) |       |       |               |                     |         | 1号機       |
| 12 | 11(*-11) |       |       |               |                     |         | 1号機       |
| 13 | 12(*-12) |       |       |               |                     |         | 1号機       |
| 14 | 13(*-13) |       |       |               |                     |         | 1号機       |
| 15 | 14(*-14) |       |       |               |                     |         | 1号機       |

(\*1) マルチCPU時、同一設定にしてください。

設定終了      キャンセル

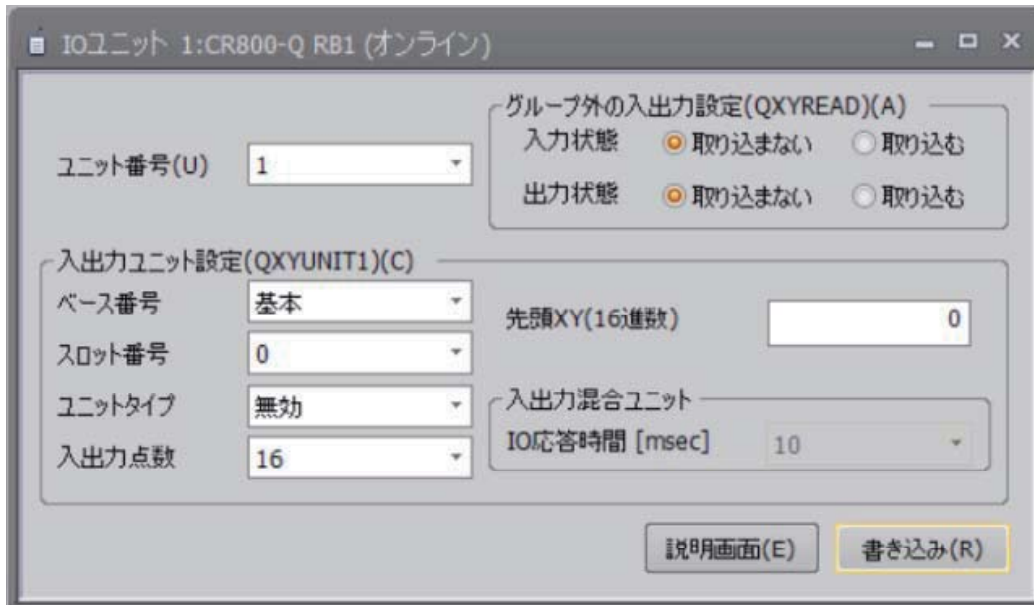
输入模块设定为可编程控制器管理，输出模块和输入输出混合模块设定为机器人 CPU1 管理。



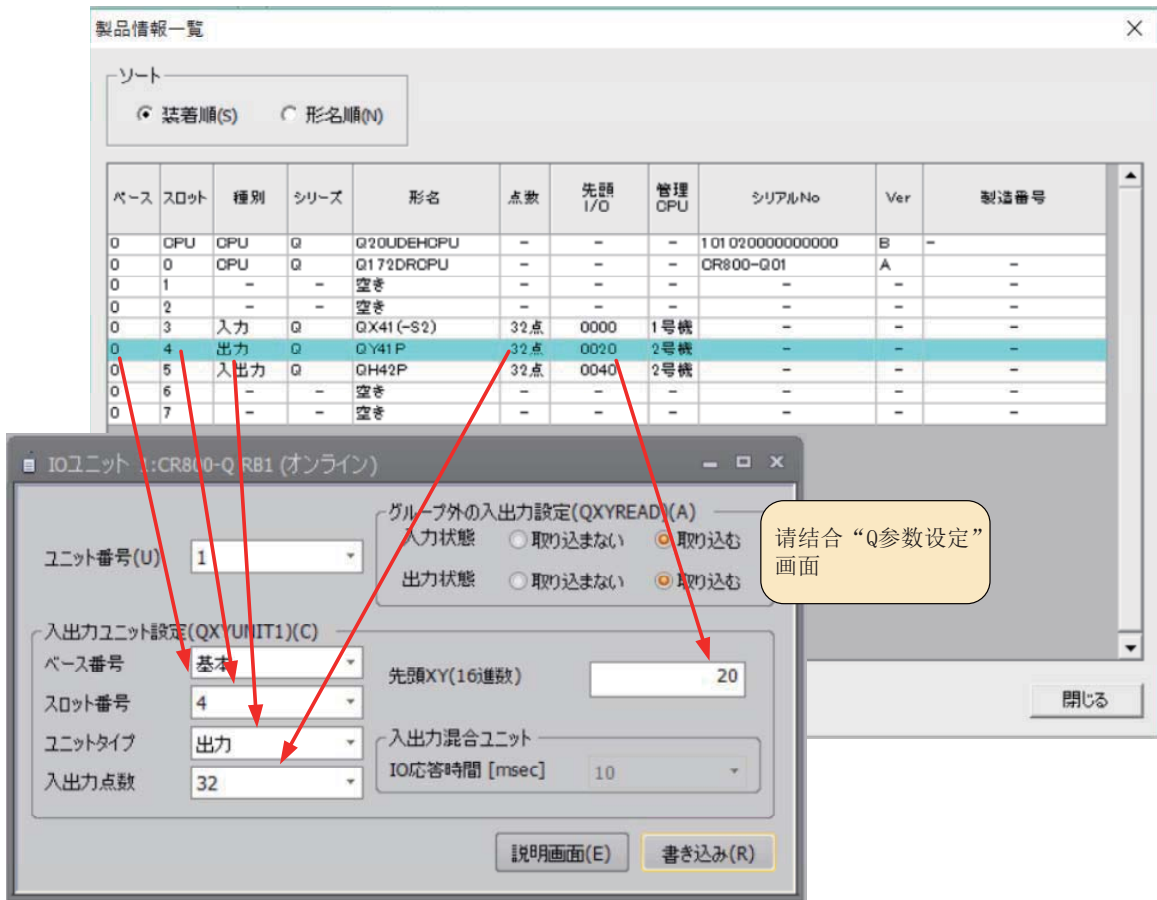
相同地，设定 RT ToolBox3 的参数，但暂且将可编程控制器的电源复位。  
可编程控制器启动后，从 GX Works2 的 [ 诊断 ] 菜单选择 [ 系统监视器 ]。



而且，关于机器人 CPU 管理的输入输出模块，在 RT ToolBox3 中设定参数。  
双击“参数”菜单的“IO 模块”后，打开“IO 模块”参数画面。

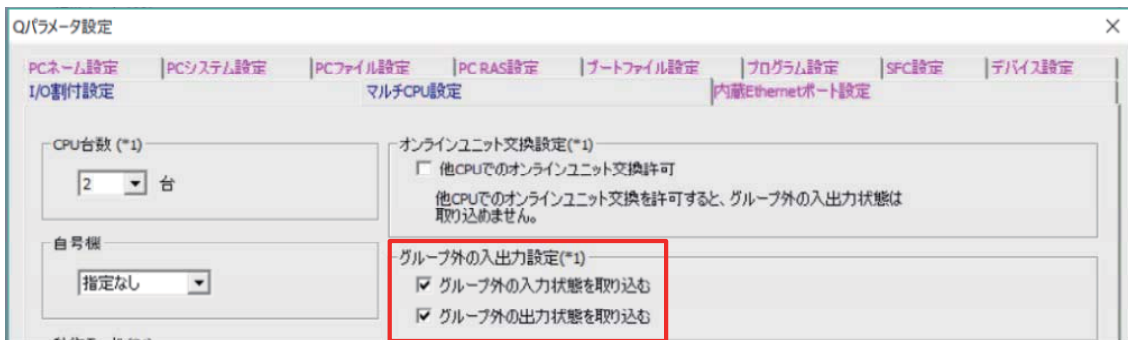


从 GX Works2 的“系统监视器”画面，点击对应的模块后，点击 [ 产品信息一览 ] 按钮，一边观察显示“产品信息一览”，一边将 RT ToolBox3 的“IO 模块”参数进行如下设定。



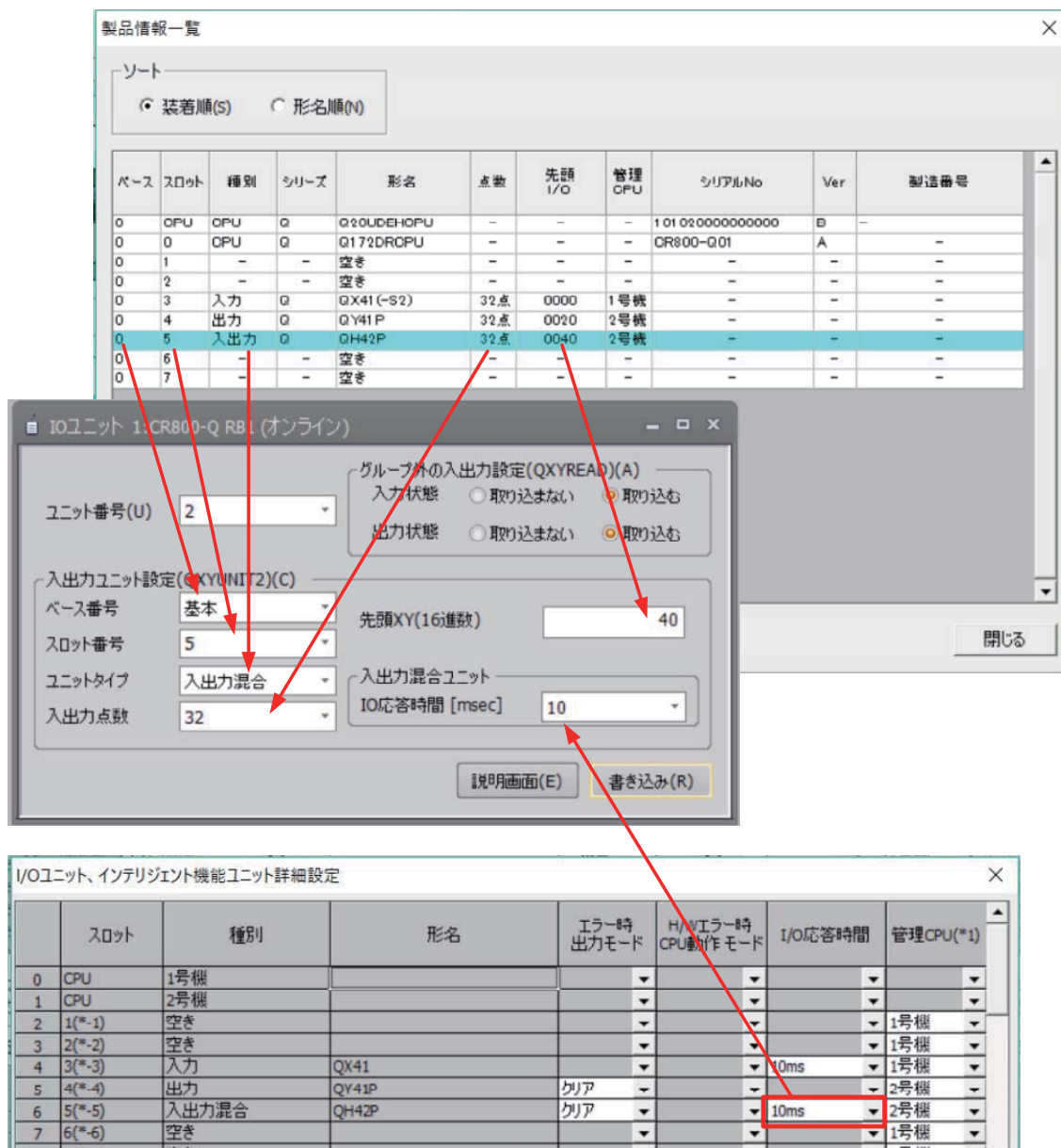
通过 1 台机器人 CPU，可以最多控制 4 台输入输出模块。  
[ 模块编号 ] 可设定为“1”~“4”，”表示其中 4 台中任意一台的设定。

关于“组外的输入输出设定 (QXYREAD)”的设定，请结合以下所示 GX Works2 的“Q 参数设定”画面的设定。



接着实施第 2 台模块的设置。

从 GX Works2 的“系统监视器”画面，点击对应的模块后，点击 [ 产品信息一览 ] 按钮，一边观察显示“产品信息一览”，一边将 RT ToolBox3 的“IO 模块”参数进行如下设定。



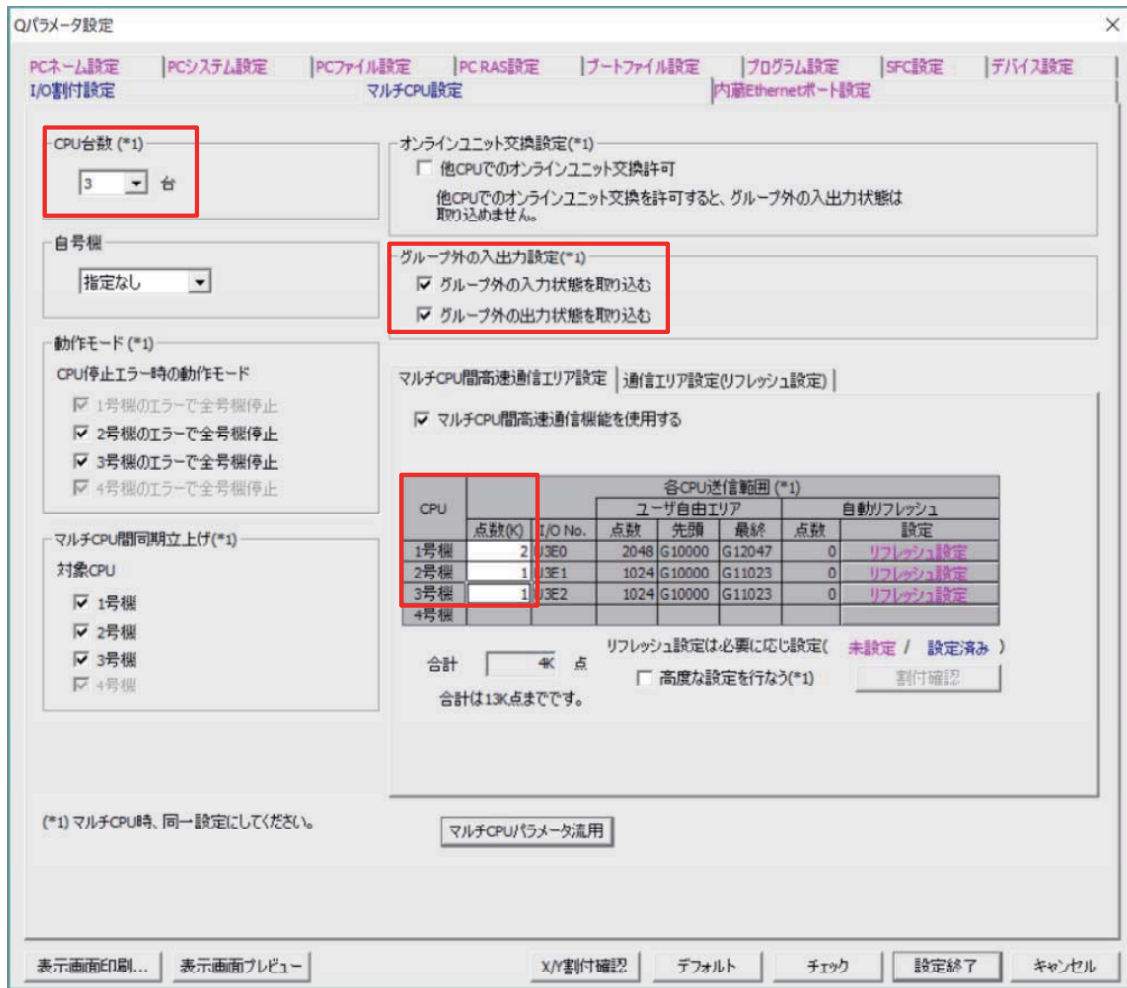
关于“IO 响应时间”，请结合点击上述“Q 参数设定”的 [ 详细设定 ] 按钮后显示的画面的值。

设定以上参数后，将可编程控制器及机器人 CPU 的电源复位。

(4) 设定系统构成示例 2 的参数

根据系统构成示例 2（机器人 CPU2 台 + 3 种输入输出模块+扩展基板）的条件，实施 GX Works2 的 PC 参数中的“多 CPU 设定”、及 RT ToolBox3 的“多 CPU 设定”。

根据 GX Works2 的 CPU 参数，如下所示实施“多 CPU 设定”。

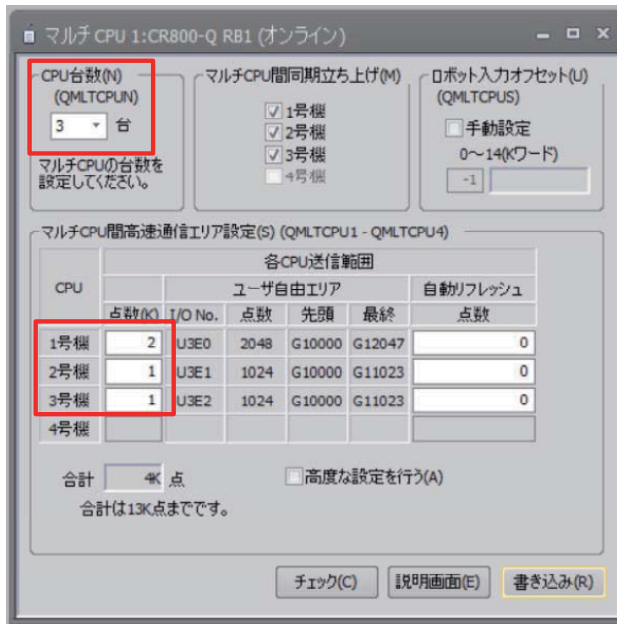


| 设定项目            | 说明                                                           | 设定值                                                                                                                                                                                           |
|-----------------|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU 台数          | 设定在多 CPU 系统中使用的 CPU 模块的台数。                                   | 3                                                                                                                                                                                             |
| 组外的输入输出设定       | 设定为可获取管理外模块的输入 (X) 的 ON/OFF 数据和其他号机 CPU 的输出 (Y) 的 ON/OFF 数据。 | 将检查设为 ON。                                                                                                                                                                                     |
| 多 CPU 间高速通信区域设定 | 使用多 CPU 间高速通信区域 <sup>注 1)</sup> 进行数据传送时设定。                   | <ul style="list-style-type: none"> <li>• 1 号机 (= 可编程控制器) . . . 向机器人发送数据的大小 (1K) 与向其他号机发送数据的大小的合计</li> <li>• 2 号机 (= 机器人 1 号机) . . . 设定 1K</li> <li>• 3 号机 (= 机器人 2 号机) . . . 设定 1K</li> </ul> |

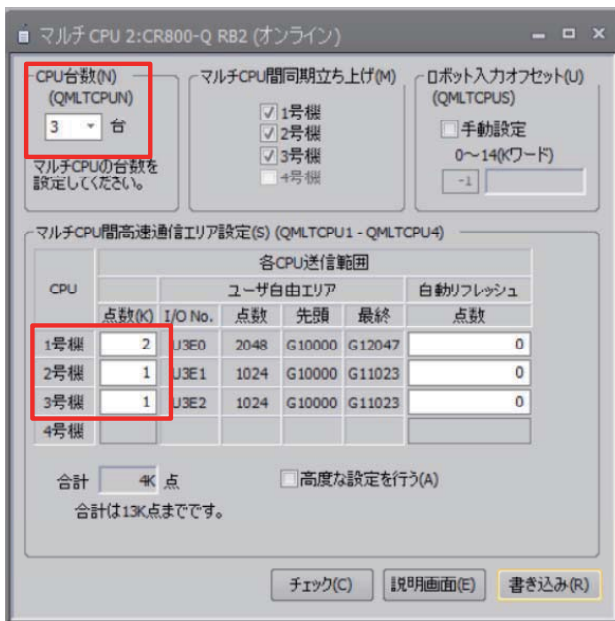
注 1) 关于多 CPU 及多 CPU 间高速通信区域，请参照 QCPU 的手册（QCPU 用户手册 多 CPU 系统篇）。

相同地，在 RT ToolBox3 中，如下所示实施机器人 CPU1 号机和 2 号机的参数菜单的“多 CPU 设定”。

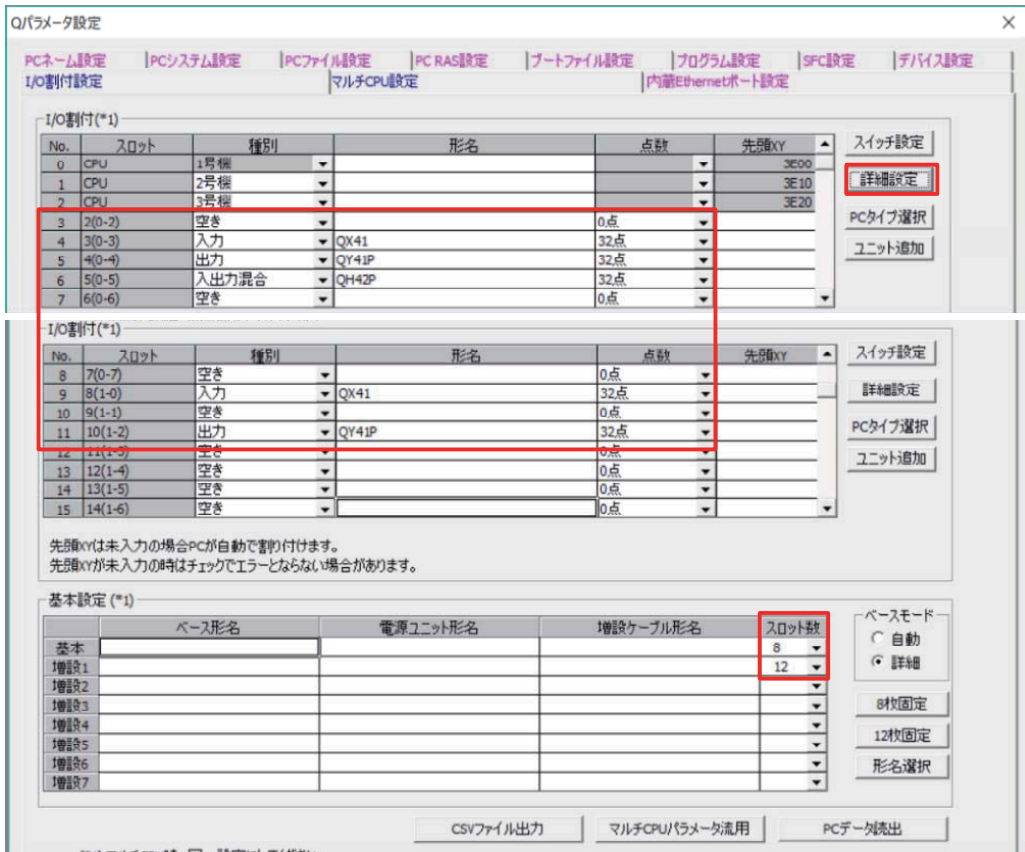
【机器人 CPU1 号机 (=2 号机)】



【机器人 CPU2 号机 (=3 号机)】



然后，根据 GX Works2 的 PC 参数，如下所示实施 “I/O 分配设定”。

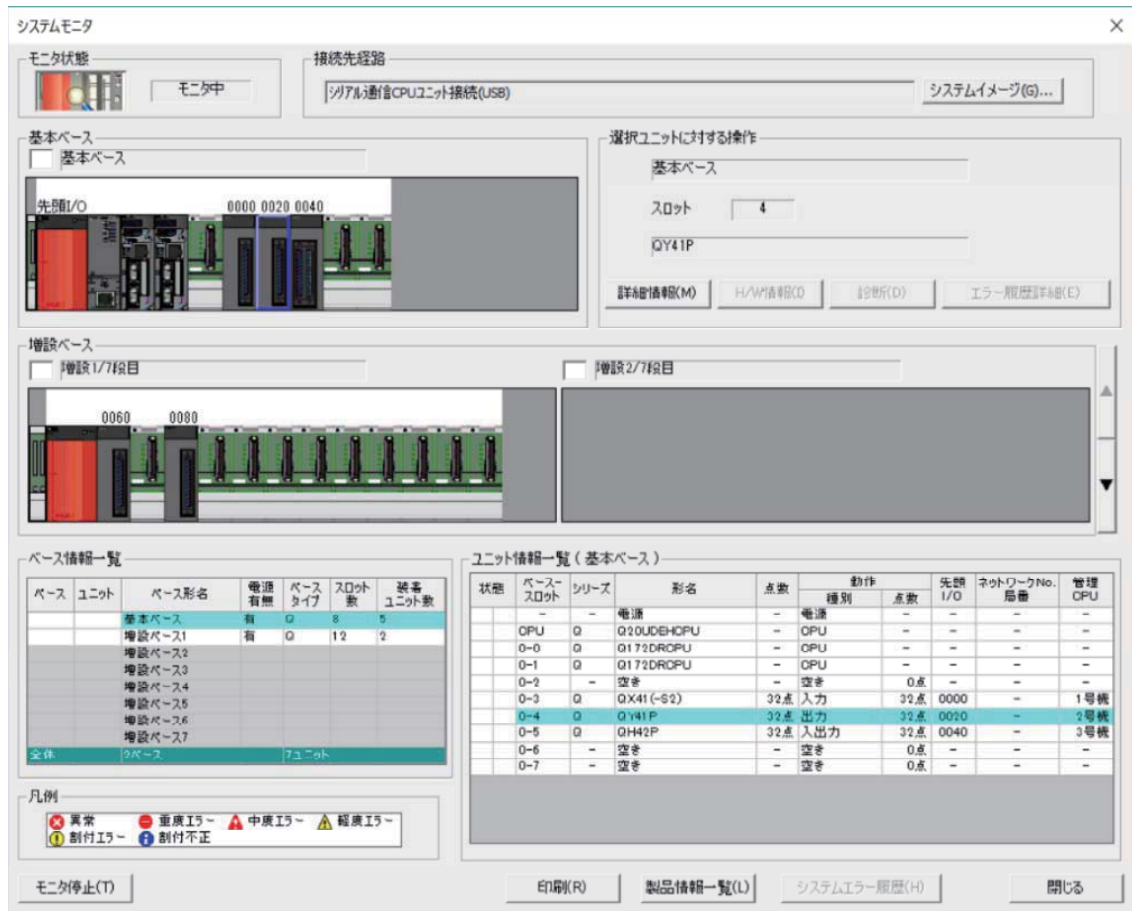


点击 [ 详细设定 ] 按钮，设定如下管理 CPU。

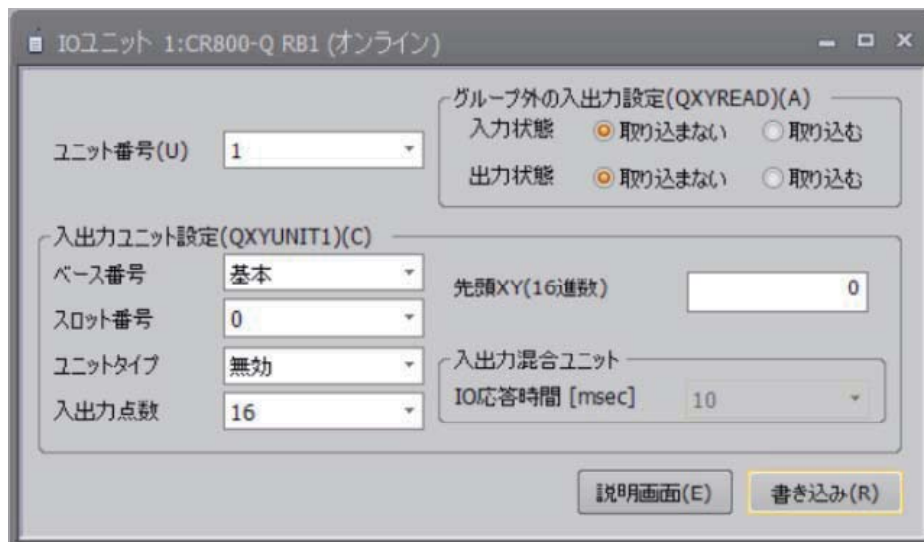


输入模块设定为可编程控制器管理，输出模块和输入输出混合模块设定为机器人 CPU 的 1 号机和 2 号机管理。

相同地，设定 RT ToolBox3 的参数，但暂且将可编程控制器的电源复位。  
可编程控制器启动后，从 GX Works2 的 [ 诊断 ] 菜单选择 [ 系统监视器 ]。

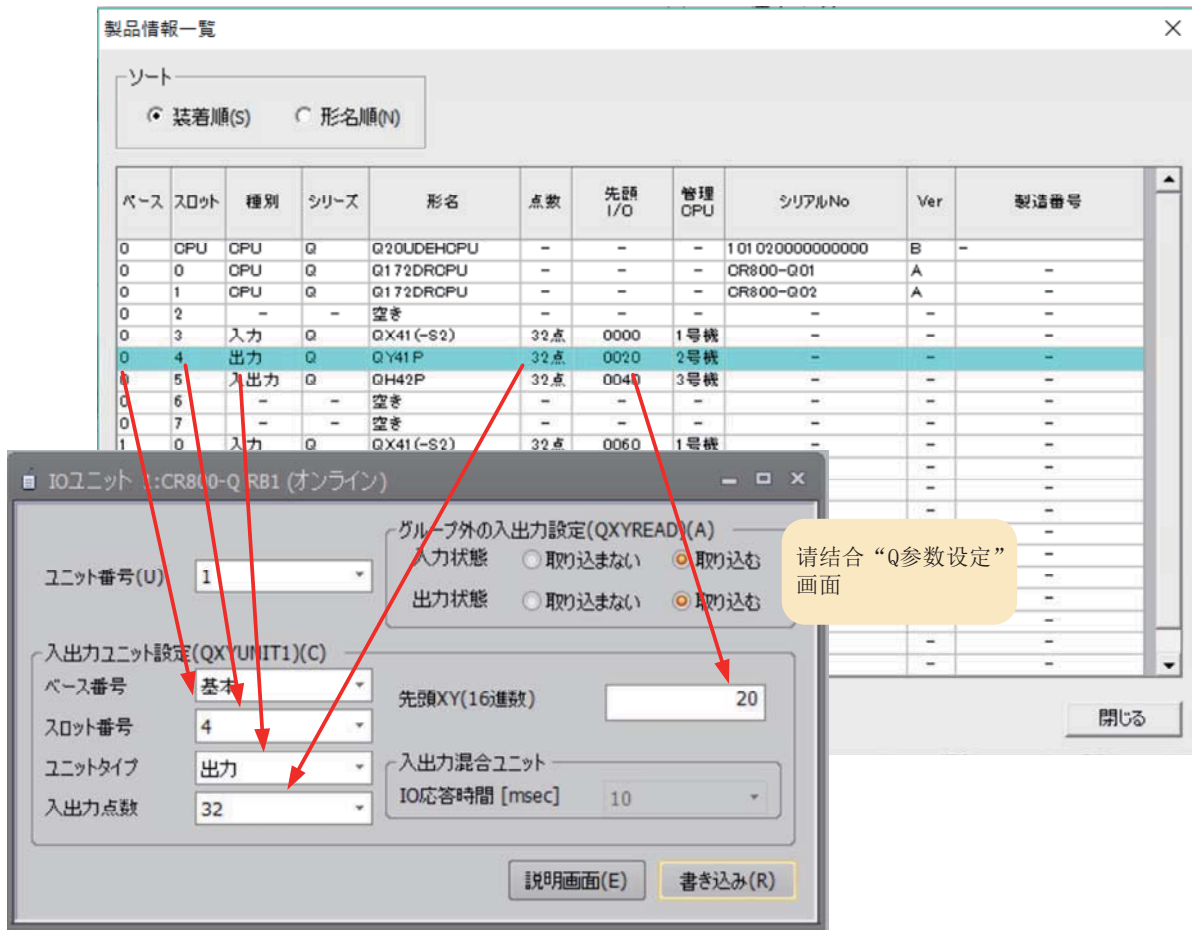


而且，关于机器人 CPU 管理的输入输出模块，在 RT ToolBox3 中设定参数。  
同时双击机器人 CPU1 号机、2 号机的“参数”菜单的“IO 模块”后，打开“IO 模块”参数画面。



首先，从机器人CPU1号机开始设定。

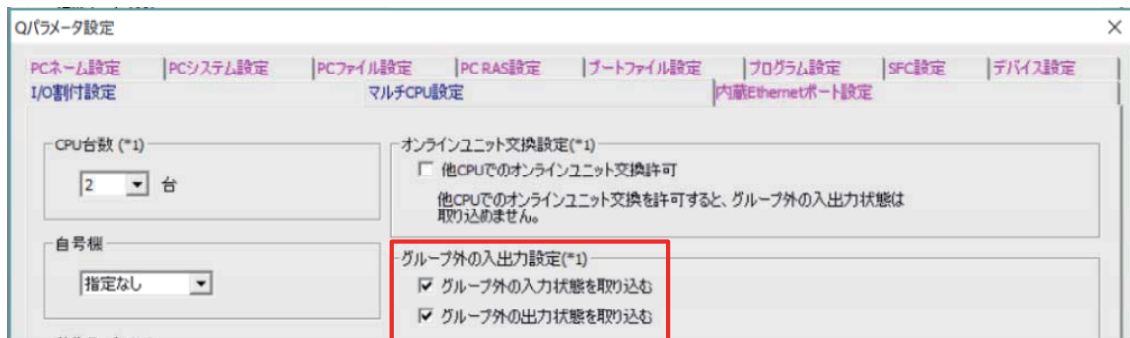
从GX Works2的“系统监视器”画面，点击对应的模块后，点击[产品信息一览]按钮，一边观察显示“产品信息一览”，一边将RT ToolBox3的“IO模块”参数进行如下设定。



通过1台机器人CPU，可以最多控制4台输入输出模块。

[模块编号]可设定为“1”~“4”，”表示其中4台中任意一台的设定。

关于“组外的输入输出设定（QXYREAD）”的设定，请结合以下所示GX Works2的“Q参数设定”画面的设定。





然后，对机器人 CPU2 号机进行设定。

从 GX Works2 的“系统监视器”画面，点击对应的模块后，点击 [产品信息一览] 按钮，一边观察显示“产品信息一览”，一边将 RT ToolBox3 的“IO 模块”参数进行如下设定。

製品情報一覧

ソート  
 装着順(S)    形名順(N)

| ベース | スロット | 種別  | シリーズ | 形名         | 点数  | 先頭 I/O | 管理 CPU | シリアルNo           | Ver | 製造番号 |
|-----|------|-----|------|------------|-----|--------|--------|------------------|-----|------|
| 0   | CPU  | CPU | Q    | Q20UDEHCPU | -   | -      | -      | 101 020000000000 | B   | -    |
| 0   | 0    | CPU | Q    | Q172DRCPU  | -   | -      | -      | CR800-Q01        | A   | -    |
| 0   | 1    | CPU | Q    | Q172DRCPU  | -   | -      | -      | CR800-Q02        | A   | -    |
| 0   | 2    | -   | -    | 空き         | -   | -      | -      | -                | -   | -    |
| 0   | 3    | 入力  | Q    | QX41(-S2)  | 32点 | 0000   | 1号機    | -                | -   | -    |
| 0   | 4    | 出力  | Q    | QY41P      | 32点 | 0020   | 2号機    | -                | -   | -    |
| 0   | 5    | 入出力 | Q    | QH42P      | 32点 | 0040   | 3号機    | -                | -   | -    |
| 0   | 6    | -   | -    | 空き         | -   | -      | -      | -                | -   | -    |

IOユニット 2: CR810-Q RB2 (オンライン)

ユニット番号(U) 1

グループ外の入出力設定(QXYREAD)(A)  
 入力状態  取り込まない  取り込む  
 出力状態  取り込まない  取り込む

入出力ユニット設定(QXYUNIT1)(C)  
 ベース番号 基本  
 スロット番号 5  
 ユニットタイプ 入出力混合  
 入出力点数 32

先頭XY(16進数) 40

入出力混合ユニット  
 IO応答時間 [msec] 10

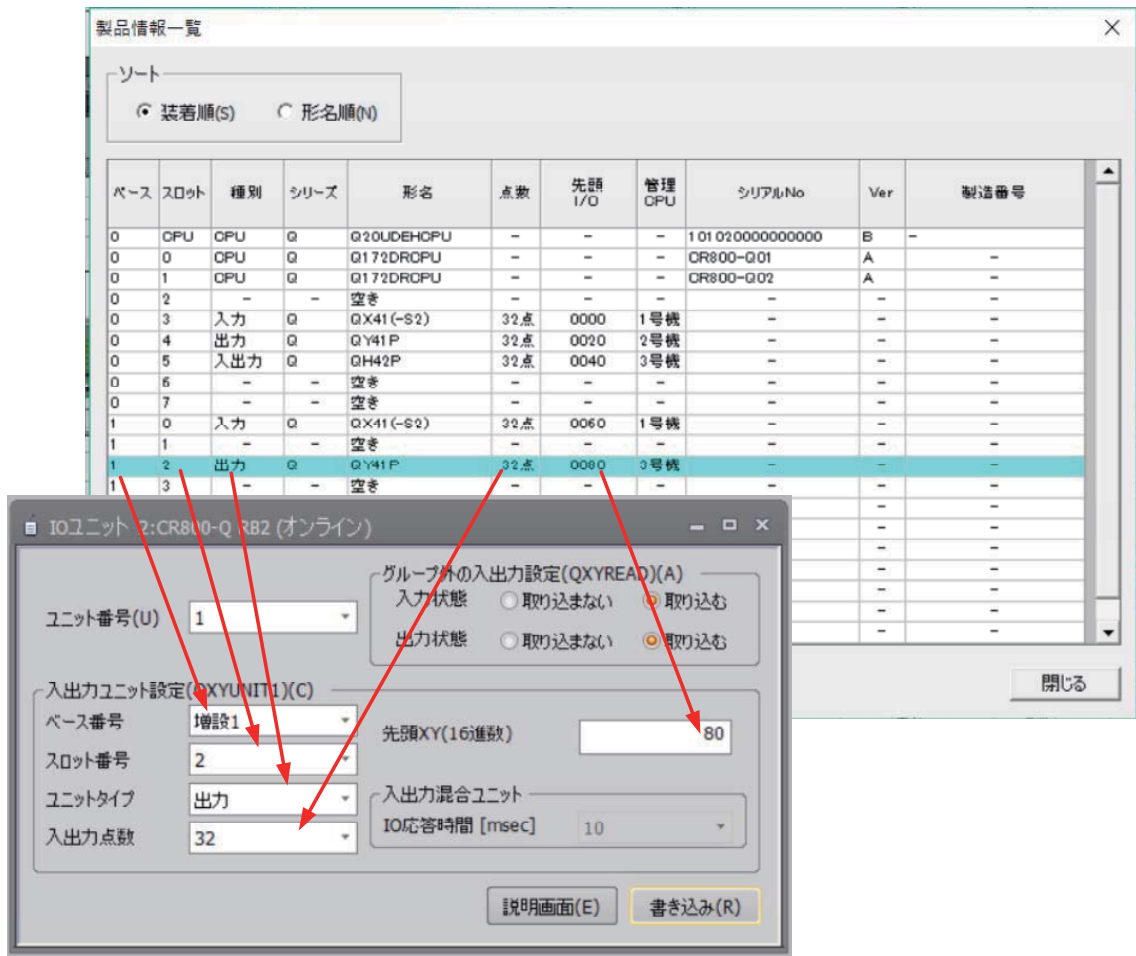
説明画面(E) 書き込み(R)

|   |        |       |       |     |  |      |     |
|---|--------|-------|-------|-----|--|------|-----|
| 5 | 4(0-4) | 出力    | QY41P | クリア |  |      | 2号機 |
| 6 | 5(0-5) | 入出力混合 | QH42P | クリア |  | 10ms | 3号機 |

关于“IO 响应时间”，请结合点击上述“Q 参数设定”的 [详细设定] 按钮后显示的画面的值。

接着实施第 2 台模块的设定。

从 GX Works2 的“系统监视器”画面，点击对应的模块后，点击 [产品信息一览] 按钮，一边观察显示“产品信息一览”，一边将 RT ToolBox3 的“IO 模块”参数进行如下设定。



设定以上参数后，将可编程控制器及机器人 CPU 的电源复位。

## (5) 关于机器人参数的详细内容

关于 RT ToolBox3 中设定的参数进行以下详细说明。

| 参数                                   | 参数名                 | 排列数<br>文字数 | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 出货时设定值              |
|--------------------------------------|---------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| PLC 输入输出信号的取入设定<br>(※ 仅限 CR800-Q 系列) | QXYREAD             | 整数 2       | 设定为读取其他号机CPU管理的输入输出模块(包含输入输出混合模块)的输入状态、及输出状态。<br>要素1: 输入信号(X)的获取[0: 无效/1: 有效]<br>要素2: 输出信号(Y)的获取[0: 无效/1: 有效]                                                                                                                                                                                                                                                                                                                                        | 0, 0                |
| PLC 输入输出单元设定<br>(※ 仅限 CR800-Q 系列)    | QXYUNITn<br>n=1 ~ 4 | 整数 7       | 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。<br>要素1: 单元类型<br>0: 无对象输入输出单元<br>1: 未使用(无意义)<br>2: 输出单元<br>3: 输入输出混合单元<br>要素2: 起始输入输出号码<br>0~4080(10进制数)<br>要素3: 基板号码<br>0: 主基板<br>1~7: 扩展基板<br>要素4: 插槽号码<br>0~11(10进制数)<br>要素5: 输入输出点数<br>0: 16点/1: 32点/2: 48点/3: 64点/4: 128点/<br>5: 256点/6: 512点/7: 1024点<br>要素6: 报警时输出模式(输出・输入输出混合单元用)<br>0: 清除<br>1: 保持<br>要素7: I/O响应时间1(输入・输入输出混合单元用)<br>0: 10ms/4: 1.0ms/5: 5.0ms/6: 20ms/7: 70ms<br>要素8: 未使用 | 0, 0, 0, 0, 0, 0, 0 |

在 RT ToolBox3 的以下画面, 设定上述参数。



(6) 使用状态变量进行控制

用于控制可编程控制器输入输出模块的状态变量存在以下类型。

| 变量名     | 内容                                                                                                                                 |
|---------|------------------------------------------------------------------------------------------------------------------------------------|
| 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)                                                     |
| 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                                     |
| 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                                     |
| 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                                  |
| M_YDev  | 以位为单位读取 / 写入 PLC 输出信号 (Y)。<br>例) 1 M_YDev (2)=1 ' 将可编程控制器输出信号 2 号设为 ON                                                             |
| 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                                      |
| 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                                    |
| 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 |

例如，系统构成示例 2 时，进行如下设定。



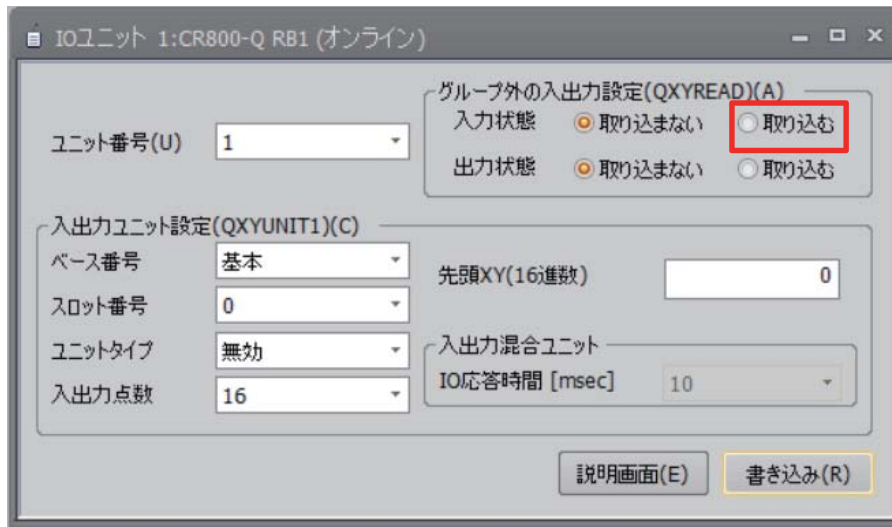
想显示安装在 1 号机 (= 可编程控制器) 管理的扩展基板 1 的插槽 0 中的输入模块的输入信息时，在 2 号机 (= 机器人 CPU1 号机) 中，准备如下机器人程序。

```
M1&=M_XDevD(&H60)
```

└─ 指定对应的模块的“起始I/O”编号。  
└─ 读取输入信号32bit的状态变量。

即使根据 3 号机 (= 机器人 CPU2 号机) 通过相同的程序可以观察可编程控制器管理的输入模块的信息。但是，需要将参数“QXYREAD”的第 1 要素设定为“1”。

即，在 RT ToolBox3 中，需要以下画面的设定。



接着，对输出至自己管理的输出模块、或输入输出模块的情况进行说明。



例如，将信号输出至安装在上图所示扩展基板 1 的插槽 2 中的输出模块时，在 3 号机（= 机器人 CPU2 号机）中，准备如下机器人程序。

```
M_YDevD(&H80) = &H1234
```

- └─ 指定想输出的数据。
- └─ 指定对应的模块的“起始I/O”编号。
- └─ 将数据进行32bit输出的状态变量。

### 5.26 在机器人CPU之间直接通信

在 CR800-R/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 CPU 缓冲存储器信息。

下面介绍大致的规格和步骤。

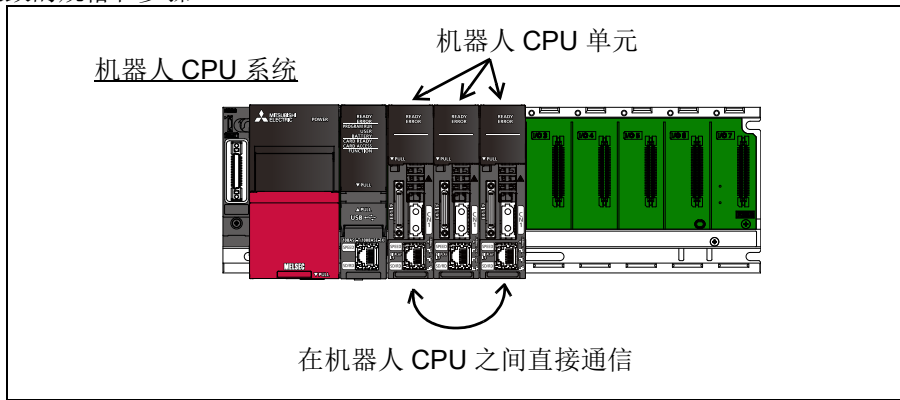


图 5-27：PLC 输入输出单元直接控制

(1) 规格

规格内容见表 5-44。

表 5-44：规格

| No. | 项目          | 规格                                                                                                                            |
|-----|-------------|-------------------------------------------------------------------------------------------------------------------------------|
| 1   | 指定软元件范围     | CPU 缓冲存储器访问软元件中可指定的范围。                                                                                                        |
| 2   | 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，将通过可编程控制器（1号机）实施。不能将专用输入输出参数分配到2号机以后的CPU缓冲存储器访问软元件。将专用输入输出信号分配到CPU缓冲存储器访问软元件以使用时，请务必分配可编程控制器（1号机）的CPU缓冲存储器以使用。 |

(2) 机器人状态变量的说明

对表 5-45 中相关的机器人状态变量的概要进行说明。详细内容请参照第 333 页的“4.13.2 各机器人状态变量（系统状态变量）的说明”的各说明页。

表 5-45：相关的机器人状态变量（CR800-R 系列）

| 变量名                         | 内容                      | 说明页     |
|-----------------------------|-------------------------|---------|
| M_GDev/ M_GDevW/ M_GDevD    | 将数值写入至 CPU 缓冲存储器或进行参照   | 第 361 页 |
| M_HGDev/ M_HGDevW/ M_HGDevD | 将数值写入至固定周期通信区域或进行参照     | 第 365 页 |
| P_GDev                      | 将位置数据写入至 CPU 缓冲存储器或进行参照 | 第 416 页 |
| P_HGDev                     | 将位置数据写入至固定周期通信区域写入或进行参照 | 第 418 页 |

表 5-46：相关的机器人状态变量（CR800-Q 系列）

| 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 □）。      | 第 395 页 |
| M_UDevD | 以双字为单位读取 / 写入多 CPU 间共有设备（U3En\G □）。     | 第 395 页 |
| P_UDev  | 将多 CPU 间共有设备（U3En\G □）以位置数据类型进行读取 / 写入。 | 第 421 页 |

## 5.27 关于冗余系统错误的动作选择参数

关于表 5-47 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] 按键时的动作。

表 5-47：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 | 模式选择器开关接线异常 |
| H0051 | 外部紧急停止配线异常  |

[RESET] 按键输入时的动作为选择表 5-48 中的任意一个。

表 5-48：参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。

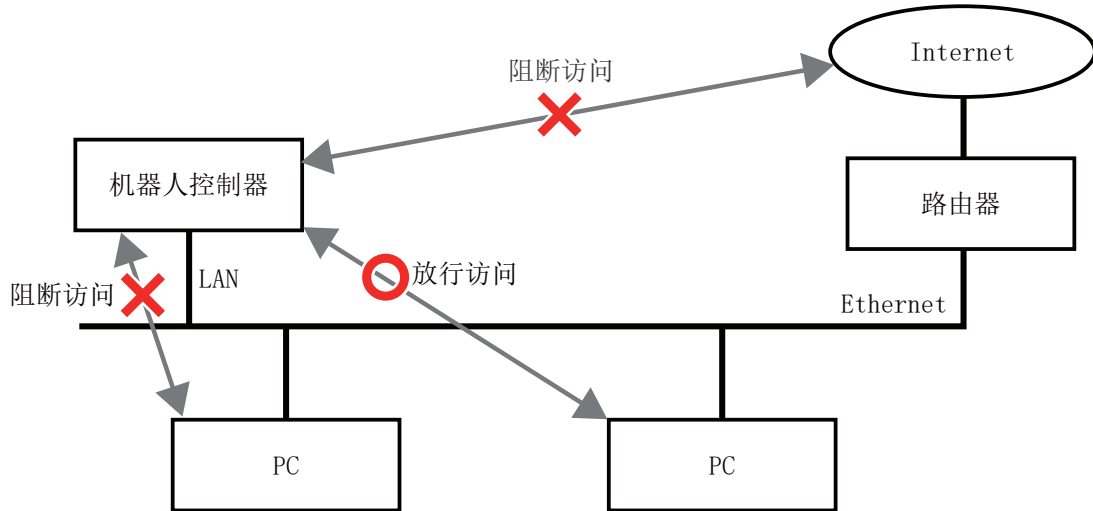
### 5.28 关于安全功能

#### (1) IP 地址过滤功能

本功能是在机器人控制器连接了以太网的情况下，对访问源的 IP 地址进行识别以防止来自计算机等外部设备的非法访问的功能。

通过设定放行或阻断的 IP 地址，可以限制来自外部设备的非法访问。

机器人控制器的软件版本为 C2 以上时该功能有效。



限制访问的方法有两种，分别是放行和阻断。通过表 5-49 所示的参数使功能有效，并设定对象 IP 地址的范围。关于参数的详细内容，请参照第 497 页的“5.5 通信参数”。

- ①放行... 对允许通信的访问源的 IP 地址范围进行了设定后，将允许来自所设定的 IP 地址范围内的访问。对来自设定范围外的 IP 地址的访问将会禁止。
- ②阻断... 对禁止通信的访问源的 IP 地址范围进行了设定后，将禁止来自所设定的 IP 地址范围内的访问。对来自设定范围外的 IP 地址的访问将会允许。



#### 注意

虽然配备了 IP 地址过滤功能，但是为了防止经由网络的外部设备的非法访问以保障机器人系统的安全时，应采取设置防火墙等相关对策。

表 5-49: IP 地址过滤功能的相关参数

| 参数名称     | 内容                                                                                                                                                                                                                                                                                                                              |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NETIPFLT | 对 IP 地址过滤功能的有效 / 无效进行指定。<br>有效时，可以任意设定为放行或阻断。对象 IP 地址可以通过参数 NETIPFLS 和 NETIPFLE 进行指定。<br>放行：允许来自指定的 IP 地址的访问。<br>阻断：阻断来自指定的 IP 地址的访问。<br><br>0: 无效（不使用 IP 地址过滤功能。允许所有访问。出厂时的设定值。）<br>1: 有效 / 放行（允许来自指定的 IP 地址的访问。）<br>2: 有效 / 阻断（禁止来自指定的 IP 地址的访问。）<br><br>将本功能设为了有效时，应使起始地址与结束地址为同一网络地址，且地址大小不可逆转。设定错误的情况下，接通电源时将发生错误且本功能将为无效。 |
| NETIPFLS | 对 IP 地址过滤功能的起始地址进行指定。<br>在 IP 地址过滤功能的有效 / 无效指定的参数 NETIPFLT 为有效（1 或 2）时可以使用。                                                                                                                                                                                                                                                     |
| NETIPFLE | 对 IP 地址过滤功能的结束地址进行指定。<br>在 IP 地址过滤功能的有效 / 无效指定的参数 NETIPFLT 为有效（1 或 2）时可以使用。                                                                                                                                                                                                                                                     |



## 6 外部输入输出的功能

### 6.1 种类

- (1) 专用输入输出..... 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 605 页的“6.2PLC 连接输入输出功能”)及、紧急停止输入 (参照第 644 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) 通用输入输出..... 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) 抓手输入输出..... 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。(抓手输入输出为选配。)
- (4) 软元件 ..... 存储与可编程控制器等同样以 1 位为单位的信息的位软元件 (X、Y 等) 及存储以 1 字为单位的信息的字软元件 (D 等)。用于与 GOT 及 SLMP 对应机器等的通信。在机器人信息的公开及从外部机器对机器人进行指示时使用。

表 6-1: 输入输出信号全体图

|            | 输入输出信号                                                                                               | 使用方法                                                                                                                                                           |
|------------|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 抓手输入输出     | CR800-D/R/Q:<br>(输入) 900 ~ 907<br>(输出) 900 ~ 907<br>CR860-D/R/Q:<br>(输入) 764 ~ 775<br>(输出) 764 ~ 771 | 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 连接输入输出 | 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。 |
| SKIP 输入    | 800 ~ 803                                                                                            | 800 为停止输入专用<br>801 ~ 803 通过 M_In 变量进行参照                                                                                                                        |

## 6.2 PLC连接输入输出功能

本功能只在 CR800-R/Q 系列中有效。RnCPU/QnUD(H)CPU（以下统称为可编程控制器 CPU）及 R16RTCPU/Q172DSRCPU（以下统称为机器人 CPU）是使用 CPU 间的 CPU 缓冲存储器，依据可编程控制器梯形图程序进行通信。通信时使用 CPU 缓冲存储器的“CPU 缓冲存储器固定周期通信区域”。机器人 CPU 的输入信号、输出信号同时使用 10000 ~ 18191 之间的信号号码。

### 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 Works 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

##### ■ CR800-R 系列

使用 GX-Works3 执行多 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。执行多 CPU 设定前，实施模块构成的设定。按照实际的模块构成，对下图的模块进行配置。

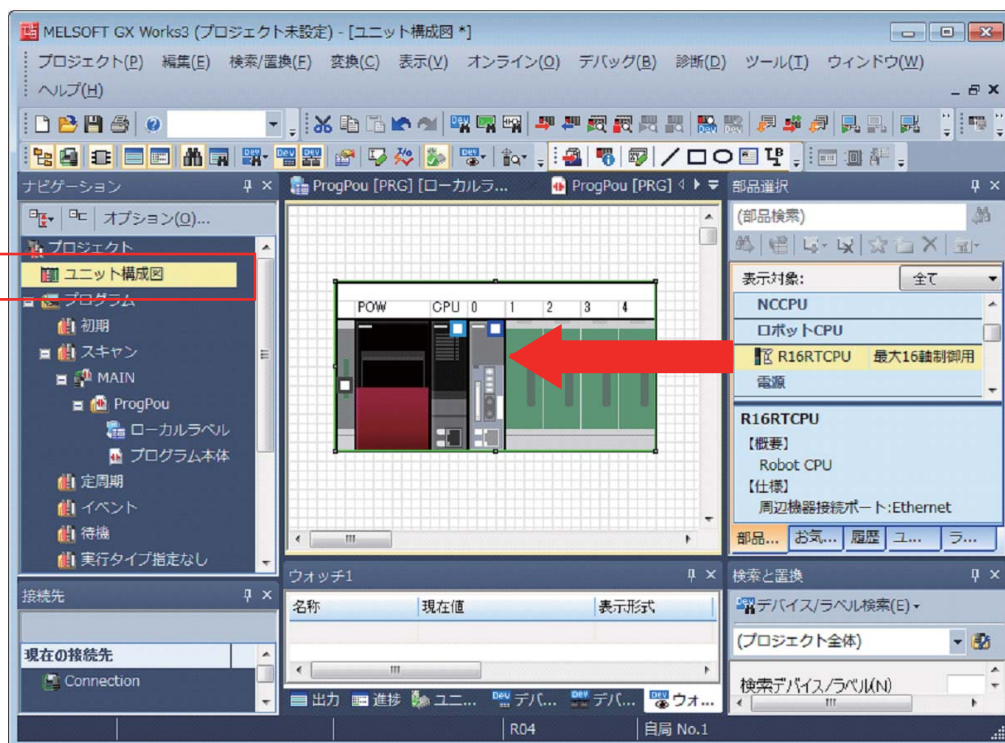
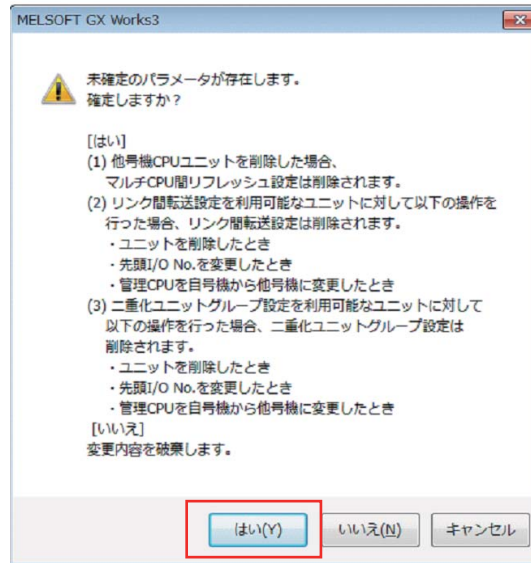


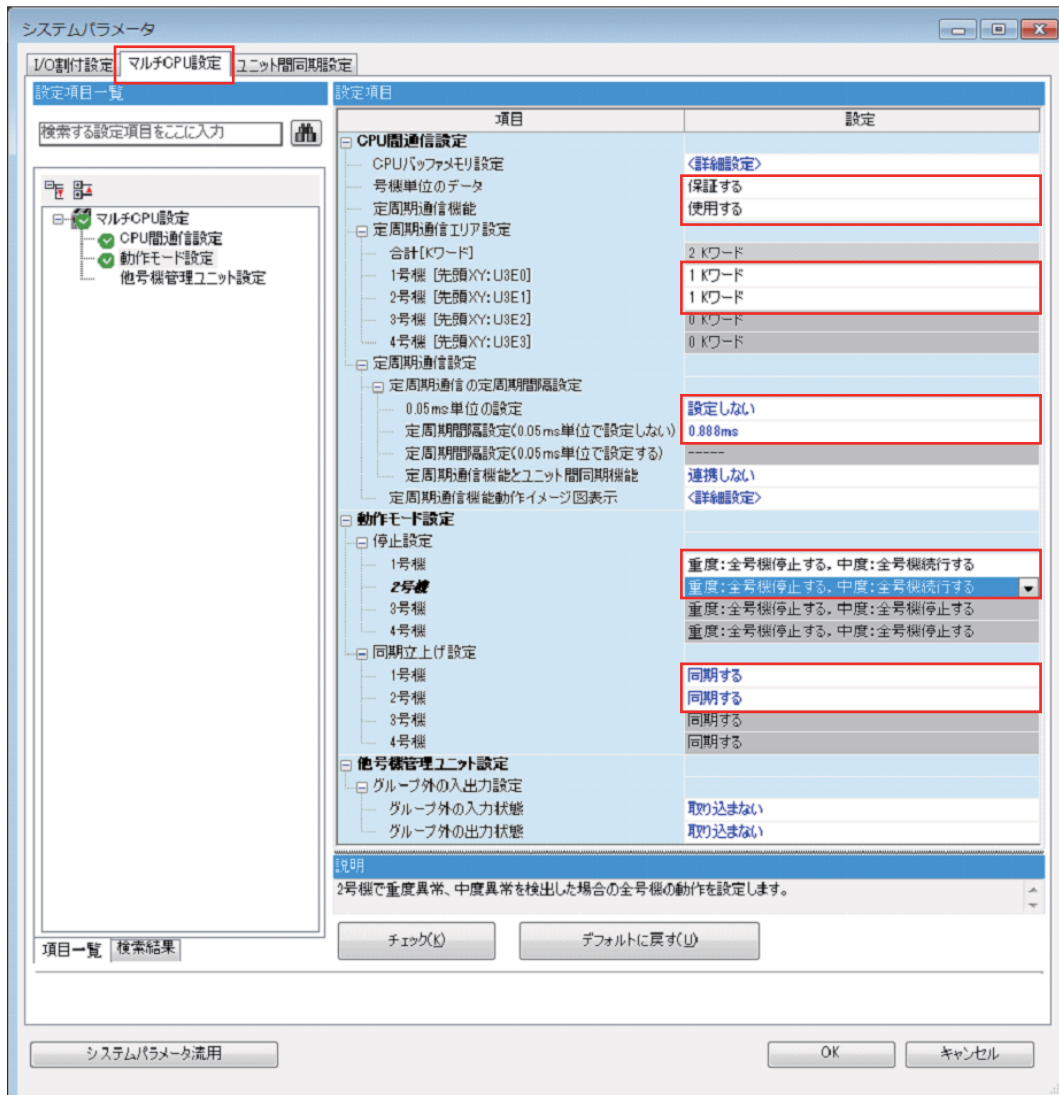
图 6-1: GX Works3 模块构成图画面（机器人 1 台）

对模块进行配置后，关闭模块构成画面，即显示下图的对话框，选择“是”。



接着，打开系统参数的画面，实施多CPU设定。

以K字为单位设定点数，关于机器人CPU，由于仅能在不足1K字时使用，因此应进行1K字的设定。



| 设定项目      |                              | 说明                                                                                                                                                                                                                    | 设定值                                                                                                                                                                                                                                                                                            |
|-----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU 间通信设定 | 号机单位的数据                      | 在通过 CPU 模块间刷新的数据通信中，防止各号机的数据分离并进行数据的发送接收时选择。处理 64 位数据以上的数据时可能发生数据的分离。                                                                                                                                                 | 保证<br>不保证<br><br>机器人 CPU 的可编程控制器链接输入输出功能的更新周期如下所示。<br><号机单位的数据 - “不保证”> 指定的固定周期通信间隔。<br><号机单位的数据 - “保证”> 即使将固定周期通信间隔设定设为“0.444ms”以下，也会变为 0.888ms。                                                                                                                                              |
|           | 固定周期通信功能                     | 设定是否使用固定周期通信功能。与机器人 CPU 组合时务必选择“使用”。                                                                                                                                                                                  | 使用                                                                                                                                                                                                                                                                                             |
|           | 固定周期通信区域设定                   | 设定固定周期通信区域 <sup>注 1)</sup> 内各号机的传送区域范围。<br>机器人所需区域如下所示。<br><CPU 缓冲存储器扩展功能 - 有效时><br>• 机器人输入区域 . . . . 1.0K<br>• 机器人输出区域 . . . . 1.0K<br><br><CPU 缓冲存储器扩展功能 - 无效时><br>• 机器人输入区域 . . . . 0.5K<br>• 机器人输出区域 . . . . 0.5K | <CPU 缓冲存储器扩展功能 - 有效><br>• 1号机 . . . . 向机器人发送数据的大小(1K) 与向其他号机发送数据的大小的合计<br>• 机器人号机 . . . . 设定 1K<br>• 其他的号机 . . . . 设定自号机 CPU 的发送数据大小<br><br><CPU 缓冲存储器扩展功能 - 无效><br>• 1号机 . . . . 向机器人发送数据的大小 (0.5K) 与向其他号机发送数据的大小的合计<br>• 机器人号机 . . . . 设定 1K <sup>注 2)</sup><br>• 其他的号机 . . . . 设定自号机的发送数据大小 |
|           | 0.05ms 单位的设定                 | 设定是否以 0.05ms 为单位设定固定周期通信周期。与机器人 CPU 组合时请务必选择“未设定”。                                                                                                                                                                    | 未设定                                                                                                                                                                                                                                                                                            |
|           | 固定周期通信间隔设定 (未以 0.05ms 为单位设定) | 从选择项目设定固定周期通信周期。请将仅使用固定周期通信功能的号机设定为同一固定周期通信周期。此外，使用机器人 CPU 的协调控制功能及干涉回避功能时，与有无号机单位的数据保证无关，请选择“0.888ms”以下。                                                                                                             | 0.222ms<br>0.444ms<br>0.888ms<br>1.777ms<br>3.555ms<br>7.111ms                                                                                                                                                                                                                                 |
| 动作模式设定    | 停止设定                         | 各号机中发生重度异常或中度异常时，对全部号机是否停止动作进行设定。                                                                                                                                                                                     | 重度：全部号机停止、中度：全部号机继续<br>※应在全部 CPU 模块中设定。                                                                                                                                                                                                                                                        |
|           | 同步启动设定                       | 在多 CPU 系统中使 CPU 模块的启动时间同步时进行设定。<br>※由于机器人 CPU 启动要花 60 秒，因此应设定为同步启动。                                                                                                                                                   | 同步                                                                                                                                                                                                                                                                                             |

注 1) 关于多 CPU 及固定周期通信区域，请参照 RCPU 的手册 (MELSEC iQ-R CPU 模块用户手册 (应用篇))。

注 2) 由于区域只能以 1K 为单位进行设定，因此 0.5K 时分配为 1K。

### ■ CR800-Q 系列

使用 GX Works 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。

Q/パラメータ設定

PC名前設定 | PCシステム設定 | PCファイル設定 | PC RAS設定 | プートファイル設定 | プログラム設定 | SFC設定 | デバイス設定

I/O割付設定 | マルチCPU設定 | 内蔵Ethernetポート設定

CPU台数 (\*1)  台

オンラインユニット交換設定 (\*1)  
 他CPUでのオンラインユニット交換許可  
 他CPUでのオンラインユニット交換を許可すると、グループ外の入出力状態は取り込めません。

自号機

グループ外の入出力設定 (\*1)  
 グループ外の入力状態を取り込む  
 グループ外の入力状態を取り込む

動作モード (\*1)  
 CPU停止エラー時の動作モード  
 1号機のエラーで全号機停止  
 2号機のエラーで全号機停止  
 3号機のエラーで全号機停止  
 4号機のエラーで全号機停止

マルチCPU間同期上げ (\*1)  
 対象CPU  
 1号機  
 2号機  
 3号機  
 4号機

マルチCPU間高速通信エリア設定 | 通信エリア設定 (リフレッシュ設定)

マルチCPU間高速通信機能を使用する

| CPU | 各CPU送信範囲 (*1) |         |      |        |        |   | 自動リフレッシュ設定 |
|-----|---------------|---------|------|--------|--------|---|------------|
|     | 点数(K)         | I/O No. | 先頭   | 最終     | 点数     |   |            |
| 1号機 | 1             | U3E0    | 1024 | G10000 | G11023 | 0 | リフレッシュ設定   |
| 2号機 | 1             | U3E1    | 1024 | G10000 | G11023 | 0 | リフレッシュ設定   |
| 3号機 |               |         |      |        |        |   |            |
| 4号機 |               |         |      |        |        |   |            |

合計  点  
 リフレッシュ設定は必要に応じ設定 (未設定 / 設定済み)  
 高度な設定を行なう (\*1)

合計は14K点までです。

(\*1) マルチCPU時、同一設定にしてください。

图 6-2: PLC CPU: 依据 GX Works 设定画面范例

在多重 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（Q172DSRCPU）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。

(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

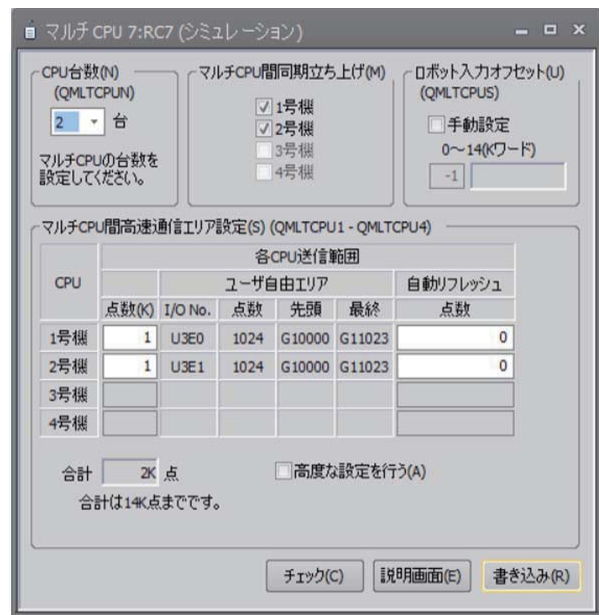
| 参数名                 | 内容                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 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 | <p><b>【CR800-R 系列】</b><br/>CPU 缓冲存储器固定周期通信区域设定 (n = 1 ~ 4)<br/>多 CPU 系统中，从 1 号机读取通过 1 ~ 4 号机的 CPU 缓冲存储器固定周期通信区域进行发送接收的点数，并自动设定。无需更改值。</p> <p>要素 1：固定周期通信区域的大小 (K 字)<br/>范围：0 ~ 12<br/>※ 全号机合计最多 24K 字</p> <p>要素 2：自动刷新的点数 (字)<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。</p> <p>要素 3：系统备用</p> <p>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。</p> <p><b>【CR800-Q 系列】</b><br/>多重 CPU<sub>n</sub> 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。</p> <p>要素 1：使用者自由区域的大小 (K 点)<br/>范围：1 ~ 14 (最大)</p> <p>表 6-3: CPU 台数别的设定范围</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/>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。</p> <p>要素 3：系统区域的大小 (K 点)<br/>范围：1 或 2</p> <p>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。</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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |         |      |   |           |   |           |   |           |            |

| 参数名        | 内容                                                                                                                                                                                    | 出货时的设定值          |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| IQMEM 注1)  | 选择 CPU 缓冲存储器扩展功能。<br>各个位分别分配有功能。1/0= 有效 / 无效<br>15 0<br>00000000 00000000<br>bit2-3、    ...bit0: 扩展功能的使用<br>5-15    ...bit1: PLC 直接执行功能<br>未使用   .....bit4: 干涉回避功能                    | 0000000000000000 |
| IQSPEC 注1) | 设定 CR800-R/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>15 0<br>00000000 00000000<br>bit1-15  ...bit0: CPU 缓冲存储器写入方向<br>未使用 =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 | 0000000000000001 |

注1) 关于扩展功能的详细内容, 请参照另一手册“使用说明书 / CR800-R / CR800-Q 系列控制器 iQ Platform 对应 扩展功能说明书 (BFP-A3580)”。



CR800-R系列



CR800-Q系列

图 6-3: 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、基座。(截止到 2018 年 3 月)

| 项目         | 型号                                                                                                                                             |                                                                                                                                               | 备注                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
|            | CR800-R 系列                                                                                                                                     | CR800-Q 系列                                                                                                                                    |                         |
| 可编程控制器 CPU | R00CPU、R01CPU、<br>R02CPU、R04CPU、<br>R08CPU、R16CPU、<br>R32CPU、R120CPU<br>安全 CPU 注 1)<br>R08SFCPU-SET、R16SFCPU-SET<br>R32SFCPU-SET、R120SFCPU-SET | 通用模型 QCPU<br>Q03UD(E)CPU、Q04UD(E)HCPU、<br>Q06UD(E)HCPU、Q10UD(E)HCPU、<br>Q13UD(E)HCPU、Q20UD(E)HCPU、<br>Q26UD(E)HCPU、Q50UDEHCPU、<br>Q100UDEHCPU | • 1 号机必须是可编程控<br>制器 CPU |
| 机器人 CPU    | R16RTCPU                                                                                                                                       | Q172DSRCPU                                                                                                                                    |                         |
| 运动 CPU     | R16MTCPUV、R32MTCPU、<br>R64MTCPU                                                                                                                | Q172DCPU、Q173DCPU、<br>Q173DSCPU                                                                                                               |                         |
| NCCPU      | R16NCCPU                                                                                                                                       | Q172NCCPU                                                                                                                                     |                         |
| 基座         | 多 CPU 间高速主基板<br>R35B、R38B、R312B                                                                                                                | 多 CPU 间高速主基板<br>Q38DB、Q312DB                                                                                                                  | • 支持多 CPU 间高速通<br>信的产品  |

注 1) 支持的版本

- 机器人控制器: Ver. A5n 以上
- 安全 CPU: Ver. 20 以上

机器人 CPU 使用 RT ToolBox 或示教单元 (R32TB、R56TB)、可编程控制器 CPU 使用 GXWorks 或 GX Developer、运动控制器 CPU 用 MT Works 或 MT Developer、NC CPU 使用 Remote Monitor Tool 进行参数设定。关于详细内容, 请参照各工具的使用说明书。



### 6.2.2 CPU缓冲存储器与机器人输入输出信号的对应

可编程控制器 CPU 中，将 CPU 缓冲存储器存取为 U3E0\HG511（CR800-Q 系列时为 U3E0\G10511）。机器人 CPU No. n 的 CPU 缓冲存储器则存取为 U3En\HG511（CR800-Q 系列时为 U3En\G10511）。

（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 缓冲存储器与各机器人输入输出信号的对应如下表所示，无法更改。

表 6-4: CPU 缓冲存储器与机器人输入输出信号的对应

| PLC（Word 设备） |                           |                           | 机器人（Bit 设备） |                               |
|--------------|---------------------------|---------------------------|-------------|-------------------------------|
| 控制<br>器      | CR800-R                   | CR800-Q                   |             |                               |
| 输出           | U3E0\HG0 ~ U3E0\HG511     | U3E0\G10000 ~ U3E0\G10511 | 输入          | 机器人 CPU No. 1 / 10000 ~ 18191 |
|              | U3E0\HG512 ~ U3E0\HG1023  | U3E0\G10512 ~ U3E0\G11023 |             | 机器人 CPU No. 2 / 10000 ~ 18191 |
|              | U3E0\HG1024 ~ U3E0\HG1535 | U3E0\G11024 ~ U3E0\G11535 |             | 机器人 CPU No. 3 / 10000 ~ 18191 |
| 输入           | U3E1\HG0 ~ U3E1\HG511     | U3E1\G10000 ~ U3E1\G10511 | 输出          | 机器人 CPU No. 1 / 10000 ~ 18191 |
|              | U3E2\HG0 ~ U3E2\HG511     | U3E2\G10000 ~ U3E2\G10511 |             | 机器人 CPU No. 2 / 10000 ~ 18191 |
|              | U3E3\HG0 ~ U3E3\HG511     | 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「机器人操作权有效按钮」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 CPU 构成如下所示。

•CR800-R 系列

1 号机：可编程控制器 RnCPU、2 号机：机器人 R16RTCPU

•CR800-Q 系列

1 号机：可编程控制器 QnUD(H)CPU、2 号机：机器人 Q172DSRCPU

**【说明】**

以下说明为 CR800-R 系列时。CR800-Q 系列时，请参考表 6-4，改读为设备编号。

< 第 0 ~ 16 行 >

将 M100 ~ M131 写入 U3E0\HG0、U3E0\HG1 的共有设备内存，作为从可编程控制器向机器人的输入。将 U3E1\HG0、U3E1\HG1 的共有设备内存读取至 M200 ~ M231 的位设备，作为从机器人向可编程控制器的输出。

< 第 17 ~ 22 行 >

开启 X0 后即开启 M105，且 M105 对应的可编程控制器的 U3E0\HG0 的位 5 开启。于是，机器人的输入 10005 开启，通过专用输入信号分配的操作权变成有效。

操作权变成有效，则通过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\HG0 的位 5 会开启。于是，U3E1\HG0 的位 5 对应的可编程控制器的 M205 开启，且 Y20 开启。

而且，在此例中位设备 M201（U3E0\HG0 的位 1/ 即机器人的输出 10001）表示控制器电源开启完成（输出外部输入信号为可接收）。

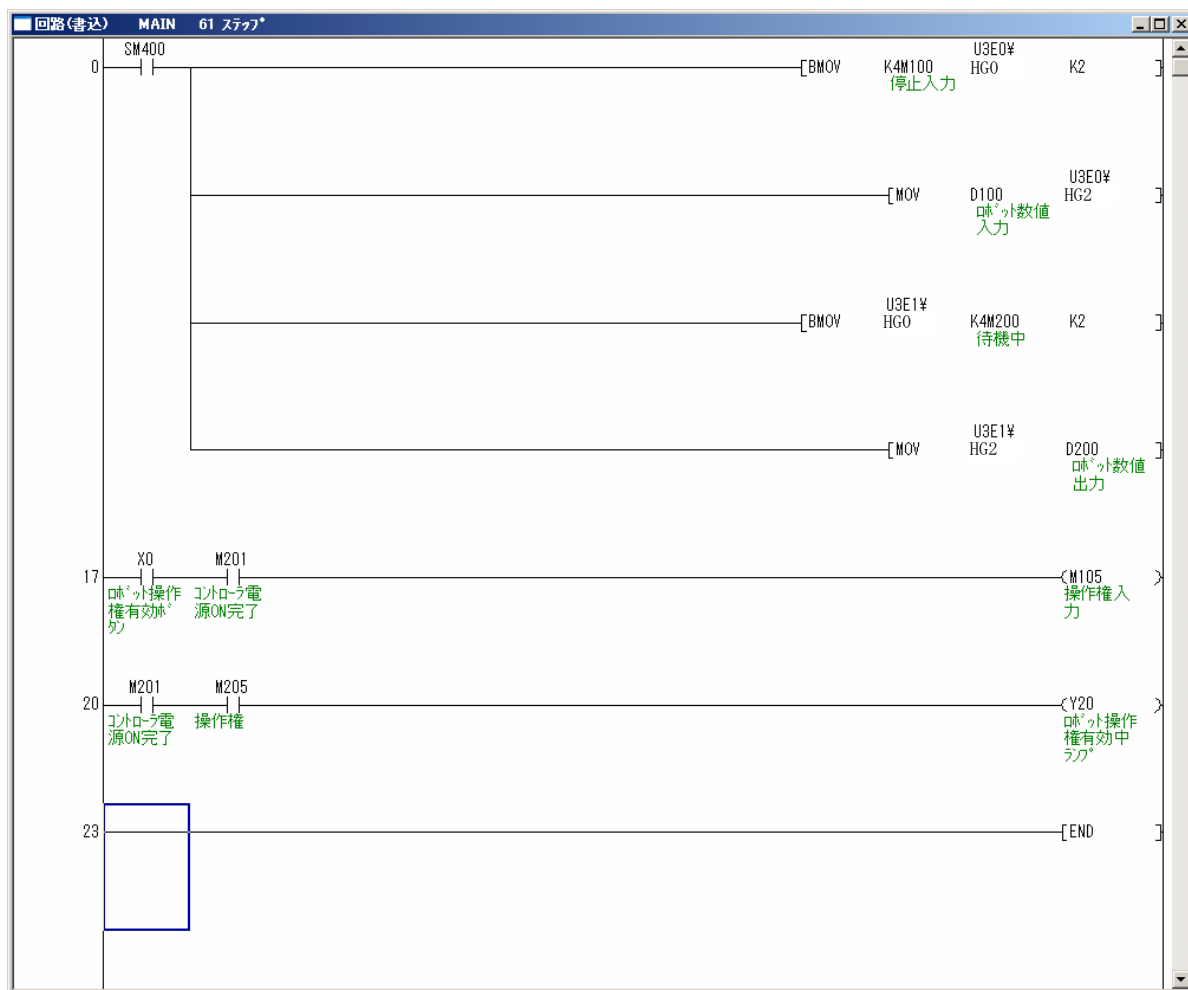


图 6-4: PLC 阶梯图例

## 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入输出信号的分配如表 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    | 设备 <sup>注1)</sup> |            |
|----------|-----------------|---------------|-------|-------|-------------------|------------|
|          |                 |               |       |       | CR800-R 系列        | CR800-Q 系列 |
| STOP     | 停止输入（不可变更分配）    | 中断中输出（不可变更分配） | 10000 | 10000 | HG0               | 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 |                   |            |
| ATOPMD   | -               | 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 | HG1               | G10001     |
| 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  | 倍率修调选择输入（※）     | -             | 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 |                   |            |
| ERRROUT  | 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 | HG2               | 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    | 设备 <sup>注1)</sup> |           |
|----------|-----------------|--------------|-------|-------|-------------------|-----------|
|          |                 |              |       |       | CR800-R系列         | CR800-Q系列 |
| IODATA   | 数值输入 4          | 数值输出 4       | 10036 | 10036 | HG2               | 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 |                   |           |
| HNDCTRL1 | -               | 抓手输出信号状态 900 | -     | 10048 | HG3               | 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 |                   |           |
|          | -               | 抓手输出信号状态 907 | -     | 10055 |                   |           |
| HNDSTS1  | -               | 抓手输入信号状态 900 | -     | 10056 | HG3               | 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 | HG4               | 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 专用输入输出

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入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 93 页的“3.15 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。

输入输出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 636 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：专用输入输出一览表

| 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 |               |
|------------------|----|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------|---------------|
|                  |    |             |                                                                                                                                                                                        |          | CR800-R/Q<br>系列   | CR800-D<br>系列 |
| 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      | 输出控制器模式为“AUTOMATIC”。为远程模式。<br>以 IO 控制的时候，此信号开启状态会变成条件。                                                                                                                                 |          | 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      | 输出控制器模式为“AUTOMATIC”。                                                                                                                                                                   |          | 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,            | 5,            |
|                  | 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。                                                                                                           |          | 10005             | 3             |
| 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 |               |
|---------------------|----|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------|---------------|
|                     |    |              |                                                                                                                                                                          |          | CR800-R/Q<br>系列    | CR800-D<br>系列 |
| 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。)<br>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。<br>但是, 以参数「S L T * *」起动条件为 ALWAYS 和 ERROR 设定的插槽除外。<br>依据参数 1ND, 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「S L T * *」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「S L T * *」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「S L T * *」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 |               |
|---------------------|----|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------|---------------|
|                     |    |                    |                                                                                                                                                                                                                              |          | CR800-R/Q<br>系列    | CR800-D<br>系列 |
| 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,<br>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。<br>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,             | -1,           |
|                     | 输出 | 循环停止动作中输出<br>信号    | 输出循环停止中动作中。<br>循环停止完成的话, 会关闭。                                                                                                                                                                                                |          | 10012              | -1            |
| 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<br>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<br>启 (ON)。<br>机器 LOCK 状态时, 机器人不会动作, 程<br>序运行会变可能。                                                                                                                                         |          | -1                 | -1            |
| SAFEPOS<br>(需要操作权)  | 输入 | 退避点回归输入信号          | 要求退避点回归动作。往参数「JSAFE」<br>的位置做关节插补动作。速度依存在速度<br>比例。请注意与周边装置的干涉。                                                                                                                                                                | Edge     | 10013,             | -1,           |
|                     | 输出 | 退避点回归中输出信<br>号     | 输出退避点回归动作中。                                                                                                                                                                                                                  |          | 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<br>后, 重新输入控制器的电源的话, 输出为<br>OFF。使下列条件也会输出。<br>• 控制器电源 OFF 累计时间超过 14600。<br>将电池消耗时间初始化的话, 输出为<br>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 ORSTO ~ ORS18160, 设定输入时<br>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<br>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<br>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<br>(冗余系统线路异常)<br>• 操作面板紧急停止错误: H0060、H0061<br>(冗余系统线路异常)<br>• 示教单元紧急停止错误: H0070、H0071<br>(冗余系统线路异常)<br>• 门开关错误: H0039、H0040<br>(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                        |                                |
|----------------------------------|----|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------|--------------------------------|
|                                  |    |                        |                                                                                                                                                                                                          |          | CR800-R/Q<br>系列                           | CR800-D<br>系列                  |
| 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, 输出程序中断中。n=1 ~ 32                                                                                                                                                                             |          |                                           |                                |
| MnSRVOFF<br>(n=1 ~ 3)            | 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)                                                                                                                                                                                          |          |                                           |                                |
| MnSRVON<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                                                                                                                                                                                         |          |                                           |                                |
| MnMELOCK<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                                                                                                                                                                                       |          |                                           |                                |
| BRKLOCK                          | 输入 | 制动闸锁定输入                | 伺服开启中锁定制动闸时, 开启本输入信号。全部制动闸轴为对象。                                                                                                                                                                          | LEVEL    | -1,<br>-1                                 | -1,<br>-1                      |
|                                  | 输出 | 制动闸锁定中输出               | 制动闸锁定输入开启时, 本输出信号变为开启。                                                                                                                                                                                   |          |                                           |                                |
| 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 15ms 以上请开始输出。此外本信号也是 15ms 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 15ms 以上输出。且本信号 15ms 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, 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 15ms 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 ERRROUT 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 15ms 以上开始读入。 |          |                                           |                                |



| 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                                       |                                                          |
|-------------------------------------------------------------------------------|----|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------------------------------------------------------|----------------------------------------------------------|
|                                                                               |    |                              |                                                                                                                                                                                                                                                                                                                                                                 |          | CR800-R/Q<br>系列                                          | CR800-D<br>系列                                            |
| 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 15ms 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。                                                                                                                                                                                                                                                                                                                                        |          | 10022                                                    | -1                                                       |
| 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 15ms 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 15ms 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                                                       |
| ERRROUT                                                                       | 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 15ms 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 724 页的 “7.3 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 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 | 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, q, 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 724 页的 “7.3 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                                                                   |                                                                          |
|--------------------|----|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
|                    |    |                                    |                                                                                                                                                                                                                                                                                                                                                                                   |          | CR800-R/Q<br>系列                                                                      | CR800-D<br>系列                                                            |
| 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<br>J1, J2, J3, J4, J5, J6, J7, J8 轴<br>直交 JOG 时: 从开始号码<br>X, Y, Z, A, B, C, L1, L2 轴<br>圆筒 JOG 时: 从开始号码<br>X, q, Z, A, B, C, L1, L2 轴<br>三轴直交 JOG 时: 从开始号码<br>X, Y, Z, J4, J5, J6 轴<br>TOOL JOG 时: 从开始号码<br>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<br>第 724 页的“7.3 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<br>作为基准的工件坐标号码 (Ex-T 坐标号<br>码)。<br>注) 本输入信号根据 JOG 有效输入信号:<br>JOGENA 的 Edge (从 OFF 到 ON 的变<br>化) 进行获取。要变更工件坐标编号<br>时, 请使 JOG 有效输入信号:<br>JOGENA 暂时从 OFF 变为 ON<br>注) 关于 EX-T 控制及 Ex-T 坐标号码, 请<br>参照第 724 页的“7.3 关于 Ex-T 控制<br>”。                                                                                                                          | Level    | 注 3)<br>-1 (起始<br>位),<br>-1 (结束<br>位),<br>-1 (起始<br>位),<br>-1 (结束<br>位),             | 注 3)<br>-1 (起始<br>位),<br>-1 (结束<br>位),<br>-1 (起始<br>位),<br>-1 (结束<br>位), |
|                    | 输出 |                                    | 输出所输入的工件坐标号码 (Ex-T 坐标号<br>码)。<br>注) 关于 EX-T 控制及 Ex-T 坐标号码, 请<br>参照第 724 页的“7.3 关于 Ex-T 控制<br>”。                                                                                                                                                                                                                                                                                   |          |                                                                                      |                                                                          |
| 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                                                                       |
| HNDCNTL1           | 输入 | -                                  | -                                                                                                                                                                                                                                                                                                                                                                                 |          |                                                                                      |                                                                          |
|                    | 输出 | 机械 1 抓手输出信号<br>状态 (开始号码, 结<br>束号码) | CR800: 输出抓手输出 900 ~ 907 的状态<br>CR860: 输出抓手输出 764 ~ 771 的状态<br>例) 想将 900 ~ 903 的 4 点在通用输出信<br>号 3 号、4 号、5 号、6 号输出的情况<br>下, 将 HNDCNTL1 设定为 (3, 6)。                                                                                                                                                                                                                                  |          | HNDCNTL1<br>10048<br>(开始位)<br>10055<br>(结束位)                                         | -1 (开始<br>位),<br>-1 (结束<br>位),                                           |
| HNDSTS1            | 输入 | -                                  | -                                                                                                                                                                                                                                                                                                                                                                                 |          |                                                                                      |                                                                          |
|                    | 输出 | 机械 1 抓手输入信号<br>状态 (开始号码, 结<br>束号码) | CR800: 输出抓手输入 900 ~ 907 的状态<br>CR860: 输出抓手输入 764 ~ 775 的状态<br>例) 想将 900 ~ 903 的 4 点在通用输出信<br>号 3 号、4 号、5 号、6 号输出的情况<br>下, 将 HNDSTS1 设定在 (3, 6)。                                                                                                                                                                                                                                   |          | HNDSTS1<br>10056<br>(开始位)<br>10063<br>(结束位)<br>※ 如果使用 9<br>点以上, 则需要<br>更改结束位<br>的分配。 | -1 (开始<br>位),<br>-1 (结束<br>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          |                        |
|----------------------------------|----|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------------------------|------------------------|
|                                  |    |                           |                                                                                                                                                                                                                                                                                                                             |          | CR800-R/Q<br>系列             | CR800-D<br>系列          |
| 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><br>-1               | -1,<br><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: 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><br>-1               | -1,<br><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><br>-1               | -1,<br><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                          | 输入 | -                         | -                                                                                                                                                                                                                                                                                                                           |          | 注 5)                        | 注 5)                   |
|                                  | 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: 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) |          | 10064 (开始位),<br>10071 (结束位) | -1 (开始位),<br>-1 (结束位), |
| MnPTEXC<br>(n=1 ~ 3)             | 输入 | -                         | -                                                                                                                                                                                                                                                                                                                           |          | -1 (无意义),<br><br>-1         | -1 (无意义),<br><br>-1    |
|                                  | 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    |                             |                        |
| 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,<br><br>-1               | -1,<br><br>-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                     |

| 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                                                          |               |
|----------------------|----|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------|---------------|
|                      |    |                 |                                                                                                                                                                                                                                          |              | CR800-R/Q<br>系列                                                             | CR800-D<br>系列 |
| 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            |
| PSSLOT               | 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                                                                                                                                                                            |              |                                                                             |               |

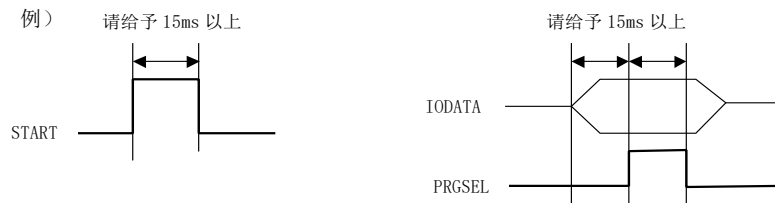
| 参数名      | 区分 | 名称            | 功能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 信号 Level | 工厂出货信号号码<br>输入, 输出 |                           |
|----------|----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------|---------------------------|
|          |    |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |          | CR800-R/Q<br>系列    | CR800-D<br>系列             |
| PSP05    | 输入 | -             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -        | -1, (无意义)          |                           |
|          | 输出 | 位置数据输出信号领域    | 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。<br>(1)CR800-R/Q 系列<br>10000 ~ 17872: CPU 缓冲存储器<br>(2)CR800-D 系列<br>2000 ~ 3632: PROFIBUS<br>6000 ~ 7728: CC-Link<br>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:<br>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。<br>可输出的数据范围<br>有符号整数 -2147483648 ~ 2147483647<br>请参阅第 640 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。 | -        | -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, -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                 | -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 |               |
|---------|----|-------|---------------------------------------------------------|----------|--------------------|---------------|
|         |    |       |                                                         |          | CR800-R/Q<br>系列    | CR800-D<br>系列 |
| 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 -1 ~ 19999。

可使用的输入输出信号号码根据不同的功能而异。

请确认所使用的外部输入输出功能的使用手册。

## 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。

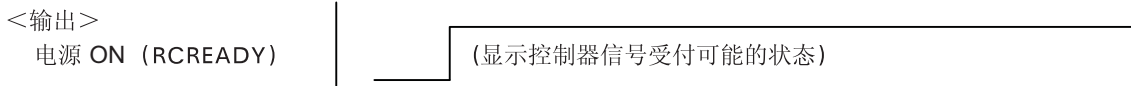
表 6-7：专用输入信号的有效 / 无效状态

| 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                         | 退避点回归输入    |                                     |                                     |
| 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                         | 退避点回归输入    |                                     |                                     |
| 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                           | 起动输入       |                                     | 在停止输入状态（STOPSTS 为 ON 状态）的时候没有功能。    |
| SAFEPOS                         | 退避点回归输入    |                                     |                                     |
| 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

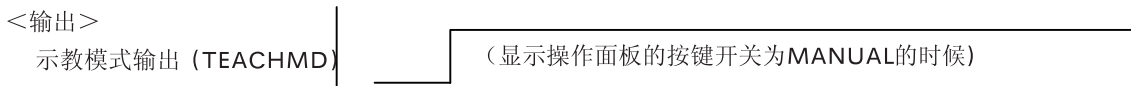
(1) RCREADY (控制器的电源 ON 完成输出)



(2) ATEXTMD (远程模式输出)



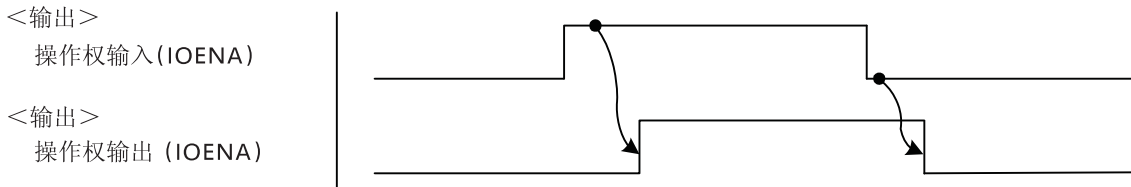
(3) TEACHMD (示教模式输出)



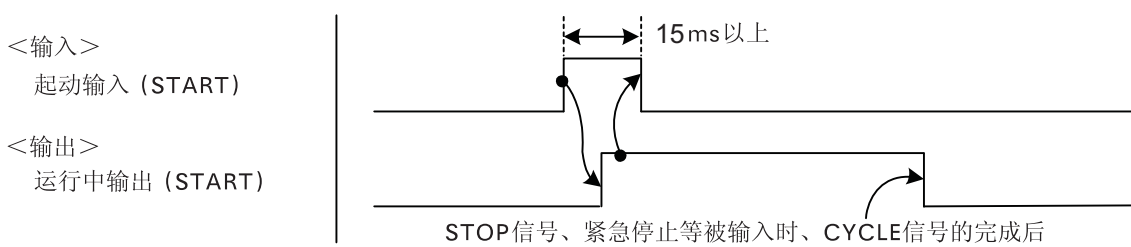
(4) ATTOPMD (自动模式输出)



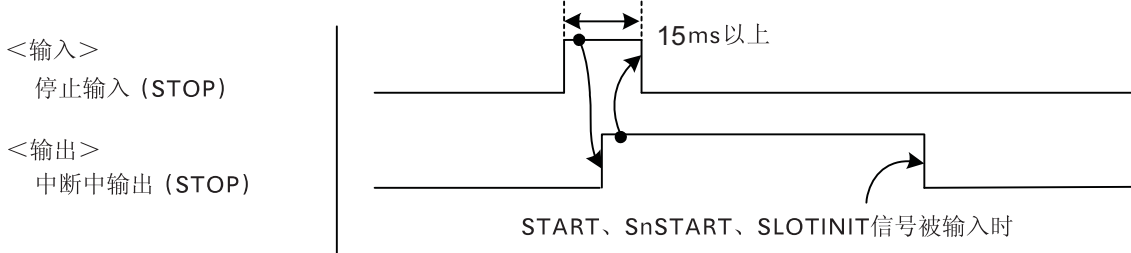
(5) IOENA (操作权输入信号 / 操作权输出信号)



(6) START (起动手输入 / 运行中输出)

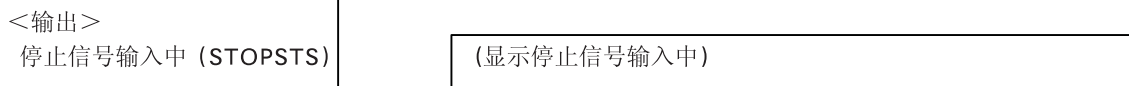


(7) STOP (停止输入 / 中断中输出)

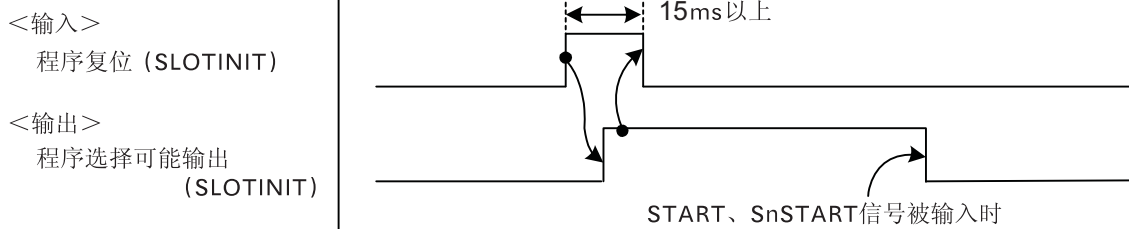




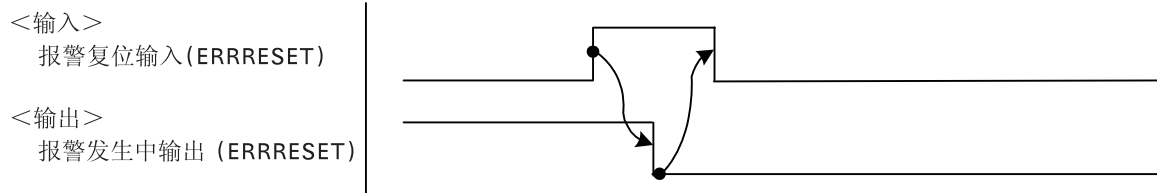
## (8) STOPSTS (停止信号输入中输出)



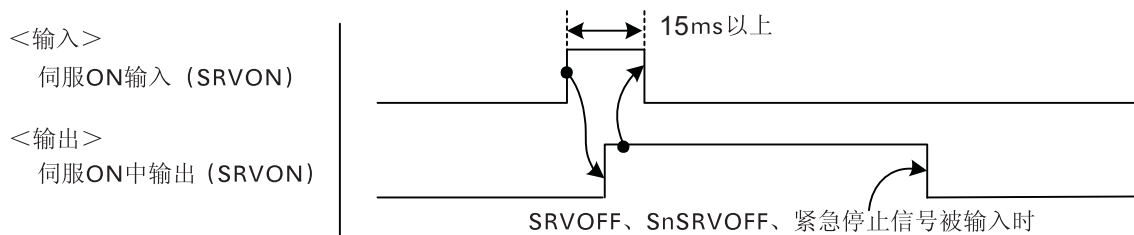
## (9) SLOTINIT (程序复位输入 / 程序选择可能输出)



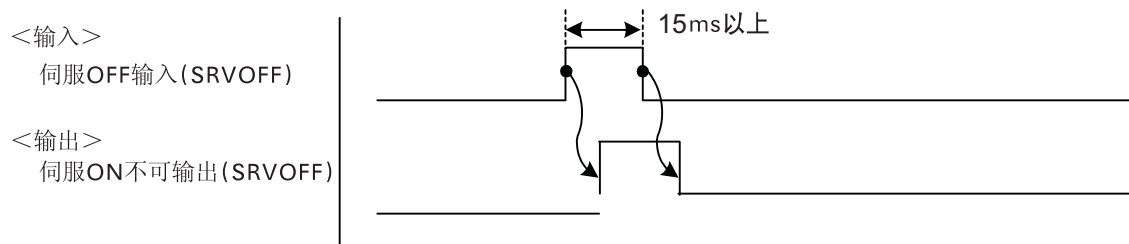
## (10) ERRRESET (报警复位输入 / 报警发生中输出)



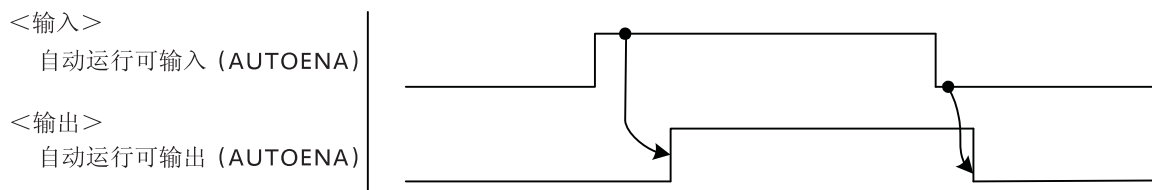
## (11) SRVON (伺服 ON 输入 / 伺服 ON 中输出)



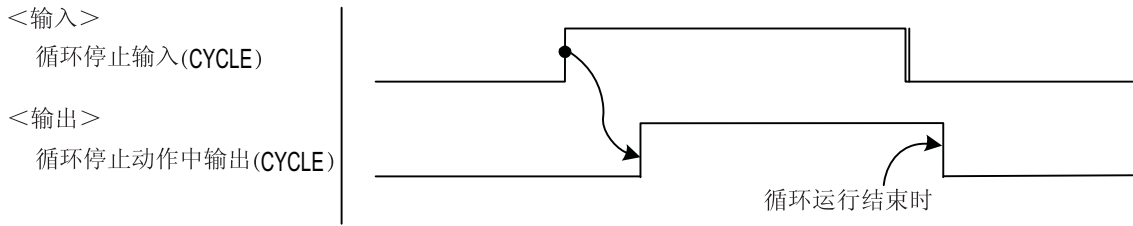
## (12) SRVOFF (伺服 OFF 输入 / 伺服 ON 不可输出)



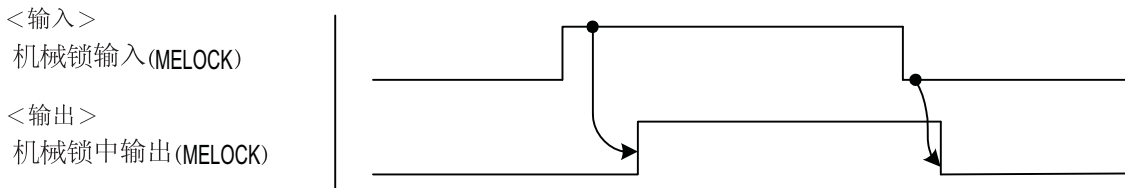
## (13) AUTOENA (自动运行输入 / 自动运行可输出)



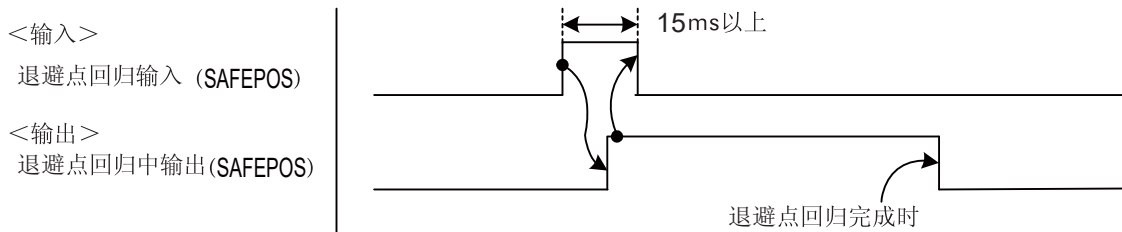
(14) CYCLE (循环停止输入 / 循环停止动作中输出)



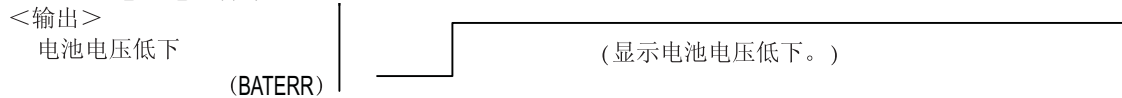
(15) MELOCK (机械锁输入 / 机械锁中输出)



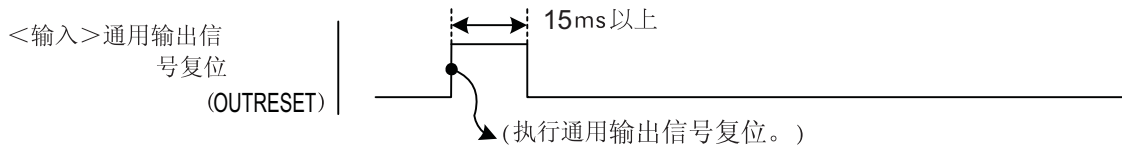
(16) SAFEPOS (退避点回归输入 / 退避点回归中输出)



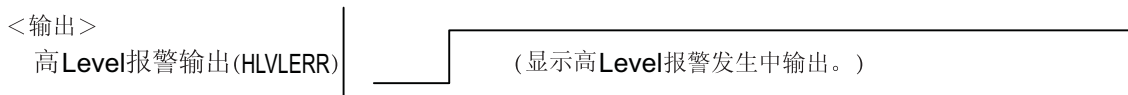
(17) BATERR (电池电压低下)



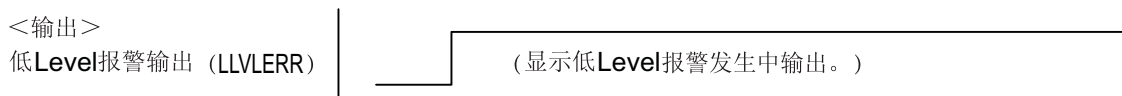
(18) OUTRESET (通用输出信号复位要求输入)



(19) HVLERR (高 Level 报警发生中输出)



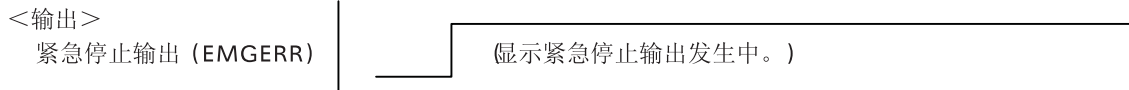
(20) LLVLERR (低 Level 报警发生中输出)



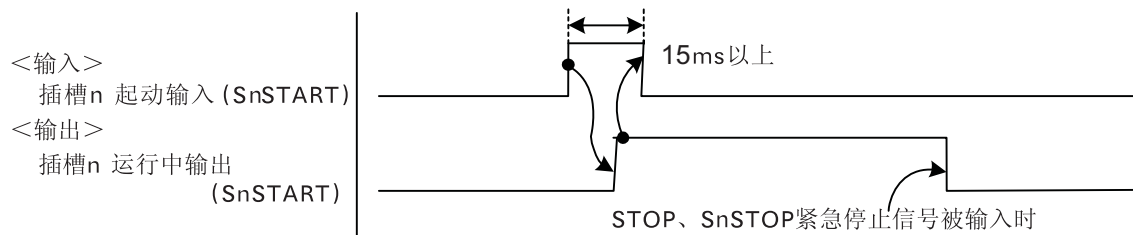
## (21) CLVLERR (警告 Level1 报警发生输出中)



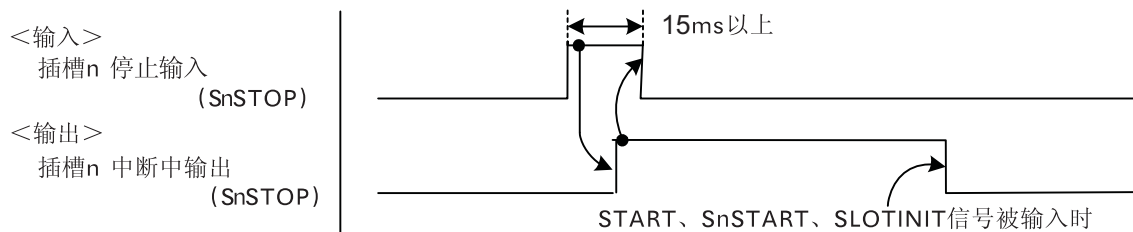
## (22) EMGERR (紧急停止中输出)



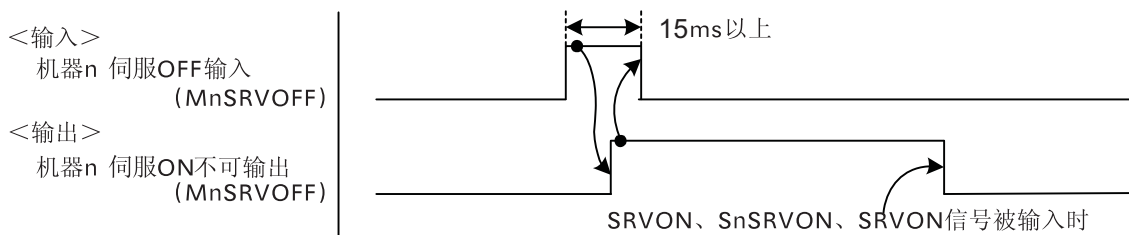
## (23) SnSTART (插槽 n 起动输入 / 插槽 n 运行中输出)



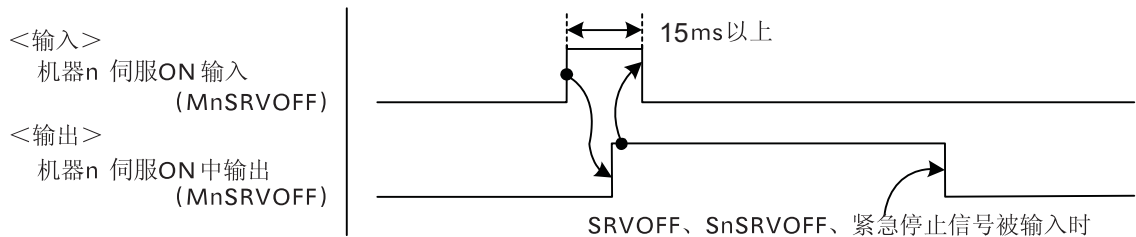
## (24) SnSTOP (插槽 n 停止输入 / 插槽 n 中断中输出)



## (25) MnSRVOFF (机器 n 伺服 OFF 输入 / 机器 n 伺服 ON 不可输出)

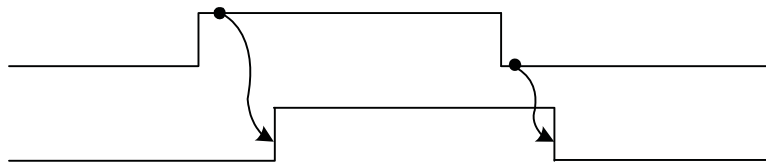


## (26) MnSRVON (机器 n 伺服 ON 输入 / 机器 n 伺服 ON 中输出)



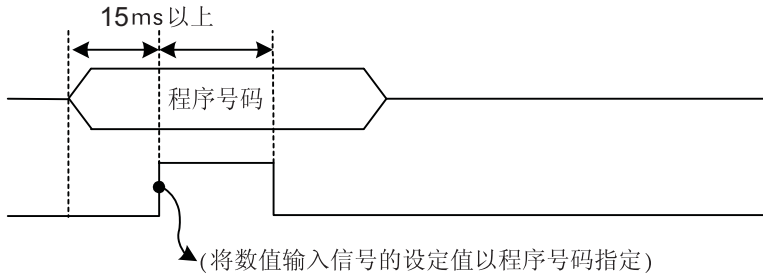
(27) MnMELOCK (机器 n 机械锁输入 / 机器 n 机械锁中输出)

<输入>  
 机器n机械锁输入  
 (MnMELOCK)  
 <输出>  
 机器n机械锁中输出  
 (MnMELOCK)



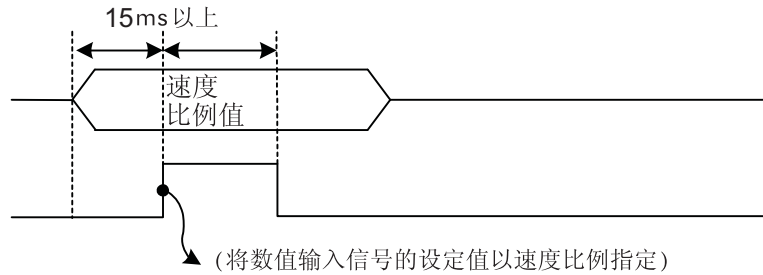
(28) PRGSEL (程序选择输入)  
 ∅ 和数值输出 (IODATA) 一起使用。

<输入>  
 数值输入 (IODATA)  
 程序选择输入 (PRGSEL)



(29) OVRDSEL (速度比例选择输入)  
 ∅ 和数值输出 (IODATA) 一起使用。

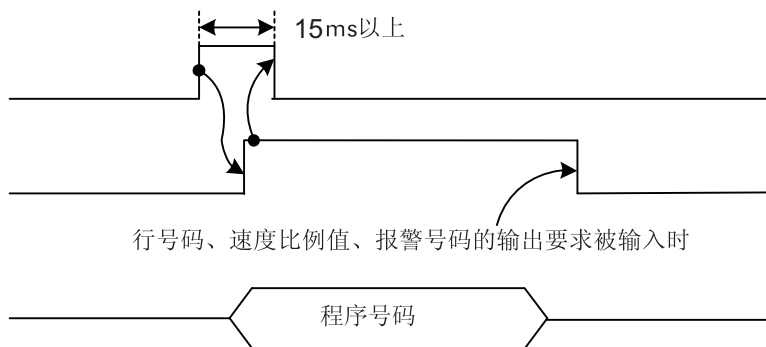
<输入>  
 数值输入 (IODATA)  
 速度比例选择输入  
 (OVRDSEL)



(30) IODATA (数值输入 / 数值输出)  
 ∅ 和 PRGSEL、OVRDSEL、PRGOUT、LINEOUT、OVRDOUT、ERRROUT 一起使用。

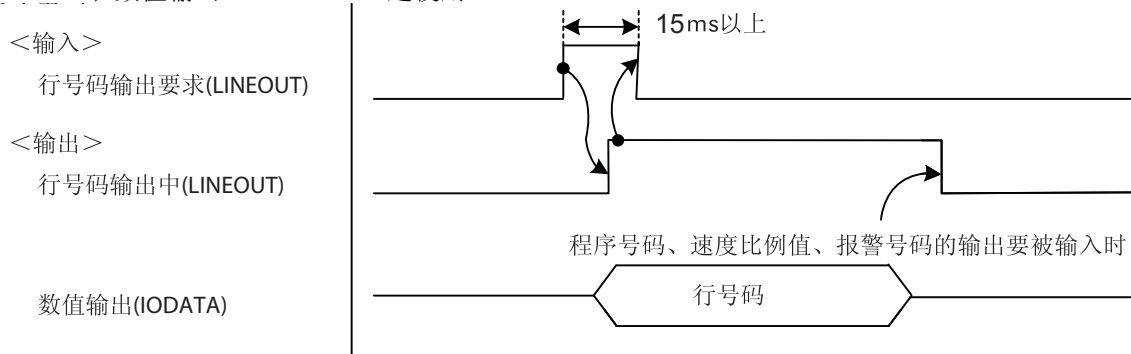
(31) PRGOUT (程序号码输出要求输入 / 程序号码输出中)  
 ∅ 和数值输出 (IODATA) 一起使用。

<输入>  
 程序号码输出要求  
 (PRGOUT)  
 <输出>  
 程序号码输出中  
 (PRGOUT)  
 数值输出 (IODATA)



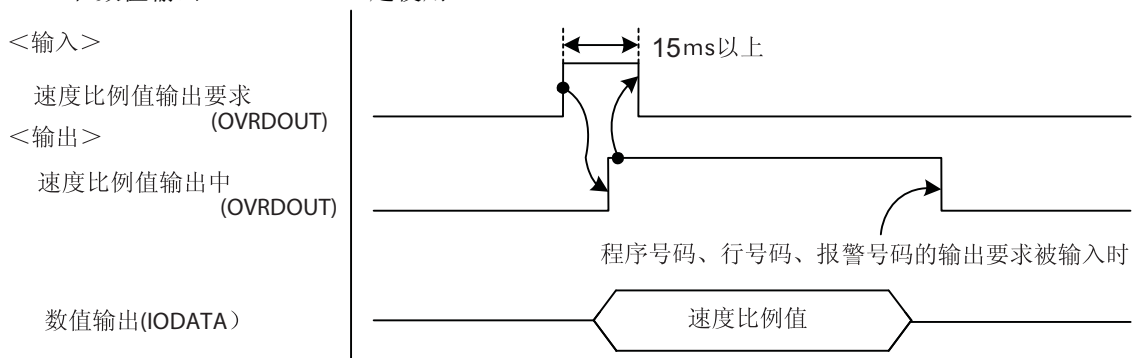
## (32) LINEOUT (行号码输出要求输入 / 行号码输出中)

φ 和数值输出 (IODATA) 一起使用



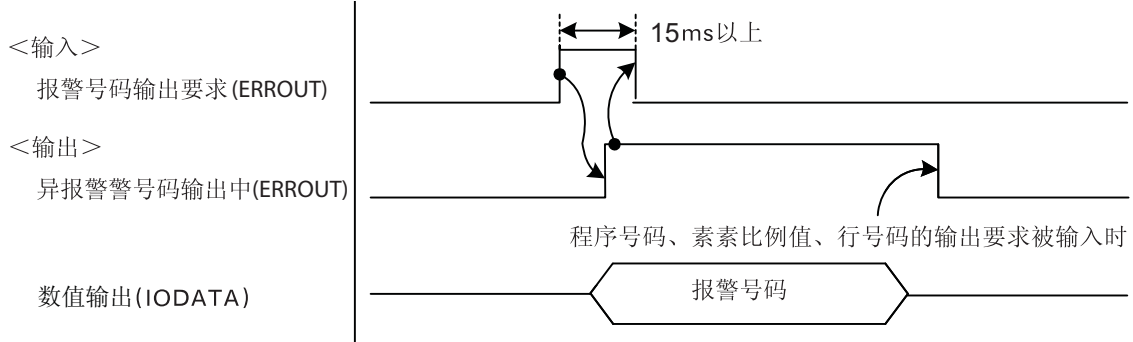
## (33) OVRDOUT (速度比例值输出要求 / 速度比例值输出中)

φ 和数值输出 (IODATA) 一起使用

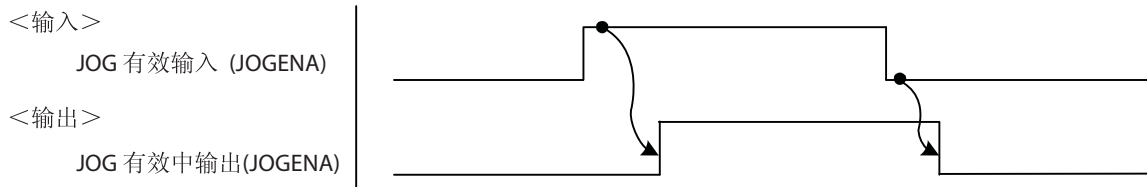


## (34) ERROUT (报警号码输出要求 / 报警号码输出中)

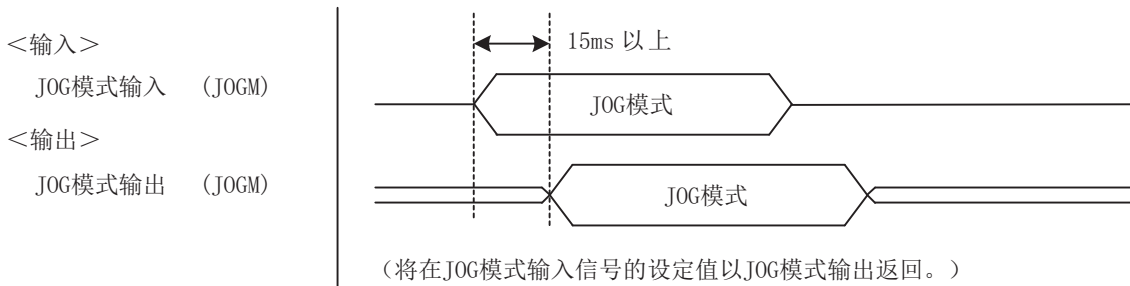
φ 和数值输出 (IODATA) 一起使用



## (35) JOGENA (JOG 有效输入 / JOG 有效中输出)



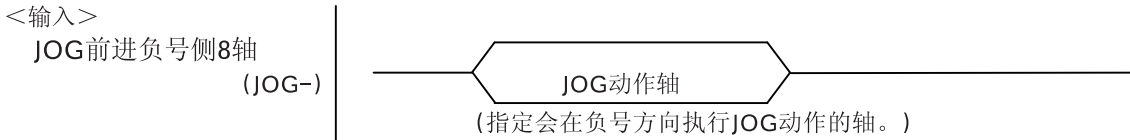
(36) JOGM (JOG 模式输入 / JOG 模式输出)



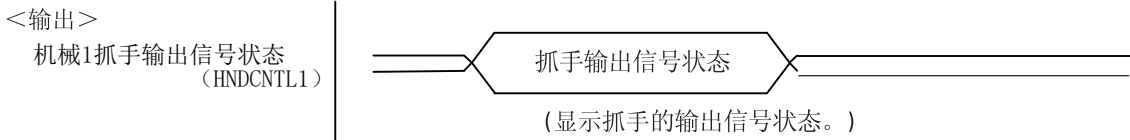
(37) JOG+ (在 JOG 前进正号侧输入 8 轴)



(38) JOG- (在 JOG 前进负号侧输入 8 轴)



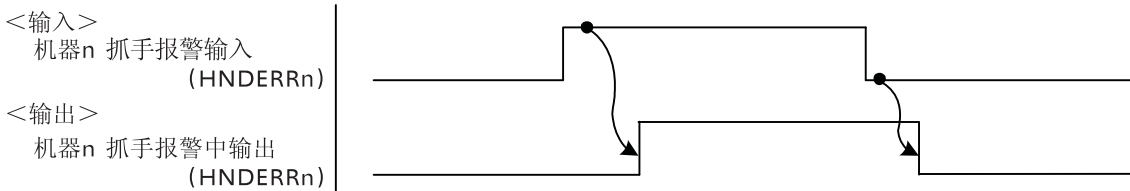
(39) HNDCTRL1 (机械 1 抓手输出信号状态)



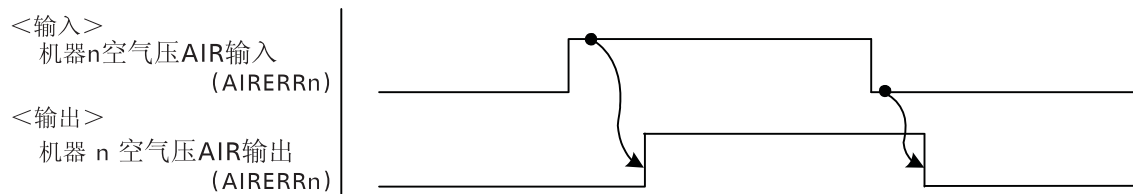
(40) HNDSTS1 (机械 1 抓手输入信号状态)



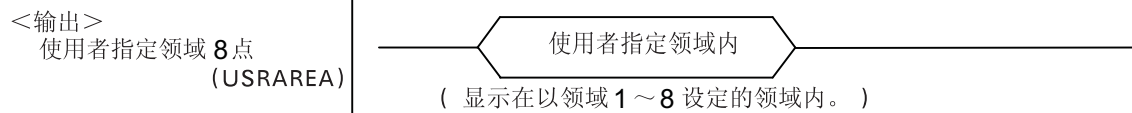
(41) HNDERRn (机器 n 抓手报警输入信号 / 机器 n 抓手报警中输出)



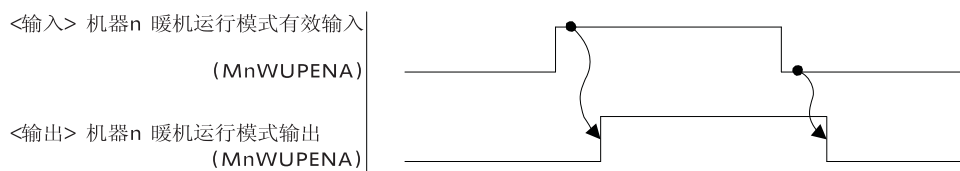
## (42) AIRERRn (机器 n 空气压 AIR 输入信号 / 机器 n 空气压 AIR 输出中)



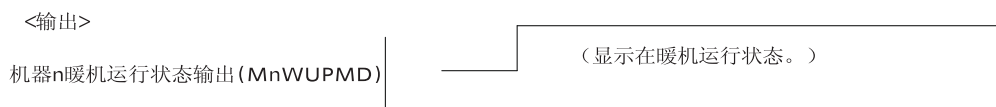
## (43) USRAREA (使用者定义领域 8 点输出)



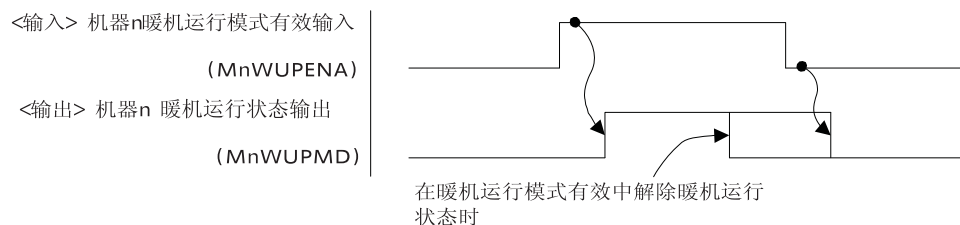
## (44) MnWUPENA (机器 n 暖机运行模式有效输入 / 机器 n 暖机运行模式输出)



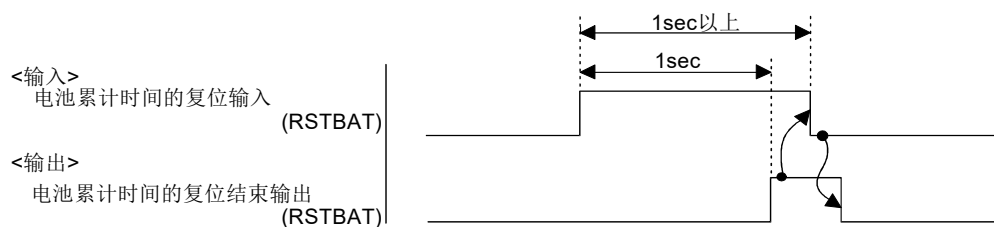
## (45) MnWUPMD (机器 n 暖机运行状态输出)



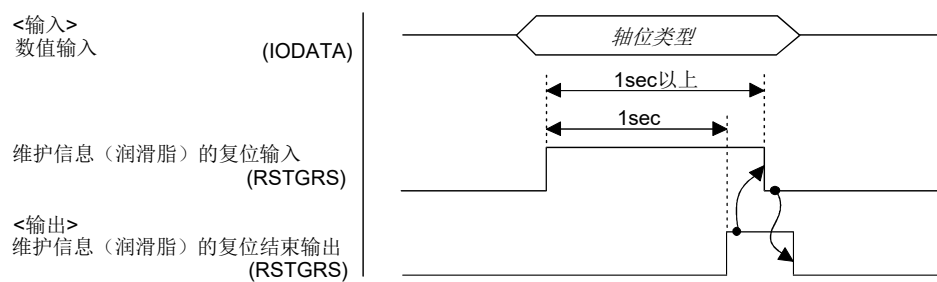
∅ 机器 n 和暖机运行模式有效输入 (MnWUPENA) 同持分配时如下所示。



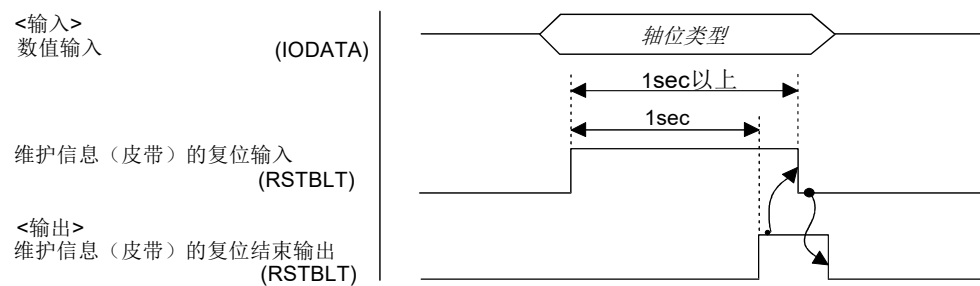
## (46) RSTBAT (电池累计时间的复位)



(47) RSTGRS (维护预报的复位 (润滑脂))



(48) RSTBLT (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)

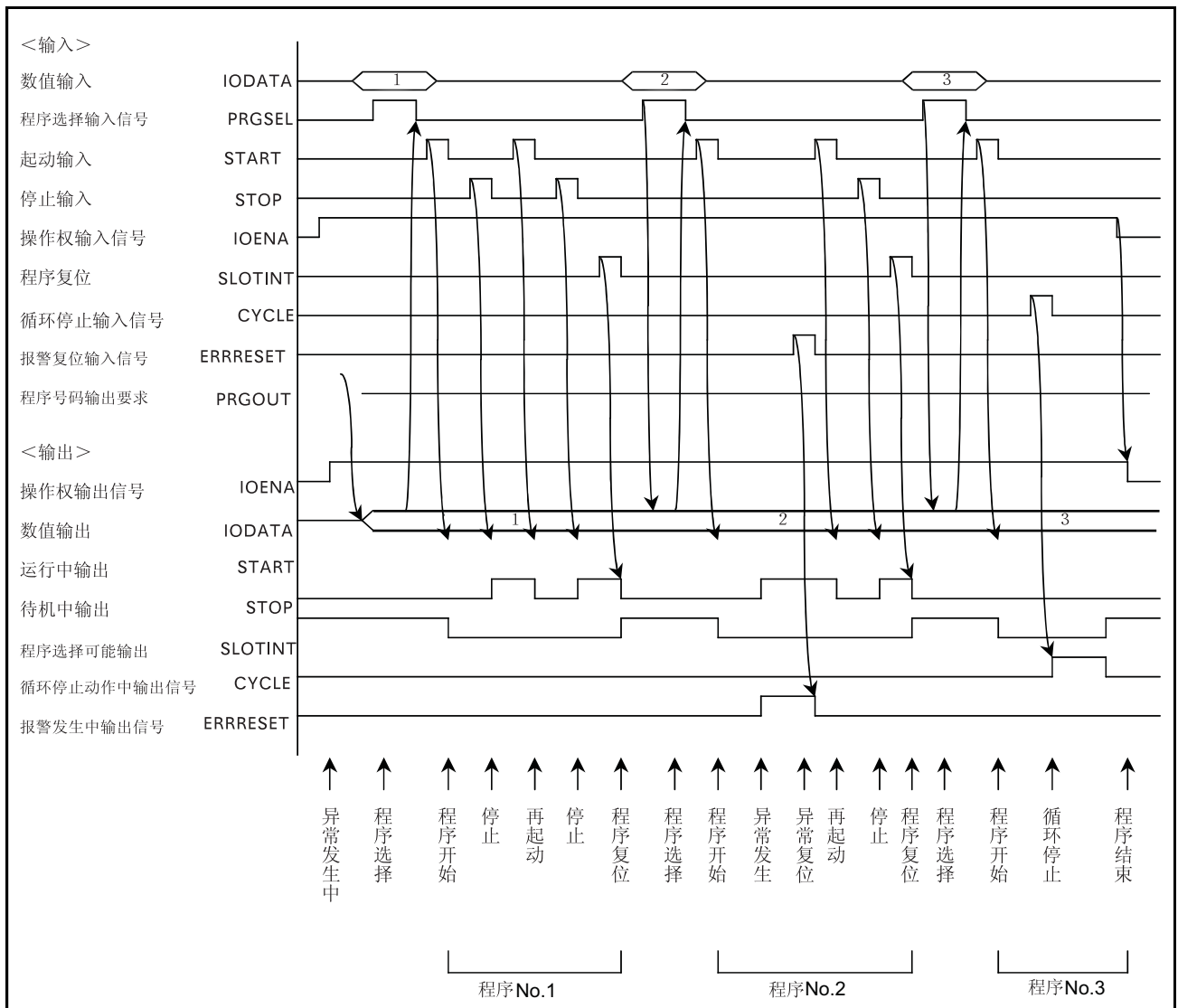


图 6-3: 外部操作时序图例 (其1)

(2) 外部信号操作时序图 (其 2)

在图 6-4 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。

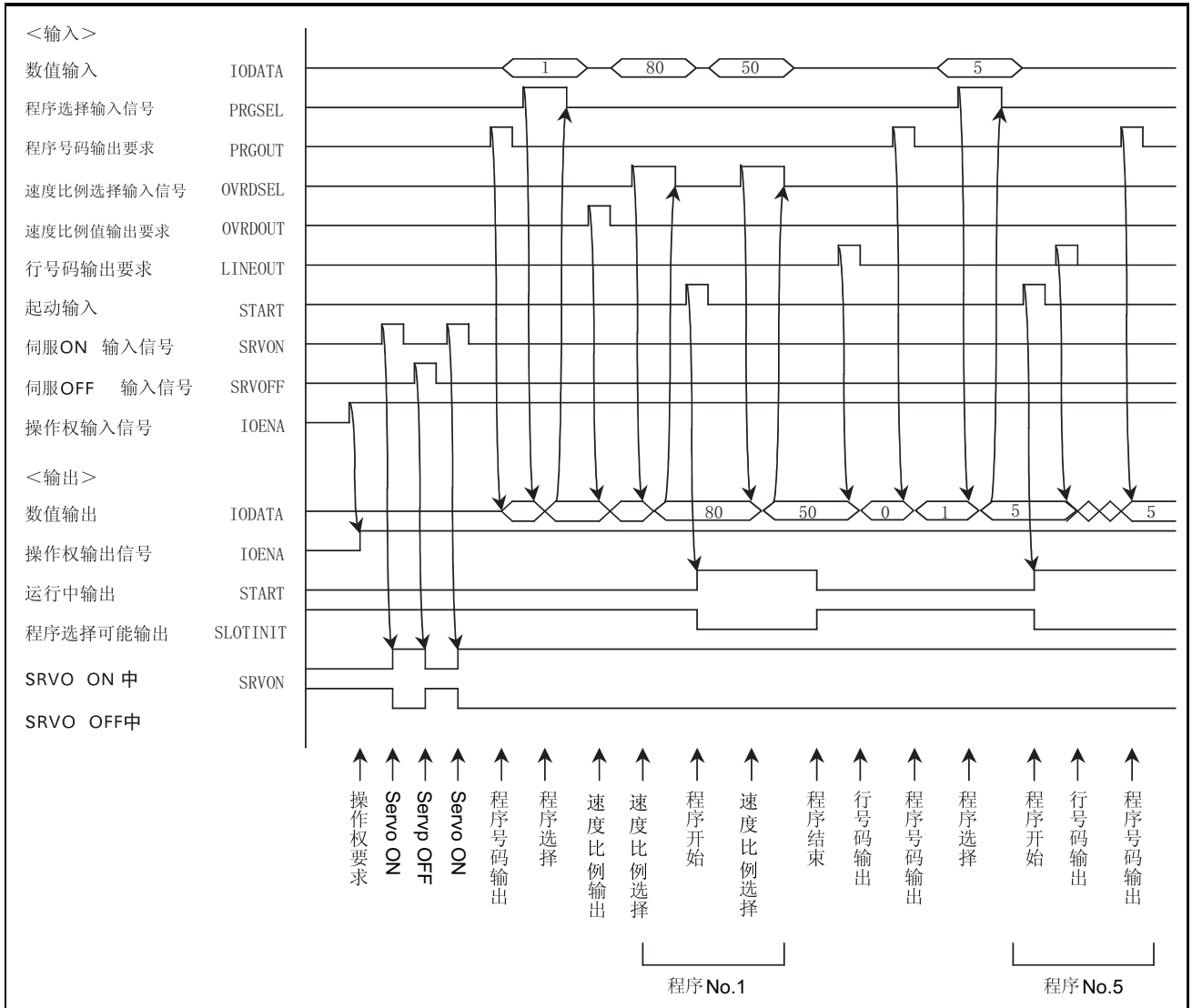


图 6-4: 外部信号操作时序图例 (其 2)

## (3) 外部信号操作时序图（其3）

在图 6-5 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。

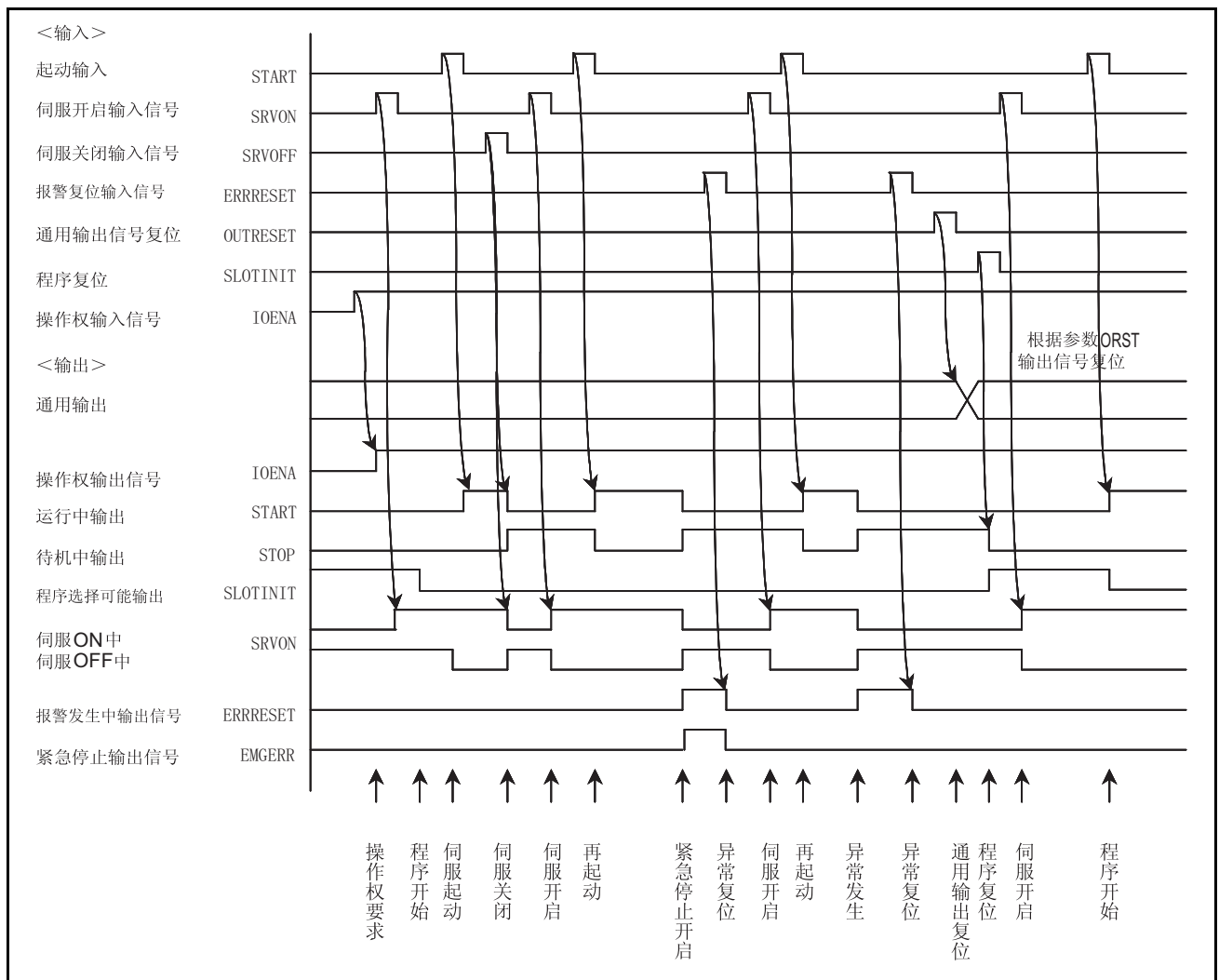


图 6-5: 外部信号操作时序图（其3）

(4) 外部信号操作时序图（其4）

在图 6-6 里显示来自外部信号的 JOG 操作、退避点回归、程序重置等的时序图例

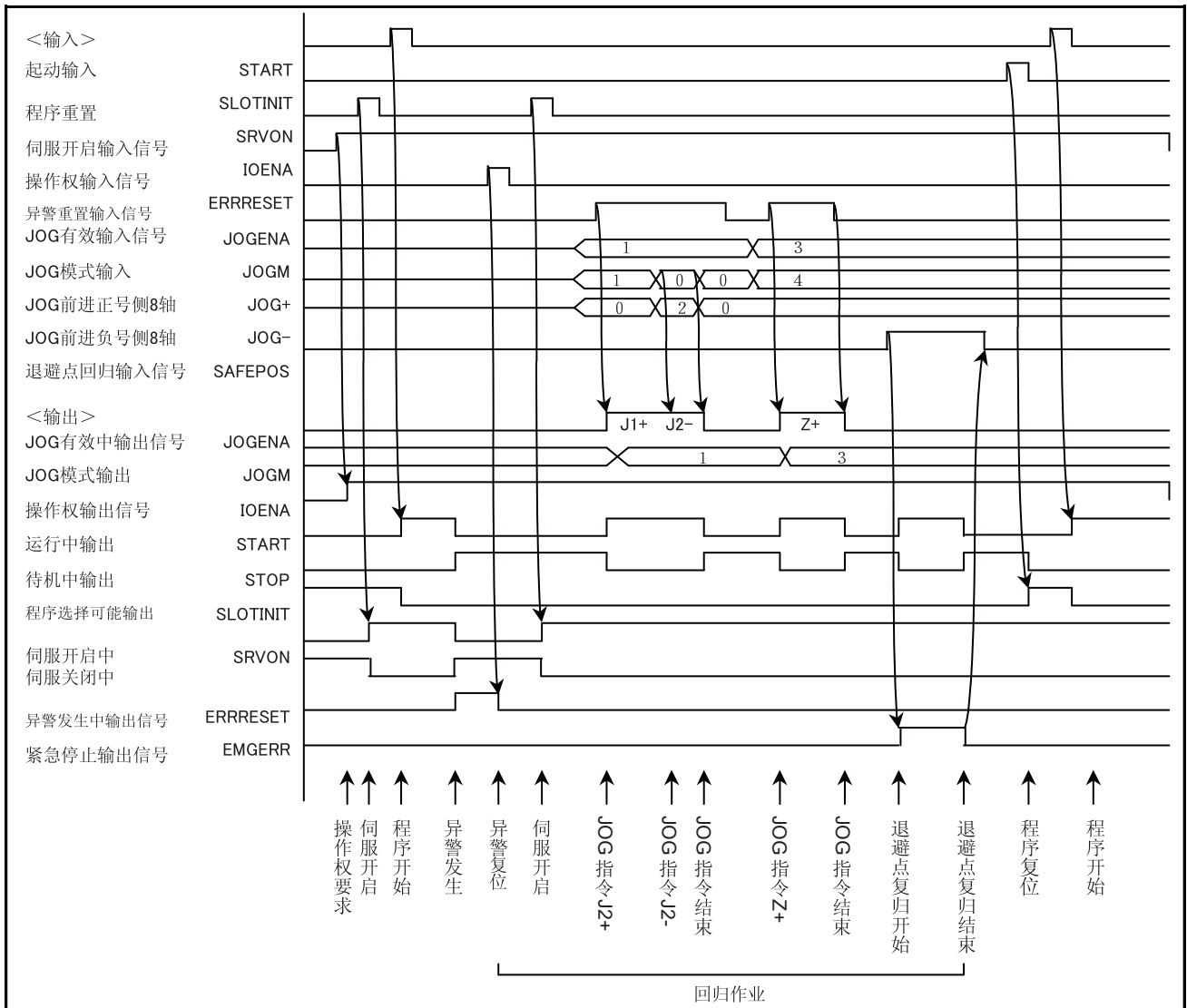


图 6-6: 外部信号操作时序图（其4）

## (5) 外部信号操作时序图（其 5）

下面是输出图 6-7 中指定的位置数据时的时序图示例。

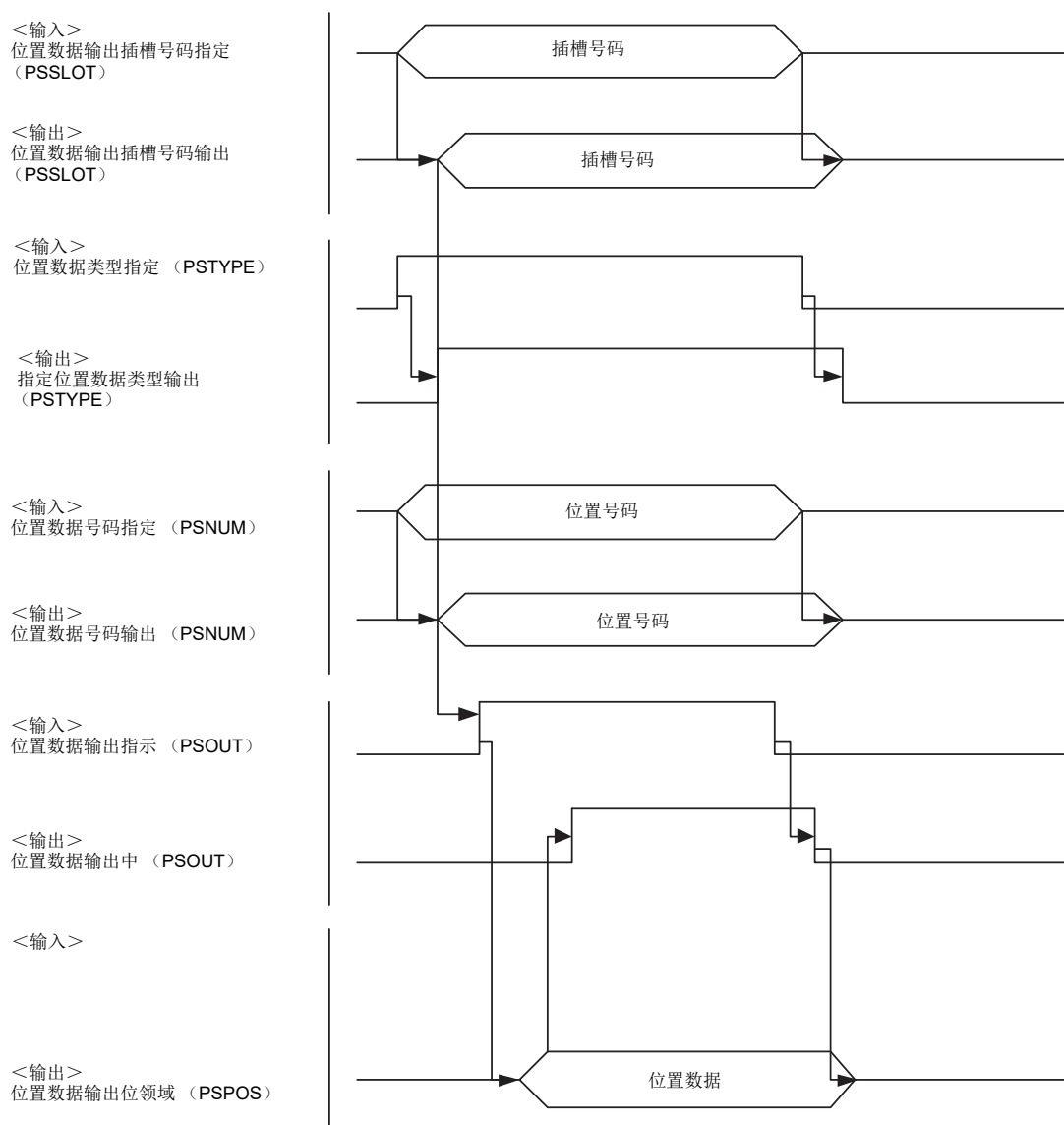


图 6-7：外部信号操作时序图（其 5）

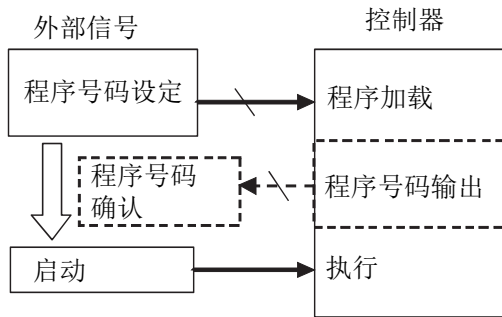
[注意]

- (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「PSSLLOT」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（PSSLLOT）、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”。

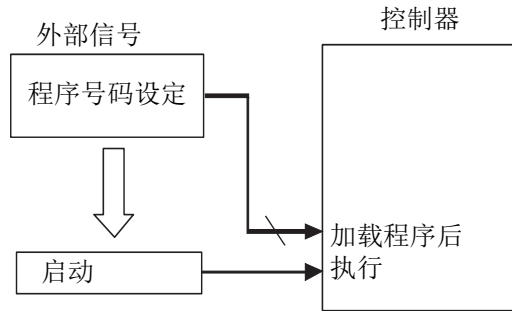
6.6 通过外部信号选择·执行程序的方法

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 RT ToolBox3 的【信号参数】→【专用输入输出信号分配】的设定画面。对图中圆圈标记部位进行设定。

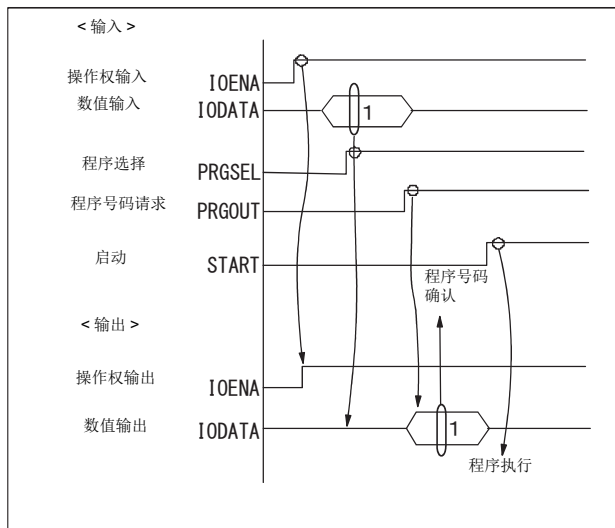
【专用输入输出信号分配】：【通用 1】画面

【专用输入输出信号分配】：【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 RT ToolBox3 的【参数设定】→【专用输入输出信号分配】的设定画面。对图中圆圈标记部位进行设定。

【专用输入输出信号分配】：【通用 1】画面

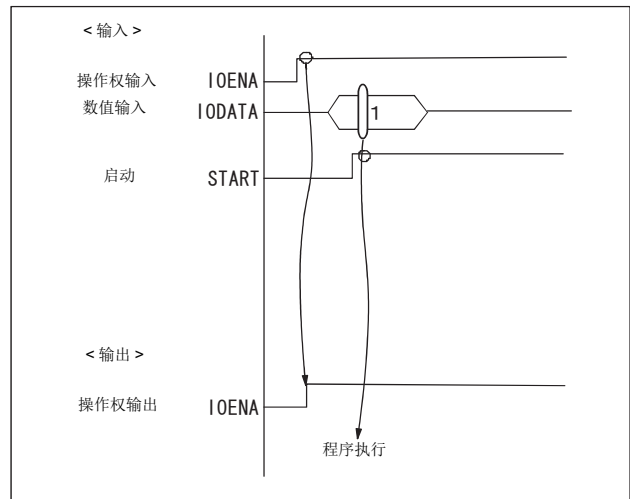


【专用输入输出信号分配】：【数据】画面



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 616 页的“6.3 专用输入输出”～第 626 页的“6.4 信号的有效无效状态”。

**注意**

通过外部信号选择并执行程序时，每次切换程序时数据将被写入至控制器内部的非易失性存储器。重复选择并执行程序时，会增加控制器内部非易失性存储器的写入次数，可能会导致故障，因此应将参数 AUTOSAVE 设定为 0 (不保存)。详细内容请参照 [5.4 程序参数] 中的 AUTOSAVE。



## 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。

## 6.8 软元件

## 6.8.1 软元件一览

从 GOT 及 SLMP 对应软元件可访问的机器人的软元件如下所示。此外，软元件范围是固定的，无法变更。

## (1) CR800-R 系列

表 6-11: 对应软元件一览

| 分类             | 类别   | 软元件名                    | 符号      | 软元件范围            |                         |       | 备注 |
|----------------|------|-------------------------|---------|------------------|-------------------------|-------|----|
|                |      |                         |         | 点数               | 设定范围                    | 表示    |    |
| 使用者软元件         | 位软元件 | 输入                      | X       | 4096 点 (4k)      | X0 ~ XFFF               | 16 进制 |    |
|                |      | 输出                      | Y       | 4096 点 (4k)      | Y0 ~ YFFF               | 16 进制 |    |
|                |      | 内部继电器                   | M       | 18432 点 (18k)    | M0 ~ M18431             | 10 进制 |    |
|                | 字软元件 | 数据寄存器                   | D       | 5120 点 (5k)      | D0 ~ D5119              | 10 进制 |    |
| 系统软元件          | 位软元件 | 特殊继电器                   | SM      | 4096 点           | SM0 ~ SM4095            | 10 进制 |    |
|                | 字软元件 | 特殊寄存器                   | SD      | 4096 点           | SD0 ~ SD4095            | 10 进制 |    |
| CPU 缓冲存储器访问软元件 | 字软元件 | CPU 缓冲存储器访问软元件          | U3En\G  | 524288 点         | U3En\G0 ~ U3En\G524287  | 10 进制 |    |
|                |      | CPU 缓冲存储器访问软元件 (定期通信区域) | U3En\HG | 最多 12288 点 (12k) | U3En\HG0 ~ U3En\HG12287 | 10 进制 |    |

## (2) CR800-D 系列

表 6-12: 对应软元件一览

| 分类             | 类别   | 软元件名                    | 符号                    | 软元件范围       |                                 |            | 备注    |
|----------------|------|-------------------------|-----------------------|-------------|---------------------------------|------------|-------|
|                |      |                         |                       | 点数          | 设定范围                            | 表示         |       |
| 使用者软元件         | 位软元件 | 输入                      | X                     | 8192 点 (4k) | X0 ~ X1FFF                      | 16 进制      |       |
|                |      | 输出                      | Y                     | 8192 点 (4k) | Y0 ~ Y1FFF                      | 16 进制      |       |
|                |      | 字软元件                    | 数据寄存器                 | D           | 5120 点 (5k)                     | D0 ~ D5119 | 10 进制 |
| 系统软元件          | 位软元件 | 特殊继电器                   | SM                    | 4096 点      | SM0 ~ SM4095                    | 10 进制      |       |
|                | 字软元件 | 特殊寄存器                   | SD                    | 4096 点      | SD0 ~ SD4095                    | 10 进制      |       |
| CPU 缓冲存储器访问软元件 | 字软元件 | CPU 缓冲存储器访问软元件 (定期通信区域) | U3En\HG (n=0, 1) 注 1) | 2048 点 (2k) | U3En\HG0 ~ U3En\HG2047 (n=0, 1) | 10 进制      |       |

注 1) U3E0/U3E1 为可读写。但是，U3E1 在“表 6-15 软元件分配”中所示功能为有效时，仅对象的软元件范围的属性变为 R。此时写入的内容将被忽略。

## (3) CR800-Q 系列

表 6-13: 对应软元件一览

| 分类           | 类别   | 软元件名          | 符号     | 软元件范围            |                           |       | 备注 |
|--------------|------|---------------|--------|------------------|---------------------------|-------|----|
|              |      |               |        | 点数               | 设定范围                      | 表示    |    |
| 使用者软元件       | 位软元件 | 输入            | X      | 4096 点 (4k)      | X0 ~ XFFF                 | 16 进制 |    |
|              |      | 输出            | Y      | 4096 点 (4k)      | Y0 ~ YFFF                 | 16 进制 |    |
|              |      | 内部继电器         | M      | 18432 点 (18k)    | M0 ~ M18431               | 10 进制 |    |
|              | 字软元件 | 数据寄存器         | D      | 5120 点 (5k)      | D0 ~ D5119                | 10 进制 |    |
| 系统软元件        | 位软元件 | 特殊继电器         | SM     | 2048 点           | SM0 ~ SM2047              | 10 进制 |    |
|              | 字软元件 | 特殊寄存器         | SD     | 2048 点           | SD0 ~ SD2047              | 10 进制 |    |
| CPU 共享存储器软元件 | 字软元件 | 多 CPU 间高速通信区域 | U3En\G | 最大 14336 点 (14k) | U3En\G10000 ~ U3En\G24335 | 10 进制 |    |

## 6.8.2 软元件分配

机器人功能及输入输出信号在软元件的分配如下所示。此外，未在功能中分配的软元件范围可自由读取 / 写入。

## (1) CR800-R 系列

表 6-14: 软元件分配

| 功能               | 从外部设备看到的软元件<br>注 1)                  | R/W 属性<br>注 2)   | 机器人的输入输出信号       | 点数       | 备注                              |
|------------------|--------------------------------------|------------------|------------------|----------|---------------------------------|
| 可编程控制器输入输出模块直接控制 | X0 ~ XFFF                            | R                | 输入 0 ~ 255       | 最多 256 点 | 搭载输入输出模块, 通过参数 QXYREAD 将功能变为有效。 |
|                  | Y0 ~ YFFF                            | R                | 输出 0 ~ 255       | 最多 256 点 |                                 |
| 抓手               | X384 ~ X38B                          | R                | 输入 900 ~ 907     | 8 点      | 始终有效。(CR800 时)                  |
|                  | X2FC ~ X307                          | R                | 输入 764 ~ 775     | 12 点     | CR860 时                         |
| 可编程控制器链接输入输出功能   | U3E0\HG0 ~ U3E0\HG511                | R/W              | 输入 10000 ~ 18191 | 8192 点   | 始终有效。                           |
|                  | U3En\HG0 ~ U3En\HG511 (n=1, 2, 3)    | R                | 输出 10000 ~ 18191 | 8192 点   |                                 |
| CPU 缓冲存储器扩展功能    | U3E0\HG512 ~ U3E0\HG1023             | R/W              | -                | -        | 通过参数 IQMEM 将功能变为有效。             |
|                  | U3En\HG512 ~ U3En\HG1023 (n=1, 2, 3) | R                | -                | -        |                                 |
| 可编程控制器软元件分配功能    | D4096 ~ D5119                        | R/W or R<br>注 3) | -                | -        | 通过参数 DDEVVL* 将功能变为有效。           |
| 特殊               | 继电器                                  | SM0 ~ SM4095     | -                | -        | 始终有效。                           |
|                  | 寄存器                                  | SD0 ~ SD4095     | -                | -        |                                 |

注 1) 未分配功能的领域及扩展功能无效时, 仍可自由写入 / 读取至软元件范围内。

注 2) 为功能有效时的 R/W 属性。功能无效时可进行 R/W。

注 3) 根据分配的变量类别, 为 R/W 或 R 的其中一个 (程序外部变量的情况下为 R/W, 状态变量的情况下为 R)。

## (2) CR800-D 系列

表 6-15: 软元件分配

| 功能           | 从外部设备看到的软元件<br>注 1) | R/W 属性<br>注 2) | 机器人的输入输出信号   | 点数              | 备注             |
|--------------|---------------------|----------------|--------------|-----------------|----------------|
| 并行输入输出模块     | X0 ~ XFF            | R              | 输入 0 ~ 255   | 256 点           | 将对象设备连接时变为有效。  |
|              | Y0 ~ YFF            | R              | 输出 0 ~ 255   | 256 点           |                |
| 并行输入输出接口     | X0 ~ X3F            | R              | 输入 0 ~ 63    | 64 点            |                |
|              | Y0 ~ Y3F            | R              | 输出 0 ~ 63    | 64 点            |                |
| GOT 链接       | X0 ~ XFF            | R/W 注 3)       | 输入 0 ~ 255   | 256 点           |                |
|              | Y0 ~ YF             | R              | 输出 0 ~ 255   | 256 点           |                |
| STOP (SKIP0) | X320                | R              | 输入 800       | 1 点             | 始终有效。          |
| SKIP (1-7)   | X321 ~ X327         | R              | 输入 801 ~ 807 | 7 点 (H/W 为 2 点) |                |
| 抓手           | X384 ~ X38B         | R              | 输入 900 ~ 907 | 8 点             | 始终有效。(CR800 时) |
|              | X2FC ~ X307         | R              | 输入 764 ~ 775 | 12 点            | CR860 时        |

| 功能             | 从外部设备看到的软元件<br>注 1)      | R/W 属性<br>注 2)   | 机器人的输入输出信号          | 点数        | 备注                    |
|----------------|--------------------------|------------------|---------------------|-----------|-----------------------|
| PROFIBUS       | X7D0 ~ X13CF             | R                | 输入 2000 ~ 5071      | 最多 3072 点 | 将选购卡连接时变为有效。          |
|                | Y7D0 ~ Y13CF             | R                | 输出 2000 ~ 5071      | 最多 3072 点 |                       |
| CC-Link        | Y1770 ~ Y1F6F            | R                | 远程输入 6000 ~ 8047    | 最多 2048 点 |                       |
|                | X1770 ~ X1F6F            | R                | 远程输出 6000 ~ 8047    | 最多 2048 点 |                       |
|                | D0 ~ D255                | R                | 远程寄存器输入 6000 ~ 6255 | 最多 128 字  |                       |
|                | D256 ~ D511              | R                | 远程寄存器输出 6000 ~ 6255 | 最多 128 字  |                       |
| CC-Link IEF    | Y1770 ~ Y1F6F            | R                | 远程输入 6000 ~ 8047    | 最多 2048 点 |                       |
|                | X1770 ~ X1F6F            | R                | 远程输出 6000 ~ 8047    | 最多 2048 点 |                       |
|                | D0 ~ D1023               | R                | 远程寄存器输入 6000 ~ 7023 | 最多 768 字  |                       |
|                | D1024 ~ D2047            | R                | 远程寄存器输出 6000 ~ 7023 | 最多 768 字  |                       |
| 可编程控制器链接输入输出功能 | U3E0\HG0 ~ U3E0\HG511    | R/W              | 输入 10000 ~ 18191    | 8192 点    | 始终有效。                 |
|                | U3E1\HG0 ~ U3E1\HG511    | R                | 输出 10000 ~ 18191    | 8192 点    |                       |
| CPU 缓冲存储器扩展功能  | U3E0\HG512 ~ U3E0\HG1023 | R/W              | -                   | -         | 通过参数 IQMEM 将功能变为有效。   |
|                | U3E1\HG512 ~ U3E1\HG1023 | R                | -                   | -         |                       |
| 可编程控制器软件元件分配功能 | D4096 ~ D5119            | R/W or R<br>注 4) | -                   | -         | 通过参数 DDEVVL* 将功能变为有效。 |
| 特殊             | 继电器                      | SM0 ~ SM4095     | R/W                 | -         | 始终有效。                 |
|                | 寄存器                      | SD0 ~ SD4095     | R/W                 | -         |                       |

注 1) 未分配功能的领域及扩展功能无效时，仍可自由写入 / 读取至软元件范围内。

注 2) 为功能有效时的 R/W 属性。功能无效时可进行 R/W。

注 3) 为了从外部设备写入 (对机器人的指令)，软元件在并行输入输出接口及并行输入输出模块中应为未使用状态。

注 4) 根据分配的变量类别，为 R/W 或 R 的其中一个 (程序外部变量的情况下为 R/W，状态变量的情况下为 R)。

## (3) CR800-Q 系列

表 6-16: 软元件分配

| 功能               | 从外部设备观察到的软元件<br>注 1)                     | R/W 属性<br>注 2)   | 机器人的输入输出信号       | 点数       | 备注                              |
|------------------|------------------------------------------|------------------|------------------|----------|---------------------------------|
| 可编程控制器输入输出模块直接控制 | X0 ~ XFFF                                | R                | 输入 0 ~ 255       | 最多 256 点 | 搭载输入输出模块, 通过参数 QXYREAD 将功能变为有效。 |
|                  | Y0 ~ YFFF                                | R                | 输出 0 ~ 255       | 最多 256 点 |                                 |
| 抓手               | X384 ~ X38B                              | R                | 输入 900 ~ 907     | 8 点      | 始终有效。(CR800 时)                  |
|                  | X2FC ~ X307                              | R                | 输入 764 ~ 775     | 12 点     | CR860 时                         |
| 可编程控制器链接输入输出功能   | U3E0\G10000 ~ U3E0\G10511<br>(2 号机时)     | R/W              | 输入 10000 ~ 18191 | 8192 点   | 始终有效。                           |
|                  | U3En\G10000 ~ U3En\G10511<br>(n=1, 2, 3) | R                | 输出 10000 ~ 18191 | 8192 点   |                                 |
| 共享存储器扩展功能        | U3E0\G10512 ~ U3E0\G11023<br>(2 号机时)     | R/W              | -                | -        | 通过参数 IQMEM 将功能变为有效。             |
|                  | U3En\G512 ~ U3En\G1023<br>(n=1, 2, 3)    | R                | -                | -        |                                 |
| 可编程控制器软元件分配功能    | D4096 ~ D5119                            | R/W or R<br>注 3) | -                | -        | 通过参数 DDEVVL* 将功能变为有效。           |
| 特殊               | 继电器                                      | SM0 ~ SM2047     | R/W              | -        | 始终有效。                           |
|                  | 寄存器                                      | SD0 ~ SD2047     | R/W              | -        |                                 |

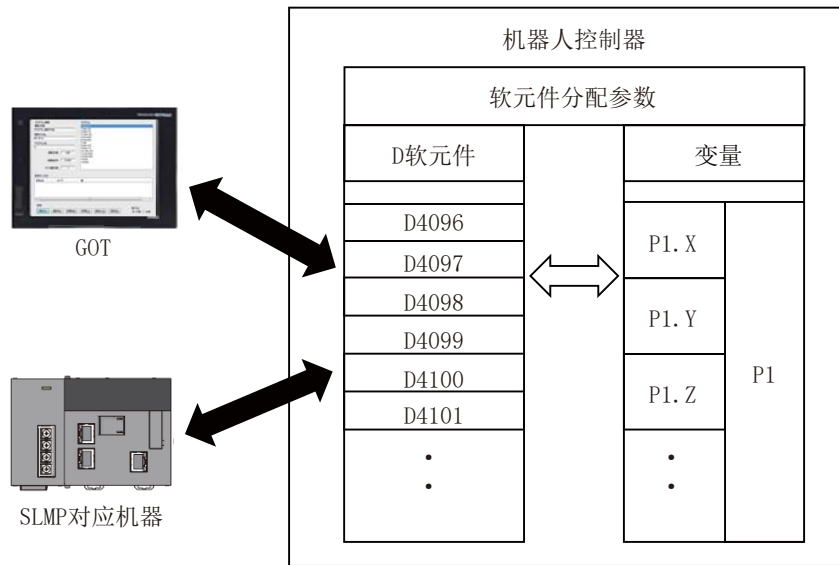
注 1) 未分配功能的领域及扩展功能无效时, 仍可自由写入 / 读取至软元件范围内。

注 2) 为功能有效时的 R/W 属性。功能无效时可进行 R/W。

注 3) 根据分配的变量类别, 为 R/W 或 R 的其中一个 (程序外部变量的情况下为 R/W, 状态变量的情况下为 R)。

6.8.3 可编程控制器软元件分配功能

通过将程序外部变量及状态变量直接分配到软元件，可经由软元件从 GOT 等 FA 设备收集机器人的信息。



- (1) 对应软元件  
可分配的软元件如下所示。

表 6-17: 对应软元件

| 类别   | 软元件名  | 符号 | 软元件范围       |               |       | 备注                      |
|------|-------|----|-------------|---------------|-------|-------------------------|
|      |       |    | 点数          | 设定范围          | 表示    |                         |
| 字软元件 | 数据寄存器 | D  | 1024 点 (1k) | D4096 ~ D5119 | 10 进制 | 2 字节 * 1024 点 = 2048 字节 |

- (2) 向机器人变量的软元件的分配  
可分配的机器人变量如下所示。

表 6-18: 可分配变量

| 分类 | 类别     | 数据类型          | 备注                                                                                                                                                                                                             |
|----|--------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 变量 | 程序外部变量 | 位置型           | 也可分别对应 1 次元排列<br>※ 关于程序外部变量，请参照第 141 页的「4.3 MELFA-BASIC VI 的详细规格」。<br><br>※ 关于状态变量，请参照第 162 页的「4.5 机器人（系统）状态变量」、第 333 页的「4.13 机器人（系统）状态变量的详细说明」。<br>此外，M_In()、M_Out() 等状态变量的情况下，不对应有输入输出功能的各种可编程控制器信号的状态变量的分配。 |
|    |        | 关节型           |                                                                                                                                                                                                                |
|    |        | 数值型（仅限双精度实数型） |                                                                                                                                                                                                                |
|    |        | 字符串型          |                                                                                                                                                                                                                |
|    | 状态变量   | 位置型           |                                                                                                                                                                                                                |
|    |        | 关节型           |                                                                                                                                                                                                                |
|    |        | 数值型           |                                                                                                                                                                                                                |
|    |        | 字符串型          |                                                                                                                                                                                                                |

通过以下参数进行变量和软元件的分配。

| 参数             | 参数名                   | 排列数<br>字符串              | 内容说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 出厂时设定值                   |
|----------------|-----------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| 对 D 软元件分配机器人变量 | DDEVVL1 ~<br>DDEVVL32 | 整数 4、<br>字符串 1、<br>整数 5 | <p>进行对 D 软元件的机器人变量分配。最多可进行 32 的分配设定。</p> <p>要素 1: 变量类别<br/>0: 未设定<br/>1: 程序外部变量<br/>2: 状态变量</p> <p>要素 2: 数据类型<br/>0: 整数型<br/>1: 长精度整数型<br/>2: 单精度实数类型<br/>3: 双精度实数型<br/>4: 位置型<br/>5: 关节型<br/>6: 工件坐标型<br/>7: 字符串型</p> <p>要素 3: (预留)</p> <p>要素 4: 具有实数值的数据型的分配方法 (仅在要素 2 中 2 ~ 6 时有效, 除此以外时忽略)<br/>0: 单精度实数型 (也包括位置型、关节型、工件坐标型的成分) 进行 <math>10^4</math> 倍处理后, 作为长精度整数型进行分配。双精度实数型进行 <math>10^8</math> 倍处理后, 作为 64 位整数 (长精度整数的 2 倍长) 进行分配。<br/>1: 以变量原型进行分配。</p> <p>要素 5: 变量名<br/>(例) M_100、J_Fbc 等</p> <p>要素 6: 自变量 1<br/>※ 外部变量时: 次元排列的要素、状态变量时: 自变量 1<br/>设定范围: 空白 (省略时) 或 0 ~</p> <p>要素 7: 自变量 2 (仅限状态变量时)<br/>设定范围: 空白 (省略时) 或 0 ~</p> <p>要素 8: (预留)</p> <p>要素 9: 字符分配字节数 (仅在要素 2 中 2 ~ 7 时有效, 除此以外时忽略)<br/>※ 对 1 (1 字节的字符串长存储领域) + 要分配的字符串变量的最大字节数进行指定。<br/>设定范围: 1 ~ 240</p> <p>要素 10: 起始 D 软元件编号<br/>设定范围: 4096 ~ 5119<br/>(例) D4096 时指定为 4096</p> | 0, 0, 0, 0, ..., 1, 4096 |

#### ■ 向数值变量的分配示例

(1) 整数

[例] M\_OP0vrd (整数)  
分配到 D4096 时。

| 软元件   | 内容                       | 更新周期 |
|-------|--------------------------|------|
| D4096 | M_OP0vrd 的值 [ 整数 (附符号) ] | 控制周期 |

[例] M\_ErCode (长精度整数)  
分配到 D4096 时。

| 软元件   | 内容                          | 更新周期 |
|-------|-----------------------------|------|
| D4096 | M_ErCode 的值 [ 长精度整数 (附符号) ] | 控制周期 |
| D4097 |                             |      |

(2) 实数

[例] M\_Timer (1) (单精度实数)  
分配到 D4500 时。

| 软元件   | 内容                                                          | 更新周期 |
|-------|-------------------------------------------------------------|------|
| D4500 | M_Timer (1) 的值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数] | 控制周期 |
| D4501 |                                                             |      |

[例] M\_100 (1) (双精度实数)  
分配到 D4500 时。

| 软元件   | 内容                                                          | 更新周期 |
|-------|-------------------------------------------------------------|------|
| D4500 | M_100 (1) 的值 [10 <sup>8</sup> 倍处理后的 64 位整数 (附符号)] 或 [双精度实数] | 控制周期 |
| D4501 |                                                             |      |
| D4502 |                                                             |      |
| D4503 |                                                             |      |

■ 向位置变量 / 工件坐标的分配示例

向 D5000 分配了位置型 / 工件坐标型变量时

| 软元件   | 内容                                                  | 更新周期 |
|-------|-----------------------------------------------------|------|
| D5000 | X 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数]  | 控制周期 |
| D5001 |                                                     |      |
| D5002 | Y 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数]  |      |
| D5003 |                                                     |      |
| D5004 | Z 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数]  |      |
| D5005 |                                                     |      |
| D5006 | A 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数]  |      |
| D5007 |                                                     |      |
| D5008 | B 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数]  |      |
| D5009 |                                                     |      |
| D5010 | C 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数]  |      |
| D5011 |                                                     |      |
| D5012 | L1 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数] |      |
| D5013 |                                                     |      |
| D5014 | L2 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数 (附符号)] 或 [单精度实数] |      |
| D5015 |                                                     |      |
| D5016 | 结构标志 <sup>注1)</sup> [长精度整数 (附符号)]                   |      |
| D5017 |                                                     |      |
| D5018 | 多旋转数据 [长精度整数 (附符号)]                                 |      |
| D5019 |                                                     |      |
| D5020 | 机器编号 (仅限 MELFA-BASIC VI) [长精度整数 (附符号)]              |      |
| D5021 |                                                     |      |

注 1) 结构标志 / 机器编号仅限下位 1 字。上位 1 字为预留领域。



■ 向关节变量的分配示例

向 D5000 分配了关节型变量时。

| 软元件   | 内容                                            | 更新周期 |
|-------|-----------------------------------------------|------|
| D5000 | J1 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] | 控制周期 |
| D5001 | ]                                             |      |
| D5002 | J2 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5003 | ]                                             |      |
| D5004 | J3 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5005 | ]                                             |      |
| D5006 | J4 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5007 | ]                                             |      |
| D5008 | J5 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5009 | ]                                             |      |
| D5010 | J6 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5011 | ]                                             |      |
| D5012 | J7 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5013 | ]                                             |      |
| D5014 | J8 坐标值 [10 <sup>4</sup> 倍处理后的长精度整数] 或 [单精度实数] |      |
| D5015 | ]                                             |      |
| D5016 | 机器编号 (仅限 MELFA-BASIC VI) [长精度整数 (附符号)]注1)     |      |
| D5017 | ]                                             |      |

注 1) 机器编号仅限低位 1 字。高位 1 字为预留领域。

■ 向字符串变量的分配示例

以字符分配字节数 = 7 向 D5100 分配了字符串型变量时。

| 软元件   | 内容                                      | 更新周期 |
|-------|-----------------------------------------|------|
| D5100 | 字符串长 [低位 8 位]、第 1 字节字符代码 [高位 8 位]       | 控制周期 |
| D5101 | 第 2 字节字符代码 [低位 8 位]、第 3 字节字符代码 [高位 8 位] |      |
| D5102 | 第 4 字节字符代码 [低位 8 位]、第 5 字节字符代码 [高位 8 位] |      |
| D5103 | 第 6 字节字符代码 [低位 8 位]                     |      |

## 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。

#### ■垂直多关节机器人时

##### (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)

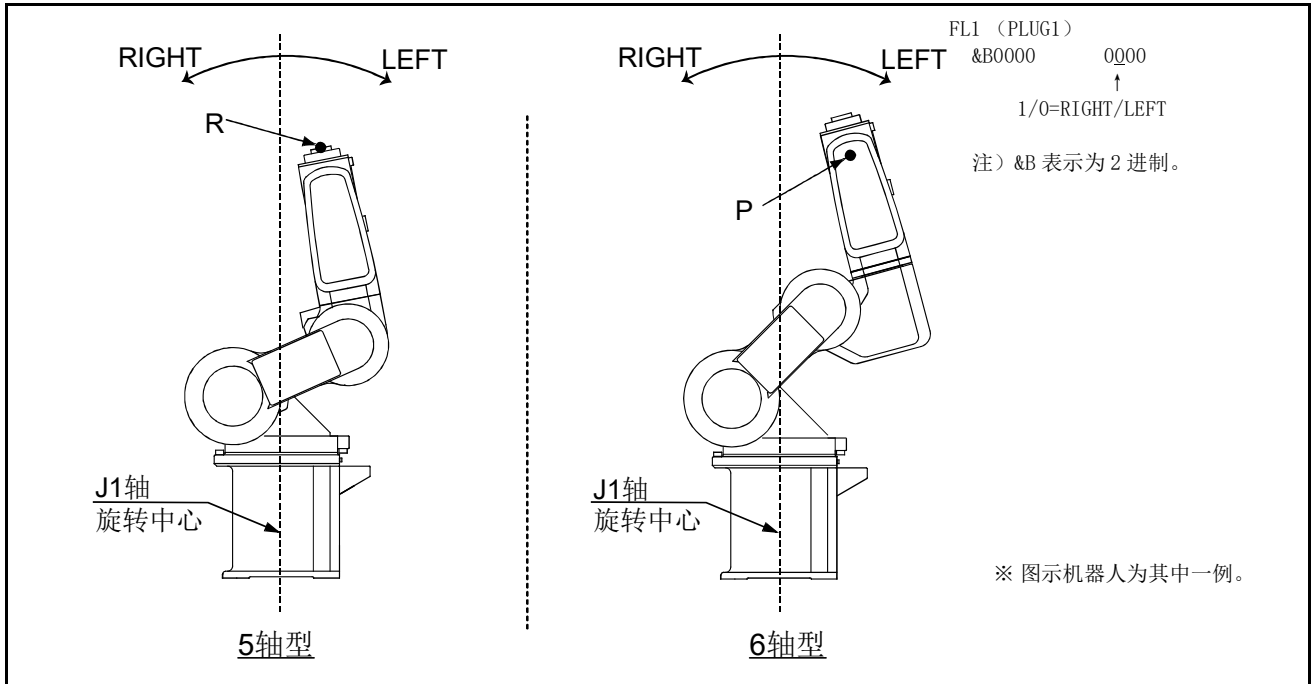


图 7-1: 构造 PLUG (RIGHT/LEFT)

##### (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)。

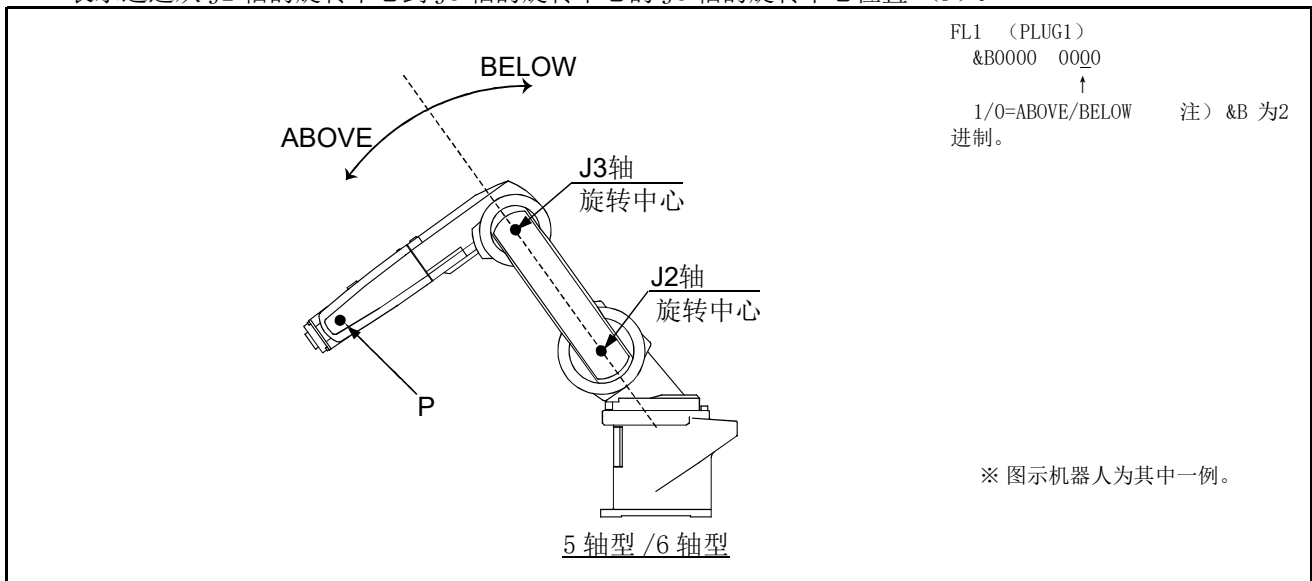


图 7-2: 构造 PLUG (ABOVE/BELOW)

## (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。

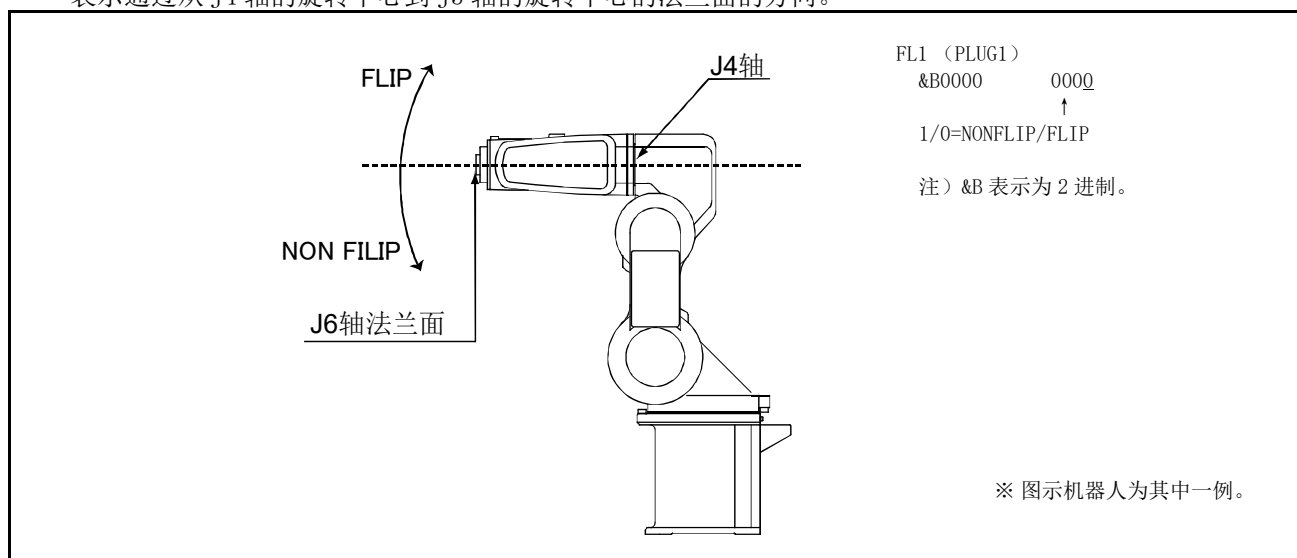


图 7-3: 构造 PLUG (NONFLIP/FLIP)

■ 水平多关节机器人时

## (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。



|                  |      |
|------------------|------|
| FL1 (PLUG1)      |      |
| &B0000           | 0000 |
|                  | ↑    |
| 1/0=NONFLIP/FLIP |      |

注) &B 表示为 2 进制。

※ 图示机器人为其中一例。

图 7-4: 构造 PLUG (RIGHT/LEFT)

## 7.2 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。

### 7.2.1 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。

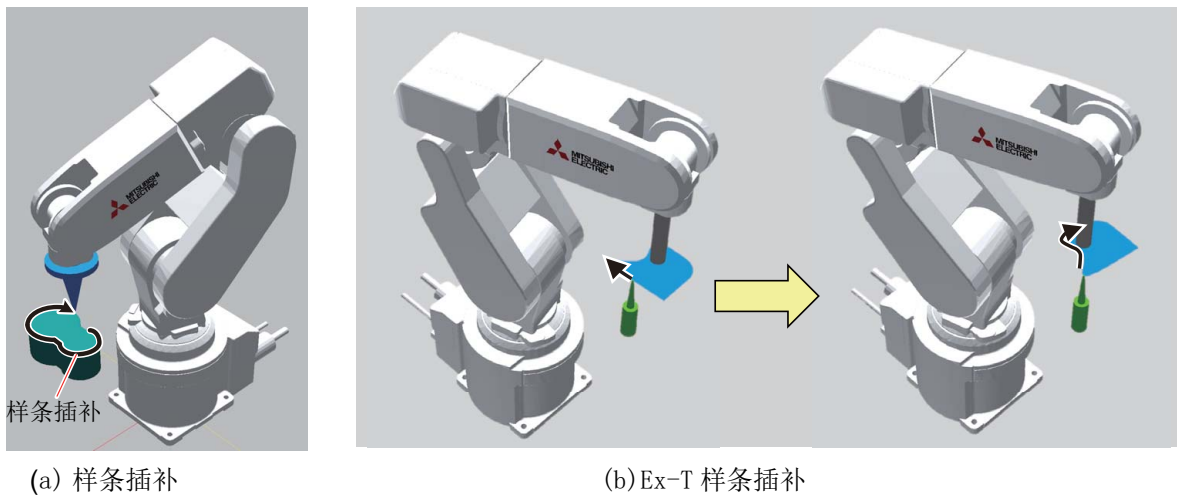


图 7-5：样条插补的概要

#### (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。

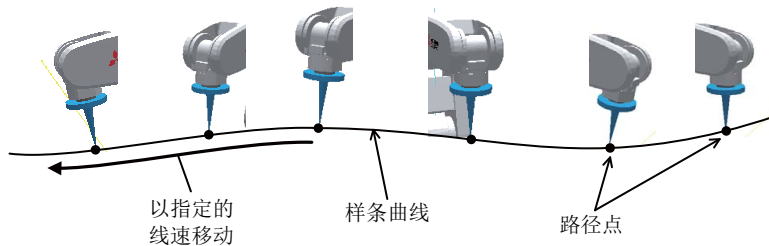


图 7-6：样条插补

- 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 ToolBox3 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。

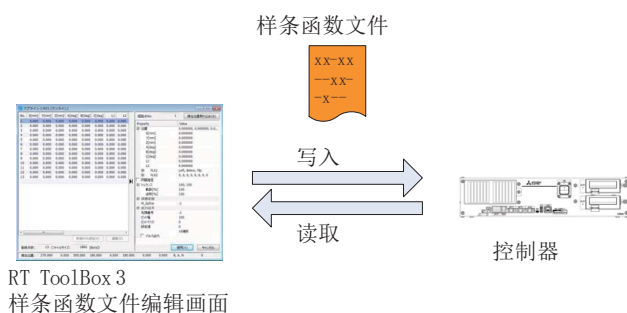


图 7-7：将路径点数据进行专用文件化

- 通过使其独立为专用的文件，可在多个机器人程序中共用样条插补，并且样条插补也可以简单地切换。

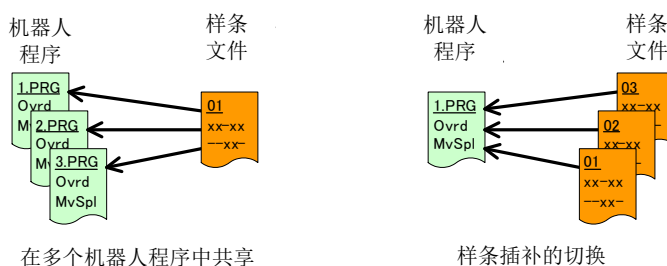


图 7-8：基于专用文件化的特征

- 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 ToolBox3。样条插补可以在表 7-1 所示的软件版本的设备中使用。

表 7-1：可以使用样条插补的软件版本

| 设备       | 功能        | 控制器        | RT ToolBox3  |
|----------|-----------|------------|--------------|
| 可使用的软件版本 | 样条插补      | Ver. A1 以上 | Ver. 1.00 以上 |
|          | Ex-T 样条插补 | Ver. A1 以上 | Ver. 1.00 以上 |

关于示教单元（R32TB/R56TB），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 7-2 所示。

表 7-2：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（直交坐标值）。               |
| 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。     |
| 样条文件编辑画面                   | RT ToolBox3 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 <sup>注 1)</sup> | RT ToolBox3 中配备的用于从 DXF 文件创建・保存样条文件的专用画面。 |

| 术语         | 说明                        |
|------------|---------------------------|
| 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。 |

注 1) RT ToolBox3 mini 版中无法使用。

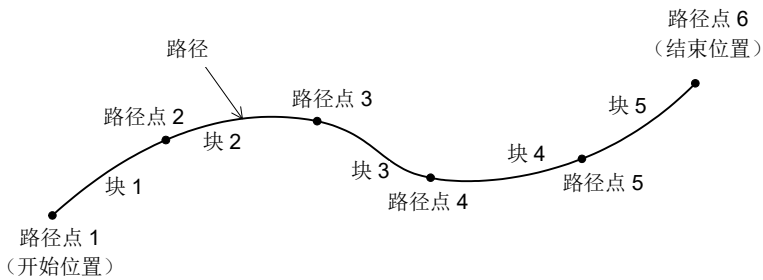


图 7-9: 样条插补的术语

## 7.2.2 规格

### (1) 基本规格

表 7-3: 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 VI 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、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 ToolBox3 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 ToolBox3 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（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 7-4: 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。                                                                                                                                                                                                                                                                                                                                                                       |      |    |       |              |        |              |         |               |        |                 |       |             |
| Oad1 指令     | 即使因 Oad1 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，无法切换工具转换数据 / 基本转换数据。                                                                                                                                                                                                                                                                                                                                                                                   |      |    |       |              |        |              |         |               |        |                 |       |             |
| 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="528 752 1147 954"> <thead> <tr> <th>状态变量</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>M_Ac1</td> <td>返回当前的加速时间比率。</td> </tr> <tr> <td>M_DAc1</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_Ac1 | 返回当前的加速时间比率。 | M_DAc1 | 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_Ac1       | 返回当前的加速时间比率。                                                                                                                                                                                                                                                                                                                                                                                                      |      |    |       |              |        |              |         |               |        |                 |       |             |
| M_DAc1      | 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 Ovrld 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。

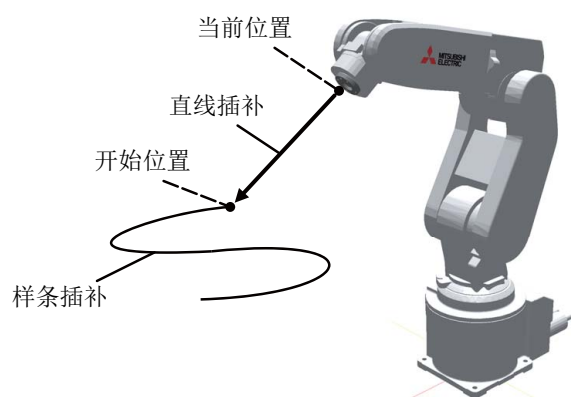


图 7-10: 当前位置与开始位置不不同时 (样条插补)

- 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 Ovrld 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。

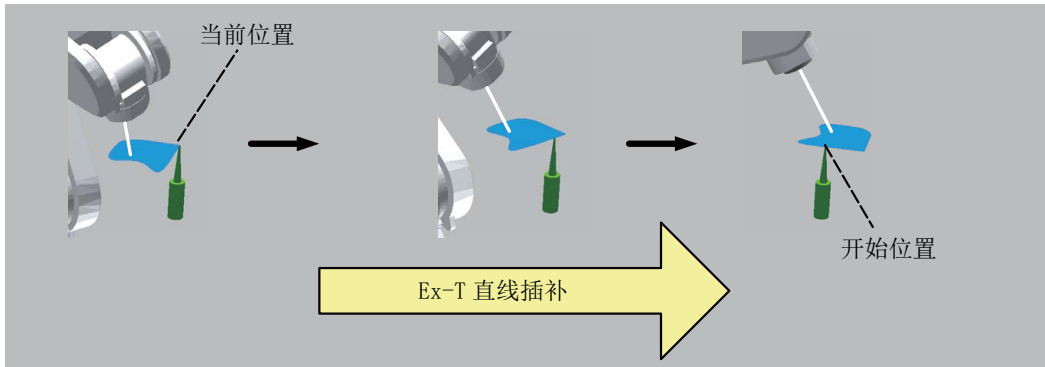


图 7-11: 当前位置与开始位置不相同 (Ex-T 样条插补)

- 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。

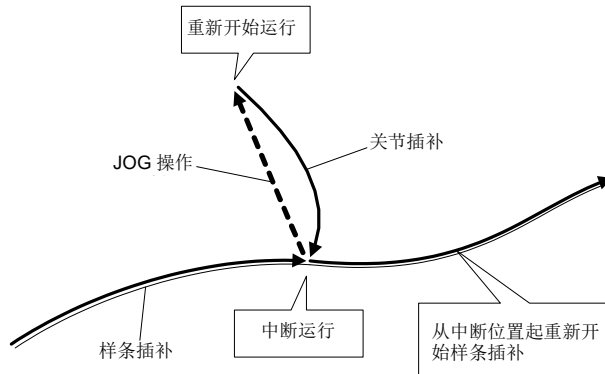


图 7-12: 中断 · 重新开始时的动作

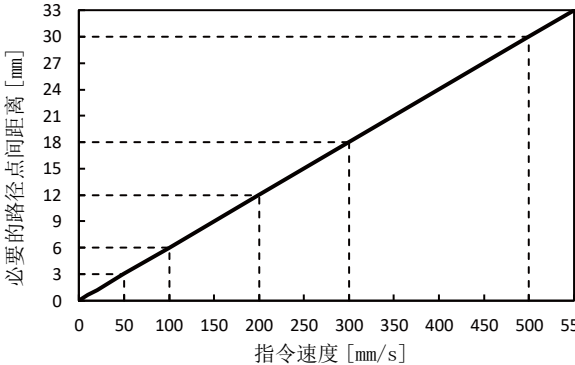
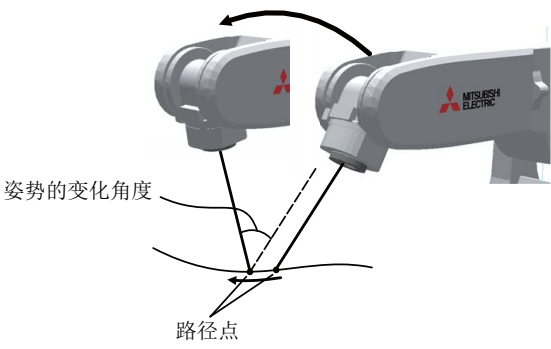
- 样条插补执行过程中通过停止输入等使运行中断后，如将中断执行的机器人程序读取至示教单元 · RT ToolBox3 的程序编辑画面，则样条插补的中断位置信息将被复位。因此，如从中断行开始重新运行，则机器人将向样条插补的开始位置移动之后再继续运行。



## (4) 路径点相关的检查

对各路径点进行表 7-5 中记载的检查。

表 7-5：路径点相关的检查

| 种类           | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |            |    |          |    |          |    |          |     |          |     |           |     |           |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------|----|----------|----|----------|----|----------|-----|----------|-----|-----------|-----|-----------|
| 路径点间的距离与指令速度 | <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。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>  <p>※ 从软件版本 A3a 开始追加了样条插补指令速度高速化功能。如果该功能有效，则可以对同一路径点间的距离，指定比现在更大的指令速度。详细内容，请参照第 722 页的“7.2.10 样条插补指令速度高速化”。</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>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |            |    |          |    |          |    |          |     |          |     |           |     |           |

### 7.2.3 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。将设定值设得越小，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。

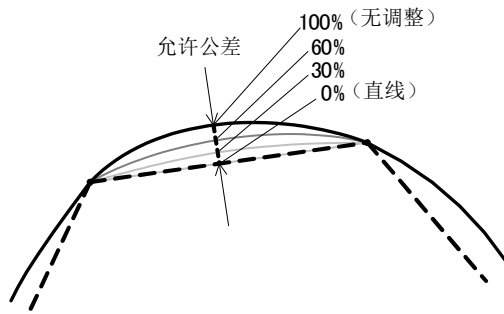
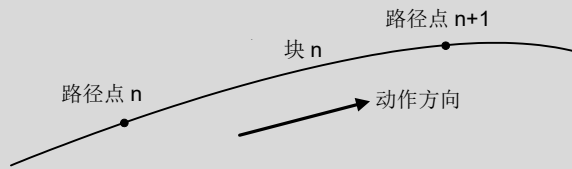


图 7-13: 允许公差

#### ◇◆◇ 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。

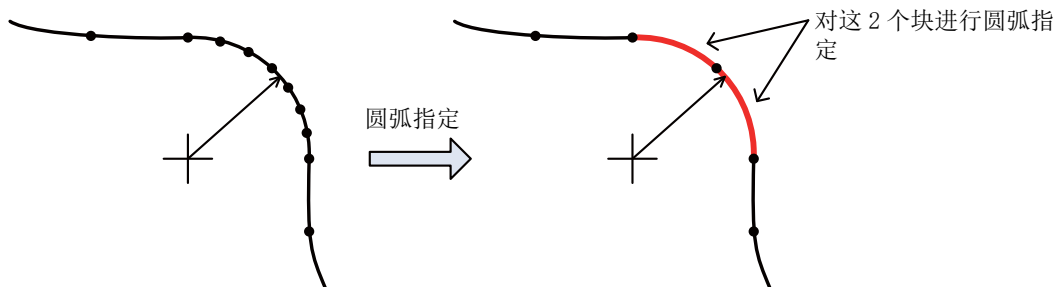


图 7-14: 圆弧指定（样条文件版本 02）

※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。

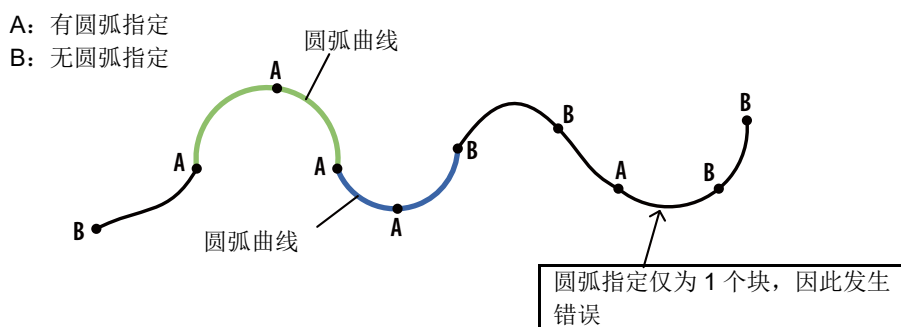


图 7-15：因圆弧指定而导致发生错误的示例（样条文件版本 02）

#### ◇◆◇样条文件版本 02 ◇◆◇

对应样条文件版本 02 的软件版本如下所示。

控制器：Ver. A1 以上

RT ToolBox3：Ver. 1.00 以上

#### 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。

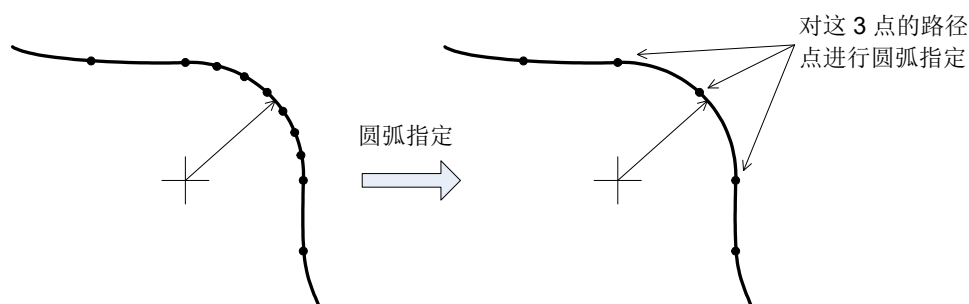


图 7-16：圆弧指定（样条文件版本 01）

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。

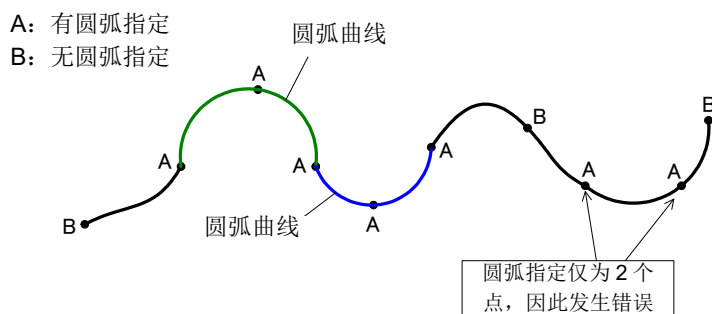
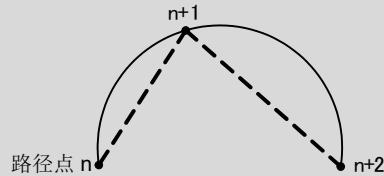


图 7-17：因圆弧指定而导致发生错误的示例（样条文件版本 01）

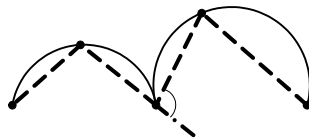
## ◇◆◇圆弧指定与其他的路径补偿功能的关系◆◆◇

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。

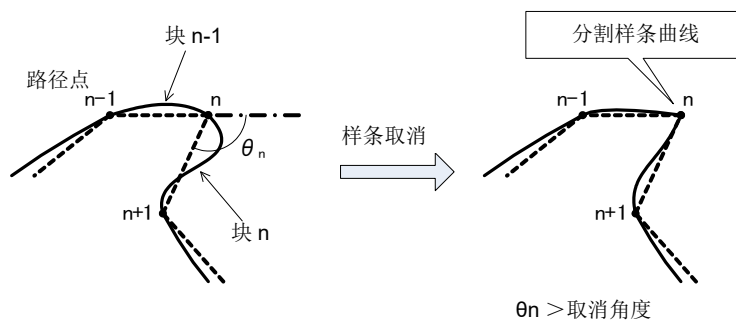


图 7-18：样条取消

例如，希望通过图 7-18 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。

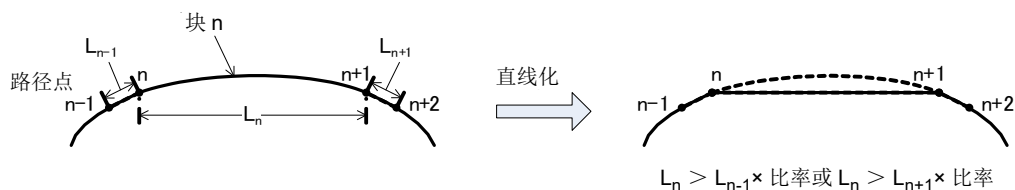


图 7-19: 块长比率导致的直线化

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 7-6 所示，动作模式有 2 种。

表 7-6: 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。

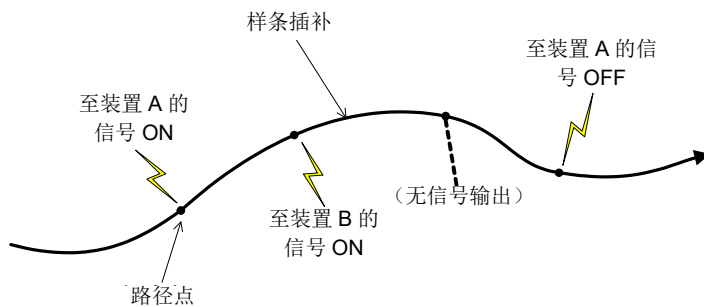


图 7-20: 信号输出

要进行信号输出，需要设定表 7-7 的项目作为路径点的数据。

表 7-7: 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> 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。 | 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> 初始值为 0。                                                                                                                             | 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 7-8 所示。

表 7-8: 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 P               |     |     |     |     |     | ○   | ●   | ○ |

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。

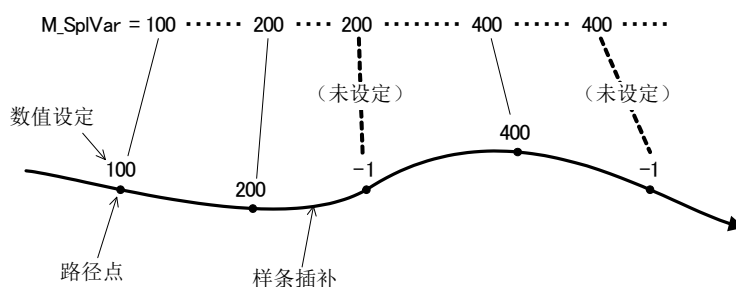


图 7-21: 数值设定

#### (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 7-22 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。

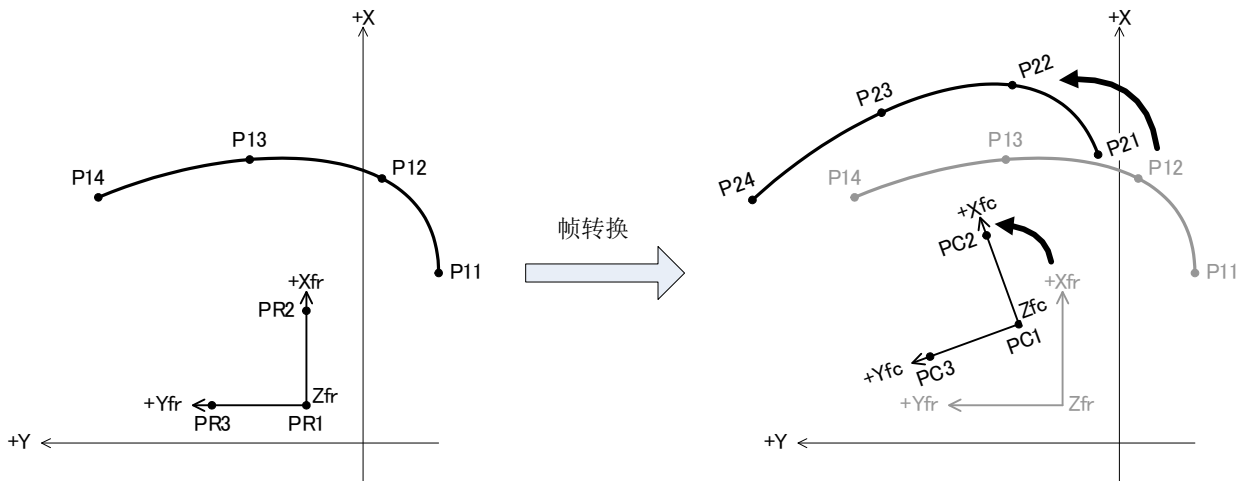


图 7-22: 帧转换 (样条插补)

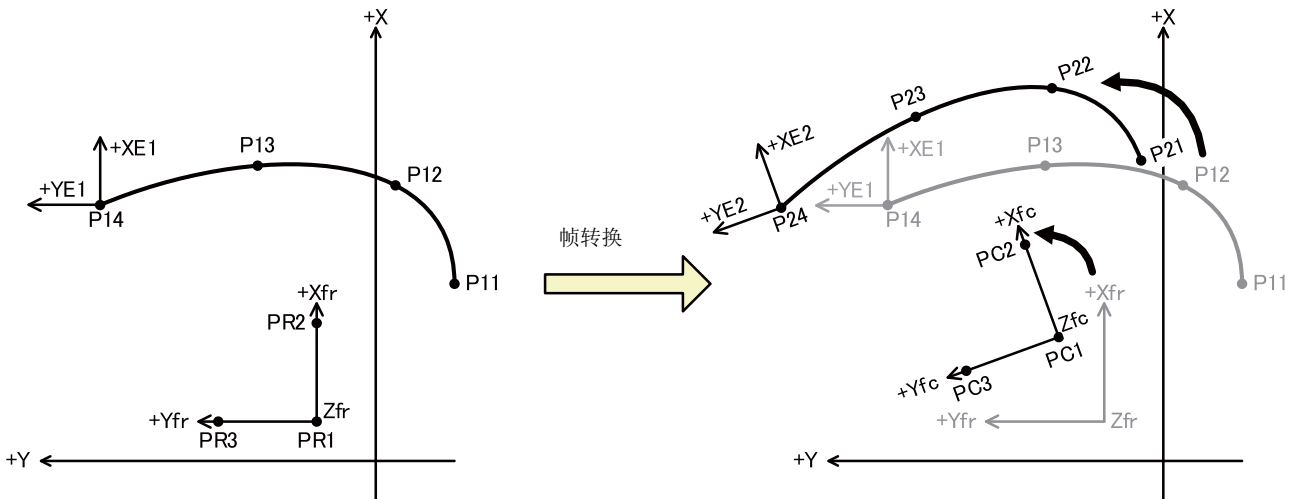


图 7-23: 帧转换 (Ex-T 样条插补)

■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 7-22 及图 7-23 的 PR1、PC1

X 轴上的位置..... 相当于图 7-22 及图 7-23 的 PR2、PC2

XY 平面的 +Y 方向的位置... 相当于图 7-22 及图 7-23 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 7-9：帧转换的执行方法

| 执行方法              | 说明                                                         | 使用案例                                                                                                            |
|-------------------|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| 通过 RT ToolBox3 执行 | 在 RT ToolBox3 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。           | <ul style="list-style-type: none"> <li>通过 JOG 操作等对转换后的路径点数据进行个别补偿时</li> <li>希望将转换后的路径点数据用作其他路径的原本数据时</li> </ul> |
| 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。                        | <ul style="list-style-type: none"> <li>希望将图纸上的坐标值留作路径点的坐标值时</li> <li>坐标系固定为 1 个时</li> </ul>                     |
| 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。 | <ul style="list-style-type: none"> <li>希望对 1 个路径适用多个坐标系时</li> <li>每次均使用传感器类对路径的偏移量进行补偿时</li> </ul>              |

## 7.2.4 作业步骤

至执行样条插补为止的作业步骤如下所示。

表 7-10：作业步骤

| No. | 步骤                     | 作业内容                                                                                                                                                                                         |
|-----|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | 样条文件的创建<br>(7.2.5 章)   | 使用 RT ToolBox3 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>(7.2.6 章)  | 创建包含样条插补动作的机器人程序。                                                                                                                                                                            |
| 3   | 动作的确认<br>(7.2.7 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>(7.2.8 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>(7.2.9 章)      | 使用 RT ToolBox3 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 ToolBox3 mini 版中无法使用。

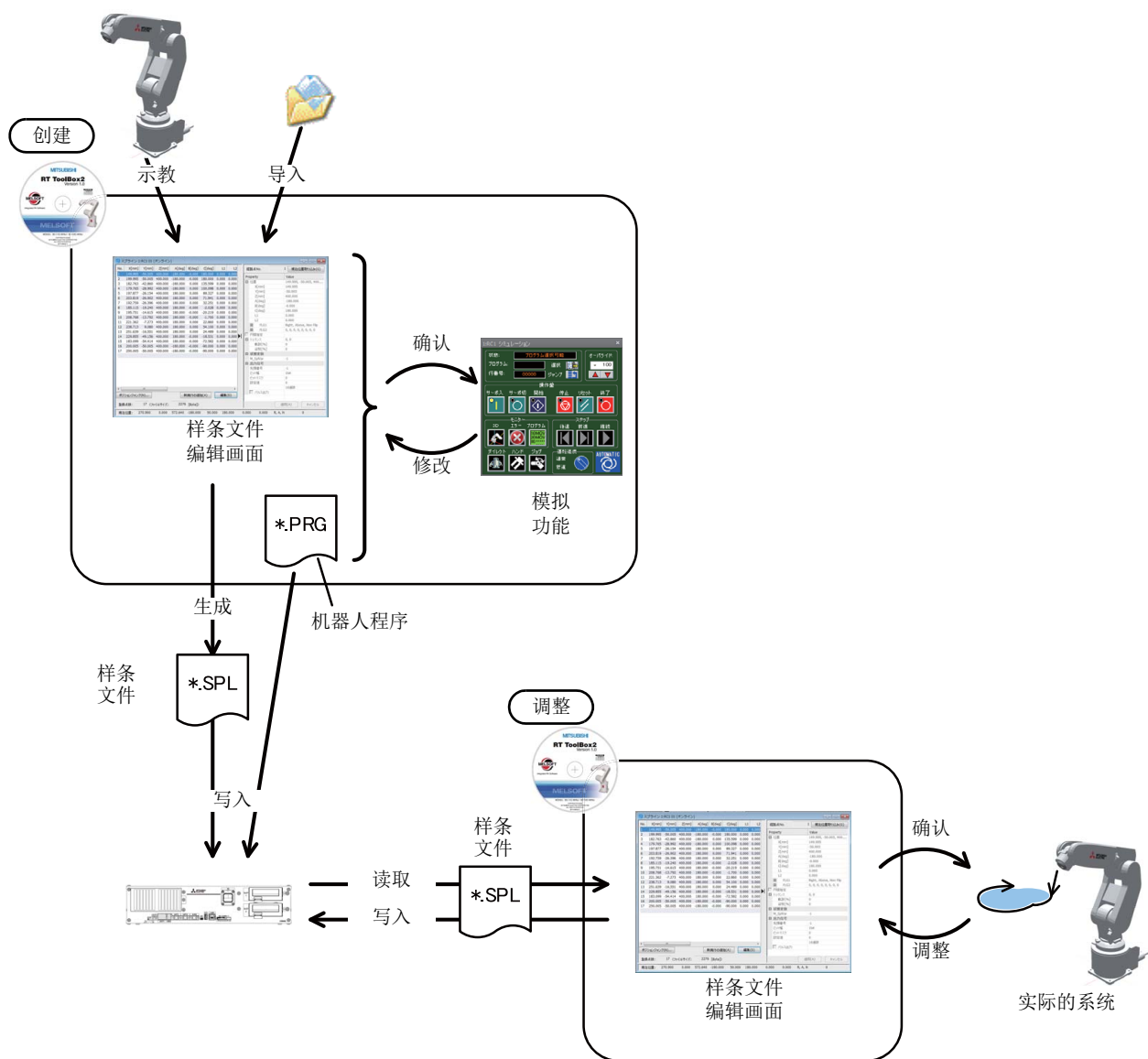


图 7-24: 作业内容的示意图

### 7.2.5 样条文件的创建

使用 RT ToolBox3 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。)

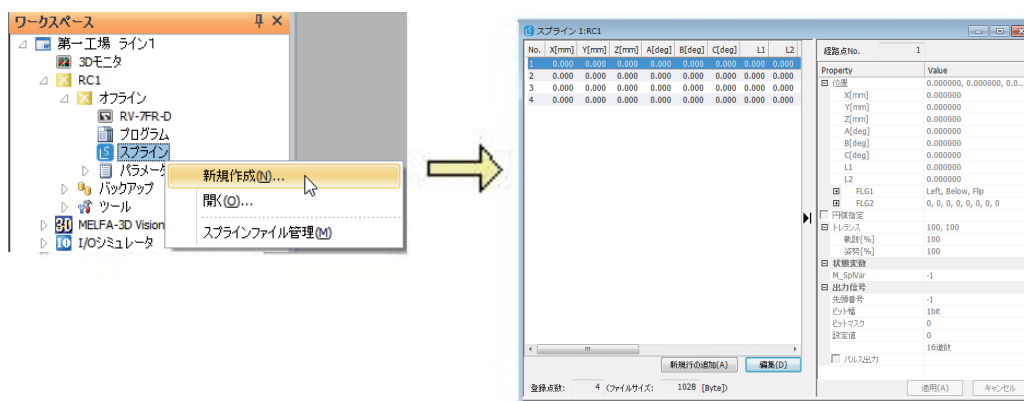


图 7-25：在计算机内新建

##### ■ 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。

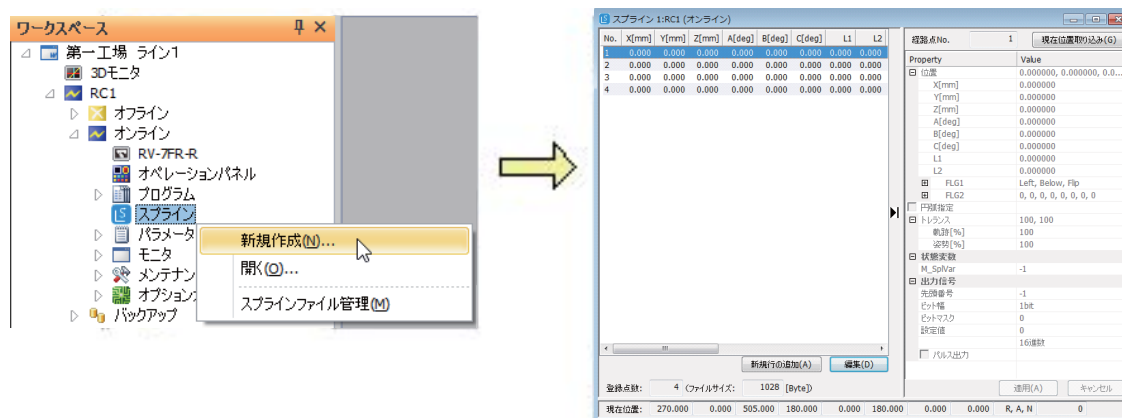


图 7-26：在控制器内新建

#### ◇◆◇ 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  
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。

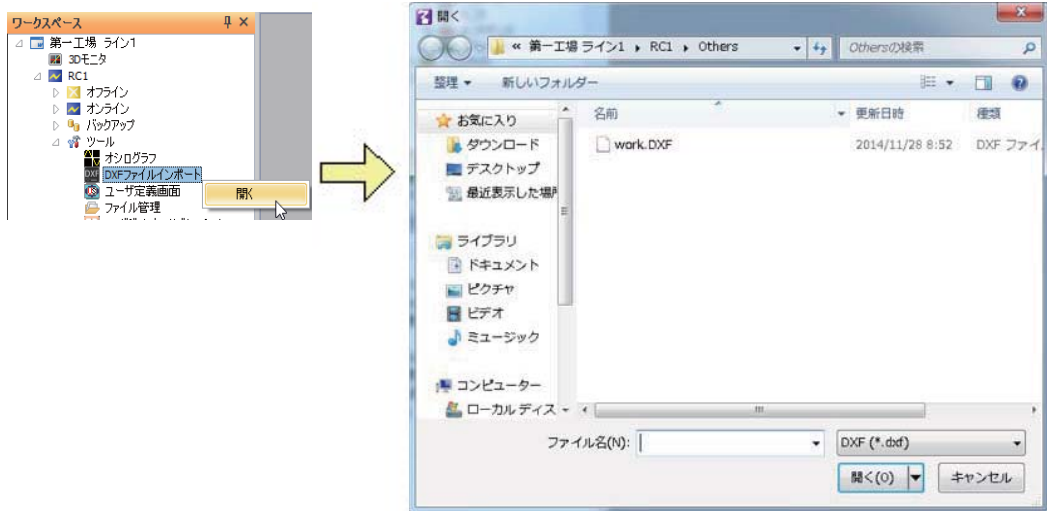


图 7-27：文件选择画面

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 7-28：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。

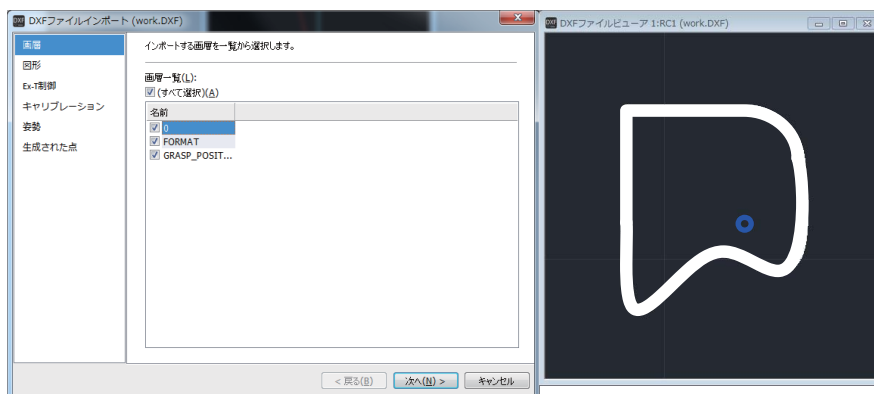


图 7-29: 画层选择画面

- 3) 图形选择  
在 [ 图形 ] 页中, 选择要导入的图形。

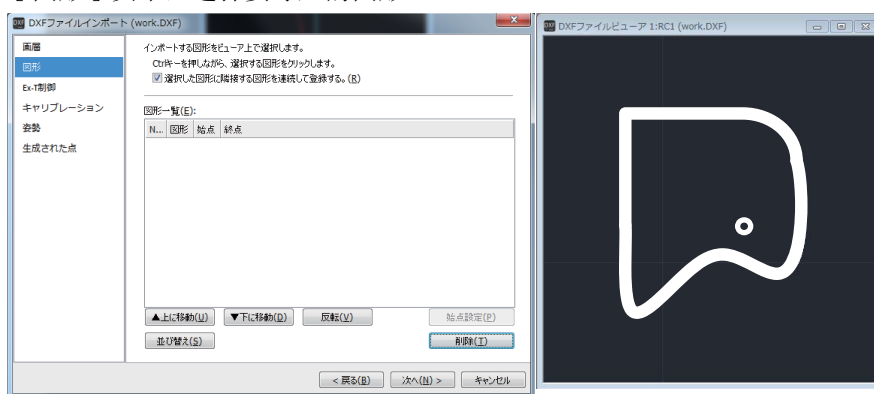


图 7-30: 图形选择画面

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。

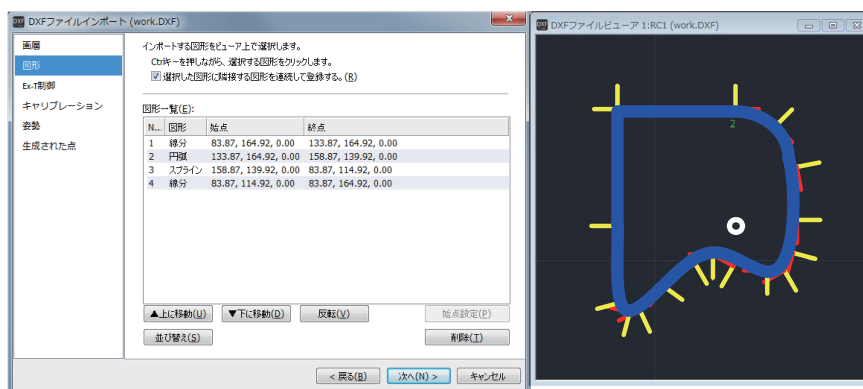


图 7-31: 图形选择

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 7-32: 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。

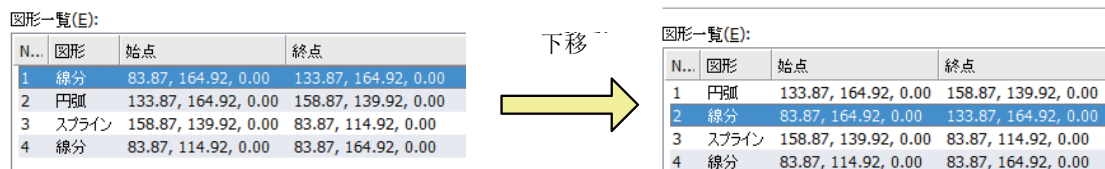


图 7-33: 图形选择 (下移)

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。

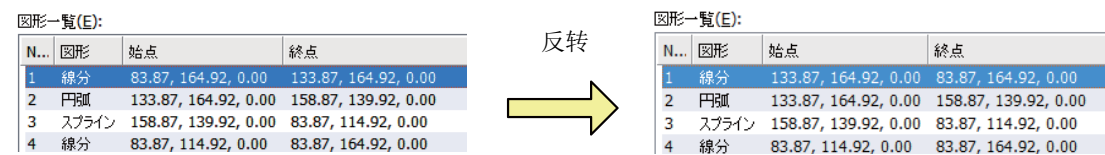


图 7-34: 图形选择 (反转)

点击 [ 排序 ] 按钮, 即显示确认对话框。

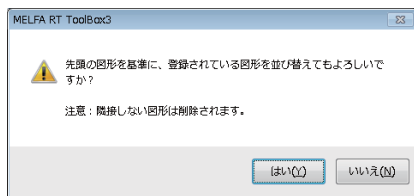


图 7-35: 排序确认对话框

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。

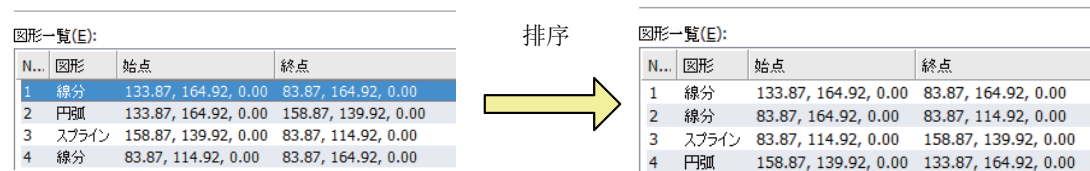


图 7-36: 图形选择 (排序)

## 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。

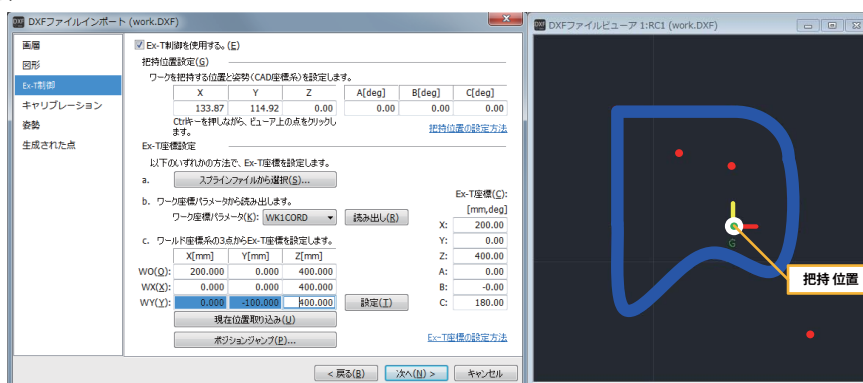


图 7-37: Ex-T 控制设定画面

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。

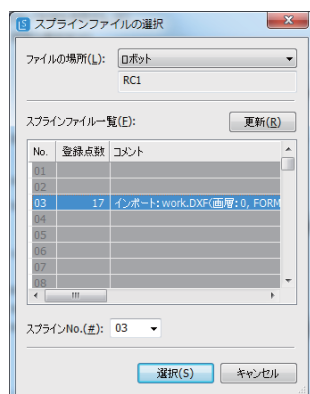


图 7-38: 样条文件的选择画面

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 7-39：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。

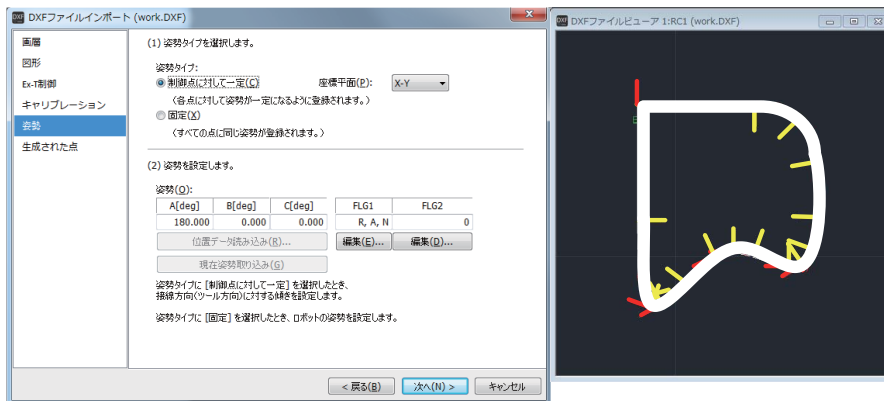


图 7-40：姿势登录画面（对控制点为固定）



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。

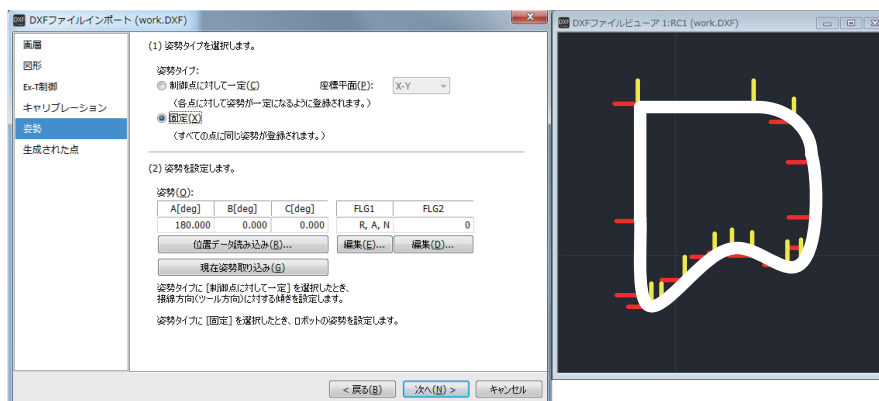


图 7-41：姿势登录画面（固定）

#### 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。

[ 将姿势公差设为 0[%]。 ] 勾选复选框后，各路径点的姿势公差将在 0% 的状态下输出至样条文件。以从 DXF 文件导入的姿势动作时，应勾选。（初始设定为已勾选状态。）

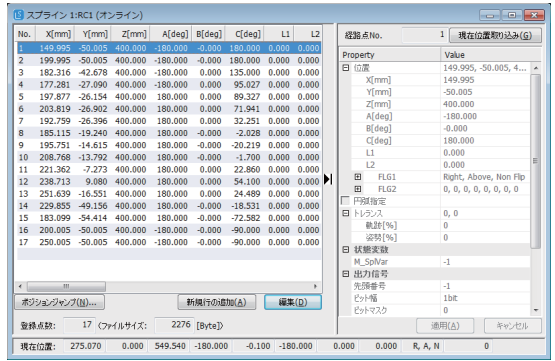
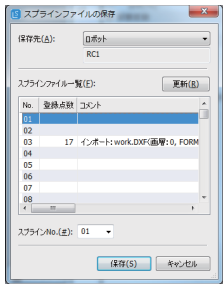
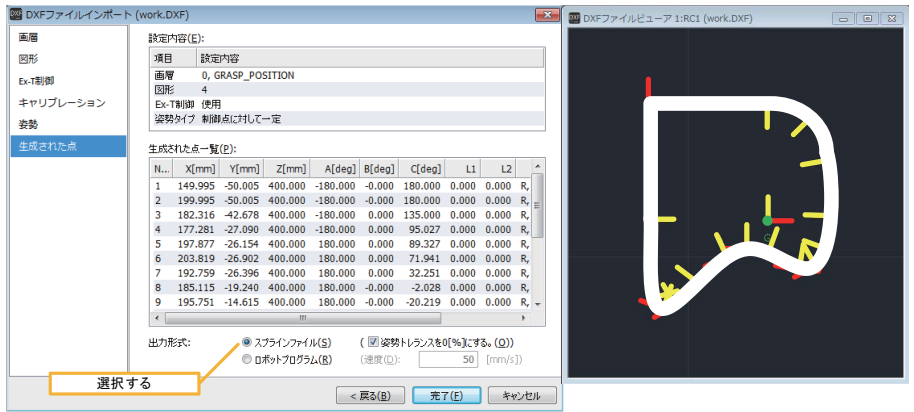


图 7-42: 生成的点画面

## (3) 用机器人语言新建

## ■用 SplWrt 指令新建样条文件

可使用 SplWrt（样条写入）指令创建样条文件。指令的详细内容请参照 [SplWrt \(Spline Write\)](#)。

在读取的文件中，对表 7-15 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 7-75 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 SplFWrt 指令编辑现有的样条文件

可以使用 SplFWrt（样条帧写入）指令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 [SplFWrt \(Spline Frame Write\)](#)。

## (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。）

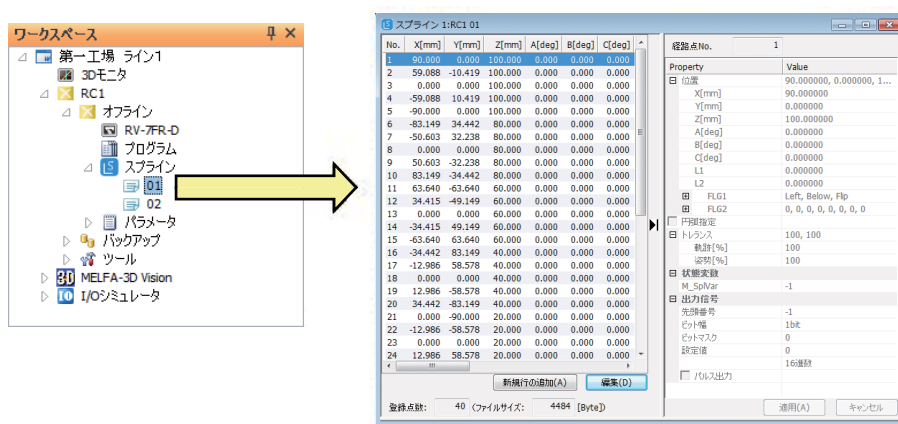


图 7-43: 打开保存在计算机中的样条文件

### ■ 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。)

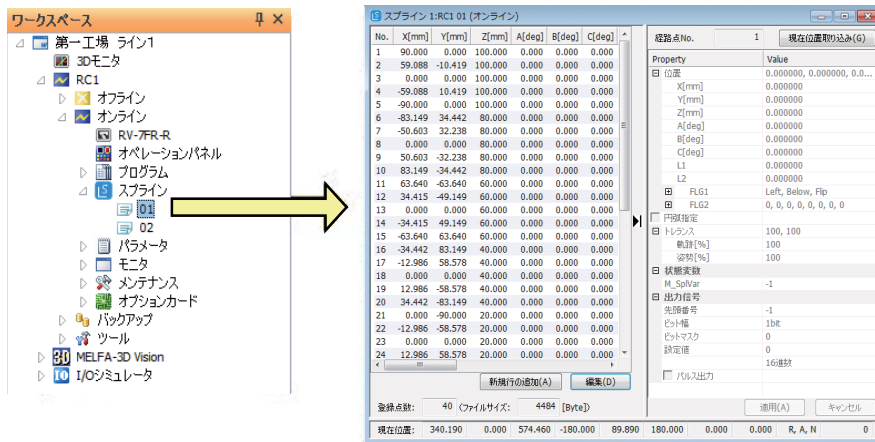


图 7-44: 打开控制器内的样条文件

### ■ 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。)

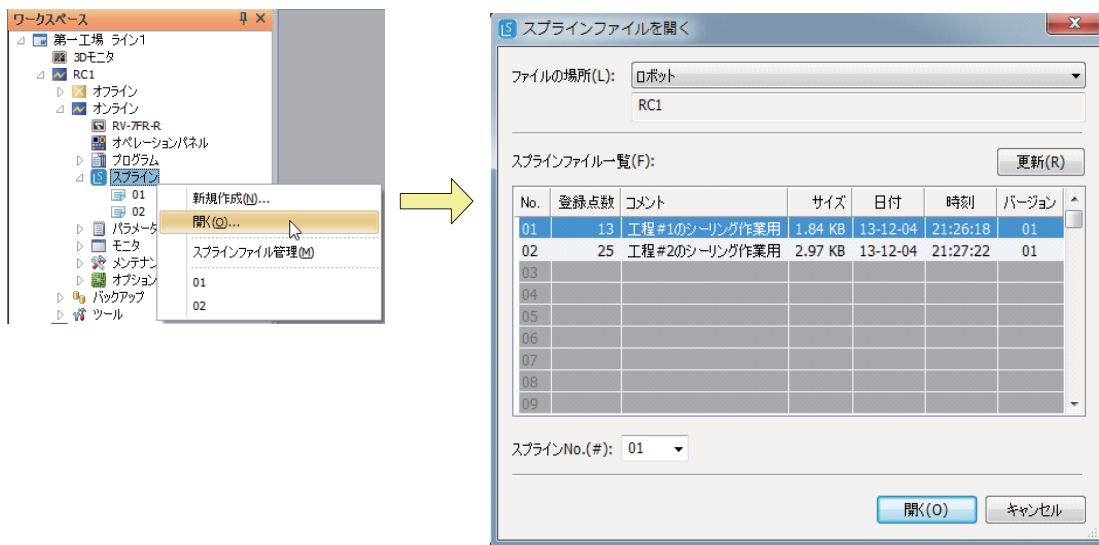


图 7-45: 从一览中选择并打开样条文件

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。

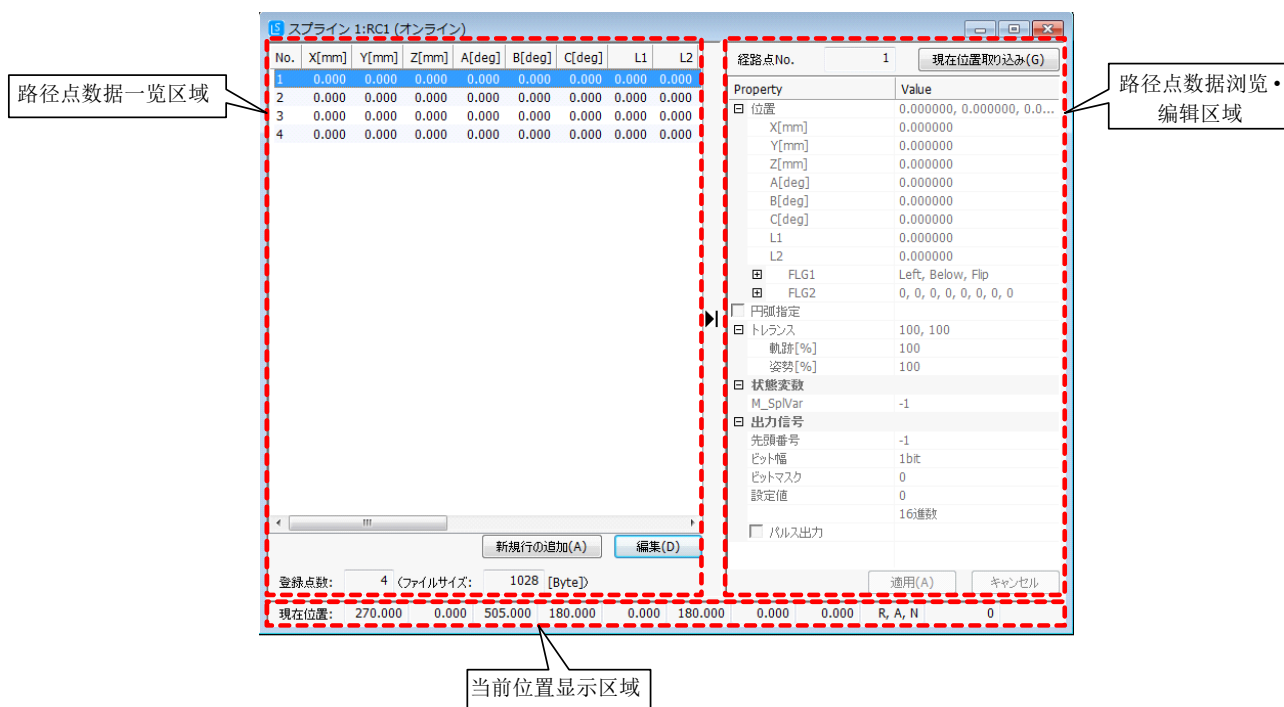


图 7-46：样条文件编辑画面

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。

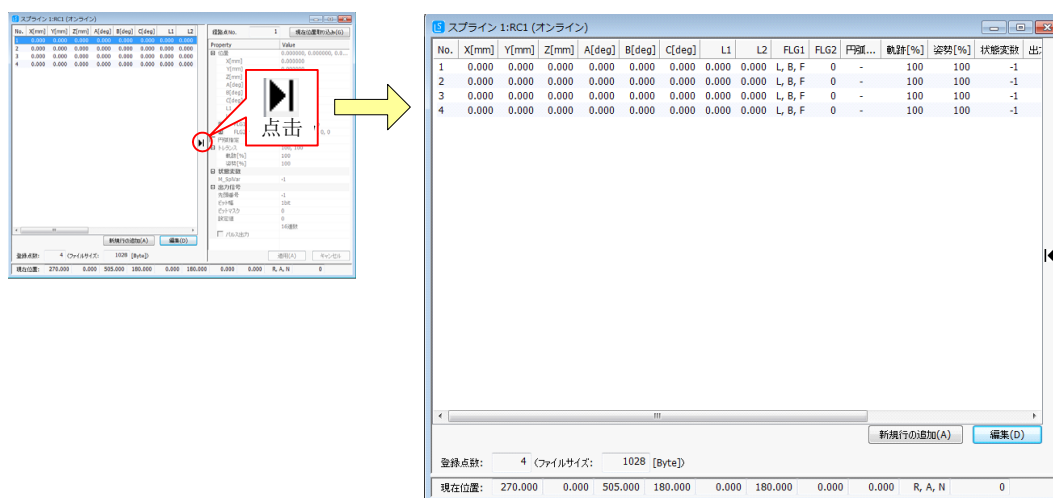


图 7-47：路径点数据一览区域的显示范围扩大后的状态

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 7-48：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 687 页的“■ 路径点数据的编辑 · 登录”。)

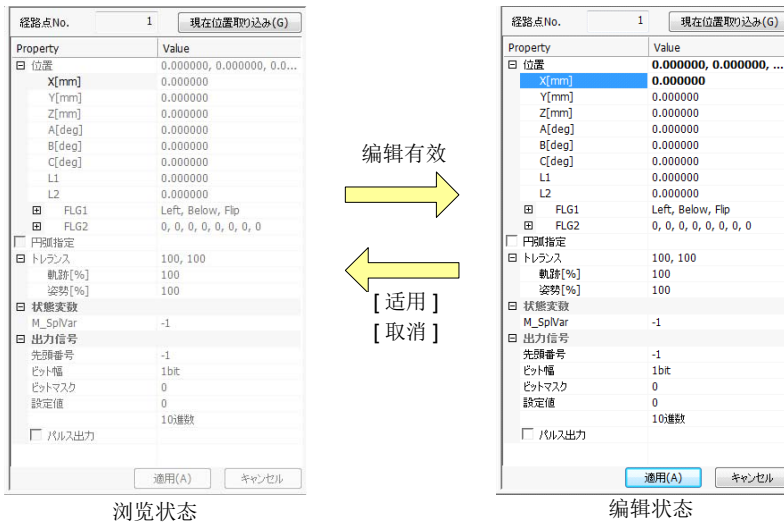


图 7-49：路径点数据浏览 · 编辑区域的浏览状态与编辑状态的切换

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。

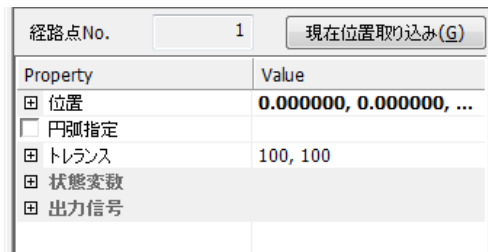


图 7-50：存储路径点数据的状态

■ 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 ( 直交坐标系 ) 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) 样条文件编辑时的功能区

样条文件编辑时，功能区中将增加“文件（F）”・“编辑（E）”・“工具（T）”标签。



图 7-51：样条文件编辑时的功能区

各个菜单的内容如下所示。

表 7-11：增加的菜单的内容

| 标签项目   | 说明                                     | 参照页码 |
|--------|----------------------------------------|------|
| 文件     |                                        |      |
| 打开     | 显示用于打开样条文件的专用画面。                       | 678  |
| 关闭     | 关闭已激活的样条文件编辑画面。                        | -    |
| 保存     | 保存编辑中的样条文件。                            | 693  |
| 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。                | 692  |
| 导入     | 从 CSV/MXT <sup>注1)</sup> 格式的文件获取路径点数据。 | 703  |
| 导出     | 向 CSV 格式的文件写出路径点数据。                    | 705  |
| 样条文件管理 | 打开样条文件管理画面。                            | 696  |

| 标签项目                                                                               | 说明                                           | 参照页码 |
|------------------------------------------------------------------------------------|----------------------------------------------|------|
| 编辑                                                                                 |                                              |      |
|  |                                              |      |
| 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。 | 706  |
| 复制                                                                                 | 将所选择的路径点数据存储为要复制的数据。                         | 706  |
| 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。        | 706  |
| 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。                   | 707  |
| 恢复                                                                                 | 再次反映通过 [ 撤销 ] 取消的内容。                         | 707  |
| 新增行的追加                                                                             | 在路径点数据的末尾追加新的路径点数据。                          | 686  |
| 追加点击位置                                                                             | 勾选后，在 3D 监视画面最下行追加已点击面的位置。                   | -    |
| 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。                     | 686  |
| 已复制路径点的插入                                                                          | 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。            | 707  |
| 全选                                                                                 | 将全部的路径点数据设为已选状态。                             | -    |
| 删除                                                                                 | 删除所选择的路径点数据。                                 | 688  |
| 编辑                                                                                 | 将所选择的路径点数据设为可编辑状态。                           | 687  |
| 批量编辑 - FLG1 结构标志                                                                   | 对所选择的路径点的 FLG1 的值进行批量编辑。                     | 688  |
| 批量编辑 - FLG2 多旋转数据                                                                  | 对所选择的路径点的 FLG2 的值进行批量编辑。                     | 689  |
| 批量编辑 - 公差量                                                                         | 对所选择的路径点的公差量进行批量编辑。                          | 689  |
| 批量编辑 - 圆弧指定                                                                        | 切换所选择的路径点的圆弧指定的 ON/OFF。                      | 689  |
| 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。            | 691  |
| 复制 - 位置数据                                                                          | 将所选择的位置数据存储为要复制的数据。                          | 707  |
| 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。                         | 707  |



| 标签项目      | 说明                                                                                 | 参照页码 |
|-----------|------------------------------------------------------------------------------------|------|
| 工具        |  |      |
| 跳位        | 显示跳位功能的专用画面。                                                                       | 720  |
| 插补设定      | 显示样条文件的插补设定画面。                                                                     | 690  |
| Ex-T 控制设定 | 显示样条文件的 Ex-T 控制设定画面。                                                               | 691  |
| 位置补偿      | 显示位置补偿功能的专用画面。                                                                     | 715  |
| 帧转换       | 显示帧转换功能的专用画面。                                                                      | 717  |
| 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。                                         | 709  |
| 路径点检查     | 显示路径点检查功能的专用画面。                                                                    | 708  |
| 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。                                             | 709  |
| 样条曲线      | 显示编辑中的样条文件的样条曲线。                                                                   | 708  |
| 程序转换      | 将编辑中的样条文件转换为机器人程序。                                                                 | 709  |
| 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。                                                        | 711  |

注 1) MXT 文件是指具有从 RT ToolBox3 Pro 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 7-12: 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。<br><table border="1" data-bbox="539 943 1002 1115"> <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 | 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。<br><table border="1" data-bbox="539 1196 1002 1368"> <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 | 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 7-12 的 “新增时的初始值” 所示的值。

(点击功能区 [ 编辑 ] → [ 新增行 ] 也可以增加。)

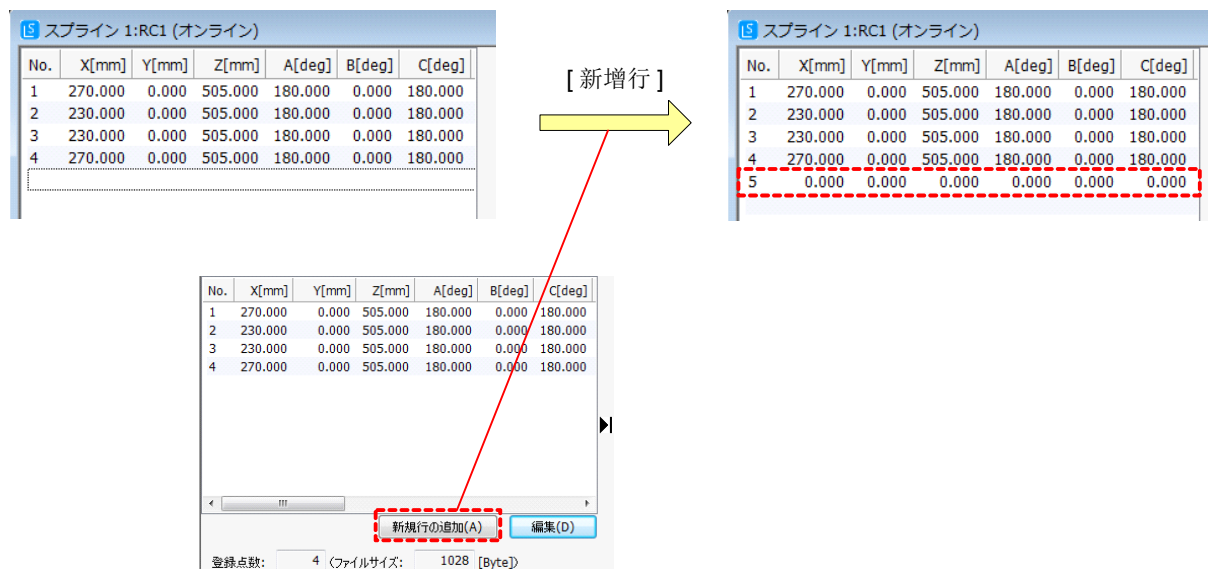


图 7-52: 路径点数据的增加

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。

(点击功能区 [ 编辑 ] → [ 插入到所选行的前面 ] 也可以插入。)

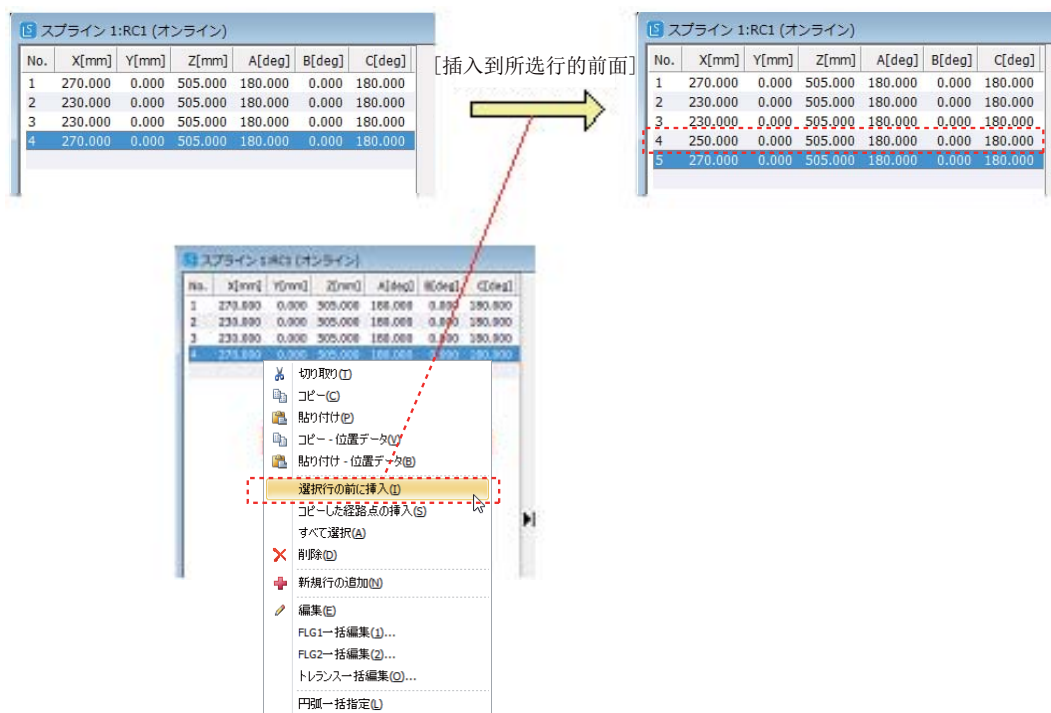


图 7-53: 路径点数据的插入

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 7-13 所示的值。

表 7-13: 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 7-12 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 7-12 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、功能区 [ 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。

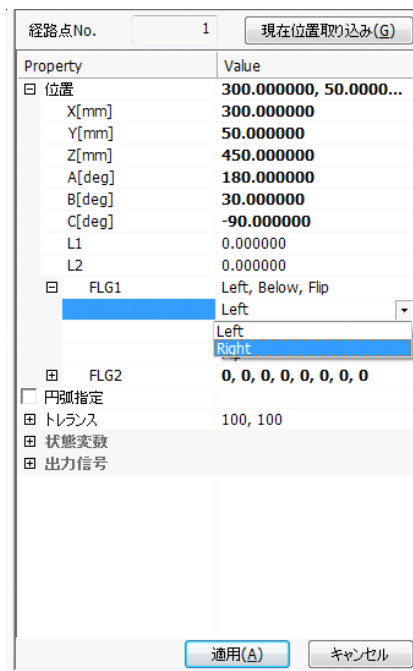


图 7-54: 路径点数据的编辑

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。

(点击功能区 [编辑] → [删除] 也可以删除。)

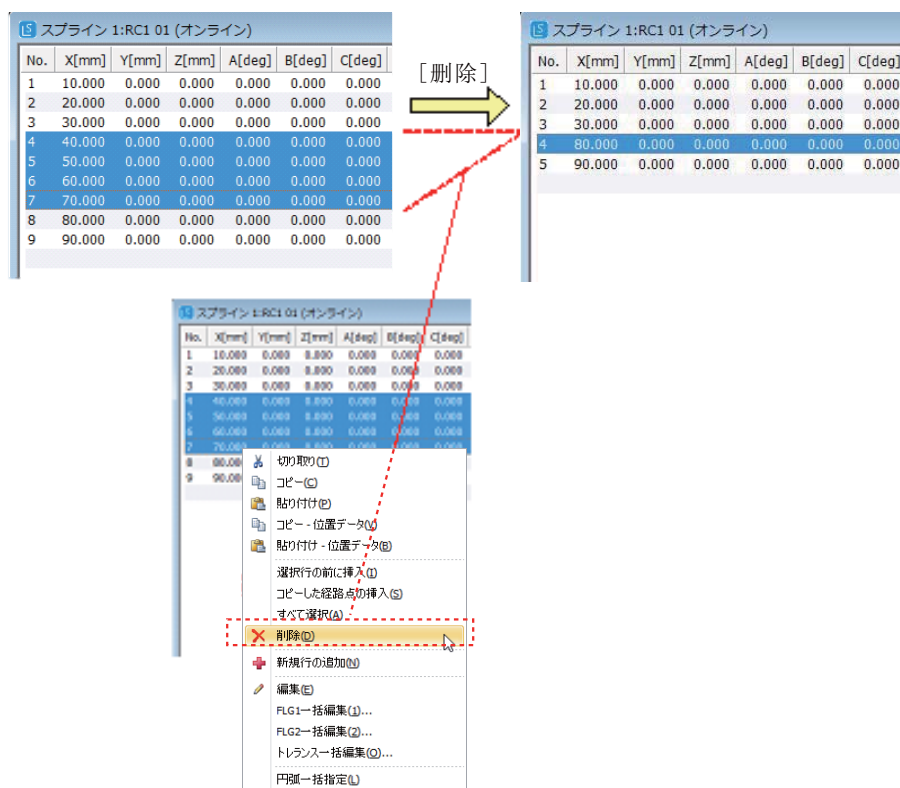


图 7-55：路径点数据的删除

### ■ 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。

(点击功能区 [编辑] → [批量编辑] → [FLG1 结构标签] 也可以显示。)

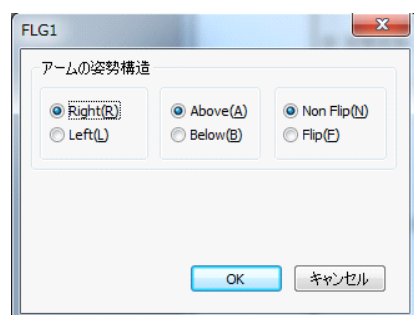


图 7-56：FLG1 批量编辑画面

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。

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。

（点击功能区 [编辑] → [批量编辑] → [FLG2 多转数据] 也可以显示。）

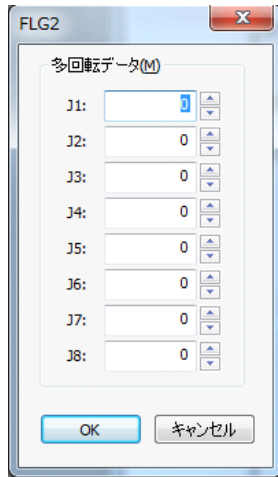


图 7-57：FLG2 批量编辑画面

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。

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。（点击功能区 [编辑] → [批量编辑] → 公差量 也可以显示。）

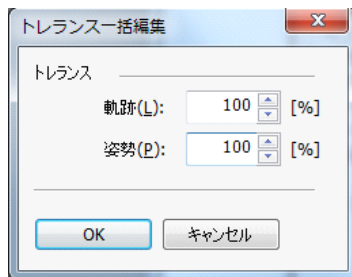


图 7-58：允许公差的批量编辑画面

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。

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。（点击功能区 [编辑] → [批量编辑] → [圆弧指定] 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    | -     | 100   | 100   | -1   |     |
| 3   | 182.763 | -42.860 | 400.000 | -180.000 | 0.000  | 135.599 | 0.000 | 0.000 | R, A, N | 0    | -     | 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 7-59：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”。

点击功能区 [ 工具 ] → [ 插补设定 ], 已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(Q): 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 7-60：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 7-14 所示。

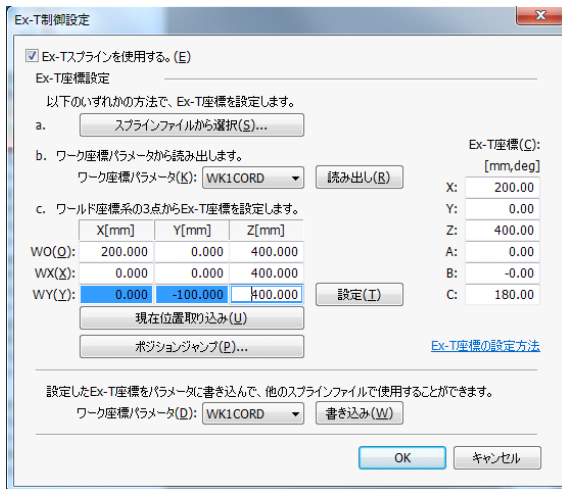
表 7-14：插补设定项目的设定范围

| 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    |

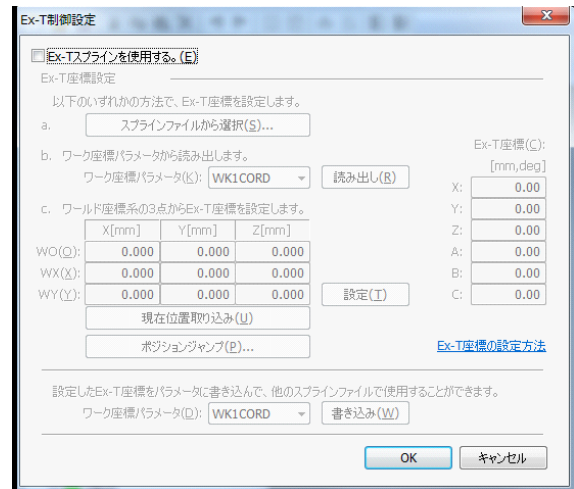
### ■ Ex-T 控制设定

进行样条插补的“Ex-T 控制设定”。

点击功能区 [ 工具 ] → [ Ex-T 控制设定 ], 已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 7-61: 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。

点击功能区 [ 编辑 ] → [ 属性 ], 已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。

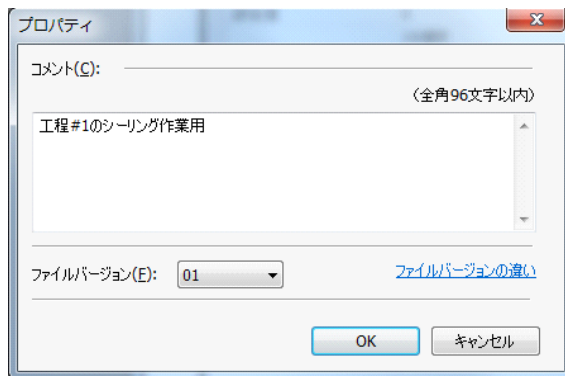


图 7-62: 属性画面

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。

点击功能区 [ 编辑 ] → [ 属性 ]，已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。

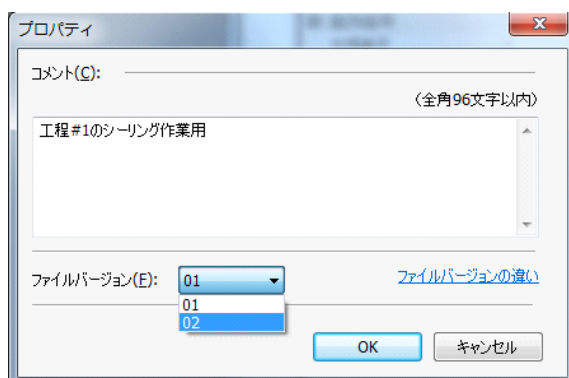


图 7-63：属性画面

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 661 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。

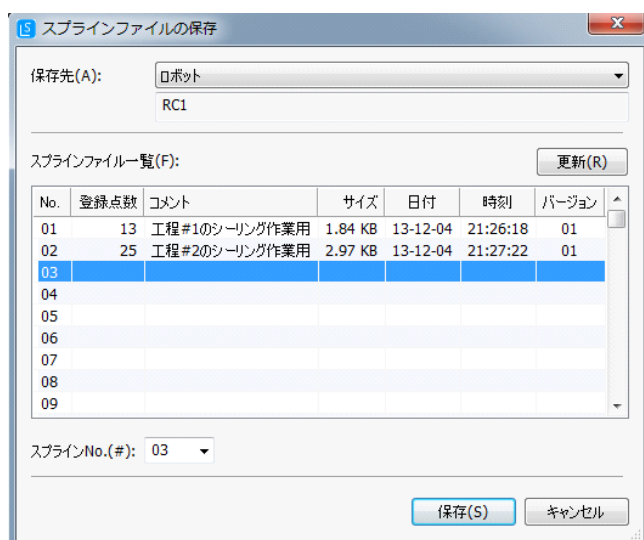


图 7-64：样条文件的保存画面

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 ToolBox3 的文件管理画面及 Microsoft Windows 的资源管理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。

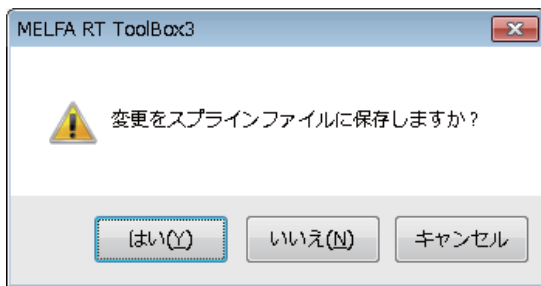


图 7-65: 保存确认对话框

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 7-66：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。

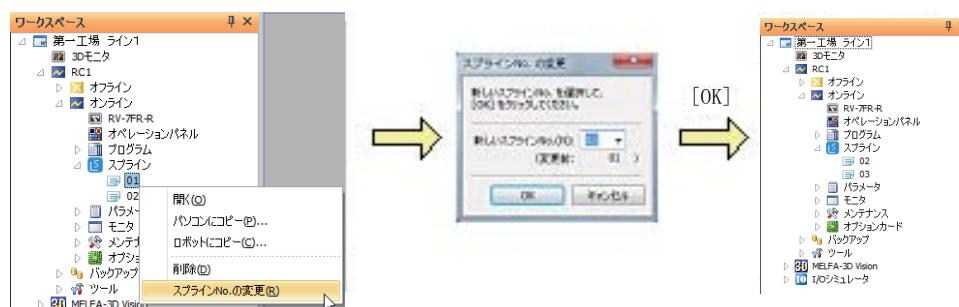


图 7-67：样条号的更改

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

可以在工程树上复制样条文件。

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。

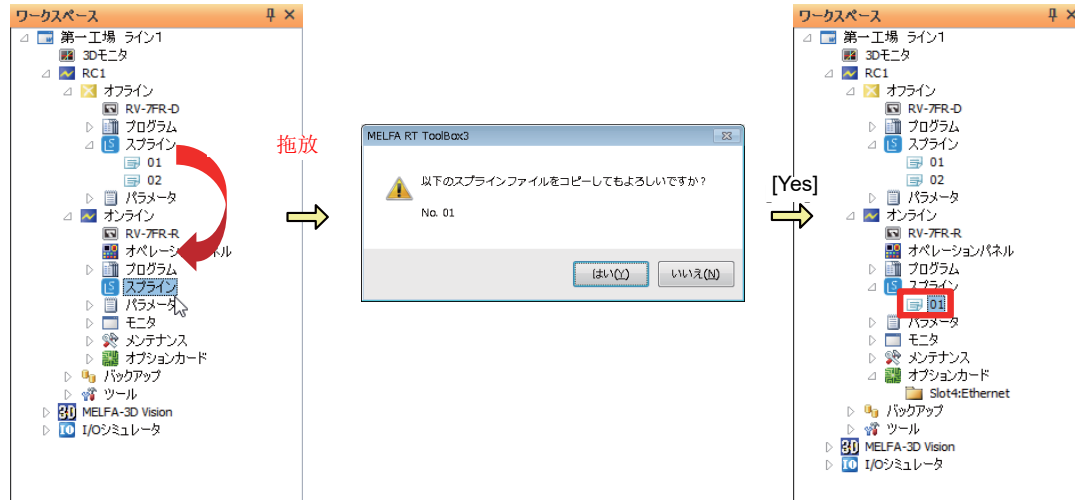


图 7-68：样条文件的复制

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。

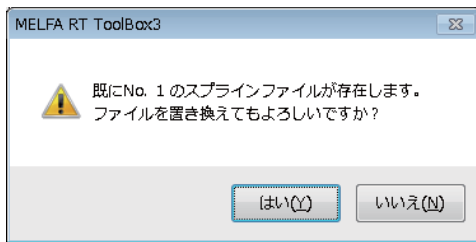


图 7-69：覆盖确认对话框

## (13) 样条文件管理

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。(功能区 [ 文件 ] → [ 样条文件管理 ])

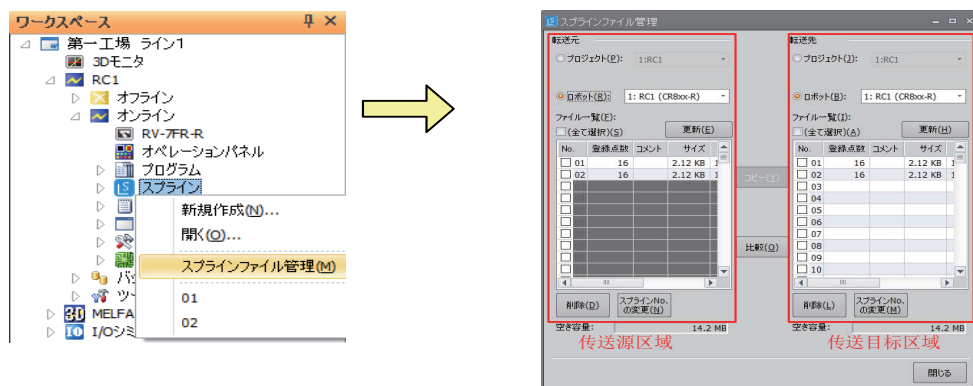


图 7-70: 样条文件管理画面

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. 。

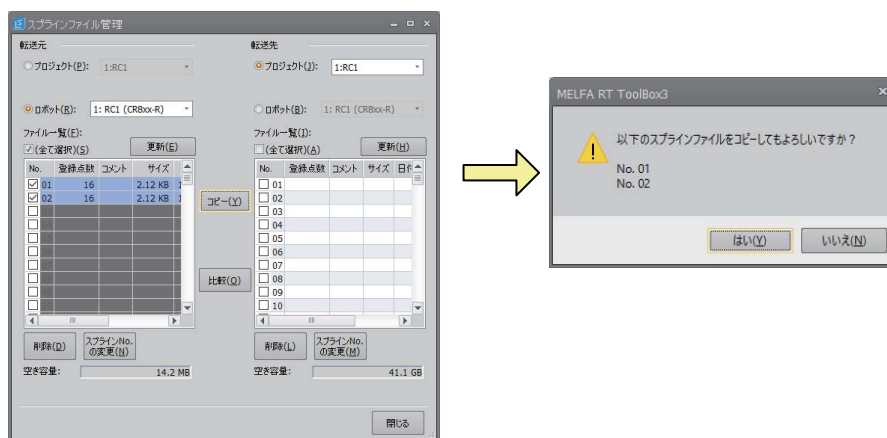


图 7-71: 样条文件管理画面上的文件复制

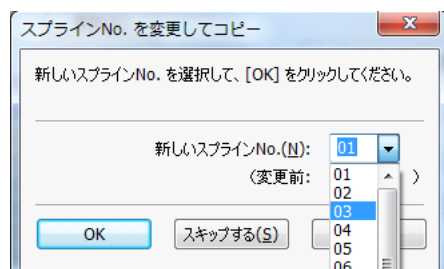


图 7-72: 更改并复制样条 No.

■ 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勾选要删除的文件，点击 [ 删除 ] 按钮，即可删除所选的样条文件。



图 7-73：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勾选要更改的文件，点击 [ 样条 No. 的更改 ] 按钮，即可更改所选的样条文件的 No. 。



图 7-74：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，也可获取从 RT Toolbox3 Pro 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 7-15 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 7-15：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  |

| 数据识别标记     | 内容                                                                                                                                                                                                                                                                                                                                     |    |              |   |     |   |        |    |          |    |              |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|--------------|---|-----|---|--------|----|----------|----|--------------|
| <B>        | 指定路径点的 B 轴坐标值。<br>单位：度                                                                                                                                                                                                                                                                                                                 |    |              |   |     |   |        |    |          |    |              |
| <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）                                                                                                                                                                                                                                                                                                       |    |              |   |     |   |        |    |          |    |              |
| <To11>     | 以整数指定位置（轨迹）的允许公差量。<br>设定范围：0 ~ 100 [%]                                                                                                                                                                                                                                                                                                 |    |              |   |     |   |        |    |          |    |              |
| <To12>     | 以整数指定姿势的允许公差量。<br>设定范围：0 ~ 100 [%]                                                                                                                                                                                                                                                                                                     |    |              |   |     |   |        |    |          |    |              |
| <To13>     | 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" data-bbox="513 1171 979 1344"> <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" data-bbox="513 1433 979 1606"> <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> | <To11> | <To12> | <To13> | <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 7-75: CSV 文件的示例

第 1 行中请务必从 <X> 开始按顺序记述表 7-15 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 RT ToolBox3 Pro 的 MELFA-Works 功能中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 功能的使用方法, 请参照“RT ToolBox3 Pro 使用说明书 (BFP-A3553: 日语版)”。

※ 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 功能的启动  
启动 MELFA-Works 功能。

- 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。

点击功能区 [MELFA-Works] 中的 [抓手设定], 显示抓手设定画面。

选择连接抓手的夹持抓手, 点击抓手状态按钮来夹持工件。

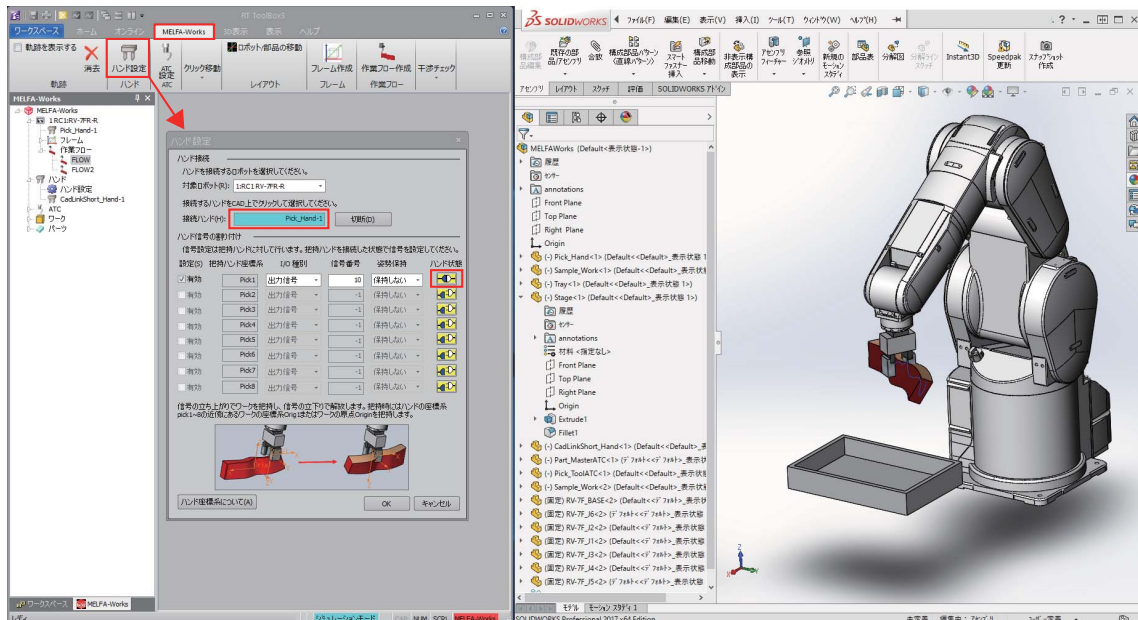


图 7-76: 工件夹持



## 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。

点击功能区 MELFA-Works 中的 [ 作业流程创建 ]，显示作业流程创建画面。编辑现有的作业流程时，显示 MELFA-Works 树中的同一个画面。点击示教标签中 [ 增加 ] → [ 当前位置获取 ]，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。

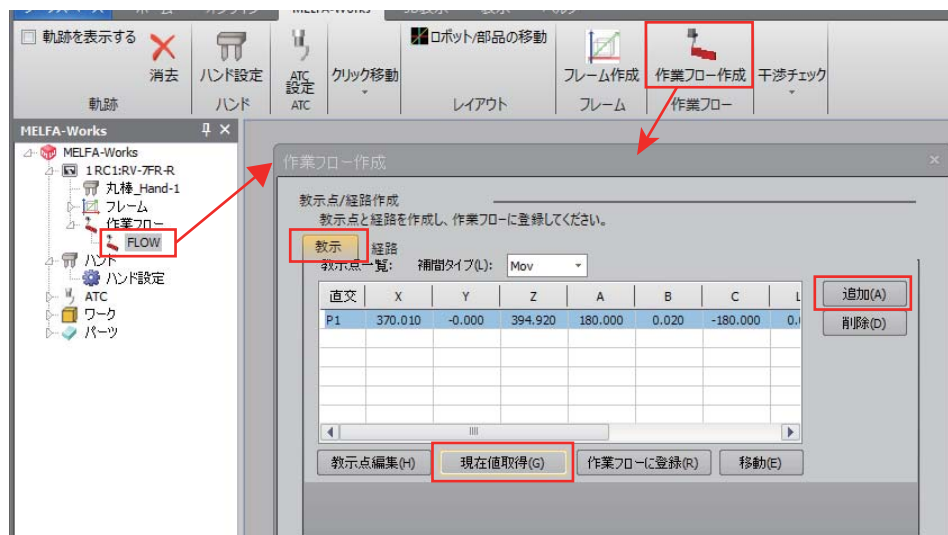


图 7-77：夹持位置的示教

## 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。

选择路径标签，点击 [ 增加 ] 按钮，即会增加加工路径。选择增加的加工路径，点击 [ 路径编辑 ] 按钮，显示路径数据的编辑画面。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，应对准 Ex-T 坐标反转 Z 轴。

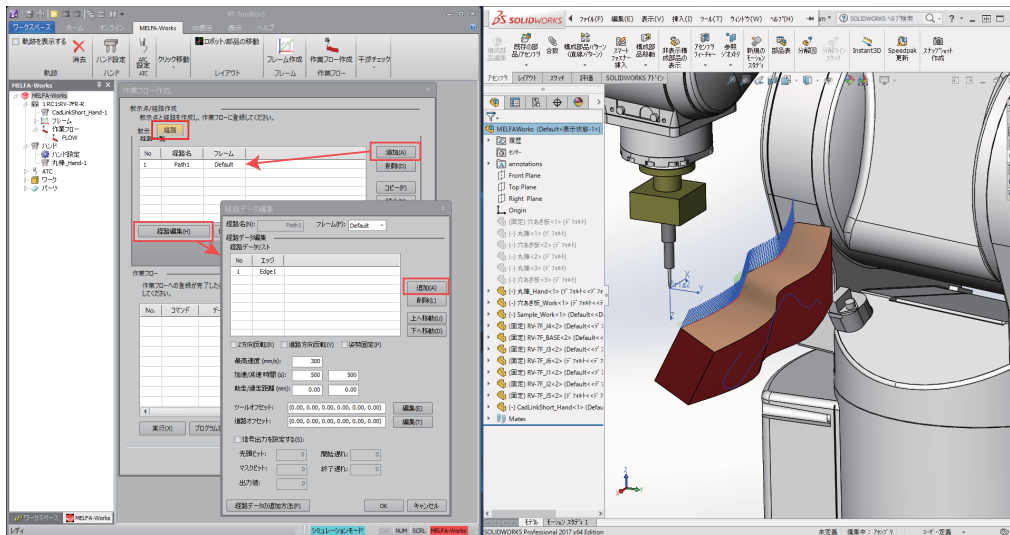


图 7-78：路径创建

## 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，点击 [ 登录作业流程 ] 按钮，即将所选的路径增加至流程中。

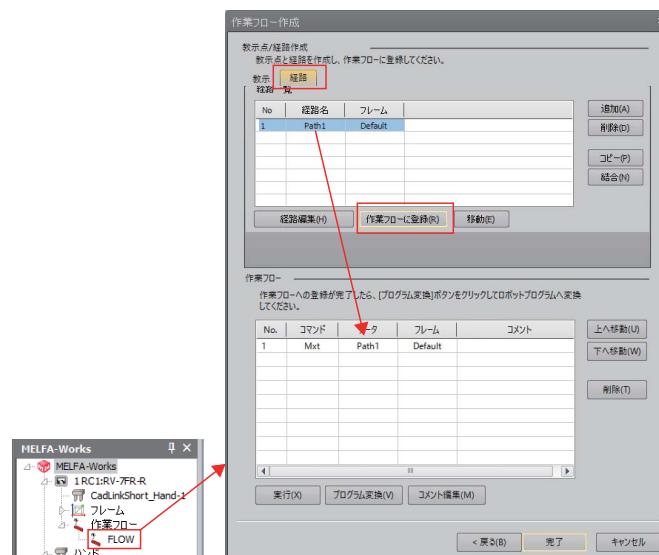


图 7-79：流程创建（加工路径登录）

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，点击 [ 登录作业流程 ] 按钮，即将所选的位置数据增加至流程中。

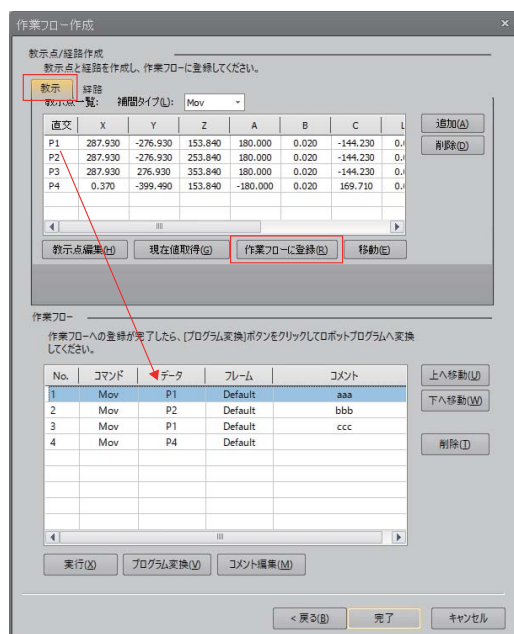


图 7-80：流程创建（夹持位置登录）

#### 6) MXT 文件的输出

输出所创建的作业流程相对应的 MXT 文件。

点击作业流程创建画面的 [ 程序转换 ] 按钮，即会显示程序转换画面。

输入要输出的程序名，点击 [OK] 按钮后，程序将输出到当前工作区，并在以下文件夹中创建 MXT 文件。

#### • MXT 文件输出目标：

< 工作区 > \ < 工程名 > \ < MELFA-Works > \ 程序名 \ \*\*\*. MXT

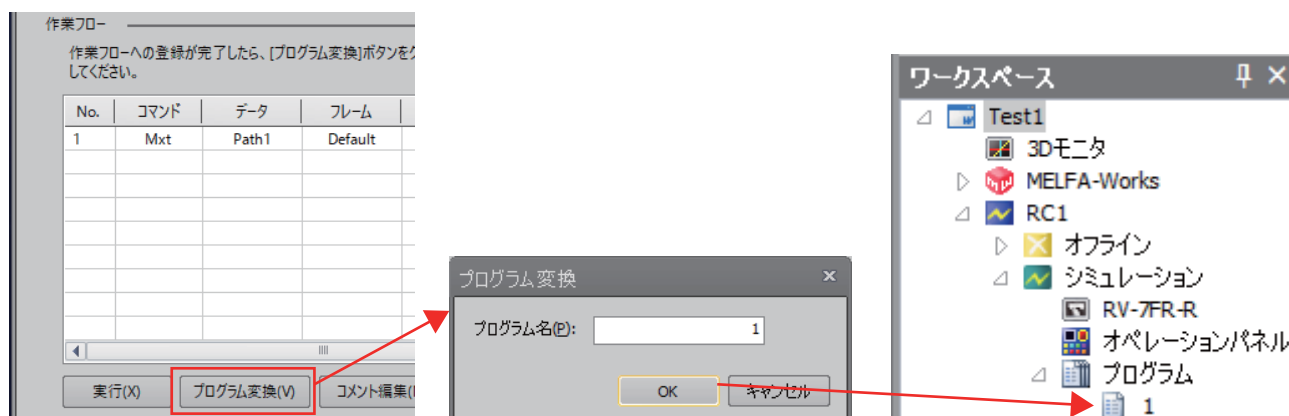


图 7-81：MXT 文件的输出

■ 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。

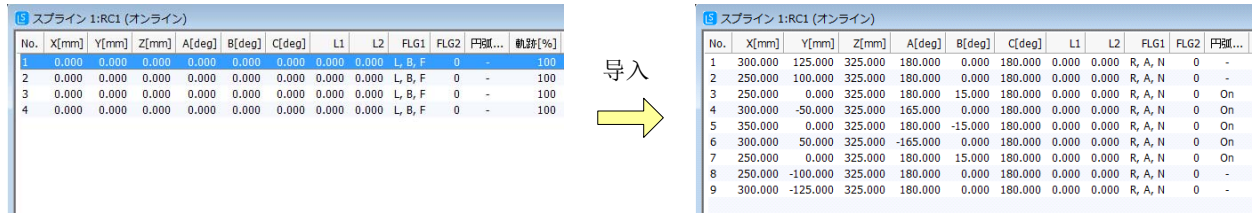


图 7-82：导入



**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

点击功能区 [ 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

点击功能区 [ 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，即显示导入设定画面。

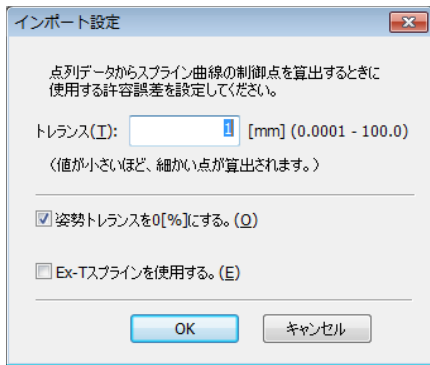


图 7-83：导入设定画面

设定点阵数据与样条曲线之间的允许误差（mm）并点击 [OK] 按钮。

在导入设定画面中，勾选 [ 将姿势公差设为 0[%]。 ] 勾选复选框后，各路径点的姿势公差在 0% 的状态下被获取。以从 MXT 文件中获取的姿势动作时，应勾选。（初始设定为已勾选状态。）

样条时，点击 [OK] 按钮，MXT 文件的内容即被获取至已激活的样条文件编辑画面。（允许误差越小，路径点算出越细。）

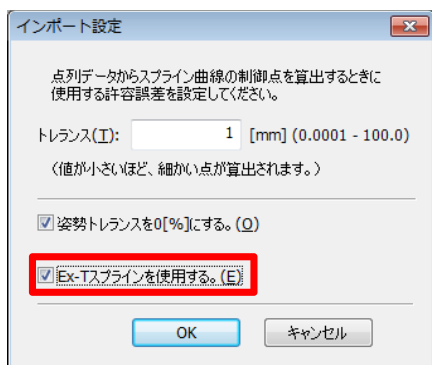


图 7-84: 导入设定画面 (Ex-T 样条时)

Ex-T 样条时, [ 使用 Ex-T 样条。 ] 勾选复选框, 点击 [OK] 按钮后, 显示用于设定夹持位置与 Ex-T 坐标的 Ex-T 控制设定画面。

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。

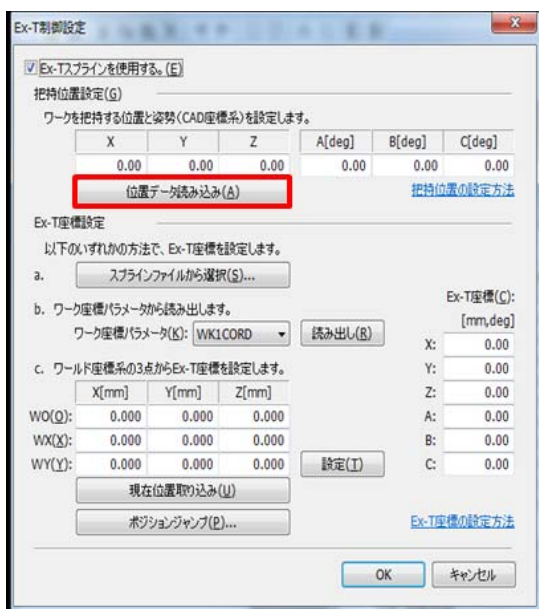


图 7-85: Ex-T 控制设定画面 (设定方法)

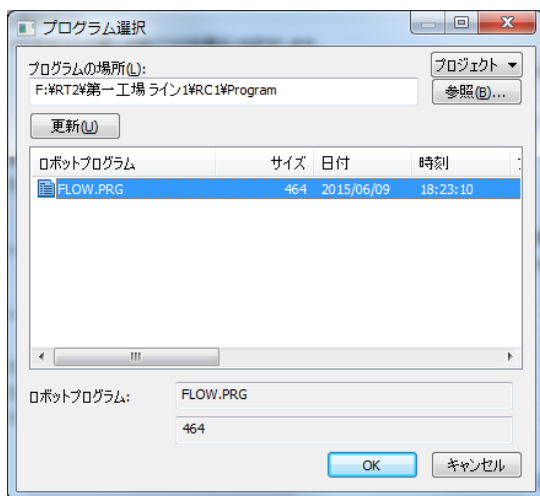


图 7-86: 程序选择画面

选择程序并点击 [OK] 按钮，显示直角位置数据选择画面。

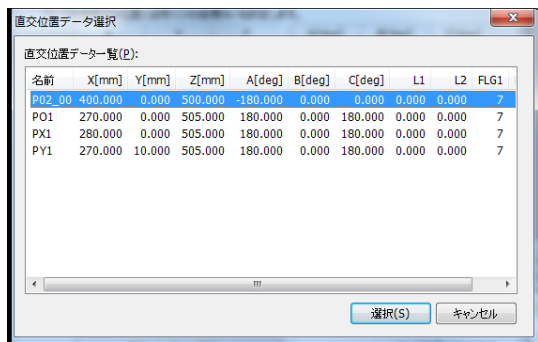


图 7-87：直角位置数据选择画面

选择要读取的位置数据（MXT 文件创建时示教的夹持位置数据）并点击 [OK] 按钮，夹持位置即被设定。

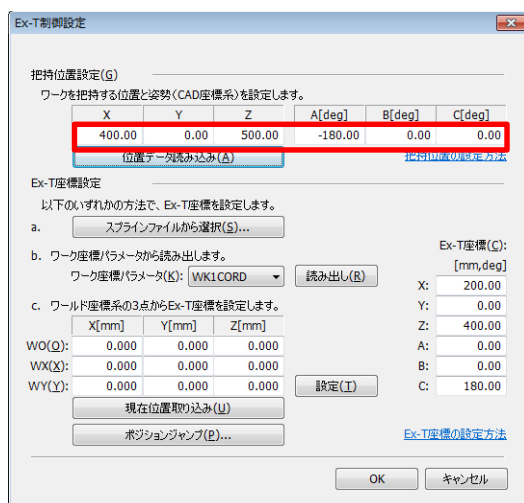


图 7-88：Ex-T 控制设定（夹持位置设定）

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获取。

### ⚠ 注意

对应的点阵的数据类型仅有直角数据。  
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 MvSp1 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&amp;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作为剪切位置的路径点数据，点击功能区 [编辑] → [剪切]。然后，选择作为粘贴目标的路径点数据，并点击功能区 [编辑] → [粘贴]，通过 [粘贴] 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。



图 7-89: 剪切&amp;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&amp;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源的路径点数据，点击功能区 [编辑] → [复制]。然后，选择作为复制目标的路径点数据，并点击功能区 [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。

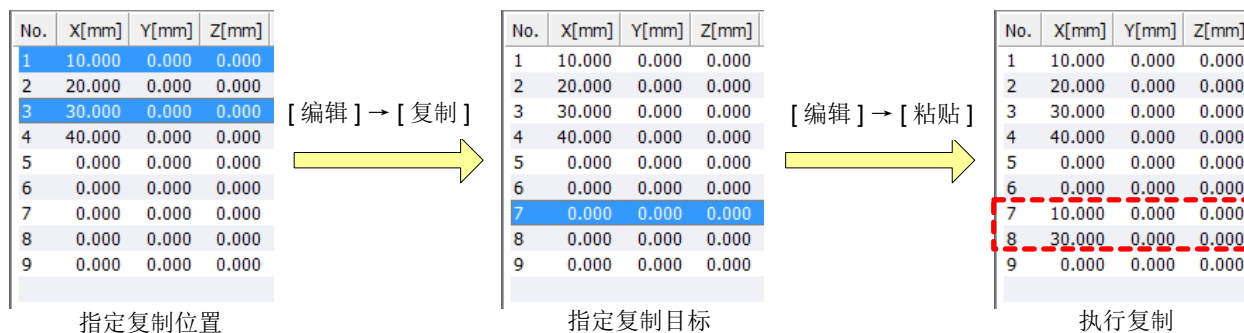


图 7-90: 复制&amp;粘贴

## ◇◆◇ 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，点击功能区 [编辑] → [复制 - 位置数据]。然后，选择作为复制目标的路径点数据，并点击功能区 [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 7-91：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。

在路径点数据一览表区域中选择作为复制源的路径点数据，点击功能区 [编辑] → [复制]。然后，选择作为插入目标的路径点数据，并点击功能区 [编辑] → [已复制路径点的插入]，将 [复制] 操作所存储的路径点数据即按顺序被插入至作为插入目标所选择的路径点数据之前。

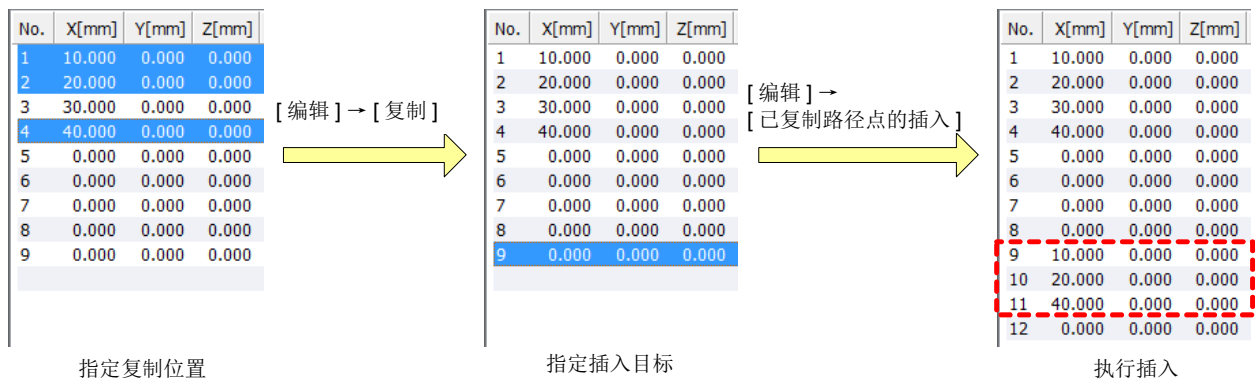


图 7-92：所复制的路径点的插入

### ■ 撤销（undo）

点击功能区 [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。

- 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）

点击功能区 [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。

点击功能区 [ 工具 ] → [ 样条曲线 ], 3D 监视画面将打开、编辑中的样条文件的样条曲线会显示在 3D 监视画面中。

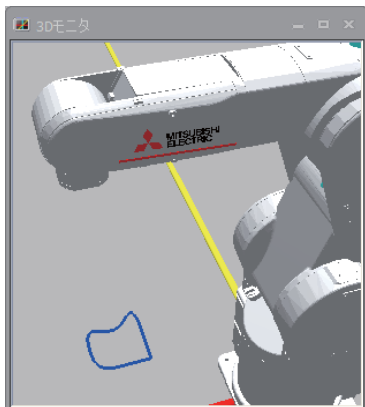


图 7-93: 样条曲线的显示

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。

点击功能区 [ 工具 ] → [ 路径点检查 ], 即显示已激活的样条编辑画面的路径点检查画面。

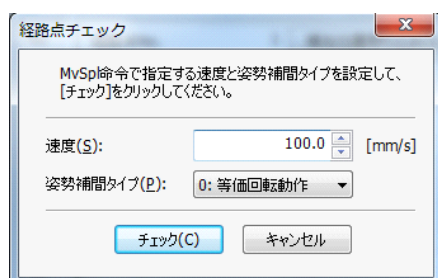


图 7-94: 路径点检查画面

设定通过 MvSp1 指令 /EMvSp1 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。

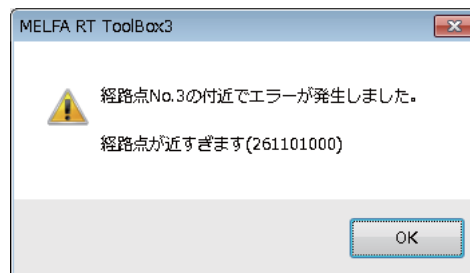
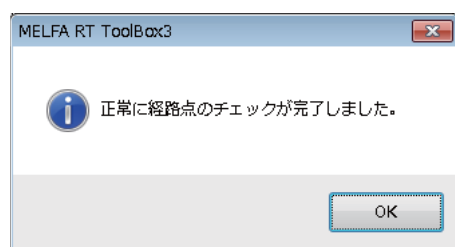


图 7-95: 路径点检查完成对话框 (左) / 错误对话框 (右)

### ■ 动作范围检查

事先对各路径点检查机器人可否动作。

点击功能区 [ 工具 ] → [ 动作范围检查 ], 即向已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。

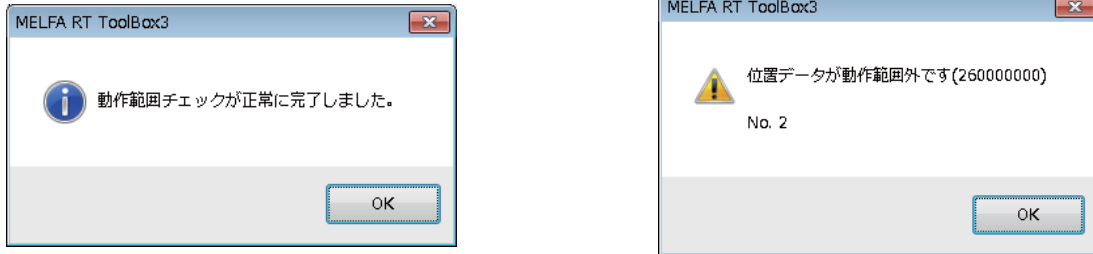


图 7-96: 动作范围检查完成对话框 (左) / 错误对话框 (右)

### ■ 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 (MvSp1 指令、EM v Sp1 指令) 时不会发生错误的可执行的最大速度。

点击功能区 [ 工具 ] → [ 最大速度算出 ], 即显示已激活的样条编辑画面的最大速度算出结果。

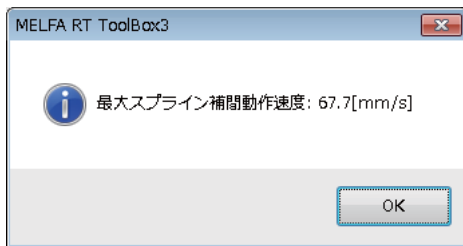


图 7-97: 最大速度算出结果

## (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。

点击功能区 [ 工具 ] → [ 机器人程序 ] → [ 程序转换 ], 即显示已激活的样条编辑画面的新机器人程序对话框。

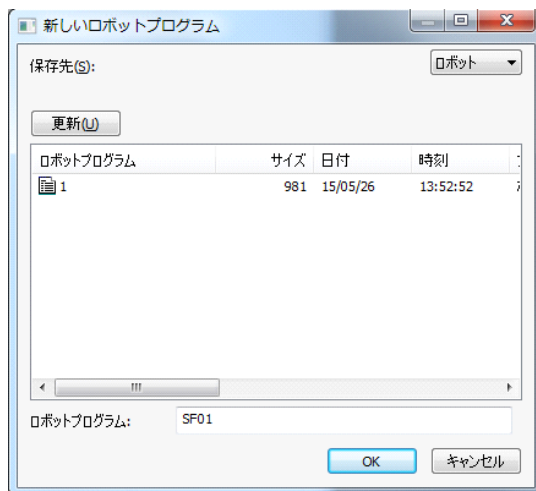


图 7-98: 新机器人程序画面

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。

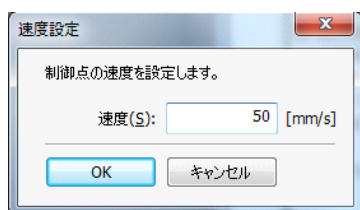


图 7-99：速度设定对话框

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。

```

1 | Convert from the spline file (#01)
2 | Servo On
3 | Ovrd 100
4 | Spd 50
5 | Points: 17
6 | Dim PP(17)
7 | Move the start position
8 | Mvs PP(1)
9 | Start the spline interpolation
10 | Line block: #1 to #2
11 | Mvs PP(2)
12 | Arc block: #2 -> #3 -> #4
13 | Mvr PP(2),PP(3),PP(4)
14 | Spline block: #4 to #5
15 | Dim Ps_004(10)
16 | For M=T To 10 Step 1
17 | Mvs Ps_004(M)
18 | Next M
19 | Mvs PP(5)
20 | Spline block: #5 to #6
21 | Dim Ps_005(10)
22 | For M=T To 10 Step 1
23 | Mvs Ps_005(M)
24 | Next M

```

### 样条插补

```

1 | Convert from the spline file (#02)
2 | Servo On
3 | Ovrd 100
4 | Spd 50
5 | Ex-T Spline
6 | P_WkCord(1)=(200.000,0.000,400.000,0.000,0.000,180.000)
7 | Points: 17
8 | Dim PP(17)
9 | Move the start position
10 | EMvs I,PP(1)
11 | Start the spline interpolation
12 | Line block: #1 to #2
13 | EMvs I,PP(2)
14 | Arc block: #2 -> #3 -> #4
15 | EMvr I,PP(2),PP(3),PP(4)
16 | Spline block: #4 to #5
17 | Dim Ps_004(10)
18 | For M=T To 10 Step 1
19 | EMvs I,Ps_004(M)
20 | Next M
21 | EMvs I,PP(5)
22 | Spline block: #5 to #6
23 | Dim Ps_005(10)
24 | For M=T To 10 Step 1

```

### Ex-T 样条插补

图 7-100：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。

点击功能区 [ 工具 ] → [ 机器人程序 ] → [ 位置数据导入 ]，即显示已激活的样条编辑画面中要导入的程序的選擇画面。

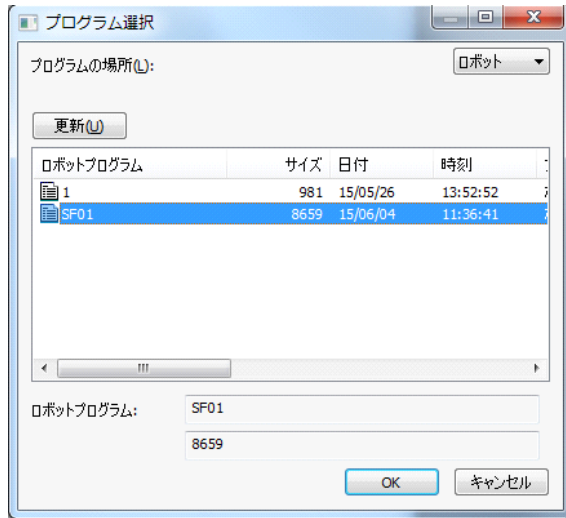


图 7-101：程序选择画面

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。)

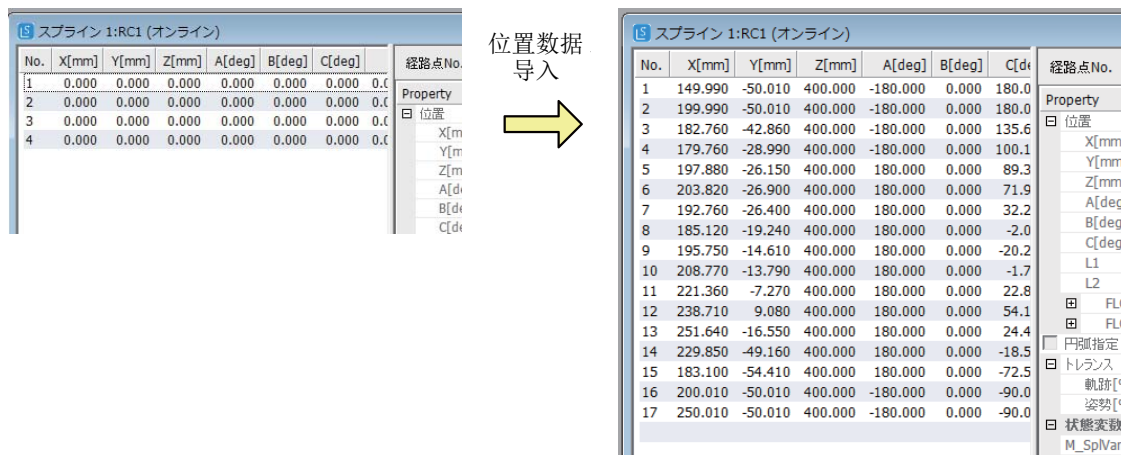


图 7-102：位置数据导入

## 7.2.6 机器人程序的创建

在 RT ToolBox3 · 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 7-16 与表 7-18 中的指令 · 机器人状态变量。关于详细内容，请参照第 173 页的“4.12 指令的详细说明” · 第 333 页的“4.13 机器人（系统）状态变量的详细说明”。

表 7-16：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。       | 275  |
| EMvSpl (E Move Spline)            | 执行 Ex-T 样条插补。 | 235  |
| SetCalFrm (Set Calibration Frame) | 设定帧转换中使用的坐标系。 | 310  |

表 7-17：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。                                               | 459  |
| 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。 | 460  |
| 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。                                       | 458  |

表 7-18：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 <sup>注1)</sup> | 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   | 388  |
| 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   | 389  |

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。

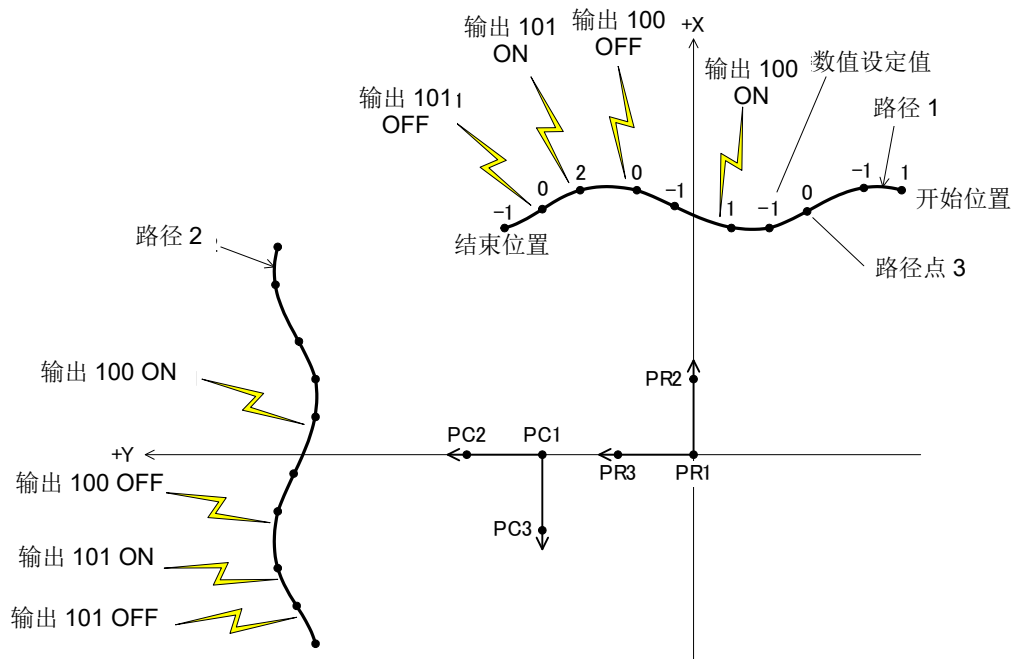


图 7-103: 采样程序的动作

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)        | ‘ 的坐标系)                                   |
| 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                             |                               |

### 7.2.7 动作的确认

使用 RT ToolBox3 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。

关于模拟功能的使用方法，请参照“RT ToolBox3/RT ToolBox3 mini 使用说明书”。

此外，在 RT ToolBox3 mini 中无法使用模拟功能。

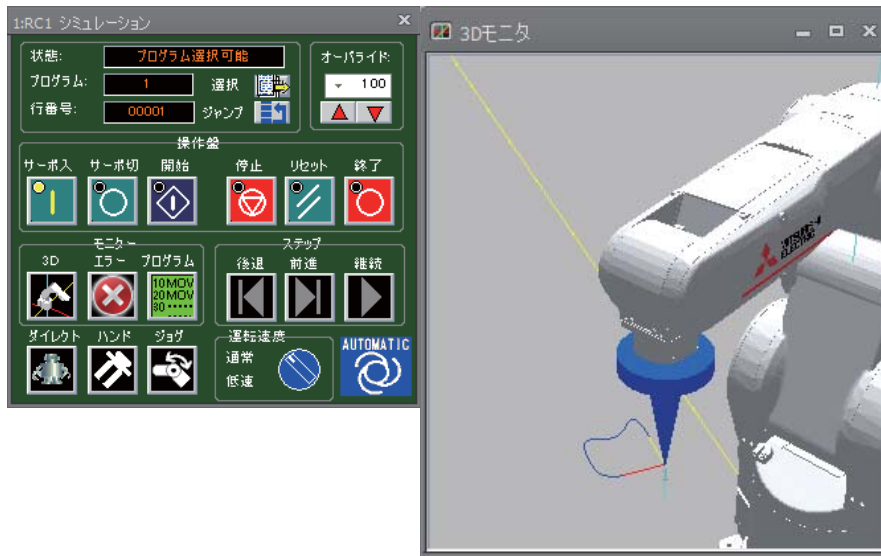


图 7-104: 模拟功能

#### ◇◆◇模拟所使用的机器人程序与样条文件◇◆◇

通过模拟执行样条插补时，机器人程序与样条文件需已登录在 RT ToolBox3 内的虚拟控制器中。启动了模拟的工程の [ 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 ToolBox3 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 692 页的“ (9) 样条文件的保存 ”。此外，进行第 695 页的“ (12) 样条文件的复制 ”。

### 7.2.8 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 692 页的“ (9) 样条文件的保存 ”。

### 7.2.9 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 ToolBox3 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。

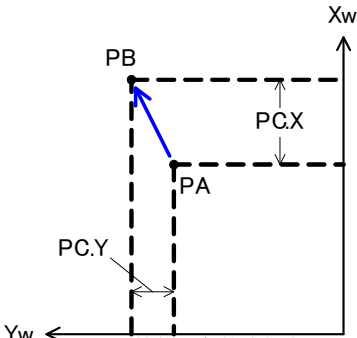
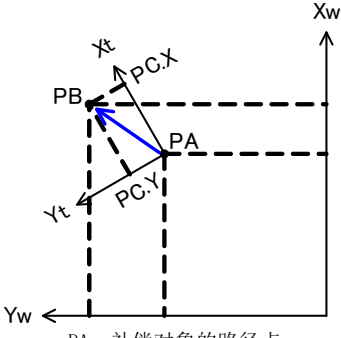
下面对为了补偿位置数据而在 RT ToolBox3 中配备的“位置补偿功能”・“帧转换功能”・“跳位功能”和参数 SPLOPTGC（有源增益控制的增益补偿率）进行说明。

#### (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 VI 的位置数据的相对运算同样的补偿。如表 7-19 所示，补偿方法有 2 种。



表 7-19: 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。</p> <p>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。（补偿后的位置 = 路径点的位置 × 补偿数据）</p> <p>是基于工具坐标系进行的补偿。</p> <p>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 x P)</p> |

使用位置补偿功能时，点击功能区 [ 工具 ] → [ 位置补偿 ]。点击 [ 位置补偿 ] 后，即显示位置补偿画面。

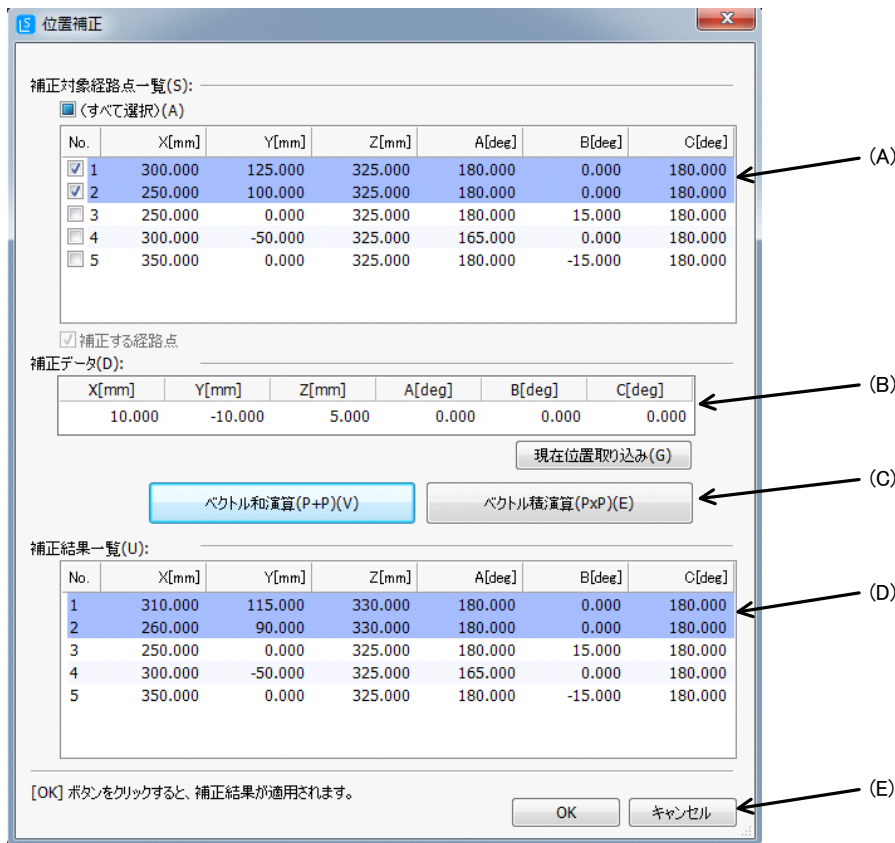


图 7-105: 位置补偿画面

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

点击功能区 [工具] → [帧转换]，即显示用于对已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 666 页的“(5) 帧转换”。)

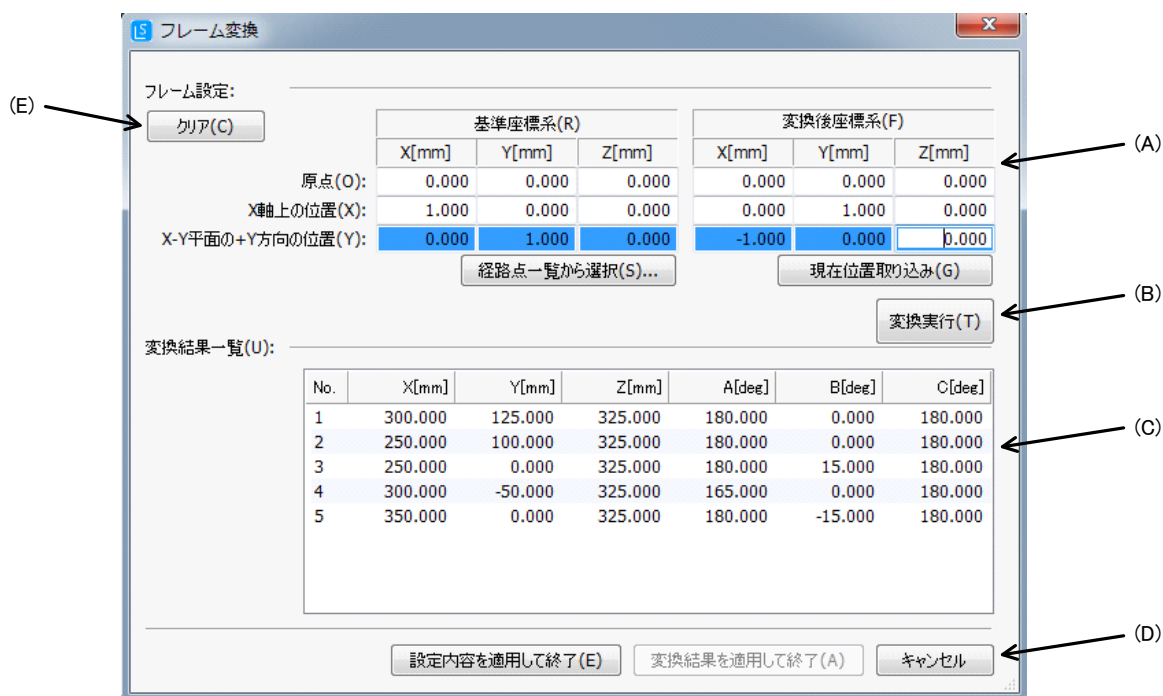


图 7-106：帧转换画面

### 坐标系的设定

图 7-106 (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。

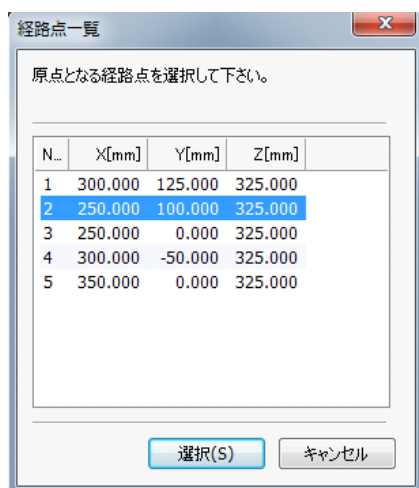


图 7-107：从路径点一览中选择

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 ToolBox3 执行帧转换

设定坐标系后，按图 7-106 (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 7-106 (C) 的“转换结果一览”画面中显示该结果。

实施帧转换后，图 7-106 (D) 的 [ 适用转换结果后结束 ] 按钮即变为有效。

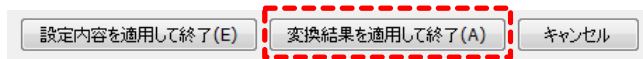


图 7-108: [ 适用转换结果后结束 ] 按钮

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 7-109: 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 7-106 (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。

MvSp1 指令 /EMvSp1 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。

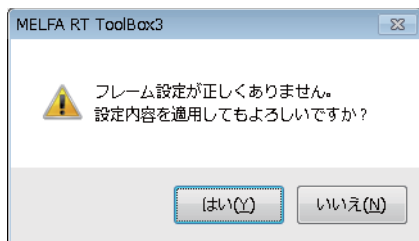


图 7-110: 坐标系的设定不正确时的对话框（保存坐标系时）

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 7-106 (E) 的 [ 清除 ] 按钮。点击该按钮，即显示确认对话框。

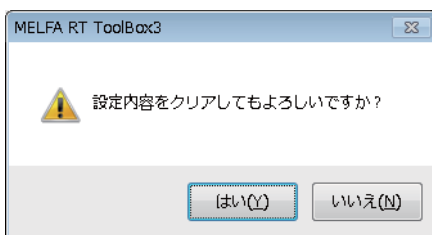


图 7-111: 坐标系清除的确认对话框

按压 [ 是 ] 按钮，存储在样条文件编辑画面中的坐标系の設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。（点击功能区 [ 工具 ] → [ 跳位 ] 也可以显示。）

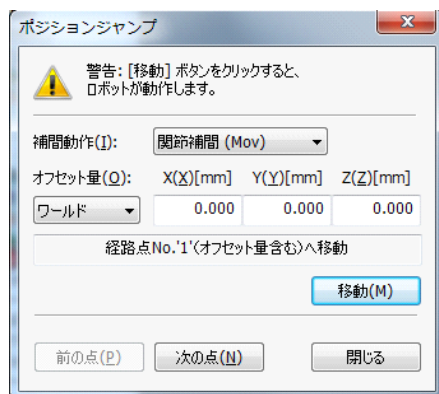


图 7-112: 跳位画面

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 )。



### 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，无法点击功能区 [ 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 7-20：参数 SPLOPTGC

| 参数           | 参数名称     | 数组数字字符数 | 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    |

## 7.2.10 样条插补指令速度高速化

## (1) 概要

通过参数 SPLHSPMD、可以指定更大的样条插补动作速度。

表 7-21: 参数 SPLHSPMD

| 参数          | 参数名称     | 数组数字字符数 | 内容说明                                                                    | 出厂时设定值 |
|-------------|----------|---------|-------------------------------------------------------------------------|--------|
| 样条插补指令速度高速化 | SPLHSPMD | 整数 1    | 切换样条插补指令速度高速化的有效 / 无效。<br>(无效 / 有效 =0/1)<br><br>※ 本参数在设定值更改后, 无须重新接通电源。 | 0      |

## (2) 必要的软件版本

样条插补指令速度高速化可用于下表所示的机器人控制器和 RT ToolBox3 的软件版本。

表 7-22: 可以使用样条插补指令速度高速化的软件版本

| 设备       | 控制器         | RT ToolBox3   |
|----------|-------------|---------------|
| 可使用的软件版本 | Ver. A3a 以上 | Ver. 1.31H 以上 |

## (3) 规格

根据参数 SPLHSPMD 的设定值, 指令速度和必要路径点间的距离关系, 从第 660 页的“(4) 路径点相关的检查”变为下表。

如果通过参数 SPLHSPMD 使指令速度高速化有效, 则可以对同一路径点间的距离, 指定比现在更大的速度。

表 7-23: 指令速度和必要的路径点间的距离

| 指令速度 [mm/s] | 必要的路径点间的距离 |            |
|-------------|------------|------------|
|             | SPLHSPMD=0 | SPLHSPMD=1 |
| 10          | 0.6mm 以上   | 0.08mm 以上  |
| 20          | 1.2mm 以上   | 0.16mm 以上  |
| 50          | 3.0mm 以上   | 0.4mm 以上   |
| 100         | 6.0mm 以上   | 0.8mm 以上   |
| 200         | 12.0mm 以上  | 1.6mm 以上   |
| 500         | 30.0mm 以上  | 4.0mm 以上   |

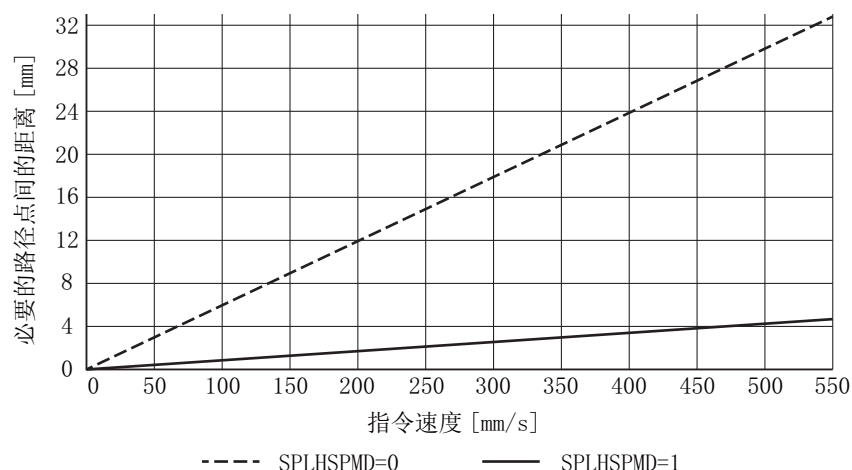
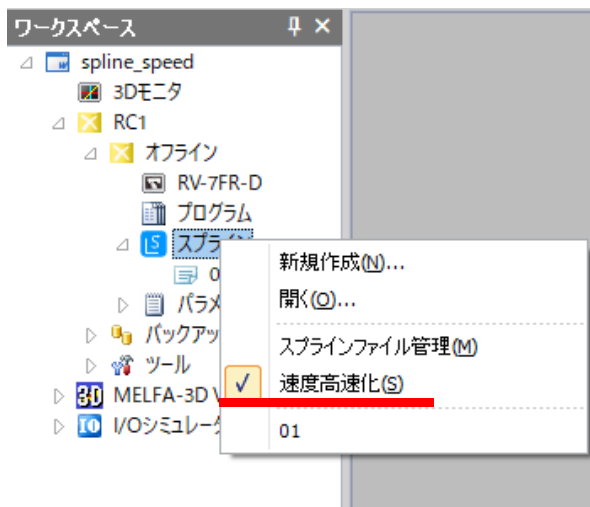


图 7-113: 指令速度和必要的路径点间的距离

## (4) RT ToolBox3 的设置

样条插补指令速度高速化也可以从 RT ToolBox3 的项目树 -[ 样条插补 ] 项的上下文菜单设定。如下图所示，通过勾选 [ 速度高速化 ] 的 ON/OFF，可以切换其有效 / 无效。



样条插补指令速度高速化有效时，如表 7-11 所示 RT ToolBox3 的“路径点检查”和“算出最大速度”是与样条插补指令速度高速化对应的结果。此外，“路径点检查”如下图所示。

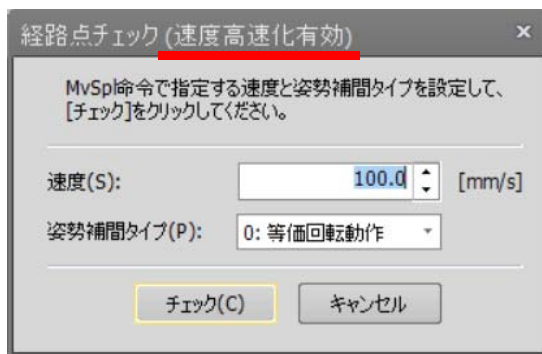


图 7-114: RT ToolBox3 路径点确认 (速度高速化有效)

## (5) 注意事项・制约事项

关于样条插补指令速度高速化，有以下注意事项・制约事项。

- 1) 如果使指令速度高速化有效，可以指定比现在更大的速度，但是根据机器人的位置・姿势，有时会发生超速，在超速的状态下将导致无法动作。
- 2) 对于已经动作中的样条插补，如果使指令速度高速化有效，位置指令可能会有细微地变化。在这种情况下，请将指令速度高速化设为无效。
- 3) 由 SplSpd 函数计算的最大速度发生变化。已经使用 SplSpd 函数的情况下，请注意计算值的变化。此外，如第 708 页的“(17) 编辑内容的确认”所示，通过 RT ToolBox3 的“算出最大速度”计算出的速度也会发生变化。
- 4) 如果脉冲输出有效，则通常在通过路径点之后输出约 14ms 的指定信号，但是如果指定的速度较大，则可能会使脉冲宽度比约 14ms 更短（最短约 3.5ms）。
- 5) 指定速度较大的情况下，会减弱抑制指定动作模式的“线速可变”时的超速效果。在发生超速的情况下，请采取减小姿势变化，扩大路径点间的距离，降低指令速度等措施。
- 6) 如果通过 RT ToolBox3 的模拟来确认样条插补的动作，请将模拟和机器人控制器的指令速度高速化的有效 / 无效设定相同。如果有效 / 无效不吻合，即使通过模拟可以动作，在实际动作中也可能报错。



## 7.3 关于Ex-T控制

### 7.3.1 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 7-115: 研磨作业的示例

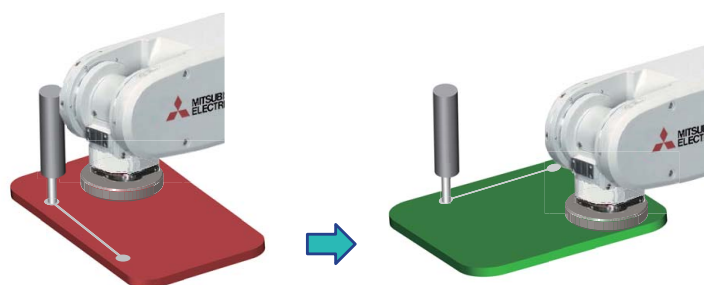


图 7-116: 涂敷作业的示例

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          | <ul style="list-style-type: none"> <li>指令 <table border="1"> <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>233</td> </tr> <tr> <td>EMvc (E Move C)</td> <td>Ex-T 控制圆形插补</td> <td>227</td> </tr> <tr> <td>EMvr (E Move R)</td> <td>Ex-T 控制圆弧插补</td> <td>228</td> </tr> <tr> <td>EMvr2 (E Move R2)</td> <td>Ex-T 控制圆弧插补 2</td> <td>230</td> </tr> <tr> <td>EMvr3 (E Move R3)</td> <td>Ex-T 控制圆弧插补 3</td> <td>231</td> </tr> </tbody> </table> </li> <li>状态变量 <table border="1"> <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>423</td> </tr> </tbody> </table> </li> </ul> | 指令  | 说明 | 页码 | EMvs (E Move S) | Ex-T 控制直线插补 | 233 | EMvc (E Move C) | Ex-T 控制圆形插补 | 227 | EMvr (E Move R) | Ex-T 控制圆弧插补 | 228 | EMvr2 (E Move R2) | Ex-T 控制圆弧插补 2 | 230 | EMvr3 (E Move R3) | Ex-T 控制圆弧插补 3 | 231 | 变量名称 | 数组元素数 | 页码 | P_WkCord | 工件坐标号（1～8） | 423 |
| 指令                | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 页码  |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvs (E Move S)   | Ex-T 控制直线插补                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 233 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvc (E Move C)   | Ex-T 控制圆形插补                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 227 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvr (E Move R)   | Ex-T 控制圆弧插补                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 228 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvr2 (E Move R2) | Ex-T 控制圆弧插补 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 230 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| EMvr3 (E Move R3) | Ex-T 控制圆弧插补 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 231 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| 变量名称              | 数组元素数                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 页码  |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| P_WkCord          | 工件坐标号（1～8）                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 423 |    |    |                 |             |     |                 |             |     |                 |             |     |                   |               |     |                   |               |     |      |       |    |          |            |     |
| 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

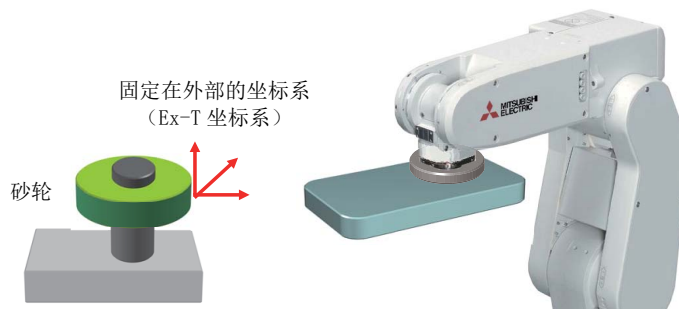
Ex-T 控制可在下表所示的机器人控制器的软件版本中使用。

表 7-24：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 | Ver. A1 以上 | R32TB: Ver. 1.3 以上<br>R56TB: Ver. 2.3 以上 |

## 7.3.2 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 ToolBox3 进行参数设定
- 对机器人程序（MELFA-BASIC VI）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502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;"> </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。请通过操作或RT ToolBox3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。请通过操作或RT ToolBox3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。请通过操作或RT ToolBox3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，如设定工具转换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 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“动作的确认”的“工件 JOG 操作”。

### 7.3.3 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。                | 与以往的工作 JOG 相同。                 |
| ABC 键的动作                  | 在保持控制点位置不变的状况下，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                 | 与以往的工作 JOG 相同                                      |
| C 键的动作                    | 在保持控制点位置不变的状况下，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。                                            |

以往的工作 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 7-117 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)。

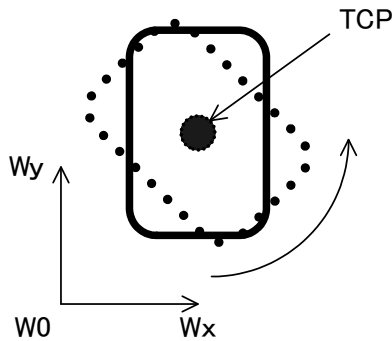


图 7-117: 工件 JOG 的 C 成分动作

(2) Ex-T JOG 中的姿势成分动作

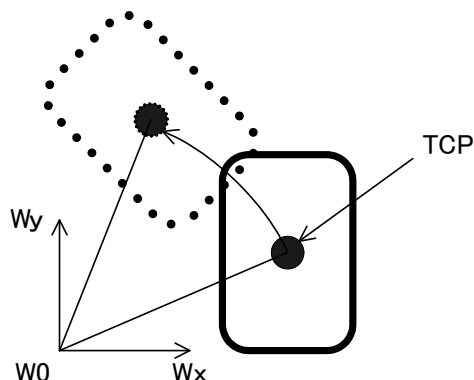
Ex-T JOG 中的姿势成分的动作，为绕 Ex-T 坐标系 (工件坐标系) 的 XYZ 轴的旋转。此时，机器人的位置也发生变化。

Ex-T JOG 中 C 成分动作示例如图 7-118 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），圆角的正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）。

<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>

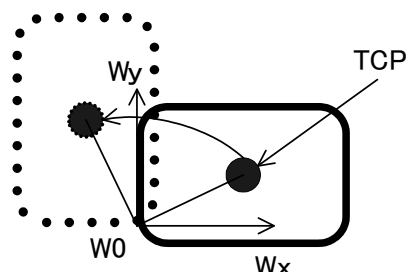


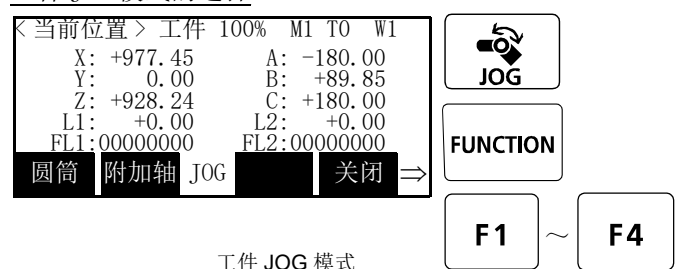
图 7-118: Ex-T JOG 的 C 成分动作

### (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



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

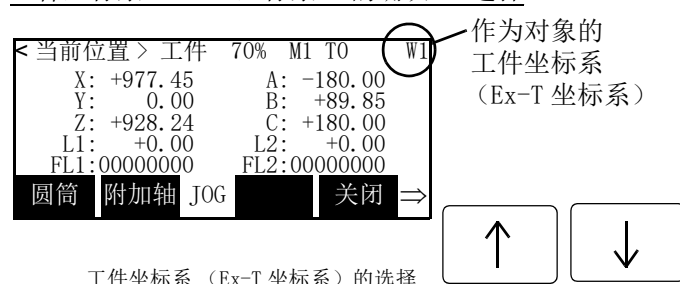
请按压“工件”对应的功能键。

（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。

注) 显示的坐标值是基于直角坐标系的值。

#### 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

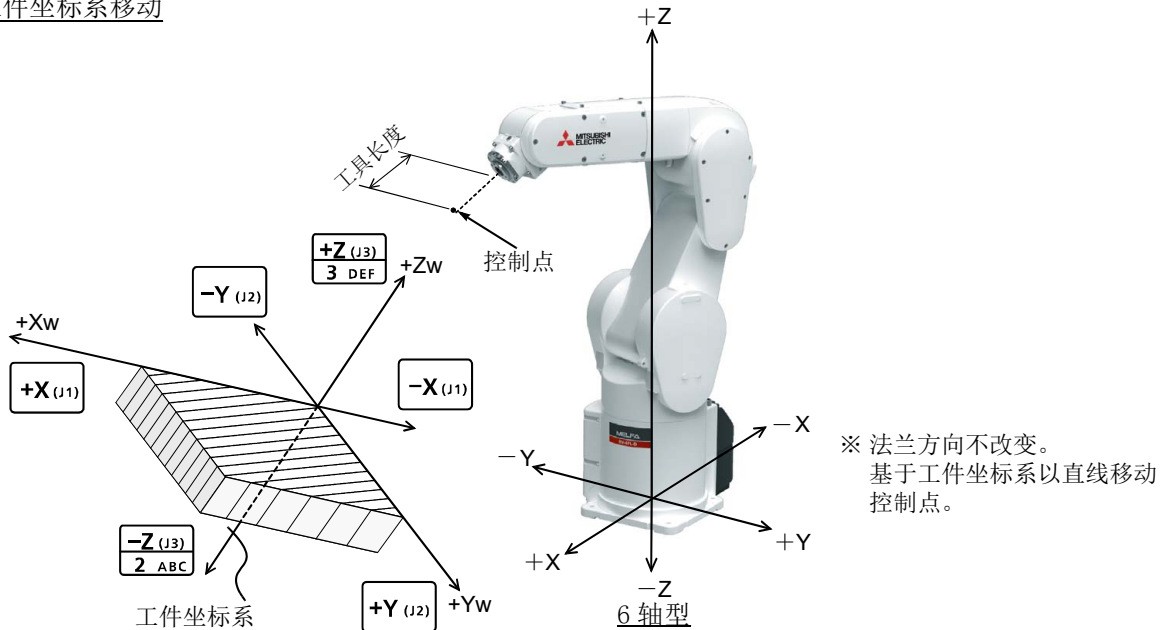
请务必确认对象工件坐标系（Ex-T 坐标系）的号码显示正确。（画面右上方的 W1 ~ W8 的显示）

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

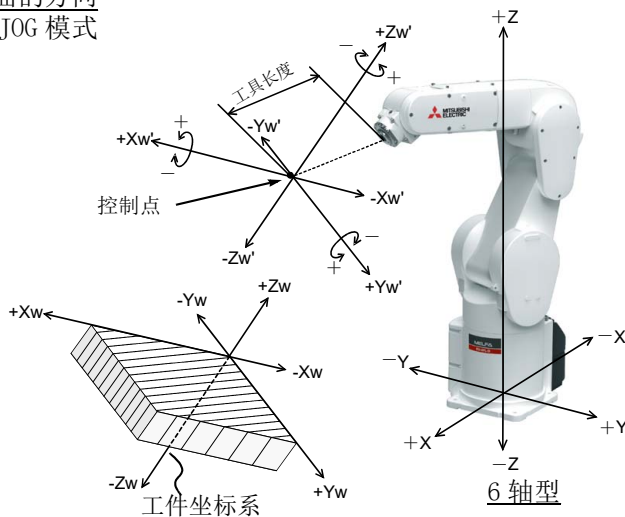


- 按压 [+X(J1)] 键时沿着工件坐标系的 X 轴 (Xw) 的正方向移动。  
按压 [-X(J1)] 键时沿着负方向移动。
- 按压 [+Y(J2)] 键时沿着工件坐标系的 Y 轴 (Yw) 的正方向移动。  
按压 [-Y(J2)] 键时沿着负方向移动。
- 按压 [+Z(J3)] 键时沿着工件坐标系的 Z 轴 (Zw) 的正方向移动。  
按压 [-Z(J3)] 键时沿着负方向移动。

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

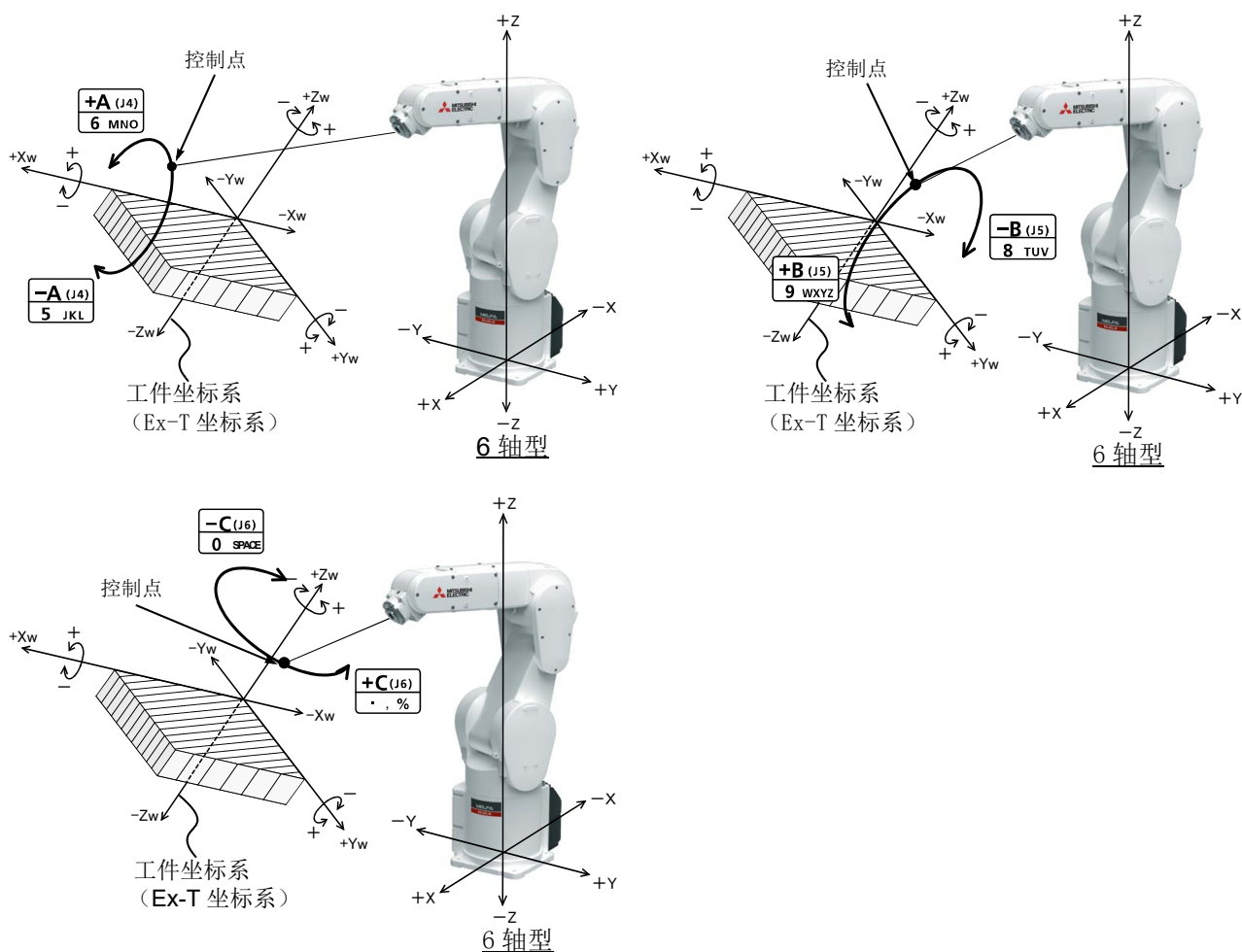
①工件 JOG 模式



※ 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。

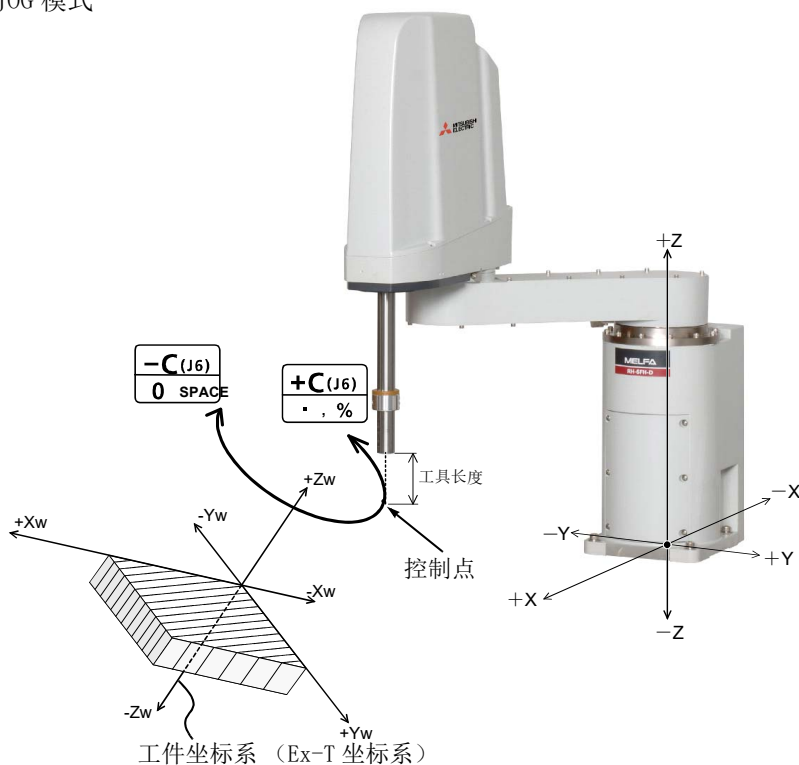
### 改变前端的方向

#### 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

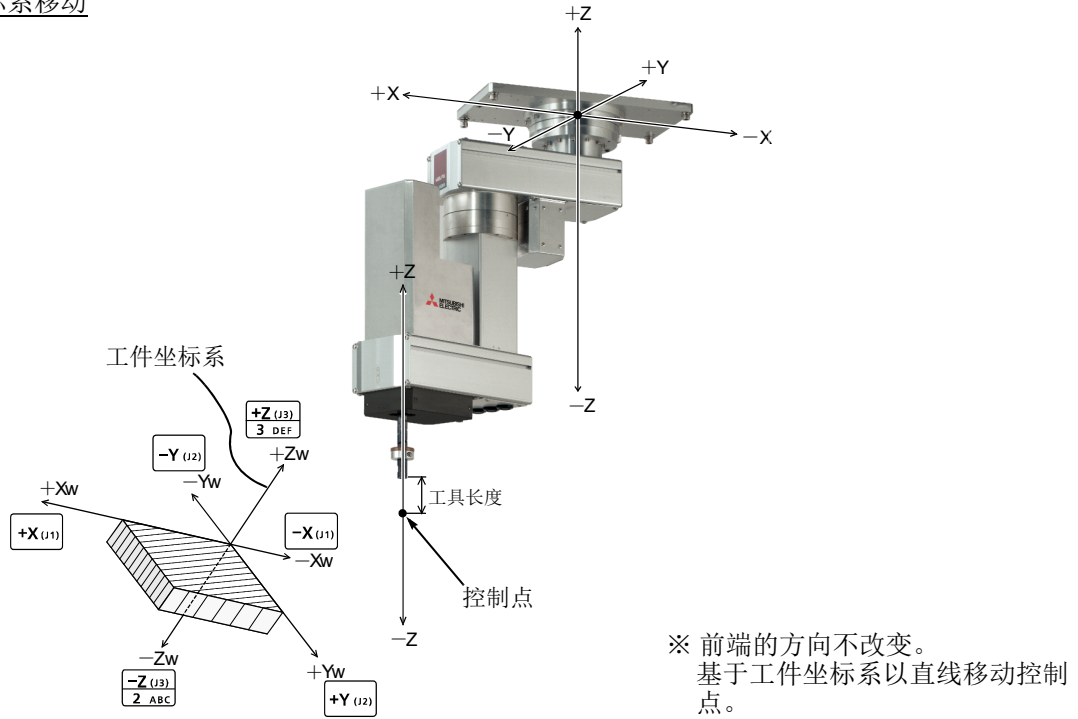


- 按压 [+C(J6)] 键时，以工件坐标系 (Ex-T 坐标系) 的 Z 轴 (Zw) 为中心，控制点向正方向旋转。  
按压 [-C(J6)] 键时，向负方向旋转。

(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

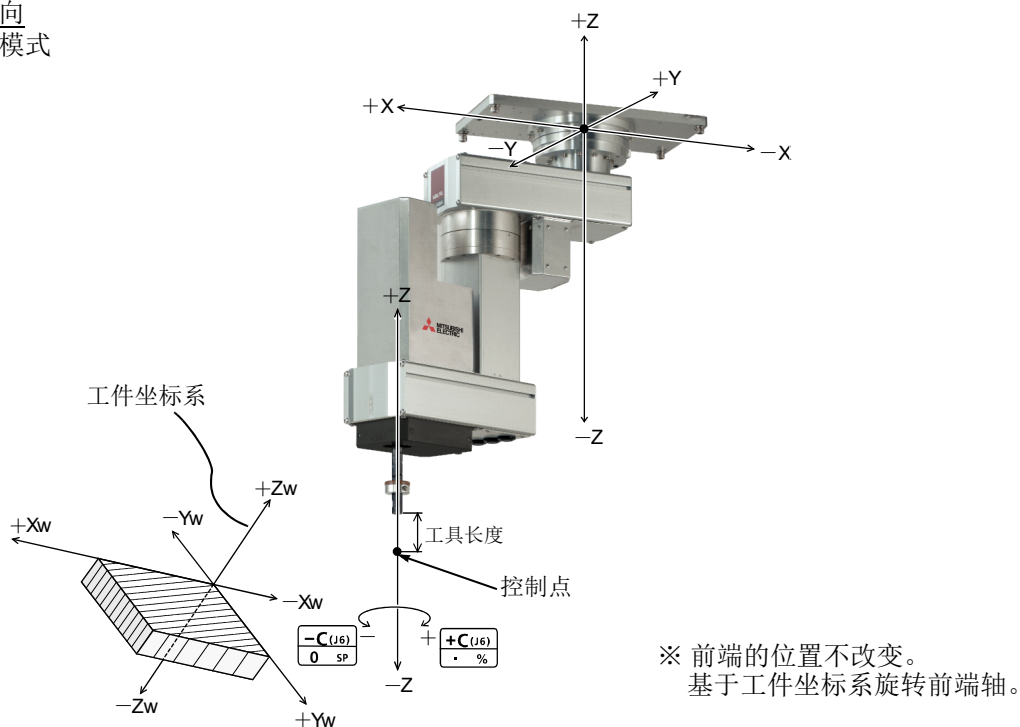


- 按压 [+X(J1)] 键时沿着工件坐标系的 X 轴 (Xw) 的正方向移动。  
按压 [-X(J1)] 键时沿着负方向移动。
- 按压 [+Y(J2)] 键时沿着工件坐标系的 Y 轴 (Yw) 的正方向移动。  
按压 [-Y(J2)] 键时沿着负方向移动。
- 按压 [+Z(J3)] 键时沿着工件坐标系的 Z 轴 (Zw) 的正方向移动。  
按压 [-Z(J3)] 键时沿着负方向移动。

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。

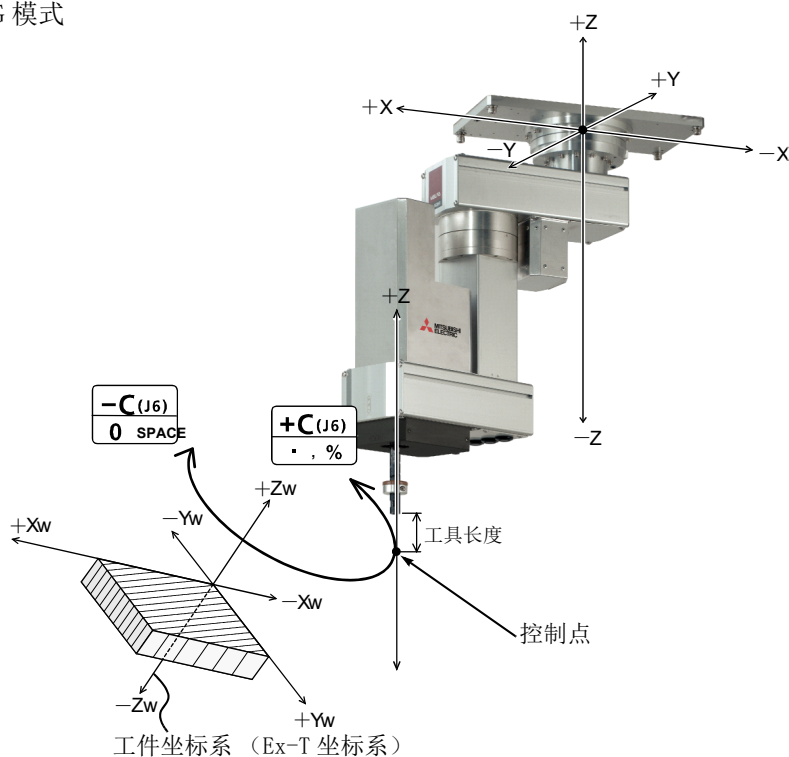
### 改变前端的方向

#### 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。

7.3.4 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。

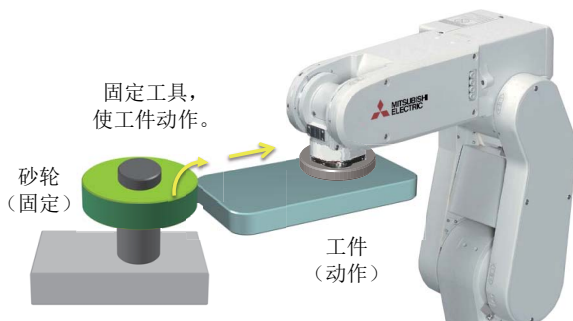
表 7-25: Ex-T 控制相关的指令

| 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 | 233  |
| 2   | 圆形插补  | EMvc (E Move C)   | 基于工件坐标 (Ex-T 坐标) 的圆形插补 | 227  |
| 3   | 圆弧插补  | EMvr (E Move R)   | 基于工件坐标 (Ex-T 坐标) 的圆弧插补 | 228  |
| 4   |       | EMvr2 (E Move R2) |                        | 230  |
| 5   |       | EMvr3 (E Move R3) |                        | 231  |

表 7-26: 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。 | 423  |

(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))

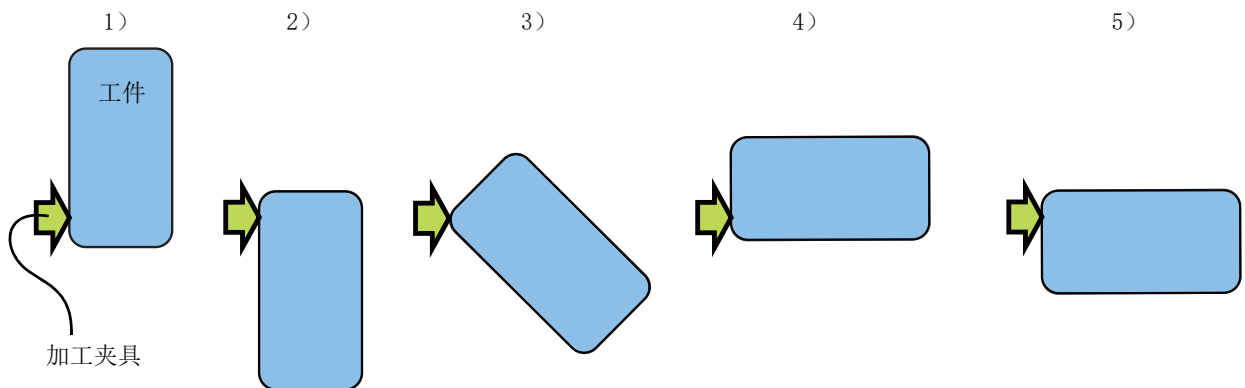


图 7-119: 作业示例 1

重叠该图则实际的动作如下所示。

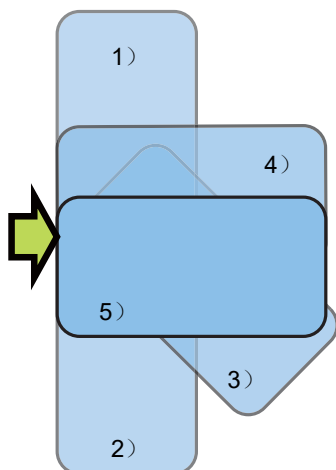
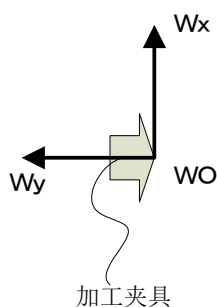


图 7-120：作业示例 2

### ■ 步骤 1：工件坐标（Ex-T 坐标）的设置

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。同时也需要根据变化情况更改程序。

## 7.4 关于高速位置获取功能

### 7.4.1 概要

高速位置获取功能（GPS 功能），是指通过高速监视来自外部传感器的输入信号，高精度地获取信号输入时的机器人的位置数据的功能。

通过正确获取外部传感器通过时的机器人位置数据，用于补偿工件位置的偏移的“校准补偿”及检测盒内工件有无信息的“映射处理”。

GPS 功能分为下表所示的两种功能。各功能使用方法的详细内容，请参照“(1) 获取传感器输入时的位置数据”、“(2) 判断盒内有无工件”。

表 7-27：GPS 功能一览

| No. | 功能            | 用途   |
|-----|---------------|------|
| 1   | 获取传感器输入时的位置数据 | 校准补偿 |
| 2   | 判断盒内有无工件      | 映射处理 |

### 7.4.2 GPS功能规格

GPS 功能规格如下表所示。

表 7-28：GPS 功能规格

| 项目            |             | 规格                                                          |
|---------------|-------------|-------------------------------------------------------------|
| 获取传感器输入时的位置数据 | 位置获取条件      | 位置获取条件通过 Def Gps 指令定义。                                      |
|               | 监视开始 / 结束控制 | 通过 Gps Chk 指令控制输入信号监视的开始 / 结束。                              |
|               | 位置数据锁存      | 将位置获取条件成立时的位置数据保存至状态变量 P_Gps1() ~ P_Gps8() 中。<br>(最多 400 个) |
| 判断盒内有无工件      | 工件判断条件      | 有无工件的判断条件通过 Def Map 指令定义。                                   |
|               | 监视开始 / 结束控制 | 通过 Gps Chk 指令控制输入信号监视的开始 / 结束。                              |
|               | 位置数据锁存      | 将位置获取条件成立时的位置数据保存至状态变量 P_Gps1() ~ P_Gps8() 中。<br>(最多 400 个) |

### 7.4.3 数字输入信号规格

数字输入信号规格如下表所示。

[注意] 伺服数字输入仅适用于 CR750 和 CR751 控制器。

表 7-29: 数字输入信号规格

| 项目     |                    | 规格值                    | 备注                                                                                                                              |
|--------|--------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 数字输入   | 频道数                | SKIP 输入: 3CH           |                                                                                                                                 |
|        | 连接器 <sup>注1)</sup> | CNUSR12 连接器            | 针脚名称 (针脚编号): 输入信号编号<br>SKIP21 (3)、SKIP22 (12): 输入信号 801<br>SKIP31 (2)、SKIP32 (11): 输入信号 802<br>SKIP41 (1)、SKIP42 (10): 输入信号 803 |
|        | 监视周期               | SKIP 输入: 888 $\mu$ s   |                                                                                                                                 |
| 信号输入条件 | 信号开启时              | DC18V ~ DC25.2V、9mA 以上 | 信号幅需达到 888 $\mu$ s 以上。                                                                                                          |
|        | 信号关闭时              | DC4V 以下、2mA 以下         |                                                                                                                                 |

注 1) 各连接器的位置请参照图 7-121。针脚分配的详细情况请参照另一手册“从控制器的安装和基本操作到保养”。

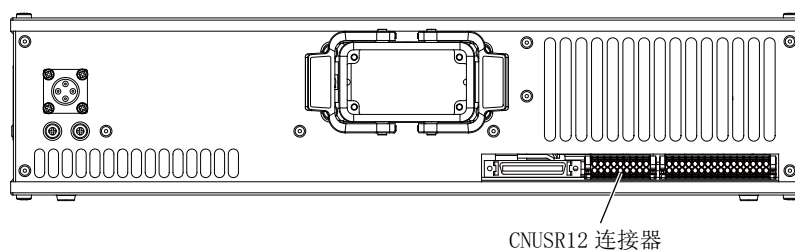


图 7-121: 连接器的位置

7.4.4 数字输入电气的规格

伺服 SKIP 输入的电气的规格如表 7-30 所示。

表 7-30：SKIP 输入电气的规格

| 项目     | 规格          | 内部电路 |
|--------|-------------|------|
| 形式     | DC 输入       |      |
| 输入点数   | 2           |      |
| 绝缘方式   | 光电耦合器绝缘     |      |
| 额定输入电压 | DC24V ± 10% |      |
| 额定输入电流 | 约 9mA       |      |
| 输入电阻   | 约 2.5k Ω    |      |
| 公共端方式  | 1 点 1 个公共端  |      |



### 7.4.5 GPS功能的使用方法

#### (1) 获取传感器输入时的位置数据

以下对获取信号输入时的位置数据时的基本的 MELFA BASIC VI 的程序记述方法进行说明。

表 7-31: MELFA BASIC VI 指令一览

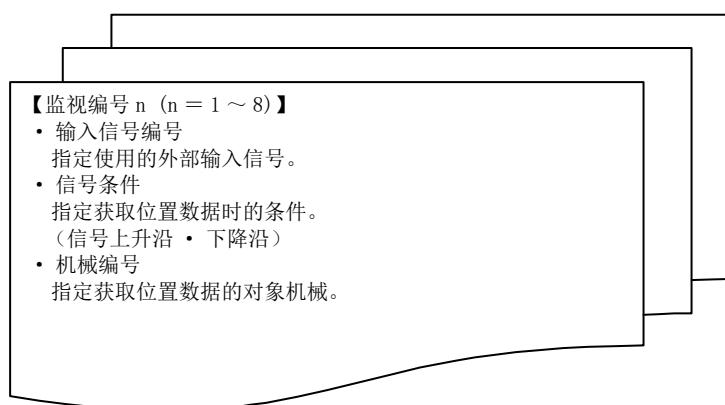
| 指令      | 说明                                  |
|---------|-------------------------------------|
| Def Gps | 通过 GPS 功能, 定义获取传感器输入时机器人的位置数据获取的条件。 |
| GpsChk  | 对已指定条件的监视的开始 / 结束进行指定。              |

表 7-32: 状态变量一览

| 变量名称                        | 排列要素数        | 内容                                         |
|-----------------------------|--------------|--------------------------------------------|
| M_Gps(n)<br>(n: 监视编号 1 ~ 8) | 1 ~ 8 (监视编号) | 保存指定监视编号 P_Gps1(x) ~ P_Gps8(x) 中储存的位置数据个数。 |
| P_Gps1(x) ~ P_Gps8(x)       | 400 (个)      | 通过 Def Gps 指令定义的条件成立时的机器人位置数据最多能保存 400 个。  |

#### 1) 监视条件的定义

首先使用 Def Gps 指令, 定义基于外部信号的输入的位置获取条件 (输入信号编号 · 信号条件 · 机械编号)。最多能设定 8 个不同的条件。各设定内容通过监视编号进行管理。



#### 2) 监视开始

执行 GpsChk On, n 指令时, 开始基于 Def Gps 指令定义的监视编号的监视, 并保存条件成立时下的位置数据。1 个监视编号最多能获取 400 个位置数据。

(n 对应监视编号 1 ~ 8。)

#### 3) 监视结束

执行 GpsChk Off, n 指令时, 结束监视编号对应的监视, 将获取的位置数据保存于状态变量 P\_Gps1() ~ P\_Gps8() 中。1 个监视编号最多能保存 400 个位置数据 (排列要素的数值对应位置数据的获取顺序)。

此外, 保存于 P\_Gps1() ~ P\_Gps8() 的位置数据个数将被保持于状态变量 M\_Gps(n) 中。M\_Gps(n)、P\_Gps1() ~ P\_Gps8() 中储存的数据, 会在下一次 GpsChk On, n 指令执行时清零。

(n 对应监视编号 1 ~ 8。)

## 〈样本程序〉

|                           |                                   |
|---------------------------|-----------------------------------|
| 1 Def Gps 1, 801, On, 1   | ‘ 监视编号 1 记录 801 号信号开启时机械 1 的位置数据。 |
| 2 GpsChk On, 1            | ‘ 监视编号 1 的条件监视开始                  |
| 3 Mvs P1                  | ‘ 向位置 P1 移动                       |
| 4 GpsChk Off, 1           | ‘ 监视编号 1 的条件监视结束 (保存所获得的位置数据)     |
| 5 M1=M_Gps(1)             | ‘ 获取保存于 P_Gps1 中的位置数据个数           |
| 6 If M1=0 Then Error 9000 | ‘ 若无法记录位置数据, 则发生错误 9000           |
| 7 P1=P_Gps1(1)            | ‘ 将 801 号信号首次开启时的位置代入 P1。         |
| 8 Hlt                     | ‘ 停止                              |

## (2) 判断盒内有无工件

以下对判断盒内有无工件信息 (映射处理) 时的基本的 MELFA BASIC VI 的程序记述方法进行说明。

表 7-33: MELFA BASIC VI 指令一览

| 指令      | 说明                      |
|---------|-------------------------|
| Def Map | 定义基于 GPS 功能判断盒内有无工件的条件。 |
| GpsChk  | 对已指定条件的监视的开始 / 结束进行指定。  |

表 7-34: 状态变量一览

| 变量名称                         | 排列要素数        | 内容                                           |
|------------------------------|--------------|----------------------------------------------|
| M_Gps (n)<br>(n: 监视编号 1 ~ 8) | 1 ~ 8 (监视编号) | 保存指定监视编号 P_Gps1(x) ~ P_Gps8(x) 中储存的位置数据个数。   |
| P_Gps1(x) ~ P_Gps8(x)        | 400 (个)      | 通过 Def Map 指令定义的输入编号条件成立时的机器人位置数据最多保存 400 个。 |
| M_Map1(x) ~ 8(x)             | 130 (段)      | 通过基于 Def Map 指令定义的条件, 工件存在的段数被保存为“1”。        |

## 1) 监视条件的定义

首先使用 Def Map 指令, 定义基于外部信号的输入的位置获取条件 (输入信号编号 · 信号条件 · 机械编号) 以及盒条件 (盒的坐标数据 · 段数等)。最多能设定 8 个不同的条件。各设定内容通过监视编号进行管理。

## 2) 监视开始

执行 GpsChk On, n 指令时, 开始基于 Def Map 指令定义的监视编号的监视, 并保存条件成立时的位置数据。1 个监视编号最多能获取 400 个位置数据。  
(n 对应监视编号 1 ~ 8。)

## 3) 监视结束

通过执行 GpsChk Off, n 指令时获得的位置数据和 Def Map 指令定义的盒形状 · 位置数据, 计算工件存在的段数, 并将结果保存至状态变量 M\_Map1() ~ M\_Map8() 中。  
此外, 所获取的位置数据个数将保持于状态变量 M\_Gps(n) 中。  
M\_Gps(n)、M\_Map1() ~ M\_Map8() 中储存的数据, 会在下一次 GpsChk On, n 指令执行时清零。  
(n 对应监视编号 1 ~ 8。)

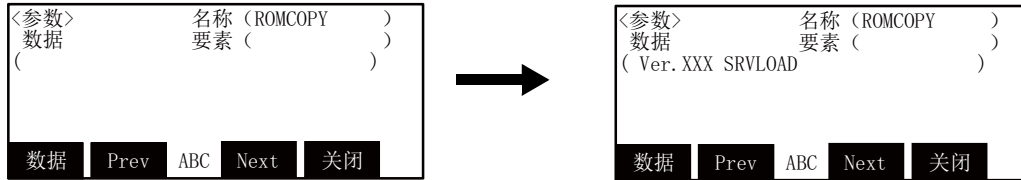
## 〈 样本程序 〉

|    |                                         |                                                                                                                      |
|----|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| 1  | Def Map 3, 851, On, 1, PC1, PC2, 20, 10 | ‘ 监视编号 3 将记录 851 号开启时机械 1 的位置数据，并执行基于所定义条件的映射处理<br>PC1: 盒第 1 段的位置, PC2: 盒最高段的位置, 20: 盒的段数 20 段, 10: 传感器灵敏度 10mm<br>宽 |
| 2  | Mov PM1                                 | ‘ 移动至映射开始位置                                                                                                          |
| 3  | GpsChk On, 3                            | ‘ 监视编号 3 的条件监视开始<br>记录 851 号开启时机械 1 的位置数据                                                                            |
| 4  | Mvs PM2                                 | ‘ 移动至映射结束位置                                                                                                          |
| 5  | GpsChk Off, 3                           | ‘ 监视编号 3 的条件监视结束<br>‘ 从记录的位置数据计算出存在工件的盒段，并将结果保存至 M_Map3                                                              |
| 6  | M1=M_Gps (3)                            | ‘ 获取保存于 P_Gps3 中的位置数据个数。                                                                                             |
| 7  | If M1=0 Then Error 9000                 | ‘ 若位置数据无法记录，则发生错误 9000                                                                                               |
| 8  | For M2=1 To 20                          | ‘                                                                                                                    |
| 9  | M_Out (6100+M2)=M_Map3 (M2)             | ‘ 映射结果从输出信号 6101 号开始依次输出                                                                                             |
| 10 | Next M2                                 | ‘ 重复盒的段数                                                                                                             |

### 7.5 伺服软件升级

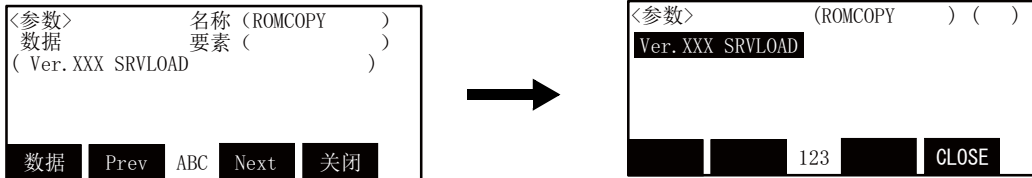
发生错误 H0099 时，需要升级伺服软件。  
 升级伺服软件时，可通过使用示教单元或 RT ToolBox3 进行操作。  
 使用示教单元升级伺服软件的操作步骤如下所示。

- 1) 显示参数画面，在名称中输入“ROMCOPY”并按 [EXE] 键。



名称的输入 [R]、[O]、[M]、[C]、[O]、[P]、[Y]  
 确定 [EXE]

- 2) 按下“数据”对应的功能键 ([F1])，显示参数输入画面。  
 在参数输入画面按下 [EXE] 键。



数据の設定 [F1]

确定 [EXE]

- 3) 确认错误 H0009 的显示后，进行控制器的电源复位。



确认显示

- 4) 控制器的蜂鸣音响起后，将控制器的电源设为 OFF。

至此，伺服软件升级完成。

使用 RT ToolBox3 时，可通过参数一览进行伺服软件的升级。

## 7.6 日志功能

以下对保存错误发生时的机器人信息的日志功能进行说明。

### 7.6.1 概要



可将错误发生时的机器人信息保存至 FTP 服务器及 SD 存储卡（仅限 CR800-D 系列）中。可保存的机器人信息有以下 4 种。

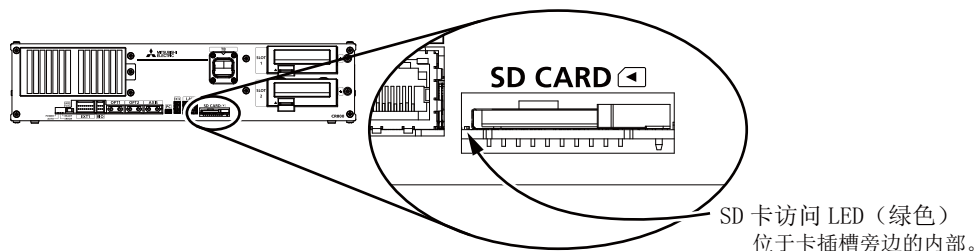
- 错误履历
- 事件履历
- 程序执行履历
- 机器人动作的内部信息

为了使用日志功能，需要进行参数的设定。

### ⚠ 注意

SD 卡访问 LED 的亮灯中，请勿取下 SD 存储卡，或将机器人控制器的电源设为 OFF。有可能导致 SD 存储卡及文件破损。

取下 SD 存储卡时，SD 存储卡可能会从卡插槽飞出，因此请以手做支撑取下。



### ⚠ 注意

CR800 系列控制器中，通过 SD 存储卡的 LOCK 开关进行的锁定设定将被忽略。即使进行锁定设定，也无法禁止向 SD 存储卡进行写入。

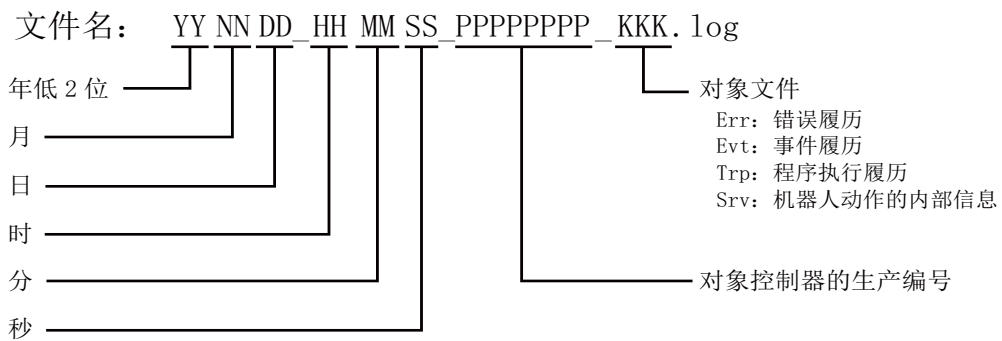
7.6.2 规格

(1) 基本规格

表 7-35：基本规格

| 项目        | 规格                                                                                                                        |
|-----------|---------------------------------------------------------------------------------------------------------------------------|
| 保存日志对象    | 以下4种。<br><ul style="list-style-type: none"> <li>• 错误履历</li> <li>• 事件履历</li> <li>• 程序执行履历</li> <li>• 机器人动作的内部信息</li> </ul> |
| 保存目标      | FTP 服务器或 SD 存储卡（仅限 CR800-D 系列）                                                                                            |
| 对象 SD 存储卡 | 2F-2GBSD（容量 2GB）                                                                                                          |

(2) 日志文件规格



例)

- 170701\_113502\_B12345678M\_Err.log
- 170701\_113502\_B12345678M\_Evt.log
- 170701\_113502\_B12345678M\_Trp.log
- 170701\_113502\_B12345678M\_Srv.log

SD 存储卡时，保存至日期的文件夹中。

表 7-36：日志内容

| 文件名          | 日志内容       | 内容确认方法                               |
|--------------|------------|--------------------------------------|
| xxxx_Err.log | 错误履历       | 可通过 RT ToolBox3（Ver1.10L 以上）的模拟进行确认。 |
| xxxx_Evt.log | 事件履历       | 可通过 RT ToolBox3（Ver1.00A 以上）的模拟进行确认。 |
| xxxx_Trp.log | 程序执行履历     | 可通过记事本等的文本编辑器进行确认。                   |
| xxxx_Srv.log | 机器人动作的内部信息 | -                                    |

## 7.6.3 日志保存对象的错误编号

表 7-37 中所示的错误从最初开始登录。追加错误时，请设定为参数 LOGTRGE。

表 7-37：日志内容

| 错误编号  | 错误内容          |
|-------|---------------|
| H0740 | 供电 主电路异常      |
| H0810 | 供电 欠电压        |
| H091n | 伺服放大器 过速度     |
| H092n | 伺服放大器 电源模块过电流 |
| H093n | 伺服放大器 电机过电流   |
| H096n | 伺服放大器 误差过大 1  |
| H101n | 冲突检知          |
| H1680 | 伺服 ON 超时      |

## 7.6.4 参数的设定

通过示教单元或 RT ToolBox3 设定参数。关于操作方法的详细内容，请参照各产品附带的使用说明书。

- R32TB/R33TB: 本说明书第 93 页的“3.15 参数画面的操作”
- R56TB/R57TB: R56TB/R57TB 使用说明书 (BFP-A8684: 英语版)
- RT ToolBox3: RT ToolBox3/RT ToolBox3 mini 使用说明书 (BFP-A3494: 日语版)

表 7-38：日志功能参数

| 参数                     | 参数名      | 序列号<br>字符数 | 内容说明                                                                                                                                                                                                                    | 出厂时设定值                                                |
|------------------------|----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| 错误日志自动传送               | LOGMODE1 | 整数 1       | 设定错误发生时的日志功能<br>1: 有效 (FTP)<br>2: 有效 (SD 存储卡)<br>※ 仅限 CR800-D 系列<br>0: 无效                                                                                                                                               | 0                                                     |
| 日志对象错误编号               | LOGTRGE  | 整数 16      | 设定日志保存的错误编号。<br>按错误编号、类型的顺序，可登录 8 种错误编号<br>类型<br>0: 错误编号直接作为对象<br>1: 错误编号的第低 1 位中输入轴编号<br>设定示例)<br>2000, 0, 1010, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,<br>“2000, 0,” 仅将错误 2000 作为对象。“1010, 1,” 将错误<br>1011~1018 作为对象。 | 0, 0, 0, 0, 0, 0,<br>0, 0, 0, 0, 0, 0,<br>0, 0, 0, 0, |
| FTP 传送 <sup>注 1)</sup> | FTPID    | 字符串 1      | 设定 FTP 通信中使用的用户 ID。<br>设定范围: 半角英文数字 (大写/小写) 8 个字符以内                                                                                                                                                                     | ftpuser                                               |
|                        | FTPPASS  | 字符串 1      | 设定 FTP 通信中使用的密码。<br>设定范围: 半角英文数字 (大写/小写)、<br>符号 (! # \$ % & = - @ . ? _ ) 16 个字符以内                                                                                                                                      | ftppassword                                           |
|                        | FTPSVRIP | 字符串 1      | 设定 FTP 通信中使用的 FTP 服务器的 IP 地址。                                                                                                                                                                                           | 192.168.0.99                                          |
|                        | FTPPATH  | 字符串 1      | 指定 FTP 传送目标的路径<br>设定范围: 半角英文数字 (大写/小写)、符号 (: /)                                                                                                                                                                         | 空白                                                    |

注 1) 使用 FTP 时需要设定。

## 7.6.5 日志文件内容的确认

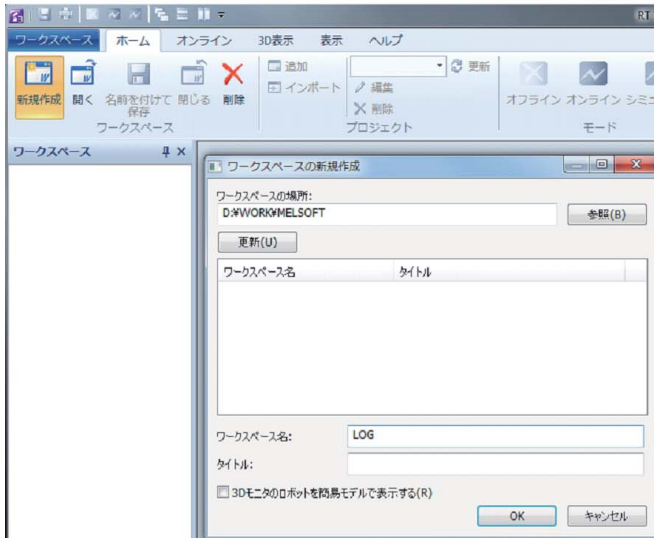
对各日志文件的确认方法进行说明。

## (1) 错误履历

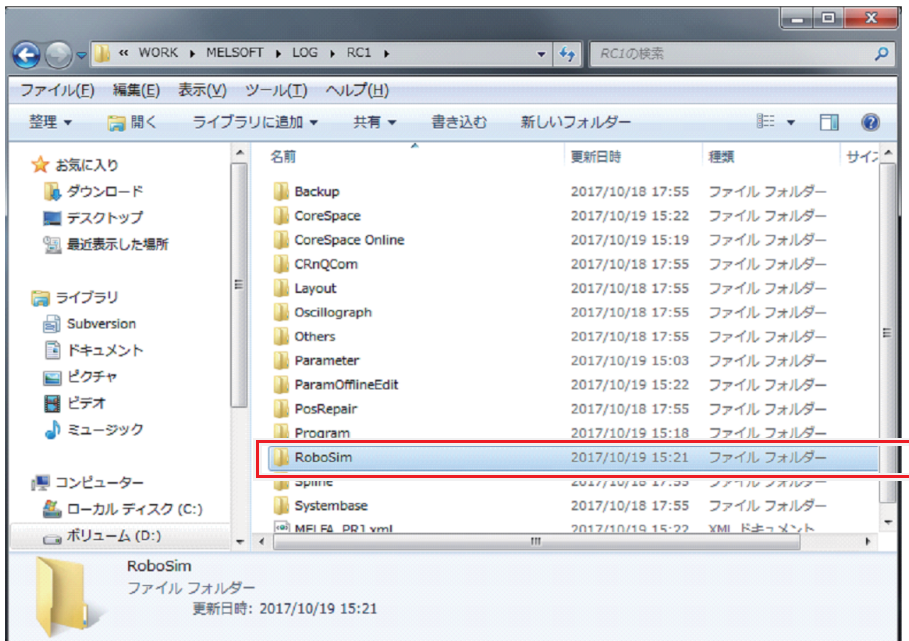
通过 RT ToolBox3 (Ver1.10L 以上) 的模拟进行确认。

## ■ 操作步骤

- 1) 将 xxxx\_Err.log 重命名为 AError.log。
- 2) 在 RT ToolBox3 中创建工作区。



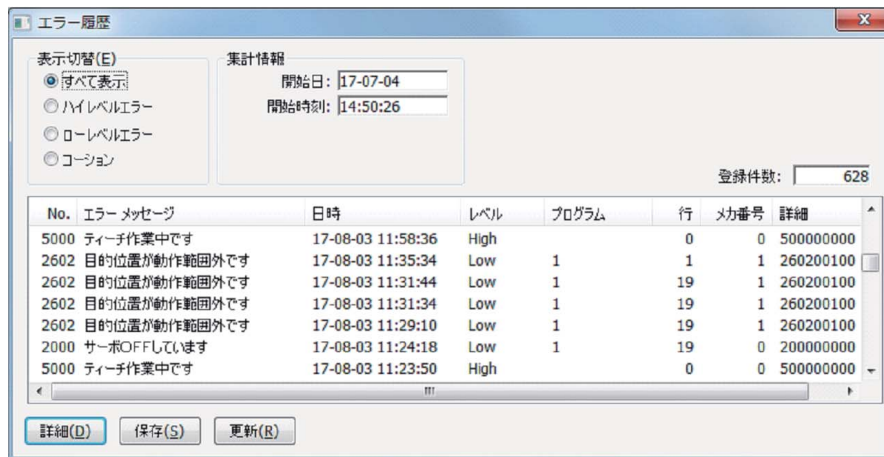
- 3) 将 AError.log 复制至模拟文件夹 “RoboSim” 中。



- 4) 启动模拟。



5) 在错误履历画面中确认内容。

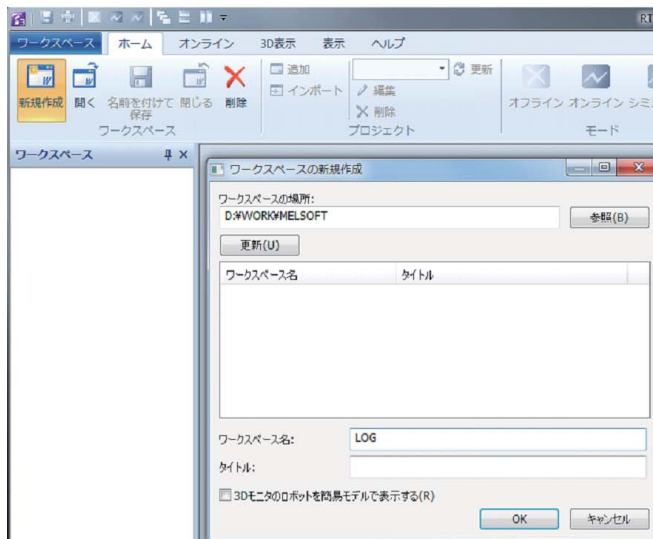


## (2) 事件履歴

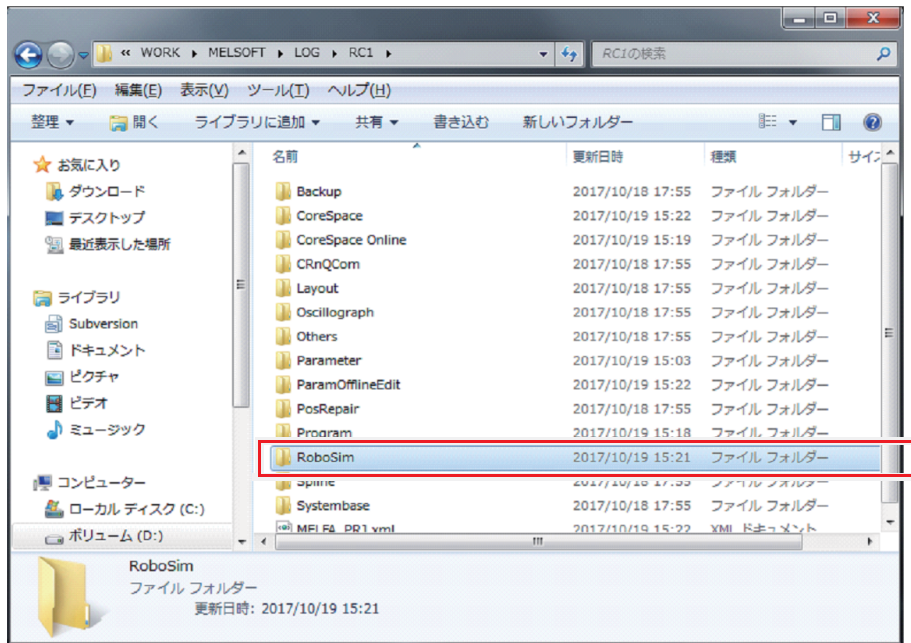
通过 RT ToolBox3 (Ver1.00A 以上) 的模拟进行确认。

### ■ 操作步骤

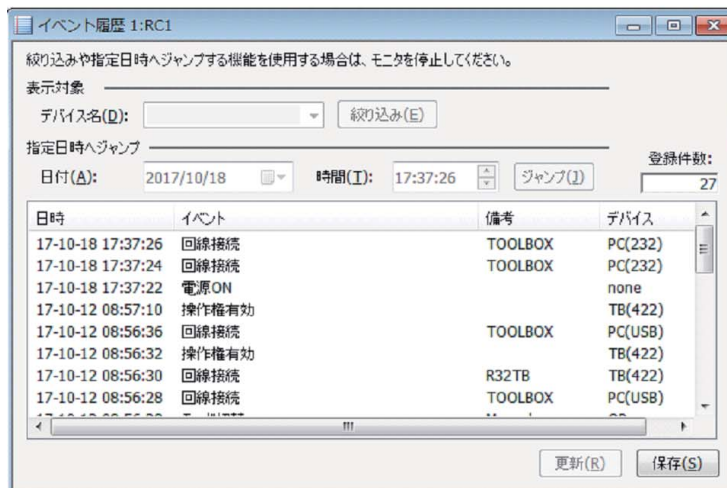
- 1) 将 xxxx\_Evt.log 重命名为 Event.log。
- 2) 在 RT ToolBox3 中创建工作区。



- 3) 将 Event.log 复制至模拟文件夹 “RoboSim” 中。



- 4) 启动模拟。  
5) 在事件履历画面中进行内容确认。



### (3) 程序执行履历

通过记事本等的文本编辑器进行确认。

#### ■ 操作步骤

- 1) 通过记事本等的文本编辑器打开 xxxx\_Trp. log。

如下所示，可确认程序的执行顺序。

```
NARC Trap system PROG log V1.0
17-10-18 11:34:42; sec/it=0.000889 ←开始数据采集的日期
17-10-18 11:37:18;0; ← 记录最终行数据的日期
0;659;PROG;5 ← <预备>; <连号>; <程序名>; <步号>
0;660;PROG;2
0;661;PROG;3
0;662;PROG;4
0;663;PROG;5
0;664;PROG;2
```

## 7.7 特殊设备

对应的特殊设备如下所示。

### 7.7.1 CR800-R系列

#### (1) 特殊继电器

对应的特殊继电器如下所示。

表 7-39: 机器人 CPU (R16RTCPU) 所响应的特殊继电器

| 编号    | 名称                   | 备注 |
|-------|----------------------|----|
| SM0   | 最新的自诊断错误 (包含报警器 ON)  |    |
| SM1   | 最新的自诊断错误 (不包含报警器 ON) |    |
| SM50  | 解除错误                 |    |
| SM51  | 电池低电量锁存              |    |
| SM52  | 电池电量低                |    |
| SM203 | STOP 接点              |    |
| SM204 | PAUSE 接点             |    |

#### (2) 特殊寄存器

对应的特殊寄存器如下所示。

表 7-40: 机器人 CPU (R16RTCPU) 所响应的特殊寄存器

| 编号    | 名称           | 备注                                        |
|-------|--------------|-------------------------------------------|
| SD0   | 最新的自诊断错误编码   | 发生机器人固有错误时, 编码 0x1800、0x3000、0x3001 将被保存。 |
| SD1   | 最新的自诊断错误发生时间 |                                           |
| SD2   | ↑            |                                           |
| SD3   | ↑            |                                           |
| SD4   | ↑            |                                           |
| SD5   | ↑            |                                           |
| SD6   | ↑            |                                           |
| SD7   | ↑            |                                           |
| SD10  | 自诊断错误编码 1    | 发生机器人固有错误时, 编码 0x1800、0x3000、0x3001 将被保存。 |
| SD11  | 自诊断错误编码 2    |                                           |
| SD12  | 自诊断错误编码 3    |                                           |
| SD13  | 自诊断错误编码 4    |                                           |
| SD14  | 自诊断错误编码 5    |                                           |
| SD15  | 自诊断错误编码 6    |                                           |
| SD16  | 自诊断错误编码 7    |                                           |
| SD17  | 自诊断错误编码 8    |                                           |
| SD18  | 自诊断错误编码 9    |                                           |
| SD19  | 自诊断错误编码 10   |                                           |
| SD20  | 自诊断错误编码 11   |                                           |
| SD21  | 自诊断错误编码 12   |                                           |
| SD22  | 自诊断错误编码 13   |                                           |
| SD23  | 自诊断错误编码 14   |                                           |
| SD24  | 自诊断错误编码 15   |                                           |
| SD25  | 自诊断错误编码 16   |                                           |
| SD201 | LED 状态       |                                           |

| 编号    | 名称         | 备注 |
|-------|------------|----|
| SD203 | CPU 运行状态   |    |
| SD228 | 多 CPU 系统信息 |    |
| SD229 | ↑          |    |
| SD230 | ↑          |    |
| SD231 | ↑          |    |
| SD232 | ↑          |    |
| SD233 | ↑          |    |
| SD260 | 位软元件分配点数   |    |
| SD261 | ↑          |    |
| SD262 | ↑          |    |
| SD263 | ↑          |    |
| SD264 | ↑          |    |
| SD265 | ↑          |    |
| SD280 | 字软元件分配点数   |    |
| SD281 | ↑          |    |

### 7.7.2 CR800-D系列

#### (1) 特殊继电器

对应的特殊继电器如下所示。

表 7-41: CR800-D 所响应的特殊继电器

| 编号    | 名称                   | 备注 |
|-------|----------------------|----|
| SM0   | 最新的自诊断错误 (包含报警器 ON)  |    |
| SM1   | 最新的自诊断错误 (不包含报警器 ON) |    |
| SM51  | 电池低电量锁存              |    |
| SM52  | 电池电量低                |    |
| SM203 | STOP 接点              |    |
| SM204 | PAUSE 接点             |    |

#### (2) 特殊寄存器

对应的特殊寄存器如下所示。

表 7-42: CR800-D 所响应的特殊寄存器

| 编号  | 名称           | 备注                                        |
|-----|--------------|-------------------------------------------|
| SD0 | 最新的自诊断错误编码   | 发生机器人固有错误时, 编码 0x1800、0x3000、0x3001 将被保存。 |
| SD1 | 最新的自诊断错误发生时间 |                                           |
| SD2 | ↑            |                                           |
| SD3 | ↑            |                                           |
| SD4 | ↑            |                                           |
| SD5 | ↑            |                                           |
| SD6 | ↑            |                                           |
| SD7 | ↑            |                                           |

| 编号    | 名称         | 备注                                       |
|-------|------------|------------------------------------------|
| SD10  | 自诊断错误编码 1  | 发生机器人固有错误时，编码 0x1800、0x3000、0x3001 将被保存。 |
| SD11  | 自诊断错误编码 2  |                                          |
| SD12  | 自诊断错误编码 3  |                                          |
| SD13  | 自诊断错误编码 4  |                                          |
| SD14  | 自诊断错误编码 5  |                                          |
| SD15  | 自诊断错误编码 6  |                                          |
| SD16  | 自诊断错误编码 7  |                                          |
| SD17  | 自诊断错误编码 8  |                                          |
| SD18  | 自诊断错误编码 9  |                                          |
| SD19  | 自诊断错误编码 10 |                                          |
| SD20  | 自诊断错误编码 11 |                                          |
| SD21  | 自诊断错误编码 12 |                                          |
| SD22  | 自诊断错误编码 13 |                                          |
| SD23  | 自诊断错误编码 14 |                                          |
| SD24  | 自诊断错误编码 15 |                                          |
| SD25  | 自诊断错误编码 16 |                                          |
| SD201 | LED 状态     |                                          |
| SD203 | CPU 运行状态   |                                          |
| SD260 | 位软元件分配点数   |                                          |
| SD261 | ↑          |                                          |
| SD262 | ↑          |                                          |
| SD263 | ↑          |                                          |
| SD264 | ↑          |                                          |
| SD265 | ↑          |                                          |
| SD280 | 字软元件分配点数   |                                          |
| SD281 | ↑          |                                          |

### 7.7.3 CR800-Q系列

#### (1) 特殊继电器

对应的特殊继电器如下所示。

表 7-43：机器人 CPU（Q172DSRCPU）所响应的特殊继电器

| 编号    | 名称                  | 备注 |
|-------|---------------------|----|
| SM0   | 最新的自诊断错误（包含报警器 ON）  |    |
| SM1   | 最新的自诊断错误（不包含报警器 ON） |    |
| SM50  | 电池低电量锁存             |    |
| SM51  | 电池电量低               |    |
| SM52  | STOP 接点             |    |
| SM53  | PAUSE 接点            |    |
| SM203 | STOP 接点             |    |
| SM204 | PAUSE 接点            |    |

## (2) 特殊寄存器

对应的特殊寄存器如下所示。

表 7-44: 机器人 CPU (Q172DSRCPU) 所响应的特殊寄存器

| 编号    | 名称                | 备注                                                               |
|-------|-------------------|------------------------------------------------------------------|
| SD0   | 最新的自诊断错误编码        | 发生机器人固有错误时糊编码 12000 (H 不可复位)、12001 (H)、12002 (L)、12003 (C) 将被保存。 |
| SD1   | 最新的自诊断错误发生时间      |                                                                  |
| SD2   | ↑                 |                                                                  |
| SD3   | ↑                 |                                                                  |
| SD4   | 错误信息区分编码 (号机 No)  |                                                                  |
| SD5   | 错误通用信息编码 (号机 No)  |                                                                  |
| SD6   | 错误通用信息编码 (I/O No) |                                                                  |
| SD51  | 电池低电量锁存           |                                                                  |
| SD52  | 电池电量低             |                                                                  |
| SD53  | 瞬停检测次数            |                                                                  |
| SD200 | I/O 操作权           |                                                                  |
| SD201 | LED 状态            |                                                                  |
| SD203 | CPU 运行状态          |                                                                  |
| SD290 | 软元件分配             |                                                                  |
| SD291 | ↑                 |                                                                  |
| SD292 | ↑                 |                                                                  |
| SD293 | ↑                 |                                                                  |
| SD294 | ↑                 |                                                                  |
| SD295 | ↑                 |                                                                  |
| SD296 | ↑                 |                                                                  |
| SD297 | ↑                 |                                                                  |
| SD298 | ↑                 |                                                                  |
| SD299 | ↑                 |                                                                  |
| SD300 | ↑                 |                                                                  |
| SD301 | ↑                 |                                                                  |
| SD302 | ↑                 |                                                                  |
| SD303 | ↑                 |                                                                  |
| SD304 | ↑                 |                                                                  |
| SD305 | ↑                 |                                                                  |
| SD393 | 多 CPU 系统信息        |                                                                  |
| SD395 | ↑                 |                                                                  |
| SD396 | ↑                 |                                                                  |
| SD397 | ↑                 |                                                                  |
| SD398 | ↑                 |                                                                  |
| SD399 | ↑                 |                                                                  |







## 三菱电机自动化(中国)有限公司

地址：上海市虹桥路1386号三菱电机自动化中心

邮编：200336

电话：86-21-2322-3030 传真：86-21-2322-3000

官网：<https://www.MitsubishiElectric-FA.cn>

技术支持热线 **400-821-3030**



内容如有更改 恕不另行通知

此印刷物发行于 2024 年 3 月，内容如有变动恕不另外通知。